

MELDINGSKØ

Robert Bauck Hamar <r.b.hamar@usit.uio.no>

Cerebrum-utvikler ved USIT

MOTIVASJON

- Event-basert oppdatering:
 - Meldinger + rest-grensesnitt
 - Meldingene skal ikke ha sensitive opplysninger, og i minst mulig grad personopplysninger
 - Hvem som helst skal kunne abonnere
- Hjelp til med intern forretningslogikk

STANDARDS

Det finnes mange standarder, f.eks. AMQP 0.9.1, AMQP 1 og Stomp, og ulike systemer støtter disse. Etter å ha utforsket området, valgte vi å prøve med RabbitMQ, som benytter AMQP 0.9.1.

RABBITMQ



Exchange



Kø

BROKER

En installasjon kalles en broker. Den kan kjøre repliserende på flere maskiner med failover, og man kan til og med oppgradere systemet uten nedetid.

En broker har flere vhosts, som gir ulike adskilte systemer.

EVENTBASERTE OPPDATERINGER

Vi har alltid ønsket oss dette. Når en ny person registreres i SAP, sender SAP en melding, som leses av bl.a. Cerebrum. Cerebrum oppretter bruker, og sender en melding om dette, og disse plukkes opp av bl.a. AD.

MELDINGENE

Under panseret sendes en melding som et lite JSON-objekt, hvis format er basert på SCIM (System for Cross-domain Identity Management):

sub

<https://tyk->

[gw.usit.uio.no:8443/cerebrum/v1/account/42](https://tyk-gw.usit.uio.no:8443/cerebrum/v1/account/42)

eventUris

["urn:ietf:params:event:SCIM:modify"]

urn:ietf:params:event:SCIM:modify

{"attributes": ["expireDate", "npType"]},

INTERN FORRETNINGSLOGIKK

Cerebrum kan konfigureres langt vha. mixins, men disse har klare begrensninger.

Vi bruker i dag changeloggen til å plukke opp hendelser fra andre jobber, og legge på ekstra forretningslogikk. Eller vi kjører scripts etter importjobben. Dette kan endres ved at Cerebrum leser egne meldinger og handler på bakgrunn av disse.

CELERY

Vi ser på systemet Celery som kan brukes til å utsette utsendelse av meldinger. Dette kan brukes f.eks. til å gjøre noe med karantener som har virkning i fremtiden.

Celery er også et produkt som potensielt kan erstatte job runner.

IMPLEMENTASJON

Pakken `Cerebrum.modules.event_publisher` inneholder kode som oversetter de gamle `log_change`-kallene til meldinger. Disse sendes automatisk når man committer til databasen. Denne bruker generell klient basert på Python-klienten Pika til å sende meldinger, og klienten kan ved behov brukes til andre ting.

RESULTAT?

- Kjappere endringer → bedre sikkerhet
- Færre store jobber som kjøres hver natt
- Fornøyde brukere

FREMTIDEN

I dag er Cerebrum stort og monolittisk. Andre som bruker meldingskø, oppnår gjerne et system bestående av mindre mikrosystemer, som snakker sammen. Allerede i dag oppnår vi dette ved at eksportene i større grad flyttes over til mottakersiden:

Cerebrum logger inn i AD og kjører kommandoer → Script på AD henter data fra Cerebrum og oppdaterer.

Cerebrum lager en fulldump til system X → System X henter endringer og oppdaterer.

ERFARINGER

For tidlig ennå med erfaringer. Vi har hatt barnesykdommer, som at cerebrum stoppet pga. full var-disk på RabbitMQ-maskin.