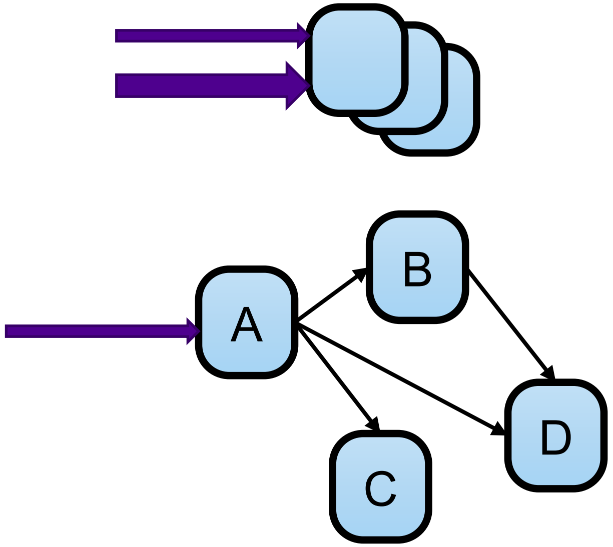
Tampere University

Intro to cloudnative
Kari Systä, 12.10.2021

What are typical cloud applications

- Networks of containers!



Logically like:

```
A() {
    B();
    C();
    D();
}
```

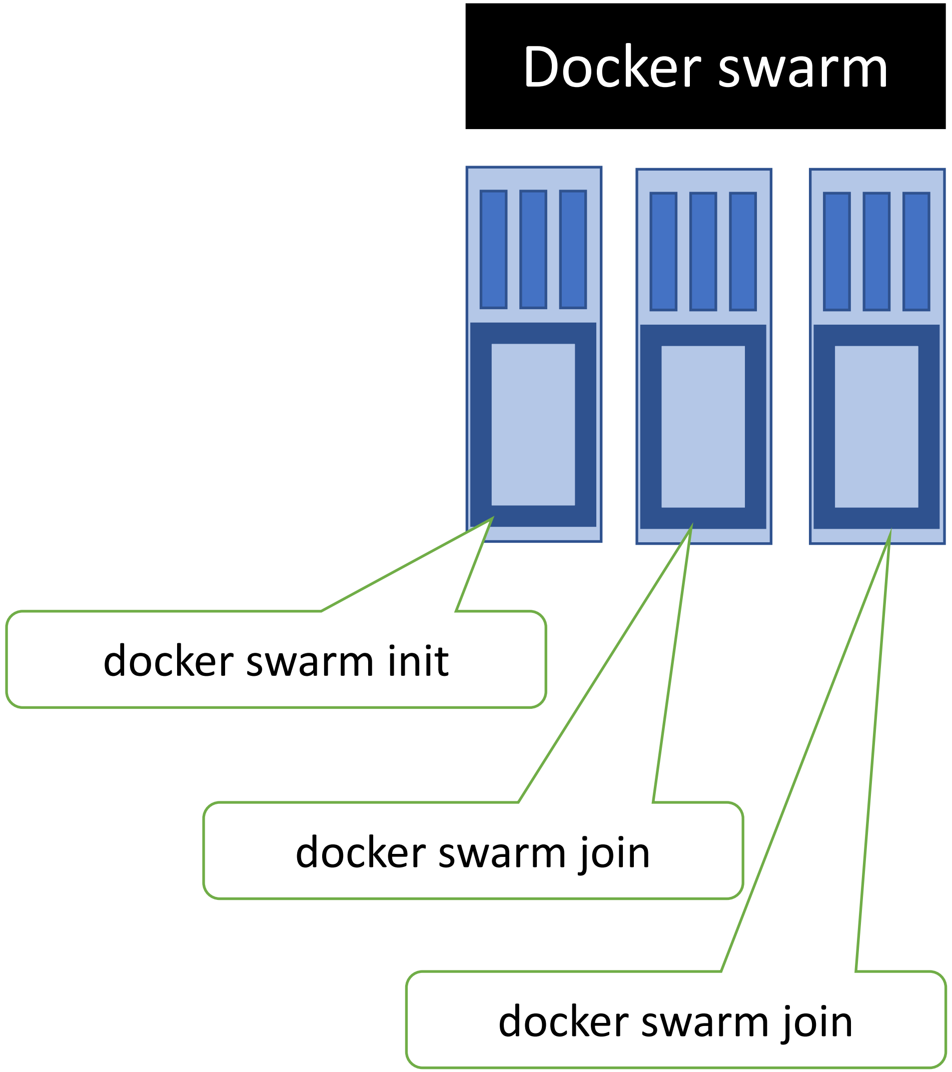
But implemented as inter-process communication.

```
A() {
    http.get(B:80);
    http.get(C:80);
    http.get(D:80);
}
```

Often considered as corner-stone of cloud-native

Docker Swarm

- Clustering for scalability
- A swarm is a group of host running docker in swarm mode
- A host can be either a *manager* or *worker*
- Workers run *services*
- Manager assigns tasks to worker nodes
  - *Load balancing*



Docker swarm

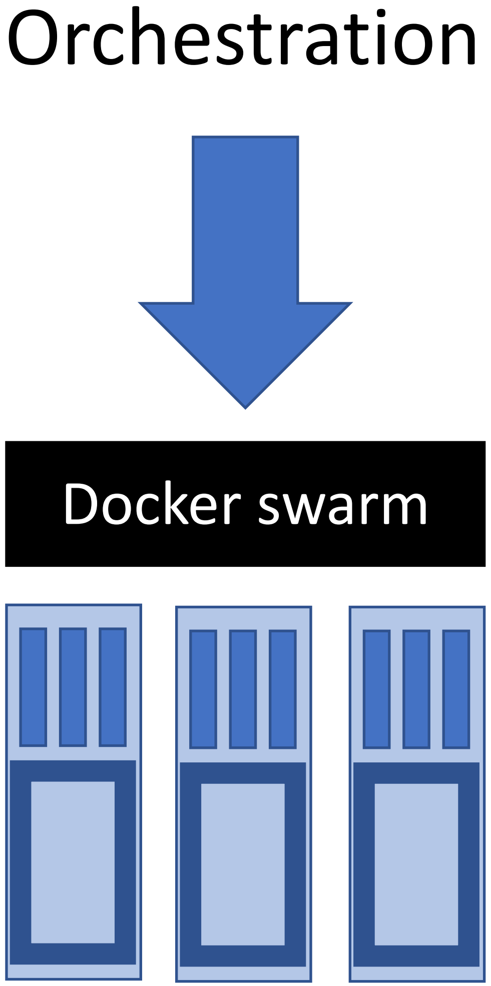
docker swarm init

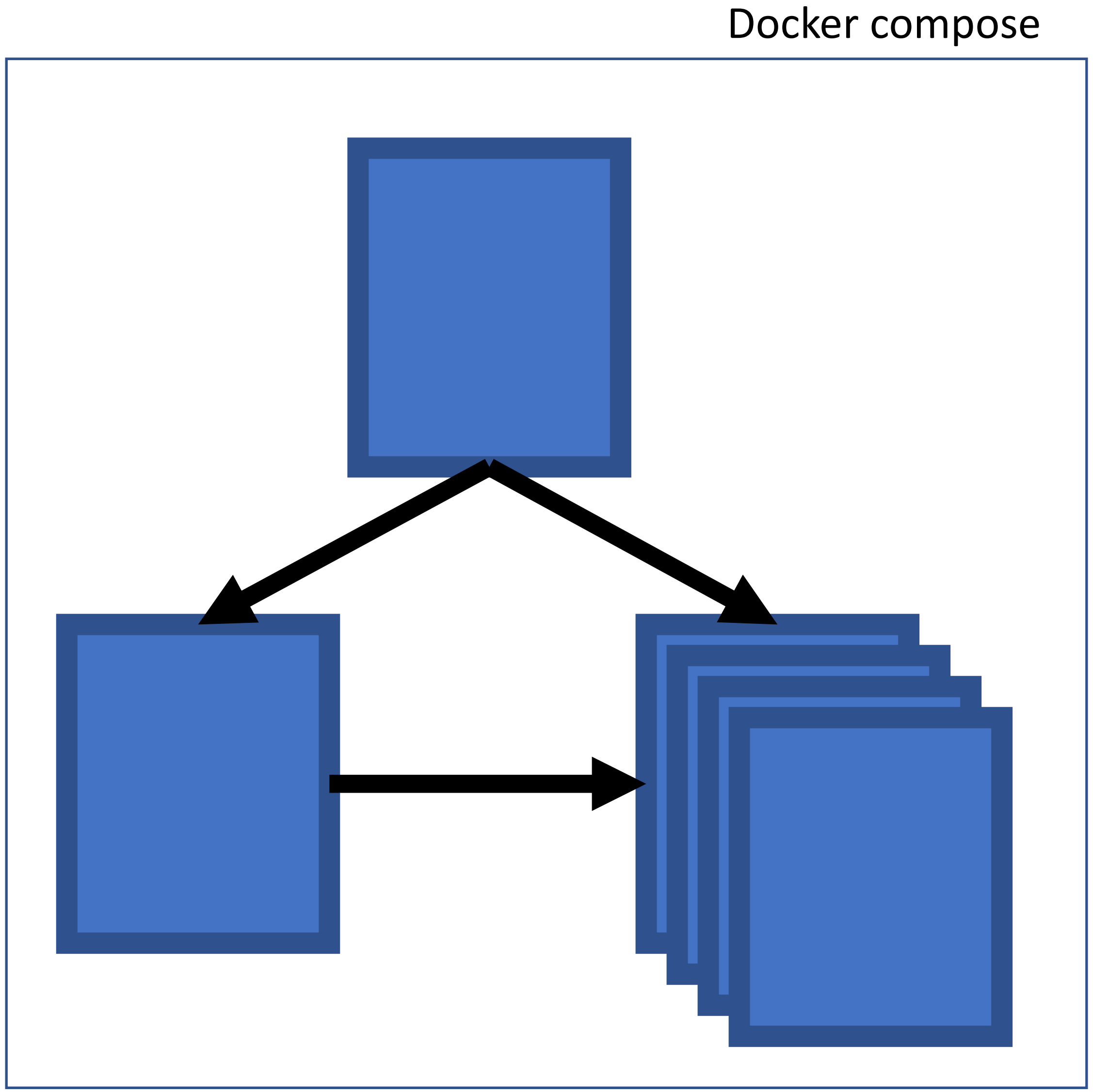
docker swarm join

docker swarm join

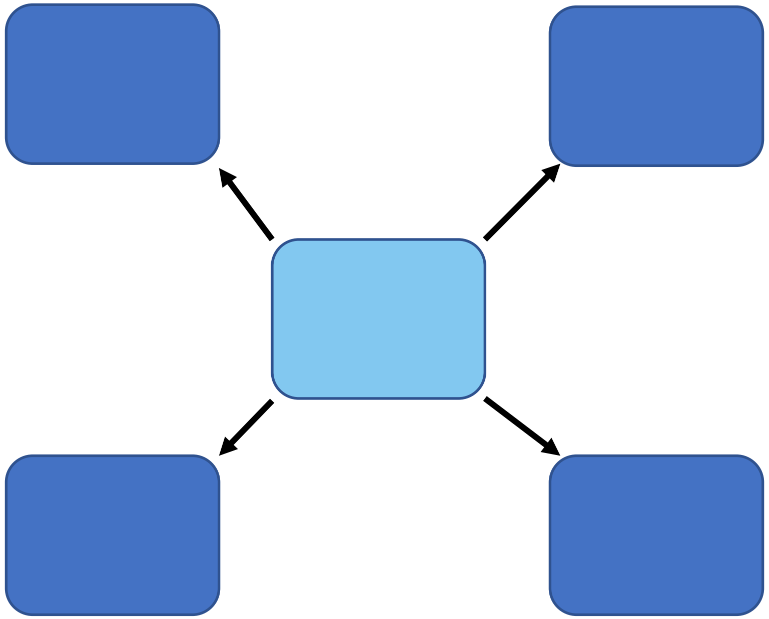
Scalability is important, too

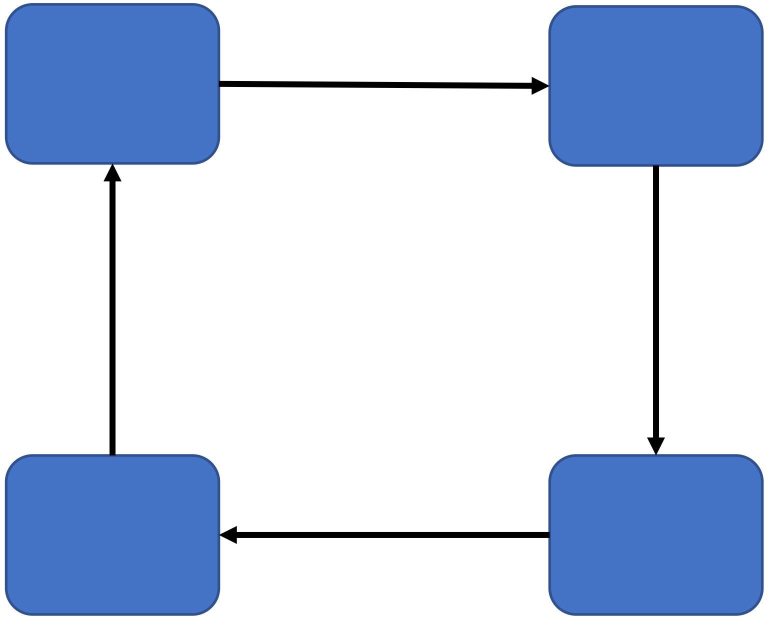
Docker swarm - docker compose both support cloud-native



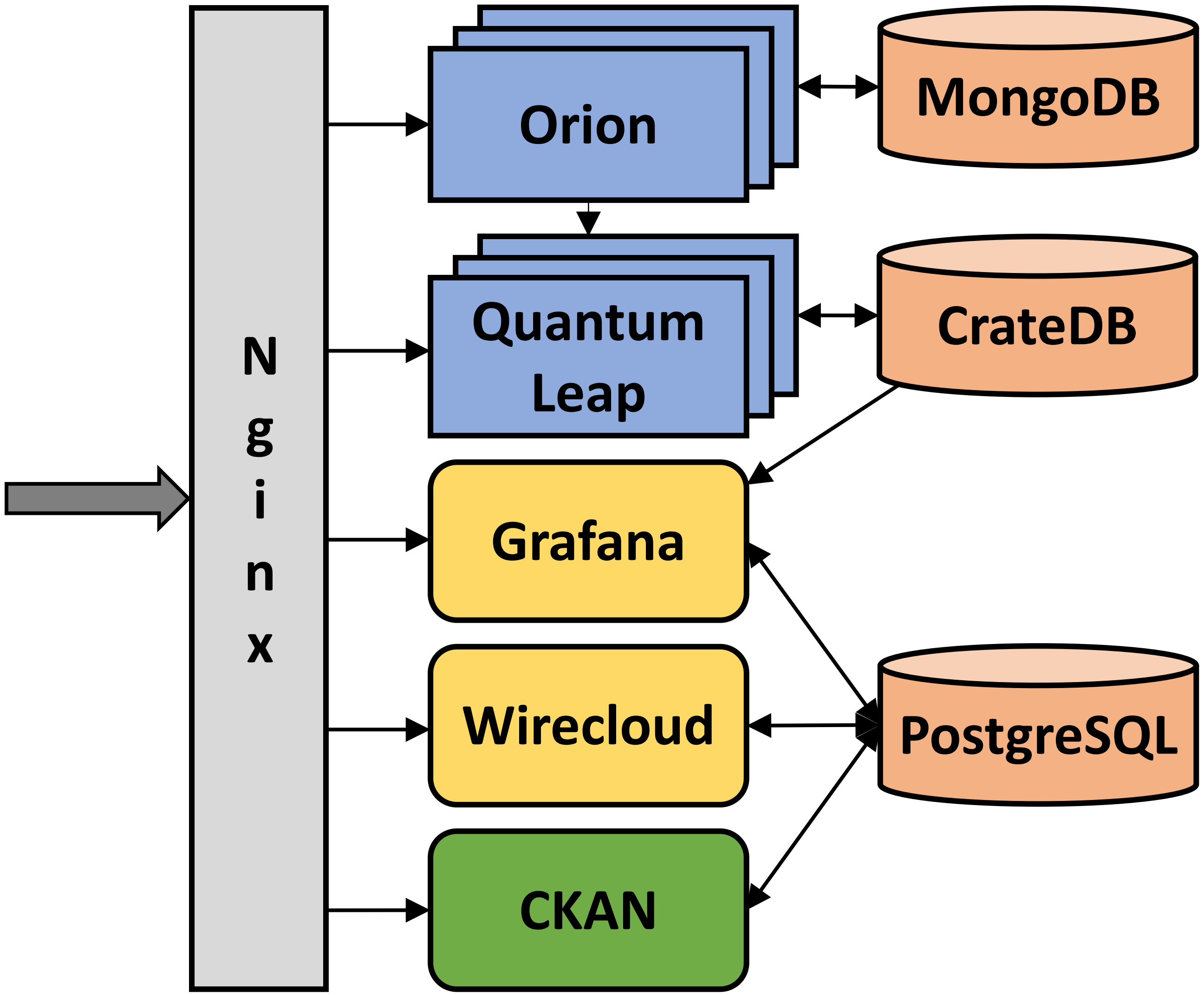


Cloud-native should support Orchestration vs Choreography





FIWARE platform architecture



FIWARE Core Component 🟦

Dashboard Component 🟨

Data Management Component 🟩

Database ▭

Access control, proxy server ⬜

FIWARE access control components (Keyrock, Wilma and AuthZForce) are not included in this document.

Cloud-native applications and architectures

Some definitions

- If an app is "cloud-native," it’s specifically designed to provide a consistent development and automated management experience across private, public, and hybrid clouds.
- A native cloud application (NCA) is a program that is designed specifically for a cloud computing architecture.
  - NCAs are designed to take advantage of cloud computing frameworks,
  - which are **composed of loosely-coupled cloud services**.
  - That means that developers must break down tasks into separate **services that can run on several servers in different locations**.
  - Because the infrastructure that supports a native cloud app does not run locally, NCAs must be **planned with redundancy** in mind so the application can withstand equipment failure and be able to re-map IP addresses automatically should hardware fail.

Some links

- 10 Key Attributes of Cloud-native Applications, <https://thenewstack.io/10-key-attributes-of-cloud-native-applications/>
- What are cloud-native applications? <https://opensource.com/article/18/7/what-are-cloud-native-apps>
- Native cloud application (NCA), <https://searchitoperations.techtarget.com/definition/native-cloud-application-NCA>
- Understanding cloud-native applications, <https://www.redhat.com/en/topics/cloud-native-apps>
- David S. Linthicum, Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between, IEEE Cloud Computing, December 2017.

Tampereen yliopisto
Tampere University

Some links

- 10 Key Attributes of Cloud-native Applications, <https://thenewstack.io/10-key-a ibutes-of-cloud-native-applications/>
- What ar applications?

pps>

ud-

n: The
ecember

1. **Packaged as lightweight containers**
2. **Developed with best-of-breed languages and frameworks**
3. **Designed as loosely coupled microservices**
4. **Centered around APIs for interaction and collaboration**
5. **Architected with a clean separation of stateless and stateful services**
6. **Isolated from server and operating system dependencies**
7. **Deployed on self-service, elastic, cloud infrastructure**
8. **Managed through agile DevOps processes**
9. **Automated capabilities**
10. **Defined, policy-driven resource allocation**

David S. Linthicum, Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between, IEEE Cloud Computing, December 2017

- **Performance**. You'll typically provide better performance than is possible with non-native features. For example, you can deal with an input/output (I/O) system that works with autoscaling and load balancing features.
- **Efficiency**. Cloud-native applications' use of cloud-native features and application programming interfaces (APIs) should provide more efficient use of underlying resources. That translates to better performance and/or lower operating costs.
- **Cost**. Applications that are more efficient and typically cost less to run. Cloud providers send you a monthly bill based upon the amount of resources consumed, so if you can do more with less, you save on dollars spent.
- **Scalability**. Because you write the applications to the native cloud interfaces, you have direct access to the autoscaling and load-balancing features of the cloud platform.