Middleware Architectures 1

Lecture 2: Service Architectures

doc. Ing. Tomáš Vitvar, Ph.D.

tomas@vitvar.com • @TomasVitvar • https://vitvar.com



Czech Technical University in Prague
Faculty of Information Technologies • Software and Web Engineering • https://vitvar.com/lectures





Modified: Sun Oct 06 2024, 20:22:07 Humla v1.0

Overview

- Service Definition
- Integrating Applications
- Integration Patterns

Service Views

Business view

A service realizes an effect that brings a business value to a service consumer
 → for example, to pay for and deliver a book

Conceptual view

 encapsulation, reusability, loose coupling, contracting, abstraction, discoverability, composability

Logical view

- service interface, description and implementation
- message-oriented and resource-oriented

Software architecture view

- business service (also application service)
 - → external, exposed functionality of an application
- middleware service
 - → internal/technical, supports processing of requests

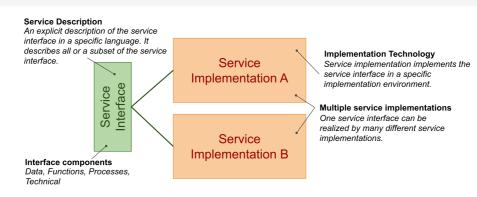
• Technology architecture view

- REST/RESTful, GraphQL
- XML-RPC/SOAP, RMI, gRPC
- WohSacket WohRTC SSF

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

- 3 -

Interface, Description and Implementation



Terminology clarification

- service ~ service interface + service implementation
- WSDL service ~ service description in WSDL language
- SOAP service ~ a service interface is possible to access through SOAP protocol; there is a WSDL description usually available too.
- REST/RESTful service ~ service interface that conforms to REST architectural style and HTTP protocol
- Microservice ~ a set of services that realize an app's capability

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

Service Interface

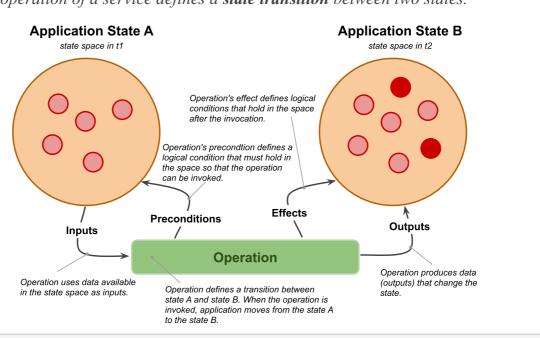
- Service interface components
 - Data
 - → Data model definition used by the service
 - \rightarrow for example, input and output messages, representation of resources
 - Functions
 - → operations and input and output data used by operations
 - Process
 - \rightarrow public process: how to consume the service's functionality
 - Technical
 - → security, usage aspects (SLA-Service Level Agreement)
 - \rightarrow other technical details such as IP addresses, ports, protocols, etc.

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

- 5 -

Public Process

- A state diagram
 - operation of a service defines a **state transition** between two states.



Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

Service Characteristics

Loose Coupling The requester agent's implementation is independent from service usage. That is, there is no "hard-wired" Contracting knowledge required to use the service. The service interface is a contract between the requester and the provider. They both agree to follow the service description in order to achieve Requester interoperability. Agent A Implementation Abstraction Service interface is abstracted from Reusability underlying service implementation as well as The service can be used in many different all software and hardware technology. Provider Agent X scenarios by different requester agents that Implementation are unforeseen during the service design. Service Interface Requester Agent C Implementation Requester Agent B Implementation Provider Agent Y Discoverability Implementation Requester can discover the service interface and decide how to use it. Encapsulation Composability The provider agent implementation is hidden to the It is possible to compose services into more complex requester agent accessing the service. The requester agent only knows the service interface to consume its processes. Such processes can again be accessed as services. functionality. Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

Overview

- Service Definition
- Integrating Applications
- Integration Patterns

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

Integration and Interoperability

Integration

- A process of connecting applications so that they can exchange and share capabilities, that is information and functionalities.
- Includes methodological approaches as well as technologies

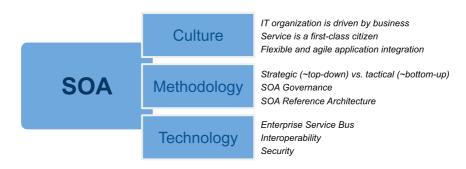
Interoperability

- Ability of two or more applications to understand each other
- Interoperability levels
 - \rightarrow Data syntax/structure and semantics
 - → Functions/Processes syntax and semantics
 - \rightarrow Technical aspects protocols, network addresses, etc.

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

- 9 -

Service Oriented Architecture



SOA supports two core business strategies

- Growing top-line revenue
 - → Enterprise reacts quickly to requirements from the market
 - → Business processes can be reconfigured rather than reimplemented
- Improving bottom-line profit
 - → Saving development costs by resuing existing services

Pre-integrated solutions

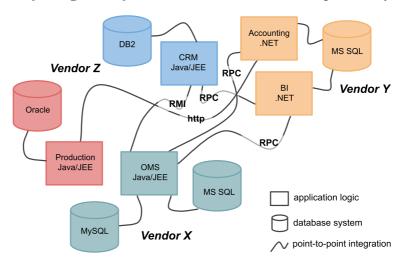
- Out-of-the-box applications and integration solutions among them

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

– 10 –

One-to-One Service Integration

- Direct integration of applications
 - Multiple protocols problem, multiple vendor problem
 - Replication of integration functionalities such as interoperability solutions

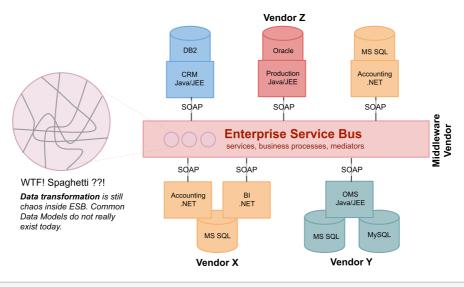


Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

- 11 -

Many-to-Many Service Integration

- Enterprise Service Bus central integration technology
 - Realizes so called Service Oriented Architecture (SOA)
 - Contains various integration components such as process server, mediators, messaging middleware, identity management, etc.



Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

- 12 -

Integration Approaches Overview http App Client A Client B Client C Server A REST, Service-oriented integration (M:N) SOAP, http http Application A integrates with application B RPC, ... and D through a middleware process REST SOAP RPC, App App Server B Server C SOAP, RPC, .. SQL Service-oriented integration (1:1) Application B integrates with application C directly by using services. SQL Database App Client D Server D Data-oriented integration SQL Application D integrates with application B through database B. **Database** Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar - 13 -

Data-oriented Integration



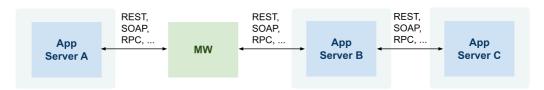
• Third-party database access

- Application D accesses a database of application B directly by using SQL and a knowledge of database B structure and constraints
- In the past: monolithic and two-tier client/server architectures
- Today: ETL (Extract, Transform, Load) technologies

Problems

- App D must understand complex structures and constraints
 - → Data very complex, includes structure and integrity constraints
 - → Functions/processes hidden in integrity constraints
 - → Technical access mechanisms can vary

Service-oriented Integration



- Integration at the application layer
 - Application exposes services that other applications consume
 - Services hide implementation details but only define interfaces for integration
- Problems
 - Can become unmanageable if not properly designed
 - Interoperability
 - \rightarrow Data limited to input and output messages only
 - → Functions/processes limited to semantics of services
 - → Technical access mechanisms can vary

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

- 15

Integration and Types of Data

- Real-time data Web services
 - Service-oriented integration
 - online, realtime communication between a client and a service
 - Usually small data and small amount of service invocation in a process
- Bulk data ETL
 - Data-oriented integration
 - processing of large amount of data in batches
 - Sometimes required for reconciliation across apps
 - → when real-time integration fails and there is poor error handling
- SOA provides both Web service and ETL capabilities

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

Enterprise Service Bus

- ESB is a central intermediary in SOA
 - Types of services: shared and infrastructure
 - Types of processes: Technical and Business
- ESB Application
 - Application running on an application server
 - Exposes functionality via Web service interface
 - Allows to communicate with various messaging protocols
- Middlware Integration Patterns
 - Technical-level interoperability message broker
 - Location transparency
 - Dynamic routing
 - Session pooling
 - Message enrichment
 - Data transformation
 - Key mapping

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

_ 17 _

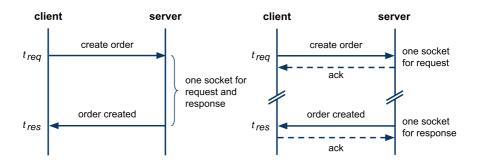
Overview

- Service Definition
- Integrating Applications
- Integration Patterns
 - Synchronous and Asynchronous Integration
 - Microservices Architecture

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

– 18 –

Synchronous and Asynchronous Integration



Synchronous

- one socket, $|t_{reg} t_{res}|$ is small
- easy to implement and deploy, only standard firewall config
- only the server defines endpoint

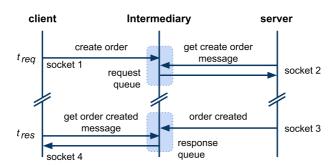
Asynchronous

- request, response each has socket, client and server define endpoints
- $-|t_{reg}-t_{res}|$ can be large (hours, even days)
- harder to do across network elements (private/public networks issue)

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

- 19 -

Asynchronous via Intermediary



Intermediary

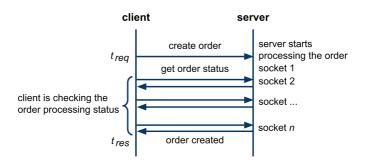
- A component that decouples a client-server communication
- It increases reliability and performance
 - \rightarrow The server may not be available when a client sends a request
 - \rightarrow There can be multiple servers that can handle the request

Further Concepts

- Message Queues (MQ) queue-based communication
- Publish/Subscribe (P/S) event-driven communication

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

Asynchronous via Polling



- Polling only clients open sockets
 - A client performs multiple request-response interactions
 - \rightarrow The first interaction initiates a process on the server
 - → Subsequent interactions check for the processing status
 - \rightarrow The last interaction retrieves the processing result
- Properties of environments
 - A server cannot open a socket with the client (network restrictions)
 - Typically on the Web (a client runs in a browser)

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

- 21 -

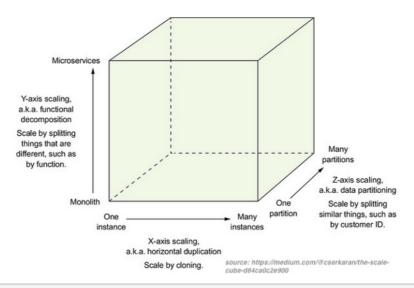
Overview

- Service Definition
- Integrating Applications
- Integration Patterns
 - Synchronous and Asynchronous Integration
 - Microservices Architecture

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

The Scale Cube

- Three-dimensional scalability model
 - X-Axis scaling requests across multiple instances
 - Y-Axis scaling decomposes an application into micro-services
 - Z-Axis scaling requests across "data partitioned" instances



Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

- 23 -

Overview

- Emerging software architecture
 - monolithic vs. decoupled applications
 - applications as independenly deployable services

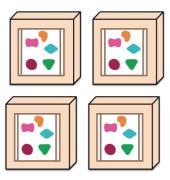
A monolithic application puts all its functionality into a single process...



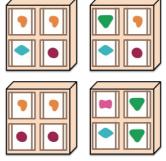
A microservices architecture puts each element of functionality into a separate service...



... and scales by replicating the monolith on multiple servers



... and scales by distributing these services across servers, replicating as needed.



Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

– 24 -

Major Characteristics

- Loosely coupled
 - Integrated using well-defined interfaces
- Technology-agnostic protocols
 - HTTP, they use REST architecture
- Independently deployable and easy to replace
 - A change in small part requires to redeploy only that part
- Organized around capabilities
 - such as accounting, billing, recommendation, etc.
- Impplemented using different technologies
 - polyglot programming languages, databases
- Owned by a small team

Lecture 2: Service Architectures, CTU Winter Semester 2024/2025, @TomasVitvar

- 25 -