

nova
net

From 100 applications

To one platform

nova
net

- Senior Consultant @ Novanet AS
- Tech lead / architect
- Various industries



HELLO
MY NAME IS

Leif Larsen

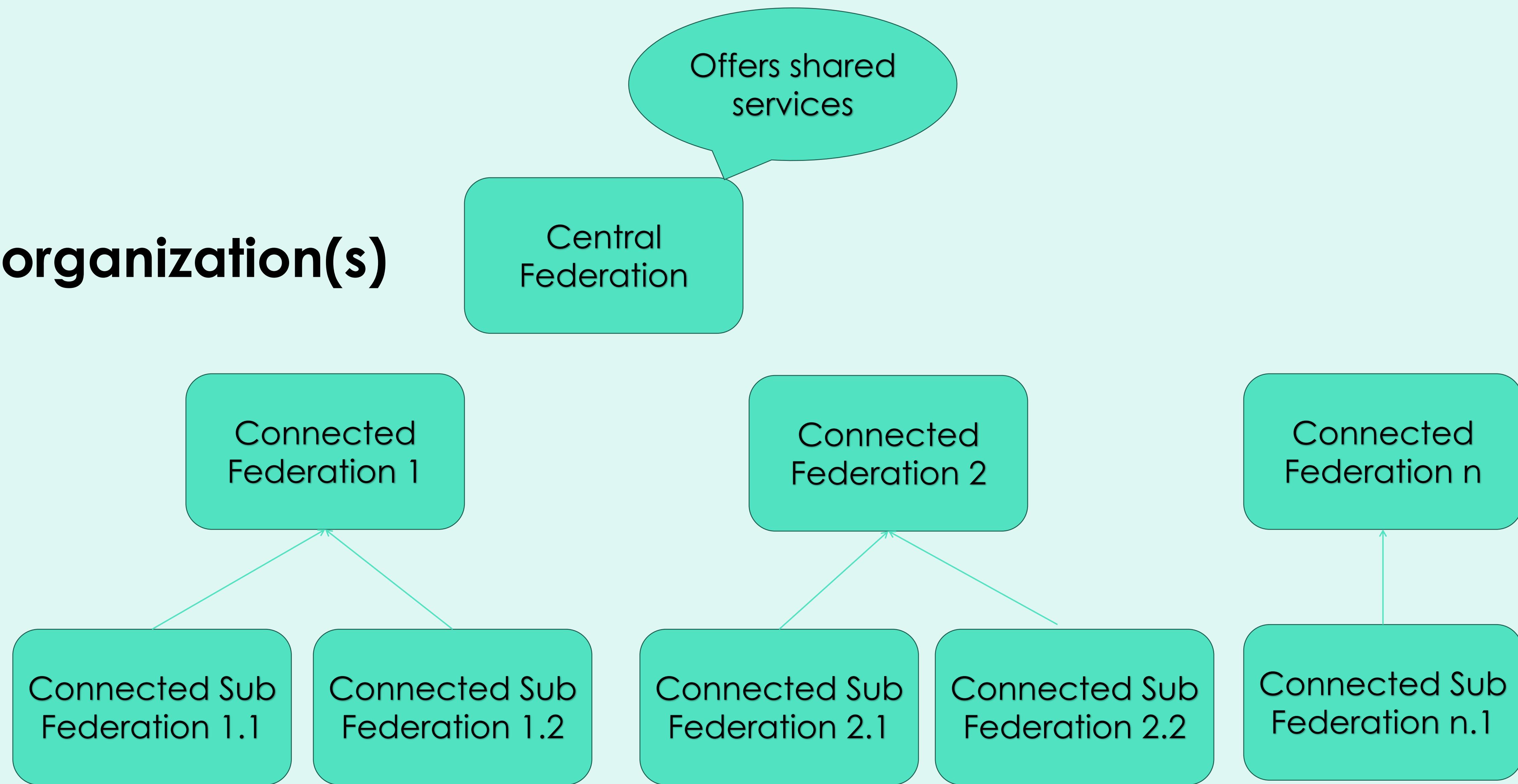


Agenda

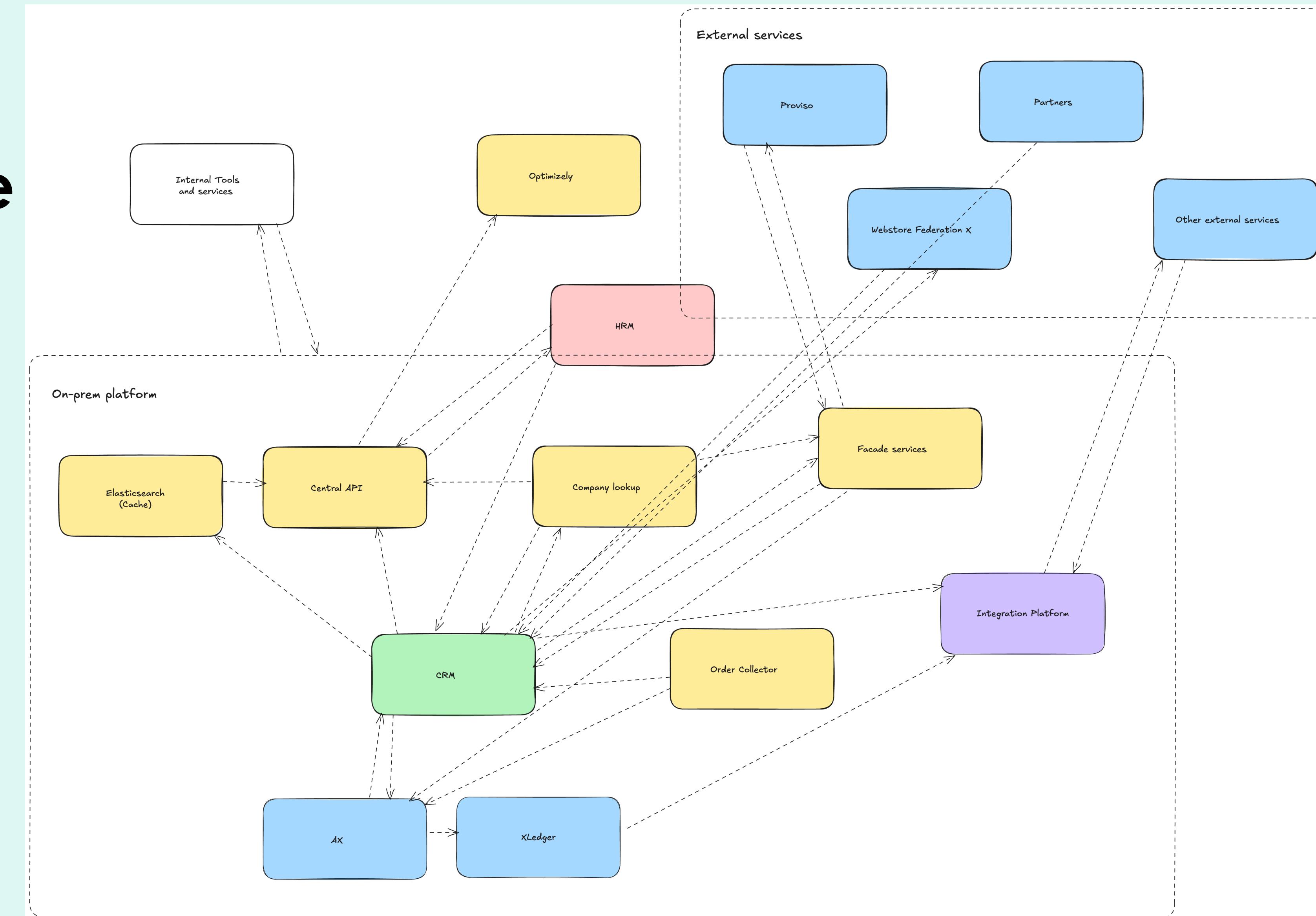
- How did we end up with 100s of applications and no platform?
- 5 strategies we used to change this
- 3 big challenges faced in the process
- How does the architecture look now?
- A look into our tech stack

A brief history lesson

The organization(s)



What it looked like



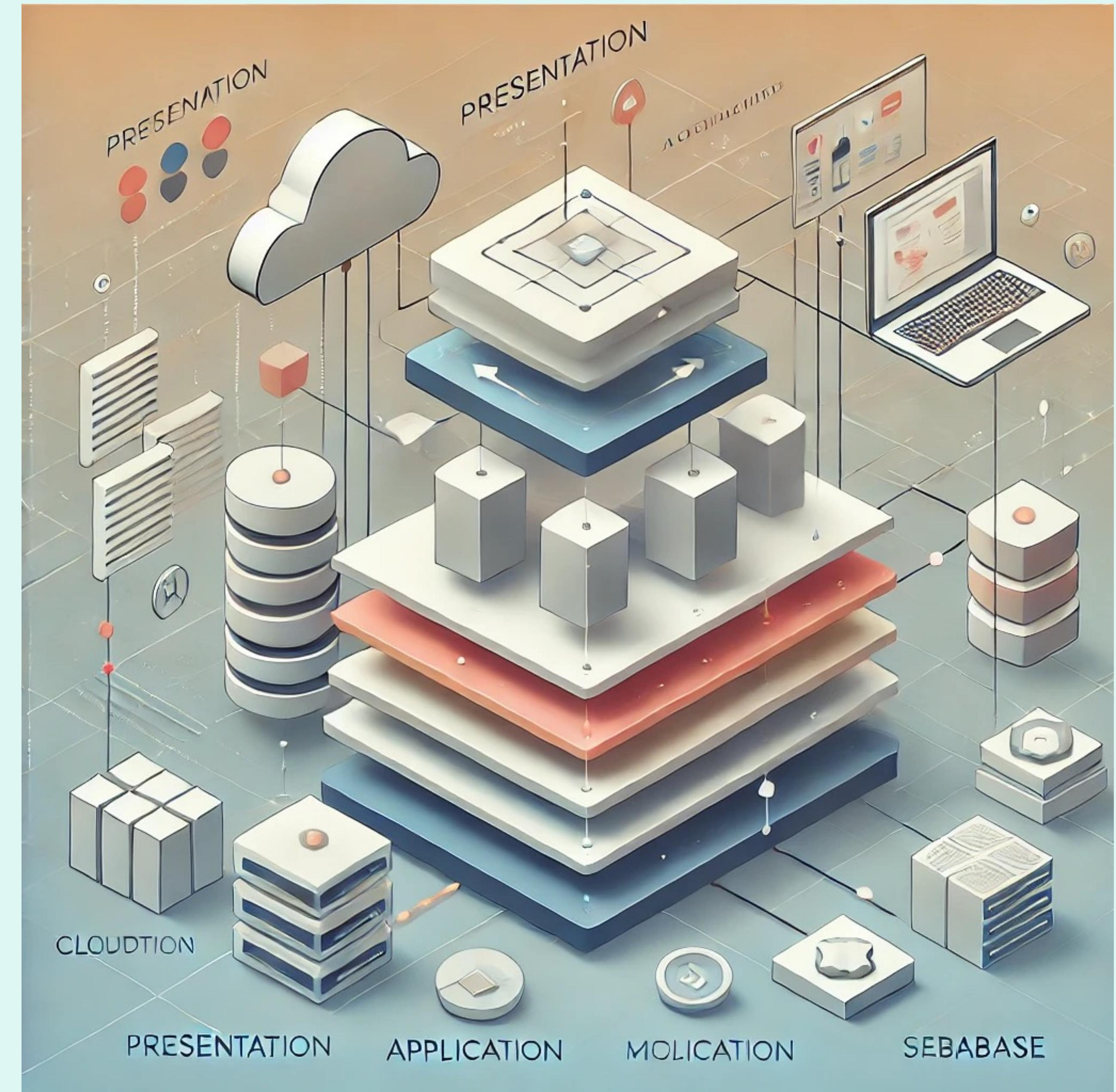
«Yes, we can do that!»

«We'll use our CRM for that»

«What do you think we should do?»

5 strategies to use to renew your IT systems

Abstractions



Service offerings

Financial services

Customer Relationship Management

Content Management Software

Email

And so on...

Common patterns and defining the future



Where do you start?



Shut down services as you go

3 challenges to be aware of

Costly affair



WHO SHOULD PAY?



RETURN OF
INVESTMENTS?

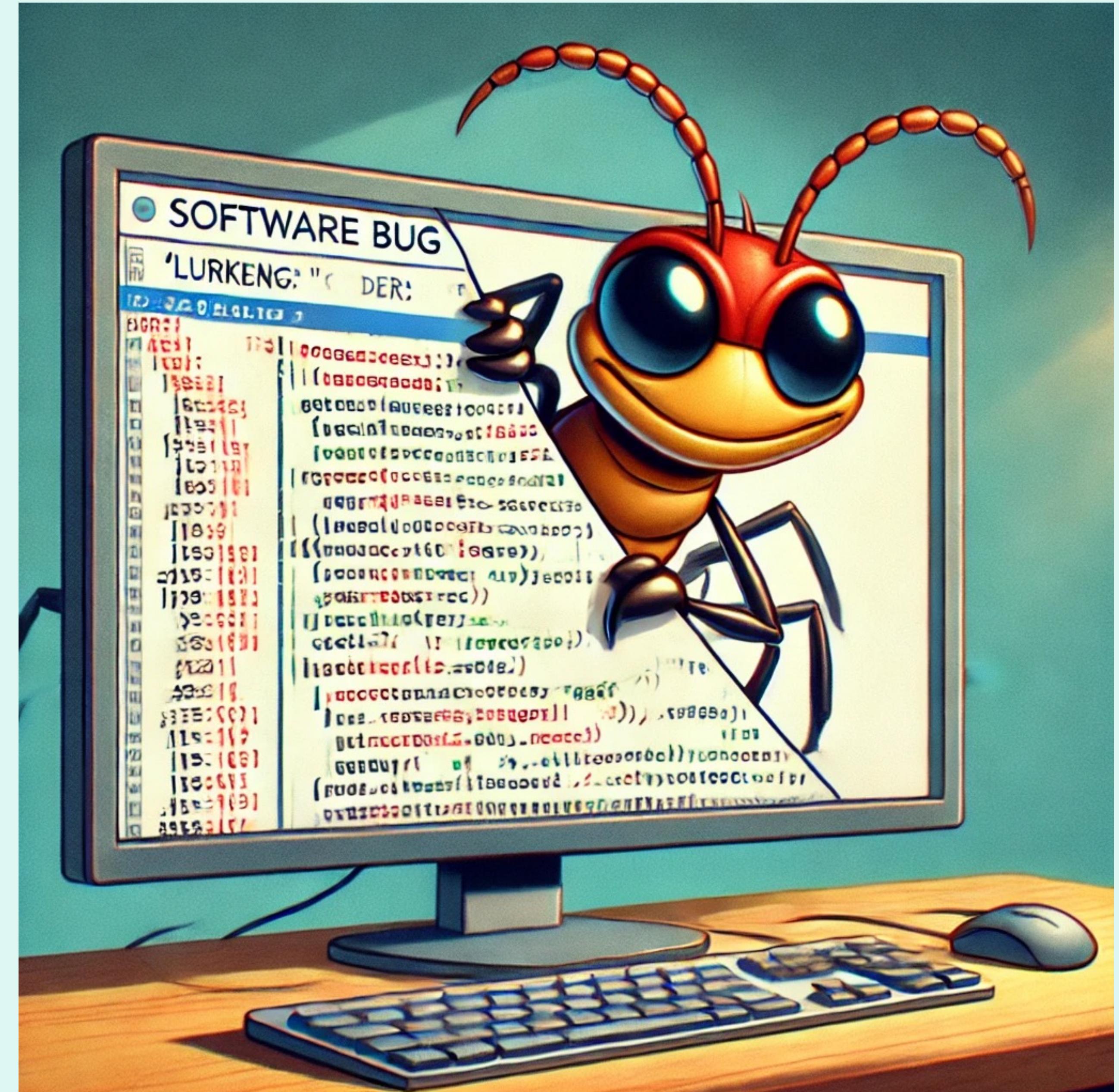


ESTIMATIONS?

Slow down to speed up

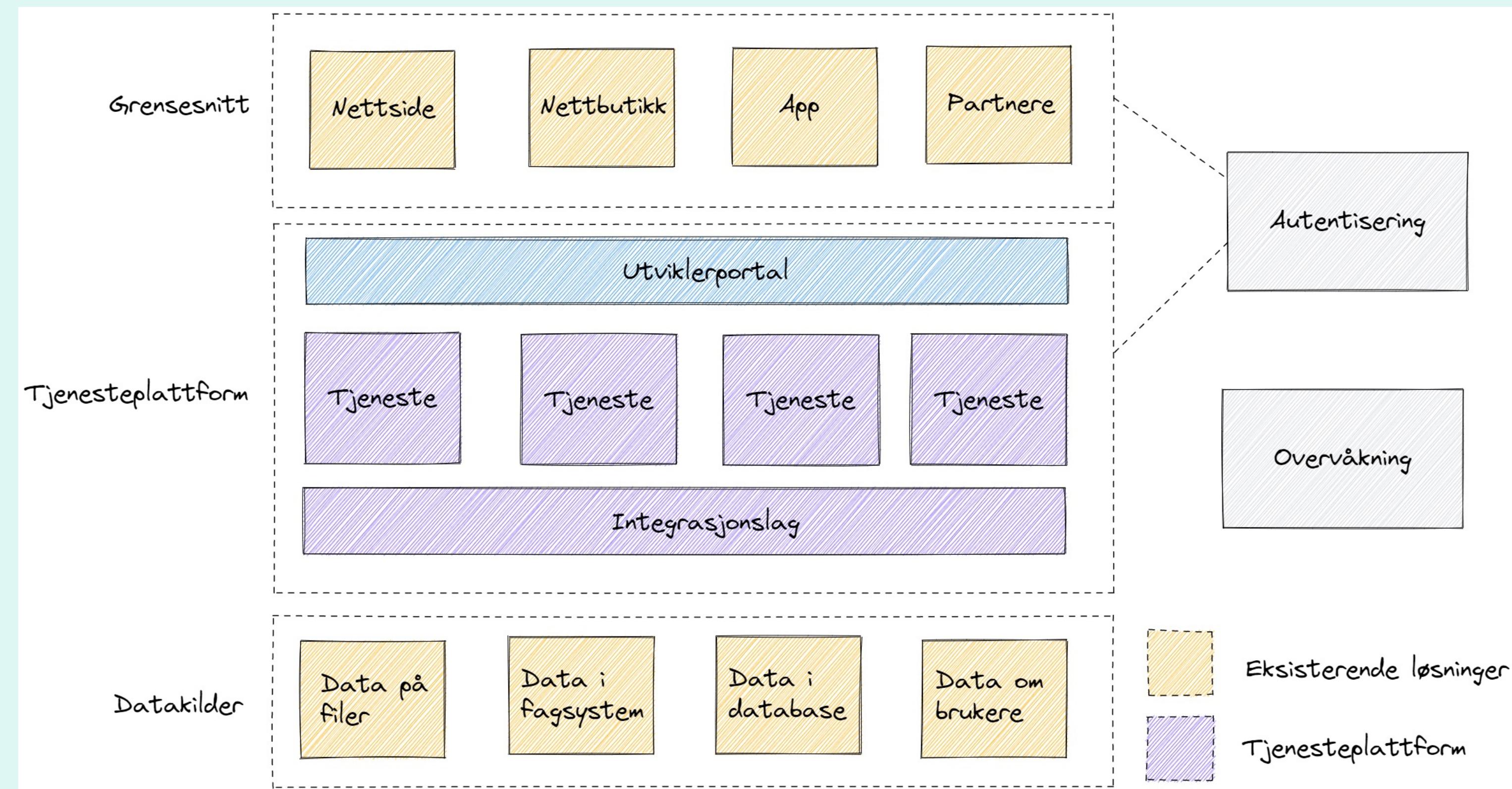


Fix bugs or live with it?

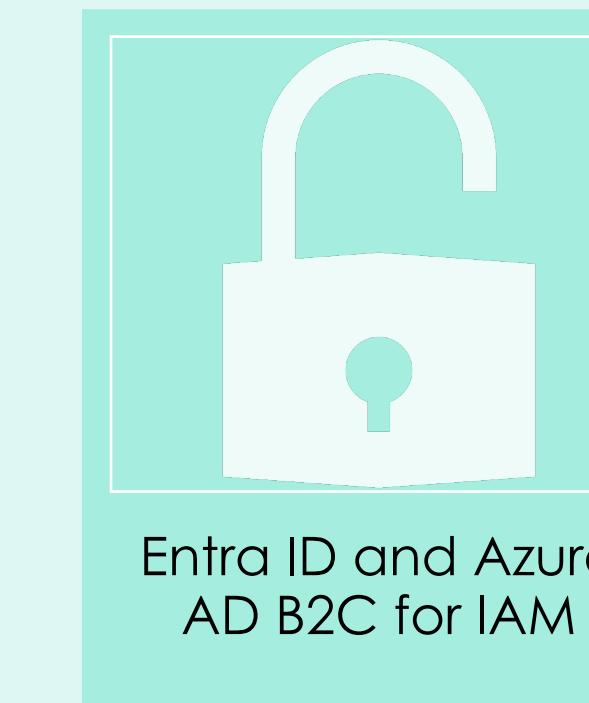
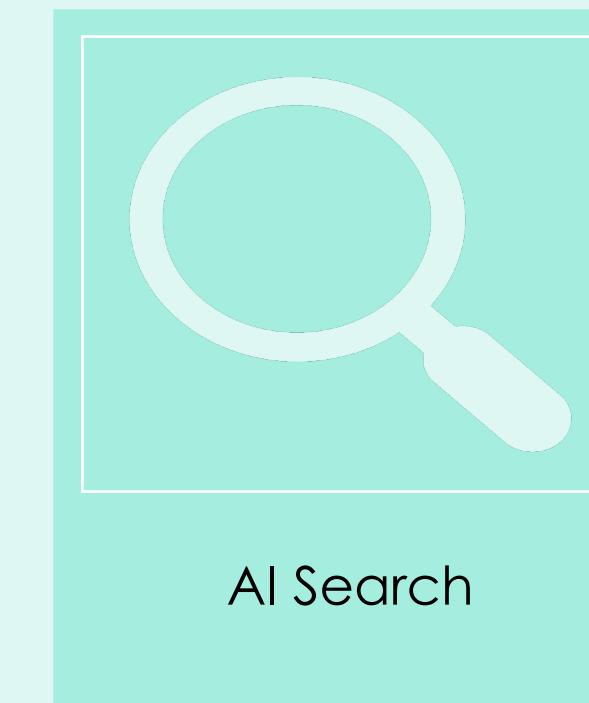
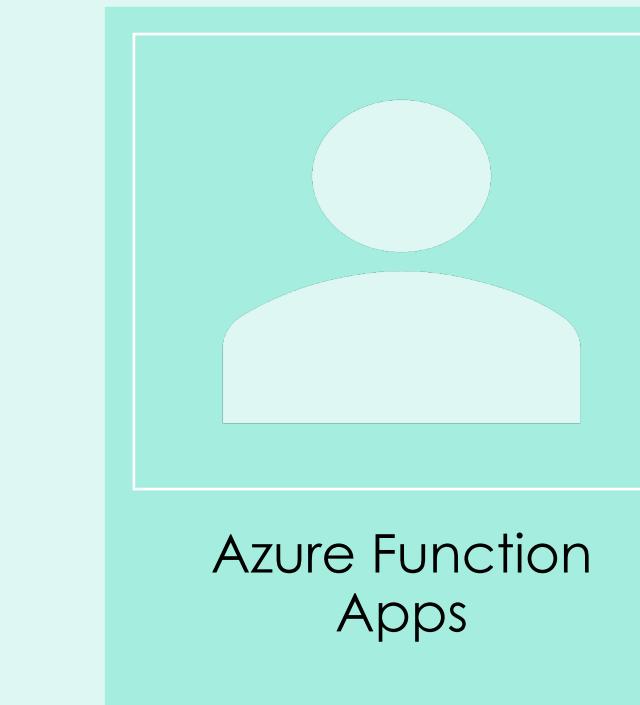
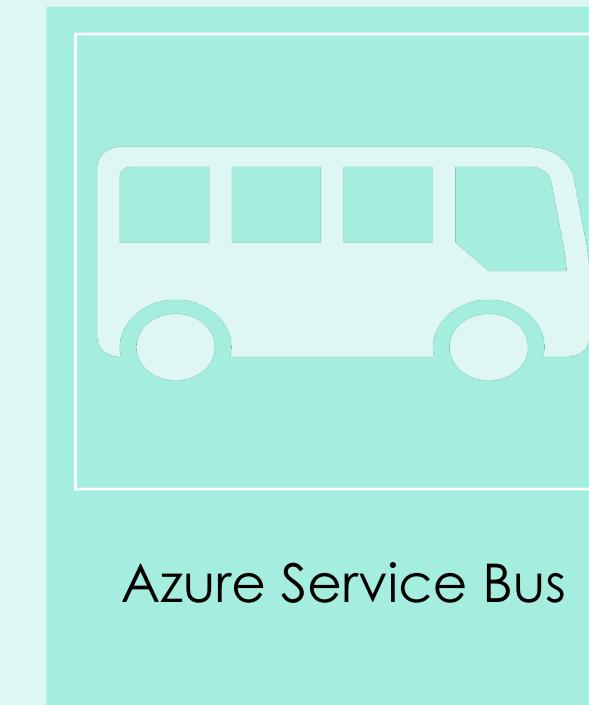
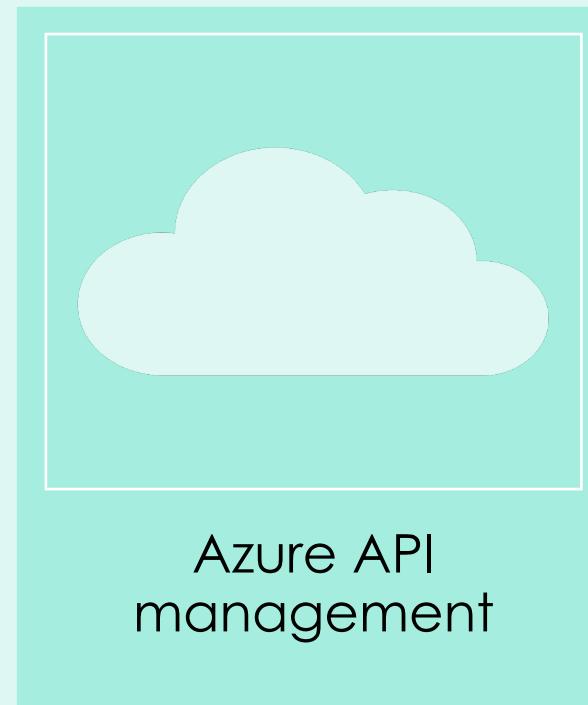


The fun stuff (AKA the tech stack)

Current architecture



Azure services



Infrastructure as Code

[bicep/modules/acn-apiconnection](#)

[bicep/modules/ai-applicationinsights](#)

[bicep/modules/apim-apimanagement](#)

[bicep/modules/apim-functionapp-api](#)

[bicep/modules/apim-namedvalue](#)

[bicep/modules/apim-webapp-api](#)

[bicep/modules/app-appservice](#)

[bicep/modules/app-appservice-project](#)

[bicep/modules/app-full-service](#)

[bicep/modules/app-linuxcontainer](#)

[bicep/modules/app-linuxcontainer-project](#)

[bicep/modules/cta-cosmostableaccount](#)

[bicep/modules/eg-systemtopic](#)

[bicep/modules/eg-systemtopicsubscription](#)

[bicep/modules/fa-functionapp](#)

[bicep/modules/fa-functionapp-prject](#)

Microservices

- .NET 8 with minimal APIs
- Web apps for APIs and business logic
- Function Apps for processing updates

How to structure your microservice

This blog post will describe a way to structure your code for a .NET microservice.

<https://blog.novonet.no/how-to-structure-your-microservice/>

Testing & DevOps



INTEGRATION TESTING



TEST CONTAINERS



AZURE DEVOPS
PIPELINES



APPLICATION INSIGHTS



DASHBOARDS WITH
STATUSES

Summary

- No big picture thinking leads to messy design
- Identifying the patterns and the future
- Shut down applications once you're done with it
- Slow down to speed up
- Azure based stack



Thank you

Leif Larsen

 leifhlarsen