

30 OKTOBER 2025

Lokale devvmiljøer, skyløsninger, og emulering

ALEKSANDRA GLESAEN ØDEGÅRD

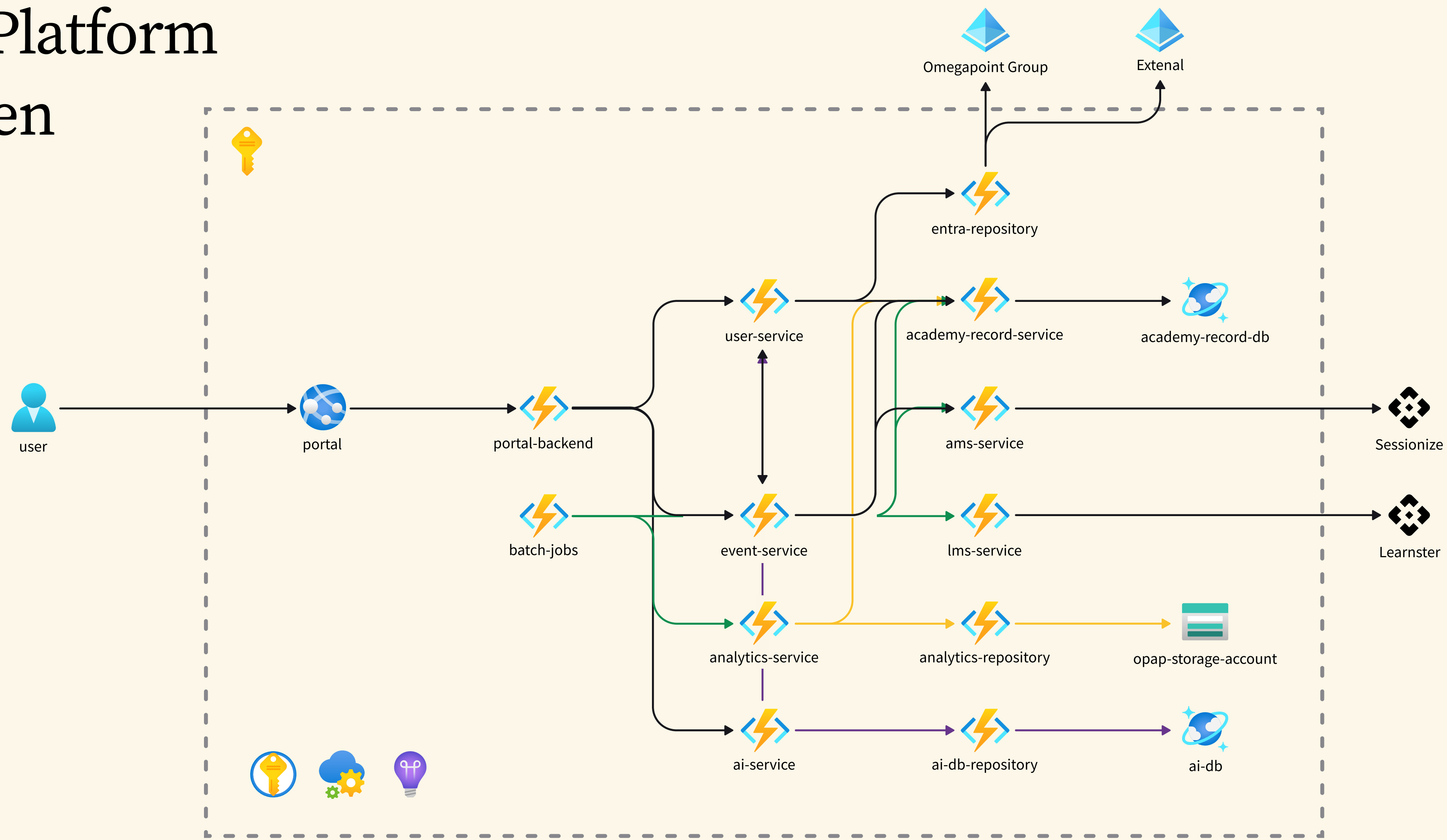
Hva handler denne presentasjonen om?

- Mine refleksjoner etter å ha laget mange devmiljøer i prosjekter jeg har vært i
- Tanker rundt dette med å emulere bort vanskelige eksterne avhengigheter
- Noen forslag til løsninger, se hva som finnes, høre hva folk gjør selv

Academy Platform arkitekturen

- Portal
- Batch jobs
- Analytics
- AI services

12 function apps
30+ dependencies



Hva mener jeg med et devmiljø?

Devmiljø er et ganske vagt begrep, men jeg mener det er samlingen av alt av verktøy, ressurser, og tester man har tilgjengelig mens man utvikler.

- IDE'er, lintere, formatteringsverktøy
- Containere, orkestreringsverktøy, ressurser
- Database-verktøy, diagnostikk
- Tester og test-runners
- Og mye mer ...

Utfordringer med sky for lokal utvikling

- Man må få tilgang til alle ressursene man bruker
- Det er gjerne én ressurs i et område, så den er delt mellom alle
- Man kan ikke alltid få **se** hva som vil skje i produksjon
- Eksterne avhengigheter har ikke alltid et testmiljø
- Løsningen blir gjerne emulering

Fordeler og ulemper med emulering

Bruke ekte tjenester

- 👍 De er de faktiske tjenestene
- 👎 Kan være dyrt
- 👎 Kan ha latency
- 👎 Concurrency & persistence
- 👎 Tilganger

Bruke emulerte tjenester

- 👍 Mer konsistente
- 👍 Ofte raskere
- 👎 Tester på et vis kun lokal logikk
- 👎 Har ikke alltid alle features

Hva vil vi faktisk emulere?

- Vil ofte emulere databaser og datalagring
 - Ting man endrer mens man kjører burde man eie selv
- Burde alltid mock'e eksterne API'er med mindre de har dedikerte testmiljøer
- Kan ha dedikerte utvikler-instanser av f.eks. key-vaults og appconfigs

Emulering tierlist

- Tier 1: SQL servere og annen datalagring
- Tier 2: Eksterne APler
- Tier 3: Telemetry, monitorering og logging
- Tier 4: Autentisering
- Tier 9: Entra

Verktøykassa

- Docker, docker compose, (lokal kubernetes?)
- Testcontainers
- API mocking som f.eks. WireMock
- Eksisterende emulatorer
- .NET Aspire?

.NET Aspire?

Det er litt vanskelig å forstå *hva* .NET Aspire er. Hva sier de selv at de er?

Key capabilities

- AppHost orchestration
- Rich integrations
- Consistent tooling

.NET Aspire?

- Infrastruktur i C#
 - Avhengigheter mellom .NET-prosjekter
 - Eksterne avhengigheter som f.eks. databaser
 - Azure infrastruktur som faktisk kan deployes
- Dashboard for å se telemetry og kjørende tjenester

Demo

Mine problemer med .NET Aspire

- Støtter for øyeblikket kun monorepo-strukturer (såvidt jeg kan se)
- I større prosjekter er infrastruktur og utvikling separate team
- Alt ansvaret på utviklerne, vi må kunne “alt”

Integrasjonstester og systemtester

- Emulering kan gi deg mye mer kontroll over miljøet testene dine kjører i
 - F.eks. TestContainers for å starte dedikerte databaser per test
- Viktig å tenke på state, lett å få for idealistiske tester

Seeding

- For å kunne kjøre og teste lokalt så trenger man god testdata
- Kan være greit å ha automatisk seeding ved oppstart
- ... noen ganger også automatisk sletting av databasen

THE END

Hva gjør dere?