

MICROSERVICES

...so what is Microservice?

- It's approach to developing application, where you split application to suite of small services, each running in its own independent space and communicating with mechanism like REST, SOAP or RPC

Characteristics of Microservice

- **Modeled around business domain**
- **Single purpose**
- **Simple, well defined interface**
- **Modular**
- **Independent**
- **Lack of centralized, shared database**
- **Independently deployable**
- **100% CI & monitoring**

What is not microservice

- **A service exposed via API by another party**

Why to use microservice arch.

- **Easy to create new projects based on existing stuff**

Why to use microservice arch.

- **Agility - deliver new features in high quality quicker with less coordination**

Why to use microservice arch.

- **Regular deploys, even multiple deploys per day if required**

Why to use microservice arch.

- **Quality of product and high UP times**

Why to use microservice arch.

- **Easy to scale team**

Why to use microservice arch.

- **Performance**

Disadvantages of microservices

- **CI/CD - absolute must, whole process of deployment need to be fully automated.**
- **Network latency**
- **Handling of partial failures**
- **Need to code of inter-process communication between microservices**

Benefits of using this architecture

- **Reusability**
- **Ideal for Agile team setup**
- **Independent feature deployment**
- **Deliver new features fast**
- **Scalability**
- **Easy to scale teams**
- **Programming language diversity**
- **Easier to test**
- **Quick time to market**

Who is using Microservices

- **Microservices becoming the default style for building enterprise applications.**
- **Ebay, Twitter, Amazon, Paypal, Google, Uber, Autoscout.de, Zalando ...**
- **They all started as monolith**

And what about Frontend?

- **frontend application is divided into components**
- **each component is developed independently**
- **Communication over custom events & parameters**
- **Final assembly is done over server-side includes like ESI tags or in browser over Web Components**

Why microfrontends?

- **Same benefits like any microservices**
- **But most important is REUSABILITY**

MicroFrontend + WebComponents

- **One of the most important innovation happening on the browser side today**
- **Downside of this approach is current browser support...but polyfill exist to make it work on all modern browsers**

Web Components frameworks

- **They bring us browser compatibility and framework benefits out of the box:**
 - Polymer
 - Stencil
 - SkateJS



Any questions?