ENTWICKLUNG SMARTER SERVICES IN PRODUZIERENDEN UNTERNEHMEN

Am Beispiel einer Demonstrator-Entwicklung

inside it‘s OWL | 30.03.2022

Folienmaster its OWL?

Titel und Logo IMPRESS in Folienmaster einfügen

Logo it‘s OWL in Folienmaster einfügen

Muster-Folie IEM

FIWARE

Diebold Nixdorf

Weidmüller

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Fraunhofer IEM

AGENDA

FÜR DIE ENTWICKLUNG SMARTER SERVICES IN PRODUZIERENDEN UNTERNEHMEN

Agenda um HNI Punkt ergänzen

1. Konzept / Ideenfindung
2. Spezifikation
3. (Rahmengebende Faktoren)
4. Implementierung

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Fraunhofer IEM

AGENDA

FÜR DIE ENTWICKLUNG SMARTER SERVICES IN PRODUZIERENDEN UNTERNEHMEN

1. Konzept / Ideenfindung
2. Spezifikation
3. (Rahmengebende Faktoren)
4. Implementierung

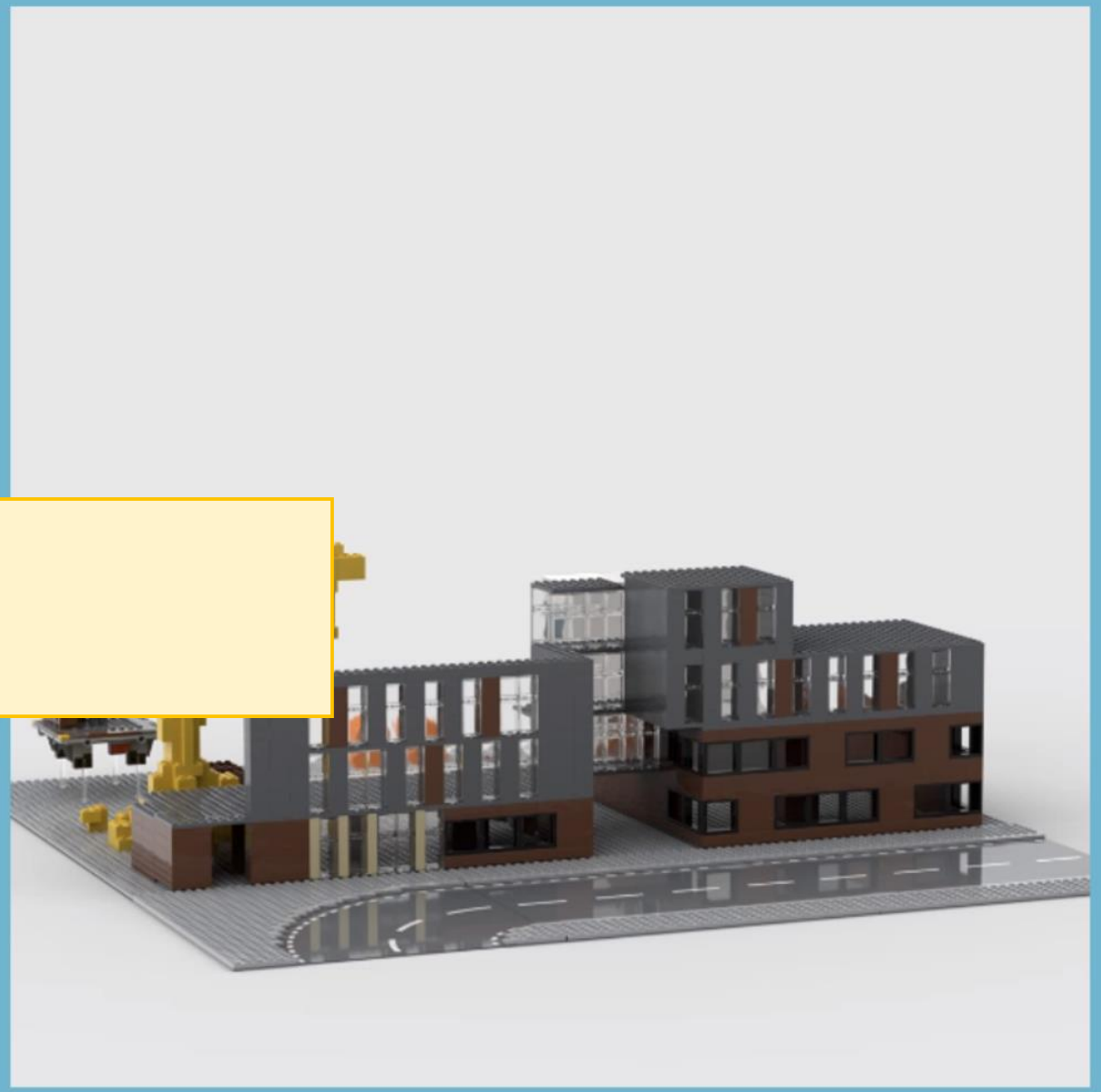
HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

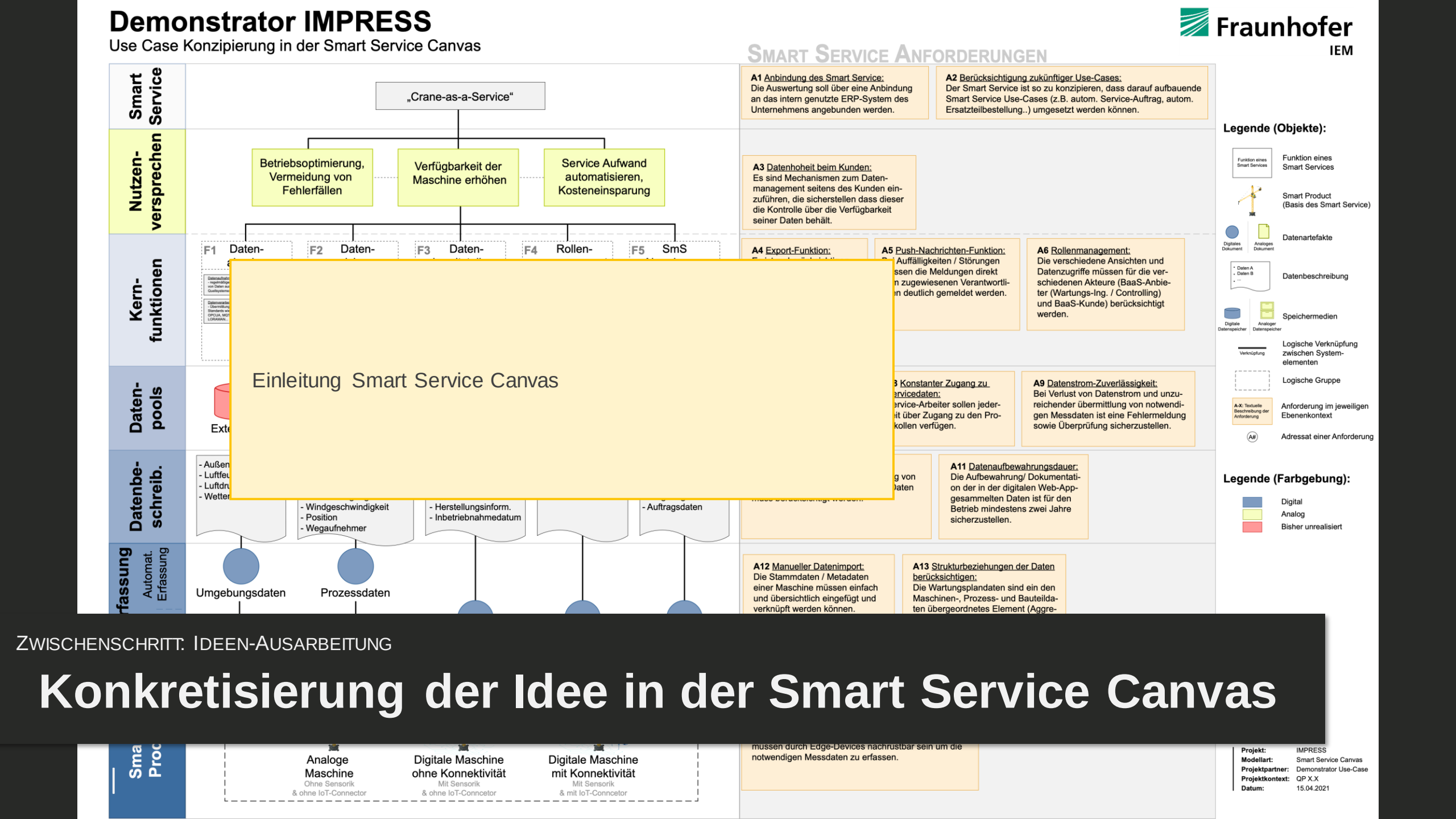
Fraunhofer
IEM

Pyramide

- Wie auf Idee gekommen

Auf voriger Folie





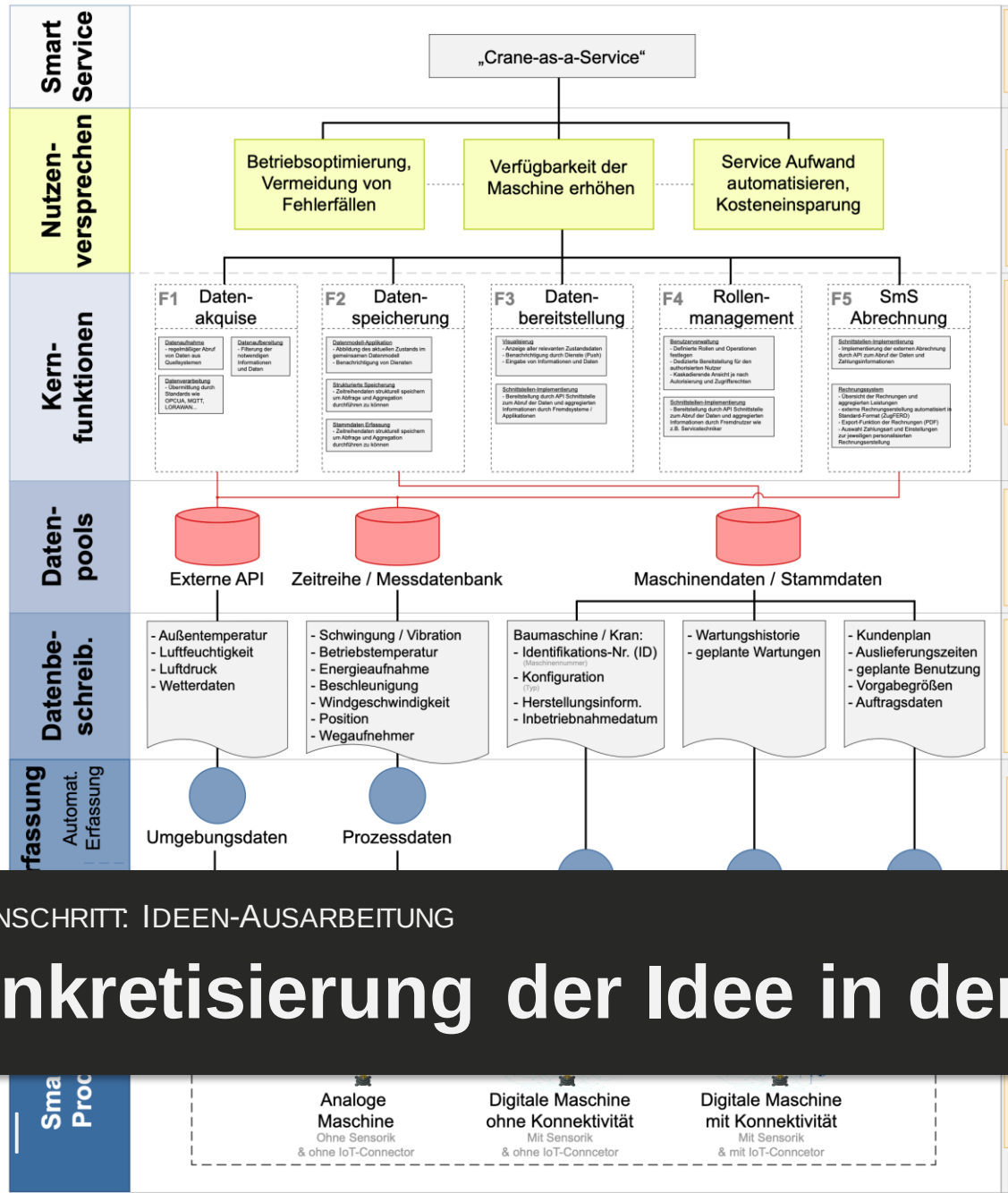
ZWISCHENSCHRITT: IDEEN-AUSARBEITUNG

Konkretisierung der Idee in der Smart Service Canvas

Demonstrator IMPRESS

Use Case Konzipierung in der Smart Service Canvas

Fraunhofer IEM



Smart Service Anforderungen

A1 Anbindung des Smart Service:
Die Auswertung soll über eine Anbindung an das intern genutzte ERP-System des Unternehmens angebunden werden.

A2 Berücksichtigung zukünftiger Use-Cases:
Der Smart Service ist so zu konzipieren, dass darauf aufbauende Smart Service Use-Cases (z.B. autom. Service-Auftrag, autom. Ersatzteilbestellung..) umgesetzt werden können.

A3 Datenhoheit beim Kunden:
Es sind Mechanismen zum Datenmanagement seitens des Kunden einzuführen, die sicherstellen dass dieser die Kontrolle über die Verfügbarkeit seiner Daten behält.

A4 Export-Funktion:
Es ist zu berücksichtigen, dass der Kunde unabhängig von seinem genutzten System eine Funktion für das Erstellen von Exports (z.B. .pdf) benötigt.

A5 Push-Nachrichten-Funktion:
Bei Auffälligkeiten / Störungen müssen die Meldungen direkt dem zugewiesenen Verantwortlichen deutlich gemeldet werden.

A6 Rollenmanagement:
Die verschiedene Ansichten und Datenzugriffe müssen für die verschiedenen Akteure (BaaS-Anbieter (Wartungs-Ing. / Controlling) und BaaS-Kunde) berücksichtigt werden.

A7 Revisionssicherheit:
Die Daten der Messreihen müssen transaktionssicher, synchron und versioniert übertragen und abgespeichert werden.

A8 Konstanter Zugang zu Servicedaten:
Service-Arbeiter sollen jederzeit über Zugang zu den Protokollen verfügen.

A9 Datenstrom-Zuverlässigkeit:
Bei Verlust von Datenstrom und unzureichender übermittlung von notwendigen Messdaten ist eine Fehlermeldung sowie Überprüfung sicherzustellen.

A10 Schutz von Prozessdaten:
Eine geeignete Form der Behandlung von manipulationssicheren und validen Daten muss berücksichtigt werden.

A11 Datenaufbewahrungsdauer:
Die Aufbewahrung/ Dokumentation der in der digitalen Web-App gesammelten Daten ist für den Betrieb mindestens zwei Jahre sicherzustellen.

A12 Manueller Datenimport:
Die Stammdaten / Metadaten einer Maschine müssen einfach und übersichtlich eingefügt und verknüpft werden können.

A13 Strukturbeziehungen der Daten berücksichtigen:
Die Wartungsplandaten sind ein den Maschinen-, Prozess- und Bauteildaten übergeordnetes Element (Aggre-

müssen durch Edge-Devices nachrüstbar sein um die notwendigen Messdaten zu erfassen.

Legende (Objekte):

- Funktion eines Smart Services
- Smart Product (Basis des Smart Service)
- Datenartefakte (Digitales Dokument, Analoges Dokument)
- Datenbeschreibung (Daten A, Daten B)
- Speichermedien (Digitale Datenspeicher, Analoger Datenspeicher)
- Logische Verknüpfung zwischen Systemelementen (Verknüpfung)
- Logische Gruppe
- Anforderung im jeweiligen Ebenenkontext (A-X: Textuelle Beschreibung der Anforderung)
- Adressat einer Anforderung (A#)

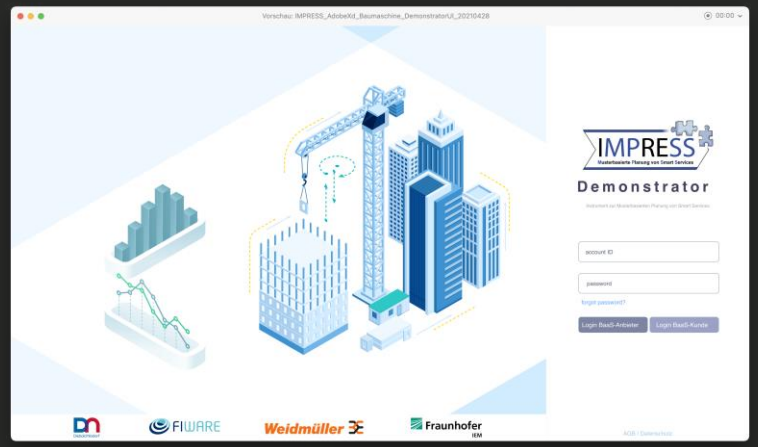
Legende (Farbgebung):

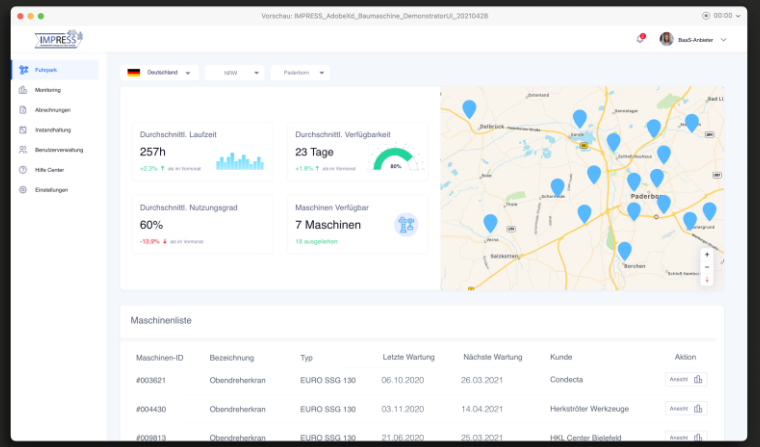
- Digital
- Analog
- Bisher unrealisiert

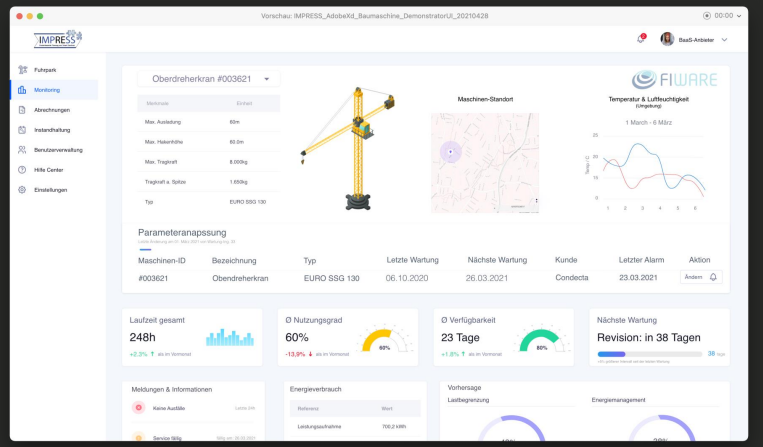
Projekt:	IMPRESS
Modellart:	Smart Service Canvas
Projektpartner:	Demonstrator Use-Case
Projektkontext:	QP X.X
Datum:	15.04.2021

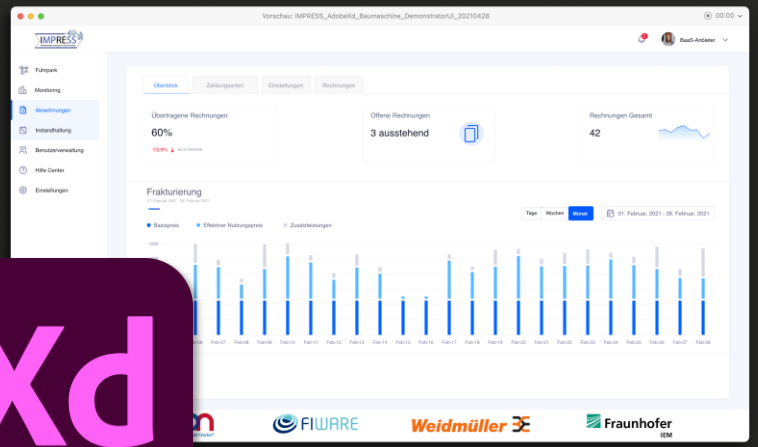
ZWISCHENSCHRITT: IDEEN-AUSARBEITUNG

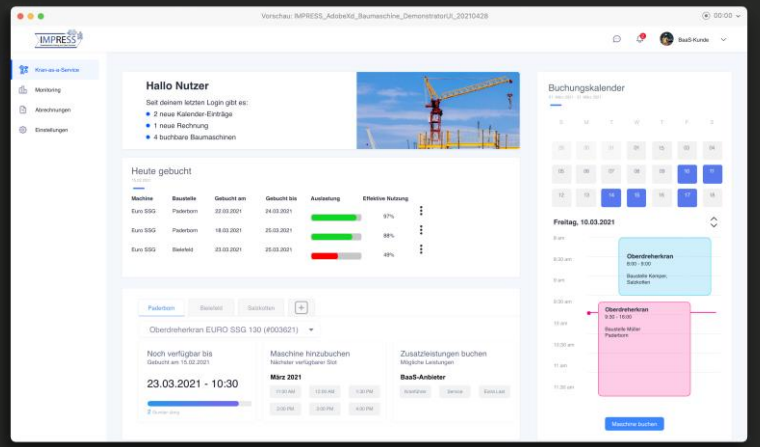
Konkretisierung der Idee in der Smart Service Canvas

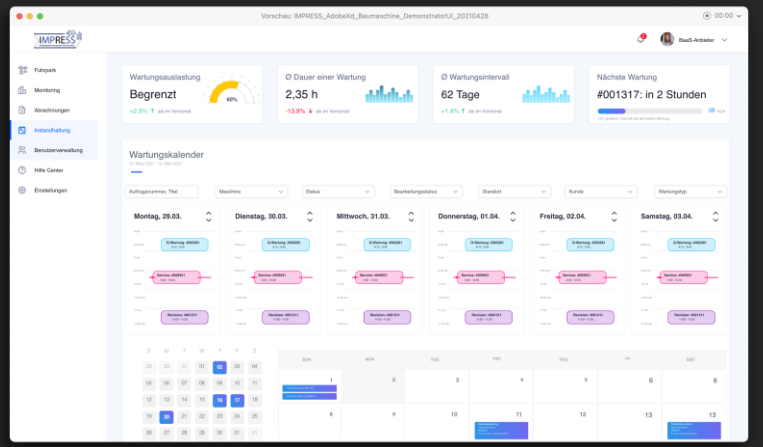


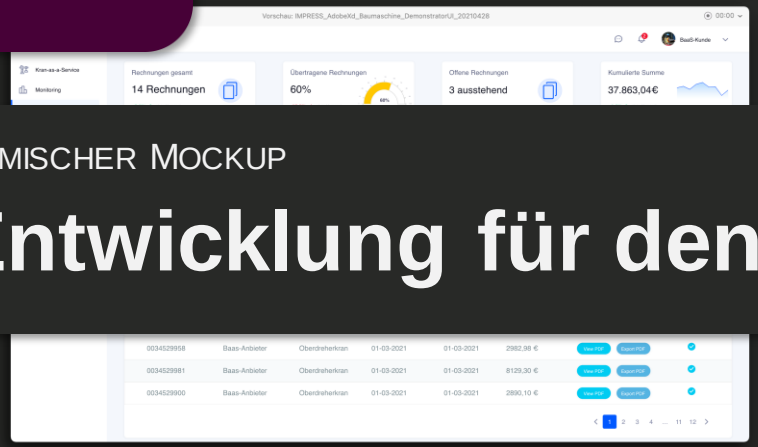


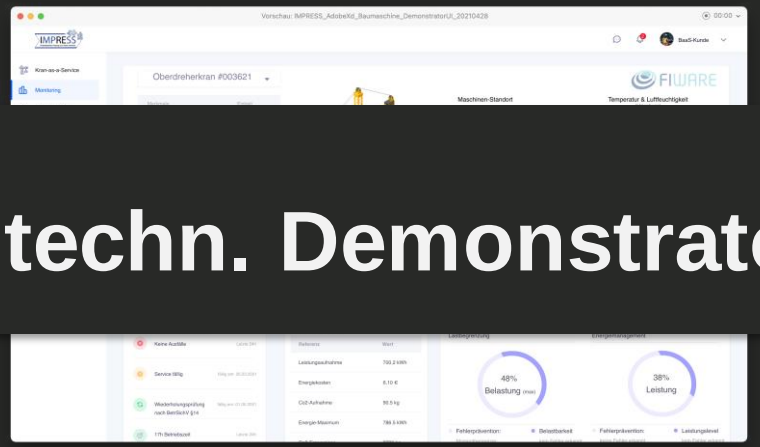


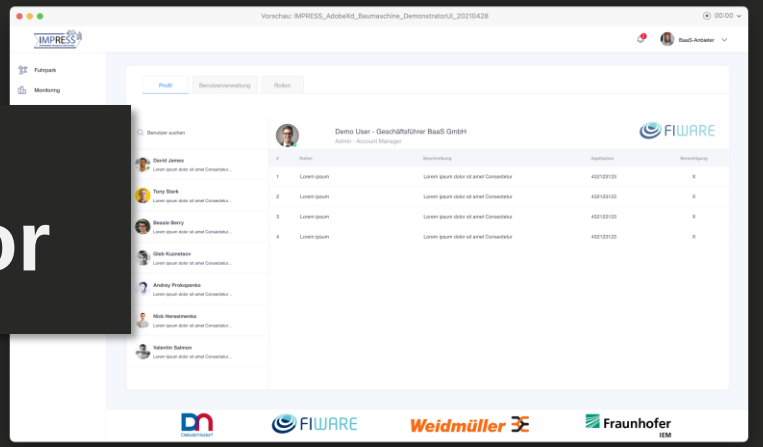












DYNAMISCHER MOCKUP

Entwicklung für den techn. Demonstrator

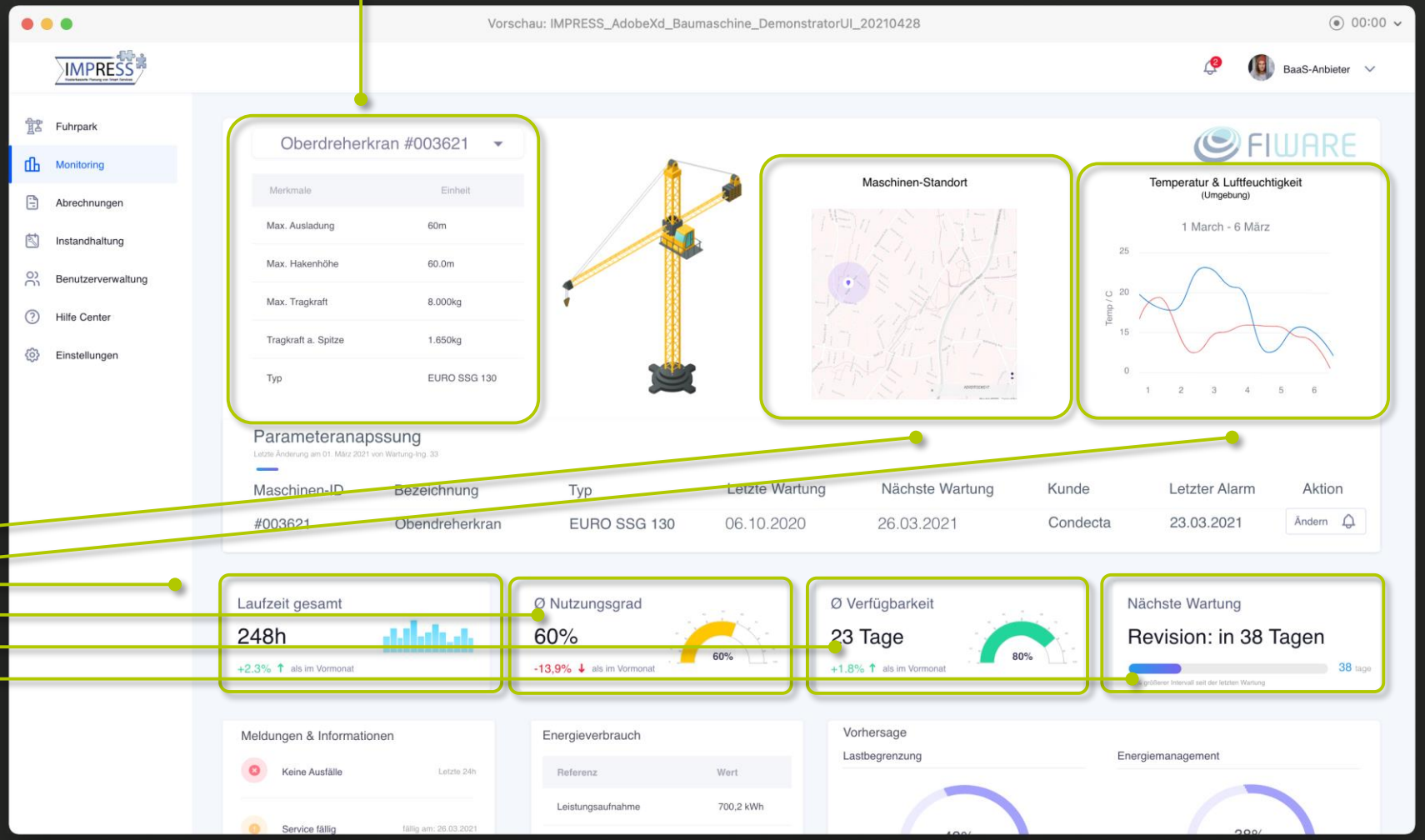
Konzept / Ideenfindung

Grundfunktionen idefinieren

- Zeigt plakativ die Komplettlösung auf und gibt Einblick in die künftige **User Experience**.
- Ist **Diskussionsgegenstand** für die Gesamtlösung bevor technische Details zu entscheiden sind.
- Lässt indirekt bereits den nötigen **Entwicklungsumfang** einschätzen.

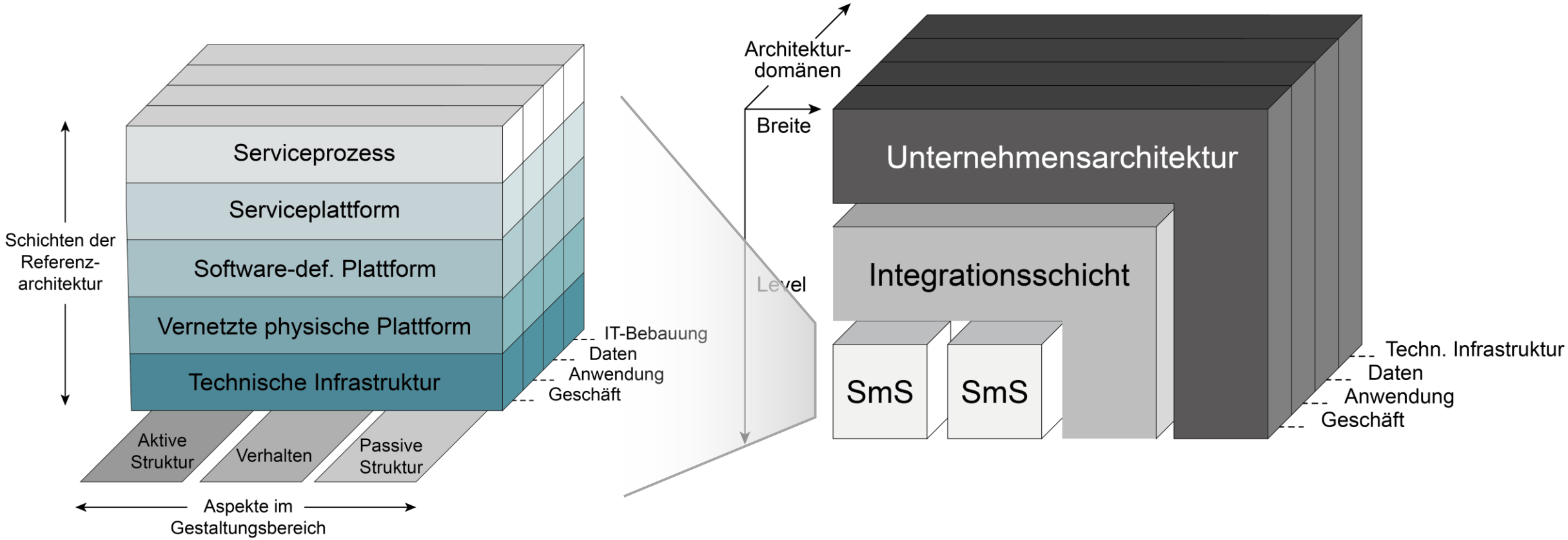
Definiert die Kern-Funktionalitäten für das Entwicklungsvorhaben.

Dient als Grundlage für das Datenmodell.



Smart Service Entwicklung im Unternehmen verorten

Gestaltungsbereich für eine Spezifikationstechnik



Smart Service Spezifikation
in den dafür relevant zu betrachtenden Ebenen

Smart Services im EAM-Kontext
in Anlehnung an TOGAF

Spezifikationstechnik

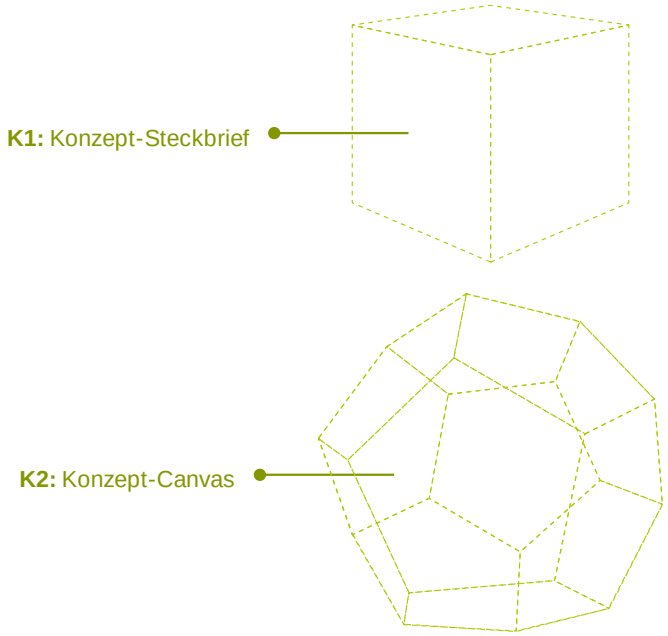
Relevante Sichten auf die Lösung in der Smart Service Spezifikation

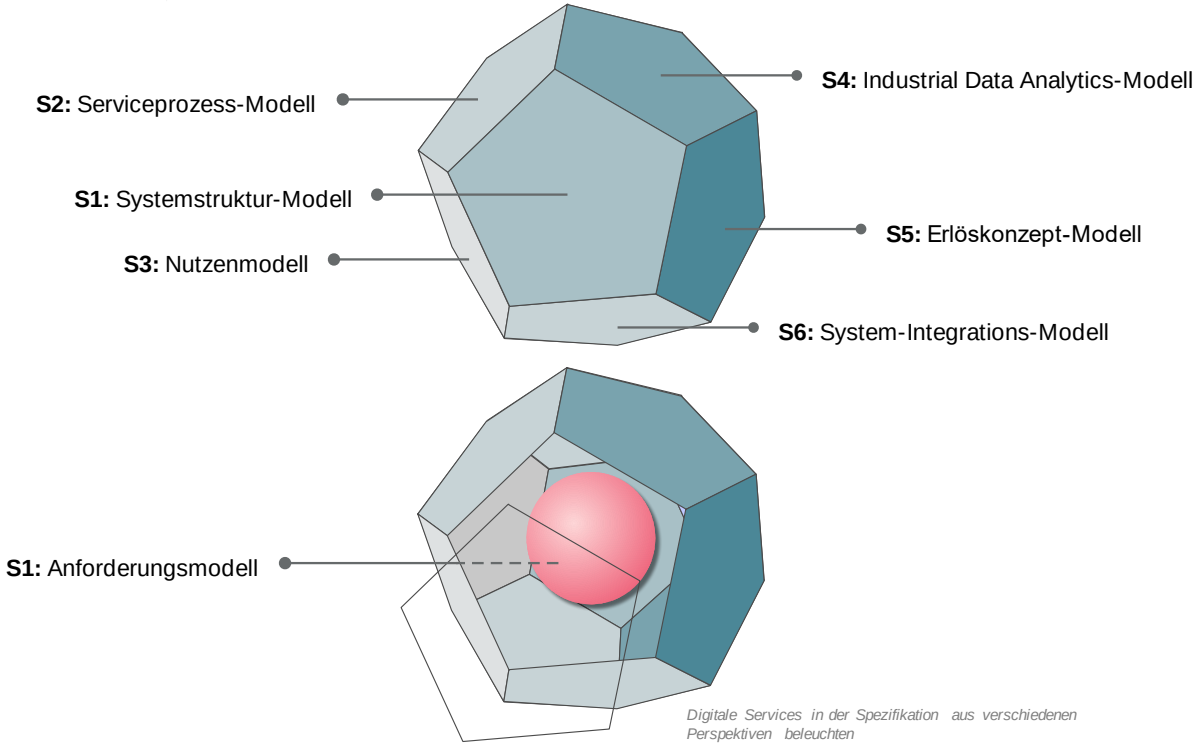
von der…

Konzeptphase

hin zur…

Spezifikation





Digitale Services in der Spezifikation aus verschiedenen Perspektiven beleuchten

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

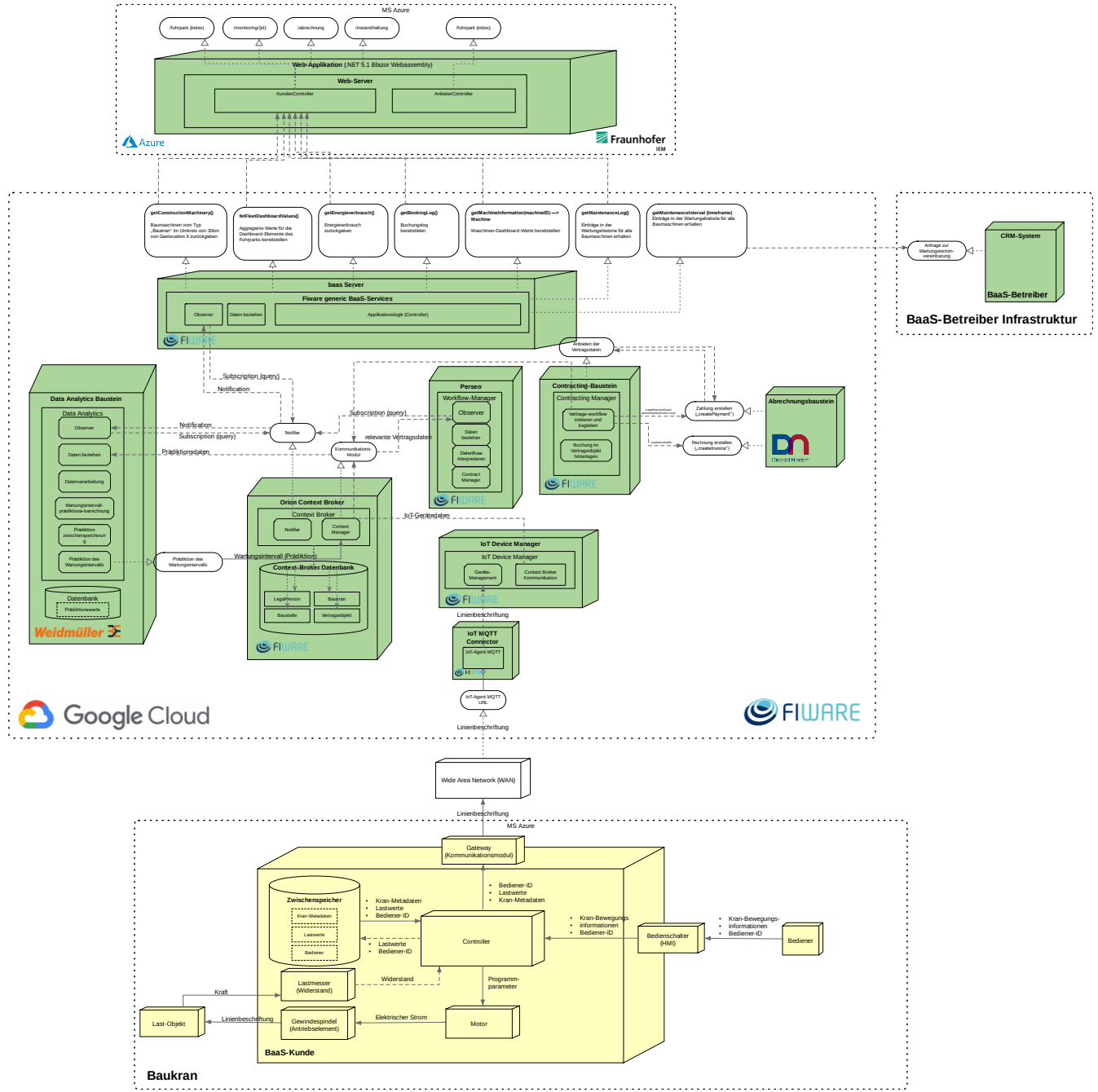
Fraunhofer IEM

Aktuelle Folie zu Elementen der Spezifikationstechnik eintragen

Beispiel für die generierten Sichten auf den Use Case in der Spezifikation am Beispiel techn. Demonstrator ergänzen

Mit Texten der Folgefolie verheiraten

Prozess der Kollaborativen Entwicklung andeuten



Spezifikation

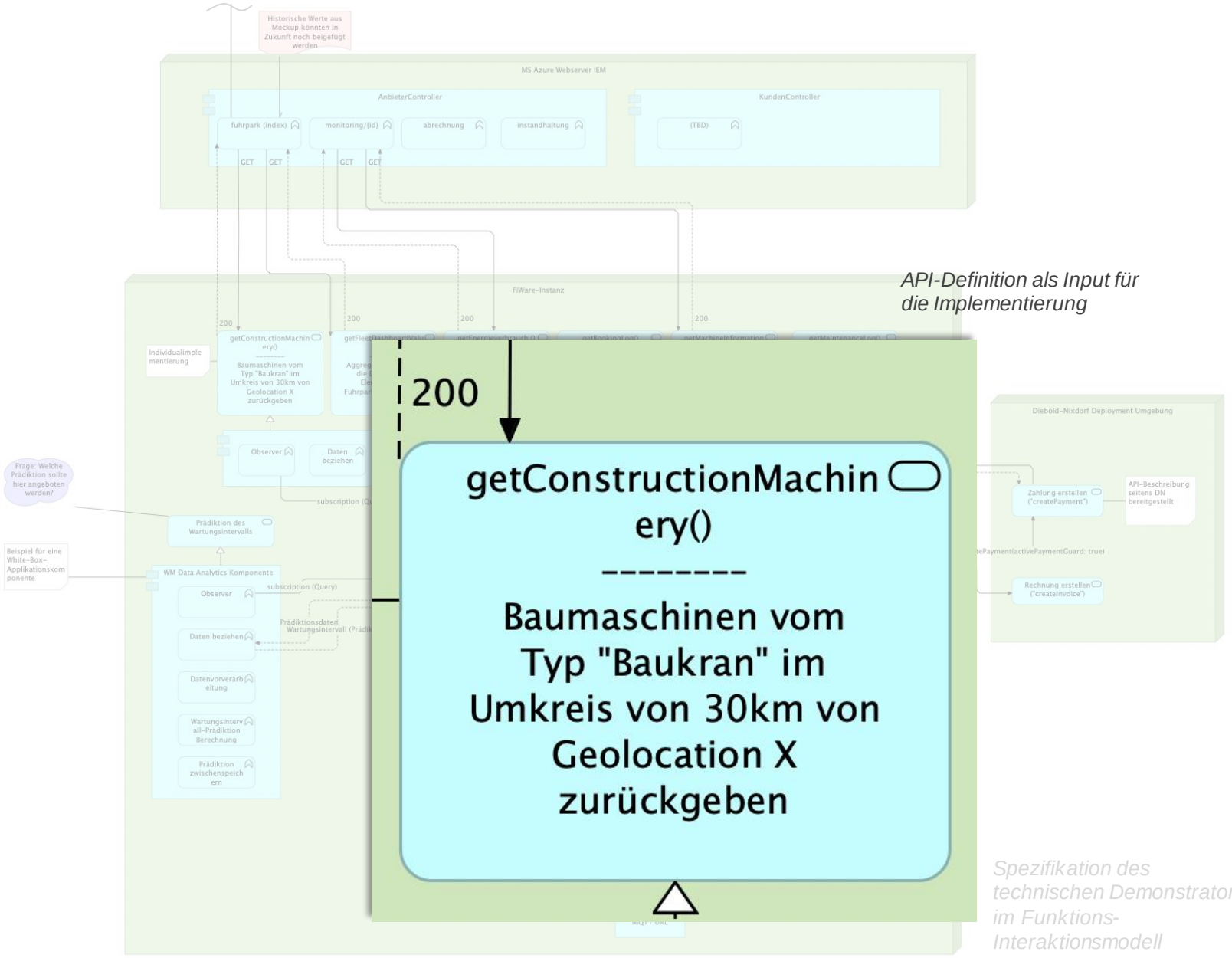
Detaillierung im Systemmodell

SPEZIFIKATION

- (vorig gezeigter) **Mockup** wird in der Spezifikation durch Lösungsansätze und -konzepte konkretisiert.
- **Weitere Partialmodelle** (neben dem hier rechts gezeigten) ergänzen das Systemmodell und konkretisieren die Lösung aus verschiedenen (relevanten) Perspektiven.

ENTWICKLUNG

- Die nun spezifizierten Inhalte dienen dann als Gegenstand der zu entwickelnden **(Software-) Artefakte**.
- „Grüne Bereiche“ zeigen die **Zuständigkeiten** für die Entwicklung der technischen Komponenten.
- Inhalte werden dementsprechend detailliert dokumentiert, dass Sie reibungsfrei in den **OpenAPI-Standard** überführt werden können (siehe Folgefolie).



API-Definition als Input für die Implementierung

Spezifikation des technischen Demonstrators im Funktions-Interaktionsmodell

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Fraunhofer IEM

AGENDA

FÜR DIE ENTWICKLUNG SMARTER SERVICES IN PRODUZIERENDEN UNTERNEHMEN

1. Konzept / Ideenfindung
2. Spezifikation
3. (Rahmengebende Faktoren)
4. Implementierung

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Fraunhofer
IEM

AGENDA

FÜR DIE ENTWICKLUNG SMARTER SERVICES IN PRODUZIERENDEN UNTERNEHMEN

1. Konzept / Ideenfindung
2. Spezifikation
3. (Rahmengebende Faktoren)
4. Implementierung

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Fraunhofer
IEM

FIWARE

Platfform basierte Implementierung

- Ziele der Implementierung:
  - Geringe Implementierungs und Wartungsaufwände
  - "Time-to-market"
  - Erweiterbarkeit

Verwendung von Open Source-Komponenten, standardisierte Schnittstellen

Lösung: FIWARE als Basis der Plattform mit Domain-spezifischen Erweiterungen

FIWARE

"Kuratiertes Framework an Open Source Plattform Komponenten."

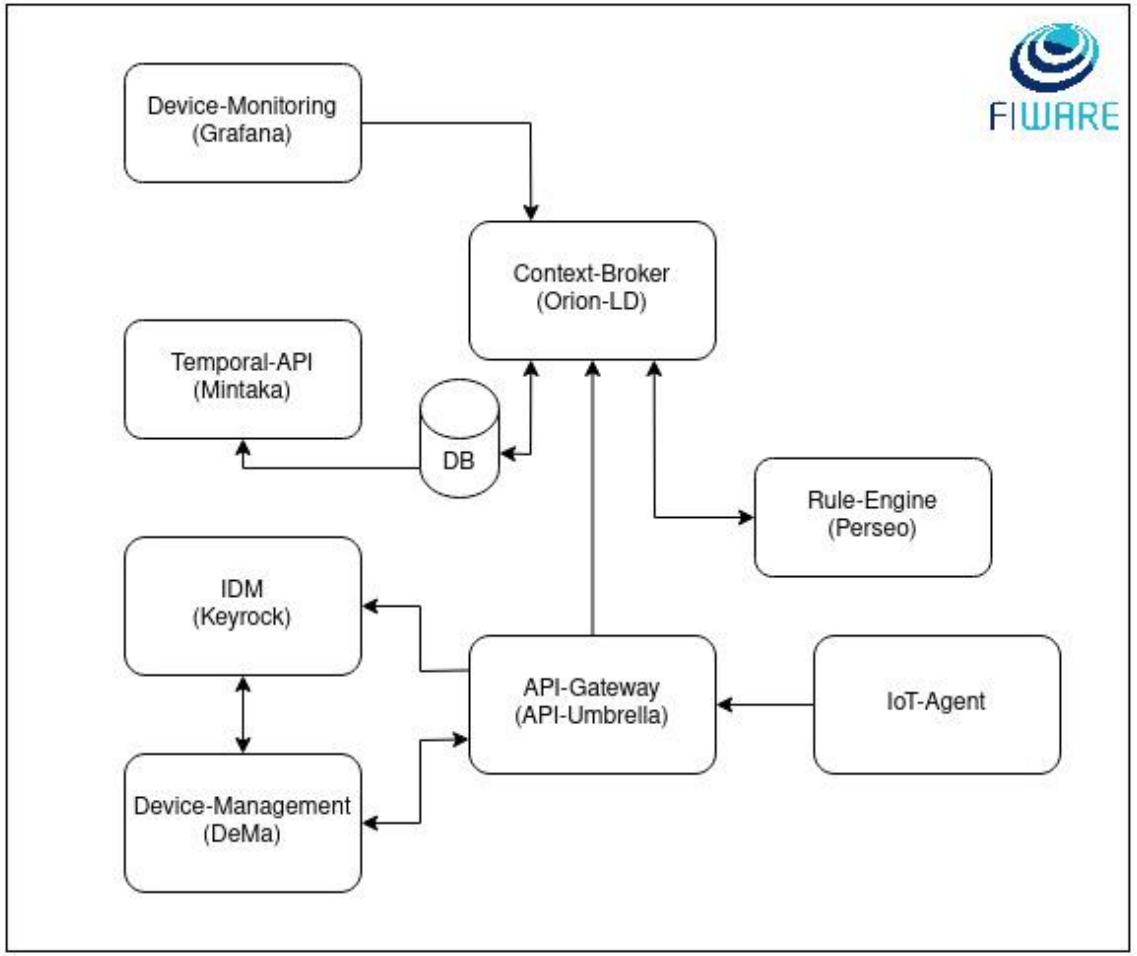
- Zentrale Schnittstelle NGSI-LD - API für generisches Kontext-Management
- Context Broker als zentrale Komponente, ergänzt durch Use-Case spezifische Lösungen für bspw.:
  - Identity&Access Management - Keyrock/API-Umbrella
  - IoT-Connectivity - DeMa/IoTAgents
  - Rule Engine – Perseo

NGSI-LD API: https://docbox.etsi.org/isg/cim/open/Latest%20release%20NGSI-LD%20API%20for%20public%20comment.pdf

FIWARE Catalogue: https://github.com/FIWARE/catalogue

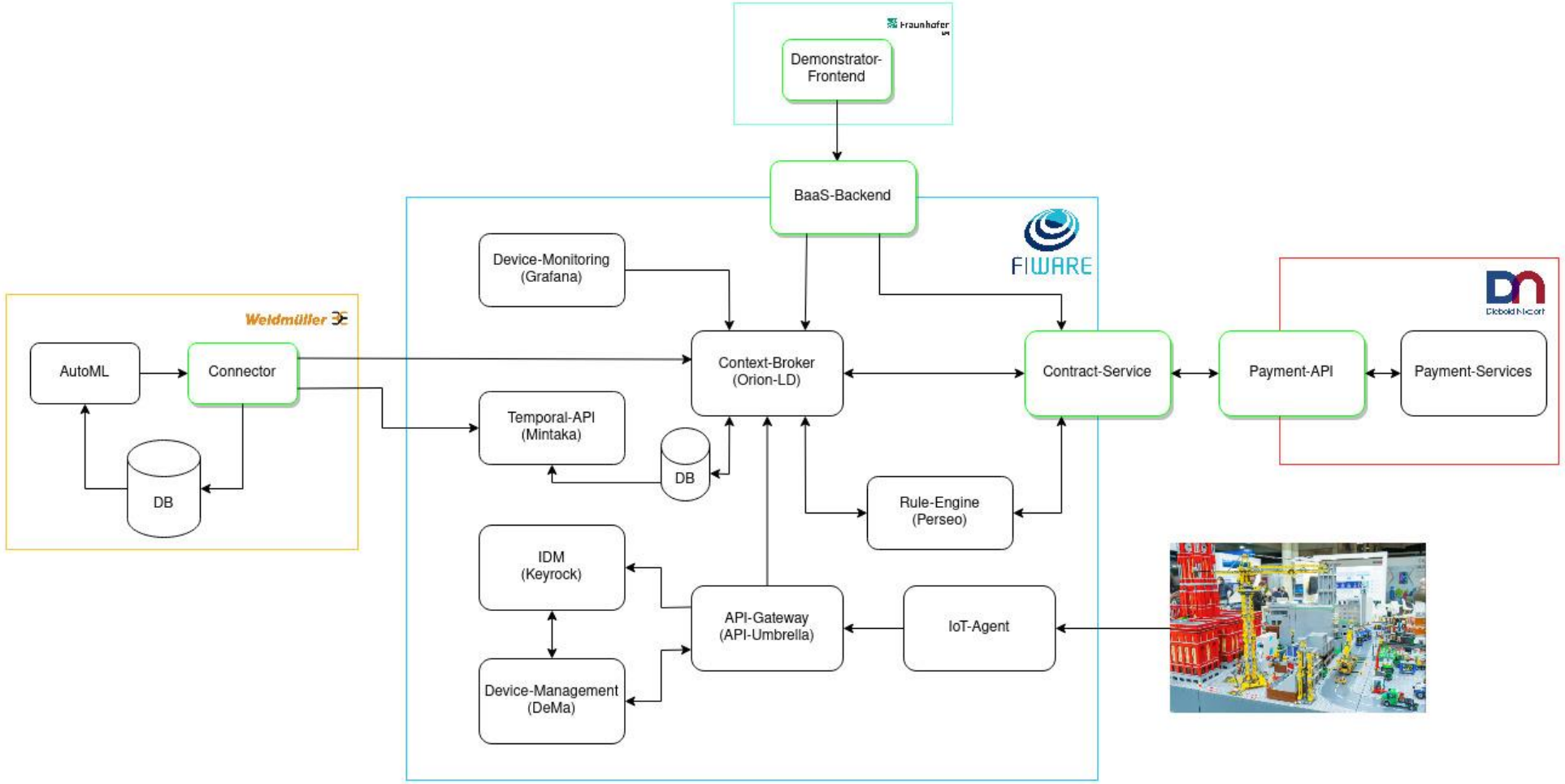
Ansatz zur Implementierung

Basis Plattform



Ansatz zur Implementierung

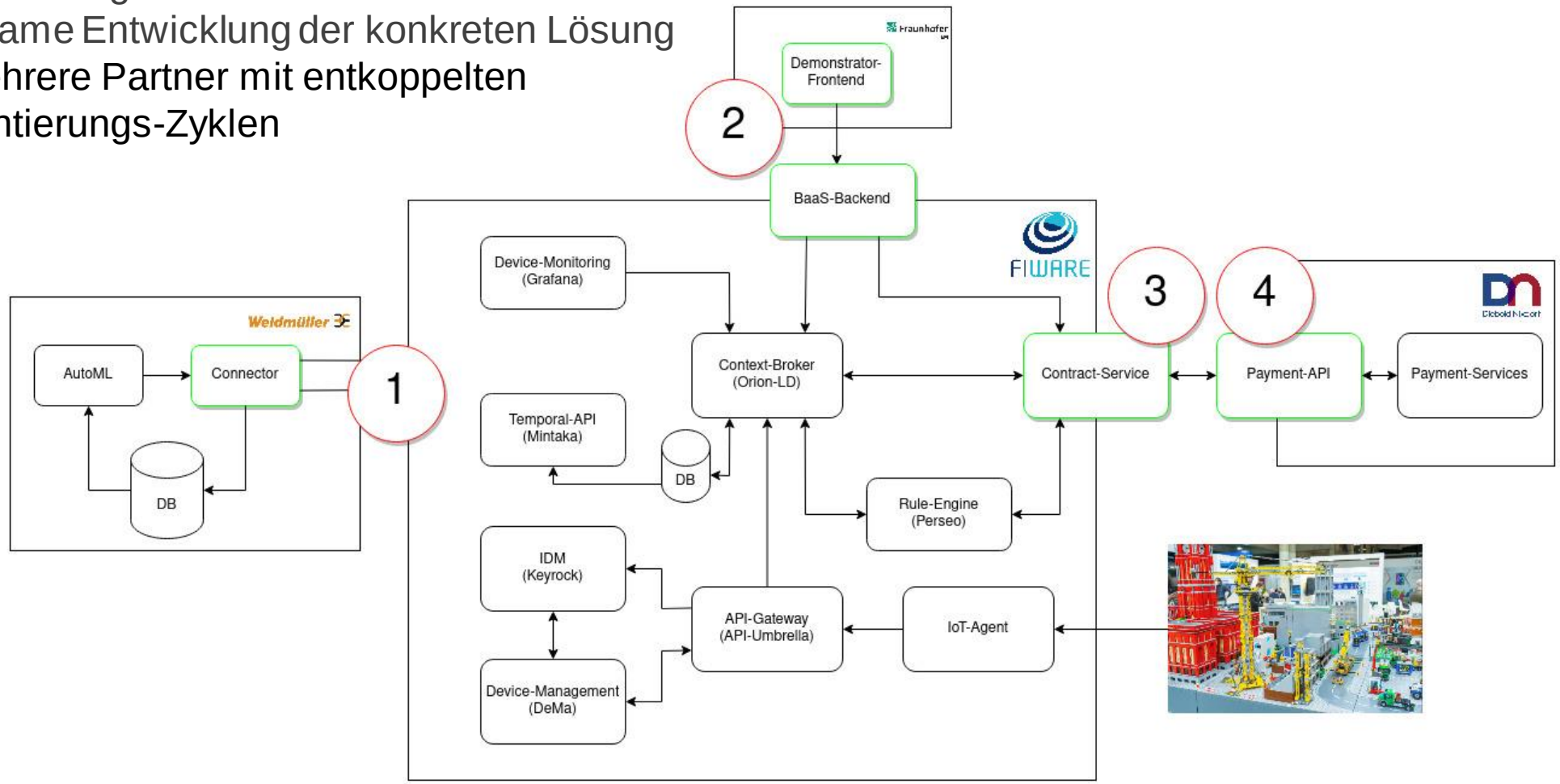
Impress



Ansatz zur Implementierung

Schnittstellen

Herausforderung:
Gemeinsame Entwicklung der konkreten Lösung durch mehrere Partner mit entkoppelten Implementierungs-Zyklen



Implementierung

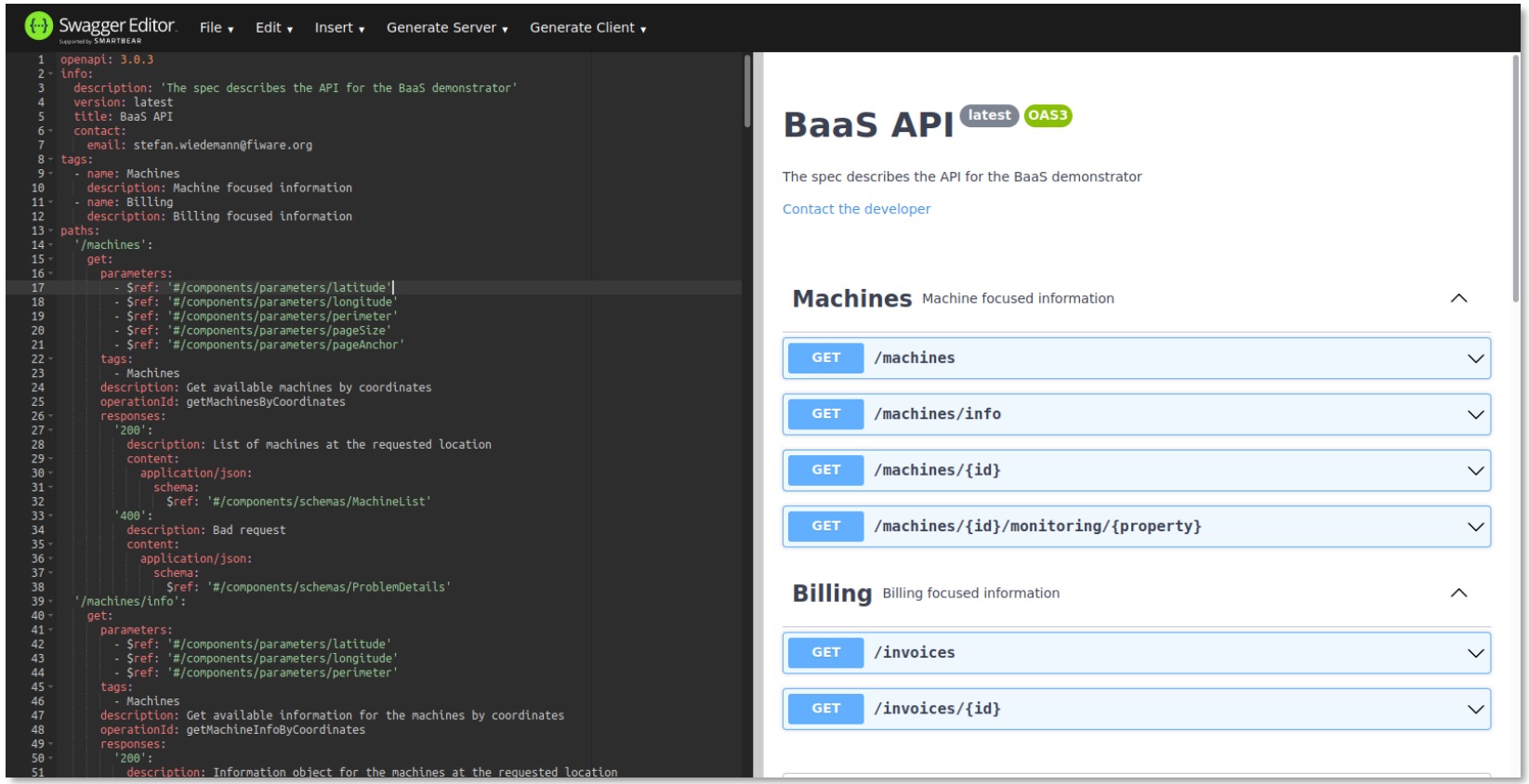
OpenAPI-Spezifikation als Intermediär zwischen Spezifikation und Implementierung

- Definition der API als **"Vertrag"** zwischen den Implementierungs-Partnern.
- Fokus der API auf jeweilige Domäne(z.b. Zahlungsabwicklung, Frontend-Darstellung, Maschine-Learning)
- Kollaborative Weiterentwicklung über **Github** – Pull Requests.
- **Open-API Spec** als SPoT, kann als Basis für Code-generierung verwendet werden (Client/Server/Tests) .

Implementierung

OpenAPI-Spezifikation am Beispiel BaaS - Frontend

- Fokus der API auf **Frontend-Flow**.
- REST-Endpunkte mit Domänen-spezifischen Resourcen(z.b. Machine, Invoice)
- Basis zur Generierung der Server-Interfaces
- Verwendung als Server-Mock zur Frontend-Entwicklung



Visualisierung der "BAAS"-.yaml-Datei in der OpenAPI-Spec mit Swagger

Implementierung

Integration in die Entwicklung

- Server Mock zur Frontend-Implementierung:

  Docker zur Bereitstellung, bspw.:

  https://hub.docker.com/r/mockserver/mockserver

```
[
  {
    "httpRequest": {"specUrlOrPayload": "file:/api/baas.yaml", "operationId": "getMachinesById"},
    "httpResponse": {"statusCode": 200, "headers": {"content-type": ["application/json"]},
      "body": {
        "id": "urn:ngsi-ld:crane:my-crane",
        "type": "Baukran",
        "model": "Euro SSG 130"
      }
    }
  }
]
```

- OpenAPI Generator zur Generierung der REST-Interfaces:

```
@io.micronaut.http.annotation.Get("/machines/info")
@io.micronaut.http.annotation.Status(io.micronaut.http.HttpStatus.OK)
@io.micronaut.http.annotation.Produces({ "application/json" })
MachineInfoVO getMachineInfoByCoordinates(
        @io.micronaut.http.annotation.QueryValue(value = "lat")
        java.lang.Double lat,
        @io.micronaut.http.annotation.QueryValue(value = "longi")
        java.lang.Double longi,
        @io.micronaut.http.annotation.QueryValue(value = "perimeter")
        java.lang.Double perimeter);
```

Danke.