Edge Intelligence

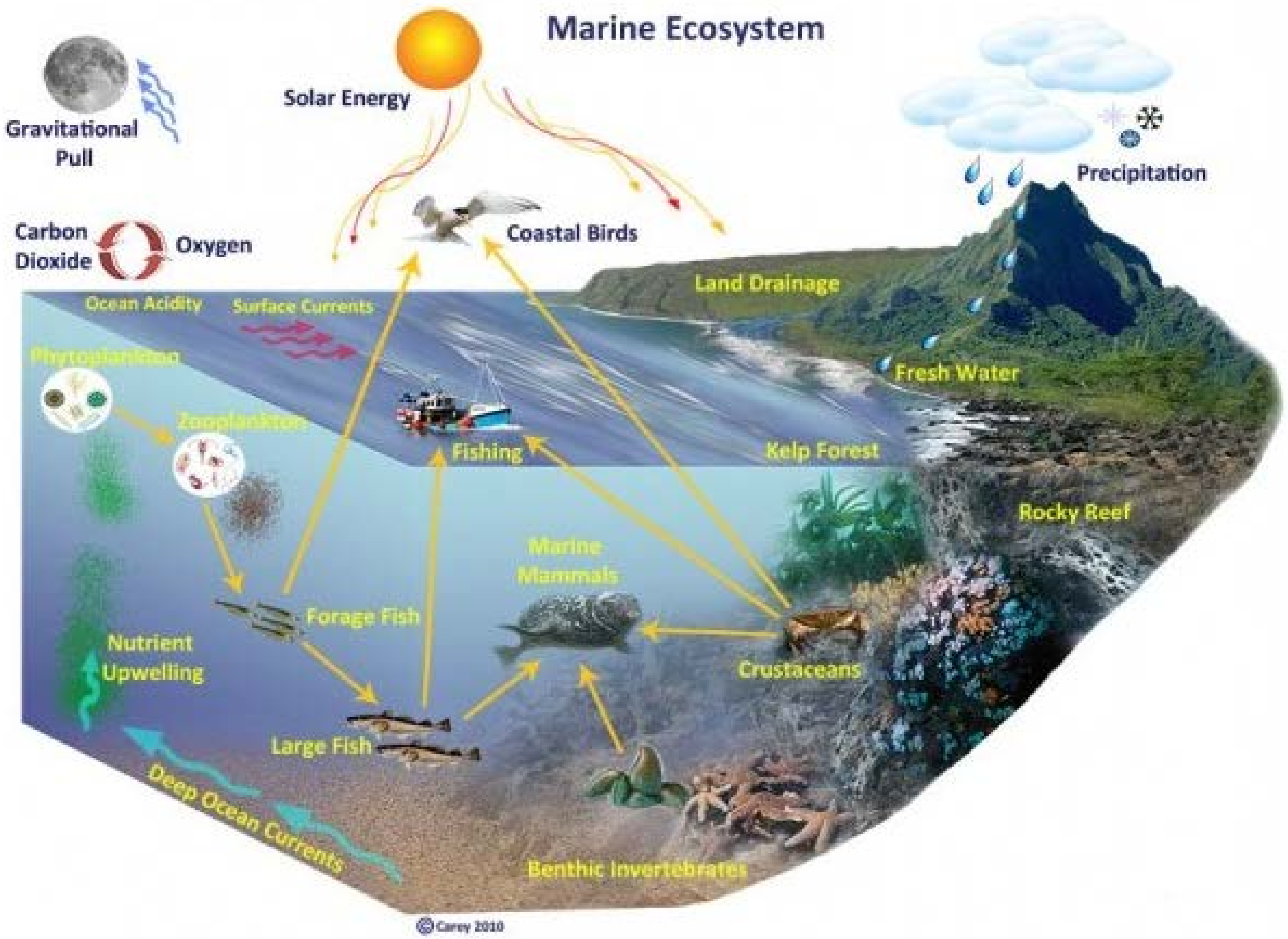
Convergence of Humans, Things, and AI

18 June 2019, SummerSOC, Crete

Schahram Dustdar

dsg.tuwien.ac.at

Ecosystems: People, Systems, and Things

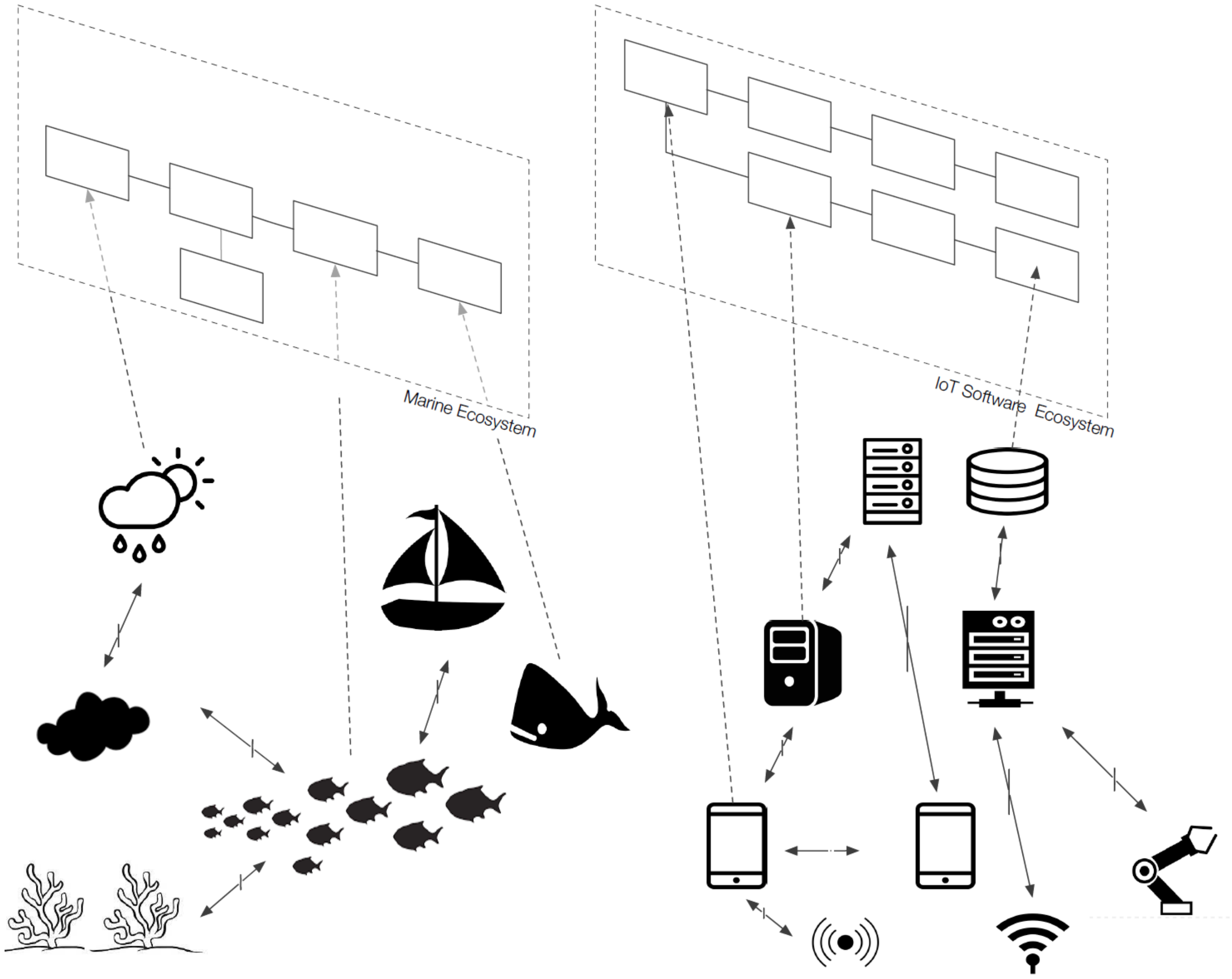


Marine Ecosystem: http://www.xbordercurrents.co.uk/wildlife/marine-ecosystem-2

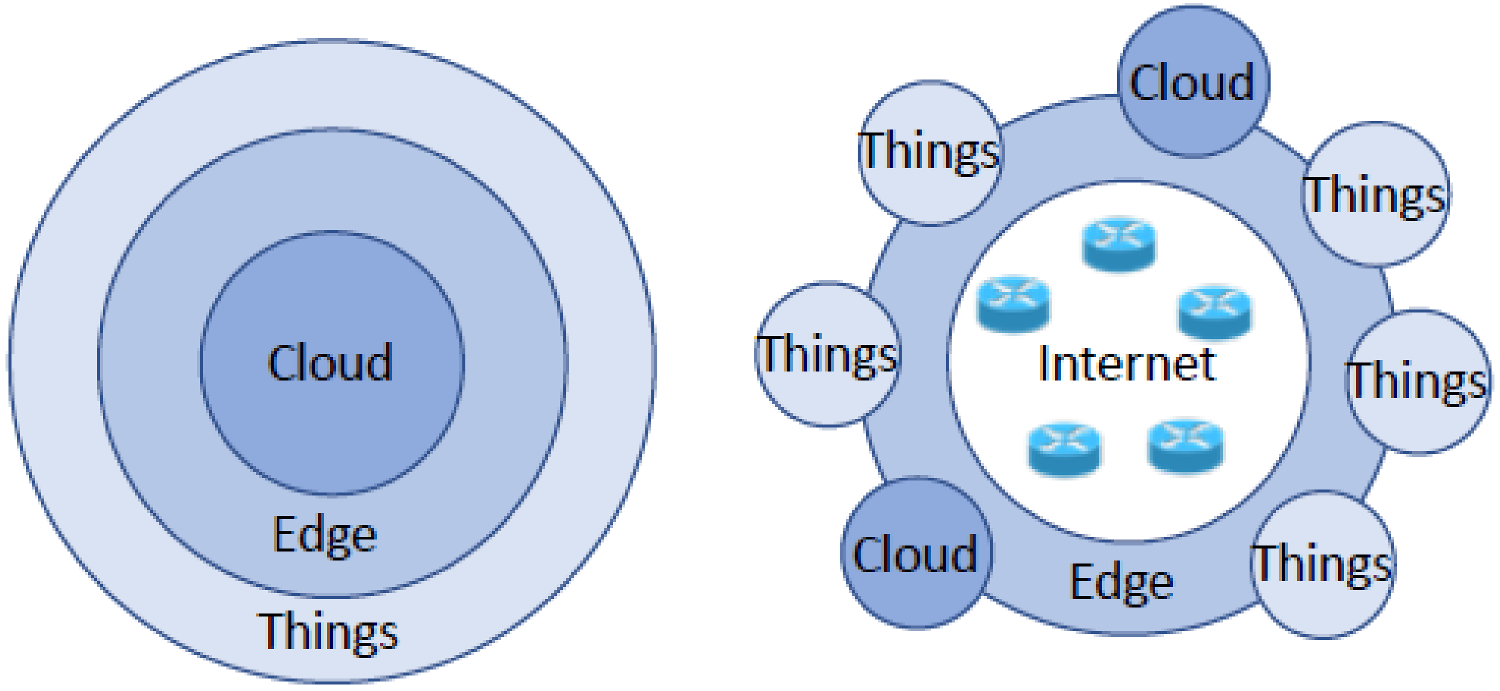
Complex system with networked dependencies and intrinsic adaptive behavior – has:

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

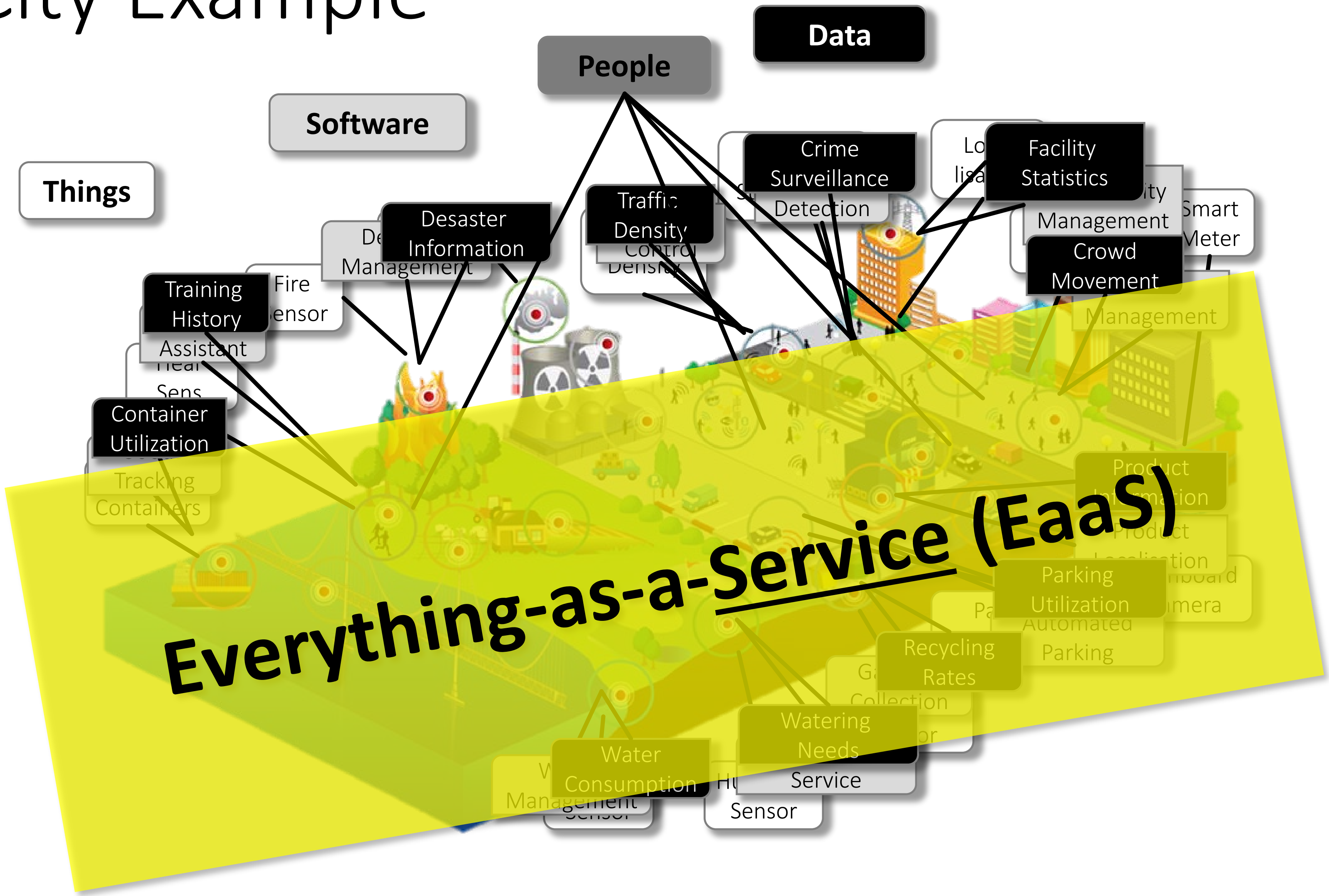
Assumptions

- 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

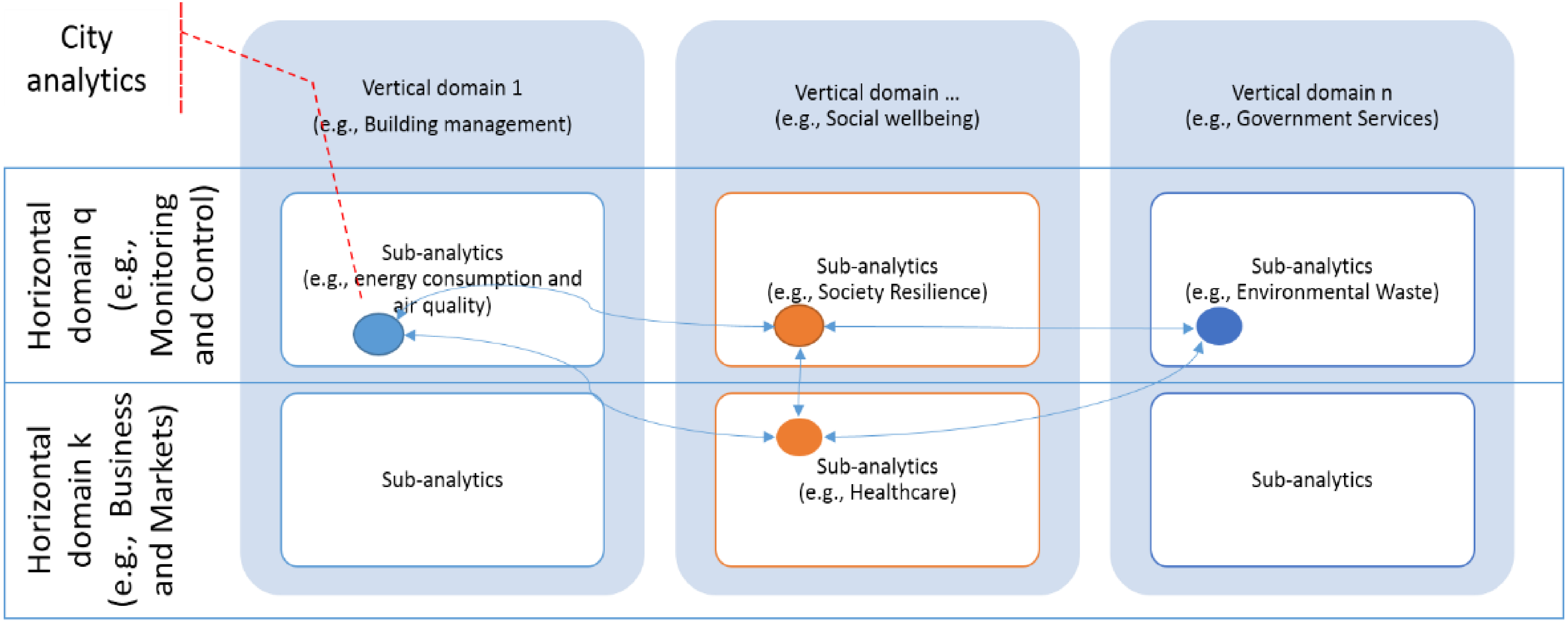
Therefore

- 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

Smart City Example



Dynamic Analytics (e.g., Smart City)

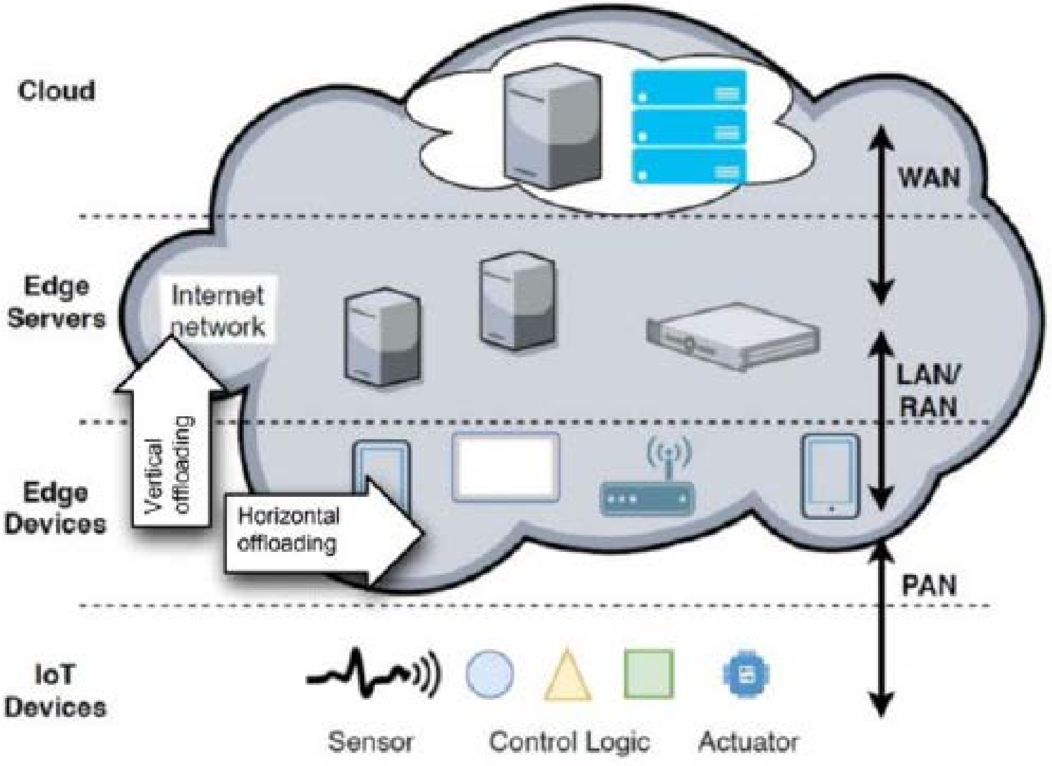


Vertical vs. Horizontal Edge Architecture

Cloud Computing

Fog Computing

Edge Computing



Kim, H., Lee, E.A., Dustdar, S. (2019). Creating a Resilient IoT With Edge Computing, *IEEE Computer 2019* forthcoming

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*

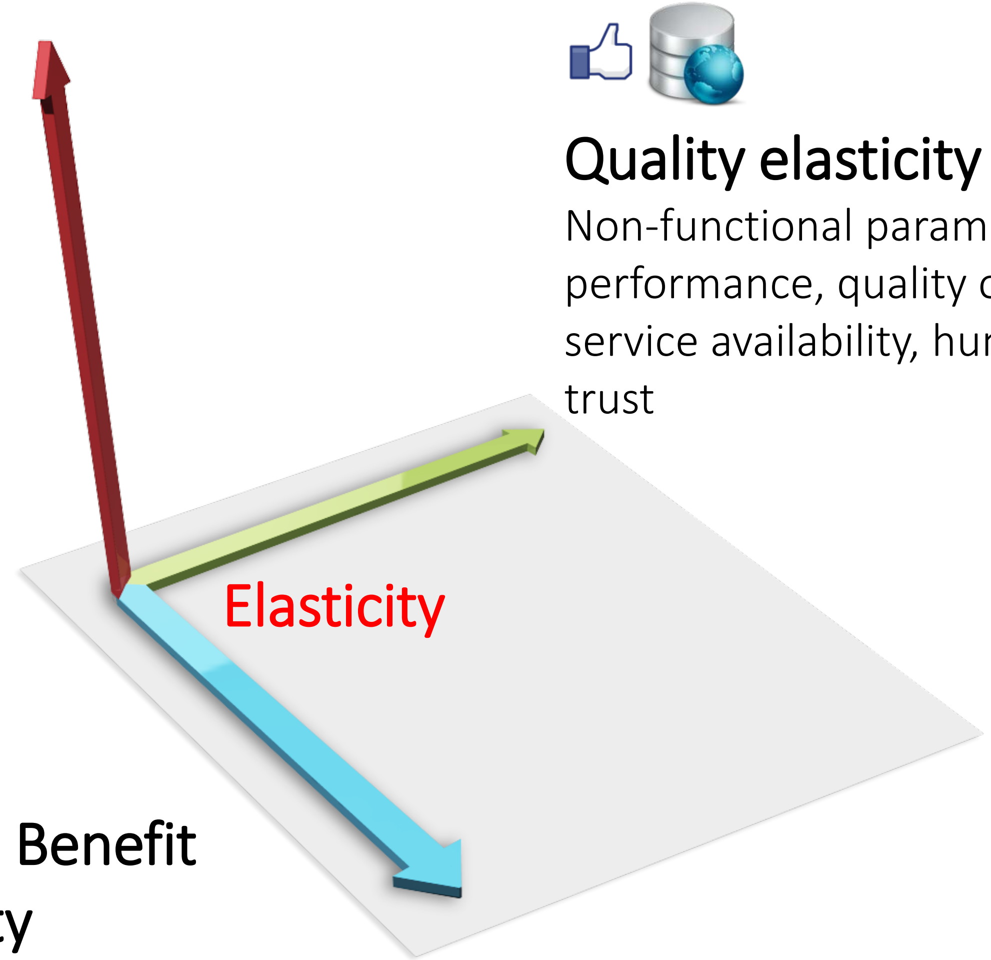
Elastic Computing > Scalability

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

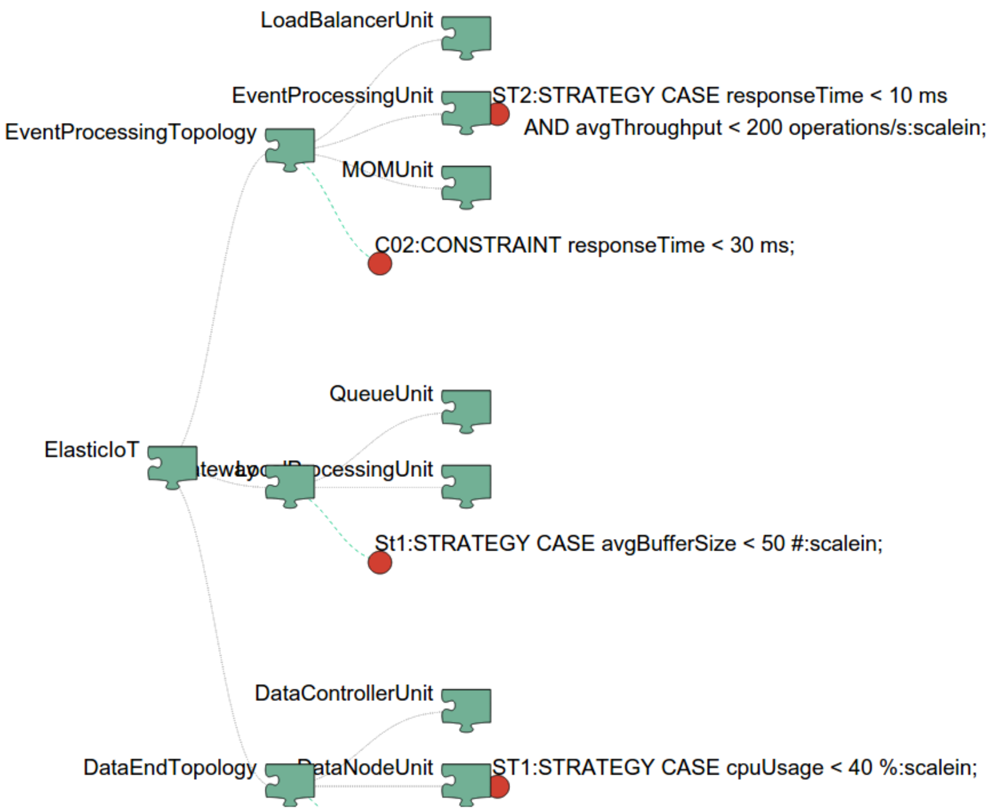
Dustdar S., Guo Y., Satzger B., Truong H. (2012) Principles of Elastic Processes, IEEE Internet Computing, Volume: 16, Issue: 6, Nov.-Dec. 2012

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



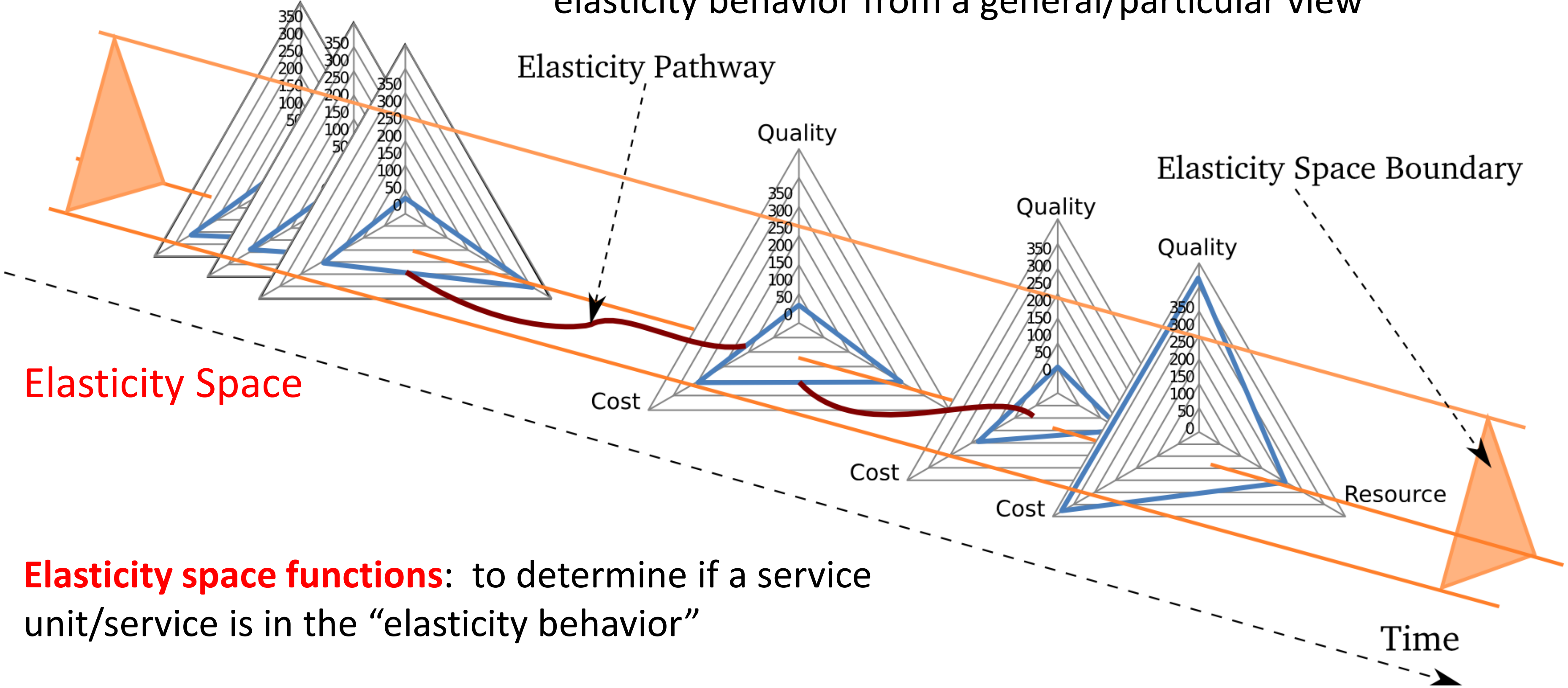
Georgiana Copil, Daniel Moldovan, Hong-Linh Truong, Schahram Dustdar, **"SYBL: an Extensible Language for Controlling Elasticity in Cloud Applications",** 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), May 14-16, 2013, Delft, Netherlands

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**

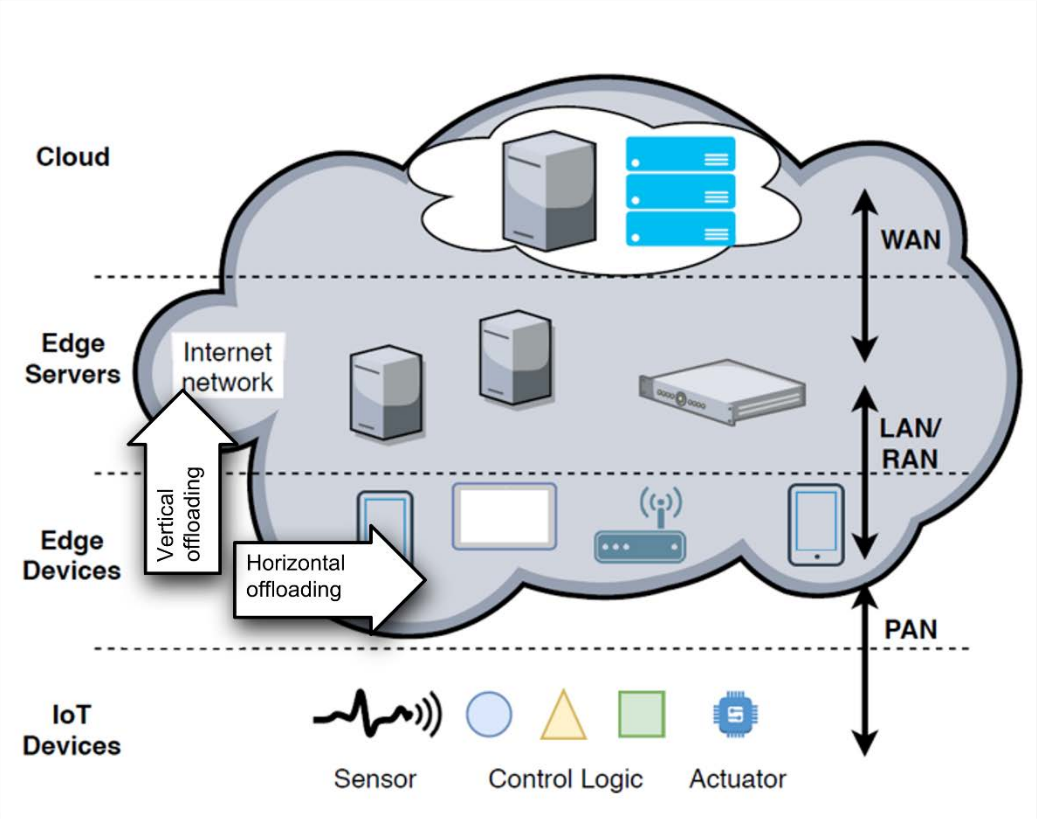
Elasticity Pathway functions: to characterize the elasticity behavior from a general/particular view

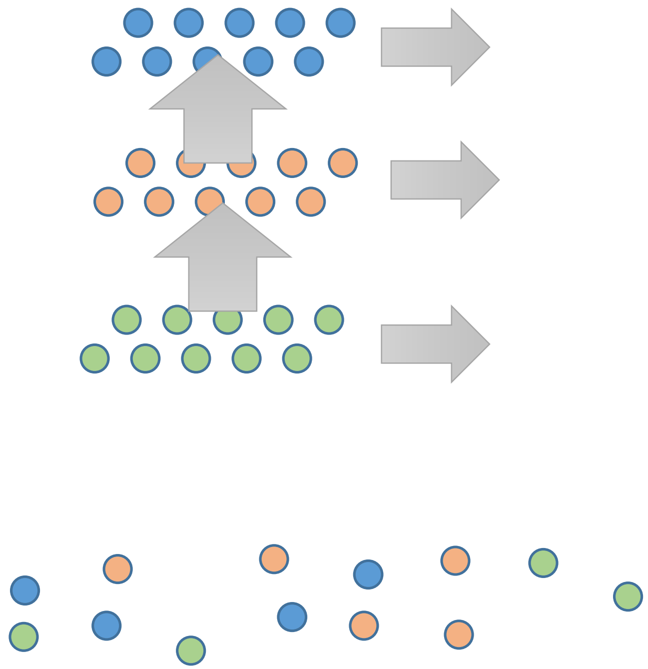


Elasticity space functions: to determine if a service unit/service is in the "elasticity behavior"

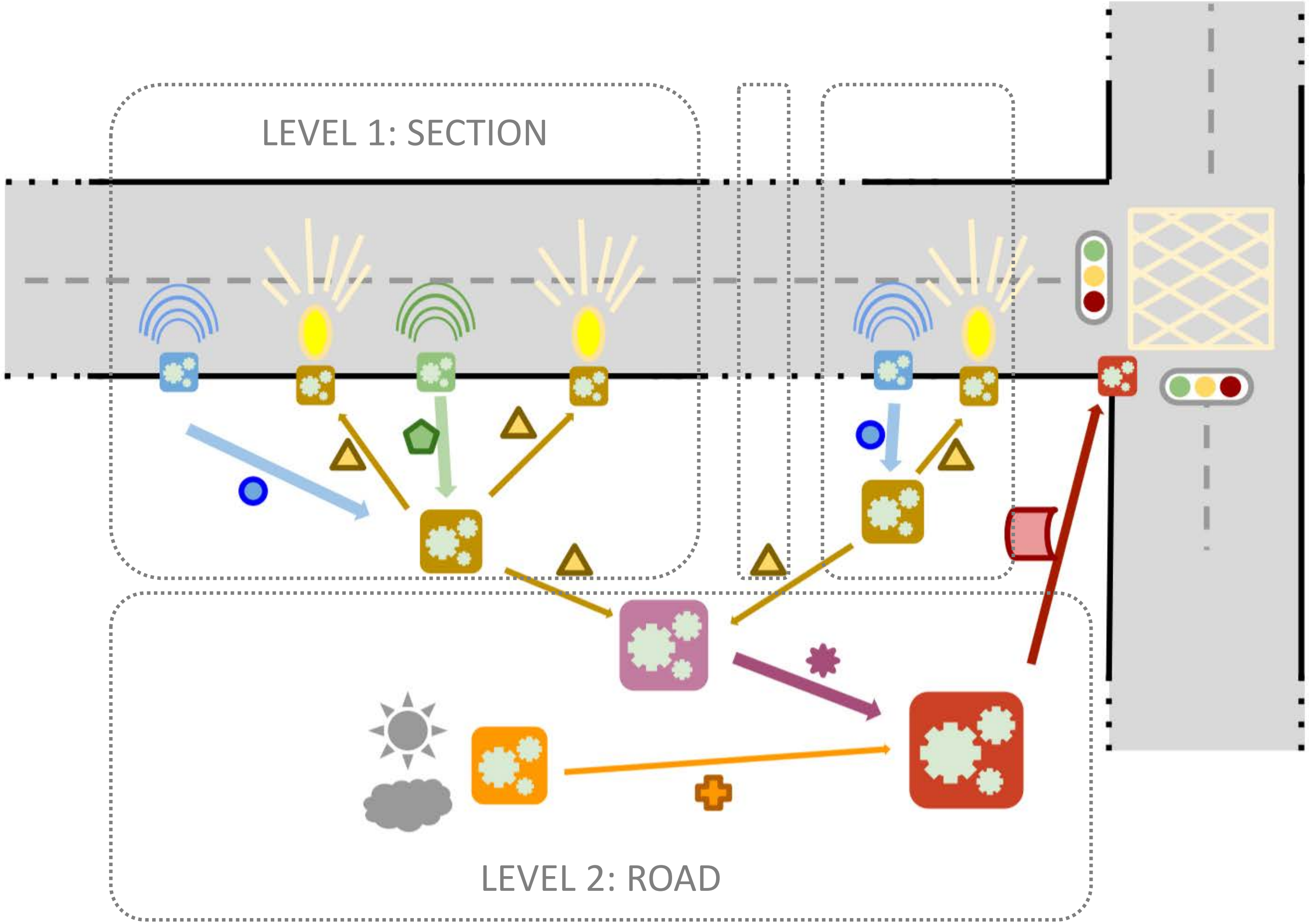
Edge & Blockchains

Dustdar S., Fernandez P., García J. M., Ruiz-Cortés A. (2019). Elastic Smart Contracts across Multiple Blockchains. 2nd International Symposium on Foundations and Applications of Blockchain (FAB 2019), April 5, 2019, Los Angeles, CA, USA

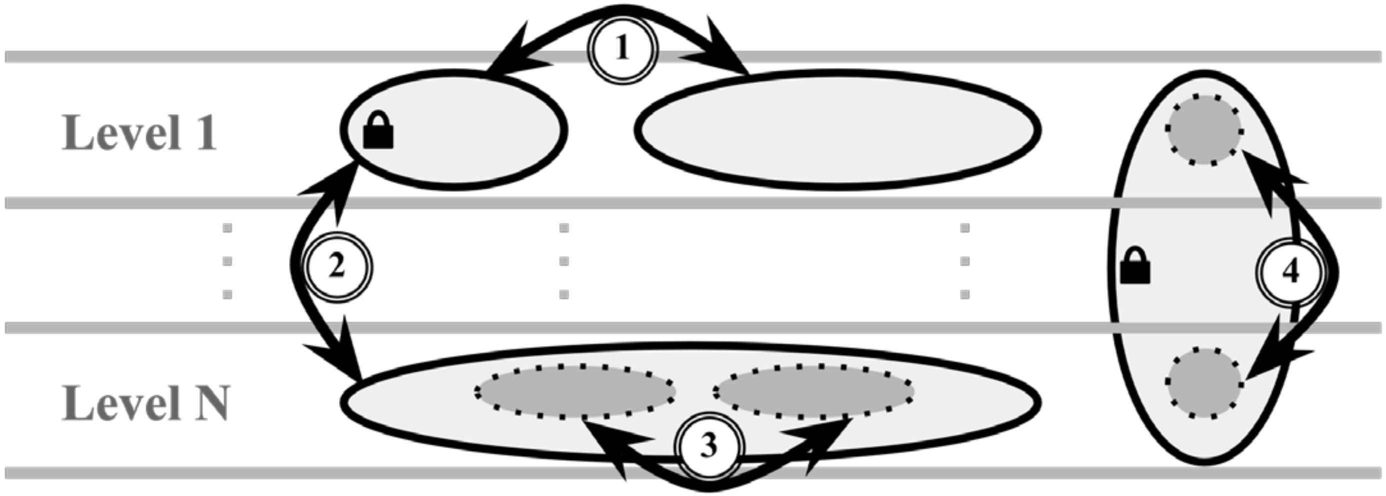




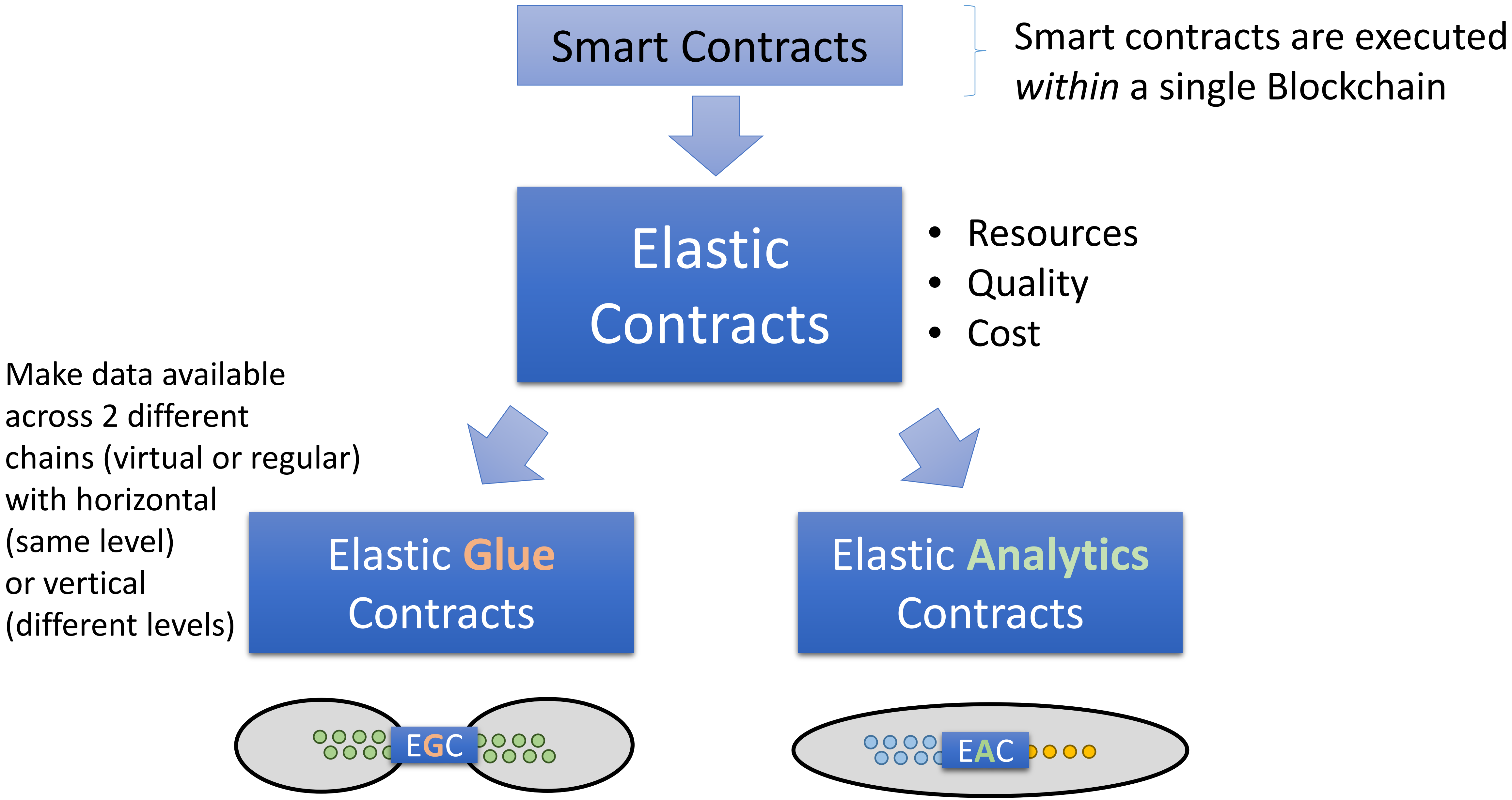
- **Granularity** (Fine/Coarse): Blockchain assigned to single level (fine) or multiple (coarse)
- **Accessibility** (Private / Public): Private BC or public open system; analyze BC by different agents?
- **Deployment Model** (Virtual/Real):Virtual BCs materialized inside a regular BC


LEVEL 1: SECTION
LEVEL 2: ROAD

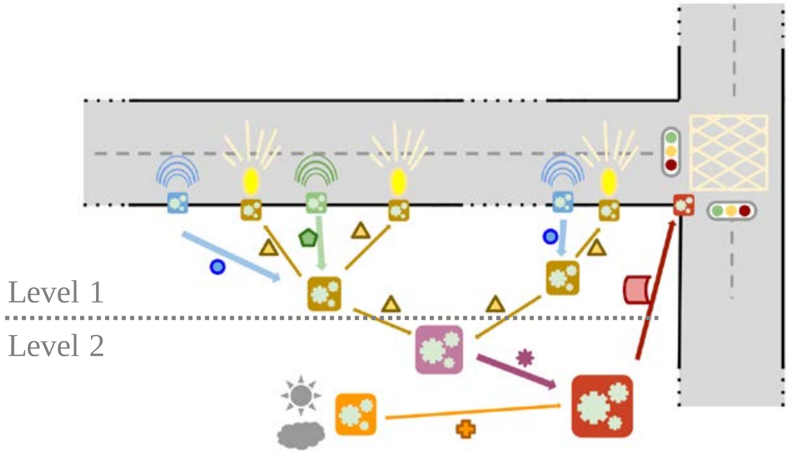
Edge & Blockchains – Integration aspects



	Granularity	Accessibility	Deployment
1	Fine	Hybrid	Regular
2	Coarse	Hybrid	Regular
3	Fine	Public	Virtual
4	Coarse	Private	Virtual


Smart Contracts
Smart contracts are executed *within* a single Blockchain
Elastic Contracts
Resources
Quality
Cost
Make data available across 2 different chains (virtual or regular) with horizontal (same level) or vertical (different levels)
Elastic Glue Contracts
Elastic Analytics Contracts
EGC
EAC

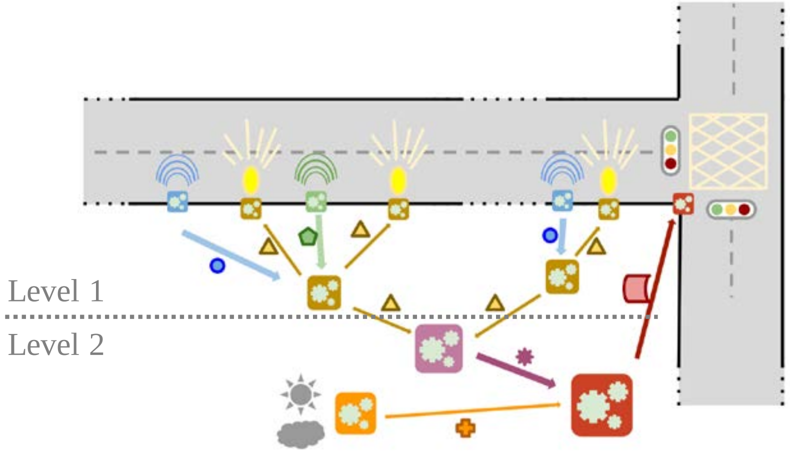
Granularity	Accessibility	Deployment
Coarse	Public	Virtual



Fragments of a Blockchain: Block i to Block i+7

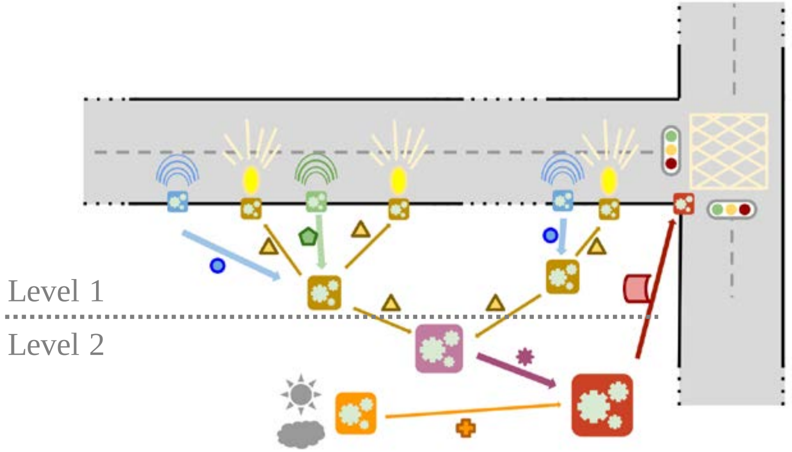
Granularity	Accessibility	Deployment
Coarse	Public	Virtual

Fragments of a Blockchain: Block i to Block i+7



S1: IoT sensor data

Granularity	Accessibility	Deployment
Coarse	Public	Virtual

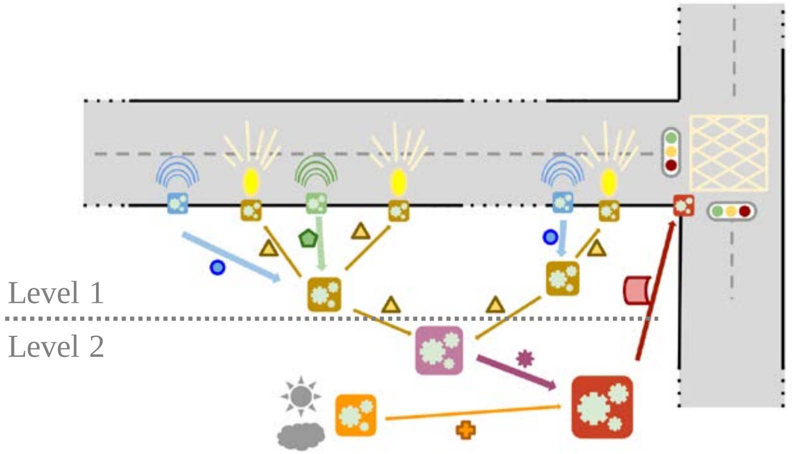


Fragments of a Blockchain: Block i to Block i+7

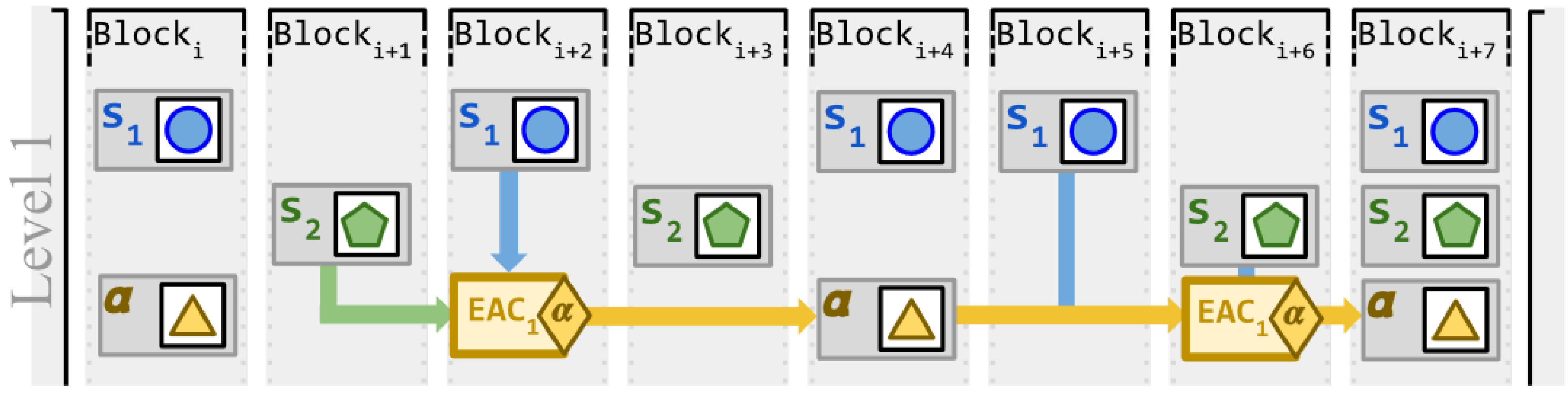
S1: IoT sensor data

S2: IoT sensor data

Granularity	Accessibility	Deployment
Coarse	Public	Virtual



Fragments of a Blockchain: Block i to Block i+7



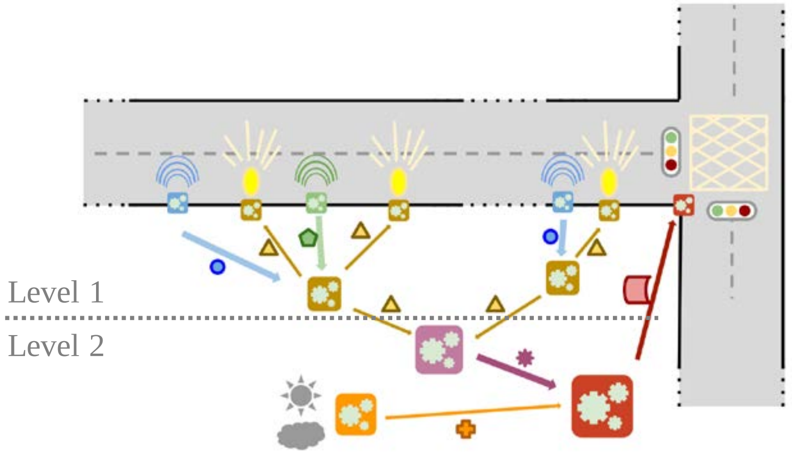
S1: IoT sensor data

S2: IoT sensor data

EAC: Elastic Analytics Contract

e.g. creating a service for “presence prediction” (alpha) for the next steps

Granularity	Accessibility	Deployment
Coarse	Public	Virtual



EGC: **Elastic Glue Contract**

2 types of EGCs:

a) **EGC1**: aggregates Level 1 info + brings it as new data to Level 2

a) **EGC2**: (implemented as an oracle) imports data off-chain to Level 2

Towards Edge Intelligence

Computational Fabric

- dispersed resources allow training, monitoring, serving of models
- Heterogeneity of applications and models requires
  - (1) flexible and modular infrastructure and
  - (2) intelligent operations mechanisms (due to the scale of the infrastructure)

Operationalization

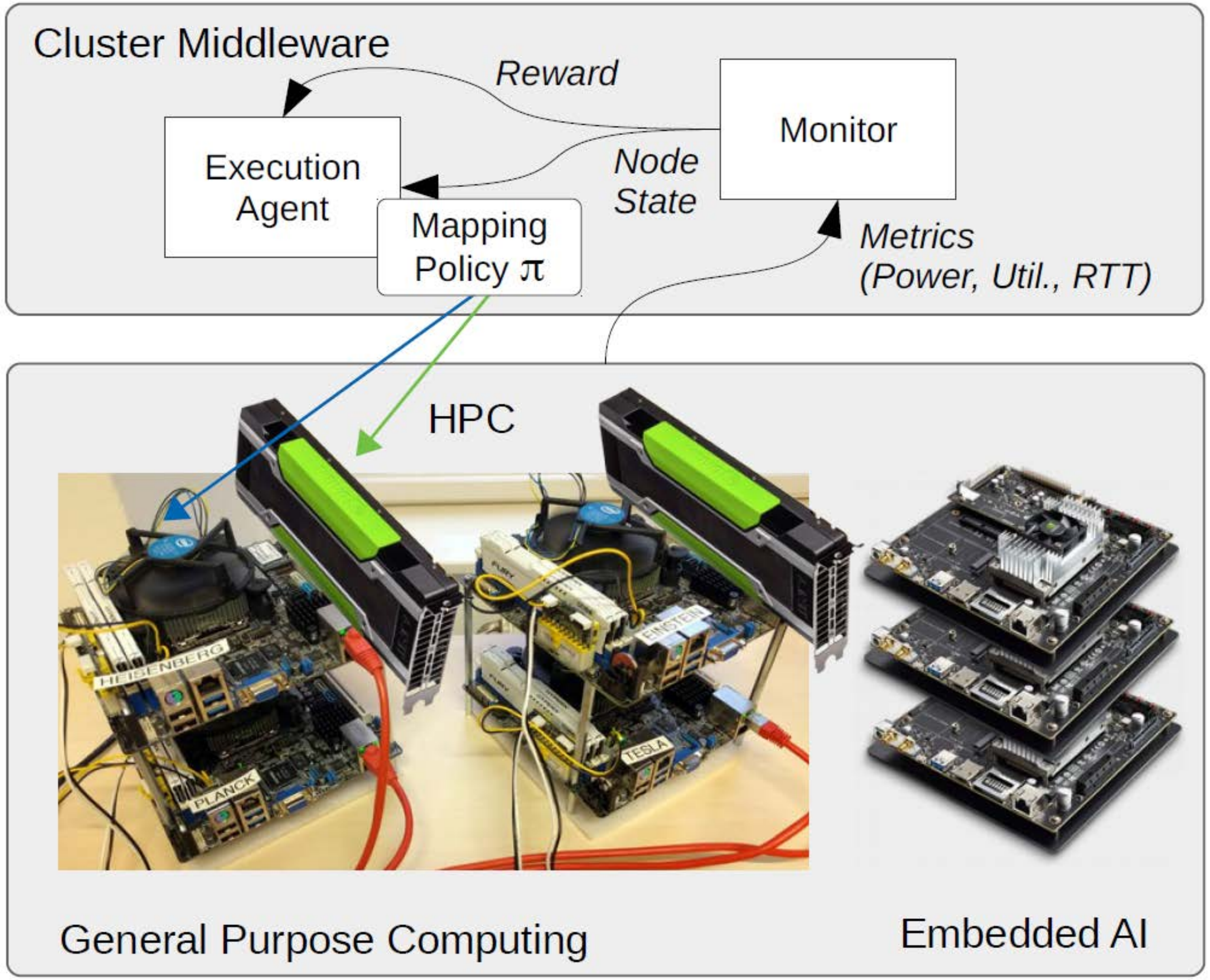
- Automated AI application lifecylce management to the Edge

Rausch, T., Dustdar, S. (2019). Edge Intelligence: The Convergence of Humans, Things, and AI. In *IEEE International Conference on Cloud Engineering (IC2E) 24-27 June 2019*.

Fabric for Edge Intelligence

1. **Sensing (Sensor Data as a Service)**
   - Large number, dynamic and mobile nature of publishers/subscribers of sensor data + QoS requirements of edge intelligence ->> rethink centralized messaging services such as AWS IoT or MS Azure IoT Hub
   - Management and governance of such a distributed/decentralized sensing infrastructure
2. **Edge computer network with modular AI capabilities**
   - New AI accelarators for edge devices (e.g., Google Edge TPU with an aplication specific integrated circuit; MS BrainWave with field-programmable gate arrays (FPGAs); Intel Neural Compute Stick; Baidu Kunlun, Huawei Atlas AI Platform
3. **Intelligent orchestration mechanisms for decentralized and distributed infrastructure**

Edge self-adaptive middleware & scheduling Wfs



Rausch T., Avasalcai C., Dustdar S. (2018). Portable Energy-Aware Cluster-Based Edge Computers. [3rd ACM/IEEE Symposium on Edge Computing (SEC 2018)], October 25-27, 2018, Bellevue, WA, USA

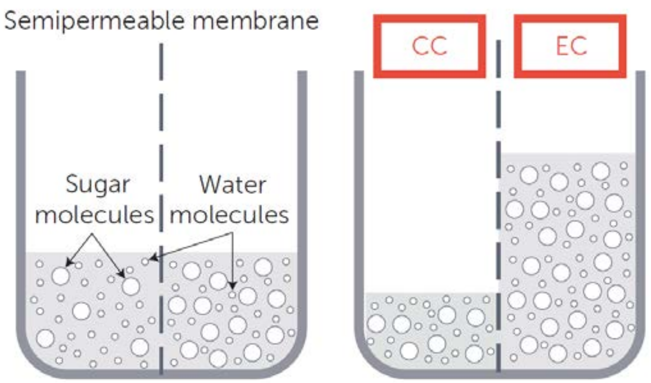
Rausch T., Dustdar S., Ranjan R. (2018). [Osmotic Message-Oriented Middleware for the Internet of Things]. *IEEE Cloud Computing*, Volume 5, Issue 2, pp. 17-25

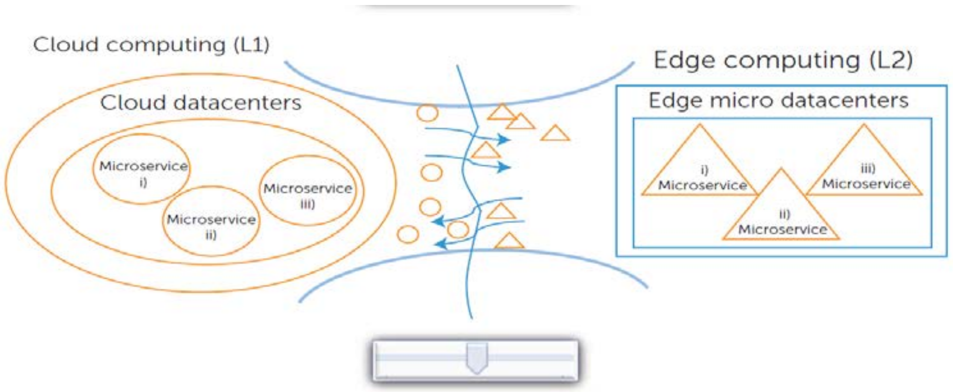
Nastic S., Rausch T., Scekic O., Dustdar S., Gusev M., Koteska B., Kostoska M., Jakimovski B., Ristov S., Prodan R. (2017). [A Serverless Real-Time Data Analytics Platform for Edge Computing]. *IEEE Internet Computing*, Volume 21, Issue 4, pp. 64-71

Paradigm 2: Osmotic Computing

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

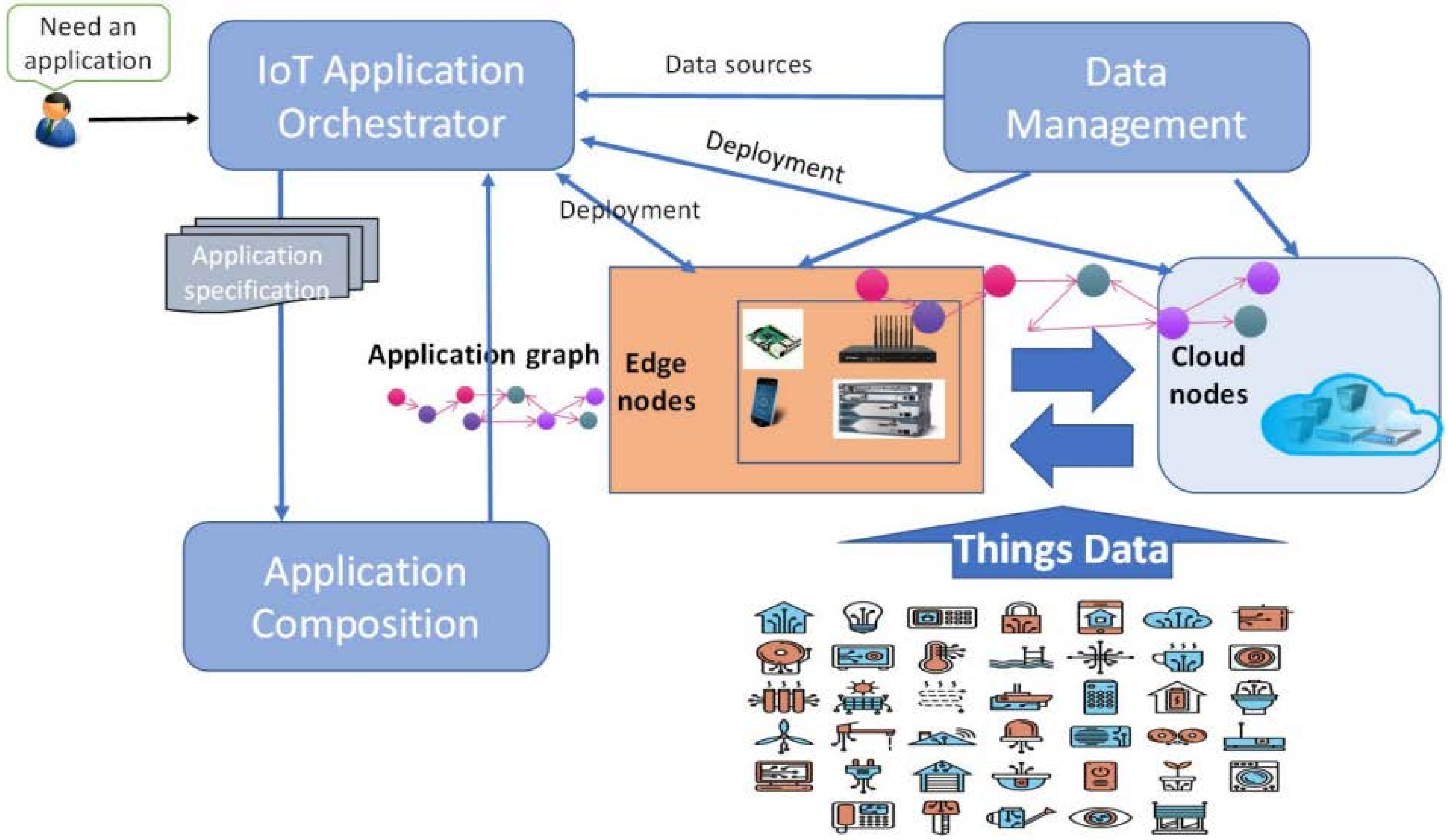
- Dynamic management of (micro)services across cloud and edge datacenters
  - deployment, networking, and security, …
  - providing reliable IoT support with specified levels of QoS.



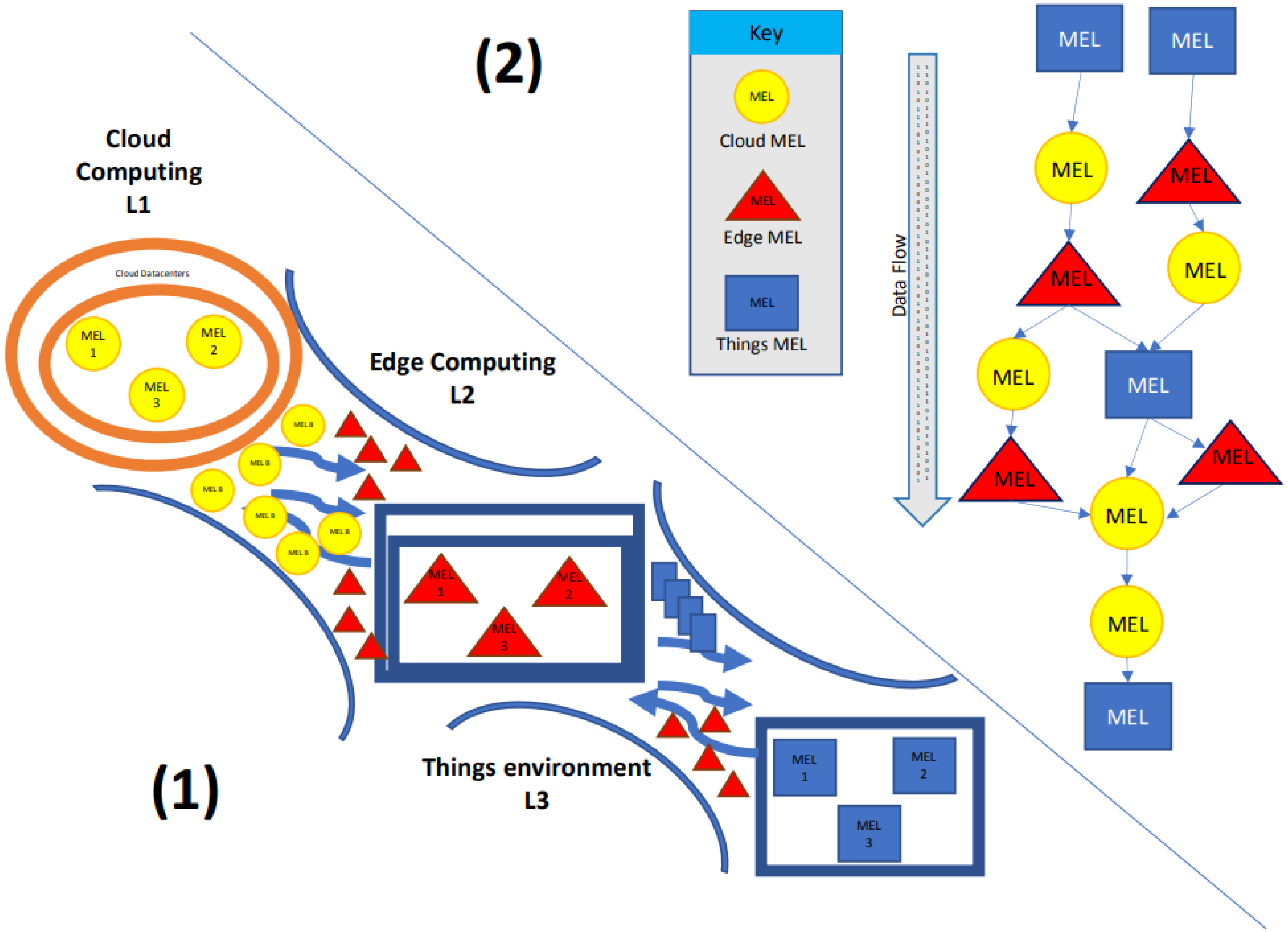


Villari M., Fazio M., Dustdar S., Rana O., Ranjan R. (2016). Osmotic Computing: A New Paradigm for Edge/Cloud Integration. *IEEE Cloud Computing*, Volume 3, Issue 6, pp. 76-83

IoT/Data/Application Orchestration



Osmotic movement of MELs in Clouds, Edge, Things



Legend:
MEL...Micro Element

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 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 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;

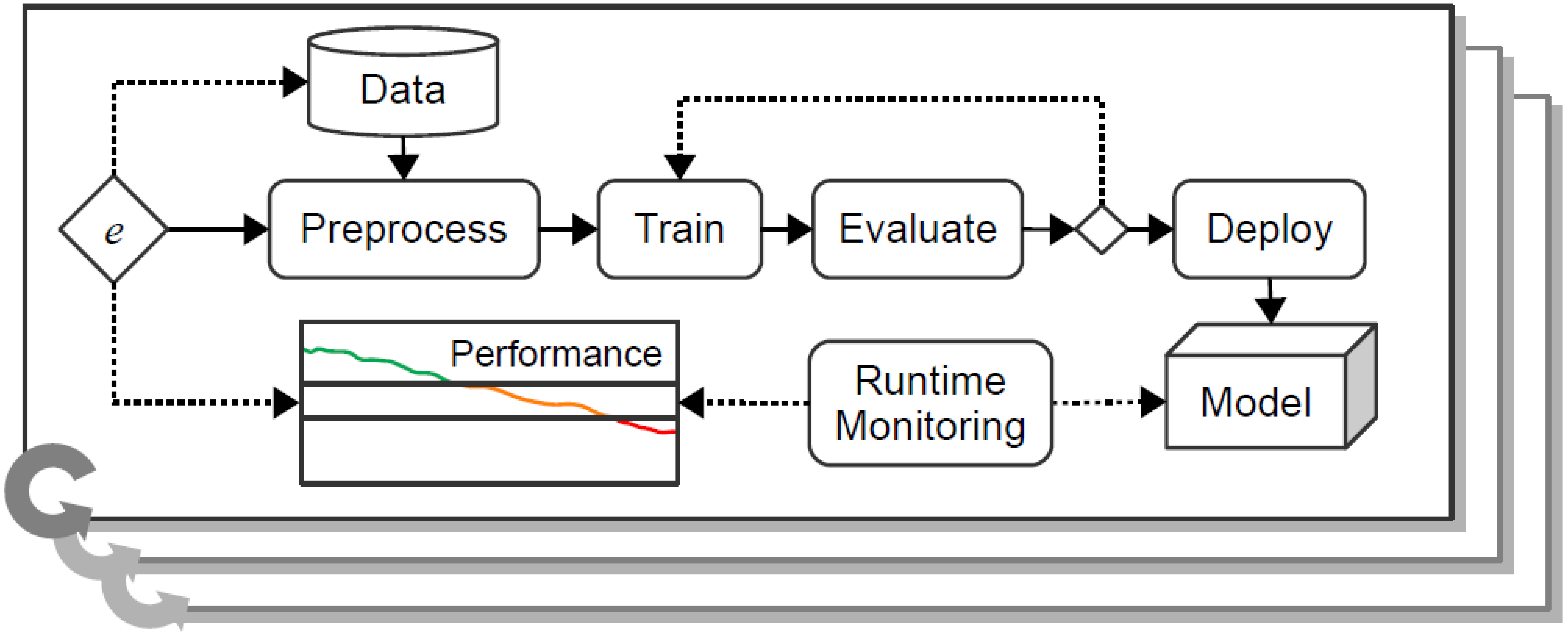
2. Identifying and integrating **real-time data from IoT devices and historical IoT data** distributed across Cloud and Edge resources;

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).

Managing the AI Lifecycle

AI lifecycle pipeline with a rule-based trigger *e* that monitors available data and runtime performance data to form an automated retraining loop



AI Operations Workflows – Edge to Cloud

	Data characteristics	Model characteristics	Enabling technologies	Example use cases
C2C	- Training data is centralized - Massive data sets	- Models are large - Huge number of inferencing requests need to be load balanced	- Scalable learning infrastructure [39] - Data warehousing	- Image search - Recommender systems
C2E	- Training data is centralized - Inferencing data may be sensitive	- Inferencing may need to happen in near-real time - Large number of model deployments - Models run on specialized hardware	- Model compression [42] - Latency/accuracy tradeoff [43] - Distributed inferencing [44] - Transfer learning [45]	- Surveillance systems - Self driving cars - Fieldwork assistants
E2C	- Training data is distributed - Training data may be sensitive	- Models can be centralized - Huge number of inferencing requests need to be load balanced	- Decentralized/federated learning [41]	- Volunteer computing - Novel Smart City use cases
E2E	- Training data is distributed - Training and inferencing data may be sensitive	- Inferencing may need to be near-real time	- Decentralized/federated learning - Distributed inferencing	- Industrial IoT (e.g., predictive maintenance) - Privacy-aware personal assistants - Novel IoT use cases

Rausch, T., Dustdar, S. (2019). Edge Intelligence: The Convergence of Humans, Things, and AI. In *IEEE International Conference on Cloud Engineering (IC2E) 24-27 June 2019*.

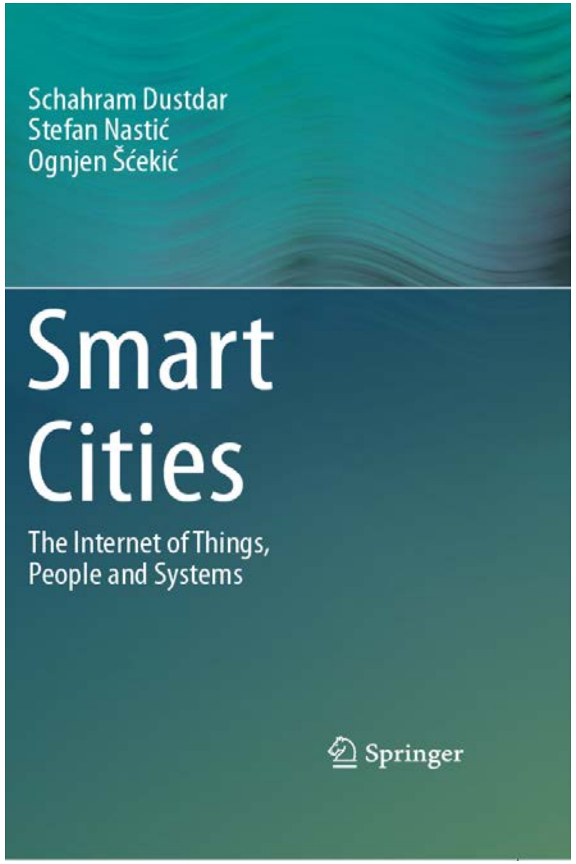
Conclusions

- Need for an Edge Intelligence AI Fabric and a “clear“ ecosystems understanding

- Better levels of integration between multiple (levels of) Blockchains and stakeholders can be achieved

- Integrate AI, IoT, and human collectives into processes distributed on the Edge and Cloud

Thanks for your attention



Prof. Schahram Dustdar

IEEE TCSVC Outstanding Leadership Award in Services Computing

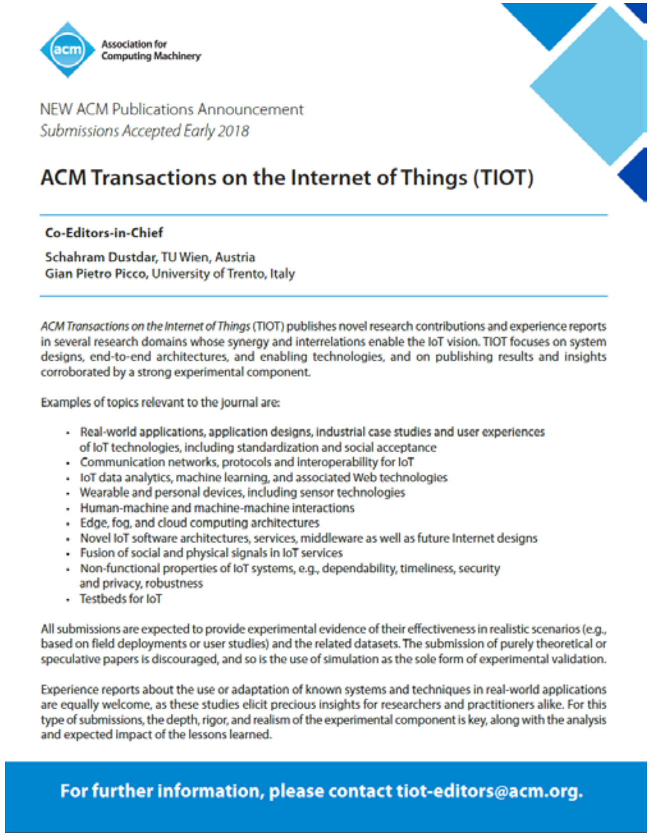
Member of Academia Europaea

IBM Faculty award

ACM Distinguished Scientist

IEEE Fellow

Distributed Systems Group
TU Wien, Austria
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.