CADEC 2016 - MICROSERVICES AND DOCKER CONTAINERS

MAGNUS LARSSON

2016-01-27 | CALLISTAENTERPRISE.SE

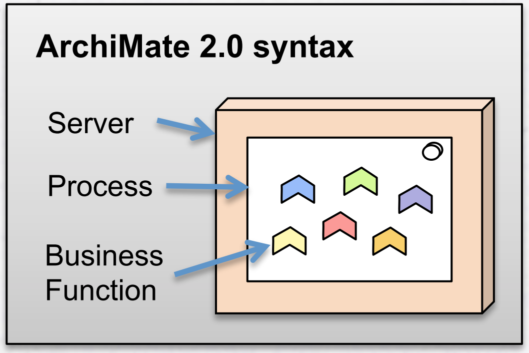
CALLISTA
— ENTERPRISE —

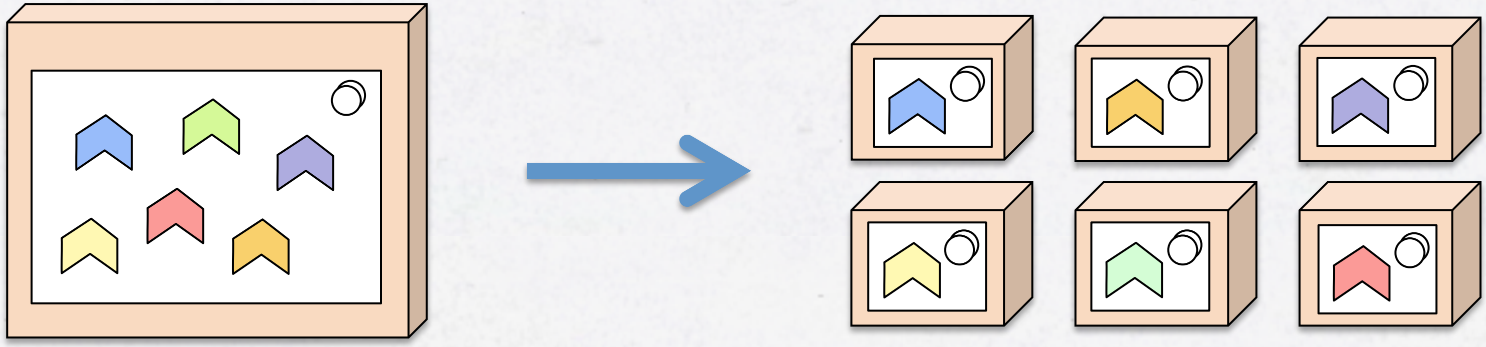
AGENDA

- Microservices in reality…
  - Managing a system landscape with many microservices
  - Very little high level architecture stuff…
  - Nothing on organizational aspects, Conway's law et al…
- Demo, demo, demo!
- Reference Case
- Summary

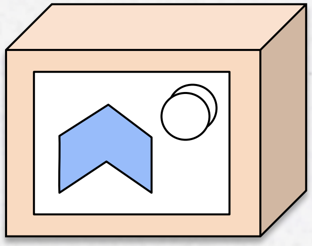
MICROSERVICES IN 5 SEC...

- First heard of 2011
- Divide the monolith!
- Easier to release
- Easier to manage


ArchiMate 2.0 syntax



WHAT'S A MICROSERVICE?

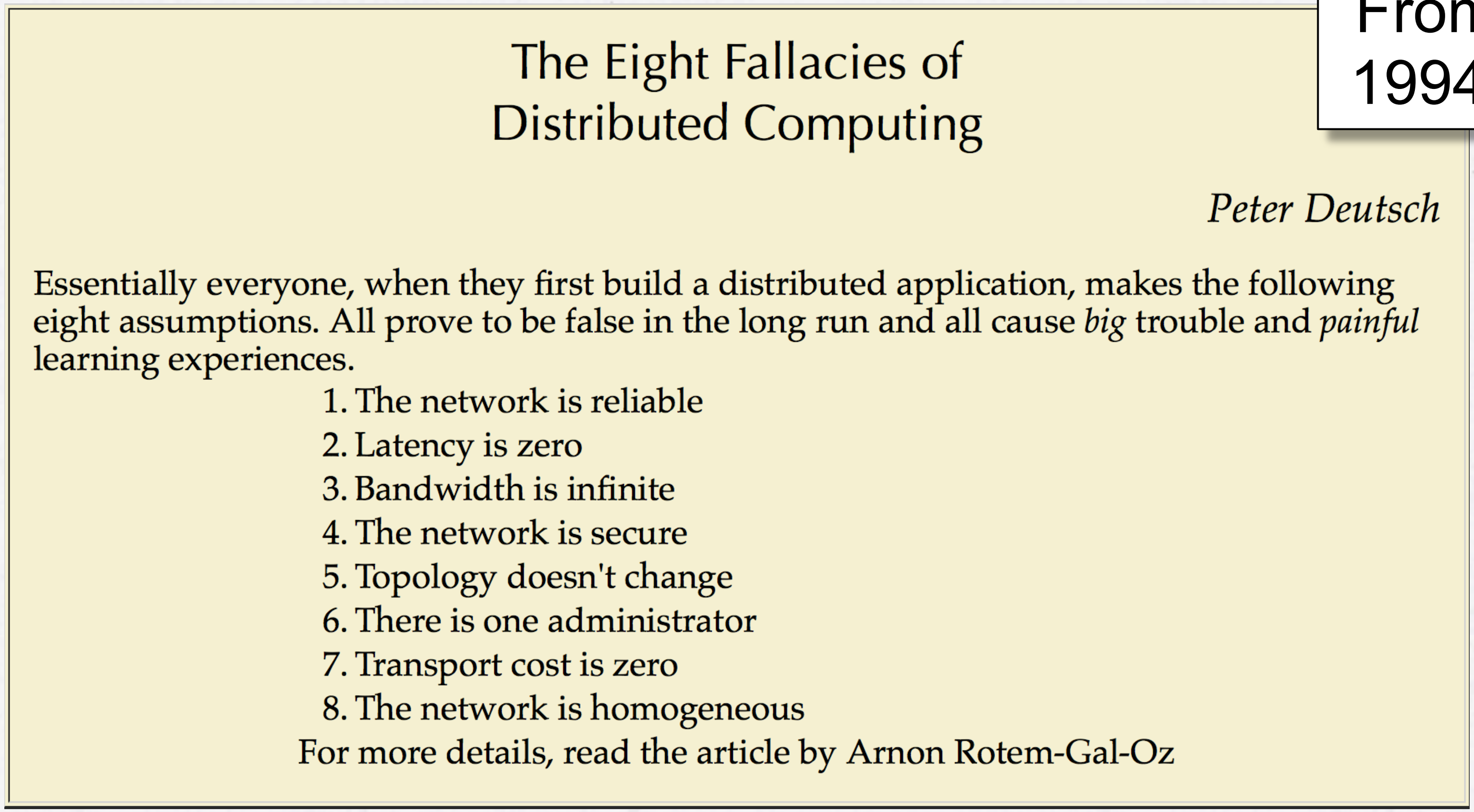


- Autonomous software component
- Share nothing architecture
- Deployed as a runtime processes
- Small enough to fit in the head of a developer
- Big enough to avoid unacceptable latency and data inconsistency…

➜ A group of microservices form a Distributed System

A DISTRIBUTED SYSTEM, SO WHAT'S THE PROBLEM?

From 1994!



The Eight Fallacies of Distributed Computing

Peter Deutsch

Essentially everyone, when they first build a distributed application, makes the following eight assumptions. All prove to be false in the long run and all cause *big* trouble and *painful* learning experiences.

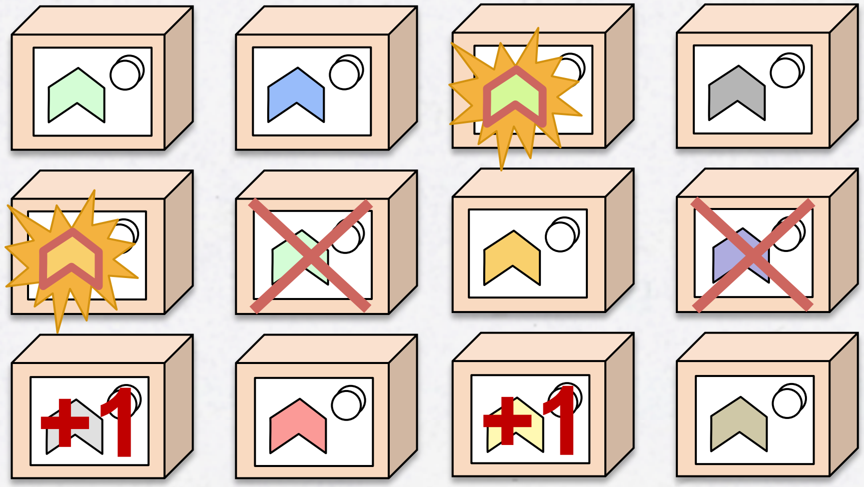
1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

For more details, read the article by Arnon Rotem-Gal-Oz

Source: https://blogs.oracle.com/jag/resource/Fallacies.html

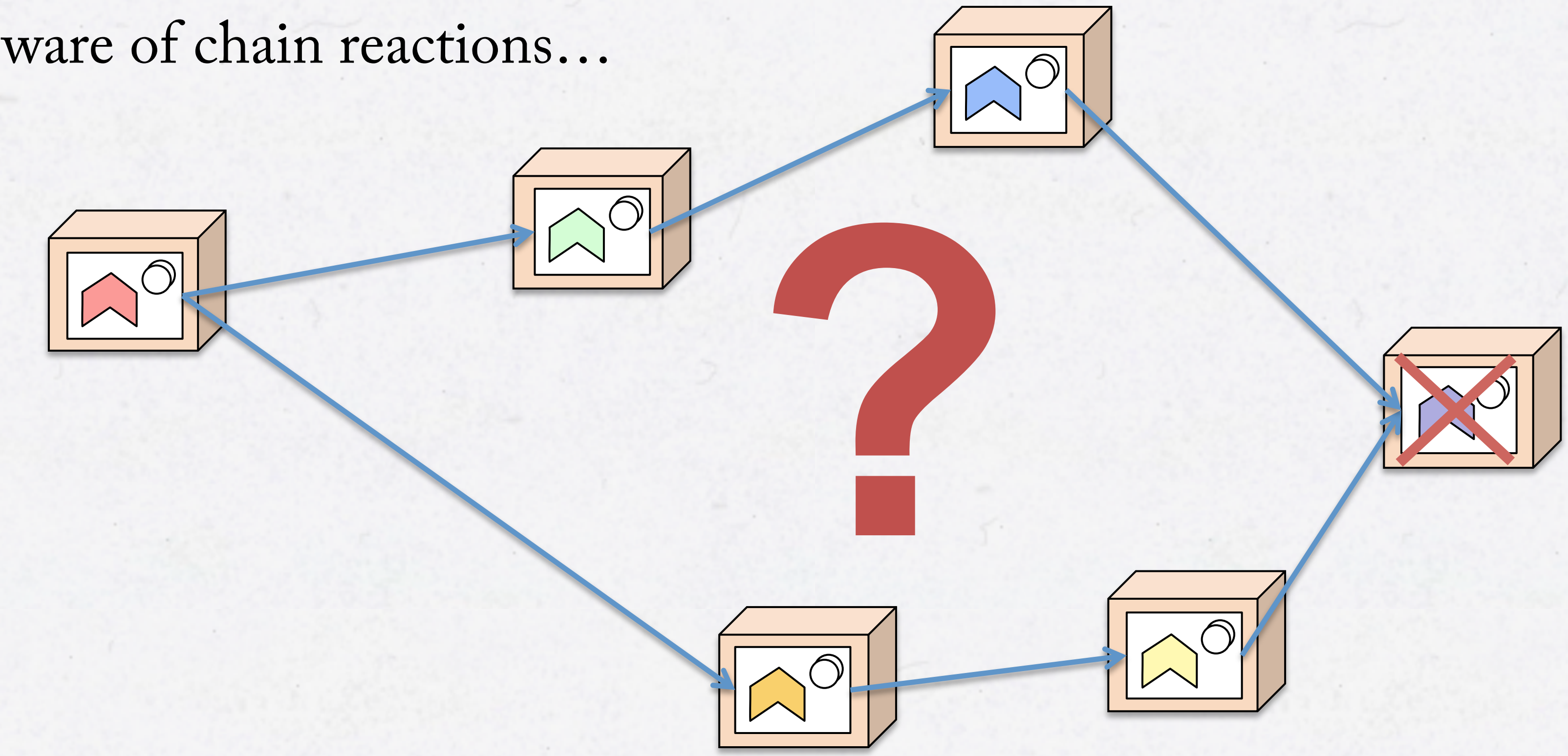
MANAGING LARGE NUMBERS OF MICROSERVICES…

- There is always something (not so good) going on…



RESILIENCE AND PREVENTION OF "*CHAIN OF FAILURE*"

- Beware of chain reactions…



MANAGING LARGE NUMBERS OF MICROSERVICES CAN BE A NIGHTMARE!

1. Where are my microservices and are they ok?
2. How to detect and route calls to microservice instances?
3. How to prevent "*chain of failure*" and recover from errors?
4. How to configure all microservices?
5. How to protect internal API's and secure public API's?
6. How to install, scale, restart, upgrade and replace microservices?

OPEN SOURCE TO THE RESCUE

- Netflix OSS
  - Since 2011, Netflix has been releasing components of their cloud platform as free and open source software
  - Obviously proven in battle…
- Spring Cloud - 2014
  - Packages and simplifies use of Netflix OSS
  - Based on Spring Boot and “*convention over configuration*”

```
dependencies {
  compile(”...spring-cloud-starter-eureka”)
  compile("...spring-cloud-starter-hystrix")
  compile("...spring-cloud-starter-config")
  compile(”...spring-cloud-starter-oauth2")
```

tio

```
@EnableDiscoveryClient
@EnableCircuitBreaker
@EnableResourceServer
@SpringBootApplication
public class MyService {
```

CAPABILITIES PROVIDED BY SPRING-CLOUD

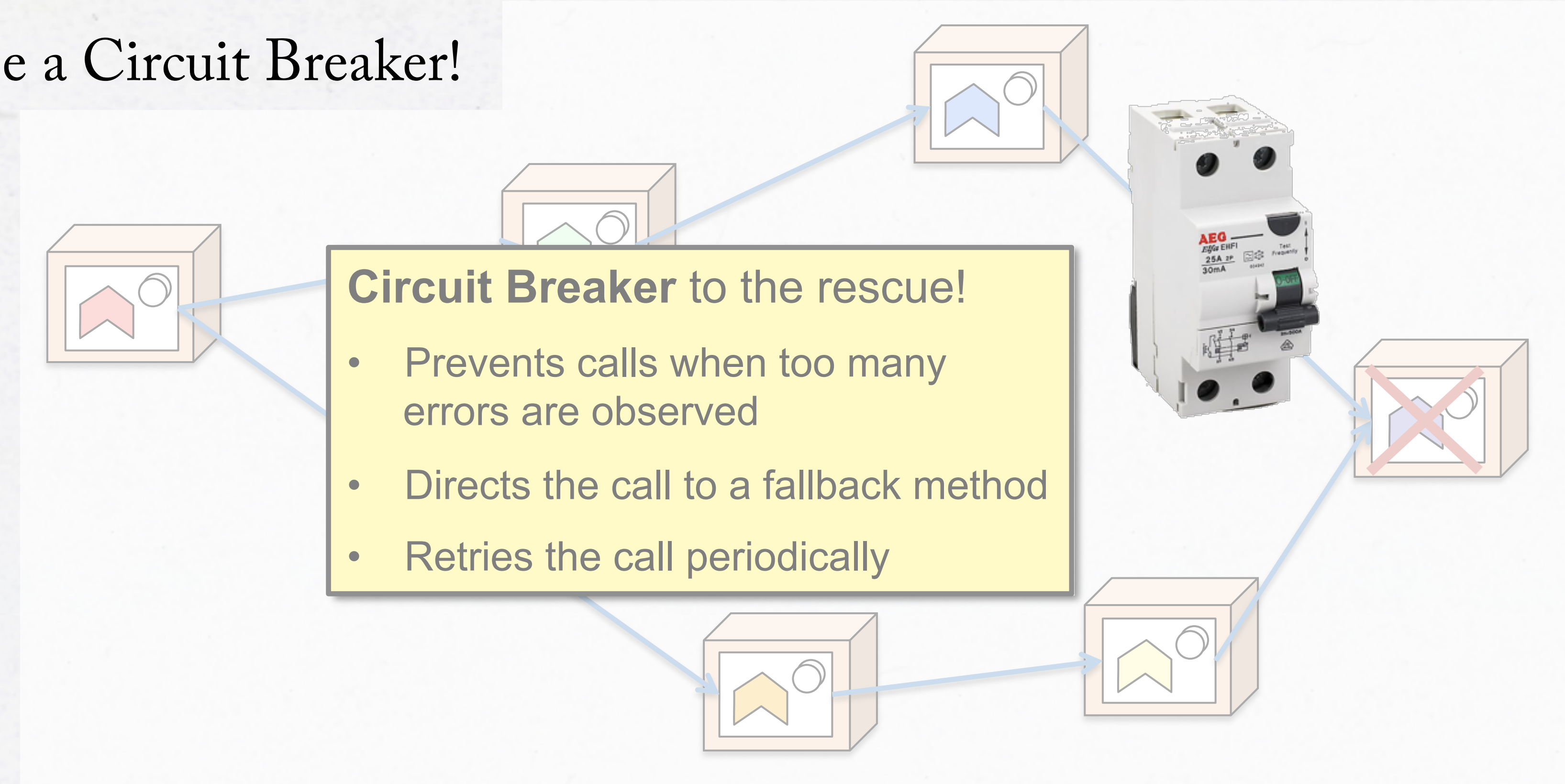
1. Discovery Server
   - New microservices can auto-register at startup
2. Client-Side Load Balancing
   - Clients can detect new instances as they are started up
3. Circuit Breaker
   - Prevent problems with chain of failures
4. Config Server
   - Centralized management of configuration
5. Event Bus
   - Lightweight message broker
6. Edge Server
   - Front door for all external requests
7. OAuth Server
   - Lightweight OAuth Server for development and test

CAPABILITIES PROVIDED BY SPRING-CLOUD

1. Discovery Server	1. Netflix Eureka, Consul, Etcd, Zookeeper
2. Client-Side Load Balancing	2. Netflix Ribbon
3. Circuit Breaker	3. Netflix Hystrix + Dashboard
4. Config Server	4. Git, Consul, Etcd, Zookeeper
5. Event Bus	5. RabbitMQ, Consul, Redis, Kafka
6. Edge Server	6. Netflix Zuul
7. OAuth Server	7. Spring Security OAuth

RESILIENCE - HOW TO PREVENT "*CHAIN OF FAILURE*" AND RECOVER?

- Use a Circuit Breaker!



Circuit Breaker to the rescue!

- Prevents calls when too many errors are observed
- Directs the call to a fallback method
- Retries the call periodically

HOW TO INSTALL, SCALE, RESTART, UPGRADE AND REPLACE SERVICES?

- 2008 - Linux Containers (LXC)
  - Lightweight Virtual Machines
  - Starts very fast, <1s

- 2013 – Docker
  - Makes Linux Containers easy to use…
    (similar to Netflix OSS and Spring-Cloud)

- A docker container runs a separate microservice instance

DOCKER TOOLS

- docker machine
  - Create Linux hosts with Docker pre-installed

```
$ docker-machine create\
  --driver virtualbox | azure | google | amazonec2 | ...
  --provider specific parameters...
```

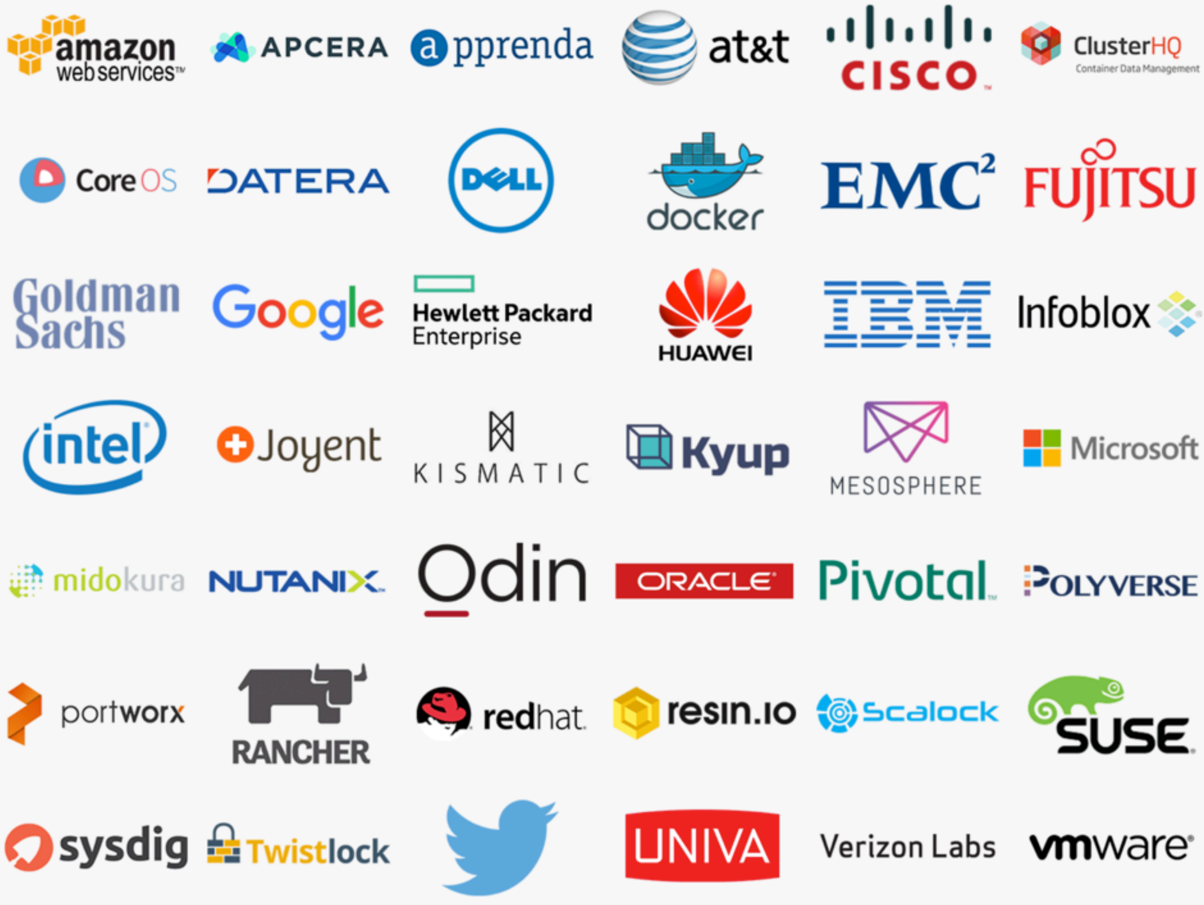
DOCKER TOOLS

- docker compose
  - Manage a group of Docker containers

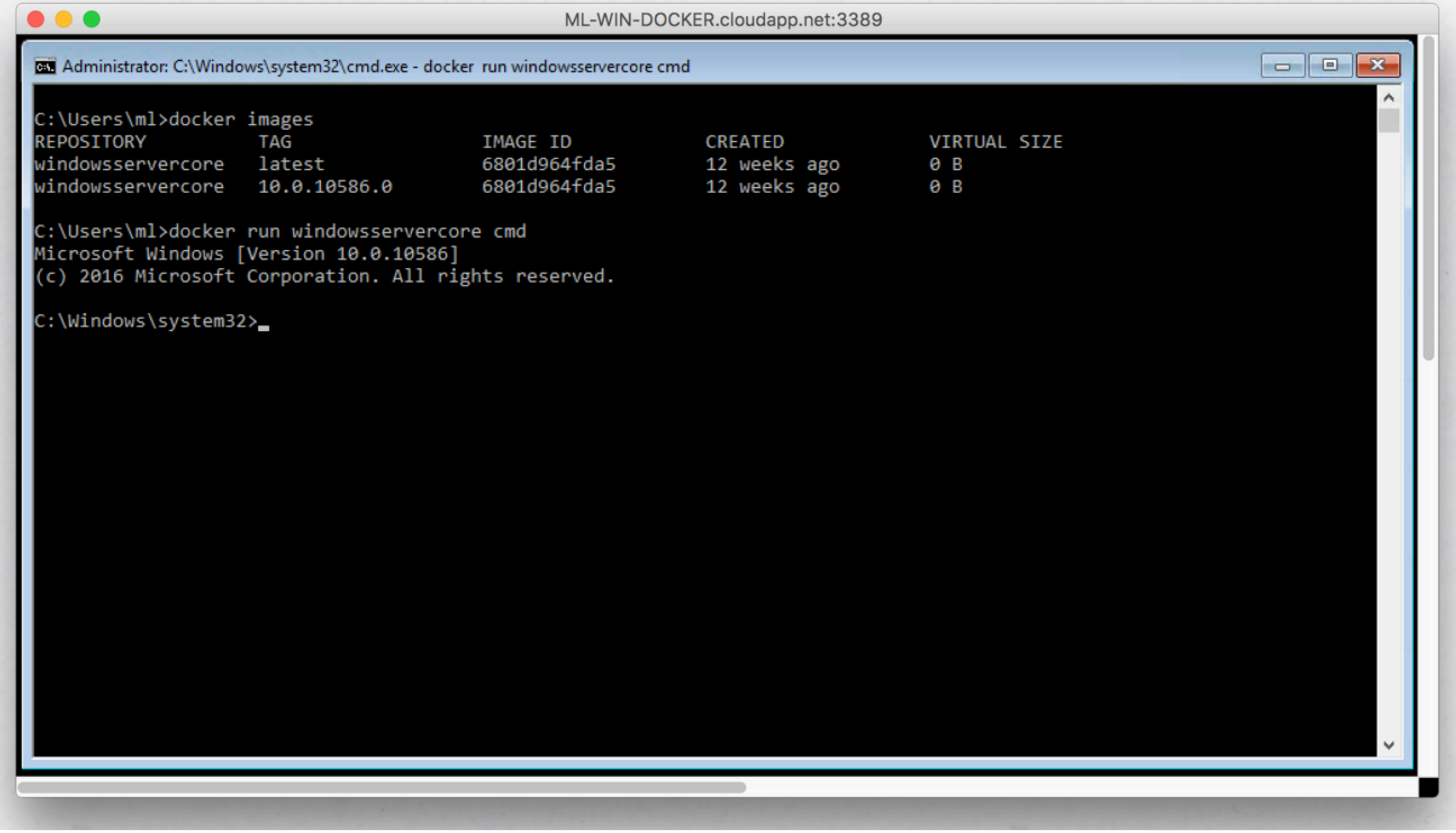
```yaml
rabbitmq:
  image: rabbitmq:3-management
  restart: always

discovery:
  image: magnuslarsson/ms-blog-discovery-server
  restart: always
  ports:
    - "8761:8761"

pro:
  image: magnuslarsson/ms-blog-product-service
  restart: always
  links:
    - discovery
```

NEW STANDARD – THE OPEN CONTAINER INITIATIVE

OPEN CONTAINER INITIATIVE



DOCKER - NO LONGER LINUX CONTAINERS ONLY…



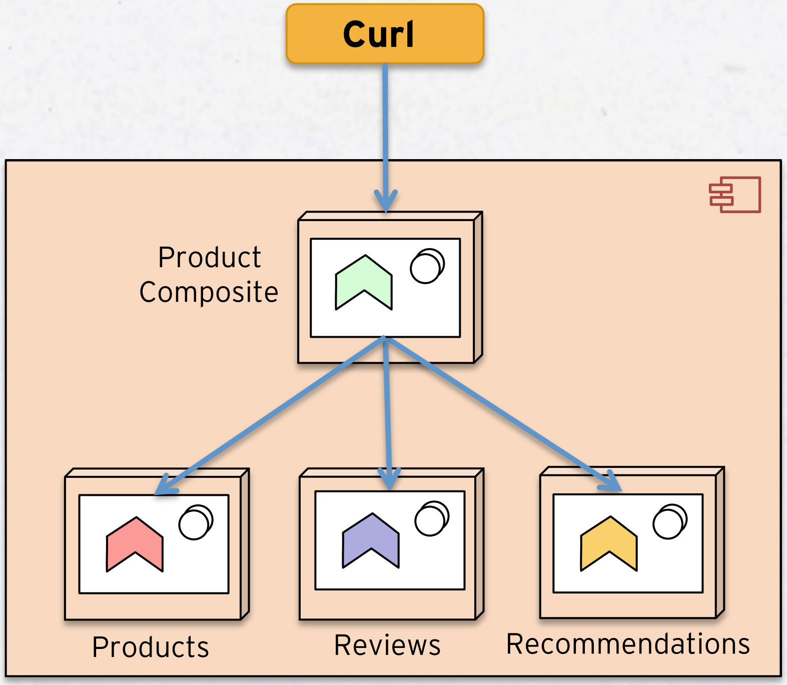
ML-WIN-DOCKER.cloudapp.net:3389
Administrator: C:\Windows\system32\cmd.exe - docker run windowsservercore cmd

```
C:\Users\ml>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
windowsservercore   latest              6801d964fda5        12 weeks ago        0 B
windowsservercore   10.0.10586.0        6801d964fda5        12 weeks ago        0 B

C:\Users\ml>docker run windowsservercore cmd
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>_
```

DEMO, DEMO, DEMO

- Start up a test landscape
- Scale up and down
- Non-responding services

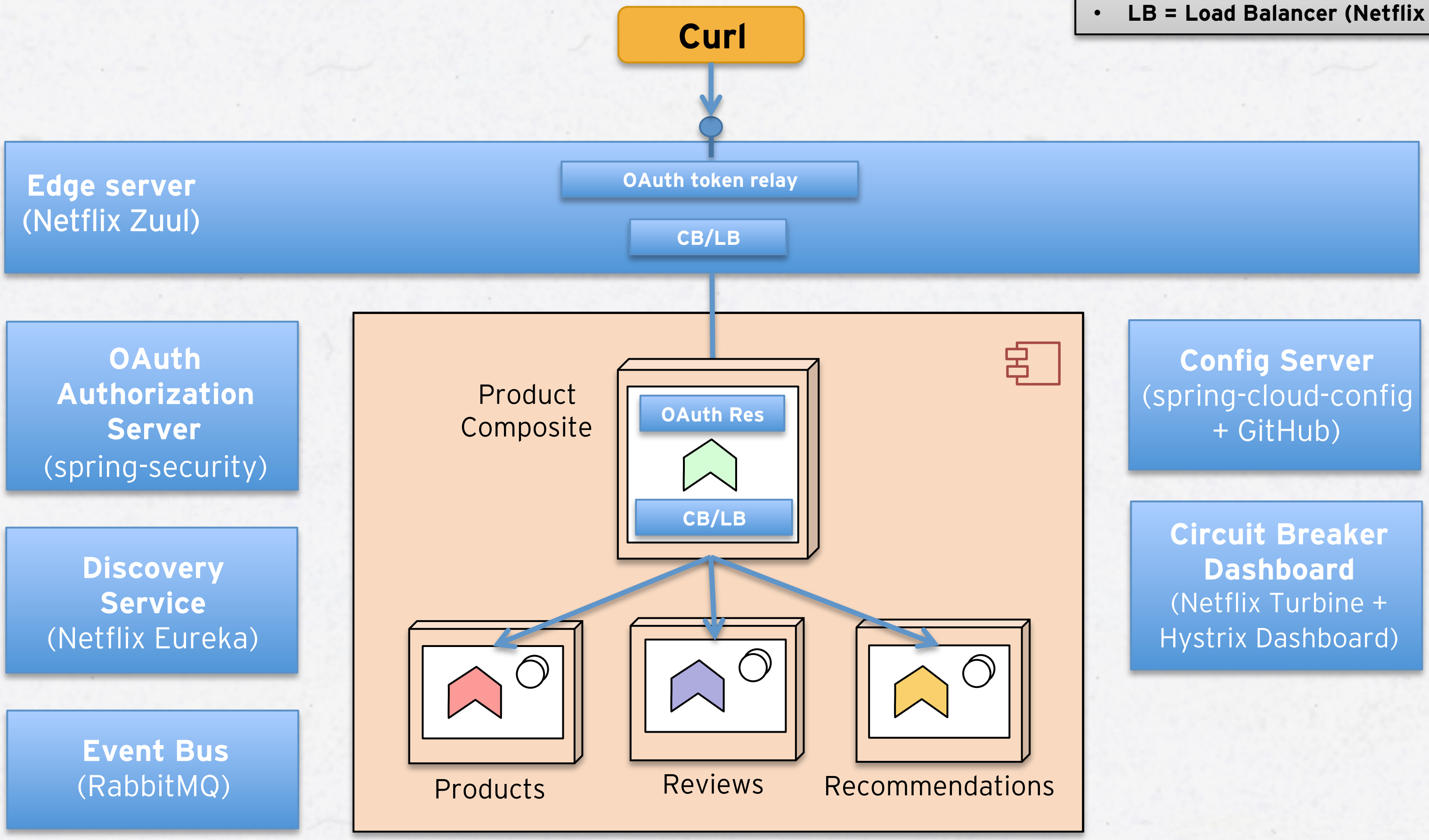
DEMO SYSTEM LANDSCAPE



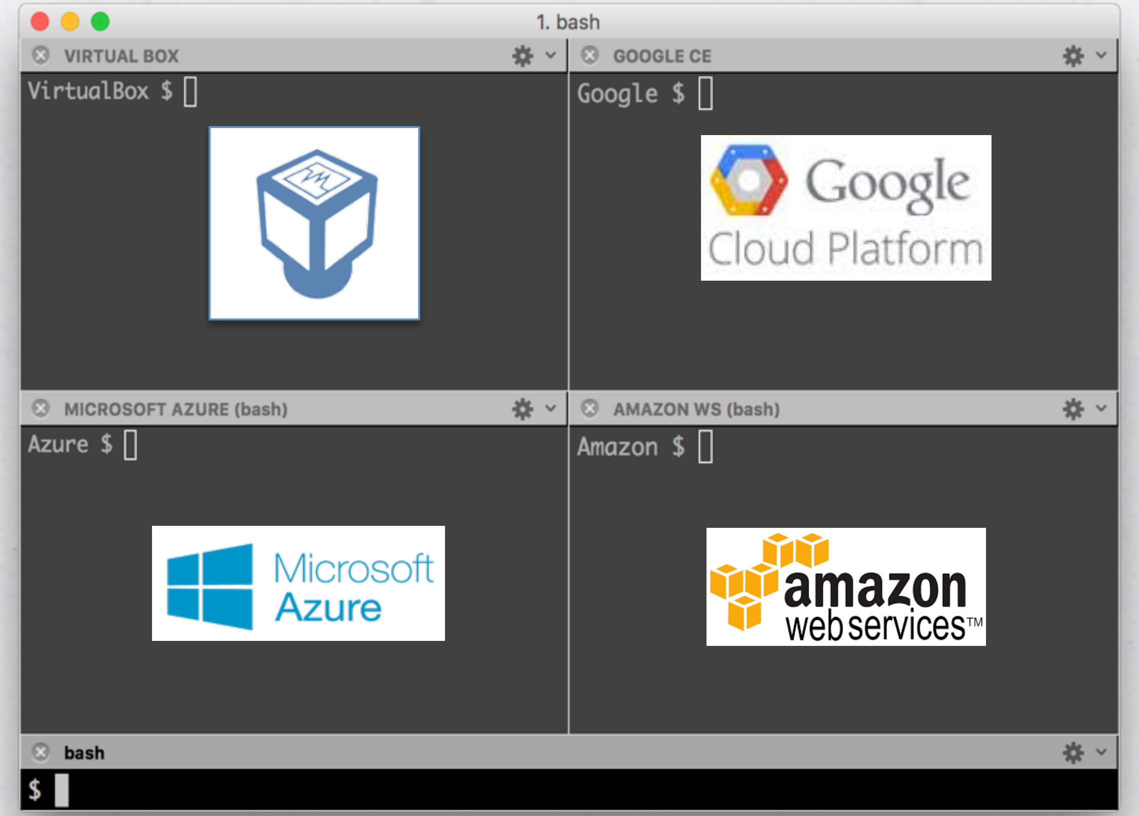
DEMO SYSTEM LANDSCAPE

Legend

- CB = Circuit Breaker (Netflix Hystrix)
- LB = Load Balancer (Netflix Ribbon)



DEMO, DEMO, DEMO



```
VirtualBox $ 
```

```
Google $ 
```

```
Azure $ 
```

```
Amazon $ 
```

```
$ 
```

DEMO, DEMO, DEMO

- test-all.sh

```
$ docker-compose up -d

Wait for: $host:8761...
Wait for: $host:8761/eureka/apps/product-service...
...

Get an OAuth Access Token...
$ curl -ks https://$host:9999/uaa/oauth/token ...

Call API with Access Token...
$ curl -ks https://$host/api/product/123 \
       -H "Authorization: Bearer $TOKEN" | jq .
```

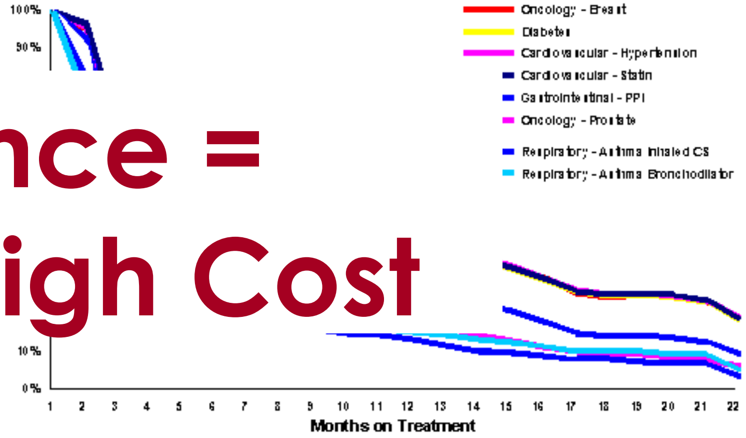
AstraZeneca

Intelligent Pharma

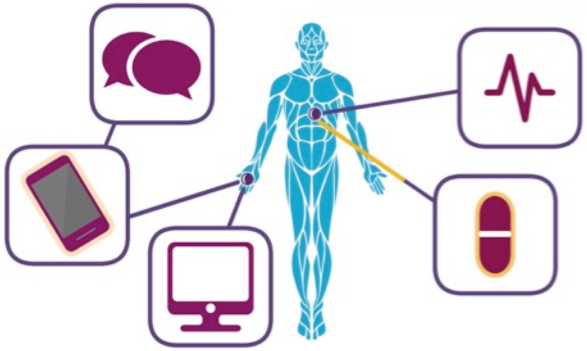
WHO Report, 2003

“Increa
adhere
far grea
popula
specific medical treatments”

Low Adherence = Poor Health & High Cost



Intelligent Pharmaceuticals

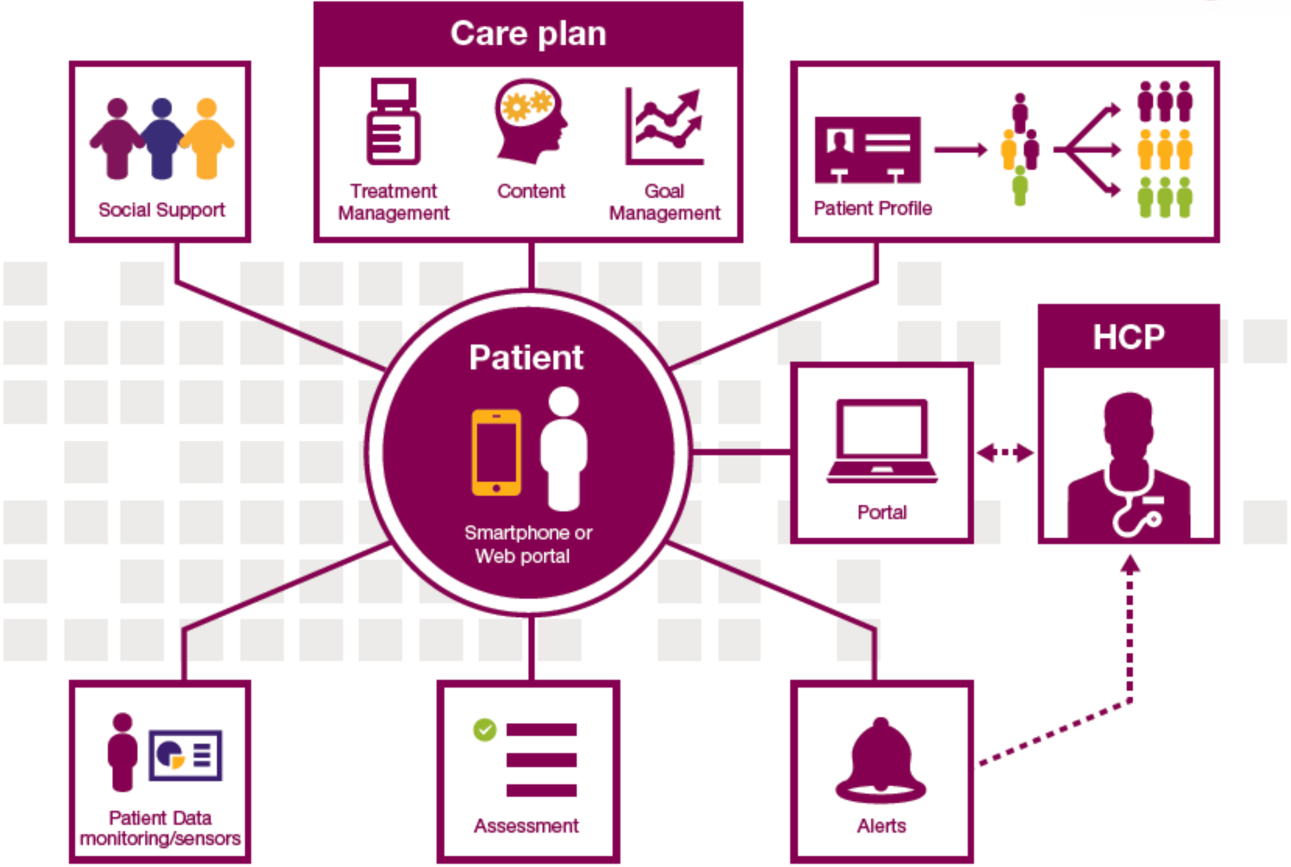


- **What:** Integrate medicines with intelligent devices, behavioral science and IT to deliver the knowledge that will help patients and physicians manage disease and achieve better health outcomes.

- **How:** Mobile phone or web based solution that deliver individualized support, education and helps you manage your disease

- **Task:** Generate clinical evidence to evaluate the application of these type of tools and recommend a future way for the company. Understand long term sustainability and scale factors to deliver these tools for the future.

The Intelligent Pharma Solution



Care plan

Treatment Management

Content

Goal Management

Social Support

Patient Profile

Patient

Smartphone or Web portal

HCP

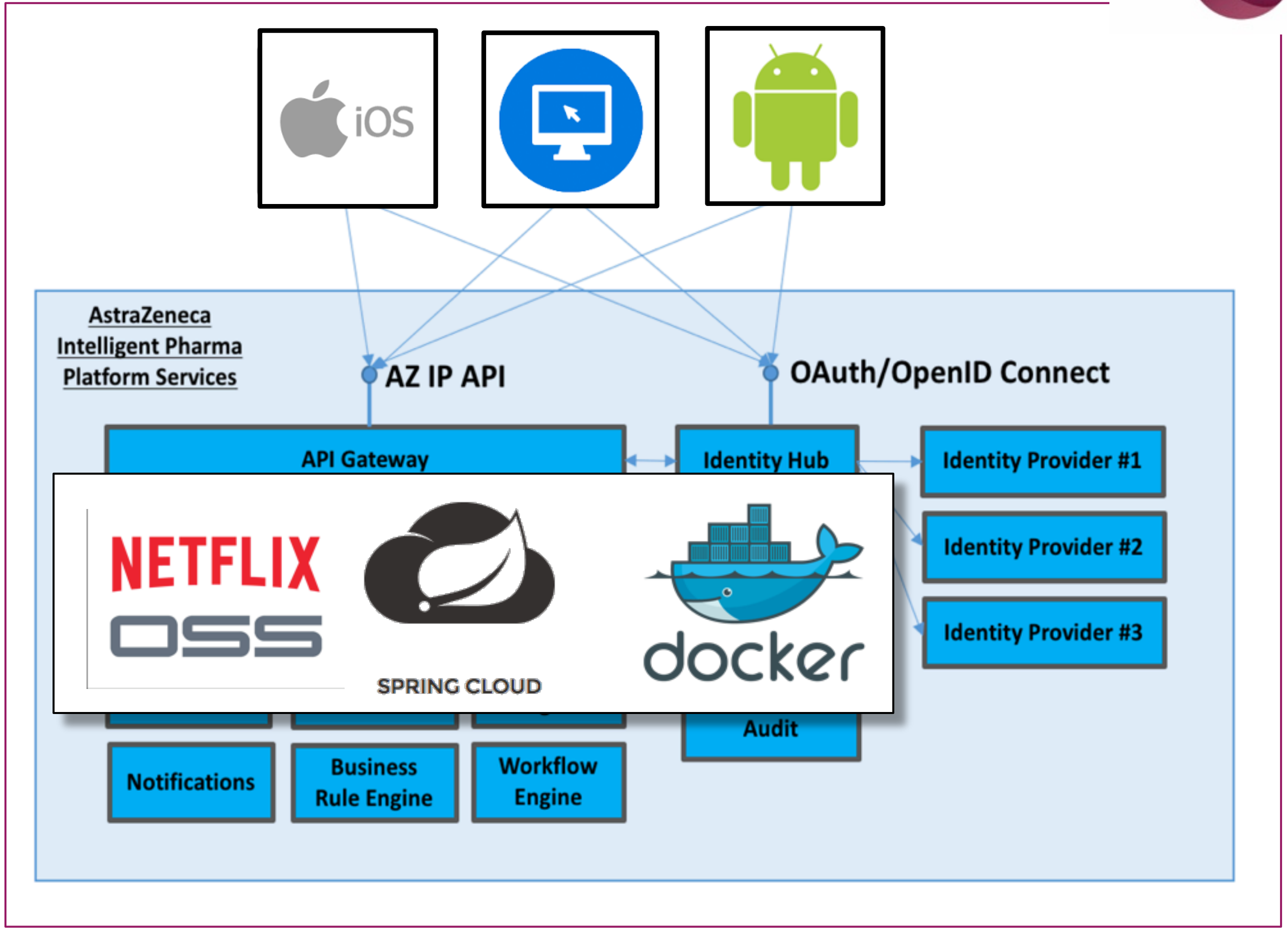
Portal

Patient Data monitoring/sensors

Assessment

Alerts

High Level Architecture



SUMMARY

- Microservices promises faster development and easier to scale
- Challenges with distributed systems applies to microservices
- Docker containers makes it easier to run many microservices
- Spring-Cloud & Netflix OSS makes it easier to manage them
- Stay tuned for clustering capabilities…
- Interested?
  - See http://callistaenterprise.se/blogg/teknik/2015/05/20/blog-series-building-microservices/

?