Fog Computing: Beyond Mobile and Cloud Centric Internet of Things & Collaboration plan

Satish Srirama

satish.srirama@ut.ee

Guest Lecture, Argonne National Laboratory
21st August 2019

Mobile & Cloud Lab

Who am I

- Head of Mobile & Cloud Lab, Institute of Computer Science, University of Tartu, Estonia

http://mc.cs.ut.ee

Mobile & Cloud Lab

Estonia pop: 1,300,000

TARTU

Pop: 100,000

Academic excellence since 1632

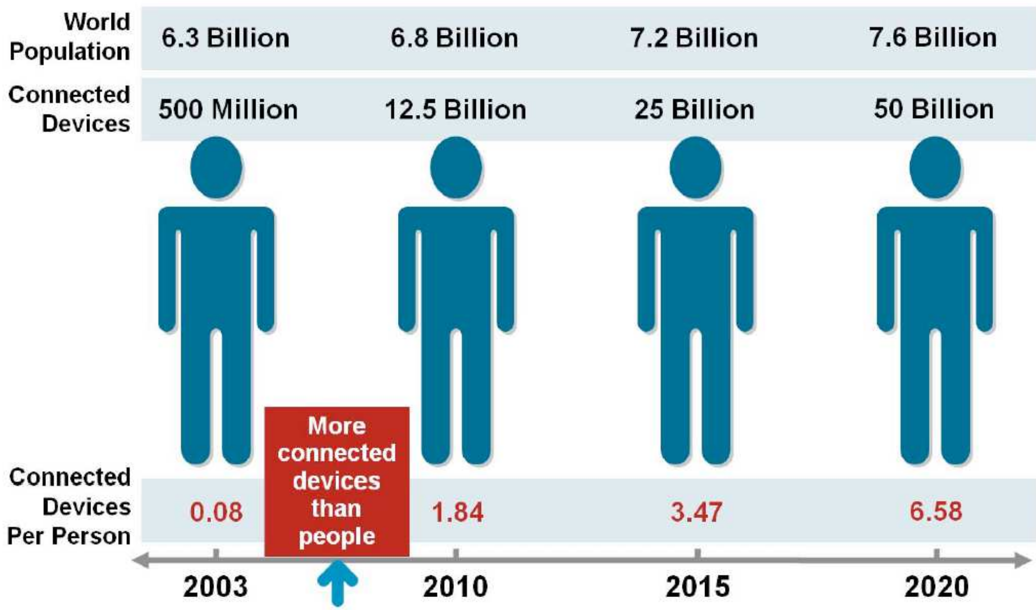
Outline

- Layers of Cloud-based Internet of Things (IoT)
- Mobile Web Services and Cloud Services
- Issues with Cloud-centric IoT
- Fog Computing & Research Roadmap

[Srirama, CSIICT 2017]

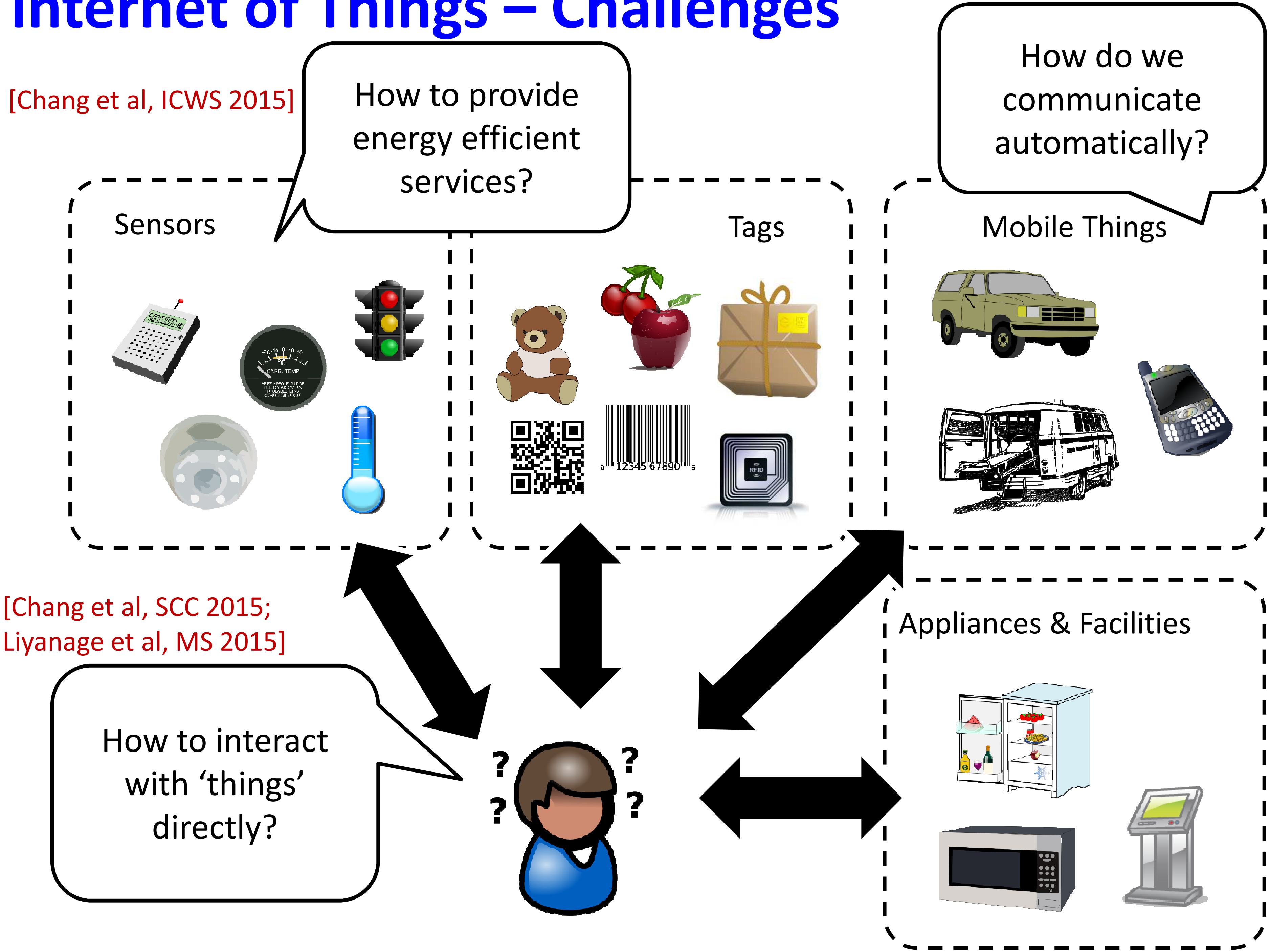
Internet of Things (IoT)

- IoT *allows people and things to be connected*
  - ***Anytime***, ***Anyplace***, *with* ***Anything*** *and* ***Anyone***, *ideally using* ***Any path/network*** *and* ***Any service***
    [European Research Cluster on IoT]
- More connected
- Cisco believes the
  trillion by 2025



Source: Cisco IBSG, April 2011

Internet of Things – Challenges

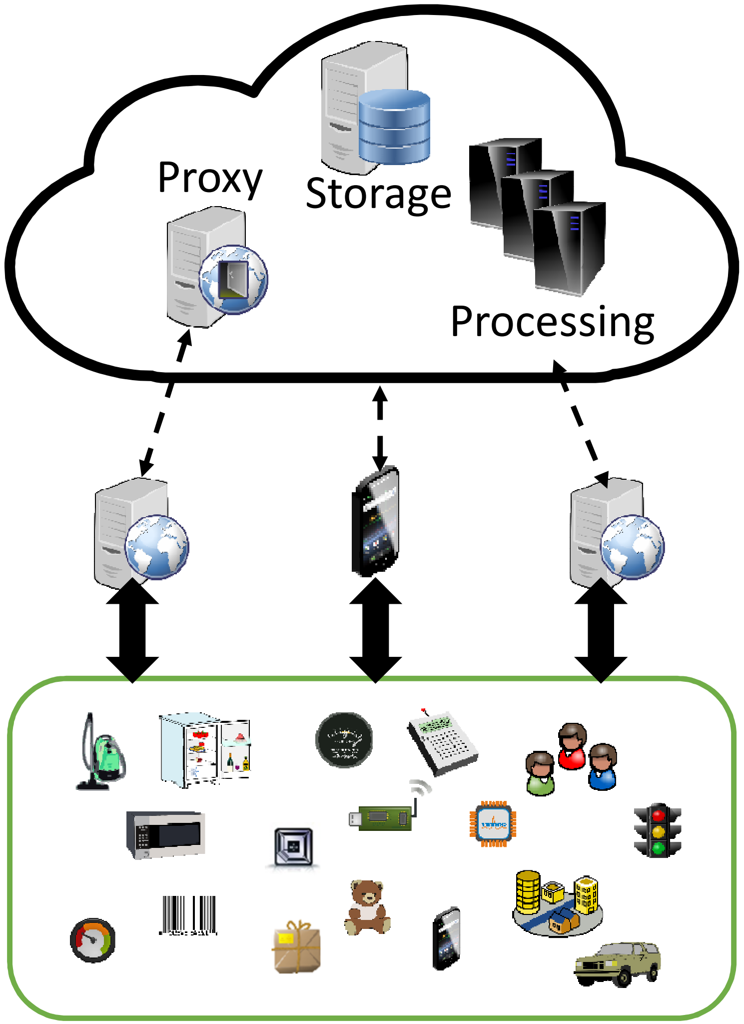


Layers of Cloud-based IoT

Remote Cloud-based processing

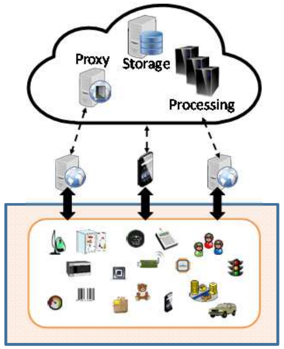
Connectivity nodes & Embedded processing

Sensing and smart devices

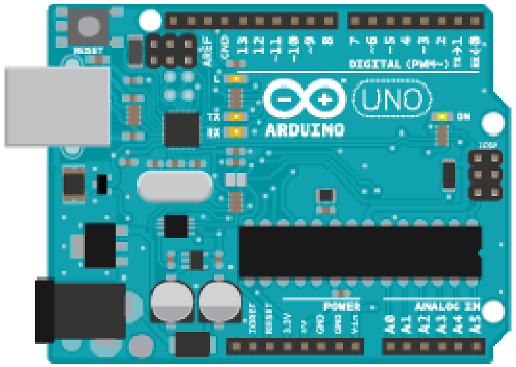


Sensing and Smart Devices

- IoT Devices
  - Sensors and actuators
  - Motion, Temp, Light, Open/Close, Video, Reading, Power on/off/dimm etc.

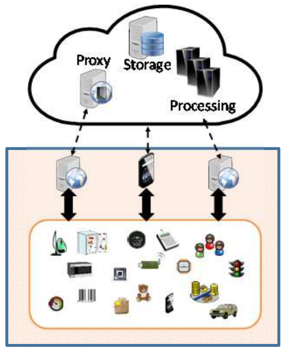


- Communication protocols
  - Wireless and wired
  - Protocols such as ZigBee, Z-Wave, Wi-Fi/Wi-Fi Direct, Bluetooth etc.
- Arduino & Raspberry PI
  - For rapid prototyping



Gateway/Connectivity Nodes

- Primarily deals with the sensor data acquisition and provisioning
- Embedded processing saves the communication latencies
- Predictive analytics
  – Collect data only occasionally
- Mobiles can also participate
  – This brings in the scope of mobile web services and mobile cloud services for IoT



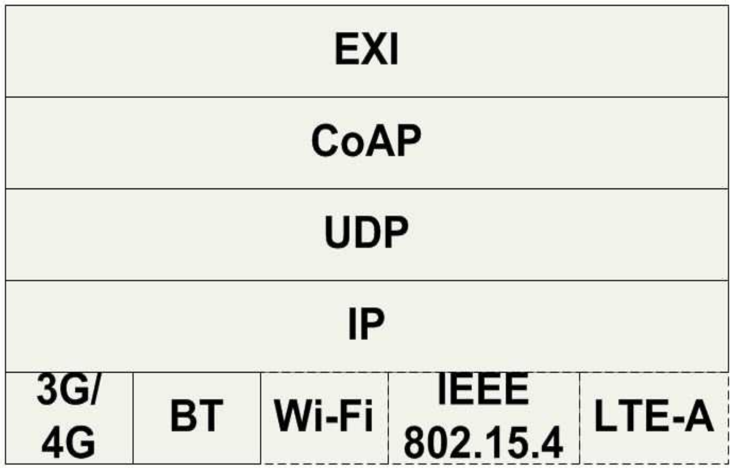
Light-weight Mobile Hosts for Sensor Mediation

- It is possible to provide services from smart phones [Srirama et al, ICIW 2006; Srirama, 2008]

- Mobile Host can directly provide the collected sensor information
  - Data can be collected based on need

- Ideal MWS Protocol Stack
  - Things have improved significantly over the years
  - Bluetooth Low Energy (BTLE) for local service discovery and interaction
  - UDP instead of TCP
  - Constrained Application Protocol (CoAP)
  - Efficient XML Interchange (EXI)

[Liyanage et al, MS 2015]

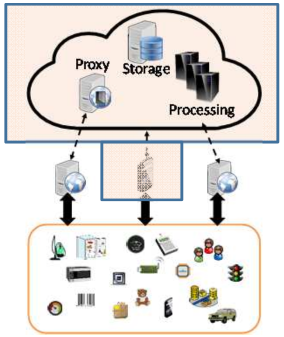


Limitations with Mobiles

- Longer battery life
  - Battery lasts only for 1-2 hours for continuous computing
- Same quality of experience as on desktops
  - Weaker CPU and memory
  - Storage capacity
- Still it is a good idea to take the support of external resources
  - For building resource intensive mobile applications
  - Brings in the scope for cloud computing

Mobile Cloud

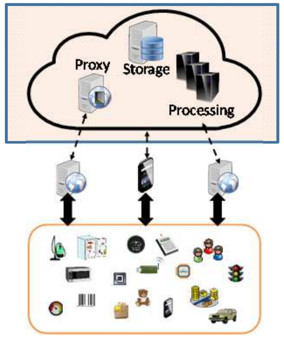
- Harness cloud computing resources from mobile devices



- Binding models
  - Task delegation [Flores and Srirama, JSS 2014]
  - Mobile code offloading [Flores et al, IEEE Communications Mag 2015; Zhou et al, TSC 2017]

- Ideal Mobile Cloud based system should take advantage of some of the key intrinsic characteristics of cloud efficiently
  - Elasticity & AutoScaling
  - Utility computing models
  - Parallelization (e.g., using MapReduce)

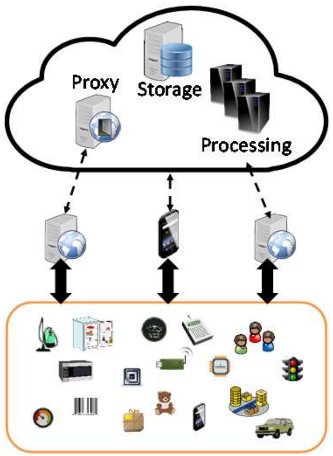
IoT Data Processing on Cloud



- Enormous amounts of unstructured data
  - In Zetabytes (10^{21} bytes) by 2020 [TelecomEngine]
  - Has to be properly stored, analysed and interpreted and presented
- Big data acquisition and analytics
- In addition to big data, IoT mostly deals with big streaming data
  - Message queues such as Apache Kafka to buffer and feed the data into stream processing systems such as Apache Storm
  - Apache Spark streaming

Issues with Cloud-centric IoT

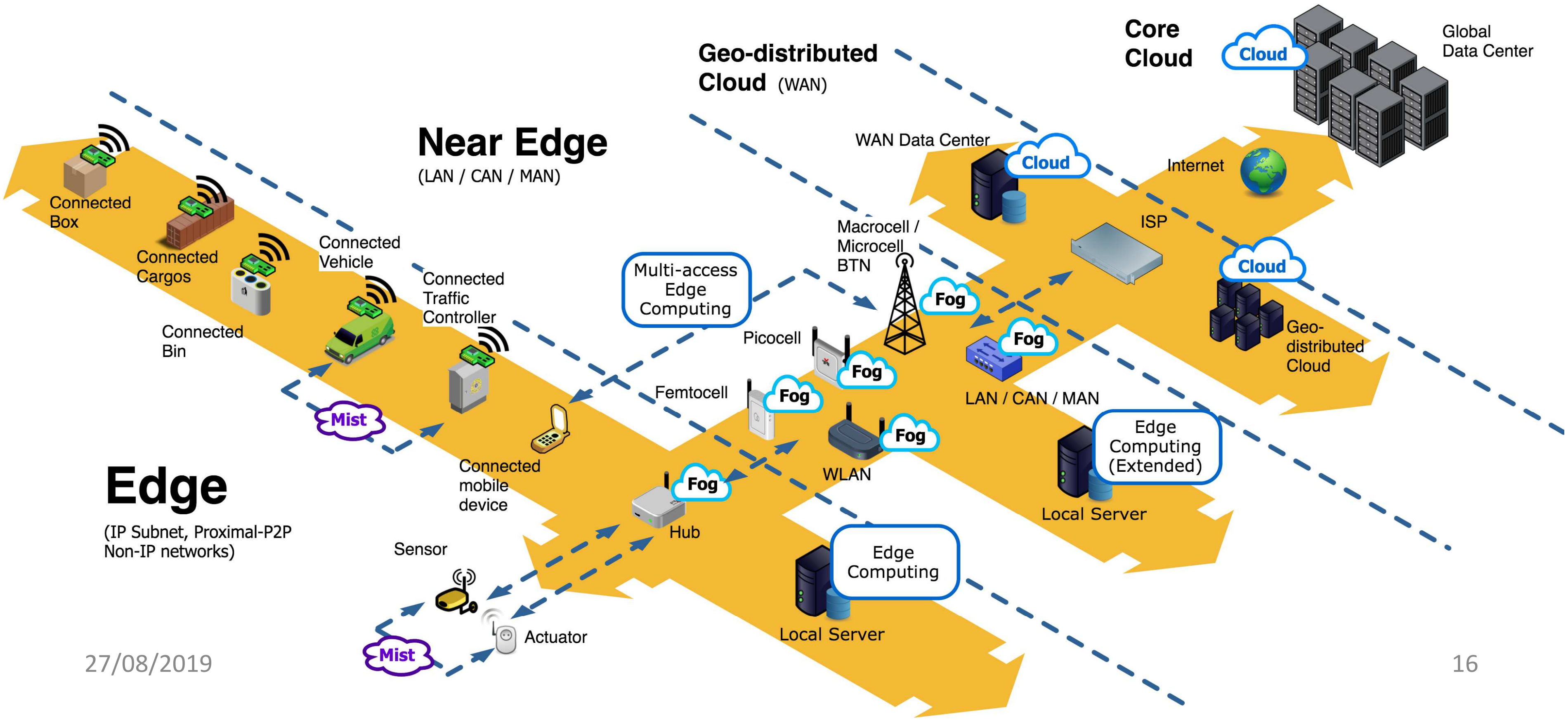
- Latency issues for applications with sub-second response requirements
  - Health care scenarios
  - Smart cities and tasks such as surveillance need real-time analysis with strict deadlines
- Network load
- Certain scenarios do not let the data move to cloud
  - Better security and deeper insights with privacy control



Fog Computing

- Processing across all the layers, including network switches/routers

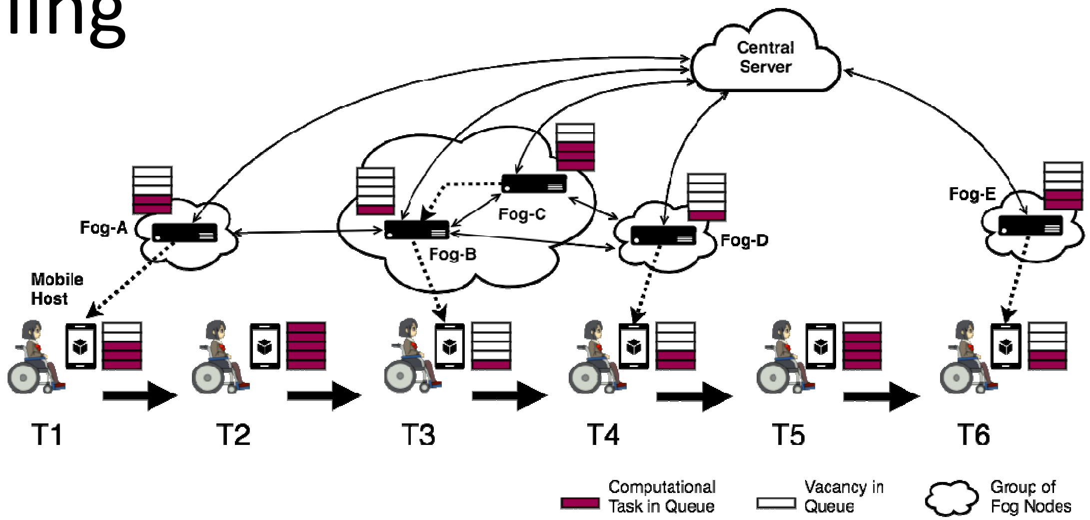
[Chang et al, AINA 2017; FEC 2019; Mass et al, SCC 2016; Liyanage et al, PDCAT 2016]



Fog Computing – Research Challenges

- Proactive Fog computing using resource-aware work-stealing

[Soo et al, IJMCMC 2017]



- Indie Fog [Chang et al, IEEE Computer 2017]
  - System architecture for enabling Fog computing with customer premise equipment

Fog Computing – Research Challenges - continued

- Dynamic Fog computing service discovery and accessing
- Distributed and fault-tolerant execution of Fog computing applications
  – Based on Actor programming model
  – Have implemented applications using the Akka framework

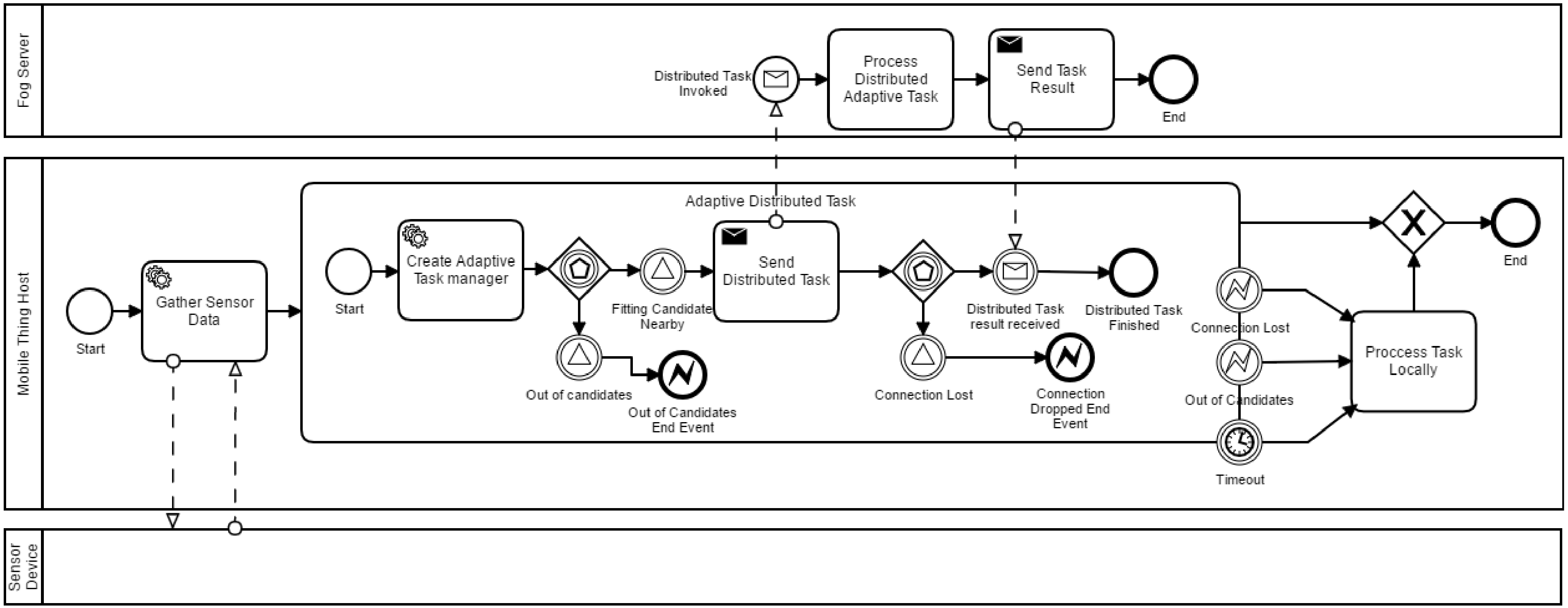
Fog Computing – Research Challenges - continued

- QoS & QoE-aware application placement across Fog topology [Mahmud et al, JPDC 2019]
  - Resource intensive tasks of IoT applications can be placed across the Fog topology
  - Latency-aware application module management
- The problem can also be formulated as multi-objective offloading strategy
  - Latency, energy-efficiency and resource management
  - Need to find ideal heuristics, metaheuristics etc.
  - Also have to consider the graph topology of the Fog nodes

QoS – Quality of Service
QoE – Quality of Experience

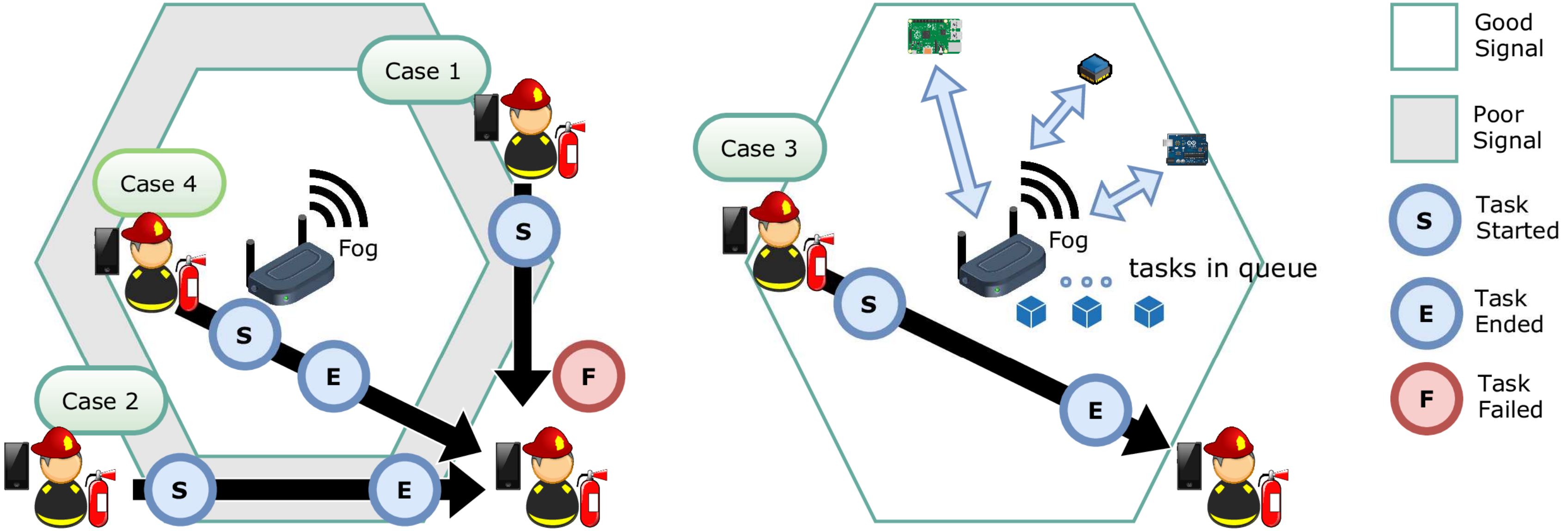
Fog Computing – Research Challenges - continued

- Process-driven Edge Computing in Mobile IoT
  [Mass et al, IoTJ 2019; CASA 2018; Chang et al, CSUR 2016]



Fog Computing – Research Challenges - continued

- Mobility also becomes critical in Fog computing [Mass et al, IoTJ 2019]

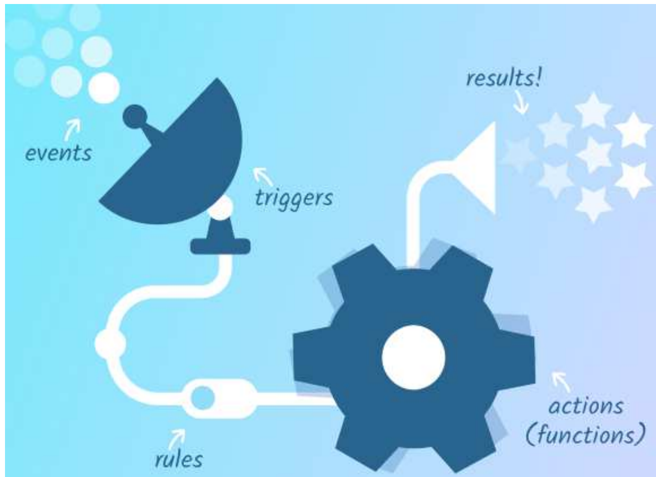


- STEP-ONE : Simulated Testbed for Edge Processes based on the Opportunistic Network Emulator
  - Extended the ONE simulator to simulate the Fog computing mobility aspects
  - Process execution based on Flowable BPMS

Serverless computing

- IoT workloads are a better fit for event driven programming
  - Execute app logic in response to sensor data
  - Similar tasks
    - Execute application logic in response to database triggers
    - Execute app logic in response to scheduled tasks etc.
- Serverless
  - Event-action platforms to execute code in response to events
  - Applications are charged by compute time (millisecond) rather than by reserved resources
- Serverless computing is ideal solution for fog processing
  - OpenFaaS, light-weight enough to place on Raspberry Pi

OPENFAAS



EU H2020 -RADON

- Rational decomposition and orchestration for serverless computing
  - Jan 2019 – Jun 2021
- Goal
  - Creating a DevOps framework to create and manage microservices-based applications
  - Tools that facilitate in designing and orchestrating data pipeline applications that involve serverless entities
  - OASIS - Topology and Orchestration Specification for Cloud Applications (TOSCA)
- Case studies
  - IoT application from healthcare
  - Tourism

Research Roadmap – IoT & Fog Computing

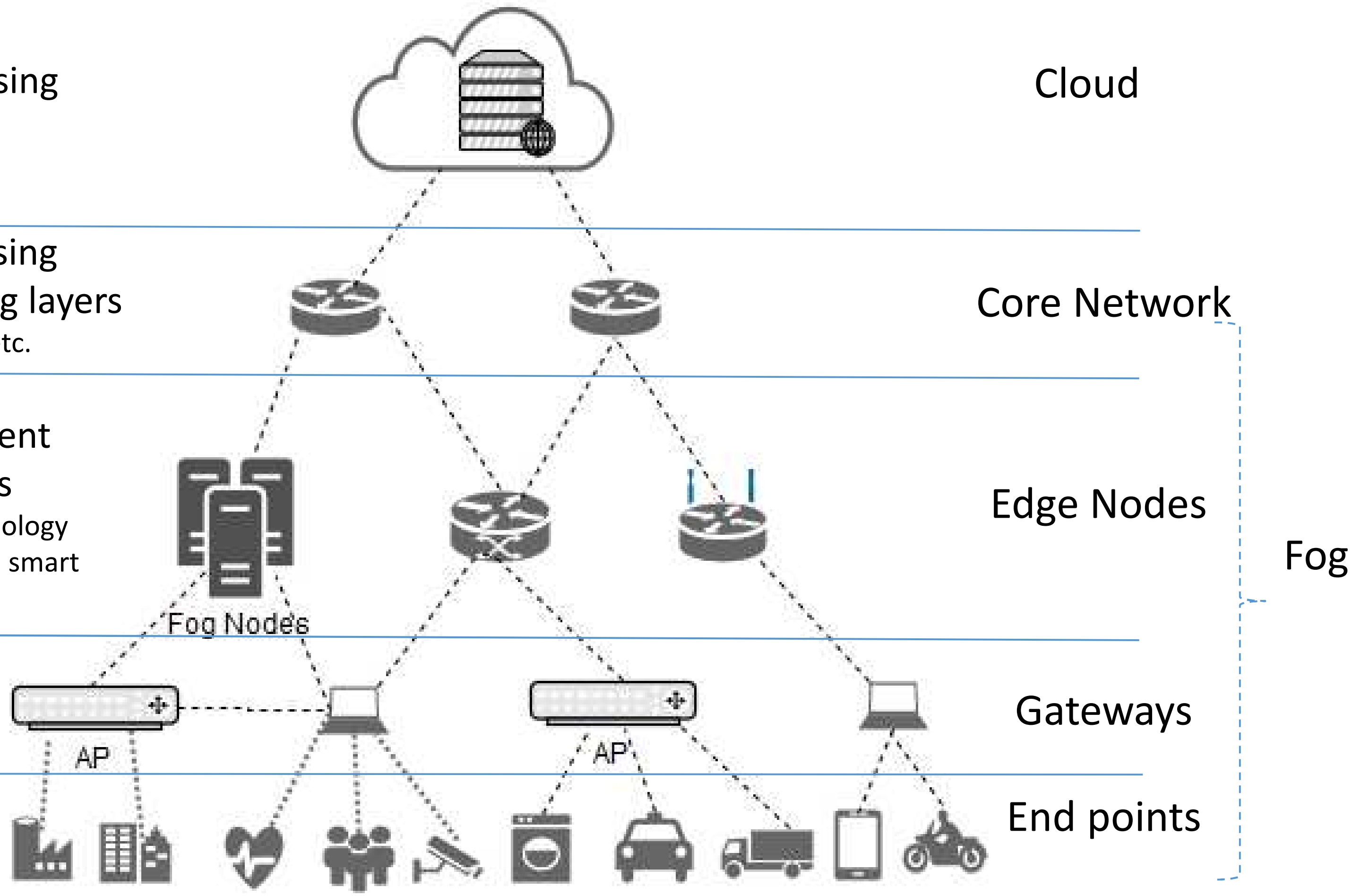
Distributed data processing on the Cloud
E.g. MapReduce, Spark

Distributed data processing across the Cloud and Fog layers
E.g. Personalized data, privacy etc.

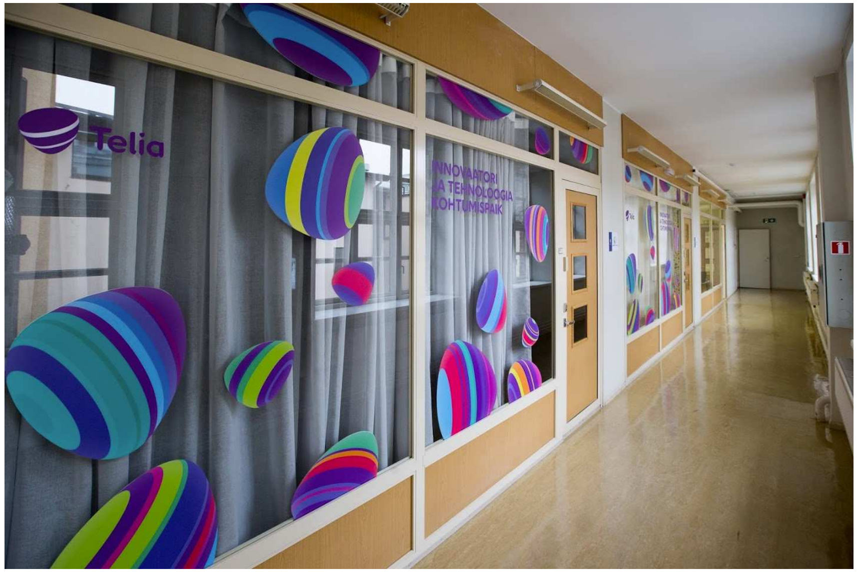
Fog topology management and scheduling the tasks
E.g. tasks run across the fog topology such as stream data processing, smart streetlights etc.

Edge analytics
E.g. filter, error detection, consolidation etc.

Intelligent sensors
E.g. vehicular networks



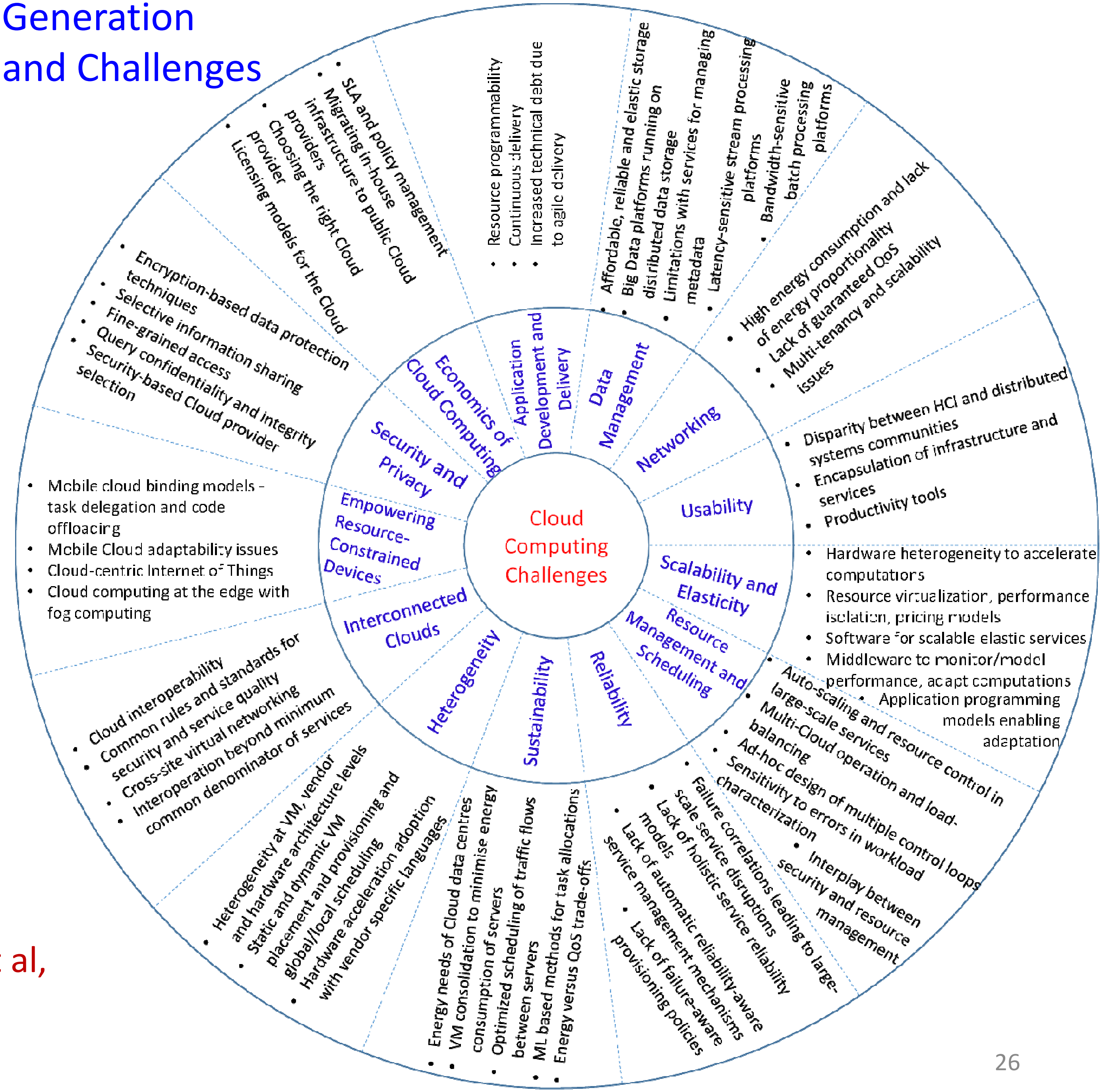
IoT and Smart Solutions Laboratory



Mobile & Cloud Lab

Telia

A Manifesto for Future Generation Cloud Computing: SOA and Challenges



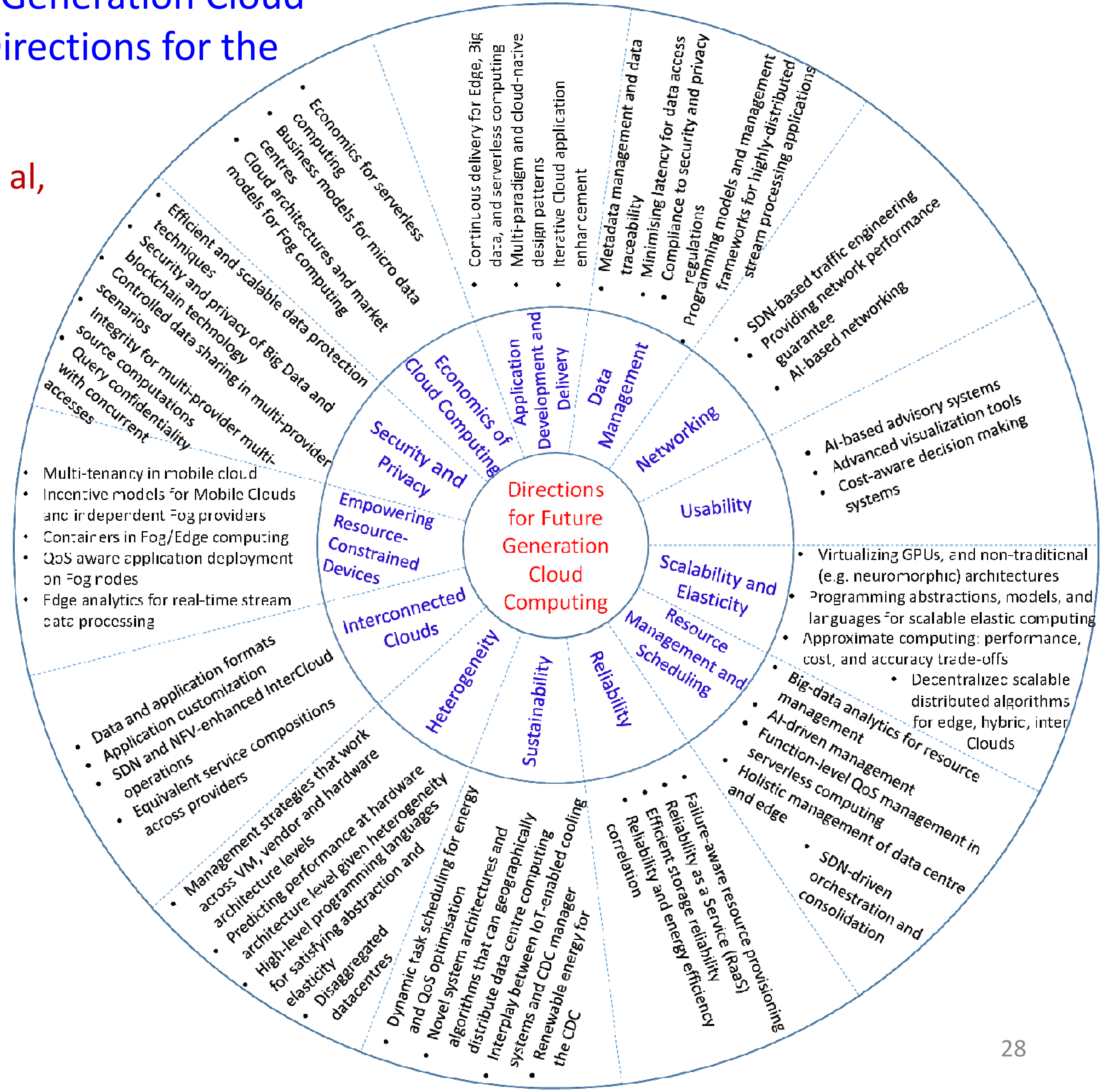
[Buyya, Srirama, Casale et al, ACM CSUR 2019]

Emerging trends and impact areas for cloud

- Containers
- Fog Computing
- Big Data
- Serverless Computing
- Software-defined Cloud Computing
- Blockchain
- Machine and Deep Learning

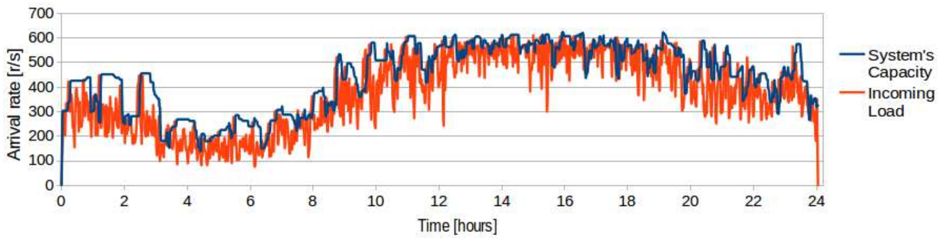
A Manifesto for Future Generation Cloud Computing: Research Directions for the Next Decade

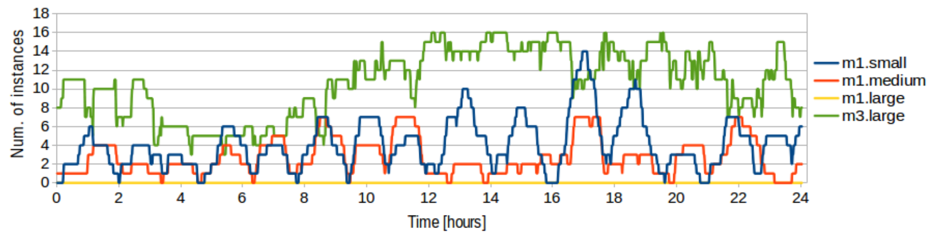
[Buyya, Srirama, Casale et al, ACM CSUR 2019]



Other cloud related research interests

- Dynamic deployment of applications on cloud
  - Standardization efforts from CloudML
    [REMICS EU FP7; MODAClouds EU FP7; Srirama et al, Cloud 2016]
  - TOSCA and extensions for serverless [RADON EU H2020]
- Auto-scaling & Resource provisioning
  - Taking advantage of cloud heterogeneity
  - Cloud cost models of fine-grained billing (e.g. hourly) [Srirama and Ostovar, CloudCom 2014; IJCC 2018]



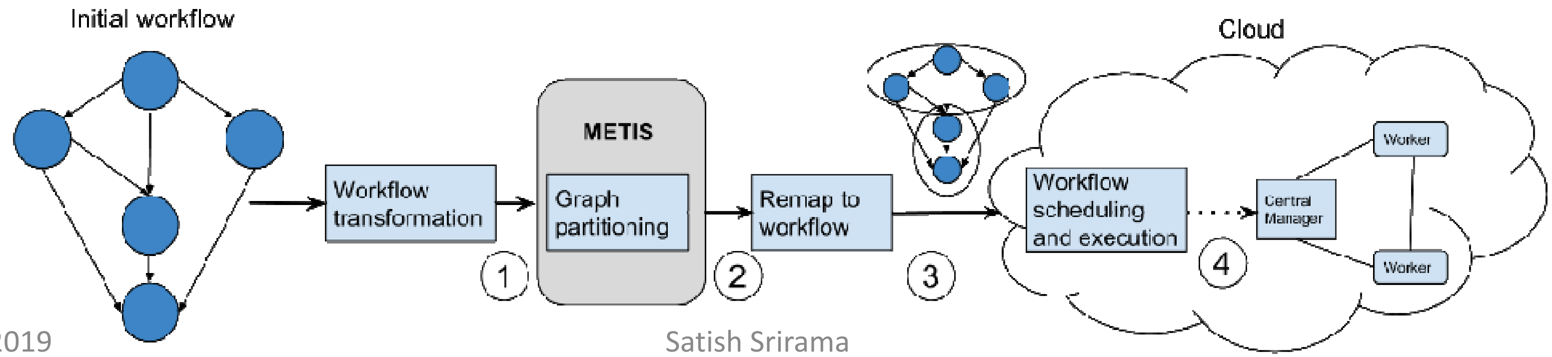


Data analytics on the cloud

- Adapting data analytics problems to cloud using MapReduce
- Designed a classification on how the algorithms can be adapted to MapReduce [Srirama et al, FGCS 2012]
  - MapReduce not ideal for iterative algorithms
  - Applicable especially for Hadoop MapReduce
- Alternative MapReduce implementations that are designed to handle iterative algorithms [Jakovits and Srirama, HPCS 2014]
  - E.g. Twister, HaLoop, Spark

Migrating Scientific Workflows to the Cloud [Srirama and Viil, HPCC 2014]

- Workflow can be represented as weighted directed acyclic graph (DAG)
- Partitioning the workflow into groups with graph partitioning techniques
  - Such that the sum of the weights of the edges connecting to vertices in different groups is minimized
  - Utilized Metis' multilevel k-way partitioning

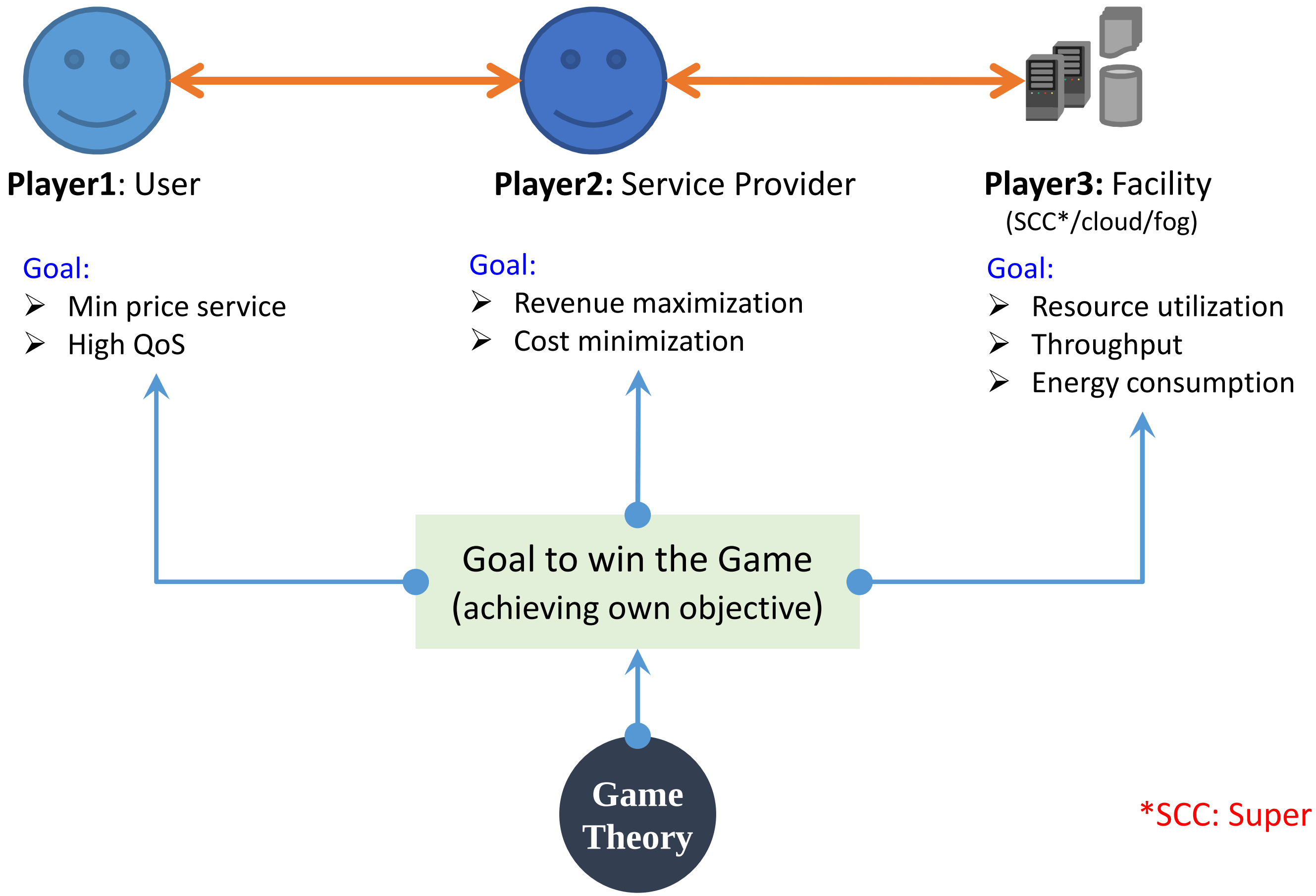


Migrating Scientific Workflows to the Cloud - continued

- Scheduling the workflows with tools like Pegasus
  - Considered peer-to-peer file manager (Mule) for Pegasus
- Framework for Automated Partitioning and Execution of Scientific Workflows in the Cloud [Viil and Srirama, JSC 2018]
  - Includes auto-scaling and dynamic deployment with CloudML

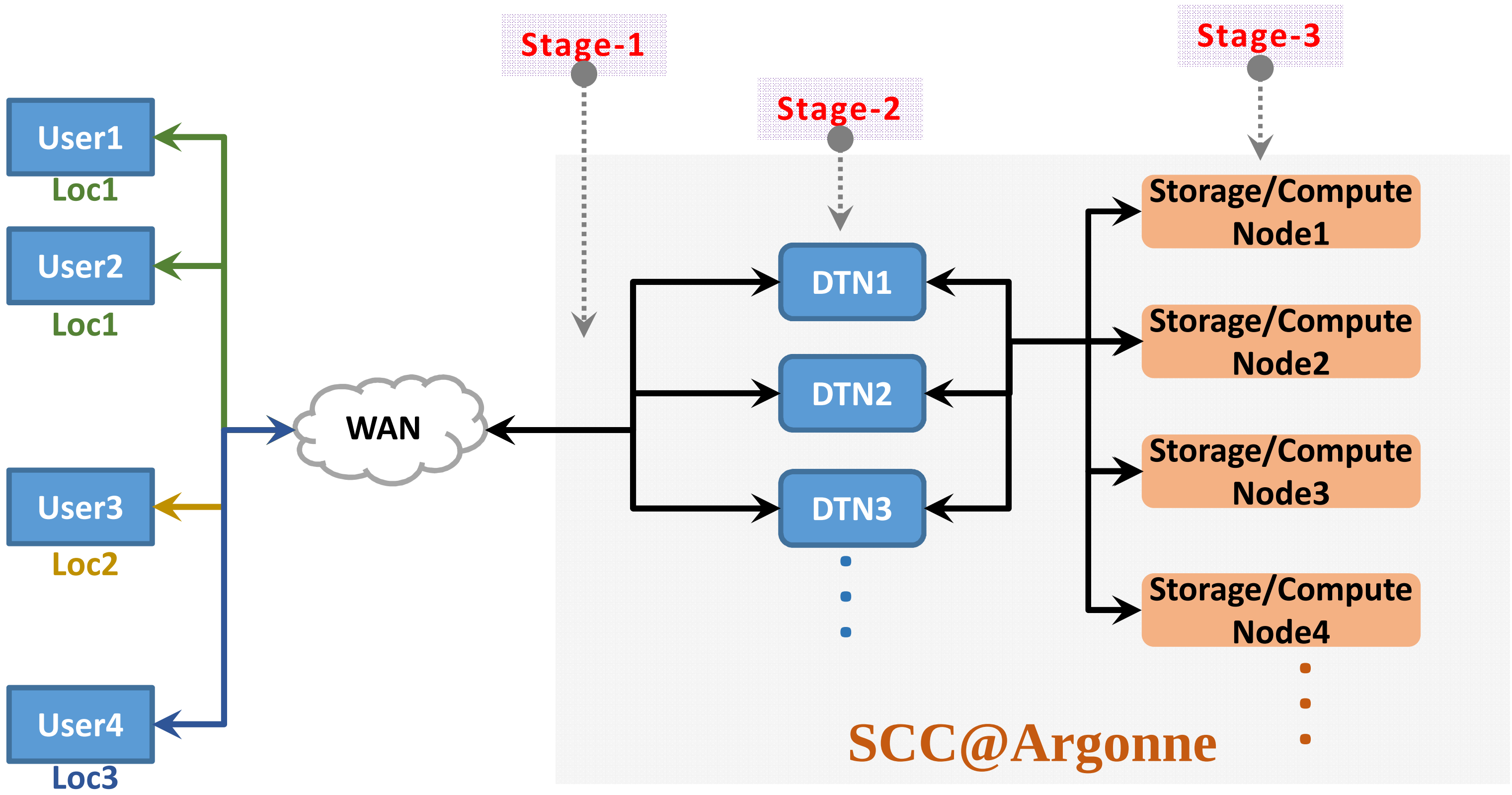
COLLABORATION PLAN

Plan 1: Towards achieving multi-players objective using Game theory



*SCC: Super Computer Center

Plan 2: On optimizing the throughput and resource utilization using machine learning approach



Plan 2 - Continued

Stage-1

- Predict *S1,* the amount of data SCC may receive in next 30min.
- E.g. Let S1 = 30GB. This mean SCC may receive 30GB of data from external users in next 30 min.
- Features from the dataset:
  - Start and end time of transfer
  - Average transfer rate
  - Number of file and directory transferred
  - Total number of bytes transferred
  - Etc.

Stage-2

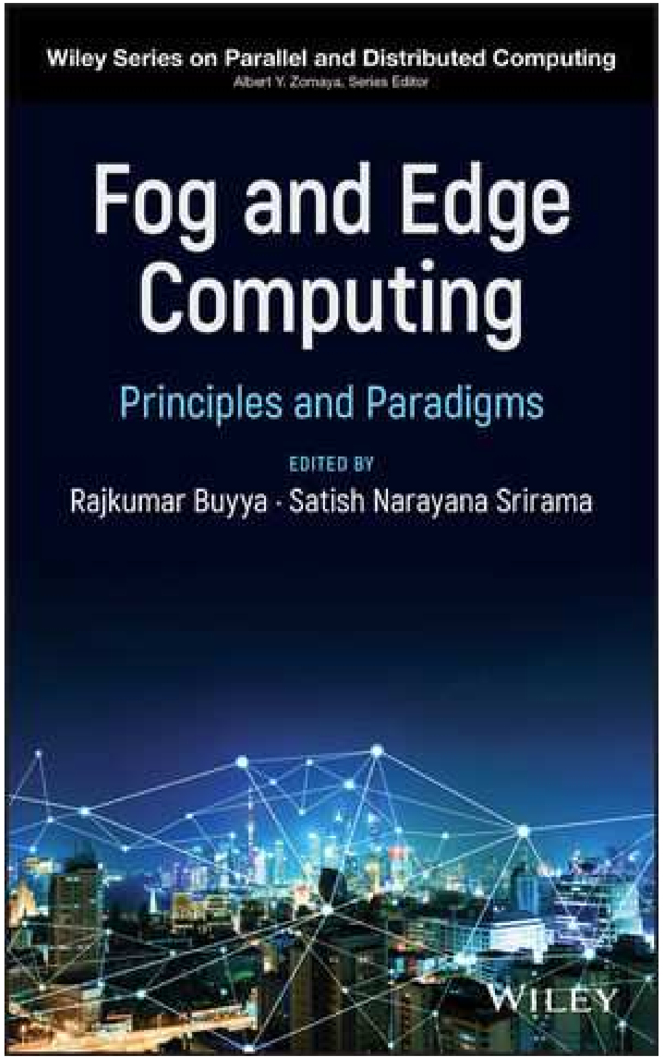
- Input to this stage is *S1*.
- Predict the class of data
- Class of data can be with respect to a specific application, or a source institution
- Based on the class of data, a portion or full amount of required resource can be pre-allocated
- Features from the dataset:
  - Source endpoint
  - Institution of source endpoint of a transfer
  - Start and end time of transfer
  - Total number of bytes transferred
  - Purpose of data (for what purpose/application the data transferred)
  - Etc.

Stage-3

- Predict *S3*, the amount of data that will be generated by a specific application/task, e.g. in next 30 min.
- Based on this prediction, a portion or full amount of required resource can be pre-allocated
- Features from the dataset:
  - Historical information of a specific application

European Commission

Eesti Teadusagentuur
Estonian Research Council



srirama@ut.ee

THANK YOU FOR YOUR ATTENTION