On Research Challenges in IoT Systems Engineering

13 November 2018, ICSOC 2018, Honghzou, China

Schahram Dustdar

Distributed Systems Group
TU Wien

dsg.tuwien.ac.at

What exactly is the

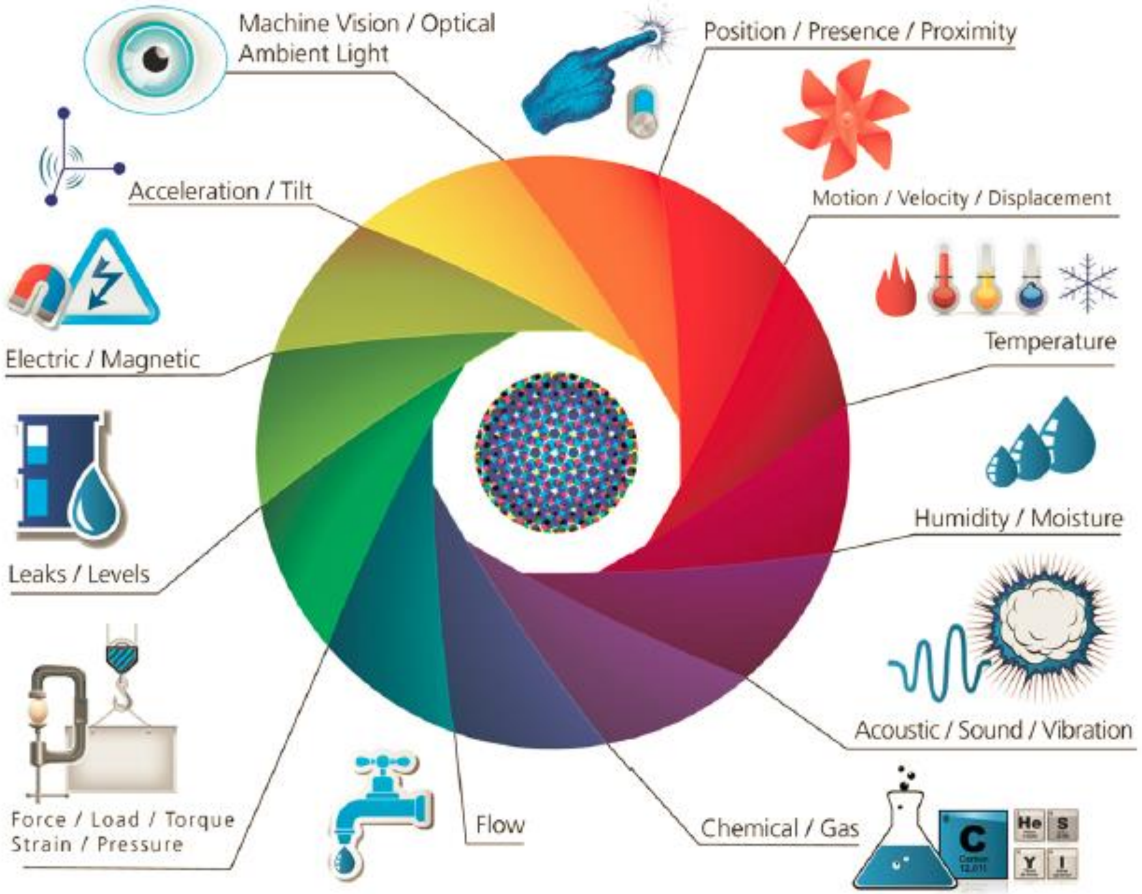
"INTERNET of THINGS"?

Smart Systems and the Internet of Things are driven by a combination of:

1. **SENSORS** & ACTUATORS
2. **CONNECTIVITY**
3. **PEOPLE & PROCESSES**

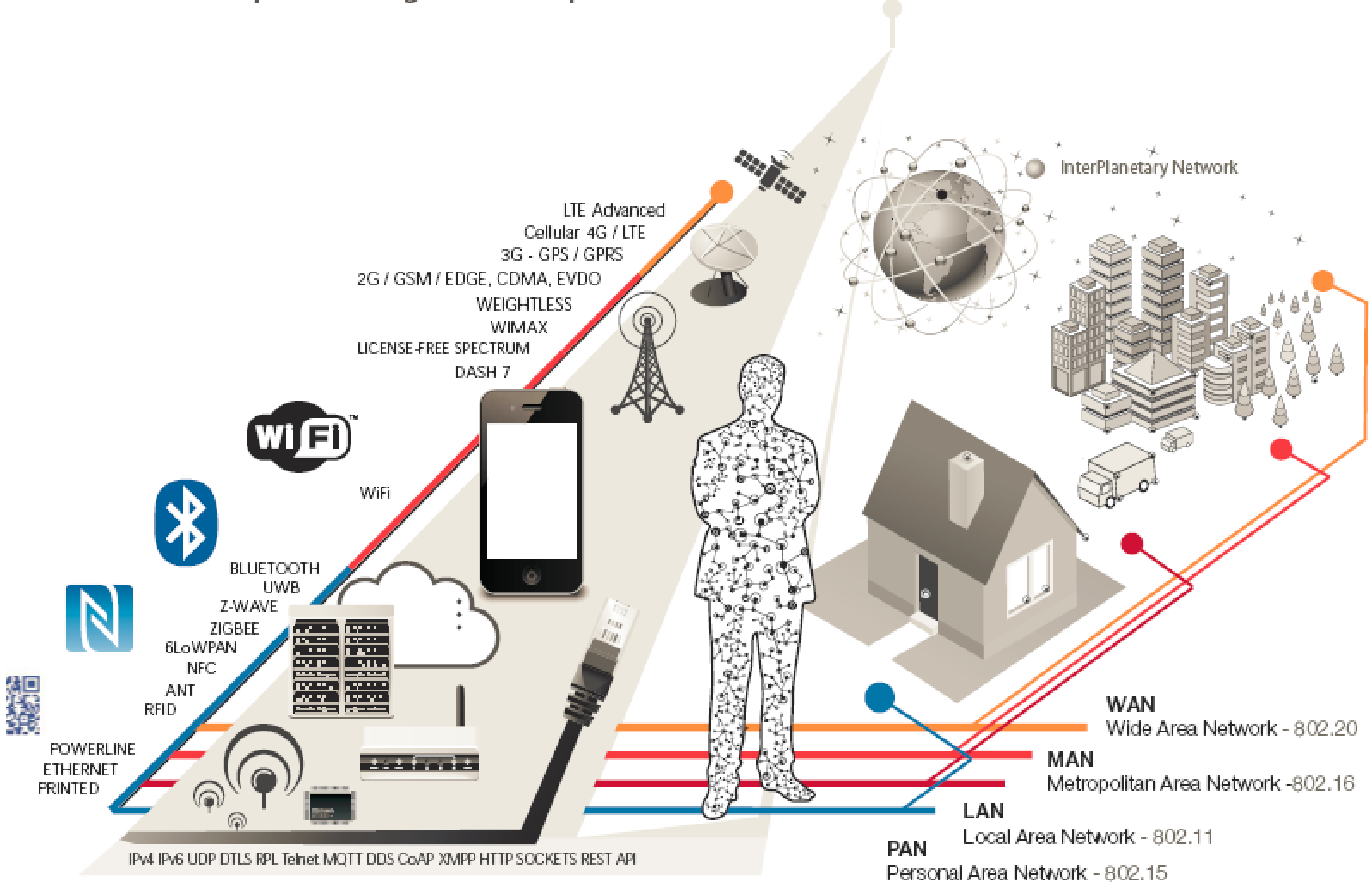
1 SENSORS & *ACTUATORS*

We are giving our world a digital nervous system. Location data using GPS sensors. Eyes and ears using cameras and microphones, along with sensory organs that can measure everything from temperature to pressure changes.



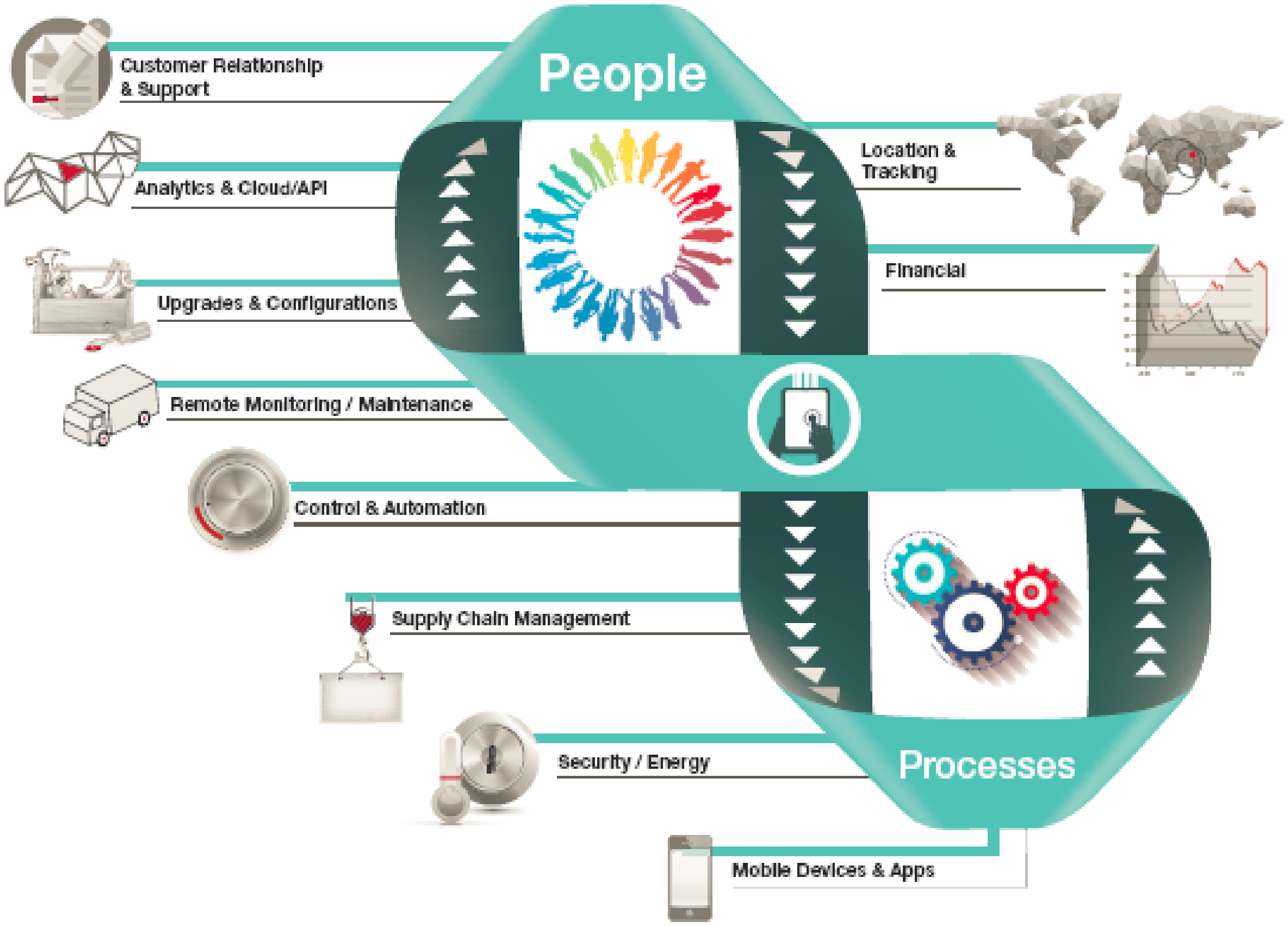
2 CONNECTIVITY

These inputs are digitized and placed onto networks.



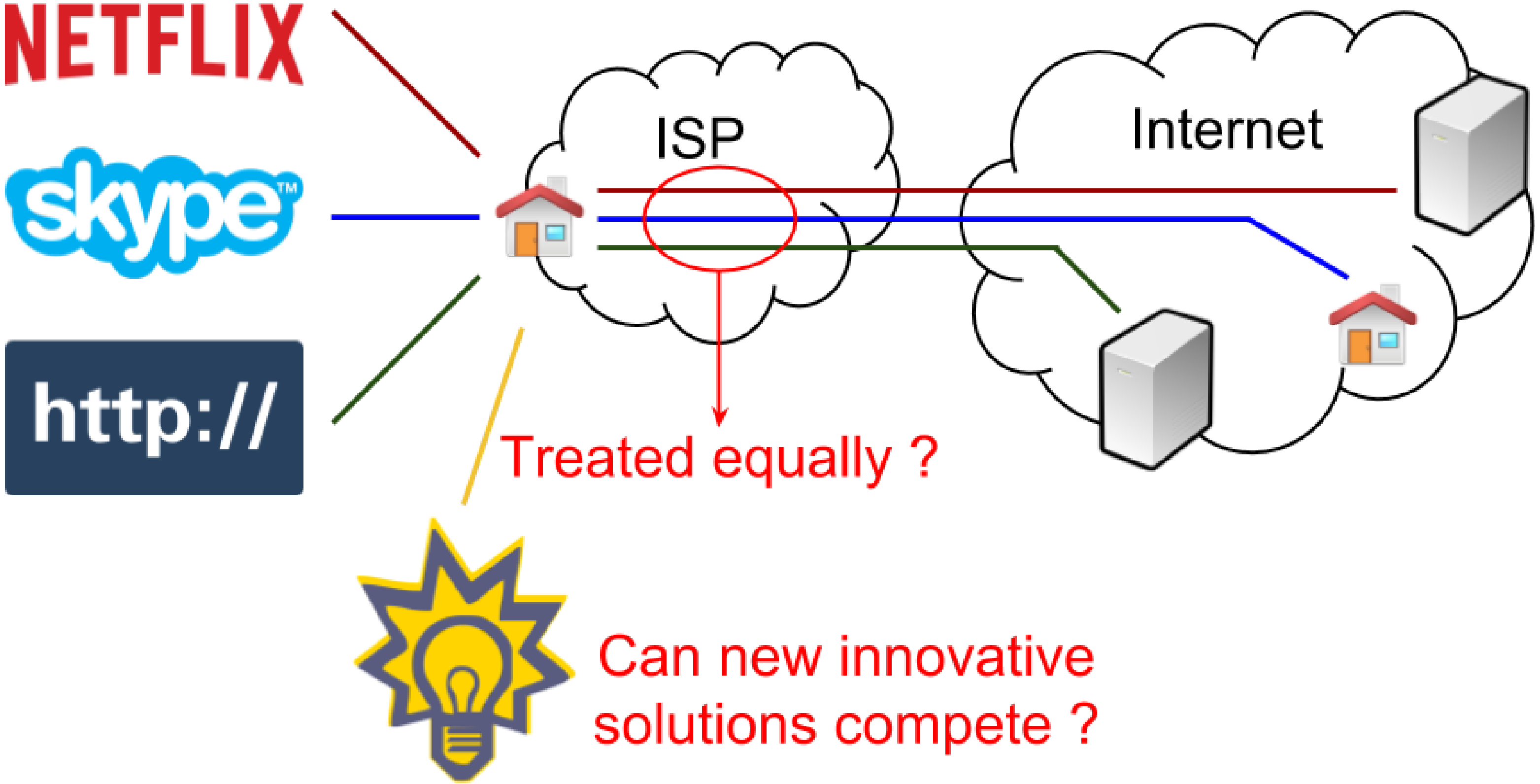
3 PEOPLE & *PROCESSES*

These networked inputs can then be combined into bi-directional systems that integrate data, people, processes and systems for better decision making.



Network Neutrality (NN)

All traffic on the Internet must be treated equally.



Network Neutrality is gone…

T-Mobile Germany Blocks iPhone Skype Over 3G and WiFi

James Kendrick Apr 6, 2009 - 5:15 PM CDT

19 Comments

Tweet Share Post

Skype has been one of the top downloaded apps for the iPhone since its release last week, even though Apple (s aapl) bowed to AT&T (s T) in the U.S. to prevent the VoIP program from working on 3G. U Skype users are re on the iPhone, som doubt concerned t whatever they can

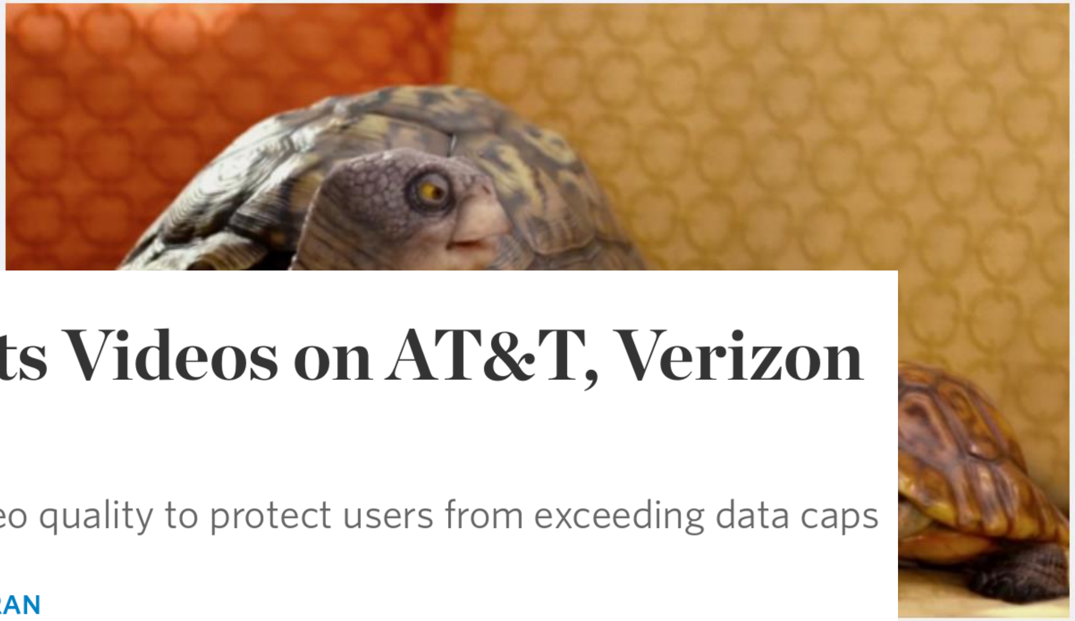
sky

TECHNOLOGY LAB —

Netflix performance on Verizon and Comcast has been dropping for months

Latest Netflix data shows some ISPs struggling, while Google Fiber soars.

JON BRODKIN - 2/10/2014, 10:30 PM



TECH

Netflix Throttles Its Videos on AT&T, Verizon Networks

Streaming service says it limits video quality to protect users from exceeding data caps

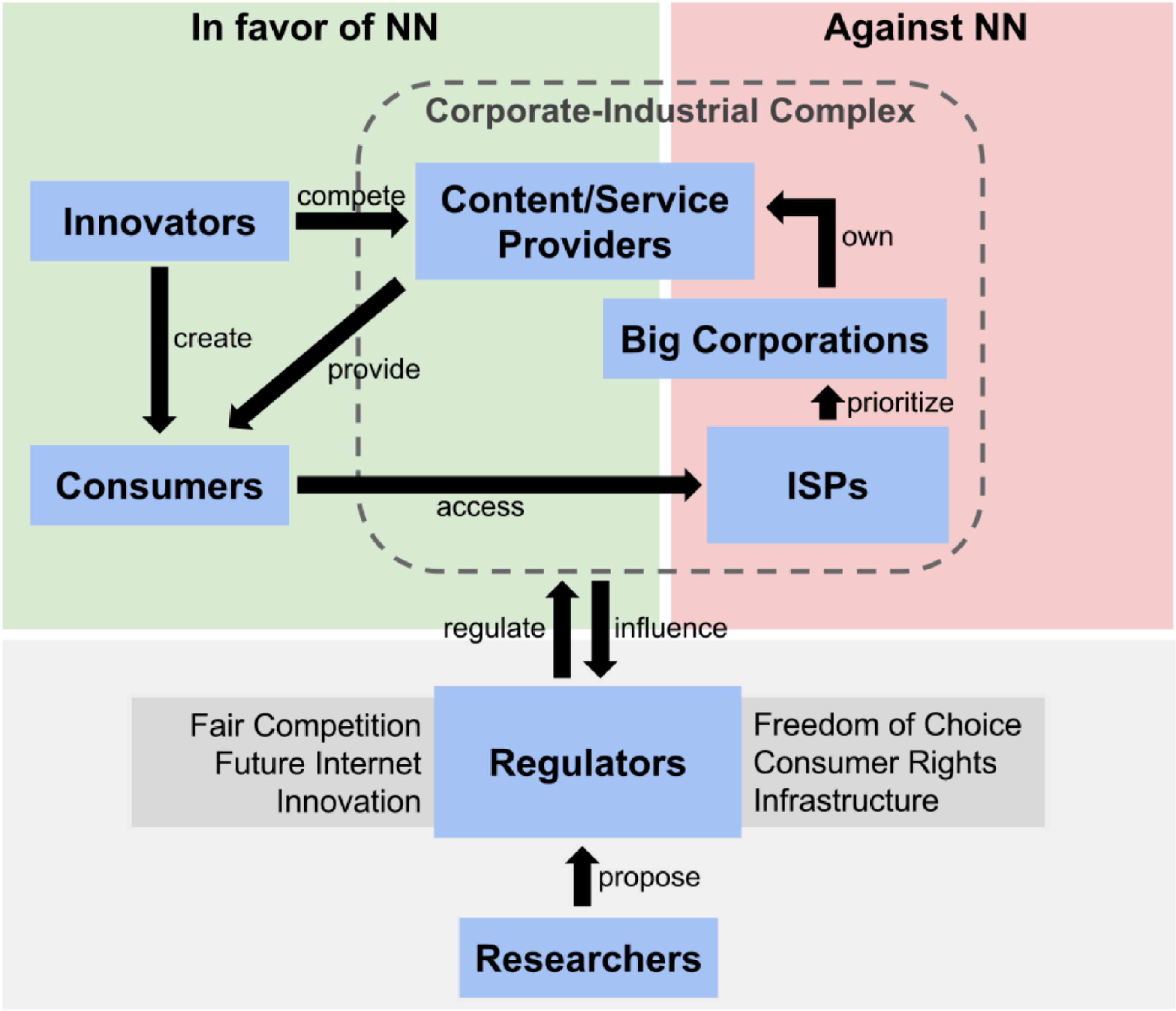
By RYAN KNUTSON and SHALINI RAMACHANDRAN

Updated March 24, 2016 10:55 p.m. ET

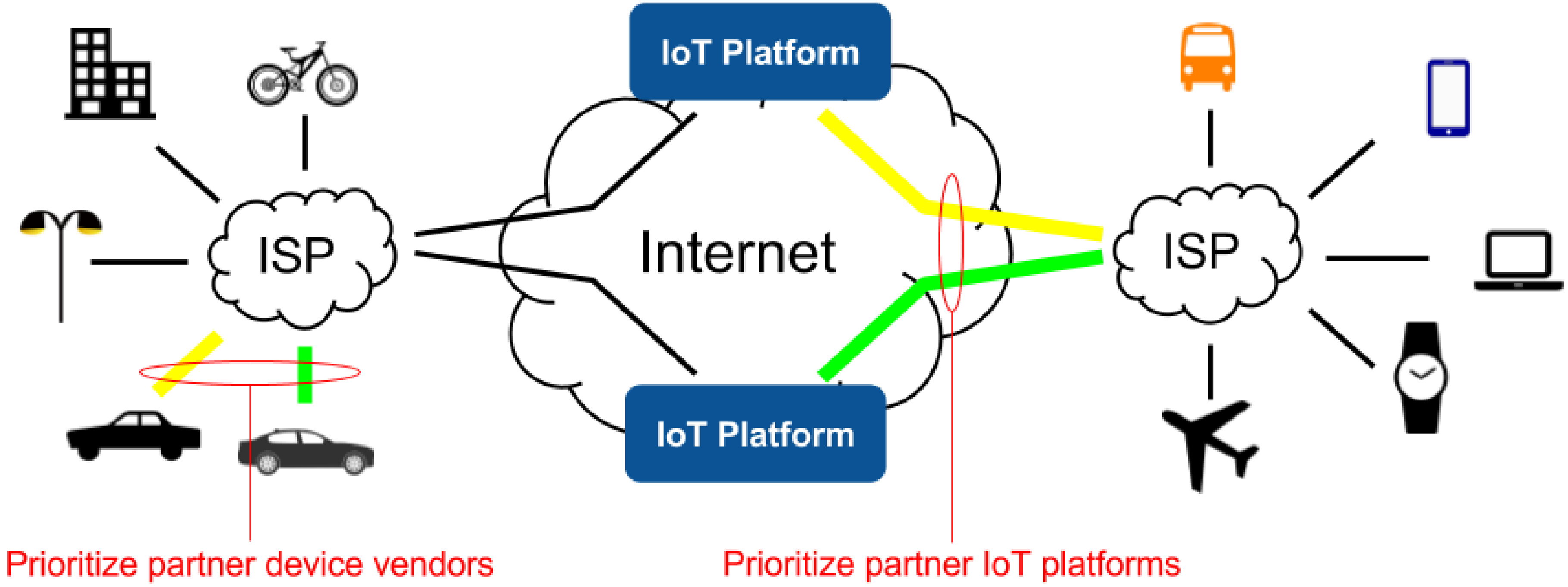
AT&T Inc. and Verizon Communications Inc. were on the defensive last week after accusations swirled they were throttling the quality of Netflix Inc. video on their wireless networks.

It turns out it was Netflix that was doing the throttling.

Understanding the context

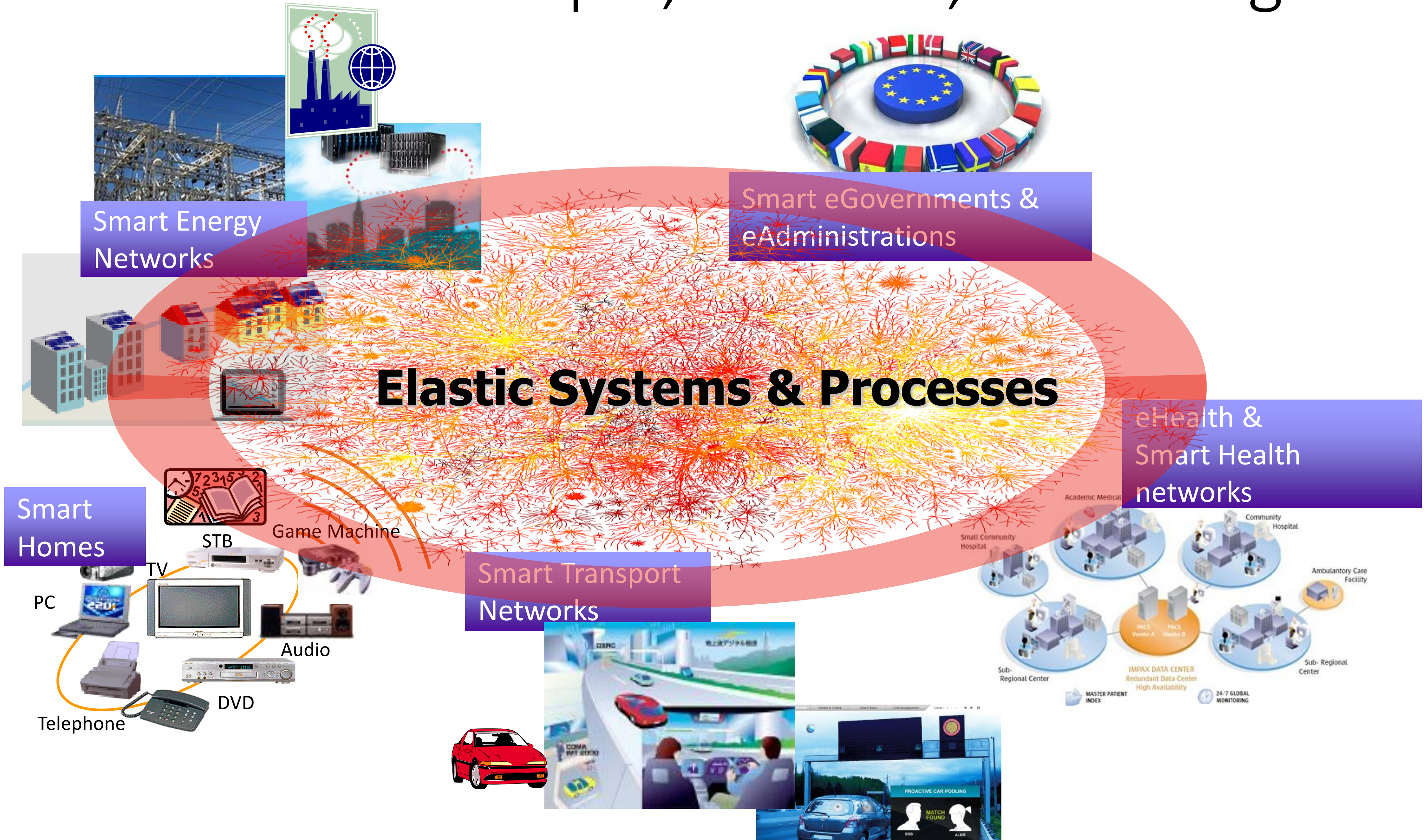


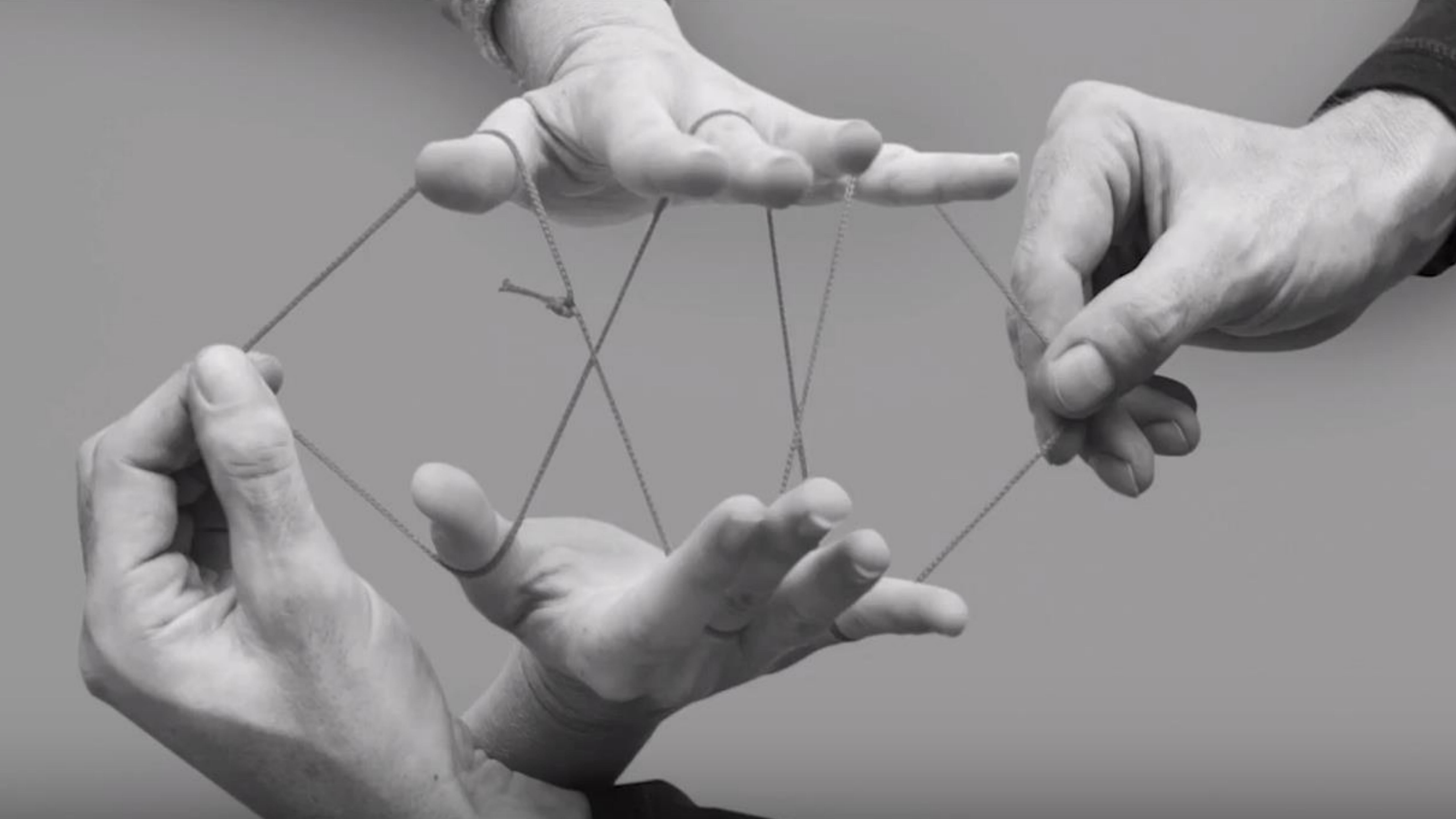
What about NN and IoT?

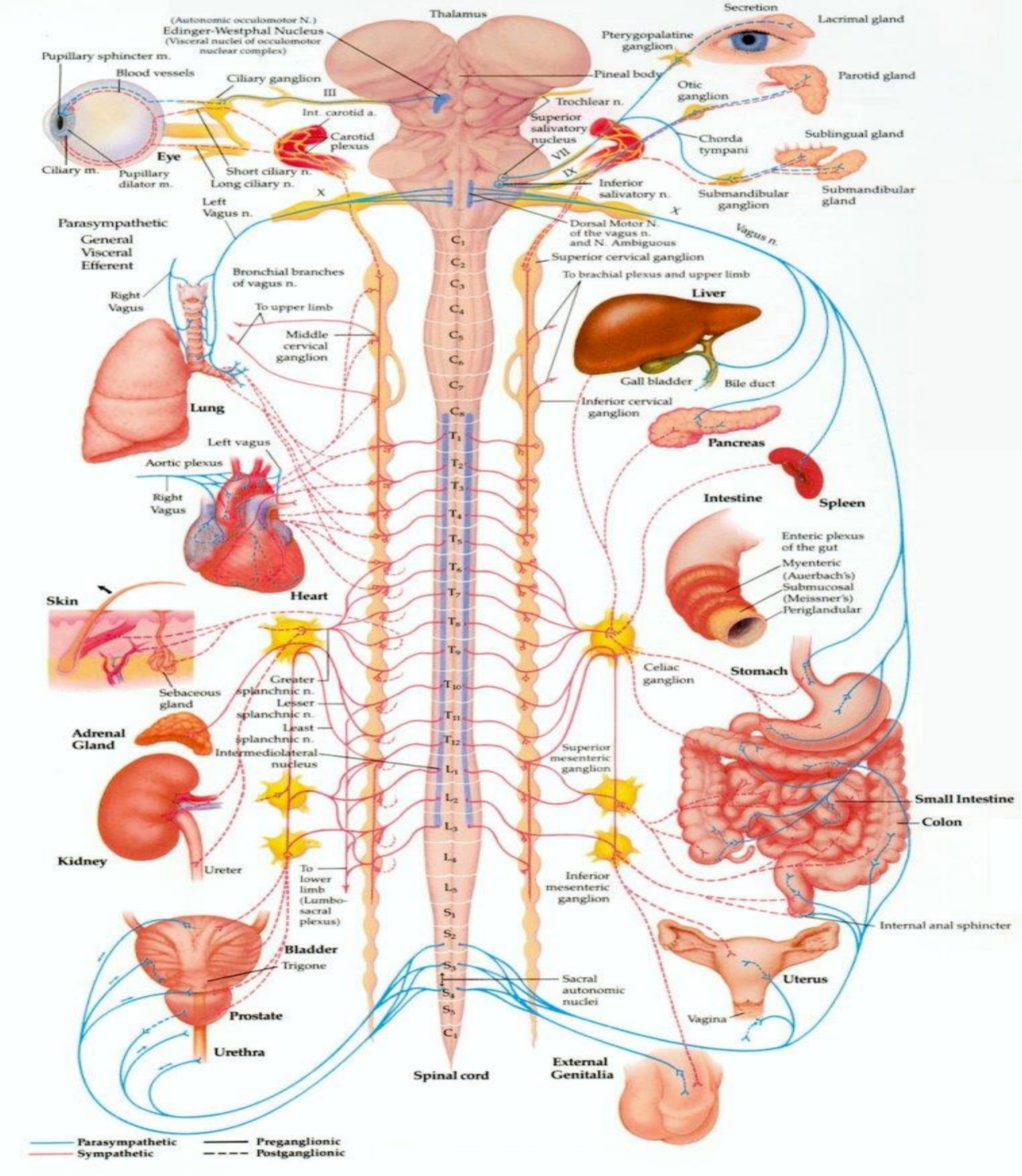


Dustdar S. and Duarte, E.. (2018) Network Neutrality and its impact on Innovation, IEEE Internet Computing, Nov/Dec 2018

Smart Evolution – People, Services, and Things







Autonomic Nervous System

Design Aspects

- Divide & Conquer
- Complexity, Coherence & Entropy

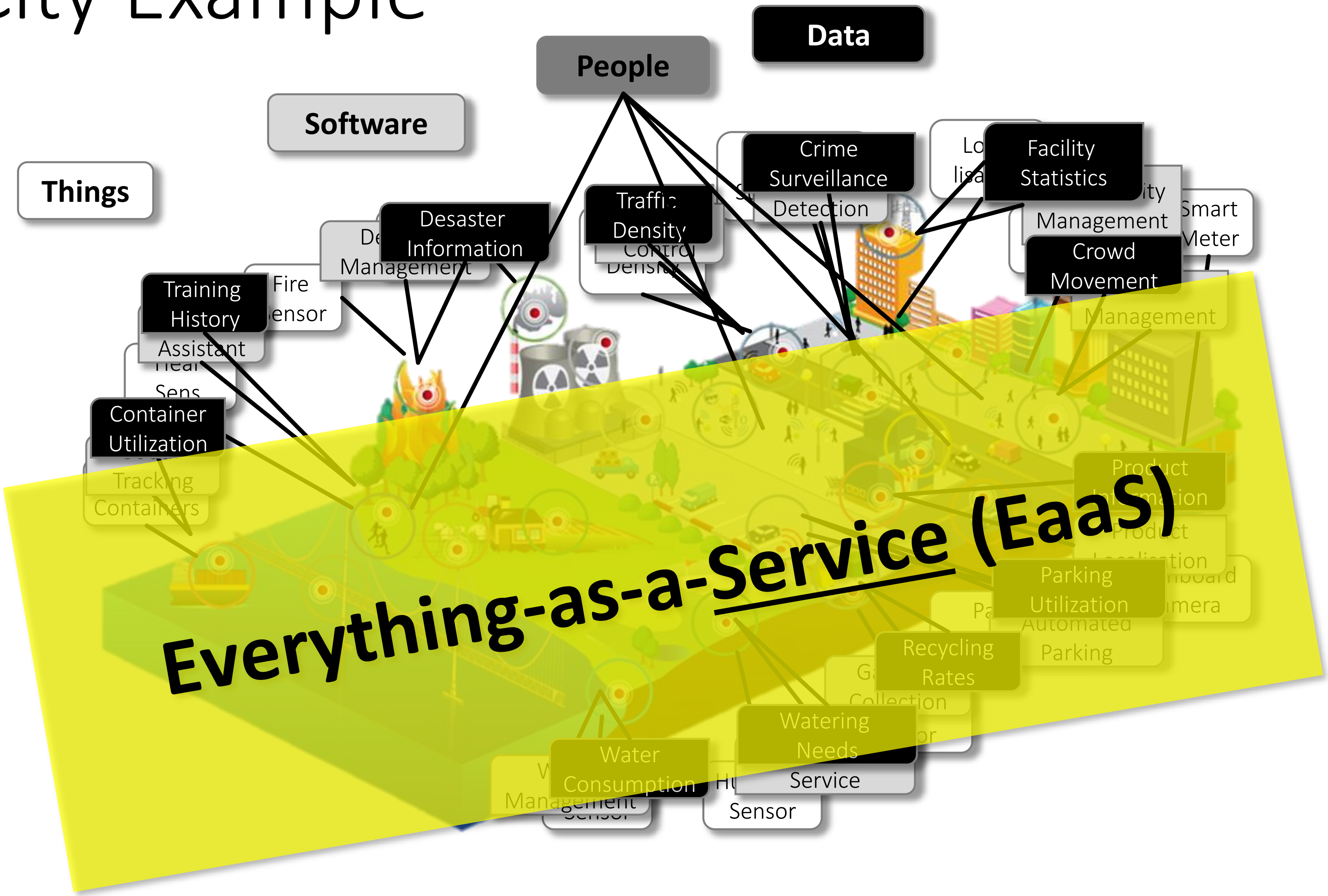
Service Engineering Design Strategy

Definition of DIVIDE AND CONQUER

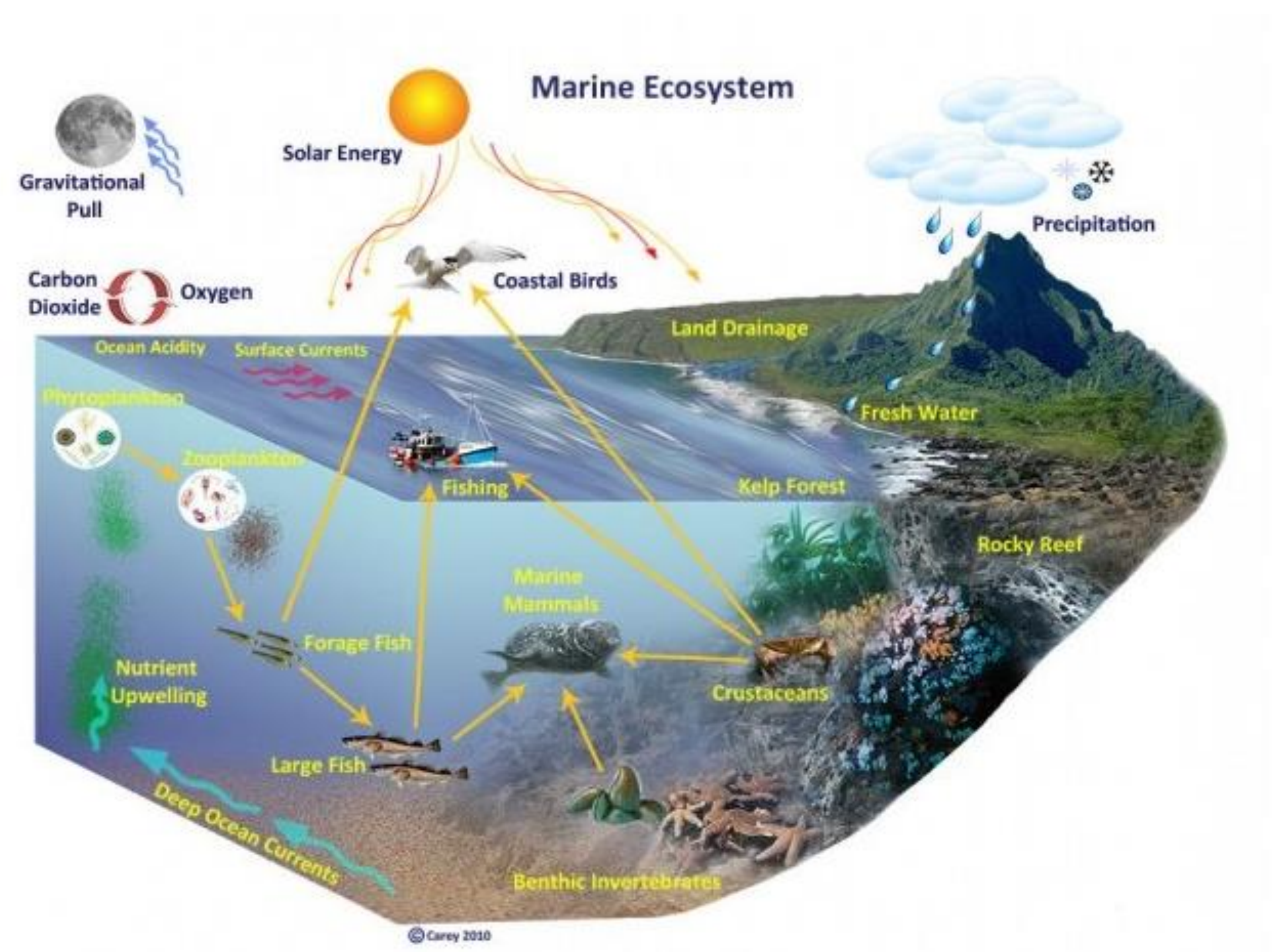
Merriam-Webster

to make a group of people disagree and fight with one another so that they will not join together against one. His military strategy is to *divide and conquer*.

Smart City Example



Ecosystems: People, Systems, and Things

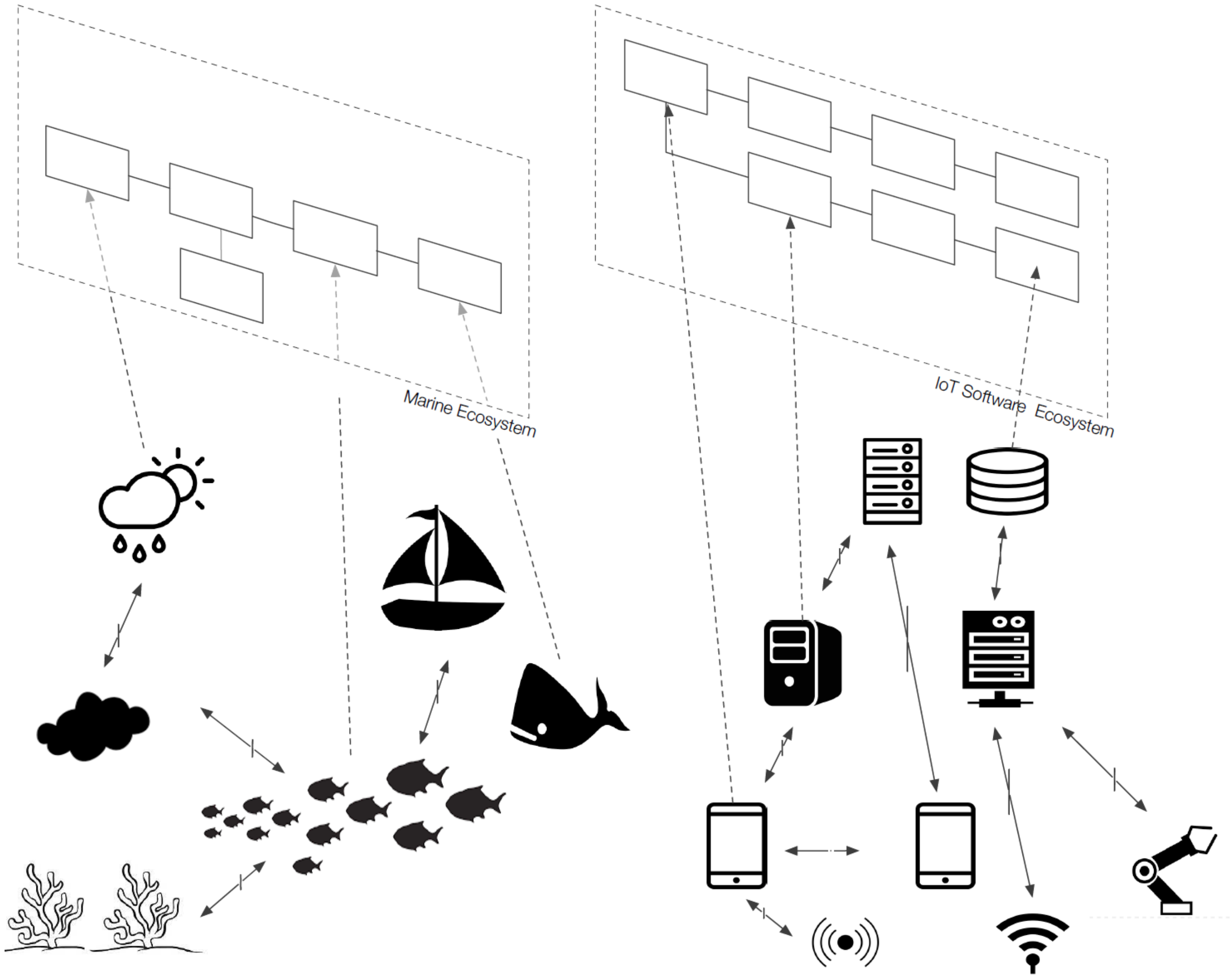


Marine Ecosystem: http://www.xbordercurrents.co.uk/wildlife/marine-ecosystem-2

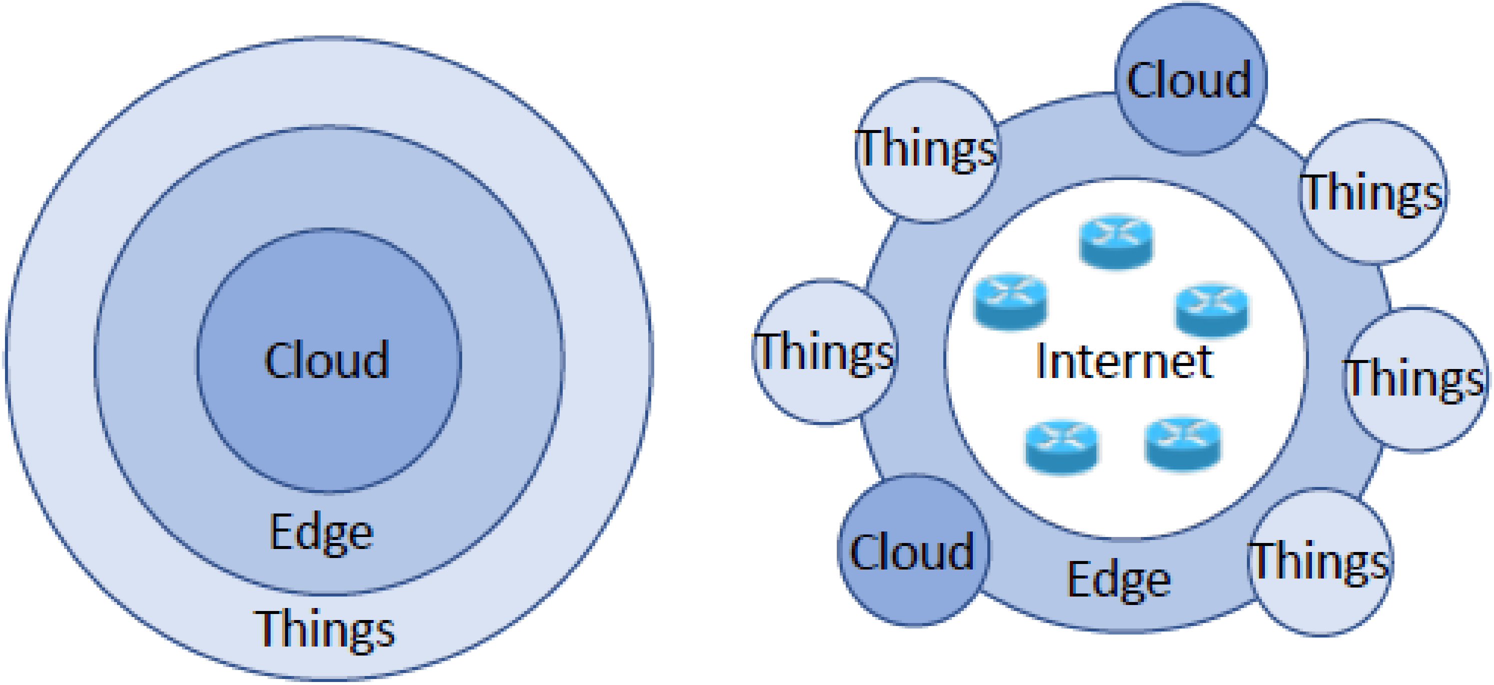
Complex system with networked dependencies and intrinsic adaptive behavior – has:

1. **Robustness & Resilience mechanisms**: achieving stability in the presence of disruption
2. **Measures of health**: diversity, population trends, other key indicators
3. **Built-in coherence**
4. **Entropy-resistence**

Ecosystems for IoT Systems



Perspectives on the IoT: Edge, Cloud, Internet



(a) A cloud-centric perspective: Edge as “edge of the cloud”

(b) An Internet-centric perspective: Edge as “edge of the Internet”

Cloud-centric perspective

Assumptions

- Cloud provides core services; Edge provides local proxies for the Cloud (offloading parts of the cloud’s workload)

Edge Computers

- play supportive role for the IoT services and applications
- Cloud computing-based IoT solutions use cloud servers for various purposes including massive computation, data storage, communication between IoT systems, and security/privacy

Missing

- In the network architecture, the cloud is also located at the network edge, not surrounded by the edge
- Computers at the edge do not always have to depend on the cloud; they can operate autonomously and collaborate with one another directly without the help of the cloud

Internet-centric perspective

Assumptions

- Internet is center of IoT architecture; Edge devices are gateways to the Internet (not the Cloud)
- Each LAN can be organized around edge devices autonomously
- Local devices do not depend on Cloud

Therefore

- Things belong to partitioned subsystems and LANs rather than to a centralized system directly
- The Cloud is connected to the Internet via the edge of the network
- Remote IoT systems can be connected directly via the Internet. Communications does not have to go via the Cloud
- The Edge can connect things to the Internet and disconnect traffic outside the LAN to protect things -> IoT system must be able to act autonomously

Edge perspectives

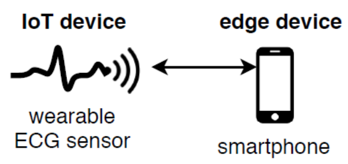


Fig. 1. Scenario with a wearable ECG sensor and a smartphone (1:1 mapping of an IoT to edge device).

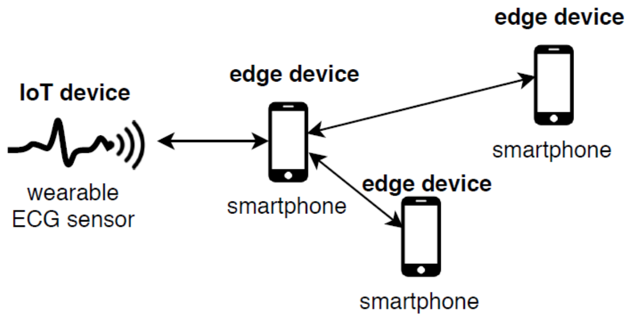


Fig. 2. A use case scenario of a wearable ECG sensor and a multiple number of smartphones (1:N mapping of an IoT to edge devices).

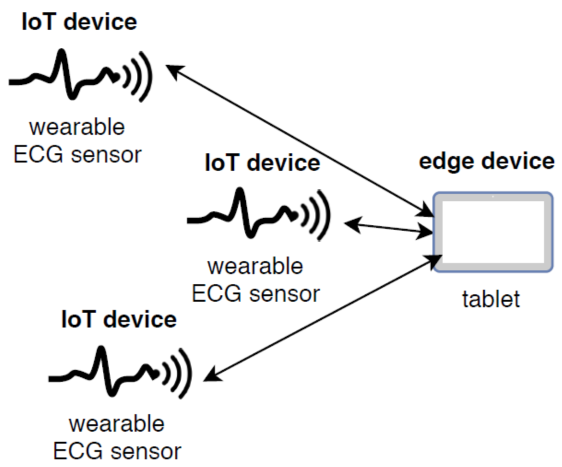


Fig. 3. Scenario with multiple wearable ECG sensors and a tablet (N:1 mapping of IoT devices to an edge device).

Cloud-IoT vs. Edge/Cloud hybrid IoT

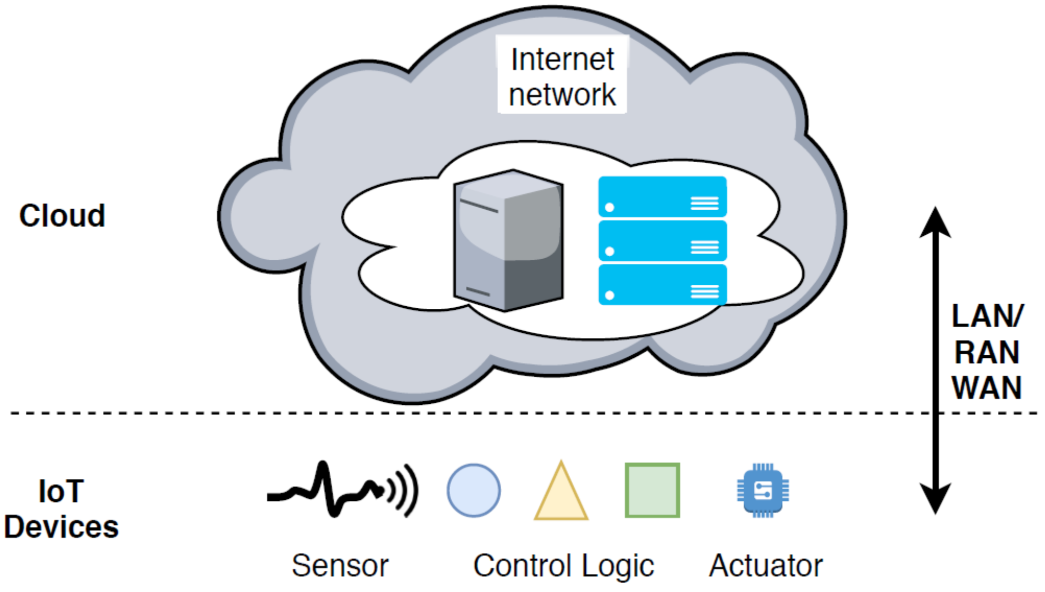


Fig. 4. Cloud-based IoT solution.

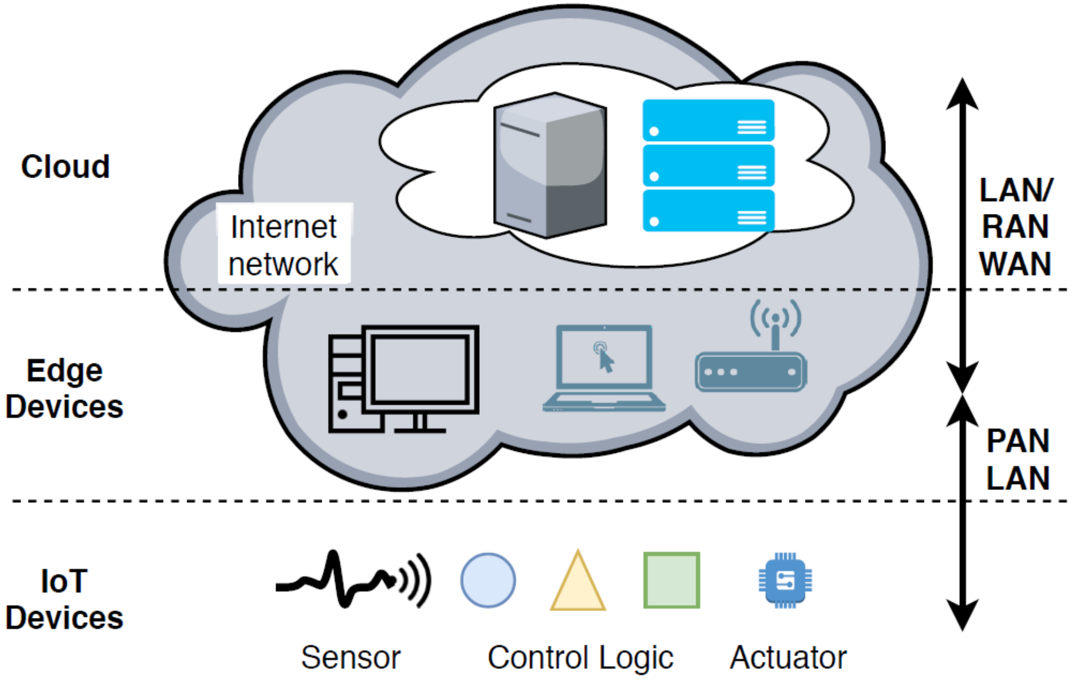


Fig. 5. Edge computing architecture for a streaming IoT solution.

Vertical vs. Horizontal Edge Architecture

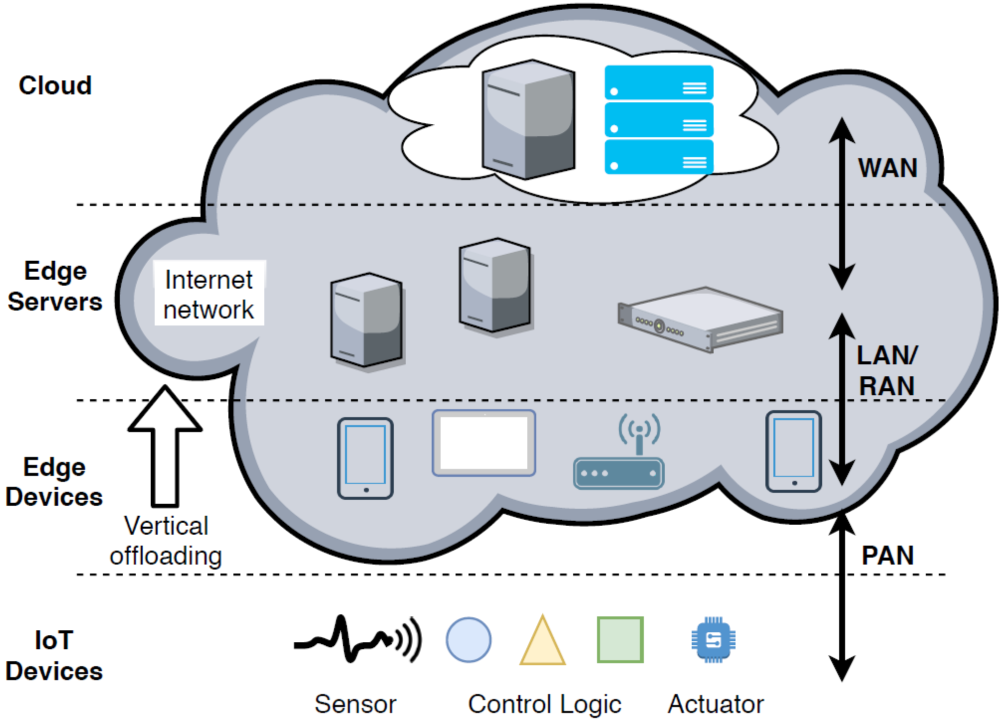


Fig. 6. Vertical edge computing architecture for a streaming IoT solution.

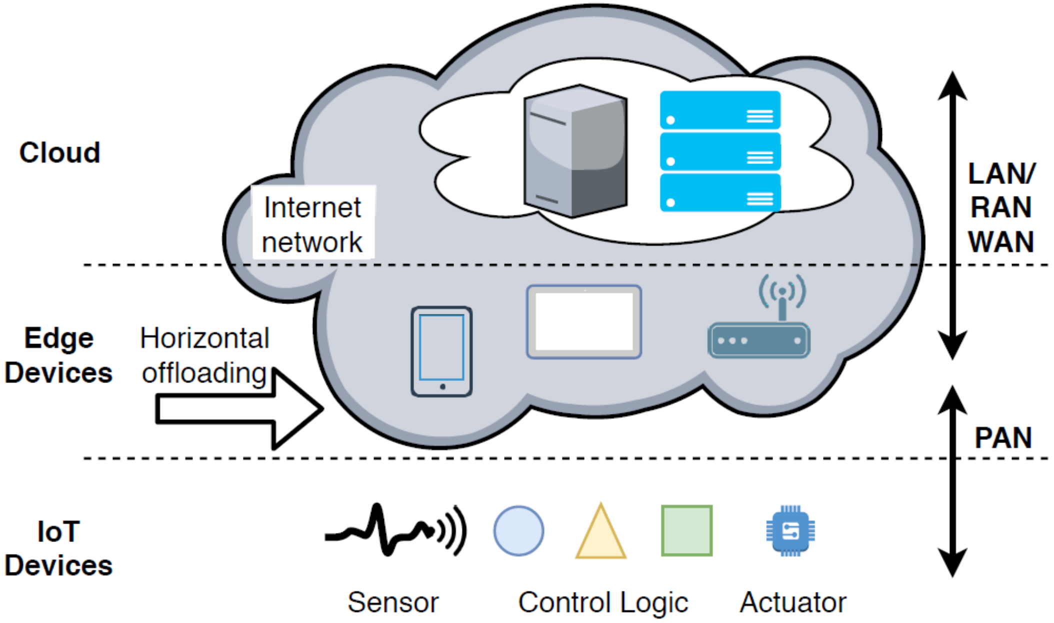
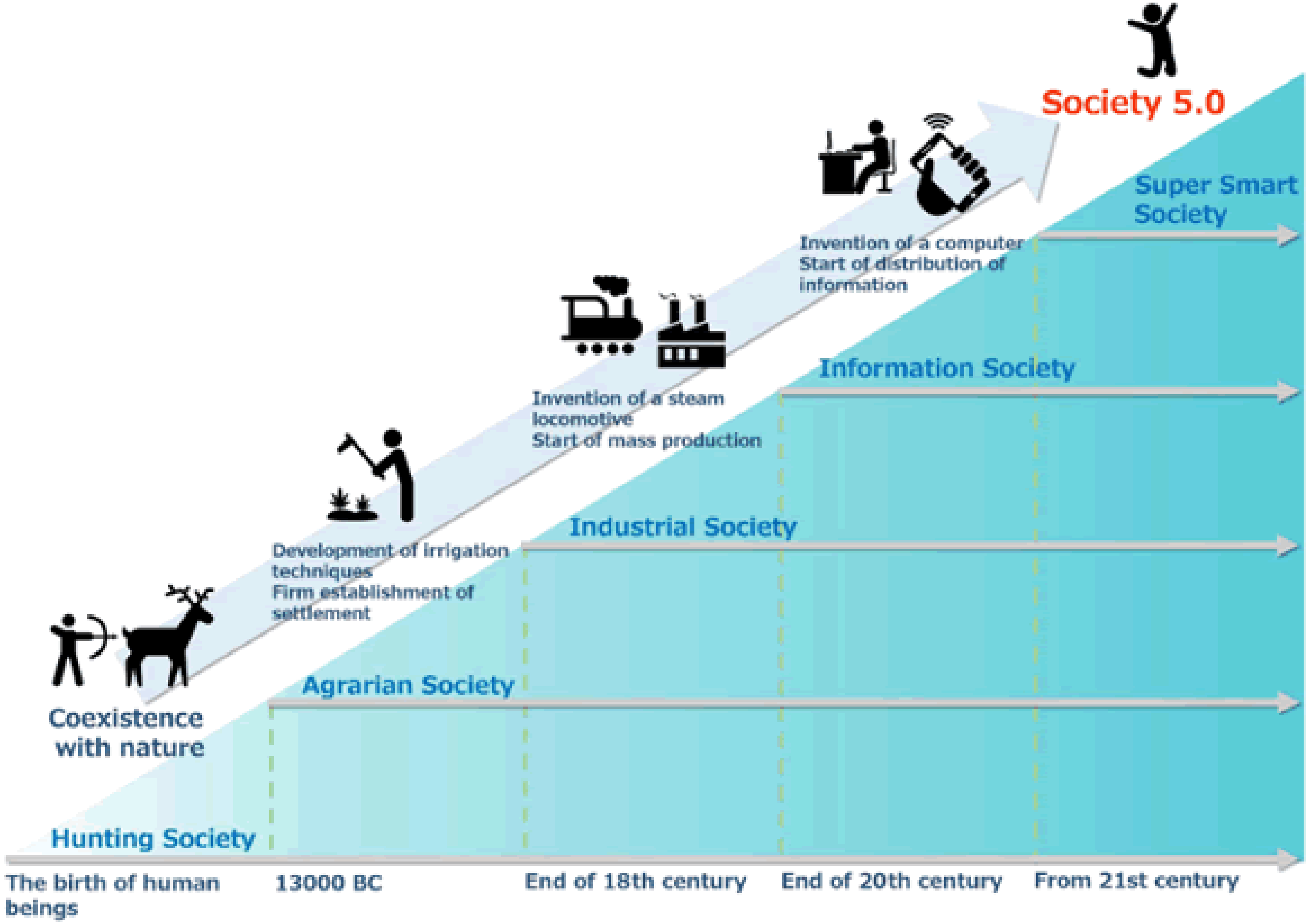


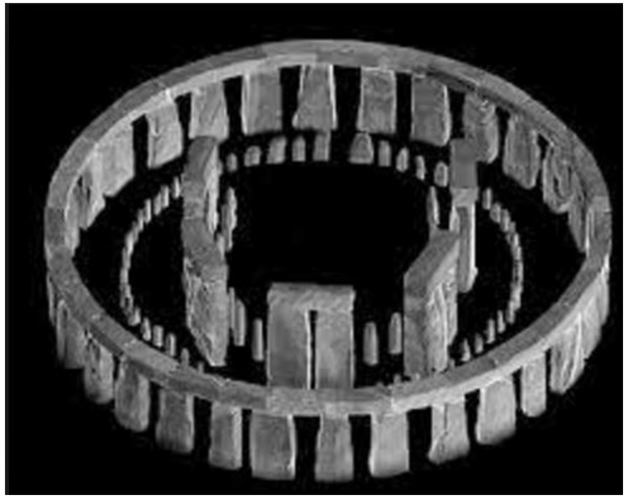
Fig. 7. Horizontal edge computing (serverless) architecture of a streaming IoT solution.

Ecosystem “Society 5.0“ (Japan)



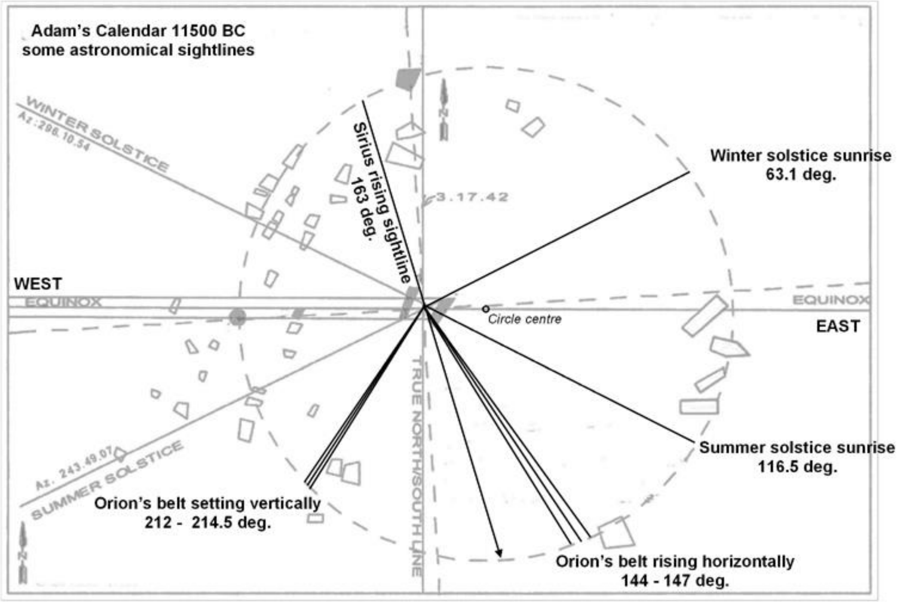
Toward realization of the new economy and society, Keidanren (Japan Business Federation), April 2016

Linear History? Ancient “Computers“

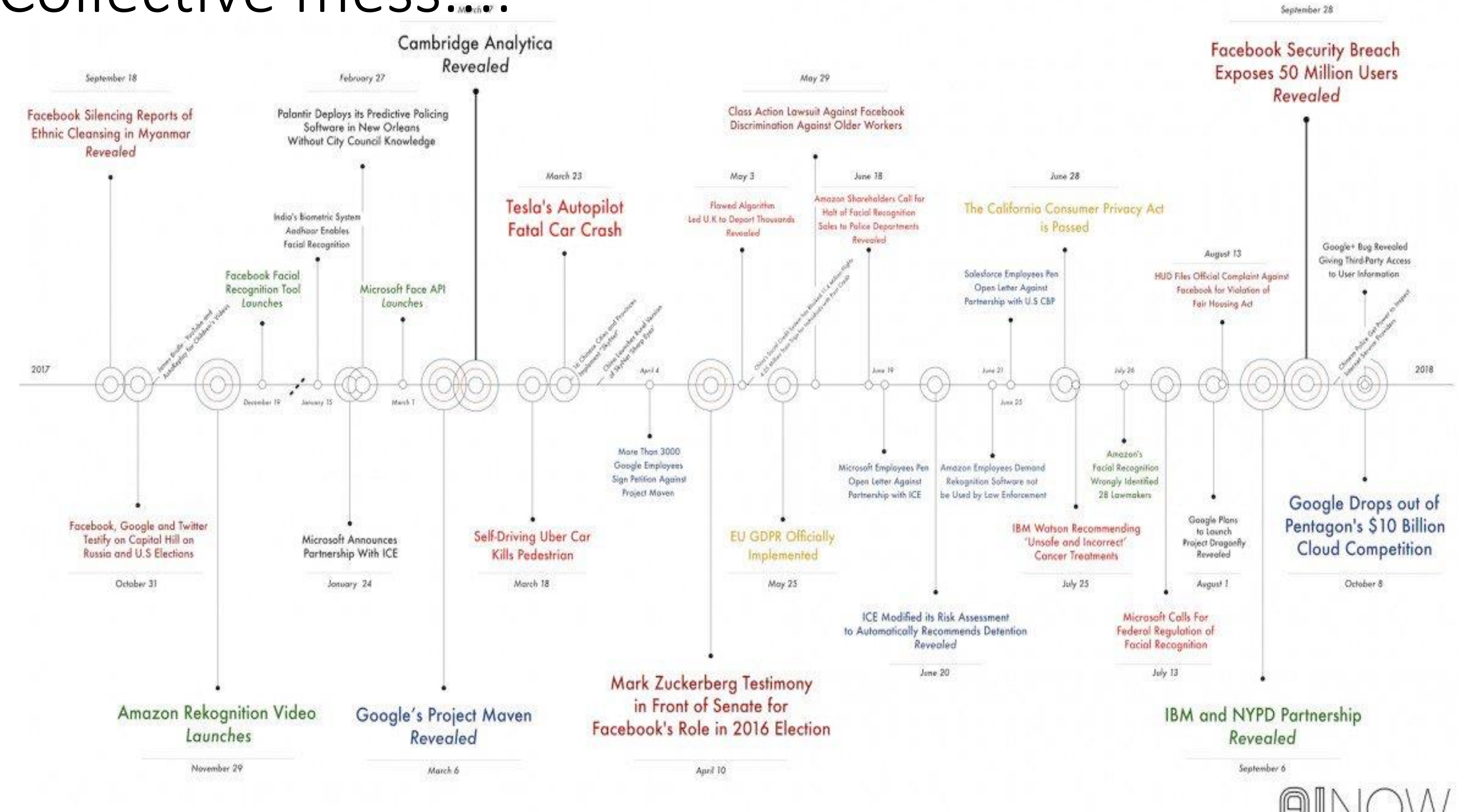


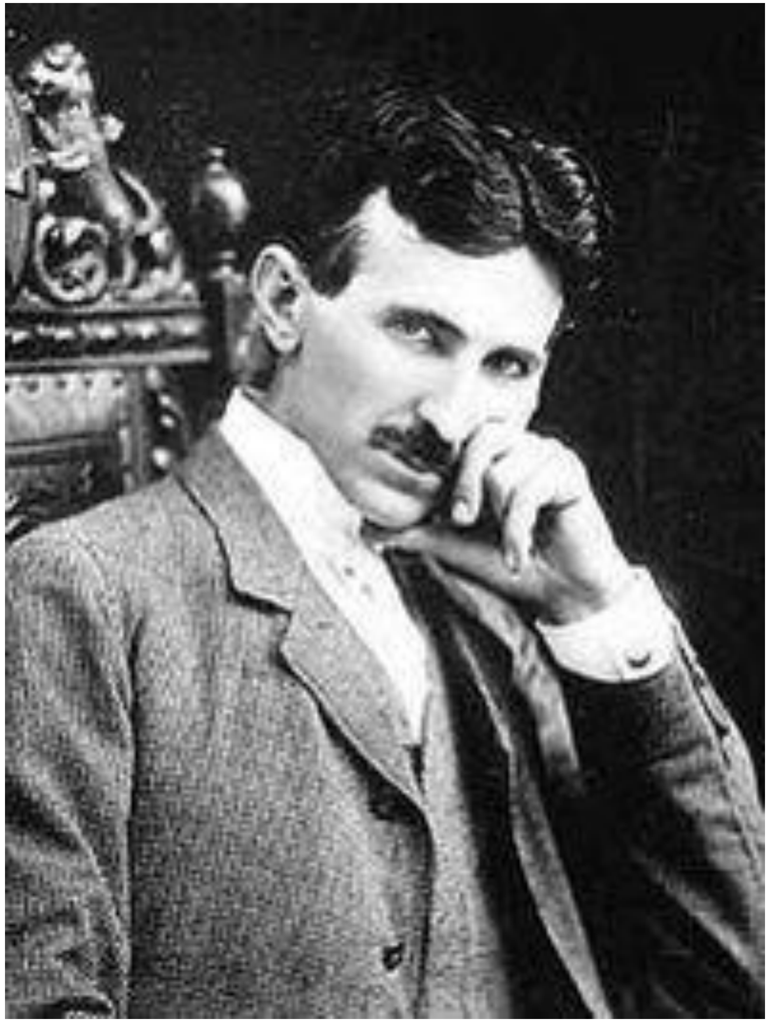
Stonehenge: A Neolithic Computer
Nature **202**, 1258 - 1261 (27 June 1964); doi:10.1038/2021258a0

Adam‘s Calender, Michael Tellinger



Collective mess…..





[Nikola Tesla](#)

"The scientists of today think *deeply* instead of *clearly*.

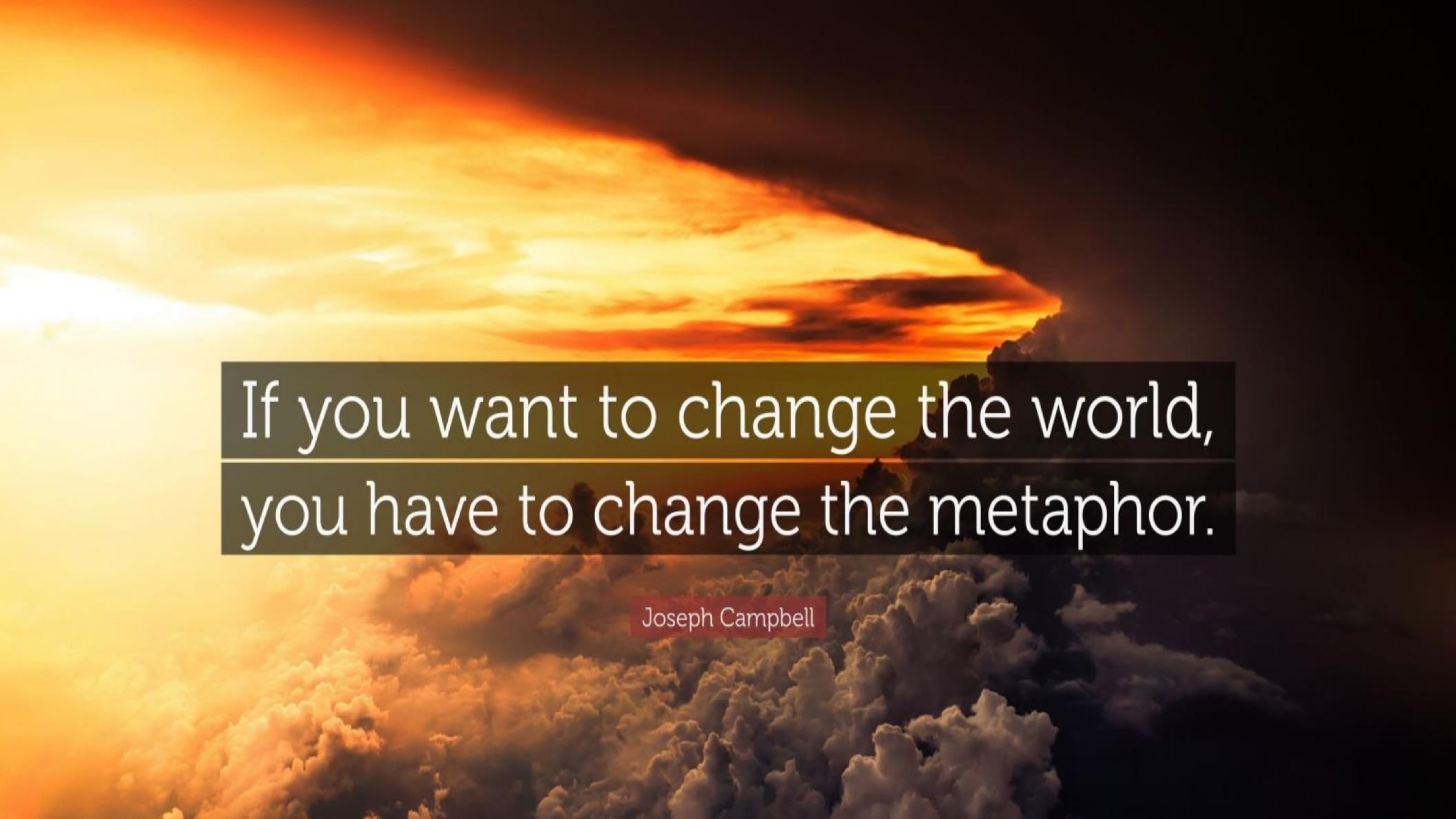
One must be sane to think clearly,
but one can think deeply and be quite insane."

"You cannot hope to build a better world without improving the individuals.

To that end each of us must work for his own improvement, and at the same time share a general responsibility for all humanity,

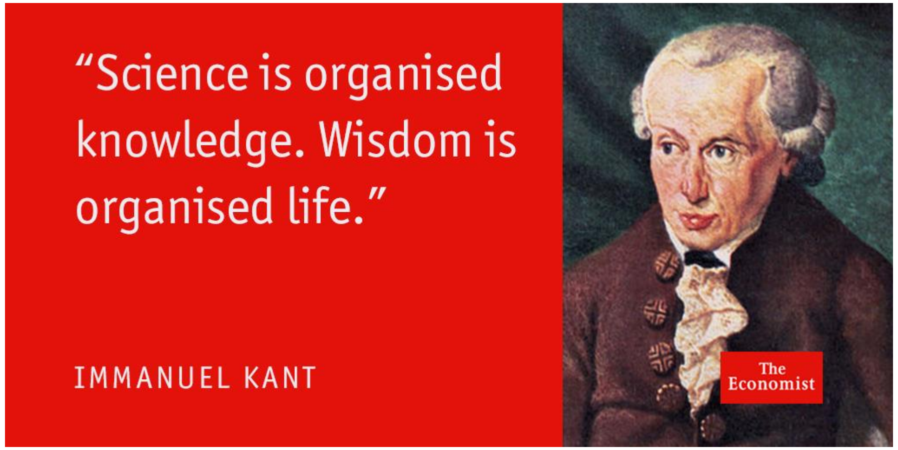
our particular duty being to aid those to whom we think we can be most useful."

Marie Curie


If you want to change the world,
you have to change the metaphor.
Joseph Campbell

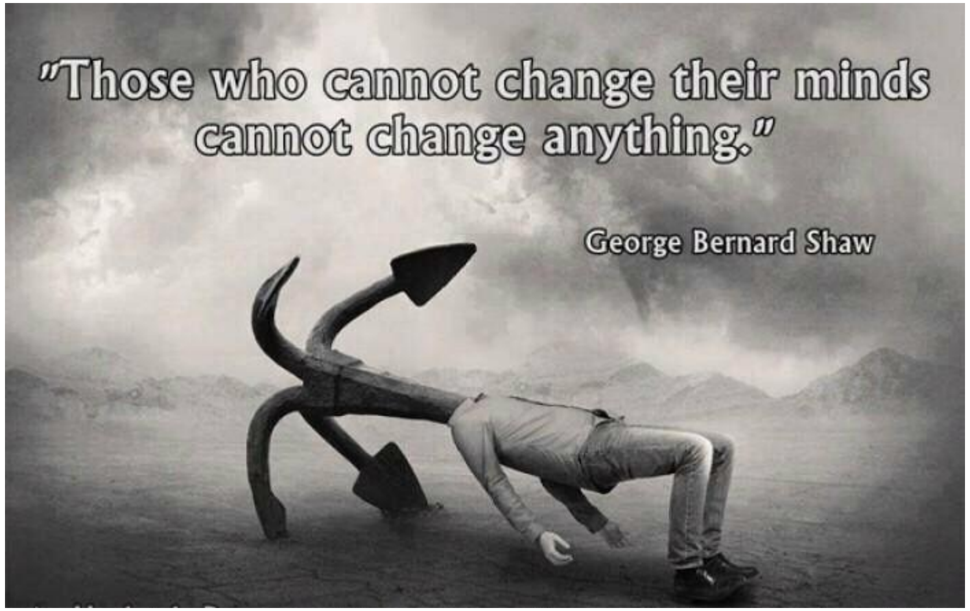
Assumptions, Models, and Abstractions

- **Co-evolution of Science & Technologies**
- Smart Cities as models of ecosystems: -> People, Things, and Systems
- Models as abstractions are useful (Platonic Forms)
- We lack a model for such an ecosystem
- From automation to creativity support
- Consciousness and creativity support -> lead to new (meta) models and understanding of technologies and science -> Architecture of Values



Layers of Paradigms

- Not reductionist
- We have to **create the abstractions and models** we want based on our understanding of human and societal needs
- **Ecosystems = Architecture, Structure + Dynamics**
- **New Paradigms**: (1) Elastic Computing, (2) Social Compute Units, (3) Osmotic Computing
- Emergent properties on higher levels with own properties



Paradigm 1: Elasticity (Resilience)

(Physics) The property of returning to an initial form or state following deformation

stretch when a force stresses them

e.g., ***acquire*** *new resources,* ***reduce*** *quality*

shrink when the stress is removed

e.g., ***release*** *resources,* ***increase*** *quality*

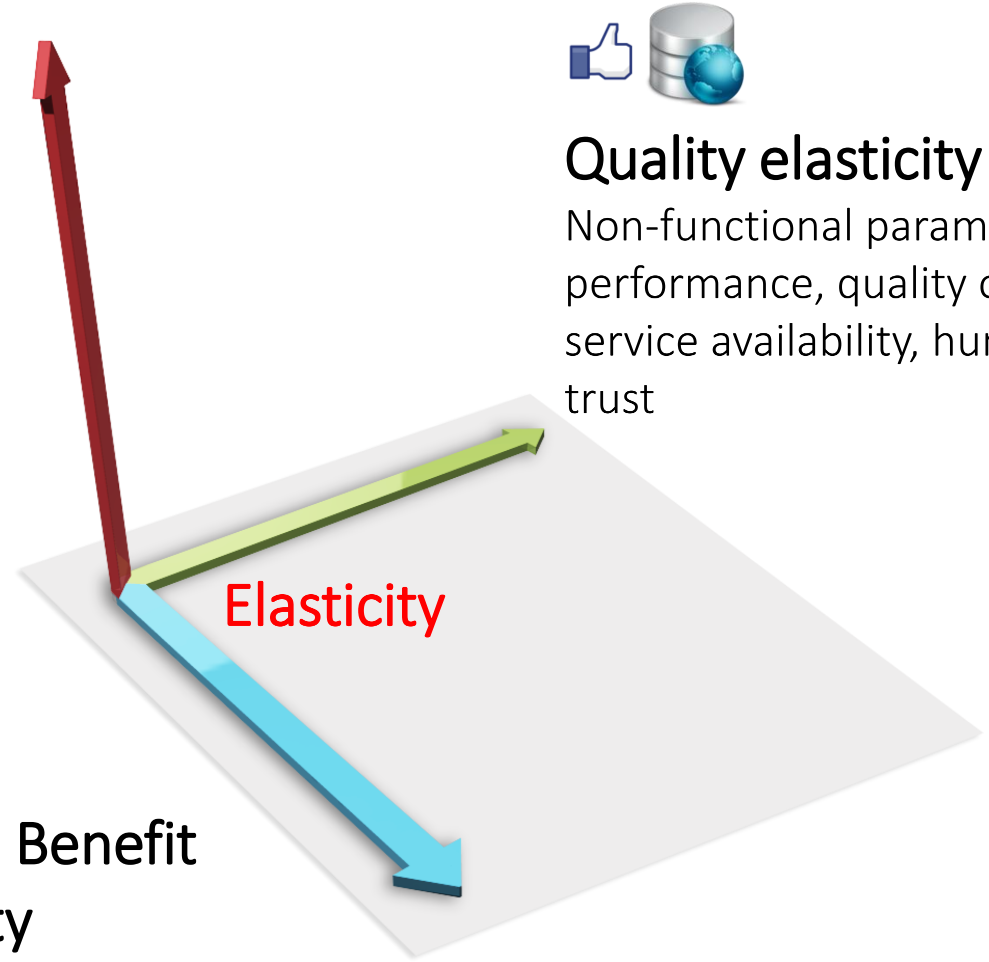
Elastic Computing > Scalability

Resource elasticity

Software / human-based computing elements, multiple clouds

Quality elasticity

Non-functional parameters e.g., performance, quality of data, service availability, human trust



Costs & Benefit elasticity

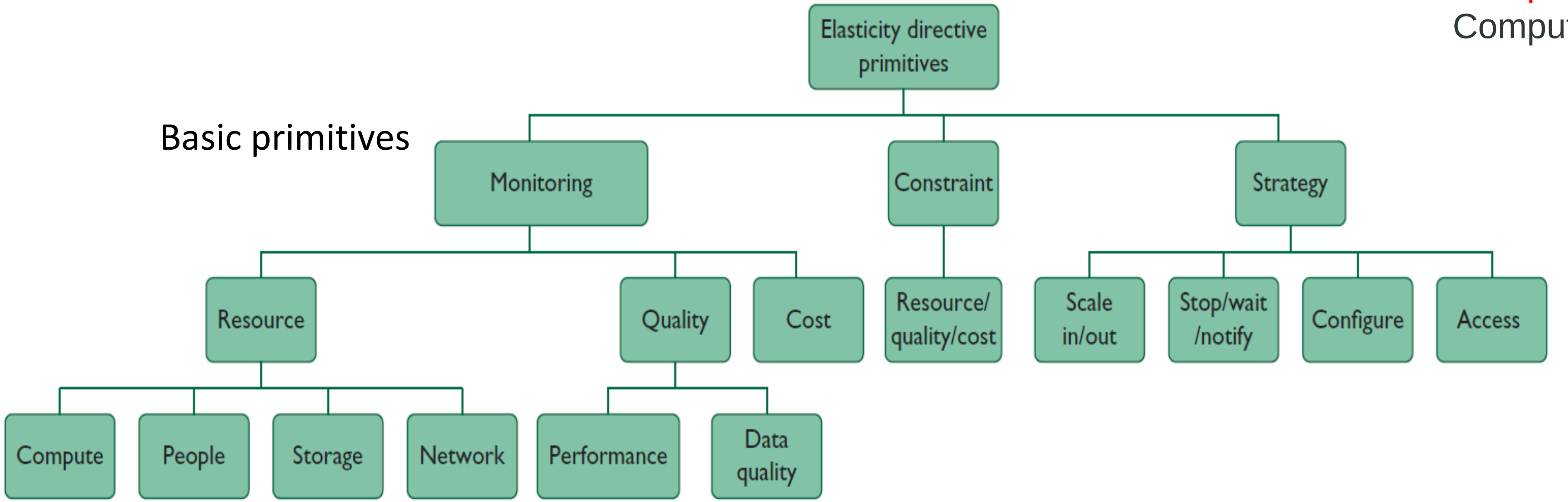
rewards, incentives

Dustdar S., Guo Y., Satzger B., Truong H. (2012) Principles of Elastic Processes, IEEE Internet Computing, Volume: 16, Issue: 6, Nov.-Dec. 2012

Specifying and controling elasticity

Dustdar, S. et al.: Programming Directives for Elastic Computing. IEEE Internet Computing 16(6): 72-77 (2012)

Basic primitives



SYBL (Simple Yet Beautiful Language) for specifying elasticity requirements

SYBL-supported requirement levels

- Cloud Service Level
- Service Topology Level
- Service Unit Level
- Relationship Level
- Programming/Code Level

Current SYBL implementation

in Java using Java annotations

@SYBLAnnotation(monitoring=„“,constraints=„“,strategies=„“)

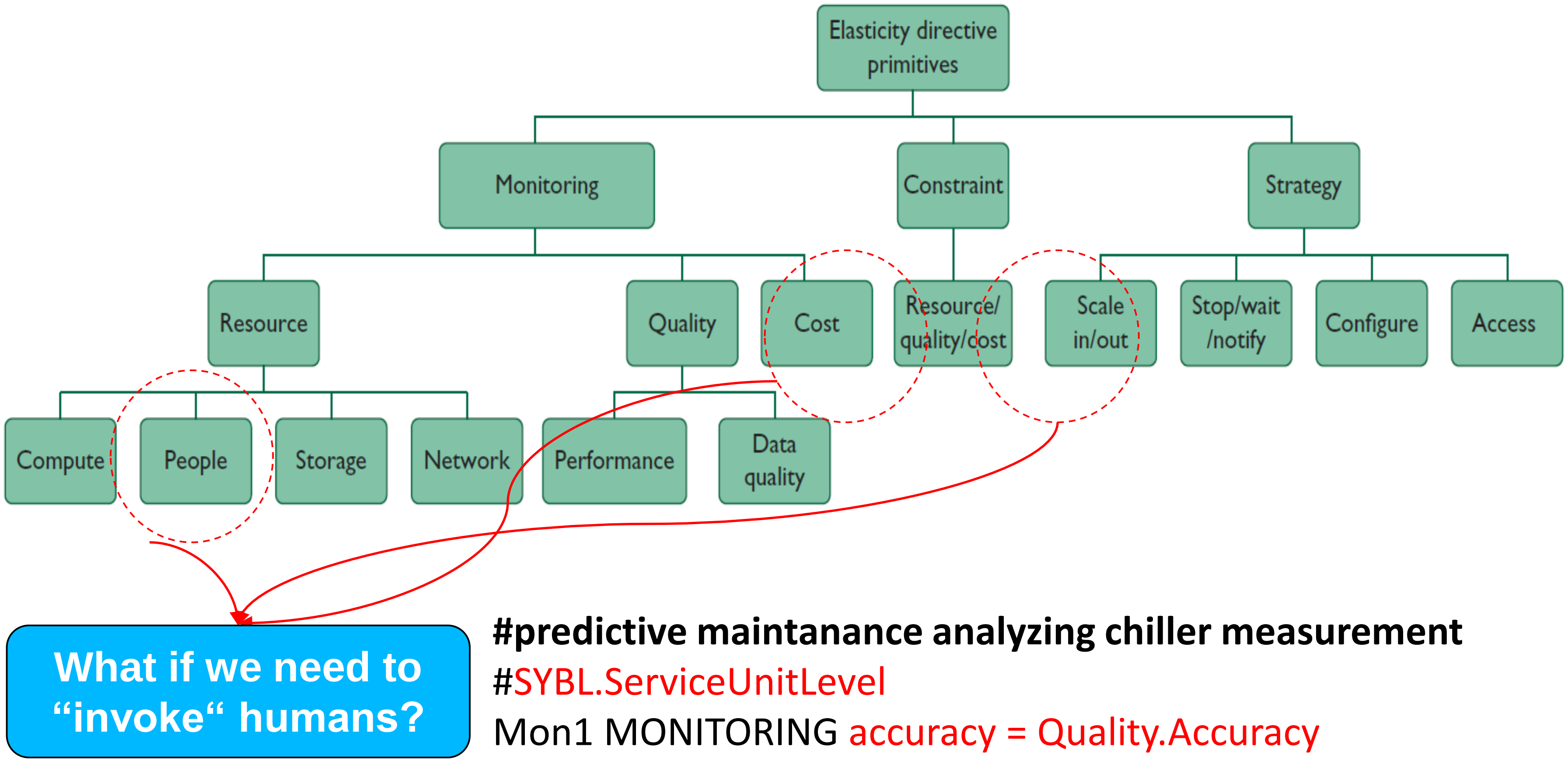
in XML

<ProgrammingDirective><Constraints><Constraint name=c1>...</Constraint></Constraints>...</ProgrammingDirective>

as TOSCA Policies

<tosca:ServiceTemplate name="PilotCloudService"> <tosca:Policy name="St1" policyType="SYBLStrategy"> St1:STRATEGY minimize(Cost) WHEN high(overallQuality) </tosca:Policy>...

Specifying and controling elasticity of human-based services



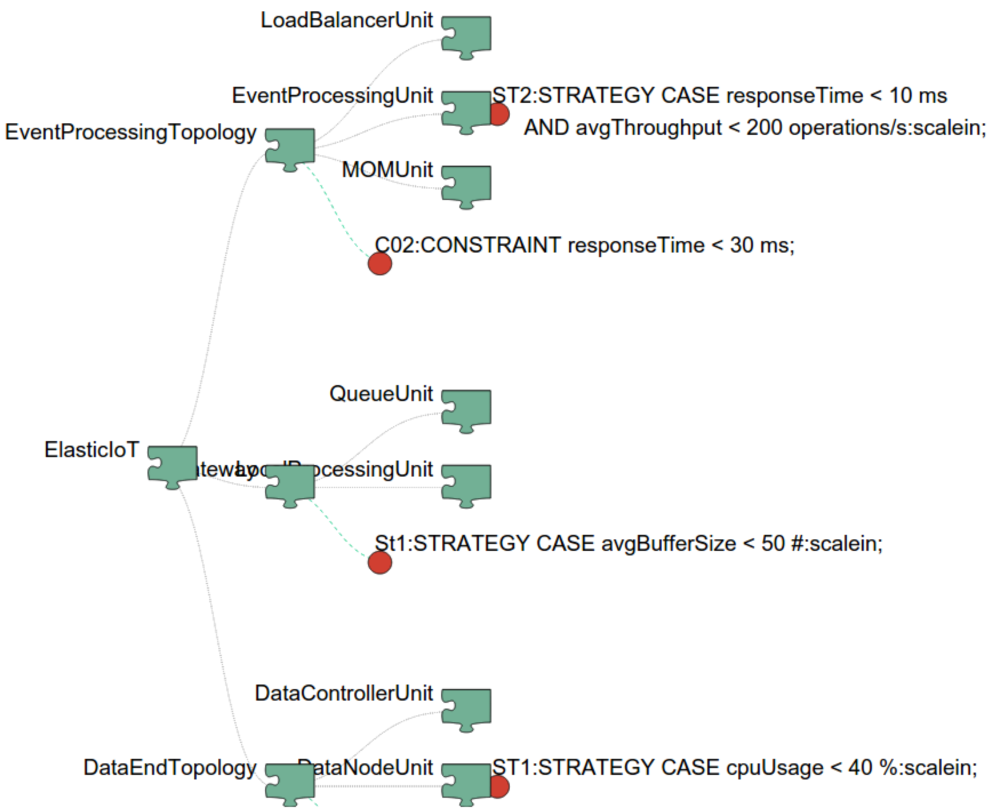
#predictive maintanance analyzing chiller measurement
#SYBL.ServiceUnitLevel
Mon1 MONITORING accuracy = Quality.Accuracy
Cons1 CONSTRAINT accuracy < 0.7
Str1 STRATEGY CASE Violated(Cons1):
Notify(Incident.DEFAULT, ServiceUnitType.HBS)

High level elasticity control

#SYBL.CloudServiceLevel
Cons1: CONSTRAINT responseTime < 5 ms
Cons2: CONSTRAINT responseTime < 10 ms
WHEN nbOfUsers > 10000
Str1: STRATEGY CASE fulfilled(Cons1) OR fulfilled(Cons2): minimize(cost)

#SYBL.ServiceUnitLevel
Str2: STRATEGY CASE ioCost < 3 Euro : maximize(dataFreshness)

#SYBL.CodeRegionLevel
Cons4: CONSTRAINT dataAccuracy>90% AND cost<4 Euro



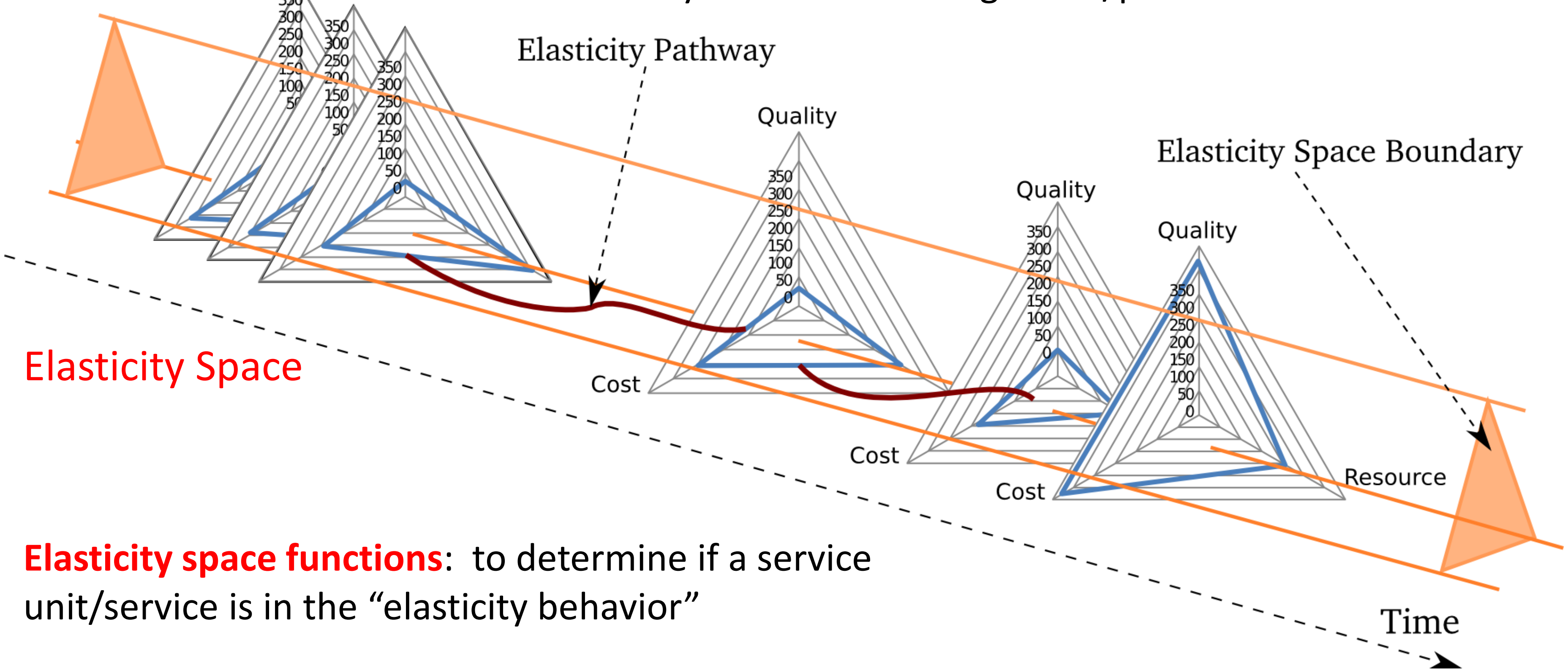
Georgiana Copil, Daniel Moldovan, Hong-Linh Truong, Schahram Dustdar, **"SYBL: an Extensible Language for Controlling Elasticity in Cloud Applications",** 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), May 14-16, 2013, Delft, Netherlands

Copil G., Moldovan D., Truong H.-L., Dustdar S. (2016). **rSYBL: a Framework for Specifying and Controlling Cloud Services Elasticity**. *ACM Transactions on Internet Technology*

Elasticity Model for Cloud Services

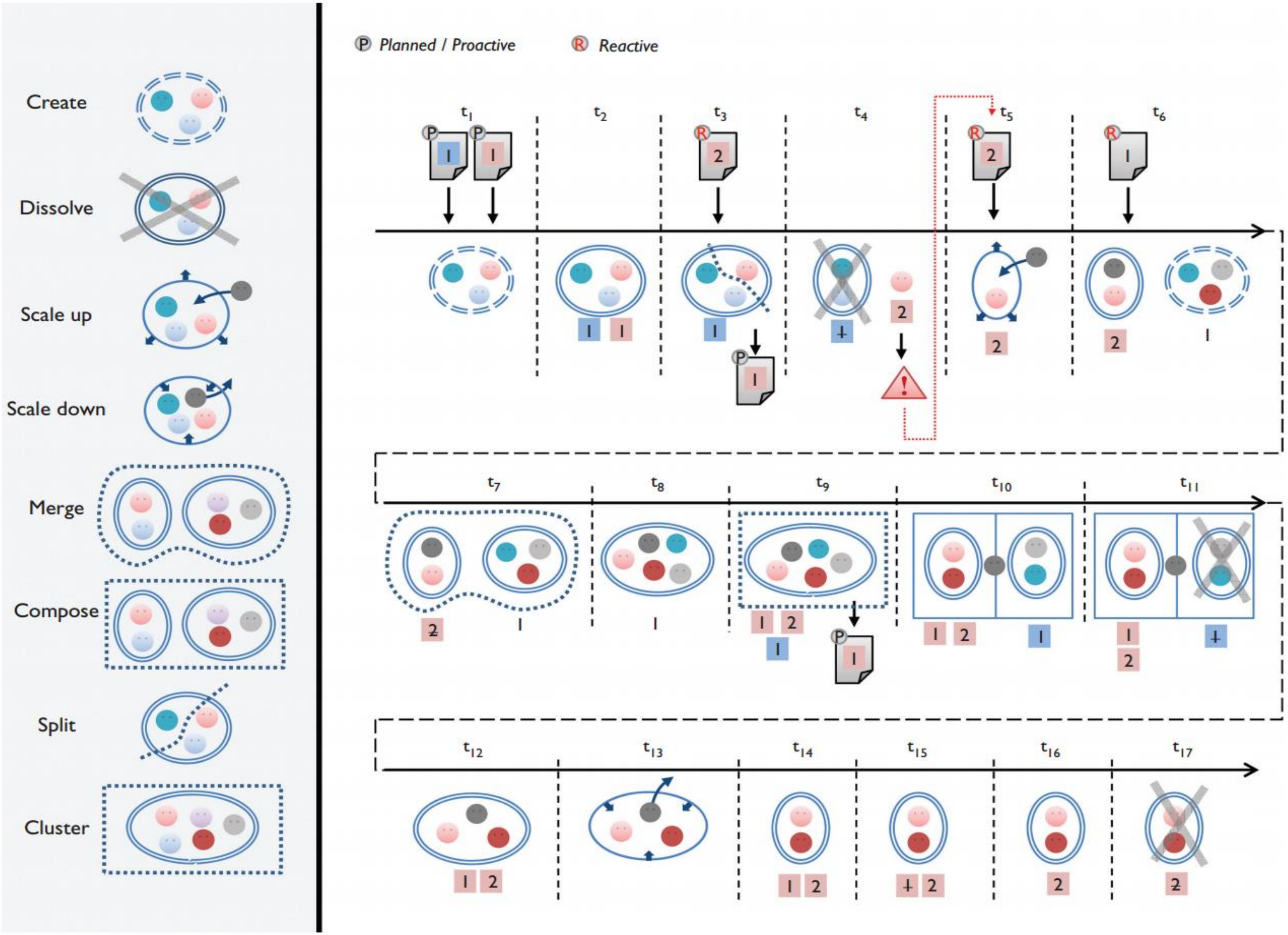
Moldovan D., G. Copil,Truong H.-L., Dustdar S. (2013). **MELA: Monitoring and Analyzing Elasticity of Cloud Service. CloudCom 2013**

Elasticity Pathway functions: to characterize the elasticity behavior from a general/particular view



Elasticity space functions: to determine if a service unit/service is in the "elasticity behavior"

Paradigm 2: Social Compute Units (SCUs)



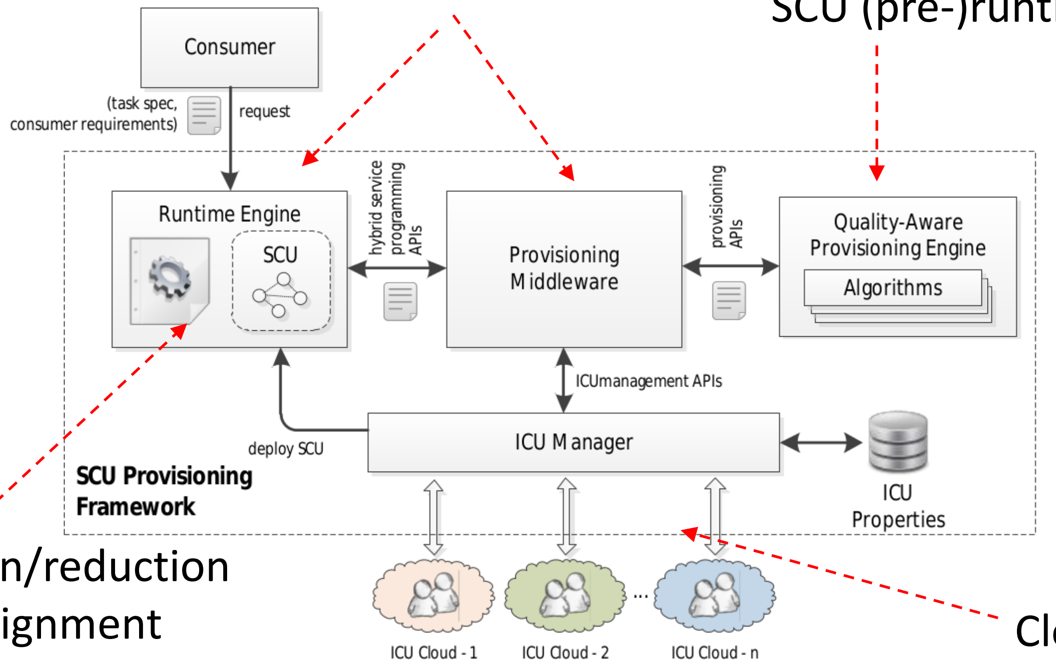
Dustdar S., Bhattacharya K. (2011). The Social Compute Unit, *IEEE Internet Computing,* Volume 15, Issue 3; pp. 64 - 69.

Fernández P., Truong H.-L., Dustdar S., Ruiz-Cortés A. (2015). Programming Elasticity and Commitment in Dynamic Processes. IEEE Internet Computing, Volume 19, Number 2, pp. 68 - 74

Elastic SCU provisioning (Paradigms 1 and 2 together)

Elastic profile

SCU (pre-)runtime/static formation



Algorithms

- Ant Colony Optimization variants
- FCFS
- Greedy

SCU extension/reduction

- Task reassignment based on trust, cost, availability

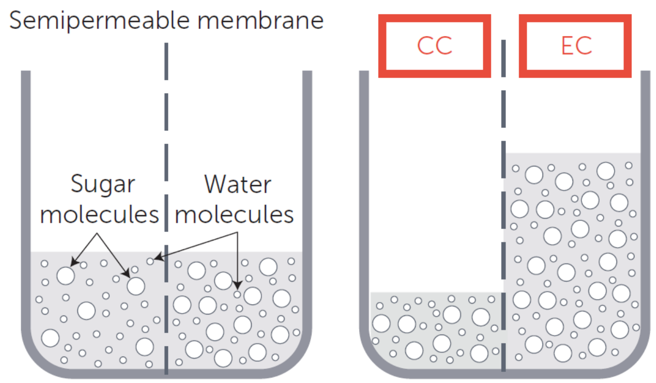
Cloud APIs

Mirela Riveni, Hong-Linh Truong, and Schahram Dustdar, **On the Elasticity of Social Compute Units, CAISE 2014**

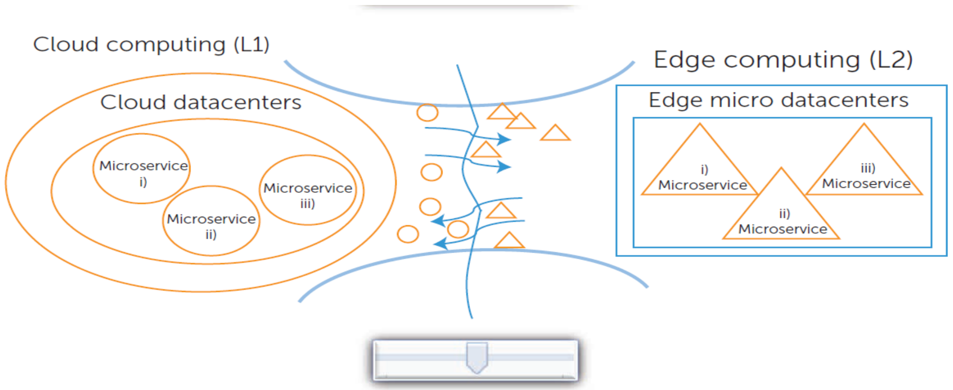
Muhammad Z.C. Candra, Hong-Linh Truong, and Schahram Dustdar, **Provisioning Quality-aware Social Compute Units in the Cloud, ICSOC 2013.**

Paradigm 3: Osmotic Computing

- In chemistry, “osmosis” represents the seamless diffusion of molecules from a higher to a lower concentration solution.



- Dynamic management of (micro)services across cloud and edge datacenters
  - deployment, networking, and security, …
  - providing reliable IoT support with specified levels of QoS.



Villari M., Fazio M., Dustdar S., Rana O., Ranjan R. (2016). Osmotic Computing: A New Paradigm for Edge/Cloud Integration. *IEEE Cloud Computing*, Volume 3, Issue 6, pp. 76-83

IoT & Data Science – Research Challenges

Ranjan R., Rana O., Nepal S., Yousif M., James P., Wen Z., Barr S., Watson P., Jayaraman P. P., Georgakopoulos D., Villari M., Fazio M., Garg S., Buyya R., Wang L., Zomaya A. Y., Dustdar S. (2018). [The Next Grand Challenges: Integrating the Internet of Things and Data Science](), *IEEE Cloud Computing*, Volume 5, Issue 3, pp. 12-26

DEPARTMENT: Blue Skies

The Next Grand Challenges

Integrating the Internet of Things and Data Science

Rajiv Ranjan
Newcastle University

Omer Rana
Cardiff University

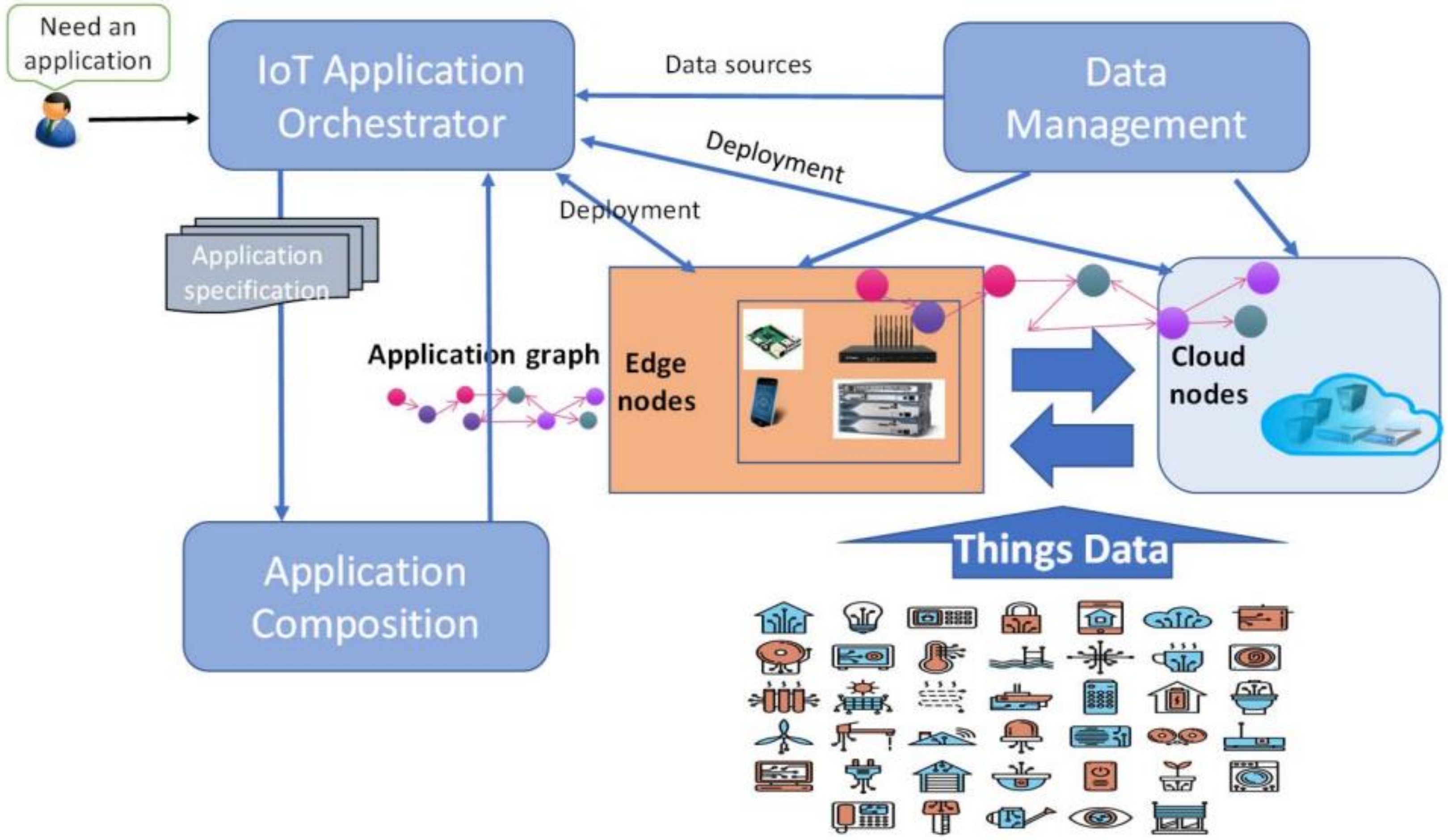
Surya Nepal
Data61, CSIRO

Mazin Yousif
T-Systems, International

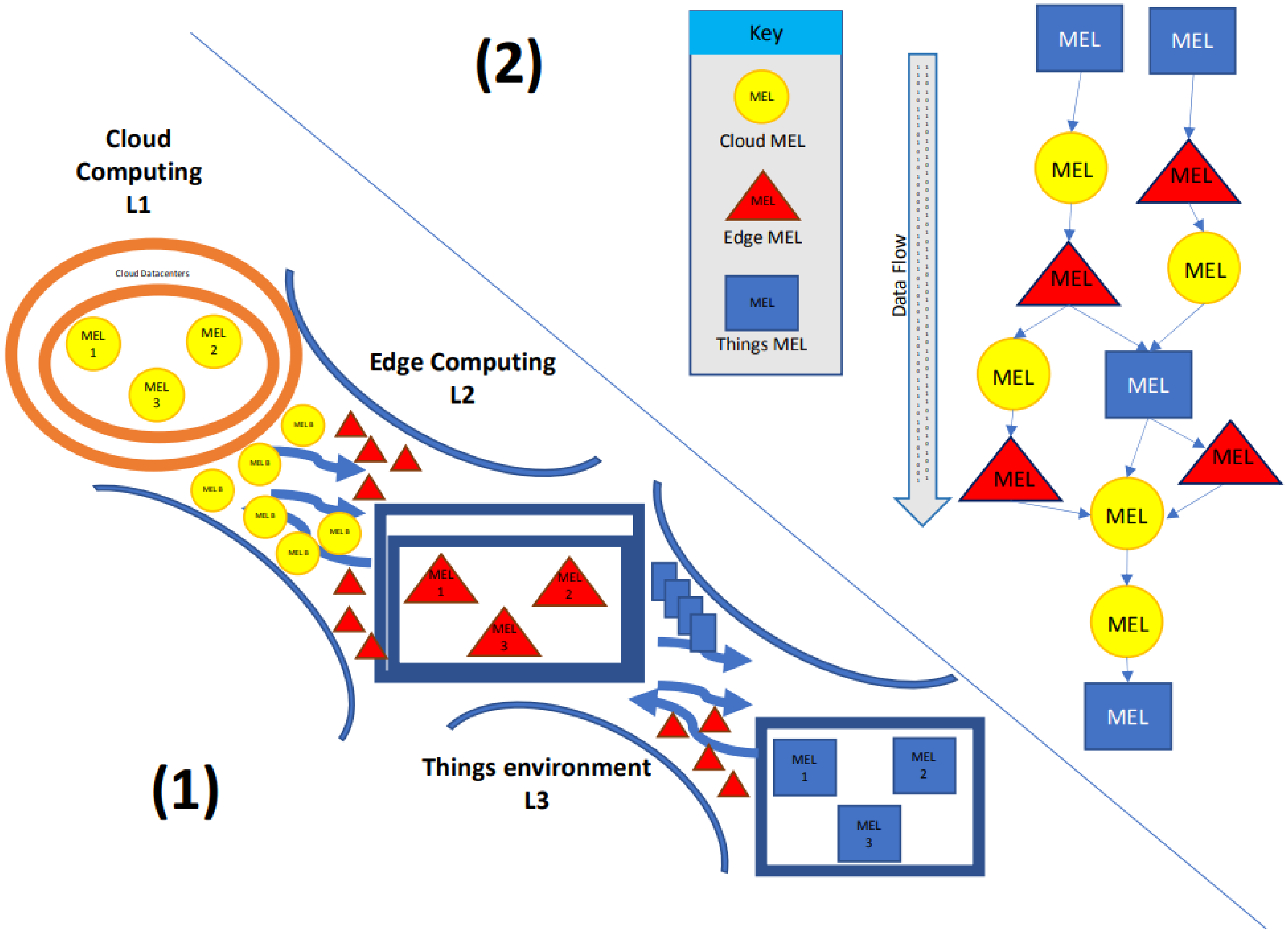
This article discusses research challenges related to devising a new IoT programming paradigm for orchestrating IoT applications' composition and data processing across heterogeneous computing infrastructure (Cloud, Edge, and Things).

IoT-driven ecosystems

IoT/Data/Application Orchestration



Osmotic movement of MELs in Clouds, Edge, Things



Legend:
MEL...Micro Element

IoT Data Sources

1. **Representation**: Structure and represent the data to facilitate multiple modalities, exploiting the complementarity and redundancy of different data sources.

2. **Translation**: Interpret data from one modality to another, i.e., provide a translator that allows the modalities to interact with each other for enabling data exchange.

3. **Alignment**: Identify the relation among modalities. This requires identifying links between different types of data.

4. **Fusion**: Fuse information from different modalities (e.g., to predict).

5. **Co-learning**: Transfer knowledge among modalities. This explores the field of how the knowledge of a modality can help or enhance a computational model trained on a different modality.

IoT Mircoelements (MELs)

1. **MicroServices** (MS), which implement specific functionalities and can be deployed and migrated across different virtualized and/or containerized infrastructures (e.g., Docker) available across Cloud, Edge, and Things layers

2. **MicroData** (MD), encodes the contextual information about (a) the sensors, actuators, edge devices, and cloud resources it needs to collect data from or send data to, (b) the specific type of data (e.g., temperature, vibration, pollution, pH, humidity) it needs to process, and (c) other data manipulation operations such as where to store data, where to forward data, and where to store results

3. **MicroComputing** (MC), executing specific types of computational tasks (machine learning, aggregation, statistical analysis, error checking, and format translation) based on a mix of historic and real-time MD data in heterogeneous formats. These MCs could be realized using a variety of data storage and analytics programming models (SQL, NoSQL, stream processing, batch processing, etc.)

4. **MicroActuator** (MA), implementing programming interfaces (e.g., for sending commands) with actuator devices for changing or controlling object states in the IoT environment

IoT Programming Patterns needed

1. **Decomposing IoT data analysis activities into fine-grained activities** (e.g., statistics, clustering, classification, anomaly detection, accumulation, filtering), each of which may impose different planning and run-time orchestration requirements;

2. Identifying and integrating **real-time data from IoT devices and historical IoT data** distributed across Cloud and Edge resources;

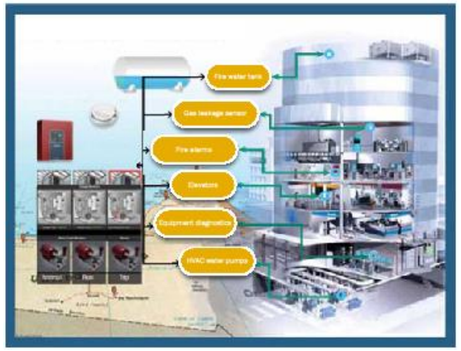
3. Identifying **data and control flow dependencies** between data analysis activities focusing on coordination and data flow variables, as well as the handling of dynamic system updates and re-configuration;

4. Defining and tagging each **data analysis activity with runtime deployment constraints** (QoS, security and privacy).

Motivating Case Studies

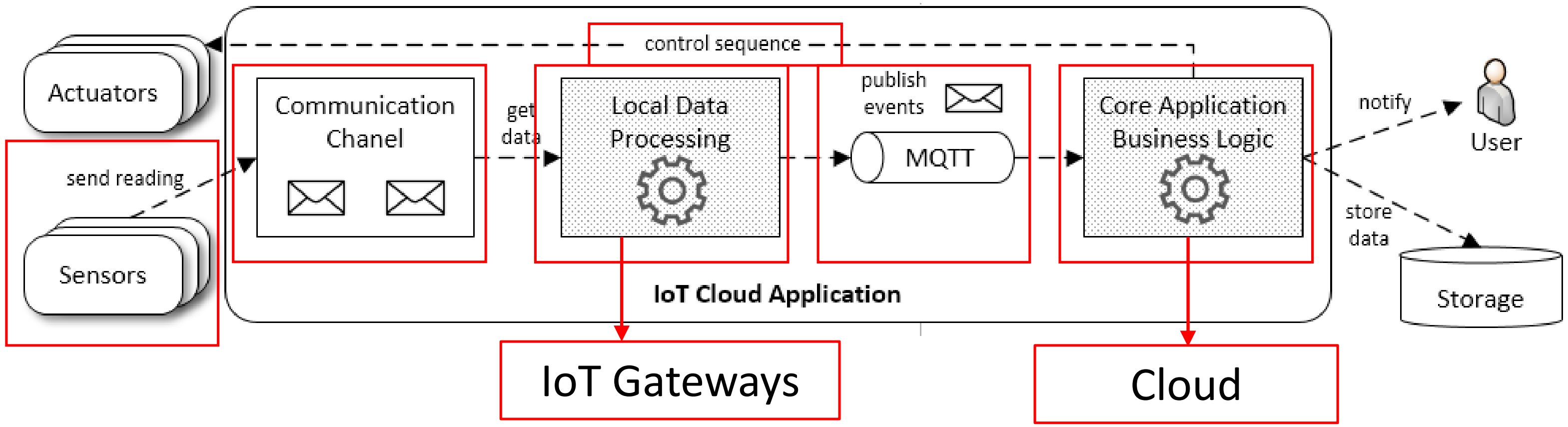
Building Management System

- Manages building facilities, e.g., HVAC systems, elevators and emergency alarms



Fleet Management System

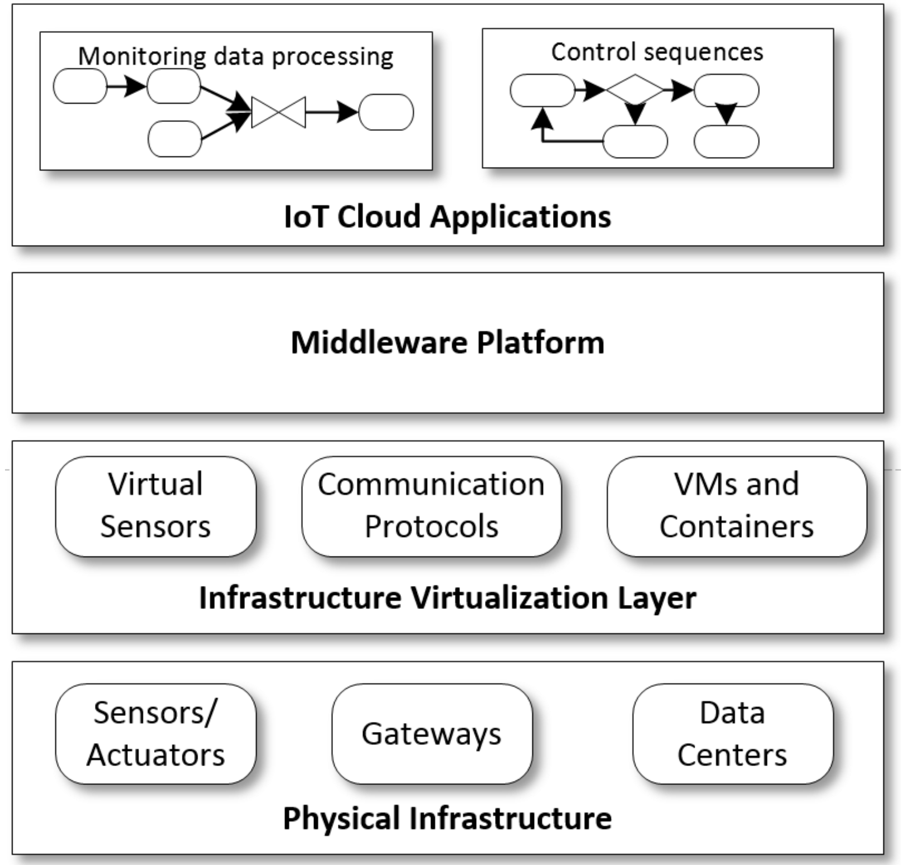
- Manages fleets of electric vehicles world-wide (e.g., on golf courses)



Motivation

- Lack of systematic support and tools for **developing**, **deploying**, and **operating** IoT systems (Cloud, Fog, and Edge)

- Today IoT systems are vertically closed and tightly coupled
  - Hard to develop and maintain applications
  - Difficult to operate and reuse existing infrastructure



Programming Model for IoT Systems

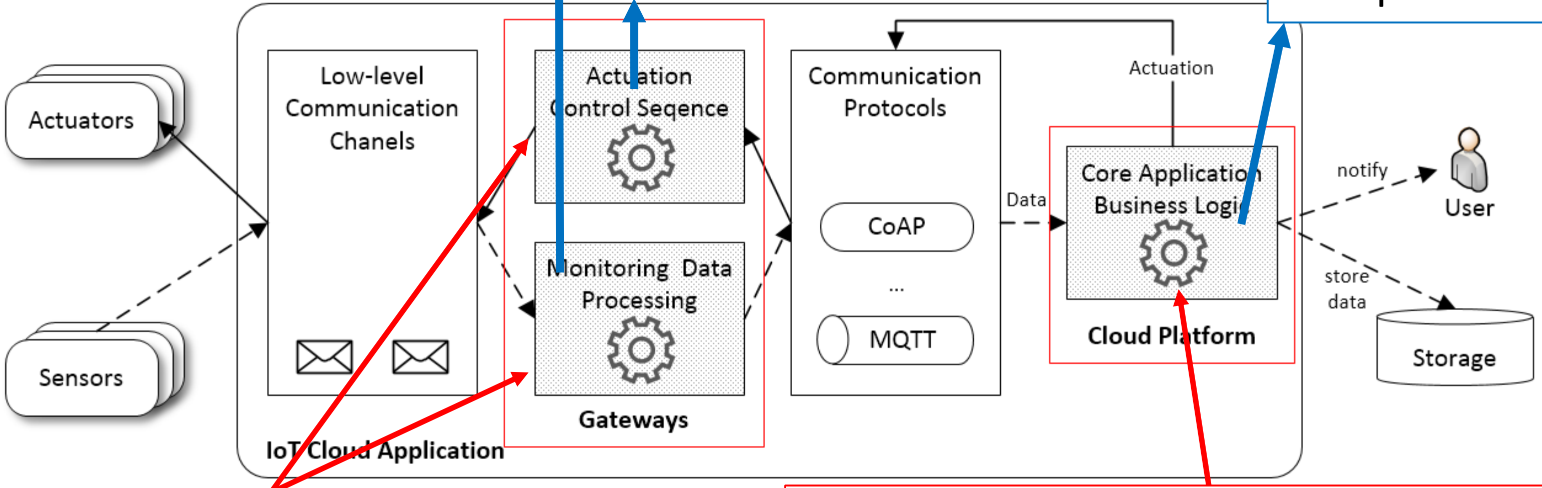
Motivation

Fleet energy usage management

Process energy consumption

Actuation steps to stop vehicle

- Detect energy fault
- Notify manager
- Stop vehicle



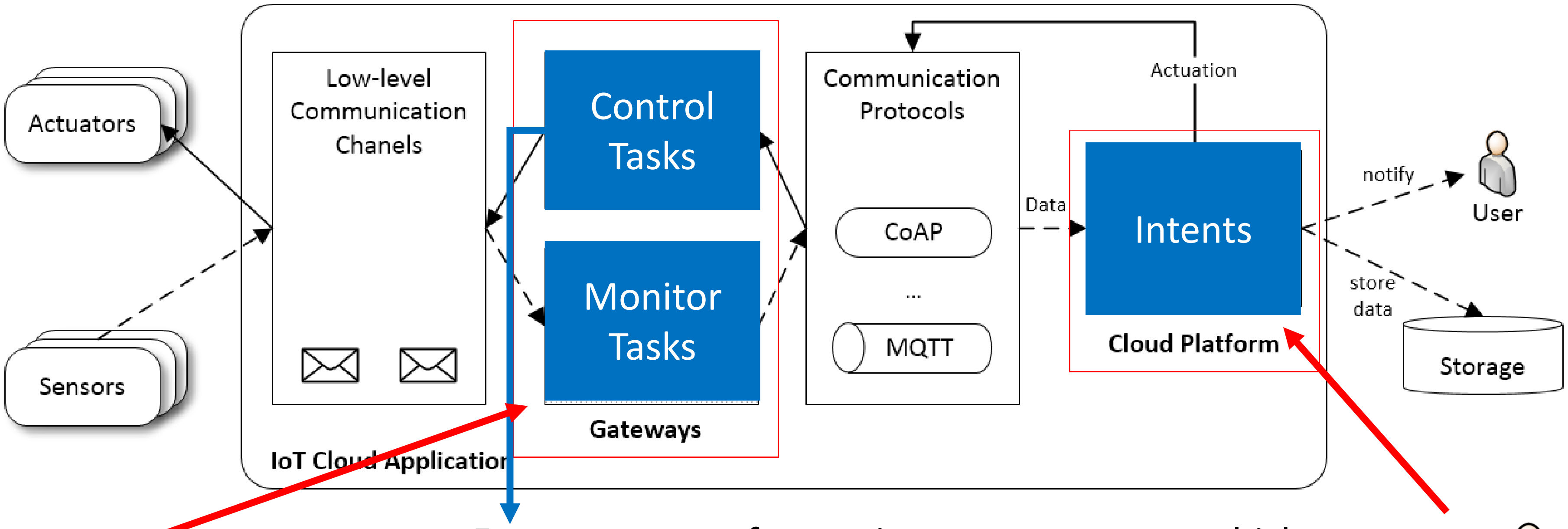
Requirements:

- *Application:* Custom configuration and behavior of Sens./Act.
- *Runtime:* Dealing with constrained resources
- *Developer:* Domain expert knowledge

Requirements:

- *Application:* Should be generic (independent of underlying devices)
- *Runtime:* Dealing with scalability and elasticity concerns
- *Developer:* Software engineering expertise

Approach



E.g., sequence of actuation steps to stop a vehicle

Domain Experts

Task - Encapsulates domain-dependent controls or analytics

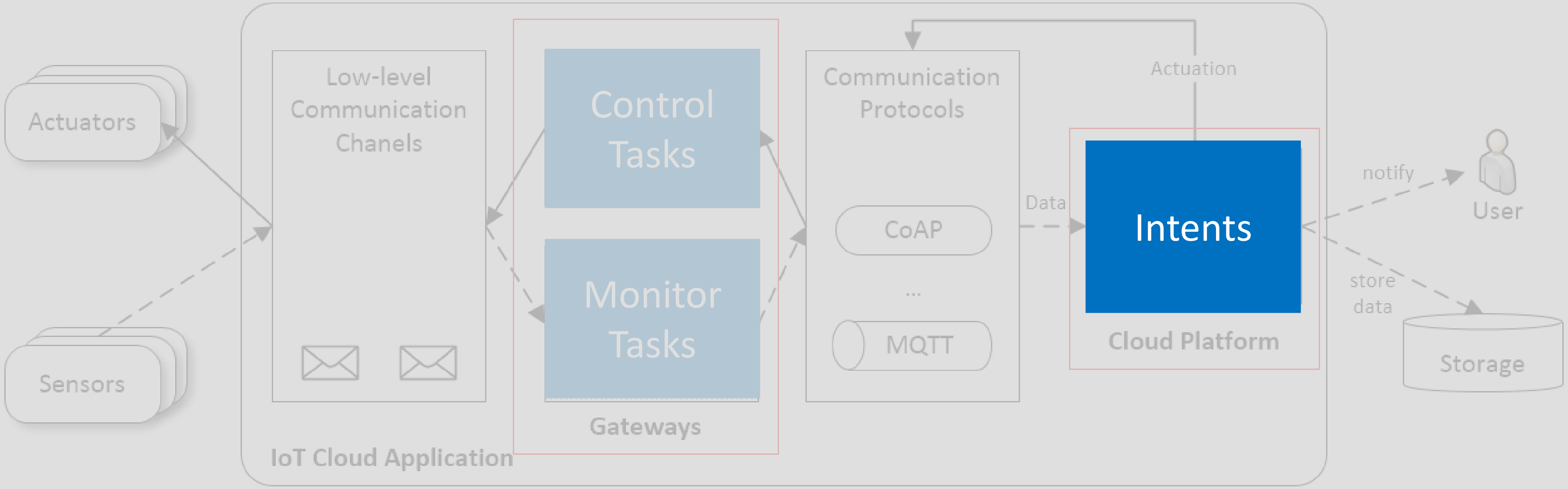
- Packaged into domain-specific libraries (e.g., vehicles management)

Intent - High-level representation of Tasks on Cloud platforms

- Used by developers to remotely invoke Tasks
- Independent of concrete Task implementation

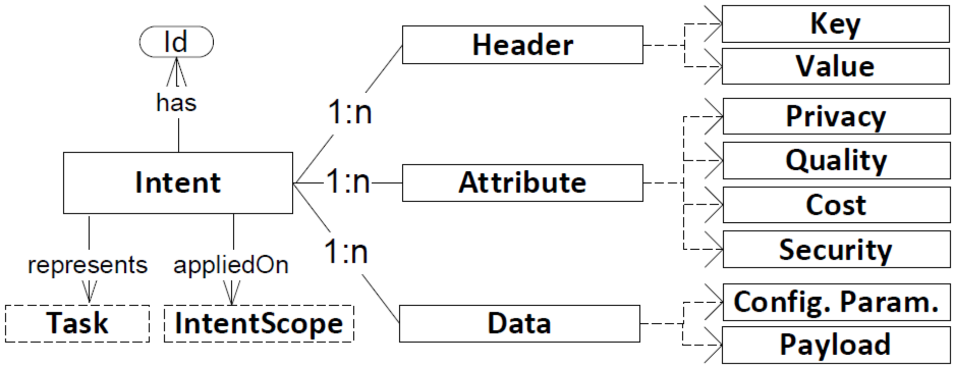
Devlopers

Stefan Nastic, Sanjin Sehic, Michael Vögler, Hong-Linh Truong, and Schahram Dustdar. ***PatRICIA - A Novel Programming Model for IoT Applications on Cloud Platforms.*** **SOCA 2013. Hawaii, USA.**


Actuators
Sensors
Low-level Communication Chanels
Control Tasks
Monitor Tasks
Gateways
Communication Protocols
CoAP
...
MQTT
Actuation
Data
Intents
Cloud Platform
notify
User
store data
Storage
IoT Cloud Application

Intent-based Programming Model

- Passive data structure which declaratively describes intended action, e.g., stop vehicle
- Generic applications (What needs to be done instead how to do it)
- Enable developing loosely coupled applications



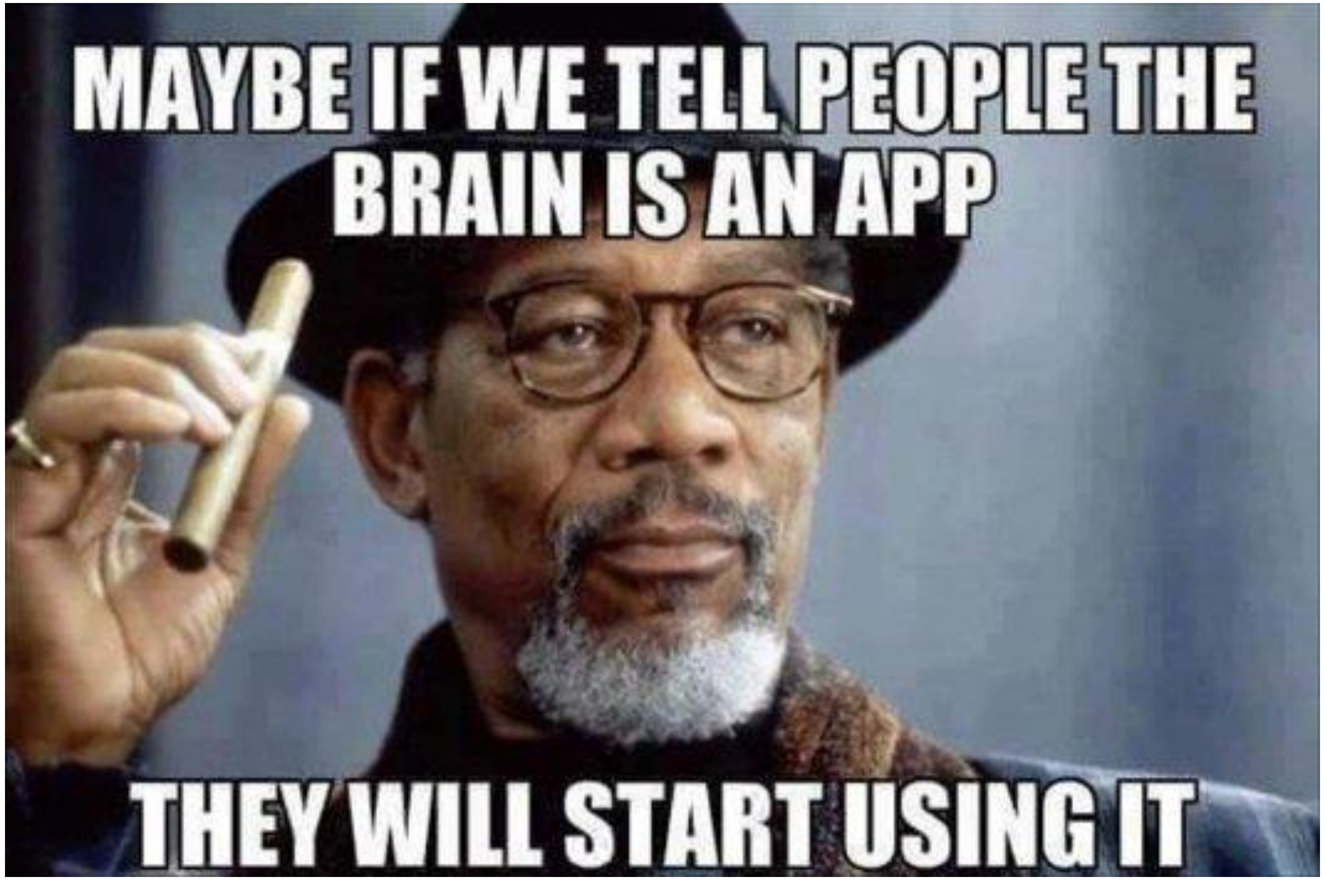
Intent Structure

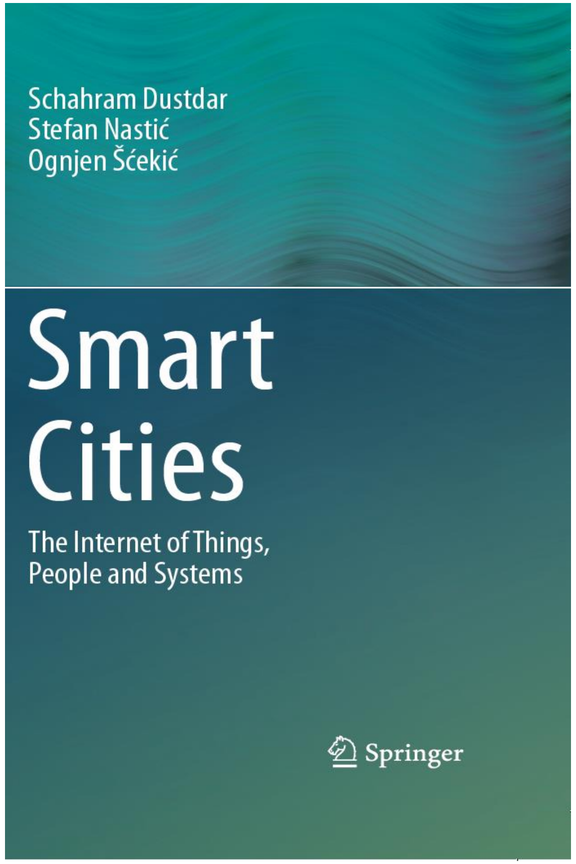
- Trade expressiveness for more flexible and easier application development

Some final reflections

Beyond Turing

- Can a machine-only system really be considered "intelligent"?
  - Going beyond Turing Test... (Alexa, Siri, Cortana)
  - Why not utilize societal intelligence? ... and not try to match the intelligence of a *single* human individual?
- Integrate AI, IoT, and human collectives into processes!





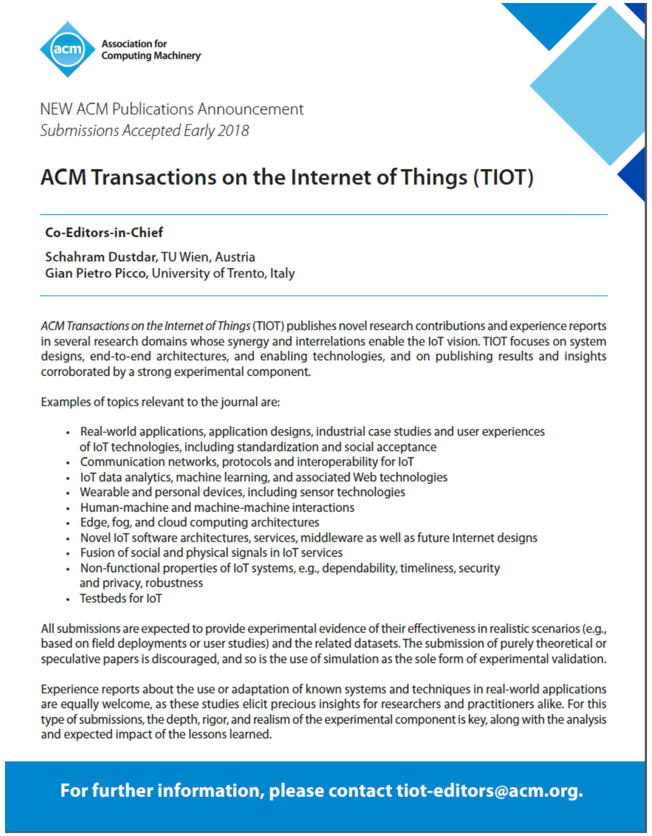
Thanks for your attention

Prof. Schahram Dustdar

IEEE TCSVC Outstanding Leadership Award in Services Computing
Member of Academia Europaea
IBM Faculty award
ACM Distinguished Scientist
IEEE Fellow

Distributed Systems Group
TU Wien

dsg.tuwien.ac.at



Association for Computing Machinery

NEW ACM Publications Announcement
Submissions Accepted Early 2018

ACM Transactions on the Internet of Things (TIOT)

Co-Editors-in-Chief

Schahram Dustdar, TU Wien, Austria
Gian Pietro Picco, University of Trento, Italy

ACM Transactions on the Internet of Things (TIOT) publishes novel research contributions and experience reports in several research domains whose synergy and interrelations enable the IoT vision. TIOT focuses on system designs, end-to-end architectures, and enabling technologies, and on publishing results and insights corroborated by a strong experimental component.

Examples of topics relevant to the journal are:

- Real-world applications, application designs, industrial case studies and user experiences of IoT technologies, including standardization and social acceptance
- Communication networks, protocols and interoperability for IoT
- IoT data analytics, machine learning, and associated Web technologies
- Wearable and personal devices, including sensor technologies
- Human-machine and machine-machine interactions
- Edge, fog, and cloud computing architectures
- Novel IoT software architectures, services, middleware as well as future Internet designs
- Fusion of social and physical signals in IoT services
- Non-functional properties of IoT systems, e.g., dependability, timeliness, security and privacy, robustness
- Testbeds for IoT

All submissions are expected to provide experimental evidence of their effectiveness in realistic scenarios (e.g., based on field deployments or user studies) and the related datasets. The submission of purely theoretical or speculative papers is discouraged, and so is the use of simulation as the sole form of experimental validation.

Experience reports about the use or adaptation of known systems and techniques in real-world applications are equally welcome, as these studies elicit precious insights for researchers and practitioners alike. For this type of submissions, the depth, rigor, and realism of the experimental component is key, along with the analysis and expected impact of the lessons learned.

For further information, please contact tiot-editors@acm.org.