

Scalable Architecture using microservices and DDD

Kort om undervisningen

Microservices har de seneste år gået sin sejrsgang, fordi man med microservices, kan lave store og komplekse backend systemer som forbliver agile. Når microservices anvendes godt, opnås fordele både for den enkelte udvikler, som opnår et effektivt og gnidningsfrit udviklingsmiljø; for systemet, som per design er robust og skalerbart; og for forretningen, som får mulighed for at opnå stor agilitet gennem continuous delivery af microservices.

Indhold

- What are microservices?
- Microservice vs monolith
 - Characteristics, advantages and disadvantages of monoliths - Why did monoliths get a bad - reputation - Coupling and scalability issues - Splitting a monolith into microservices - Services != decoupled design*
- The 11 fallacies of distributed computing
- Challenges with Remote calls
 - Essential vs accidental complexity when performing request/response style service orchestration - Distributed Transactions
- 1st exercise
 - Hotel frontend “search for room availability”
- Life beyond distributed transactions
 - Service autonomy - DDD: Aggregate/Entity/Value-Object introduction
- Introduction to Events
- The coupling matrix
- How to discover events (event storming / event modeling)
- Business Processes and eventual consistency
 - Event Driven Architecture (EDA) - Choreography & Process Managers - CAP theorem - ACID vs BASE
- Event integration approaches (push vs pull)
 - Internal vs External events
- We need to fight complexity
 - Cost of software development - Alignment trap, maintenance zone, Well oiled IT, IT enabled growth (Bain Analysis) - Neil Nickolaisen Purpose Alignment Model - Wardley maps
- Service and domain boundaries
 - Bounded Contexts and Business Capabilities - Use cases and task based UI's - Focus on nouns and not on entities - The color game - Align with business capabilities - Conway's law
- Service design principles
 - A service a logical boundary - Where do microservices fit in? - Service deployment (applications vs layers vs

- tiers) - What about security? - Autonomous Components - Ports & Adapters / Hexagonal architecture / Onion architecture
- Application vs Services
- UI (Micro frontends / Composite UI's)
 - Backend for a Frontend (BFF) - Backend vs Frontend Composition - Query across services
- 2nd exercise
 - Hotel frontend “search for room availability” - Hotel frontend “book a room” (if time permits)
- CQRS + Event Sourcing
 - Principles - Projections - Examples - Event replay
- Microservice/EDA architecture building blocks (if time permits)
 - Broker based vs Federated - Publish/Subscribe (Topics) - Client Handled Subscriptions - Durable Queue - Error Handling (Error Queues, Redelivery strategy) - Notifications - Event Monitoring

Forudsætninger

Der er ingen forudsætninger. Det er en fordel, hvis du har erfaring med at udvikle software i et par år.

Målgruppe

Arkitekter, der arbejder med distribueret arkitektur. Udviklere der arbejder eller ønsker at arbejde med microservices.

Efter kurset kan deltageren

Kommende afholdelsesdatoer

Ingen planlagte datoer, anvend kontakttinformationerne nedenfor.

Oplysning om yderligere afholdelser findes på vores [hjemmeside](#). Andre spørgsmål besvares meget gerne ved brug af vores [kontaktformular](#) eller på telefon (+45) 33 861 861