



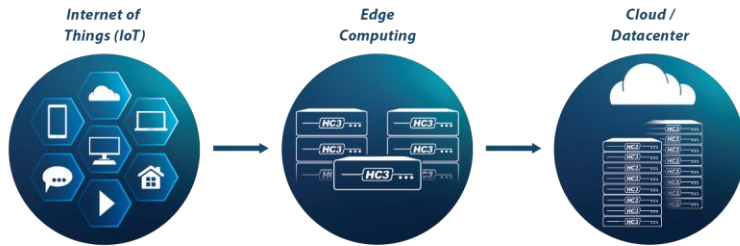
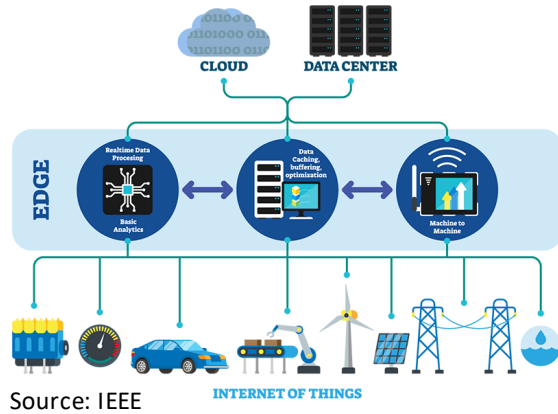
# Kubernetes on the Edge

## A Peer-to-Peer Approach

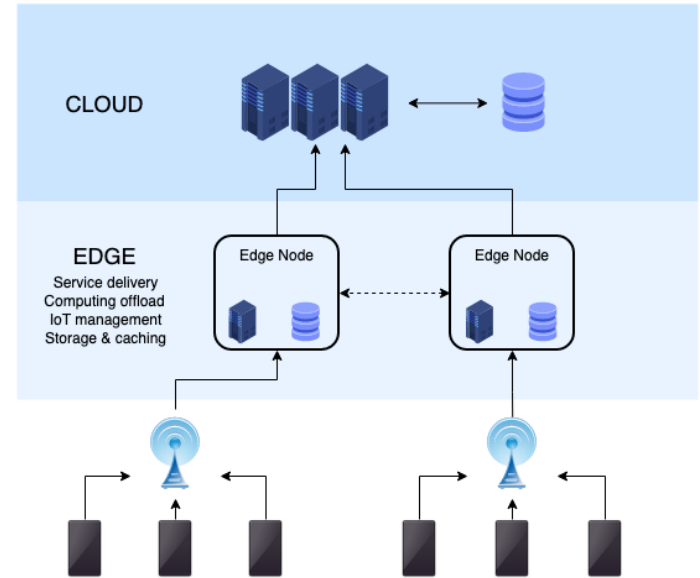
Jan S. Rellermeyer    Michalis Vrachasotakis  
Distributed Systems Group  
TU Delft

Chris Adeniyi-Jones  
Arm Research  
Cambridge, UK

# Edge Computing



Source: scalecomputing.com



Source: Wikipedia

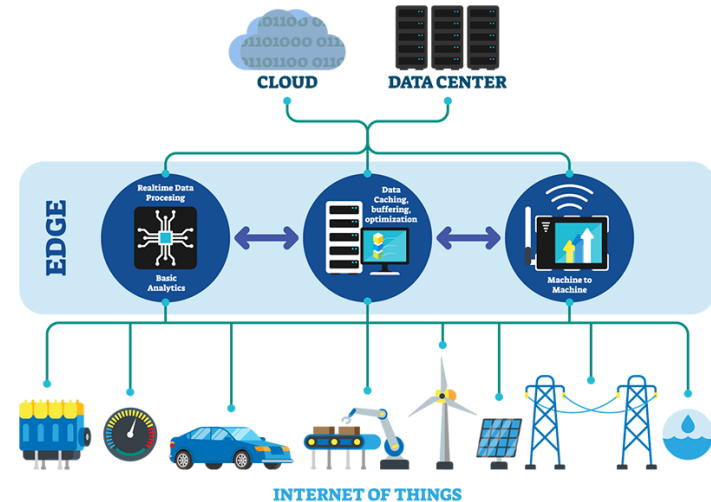
# Edge Computing

## ▶ Data Flow:

- ▶ From IoT through Edge to Cloud
- ▶ Offloading into a two tiers

## ▶ Control Flow?

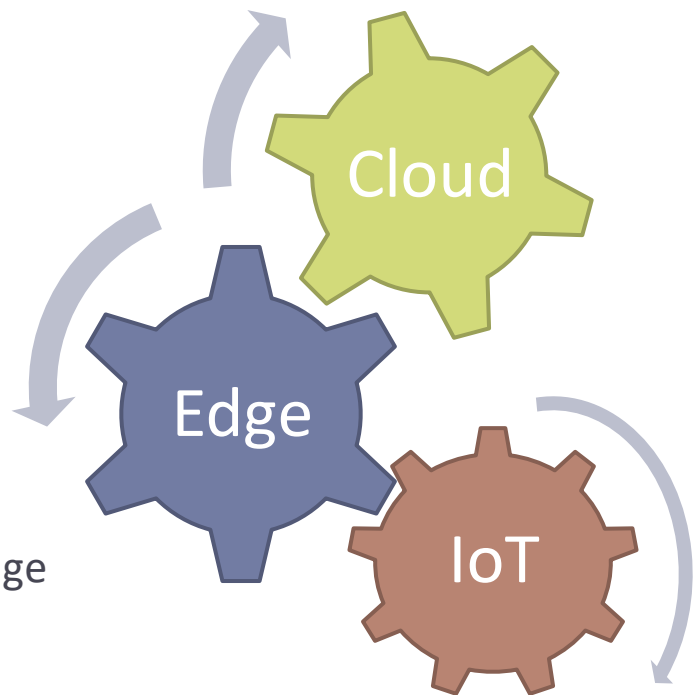
- ▶ Where is the application?
- ▶ Who is in control?



# Broader Vision

---

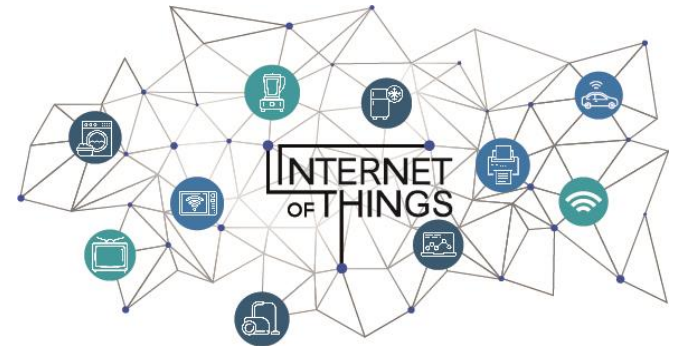
- ▶ Make Edge a seamless extension of IoT and Cloud
- ▶ Support many different application scenarios
  - ▶ Data Flow
  - ▶ Control
  - ▶ Ownership
- ▶ Extend existing cloud stack to embrace the Edge
  - ▶ Assumption: people do not want new stacks for the Edge



# Specific Challenges

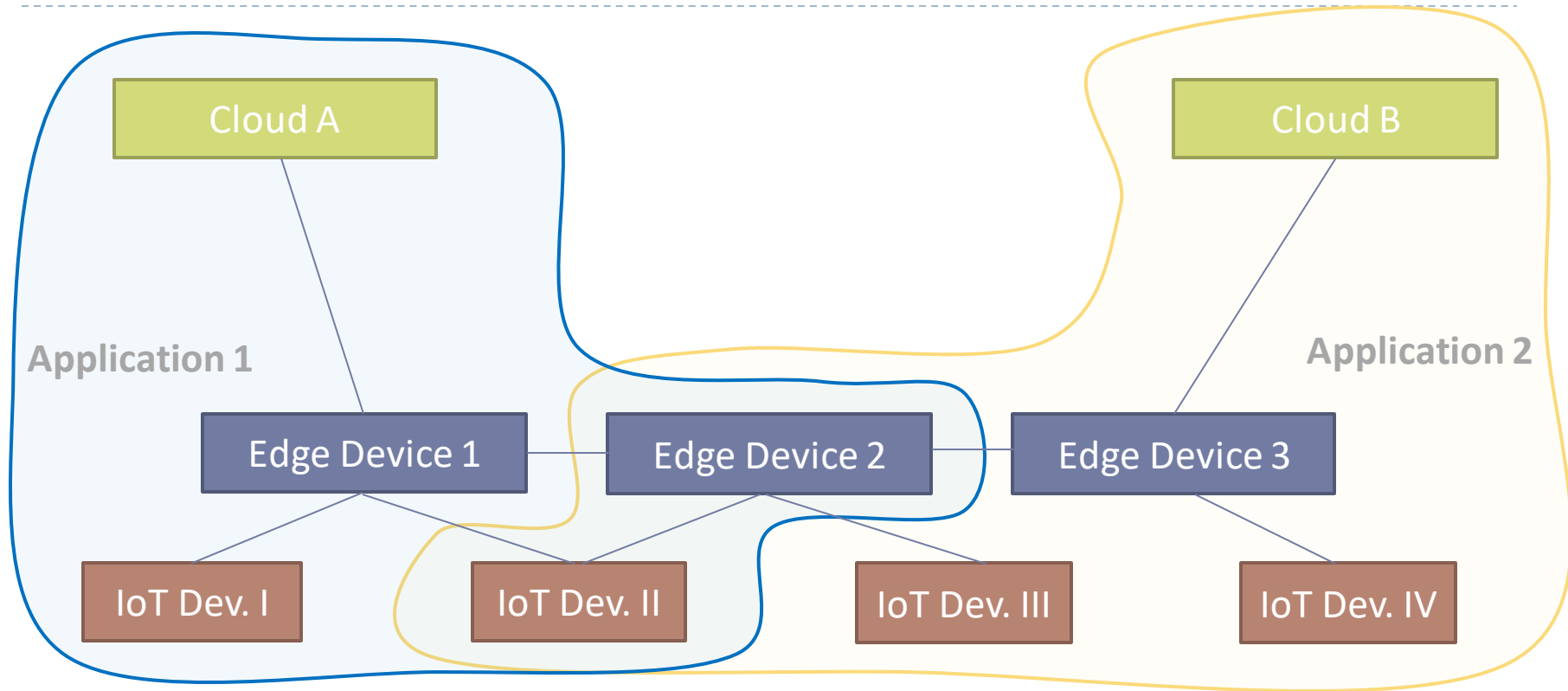
---

- ▶ Resources
  - ▶ Edge often unable to run a full-scale cloud stack
- ▶ Multi-tenancy
  - ▶ Edge might be lacking hardware support for full virtualization
  - ▶ Multiple applications overlap
- ▶ Control
  - ▶ Edge devices can be owned by different parties
- ▶ Energy efficiency
- ▶ Security
- ▶ Resilience



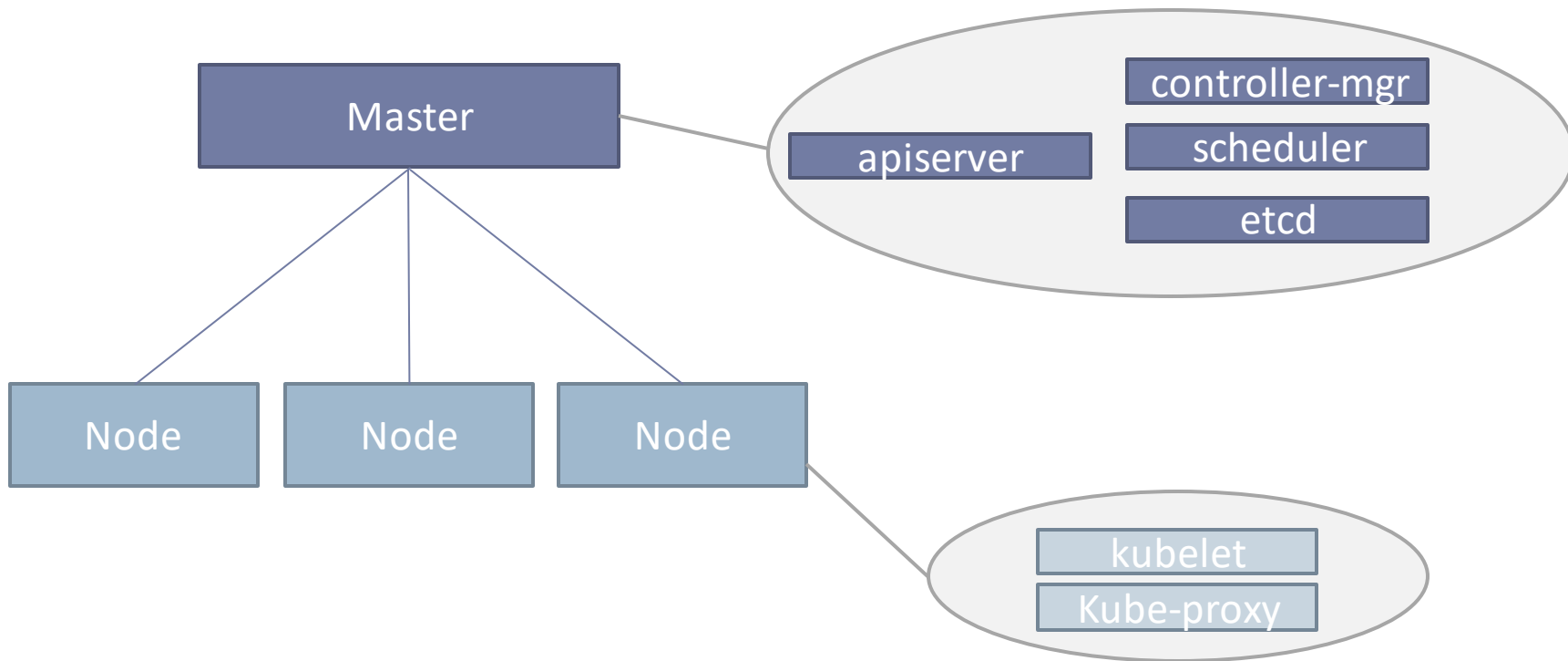
Source: OFCOM Switzerland

# Edge Computing: Federation of Resources



# Kubernetes

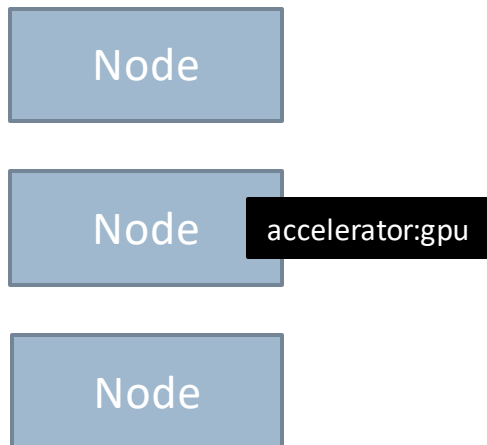
---



# Node Selection

---

- ▶ Finding nodes with matching properties

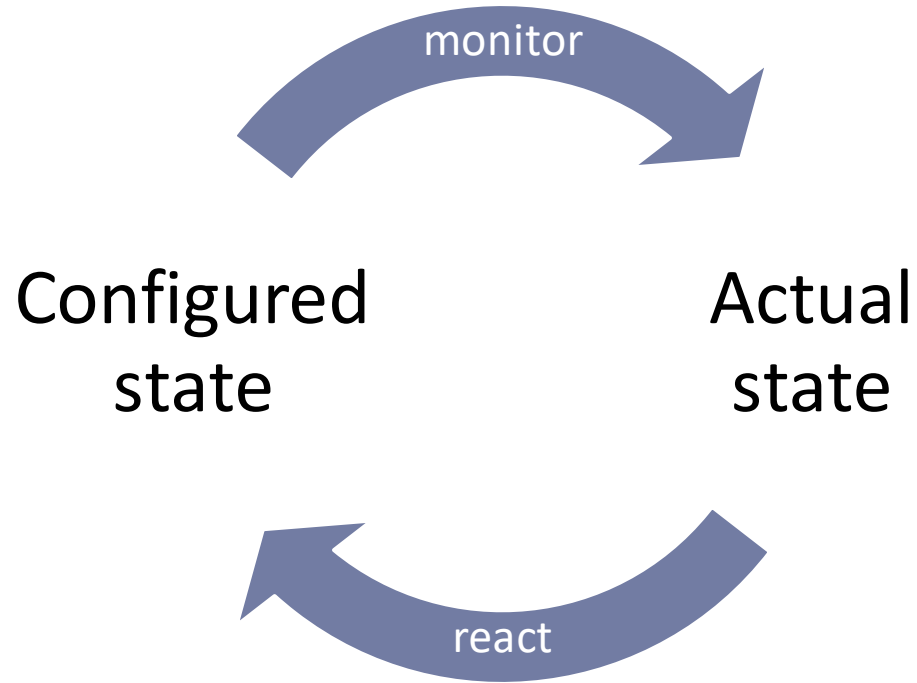


- ▶ Matching labels
  - ▶ e.g., OS, Arch, etc.
  - ▶ Custom labels
- ▶ Affinities / Anti-affinities
  - ▶ E.g., always collocate, never collocate



# Controller

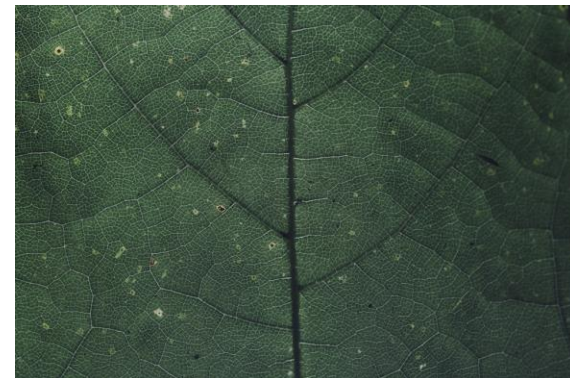
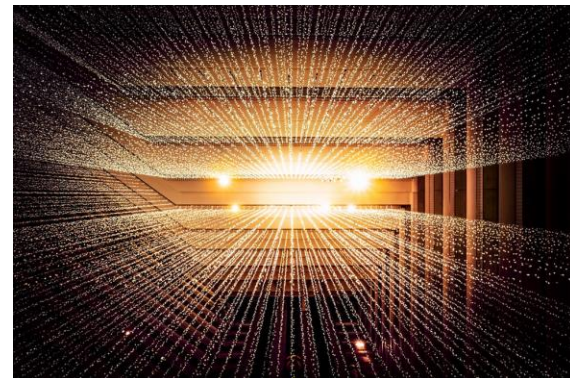
---



# Mismatches with the Edge

---

- ▶ **Centralized and Hierarchical**
  - ▶ The Edge as a whole is neither
- ▶ **Global**
  - ▶ Incompatible with the organic nature of the Edge
- ▶ **Consistency over Availability**
  - ▶ Incompatible with the QOS of edge devices and WANs

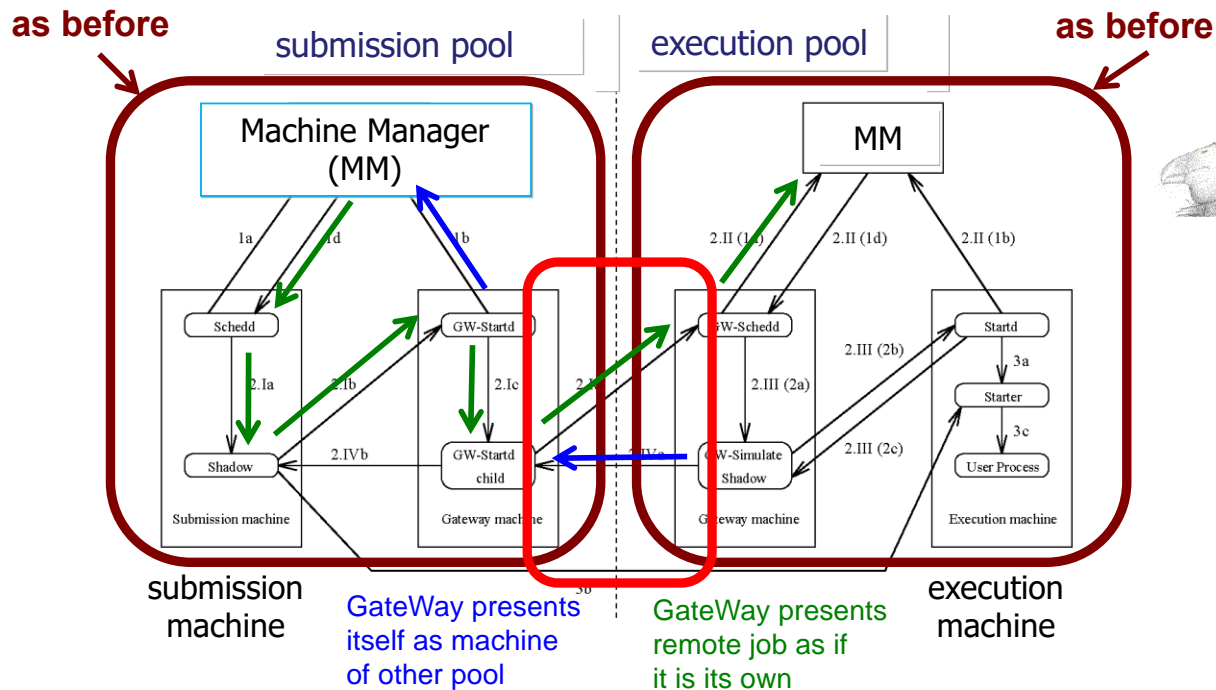


# Efforts within Kubernetes

---

- ▶ **K3s**
  - ▶ Kubernetes cluster on the Edge
- ▶ **KubeEdge**
  - ▶ Connects edge nodes to cloud with additional Kubernetes components
- ▶ **Virtual Kubelet**
  - ▶ Hybrid Cloud-Edge cluster
  
- ▶ All solutions are still centralized and controlled from the cloud

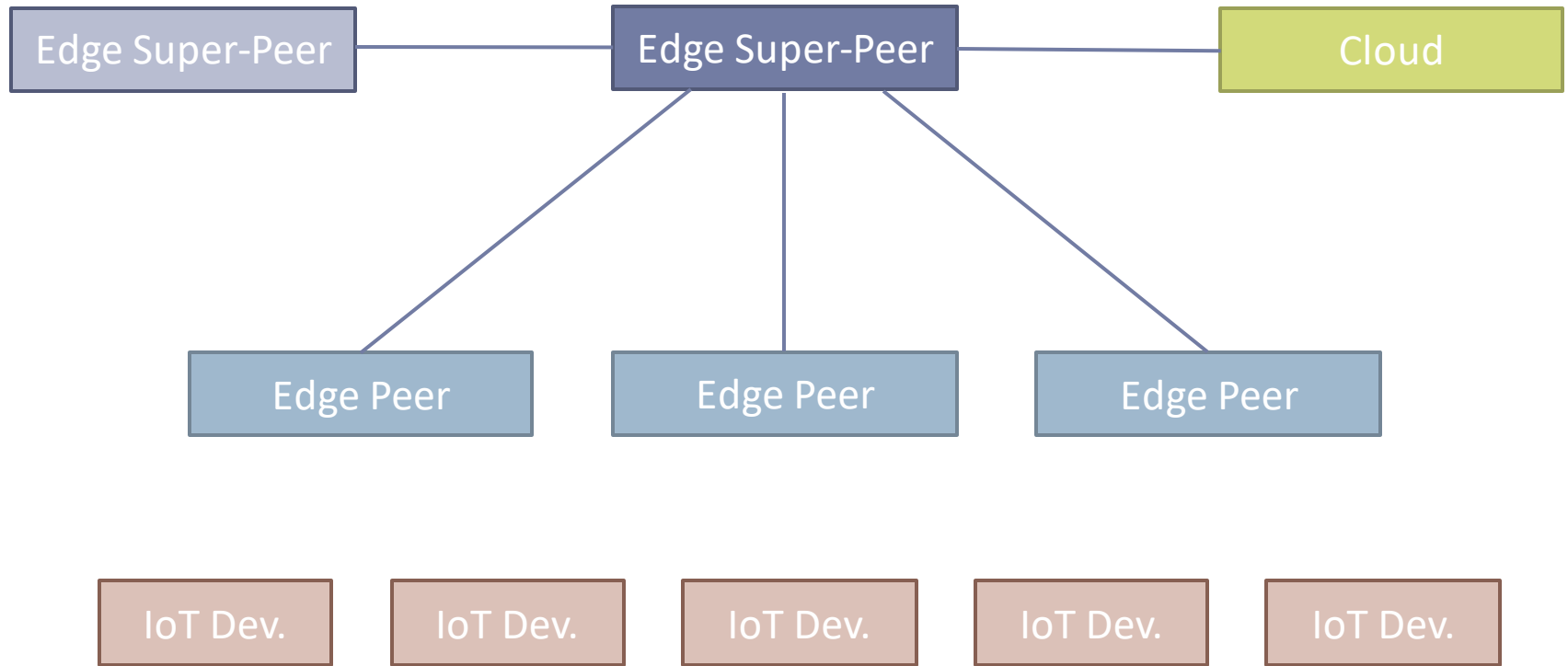
# Inspiration from Grid Computing



D.H.J. Epema, M. Livny, R. van Dantzig, X. Evers, and J. Pruyne, "A Worldwide Flock of Condors: Load Sharing among Workstation Clusters," *Future Generation Computer Systems*, Vol. 12, pp. 53-65, 1996.

# Super-Peer Architecture

---



# Kubernetes on the Edge: Project Stolos

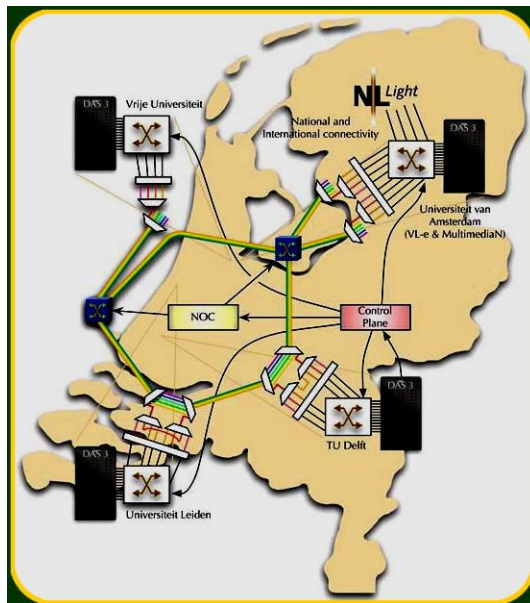
---

- ▶ P2P system and DHT instead of hierarchical control and etcd
  - ▶ Make resources discoverable
  - ▶ Allows to federate resources dynamically
  - ▶ Disseminate configuration information
- ▶ Scopes for multi-tenancy and federation
  - ▶ Limit visibility
  - ▶ Enforce access control
  - ▶ Decentralize control



# Testbed

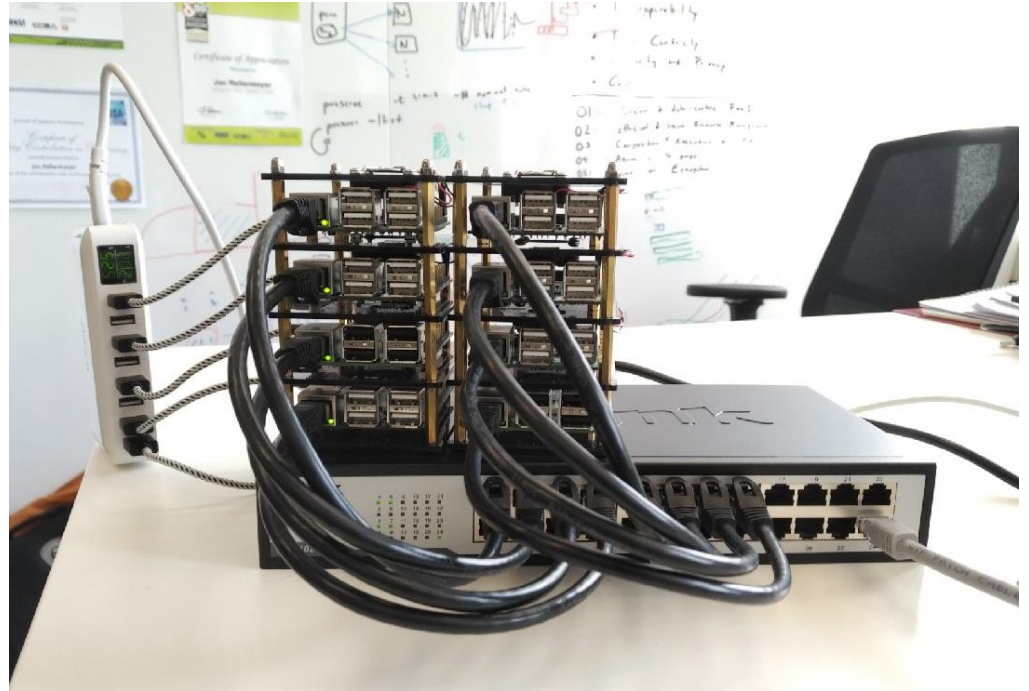
- ▶ DAS-5
- ▶ Funded by the Dutch Science Foundation
- ▶ Cluster of Clusters across the Netherlands
- ▶ 48 machines in Delft
- ▶ 100+ in Amsterdam



# Edge Testbed

---

- ▶ 100+ Raspberry Pi B+
- ▶ Goal: run on a sizable, distributed infrastructure
- ▶ Validate design choices through experimentation

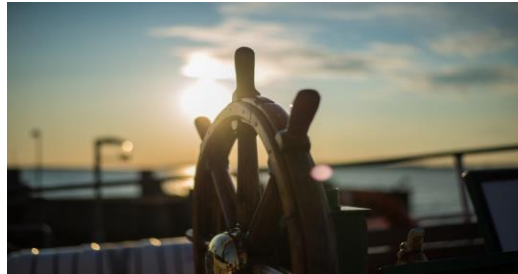




# Next Steps

---

- ▶ Understand the performance and possible limitations of the different approaches
- ▶ Implement ideas for decentralized container sharing
- ▶ Advanced security with fortified containers



# Conclusions

---

- ▶ Kubernetes is the quasi industry standard for container orchestration in the cloud
- ▶ We can seamlessly extend it to the edge
- ▶ Decentralization is key to leveraging the benefits of the edge and support advanced applications

