



## Kursus:

### *Modernisering – Fra Monolit til Microservice arkitektur*

Den klassiske trelagsarkitektur med de velkendte tre lag: præsentation i et klient lag, en applikationsserver, og et databaselag; har mange fordele og betegnes ofte som en *monolit*. Imidlertid er den blevet udfordret af *microservice arkitekturen*, som er karakteriseret ved at mange selvstændigt idriftsatte services, hver med sin egen database og tilknyttede udviklerteam, samarbejder om at levere systems overordnede funktionalitet til kunderne. Fordelene er mange: teams arbejder på en specifik delmængde af systemet, hvilket giver fremdrift og ejerskab; systemet kan løbende fejlrettes og opdateres (continuous deployment) uden at "hele systemet" skal lukkes ned og atter idriftsættes; nye ønsker til funktionalitet kan hurtigere udvikles; og en alvorlig fejl eller nedbrud kan begrænses til en del af systemet og ikke *hele systemet*.

Men hvad gør man så, når man nu står med en succesfuld monolit, udviklet gennem mange år, og vil til at prøve kræfter med microservice arkitekturen? Hvad er "do's og don'ts"? I dette kursus vil jeg komme ind på taktikker, patterns, metoder, og konkrete programmeringsteknikker til **modernisering**, som iterativt og via *små skridt*, gradvis refaktorerer en monolit så større og større delmængder af funktionaliteten varetages af microservices. Desuden vil jeg gennemgå elementer af **containerisering og automatisering** og de værktøjer der hører til at idriftsætte en microservice arkitektur, samt lægge vægt på **microservice tests**, som er noget mere kompleks end tilfældet er for test af en monolit.

De overordnede guidelines vil blive konkretiseret igennem eksempler fra et case, som også danner rammen omkring hands-on øvelser for kursisterne.

## Indhold

- Hvad er en microservice – og hvad er den ikke?
- Hvilke fordele giver en microservice arkitektur; hvad er fælderne?
- Toolstack til containerisering og automatisering: Docker, swarm, continuous integration og deployment.
- Taktikker/patterns til modernisering fra Richardson, Newman, og Fowler: Strangler Application, Branch by Abstraction, Presentation Tier Splitting, Database Splitting, Parallel Run, osv.
- Integrations- og systemtest for en microservice arkitektur: service tests, consumer-driven contract tests, end-to-end tests.

# Imhotep

Vidensformidling inden for  
Softwareudvikling



- Tools og teknikker til automatiserede service- og end-to-end tests, bl.a. TestContainers.
- Evt. kort om teknikker til at gøre en monolit til et testbart system: application modernization.

Kurset vil veksle mellem præsentationer, diskussioner, og hands-on øvelser. De central programmeringsteknikker/patterns bliver der arbejdet med igennem konkrete øvelser, demonstrationer, og hands-on arbejde på et konkret case study. Kodebase og udviklingsmiljø ligger på en virtuel maskine, som bliver udleveret før kursusstart.

Kursus materiale, case, og kode biblioteker er i Java, men alle teknikker og metoder er anvendelige i ethvert moderne objekt-orienteret sprog.

## Forudsætninger

Case-study og øvelser er baseret på en Java og Gradle stack. Jeg udleverer en VMWare virtuel maskine (VM), der indeholder hele teknologi-stakken i et Ubuntu Linux miljø, samt det case study som opgaverne baserer sig på, så der er ingen opsætningsvanskeligheder. VM'en indeholder desuden IntelliJ som foreslået udviklingsmiljø. Der vil blive en mindre del udvikling i Java, men der vil være stor vejledning i opgaverne, så selve programmeringsprogsdelen vil være mindre krævende, det primære fokus er på bredt anvendelige teknikker.

Det er derfor et krav at kursisterne har adgang til en moderne 64-bit laptop med nok hestekræfter til at drive en VM på fornuftig vis. Det er en fordel at kursisterne er nogenlunde rutinerede i almindelige Linux shell-kommandoer.

## Materiale

Slides og VM udleveres ca. en uge før kursusstart inklusiv en guide til installation og start.

*Rambøll må benytte materialet internt i organisationen men intet af materialet må distribueres til 3.part eller gøres til genstand for offentliggørelse.*

# Imhotep

Vidensformidling inden for  
Softwareudvikling



## Overordnet program for kursusdagen

### Formiddag

- Microservice definition – fordele og ulemper.
- Introduktion til case study samt toolstack.
- Øvelse: Introduktion til kodebasen.
- Modernisering – taktikker og patterns for gradvis refaktorisering af monolit til en microservice arkitektur.

### Eftermiddag

- Øvelse: Modernisering af case study – isoler en rolle i en microservice.
- Microservice testing – test niveauer med speciel fokus på service tests, consumer-driven contract tests, connector tests, og end-to-end tests.
- Øvelse: Service og consumer-driven contract tests.

## Kursusholder

Henrik Bærbak Christensen har en ph.d. inden for softwarearkitektur fra Århus Universitet, og har været ansat som lektor selv samme sted siden 2003. Før hans ph.d. arbejdede han en årrække i industrien, og har igennem hele sit akademiske virke fastholdt en interesse og fokus på industriel softwareudvikling og at teorier er til for at blive omsat til praksis. Han har arbejdet sammen med mange it-virksomheder igennem årenes løb såsom Uber, Google, Systematic, B&O, Danfoss, Grundfos, KMD, Jyske Bank, TDC, Kamstrup, Mjølner og mange andre. Dette tætte parløb med industrien ses også i, at Henrik er leder af Efter- og Videreuddannelsen "Master i IT" i Softwarekonstruktion, en del af IT-Vest, hvor fokus er på opkvalificering af ansatte i den danske IT-industri.