IT Ortenau

Breaking the monolith

Learnings from breaking the monolith into Microservices

Offenburg, 28.06.2023

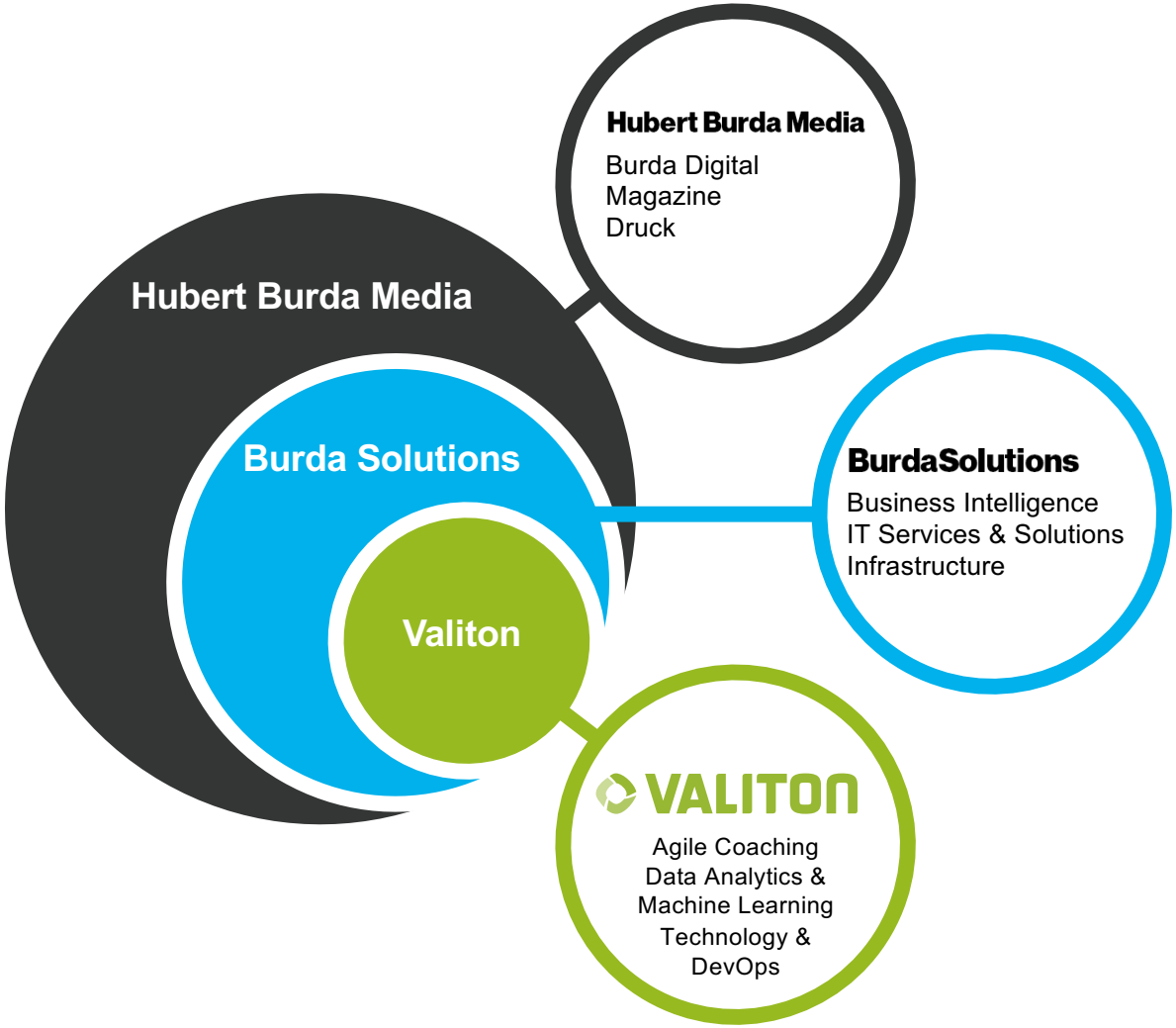
VALITON

Ein BurdaSolutions Unternehmen

AGENDA

1. Valiton in a Nutshell
2. Monolith vs Microservices Software Architecture
3. Challenges of a Microservice Architecture
4. Breaking the monolith (Pitfalls & Learnings)
5. Summary (Monolith frist)

1 Valiton in a Nutshell



Hubert Burda Media

Die TECH und MEDIA COMPANY


meine Familie & ich
XING
cyberport
ELLE
CHIP
Burda...
HolidayCheck
BUNTE
NetDoktor
DLD
BAMBI
netmoms
nebenan.de
Vinted
FREIZEIT REVUE
mein schöner Garten
EFAHRER
IMMEDIATE MEDIA Co
...in good company.
FOCUS
Finanzen 100
VALITON
InStyle
freundin

Valiton @ your service

Machine Learning & AI

Software Development
Data Engineering

Data Analytics
Data Warehouse
Conversion Optimierung

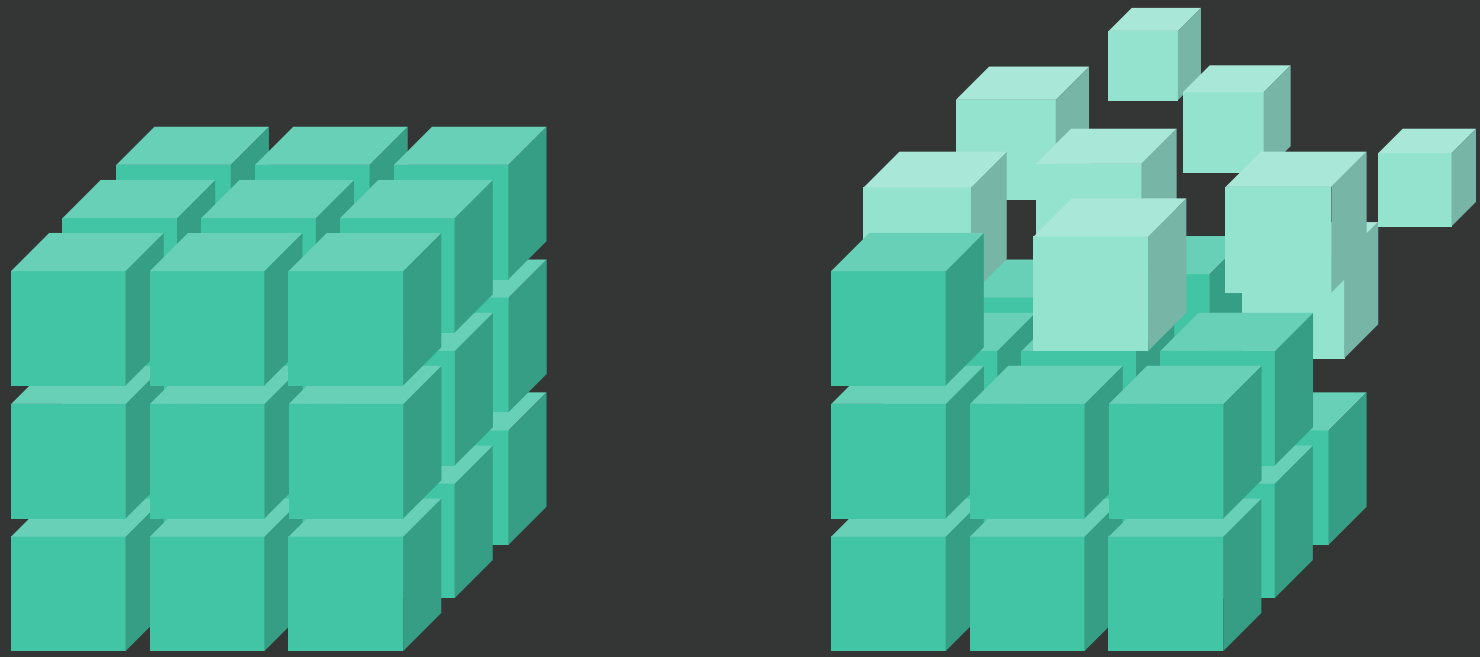
DevOps
Infrastructure as Code (IaC)

Agile Coaching
Project Management

Valiton key facts

- 55 employees
- Our offices:
  - Offenburg
  - Munich
  - Berlin
  - (Hamburg)

BREAKING THE MONOLITH...

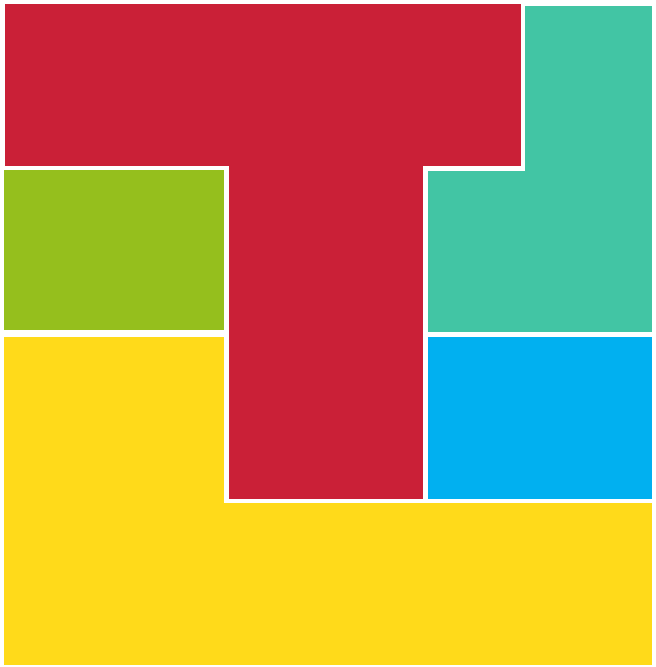


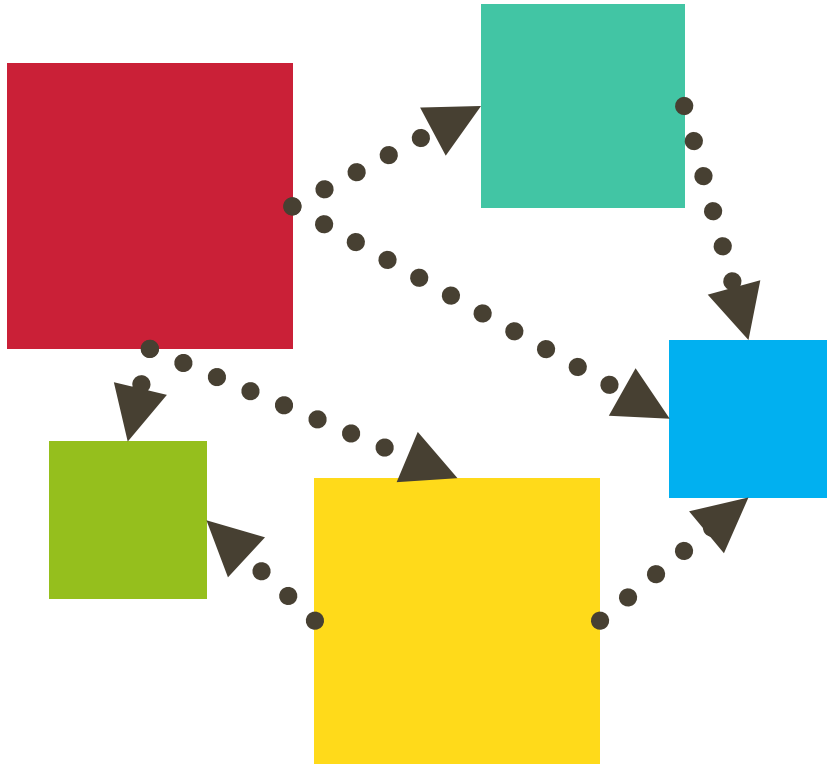
2

Monolith vs Microservices

- Software Architecture

Monolith vs Microservices





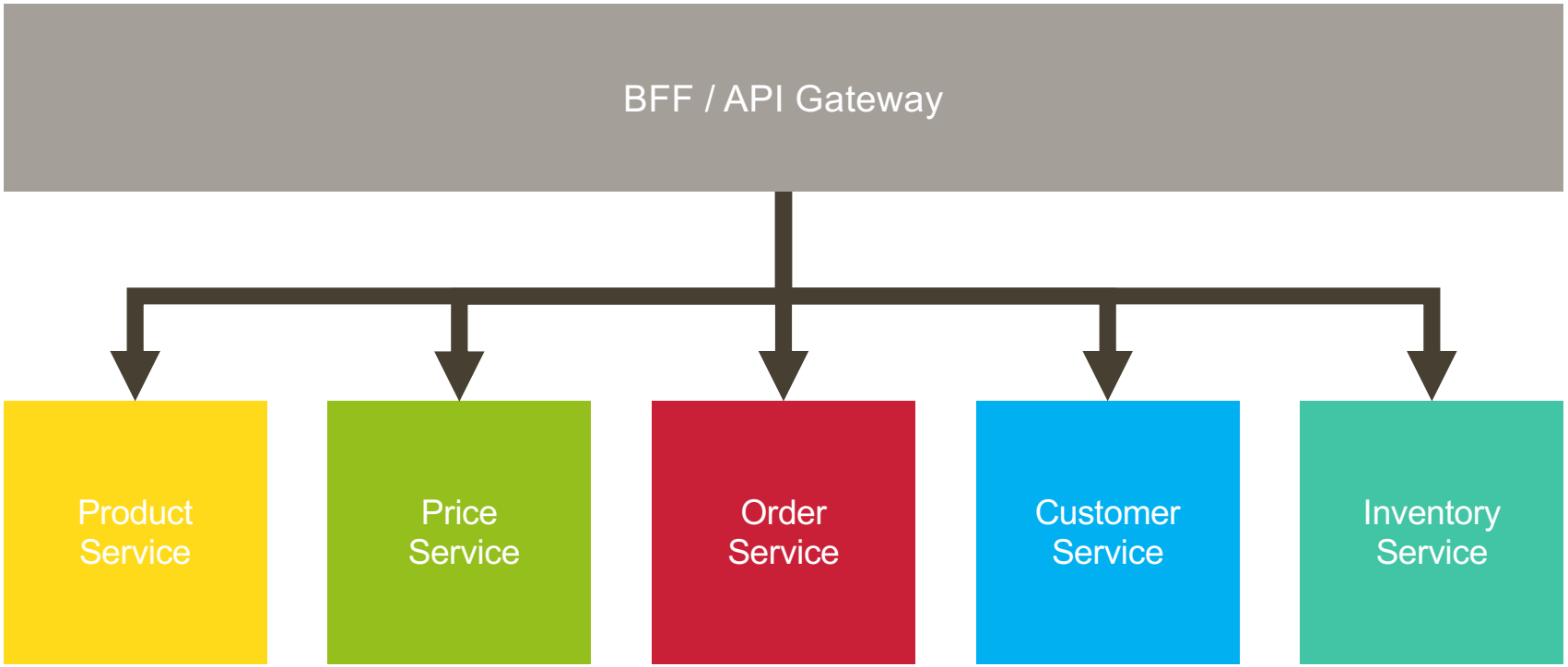
Microservices Pro & Cons

- module boundaries are strictly separated by services
- independent changes and updates (better maintainable/expandable)
- individual horizontal and vertical scaling
- individual tech-stack per service

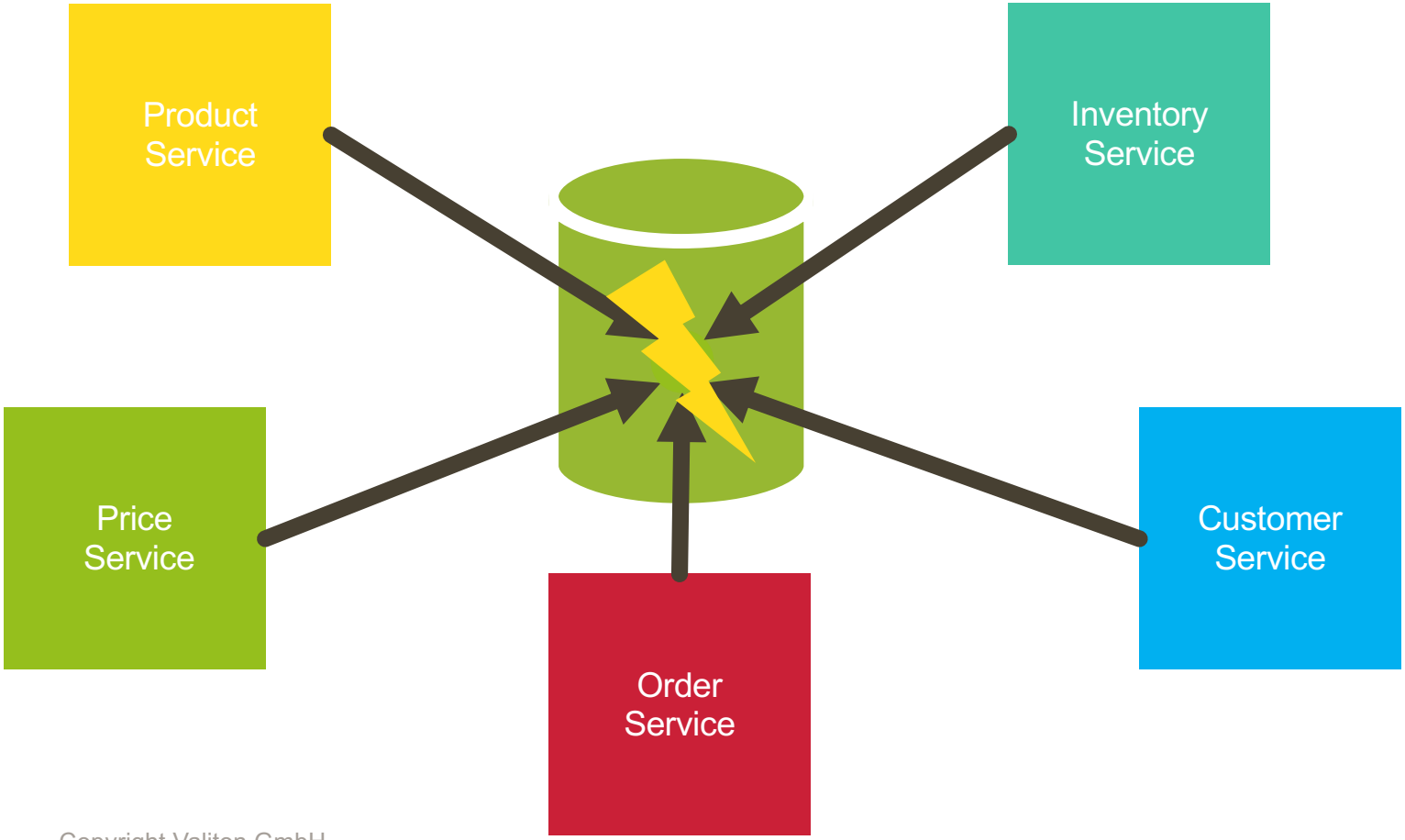
- higher complexity due to many services
- higher costs for operation, deployment and monitoring
- enormous migration effort
- higher latency (network, messaging)
- consistency (transactions)

3 Challenges of a Microservice Architecture

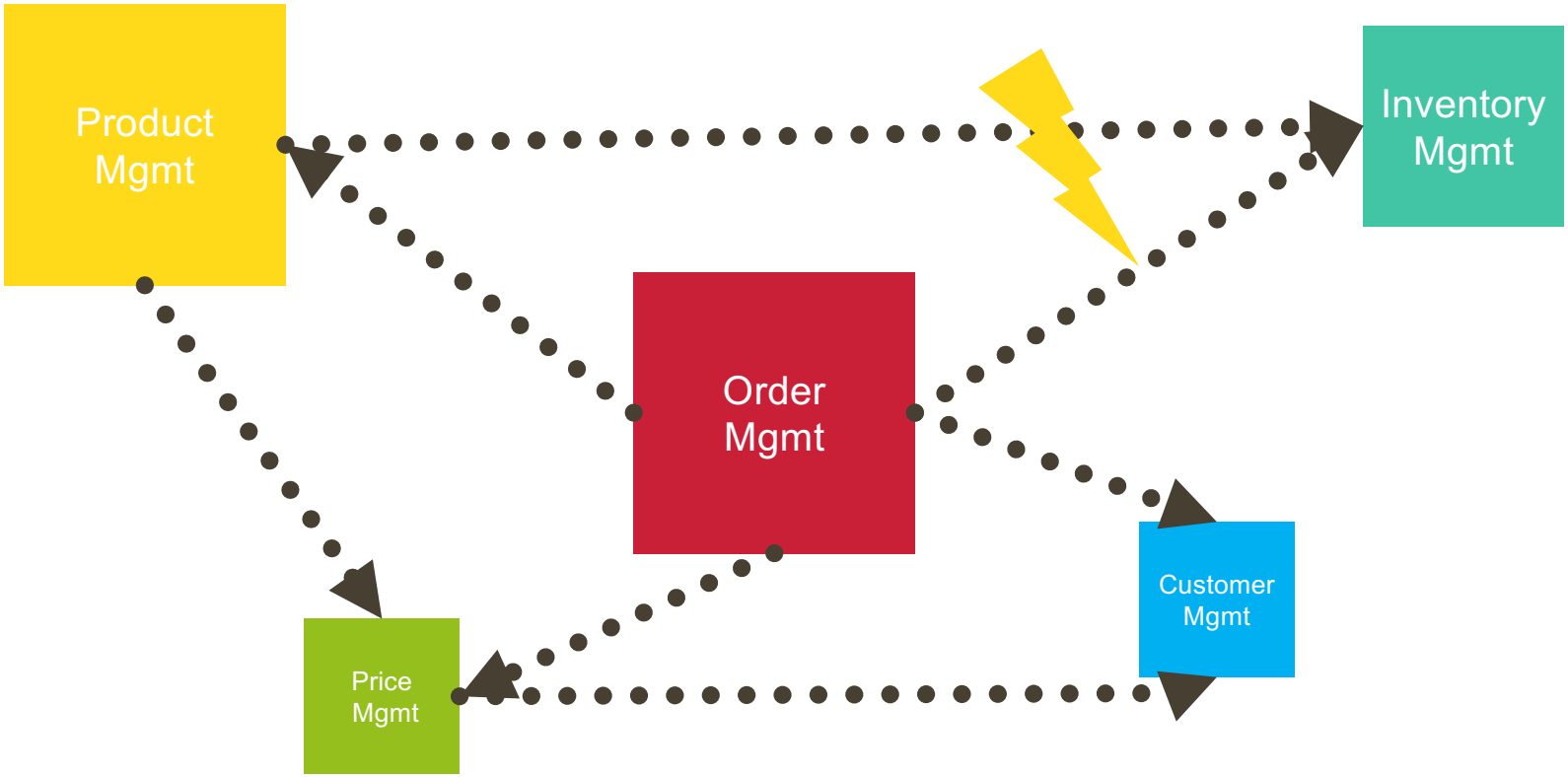
Orchestrierung - Backend for Frontend (BFF) / API Gateway



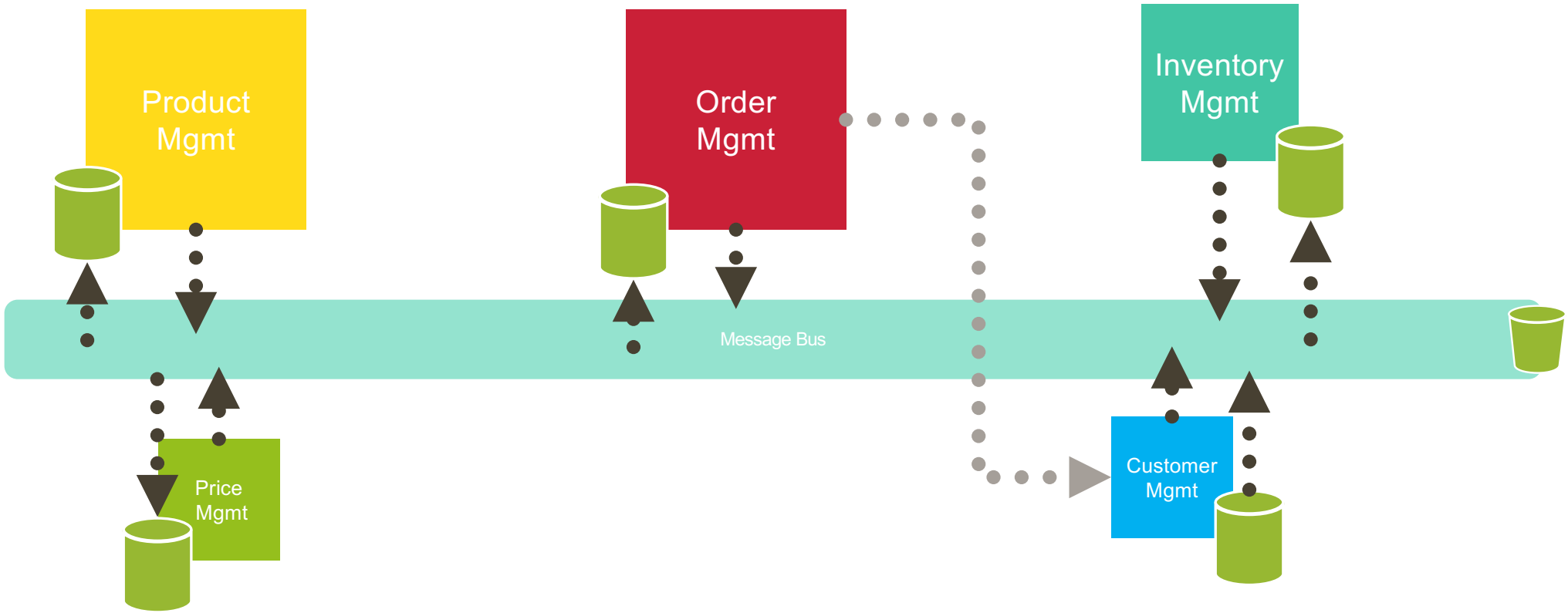
Don't share Databases or Schemas



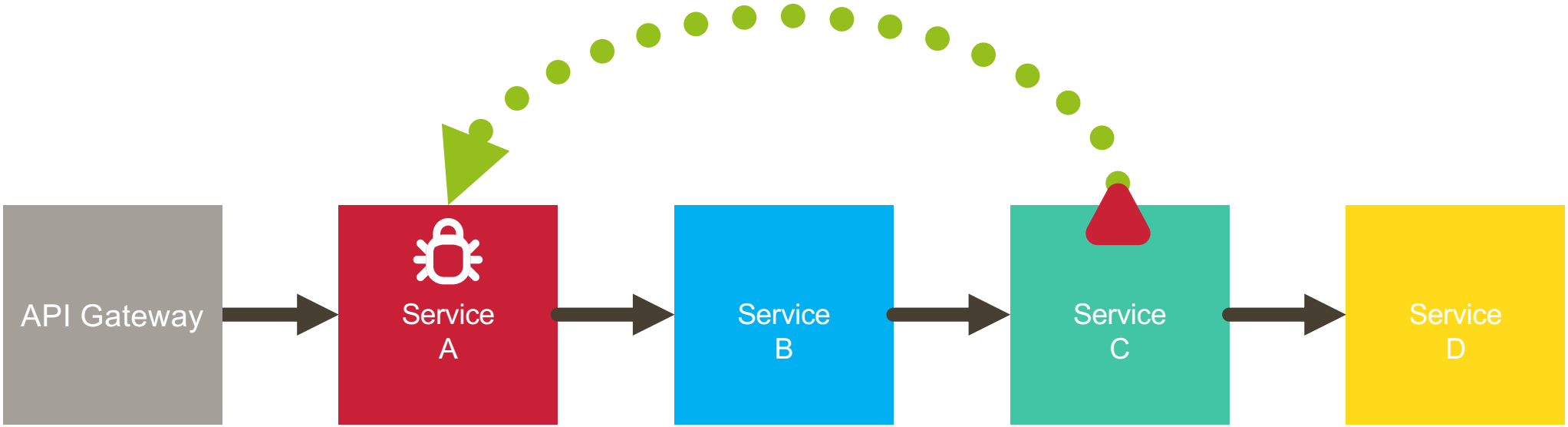
Avoid sync. communication - Distributed Monolith



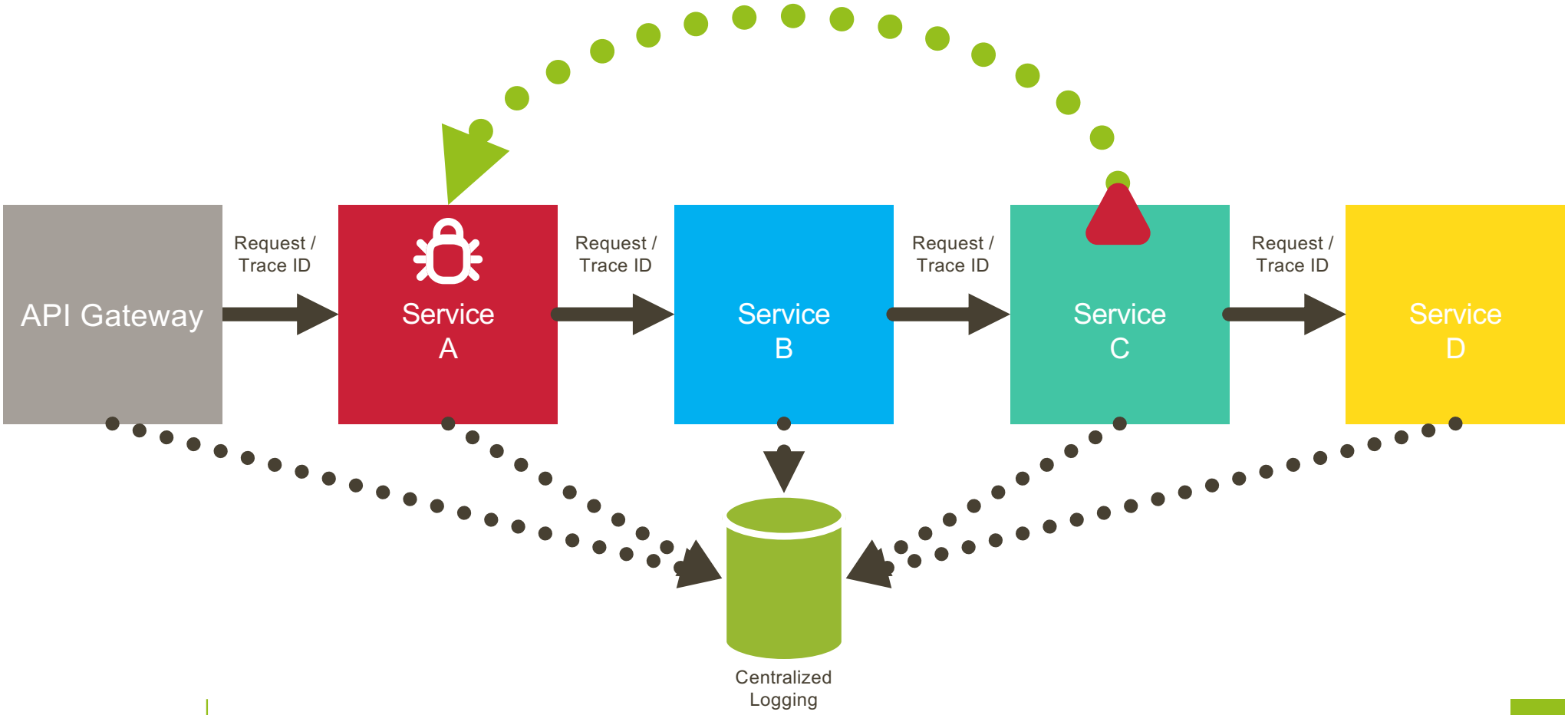
Loosely coupled services



Observability



Distributed Tracing & Centralized Logging

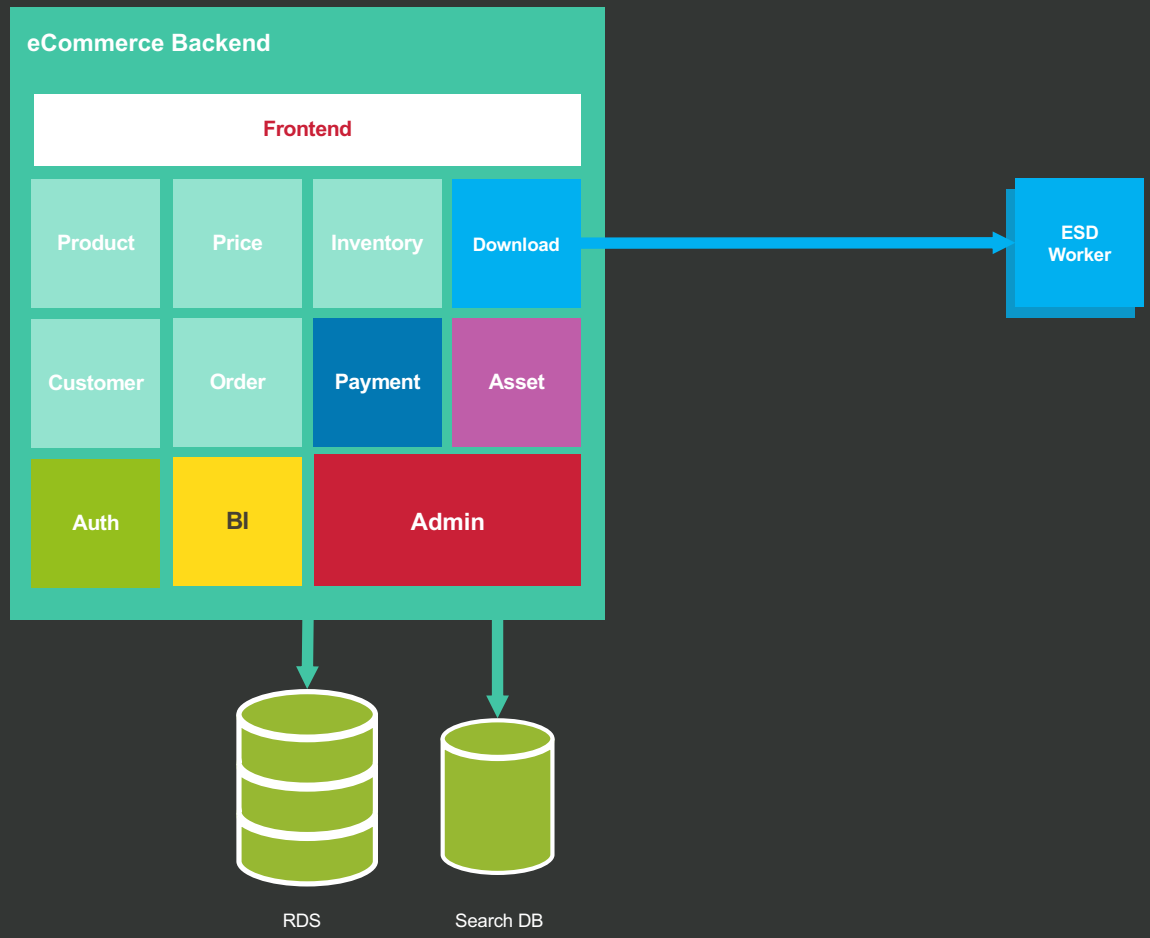


4 Breaking the monolith

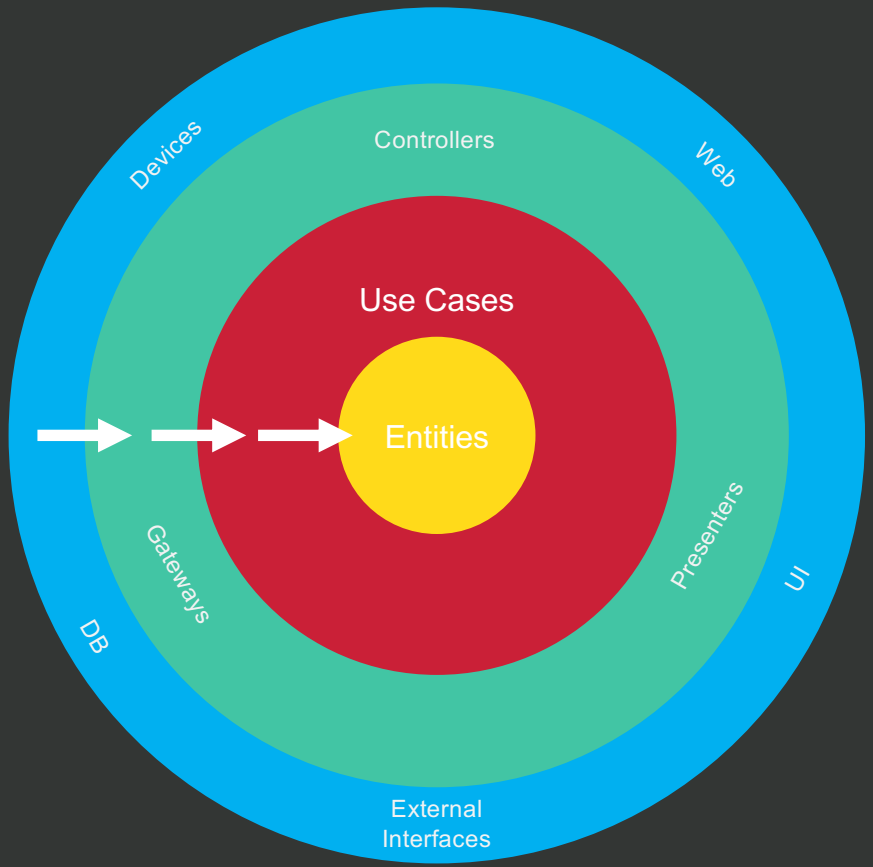
- our approach to break the monolith
- Pitfalls & Learnings

Example: Monolith

Before



Clean Architecture



- Enterprise Business Rules
- Application Business Rules
- Interface Adapters
- Framework & Drivers

Source: Robert C. Martin, Clean Architecture, Chapter 22 Figure 22.1
https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html

SOLID Principles

Single-responsibility principle

Open–closed principle

Liskov substitution principle

Interface segregation principle

Dependency inversion principle

Clean Architecture

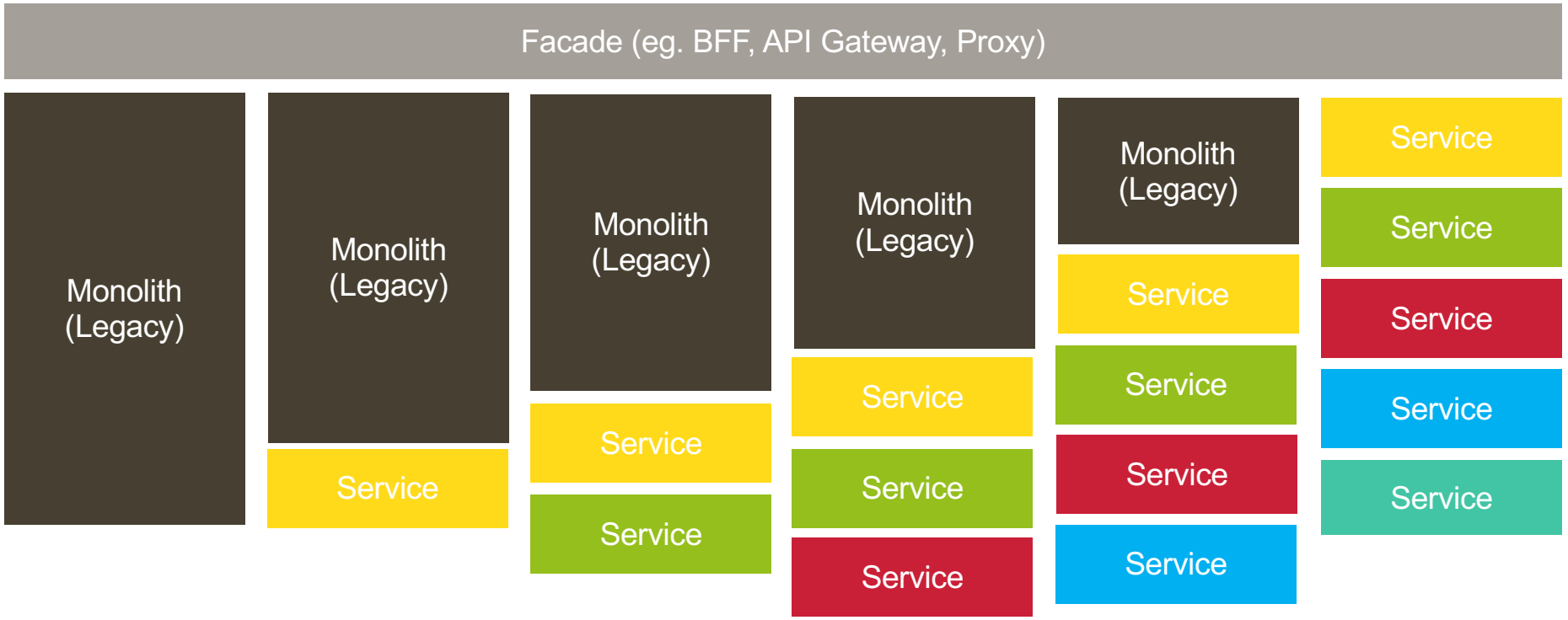
« A good architecture will allow a system to be born as a monolith, deployed in a single file, but then to grow into a set of independently deployable units, and then all the way to independent services and/or micro-services. »

Robert C. Martin, Book "Clean Architecture"

source: https://martinfowler.com/bliki/MonolithFirst.html

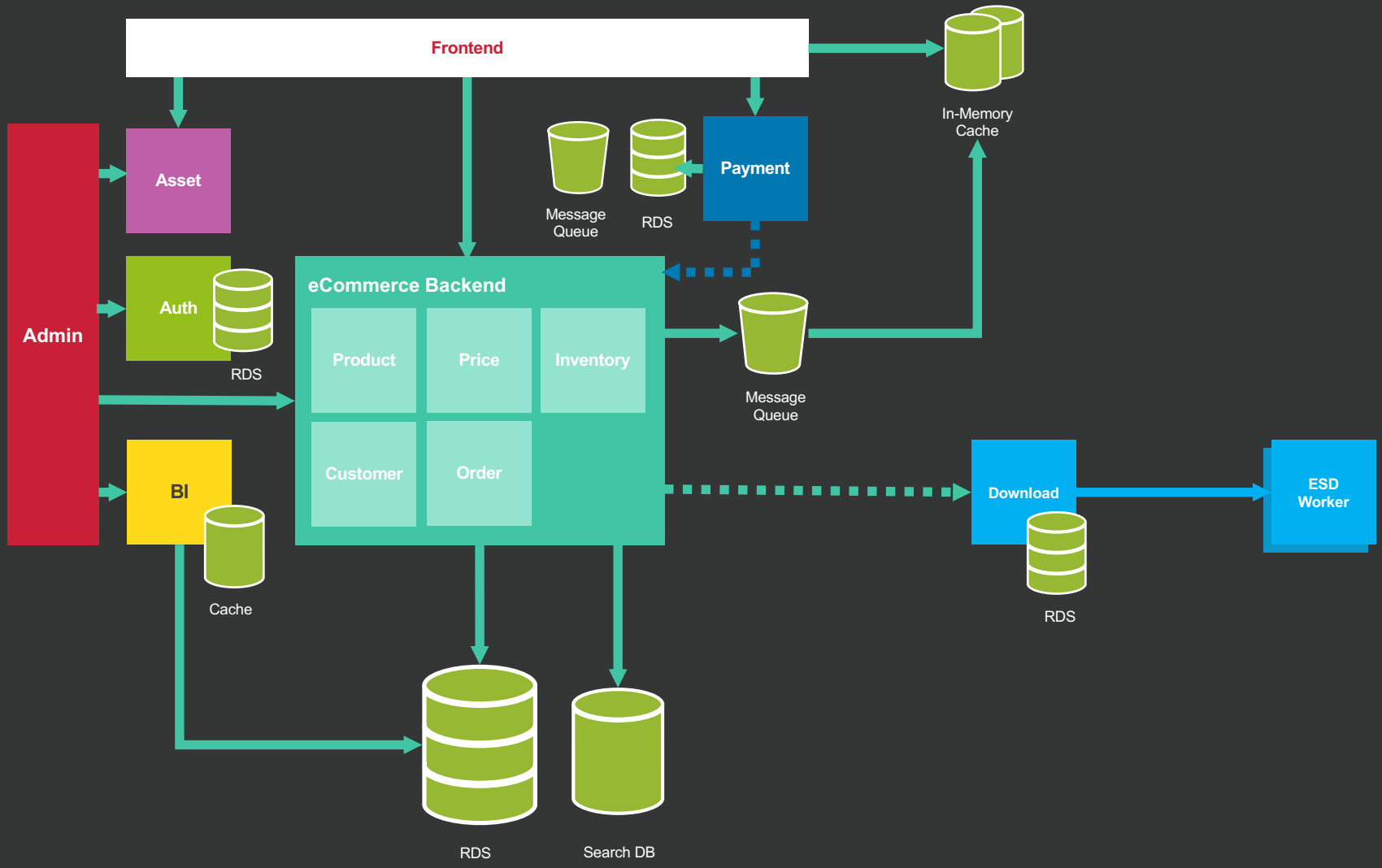
Strangling the monolith

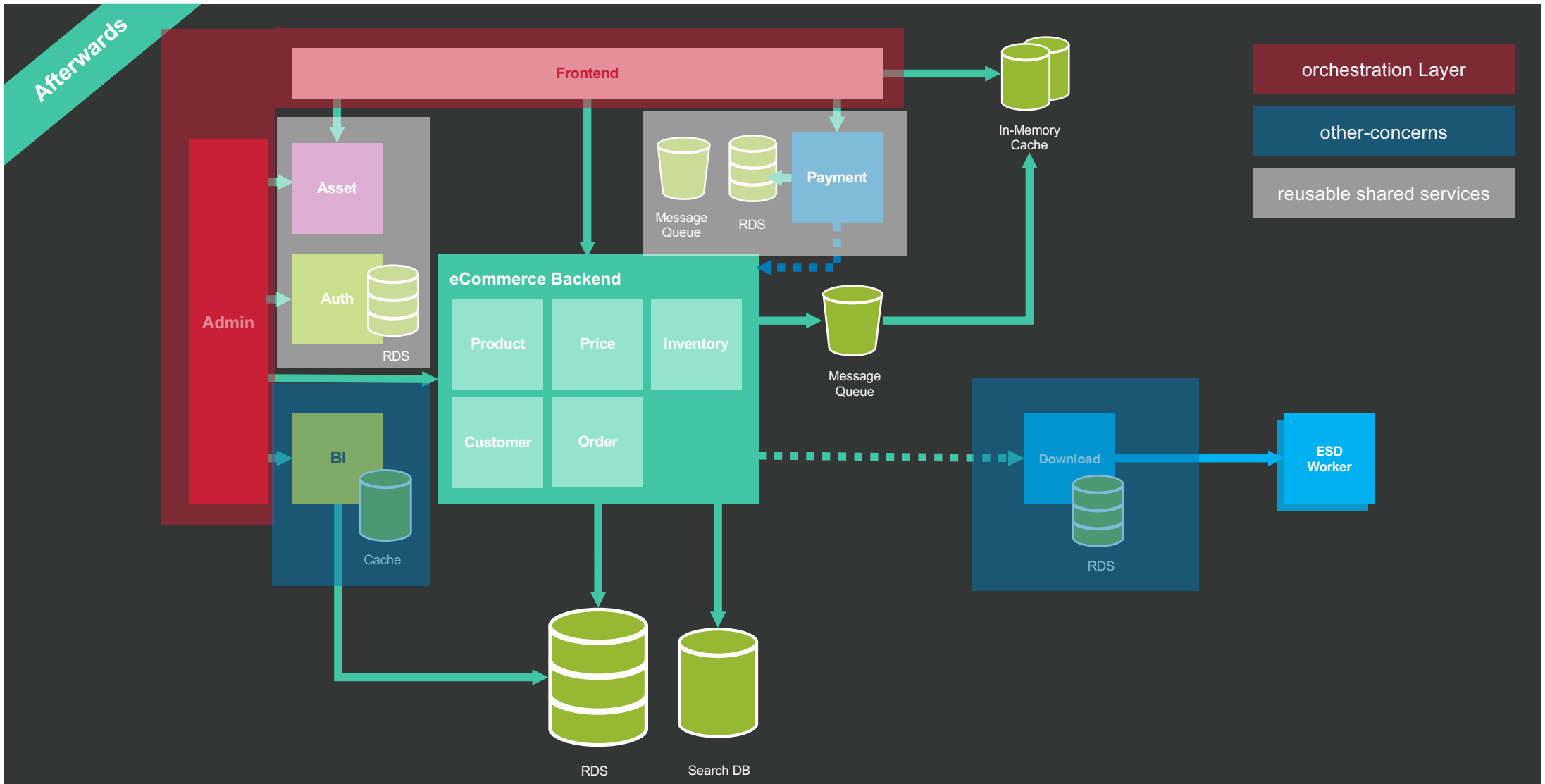
- Strangler (figs) application pattern



https://microservices.io/patterns/refactoring/strangler-application.html
https://martinfowler.com/bliki/StranglerFigApplication.html

Afterwards

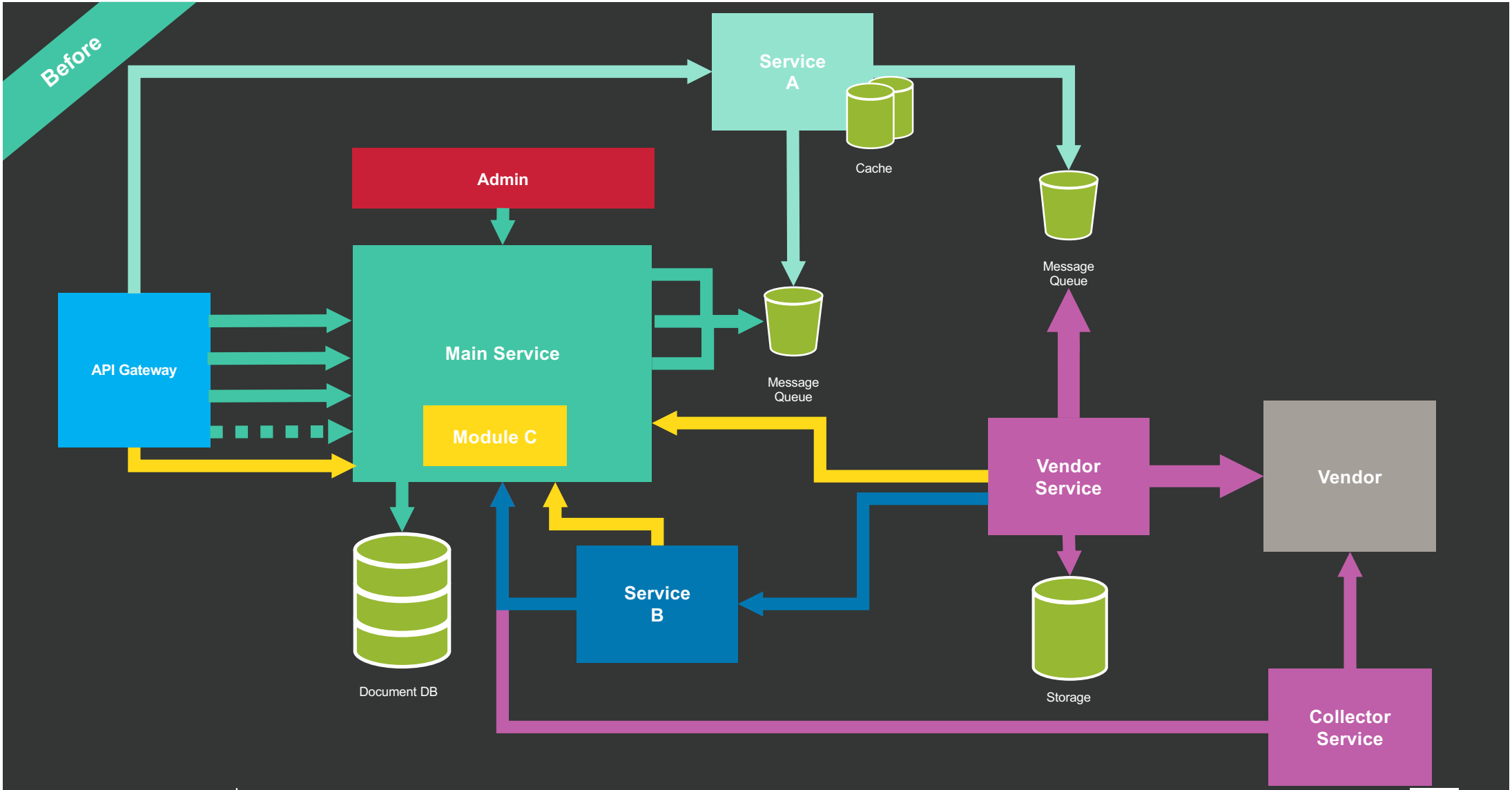


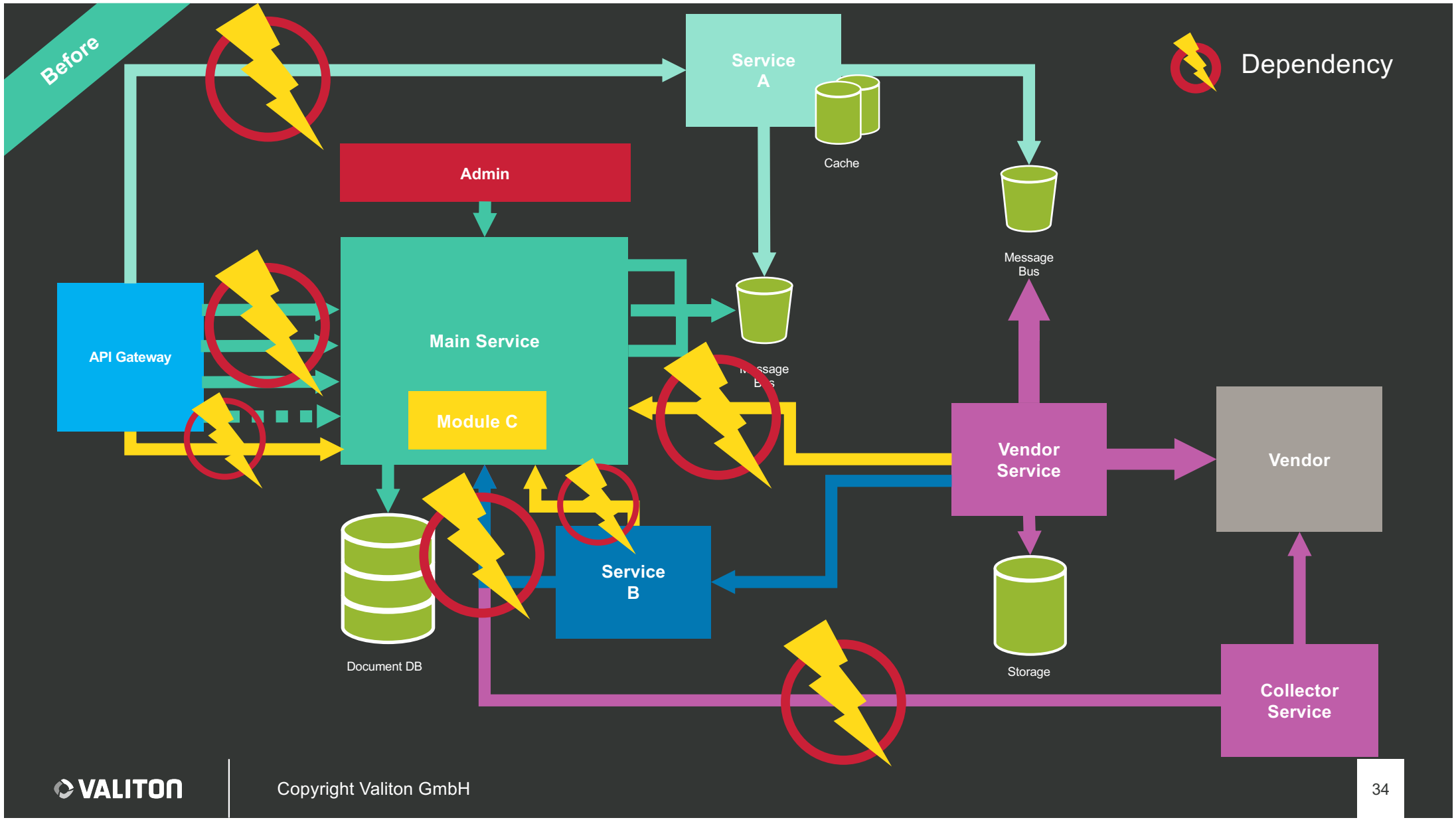


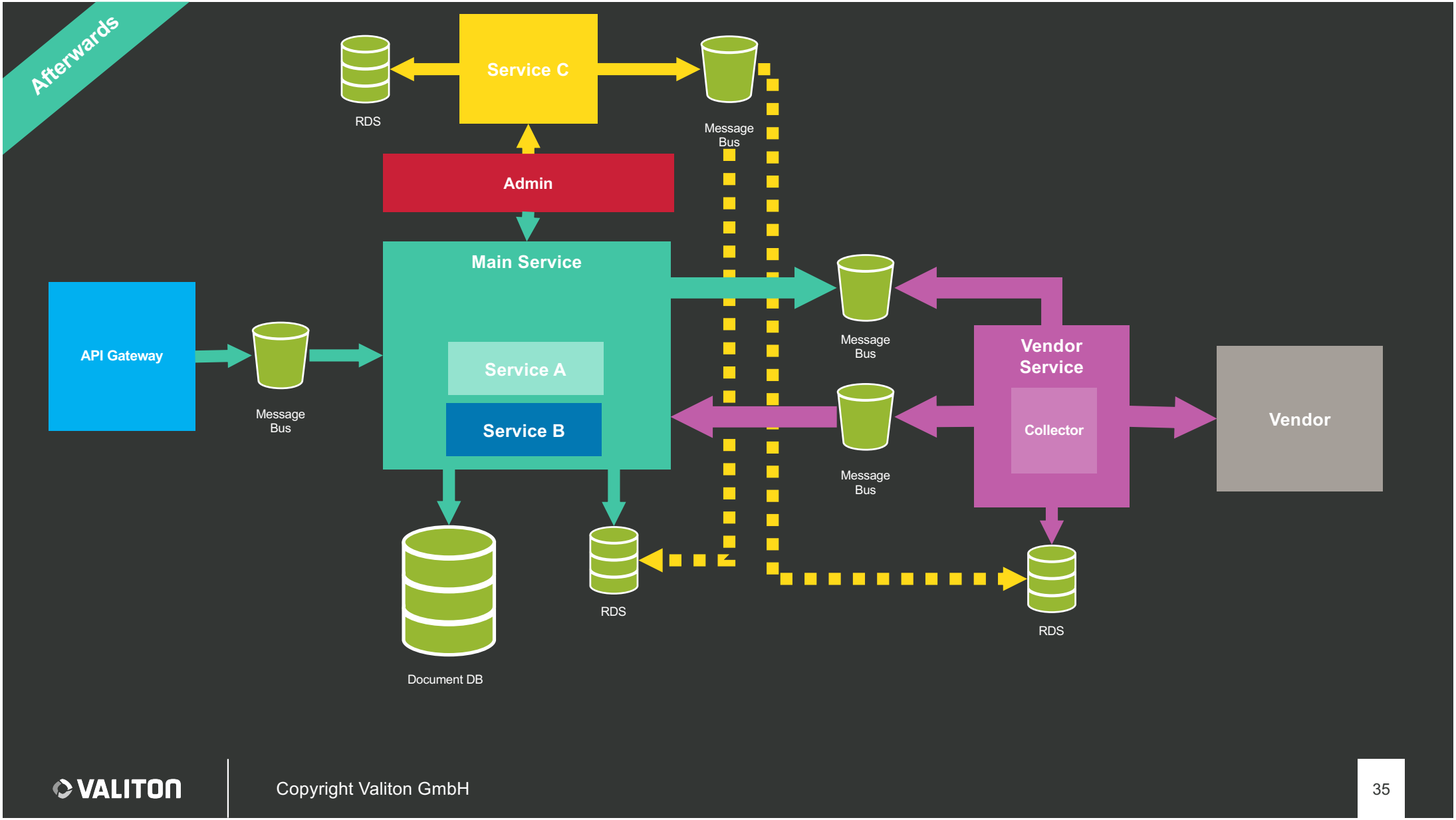
Learnings

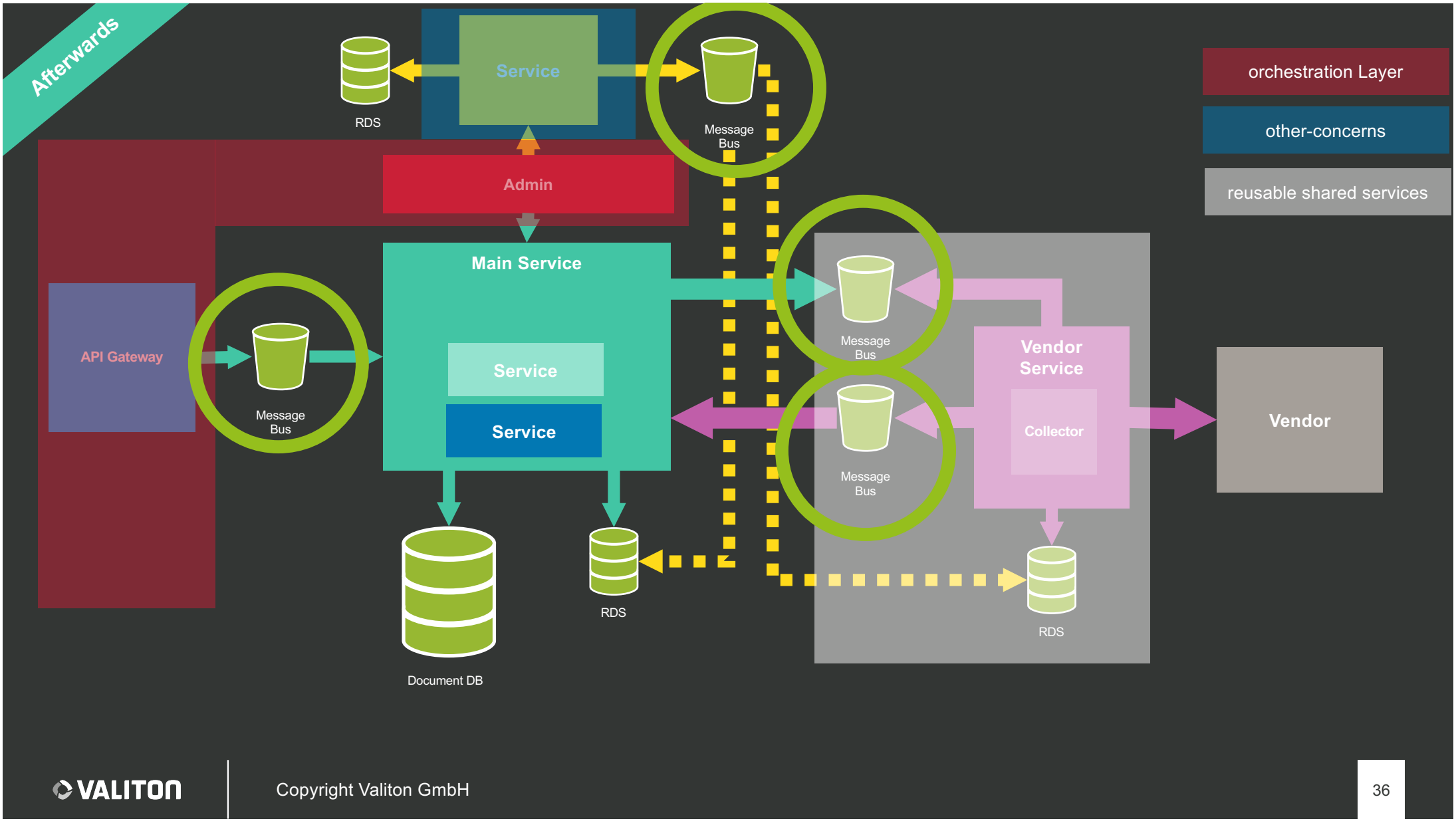
- first start refactoring your codebase, step by step (Boy-Scout principle)
  - strict boundries between modules (Clean Architecture / SOLID)
  - high test coverage
- extract frontend (orchestration layer)
- extract shared services eg. authentication

Example: Distributed Monolith









Learning

- don't depend on other services
- avoid synchronous communication between services
- message bus provides failover
- don't slice your services to small - prefer micro-monolith over atomic services

5 Summary

Monolith First

« you shouldn't start a new project with microservices, even if you're sure your application will be big enough to make it worthwhile. »

Martin Fowler, Author of "Patterns of Enterprise Application Architecture"

source: https://martinfowler.com/bliki/MonolithFirst.html

Take Aways – Monolith first

- don't do it because others do
- migration from a Monolith to Microservices is an Epic journey
- takes a lot of time and money
- brings tons of challenges
- first start refactoring your monolith step by step (continuous refactoring)
  - identify bountries and refactor into separate modules
    (SOLID, DDD, Clean Architecture)

Take Aways - Microservice the right way

- loosely coupled services are the key
  - message queues or message bus (event driven architecture)
  - separate databases and data schemas
- don't slice your services to small (fragmentation)
- invest in Observability: centralized logging and distributed tracing
- automate deployment: Continues Integration and Deployment (CI/CD)

Valiton GmbH
München

Arabellastraße 23
81925 München

Valiton GmbH
Berlin

Potsdamer Straße 7
10785 Berlin

Valiton GmbH
Offenburg

Hubert-Burda-Platz 1
77652 Offenburg

Sven Sanzenbacher

Teamlead TECH OG "Atlas"

sven.sanzenbacher@valiton.com

+49 781 84 6309

VALITON
Ein BurdaSolutions Unternehmen