CENTRO UNIVERSITÁRIO
UNI METROCAMP | WYDEN

Sistemas Distribuídos _ Microservices

santiago.robles@unimetrocamp.edu.br

maio/19

O que é um *Service* ?

- AUTÔNOMO
  - é uma unidade de software que executa uma função de negócio;
- CONTRATO FORMAL
  - estabelece um contrato bem definido: entradas, saídas, restrições, premissas, comportamento (lógica de negócio);
- ABSTRAÇÃO
  - esconde do seu consumidor todos os detalhes de implementação, incluindo infraestrutura e lógica interna;
- ENCAPSULAMENTO
  - realiza uma função de negócio integralmente, de forma consistente;
- BAIXO ACOPLAMENTO
  - é auto-contido, independente e modular;
- REUTILIZÁVEL
  - pode ser reutilizado em outros contextos;
- CAPAZES DE SEREM DESCOBERTOS
  - pode ser disponibilizado e administrado individualmente;
- CAPAZES DE SE COMPOR
  - pode ser combinado com outros serviços para compor um serviço de mais alto nível ou um processo de negócio;

SOA é uma arquitetura para toda a empresa, buscando uma integração efetiva entre as aplicações.

Microservices é uma arquitetura de aplicação (Componentes)

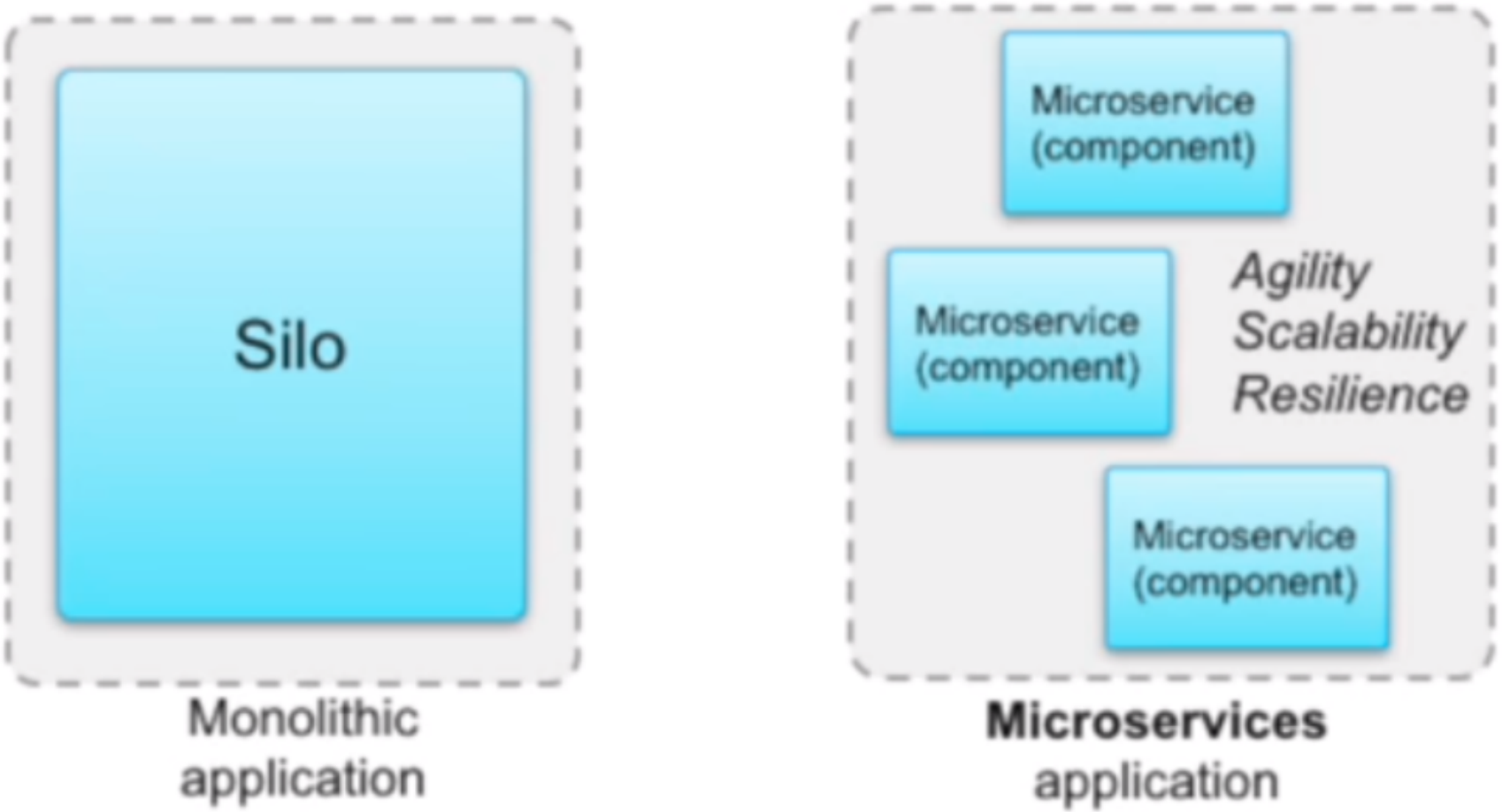
Microserviços (Microservices)

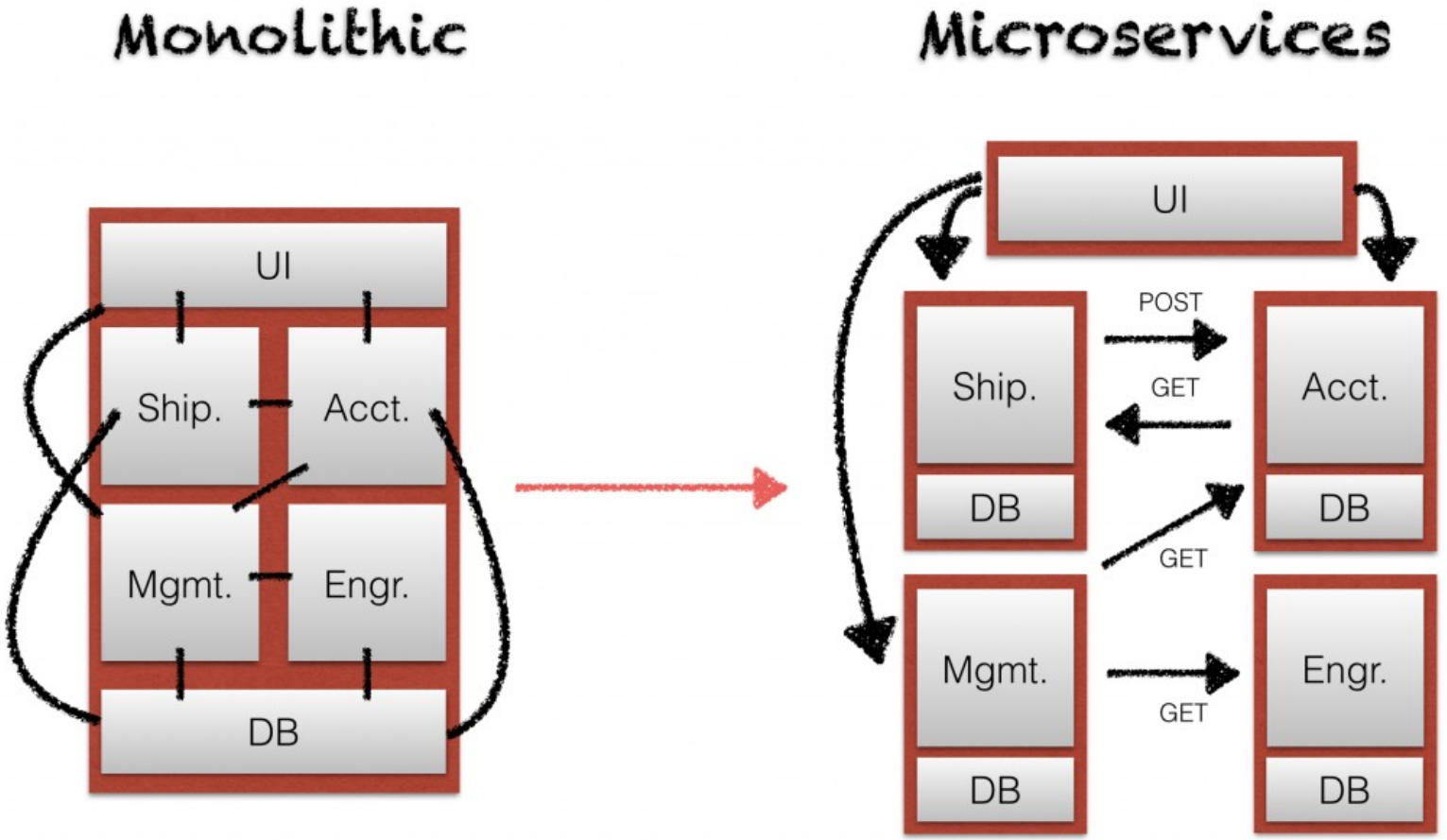
Microserviços é uma maneira particular de desenvolver aplicações de maneira que cada módulo do software é um serviço *standalone* cujo *deploy* e escala acontecem de maneira independentes da “aplicação principal”.

Microserviços (Microservices)

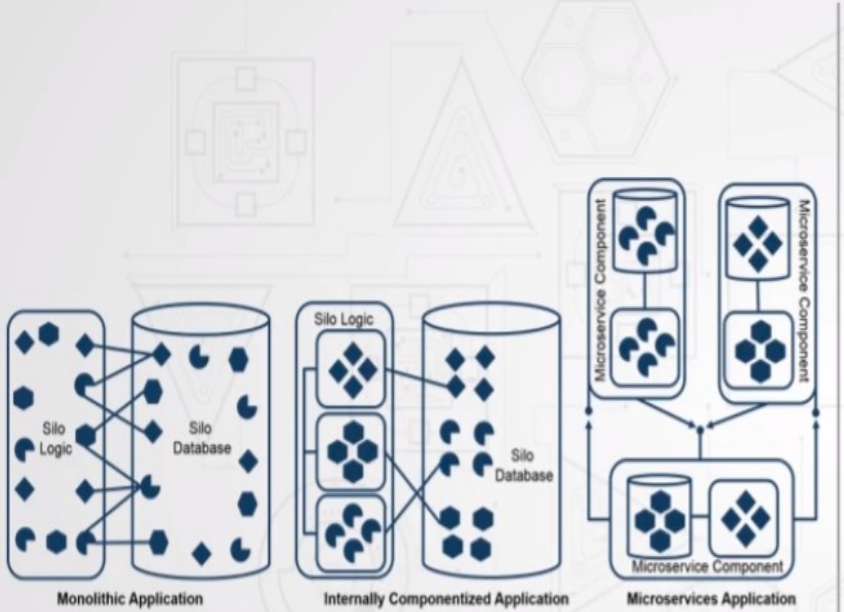
Enquanto na arquitetura tradicional de software, chamada monolítica, quebramos uma grande aplicação em bibliotecas, cujos objetos são utilizados in-process, em uma aplicação modular como proposta na arquitetura de microservices cada módulo recebe requisições, as processa e devolve ao seu requerente o resultado.

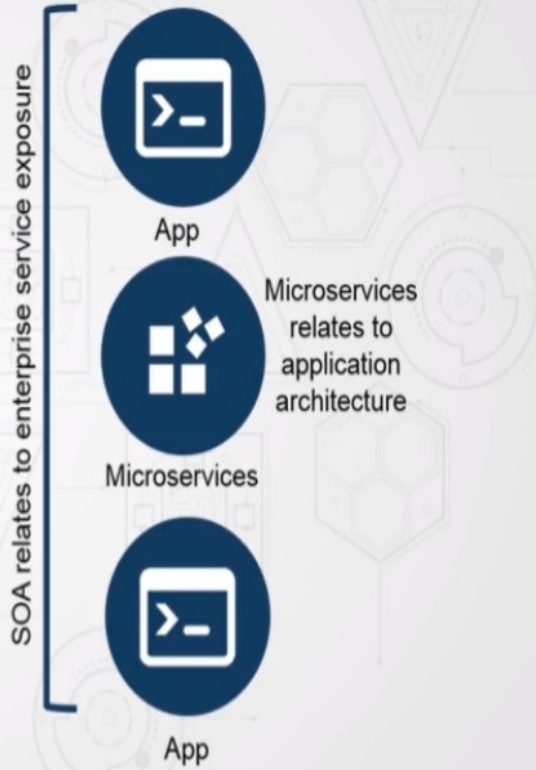
O QUE É A ARQUITETURA DE MICROSERVIÇOS ?

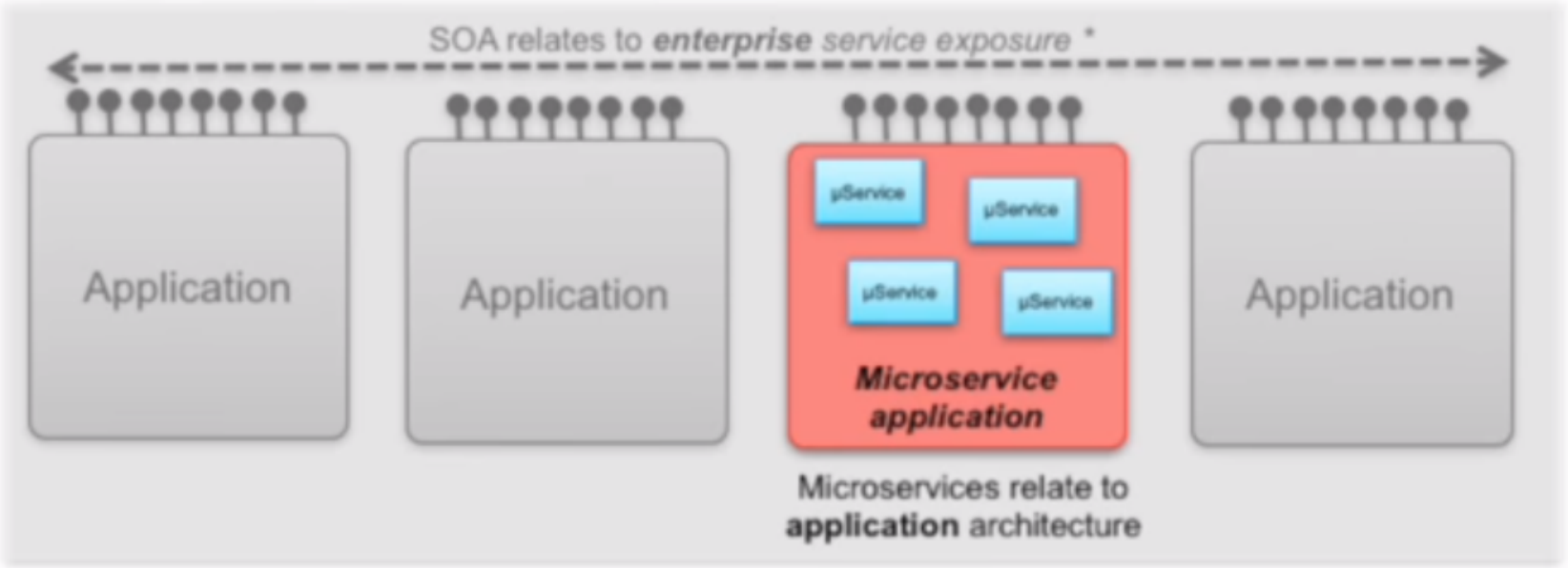


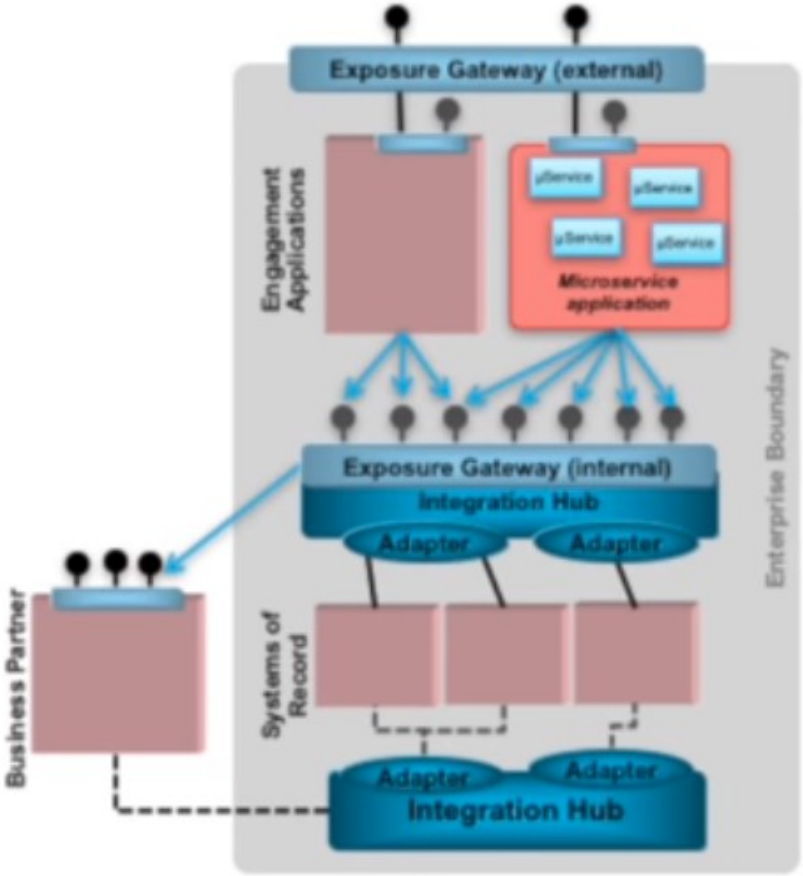


Difference Between SOA and Microservices



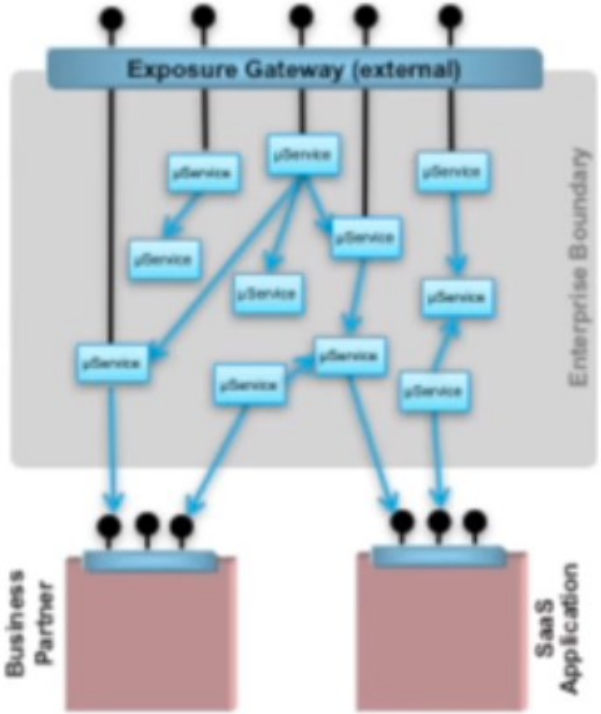



SOA relates to *enterprise* service exposure *
Application
Application
µService
µService
µService
µService
Microservice application
Application
Microservices relate to **application** architecture



Mature large enterprise

Microservices are just one style of application
Exposing services is an *integration and data* challenge



Green field online start-up

Much of landscape could be microservice based
The landscape *is* a (micro)service oriented architecture

- REUSO não é um objetivo
  - Evitar dependências de código. Busca por **baixo acoplamento** fazendo reuso com cópias
- SINCRONISMO é ruim
  - Evitar dependências de comunicação em tempo real. Uso de mensagens
- Serviços devem ser descobertos em tempo real (runtime)
- DUPLICIDADE DE DADOS
  - Consistência e Replicação

BENEFÍCIOS DA ARQUITETURA DE MICROSERVIÇOS

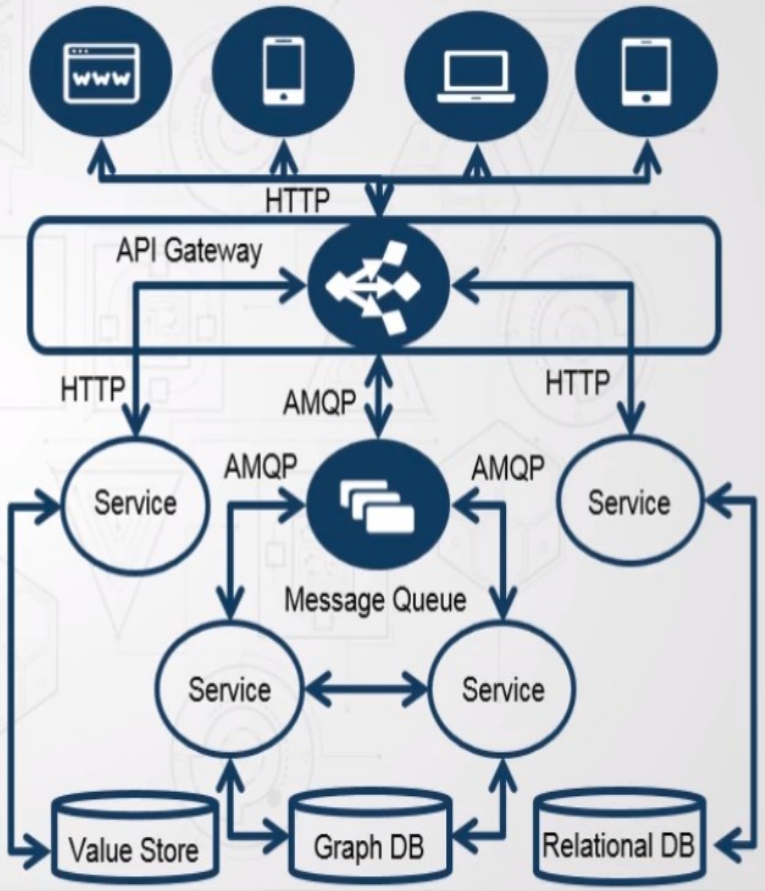
- estruturar uma aplicação como uma coleção de seviços com baixo acoplamento [RESILIÊNCIA]
  - serviços refinados
- Modularidade: Melhora o entendimento do sistema, manutenção do código e testes [ESCALABILIDADE]
- Desenvolvimento em paralelo: Times de desenvolvimento trabalhando de forma independente [PRODUTIVIDADE]
  - continuous refactoring
- continuous delivery and deployment [DEVOPS]

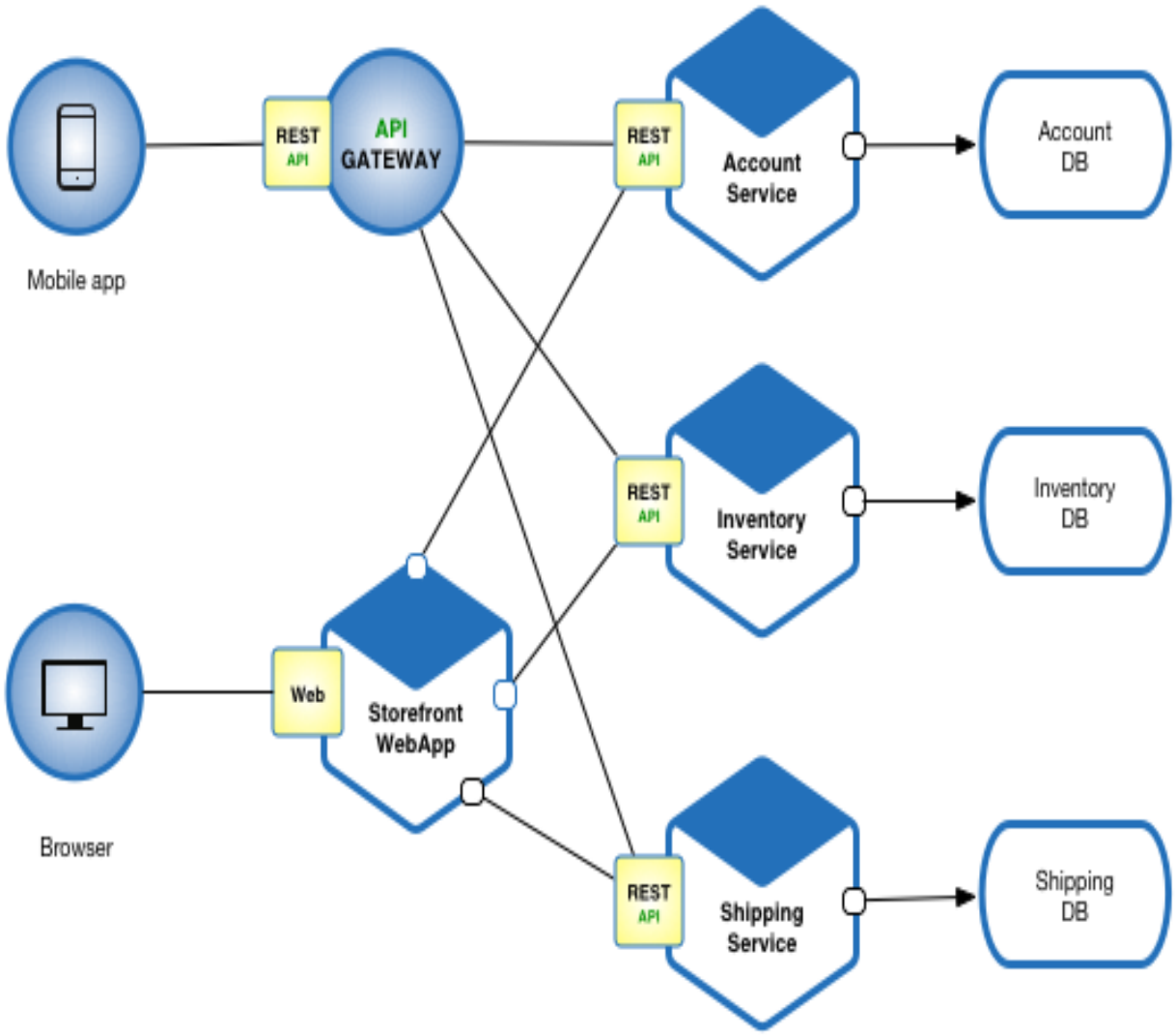
Benefits of Microservices Implementation

Monolithic	Feature	Microservices
• Code base size increases -> decrease maintainability • Overloaded IDE -> slow IDE -> decrease productivity • Overload web server -> long start -> decrease productivity	Maintainability	• Small code base size per MS • Faster IDE • WS fast start
• Redeploy the entire app • Influence background tasks • Increases the risk -> discourages frequent updates	Continuous deployment	• Each service can be deployed independently • Smaller apps are easier to deploy • Downtime
• Running more copies of the same app • Different components have different requirements	Scalability	• Scale on the level of MS
• Low fault isolation	Availability	• Improved fault isolation • Increased complexity - fall fast
• Long time commitment to technology stack	Take advantage of emerging technologies	• Each MS can be developed using different stack
• Hard to manage large teams • Agile is not scaling well • Hard-to-scale development	Efficient governance	• Each team responsible for one MS • Agile is working well • Easy-to-scale development

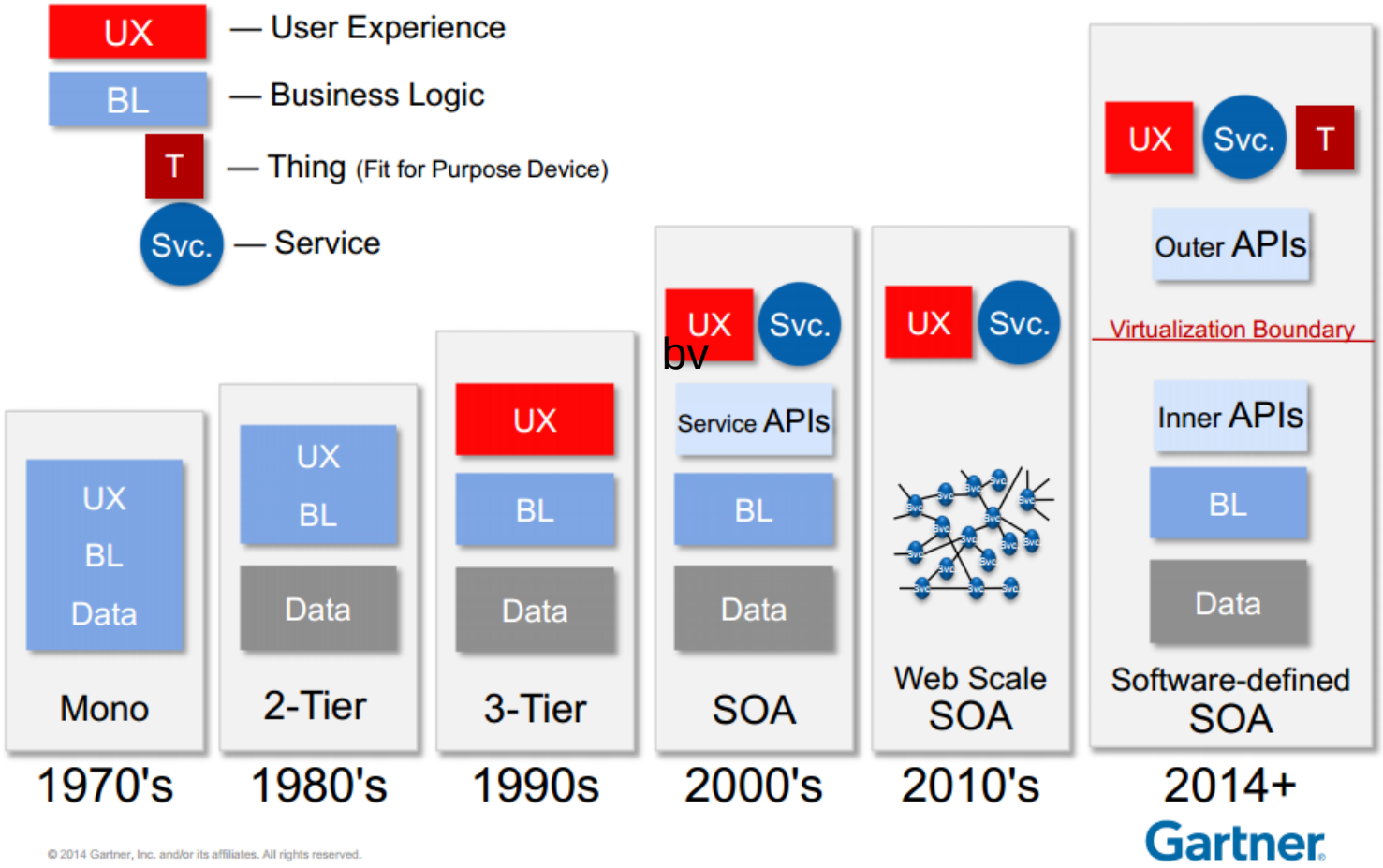
Microservices Architecture

- Method of developing software applications as a small unit of
  - Independently deployable modular services
  - Each service runs a unique process
  - Service communicates through
    - Well-defined
    - Lightweight mechanism to serve a business goal




Mobile app
REST API
API GATEWAY
REST API
Account Service
Account DB
REST API
Inventory Service
Inventory DB
Browser
Web
Storefront WebApp
REST API
Shipping Service
Shipping DB

Software-defined Applications on the Application Architecture Road Map



Faci facid FACIMP FBV fmc Faculdade Martha Falcão ISL UNIFAVIP UNI METROCAMP RUY BARBOSA | AREA1 UniFBV UniFanor