
UCL ERHVERVSAKADEMI OG
PROFESSIONSHØJSKOLE

SYNOPSIS

# .NET Core og Microservices

**Koordinering af forretningsprocesser på tværs af services**

*Julian Mathias Kock*

Underviser
Kaj BROMOSE

19. maj 2020

# Indhold

# 1 Indledning

Vores verden er blevet en digital verden. Alt lige fra ens køleskab og elkedler til valutaer og telefoner er blevet digitale. Internetforbindelserne bevæger sig nu med lysetshastighed og en efterspørgelse på ingen nedetid, hurtige svartider og overskuelighed er blevet hverdag for udviklere.

Softwaren der udvikles har aldrig været under så stort et pres. Udviklere bliver sat til at løse endnu mere komplekse forretningsprocesser hvor hver enkelt kodeblok skal være let læselig, skalerbart og omskiftelig. Samtidig skal koden være gennemtestet, virke i alle mulige forskellige miljøer og integrere let mod andet software.

Med andre ord er arkitekturen samt infrastrukturen blevet vigtigere end aldrig før. For at imødekomme alle disse krav har man kigget mod opdeling af kode i en service baseret arkitektur, hvor hver service tildeles et ansvar og kan blive udviklet og skaleret individuelt. Fokus i dette fag har været Microservices som er et term for mange autonome services der arbejder sammen for at udarbejde og give en forretningsmæssigværdi[1].

Denne opgave vil fokusere på den koordinering der skal til for at Microservices kan løse forretningsmæssigeopgaver, samtidig med at services skal være autonome og opnå eventual consistency.

# 2 Problemformulering

**Hvordan opnår man en fornuftig koordinering af Microservices til at løse forretningsmæssigeopgaver?**

Del spørgsmål

1 Hvilke mønstre findes der til at koordinere Microservices?

2 Hvordan kan man sikre autonome Microservices?

3 Hvad skal din Microservices tage højde for at opnå eventual consistency?

# 3 1. delspørgsmål

Hvilke mønstre findes der til at koordinere Microservices?

## 3.1 Orchestration

Når der tales om koordineringen af Microservices menes der den interne kommunikation/dataudveksling der skal til for at services kan udføre en forretningsproces. Et eksempel på en forretningsproces kunne f.eks. være at lave en ordre der tilsidst afsendes og modtages af en kunde. I en traditionel arkitektur kunne man bruge ACID transaktioner, men når der er tale om flere services er det svært at lave isoleret transaktioner. En måde hvorpå man kan imødekomme dette er ved brug af distrubureret transaktioner, så som 2PC (two-phase commit)[2]. Denne løsning kan være tilstrækkelig, men den giver dog nogle ekstra problemer.

# Litteratur

[1] Sam Newman. *Building microservices: designing fine-grained systems*. "O'Reilly Media, Inc.", 2015.

[2] Chris Richardson. *Microservices Patterns: With Examples in Java*. Manning Publications, 2019.