

# Webbasiertes Framework und Apps für die Montage

## Entwicklung eines Frameworks und Beispiel-Apps für Montageverfahren in Fertigungsbetrieben

Leif Goldhahn,  
Dorit Bock und  
Robert Eckardt, Mittweida  
Marcus Friedemann, Dresden\*)

Die Entwicklung eines industrieorientierten Frameworks zur mobilen Anreicherung, Auswertung und synergetischen Applikation von Daten zwischen innovativen Applikationen (Apps) und ERP-Systemen wird in diesem Beitrag beschrieben. Dabei werden die Anwendersicht (Beispiele für Szenarien und Apps) und die Programmiersicht (Framework mit Anbindung der Apps) betrachtet. \*\*)

### Einleitung und Motivation

Wie die Studie der Trovarit AG (2019) zeigt, sind Unternehmen nach wie vor mit der mobilen Einsetzbarkeit der ERP-Software nicht zufrieden, wobei sich das Angebot (z. B. MEiAPPS Montagebericht [1]) langsam an die Erwartungshaltung anpasst. Weitere wichtige Anliegen der Unternehmen sind die Daten- und Informationssicherheit sowie das Datenmanagement, die gerade im Bereich der mobilen Einsetzbarkeit eine große Rolle spielen [2]. Mittels der Entwicklung eines praxistauglichen und industrieorientierten Frameworks im Verbundforschungsprojekt Framework Enriched Data Assembly – FrEDA soll ein Austausch zwischen Mini- oder Kleinanwendungen (Apps) und datenbankbasierter Software (ERP-System) entstehen. Dabei sollen die Reduzierung großer Datenmengen im mobilen Bereich und die Eliminierung von Defiziten, wie z. B. unklare Datenhoheit, Ungenauigkeit und fehlende Transparenz, eine wichtige Rolle spielen. Das Framework zielt auf Anwendungen in Montageprozessen und der dazugehörigen Logistik.

Im Beitrag werden zunächst Motivation, Begrifflichkeiten und Grundlagen geklärt. Es folgt das konzeptionelle Vorgehen zur Entwicklung des Frameworks.

Die Einsatzszenarien zeigen dann anwendungsorientierte Beispiel-Apps, die sich auf die Datenbereitstellung für die Montage, eine einfache multimediale Datengenerierung in der Montage und logistische Aufgaben an den Schnittstellen der Montage zur Logistik fokussieren. Diese Apps wurden als Referenz für Anwender erstellt und mit dem entwickelten Framework FrEDA verknüpft, welches abschließend auch aus Programmersicht vorgestellt wird.

### Grundlagen

#### Montage

Die Montage in industriellen Unternehmen verantwortet viele unterschiedliche Verfahren. Ein breites Spektrum von Bauteilen und Baugruppen wird häufig manuell oder hybrid (Mensch und Anlage) mittels diverser Werkzeuge, Vorrichtungen und Hilfsstoffe gefertigt, getestet und verpackt. Logistische Aufgaben spielen vor allem bei der Materialbereitstellung, der Aufgabendisposition, der Personaleinsatzplanung und der Betriebsdatenerfassung eine große Rolle [3].

Daraus resultieren ausgesprochen viele Informationsflüsse, die teilweise individuell und spontan ablaufen. Diese Bedarfe werden jedoch von standardisierten IT-Systemen nicht gedeckt. Hier liegt vielmehr das Einsatzpotential kleiner, möglichst mobiler Anwendungen (Apps). Diese müssen im Sinne der Konsistenz und Verfügbarkeit der Daten zukünftig über ein Framework mit den größeren Systemen verknüpft werden. Die Vielfalt der Datenarten und potentieller Anwendungen gilt es zu berücksichtigen.

#### Framework

„Ein Framework ist ein durch den Softwareentwickler anpassbares oder erweiterbares System kooperierender Klassen, die einen wiederverwendbaren Entwurf für einen bestimmten Anwendungsbereich implementieren. Es besteht aus konkreten und abstrakten Klassen, die Schnittstellen definieren“ [4]

#### \*\*Danksagung

Die Autoren danken dem Europäischen Fonds für regionale Entwicklung und dem Freistaat Sachsen für die finanzielle Unterstützung des Forschungsprojektes „Entwicklung eines neuartigen Frameworks zur mobilen Anreicherung, Auswertung und synergetischen Applikation von Daten für Montageverfahren in Fertigungsbetrieben“ (FrEDA). Das Forschungsprojekt wird vom Zentrum für innovative Arbeitsplanung und Arbeitswissenschaft (InnArbeit) der Hochschule Mittweida und der CIMPCS GmbH Dresden bearbeitet.

#### \*) Hinweis

Bei diesem Beitrag handelt es sich um einen von den Mitgliedern des ZWF-Advisory Board wissenschaftlich begutachteten Fachaufsatz (Peer-Review).



Europäische Union



Europa fördert Sachsen.  
EFRE  
Europäischer Fonds für regionale Entwicklung



Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des vom Sächsischen Landtag beschlossenen Haushaltes.

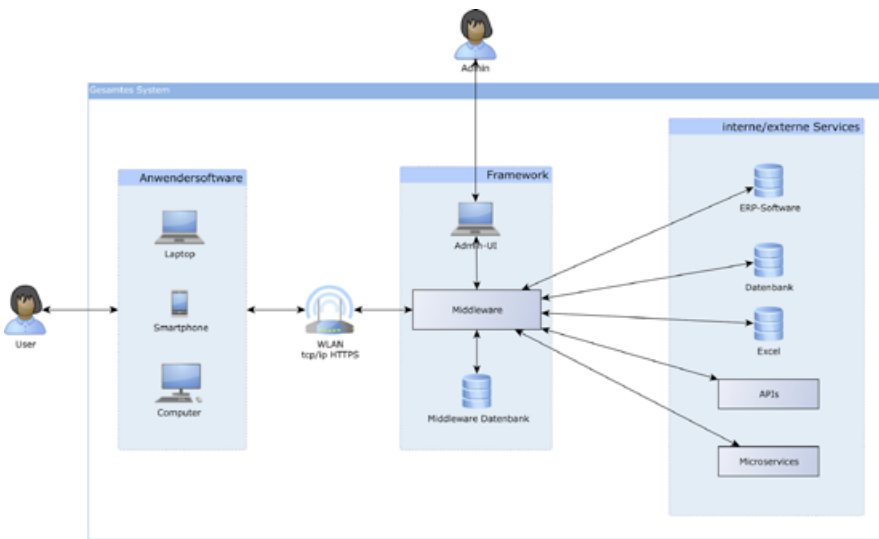


Bild 1. Systematisierung Framework im Projekt FrEDA

Derzeitige Entwicklungen zeigen, „dass sich die aktuell im Einsatz befindlichen ERP-Lösungen verhältnismäßig schwer tun, wenn es beim ERP-Einsatz heißt, zu jeder Zeit, an jedem Ort und über jedes Endgerät“ zu arbeiten. Eine erforderliche Umstellung von ERP-Software auf eine App-artige Nutzungscharakteristik bringt offenbar eine Vielzahl technologischer Herausforderungen mit sich (z.B. Plattformunabhängigkeit, kontext-sensitive Benutzeroberfläche, Use Case-spezifische „Applifizierung“ umfassender Business-Software-Lösungen)“ [2]. Webbasierte Apps sind aber bereits in verschiedenen Bereichen zum Beispiel des Vertriebs, des Engineering oder der Produktionsplanung zu finden. Deshalb ist es für ERP-Anbieter zukünftig essentiell, ein Framework im Sinne komplexer Middleware zwischen datenbankbasiertem ERP-System und den zahlreich zu erwartenden Kleinanwendungen zu entwickeln. Möglichst zeitnah darf es im Unternehmen nur eine verbindliche Datenbasis je Information geben.

Im Forschungsprojekt wurde das Framework modular aufgebaut, um ein auf den jeweiligen Einsatzfall (Industrieanforderungen) anpassbares und erweiterbares System zu schaffen. Dabei wurden Klassen zum Beispiel für das Administrations User Interface (Admin-UI) und die Middleware-Datenbank (Bild 1) implementiert, welche unabhängig von mehreren Apps gemeinsam genutzt werden können. Eine Anbindung an die Unternehmens-ERP-Datenbank als Datenquelle und zur automatischen Erkennung der Daten-Schemata wird über eine GraphQL-standardisierte Schnittstellensprache realisiert. Die Frei-

gabe von Schreib- und Leserechten sowie der zur Verfügung stehenden Funktionalität des Frameworks für die Apps wird über die gleiche Schnittstelle ermöglicht.

**App**

Von der acatech – Deutsche Akademie der Technikwissenschaften wird gefordert, bedarfsgerechte Informationen in Apps anzubieten und gerade auch für den Fertigungsbereich aufgabenspezifische Apps zu entwickeln [5].

Die Entwicklung von Apps erfolgt meist um eine einfache Kernfunktion, wie bspw. das Auslesen eines QR- (Quick Response) Codes zu erfüllen. Apps sind im Regelfall klein und hochspezialisiert, informationsorientiert sowie mobil und mit Touch-

Oberfläche nutzbar. Außerdem sind sie einfach bedienbar und plattformspezifisch [6]. Es gibt drei Klassifizierungen von Apps: die nativen Apps, die WebApps und die hybriden Apps. Native Apps können auf alle Hardwarefunktionen des mobilen Endgerätes zugreifen, aber sie sind speziell für das Betriebssystem des jeweiligen Endgerätes konzipiert. WebApps werden über einen Browser gestartet und sind deshalb unabhängig vom mobilen Endgerät, aber sie haben keinen vollen Zugriff auf die Hardware des Gerätes. Die hybriden Apps sind eine Mischform aus nativen und WebApps [7]. Im Projekt wurden WebApps implementiert.

Die Vorteile dieser WebApps sind u.a. die Ausführbarkeit auf allen mobilen Plattformen, meist ohne Anpassungsaufwand mittels der plattformübergreifenden Unterstützung in modernen Browsern und die geringeren Entwicklungskosten. Dadurch ist eine browser-unabhängige Verwaltung, Nutzung und Aktualisierung durch den Plattformbetreiber möglich. Ein weiterer Aspekt der für die Nutzung von WebApps spricht, ist die Offline-Speicherung von Daten und die damit verbundene Nutzung der einmal geladenen WebApp ohne ständigen Internetzugang, was im industriellen Umfeld teilweise notwendig wird, da meist keine hundertprozentige Netzverfügbarkeit gewährleistet werden kann. Es gibt aber Nachteile, wie die Begrenzung auf den Funktionsumfang, den der Browser bereitstellt, den eingeschränkten Zugriff auf Hardwarefunktionen (z.B. Kamera) oder dass die WebApps etwas langsamer sind als native Anwendungen [7, 8].

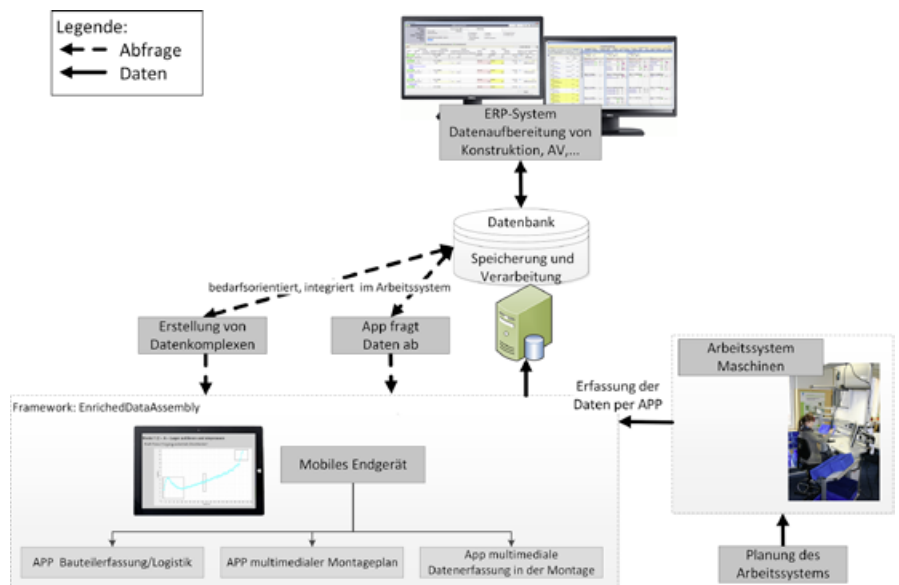


Bild 2. Framework für den Forschungsansatz FrEDA

## Konzeptionelle Entwicklung des Frameworks

Das übergeordnete technische Arbeitsziel im Forschungsprojekt ist die Entwicklung eines industrieorientierten Frameworks als Middleware zwischen den unternehmensinternen verteilten (heterogenen) Systemen und externen Webservices bzw. frei verfügbaren Kleinanwendungen. Ziel des Frameworks ist es, die Komplexität bei der Entwicklung problemlösungsorientierter Apps, welche auf der Grundlage der industriellen Anforderungen auftreten, deutlich zu reduzieren. Weiterhin werden durch die Entwicklung Fragen der Operabilität, Kompatibilität, Schnittstellen, mobilen Standards, Datenverfügbarkeit, Datenübertragung, Mehrmedialität und Datenhoheit proaktiv beantwortet.

Mittels der zu konzipierenden Beispiel-Apps multimedialer Montageplan, multimediale Datenerfassung in der Montage und im Bereich Bauteilerfassung/Logistik werden die Funktionalität sowie die Praxistauglichkeit des Frameworks für Fertigungsunternehmen validiert. Innerhalb des Frameworks werden Schnittstellen generiert, die die Verbindungen zwischen einem ERP-System (Datenbasis aller Unternehmensressourcen), den Apps und den Arbeitssystemen herstellen (Bild 2). Das REFA-Arbeitssystem [8] wurde als Modell-Grundlage für das FrEDA-Framework verwendet. Dabei sind die wesentlichen Elemente des Arbeitssystems, der Mensch und das Betriebsmittel, durch die wesentlichen Elemente des Frameworks (Funktionen, Programmstrukturen) ersetzt.

Dazu waren zunächst die Definition und die Festlegung von Anforderungen an das zu entwickelnde Framework notwendig. Gemeinsam mit potentiellen Anwenderfirmen wurden Fragen an das Framework formuliert:

- Wie ist das Framework aufgebaut?
- Welche Funktionen übernimmt das Framework?
- Welche Daten werden im Framework gespeichert bzw. angezeigt?
- Mit welchen Programmiersprachen erfolgt die Umsetzung des Frameworks?
- Wie erfolgt die Datenübertragung vom Framework zur App bzw. vom Framework zum ERP-System (Aspekt Datensicherheit/Datenschutz)? Welche Daten werden wo gespeichert bzw. zwischengespeichert?

Tabelle 1. Einsatzszenarien und Apps

| Einsatzszenario                | Anzeige weiterer Artikeldaten/ Arbeitsplandaten mittels Auslesen des QR-Codes auf Auftragsbegleitkarte  | Mit mobilen Endgerät erstellte Daten (Bilder, Videos, Dateien) ins ERP-System übernehmen  | Mobilis Anzeigen und Bearbeiten einer Liste offener Vorgänge mit Datenbasis im ERP-System |
|--------------------------------|---|---|---|
| Arbeitsaufgabe                 | Baugruppen- und Endmontage von Sondergetrieben  | Montage von komplexen Baugruppen, Geräten oder Werkzeugen   | Auslagern und Ausbuchen von Artikeln  |
| Beispiel                       | Schaltgabel auf Antriebswelle montieren   | Multimediale Daten zum Arbeitsschritt hinzufügen  | Kommissionieren von Bauteilen für einen Montageauftrag                                    |
| App auf Tablet oder Smartphone | <p><b>Multimedialer Montageplan</b></p> <p>QR-Code von Auftragsbegleitkarte scannen</p> <p>Anzeige des multimedialen Arbeitsplans mit weiteren Informationen/Daten zum Arbeitsgang/Arbeitsschritt</p> | <p><b>Erfassung von Montagedaten</b></p> <p>Button „Bild aufnehmen/Daten hinzufügen“ anklicken</p> <p>Bild/Video mit Tablet aufnehmen oder Daten von Tablet hochladen</p> <p>Button „speichern“ anklicken</p> | <p><b>Lagerausgang</b></p> <p>Artikelzeile aktivieren</p> <p>Ausbuchung durch Scannen</p> |
| ERP-System                     | Bereitstellung der Daten für Framework  | Übernahme der Daten in ERP-System (bei Offline-Modus → automatische Übergabe der Daten an ERP-System, sobald Tablet wieder im Online-Modus)   | Übernahme der Daten in ERP-System   |

- Wie und wann findet der Datenaustausch statt, wenn im Offline-Modus gearbeitet wird?

Darüber hinaus musste geklärt werden, wie die Benutzerinteraktionen ablaufen bzw. realisiert werden sollen. In der daran anschließenden Lastenhefterstellung wurden Erfordernisse und Lösungsansätze sowohl für das Framework (z. B. Nutzer-/Rollenmanagement, Datei-Upload, Gerätefreischaltung) als auch für die Apps (z. B. Login-Maske, Benutzeroberfläche, Menü) definiert [10].

### ■ Einsatzszenarien und erste Apps

Aktuelle mobile Geräte bieten viele integrierte Schnittstellen und Hardwarekomponenten, die im Framework FrEDA und den Apps für unterschiedliche Zwecke genutzt werden. Das Schnittstellenmanagement zwischen den Apps, dem Framework und dem ERP-System wird hardwareseitig mittels Kommunikationsschnittstellen, wie z. B. WLAN oder Bluetooth, realisiert. Es können auch Dienste, wie z. B. Hot-Spots oder Geolocation, verwendet werden, welche aber in den bisherigen Beispielen noch keinen Einsatz fanden. Hardwarekomponenten, wie z. B. Mikrofon oder Kamera, werden direkt in der App genutzt. Die Aktivierung von Sensoren zur Temperatur- oder Beschleunigungsmessung ist denkbar. Eine vollständige Übersicht der häufigsten Schnittstellen (Kommunikation, Sensoren, allgemeine Dienste) von mobilen Endgeräten ist im Forum IT@Automation [6] aufgeführt.

Auf der Basis von entwickelten Einsatzszenarien werden im Forschungsprojekt gegenwärtig WebApps erstellt. In den ersten beiden Einsatzszenarien (Tabelle 1 und Bild 3) wird die Getriebemontage an einem manuellen Montagearbeitsplatz als Referenzbeispiel für die Montage in Einzel-, Klein- und Mittelserienfertigung genutzt. Das dritte Szenario bezieht sich auf Lagerbuchungen an den Schnittstellen zwischen Montage und Logistik.

Die Ein- und Ausgabe des Frameworks FrEDA erfolgt einerseits zum ERP-System, wo die von den Apps benötigten Daten (z. B. Artikelnummer, -bezeichnung für App Multimedialer Montageplan) bzw. die mit den Apps erstellten Daten (z. B. Bilder, Videos von App Erfassung von Montagedaten) zur Verfügung gestellt bzw. gespeichert werden. Die Daten aus dem ERP-System werden im Framework weiterverarbeitet und der entsprechenden App zur Verfügung gestellt. Andererseits werden die mit der App zwischengespeicherten Bilder/Videos über das Framework an das ERP-System übermittelt.

Die App Lagerausgang stellte ebenfalls einen Anwendungsfall dar, um das Framework und die Darstellung der Browser-App für die Nutzer zu demonstrieren und prinzipielle Anforderungen an die Gebrauchstauglichkeit umzusetzen. Gerade die Montage erfordert zahlreiche Lagervorgänge, die im Wareneingangs-, Kommissionier- oder Versandlager sachgerecht ein- und auszubuchen sind. Stellvertretend für viele weitere Szenarien ähnlicher Art, handelt es sich im Kontext be-



Bild 3. App Multimedialer Montageplan auf Tablet am Beispiel Getriebe (Montageschritt Schaltgabel auf Antriebswelle montieren)

trieblicher Aufgaben häufig um eine Liste offener Vorgänge und einer damit verbundenen Aktion auf die aktivierte Datenzeile. Im konkreten Fall dieser Anwendung werden alle auszulagernden Artikel in einer Listenansicht gezeigt. Der Mitarbeiter, der die Ware übergibt, muss lediglich die gewünschte Zeile aktivieren und bestätigen, dass die Ware jetzt ausgebucht wird und physisch das Lager verlassen hat. Alternativ kann die Ware durch Scannen des gedruckten Lieferscheins über die integrierte Kamera oder einen Barcode-Scanner des Mobilgeräts gebucht werden. Im Gegensatz zum klassischen Desktop-Ansatz wird die Bedienbarkeit durch die Reduzierung auf den konkreten Anwendungsfall erreicht. Die Integration der gerätespezifischen Funktionen (z. B. Kamera oder Scanner) reduziert die Komplexität der Bedienung weiter.

Mit dem Ziel der Dezentralisierung der Anwendungen werden also spezialisierte Dienste und Apps entwickelt, die je nach Geschäftsprozess beliebig miteinander kombiniert und wiederverwendet werden können. Durch diese Entkopplung kann jeder benötigte Geschäftsprozess durch Kombination der Dienste ermöglicht werden. Als Erweiterung des Funktionsumfangs der kurz skizzierten Lagerausgangsanwendung kann auf diese Weise eine Anwendung Artikelverzeichnis hinzugefügt werden. Diese Anwendung ist in sich selbständig und kann auch aus dem Startmenü aufgerufen werden, um den kompletten Artikelstamm eines Teiles, einer Baugruppe oder eines Produk-

tes aufzulisten. In Kombination mit dem Lagerausgang kann der Nutzer weitere relevante Informationen zum Artikel anzeigen, beispielsweise dessen Masse, Dimension, Lagerbestand oder ein Foto. Auf gleichem Wege ließe sich die Verwaltung der Zielorte integrieren.

### Realisierung des FrEDA-Frameworks

Ein Framework für industrielle Zwecke zu erstellen bedeutet aus Sicht der Programmierung sowohl die technologische Basis in Form von modularen Bausteinen und Schnittstellen, als auch die geeignete Bedienbarkeit für die Endanwender im mobilen Nutzungskontext zu gewährleisten. Aus Programmierer-Sicht war es wichtig, ein Modell zu erstellen, mit welchem das Framework realisiert werden kann. Dazu mussten zuerst die Voraussetzungen geschaffen werden, um die für die Apps relevanten Daten mittels des Frameworks aus dem ERP-System verfügbar zu machen (Schnittstellendefinition). So waren für die Apps beispielsweise Daten wie Artikelnummer, Artikelbezeichnung, Arbeitsgangbezeichnung, Arbeitsgangbeschreibung, Bilder, Werkzeuge und QR-Codes interessant. Mit der zweiten App sollten die aufgenommenen und hochgeladenen Daten wieder korrekt im ERP-System gespeichert werden. Dabei musste auch der Offline-Betrieb der App berücksichtigt werden (Datensynchronisation). Der Aufruf der Apps sollte mittels Navigation am mobilen Gerät

oder über QR-Codes am Arbeitsplatz möglich sein. Die Entwicklung erfolgte mit dem Responsive Webdesign, um anpassungsfähige Web-Apps an die Displaygröße des mobilen Gerätes zu erstellen und programmieren zu können [11].

Für die Umsetzung des Frameworks wurden die Abfragesprachen GraphQL und REST (Representational State Transfer) jeweils auf ihrer Implementierungseignung im Forschungsprojekt getestet, wobei sich GraphQL als geeignet erwies (s. u.).

Die WebApps können sich über GraphQL mit der Middleware im Framework verbinden, um Daten zu lesen, zu schreiben oder Logik und Prozessketten anzustoßen. Die PostgreSQL-Datenbank des ERP-Systems PRODAT kann mithilfe des GraphQL-Tools PostGraphile über eine GraphQL-API (application programming interface) nach außen veröffentlicht werden und CRUD- (Create, Read, Update, Delete) Funktionalitäten zur Verfügung stellen. Diese, nach dem Datenbankschema generierte GraphQL-API, wird in die eigene GraphQL-API eingebettet, mit einer Logik ausgestattet und ggf. sind Felder und Funktionsaufrufe umzubenennen.

Des Weiteren wurde ein Sicherheitskonzept erarbeitet, um die unternehmensinternen Anwendungen vor Angriffen zu schützen. Dafür gibt es u. a. die 2-Faktorauthentifizierung. Faktor Eins ist das Gerät, welches in dem Admin-User-Interface des Frameworks hinterlegt werden muss, damit es Zugriff auf die Anwender-App bekommt. Faktor Zwei ist die Nutzeranmeldung selbst, in welcher sich der Nutzer per E-Mail-Adresse und Passwort oder per RFID-Chip authentifizieren muss. Die Nutzer werden vorab in der Admin-UI angelegt und autorisiert. Optional kann für jeden Nutzer ein RFID-Code hinterlegt werden, welcher eine einfache Anmeldung per Chip ermöglicht. Nachdem die Nutzer angelegt wurden, müssen diese den zuvor konfigurierten Unternehmensrollen zugewiesen werden, sodass entsprechende Module im Frontend (Benutzeroberfläche) freigeschaltet werden. Die Benutzerrechtsteuerung und Modulfreischaltung ist mit den Rollen verknüpft. Die verwendeten Technologien, basierend auf GraphQL, ermöglichen eine Granularität der Autorisierung bis auf Feldebene in den jeweiligen Modulen. Zu Beginn der Umsetzungsphase wird sich vorerst nur auf Module konzentriert, an welchen eine Autorisierung stattfindet.

Im vorliegenden Framework werden mehrere Datenquellen in eine einzige API zusammengeführt. Dabei werden die Relationen zwischen den verschiedenen Datenquellen (z.B. Datenbanken, APIs oder Dokumente) und deren Feldern definiert. Die Transformierung der Felder bspw. durch Umbenennung der Feldnamen oder Delegation von Anfragen in externe Services wird ebenfalls realisiert.

Die Vorteile des FrEDA-Frameworks liegen darin, dass der Middleware (Bild 4 mittlerer Teil) entwicklungsseitig einmalig Datenquellen (Data Provider, Bild 4 unterer Teil) bekannt gegeben werden müssen, die dann die Basis für eine einheitliche Systemumgebung schaffen. Dabei können Anwendungen unterschiedlicher Datenherkunft über die FrEDA-Schaltzentrale integriert werden. Das sind beispielsweise das Unternehmensverwaltungssystem (ERP), datenbankbasierte Tools oder externe Webservices. Für diese Datenquellen werden Abfragen (Queries) definiert, die entweder in sich eine Geschäftslogik enthalten oder als eigenständige Module bzw. Microservices, gekapselt zwischen Abfrage und Befehlsausführung zur Datenmanipulation, integriert werden. Während andere HTTP-basierte und ressourcenorientierte Schnittstellen-Architekturen (z.B. REST) nur sehr strikte Abfragen ermöglichen bzw. den Zugriff oft auf eine einzelne Ressource beschränken, zeichnen sich GraphQL-Queries (Bild 4 oberer Teil) durch hohe Flexibilität aus. Es kann gezielt definiert werden, welche Datenfelder abgefragt werden sollen und es gibt keine Einschränkung für die Anzahl abgefragter Ressourcen. Die große Stärke von GraphQL zeigt sich darin, dass man mit einer einzigen Abfrage Zugriff auf alle benötigten Daten erhält. Es werden weder mehr noch weniger Daten als benötigt über die Schnittstelle versendet, was der Performance der Applikationen zugutekommt.

Aus Gründen der Datensicherheit müssen die Abfragen den Nutzerrollen zugeordnet werden, die Zugriff auf die Daten erhalten dürfen. Die Abfragen werden schließlich in einer Frontend-App durch entsprechende vordefinierte HTML-Komponenten zur Ausgabe gebracht. Ein Werkzeugkasten an festgelegten Klassen sorgt für Einheitlichkeit der Apps. Für die Anwender soll die durchgehend gleiche Bedienung vor allem schnelles Zurechtfinden ermöglichen. Weiterhin werden über diese Klassen eine schnelle Ent-

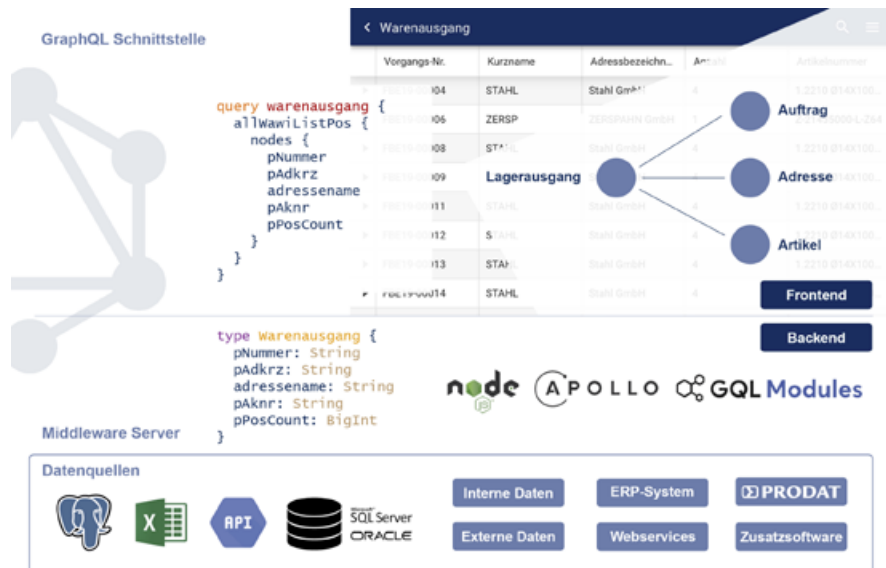


Bild 4. Sichten auf die Integration einer App im Framework mit Data Provider, Queries und Frontend-App

wicklung und die Kapselung aller notwendigen Funktionen erreicht. Ein Beispiel dafür ist die Anzeige einer Zeichnung zum Artikel im PDF-Format. Ein vordefinierter Container für Vorschau-Dokumente bringt alle Funktionen für Preview und Download mit.

Zusätzlich zu den visuellen Komponenten gibt es Backend-Funktionen, welche sich beispielsweise um die Validierung der Eingaben der Nutzer kümmern, die Authentifizierung von Nutzern und Rollen, die Mehrsprachigkeit, globale Suchmechanismen, die einfache Integration neuer Abfragen und perspektivisch die Offline-Funktionalität der Apps ermöglichen.

Ein Vorteil, der sich aus dieser Architektur ergibt, ist dabei, dass das Entwicklerteam für das Backend vollkommen unabhängig von dem Team agieren kann, das für die Frontend-Komponente verantwortlich ist.

## Zusammenfassung und Ausblick

Der Beitrag beschreibt das allgemeine Vorgehen bei der Entwicklung des industrieorientierten Frameworks FrEDA zur mobilen Anreicherung, Auswertung und synergetischen Applikation von Daten zwischen innovativen Applikationen (Apps) und ERP-Systemen. Ziel des Frameworks ist es, die Komplexität bei der Entwicklung und Integration problemlösungsorientierter Apps, welche durch die vielfältigen industriellen Anforderungen auftreten, deutlich zu reduzieren. Dazu

musste zunächst eine fundierte Anforderungsanalyse, inkl. Lastenhefterstellung durchgeführt werden. Darüber hinaus wurden auf der Grundlage dieser Analyse und der daraus resultierenden Framework-Entwicklung Fragen der Operabilität, Kompatibilität, Schnittstellen, des mobilen Standards, Datenverfügbarkeit, Datenübertragung, Mehrmedialität und Datenhoheit beantwortet. Der aktuelle Stand der Entwicklungsarbeiten umfasst die szenarienbasierte Konzeption beispielhafter industrienahe Apps aus dem Fertigungsbereich, speziell der Montage, welche weiter technologisch validiert werden müssen, womit die Funktionalität und die Praxistauglichkeit des Frameworks für Fertigungsunternehmen getestet wird. Im weiteren Forschungsprojektverlauf erfolgt die Weiterentwicklung der Schnittstellen des Frameworks, welche die Verbindungen zwischen einem ERP-System und den Apps herstellen werden.

## Literatur

1. MEiAPPS: Montagebericht HINZ Steuerungs- & Datentechnik e.K.. Online unter <http://www.meiapps.org/produkte/applikationen/montagebericht.html>, 2020
2. Trovarit AG: ERP in der Praxis 2018/2019 - Anwenderzufriedenheit, Nutzen, Perspektiven. Aachen, 2019
3. Goldhahn, L.; Müller-Eppendorfer, K: Holistic Planning of Material Provision for Assembly In: Proceedings of the 20<sup>th</sup> Congress of the International Ergonomics Association (IEA 2018) Volume VIII: Ergonomics and Human Factors in Manufacturing, Agriculture, Building and Construction, Sustainable Development and Mining. IEA Flor-

ence; Springer Nature Switzerland AG, 2019, S. 258 ff.

DOI: 10.1007/978-3-319-96068-5\_29

4. Balzert, H.: Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb. 3. Aufl., Spektrum Akademischer Verlag Heidelberg 2011  
DOI: 10.1007/978-3-8274-2246-0
5. Schuh, G.; Anderl, R.; Gausemeier, J.; ten Hompel, M.; Wahlster, W. (Hrsg.): Industrie 4.0 Maturity Index – Die digitale Transformation von Unternehmen gestalten, acatech – Deutsche Akademie der Technikwissenschaften, 2017, S. 27, 41, 60
6. Forum IT@Automation: App-Entwicklung in der Industrie: Grundlagen und Entscheidungshilfe, Industriekreis Smart Devices, VDMA, 2014
7. Franke, F.; Ippen, J.: Apps mit HTML5 und CSS3. Galileo Press, Bonn, 2012, S. 22 ff.
8. BITKOM Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V. (Hrsg.): Apps & Mobile Services – Tipps für Unternehmen. 2. erw. Aufl., Berlin 2014
9. Autorenteam REFA/REFA-Institut (Hrsg.): REFA Kompakt-Grundausbildung 2.0 – Das Basis-Know-how in Industrial Engineering. 2. überarbeitete Aufl., 2013, S. 64
10. Goldhahn, L.; Bock, D.: Bedienkonzepte maschineller Anlagen und deren virtuelle Validierung, In: VAR<sup>2</sup> 2017 – Realität erweitern. In: Tagungsband zum 4. Fachkonferenz zu VR/AR-Technologien in Anwendung und

Forschung an der Professur Werkzeugmaschinen und Umformtechnik, Technische Universität Chemnitz, 2017, S. 129 ff.

11. Ertel, A.; Laborenz, K.: Resonsive Webdesign – Anpassungsfähige Websites programmieren und gestalten. Galileo Press, Bonn 2014, S. 22 ff.

### ■ Die Autoren dieses Beitrags

Prof. Dr.-Ing. Leif Goldhahn, geb. 1967, lernte den Beruf des Werkzeugmachers, studierte Maschinenbau und promovierte zum Dr.-Ing. an der Technischen Universität Chemnitz. Beruflich tätig war er in Forschung und Industrie. Ende 2001 wurde er als Professor für Produktionsinformatik an der Fakultät Ingenieurwissenschaften der Hochschule Mittweida berufen. Dort gründete er 2003 das Institut InnArbeit – Zentrum für innovative Arbeitsplanung und Arbeitswissenschaft. 2018 konnte hier die Trainingsfabrik 4.0 als komplexes Testfeld aufgebaut werden. Seine Forschungsgebiete sind Arbeitsplanung, Rationalisierung von Fertigungsprozessen/Arbeitsorganisation sowie Ergonomie/Wissensmanagement.

Dipl.-Inf. (FH) Dorit Bock, geb. 1975, studierte an der Hochschule Mittweida Wirtschaftsinformatik, war danach als Innovationsassistentin und Softwareentwicklerin tätig und arbeitet jetzt als Forschungsingenieurin am Institut InnArbeit der Hochschule Mittweida.

Robert Eckardt, M.Eng., geb. 1983, absolvierte eine Ausbildung zum technischen Assistenten für Informatik, studierte an der Hochschule Mittweida Maschinenbau und ist Forschungsingenieur am Institut InnArbeit der Hochschule Mittweida.

Dipl.-Wirt.-Inf. (FH) Marcus Friedemann, geb. 1982, studierte Wirtschaftsinformatik an der HTW Dresden und ist als Projektleiter bei der Firma CIMPCS zuständig für Produktmanagement und Qualitätssicherung der ERP-Software PRODAT.

### ■ Summary

**Development of Web Based Framework and Apps for Assembly Processes in Manufacturing Companies.** The development of an industry-oriented framework for the mobile enrichment, evaluation and synergetic application of data between innovative applications (apps) and ERP systems is described in this paper. The user view (examples for scenarios and apps) and programmer view (framework with connection to the apps) are considered.

### Bibliography

DOI 10.3139/104.112291

ZWF 115 (2020) 5; page 321–326

© Carl Hanser Verlag GmbH & Co. KG

ISSN 0947–0085