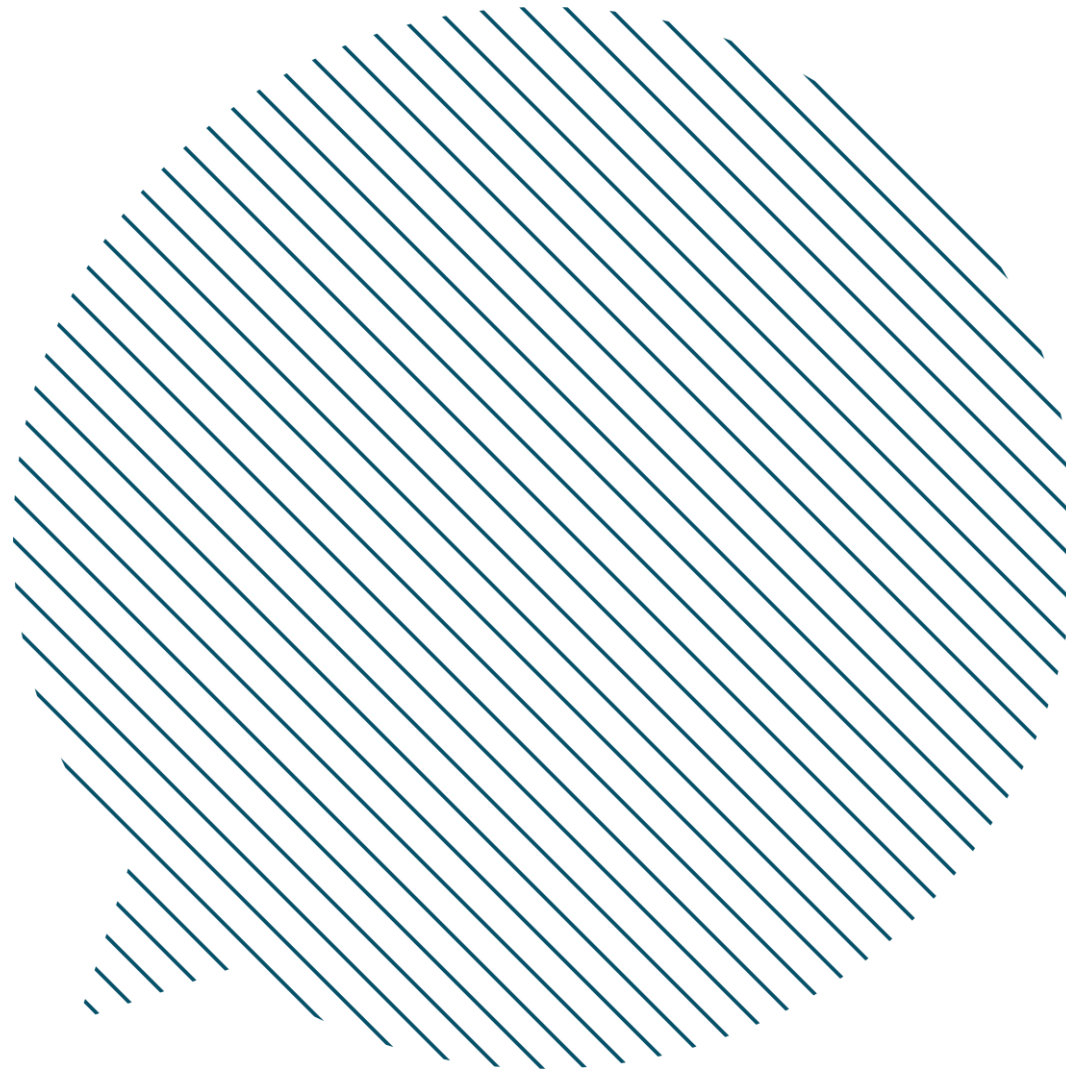


MS FHIR Server hos HDIR

FHIR fagforum, 1. September 2021

Magnar Wium



Agenda

- Bakgrunn
- Krav
- Status
- Bakgrunn valg
- MS FHIR Server / Cosmos DB
- Løsning
- Erfaringer

Bakgrunn - HDIR/Helfo

- Mange interne kontaktregistre
- «Hopping» mellom systemer
- Stor grad av fritekstføring
 - Dårlig datakvalitet
- Unødig punsjing for Helfo



Funksjonelle «epos»

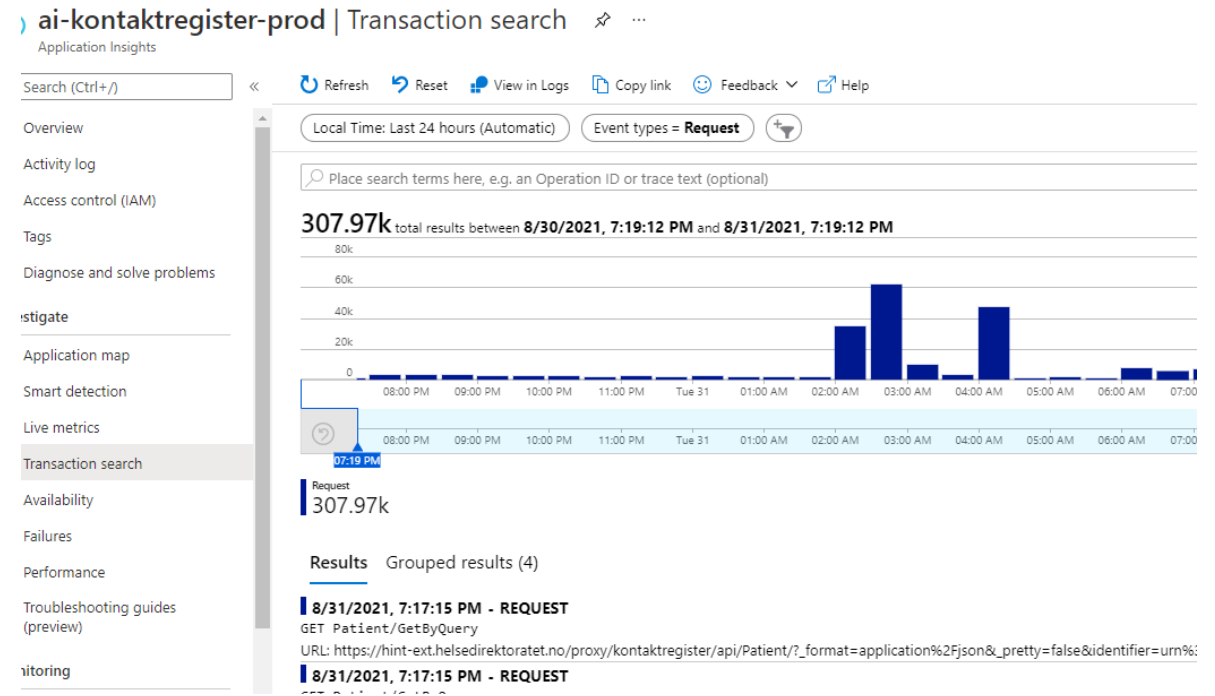
- Administrasjon av helsepersonell som har refusjonsrett fra Helfo.
- Administrasjon av virksomheter som har refusjonsrett fra Helfo.
- Kontaktregister for privatpersoner for Helfos fagsystemer (og hdir generelt).
- Kontaktregister for organisasjoner for Helfos fagsystemer (og hdir generelt).
- Rollebasert tilgangsstyring for Helseaktørportalen og dennes tjenester.
- Master for personer uten norsk identifikator.

Utvalgte tekniske mål

- Krav til bruk av FHIR
- Hendelsebasert kommunikasjon
- Høy ytelse på søk og oppslag
- Uavhengig av infrastruktur
- «Multi tenant» sikkerhetsmodell
 - Helseld
 - NTML(On-prem AD)
 - HelseNorge
- «Bane vei for HDIR's skystrategi og personopplysninger i sky
 - PaaS
 - Utilgjengelig fra internett
 - Hybrid
 - Tilgangsstyring

Status

- Første klienter på løsningen i 2020
- 12 systemer har koblet seg på så langt
- Full produksjon med sentrale NAV system juni 2021
- Prosjektet er «avsluttet»
- Ruller på fortløpende i forvaltning
 - Neste er Frikort-løsningen i NAV for oppslag på kontonummer
- På en vanlig dag 300-600k oppslag



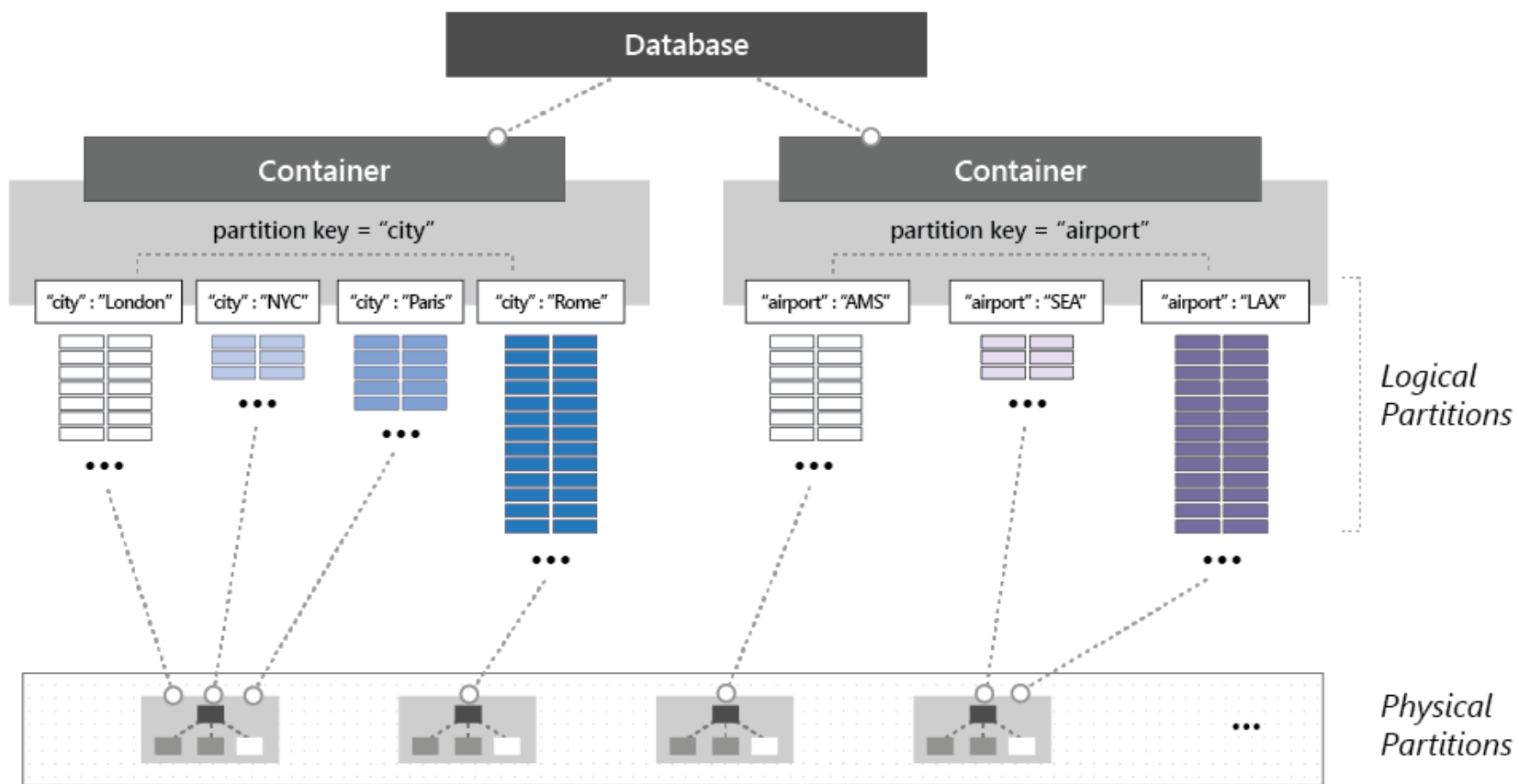
Hvorfor valgte vi MS FHIR Server

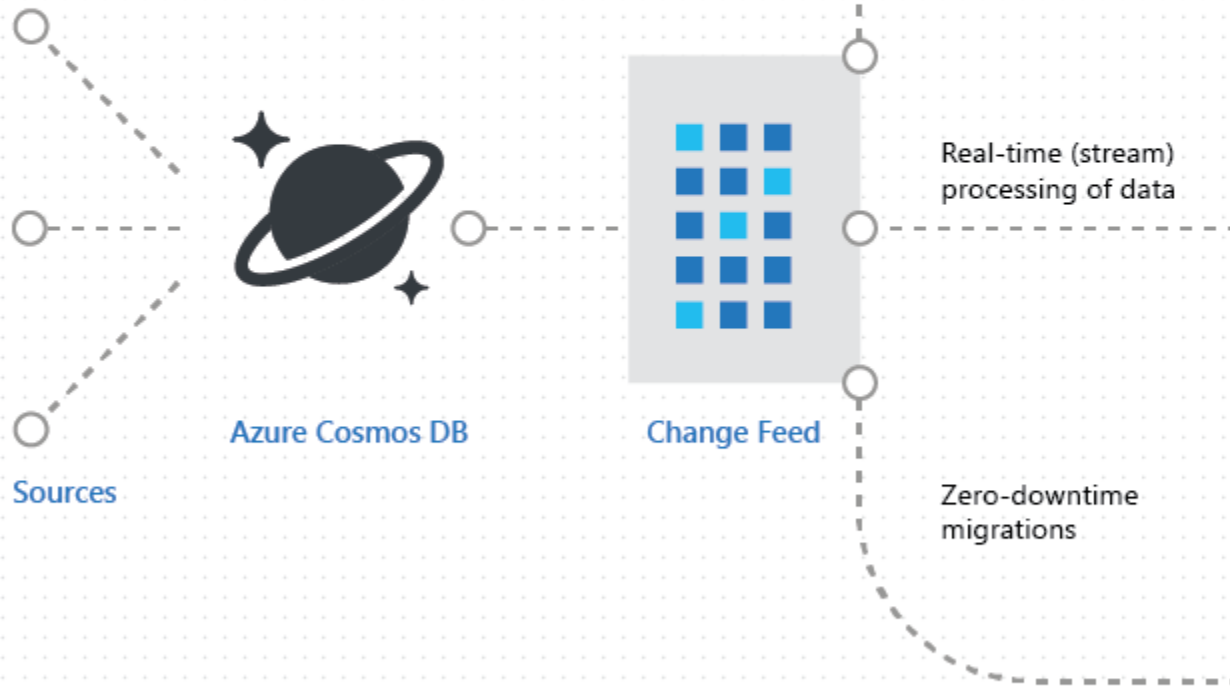
- Dekkende HI7 FHIR Rest api
- Åpen kildekode med stort «community»
- Kjent teknologi for hdir (C#, asp .net)
- Kjører på IIS og Azure PaaS

MS FHIR Server

- Åpen kildekode
- Avansert søk, eksport og validering
- Finnes nå også som en PaaS-tjeneste i Azure
- Hosting layer : Støtter kjøring i ulike miljøer; Azure Paas, IIS, k8s
 - Ligger ARM-script og docker-container i repo
 - Vi har brukt ARM-script som utgangspunkt for vår infra
- RESTful API layer : API definert i HL7 FHIR spesifikasjon
- Core logic Layer – Implementasjon av kjernelogikken. Egen logikk kan «plugges» inn gjennom såkalte «pipeline behaviours»
- Persistence layer – «Pluggable» . Støtter MS SQL Server og Azure Cosmos db

Cosmos db





Event-Computing and Notifications

Retail, Gaming, Content management



Azure
Functions



Azure
Notification Hubs



Azure
App Service

Trigger call to an API
when a document is
inserted or modified

Stream Processing

IoT processing, Data Science & analytics



Azure
Stream Analytics



Azure
HDInsight



Apache
Spark



Apache
Storm

Real-time (stream)
processing of data

Data movement

Enterprise data management



Azure
Storage Blob



Azure
Storage Table



Azure
Data Lake

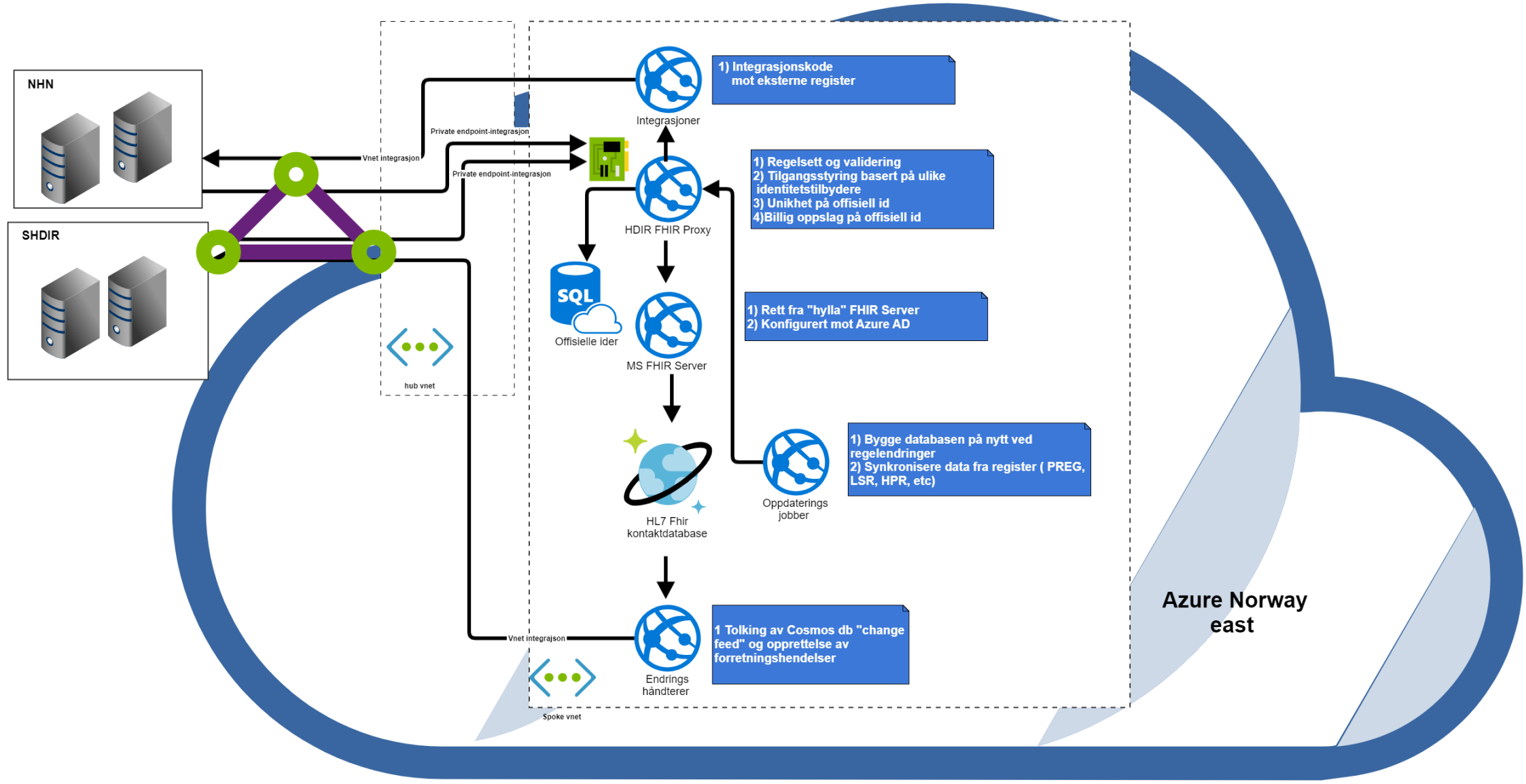


Azure
Cosmos DB

Zero-downtime
migrations

MS FHIR Server + Cosmos Db

- Alle FHIR-ressurser i en container
- Syntetisk ressurs-id er partisjonsnøkkel
- Effektivt søk er løst gjennom «Meta»-modell
 - Dokument med promoterte felter som peker rådokument
 - Søke parametere som egne dokumenter
 - Nye parametere krever at alle meta-dokumenter blir oppdatert
- Transaksjoner ikke støttet



Erfaringer

- FHIR Server var et godt valg
 - Hadde aldri kommet i mål uten
 - Godt «produkt»
 - Community som lytter til våre tilbakemeldinger
- “Proxy”-pattern gav oss fleksibiliteten vi trenger
 - Suksessfaktor at vi splittet
- Fagsystemer kobler seg på i god takt
- Malen for infrastrukturen vi har satt opp i Azure brukes nå av flere løsninger

Erfaringer

- Validering fra start
- Søk er dyrt
 - Whitelist av endepunkter og parametere vi tiltaler
- Størrelsen vokser og ytelsen blir dårligere
 - Vi må finne en løsning for dette
- Vår implementasjon av hendelser krever stadige kodeendringer
- Mye ekstra kode for håndtering offisielle ider
- Søk blir dyrt, vi må passe på hva vi åpner opp for
 - Whitelist
- Rapportering og innsikt er ikke trivielt
- Kort sagt; Vi tror at SQL Server hadde vært et bedre DBMS