Advanced Services Engineering, Summer 2018

The role of IoT, Cloud, Blockchain and Machine Learning as Service

Hong-Linh Truong

Faculty of Informatics, TU Wien

hong-linh.truong@tuwien.ac.at
http://www.infosys.tuwien.ac.at/staff/truong
@linhsolar

Outline

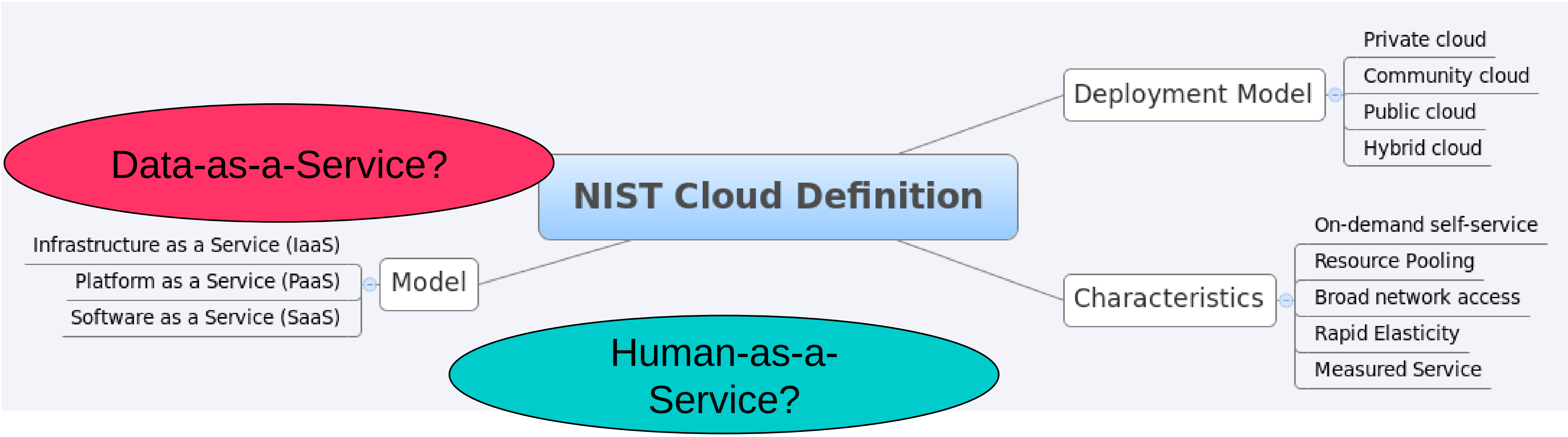
- Cloud computing
- Internet of Things (IoT)
- IoT Cloud Systems
  - Integration models
  - Services Engineering for IoT cloud systems
- Machine Learning
- Blockchain

CLOUD

NIST Cloud definitions

"This cloud model promotes availability and is composed of five essential **characteristics,** three **service models**, and four **deployment models**."

Source: NIST Definition of Cloud Computing v15, http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc

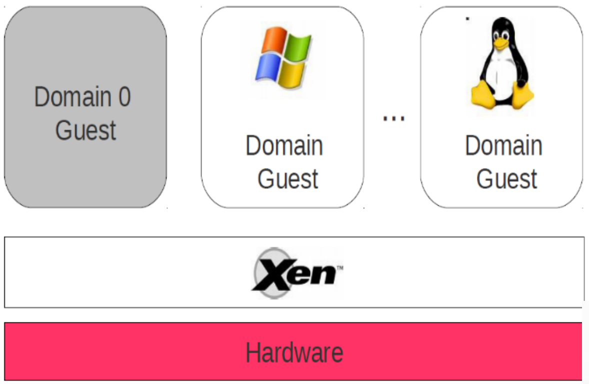


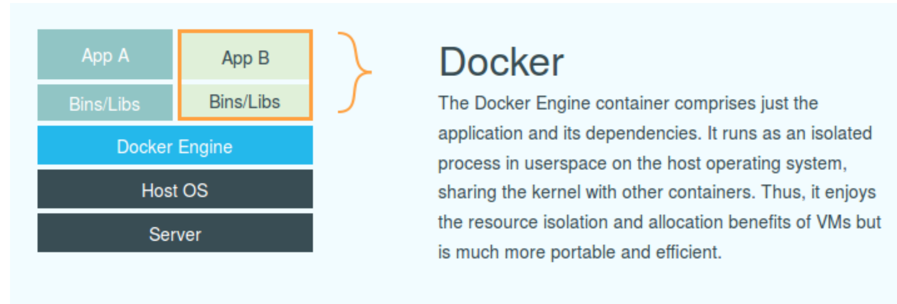
Key engineering techniques

- Virtualization
  - Compute resources (VMs, containers), networks, data, etc.
- Composition and orchestration
  - REST/SOAP services, scalable protocols and rich sets of connectors for integration
  - Private, Public and Hybrid clouds
  - Complex topologies of resources/services
- On demand and pay-per-use
  - Dynamic and runtime features (for virtualization, composition, and orchestration)
  - Hot deployment techniques, fine-grained monitoring
- Elasticity engineering
  - Resources, quality and cost based on customer-specific runtime constraints

Virtualization of execution environments

Source: The XEN Hypervisor (http://www.xen.org/)





Virtualization is a powerful concept: we can apply virtualization techniques virtually for everything!

Key applications

- Enterprise services
- IoT
- Big data
- High Performance Computing

We have private, public and hybrid models of cloud systems

Engineering your services on top of distributed clouds

- Interoperability protocols for multiple level of abstractions
  - Virtual machines, networks, cloud management APIs
- Security cross multiple domains
- Complex data governance policies and service contracts
- Complex billing and monitoring
- Which resource models do you need?
  - Cloud bursting
  - Multi-cloud distributed services

What we need to prepare?

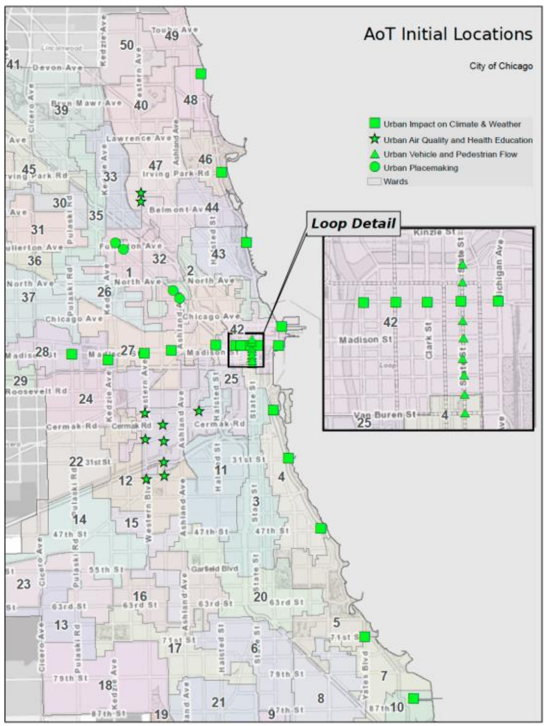
- Provision infrastructure services
  - VM provisioning, e.g. OpenStack with VmWave/VirtualBox
  - Container provisioning, e.g. Kubernetes or Mesos with Docker.
  - Managing Cloud infrastructures (e.g., with Terraform/Ansible)
- Select suitable platform services
  - Database, message broker (pub/sub), etc

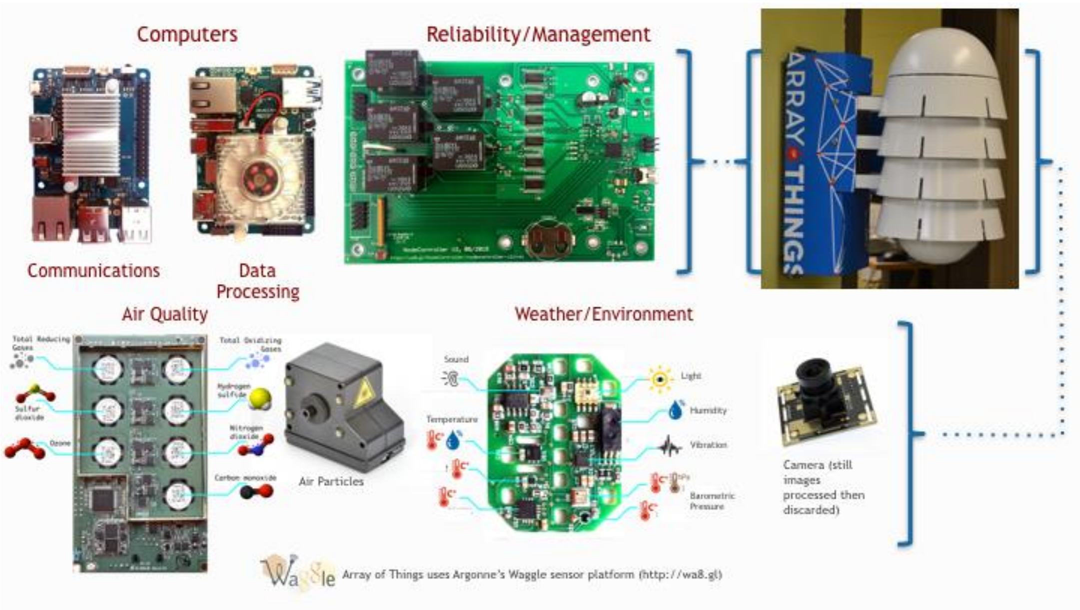
INTERNET OF THINGS (IOT)

Scenarios

- Here is a typical way:
  - PC connects to Internet
  - Mobile phone connects to Internet
- Now we have
  - Coffee machine tells us that it is ready to serve us
  - Washing machine sends its status
  - Fridge will inform us that the food has been expired
  - Etc.

IoT data in Cities



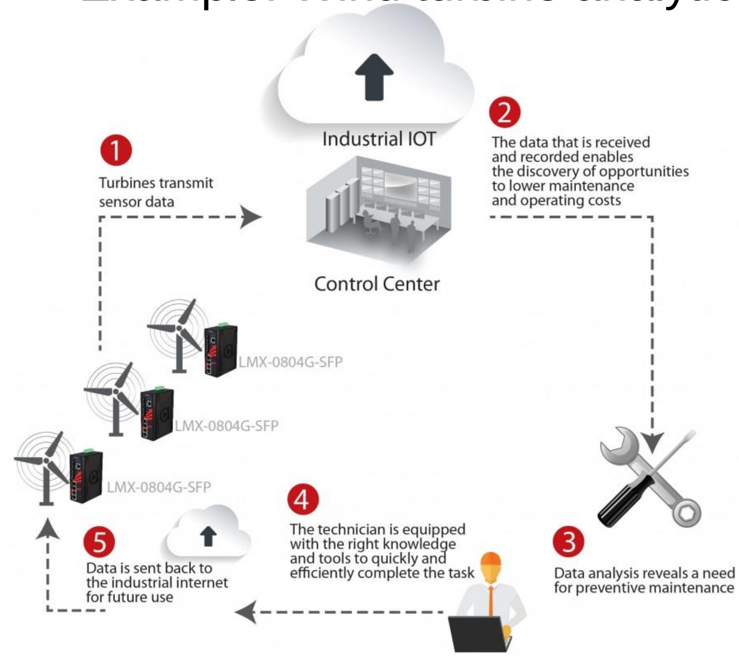


https://arrayofthings.github.io/node.html

IoT in Industrial Internet

- Monitoring industrial machines
- Industry 4.0
  - IoT and big data analytics are an essential part in manufacturing processes

Example: Wind turbine analytics



Figures source: http://www.windpowerengineering.com/design/electrical/controls/wind-farm-networks/talking-turbines-internet-things/

Smart Agriculture

Quality of water management for fish farms

Source: Erik Christensen, http://www.sensorfish.eu/

What are common things in these systems/applications?

Internet of Things (IoT)

- Things and Objects
  - Home
  - Official Business,
  - Hospital
  - Factory
  - Infrastructure

http://www.control4.com/blog/2014/03/the-internet-of-things-and-the-connected-home

- How to make such things and objects being connected and interacting each other? To be part of the Internet?
  - Why do we need this?

IoT

- Connecting physical and virtual Things to provide information infrastructures for advanced services and analytics
- Leveraging and integrating various communication protocols
- Different types of hardware/software components
  - Sensors & Actuators
  - Smart Tags and Transceiver/Receiver
  - IoT Gateways

Sensing and Actuating physical Things

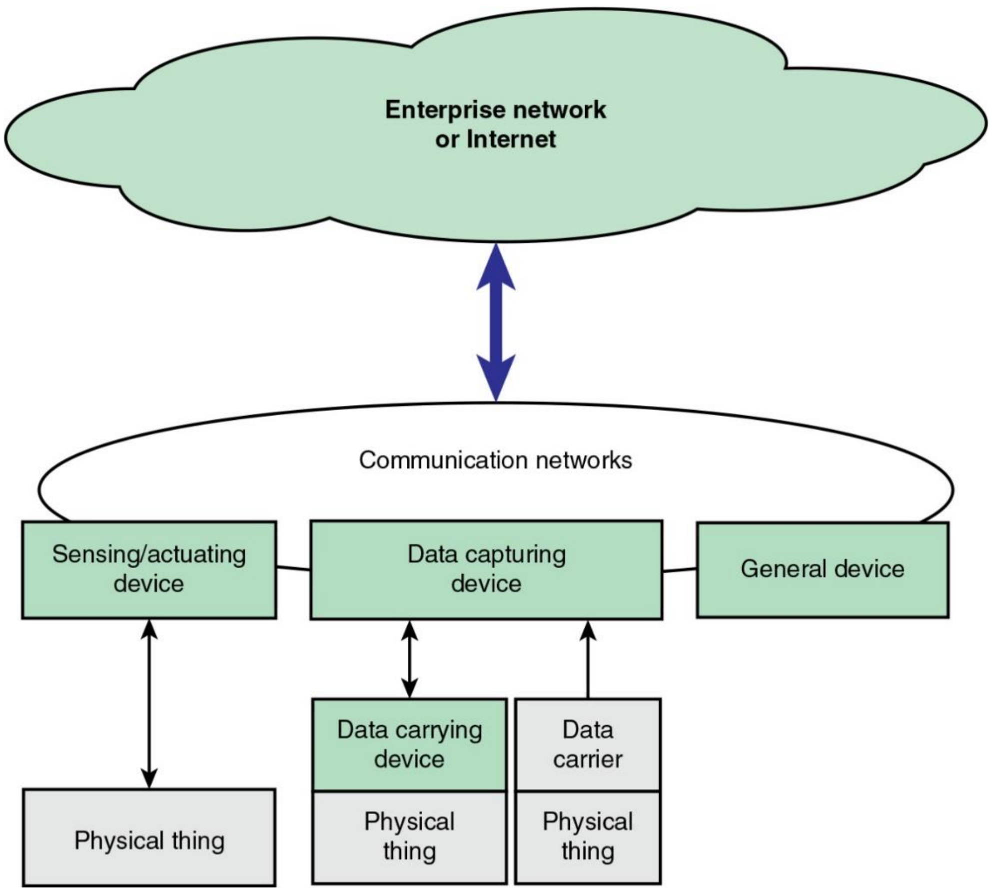


Figure source: From *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud* by William Stallings (0134175395),

Sensing and Actuating Things (2)

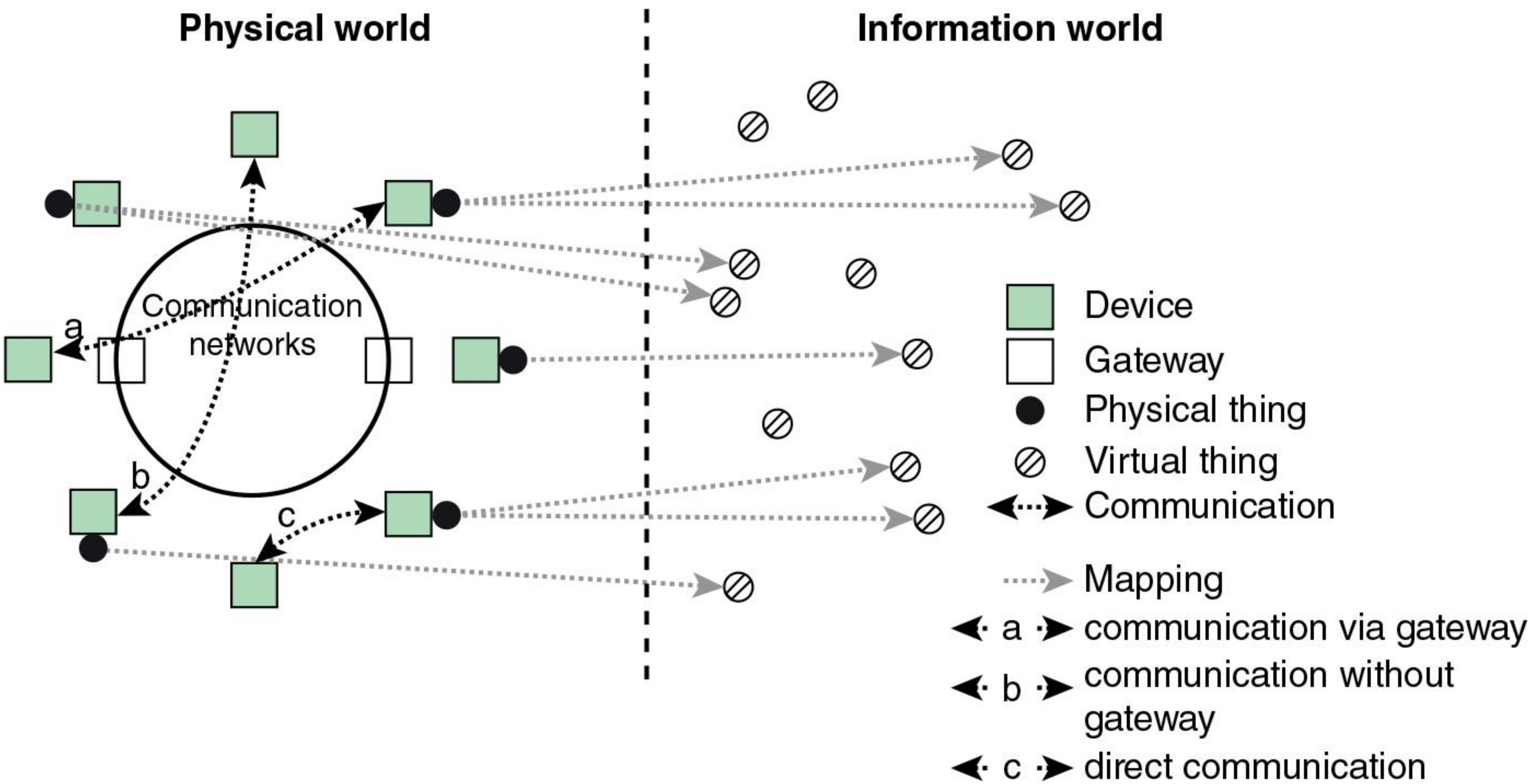
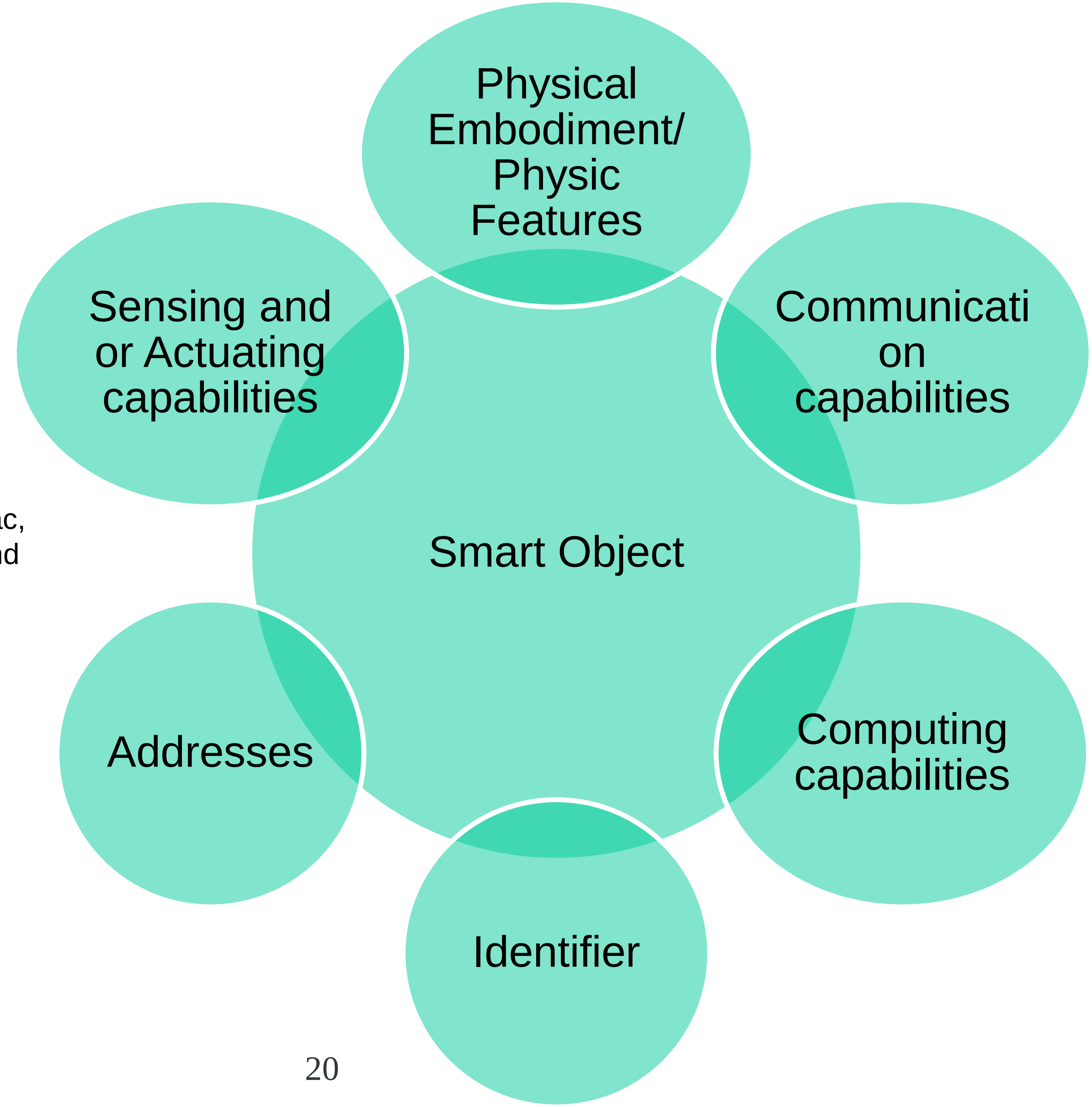


Figure source: From *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud* by William Stallings (0134175395),

Smart Object Encapsulating Thing



Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, Imrich Chlamtac, Internet of things: Vision, applications and research challenges, Ad Hoc Networks, Volume 10, Issue 7, September 2012, Pages 1497-1516

Layers of IoT Data Exchange and Processing

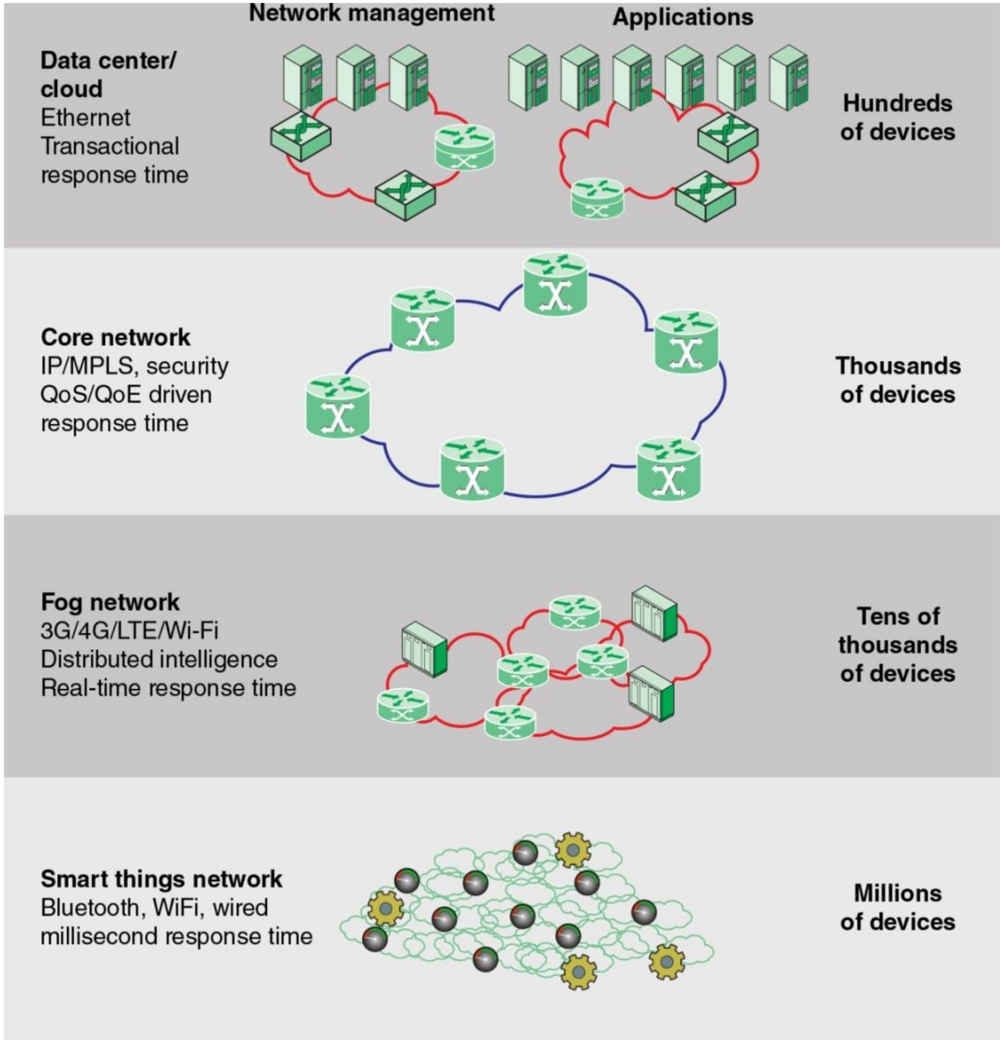


Figure source: From *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud* by William Stallings (0134175395),

Key properties

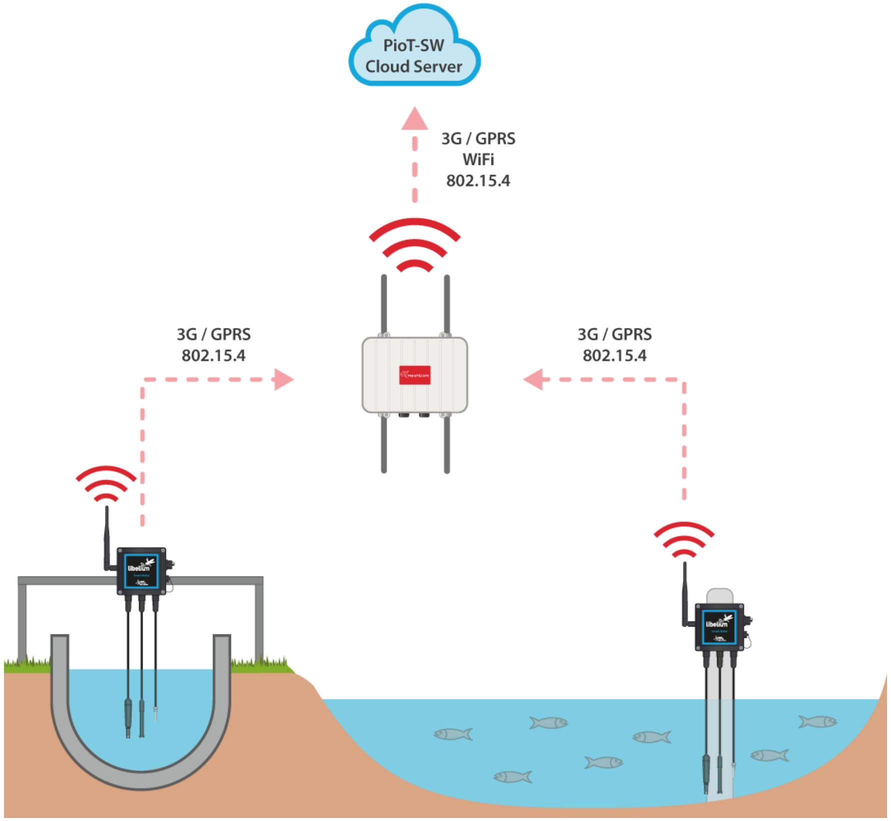
- Diversity
  - Networks: Bluetooth, Ethernet, GPRS, LoRa, ZigBee, Wi-Fi, NFC/RFID, GPS, LonWorks, Modbus, RS-232, RS-485, etc.
  - Application protocols (HTTP, MQTT, CoAP, etc.)
- Scale
  - Network scale: Body → Home → City → Internet-scale!
  - Vertical/horizontal domain objects to be studied/managed
- Complexity
  - Software stack, Network topology

3G/4G for IoT

- IoT devices have 3G/4G networks
- Hardware/software sensing and actuating Things
- Data and control commands are exchanged through mobile networks of 3G/4G

Application Example

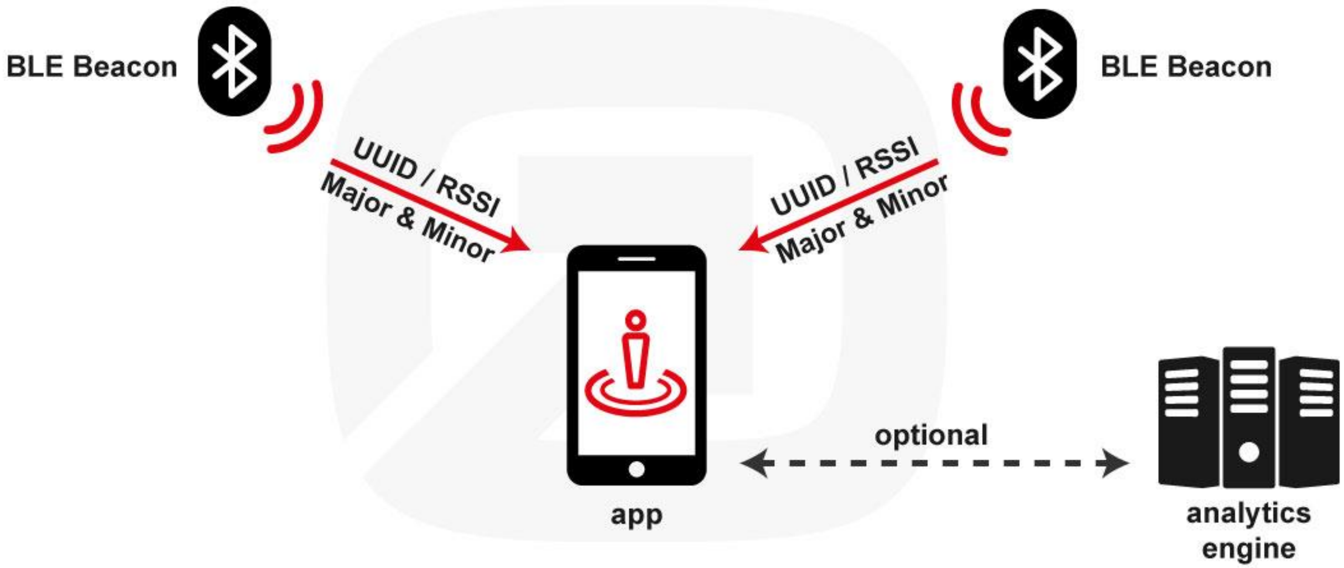
Source: http://www.libelium.com/fish-farm-monitoring-in-vietnam-by-controlling-water-quality-in-ponds-and-tanks/



BLE

- Wireless personal area networks
- Applications:
  - Indoor location
  - Asset tracking
  - Etc.

Indoor navigation



Source: http://www.indoornavigation.com/wiki-en/indoor-navigation-using-bluetooth-ble-and-beacons

Beacons

Source: http://estimote.com/

BACNet

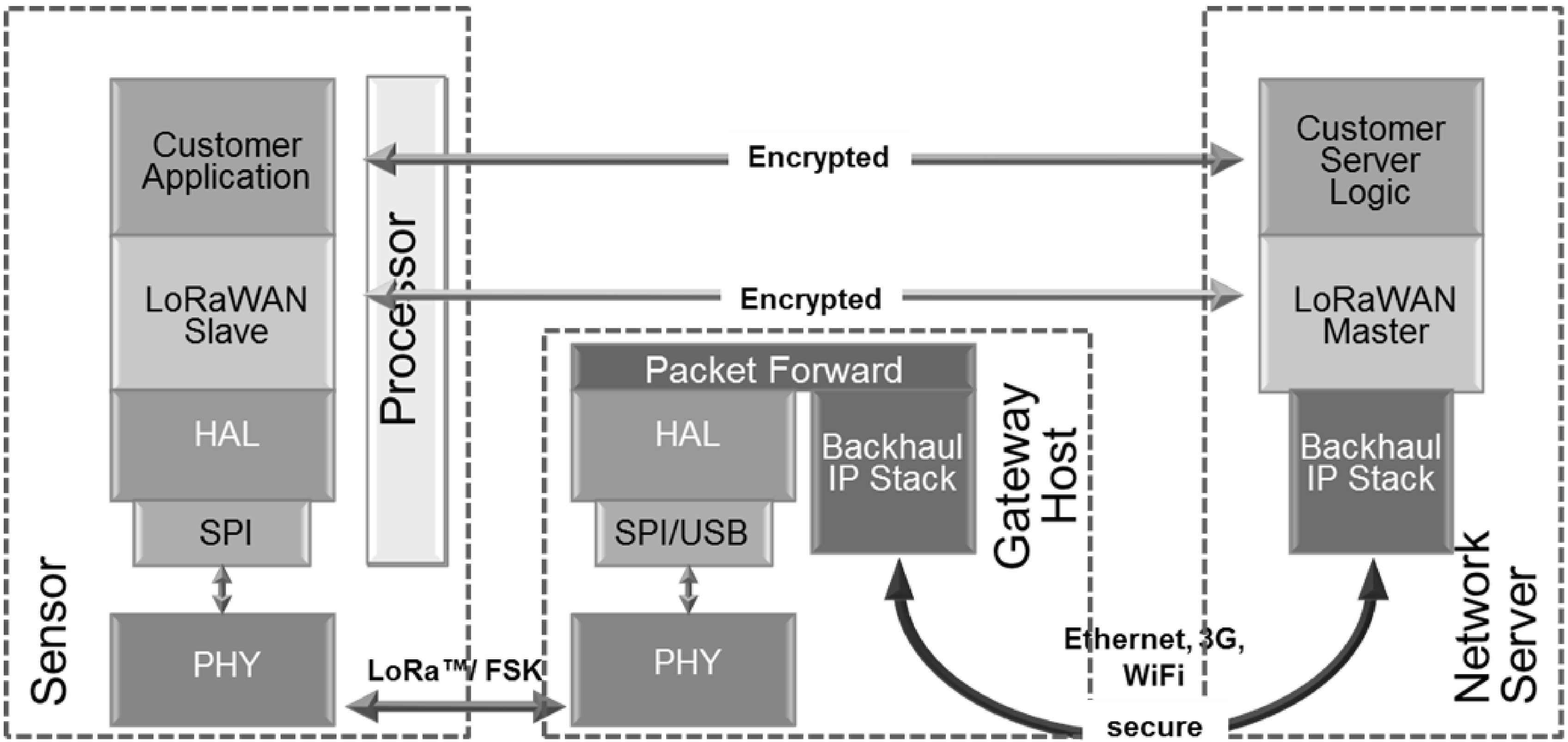
- Data Communication Protocol for Building Automation and Control Networks
  - http://www.bacnet.org/
- Applications:
  - HVAC (heating, ventilating and air conditioning) applications, lighting control, fire alarm, etc.

ZigBee

- http://www.zigbee.org/
- On top of 802.15.4 for personal network area (PAN)
- Suitable for smart home applications, metering, lighting systems

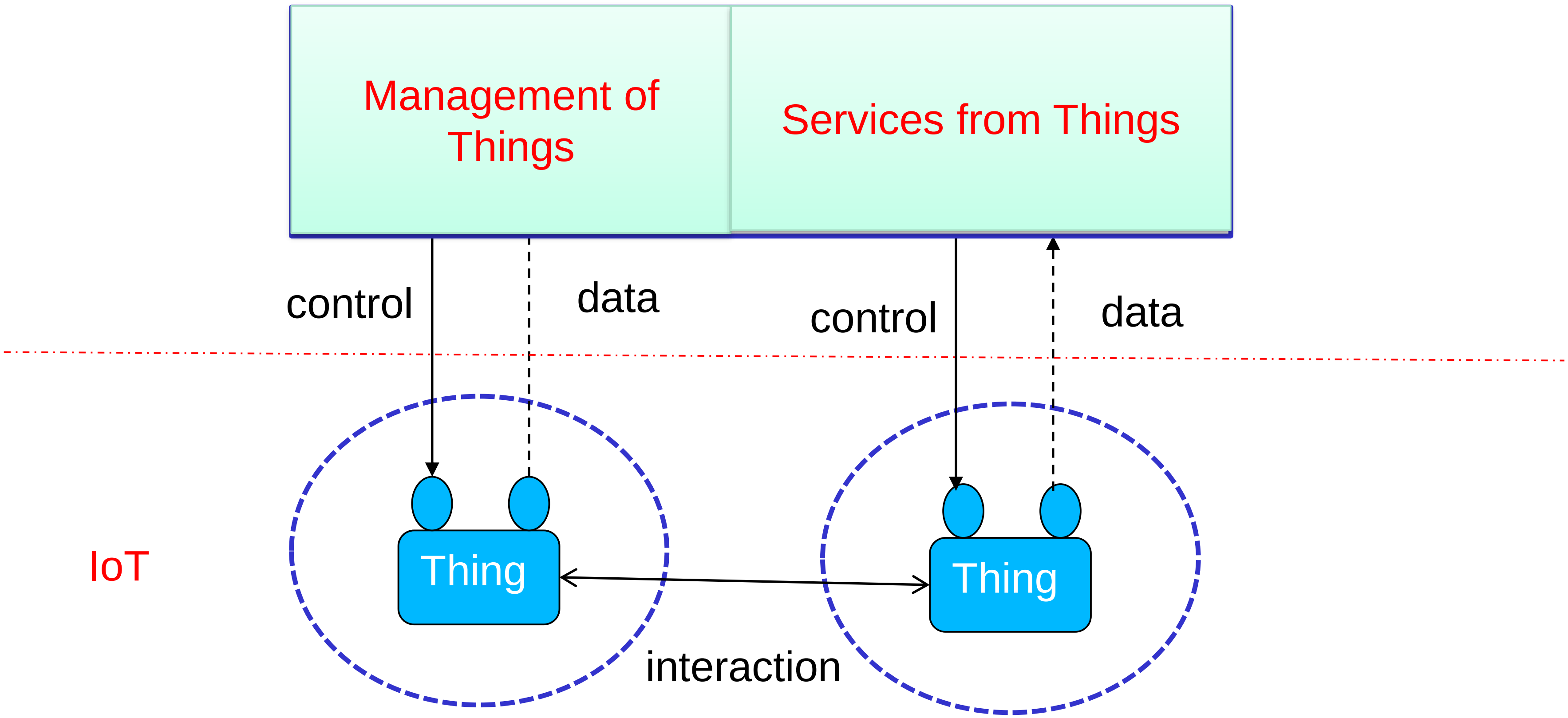
LoraWAN

- Very long distance, e.g., 22km in https://www.cooking-hacks.com/documentation/tutorials/extreme-range-lora-sx1272-module-shield-arduino-raspberry-pi-intel-galileo/



Source: https://www.lora-alliance.org/What-Is-LoRa/Technology

Management versus Service Offering



https://www.thethingsnetwork.org/community

Large-scale IoT infrastructure

https://www.sigfox.com/en/coverage

IoT Marketplaces

IoTapplications and components can be bought and deployed from marketplaces

similar to VMs, docker, and software services in the cloud

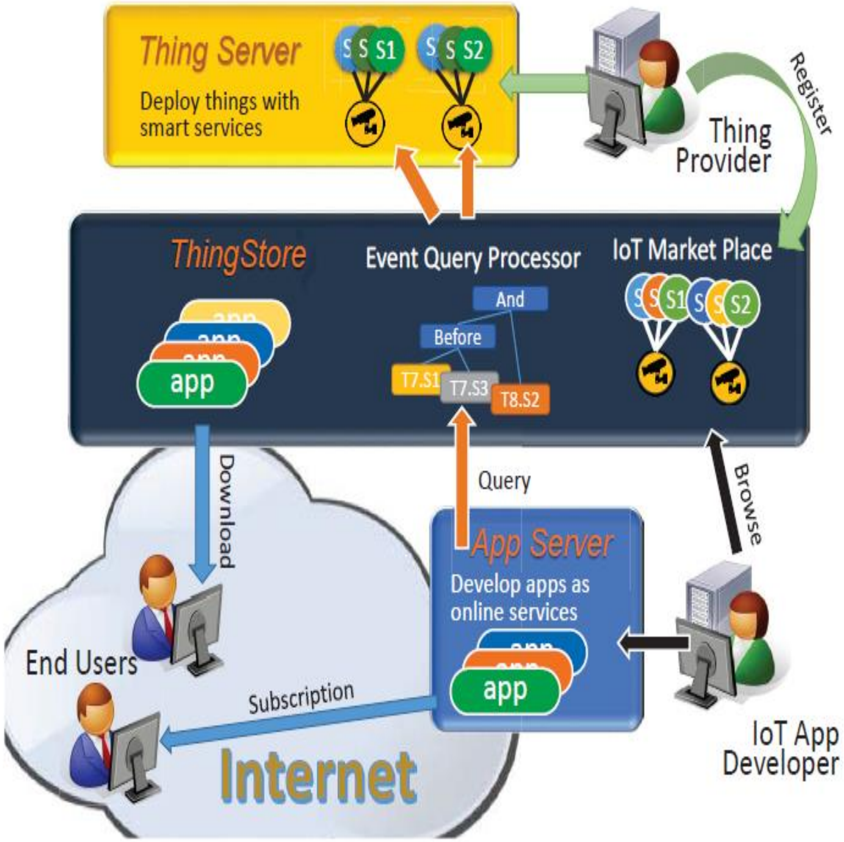
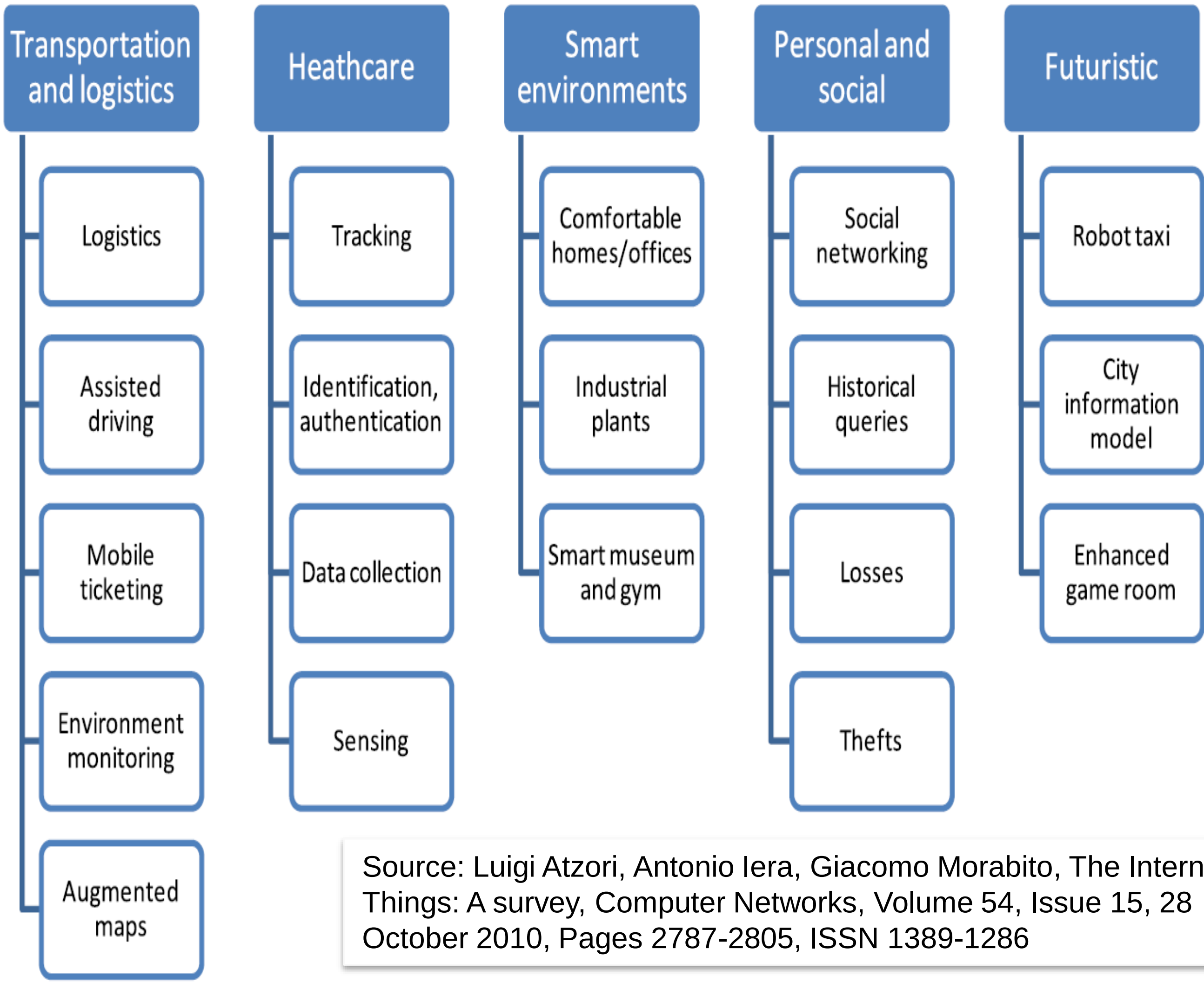


Figure 1: The IoT environment and ThingStore

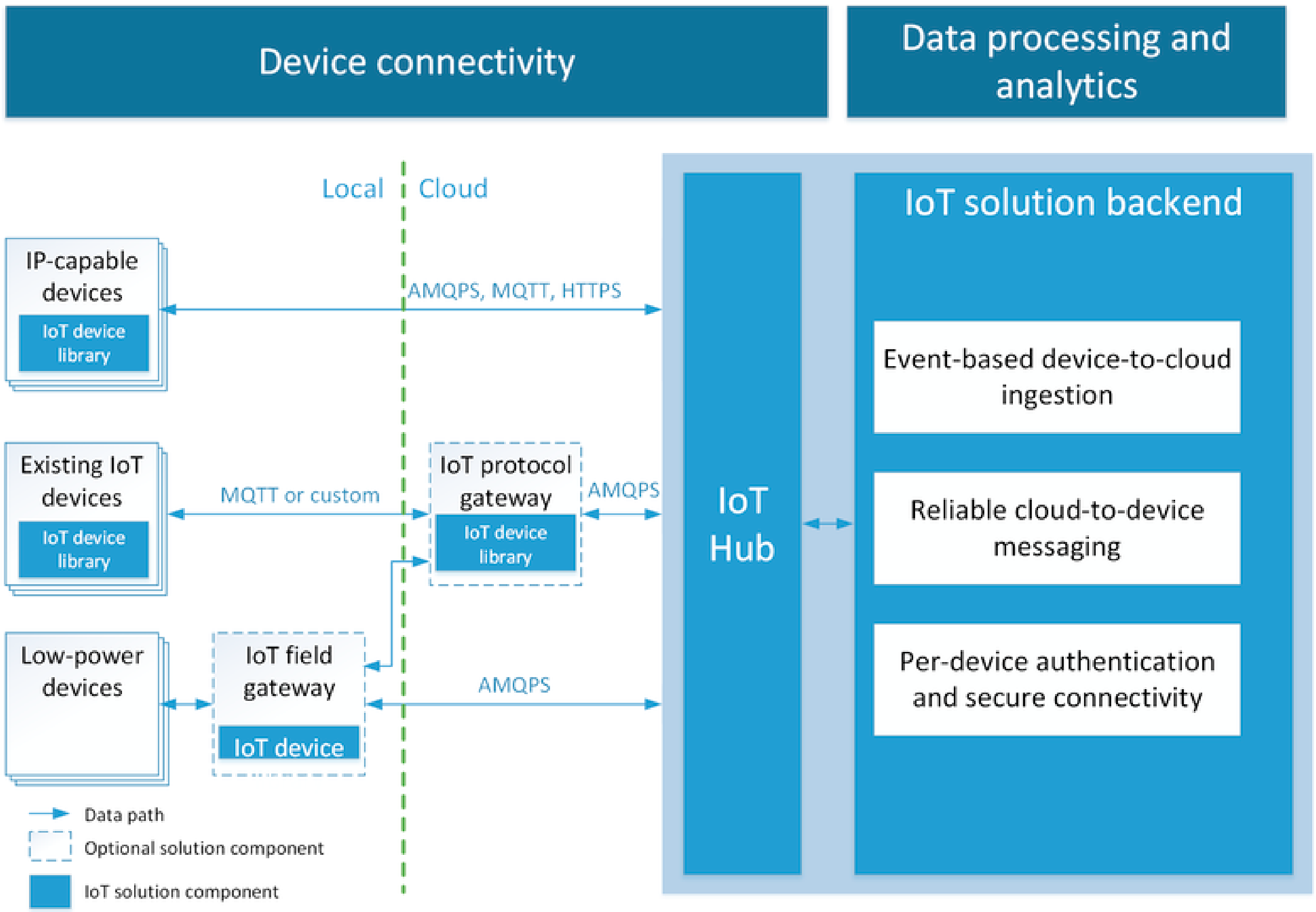
Source: Kutalmis Akpinar, Kien A. Hua, Kai Li:
ThingStore: a platform for internet-of-things application development and deployment. DEBS 2015: 162-173

Application domains



Source: Luigi Atzori, Antonio Iera, Giacomo Morabito, The Internet of Things: A survey, Computer Networks, Volume 54, Issue 15, 28 October 2010, Pages 2787-2805, ISSN 1389-1286

Azure IoT



Source: https://azure.microsoft.com/en-us/documentation/articles/iot-hub-what-is-iot-hub/

Amazon

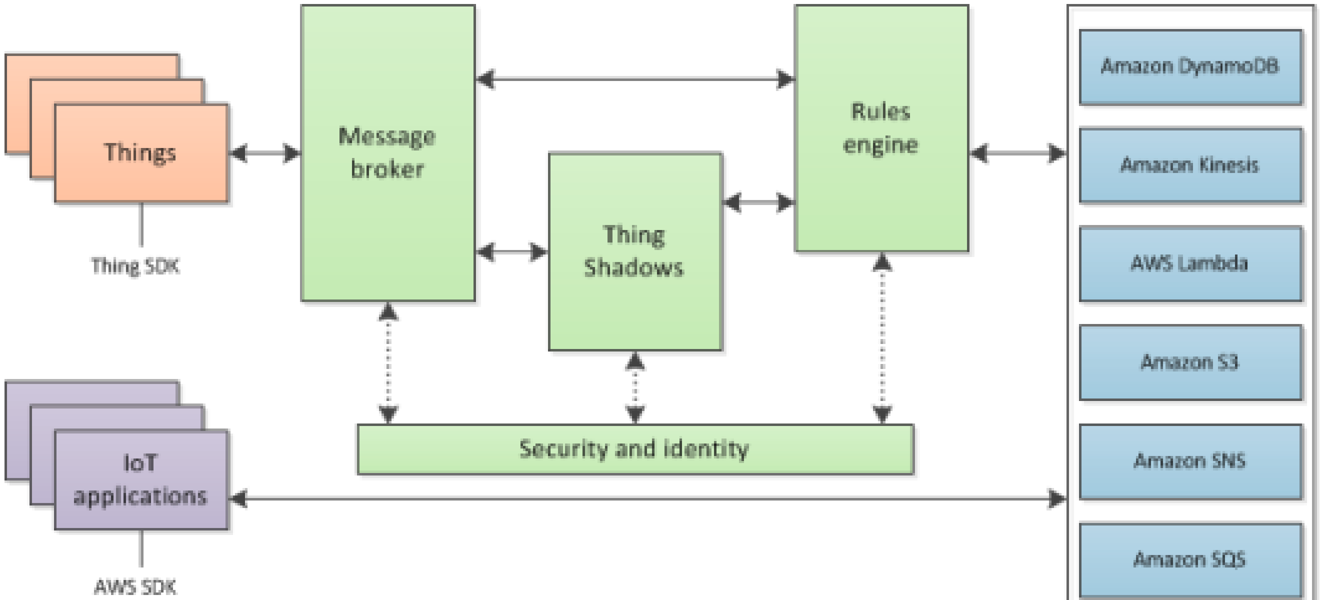
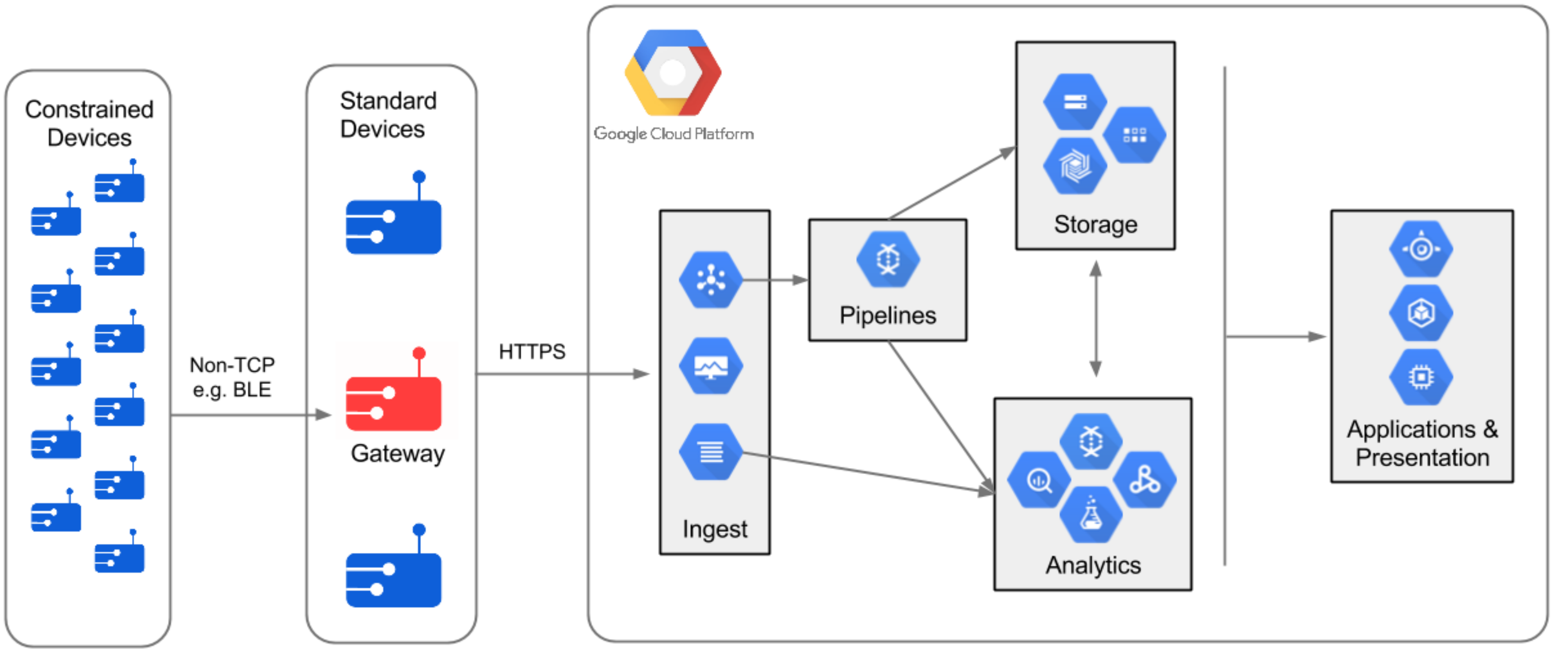


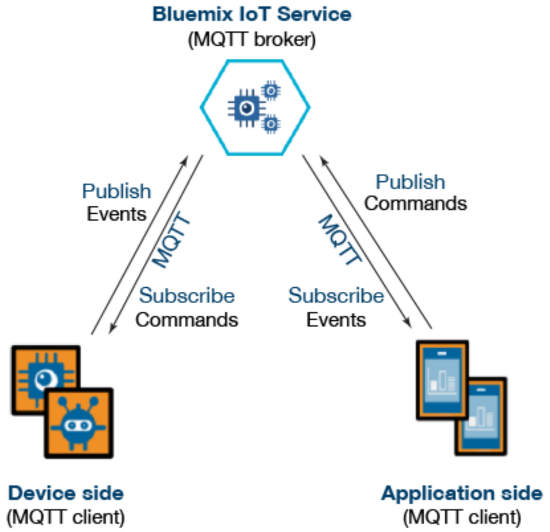
Figure source: http://docs.aws.amazon.com/iot/latest/developerguide/aws-iot-how-it-works.html

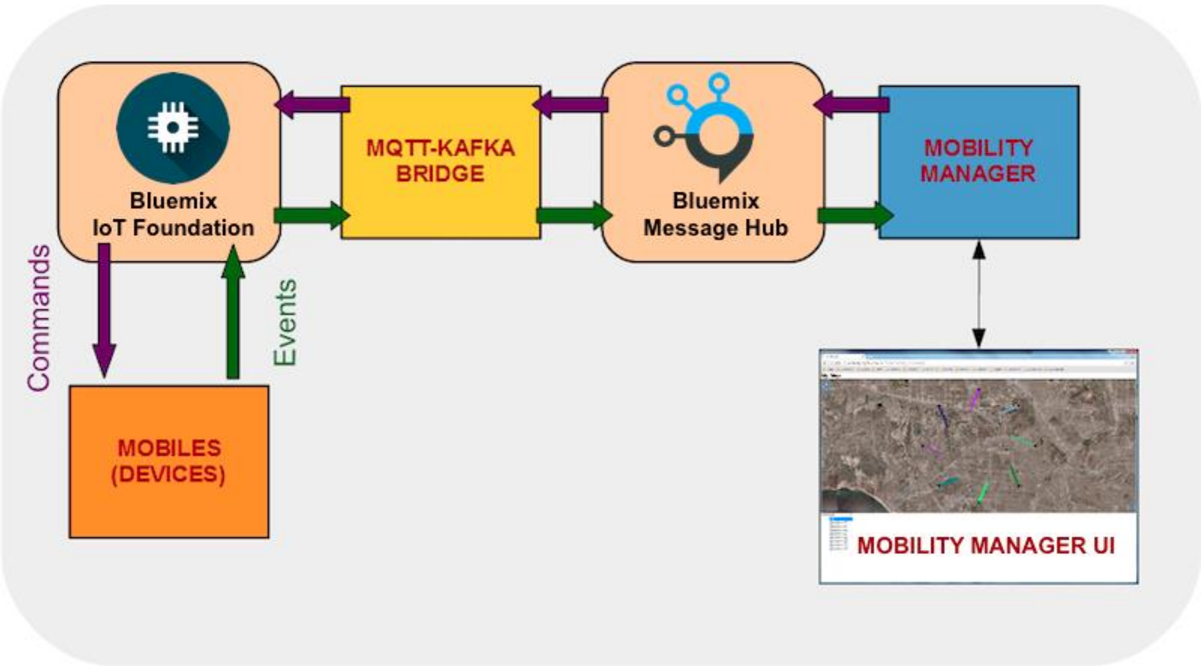
Example: IoT scenario



Source: https://cloud.google.com/solutions/architecture/streamprocessing

BlueMIX





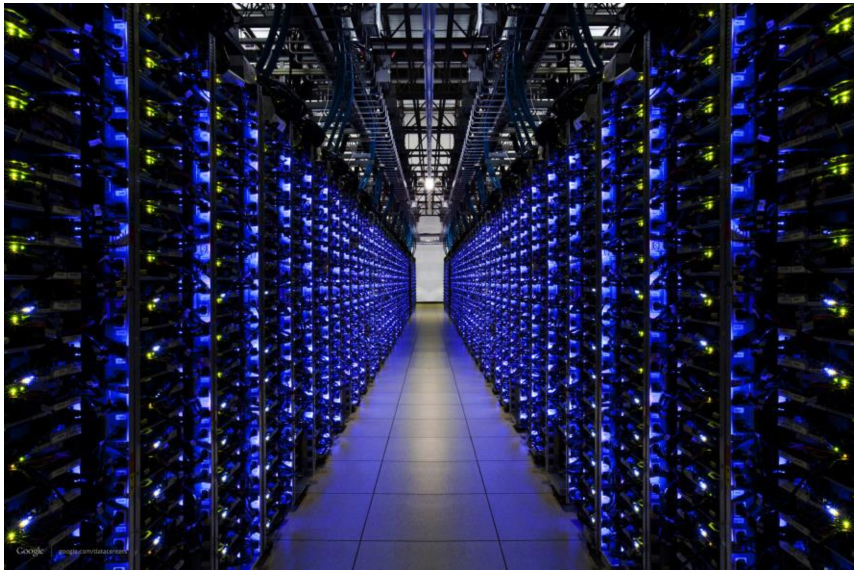
Source: http://www.ibm.com/developerworks/cloud/library/cl-mqtt-bluemix-iot-node-red-app/

Source: https://www.ibm.com/blogs/bluemix/2016/02/managing-iot-devices-with-kafka-and-mqtt/

IOT + CLOUD INTEGRATION

Connecting data centers to IoT

Data Center: Processing, Storage, Networking, Management, Distribution



Source: http://www.infoescola.com/wp-content/uploads/2013/01/datacenter-google.jpg

IoT devices: Gateways, Sensors, Actuators, Topologies of Gateways

Source: http://www.control4.com/blog/2014/03/the-internet-of-things-and-the-connected-home

IoT and Clouds

- IoT elements at the edge
  - Software sensors and actuators interfacing “Things”
  - IoT Gateways: processing, relaying and controlling
    - lightweighted hardware/software acting as intermediate nodes between sensors/actuators and back-end cloud services
- Cloud services at (big) centralized data centers
  - Software services and data: complex event processing, data services, data analytics, etc.
- Connectivity
  - Network middleware and protocols within/among IoT and clouds
- IoT-to-Cloud
  - Several systems in the middleware

IoT Cloud Systems

- We have a lot of data
- IoT sends data to the cloud
- Cloud services handle data
  - Ingest data
  - Store data
  - Analyze data

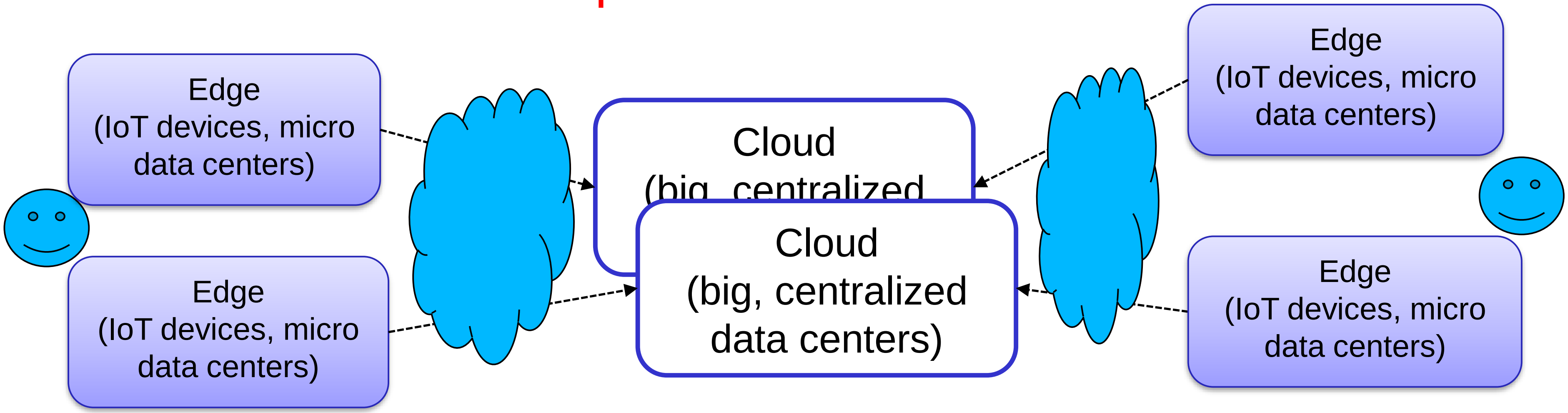
Networking issues:

- Within IoT networks
- In connectivity from IoT to cloud,
- Among services handling IoT data in Cloud

Current trends in IoT/CPS and Clouds

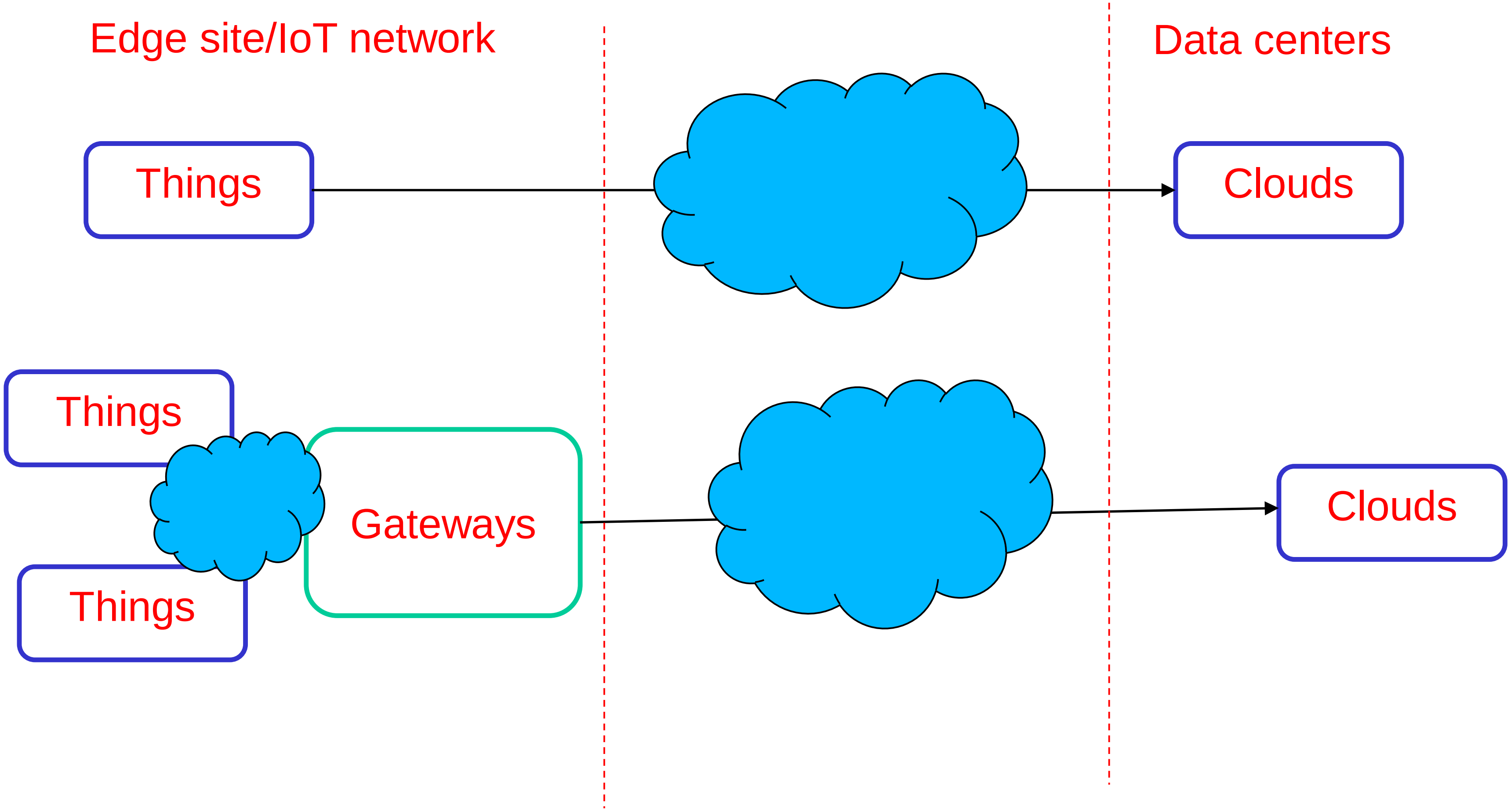
- Fog computing, edge computing, Mobile-Edge Computing, Edge-centric computing, cloudlets, Cloud of IoTs, distributed clouds (centralized data centers and micro data centers), cloud-assisted IoT, etc.

Simple view

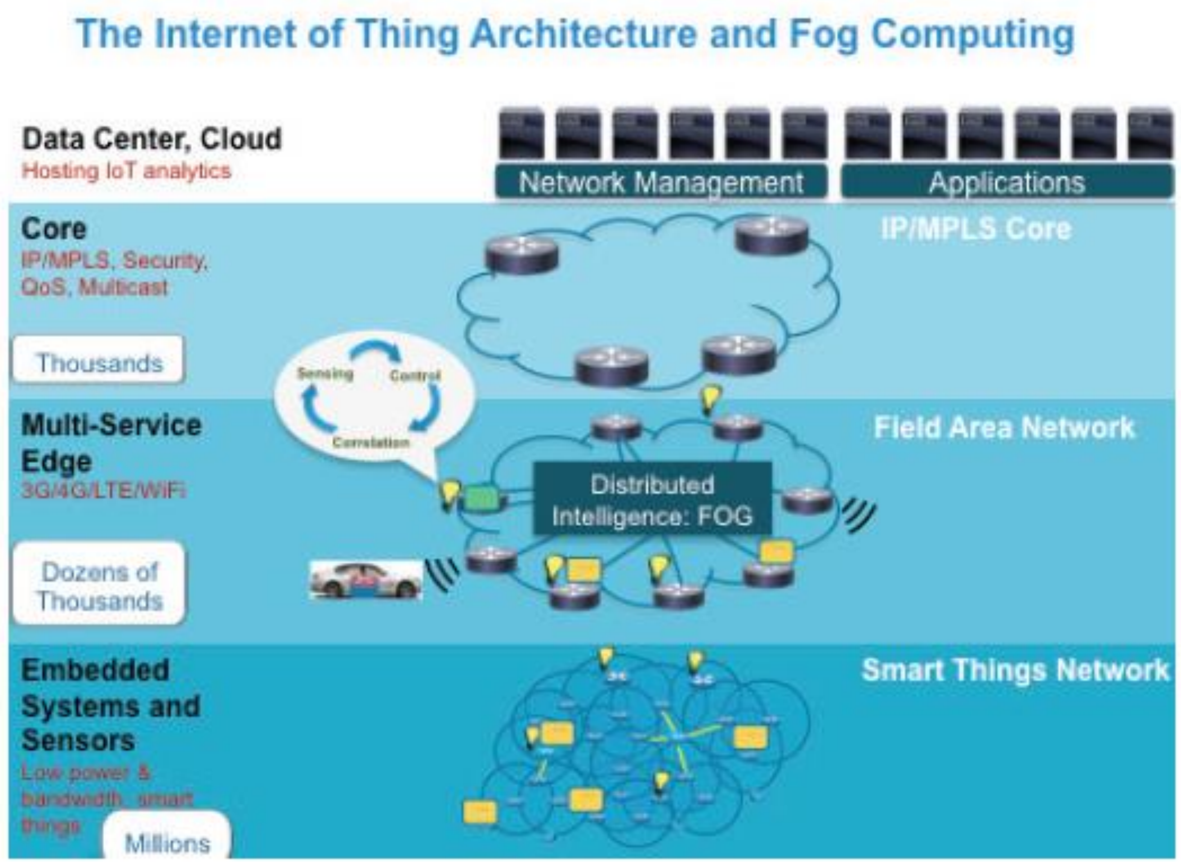


Cloud-assisted IoT

Cloud-centric IoT approach



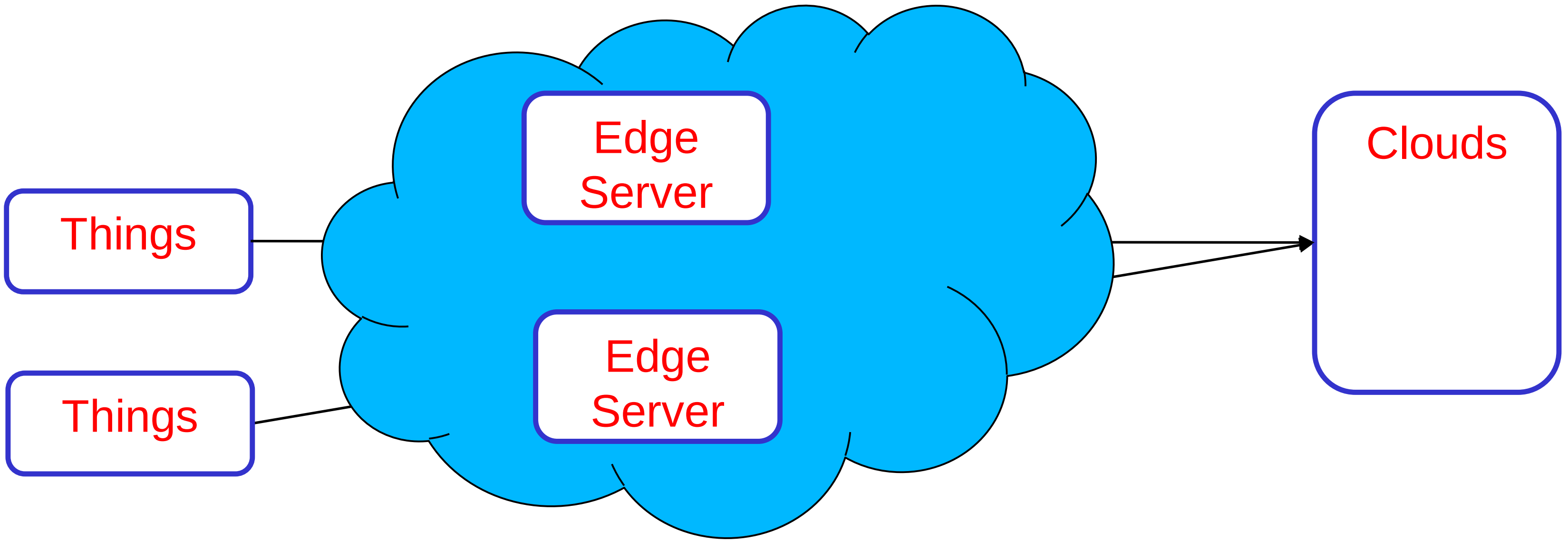
Fog computing



Source: Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing (MCC '12). ACM, New York, NY, USA, 13-16. DOI=http://dx.doi.org/10.1145/2342509.2342513

Mobile-Edge computing/Fog Computing

Edge-centric IoT approach



Edge server: network functions, storage, processing, etc

Mobile Edge Computing

"Mobile Edge Computing provides an IT service environment and cloud computing capabilities at the edge of the mobile network, within the Radio Access Network (RAN) and in close proximity to mobile subscribers. The aim is to reduce latency, ensure highly efficient network operation and service delivery, and offer an improved user experience."

Source ETSI, http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf

Video analytics + business applications/public security

Use Case 3: Video Analytics

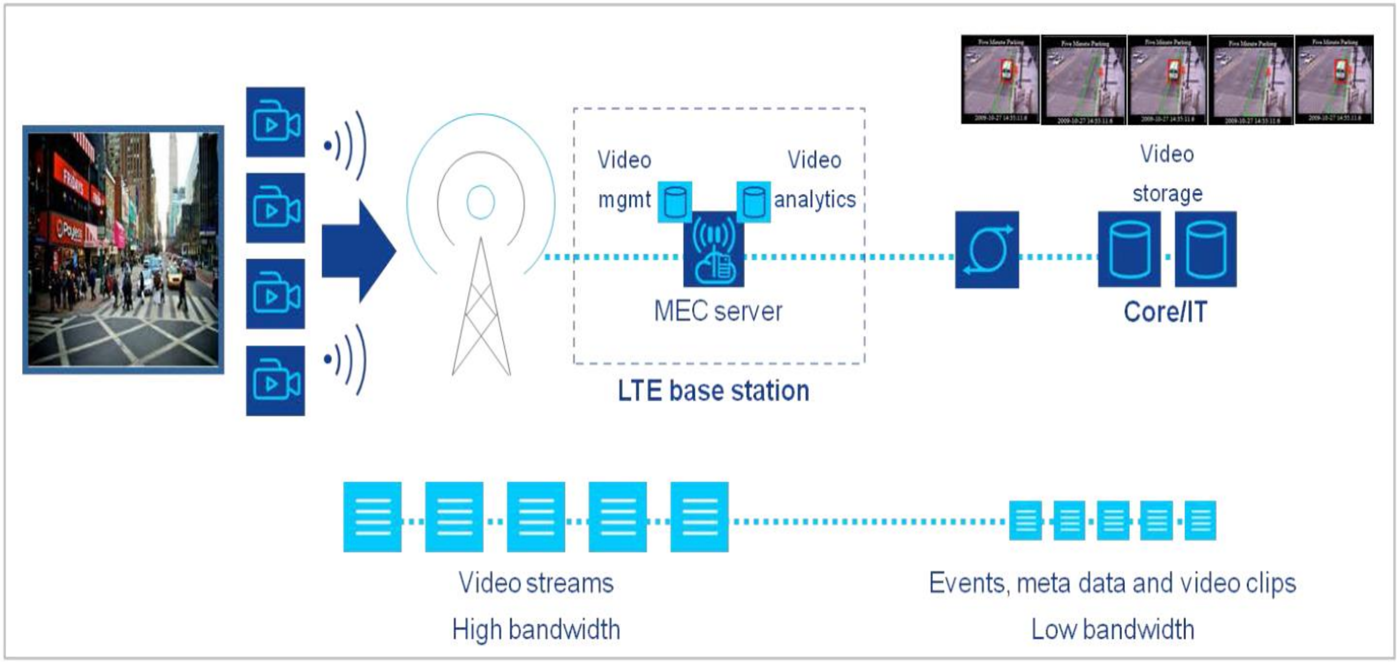


Figure 4: Example of video analytics

Figure source:
https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf

Micro data center

Source:Mohammad Aazam, Eui-Nam Huh:
Fog Computing Micro Datacenter Based Dynamic Resource Estimation and Pricing Model for IoT. AINA 2015: 687-694

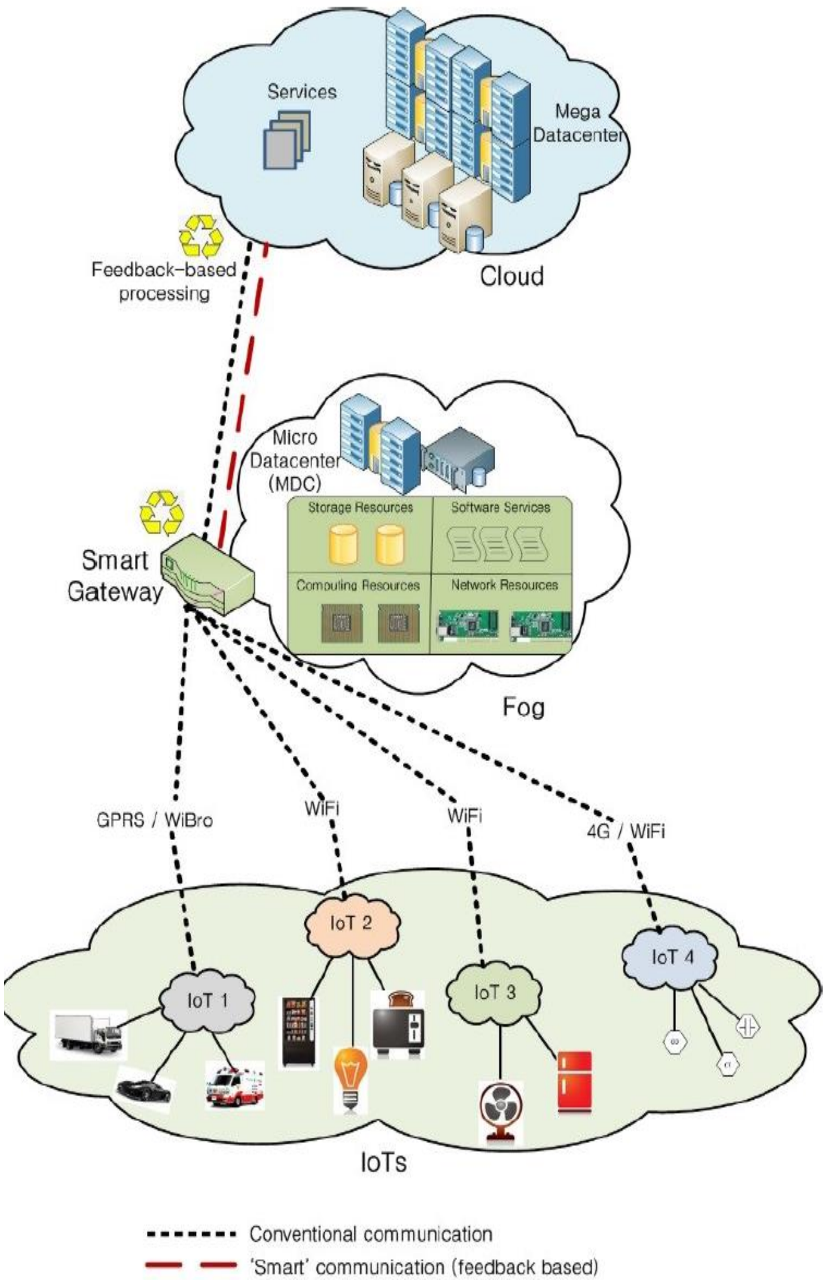
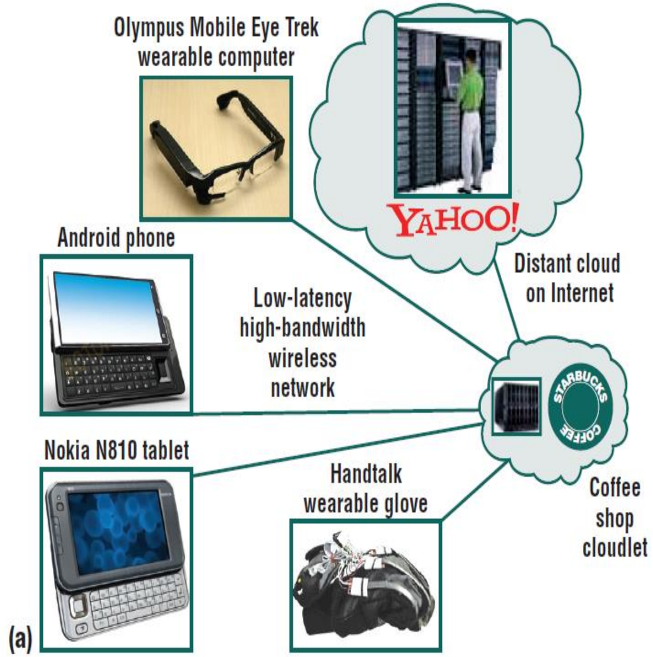


Figure 1. Smart Gateway with Fog computing.

Cloudlet



(b)

	Cloudlet	Cloud
State	Only soft state	Hard and soft state
Management	Self-managed; little to no professional attention	Professionally administered, 24X 7 operator
Environment	"Datacenter in a box" at business premises	Machine room with power conditioning and cooling
Ownership	Decentralized ownership by local business	Centralized ownership by Amazon, Yahoo!, etc.
Network	LAN latency/ bandwidth	Internet latency/ bandwidth
Sharing	Few users at a time	100s-1000s of users at a time

Figure 4. What is a cloudlet? (a) The cloudlet concept involves proximate computing infrastructure that can be leveraged by mobile devices; it has (b) some key differences with the basic cloud computing concept.

Source: Mahadev Satyanarayanan, Paramvir Bahl, Ramón Cáceres, Nigel Davies:
The Case for VM-Based Cloudlets in Mobile Computing. IEEE Pervasive Computing 8(4): 14-23 (2009)

Cloudlet

Mahadev Satyanarayanan, Pieter Simoens, Yu Xiao, Padmanabhan Pillai, Zhuo Chen, Kiryong Ha, Wenlu Hu, Brandon Amos:
Edge Analytics in the Internet of Things. IEEE Pervasive Computing 14(2): 24-31 (2015)

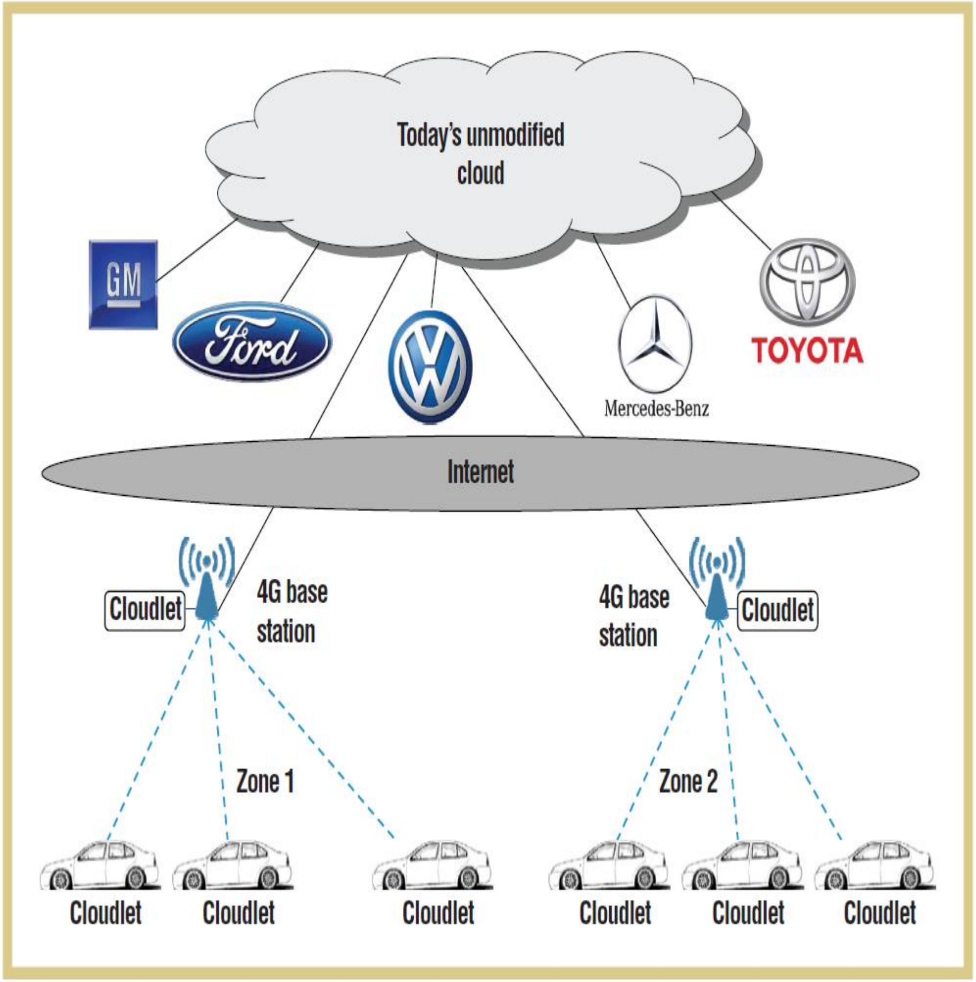


Figure 4. GigaSight for cars. Continuous capture and real-time analytics of car-mounted video cameras can help to improve road safety when shared via cell tower cloudlets.

Today‘s cyber-physical systems

“Cyber-physical systems integrate computation, communication, sensing, and actuation with physical systems to fulfill time-sensitive functions with varying degrees of interaction with the environment, including human interaction.”,

Source NIST, http://www.cpspwg.org/Portals/3/docs/CPS%20PWG%20Draft%20Framework%20for%20Cyber-Physical%20Systems%20Release%200.8%20September%202015.pdf

Today‘s cyber-physical systems

- Physical systems:
  - Things/physical entities
  - Network, sensors, actuators
- Cyber systems
  - Networked services/Cloud services
- Similar to IoT+Cloud?
- But:

Higher degree of combination and coordination between physical and cyber entities, especially control flows.

IoT + Cloud integration models

- What are common network structures?
- What are common data and control flows or interactions between IoT and clouds?
- What are common types of applications?
- What are common data models for IoT?
- Quality attributes: Performance, scalability, security, privacy, etc.

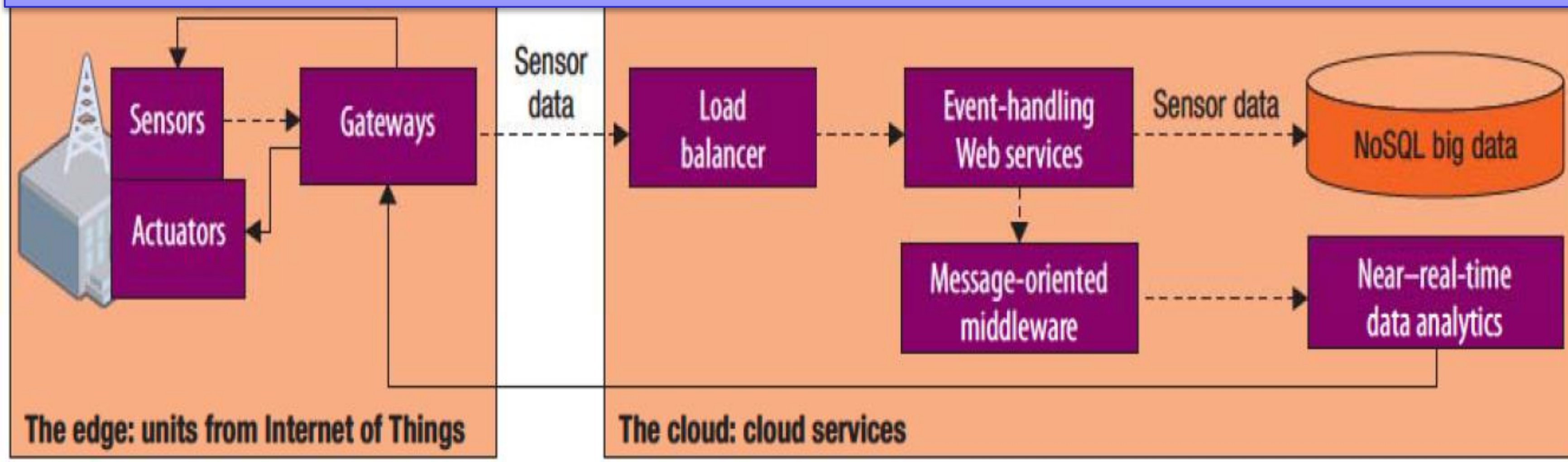
EXPERIENCES FROM IOT CLOUD ENGINEERING

Key views

- Neither edge-centric nor cloud-centric approach
  - Should depending on application scenarios
- Resources from IoT, network functions and clouds are blending.
- Data and control follows
  - Edge → cloud
  - Cloud → edge
- Depending on the situation:
  - edge-centric or cloud-centric usage

Application

IoT Cloud System

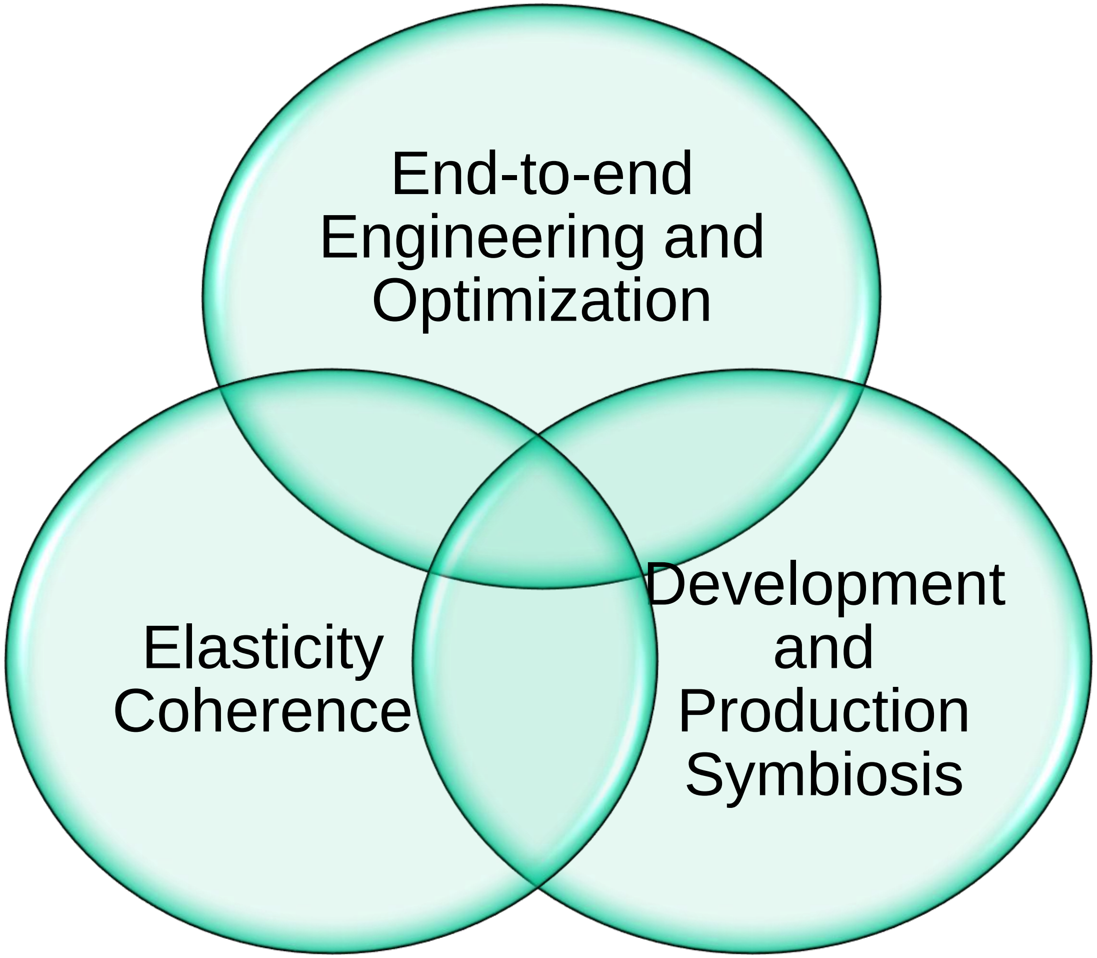


- But we do not want a separation!
- How to build a coherent view!

Lifting IoT software stack

- Software artifact management
- Virtualization
  - Virtualization of software components for certain requirements: e.g., for a single application/domain
- Composition and orchestration
  - Complex topologies of IoT components
- Software-defined capabilities
  - Management done via APIs at runtime
- Cloud connectivity
  - Hide low level network stuffs

Engineering perspectives



Hong Linh Truong, Schahram Dustdar: Principles for Engineering IoT Cloud Systems. IEEE Cloud Computing 2(2): 68-76 (2015)

Challenges

1. Enable virtualization and composition of IoT components as unit

   Selection, composition, pay-per-use

2. Enable emulated/simulated IoT parts working with production IoT/cloud services

   Symbiotic development and operation

3. Enable dynamic provisioning of IoT and cloud service units through uniform marketplaces and repositories for multiple stakeholders

Challenges (2)

4. Provide multi-level software stack deployment and configuration

5. Provide software-defined elasticity and governance primitive functions for all IoT units and cloud service units

6. Provide monitoring and analysis for an end-to-end view on elasticity and dependability properties

Challenges (3)

7. Coordinate elasticity to enable a coherent elastic execution through the whole IoT cloud systems

8. Deal with uncertainties

9. Assure security and privacy during system on-demand provisioning and elasticity

Tools for building IoT Cloud tests

- Sensors can be written in Java/Python
  - Simulated/emulated sensors just read data from sample files
  - Sensors to gateways communication: TCP/IP, MQTT, BLE, etc.
- Gateways:
  - Raspberry Pi
  - Gateway-to-cloud: MQTT/AMQP, Web socket, etc. over LAN or 4G
- Clouds:
  - Your own clouds or using Google, Microsoft, Amazon
- Take a look at:
  - https://github.com/rdsea/IoTCloudSamples

Common goals for IoT Cloud service engineering analytics

- Type 1
  - **Mainly focus** on IoT networks: sensors, IoT gateways, IoT-to-cloud connectivity (e.g., connect to predix.io, IBM Bluemix, Amazon IoT, etc.)
- Type 2
  - **Mainly focus** on (public/private) services in data centers: e.g., load balancer, NoSQL databases, and big data ingest systems
  - Using both open sources and cloud-provided services
- Type 3
  - **Equally focus** on both IoT and cloud sides and have the need to control at both sides
  - Highly interactions between the two sides, not just data flows from IoT to clouds

Software components

Requirements

- Not just cloud services but also different types of (virtual) sensors, gateways, and connectivity components
- Different implementation models and pay-per-use models

Solutions

- Treat all software components as service units
- Use *IoT* ***software-defined unit concept*** to abstract functions, provisioning and management mechanisms

Examples:

- Software-defined IoT units: sensors, gateways, lightweighted analyzers, light storage, queues, VMs, cloud services (NoSQL databases, streaming data analyzer, etc.)
- Leverage a combination of Docker Hub, Git-based repository, Maven, and IoT marketplaces

(Virtual) Software sensors

Requirements

- Emulated + real software sensors in different scenarios
- Work with different types of APIs

Solutions

- Sensors with dynamic control capabilities
- Solutions for individual sensors and sensor topologies
- Coupling sensors with virtual environments

Examples:

- Cloud connectivity: e.g., MQTT and AMQP
- Data Point/Control Points associated with Things: emulation with data sources from files, accessing Things with software-defined gateway API profiles or with Thing-specific APIs
- Management API: REST, MQTT, and even shellscripts

IoT Gateways

Requirements

- Light-weighted gateways but (partially) support fog/edge computing models

Solutions

- *Software-defined gateways* as virtual environments for deploying different units
- Dynamic configuration of software components

Examples:

- Utilizing lightweight virtual containers/VMs
- ***New middleware*** with high-level APIs that abstract Data Points, Control Points, Cloud connectivity, etc.
  - https://github.com/tuwiendsg/SoftwareDefinedGateways
- ***Composition of existing middleware***
  - Leverage microservice models to enable flexible adaptation,

Software deployment and configuration

Requirements

- Support different types of IoT gateways with different interfaces
- Complex configuration protocols for different topologies

Solutions

- Integrate different configuration tools and languages
- Develop protocols and programming APIs for exchanging configuration commands

Examples:

- Use *TOSCA/YAML (or any configuration language)* for both IoT and clouds based on deployment agents supporting TOSCA
- REST/MQTT-based APIs for configuration at the IoT gateways
- Leverage known tools (Vagrant, docker) for the emulated sensors, gateways and cloud services

Monitoring and analytics

Requirements

- Cross IoT and cloud analytics
- High level end-to-end metrics
- Correlate monitoring data from different service/component topologies

Solutions

- Expand monitoring capabilities to the IoT gateways
- Interfaces to deployment, configuration and control systems
- User-defined metrics

Examples:

- Relay basic metrics from IoT gateways (e.g., nr. sensors, queue length, etc.) to centralized monitoring services
- **All** management services subscribe/exchange monitoring data through scalable queues
- Analyze based on service topologies

Control

Requirements

- Able to control at the IoT side
- Must work with various protocols/systems (e.g., using public clouds)
- Control in *combined with business data analytics*

Solutions

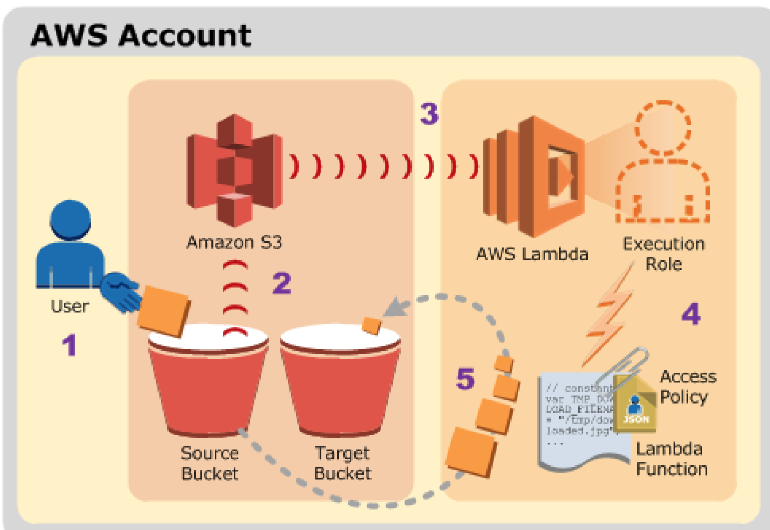
- Primitive actions for gateways: control sensors, cloud connectivity, etc.
- High-level coordination working with different interfaces

Examples

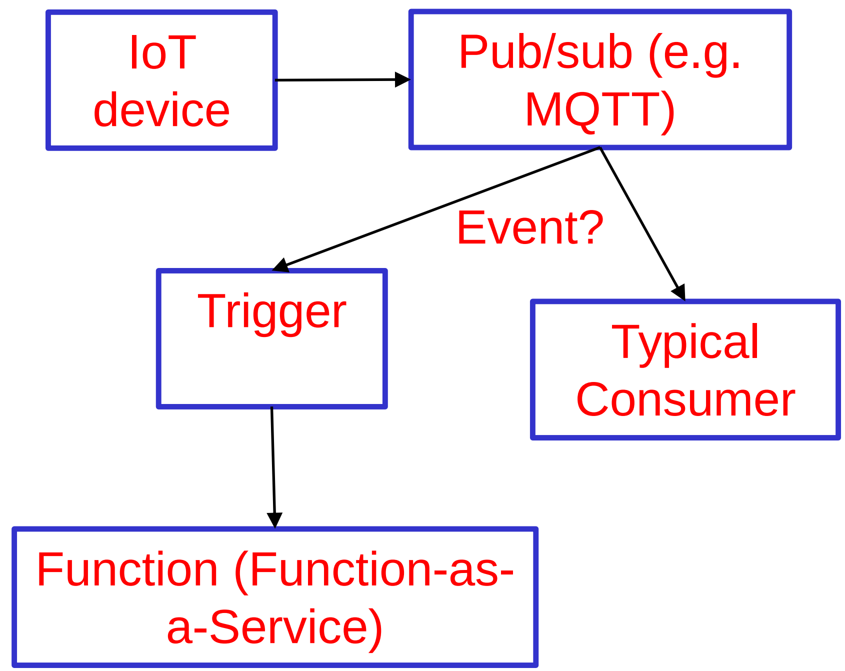
- Using REST API/shellscripts to control IoT gateways + elasticity at the data centers
- Programing-language based workflows for coordination-aware controls to enable different connectors
- Deal with uncertainties

IoT and Serverless function

- Serverless function is triggered based on events
- Which types of events in IoT could be used to trigger serverless functions



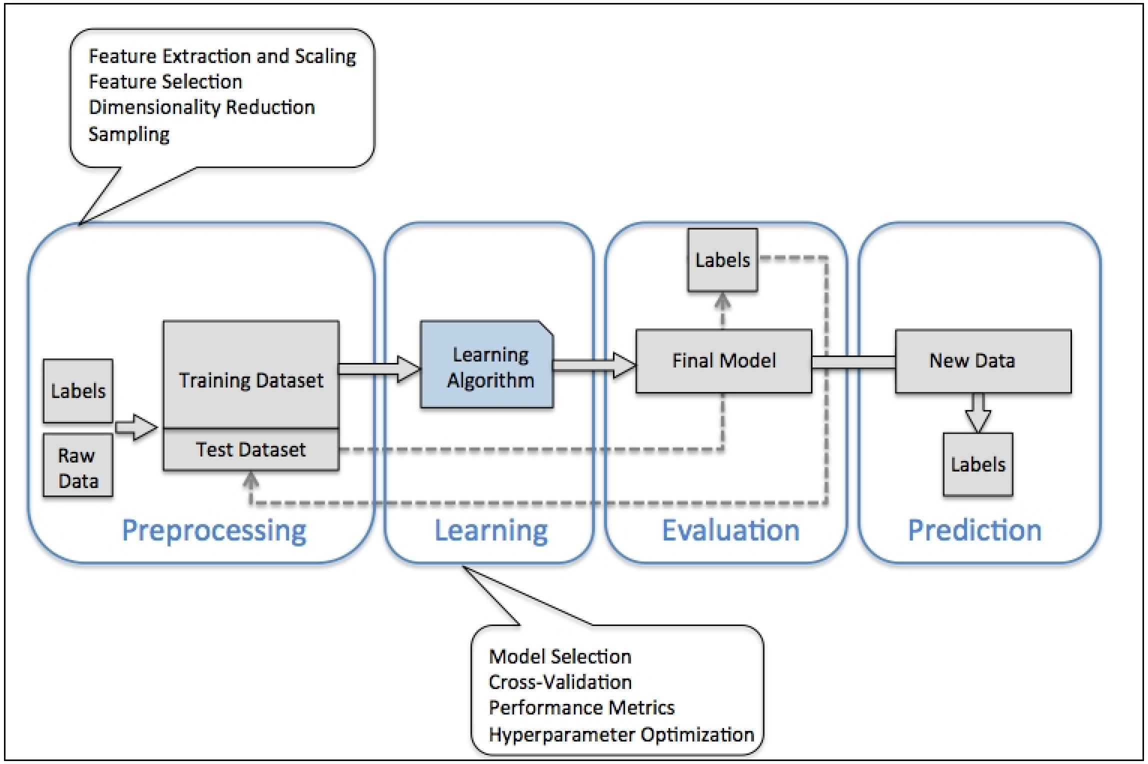
Source: http://docs.aws.amazon.com/lambda/latest/dg/with-s3-example.html



MACHINE LEARNING

Machine Learning

Create models learned from data and use models to predict data



Source: Python: Real-World Data Science by Dusty Phillips; Fabrizio Romano; Phuong Vo.T.H; Martin Czygan; Robert Layton; Sebastian Raschka, Web ISBN-13: 978-1-78646-841-3

Machine learning Techniques

- Predictive analytics:
  - Can we predict the future behavior of our services, consumers, etc? based on monitoring data and current consumer requests?
  - Business domain: banking, heathcare, …
- Various algorithms
  - Linear Regression, Logistic Regression, Clustering, Decision tree, Neutral networks
  - Supervised versus un-supervised algorithms

In the context of our course

Predictive analytics for optimizing our service operations:

- Optimizing costs and service provisioning
- Controlling objects that service manage
  - E.g., equipment
- Resource allocation
  - Can we use humans versus software

Writing your prediction

- scikit-learn
- Apache Spark
- Tensorflow
- Keras
- Azure ML (https://docs.microsoft.com/en-us/azure/machine-learning/studio/studio-overview-diagram)
- Google ML (https://cloud.google.com/ml-engine/docs/technical-overview)
- Provisioning as a service: Predict.IO, AzureML, Google ML

Scikit-learn

http://scikit-learn.org

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

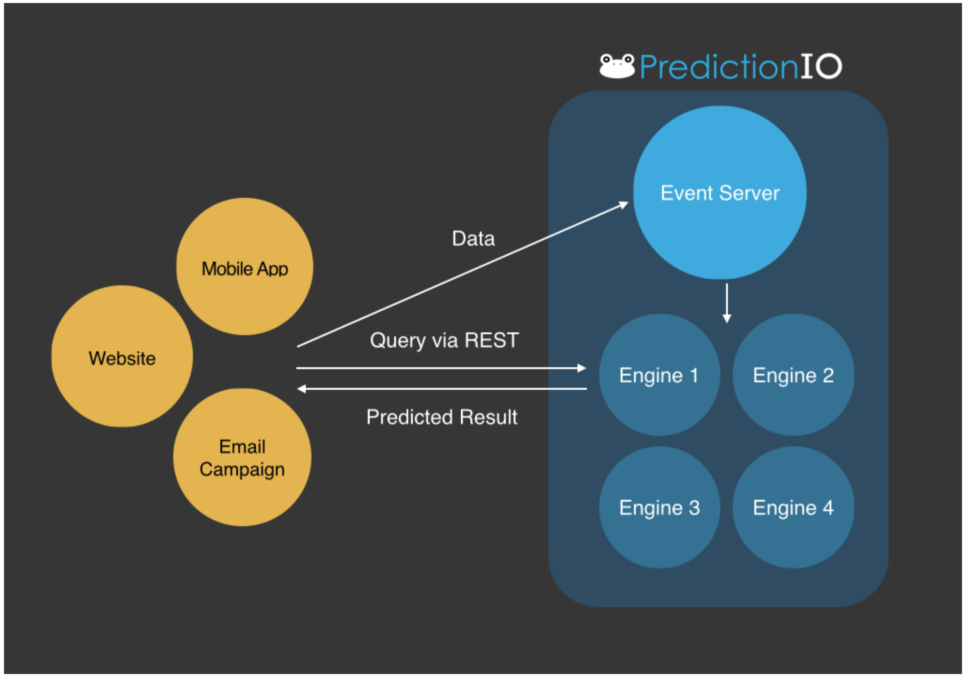
Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

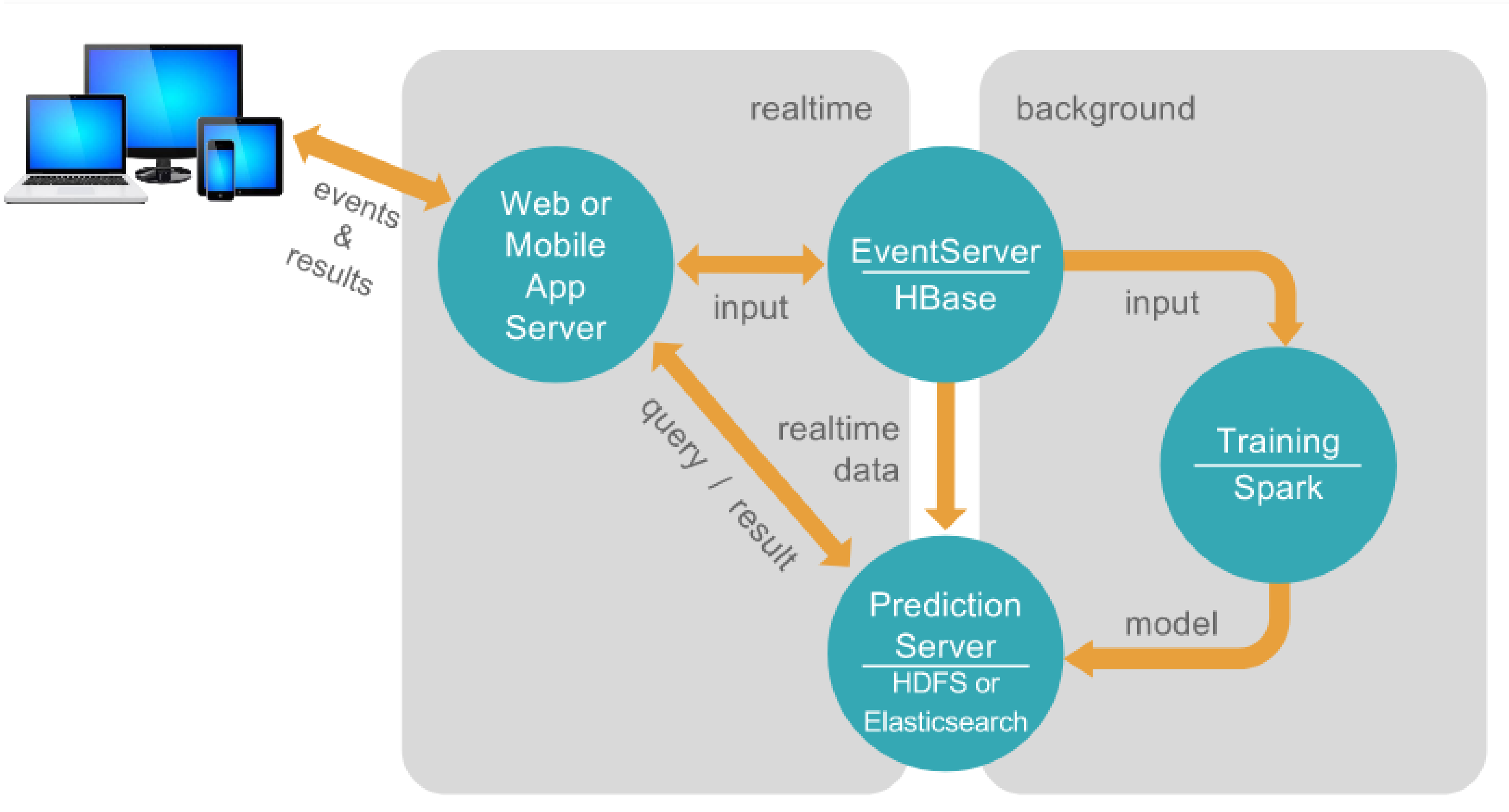
Predictiion.IO

Making a prediction algorithm as a service



Souce: https://predictionio.apache.org/start/

Prediction.IO



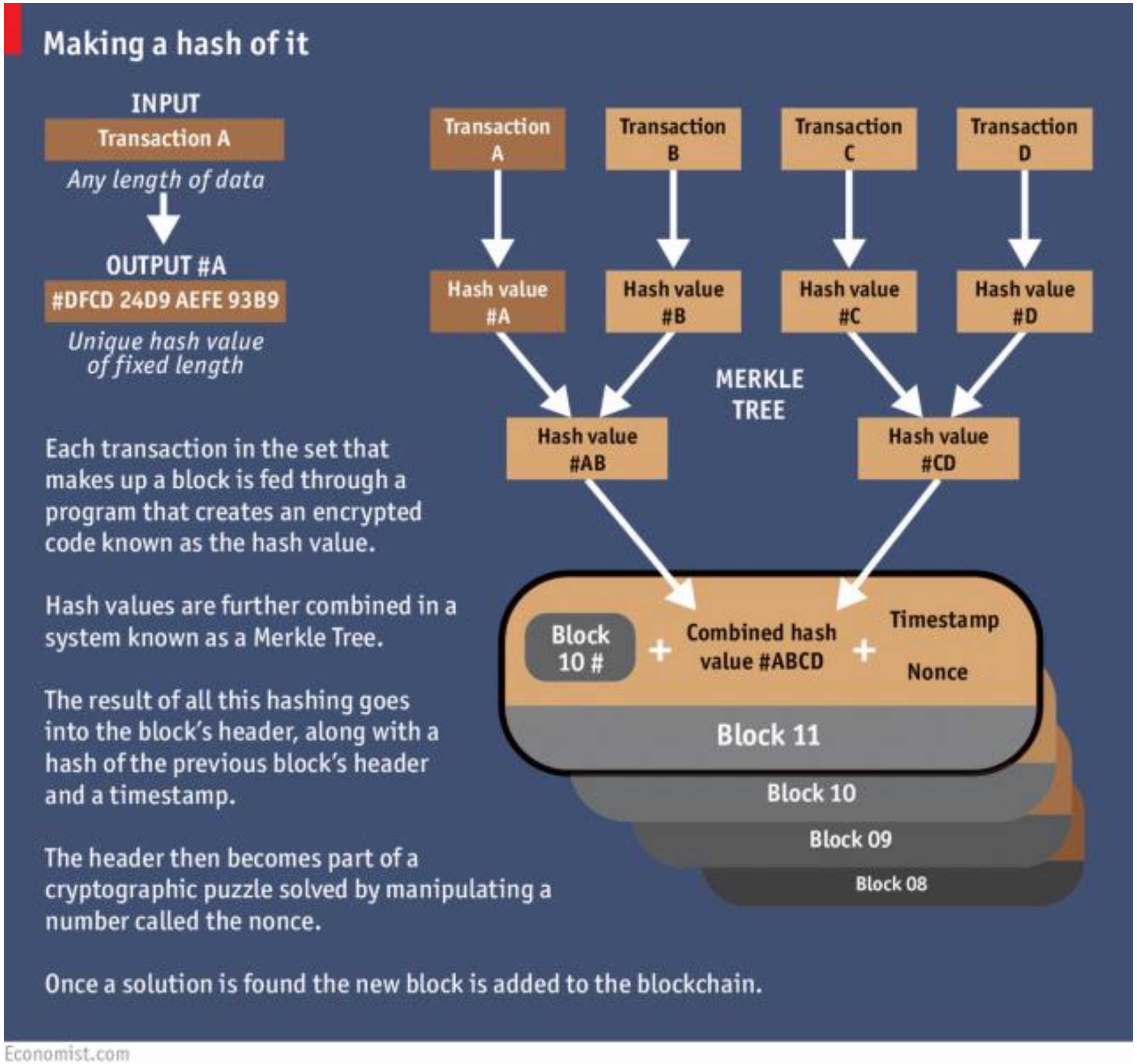
Source: https://predictionio.apache.org/system/

BLOCKCHAIN

Blockchain key concepts

- Blocks are linked and secured using cryptophytic techniques, establishing a chain of blocks
- Block includes cryptographic hash of the previous block, timestamp, transaction data
  - Transaction data is verifiable and permanent
- A blockchain is managed by a peer-to-peer network of nodes
- Nodes are working together based on consensus protocols
- Decentralization, public ledger, secure by design

Blockchain key concepts



Source: https://www.economist.com/news/briefing/21677228-technology-behind-bitcoin-lets-people-who-do-not-know-or-trust-each-other-build-dependable

How do we view blockchain in ASE context?

- It is a service → we can use it, e.g., for storing transactions, performing payment, keeping assets, etc., like a blackbox?
- It is a framework → we can use it and plug-in our services in?
- It is a concept or architecture model → so we use it’s model to design new services/businesses?

Blockchain – how do we start?

- Which blockchain systems?
  - Generic frameworks
    - Ethereum? Multichain(https://www.multichain.com/)? Hyperledger
  - Or specific services
    - E.g., https://www.bigchaindb.com/
- Which features of blockchain we need?
  - Walletting? Smart contract?
- Which programming languages to be used?

See the connection with other topics

Example:

https://www.ibm.com/us-en/marketplace/iot-blockchain

Exercises

- Read papers mentioned in slides
- Check services mentioned in examples
- Program some simple sensors sending data to public clouds

Important notes

- 2nd assignment released today
  - with report
- Scenario and service design presentation
  - Provide 2-3 slides (pdf) and present them
- Submission through TUWEL
- Check the exact deadlines from TUWEL

Thanks for your attention

Hong-Linh Truong
Faculty of Informatics, TU Wien
hong-linh.truong@tuwien.ac.at
http://www.infosys.tuwien.ac.at/staff/truong