

Plattformen und Ökosysteme als Chance zur Befähigung

- Lösungen, Neue Geschäftsmodelle, Kooperationen, Business Transformation -

Markus Warg
Institut für Service Design, Hamburg



www.IfSD.hamburg

Von der Produkt- zur Service-dominanz

“Die Top-Serviceerlebnisse von heute prägen die Erwartung Ihrer Kunden”



Produkt-dominiert
DVD Verkäufer



Service-dominiert
Plattform Company



Service-dominiert
Content Provider



160 Mio. subscriber

\$ 6 Mrd.

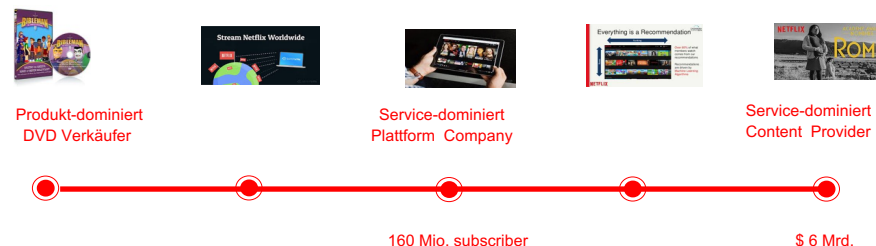
Von der Produkt- zur Service-dominanz

“Die Top-Serviceerlebnisse von heute prägen die Erwartung Ihrer Kunden”

Service

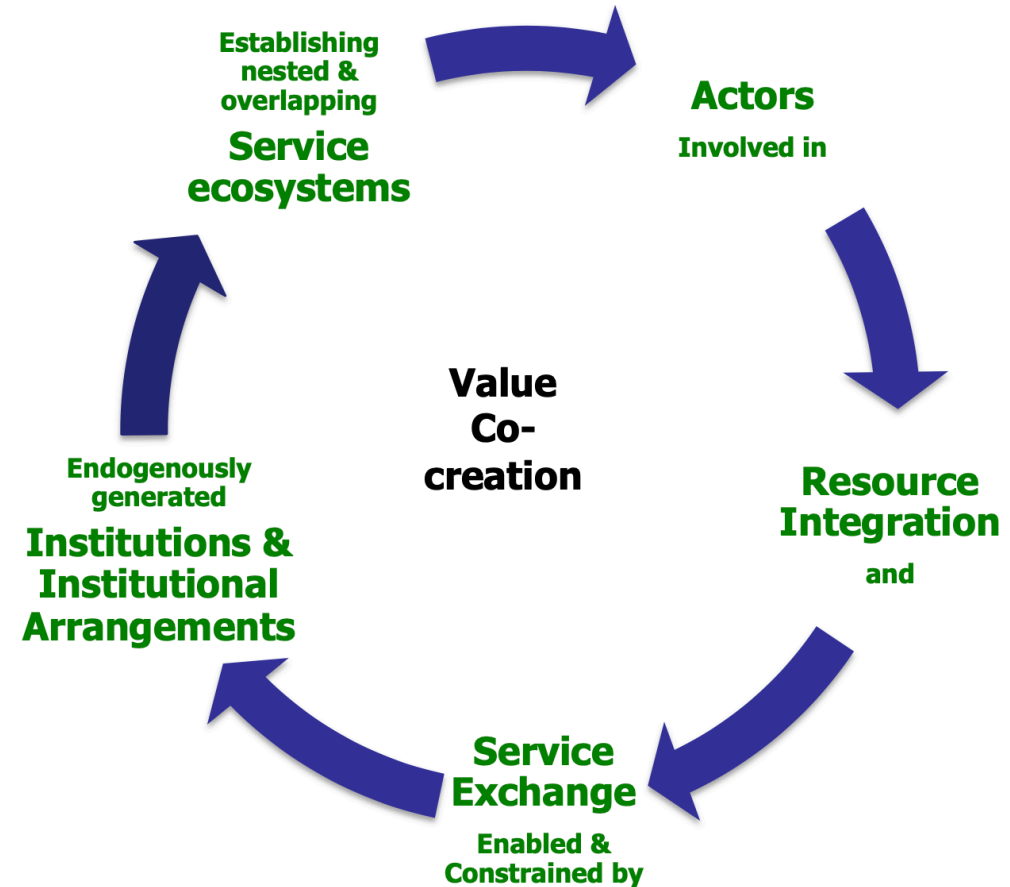
- Service als Gebrauchsnutzen (value in use), der bei der Anwendung eines Wertangebotes (Produkt, Lösung) entsteht. (Vargo, Stephen L., & Lusch, Robert F. (2004). Evolving to a New Dominant Logic for Marketing. *Journal of Marketing*, 68(January), 1-17.)
- Service als Prozess bei dem Ressourcen angewendet werden und Nutzen für einen anderen Akteur entsteht. (Spohrer, J, Maglio, PP, Bailey, J, & Gruhl, D. (2007). Steps toward a science of service systems. *Computer*, 40(1), 71-77.)
- daher ist der Nutzen immer interaktiv und von mehr als einem Akteur (Mensch, Technologie) "co-created" (*FP 6)
- Nutzen ist immer einzigartig und hängt von den Präferenzen und dem Kontext (value in context) des Akteurs ab (*FP 10)

*FP = Foundational Premises der Service-Dominant Logic



„actors fundamentally do the same thing: they integrate resources and engage in service exchange all in the process of cocreating value“

Vargo, S. L., and Lusch, R. F. 2016. "Institutions and Axioms: An Extension and Update of Service-Dominant Logic," *Journal of the Academy of Marketing Science* (44:1), pp. 5-23.



Von der Produkt- zur Service-dominanz



Produkt-dominiert

Tauschwert

Wert

Tauschgegenstand

Kunde



Service-dominiert

Gebrauchsnutzen

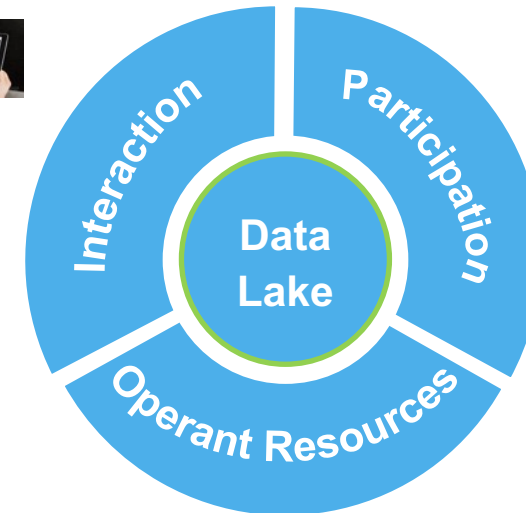
Fähigkeiten

Interaktiv

Interaktion
Gebrauchsnutzen



Externe Fähigkeiten



Vorhandene Fähigkeiten

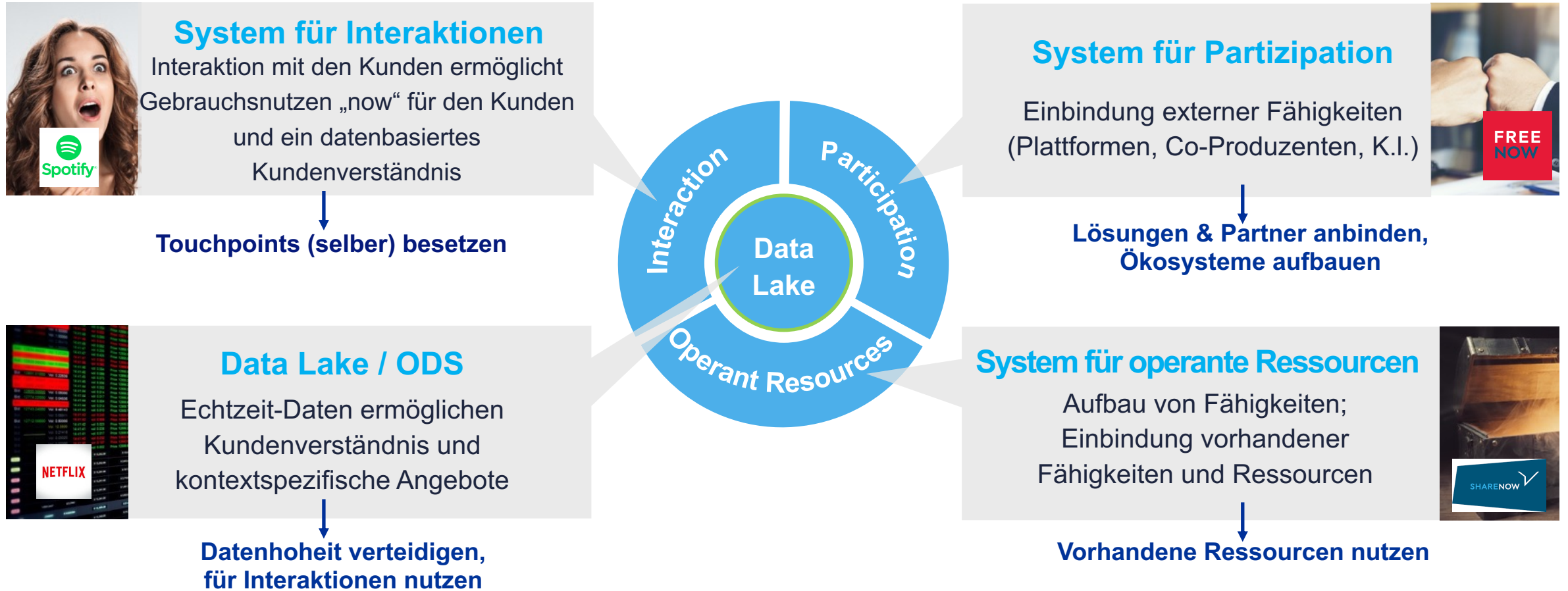
Service Dominierte Architektur



Service-dominierte Logik

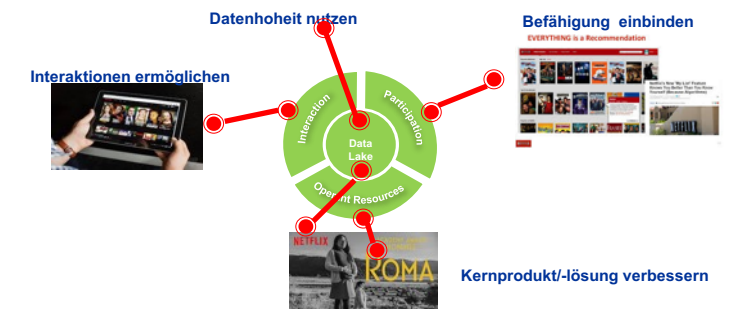
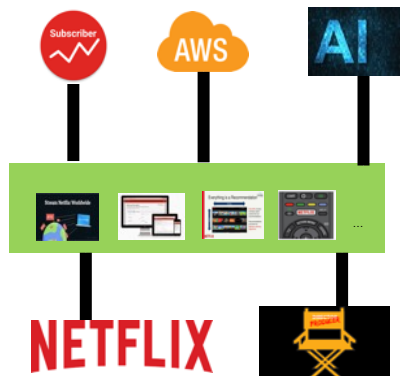


Service Dominierte Architektur



Plattformen und Ökosysteme als Chance zur Befähigung

1. **Service Platform:** Netzwerk, Ressourcenintegration, Ressourcendichte, Interaktion
2. **Service Ecosystem:** actor-to-actor orientation, setzt institutionelle Regeln
3. **Service:** Value Cocreation; Service als Anwendung von Ressourcen zum Vorteil eines Anderen



Vorgehen zur Entwicklung von Plattformen und Ökosystemen

SDA Human
Centered Service
und Lösungen



Service als Prozess
Human Centered Value

SDA fachliche 'Bausteine'



Fachliche Bricks & Service Stacks
– like insurance layer

SDA Basis ('Platten')



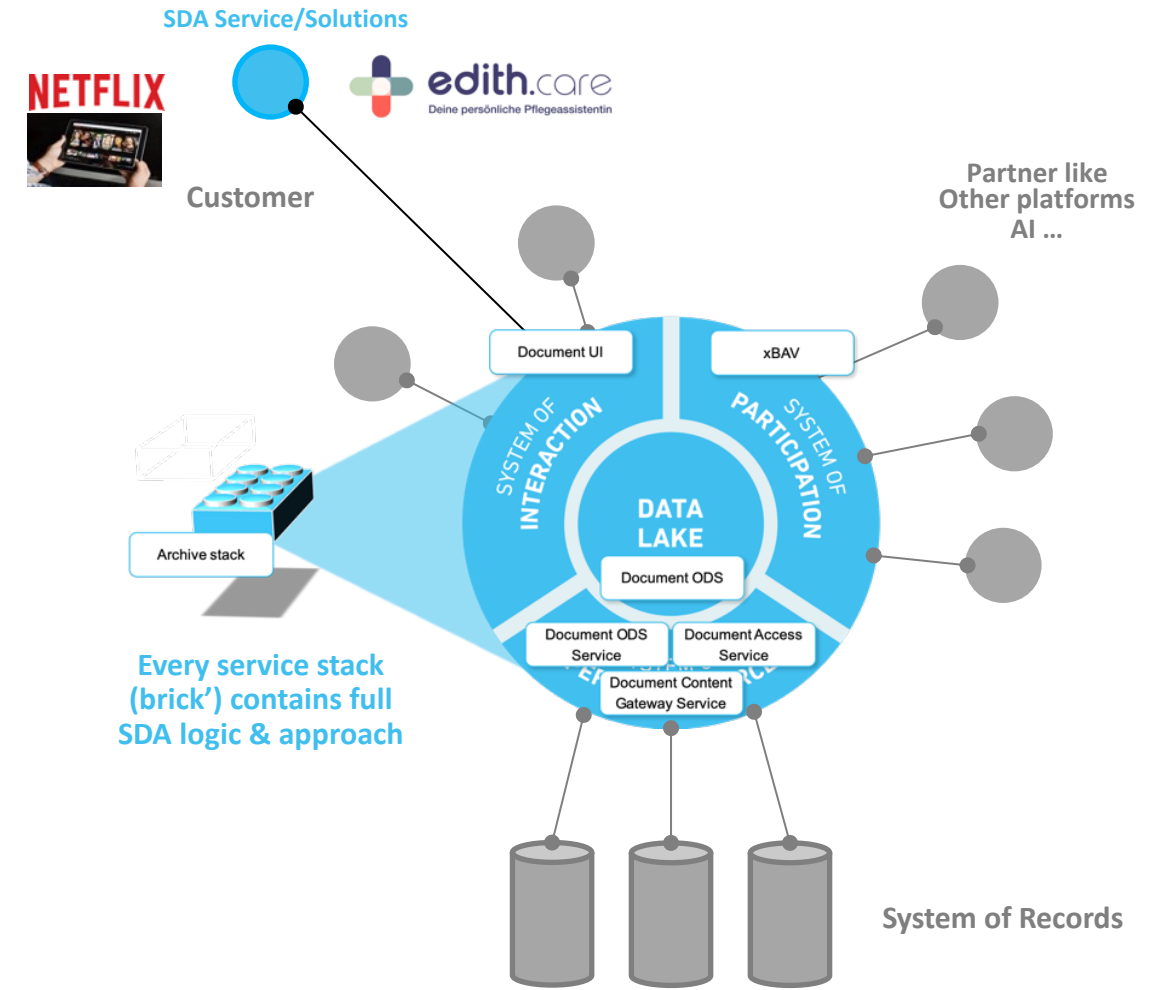
Technische Basis (micro) Services
= Output

SDA
Transformation
Enablement



Strategie
Proof of Concept

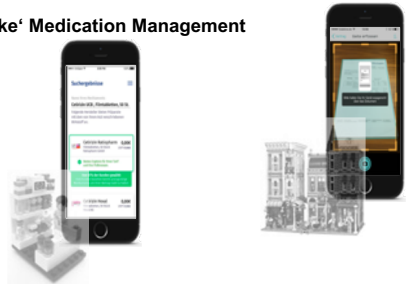
www.IfSD.hamburg



Business Transformation: Lösungen, Neue Geschäftsmodelle, Kooperationen, Transformation

Lösungen

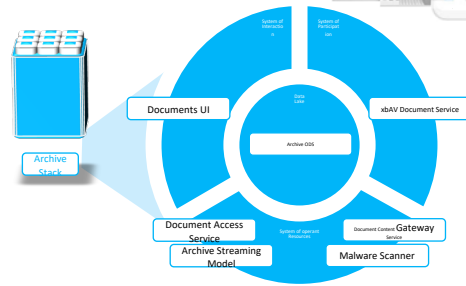
'Apotheke' Medication Management



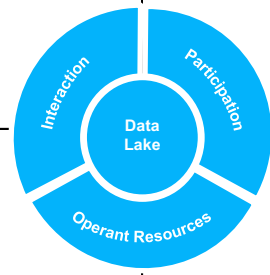
'Shopping Mall' Einreicher App



'Klinik' digitaler Arztbesuch



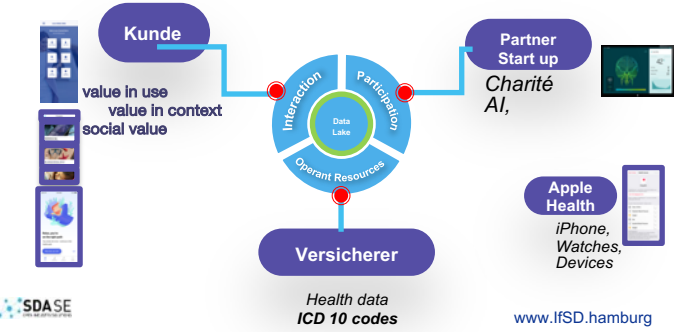
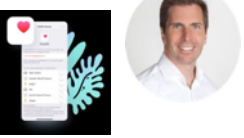
Geschäftsmodelle



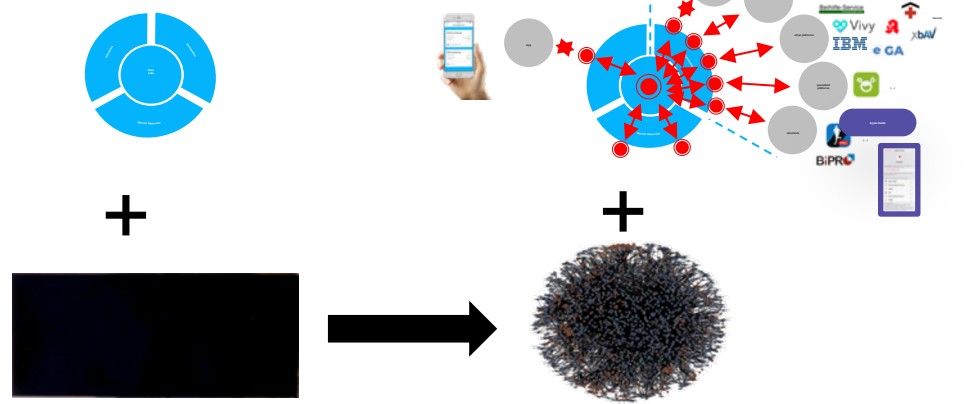
AI4medicine



Digital Health Award 2018 | Projekt Zukunft projektzukunft.berlin.de



Kooperationen



Transformation

Logik der Lösungen: 'Ritterburg' Vertragsanzeige



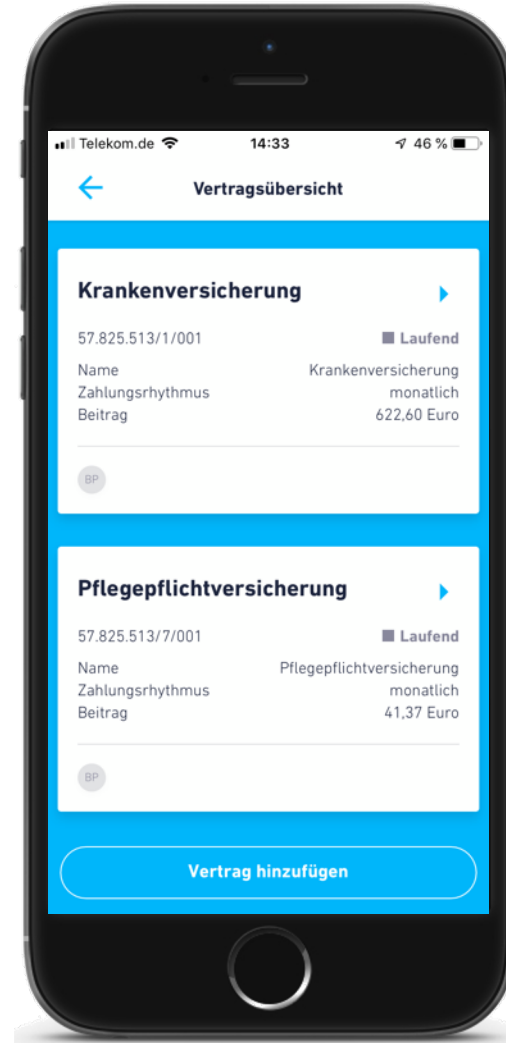
Partner Stack

Contract Stack

Basis Plate

App Plate

www.IfSD.hamburg



Herausforderung

1. Endkunde: Vertragsunterlagen nicht zur Hand oder zu suchen im Papierordner
2. Versicherung: Reduktion von Service-Calls und Versendung von Unterlagen

Lösung

Übersichtliche Darstellung der Verträge in der App

Status

läuft

Logik der Lösungen: weitere Beispiele

‘Apotheke‘ Medication Management

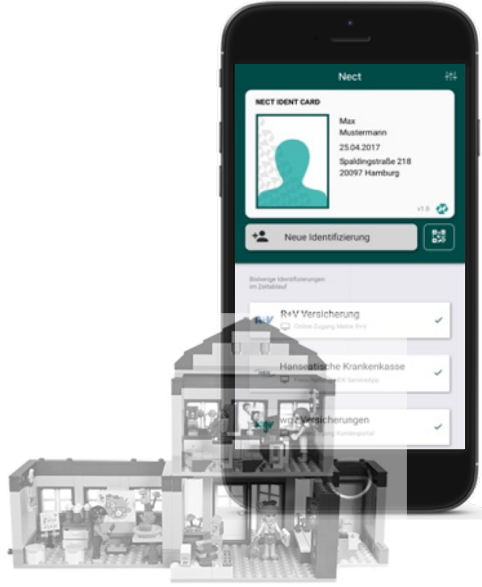


www.IfSD.hamburg

‘Shopping Mall‘ Einreicher App

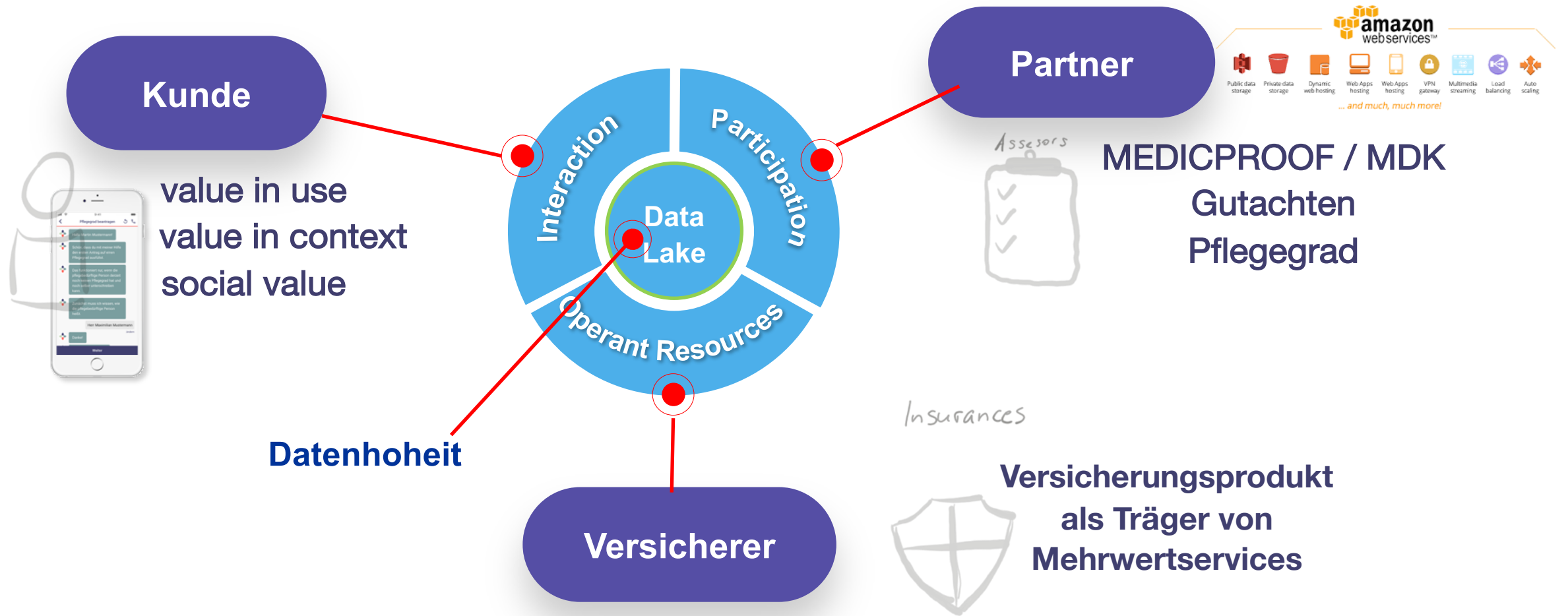


‘Schule‘ Selfie-Ident (‘Nect‘)

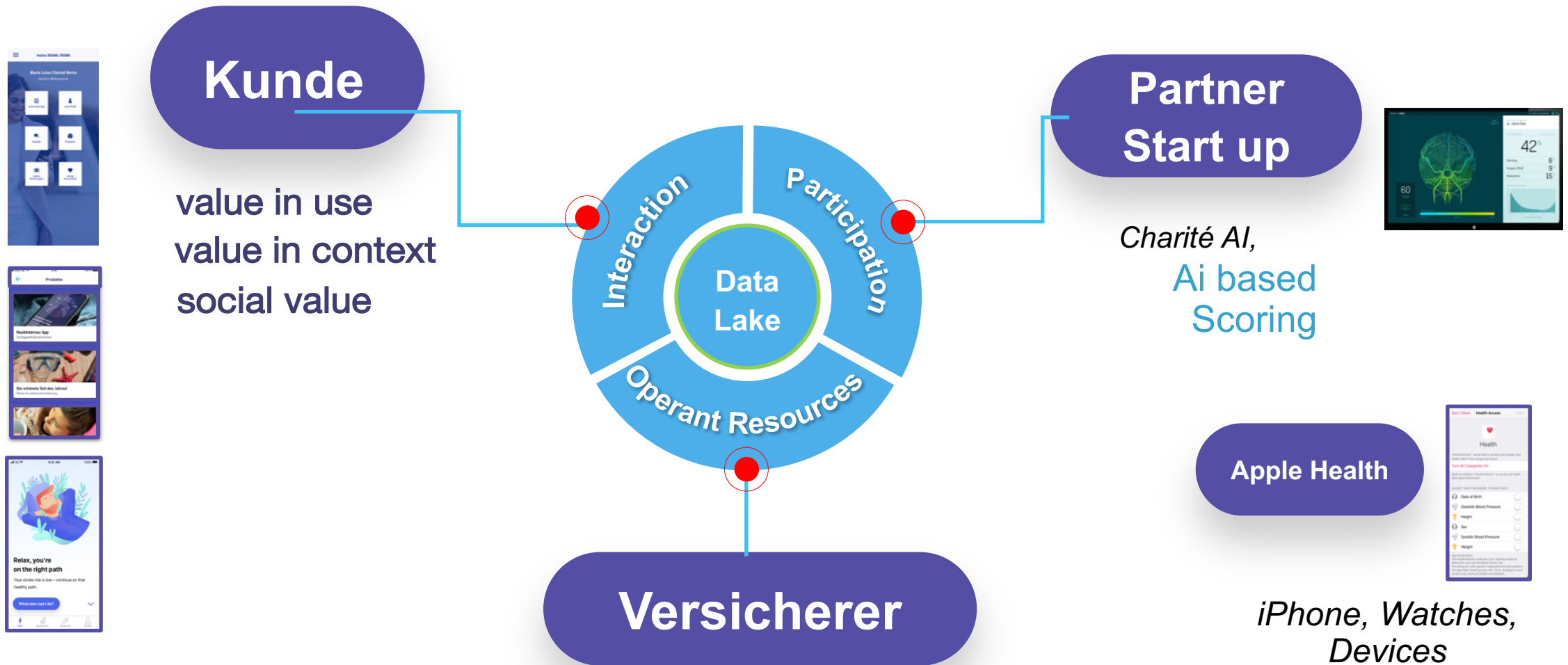


■ ■ ■

Neue Geschäftsmodelle: edith.care Pflegeantrag in 10 Minuten statt 10 Tagen

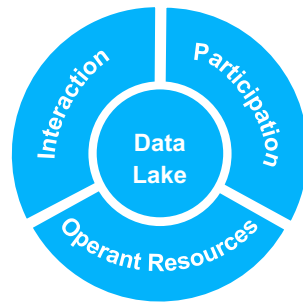


Kooperationen zwischen Start Up und tradierten Versicherern: Private Health Advisor – Schlaganfall Prävention

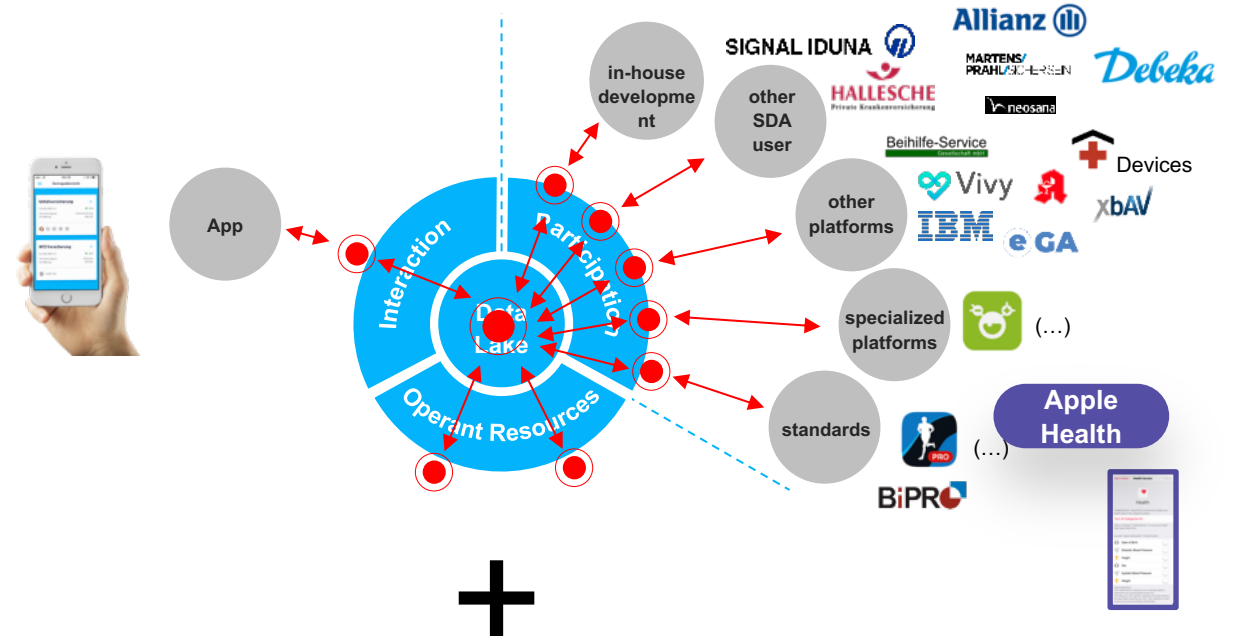


Business Transformation

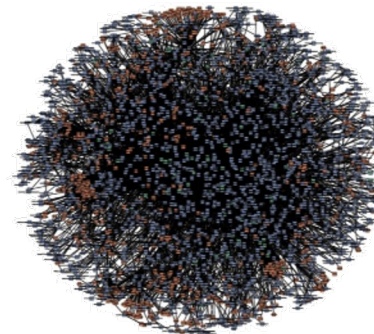
Der Weg zur Plattform Organisation



+



+





Open/Open Source



Co-laboration/Co-Creation



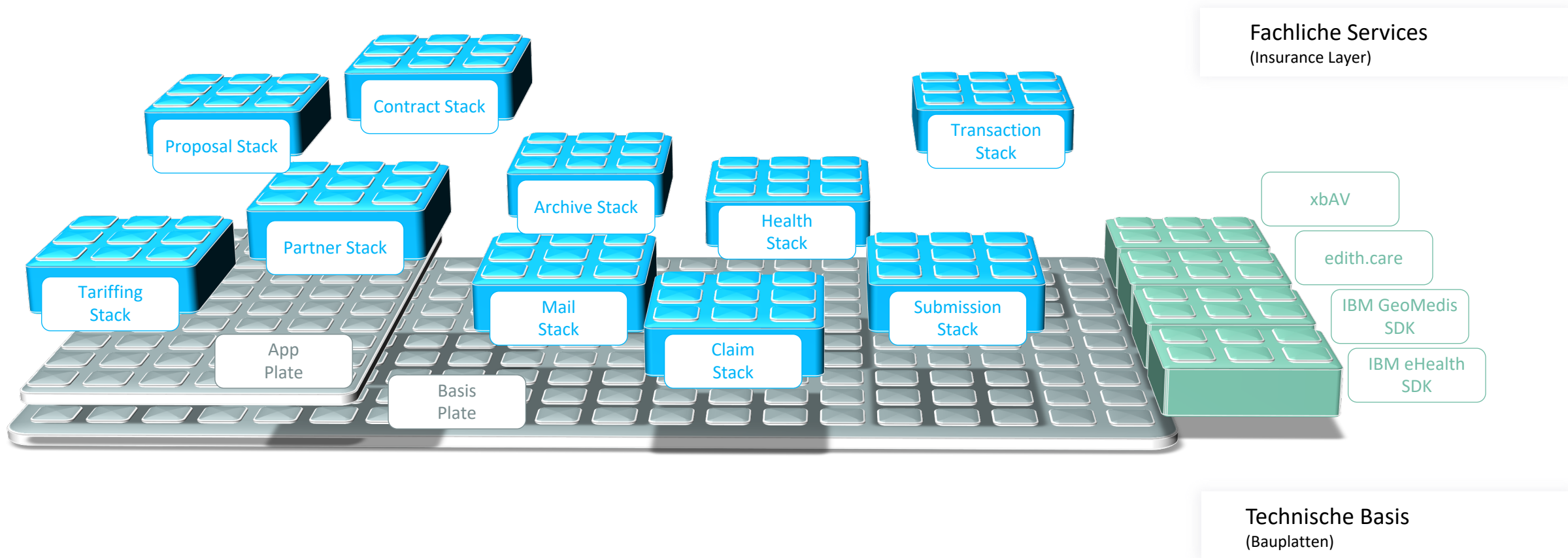
Modular/MicroServices



www.IfSD.hamburg  Trends
Strategien
Lösungen
Institut für Service Design®

SDA liefert den ‚Baukasten‘ zur Erreichung ihrer Ziele: (a) eine (technische) Bauplatte sowie (b) versicherungsfachliche Bausteine

3 MODULARISIERUNG → PRODUKTE



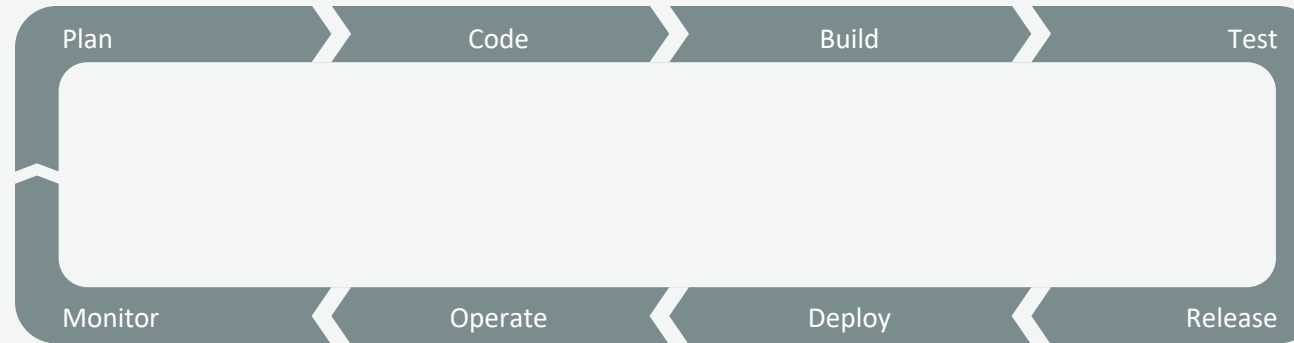
- SDA als Microservice-basiertes Architekturframework
- (A-)Synchrone Kommunikation (*REST / Kafka*)
- Identity Provider (*Keycloak*)
- Autorisierungskonzept (*Open Policy Agent*)

- Logging-Stack (*ELK, EFK*)
- Monitoring-Stack (*Prometheus, Grafana, inkl. Dashboards*)
- Alerting (*Grafana, Slack*)
- Tracing (*OpenTracing & Jaeger*)

- Microservice Framework (*sda-commons*)
- Bereitstellung Developer und Operations Guidelines, Tooldiskussionen und Onboardings

- Auf Plattform abgestimmte Build-Pipeline (*Jenkins*) / *Container-Erst. (Docker)*
- Div. automatische Security und Compliance Checks in Pipeline

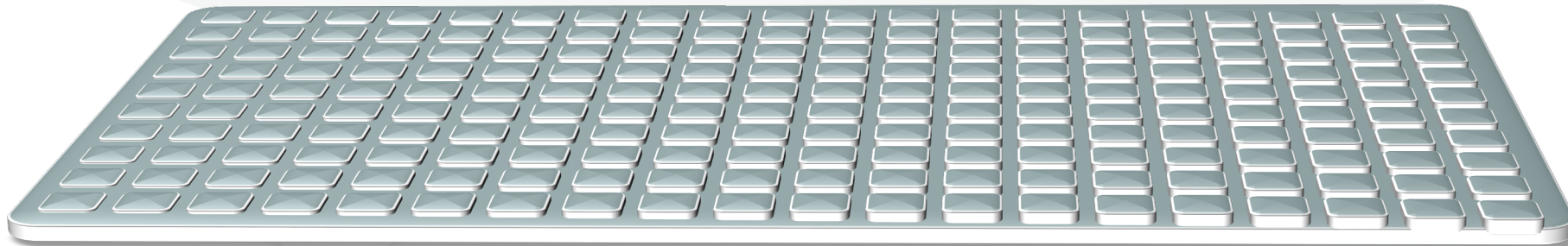
- Behavior Driven Development über Akzeptanzkriterien (*Gherkin, Cucumber*)
- Ende-zu-Ende Tests (*Selenium*)
- Testausführung in Build-Pipeline



- Bereitstellung von Docker-Containern in Repositories
- Source-Code-Verwaltung (*GitHub*)
- Container-Registry (*quay.io*)
- Semantic Versioning

- Cloud-agnostischer Betrieb
Infrastrukturabstraktion und Orchestration (*Kubernetes*)
- Service Mesh (*Istio*)
- Dezentrales Metering & Billing

- Deployment und Infrastruktur als Code
- Automatisierte Infrastruktur-Konfiguration (*Terraform*)
- GitOps (*ArgoCD*)

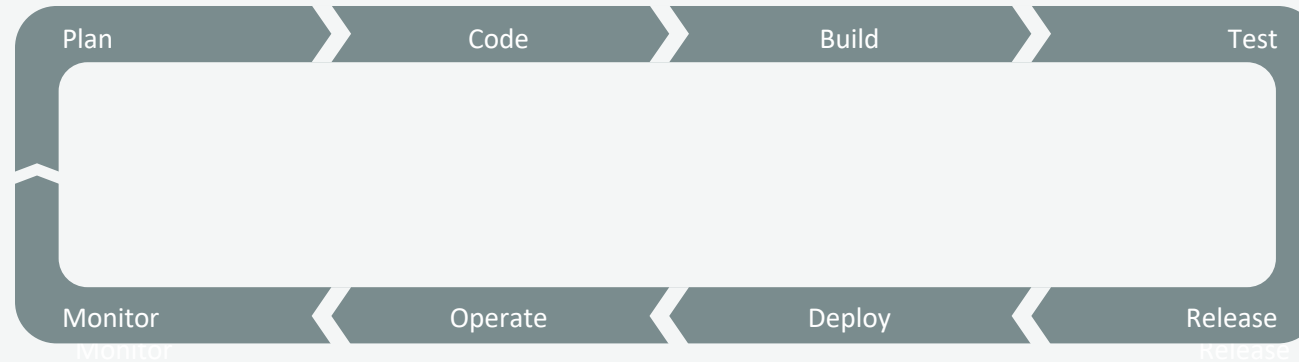


- Modulares Micro-Frontend, in das einfach Plugins implementiert werden können
- Monorepository Architektur (*NX workspaces*)
- Hybride Entwicklungsframeworks (*Angular, Ionic*)
- Nutzungsanalysen (*Google Analytics, Tealium*)
- Crash Monitoring (*Crashlytics*)

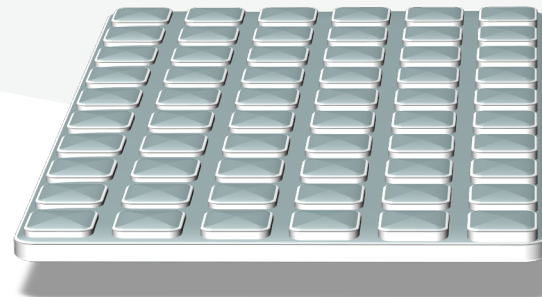
- SDA AppCore Layer für übergreifende Funktionen und Verhalten (z.B. Authentifizierung, Logging, Zugriffe, UI-Elemente)
- Code Generatoren / Schemata für Plugins

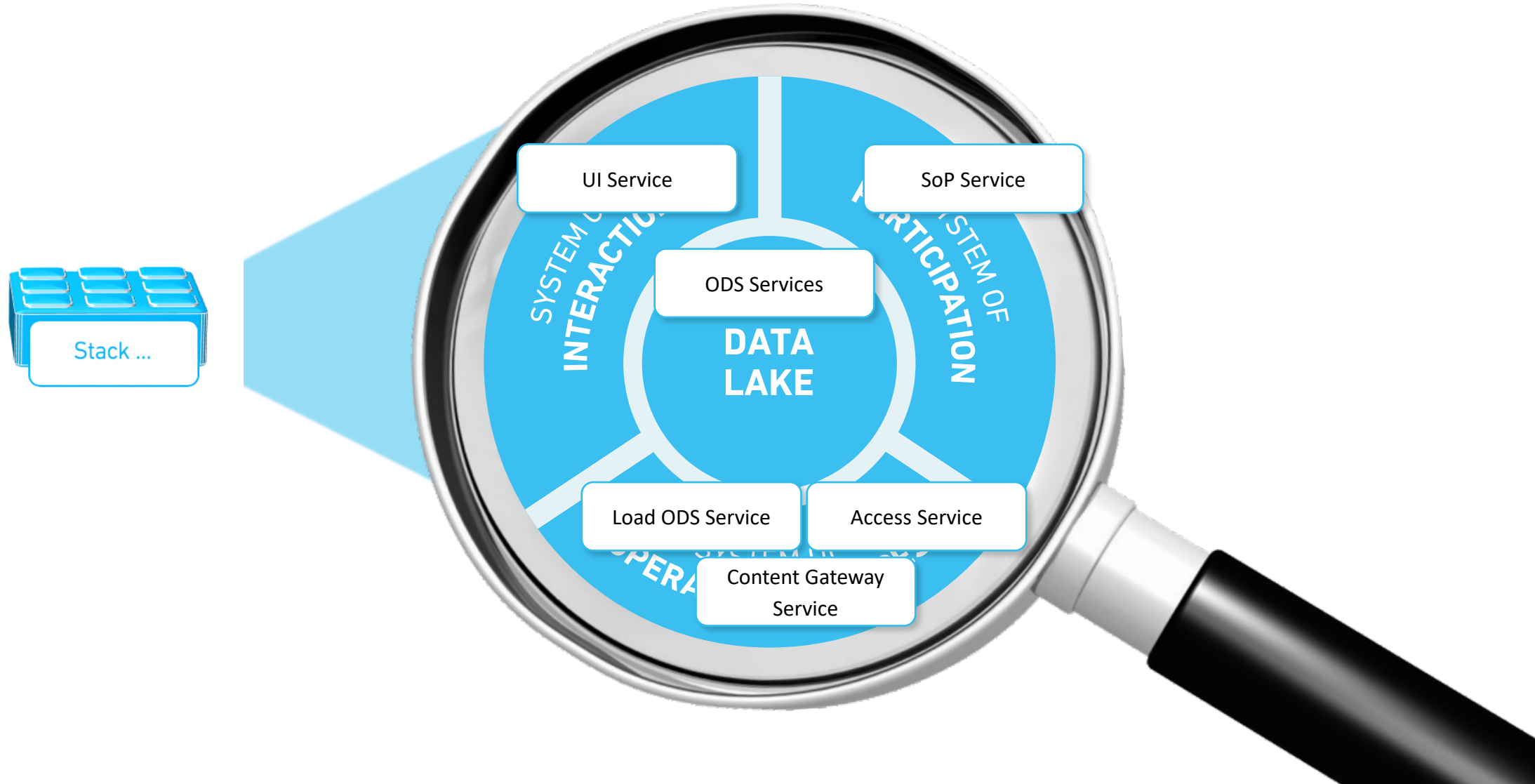
- Lokale Builds für Web, iOS und Android
- Build Scripts (*NG und NX builders für angular und Angular Workspaces*)

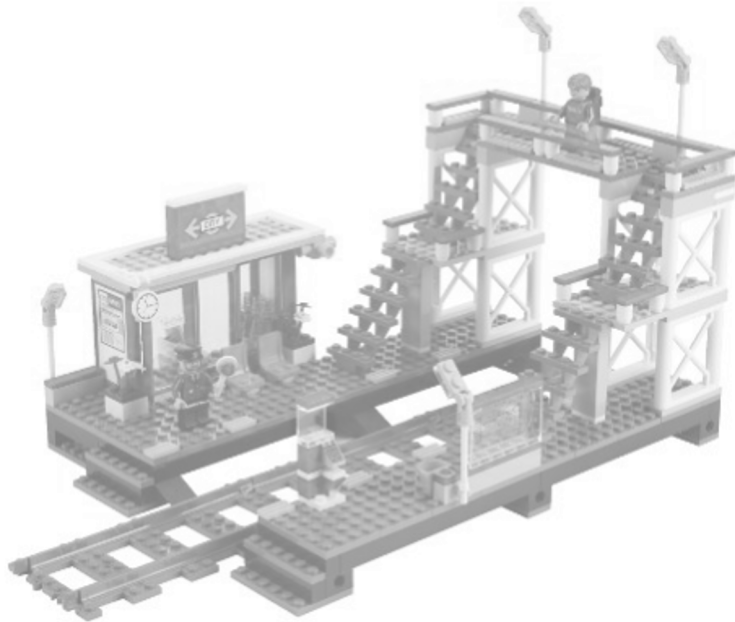
- Parallele Unittests (*Jest*)
- Native App Builds für manuelles Testing (*Bitrise*)
- Automatisierte Ende-zu-Ende Tests in der Build-Pipeline (*Jenkins*)
- Scripts für den automatisierten Upload im PlayStore und AppStore (*Bitrise und Fastlane*)
- Scripts für automatisierte Releases und Release-Notes



- Push-Notifications angesteuert über SDA Notification Services (*Firebase*)
- Breaking News Feature zum Aussteuern selbiger in der App an den Endkunden



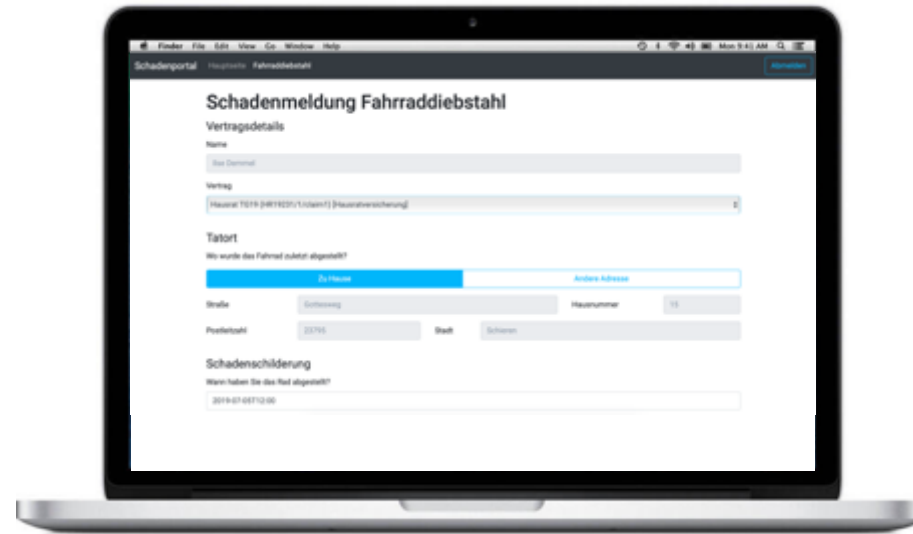




Contract Stack

Basis Plate

Claim Stack



Herausforderung

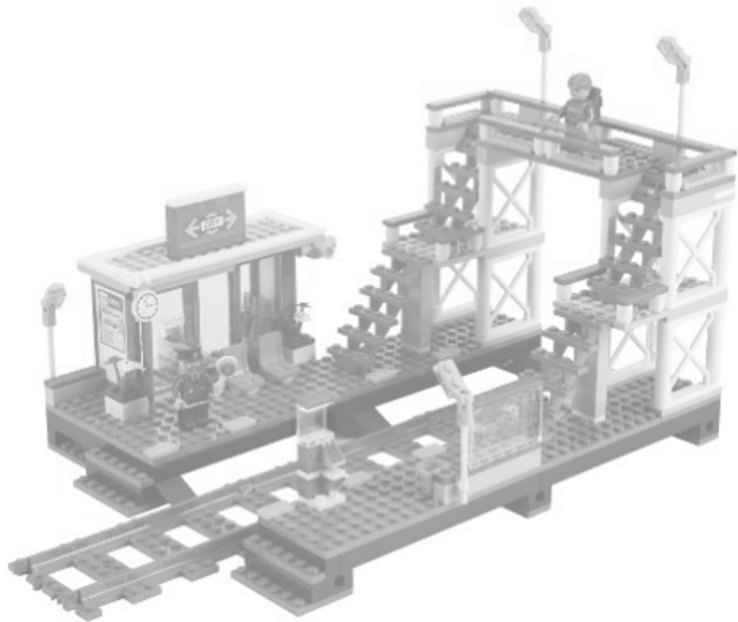
1. Endkunde: Bei Fahrraddiebstahl, muss ich den Schadensprozess aufwendig mit der Versicherung per Post und Telefon abwickeln
2. Versicherung: Hoher Aufwand für manuelle Sachbearbeitung, keine strukturierten und validierten Daten für eine Dunkelverarbeitung

Lösung

Einreichen des Schadens, inkl. aller notwendigen Informationen und Prüfung der Schadensabdeckung über ein Webportal zur Einreichung des Fahrraddiebstahls

Status

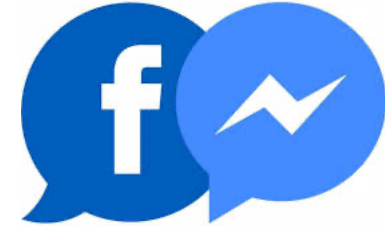
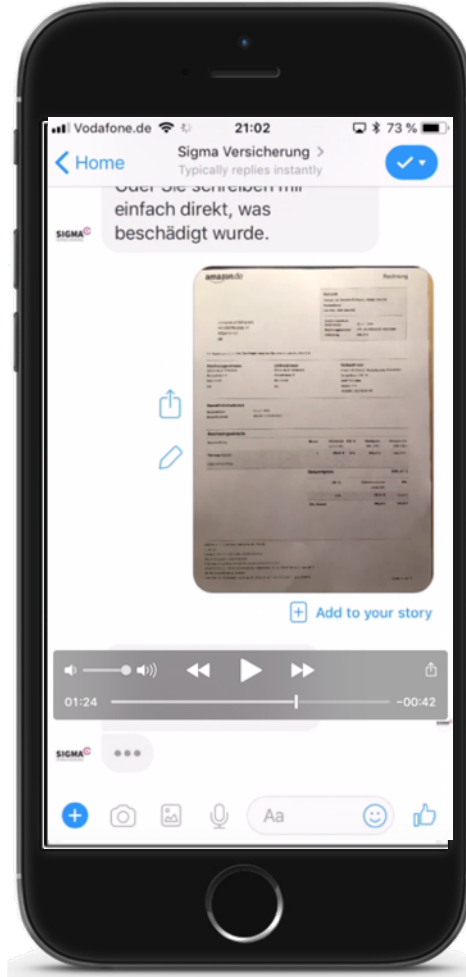
Umgesetzt, bisher noch nicht bei Kunden produktiv gesetzt



Contract Stack

Basis Plate

Claim Stack

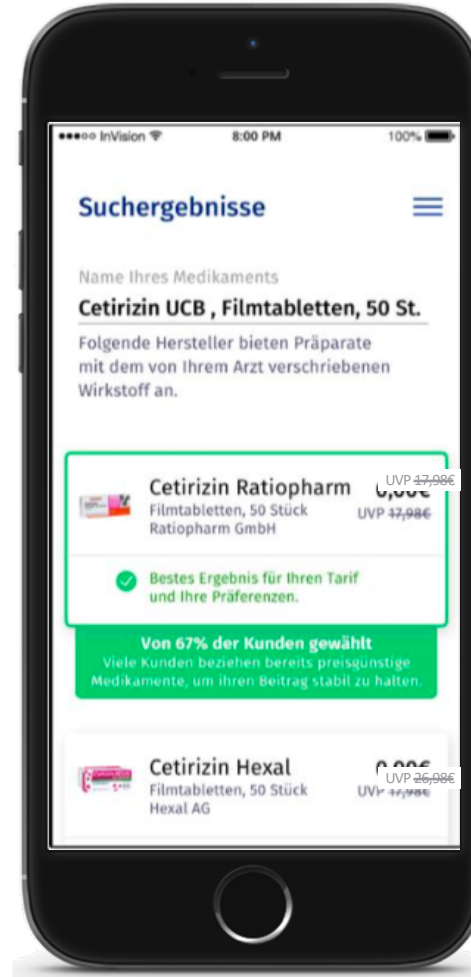




Partner Stack

Basis Plate

Contract Stack



Herausforderung

1. Endkunde: Ich muss Medikamente zunächst selber bezahlen und dann eine Rechnung bei der Versicherung einreichen
2. Versicherung: Ich möchte den Versicherten dazu motivieren, dass er günstigere Generika kauft, um Leistungsausgaben zu sparen

Lösung

Eigene App für das Medikamentenmanagement (Scan Rezept, Finden Apotheke, Präsenz von Generika, Direktes Bezahlen, autom. Verrechnung Endkunde)

Status

In Konzeption