How to Create Resilient Microservices With a PostgreSQL Dependency

Glen Gomez Zuazo
Senior Solutions Architect
September 13, 2019

Meet Glen

User Profile

- Senior Solutions Architect
- On the Go - Running
- LatinX Representative

User Pain Point

- Time Allocation
- Where is Glen?
- Accent :)

User Requirements

Glen Wants

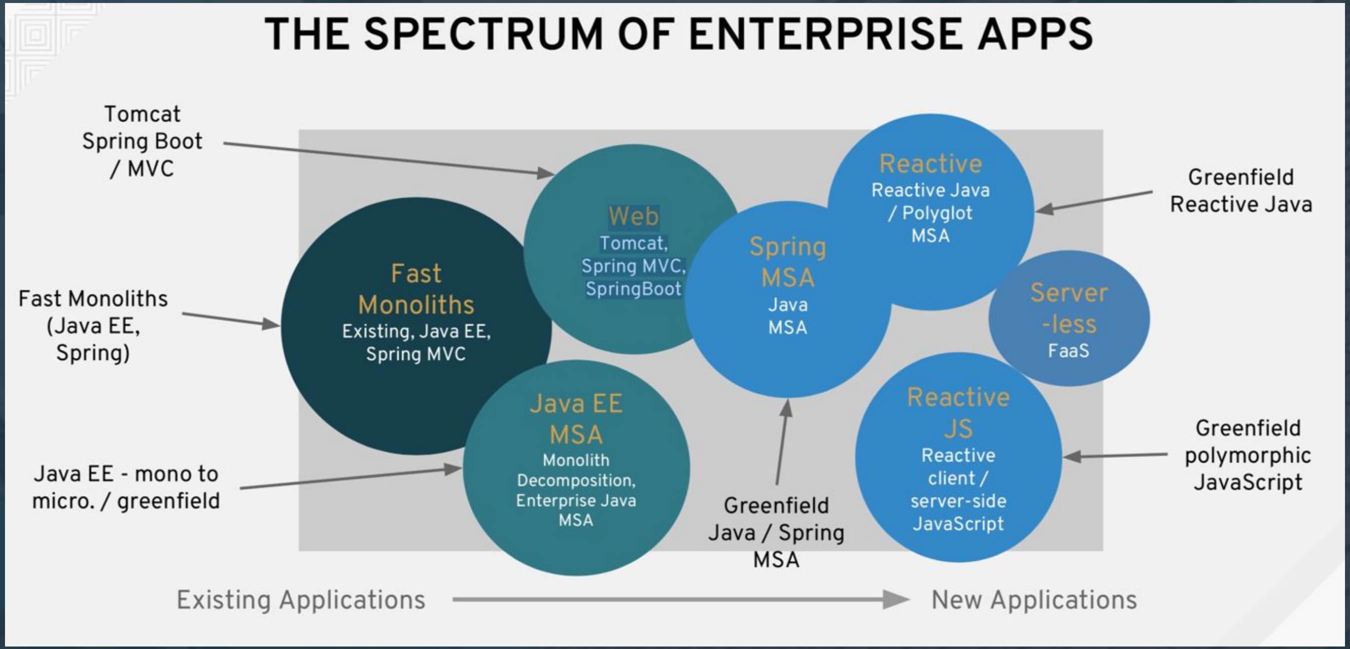
"I want to return to my community and help encourage STEM learning in early stages (middle and high school)"

"I want to help my teams build well-architected solutions using new and emerging technologies"

"I want a 28-hour day..."

Microservices, why should I care?

- Understanding Cost (Operational and Cultural)
  - Operational Cost (CI/CD Pipeline, Login, Monitoring, Tracing)
  - Cultural Cost (Collaboration, Waterfall Mentality, Coordination, Colocation)
- Capabilities and Bounded Context
  - How to identify and why do I need them? Which context I should care? Business or technology
- Understanding the Spectrum of Enterprise Applications
  - Existing Applications
    - Web Tomcat
    - Fast Monoliths
    - Java EE MSA
  - New Applications
    - Spring MSA
    - Reactive
    - Serverless
    - Reactive JS


THE SPECTRUM OF ENTERPRISE APPS
Tomcat Spring Boot / MVC
Fast Monoliths (Java EE, Spring)
Java EE - mono to micro. / greenfield
Fast Monoliths Existing, Java EE, Spring MVC
Web Tomcat, Spring MVC, SpringBoot
Java EE MSA Monolith Decomposition, Enterprise Java MSA
Spring MSA Java MSA
Reactive Reactive Java / Polyglot MSA
Server -less FaaS
Reactive JS Reactive client / server-side JavaScript
Greenfield Java / Spring MSA
Greenfield Reactive Java
Greenfield polymorphic JavaScript
Existing Applications
New Applications

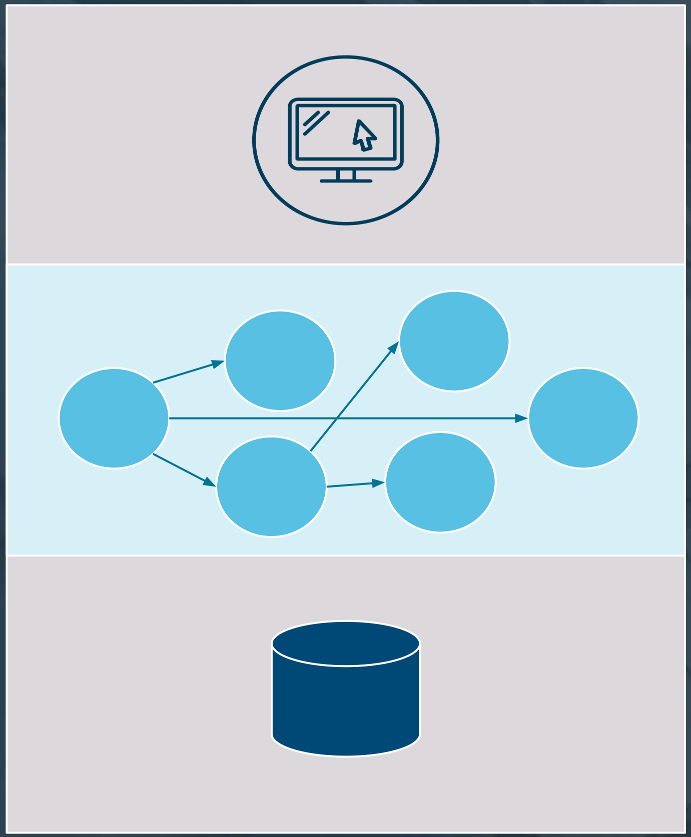
Lift & Shift

Connect & Extend

Rip & Re-Write

API or Microservice: What's the difference?

Change for Insulation



UI Monolith

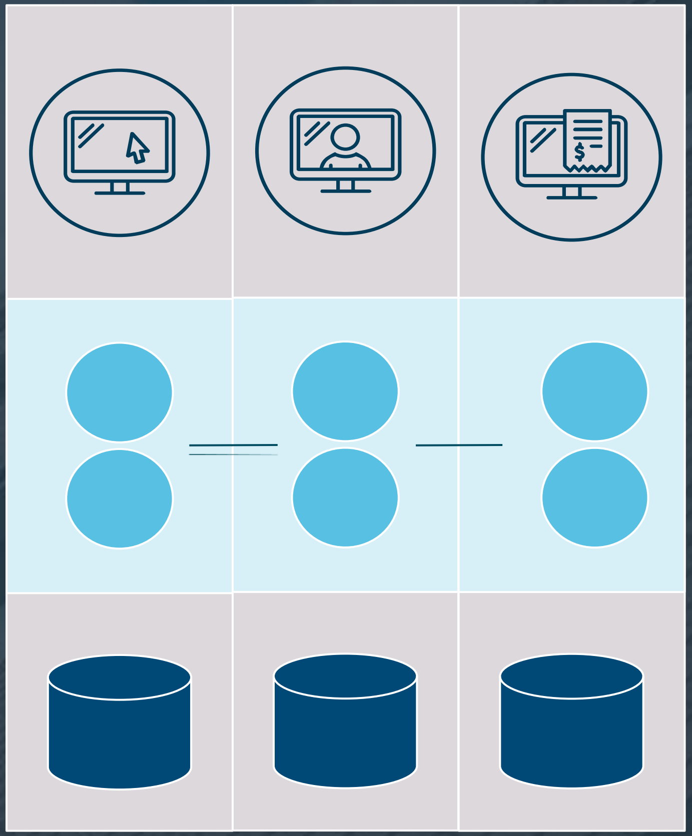
Dependent Services

Data Monolith

UI per Subdomain

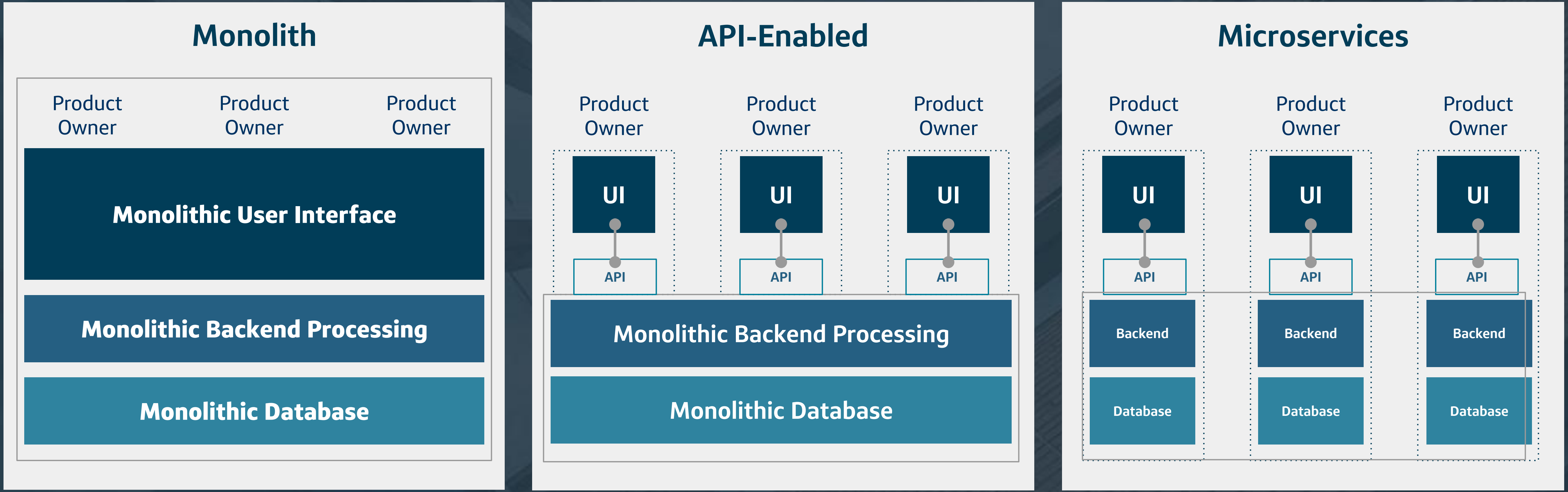
Independent Services

Data per Service



Monolithic setups slow down delivery and innovation

Microservices are the key to creating small, independent, and fully functional bits of software



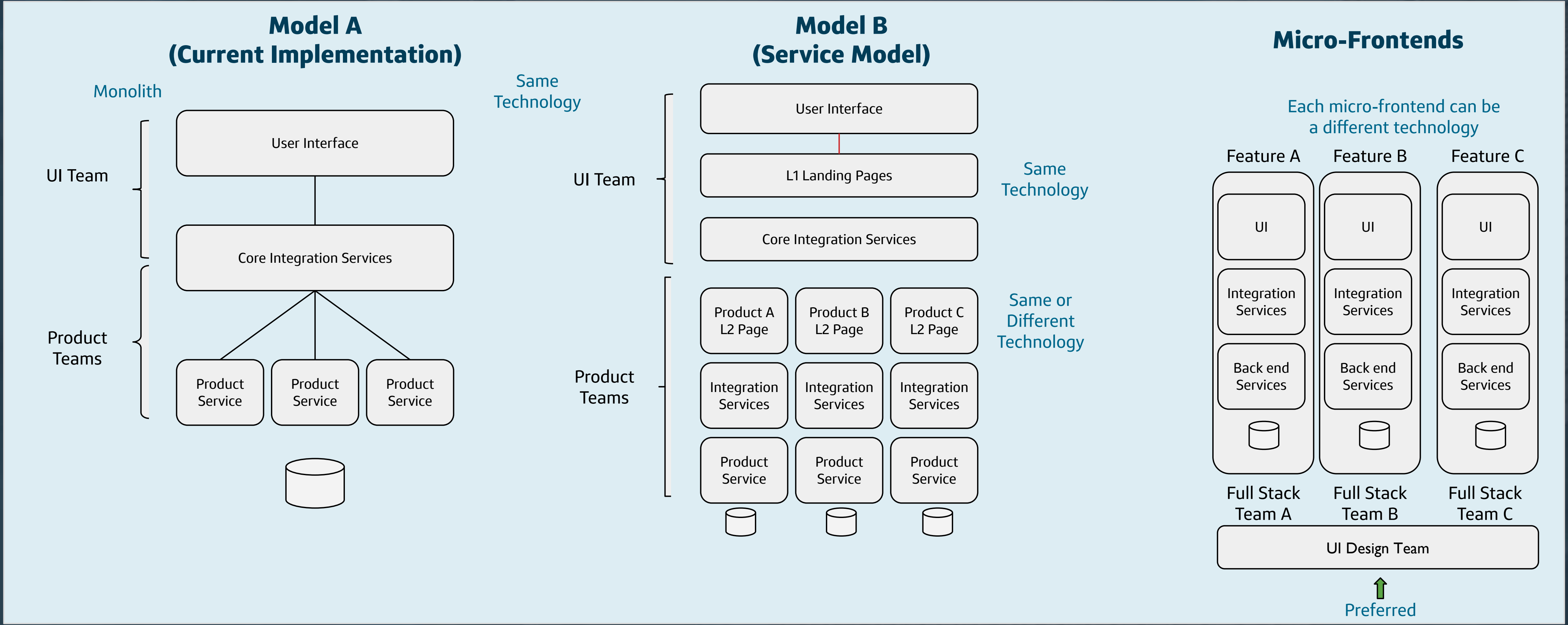
Interconnected services are helping us reduce cross-team dependencies

Monolith

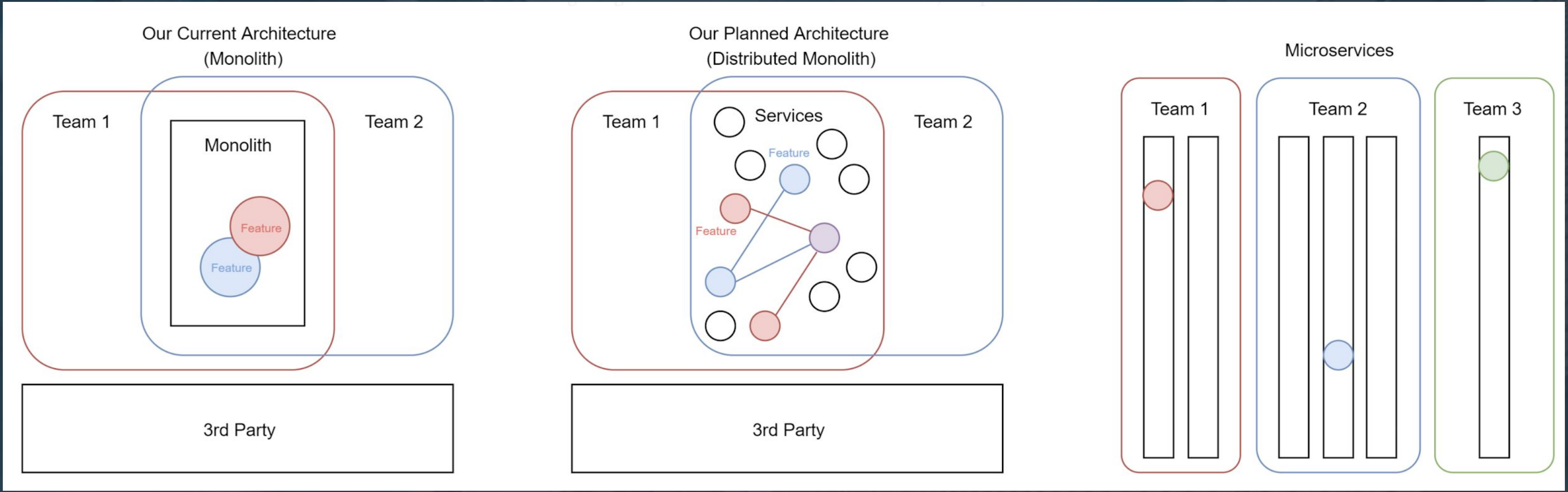
API-Enabled

Microservices

Restructuring Delivery Model



Restructuring Team Ownership

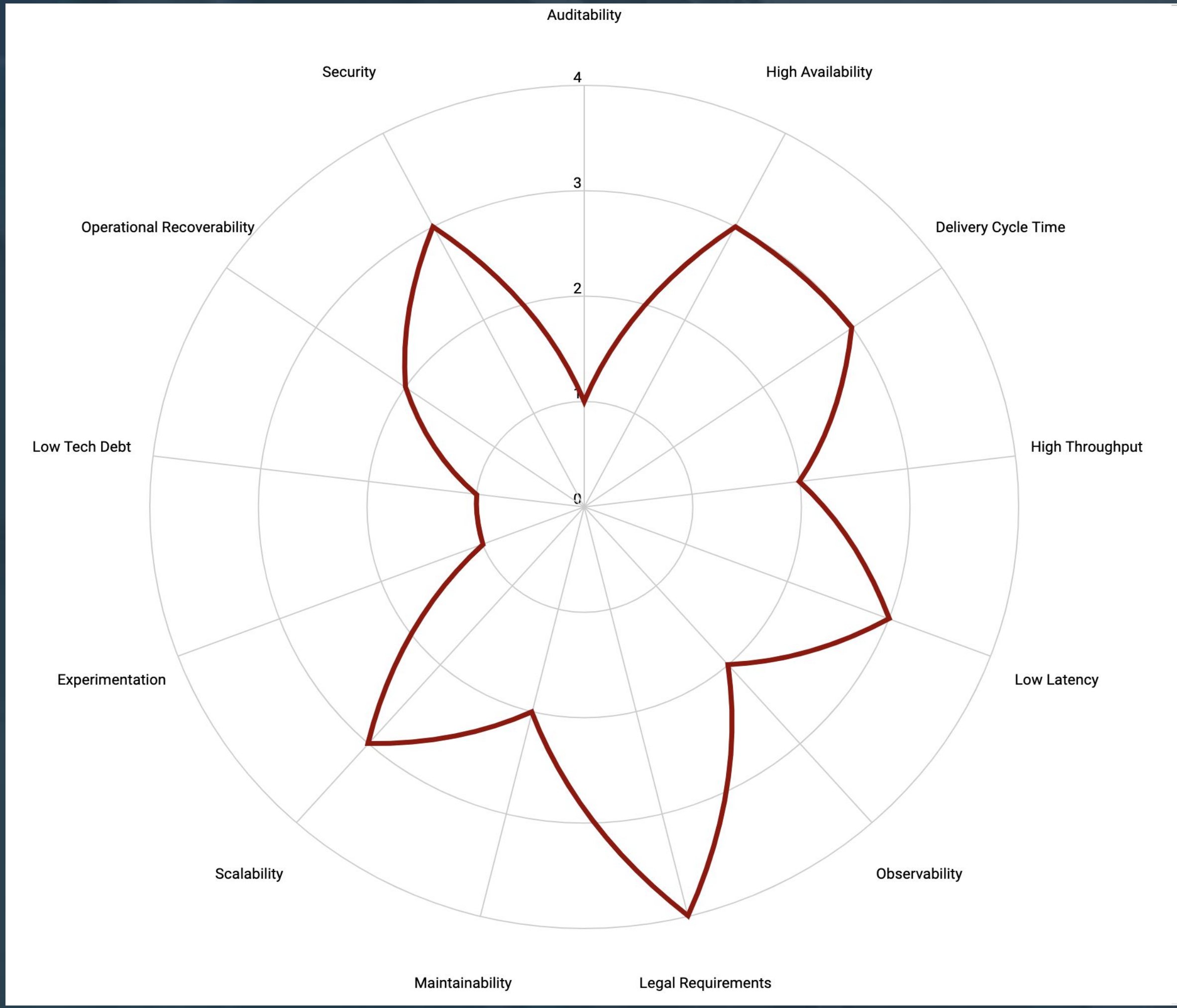


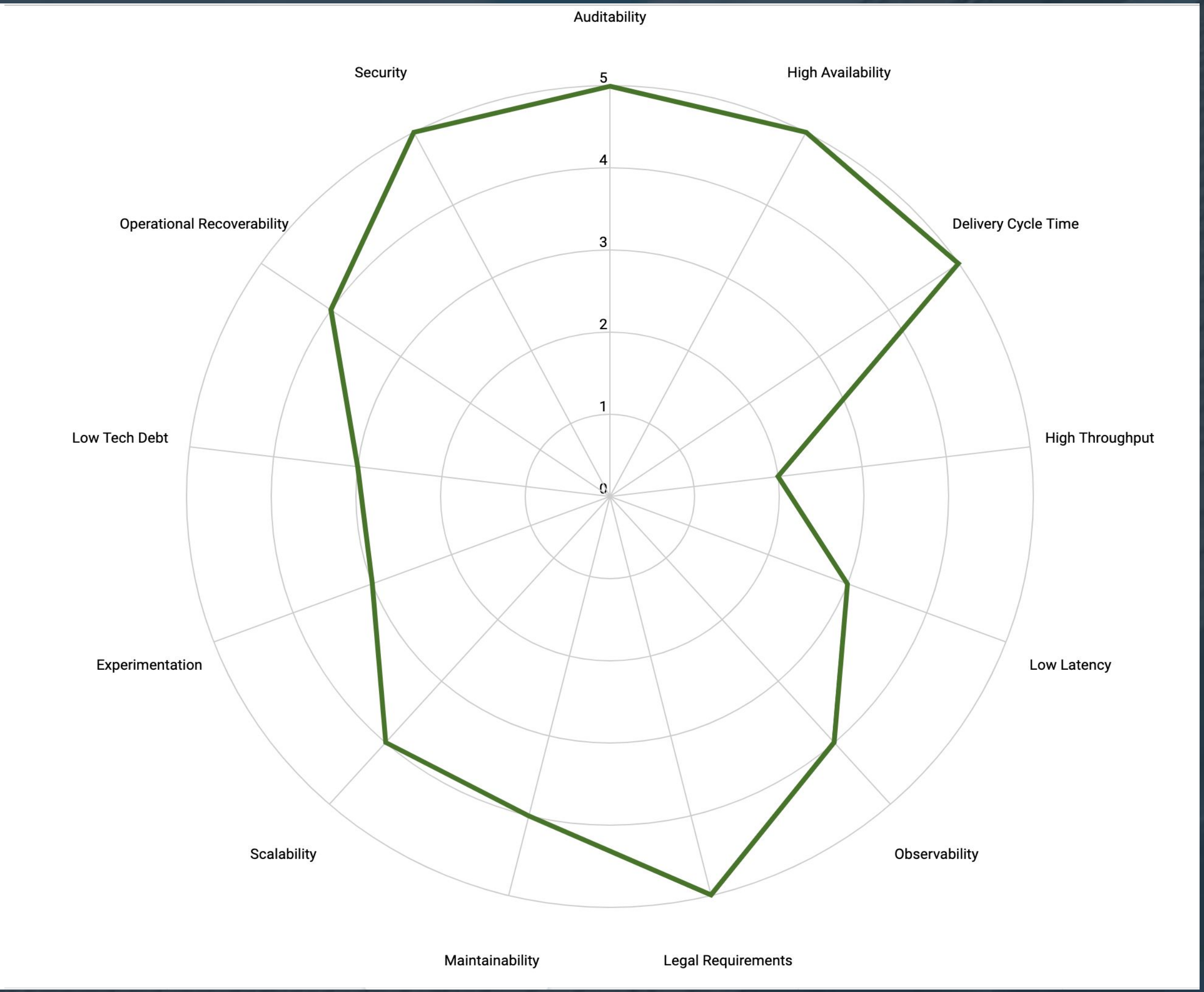
Monolithic vs. Microservices

Multiple Identities	Singular Identity
Operational Coupling	Operational Isolation
Binary Coupling	No Binary Coupling
Synchronous Communication	Asynchronous Communication
Only Java	Beyond Java (Polyglot Support)
Weekly Release	Anytime Release
Data Monolith	Explicit Data
UI Monolith	Micro Frontend
	Fitness Function Guided Evolutionary Architecture

Fitness Function

Move to an architecture that supports evolution





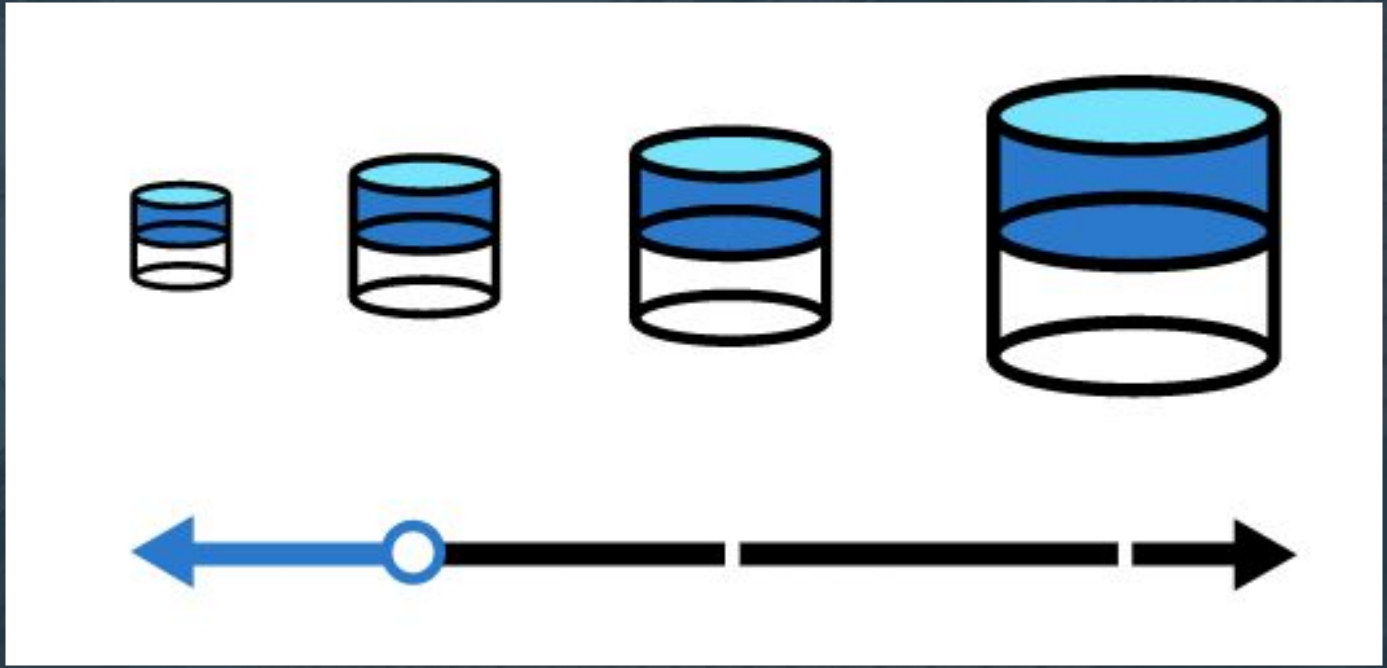
AWS RDS, Aurora, EC2, or Azure PostgreSQL (Citrus)

Considerations

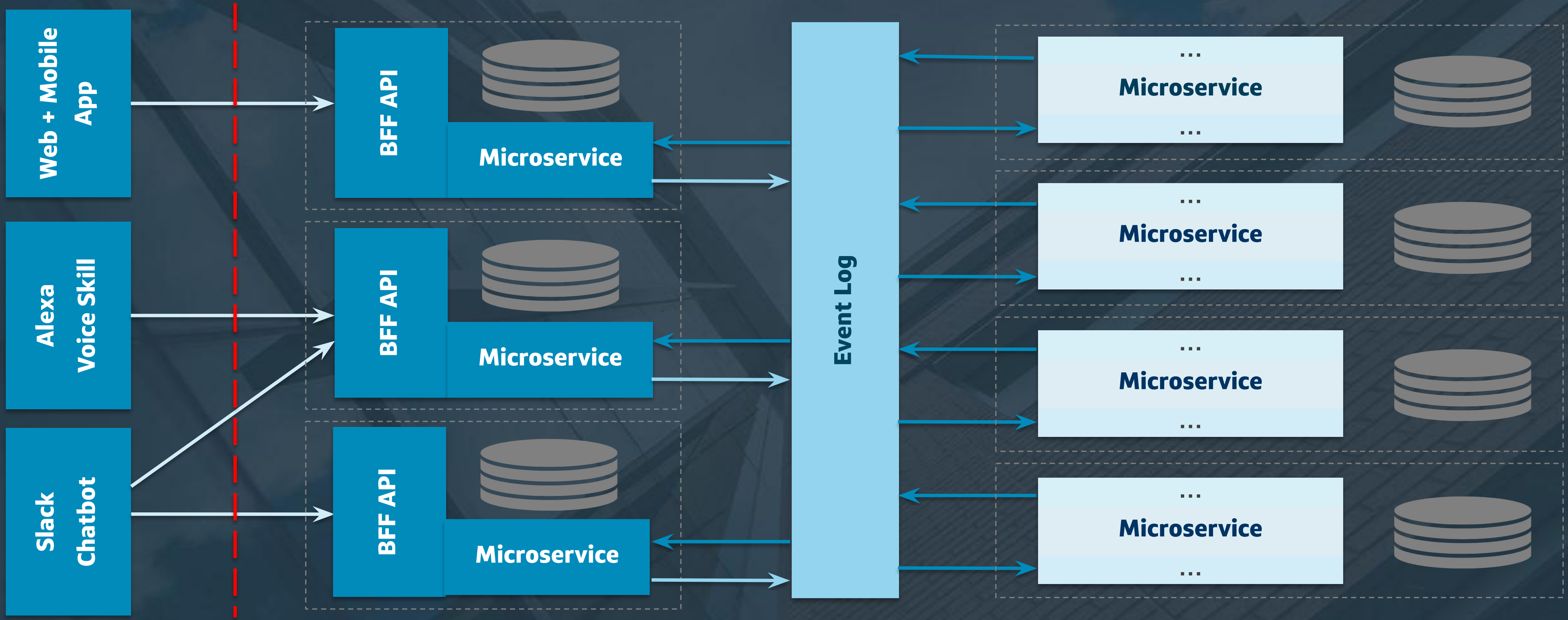
- No OS Patch (Server Maintenance)
- Optimized Performance 3x
- Full DB Admin Control
- Hyperscale (Single- & Multi-Nodes)

Why PostgreSQL

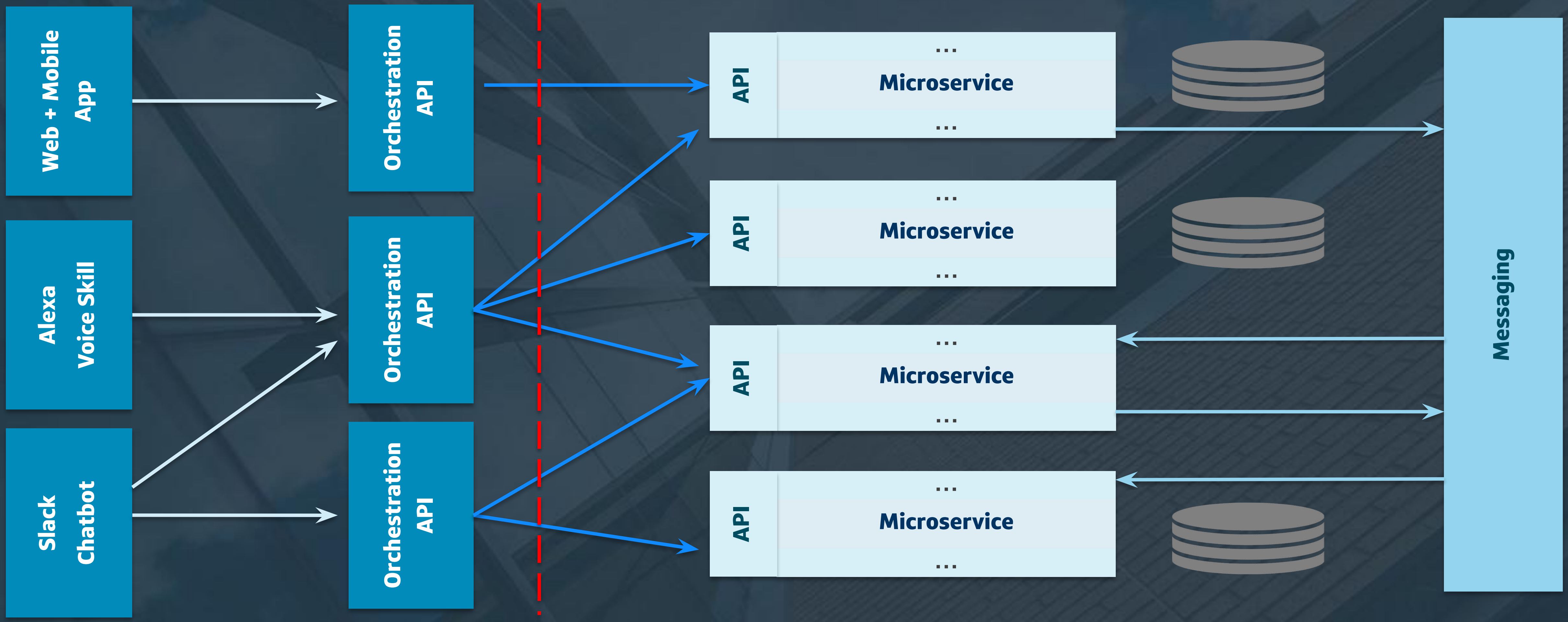
- Open Source
- Cloud Provider Agnostic
- Scalable and Highly Available
- Hybrid (JSONB) - Key/Value



Synchronous Microservices



Asynchronous CQRS & ES Microservices



Q&A

Thank You