Migrating IoT Processing to Fog Gateways

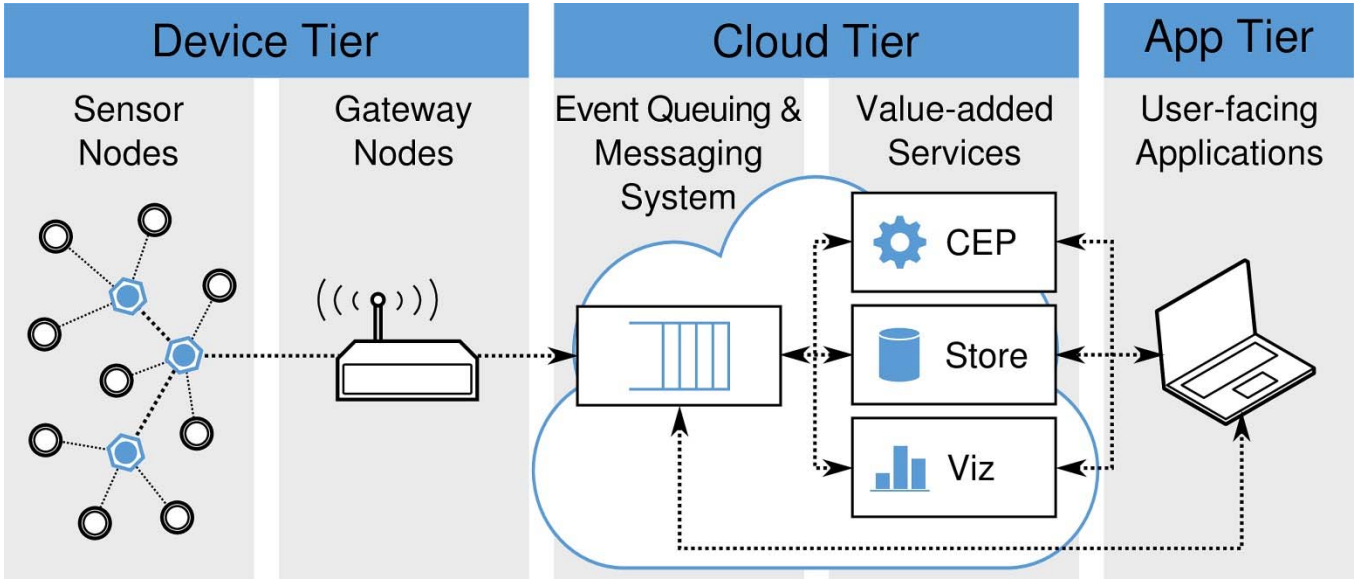
Daniel Happ, Sanjeet Raj Pandey, Vlado Handziski
Telecommunication Networks Group (TKN)
Technische Universität Berlin

Telecommunication Networks Group
Technische Universität Berlin

Cloud centric Internet of Things

- Internet of Things:
  - Connect everything and everyone everywhere to everything and everyone else
  - Abstracting services of trillions of interconnected (sensor) devices
  - Providing interoperability in face of heterogeneity
- Sharing economy of heterogeneous sensors:
  - Sensor devices used for many applications over Internet
  - Middleware for efficiently distribution, storage and processing of sensor data
- Cloud for storage, processing, distribution
  - Flexibility
  - Reliability
  - Pay-as-you-go

Cloud centric IoT reference Architecture



- Producers, services, consumers loosely coupled by event queuing system
- Pub/sub well established data dissemination pattern for IoT

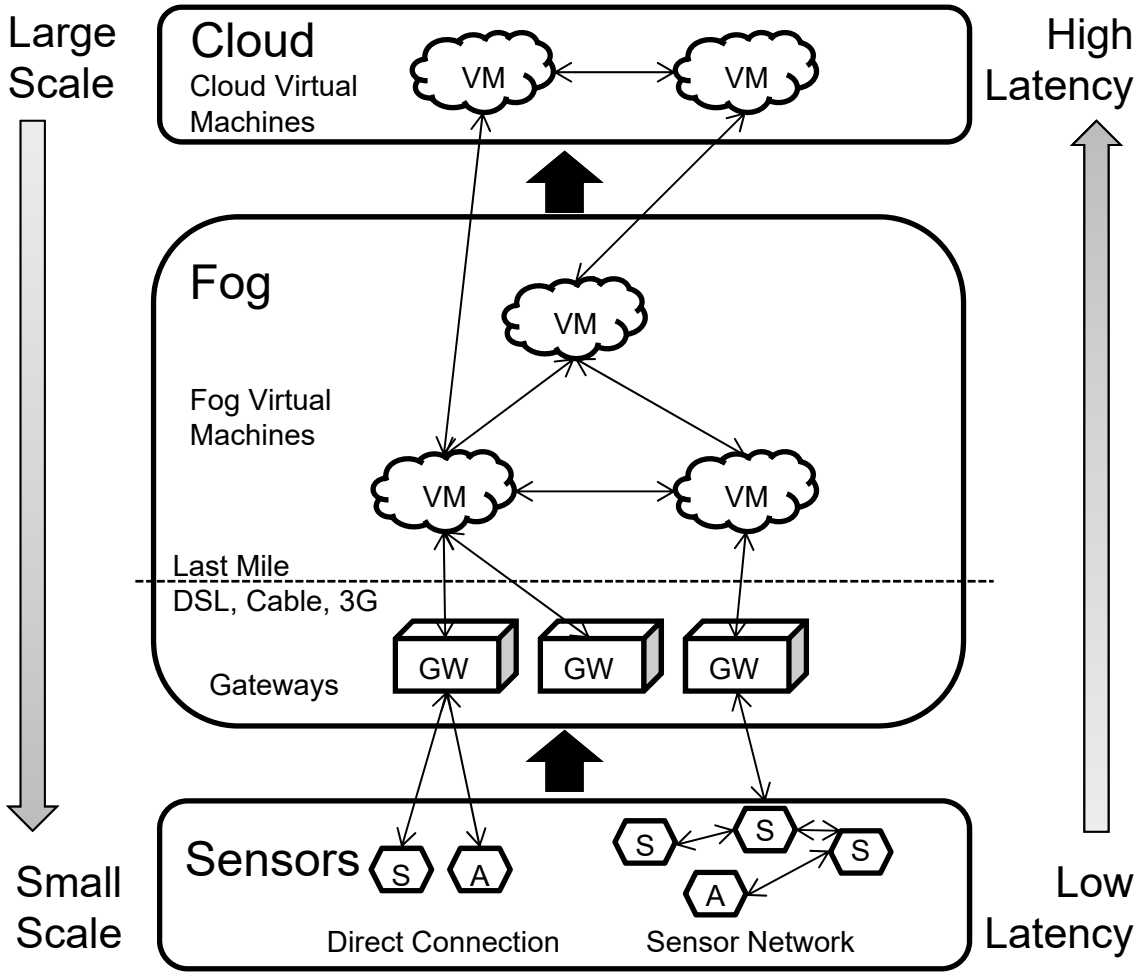
Advantages of Pub/Sub for IoT

- Publish/subscribe properties making it suitable for IoT:
  - Efficient monitoring: "Sense once, notify many"
- Decoupling:
  - Time decoupling
    - do not have to be online at the same time
  - Synchronization decoupling
    - asynchronous communication
  - Space decoupling
    - do not have to explicitly address messages (topic based in this work)

Limitations of Cloud centric IoT

- Placement on "other" edge of the network:
  - Usually not near the customer, but where economically feasible
  - Considerable (unnecessary) delay
  - Usually limited uplink throughput available
- Limited control over data:
  - Unknown or ambiguous privacy policies
  - Privacy protection laws
- Possible solution:
  - Leverage local processing on existing **gateways**
  - Offload processing on demand to Cloud

Fog-based IoT System Model

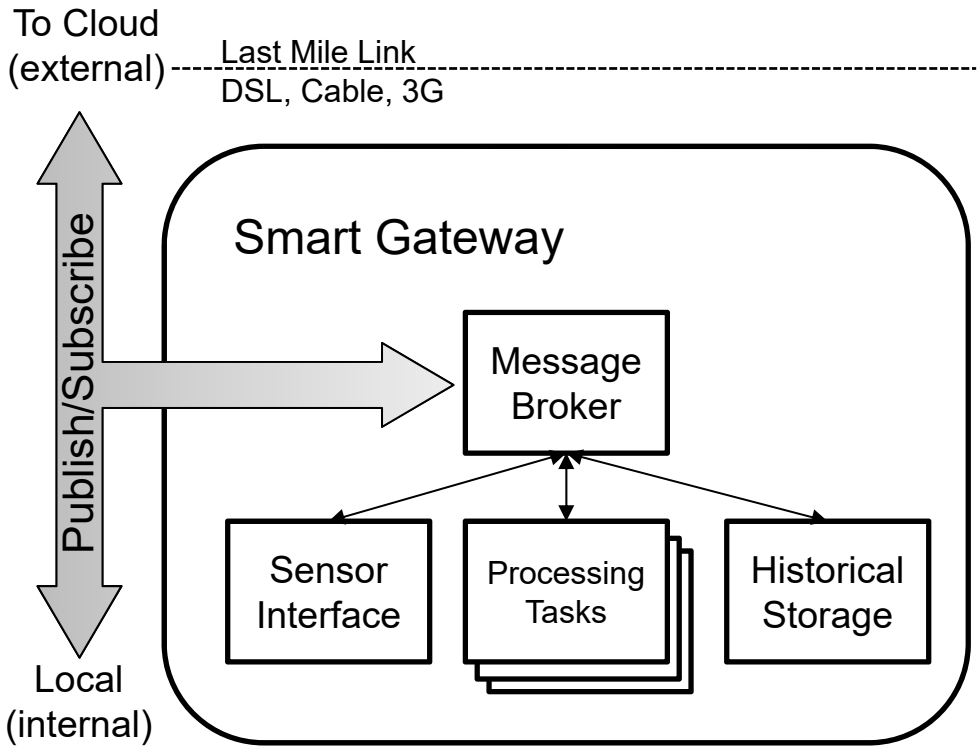


Examples of Gateway Hardware

Device	CPU Clock	Memory	Storage	Price
Intel NUC	1.3 GHz	16 GB	SATA	~$300
BeagleBone Black	1 GHz	512 MB	4GB EMMC, SD	$55
Raspberry Pi 3	1.2 GHz	1 GB	SD	~$35
TP-Link TL-WR841N	650 MHz	32 MB	4 MB	~$20

- Existing gateways potentially have unused processing and storage capabilities right at the edge
- Gateways still severely constrained:
  - Limited CPU power
  - Limited memory
  - Limited Storage
  - Limited Uplink Bandwidth
  - But: Usually no power constraints

Proposed Gateway Model



- Application view: Pub/sub is virtual bus
- Local broker enables local dissemination, processing, storage of messages

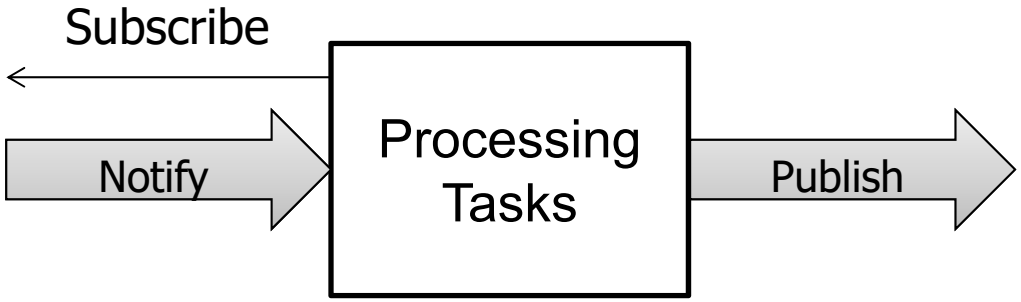
Problem Statement

- On demand offloading of processing tasks to Cloud necessary

- How can publish/subscribe help moving processing to fog gateways?
  - Task Migration
  - (Profiling)

- In this work:
  - (Architecture outline for offloading of IoT processing tasks)
  - Processing task offloading scheme on top of pub/sub
  - Proof-of-concept implementation and evaluation of offloading on top of MQTT

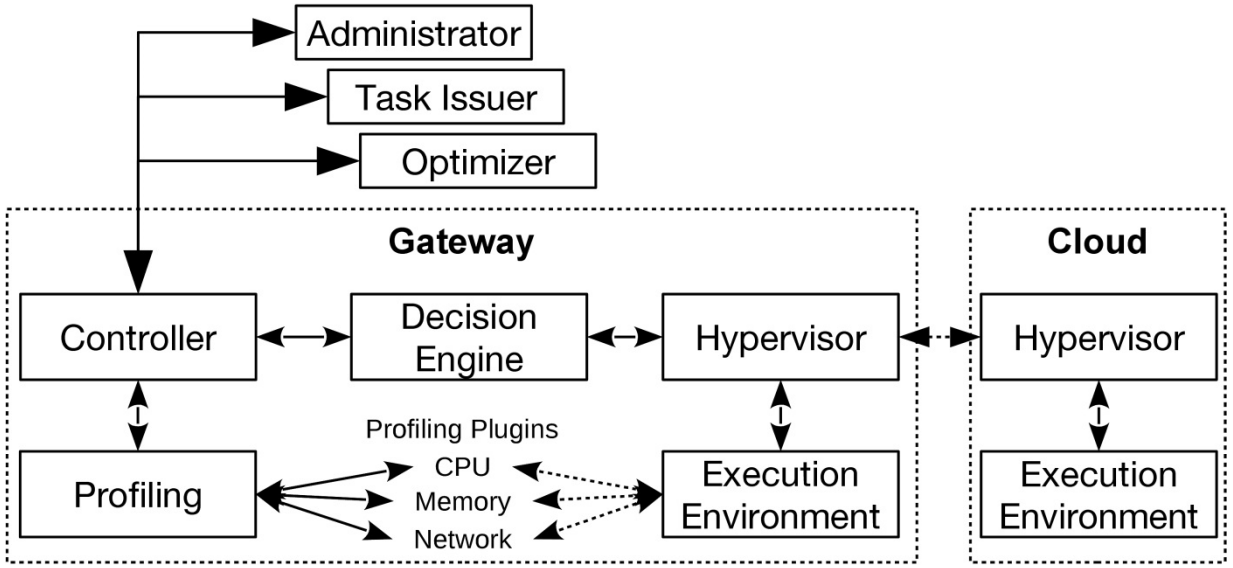
Sensor Data Processing Task



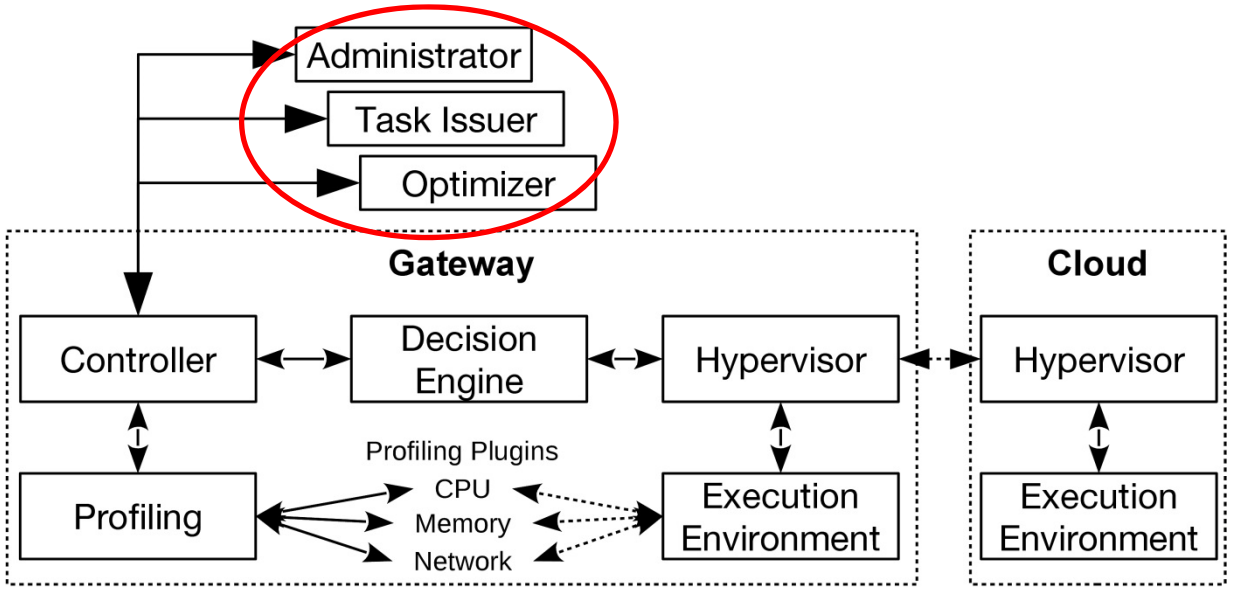
- Continuously running
- Input: Subscription on a set of input topics
- Computation: Arbitrary
- Output: Publication(s) on set of output topics

- Default place of processing: **Gateway**
- Offloaded to Cloud when resources not available
- Issuer defines additional constraints and requirements

Offloading Architecture

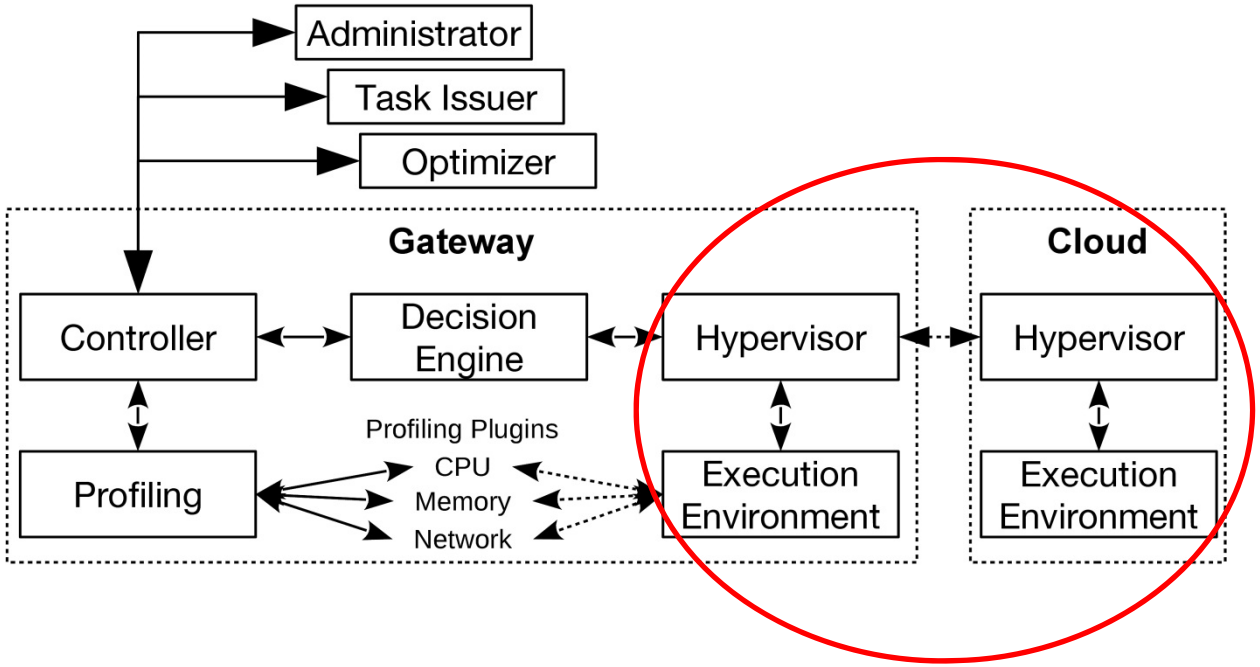


Offloading Architecture



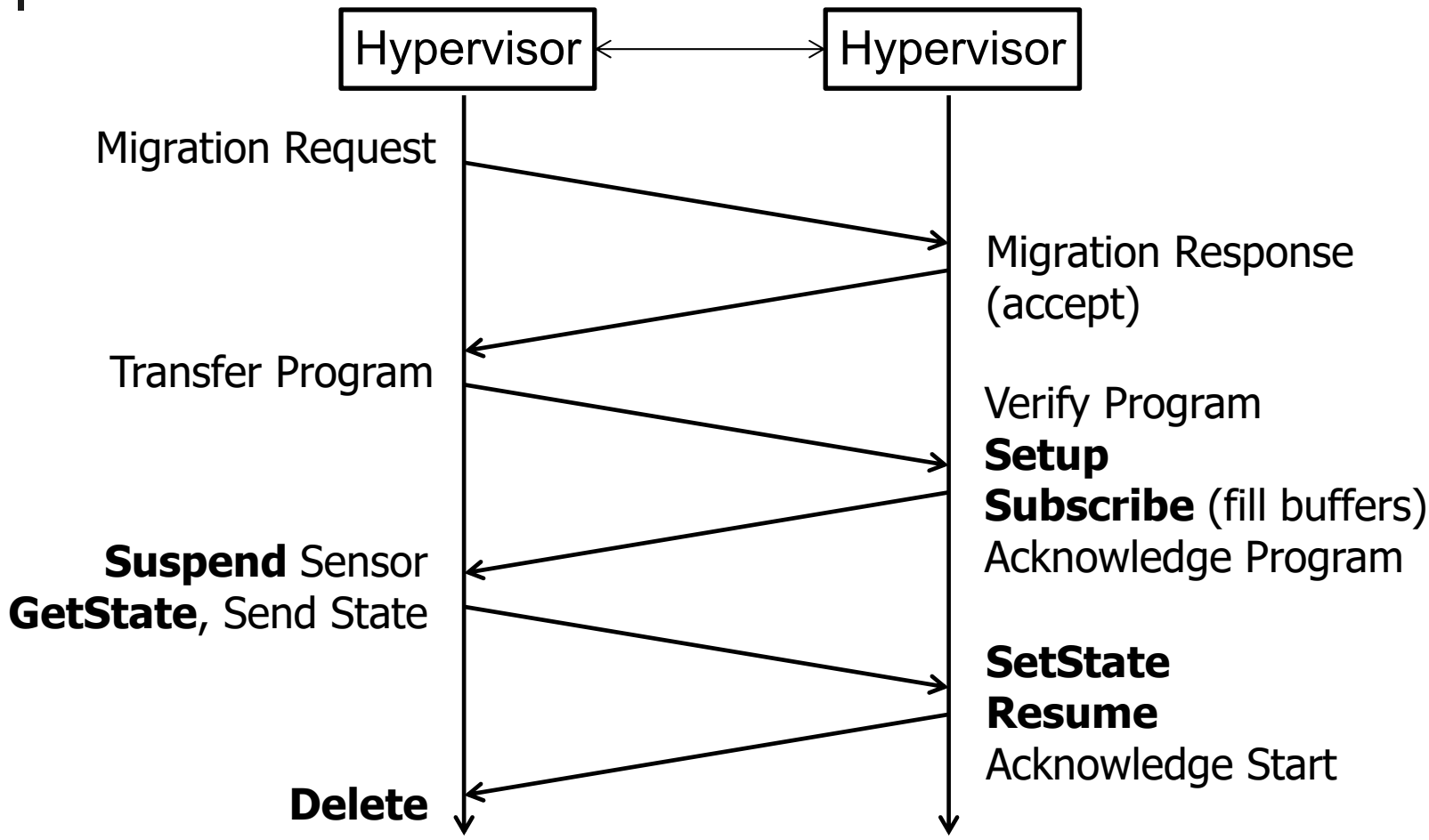
- 3 distinct roles that interact with the system:
  - Administrator defines computational constraints
  - Issuer defines task and privacy and computation requirements
  - Optimizer defines the optimization goal

Offloading Architecture



- Migration Framework executes decision to transfer

Virtual Sensor Migration Sequence



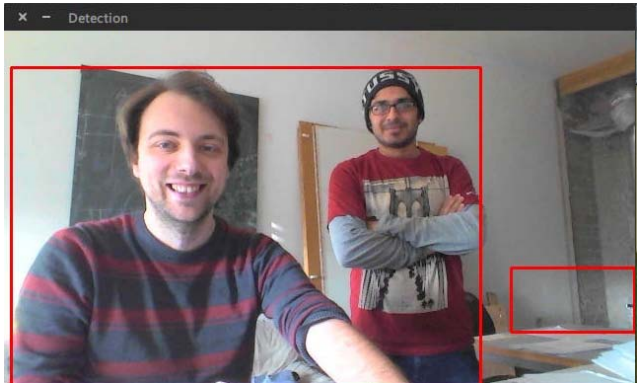
Virtual Sensor Migration Interface

Interface provided by processing task:

- **setup** (called before launch to initialize task)
- **subscribe** (subscribe to input topics and fill buffers)
- **setState** (set initial state to representation given)
- **resume** (resume processing)
- **suspend** (stop processing)
- **getState** (get representation of the process state)
- **delete** (terminate the processing task)

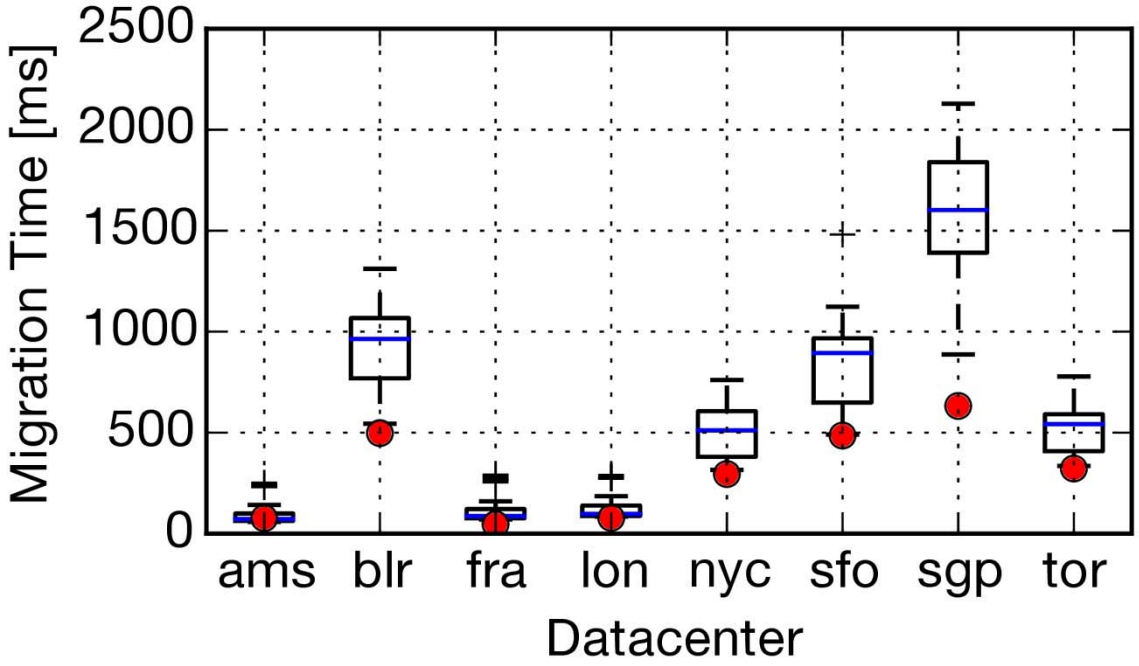
Migration Prototype & Evaluation

- Sample Application:
  - Motion detection App
  - Using python, cv2, MQTT
  - Code: 3kb
  - State (Background image): 21kb
  - Resolution: 640x360, 2fps



- Setup:
  - Beaglebone Black (TU Berlin) + Cloud (Digital Ocean)
  - Series of 30s with camera pictures repeated
  - Process is running on gateway and migrated back and forth 64 times to and from Cloud (8 locations)

Evaluation of Migration Time



- Estimated time using throughput and RTT (red)
- Takeaway:
  - With current technologies: migration in few seconds
  - Estimate lower bound for migration time

Conclusion

- Full potential of IoT only fully utilized through increased edge processing
- Presented offloading scheme on top of pub/sub
- Prototypical implementation for MQTT
- Migration time in the order of seconds

- Future work
  - Policy (Placement Problem)
  - Mapping of virtual bus to heterogeneous implementations (MQTT, Kafka, RabbitMQ, etc.)
  - Broker Placement, Routing

Q&A

Questions?

Optimization Problem Formulation (1)

- Optimizing for (monetary) cost:

$$\min_{x_i \in 0,1} (c_{transfer} \cdot \omega_{tr} + c_{memory} \cdot \omega_{mem} + c_{cpu} \cdot \omega_{cpu}) \quad (1)$$

where

$$c_{transfer} = \sum_{i=1}^{n} x_i \quad (rx_{i,loc} + tx_{i,loc}) + (1 - x_i)(rx_{i,rem} + tx_{i,rem}) \quad (2)$$

$$c_{memory} = \sum_{i=1}^{n} mem_i \cdot x_i \quad (3)$$

$$c_{cpu} = \sum_{i=1}^{n} cpu_i \cdot x_i \quad (4)$$

Optimization Problem Formulation (2)

- Constraints of Gateway Hardware:

$$\sum_{i=1}^{n} cpu_i \cdot (1 - x_i) \leq avail_{cpu} \cdot f_{cpu} \quad (5)$$

$$\sum_{i=1}^{n} mem_i \cdot (1 - x_i) \leq avail_{mem} \cdot f_{mem} \quad (6)$$

$$\sum_{i=1}^{n} rx_{i,rem} \cdot (1 - x_i) + tx_{i,loc} \cdot x_i \leq avail_{bw\downarrow} * f_{rx} \quad (7)$$

$$\sum_{i=1}^{n} tx_{i,rem} \cdot (1 - x_i) + rx_{i,loc} \cdot x_i \leq avail_{bw\uparrow} * f_{tx} \quad (8)$$