Principles and Methods for Elastic Computing -

Lund, 7 May 2014

Schahram Dustdar

Distributed Systems Group
TU Vienna

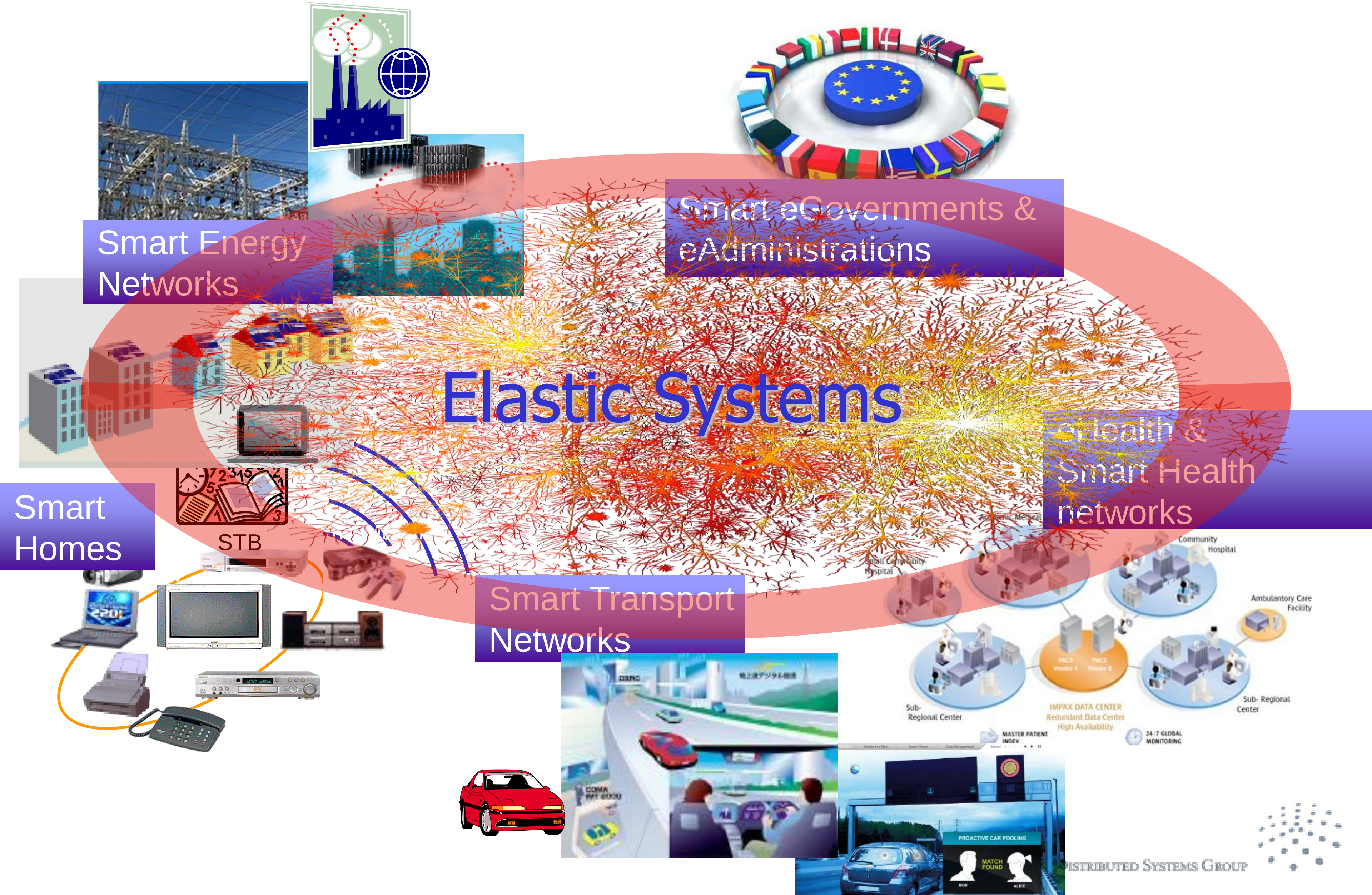
http://dsg.tuwien.ac.at/research/viecom/

Acknowledgements

Includes some joint work with Hong-Linh Truong, Alessio Gambi, Muhammad Z.C. Candra, Georgiana Copil, Duc-Hung Le, Daniel Moldovan, Stefan Nastic, Mirela Riveri, Sanjin Sehic, Ognjen Scekic

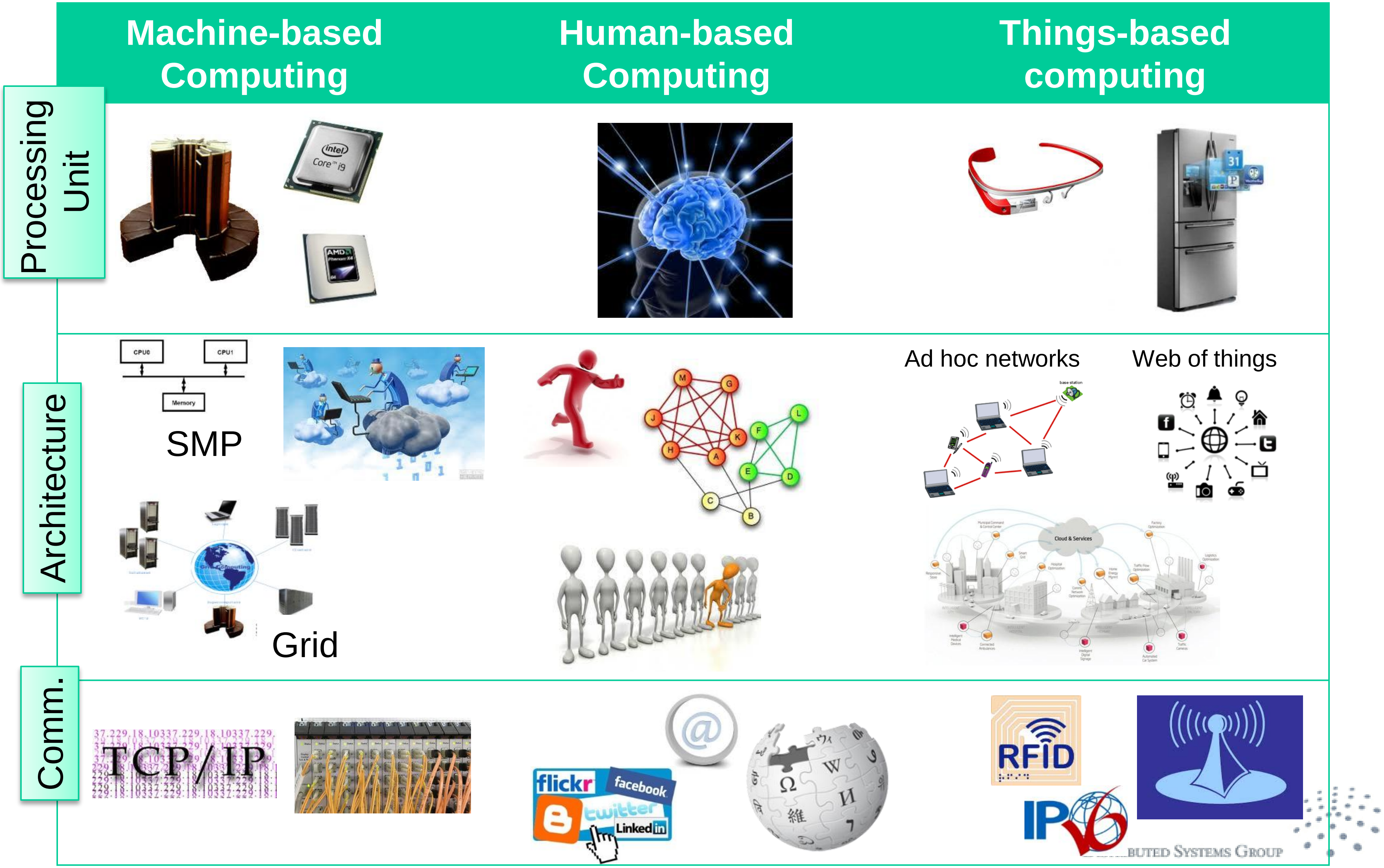
NOTE: The content includes some ongoing work

Smart Evolution – People, Services, Things

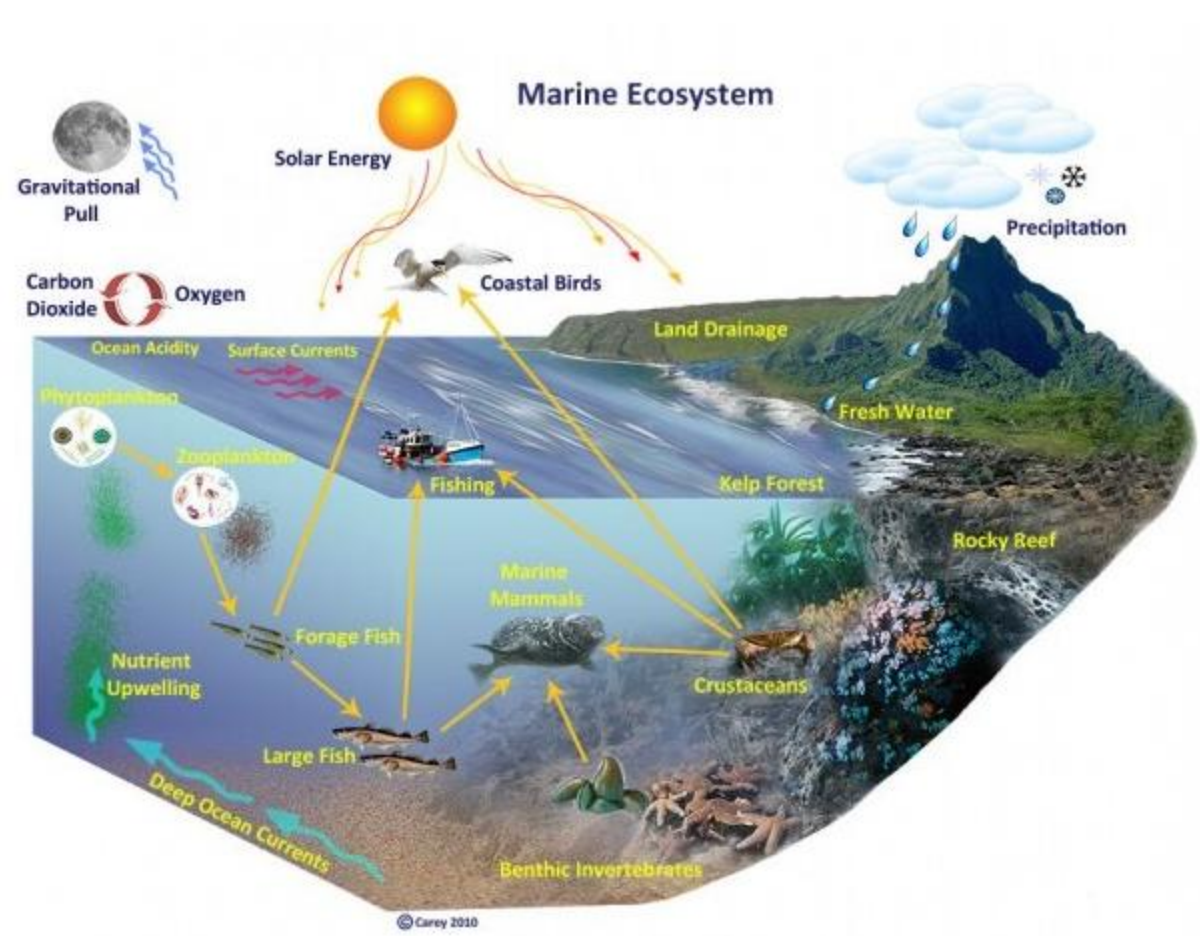


Computing Models

S. Dustdar, H. Truong, “Virtualizing Software and Humans for Elastic Processes in Multiple Clouds – a Service Management Perspective”, in *International Journal of Next Generation Computing*, 2012



Think Ecosystems:
People, Services, Things



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

1. **Robustness mechanisms**: achieving stability in the presence of disruption
2. **Measures of health**: diversity, population trends, other key indicators

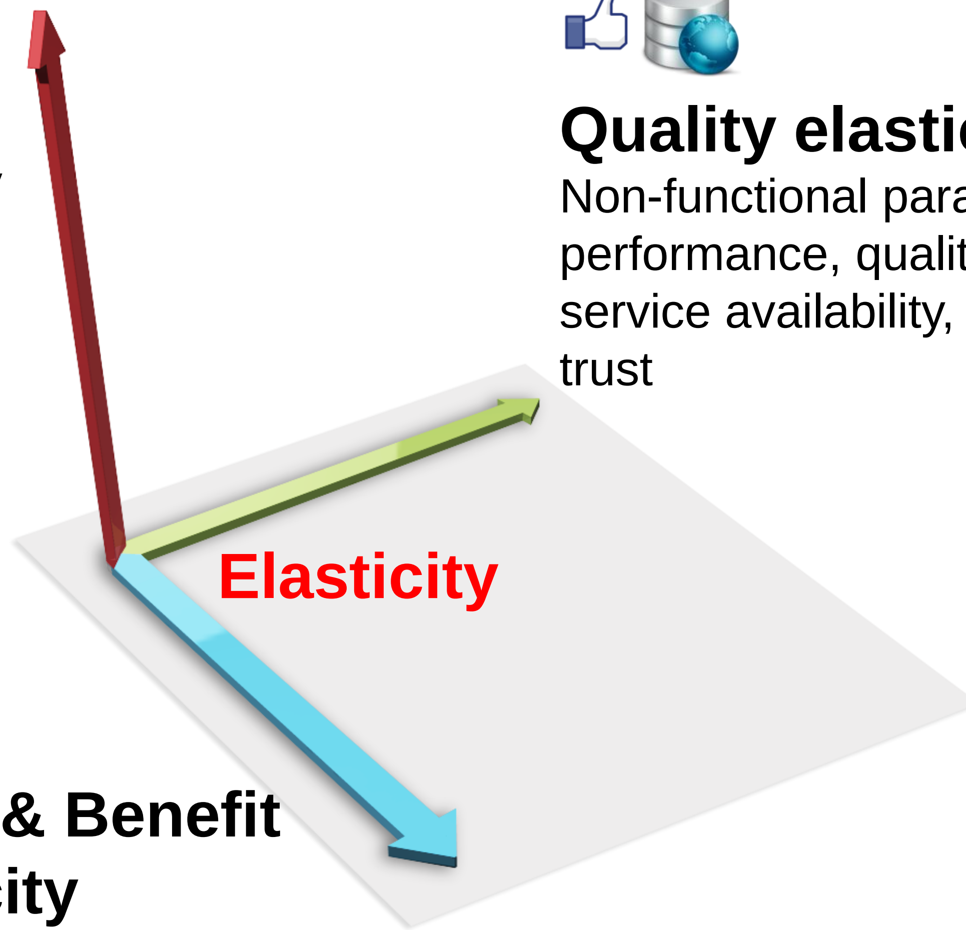
Our Approach

- **Unified service unit model** (Consumption, ownership, provisioning, price, function, etc.)
- **Connecting Data Centers to IoT**
  - From physically isolated verticals to virtual verticals
  - Software-defined elastic data centers and IoT ecosystems
  - SD units are described with well-defined API
  - Provisioning units for customized gateways
  - Dynamically composing units into runtime topologies
  - Runtime controlling and optimization via configuration policies (DevOps principle)
- **Human Augmentation**
  - Human computation capabilities under elastic service units
  - Programming human-based units together with software-based units

Elasticity ≠ Scaleability

Resource elasticity
Software / human-based computing elements, multiple clouds

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

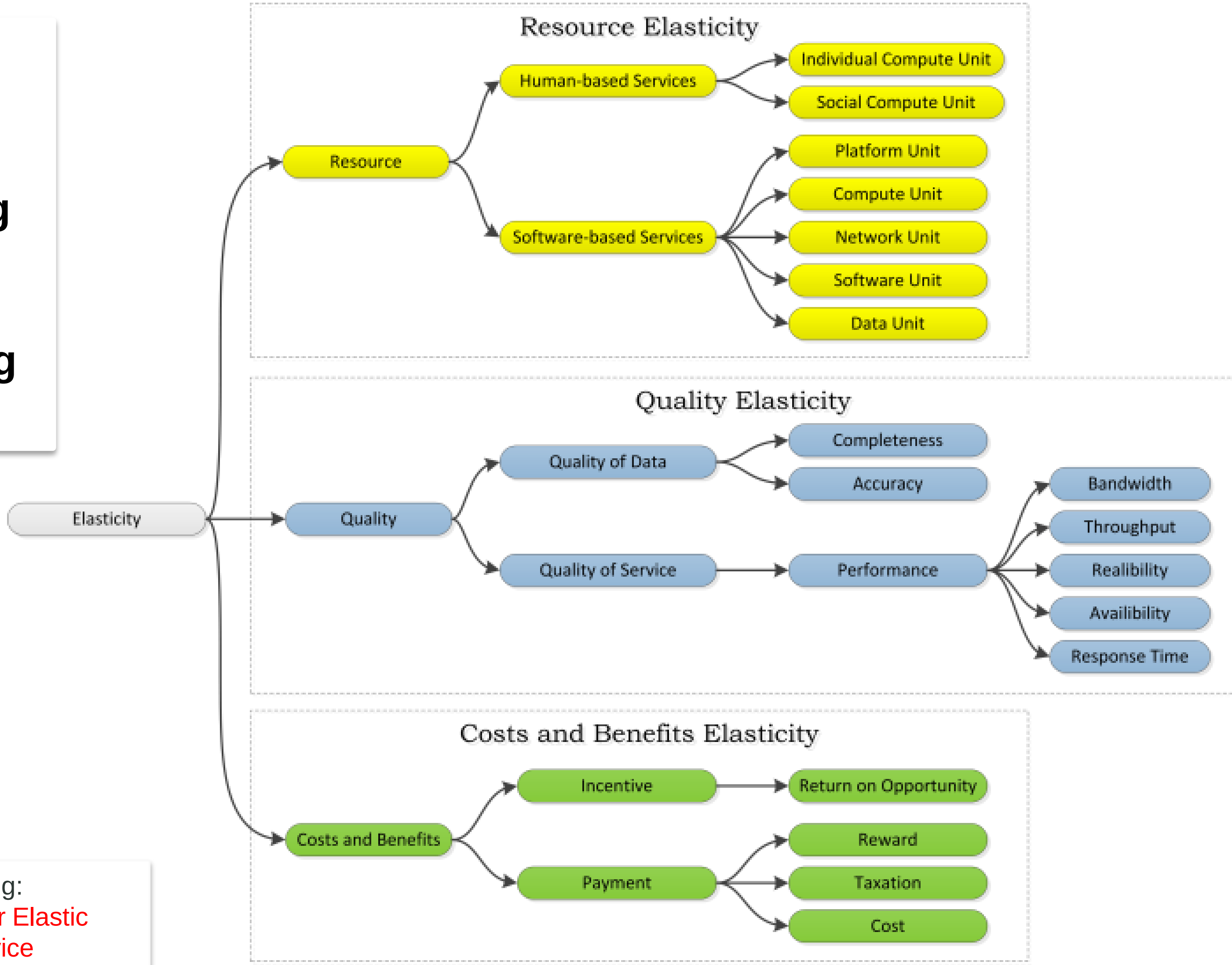


Costs & Benefit elasticity
rewards, incentives

Vienna Elastic Computing Model

dsg.tuwien.ac.at/research/viecom

- **Multi-dimensional Elasticity**
- **Service computing models**
- **Cloud provisioning models**



Schahram Dustdar, Hong Linh Truong: Virtualizing Software and Humans for Elastic Processes in Multiple Clouds- a Service Management Perspective. IJNGC 3(2) (2012)

Diverse types of elasticity requirements

- ***Application user***: “If the cost is greater than 800 Euro, there should be a scale-in action for keeping costs in acceptable limits”
- ***Software provider***: “Response time should be less than amount X varying with the number of users.”
- ***Developer***: “The result from the data analytics algorithm must reach a certain data accuracy under a cost constraint. I don’t care about how many resources should be used for executing this code.”
- ***Cloud provider***: “When availability is higher than 99% for a period of time, and the cost is the same as for availability 80%, the cost should increase with 10%.”

Data Center - Engineering Techniques

High Level Description of Elasticity Requirements

SYBL (Simple Yet Beautiful Language) for specifying elasticity requirements

SYBL-supported requirement levels

1. Cloud Service Level
2. Service Topology Level
3. Service Unit Level
4. Relationship Level
5. Programming/Code Level

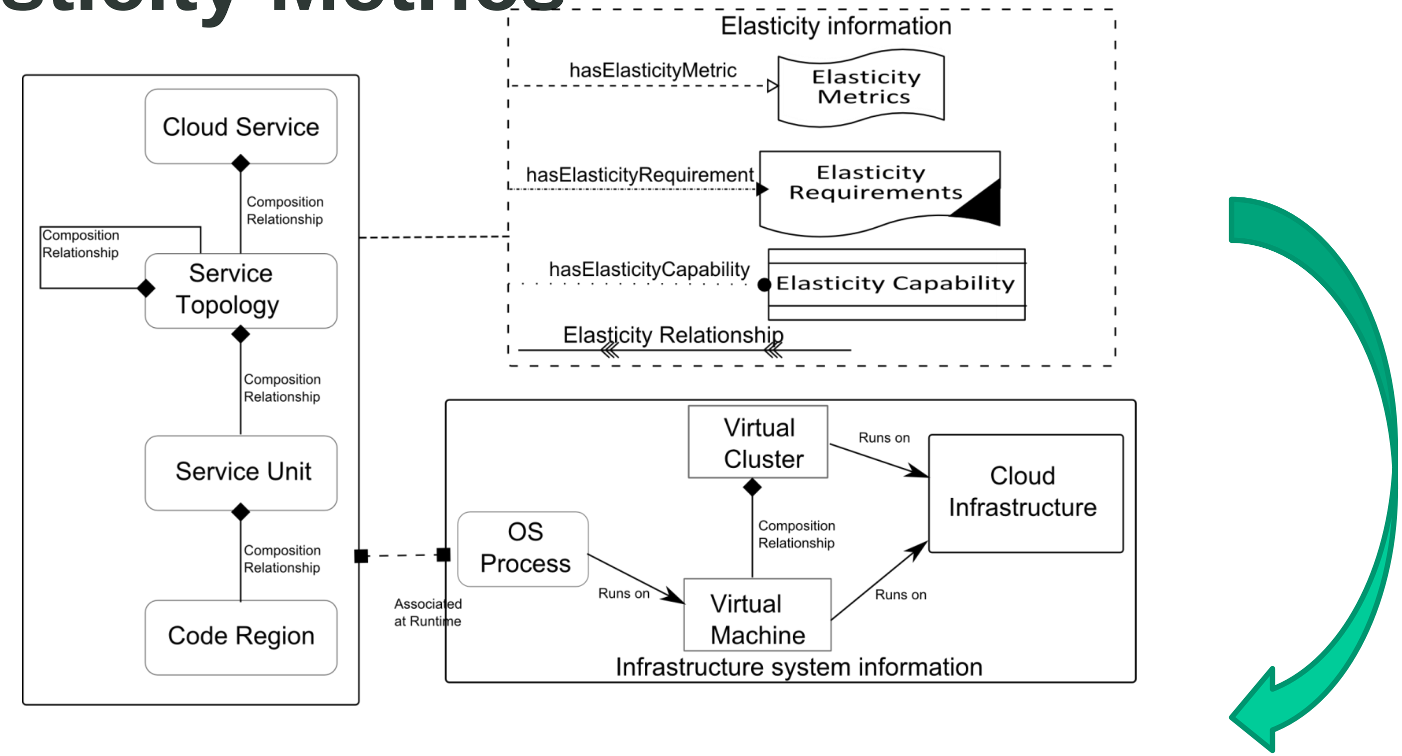
#SYBL.CloudServiceLevel
Cons1: CONSTRAINT responseTime < 5 ms
Cons2: CONSTRAINT responseTime < 10 ms WHEN nbOfUsers > 10000
Str1: STRATEGY CASE fulfilled(Cons1) OR fulfilled(Cons2): minimize(cost)

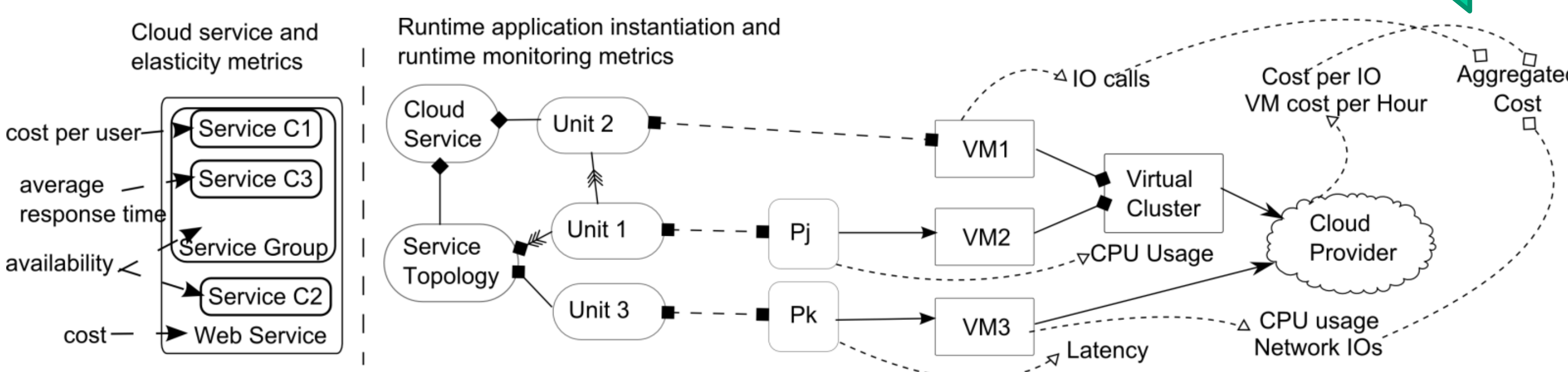
#SYBL.ServiceUnitLevel
Str2: STRATEGY CASE ioCost < 3 Euro : maximize(dataFreshness)

#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

Mapping Services Structures to Elasticity Metrics

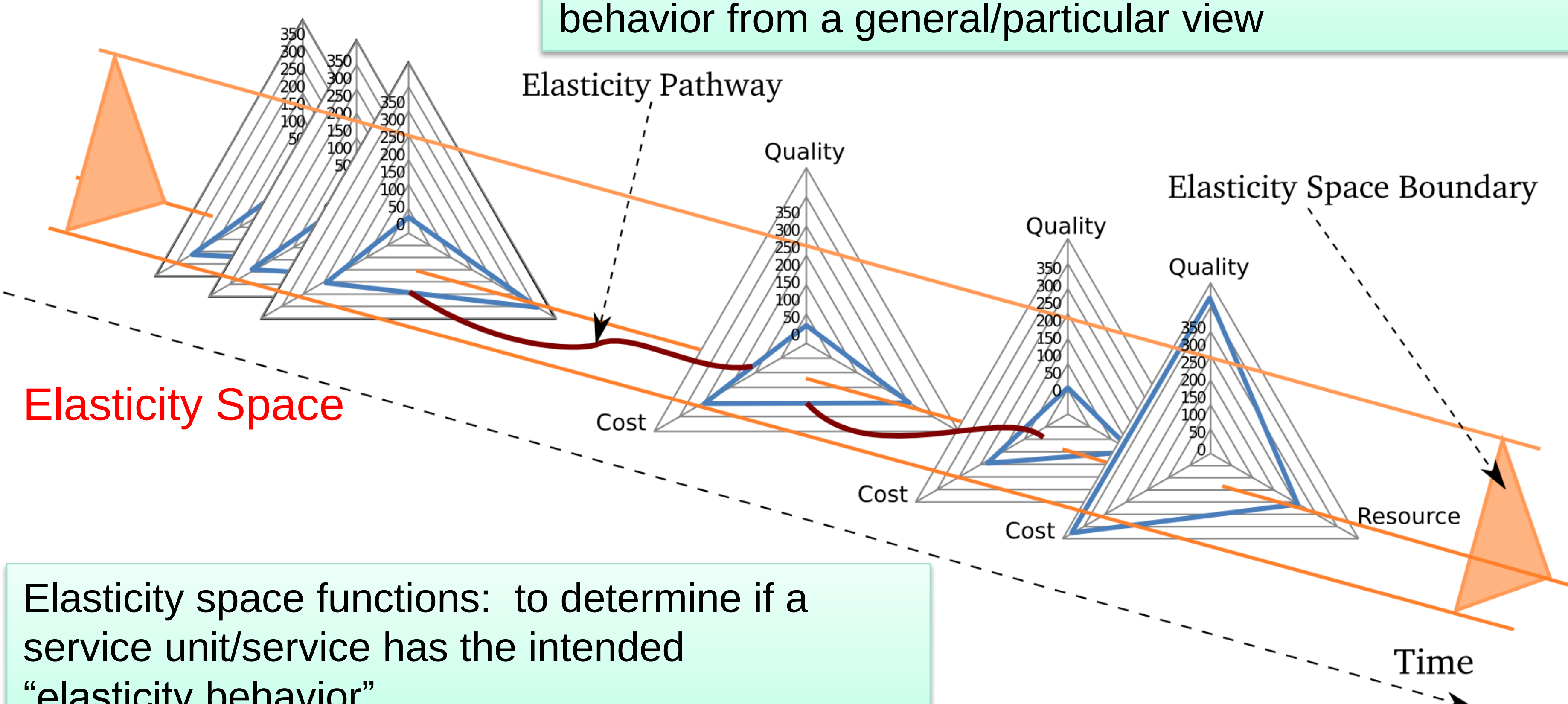




Elasticity Model for Cloud Services

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 has the intended "elasticity behavior"

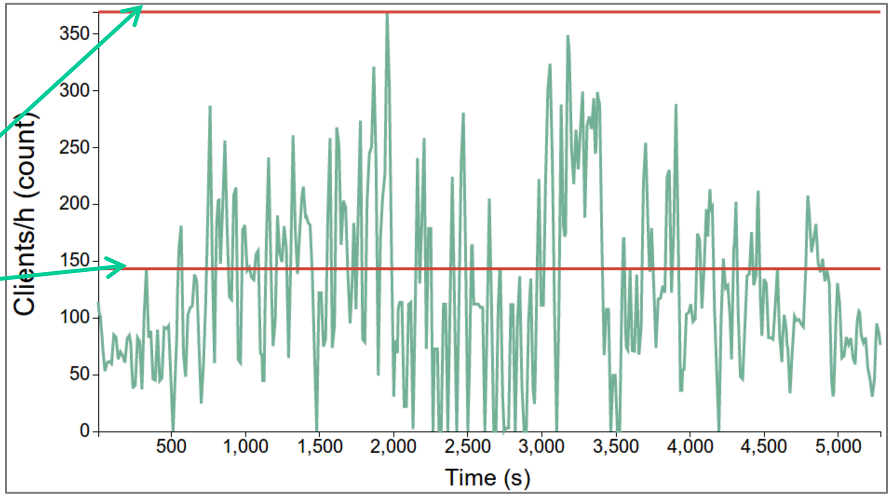
Multi-Level Elasticity Space

Service requirement

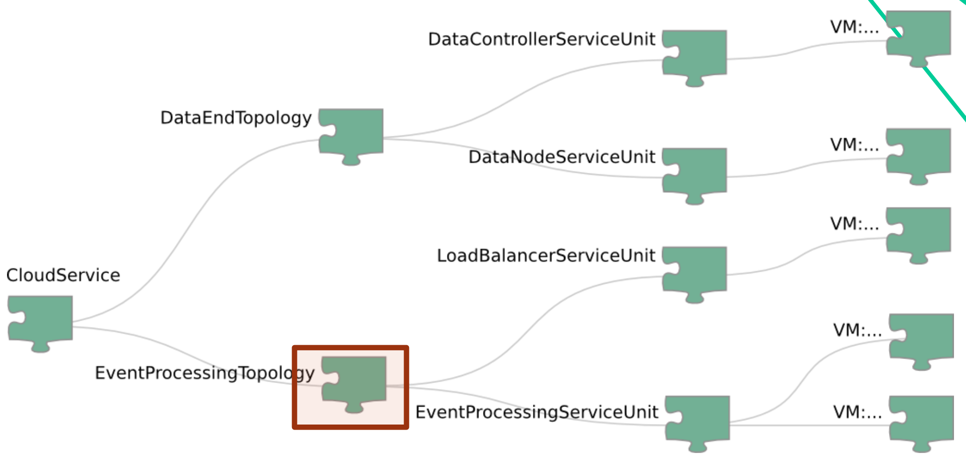
COST<= 0.0034$/client/h

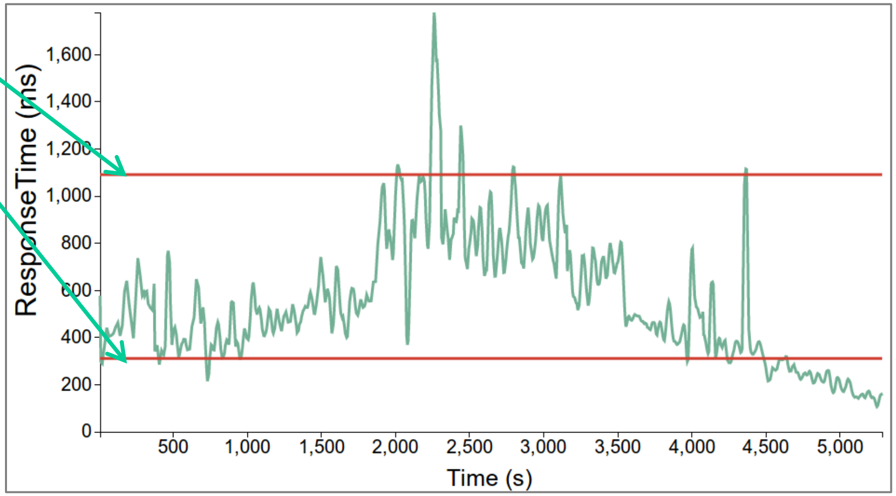
2.5$ monthly subscription for each service client (sensor)

- Determined Elasticity Space Boundaries
  - Clients/h > 148
  - 300ms ≤ ResponseTime ≤ 1100 ms



Elasticity Space "Clients/h" Dimension



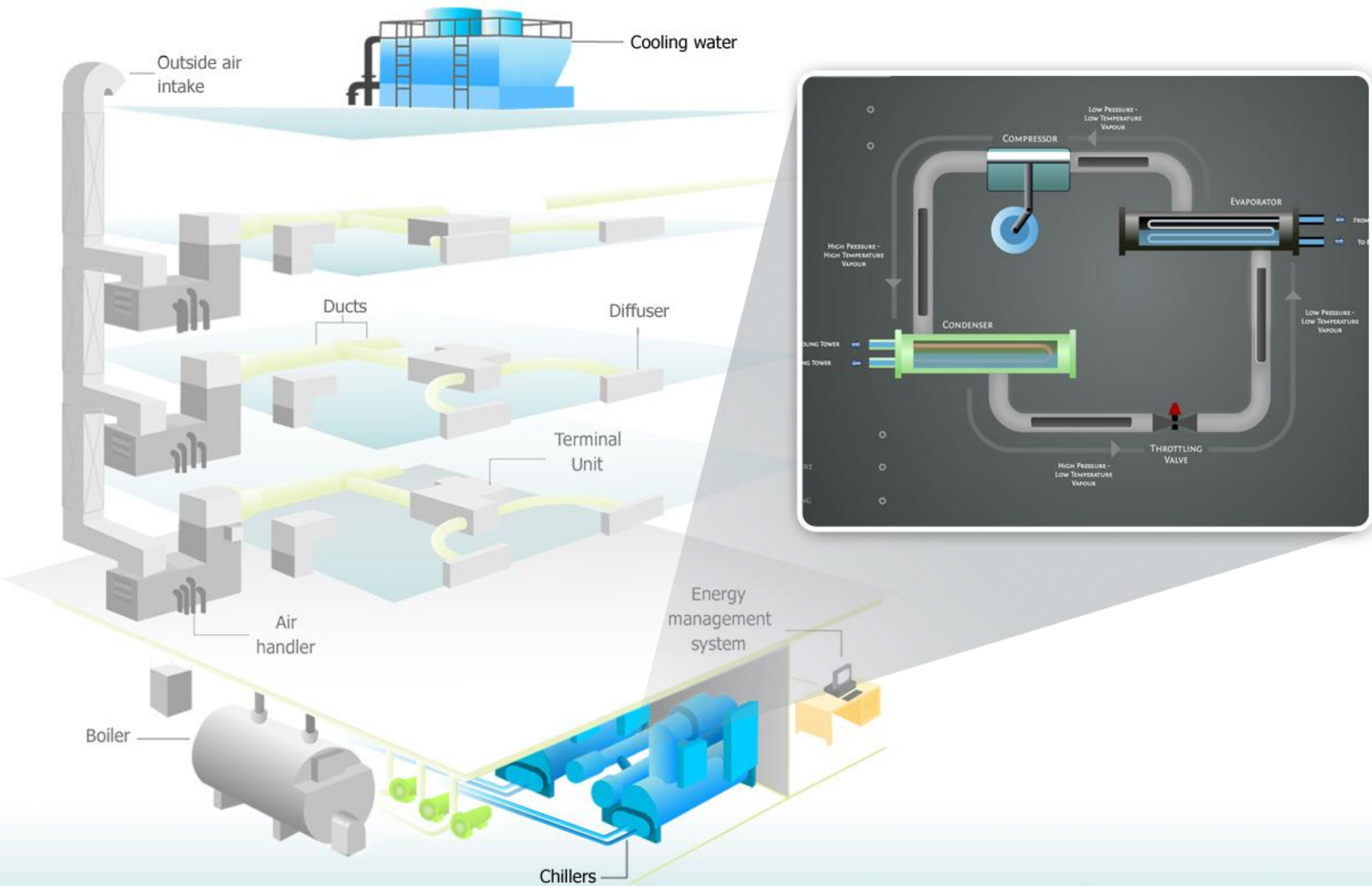


Elasticity Space "Response Time" Dimension

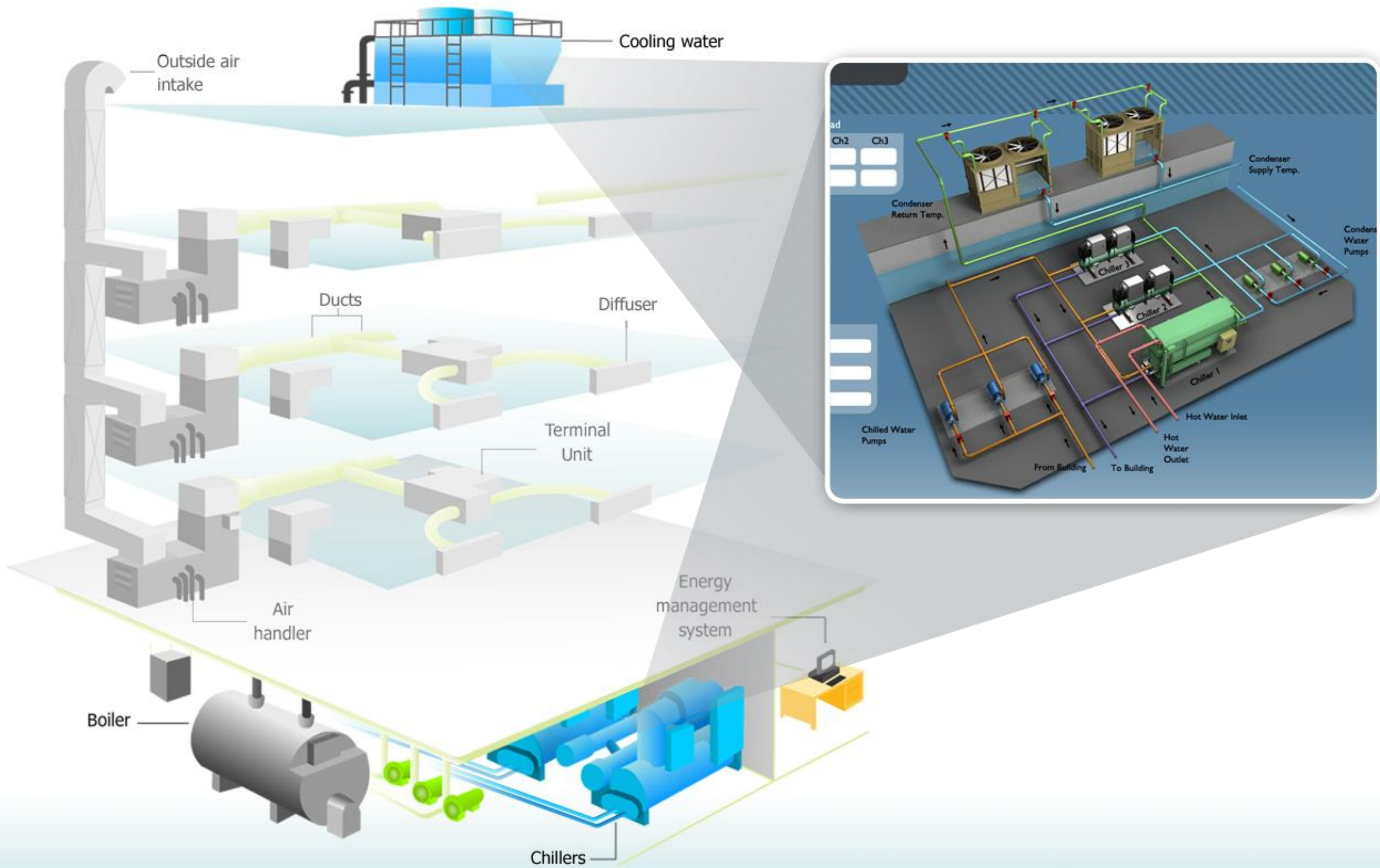
IoT- Engineering Techniques


Smart City Dubai
Pacific Controls
Command Control Center
SMEs
Dashboards
User interfaces
Report
Reports
CO2
Carbon footprint measurement
Benchmarking
Remote monitoring
Engineers
Villas
Fire
Safety & security
Energy
HVAC
CCTV
Carbon footprint
Factories
Fire
Lift
Safety & security
Energy
Chiller / HVAC
Boiler
CCTV
Carbon footprint
Schools
Fire
Safety & security
Energy
Chiller / HVAC
CCTV
Carbon footprint
Commercial & residential buildings
Fire
Lift
Safety & security
Energy
Chiller / HVAC
Boiler
CCTV
Carbon footprint
Utilities
Sewage pumps
Water treatment plants
Irrigation
Hospitals
Fire
Lift
Safety & security
Energy
Chiller / HVAC
Boiler
CCTV
Carbon footprint
© Copyright 2010 Pacific Control Systems. All Rights Reserved.

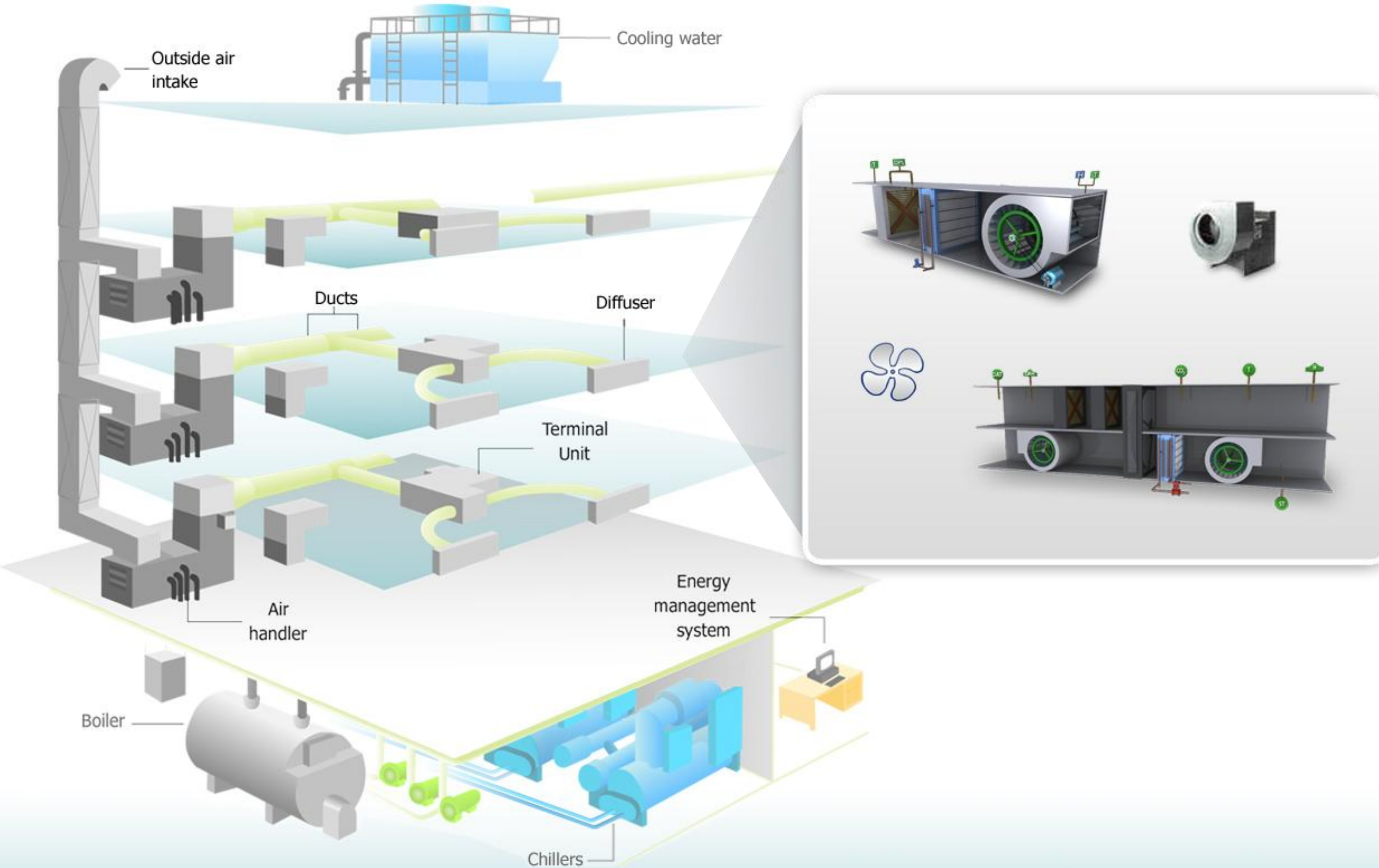
HVAC (Heating, Ventilation, Air Conditioning) Ecosystem



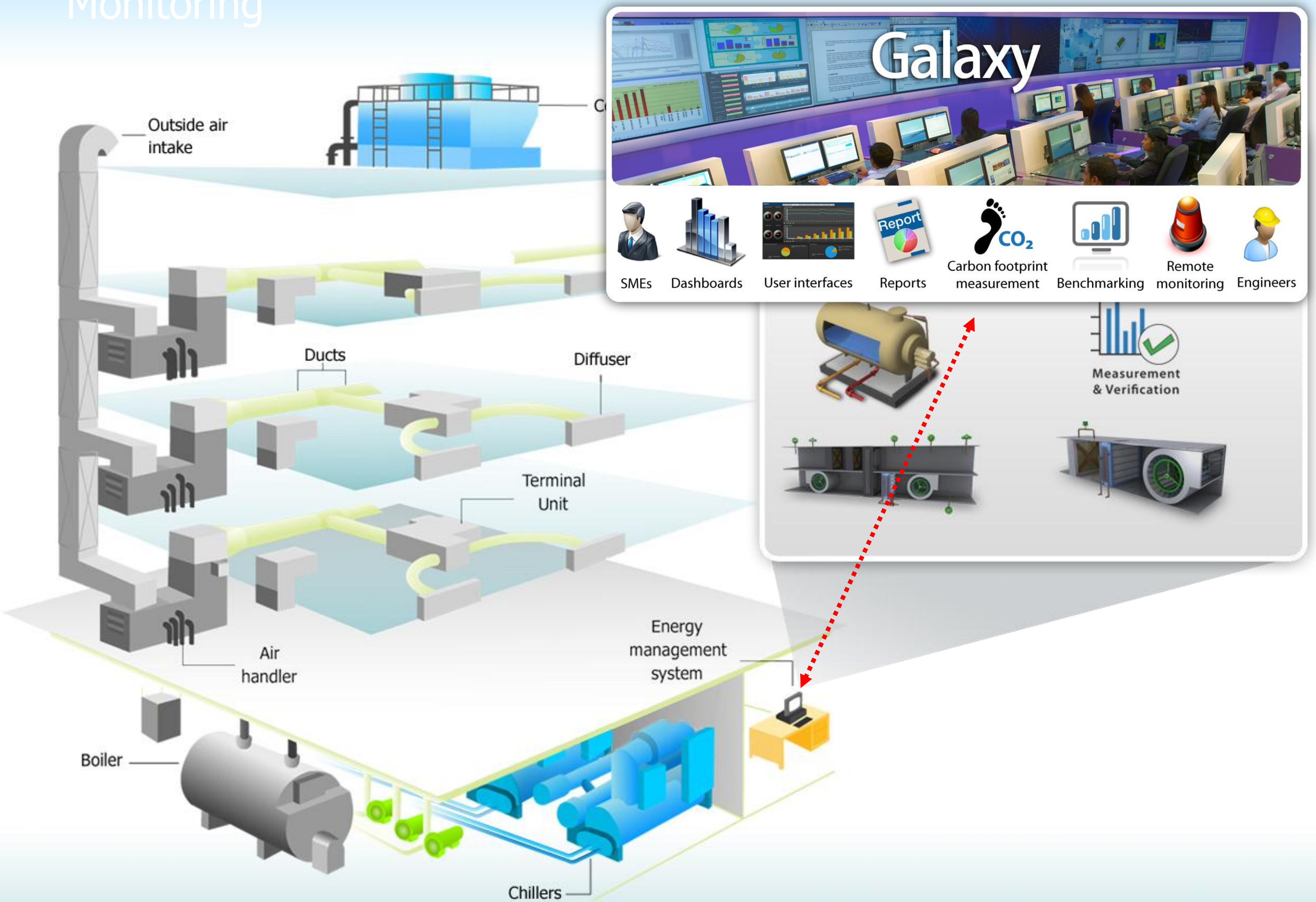
Water Ecosystem



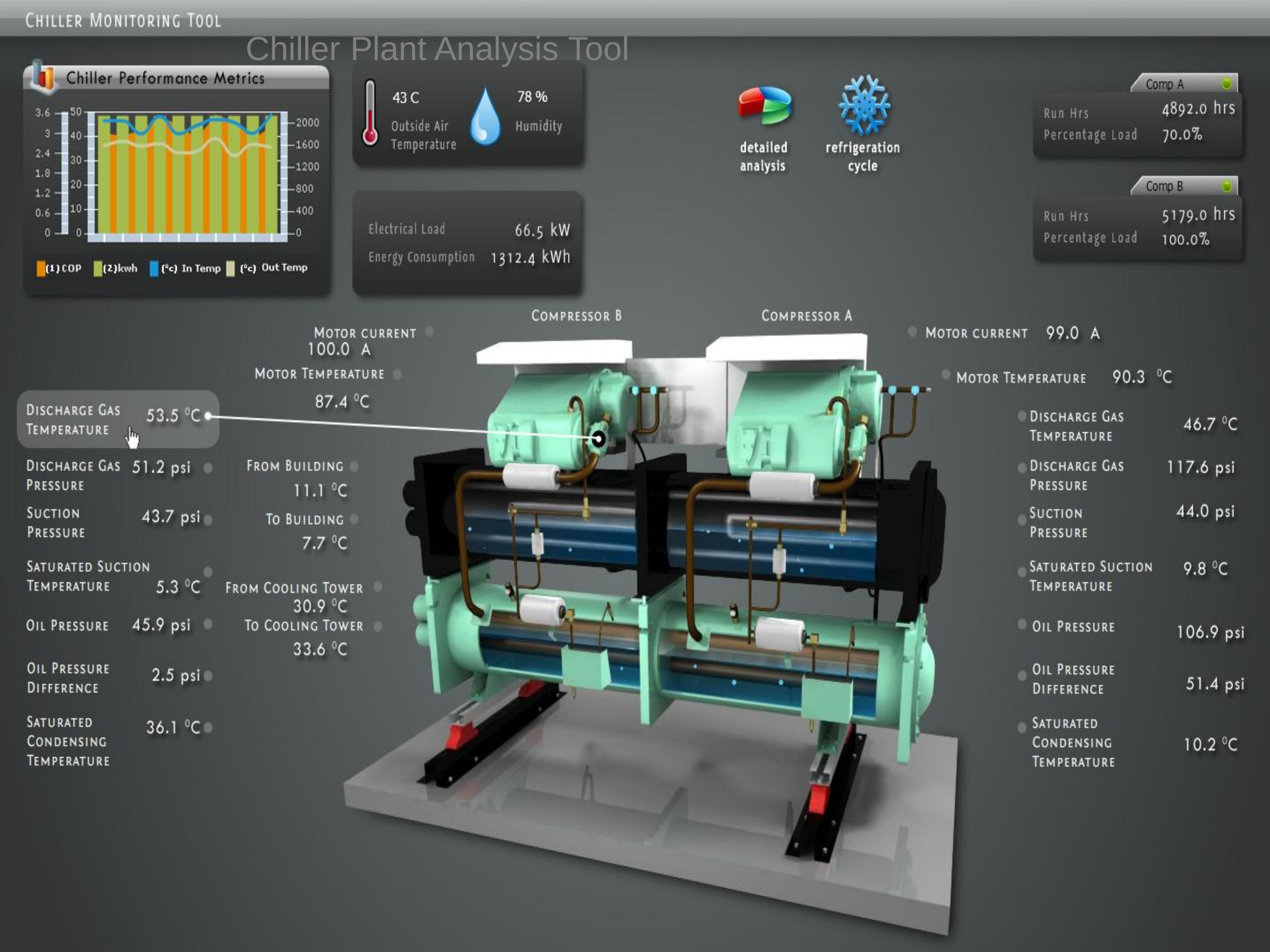
Air Ecosystem



Monitoring

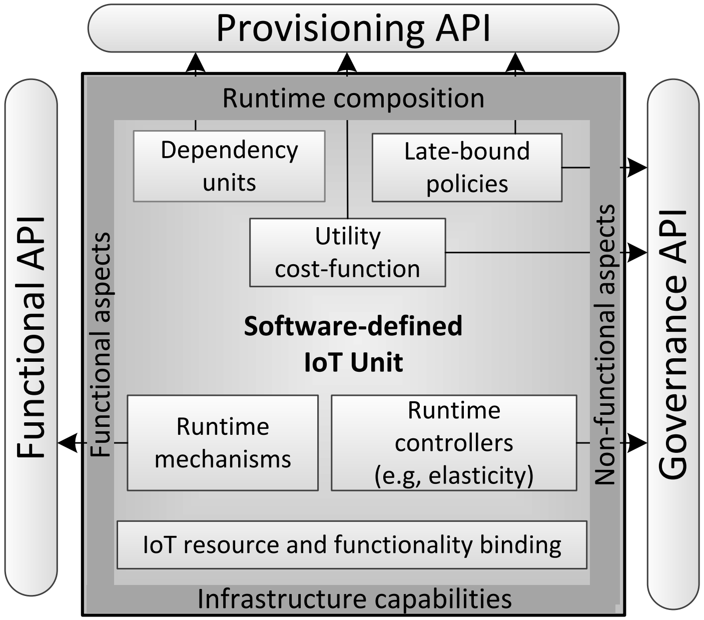

Galaxy
SMEs
Dashboards
User interfaces
Reports
Carbon footprint measurement
Benchmarking
Remote monitoring
Engineers
Measurement & Verification
Outside air intake
Ducts
Diffuser
Terminal Unit
Air handler
Energy management system
Boiler
Chillers

Chiller Plant Analysis Tool



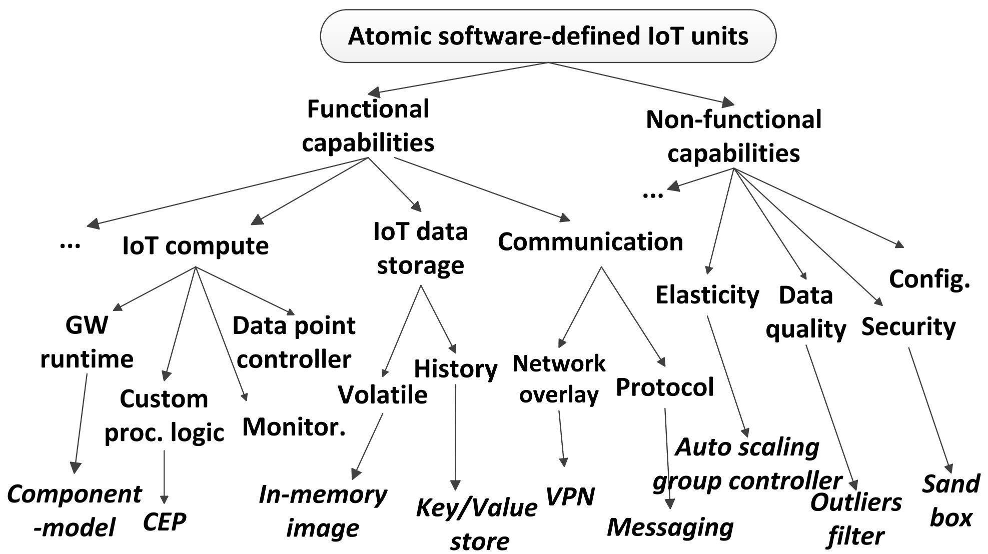
Software-defined IoT units

- Provide software-defined API for accessing, configuring and controlling units
- Support fine-grained internal configurations, e.g, adding functional capabilities like different communication protocols, at runtime.
- Can be composed at higher-level, via dependency units, creating virtual topologies (of multiple gateways) that can be (re)configured at runtime.
- Enable decoupled and managed configuration (via late-bound policies) to provision the units dynamically and on-demand.
- Have utility cost-functions that enable pricing the IoT resources as utilities.

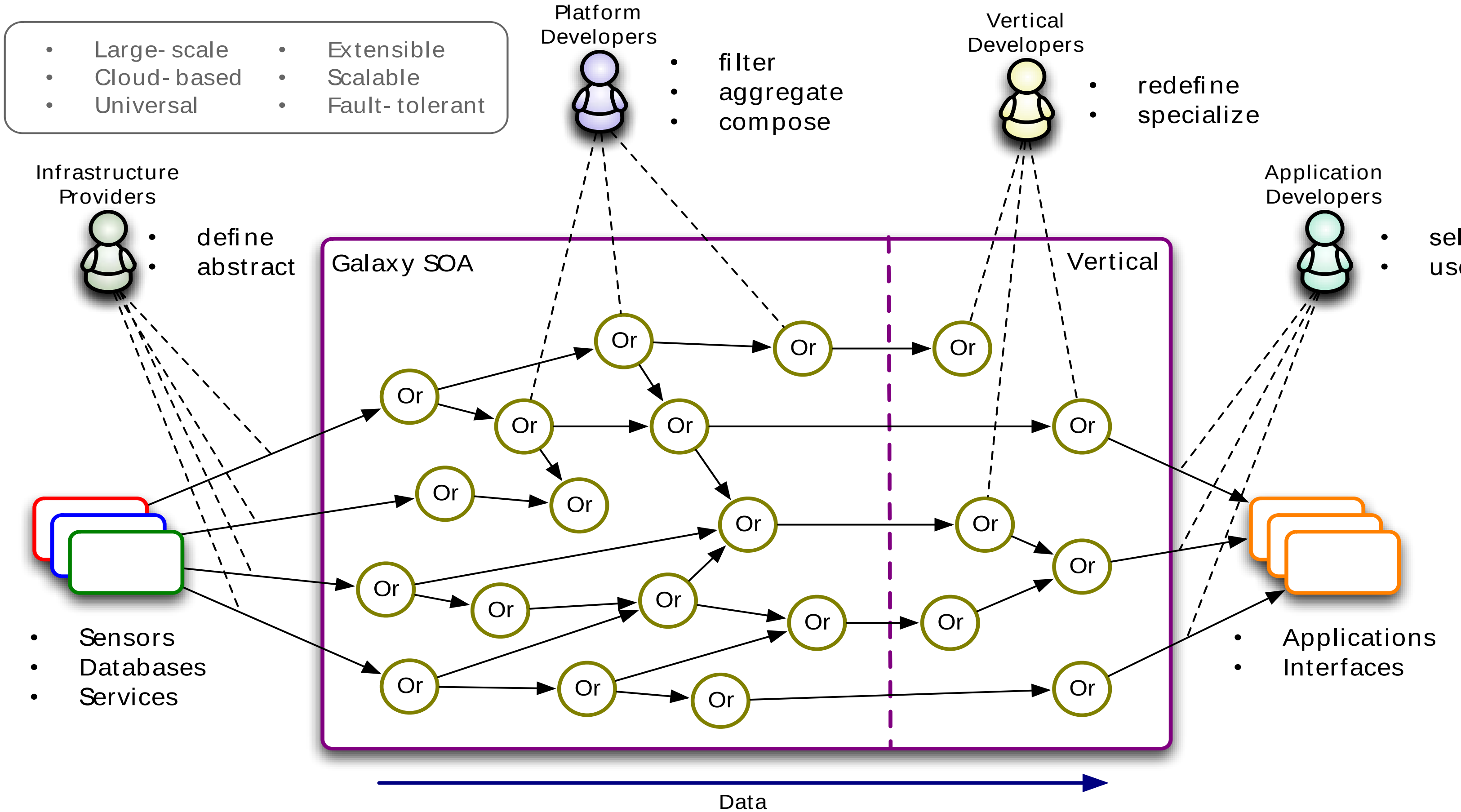


Ecosystem for software-defined IoT systems

- Create an ecosystem of software-defined IoT units for the creation of software-defined IoT systems.
- Distributing IoT units in a market-like fashion, e.g., via IoT AppStore.



The Programming Model (Origins)

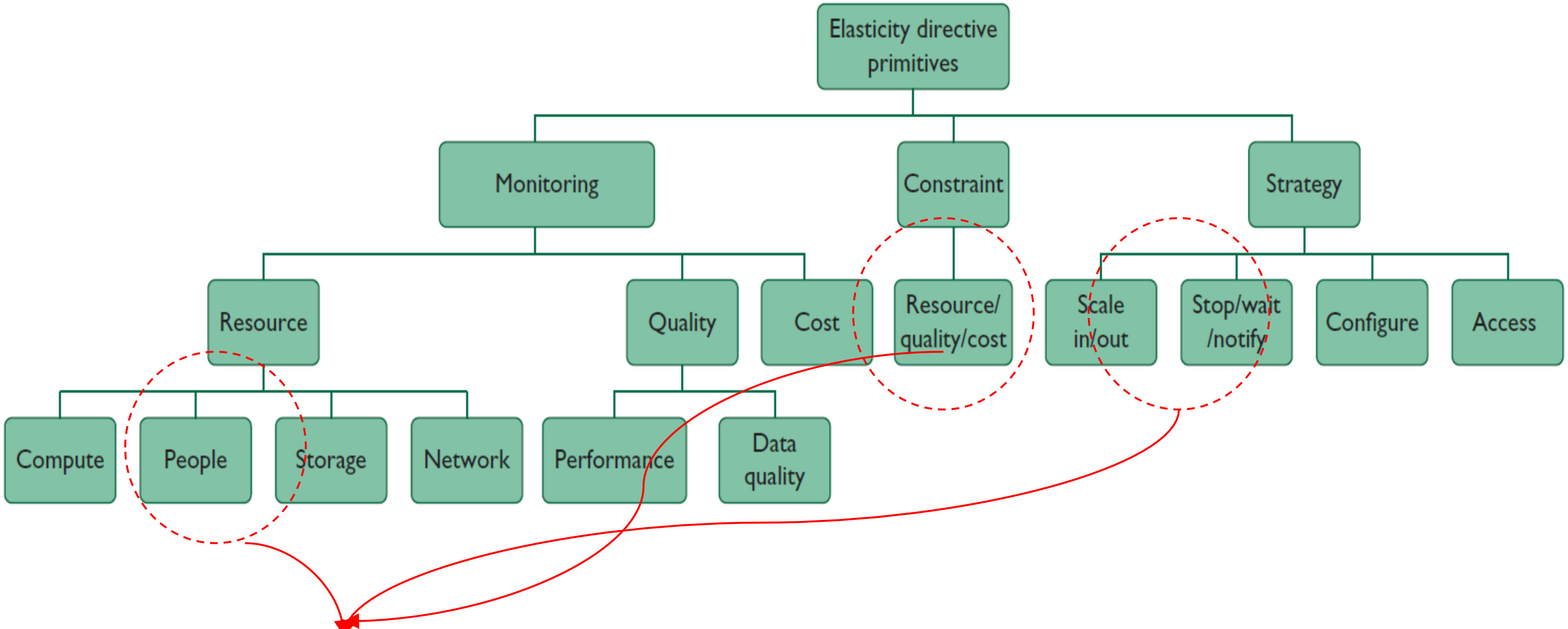


Sehic, S. Li, F., Nastic, S., Dustdar, S.. A Programming Model for Context-Aware Applications in Large-Scale Pervasive Systems, The 8th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2012), 8-10 October 2012, Barcelona, Spain.

Sehic, S., Nastic, S., Vögler, M., Li, F., Dustdar, S. Entity-Adaptation - A Programming Model for Development of Context-Aware Applications, 29th Symposium On Applied Computing (SAC 2014), Mobile Computing and Applications (MCA) track, 24-28 March 2014, Gyeongju, Republic of Korea

Human augmentation – Engineering Techniques

Specifying and controling elasticity of human-based services

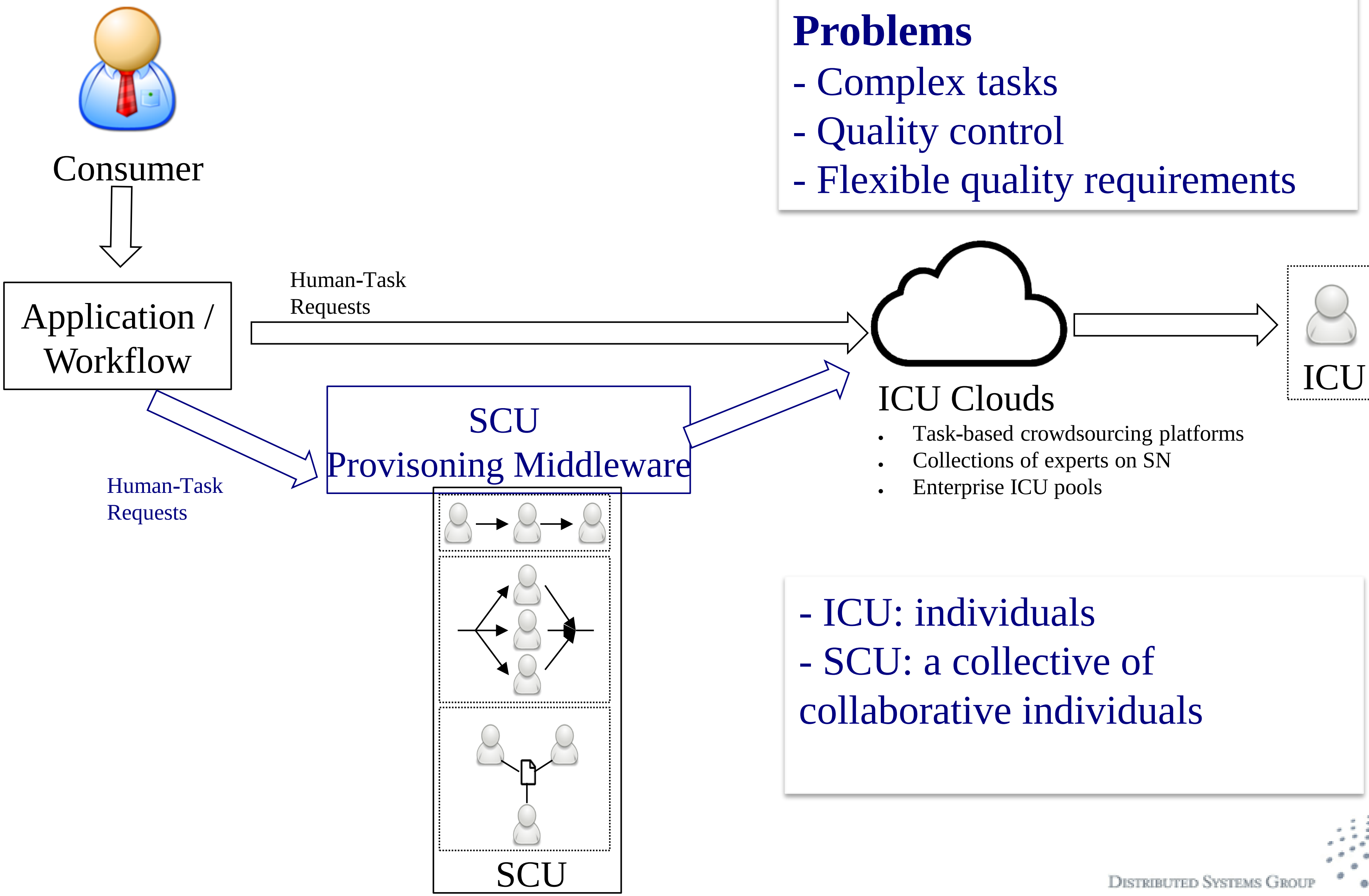


What if we need to invoke a human?

#for a service unit 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)

SCU for independent tasks

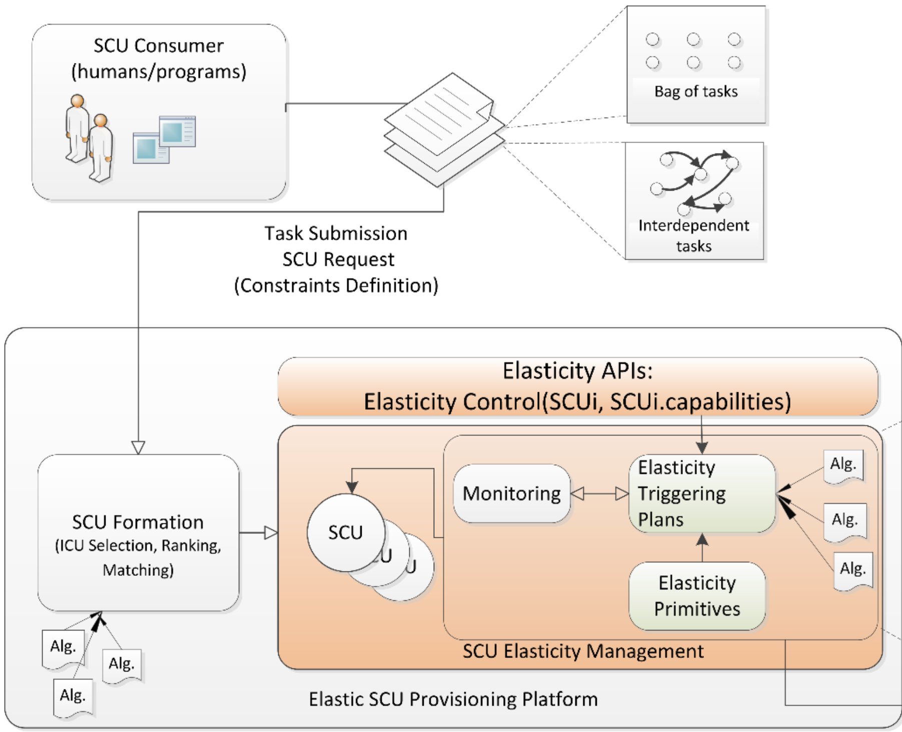
Problems
- Complex tasks
- Quality control
- Flexible quality requirements

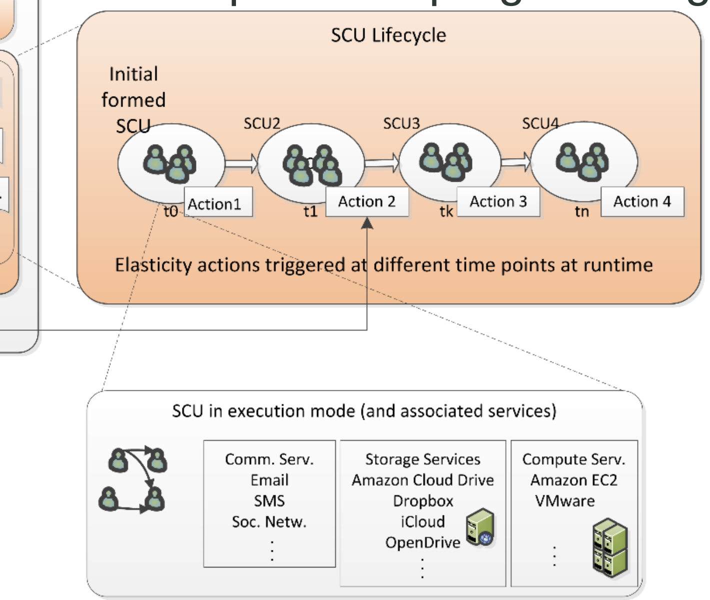


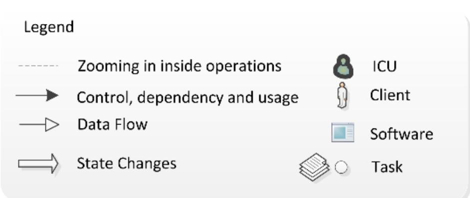
- ICU: individuals
- SCU: a collective of collaborative individuals

Elasticity Capabilities and APIs

- SCU execution model and lifecycle management
- Software-defined APIs allow controlling the elasticity of SCU
- Algorithms to deal with elasticity capabilities programming







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

Conclusions (1) – Engineering Elasticity

- The evolution of underlying systems and the utilization of different types of resources under different models for elasticity requires
  - Complex, open hybrid service unit provisioning frameworks
  - Different strategies for dealing with different types of tasks
  - Quality issues for software, data, and people in an integrated manner for different perspectives

- We are just at an early stage of developing techniques for engineering elastic applications wrt multi-dimensional elasticity

Conclusions (2) – Engineering Elasticity

Service engineering analytics of elastic systems

- Programming hybrid compute units for elastic processes
- Elasticity specifications and reasoning techniques
- Elasticity spaces analytics

Application domains

- “Social computer“ and smart cities (FP 7 FET Smart Cities and PC3L)
- Computational science and engineering (FP 7 CELAR)

Thanks for your attention!

Prof. Dr. Schahram Dustdar

Distributed Systems Group
TU Wien

dsg.tuwien.ac.at