universität innsbruck

Advanced Distributed Systems

Lecture 02-Service Oriented Architecture (SOA) and Microservices

Dr. Zahra Najafabadi Samani

Agenda

- Monolithic application
- Service-oriented architecture (SOA)
- Microservices
- SOA vs Microservices

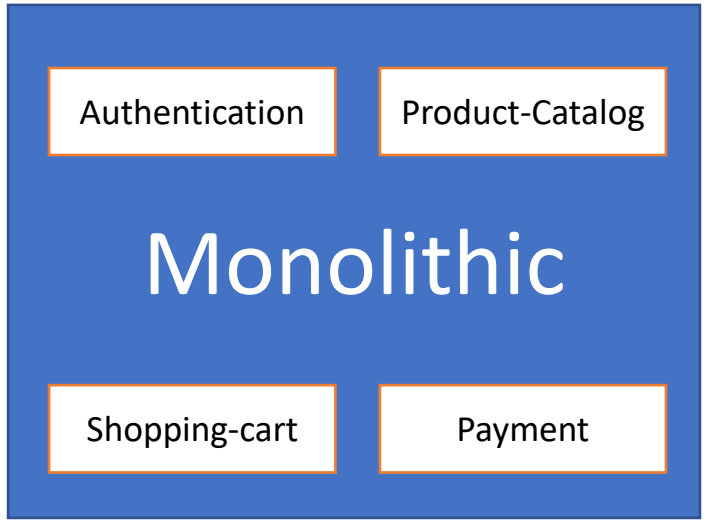
Monolithic application

Monolithic application

- Before SOA, monolithic application was standard.

Monolithic application

- Before SOA, monolithic application was standard.
- All code and application component are part of the single units
- Everything developed, deployed as one unit
- Applanation is in single language



Challenges of monolithic application

- Application is too large and complex
- Parts are more tangle into each other
- Difficulty if service need different dependency version
- Scaling the entire app, instead of a specific service
- Release process takes longer
- On every change, the entire application needs to be tested
- Bug in any module can bring down the entire application

Service-Oriented Architecture (SOA)

Service oriented architecture (SOA)

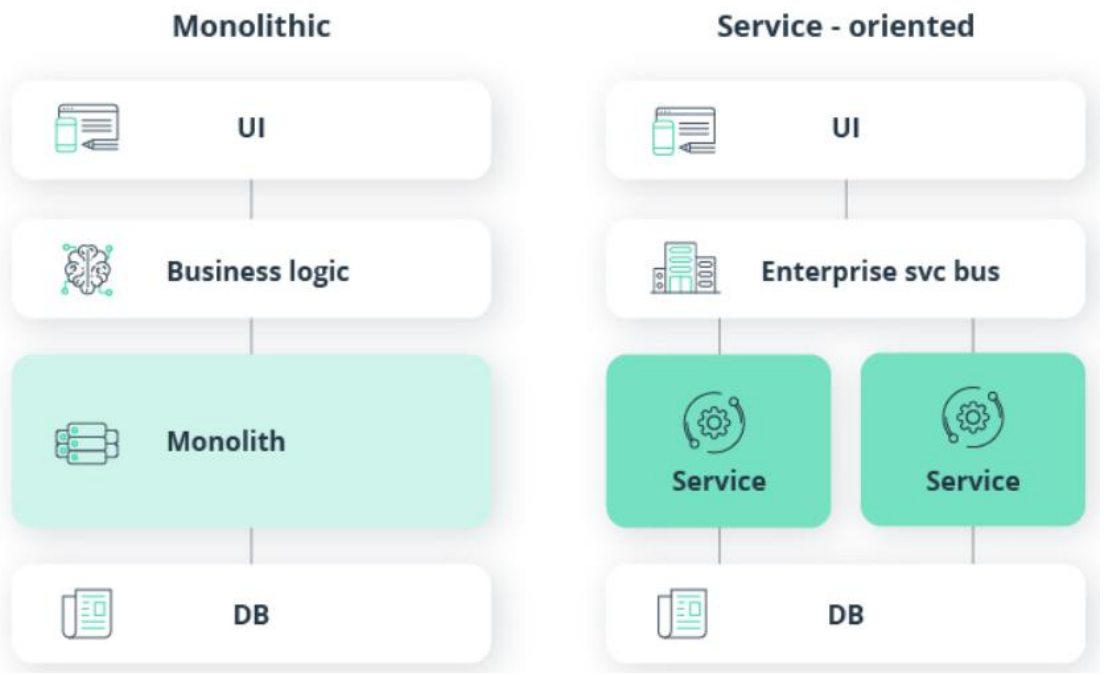
- Service Oriented Architecture is an architectural style/pattern for creating and using business processes as **“services”**.
- An approach for building **distributed computing systems** based on encapsulating **business functions** as **services** that can be easily accessed in a **loosely coupled** fashion.
- Application’s business logic or individual functions are **modularized** and presented as **services** to client/consumer applications.

In essence…….

A shift in focus away from “applications” towards business processes

An architecture in which processes are constructed either in whole or in part by assembling reusable services.

Monolithic vs Service oriented architecture



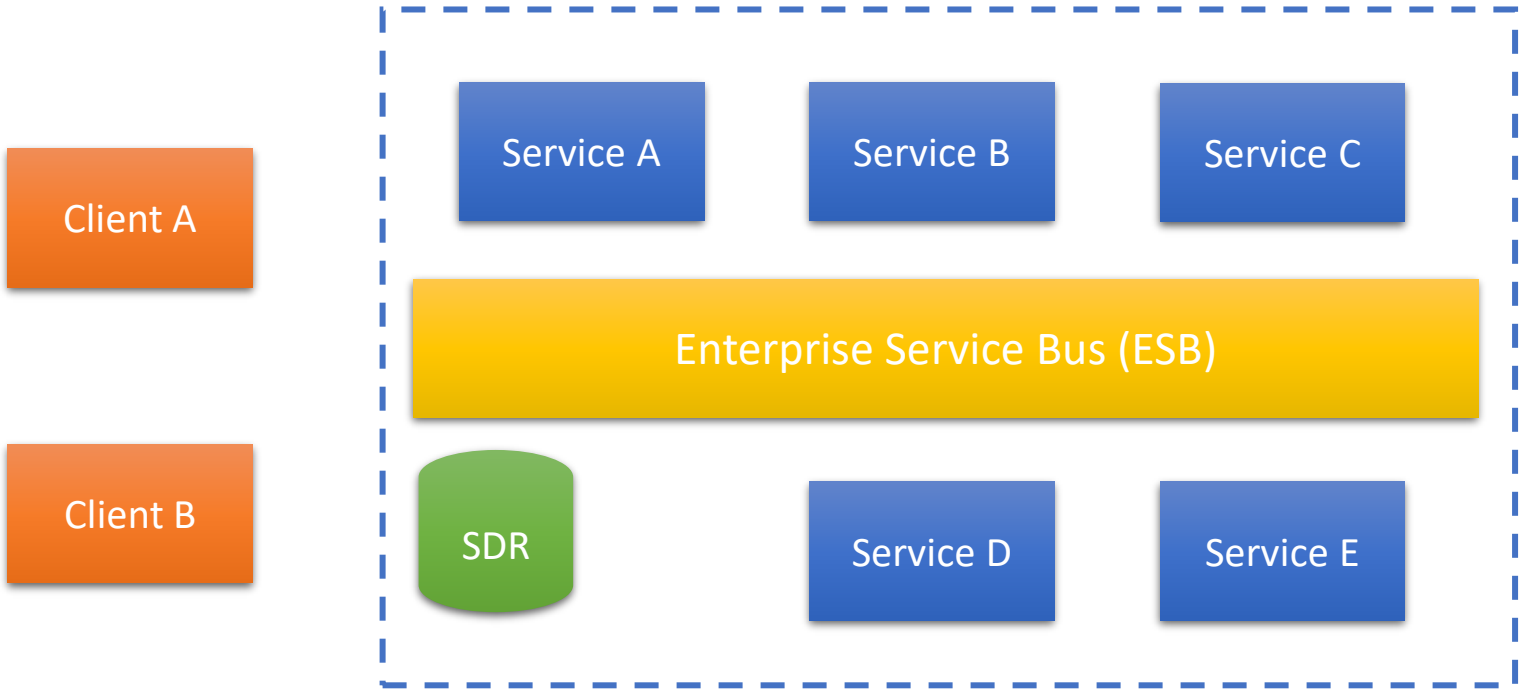
SOA characteristics

- **Loosely coupled**: Services are independent and interact with each other with minimal dependency.
- **Reusable**: SOA services are built to be easily discovered and incorporated into other applications.
- **Protocol-based interactions.** The standard network protocols SOA services use to read/exchange data are SOAP (simple object access protocol) plus HTTP or REST and HTTP.
- **No max size.** Services don't have fixed boundaries — they can be delivered individually or combined in composite service stacks.
- **Process matched.** The scope of a service is modeled to power a specific business process within a company.
- **Self-contained**: Services can remain independent of other system elements.

Benefit of SOA

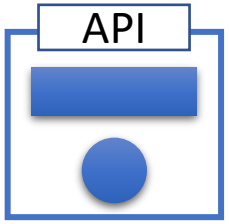
- **Maintainability**: maintaining the individual services without worrying about the rest of the system.
- **Lower total cost of ownership:** resulting from increased maintainability.
- **Reusability**: reused easily whenever needed in different contexts and by different applications.
- **Greater agility**: reusability and loose coupling decrease the development timeline for new services and functionality.
- **Scalability and availability:** scaling the system becomes as easy as scaling and running multiple instances of single services.
- **Platform independence:** integrating different products regardless of the underlying technology or infrastructure.
- **Responsiveness and improve TTM:** improving responsiveness to business changes and customer requirements with agility.

SOA architecture



SOA components

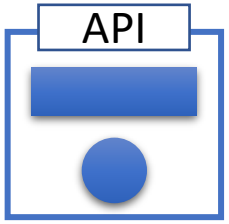
Services



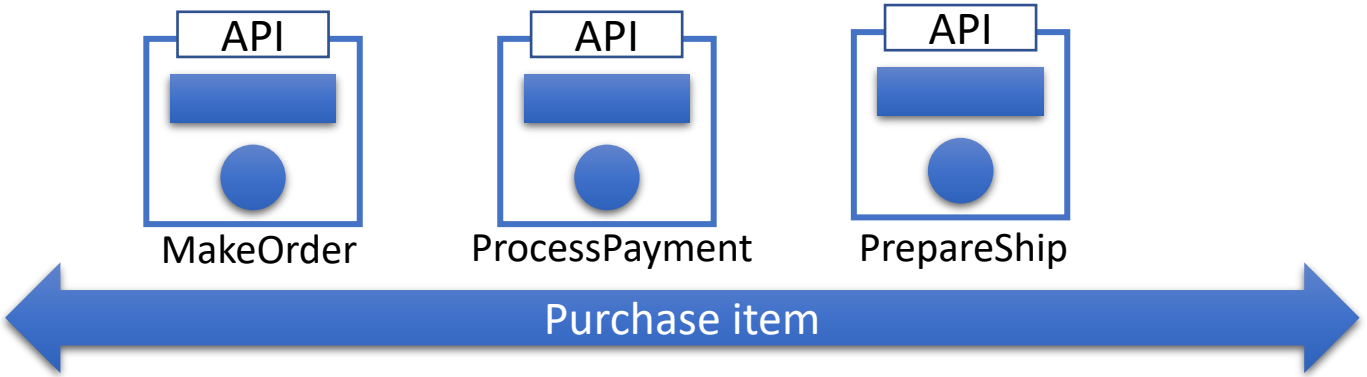
- A service has four properties according to one of many definitions of SOA:
  1. It logically represents a repeatable business activity with a specified outcome.
  2. Autonomous self-contained and accessible
  3. A black box for its consumers, meaning the consumer does not have to be aware of the service's inner workings.
  4. May be composed of other services

SOA components

Services



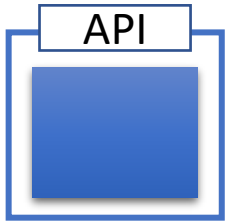
- A service has four properties according to one of many definitions of SOA:
  1. It logically represents a repeatable business activity with a specified outcome.
  2. autonomous self-contained and accessible
  3. A black box for its consumers, meaning the consumer does not have to be aware of the service's inner workings.
  4. May be composed of other services
- Example of Services In online shopping application



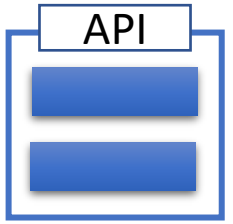
SOA components

Services classification

- Atomic services
  - Independent and not depend on other services

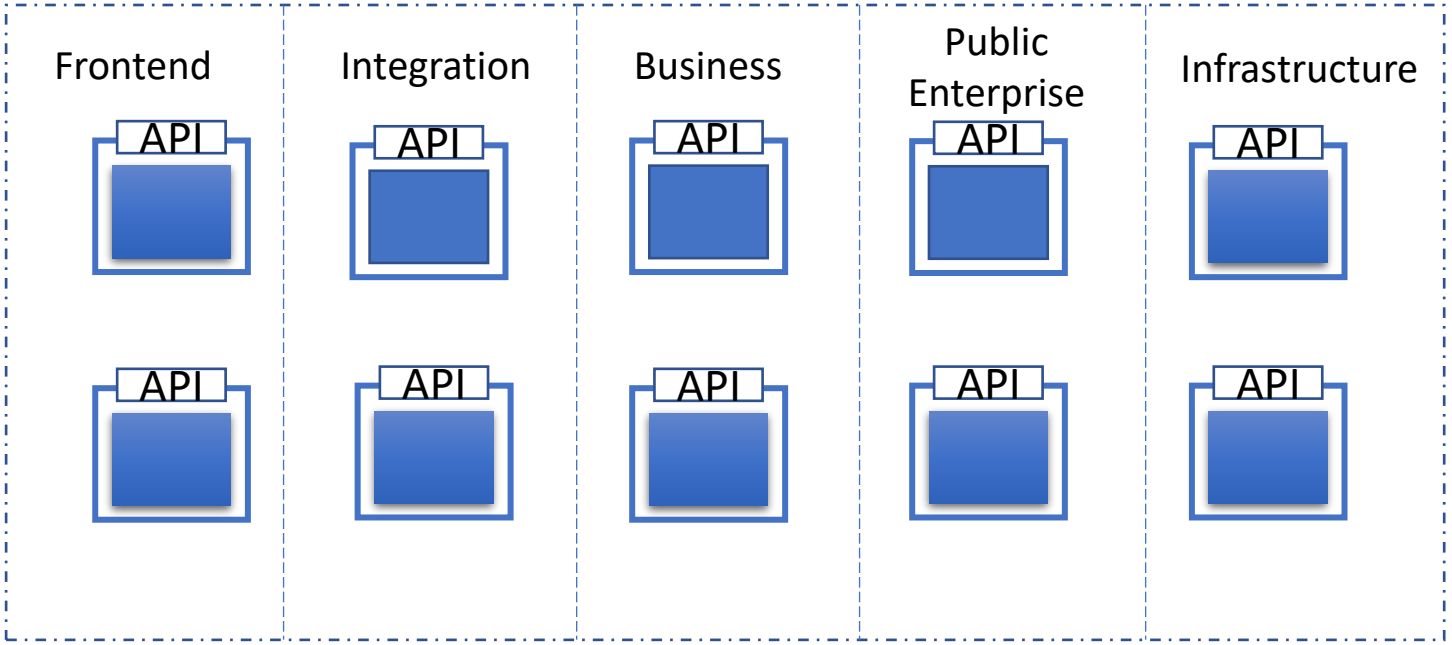


- Composite services
  - A fusion or composition of two or more atomic services. The services composing the composite service may depend on each other.



SOA components

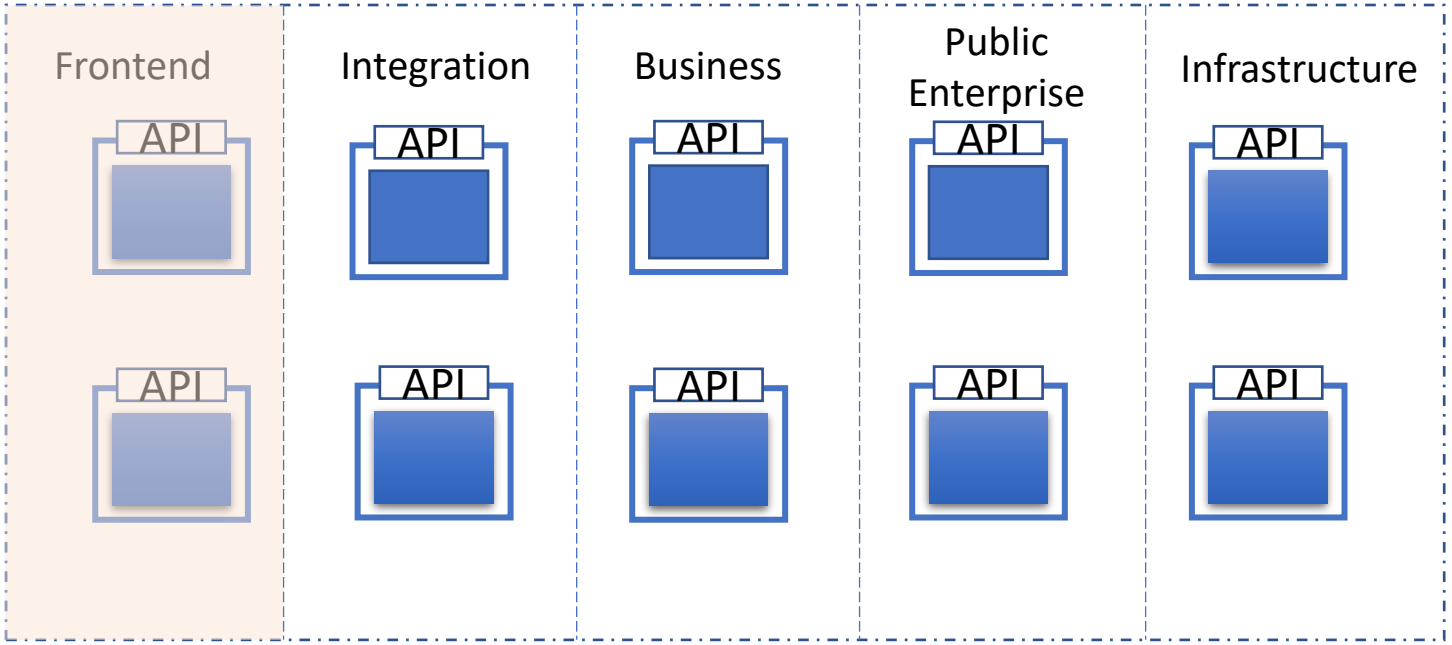
Services classification



SOA components

Services classification

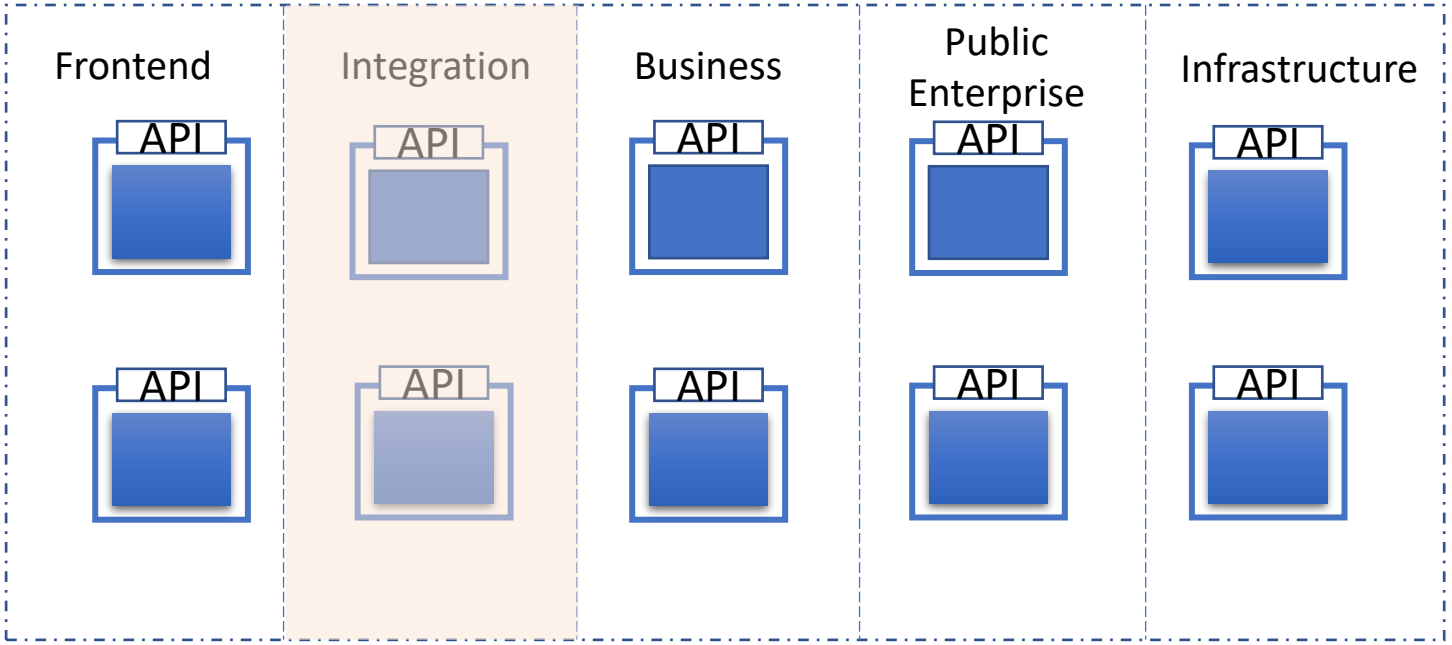
Front-end services are software components that initiate business processes and receive results. They are also called service consumers they could be end-user services. Example: checkout service



SOA components

Services classification

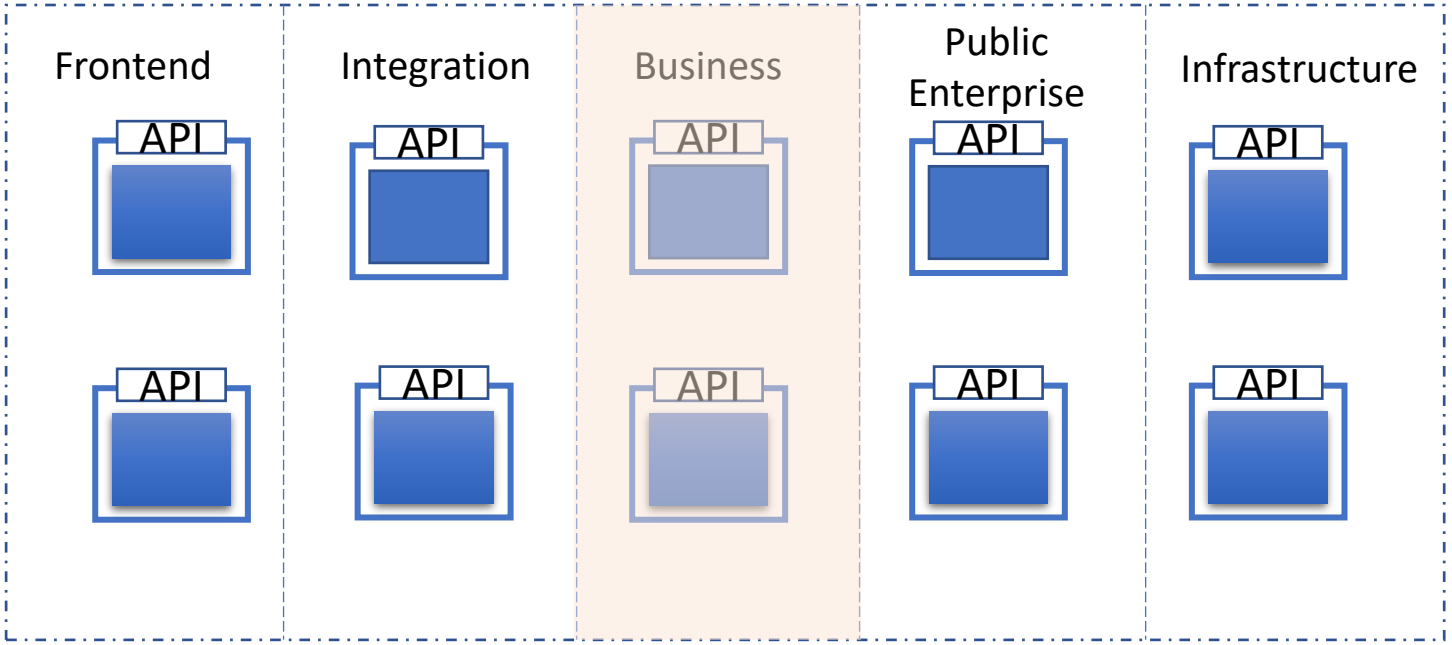
Integration services deal with issues related with the integration. They may include gateways, adapters and so on. For instance, it could convert data from a proprietary format used by one system to a standardized format for another.



SOA components

Services classification

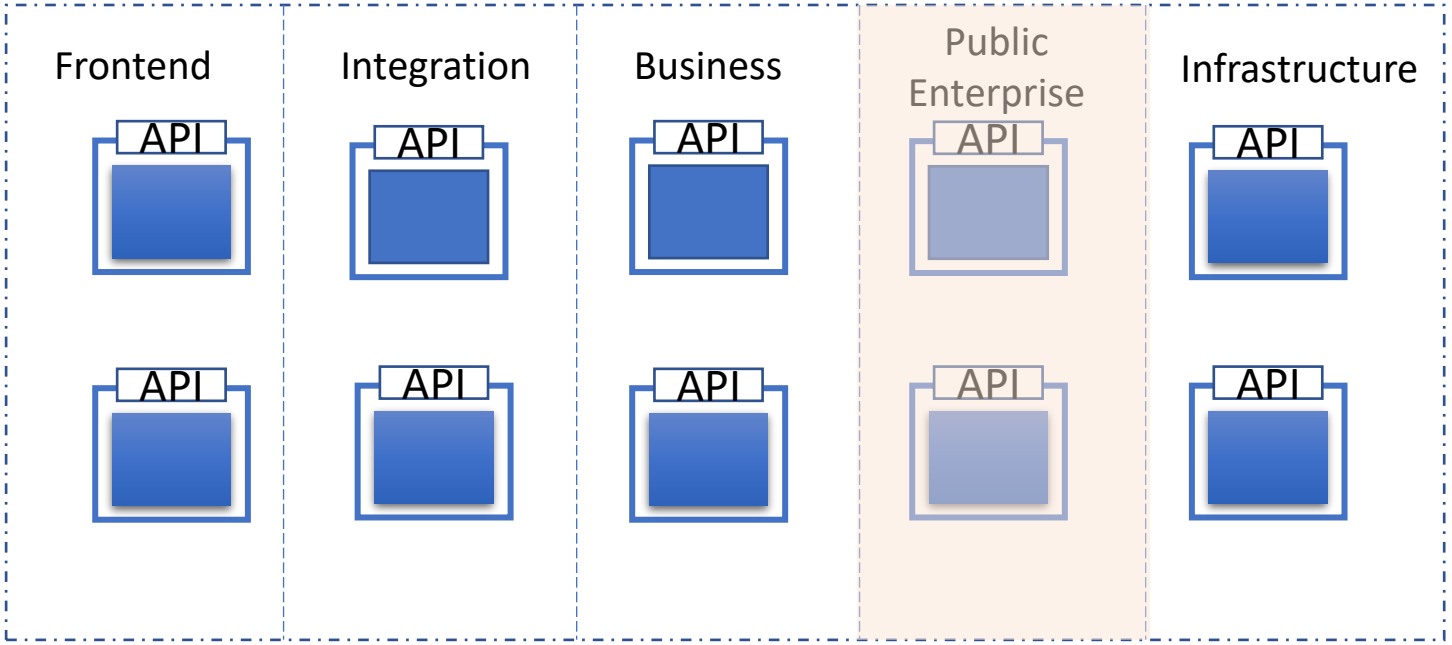
Business services comprise functionality directly related to the business domain. Example: order service.



SOA components

Services classification

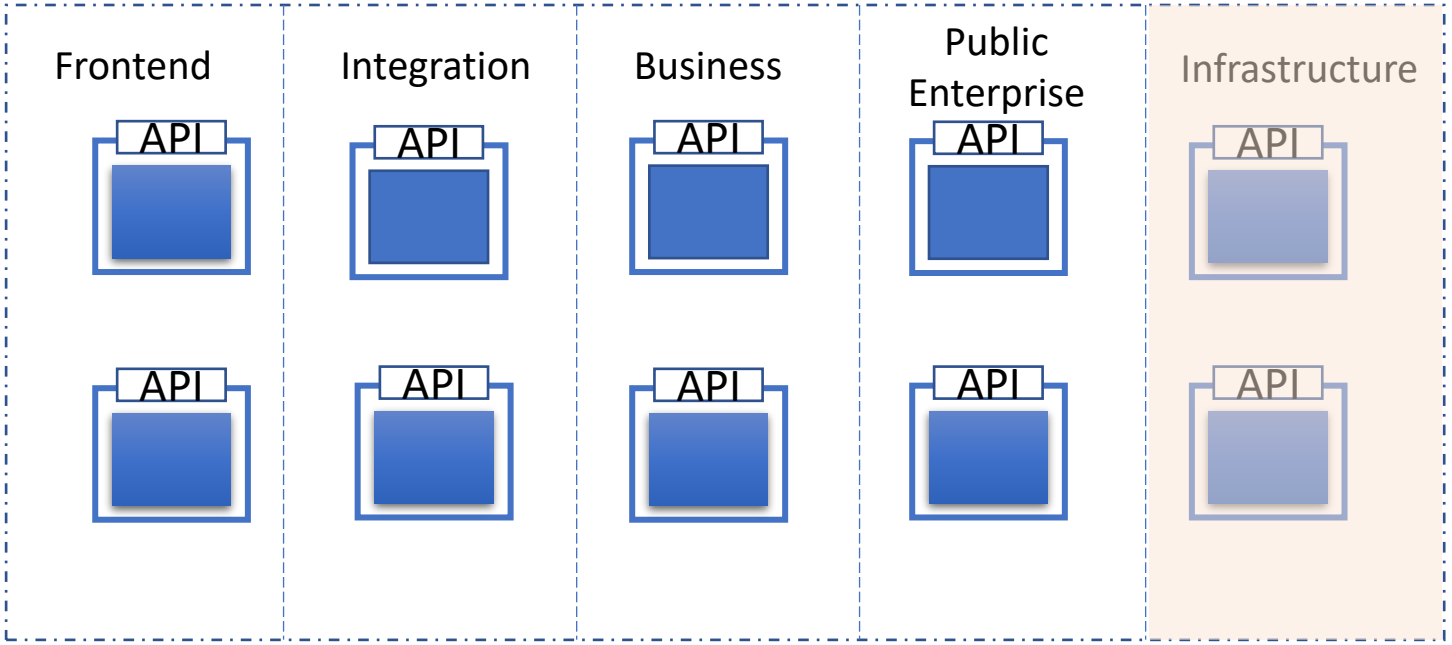
Public Enterprise Services are cross Enterprise Services that could be used nearly by all internal entities of the enterprise. They could also be used externally examples of such services are security services authentication services.



SOA components

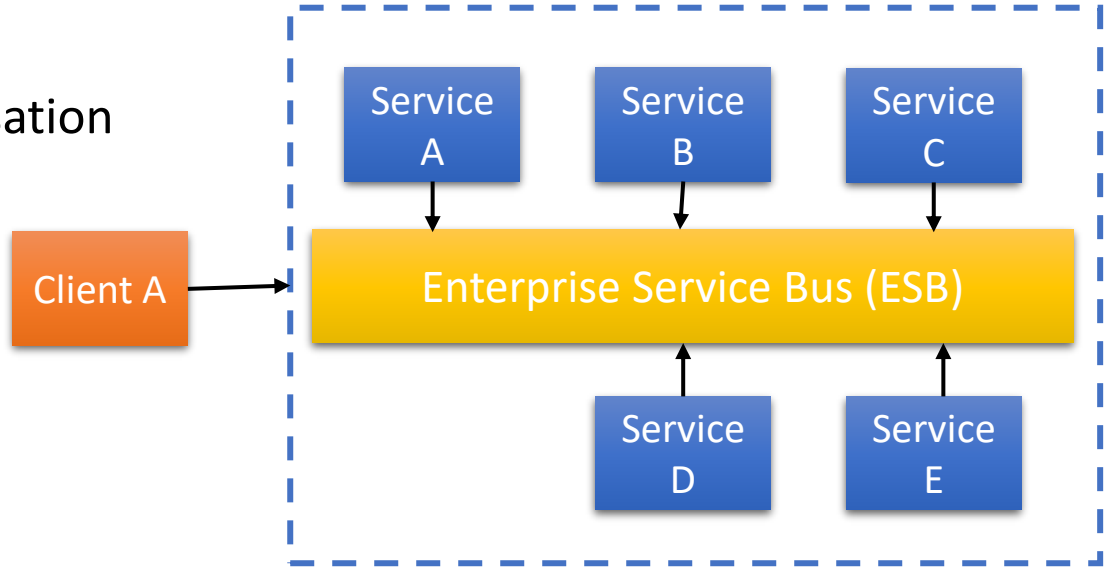
Services classification

Infrastructure services take care of basic functionality used by all parts of the system such as event login, exception handling.



SOA components

- Enterprise service bus (ESB) is the main middleware component for enabling service-to-service communications. In short, an ESB acts as a “lobby” where different people (services) can come to exchange information.
- Responsibility:
  - Providing connectivity
  - Data Transformation and message conversation
  - Routing
  - Security



SOA components

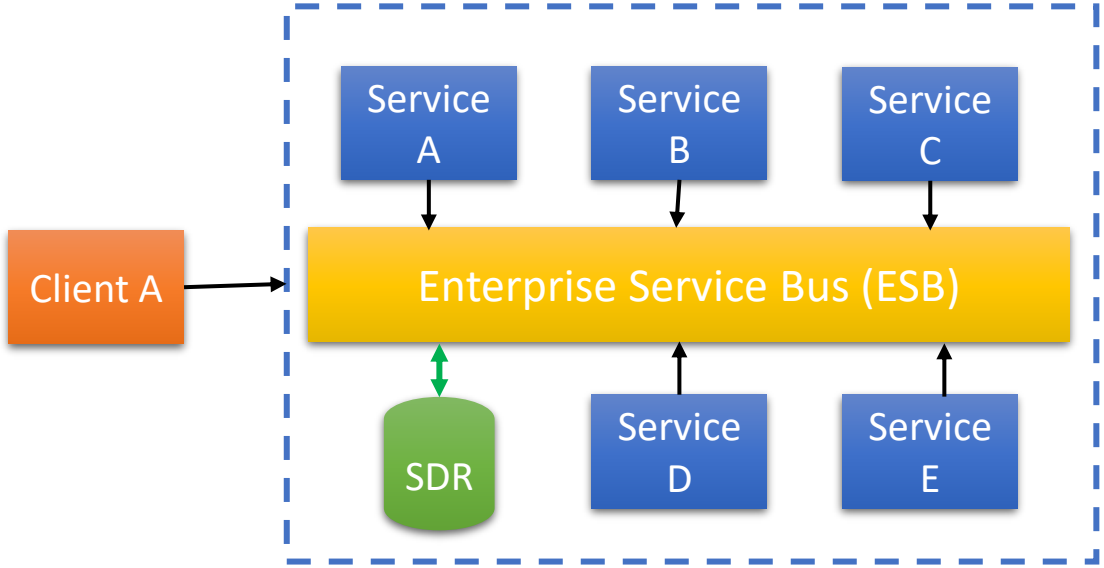
Benefits of ESB include:

- **Service communication:** It is an effective way to expose internal applications to their counterparts without building a point-to-point integration.
- **Protocol transformation and standardization**: If your services use different protocols (REST, SOAP, FTP, etc.), an ESB can act as a "converter."
- **Simplified transactional message flows**: It can be used to securely process messages between two or more heterogeneous transactional data sources.
- **Service orchestration**: It helps you create the optimal service hierarchy and ensure smooth routing.
- **Gateway to the World:** It can connect to the different services running in the other networks.

SOA components

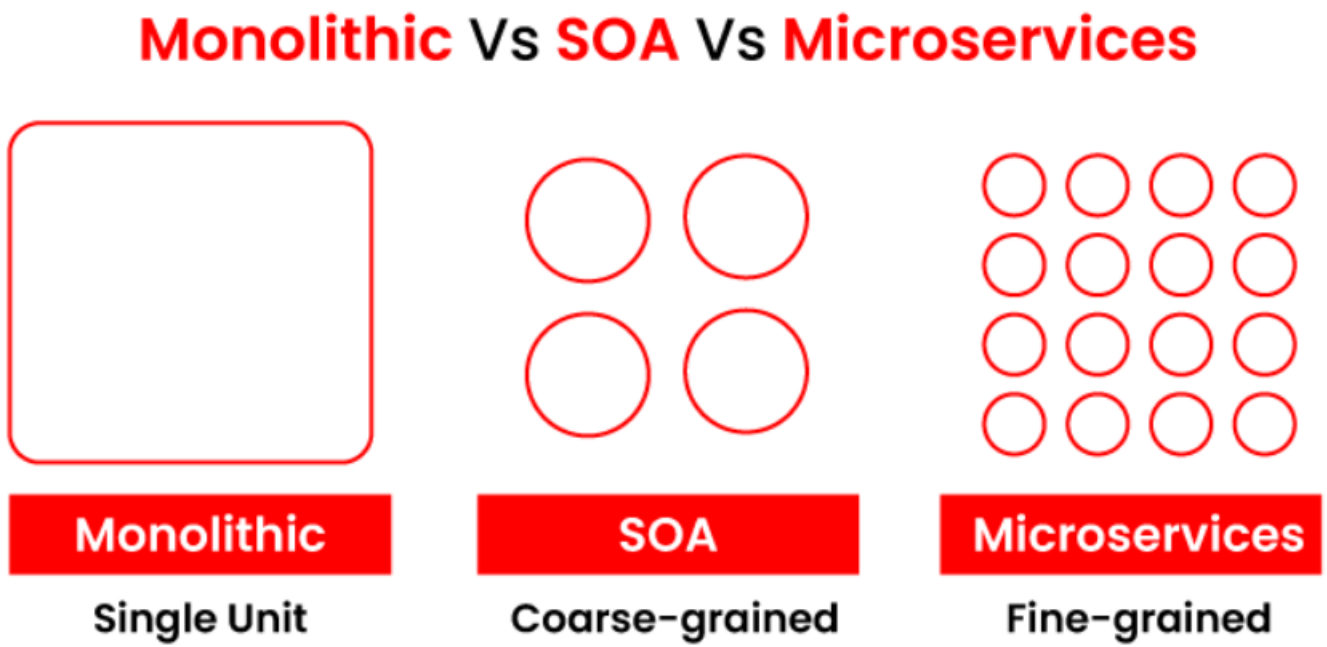
Service description repository/registry

- discover services for service consumers and learn the proper way to connect and communicate
- It becomes necessary when a SOA system grows large and complex and keeping track of available services and how to use them for service consumers and clients becomes harder



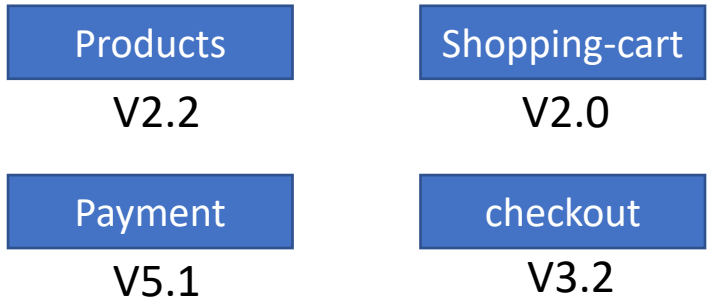
Microservices

Monolithic vs SOA vs Microservices



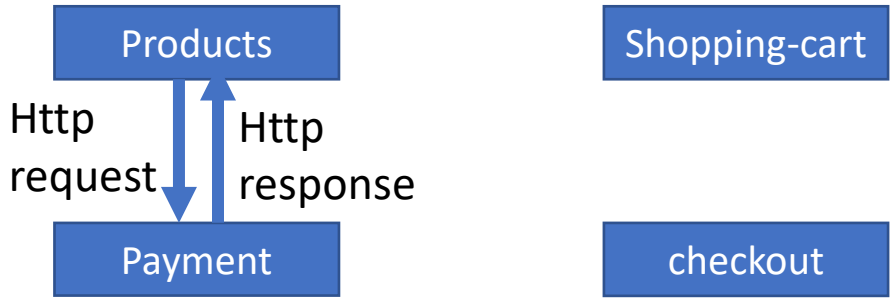
Microservices

- Split application into smaller and independent services based on the business functionality
- Each microservice is responsible for one specific job
- Self-contained and independent to deployed, develop and scaled separately
- Each microservices have individual and independent version
- Example:



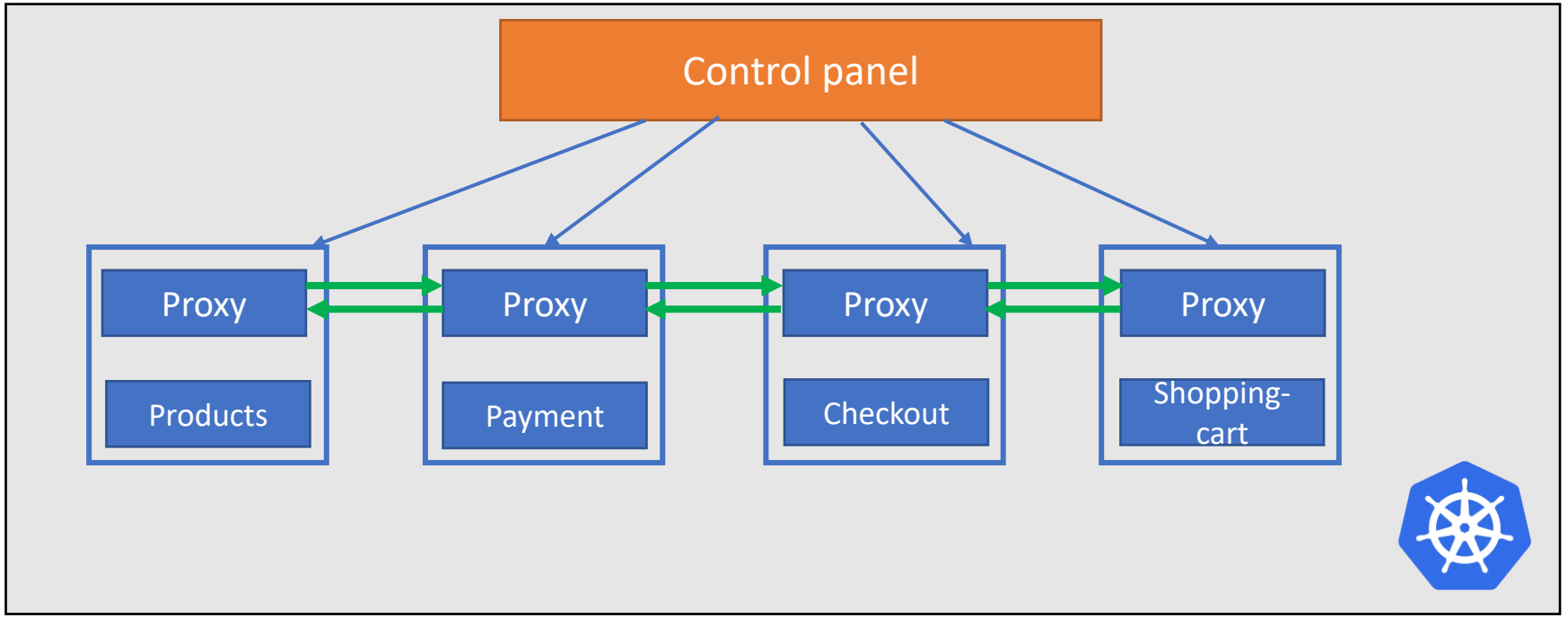
Communication between microservices

- Synchronous communication
  - Wait for response



Communication between microservices

- Communication with service mesh in Kubernetes



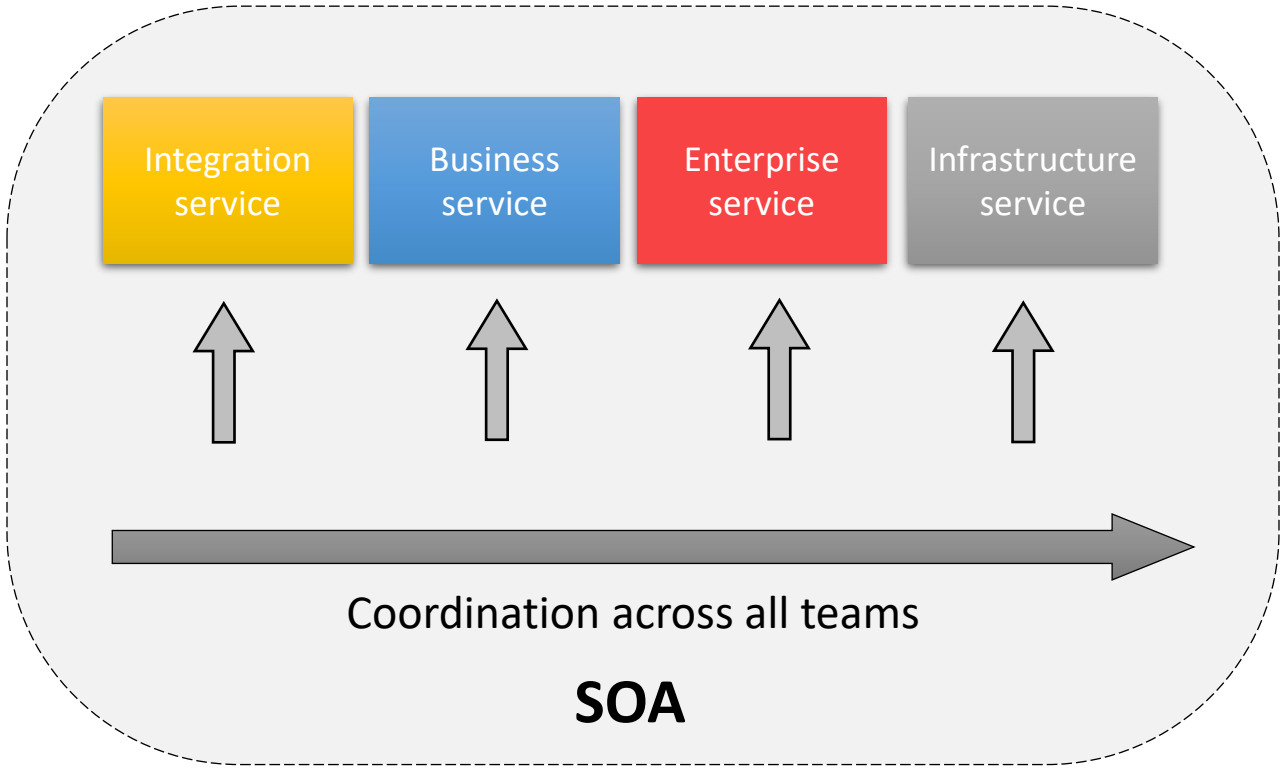
SOA vs microservices

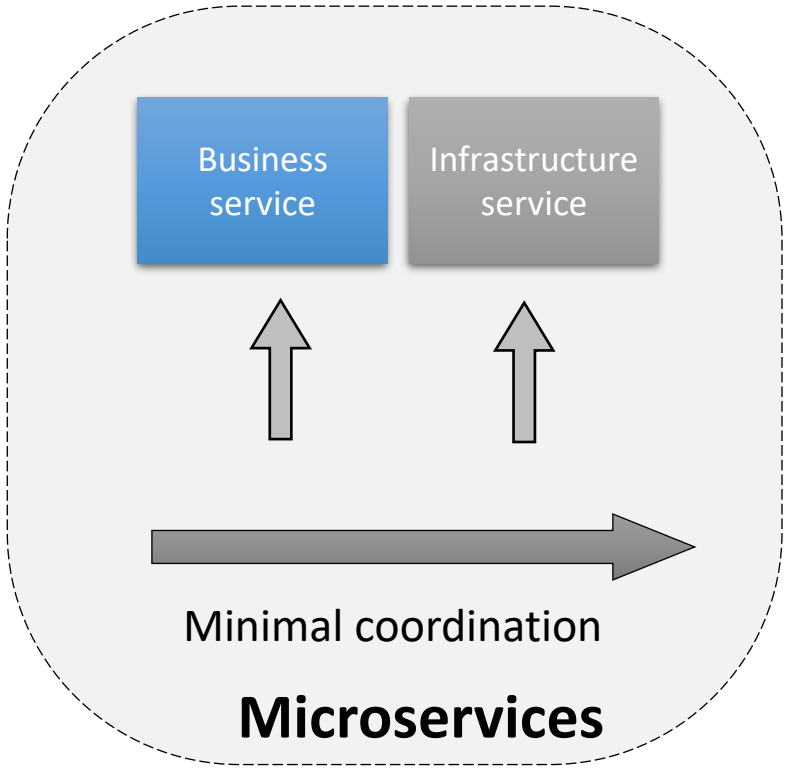
Architecture and coordination

SOA is like an orchestra where each artist is performing with their instruments while the music director guide them all.

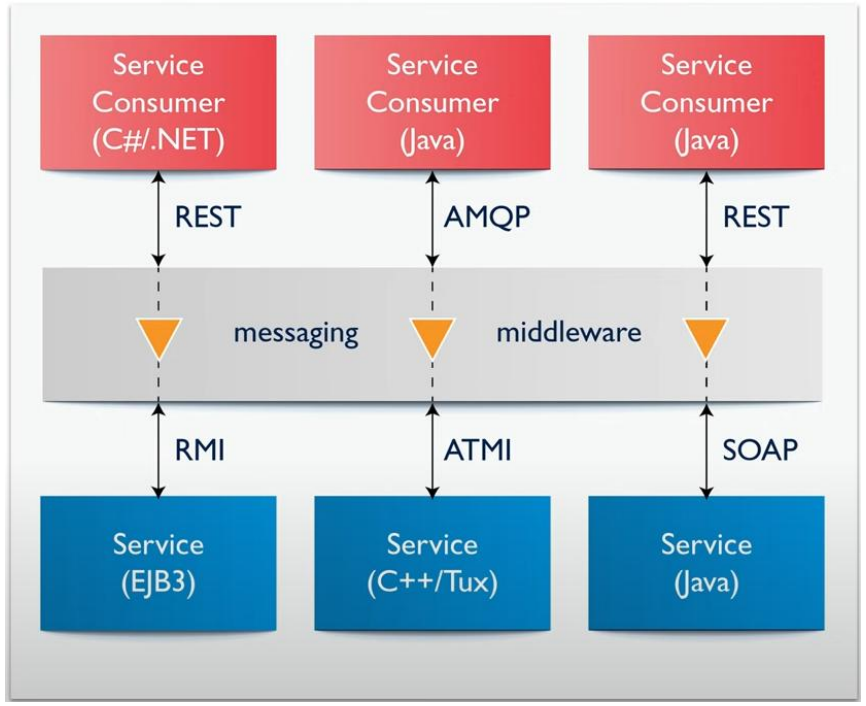
With microservices each dancer is independent and know what they need to do. If they miss some steps they know how to get back on the sequence.

Architecture and coordination

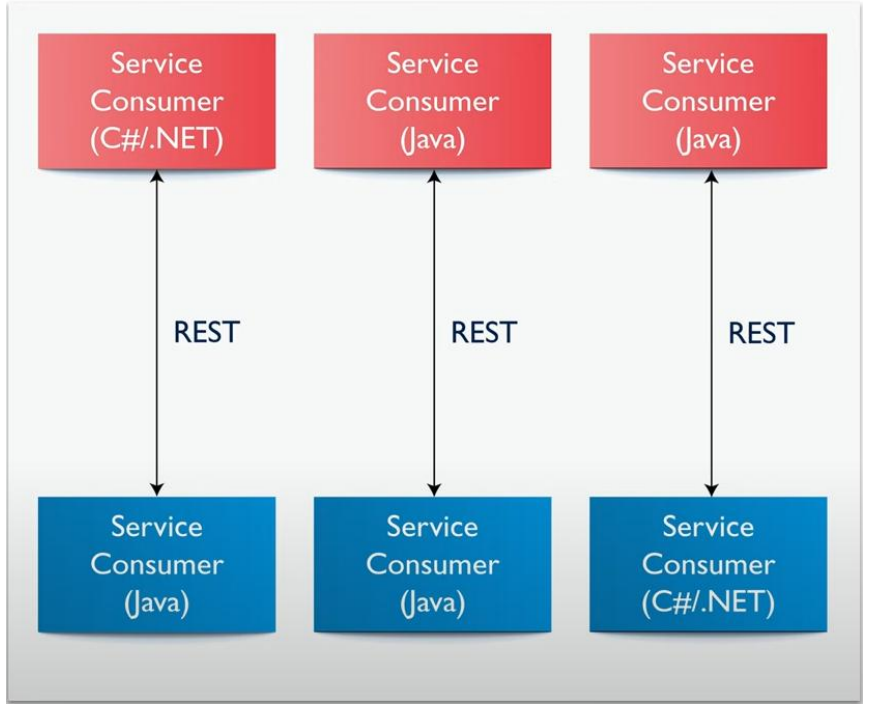




Communication between services

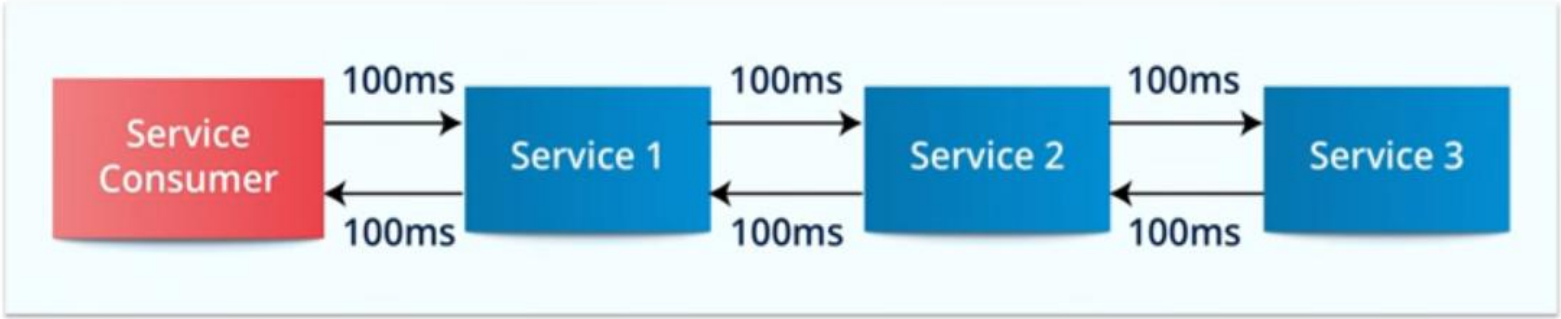


SOA

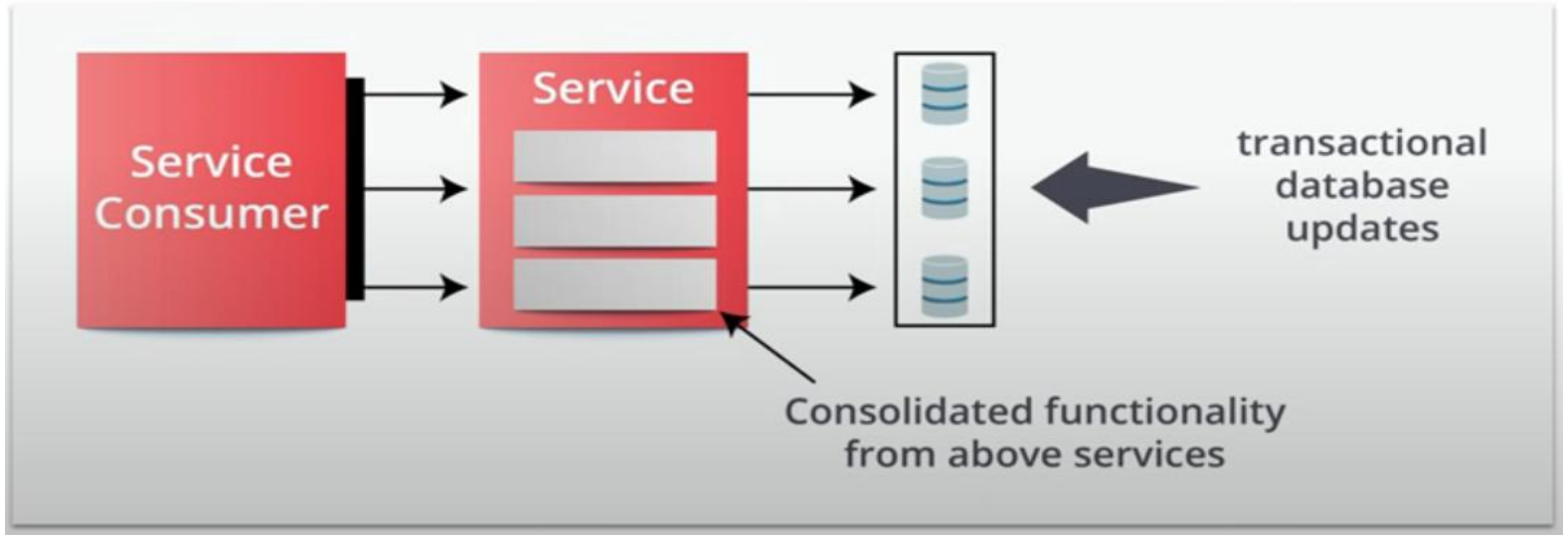


Microservices

Service granularity

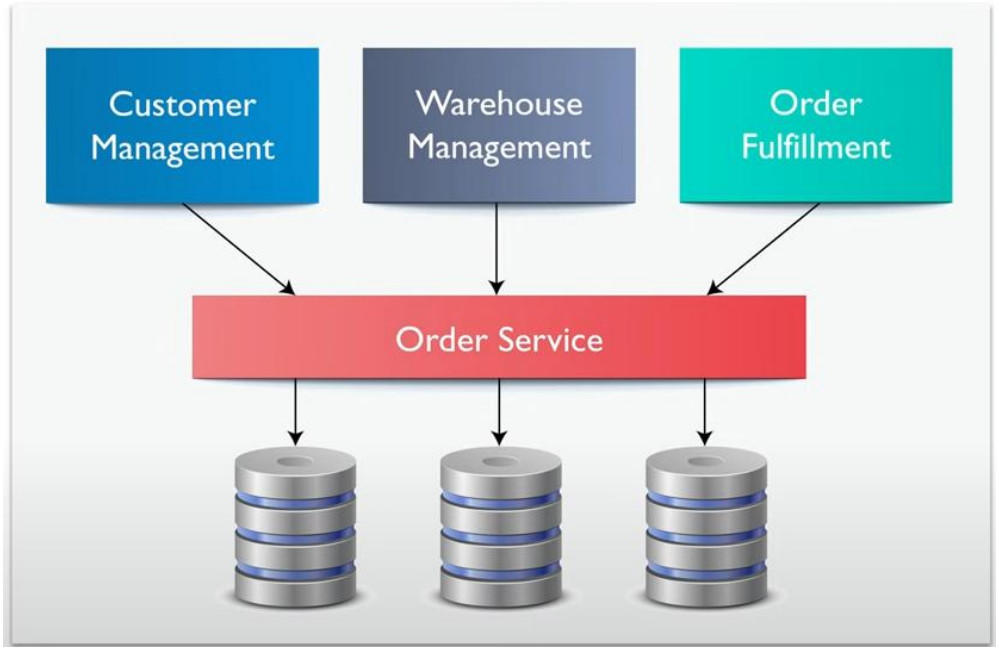


Microservices

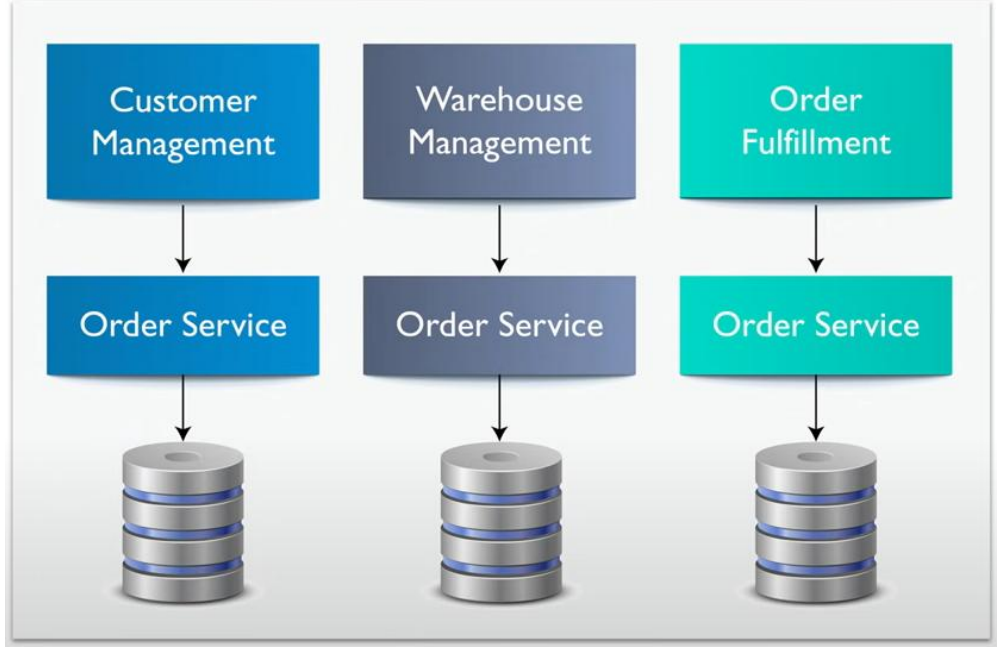


SOA

Component sharing



SOA



Microservices

SOA vs Microservices

SOA	MSA
Follows "**share-as-much-as-possible**" architecture approach	Follows "**share-as-little-as-possible**" architecture approach
Importance is on **business functionality** reuse	Importance is on the concept of "**bounded context**"
They have **common governance** and **standards**	They focus on **people, collaboration** and freedom of other options
Uses **Enterprise Service bus (ESB)** for communication	Simple messaging system
They support **multiple message protocols**	They use **lightweight protocols** such as **HTTP/REST** etc.
Multi-threaded with more overheads to handle I/O	**Single-threaded** usually with the use of Event Loop features for non-locking I/O handling
Maximizes application service reusability	Focuses on **decoupling**
Traditional Relational Databases are more often used	**Modern Relational Databases** are more often used
A systematic change requires modifying the monolith	A systematic change is to create a new service
DevOps / Continuous Delivery is becoming popular, but not yet mainstream	Strong focus on DevOps / Continuous Delivery

SOA and Microservices use cases

- SOA suited for:
  - large and complex business application environments
  - requiring integration with many heterogeneous applications

- Microservices suited for:
  - smaller and well-partitioned
  - web-based systems in which microservices give you much greater control as a developer

References

- Newman, Sam. *Building microservices*. " O'Reilly Media, Inc.", 2021
- Hwang, Kai, Jack Dongarra, and Geoffrey C. Fox (2013). *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan kaufmann.
- https://en.wikipedia.org/wiki/Service-oriented_architecture

universität innsbruck

Thank you for your attention☺

Dr. Zahra Najafabadi Samani

Email: Zahra.Najafabadi-Samani@uibk.ac.at