



Rules of Play @ EPD



Oliver Wehrens

Why do we have Rules of Play ?



- E-POST started 2008 ('E2' Architecture), SOAP, PHP, C, Java
- 'E42' Architecture – REST, Java started 2011
- Business required us to deploy faster ~ 2012
- Introduction DevOps and more independent teams
- We need some rules on how Development and Operations can scale
- 'Rules of Play' ~ 2013, guided by principles like 12factor.net

Short: Our DevOps Guidelines

I ndependence

- Independence of Teams favours faster development of features. Decentralized decisions.

S ervices

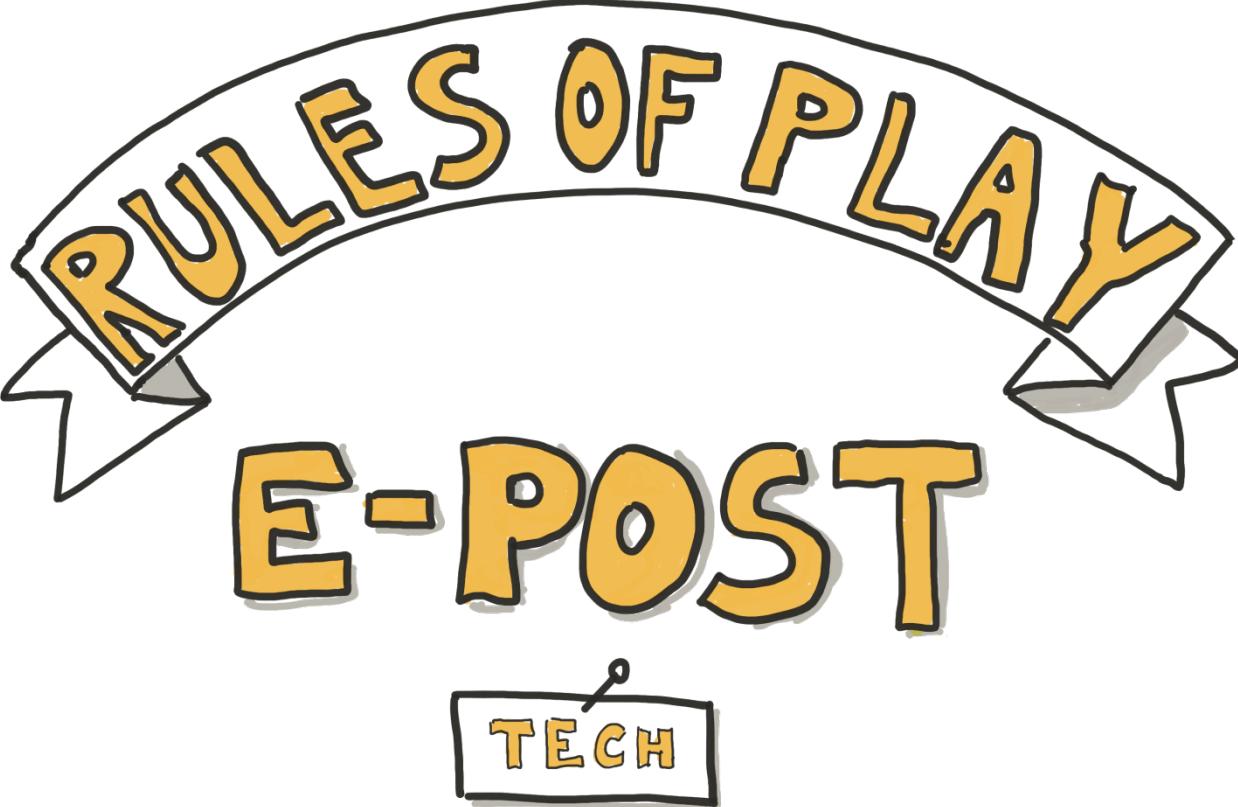
- Verticals and focus on a single business capability define a clear contract and flexibility within the service.

D evelopment

- Few but strong guidelines providing unified analysis and ‚how to run in production‘ mechanisms.

P roduction

- Unified tooling from development to production providing much faster feedback on potential errors in later stages.

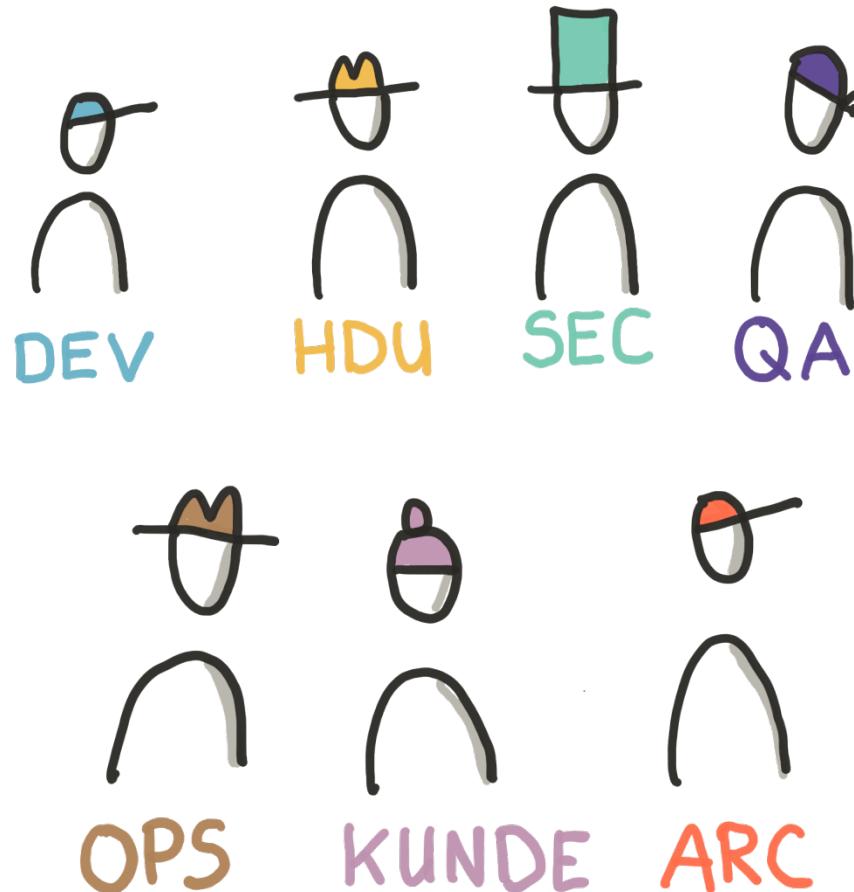


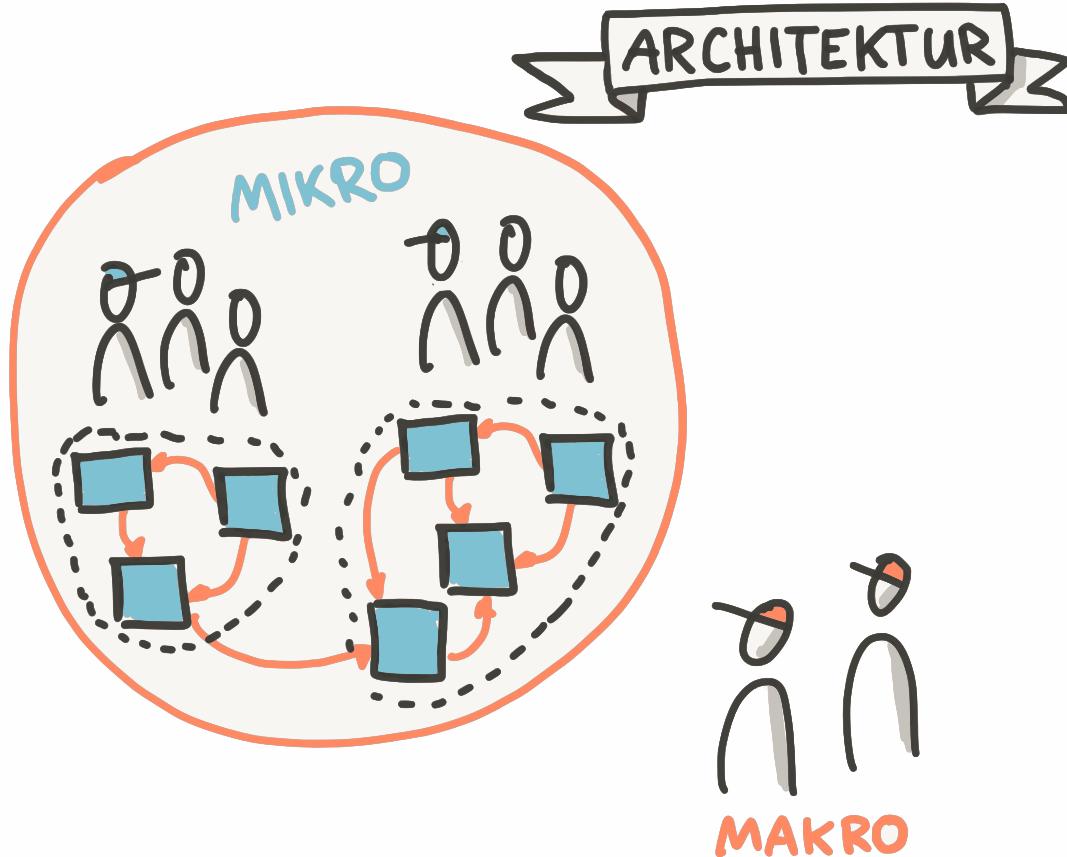
RULES OF PLAY

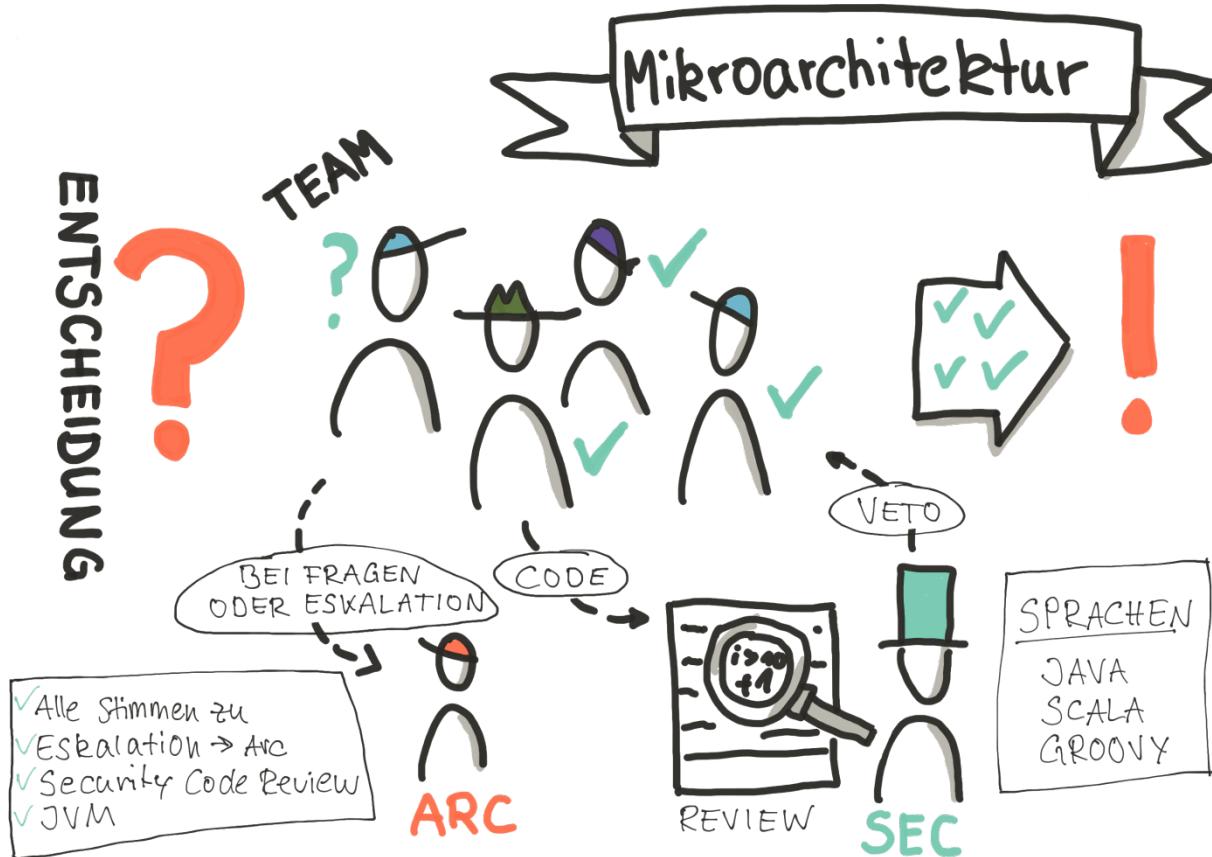
E-POST

TECH

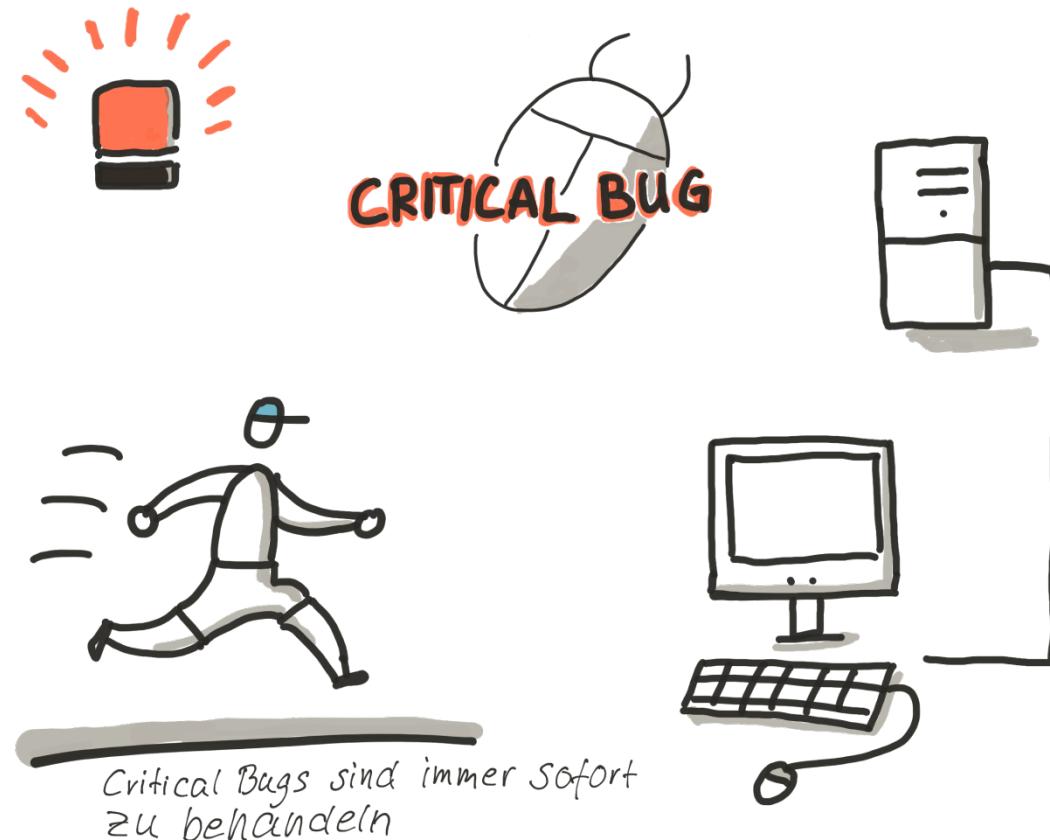
Stakeholder



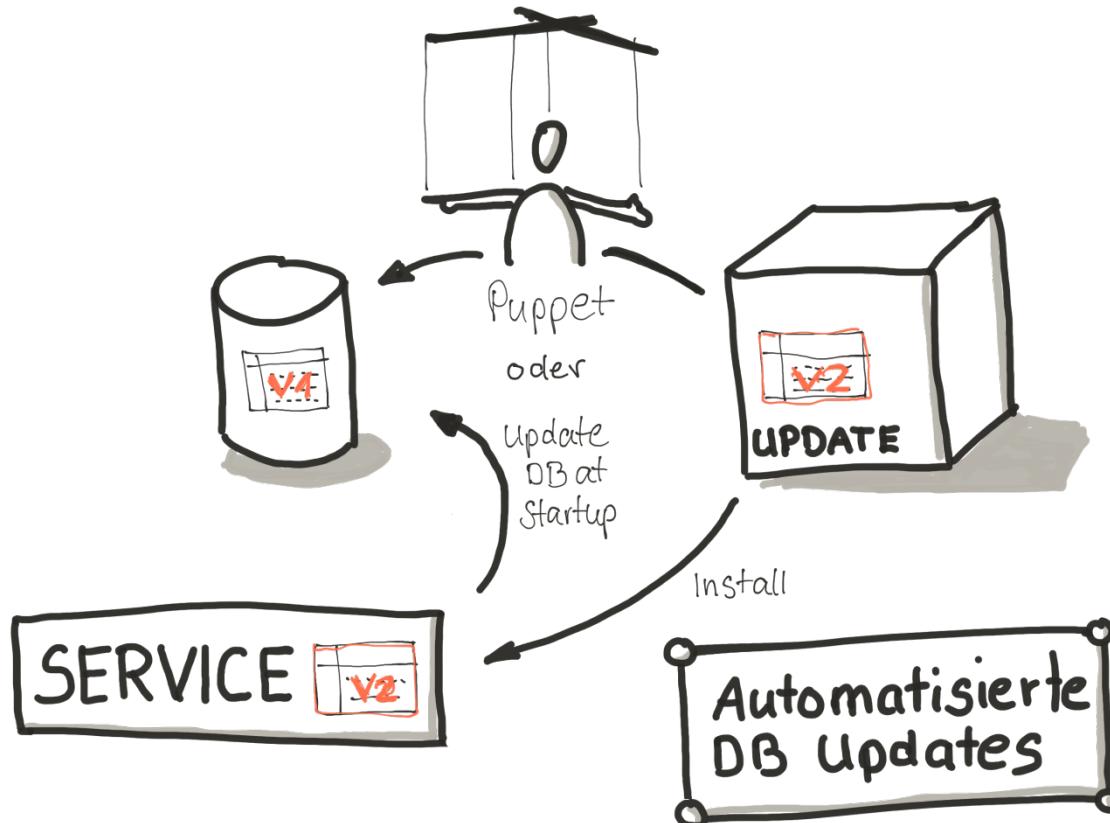




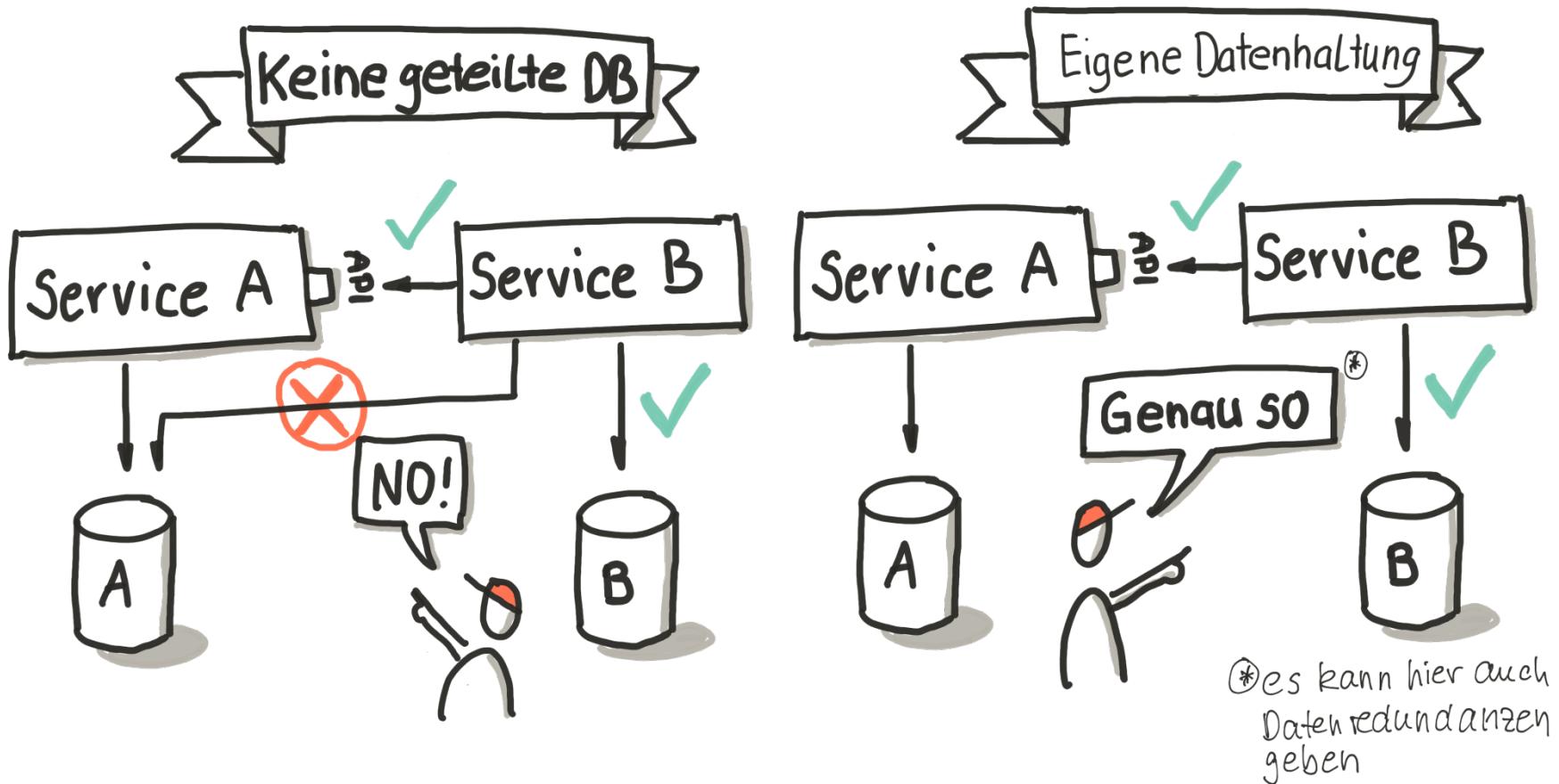
Every team needs to be able to fix a critical bug immediately.

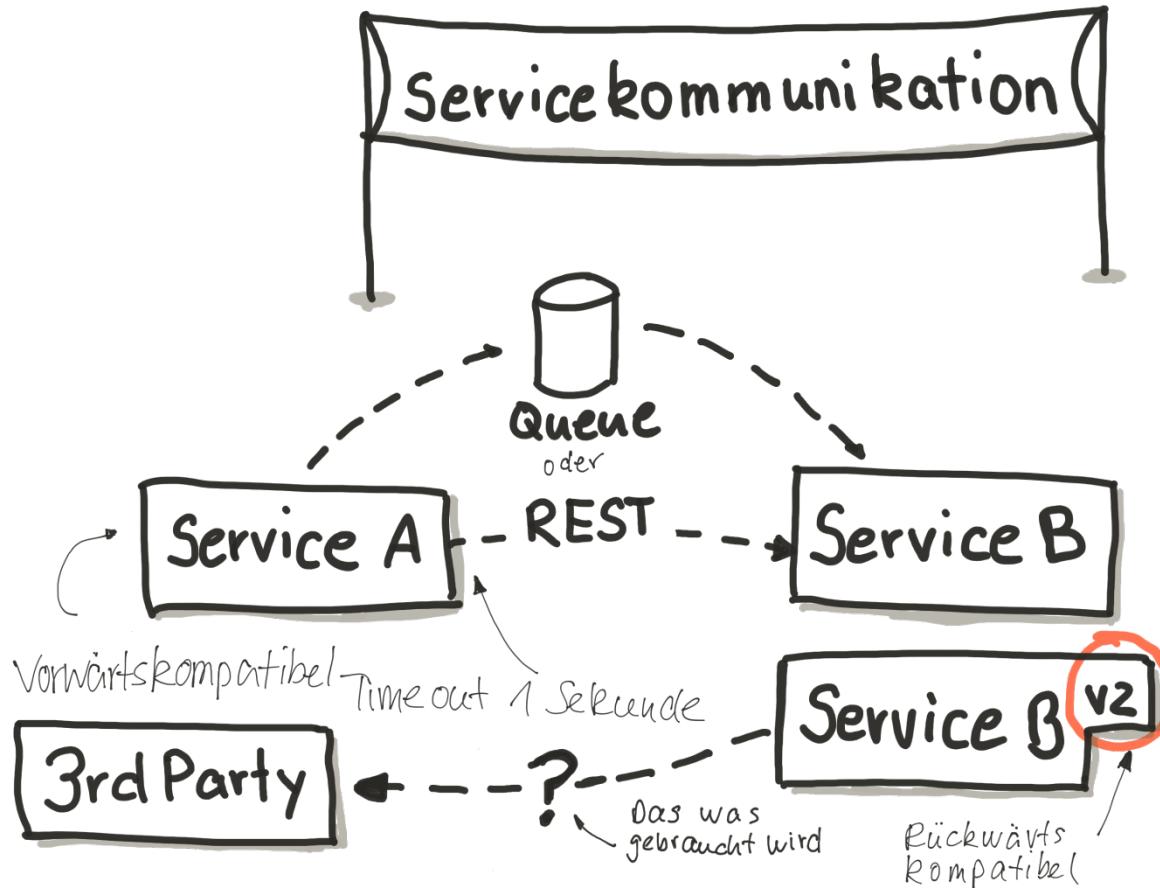


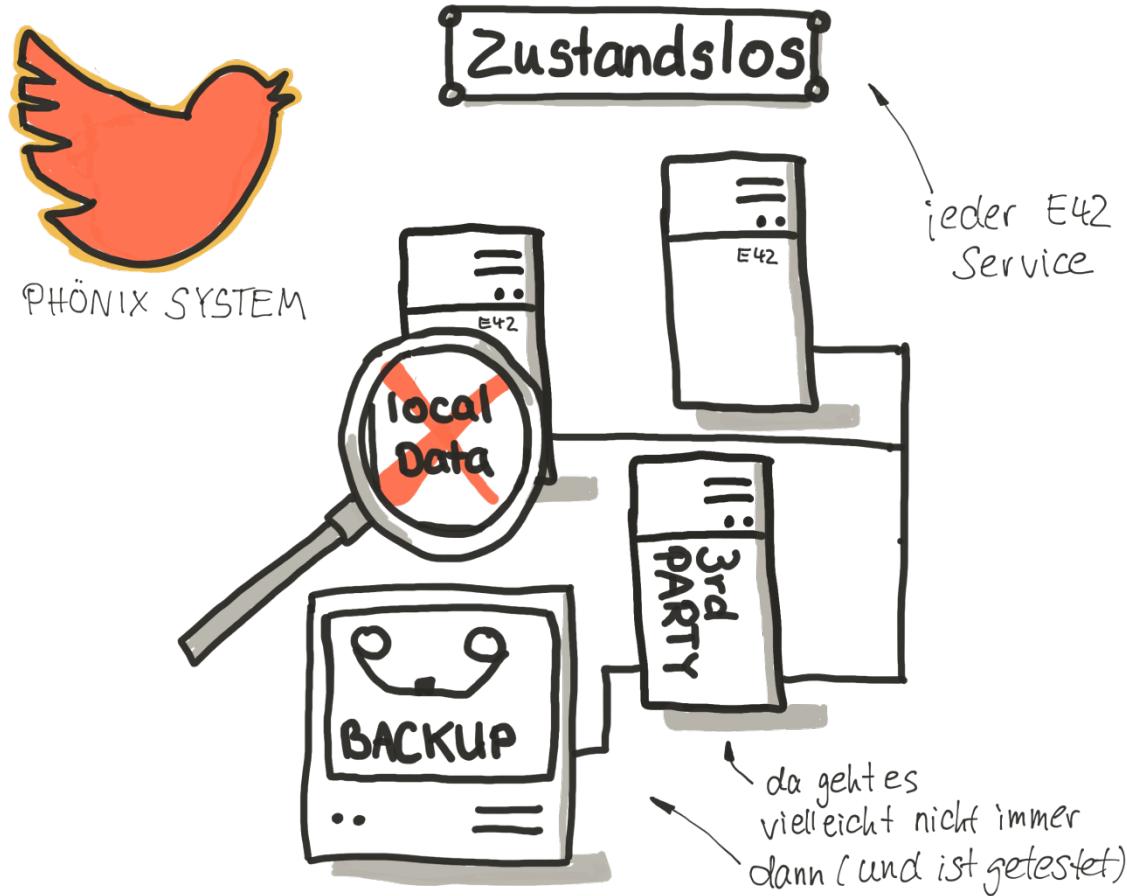
Automated DB Updates



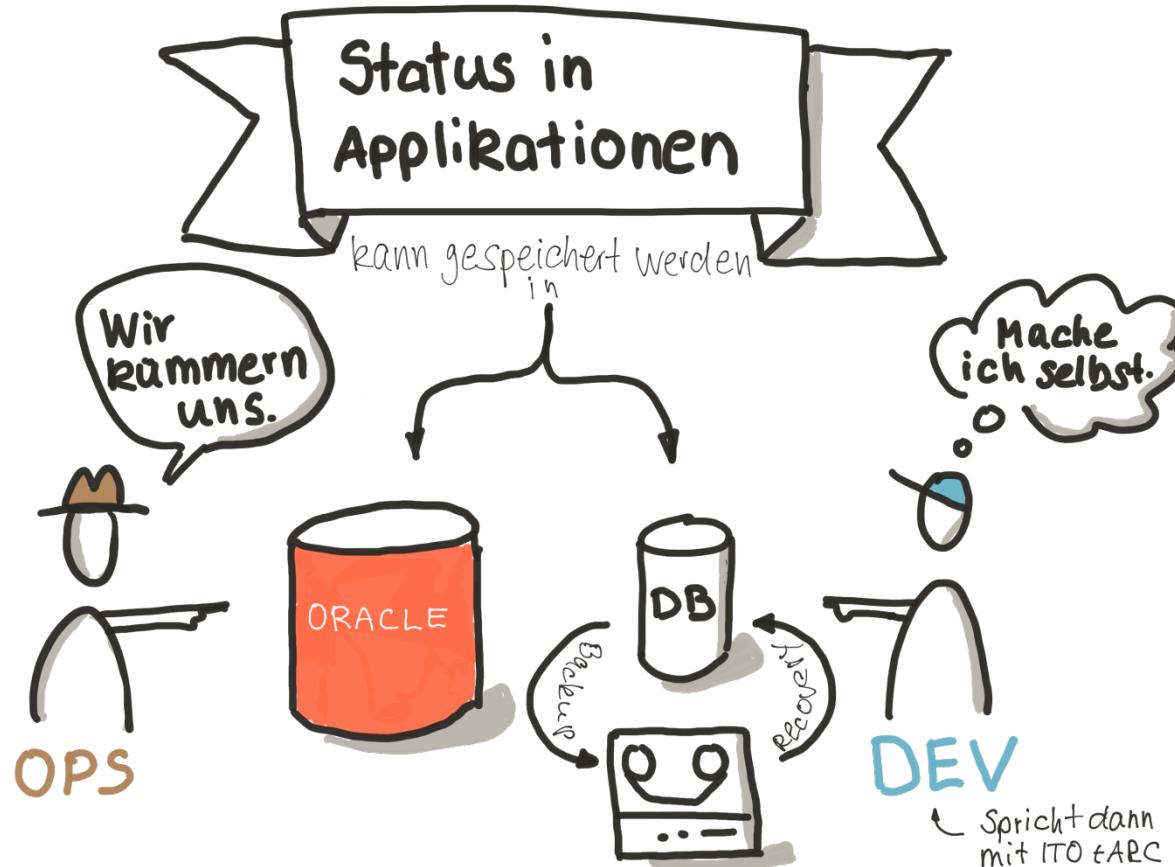
No shared Data(base) between services

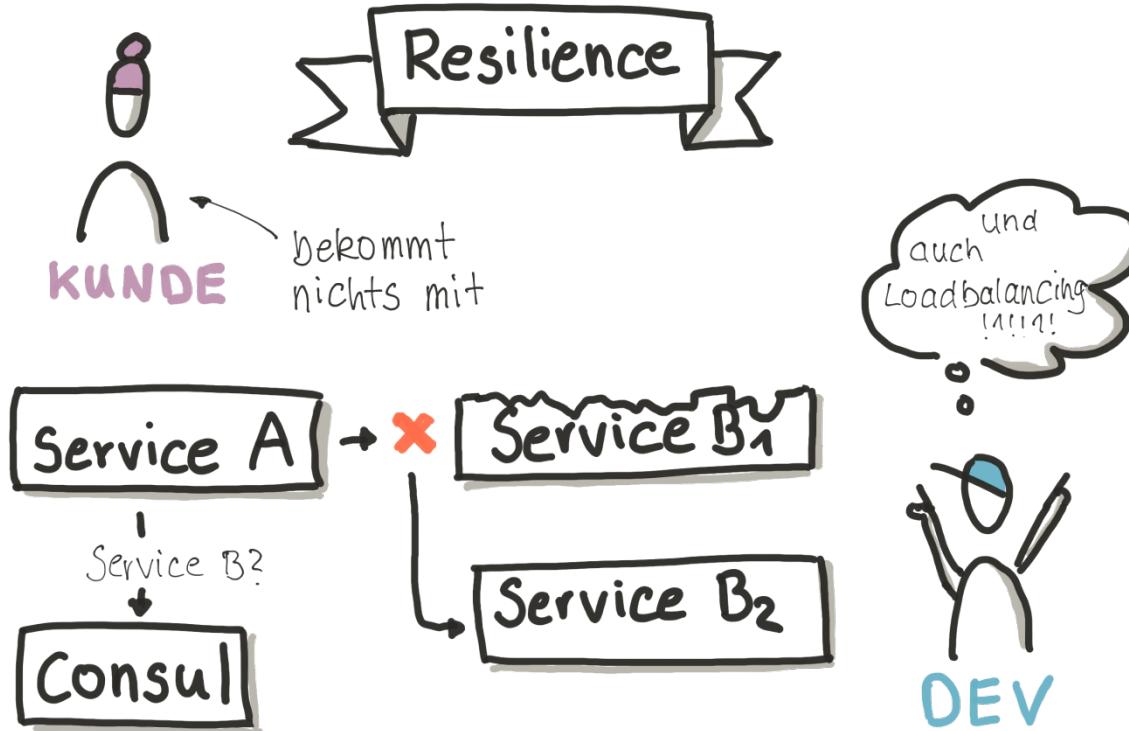


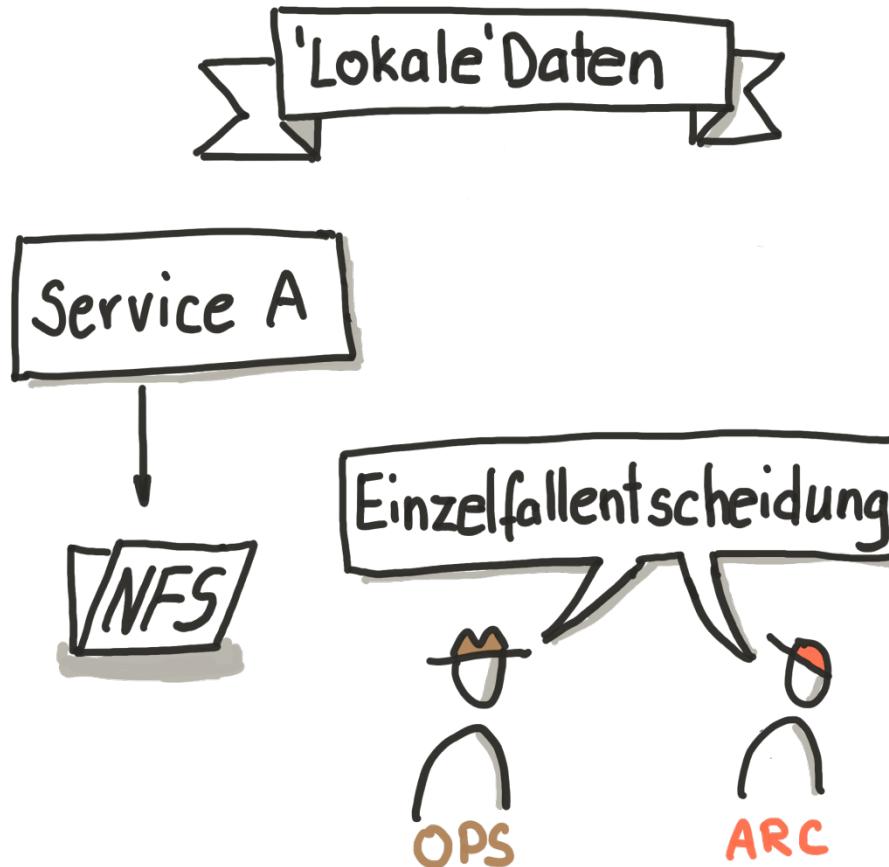


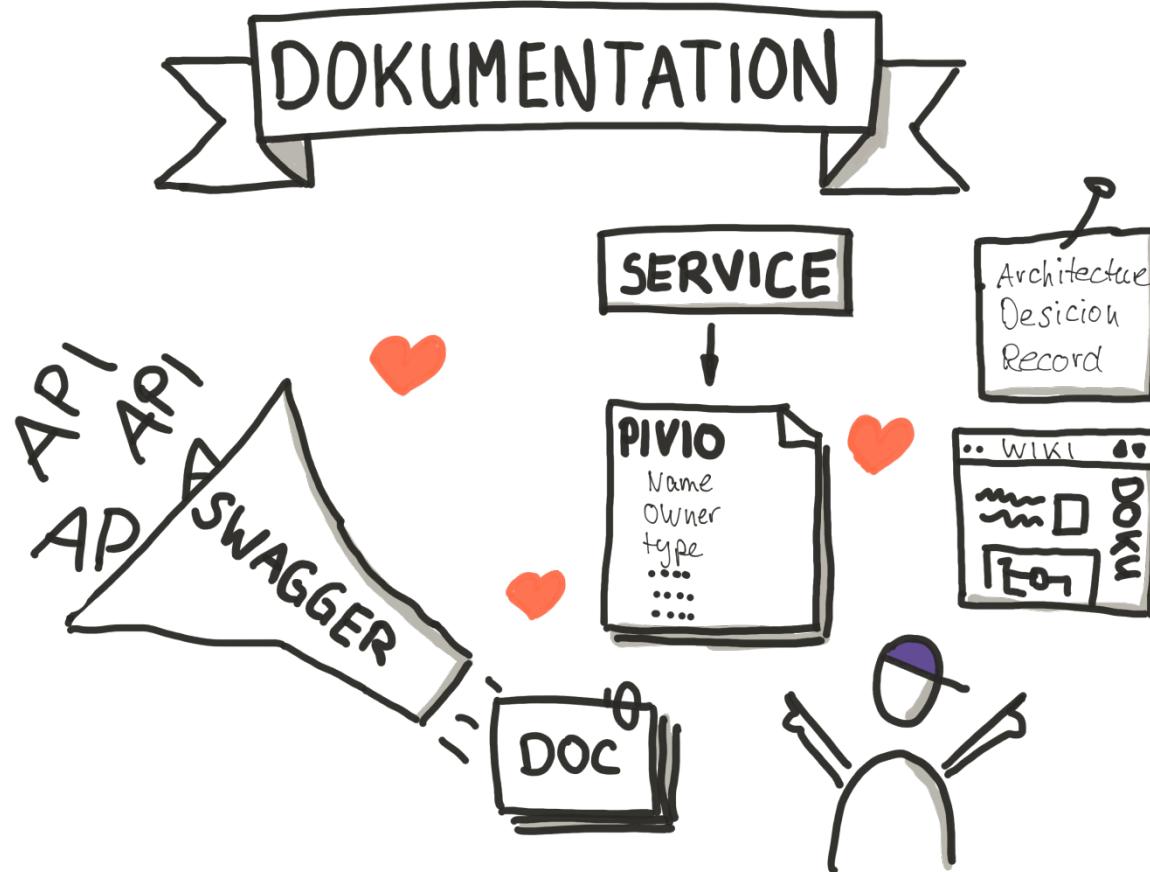


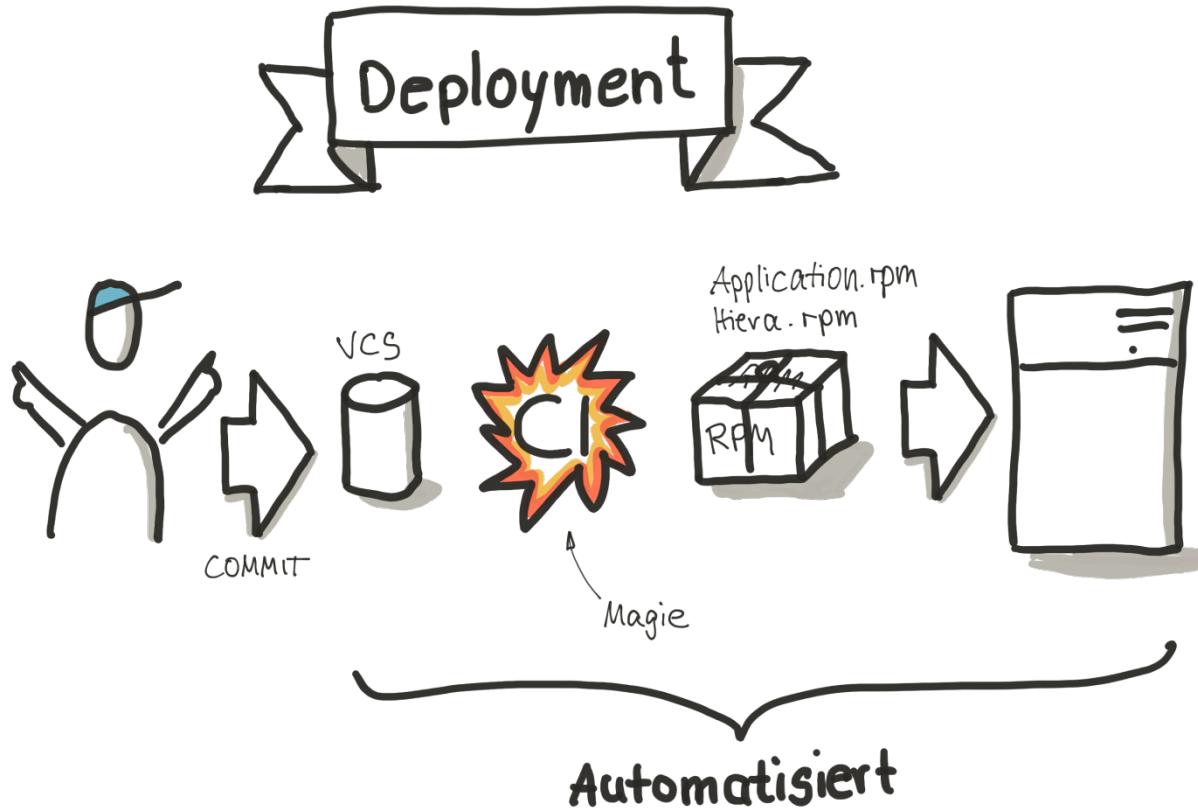
State lives only on databases



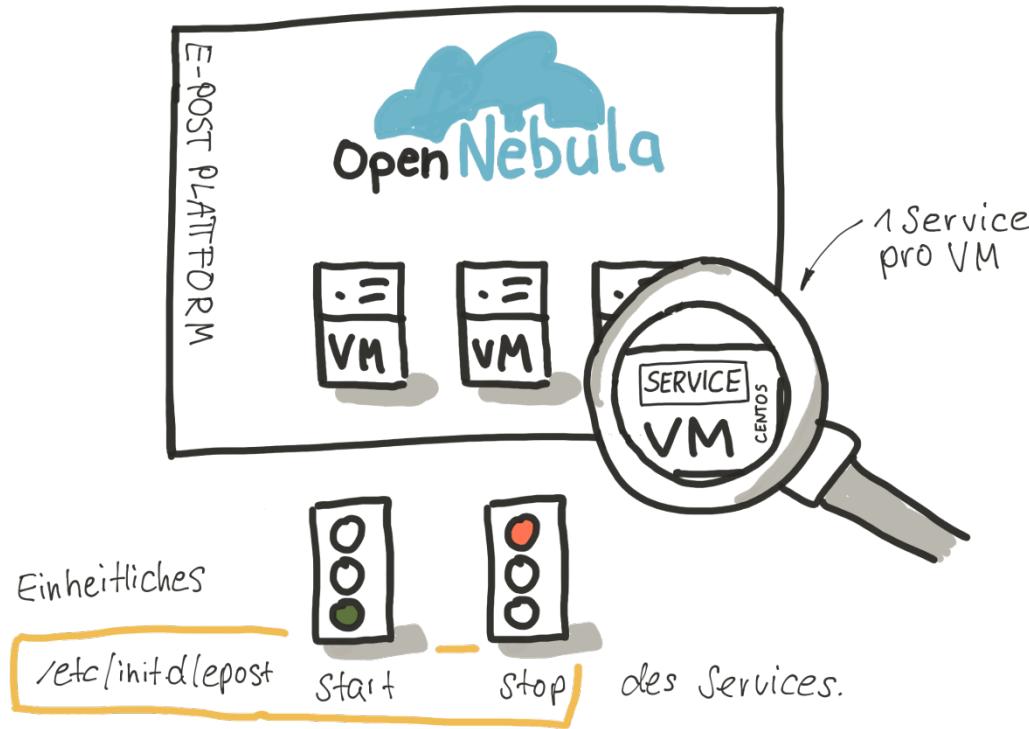




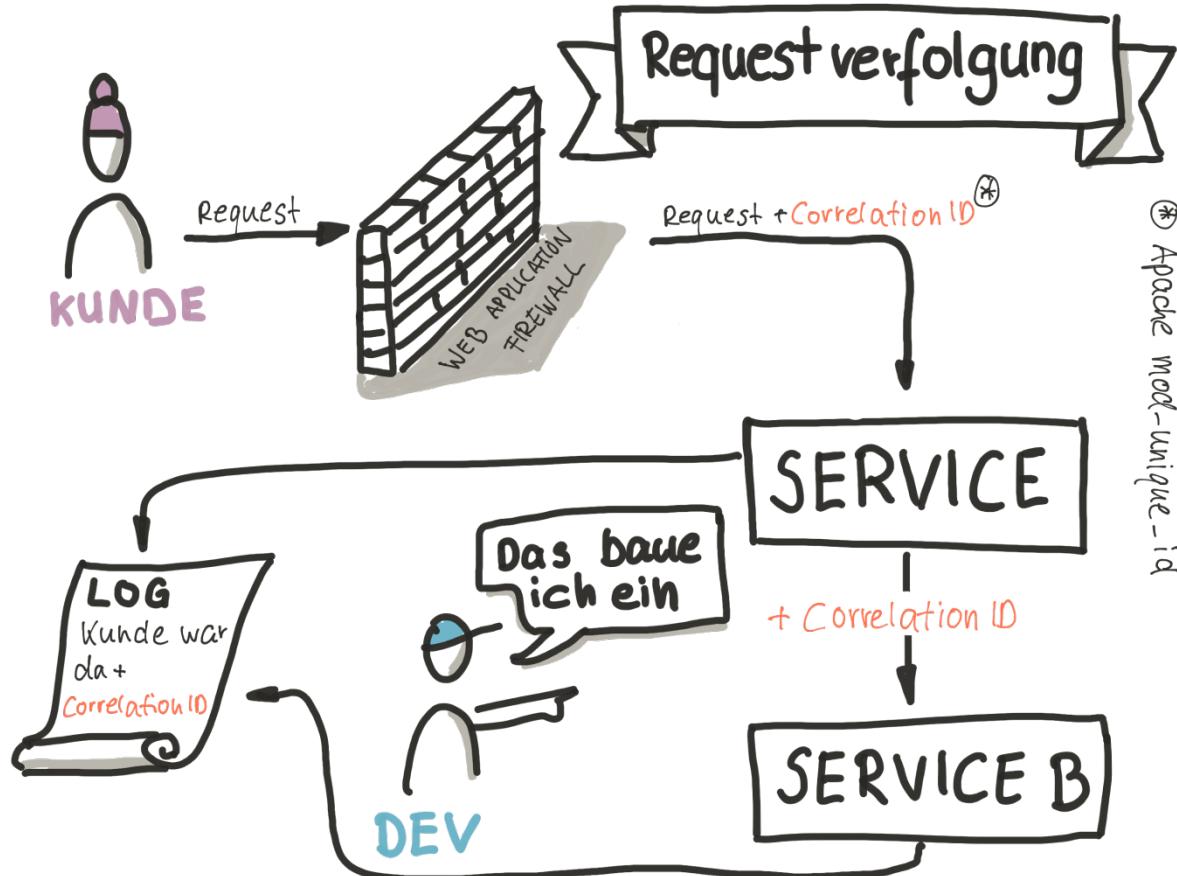


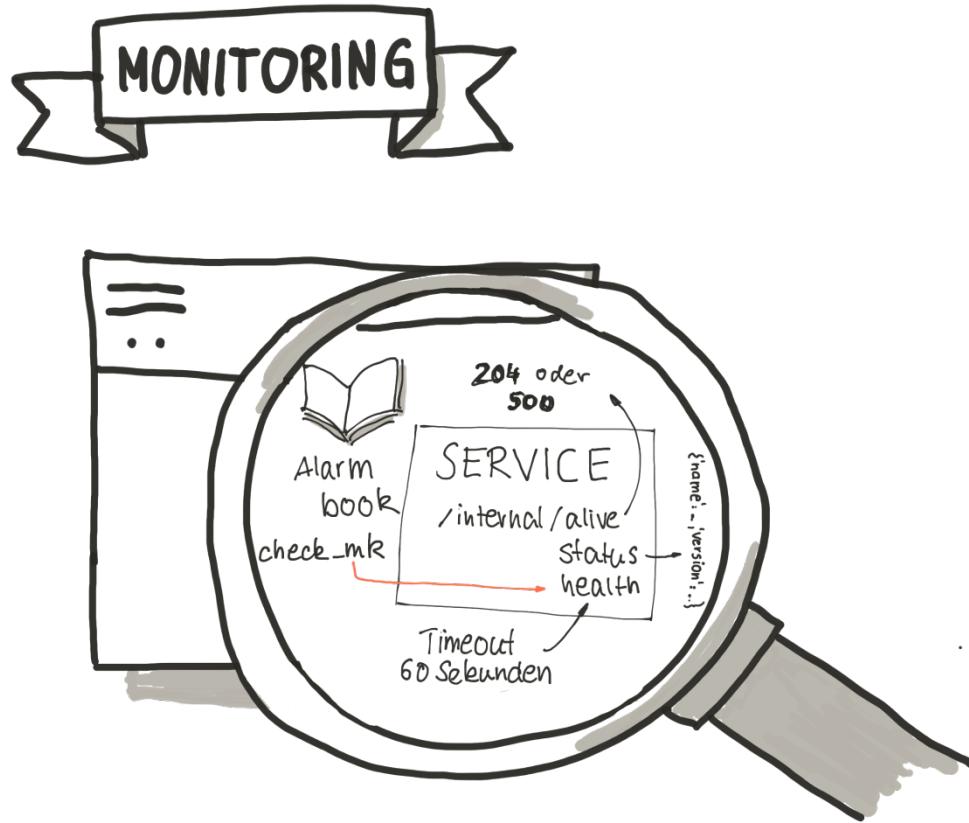


Deploying on E-POST Platform



Request Tracking





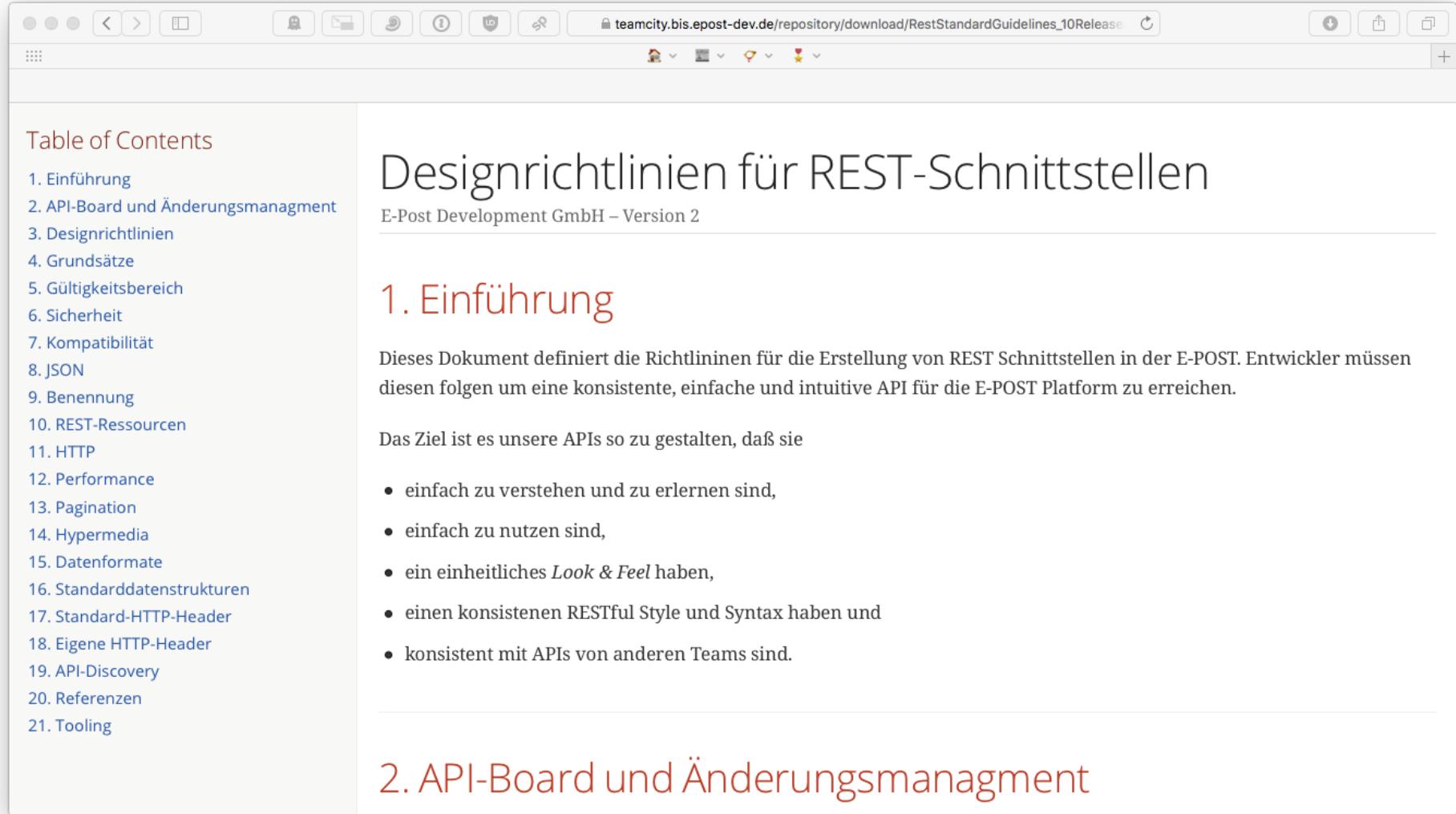
Questions



REST API Backwards Compatibility



- 'Good' REST Design and principles at start
- Once defined:
 - Never remove
 - Never change
 - Only 'optional' add
- Clients need to be 'forward' compatible, meaning ignoring additional values they do not understand
- Two types of APIs
 - Internal
 - External
- Internal:
 - Depending on usage we can introduce breaking changes
 - Talking to all teams which are using it (check 'pivio') and prioritizing it in their sprints
- External:
 - Never change
- To foster same understanding of our APIs we introduced last year an API Design Guide



The screenshot shows a web browser window displaying the "Designrichtlinien für REST-Schnittstellen" document. The browser's address bar shows the URL: teamcity.bis.epost-dev.de/repository/download/RestStandardGuidelines_10Release. The page title is "Designrichtlinien für REST-Schnittstellen" and it is noted as "E-Post Development GmbH – Version 2".

Table of Contents

- 1. Einführung
- 2. API-Board und Änderungsmanagement
- 3. Designrichtlinien
- 4. Grundsätze
- 5. Gültigkeitsbereich
- 6. Sicherheit
- 7. Kompatibilität
- 8. JSON
- 9. Benennung
- 10. REST-Ressourcen
- 11. HTTP
- 12. Performance
- 13. Pagination
- 14. Hypermedia
- 15. Datenformate
- 16. Standarddatenstrukturen
- 17. Standard-HTTP-Header
- 18. Eigene HTTP-Header
- 19. API-Discovery
- 20. Referenzen
- 21. Tooling

1. Einführung

Dieses Dokument definiert die Richtlinien für die Erstellung von REST Schnittstellen in der E-POST. Entwickler müssen diesen folgen um eine konsistente, einfache und intuitive API für die E-POST Platform zu erreichen.

Das Ziel ist es unsere APIs so zu gestalten, daß sie

- einfach zu verstehen und zu erlernen sind,
- einfach zu nutzen sind,
- ein einheitliches *Look & Feel* haben,
- einen konsistenten RESTful Style und Syntax haben und
- konsistent mit APIs von anderen Teams sind.

2. API-Board und Änderungsmanagement

Questions

