IoT Systems Engineering

Tutorial

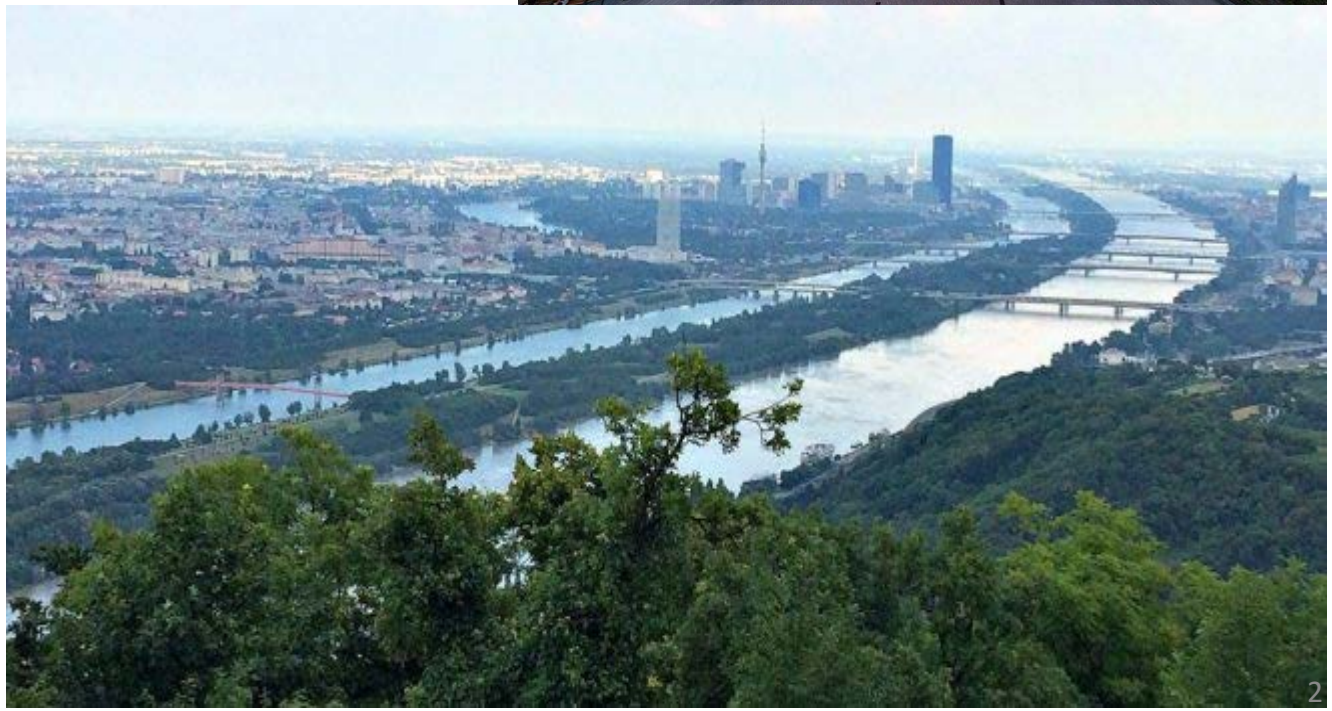
24 April 2018, CIbSE 2018, Bogotá, Colombia

Schahram Dustdar

Distributed Systems Group
TU Wien

dsg.tuwien.ac.at

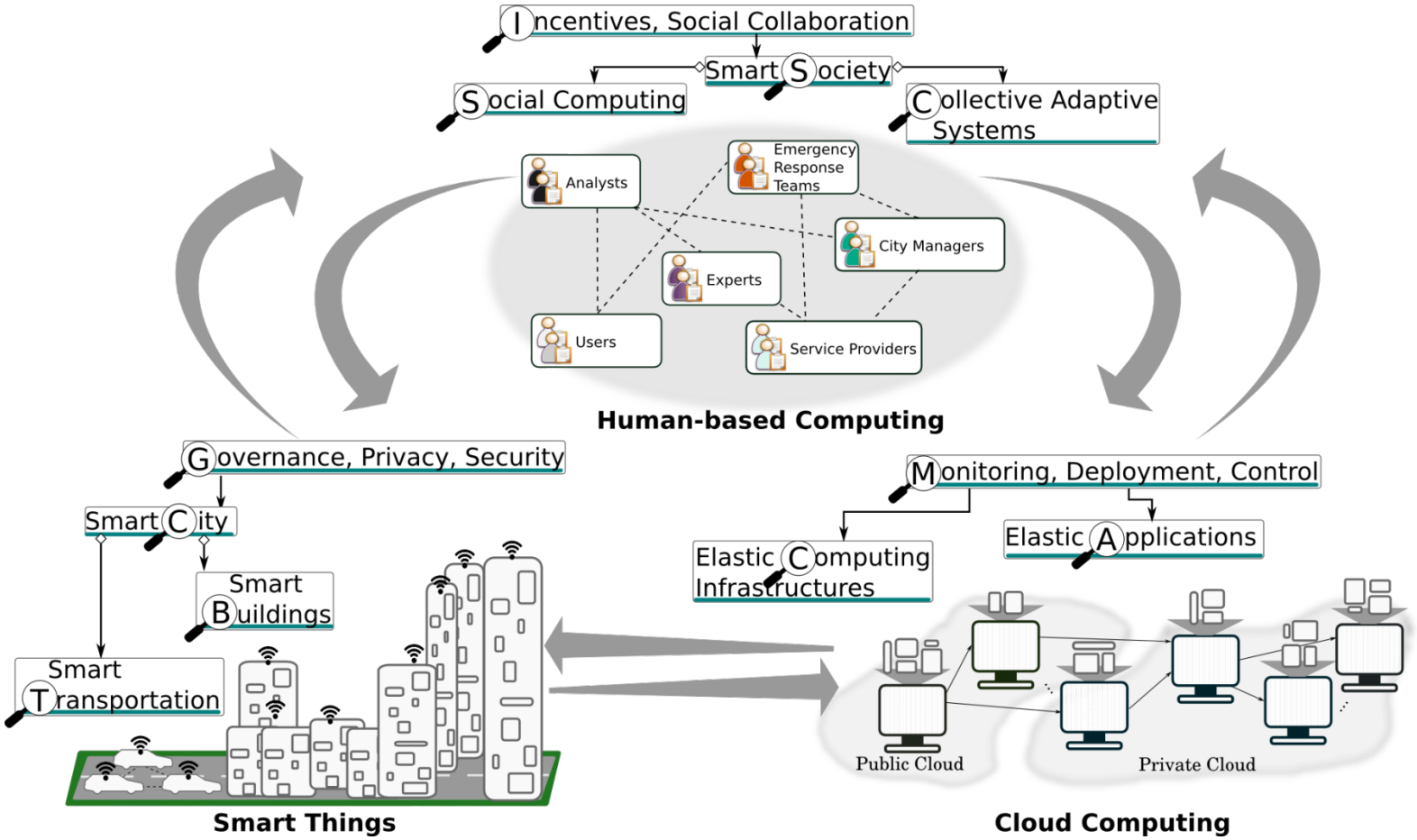
Vienna, Austria

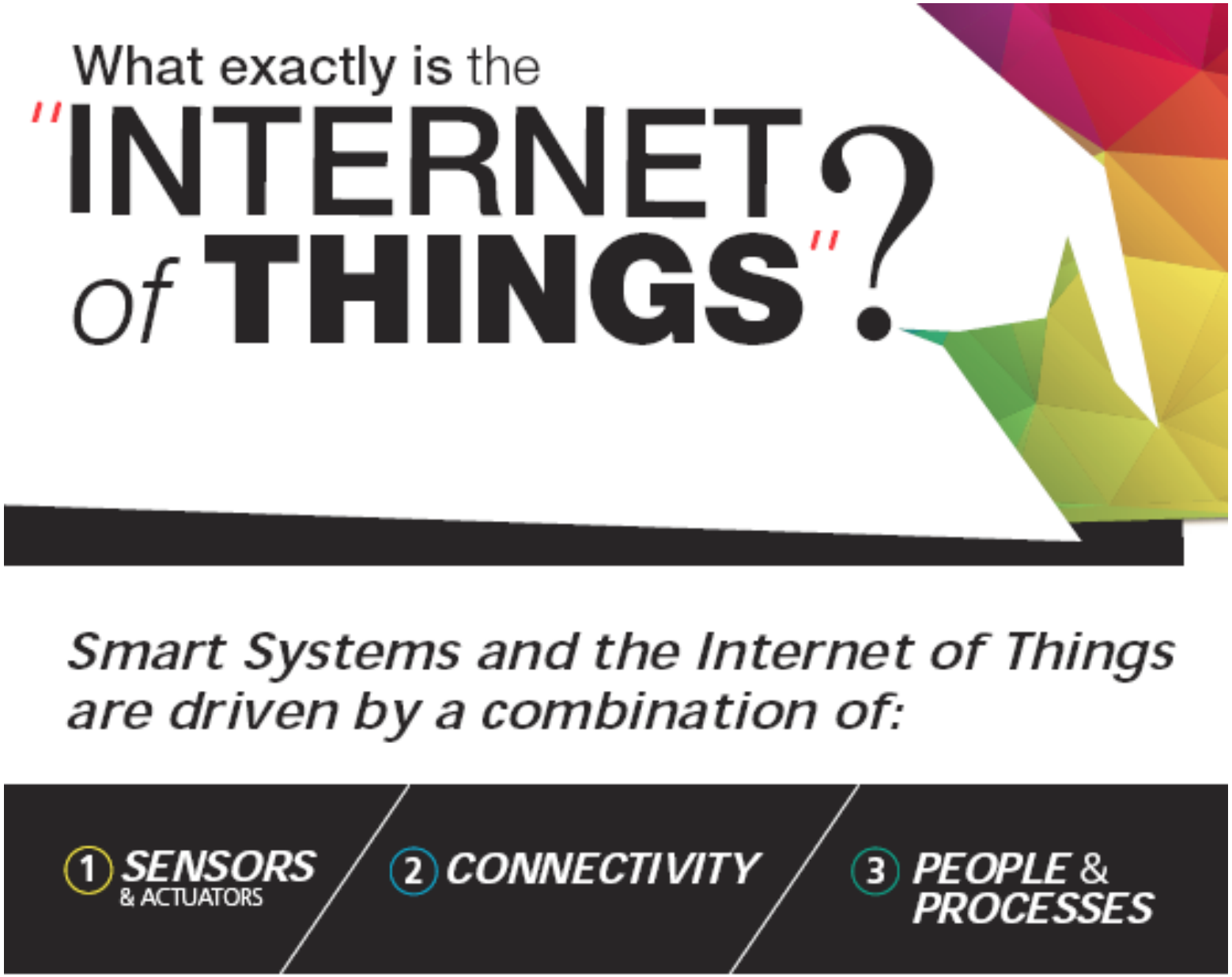


Between 20 and 35 members in the Distributed Systems Group

Multi-cultural Background of members over the years: e.g., Austria, Germany, Italy, Kroatia, Serbia, Macedonia, Romania, Ukraine, Pakistan, Vietnam, China,...

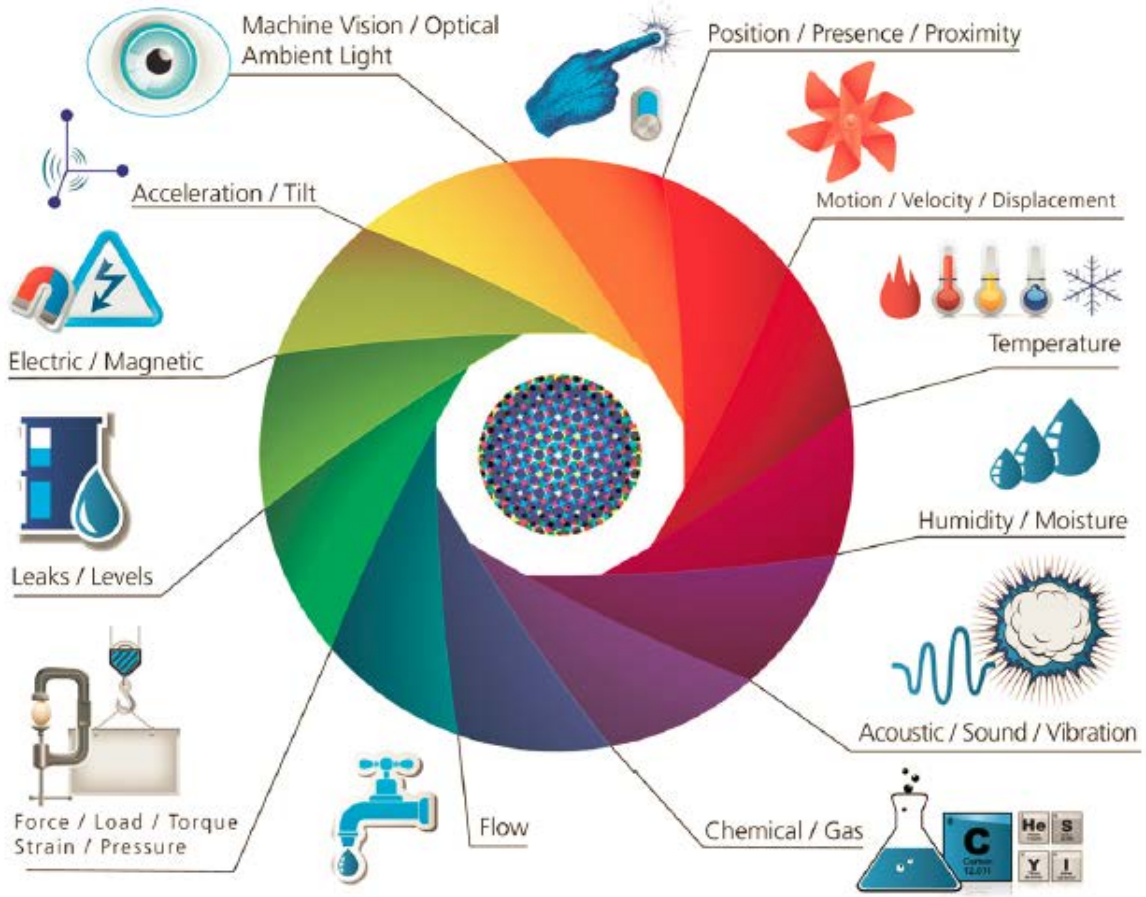
Ecosystems of People, Systems, and Things




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

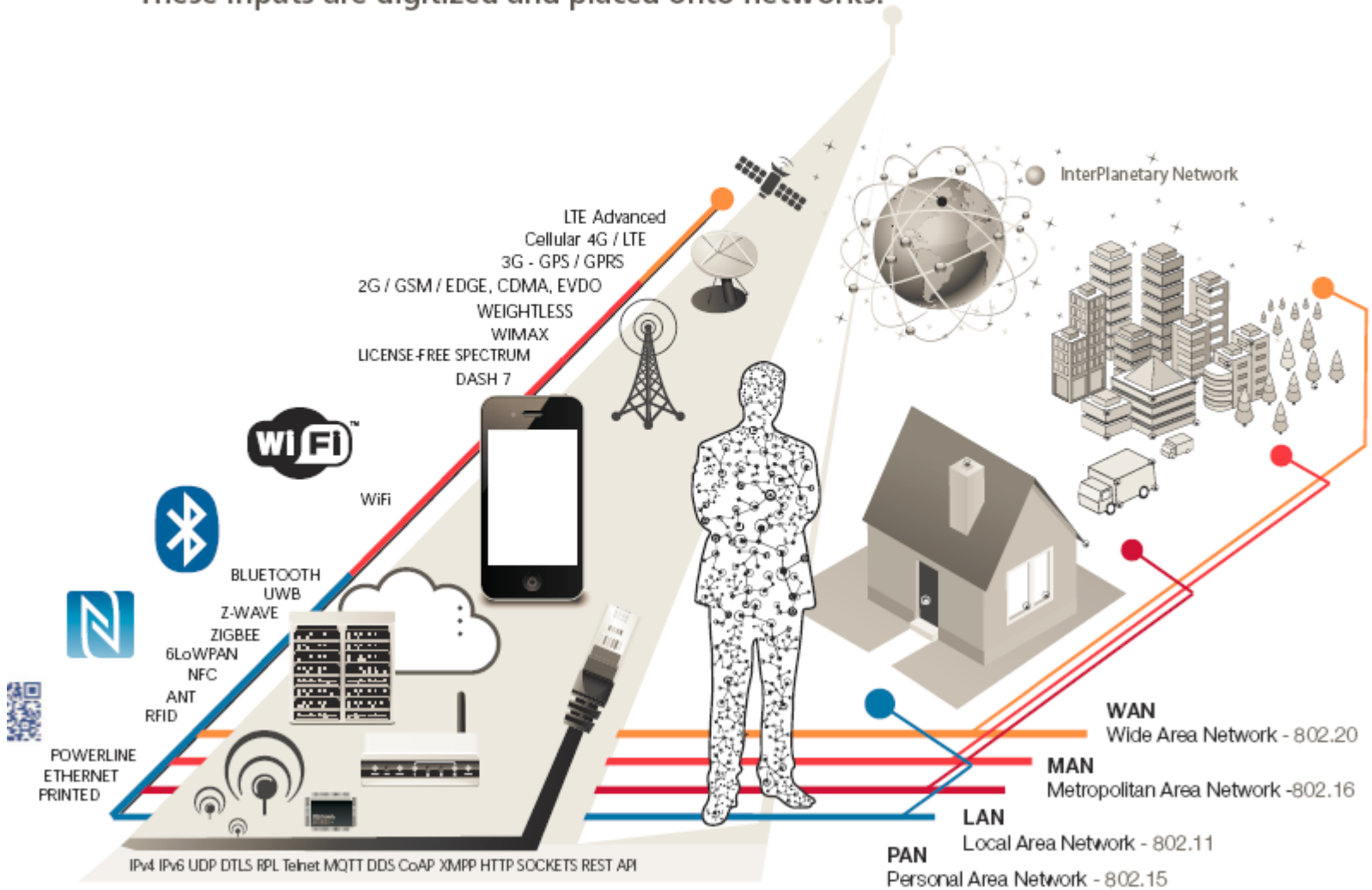
1 SENSORS & *ACTUATORS*

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.



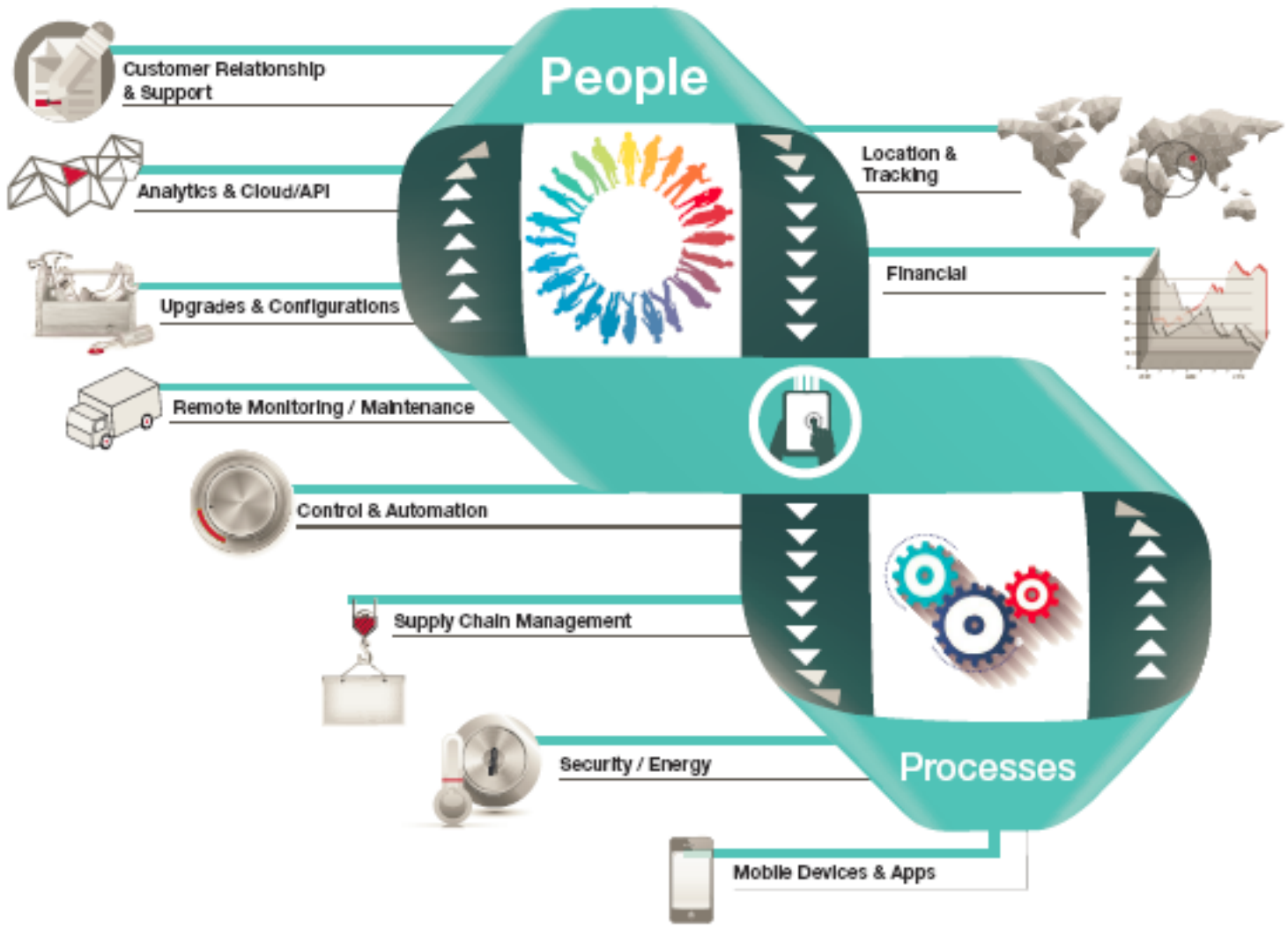
2 CONNECTIVITY

These inputs are digitized and placed onto networks.



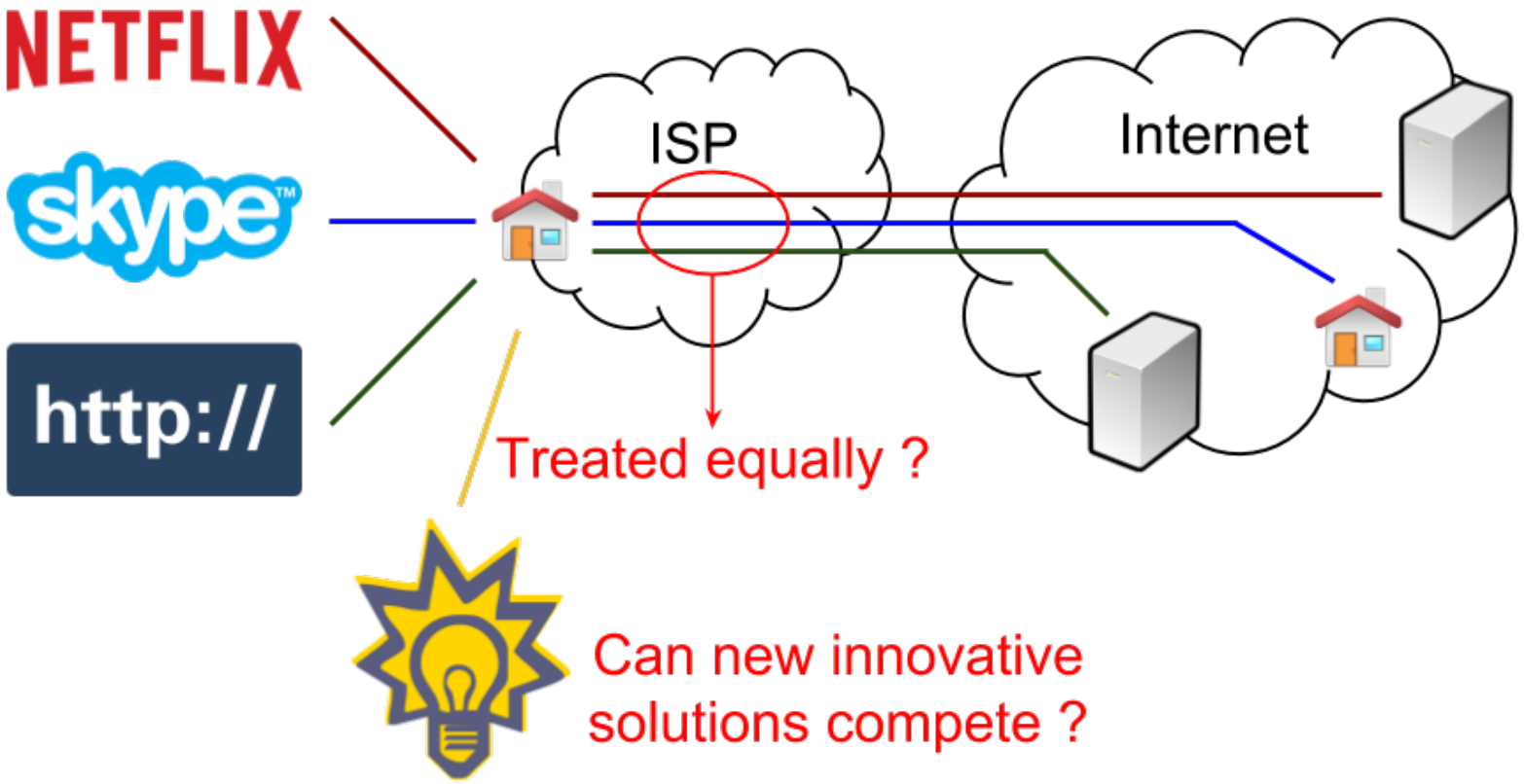
3 PEOPLE & *PROCESSES*

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.



Real cases

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

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

19 Comments

Tweet Share Post

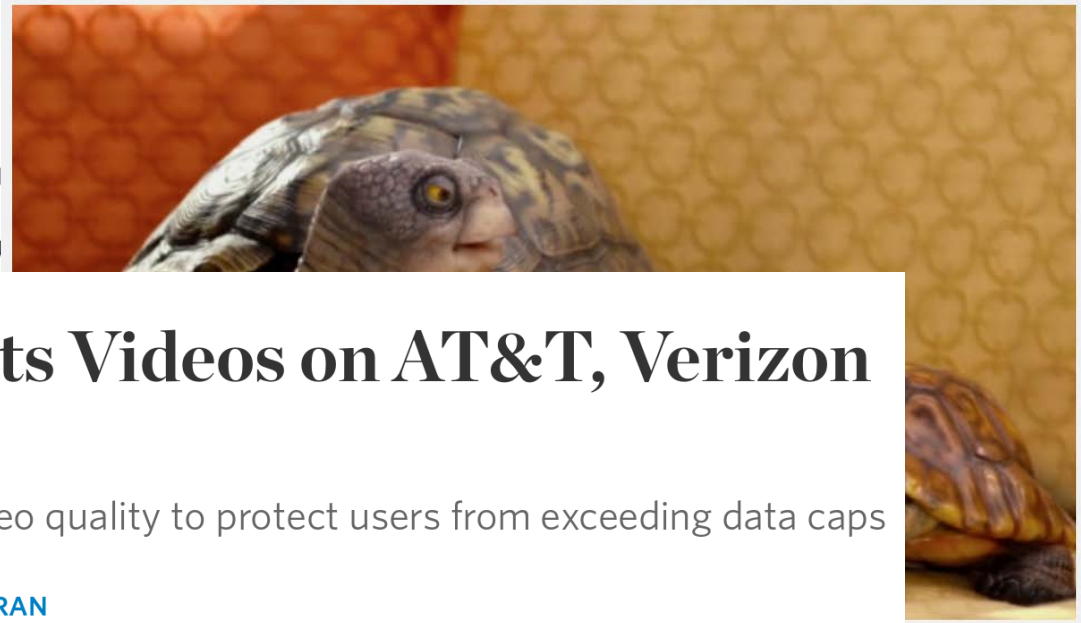
Skype has been one of the top downloaded apps for the iPhone since its release last week, even though Apple (s aapl) bowed to AT&T (s T) in the U.S. to prevent the VoIP program from working on 3G. U Skype users are re on the iPhone, som doubt concerned t whatever they can

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

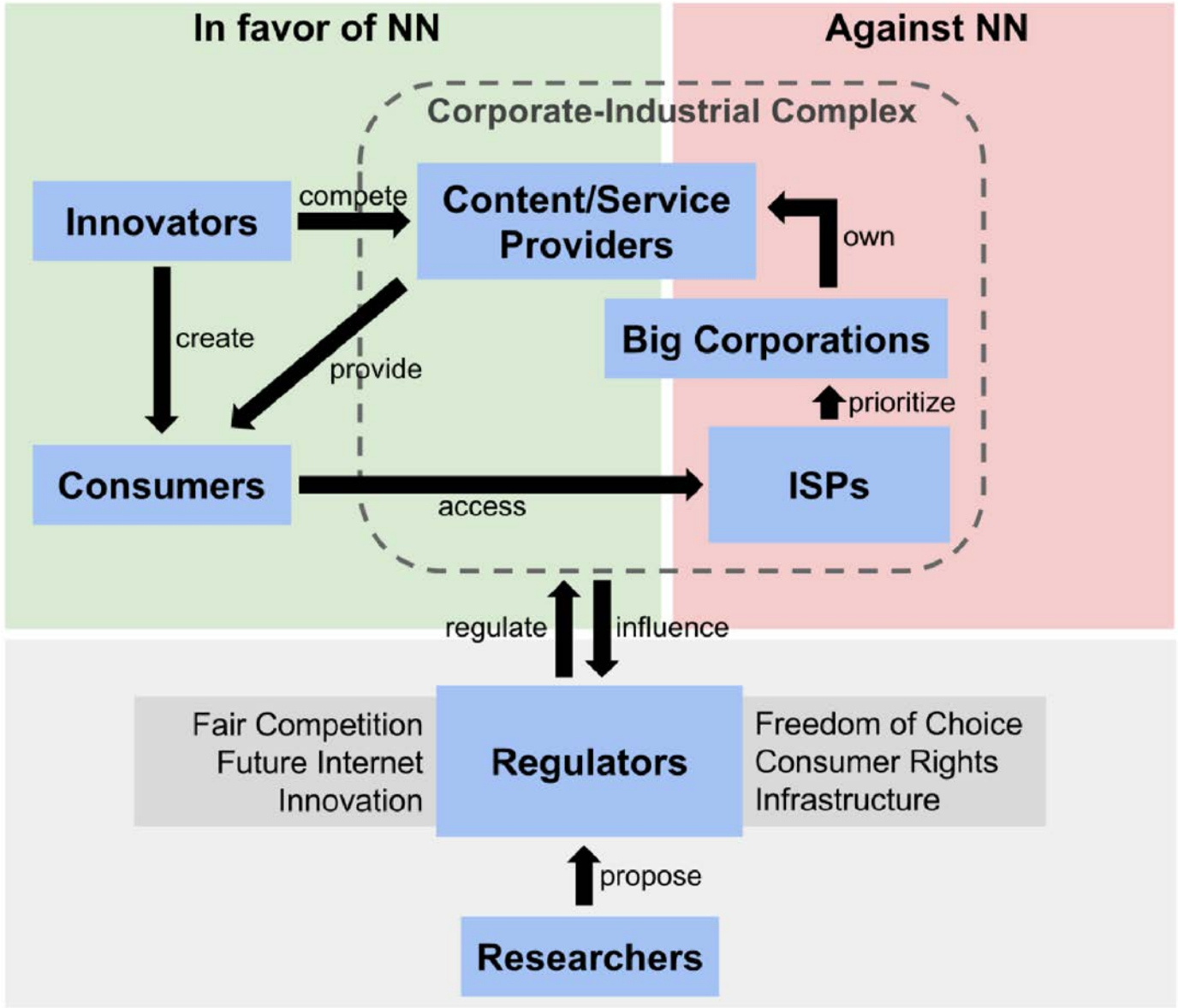
By RYAN KNUTSON and SHALINI RAMACHANDRAN

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

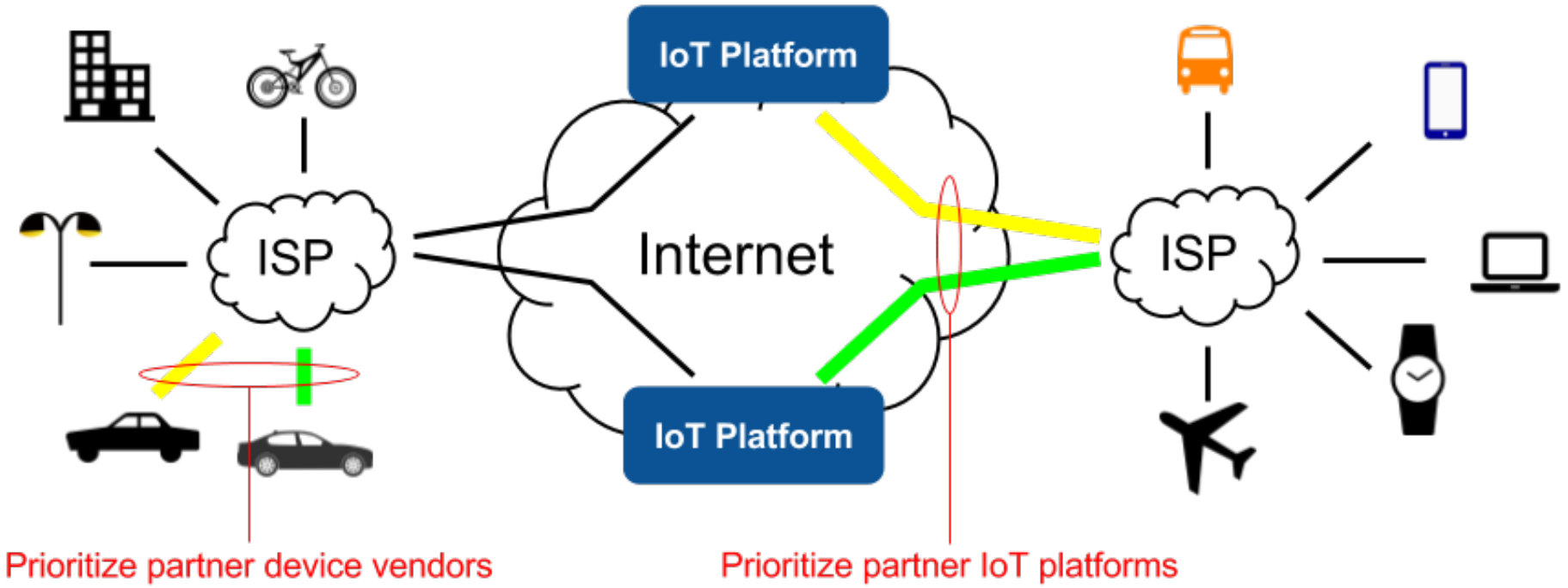
AT&T Inc. and Verizon Communications Inc. were on the defensive last week after accusations swirled they were throttling the quality of Netflix Inc. video on their wireless networks.

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

Understanding the context



What about the Internet of Things (IoT) ?



Smart Evolution – People, Services, and Things

Smart Energy Networks

Smart eGovernments & eAdministrations

Elastic Systems & Processes

Smart Homes

STB

Game Machine

TV

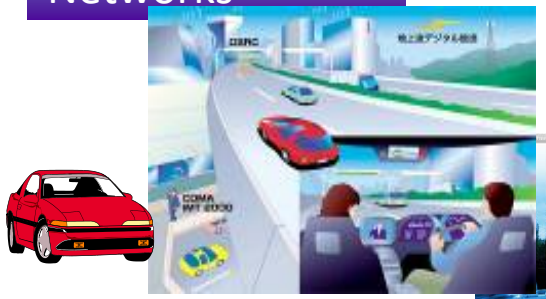
PC

Audio

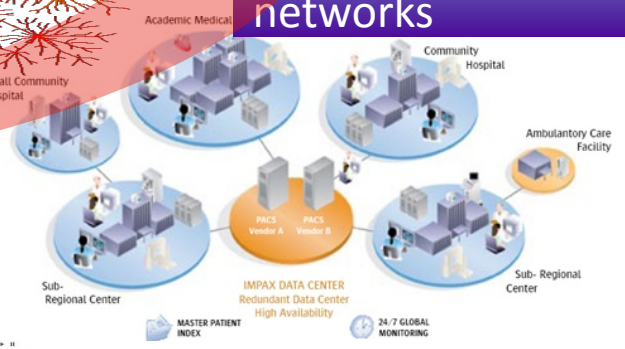
DVD

Telephone

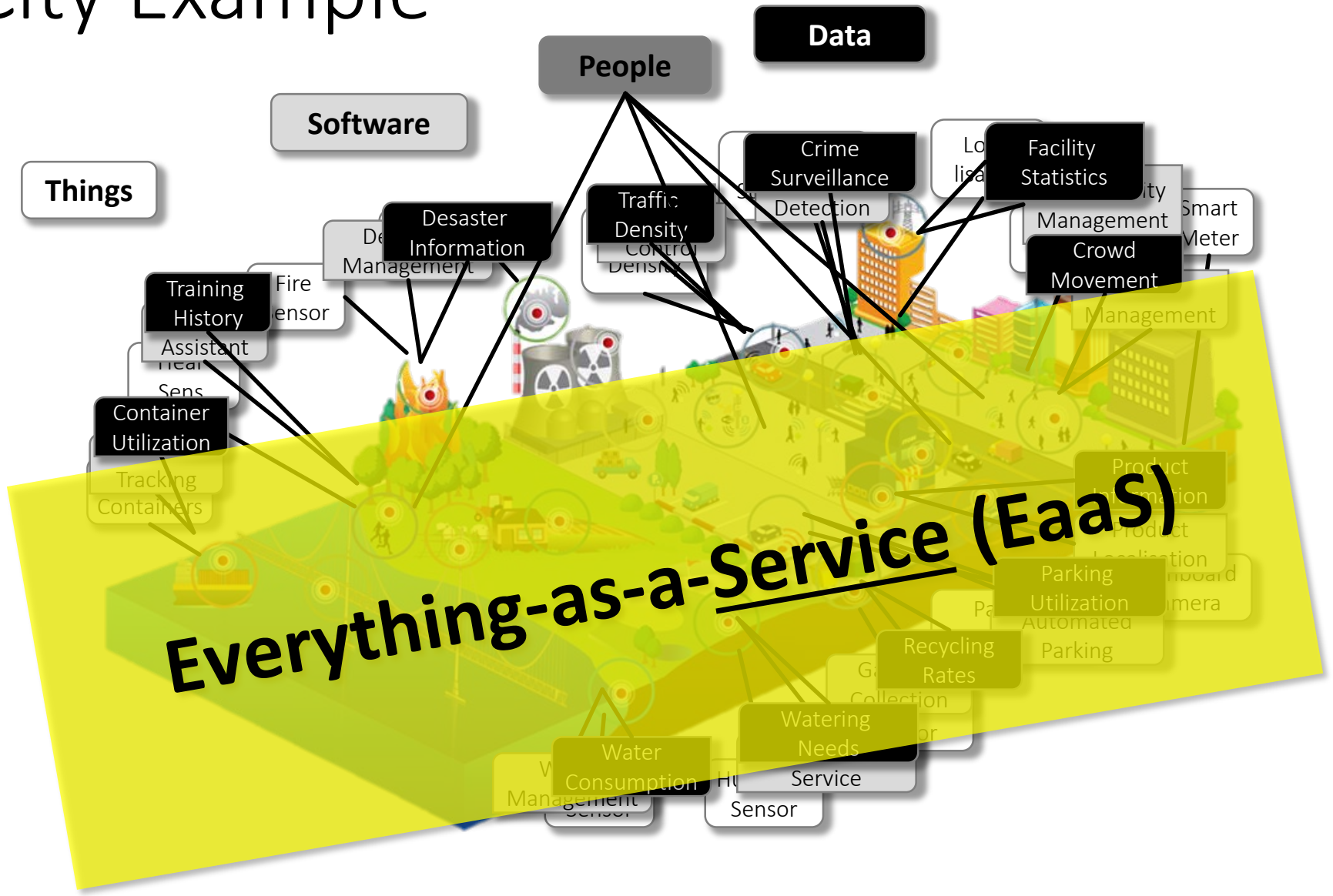
Smart Transport Networks



eHealth & Smart Health networks



Smart City Example



Smart City IoT Ecosystem

Ubiquitous Managed Services Solution Across Business Verticals

CCTV Monitoring

Public Safety

HealthCare

Facilities Control

Industrial process parameters

Parking Control

Waste Management

KIOSK Monitoring

Numerous Forms Of Smart Services…

Managed Services
- Portfolio management
- Event management
- Analytics

Provisioning
- Services
- SIM profile configuration
- Network configuration

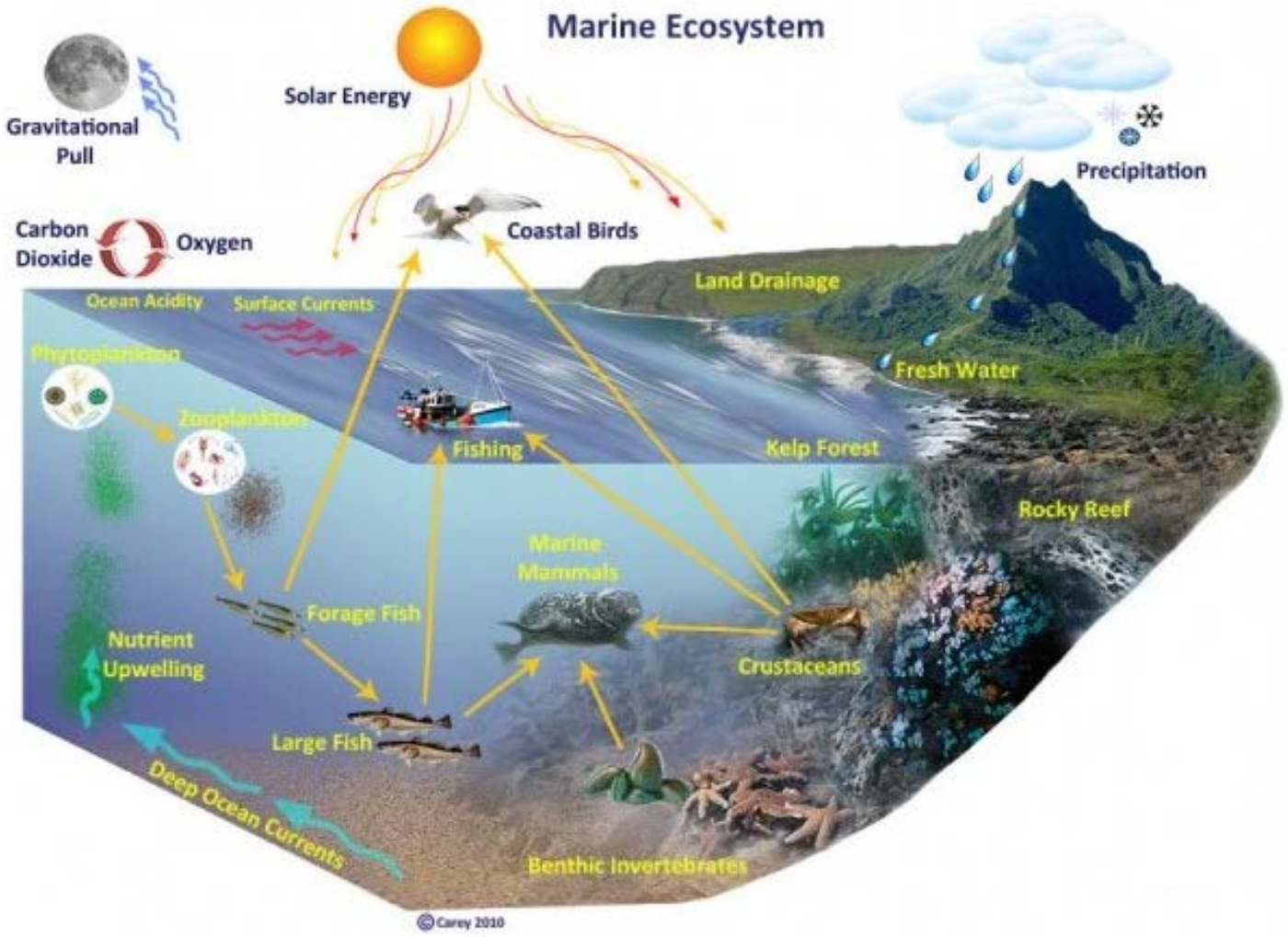
- Activation
- Deactivation
- Privacy
- Security

Transaction Mgmt.
- Visibility
- Billing
- Reporting

IoT and Cloud Computing enable smart services ecosystem and collaboration opportunities

Some 200 billion devices and sensors exist for M2M applications

Think Ecosystems: People, Systems, and Things

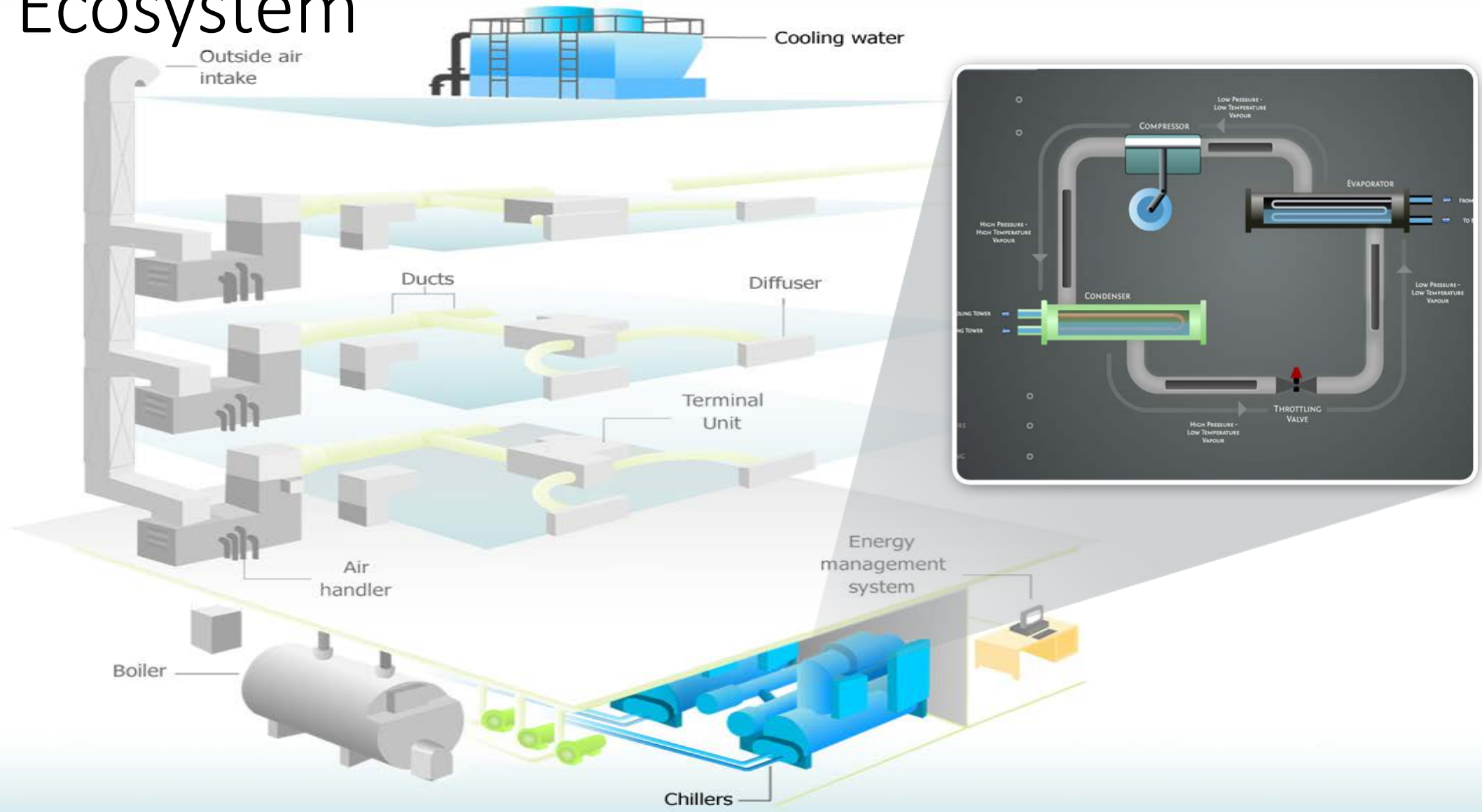


Diverse users with complex networked dependencies and intrinsic adaptive behavior – has:

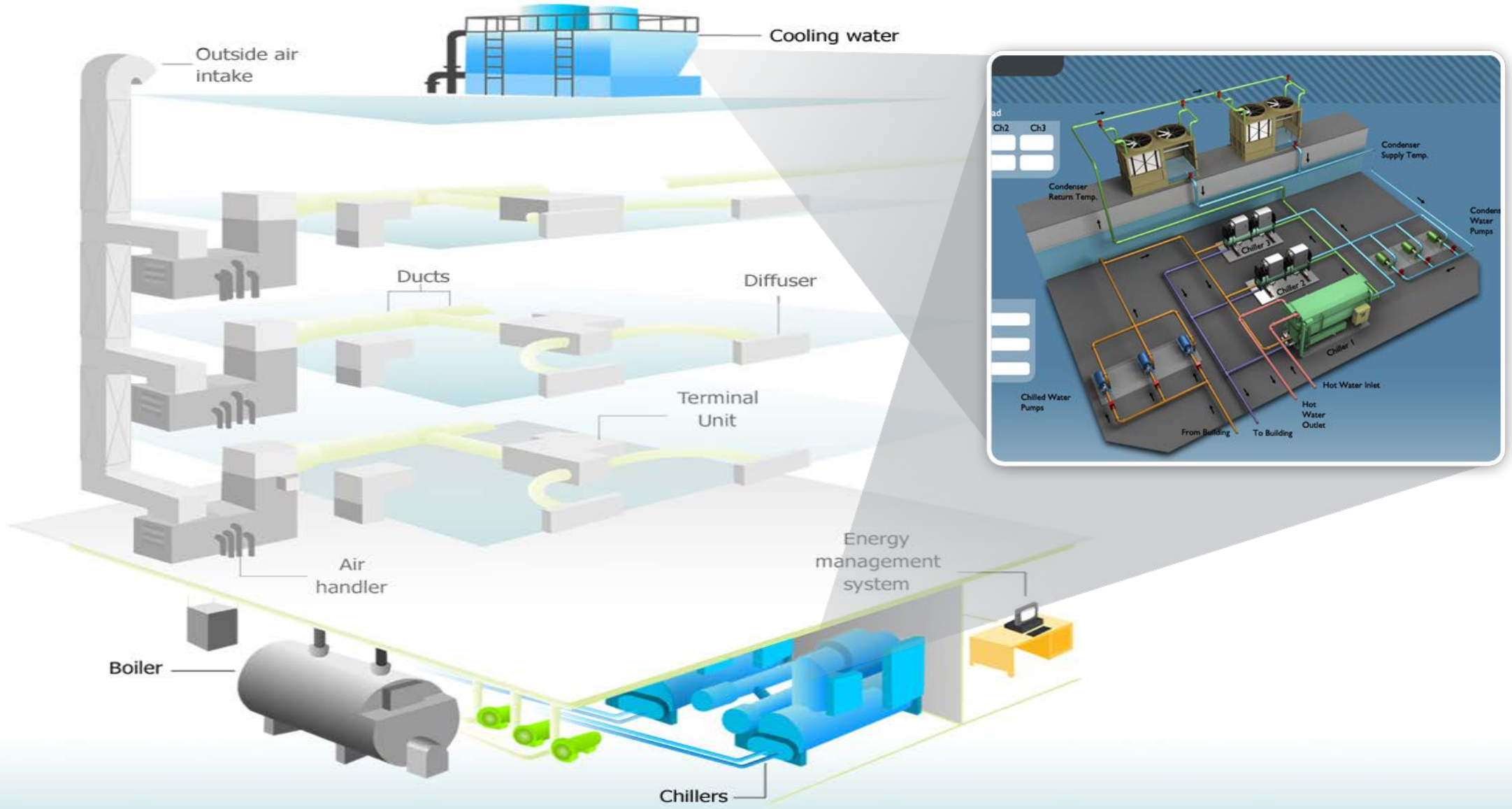
1. **Robustness & Resilience mechanisms**: achieving stability in the presence of disruption

2. **Measures of health**: diversity, population trends, other key indicators

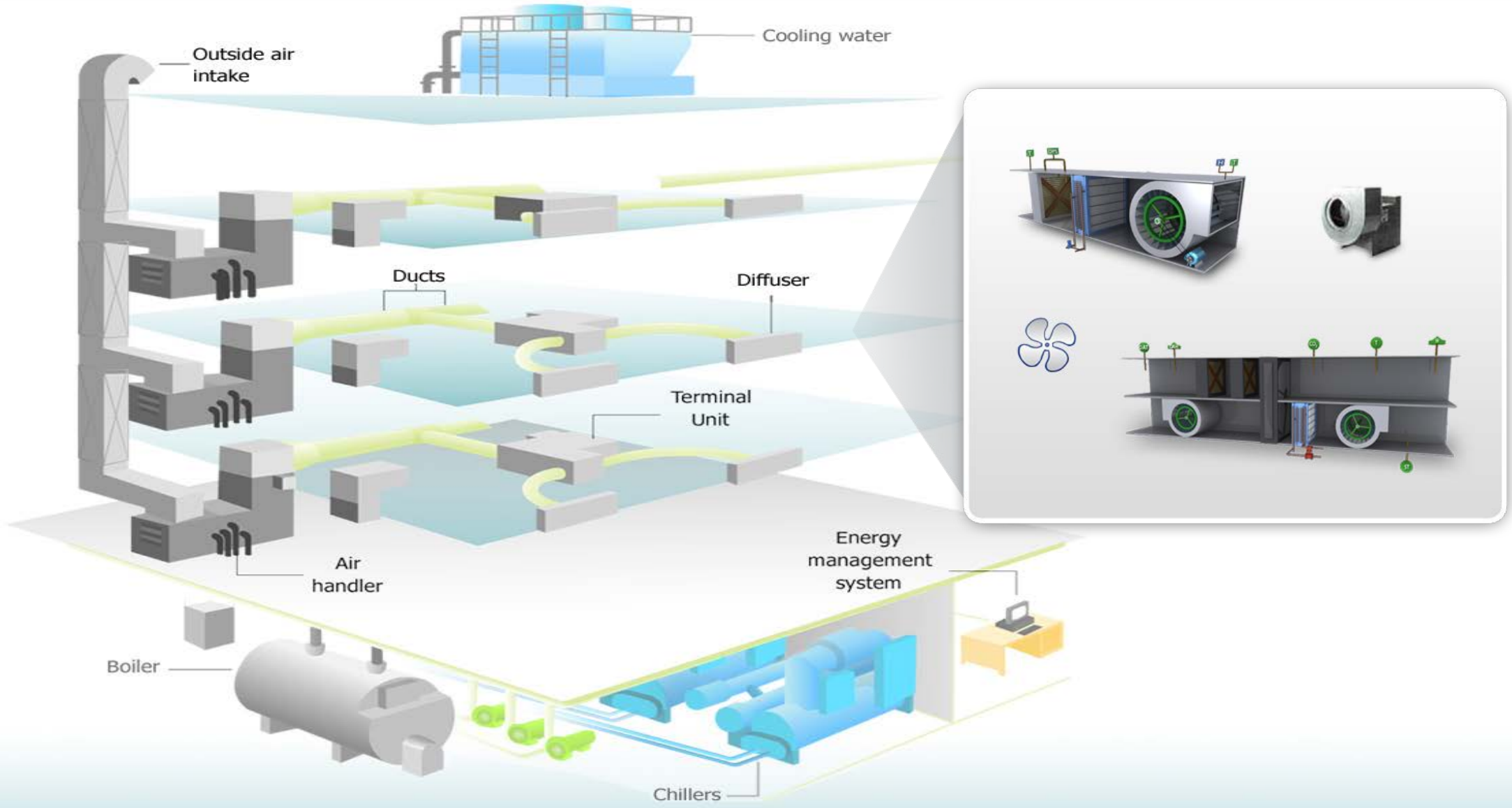
HVAC (Heating, Ventilation, Air Conditioning) Ecosystem



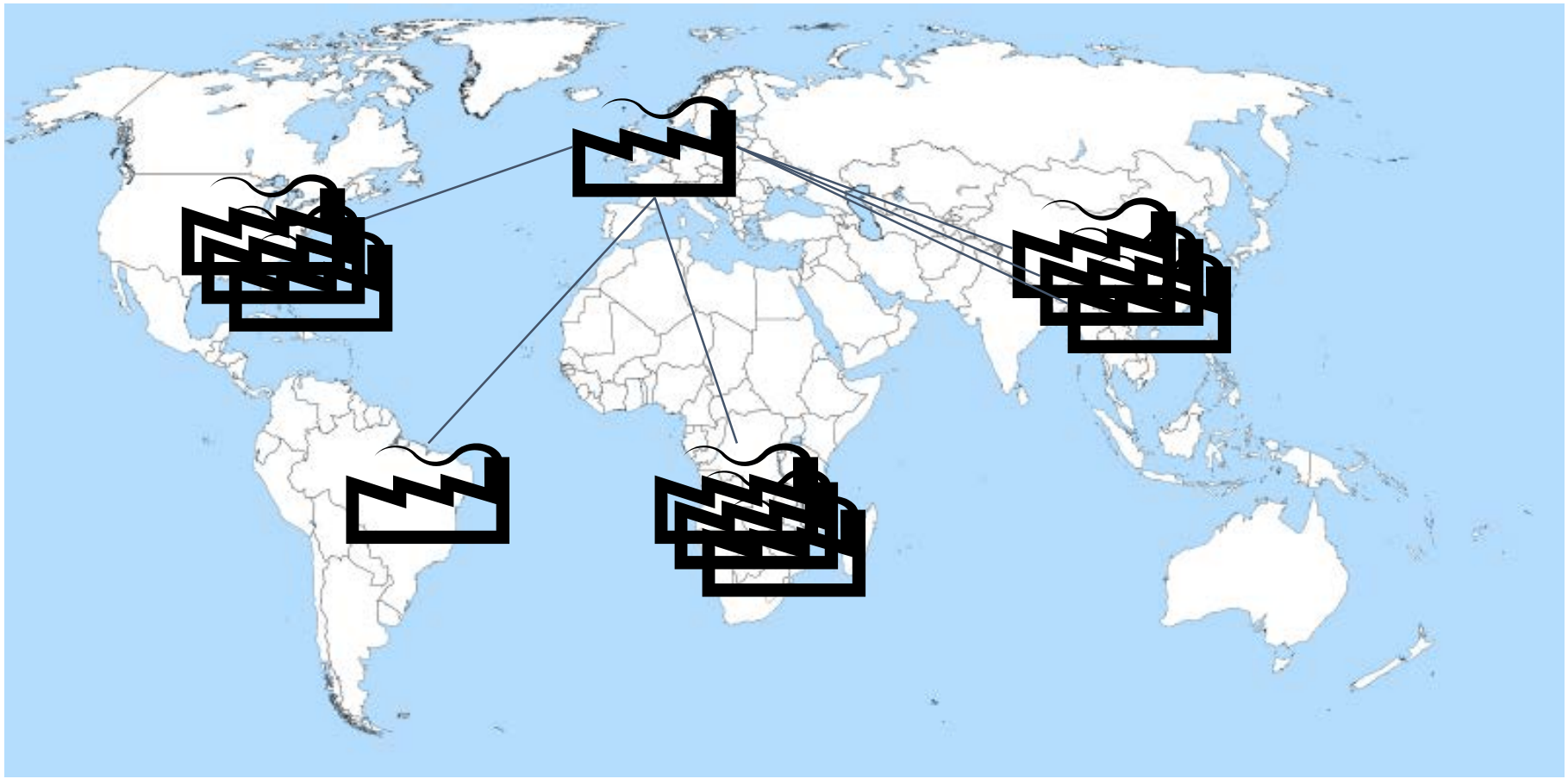
Water Ecosystem



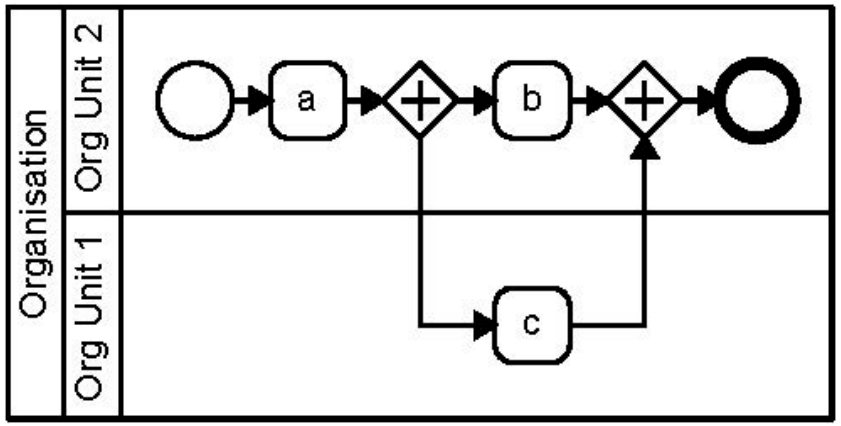
Air Ecosystem



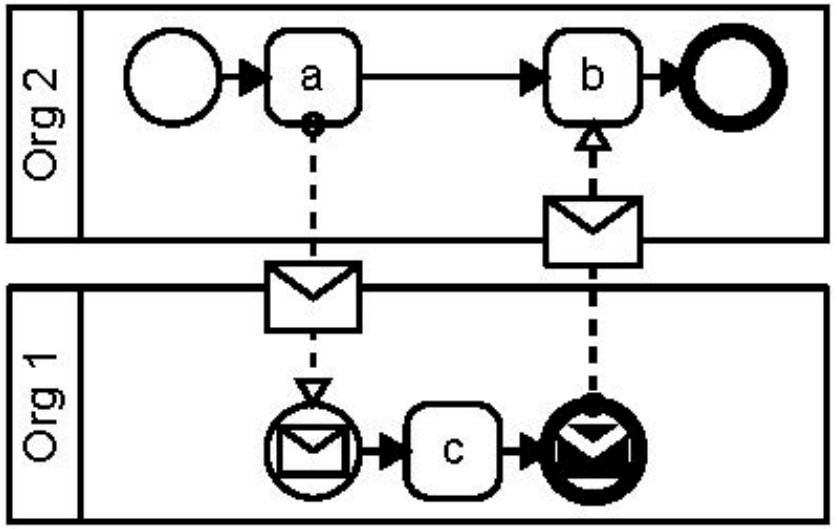
Towards “Industry 4.0”



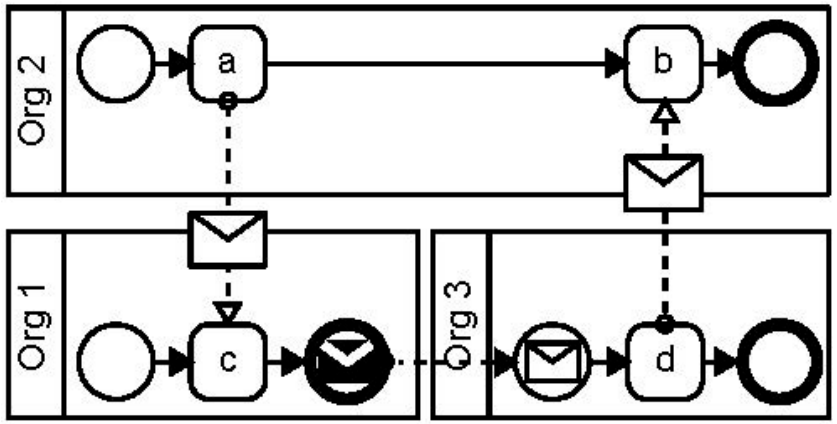
From Centralized to Distributed Control



Suborganizations, Centralized Control

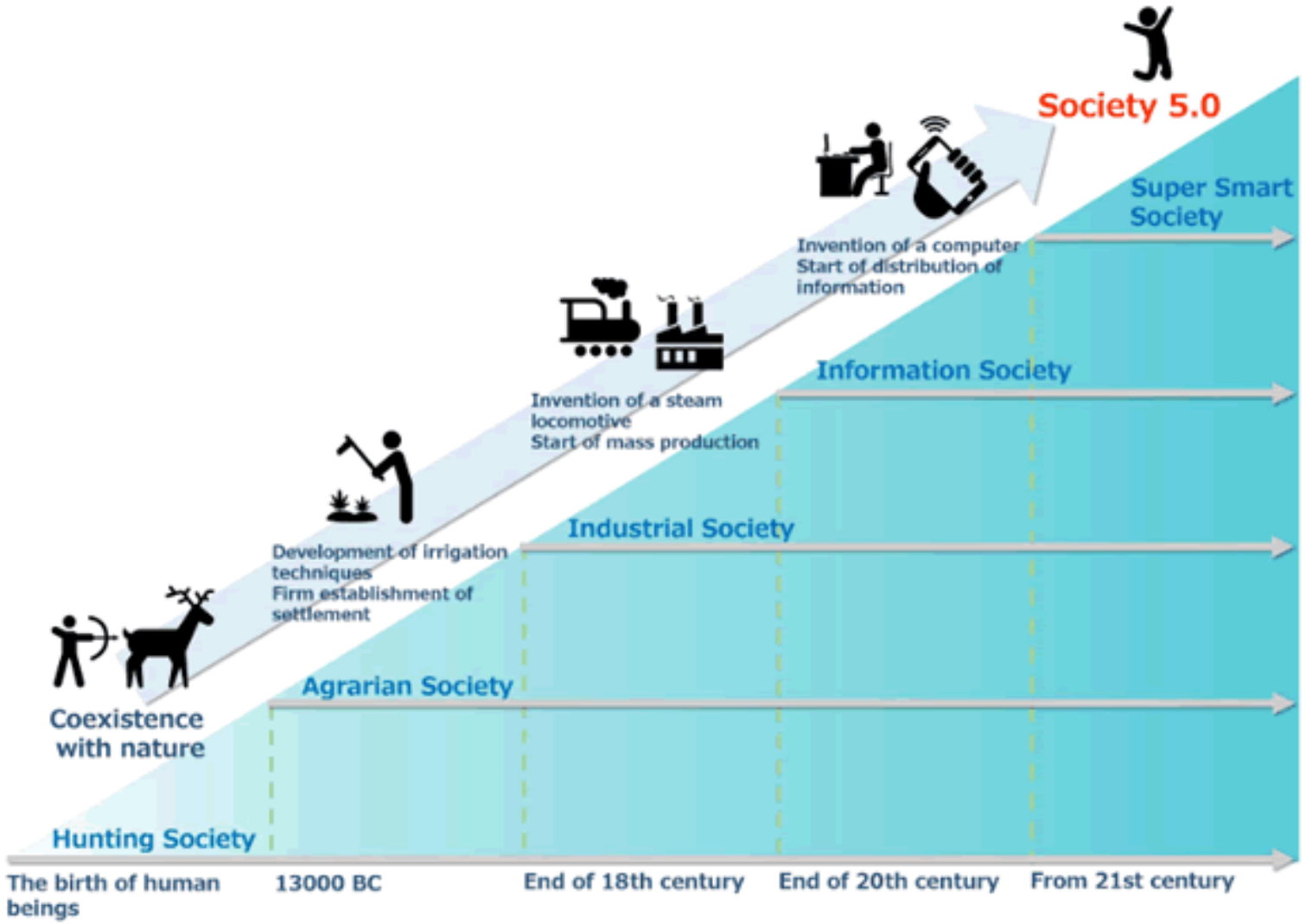


Multiple Organizations, Centralized Control



Multiple Organizations, Distributed Control

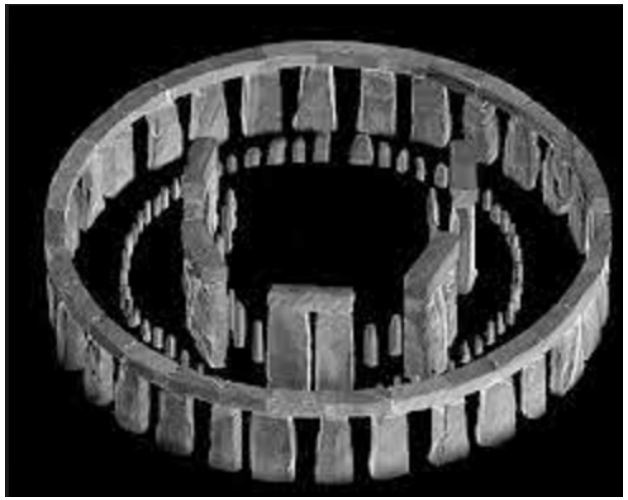
Towards “Society 5.0“ in Japan



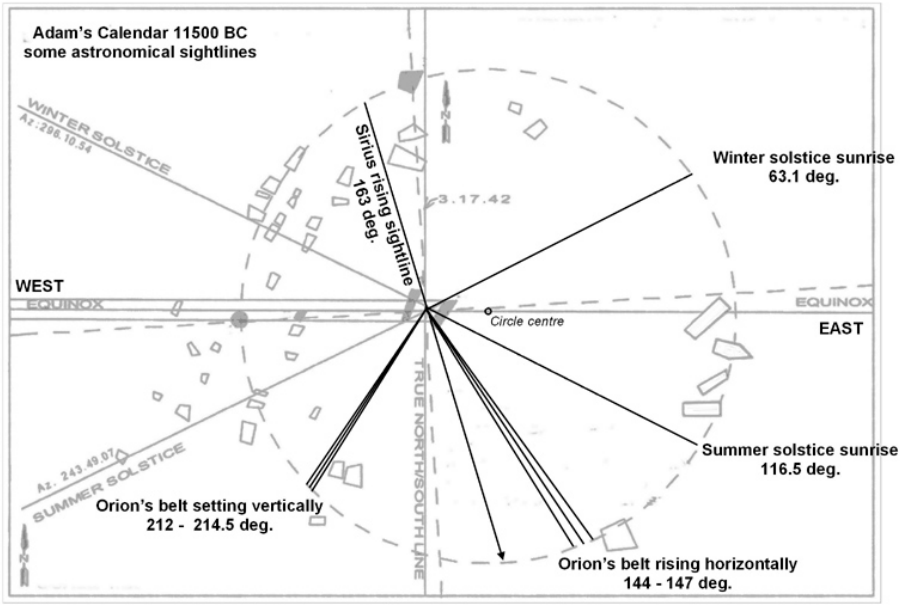
Toward realization of the new economy and society, Keidanren (Japan Business Federation), April 2016

Linear History? Ancient Computers

Adam‘s Calender, Michael Tellinger



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



Assumptions, Models, and Abstractions

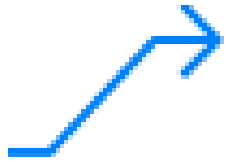
- **Co-evolution of Science & Technologies**
- Smart Cities as models of ecosystems: -> People, Things, and Systems
- Models as abstractions are useful (Platonic Forms)
- We lack a model and fitting paradigms for such an ecosystem
- From automation to creativity support
- Consciousness and creativity 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/societal needs
- Ecosystems = Architecture, Structure + Dynamics
- New Paradigms: (1) Elastic - & (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

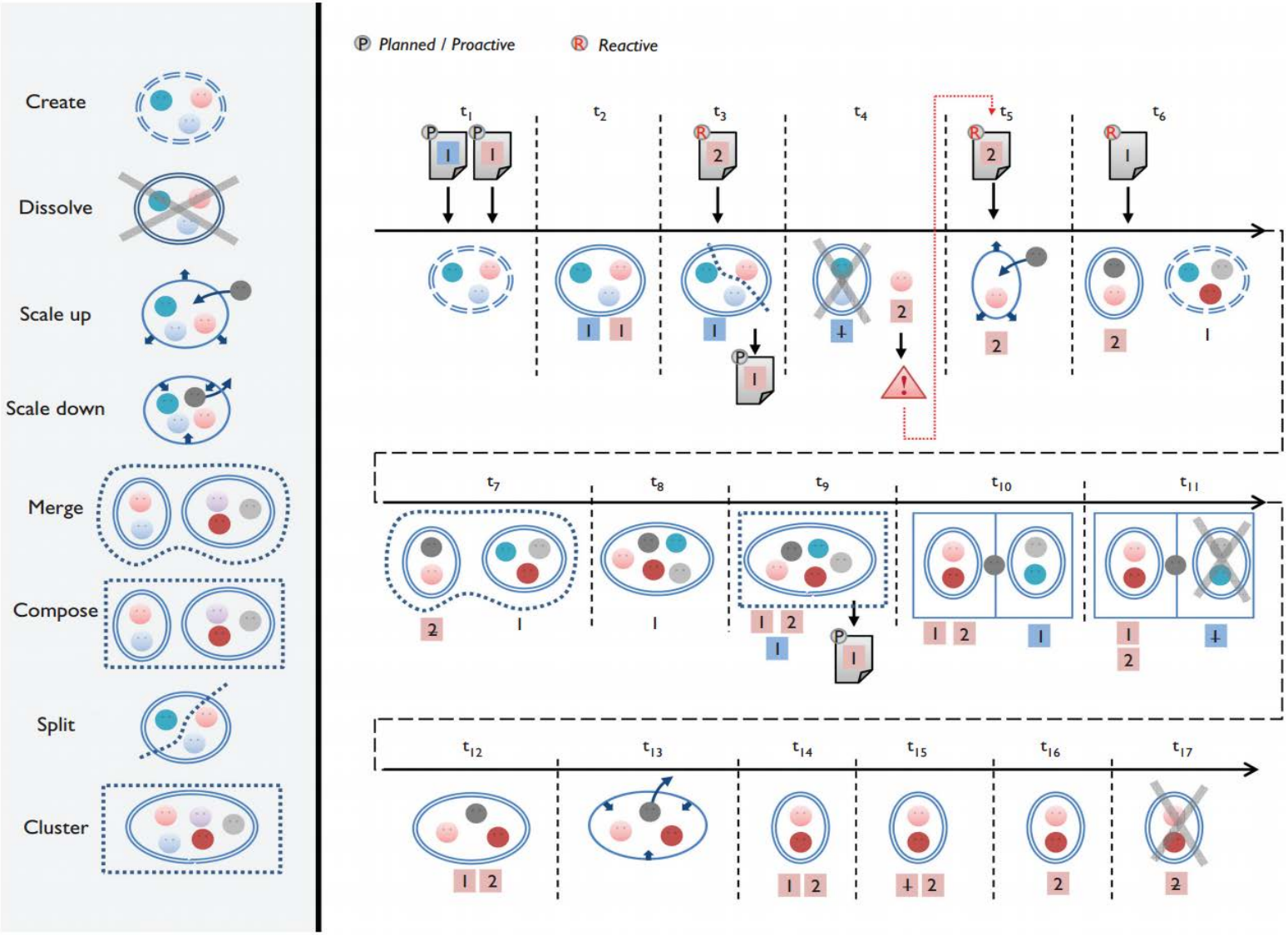
Non-functional parameters e.g., performance, quality of data, service availability, human trust

Costs & Benefit elasticity

rewards, incentives

Dustdar S., Guo Y., Satzger B., Truong H. (2011) Principles of Elastic Processes, *IEEE Internet Computing*, Volume 15 (2011), Issue 5; pp. 66 - 71.

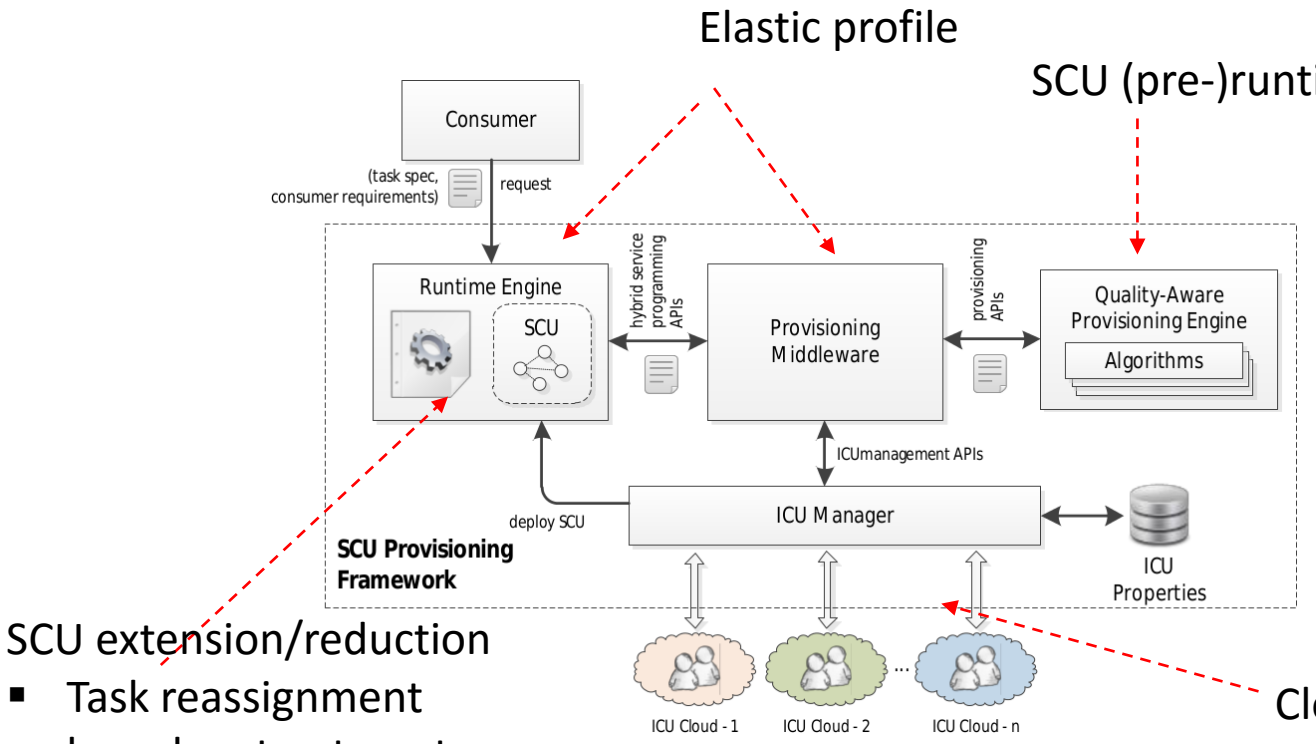
Paradigm 2: Social Compute Units (SCUs)



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)



Algorithms

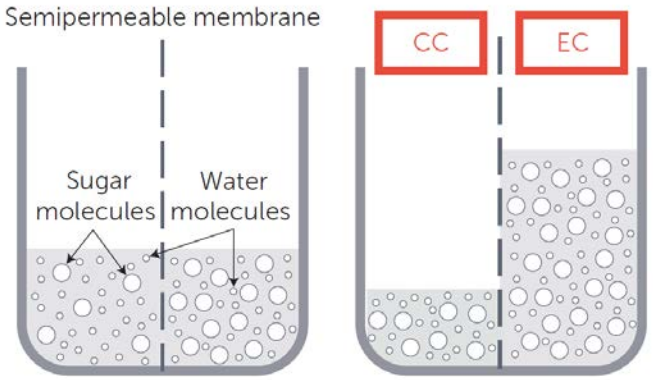
- Ant Colony Optimization variants
- FCFS
- Greedy

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

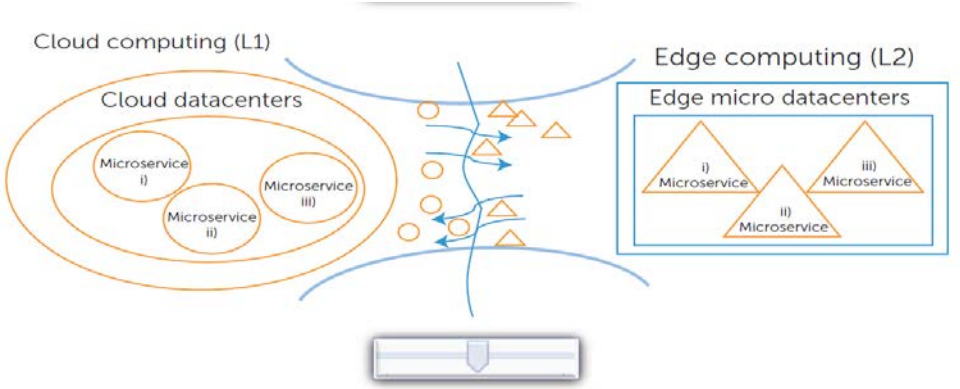
Muhammad Z.C. Candra, Hong-Linh Truong, and Schahram Dustdar, **Provisioning Quality-aware Social Compute Units in the Cloud, ICSOC 2013.**

Paradigm 3: Osmotic Computing

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

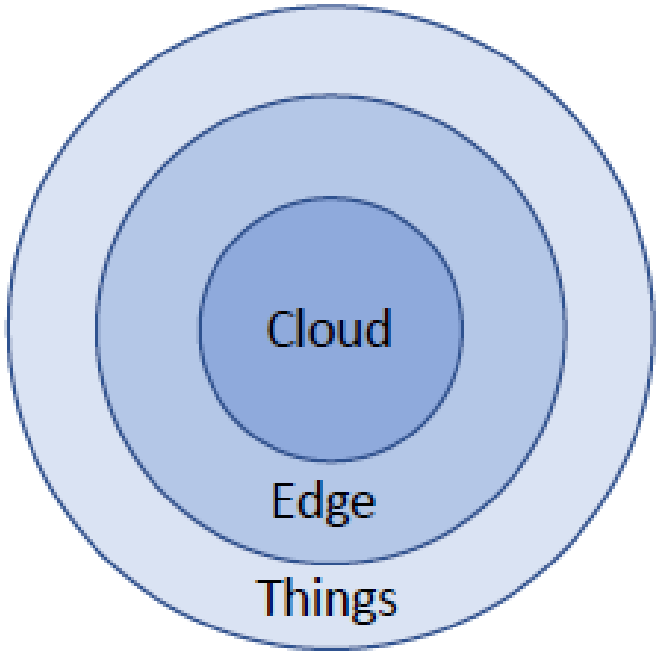


- Dynamic management of (micro)services across cloud and edge datacenters
  - deployment, networking, and security, …
  - providing reliable IoT support with specified levels of QoS.

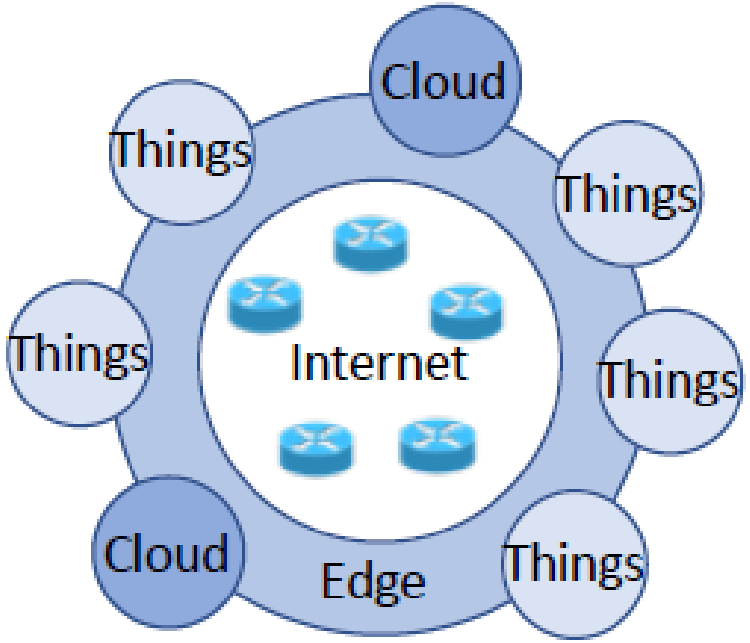


Villari M., Fazio M., Dustdar S., Rana O., Ranjan R. (2016). Osmotic Computing: A New Paradigm for Edge/Cloud Integration. *IEEE Cloud Computing*, Volume 3, Issue 6, pp. 76-83

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

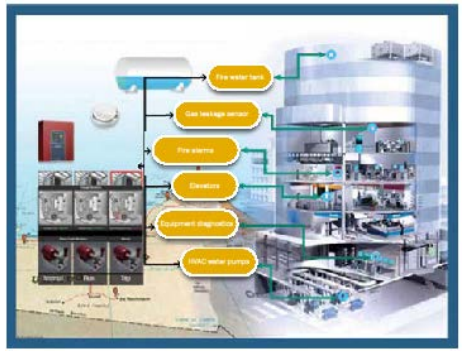
Therefore

- 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

Motivating Case Studies

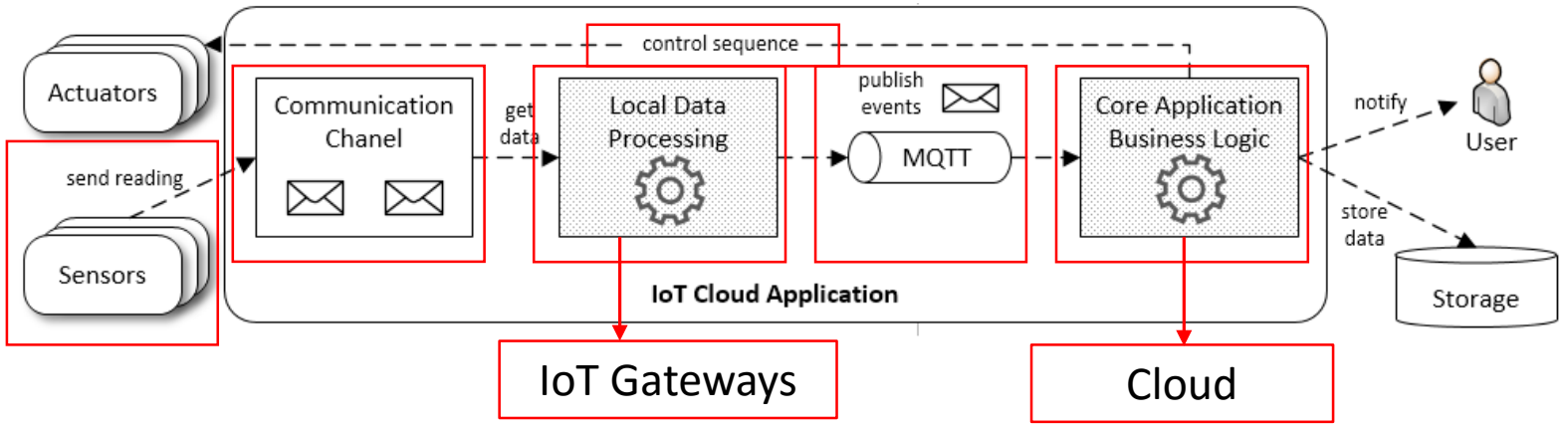
Building Management System

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



Fleet Management System

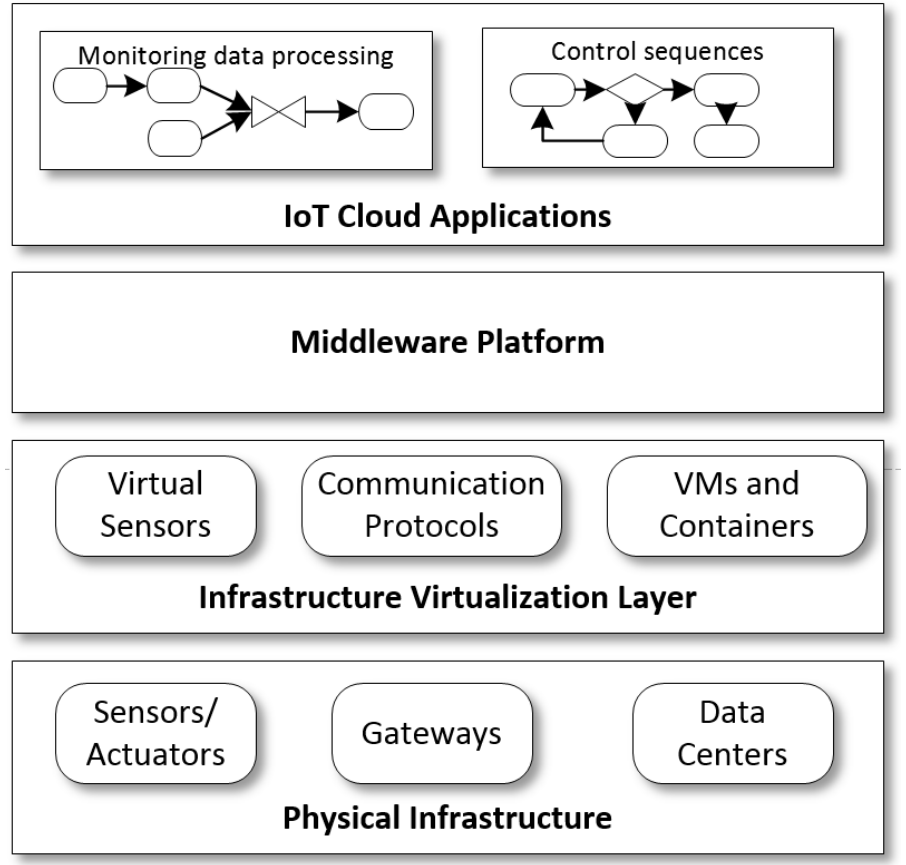
- Manages fleets of electric vehicles world-wide (e.g., on golf courses)



Motivation

- Lack of systematic support and tools for **developing** and **operating** IoT Cloud systems

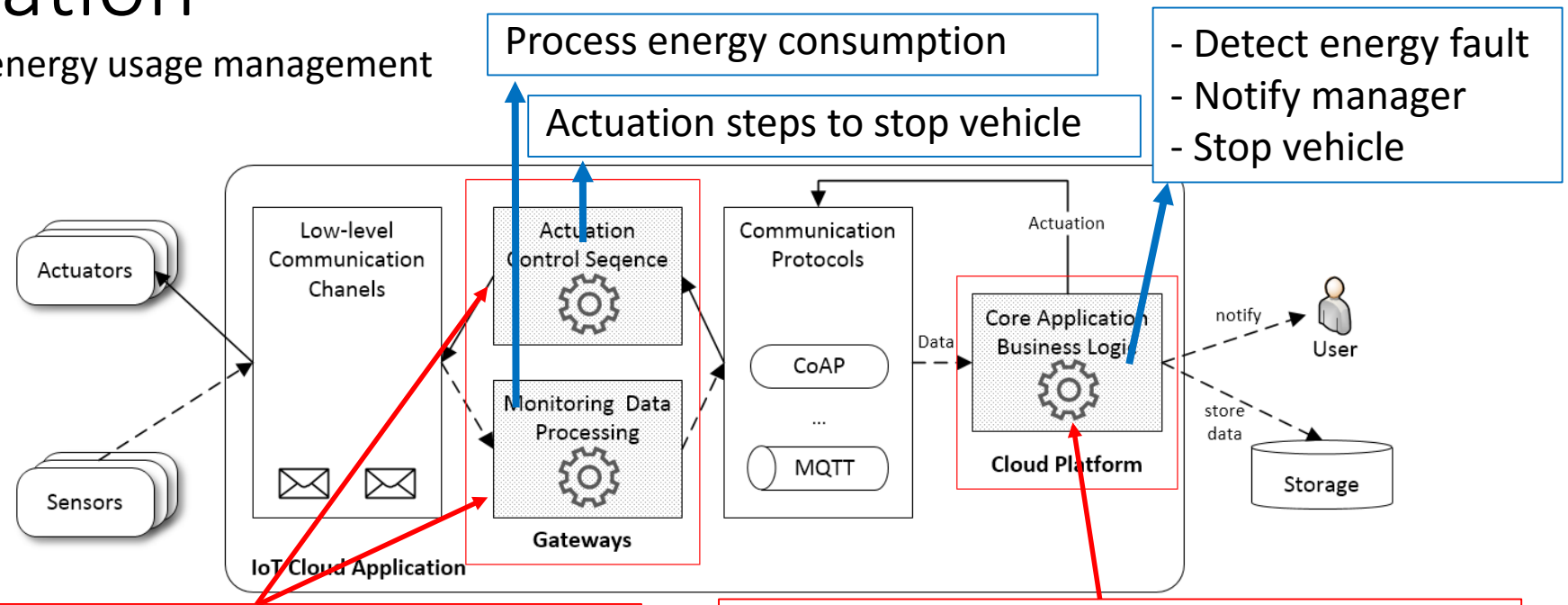
- Today IoT Cloud systems are vertically closed and tightly coupled
  - Hard to develop and maintain applications
  - Difficult to operate and reuse existing infrastructure



Programming Model for IoT Cloud Applications

C1: Motivation

Fleet energy usage management



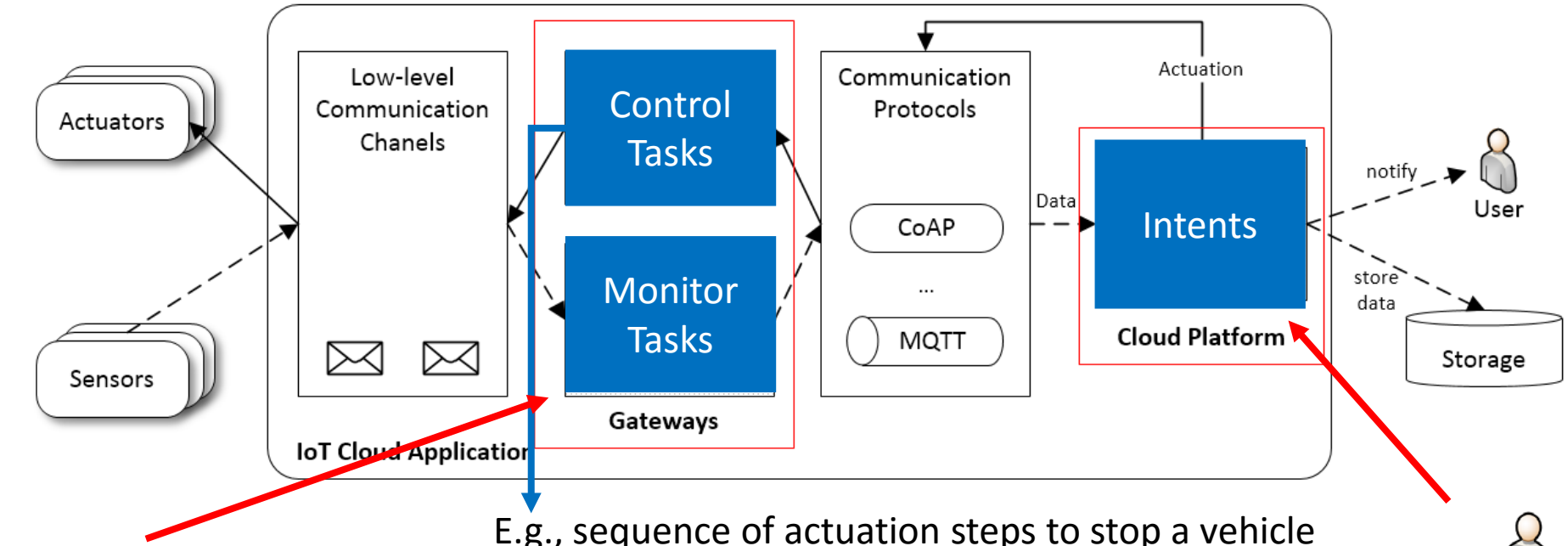
Requirements:

- *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

C1: Approach



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

Domain Experts

Task - Encapsulates domain-dependent controls or analytics

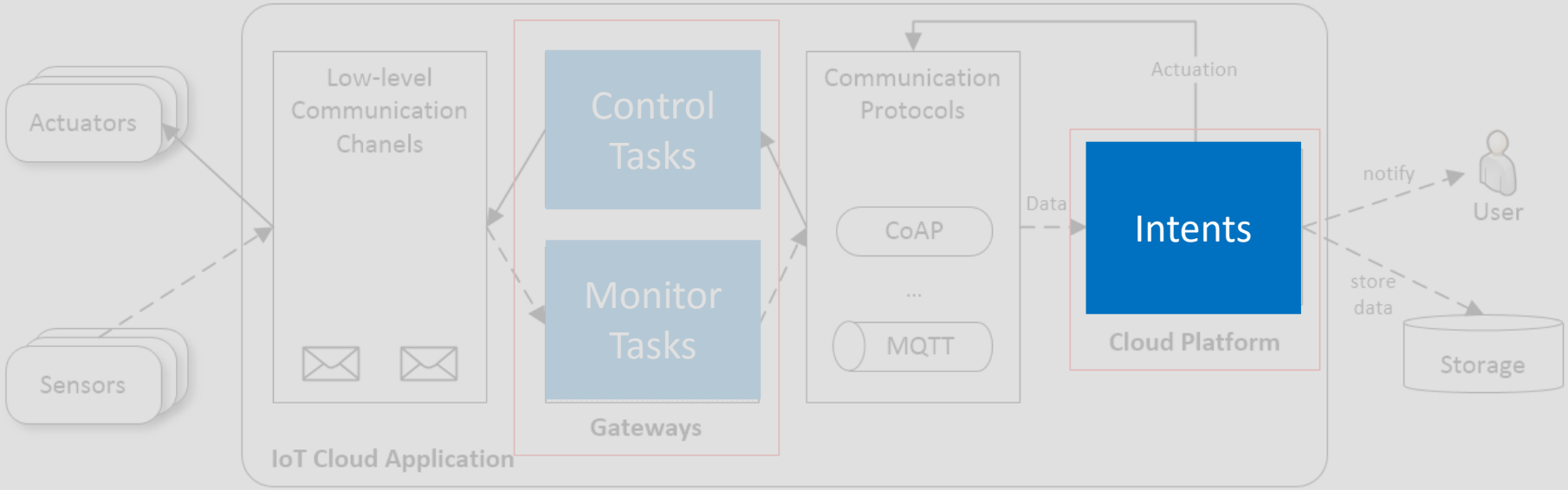
- Packaged into domain-specific libraries (e.g., vehicles management)

Intent - High-level representation of Tasks on Cloud platforms

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

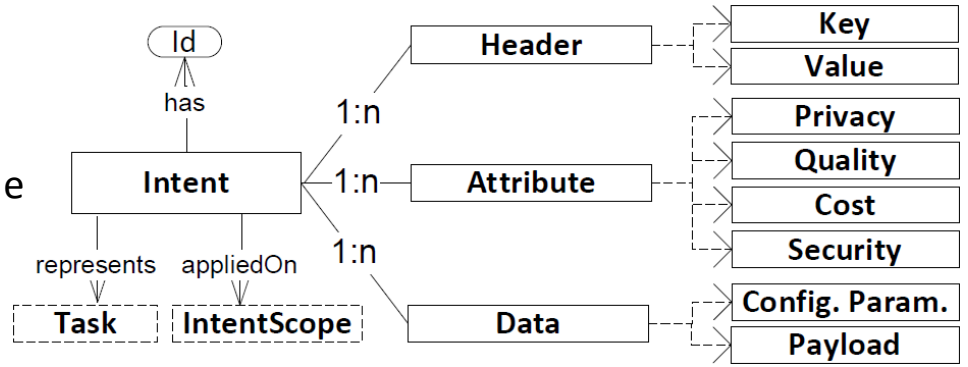
Devlopers

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
IoT Cloud Application
Communication
Protocols
CoAP
...
MQTT
Data
Actuation
Intents
Cloud Platform
notify
User
store
data
Storage

C1: 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

Provisioning solutions for Smart Cities

“abstractions, concepts and processes“

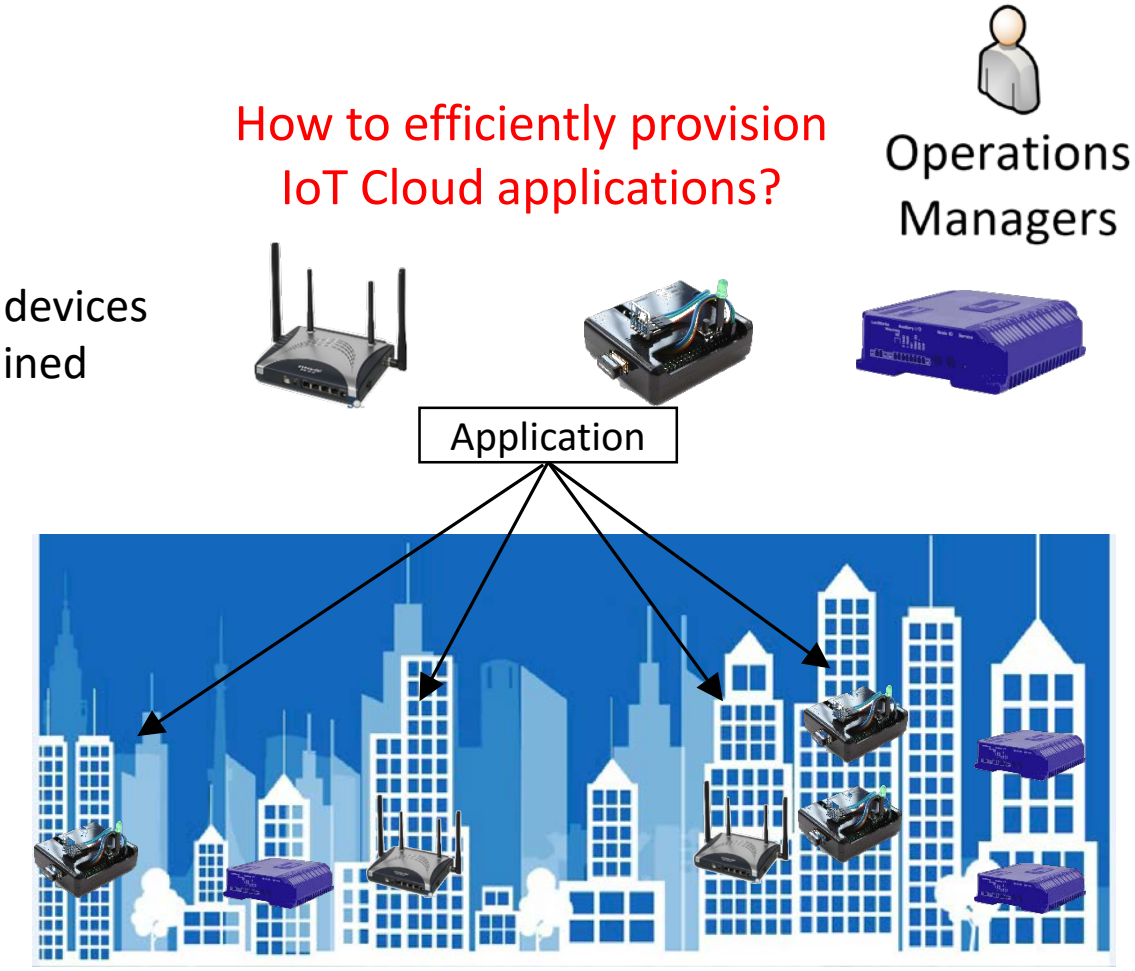
Motivation

Consider provisioning a simple application for monitoring environmental conditions in Smart City buildings

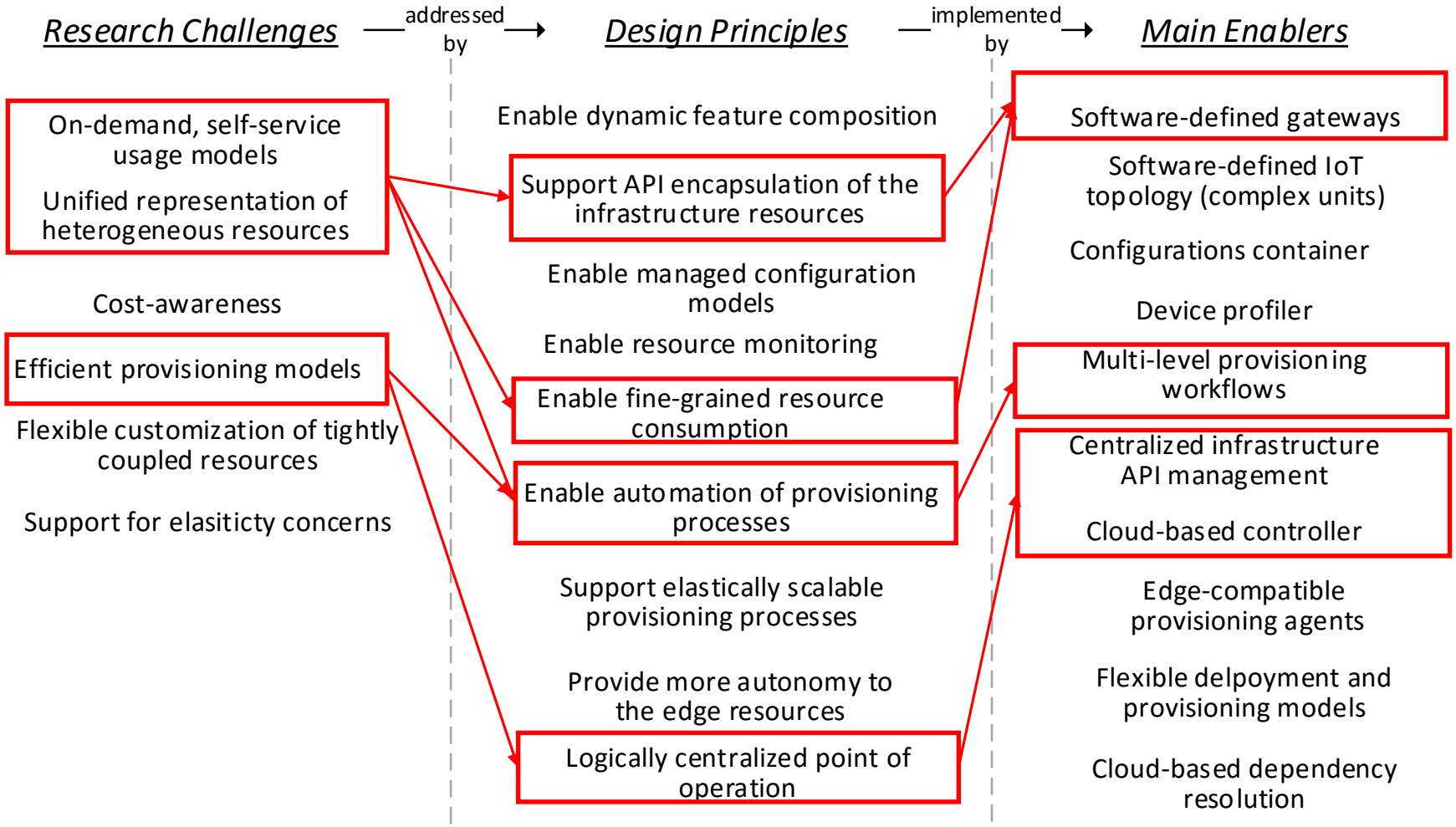
Current provisioning solutions:

- Require on site presence
- Require manual interaction with devices
- Not suitable for resource constrained devices

How to efficiently provision IoT Cloud applications?



Research Challenges Overview



Approach: Core principles

- *From physically isolated, rigid Edge/IoT infrastructure to virtualized, elastic IoT Cloud, by utilizing software-defined principles.*

- *From task-specific solutions to fully-fledged ecosystem and management processes, based on DevOps best practices.*

Approach: Design Principles vs. Main Enablers

Uniform representation of IoT infrastructure

API encapsulation of infrastructure resources

Logically centralized operation

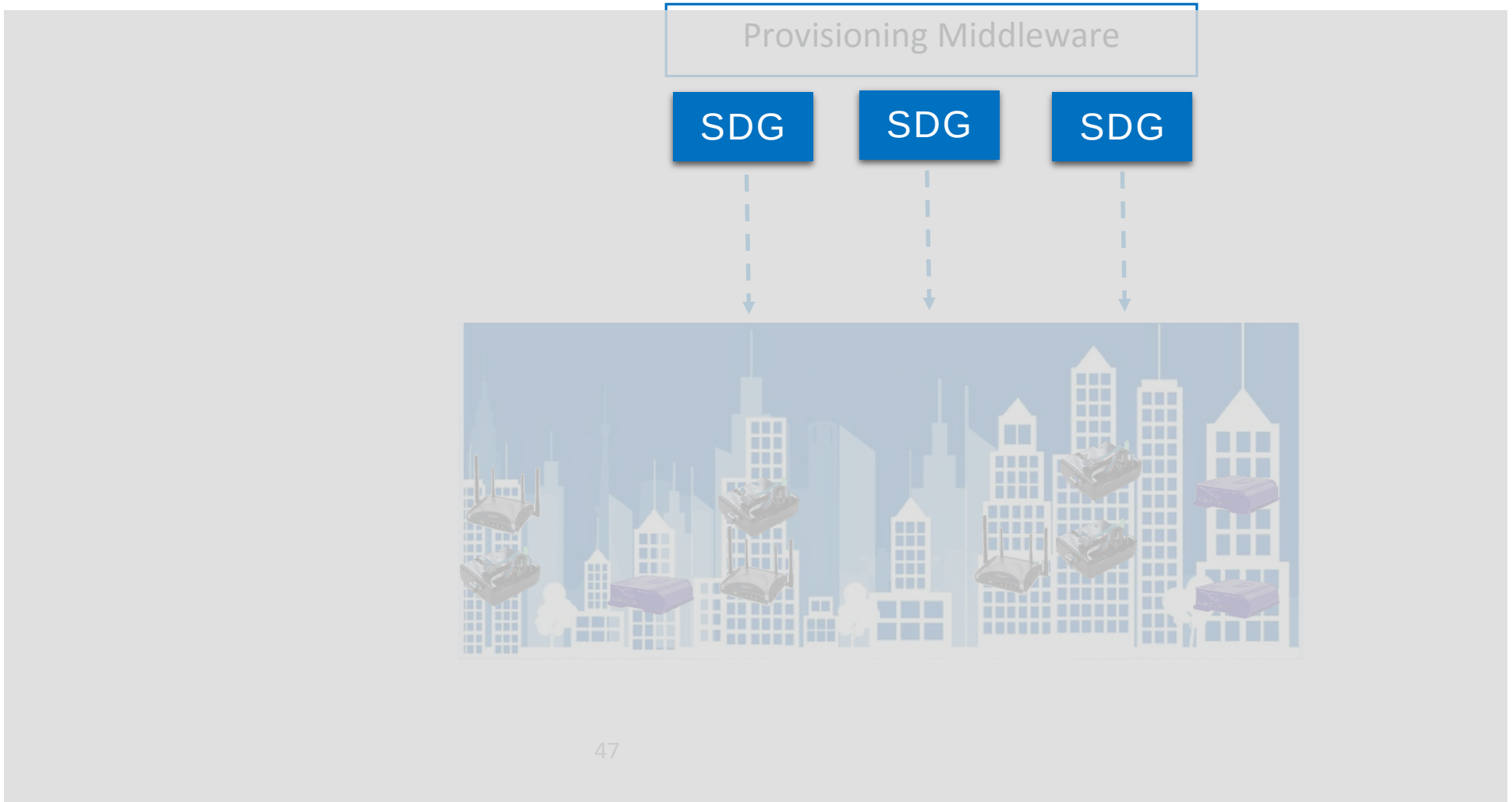
Automation of provisioning processes

Software-Defined Gateways

Provisioning Middleware

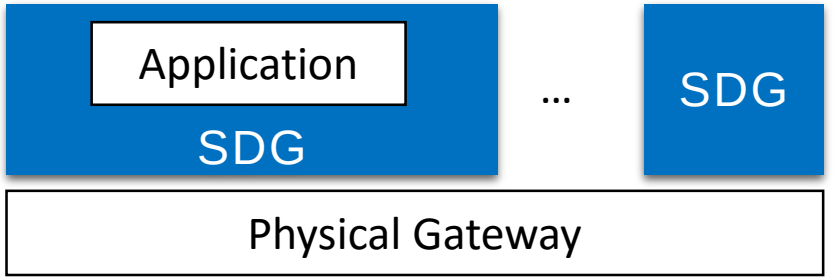
SDG SDG SDG

C2: Approach



Software-Defined Gateways – Overview

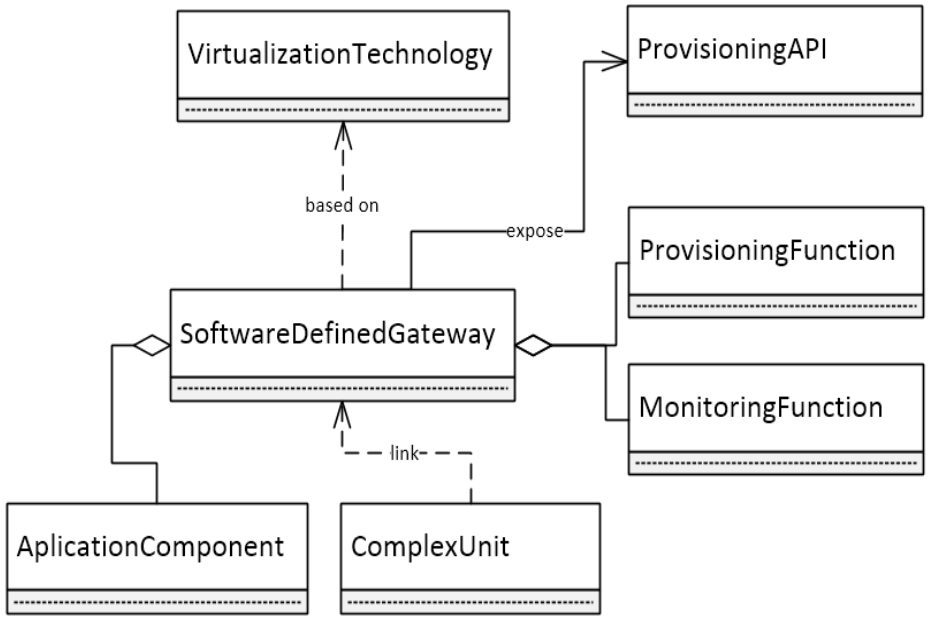
- Execute atop physical gateways
- Virtualize gateways compute and memory resources
- Act as isolated containers for applications => lightweight execution environment
- Enable on-demand provisioning of application, libraries and configuration models



Stefan Nastic, Sanjin Sehic, Le-Duc Hung, Hong-Linh Truong, and Schahram Dustdar. *Provisioning Software-defined IoT Cloud Systems*. FiCloud 2014. Barcelona, Spain.

Software-Defined Gateways – Provisioning Model

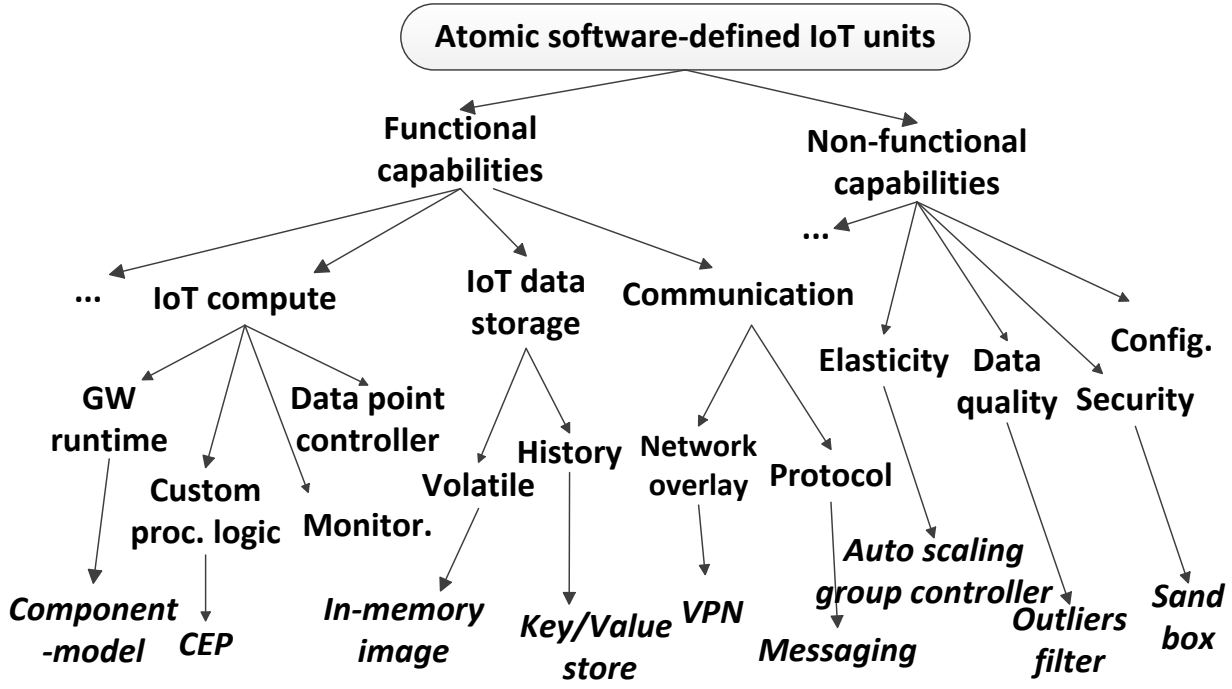
- Independent of the underlying virtualization technology
- Built from SDG prototypes
  - Based on kernel-supported virtualization: LXCs, libvirt-sandbox, chroot, etc.
  - Preconfigured with different functionalities, e.g., monitoring mechanisms
- Expose provisioning APIs used to deliver complex functionality
- SDG IoT Units enable encapsulating application components, libraries and configuration models



Provisioning Model (partial view)

SDGs Ecosystem

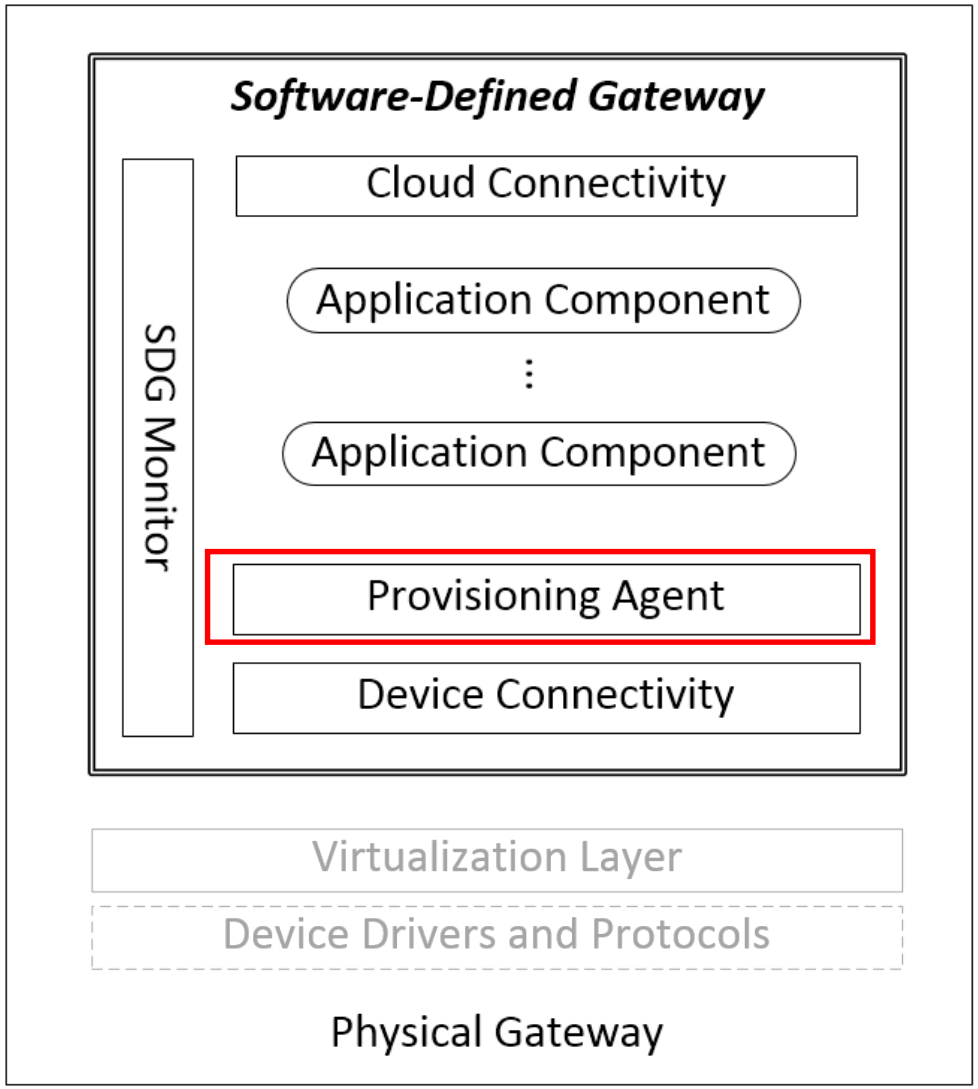
- Hierarchical structure of SDG components and capabilities.
- Enables distributing SDGs & SDG IoT Units in a market-like fashion, e.g., via SDG AppStore.



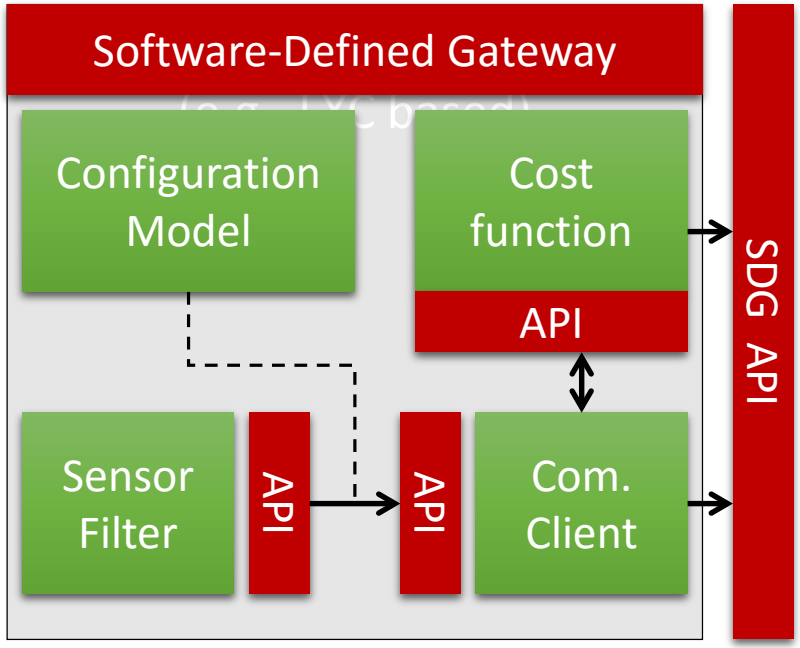
Hierarchy of basic SDG components (partial view)

Software-Defined Gateways – Architectural View

- Provisioning Agent
  - Handling remote provisioning requests
  - Dynamically downloading application components
  - Local installation of application components

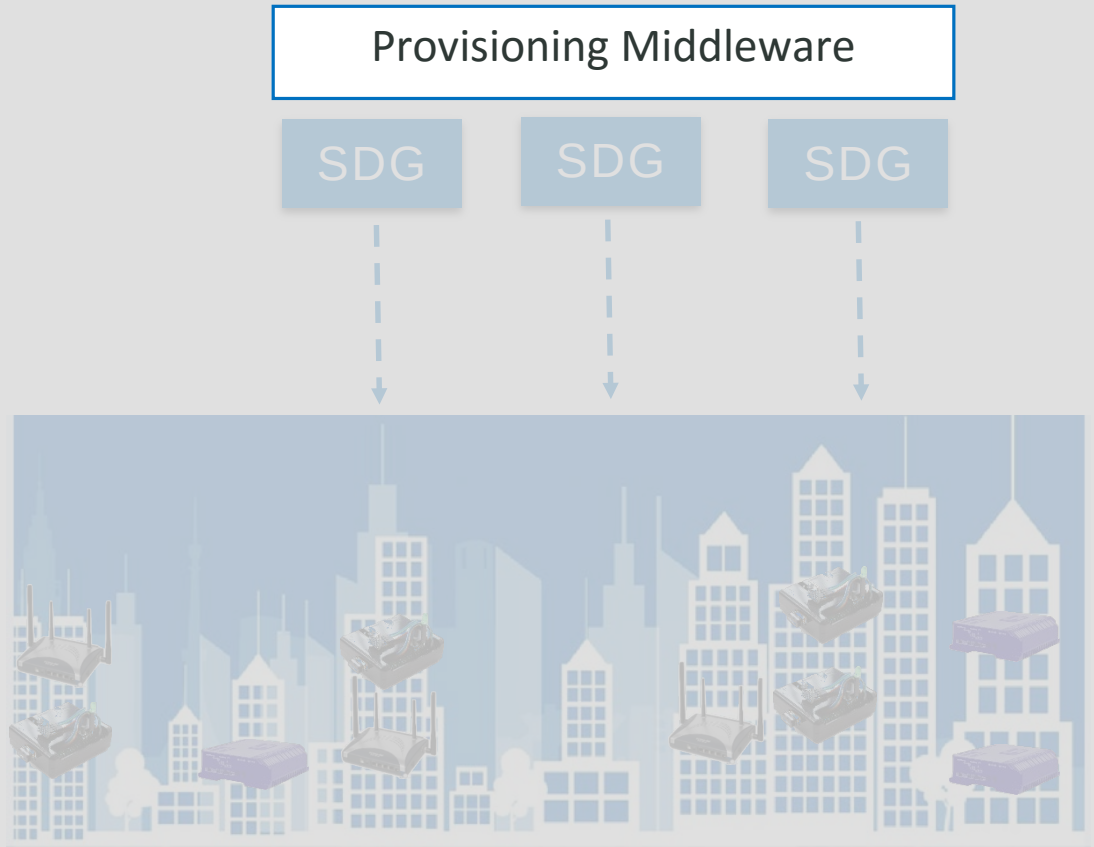


Software-defined Gateway - Example



Ready to be deployed on IoT devices such as physical gateways or cloudlets

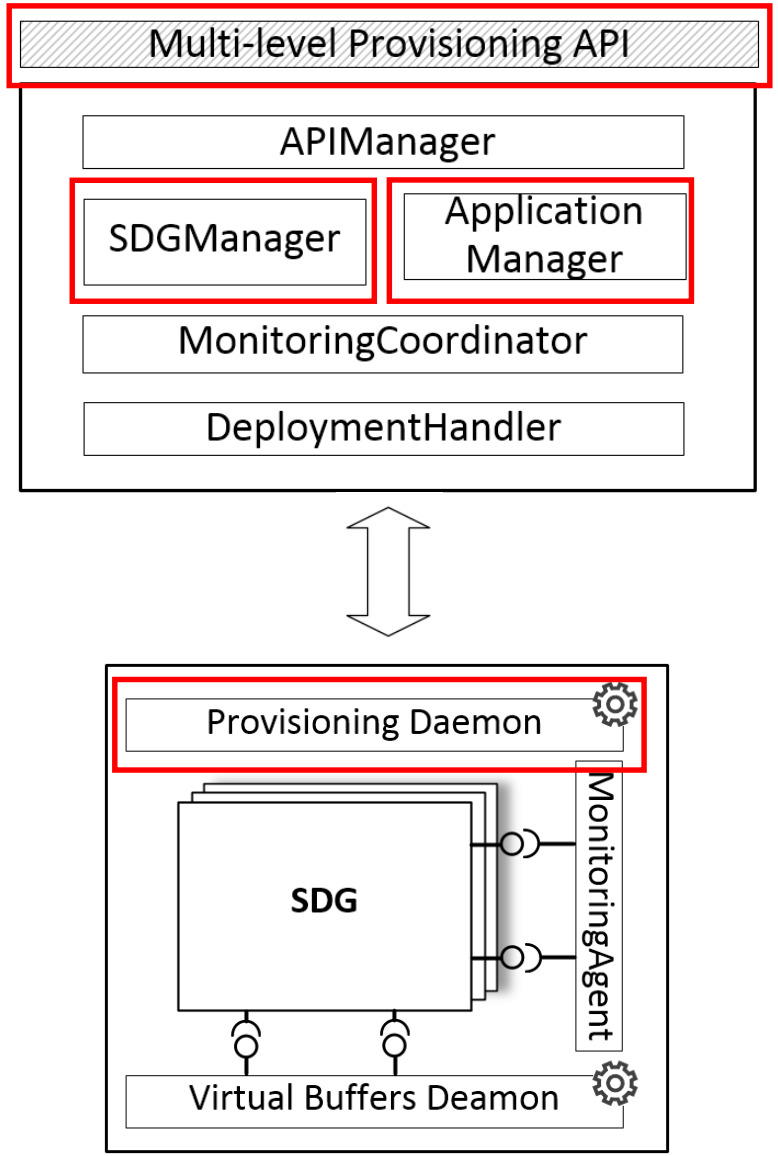
C2: Approach



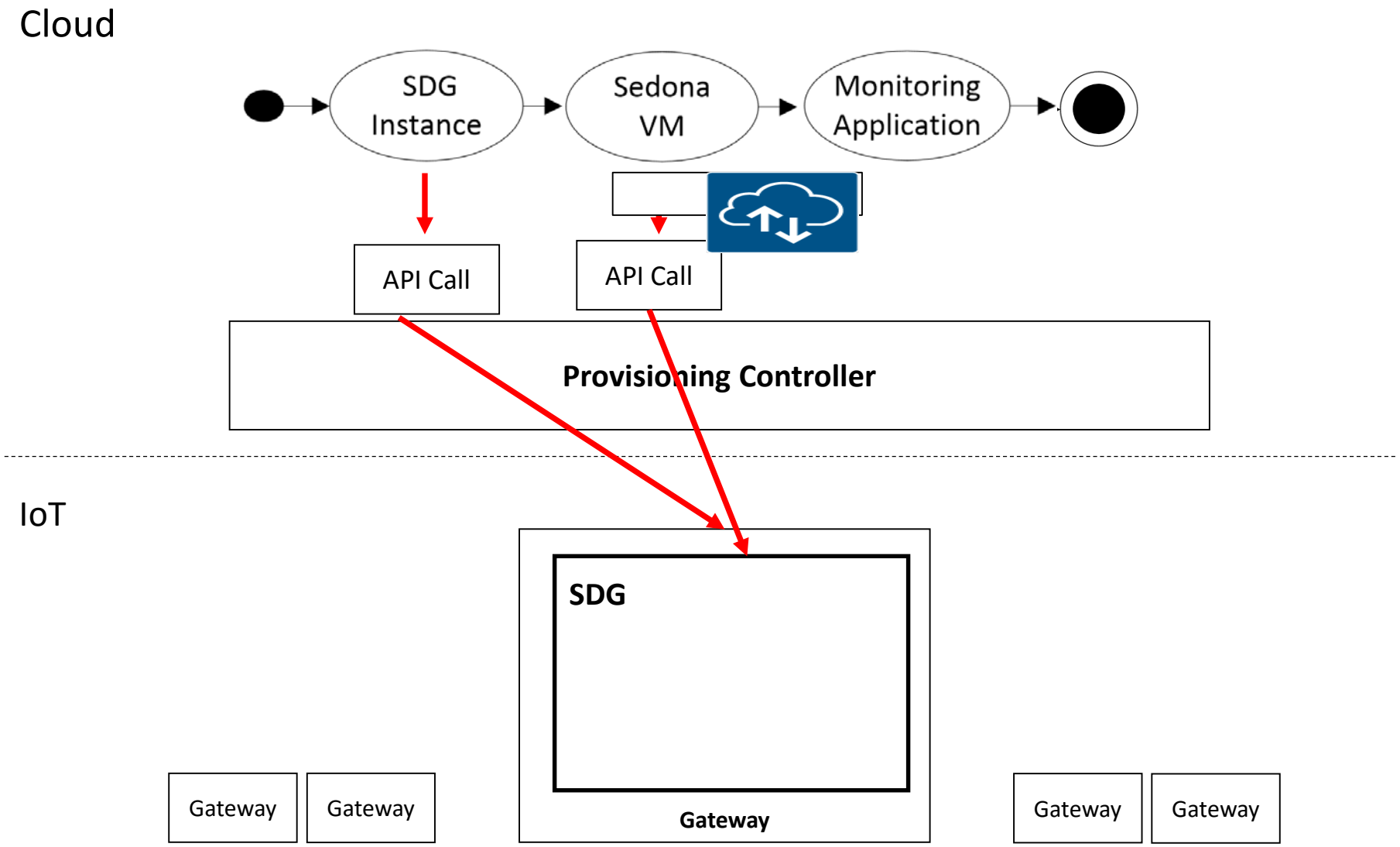
Provisioning Middleware

Cloud

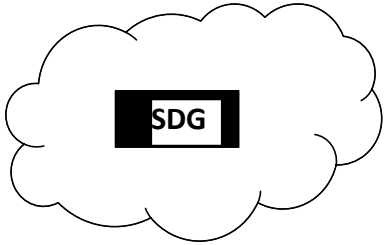
Edge device/
IoT Gateway



Example of SDG-driven Provisioning Process



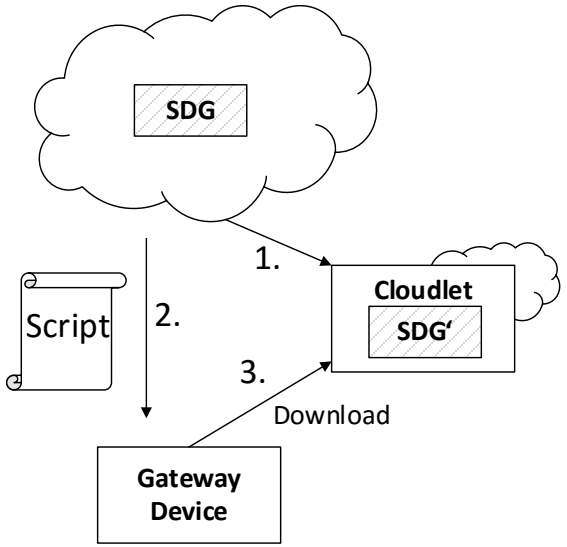
SDG Delivery Models

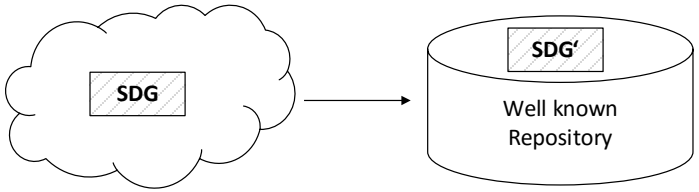


Push

Pull

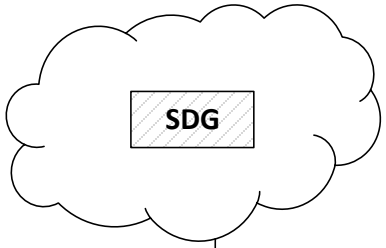
Gateway Device





Download

Gateway Device



Gateway Device

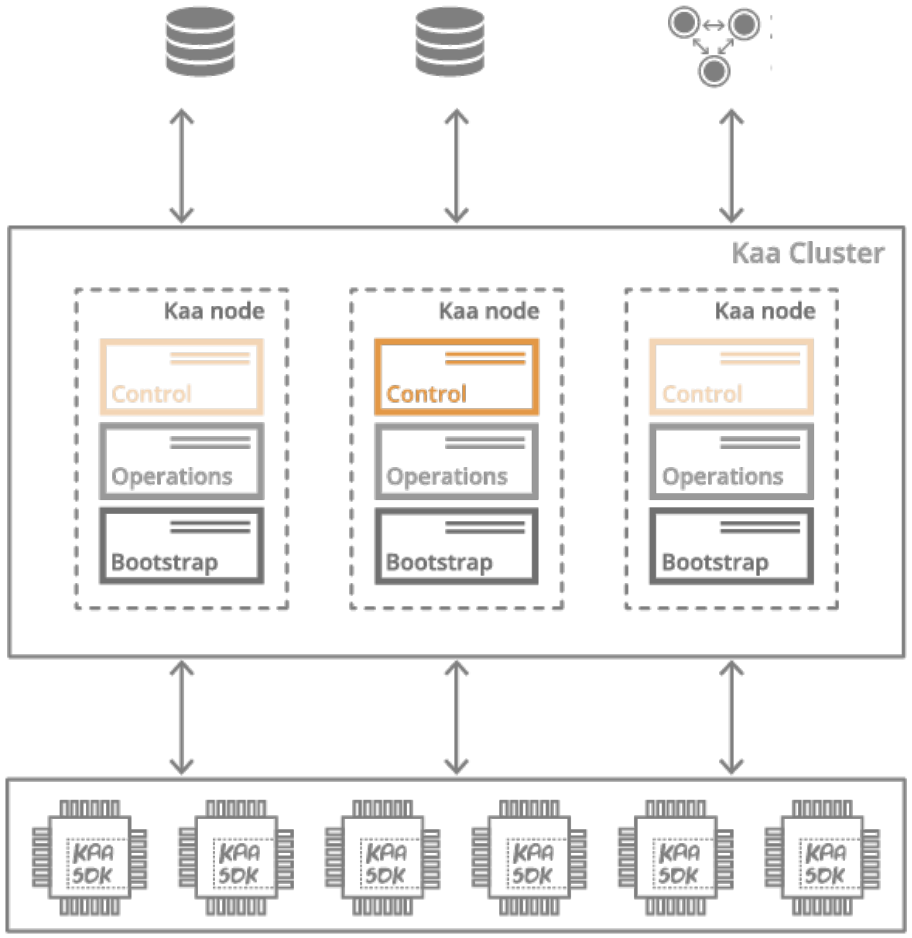
Technologies

3 IoT cloud platforms

KAA – data-centered middleware IoT platform

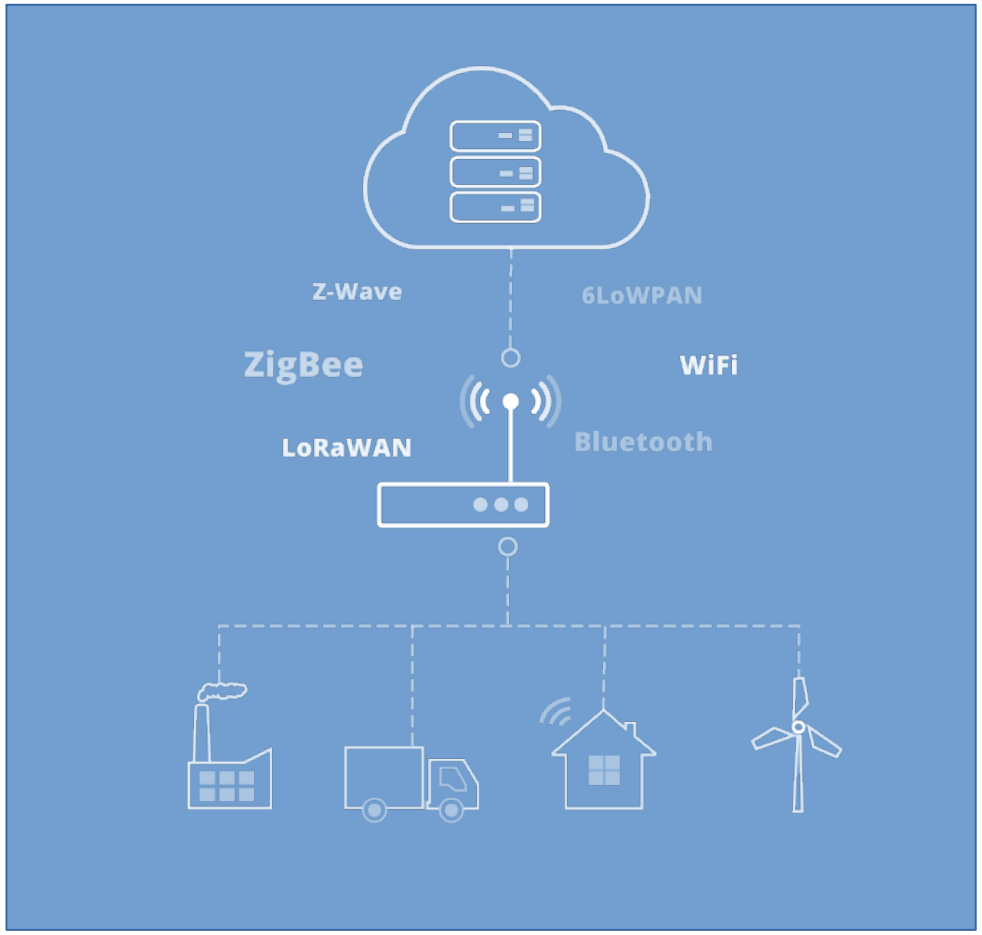
- Facilitates data exchange among attached devices, analytics/visualization, cloud services integration
- Hardware- and transport- agnostic
- Programming abstractions embedded in devices
- Open source -> www.kaaproject.org

KAA architecture



- KAA back-end used to manage tenants, applications, users and devices, exposes integration interfaces and administration
- Endpoint SDKs providing client-side APIs and handling communication, data marshalling, persistence etc.
- Databases store endpoint data/metadata

KAA: towards the edge?

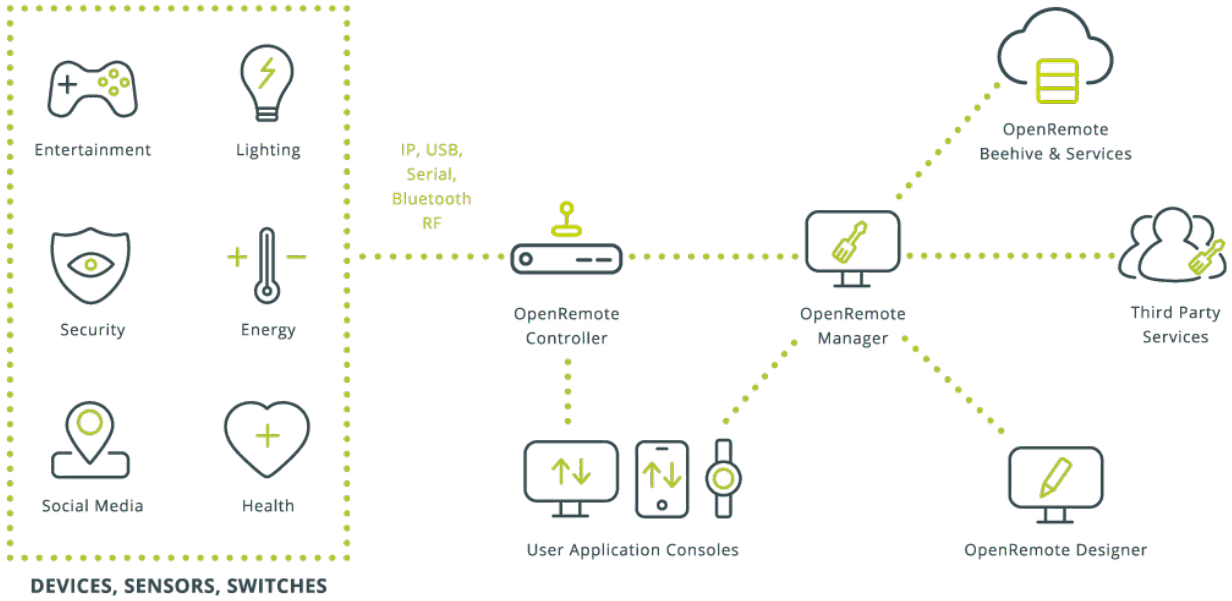


- Can be used to enable IoT features for gateways
- Edge-centered data collection and aggregation

OpenRemote

- Integrates different protocols and solutions for smart building/city automation
- Offers tailored visualization and administration facilities
- Open Source Middleware for IoT -> www.openremote.com

OpenRemote architecture

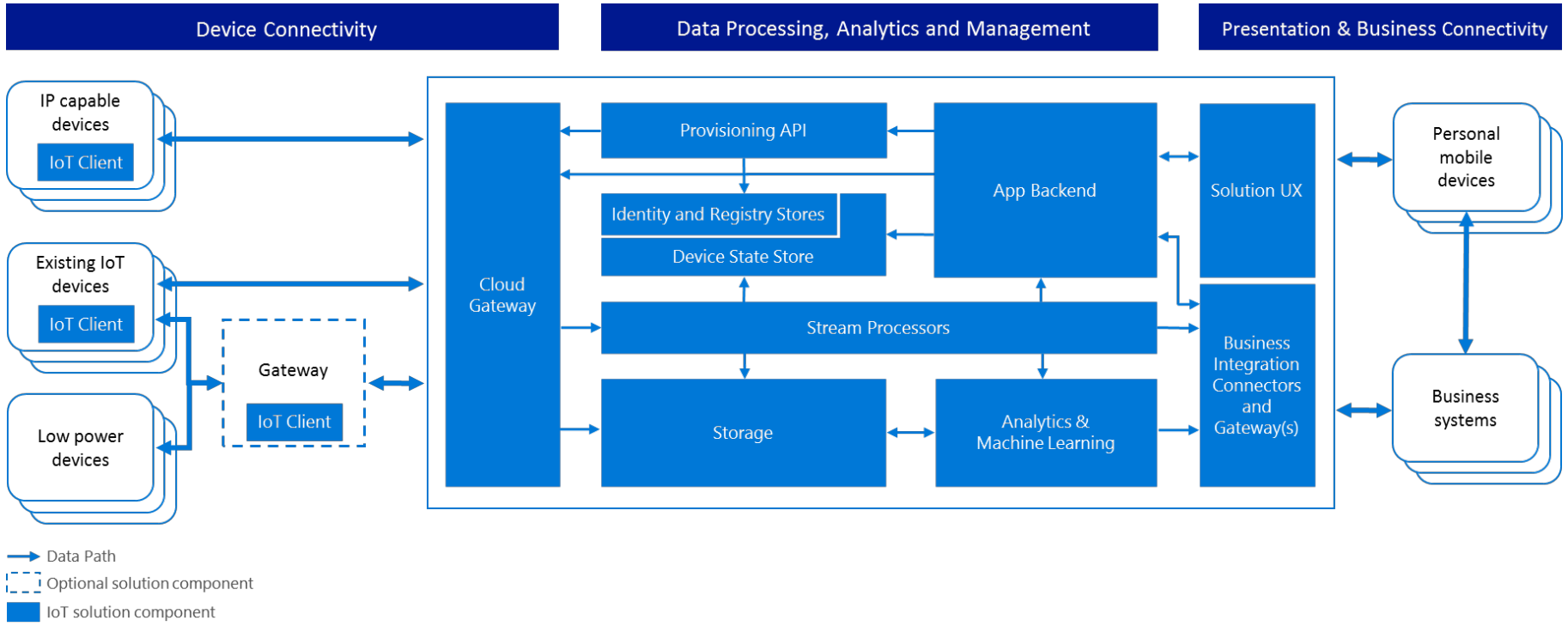


- Object model abstracting device, commands and data specifics
- Integration and automation logic of devices and sensors organized by a local runtime controller
- Technical integration and UI design facilitated by external hooks

OpenRemote architecture

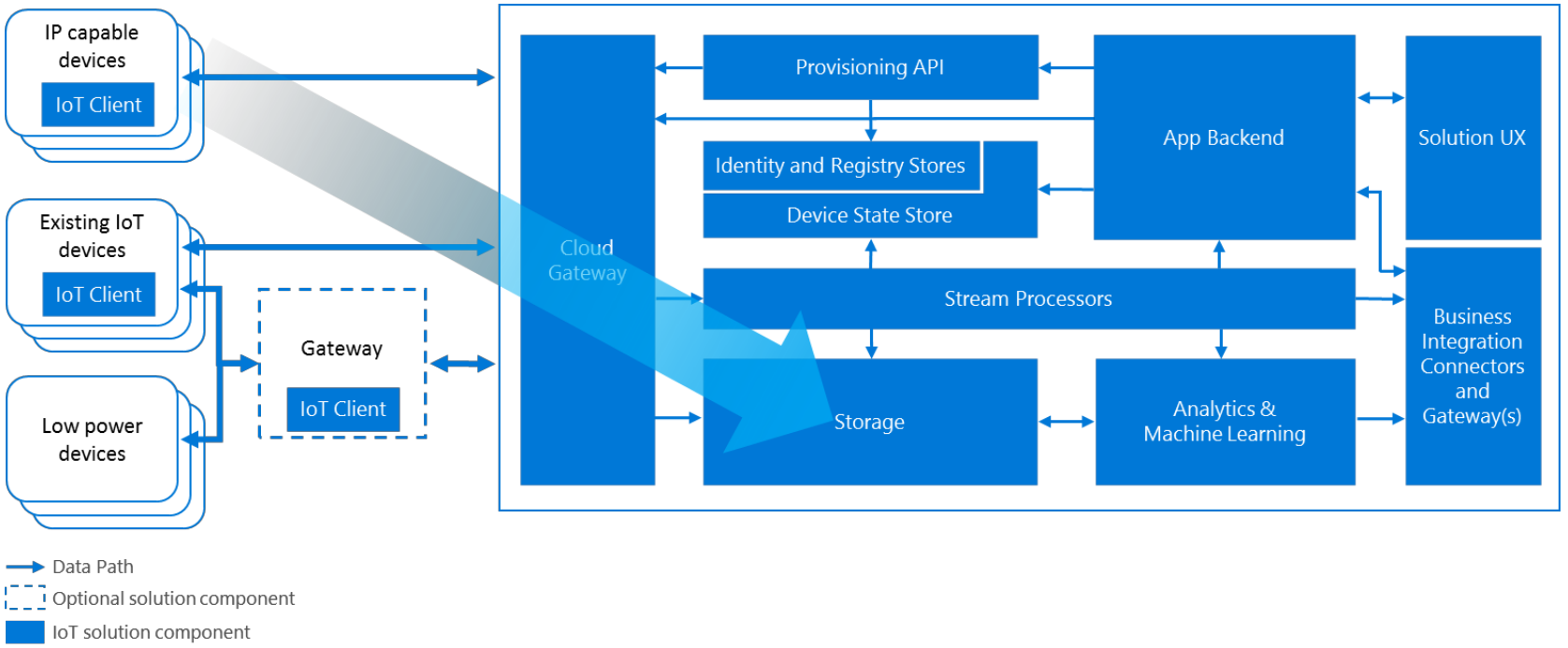
- Designed for large-scale industrial IoT environments
- Focused on a data stream abstraction
- Business systems integration
- Full-featured solution for enterprise use

Azure IoT architecture



- Layered over Device connectivity, Data processing and Business interface
- Reference architecture designed with scalability in mind

Azure IoT architecture



- Adopts fundamental abstraction of data streams - device and data models are not required to flow, route, or store information in the core platform components
- Data model is strictly neutral

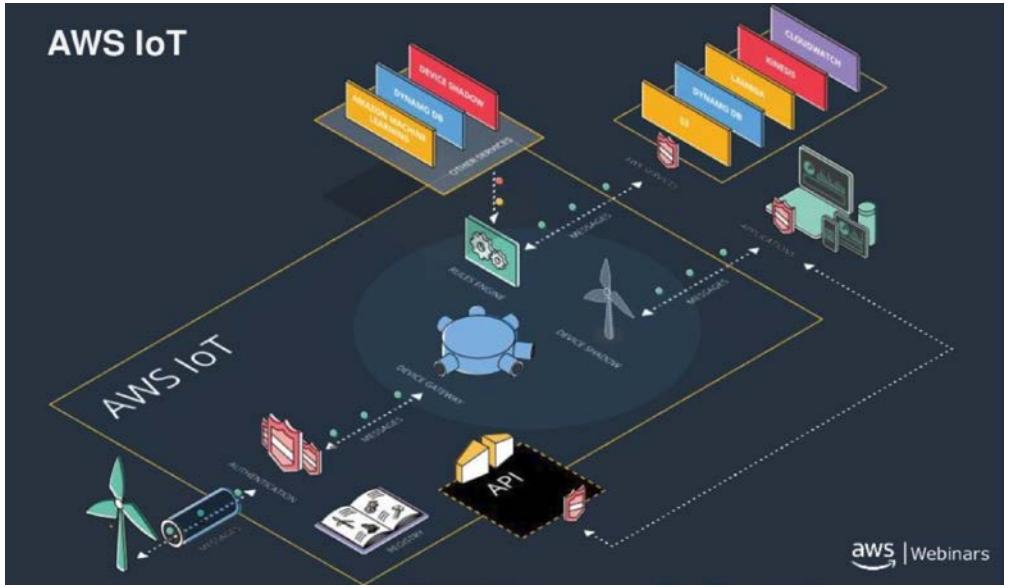
Towards the edge? Some thoughts on IoT cloud platforms

- More device heterogeneity introduced at the system’s edge
- Data models need to be re-thought (case for streams?)
- Increased complexity in coordination, communication - "gateways" are now first-class entities?

AWS Greengrass

AWS IoT – Background

- A managed platform that enables IoT devices interact with Cloud and other IoT devices
- Key concepts:
  - Device Shadow - A virtual representation of IoT device on the Cloud
  - Rules engine – Supports applying custom rules on IoT devices, e.g., to filter data



AWS Greengrass

- AWS Greengrass extends AWS Cloud to devices enabling them to act locally on the data
- Local compute, messaging, data caching, sync and ML models for IoT devices

AWS Greengrass key concepts:

- Greengrasscore (GGC)
- Device Shadows
- Lambda at the Edge
- Messaging Support

AWS Greengrass

- **GGC – the "hart" of AWS Greengrass**
- The runtime responsible for Lambda execution, messaging, device shadows, security, and for interacting directly with the cloud

- **AWS Lambda- AWS FaaS solution for event-driven compute functions**
- AWS Greengrass enables Lambda-at-the-Edge
- Deploying and executing the compute functions in IoT Devices

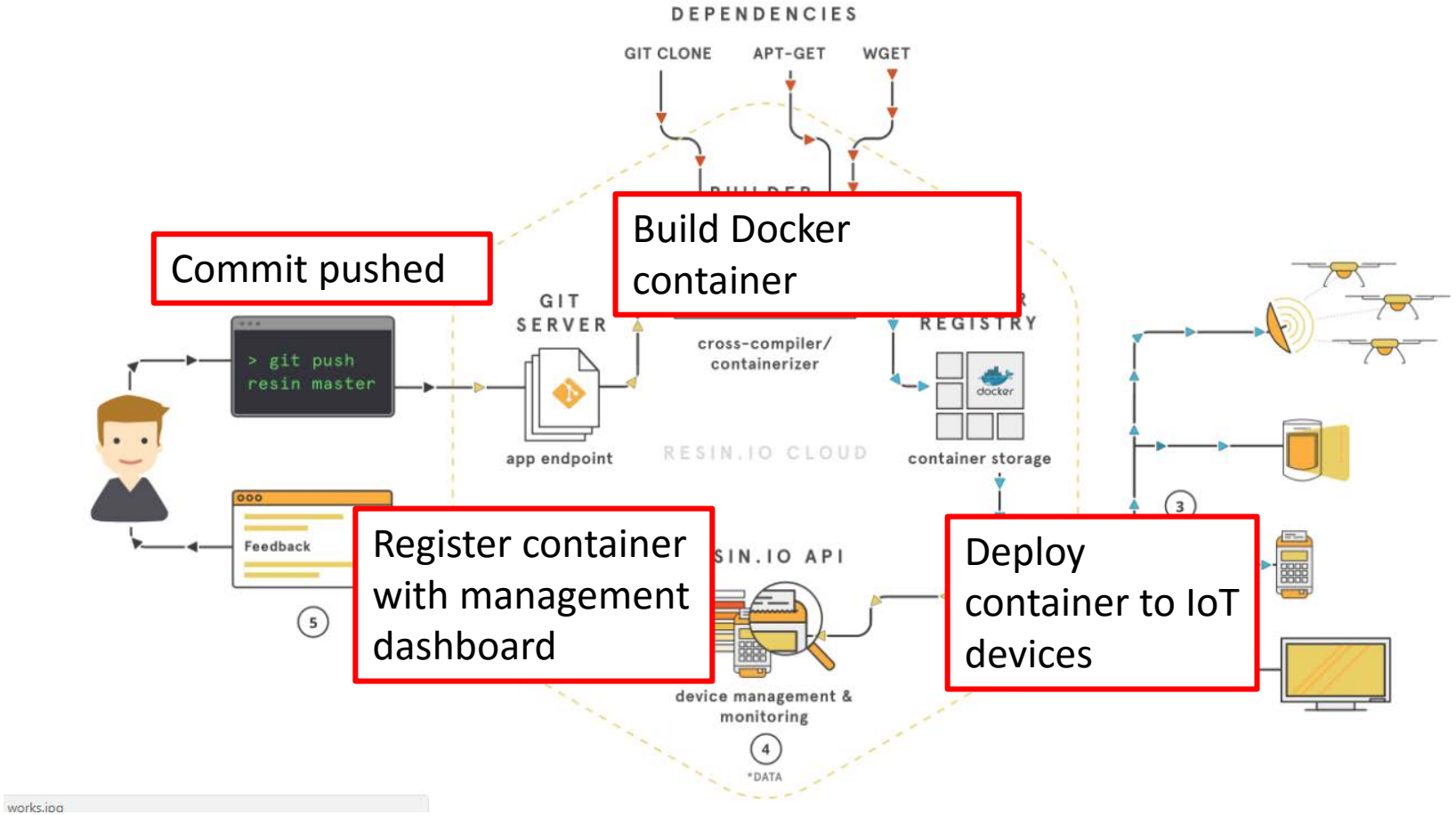
- **Local MQTT pub/sub messaging**
- Define subscriptions among Publishers & Subscriber (Cloud or IoT)
- Support for MQTT features such as topic-based filtering

Resin.io

- Resin.io is a container-based platform to develop, deploy, and manage code running on remote IoT devices
- Trying to bring DevOps to IoT world
- Similar idea to our SDGs

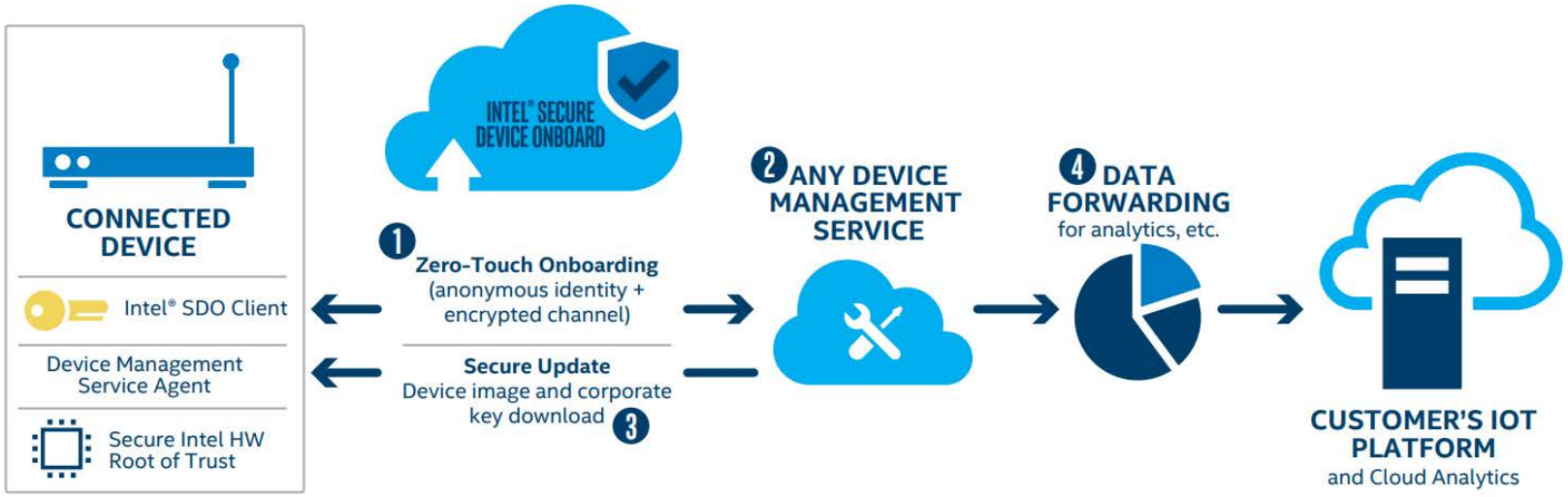
Key concept: Deploying incremental changes only – similar to “git diffs”

Resign.io Workflow



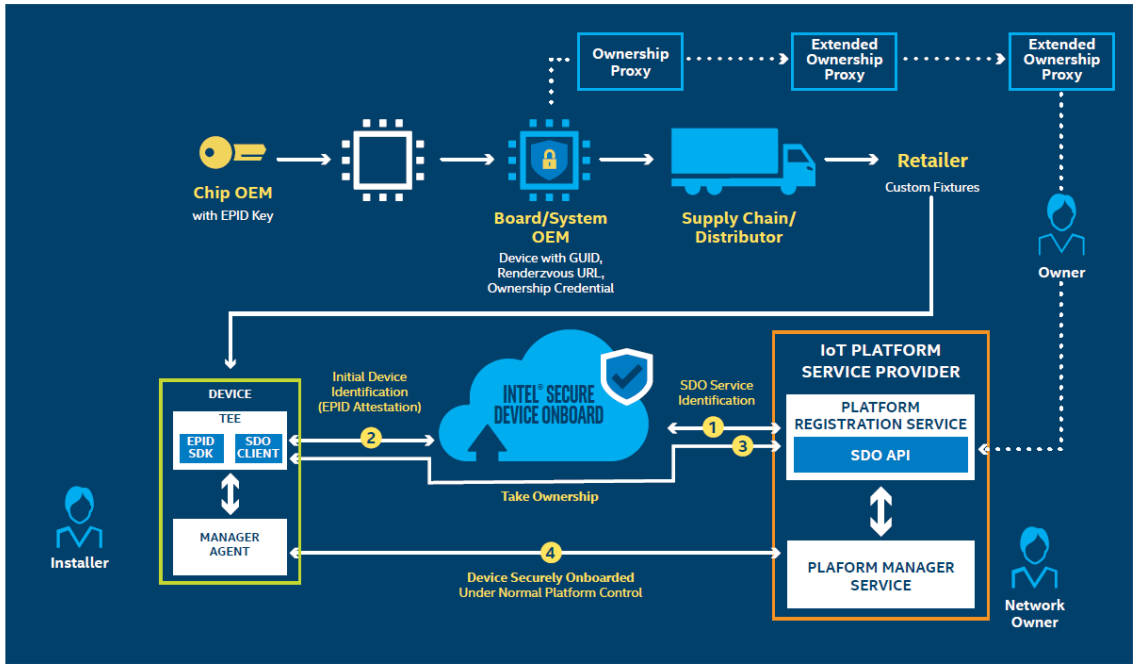
Intel Secure Device Onboard (SDO)

- SDO is and automated service that enables provisioning of IoT devices with any IoT platform
- Key features:
  - Compared to other solutions there is Hardware-based provisioning support
  - Mainly focuses on Device Identity Management
  - Also includes: IoT device management platform and different communication protocols
- Still not widely used – previews available



Intel Secure Device Onboard

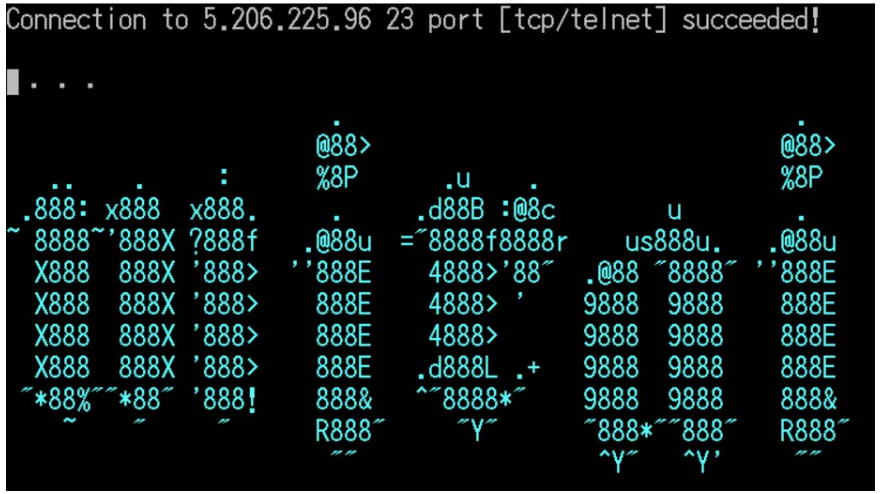
- Targets wider ecosystem:
  - Chip OEMs – Support for integrating Identity Management in silicon during manufacturing
  - IoT device manufacturer – Toolkit for inserting custom software into boot code
  - Device owner – Custom configuration of device ownership and Identity
  - IoT Platform – SDO provides service APIs to register containers running on the device with device owners account



IoT Security – Research Challenges

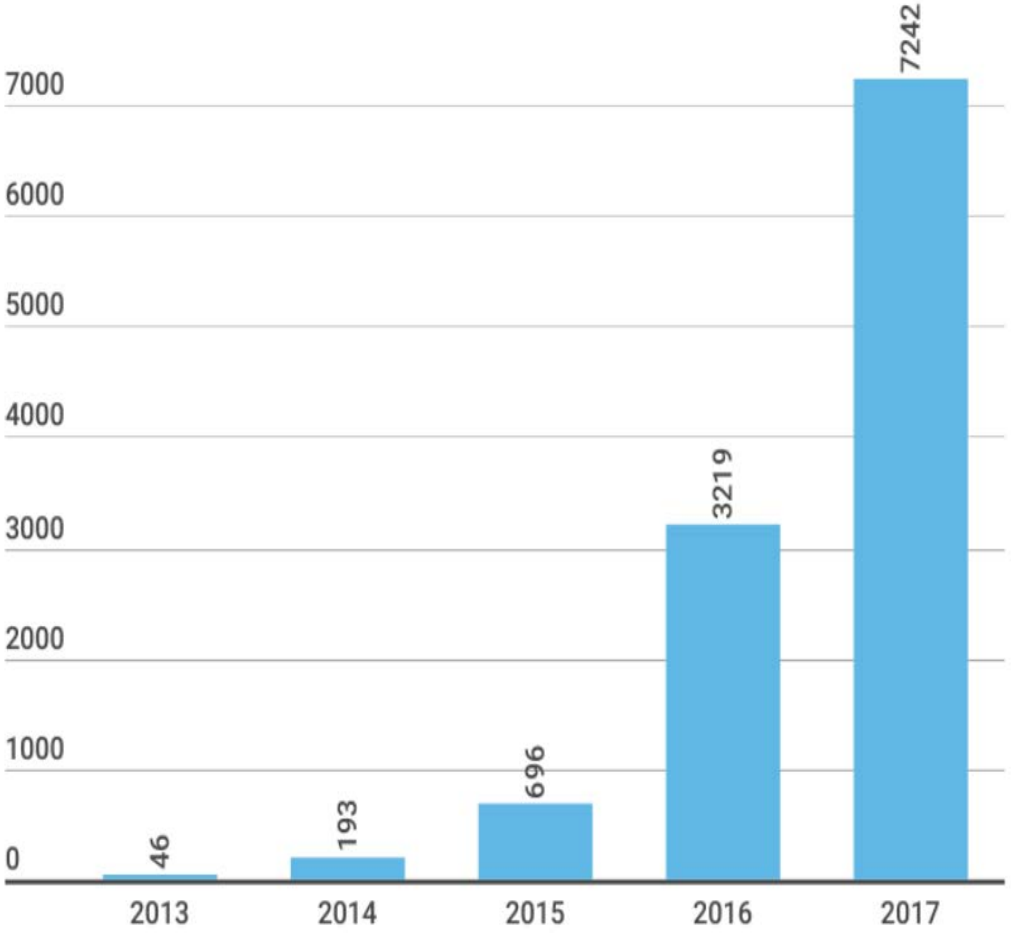
Credits and thanks to Prof Xiuzhen Cheng
George Washington University

Recent Incidents

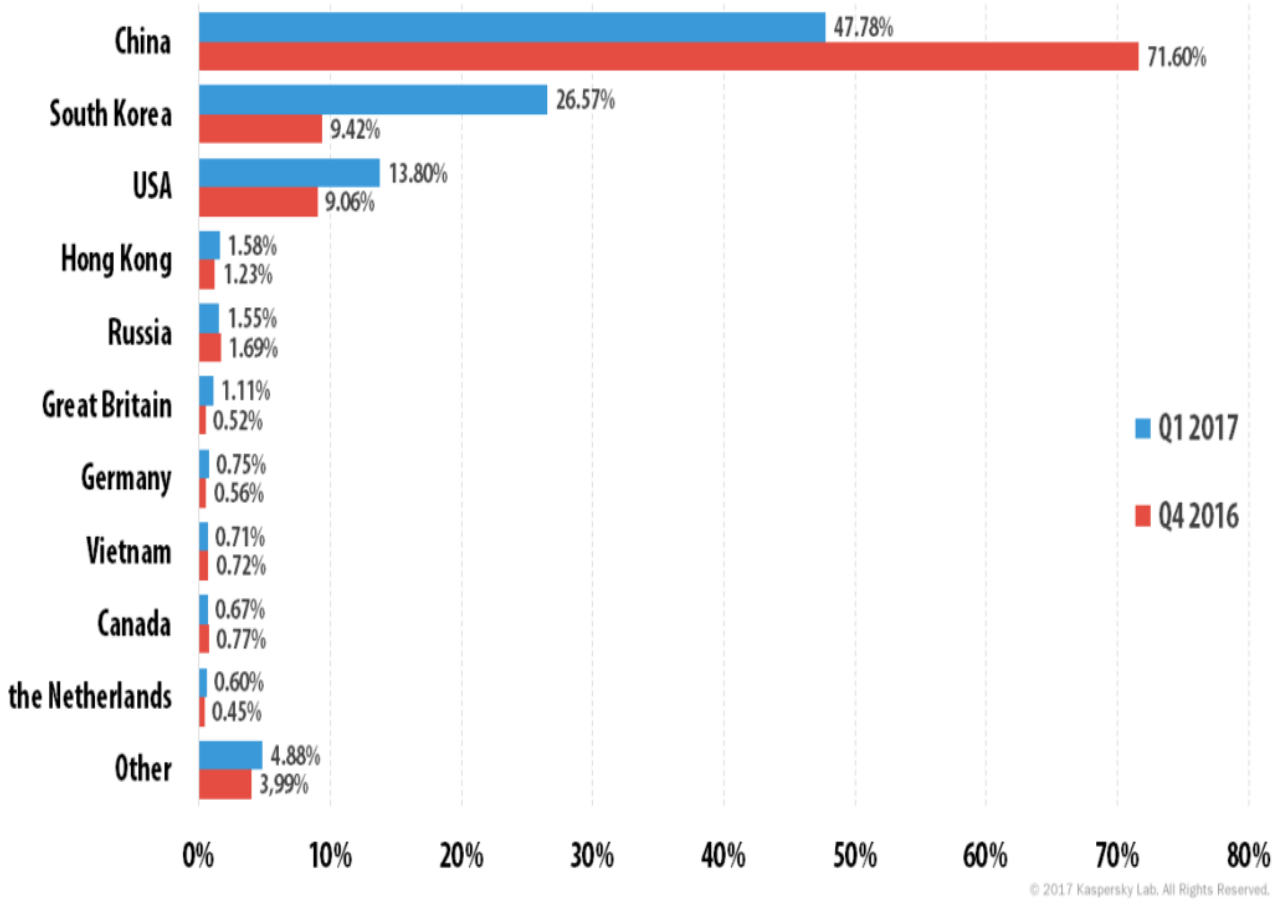


1*. https://bestsecuritysearch.com/mirai-linux-backdoor-attacks-iot-devices
2*. http://www.bbc.com/news/technology-33650491
3*. http://www.wipro.com/industries/medical-devices
4*. https://support.myharmony.com/en-us/harmony-experience-with-google-assistant
5*. https://www.bhphotovideo.com/c/product/1187819-REG/amazon_b00x4whp5e_echo.html
6*. https://www.pcmag.com/article/348496/google-home-vs-amazon-echo-which-one-should-rule-your-smar

IoT Security Is Serious



Number of Captured IoT Malwares



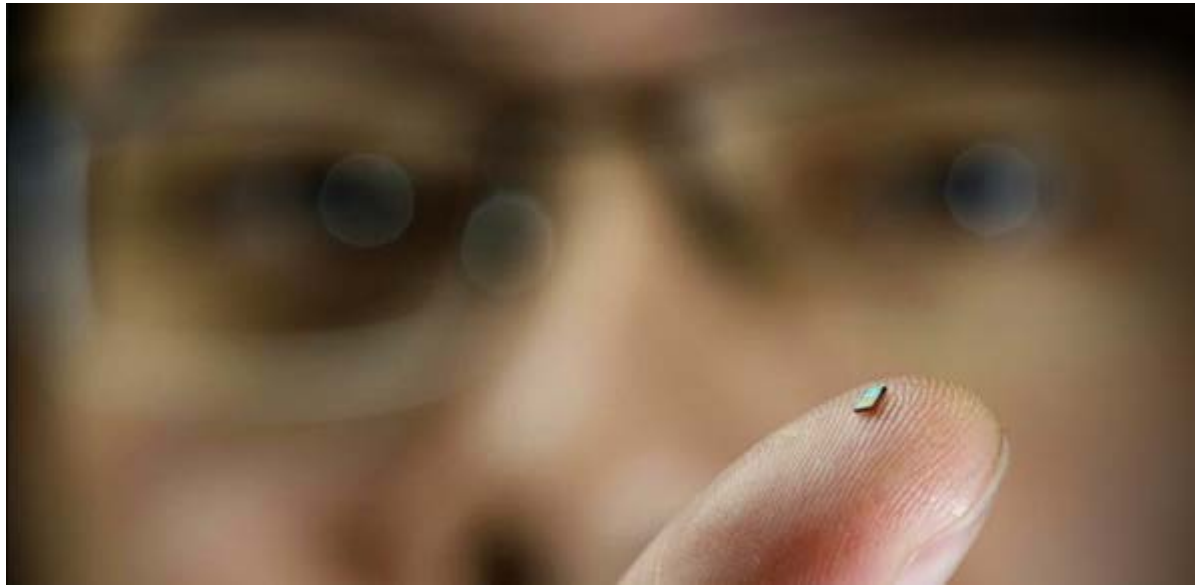
Top 10 Countries/Regions Being Victims of IoT Attacks

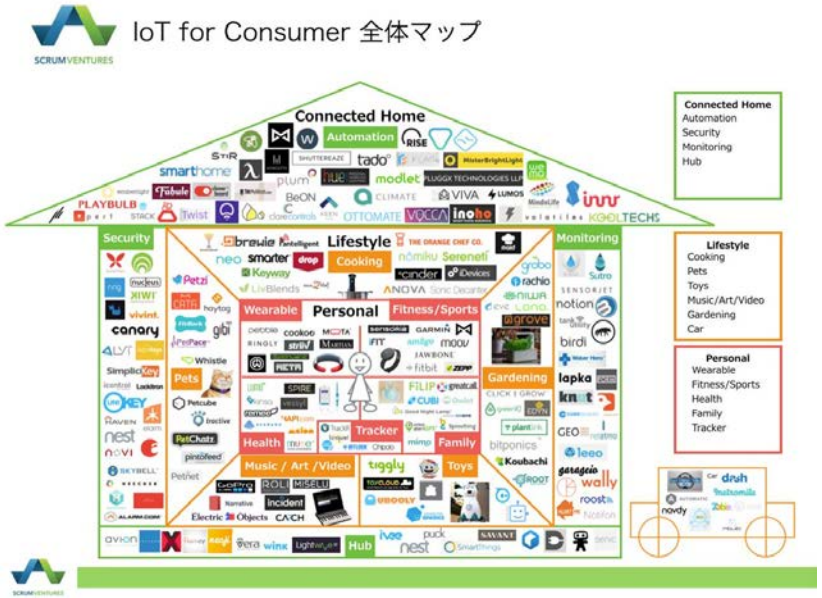
https://securelist.com/honeypots-and-the-internet-of-things/78751/
https://securelist.com/ddos-attacks-in-q1-2017/78285/

Challenges

Limited processing power and device interface; common mis-implementations and mis-configurations; conflict between user experience and security; etc

Diverse vendors and protocols; large amount of data; more device more problems; missing firmware upgrades for patching; lazy consumers; etc





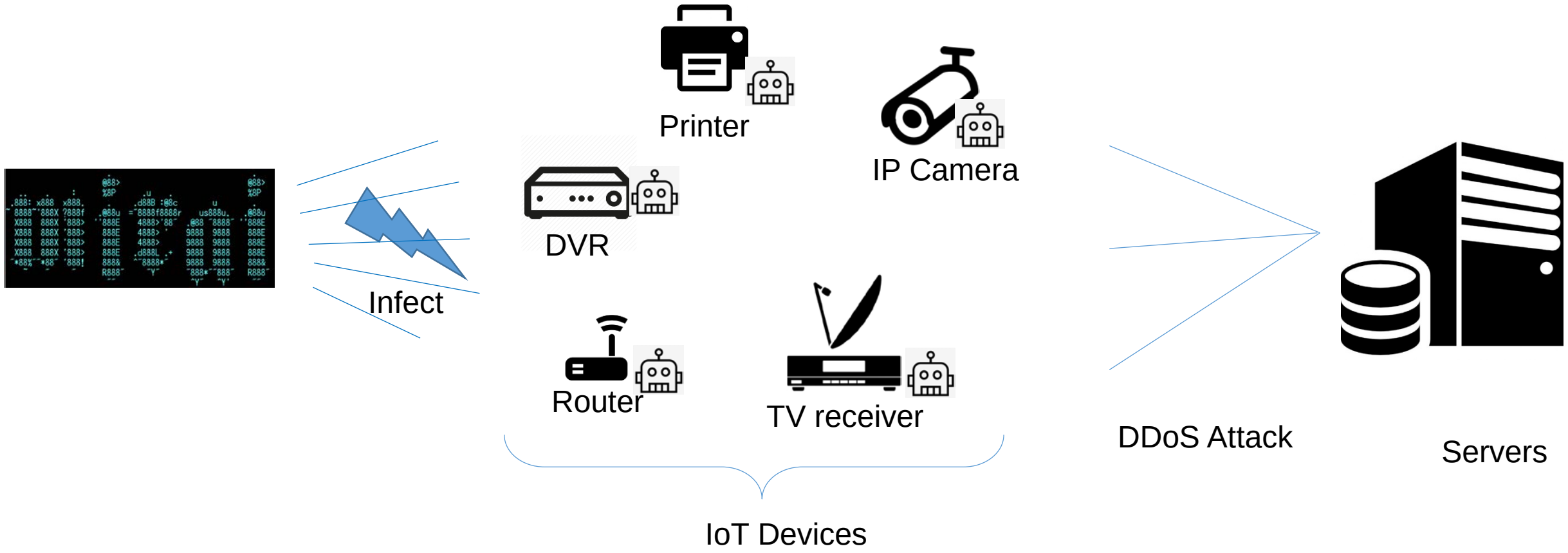
Smart home IoT brands

1*. http://www.toptechnews.com/article/index.php?story_id=0010002RHJ67

2*. https://iotnewsletter.org/the-internet-of-packaging-the-future-for-consumer-brands-iot-branding/

The Mirai Botnet

Mirai Botnet is a malware that focuses infections on IoT devices to collectively perform massive DDoS attacks.

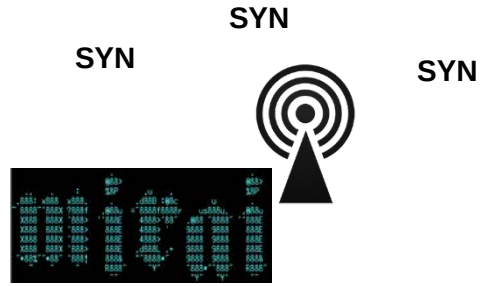


“IoT botnets are the new norm of DDoS attacks!”

How Does Mirai work?

Phase 1: Rapid Scanning

Flooding the network on TCP port 23 (the infamous **Telnet**) and 2323 with **SYN**. Search for open devices. If found:

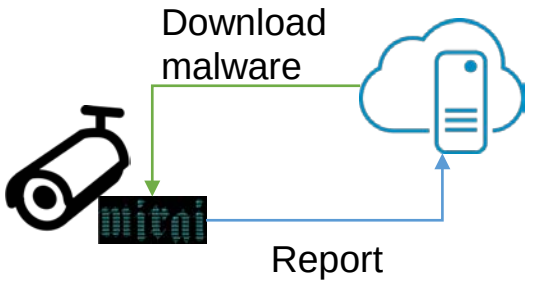


Phase 2: Brute-Force Login

Try **empirical passwords** to **brute-force login**, seeking Telnet connections. If connected:

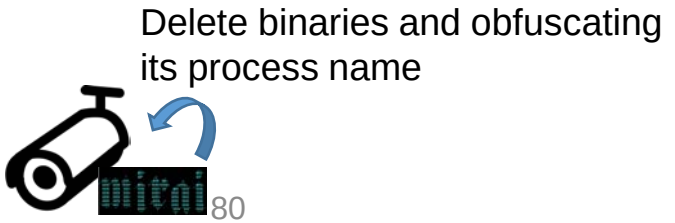
Phase 3: Loader Program

Send victim IP and credentials to the *report server*.
Download the **architecture-specific malware**.



Phase 4: Execution and concealment

Execute malware and conceal itself.



Impact: What Did Mirai Do?

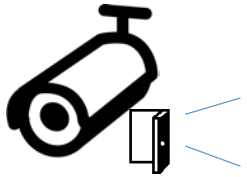
Time	Event	Impact detail
Sep 2016	**Emergence:** DDoS attacks on KrebsonSecurity (In-depth security news and investigation) and OVH	• **65,000+** burst infections in 20h, • **600Gbps** DDoS volume - one of the largest on record!
	A Spirit of Sharing: Mirai source code publicly released.	• @hackforums.net • Permits variants to **evolve**.
Oct 2016	**Variant:** DDoS attacks on DNS provider: Dyn and Lonestar (Lonestar Cell)	• Consistent **SYN/ACK** flood.
Nov 2016	**Variant:** exploiting routers through CPE WAN Management Protocol, struck Deutsche Telekom	• Target on **routers** • Knocked **900,000+** Germans offline.
…		
Dec 2017	**Variant:** exploiting a zero-day flaw in *Huawei* HG532 routers	
Jan 2018	**Variant:** hijack Cryptocurrency mining operations	
Jan 2018	**Variant:** weaponises EDB 38722 D-Link router's exploit to enlist further vulnerable IoT devices	

Event timeline

Reflections: Problems Revealed

IoT device fragility is exposed:

1. Many IoT devices' ports are **open by default**.

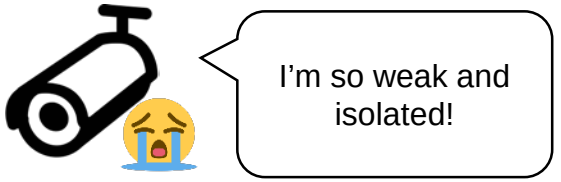


Everybody welcome!

2. Simple, default, and **static passwords**.

Password	Device Type	Password	Device Type
123456	ACTi IP Camera	klv1234	HiSilicon IP Camera
anko	ANKO Products DVR	jvbzd	HiSilicon IP Camera
pass	Axis IP Camera	admin	IPX-DDK Network Camera
888888	Dahua DVR	system	IQinVision Cameras
666666	Dahua DVR	meinsm	Mobotix Network Camera
vizxv	Dahua IP Camera	54321	Packet8 VOIP Phone

3. Lack of timely **update, roll-back,** and **alerting** mechanisms.



4. Lack of **defragmented** operating system.

Too heterogeneous

Overprivilege in IoT

Overprivilege Issue

- Samsung's **SmartThings** [SP'16]

Defense Mechanisms

- ContextIoT [NDSS'17]
- FlowFence [USENIX Security '17]
- SmartAuth [USENIX Security '17]

[Samsung's SmartThings] E. Fernandes, J. Jung, and A. Prakash, Security Analysis of Emerging Smart Home Applications, 2016 IEEE S&P, pp. 636-654.

[ContexIoT] Y. Jia, Q. Chen, S. Wang, A. Rahmati, E. Fernandes, Z. Mao, and A. Prakash, ContexIoT: Towards Providing Contextual Integrity to Appified IoT Platforms, in NDSS'17.

[FlowFence] E. Fernandes, J. Paupore, A. Rahmati, D. Simionato, M. Conti, and A. Prakash, FlowFence: Practical Data Protection for Emerging IoT Application Frameworks, in USENIX Security 2017.

[SmartAuth] : Y. Tian et al. SmartAuth: User-Centered Authorization for the Internet of Things, in USENIX Security 2017.

Why SmartThings?

- Has a growing set of SmartApps (>500) downloadable from the app store
  - Far more than other frameworks
- Has native support for 132 device types from major manufactures
- Shares key security design principles with other frameworks
  - Has a privilege separation mechanism called **capabilities**
    - Specify the set of operations a SmartApp may issue to a compatible smart home device
  - Event-driven processing allows SmartApps to register events generated by devices

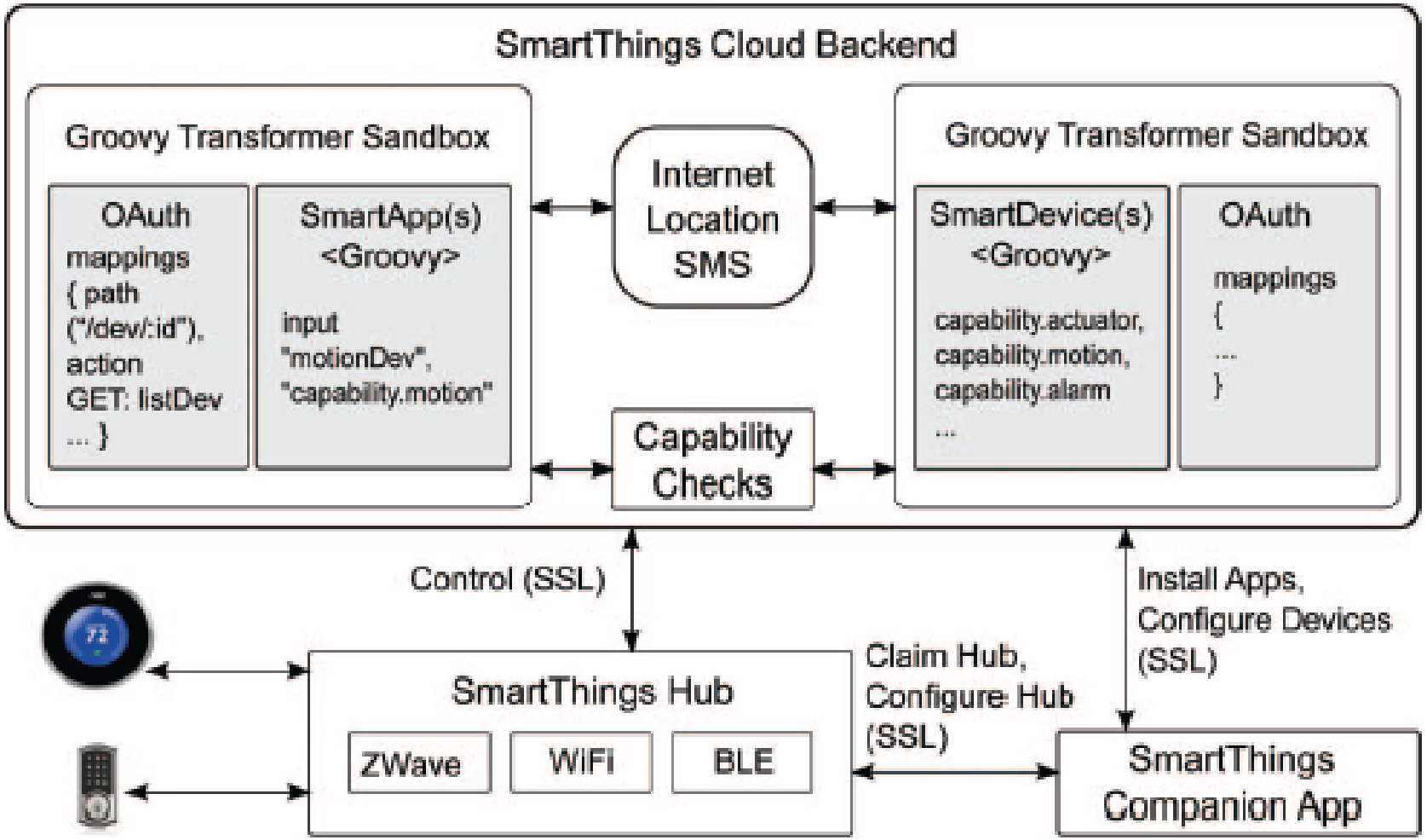
[Samsung's SmartThings]

```
definition(
    name: "DemoApp", namespace: "com.testing",
    author: "IoTPaper", description: "Test App",
    category: "Utility")

//query the user for capabilities
preferences {
  section("Select Devices") {
    input "lock1", "capability.lock", title:
        "Select a lock"
    input "sw1", "capability.switch", title:
        "Select a switch"
  }
}

def updated() {
  unsubscribe()
  initialize()
}

def installed() {
  subscribe sw1, "switch.on", onHandler
  subscribe sw1, "switch.off", offHandler
}

def onHandler(evt) {
  lock1.unlock()
}

def offHandler(evt) {
  lock1.lock()
}
```

An example SmartApp that locks/unlocks a physical door lock based on the on/off state of a switch

Two event subscriptions to sw1

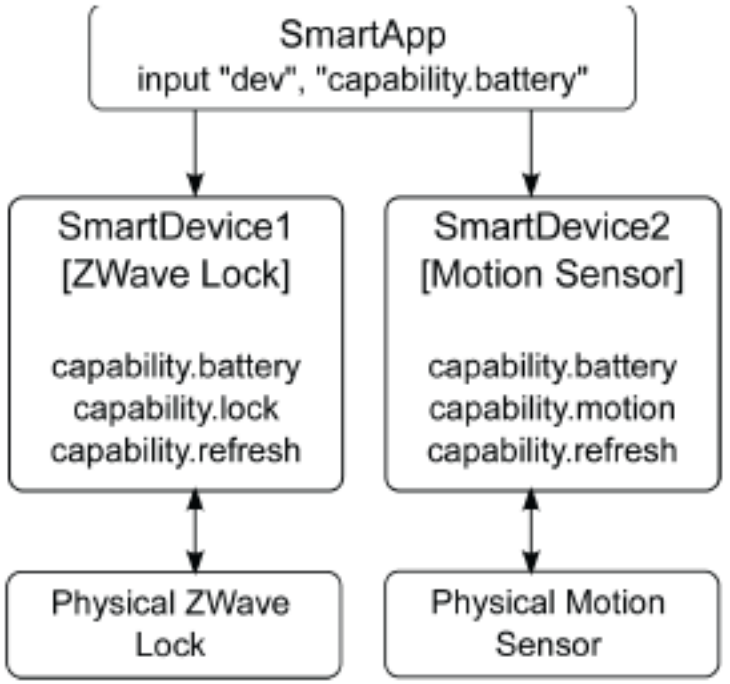
SmartThings Architecture Overview [Samsung's SmartThings]



SmartThings Capability Model

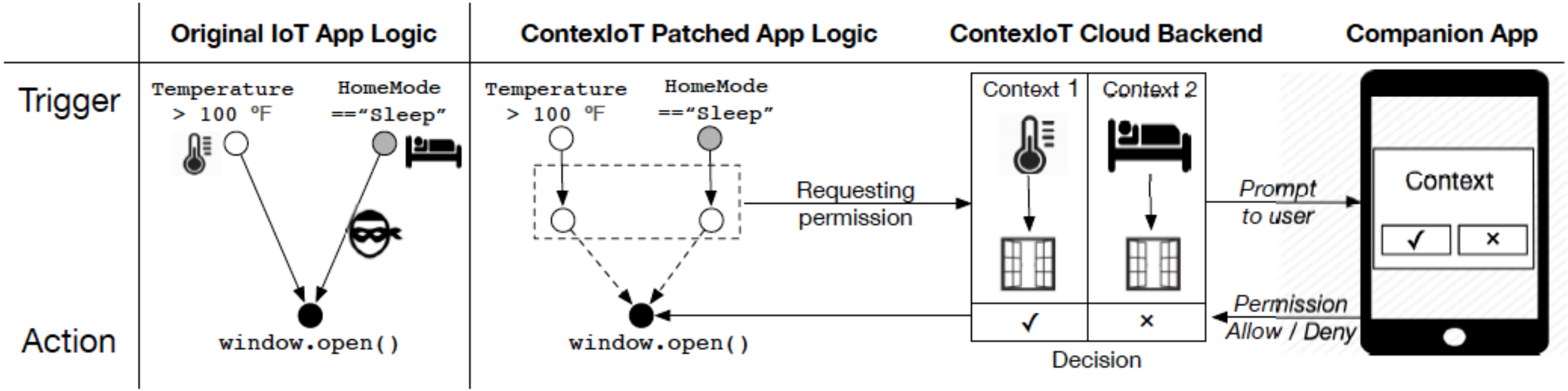
- A capability is a set of commands and attributes
- Multiple devices can be bound to a capability variable
- Overprivilege Issues
  - Coarse-grained capabilities
    - An AutoLoc app got access to the unlock commands, increasing the attack surface
    - Asymmetry in risks with device commands
    - Causing 55% of SmartApps overprivileged
  - Coarse SmartApp-SmartDevice bindings
    - Get access to all commands and attributes of all capabilities once a device is granted one capability
    - Causing 42% of existing SmartApps overprivileged
  - Event leakage via capability based access
    - A SmartApp can monitor all events of a bound device
    - Some events contain sensitive data (e.g. the lock pin-code)

Capability	Commands	Attributes
`capability.lock`	`lock()`, `unlock()`	`lock` (lock status)
`capability.battery`	N/A	`battery` (battery status)
`capability.switch`	`on()`, `off()`	`switch` (switch status)
`capability.alarm`	`off()`, `strobe()`, `siren()`, `both()`	`alarm` (alarm status)
`capability.refresh`	`refresh()`	N/A



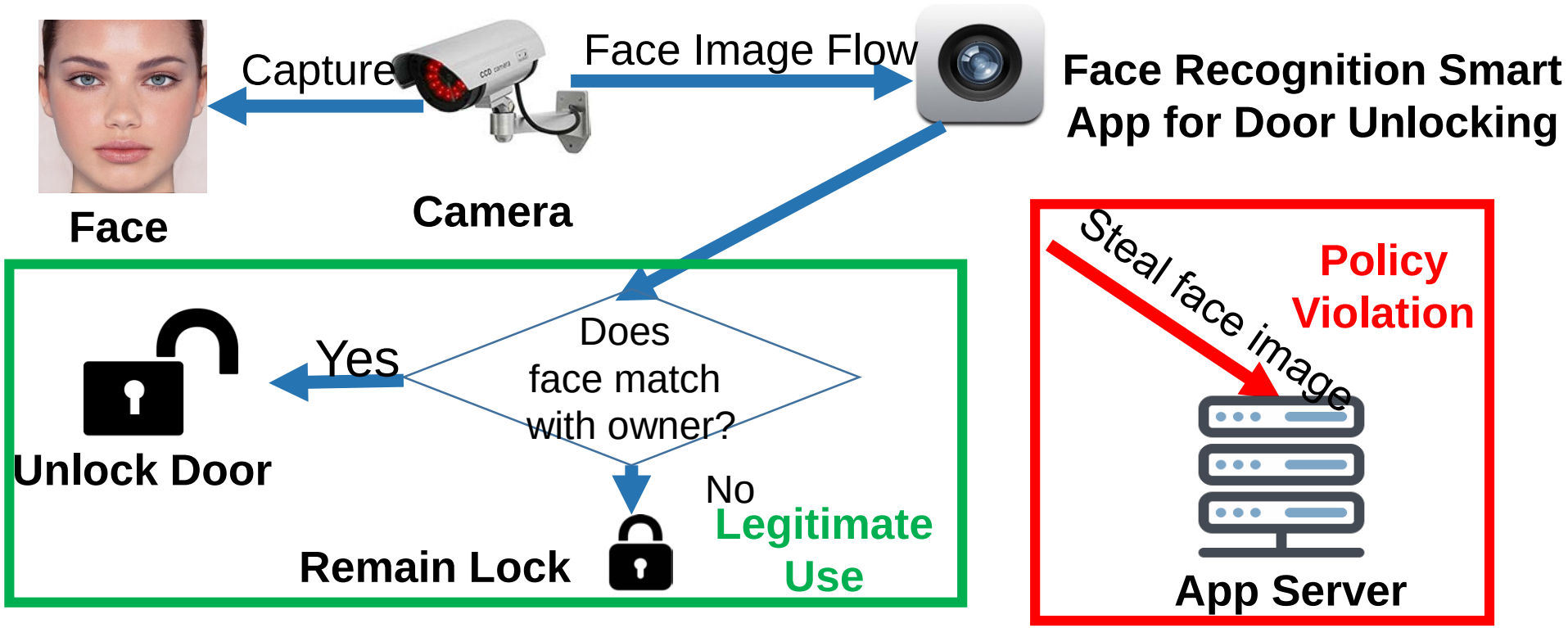
Defense – ContexIoT [NDSS'17]

- Ensure "contextual integrity" – context is defined to be the set of information that is essential to distinguish the attack and benign logic in an app at runtime
  - UID/GID (app id); UI Activity (usually not available for IoT); Control flow (a remote unlock command *vs.* correct pin code); Runtime value; Data flow (data dependency)
  - ContexIoT: 1) at app installation: patch the app during installation; 2) at runtime: collect context, request permission, and act; the backend server processes requests made by the patched app



Defense – FlowFence [USENIX Security'17]

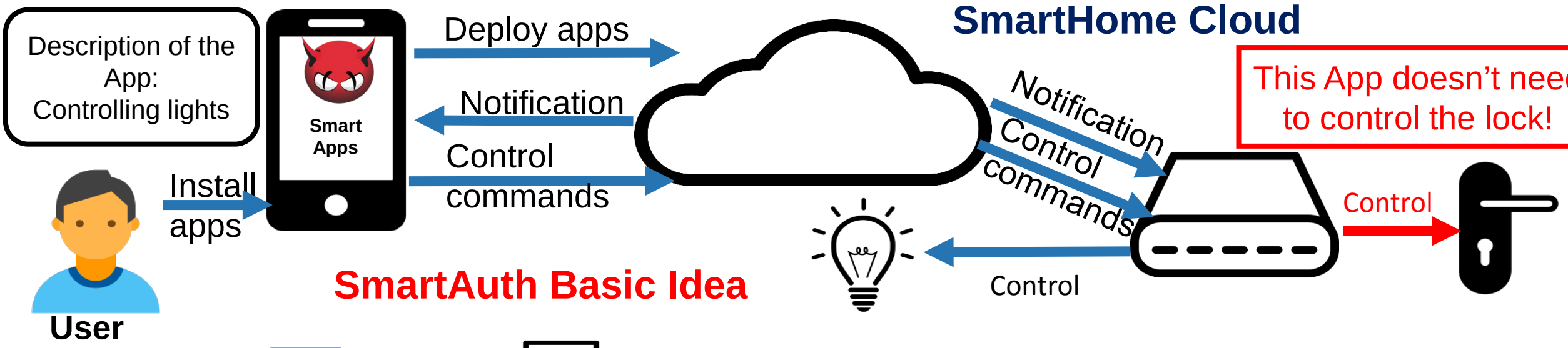
- Permission-based access control is ineffective at controlling *how* apps use data once they gain access – it can only control *what* data an app gets



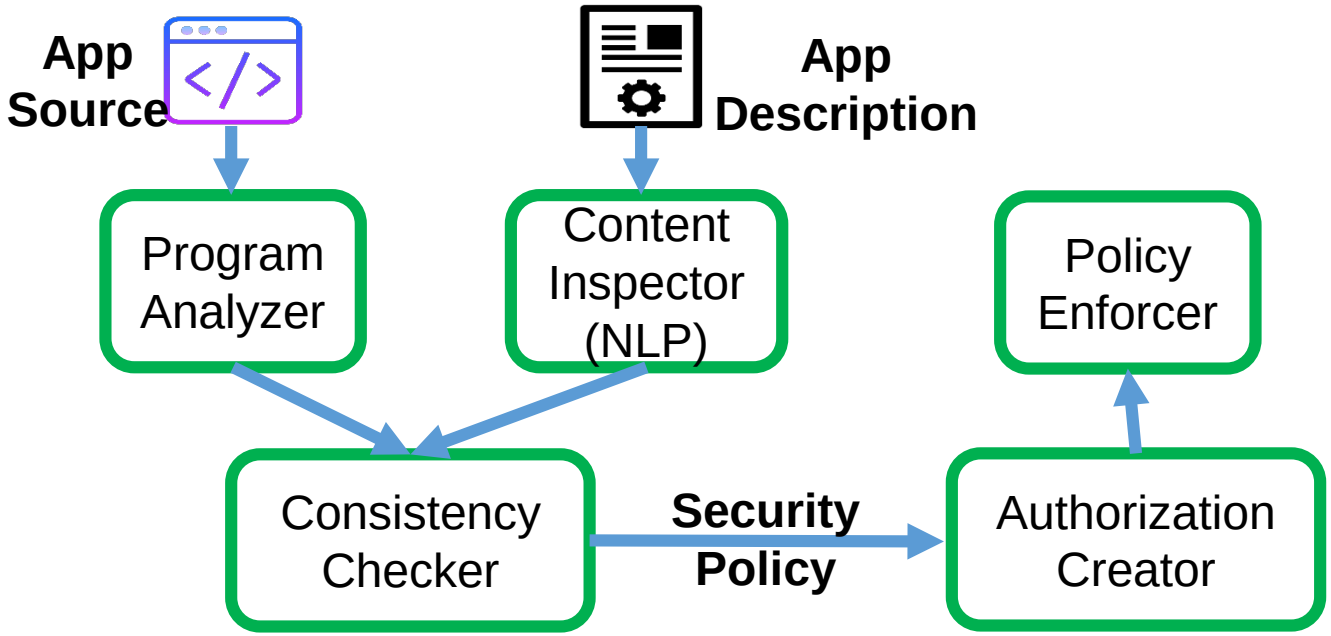
FlowFence idea:

- Any function accessing sensitive data has to run in the sandbox
- All sensitive data are only available in the sandbox
- Trusted sinks check the flow policy to make sure no violation occurs

Defense – SmartAuth [USENIX Security'17]



SmartAuth Basic Idea



Functionalities explained to the user

≠

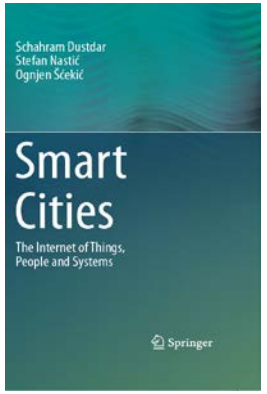
Operations that the app indeed performs

Foundational Conclusion

- Partnership Model for IoT/Edge/Cloud

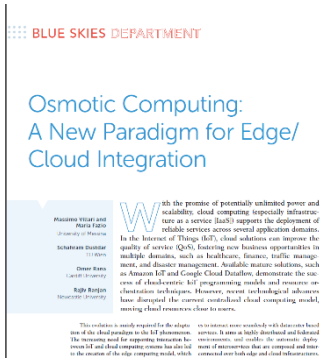
  Garcia J. M., Fernandez P., Ruiz-Cortes A., Dustdar S., Toro M. (2017). Edge and Cloud Pricing for the Sharing Economy. *IEEE Internet Computing*, Volume 21, Issue 2, pp. 78-84

- Book (IoT, People and Systems)



- 3 Major Paradigms (E, S, O)

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



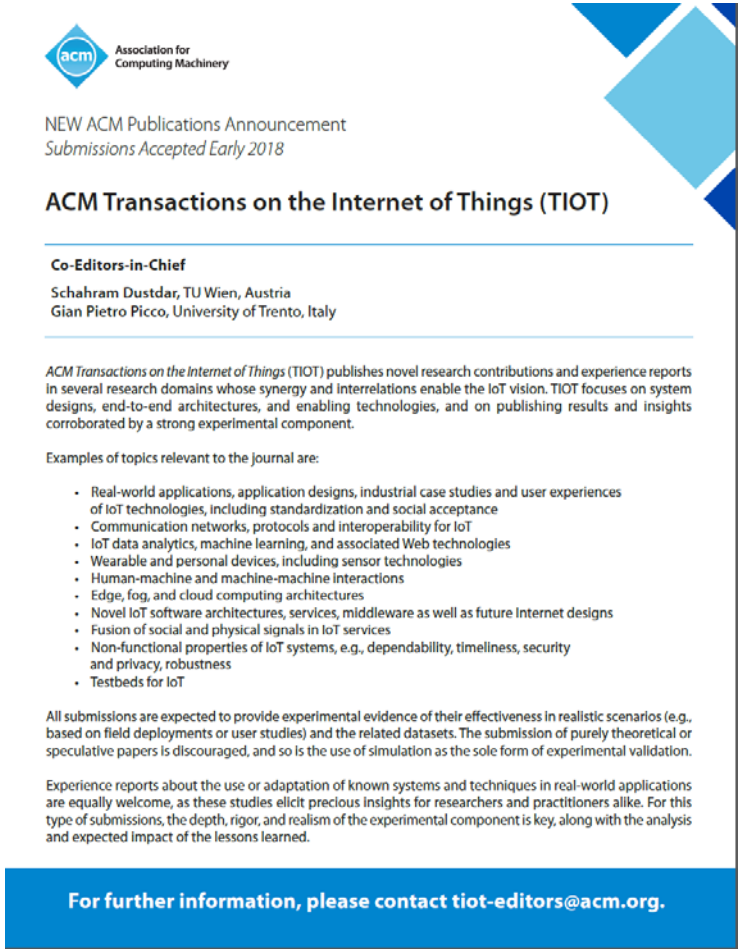
Thanks for your attention

Prof. Schahram Dustdar

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.