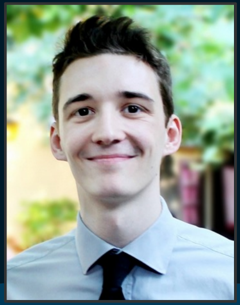
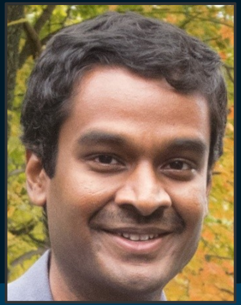
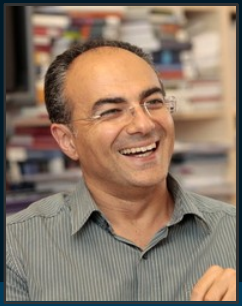
Towards a Serverless Platform for Edge AI

Thomas Rausch @thrauat
Waldemar Hummer
Vinod Muthusamy
Alexander Rashed
Schahram Dustdar

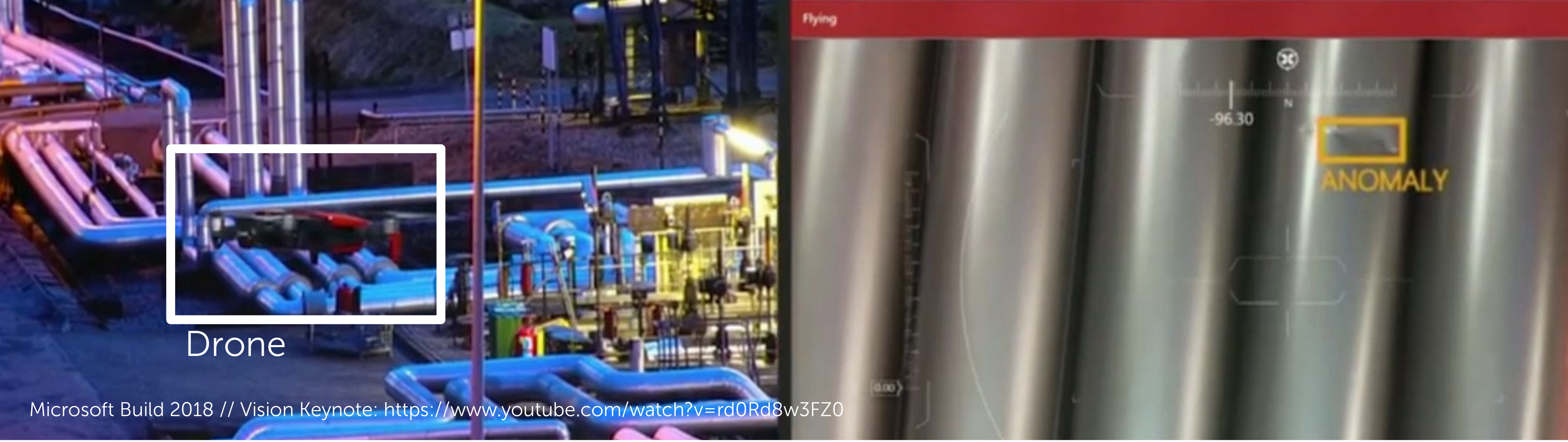


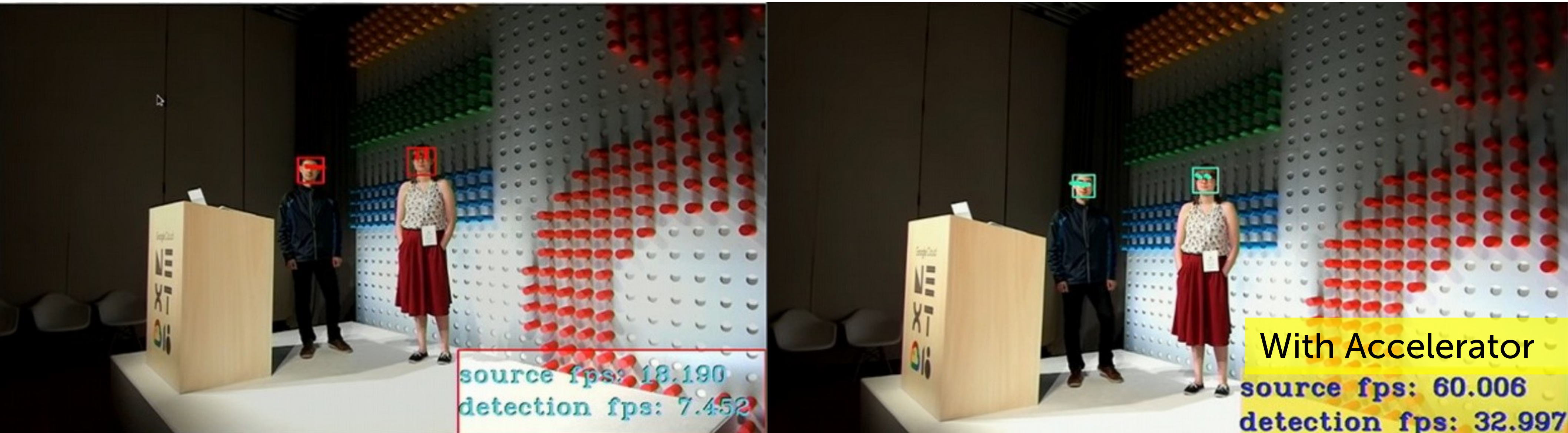




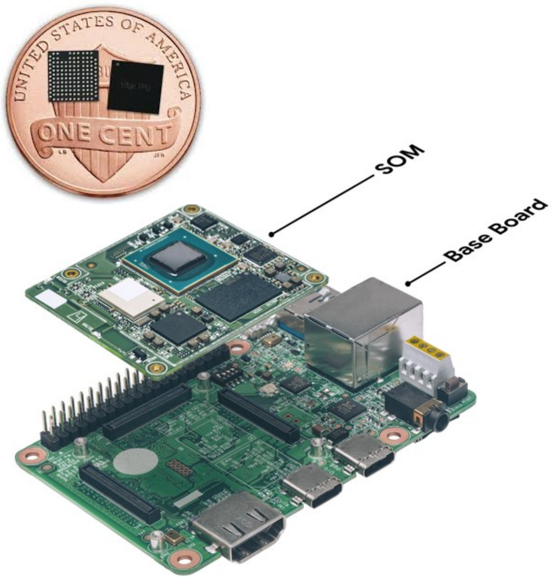
HotEdge'19, Renton, WA

TU Wien, Vienna Austria
Distributed Systems Group
https://dsg.tuwien.ac.at

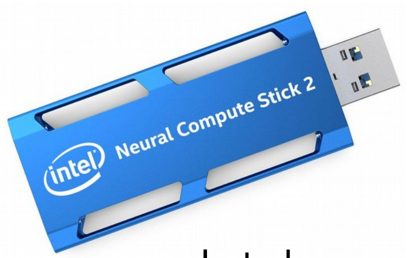

Drone
Flying
-96.30
ANOMALY
Microsoft Build 2018 // Vision Keynote: https://www.youtube.com/watch?v=rd0Rd8w3FZ0


source fps: 18.190
detection fps: 7.452
With Accelerator
source fps: 60.006
detection fps: 32.997

Edge AI Accelerators



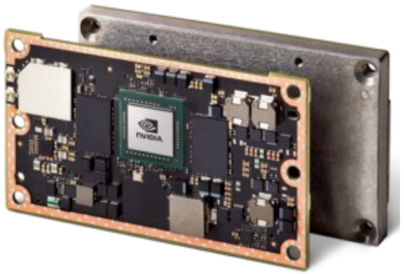
Google Edge TPU



Intel
Neural Compute Stick

Baidu Kunlun

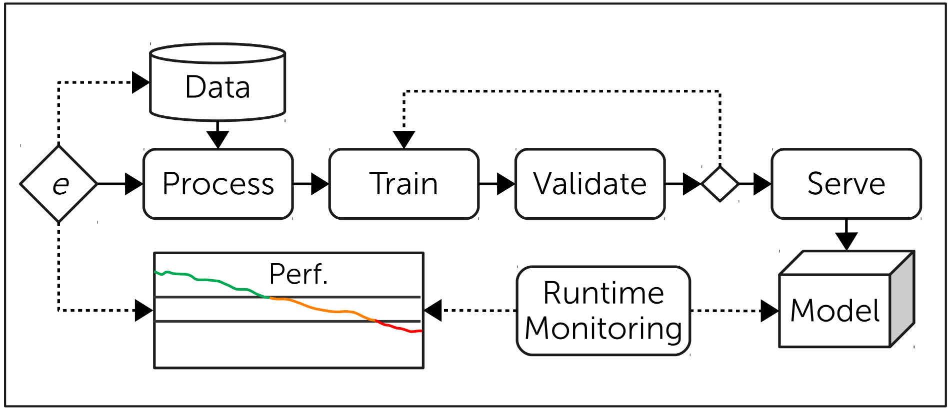
Microsoft
Project BrainWave

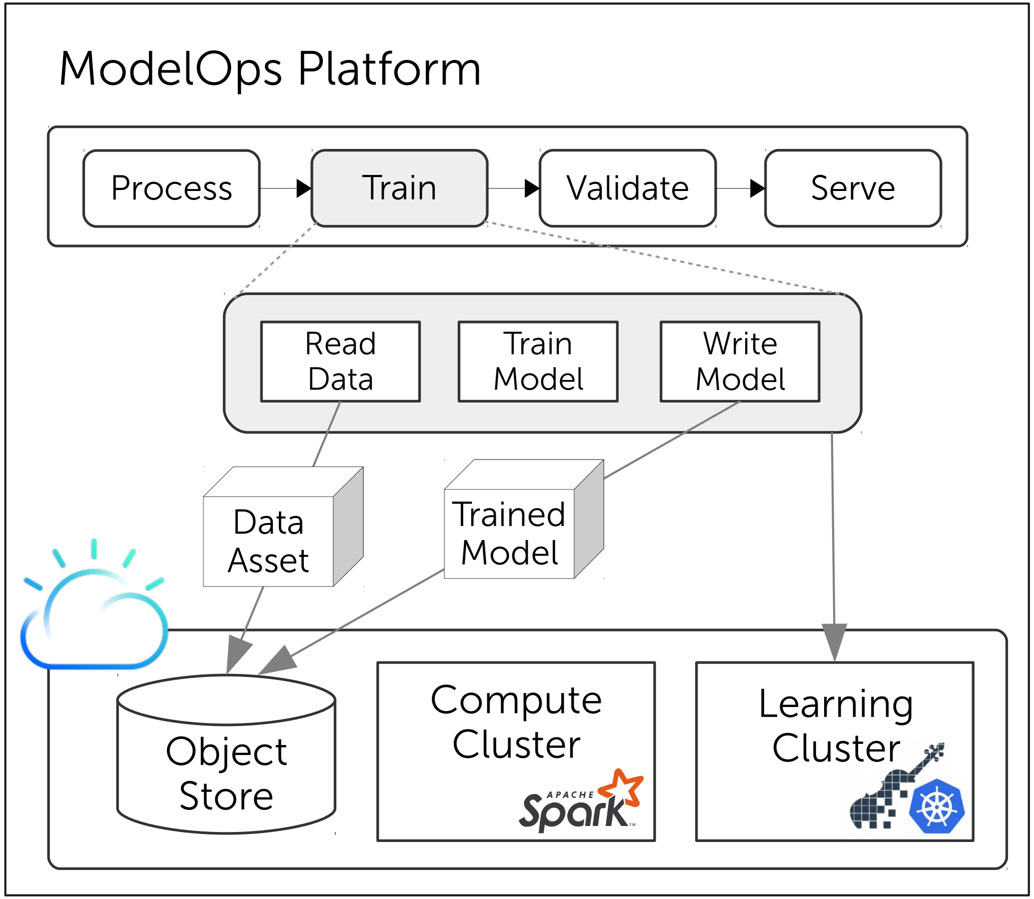


NVIDIA Jetson

Huawei Atlas

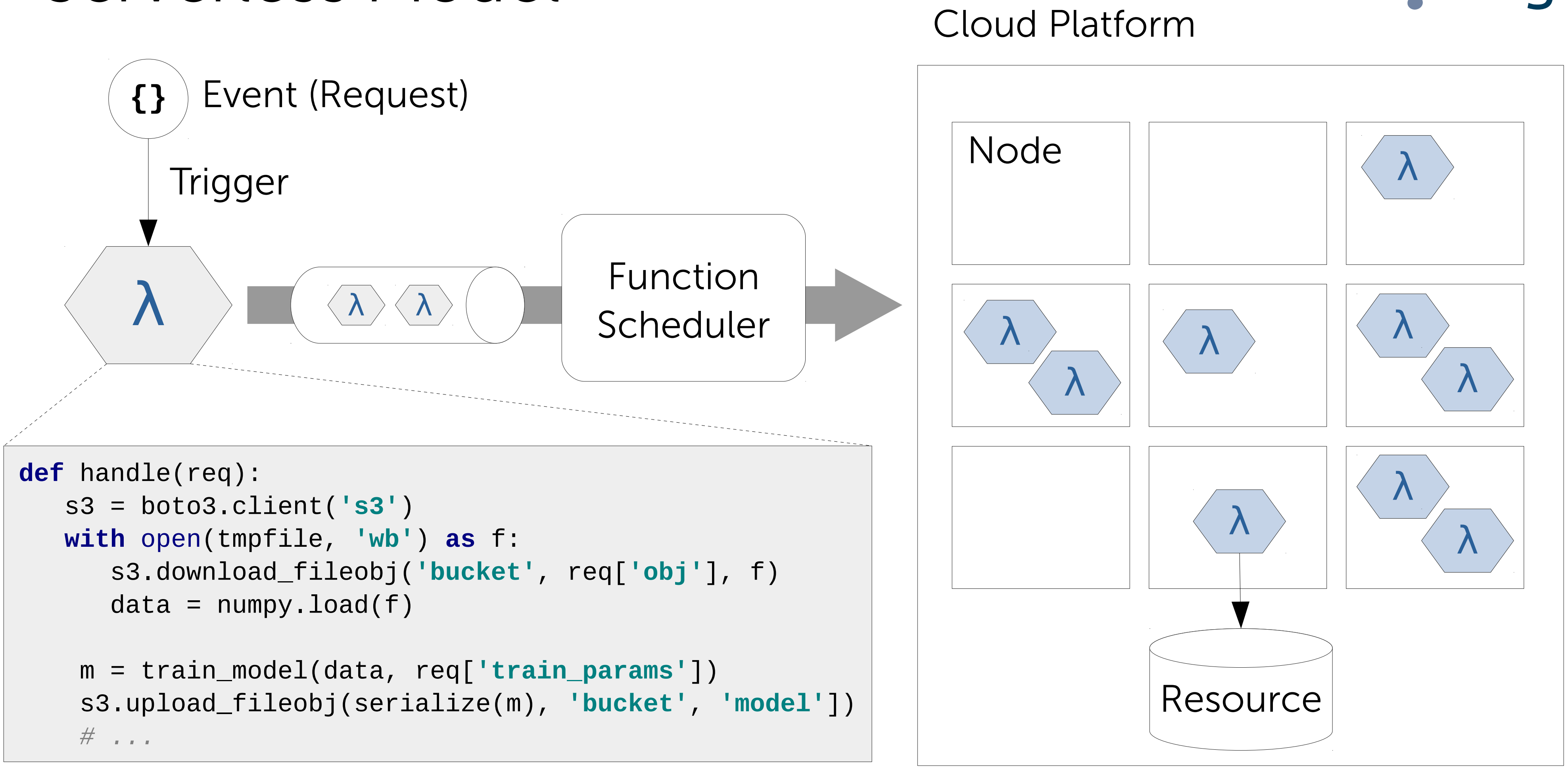
AI Operationalization





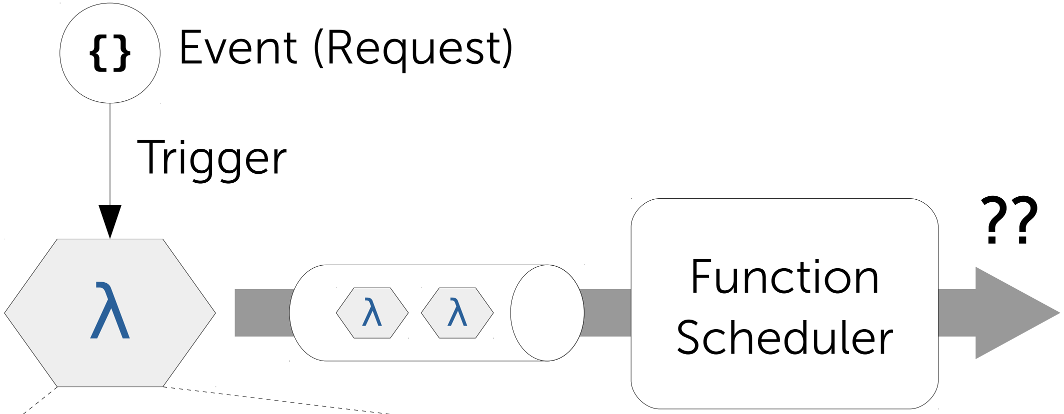
Hummer et al., ModelOps: Cloud-based Lifecycle Management for Reliable and Trusted AI. IC2E'19.

Serverless Model



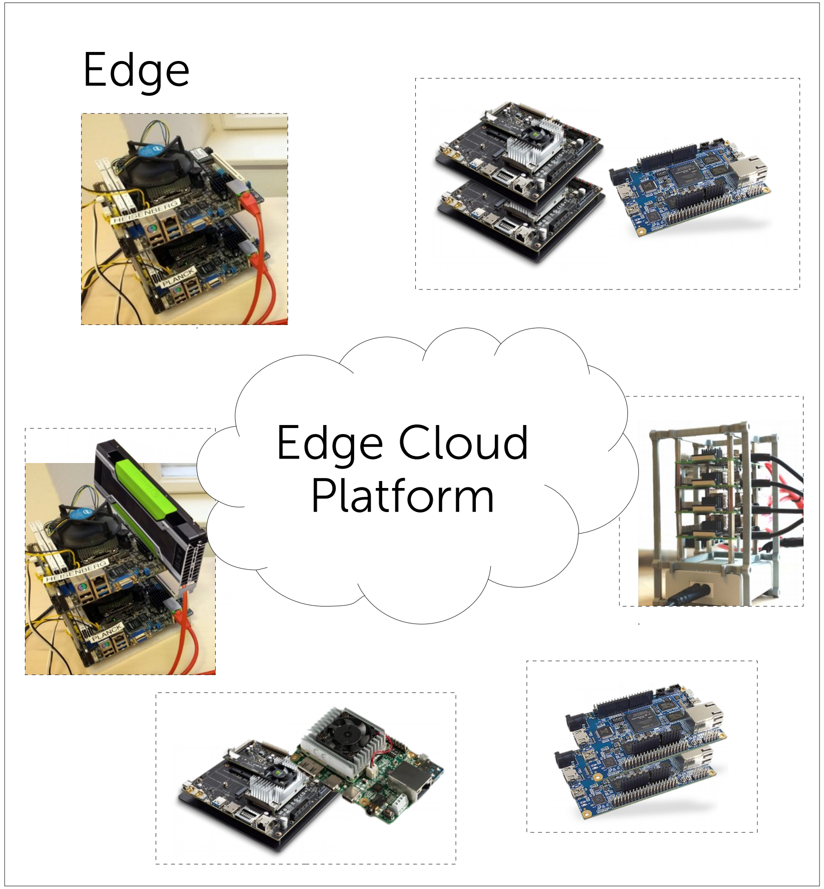

```python
def handle(req):
    s3 = boto3.client('s3')
    with open(tmpfile, 'wb') as f:
        s3.download_fileobj('bucket', req['obj'], f)
        data = numpy.load(f)

   m = train_model(data, req['train_params'])
   s3.upload_fileobj(serialize(m), 'bucket', 'model'])
   # ...
```

Deviceless Model



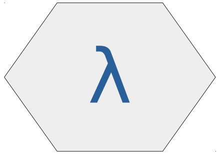

```python
def handle(req):
   s3 = boto3.client('s3')
   with open(tmpfile, 'wb') as f:
      s3.download_fileobj('bucket', req['obj'], f)
      data = numpy.load(f)

   m = train_model(data, req['train_params'])
   s3.upload_fileobj(serialize(m), 'bucket', 'model'])
   # ...
```

Edge Cloud



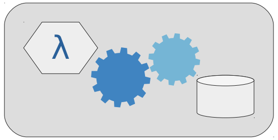
A Serverless Platform for Edge AI

AI Workflow
Programming Model



- Data and Models as first-class citizens
- Model Selectors
- Policies
- Gates

Execution Platform



- Deviceless function scheduling
- Policy enactment
- Context awareness
- Data locality awareness

```python
@consumes.data(
    selector={'urn': 'mnist:data'},
    holdout=0.2)
@produce.model(
    type='classifier',
    urn='mnist:model')
def train(data: Data, request) -> Model:
    arr = data.to_ndarray()
    return Model(train_model(arr))
```

```python
@consumes.model(selector={
    'type': 'image_classifier',
    'data_tags': ['machine_x'],
    'accuracy': '>=0.88'
})
def inference(model: Model, request):
    data = request['input']
    # data prep tasks
    prediction = model.estimate(data)
```

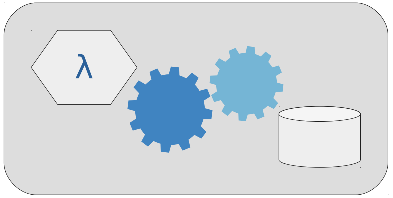
λ

```python
@policy.deadline('2s')

@policy.fn(node = 'user_device',
           capability = 'gpu')

@policy.data(network=['company_network'],
             strict=True)
```

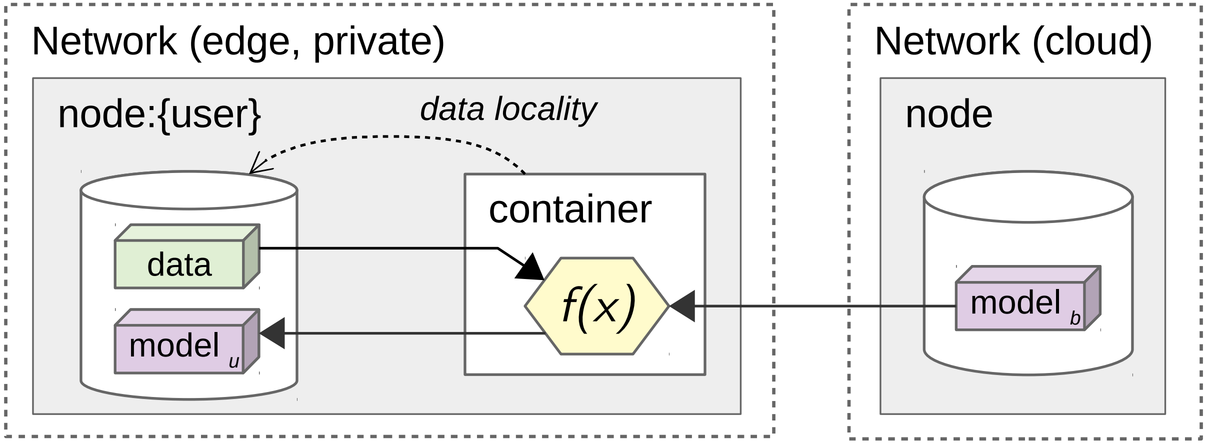
```python
@gate.bias(attribute = 'age',
           predicate = '<0.8')

@gate.drift(metric = 'confidence',
            predicate = '<0.2')
```

```python
@consumes.model(selector={'urn': 'model:base'})
@consumes.data(batch = 100, selector=...)
@produces.model(type='regressor', urn='model:user:{usr}')
@policy.fn(node = 'local')
@policy.data(network = 'local', strict=True)
def refine(model: Model, data: Data):
     ndarr = data.to_ndarray() # data artifact API
     # transfer learning code
     return refined_model
```

Function preprocessor



Scheduler

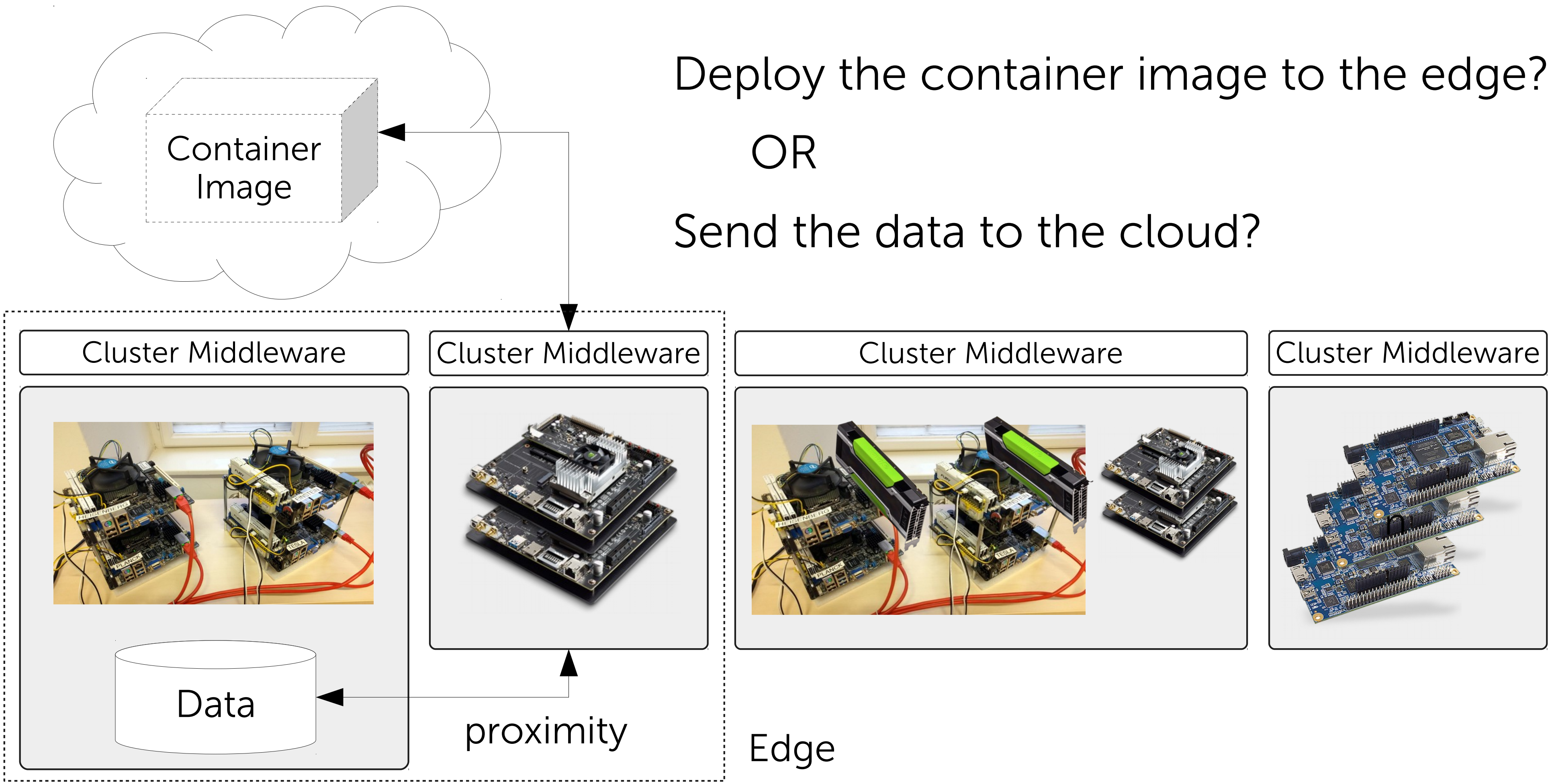


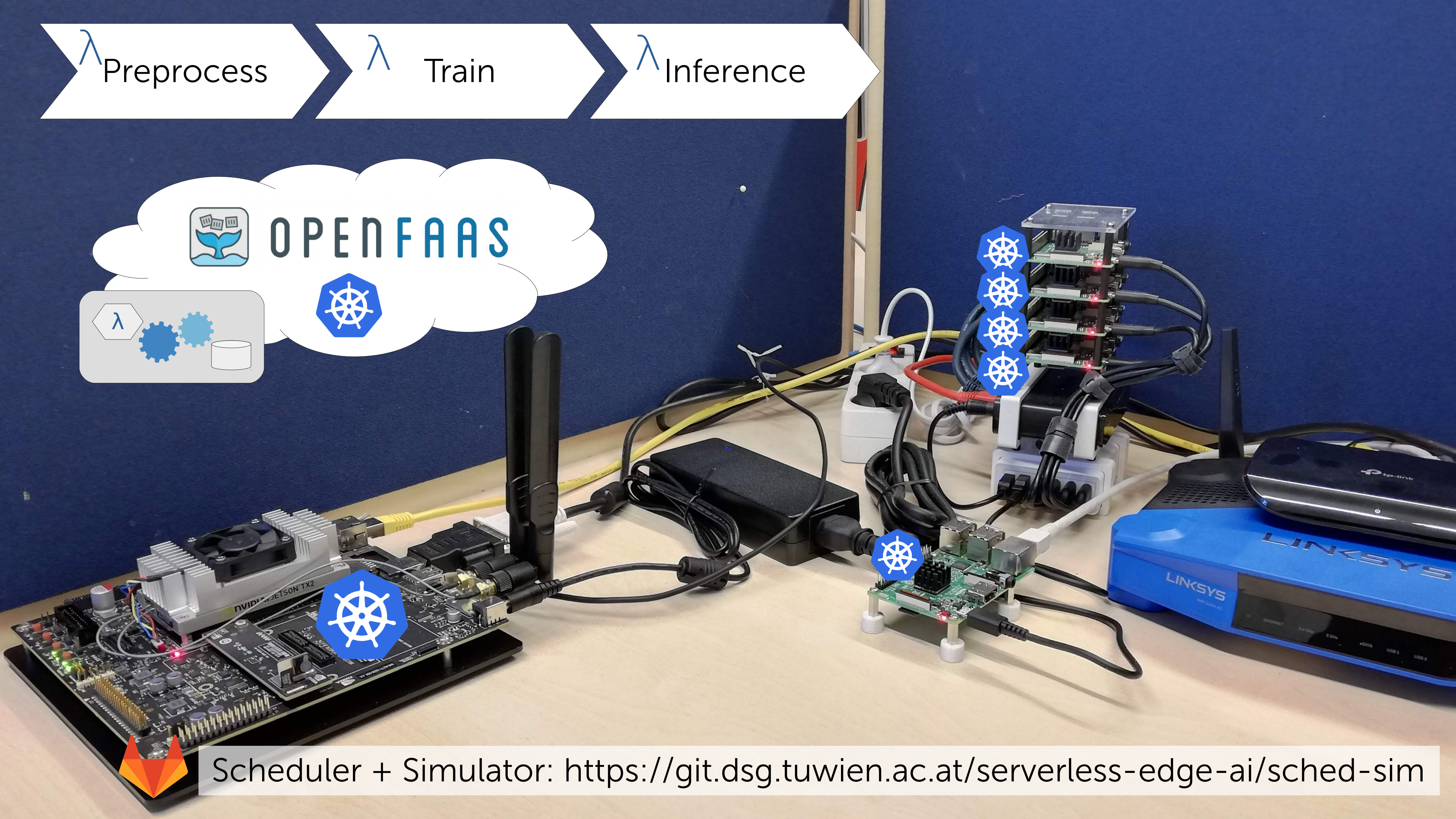
Data Locality Tradeoffs

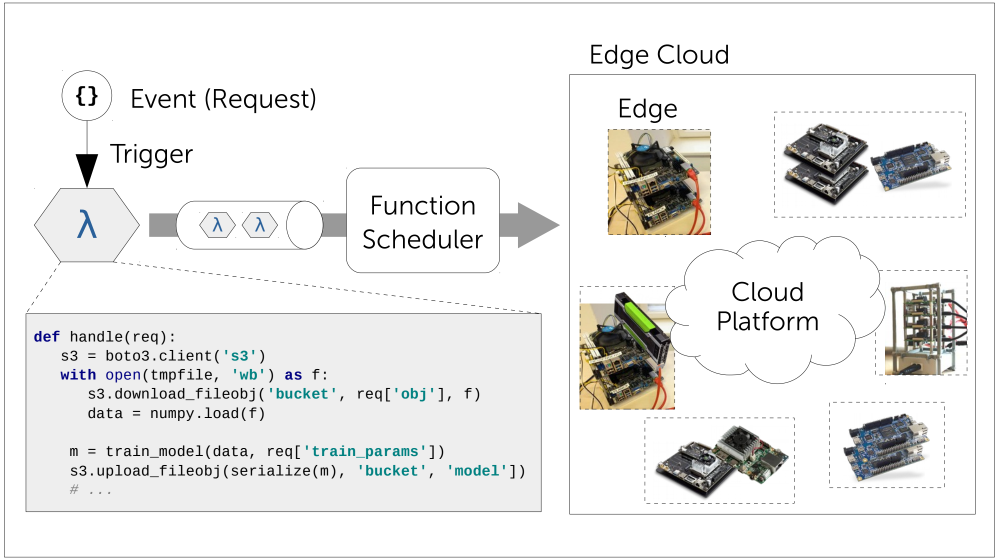
Deploy the container image to the edge?

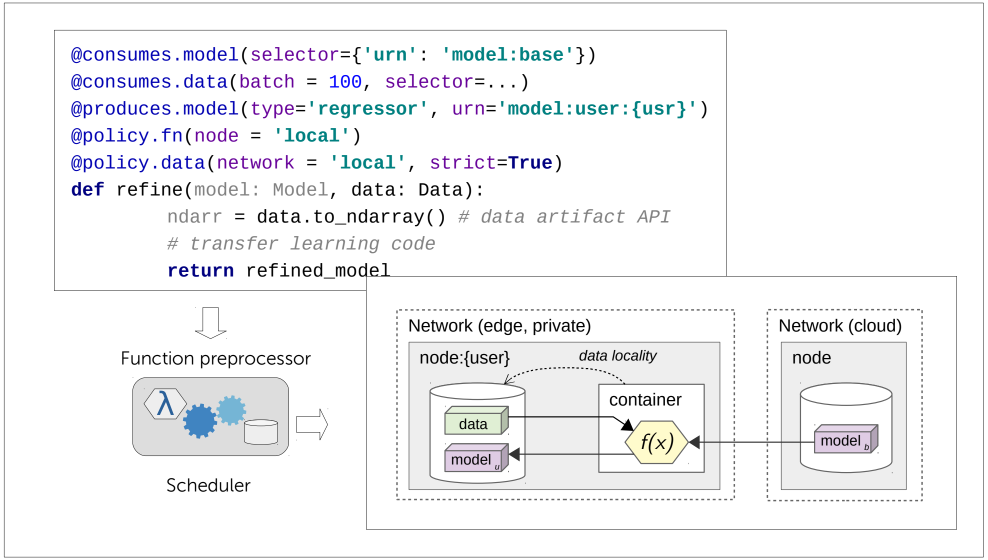
OR

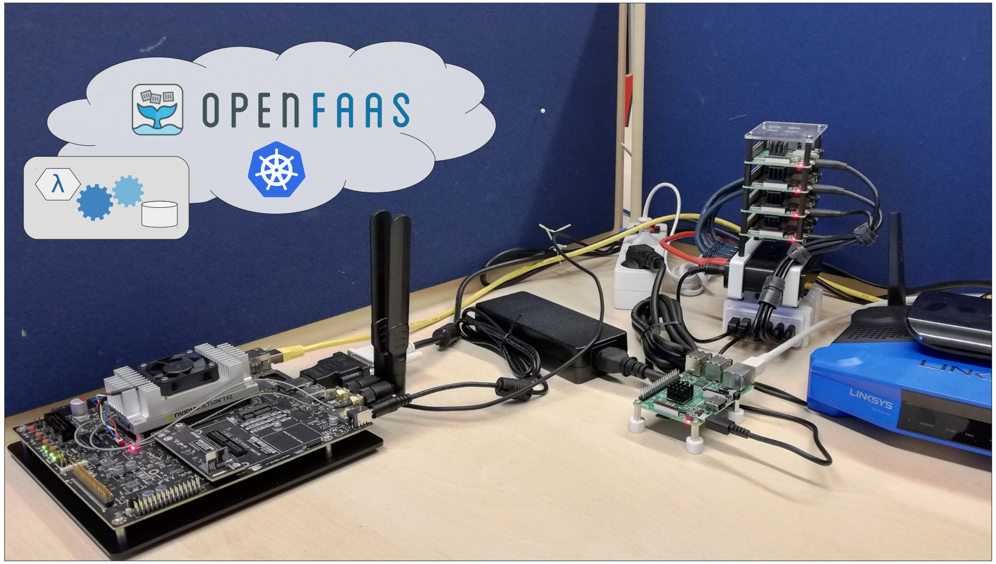
Send the data to the cloud?




λ Preprocess
λ Train
λ Inference
OPENFAAS
λ
LINKSYS
Scheduler + Simulator: https://git.dsg.tuwien.ac.at/serverless-edge-ai/sched-sim







TU WIEN
FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

dsg
DISTRIBUTED SYSTEMS GROUP

Dipl.-Ing. (MSc), BSc
Thomas Rausch
Research Assistant

TU Wien
Institute of Information Systems Engineering
Argentinierstrasse 8-194-02, Vienna, Austria
T: +43 1 58801-184838
E: trausch@dsg.tuwien.ac.at
https://dsg.tuwien.ac.at/staff/trausch

Discussion

i Feedback

- Correct level of abstraction?
- API/SDK features?
- Validation criteria?

ii Controversial points

- Deviceless model (does it work?)
- Transparent data management

iii Open issues

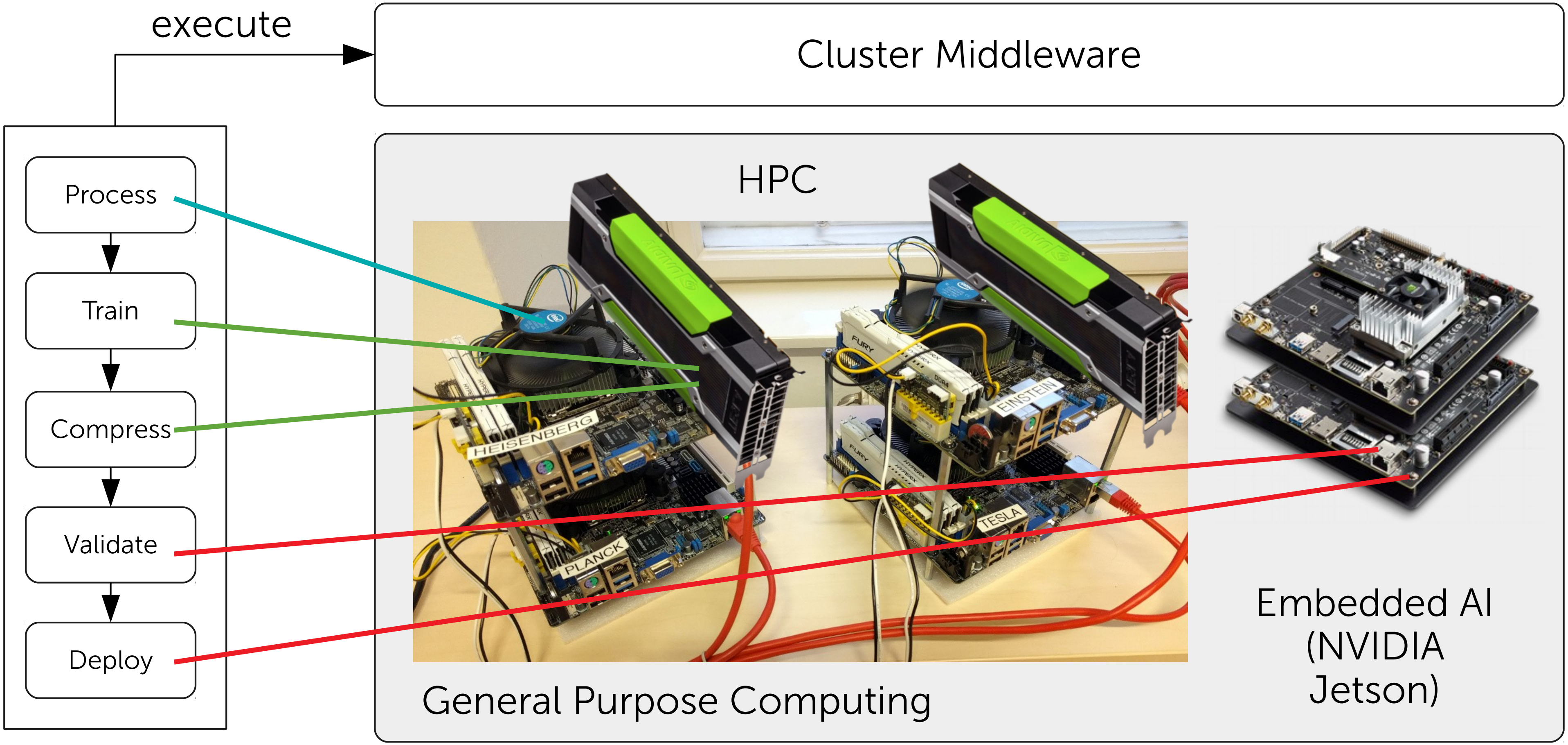
- Request routing architecture
- Proximity and bandwidth monitoring
- Learning optimal placements

iv Failure risks

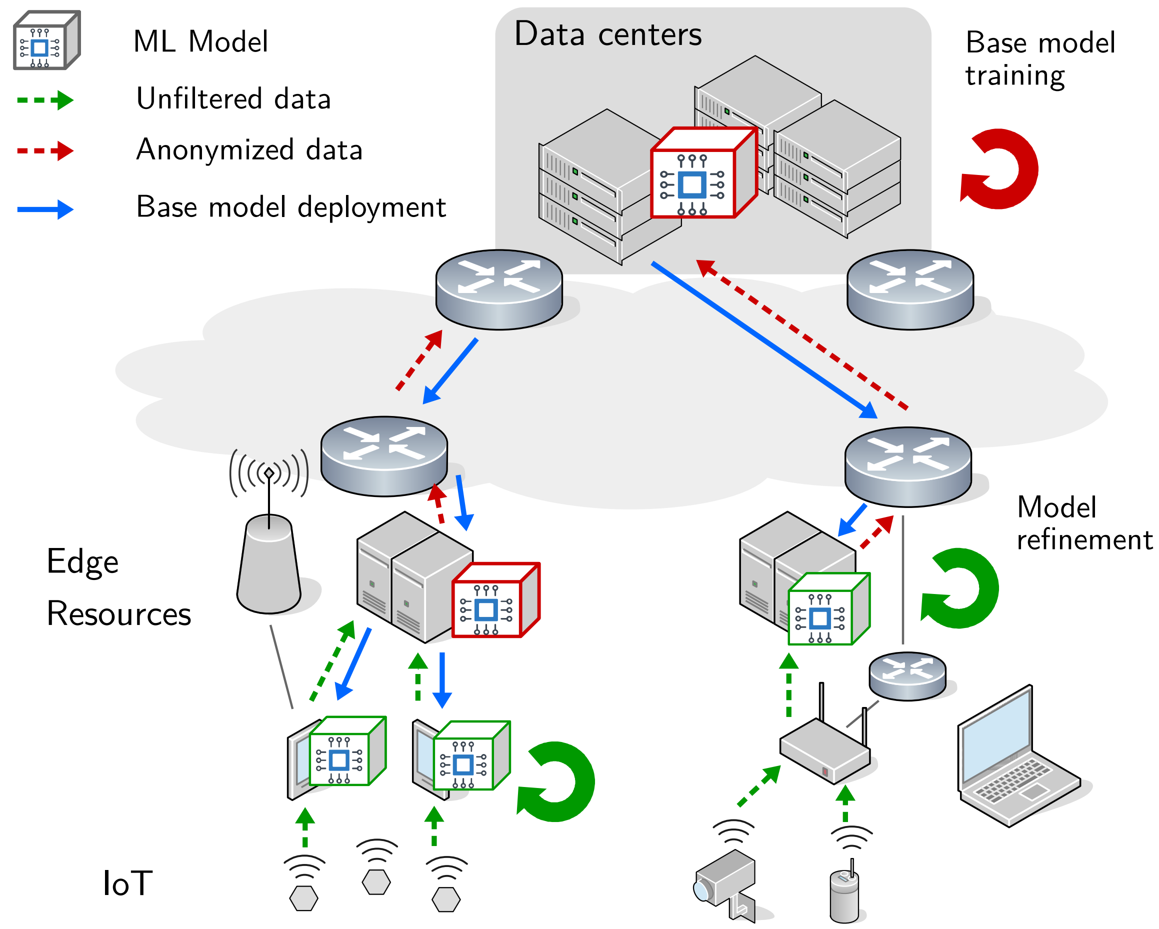
- Model too high-level for scheduler
- “Bring-your-own-device” will fail

Backup Slides

Capability-Aware Pipeline Execution



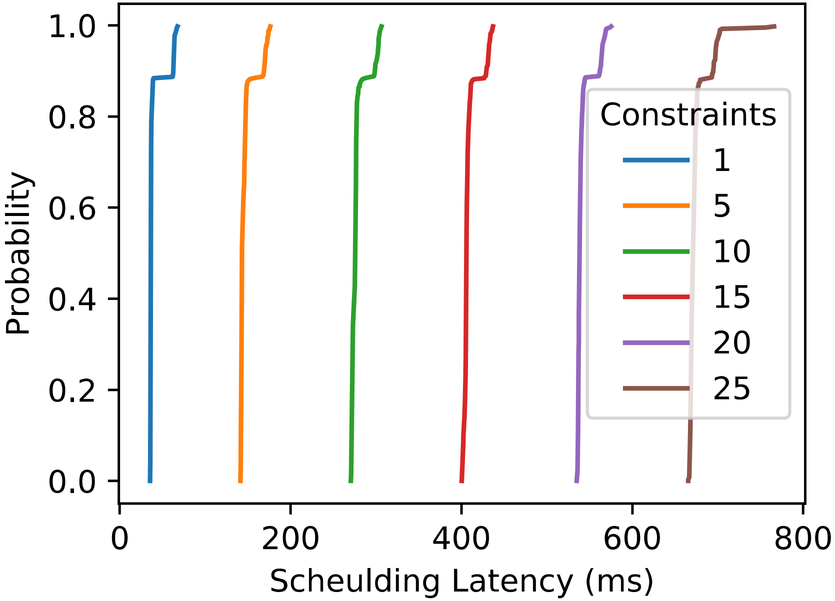
Hierarchical Model Deployments

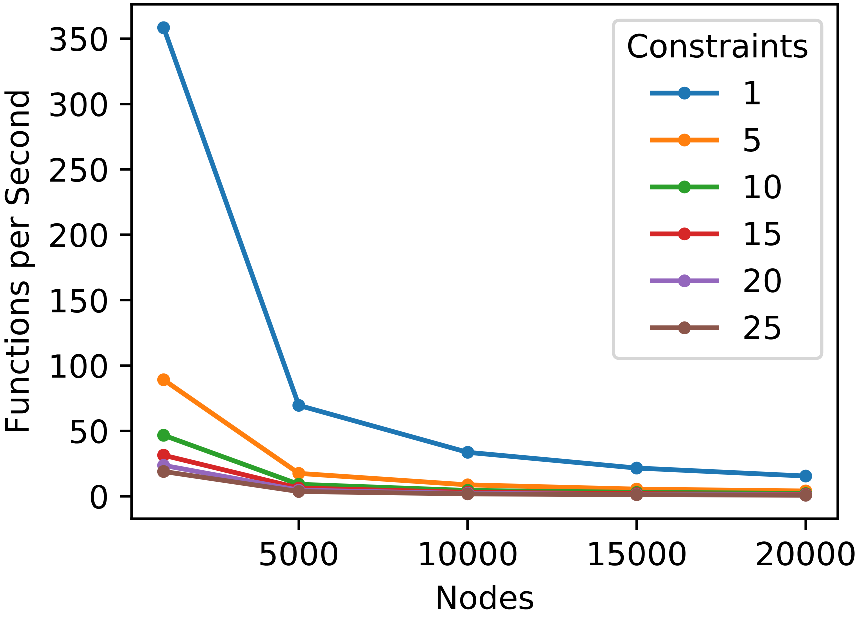


Personal Assistant

Source:
Medtronic
Sugar.IQ
IBM

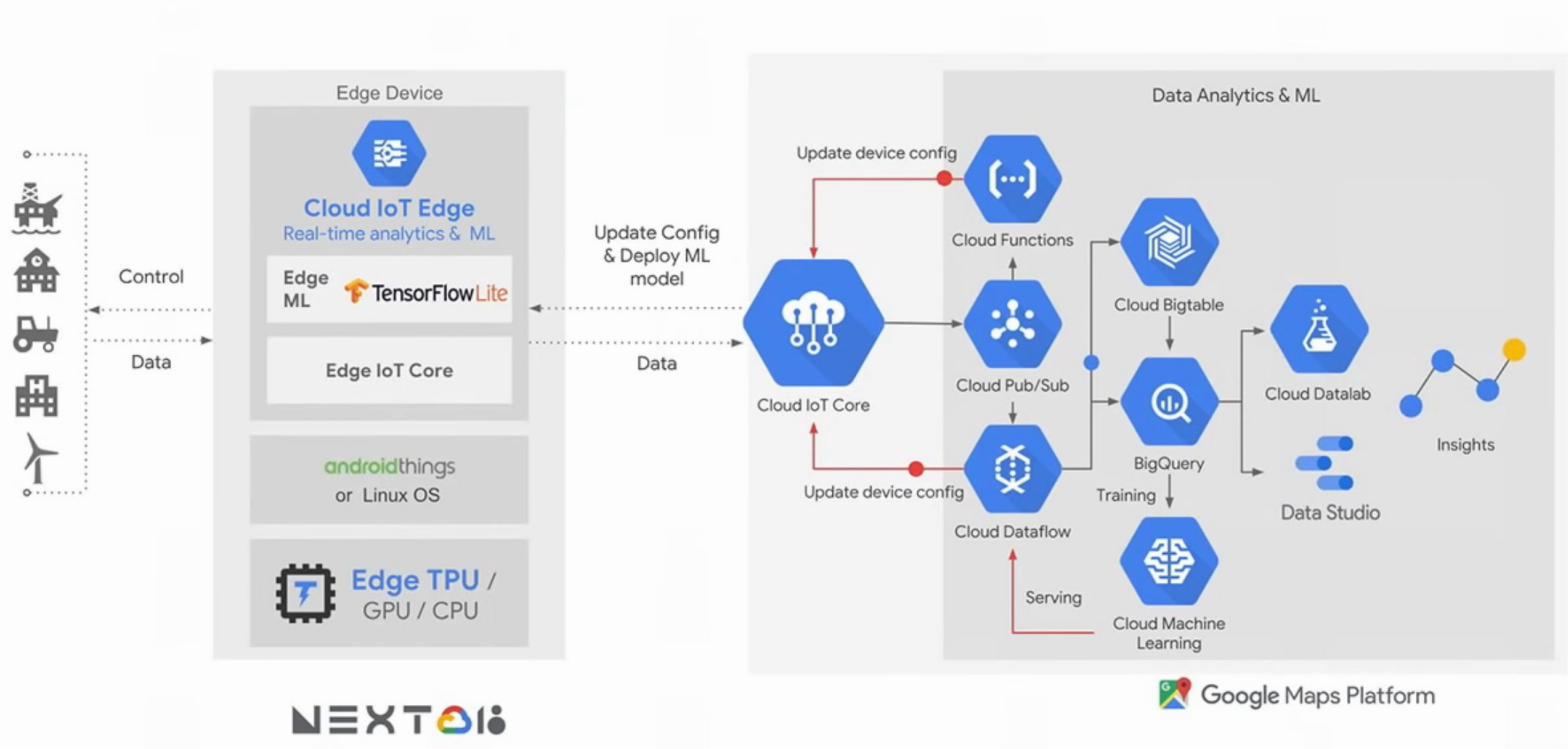
Kubernetes Scheduling Performance





Platform Perspective

dsg

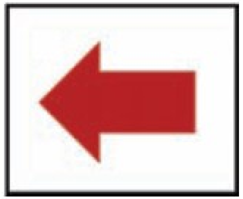
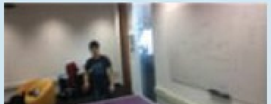


Source: Google Cloud IoT Edge

Cognitive Assistance Applications

Satyanarayanan, M. (2017) *The Emergence of Edge Computing*. Computer.

TABLE 1. Example wearable cognitive assistance applications.

App name	Example input video frame	App description	Symbolic representation	Example guidance
Face		Jogs user's memory of a familiar face whose name cannot be recalled. Detects and extracts a tightly cropped image of each face, then applies popular open source face recognizer OpenFace (cmusatyalab.github.io/openface), which is based on a deep neural network (DNN) algorithm. Whispers name of person. Can be used in combination with mood detection algorithms to offer conversational hints.	ASCII text of name	Whispers "Barack Obama"
Pool		Helps novice pool player aim correctly. Gives continuous visual feedback (left arrow, right arrow, or thumbs up) as user turns cue stick. Correct shot angle is calculated based on widely used fractional aiming system. Uses color, line, contour, and shape detection. Symbolic representation describes positions of cue ball, object ball, target pocket, and top and bottom of cue stick.	<Pocket, object ball, cue ball, cue top, cue bottom>	
Ping-Pong		Tells novice player to hit ball to left or right, depending on which is more likely to beat opponent. Uses color, line, and optical-flow-based motion detection to detect ball, table,	<InRally, ball position, opponent position>	Whispers "Left"