Oakestra

An Orchestration Framework for Edge Computing

Giovanni Bartolomeo, Simon Bäurle, Nitinder Mohan, Jörg Ott
giovanni.bartolomeo@tum.de simon.baeurle@tum.de mohan@in.tum.de ott@in.tum.de

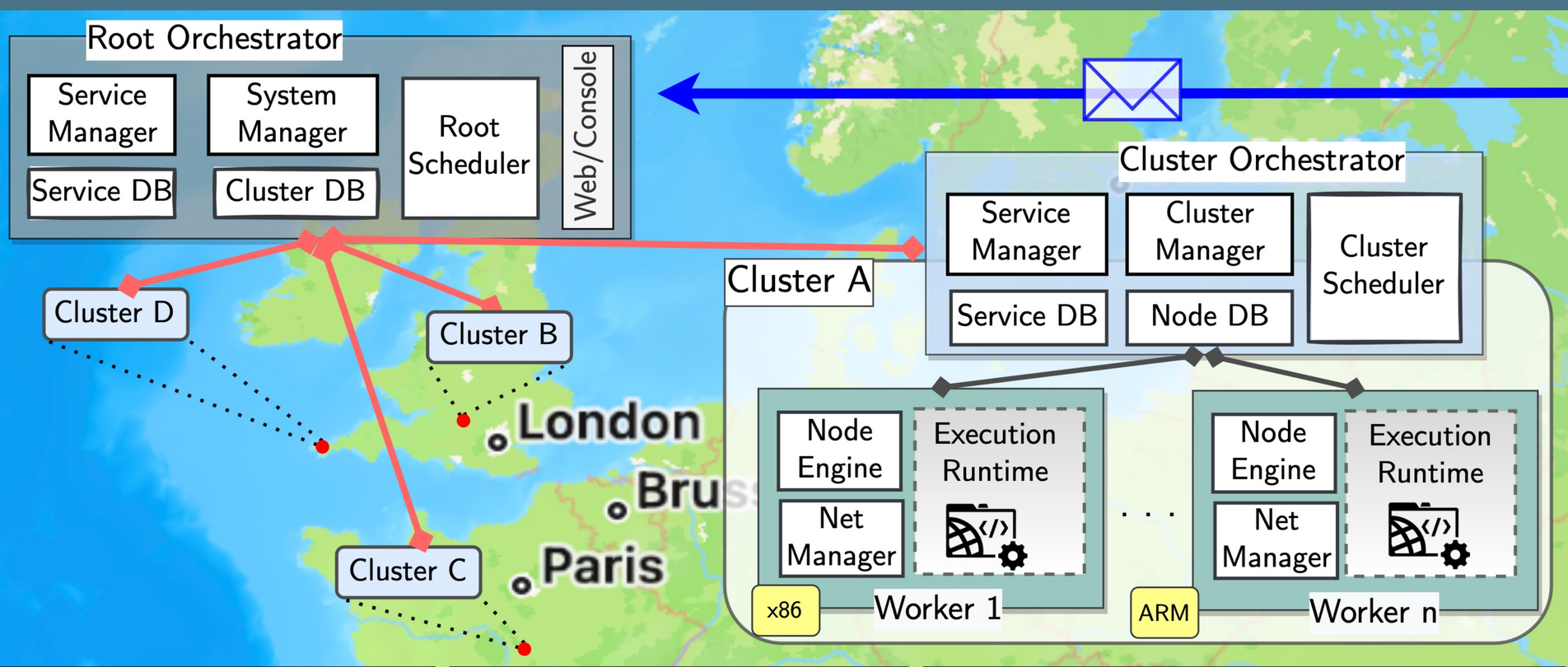
Limitations of State-of-the-Art Orchestration Frameworks on Edge

- Heterogeneity in hardware is still a problem for most frameworks [1,2]
- No support for diverse and dynamic networking conditions
- Service scheduling in this environment is a well-known NP-hard problem [3]
- No support for multiple virtualization technologies, e.g., Unikernels
- Different edge providers cannot collaborate in a federated environment



MicroK8s K3S ioFog

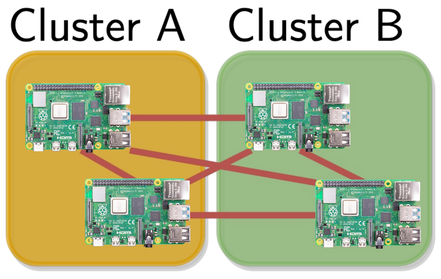
Oakestra: Orchestration at the Edge



Deployment Descriptor

```
constraints: [{
    microservice_id: 8449234M
    properties: [{
        memory: 100,
        vcpus: 2,
        vgpus: 1,
       latency: 50,
        area: 2km,
        location: munich,
        threshold: 0.25,
        rigidness: 0.5,
        convergence_time: 1000,
…}]
```

Semantic Addressing



- Dynamic Load Balancing
- Flexible algorithms designed for the Edge
  e.g. Closest Instance, Highest Capacity

POST
`http://app.login.closest`

Supported Features

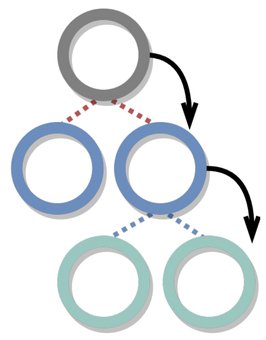
- Heterogeneous and Constrained Hardware
- Scalable Orchestration
- Federated Multi Operator Edge Infrastructure
- Multiple Virtualization Technology Support
- Multiple Cloud Operators

Modular Design

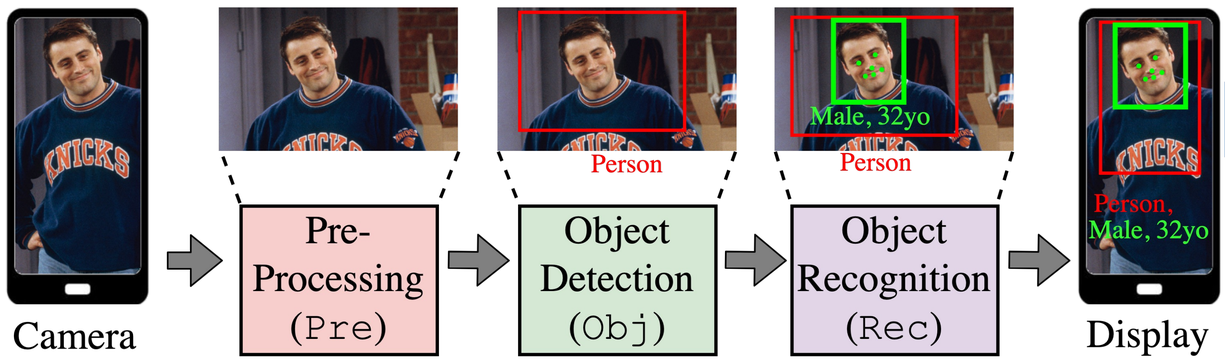
- Standardized APIs for CLI and Dashboard integration
- Extensible Scheduling Components
- Networking and Execution Runtimes as a Plug-In
- Scalable Components
- Extensible SLA design

Delegated Service Scheduling

- Clusters represented as aggregated resources
- The root scheduler finds a suitable cluster
- The cluster scheduler finds a worker node
- Multiple placement strategies available



Demo Setup



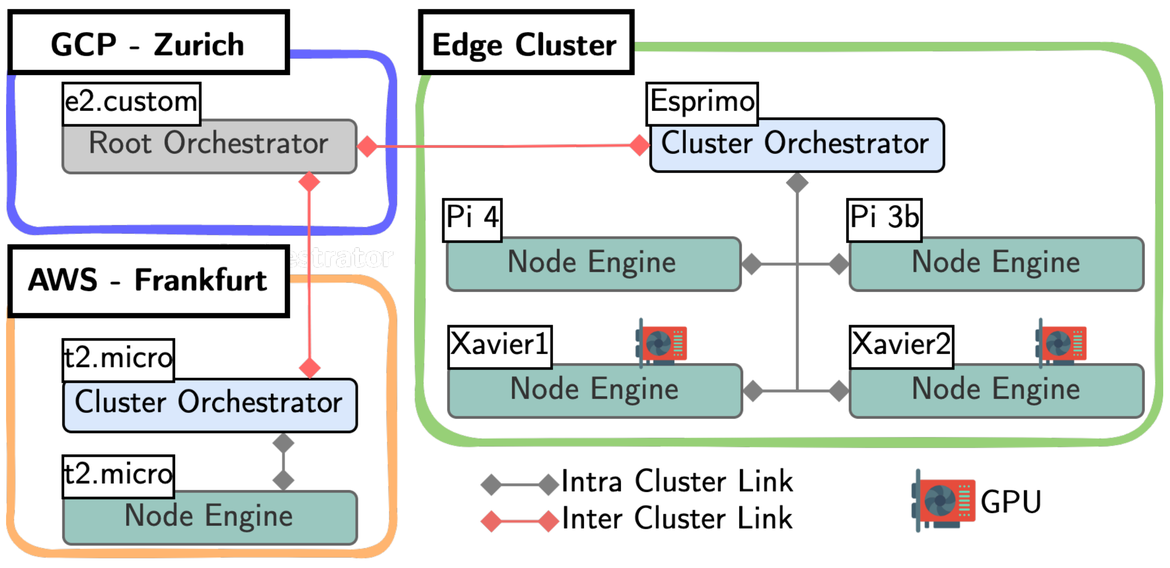
Captures and sends the camera frames to the pipeline entry point.

It reduces the frame sizes by adapting them to the Obj input format.

Detects and identifies the bounding boxes in the pictures.

If available, it detects subject details and facial landmarks.

It shows the frames with the pipeline-generated details.



Orchestration of the services across a Cloud & Edge infrastructure

Future Work

- Federated cluster authentication, and authorization
- Stateful application support
- Distributed code and image caching
- Support for persistent volumes
- Sensors and Drivers mapping

GitHub

References

[1] Sebastian Böhm and Guido Wirtz. 2021. Profiling Lightweight Container Plat- forms: MicroK8s and K3s in Comparison to Kubernetes. In ZEUS.
[2] Andrew Jeffery, Heidi Howard, and Richard Mortier. 2021. Rearchitecting Kuber- netes for the Edge. 4th ACM EdgeSys (2021).
[3] Hongyan Cui, Yang Li, Xiaofei Liu, Nirwan Ansari, and Yunjie Liu. 2017. Cloud service reliability modelling and optimal task scheduling. Iet Communications (2017).