Dynamic and Scalable Deployment of Edge Internet-of-Things Analytics

蔡霈萱 Pei-Hsuan Tsai

Outline

- Introduction and Motivation
- Dynamic Deployment
- Edge Analytics
- System Overview
- Implementation and Demo Scenarios
- Evaluations
- Related Works
- Conclusion and Future Work

Introduction and Motivation

Motivation

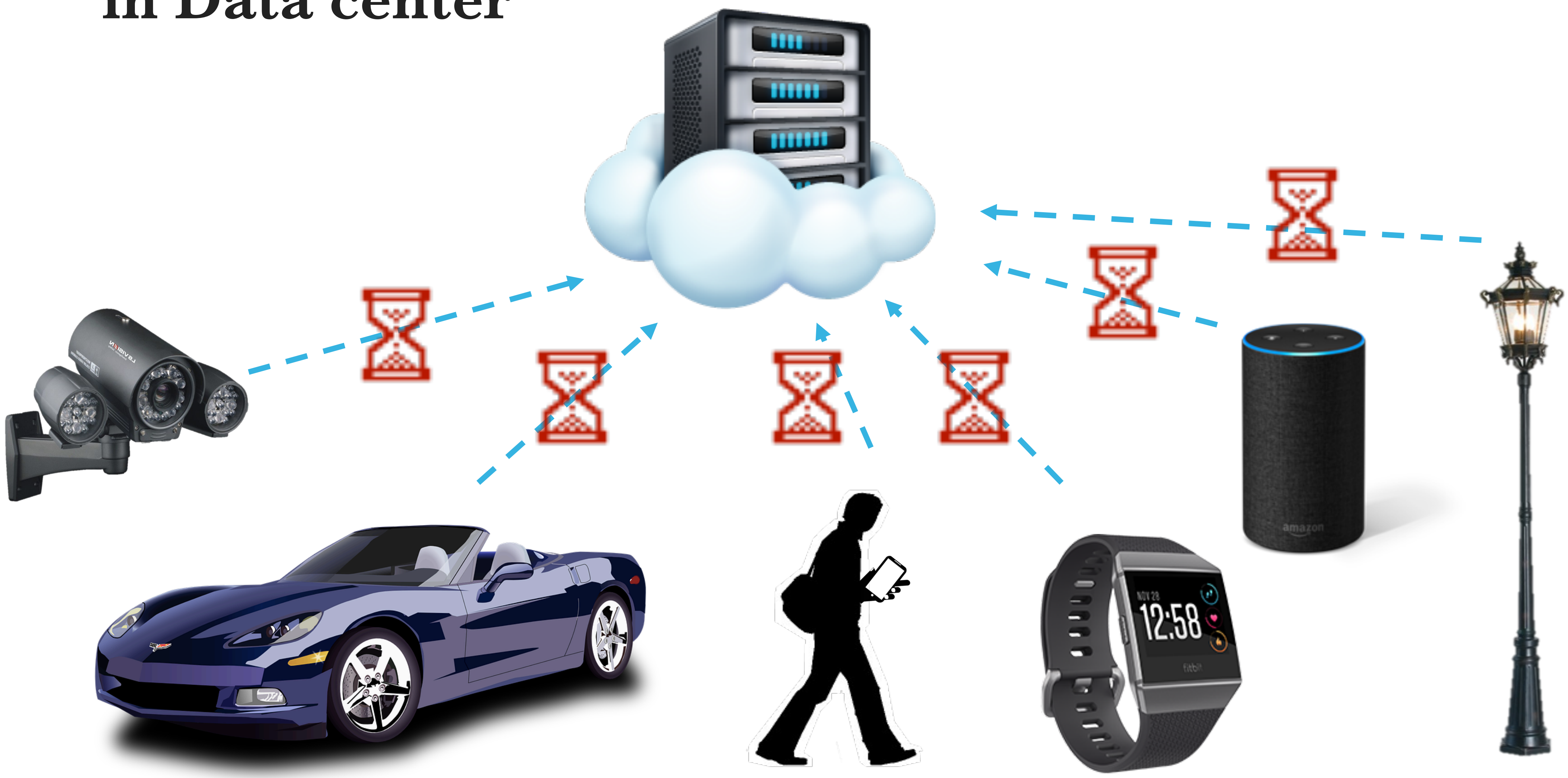
- Internet of Things (IoT) grows rapidly
- Produce incredible amount of data
  - Overload the data centers and congest the networks seriously

Category	2016	2017	2018	2020
Consumer	3,963.0	5,244.3	7,036.3	12,863.0
Business: Cross-Industry	1,102.1	1,501.0	2,132.6	4,381.4
Business: Vertical-Specific	1,316.6	1,635.4	2,027.7	3,171.0
Grand Total	**6,381.8**	**8,380.6**	**11,196.6**	**20,415.4**

Source: Gartner (January 2017)

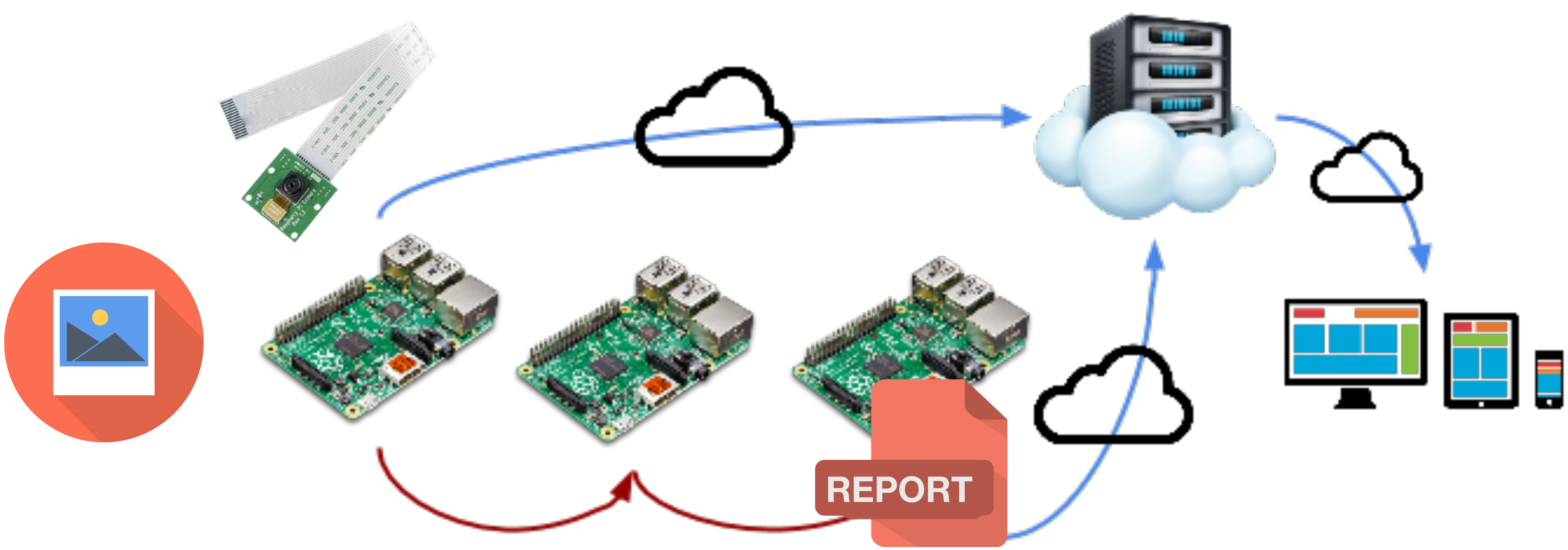
Limitations of Current Solution

Analyze and Compute
Huge Amount of Data
in Data center



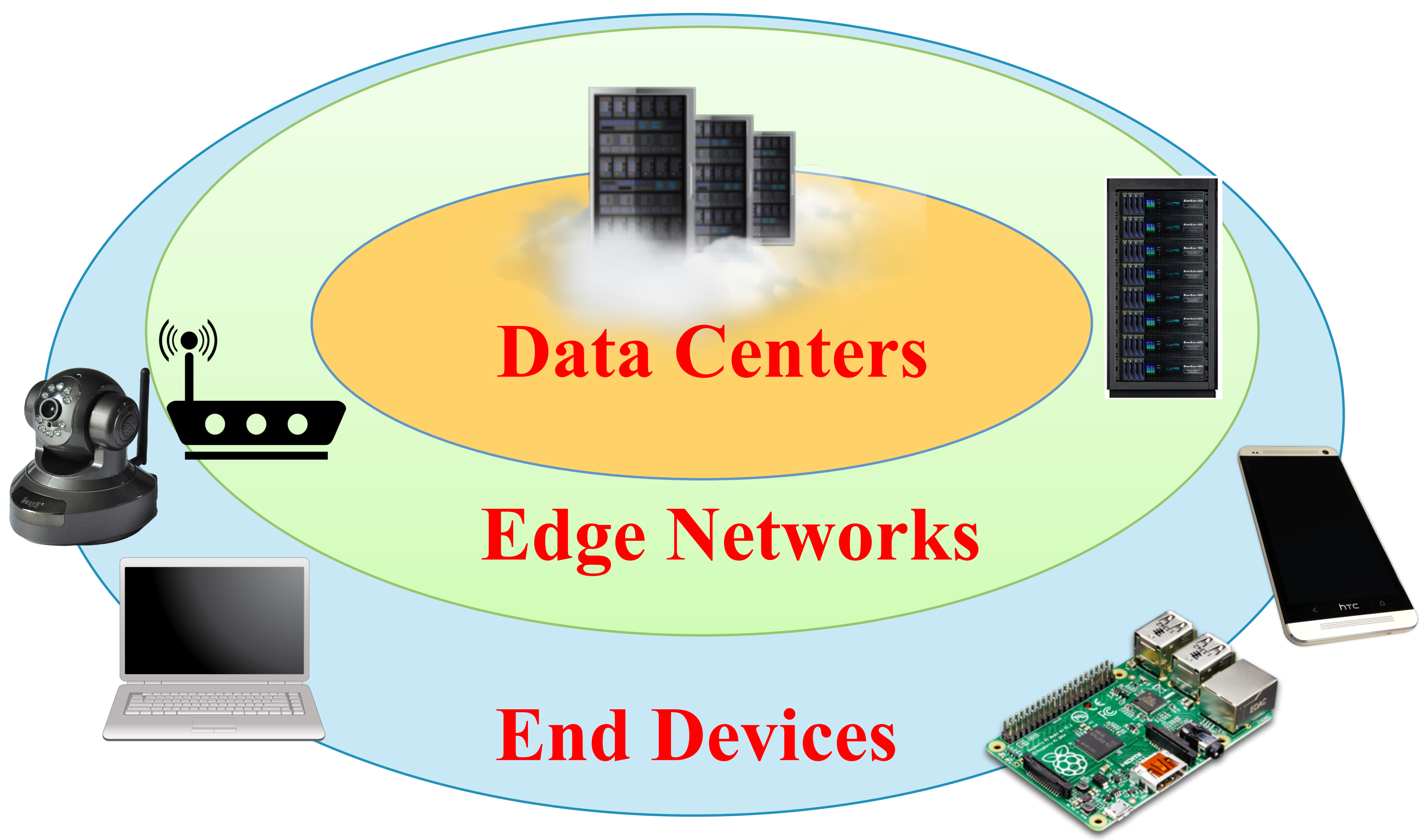
Edge Analytics - Pre-processing

- Reduce latency
- Reduce network traffic
- Reduce the load of data centers



Fog Computing

- Fog computing leverages devices in data centers, edge networks, and end devices in simultaneously

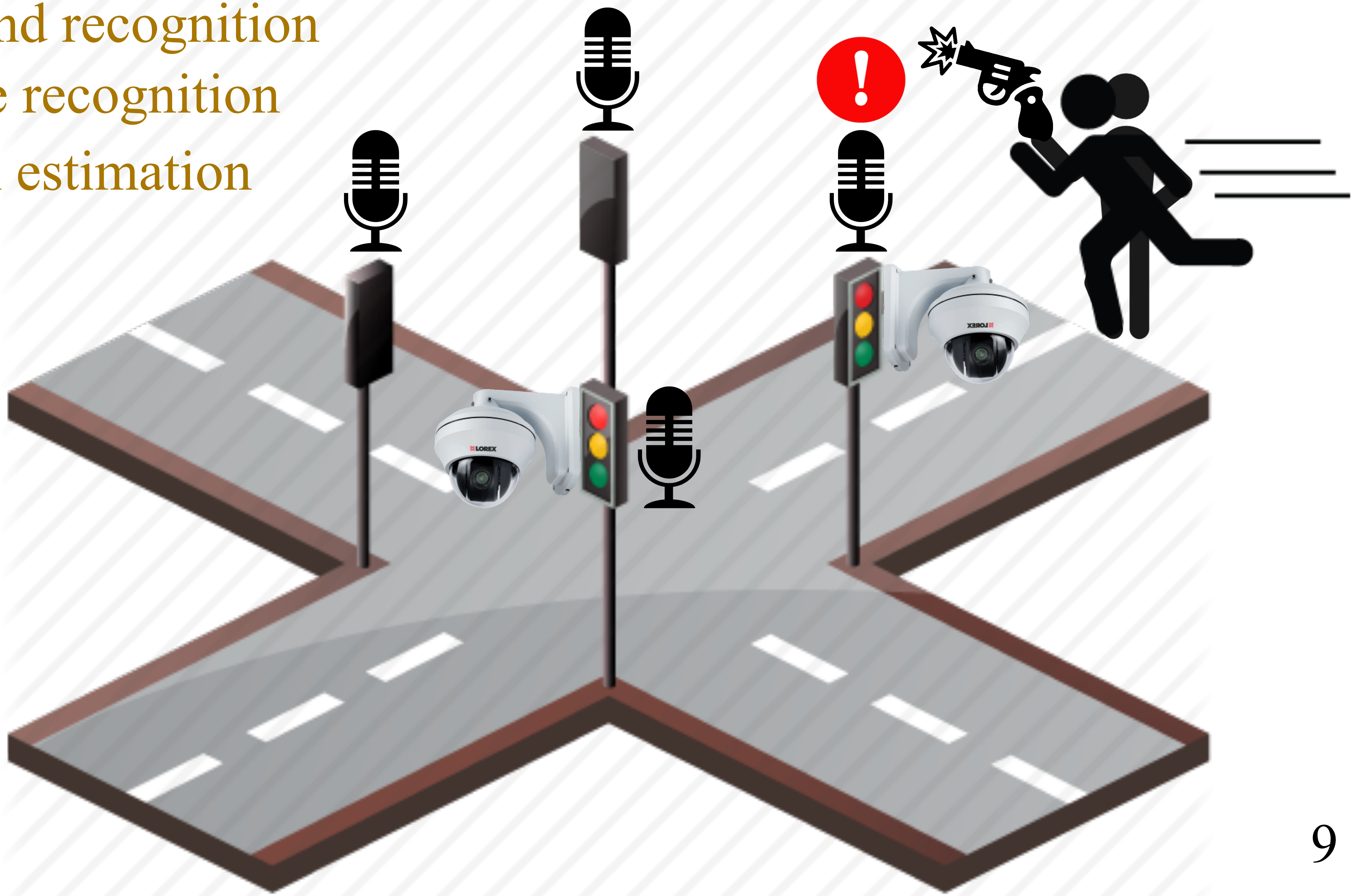


Advantages: Fog >> Cloud

- Diverse kinds of resources
  - Computations, communications, storage, and sensors
- Utilize wasted resources
- Reduce network traffic
- Short response time
- Low cost
- Low carbon foot print
- …

Shooter Tracking Usage Scenario

- Sound recognition
- Face recognition
- Path estimation



Dynamic Deployment

Dynamic Deployment Mechanism

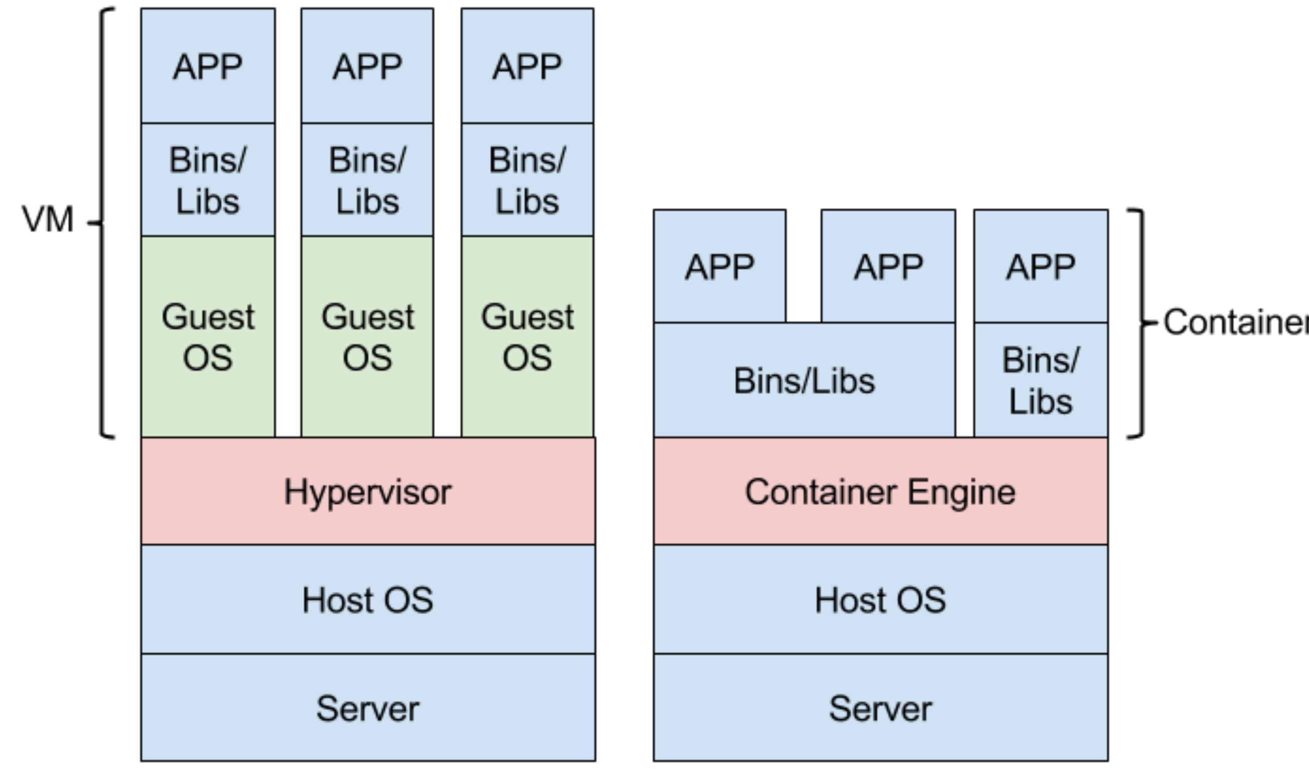
- Frequently updating or replacing the applications
  - Container-based applications
- Managing lots of fog devices and applications
  - Orchestration tool
- Triggering another application when something happened
  - Event-driven mechanism

Virtualization Technology

- Virtualized modules
  - Dynamically placed on the fog devices
  - Migrated among the fog devices
  - Allocated the resource on-demand
  - More private
- Traditional virtual machine v.s. container
  - Xen, KVM
  - LXC, Docker

Traditional VM v.s Container

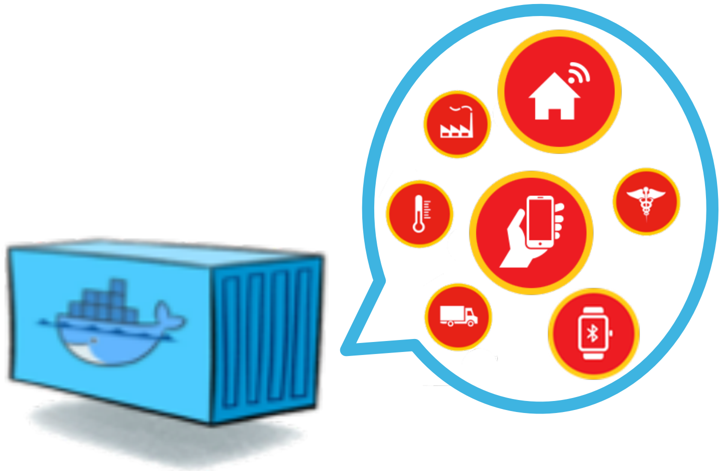
- Container
  - Share the same OS kernel, and use the namespaces to distinguish one from another
- Traditional Virtual Machine
  - Need large storage space and more computing power



What is a Container [Digital image]. (n.d.). Retrieved from https://www.docker.com/what-container

Container-based Applications

- Lightweight
  - Quick start
  - Easy to replace the configuration of the applications



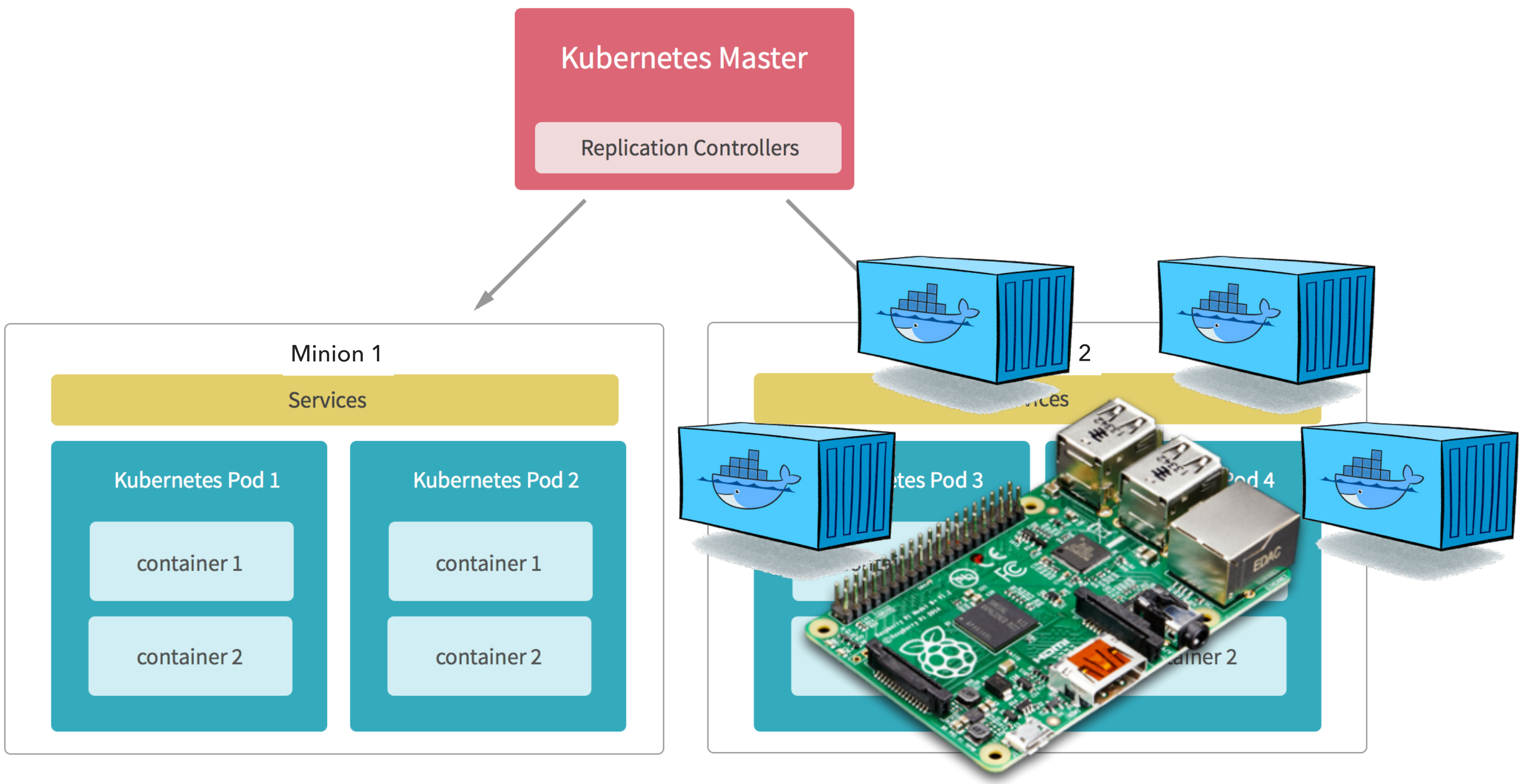
	Virtual Machine	Container
Size	GB	MB
Startup	Minute	Second

Orchestration Tools

- SaltStack
  - Remote execution tool and configuration management system
- OpenStack
  - Used to manage virtual machines in data centers
- Swarm
  - Native clustering system for Docker
- Kubernetes
  - Automating deployment, scaling, and management of containerized applications

Kubernetes Architecture

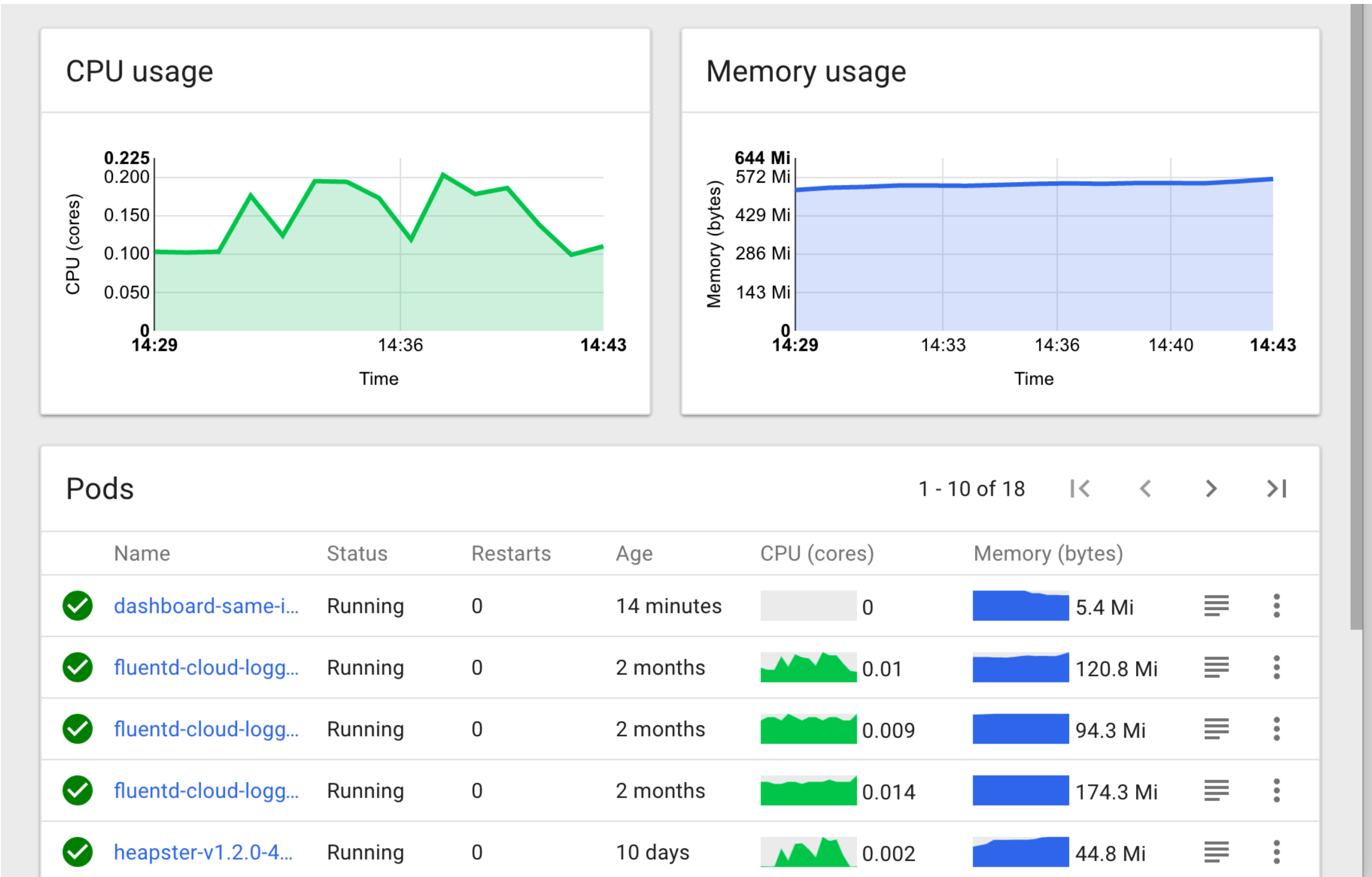
- Each fog devices hosts several containers, can be assembled into pod
- A service is a group of pods that are running on the cluster



Resource Monitoring

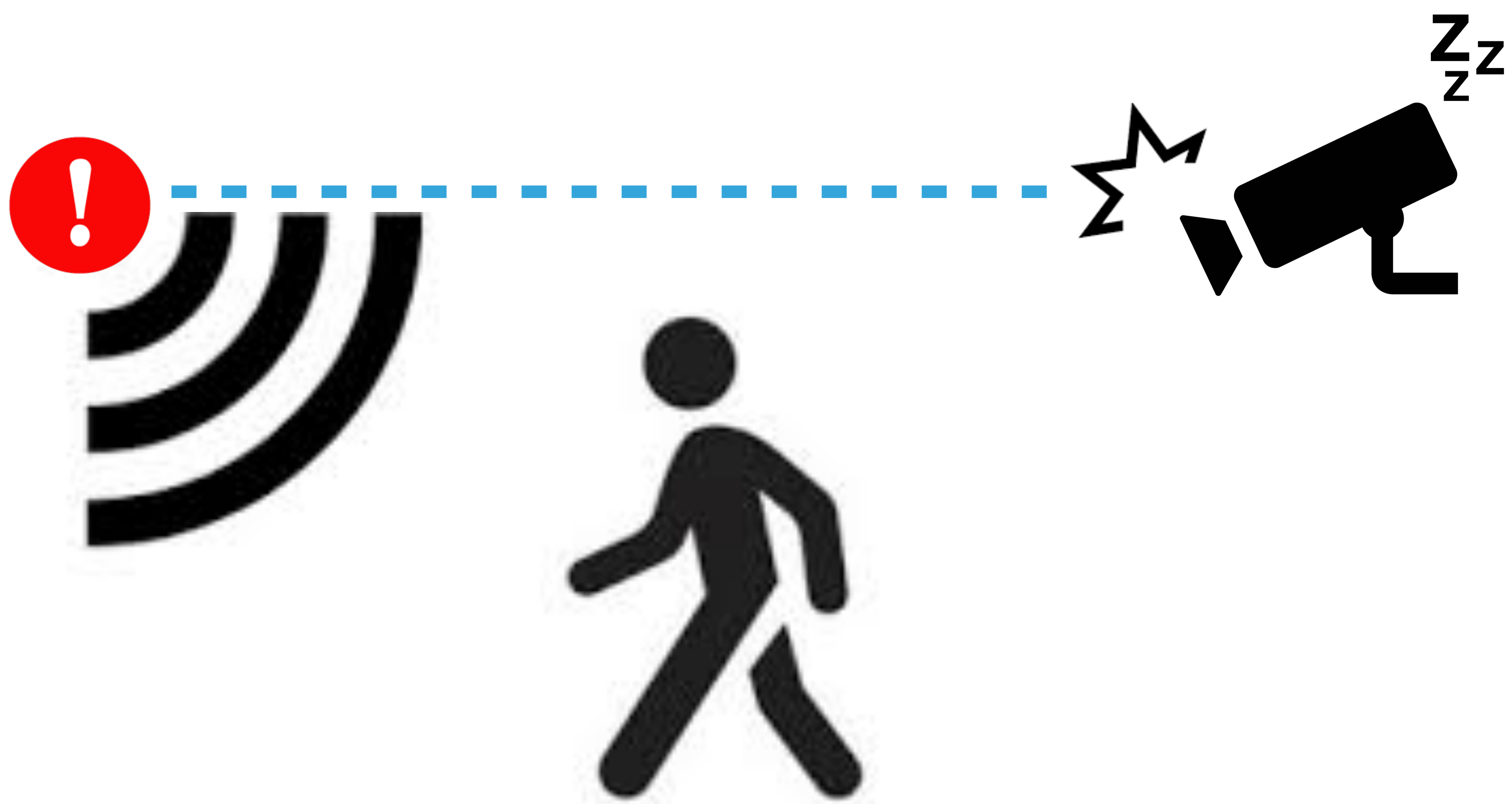
- Fog devices situation monitoring
  - Resource usage (CPU, Memory)
  - Containers status

Providing the important information for deployment strategy.



Event-driven Mechanism

- Allows developers to make the logical rules that automatically deploy another application when a specific event is triggered
  - Motion detected –> Capture an image

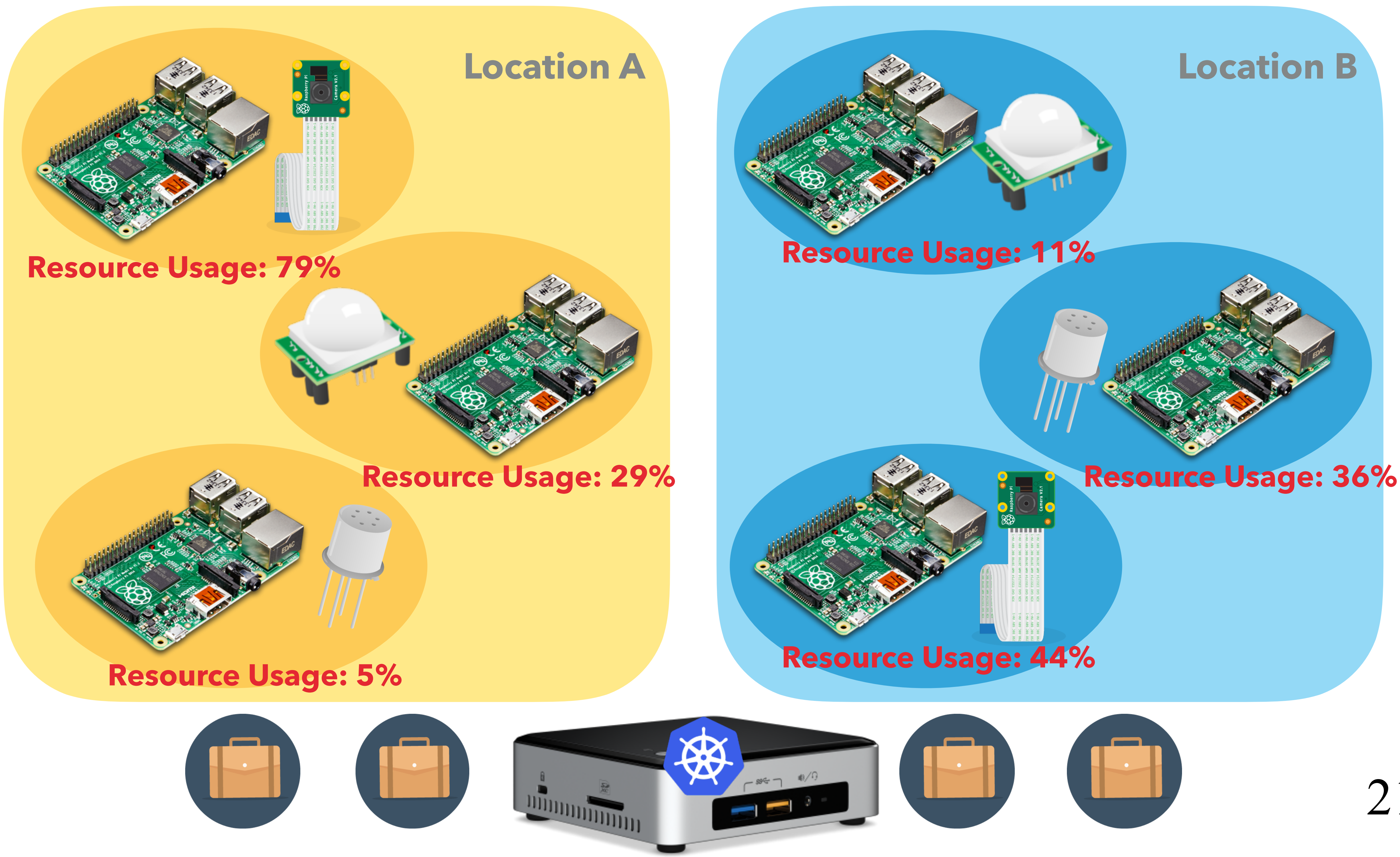


Edge Analytics

Requirement of Edge IoT Analytics

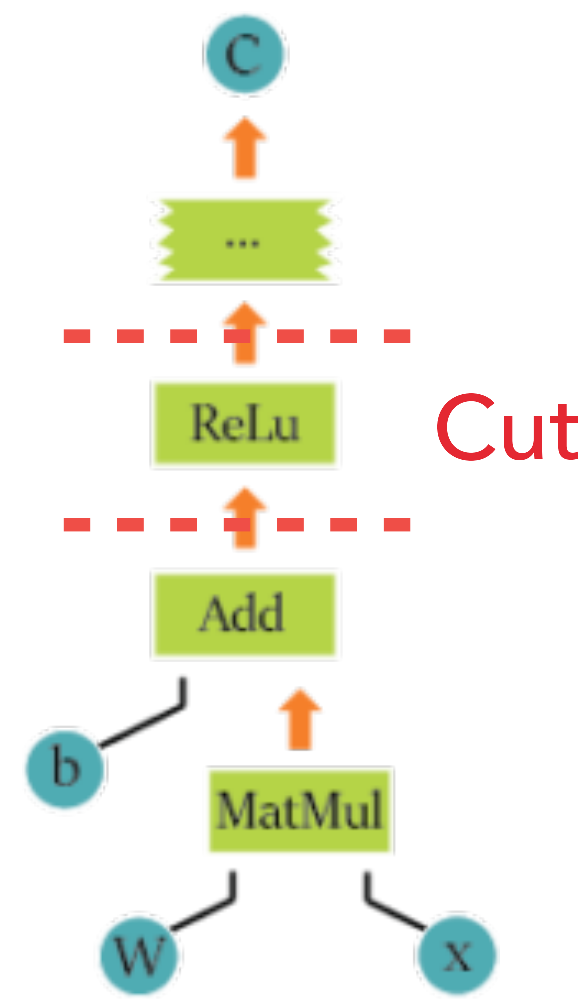
- Location-based and sensor-based services
    - Tag the devices
- Raw sensor data are huge
    - Deep learning
- Resource-constrained fog computing devices
    - Distributed computing

Tag the Devices



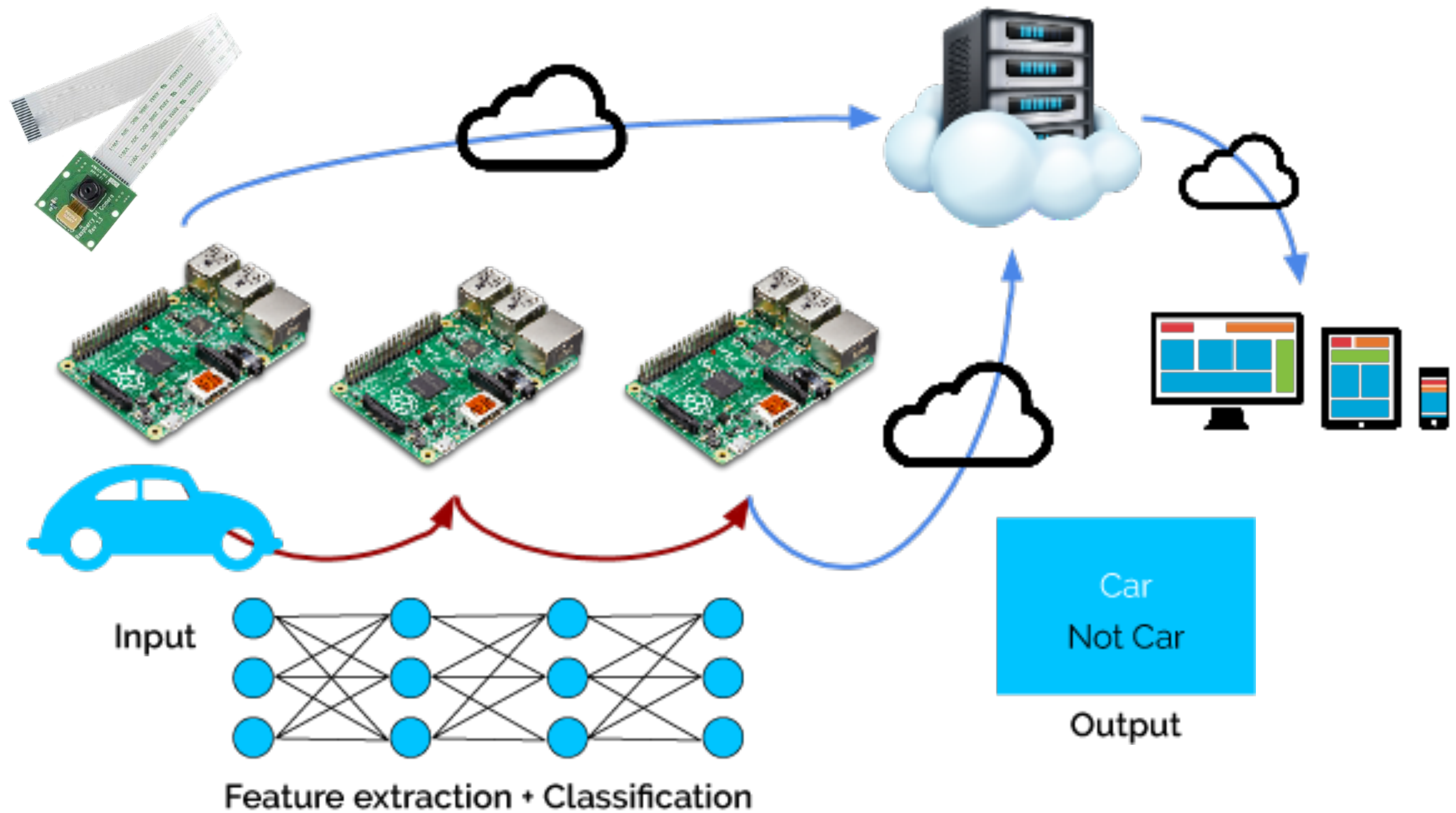
Pre-processing with Deep Learning

- Tensorflow
  - An open-source software library for Machine Intelligence
  - Data flow graphs
    - Nodes - mathematical operations
    - Edges - multidimensional data arrays (tensors)



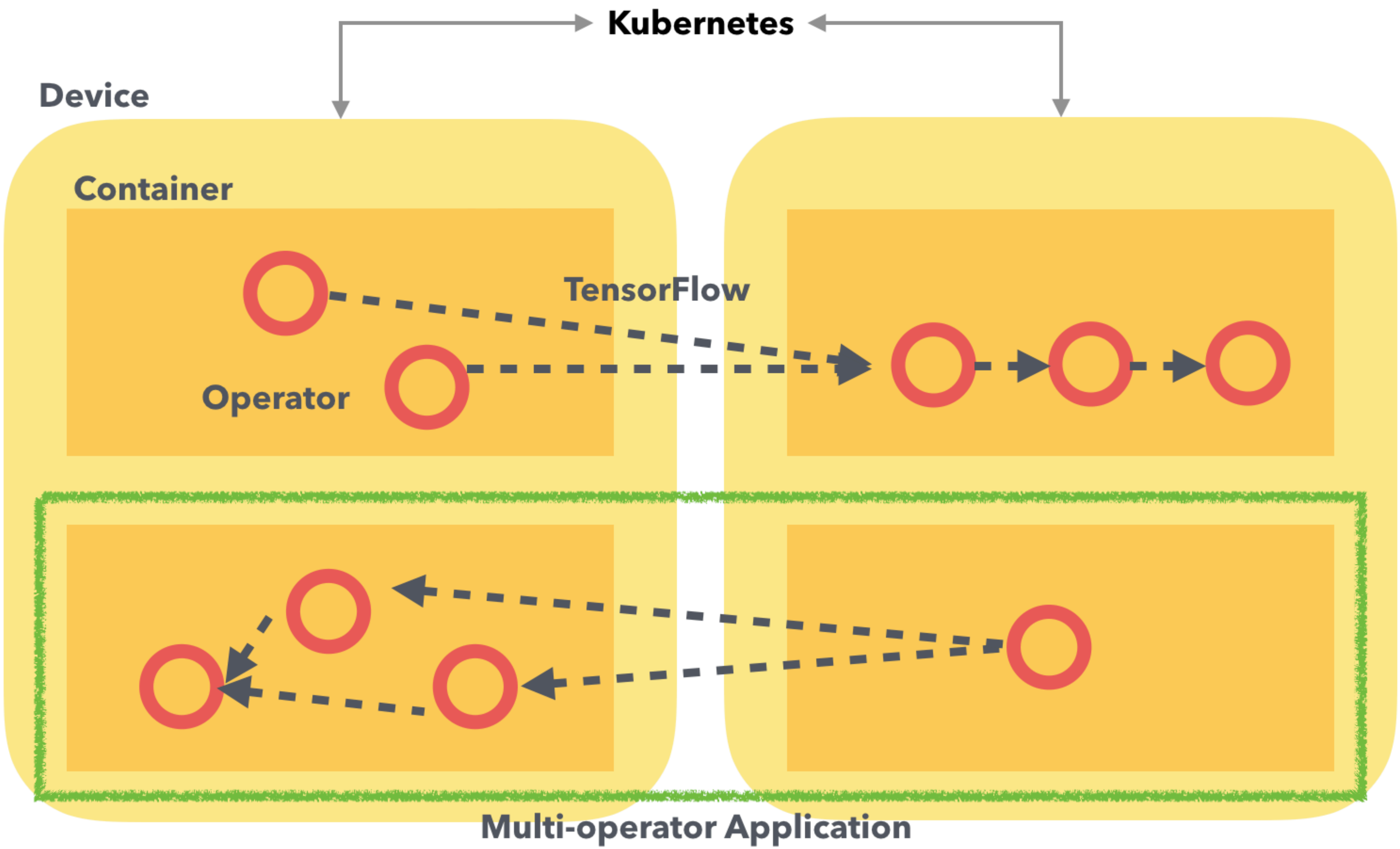
Distributed Computing

- Collecting the resource from several heterogeneous fog devices

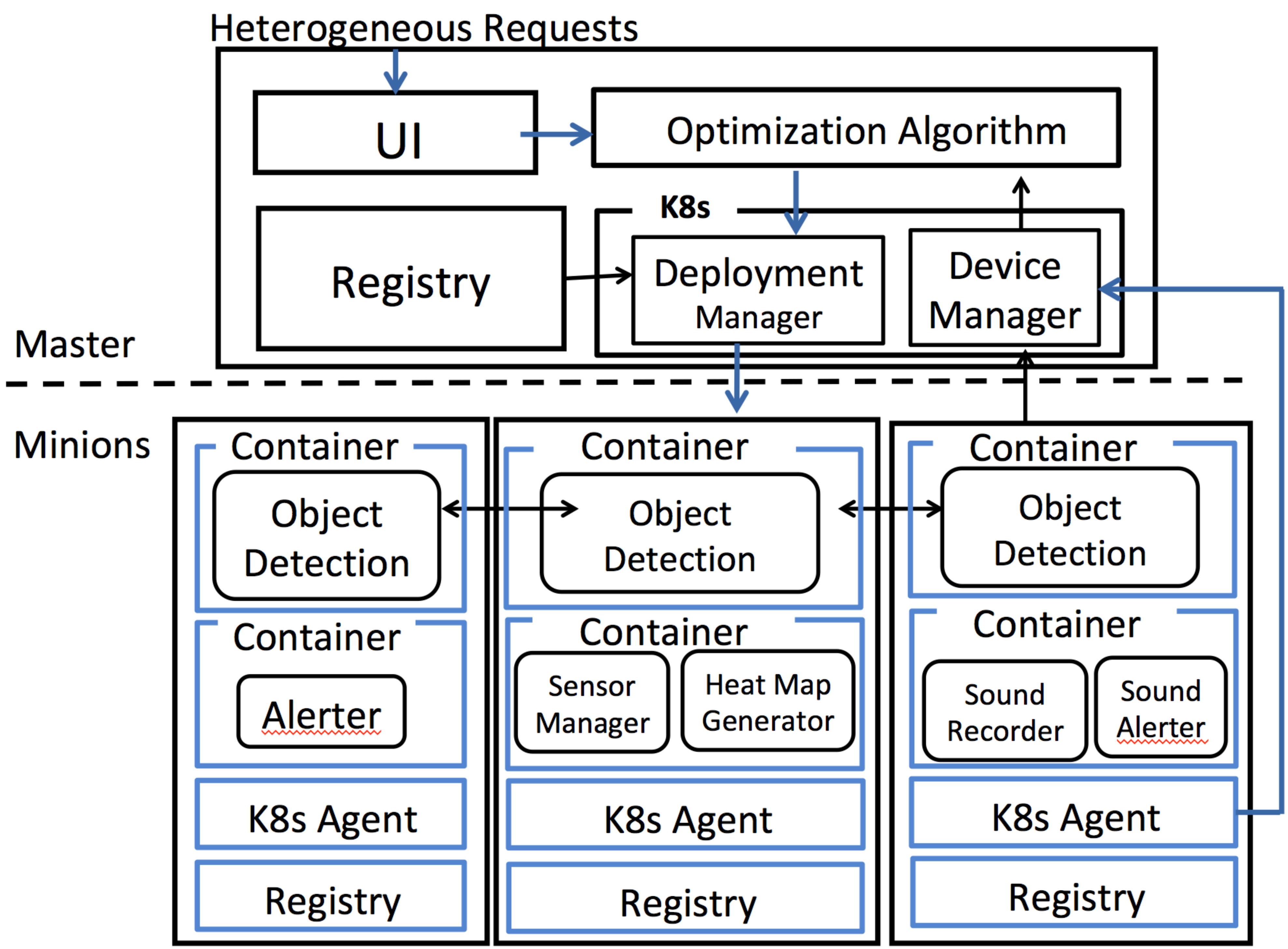


System Overview

Programming Model

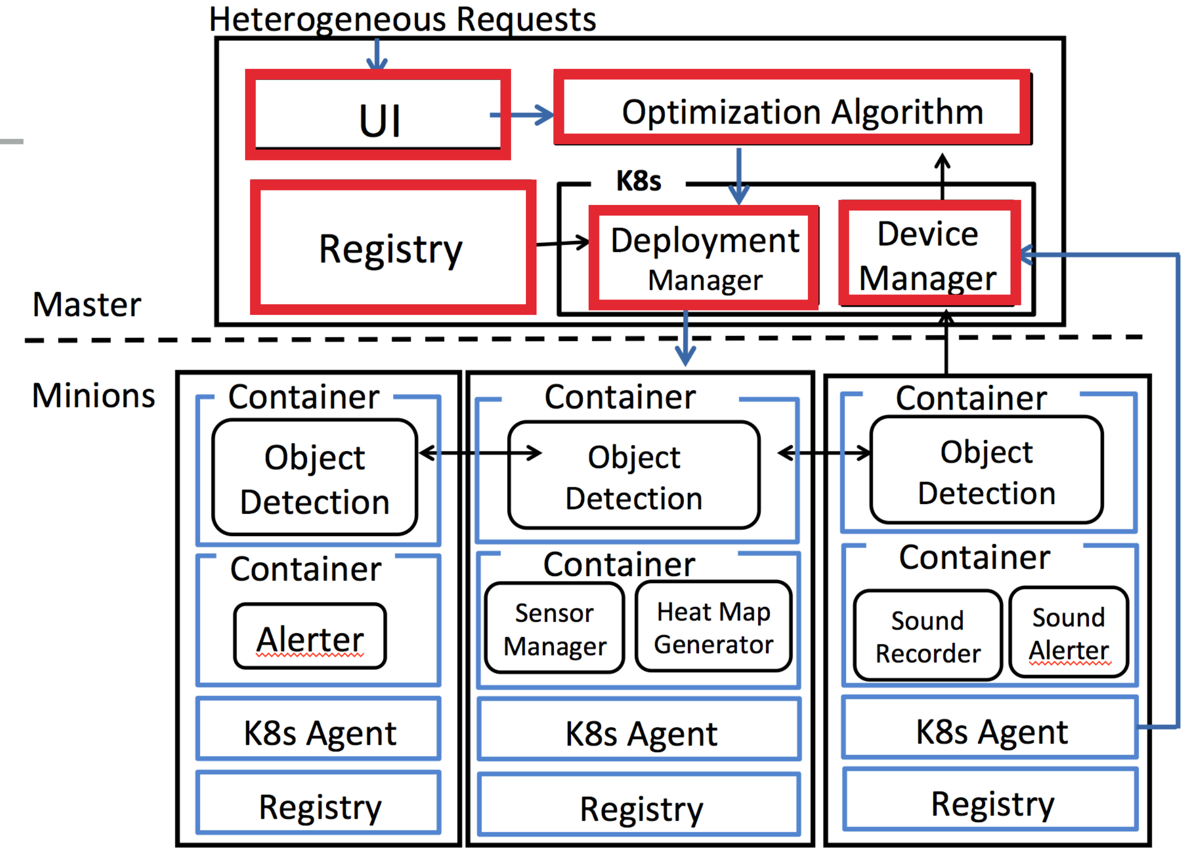


System Overview



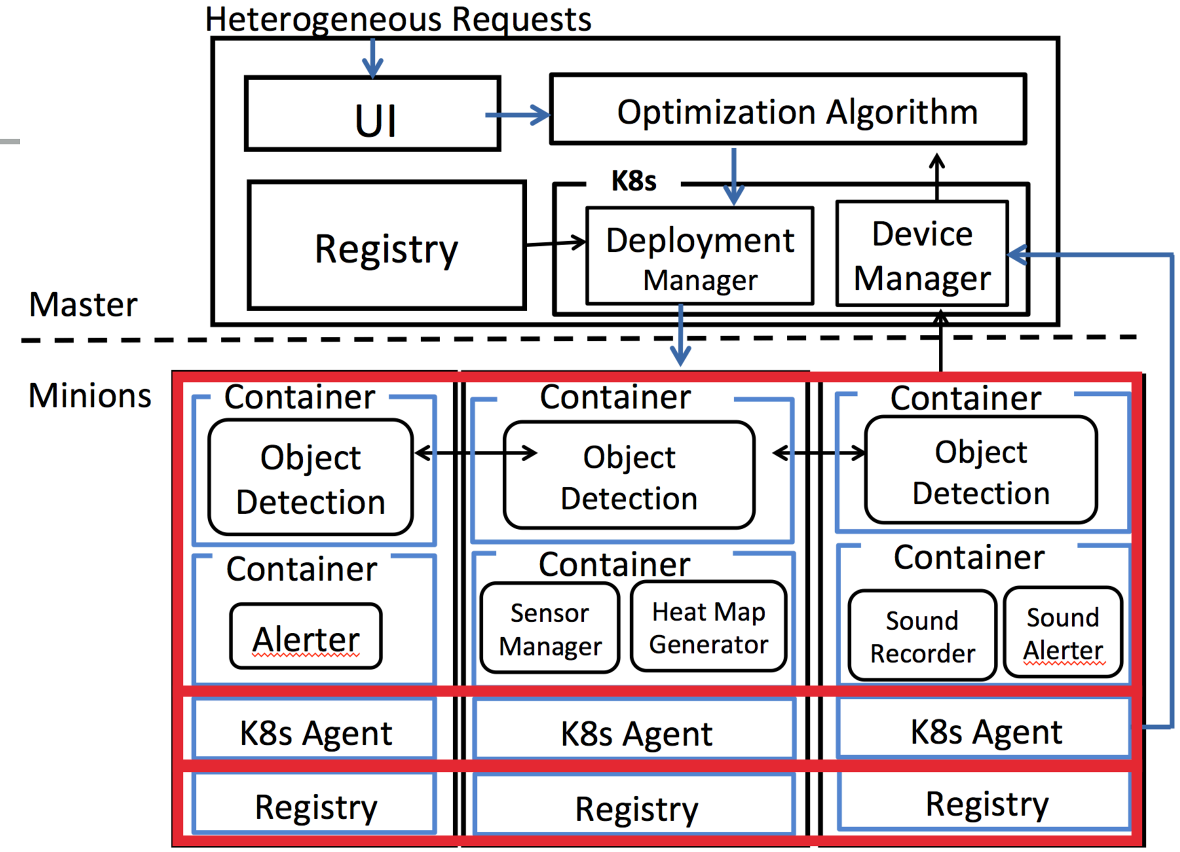
Master

- User Interface
- Operator deployment algorithm
  - Decide deploying which operators on which minions
- Device manager
  - Collect crucial device status
- Deployment manager
  - Launch specific Docker images on chosen minions
- Registry
  - Images are stored in the registry at the server

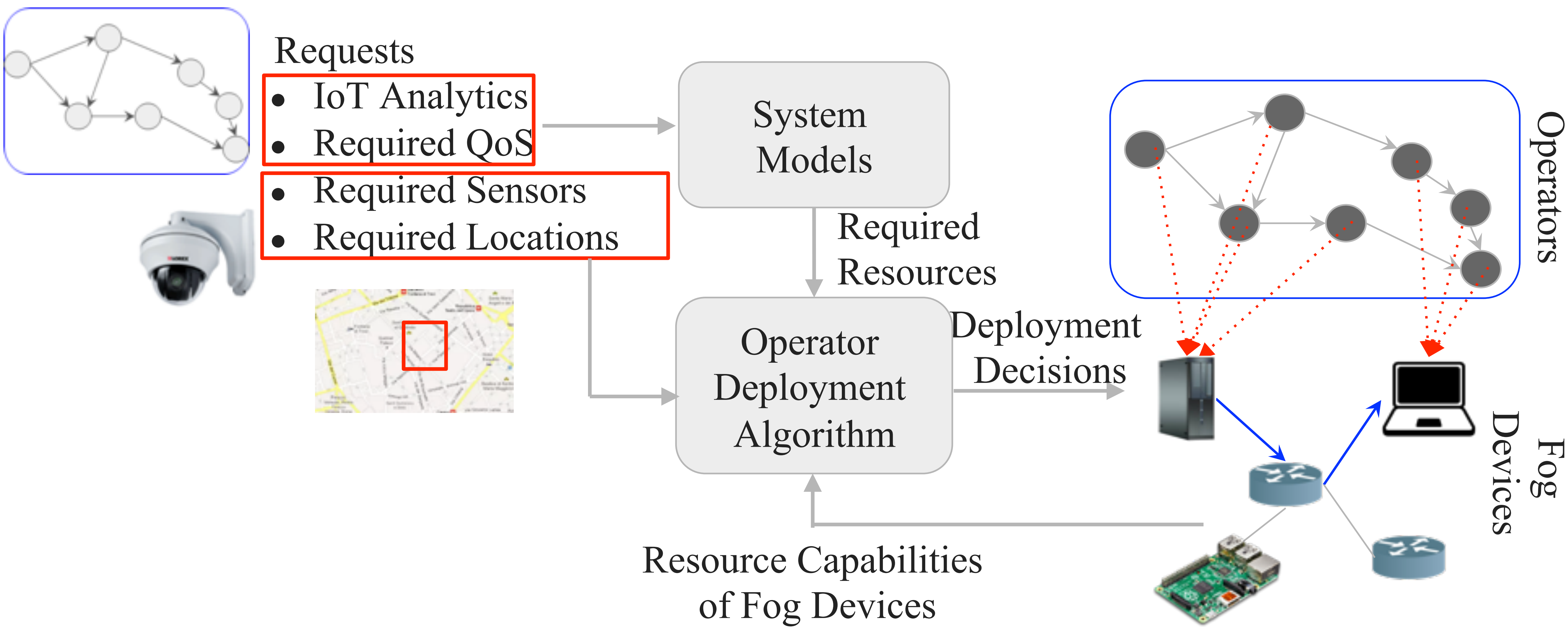


Minions

- TensorFlow-enabled container
  - Docker containers including TensorFlow and its analytic libraries
- k8s agent
  - Monitor and report the status of minions and pod to device manager
- Local Registry



Algorithm - System Model Derivation

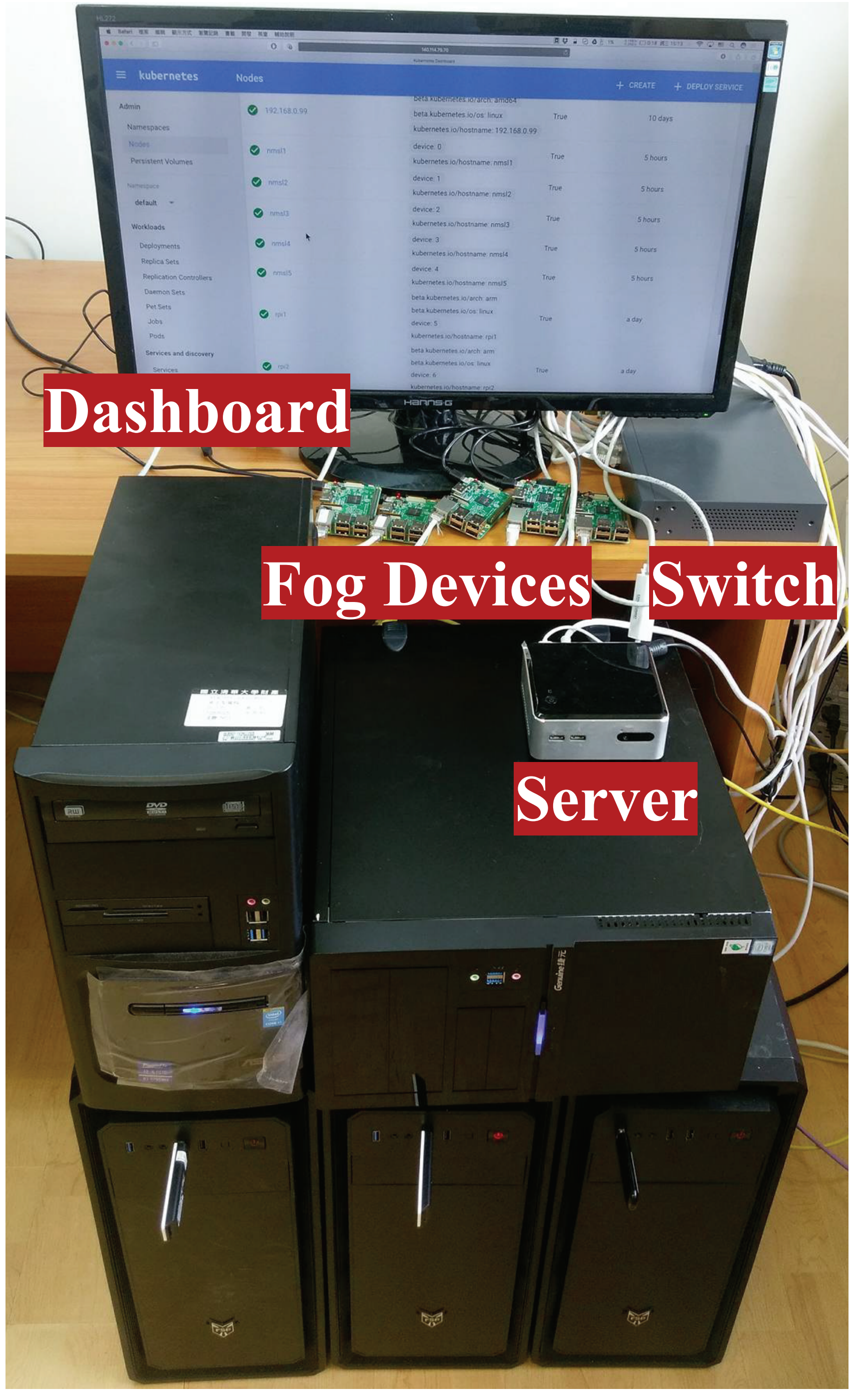


H. Hong, **P. Tsai**, A. Cheng, and C. Hsu, "Supporting Internet-of-Things Analytics in a Fog Computing Platform," in Proc. of IEEE International Conference on Cloud Computing Technology and Science (CloudCom'17), Hong Kong, China, December 2017.

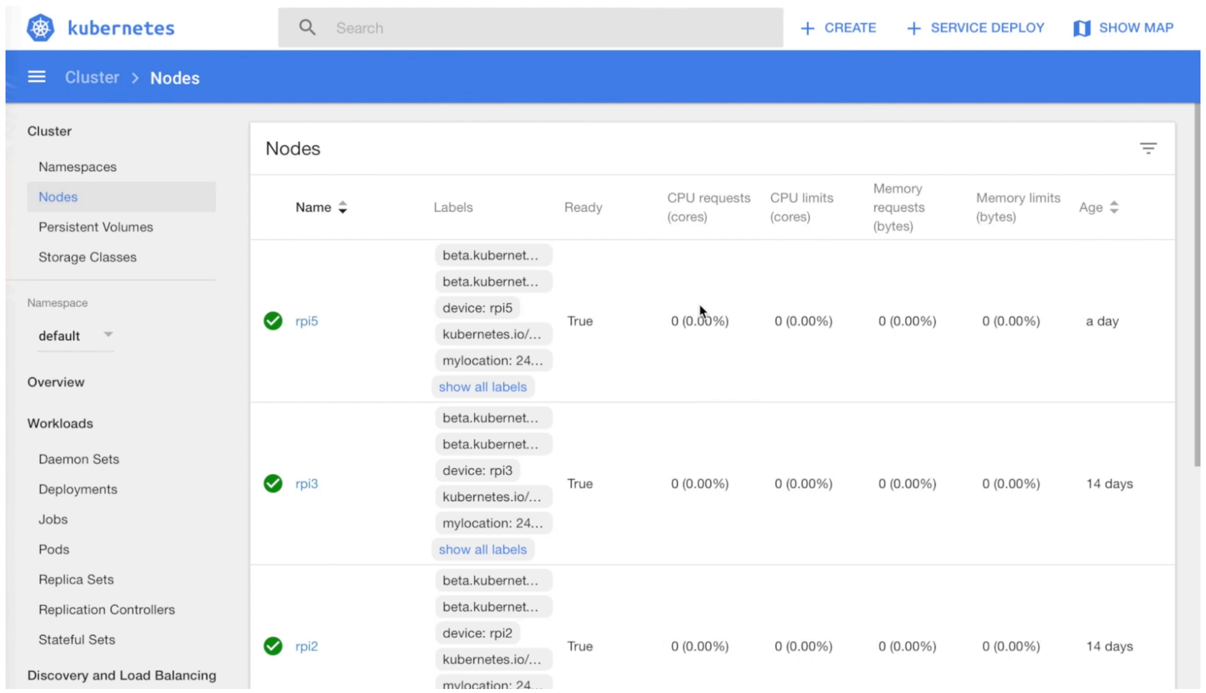
Implementations and Demo Scenarios

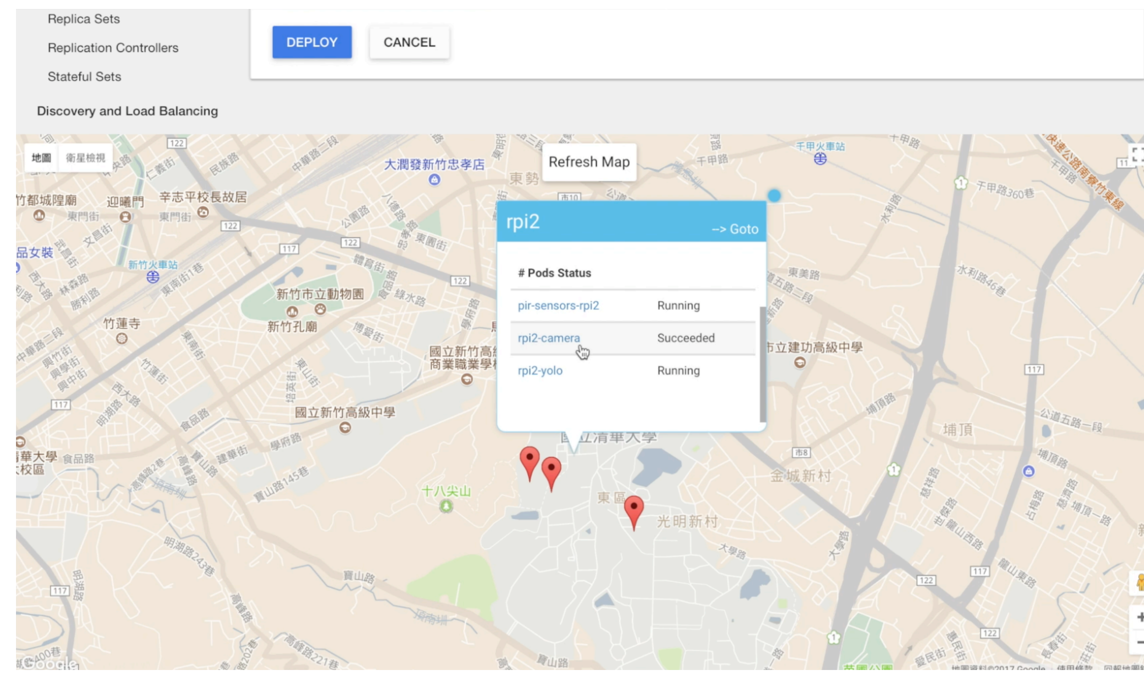
Fog Computing Platform

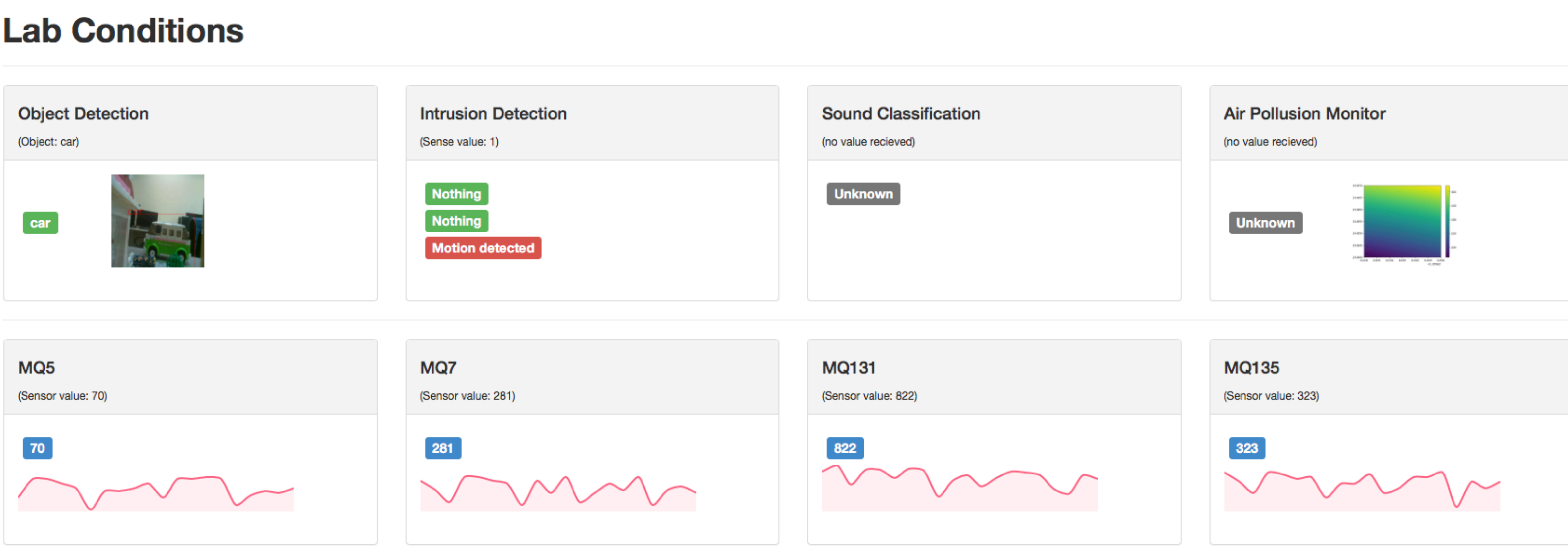
- Auto-deployment
  - Docker (container virtualization)
  - Kubernetes (deployment, resource monitoring)
  - Deployment Algorithm
- IoT edge analytics
  - TensorFlow



User Interfaces







Experiment Setup

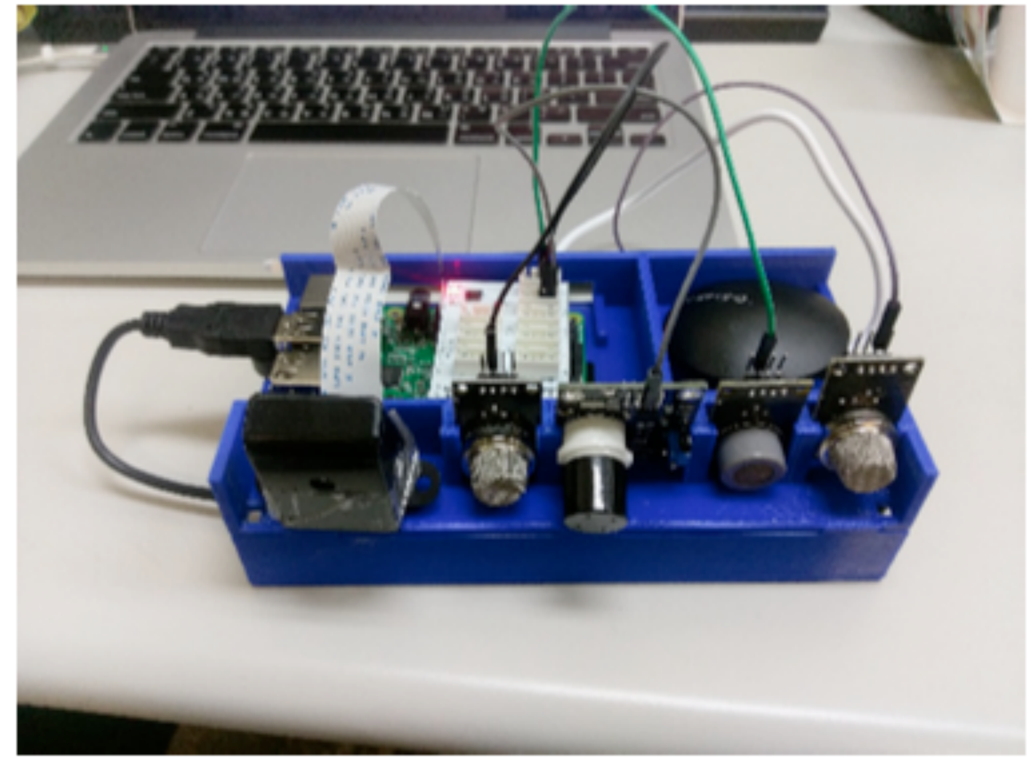
- Master
  - i5 CPU PC installed with Kubernetes
- Fog Devices
  - 5 Intel PC (1.8 GHz 8-core i7 CPUs)
  - 5 Raspberry Pi (1.2 GHz 4-core ARM CPUs)
- Bandwidth throttle: Wonder Shaper
  - 8 Mbps (close to common WiFi bandwidth)

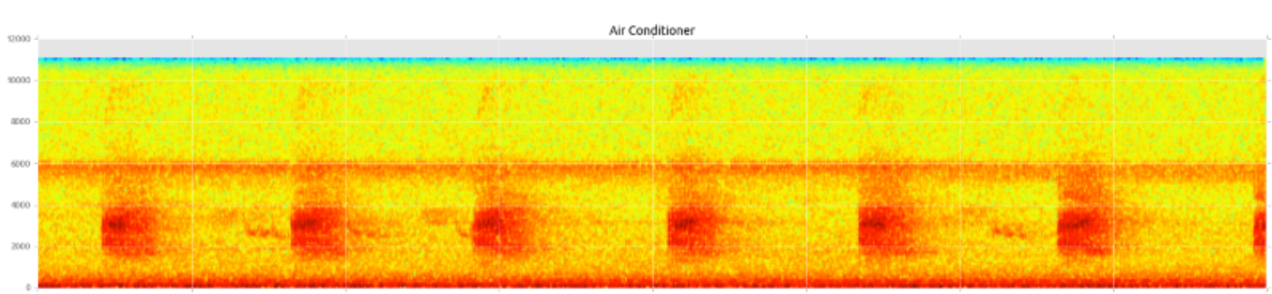
Experiment Setup

- Run applications using multiple threads to fully utilize the fog devices
- Add a queue between any two adjacent operators to increase the overall performance
- Run each experiment 5 times and present the average results

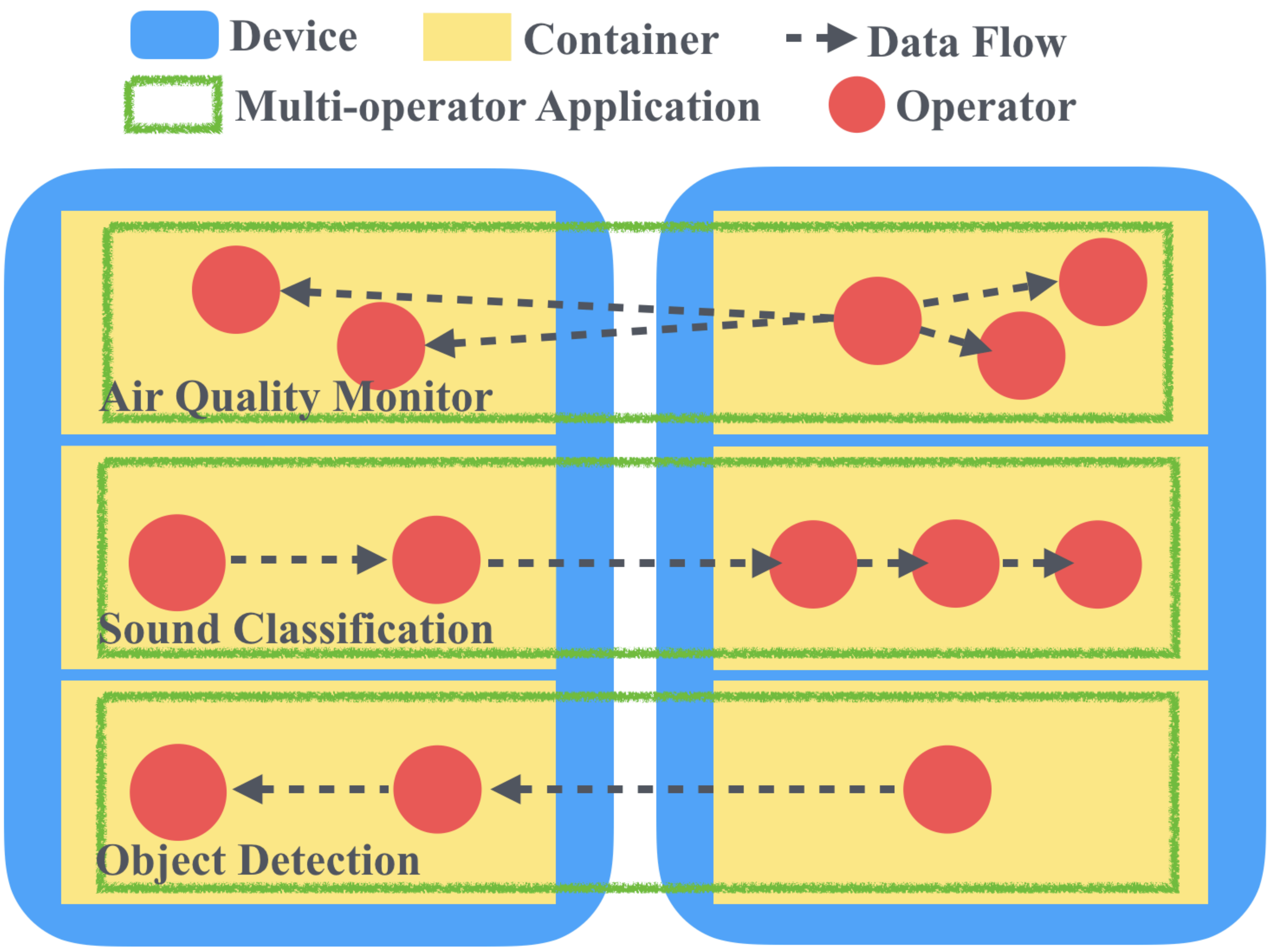
3 Applications

- Air quality monitor
- Sound classification
- Object detection

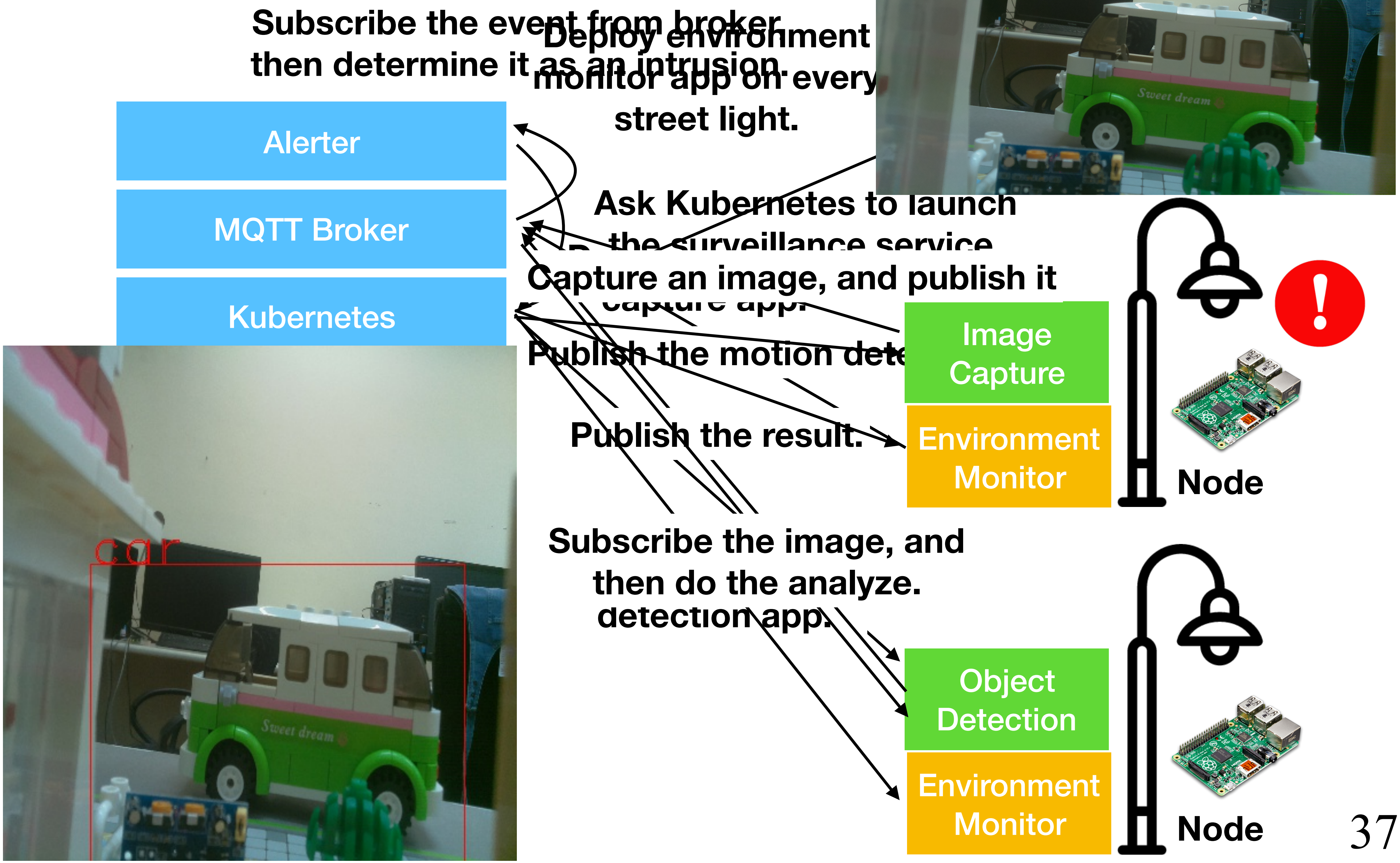




3 Applications



Scenario

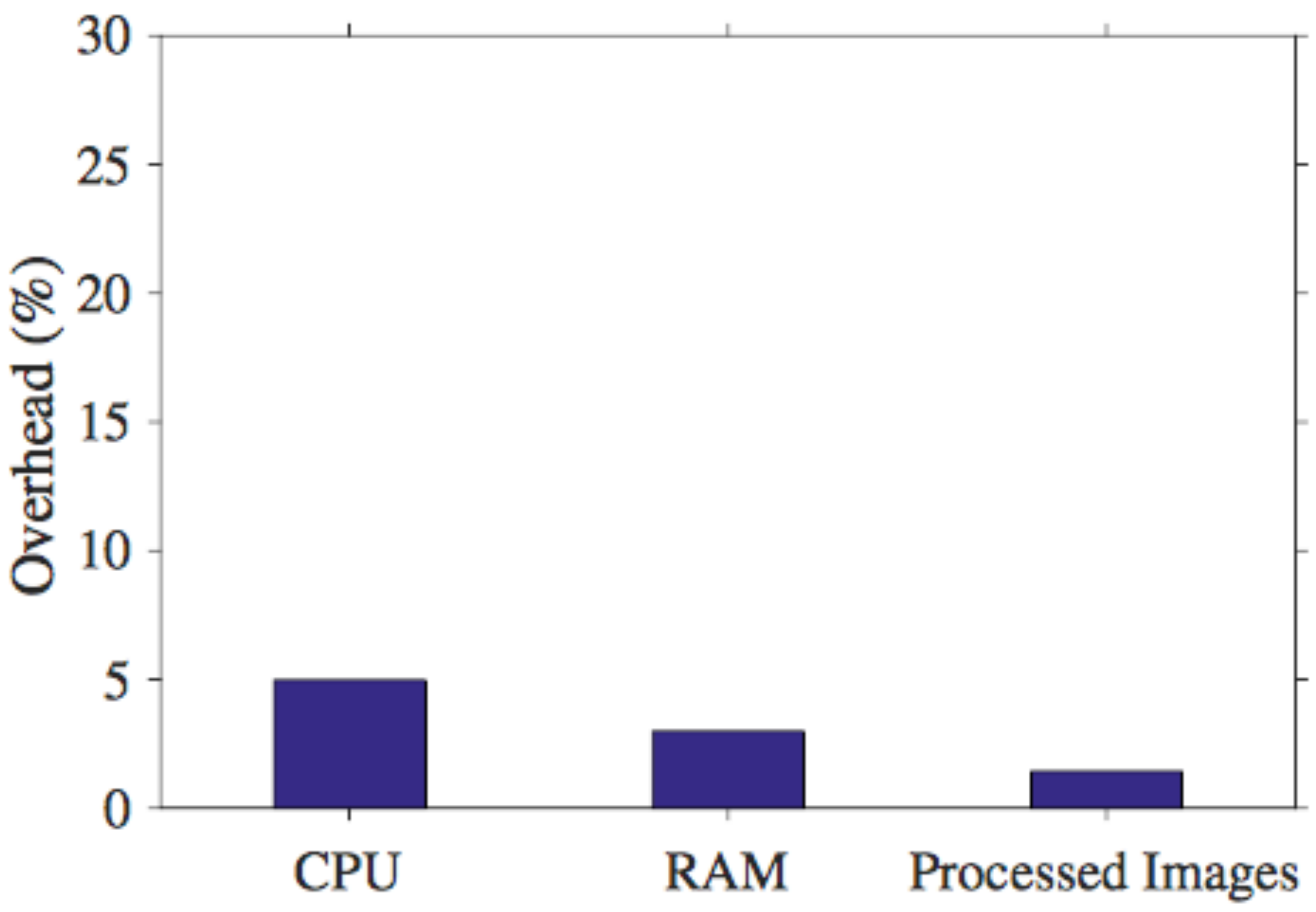

Subscribe the event from broker
then determine it as an intrusion.
Deploy environment
monitor app on every
street light.
Alerter
MQTT Broker
Kubernetes
Ask Kubernetes to launch
the surveillance service
Capture an image, and publish it
Publish the result.
Image
Capture
Environment
Monitor
Node
car
Subscribe the image, and
then do the analyze.
detection app.
Object
Detection
Environment
Monitor
Node

Evaluations

Dynamic Deployment

Container Overhead

- Setup: with and without Docker
  - Overhead caused by Docker virtualization on Raspberry Pi less than 5%



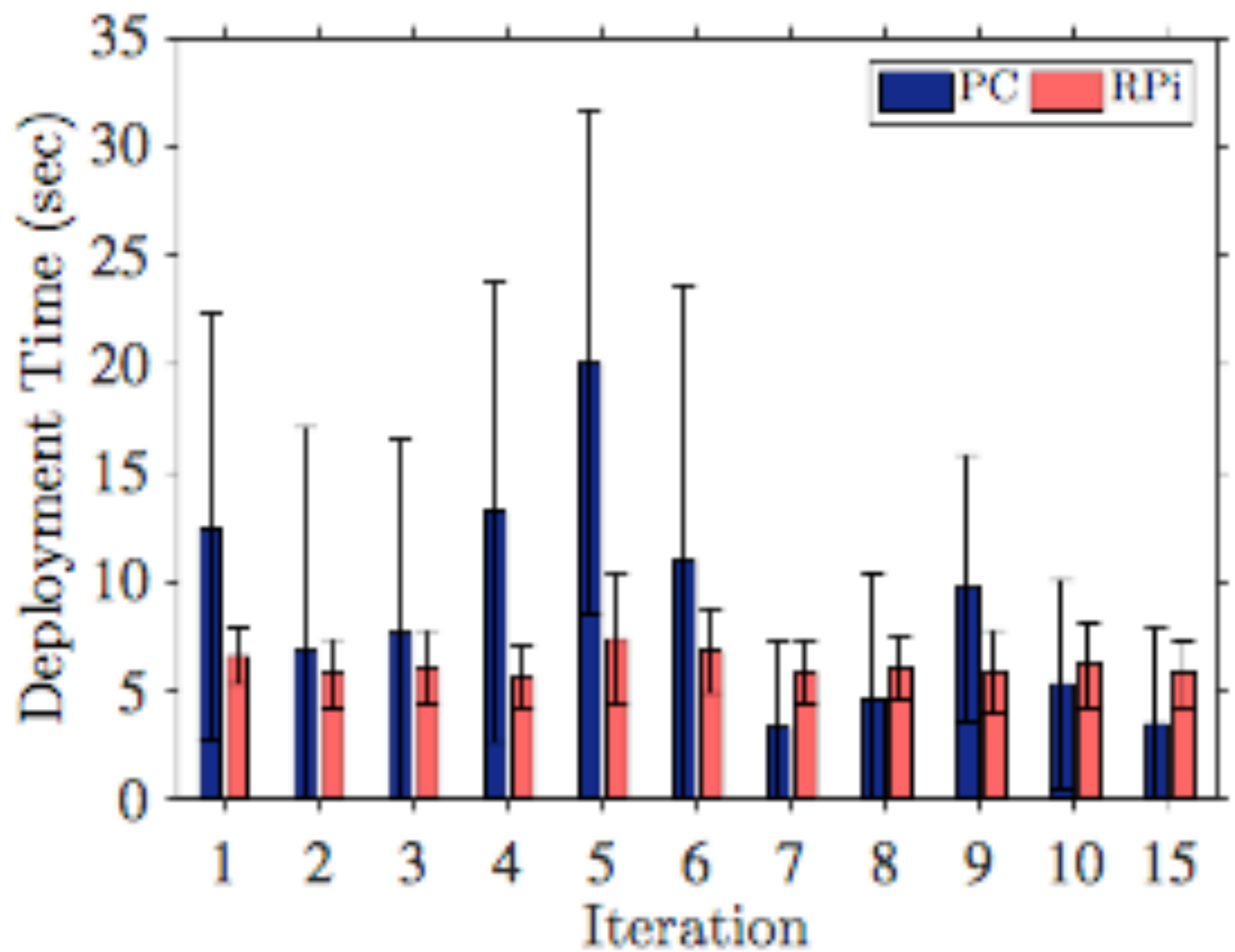
Efficiency of Deployment - Algorithm

- Update the system models online to iteratively derive the customized system models
- Run the experiments for 15 iterations, and update our system models after each iteration
- Generate a large number of requests, and execute our algorithm to deploy as many requests as possible

H. Hong, **P. Tsai**, A. Cheng, and C. Hsu, "Supporting Internet-of-Things Analytics in a Fog Computing Platform," in Proc. of IEEE International Conference on Cloud Computing Technology and Science (CloudCom'17), Hong Kong, China, December 2017.

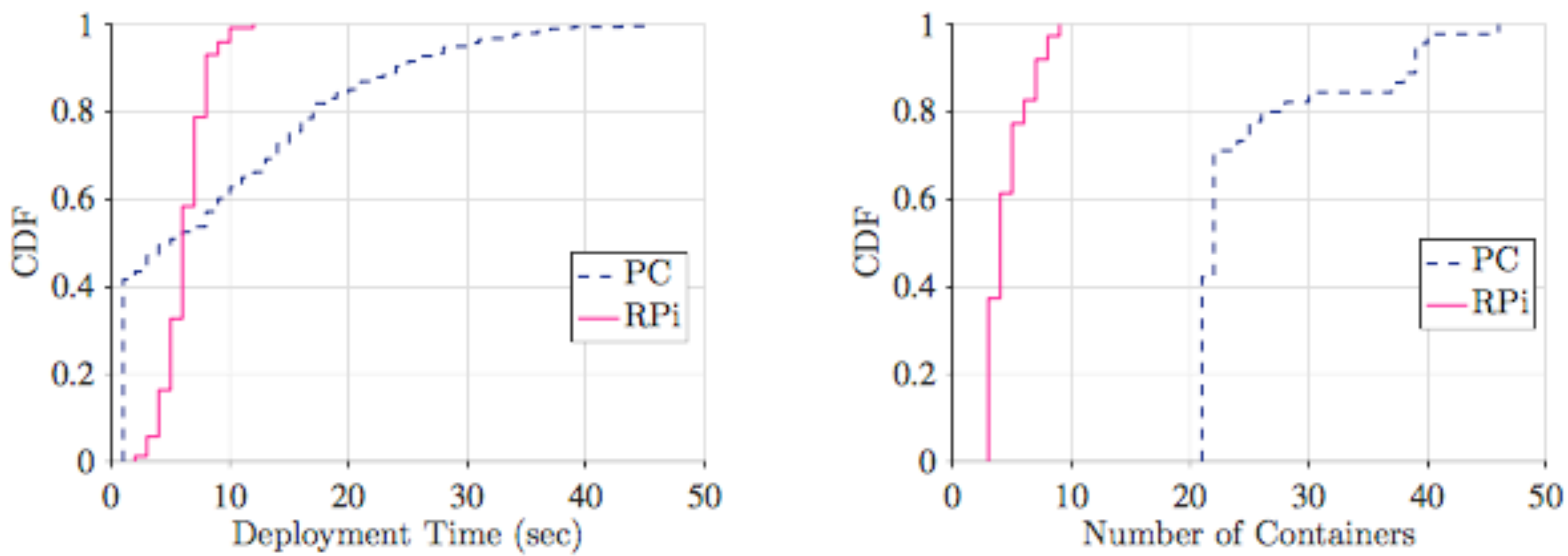
Efficiency of Deployment

- Deployed almost instantly in our platform
    - Operators take less than 20 seconds on average to be deployed



Efficiency of Deployment

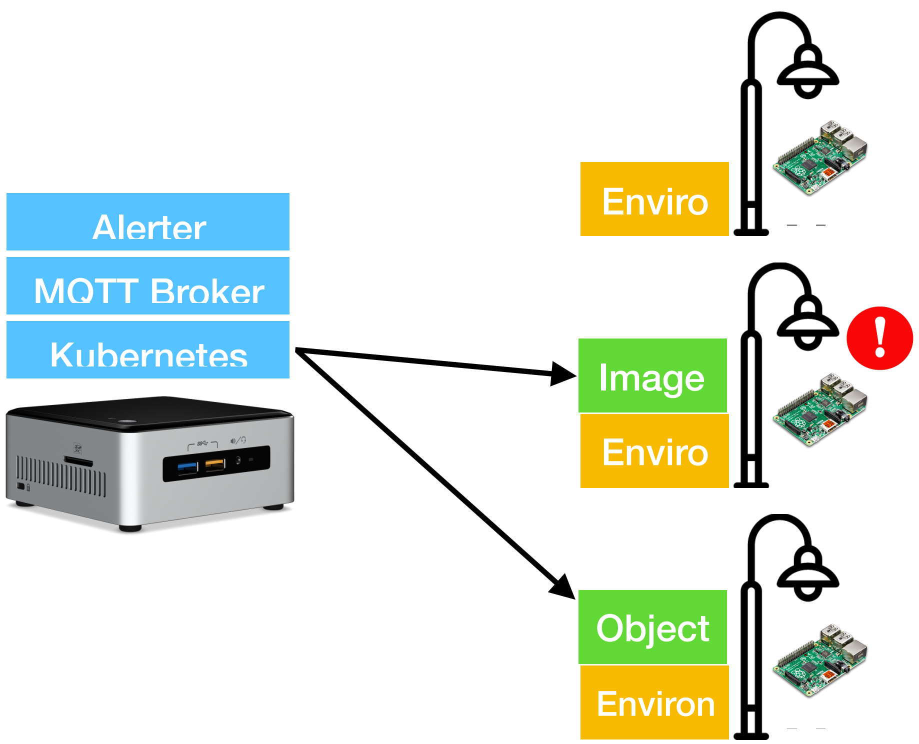
- When a large number of containers are simultaneously created on the same device, the I/O overhead is significantly increased



Event-driven in Short Time

- Need 32.9 secs to finish the whole object detection scenario
  - Only need 4.8 secs to trigger the new application

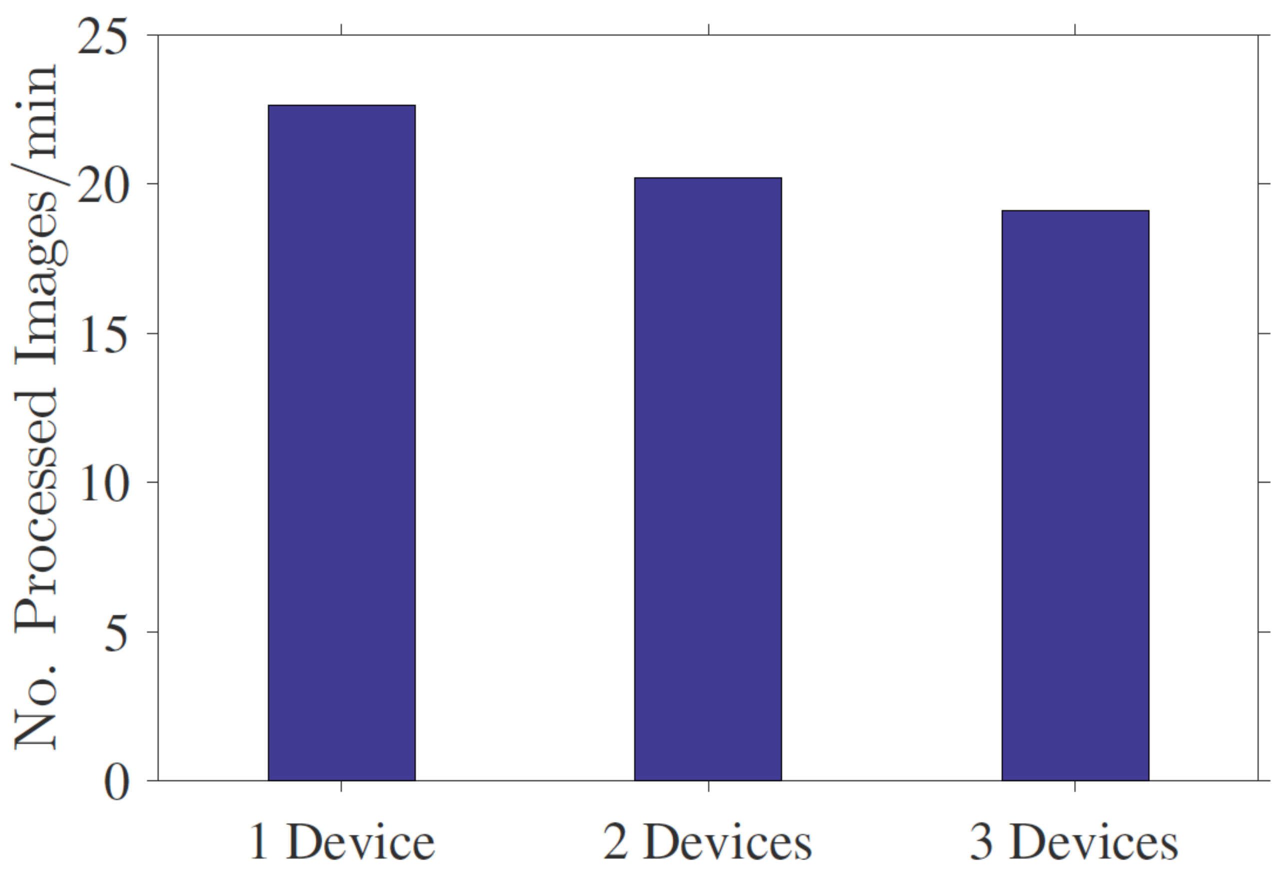
(scale box) PIR sensor send the value to MQTT	0
(monitor) Define it as a intrusion	0.0076
(monitor) Ask master deploy surveillance applicaion	0.0108
(master) Start the yolo container (object detection)	4.3236
(master) Start the capture container (capture)	4.8292
(capture) Capture the image	4.9862
(capture) Publish the image to MQTT broker	7.7692
(yolo) Start to subscribe the image from broker	8.0892
(yolo) Receive the image and start to analyze	18.6576
(yolo) Finish the analyze	32.9148
(master) Get the detect result	32.9228



Edge Analytics

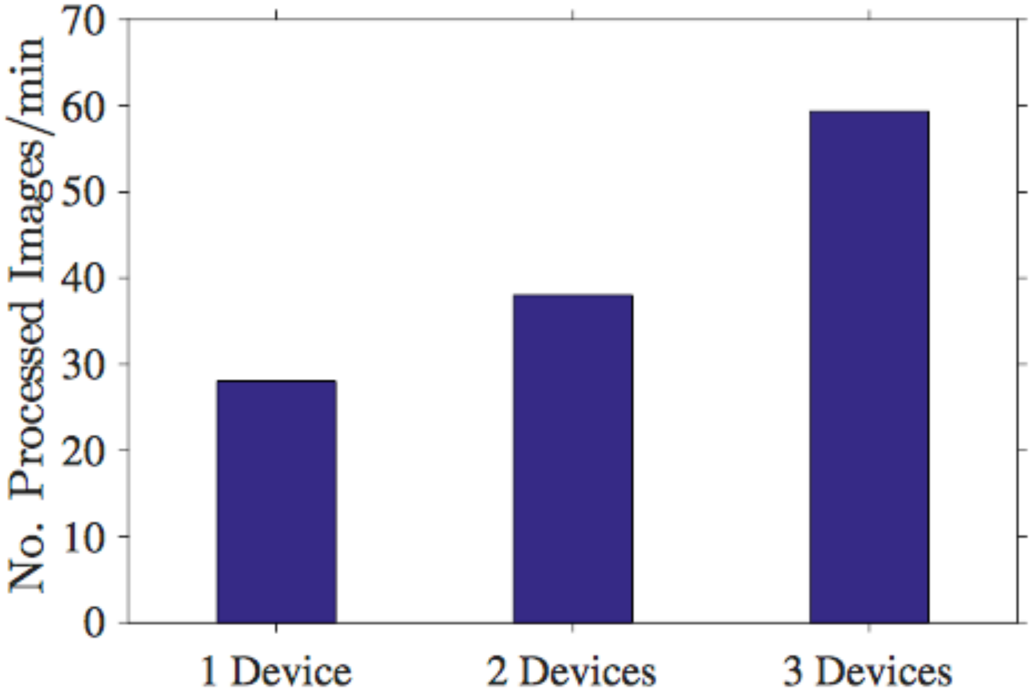
TensorFlow Achieves Low Collaboration Overhead

- Setup: run object detection without threading and container
  - Overhead adding one more device leads to only up to 10% overhead.



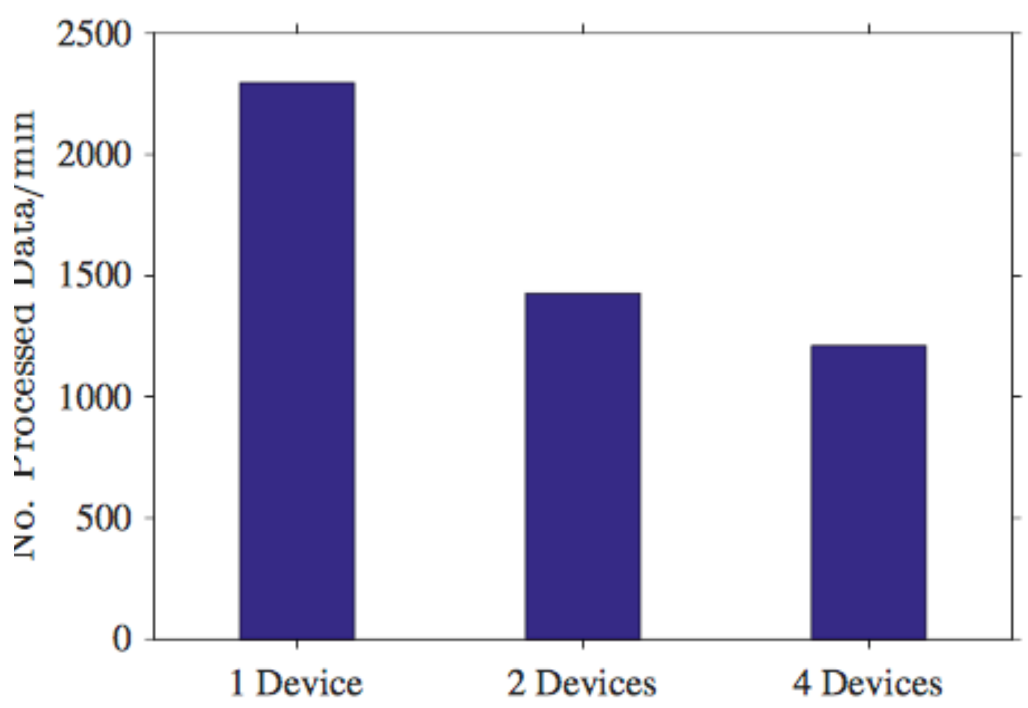
Benefits from Distributed Analytics

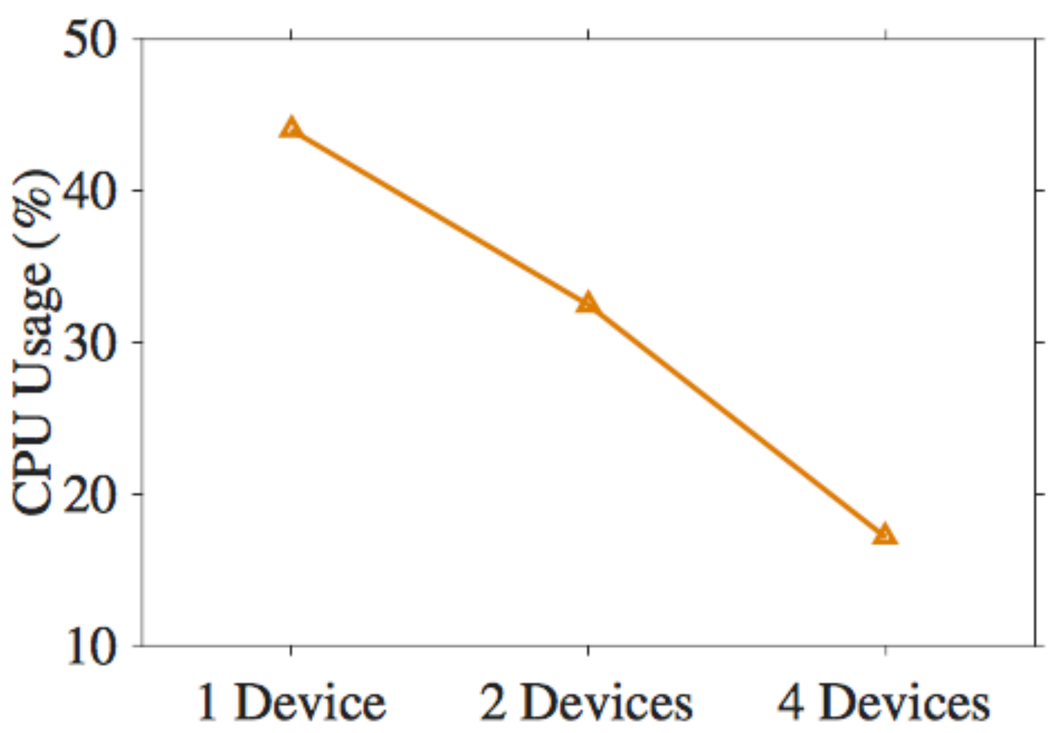
- Setup: run object recognizer on two fog devices (different cutting points)
- For heavy analytics applications (Object detection), distributed analytics results in large improvements

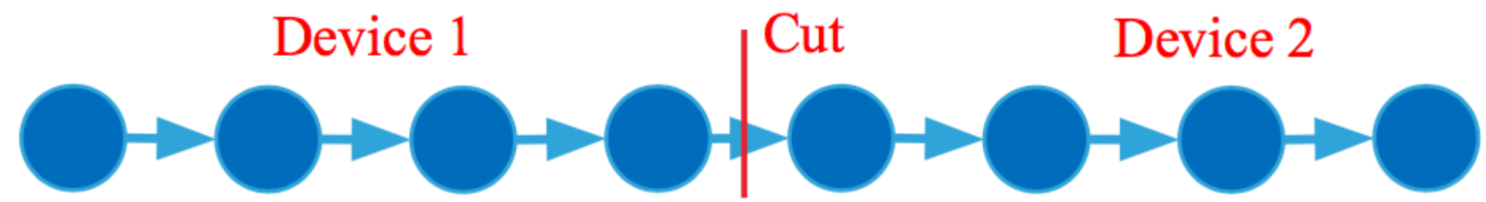


Benefits from Distributed Analytics

- Does not results in better performance when application's analytics is quite simple (Air Pollution)

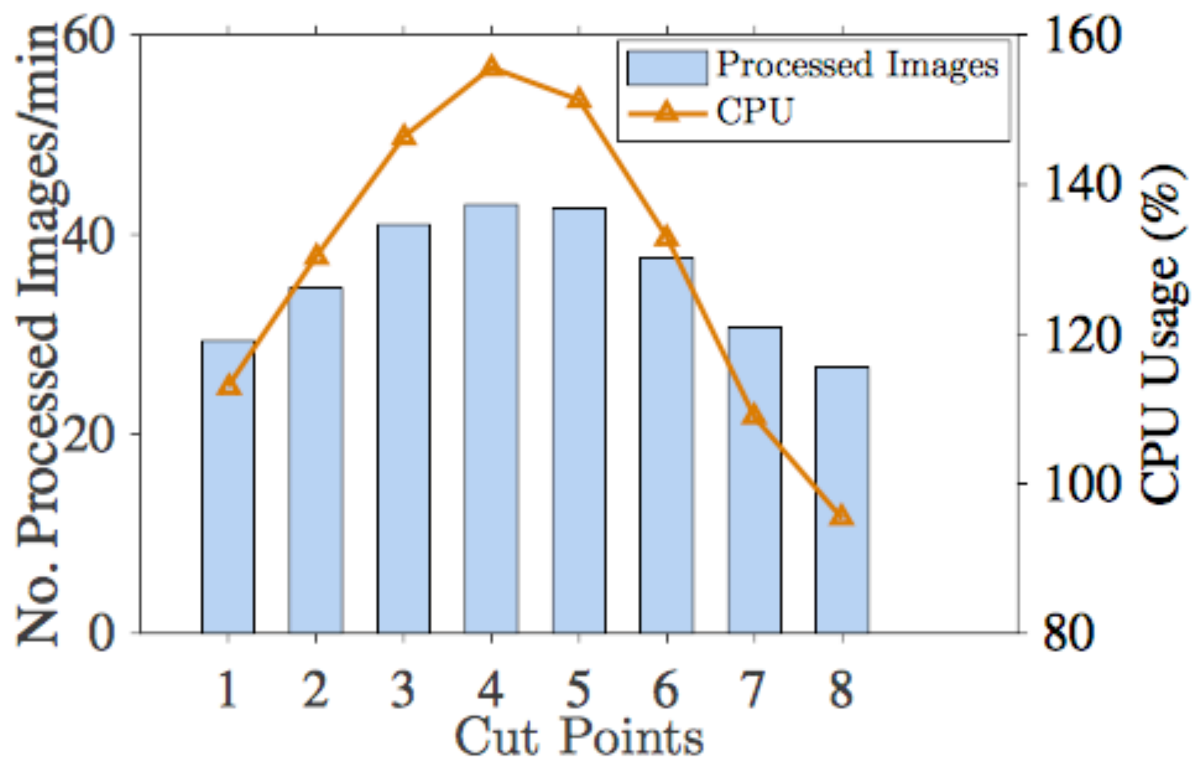


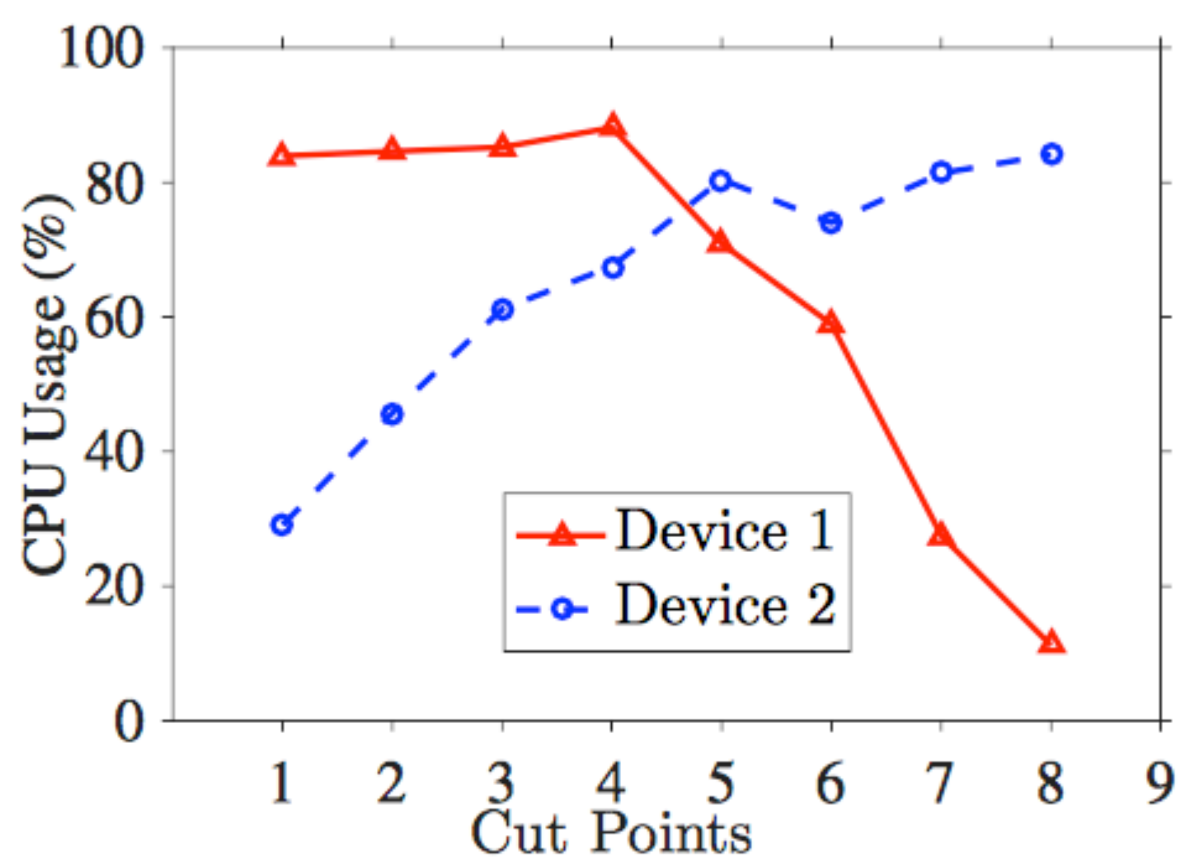


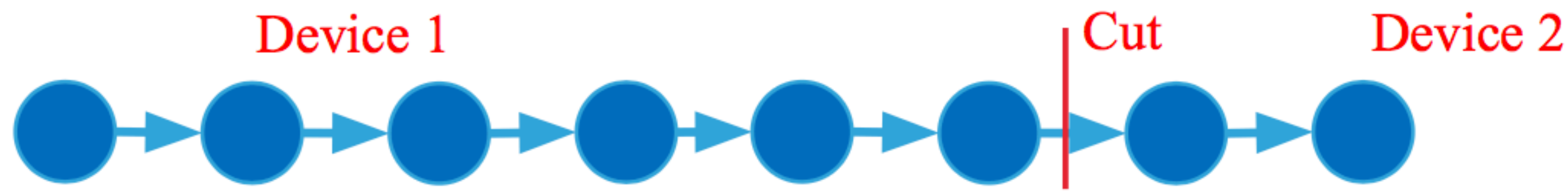


Different Service Quality Caused By Different Cutting Points

- Setup: 8 different cut points for object detection app.
  - Cutting into smaller operators with equal complexity results in the best performance

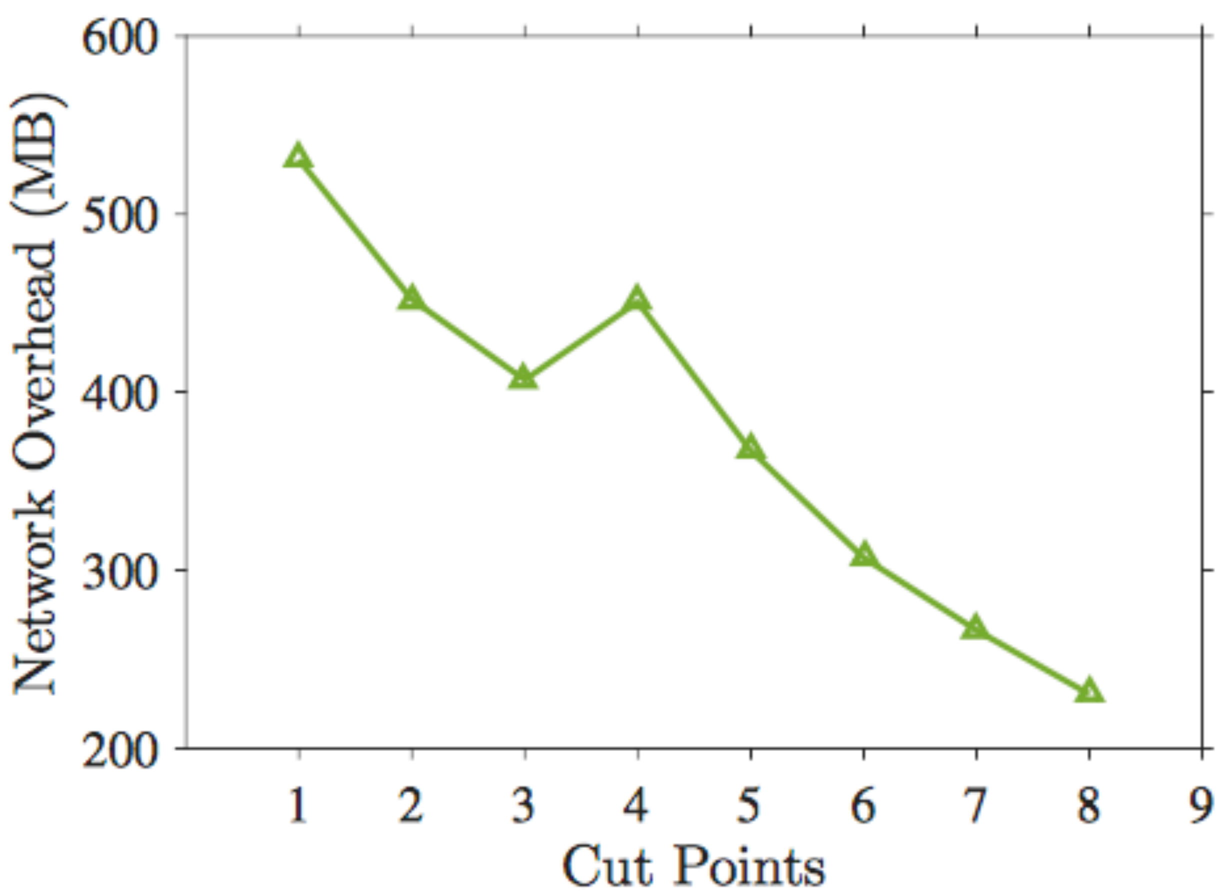






Different Service Quality Caused By Different Cutting Points

- When network resources is the bottleneck, we may not prefer equally-loaded splitting decisions



Related Works

Related Works

	Support Heterogeneous Devices	Dynamic Deployment	Event-trigger	Pre-processing	Deep Learning	Distributed Computing
Our Platform	v	v	v	v	v	v
AWS Greengrass	v		v	v	v	
IBM Watson	v	v	v	v	v	
Azure IoT Suite	v		v	v	v	
AT&T IoT Platform	v		v	v	v	

Conclusion and Future Work

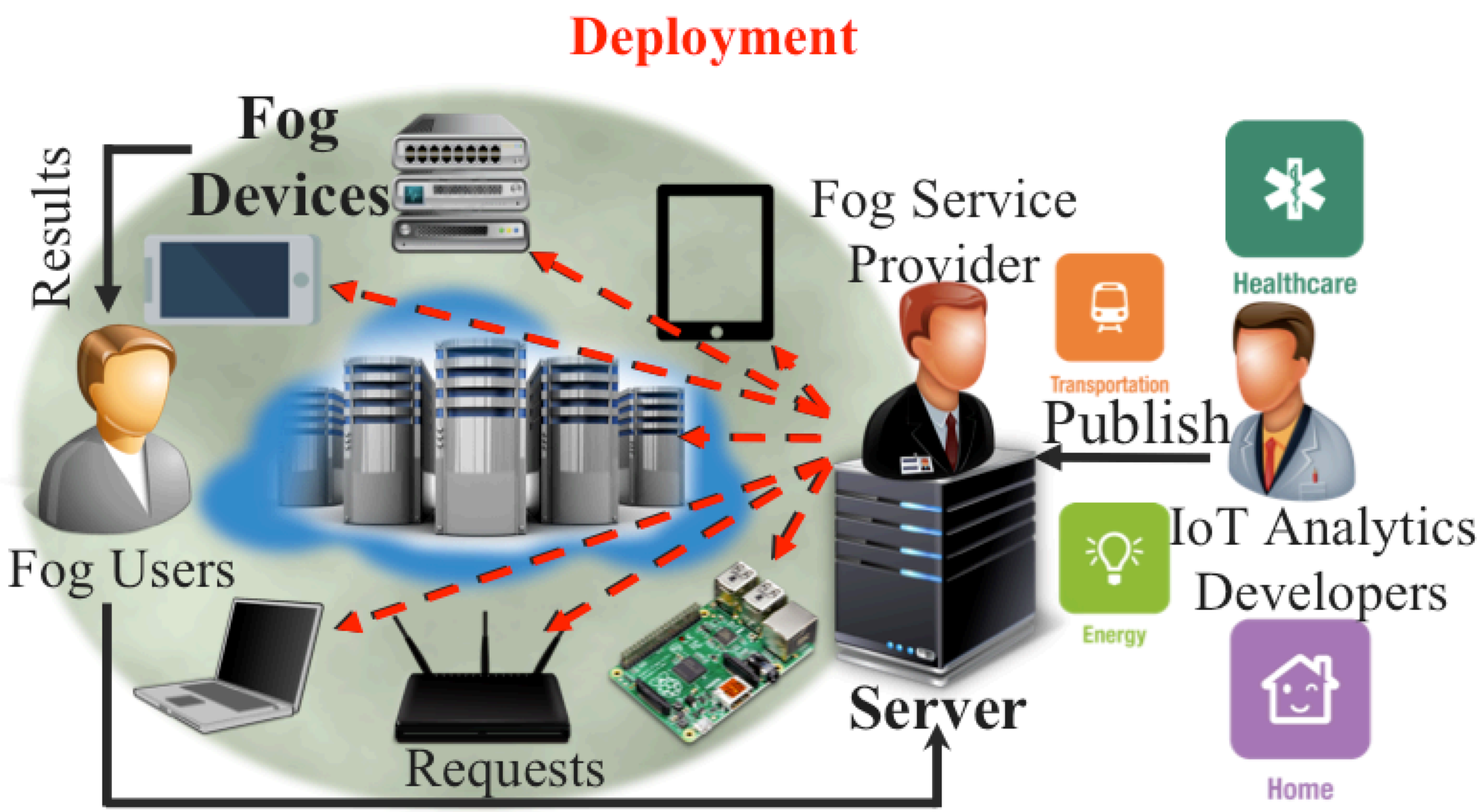
Demo Video

Conclusion

- Implementing a platform and programming model for IoT edge analytics
  - Dynamic Deployment –> Docker, Kubernetes
  - Edge Analytics –> TensorFlow
- Build a real testbed to evaluate and show the practicality and efficiency of my platform
  - Better performance of distributed analytics
  - Low overhead caused by Docker and TensorFlow
  - Tradeoff of different cut points

Future Work

- A complete eco-system for IoT



Q&A

H. Hong, **P. Tsai**, A. Cheng, and C. Hsu, “Supporting Internet-of-Things Analytics in a Fog Computing Platform,” in Proc. of IEEE International Conference on Cloud Computing Technology and Science (CloudCom’17), Hong Kong, China, December 2017.

P. Tsai, H. Hong, A. Cheng, and C. Hsu, “Distributed Analytics in Fog Computing Platforms Using Tensorflow and Kubernetes,” in Proc. of Asia-Pacific Network Operations and Management Symposium (APNOMS’17), Seoul, Korea, September 2017.

H. Hong, **P. Tsai**, and C. Hsu, “Dynamic Module Deployment in a Fog Computing Platform,” in Proc. of Asia-Pacific Network Operations and Management Symposium (APNOMS’16), Kanazawa, Japan, October 2016, Best Paper Award.