Towards building Cyber-Physical ecosystems of People, Processes, and Things

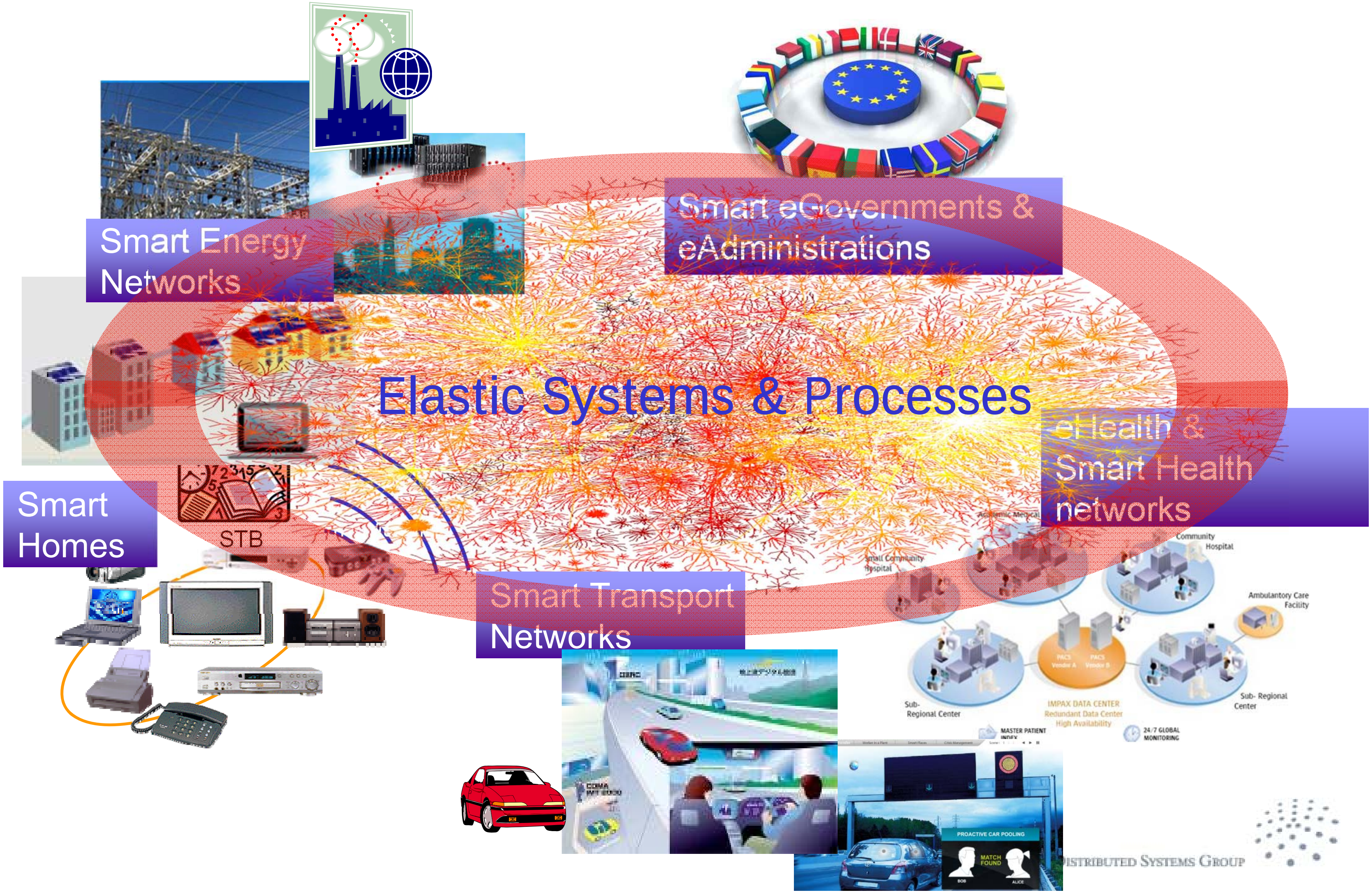
Schahram Dustdar

Distributed Systems Group
TU Wien

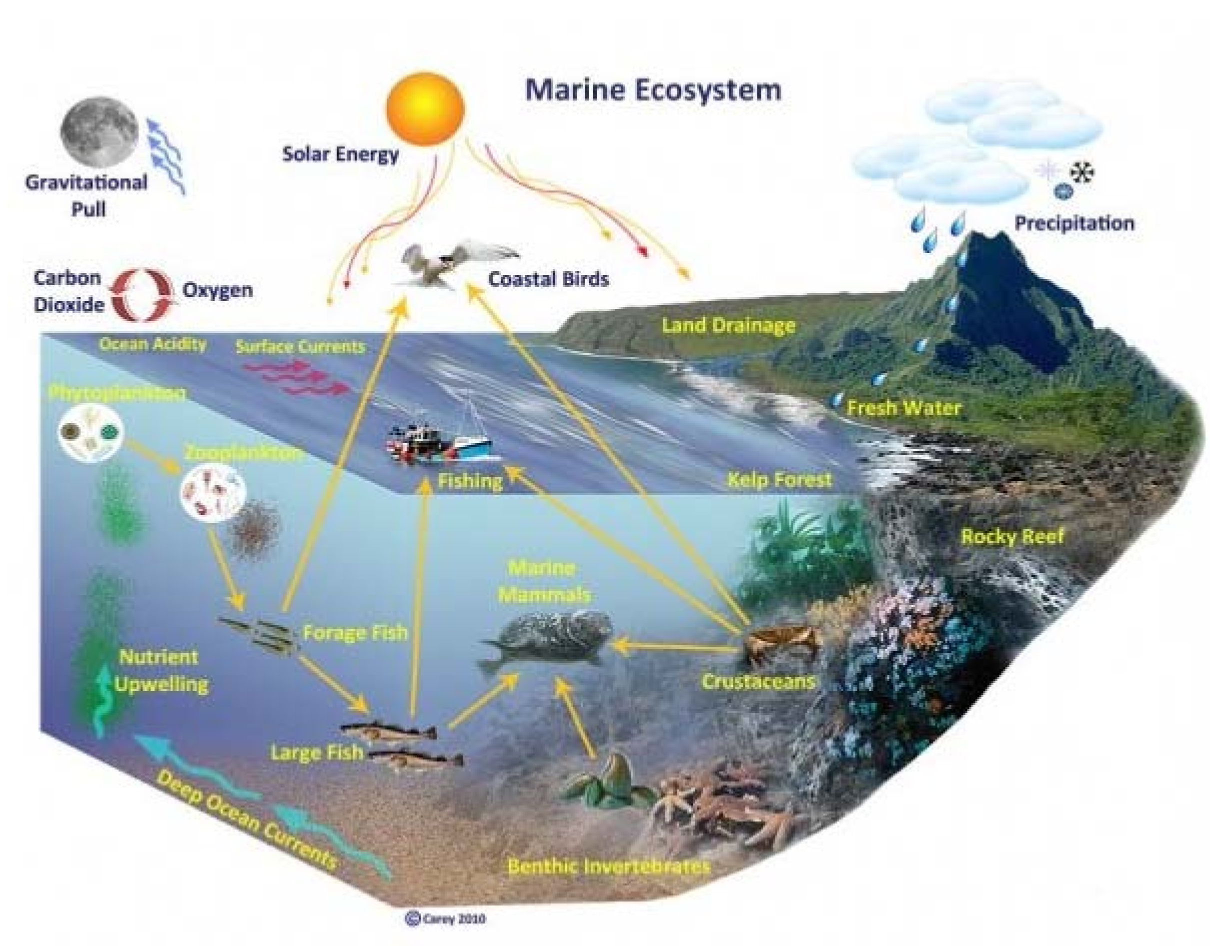
dsg.tuwien.ac.at

Distributed Systems Group

Smart Evolution – People, Services, Things



Think Ecosystems: People, Services/Processes, Things



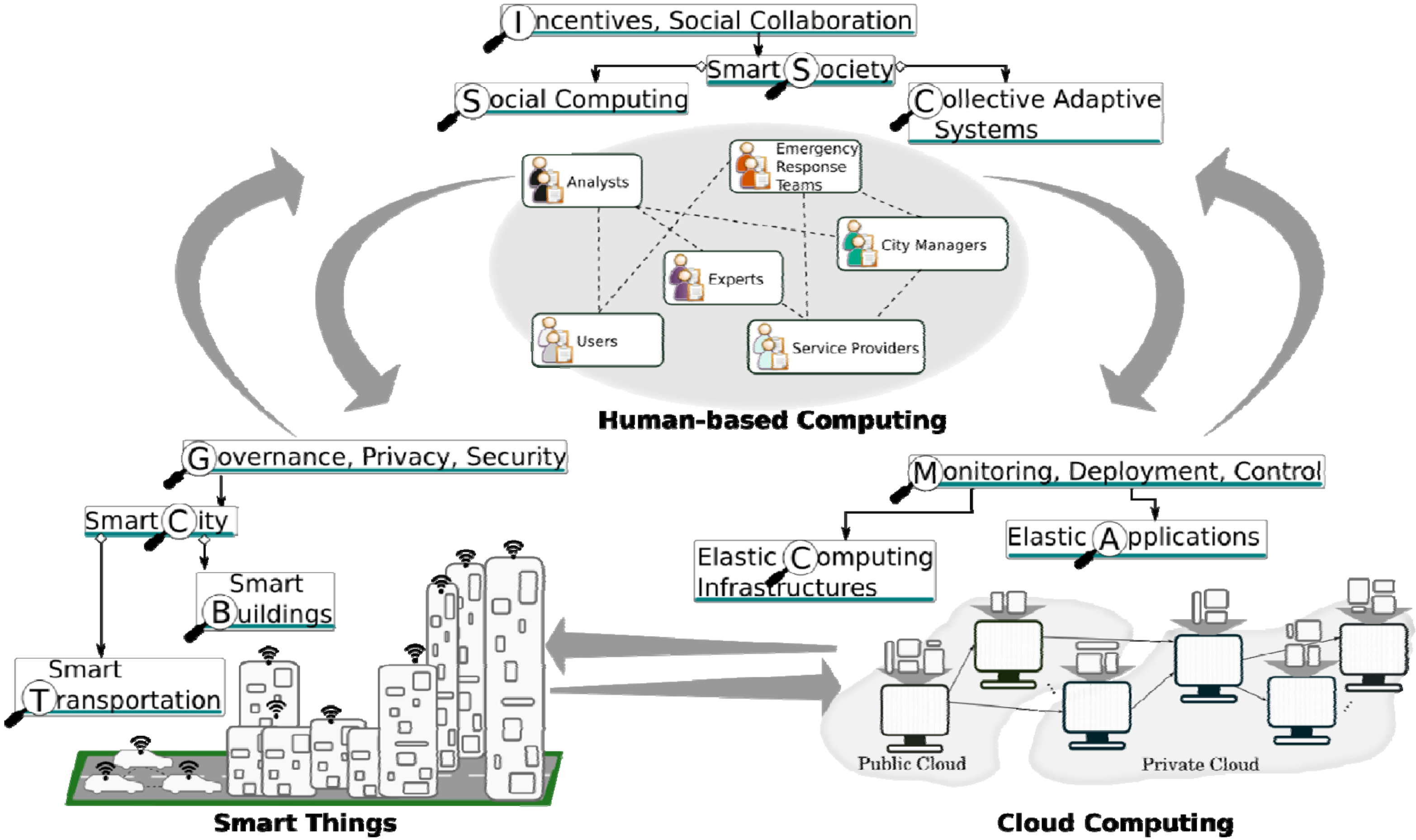
Marine Ecosystem: http://www.xbordercurrents.co.uk/wildlife/marine-ecosystem-2

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

Connecting People, Processes, and Things



Elasticity: Enabling properties

1957: Time Sharing
1961: Utility Computing
1968: Component-based Systems
1978: Client-Server Architecture
1990s: SOA, Grid Computing
2006: Cloud Computing

Hardware and Software Reusability Development Timeline

Elasticity: Enabling properties

- Replaceable and reusable components
- On-demand computing
- Distributed computing
- Self-contained units of functionality

Elasticity: Enabling properties

1951: Socio-technical Systems

1986: Interactive Evolutionary Computation

2005: Crowdsourcing

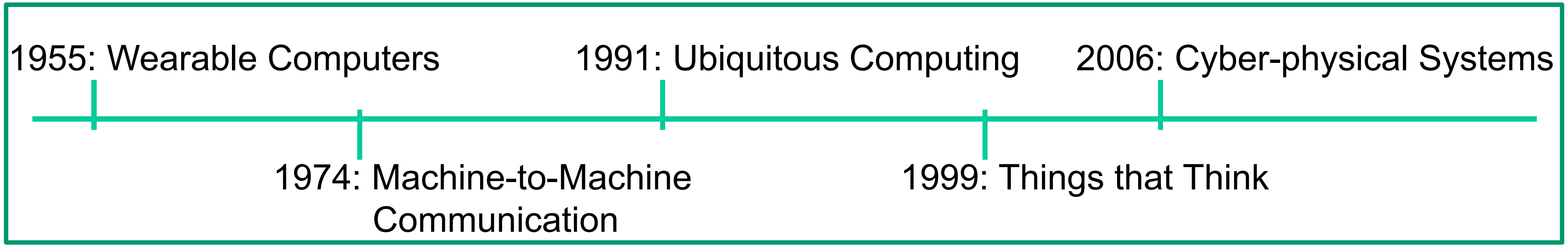
2011: Social Compute Unit

Human Computing Development Timeline

Elasticity: Enabling properties

- Replaceable and reusable components
- On-demand computing
- Distributed computing
- Self-contained units of functionality
- Human-software composition

Elasticity: Enabling properties



Smart Things Development Timeline

Elasticity: Enabling properties

- Replaceable and reusable components
- On-demand computing
- Distributed computing
- Self-contained units of functionality
- Human-software composition
- Interoperability
- Span cyber and physical worlds

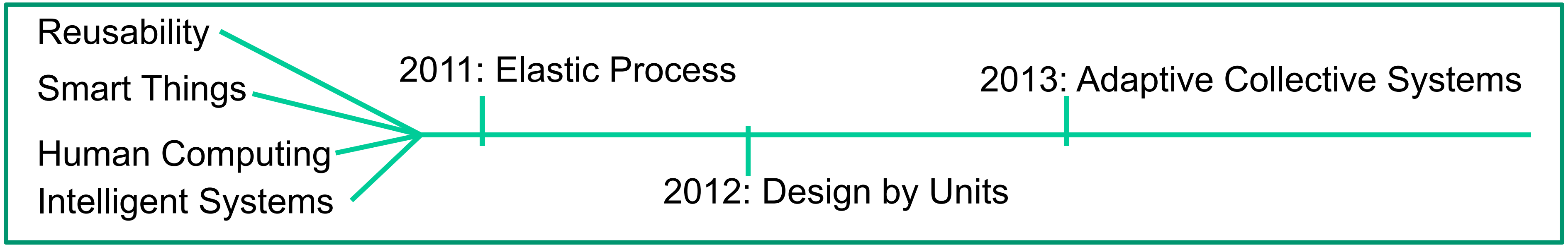
Elasticity: Enabling properties

Intelligent and Autonomic Systems Development Timeline

Elasticity: Enabling properties

- Replaceable and reusable components
- On-demand computing
- Distributed computing
- Self-contained units of functionality
- Human-software composition
- Interoperability
- Span cyber and physical worlds
- Adaptation

Elasticity: Enabling properties

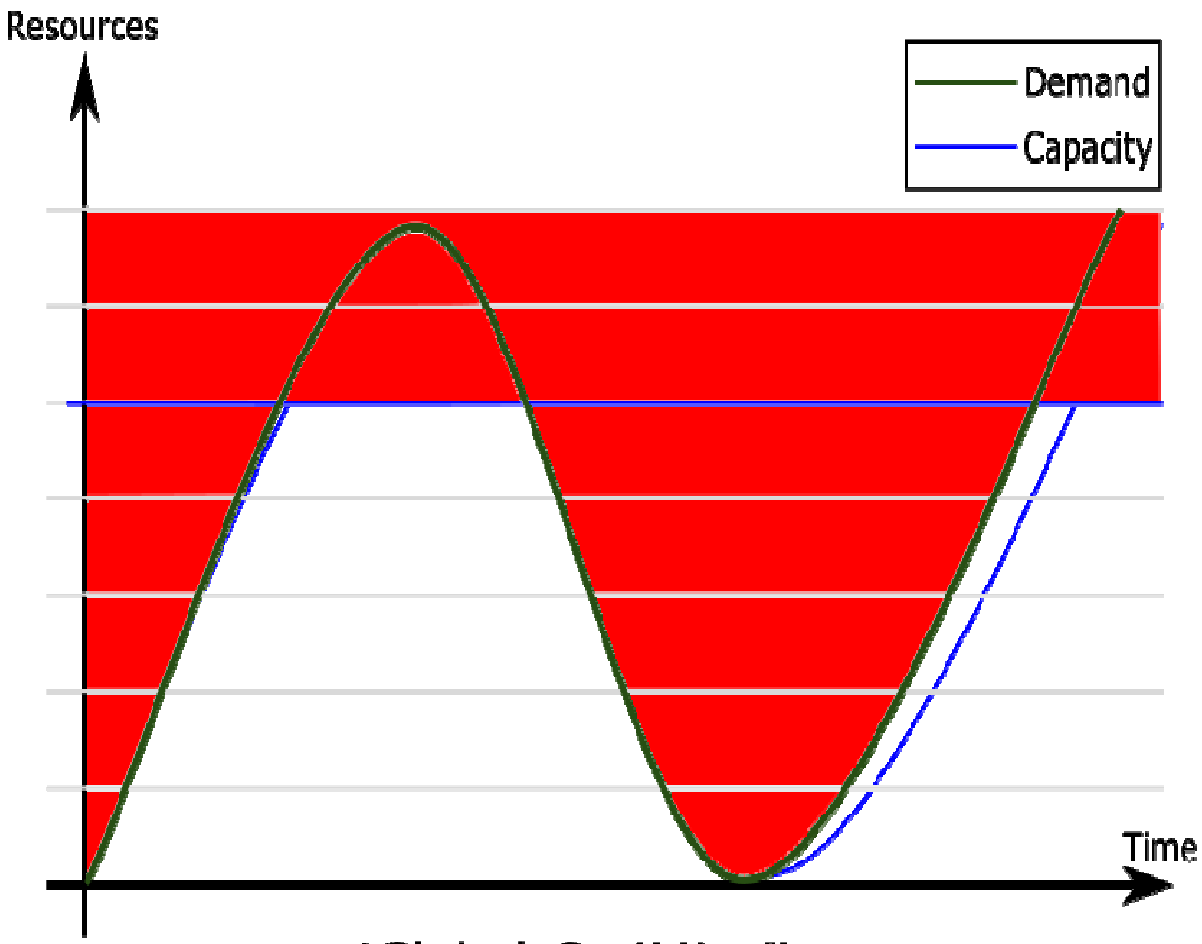


Elastic systems: connecting people, processes, and things

Elasticity: Enabling properties

Replaceable and reusable components	Interoperability
On-demand computing	Span cyber and physical worlds
Distributed computing	Adaptation
Self-contained units of functionality	Dynamic-perspective
Human-software composition	Business requirements orientation

Cloud Resource Provisioning



e·las·tic·i·ty |iˌlaˈstisitē; ēˌla-|

(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*

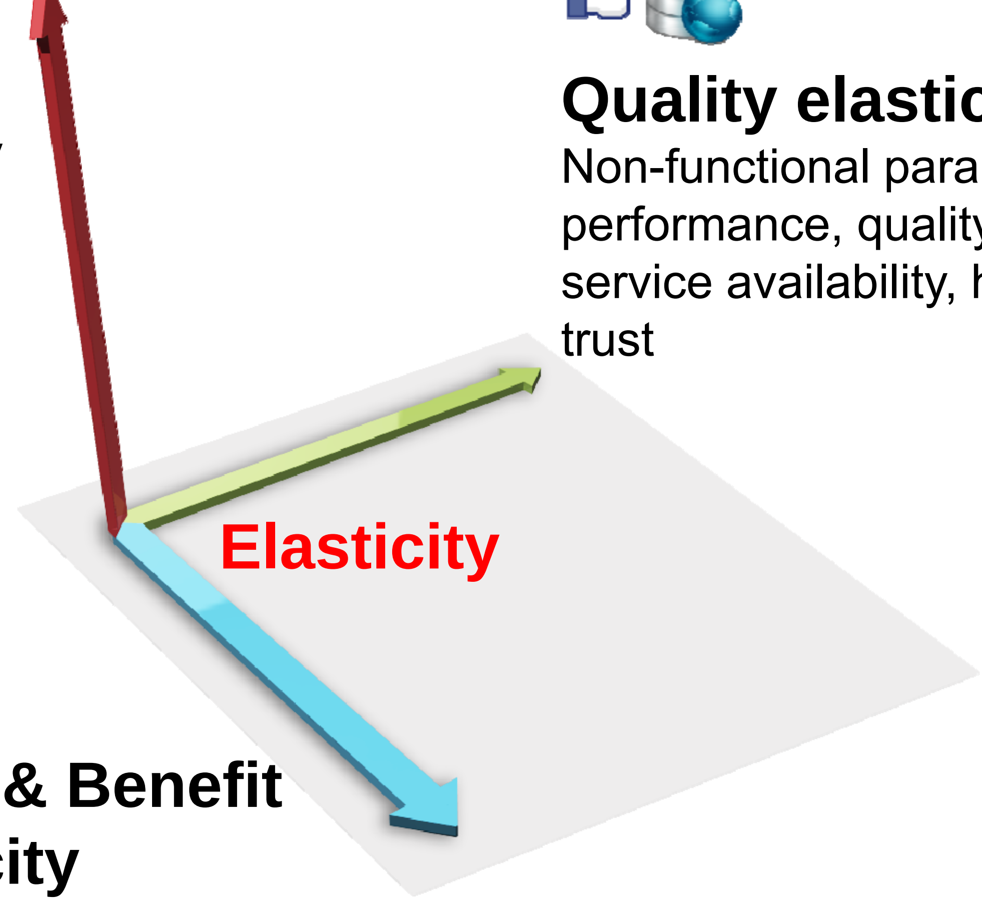
shrink when the stress is removed

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

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

Towards Elastic Systems **Design**

Which interactions between people, processes, and things are important?

Most programming languages consider humans as users, not “functional” entities

Cyber-Physical Embedded

Surrogate/Regression Models
Discrete Events
Actor Models

Differential Equations
Data Flow Languages
Synchronous Digital Logic

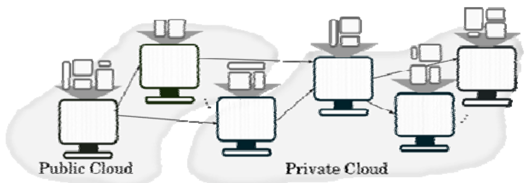
Smart Things

Distributed Systems

Adaptive Systems

Petri Nets
State Charts

Business Process Models
Boolean Circuits
State Machines

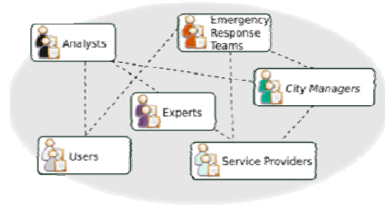


Cloud Computing

Human-based Systems

Organizations
Teams

Human-based Computing



Towards Elastic Systems Run-Time

How can we leverage heterogeneous capabilities of humans, processes, things?

Can people be monitored and controlled similar to computing resources?

Cyber-Physical

Control Theory
Finite State Automata

Embedded

Finite State Automata
Programmable Controller

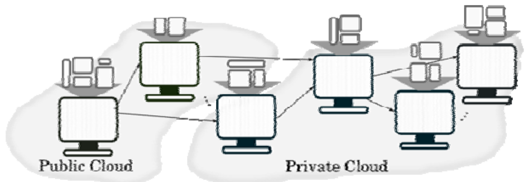
Smart Things

Distributed Systems

Adaptive Systems

Neural Networks
Probabilistic Methods
Autonomic Computing

Control Theory
Finite State Automata
Choreography/Orchestration

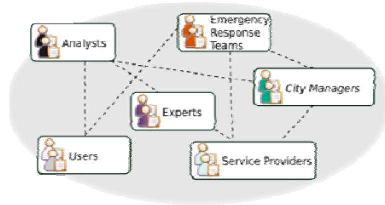


Cloud Computing

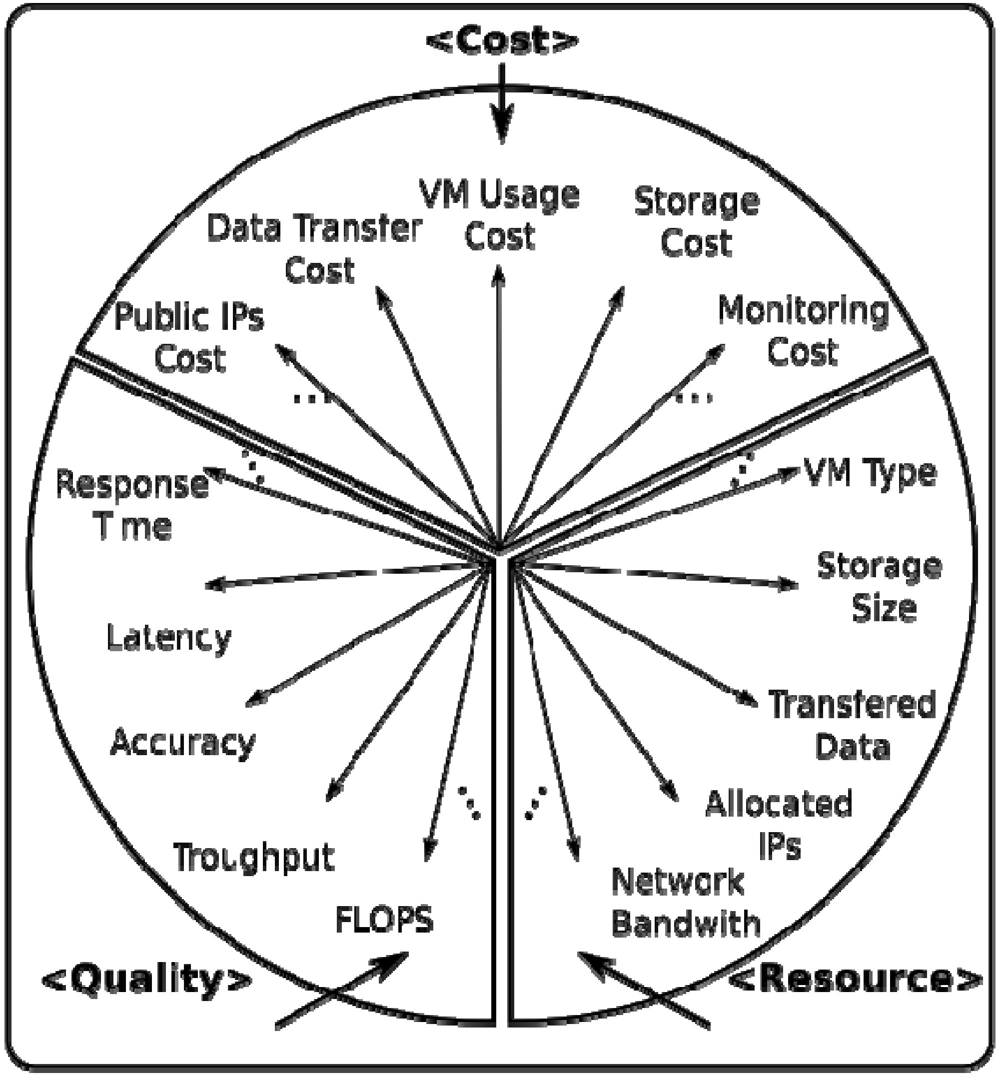
Human-based Systems

Coordination
Collaboration
Incentives

Human-based Computing

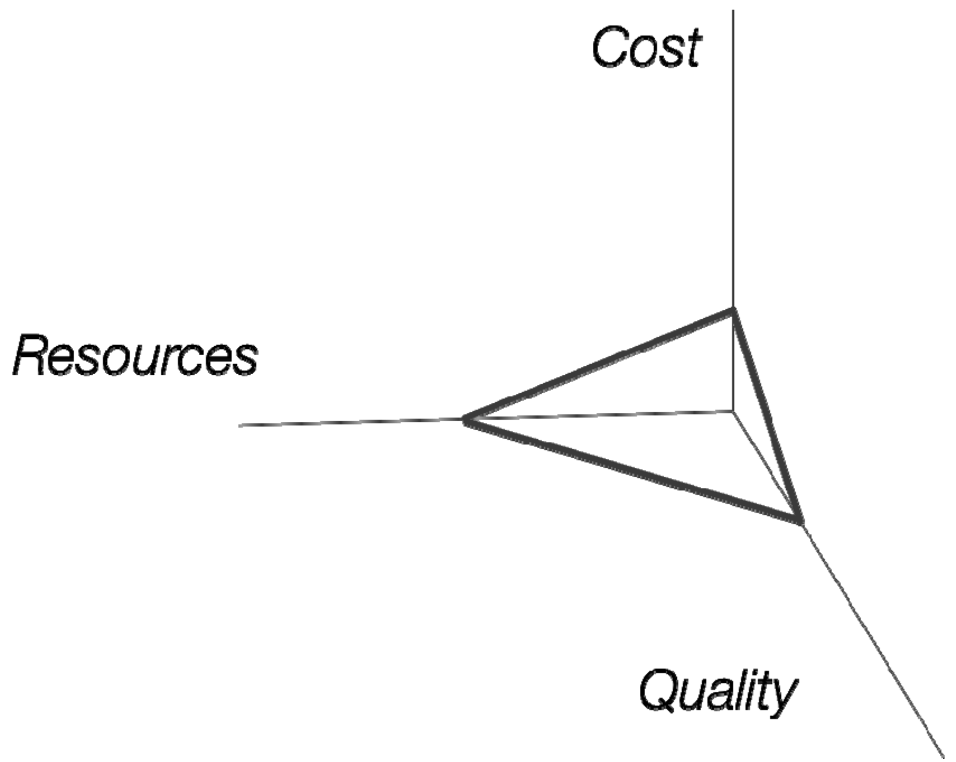


Multidimensional Elasticity



Elasticity Model

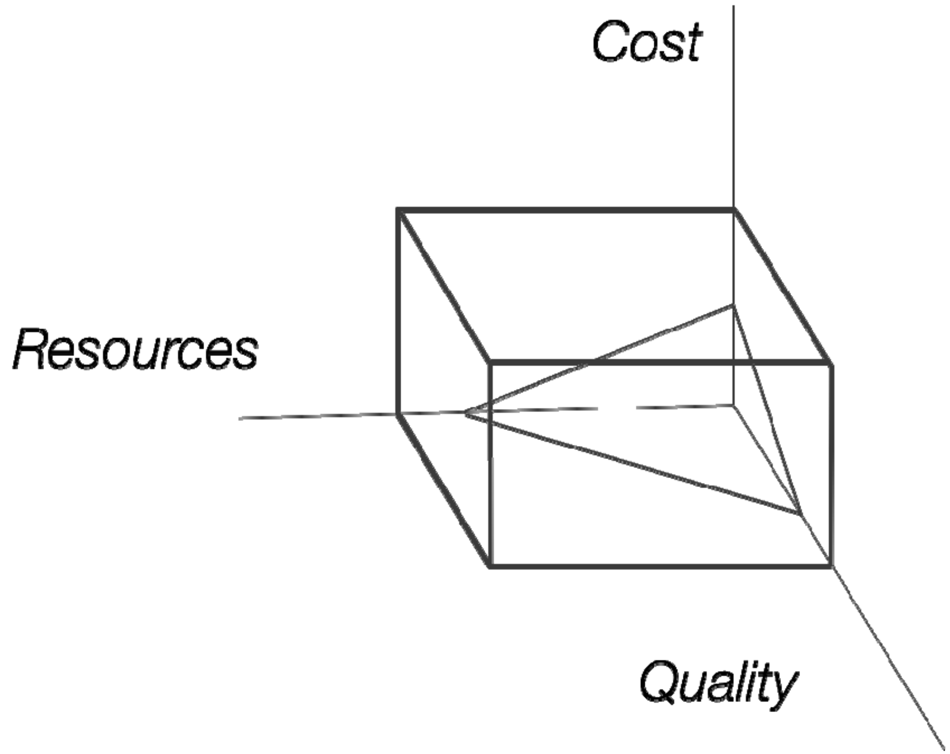
Elasticity Signature



Elasticity Model

Elasticity Signature

Elasticity Space



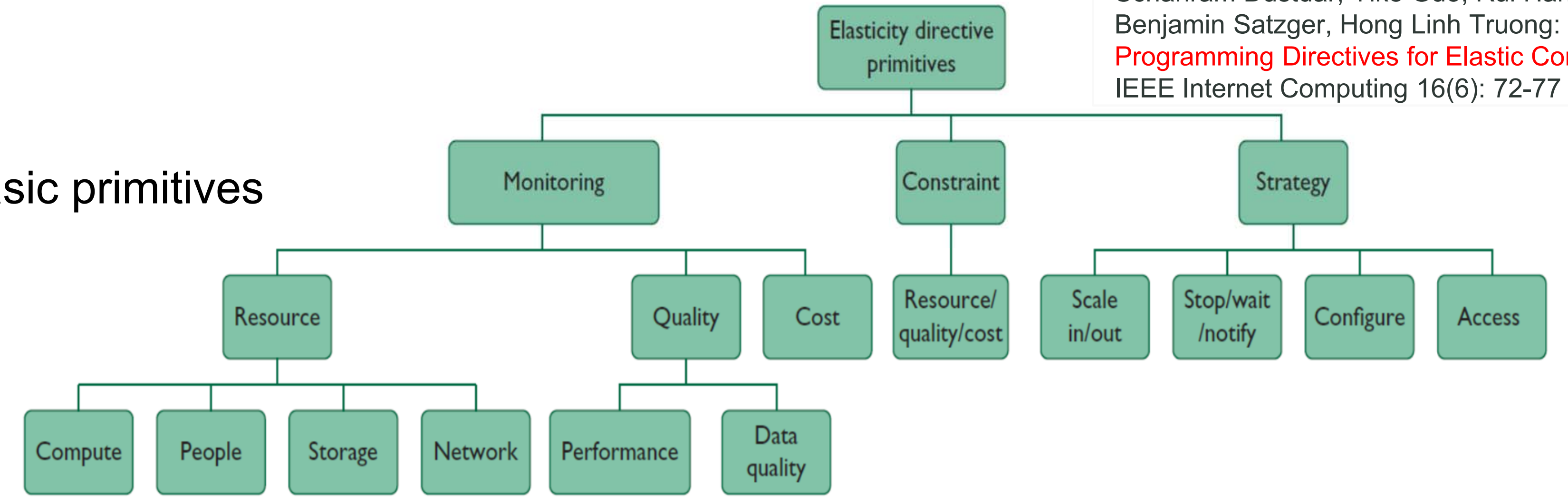
Elasticity Analytics – Some Scenarios

- Elasticity of data resources
  - Activate/change sensor deployment/configurations for required data; changing types of data sources for analytics
- Elasticity of cloud platform services
  - Deploy/reconfigure cloud services handling changing data
- Elasticity of data analytics
  - Switch and combine different types of data analytics processes and engines due to the severity of problems and quality of results
- Elasticity of teams of human experts
  - Forming and changing different configurations of teams during the specific problems and problem severity

Specifying and controling elasticity

Schahram Dustdar, Yike Guo, Rui Han, Benjamin Satzger, Hong Linh Truong: 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

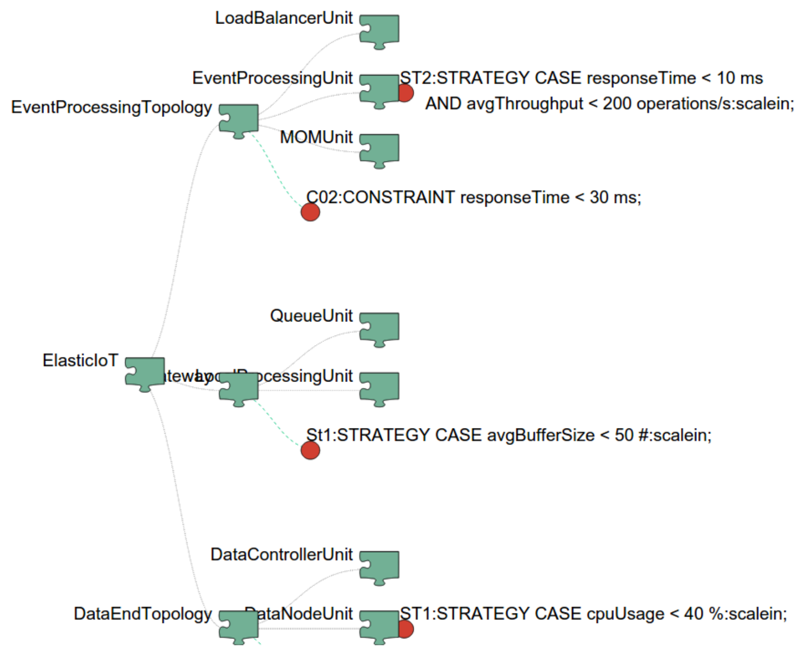
<tosca:ServiceTemplate name="PilotCloudService">
<tosca:Policy name="St1" policyType="SYBLStrategy"> St1:STRATEGY minimize(Cost) WHEN high(overallQuality) </tosca:Policy>...

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



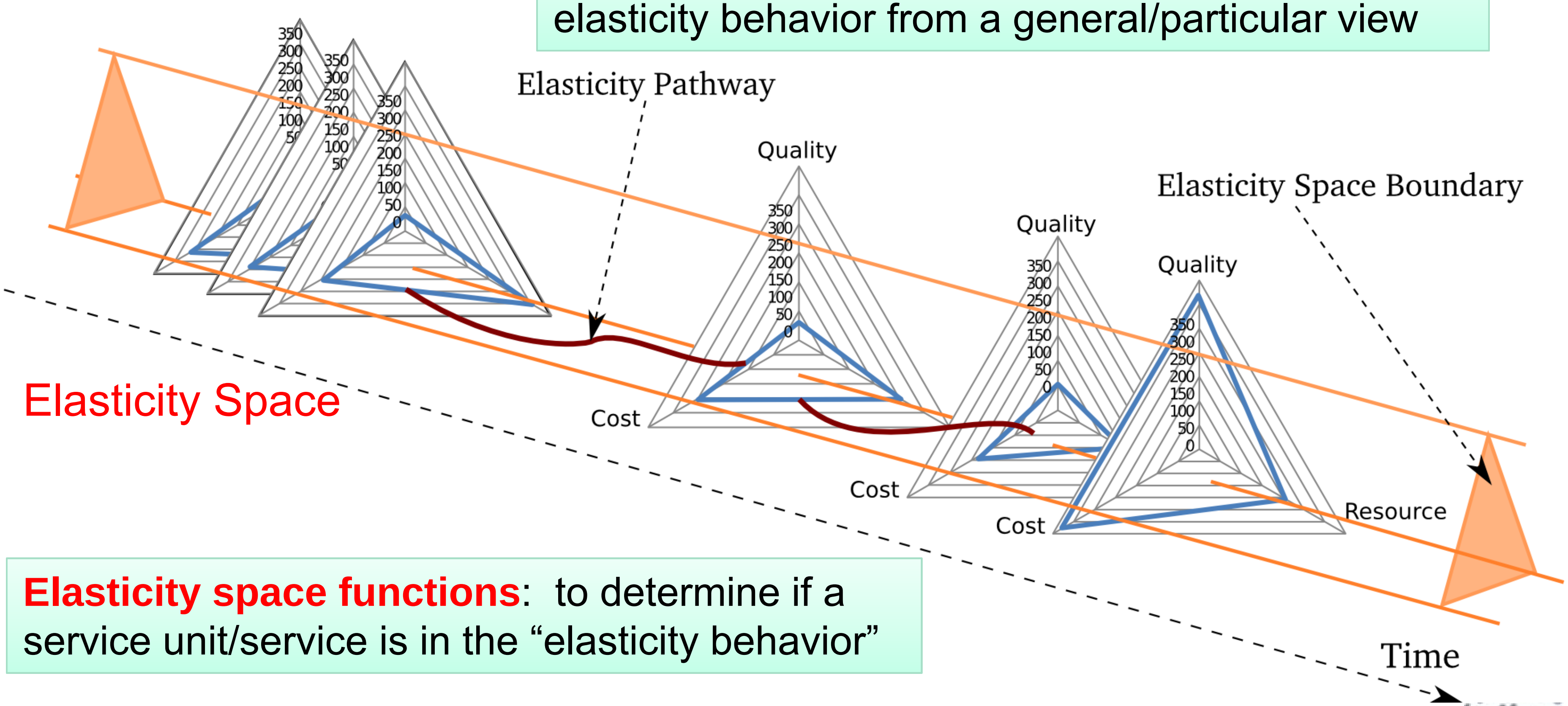
Georgiana Copil, Daniel Moldovan, Hong-Linh Truong, Schahram Dustdar, **"SYBL: an Extensible Language for Controlling Elasticity in Cloud Applications"**, 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), May 14-16, 2013, Delft, Netherlands

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"

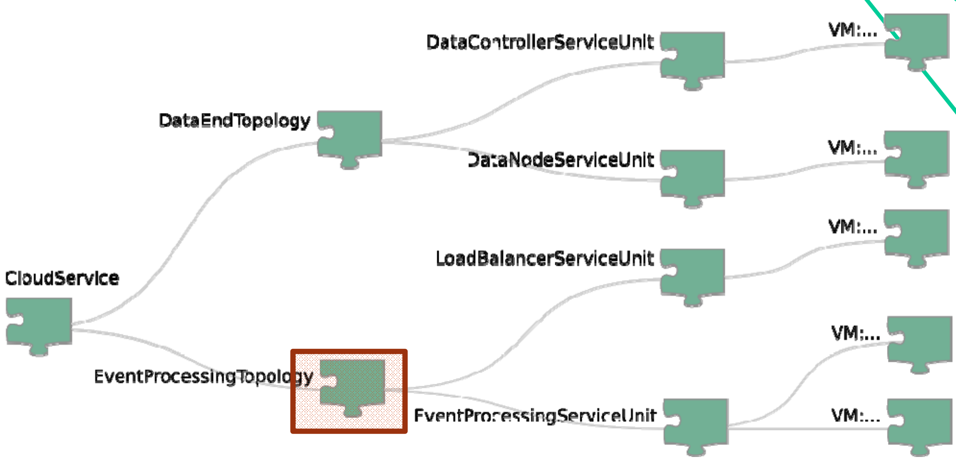
Multi-Level Elasticity Space

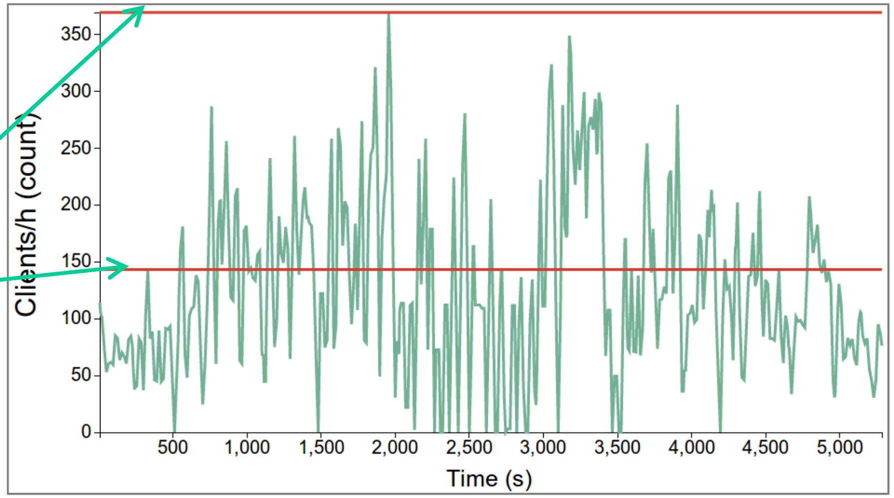
Service requirement

COST<= 0.0034$/client/h

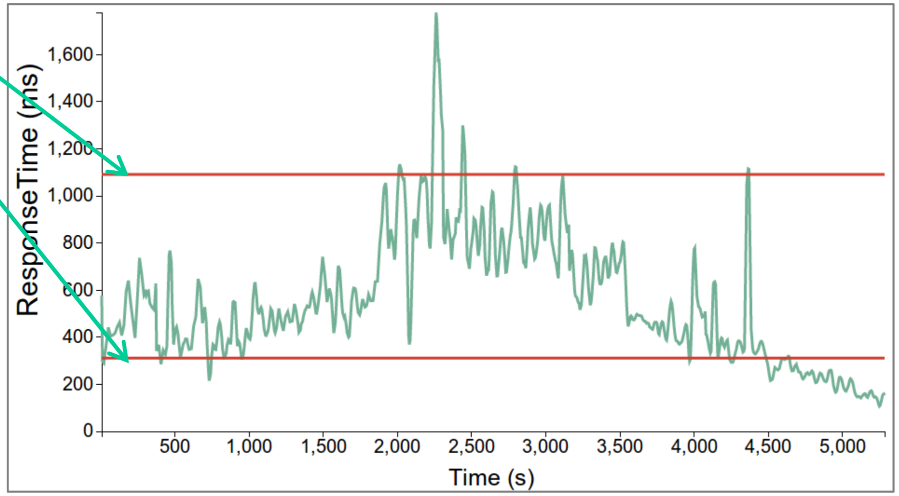
2.5$ monthly subscription for each service client (sensor)

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





Elasticity Space "Clients/h" Dimension

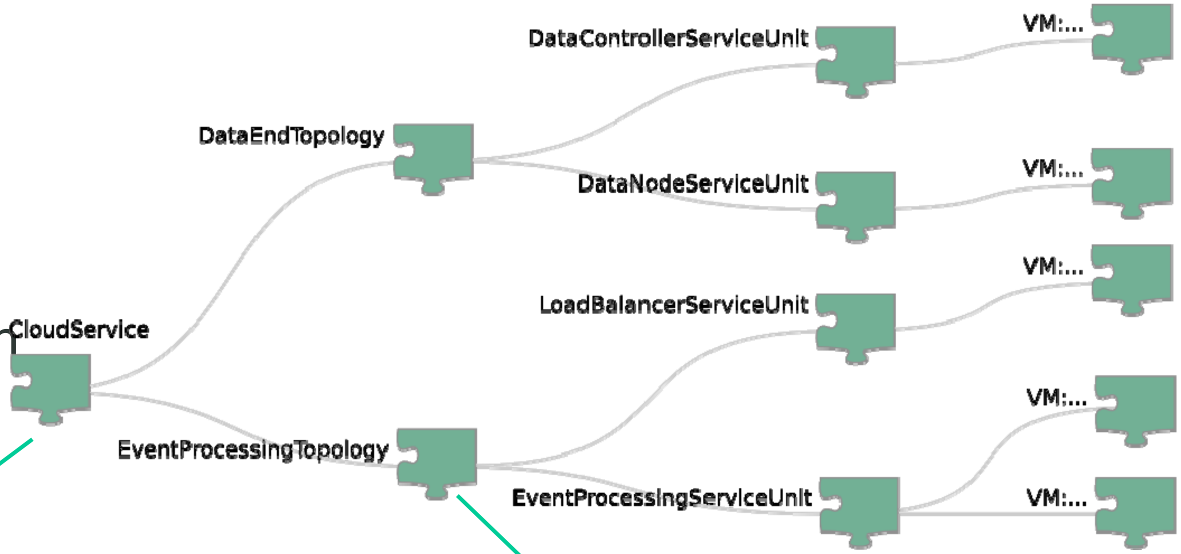


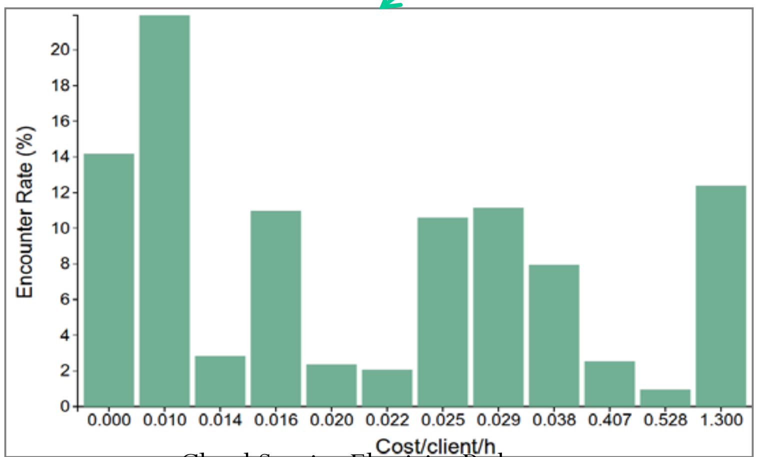
Elasticity Space "Response Time" Dimension

Multi-Level Elasticity Pathway

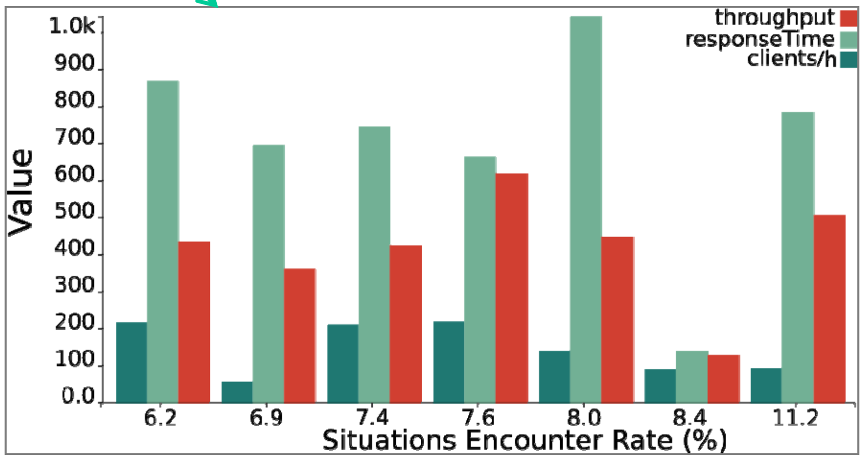
Service requirement

- COST<=
  0.0034$/client/h
- 2.5$ monthly
  subscription for each
  service client
  (sensor)





Cloud Service Elasticity Pathway



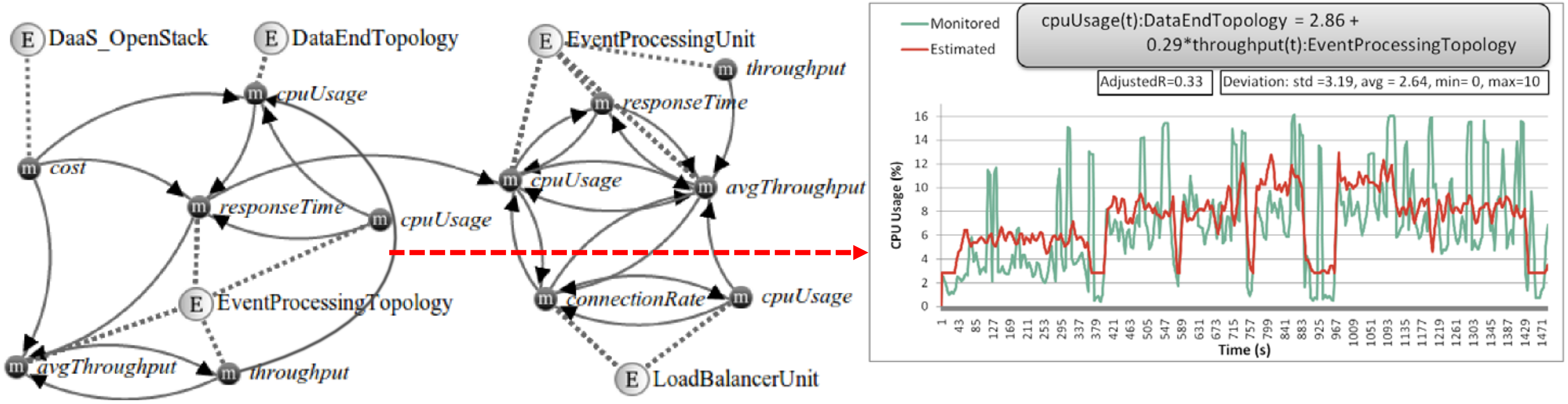
Event Processing service unit Elasticity Pathway

Elasticity space and pathway analytics

Daniel Moldovan, Georgiana Copil, Hong-Linh Truong, Schahram Dustdar, **MELA: Elasticity Analytics for Cloud Services**, International Journal of Big Data Intelligence, 2014

Elasticity dependency analysis

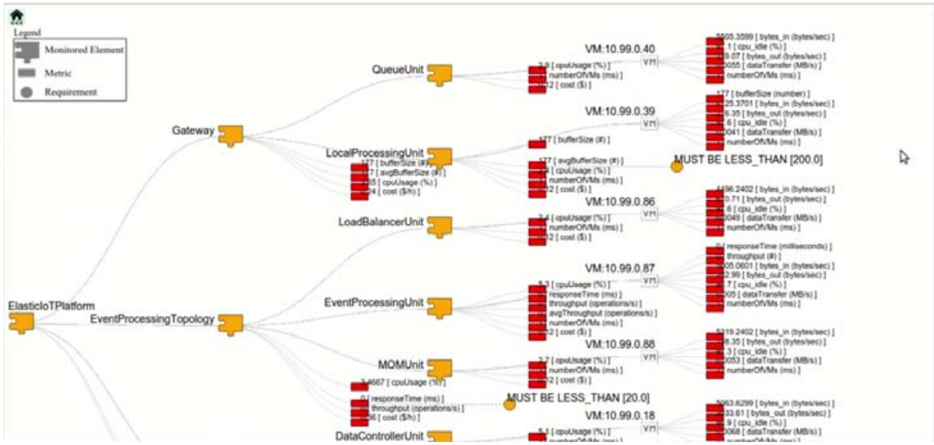
- The elasticity of a service unit affects the elasticity of another unit. How to characterize such cause-effect: **elasticity dependency**
- Modeling **collective metrics evolution** in relation to requirements



Daniel Moldovan, Georgiana Copil, Hong-Linh Truong, Schahram Dustdar, **On Analyzing Elasticity Relationships of Cloud Services**, 6th International Conference on Cloud Computing Technology and Science, 15-18 December 2014, Singapore

Enable elasticity reconfiguration at runtime

Analysis detects problems but predefined strategies do not always work!

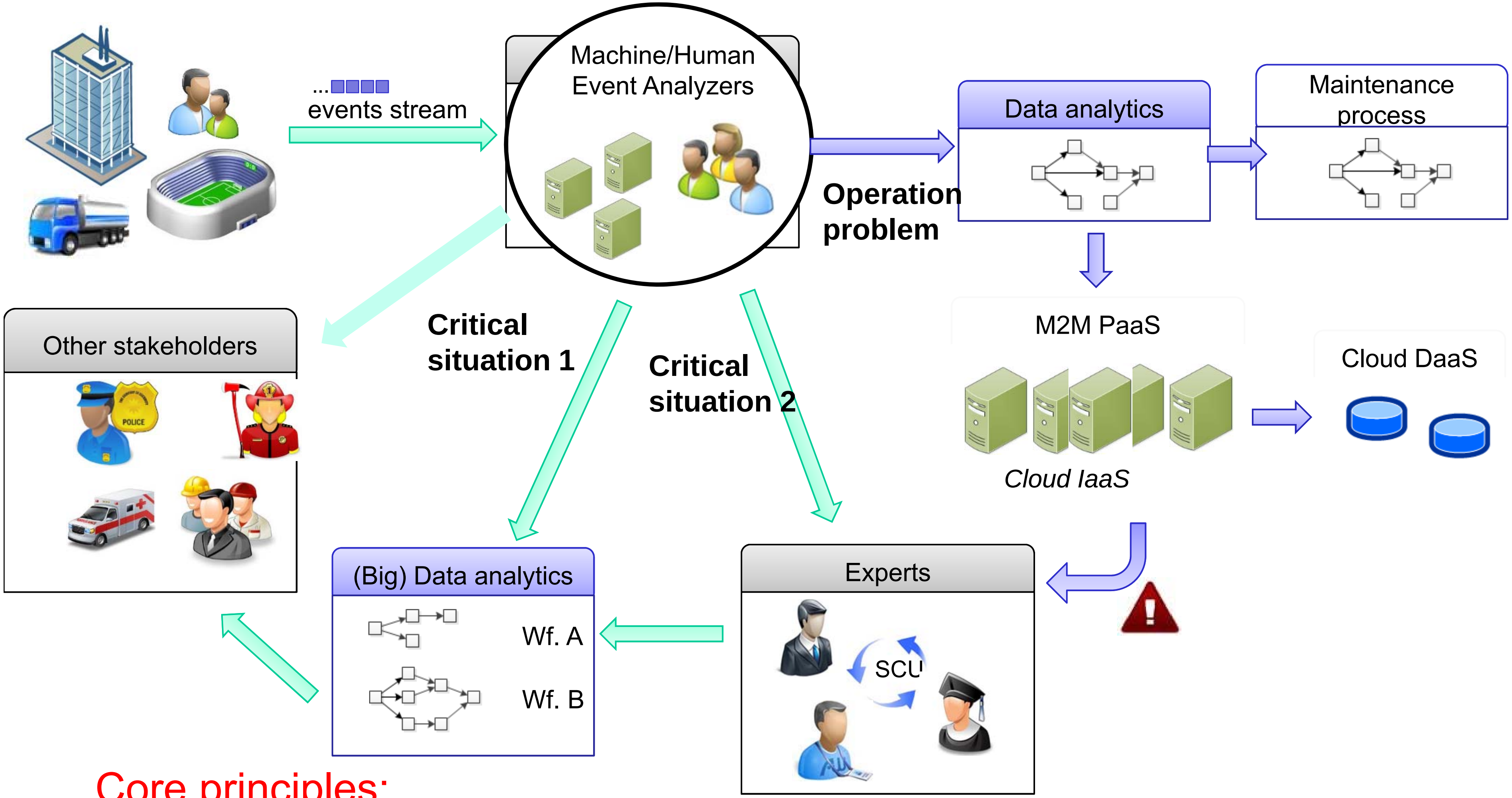


Changing elasticity specifications at runtime without stoping services

Elastic Computing for the Internet of Things

Smart City Dubai
Pacific Controls
Command Control Center
SMEs
Dashboards
User interfaces
Reports
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

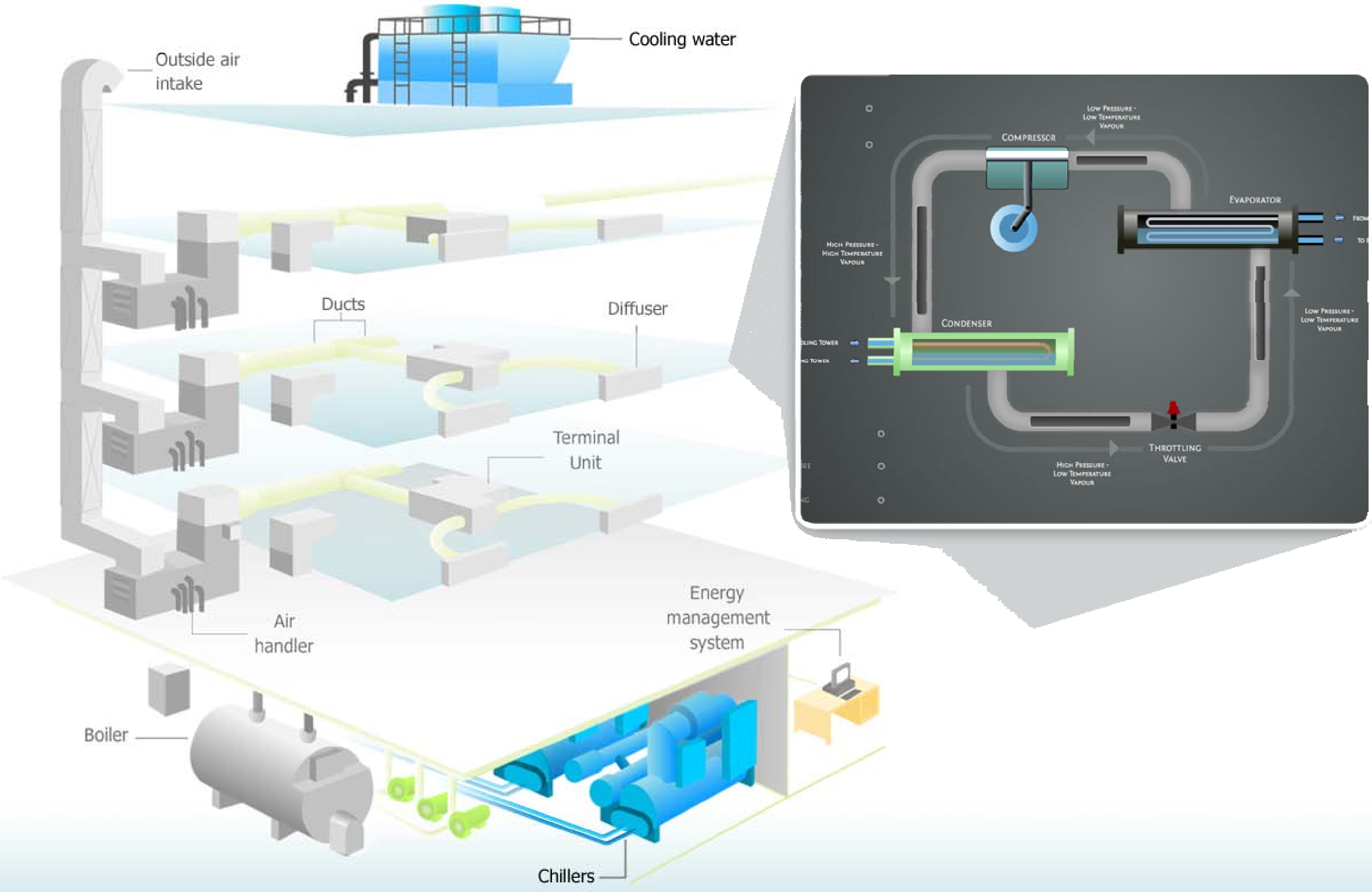
Processes with machines and people



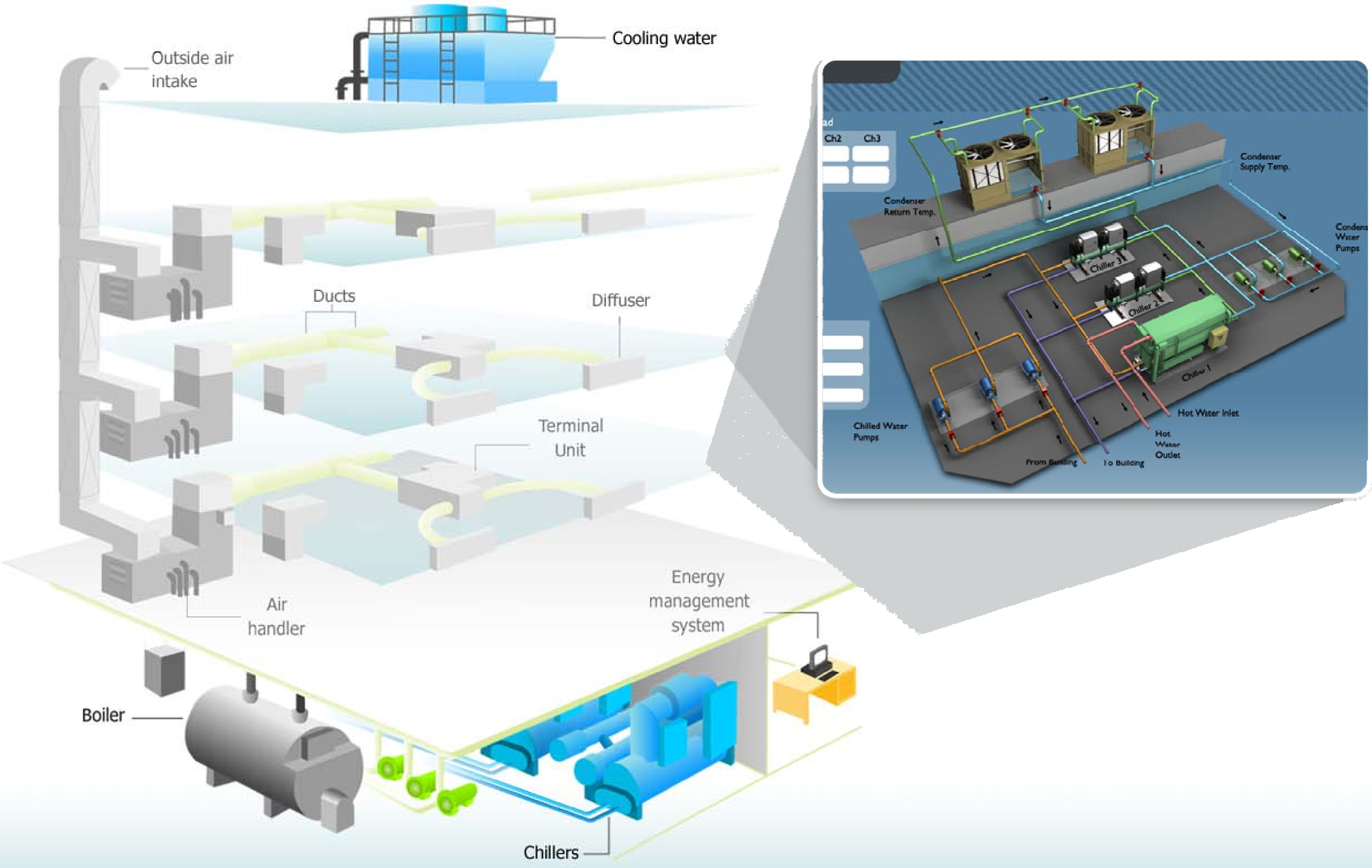
Core principles:

- Human computation capabilities under elastic service units
- “Programming“ human-based units together with software-based units

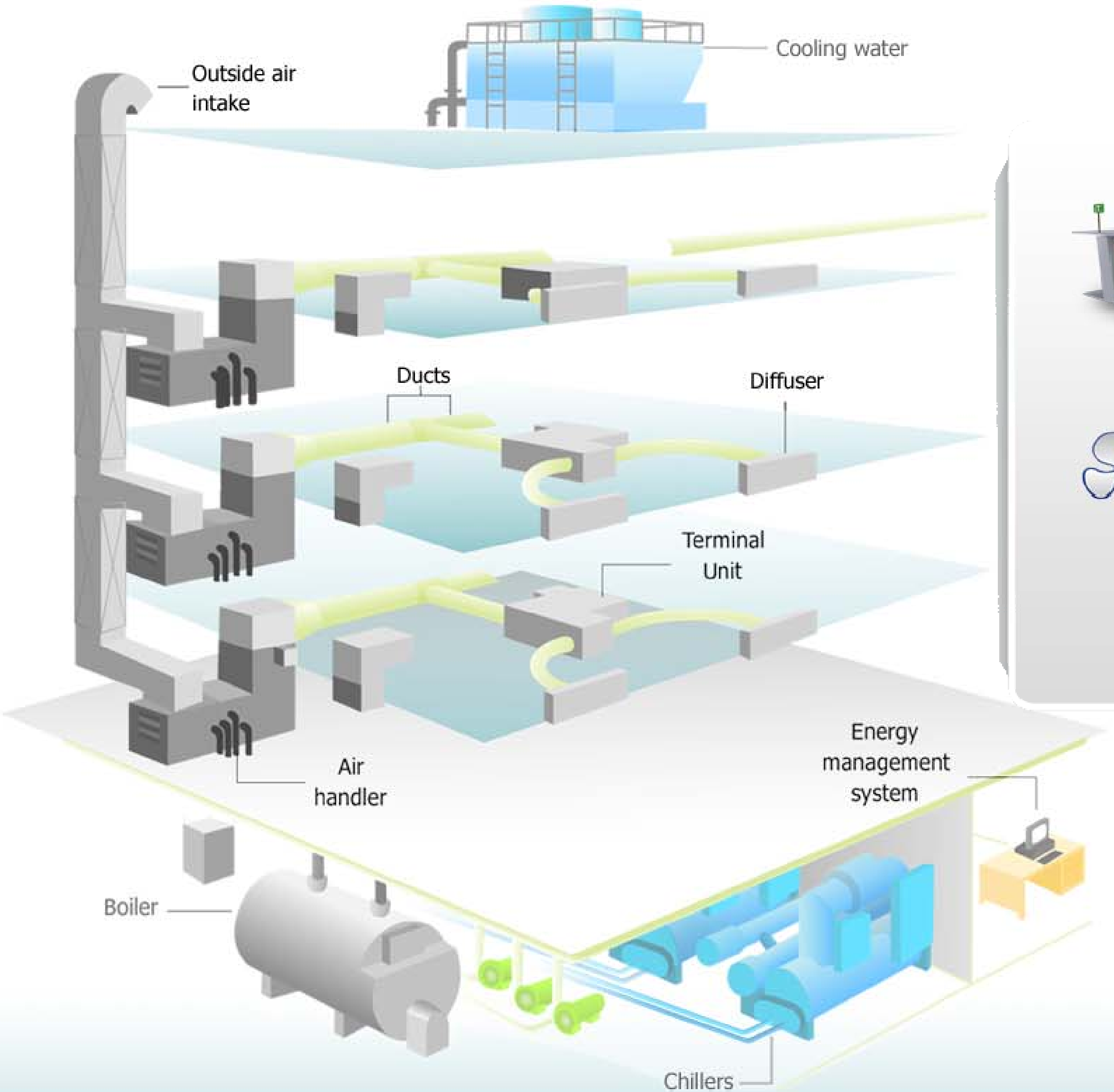
HVAC (Heating, Ventilation, Air Conditioning) Ecosystem

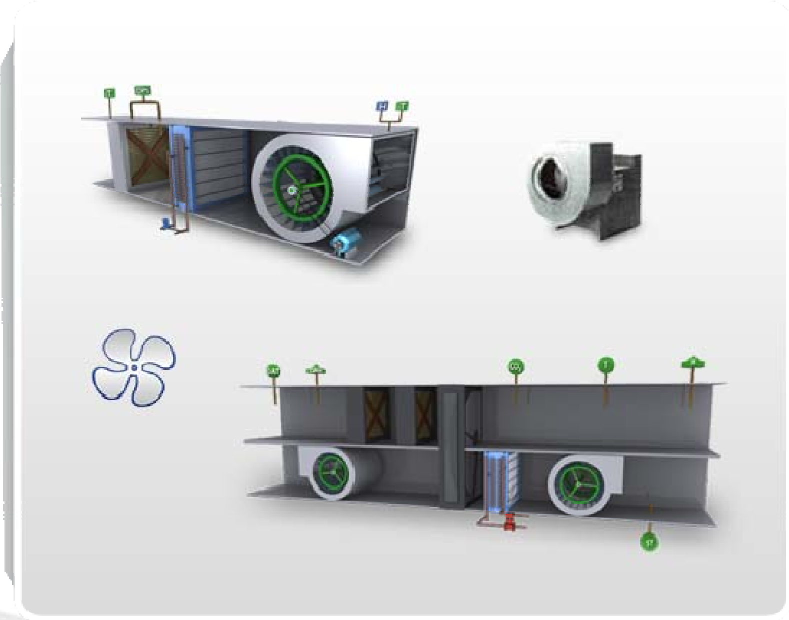


Water Ecosystem

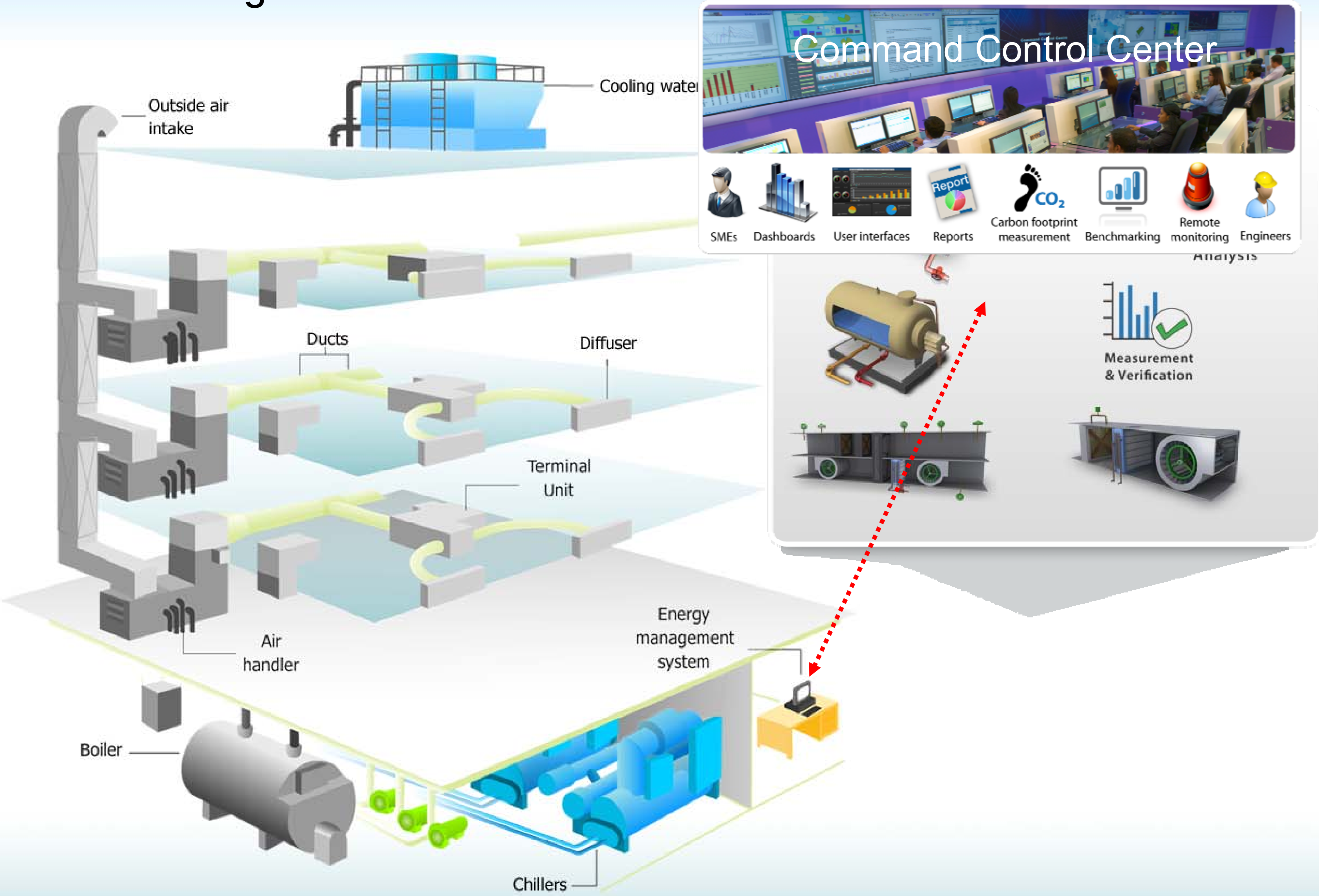


Air Ecosystem



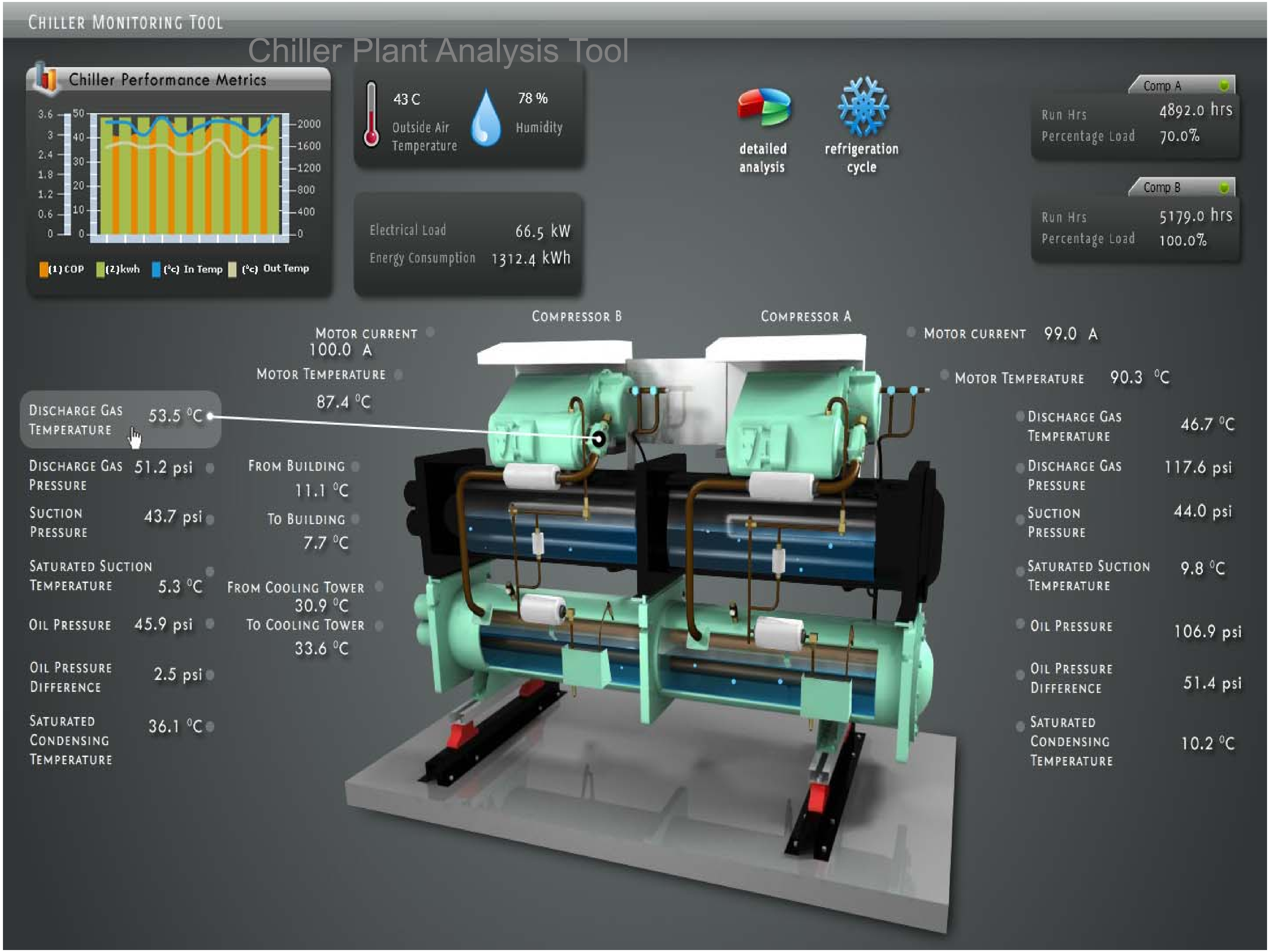


Monitoring

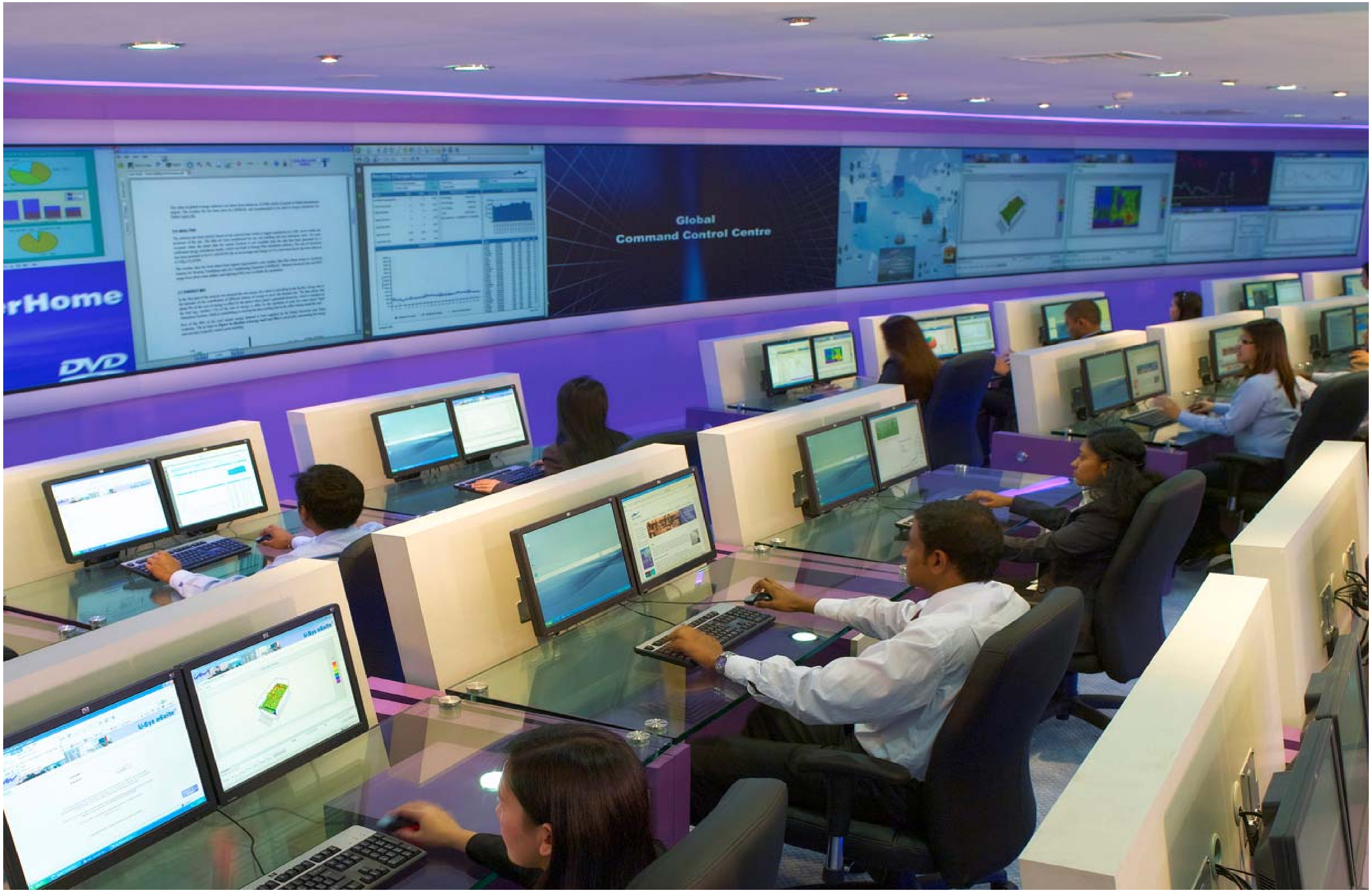


Command Control Center
SMEs
Dashboards
User interfaces
Reports
Carbon footprint measurement
Benchmarking
Remote monitoring
Engineers
Measurement & Verification
Cooling water
Outside air intake
Ducts
Diffuser
Terminal Unit
Air handler
Energy management system
Boiler
Chillers

Chiller Plant Analysis Tool



Command Control Center for Managed Services



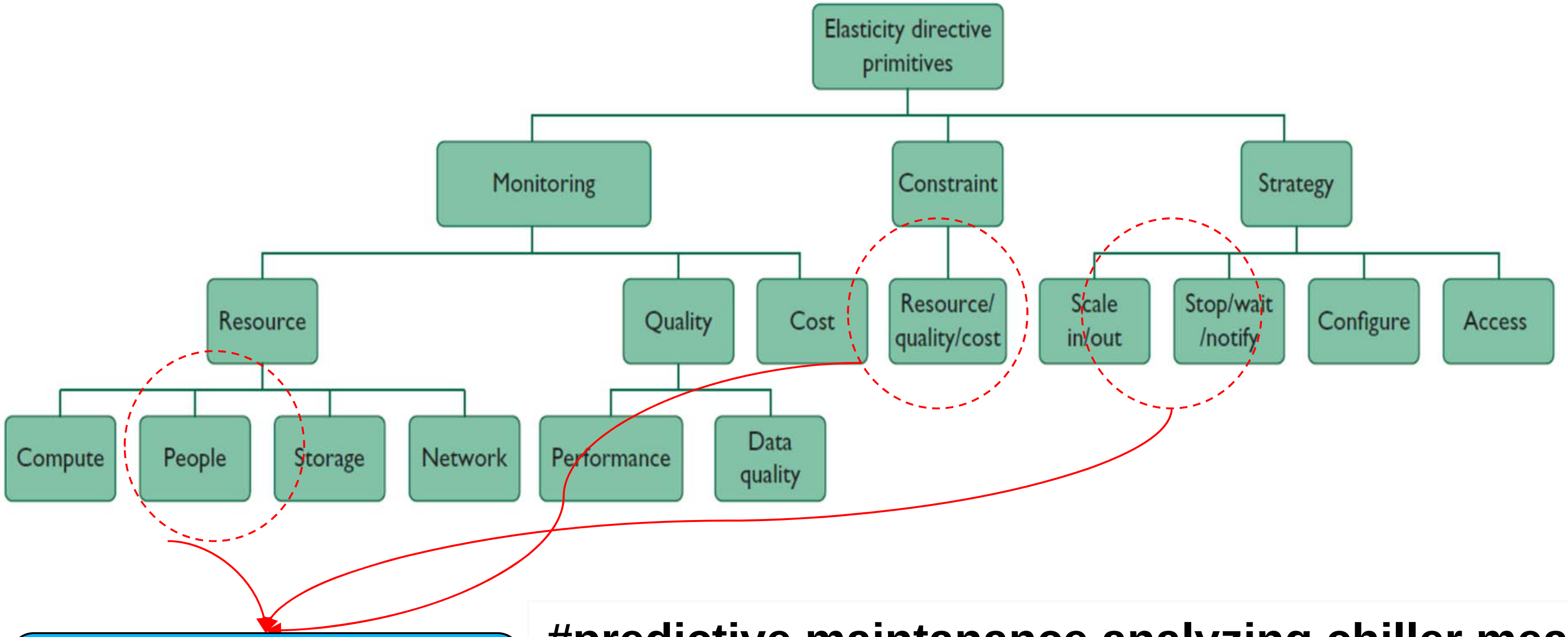
Elastic Computing for People

Distributed Systems Group

Human-based service elasticity

- Which types of human-based service instances can we provision?
- How to provision these instances?
- How to utilize these instances for different types of tasks?
- Can we program these human-based services together with software-based services
- How to program incentive strategies for human services?

Specifying and controling elasticity of human-based services



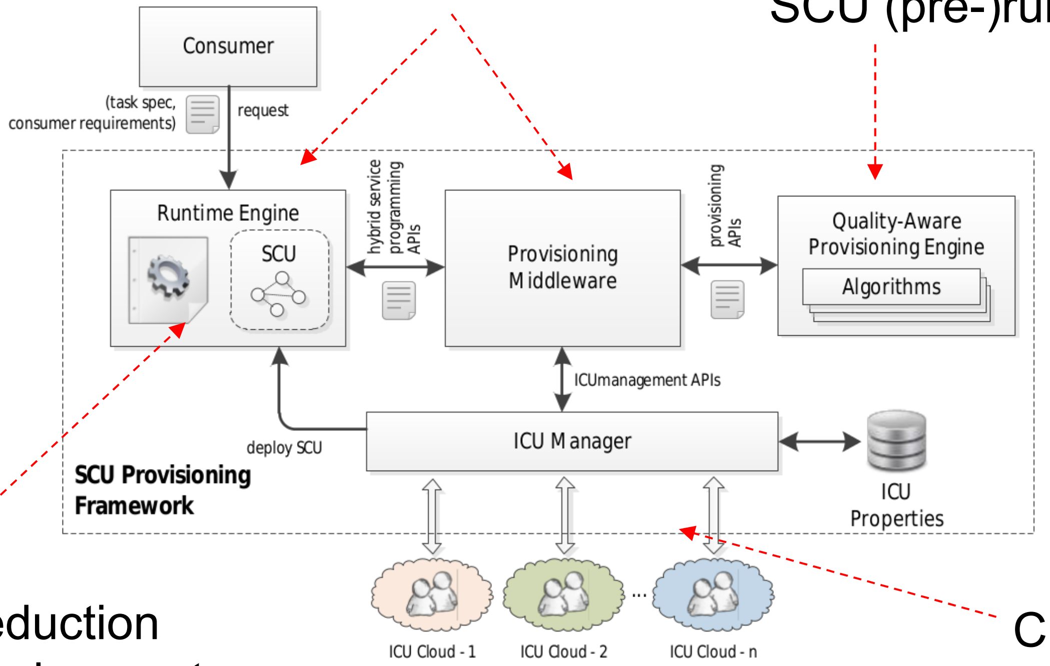
What if we need to invoke a human?

#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)

Elastic SCU provisioning

Elastic profile

SCU (pre-)runtime/static formation



Algorithms
- Ant Colony Optimization variants
- FCFS
- Greedy

SCU extension/reduction
- Task reassignment based on trust, cost, availability

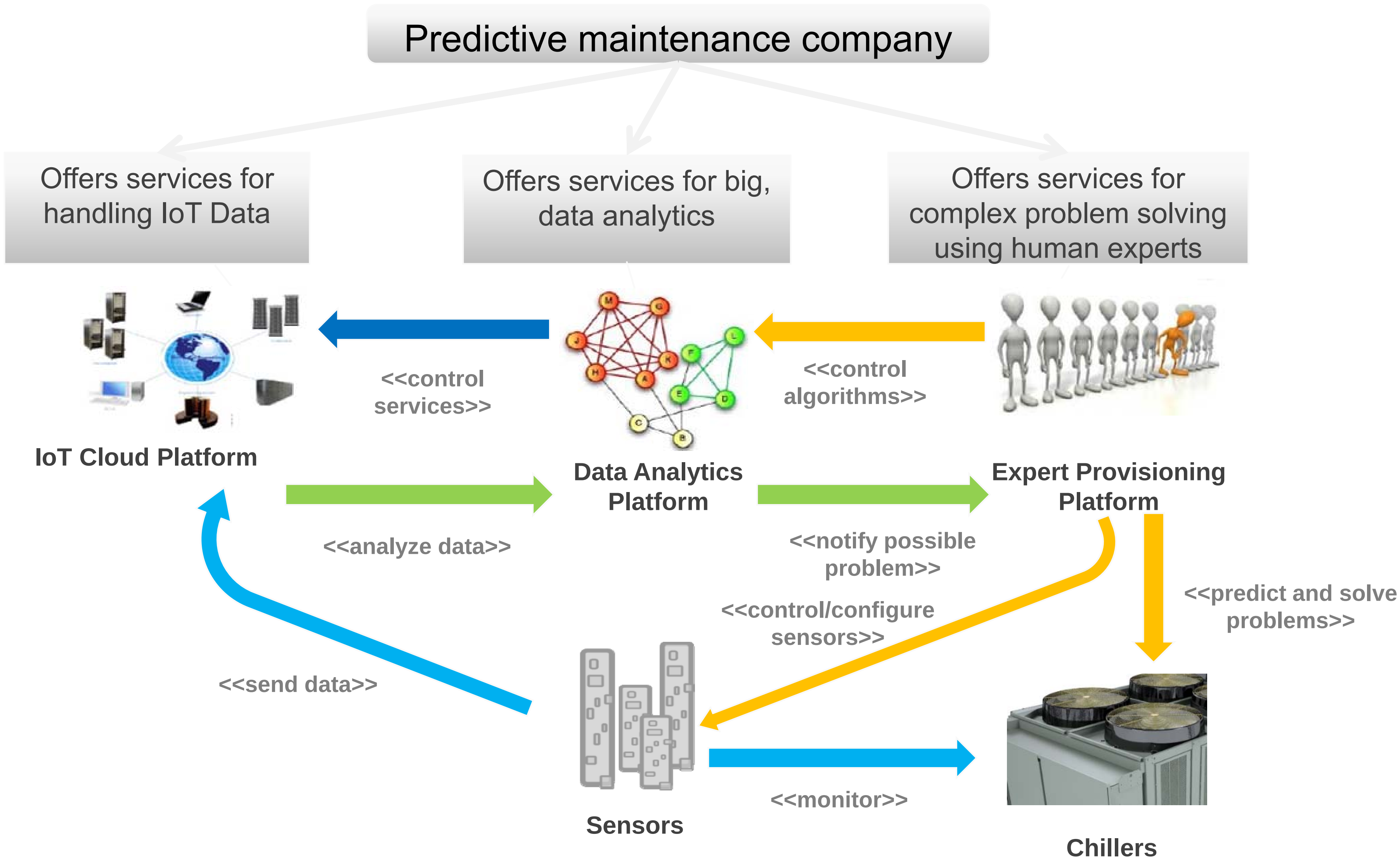
Cloud APIs

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.**

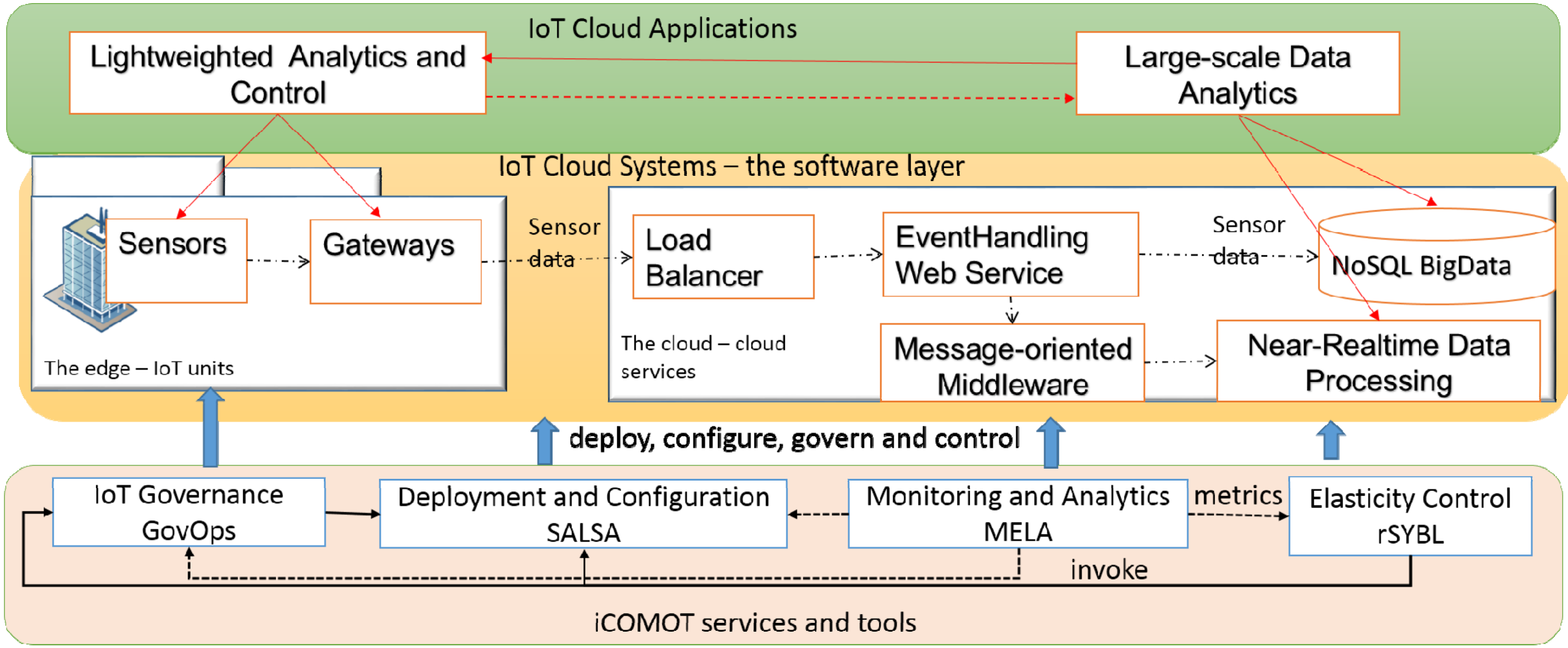
Tooling

Scenario



Predictive maintenance company
Offers services for handling IoT Data
Offers services for big, data analytics
Offers services for complex problem solving using human experts
<<control services>>
<<control algorithms>>
IoT Cloud Platform
Data Analytics Platform
Expert Provisioning Platform
<<analyze data>>
<<notify possible problem>>
<<control/configure sensors>>
<<predict and solve problems>>
<<send data>>
<<monitor>>
Sensors
Chillers

iCOMOT – IoT and Cloud Monitoring, Control and Testing



Check: http://tuwiendsg.github.io/iCOMOT/
http://tuwiendsg.github.io/COMOT/elasticityDemo.html

Conclusions and Outlook

- **Elasticity**
  - Crucial for ensuring **quality of results** in a continuum of different computing platforms integrated software, people, and things
  - Coordinating elasticity across platforms needs concepts of elastic objects and fundamental building blocks for engineering an end-to-end elasticity for cloud services → see our prototypes

- **Ongoing work**
  - Programming languages for elastic objects
  - Elasticity coordination

Thanks for your attention!

Prof. Schahram Dustdar,
IEEE Fellow

Distributed Systems Group
TU Wien

dsg.tuwien.ac.at

VieCom The Vienna Elastic Computing Model

Overview Challenges People Publications Prototypes Projects

- SmartCom
- QUELLE
- SALSA
- Estimating Actuation Delays in Elastic Computing Systems
- SYBL
- ADVISE
- MELA
- Hybrid service ecosystems
- Elasticity Profile
- SCU Elasticity

SmartCom - A Communication Middleware for Hybrid Diversity-Aware Collective Adaptive Systems (HDA-CAS)

- Prototype and description

QUELLE - Framework for Accelerating the Development of Elastic Systems

- Prototype and description

SALSA - Framework for Dynamic Configuration of Cloud Services

- Prototype and description

Software-defined IoT Cloud systems

- Provisioning and governance framework

On Estimating Actuation Delays in Elastic Computing Systems

- Description and experiments

SYBL - Simple Yet Beautiful Language for Elasticity Controls

- SYBL Design and Runtime
- SYBL + MELA Demo

ADVISE - a Framework for Evaluating Cloud Service Elasticity Behavior

- Prototype and description

MELA: Monitoring And Analyzing Elasticity Of Cloud Services

- Prototype, documentation and demos

Hybrid Service Ecosystems

- A framework for managing service ecosystems in the Vienna Elastic Computing Model

Elasticity Profile

- Elasticity Modeling for Mixed Systems

On the Elasticity of Social Compute Units

- Discussion and details