Erik Drolshammer

altran

Arkitekturerfaringer

Whydah

Agenda

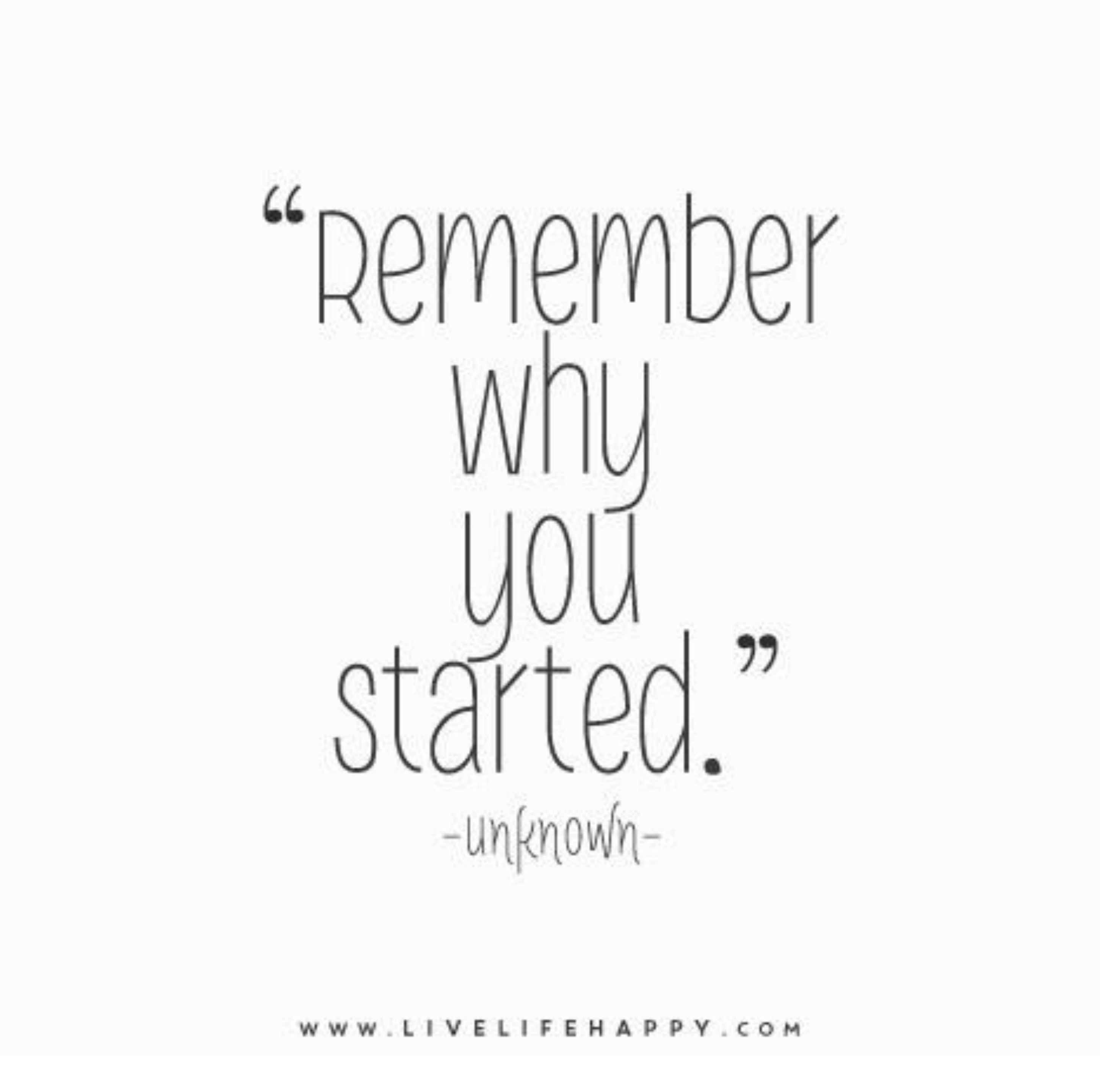
1. Hva er IAM, Whydah
2. Arkitekturgjennomgang

i am me.

Features

- User authentication & authorization
- Single Sign-On
- Application authentication & authorization

- Admin GUI
- Admin API
- High Availability & High Scalability

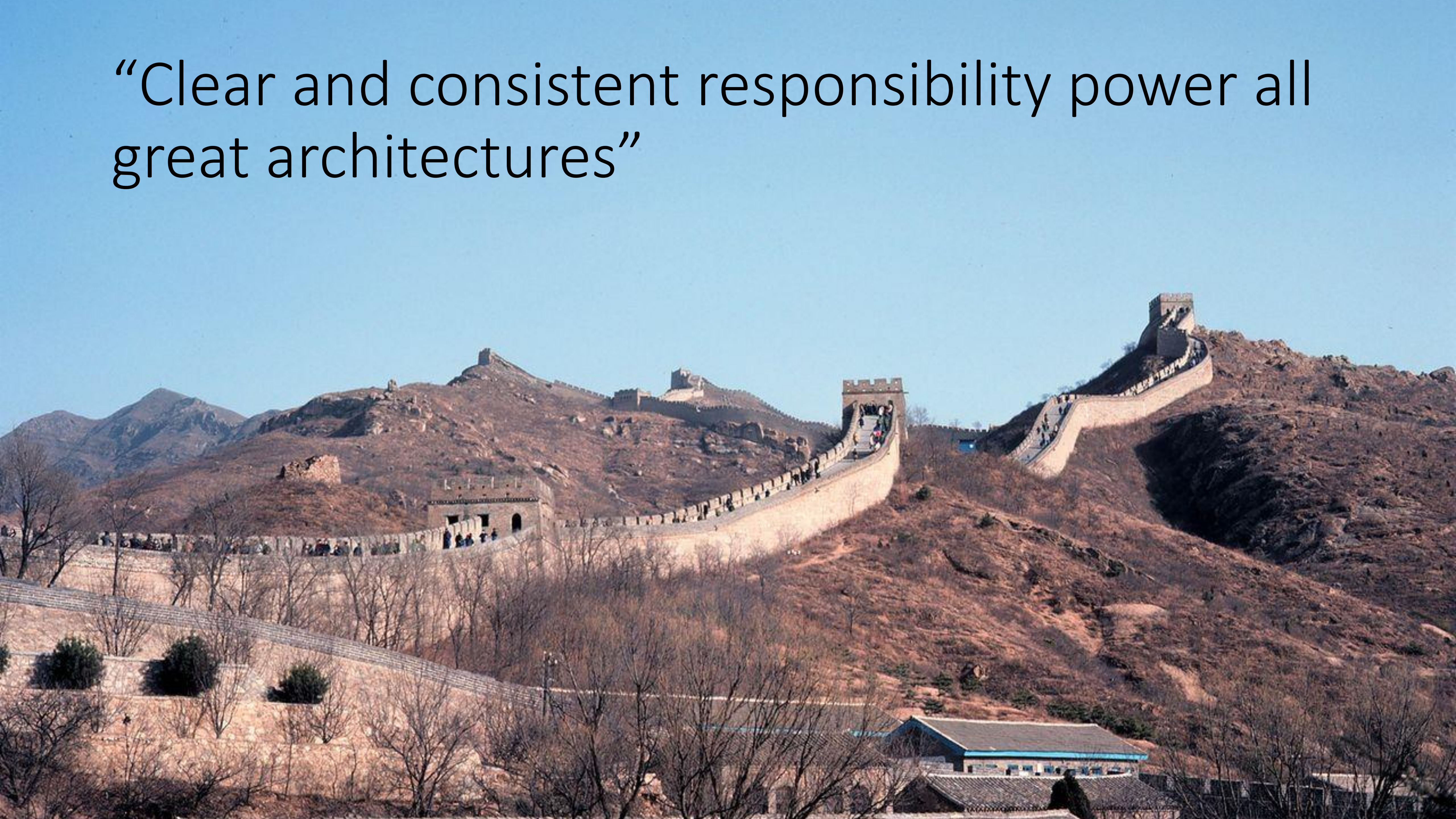


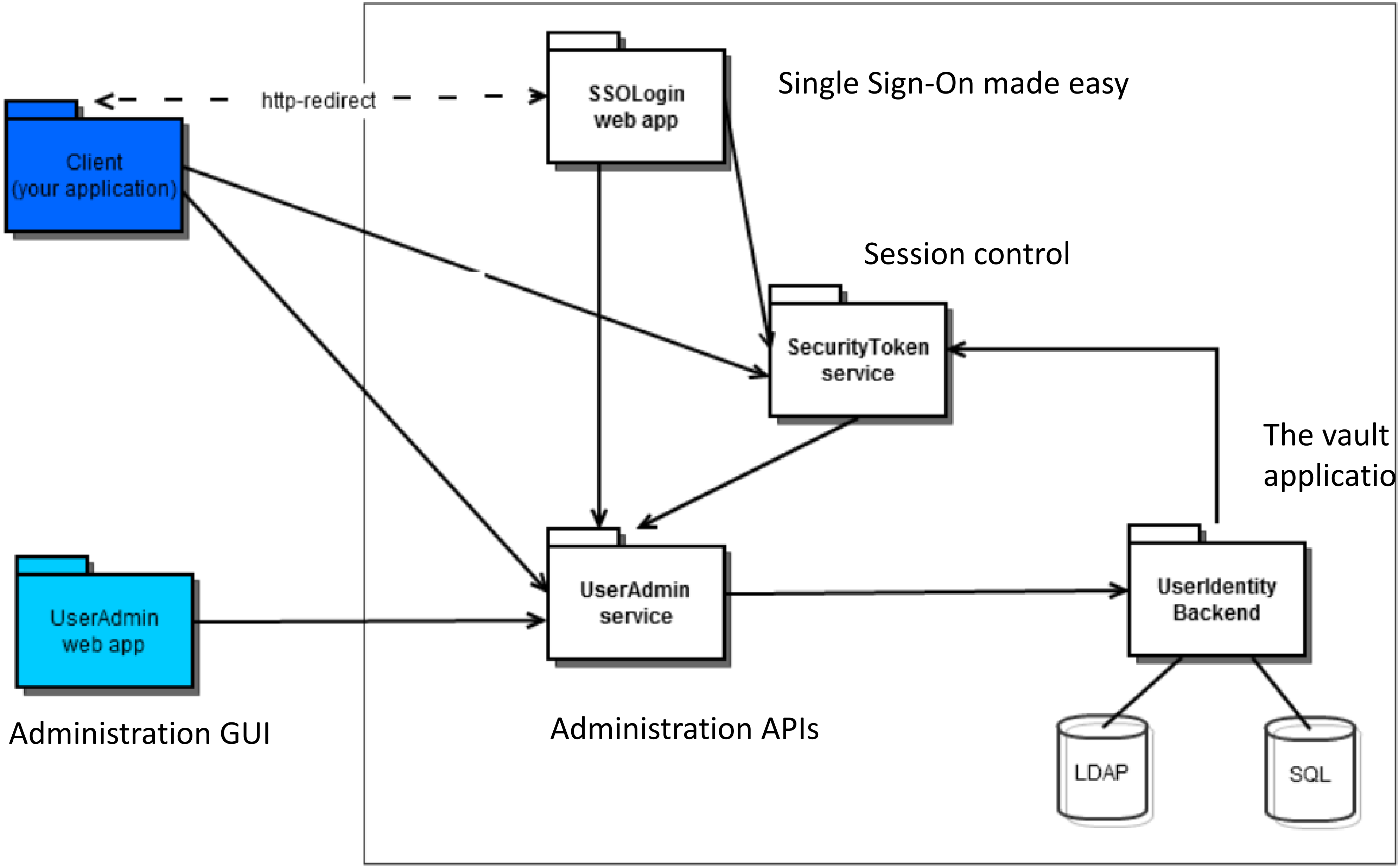
“Remember why you started.”

-unknown-

WWW.LIVELIFEHAPPY.COM

“Clear and consistent responsibility power all great architectures”





Single Sign-On made easy

Session control

The vault of users and applications

Administration GUI

Administration APIs

Guts on the Outside

Inner architecture

” the implementation architecture of the microservices themselves”

Outer architecture

”the platform capabilities you need to help all those simple little microservices (and their DevOps teams) work together to make good on the promises of flexible and scalable development and deployment.”

Gary Olliffe, http://blogs.gartner.com/gary-olliffe/2015/01/30/microservices-guts-on-the-outside/

Characteristics of a Microservice Architecture

1. Componentization via Services
2. (Team) Organized around Business Capabilities
3. Products not Projects
4. Smart endpoints and dumb pipes
5. Decentralized Governance
6. Decentralized Data Management
7. Infrastructure Automation
8. Design for failure
9. Evolutionary Design

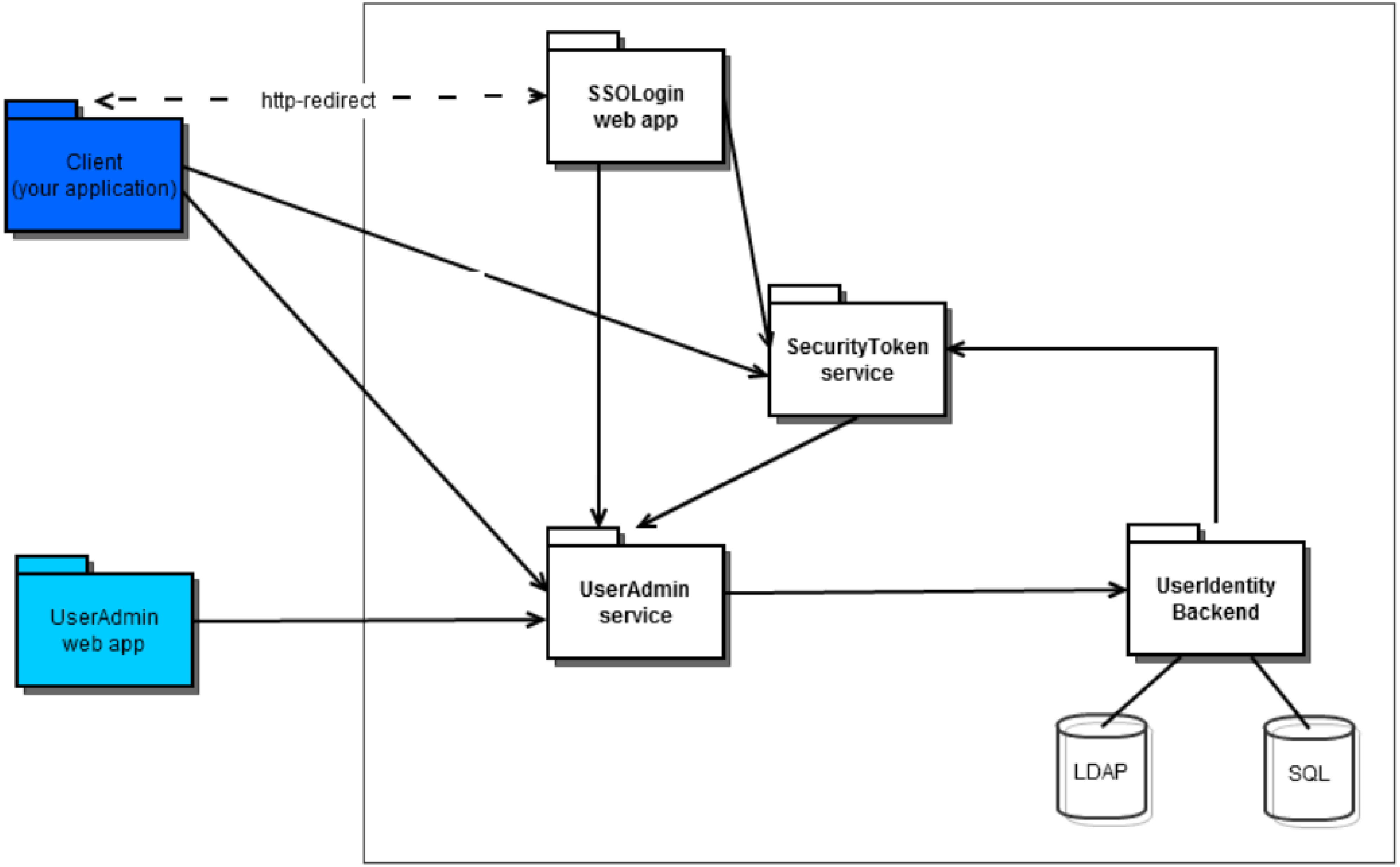
James Lewis

Martin Fowler

http://martinfowler.com/articles/microservices.html

Characteristics of a Microservice Architecture

1. **Componentization via Services**
2. **(Team) Organized around Business Capabilities**
3. **Products not Projects**
4. Smart endpoints and dumb pipes
5. Decentralized Governance
6. Decentralized Data Management
7. Infrastructure Automation
8. Design for failure
9. Evolutionary Design



© 2005, 2006, 2007 by Bjørn Erik Pedersen

Characteristics of a Microservice Architecture

1. Componentization via Services
2. (Team) Organized around Business Capabilities
3. Products not Projects
4. **Smart endpoints and dumb pipes**
5. Decentralized Governance
6. Decentralized Data Management
7. Infrastructure Automation
8. Design for failure
9. Evolutionary Design

🟧 **Portal Tier**
Oracle WebLogic Portal/
Oracle WebCenter Interaction

🟦 **Process Tier**
Oracle WebLogic Integration/
Oracle BPM

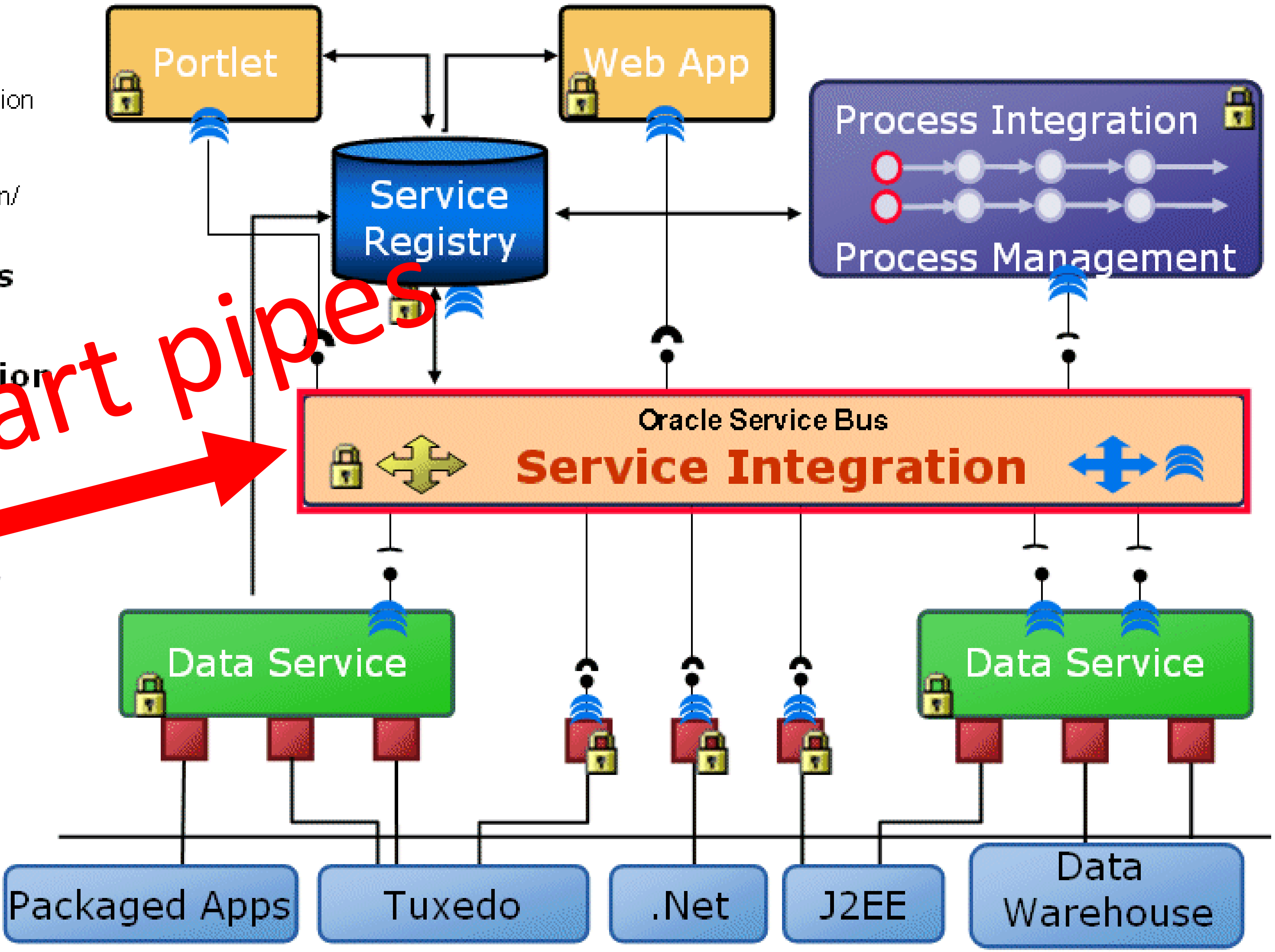
🔒 **Security Services**
Oracle Enterprise Security

▨ **Service Integration**
Oracle Service Bus

🟩 **Data Services**
Oracle Data Services Platf

🟦 **Service Registry**
Oracle Service Registry

End-to-end Web Services Management



Smart pipes

Robustness principle, aka. Postel’s law:

"Be conservative in what you send, be liberal in what you accept"

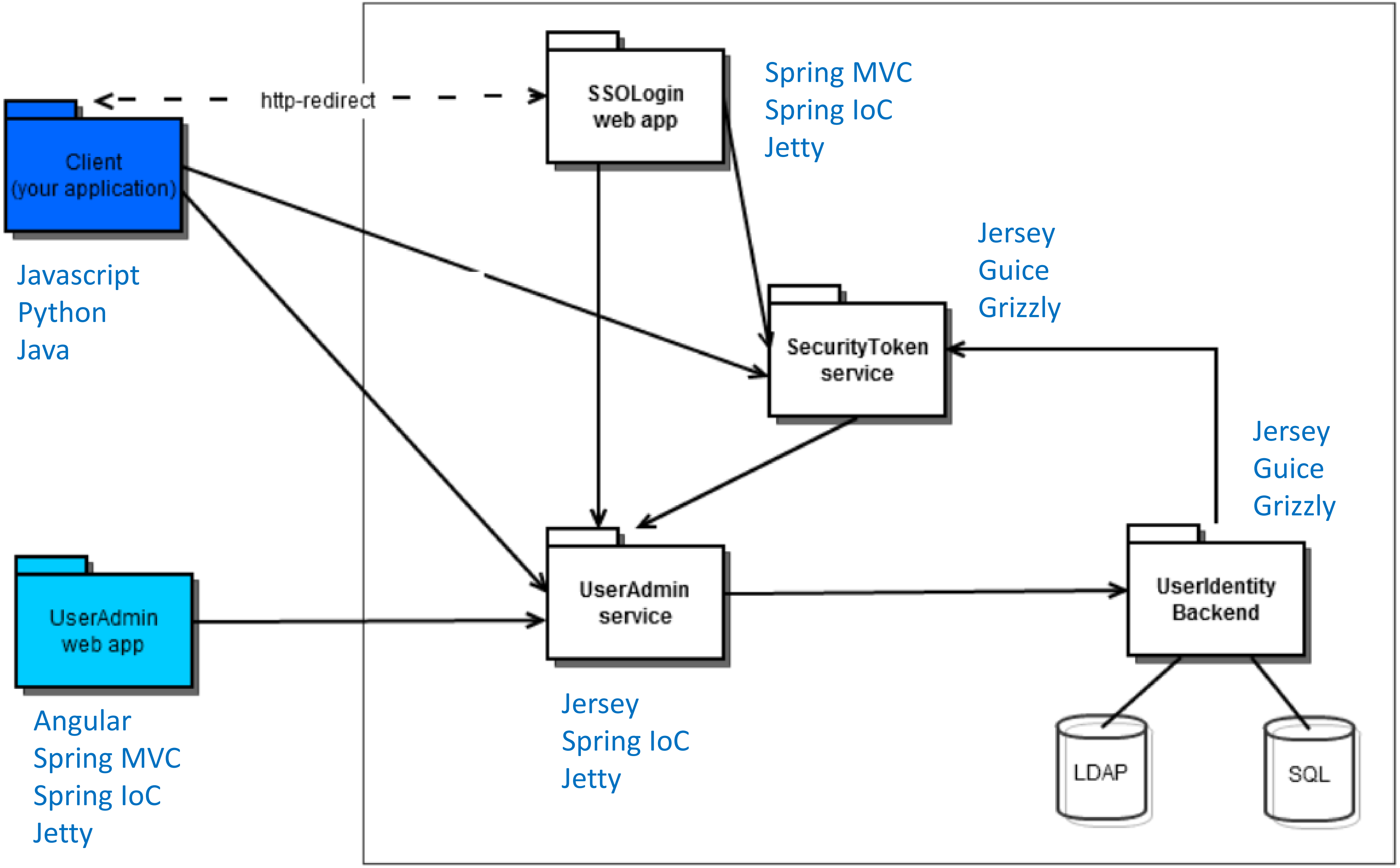
Evolving Service Endpoint

Consumer-Driven Contracts

Hypermedia as the Engine of Application State (HATEOAS)

Characteristics of a Microservice Architecture

1. Componentization via Services
2. (Team) Organized around Business Capabilities
3. Products not Projects
4. Smart endpoints and dumb pipes
5. **Decentralized Governance**
6. Decentralized Data Management
7. Infrastructure Automation
8. Design for failure
9. Evolutionary Design



Javascript
Python
Java

Spring MVC
Spring IoC
Jetty

Jersey
Guice
Grizzly

Jersey
Guice
Grizzly

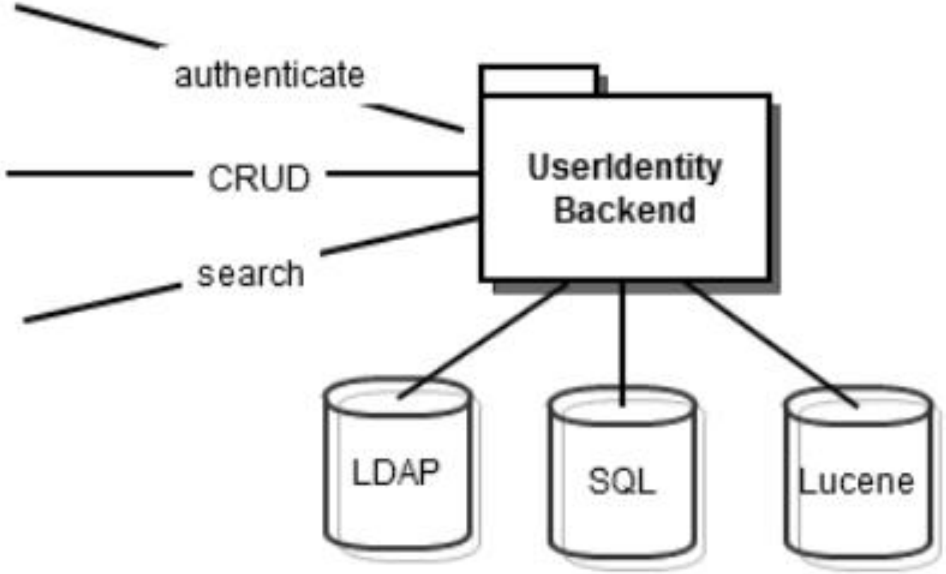
Angular
Spring MVC
Spring IoC
Jetty

Jersey
Spring IoC
Jetty

Characteristics of a Microservice Architecture

1. Componentization via Services
2. (Team) Organized around Business Capabilities
3. Products not Projects
4. Smart endpoints and dumb pipes
5. Decentralized Governance
6. **Decentralized Data Management**
7. Infrastructure Automation
8. Design for failure
9. Evolutionary Design

Polyglot persistence



Lucene

OpenLDAP™
http://www.OpenLDAP.org

Apache Directory™

Microsoft® SQL Server™

MySQL®

HyperSQL
HSQLDB - 100% Java Database

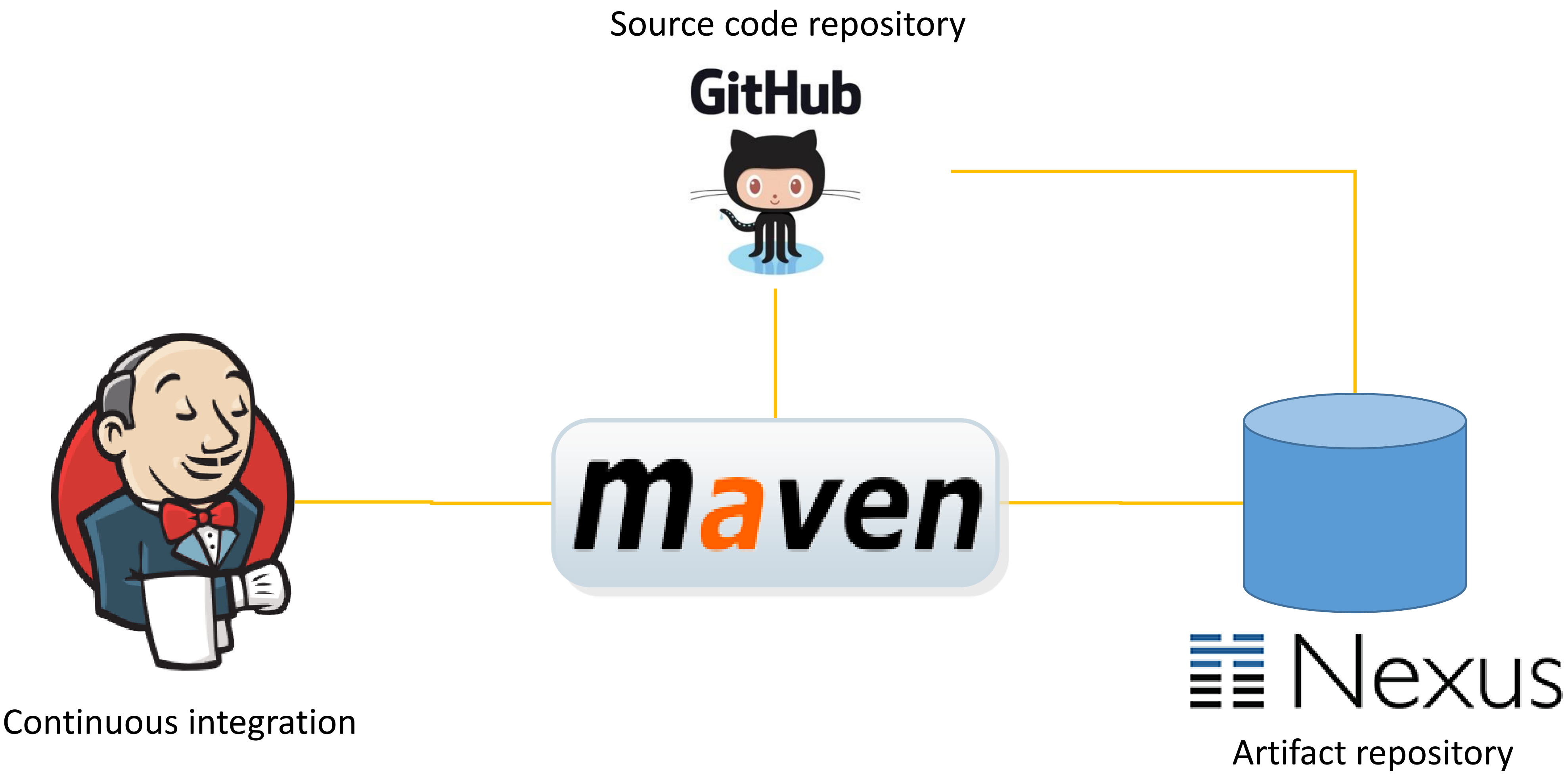
Characteristics of a Microservice Architecture

1. Componentization via Services
2. (Team) Organized around Business Capabilities
3. Products not Projects
4. Smart endpoints and dumb pipes
5. Decentralized Governance
6. Decentralized Data Management
7. **Infrastructure Automation**
8. Design for failure
9. Evolutionary Design

… skal Stig snakke mer om etterpå

1. Enterprise Maven Infrastructure
2. Provisjonering: Ansible + Amazon EC2
3. *Automated deployment (*wget, upstart, java –jar)
4. *Automatic (automated* + cron)

Enterprise Maven Infrastructure

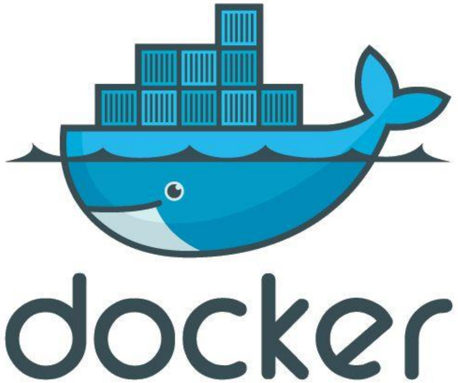


Source code repository

Continuous integration

Artifact repository

Provisjonering



Elastic Compute Cloud (EC2)

Automated deployment

Automatic deployment

Job scheduler

aka. Continuous delivery

Characteristics of a Microservice Architecture

1. Componentization via Services
2. (Team) Organized around Business Capabilities
3. Products not Projects
4. Smart endpoints and dumb pipes
5. Decentralized Governance
6. Decentralized Data Management
7. Infrastructure Automation
8. **Design for failure**
9. Evolutionary Design

Tjenester vil feile!

- Tjenester må kunne startes i vilkårlig rekkefølge
- Retry-mekanismer hvis en tjeneste er utilgjengelig.
- Håndtere feilsituasjoner
- Robuste start/stopp-script

- Saklig logging under oppstart
- Logge inngående og utgående kall
  - Bruk log levels aktivt

SLF4J
Simple Logging
Facade for Java

LOGBack™
The Generic, Reliable
Fast & Flexible
Logging Framework

HYSTRIX
DEFEND YOUR APP

Infrastruktur

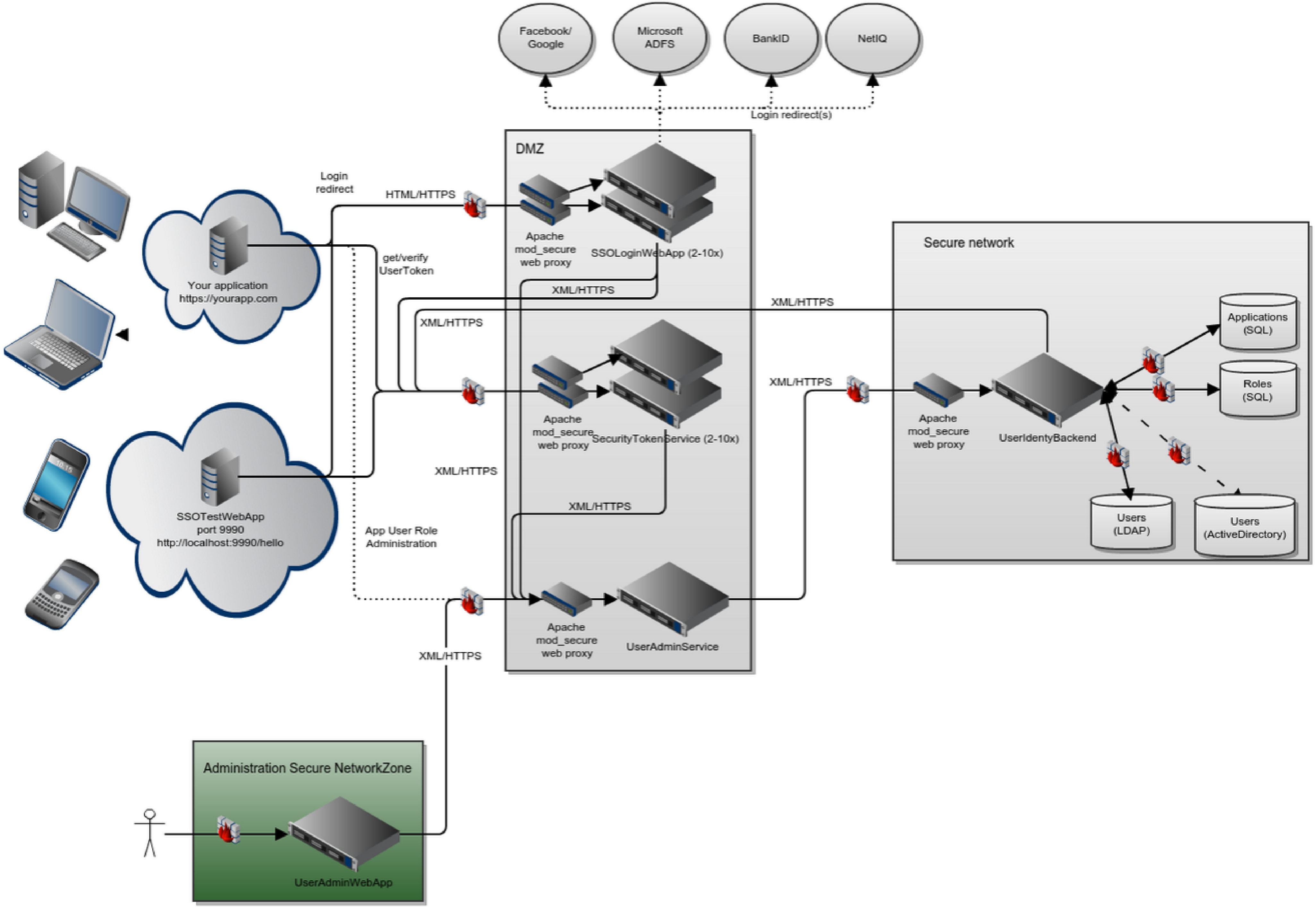
- **HTTPS overalt, alltid!**
- Web proxy
- Lastbalanserer
- Discovery
- Hazelcast for HA

APACHE
HTTP SERVER

amazon
web services™

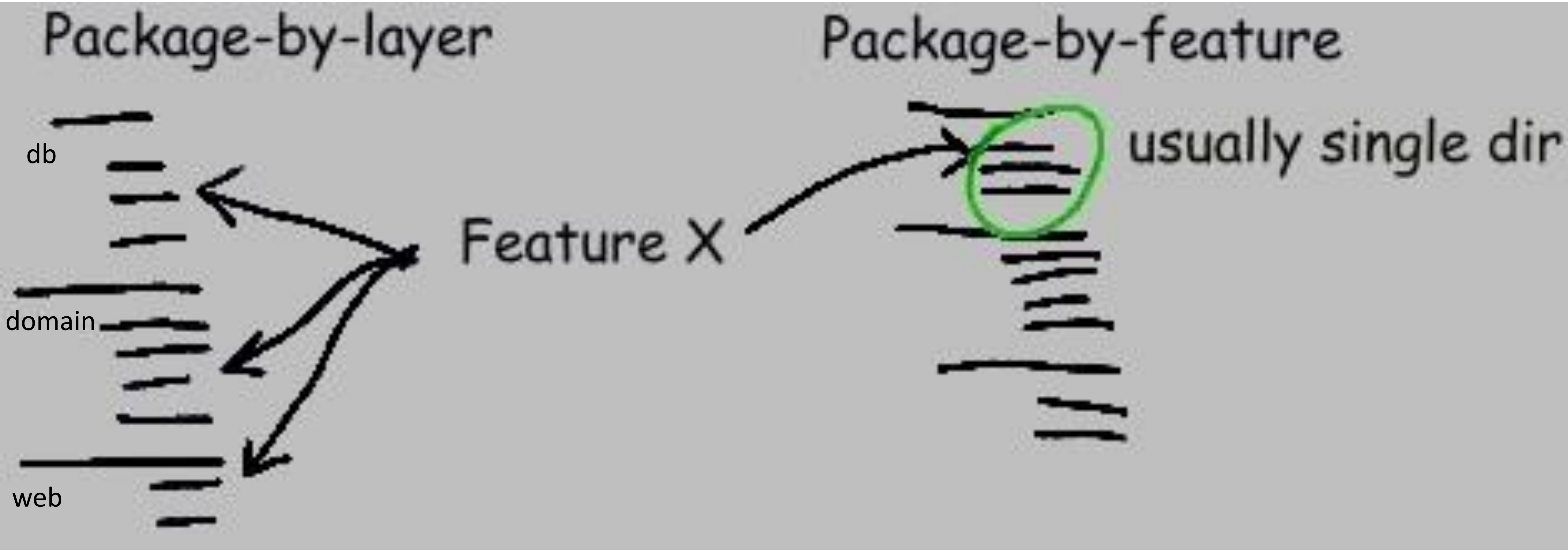
Elastic Load Balancing

hazelcast

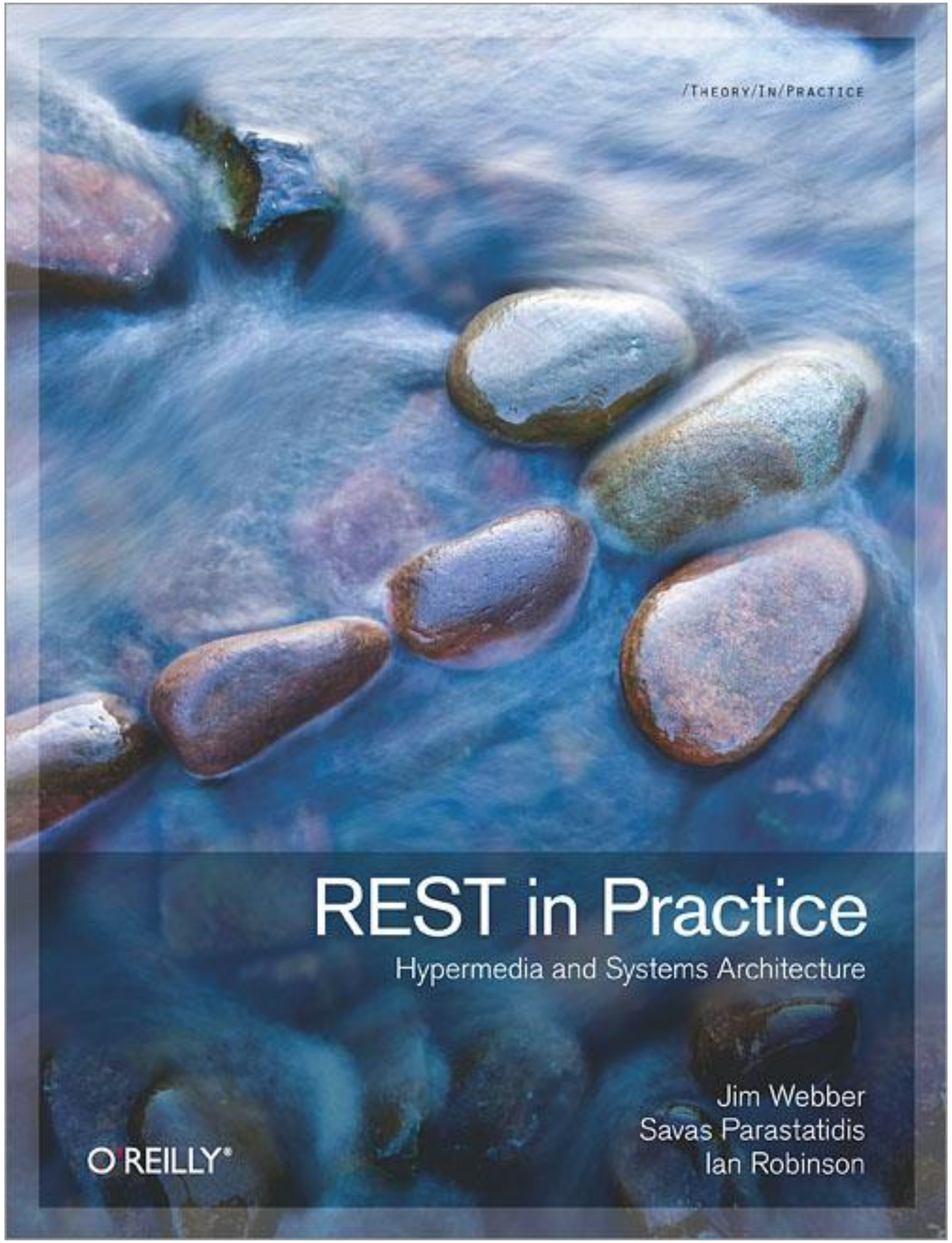

Facebook/ Google
Microsoft ADFS
BankID
NetIQ
Login redirect(s)
DMZ
Login redirect
HTML/HTTPS
Apache mod_secure web proxy
SSOLoginWebApp (2-10x)
Your application https://yourapp.com
get/verify UserToken
XML/HTTPS
XML/HTTPS
XML/HTTPS
Secure network
Applications (SQL)
Roles (SQL)
XML/HTTPS
Apache mod_secure web proxy
SecurityTokenService (2-10x)
Apache mod_secure web proxy
UserIdentyBackend
XML/HTTPS
SSOTestWebApp port 9990 http://localhost:9990/hello
XML/HTTPS
App User Role Administration
Users (LDAP)
Users (ActiveDirectory)
Apache mod_secure web proxy
UserAdminService
XML/HTTPS
Administration Secure NetworkZone
UserAdminWebApp

Characteristics of a Microservice Architecture

1. Componentization via Services
2. (Team) Organized around Business Capabilities
3. Products not Projects
4. Smart endpoints and dumb pipes
5. Decentralized Governance
6. Decentralized Data Management
7. Infrastructure Automation
8. Design for failure
9. **Evolutionary Design**



http://www.javapractices.com/topic/TopicAction.do?Id=205,
http://www.web4j.com



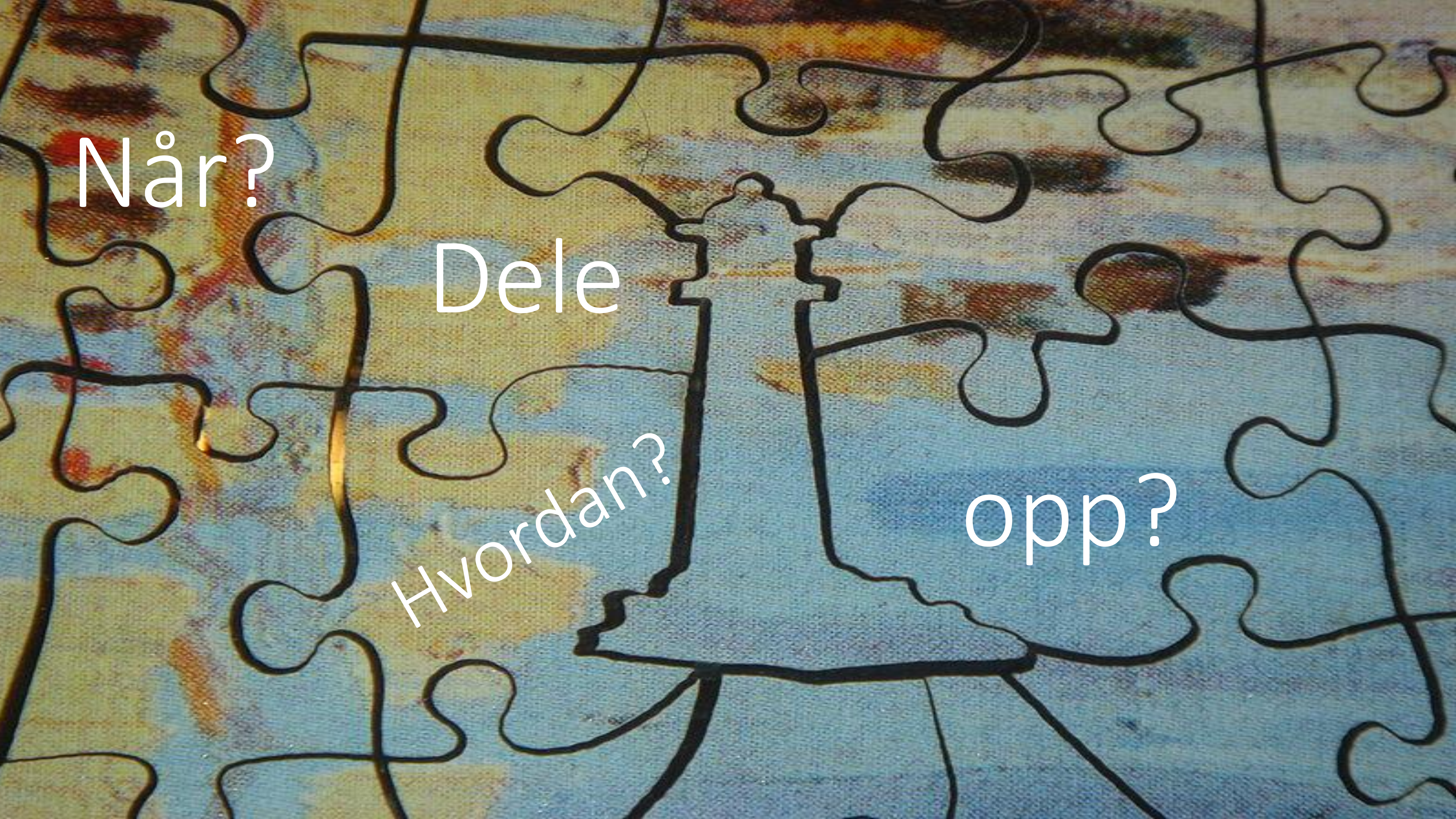
/Theory/In/Practice

REST in Practice

Hypermedia and Systems Architecture

Jim Webber
Savas Parastatidis
Ian Robinson

O'REILLY®



Når?

Dele

Hvordan?

opp?

Erik Drolshammer
erik@fjas.no
@Sherriff1