On Privacy–Awareness in Social Networking Services

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.–Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von Dipl.–Inform. Alexander Korth

geb. am 11. September 1974 in Minden

Gutachter:
Prof. Dr. Andreas Nürnberger
Prof. Dr. Christian Bauckhage
Prof. Dr. Andreas Hotho

Ort und Datum des Promotionskolloquiums: Magdeburg, 16. Dezember 2011

Gewidmet meinen Eltern Karin und Hans–Peter,
die mich stets nach Kräften unterstützen.

Zusammenfassung

Kurzform

Die vorliegende Arbeit befasst sich mit der Privatsphäre von Nutzern in sozialen Netzwerken. Sie beleuchtet die involvierten Problembereiche aus interdisziplinärer Sicht und leitet Anforderungen an soziale Netzwerke her, um diesen zu entgegenzuwirken. Ein prototypischer Ansatz zur Adressierung der Teilprobleme von Transparenz und Kontrolle von persönlichen Informationen wird vorgestellt und evaluiert.

Langform

Soziale Netzwerke, auch Online Communities genannt, haben in den letzten Jahren regen Ansturm verzeichnet: Heutzutage sind mehr als eine halbe Milliarde Menschen Mitglieder dieser digitalen Gemeinschaften. Menschen verbinden sich in diesen Communities mit anderen Menschen um sozial zu interagieren. Die grundlegenden Gesetze dieser Interaktion sind soziale Normen, die die Menschen aus dem echten Leben mitbringen und in der digitalen Welt intuitiv anwenden. Beispiele sind der durch den Anbieter suggerierte und durch den Nutzer subjektiv empfundene soziale Kontext der Kommunikation, also die Empfängerschaft Nutzer-publizierter Information, sowie die daraus resultierenden Implikationen auf den Kommunikationskontext, also den Inhalt der Information. Heutige Community-Plattformen werden dem Anspruch an den Schutz der Privatsphäre der Nutzer in diesem und anderen Bereichen nicht gerecht. Mangelnde Funktionen zur Kontrolle und Transparenz über Nutzer-publizierte Informationen sowie weitere Faktoren führen zu Missverständnissen und Verletzungen der Privatsphären der Nutzer, herbei geführt sowohl durch den Anbieter als auch durch die Nutzer selbst.

Diese Arbeit beginnt mit der Entwicklung einer Taxonomie für die Problembereiche, die allesamt auf ihre Weise zur Verletzung der Privatsphäre der Nutzer beitragen. Ein grundlegender Bereich ist beispielsweise der Nutzer selbst, der durch das Veröffentlichen von Informationen auf der Entscheidungsbasis fehl eingeschätzten Kontexts sich selbst und anderen Schaden zufügt. Eine weitere Quelle von Problemen sind die Betreiber von Communities, die den Schutz der Nutzer-Privatsphäre der Zielerreichung des Wachstums und der Maximierung des Netzwerkeffekts unterordnen. Die Folge sind nicht vorhandene oder mangelhaft nutzbare Werkzeuge zum Schutz der Privatsphäre. Ein abschließendes Beispiel eines Problembereichs sind Software-Mängel: Nicht vorhandene Zugriffskontrollmechanismen und Software-Fehler erlauben unautorisierten Personen den Zugriff auf private Daten der Nutzer.

Die Arbeit fährt fort mit der Aufgabe, bestehende wissenschaftliche Ansätze aus verschiedenen Disziplinen und Themenbereichen auf ihr Potenzial zur Lösung der aufgeführten Probleme zu gruppieren und zu prüfen. Diese Ansätze beinhalten

unter Anderem ökonomische Modelle zur theoretischen Erklärung von scheinbar irrationalem Benutzerverhalten, theoretische Arbeiten betreffend der gesetzlichen Regulation des Marktes, algorithmische Ansätze zur Kontrolle der Reichweite in der Streuung von Inhalten, ganzheitliche, zentralistische und auch verteilte Ansätze für Software-Architekturen von Online Communities, Ansätze mit dem Fokus auf grafischen Nutzeroberflächen und der Nutzbarkeit derer und Arbeiten bezüglich Datenzugriffs- sowie Datenhandhabungsrichtlinien.

Auf Basis der erfassten Problembereiche sowie der wissenschaftlichen Ansätze wird im Folgenden eine Anforderungsanalyse für soziale Netzwerke erstellt, die im Falle ihrer Erfüllung die Privatsphäre der Nutzer schützt. Eine Auswahl von vier zeitgenössischen sozialen Netzwerken wird auf die Qualität der Erfüllung dieser Anforderungen evaluiert.

Die Arbeit fährt fort mit dem Ziel einen eigenen prototypischen Ansatz einer Plattform für den Großteil der Anforderungen zu entwerfen und konzentriert sich dabei auf die Teilprobleme der Kontrolle und Transparenz. Nutzers benötigen diese Aspekte um sozialen Kontext und Kommunikationskontext nutzen zu können und somit die eigene und die Privatsphäre anderer Nutzer zu schützen. Dazu wird eine Plattform bestehend aus einem Software-Framework und einer darauf basierenden Applikation entworfen und vorgestellt, die diesem Anspruch gerecht wird. Die vorgestellte Lösung ist ein eigenständiges und voll funktionales soziales Netzwerk. Die besonderen Herausforderungen an die Zugriffskontrollmechnismen im Backend sowie die benötigten Kontroll- und Transparenz-Funktionen in Backend und Frontend werden detailliert vorgestellt.

Die Arbeit evaluiert die Plattform in dreierlei Hinsicht: Zum Einen in einer empirischen Untersuchung an den Nutzern, welche befragt werden, ob sie die Kontroll- und Transparenz-Funktionen bemerkt, verstanden, zu schätzen gewusst und genutzt haben. Die Aussagen werden mit denen über andere soziale Netzwerke verglichen. Zum Anderen wird untersucht inwieweit sich der auf der Plattform entwickelte soziale Graph aus Nutzern und Gruppen von denen anderer sozialer Netzwerke unterscheidet, welche Bedeutung das hat und welche Gründe dazu führen. Zu guter Letzt wird die Plattform bezüglich ihrer Erfüllungsqualität der Anforderungen an den Schutz der Privatsphäre geprüft und mit anderen Plattformen verglichen.

Die Arbeit schließt ab mit einer Diskussion über ihren Ansatz, die Ergebnisse und Erkenntnisse gefolgt von einem Ausblick auf die zukünftige Entwicklung des Phänomens sozialer Netzwerke.

Abstract

Brief Version

This work deals with the area of user privacy in online social networks. It analyzes relevant problem areas from an interdisciplinary viewpoint and derives requirements for social platforms to address these. A prototypic approach for addressing the issues of transparency and control over personal information is introduced and evaluated.

Long Version

Social Networking Services, also called online social networks, have grown very large in recent years: Today, more than half a billion people are members of these digital communities. Within them, people connect with other people for social interaction. The basic rules of these interactions are social norms, which people bring from their real lives and apply intuitively. Examples of these are the social context, i.e. the audience and receivers of user-generated content, and the resulting implications on the communication context, i.e. the communicated content. These rules are suggested by the platform provider and perceived subjectively by users. Contemporary social networks do not meet the requirements of protecting its users' privacies in these and other aspects. They lack features to control or to make transparent the recipients of user-generated content as well as other factors. This leads to misunderstandings, misconceptions and, in the end, privacy violations caused by platform providers and users themselves.

The work starts with the development of a novel taxonomy of problem areas that account for user privacy violations. A basic area, for instance, is the users themselves, who publish information on the basis of a misestimated context and thereby hurt their own and the privacy of others. Another source of problems are the providers of communities, who subordinate user privacy to the growth and network effect of their platforms. The results are non-existent or intentionally unusable tools for privacy protection. A final problem area example is software flaws: Non-existent access control mechanisms and software bugs allow unauthorized parties access to private information about users.

The thesis proceeds with the attempt to map existing works from different scientific areas and disciplines to the named problem areas and to evaluate them on their potential to resolve them. Inter alia, these approaches comprehend economic models to explain seemingly irrational user behavior, theoretic works that regard to legal regulation of the market, algorithmic approaches to control the reach of spread content, holistic, centralized and also distributed approaches for software architectures for communities, works focusing on graphical user interfaces and their usability, and works dealing with data access and data handling policies.

Based on the problem areas and the related works, a list of requirements is defined to protect user privacies in online social networks. A selection of four contemporary online social networks is chosen and evaluated regarding the quality of their fulfillment of these requirements.

The thesis continues with the goal of creating a prototypic approach for a platform to address the majority of requirements and therefore, focuses on the sub-problems of control and transparency. Users need these aspects when publishing private information to protect their own and the privacy of others. Control and transparency are essentially needed to communicate and make social context and communication context available to the user. Therefore, a prototypic platform, consisting of a software framework and an application, to fulfill these requirements is engineered and presented. The result is a fully functional social network. Special challenges, such as the access control engine and the required transparency and access control features located at the frontend and backend, are presented in detail.

The work evaluates its success on three levels: Firstly, in an empiric survey users were asked if they noticed, understood, admired and used the provided tools for control and transparency. The answers are discussed and measured against those of other contemporary social networks. Secondly, it is evaluated if and why the social graph that emerged behind the scenes of the solution differs from those of other social networks. Finally, the platform's requirements fulfillment quality is measured and compared to the scores of other platforms.

Finally, the work closes with a discussion of its approach, results and insights followed by an outlook on the future development of the phenomenon of social networks.

Contents

List of Figures

List of Tables

1. Introduction

Today, we see a Web that has changed dramatically within the past ten years. In parallel to conventional Websites that deliver quasi-static, editorial content in an uni-directional manner, site types and techniques have emerged that turn users into the main content contributors. Starting from blogs that empower readers with a reverse channel for comments on articles, today Social Media sites (formally referred to as Web 2.0 sites) concentrate on how to provide an eco-system that enables and motivates users to create the sites' content by themselves, and voluntarily. A basic site type are Content-Sharing Sites that deal with a particular content type, such as video, audio, encyclopedic articles or photos. Users create this content by uploading or authoring it. The site type that this thesis focuses on are Social Networking Services (SNS), which provide the users themselves as main content type through user profile pages to navigate through. On these, users populate their own profile with content, link to acquaintances' and friends' profile pages, and start to socialize via communication features.

All these sites have in common a paradigm shift that turns consumers into content producers and contributors. The term Social Media does not only stand for equipping users with a voice or a tool to contribute, it refers to giving sites a social nuance by making users visible as individuals and allowing them to communicate with each other.

The paradigm shift also has its downsides: The new tools for content creation and social interaction are easily misunderstood or abused by users, which often results in harm to the privacy of others. Confidential information can easily or accidentally be undisclosed to friends of a user by writing on his SNS profile page, awkward photos can be marked to link to an illustrated user, user comments can verbally attack other users.

Users have a basic need for socialization, which motivates them to communicate. Generally, social interaction and communication naturally constitute a threat to privacy which humans know how to control by learned social norms. To communicate appropriately in the new social online world, users intuitively try to adapt social communication norms from the offline world. Unfortunately, the online world usually only provides insufficient features to do so. However, this lack of features or innovation does not happen by coincidence. The basic reason why the problems with privacy cannot be solved in a simple way has an economic background. Users and providers are driven by conflicting interests. As users want to satisfy their need for socialization and at the same time protect their privacies [89], providers tend to maximize the growth of their content-base, for which they seek to maximize the network effect of their application [116].

The research work presented in this thesis is dedicated to analyze and to improve user privacies in the area of Social Networking Services. More specifically, this work addresses a range of research challenges in this area: It presents a requirements catalogue, and an prototypic approach to address a selection of these.

1.1. Research Contribution and Outline

The field of privacy in the online world has been addressed by a broad range of research disciplines such as sociology, economics, and computer science, especially machine learning, human-computer interaction and Semantic Web. The approaches range from social studies to understand users, guidelines for improvements, design of better user interfaces to holistic software-architectural approaches, computational analysis of social network graphs and automation of tasks and settings for privacy protection.

The work presented in this thesis has been conducted from a software architectural perspective, but examines the topic from viewpoints beyond the scope of the discipline of computer science. In the following, the structure of this thesis is presented and its broad research contribution is highlighted.

Chapter 2 briefly introduces fundamental topics from different research disciplines. Starting with human needs and an introduction to privacy as such, sites types of Social Media are outlined and theoretic background on user motivation to contribute is given. Tools to build such systems and measures to analyze emerging social network graphs are outlined.

Research Question 1: What are the areas and sources of problems regarding user privacies on Social Networking Services?

Research Question 1 is addressed in Chapter 3, which introduces and explains a novel problem taxonomy to describe the heterogenous areas as sources for today's privacy-related problems in SNS. The taxonomy is the *first minor research contribution* of this thesis and structures the problem space from both the user and provider perspective, defines and discusses problem area and relates them to both the inducing entity and related problem areas.

Chapter 4 reviews numerous publications for approaches related to the identified problem areas from different research disciplines. Thematic topic clusters are built to group the works and put in relation to the problem areas of the problems taxonomy.

Research Question 2: What are the requirements a Social Networking Service has to fulfill in order to provide effective privacy protection?

We target Research Question 2 in Chapter 5. A privacy-related requirements list that is exhaustive regarding the defined problem areas is defined in this chapter and constitutes the *first major research contribution* of this thesis. A selection of contemporary SNSs is evaluated on the quality of requirements fulfillment.

As a basis for answering the remaining research questions, we introduce a prototypic approach for a privacy-preserving SNS platform in Chapter 6. The platform acts as a proof of concept for almost all privacy-related requirements defined in Chapter 5 and represents the *second major research contribution* of this thesis. Starting with a concept and a scenario, application-specific requirements are defined and, combined with the privacy-related requirements, translated to a feature list. A framework and an application are designed and implemented to address these features. The peculiarities regarding privacy protection, e.g. particular software design challenges and their resolutions, are described in detail.

Research Question 3: Does the approach empower users with transparency and control to protect their privacies online when communicating online?

Research Question 4: Do the transparency and control features provided emerge in more relevant content for the recipients?

Research Question 5: How will the sociogram that emerged from user activity differentiate from other SNSs?

After describing the corpus that has been built through user activity on the platform, which is the *second minor research contribution* of this thesis, and giving a brief overview of the platform traffic, Chapter 7 evaluates the prototypic approach introduced in Chapter 6 from different perspectives. To provide insights for Research Questions 3 and 4, a quantitative user survey was conducted. Its results are presented and discussed in this chapter. To answer Research Questions 5, the platform's resulting social network graph is briefly analyzed and compared to those of other SNS.

Chapter 8 discusses findings that were observed during runtime, suggestions for short-term improvements of the platform, and recommendations for both platform-related and general future work. Chapter 9 finalizes this thesis by outlining future market trends and a long-term scenario for a future Web.

2. Fundamentals

This chapter gives an overview over fundamental concepts and techniques used throughout this work. As the analysises and approaches that is reported of in the following chapters are originated from various disciplines, the following concepts are grouped by these, i.e. sociology, economics and computer science.

For the sociological perspective, we start with a brief discussion of human's basic need for socialization and the resulting potential damages to people's privacies. Furthermore, a definition and overview over the concept of privacy is given. For the economic viewpoint, the status quo of Web applications that empower users to socialize online is introduced and market mechanics are explained. For the technical point of view to create and analyze such platforms, the concepts of Social Network Analysis and the Semantic Web are explained. Following this, the theoretical software architectural concepts are introduced: Firstly, the Model View Controller programming pattern to architecturally structure complex software applications. Secondly, regarding human-computer interaction, a norm for software dialog principles as well as several disciplines to improve information presentation and navigation are explained.

2.1. Human Needs and Privacy

This section contains selected theoretic basics concerning user motivation and privacy from the discipline of sociology.

2.1.1. Socialization and Communication

Humans are social beings. They have social needs that have to be satisfied. A very basic social need is our sense of belonging, for the satisfaction of which we have family, friends and partners. Another social need is esteem. To serve this need, we seek for achievements, confidence and respect. Aristotle in his theory of the good life for humans (Eudaimonia) states that to have a good, successful and happy life, humans need to socialize with others. For a person to flourish, the social life and friendships in a community is a necessity.

Abraham Maslow developed a hierarchy of human needs in 1943 [101] (Figure 2.1). He positions social needs as immediately following psychological and safety needs, i.e. inside the layers for love, belonging, and esteem.

Consequently, humans do not only exist for themselves, but also for others. Thereby we play our role in a network of relations between humans. Thus, we are partially formed by our social connections [61].

When communicating with others, there are many rules, factors and behavioral patterns humans have learned, follow and act according to. They affect *what* information is communicated to *whom*.

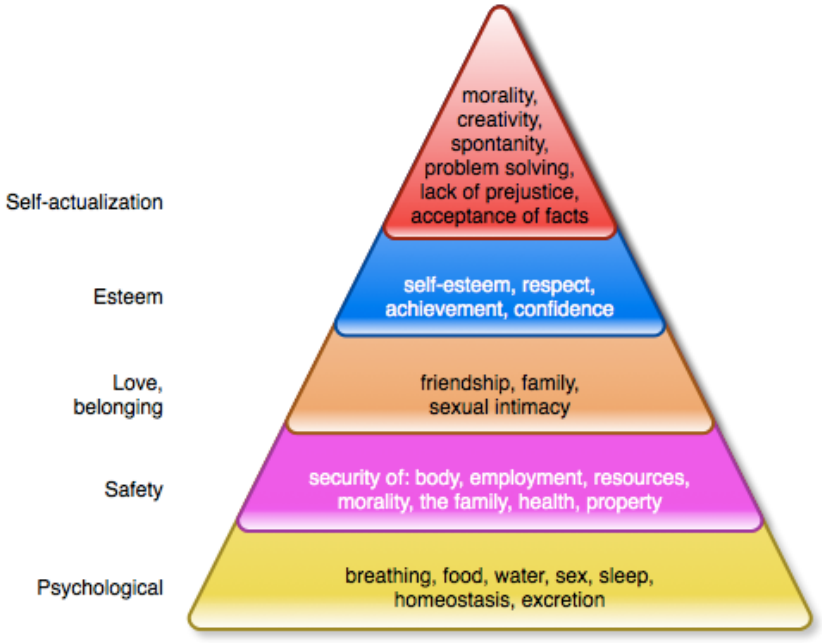

Figure 2.1.: Maslow's hierarchy of needs. The human need to socialize and communicate is located in the layers of love and belonging, and esteem.

Some rules are physical, e.g. the physical limits of the sound of the spoken word to be received by others. Speaking to someone at a certain volume lets us control the receivers of our information. It may be acceptable that the neighboring table hears a communication, but others are most certainly excluded. Also, spoken information is most of the times synchronous and ephemeral, i.e. usually not persisted and available only here and now.

Some rules are social norms. We know how a message is dealt with in our society. When writing a personal message to someone (an email, a telegram or a note left at the reception of a hotel) we know and accept that some service personnel will be granted access to the message's content by processing it, others, however, will not. A letter allows people to control the receivers even better, since it is contained in an envelope. It might mistakingly be opened by the wrong person but, however, people know that.

Once able to control the audience for a communicated information, we adjust its contents by questions like: Was this topic already discussed before? What language is appropriate or polite for this situation? How must the content be formulated for that specific receiver to achieve or to avoid a certain effect, e.g. fear? All these factors affect the *what* and *who* of communication.

As stated above, what information is communicated to whom is affected by social rules and factors. Prior works by Mani and Choudhury et al. have defined two context areas to cover the aspects and drivers of social communication:

Communication Context is a set of attributes that affect communication between two individuals [97]. Choudhury et al. [37] define further sub-attributes as:

- **Neighborhood context** to describe the effect of an communicated information among the receivers.
- **Topic context** to describe the effect of the publishers of past communication on the topic.
- **Recipient context** to define the effect of the person of the receiver on the information to be communicated.

Communication Context accordingly describes the effect of content and receivers on the *content* of the information to be published.

Social Context is a the set of attributes that refers to who is communicating with whom and the influence of the strength of the relationship shared between them [36]. Choudhury et al. define two basic attributes affecting Social Context:

- **Information roles** describe the role of the publisher: There are generators, mediators, and receptors.
- **Strong and weak ties** refer to the nature of relationship between two individuals affecting their communication.

Social Context therefore affects to *whom* information is communicated.

However, communication is a complex task. In the offline world, people have learned to control Social Context and Communication Context. Providing an online equivalent is not easy to achieve and mainly remains an unresolved question. How

missing tools to control these contexts as well as conflicting interests of participating parties cause problems by threatening people's privacies online is reported of in the following chapter. The next section, however, puts privacy into relation to socialization.

2.1.2. Privacy

This section provides a definition of the meaning of privacy used in this work, followed by an overview of the circumstances under which privacy is considered to be harmed, possibles sources to privacy threats, and types of privacies.

Although there is no single definition or meaning of the term [138], privacy (from Latin: *privatus*) is often understood as the ability of individuals (or groups) to selectively reveal information about themselves to others. In his book Privacy and Freedom [143], Alan Westin defines privacy as follows:

> Privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others.
> (Alan Westin [143])

From a cultural point of view, the object and degree of what is considered private or what constitutes an invasion of privacy differs slightly between different cultures, though remaining comparable in principle. In many countries, citizens' privacies are protected by law to a certain degree. These laws prohibit the unsanctioned invasion of privacies by other individuals, companies, or the government. Some laws also limit privacy, e.g., taxation laws which make the citizen's agreement on accessing income and earning information obsolete [138].

Types of privacy include (but are not limited to):

Physical privacy refers to the intrusion or the protection of an individual's physical space or solitude.

Organizational privacy deals with the protection of governmental or corporate information and knowledge.

Informational privacy concerns the handling of information about a person and is aimed at protecting reputation, avoiding personal embarrassment, and allowing control who is granted access to information. It is this type of privacy the remaining work deals with.

Regarding sources to privacy breaches, it is not only third persons or parties who cause harm to privacies. It may also be the people themselves who choose to voluntarily sacrifice parts of their privacies for some benefit, e.g., by exposing personal information in order to participate in a lottery. Also, to purchase goods online, personal information is asked for by the commercial service.

However, the degree to which exposure of personal information harms privacy depends on how (and whether) it is publicly received regarding place and time. As defined by Westin, privacy threat is thus not only a question *what* information is exposed, but also to *whom* and *when*.

New technologies always came with new ways of threatening people's privacies. E.g., printing technology made possible the spreading of information and photographies simultaneously to numerous recipients [142]. It changed how information could flow to *whom* and *when*. Information technology brought new dimensions of how information can be gathered and recorded. Also, the inferring of knowledge from stored information is a factor affecting *what* information is conditioned or computed and made accessible to *whom*. Web technology, where services are run remotely on providers' servers and used by countless customers worldwide, allows companies to centrally collect all personal and interaction data from their users and store them for an unlimited amount of time. With minimal means for users to overlook and control the data stored about them, these information are maintained, conditioned and sold to business partners. It remains unclear what knowledge companies will be able to derive in the future from the information stored today. To date, there is no international standard, no commonly agreed-on rules, and no world-wide regulation of how people's informational privacies are protected or dealt with.

Having said all that, this work analyzes contemporary problems regarding people's informational privacies in the realm of Web technology and a prototypic approach to solving parts of these problems.

2.2. Site Types and Market Mechanics

This section introduces Social Media and its site types, which enable users to socialize and communicate on the Internet. It briefly provides the reader with background information about market mechanics.

2.2.1. Social Media or Web 2.0

> Web 2.0 is a set of economic, social, and technology trends that collectively form the basis for the next generation of the Internet – a more mature, distinctive medium characterized by user participation, openness, and network effects.
> (Tim O'Reilly [108])

The term Web 2.0 refers to a paradigm shift in Websites that change from static, uni-directional deliverers of content to a generation of bidirectional, user-involving, sometimes user-centric sites. This new generation of Websites enable its users to participate in terms of generating and sharing own content, and to communicate and collaborate with others. Web 2.0 is nowadays referred to as Social Media. Web applications that provide social features are often called social applications.

According to O'Reilly, Web 2.0 is not a technology, but an attitude that services and applications provide collectively. The following list reports O'Reilly's core principles of Web 2.0 [116]:

The Web as a platform: Positioning a service not separately, but as part of a platform increases its value according to the Network Effect [75]. Service providers utilize the Web by providing lightweight programming models (see below) to increase the reach of their service.

Harnessing collective intelligence by enabling users to create, share, improve and correct content as a basic idea of Web 2.0.

Data-drivenness: The biggest value of a service is its data. The providers aim to create a database that is hard to recreate by competitors, and seek to improve both, the quality and the quantity of it.

End of the software release cycle: To provide a service running on a central server system on the Web instead of a boxed product has many advantages. The most appealing one is that updates can be run and delivered at any time. This ease implies the possibility of releasing software at a less mature level to test it on users at an early stage. The term *perpetual beta* was coined, because many sites were launched early and perpetually improved via user feedback.

Lightweight programming models: Supporting a lightweight programming model for loosely coupled systems enables amateur developers to easily re-use data and to innovate with lower technical and monetary barriers.

Software above the level of a single device: A service running on the Web has advantages for both, providers and users. Providers can release, manage and scale more easy, and users can access the service through multiple devices and applications.

Rich user experience: Web 2.0 services also represent technical progress. Technologies, such as AJAX (Asynchronous JavaScript and XML[1]), allow partial updates of the site's pages, which gives users a more dynamic and alive experience of the site. This makes static sites feel more like applications the user knows from his or her desktop, which explains the term *Web application*. The richer user experience of Web applications compared to quasi-static Websites of the past also represents a trend to move desktop applications to the Web. Examples are email clients and text processors. The latter enable collaborative editing of documents.

Creating a service following these ideas comes with its own specialities and peculiarities. All Web 2.0 applications flourish through the knowledge their users produce. This comes with its own downsides and upsides. Since the sheer mass of content produced is practically not manually reviewable to assure quality, service providers need to trust their users to a certain degree. An approach is to provide users with tools to flag or correct malicious content. As an upside example, having users produce massive knowledge and content makes it possible for providers to address the Long Tail [9] of, e.g., information and products, to differentiate themselves from their competitors. For instance, Amazon[2] recommends articles of the Long Tail to customers, based on users with a similar buying behavior (Collaborative Filtering).

An application gets better the more people use it. In Web 2.0, using a service is a form of participation, rather than consumption. Services become alive by changing and updating constantly through user contributions. However, user participation is

[1] Extensible Markup Language
[2] `http://www.amazon.com`

not equally distributed: Nielsen's rule of Participation Inequality [111] (Figure 2.2), also called the 90-9-1 rule, defines that on Web 2.0 sites only 1% of the users are heavy contributors and 9% contribute only intermittently. The remaining 90% of users are observers only. This rule challenges providers to increase user involvement cleverly to maximize data production by, e.g., Game Mechanics (Section 2.2.3).

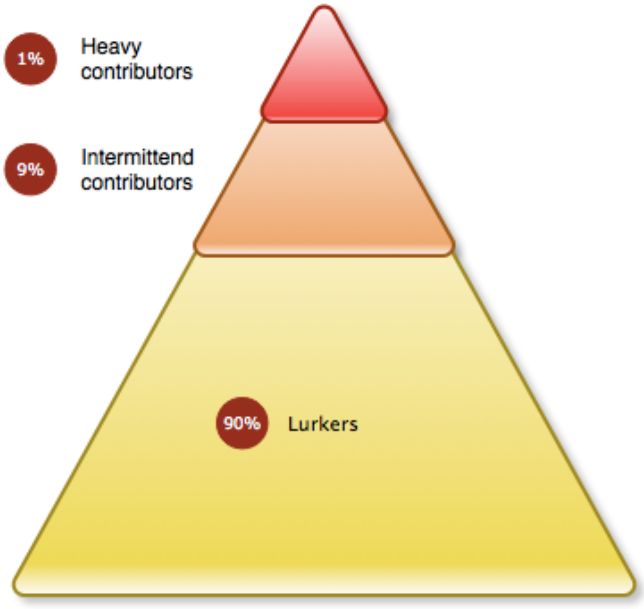

Figure 2.2.: Jakob Nielson's pyramid of community Participation Inequality depicts the unequally distributed contribution types of users of Web 2.0 sites. The site's content and contributions in general is created by 1% heavy contributors and 9% intermittent contributors. The remaining 90% of users are lurkers. There are ways and tools to affect these values, e.g. Game Mechanics.

The world-wide adoption of Web 2.0 led to the invention of a set of new Website types of which the most important are introduced in the following:

Content-sharing Sites are Web applications that provide a catalog of a particular content type that is created and shared by its users. This type of content is called User-generated Content (UGC). Content types range from videos, articles, photos, blogs to micro-blogs. Examples of these are Youtube[3], Wikipedia[4], Flickr[5], Livejournal[6] and Twitter[7], respectively. The providers'

[3] `http://www.youtube.com`
[4] `http://www.wikipedia.org`
[5] `http://www.flickr.com`
[6] `http://www.livejournal.com`
[7] `http://www.twitter.com`

motivation to create and run these sites is that there are no license fees to be paid for the contents as newspaper would have to. Also, the database created is hard to recreate by competitors, as stated above (Data-drivenness). The downside of this site type is the challenge of assuring quality of content and motivating the users to create content voluntarily and free of charge. A way to achieve the latter is, e.g., Game Mechanics (cf. Section 2.2.3).

Social Bookmarking Services empower users to annotate URL-based resources, e.g. information or objects, with free tags[8] [139]. The users thereby create lightweight conceptual structures called Folksonomies [64]. The term Folksonomy is a blend of the words folk and taxonomy and refers to a user-generated taxonomy. The created taxonomy typically is a tri-partite graph consisting of the entities users, tags and the resources tagged. Folksonomies are a means for making users generate meta-information for a resource and thus, help to create knowledge about a content catalog that is needed for operations such as search, discovery or trend detection [64, 65]. Examples of Social Bookmarking Services are Delicious[9] and Bibsonomy[10] [66]. Also, most content-sharing sites allow users to tag content and thus, create Folksonomies as a by-product.

Social Networking Services (SNS) are Websites that focus on its users throughout their profile pages and social connections to others as their main content type. Examples of SNSs are Facebook[11], MySpace[12], XING[13], and LinkedIn[14]. People use these sites to connect and to socialize with others, e.g. friends, acquaintances and family. The sociogram users create by befriending other users is a uni-partite graph of users and their connections and is often referred to as Social Graph. SNSs are introduced in-depth in the upcoming section 2.2.2.

Mashups are Websites that are, by a significant degree, mashed together from content and services provided by third parties. Through the provision of Application Programming Interfaces (APIs) by service providers of different Websites or data sources, fragments of their content and also functions become machine-accessible and thus re-usable by others. This lowers the burden for developers and companies of creating complex or feature-rich new Websites because functionality blocks or content-bases of third party providers are reused. The background of providing APIs is to enable end-users and amateurs to easily create new services and functions, similar to UGC. API providers have the advantage of gaining distribution of their functions and content, and often embeds links back to their Websites to increase traffic. Some providers charge license fees for using their APIs.

After this section has introduced different kinds of Social Media site types, the following section takes a closer look at SNSs.

[8] A tag is a keyword annotation of an item
[9] `http://www.delicious.com`
[10] `http://www.bibsonomy.org`
[11] `http://www.facebook.com`
[12] `http://www.myspace.com`
[13] `http://www.xing.com`
[14] `http://www.linkedin.com`

2.2.2. Social Networking Services

As defined in Section 2.2.1, SNSs are Websites focusing on user profile pages and their social connections. These sites serve the human people's natural need to socialize and for esteem.

In [31], Boyd defines the SNS core feature set to consist of

1. a personal profile. The personal profile is a page dedicated to a user and information about her. All users have their own profile pages.

2. an articulatable social connections list. This list of links to other users, mostly called a friends list, is a significant part of the profile page.

3. the function to traverse the social connections to others. By links leading to the social connections' profile pages, the site's sociogram becomes traversable. The viewer browses from profile page to profile page.

According to this definition, SixDegrees[15] was the first SNS to provide this feature set in 1998. Since then, countless SNS were launched for both, wider and narrower target groups.

From a business-strategic point of view, every commercial SNS is optimized to maximize its user base, the traffic created by its users or the content its users produce (UGC). The service provider's business models are based on at least one of these coherent goals. If it sells advertisements, such as MySpace, charges premium membership fees, such as XING, provides an application platform, such as Facebook or sells professional information and social connections to third parties, such as LinkedIn, most of these goals are of central interest to maximize business volumes.

In the first line, these goals are achieved by providing a feature set to boost the Network Effect[16]. The Network Effect describes the fact that a service becomes more valuable to all its participants the more participants there are. The classic example is the telephone. The Network Effect is also called the Fax-Effect[17] or the Law of Plentitude[18]. The Network Effect increases the user-base itself, the quantity of each user's visits and the traffic every user produces per visit.

SNSs globally moved from niche phenomenon to mass adoption in 2005 and 2006 [58, 31, 38] and nowadays occupy a significant amount of ranks within the world's top Web sites. Today, three out of the world's ten most popular sites are SNSs[19]. Facebook, for instance, is the world's biggest SNS counting 100 million members in 2008, 250 million users in mid 2009 and 500 million users as of mid 2010[20].

2.2.3. Game Mechanics

Many SNSs and other social applications apply Game Mechanics to further increase the contribution of information and the time spent on the platform by its users.

[15] `http://www.sixdegrees.com`
[16] `http://en.wikipedia.org/wiki/Network_effect`, accessed August 2009
[17] `http://kk.org/newrules/newrules-3.html`, accessed August 2009
[18] `http://en.wikipedia.org/wiki/Metcalfe's_law`, accessed August 2009
[19] `http://www.alexa.com/site/ds/top_sites?ts_mode=global`, accessed October 2008
[20] `http://www.facebook.com/press/info.php?statistics`, accessed November 2010

Game Mechanics are playful basics derived from games applied to applications and according to Kim defined as [77]:

Collecting things: Users can collect something, e.g. friends or items and present their collection to others.

Earning points: Many social services, such as Ebay[21] or Flickr, grant social points for certain activities on the platforms. They also communicate how to earn further points and how many points others have gained.

System feedback makes activities more fun and communicates a reward to the user for a performed action. It is also applied to communicate to the user the status of the game and what to do next, e.g. to gain more points.

Value exchanges are structured social interactions and a form of social engagement. Many SNSs, such as Facebook, provide virtual gifts for exchange for which it charges real money.

Customization and personalization let users feel more comfortable on the platform. They also increase the barrier to leave, based on the effort users have put into a platform.

Game Mechanics are one example to affect user contribution behavior in applications. Introducing others, however, would exceed the scope of this work.

2.3. Creating and Analyzing Social Platforms

This section lists concepts from the discipline of computer science for creating and analyzing SNSs.

2.3.1. Social Network Analysis

A social network is a structure consisting of individuals connected to each other by some form of social relationship, e.g., friendship, neighborhood or kinship. Social Network Analysis (SNA) examines social networks from the viewpoint of a network structure where individuals represent the nodes and their connections form the ties. SNA allows the examination of the graph structure in many ways, e.g., for measuring the graph's characteristics as a whole or for analyzing the properties of the individuals' connections.

Just as some measurements are numerical and computable, others are conceptual. An important sociological conceptual measure is Social Capital, which describes the value that individuals get from the network. Based on this and as mentioned in Section 2.2.2, the Network Effect describes the heightening of this value with an increasing number of participants (nodes).

Research in this area has been done from diverse viewpoints. In 1967, Milgram [105] ran an experiment to prove the small world phenomenon hypothesis. The hypothesis assumes that in a social network the path length to connect two arbitrary nodes is relatively short. In his *small world study*, he asked participants

[21] `http://www.ebay.com`

to pass a message along a chain of acquaintances in order to reach a target person, Milgram observed the number of six separation steps which coined the phrase *six degrees of separation*. Rogers with his diffusion of innovations theory [127] explained how ideas and innovations spread through a social network and how different personae, e.g., opinion leaders, affect this process. Hill and Dunbar [62] report that the typical size of an egocentric network, i.e. the view of a social network from the perspective of a particular individual, consists of 150 connections. They explain this number by physical limitations of the human's neocortex. The factual knowledge that humans have problems handling more than 150 people can be used to optimize social structures, e.g. villages or company (or department) sizes. Granovetter [56] examined the properties of strong and weak ties between individuals and points out the strength of weak ties regarding the seeking for information and innovation. He bases his theory on the fact that cliques, i.e. nodes connected by strong ties, tend to have homogenous opinions, which hinders the adoption of novel thoughts.

For decades, SNA has been a theoretical construct that was hard to measure, because the particular connections of the social network under observation were often impossible to gather. The advent of SNSs changed this problem, because the networks they run are computationally processable.

The measures of SNA provide insights of both, a holistic view of social network parameters and its properties from an ego-centric perspective. The following selection lists some common measures:

Betweenness is a measure to describe how a certain node is located between other nodes. It refers to the amount of nodes the node is connected to.

Bridge is a property an edge can feature: A graph can be grouped to clusters that are connected by singular or rare bridge edges, which are thus valuable for the graph as a whole. The bridge measure expresses that value.

Centrality describes how well a node connects the network as such.

Closeness describes how near a node is to all other nodes of the graph.

Clustering Coefficient describes the probability that two neighboring nodes of a node are connected themselves, creating a local cluster of nodes.

Degree counts the number of edges to other nodes.

Diameter is determined by the maximum Eccentricity across all nodes.

Eccentricity is the maximal length of all shortest path lengths between nodes.

Path Length describes the length of the shortest path between two nodes. The Average Path Length measures the mean path length of all nodes of the network.

Radius describes the minimum Eccentricity across all nodes.

These (and other) measures can be used to, e.g., compare a graphs individuals or different graphs.

2.3.2. Semantic Web

On the Internet in general, the relevance of finding the right information is becoming more important. To cope with the ever-increasing flood of information, users constantly need better searching and filtering tools. Tools and habits that work fine for us today, may not be sufficient anymore tomorrow. The act of searching must be seen from a more general point of view: It is not only users searching for something directly, but also machines searching for something to recommend to users, e.g. a meeting room or hotel. Search is the backbone for tools solving people's everyday tasks. To do so, a computer needs access to information and the semantic understanding of the things it analyzes.

Although the Internet as we use it was originally designed as a Web for exchanging data [17], over the years it emerged in Websites intermixing presentation and information designed and optimized for human consumption which effectively avoids machine-accessibility of information [104]. Machines' accessibility to information is the basis to semantic understanding and can be achieved by

- directly granting access to a database via an API. This involves the service provider to set up the interface.
- re-using the human presentation layer for machines by invisibly annotating Websites. This also involves the provider manipulating its content presentation templates to support the annotation formats.
- statistically trying to filter the information from the Website. This is the approach most search engine providers are using when indexing Websites.

The term Semantic Web was coined by Tim Berners-Lee, inventor of the Web [17] and board leader of the World Wide Web Consortium (W3C), in 1996 with the idea of making the human-focused Web machine-readable.

Machines have a different cognition from humans, because they lack common sense. Nowadays, for most computers reading a text document, seeing an image or listening to an audio track is like doing this for the first time, every time. To understand semantics, one has not only to sense what the item is about, one has also to classify, generalize, contextualize and to relate it to other entities.

The affected research disciplines dealing with this problem include:

Information Extraction denotes the collection, generation, conjunction, computation of knowledge about an item.

Entity Recognition describes the detection of entities within items and interrelations between a collection of items, e.g. an author within a free text, a set of tracks by the same artist, or a set of pictures where taken at a similar location.

Classification and Generalization to detect that subsets of items or entities have something in common or belonging to the same abstract class. Backed by an ontology, a computer can find out the relation between entities, e.g. superclasses and sinisters.

All these disciplines have trivial as well as highly complex and unsolved subtasks. The Semantic Web intends to overcome the computers' lack of a common sense and thus helps to solve the above-named problems by augmenting documents with meta-information through humans. An Information Extraction algorithm can benefit from that information by fusioning user-given meta-knowledge with computed knowledge. There are quite a few of top-down and bottom-up concepts, all with the same goal of creating structured, machine-readable information. The most notable extremes are ontologies and tagging, respectively. Ontologies are hierarchical graphs defining classes, their parents, children and sinisters. When a machine classifies an item to a class, this gives it knowledge about the hierarchical context. The most important techniques are the Resource Description Framework (RDF) [99] and Web Ontology Language (OWL) [15]. Tagging describes the annotation of items by keywords (tags), which must be seen from a twofold perspective: Editorial tagging by the author or service provider and user tagging through Website visitors. Authors and providers apply standards, such as Meta-Tags[22], ID3-Tags[23] and Microformats[24] [76], to their items to augment them with an standardized set of meta-information. Many modern Websites, such as Delicious[25], Flickr[26] or Last.fm[27], provide their visitors with a feature to freely annotate existing items themselves (Section 2.2.1). This helps the sites gathering knowledge about their items, i.e. bookmarks, photos and songs, successively, and to create a meaningful catalog and categorization out of the initially wild collection. Research has shown that, having access to vast amounts of these tag assignments, allows even higher aggregated knowledge, such as trends, over time to be computed [145, 146, 147].

Many standards and specifications created within the scope of the Semantic Web are also used as exchange formats in the Social Web realm (Section 4.7).

A future vision of the Semantic Web and its services given a semantic knowledge backbone for machines is drawn in Section 9.2.

2.3.3. Model View Controller

Model View Controller (MVC) is an architectural pattern in software engineering to structure applications into logical modules as layers with different responsibilities and abilities: A data model layer, a presentation layer and a layer for controlling the application's behavior [52]. The pattern is aimed at a more flexible application that is easier to maintain and to extend.

The problem with an application structure that mixes functionality of the different layers is that it becomes, firstly, hardly reusable for, e.g. a second presentation variant for the same model, because logical and presentation code has been intermixed. Secondly, intermixed code increases the effort that goes into the creation and maintenance of automated software tests for the software. Thirdly, the different layers of the MVC pattern also accord to different implementing skills of employees, allowing a project manager to assign differently skilled programmers to the split

[22] http://en.wikipedia.org/wiki/Meta-tag
[23] http://en.wikipedia.org/wiki/ID3-tag
[24] http://microformats.org
[25] http://www.delicious.com
[26] http://www.flickr.com
[27] http://www.last.fm

application parts.

Figure 2.3 depicts the different layers of the MVC pattern and their core functional responsibility.

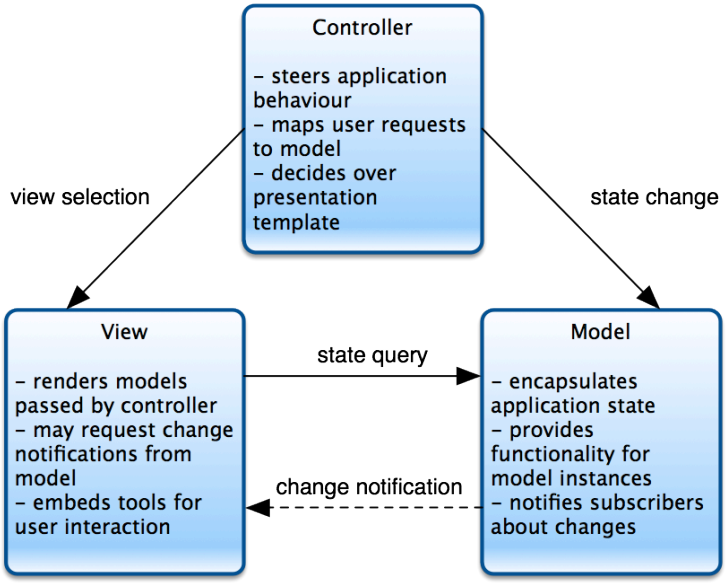

Figure 2.3.: The MVC pattern's layers, i.e. model, view and controller and their respective functional responsibilities and associations.

Model: The model layer represents and encapsulates the application state. It provides the functionality to create, read, update and to delete (CRUD) enterprise data. Enterprise data is the collection of entities contained in the application. Particular entities, i.e. a model, are on request instantiated and returned to the other layers. In some application types, the model layer also notifies subscribed functional modules of changes.

View: The view layer is responsible for rendering the data included in (data) models correctly according to presentation and actuality, and to give the user the possibility for interaction by, e.g., embedding push buttons or hyperlinks in the presentation. To ensure actuality of the data, the view layer can either subscribe itself as a listener to model updates to the model layer (push) or it can pull updates when needed by, e.g., a user request.

Controller: The controller layer defines the application's behavior. It receives user interactions, translates them into actions to be performed by the model layer and selects an appropriate template to be used by the view layer. When models need to be rendered, the controller layer requests them from the model layer, selects a template and calls the view layer to render the model's contents into a presentation.

The actual application of the MVC pattern to a given technology, e.g. HTTP-based Web applications, differs and is affected by the boundary conditions and the technical possibilities of the technology itself. The inability of HTTP to notify the client via *push*, for instance, forces a user-request driven behavior of the application which makes the view layer's responsibility for delivering up-to-date model data obsolete and thus, the model layer's change notifications unnecessary. In the following, we want to focus on the MVC pattern in the Web application domain. Furthermore, literature does not define the location of functional blocks in the layers of the MVC pattern in detail.

Figure 2.4 illustrates the MVC pattern applied to a typical Hypertext Transfer Protocol-based (HTTP) request-response-cycle in the Web application realm. A user views a page of a Web application in the browser and clicks on a hyperlink or button which was embedded by the view layer during the previous cycle. The browser creates a HTTP request and sends it to the Web application's Web server. The Web server analyzes the request, selects a controller of the controller layer and forwards the request to it. Typically, particular controllers handle requests regarding a certain entity type, e.g. a user controller handles all user-related requests. The controller evaluates the request and decides if an interaction with the model layer is requested and needed. If so, it calls the model layer to, e.g. create a new datum and validate it with the content passed, instantiate and pass a particular model to be displayed, or to update or delete a model. It then selects an appropriate presentation template to call the view layer to render the instantiated and passed model, to display an error message, or a static error page. The view layer embeds interaction tools (push buttons and links) for possible future request-response cycles. In the Web application domain, the server-side rendering ends with Hypertext Markup Language (HTML) code created by the view layer. The code is returned – by the Web server through a HTTP response – to the user's browser which renders the code in a visual representation.

2.3.4. Graphical User Interfaces

The importance of the Graphical User Interface (GUI) and its ergonomics is an often underestimated factor regarding the comprehension and acceptance of a software by its users. The interface between an application and its users determines if and how it can be operated both, effectively and efficiently. This section briefly introduces norms and disciplines relevant for this work. More detailed information about the wider topic can be found in, e.g., Nielson and Loranger [112] or Raskin [124].

Software Dialog Principles

The International Organization for Standardization (ISO) defines software dialog principles in its ISO 9241 part 110[28] norm. The principles are aimed at improving ergonomic principles applying to designing dialogs between humans and information systems, and consist of

Suitability for the task. This principle addresses how well the software suits the actual task, users are intended to perform. The software must ensure effec-

[28] `http://en.wikipedia.org/wiki/ISO_9241`, accessed October 2009

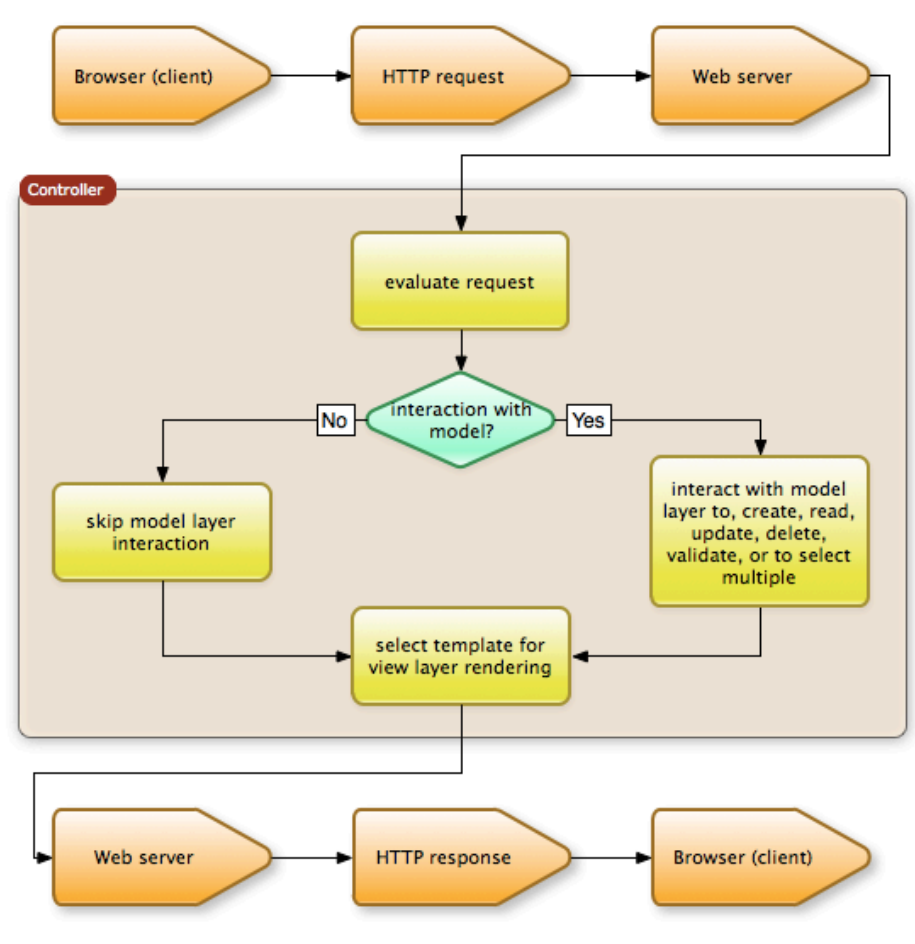

Figure 2.4.: The figure depicts the MVC pattern applied to the request-response-cycle-based Web application realm.

tiveness, but also efficiency, i.e. it should not provide more features or options than needed in order not to increase the learning effort.

Self-descriptiveness. A software should be self-descriptive and transparent. Users should not unnecessarily be forced or asked to read instructions or hint texts. They should be enabled to understand what is displayed at a glance.

Suitability for learning. The application should support the learnability of its usage for the user. This regards the learning process the user experiences when using the application for the first time until he or she becomes an advanced user, and includes the adapting of the software to the degree of the user's learning progress.

Suitability for individualization. Applications should adapt to or be adaptable (personalizable) by its users.

Conformity with user expectations. Software systems should go conform with user expectations. Users have expectations which they derive from associations they have from what they knew and learned before and what is similar to the tasks or use-cases of the software. A software should take this into account and work similarly.

Controllability. A software is controllable for users if they can control the speed of the application flow as well as adjust the amount of general input and output.

Error tolerance. Software systems should be error tolerant regarding user input and interaction.

It is noteworthy that the two major aspects of privacy, i.e. transparency and control (over personal data), are both dealt with by the principles of *self-descriptiveness* and *controllability*, respectively.

Information Architecture, Navigation– and Information Design

In order to bring information to the GUI and enable users to affect it through functionality, a number of design disciplines can be applied to improve the application's usability. A selection of important ones are

Information Architecture (IA) works on the defined entities and their relations, e.g. defined through an Entity-Relationship Diagram, and defines a basic concept of how users are operating a GUI to navigate through the information space. A good IA significantly affects the application's usability by improving the efficiency of accessing information, e.g. by reducing interaction steps, such as clicks, and the intuitiveness of the application itself, e.g. by making the relations of information plausible through underlining hierarchical relations between entities.

Navigation Design. Intuitive and easy navigation through the information space is critical to applications. Navigation Design is, on the one hand, strongly affected by the underlying IA, while, on the other hand, having its own peculiarities and tricks to further increase usability by, e.g., minimizing navigation

depth, and offering redundant paths and short cuts to information. Also, a good navigation is intuitively understandable by users.

Information Design is aimed at easing the understanding of the type of information displayed [53]. When displaying information of different entities, Information Design can dramatically improve the user's intuitiveness in understanding what he or she sees.

All of the named aspects significantly affect the abovementioned principles of *self-descriptiveness* and *conformity with user expectations* of ISO 9241-110.

2.4. Summary

In this chapter human people's basic need to socialize and for esteem was described and the term of privacy was defined, which is easily harmed in online communication. Furthermore, a new generation of communication tools of the online world was introduced. Using these tools, people are intuitively tempted to apply social norms and rules for communication from the offline to the online world. The problems concerning privacy threats are described and arranged in a novel topology in the following chapter.

Additionally, theoretical concepts regarding data analysis, software engineering and human-computer interfaces were introduced, which are used throughout this thesis.

3. Problem Areas of Privacy in Social Networks

As the previous chapters provided a selection of fundamental topics and introduced the realm of Social Media, this chapter is dedicated to describing privacy-related problem areas in this field revealing the underlying causal relationships.

With the advent of the era of Social Media and its applications, users got a voice and tools to interact with others. As described in Section 2.1.1, users naturally try to adopt social interaction norms and cultures from their real lives in the offline world to the online world. Because these norms are only rudimentarily applicable or featured in the online world, a set of privacy-related problems are caused.

This chapter provides a novel taxonomy to explain privacy-related problem areas of today's Social Networking Services. After an introduction of the identified problem areas is given, examples are listed for both, comparisons for offline and online world communication habits, and conflicting user and provider interests and the problem areas in question. Finally, a big picture is outlined to illustrate the causal dependencies and origins of the problem areas. The taxonomy has been published in the Proceedings of the CHI 2011 Workshop on Networked Privacy [84] and as a three-part article series at ReadWriteWeb[1] [83].

3.1. Privacy Balance and the Privacy Paradox

> Each individual must, within the larger context of his culture, his status, and his personal situation, make a continuous adjustment between his needs for solitude and companionship; for intimacy and general social intercourse; for anonymity and responsible participation in society; for reserve and disclosure.
> (Alan Westin [143])

Every human has a personal, subjective and contextual sense of Privacy Balance [35] that makes him decide whether or not to participate or to contribute to something that in return may affect his privacy. Westin allocates people to a taxonomy of three categories[2], as e.g. Kumaraguru and Cranor [92] report:

Privacy Unconcerned (about 18%), who are generally not concerned about threats to their privacy.

Privacy Pragmatists (about 57%), who know about potential privacy implications, but risk them for the potential benefit.

[1] `http://www.readwriteweb.com`
[2] The category sizes may vary slightly over time, age, and survey questions.

Privacy Fundamentalists (about 25%), who are so concerned about their privacy that they forego the benefit.

Bansal et al. [13] present a study, proving the importance of privacy policy statements and privacy assurance cues on subsequent decisions to disclose private information online in the contexts of finance, e-commerce and health. They found out that individuals with high and low privacy concerns use different strategies to form trust in a site. In the e-commerce area, the authors report that users with a high privacy concern rely more on the adequacy of the privacy policy and to a lesser degree on the company information and the Website design. Users with a low privacy concern answered the opposite. Whereas all users additionally include the Website's reputation in their decision, high privacy concerned users also take into account their prior experiences into trust formation. The authors also note that the understandability of privacy-policy statement has no effect on trust in the e-commerce context (but so does its adequacy for users with a high privacy concern). The authors recommend three aspects to be considered by service operators in order to alleviate users' privacy concerns:

1. a clear and adequate privacy policy statement
2. company information and reputation. The way a company is presented and viewed by the social community
3. the quality of the Website content and its design

Users with different levels of privacy concern use a combination of these aspects to form their decision about providing private information.

To research user concerns about their privacy in online systems, Cranor [40], Smith [132] and Westin [144] have undertaken surveys, which consistently revealed a high concern for privacy. Westin found out that 81% of internet users are concerned about threats to their privacy while online (i.e. the sum of Privacy Pragmatics and Privacy Fundamentalists). In a survey about the privacy concerns of users, who's listening behavior was logged for a music information retrieval application, Stober et al. [137] report that for many participants the logging came close to surveillance and that users must be fully informed and in control of what data is logged.

Privacy Balance has been researched by Varian [140], Cate [35], Preibusch [121], and Hann [60] from an economic point of view. The problem has been modeled as a trade-off function between disclaiming privacy by disclosing private information and gaining access to social services and their functions and social connections.

In contradiction to the above findings of rational attitudes of users concerning their privacies, Ackerman [2], Spiekermann [135], Gideon [54], and Poindexter [119] observed users in their interaction in online systems and report of an observed behavior of users acting *against* their stated privacy concerns by switching off privacy preserving controls or massively exposing private information. Also, 50% of the probands surveyed by Milne and Culnan in [106] never or rarely read Websites' privacy policies.

Vila [141] and Acquisti [3, 5] attempt to explain this contradiction which they call the Privacy Paradox. Acquisti shows that it is unrealistic to expect users' rationality in the context of decision making to privacy in electronic commerce. Faced with an immediate gratification, users who claimed to protect their privacy

are distorted in their decision-making process and tend to disclose more private information. They trade off long-term privacy for short-term benefits.

Privacy Balance is the users' rational and conscious trade-off function between privacy and benefits. The Privacy Paradox defines an irrational counterpart which is caused by users finding themselves in the middle of Maslow's hierarchy of needs [101] (Figure 2.1) when using social applications [89]. They find their privacy concerns subconsciously overridden by their needs for belonging, self-esteem and respect by others. The needs distort their decision-making and leads to privacy damages. Raynes-Goldie [125] additionally reports that the sheer mass of users, which a site has, leads to an increased importance to be member of the platform for the users' social lives.

Examples of the above named problem categories exploited by popular commercial service providers are:

Privacy Balance: Google provides a plethora of convenient services, e.g. Google Mail, Google Apps, Google Profiles, Orkut, Google Earth, and YouTube. They all have the common goals of convincing users to opt-in to access handy and free tools for the price of behavioral and personal data [148]. The threat to the users' long-term privacies is not qualifiable today, because it is not known what knowledge Google will be able to compute in the future on the basis of the behavioral data collected today. However, users can rationally decide about the tradeoff of whether to use the services or not because the satisfaction of social needs is unaffected (cf. Section 2.1.1).

Privacy Paradox: Facebook runs the world's biggest SNS with more than 350 million active users (as of January 2010[3]) contributing 35 million status updates a day and uploading 1 billion photos and 10 million videos every month. Also with Facebook, the threat to long-term privacy of the users is unknown. But in contrast to Google's or other service where social interaction plays no or a minor role, such as Ebay or Amazon, users find basic needs satisfied by interacting with other users and are thus distorted in their rational decision making process.

3.2. Privacy Theatre

While Privacy Balance is a rational way of humans deciding about the disclosure of private information, the Privacy Paradox is its irrational pendant, favoring the disclosure through a distorted decision making. Still, both phenomena are solely in the hands of the affected person. This section describes the interests of commercial service providers and how they bias user behavior to their advantages.

Although Pollach [120] states that privacy policies tend to intensify privacy concerns rather than engender trust, Bonneau and Preibusch [27], Antón et al. [12] as well as Milne and Culnan [106] revealed that privacy policies are not understood by users. In their field study of data protection on SNS, Bonneau and Preibusch [27] evaluated 45 SNSs using 260 criteria covering functionality, privacy controls, privacy policies, P3P (Platform for Privacy Preferences, cf. Section 4.4) policies, and

[3] `http://www.facebook.com/press/info.php?statistics`, accessed January 2010

metadata. They report a trend of SNS service providers to advertise and provide privacy features and privacy policies to escape criticism from privacy fundamentalists. At the same time they hide, or make unusable, privacy controls to their users and obfuscate their privacy policies with legal jargon. Unusable or non-existent privacy controls keep users from controlling their privacy by limiting the accessibility to information. E.g. Bonneau and Preibusch [27] report User Interface (UI) problems that prevent users from effectively using privacy controls. This maximizes information sharing among the majority of users which drives the growth of the community. All this is possible, because nowadays there is no means of enforcement that privacy policies and the information system implemented need to comply with [12].

The providers' strategical behavioral inconsistency of claiming to preserve user privacy, and maximizing information sharing behind the scenes, we call Privacy Theatre. Privacy Theatre is a provider-focused issue which acts commercially according to Data-drivenness (Section 2.2.1) and knows full well about the users' inability to understand privacy policies and their tendency to expose information according to Privacy Balance and Privacy Paradox. The providers' strategies are further refined by providing privacy controls, knowing that these are not used by 80-99% of users and almost universally set to an open default setting [27, 58].

3.3. Befriending Strangers

Basically, in SNSs users control their Social Context (Section 2.1.1) through friendships. If two users befriend each other, they thereby grant permission to access their profile page and other information to the other. Facebook, for instance, additionally includes a stream of news, called NewsFeed, which is populated with recent activities of friends. Friendships on SNSs often lack a means of qualification (the strength of the tie), causing the problem of Befriending Strangers. The problem is threefold:

- Firstly, SNSs mostly lack a friend definition that is sufficiently qualifiable. The created link between two users, which affects upcoming information flow, lacks context, intensity and direction. Intensity is also time-relevant in real life.
- Secondly, trust in online systems has a lesser perceived necessity, encouraging people to befriend strangers as a result of disembodiment and dissociation. The problem is missing feedback functions or a reminder that future information flow is received by these strangers (see also Section 3.4).
- Thirdly, users are encouraged to continue befriending others, because they are driven by the Game Mechanics of the platform (cf. Section 2.2.3) and the need for social inclusion.

Gross and Acquisti [58] researched the behavior of users on a Facebook network[4] of more than 4,000 Carnegie Mellon University students. They point out the gap between offline and online social networks. Offline social networks are extremely

[4] A Facebook network is a big SNS-internal group of users open to everyone to join.

diverse in how intimate a person perceives a connection to another person, i.e. how weak or strong a tie is. Online social networks, i.e. SNSs, do only offer a binary friendship link, resulting in huge friendship collections ranging from close friends to, sometimes, strangers.

Dwyer et al. analyzed in an online survey [43] about Facebook and MySpace how privacy concerns and trust influences social interaction within SNS. They noticed that in creating new relationships, trust is of lesser perceived necessity in online interaction than in face to face encounters. This proves Gross and Acquisti's point of befriended strangers accidentally being granted access to private information. Sophos showed in a study that 41% of 200 contacted Facebook users befriended a plastic frog figure, granting it access to private information [134].

When a user is about to publish information, the actual audience of it is often communicated and presented insufficiently to its users. The problem of this misconception is discussed in the following section.

3.4. Misunderstood Reach

As mentioned in Section 3.2, from the service provider's point of view, privacy and reduced accessibility of user profiles and information hinder the Network Effect. The level of detail of data an SNS can gain, process, share with or sell to business partners would be limited. Therefore, SNS providers mostly lack clearly communicated terms of accessibility of profile information and flow of information towards the user: They do not provide transparency. This leads to misunderstandings of the users regarding the reach of their information. The problem goes beyond the problem of Befriending Strangers, because it is not only about which users are granted access to a profile but also which third parties (cf. Section 3.8), e.g. business partners or Web crawlers, have access to *what* information and how the information is processed, forwarded, copied, augmented, conditioned or sold on the platform or by the provider.

Acquisti and Gross [4] compared a survey they produced in [58] on the abovementioned sample of more than 4,000 Carnegie Mellon University students and detected significant misconceptions among the members about the anticipated community's reach and their profile's visibility. They motivate that a network must better inform its users about the flow of revealed information from user to user, because often information flows to befriended strangers accidentally.

Krishnamurthy and Wills [90] provide a study of popular SNSs and their potential of information leakage. They motivate this work by users being willing to share information without clearly knowing the resulting accessibility of it.

Grimmelmann [57] underlines peoples' misunderstanding of the risks involved in socializing on SNSs with the example of Facebook. He claims that for the sake of socialization, people turn off privacy controls that are in the way and thereby unintentionally perform peer-to-peer privacy violations. In his search for a solution, which also considers regulation, he states that neither the market, nor better privacy policies or privacy controls, nor an access restriction, nor giving users the ownership over their information will solve this issue. The proposals he makes follow the goal of reducing the gap between what happens to their private information and the users' expectation of it, i.e. increasing the predictability of information accessibility.

Boyd et al. [30] discussed how user behavior and privacy can be regulated. They emphasized compliance with Fair Information Practices, promotion of self-awareness on how to visualize exposure, privacy management techniques and assessing risk and exposure of untrusted information.

The mentioned incomplete communication of profile visibility and information flow is further affected by service providers sometimes changing their rules of dealing with and conditioning information. From time to time, SNSs irritate and upset their users. Two examples are Facebook's NewsFeed and Beacon feature. NewsFeed aggregated user activities and published them widely within the service [29], and Beacon took NewsFeed facts and communicated them to business partners. Both features were significantly limited as a result of user petitions within days after their launch.

Boyd [29] reported in detail about the privacy concerns of users as Facebook launched the NewsFeed feature. Though the feature only made visible data that was already accessible with effort, it underlines the misunderstanding of users of the information flow between them. Interestingly, Boyd found out that every time, the users' subjective sense of control over as well as the reach of their data is proven wrong by the SNS changing its rules of spreading private information, the users change their communication habits accordingly over time to adapt to the changed social norms of the SNS.

In the Architecture of Privacy [94], Lessig writes that privacy partially depends on how information flows through any given space. This flow is determined by the space's architecture and technology.

What service providers need to keep up in order not to upset their users is what Nissenbaum calls Contextual Integrity [113]: Contextual Integrity is maintained when both, the norms of appropriateness and the norms of distribution are respected. The latter is often not communicated clearly to the users.

3.5. Absence of Control and Oblivion

Besides the problem of the misunderstood reach of information, an absence of appropriate features to control information flow as well as a lacking oblivion feature to erase outdated information or to limit its lifetime are discussed in this section.

Bellotti and Sellen [16] identified problems in the combination of inadequate *feedback* and the users' inability to *control* information flow and its audience. In mediated systems less information from the audience as feedback is conveyable through *disembodiment* and *dissociation*, e.g. position, gesture, posture, facial expression, speech, voice volume and intonation. They state that lacking feedback and control result in a disruption of social communication norms with respect to communication contents and behavior towards others.

The International Working Group on Data Protection in Telecommunications reports in its Rome Memorandum [69] that SNSs are an unprecedented case of user-consented publishing of private information. They point out the lack of rules governing user-initiated publishing of personal data. The technical presence and the first generation of humans growing up with an existing Internet—the Digital Natives—who are even more willing to publish private data than the Digital Immigrants (the generation of users who witnessed the Internet emerging), reinforce

the case. The group identifies an exhaustive list of risks for privacy and security. It includes the absence of oblivion of data, the service providers' misleading notion of their community and the lack of information to the users about which data is shared with whom and how it can be controlled, and finally the service provider's exposure of private content and behavioral data to third parties (cf. Section 3.8).

> We assumed the digital footprints we left behind–our clickstream exhaust, so to speak–were as ephemeral as a phone call, fleeting, passing, unrecorded. [... In fact,] our tracks through the digital sand are eternal. (Tom Zeller Jr. [150])

Solove [133] noted that public sharing of private lives led to a re-thinking of privacy concepts. Gossip that used to remain in a social circle's boundaries now may spread world-wide. He compared today's SNSs with a close-knit pre-industrial society where everybody knew everything about their neighbors. He pointed out that today's SNSs are a threat to privacy and that preserving privacy details is essential to reputation. He suggested that people must have an option to change themselves, meaning that they must have an option to make pieces of their past forgotten in order to change and grow. He found that nowadays users want to be in control of how private details are disseminated and that their privacy objection is about accessibility, not secrecy. He drew an analogy to copyright law, which provides a balance between freedom and control. He underlined that society must develop a nuanced understanding of privacy. If private information is available on a site, features are necessary to control what information is shared and distributed.

In their evaluation of 45 SNSs, Bonneau and Preibusch [27] found only three that define a data retention period in their privacy policy, namely Bebo[5], MeinVZ[6] and Plaxo[7].

3.6. Secondary Privacy Damage

Privacy damage is not only an issue between single users and providers. It is also a problem that users can damage the privacy of other users accidentially.

Krishnamurthy and Willis [91] point to users damaging the privacy of other users by involving them without knowledge with third parties by, e.g. emailing via a Webmail client that belongs to one of the named third parties, or disclosing their email address by inviting them to an SNS, or tagging them in a photo in an SNS. They claim that there is no way to prevent such secondary information leakage.

Squicciarini et al. [136] focus on the problem of users' control over co-authored content: Many assets published on SNSs do not solely belong to a single user, but are either co-authored or contributed by others, e.g. by commenting or tagging. The authors take a game-theoretic approach to automatically enforcing privacy policies and thus collaborative management of shared data through the involved users. They state that their approach ensures the sharing of content according to all stakeholder's interest.

[5] `http://www.bebo.com`
[6] `http://www.meinvz.de`
[7] `http://www.plaxo.com`

3.7. Security and Data Protection

The problem of security and data protection must be seen from three points of view:

1. Threats and risks in general through wrongdoers accessing private information.

2. Security flaws and implementation errors from the side of the SNS itself.

3. Social graph privacy threatened through computational attacks to private information via the friendship connections and group membership of the user.

The European Network and Information Security Agency (ENISA) published two technical reports [45, 14] remarking concerns about SNSs' observed vulnerability to Cross-site Scripting (XSS) attacks, viruses and worms through embeddable third party widgets, and phishing attacks [70].

Due to the fact that it is relatively easy for wrongdoers to gain access to a private circle of friends (Section 3.3), according to ENISA [45], Rosenblum [128], and Bilge et al. [19], the user's privacy is threatened by infiltration of his friends network. This may cause profile-squatting and reputation slander through ID-theft, stalking, bullying and corporate espionage.

These threats gain in severity, because many SNSs want their users to use their real names instead of pseudonyms, which makes it easy for wrongdoers to link from an account to an individual [93, 69]. Langheinrich [93] and The International Working Group on Data Protection in Telecommunications [69] appeal to service providers to introduce the option of a right to pseudonymous use (cf. Section 4.4). The service providers would increase the security of their users but alleviate the quality and value of their user-base which contradicts the aspect of Data-drivenness (Section 2.2.1).

Felt [47] demonstrates the vulnerability of the Facebook Platform, a platform for third party applications that are uploaded to and hosted by Facebook and then executable on the Facebook SNS. The work describes the applied security mechanisms and indicates a XSS vulnerability allowing arbitrary JavaScript code to be embedded on the application user's profile. Also, Jones and Soltren [71] list security holes in Facebook and give recommendations on how to fix them.

Bonneau et al. [26] researched the crawlability of Facebook and indicate that the applied security mechanism are not sufficient. SNS often are not closed eco-systems anymore. Many partner with third parties and allow their users to run a public profile accessible to the Web crawlers of search engines. However, the authors applied several starting points for their crawlings, including fetching the public listing, using fake profiles (sending out false friendship requests), profile compromising and phishing, malicious Facebook applications, and through the Facebook Query Language[8]. Through these techniques, they were able to bypass the privacy settings of users by appearing as a friend and got access to profile information and further links to profiles. Interestingly, in some cases they were granted access regardless of people's privacy settings.

[8]Facebook Query Language (FQL) is an SQL-like (Simple Query Language) query language to access data via the Facebook API.

Bonneau and Preibusch [27] reported that far too few of the surveyed 45 SNS adopted Transport Layer Security (TLS) and phishing prevention mechanisms during the login process.

Bonneau et al. [25] focused on Facebook users' publicly accessible profiles which all display eight friends. Facebook displayed only eight of every users friends publicly to protect their privacies[9]. They show how from this limited data accurate information can be computed about the social graph of users, e.g. the degree and centrality of nodes, find short paths between users, and detect community structure (cf. Section 2.3.1).

Zheleva and Getoor showed in [151] that it is computationally possible to predict profile attributes that are explicitly set private via available public information, such as public profiles of befriended users and group memberships. They hope to motivate SNS providers to enable a better control over the release of private information.

Korolova et al. [78] analyzed all SNS's basic building blocks, the traversability of the users' connection from an attackers point of view and recommend a lookahead depth limitation implementation by SNSs to preserve privacy.

Felt et al. [49] study the threats to privacy, especially impersonation and data exposure through Mashups. They focus on the disability of browsers to differentiate content sources in Mashups and platforms embedding and running third-party applications. Such proxied content makes browser-based security mechanisms base on the Same Origin Policy ineffective. They propose a new *untrusted* XHTML-tag to be embedded in third party code by mashup providers to enable browser-based filtering mechanisms to work.

3.8. Information Diffusion to Third Parties

Personal information that is secretly diffused to third parties is not only a problem of SNSs. E.g., Hoppe et al. report of privacy-threatening scenarios of personal data that is stored in today's cars [63]. However, the risks caused by today's SNS providers in this area have the following reasons:

- Firstly, the use of APIs enables third parties to re-use data and functionality, which on the one hand increases the network value according to the Network Effect. On the other hand, it also creates users, data, and also functionality provided by others. Thus, it is a win-win situation for the service provider which on the downside creates an uncontrollable replication of private information for users.

- Secondly, service providers tend to offer a reputation service to their users by allowing the partial opening of user profiles to the public. This includes search engine crawlers, which make the user profiles appear in search engine result pages of other people searching for that person. The problem of ranking the result high on the result page is solved by service providers performing a Search Engine Optimization (SEO) for the public profile pages.

[9] Facebook users have an average of 130 friends, `http://www.facebook.com/press/info.php?statistics`, accessed July 2010

- Last but not least, service providers expose private and behavioral data about their users to third parties to outsource advertisement targeting to advertisement partners, and to get free access to service monitoring and statistics tools, e.g. Google Analytics. The problem is that these third party companies track users in their meta-site usage behavior and create sophisticated user models that is beyond the control of the users.

These issues, combined with the providers' Privacy Theatre around secrecy of information, cause several problems and risks for the users' privacy and security.

The ENISA [45] states a concern regarding the use of real names. The disclosure of private data, such as location and personal information is a fundamental risk in these systems because they point to individuals and allow linking to other profiles of the same person in different SNSs via the name, image recognition techniques, such as face recognition or Content-based Image Retrieval (CBIR). It further points to the ability of third parties to collect personal data for digital dossier aggregation.

Felt and Evans [48] focus on the privacy risks of information exposure via APIs to third-parties. They studied the unnecessary exposure of private information to 150 popular Facebook applications and found out that nearly all applications could keep up their functionality with a limited and anonymized access to user data, for which they propose a privacy proxy solution.

Krishnamurthy and Willis [91] report privacy diffusion on a meta-SNS level through the gathering and aggregation of data by a decreasing number of third parties. They show up the deployed techniques and the depth of data collected. They point out the increasing threat to privacy through third-parties gaining knowledge about individuals' Website visiting and browsing behavior.

3.9. Property and Data Portability

Caused by Data-drivenness (Section 2.2.1), service providers aim to collect as much data about users and their content as possible. Analogous to the problem of Privacy Theatre, service providers often do not openly communicate that they own the data produced by their users. If the possession is legally not possible, the providers usually create a license allowing them to use and sell the data produced on their platforms. Furthermore, service providers put an effort into hindering users from changing to a similar service by, e.g., not providing export functions for the data produced to port the data. These burdens are referred to as data-silos or lock-in effects. Many Web 2.0 applications systems were built on this unsteady basis of conflicting property interests and lock-in effects.

In February 2009, Facebook introduced an unannounced change in the site's terms and conditions to strengthen its intellectual property of information produced through the service. As with the release of the Beacon and NewsFeed features (Section 3.4), the reaction was bad press and user petitions [28]. In exactly the same way that providers claim privacy to be preserved through Privacy Theatre, Facebook claimed openness and equality in two documents released for public review and commenting in order to replace the existing Terms of Service: the Face-

book Principles[10] and a Statement of Rights and Responsibilities[11]. Bonneau et al. [28] found numerous issues within the documents, including failing the goal "to simplify the language so you have a clear understanding of how Facebook will be run" (cf. Section 3.4), vague terms, legal jargon, contradictions and an asymmetry of power concerning user obligations versus provider responsibilities. They find Facebook making "aggressive claims on the intellectual property rights of content associated with the platform without providing effective limits on how they may be used."

There are two sorts of activities approaching this problem: on the one hand, standards are invented for protocols and exchange formats to authenticate, authorize, port, synchronize, and syndicate information (Section 4.7). On the other hand, researchers invent distributed architectures for SNSs assuring data portability and property to users (Section 4.2), often in combination with global identity management systems.

3.10. Examples

As stated before, many of the problems are caused by users naturally transporting offline, real-life learned social norms and rules of communication to the online world's social applications. These applications are not capable of providing functionality that works equivalently or as expected. Reasons for this vary from saving implementation costs, missing expertise within the company or strategies to maximize the named provider goals, i.e. user base and profit.

Table 3.1 compares the differences of exemplary one-to-many communication between friends in the offline and online world referring to some of the problem areas identified. The offline world situation of a person communicating with their friends is compared to the online world counterpart of that person posting content to their profile page which is only visible to their befriended users.

Table 3.2 compares the conflicting interests between users and service providers. Exemplary, it names some extreme conflicts of interests to underline the identified problem areas. The named user needs and interests were taken from numerous user surveys, e.g. [89, 88].

3.11. Summary and Conclusion

This chapter developed a novel problem taxonomy regarding privacy threats in the area of SNSs. To round this off, the problem areas found and described above are briefly summarized.

Privacy Balance and The Privacy Paradox are two user-focused decision-making phenomena. Privacy Balance is a rational adjustment each individual performs between the protection or disclosure of privacy for a given benefit. Pri-

[10] `http://www.facebook.com/topic.php?uid=54964476066andtopic=7960`, accessed September 2009

[11] `http://www.facebook.com/topic.php?uid=67758697570andtopic=7569`, accessed September 2009

Offline world	Online world	Referred problem
Primary audience visible in direct communication.	Primary audience invisible; may contain forgotten befriended strangers.	Misunderstood Reach, Befriending Strangers
Immediate interaction between people provides feedback and control over communicated content and audience. Social norms for communication and behavior are learned, controllable and thus preserved.	Mediated interaction via technology lacks to provide both, proper feedback and control over communicated content and audience. The reach of content is often subjective, not known or not clearly communicated by the provider. Social norms are easily disrupted.	Misunderstood Reach, Absence of Control and Oblivion
Indirect reach through delayed retelling, recording or notes present but predictable.	Indirect reach unpredictable and uncontrollable: may flow to third parties where it is replicated and not erasable.	Security and Data Protection, Information Diffusion to Third Parties
Communicated information is ephemeral; it may not have been written down and may be forgotten.	Communicated information is persistent, and usually never deleted.	Absence of Control and Oblivion
Recorded communications are hard to access, search, analyze or to condition in batch.	Persisted communication can easily be accessed, search, analyzed or conditioned in batch.	Security and Data Protection
Qualified and differentiated friendships.	Unqualified and equivalent friendships.	Befriending Strangers
Hard to infiltrate a circle of friends. More trust necessary through immediate communication.	Easy to infiltrate a circle of friends. Less trust necessary through more indirect communication.	Befriending Strangers

Table 3.1.: Aspects of privacy in a communication compared in the offline and online world: People transport learned social norms and rules of communication from the offline to the online world, causing a significant part of today's problems reported with social software.

User interest	Service provider interest	Referred problem
The users want to expose as little of their privacy as possible when interacting online.	The providers want to maximize the exposure of information to increase the Network Effect and their content database.	Privacy Balance and The Privacy Paradox, Security and Data Protection
The users need to feel safe when communicating privately with friends.	The providers emphasize safety and the protection of privacy in their privacy policies and marketing statements. Facts indicating the factual audience and invoked reach of the information flow are concealed.	Privacy Theatre, Absence of Control and Oblivion, Security and Data Protection
The users want to control to whom they are communicating a piece of information, see the audience, and control the information flow.	Any limitation of the audience chokes the Network Effect and the size of the information flow and thus is suppressed by the provider by hiding the options.	Absence of Control and Oblivion, Befriending Strangers, Information Diffusion to Third Parties
The users want to have an option to delete outdated information or a limited information retention.	Any deletion of parts of the content database decreases the knowledge the service provider owns in its content database, and thus its valuation.	Absence of Control and Oblivion, Information Diffusion to Third Parties
The users want to keep property of their data.	The providers needs the property of data to maximize trading possibilities and duration.	Property and Data Portability
The users want to export or synchronize their data with different services.	The providers do not want to give away data produced through their service to a competing service.	Property and Data Portability

Table 3.2.: Conflicting interests between users and service providers of social software causing problems.

vacy Paradox stands for an irrational and distorted decision making process of an individual trading off long-term privacy for short-term benefits.

Privacy Theatre describes the behavioral inconsistency of providers claiming the protection of their users' privacy in the press and privacy policies, and maximizing information sharing behind the scenes.

Befriending Strangers describes the lack of qualifiable friendship connections on platforms. Together with people's tendency to befriend strangers through an mistaken feeling of trust, and the featured Game Mechanics making this fun, the described three aspects cause a shrinking of control over information flow and increase the invisible audience.

Misunderstood Reach is caused by the above three problems plus the service provider not clearly communicating the users' profiles' visibility and the reach of information flow. In a subjective feeling of control and through misconception about information flow, users communicate and chat on these platforms sharing private facts, gossip, pictures and rumors.

Absence of Control and Oblivion is a follow-up problem of Misunderstood Reach. The lack of communicated feedback about the reach of information for the sake of the Network Effect caused a vacuum of controls for users to steer their information flow and visibility. This implies a definable life-time and clearance of information. Not surprisingly, oblivion of information is very seldomly featured at this point in time.

Secondary Privacy Damage is a problem that arises within social applications and the problem of Misunderstood Reach. It describes individuals inadvertently threatening other individuals' privacies without their knowledge or opting into a service by disclosing or linking to their identity.

Security and Data Protection are problems caused by software as such and including missing barriers hindering wrongdoers of approaching users, software flaws and errors, and computational predictability of protected information via the users' social graphs or other algorithmic threats to users.

Information Diffusion to Third Parties is an issue threatening user privacy, which further exacerbates the above named problems by selling and replicating private information to further parties, making information and especially its clearance effectively uncontrollable. Usually these practices are not openly communicated to a platform's users.

Property and Data Portability are based on Web 2.0's aspect of Data-drivenness as part of service providers' strategy. Accordingly, providers tend to maximize the property of data produced by their users and to minimize the technical possibilities to export or synchronize data with other SNSs.

The origin for the beforementioned problem areas is a conflict of interests between providers and users. As providers aim to maximize profits, user base and business intelligence, the intention of the users is to communicate and to socialize. Figure 3.1 illustrates an overview of the causal dependencies of party goals and problem

areas. It is obvious that the providers' party causes or is at least involved in all but one of the described problems. However, it becomes obvious that most problem areas could be approached by offering a greater transparency and control over the information flow of their personal information and content. As said before, this drawback is not unintentional on the part of the service providers.

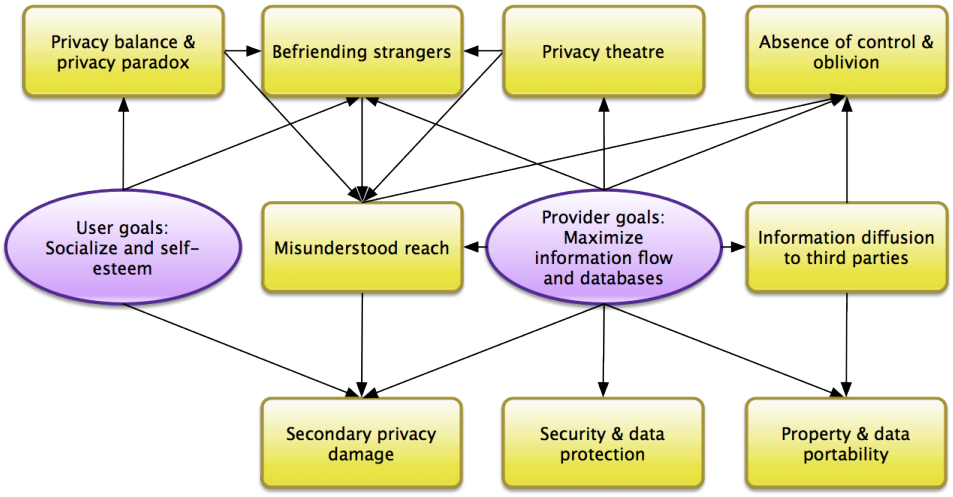

Figure 3.1.: Overview of the causal dependencies between party goals (ovals) and problem areas (boxes).

Having said all that, there are of course different ways to approach a problem of this nature. Krasnova et al. [88] built their taxonomy by clustering potential sources for privacy threats from a survey interrogating users. Hence, their taxonomy reflects the involved problems only from the user's perspective. They name *organizational threats* and *social threats* to constitute the two underlying dimensions for all privacy concerns. Preibusch and Bonneau [27] approached the related problem of data protection in SNSs from a technical and functional perspective, but also enlightened privacy policies. The authors created no taxonomy, but evaluated 45 SNSs using 260 criteria to detect problems concerning user data protection in SNSs. However, we chose to base our problem areas on the involved parties to emphasize their contradicting interests and tried to highlight the relevant problems from sociological, technical and economical points of views. Doing so, we hope that our more interdisciplinary approach more holistically guides future developments in this area.

Concerning how to proceed in order to provide answers to the named problem realm, our taxonomy made obvious that a software solution alone will not be able to solve all of the above problems. Therefore, the following chapter will introduce interdisciplinary approaches and partial solutions for one or more of the problem areas discussed.

4. Addressing the Problem Areas of Privacy: Related Work

The previous chapter has introduced a novel taxonomy for privacy-related problem areas in the field of Social Media. This chapter deals with theoretical and practical works to approach these areas.

Various theoretical and practical work from different scientific disciplines has been published to address the previous chapter's problem areas. The types of approaches range from holistic to very specific as well as from theoretical to practical works. In the following, these approaches are grouped to approach fields, described and put into relation to problem areas of the previous chapter. This chapter's summary section provides short summaries of the presented approach fields.

4.1. Access Control

The following works deal with directly or indirectly empowering the user to control the reach of communicated information. By openly communicating a receivers list to the user, besides the problem of Absence of Control and Oblivion, the basic problems of Misunderstood Reach and Befriending Strangers, which are the product of missing transparency, are also addressed.

Mannan and van Oorschot [98] intend to restrict personal content to a selected group of contacts and assert that this is not possible on personal Websites. In their approach, they re-use an existing circle of trust, i.e. the contact list of an instant messaging client, to limit access to a Website. Their approach gains importance when content access must be restricted on a meta-SNS level (Section 4.7), e.g. in the emerging Social Web (Section 9.1) which is explained in the outlook chapter (Chapter 9). On SNS level, limiting access to information to a pre-defined, static friends list is usually available as a feature, but would be worth revisiting to answer the Befriending Strangers problem.

Jones and O'Neill second the idea of people sharing their information with selected groups of recipients for privacy protection [72]. In a survey, they detected six commonly considered criteria people used in a card sorting game for creating groups of their contacts, e.g. social circles, tie strength, geographical locations, or organizational boundaries. Since they assumed that the configuration of groups would be a considerable burden, they researched an automated group creation algorithm and present potential for reduce this burden.

Indratmo and Vassileva [67] propose an access control framework for blogs, which are, such as Websites, usually accessible by everyone. Unlike other approaches, the framework grants access to pre-definable groups of people instead of content items, such as articles. The authors perform a study concerning the acceptance and usability of the tool and report a positive feedback. They emphasize that

when designing an access control mechanism, its effectiveness must be ensured by studying users. Otherwise, users could mistakenly believe that they have protected their data and privacy. Having introduced an access control system for individuals, the authors proceed with the logical steps to grant write permissions to groups, making the mechanism a collaborative blogging tool or the foundation for even more sophisticated workflows.

Choudhury et al. [36, 37] take a different approach to enabling users to (indirectly) control their audience and thus, the reach of their published information. The authors define the modeling and the selection of relevant features needed to automatically predict the intended audience, i.e. the Social Context (Section 2.1.1), and delay for a piece of published information. They report a mean error of about 10-15% for predicting intent and delay, and found out that there are further latent factors involved that exceed the context of the communication, e.g. mood, sentiment, and location. It must be noted that the data set[1] of this work consists of messages sent on the MySpace SNS, which is a one-to-one communication only. It is not clear whether in one-to-many communication, which undoubtedly changes the Social Context, the modeled features are still delivering correct values. Furthermore, it is unclear whether a different SNS with a smaller data set can still compute correct prediction due to a smaller training set.

None of the presented approaches provides a solution to sufficiently control heterogenous Social Contexts. Users need control over different receiver groups for different topics they publish. Also, automated solutions for both group creation and receiver selection for content may enable a degree of control, but have to provide transparency to users. If such an algorithm fails, privacy may be harmed more than protected through a violation of Contextual Integrity. However, aspects and ideas of these approaches, e.g. friendship qualification and group features, are under consideration for the approach to be designed.

4.2. Architectures and Frameworks

Numerous works approached the defined problems by presenting both theoretical and practical, holistic approaches of privacy-preserving architectures and frameworks. The works can be divided into centralistic and distributed approaches.

Centralistic approaches design a system architecture that is run by a single provider. Mostly, these works focus on theoretical system basics, e.g. policies, that aim at improving transparency and control for users.

Distributed approaches focus on applying a peer-to-peer architecture to the problem. By distributing content, several problems are covered simultaneously: Firstly, a centralistic provider, who has access to all data and that needs to be trusted not to harm user privacies (Panoptic Provider) is made obsolete. Secondly, the question of data ownership and possession can be answered by, e.g., empowering people to host their own data or at least choose from a number of Identity Providers (IDP) to do so.

[1] The data set consists of approx. 20,000 MySpace users, who have exchanged about 1,4 million private messages between September 2005 and April 2007.

4.2.1. Centralistic Approaches

Bellotti and Sellen [16] identified a lack of Contextual Integrity in mediated interaction between people via technology. They underline that technology directly affects the control people have over their personal data, i.e. their privacy, and that privacy should be a central issue in designing interfaces and the technology of systems. The authors present a theoretical framework for providing feedback and control to users about the following types of behaviors:

Capture. When and what information about users gets into the system and how they can control it. The authors emphasize this point because their framework is for arbitrary ubiquitous computing environments including multimedia conferencing tools or systems including sensor data.

Construction. What happens to information about users once it is in the system? Can users affect system behavior and permissions?

Accessibility. Which people and what software has access to information about or of a user? How can access or usage be controlled?

Purpose. For what purpose do people want information about a user?

To evaluate these behaviors, the authors provide a set of criteria to optimize system design. These criteria give advice on how and when feedback should be given, and tools for control should be provided to the users. The framework is applied as a question catalogue, evaluating existing systems according to the named behaviors to detect problems, and to point out possible solutions. The feedback collected refers to the problem of Misunderstood Reach as well as Absence of Control. Once both aspects are present, users are empowered to make informed choices.

EU-funded project PrimeLife[2] (Privacy and Identity Management in Europe for Life) is aimed at bringing life-long privacy and user-control over personal information and autonomy to the information society. PrimeLife focuses on putting users in control of the personal information stored and transmitted about him. The project is split into different activities:

Privacy Life focusses on the provision of privacy-enabled identity management for the entire life of people. The activity's focuses are user-centric control over personal data in terms of sharing, selective access-control to user information and data, and managing identity, trust and privacy throughout life.

Mechanisms for guaranteeing privacy and trust in the electronic society.

User Interfaces for representing privacy-enhancing identity management tools, and for trust, assurance and policy display and administration.

Policies that focus on security and privacy policy systems which include legal requirements, new policy mechanisms, machine-readable policy languages, and policy decision engines.

[2] `http://www.primelife.eu`, accessed December 2008

The project has not yet produced any practical solutions, although it has published valuable work on reporting the current state of the art in the different areas. These include Web protocols, concepts and requirements for privacy-enhancing access control in SNSs and collaborative workspaces (CW), topic-related cryptographic mechanisms, and intuitive and usable user interfaces for the above abovementioned tasks. The report on concepts and requirements for privacy-enhancing access control in SNSs and CWs gives an overview over use-cases continued by observed issues. It then lists mechanism groups with potential solutions to approach the issues, and finishes by listing collected requirements.

Antón et al. [12] present a theoretical framework for online privacy policy management. They separate the challenge into a user side and an enterprise side. Both sides are affected by society, law, economics as well as human factor based design and usability. Their approach is a three-tier architecture, consisting of a top tier to provide privacy policies in both natural and formal language (whereas the latter is for machine interpretability). A middle tier deals with access control and audit policies, and privacy protecting information flow control. A bottom tier features enforcement in the physical layer, i.e. the application of fine-grained access control and auditing policies to the information repository.

It is important to note that privacy policies as such are very different to access control mechanisms, because they are both, static and more a form of promise made to users. They are, compared to access control rules, that may be changed daily by users, more abstract and high-level than concrete rules. Today, service operators provide neither a formal version of their privacy policy nor do they ensure that their features' functionality conforms with the stated privacy policy. The authors also find the requirement for information flow control policies to ensure the compliance of differing privacy policies of different providers. Furthermore, a formal semantic privacy policy language could help to ensure compliance with social norms and laws.

The most challenging research issue the authors identify is the binding and translation between the defined layers: Between top and middle tier to provide authentication, access control and information flow control that enforce the privacy policies and security requirements; Between middle and bottom tier to provide fine-grained access control mechanisms on a database cell level.

The user side of the framework defines a user agent that provides policy processing and presentation to the user, as well as the generation of privacy preferences from the user's privacy protection objectives, and the automatic matching of the providers' privacy policies and the users' preferences. The authors emphasize the usability of privacy management tools because a high user engagement is necessary for preference specification. The authors list research issues on how end-users can be empowered to control their privacy objectives and also to protect them from making errors, because technical aspects are often too complicated to comprehend.

The authors limit the preferences users have to control their privacy to the top level, which is not appropriate for SNS. Without a doubt, users want to understand high-level privacy policies, but the main problem is to control one's privacy in different contexts and in every situation.

Anderson et al. motivate their work [10] with the problem of users being forced to trust service providers, who are in most cases of today's SNSs an omniscient and omni-possessing entity of all the network's data (Panoptic Provider). They

doubt the providers' ability to protect user data from malicious software that made private information public, i.e. the problem of Security and Data Protection, and intend to delegate access control to content to user clients. The authors present an approach of an untrusted client-server architecture enabling users to protect their personal content and links (friendships, group memberships, social browsing history) from unintended sharing with others and from the service operator itself (touching the problem of Information Diffusion to Third Parties and Secondary Privacy Damage). They place the responsibility for confidentiality and integrity with the clients which follow a given specification. The core functionality lies in a cryptography layer providing features for confidentiality and integrity to safeguard access control to content (the problem of Absence of Control and Oblivion), and the prove of genuineness, respectively. The content itself is encrypted and hosted by the untrusted server. Only clients possessing a key to decrypt the content and its links can decrypt the information.

Similar approaches to encrypting user content to protect it from the service provider and unauthorized access, have been created by Lucas and Borisov [96] called FlyByNight, which encrypts user communication on Facebook, and by Guha et al. called NOYB (None of your Business), which uses a browser plugin to allows its users to encrypt their Facebook profile and decrypt those of their friends by a password [59].

4.2.2. Distributed Approaches

Buchegger and Datta go one step further by envisioning a solution that takes not only the omniscience, i.e. the general accessibility of user information to service providers (Panoptic Provider), but also the possession of data away from service providers by proposing a peer-to-peer architecture [33]. They underline the approach's motivation by the users of such a network to be granted control of who accesses their information and content (the problem of Absence of Control and Oblivion), the intellectual property (the problem of Property and Data Portability), possibilities of licensing user content, and the possibility of unmediated exchange of data. The authors list the challenges of peer-to-peer social networking that arise compared to client-server architectures. In terms of porting SNS features to a peer-to-peer architecture, they discuss the question of storage, the subscription to system updates, the reliability of the network itself and its topology, the availability of user content when nodes are offline, and the searchability of a distributed architecture. Concerning the problem of empowering users with control, they list issues of authorizing friends by distributing and maintaining keys to encrypted content, non-repudiation, impersonation prevention and robustness against misbehavior.

Speck and his team worked on a practical decentralized SNS approach called HelloWorld[3]. The research project is aimed at developing an open source protocol that allows its users to build a decentralized and secure SNS. HelloWorld intends to ensure security through the encryption of data, and control over their data through allowing its users to decide where their data is stored. The approach focuses on solving the problems of Property and Data Portability and Information Diffusion to third parties by avoiding a central and Panoptic Provider by moving the data

[3] `http://www.helloworld-network.org`, accessed October 2009

to a host of the user's choice. The encryption of data weakens the problem of Security and Data Protection. The project intends to support user personas, i.e. a user can create and appear to others as different personas, providing different sets of profile information with different levels of detail. Thus, the user can control their appearance to others in more detail according to a particular Communication Context and Social Context, e.g. to business partners or to family (cf. Section 2.1.1). The project has been discontinued and was never released.

Canter [34] outlined his vision of an architecture to host user data, called the Open Mesh. He segmented typical assets of a user profile into pieces, e.g. address book, files, texts, comments. He envisioned how this content can be hosted, distributed and made ubiquitously accessible. His approach is a distributed system of providers hosting personal content. It focuses on the problem of Property and Data Portability.

For the approach discussed in the next chapter, distributed architectures and encryption are beyond the scope of this thesis. However, a look-ahead to future architectures and trends in this area is given in Chapter 9.

4.3. User Interface and User Experience

The following works focus on improving User Interfaces (UI) and User Experience (UX) for empowering users to make informed decisions. Mostly, these approaches attempt to solve or improve the situation arising in regards to the problem areas of Privacy Paradox, Privacy Theatre, Misunderstood Reach and Control.

It is important to understand that good software usability and ergonomics are in themselves aspects to approach some of the abovementioned problem areas (Chapter 3). E.g., ISO 9241 part 110 (cf. Section 2.3.4), a norm to improve ergonomics of software dialogs, lists a set of principles of which two refer to the basics of privacy: Transparency and control. Transparency is required by the norm's principle of self-descriptiveness, and control by the principle of controllability. Numerous works deal with assisting or helping users in understanding or operating software and thus, its ergonomics. Nielson and Loranger [112] provide an exhaustive catalogue of how to improve the usability of today's Web applications. Many of their tips are quasi-standards today and were applied to the application introduced in Section 6.7. Norman [114] describes how to improve software ergonomics for users by, e.g., providing simplicity, intuitiveness and transparency by minimizing the gap between system state and the imagination users have from it. Similar principles have been defined by Shneiderman with his Eight Golden Rules of Interface Design [130] defined principles as a guide to good interaction design. While similar to the principles of ISO 9241 part 110, he adds easy un-doing of actions (which Jef Raskin also advocates [124]), consistency regarding the wording and actions in similar situations, the use of informative feedback, and the reduction of short-term memory load of users. However, the majority of advices and principles formulated in these works are based on or are coherent with those formulated in ISO 9241 part 110.

Despite the general problem of Privacy Theatre, which keeps service operators from providing tools for privacy policy creation, high quality privacy rules editing is a means for automated privacy access decision processes to effectively and

discretely preserve people's privacies. Karat et al. [73] showed that novices in privacy policy authoring were able to create high quality privacy rules. The probands were presented with scenarios and succeeded in rule authoring in two ways, with a structured list tool and with natural language with a rule guide tool.

The Platform for Privacy Preferences (P3P) project [39] is a specification recommended by the World Wide Web Consortium (W3C) for Websites to communicate their privacy practices in a standard format to user agents. The P3P user agent allows the notification of users, and an automated decision-making processes. Websites can translate their privacy policies to this format, relieving the user of the need for reading and understanding it. The user, on the other hand, configures their privacy settings at the user agent which in return decides on warning the user when accessing a site with different practices.

However, it is unlikely that the problem of Privacy Theatre could be solved if SNSs supported P3P regardless of the counter-productivity for the service providers' goals, because P3P comes with the issue of failing to provide clear semantics. It can thus be misinterpreted by different user agents [12]. Furthermore, P3P is reported to represent policies inconsistently and fails to represent important statement, e.g. lifetime of data, security mechanism applied to protect data, and what data is *not* collected.

Motivated by the problem of users not understanding and not reading Websites' privacy policies, Engelman et al. [44] studied the timing and placement of icons as privacy indicators on an e-commerce Website and its effect on the probands' behavior. They found a serious impact of displayed privacy information on the users' decision on how much to pay for the goods. Furthermore, they were able to prove that people pay more attention to privacy information when they were about to purchase privacy-sensitive items.

Lipford et al. [95] investigated mechanisms for socially appropriate privacy management in SNSs. They propose an audience-oriented view of one's profile information to improve the users' understanding of profile settings and to more accurately reflect the users' mental model of privacy settings. The audiences they introduce are search engine, network, friend, and self. The participants were asked to perform several tasks concerning the adjustment of privacy settings on both Facebook and a prototypic improvement of it featuring audience-view tabs. The authors report that both accuracy and comfort level increased significantly when using the prototype almost constantly. Although this approach only provides a simple and one-dimensional differentiation of audiences and no proper reflection of one's Social Context, it is still a helpful feature that is, by the time of writing, only proposed by a couple of SNSs, i.e. Facebook[4] and LinkedIn [27]. Both networks provide an individual view of one's profile through the eyes of a friend of choice. Though the feature is important to check one's profile page's accessibility regarding a particular user, the user still misses a transparency feature that provides an overview of who has access to what.

Many works report that empowering users to manage complex access control rules is a hard task because with expressibility comes complexity [131]. For service providers, the provision of intuitively usable tools is very costly to develop and often counter-productive to the providers' goals, because the flow of information

[4]Facebook introduced this feature after the named paper was written.

and thus, the Network Effect is limited by users. According to the problem of Privacy Theatre, these features are therefore poorly developed and featured on SNSs (cf. Section 3.2). However, Facebook provides a feature called Friend List, which Onwuasoanya et al. researched [115]. This feature allows users to group their friends and apply simple access control preferences to them. The authors in a study forced participants to create groups of friends and to apply rules to them by opting-in. They found almost one third of the participants creating different groups and restricting particular profile information from them. The problem with these lists is that they only allow users to restrict a particular profile area or feature, e.g. a status update, to be statically limited to such a list. It is not possible to publish something today to one list and tomorrow to another without changing general preferences. This insight underlines that one-dimensional access control mechanisms, i.e. limiting access to a particular profile information to friends, friends-of-friends, network, and search engines, do not sufficiently reflect the needs of users to control access to their personal information.

The authors report that only 25% of the participants knew and used Facebook's feature and only 5% found it easy to use. This may be a result of the platform's missing advertisement of the feature. Unsurprisingly, the groups created by the teenage volunteers were all related to a Social Context and sometimes even congruent, e.g. activities, school and friends. The authors show that the degree of closeness of a person does not automatically imply a willingness to allow access to private information, e.g., as a home number is not shown to school friends, relationship status, photos and videos are often hidden from relatives.

It is obvious that UI and UX play a major role in order to create and effective and efficient tool for privacy protection. If a software is not understandable or usable, privacy can neither be controlled nor transparency can be provided in order to avoid violations. Therefore, the approach taken (cf. Chapter 6) emphasizes the design of the user interface and its ergonomics.

4.4. Guidance and Regulation

To overcome privacy issues in the field of ubiquitous computing, Langheinrich in 2001 developed six principles for guiding system design for Fair Information Practices [93]. These principles are:

1. Notice, or the principle of openness, towards the user being monitored regarding the communication and acceptance of privacy policies by, e.g., Platform for Privacy Preferences (P3P).
2. Choice and consent of the user being tracked or monitored.
3. Anonymity and pseudonymity being offered to the user.
4. Proximity and locality to limit activities to situations the user is present in.
5. Adequate security for communication and storage.
6. Access and recourse, to effectively limit access to resources in terms of purpose relevancy and time.

Similarly, the International Working Group on Data Protection in Telecommunications in its Report and Guidance on Privacy in Social Network Services (Rome Memorandum) [69] created a guidance for regulators and service providers, and users of social networks to overcome the risks for privacy and security found by the group.

It guides *regulators* to, e.g., ensure that service providers limit data collection and allow users to make informed choices, introduce an obligation to data breach notification for providers, and to improve integration of privacy issues into the educational system.

The group continues to guide SNS *providers* to, e.g., be transparent and open information of users, provide privacy-friendly default settings, improve user control over use of profile data within the community, and to improve and maintain security of information systems.

Finally, the advise the group addresses to *users* of social networks is, e.g., avoid the use of real names in a profile, to respect the privacy of others, to be informed about the service provider and its surroundings, to configure privacy friendly settings, and to use opportunities to control how data is used.

It is disputable whether any commercial service provider would agree to design a service according to these guidances: The requirements listed are counterproductive to its goals. Consequently, the integration of regulations to solve the defined problems is indispensable, because the interests of users and providers are at odds.

The two named guidances address a lot of problems named in Chapter 3, but fail to address the problems of Privacy Balance, Privacy Paradox and Befriending Strangers. To approach these problems, users should be educated by their governments to learn how to use these systems in order to make better informed choices.

For the approach that is about to be introduced in the following chapter, these works provide theoretical background knowledge and many aspects that will be taken into account when designing a privacy-preserving platform. Since the approach is uncommercial, the advices addressed to providers will be taken into account in order to provide users with the necessary transparency and control features to effectively protect their privacies.

4.5. Policies

An important means to solve problem areas defined in Chapter 3 are Access Control Policies and Data Handling Policies. These policies can be implemented and used as tools provided to users to control who has access to what information under what conditions, and to define what is allowed to happen with an information uploaded or produced on a service.

4.5.1. Access Control Policies

Access Control Policies (ACP) define what kind of access is granted to which data to whom under what conditions. The content to be accessed can be a file, a database record, a Website, also an SNS profile page, or a service to be called. As Fong et al. concentrated on modeling the access control model of Facebook [51], Pekárek

and Pötzsch [118] provide a more general list of different access control mechanisms for use in SNSs and CWs:

Access based on identifier: The access to a piece of information is granted based on the requestor's identifier, i.e. their user ID or handle. E.g., a user's profile page or personal content is accessible for befriended users in an SNS.

Access based on roles: Many CWs and some SNSs provide roles that can be assigned to users to grant or limit access to particular information. Although roles are an intuitive concept, the authoring of rules can be difficult for users [110].

Access based on groups: Groups are a collection of user identifiers and thus a proper tool of granting a collection of users access to information. Furthermore, a group implies roles, e.g. creator and member, which can be used for further access details, e.g. read and/or write access or the right to invite further users.

Access based on properties: This access mechanism uses properties of the accessing subject, e.g. its age, or the object about to be accessed, to decide on whether to grant access.

Access based on context: This mechanisms uses the context of the access, e.g. the time or location, but also the system state, such as its load, to decide on the access being granted.

ACPs are an important approach to the problem of Absence of Control in the form of Access Control Lists (ACL). ACLs are a set of rules that can combine different ACPs. E.g., an SNS could implement an ACL that defines the accessibility of an information to a certain group of individuals plus users having an administrator role.

The challenge of a privacy-preserving SNS framework is to, on the one hand, provide an Access Control Engine (ACE) to implement the ACLs derived from the features, and on the other hand to provide the users with an understandable Graphical User Interface (GUI) to effectively control the advanced ACPs to protect and manage the visibility of their data. However, the provision of usable ACLs does not solve the problem of Misunderstood Reach which arises from a missing transparency of information flow.

4.5.2. Data Handling Policy

A Data Handling Policy (DHP) is a set of rules defining the treatment of a piece of sensitive information. This includes for what purpose it may be used, e.g. marketing or research, to which third parties it may be disclosed and how long it may be stored on the server. In order to provide DHPs on an SNS, the following features are required:

- A form provided to the user about to upload or create an information to display or configure the DHP that is to be attached as meta-data to the information.

- A mechanism to package data and meta-data for transmission.
- An enforcement system to interpret the DHP and to take action accordingly.

Today's SNS provide DHPs usually as marginally configurable preferences. As reported in Section 4.3, the P3P project [39] is an approach that aims at providing users with a tool to configure privacy settings on a meta-level. However, DHPs as a concept touch on the problems of Absence of Oblivion, Privacy Theatre, and Information Diffusion to Third Parties.

4.6. Economic Models

As described in Section 3.1, Privacy Balance is a rational decision making process of users. Varian [140], Cate [35], Preibusch [121], and Hann [60] have researched Privacy Balance as an economic trade-off function. These approaches are important to understand how such decisions are made and how they can be modeled. Bonneau and Preibusch [27] question that the problem of Privacy Balance and Privacy Paradox can be solved with privacy designs, because the user is no *homo economicus*[5], but psychologically distorted in his or her decision making process. The problem is not the utility maximization problem, but the user's unforeseeable subjective valuation of benefit and privacy costs. Furthermore, the authors are anxious about service providers exploiting the user's inability to make fully informed choices. They also exclude heuristics as decision rules, since they can be tricked, excluding them as a solution for the problem of Privacy Theatre (Section 3.2).

Accordingly, the named approaches can not be used. The problems of Privacy Balance and the Privacy Paradox will be approached by an increased transparency of the information flow to users in order to empower them to make fully informed choices.

4.7. Openness and Data Portability

Openness and data portability are approaches to the Property and Data Portability problem (Section 3.9) and technically a matter of machine-accessibility to information.

To realize machine-accessibility to information (cf. Section 2.3.2 about the Semantic Web), numerous works were published on standards for protocols and exchange formats for authentication, authorization, exchange of information, syndication, interlinking of services, and synchronization. These fragments of technology are fundamental for the emergence of the Social Web (Section 9.1) which is discussed in the outlook chapter. The current fashion is to focus on small building blocks instead of a single specification [126] resulting in a toolbox of functionality and formats that is openly accessible and combinable to speed up innovation. These building blocks are combined to higher aggregated approaches by non-profit ini-

[5] `http://en.wikipedia.org/wiki/Homo_economicus`, accessed November 2010

tiatives, such as the DataPortability Project[6], the Open Stack[7] and DiSo Project[8] (Distributed Social Networks). Non-profit initiatives, such as the OpenWeb Foundation[9] support community-driven non-proprietary standards with a lightweight framework for dealing with the legal requirements.

In the following, the most important standards are introduced.

4.7.1. Authentication

A protocol for authentication currently adopted widely by services like, e.g., Google, Yahoo and MySpace, is OpenID[10] [126]. OpenID is a federated, user-centric identity system on the basis of the Hypertext Transfer Protocol (HTTP) that supports both, Uniform Resource Identifiers (URI) and Extensible Resource Identifiers (XRI). Having one account with credentials per service is a burden for users who want to adopt new services. Being forced to remember many passwords, users tend to reuse or simplify passwords increasing the vulnerability of accounts. Also, the barrier to trying a new service is lowered if the process of creating a new account is simplified or partially automated. With OpenID, users are enabled of reusing one account for many services. OpenID provides an authentication mechanism and very basic user-controllable profile information. Alternatively to OpenID, which uses address-based identifiers, there are also card-based identity approaches, such as Windows Cardspace or The Higgins Project, which provide similar functionality.

The principle of using one account for many services is not new: many approaches have been implemented, including Windows Cardspace, Google Account, Yahoo ID, and Deutsche Telekom Netzausweis. These systems intend to improve the quality of services by accumulating knowledge about users across services centrally and providing this benefit back to the connected services [87]. Architecturally, they are centralized approaches hosting the user models at a central server. OpenID is a federated approach allowing several independent Identity Providers and Relying Parties to interact by supporting the same standard. Approaches allowing the user to re-use login credentials are called Single-Sign-On (SSO) solutions.

Some approaches, namely Google Account, MySpaceID and Yahoo ID, strategically decided to support OpenID, and became OpenID Identity Providers allowing their users to use their accounts for foreign services supporting Open ID login (Relying Party). It is important to note that Google and Yahoo are not OpenID Relying Parties. The strategy behind this is to reach out to the Web allowing Web sites to use authentication mechanisms for, e.g. logging in to comment a blog article, and embed functionality blocks into their sites, while at the same time limiting the principle of Data-Drivenness (Section 2.2.1) by not allowing users to use services without creating an account locally [117].

[6] `http://www.dataportability.org`, accessed September 2009
[7] `http://developer.yahoo.net/blog/archives/2008/12/the_open_stack.html`, accessed September 2009
[8] `http://www.diso-project.org`, accessed September 2009
[9] `http://www.openwebfoundation.org`, accessed September 2009
[10] `http://www.openid.net`, accessed September 2009

4.7.2. Authorization

As described in Section 2.2.1, a core concept of Web 2.0 is the understanding of the Web as a Platform. This implies the interlacing of services and functions via interfaces. As OpenID simplifies the authentication of users at many services, OAuth[11] standardizes the authorization of a client to access an API of another service's server. The intended access to data often affects private information, which involves the user in the authorization process. The key idea is to permanently grant a client a revokable authorization to access or change a particular subset of information.

Following the lightweight programming model of Web 2.0 (Section 2.2.1), many services, e.g. Twitter[12], cultivated an eco-system of countless third-party applications accessing data to reuse it for other or similar purposes, e.g. the porting of functionality to a different device or the mashing of different services' data to something entirely new (Mashup)[13]. Without an authorization mechanism, such as OAuth, users are forced to expose their service's login credentials to the third-party service in order to enable it to access the user's data via an API. The increased vulnerability of the account is aggravated by the necessity of the third-party to store the credentials in an unencrypted form, which not even the original service provider does according to state-of-the-art data storage practices. This issue is called the Password Anti-Pattern, belongs to the problem of Security and Data Protection and is solved by the support of OAuth by the involved parties.

4.7.3. Application Programming Interfaces

In the past years of social software, a plethora of different APIs and exchange formats have been invented and released. Focusing on the concept of the Web as a Platform (Section 2.2.1), this heterogeneity hinders a large-scale or automated interweaving of the different platforms: Lacking a semantic description of the interfaces, a machine can effectively not understand the function or contents of the output of an API function (for further reading cf. Section 9.2.1 of the outlook chapter).

In late 2007, Google released OpenSocial[14], which is a set of APIs designed to help accessing data across social software systems. OpenSocial acts as an intermediate layer which needs to be supported by both parties, and can be used

- to synchronize or bridge two SNSs,
- to carry small building blocks of social software to Web sites, e.g. friend lists to embed; or
- for developers to have a standardized interface at service providers to dock their gadgets on.

OpenSocial—now a product of the OpenSocial Foundation—was mainly adopted for the latter purpose: Following the trend to provide application platforms on SNSs,

[11] `http://www.oauth.net`, accessed September 2009
[12] `http://www.twitter.com`, accessed September 2009
[13] In 2007, Twitter counted ten times more traffic on their API than on their Website (`www.readwritetalk.com/2007/09/05/biz-stone-co-founder-twitter`, accessed December 2008)
[14] `http://www.opensocial.org`, accessed September 2009

the service providers featured OpenSocial to ease the effort for external developers porting their software to the platform. This way, developers can relatively easily port one application to two platforms supporting OpenSocial, e.g. to iGoogle[15], XING or StudiVZ[16].

4.7.4. Exchange Formats

To receive content from a service in a standardized format from a provider, Real Simple Syndication (RSS) and Atom Syndication Format are the de-facto standards. They annotate article-like information with a standardized set of meta-information and render it in Extensible Markup Language (XML). The client can, according to the specification, transform the article to its own database or presentation view.

Breslin et al. in [32] and later Bojars et al. [21, 22, 23, 24] describe how to port and interlink the data a user created on an SNS by using the Semantic Web ontologies Semantically-Interlinked Online Communities (SIOC) and Friend-of-a-Friend (FOAF). The SIOC ontology describes concepts in the SNS realm, including users and their profile information, as well as their friendships, comments etc. FOAF is an ontology for describing the relationship between users spanning a Social Graph. Using standardized formats for this information makes user information and user-generated content machine-readable across SNSs. A driver for the authors was the inferring of knowledge across SNS.

Ankolekar et al. in [11] also underlines how the emerging Social Web can benefit from Semantic Web technologies and infrastructures by using it to interchange data and by making knowledge machine accessible. Accordingly, Mika in [103] presents a system called Flink that leverages Semantic Web technology to reason on users' profile information and performing Social Network Analysis (SNA) on the social connections which it extracted from a number of sources.

4.7.5. Annotation

Two important standards to invisibly annotate Web sites made for human consumption and to make information accessible to machines are Microformats[17] [76, 8] and the Resource Description Framework in Attributes (RDFa) [6].

RDFa is an extension to the Resource Description Framework (RDF) [99] and specifies a set of Extensible Hypertext Markup Language (XHTML) attributes that can be used by the service provider to embed machine-readable information inside Web sites. RDFa provides an expression of simple and more complex datasets, i.e. knowledge about entities in form of attributes and relations.

Microformats are a collection of simple, open data formats to embed machine-readable information inside Web sites. These data formats are embedded via the *rel*-tag attributes of XHTML nodes or simple hierarchies of nodes. Microformats can only express very simple information, e.g. people, events, ratings, reviews, qualified friendships via XHTML Friends Network (XFN), but no sophisticated relation between information.

[15] `http://www.google.com/ig`, accessed September 2009
[16] `http://www.studivz.net`, accessed September 2009
[17] `http://microformats.org`

4.7.6. Real-Time Protocols

A trending topic since 2009 is the Real-Time Web, which stands for real-time communication and spreading of information for syndication on the Web[18]. A very important aspect of the Real-Time Web is the syndication of content, i.e. real-time distribution of time-critical information, such as news, stock prices or sensor data, for re-use on other platforms, applications or sites. The traditional mechanism has a publisher providing a newsfeed that interested users subscribe to via a client, which regularly pulles updates from the publisher. The problem is that a real-time push mechanism typically is too costly for publishers if there is too much content to send or too many subscribers to notify. A recent approach to this problem is the PubSubHubbub protocol[19], which is a publish/subscribe protocol trying to minimize traffic and managing costs for publishers by placing a hub between the publisher and the subscriber to manage the distribution of topics to subscribers.

In a nutshell, PubSubHubbub consists of the following aspects:

- Instead of hosting their own newsfeed, publishers indicate the hub(s) providing their content. Interested users subscribe to one or more of the advertised hubs in order to receive updates.
- Publishers ping their hubs about changed topics.
- The hub fetches minimal meta-data about the changed topic, compares it with the topic's previous state and, in case of a difference, re-fetches the topic and enqueues all subscribers for notification.

It uses RSS or Atom as exchange formats. An alternative and similar mechanism is RSS Cloud[20].

A protocol concentrating on real-time communication, such as chatting or the spreading of presence information, is the XML-based Extensible Messaging and Presence Protocol (XMPP). XMPP acts as a message-oriented middleware for cross-server communication.

Both, PubSubHubbub and XMPP specify a decentralized architecture for an open and distributed way of spreading content across platform borders. In the context of the emerging Social Web (Section 9.1), XMPP will play a role as technical backbone for cross-SNS communication, e.g. chatting, and notification of presence information, e.g. online status of friends. PubSubHubbub enables the meta-level spreading of information, e.g. status updates, news and sensor data.

4.8. Summary

This chapter presented different approaches and research disciplines and grouped them by approach fields. All approaches intend to solve one or more problem areas described in the previous chapter. In the following, the described approach fields are summarized briefly:

[18] `http://www.readwriteweb.com/archives/top_5_web_trends_of_2009_the_real-time_web.php`, accessed June 2010

[19] `http://pubsubhubbub.googlecode.com/svn/trunk/pubsubhubbub-core-0.2.html`, accessed September 2009

[20] `http://en.wikipedia.org/wiki/RSS_Cloud`, accessed September 2009

Access Control. The presented works focus on controlling the access to Web sites. The main problem with access control is an unreasonably large effort for users to manually select the audience for every bit of published information. Most of the works therefore research access control in an automated manner, i.e. trying to compute the Social Context. In remains unclear whether users want this selection to be automatically computed or whether they prefer to control the audience by hand.

Architectures and Frameworks. Many researchers worked on different centralistic and distributed approaches to solve privacy problems on SNSs. Some focus on encryption and distribution of data to solve the problem of the Panoptic Provider, which indirectly solves other problems, e.g. Security or Information Diffusion to Third Parties. Others presented interesting theoretical work on granting users feedback and control over their data in order to make informed choices.

User Interface and User Experience. Researchers in this area presented various approaches, e.g., usable policy authoring tools to empower users to create their own privacy policies. Others helped users control their privacies by providing a feature to view their profile pages through the eyes of another user of their choice. However, the UI is an important area to empower users to make informed choices. Also, UX, i.e. usability, intuitiveness, easy and joy of use are key to this goal. Not enough work has been done in this area.

Guidance and Regulation. Various works list aspects and features to be fulfilled by providers in order to preserve their users' privacies. Generally, regulation is a promising way to approach those problems which are intentionally caused by providers to better reach their strategic goals of increasing their databases and gaining profits. Guidance for users is an important approach if users, especially younger ones, are educated regarding the implications of their online activities in order to protect their own and the privacy of others.

Policies, namely Access Control Policies and Data Handling Policies, are a means to express who has access to what information under what conditions, and to define what is allowed to happen with information uploaded or produced on a service. Policies are a good way to explain how a system works and to avoid future misunderstandings on how data is handled.

Economic Models are an approach of some researchers that attempts to model Privacy Balance through economic models. Also, works have been discussed that deny the appropriateness of these models for the problems of Privacy Balance and Privacy Paradox based on the fact that humans are no *homo economicus*, but distorted in their decision making process.

Openness and Data Portability. The works presented in this field deal with the problems of data property and portability. Because providers seek to increase their databases (cf. Web 2.0's key factor of Data-Drivenness, Section 2.2.1), they grant access to as little information as possible to other parties. However, this problem will be solved in small steps through the pressure of users and competitors and emerges in a Social Web (Section 9.1). The works presented

deal with standards for functionality and exchange formats that server as building blocks for inter-SNS operability.

Table 4.1 maps the discussed approach fields with their theoretical and practical works to the problem areas proposed in the previous chapter.

	Problem Areas (Chapter 3)								
	Privacy Balance and The Privacy Paradox	Privacy Theatre	Befriending Strangers	Misunderstood Reach	Absence of Control and Oblivion	Secondary Privacy Damage	Security and Data Protection	Information Diffusion to Third Parties	Property and Data Portability
Access Control			π	π	π				
Architectures and Frameworks				π	π	π	π	π	τ
User Interface and User Experience	τ	π	τ	π	τ				
Guidance and Regulation				τ	τ	τ	τ	τ	τ
Policies		τ			τ			τ	
Economic Models	τ								
Openness and Data Portability									π

Table 4.1.: A mapping of approach fields and their theoretical and practical works to the problem areas. τ denotes theoretic and conceptual approaches, π indicates practical (partial) solutions.

The following chapter defines requirements for SNS, which also form the fundament for the prototypic platform approach introduced in the subsequent chapter.

5. Addressing the Problem Areas of Privacy: Requirements

After having introduced a novel taxonomy for problem areas in the area of SNS (Chapter 3), we reported relevant works done in these areas (Chapter 4) and grouped them by thematic approach fields.

This chapter intends to formulate requirements for SNS platforms in order to both, provide their users with the necessary privacy awareness and tools for privacy protection. Consecutively, the list of requirements is evaluated with a selection of representative SNSs, and the results are discussed.

5.1. Privacy Requirements

In the following, we define a list of requirements for providers to offer a platform that protects users' privacies or empowers them with the needed awareness. The requirements can be clustered into four groups: transparency and access control, relationships, identity management and data handling. Every requirement explicitly lists the problem areas of Chapter 3 as well as the work sections of Chapter 4 it relates to.

5.1.1. Transparency and Access Control

The following requirements (R) demand functionality to provide users with transparency and access control.

R1: Visualize to users who has access to their data. An application must provide its users with transparency of who has access to their profile page[1] or any other information created or files uploaded. Such features decrease the problems of the Privacy Paradox, Privacy Theatre and Misunderstood Reach (Chapter 3) by permanently reminding the user of the possible threat to privacy by presently listing the audience that would otherwise have been fully or partially hidden. This requirement furthermore warns users from Befriending Strangers, since the effect of them being granted access to sensitive information is made transparent. Theoretical work on this requirement was done concerning Economic Models, Architectures, Guides and Regulation proposes, Policies and UIs (Chapter 4).

R2: Allow users to control the accessibility of their data. In addition to R1, control mechanisms are an important means to empower users to not only understand but to control the accessibility to their profile page and all other

[1] As stated in Section 2.2.2, the profile page is by definition a basic building block of an SNS.

information created on or uploaded to the application. In general, these settings should be *restrictive by default*. This requirement approaches the problems of Misunderstood Reach and Absence of Control and Oblivion, because users are empowered with control over what profile information is accessible to whom. Theoretic relevant work has been done on Architectures, Control, UIs and Policies.

R3: Prevent unauthorized access to and downloading of user profiles. It is fundamental for applications to implement access control mechanisms that forbid unauthorized access to user profiles, imcluding machine access for downloading profiles (crawling) for digital dossier creation and human access. The requirement limits a profile's invisible audience, i.e. the problem of Misunderstood Reach, increases Security and Data Protection, and prevents unintended Information Diffusion to Third Parties. Related work has been published as Guides and Regulation proposes, theoretic Architectures, and Policies.

R4: Prevent full access to user data by the service provider. A Panoptic Provider itself is a threat to users' privacies. Providing an infrastructure that technically avoids full access to primary user data (data created by users) and secondary user data (data or information created about, or links created to users) prevents the provider from being able to access all information. The provider is furthermore hindered from diffusing information to third parties, which effectively diminishes the problems of Security and Data Protection, Information Diffusion to Third Parties as well as Property of data. Related work has been done in the areas of Architectures and Guides and Regulation.

R5: Provide DHPs to be respected. As described in Section 4.5, Data Handling Policies let users understand and control what happens to their information on a meta-level. This requirement provides partial solutions to the problems of Absence of Control and Oblivion because users can define (by a DHP) the purpose and lifetime of information or a file, and the problem of Information Diffusion to Third Parties, because users can declare a secondary usage purpose that way. DHPs also hinder others from damaging user privacies by being prohibited from, e.g. tagging users in a picture which they uploaded. This aspect was touched on related work under Policies.

R6: Empower users to control their user model. As described before, service providers create user models that contain inferred assumptions about the user's interests and habits. The model revolves around information entered both, explicitly, e.g. profile information entered, and implicitly, e.g. application browsing behavior viewing users or items, friendship links, group memberships, event attendances (user tracking), or inferred information. This knowledge about users is often sold to business partners and used for, e.g., targeted advertisement. However, users must have a transparency of the model created about them and be able to review and control the information within it, e.g., by editing or deleting details. The requirement pertains to the problems of Absence of Control and Oblivion, since the user can control information about him or her, and Information Diffusion to Third Parties, because exposure can be limited or prevented. Related work has been done in the areas of Architectures and Guidance and Regulation.

5.1.2. Relationships

While the previous requirements focus on empowering users to have transparency and control regarding the access to and purpose of their information, the following deal with the management of relations to others.

R7: Creation and management of friendship links. This requirement enables users to maintain Social Context (Section 2.1.1) in their online lives by befriending other users, which implies the granting of access rights to mutual information for (a more engaged) future interaction. To overcome the problem of Befriending Strangers, applications should openly communicate befriended users and offer ways to further differentiate these by, e.g., offering a friend lists feature that allow users to segment friends to, e.g., family, friends and acquaintances. Such qualified friend links allow for an improved mapping of users' Social Context to the application and ensure Contextual Integrity for future communication amongst users. Besides Befriending Strangers, the requirements concerns the problems of Absence of Control and was dealt with in works about theoretical Architectures.

R8: Management of groups and their data. Besides friendship links, groups also empower users to map Social Context to online interaction through group memberships. Groups also help qualifying social links between users. Functionally, groups form a platform for further content, e.g. photos, events or articles, for a defined audience, i.e. a recipient list. Therefore, users need to be empowered to create groups, and manage them by inviting, joining and leaving. Through group member management, access control to data and information, that belongs to a group, is provided. This requirement decreases the problems of Befriending Strangers by offering a tool for qualifying a relation to other users, Misunderstood Reach by communicating group memberships as the audience for information created within, and Absence of Control by providing a tool for targeted communication to a user according to a Social Context. Related work was reported of in the area of theoretical Architectures.

5.1.3. Identity Management

The following requirements deal with identity management and account deletion.

R9: Allow for export and synchronization. An application should allow its users to synchronize or export their data for use in other services. Existing exchange formats and protocols should be considered for this purpose. This requirement regards the problem of Property and Data Portability and was dealt with in related work in the area of Architectures.

R10: Allow for account deletion. Applications should provide functionality to delete an account and all related information and files. Users need the possibility to opt-out of a service by wiping all information provided by them. Besides the problem of Property and Data Portability, this function also relates to the problems of Absence of Control and Oblivion, because users can make a

service forget about them. This aspect was also listed in the related work areas of Guidance and Regulation and Architectures.

5.1.4. Data Handling

This section's requirements concern the providers' dealing with data. They cover the amount of stored data as well as mechanisms to provide the needed security.

R11: Provide purpose limitation for data usage. This requirement describes the principle that data tracked for user modeling, i.e. the sensoring and storage of data about user activity, should not be collected blindly, but limited to named purposes. This includes data exposed to third parties. An application should not only follow this principle, but also communicate the purposes and potential third party recipients, e.g. in an understandable way in its terms of service. This requirement concerns the problems of Security and Data Protection, because the data stored is limited, and the problem of Information Diffusion to Third Parties, because the flow is restricted to communicated purposes and parties. The issue was mentioned in the context of Fair Information Practices in the area of Guidance and Regulation and Policies.

R12: Ensure data storage security. An application should ensure that the data stored about users and their information is secured against unauthorized access and manipulation, including identity theft and spamming. This point regards the problem of Security and Data Protection and is also part of Fair Information Practices covered in the area of Guidance and Regulation.

R13: Minimize collected data. In addition to R11, an application should not only limit data collection to communicated purposes, but also follow a principle of minimizing its dataset by, e.g., erasing data of deleted accounts (R10) or by providing an oblivion feature for deleting data after a certain retention time. Touching the problems of Absence of Oblivion and Security and Data Protection, this requirement was also dealt with in Fair Information Practices mentioned in Guidance and Regulation.

R14: Declare data ownership to users. An application should openly declare and communicate data ownership to its users. Although ownership of produced data is implicitly regulated by legislation in some countries, an application should clearly communicate the question of data property to its users as part of Fair Information Practices. The aspect approaches the problem of Property and Data Portability and was covered in Guidance and Regulation.

R15: Respect national privacy standards. Applications acting internationally should respect the national privacy standards and laws of the countries they are running their services in. The problem areas this requirement relates to depends on the particular countries' laws. Legal compliance is covered in the guides to regulation discussed in the work section Guidance and Regulation.

Table 5.1 provides an overview of the relation between the named requirements and the problem areas defined in Chapter 3.

		Problem Areas (Chapter 3)								
		Privacy Balance and The Privacy Paradox	Privacy Theatre	Befriending Strangers	Misunderstood Reach	Absence of Control and Oblivion	Secondary Privacy Damage	Security and Data Protection	Information Diffusion to Third Parties	Property and Data Portability
Transparency	R1	√	√	√	√					
Access control	R2				√	√				
Unauthorized access	R3				√			√	√	
Panoptic Provider	R4							√	√	√
DHPs	R5						√	√	√	
User model control	R6					√			√	
Friendships	R7			√	√	√				
Groups	R8			√	√	√				
Data export and sync.	R9									√
Account deletion	R10					√				√
Purpose limitation	R11							√	√	
Data security	R12							√		
Minimize data collection	R13					√		√		
Data ownership	R14									√
National laws	R15				√	√	√	√	√	√

Table 5.1.: The table provides an overview of the mapping of requirements to the problems areas defined in Chapter 3.

5.2. Evaluation of Existing Platforms

This section evaluates how a selection of today's most popular SNSs perform in terms of the requirements defined in the previous section.

The selection consists of two general-purpose SNSs and two business-networking SNSs. All of them comply with Boyd's definition of SNSs [31]. The two categories were chosen, since the platforms provide similar features, but emphasize different use-cases. As private conversation with friends, including the sharing of multimedia content, is the primary use-case of general-purpose SNSs, user activity on business-networking SNSs consists more of administering CVs and connecting to business contacts.

The categories also differ regarding the platforms' revenue models. As general purpose SNSs focus on advertising, business networking SNSs earn a significant share of their revenue with fee-based premium accounts[2]. Further possible revenue models are micro payments, partnerships or virtual currencies. Although the focussed revenue model affects the platform strategy, all models require the maximization of the user-base and the network effect to increase the perceived value of the service.

The selection represents one general-purpose SNS and one business-networking SNS per country. Germany was selected as an exemplary country with stricter laws concerning privacy protection, and the USA as one with moderate laws regarding the protection of the privacies of users. Table 5.2 provides an overview over the selection of SNSs: All of them have millions of users (rounded) and rank high in traffic ranks provided by the Alexa service[3]. VZ Netzwerke consists of the three equivalent platforms StudiVZ, MeinVZ and SchülerVZ.

Name	Category	Users (M)	Country	Traffic Rank
Facebook	General-purpose	500	USA	2
VZ Netzwerke	General-purpose	15	Germany	409/489/1042
XING	Business networking	10	Germany	209
LinkedIn	Business networking	70	USA	23

Table 5.2.: The selection of SNSs to evaluate the fulfillment of the requirements on.

5.2.1. Facebook

Facebook was founded in 2004 and counts more than 500 million active users. The average user creates 90 pieces of content each month, has 130 friends and is connected to 80 community pages, groups and events.[4]

Most of Facebook's revenue comes from advertising[5]. Consequently, it seeks to maximize its user-base in both aspects, i.e. quantitatively and qualitatively: It encourages the acquisition of new users plus the networking amongst them to

[2] `http://corporate.xing.com/english/investor-relations/basic-information`, accessed December 2010

[3] http://www.alexa.com

[4] `http://www.facebook.com/press/info.php?statistics`, accessed December 2010

[5] `http://en.wikipedia.org/wiki/Facebook`, accessed December 2010

stimulate content creation. Hence, the user-base and knowledge about users is maximized to improve advertisement targeting.

Transparency (R1): Transparency is not conditioned. For data or information published on the profile page, no list of recipients is presented to make the audience transparent. For users, it is hard to comprehend who has access to the profile page (or parts of it). Additionally, users are stimulated to befriend others, which makes their profile page accessible. Good: Users can watch their profile page through the eyes of another user to check the information presented.

Access control (R2): The ACE implements *access based on identifier*. For profile parts, radius-style access control is configurable, e.g.friends-only, friends-of-friends, network or public. Friend lists and groups can be defined and used to qualify friends and to provide *access based on groups*. Friend lists can be used to limit the audience on the general NewsFeed, groups to have a separate chat-room-like space for conversation. All features are present, but not promoted, hard to find and use, and not restrictive by default [27].

Unauthorized access (R3): It is not possible to say whether the software mechanisms of the ACE to authorize access to profile pages is free from errors. However, the provider claims to have implemented mechanisms to detect and prevent automated crawling of profiles.

Panoptic Provider (R4): The provider has access to all user data.

DHPs (R5): The provider does not support user-configurable DHPs.

User model control (R6): Users can edit their explicit user model, i.e. the general data, such as interests, they added to their profile page. The provider emphasizes user modeling through the liking of brand pages. The deletion of this, i.e. the un-liking, is solved rather un-ergonomically to keep users from doing so. It is unknown to what extend the provider performs implicit user modeling and how it can be controlled or affected by the users.

Friendships (R7): The provider supports the befriending of users and utilizes this for mutual profile page and profile parts accessibility. Friend lists are provided for controlling Contextual Integrity, but not sufficiently promoted as a means for access control.

Groups (R8): Groups are provided and allow users to create and control Contextual Integrity by limiting group content to its members. In a PR event, groups were introduced as a means for this purpose, but their creation and usage is insufficiently promoted on the platform itself. Consequently, the groups feature is used by only very few users.

Data export (R9): A feature for data exporting is provided, but insufficient, since profile data is exported as a static file that the majority of users do not know how to handle. Data synchronization is insufficiently provided through a proprietary API.

Account deletion (R10): Accounts are removable, but the data produced on the platform is not erased. The function is not easy to find and camouflaged with a feature for account deactivation. Deactivated users are not deleted and can still be added to groups or tagged in photos. Users still receive emails and are able to re-activate their account with a single click.

Purpose limitation (R11): The provider collects far too much data (it has a data collection score of 0.9 in an interval from 0 to 1, where 1.0 represents the worst possible data collection behavior [27]). The provider states in its privacy policies that it shares *un-anonymized* data with third parties.

Data security (R12): It is unknown how securely the provider stores its data. The provider names the geographic data location in its privacy policies.

Minimize data collection (R13): No data retention limitation is specified in the privacy policies.

Data ownership (R14): Data ownership is not attributed to the user.

National laws (R15): It is unknown to what extent the provider respects the national laws of the countries they act in. For the EU, the provider claims to be Safe Harbor-compliant. It specifies national laws in its privacy policies.

5.2.2. VZ Netzwerke

VZ Netzwerke consist of three SNSs: StudiVZ, SchülerVZ and MeinVZ, of which the first, StudiVZ, was founded in 2006. The Platforms count a total of 17 million users. Their revenue model mainly consists of advertisement[6].

Transparency (R1): Transparency is not conditioned. For data or information published on the profile page, no list of recipients is presented to make the audience transparent. For users it is hard to comprehend who has access to the profile page (or parts of it). Additionally, users are motivated to befriend with others which makes accessible their profile page.

Access control (R2): The ACE implements *access based on identifier*. For profile parts, radius-style access control is configurable, e.g.friends-only, friends-of-friends, network or platform-public. Profile pages cannot be set public to protect user privacies. No further configurable access control mechanism is given to qualify friends or to separate audiences based on context. The settings are barely promoted and not restrictive by default [27].

Unauthorized access (R3): It is not possible to say whether the software mechanisms of the ACE to authorize access to profile pages is free from errors. The platform claims to have implemented mechanisms to detect and prevent automated crawling of profiles.

Panoptic Provider (R4): The provider has access to all user data.

DHPs (R5): The provider does not support user-configurable DHPs.

[6] `http://www.studivz.net/l/about_us/1`, accessed December 2010

User model control (R6): Users can edit their explicit user model, i.e. the general data, such as interests, they added to their profile page. It is unknown to what extent the provider performs implicit user modeling and how it can be controlled or affected by the users.

Friendships (R7): The provider supports the befriending of users and uses this for mutual profile page and profile parts accessibility. No friend lists or other tools for qualifying friendship links are provided as a means of controlling Contextual Integrity.

Groups (R8): Groups are provided and allow users to create and control Contextual Integrity by limiting group content to its members. Groups are not promoted as a means for privacy protection.

Data export (R9): No export or synchronization functionality is provided.

Account deletion (R10): Accounts are deletable, including directly related data. The presence of the link to this feature is sufficient.

Purpose limitation (R11): The provider collects too much data (data collection score: 0.76). The provider states in its privacy policies that it does *not* share data with third parties.

Data security (R12): It is unknown how securely the provider stores its data. The provider names the geographic data location in its privacy policies.

Minimize data collection (R13): A data retention limitation is specified in the privacy policies.

Data ownership (R14): Data ownership is not explicitly attributed to the user.

National laws (R15): It is unknown to what extent the provider respects the national laws of the countries they act in. It specifies national laws in its privacy policies.

5.2.3. XING

XING was founded in 2003 and counts more than 10 million members[7]. Its revenue model is based on fee-based premium memberships, e-recruiting and advertising.

Transparency (R1): Transparency is not conditioned. For data or information published on the profile page, no list of recipients is presented to make the audience transparent. For users, it is hard to comprehend who has access to the profile page (or parts of it). Additionally, users are stimulated to befriend others, which makes their profile page accessible.

Access control (R2): The ACE implements *access based on identifier*. For profile parts, radius-style access control is configurable, e.g.friends-only, friends-of-friends, platform-public, or public (including Web crawlers). The profile page

[7] `http://corporate.xing.com/english/investor-relations/basic-information/overview-xing-ag`, accessed December 2010

itself cannot be restricted to be accessible by friends only. Further adjustable access control mechanisms to qualify friend links for contextual conversation are not provided but are of lesser importance: As a business platform, this SNS is a catalog of CVs and business connections. Compared to a general purpose SNS, fewer conversation takes place here that needs to be controlled.

Unauthorized access (R3): It is not possible to say if the software mechanisms of the ACE to authorize access to profile pages is free of errors. The accessibility of profiles to Web crawlers can explicitly be allowed by profile owners. It is *not* known if the platform implemented mechanisms to detect and prevent automated crawling of profiles.

Panoptic Provider (R4): The provider has access to all user data.

DHPs (R5): The provider does not support user-configurable DHPs.

User model control (R6): Users can edit their explicit user model, i.e. the general data, such as interests, which they added to their profile page. It is unknown to what extend the provider performs an implicit user modeling and how it can be controlled or affected by the users.

Friendships (R7): The provider supports the befriending of users and uses this for profile parts accessibility. The profile page itself is accessible to all users. No friend lists or other tools for qualifying friendship links are provided as a means of controlling Contextual Integrity.

Groups (R8): Groups are provided and allow users to create and control Contextual Integrity by limiting group content to its members. Groups are not promoted as a means for privacy protection.

Data export (R9): Contacts (friends) can be exported in a non-proprietary exchange format (VCard). Synchronization of contacts is provided as an Outlook-plugin. No other data can be exported or synchronized.

Account deletion (R10): Accounts are deletable, including directly related data. The presence of the link to this feature is *not* sufficient.

Purpose limitation (R11): The provider collects too much data (data collection score: 0.62). The provider states in its privacy policies that it does not share data with third parties.

Data security (R12): It is unknown how securly the provider stores its data. The provider does *not* name a geographic data location in its privacy policies.

Minimize data collection (R13): No data retention limitation specified in the privacy policies.

Data ownership (R14): Data ownership is not attributed to the user.

National laws (R15): It is unknown to what extent the provider respects the national laws of the countries they act in. It specifies national laws in its privacy policies.

5.2.4. LinkedIn

LinkedIn has more than 85 million members in over 200 countries. Its three revenue streams are corporate accounts, advertising and fee-based premium accounts[8].

Transparency (R1): Transparency is not given at all. It is not clearly communicated who has access to the user's profile page, which is thus hard to comprehend. Additionally, users are motivated to befriend others, which makes their profile page accessible.

Access control (R2): The ACE implements *access based on identifier*. For profile parts, radius-style access control is configurable, e.g.friends-only, friends-of-friends, platform-public, or public (including Web crawlers). The profile page itself cannot be restricted to be accessible by friends only. Further adjustable access control mechanisms to qualify friend links for contextual conversation are not provided but are of lesser importance: As a business platform, this SNS is a catalog of CVs and business connections. Compared to a general purpose SNS, fewer conversation takes place here that needs to be controlled.

Unauthorized access (R3): It is not possible to say if the software mechanisms of the ACE to authorize access to profile pages is free of errors. The accessibility of profiles to Web crawlers can explicitly be allowed by profile owners. It is *not* known if the platform implemented mechanisms to detect and prevent automated crawling of profiles.

Panoptic Provider (R4): The provider has access to all user data.

DHPs (R5): The provider does not support user-configurable DHPs.

User model control (R6): Users can edit their explicit user model, i.e. the general data like interests they added to their profile page. It is unknown to what extend the provider performs an implicit user modeling and how it can be controlled or affected by the users.

Friendships (R7): The provider supports the befriending of users and uses this for profile parts accessibility. The profile page itself is accessible to all users. No friend lists or other tools for qualifying friendship links are provided as a means of controlling Contextual Integrity.

Groups (R8): Groups are provided and allow users to create and control Contextual Integrity by limiting group content to its members. Groups are not promoted as a means for privacy protection.

Data export (R9): Contacts (friends) can be exported in a non-proprietary exchange format (VCard). Synchronization of contacts is provided as an Outlook-plugin. No other data can be exported or synchronized.

Account deletion (R10): Accounts are not deletable, but closable. User data is kept for several reasons, e.g. account recovery.

[8] `http://press.linkedin.com/`, accessed December 2010

Purpose limitation (R11): The provider collects much data (data collection score: 0.48). The provider states in its privacy policies that it shares anonymized data with third parties.

Data security (R12): It is unknown how securely the provider stores its data. The provider names the geographic data location in its privacy policies.

Minimize data collection (R13): No data retention limitation specified in the privacy policies. Additionally, closed accounts are not deleted.

Data ownership (R14): Data ownership is not attributed to the user.

National laws (R15): It is unknown to what extent the provider respects the national laws of the countries they act in. For the EU, the provider claims to be Safe Harbor-compliant. It specifies national laws in its privacy policies.

5.3. Discussion

The evaluation has shown that the question of the fulfillment of the requirements has no binary answer. There are numerous cases where a technical solution is present but remains unused because it is intentionally unergonomic to use or not promoted. Chapter 3 defined this phenomenon as Privacy Theatre (Section 3.2).

Table 5.4 summarizes the above evaluation by assigning ratings to reflect the performance of the requirement fulfillment. The ratings and their scores are defined in Table 5.3.

Rating	Score	Description
−	0	Requirement unfulfilled or fulfillment generally unknown
○	1	Requirement partially fulfilled, fulfillment partially unknown, or realization ineffective
+	2	Requirement fulfilled but hidden from the user, hardly promoted or hardly explained
++	3	Requirement fulfilled, effective, and ergonomic

Table 5.3.: Key for the requirement fulfillment ratings and scores.

For simplicity's sake, the discrete scores for requirement fulfillment quality are equidistant and the requirements as such are equally valued. The goal of the following numeric evaluation is to get a rough picture of how the SNSs score in relation to each other, and in what area of the possible range of points the SNSs are located.

Table 5.4 shows that all SNSs score very low in the realization of requirements for privacy protection of users. In a points range of 0 to 45 possible points, all of them are located in the lower third with their total sum of scores.

Though the SNSs' total scores sum up similarly, the points were gained in different requirement clusters: As Facebook scored high with features for privacy protection (Transparency and Access Control Cluster and Relationships Cluster), the German SNSs, constrained by national laws for privacy protection, gained three times as many points in the Identity Management Cluster compared to their US competitors

Requirement		Facebook	VZ Netzwerke	XING	LinkedIn
Transparency	R1	○	−	−	−
Access control	R2	○	−	○	○
Unauthorized access	R3	○	○	○	○
Panoptic Provider	R4	−	−	−	−
DHPs	R5	−	−	−	−
User model control	R6	○	○	○	○
Transp. and Access Control sub-total		4	2	3	3
Friendships	R7	+	○	○	○
Groups	R8	+	○	○	○
Relationships sub-total		4	2	2	2
Data export	R9	○	−	○	○
Account deletion	R10	−	++	+	−
Identity Management sub-total		1	3	3	1
Purpose limitation	R11	−	−	−	−
Data security	R12	○	○	−	○
Minimize data collection	R13	−	++	○	−
Data ownership	R14	−	−	−	−
National laws	R15	○	○	○	○
Data Handling sub-total		2	5	2	2
Total score		11	12	10	8

Table 5.4.: Privacy rating and scoring of the above evaluation of the SNSs' performance in terms of the realization of the stated requirements. All SNSs score about equally low but gained their points from different requirement clusters.

(cf. Figure 5.1). VZ Netzwerke managed to score twice as many points in the Data Handling Cluster basically by limiting data retention through their privacy policies.

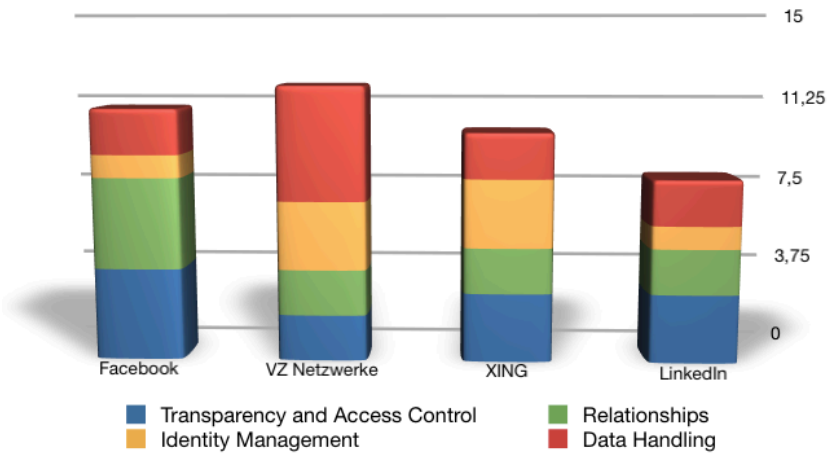

Figure 5.1.: Privacy score the SNSs gained throughout the requirements clusters. Although the four SNSs score comparatively low, the points were gained in different clusters.

However, the evaluation proves that the source of problems is not a lack of knowledge or innovation on the part of the provider regarding how to provide a proper privacy protection, but the beforementioned conflict of interests between providers and users. As providers have to maximize the data and knowledge they collect about and from users, the users are driven by their need to socialize and for esteem and thus communicate with others, ignoring potential risks to their privacies. One way of achieving increased privacy protection scores is the application of regulation and laws. Furthermore, governments should better educate citizens about the implications of sharing private information online.

5.4. Summary

This chapter represents a major contribution of this thesis: A list of requirements was defined to ensure privacy awareness among users and to provide them with tools for privacy protection. The requirements were evaluated in a selection of representative SNS platforms. A privacy score was introduced and computed for the SNSs, and the results were discussed.

The bottom line of the discussion is that SNS platforms to date protect the privacies of their users insufficiently. Tools for privacy protection are either not provided or poorly promoted, so that they remain either ineffective or inefficient. The reason for this is not based on a lack of knowledge or innovation, but the named conflict of interest between providers and users. Much work has to be done in the areas of national regulation to force providers to implement better privacy

protection on their platforms, and education to teach citizens about the implications of sharing private information online.

The next chapters report and discuss the creation of a prototypic approach to provide users with an SNS platform which preserves their privacies. Chapter 6 describes its realization, while Chapter 7 evaluates its effectiveness and efficiency for privacy protection as well as the resulting social network graph. Section 7.5 rates the approach's privacy score and compares it to those of the platforms discussed in this chapter.

6. An Approach for a Privacy–Preserving Social Network Platform

In the previous chapters, we have analyzed the problem areas regarding user privacies in the area of social networking (Chapter 3) and works to approach these areas (Chapter 4). Consequently, we have defined a list of privacy requirements to provide users a privacy awareness as well as tools to make user privacies both, transparent and controllable (Section 5.1). Chapter 5 continued by evaluating four major social network platforms regarding their fulfillment of these requirements.

This chapter introduces a different approach to an SNS platform to provide transparency and control to users in order to enable them to make informed choices to preserve their privacies. The goal is to provide users with an efficient tool for the crucial problem of privacy awareness and protection. Great importance is attached to not only the effectiveness of the tool, but also its efficiency and ergonomics for handling privacy-related tasks.

After outlining the focus of the SNS in a concept and a scenario, a list of application-specific requirements is defined in addition to the privacy-related requirements to analyze the attributes and functionality of the application described by the concept and the scenario. From both requirements lists, a list of features is derived and the realization of the SNS is described in detail: After describing entities and their relations, a framework is introduced to implement the required back-end functionality. On top of this, an application is built to make the features applicable for users. The steps of its creation and aspects to improve the efficiency of transparency and control features are described in detail.

6.1. Concept

The concept presented in the following provides an overview of an SNS that enables its users to share private information and communicate with each other.

The goal is to create a privacy-preserving online communication platform for users and their socio-economic environment. Based on the discussed literature of Chapters 2 and 3, it is assumed that a platform, which provides transparency and control over personal information (and its flow), motivates its user to share highly relevant personal information, since they perceive a more intimate, secure and private atmosphere. Accordingly, the users is given an entity to recommend to each other and to communicate about.

This entity, the central medium for communication of the platform, is a (secret) recommendation, a “Tip”. A Tip is a location-based recommendation for lifestyle and recreational activities. It constitutes a personal recommendation from a friend

or acquaintance and is thereby highly valuable and relevant to its receivers. Because of its relation to the contexts of lifestyle and recreational activities, Tips often have a reference to both, location and time. Consequently, the application must be location-sensitive regarding both entities, "Users"and Tips, in order to compute a relevancy measure for a Tip regarding the user's current location. Furthermore, a Tip features a communication platform that can be used by authenticated receivers for contextual conversation, e.g. for comments.

The Tips as recommendations from other users are so valuable, because the audience is limited to a group of users that know each other personally [56]. That means that the quality of the Social Context, which defines with whom an information is shared (cf. Section 2.1.1), is maximized. These groups of the application are called "Families". As said above, it is expected that the intimate atmosphere of a given and visible Social Context directly and positively influences the quality of the information to be produced and communicated, i.e. the Communication Context (cf. Section 2.1.1), by creating a high relevancy for the audience regarding

- Neighborhood Context, since the communicated information was created by a known person.
- Topic Context, because the sender and the receivers probably have past experiences with the communicated information.
- Recipient Context, since the user that is about to create an information to be communicated is positively affected by the fact that the receivers are friends or acquaintances.

Accordingly, Tips can not be accessed by anyone but the members of a particular Family, which can be created and maintained by users. Families can thus be interpreted not only as conventional groups, but as micro-communities consisting of a defined set of Users and a set of Tips that act as communication mediums. Users are empowered to map their personal socio-economic environment and personas to Families by creating or joining them and by inviting friends.

Access control via Families is inspired by Pekárek's ACP *access based on groups* (Section 4.5.1). Groups are an intuitively understandable means for users to imagine and maintain Social Contexts.

This concept preserves its users' privacy *by design* through the restriction of communication to relevant, closed micro-communities. Users are guaranteed that nobody except a transparent members list of a given Family is granted access to Tips. This paves the way for sharing sensitive and private information, such as private parties.

A drawback of this concept is that users cannot browse around for interesting Families, but must be invited to the groups by friends.

6.2. Scenario

The following scenario gives a formalized example of the named concept to explain the peculiarities of the Families-concept and the underlying access control mechanisms. Exemplary Families, Users, and assigned Tips are described and illustrated, and the implications of the constellations are underlined. The methods used in

the following are formally defined in the upcoming Section 6.6.2, which covers the Access Control Engine.

Let the sets Families F, Tips T and Users U be defined as

$$\begin{aligned} F &:= \{f, g, h\} \\ T &:= \{r, s, t\} \\ U &:= \{u, v, w, x, y, z\} \end{aligned}$$

Let the *members* function (cf. Equation 6.6) return the members of a given Family. For the above defined Families, it returns

$$\begin{aligned} members(f) &= \{u, v\} \\ members(g) &= \{u, x, y\} \\ members(h) &= \{w, z\} \end{aligned}$$

Figure 6.1 illustrates the scenario's Families, Users and the memberships to connect them.

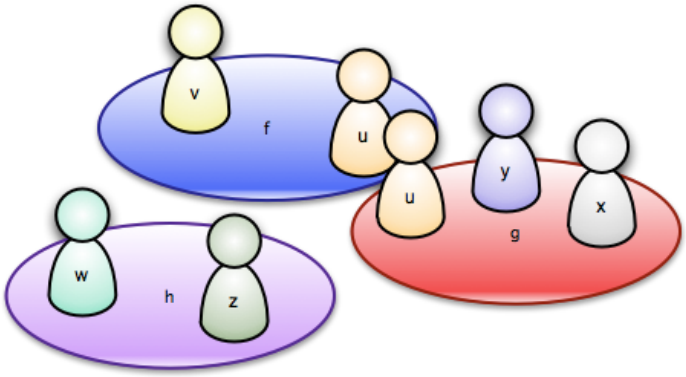

Figure 6.1.: The figure depicts the scenario's Families, Users and their memberships. The accessibility of user profiles for a particular User is controlled through Family memberships. Accordingly, User u has access to all members of Families f and g. Family h and its members are disjoint from Families f and g and thereby form a separated micro-community.

Imagine only Users $x, v \in U$ defined a friendship. That lets the $friends$ function (cf. Equation 6.12) to return the set of friends for a given User deliver

$$\begin{aligned} friends(x) &= \{v\} \\ friends(v) &= \{x\} \end{aligned}$$

A function *users* (cf. Equation 6.14) returns a set of Users that are granted access to the profile page of a given User. For the scenario's Users, the function answers

$$users(u) = members(f) \cup members(g) \quad = \quad \{u, v, x, y\}$$

$$
\begin{aligned}
users(v) = members(f) \cup friends(v) &= \{u, v, x\} \\
users(x) = members(g) \cup friends(x) &= \{u, x, y, v\} \\
users(y) = members(g) &= \{u, x, y\} \\
users(w) = users(z) = members(h) &= \{w, z\}
\end{aligned}
$$

That defines that, e.g., User u has no access to the user profiles of Users w and z.

$$
(members(f) \cup members(g)) \cap members(h) = \emptyset
$$

Note that the scenario defines two disjoint micro-communities whose members neither notice or have access to each other's profile pages, nor do they notice or have access to each other's Tips (cf. Figure 6.1).

To proceed with the scenario, we introduce Tips that Users assigned to Families. Refer to Figure 6.2 for an illustration of the extended scenario. Let the function *assignments* (cf. Equation 6.17) return a set of Tips for a Family in question. The Families in F contain the following Tips

$$
\begin{aligned}
assignments(f) &= \{r, s\} \\
assignments(g) &= \{s\} \\
assignments(h) &= \{t\}
\end{aligned}
$$

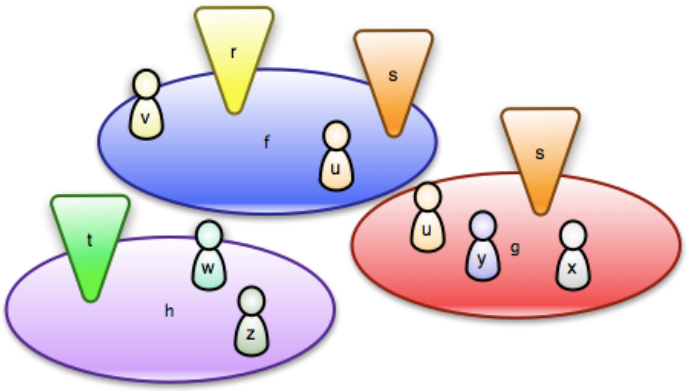

Figure 6.2.: The figure illustrates the Tip assignments to Families of the scenario. The accessibility of a Tip is—analogue to the accessibility of a User—controlled by a Family assignments. Tips can be assigned to more than one Family and are accordingly only accessible by the union of the members of these Families.

Note that Tips may be assigned to more than a single Family (e.g., s is assigned to $\{f, g\}$) which makes them accessible for the union of the members of the affected Families. The function *tips* (cf. Equation 6.19) returns the Tips that are accessible

to the Users in U via their Family memberships:

$$\begin{aligned} tips(u) = assignments(f) \cup assignments(g) &= \{r, s\} \\ tips(v) = assignments(f) &= \{r, s\} \\ tips(x) = tips(y) = assignments(g) &= \{s\} \\ tips(w) = tips(z) = assignments(h) &= \{t\} \end{aligned}$$

In the case of Tip s, which was assigned to Families $\{f, g\}$, the Users who are only member of only one of those Families, i.e.

$$(members(f) \cup members(g)) \setminus u = \{v, x, y\}$$

should be provided with a hint that the Tip s is assigned to a further, unaccessible Family and that their activities on that Tip are thereby exposed to it. They do not see any details of the other Family.

Figure 6.3 illustrates the scenario from another view-point to make obvious the disjunction of the two micro-communities that have emerged through the scenario. The edges between Users and Families are *memberships*, edges between Users are *friendships*, and edges between Tips and Families are *assignments*.

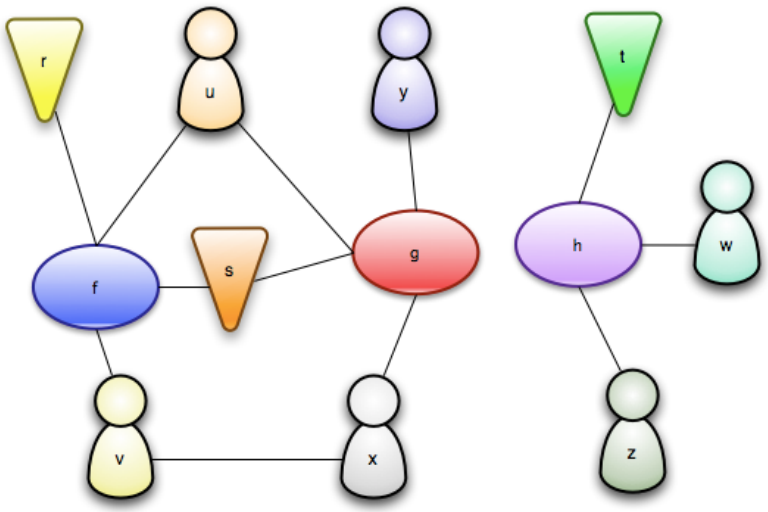

Figure 6.3.: The scenario from a view-point that reveals two resulting, disjoint micro-communities.

6.3. Application Requirements

As mentioned before, the SNS described in this chapter concentrates on fulfilling the privacy-related requirements of Section 5.1 better than the SNSs evaluated.

However, these requirements ensure the preservation and awareness of privacy but do not demand any particular features to define the purpose of a platform.

This section defines a list of application-specific requirements (AR) to define the platform's application-specific attributes and functionalities demanded by the concept and scenario.

AR1: Tips. Provide a Tip as a means for users to recommend locations to friends, and for communication.

AR2: User management. Provide functionality for user management including session and account management.

AR3: Families. Provide Families as groups. Users can join Families on an invitation, and Families can be equipped with Tips for content sharing. Family content and members can be accessed exclusively by members.

AR4: Information flow. To optimize the attention gained from users, maximize information flow to motivate user contributions and application visits.

AR5: Ergonomics. Provide software dialog ergonomics as defined by ISO 9241 part 110 (Section 2.3.4).

6.4. Features

This section defines a set of features (F). The features are aimed at fulfilling the privacy-related requirements (R) defined in Section 5.1 and the application-specific requirements (AR) of Section 6.3.

6.4.1. Basics

The following features are basics which describe principles of entity accessibility, default restrictiveness, and groups as a tool for information management.

F1: Binary entity accessibility. In order to create an application that is both privacy-preserving and intuitive to use, it is necessary to define rules that are easy to understand, but not confusing by having exceptions or by being not clearly communicated. Often, SNS profile pages of members are accessible to everyone. Accessibility is only limited by the level of detail of information. That is, on one hand, reducing the learning effort for new users of the SNS, because it works similar to SNSs they already know. On the other hand, it is confusing, because there is no black and white, i.e. a particular profile page is accessible or not, but some gray tone in between. That is, some profile information is accessible, because the reader is in a friend-of-a-friend relation to the profile owner, but not the whole profile page, because the reader is no direct friend. The nuances of these gray tones are different on every SNS which makes the rules of accessibility hard to understand and learn. This feature makes entities accessible or not. It goes so far as to not even making the name of the entity visible if accessibility is not granted. There are no partially accessible entities or exceptions. The rules are thus easy to understand and to comprehend.

This concept is fundamental for transparency requirements R1 and R3, and increases the *conformity with user expectations* (AR5, i.e. ISO 9241 part 110, cf. Section 2.3.4).

F2: Family as central grouping tool. This feature is aimed at providing the users with a tool to mirror their Social Contexts online by a group feature called Family (R8). Users are able to maintain Social Contexts and Contextual Integrity in their online lives through Families which they create, read, update, delete (CRUD), invite friends to, and communicate within. Family members have access to all Tips and User profiles within, whereas non-members have access to no information at all, not even the group's name (F1, F3). The feature addresses requirements R1, R2 and AR3.

F3: Restrictiveness by default. Instead of most other SNSs which are permissive by default, the application is restrictive by default regarding the accessibility of entities and information to protect privacy and to provide a more intuitive access control. The restrictiveness ensures users that their Tips are only accessible by the members of the Families the Tip was assigned to. This paves the way for a Communication Context of high intimacy and quality. This feature is targeted at forming a basis for answering requirement R1, but also affects R12 by limiting access per default.

6.4.2. The Main Task

The application-specific main task that should be performed by users is required by application-specific requirements rather than privacy requirements.

F4: The task is to share Tips. As described before, the Tip is the central, location-based entity to be dealt with (AR1). Tips are created, shared with friends, and provide a platform for communication with respect to all Communication Contexts. They also feature Social Contexts since Tips must be assigned to Families (F2) and are thereby only accessible to Family members (F1, F3, AR3). Although the application is generally easily extendable to further entities that are assigned to Families, for now and for the sake of simplicity and *suitability for the task* (AR5), Tips are the only entity for now.

6.4.3. Transparency

The following features is aimed at providing the needed transparency to users.

F5: Families provide transparency. To ensure transparency to users (R1), a displayed Family always lists its members to the viewing user. This ensures that the user is permanently made aware of the audience of information he or she is about to create, e.g. Tips, comments or users to invite to a Family.

F6: Tips provide transparency. To feature permanent transparency and feedback regarding the audience of information (content created on the Tip as a communication platform) that are created (R1), a Tip lists the users that have access to it and the Families it was assigned to. This ensures that the user

is permanently aware of the receivers of comments, contributions, photos or ratings assigned to a particular Tip.

6.4.4. Access Control

This section lists features to empower users with tools to control access to information.

F7: Control of Tip accessibility. When creating a new Tip, the user is provided with control over its audience by being able to select the Families for it to be assigned to (R2, R5). The families listed display their number of members and are clickable for an in-detail look of the particular members. The expansion of accessibility can be achieved by assigning the Tip to further Families later or by inviting further users to present Families.

F8: Revocation of Tip accessibility. The accessibility of a Tip can be revoked by the author by having it de-assigned from a Family by the Family creator. This supports post-creation control over accessibility and an oblivion feature (R2, R5).

F9: Access control via Family membership. As stated in F2, Families are an instrument to control the accessibility of user profiles and Tips. Members of a family are granted access to all members' user profiles and Tips assigned to the Family. The adding of Tips and inviting of further members can be controlled by the Family creator. The creator can also define whether members can do so, too. The removal of members and Tips from a Family can only be controlled by its creator. The feature further refines F2 and provides a solution for requirements R2 and R8.

F10: Access control via friendships. As required by R2 and R7, an SNS should provide qualifiable friendship links to control Social Context. This feature allows users to befriend other users and to qualify the friendship by defining XFN attributes. From the concept's point of view (cf. Section 6.1), friendships play a minor role in access control management since the access control to user profiles is mainly covered by F9. Nevertheless, this feature grants mutual access to the user profile of the friends. A given friendship is displayed to the viewer, e.g. when looking at a list of users.

F11: Account deletion. As an answer to requirement R10 and part of AR2, a user is able to delete her account. All created Tips are hereby deleted and thus, not accessible anymore, including all information added to the Tip, e.g. photos and comments. Created Families are not deleted, but continue to exist.

F12: User profile. A user can edit his user profile to control the information that is communicated to others about him. That includes a profile image, his name, location, a description about himself, and the management of comments left by others. On his user profile, the user transparently sees his Family memberships and the users that have access to his profile (as a consequence of his memberships), and accessible Tips. To prohibit the accessibility of his profile to other users, the user can choose to release all relations to that user

through shared Family memberships (F9) which are displayed when addressing another user's profile and a friendship (F10).

6.4.5. Infrastructure

Infrastructural features are demanded by both requirements lists, privacy requirements and application requirements.

F15: Invitation mechanisms. As stated above, unaccessible Families and Tips are kept secret and can not be browsed through the privacy-preserving concept. In order to gain access to user profiles and Tips, users must be invited to Families by members (AR3). This implies that it does not make sense to register to the service without having at least one Family invitation: The user would find an unappealing empty site, the so-called Blank Slate State [1]. This state must be avoided, because it is un-intuitive and demands a very high involvement of creating Social Contexts from scratch, instead of joining that of friends (which are congruent or at least very similar). Thus, the service must be invite-only. Users initiate a service invitation indirectly by creating a Family invitation: Users can invite friends to a Family regardless of whether they are already registered with the service. By entering email addresses or user names, the application looks up if a service membership already exists and, if not, invites the user not only to the given Families, but also to the service itself (F16). Accordingly, newly registered users always find at least one Family membership containing friends and highly relevant Tips and friend conversations.

Furthermore, the invitation mechanism increases the information flow since content is made accessible to further users (AR4).

F16: Authentication, Authorization, and Accounting (AAA). According to F1 and F3, there is no publicly accessible content. Accordingly, every user must have an account and authenticate himself or herself with the application in order to enter (AR2). That requires a registration feature (that users are led to via F15) to create new user accounts, a session management and cookie management system to serve logged-in users.

F17: Internal messaging is a feature for users to exchange messages similar to emails. The messages have a sender, a receiver, a body and a subject and can be created by users and the application itself. The application can send messages to notify users about important happenings, e.g. an invitation to a Family (AR3, AR4).

F18: Notification. As a way to lessen the problem of Participation Inequality [111], notification mechanisms are needed to motivate reactions of users (AR4). Notifications are spread on different channels by the applications if a user performs an action of significant interest for other users. The channels include email to entice inactive users back to the application, internal messages (F17) to lead them to the according entity that the action was performed on and the application-internal activity stream that lists notifications (F19).

F19: Activity stream. Tips, Families and the top-level view of the application provide its users with an individual and context-related stream of notifications of accessible users performing interesting actions on accessible entities (AR4). These notifications contain clickable links leading to the related entity.

F20: Game mechanics. According to Section 2.2.3 and to increase the percentage of heavy and intermittent contributors of Nielson's Participation Inequality [111], the application features Game Mechanics to motivate the activity of its users (AR4). For now, this consists of a simple rewarding function for gaining points for activity.

F21: Cross-site request forgery. To provide protection against cross-site request forgery (XSRF)[1], the application only accepts requests that carry a token that was delivered to the user with the successive response. This protects the user from invoking unintended requests that an attacker planted on him.

F22: Filtering and sorting preferences. The application provides filtering and sorting mechanisms for entity lists that are adjustable and persistent. A user is able to sort and filter the main entities Users, Tips and Families in different ways to ease access (AR5).

F23: User tracking. The application performs basic user interaction tracking by simple entity impression counting. Third party tools that expose user privacy to outside the application boundaries are not used, e.g. Google Analytics. The numbers can be used in a future version for, e.g. entity popularity ranking or automated recommendations (AR1).

A detailed description of how the approach by its functional and non-functional features fulfills the requirement to be ergonomic (AR5) is given in Section 6.7.5.

6.4.6. Disregarded Requirements

The creation of an approach that fulfills all privacy requirements would have exceeded the focus, which is to empower users with transparency and control features to protect their privacies. Consequently, requirements that contribute too little to this focus were disregarded. Namely, these are:

Panoptic Provider (R4). To address this requirement a distributed architectural approach has to be taken to prohibit a single instance to access all data. However, since the most relevant revenue stream of many commercial SNSs is advertisement (cf. Section 5.2), providers is not helped by limiting ways to access customer data. Therefore, we did not choose to focus on this type of approach. Other researchers, who concentrated on data distribution and encryption, provided promising approaches in this area (cf. Section 4.2).

DHPs (R5). As the P3P project has shown, controlling privacy through Data Handling Policies is possible, but comes with its weaknesses (cf. Sections 4.3 and 4.5). However, this approach is dedicated to address the controlling of user privacy on a more fine-granular level of adjustment.

[1] `http://en.wikipedia.org/wiki/Cross-site_request_forgery`, accessed October 2009

Data export (R9). Although there are standards emerging or existent for both data export and synchronization (cf. Section 4.7), this requirement is a future topic of the emerging Social Web (cf. Section 9.1) and thus, out of the scope of this approach.

Data ownership (R14). For empowering users with transparency and control to protect their privacies, data ownership of secondary importance. Thus, this requirement has been disregarded for the approach to be discussed.

National laws (R15). For the realization of an usable tool to make user privacies controllable, a functional differentiation according to particular national laws plays a minor role and has thus been disregarded for our approach.

6.4.7. Tabular Mapping

Table 6.1 provides an overview of the mapping of features to privacy requirements defined in Section 5.1. The mapping of features to application requirements is shown in Table 6.2.

6.5. Entities and Relations

According to the concept (Section 6.1) and the feature list (Section 6.4), especially Feature F4, the main entity to share is a Tip. Therefore, it gets assigned to Families by its creator in order to share it with friends. Families maintain memberships of Users and thereby build a bridge for Users in order to access Tips. Figure 6.4 illustrates a entity-relationship (ER) diagram of these basic, access-controlling relations between the primary entities, i.e. Families, Tips and Users.

Figure 6.4.: Entity-Relationship diagram of the basic, access-controlling relation between the main entities User, Tip and Family. In contrast to common practice, only an existing direct or indirect relation between the instances grants access to it. Access control for a Tip instance to a particular User is implied by a relation of both instances to at least one common Family.

According to Feature F2, Families are a central grouping tool that stands in a relation to both, Tips and Users. Figure 6.4 depicts that a User can have multiple Family memberships and that a Family can consist of many Users. Also, a Tip can be assigned to more than one Family, and a Family can host multiple Tips.

Figure 6.5 shows that besides the named primary entities User, Family and Tip, there are numerous secondary as well as adherent entities to augment the primary

		R1 Transparency	R2 Access control	R3 Unauthorized access	R4 Panoptic Provider	R5 DHPs	R6 User model control	R7 Friendships	R8 Groups	R9 Data export	R10 Account deletion	R11 Purpose limitation	R12 Data security	R13 Minimize data collection	R14 Data ownership	R15 National laws
Entity access	F1	√		√												
Families	F2	√	√				√		√							
Restrictiveness	F3	√										√				
Main task	F4															
Family transparency	F5	√														
Tips transparency	F6	√														
Control Tip access	F7		√													
Revoke Tip access	F8		√													
Family access control	F9		√						√							
Friends access control	F10		√				√	√								
Account deletion	F11		√								√					
User profile	F12	√					√							√		
Invitations	F15		√													
AAA	F16												√			
Internal messaging	F17															
Notifications	F18															
Activity stream	F19															
Game mechanics	F20															
XSRF	F21			√									√			
Filtering and sorting	F22	√														
User tracking	F23													√		
Disregarded					X	X				X					X	X

Table 6.1.: Mapping of features and privacy requirements (R; cf. Section 5.1).

		AR1 Tips	AR2 User management	AR3 Families	AR4 Information flow	AR5 Ergonomics
Entity access	F1					√
Families	F2			√		
Restrictiveness	F3			√		√
Main task	F4	√		√		√
Family transparency	F5					√
Tips transparency	F6					√
Control Tip access	F7					√
Revoke Tip access	F8					√
Family access control	F9					√
Friends access control	F10					√
Account deletion	F11		√			√
User profile	F12					√
Invitations	F15			√	√	
AAA	F16		√			
Internal messaging	F17			√	√	
Notifications	F18				√	
Activity stream	F19				√	
Game mechanics	F20				√	
XSRF	F21					
Filtering and sorting	F22					√
User tracking	F23	√				

Table 6.2.: Mapping of features and application requirements (AR; cf. Section 6.3).

ones. In the following, these entities are introduced in brief. Additionally, Users have numerous, sometimes multiple relations to other entities (cf. the left hand side relations of the figure except the top row). These are relations to entities as creators.

In order to fulfill infrastructural features, e.g. to act as a communicational basis, the primary entities feature a number of adherent entities (cf. most of the right hand side relations in Figure 6.5 except the top row).

Tip. A Tip is created by a User and assigned to Families (cf. e.g. Feature F4). It provides a platform for Communication Context in a Social Context for users. Therefore, it features the auxiliary entities Contribution, which provides room for textual and multi-medial contributions from its users, and Comment, which allows users to enter small pieces of text that appear similar to a dialog. To accommodate further images, a Tip features the entity Asset that allows users to upload and share pictures on a Tip. A Tip's author can additionally define an Avatar image for it. As described before, a Tip furthermore features a Location for its geographic coordinates and address.

Family. Besides their relations to Users and Tips and the implicit importance as an access control basis, Families allow minimal Communication Context in a given Social Context through Comments. They also allow its creator to upload and edit an Avatar image that is displayed in different views.

User. According to Feature F12, every User features his or her own profile page to express themselves. Here the User's address and location is displayed which is provided through the Location entity. As Families and Tips, Users have Comments allowing dialogs and greetings from other users on profile pages. The User chooses an Avatar image to upload and to be displayed on his or her profile page.

Besides the primary and their adherent entities, there is a number of secondary entities:

Invitation. According to Feature F15, the strict access control design requires sophisticated invitation mechanisms to invite friends to the application and to Families. The entity Invitations maintains application invitations that are transformed into new Users when converted. Family invitations for existent Users are realized through the relation table between Users and Families.

Message. To realize Features F17 and F18, the Message entity models bilateral and email-like communication between Users. Some notification types of F18, e.g. Family invitations, are communicated as Messages sent by the application itself.

Friendship. As the basis for Feature F10, the Friendship entity carries friendships between Users.

Bulletin. Feature F18 requires a Bulletin entity for maintaining application-internal, contextual bulletins. These bulletins are created about user activity, delivered to users standing in Social Context, and contain a message text and a link to the related entity.

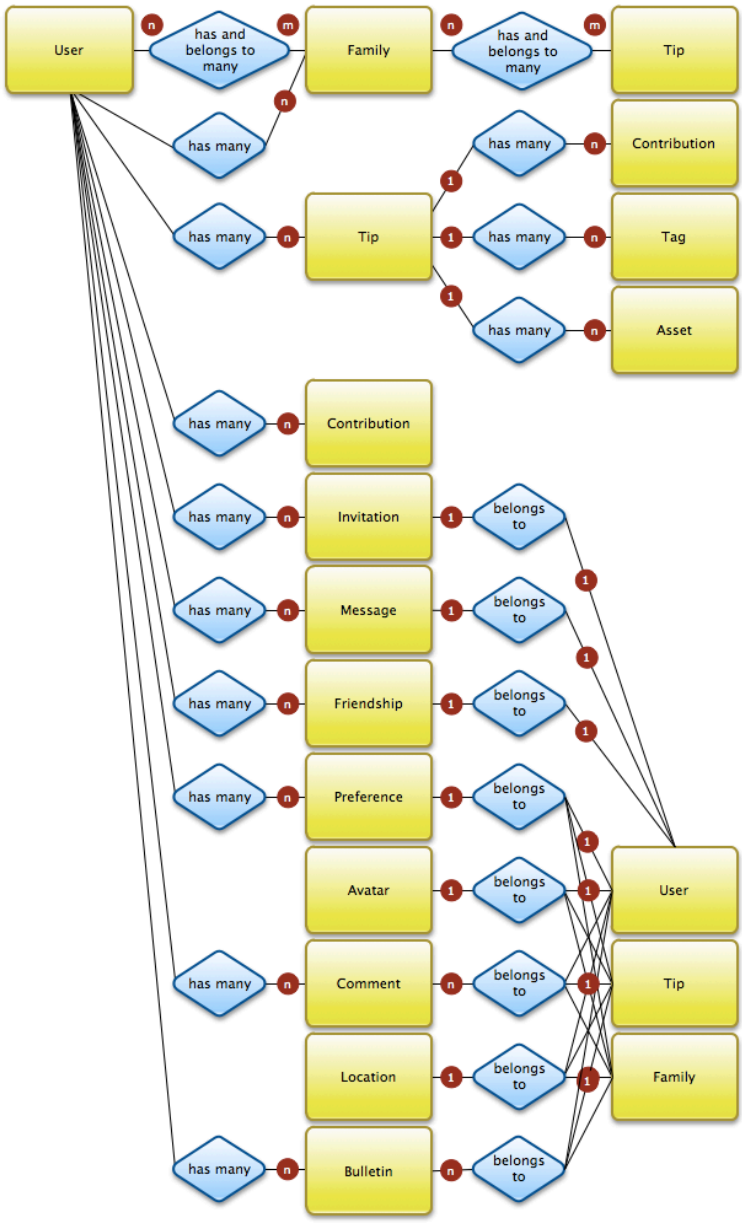

Figure 6.5.: Entity-Relationship diagram of the most important primary, secondary and adherent entities. Some adherent entities feature Single Table Inheritance so that they can be assigned to entities of different types. The entity Preference is a triple-store that can store name-value pairs as predicates for both, arbitrary-typed subjects and objects.

Preference. The Preference entity is a special case. It is modeled as a triple-store to manage preferences as name-value pairs for arbitrary-typed subjects regarding an arbitrarily-typed object. Through its generality, it often omits future migrations to data models on addition or removal of attributes.

6.6. Framework

As a functional and architectural foundation for the named features, entities and their relations, a framework was created that is described in this section. The framework was designed as a foundation for providing general privacy-preserving core functionality for privacy-preserving applications that users interact and work with. An application to provide the described concrete use-case is presented in the upcoming Section 6.7. The separation was introduced to preserve the option to re-use the framework for a different application.

The framework is designed to act behind an application that is served through a Web application server. Consequently, user interaction with an application is thus exclusively driven by requests (*pull* only), since the HTTP is stateless and request-driven. Every reaction from the application is based on an action (request) from a user. Architecturally, the framework is based on the Model View Controller (MVC) programming pattern (Section 2.3.3).

6.6.1. Business Logic

Concerning the proposed concept, there are a number of required modules that contain business logic functionality and do not fit into the MVC pattern's layers. Figure 6.6 illustrates the modules and their functionality.

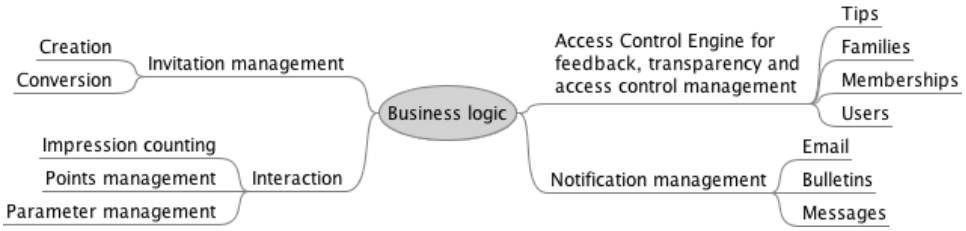

Figure 6.6.: The figure gives an overview of functionality modules of the business logic block. The modules are invitation management, Access Control Engine, notification management, and interaction.

The above defined features require several functional modules related to business logic:

Invitation management. The basic Features F1-F3 do not work without an invitation management (F15) for the creation and conversion of both, user invitations to Families and consequently, user invitations to the application (if a user invited to a Family by an email address that is not yet registered with the application). The invitations work flow is illustrated in Figure 6.7 which

gives an impression of the complexity of the business logic involved in the invitation process.

The module furthermore provides Family and application invitation limitation functionality, which is described in a later section.

Access Control Engine. The Features F5-F12 for feedback, transparency and access control require numerous computationally complex access-checking and context-related entity instantiation functions: for many heterogenous situations (cf. the named features) an Access Control Engine (ACE; in detail described in the forthcoming Section 6.6.2) must check read and write access to different kinds of entities in different contexts, and maintain the instantiation of entity lists depending on the requestor.

Interaction. The interaction module provides functionality for improving and tracking user interaction. The three submodules are firstly *impression counting* to count entity impressions by particular users (F23). Secondly, a points management for computing and updating the User's points for content contribution (F20). Thirdly, a parameter management submodule to provide filtering and sorting persistency for user preferences regarding all main entities (F22).

Notification management. Notifying users about another user's activity belongs only peripherally to the request-response-cycle depicted in Figure 2.4, since it does not affect the user that is acting. It is literally a different thread that is started, but must not be waited for in order to answer the current request. Sending emails, Bulletins and Messages (F17-F19) is strongly context dependent: the selection of the recipients utilizes functionality provided by the ACE.

We propose the introduction of an additional layer for business logics to provide space for the named functionality modules. Figure 6.8 illustrates how the layer is associated to the existent ones. The business logic layer is solely consulted by the controller layer in order to process the current request. In cases where entities are requested to be read or written, the business logic layer accesses the model layer to perform CRUD operations.

The following flow charts underline the necessity, functionality and the consulting of the business logic layer on selected exemplary user requests to the framework presented. As described in Section 2.3.3, the controller layer defines the application behavior. Therefore, the steps in general take place within the controller layer. Activities of other layers are explicitly highlighted.

Figure 6.9 depicts a detailed flow-chart of a request-response cycle of a user requesting to view a particular Tip. General access control is performed in the controller layer via a check of the session. Requestors without a Session are redirected to the login page. If a proper Session is present, the request is delegated to the suitable controller which consults the specific access control mechanism, i.e. the ACE located in the business logic layer, to check if the given user has the rights to access the requested Tip. If so, the controller instantiates the Tip through the model layer and lets it be rendered by the view layer.

Figure 6.10 presents a flow-chart for a request to list all Tips from the perspective of the requesting user. The ACE therefore constructs a query that takes into

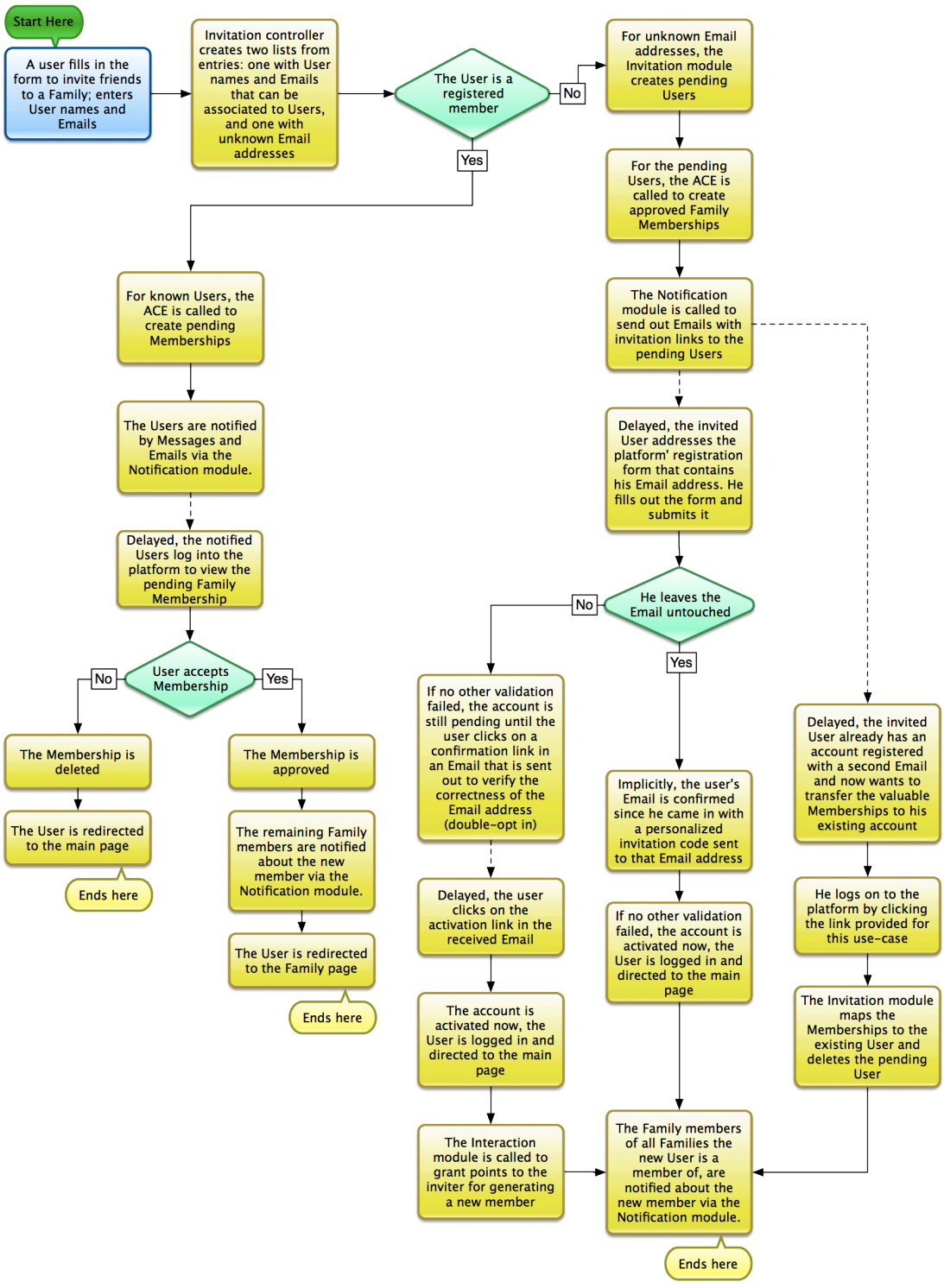

Figure 6.7.: The figure illustrates the complexity of the invitation process of the framework. The flow chart covers all possibilities and circumstances of an invitation triggered by a user.

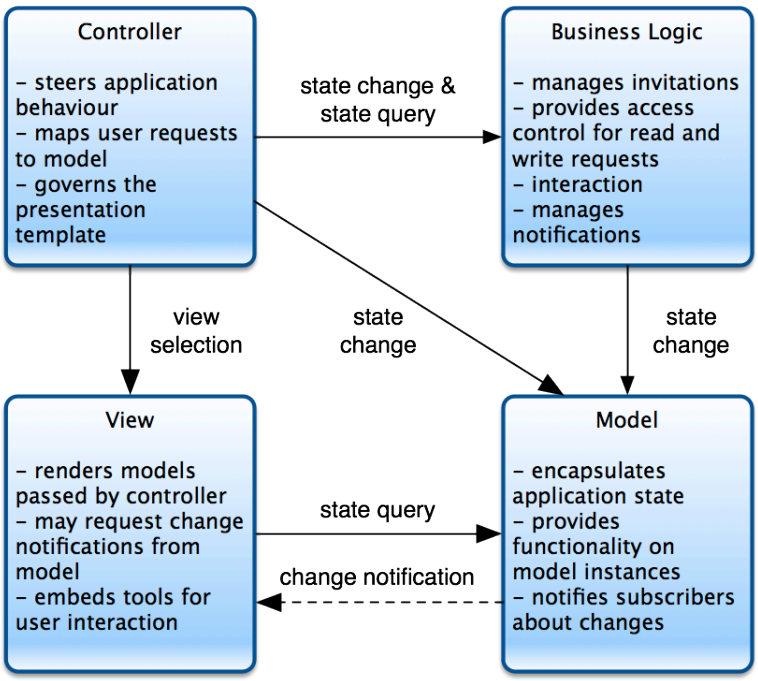

Figure 6.8.: The proposed business logic layer provides complex functionality for the controller layer to, e.g., check accessibility (read or write) of entities, instantiate entity lists regarding a requesting user, or to perform notification operations regarding particular user activity.

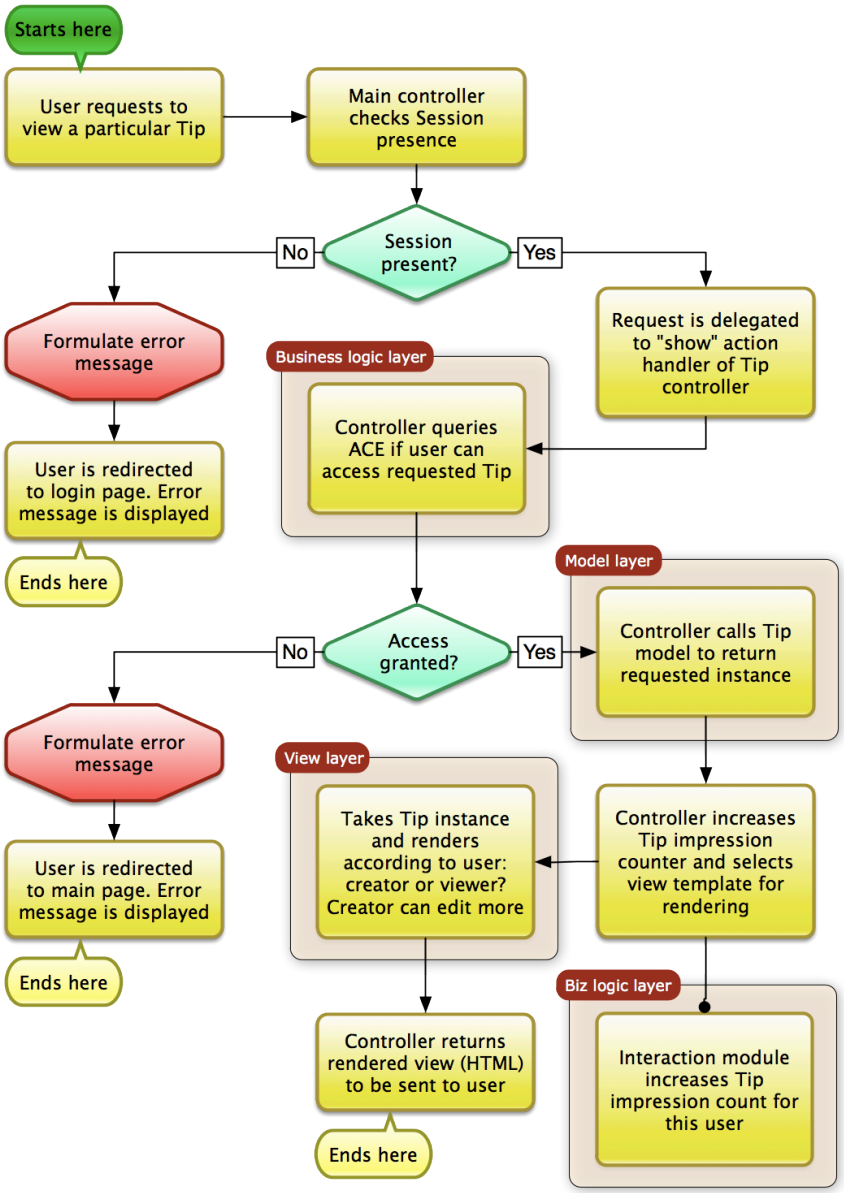

Figure 6.9.: A flow-chart of a request-response cycle of a user requesting to view a particular Tip. General and specific access controls are located in the controller and the business logic layer, respectively. General access control performs a session check and specific access control provides entity read and write accessibility checks through the ACE.

account the User's Family memberships, Tip creations, and his or her stored preferences for filtering and sorting Tips, which are managed by the interaction module. For the sake of brevity, the Session check is now preconditioned. Note that in this case, the business logic layer's ACE instantiates the Tips list directly through the model layer and returns it to the controller which renders and returns it.

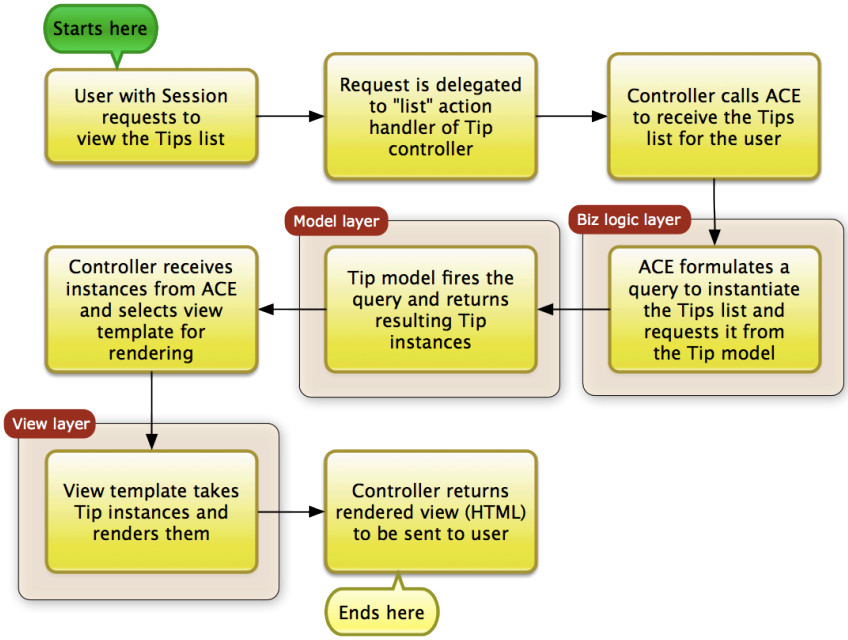

Figure 6.10.: A flow-chart of a request-response cycle of a User requesting to list Tips. The ACE, which constructs the query, takes into account not only the User's accessibility to Tips but also his or her stored preferences concerning Tip filtering and sorting which is managed by the interaction module. Session presence is preconditioned. The list of Tips is instantiated by the business logic layer through the model layer.

A similar flow can be described by a user request for a form in order to create a new Tip (Figure 6.11). The special case about this request is that, according to the concept, the user must be provided with a list of Families that the Tip can be assigned to. To instantiate this list for later embedding into the form, the business logic layer acts analogous to the described request for a Tip list.

Figure 6.12 illustrates the flow-chart for a user sending a request that contains a filled form for creating a new Tip (cf. Figure 6.11). In contrast to common Web application which only validate the attributes of the model and adherent entities, the concept forces a validation of the entities that the Tip is requested to be assigned to. For security reasons, it is necessary for the ACE to repeat the request for Families that the user is allowed to assign Tips to protect the application from request forgery. If the requested assignment is valid, the controller lets the model

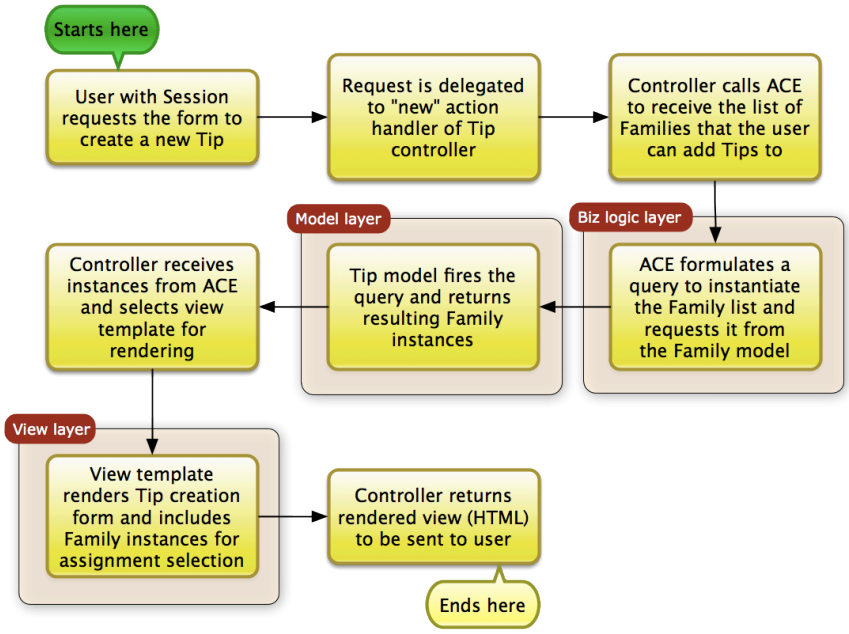

Figure 6.11.: A flow-chart of a request-response cycle of a user requesting a form to create a new Tip. Very similar to flow chart 6.10, business logic layer instantiates the list of Families through the model layer.

layer validate the attributes, and create the Tip in the positive case. Subsequently, the controller calls the business logic layer's ACE to assign the created Tip to the requested Families. The members of the Families are notified about the activity via Bulletins and emails.

6.6.2. Access Control Engine

In order to provide a solution for the basic features F1-F3, the foundation for the feedback features F5 and F6 and the access control features F7-F10, an Access Control Engine (ACE) is needed to provide sophisticated access control functionality. The ACE is furthermore needed by the business logic modules for invitation and notification management. The basic functions of the ACE are formalized in the following.

Formal Definition

Let the three main entities Families, Tips and Users be represented by the sets F, T and U, respectively.

$$F = \{f_1 \dots f_n\} \tag{6.1}$$
$$T = \{t_1 \dots t_n\} \tag{6.2}$$
$$U = \{u_1 \dots u_n\} \tag{6.3}$$

Let a correspondence M contain membership definitions for Users being members of Families, and a function $members(f)$ to return the members for a Family be defined as

$$M \subseteq F \times U \tag{6.4}$$
$$members : F \to U \tag{6.5}$$
$$members(f) := \{u | f \in F \wedge u \in U \wedge (f,u) \in M\} \tag{6.6}$$

Reversely, let a function $families(u)$ return the Families that are accessible to a given User through their memberships M be defined as

$$M \subseteq F \times U \tag{6.7}$$
$$families : U \to F \tag{6.8}$$
$$families(u) := \{f | u \in U \wedge f \in F \wedge (f,u) \in M\} \tag{6.9}$$

Furthermore, let a correspondence B contain friendship definitions and a function $friends(u)$ return the friends of a given User.

$$B \subseteq U \times U \tag{6.10}$$
$$friends : U \to U \tag{6.11}$$
$$friends(u) := \{v | u \in U \wedge v \in U \wedge ((u,v) \in B \vee (v,u) \in B)\} \tag{6.12}$$

Let the function $users(u)$ define User profiles that are accessible to a given Users. A User is granted access to all User profiles that he shares a Family membership

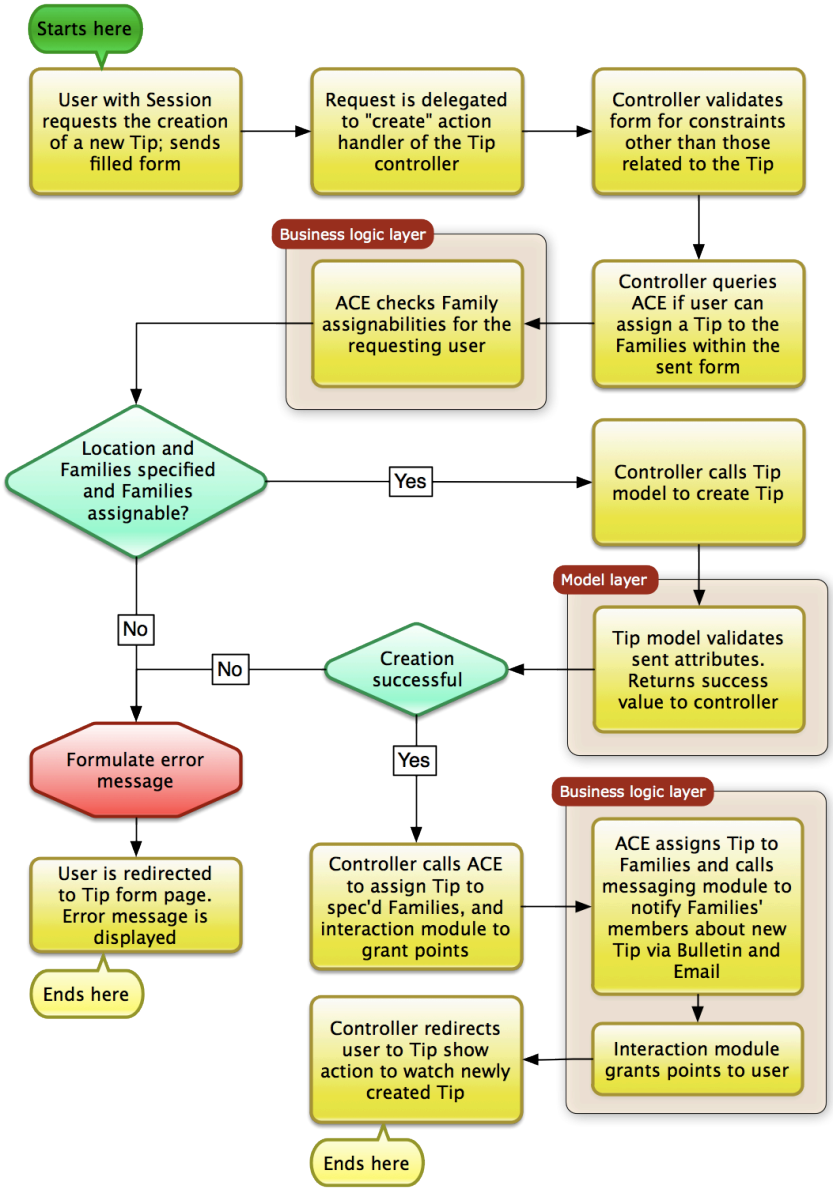

Figure 6.12.: A flow-chart of a request-response cycle of a user sending a filled form to create a new Tip through a successive request after having requested the form (Figure 6.11). The business logic layer is called in two cases: firstly to check the Tip-assignability of the Families sent in the request to avoid request forgery, and secondly to assign the validated and created Tip to those Families followed by a notification of the members.

with or that he is friends with.

$$users : U \to U \tag{6.13}$$

$$users(u) := \{v | u \in U \land v \in U \land f \in F \land (u \in members(f) \land v \in members(f)) \lor v \in friends(u)\} \tag{6.14}$$

Let there be a correspondence A for assignments of Tips to Families. The function $assignments(f)$ returns the set of Tips for a given Family.

$$A \subseteq T \times F \tag{6.15}$$

$$assignments : F \to T \tag{6.16}$$

$$assignments(f) := \{t | f \in F \land t \in T \land (f, t) \in A\} \tag{6.17}$$

Let the function $tips(u)$ return Tips that are accessible to a particular User. According to Figure 6.4, there is no direct relation between the entities of Tips and Users. The access is granted if the User is member of at least one Family that the Tip is assigned to.

$$tips : U \to T \tag{6.18}$$

$$tips(u) := \{t | u \in U \land f \in F \land t \in T \land (u \in members(f) \land t \in assignments(f)\} \tag{6.19}$$

Filtering and Sorting

The listing functions for main entities, i.e. $families$, $users$ and $tips$, provide filtering and sorting options for use in different contexts and better accessibility.

Let the functions $families\_by(u)$ and $tips\_by(u)$ return the created Families and authored Tips, respectively, for a given User. Therefore, let the correspondence F_c contain Family creations by Users, and T_c contain Tip creations by Users.

$$F_c \subseteq U \times F \tag{6.20}$$

$$families\_by(u) : U \to F \tag{6.21}$$

$$families\_by(u) := \{f | u \in U \land f \in F \land (u, f) \in F_c\} \tag{6.22}$$

$$T_c \subseteq U \times T \tag{6.23}$$

$$tips\_by(u) : U \to T \tag{6.24}$$

$$tips\_by(u) := \{t | u \in U \land t \in T \land (u, t) \in T_c\} \tag{6.25}$$

Let F_a be a subset of F, T_a be a subset of T and U_a be a subset of U to contain Families, Tips, and Users, respectively, that feature an Avatar image. Accordingly, let the functions $avatared\_families()$, $avatared\_tips()$, and $avatared\_users()$ return Families, Tips, and Users contained in these sets.

$$F_a \subseteq F \tag{6.26}$$

$$avatared\_families() := \{f | f \in F_a\} \tag{6.27}$$

$$T_a \subseteq T \tag{6.28}$$

$$avatared\_tips() := \{t | t \in T_a\} \tag{6.29}$$

$$U_a \subseteq U \tag{6.30}$$

$$avatared\_users() := \{u | u \in U_a\} \tag{6.31}$$

The $families$ function can be called to sort the resulting instances alphanumerically or by creation date. It can filter accessible Families by

- Those a given Tip was assigned to (Equation 6.32). This filter is needed when displaying the Families list of a Tip.
- Those mutually accessible together with a given User (Equation 6.33). This feature is provided when a user visits another user's profile page. This function defines the Families list to display.
- Those created by the requester or another given User (Equation 6.34). This filter can be selected when visiting another user's profile page to display its created Families.
- Those that fulfill the criterion to be created by friends (Equation 6.35). This feature is provided to filter the main Tips list.
- Those that provide an avatar image (Equation 6.36). This feature is provided to filter the main Tips list.

$$
\begin{aligned}
families'(u,t) &:= \{f | u \in U \wedge f \in F \wedge t \in T \wedge \\
&\quad f \in families(u) \wedge t \in assignments(f)\} && (6.32)\\
families''(u,v) &:= \{f | u \in U \wedge v \in U \wedge \\
&\quad f \in families(u) \wedge f \in families(v)\} && (6.33)\\
families'''(u,v) &:= \{f | u \in U \wedge v \in U \wedge \\
&\quad f \in families(u) \wedge f \in families\_by(v)\} && (6.34)\\
families^{(4)}(u) &:= \{f | u \in U \wedge v \in U \wedge f \in families(u) \wedge \\
&\quad v \in friends(u) \wedge f \in families\_by(v)\} && (6.35)\\
families^{(5)}(u) &:= \{f | u \in U \wedge f \in families(u) \wedge \\
&\quad f \in avatared\_families()\} && (6.36)
\end{aligned}
$$

The $users$ function can be instructed to order the Users alphanumerically, by registration date or by points earned. It can filter accessible Users by

- Other Users that have mutual access to a given Tip (Equation 6.37). This function is needed for listing users with access to a given Tip.
- Other Users that have mutual access to a given Family (Equation 6.38). This function is needed for listing users with access to a given Family.
- Friends (Equation 6.39). This feature is provided to filter the main Users list.
- Those with an Avatar image (Equation 6.40). This feature is provided to filter the main Users list.

$$
\begin{aligned}
users'(u,t) &:= \{v | u \in U \wedge v \in U \wedge t \in T \wedge \\
&\quad t \in tips(u) \wedge t \in tips(v)\} && (6.37)\\
users''(u,f) &:= \{v | u \in U \wedge v \in U \wedge f \in F \wedge \\
&\quad u \in members(f) \wedge v \in members(f)\} && (6.38)\\
users'''(u) &:= \{v | u \in U \wedge v \in friends(u)\} && (6.39)\\
users^{(4)}(u) &:= \{v | u \in U \wedge v \in users(u) \wedge \\
&\quad v \in avatared\_users()\} && (6.40)
\end{aligned}
$$

The *tips* function can be requested to return Tips sorted alphanumerically, by creation date or by event date. It can filter Tips to be[2]

- Mutually accessible to another User (Equation 6.41). This feature is provided when a user visits another user's profile page. This function defines the Tips list to display.
- Created by another given User (Equation 6.42). This filter can be selected when visiting another user's profile page to display its created Tips.
- Created by friends (Equation 6.43). This feature is provided to filter the main Tips list.
- Assigned to only a given Family (Equation 6.44). This filter is needed for displaying only accessible Tips that are assigned to a given Family which is currently displayed.
- Those with an Avatar image (Equation 6.45). This feature is provided to filter the main Tips list.

$$
\begin{aligned}
tips'(u,v) &:= \{t | u \in U \wedge v \in U \wedge t \in T \wedge \\
&\quad t \in tips(u) \wedge t \in tips(v)\} && (6.41)\\
tips''(u,v) &:= \{t | u \in U \wedge v \in U \wedge t \in T \wedge \\
&\quad t \in tips(u) \wedge t \in tips\_by(v)\} && (6.42)\\
tips'''(u) &:= \{t | u \in U \wedge v \in U \wedge t \in T \wedge \\
&\quad t \in tips(u) \wedge t \in tips\_by(v) \wedge v \in friends(u)\} && (6.43)\\
tips^{(4)}(u,f) &:= \{t | u \in U \wedge f \in F \wedge t \in T \wedge \\
&\quad t \in tips(u) \wedge t \in assignments(f)\} && (6.44)\\
tips^{(5)}(u) &:= \{t | u \in U \wedge t \in T \wedge \\
&\quad t \in tips(u) \wedge t \in avatared\_tips()\} && (6.45)
\end{aligned}
$$

[2]Tips can furthermore be filtered to fulfill the criterion to be events, upcoming events, past events, within a geographic radius around the requestor's location, but that is out of this thesis' scope.

6.7. Application and User Interface

This section describes an exemplary application built on top of the framework to make the concept and functionality applicable for users.

Building a good application is no trivial task. Basically, application-specific requirements and features have to be fulfilled and provided, respectively, in order to empower users to *effectively* fulfill the tasks and purpose it was made for. A challenge that this section is faced with is to improve the users' *efficiency*, which is realized by the application's usability and software ergonomics (AR5). This section describes the implementation of an application that is not only effective, but also efficient for users to work with.

To ensure usability and software ergonomics, the task of creating an application follows the dialog principles of software ergonomics of ISO 9241 part 110 (Section 2.3.4), as demanded by AR5. The upcoming parts of this section, esp. Section 6.7.5, will refer to the contained principles.

6.7.1. Formative Evaluation

The application's layout, navigation design and visual design have undergone formative evaluations for iterative, evolutionary improvements and optimization. They have been iterated, until the state described throughout this section was reached. Appendix C lists some early stage sketches and describes them in brief.

At a more mature stage, a qualitative laboratory user test was applied, asking ten probands to solve tasks on a functional prototype. The goal of the test was to evaluate the ergonomics of the application. The probands were confronted with a list of tasks, e.g. registering with the platform based on an invitation email, founding a group for maintaining a certain Social Context, and creating a Tip to share it with friends. The tasks were formulated in a solution-neutral format so that the proband had to understand the application semantically to solve it. The users were observed and their activities were documented. The insights derived from the observations were utilized to solve usability problems and to improve, e.g., explanatory texts and graphics.

The remaining section describes the final version of the application.

6.7.2. Information Architecture and Navigation Design

> We define information architecture as the art and science of organizing and labeling websites, intranets, online communities and software to support usability.
> (The Information Architecture Institute[3])

In the following, the steps concerning Information Architecture (IA) and Navigation Design are presented to improve the resulting application's usability [112].

Basically, information space is designed to be displayed on a page-basis. Thus, users navigate from a page that contains a list of entities (called *list-action*) to a page that renders a particular entity's details (called *show-action*). Though the latter will additionally contain links to peripheral, contextual entities, the main

[3] `http://iainstitute.org/`, accessed October 2009

content area (cf. Section 6.7.3) displays content that refers to a list of entities or content rendering entity details. These *show-actions* have similarities according to the shown entity. If a Family, a Tip or User is shown, the user has to understand that it is a *show-action* and what he or she can do now. *List-actions* are dealt with analogously. By doing this, the learning effort for the user to learn to use the application is lowered (*self-descriptiveness*, *conformity with user expectations*, ISO 9241 part 110, cf. Section 2.3.4).

Figure 6.13 illustrates a simple site topology for entities that need to be represented by an application. It illustrates the main entities and their relation which can only be navigated vertically (i.e. hierarchically down- or upwards). Although correct in principle with respect to the entity hierarchy, it requires a lot of interaction effort from the user to navigate from one Tip to another. E.g., after navigating all the way up to the list of Families, the user has to navigate down again to the Family that contains the Tip in question.

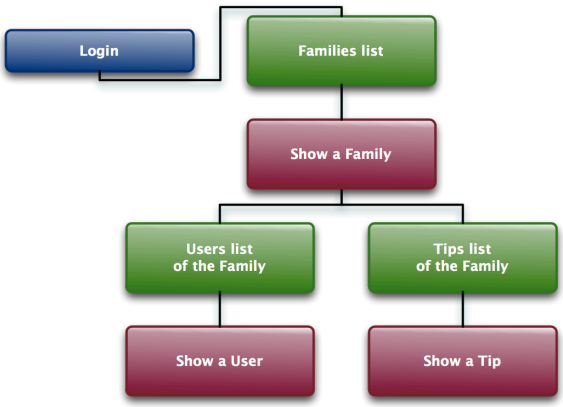

Figure 6.13.: Simplified page navigation that was derived strictly from the entity hierarchy. The navigation design provided is solely vertical. The topology implies considerable interaction efforts for the user by not providing horizontal links to other entities.

An approach to significantly improve this is the concept of equaled entities. If Families, Tips and Users would be presented to users as equaled entities (entities of the same hierarchical level), a set of additional opportunities and improvements becomes possible:

- Users can browse through their individual *global* list of Users (through the *users* function) and affect it through filtering and sorting.
- Users can browse through their individual *global* list of Tips (through the *tips* function) and affect it through filtering and sorting.

The original concept defined that when a user views a particular Family, he can be provided with the related lists of Tips and Users. With the concept of equaled entities, this can be taken further:

- When the user watches a particular Tip, the application can now present a list of related Users which are the unified accessors via Family memberships, and a list of Families the Tip is assigned to.

- A User addressing a User (him or herself or another User), can be provided with even more information regarding the other two entities of Tips and Families:
  - A list of Tips that the viewer and the viewed User have mutual access to.
  - A list of Tips that the viewed User created and that the viewer has access to.
  - A list of Families that the viewer and the viewed User have mutual access to.
  - A list of Families that the viewed User created and that the viewer has access to.

Figure 6.14 shows a significantly changed site topology after applying the concept of equaled entities to it. Firstly, it moves the lists of all main entities to the same level, which dramatically decreases the navigation depth of the application. This correlates with the interaction effort for navigating through the information space. Secondly, the concept includes lists of related entities directly into the template for showing an entity. The depicted topology also enables horizontal navigation. This is made possible through the presentation of entities related to the displayed one. In contrast to the topology shown in Figure 6.13, where a shown User is solely presented in the context of the Family the user navigated to, this topology allows the release of the context by performing a sidewards (horizontal) navigation, releasing the current filter. By doing this, the new entity is focused and presents its related entities. By breaking with the original concept's information hierarchy, this principle can be analogously applied to all entities to support interaction intuitiveness on the part of the user. With this IA, the user can navigate horizontally to literally observe the information space from ever-changing view points.

Figure 6.15 depicts the same site topology in a more feature-complete state[4]. The illustration has the following peculiarities: The green or second layer contains the pages that are reachable from the main menu and the header, both of which are omnipresent if the user is logged in (labeled *"from any inside page"*). The omnipresent availability of the pages is, for the purpose of this illustration, simplified in the form of the blue oval of the first layer. The second layer's pages can link to themselves by filtering or sorting in the cases of Users, Tips and Families, or in the case of Messages, switch between inbox and outbox.

Focussing on the arrows between the second and third layer, the main entities' list pages, i.e. Users, Tips and Families, in the first line provide links to their particular entities, but may also contain links to foreign main entities through their activity streams.

[4]The illustration of interaction elements was left out for the sake of simplicity and size of the figure.

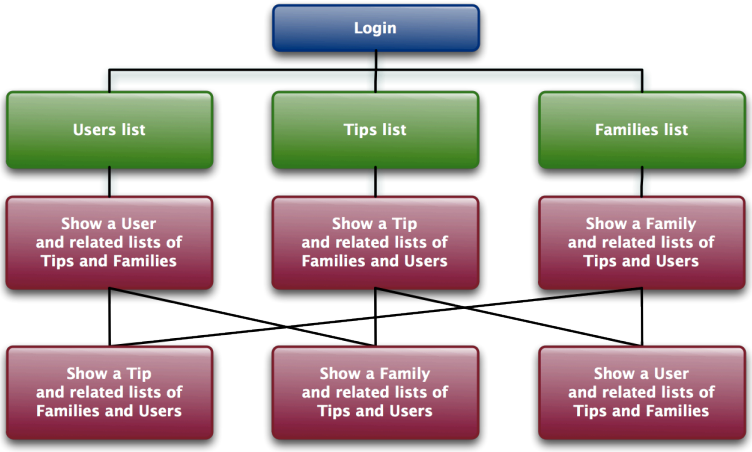

Figure 6.14.: The improved page navigation following the concept of equaled entities. The vertical navigation depth was reduced and horizontal navigation introduced. The reduction of navigation depth decreases the user-perceived complexity of an application by making the navigation space less time-consuming to explore.

The third or purple layer contains the show pages for the main entities. As explained above, an entity always shows a sub-menu that contains the lists of related entities of the related other entity types. The embedded links are those that interlink the pages of this layer: The pages show up a very strong linkage to enable horizontal navigation. Pages also link to themselves (not displayed) in case of on-page activities, such as commenting, contributing or rating.

The bottom (yellow) layer consists of pages containing forms for entity creation or user invitation. Incoming arrows stand for activity links embedded in other pages. Outgoing links lead the user to the target page in the case of a successful entity creation. In an unsuccessful case, the user remains on the form page.

The very shallow click-depth of the site of circa four clicks positively affects the understandability of the application because exploration time is dramatically minimized.

Information Architecture and Navigation Design are not only a tools for increasing the usability of an application by optimizing the way information and its relations are presented and navigated. A positive side effect is that a good IA and Navigation Design to a certain degree explains the structure and the core workings of an application. This decreases the learning effort the user has to put into a new application (*self-descriptiveness*, ISO 9241 part 110, cf. Section 2.3.4).

6.7.3. Layout and Grid

The application has to visualize much information in the form of content and navigation elements to provide the needed contextual transparency (R1, F5, F6) and interaction possibilities. Content-wise, entities have to be displayed and related con-

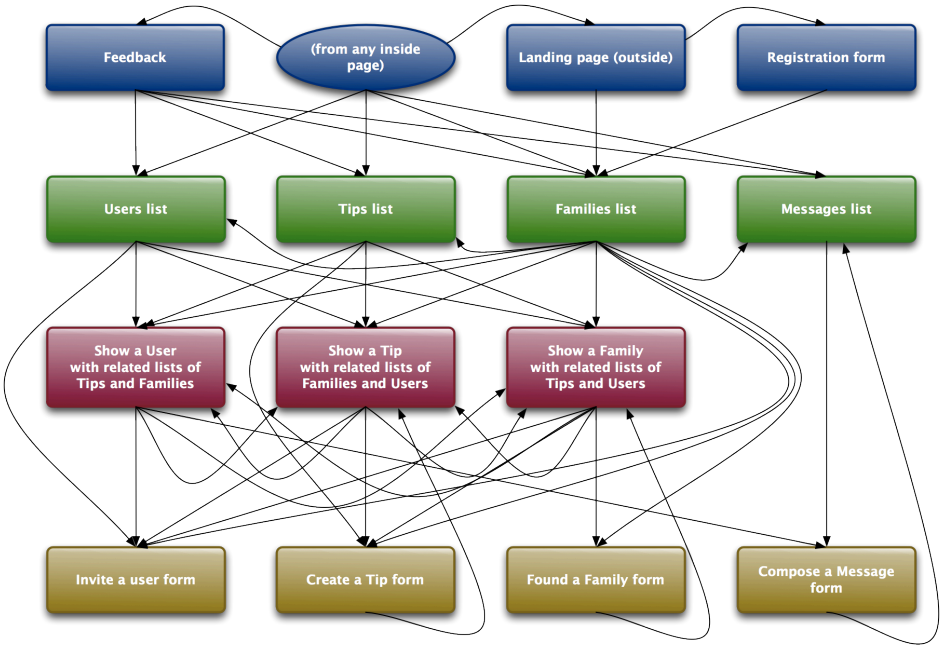

Figure 6.15.: A more page-complete view of the improved page navigation shown in Figure 6.14. The application is streamlined for accessibility by the very shallow and horizontal navigation by strong linkage of the pages. The interaction mainly takes place on 2nd and 3rd layer pages (green and purple layer). The yellow layer contains content creation pages and the blue layer infrastructural or outside pages.

tent has to be listed to provide the needed transparency. Additionally, control (R2, F7–F10) and activity features (F19) have to be made accessible and thus listed on screen. Navigation-wise, there are more links to entities planned than only through the main navigation: Entities will also be reachable through contextual navigation related to a displayed entity, a search function, and contextual activity streams, which directly link to recently changed entities through activities conducted by others.

In order to provide this space, the application dynamically scales to 100% of the available width and height of the user's browser window (usually, applications of similar type provide a fixed width of max. 1000 pixels and must be scrolled vertically). This way, no content is placed *below the fold* and thus hidden to the user when not scrolling[5] [112]. Figure 6.16 depicts the application's minimum size of 1000×578 pixels and its rasterization to 10×10 pixel blocks forming the grid for any element positioning and alignment.

Figure 6.16.: The application's layout is based on a 10x10 pixel grid. The minimum dimensions of the visualization that is scalable in both dimensions are 1000x578 pixels.

A three-column layout was applied to maximize the space needed for all information that have to be displayed. Figure 6.17 illustrates the layout of the application. The black arrow indicates the navigation path starting from primary navigation via secondary navigation to the content. This path correlates with the way users consume Website content (derived from the reading direction, i.e. left to right and top to bottom) [102].

[5] Taken from newspapers, the fold describes the horizontal border from which Website content is not visible until the user scrolls the page. Since the impression of content to users dramatically decreases for content below the fold, Website owners try to utilize the area above the fold as much as possible, since it is the most valuable.

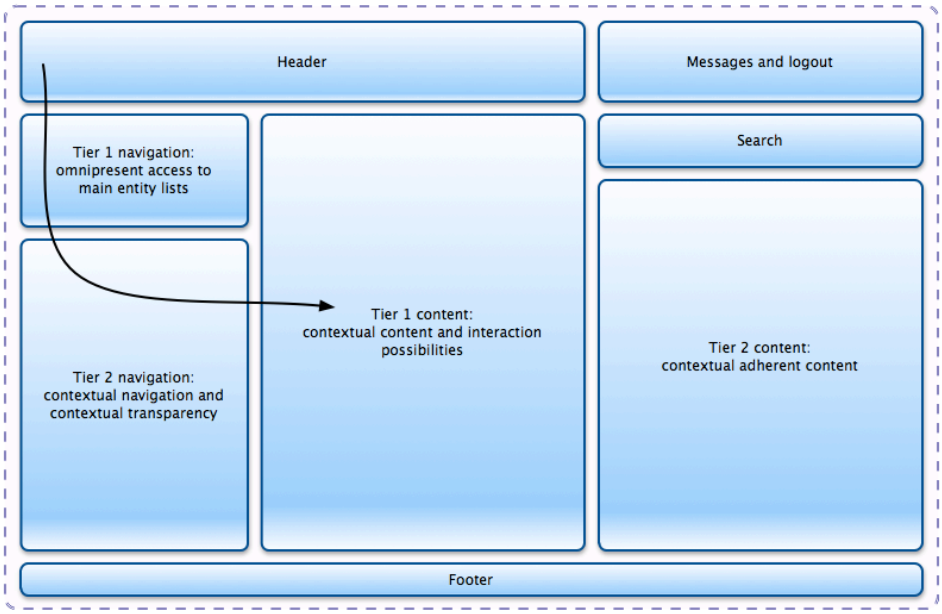

Figure 6.17.: The figure illustrates the application's layout. A scalable three-column layout was applied to carry the amount of content and contextual information defined above to provide transparency. The positioning of elements is based on usability standards.

The following list describes the layout's main elements and their contribution to the above named requirements:

Header: The application header shows a logo only. The logo is linked to the application's main page for shortcut navigation (quasi-standard in usability) [112].

Messages and logout: Omnipresent access to the user's Messages and a logout button. These elements are located at the top right according to user expectations [129] (*conformity with user expectations*, ISO 9241 part 110, cf. Section 2.3.4).

Search: An omnipresent search form is located at the top right of the screen to fulfill user expectations [129] (*conformity with user expectations*, ISO 9241 part 110, cf. Section 2.3.4). It can be used to search Users and Tips and releases any filter set through the currently displayed entity.

Tier 1 Navigation: This primary and omnipresent navigation area provides direct access to main entity lists: Families, Tips and Users. By clicking on one of the three buttons, the filter given through the currently displayed entity is released and a top level list-view is presented. The main entity lists can be permanently filtered and sorted by the user (F22).

Tier 2 Navigation: This secondary navigation area shows context-dependent information depending on the action. In a *list-action* context, it lists entities and minimal information about them. In a *show-action*, it provides minimal information about the entity shown in the Tier 1 Content area plus the above defined list of contextual entities for granting the user contextual understandability and transparency over related entities (R1, F5, F6) as well as contextual navigation through them. Clicking on a link in one of the actions changes the Tier 1 and 2 Content areas as well as Tier 2 Navigation area, the former to display the selected entity and the latter to display related entities.

Tier 1 Content: The primary content area to provide contextual content and interaction possibilities. In a *show-action*, the particular entity is presented and available interaction possibilities are featured. In a *list-action*, contextual informations, explanations and context-related interaction possibilities are listed. In both actions, an activity stream is displayed to list other users' activities on this or related entities, including links to them. In a *content-creation-action*, a form to create the wanted entity is rendered.

Tier 2 Content: A secondary content area to present additional, adherent content in relation to the current entity or entity list, e.g. a map showing contextual entities' locations.

Footer: The footer contains links to static pages, e.g. the terms of service, the imprint, help and about pages.

Figure 6.18 illustrates a mapping of the introduced 3-column layout to the grid. Tier 1 Navigation area (depicted in plain grey) has an exposed location to build a bridge between the header and Tier 2 Navigation.

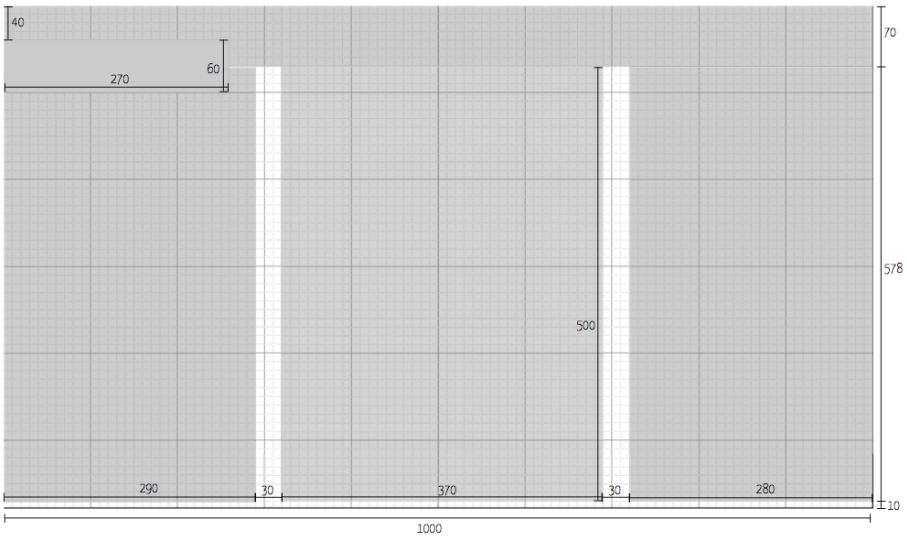

Figure 6.18.: The figure depicts the mapping of the 3-column layout to the 10×10 pixels grid.

6.7.4. Visual Design

This section introduces the application's visual elements. Before going into details, an impression of the overall visual page design and appearance of the application is given. The final part of this section goes through all relevant pages and lists their relevant elements.

Figure 6.19 illustrates a screen shot of the *Family list* page (cf. Figure 6.15 for pages referred to), which is displayed immediately after the user logs in. The implementation of the three-column layout and the page areas described in Section 6.7.3 is obvious. The columns are visually separated by deepened and shaded spacers. Tier 1 Navigation area (cf. Figure 6.17 for areas referred to) presents omnipresent links to the main entity list views. Underneath, a small menu is displayed to filter and sort the listed entities (F22). Tier 2 Navigation area shows a scrollable list of Families accessible to the user, which consist of links to the particular Family, its members and Tips. The latter is intended to give a quantitative impression of the Families' contents without having to navigate to them. Tier 1 Content area starts with a colored block to indicate the entity type dealt within it (cf. colors section 6.7.4), followed by a graphic indicating to the user that the list of Families displayed in Tier 2 Navigation area is an individualized list according to the $families$ function defined in Equation 6.9 in Section 6.6.2. Next is a list of actions available in the current context, followed by a general explanation of what a Family is, and a contextual activity stream (F19) of interesting happenings by other users that are accessible according to the *users* function (cf. Equation 6.14). The header area displays a logo. The messages (F17) and logout (F16) area links to the *Messages list* page, shows the count of unread messages, and links to the user's own

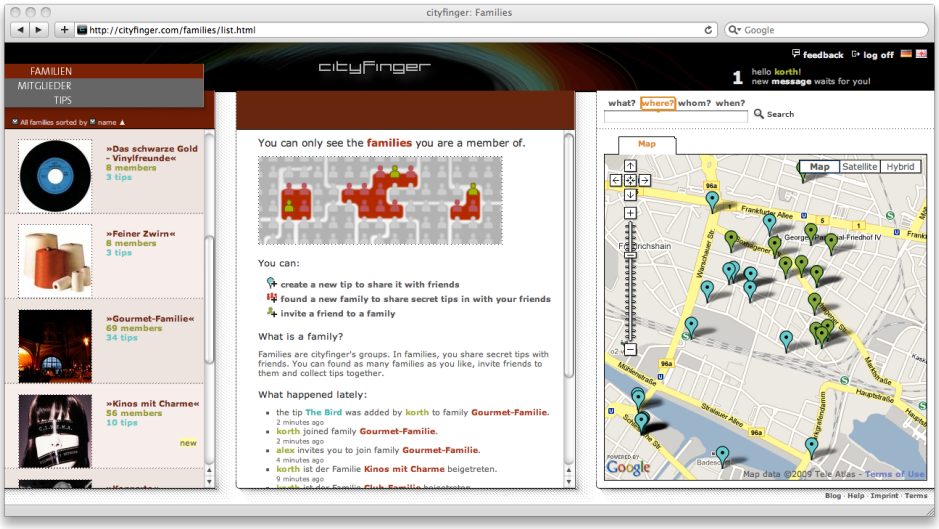

Figure 6.19.: The *Families list* page. As noted in Section 6.7.2 (Figure 6.15), the user is provided with a main page when logged in. This page is the *Family list* page. From here, all information can be accessed with a minimum amount of clicks. The left column shows three top level buttons (Tier 1 Navigation area), followed by a list of Families (Tier 2 Navigation area). The middle column shows options for contextual activity, an activity stream of other users' activities and some explanatory texts and graphics. The right column presents a search form to find Users and Tips, and a contextual map marking Users and Tips that are assigned to the Families (and listed on the left). More pages are presented in the final part of this section.

profile page. Furthermore, it presents links to a feedback form, the logout function and a language selector, allowing the users to toggle between German and English language for non-UGC texts. The search area presents a form to search for Tips, Tips near locations, Users and event dates. Tier 2 Content area contains a map that displays Users (green markers) and Tips (cyan markers) that are contained in the Families listed in Tier 1 Navigation area. The footer area links to static pages.

Selected page elements are described in detail in the following sections.

Colors

The selection of colors does not have a solely aesthetic rationale. Colors and their consistent usage on the site improve the users' understanding of what is displayed and the entity type of it[6] (*self-descriptiveness, conformity with user expectations*, ISO 9241-110, cf. Section 6.7).

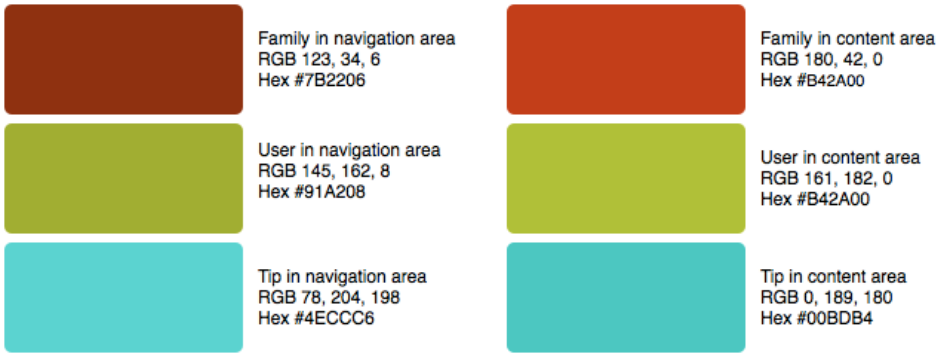

Figure 6.20.: Primary colors for the main entities, depending on usage and positioning: The left hand colors are used in the navigation area for backgrounds and text colors. Both of them are placed on non-white background. The left hand colors are used for coloring textual links on white background.

Figure 6.20 illustrates the application's primary color palette for the main entities. The left hand column colors are used in the navigation area for backgrounds and text colors on non-white background, and at the top of Tier 1 Content area (cf. Figure 6.19). The right hand column colors are slight nuances used for coloring textual links on white background. The primary colors feature a similar hue value which makes them equally visible on different background colors.

The following lists gives an overview of uses for primary colors and their nuances to support transparency of the displayed entity type:

- The colored links to entities used throughout the application, effectively typing them.

[6] Color-typing belongs to the discipline of Information Design.

- The color emphasizing Tier 1 Navigation area: the bar and the background of the filtering and sorting menu.
- The colored bar on top of Tier 1 Content area.
- The alternating background colors used in lists in the Tier 2 Navigation area.
- Icon colors, putting them in a context to an entity type.
- Map marker colors.

Further primary color nuances and secondary colors used throughout the application are illustrated in Appendix Section B.1.

Logo and Typofaces

For the application, a logo was designed and a set of typofaces were used. Both elements are important parts of the visual design and the usability of applications [112], but not relevant for the answering of the scientific questions. Accordingly, these details are listed in the Appendix Section B.2.

Avatar Images

Avatar images of the application are uploaded by its users and thus unpredictable in size, aspect ratio, color and content. To create a uniform overall picture for the application, the images have to be unified. As a solution, uploaded images are cropped to certain sizes and aspect ratios depending on the entity they are assigned to. Presenting entity avatar images with a common ratio supports the usability of the application and the transparency regarding to what is displayed. Besides color-typing (Section 6.7.4), image ratios for entity types are a second approach to improve understandability of displayed information to users. The technique belongs to the discipline of Information Design and increases *self-descriptiveness* and *conformity with user expectations* (ISO 9241-110, cf. Section 6.7).

Family avatar images are cropped to a square aspect ratio and scaled to three different sizes. The images are mainly used by the *Families list* and *Show a Family* page (Figures 6.21 and 6.15).

User avatar images are cropped to a portrait aspect ratio (3 by 4) and scaled to three sizes needed by *Users list* and *Show a User* page (Figure 6.22).

Accordingly, Tip avatar images are resized to fit a landscape aspect ratio (4 by 3) and scaled to different sizes used on *Tips list* and *Show a Tip* page (Figure 6.23).

The different image sizes are needed for displaying the entities in Tier 2 Navigation area and Tier 1 Content area on the pages (for areas, cf. Figure 6.15).

Pages

This part of the section is intended to present the visual design of application's most relevant pages. The selection refers to their linking structure illustrated in Figure 6.15 (Section 6.7.2).

Figure 6.19 illustrates the *Families list* page. The page's details were explained at the beginning of the visual design section (6.7.4). As noted in the IA Section

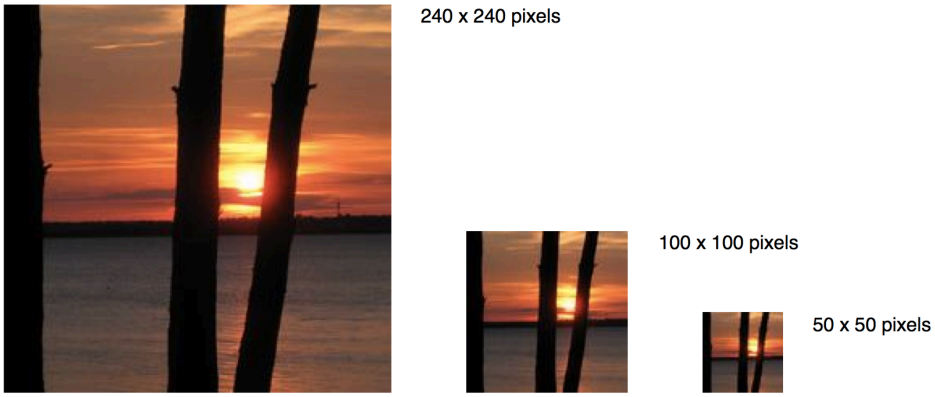

Figure 6.21.: Family avatar images have a square aspect ratio to aid the understandability of the displayed entity type *Family*. The different sizes are used in the different navigation and content areas of the pages.

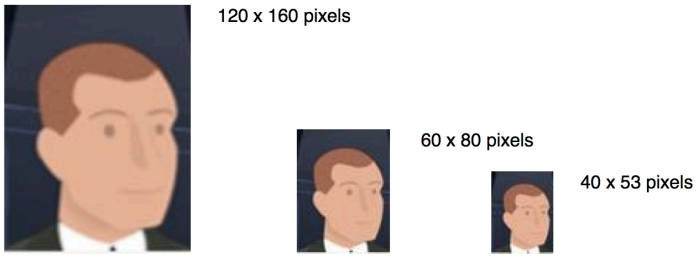

Figure 6.22.: User avatar images feature a portrait aspect ratio to make them recognizable as an entity of type *User*. This is intuitive because it is known from, e.g. passport photos.

Figure 6.23.: All Tip avatar images have a landscape aspect ratio. As with Family and User avatars, pictures uploaded by users are cropped and resized.

(6.7.2), the logged-in user starts navigating from this page. From here, the user has—according to the concept of minimizing the navigation depth and maximizing vertical navigation—multiple options for navigation.

The following two figures present the *Users list* page (Figure 6.24) and the *Tips list* page (Figure 6.25), respectively, as navigable through Tier 1 Navigation area (cf. Section 6.7.3).

On every *list* page[7], the entity list in the Tier 2 Navigation area is always paginated for performance reasons. That means that the area does not list all entities a User can access, but groups them to pages to be addressed separately. The bottom of the left column of figure 6.24 shows the pagination controls to navigate between the pages. Similar to the *Families list* page, the two figures localize the current page's entities on Tier 2 Content area's map. Tier 1 Content area (the middle column) depicts an explanation graphic, options for contextual activities, and an activity stream listing recent activities of other Users. The options listed are determined by the currently listed entity type. Tier 1 Navigation area and the top of the Tier 1 Content area are colored in the respective entity colors (Section 6.7.4) to make obvious the type of the page.

Both figures show a filter that is set: The *User list* page has been filtered to list only users that have an avatar image and the *Tips list* page was filtered to return only Tips that were created by friends. The filters and sorting preferences are located under the Tier 1 Navigation area and remain for future use.

After presenting the most important *list* pages of Figure 6.15, the next paragraphs focus on *show* pages of the entities. The *show* pages[8] are optimized to communicate what entity is presented and what other entities are related, e.g. assigned, to it.

The following two figures depict two versions of the *Show a Family* page. One renders the Family with its Tips (Figure 6.27) and the other with its members (Figure 6.26) listed in Tier 2 Navigation area.

[7] *List* pages are pages that list a particular entity, e.g., Families, Tips or Users. There are further list pages for secondary entities, e.g. Messages or Invitations, which are left out here for the sake of brevity.

[8] *Show* pages are pages that show a particular entity, e.g., a Family, Tip or User.

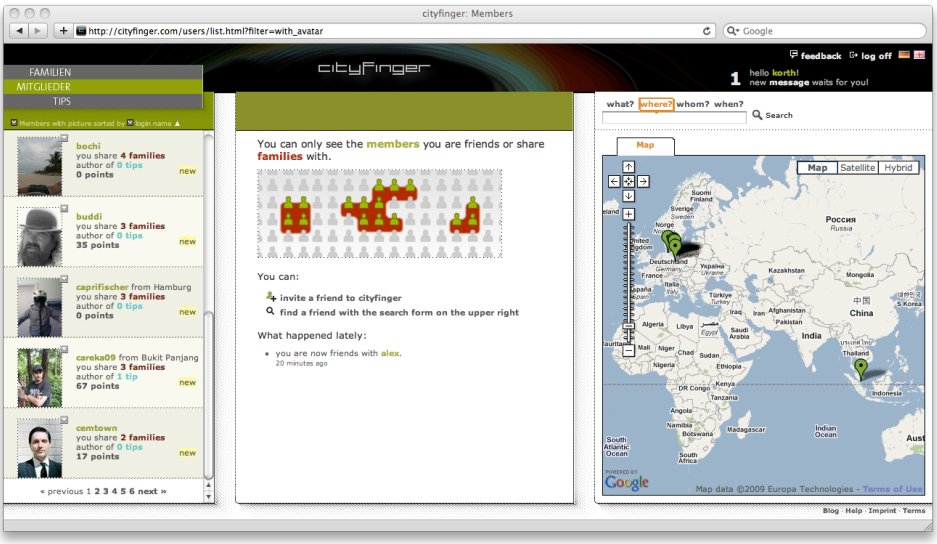

Figure 6.24.: The *Users List* page. Similar to the *Families list* page (Figure 6.19), the viewer is provided with a list of Users accessible to him via his cumulative Family memberships. Tier 1 Navigation area displays the list filter that can be set to, e.g., show only Users with an avatar image sorted ascending by login names. The avatar images have a drop-down menu in their top right which contains user interaction shortcuts to functions, e.g., messaging or inviting. Tier 1 Content area makes transparent that the displayed User list is not global but individualized according to the viewer. Furthermore, it offers contextual interaction possibilities and a contextual activity stream. Tier 2 Content area shows a map locating the Users listed on the left hand column (Tier 2 Navigation area).

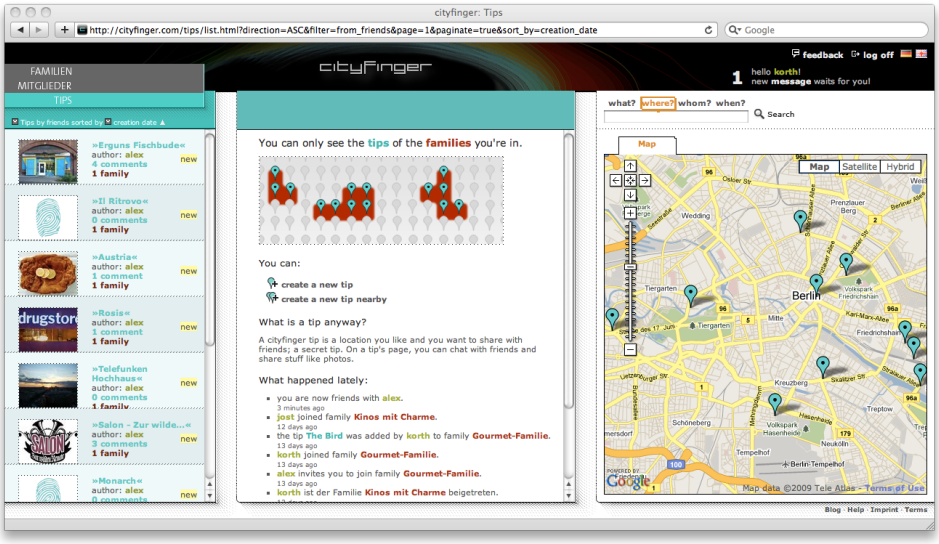

Figure 6.25.: The *Tips List* page listing individually accessible Tips (filtered to those created by friends and sorted by ascending creation date). Tier 1 Navigation area summarizes the Tips by linking to them and their author as well as informing about comments and Family assignment counts. The Tier 1 Content area explains that the list is individualized, followed by contextual interaction possibilities, an explanation text and a contextual activity stream. Tier 2 Content area locates the Tips on a contextual map.

It is essential for the viewer to understand what he or she sees and what it is related to (F5). To support this, Tier 2 Navigation area starts with a small area as a teaser for the Family that is displayed (cf. both figures' Tier 1 Content areas). This teaser is necessary to indicate that both versions of the *Show a Family* page deal with the same Family. The two tabs labeled *members* and *tips* allow the viewer to toggle between the two versions of the page. The toggling also affects the content of the map. Immediately underneath the tabs, a short text informs the viewer about what exactly is listed in the area below. All these elements support the transparency provided to the user about the current interaction situation and view point. This way of constructing the navigation area and providing transparency will be repeated analogously in the forthcoming descriptions of the *Show a Tip* and *Show a User* pages.

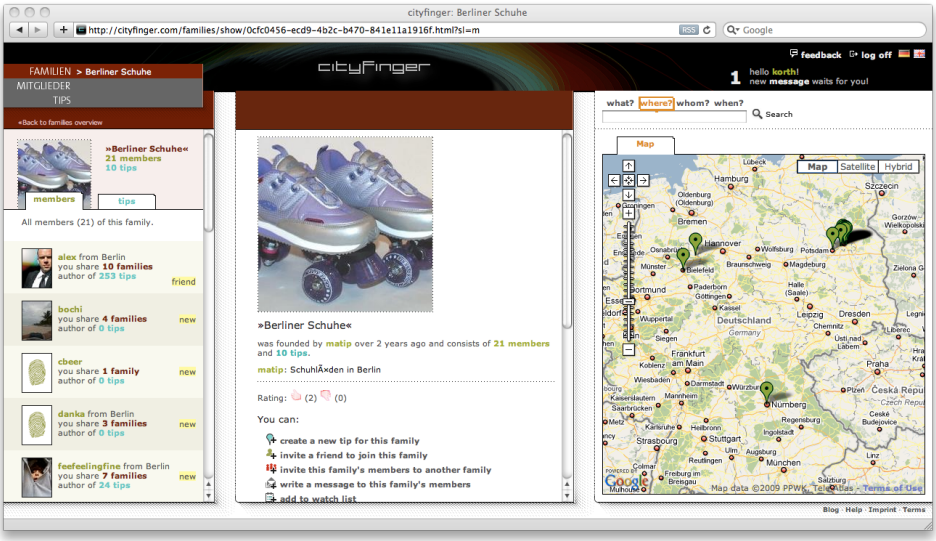

Figure 6.26.: The *Show a Family* page focussing on its members in Tier 2 Navigation area and on its map. Tier 1 Content area is identical with the one in Figure 6.26. It contains a square avatar image, the Family's description through its founder, ratings, contextual actions, an activity stream, and a comments section where users can communicate.

Both figures render an identical Tier 1 Content area: Starting with a red-brown bar to indicate the type of the displayed entity (Section 6.7.4 about colors), the Family's avatar image is shown and features a square aspect ration (Section 6.7.4). Between the Family's title and description, which were given by its founder, a machine-generated sentence concluding details about founder, creation date, tips and members count is given. User-generated texts are rendered in a slightly darker toner than system-generated ones to emphasize them for the human eye.

A dotted line separates the header section, which is changeable only by the creator, from areas that every user can affect or click. After it, a rating section is

displayed to allow users to rate the Family, followed by a list of contextual actions is offered to the viewer depending on the viewer's Family role and rights. Family creators can choose to remove Tips from a Family (F8) and edit Family details. Casual members might be able to add Tips and Users to the Family if the Family founder has granted the respective rights. A comments section is rendered next, allowing all members to communicate on a Family basis. All comments can be deleted by the comment creator and the entity creator.

Compared to the following *Show a Tip* and *Show a User* pages, and due to the fact that a Family has no dedicated location compared to Tips and Users, the map in the Tier 2 Content area differs in the two displayed versions of the *Show a Family* page: The assigned Tips and Users are marked on it, respectively. When clicked, the markers open a bubble displaying image avatar, name and a direct link to the instance.

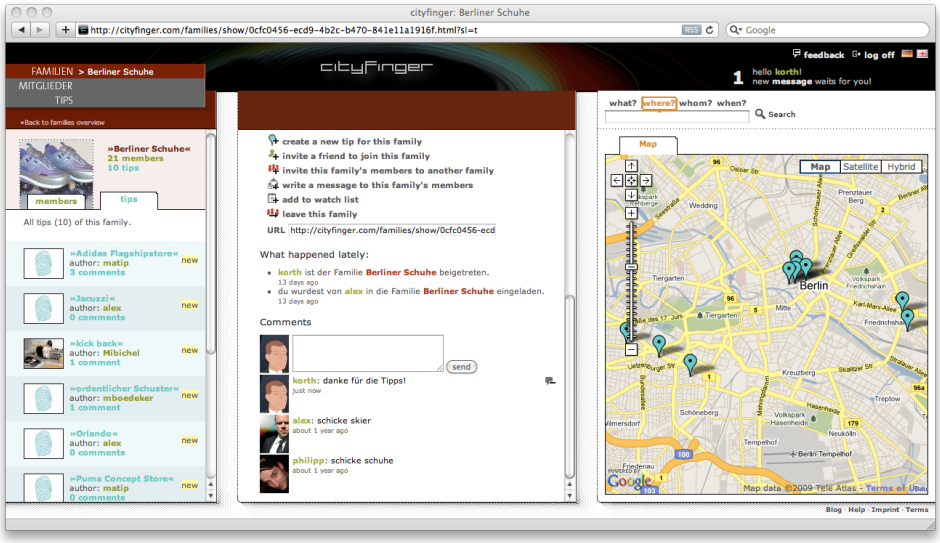

Figure 6.27.: *Show a Family* page focussing on its Tips in the Tier 2 Navigation area and on its map. The Tier 1 Content area is scrolled to the bottom, but identical with the one of figure 6.26.

The next two figures (Figure 6.28 and Figure 6.29) illustrate *Show a Tip* pages of a particular Tip. Analogously to the preceding paragraph, the first figure in Tier 2 Navigation area lists the Families, the Tip was assigned to, as the second figure lists the Users that have access to it (F6). Above both lists, the Tip as such is displayed as a teaser.

Figure 6.28 in Tier 2 Navigation area shows a list of Families that the Tip was assigned to by its author (a User named *alex*). According to the concept (Section 6.1), a Tip can only be accessed by a User if he is member of at least one Family the Tip was assigned to (F1, F3). Obviously, the Tip was assigned to an additional Family that the User *korth*, who is logged in (the logged in user's login name is displayed

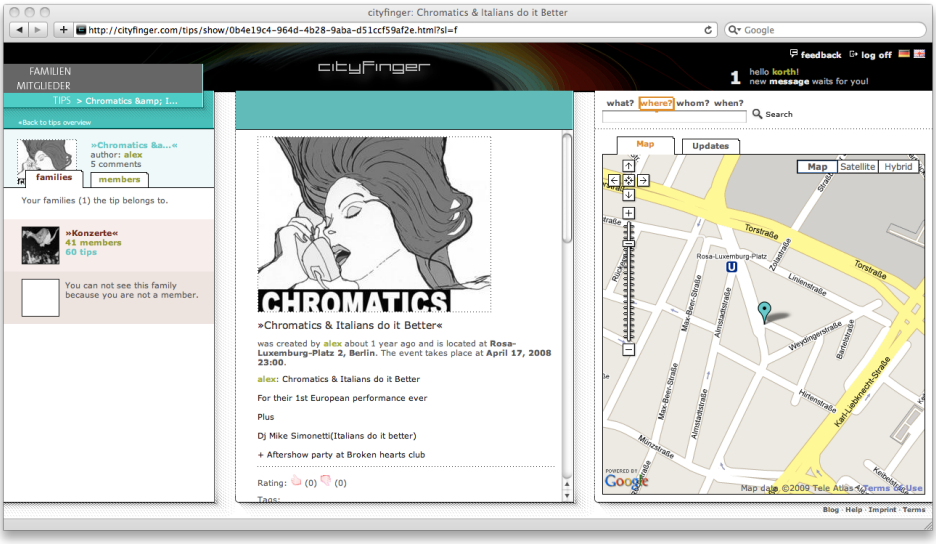

Figure 6.28.: The *Show a Tip* page focussing on its Families in Tier 2 Navigation area. Tier 1 Content area is identical with the one in figure 6.29. It contains an avatar image in landscape aspect ratio, the Tip's location and event date (only if the Tip is an event), followed by its description by its creator, ratings, tags, attendance list (events only), contributions by other users, contextual actions, a photo album for users to upload pictures, an activity stream, and a comments section.

in the messages and logout area), is not a member of. The transparency that the system is giving to the User *korth* is that there are other Families whose members have access to this Tip and to assets that User *korth* would produce, e.g. disclosing his attendance, posting a contribution, uploading a picture, or leaving a comment. All details of that Family, except the indication of its existence, are hidden.

Tier 1 Content area contains all the Tip's attributes and assets. It starts with a cyan-colored bar to indicate that an entity of type Tip is displayed (cf. 6.7.4). The bar is followed by the avatar image and the Tip's title. As described in Section 6.7.4, all Tips have an avatar image with a landscape aspect ratio to differentiate the entity from Users and Families. The title is followed by a machine-generated sentence summarizing the Tip's creator, creation age, location and event date (if it is an event-type Tip). Next is the author's description: The dotted separator is followed by a rating and tagging area. It the Tip is of type event, the next field is an attendance list for everyone to see who is coming to the event. The forthcoming contributions section is a format for users to create contributions of higher value and presence than a comment. Every user can create only one contribution that can be changed and deleted at will. Next is the omnipresent and context-related list of possible actions. It provides functions depending on the viewer. The Tip creator for instance is provided with an option to add the Tip to further Families (F7), whereas the common user is asked to show the Tip to further users by inviting them to the Families the Tip was assigned to (F9). The Tip's photo album is shown underneath, where all users can upload an unlimited amount of pictures with caption texts. Tier 1 Content area is finalized by a comments section for users to communicate.

In contrast to *Show a Family* pages, *Show a Tip* pages in Tier 2 Content area always show the Tip's location on the map, regardless of the selection in Tier 2 Navigation area.

Figure 6.29 lists Users with access to the Tip. Similar to Figure 6.28, the Users list ends with Users that are obfuscated to make transparent that unaccessible Users will be able to access the assets added to the Tip. Every User that is listed and accessible, shows not only his or her name and the avatar image (in portrait aspect ratio) but also the number of shared Families, which is a simple and understandable measurement for a quantitive commonality. Also, the amount of Tip creations is displayed, which is a simple measurement for the user's platform activity. Both numbers are linked to lists showing further details. Finally, Users that are friends or that have never been addressed before are marked to emphasize a distinctiveness compared to the rest of Users.

The next two figures illustrate two versions of the *Show a User* page. The first Figure 6.30 displays the viewer's own User profile. Analogous to the *Show a Family* and *Show a Tip* page, Tier 2 Navigation area starts with a teaser of the selected User. After the tabs to toggle between the Families and Tips lists regarding the User, a short text again briefly explains what exactly is listed underneath.

Tier 1 Content area starts with a green bar to indicate the User-type of the following. As described in Section 6.7.4, Users are distinguished from Families and Tips also by the portrait aspect ratio of the avatar image. To the right of the image, the User's profile details are communicated. The pen icon indicates to the viewer that an attribute is editable (F12). Upon clicking, it changes the field to a text area as shown with the User's surname. An editing of the address calls a geo-coder service in the background on the server-side, which returns geo-coordinates

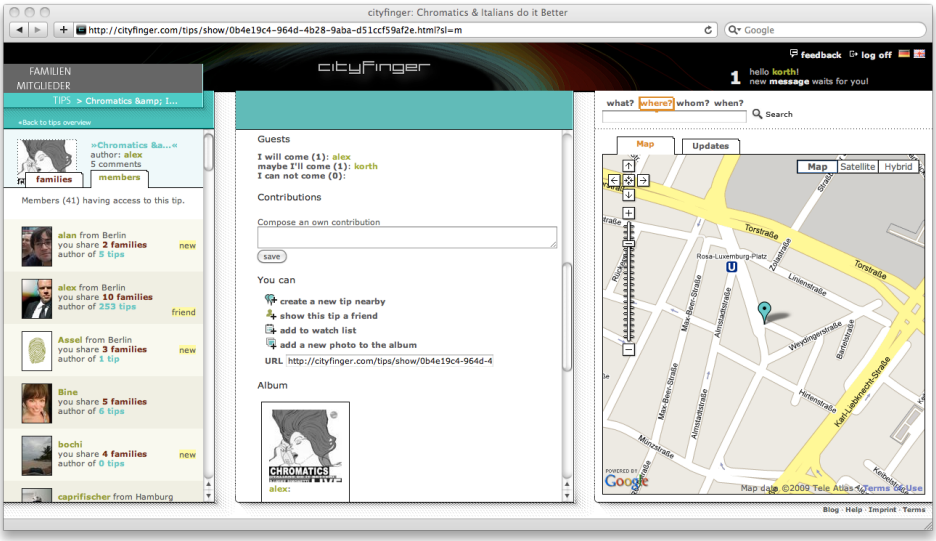

Figure 6.29.: The *Show a Tip* page listing Users that have access to it in the Tier 2 Navigation area. Everything else is identical to Figure 6.28.

of the address, which are used to re-set the marker on the map displayed on the right hand. The zoom level of the map is co-related to an estimated accuracy level returned by the geo-coder. The address is followed by a handful of achievements and memberships the User made or has. Clicking on them switches and filters the entity list in Tier 2 Navigation area to, e.g., only Families the shown User founded. After a self-description area, which User *alex* used to embed a video (Figure 6.31), the viewer again finds contextual interaction possibilities, including a function to delete the profile (F11) followed by a comments section for Users with access to communicate.

Tier 2 Content area marks the address of the displayed User. This might be used for the User's current location in a future version of the application.

Figure 6.32 presents the *Create a Tip form* page. According to feature F7, the creating user can, after choosing a Tip name, select the Families the Tip shall be assigned to. The Family list contains only Families that the user has the right to add Tips to. User and Tip counts shown under the Family names offer the user transparency and control over the audience he is about to make the new Tip accessible to (F5). The form continues (not displayed on the figure) by asking the creator for the Tip's location, type, profile picture, description and a hyperlink, of which only the location is a mandatory information. The location can be provided by either dragging the marker on the map or by entering an address that is geo-located to geo-coordinates.

Figure 6.33 illustrates the *invite a User form* page. Tier 2 Navigation area lists accessible Users to be invited to Families listed in Tier 1 Content area. Inviting Users to Families is part of Features F2 and F9 to control Family memberships.

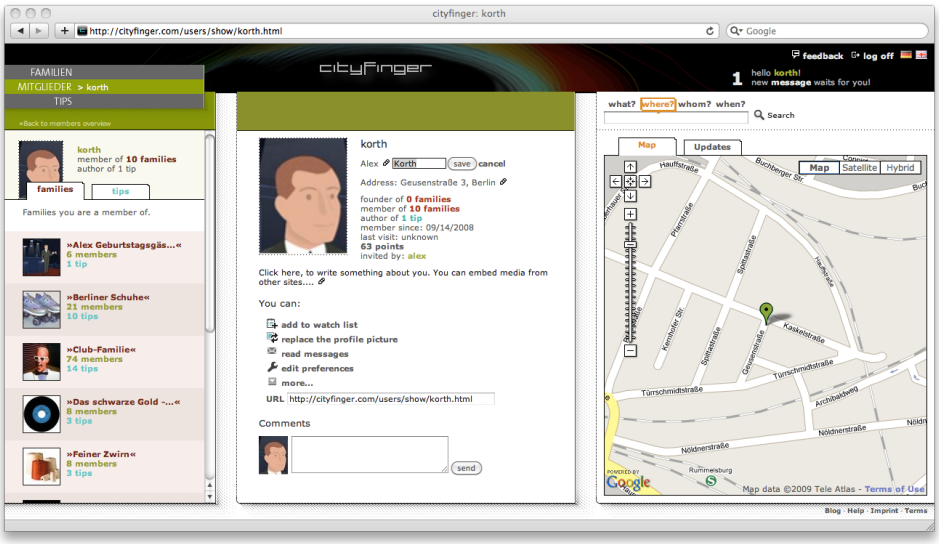

Figure 6.30.: *Show a User* page of a logged in user. Tier 2 Navigation area lists Families the user is a member of, as the text under the tabs says. Tier 1 Content area lists the User profile itself, starting with an avatar image in portrait aspect ratio. The pen icons behind some attributes indicate that it is editable and, when clicked, replace the attribute with a text entry field, as illustrated with the User's surname (F12). After the User's login, forename, surname and address, his achievements and memberships are listed. Followed by a self-description, there are contextual activity possibilities listed. These include a link to delete the profile (F11). The column ends with a comments section. Tier 2 Content area (the map) marks the User's address which can by dragged by mouse for relocation or which auto-sets if the address is edited.

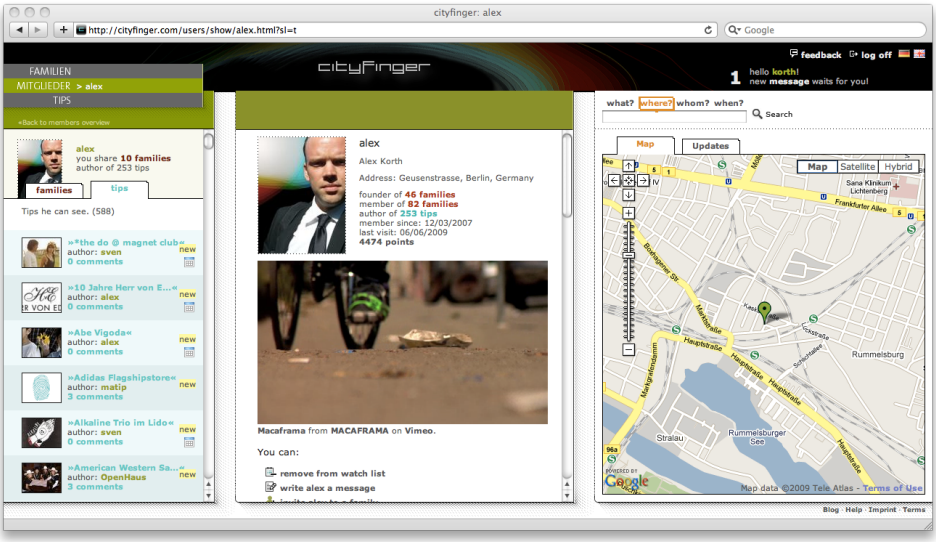

Figure 6.31.: *Show a User* page of a foreign User with the Tips tab selected. The lists shows Tips that the shown User can access (in this case 588), of which the viewer may not be allowed to access all (unaccessible Tips are again obfuscated). Tier 1 Content area now misses the pen icons, because the viewer is not allowed to edit another User's details. The achievements and memberships are clickable and filter Tier 2 Navigation area accordingly. The viewed User embedded a video in the description area.

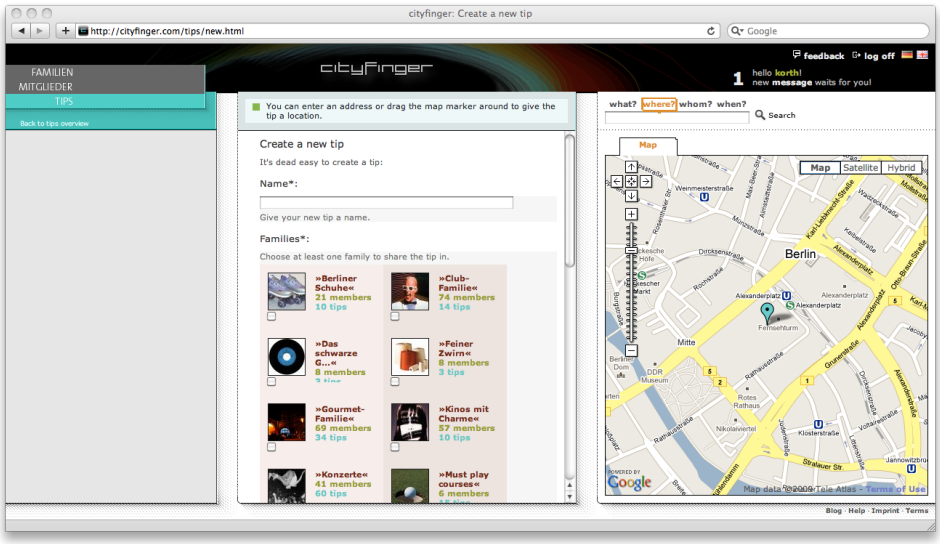

Figure 6.32.: The *Create a Tip form* page. Besides the Tip attribute form fields, the creator is provided with a list of Families to assign the Tip to (F7), including the particular member and Tip counts (F5). The Tip location can be set by dragging the marker on the map or by an address that is geo-coded to latitude and longitude.

As described in detail in Figure 6.7 in the framework section (6.6), the user can not only invite multiple existing Users to multiple Families, but can also add email addresses of personal friends to the form that are invited to the platform if they are not Users already.

Figure 6.33.: The *Invite a User form* page provides users with a tool to invite existing Users and external friends by email (Features F2, F9). A Families list of the Families that the user has the rights to invite Users to, is displayed providing member and Tip counts to make their respective sizes transparent.

6.7.5. Software Ergonomics

The application is required (AR5) to provide software ergonomics (ISO 9241-110, cf. Section 2.3.4 to improve user efficiency when working with it. This section concludes how application features, both functional and non-functional, contribute to the standard's principles.

Suitability for the task

As described in the features section (6.4), the application only has one purpose: Sharing secret Tips with friends (F4). No superfluous features are provided that exceed this main task. The goal was to keep the application as simple as possible.

Self-descriptiveness

The transparency the application aims to provide, is not solely achieved by Features F5 and F6. The application and its visual design provide a number of features and

displays to optimize its transparency and self-descriptiveness.

- The wording and order of Tier 1 Navigation area's elements already provides a short summary of what the application is about.
- The graphics and the explanation texts emphasize that Tips are secret (located on top of Tier 1 Content area on *list* pages).
- Tier 2 Navigation area on *show* pages makes clear what is being displayed and that the lists below are entities that have a relation to the one displayed.
- Inaccessible (obfuscated) Tips and Families in lists on, e.g. *User show* pages.
- The colored counts of related entities of a listed or displayed entity. These underline entity relationships and their general taxonomy, e.g. Families consisting of Tips and Users.

Suitability for learning

On top of its self-descriptiveness, the application minimizes the required learning time by providing feedback on user actions and by displaying descriptive text elements in, e.g. forms.

Suitability for individualization

The application is *individualized by design*: There are no two users that see the same application since it is individualized by Family memberships and the resulting accessibility of entities. Furthermore, every user can individualize his or her user profile.

Conformity with user expectations

The application is consistent regarding its explanation texts and wordings to provide a maximum of conformity with user expectations. Quasi-standards, such as logo or search form location, are re-used to meet expectations and to lower the learning effort [112].

Controllability

Besides the access control Features (F7-F12), the application provides the following additional control features:

- The ability to edit most attributes of created entities, including the own user profile.
- The feature to delete entity assets, such as comments and contributions.
- The filtering and sorting menu of Tier 1 Navigation area on *list* pages.

Error tolerance

The application is error tolerant in the creation and editing of entities. It helps the user by correcting erroneous entries, by guiding him through long forms and by pre-filling erroneous forms with the given correct values.

6.8. Summary

This chapter represents the second major contribution of this thesis: The creation of a prototypic platform consisting of a software framework and an application to preserve user privacies. Therefore, it based on the problem areas reported of in Chapter 3, related works listed in Chapter 4 to address the problems and the privacy-related requirements listed in Chapter 5 to approach this challenge.

After providing the reader with a concept and a scenario for an application that contains a use-case which requires user privacies and content to be protected, the chapter continued with defining application-specific requirements and a features list to cumulatively realize the concept and to provide an approach for the majority of the privacy-related requirements. Entities and their relations were designed as a software-architectural basis for the underlying framework. The framework and its peculiarities were introduced and explained in detail.

Since many features for transparency and control of user privacies are located on the GUI, there is no privacy protection without an application with a GUI. Accordingly, an exemplary application was build on top of the framework and described in detail. Starting by creating an Information Architecture, a Navigation Design and a general layout, the visual design of the most important elements and pages was described, focusing on how users are provided with transparency and control to effectively and efficiently protect their privacies.

Finally, the application's features and properties were put into relation to the principles of software ergonomics standard ISO 9241-110 to improve users' efficiency on using the application.

7. Evaluation of the Platform Approach

Chapter 6 introduced this thesis' second major contribution: A prototypic platform called Cityfinger, which consists of a framework and an application to empower users to efficiently protect their privacies.

After outlining the corpus that was created by people using the platform, and providing a brief overview over the users' visiting traffic, this chapter evaluates the platform from three perspectives: Firstly, a user survey was conducted to evaluate the platform's effectiveness and efficiency regarding privacy protection, also in comparison to other SNSs. Secondly, the platform's social network graph as it emerged during usage, is compared to those of other platforms. Thirdly, the platform's privacy score is computed and compared to those of the platforms in Table 5.4 (Chapter 5).

7.1. Corpus Description

The corpus was produced between December 2007 and January 2010. The majority of data was generated in 2008.

Primary entities	Families	130
	Tips	644
	Users	191
Primary entity relations	Memberships	1,875
	Family-Tip assignments	745
Secondary entities	Invitations	288
	Messages (total)	2,193
	Messages (user-generated)	566
	Friendships	120
	Preferences	12,032
	Bulletins	35,377
Adherent entities	Avatars	515
	Contributions	39
	Comments	1,024
	Assets	221
	Taggings	762

Table 7.1.: Data corpus produced by users between December 2007 and January 2010.

Since its launch, 965 primary entities, i.e. 191 Users, 644 Tips and 130 Families, were created by users. The users invited friends to Families 1,875 times. The 644 Tips created were assigned to Families 745 times.

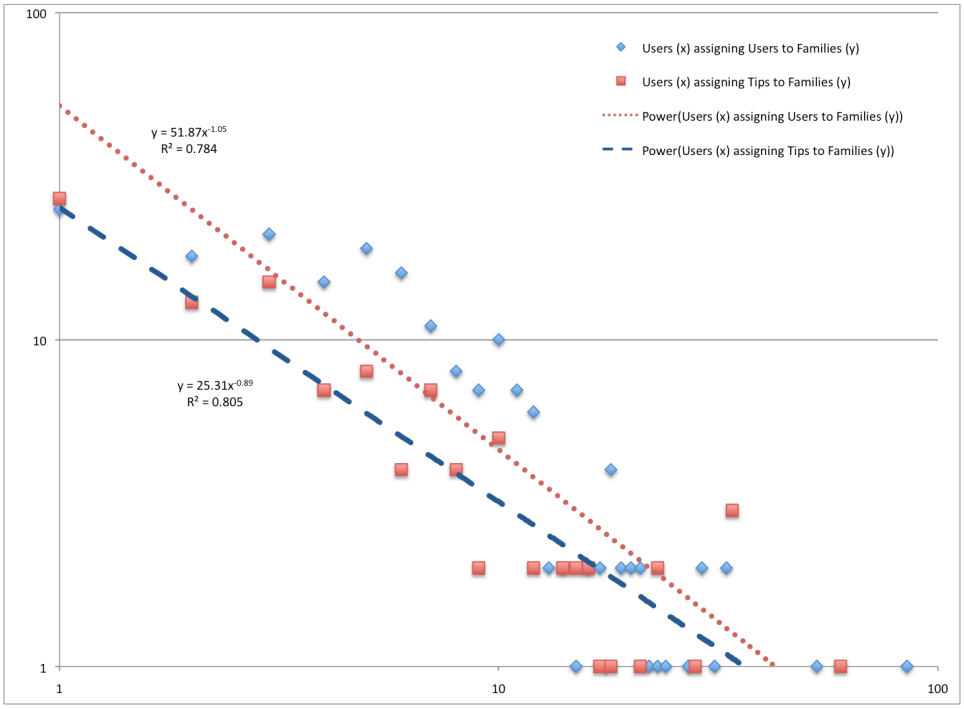

Figure 7.1.: Assignment distribution of Users to Families (Memberships) and Tips to Families. The dotted and dashed lines represent the estimated Power Law functions of the distributions.

Figure 7.1 illustrates the assignment distribution of Users to Families (Memberships) and Tips to Families. Although the quantities are small, both distributions indicate a Power Law distribution [109]. Accordingly, and referring to Nielson's 90-9-1 rule (Section 2.2.1), the flatter slope of the trend line of Tip assignments to Families (blue, dashed line) might indicate that this action is easier to perform than creating a new Family Membership by inviting Users (red, dotted line).

2,193 Messages were created, of which 566 were user-generated. The rest were created automatically, triggered by user activity to notify other users to, e.g., opt-in to a Family invitation. Only 120 Friendships were created, which may be caused by the fact that a Family membership itself provides a sufficient tool to obtain Social Context. Besides the 12,032 Preferences that the platform stored for its users, the notification mechanisms produced 35,377 Bulletins for the platform's activity stream to keep users informed about activities of others.

Concerning adherent entities, the users created 515 avatars, meaning that more than half of all primary entities were equipped with an avatar image. The low number of 39 contributions is explained by the fact that the feature was added very

late and thus was not used much. The total amount of 1,024 Comments appears low due to the fact that all primary entities can be commented. This may be caused by the fact that users have to scroll within the Tier 1 Content area to reach the comments section, resulting in lower usage numbers (the comments section is *below the fold*, cf. Section 6.7.3). The number of 515 Assets, i.e. photo album pictures of Tips, means that on average, every Tip was augmented with almost two pictures. The number of tags used for Tips averages one tag per Tip, indicating a rare usage of tags to categorize Tips.

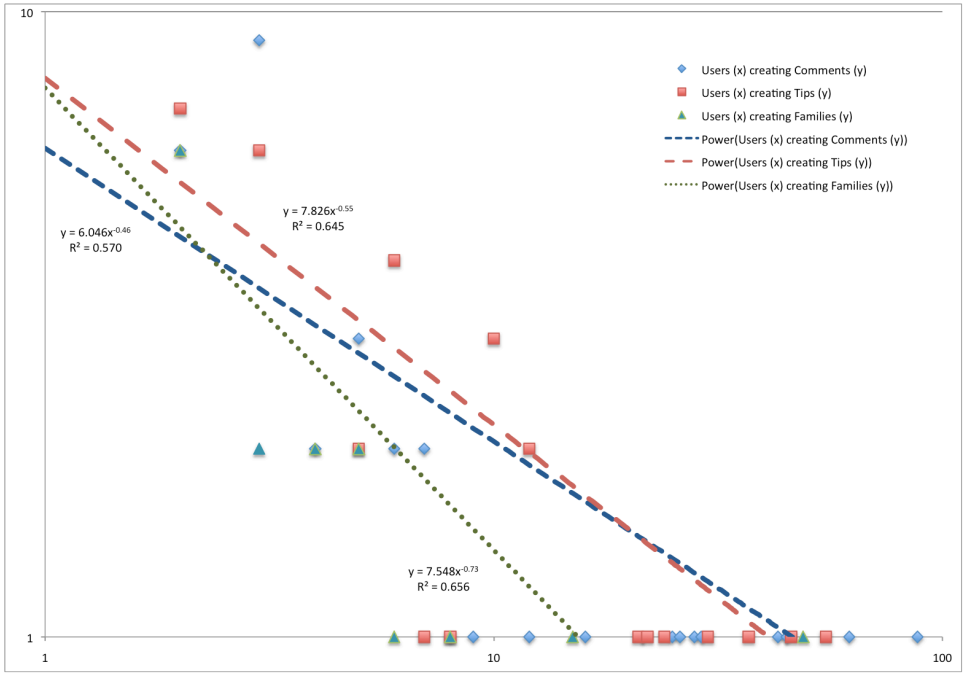

Figure 7.2.: Entity creation distribution of Users creating Comments, Tips and Families, respectively. (Some runaway-values were not plotted.)

Figure 7.2 analyzes the creation of entities. It shows the distribution of Users creating Comments, Tips and Families, respectively. The lower exponent of the trend line of Comment creations (blue, dashed line) indicates that creating a Comment is easier to achieve than creating a Tip (red, dashed line) or a Family (green, dotted line): Creating a Comment requires significantly less effort from the user compared to the forms that need to be filled to create a Tip or Family.

7.2. Usage Statistics

Although the platform still exists, only usage statistics from December 6, 2007 until August 7, 2008, were tracked. As mentioned before, the majority of overall platform activity took place within this time interval. The following numbers were observed:

Unique visitors	822
Visits	2,481
Page impressions	18,476
Primary entity impressions	6,078

Table 7.2.: Usage statistics between December 6, 2007 and August 7, 2008.

The platform manifests a total of 822 unique visitors during the named period of time, who visited the platform for a total of 2,481 times. During these visits, they requested a total of 18,476 page impressions, of which 6,078 were primary entity impressions (*show actions*). The users viewed an average of 7.45 pages per visit and stayed for an average of almost 7 minutes.

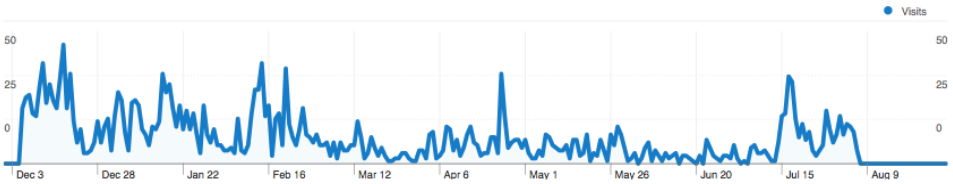

Figure 7.3.: Visit statistics between December 6, 2007 and August 7, 2008.

The majority of traffic sources were direct traffic (64%), which indicates that most users entered the URL directly, bookmarked the page, or were lead on site by, e.g. a deep link inside an email the platform sent about an activity of another user. The remaining traffic originated from referring sites (20%) and search engines (16%). On the latter, the keyword *cityfinger* was used in 88.83% of search cases to find the platform.

7.3. Survey

A user survey consisting of 20 questions was performed amongst the users of the platform to evaluate its quality regarding user understanding and usability, its effectiveness regarding privacy awareness and protection, and the subjectively perceived relevancy of users and content. Although the sample ($N = 67$) is small for a quantitative analysis, it allows for tendency conclusions.

The majority of respondents (ca. 61%) can be classified as Privacy Pragmatists [92] (Section 3.1). Most of them are male (87%) and between 26 and 35 years of age (71%). They earn an average of ca. 47,000 Euro annually and most possess an academic degree (75%). 70% actively use the Internet between 2–8 hours daily and 43% use SNSs actively for about 0.5–1 hours every day. The questionnaire and the answers given by the respondents are listed in Appendix A.

This section provides answers to Research Questions 3 and 4 (cf. Section 1.1).

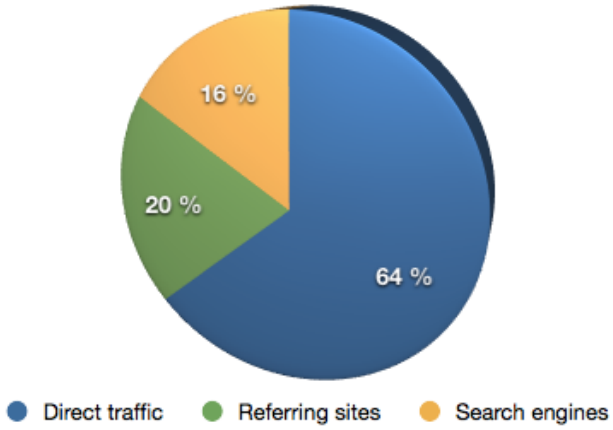

Figure 7.4.: Traffic sources distribution.

7.3.1. Platform Effectiveness for Privacy Protection

To answer Research Question 3 (cf. Section 1.1), the following evaluation looks firstly at whether users understood the value proposition of the platform and how to use it, especially its control and transparency features. Secondly, we evaluate how the platform as well as its control and transparency functionality were rated by users in comparison to other SNSs.

Survey Questions (SQ) 4–6 (cf. Section A.2) investigated the understanding of the platform's value proposition, how information is organized, and how to navigate it.

The majority of users (77%) found it easy to understand what to do with Cityfinger (Figure 7.5). They answered the question (SQ4) with *fully agree* (37%) or *partially agree* (40%) which shows that the platform's value proposition was non-ambiguous and clearly sensible to the majority of users.

Survey Question 5 was aimed at measuring whether the Information Architecture (Section 6.7.2) was understood by users. 63% of users answered the question with *fully agree* (19%) or *partially agree* (43%), which indicate a working concept (Figure 7.5).

Survey Question 6 tested the comprehensibility of the Navigation Design (Section 6.7.2). 69% of the respondents verified the intuitiveness of information navigation (Figure 7.5) by answering *fully agree* (27%) or *partially agree* (42%).

Table 7.3 analyses the correlation (Kendall's τ) regarding the respondents in Survey Questions 4–6. The correlations between questions are very significant: The same group of users understood the platform's value proposition, its Information Architecture and Navigation Design.

Survey Question 8 (Figure 7.6) asked if the provided features to control the audience for information were understood. 60% of users answered with *fully agree* (39%) or *partially agree* (21 %), with an average value in the middle of the *partially agree* interval. Whereas 39% where *unsure*, no one *partially disagreed* and only one

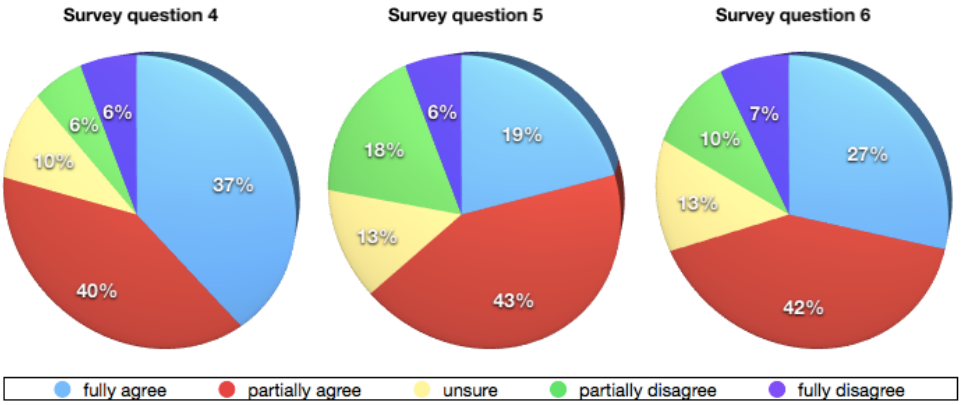

Figure 7.5.: Results to Survey Questions 4–6: 77% of users found it easy to understand Cityfinger's value proposition. 63% comprehended how information is organized, and 69% of the respondents confirmed the intuitiveness of information navigation.

		SQ4	SQ5	SQ6
SQ4	Correlation coefficient	1.000	.432**	.362**
	Significance (2-tailed)		.000	.001
	N	67	67	67
SQ5	Correlation coefficient	.432**	1.000	.486**
	Significance (2-tailed)	.000		.000
	N	67	67	67
SQ6	Correlation coefficient	.362**	.486**	1.000
	Significance (2-tailed)	.001	.000	
	N	67	67	67

** Correlation is significant at the 0.01 level (2-tailed).

Table 7.3.: Correlations (Kendall's τ) of questions 4–6. The same group of users understood the platform's value proposition, its Information Architecture and Navigation Design.

respondent (1.6%) *fully disagreed* on being provided features to control the audience for information.

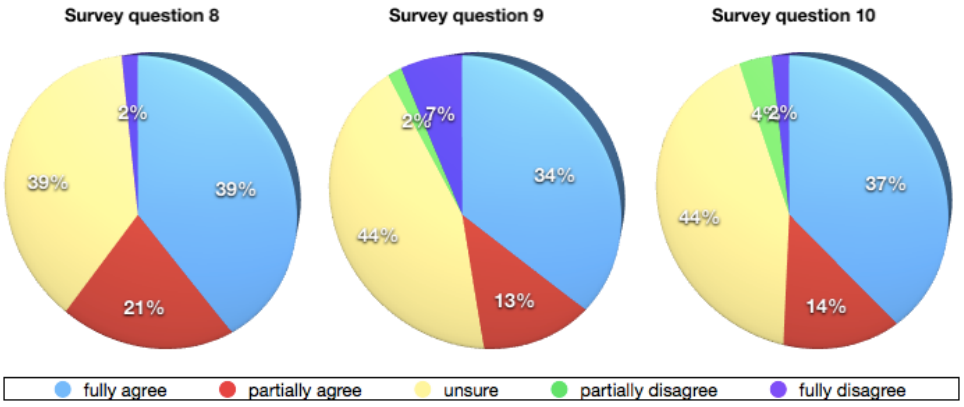

Figure 7.6.: Results to Survey Questions 8–10: 60% of users understood the provided features to control the audience for information. 48% of users confirmed the transparency of who has access to published information. 51% of users communicate very openly with their friends, because they have a transparency about who has access to published information.

Survey Questions 9 and 10 asked the respondents concerning transparency features (Figure 7.6). 48% *fully agreed* (34%) or *partially agreed* (13%) on the platform's transparency concerning who has access to published information (Survey Question 9). The same results were observed (51%) when asking users if they can communicate openly with their friends, because they have transparency about who has access (Survey Question 10). Both average values range in the *partially agree* interval.

Table 7.4 lists correlations between questions 4–6 and questions 8–10 (answers to Cityfinger). The numerous significant correlations show that users who understood Cityfinger's value proposition, Information Architecture, and Navigation Design also understood its control features and, especially, its transparency features. Consequently, it can be assumed that further improvement of the intuitiveness of the platform will result in a better understanding of the control and transparency features (discussed in Chapter 8).

The evaluation so far has shown that about half of users were effectively empowered with transparency and control features to remain aware of and steer the audience of published information, i.e. their privacy.

In the following, Cityfinger is compared to other SNSs, namely Facebook, StudiVZ[1] / MeinVZ[2], XING, Lokalisten[3], Wer Kennt Wen (WKW)[4], LinkedIn and MySpace. Unfortunately, all SNSs except Facebook and XING had to be omitted

[1] `http://www.studivz.net`
[2] `http://www.meinvz.net`
[3] `http://www.lokalisten.de`
[4] `http://www.wer-kennt-wen.de`

		SQ8	SQ9	SQ10
SQ4	Correlation coefficient	.169	.273*	.440**
	Significance (2-tailed)	.132	.015	.000
	N	62	61	57
SQ5	Correlation coefficient	.153	.308**	.395**
	Significance (2-tailed)	.166	.005	.001
	N	62	61	57
SQ6	Correlation coefficient	.228*	.384**	.361**
	Significance (2-tailed)	.039	.001	.002
	N	62	61	57

** Correlation is significant at the 0.01 level (2-tailed).
* Correlation is significant at the 0.05 level (2-tailed).

Table 7.4.: Numerous significant correlations between questions 4–6 and questions 8–10 (answers to Cityfinger) state that users who understood Cityfinger's value proposition, Information Architecture and Navigation Design also understood its control and, especially, its transparency features.

for comparison due to small usage rates amongst survey participants. Facebook is used by 55 of the respondents, XING by 46.

Concerning the existence of control features (Survey Question 8), Cityfinger (59.68% *agreed*[5] at an average value of 3.95[6]) scored only slightly higher than Facebook (65.63% *agreed* at an average value of 3.63) and XING (63.46% *agreed* at an average value of 3.67). Similar results of users rating the SNSs almost equally were observed through Survey Questions 9 and 10, which asked users about the perceived transparency of information reach when posting online (Table 7.5).

	SQ8	SQ9	SQ10
Cityfinger	60% *agreed*, avg. 3.95	48% *agreed*, avg. 3.67	51% *agreed*, avg. 3.81
Facebook	66% *agreed*, avg. 3.63	43% *agreed*, avg. 3.10	53% *agreed*, avg. 3.46
XING	63% *agreed*, avg. 3.67	49% *agreed*, avg. 3.41	49% *agreed*, avg. 3.66

Table 7.5.: At first sight, the respondents rated the perceived features for control (SQ8) and transparency (SQs 9 and 10) of the SNSs equally.

Respondents rating Cityfinger's control (Table 7.6) and transparency (Tables 7.7 and 7.8) highly did not do so for Facebook's and XING's. The latter two SNSs' control and transparency features, on the other hand, show a high correlation regarding respondents' ratings.

It was shown that the fraction of respondents who understood the platform Cityfinger as such (SQs 4–6), also rated its control and transparency features as be-

[5] In the following, the term *agreed* refers to the sum of respondents answering with *fully agree* plus *partially agree*.

[6] Average values were computed by interpreting answer options as ordinate and equidistant with a value of 1 for *fully disagree* to 5 for *fully agree*.

Survey Question 8		Cityfinger	Facebook	XING
Cityfinger	Correlation coefficient	1.000	.076	.040
	Significance (2-tailed)		.499	.753
	N	62	60	48
Facebook	Correlation coefficient	.076	1.000	.386**
	Significance (2-tailed)	.499		.001
	N	60	64	51
XING	Correlation coefficient	.040	.386**	1.000
	Significance (2-tailed)	.753	.001	
	N	48	51	52

** Correlation is significant at the 0.01 level (2-tailed).

Table 7.6.: Correlations between Cityfinger, Facebook and XING in Survey Question 8. While there is no correlation between Cityfinger and the other options, Facebook and XING correlate significantly.

Survey Question 9		Cityfinger	Facebook	XING
Cityfinger	Correlation coefficient	1.000	.160	-.077
	Significance (2-tailed)		.153	.538
	N	61	58	48
Facebook	Correlation coefficient	.160	1.000	.355**
	Significance (2-tailed)	.154		.002
	N	58	63	50
XING	Correlation coefficient	-.077	.355**	1.000
	Significance (2-tailed)	.538	.002	
	N	48	50	51

** Correlation is significant at the 0.01 level (2-tailed).

Table 7.7.: Correlations between Cityfinger, Facebook and XING in Survey Question 9. While there is no correlation between Cityfinger and the other options, Facebook and XING correlate significantly.

Survey Question 10		Cityfinger	Facebook	XING
Cityfinger	Correlation coefficient	1.000	-.030	-.090
	Significance (2-tailed)		.796	.496
	N	57	54	44
Facebook	Correlation coefficient	-.030	1.000	.421**
	Significance (2-tailed)	.796		.001
	N	54	59	46
XING	Correlation coefficient	-.090	.421**	1.000
	Significance (2-tailed)	.496	.001	
	N	44	46	47

** Correlation is significant at the 0.01 level (2-tailed).

Table 7.8.: Correlations between Cityfinger, Facebook and XING in Survey Question 10. While there is no correlation between Cityfinger and the other options, Facebook and XING correlate significantly.

ing effective. Since there is no correlation between respondents who rated Cityfinger and Facebook (which correlates significantly with XING) highly, it can be concluded that the users who rated Cityfinger highly regarding its control and transparency features gave Facebook and XING low ratings or were unsure. This indicates that Cityfinger's features for user privacy preservation were rated more positively than those of Facebook and XING.

7.3.2. Increased Content Relevancy

This section evaluates Research Question 4 (cf. Section 1.1), i.e. the question of whether transparency and control features lead to more relevant content, as subjectively perceived by recipients. The results have been published in the Proceedings of the 2nd Workshop on Social Recommender Systems (SRS) [86].

Through Survey Question 10, more than half of the respondents (51%) stated that by the available transparency about the audience of information, they feel empowered to communicate more openly to their friends (*fully agree*: 37%, *partially agree*: 14%, average value in the *partially agree* interval).

On Survey Question 11, two-thirds of users (66%) answered that their friends on Cityfinger are mainly real friends (Figure 7.7). Survey Question 12 provided the insight that 72% of users know the majority of group members on Cityfinger in person.

Survey Question 13 showed that the content shared throughout the groups of Cityfinger is, based on the transparency and control features, highly relevant to users, because it comes from real friends (Figure 7.7). More than half of the respondents *agreed* on that question.

The respondents that *agreed* on Survey Question 13 are the same that understood the platform. Table 7.9 lists the significant correlations between the respondents and the relevant questions. As stated earlier, it must be assumed that the fraction of users agreeing on SQ13 can be increased by further improving the basic

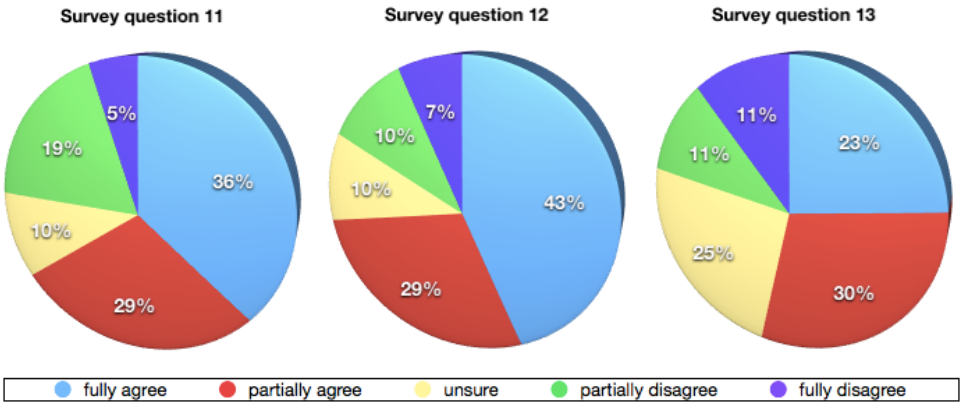

Figure 7.7.: Results to Survey Questions 11-13: Two-thirds of users (66%) answered that their friends on Cityfinger are mainly real friends (SQ11). 72% of users know the majority of group members on Cityfinger in person (SQ12). Based on the transparency and control features, the content shared on Cityfinger is highly relevant to users, because it comes from real friends, 54% of users answered (SQ13).

understandability of the platform.

		SQ4	SQ5	SQ6
SQ13	Correlation coefficient	.293*	.376**	.251*
	Significance (2-tailed)	.010	.001	.024
	N	56	56	56

** Correlation is significant at the 0.01 level (2-tailed).
* Correlation is significant at the 0.05 level (2-tailed).

Table 7.9.: Significant correlations between questions 4–6 related and question 13 (answers to Cityfinger) reveal that users who understood Cityfinger's value proposition, Information Architecture, and Navigation Design also found its content relevant, since it comes from real friends.

The comparison with other SNSs, i.e. Facebook and XING, has analogies with the analysis of the beforementioned section (Table 7.10). Whereas Facebook and XING correlate significantly in a positive manner, Facebook and Cityfinger do not correlate. Interestingly, XING and Cityfinger correlate significantly in a negative manner, which shows that users, who sense a higher relevancy of content on Cityfinger, answered that they do not feel so on XING.

Survey Question 13		Cityfinger	Facebook	XING
Cityfinger	Correlation coefficient	1.000	-.035	-.292*
	Significance (2-tailed)		.764	.032
	N	56	50	38
Facebook	Correlation coefficient	-.035	1.000	.296*
	Significance (2-tailed)	.764		.026
	N	50	55	40
XING	Correlation coefficient	-.292*	.296*	1.000
	Significance (2-tailed)	.032	.026	
	N	38	40	42

* Correlation is significant at the 0.05 level (2-tailed).

Table 7.10.: Correlations between Cityfinger, Facebook and XING in Survey Question 13. While Facebook and XING correlate significantly positively, Facebook and Cityfinger do not correlate. Interestingly, XING and Cityfinger correlate significantly negatively.

7.4. Social Network Analysis

This section analyzes the network graph between users, which emerged through the usage of the platform. Key structural properties are reported and compared to other networks in order to answer Research Question 5 (Section 1.1). Social Network Analysis (SNA) and a selection of its measures were introduced in Section 2.3.1.

Table 7.11 lists the platform's (Cityfinger) high-level statistics and compares them to those of the platforms analyzed by Mislove et al. [107]. The comparison underlines that friendship links play a minor role on Cityfinger. As concluded in Section 8.1.1 in detail, on Cityfinger, friendship links are effectively functionally substituted by group memberships: As friendship links enable access to user profiles on platforms that implement *access based on identifier* (Section 4.5), group memberships are a means of platforms that realized *access based on groups*, e.g. Cityfinger.

Having said that, the following analysis regarding the key structural properties of the network graph compares edges formed by friendship links (i.e. *friends* function; cf. Equation 6.12) with those created by group memberships (i.e. *users* function; cf. Equation 6.14) . If a friendship link is substituted by profile accessibility through a shared group membership, the platform's number of friend links is 16,147 and the average number of friends per user is 86.

Cityfinger's average number of friends, i.e. accessible user profiles, per user (85.89) is comparable to that of Orkut (106.1), the only other SNS per Boyd's definition [31]. Orkut, on the other hand, shows a tenfold number of group memberships compared to Cityfinger. Orkut's high numbers of friendships and group membership might indicate a threat to user privacies through a provider that encourages users to grant access to personal data for a greater information flow.

Table 7.12 compares the networks' average path length, radius and diameter. The values were computed for both edges represented through friendship links as well as through mutual accessibility by group memberships. Cityfinger's graph

	Cityfinger	Flickr	LiveJournal	Orkut	YouTube
Number of users	191	1,846,198	5,284,457	3,072,441	1,157,827
Est. fract. crawled	100.0%	26.9%	95.4%	11.3%	unknown
Month of crawl	Mar 2010	Jan 2007	Dec 2006	Oct 2006	Jan 2007
No. of friend links	120/16,147	22,613,981	77,402,652	223,534,301	4,945,382
Friends per user	0.63/85.89	12.24	16.97	106.1	4.29
Symmetric link %	100.0%	62.0%	73.5%	100.0%	79.1%
Number of groups	130	103,648	7,489,073	8,730,859	30,087
Groups per user	9.82	4.62	21.25	106.44	0.25

Table 7.11.: Cityfinger's high level statistics compared to those of the networks analyzed by Mislove [107]. Cityfinger provides two measures for the amount of friendship links: The first is defined by the *friends* function, the second by the *users* function.

has a significantly higher connectedness compared to those of the other networks[7]. The paths between the nodes are very short. Regardless of whether edges between nodes are defined by friend links or shared group memberships, the average path length differs by a factor of up to 2, and the radius and diameter by 3 and about 2, respectively, from Orkut, which provides the smallest numbers in Mislove's analysis. Although Cityfinger's graph created through friendship links might be too small for comparison, the group membership graph indicates a denser connected community.

Orkut and Cityfinger are the only networks to feature undirected links between users. All other networks have directed connections as they allow users to link to arbitrary users (or Web sites) [107]. However, Mislove reports that the social networks, i.e. all except *Web* (cf. [107]), have a significant degree of symmetry. For comparison, path statistics of the Web have been included. Mislove assumes the obvious differences to be sourced in the high degree of reciprocity within social networks.

Network	Avg. path length	Radius	Diameter
Cityfinger (*friends* function, Section 6.6.2)	2.72	2	5
Cityfinger (*users* function, Section 6.6.2)	1.63	2	4
Flickr	5.67	13	27
LiveJournal	5.88	12	20
Orkut	4.25	6	9
YouTube	5.10	13	21
Web	16.12	475	905

Table 7.12.: Comparison of the SNSs' average path lengths, radiuses and diameters.

Table 7.13 compares the networks' clustering coefficient C. It denotes the probability that a node is connected to its neighbors[8]. Social networks have a natural

[7]Since not all users created friend links or were befriended, the number of nodes is 77 when using the *friends* function.

[8]The clustering coefficient is defined as a node's number of neighbors divided by the number of

explanation for a high coefficient: Users tend to be friends with their friends' friends [107] and are additionally encouraged by SNS providers (cf. Section 3.3).

Network	C
Cityfinger (*friends* function)	0.054
Cityfinger (*users* function)	0.831
Flickr	0.313
LiveJournal	0.330
Orkut	0.171
YouTube	0.136
Web	0.081

Table 7.13.: Comparison of the SNSs' clustering coefficients C.

The low coefficient of Cityfinger using the *friends* function is explained by the low usage of the friendship feature: As said before, only 77 users are connected through, it which makes the probability of connected users low. On the other hand, Cityfinger using the *users* function features a high coefficient of 0.831. As stated in Section 7.1, group memberships tend to follow a Power Law distribution, which was also detected by Mislove regarding the platforms he evaluated [107]. However, as there are some groups with many members, the probability that users are granted mutual access to their profiles becomes very high.

7.5. Privacy Score

This section evaluates the platform's privacy score and compares it to those of the platforms described in Chapter 5.

Transparency (R1): Transparency is conditioned and omnipresent.

Access control (R2): ACE implements *access based on groups*. Access to users is granted by shared group memberships or friendships, access to Tips by group memberships the Tip was assigned to. The revoking of access to the own user profile is achieved by un-friending a user or leaving a group. The latter also revokes access to further users' profiles and assigned content (Tips). Access to any content is *restrictive by default*.

Unauthorized access (R3): To the best of our knowledge, the platform does not allow unauthorized access. Several redundant mechanisms have been applied to audit requests for correctness. No mechanisms for crawler detection have been implemented.

Panoptic Provider (R4): The platform's provider has access to all user data.

DHPs (R5): The platform does not support any user-configurable Data Handling Policies.

possible connections. The clustering coefficient of a graph is the average coefficient of all its nodes.

User model control (R6): Users can edit their user model by editing their profile page and by joining or leaving groups. No further data about the user is stored.

Friendships (R7): The platform supports the befriending of users and utilizes this for mutual profile page accessibility.

Groups (R8): Groups are provided and allow users to create and control Contextual Integrity by limiting accessibility to content to its members. Groups are used and promoted as a means for access control.

Data export (R9): No synchronization functionality is provided. For data export, Microformats were implemented for exporting users and events in standard formats.

Account deletion (R10): Accounts are deletable, including directly related data. The presence of the link to this feature is sufficient.

Purpose limitation (R11): The platform collects a minimum amount of data. No data is exchanged or sold to third parties.

Data security (R12): Data is secured on a standard level: Access control mechanisms and XSRF protection were implemented to prohibit unauthorized access. Data is backed up, but not encrypted.

Minimize data collection (R13): No data retention limitation is specified in the privacy policies. Data of deleted accounts is erased.

Data ownership (R14): Data ownership is not explicitly attributed to the user.

National laws (R15): The platform does not differentiate between the privacy laws of the countries of its users.

Table 7.14 evaluates the platform's fulfillment of the requirements and compares them to those of the platforms evaluated in Table 5.4. A key for the symbols and scores can be found in Table 5.3.

Although the platforms are not comparable due to a lack of knowledge of the requirement fulfillment quality, the table gives a rough estimate of the respective platform's areas of quality (as discussed in Chapter 5). Cityfinger performs well in the cluster of transparency and access control where it reaches almost three times the score of the best competitor. In the remaining clusters it outperforms the other SNSs, but not at the same significance (cf. Figure 7.8).

However, as the maximum possible score is 45, there is still room for improvements in future work.

7.6. Summary

This chapter evaluated the platform regarding its effectiveness and efficiency to protect user privacies, and compared it to other SNS platforms.

The user survey showed the effectiveness and efficiency of the applied mechanisms for transparency and access control. The respondents' answers showed that

Requirement		Facebook	VZ	XING	LinkedIn	Cityfinger
Transparency	R1	○	−	−	−	++
Access control	R2	○	−	○	○	++
Unauthorized access	R3	○	○	○	○	+
Panoptic Provider	R4	−	−	−	−	−
DHPs	R5	−	−	−	−	−
User model control	R6	○	○	○	○	++
Transp. and Access Ctrl. sub-total		4	2	3	3	11
Friendships	R7	+	○	○	○	+
Groups	R8	+	○	○	○	++
Relationships sub-total		4	2	2	2	5
Data export	R9	○	−	○	○	○
Account deletion	R10	−	++	+	−	++
Identity Management sub-total		1	3	3	1	4
Purpose limitation	R11	−	−	−	−	++
Data security	R12	○	○	−	○	○
Minimize data collection	R13	−	++	○	−	+
Data ownership	R14	−	−	−	−	−
National laws	R15	○	○	○	○	−
Data Handling sub-total		2	5	2	2	6
Total score		11	12	10	8	26

Table 7.14.: Privacy ratings and scores of the platform compared to those of the SNSs evaluated in Chapter 5. The platform scored more than twice as many points as the best of the remaining SNSs.

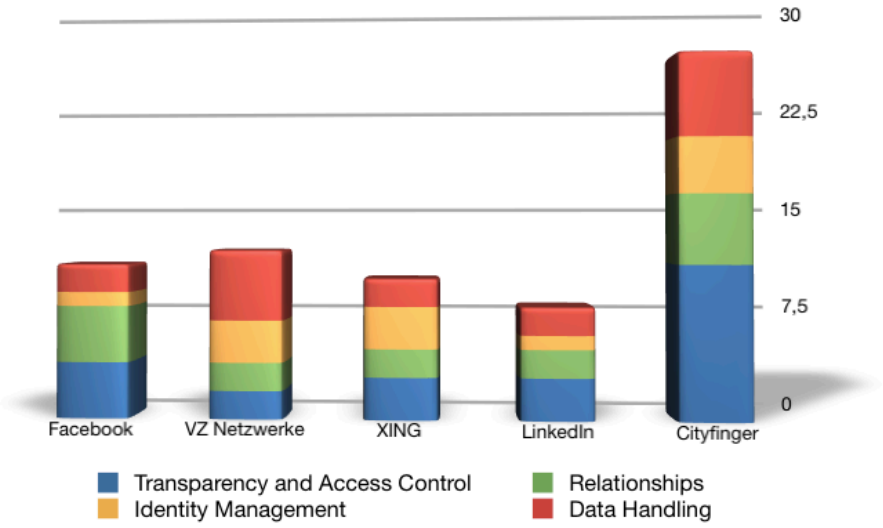

Figure 7.8.: Privacy score of the platform compared to those of the SNSs evaluated in Chapter 5. The platform gained two times the points compared to the other SNSs, most of which from the Transparency and Access Control Cluster.

the platform's value proposition was understood, how it is used and how information is organized and addressed. The majority of users knew how to control the accessibility to their information and half of them understood the features to make this transparent. Half of the users felt the opportunity to communicate more openly through the given tools for privacy protection.

We were able to show that users who rated our platform's transparency and control features highly, rated those of other SNS platforms low. This proves better privacy protection of the platform compared to the competitors. Furthermore, we were able to show that through the given privacy protection, content and users subjectively appeared more relevant to respondents.

The evaluation of the emerged social network graph proved that our platform's graph shows some interesting differences regarding its key structural properties compared to those of other platforms. Although the average number of friends was almost equal for Cityfinger compared with Orkut, our platform's path lengths between nodes were shorter by at least half, which proves its higher connectivity.

To compare our platform with others regarding its privacy score, its requirement fulfillment quality was evaluated and rated. The scores gained in the requirements cluster of transparency and access control reached more than twice the score of the best opponent. This backs the results found in the user survey discussion.

8. Discussion

The previous chapter has evaluated our platform approach in a user survey and discussed its emerged social network graph' key structural properties with those of other platforms. It also rated the approach concerning its fulfillment quality of the privacy requirements defined in Chapter 5.

Looking back, this chapter discusses this thesis regarding its two major contributions: The requirements defined for privacy protection and the presented platform approach including its evaluation. Starting with a report of findings and insights concerning the approach, a number of improvements is introduced in order to overcome problems or improve its functioning. Afterwards, future work is suggested for further research in this area to approach the greater scope, i.e. the requirements for privacy protection.

8.1. Findings of the Platform

This section reports of findings observed during the analysis of the application's traffic, corpus and user activity.

8.1.1. Friendship

As described in Section 6.1, an access control mechanism based on groups (Families) was chosen as the basic ACP. The evaluation (Chapter 7) proved that the concept worked as users understood how to control and overview access to their content. Most other SNS implemented *access based on identifier* (cf. Section 4.5). Thus, access is granted by whether a friendship link exists between users. Nevertheless, we introduced a friendship feature that grants mutual access to user profiles (Feature 10, Section 6.4).

The feature was barely used (192 Users created only 120 Friendships; cf. Sections 7.1 and 7.4), because it is almost redundant and thus, has no benefit: Although a friendship effectively does grant mutual user profile access even if no common Family memberships exist, users can only befriend one another if they already are granted profile access through at least one common Family membership. Benefits of a friendship are a user list filtering feature by friends and a graphical highlighting of friends in lists of users.

8.1.2. Perishing Activity

For every application, it is essential to remain interesting to its users over time. For social applications, two major factors are relevancy and activity. Since relevancy is implicitly given by social connections between users (content and recommendations by friends are highly relevant) [86], social applications focus on motivating user

activity and on distributing their content. Optimally, the stream of news and activities never stops and is updated every time a user returns. Only by achieving this does an application remain appealing to users over a longer period of time. For social applications that distribute UGC, the task is thus to, on one hand, activate users to participate and to contribute, and on the other, to maximize the number of receivers for content distribution.

Concerning user-activation, there are rules to follow and tools to apply. Nielson's 90-9-1 rule was introduced in Section 2.2.1 to explain different types of users regarding their contribution behavior. It makes clear that users who contribute content are more an exception than norm, i.e. ca. 10%. Game Mechanics were introduced as a means to drive user activity through atomic mechanics we know from games (Section 2.2.3; Feature 20, Section 6.4). Another tool for user activation is a notification mechanism (Feature 18, Section 6.4). Triggered by user activity, notifications are sent out to friends to lead them back to the platform in order to gain traffic and to motivate the contribution of a reaction, e.g. a comment. Notifications are usually sent out both, platform-internally and externally depending on the particular type of activity. On many platforms, user activities are additionally conditioned as a stream of news in chronological order (cf. Feature 19, Section 6.4).

However, the main factor to maximize content distribution is the amount of contacts a user has. The more—irrespective of whether these are friendships or group memberships—the higher the number of potential recipients for a piece of information that is produced by that user. The more recipients there are for Nielson's rare 10% of users, the earlier the amount of contributions results in a stream of news that is sufficient to keep the platform interesting to its users. If the stream of news about platform activity is insufficient, users will stop returning to the platform sooner or later. Since editorial creation of content is impossible by the design of the application presented, the above named aspects are the only options a provider has.

The work introduced, implemented and evaluated a novel approach to preserve users' privacies *by design*. It forces users to assign self-generated content to groups, i.e. Tips to Families. Families represent Social Contexts and provide both, transparency and control over audience and reach. This does empower users to control their privacy, but also forces them to configure receivers for their content (by selecting Families for Tip assignments) every time. It is not possible to publicly publish content.

Chapter 7 proved the concept to be effective: Users understood it and their privacies were preserved. Content was reported to be more relevant, since it was produced by real friends. However, the downside is that user activity could not be maintained over time. The limitation of content flow to relatively small groups and thus small amounts of receivers artificially handicapped the achievement of a Tipping Point [55] from where groups manage to produce enough content by themselves to provide a sufficient application activity. Ways to overcome this problem are discussed in the following section.

8.2. Improvements for the Platform

This section lists suggestions for improving the application with respect to reported findings.

8.2.1. Tipping Point

Every social application has a Tipping Point that needs to be achieved in order to produce a sufficient activity by itself. It is assumed that the ease of reaching the Tipping Point correlates with the opportunities for distributing content. That means that the wider UGC can be spread, the easier it is to reach the Tipping Point. Without a doubt, the number of users is too small (cf. Section 7.1). More users should be acquired in order to observe the problem of perishing activity at greater usage numbers.

A related problem is the fact that users are *by design* forced to invite new application users not only to join the application, but also selected Families inside of it. This, on one hand, improves content and user relevancy, but, on the other hand, complicates the invitation process (Feature 15, Section 6.4). This feature should be improved to solve this issue by both, simplification and higher rewards via Game Mechanics. Currently, users only gain points for creating invitations. Graphical badges or decorations that are placed on user profiles could affect invitation behavior.

8.2.2. Power Law

As Section 7.1 reported, it is expected that content contribution behavior follows a Power Law distribution, i.e. very few users contribute the vast majority of content (power users). A measure for the ease of use of a feature is the flatness of a Power Law curve: The easier a feature is, the bigger is the exponent of its Power Law formula. Plots comparing Power Laws, e.g. Figure 7.2, can thus be used to prove if one feature is easier to use than another. Also, usability problems can be detected if two equally easy features show significantly different exponents.

Concerning the creation of Tips and Families, a user survey should be applied to clarify if users found their Social Contexts already configured and secret tips already added by other users, keeping them from creating further Families and Tips.

8.2.3. Game Mechanics

In general, the applied Game Mechanics (Feature 20, Section 6.4) are too few and offer too little differentiation. Users earn points for key activities, e.g. inviting friends and creating content. User profiles expose the amount of points a user has and user lists can be sorted by points. As explained in Section 2.2.3, there are more mechanics that could be applied in order to activate users. E.g., graphical badges could be awarded to users and then added to the activity stream to notify others to motivate contribution.

8.2.4. Featuring Openness

Both activity and growing the user base can be improved by featuring open standards and protocols (Section 4.7). For instance, the registration process can be simplified by supporting OpenID authentication. Users could use their, e.g., Google Account to register for the application.

Open standards could be applied to the notification mechanism to spread user activity to other SNSs, e.g. Facebook's NewsFeed, or Twitter. Also vice versa, other SNSs' content can be imported to populate activity streams.

Technically, it is nowadays possible to port particular functionality to an SNS if it provides an application platform, e.g. Facebook Platform or Open Social. For instance, a Facebook App could be introduced that allows users with a Cityfinger account to create Tips on Facebook which are also synchronized back to Cityfinger. Doing so, Facebook friends of the author can also be notified via Facebook and led to Cityfinger. This increases traffic and the user base.

8.2.5. Mobile Usage

To serve features to users while they are on the go, a mobile client could be developed. The mobile creation of location-based Tips is even more natural, since the user is in a location-based context and ad-hoc use-case for content creation. This simplifies the Tip creation process, since localization can be assisted by an automated determination of to the user's current position. Also, further appealing ad-hoc use-cases can be realized, e.g. the notification of friends about the current position in order to convince them to come by.

8.2.6. Public Content

To increase content distribution, public Families could be provided, publicly listed and open for every user to join. While being promising for maximizing content distribution, this feature also can help to solve the problem of the Blank Slate State[1]. Public listings of open Families is content that is displayable to new users, and could free inviting users from being forced to choose Families to invite someone to. Unfortunately, new users are at risk of misunderstanding the application's purpose, since Families and content consequently become less relevant. This feature was omitted during the platform conception phase to ease and make comprehending the application's purpose more intuitive. Care must be taken to ensure that intuitiveness does not suffer from this feature.

8.3. Future Work

As pointed out many times throughout this work, provider interests and user interests are contradictory. Providers seek to maximize content distribution and

[1]The Blank Slate State describes the problem of introducing newly registered users to an application. The goal is to avoid to displaying an empty page due to the lack of social connections and thus, content. Cityfinger circumvented the problem by forcing inviting users to choose Families to assign a new user to on arrival (Feature 15, Section 6.4). Other applications help new users by, e.g., finding first friends to socialize with.

the Network Effect, whereas users want to socialize, but preserve their privacy at the same time. Figure 8.1 depicts a two-dimensional, four quarters grid between provider and user interests.

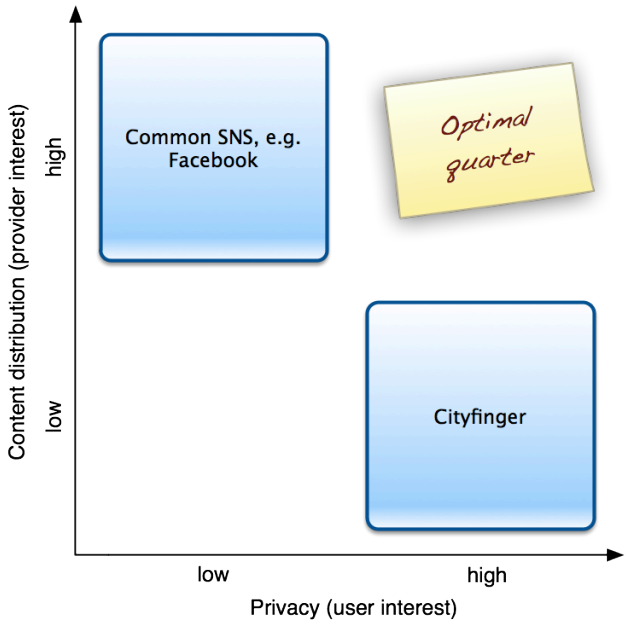

Figure 8.1.: Four quarters grid of provider versus user interests.

Although this work has managed to create an SNS to preserve its users' privacies, not all problems reported in Chapter 3 were addressed and solved.

Future work must provide a *usable* privacy-preserving application that at the same time allows its provider to spread user content more widely (given their consent). The presented application circumvented the problem of usable privacy controls by preserving privacy *by design*, i.e. by its ACP. Future works should provide greater flexibility regarding privacy controls, while remaining usable.

8.3.1. Individual Groups

A promising concept is to allow users to define *individual groups* of receivers with respect to *what* they intend to publish. Such group definitions provide Social Contexts and can be stored, maintained and re-used to simplify future configuration of receivers for content. The concept of individual groups would also solve a minor issue of the presented application: Commonly used groups are diluted regarding the Social Contexts of individuals if more than one user adds receivers. Once they have defined individual groups (a graphical representation of users could assist users to configure these Social Contexts), users can start publishing information and control receivers by choosing one or more individual groups.

8.3.2. Pareto Optimum

In order to find optimal application solutions to cover both user and provider interest, the problem could be modeled in order to detect Pareto optima. Assumed that an n-tuple $x = (x_1, x_2, ..., x_n)$ describes the characteristics of a particular SNS. Its n dimensions represent features that affect the application to be more privacy-preserving or content-distribution-maximizing. Thus, providers can control the position of their application within the n-dimensional space by changing these features. In order to improve the application, providers can perform the sensoring application-internally, e.g. by traffic analysis or user surveys, or across applications by being influenced by how competitors shaped their applications. The process is iterative: Providers sensor user-reactions or perform a user-evaluation on their application, perform an action in terms of changing their vector, and sense again.

The approach can be phased in three steps of complexity for modeling user preferences:

1. All users are assumed to be the same, e.g. Privacy Pragmatists. They all have exactly the same preferences.

2. Users have different preferences, but act and decide independently from others.

3. Users have different preferences, and influence each other. They are biased by what friends do, by peer pressure, and on action and reaction on an application.

This chapter presented findings and insights observed on the application. It introduced a list of suggestions for improvement. Future work was recommended to approach the problem of conflicting interests of providers and users.

9. Outlook

As the previous chapter discussed this thesis' contributions from a retrospective point of view, a long-term perspective of social applications and social data on the Web is outlined in this chapter. Two visions outline what can be expected from the future of this field. Firstly, the vision of the Social Web is explained, standing for opening up and linking of social data and functionality. Secondly, the technical interconnection and interplay of data and service sources is outlined. A vision is sketched of a future generation of applications that are empowered by factual knowledge, people data and a pool of services.

9.1. The Social Web

The contemporary market of social applications is heterogenous and insufficiently interoperable. Service providers focus on maximizing their user-bases and content-bases (cf. Data-drivenness; Section 2.2.1). The metaphor of *walled gardens* describes how the landscape of applications is perceived by users: It is costly for users to change from one garden (application) to another. When signing up for a new application, users have to re-enter personal information, preferences and relationships to others. Synchronization and export features for user data and friend definitions are missing or poorly implemented. Interoperability is only provided if it is beneficial to provider's goals.

However, user needs for, e.g., service interoperability, privacy and data ownership will force providers to remain meaningful to their customers by providing solutions for their demands. Open standards (Section 4.7) are developed and adopted by providers which, in the long term, will induce more and more interoperability between applications.

Another trend is the separation of identity management and social functionality. There are many reasons driving this trend: As some providers re-use their ID management functionality for different social applications, users are also becoming less willing to create new accounts for every application they use. Providers start lowering this burden by providing SSO mechanisms. Some even do not implement their own authorization, authentication and accounting (AAA) functionality for their application any more. They can concentrate fully on their application.

However, when allowing users from different Identity Providers (IDP) to log on to an application, the market is forced to become interoperable. In the long-term, IDPs will emerge and provide a similar feature set to its customers. Users will choose one (or more) ID providers whom they trust to host their identity and social data.

IDPs will host the following data for users.

- identity (the user's person)

- personae (users must be able to differentiate between multiple personae, e.g. a private and a business persona)
  - profile data
  - social graph,
  - groups, lists
  - live-stream, instant messaging and other live data, messages
  - files, assets
  - reviews, comments, ratings, feedback
  - presence information, e.g. location, mood

They will provide functionality for data management, privacy, access control, security, trust, AAA etc. Given the permission of the user, social applications can read and write the required pieces of this data from and to the IDP.

In the long term, this leads to an interwoven Web of content and social functionality: The Social Web. The border between the quasi-static, uni-directional Web and social applications will vanish.

9.2. A Trilogy of Web for Machines

In the coming years, we will see a revolution of machine knowledge and abilities which will emerge from different activities and trends in three distinct areas connected to the Internet: The emerging *Web of Data* and *Web of Services*, as well as the *Web of Identities*. These areas are about making accessible, connectable and processable semantic knowledge about data, functions and individuals, respectively. This section introduces these webs and sketches visionary scenarios that exploit the aggregate knowledge of them as well as their impact on the future of the Internet. It has been published in the Proceedings of the Workshop on Linked Data in the Future Internet at the Future Internet Assembly (LinkedDataFIA) [85] and as a four-part article series at ReadWriteWeb [79, 80, 81, 82].

The amount of available information is growing exponentially. Sources are developed or made accessible, content is produced *en masse* through the paradigm shift of consumers becoming producers in the Web 2.0 (Section 2.2.1), users leave data about themselves and their social connections, companies are making useful services accessible.

Every day, it is becoming more complex and difficult to make information findable and usable. In the near future, e.g., a search engine's one-dimensional list of ten results returned on the first page for a query will not be sufficient to cover our needs. Users will demand more sophisticated, natural queries. To answer those, today's search engines' primitive semantic understanding of the content they index is not sufficient as a knowledge base.

It is indispensable that machines are taken to another level of understanding: Understanding what terms are about and semantically connected to. Understanding what services do and successively invoking them to generate further or process existing information, and finally, understanding the user. That is, both understanding an acute need, such as a query, as well as having access to profile information to understand who a user is and what he or she likes, given access authorization.

An answer is sketched to the challenge of providing tomorrow's machines with a toolbox to find and interpret knowledge, to discover, orchestrate, and invoke services with the knowledge gained to solve highly complex tasks.

The coming section discusses selected activities in the areas of the Web of Data, the Web of Services and the Web of Identities and illustrates how these interrelate and form an emerging big picture of a trilogy of webs for machines.

9.2.1. The Webs

Current research activities focus on making semantic knowledge from open data sets(*Web of Data*) accessible and making semantically annotated services (*Web of Services*) accessible to machines. Emerging within the Social Web (Section 9.1), we want to introduce the *Web of Identities*. This web is about making user data machine-accessible: their assets, facts, preferences, social graphs, etc. This content is highly privacy-sensitive, volatile and valuable. Different user-centric access control mechanisms will be needed compared to the Web of Data.

Web of Data

The idea of the Web of Data [79] originated within the Semantic Web [18] (Section 2.3.2). The inability of machines to understand Web pages to a significant extent led to several initiatives to overcome this weakness. Initially, the aim of the Semantic Web was to invisibly annotate Web pages with a set of meta-attributes and categories in order to enable machines to interpret parts of the knowledge included in the text and to put it into some kind of context. This approach did not succeed, since the annotation was too complicated for humans with no technical background. As described in Section 4.7.5, current markup-based approaches, such as Microformats and RDFa, specify attributes to express structured data in any markup language, follow similar ideas of making the markup process easier and thereby allowing more users to participate and thus to overcome the cold-start problem. On the other end of the technology spectrum, we find full-blown ontologies that describe domain knowledge with the help of formal logic. This allows inference of new information from a set of facts, but is difficult to do with the right level of detail.

All these approaches have in common that they try to improve the machine-readability of Web pages that are designed for humans. But the horizon or depth of machine-readable knowledge that can be added to a page is limited: only the page itself and particular elements on it can be marked-up by applying these approaches.

This limitation and the fact that there are already existing data sets containing lots of structured data about all kinds of information distributed over the world lead to the idea of creating a dedicated Web of Data. If these data sets are semantically described and interconnected, a machine can traverse through this web to gather semantic knowledge about arbitrary entities and domains, independent of the information contained in the original Web page.

A promising approach is the W3C SWEO Linking Open Data community project[1] [20] (LOD). The project is uncoupled from the Web for humans and interconnects ex-

[1] `http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`, accessed December 2008

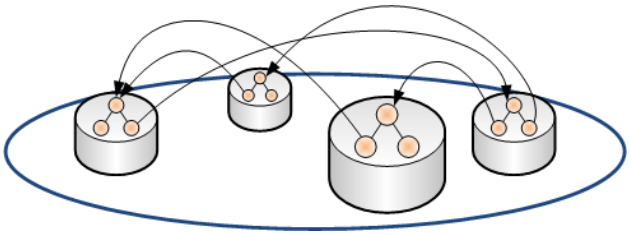

Figure 9.1.: Web of Data: A distributed web of semantic interconnected data sets containing general knowledge. The data sets each consist of a triple store containing structured knowledge which may link to knowledge contained in the same or another data set.

isting open data sets. Figure 9.2 illustrates the current state of connected data sets. The data sets contribute by granting access to their semantically linked knowledge and by linking to items of other data sets. This way, the project follows basic design principles of the World Wide Web [17], e.g. simplicity, tolerance, modular design and decentralization. The LOD project currently counts more than 2 billion[2] RDF triples. The LOD data sets can be accessed in various ways, e.g., by a Semantic Web browser or crawled by a spider of a semantic search engine.

With every fact and link added to the Web of Data, more general and specific knowledge is made accessible to machines. The Web of Data will enable a whole new generation of services. Through the semantic structuring of the data within the data sets and the interconnection of lots of different data sets, highly sophisticated queries become machine-processable and can be answered through a next generation of search services. Querying languages, such as SPARQL [122] and RQL [74], are already available.

Web of Services

The services sector has become the world's biggest business sector, forming 64% of the world-wide gross domestic product [68]. This sector is under pressure to make their services easier and more widely accessible, as well as to adapt to ever faster changes in the market environment.

The Service-Oriented Architecture (SOA) paradigm has become the predominant approach to (enterprise) software engineering, to streamline the IT infrastructure within an organization as well as to interact with external entities. Its principles [46] call for services that have their formally described interface de-coupled from its functionality and described in an abstract fashion. While SOA can be implemented with a host of different technologies, Web Services have become the technology of choice.

However, it should be noted here that, while services are all the rage, there is no clear definition as to what constitutes a service, neither on a technological nor conceptual level. In the former case, Web Services offer a quasi-standard, but the subset of standards that is agreed upon is neither powerful nor expressive enough to handle the possible applications of services. Also, the standards so far lack any

[2] `http://www.w3.org/2008/Talks/WWW2008-W3CTrack-LOD.pdf`, accessed December 2008

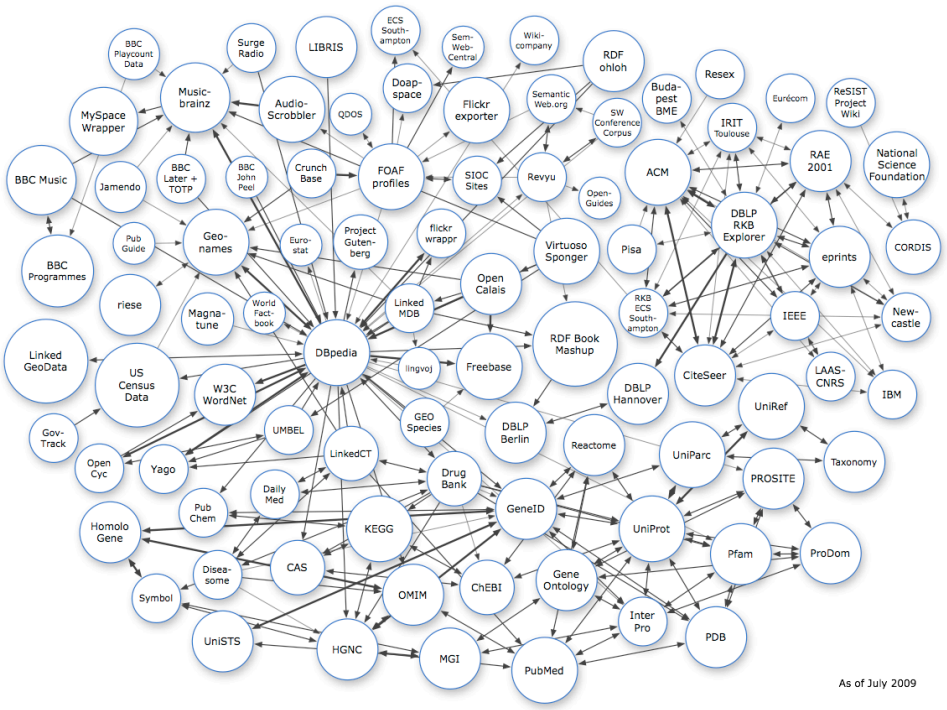

Figure 9.2.: Data sets of the Linking Open Data project. The circles are data sets containing knowledge covering different domains. The thickness of the arrows illustrates a measure of bi-directional connectedness. Image taken from LOD project.

semantic description, making its use within automated scenarios all but impossible. On a conceptual level, the situation is possibly worse as it is by no means clear what actually constitutes a service once we move out from the narrow definition of Web Services towards higher-level services.

Today, there are already all kinds of services at all levels of complexity on the Web and their number is expected to grow exponentially. The services follow different standards and a lot of them are proprietary, uni-directional and designed to be used by humans to mash-up something new from them. There are editorial catalogs, e.g. ProgrammableWeb[3], designed for humans searching for a particular service. A lot of Web 2.0 services provide services to read existing or create new data, exposing almost all of their functionality through their API[4]. There are even human-based services, such as Amazon's Mechanical Turk[5] product. As mentioned above, Web Services follow an agreed-upon standard concerning the service definition, but currently lack a semantic description. While there are a number of different approaches to adding a semantic description to Web Services, such as OWL-S [100], WSMO [50], or WSDL-S [7], none has so far managed to break out of its academic confinement.

Once services are annotated semantically, they can be accessed by machines, automating service discovery, execution, billing or revenue sharing, orchestration, replacement on failure based on experience (Quality of Service, QoS) etc. These services will be brought together in a Web of Services [81] according to Web principles.

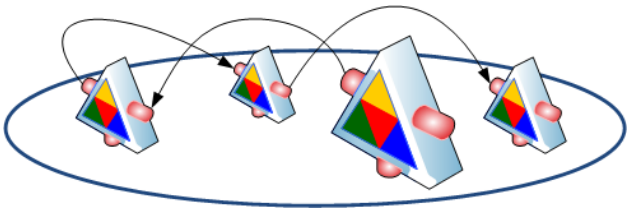

Figure 9.3.: Web of Services: A web of distributed, semantically annotated services are freely accessible to be, e.g., discovered, invoked, orchestrated or chained. The arrows depict a potential service chain, which might be discovered and invoked by an orchestrator.

Many works in this area deal with the topics Internet of Services and SOA in general, in research as well as industry. Closest to the concept is the SOA4All project[6]. It addresses its addressed issues through four cornerstones [42]: Firstly, Web principles (which we already noted in the context of the Web of Data) and Web technology as the underlying infrastructure are used in the Web of Services. Secondly, they plan to implement user participation in terms of, e.g., ranking of services. Thirdly, they want to facilitate Semantic Web technology to abstract from

[3] `http://www.programmableweb.com`, accessed December 2008

[4] In 2007, Twitter counted ten times more traffic on their API than on their Website (`http://www.readwritetalk.com/2007/09/05/biz-stone-co-founder-twitter`, accessed December 2008)

[5] `http://www.mturk.com`, accessed December 2008

[6] `http://www.soa4all.eu`, accessed December 2008

syntax to semantics to grant machines knowledge about the services. Last, but not least, they plan to implement a context management to enable processing of user requirements when it comes to service contracting or orchestration.

The TripCom project[7] concerns itself with the design and implementation of an architecture for application integration based on the combination of Semantic Web, Web Services, and tuple spaces, called the triple space service technology. There, services can persistently publish semantically annotated data in order to facilitate orchestration and choreography of services.

The SUPER project[8] focuses on elevating business processes from the IT to the business level. To do so, appropriate ontologies are defined and services are semantically annotated to allow the context-aware automated integration of services in business processes.

The SHAPE project[9] provides a unified approach to the definition of semantically enhanced SOA. The focus lies on the integration of model-driven approaches with semantics and SOA.

Somewhat at odds with the development of services is the large research area of Multi Agent Systems [149]. There, similar to SOA, distributed agents communicate and cooperate to achieve some goal. However, while services are generally considered passive, agents are autonomous in the sense of having the ability to make their own decisions, and proactive. Given some semantically described goal, an agent tries to bring about a situation where the goal holds true. He does this by interacting with other agents, cooperating with them to change the state of the world. In the context of the Web of Services, agents play a vital role in that they, at least in the realm of academia, already created a Web of Services, where machines, i.e. agents, autonomously searched for functionality and used different services based on their semantically described capabilities. In 2003, the Agentcities [41] project tried to create a global, open, heterogeneous network of agent platforms and services to which any agent researcher could connect their agents. Services could automatically be offered and used.

The Web of Services will enable machines to work with a huge toolbox of functionalities. Services might answer queries (from humans or other services), or create further knowledge which could also flow back to the Web of Data. Automated service orchestration and service chaining will be an important tool to quicken innovation cycles.

Web of Identities

As discussed in Section 9.1, within the Social Web there is a trend of splitting ID management from social features, and a market of similar IDPs is emerging. When IDPs and social applications become interoperable, it is a logical next step that IDPs are implicitly connected through friendship links of users of different IDPs. Thus, one can understand this emerging infrastructure as an analogy to the Web of Data for making machine-accessible people data: a Web of Identities [80].

As the Web of Data and the Web of Services, the Web of Identities should follow basic Web principles. It is essential that IDPs emerge from both sides, bottom-up,

[7] `http://www.tripcom.org`, accessed December 2008
[8] `http://www.ip-super.org`, accessed December 2008
[9] `http://www.shape-project.eu`, accessed December 2008

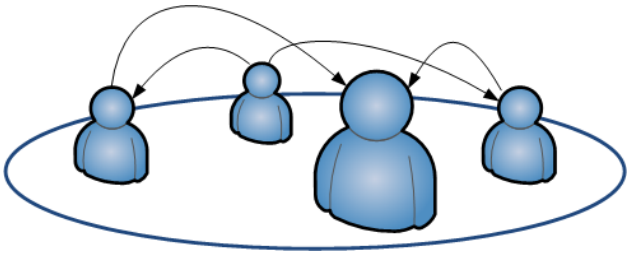

Figure 9.4.: Web of Identities: A distributed web of identity providers managing the identity, personae, social graphs and assets of their customers. The arrows exemplify identity linkage through the social graphs of the customers.

e.g. a solution developed from scratch, and top-down, e.g. the big players, such as Google or Facebook, opening up more and more, can converge in an interoperable Web of Identities, i.e. IDPs, supporting a common set of standards. That way, every user can decide which IDP to trust and to choose as his personal identity provider.

There are bottom-up as well as top-down approaches, some are driven by commercial interests and some are non-profit. All of them are to a certain extent coherent with the named vision. All bottom-up approaches have in common that they combine standards for protocols and exchange formats to a greater functionality block (cf. Section 4.7) ending up in an IDP.

The big picture for all bottom-up approaches is drawn by Marc Canter by his Open Mesh[10]. Canter outlines a vision of what building blocks are needed and how they could be put together in a common infrastructure.

The non-profit DataPortability[11] group deals with the establishment of open standards and protocols for the exchange of data between applications and vendors. The protocols and standards are already widely agreed upon and now need to be further diffused and adopted. The development of open, non-proprietary specifications for Web technologies is also the dedication of the non-profit Open Web Foundation[12]. A similar approach is taken by non-profit Identity Commons[13]. They focus on the users' identities and social graphs. A distributed initiative of providing SNS building block functionality is taken by the non-profit DiSo Project[14] (Distributed Social Networks). The team implements a WordPress[15] plugin that implements some of the standards supported by the DataPortability group.

A top-down approach is the EU-funded project PrimeLife[16]. PrimeLife is aimed at bringing life-long privacy and user-control over personal information and autonomy to the Information Society. The project at its current state is promising to

[10] `http://blog.broadbandmechanics.com/2008/05/how-to-build-the-open-mesh`, accessed December 2008
[11] `http://www.dataportability.org`, accessed December 2008
[12] `http://www.openwebfoundation.org`, accessed December 2008
[13] `http://www.identitycommons.net`, accessed December 2008
[14] `http://www.diso-project.org`, accessed December 2008
[15] `http://www.wordpress.com`
[16] `http://www.primelife.eu`, accessed December 2008

end up as an IDP implementation.

Last, but not least, Google, Yahoo, Facebook, Microsoft and MySpace are opening up slowly (cf. Section 9.1) and will become IDPs themselves. Features, such as Google's Friend Connect[17], Facebook's Connect[18] and MySpace's ID[19], are all aimed at spreading a fragment of the platform's features beyond the platform itself. That serves the users' needs of accessing locked-in data for a transitional time, but does not solve the problem of data ownership (Chapter 3). Microsoft's Live Mesh allows its users to synchronize files across devices and platforms. Yahoo! opened up through the exposure of lots of services according to its Y!OS[20] (Yahoo! Open Strategy).

However, research has to be done in the areas of empowering the user to take control of his data. Features, such as reach control, revokable access rights and the management of what third party service can read or write what fragment of which persona's data, are necessary, but very hard to translate to an intuitive user interface and user experience.

If this vision comes true, we will see a user centric, user friendly, privacy-preserving and meaningful tool. Users can explicitly grant online marketers access rights to attention data or purchasing history data to empower them to target meaningful ads that may take into account what direct friends recommend.

For both machines and applications for the emerging Social Web, the Web of Identities is a very important infrastructure for looking up user-related private, volatile personal and contextual data.

9.2.2. Interplay of the Webs

With the named trilogy of webs for machines as a backbone, the Internet as a tool will change, because interconnected knowledge and toolsets catapult machines to a new level of ability. New services will emerge, based on the foundation of the webs. Also, from the Human-Computer Interaction (HCI) side, the way we use existing services will change dramatically.

The distinction between data items stored and retrieved and the use of services will continue to blur until requests will freely traverse the webs, retrieving items of data, feeding chains of services that use personal information from the stored identities.

The following example scenarios give an impression of how applications can utilize and interlink the webs.

Scenario: Social Recommendations. Bob visits a search engine application that is based on the Webs. He queries *Recommend books about Berlin for my mother for Christmas.* The Natural Language Processing (NLP) component of the application analyzes his query and splits it up to a chain of subtasks, which the application now starts to process: From the Web of Data, the application gathers general knowledge about the terms *mother*, *Berlin*, *Christmas.* It

[17] `http://www.google.com/friendconnect`, accessed December 2008
[18] `developers.facebook.com/connect.php`, accessed December 2008
[19] `developer.myspace.com/community/myspace/dataAvailability.aspx`, accessed December 2008
[20] `developer.yahoo.com/yos`, accessed December 2008

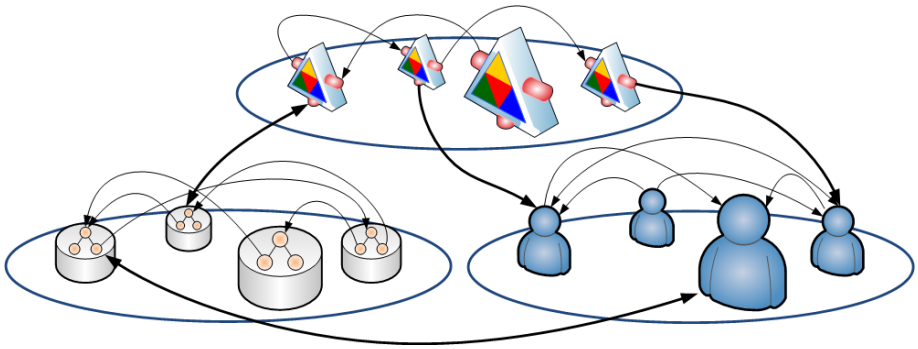

Figure 9.5.: The trilogy and interplay of the Webs. The thick arrows illustrate the data, services and identities refering to each other. That said, a service can look up user data, a user can refer to encyclopedic knowledge about facts from within his profile.

proceeds by querying a service that indexed the Web of Data for all books covering Berlin or authors born or living in Berlin. Given permission from the user, his IDP is called to return his mother's identity URI from his social graph. His IDP searches all of his personae for his mother and finds her in his private persona's social graph. The mother's IDP is called to access her interest information limited to the wider topic fields *books* and *Berlin*. The mother's personae's social graphs are searched for her back-link to Peter. As the private persona is found, that persona's information is selected for access limitation. From the private persona, the mother's IDP returns a set of information the mother explicitly granted access to. The set contains general interests, some purchases, reviews, comments, ratings and some attention data. URIs of the mother's private friends are also returned. The application continues by querying the mother's closest friends' IDPs if one of them liked or recommends books about Berlin, since friends' recommendations are the most valuable. The application identifies the term *recommend* as a service request term and searches the Web of Services for a recommendation service that can handle books, personal interests and recommendations as filtering and ranking criteria. The initial set of books the application retrieved from the Web of Data and the information collected from the Web of Identities is now sent to a filtering and ranking service. As the term *Christmas* is recognized as contextual term for the task, the application now searches the Web of Services for e-commerce services offering books. The filtered and ranked list of books is sent there to retrieve price proposals with a delivery date before December 24th. Finally, the list of books is augmented by prices and dates and presented to the user. The application tracks feedback for the book recommendations and assigns it to its QoS ratings for the services invoked.

Scenario: Mass Customization. Alice recently graduated from university. She knows that she needs an insurance package, but has no idea what exactly it should consist of. She heard of this intelligent insurance packaging brokerage application which she now visits with her browser. She logs into the applica-

tion with her ID. From the Web of Identities and given her permission, the application initiates a profile lookup at her IDP to gather information needed for the configuration of the components of the insurance package. It queries for information, such as private address, marriage status, age and gender. Since it cannot find her current income, it prompts her directly. From the Web of Data, the application now queries for her neighborhood's crime statistics for risk estimates. The application now looks up all insurance services it can find in the Web of Services. It configures the services with the knowledge gathered, selects the best offers and combines them to a personalized insurance package. The package consists of products from different insurers around the world. She signs the contracts through the broker and logs out with the satisfaction that she now is optimally, i.e. neither under- nor overinsured.

Belief, Desire, Intention

If you will, the Webs can be compared to the BDI (belief, desire, intention) model of Rao and Georgeff [123] that describes a formal model of the mental state of software agents. They describe the mental state of a single agent with the help of an (incomplete) model of the world, i.e. beliefs, a set of plans, i.e. possible courses of actions, or intentions) and a set of goals, i.e. desired states of the world. On a global scale, the Web of Data represents the belief state of the world, while the services and their composition provide possible courses of actions, and identities contain goals and desired states. In this reading, the Webs of Data, Services, and Identities indeed move the world a bit closer to machine understanding.

9.2.3. Conclusion

We have outlined the concepts of the Web of Data and the Web of Services, and introduced the Web of Identities. We have demonstrated how, in parallel to the Web for humans, these interplaying webs will provide a new level of machine understanding and interoperability, which one could see as common sense for machines.

We want to note that it is indispensable that all Webs ensure *security*, *privacy* and *trust*, internally as well as in their interaction. The notions as described here however allow including mechanisms that support these, just as the World Wide Web provides the basis for secure transactions without prescribing technologies.

Our vision of three interlinked, yet clearly definable areas within the future Internet allows for focussed research and development in either each of the webs or in their interactions, some of which we hinted at in the scenarios. However, other interactions are of course possible. We believe that research areas and business cases can and will arise from the Web of Machines [85, 82].

9.3. Summary

This chapter concluded this work by outlining a long-term perspective of two future trends concerning social data and social applications in the wider research area.

In the next years, users will begin to demand more privacy protection in social networking systems and thereby force providers to provide solutions. Without a

doubt, the problems described in this work arose from a very young trend and techniques of SNSs, which soon will start to settle and normalize. Numerous ways to approach these problems have been recommended of which one or the other will hopefully make users demand a better protection of their privacies, soon.

A. Questionnaire

This chapter lists the original questionnaire, including the respondents' answers. The questionnaire was presented in German, because the majority of users are Germans. The total number of respondents is 67. Answer options are translated at their first occurrence.

A.1. Privacy Category

The following Survey Questions (SQ) are aimed at categorizing respondents into Westin's privacy categories, i.e. Privacy Fundamentalists, Privacy Unconcerned and Privacy Pragmatists (Section 3.1) [92].

Survey Question 1

Ich habe schon viel zu viel von meiner Privatsphäre verloren und möchte jeden weiteren Verlust verhindern.
I have already lost much too much of my privacy and try avoid any further loss.
(Table A.1)

trifft zu	fully agree	5
trifft teilweise zu	partially agree	22
weiß nicht	unsure	7
trifft kaum zu	partially disagree	23
trifft nicht zu	fully disagree	10

Table A.1.: I have already lost much too much of my privacy and try avoid any further loss (SQ1).

Survey Question 2

Was andere Personen und Unternehmen mit privaten Informationen über mich tun interessiert mich nicht.
I am not interesting in what other persons or companies do with private information about myself.
(Table A.2)

Survey Question 3

Ich achte darauf, dass andere Personen oder Unternehmen meiner Privatsphäre nicht schaden oder sie missbrauchen.

trifft zu	2
trifft teilweise zu	7
weiß nicht	1
trifft kaum zu	11
trifft nicht zu	46

Table A.2.: I am not interesting in what other persons or companies do with private information about myself (SQ2).

I am taking care that other persons or companies do not harm or abuse my privacy. (Table A.3)

trifft zu	37
trifft teilweise zu	26
weiß nicht	1
trifft kaum zu	1
trifft nicht zu	2

Table A.3.: I am taking care that other persons or companies do not harm or abuse my privacy (SQ3).

A.2. Intuitivity and Usability

The following questions ask the respondent about his or her intuitive understanding of the application's purpose and its usability.

Survey Question 4

Ich fand es einfach zu verstehen, was man mit Cityfinger machen kann.
I found it easy to understand what I can do with Cityfinger.
(Table A.4)

trifft zu	25
trifft teilweise zu	27
weiß nicht	7
trifft kaum zu	4
trifft nicht zu	4

Table A.4.: I found it easy to understand what I can do with Cityfinger (SQ4).

Survey Question 5

Ich habe sofort verstanden, wie Informationen bei Cityfinger organisiert sind.
I immediately understood how information is organized at Cityfinger.
(Table A.5)

trifft zu	13
trifft teilweise zu	29
weiß nicht	9
trifft kaum zu	12
trifft nicht zu	4

Table A.5.: I immediately understood how information is organized at Cityfinger (SQ5).

Survey Question 6

Ich fand es intuitiv, wie man bei Cityfinger zu Tips und anderen Informationen navigiert.
I found navigating to Tips and other information on Cityfinger intuitive.
(Table A.6)

trifft zu	18
trifft teilweise zu	28
weiß nicht	9
trifft kaum zu	7
trifft nicht zu	5

Table A.6.: I found it intuitive how to navigate to Tips and other information on Cityfinger (SQ6).

A.3. SNS Usage

The following questions ask about the respondents' usage habits of Cityfinger and other social networks for comparison. They continue questioning the subjectively felt control, transparency and friendship relevancy.

Survey Question 7

Welche sozialen Netzwerke (Communities) nutzt du?
Which social networks do you use?
(Table A.7)

	mehrmals täglich	täglich	früher täglich	wöchentlich	früher wöchentlich	monatlich	garnicht
Cityfinger	1	0	2	1	11	33	18
Facebook	18	21	2	5	5	3	12
StudiVZ / MeinVZ	0	2	2	5	7	6	42
XING	2	9	3	18	3	6	21
Lokalisten	0	0	1	0	0	0	62
Wer kennt wen	0	0	1	1	0	1	58
LinkedIn	1	2	1	9	2	12	36
MySpace	0	0	2	3	6	7	44

Table A.7.: Which SNSs do you use (SQ7)?

mehrmals täglich	several times a day
täglich	daily
früher täglich	used to be daily
wöchentlich	weekly
früher wöchentlich	used to be weekly
monatlich	monthly
garnicht	not at all

Survey Question 8

Wenn ich etwas online erstelle oder hochlade habe ich Kontrollmöglichkeiten darüber, wer darauf Zugriff haben wird.
When I put something online or upload it, I have ways to control who will be granted access to it.
(Table A.8)

	trifft zu	trifft teilweise zu	weiß nich	trifft kaum zu	trifft nicht zu
Cityfinger	24	13	24	0	1
Facebook	12	30	11	8	3
StudiVZ / MeinVZ	5	10	19	4	4
XING	10	23	14	2	3
Lokalisten	0	0	33	0	5
Wer kennt wen	0	0	33	1	3
LinkedIn	4	11	28	2	3
MySpace	2	4	28	5	4

Table A.8.: When I put something online or upload it, I have ways to control who will be granted access to it (SQ8).

Survey Question 9

Wenn ich etwas online gestellt habe, kann ich danach sehen, wer nun Zugriff darauf hat.
When I have put something online, I can afterwards see who has access to it.
(Table A.9)

Survey Question 10

Ich kann ganz offen mit meinem Kontakten online kommunizieren weil ich einen Überblick habe wer mitlesen kann und wer nicht.
I can talk openly to my friends, because I have an overview of who can read our communication.
(Table A.10)

Survey Question 11

Meine Freunde in der Community sind hauptsächlich auch richtige Freunde.
My friends on that SNS are mainly real friends.
(Table A.11)

	trifft zu	trifft teilweise zu	weiß nich	trifft kaum zu	trifft nicht zu
Cityfinger	21	8	27	1	4
Facebook	7	20	15	14	7
StudiVZ / MeinVZ	2	5	20	8	4
XING	9	16	15	9	2
Lokalisten	0	0	29	2	2
Wer kennt wen	0	0	29	2	2
LinkedIn	1	7	28	4	3
MySpace	0	2	29	6	2

Table A.9.: When I have put something online, I can afterwards see who will has access to it (SQ9).

	trifft zu	trifft teilweise zu	weiß nich	trifft kaum zu	trifft nicht zu
Cityfinger	21	8	25	2	1
Facebook	10	21	15	12	1
StudiVZ / MeinVZ	2	7	18	5	2
XING	16	7	17	6	1
Lokalisten	0	0	23	1	2
Wer kennt wen	0	1	23	0	1
LinkedIn	5	3	25	4	1
MySpace	0	3	24	4	2

Table A.10.: I can talk openly to my friends, because I have an overview of who can read our communication (SQ10).

	trifft zu	trifft teilweise zu	weiß nich	trifft kaum zu	trifft nicht zu
Cityfinger	21	17	6	11	3
Facebook	10	36	3	7	4
StudiVZ / MeinVZ	6	10	8	6	4
XING	1	18	5	15	7
Lokalisten	0	0	15	1	7
Wer kennt wen	0	1	16	1	5
LinkedIn	0	8	9	10	9
MySpace	1	4	13	5	6

Table A.11.: My friends on that SNS are mainly real friends (SQ11).

Survey Question 12

Die Gruppen der Community bestehen zu einem Großteil aus Menschen, die ich persönlich kenne.
The community's groups mainly consist of people I know personally.
(Table A.12)

Survey Question 13

Der meiste Content in der Community ist sehr relevant für mich weil er von richtigen Freunden kommt.
Most of the content is highly relevant to me, since it comes from real friends.
(Table A.13)

A.4. Demographics

The next questions gather demographic data from the participants.

Survey Question 14

Bist du männlich oder weiblich?
Are you male or female?
(Table A.14)

Survey Question 15

Wie alt bist Du?
How old are you?
(Table A.15)

	trifft zu	trifft teilweise zu	weiß nich	trifft kaum zu	trifft nicht zu
Cityfinger	25	17	6	6	4
Facebook	22	15	5	7	7
StudiVZ / MeinVZ	6	5	8	6	6
XING	11	11	10	10	3
Lokalisten	0	0	13	0	6
Wer kennt wen	1	1	13	0	4
LinkedIn	8	6	9	5	5
MySpace	1	6	12	4	4

Table A.12.: The community's groups mainly consist of people I know personally (SQ12).

	trifft zu	trifft teilweise zu	weiß nich	trifft kaum zu	trifft nicht zu
Cityfinger	13	17	14	6	6
Facebook	7	25	8	10	5
StudiVZ / MeinVZ	2	5	8	7	8
XING	1	9	9	16	7
Lokalisten	0	0	12	1	5
Wer kennt wen	1	1	13	0	3
LinkedIn	0	4	12	8	7
MySpace	0	4	11	7	5

Table A.13.: Most of the content is highly relevant to me, since it comes from real friends (SQ13).

weiblich	female	9
männlich	male	58

Table A.14.: Are you male or female (SQ14)?

jünger als 20	younger than 20	0
21–25		2
26–30		19
31–35		26
36–40		11
41–45		4
46–50		1
51–60		0
älter als 60	older than 60	0

Table A.15.: How old are you (SQ15)?

Survey Question 16

Wieviel verdienst du im Jahresbrutto?
What is your annual income before tax?
(Table A.16)

20.000–30.000 Euro	16
30.000–40.000 Euro	15
40.000–50.000 Euro	10
50.000–75.000 Euro	7
75.000–100.000 Euro	3
$>$ 100.000 Euro	2

Table A.16.: What is your annual income before tax (SQ16)?

Survey Question 17

Wie ist dein Familienstand?
What is your marital status?
(Table A.17)

ledig	single	56
verheiratet	married	11
geschieden	divorced	0

Table A.17.: What is your marital status (SQ17)?

Survey Question 18

Welchen Ausbildungsstand hast Du?
What is your educational level?
(Table A.18)

Doktortitel	doctor's degree	3
Hochschulabschluss	university degree	48
Abitur	university-entrance diploma	14
Realschulabschluss	school-leaving certificate	2
Hauptschulabschluss	certificate of secondary education	0
Ohne Abschluss	without degree	0

Table A.18.: What is your educational level (SQ18)?

A.5. Internet Usage

The last set of questions investigate the respondents' Internet usage habits.

Survey Question 19

Wieviele Stunden pro Tag nutzt Du aktiv das Internet?
How many hours a day do you actively use the Internet?
(Table A.19)

0–15 Minuten	0–15 minutes	0
ca. 30 Minuten	about half an hour	2
ca. eine Stunde	about an hour	8
2–4 Stunden	2–4 hours	26
5–8 Stunden	5–8 hours	21
9–12 Stunden	9–12 hours	10

Table A.19.: How many hours a day do you actively use the Internet (SQ19)?

Survey Question 20

Wieviele Stunden pro Tag nutzt Du Communities aktiv?
How many hours a day do you actively use social networks?
(Table A.20)

0–15 Minuten	29
ca. 30 Minuten	17
ca. eine Stunde	12
2–4 Stunden	6
5–8 Stunden	1
9–12 Stunden	2

Table A.20.: How many hours a day do you actively use social networks (SQ20)?

B. Visual Design

This chapter provides selected further reading regarding the visual design applied to the application.

B.1. Further Colors

In addition to Section 6.7.4, this section adds further details to applied colors and their palettes.

Entities listed in the Tier 2 Navigation area have two alternating background colors which are lighter versions of the already described entity color-codes (Section 6.7.4). These colors are depicted in Figure B.1.

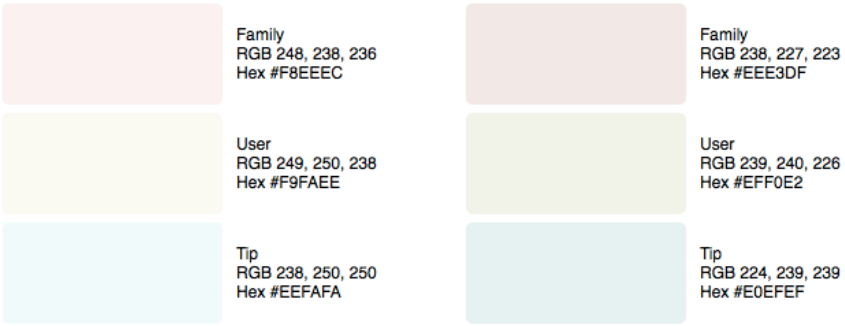

Figure B.1.: Primary colors used for alternating backgrounds in the Tier 2 Navigation area.

Figure B.2 lists further colors used for links to functions and other page elements, e.g., backgrounds, lines, frames, design elements and markings. Examples of applications can, e.g., be found in Figure 6.19: The colorization of the search area, the link colors of the message and logout area, and the action links in Tier 1 Content area.

B.2. Logo and typefaces

As introduced in Section 6.7.4, the application was equipped with a logo and several typefaces. This section lists a selection of details.

The application's logo is embedded in the center of the header. Figure B.3 displays the header containing the logo and Figure B.4 illustrates variations of the

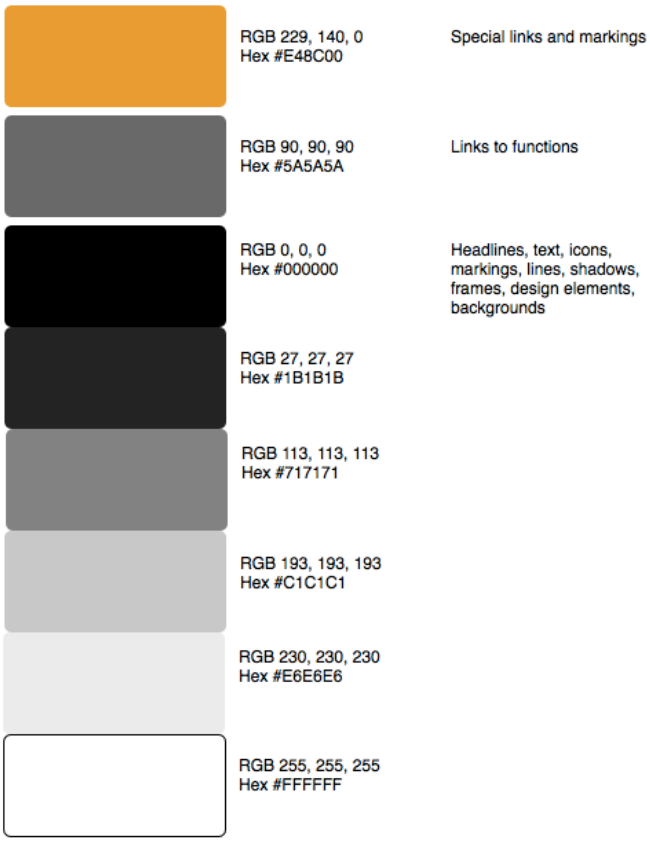

Figure B.2.: Secondary colors used for links to functions and other page elements.

logo on black and white background. The fingerprint stands for the individuality of the secret tips users share with their friends through the application.

Figure B.3.: Web application header graphics with the embedded logo.

Figure B.4.: Logo variations on black and white background.

The application comes with three typefaces: For the logo, the typeface Cityfinger was developed and used (Figures B.5 and 6.19).

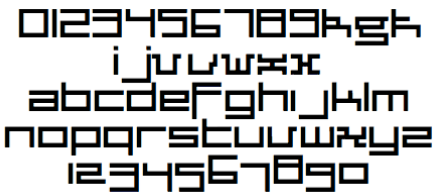

Figure B.5.: The Cityfinger typeface was developed for the logo.

For the Tier 1 Navigation area, the typeface The Sans was applied (Figures B.6, B.7 and 6.19). For all remaining texts, the typefaces Verdana and Verdana Bold (not displayed) were used (Figures B.8 and 6.19).

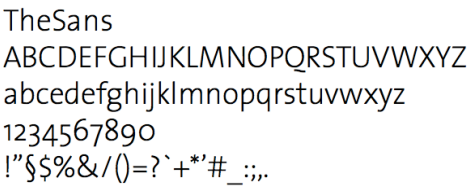

Figure B.6.: The typeface The Sans was used for the Tier 1 Navigation area.

Figure B.7.: The typeface The Sans was only used to render the names of the three main entity types in the Tier 1 Navigation area.

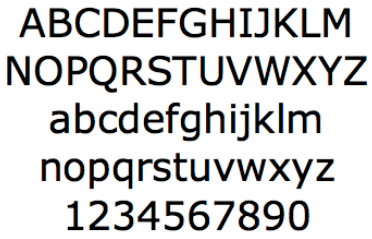

Figure B.8.: The typefaced Verdana and Verdana Bold (not displayed) were used for all remaining texts.

C. Early Stage Sketches

This chapter presents a selection of early stage sketches to demonstrate the process of the development and optimization the application's layout, navigation design and visual design (cf. Section 6.7.1).

Figure C.1 shows an early stage layout sketch which contains a map in the left column, a list of Tips in the middle, and a grid view of friends in the right column. It transparently lists the Tips and the Users of a particular Family. Under the header, links to Families are listed. Underneath, important call-to-action buttons for Family and Tips creation as well as friend invitations are displayed. The map was moved to the right column in later layouts.

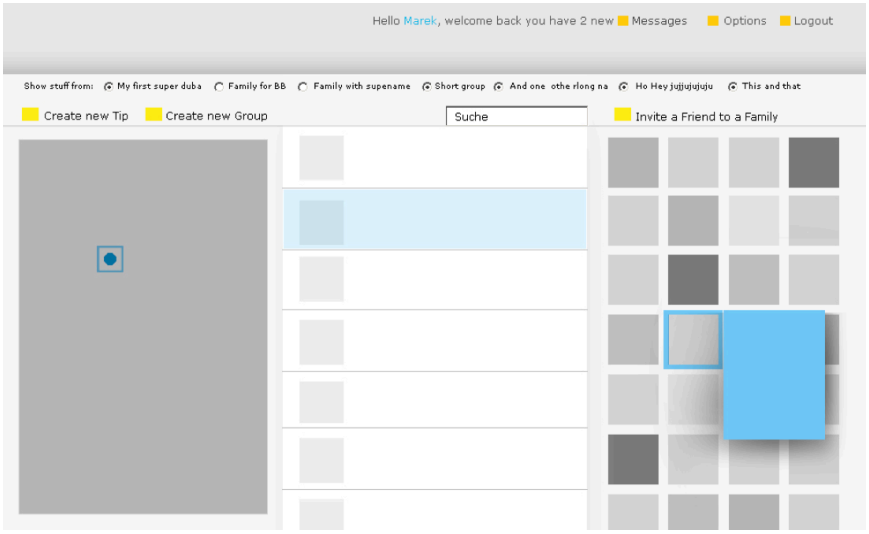

Figure C.1.: Early stage layout sketch.

Figure C.2 illustrates an early stage screen with a different navigation design. As friends permanently occupy the left column in a grid layout, the remaining entities shared the center column for their list-views and show-views. Figure C.3 depicts a screen dummy which features the navigation of the final version, but an early visual design. Also, the separation of page content by columns has almost reached the final stage.

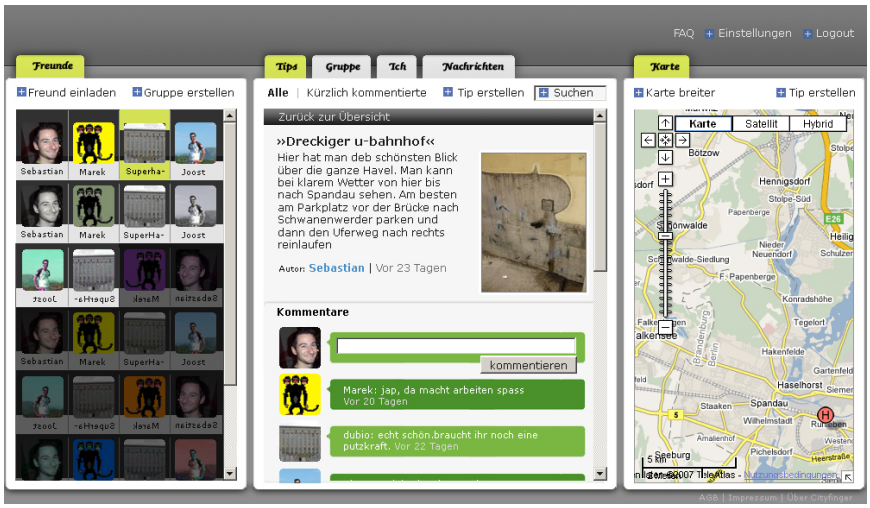

Figure C.2.: Early stage screen dummy featuring top-level navigation over the center column.

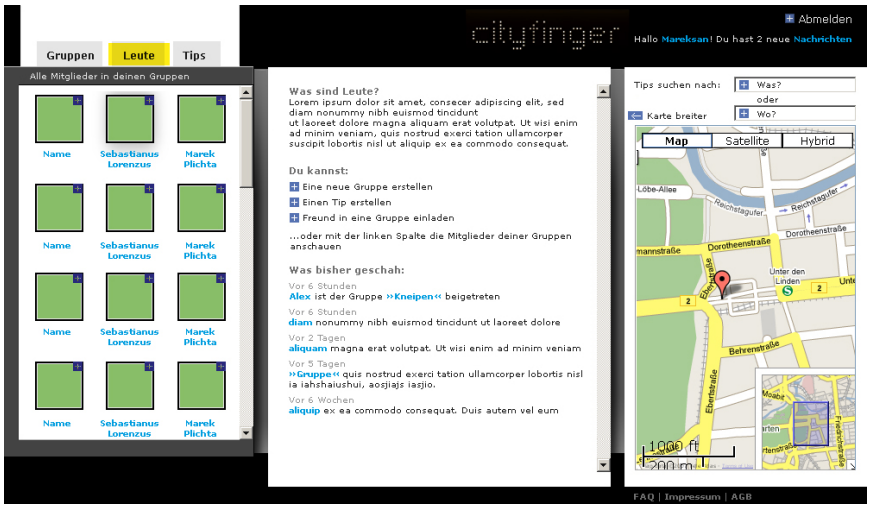

Figure C.3.: Screen dummy with final navigation design, but early visual design.

Bibliography

[1] 37signals. *Getting Real: The smarter, faster, easier way to build a successful web application.* 37signals, LLC., 2006.

[2] Mark S. Ackerman, Lorrie Faith Cranor, and Joseph Reagle. Privacy in e-commerce: Examining user scenarios and privacy preferences. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 1–8. ACM, 1999.

[3] Alessandro Acquisti. Privacy in electronic commerce and the economics of immediate gratification. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 21–29. ACM, 2004.

[4] Alessandro Acquisti and Ralph Gross. Imagined communities: Awareness, information sharing, and privacy on the Facebook. In *Proceedings of the 6th Workshop on Privacy Enhancing Technologies (PET)*, pages 36–58, 2006.

[5] Alessandro Acquisti and Jens Grossklags. Privacy and rationality in individual decision making. *IEEE Security & Privacy*, 3(1):26–33, 2005.

[6] Ben Adida and Mark Birbeck. RDFa primer: Bridging the human and data webs. `http://www.w3.org/TR/xhtml-rdfa-primer` (accessed September 2009), October 2008.

[7] R. Akkiraju, J. Farrell, J.Miller, M. Nagarajan, M. Schmidt, A. Sheth, and K. Verma. Web service semantics - WSDL-S, a joint UGA-IBM technical note, version 1.0. Technical report, IBM and the University of Georgia, April 2005.

[8] John Allsopp. *Microformats - Empowering Your Markup for Web 2.0.* Springer, 2007.

[9] Chris Anderson. *The Long Tail: Why the Future of Business is Selling Less of More.* Hyperion, 2006.

[10] Jonathan Anderson, Claudia Diaz, Joseph Bonneau, and Frank Stajano. Privacy-preserving social networking over untrusted networks. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Online Social Networks (WOSN).* ACM, 2009.

[11] Anupriya Ankolekar, Markus Krötzsch, Thanh Tran, and Denny Vrandecic. The two cultures: Mashing up web 2.0 and the semantic web. In *Proceedings of the 16th International World Wide Web Conference (WWW)*, pages 825–834. ACM, 2007.

[12] Annie I. Antón, Elisa Bertino, Ninghui Li, and Ting Yu. A roadmap for comprehensive online privacy policy management. *Communications of the ACM*, 50(7):109–116, 2007.

[13] Gaurav Bansal, Fatemeh Zahedi, and David Gefen. The moderating influence of privacy concern on the efficacy of privacy assurance mechanisms fo building trust: A multiple context investigation. In *Proceedings of the International Conference on Information Systems (ICIS)*, 2008.

[14] David Barroso. Virtual worlds, real money: Security and privacy in massively-multiplayer online games and social and corporate virtual worlds. Technical report, ENISA - European Network and Information Security Agency, 2008.

[15] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein, and Franklin W. Olin. OWL web ontology language reference. `http://www.w3.org/TR/2003/PR-owl-ref-20031215` (accessed September 2009), December 2003.

[16] Victoria Bellotti and Abigail Sellen. Design for privacy in ubiquitous computing environments. In *Proceedings of the 3rd Conference on European Conference on Computer-Supported Cooperative Work (ECSCW)*, pages 77–92. Kluwer Academic Publishers, 1993.

[17] Tim Berners-Lee and Marc Fischetti. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. Harper, 1999.

[18] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):28–37, 2001.

[19] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. All your contacts are belong to us: Automated identity theft attacks on social networks. In *Proceedings of the 18th International World Wide Web Conference (WWW)*, pages 551–560, 2009.

[20] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked data on the web (LDOW2008). In *Proceedings of the 17th International World Wide Web Conference (WWW)*, pages 1265–1266. ACM, 2008.

[21] Uldis Bojars, John G. Breslin, Aidan Finn, and Stefan Decker. Using the semantic web for linking and reusing data across web 2.0 communities. *Web Semantics*, 6(1):21–28, 2008.

[22] Uldis Bojars, John G. Breslin, Vassilios Peristeras, Giovanni Tummarello, and Stefan Decker. Interlinking the social web with semantics. *IEEE Intelligent Systems*, 23(3):29–40, 2008.

[23] Uldis Bojars, Alexandre Passant, John G. Breslin, and Stefan Decker. Data portability with SIOC and FOAF. In *Proceedings of the XTech Conference*, Dublin, Ireland, 2008.

[24] Uldis Bojars, Alexandre Passant, John G. Breslin, and Stefan Decker. Social network and data portability using semantic web technologies. In *Proceedings of the 2nd Workshop on Social Aspects of the Web (SAW) at BIS2008*, pages 5–19, 2008.

[25] J. Bonneau, J. Anderson, F. Stajano, and R. Anderson. Eight friends are enough: Social graph approximation via public listings. In *Proceeding of the 2nd ACM Workshop on Social Network Systems (SNS)*, pages 13–18, 2009.

[26] Joseph Bonneau, Jonathan Anderson, and George Danezis. Prying data out of a social network. In *Proceedings of the Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 249–254, 2009.

[27] Joseph Bonneau and Sören Preibusch. *Economics of Information Security and Privacy*, chapter The Privacy Jungle: On the Market for Data Protection in Social Networks, pages 121–167. Springer, 2010.

[28] Joseph Bonneau, Sören Preibusch, Jonathan Anderson, Richard Clayton, and Ross Anderson. Democracy theatre: Comments on facebook's proposed governance scheme. `http://preibusch.de/publications/Bonneau_Preibusch_Anderson_Clayton_Anderson__Facebook_Governance_Comments.pdf` (accessed September 2009), March 2009.

[29] Danah Boyd. Facebook's privacy trainwreck: Exposure, invasion, and social convergence. *Convergence: The International Journal of Research into New Media Technologies*, 14(1):13–20, February 2008.

[30] Danah Boyd, Carlos Jensen, Scott Lederer, and David H. Nguyen. Privacy in digital environments: Empowering users (workshop organizer's proposal). `http://citeseer.ist.psu.edu/boyd02privacy.html` (accessed December 2008), November 2002.

[31] Danah M. Boyd and Nicole B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, October 2007.

[32] J.G. Breslin, A. Harth, U. Bojars, and S. Decker. Towards semantically-interlinked online communities. In *Proceedings of the 2nd European Semantic Web Conference (ESWC)*, pages 500–514, 2005.

[33] Sonja Buchegger and Anwitaman Datta. A case for P2P infrastructure for social networks - opportunities & challenges. In *Proceedings of the 6th International Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 161–168, 2009.

[34] Marc Canter. How to build the open mesh. `http://blog.broadbandmechanics.com/2008/05/02/how-to-build-the-open-mesh` (accessed September 2009), 2008.

[35] Fred H. Cate. *Privacy in the information age*. Brookings Institution Press, 1997.

[36] Munmun De Choudhury, Hari Sundaram, Ajita John, and Doree Seligmann. Dynamic prediction of communication flow using social context. In *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia (HT)*, pages 49–54, 2008.

[37] Munmun De Choudhury, Hari Sundaram, Ajita John, and Doree D. Seligmann. Contextual prediction of communication flow in social networks. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 57–65, 2007.

[38] comscore. Major social networking sites substantially expanded their global visitor base during past year. `http://www.comscore.com/press/release.asp?press=1555` (accessed December 2008), July 2007.

[39] Lorrie F. Cranor. *Web Privacy with P3P*. O'Reilly Media, Inc., September 2002.

[40] Lorrie Faith Cranor, Joseph Reagle, and Mark S. Ackerman. Beyond concern: Understanding net users'attitudes about online privacy. Technical report, The Internet Upheaval: Raising Questions, Seeking Answers in Communications Policy, 1999.

[41] Jonathan Dale, Bernard Burg, and Steven Willmott. *Innovative Concepts for Agent-Based Systems*, volume 2564/2003 of *Lecture Notes in Computer Science*, chapter The Agentcities Initiative: Connecting Agents Across the World, pages 453–457. Springer, 2003.

[42] John Domingue, Dieter Fensel, and Rafael Gonzalez-Cabero. SOA4All, enabling the SOA revolution on a world wide scale. In *Proceeding of the 2nd IEEE International Conference on Semantic Computing (ICSC)*, pages 530–537. IEEE Computer Society, 2008.

[43] Catherine Dwyer, Starr Roxanne Hiltz, and Katia Passerini. Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. In *Proceedings of the 13th Americas Conference on Information Systems (AMCIS)*, 2007.

[44] Serge Egelman, Janice Y. Tsai, Lorrie Faith Cranor, and Alessandro Acquisti. Timing is everything? The effects of timing and placement of online privacy indicators. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI)*, pages 319–328. ACM, 2009.

[45] ENISA. Security issues and recommendations for online social networks. Technical report, European Network Information Security Agency (ENISA), 2007.

[46] Thomas Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. The Prentice Hall Service-Oriented Computing Series from Thomas Erl. Prentice Hall, USA, 2005.

[47] Adrienne Felt. Defacing facebook: A security case study. `http://www.cs.virginia.edu/felt/fbook/facebook-xss.pdf` (accessed August 2009), 2007.

[48] Adrienne Felt and David Evans. Privacy protection for social networking APIs. In *Proceedings of the 1st Workshop on Web 2.0 Security and Privacy (W2SP)*, 2008.

[49] Adrienne Felt, Pieter Hooimeijer, David Evans, and Westley Weimer. Talking to strangers without taking their candy: Isolating proxied content. In *Proceedings of the 1st Workshop on Social Network Systems (SocialNets)*, pages 25–30. ACM, 2008.

[50] Dieter Fensel, Holger Lausen, Axel Polleres, De Bruijn, Jos, Michael Stollberg, Dumitru Roman, and John Domingue. *Enabling Semantic Web Services: The Web Service Modeling Ontology*. Springer, 2006.

[51] Philip W. L. Fong, Mohd Anwar, and Zhen Zhao. A privacy preservation model for Facebook-style social network systems. In *Proceedings of the 14th European Conference on Research in Computer Security (ESORICS)*, pages 303–320. Springer, 2009.

[52] Eric Freeman. *Head First Design Patterns*. O'Reilly Media, 2004.

[53] Jesse James Garrett. *The Elements of User Experience: User-Centered Design for the Web*. Peachpit Press, October 2002.

[54] Julia Gideon, Lorrie Faith Cranor, Serge Egelman, and Alessandro Acquisti. Power strips, prophylactics, and privacy, oh my! In *Proceedings of the 2nd Symposium on Usable Privacy and Security (SOUPS)*, volume 149 of *ACM International Conference Proceeding Series*, pages 133–144. ACM, 2006.

[55] Malcom Gladwell. *Tipping Point*. Goldmann, 2000.

[56] Mark S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 1973.

[57] James Grimmelmann. Facebook and the social dynamics of privacy. `http://works.bepress.com/cgi/viewcontent.cgi?article=1019&context=james_grimmelmann` (accessed December 2008), 2008.

[58] Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks (the facebook case). In *Proceedings of the ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 71–80, November 2005.

[59] Saikat Guha, Kevin Tang, and Paul Francis. NOYB: Privacy in online social networks. In *Proceedings of the 1st workshop on Online Social Networks (WOSP)*, pages 49–54. ACM, 2008.

[60] Il-Horn Hann, Kai-Lung Hui, Tom S. Lee, and I. P. L. Png. Online information privacy: Measuring the cost-benefit trade-off. In *Proceedings of the 23rd International Conference on Information Systems*, 2002.

[61] Rom Harré. *Social Being*. Wiley-Blackwell, 2nd edition, 1993.

[62] R. A. Hill and R. I. M. Dunbar. Social network size in humans. *Human Nature*, 14(1):53–72, 2002.

[63] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. Automotive IT-security as a challenge: Basic attacks from the black box perspective on the example of privacy threats. In *Proceedings of the 28th International Conference on Computer Safety, Reliability, and Security*, pages 145–158, 2009.

[64] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*, volume 4011 of *LNAI*, pages 411–426. Springer, 2006.

[65] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Trend detection in folksonomies. In *Proceedings of the 1st International Conference on Semantics And Digital Media Technology (SAMT)*, volume 4306 of *LNCS*, pages 56–70. Springer, 2006.

[66] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Bibsonomy: A social bookmark and publication sharing system. In *Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures*, pages 87–102. Aalborg University Press, 2006.

[67] Indratmo and Julita Vassileva. A usability study of an access control system for group blogs. In *Proceedings of the International Conference on Weblogs and Social Media*, 2007.

[68] Information Society Technologies Advisory Group. Working group on web-based service industry. `ftp://ftp.cordis.europa.eu/pub/ist/docs/web-based-service-industry-istag_en.pdf` (accessed December 2008), February 2008.

[69] International Working Group on Data Protection in Telecommunications. Report and guidance on social network services - Rome Memorandum. Technical report, International Working Group on Data Protection in Telecommunications, 2008.

[70] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson, and Filippo Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, 2007.

[71] Harvey Jones and José H. Soltren. Facebook: Threats to privacy. `http://groups.csail.mit.edu/mac/classes/6.805/student-papers/fall05-papers/facebook.pdf` (accessed December 2008), December 2005.

[72] Simon Jones and Eamonn O'Neill. Feasibility of structural network clustering for group-based privacy control in social networks. In *Proceedings of the 6th Symposium on Usable Privacy and Security (SOUPS)*, pages 1–13. ACM, 2010.

[73] Clare-Marie Karat, John Karat, Carolyn Brodie, and Jinjuan Feng. Evaluating interfaces for privacy policy rule authoring. In *Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI)*, pages 83–92. ACM, 2006.

[74] G. Karvounarakis, A. Magkanaraki, S. Alexaki, V. Christophides, D. Plexousakis, M. Scholl, and K. Tolle. RQL: A functional query language for RDF. In *The Functional Approach to Data Management: Modelling, Analyzing and Integrating Heterogeneous Data*, pages 435–465. Springer, 2004.

[75] Kevin Kelly. *New Rules for the New Economy: 10 Radical Strategies for a Connected World*. Viking Adult, 1998.

[76] Rohit Khare. Microformats: The next (small) thing on the semantic web? *IEEE Internet Computing*, 10(1):68–75, 2006.

[77] Peter Kim. Applying game mechanics to social media. `http://www.beingpeterkim.com/2008/07/applying-game-m.html` (accessed August 2009), 2008.

[78] Aleksandra Korolova, Rajeev Motwani, Shubha U. Nabar, and Ying Xu. Link privacy in social networks. In *Proceeding of the 17th ACM Conference on Information and Knowledge Mining (CIKM)*, pages 289–298, New York, NY, USA, 2008. ACM.

[79] Alexander Korth. The Web of data: Creating machine-accessible information. ReadWriteWeb, `http://www.readwriteweb.com/archives/web_of_data_machine_accessible_information.php` (accessed August 2009), April 2009.

[80] Alexander Korth. The Web of identities: Making machine-accessible people data. ReadWriteWeb, `http://www.readwriteweb.com/archives/web_of_identities_making_machine-accessible_people_data.php` (accessed August 2009), July 2009.

[81] Alexander Korth. The Web of services: Machine-accessible services. ReadWriteWeb, `http://www.readwriteweb.com/archives/web_of_services_machine-accessible_services.php` (accessed August 2009), October 2009.

[82] Alexander Korth. The trilogy of webs for machines: Mashing it all together. ReadWriteWeb[1], `http://www.readwriteweb.com/archives/the_trilogy_of_webs_for_machines_mashing_it_all_together.php` (accessed January 2011), June 2010.

[83] Alexander Korth. On privacy in social networks: What drives users? ReadWriteWeb, `http://www.readwriteweb.com/archives/on_privacy_in_social_networks_what_drives_users.php` (accessed April 2011), April 2011.

[1] ReadWriteWeb is one of the most popular and influential technology blogs in the world. As of February 2011, Technorati, the world's leading blog directory, ranked it #13 in its overall rankings, #4 in Info Tech, and #7 in Technology.

[84] Alexander Korth, Stephan Baumann, and Andreas Nürnberger. An interdisciplinary problem taxonomy for user privacy in social networking services. In *Proceedings of the CHI 2011 Workshop on Networked Privacy*, 2011.

[85] Alexander Korth, Benjamin Hirsch, Till Plumbaum, and Andreas Nürnberger. A trilogy of webs for machines. In *Proceedings of the Workshop on Linked Data in the Future Internet at the Future Internet Assembly (LinkedDataFIA)*, December 2010.

[86] Alexander Korth and Andreas Nürnberger. Improving social recommendations through privacy-awareness. In *Proceedings of the 2nd Workshop on Social Recommender Systems (SRS)*, 2011.

[87] Alexander Korth and Till Plumbaum. A framework for ubiquitous user modeling. In *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI)*, pages 291–297. IEEE Computer Society Press, 2007.

[88] Hanna Krasnova, Oliver Günther, Sarah Spiekermann, and Ksenia Koroleva. Privacy concerns and identity in online social networks. *Identity in the Information Society*, 2(1):39–63, 2009.

[89] Hanna Krasnova, Thomas Hildebrand, Oliver Günther, Alexander Kovrigin, and Aneta Nowobilska. Why participate in an online social network: An empirical analysis. In *Proceedings of the 16th European Conference on Information Systems (ECIS)*, 2008.

[90] Balachander Krishnamurthy and Craig E. Wills. Characterizing privacy in online social networks. In *Proceedings of the 1st workshop on Online Social Networks (WOSP)*, pages 37–42, 2008.

[91] Balachander Krishnamurthy and Craig E. Wills. Privacy diffusion on the web: A longitudinal perspective. In *Proceedings of the 18th International World Wide Web Conference (WWW)*, pages 541–550, 2009.

[92] Ponnurangam Kumaraguru and Lorrie Faith Cranor. Privacy indexes: A survey of Westin's studies. Technical report, Institute for Software Research International, School of Computer Science, 2005.

[93] Marc Langheinrich. Privacy by design - principles of privacy-aware ubiquitous systems. In *Proceedings of the UbiComp (Ubiquitous Computing)*, pages 273–291, September 2001.

[94] Lawrence Lessig. The architecture of privacy. `http://lessig.org/content/articles/works/architecture_priv.pdf` (accessed August 2009), 1998.

[95] Heather Richter Lipford, Andrew Besmer, and Jason Watson. Understanding privacy settings in Facebook with an audience view. In *Proceedings of the 1st Conference on Usability, Psychology, and Security*, pages 2:1–2:8. USENIX Association, 2008.

[96] Matthew M. Lucas and Nikita Borisov. FlyByNight: Mitigating the privacy risks of social networking. In *Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 1–8. ACM, 2008.

[97] Ankur Mani and Hari Sundaram. Modeling user context with applications to media retrieval. *Multimedia Systems*, 12(4-5):339–353, 2007.

[98] Mohammad Mannan and Paul C. van Oorschot. Privacy-enhanced sharing of personal content on the web. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 487–496, 2008.

[99] Frank Manola and Eric Miller. RDF primer. `www.w3.org/TR/REC-rdf-syntax` (accessed December 2008), February 2004.

[100] David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, and Katia Sycara. OWL-S: Semantic markup for web services. `http://www.daml.org/services/owl-s/1.0/owl-s.pdf` (accessed December 2008), 2004.

[101] Abraham H. Maslow. A theory of human motivation. *Psychological Review*, 50(4):370–396, 1943.

[102] MediaAnalyzer. Webseitenstudie. `http://www.mediaanalyzer.com/know-how/studien/MA_P_WebWahrnehmung_020515` (accessed November 2009), 2002.

[103] Peter Mika. Flink: Semantic web technology for the extraction and analysis of social networks. *Journal of Web Semantics*, 3(2):211–223, 2005.

[104] Peter Mika. *Social Networks and the Semantic Web*. Springer, 2007.

[105] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[106] George Milne and Mary Culnan. The culnan milne survey on consumers online privacy notices. *Journal of Interactive Marketing*, 18(3), 2004.

[107] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC)*, pages 29–42, 2007.

[108] John Musser and Tim O'Reilly. Web 2.0 principles and best practices. `http://radar.oreilly.com/research/web2-report.html` (accessed August 2009), 2006.

[109] Mark E. J. Newman. Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46:323–351, 2005.

[110] Qun Ni and Elisa Bertino. Privacy-aware role-based access control. *ACM Transactions on Information and System Security*, 13:41–50, 2010.

[111] Jakob Nielsen. Participation inequality: Encouraging more users to contribute. http://www.useit.com/alertbox/participation_inequality.html (accessed August 2009), 2006.

[112] Jakob Nielsen and Hoa Loranger. *Web Usability*. Addison-Wesley, 2008.

[113] Helen Nissenbaum. Protecting privacy in an information age: The problem of privacy in public. *Law and Philosophy*, 17:559–596, 1998.

[114] Donald Norman. *The Design of Everyday Things*. Perseus Books, 1988.

[115] Anthony Onwuasoanya, Maxim Skornyakov, and Jonathan Post. Enhancing privacy on social networks by segregating different social spheres. *Rutgers Governor's School of Engineering and Technology Research Journal*, 2008.

[116] Tim O'Reilly. What is Web 2.0: Design patterns and business models for the next generation of software. http://oreilly.com/web2/archive/what-is-web-20.html (accessed January 2011), September 2005.

[117] Jeremiah Owyang. The future of the social web: In five eras. http://www.web-strategist.com/blog/2009/04/27/future-of-the-social-web (accessed August 2009), April 2009.

[118] Martin Pekárek and Stefanie Pötzsch. Requirements and concepts for privacy-enhancing access control in social networks and collaborative workspaces. http://www.primelife.eu/images/stories/deliverables/h1.2.5-requirements_selective_access_control-public.pdf (accessed September 2009), July 2009.

[119] J. C. Poindexter, Julia Brande Earp, and David L. Baumer. An experimental economics approach toward quantifying online privacy choices. *Information Systems Frontiers*, 8(5):363–374, 2006.

[120] Irene Pollach. What's wrong with online privacy policies? *Communications of the ACM*, 50(9):103–108, September 2007.

[121] Sören Preibusch. Implementing privacy negotiations in e-commerce. In *AP-Web*, volume 3841 of *Lecture Notes in Computer Science*, pages 604–615. Springer, 2006.

[122] Eric Prud'hommeaux and Andy Seaborne. SPARQL query language for RDF. www.w3.org/TR/rdf-sparql-query (accessed December 2008), January 2008.

[123] Anand S. Rao and Michael P. Georgeff. Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, *Principles of Knowledge Representation and Reasoning: Proc. of the Second International Conference (KR'91)*, pages 473–484. Morgan Kaufmann, San Mateo, CA, 1991.

[124] Jef Raskin. *The Humane Interface*. Addison-Wesley, 2000.

[125] Kate Raynes-Goldie. Aliases, creeping, and wall cleaning: Understanding privacy in the age of Facebook. *First Monday*, 15(1-4), 2010.

[126] David Recordon and Drummond Reed. OpenID 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital Identity Management (DIM)*, pages 11–16. ACM, 2006.

[127] Everett M. Rogers. *Diffusion of Innovations*. Free Press, 1995.

[128] David Rosenblum. What anyone can know: The privacy risks of social networking sites. *IEEE Security & Privacy*, 5(3):40–49, 2007.

[129] A. Dawn Shaikh and Kelsi Lenz. Where's the search? Re-examining user expectations of web objects. *Usability News*, 8(1), February 2006.

[130] Ben Shneiderman and Catherine Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison Wesley, 4. edition, 2004.

[131] Andrew Simpson. On the need for user-defined fine-grained access control policies for social networking applications. In *Proceedings of the Workshop on Security in Opportunistic and SOCial networks (SOSOC)*, pages 1:1–1:8. ACM, 2008.

[132] H. Jeff Smith, Sandra J. Milberg, and Sandra J. Burke. Information privacy: Measuring individuals' concerns about organizational practices. *MIS Quarterly*, 20(2):167–196, June 1996.

[133] Daniel J. Solove. Do social networks bring the the end of privacy? *Scientific American*, 299(3):78–83, September 2008.

[134] Sophos. Sophos Facebook ID probe shows 41% of users happy to reveal all to potential identity thieves. `http://www.sophos.com/pressoffice/news/articles/2007/08/facebook.html` (accessed September 2009), August 2007.

[135] Sarah Spiekermann, Jens Grossklags, and Bettina Berendt. E-privacy in 2nd generation e-commerce: privacy preferences versus actual behavior. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 38–47. ACM, 2001.

[136] Anna Cinzia Squicciarini, Mohamed Shehab, and Federica Paci. Collective privacy management in social networks. In *Proceedings of the 18th International World Wide Web Conference (WWW)*, pages 521–530. ACM, 2009.

[137] Sebastian Stober, Matthias Steinbrecher, and Andreas Nürnberger. A survey on the acceptance of listening context logging for MIR applications. In *Proceedings of the 3rd Workshop on Learning the Semantics of Audio Signals (LSAS)*, pages 45–57, 2009.

[138] The Stanford Encyclopedia of Philosophy. Privacy. `http://plato.stanford.edu/entries/privacy` (accessed November 2010), September 2006.

[139] Thomas Vander Wal. Folksonomy coinage and definition. `http://vanderwal.net/folksonomy.html` (accessed July 2010), February 2007.

[140] Hal R. Varian. *Cyber policy and economics in an Internet age*, chapter Economic Aspects of Personal Privacy, pages 101–109. Topics in regulatory economics and policy. Kluwer Academic Publishers, 2002.

[141] Tony Vila, Rachel Greenstadt, and David Molnar. Why we can't be bothered to read privacy policies models of privacy economics as a lemons market. In *Proceedings of the 5th International Conference on Electronic Commerce (ICEC)*, pages 403–407. ACM, 2003.

[142] Samuel Warren and Louis Brandeis. *The Right to Privacy.* Harvard Law Review, 1890.

[143] Alan F. Westin. *Privacy and Freedom.* The Bodley Head, 1967.

[144] Alan F. Westin. E-commerce and privacy: What net users want. Sponsored by Price Waterhouse and Privacy & American Business. Hackensack, NJ: P & AB, 1998.

[145] Robert Wetzker, Tansu Alpcan, Christian Bauckhage, Winfried Umbrath, and Sahin Albayrak. An unsupervised hierarchical method for automated document categorization. In *Proceedings of the International Conference on Web Intelligence (WI)*. IEEE Computer Society Press, 2007.

[146] Robert Wetzker, Till Plumbaum, Alexander Korth, Christian Bauckhage, Tansu Alpcan, and Florian Metze. Detecting trends in social bookmarking systems using a probabilistic generative model and smoothing. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008.

[147] Robert Wetzker, Carsten Zimmermann, and Christian Bauckhage. Analyzing social bookmarking systems: A del.icio.us cookbook. In *Proceedings of the ECAI 2008 Mining Social Data Workshop*, pages 26–30. IOS Press, 2008.

[148] David A. Wise and Mark Malseed. *The Google Story: Inside the Hottest Business, Media, and Technology Success of Our Time.* Random House Inc., 2006.

[149] Michael Wooldridge. *An Introduction to Multiagent Systems.* John Wiley & Sons, 2002.

[150] Tom Zeller. Link by link: Lest we regret our digital bread crumbs. `http://www.nytimes.com/2006/06/12/technology/12link.html` (accessed August 2009), June 2006.

[151] Elena Zheleva and Lise Getoor. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th International World Wide Web Conference (WWW)*, pages 531–540, 2009.

Glossary

90-9-1 Rule Jakob Nielsen's rule describing the distribution of user participation in social software, 10

AAA Authorization, Authentication, and Accounting, 81, 151
ACE Access Control Engine, 48, 95
ACL Access Control List, 48
ACP Access Control Policy, 48
AJAX Asynchronous JavaScript and XML, 10
API Application Programming Interface, 12
AR Application-specific Requirement, 77
Atom Syndication Format Atom Syndication Format is an XML-based exchange format for information, 52

Blank Slate State The Blank Slate State describes the problem of introducing newly registered users to an application. The goal is to avoid to displaying an empty page due to the lack of social connections and thus, content, 81, 148
Blog Word combination of Web and log. A type of Website (and software) that allows its author to easily publish articles on the Web. A typical feature to distinguish a blog from a normal Website is the reverse channel that is given to its readers through a comment feature, 1

CBIR Content-based Image Retrieval, 32
Communication Context Communication Context is the set of attributes that affect communication between two individuals, 7
Content-sharing Site Content-sharing Sites are Websites that provide a catalog of a particular content type that was created and shared by its users, 12
Contextual Integrity Contextual Integrity is maintained when both, the norms of appropriateness and the norms of distribution are respected, 28
CRUD Create, read, update, delete. Typical operations on entities of database-backed applications leading though their life-cycle, 18, 79
CW Collaborative Workspace, 41, 48

DHP Data Handling Policy, 49, 58
DiSo Distributed Social Networks, 50

ENISA European Network and Information Security Agency, 30
ER diagram Entity-relationship diagram, 83

F Feature, 78
Family A restrictive-by-default group and access control tool of the application that consists of members and Tips, 74
FOAF Friend-of-a-Friend, 52
Folksonomy A Folksonomy is a lightweight conceptual structure consisting of the entities user, tag and resource, which is created by users on, e.g., Social Bookmarking Services, 12
FQL Facebook Query Language, 30

GUI Graphical User Interface, 19, 48

HCI Human-Computer Interaction, 159
HTML Hypertext Markup Language, 19
HTTP Hypertext Transfer Protocol, 19, 50

IA Information Architecture, 22
IDP Identity Provider, 40, 151
ISO International Organization for Standardization, 19

LOD Linking Open Data, 153

Mashup Mashups are Websites that are to a significant degree combined from content or functions of third parties, 12
Microformats Microformats are a set of simple, open data formats to embed machine-readable information inside Web sites, 52, 153
Microformats Microformats are a set of simple, open data formats to embed machine-readable information inside Websites, 17
MVC Model View Controller programming pattern, 17

NLP Natural Language Processing, 160

OAuth OAuth is an open protocol to allow secure API authorization, 51

OpenID OpenID is a user-centric and URI-based identity system on the basis of the Hypertext Transfer Protocol (HTTP), 50

OpenSocial OpenSocial defines a common API for social applications across multiple Websites, 52

P3P Platform for Privacy Preferences, 25, 45, 46

Participation Inequality The distribution of user participation in social software by Jakob Nielson, 10

PubSubHubbub A simple, open, server-to-server web-hook-based publish/subscribe protocol as an extension to Atom and RSS, 53

R Requirement, 57

RDF Resource Description Framework, 17, 52, 153

RDFa Resource Description Framework in Attributes. RDFa specifies a set of XHTML attributes that can be used to embed machine-readable information inside Web sites, 52, 153

Real-Time Web Real-Time Web stands for real-time communication and spreading of information for syndication on the Web, 53

RSS Real Simple Syndication. RSS is an XML-based exchange format for information, 52

RSS Cloud RSS Cloud enables real-time push notifications or distributed publish/subscribe communication for news feeds, 53

SEO Search Engine Optimization, 32

SIOC Semantically-Interlinked Online Communities, 52

SNA Social Network Analysis, 14, 31, 52

SNS Social Networking Service. SNSs are Websites that focus their users' profile pages and their social connections, 1, 12

SOA Service-Oriented Architecture, 154

Social Bookmarking Service A Social Bookmarking Service is a system for empowering users to annotate URL-based resources with free tags, 12

Social Context Social Context is the set of attributes that refers to who is communicating with whom and what is the strength of relationship shared between them, 7, 59

Social Graph An uni-partite graph of users (nodes) and their connections (edges) to other users, 12

Social Media A new type of media consisting of Websites enabling user participation and socialization through content sharing and communication, 1, 9

Social Networking Service Social Networking Services (SNS) are Websites that focus their users' profile pages and their social connections, 12

SQ Survey Question, 131, 163

SQL Simple Query Language, 30

SSO Single-Sign-On, 50

Tag A keyword annotation of an item, 12

Tip A Tip is the central, location-based entity and communication spot of the presented application, 73

TLS Transport Layer Security, 30

UGC User-generated Content, 12, 145

UI User Interface, 25, 44

URI Uniform Resource Identifier, 50

UX User Experience, 44

W3C World Wide Web Consortium, 16, 45

Web 2.0 A new generation of Websites enabling user participation and socialization through content sharing and communication, 1, 9

Web of Data The Web of Data is a distributed web of interconnected data sets of semantically annotated data, 153

Web of Identities The Web of Identities is a distributed web about users: their personae, their social graphs and their assets. It provides privacy-preserving accessibility to user information, 157

Web of Services The Web of Services is a distributed web of semantically annotated services that freely accessible to be e.g., discovered, invoked, orchestrated or chained, 154

XFN XHTML Friends Network. A hyperlink-based approach to express human relationship, 52, 80

XHTML Extensible Hypertext Markup Language, 52

XML Extensible Markup Language, 10, 52

XMPP Extensible Messaging and Presence Protocol. An XML-based protocol for near-real-time instant messaging and the exchange of presence information, 53

XRI Extensible Resource Identifier, 50

XSRF Cross-site Request Forgery, 82

XSS Cross-site Scripting, 30