



inosca

Die kantonale Open Source Community

 Adfinis

Christian Zosel

Software Engineer

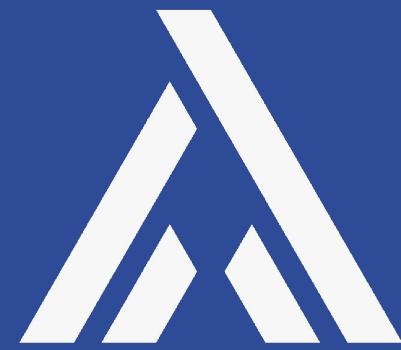
Adfinis

⌚ Bern, Switzerland

✉️ christian.zosel@adfinis.com

㏌ <https://linkedin.com/in/czosel>





Adfinis

Vorstellung Adfinis



Wir sind ein Team von über 100 Open-Source-Experten, die Sie bei Ihren technischen Herausforderungen unterstützen



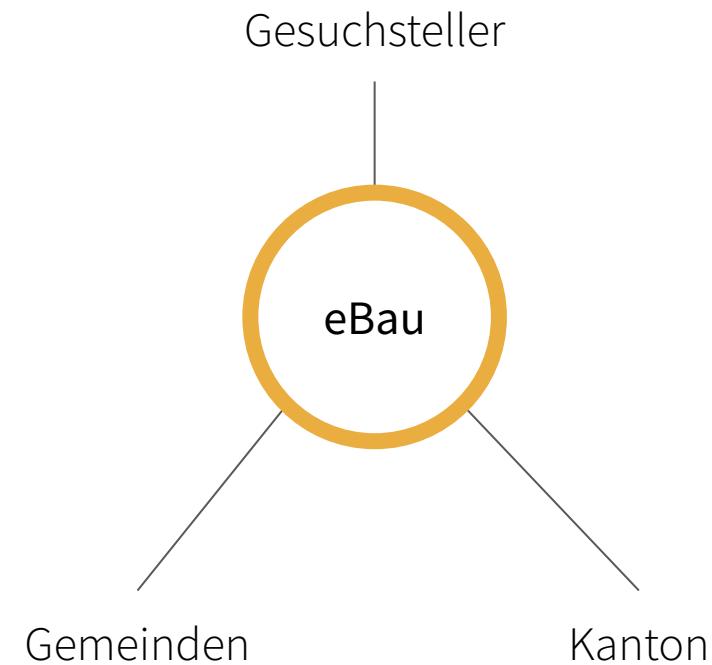
Wir sind Experten in der Arbeit für stark regulierte Branchen und konzentrieren uns auf Qualität statt Quantität



Wir teilen unser Wissen und befähigen Ihr Team durch unser Inner Source Mindset



inosca Community



Inhalt

- › Die inosca-Story
- › Organisation
- › Zusammenarbeit
- › Architektur
- › Betrieb
- › Geschäftsmodell
- › Auswirkungen



Die inosca-Story



Die inosca-Story



2011

› Individuelle Lösung



Story time



- › Gemeinsamkeiten und Unterschiede verstehen
 - › Startpunkt für gemeinsame Lösungen
 - › Beginn technischer Modernisierung



Story time



- › Einheitliche Code-Basis und Entwicklungsprozess
- › Etablierte Referenzimplementation für neue Kantone
- › Modernisierte technische Basis



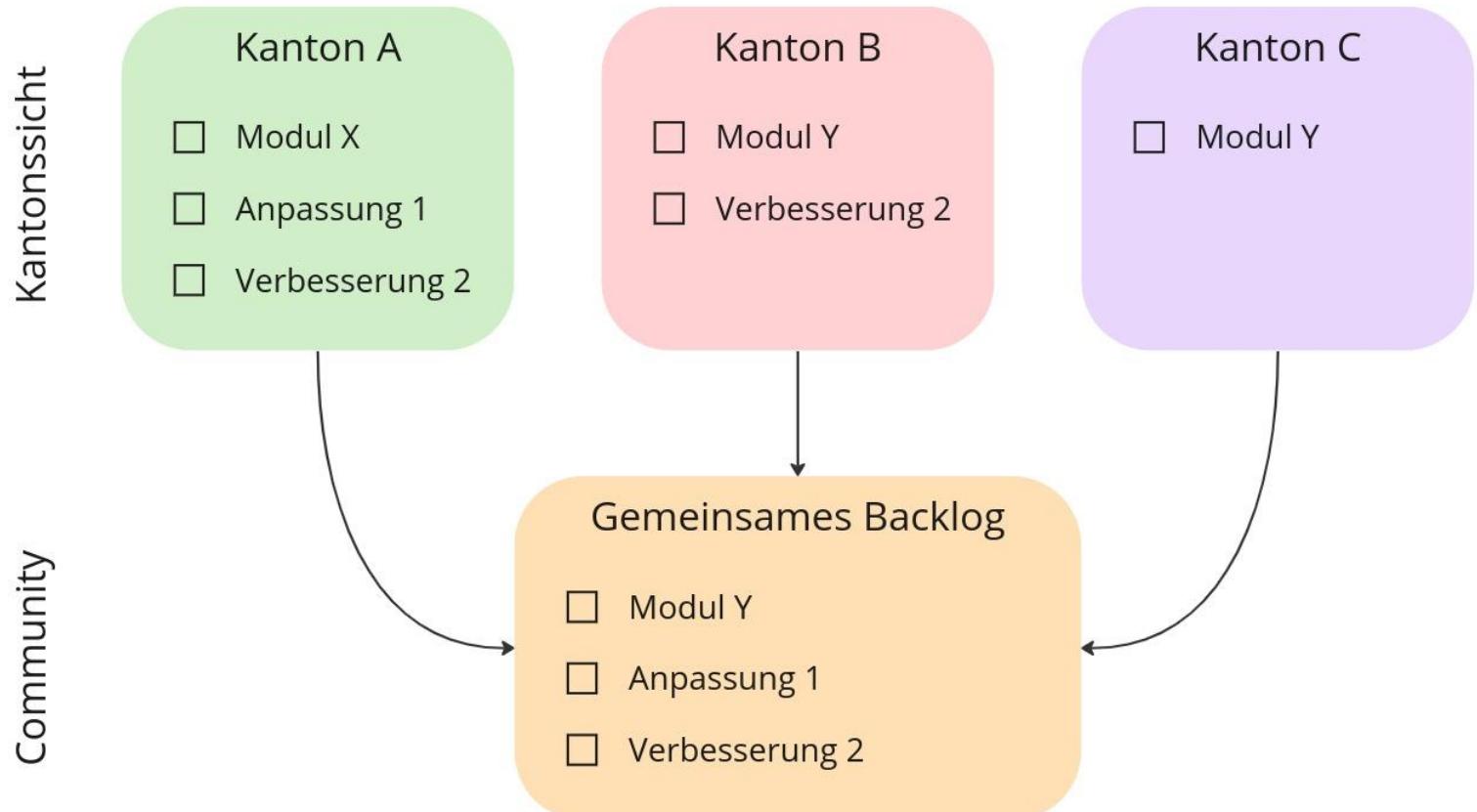
Organisation



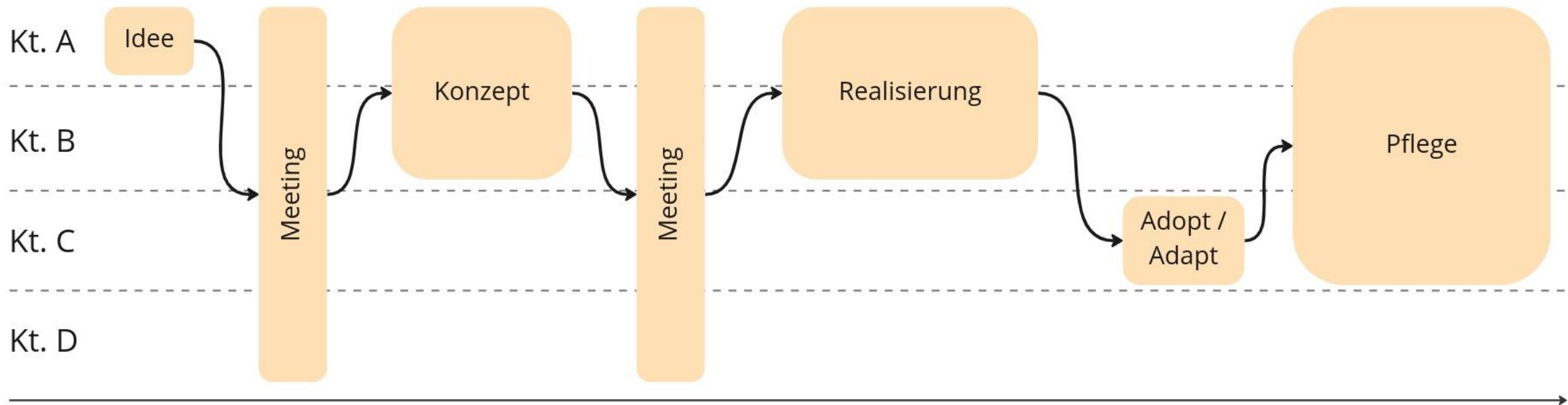
Organisation

Regelmässige
Online-Meetings

- › Genereller Austausch
- › Gemeinsames
Change-Management



Lifecycle eines Features



Spielregeln

- › Entwicklungsergebnisse sind frei verfügbar
- › Wer spezifiziert, finanziert
- › Kostenschlüssel

inosca.ch
Bewilligungsprozesse
UR, SZ, BE, SO

Interkantonale Entwicklungsgemeinschaft für elektronische

inosca

Statement Stand 2.11.2022 / Version 5

Einführung

inosca.ch ist eine interkantonale Software-Entwicklungsgemeinschaft für elektronische Bewilligungsprozesse (in=innovativ, os=open source, ca = canton). Sie wurde im Jahre 2020 von den Kantonen Uri, Schwyz, Bern und Solothurn gegründet. Zuvor bestand eine mehrjährige Zusammenarbeit unter anderem Namen. Mit dem vorliegenden Statement regelt die Gemeinschaft die Mechanismen der Zusammenarbeit.

Strategie / Ziele

Aktuell werden unter dem Label von inosca verschiedene Software Varianten eingesetzt. Dieser Zustand ist vorübergehend. Langfristig wird für alle Projekte innerhalb der Gemeinschaft inosca eine einheitliche technische Basis angestrebt (Roadmap siehe Anhang). Dadurch können Synergien genutzt werden, die sich sowohl inhaltlich als auch finanziell positiv auf die Projekte auswirken. Ziel ist es einerseits, dass alle Mitgliederkantone von inosca einheitliche Technologien verwenden (z.B. Keycloak / CALUMA). Andererseits soll angestrebt werden, dass die Kantone künftig in möglichst vielen Bereichen (Formular, Workflow, Module, Layout usw.) auch die gleichen Applikationen und Prozesse verwenden und ein gemeinsames Changemanagement anstreben.

Dank dem Grundsatz, künftig Open Source Technologien einzusetzen, die für jeden Dienstleister frei zugänglich sind, besteht die Möglichkeit, innerhalb der Gemeinschaft inosca.ch mehrere Dienstleister zu berücksichtigen.

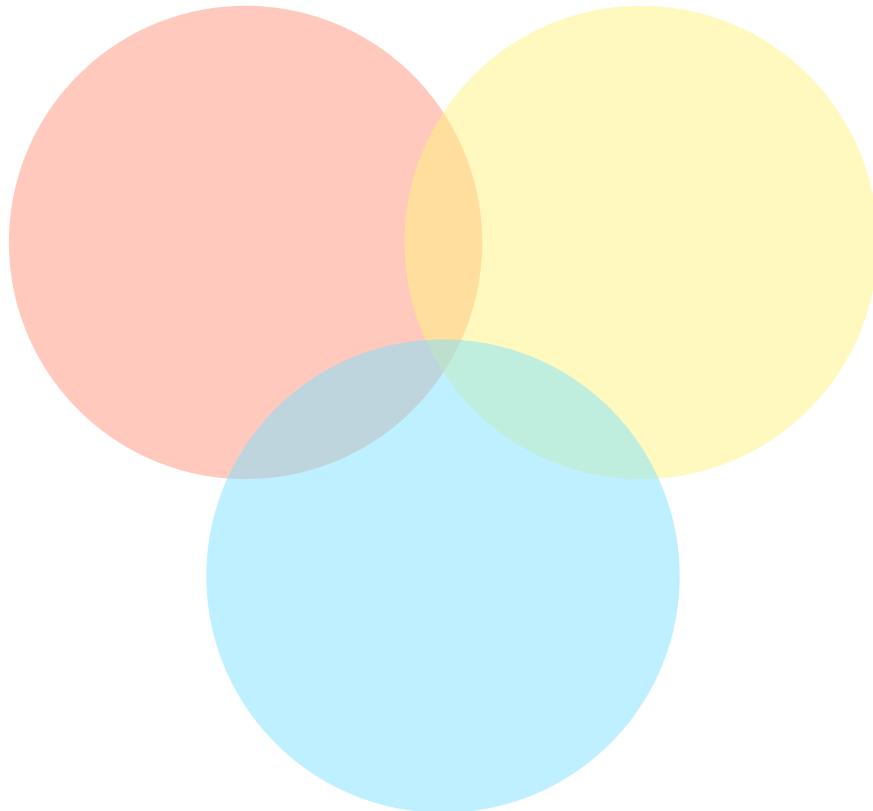
Die Teilnehmer an inosca wirken bei der Verbreitung von Informationen über die Gemeinschaft und deren Strategie proaktiv mit, indem sie diese interessierten Kantonen vorstellen. Sie engagieren sich, die Gemeinschaft inosca und deren Philosophie im eigenen Interesse und im Interesse aller beteiligten Kantone mit



Zusammenarbeit

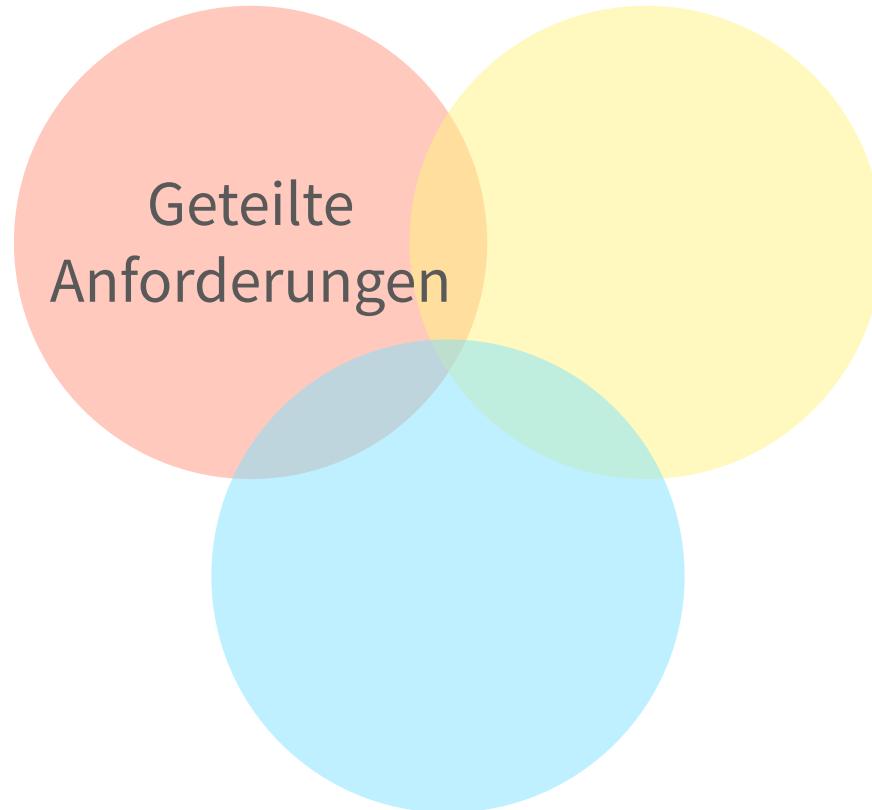


Grundlagen für erfolgreiche Zusammenarbeit



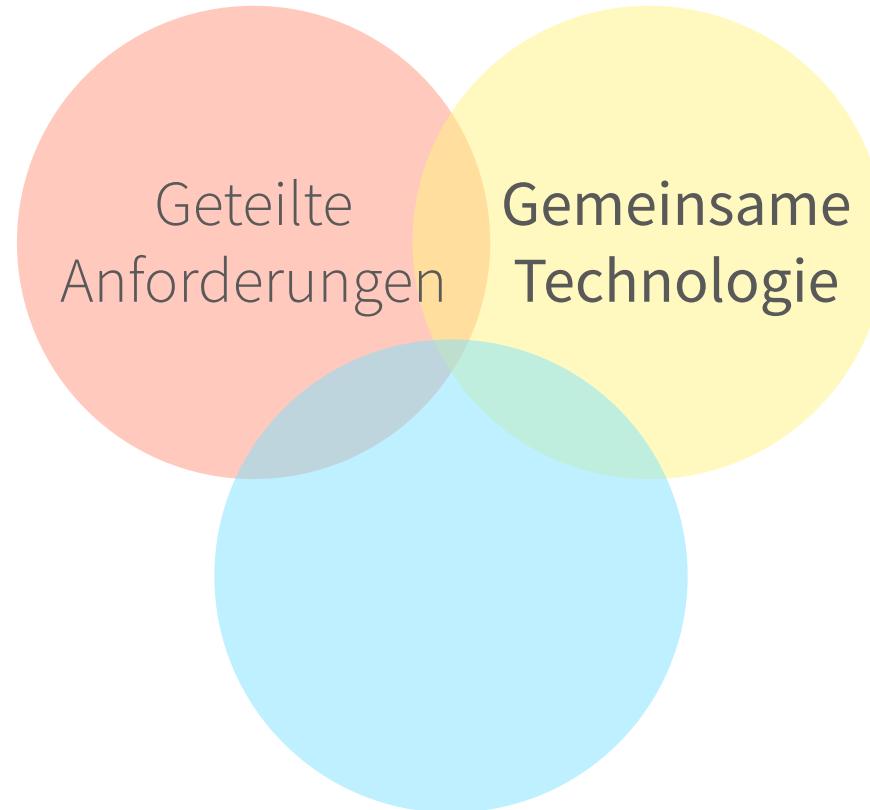
Grundlagen für erfolgreiche Zusammenarbeit

- › 80-90% Übereinstimmung



Grundlagen für erfolgreiche Zusammenarbeit

- › 80-90% Übereinstimmung

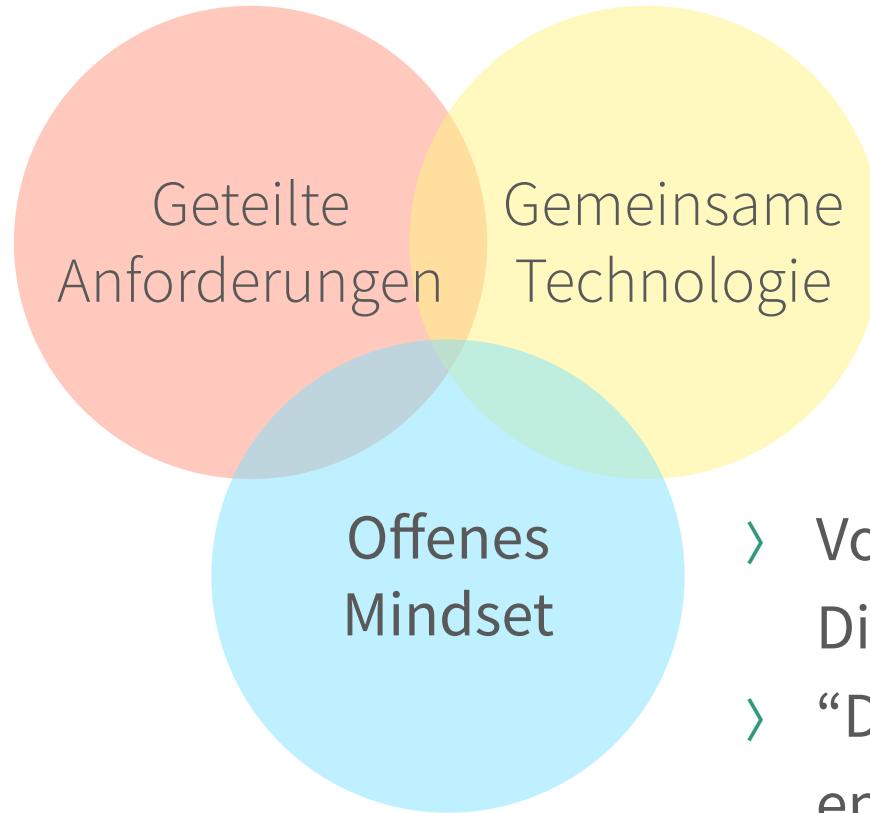


- › Synergieeffekt bis hin zur technischen Umsetzung
- › Handlungsfähigkeit



Grundlagen für erfolgreiche Zusammenarbeit

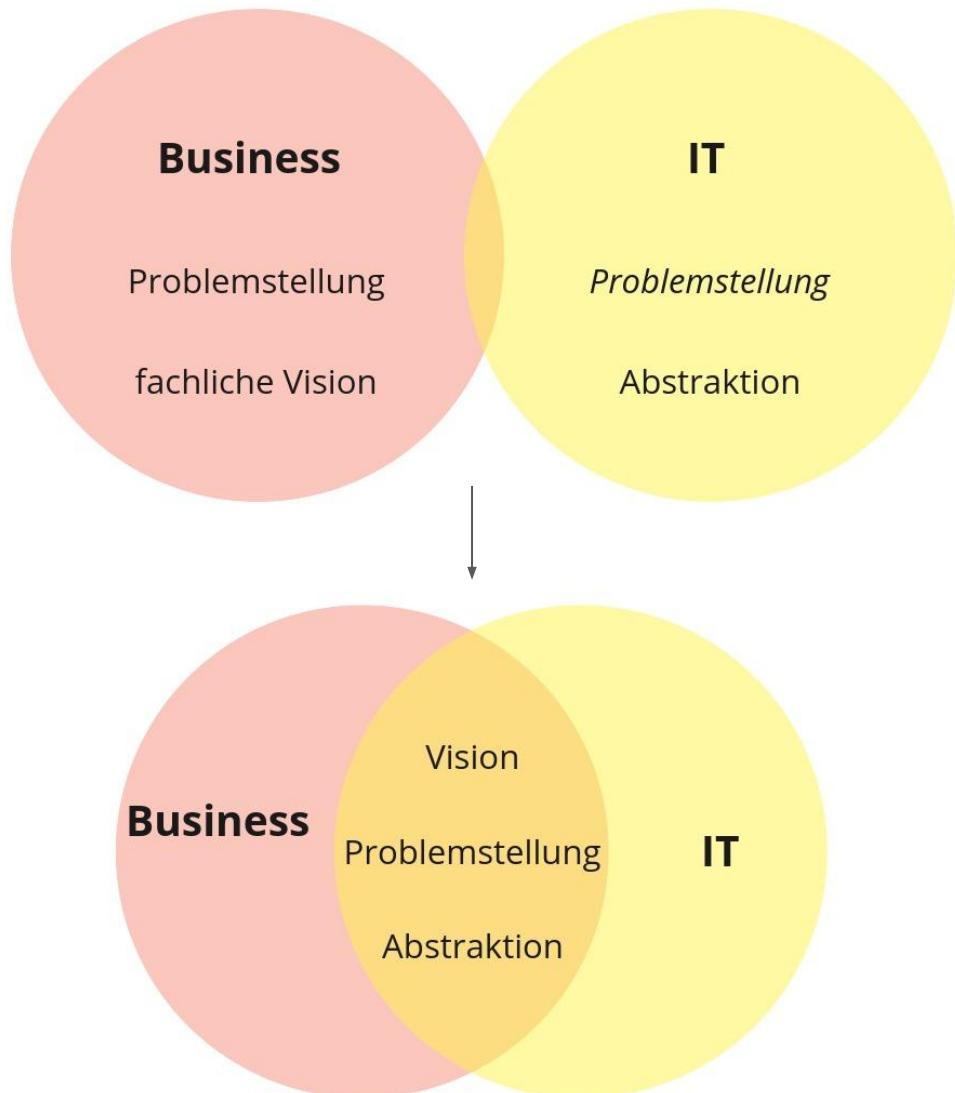
- › 80-90% Übereinstimmung



- › Synergieeffekt bis hin zur technischen Umsetzung
- › Handlungsfähigkeit
- › Von Kantonen *und* Dienstleistern!
- › “Die besten Lösungen entstehen gemeinsam”
- › “Wir teilen unsere Software offen”



Zusammenarbeit Business und IT

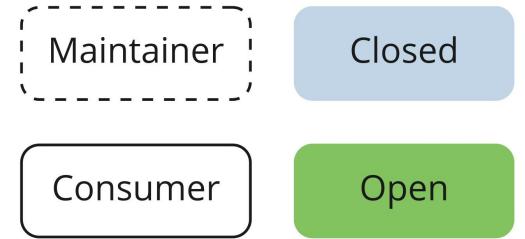


- › Technisches vs. fachliches Know-how:
Horizonterweiterung auf beiden Seiten
- › Gemeinsame Sprache finden
- › Austausch auf Augenhöhe, Transparenz
- › Vertrauen und gegenseitige Wertschätzung

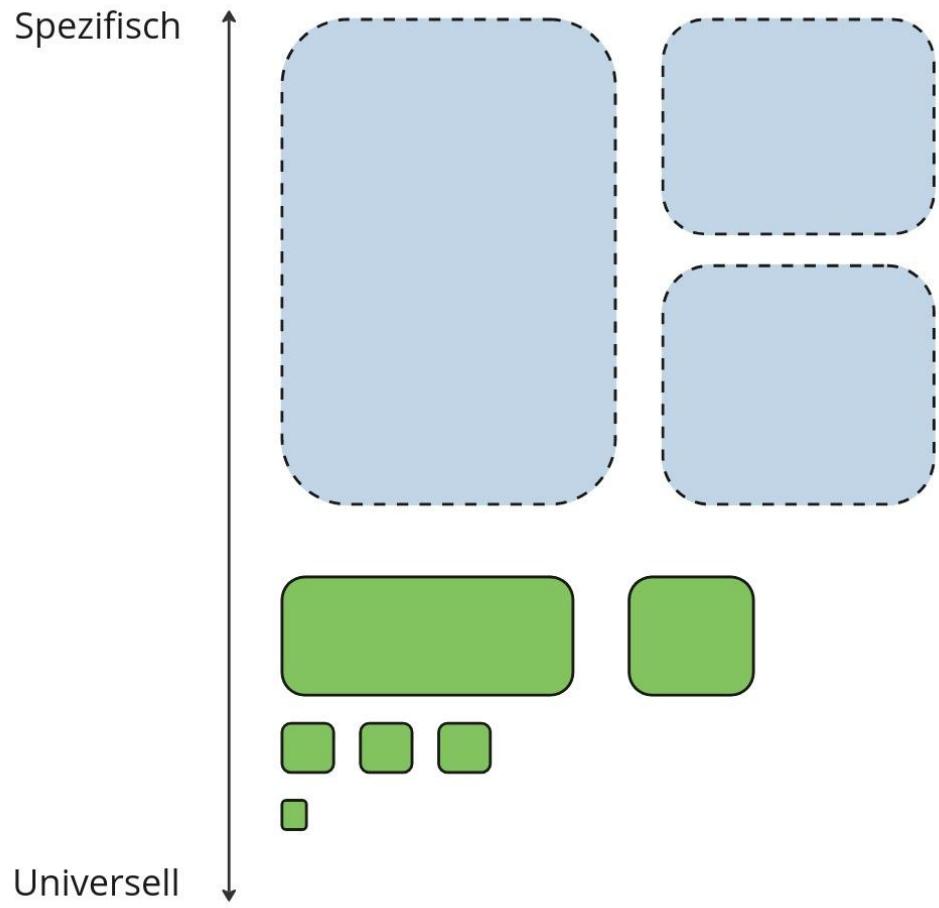


Architektur

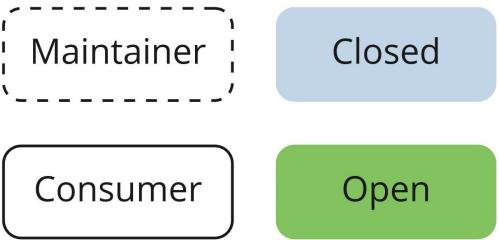
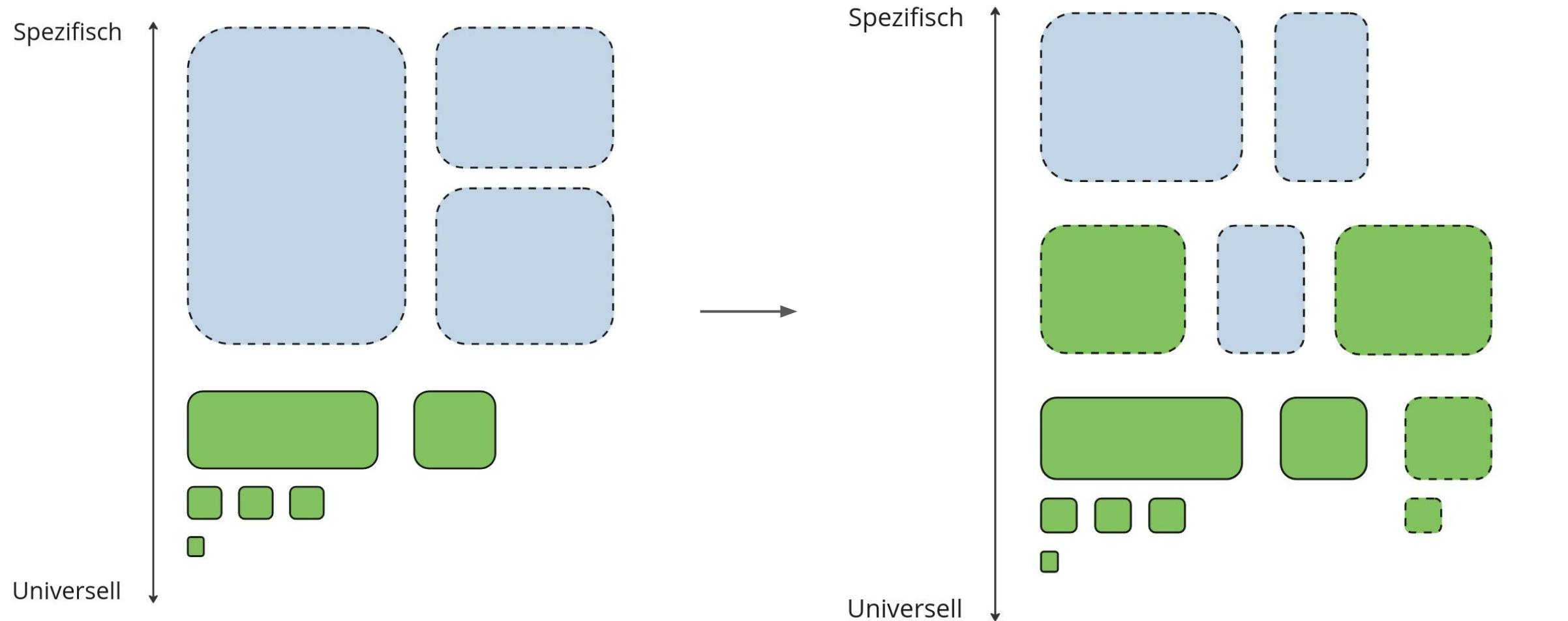




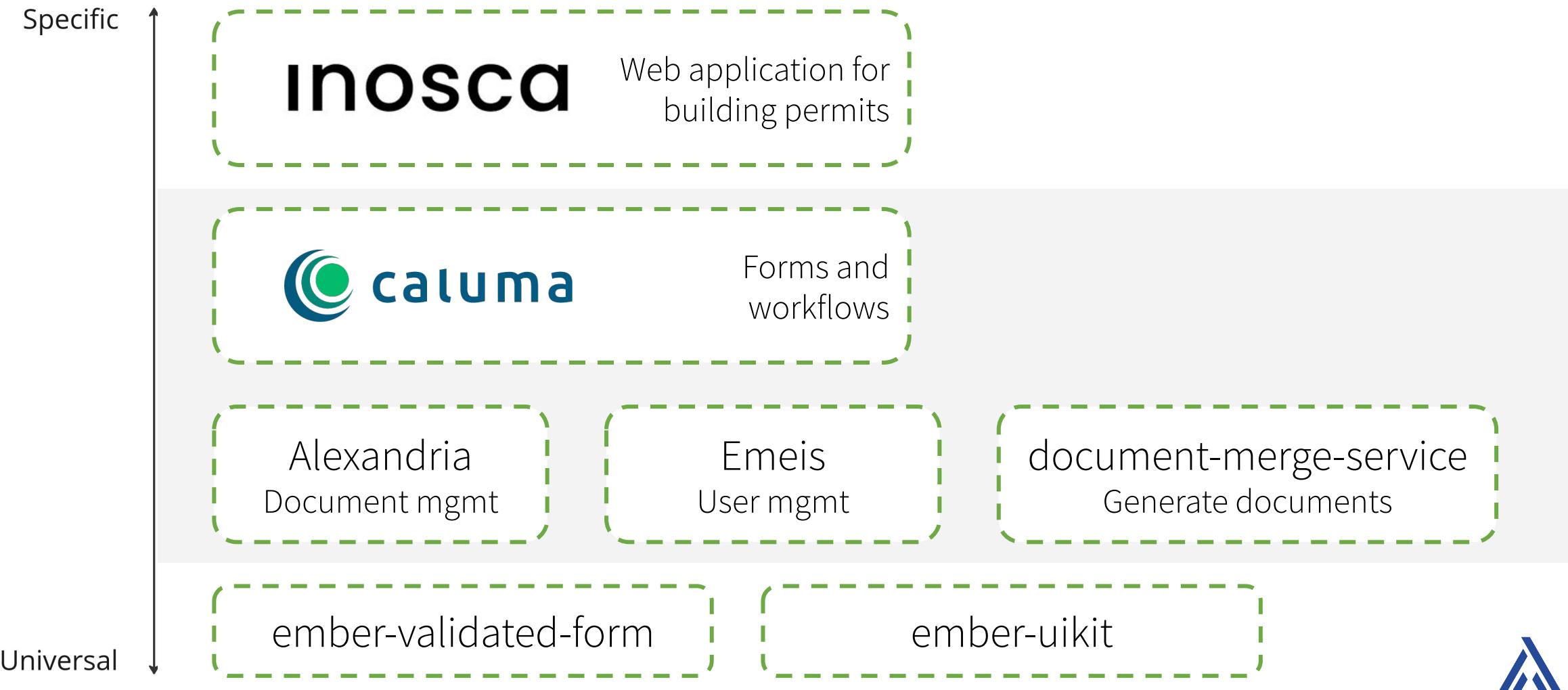
Open Source Architektur



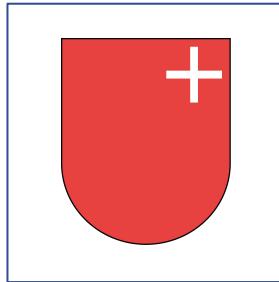
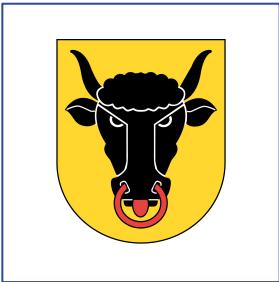
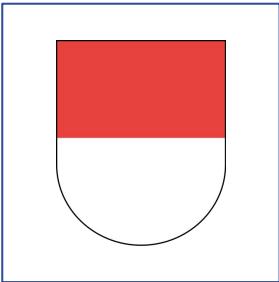
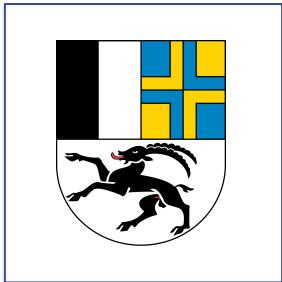
Open Source Architektur



Open Source Komponenten



Gemeinsame Basis, individuelle Ausgestaltung



Kantonsspezifische
Konfigurationen

Standardisierte
Fachapplikation

Generisches
Framework



inosca Technologiestack

› Programmiersprachen



› Frameworks



› Storage

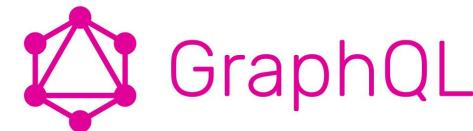


› Weiteres



Offene Standards

- › Authentifizierung
- › Kommunikation Frontend - Backend
- › Schnittstellen
(Gemeindesoftware, GWR)



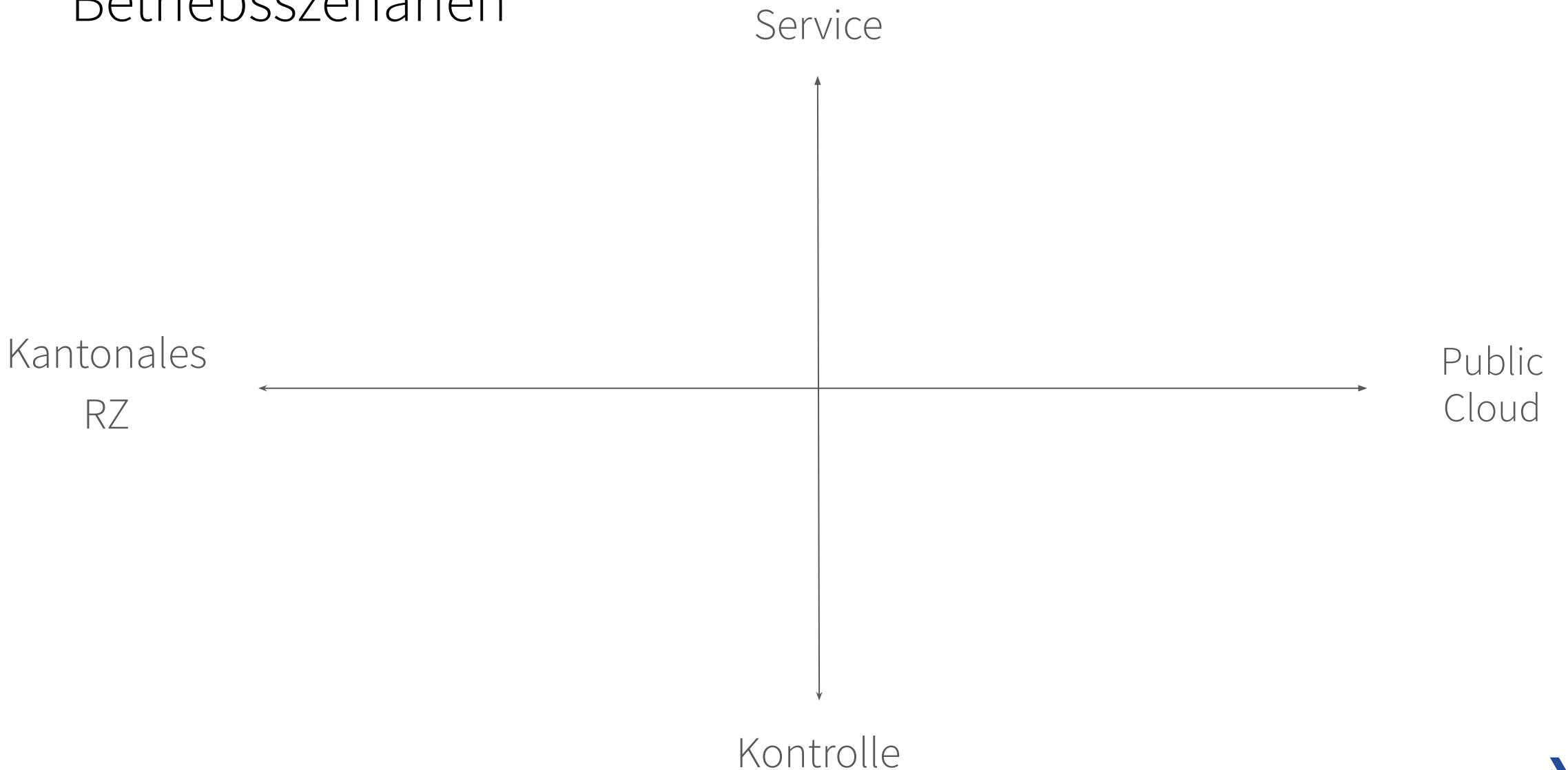
{REST:API}



Betrieb



Betriebsszenarien



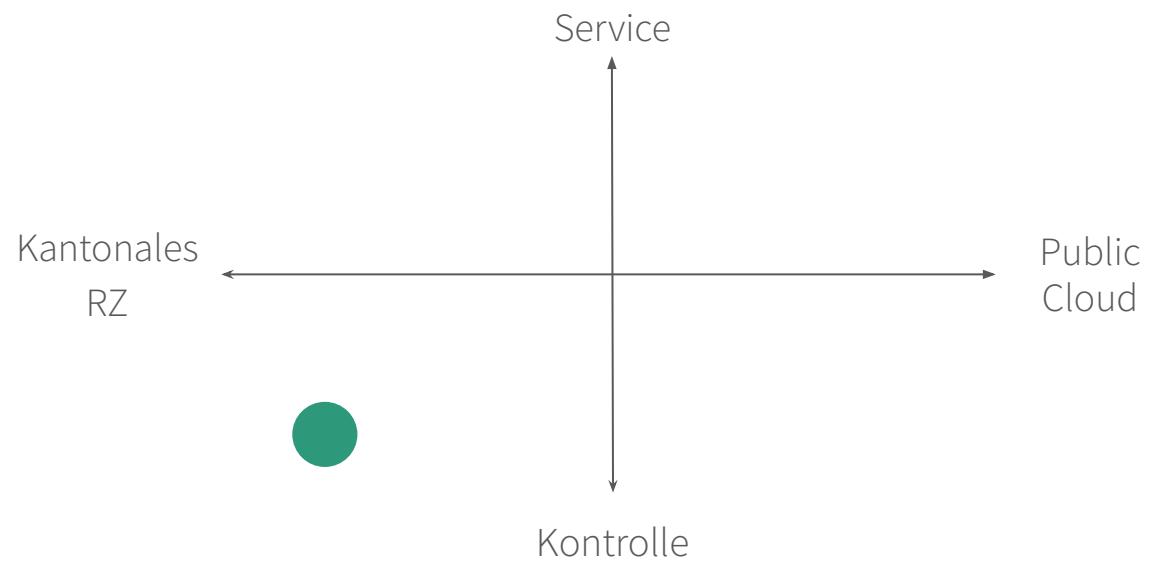
Szenario 1: Betrieb durch Kanton, volle Kontrolle

Kantonale Informatik

- › Hardware, Netzwerk
- › Monitoring + Backup
- › OS Updates
- › Einspielen neuer Releases

Dienstleister

- › Ausliefern neuer Releases



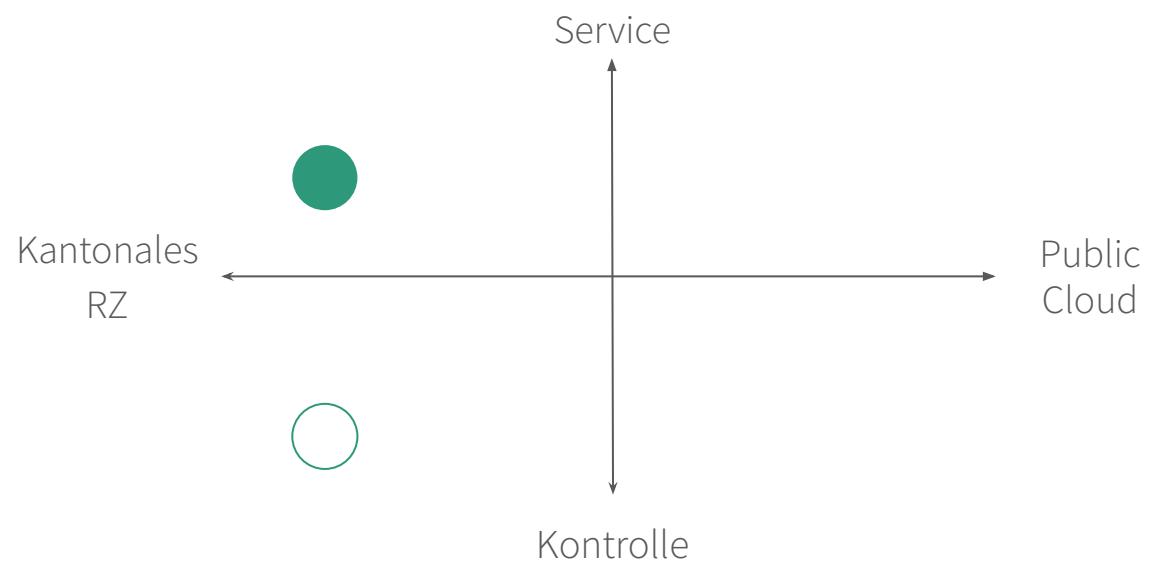
Szenario 2: Betrieb durch Kanton, Pflege durch Dienstleister

Kantonale Informatik

- › Hardware, Netzwerk

Dienstleister

- › Releases
- › OS Updates
- › Monitoring + Backup



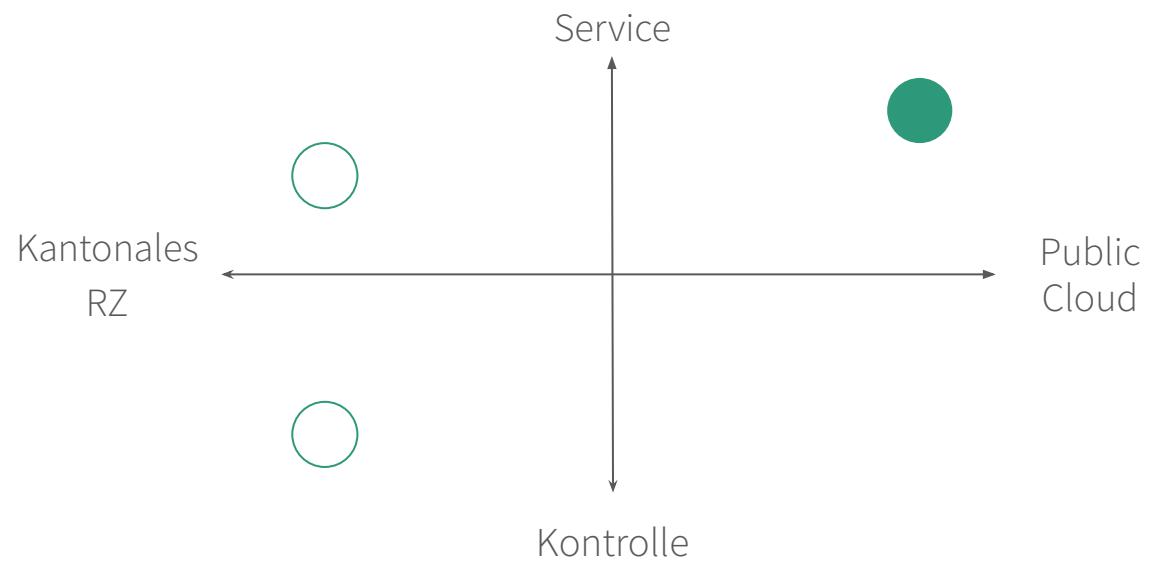
Szenario 3: Betrieb in Cloud, Pflege durch Dienstleister

Cloud Provider

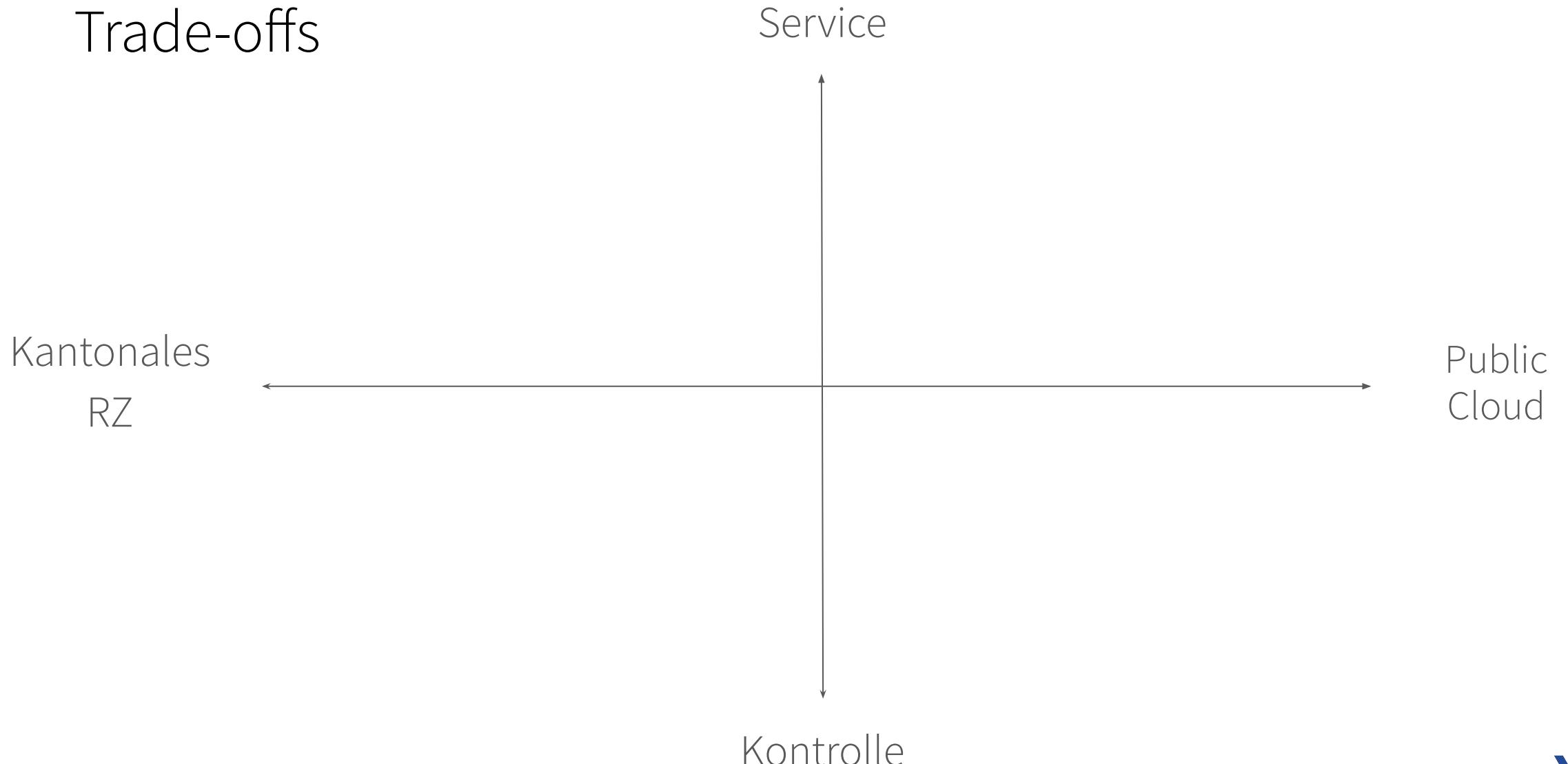
- › Hardware, Netzwerk
- › Basisdienste (Datenbank, Storage, Kubernetes Cluster)

Dienstleister

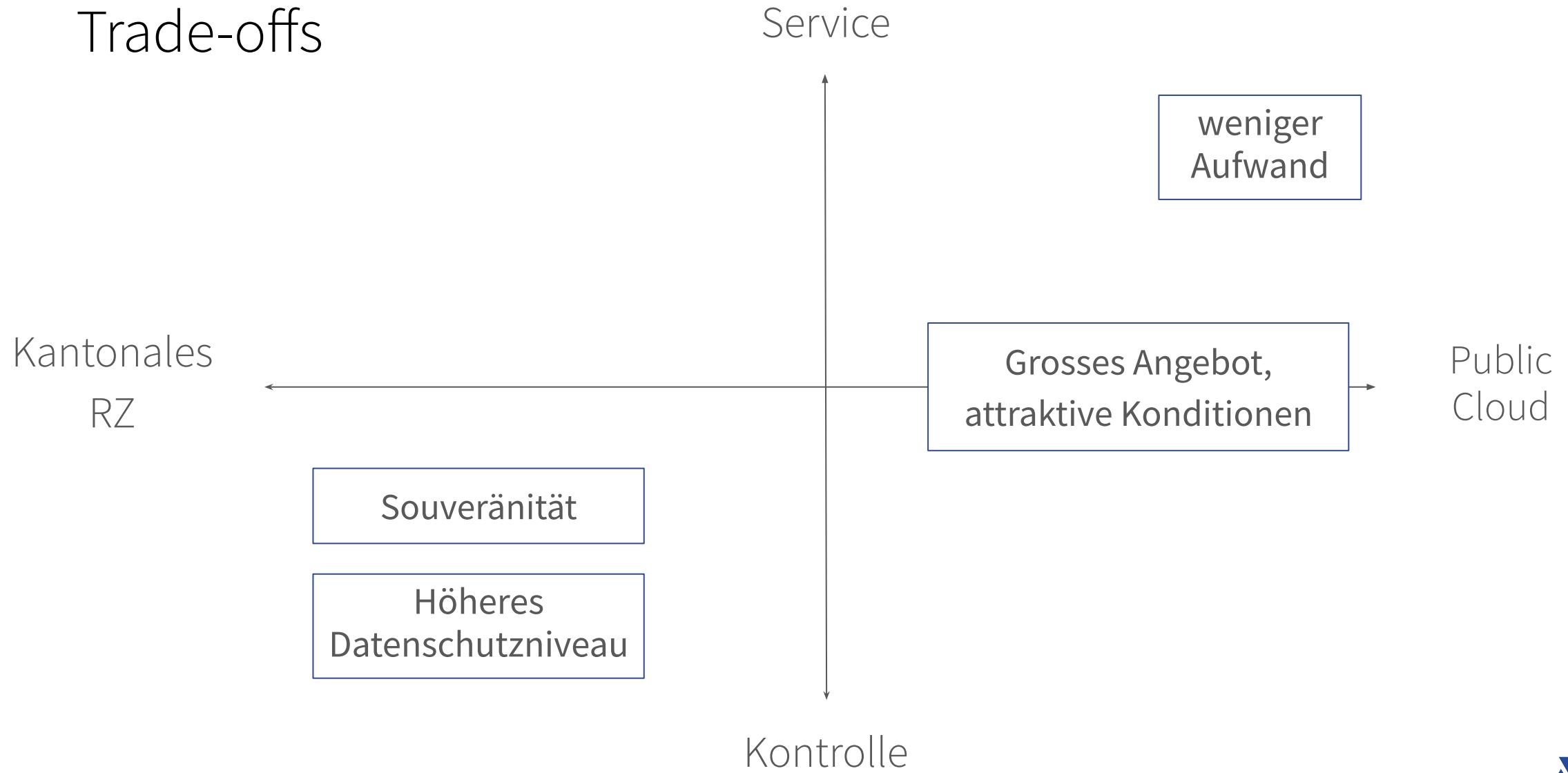
- › Releases
- › Monitoring + Backup
- › OS Updates (falls nötig)



Betriebsszenarien: Trade-offs



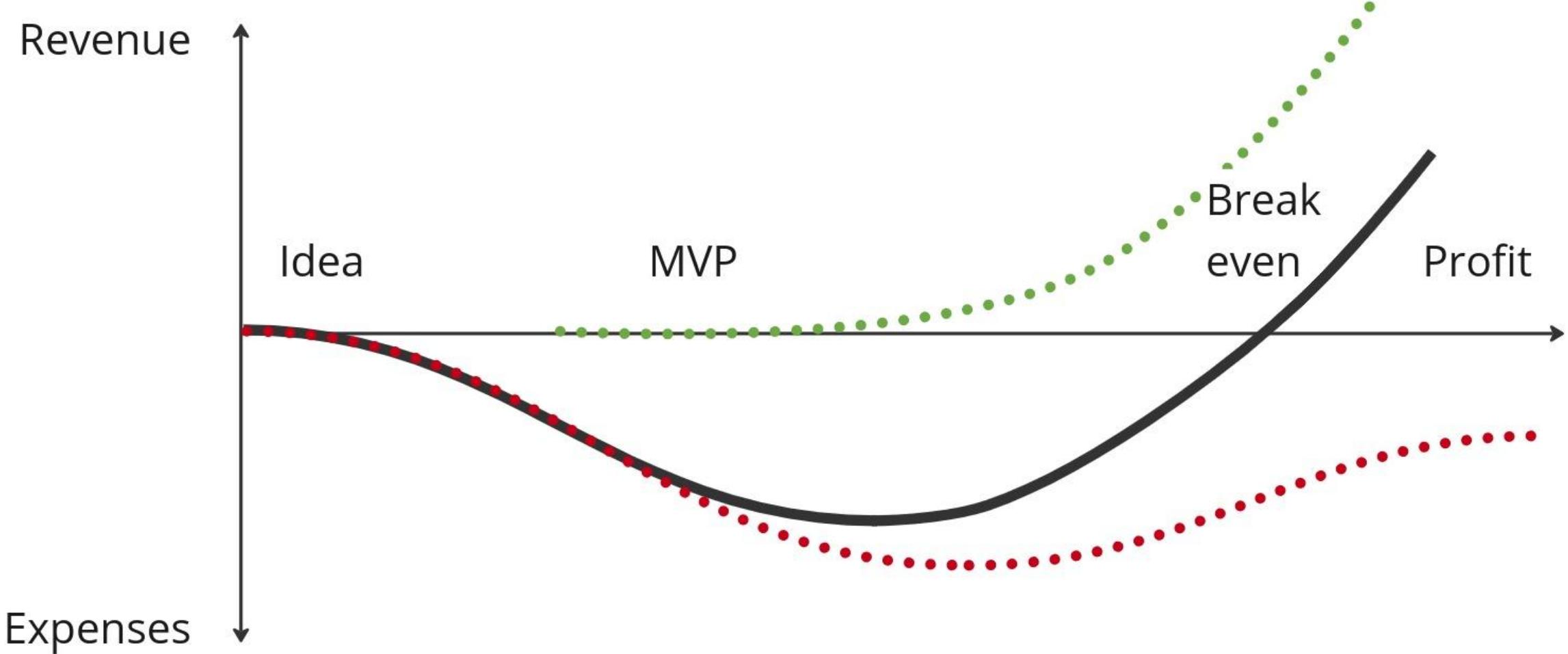
Betriebsszenarien: Trade-offs



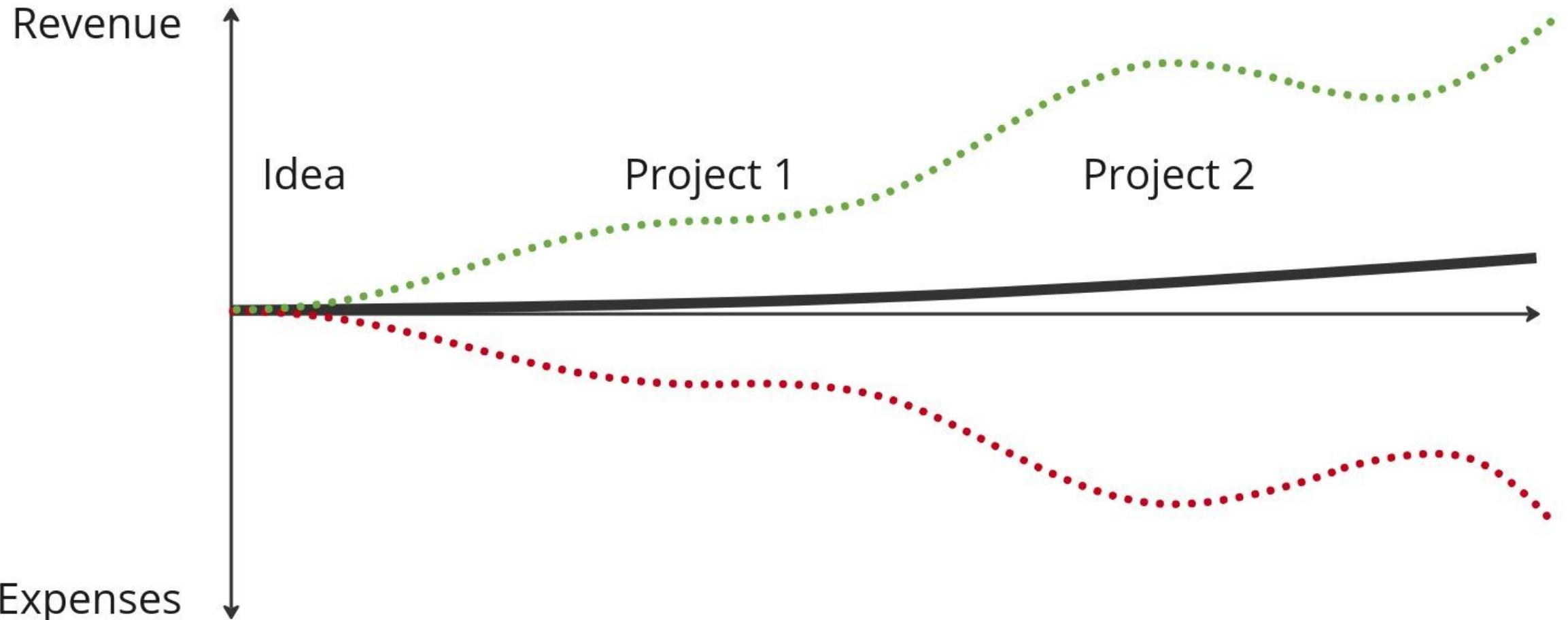
Geschäftsmodell



Geschäftsmodell «Investment»



Geschäftsmodell «open source»



Geschäftsmodelle im Vergleich

«Invest»

Einnahmen

Lizenzgebühren

«Open Source»

Projekte, Support, Hosting



Geschäftsmodelle im Vergleich

«Invest»

Einnahmen

Lizenzgebühren

Differenzierung

IP, Marktanteil

«Open Source»

Projekte, Support, Hosting

Know-How, Kundenbeziehung



Geschäftsmodelle im Vergleich

«Invest»

Einnahmen

Lizenzgebühren

Differenzierung

IP, Marktanteil

Kontrolle

Hersteller

«Open Source»

Projekte, Support, Hosting

Know-How, Kundenbeziehung

Community



Geschäftsmodelle im Vergleich

«Invest»

Einnahmen

Lizenzgebühren

Differenzierung

IP, Marktanteil

Kontrolle

Hersteller

Interessen

Investoren

«Open Source»

Projekte, Support, Hosting

Know-How, Kundenbeziehung

Community

Kunden, Endnutzer



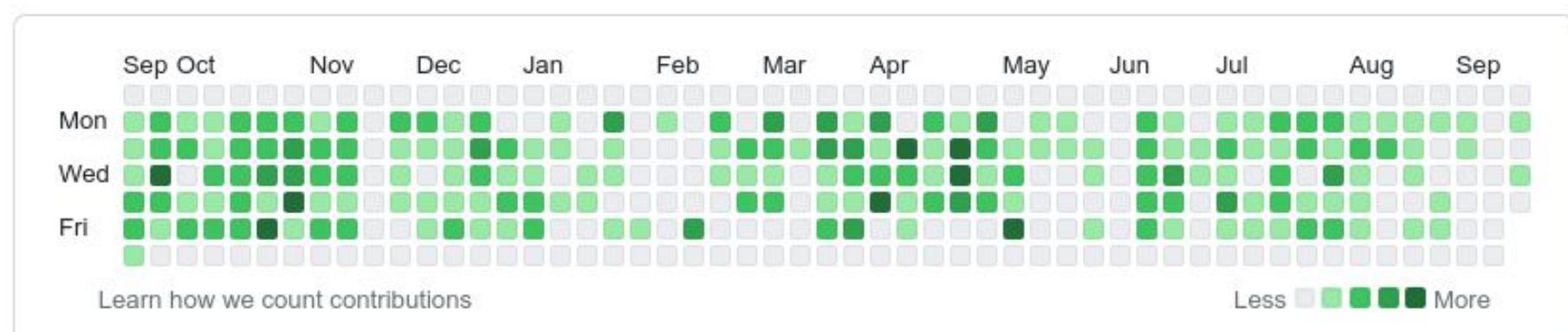
Auswirkungen



... für Dienstleister

- › Effizienzsteigerung, Synergien zwischen Projektteams
- › Fokussierung vom Portfolio
- › Code-Qualität, Architektur
- › Arbeitgeberattraktivität
 - › Individuelles Portfolio
 - › Portabilität von Fähigkeiten
 - › Nachhaltigkeit, «Zurückgeben»

1,631 contributions in the last year



... für Beschaffende

- › Know-How Transfer auf fachlicher Ebene
- › Synergieeffekte
 - › Kostenersparnis
 - › Entwicklungsgeschwindigkeit
 - › Reduzierte Projektrisiken
- › **Flexibilität** und reduzierter Vendor Lock-In
- › Grösserer **Einfluss** auf Roadmap
- › **Transparenz** und unabhängige Auditierbarkeit



... für Endnutzer

- › Transparenz / Datenschutz
- › Möglichkeit, sich zu **beteiligen** (Civic tech)



Vielen Dank!



Stay in Touch

Inosca

inosca.ch

 **Adfinis**

adfinis.com



Anhang



Open Source



inosca / ebau

Type to search

Code Issues 1 Pull requests 1 Actions Projects Wiki Security 31 Insights Settings

ebau Public

Edit Pins Unwatch Fork Starred

main 2 branches 0 tags Go to file Add file Code

About

Electronic building permit application for swiss cantons.

inosa.ch/

switzerland egovernment building-permits

Readme EUPL-1.2 license Activity 9 stars 2 watching 4 forks Report repository

Table of Contents

- Overview
 - Folder structure
 - Modules
- Requirements
- Development
 - Basic setup
 - Predefined credentials

Contributors 15

eBau by inosca

Electronic building permit application for Swiss cantons.

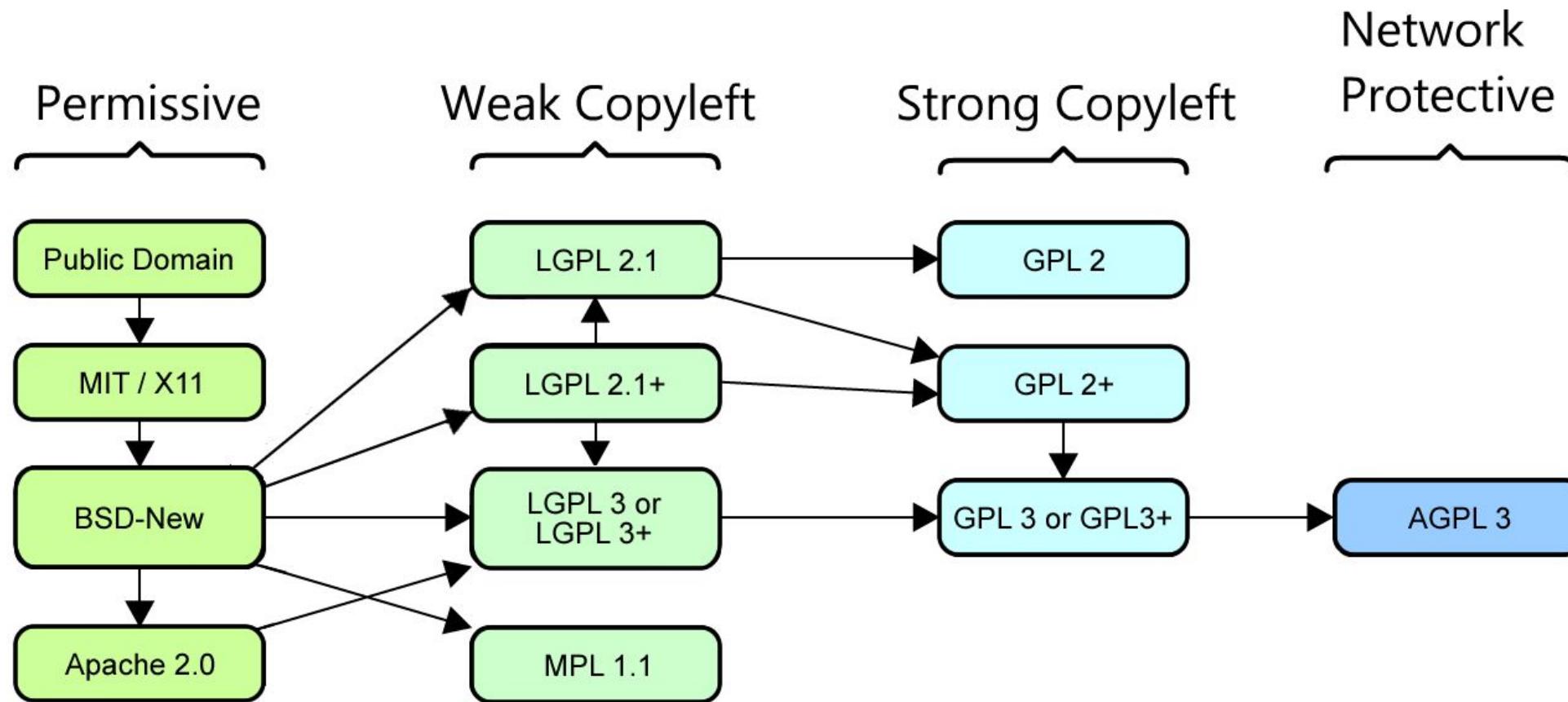
Table of Contents

- Overview
 - Folder structure
 - Modules
- Requirements
- Development
 - Basic setup
 - Predefined credentials

Report repository

Contributors 15

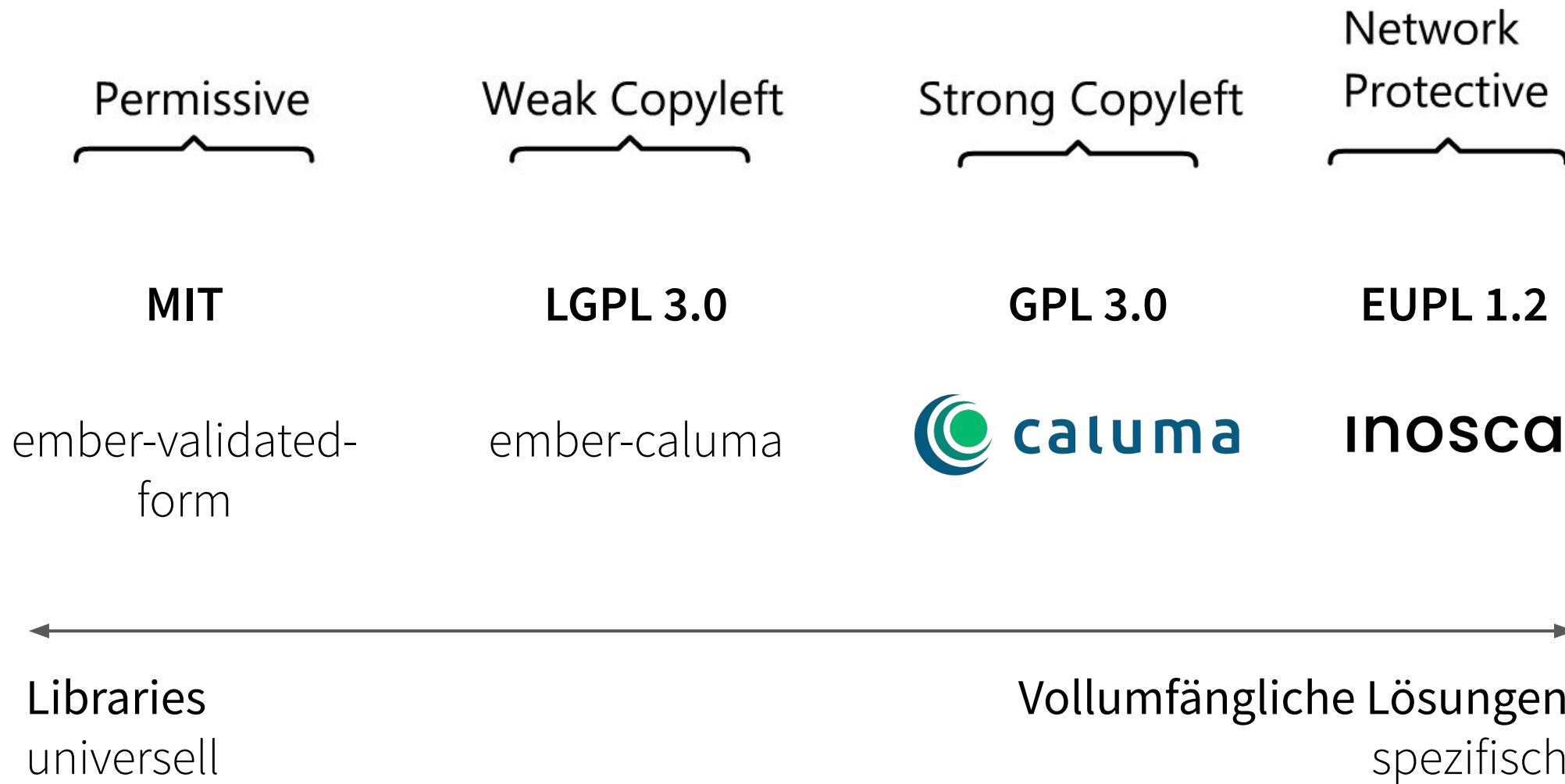
Lizenzwahl



[Speakman, F., Compatibility Relationships Between Popular F/LOSS Licences, CC BY 3.0](#)



Lizenzwahl



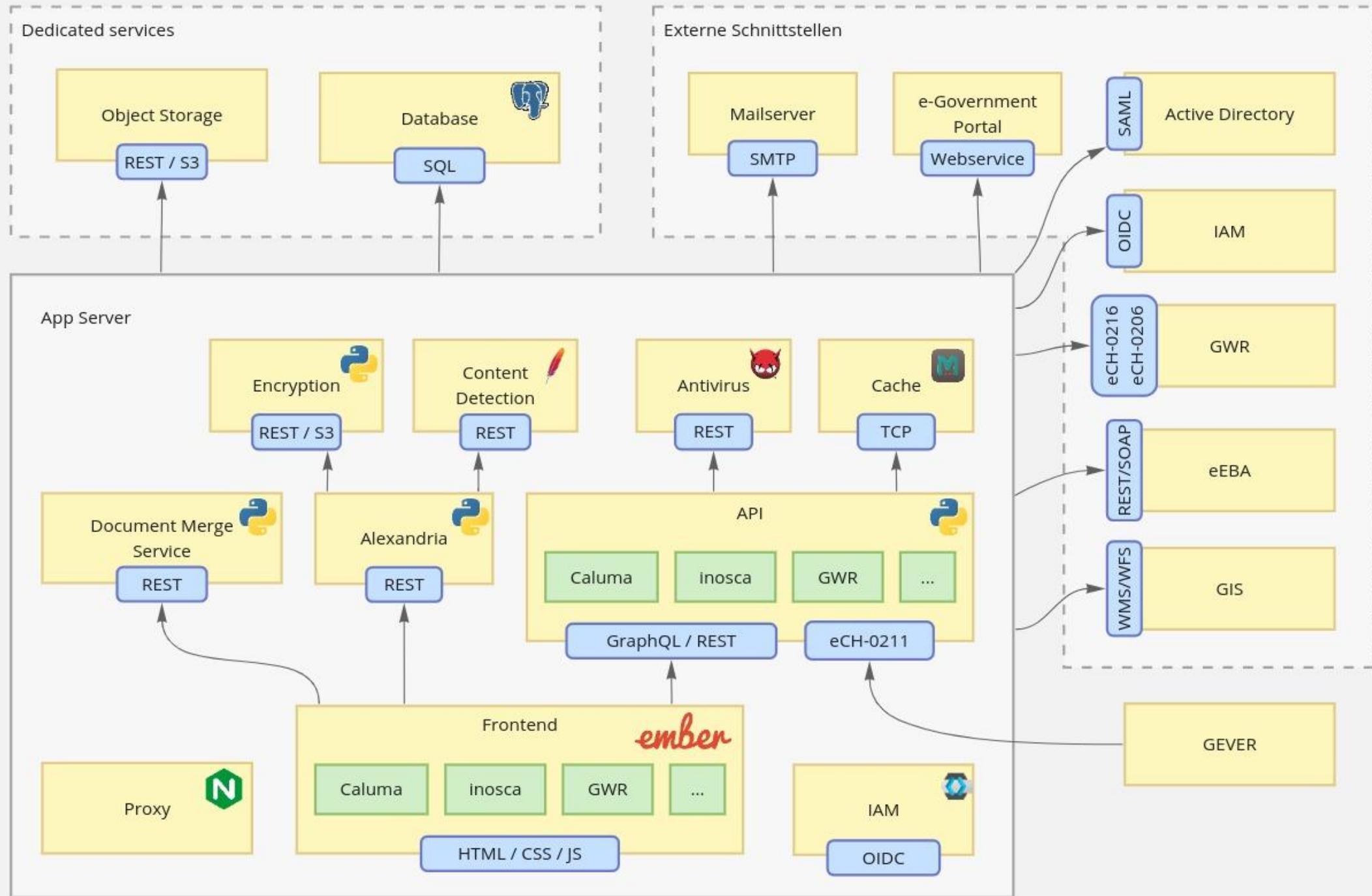
EUPL-1.2 Lizenz

- › Copyleft-Lizenz
- › Übersetzt in 23 Amtssprachen der EU
- › Siehe auch [Leitfaden Lizenzwahl, Kanton Bern](#)



Architektur

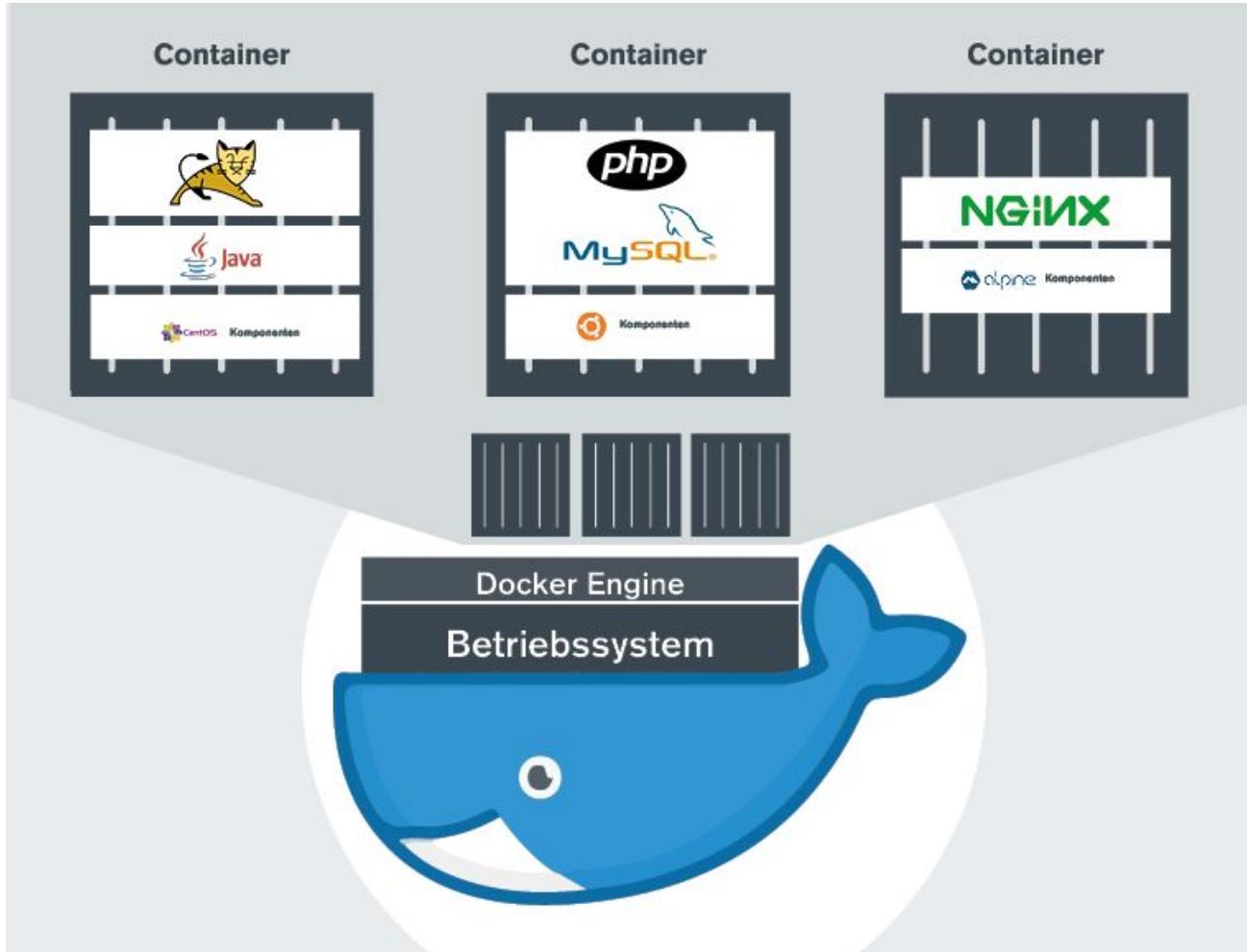




Container



Container



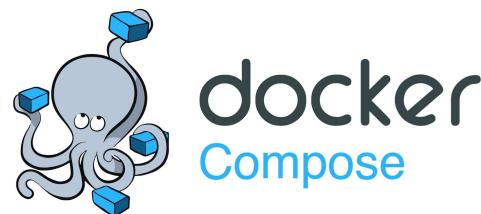
- › Effizient: Leichtgewichtige Form der Virtualisierung
- › Portabel: Enthalten sämtliche Abhängigkeiten
- › Sicher: Laufen isoliert voneinander

Quelle: claranet.de



Container Orchestrierung

- › Ausfallsicherheit
 - › Bei Ausfall einer Festplatte, eines Servers, eines Rechenzentrums
 - › Unterbruchsfreie Deployments
- › Automatisches Skalieren
- › Automatisierung (CI/CD)



Einfach



kubernetes

Vollumfänglich



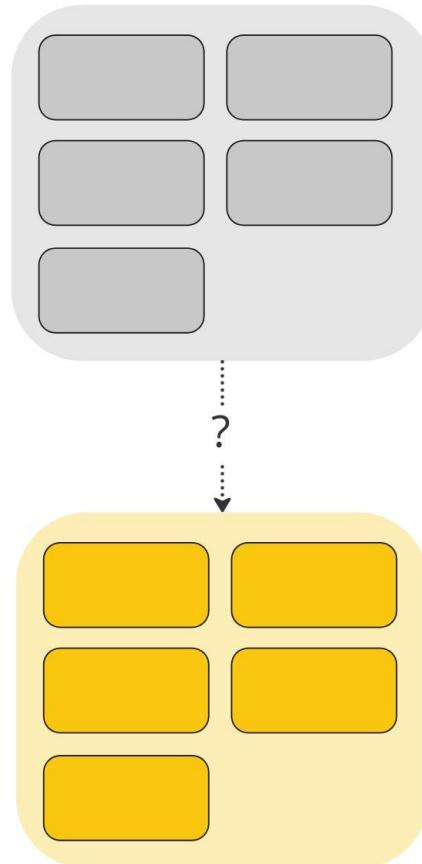
Fallbeispiel: Modernisierung von eBau



Fallbeispiel: Modernisierung von eBau

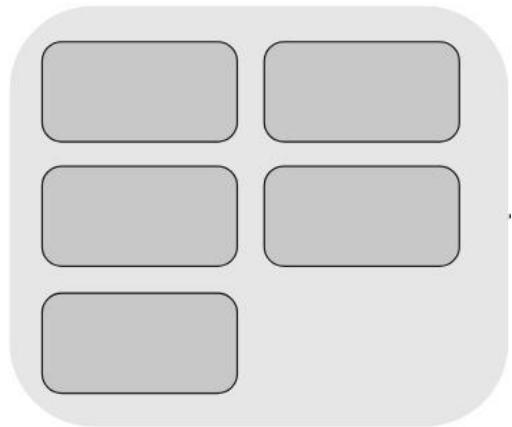
Unübersichtliche Ausgangslage

- › Grosse Community
- › Heterogene Lösungen
- › Geringer Anteil gemeinsam genutzter Komponenten
- › Keine einheitliche Vision



Big Bang / Rewrite

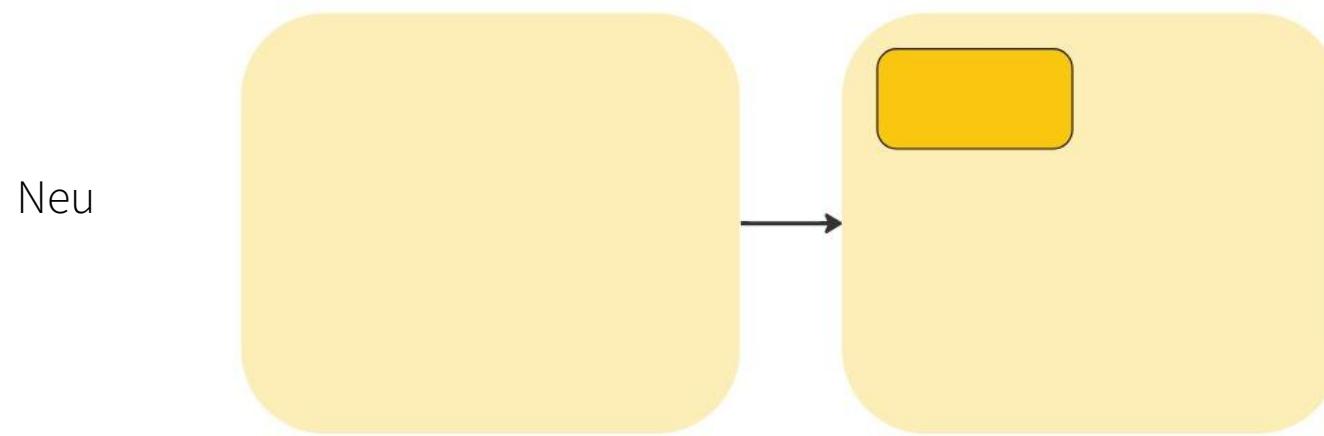
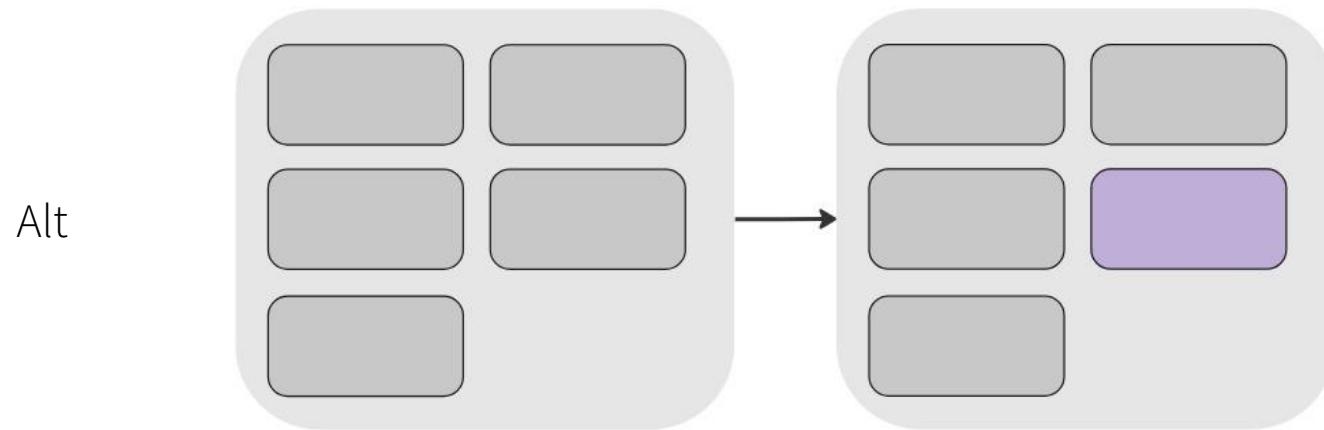
Alt



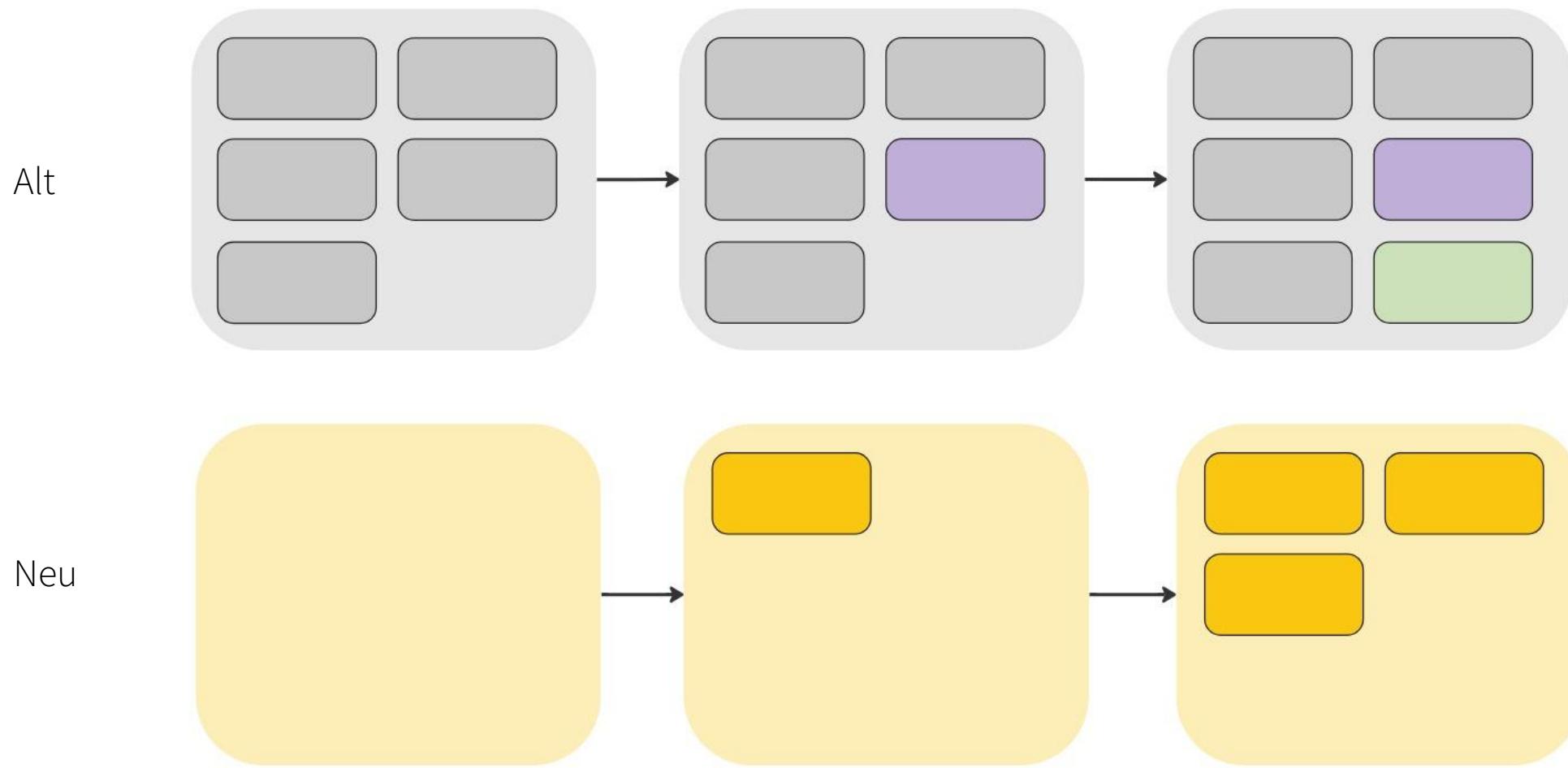
Neu



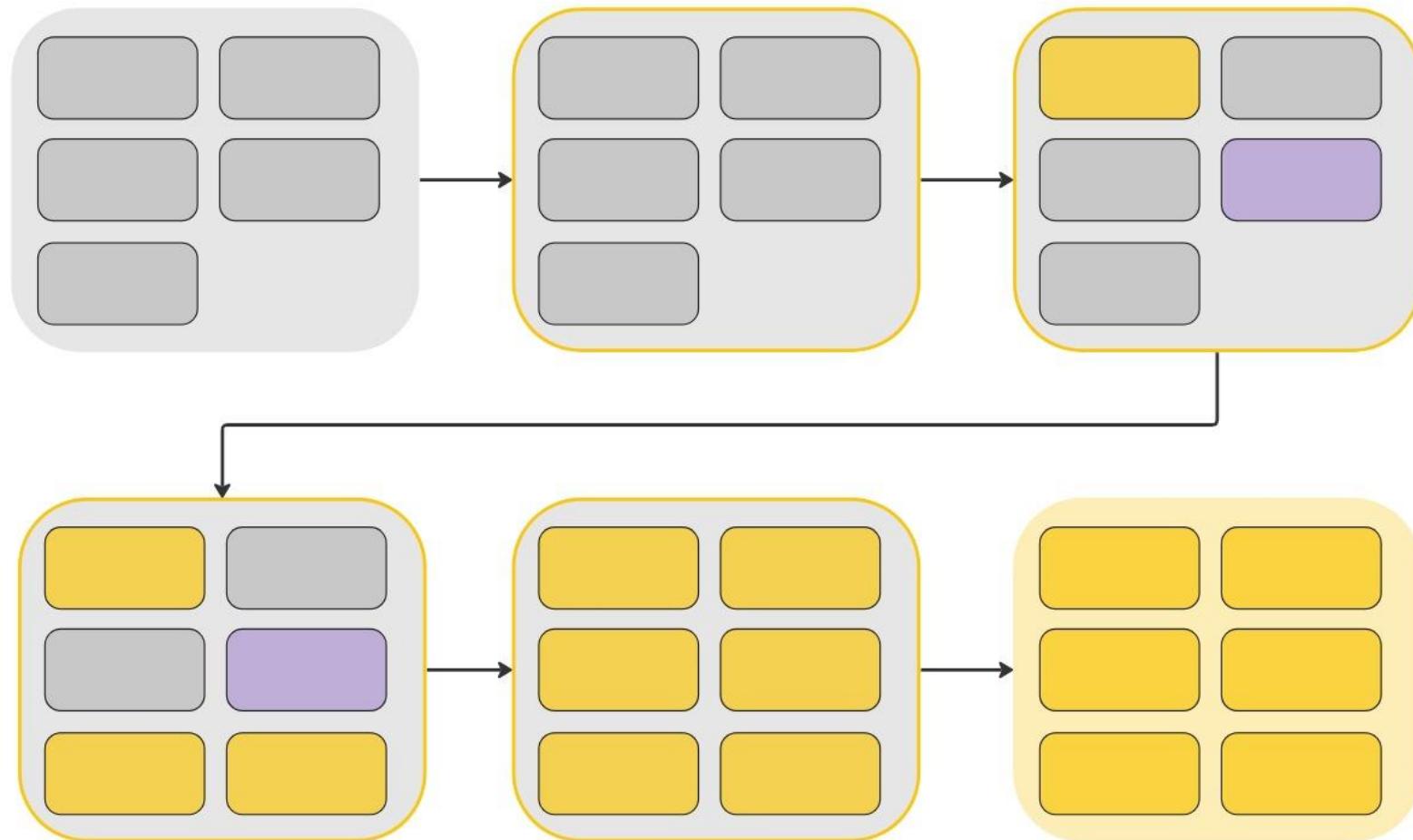
Big Bang / Rewrite



Big Bang / Rewrite



Inkrementell



Erkenntnisse nach der Modernisierung

Start simple

- › Grundlagen legen (Architektur, PoC, Vision, ...)
- › Lösungsraum erkunden

Release early

- › Feedback einholen
- › Community aufbauen
- › Lösung muss bzw. sollte noch nicht «fertig» sein

