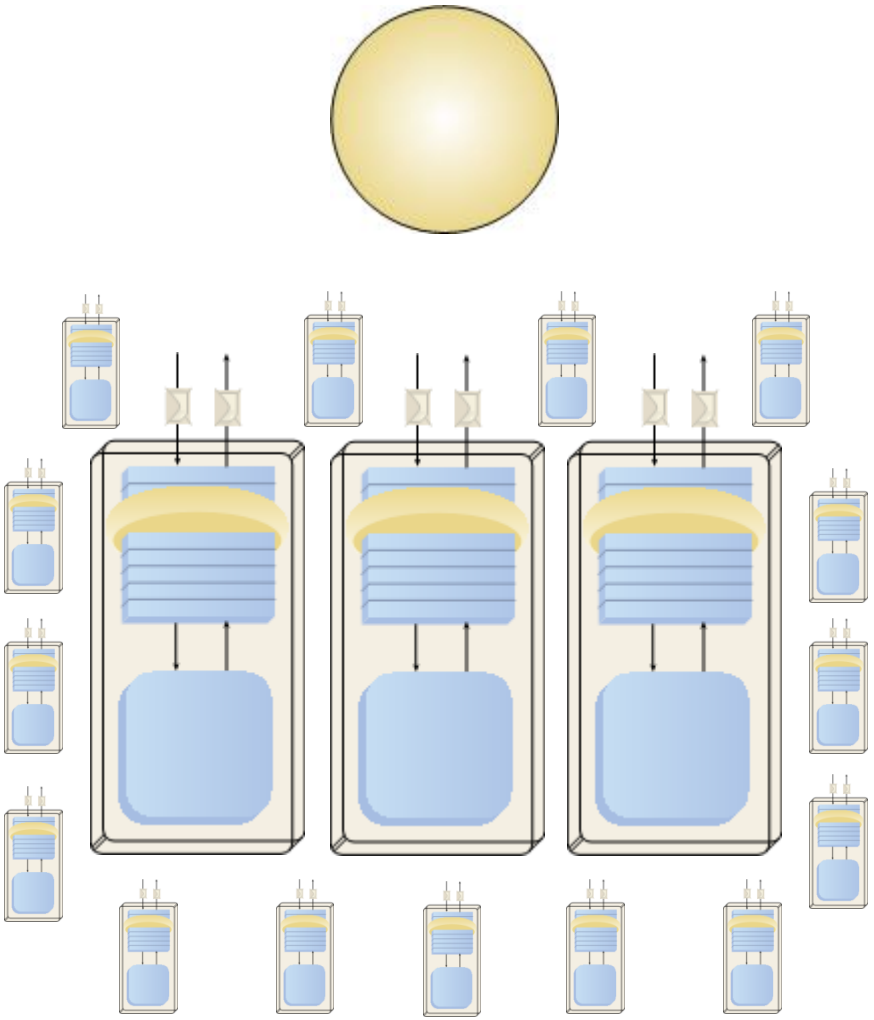
SOA Design mönster i molnet

Herbjörn Wilhelmsen
herbjorn.wilhelmsen@objectware.se

OBJECTWARE

Redundant Implementation



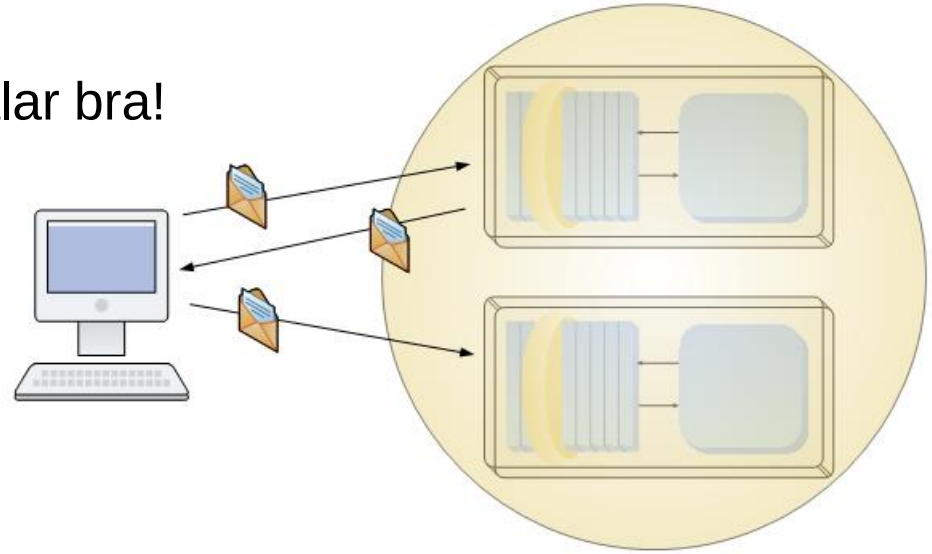
Logisk tjänst

Implementation av tjänst

Skala mycket, enkelt och kostnadseffektivt i molnet

Stateless vs Stateful

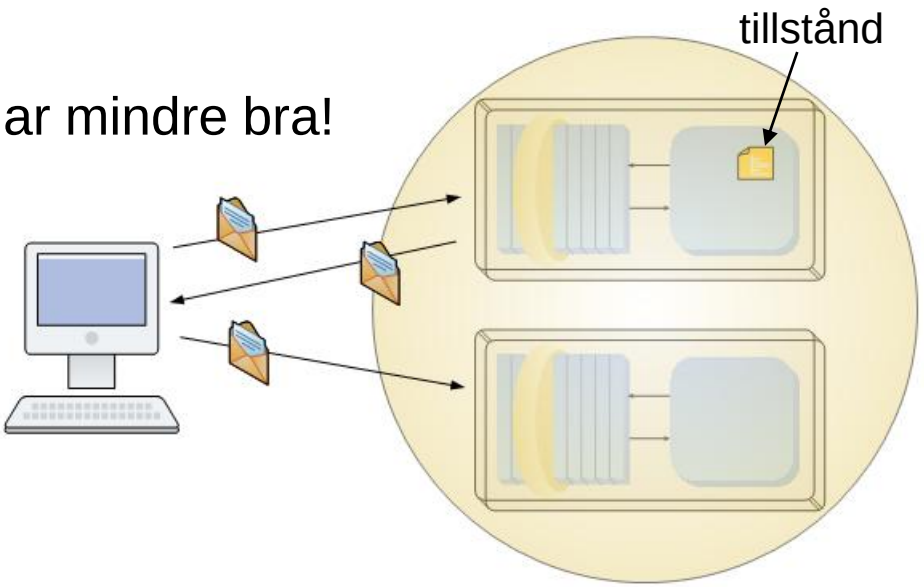
Skalar bra!



Stateless:
Varje anrop sker utan hänsyn till tidigare anrop

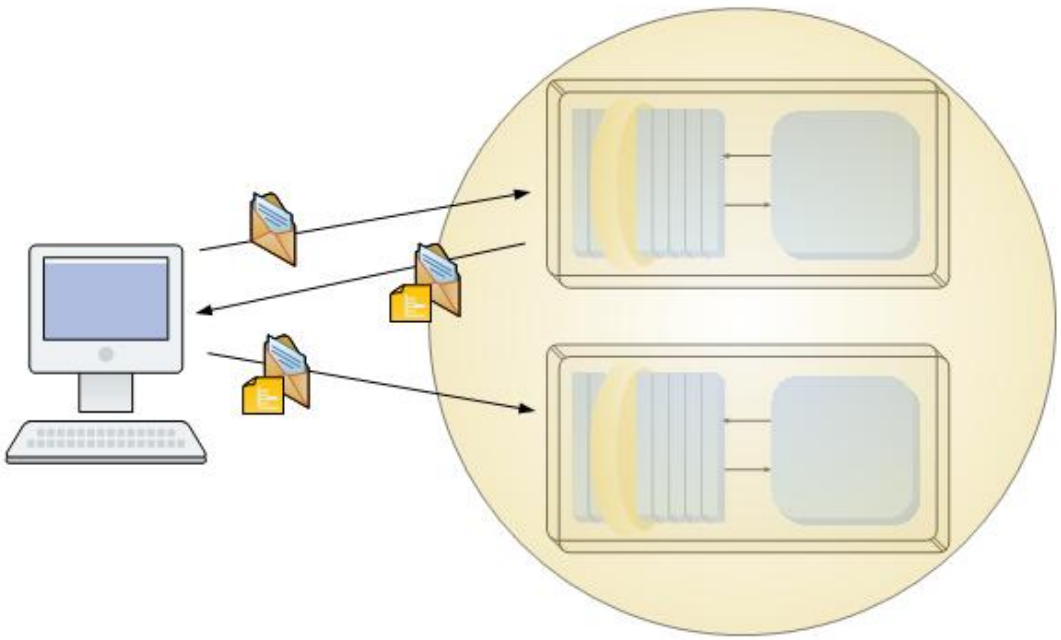
Ibland behövs tillstånd!

Skalar mindre bra!



Stateful:
Anrop anrop sker där hänsyn till tidigare anrop måste tas

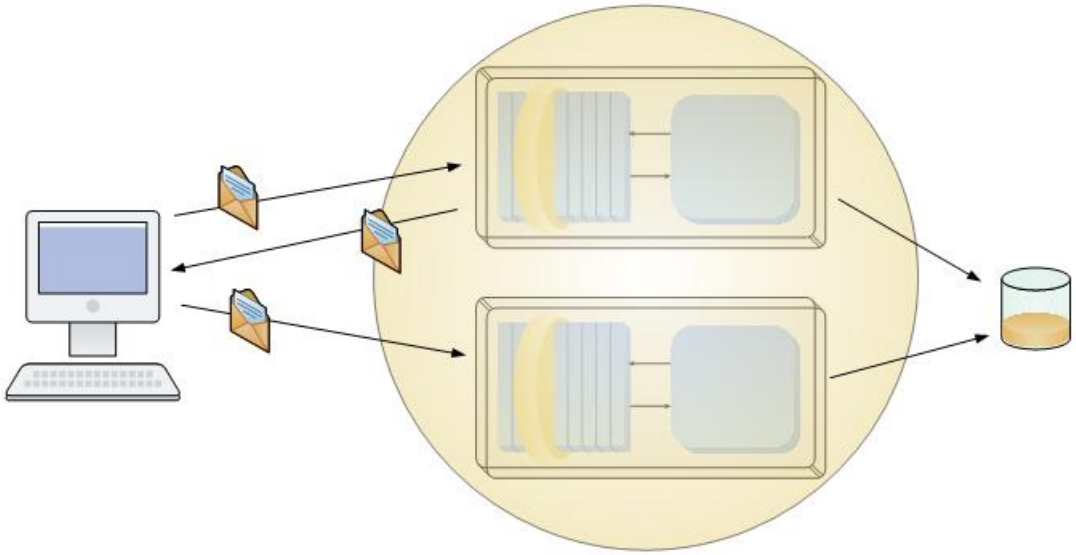
State Messaging



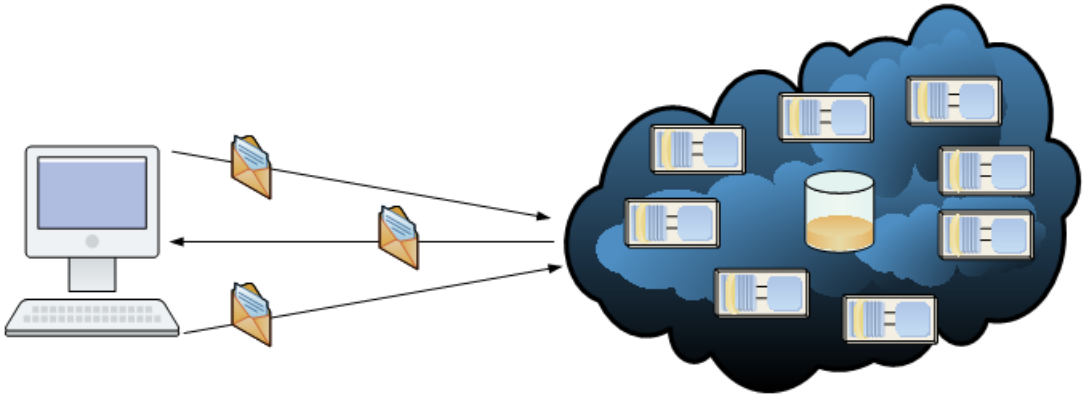
- Skicka tillståndet med i meddelandet
- Skalar lika bra som stateless
- Fungerar inte för
  - delat tillstånd

Fungerar sämre för
- stora mängder tillstånd
- sekretessbelagt tillstånd

State Repository (1 av 2)

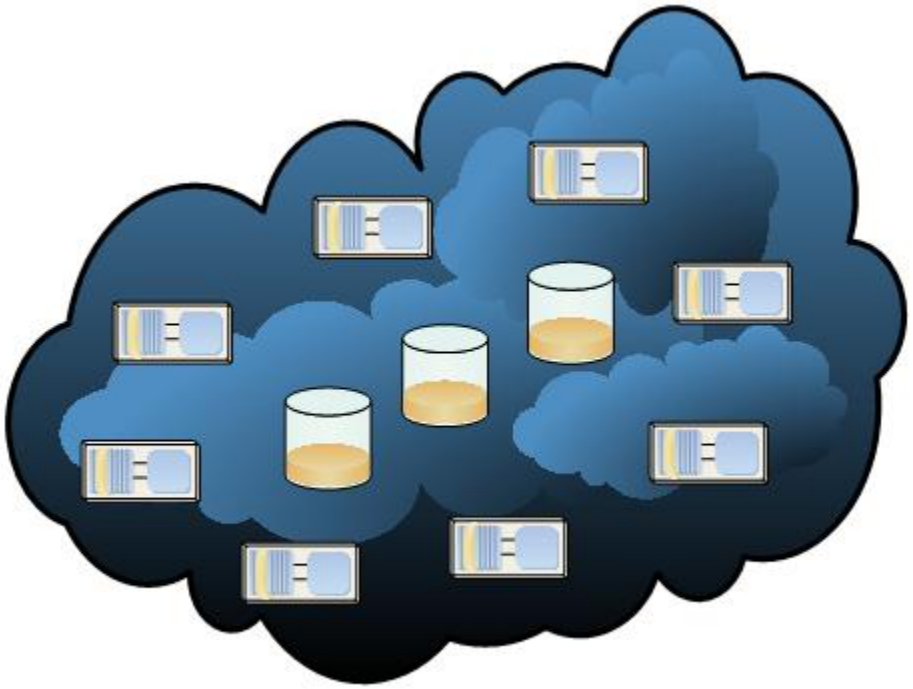


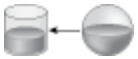
Fungerar bra för
- delat tillstånd
- stora mängder tillstånd
- sekretessbelagt tillstånd



Skalar inte så bra i molnet
- single point-of-failure
- STOR flaskhals

State Repository (2 av 2)





Flera state repositories:
- ingen single-point-of-failure
- repositories måste synkroniseras

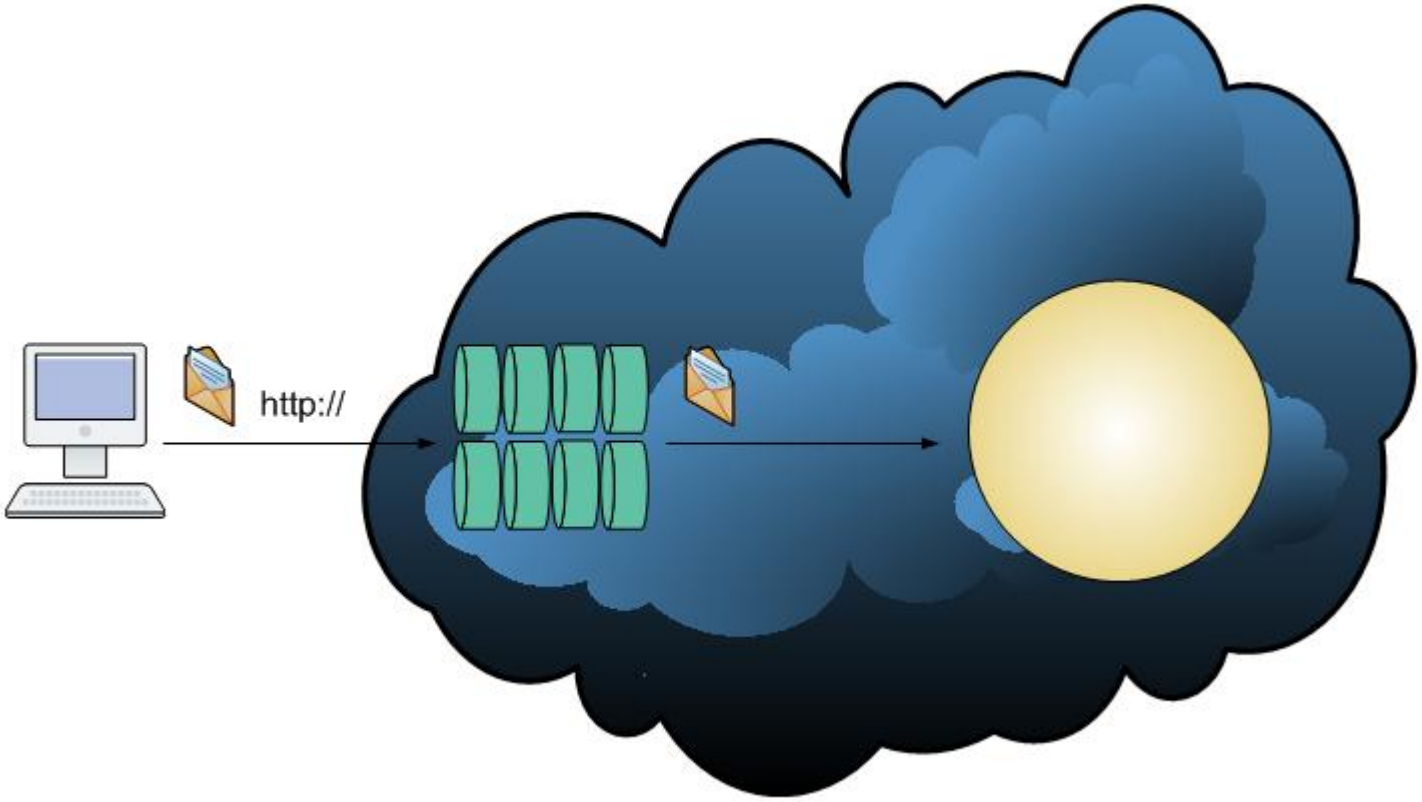
CAP (Consistency-Availability-Partition tolerance teoremet) förstör!

Vi måste välja mellan

Hög konsistens:
- skriv till 2 av 3
- läs från 2 av 3

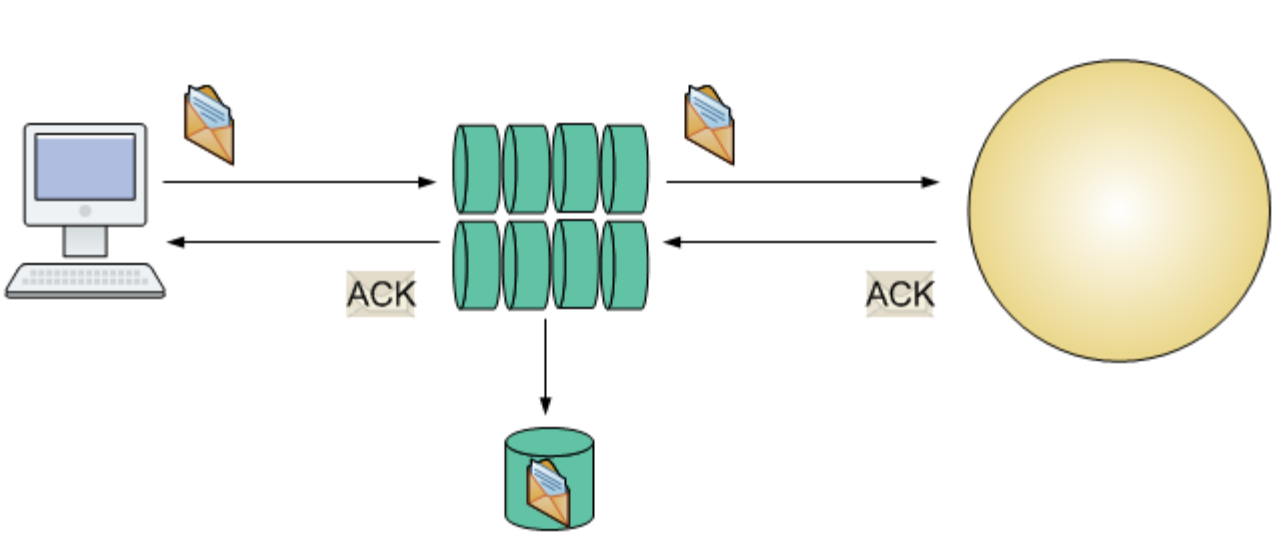
Hög tillgänglighet:
- skriv till 1 av 3
- läst från 1 av 3

Asyncronous Queuing

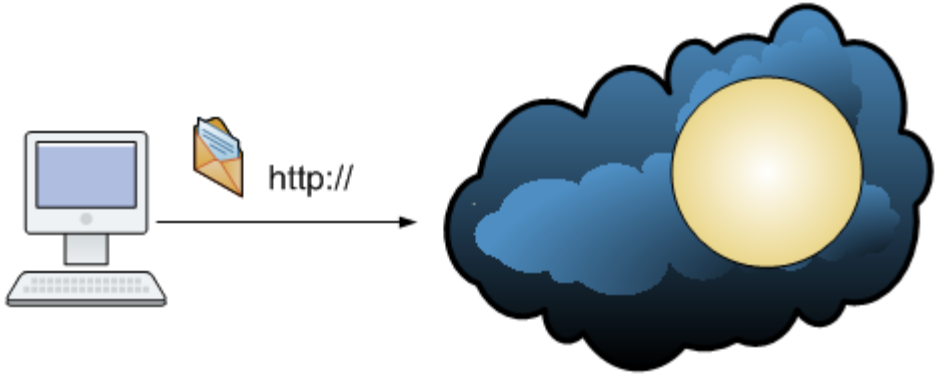


- Asynkrona köer kan minska problem med tjänster som är nere
- Köer finns och kan fungera bra i molnet

Reliable Messaging

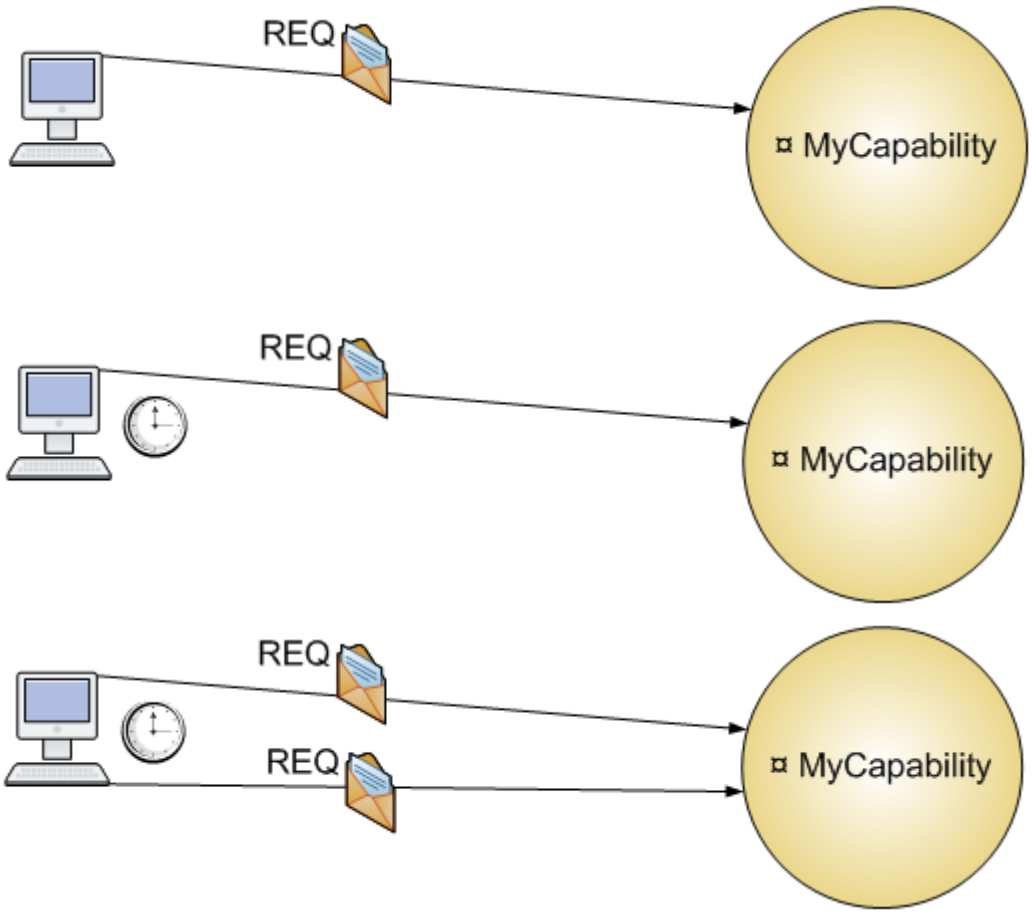


Tillförlitligt
men komplicerat



- http är inte tillförlitligt
- svårt att ställa krav på konsumenter när det gäller
  - eget nätverk
  - teknikval

Idempotent Capability (1 av 3)



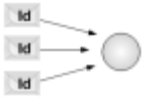
Skicka ett anropsmeddelande

Vänta på ett svar som inte kommer…

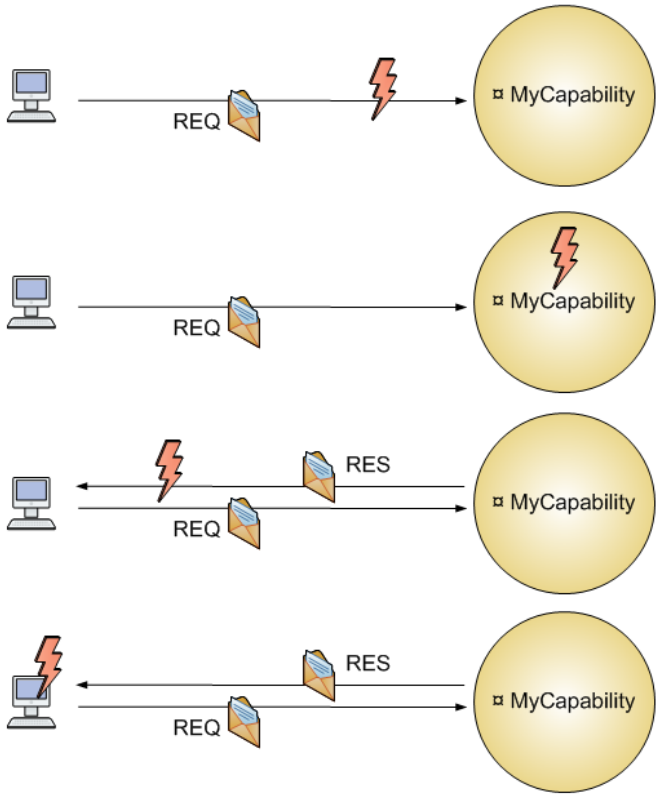
Vad händer sedan?

Att skicka om anropsmeddelandet skulle kunna vara den enda rätta lösningen…

Idempotent Capability (2 av 3)



När ett förväntat svar inte kommer tillbaka kan det ha gått fel…



… på väg till tjänsten

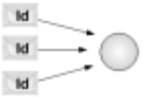
… inuti tjänsten

… på väg tillbaka

… efter att det kom tillbaka

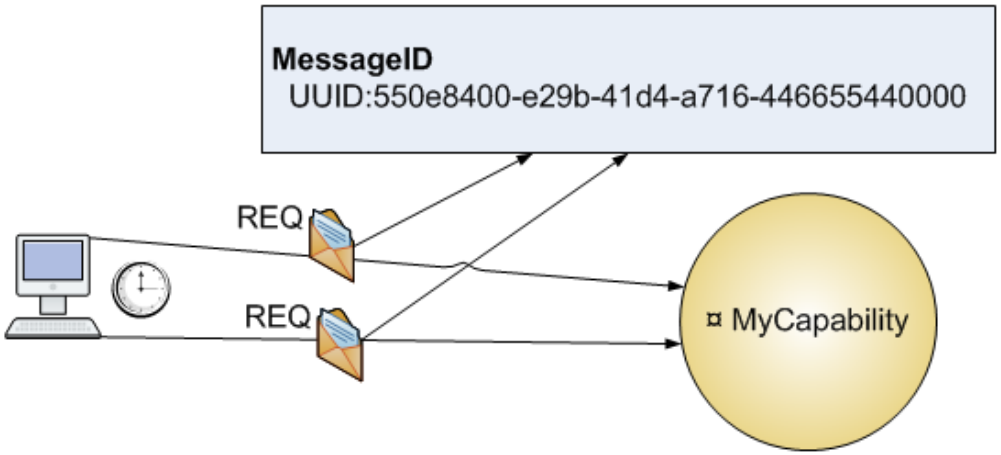
Det spelar stor roll var det gick fel!

Idempotent Capability (3 av 3)

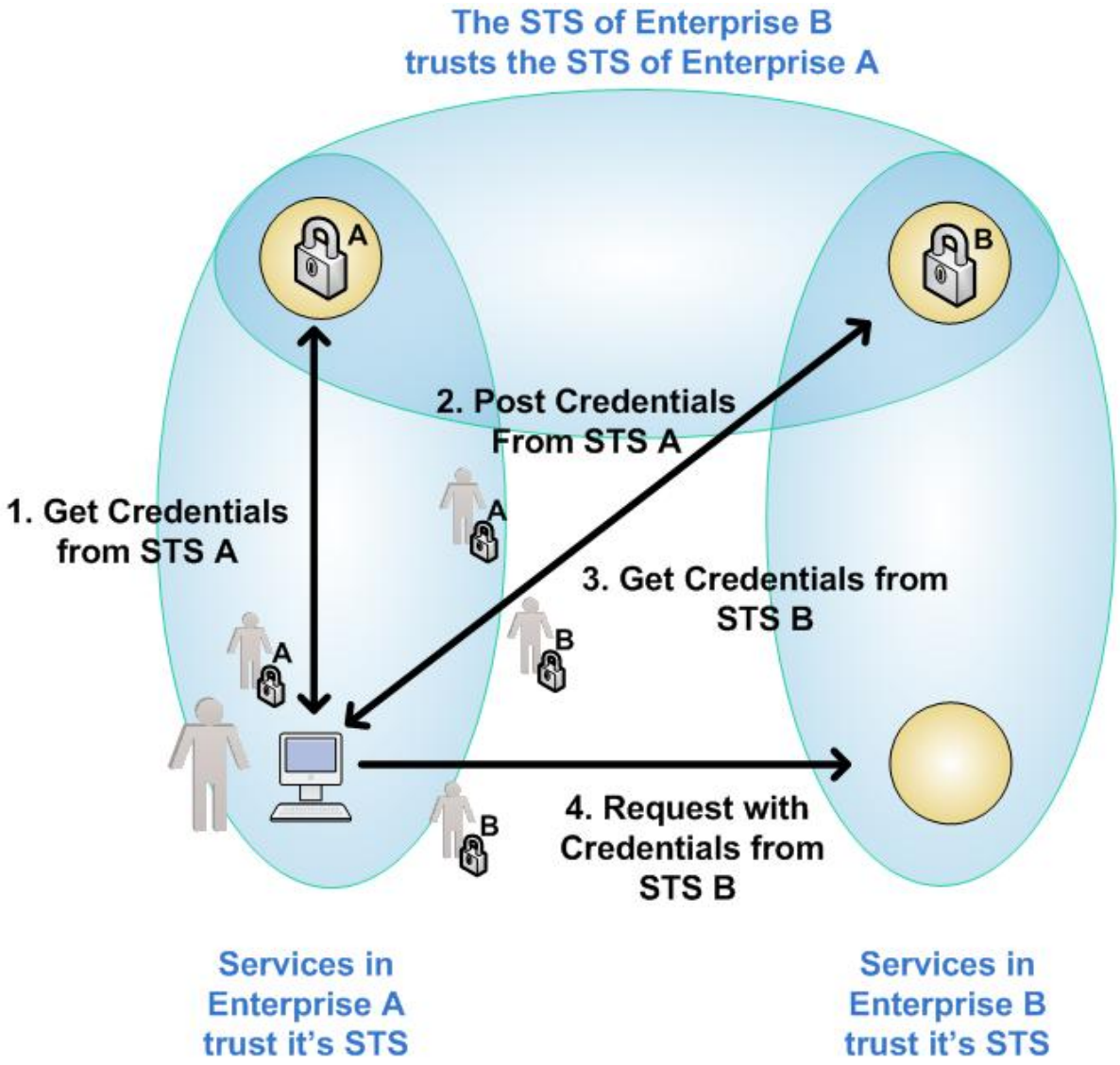


Se till att det inte spelar någon roll hur många gånger ett meddelande skickas:

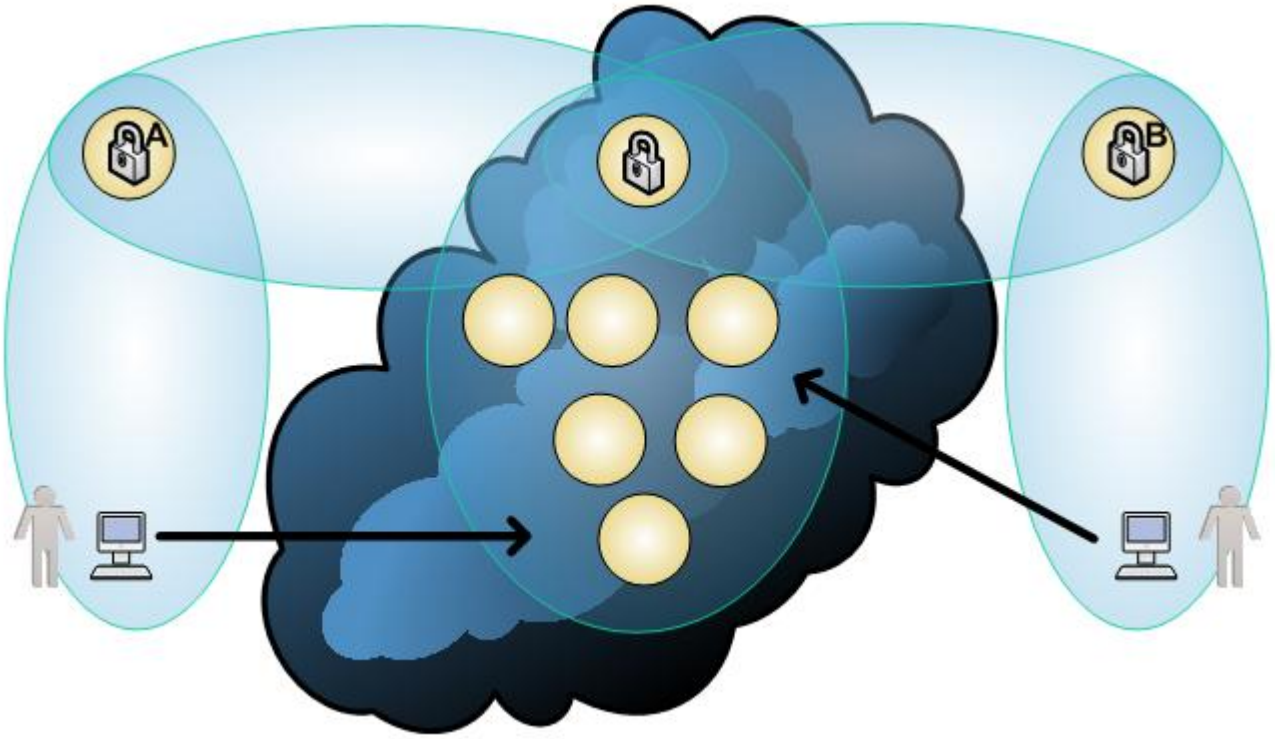
1. Logik som inte har några sidoeffekter alls ← Kan vara svårt!
2. Identifiera varje meddelande unikt och spara undan vilka som har behandlats



Federated Identity (1 av 2)

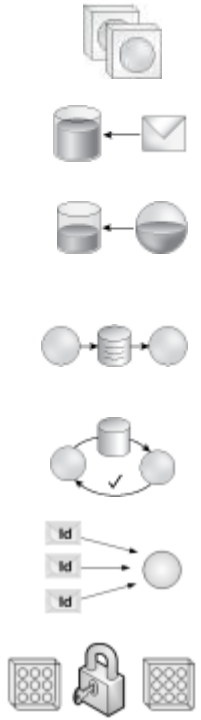


Federated Identity (2 av 2)



Federated Identity fungerar utmärkt i molnet!

Frågor?



http://soapatterns.org