Internet of Things, People, and Processes

FICloud Keynote – Barcelona 28 Aug 2014

Schahram Dustdar

Distributed Systems Group
TU Vienna

dsg.tuwien.ac.at

Distributed Systems Group

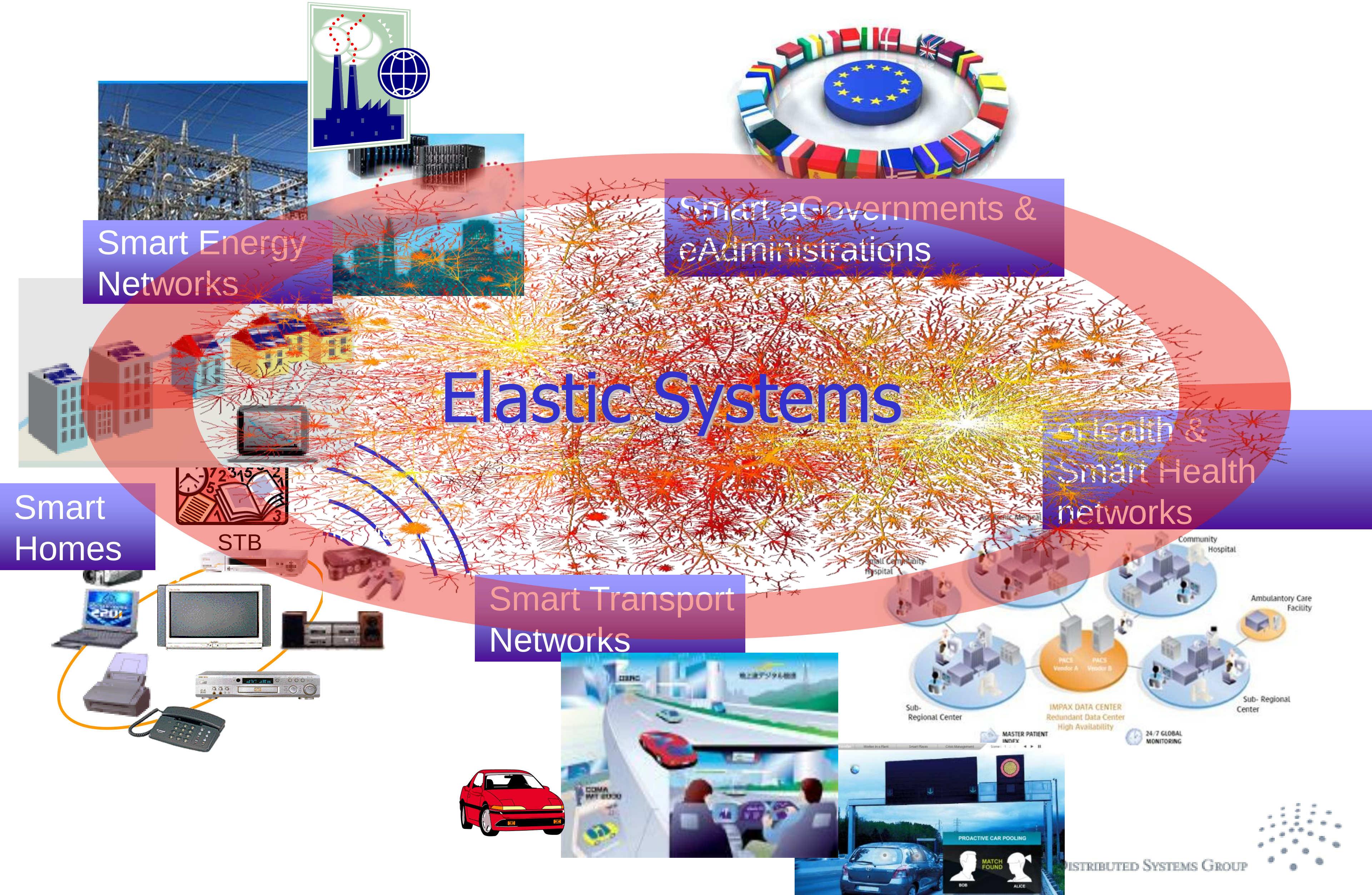
Acknowledgements

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

celar SmartSociety PACIFIC CONTROLS

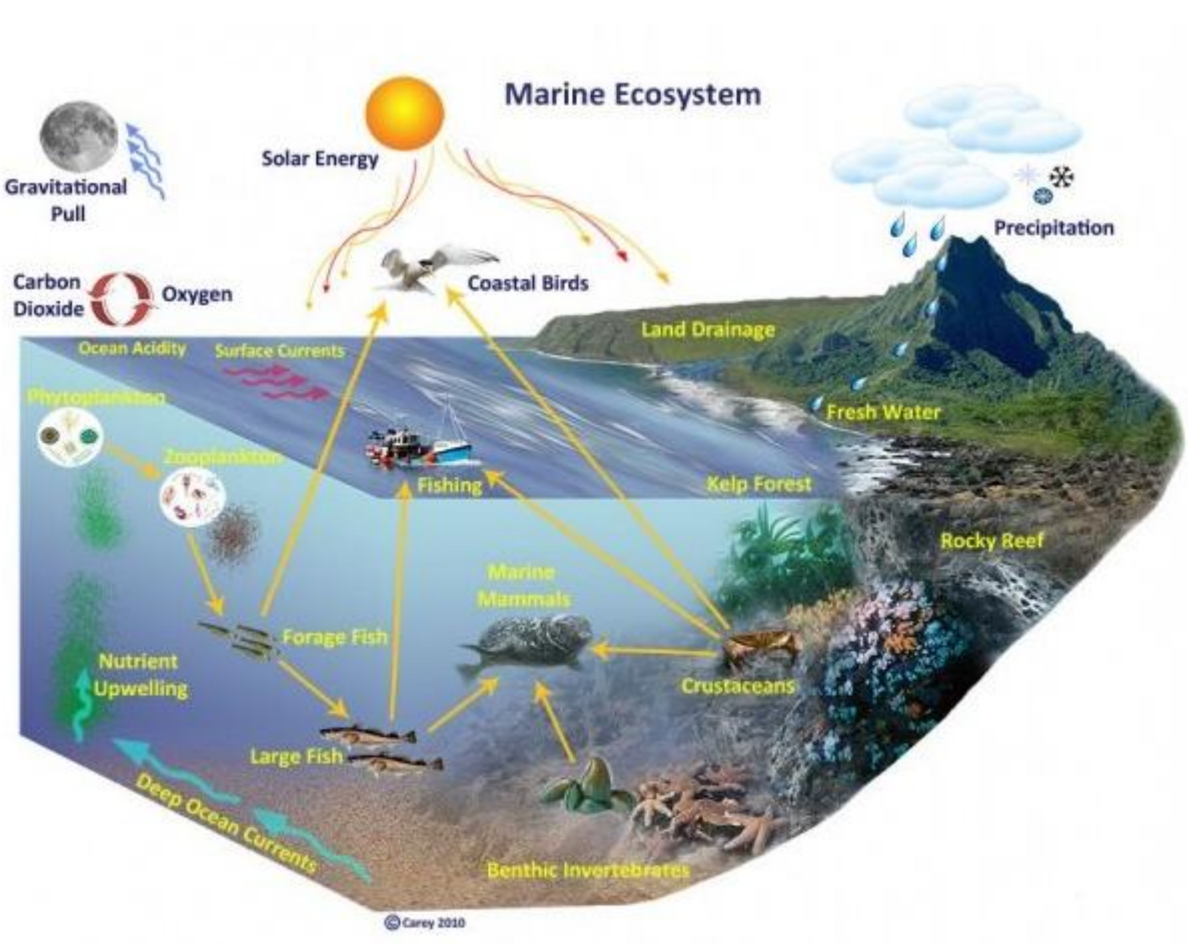
NOTE: The content includes some ongoing work

Smart Evolution – People, Services, Things



1. "Resources" provided as services
2. Illusion of infinite resources
3. Usage-based pricing model ->
   New and connected business models

Think Ecosystems:
People, Services, Things

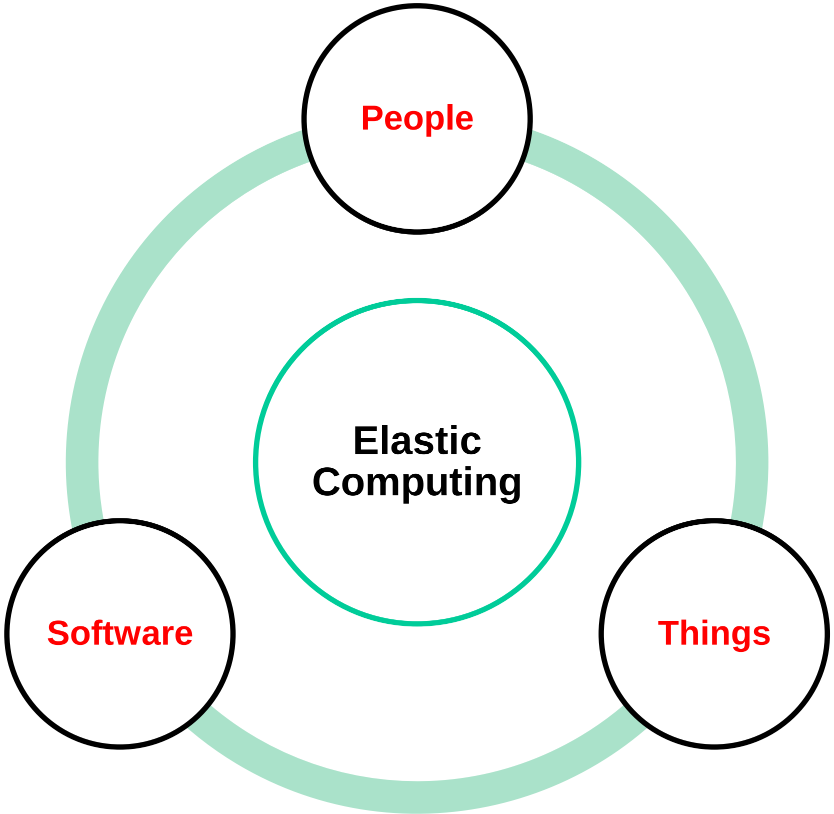


Marine Ecosystem: http://www.xbordercurrents.co.uk/wildlife/marine-ecosystem-2

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

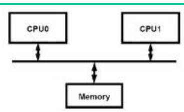
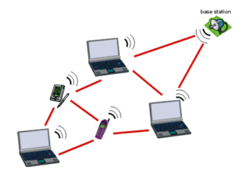
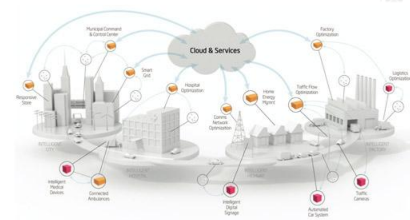
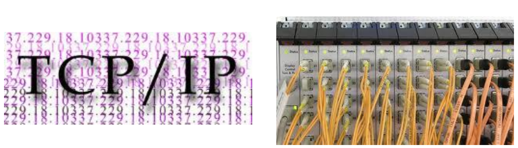
1. **Robustness mechanisms**: achieving stability in the presence of disruption
2. **Measures of health**: diversity, population trends, other key indicators

Approach

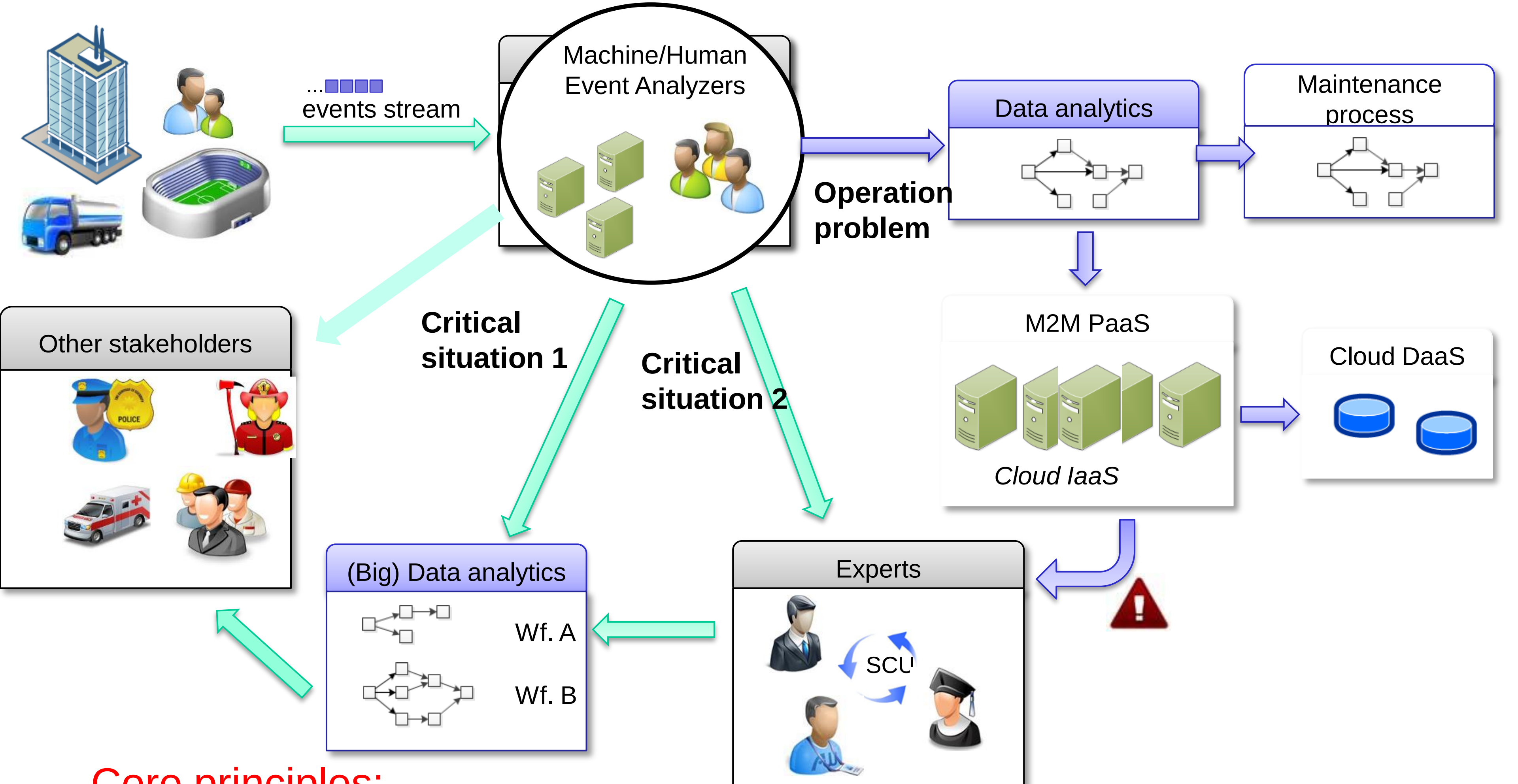


Computing Models

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

	Machine-based Computing	Human-based Computing	Things-based computing
Processing Unit			
Architecture	 SMP   Grid		Ad hoc networks  Web of things  
Comm.			

Connecting machines and people



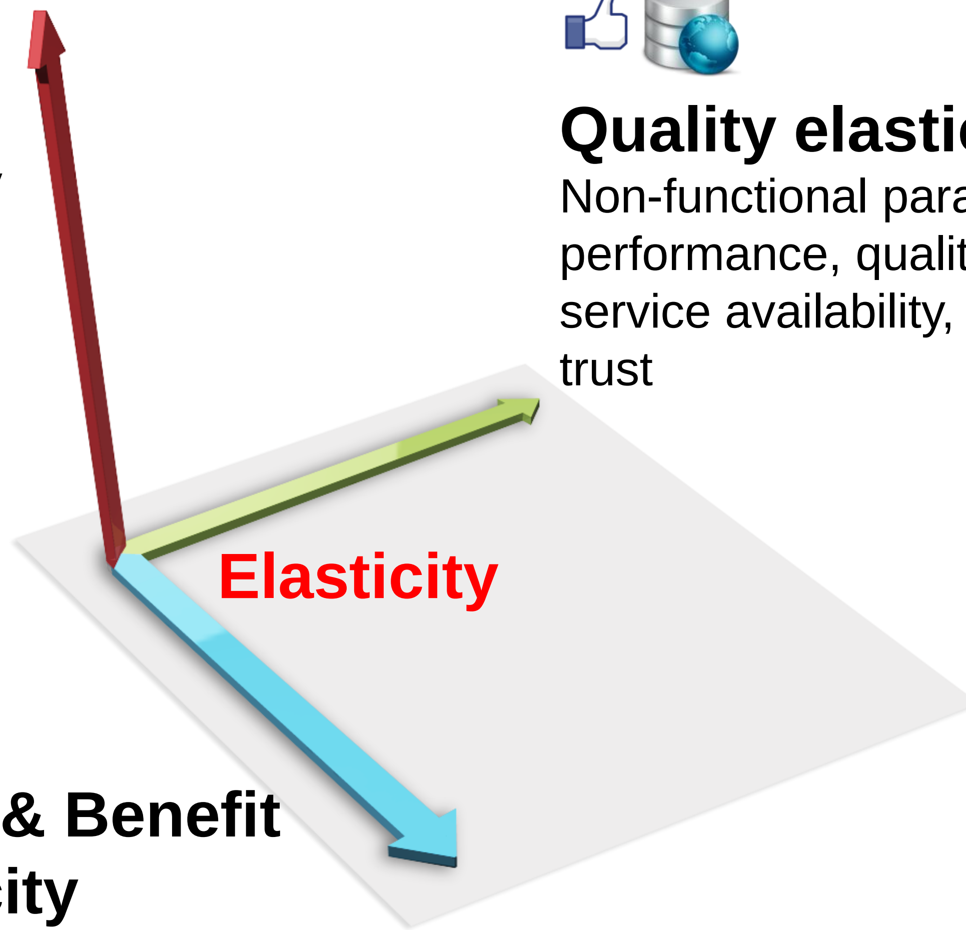
Core principles:

- 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)

Elasticity in computing – broad view

1. Demand elasticity

   Elastic demands from consumers

2. Output elasticity

   Multiple outputs with different price and quality

3. Input elasticity

   Elastic data inputs, e.g., deal with opportunistic data

4. Elastic pricing and quality models associated resources

Diverse types of elasticity requirements

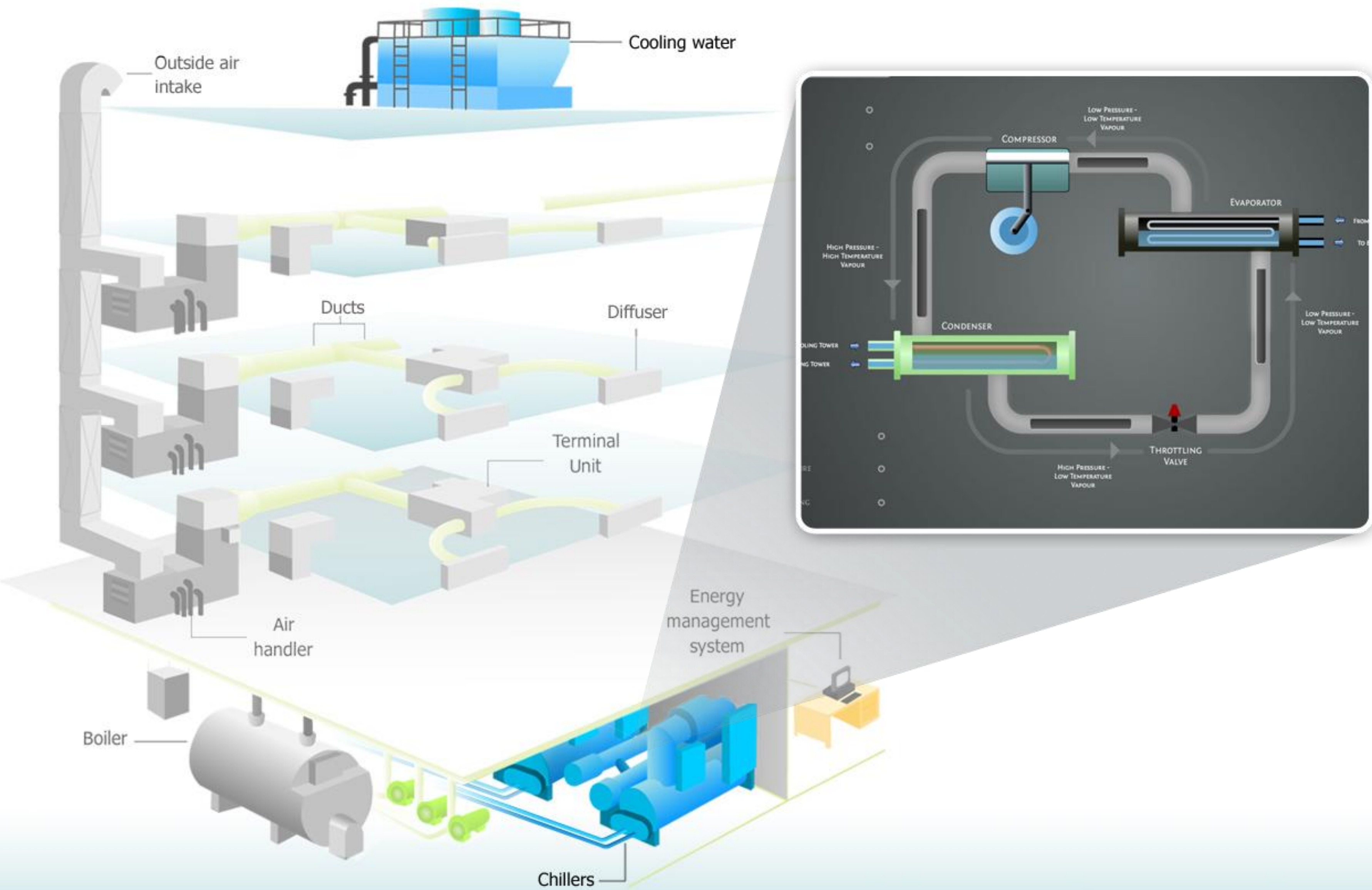
- ***Application user***: "If the cost is greater than 800 Euro, there should be a scale-in action for keeping costs in acceptable limits"
- ***Software provider***: "Response time should be less than amount X varying with the number of users."
- ***Developer***: "The result from the data analytics algorithm 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%."

Internet of Things

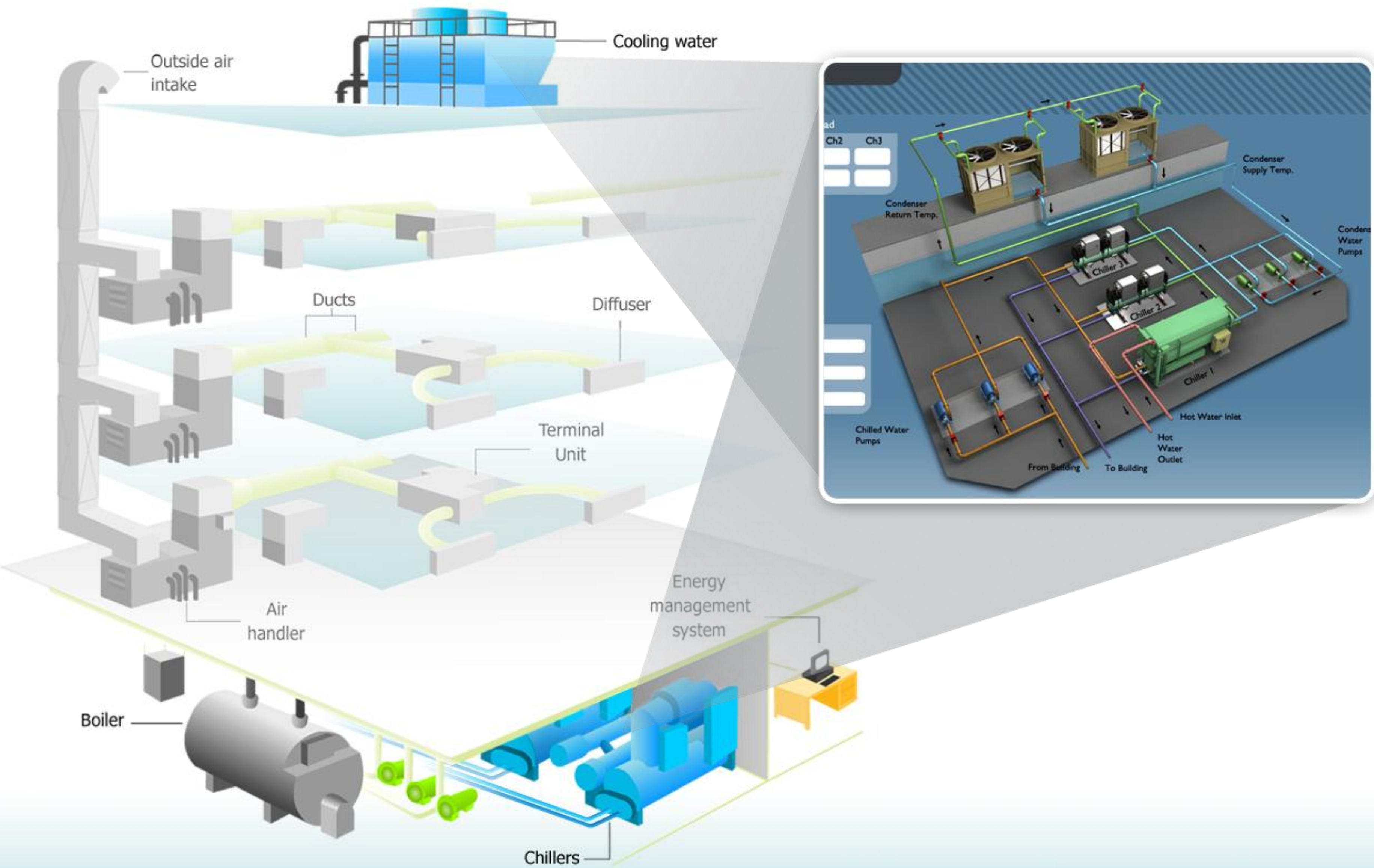
and elasticity


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

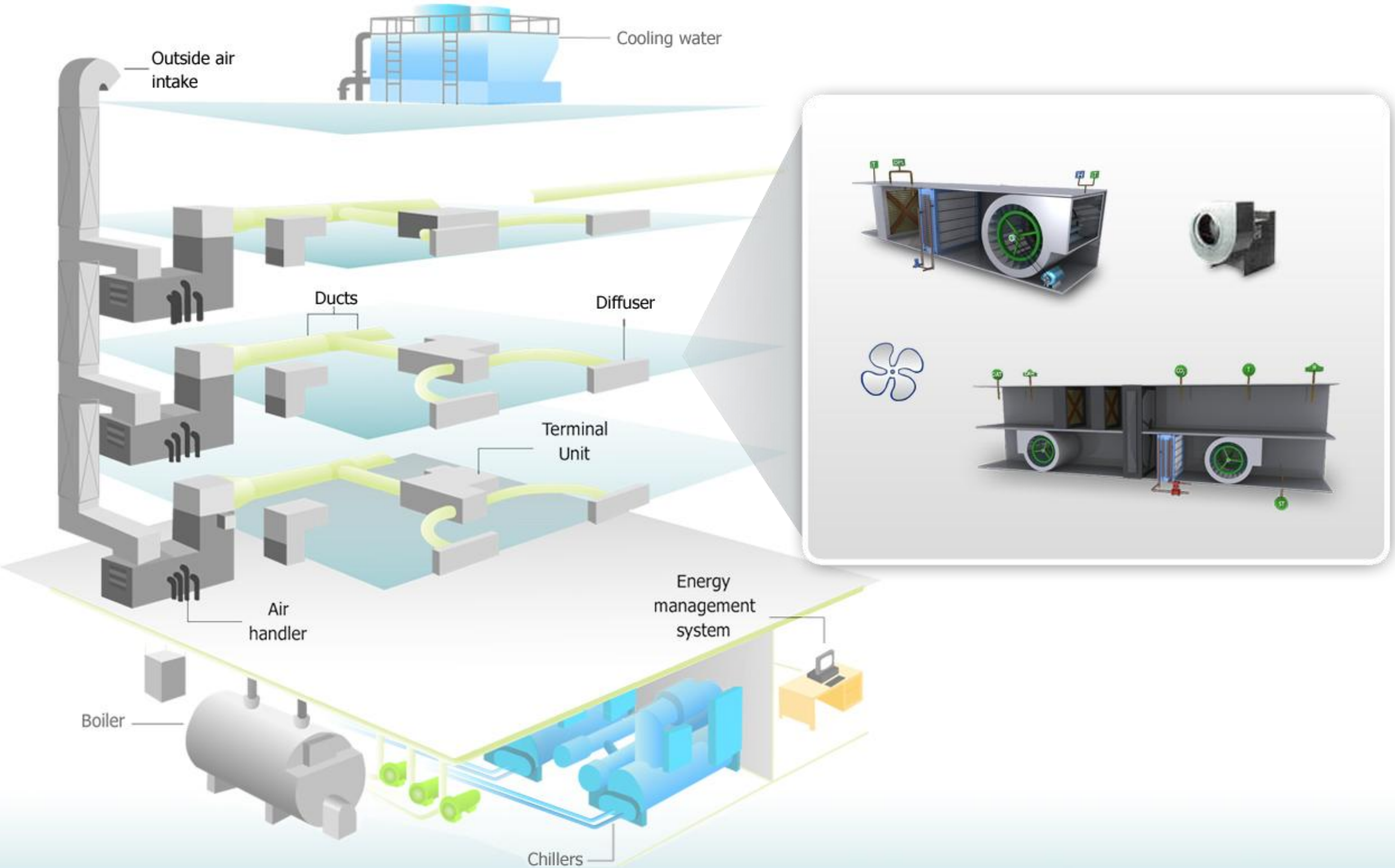
HVAC (Heating, Ventilation, Air Conditioning) Ecosystem



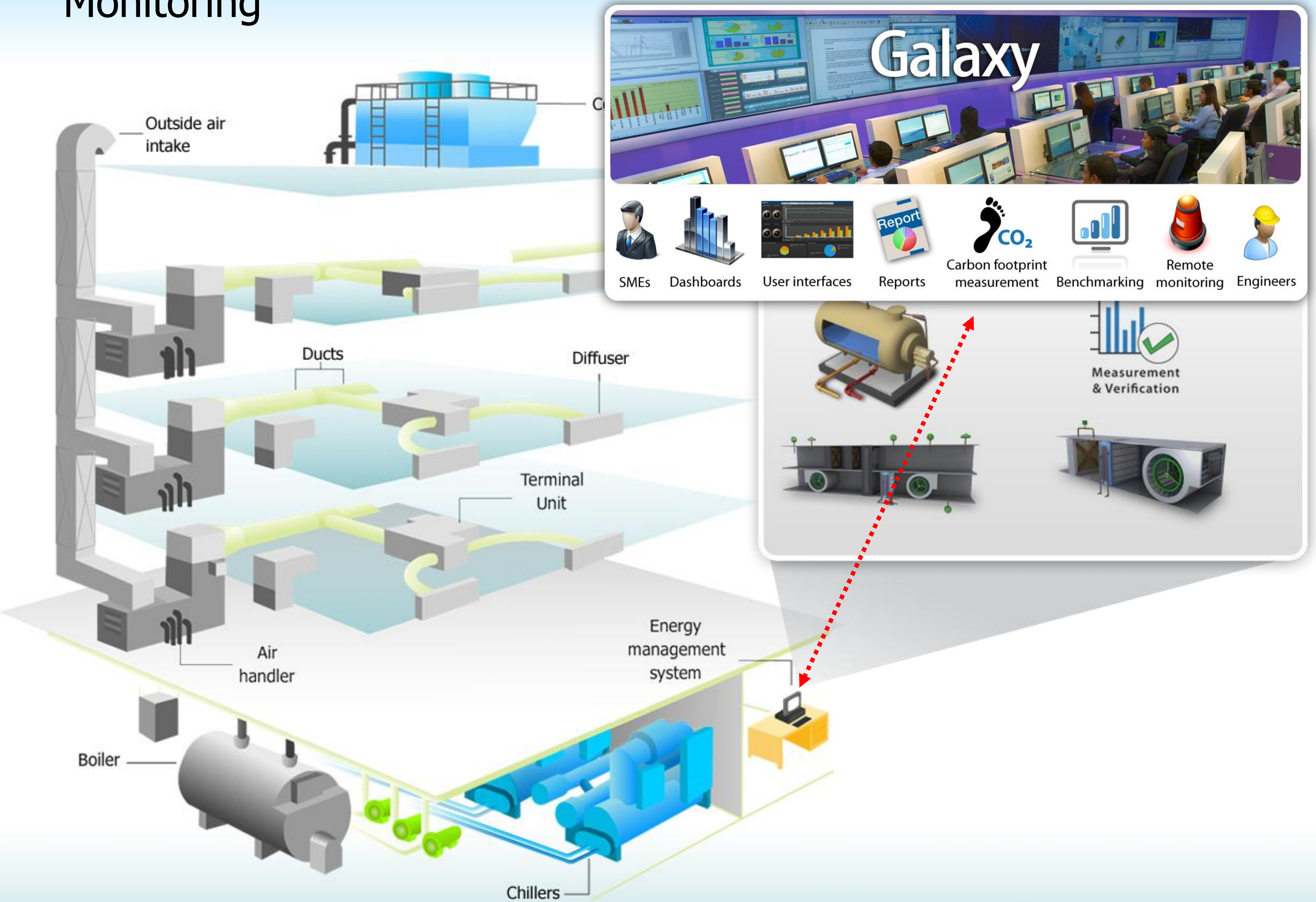
Water Ecosystem



Air Ecosystem

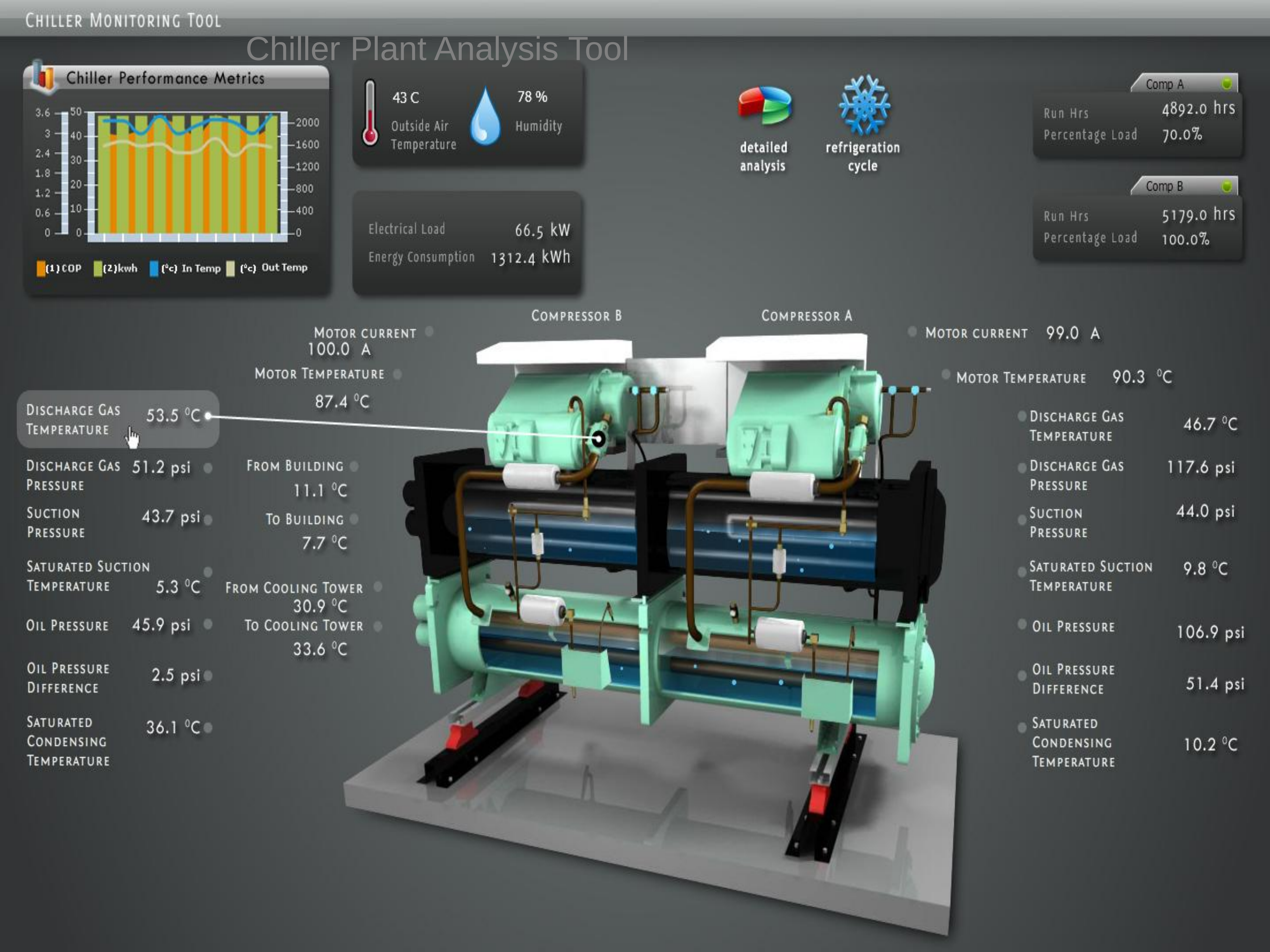


Monitoring



Galaxy
SMEs
Dashboards
User interfaces
Report
Reports
CO_2
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



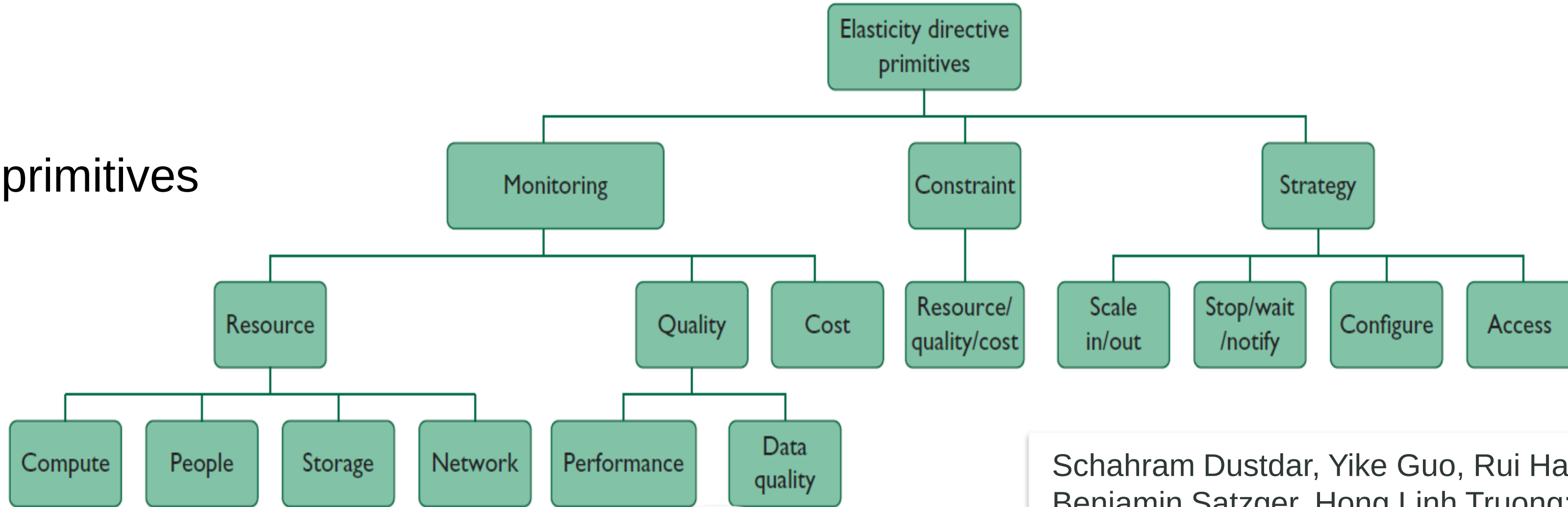
Managed City Governance Service Oriented Architecture

Command Control Center for Managed Services

Elasticity Engineering

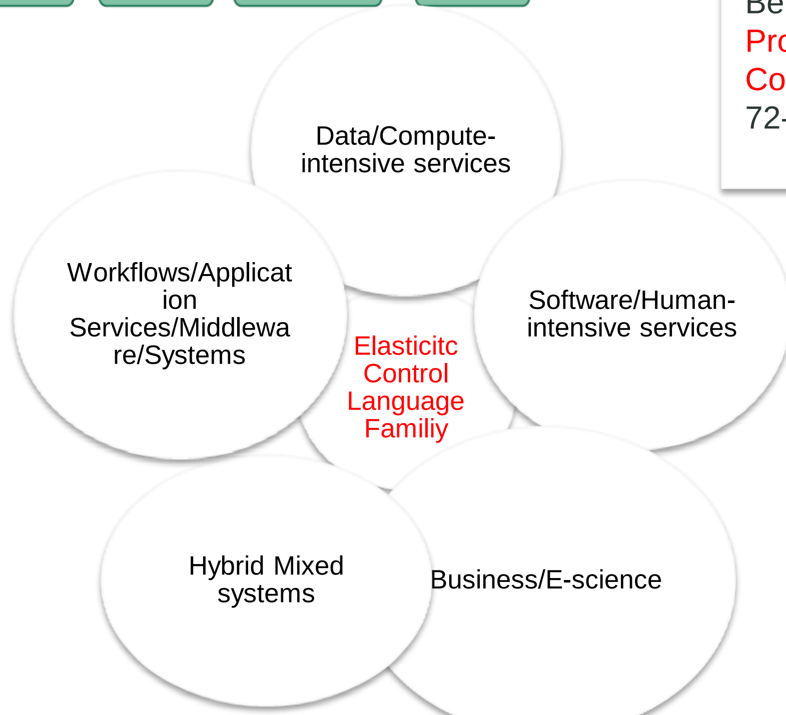
Specifying and controling elasticity

Basic primitives



Schahram Dustdar, Yike Guo, Rui Han, Benjamin Satzger, Hong Linh Truong: Programming Directives for Elastic Computing. IEEE Internet Computing 16(6): 72-77 (2012)

Domain-specific/
Customized features



High Level Description of Elasticity Requirements

- 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

#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

High Level Description of Elasticity Requirements

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>...

Other possibilities

C# Attributes

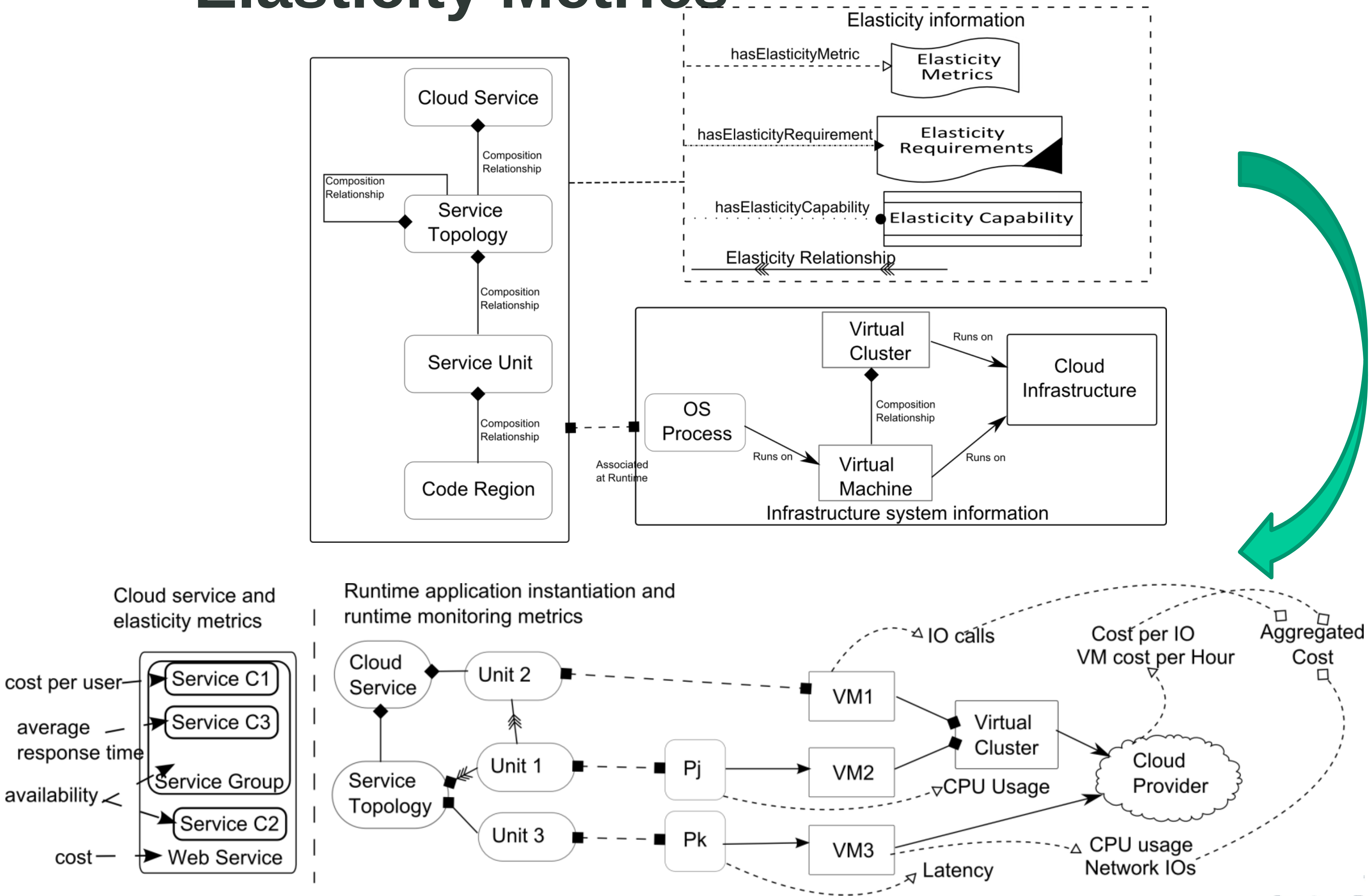
[ProgrammingAttribute(monitoring=„“,constraints=„“,strategies=„“)]

Python Decorators

@ProgrammingDecorator(monitoring,constraints,strategies)

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

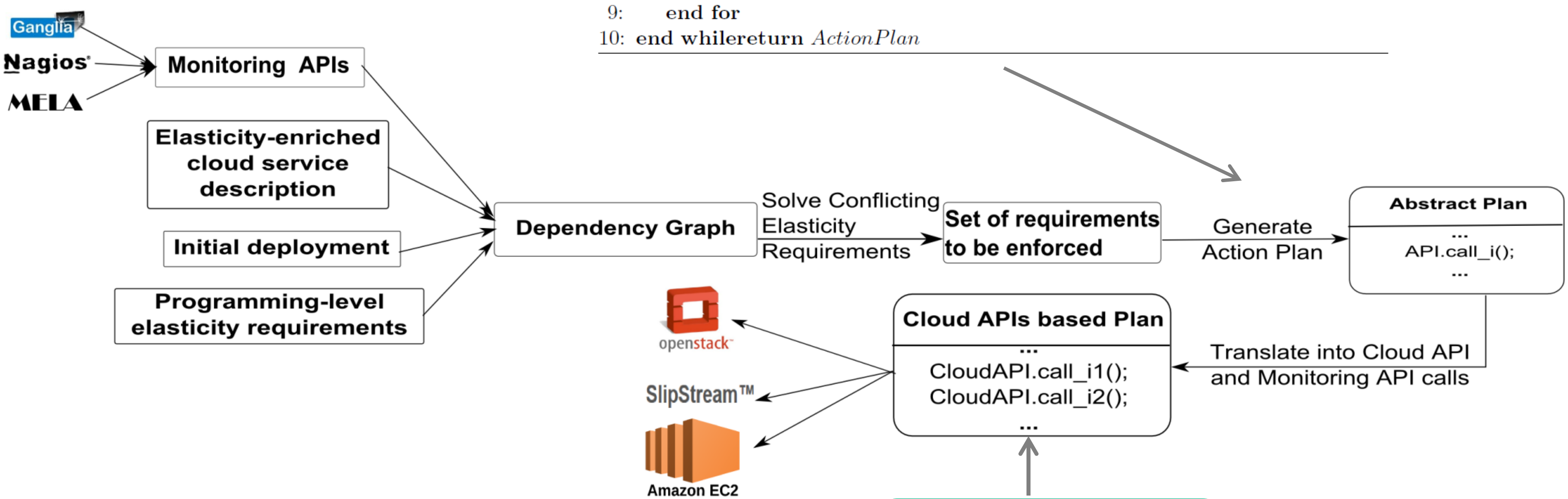


Multi-level Control Runtime:
Generating Elasticity Control Plans

Algorithm 3 Constraint enforcement

Input: *graph* - Cloud Service Dependency Graph
Output: *ActionPlan*

```
violatedConstraints = findAllViolatedConstraints(model)
while size(violatedConstraints) > 0 do
    for each level in cloudServiceAbstractionLevel do
        violatedConstraintsL=selectConstraints(violatedConstraints, level)
        actionSet=evaluateLevelEnabledActions(graph, violatedConstraintsL)
        Action=findAction(actionSet) with max(constraints fulfilled - violated)
        Add action Action to ActionPlan
        violatedConstraints = findAllViolatedConstraints(graph,
estimatedActionEffect(Action))
    end for
end whilereturn ActionPlan
```

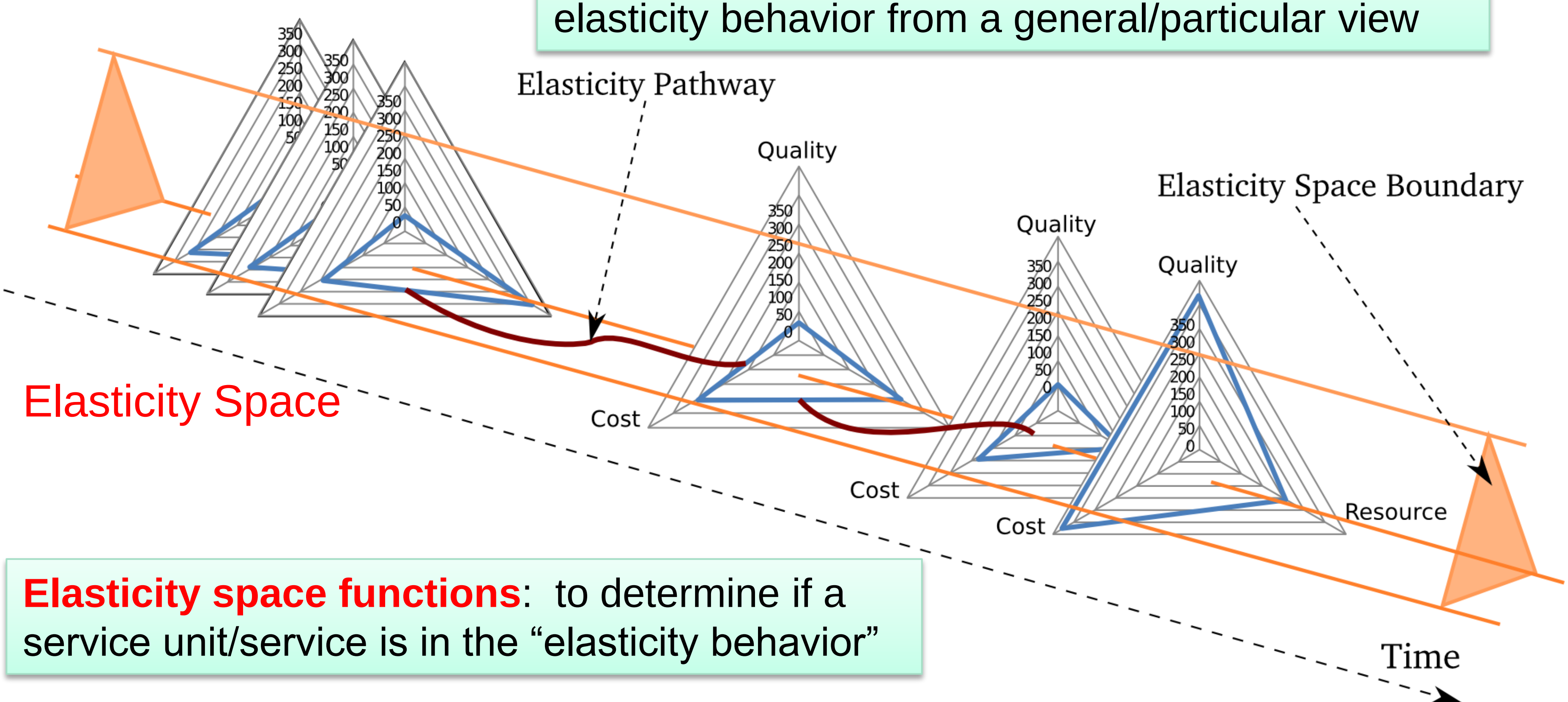


Cloud Providers/Tools must support higher and richer APIs for elasticity controls

Elasticity Model for Cloud Services

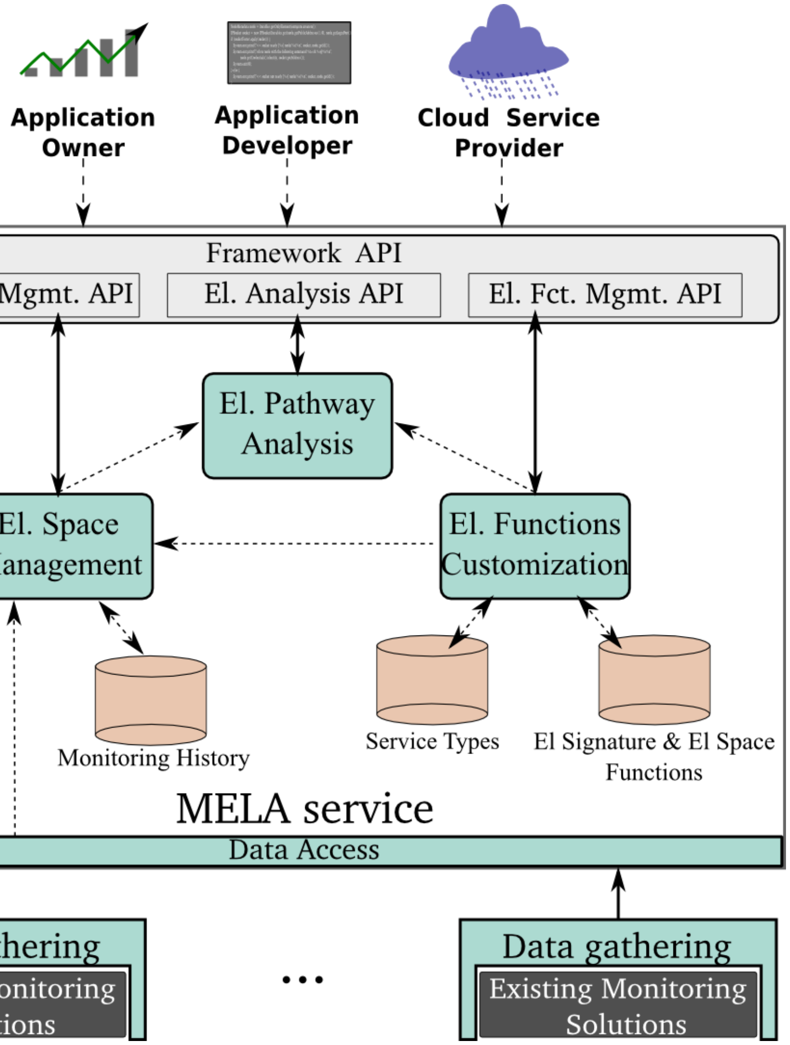
Moldovan D., G. Copil,Truong H.-L., Dustdar S. (2013). **MELA: Monitoring and Analyzing Elasticity of Cloud Service. CloudCom 2013**

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”

MELA -- Elasticity Monitoring as a Service



Daniel Moldovan, Georgiana Copil, Hong-Linh Truong, Schahram Dustdar, **MELA - MELA: Monitoring and Analyzing Elasticity of Cloud Services. CloudCom 2013**

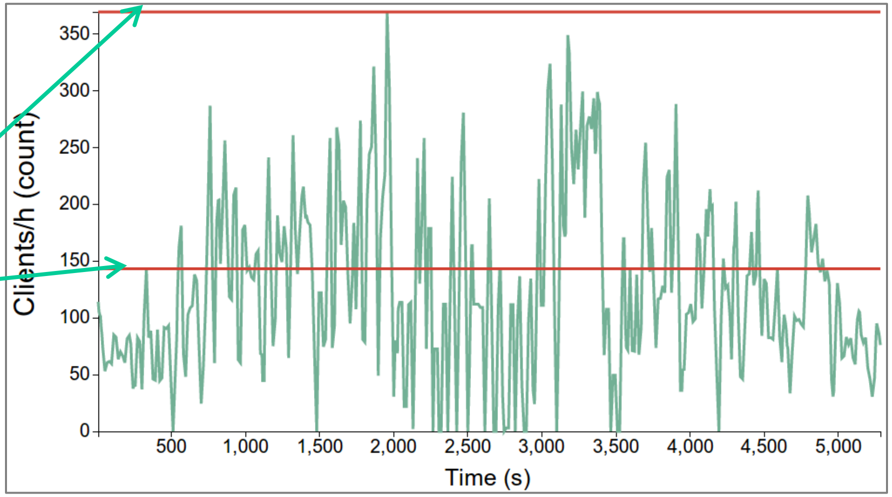
Multi-Level Elasticity Space

Service requirement

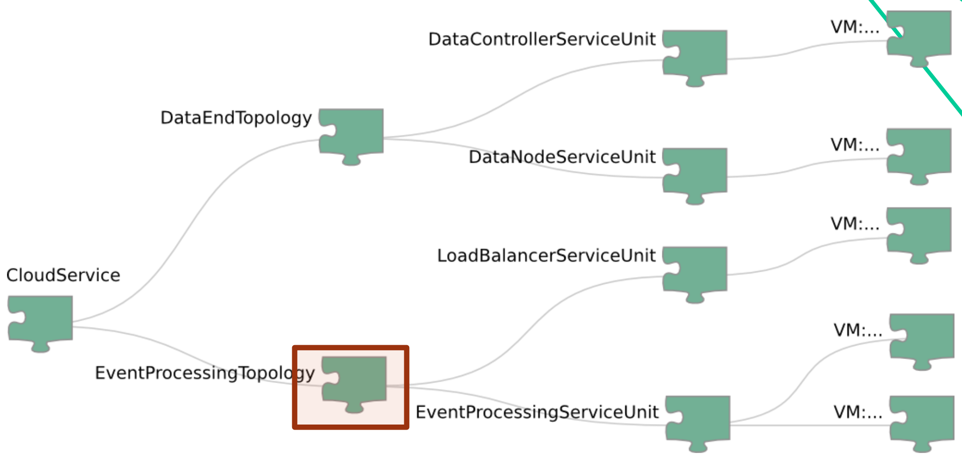
COST<= 0.0034$/client/h

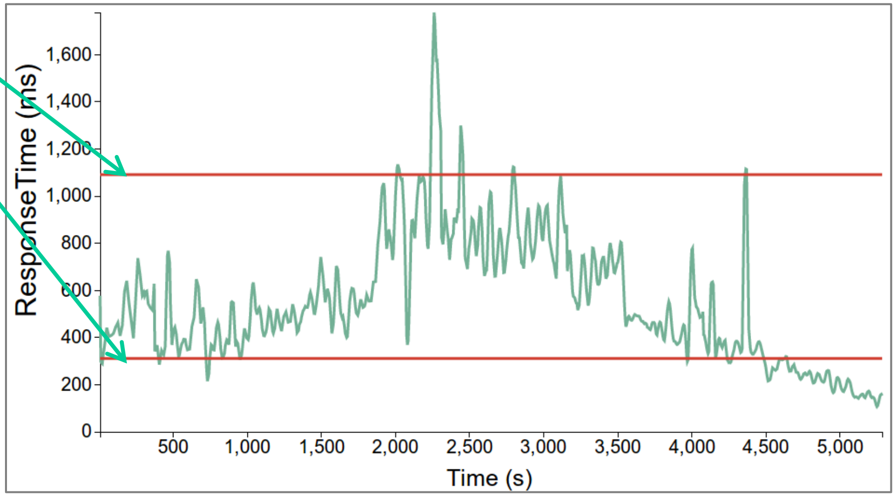
2.5$ monthly subscription for each service client (sensor)

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



Elasticity Space "Clients/h" Dimension



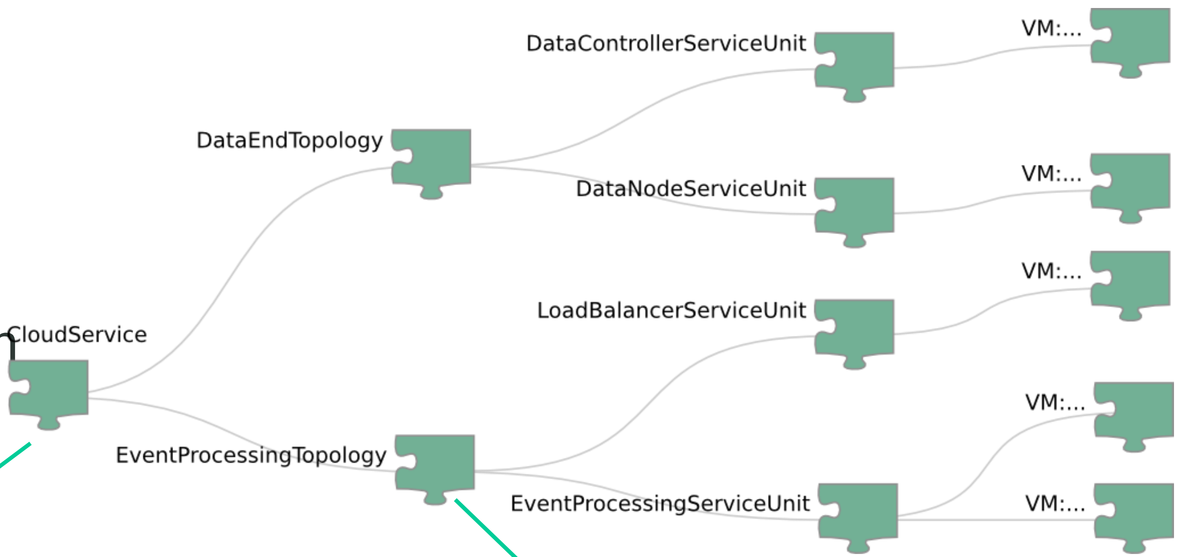


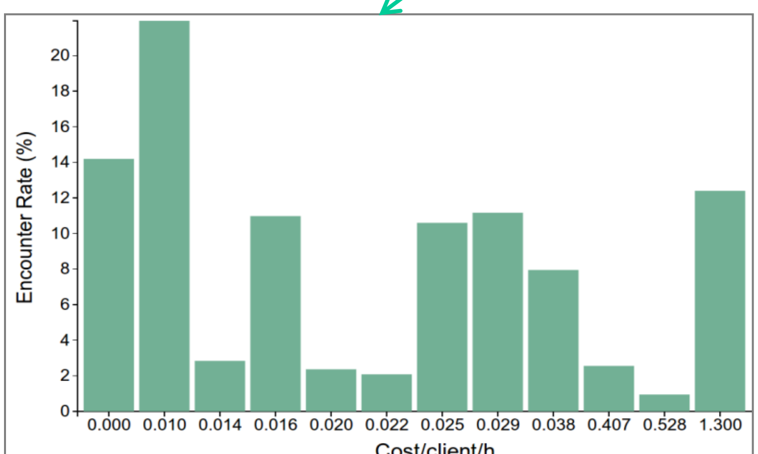
Elasticity Space "Response Time" Dimension

Multi-Level Elasticity Pathway

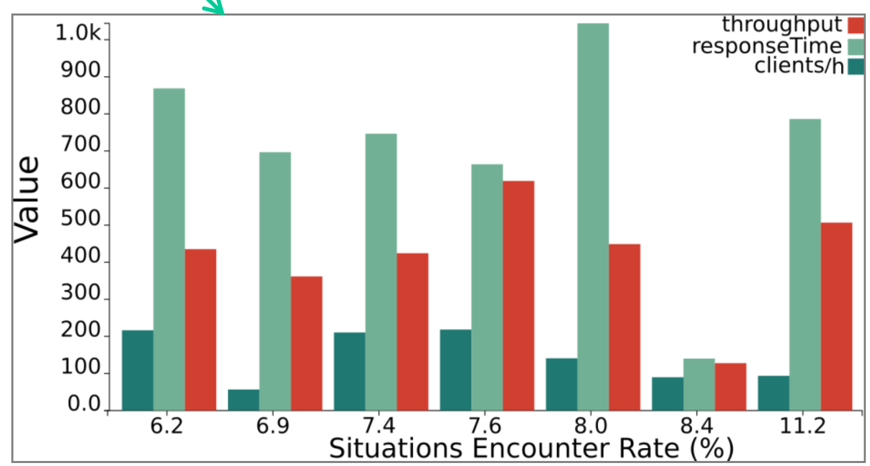
Service requirement

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



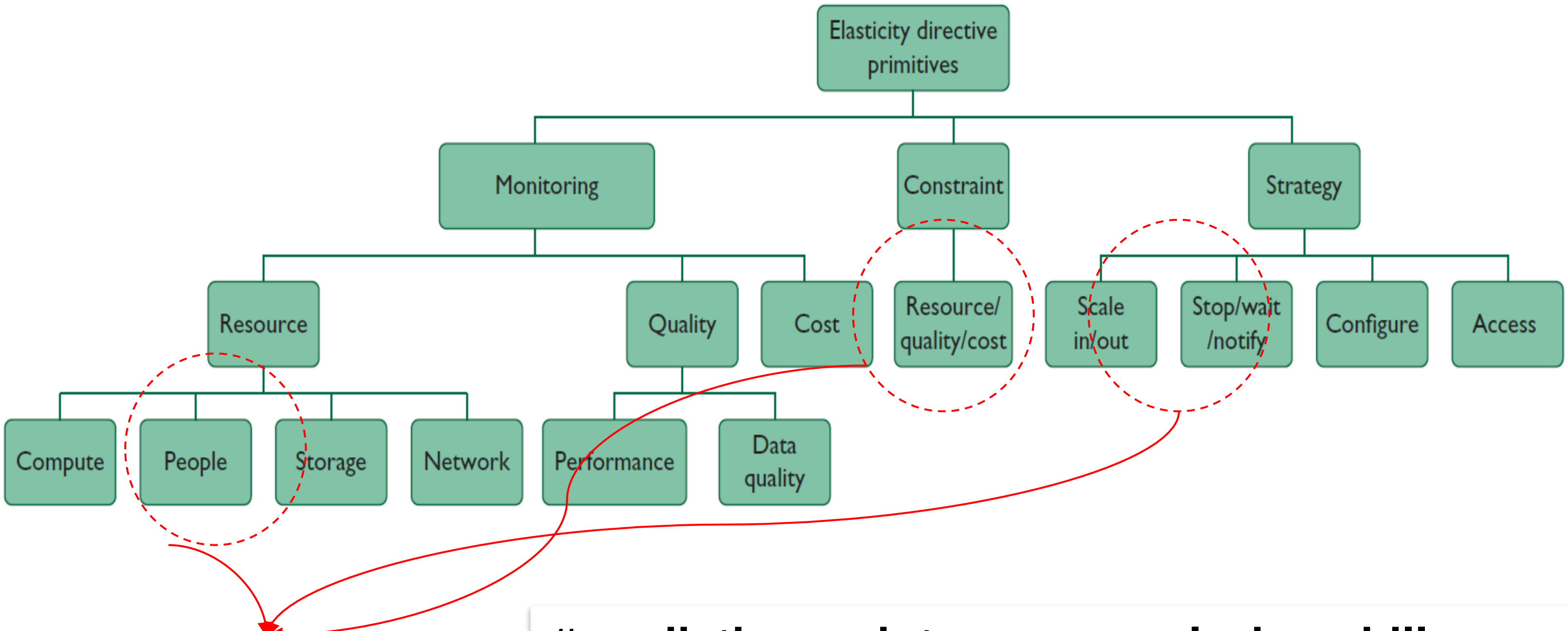


Cloud Service Elasticity Pathway



Event Processing service unit Elasticity Pathway

Specifying and controling elasticity of human-based services



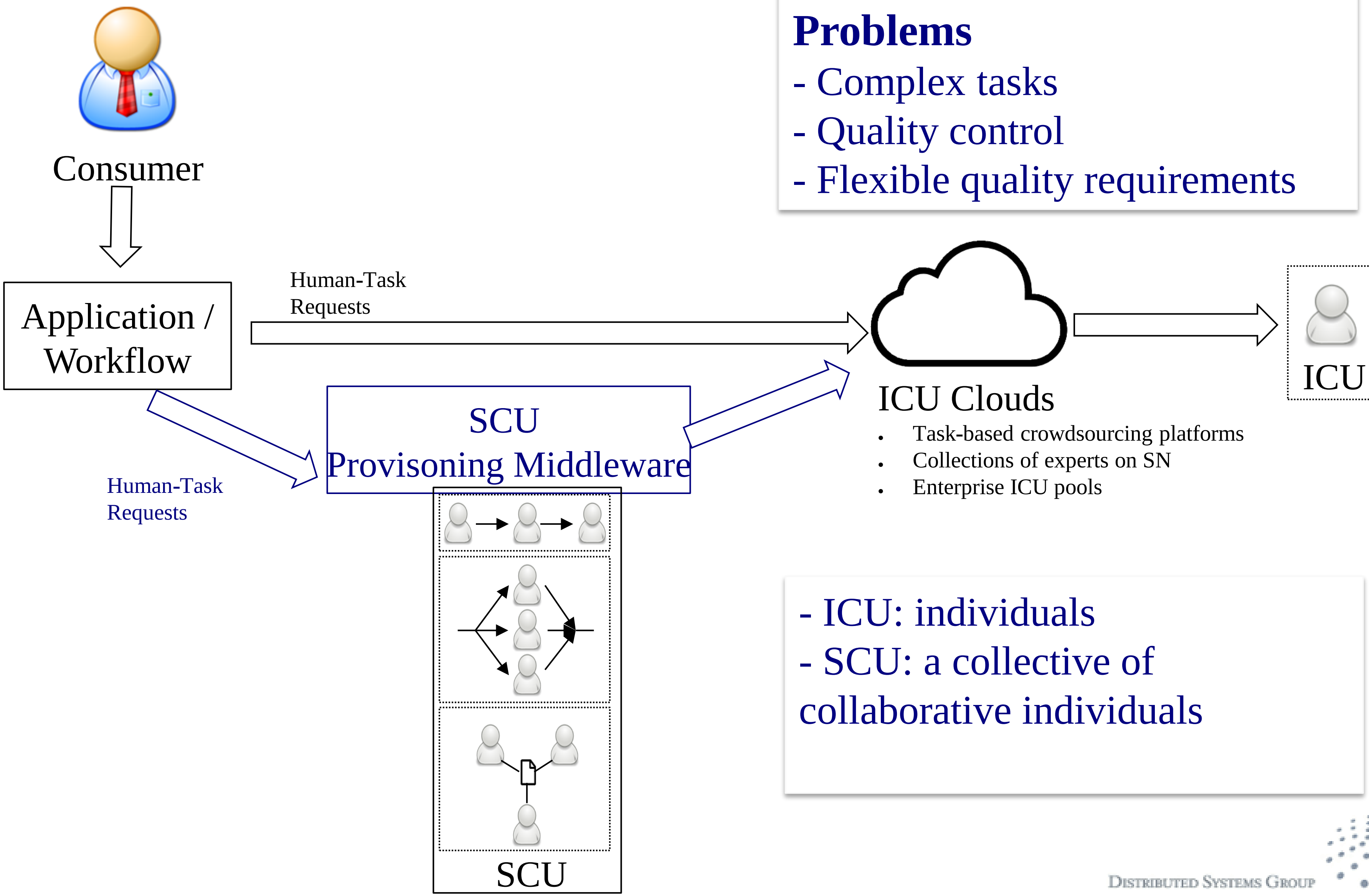
What if we need to invoke a human?

#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)

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?

ICU/SCU for independent tasks



Problems

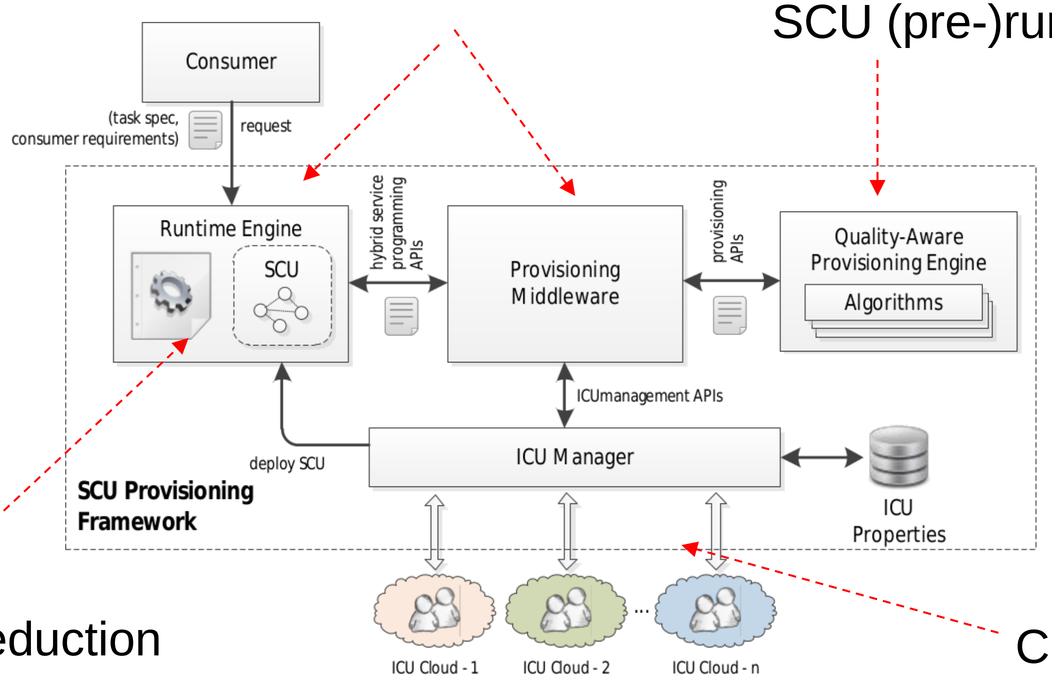
- Complex tasks
- Quality control
- Flexible quality requirements

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

Elastic SCU provisioning atop ICUs

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.**

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

Conclusions (2) – Engineering Elasticity

- Service engineering analytics of elastic systems
  - Programming hybrid compute units for elastic processes
  - Elasticity specifications and reasoning techniques
  - Elasticity space/pathway 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