

Anwendungsbeispiele eines offenen Industrial APP Marketplace

Patrick Heidemann, Sascha Heymann, Nissrin Perez, Jan Alsters
Fraunhofer IOSB-INA

Institutsteil für industrielle Automation

Campusallee 1

32657 Lemgo

patrick.heidemann@iosb-ina.fraunhofer.de

sascha.heyman@iosb-ina.fraunhofer.de

nissrin.perez@iosb-ina.fraunhofer.de

jan.alsters@stud.th-owl.de

Abstract: Ein Anwendungsmarktplatz ist eine digitale Vertriebsplattform für Anwendungen (APPs), die meist im mobilen Kontext genutzt werden. Marktplätze wie der App Store von Apple oder Googles Play Store zeigen, wie florierend das Marktplatzgeschäft ist. Diese bieten Entwicklern eine Plattform für den Vertrieb ihrer Anwendungen und erreichen durch ein gemeinsames Ökosystem einen breiten Anwenderkreis. Ein Industrial APP Marketplace, kombiniert mit dem Industrial Internet of Things (IIoT), bietet im industriellen Bereich eine Lösung für eine schnellere Integration und Inbetriebnahme von industriellen Anwendungen in der Produktion aber z.B. auch Lösungen für den SmartCity Bereich. Dieses Paper zeigt zwei Anwendungsbeispiele eines herstellerneutralen Industrial APP Marketplaces, an den besondere Anforderungen an Interoperabilität, Usability und Sicherheit gestellt werden. Bei der Bewertung der Anwendungsbeispiele wird die gesamte Umsetzungskette vom Kauf bis zum Deployment einer APP auf ein Endgerät betrachtet. Bei den Anwendungsbeispielen handelt es sich um einen IIoT Ansatz, der eine direkte Internetverbindung der Endgeräte vorsieht und einen Edge Ansatz, bei dem Endgeräte in einem lokalen Netzwerk nur über ein gesichertes Edge-Gateway mit dem Internet kommunizieren können.

1 Einleitung und Motivation

Während im Konsumentenbereich Marktplätze breite Anwendung finden, ist im industriellen Bereich das Potenzial noch nicht erschlossen. Dies liegt an der fehlenden Architektur von interoperablen und einfachen Marktplätzen für industrielle Anwendungen [Hey21]. Dabei bietet das IIoT der Industrie neue intelligente Technologien als Lösung. Auch die Nachfrage für leistungsstarke Feldgeräte mit Edge-Computing Funktionen wächst stetig und damit auch die Anzahl der Hardware-Hersteller. Da in diesen Feldgeräten meist heterogene CPUs und eigene maßgeschneiderte Linux-Distributionen oder Echtzeitbetriebssysteme verwendet werden, ist es im Vergleich zu mobilen Geräten für unabhängige Entwickler eine Herausforderung interoperable APPs zu entwickeln. Aus diesem Grund bietet die Containervirtualisierung als fundamentale Abstraktionsschicht die Basis für APPs auf industriellen Geräten. Anwendungen werden in standardisierte Einheiten verpackt, welche als Container bezeichnet werden und zusätzlich zu dem Quellcode der Anwendung auch die Runtime, Systemtools und Bibliotheken beinhalten. Das ermöglicht den Software-Entwicklern APPs im Feld auf modulare Art und Weise plattformunabhängig zu entwickeln [MBP20]. Dies und eine Lösung für einen unkomplizierten Rollout der APPs auf die Feldgeräte sind die beiden grundlegenden Voraussetzungen für einen offenen industriellen APP-Marktplatz [Bar20].

2 Stand der Technik

2.1 Industrial APP Marketplace

Der Industrial APP Marketplace etabliert einen herstellerneutralen Marktplatz für I4.0-Komponenten zum Kauf und Verkauf von industrieller Hardware, Software und Dienstleistungen. Grundsätzlich versteht man unter APPs moderne Software-Anwendungen, die modular aufgebaut sind und ein benutzerfreundliches grafisches Interface haben, im Kontext eines Industrial APP Marketplace aber auch Dienste und Lösungen. Die aus einer Kooperation zwischen OWL Maschinenbau, der SmartFactoryOWL und der Open Industry 4.0 Alliance Community in 2020 entstandene Idee ist mittlerweile ein Zusammenschluss mehrerer Unternehmen und weiterer Akteure, die zum einen gemeinsam an der Plattform weiterentwickeln und zum anderen als APP Entwickler, Hardware Entwickler oder Integrator eigene Services auf dem Industrial APP Marketplace anbieten. Die gemeinsame Schnittmenge dieser drei Rollen und der Rolle des App-Benutzers bilden den Industrial APP Marketplace und damit ein hohes Innovationspotential. [Hey21]

2.2 OI4 Community App-Store

Der *Open Industry 4.0 Alliance Community App Store* ist das erste von der Allianz abgesegnete Projekt. Der App-Store wurde im Rahmen der Hannover Messe 2021 Digital vorgestellt. Die Idee hinter dem abgekürzt bezeichneten OI4 Community App-Store sei, dass das, was für mobile Endgeräte im Consumer-Bereich gut funktioniere, auch ein Modell für den industriellen Markt sein könnte. Auf Edge-Geräten laufen schon heute unterschiedliche Container-Applikationen diverser Hersteller, die häufig miteinander interagieren müssen. Als Non-Profit-Organisation erteilt die Allianz deshalb dem App-Store der Hilscher-Gesellschaft für Systemautomation mbH erstmals das Label „Open Industry 4.0 Alliance Community“. Der OI4 Community App-Store soll den Mitgliedern als eine übergreifende Verkaufsplattform für ihre Apps zur Verfügung gestellt werden, die ohne Investitions- und Unterhaltskosten auskommt. Der Fokus ist hier auf Apps, die auf der mit der Allianz konformen *Open Edge Computing Plattform* laufen. Der OI4 Community App-Store stehe ab Ende des dritten Quartals 2021 für Mitglieder der Open Industry 4.0 Alliance bereit, die dann getestete, Docker-kompatible sowie Hardware-unabhängige industrielle APPs beziehen können. [Man19]

2.3 PLCNext Store

Der PLCnext Store ist eine Online-Handelsplattform, in der Softwarefunktionen für die Steuerungsplattform sowohl von Phoenix Contact als auch von Drittanbietern zum Herunterladen bereitstehen. Das Angebot umfasst Funktionsbausteine, Funktionserweiterungen, Cloud Connectoren bis hin zu weiteren Laufzeitumgebungen wie z.B. Codesys, wodurch die PLCnext Steuerungen von Phoenix Contact auch mit weiteren Programmiersprachen erweitert werden können. Der PLCnext Store steht Software-Entwicklern als Handelsplattform für eigen entwickelte APPs zur Verfügung. Dabei legen die Entwickler den Preis für ihre Software-Lösungen fest, die Lizenzierung und Kaufabwicklung erfolgt über Phoenix Contact. Die Installation kann online oder offline auf ein Gerät übertragen werden. [DB20]

3 Anforderungen

Um einen Industrial APP Marketplace als Vertriebskanal in ein Geschäftsmodell integrieren zu können, muss dieser Marktplatz bestimmte Anforderungen erfüllen und sowohl Kunden, als auch Anbietern einen niederschweligen Einstieg ermöglichen. Maxim Barbarinow nennt in einem LinkedIn-Artikel¹ zehn Anforderung für einen Industrial APP Marketplace, die in folgender Tabelle gelistet werden:

¹<https://www.linkedin.com/pulse/industrial-iot-application-store-next-big-thing-maxim-babarinow/>

	Benennung	Beschreibung
1	Geräte Zertifizierung	Zertifizierung zur Kompatibilität zwischen Geräten und Anwendungen
2	Anwendungszertifizierung	Zertifizierung zur Robustheit, Performanz, Sicherheit und Datenschutz der Applikationen
3	Diversität der Endgeräte	Unterstützung einer breiten Palette von Geräten
4	1-Klick Deployment	Remote-Bereitstellung von Anwendungen für Geräteflotten direkt vom Marktplatz aus
5	Anwendungskonfiguration	Unterstützung von Anwendungskonfigurationen
6	Deployment und Anwendungsüberwachung	Statusüberwachung von Gerätefunktionen mit Hilfe von Diagnoseprotokollen
7	Anwendungslizenzierung	Bereitstellung von Lizenzierungen und Urheberschutzrechten für Anwendungen durch Marktplatzbetreiber
8	Geschäftsmodell-Flexibilität	Unterstützung verschiedener Lizenzmodelle und Zahlungsmethoden
9	Seller's Insights	Prognosen von Nachfragen und Trends zur Unterstützung der Verkäufer
10	Multi-Country Support	lokale und mehrsprachige Supportstruktur

Lt. Barbarinow müssen technische Zertifizierungen vorangestellt werden, um die Funktionalität neuer Anwendungen auf vorhandene Geräte sicherzustellen. Die Installation sollte durch offene Prozessarchitekturen, die verschiedene Protokolle unterstützen, sowie durch die Ermöglichung eines 1-Klick Deployments vereinfacht werden. Durch die Unterstützung von Startkonfigurationen und einer Deployment Kontrolle kann sichergestellt werden, dass das Deployment erfolgreich war. Lizenzbedingungen für einen Urheberschutz und für die Abbildung verschiedener Lizenzmodelle und daraus folgender Zahlungsmethoden sollten auf dem Marktplatz geschaffen werden. Interessant sind für die Anbieter zudem Verkaufsanalysen, welche durch die Erhebung und Analyse von entstehenden Daten entstehen. Und zuletzt ist es essentiell, einen breiten Markt zu schaffen, indem die Plattform international mit entsprechenden Sprachen, Währungen, Zahlungsmethoden etc. aufgebaut wird. Neben diesen zehn von Barbarinow gelisteten Anforderungen nennt er weitere Erfolgskriterien, wie eine gute Auffindbarkeit des Marktplatzes, ein positives Nutzererlebnis, eine Partnerverwaltung mit Wiederverkaufsoptionen, Rezensionen und Empfehlungen, Supportangebote und Unterstützung der Finanzbuchhaltung [Bar20]. Auch on-demand Services und mögliche Rollbacks der Anwendungsbereitstellung werden als sinnvolle funktionale Ergänzungen des Marktplatzes angesehen [Bar20]. Eine wichtige Voraussetzung, welche die Open Industry 4.0 Alliance benennt, ist die Nutzung einer offenen auf RAMI 4.0 basierten Architektur mit den Bausteinen 1. Device Connectivity, 2. Edge Operator Cloud, 3. Cloud Central und 4. einem entsprechenden Dienstleistungsangebot. [?] Nur durch die Verwendung offener Standardschnittstellen auf Grundlage der Industrie 4.0 Verwaltungsschale kann lt. der Open Industry 4.0 Alliance eine automatisierte Integration von Assets – das sogenannte „Asset Automatic Onboarding“ realisiert werden. [Man19]

4 Technologien

4.1 Eclipse HawkBit

HawkBit ist ein domänen-unabhängiges Backend-Framework zum Ausrollen von Software-Updates und Firmware auf IoT-Geräten. Dies ist eine open-source Software der Eclipse Foundation und besteht aus einem Update-Server, welcher das Update- und Rollout-Management, Content-Delivery und ein Geräte- und Software-Repository besitzt. Zusätzlich werden drei Schnittstellen in Form von WEB-APIs und ein Webinterface bereitgestellt. Der Software-Rollout kann entweder direkt über die *Management-API* oder über das Webinterface (*Management-UI*) verwaltet werden. Für die direkte Geräteintegration bietet *hawkBit* eine einfache REST-API (*Direct-Device-Integration API*). Eine Indirekte Geräteintegration (andere Konnektivitätsschicht) wird mit der *Device-Management-Federation-API* bereitgestellt. Diese ist erforderlich wenn ein eingeschränktes Gerät keine TLS/HTTP-Verbindung verarbeiten kann, aber ein Standard-Geräteverwaltungsprotokoll wie OMA-DM oder LWM2M unterstützt. [Fou21]

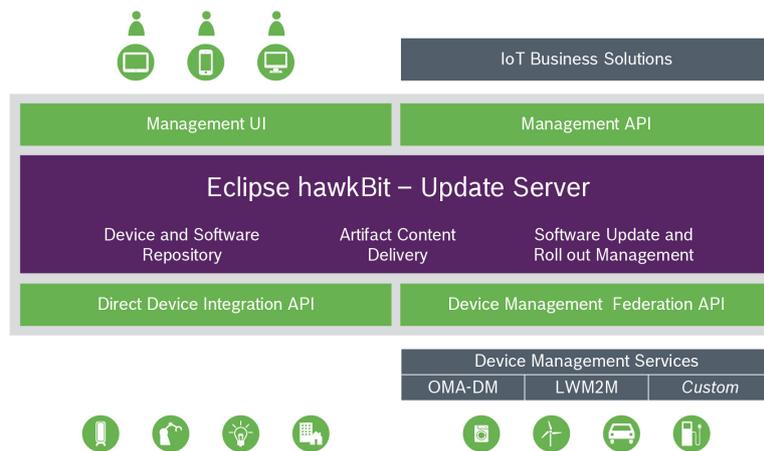


Abbildung 1: HawkBit Architektur

4.2 Docker

Docker ist eine quelloffene Software zur Isolierung von Anwendungen mit Hilfe von Container-Virtualisierung. Docker-Container sind standardisierte Einheiten, welche neben der Applikation auch Bibliotheken, Systemtools und Laufzeit beinhalten. Dadurch ist es möglich, Anwendungen in jeder Umgebung schnell bereitzustellen und zu skalieren. Ein Docker-Image bildet die Basis für einen Container und kann in einem Repository wie DockerHub gespeichert und verwaltet werden. [Edu21]

5 Anwendungsbeispiele

In diesem Kapitel werden zwei verschiedene Ansätze für eine Deployment-Lösung von APPs auf Feldgeräten vorgestellt.

5.1 Grundlagen

5.1.1 Artefakt

Ein Artefakt ist in diesem Kontext eine Datei, welche bei dem Rollout-Prozess auf die Geräte geladen wird. Hierbei handelt es sich nicht um eine Applikation, sondern lediglich um eine Manifest-Datei, welche alle relevanten Information über eine APP enthält. Dazu gehören das Docker Basis-Image, der APP Name, benötigte Ports und weitere Parameter. Als APP wird dann der lauffähige Docker-Container bezeichnet.

5.1.2 hbloader

hbloader ist ein erweiterter HawkBit-Client, welcher in der Programmiersprache Python geschrieben ist und auf dem rauc-hawkBit-Client aufgebaut ist. Der vorkonfigurierte Client meldet sich über die *Direct-Device-Integration-API* am *HawkBit* an und prüft zyklisch über die *Management-API*, ob ein neues Deployment aussteht. Sobald dies der Fall ist, wird das entsprechende Artefakt heruntergeladen und mit Hilfe einer Prüfsumme auf Echtheit geprüft. Anschließend wird ein Docker-Container mit den angegebenen Parametern gestartet.

5.1.3 hbpull

hbpull ist eine Software zur Koppelung einer öffentlich erreichbaren und einer lokalen HawkBit-Instanz. Ein HawkBit-Client meldet sich als Gerät bei der Cloud-HawkBit-Instanz an und überträgt das Artefakt bei einem Rollout an die lokale HawkBit-Instanz und legt die APP dort als Software-Modul an.

5.2 IIoT-Ansatz

Bei dem IIoT-Ansatz ist jedes Feldgerät, in diesem Fall eine IoT-Sensorbox, direkt mit dem Internet verbunden.

Nach dem Erwerb einer APP wird das entsprechende Artefakt an die *HawkBit*-Instanz übertragen und dort als Software-Modul angelegt. Über das Management-UI kann das Modul dann in eine Distribution gelegt und diese auf verfügbare Feldgeräte ausgerollt werden. Der *hbloader* installiert dann die APP als Docker-Container auf dem jeweiligen Gerät. Das benötigte Docker-Image wird von einer öffentlich erreichbaren Docker-Registry heruntergeladen.

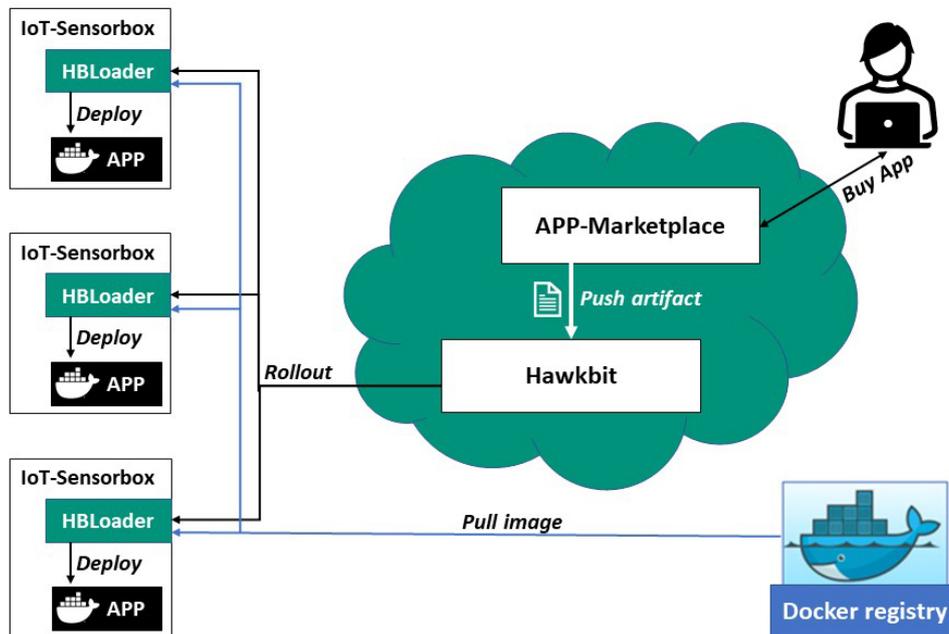


Abbildung 2: APP-Deployment bei dem IIoT-Ansatz

5.3 EDGE-Ansatz

Bei dem Edge-Ansatz sind die Feldgeräte in einem separaten Maschinen-Netz. Ein Edge-Gerät mit zwei Netzwerk-Schnittstellen ist mit diesem und über eine Firewall mit dem Internet verbunden. Auf dem Edge-Gerät läuft eine lokale *HawkBit*-Instanz und eine Docker-Registry.

Nach dem Erwerb einer APP wird diese an das Edge-Gerät übertragen. Dafür ist *hbpull* bei dem im Marketplace integrierten HawkBit als Zielgerät angemeldet und prüft zyklisch auf neue Deployments. Sobald eine neue APP verfügbar ist, wird das Artefakt heruntergeladen, in das lokale *HawkBit* übertragen und dort als Software-Modul angelegt. Von dort aus kann die APP dann über ein *Management-UI* auf die lokalen Feldgeräte ausgerollt werden. Bei der Installation wird das benötigte Docker-Image über die lokalen Docker-Registry heruntergeladen. Diese ist als *Pull-Through-Cache* konfiguriert. Ein Docker-Daemon auf einem der Feldgeräte fordert bei der lokalen Docker-Registry ein Image an. Anschließend wird überprüft, ob das angeforderte Image bereits im lokalen Repository vorhanden ist. Wenn dies der Fall ist, wird das Image direkt an das entsprechende Gerät zurückgegeben. Andererseits wird es von einer öffentlich erreichbaren Registry heruntergeladen und für künftige Anfragen lokal gespeichert.

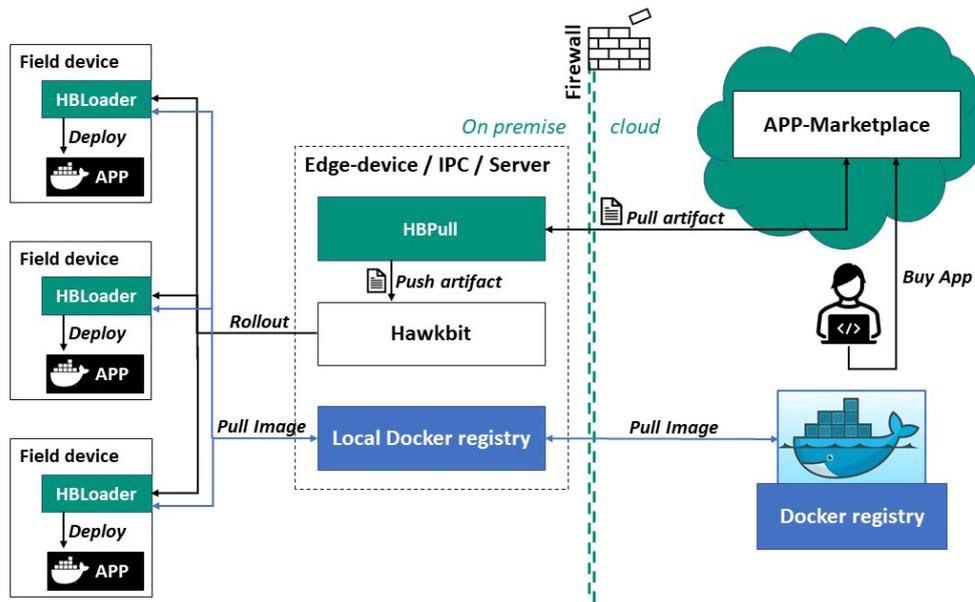


Abbildung 3: APP-Deployment bei dem Edge-Ansatz

6 Geräte- und APP-Sicherheit

Um zu gewährleisten, dass die im Marketplace angebotenen APPs sicher sind, sollten diese einen Prüfprozess durchlaufen. Dabei muss festgestellt werden, ob die bereitgestellte Anwendung robust, performant und sicher ist. Hierfür kann ein Tool wie Nexus-Container¹ verwendet werden. Dieses bietet Sicherheit für den gesamten Lebenszyklus eines Containers, in dem es auf Schwachstellen überprüft und eine Konformitätsprüfung durchführt. Außerdem kann der gesamte Netzwerkverkehr überwacht werden. [Son21] Den Entwicklern müssen zusätzlich Richtlinien und Ressourcen zur Verfügung gestellt werden. [Bar20]

6.1 Lizenzmodelle

Da die Entwicklung von Software ein komplexer, zeitaufwändiger und teurer Prozess ist, führt Software-Piraterie zu großen Einnahmeverlusten für Technologieunternehmen. Um die Nutzung und Weitergabe zu regeln und illegale Kopien zu verhindern, muss der Marktplatz sowohl einen Lizenzierungsmechanismus als auch Lizenzmodelle für Anwendungen unterstützen. [Bar20]

In der folgenden Tabelle sind drei verschiedene Lizenzmodelle aufgelistet:

¹<https://de.sonatype.com/nexus/container>

Modell	Beschreibung
Freeware	APPs können kostenlos und unbegrenzt benutzt werden
On Premise	Die Kauflizenz. Für die unbegrenzte Nutzung einer APP wird einmalig gezahlt
Pay per Use	Die Nutzung einer APP wird basierend auf der Nutzung bestimmter Ressourcen abgerechnet

Die Lizenzen könnten entweder per signierte JSON Web Token(JWT²) auf die Geräte gebracht werden oder über einen Lizenzserver verwaltet werden.

7 Zusammenfassung

Dieser Beitrag hat zwei Anwendungsbeispiele für einen offenen Industriellen APP Marketplace vorgestellt. Dafür wurden verschiedene Anforderungen betrachtet und basierend darauf zwei Konzepte entwickelt, wie Applikationen auf Feldgeräte ausgerollt und installiert werden können. Bei dem IIoT-Ansatz ist jedes Feldgerät direkt mit dem Internet verbunden und kann über eine Cloud-HawkBit Instanz verwaltet werden. Für den Fall, dass die Feldgeräte aus Sicherheitsgründen nur innerhalb eines lokalen Maschinennetzes erreichbar sind, wird ein zusätzliches Edge-Gerät benötigt. Dieses ist sowohl mit dem internen Maschinennetz als auch über eine Firewall mit dem Internet verbunden.

Die Docker Container Technologie ist ein wichtiger enabler, um einen universellen und offenen Marktplatz zu betreiben. Sie ermöglicht, Software Anwendungen (APPs) anbieterunabhängig auf verschiedensten Plattformen z.B. Steuerungen oder Edge-Gateways zu betreiben. Damit sind Hard- und Software-Lieferant nicht mehr identisch, was wiederum in der industriellen Digitalisierung ganz neue Geschäftsmodelle ermöglicht. Ein weiterer genutzter „quasi“ Standard neben Docker ist *HawkBit* als Rollout Technologie, um Software auf die Feldgeräte, Edge-Koppler, Industrie-PCs, Server und Kubernetes Cluster auszurollen.

²<https://jwt.io/>

Literaturverzeichnis

- [Bar20] Maxim Barbarinow. Industrial iot application store: The next big thing?, 2020.
- [DB20] Sebastian Gerstl Daniel Buschatzky. Applikationen mit dem plcnext store einfacher und schneller umsetzen, 2020.
- [Edu21] IBM Cloud Education. Docker, 2021.
- [Fou21] Eclipse Foundation. Eclipse hawkbit, 2021.
- [Gö21] Ulrike Götz. Ein app-store für die industrie – eine initiative aus der open industry 4.0 alliance community, 2021.
- [Hey21] Sascha Heymann. Industrial app marketplace, 2021.
- [Man19] Digital Manufacturing. Was das neue framework der open industry 4.0 alliance kann, 2019.
- [MBP20] Jonas Kulawik Maj-Britt Peters. Volkswagen brings additional partners to industrial cloud, 2020.
- [mSA15] myfactory Schweiz AG. 4 lizenzmodelle für erp-software im vergleich, 2015.
- [Son21] Sonartype. Nexus container integration, 2021.