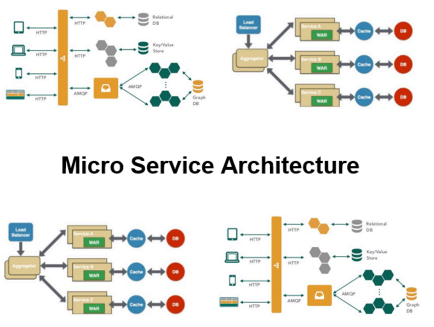
Microservices Architecture

Duration: 3 days **Price: € 1999**

Open Schedule


Micro Service Architecture

Audience Course Microservices Architecture

The course Microservices Architecture course is intended for developers, architects and others who want to understand the why and how of a Microservices Architecture.

Prerequisites training Microservices Architecture

In order to participate in the course Microservices Architecture, general knowledge of software development and software design is desirable. Previous knowledge of modern programming languages is beneficial to understanding.

Realization course Microservices Architecture

The theory is discussed on the basis of presentations. Illustrative demos are used to clarify the concepts. There is ample opportunity to practice and theory and practice are interchanged. Course times are from 9:30 to 16:30.

Certification Microservices Architecture

After successful completion of the course participants receive an official certificate Microservices Architecture.

Content Course Microservices Architecture

In the course Microservices Architecture participants learn the advantages of developing applications in a Microservices Architecture compared to a Monolithic Architecture. In the Microservices Architecture, small teams are responsible for development and deployment of the microservices, each of which can have its own database and user interface.

Intro Microservices

The course Microservices Architecture starts with a discussion of the problems of monolithic applications. Subsequently the characteristics of a Microservices Architecture such as independent deployability, independent scaling and separate databases are discussed.

Design Considerations

Next it is explained how according to Conway's law the architecture of applications is reflected in the organizational structure. The switch to a Microservices Architecture therefore requires adjustments in the organization. The principles of Domain Driven Design are also treated.

Interprocess Communication

Microservices are separate processes and communicate via interprocess communication. Attention is paid to synchronous communication via REST, asynchronous communication via messaging and communication via a binary protocol such as Thrift.

Micro Frontends

Then it is time for a discussion of the user interfaces of Microservices. Various approaches for integrating a user interface are discussed such as custom elements, server side templates and built-time integration of JavaScript libraries.

Data Management

Data management in a Microservices Architecture is also covered. The different patterns for data storage such as database per service, shared database, the saga pattern and event sourcing are discussed in this respect.

Discovery and Deployment

Finally options for service discovery are covered, such as client and server side discovery and the use of service registrars. Deployment options of microservices such as in virtual machines and in containers are also treated.

SpiralTrain
developer courses

Modules Course Microservices Architecture

Module 1 : Intro Microservices	Module 2 : Design Considerations	Module 3 : Interprocess Communication
Microservices Architecture	Conways's Law	Communicating Processes
Monolithic Applications	Law as Enabler	Interaction Styles
Software Monolith	Domain Driven Design	Service Communications Styles
Problems of Monoliths	Building Blocks	Defining API’s
Layered Architecture	Bounded Context	Netflix Hystrix
Growing Beyond Limits	Reactive Manifesto	Asynchronous Messaging
Microservices Characteristics	Reactive Microservices	Messaging Models
Underlying Principles	Microservices with UI	Publish and Subscribe
Independent Deployability	Microservices Benefits	Point to Point
Independent Scaling	Effective Modularization	Advantages of Messaging
Separated Databases	Replaceability Microservices	Synchronous IPC
Size of Microservices	Continuous Delivery Pipeline	REST Services
Frontend Monolith	Free Technology Choice	Resource URI Access
Micro Frontend Architecture	Team Independence	Thrift
Blurry Service Boundaries	Microservices and Languages	Content Negotiation
Module 4 : Micro Frontends	**Module 5 : Reactive Microservices**	**Module 6 : Data Management**
What are Micro Frontends?	What is Reactive?	Distributed Data
Monolith versus Micro Frontends	Reactive Programming	Complex Data Access
Benefits Micro Frontends	Reactive Extensions	Polyglot Architecture
Delivery Pipeline per Service	Observables	Private Access
Avoid Shared Artifacts	Reactive Systems	Design Patterns
FrontEnd Integration	Elasticity and Resilience	Database per Service
ESI Composition	Reactive Microservices	Shared Database Pattern
Links and JavaScript	Asynchronous Development	Saga Pattern
Custom Elements	Event Loop	Event Publishing
Integration Approaches	Reactor Pattern	Consuming Events
Server Side Templates	Multireactor Pattern	Responding to Events
Built Time Integration	Verticles	Base Model Transactions
Shared Component Libraries	Callbacks vs Observables	Local Transactions
Cross Application Communication	RxJava API	Database Transaction Log
Backend Communication	Monitoring	Event Sourcing
Module 7 : Service Discovery	**Module 8 : Deployment Strategies**	**Module 9 : Security**
Why Service Discovery	Deployment Patterns	Microservices Security
Finding Services	Virtual Machines	Security Challenges
Client Side Discovery	Creating Virtual Machines	Key Security Fundamentals
Service Registry	Drawbacks of VM’s	Confidentiality
Server Side Discovery	Containers	Edge Security
Load Balancers	Containers versus VM’s	Securing with OAuth2
Service Registries	Container Orchestration	Authorization Server Interactions
Self Registration Pattern	Kubernetes	Actors OAuth2.0 Flow
Third Party Registration Pattern	Multiple Instances Per Host	OAuth Roles
Service Registrars	Service Instance per Host	Application Registration
Netflix Eureka	Service Instance per Container	Securing with API Gateway
HashiCorp Consul	Serverless Deployment	Zuul Proxy and OAuth2

SpiralTrain BV
Standerdmolen 10, 2e verdieping
3995 AA Houten

info@spiraltrain.nl
www.spiraltrain.nl
Tel.: +31 (0) 30 – 737 0661

Locations
Houten, Amsterdam, Rotterdam, Eindhoven, Zwolle, Online