# Elastic Processes on Clouds: Principles, Research Challenges and Approaches

Hong-Linh Truong

Distributed Systems Group, Vienna University of Technology

truong@infosys.tuwien.ac.at
http://www.infosys.tuwien.ac.at/Staff/truong

Joint work with: Schahram Dustdar, Frank Leymann, Michael Reiter, Marco Comerio, GR Gangadharan, Tran Vu Pham, Flavio De Paoli, Yike Guo, Benjamin Satzger, Uwe Breitenbuecher, Dimka Karastoyanova

# Outline

- Elasticity and elastic processes
- The Vienna Elastic Computing Model
- Enabling techniques for elastic processes in hybrid systems

# A scenario – data intensive processing (1)

- Multiple realtime data sources, e.g.
  - Stream monitoring sensors in smart environments
  - Earth monitoring, satellite images
  - Social media Web information
- A set of data processing algorithms
  - Data conversion, data enrichment, data extraction, data mining, domain-specific computational analysis (e.g., CO2 calculation), etc
  - Algorithms can be wrapped into processing elements/workflow activities

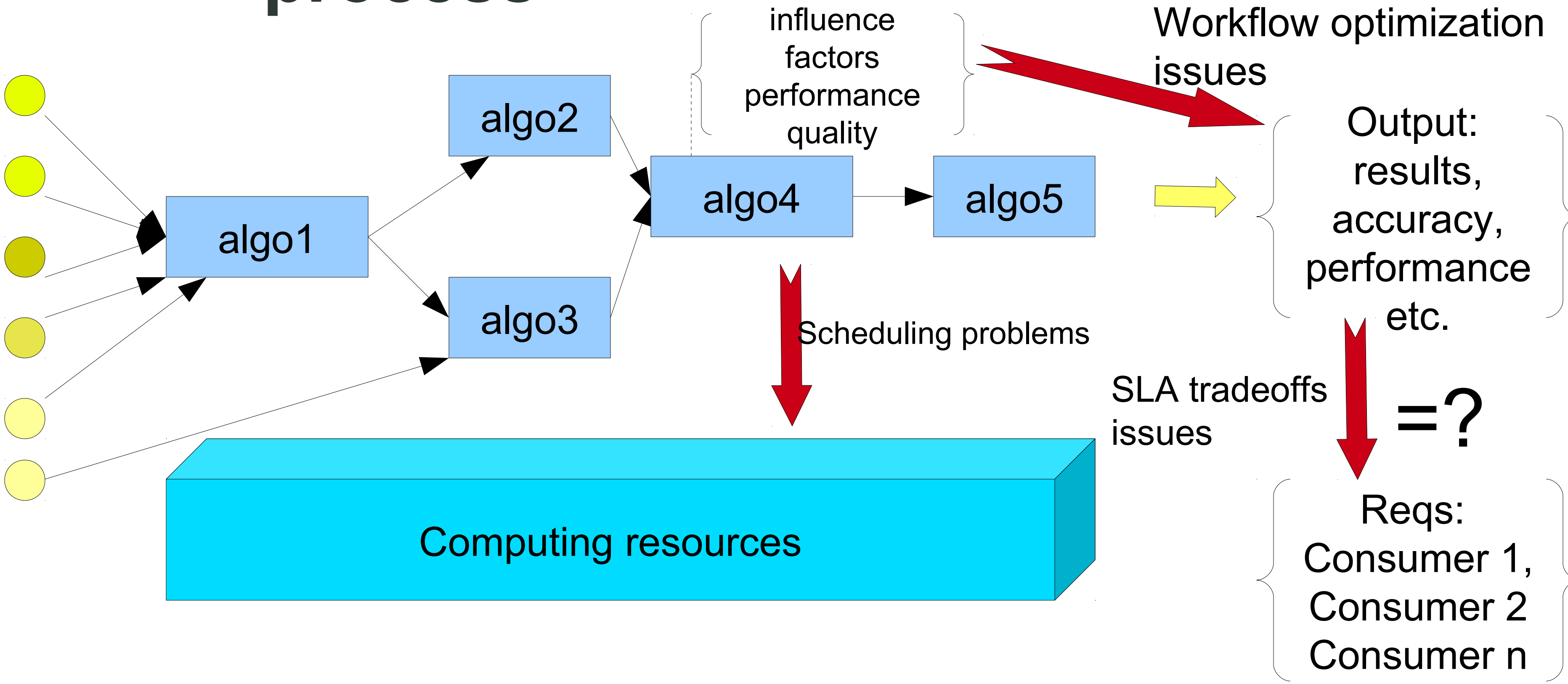
# A scenario – data intensive processing (2)

- Influence factors for the performance and quality of algorithms
  - Data rates, quality of data sources, underlying systems, etc.
- The quality of the output of algorithms is varying
- Multiple concurrent consumers
  - Dynamic requirements for SLA, quality of outputs, etc.

Data processing service provider: my process must be elastic!

- to optimize resource usage, prices, etc., but must meet consumer's requirements

# Example of a business/e-science process




- One processing workflow for all consumers
  - Scheduling, SLA tradeoffs, and individual algorithm impact on the whole processing chain

# What do we know about “elasticity”?

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.

Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete

## Elasticity

From Wikipedia, the free encyclopedia

**Elasticity** may refer to:

- Elasticity (physics), continuum mechanics of bo

Numerous uses are derived from this physical sense of the term, which is inherently mathematical, Construction and variously in Economics:

- Elasticity (data store), the flexibility of the data model and the clustering
- Elasticity (economics), a general term for a ratio of change. For more specific economic forms of
  - Beta coefficient
  - Cross elasticity of demand
  - Elasticity of substitution
  - Frisch elasticity of labor supply
  - Income elasticity of demand
  - Output elasticity
  - Price elasticity of demand
  - Price elasticity of supply
  - Yield elasticity of bond value
- Elasticity (mathematics), a mathematica
  - Arc elasticity

Cloudscaling Blog

**Elasticity is NOT #Cloud Computing … Just Ask Google**

Posted on November 1, 2010 by randybias



VITALAB VIENNA INTERNET TECHNOLOGIES ADVANCED RESEARCH LAB
Distributed Systems Group

# Elasticity in computing

„Elastic computing is the use of computer resources which vary dynamically to meet a variable workload" – http://en.wikipedia.org/wiki/Elastic_computing

„Clustering elasticity is the ease of adding or removing nodes from the distributed data store" – http://en.wikipedia.org/wiki/Elasticity_(data_store)

„What elasticity means to cloud users is that they should design their applications to scale their resource requirements up and down whenever possible.", David Chiu – http://xrds.acm.org/article.cfm?aid=1734162

# Elasticity in physics and economics

„elasticity (or stretchiness) is the physical property of a material that returns to its original shape after the stress (e.g. external forces) that made it deform or distort is removed" – http://en.wikipedia.org/wiki/Elasticity_(physics)

„elasticity is the measurement of how changing one economic variable affects others" – http://en.wikipedia.org/wiki/Elasticity_(economics)

- What are "stress" and "strain" for the structure of processes
  - Workload? Quality?
  - Process activities? Workflow fragments?
- Which are economic variables of processes
  - Price? Consumer request?

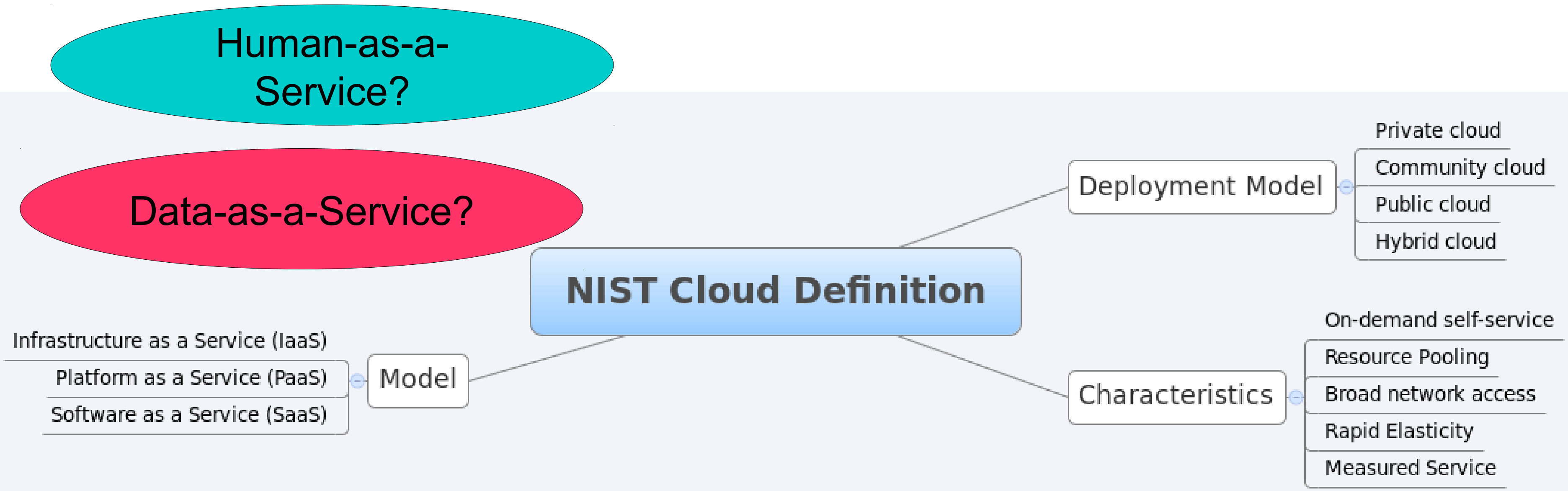
# Elasticity in computing – a broad view

- Elastic demands from consumers
- Multiple outputs with different price and quality (output elasticity)
- Elastic data inputs, e.g., deal with opportunistic data (scaling in/out of data inputs)
- Elastic pricing and quality models associated resources

The service/process itself is elastic in terms of economic theory

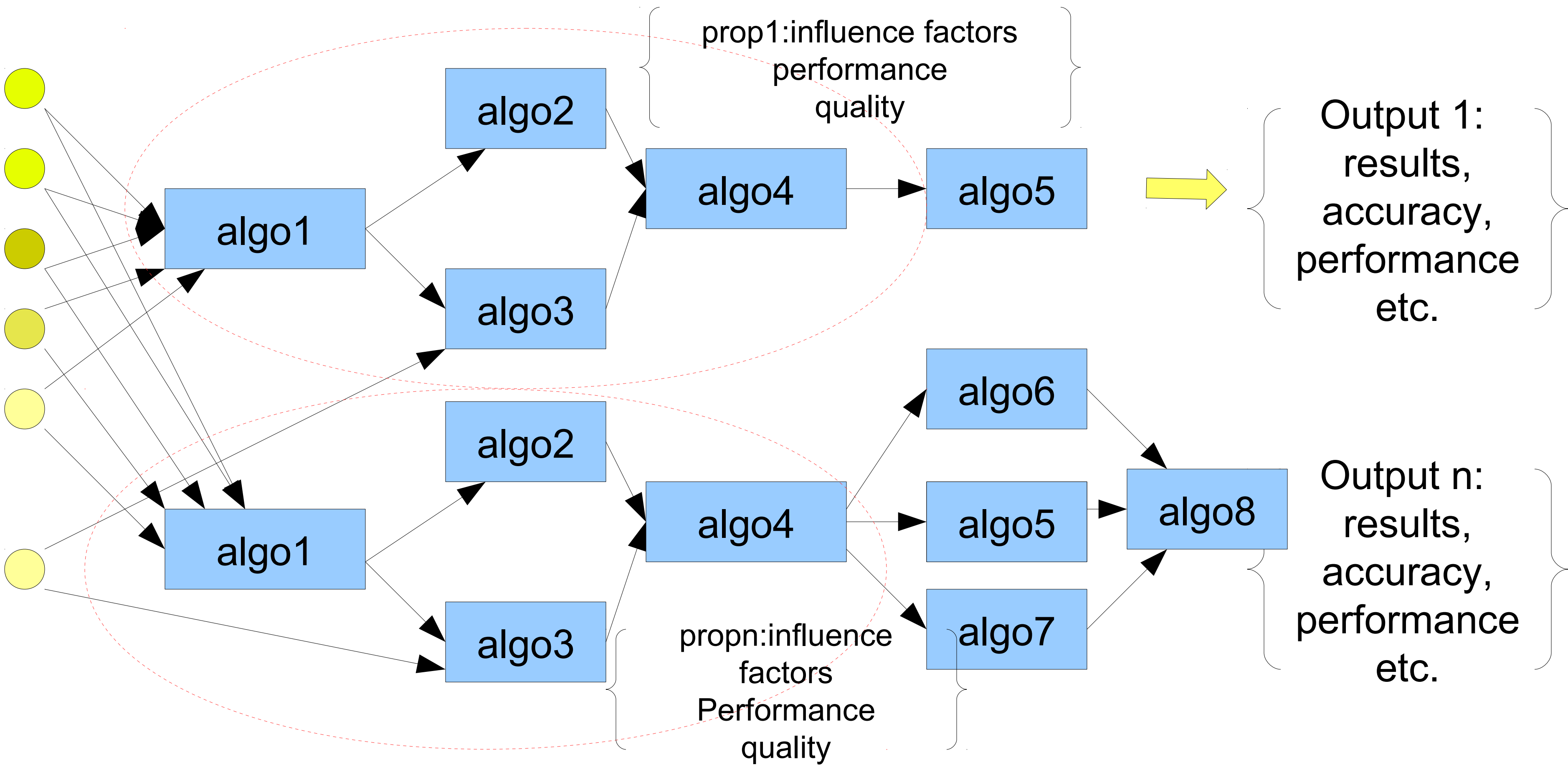
The service/process structure is elastic in terms of physics

# NOT just getting more hardware resources!




Source: NIST Definition of Cloud Computing v15, www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf

# Example of elasticity in clouds (1)

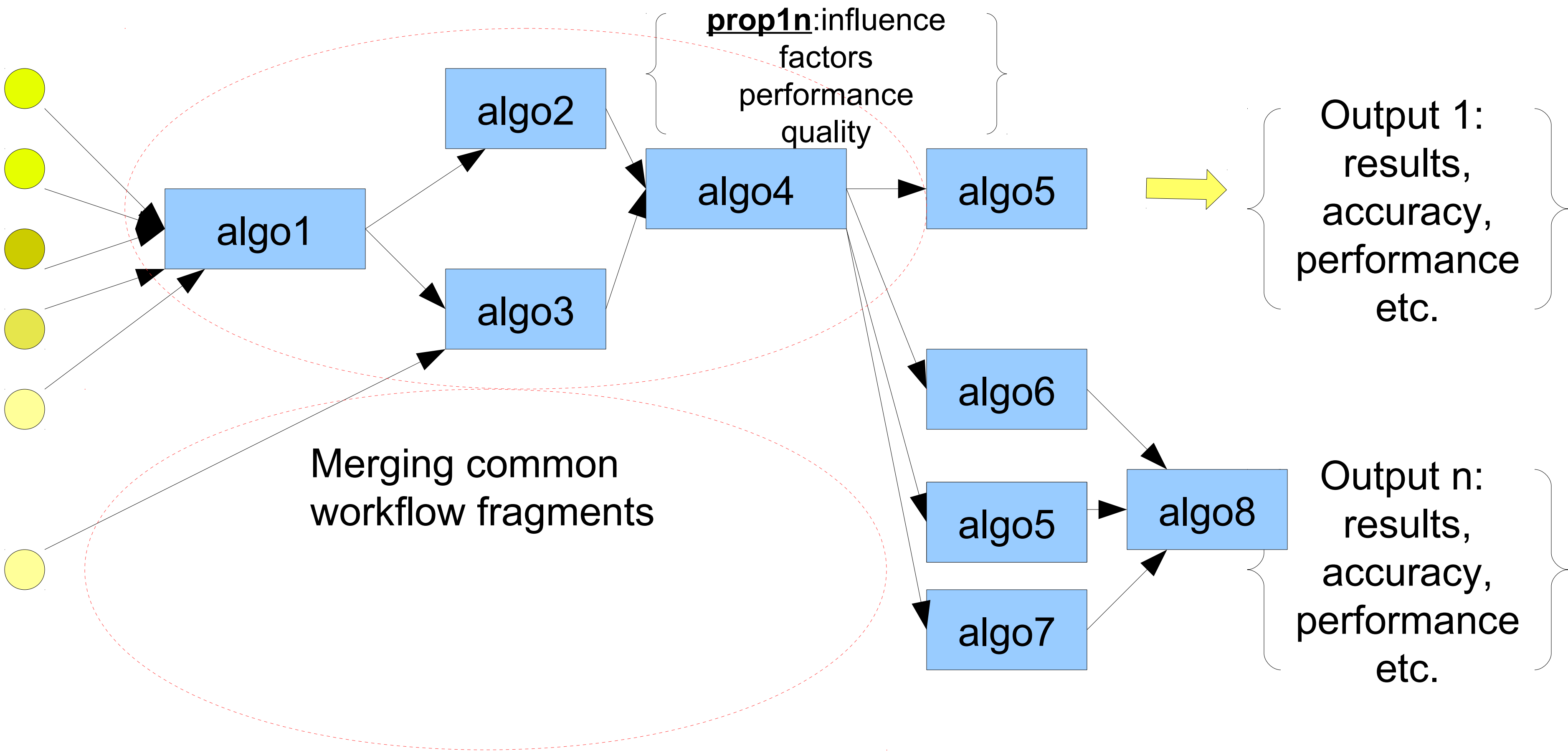



- N separated workflows for M consumers
  - Common activities but different influence factors/expected quality



VITALAB Vienna Internet Technologies Advanced Research Lab Distributed Systems Group

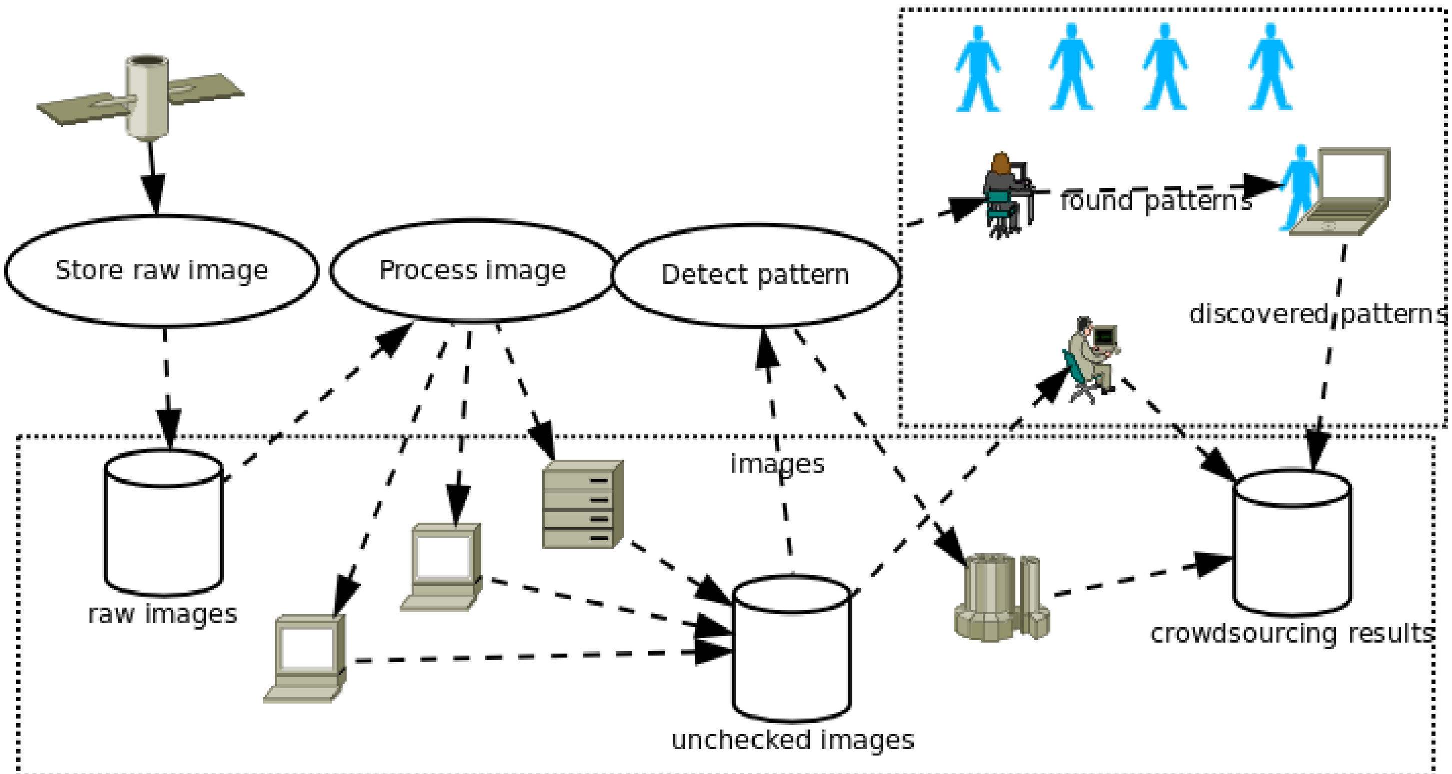
# Example of elasticity in clouds (2)




- Merging common workflow fragments
- Dynamic refinements

# Example of elasticity in clouds (3)

- Jobs of software-based computing elements (SEs) and human-based computing elements (HEs) must be integrated into a single system
- Quality constraints are important: not just volunteering work!

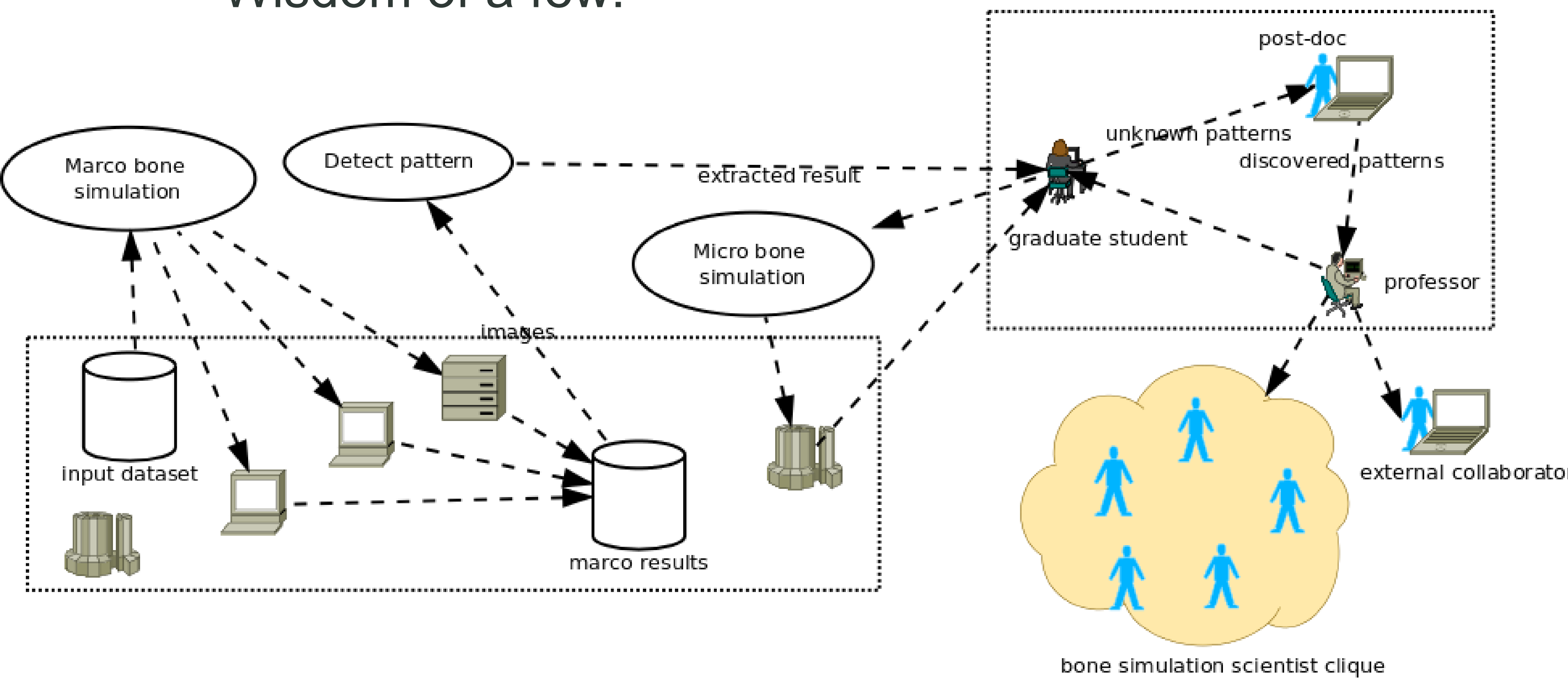





VITALAB Distributed Systems Group

# Example of elasticity in clouds (4)

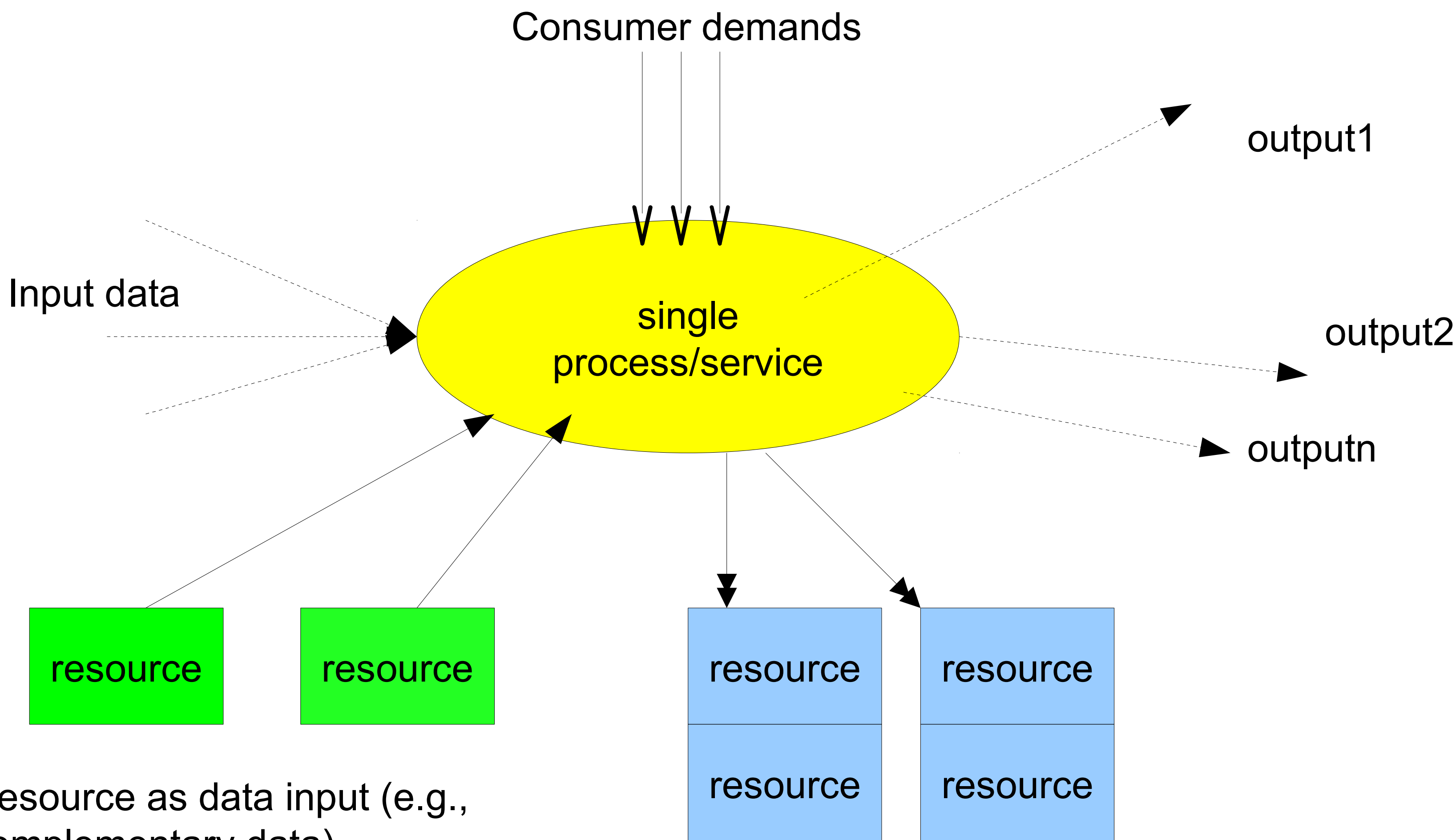
- You need more processional teams/cliques
  - Wisdom of a few!



## But what are principles of Elastic Processes?

"Schahram Dustdar, Yike Guo, Benjamin Satzger, Hong Linh Truong: Principles of Elastic Processes. IEEE Internet Computing 15(5): 66-71 (2011)"

# Basic view




Resource as data input (e.g., complementary data)
Human-as-a-sensor

Resources as executable hosts (e.g., machine, human, software)

# Elastic Properties – price elasticity (1)

Price elasticity describes a resource/service's provision's responsiveness to change in price

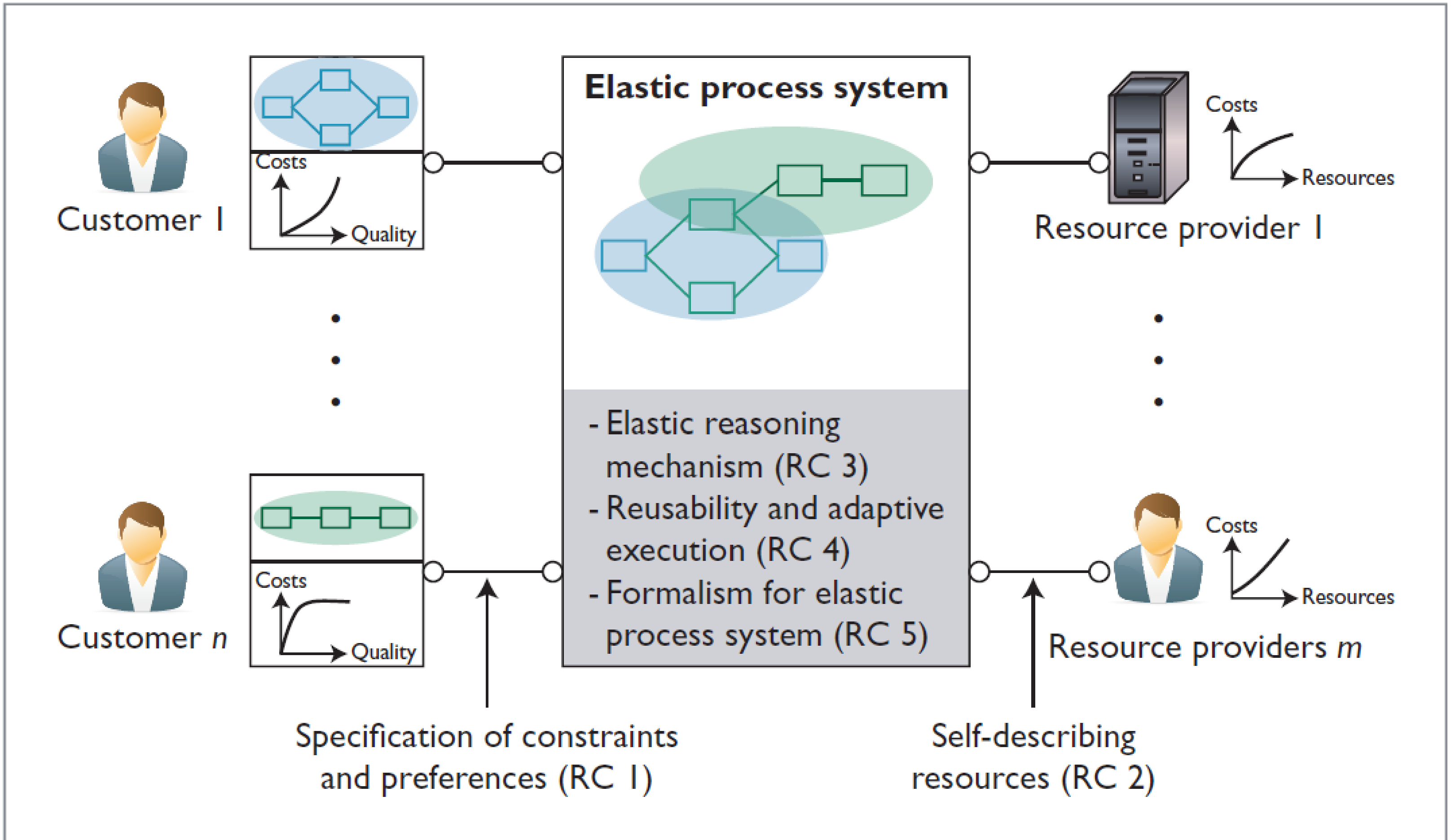
- Price items: investment, provisioning, maintenance, etc.
- Dynamic pricing models can be built based on elasticity concept
  - e.g., Price elasticity of demands
  -
- Even with fixed price/quality resources
  - Utilize different competitive resources to offer price elasticity and quality elasticity

# Elastic Properties – quality elasticity

Quality elasticity measures how responsive quality is to a change in resource usage

- Service's quality improvement should be monotonic to the consumption of resources needed
  - Adding more resources does not automatically increase the quality!
- Multiple quality dimensions
  - Performance, quality of data, etc.
- Complex dependencies among quality measurement and price function

# Conceptual architecture of elastic process environment



# Conceptual model: physical elasticity properties

- EP must optimize resource usage
  - Dynamic environment with diverse resource types
  - Dynamic resources

    → scaling in/out of dynamic resources (inputs and underlying execution environments)
- EP can produce multiple outputs
  - For demands with similar requirements

    → scaling in/out of multiple outputs and "deformation" of the structure of EP

# Conceptual model: economic elasticity properties

- EP's function is a static property
  - Accepts inputs and produces outputs
- Physical elastic properties of EP lead to EP's economic elastic properties
  - Resource elasticity: several instances of EP can be provided elastic
  - Price elasticity: each instance of EP can have different pricing models
  - Quality elasticity: each instance of EP can offer different quality metrics

# Conceptual model: operation and modeling principles (1)

- Operation principles
  - Monitor, manage and describe dynamic properties
  - Dynamically refine process functions based on quality
  - Determine costs based on multiple resource cost models
  - Provide elasticity across providers
- An EP can deal with multiple service objectives
  - N concurrent consumers, access markets of M providers, with K different requirements: K <= N

# Conceptual model: operation and modeling principles (2)

- Modeling
  - Overlaying EPs, functional composition, and dynamic property composition
- Modeling principles
  - EP's function as a static property
  - EP's results based on requirements concerning price and quality – modeled as a set of constraints influencing resource elasticity
  - Communication can be based on SOAP/REST
  - Refinement and composition of resource, price and quality at multiple levels: activities, fragments, or the whole EP

# Research challenges

- Specification of constraints and preferences
  - How to enrich processes with constraints and preferences specifying pricing and preferences
  - How to allow relevant stakeholders to control trade-offs
- Self-describing resources
  - Diverse types of resources: hardware, software and humans
- Elastic reasoning mechanism
  - How to deal with trade-offs and realtime reasonings
- Reusability and adaptive execution
  - Resoures and fragments refinements based on price and quality
- Formalism for elastic process systems

# To be elastic in hybrid systems: the Vienna Elastic Computing Model (VCM)

# VCM – elasticity in hybrid systems

| E-science Application | Business Application |
|---|---|

- Elastic Service
- Elastic Alignment
- IT Elastic Process
- Elastic Runtime Management
- IT Elastic Service
- Virtualization

→

- Service modeling and interface
- Alignment techniques
- Virtualization techniques
- Monitoring and analysis techniques
- Non-functional parameters characterization
- Programming models
- Governance and service evolution

| Software-based Computing Element (SE) | Human-based Computing Element (HE) |
|---|---|
| SpotCloud, amazon webservices | amazon mechanical turk, LinkedIn, Facebook |

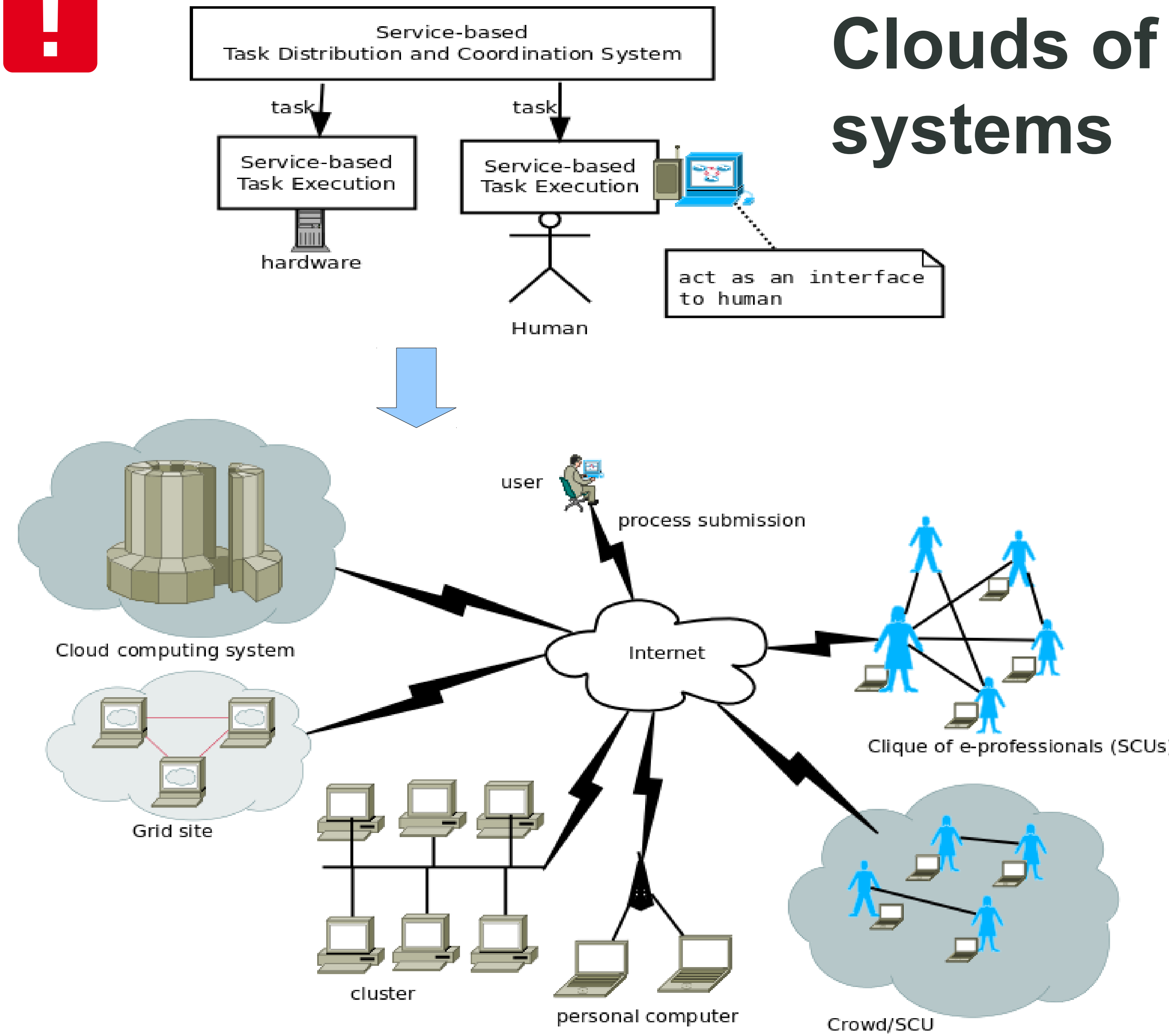
→

- Heavy crowded in cloud computing and social computing
- Ready to use but in isolation (single cloud and single type of computing elements)



VITALAB Distributed Systems Group

# Clouds of hybrid systems

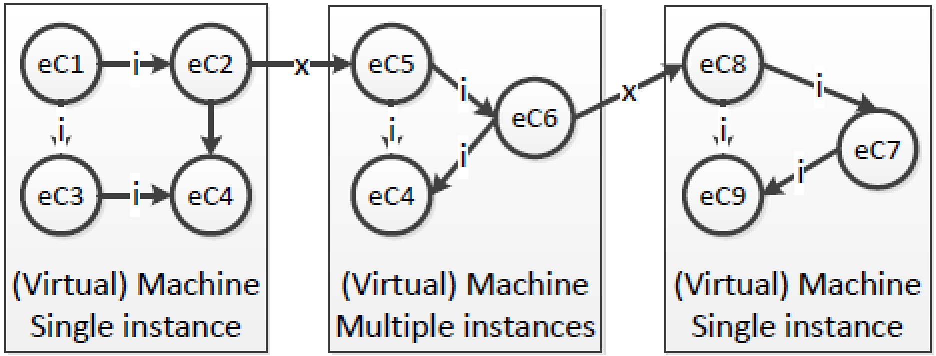


# The list of challenges

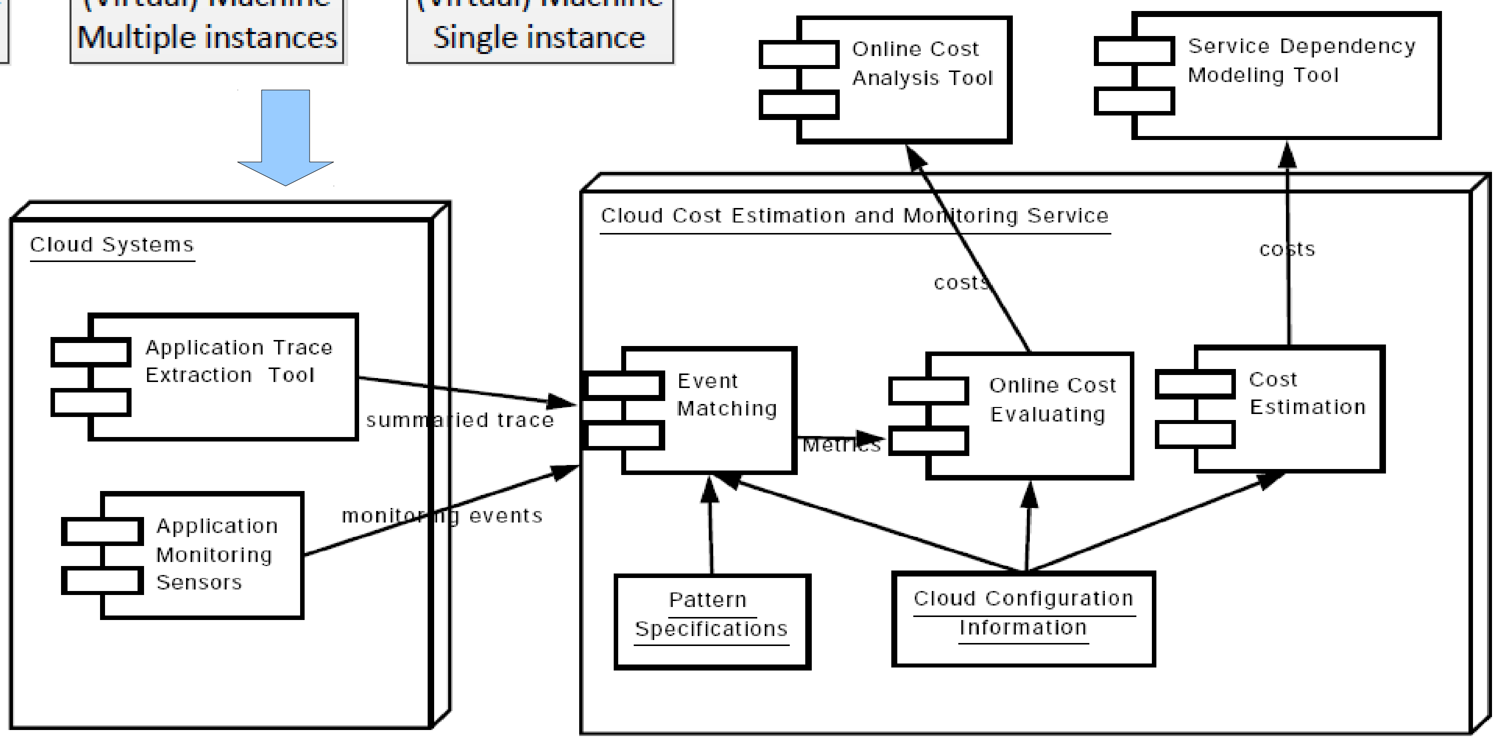
- System software and middleware
- Elastic service modeling and alignment
- Sevice evolution
- Programming language and Tools
- Dynamic refinement and adaptation
- Charactering and monitoring non-functional parameters
- Rewarding mechanisms
- Governance and compliance

Which are enabling techniques that can be used and challenges?

# Composable cost evaluation




Tran Vu Pham, Hong-Linh Truong, Schahram Dustdar "Elastic High Performance Applications - A Composition Framework", The 2011 Asia-Pacific Services Computing Conference (IEEE APSCC 2011), (c) IEEE Computer Society, December 12 - 15, 2011, Jeju, Korea




Hong Linh Truong, Schahram Dustdar: Composable cost estimation and monitoring for computational applications in cloud computing environments. Procedia CS 1(1): 2175-2184 (2010)

# Examples of fine-grained composition cost models

| Model | Activities | Cost |
|---|---|---|
| $M_{ds}$ | Data storage | $size(total) \times t_{sub} \times cost(storage)$ where $t_{sub}$ is the subscription time |
| $M_{cm}$ | Computational machine | $cost(machine)$ |
| $M_{dfi}$ | Data transfer into the cloud | $cost(transfer_{in})$ |
| $M_{dfo}$ | Data transfer out to the cloud | $cost(transfer_{out})$ |
| $M_{sd}$ | Single data transfer without the cost for machines performing the transfer | $size(in) \times M_{dfi} + size(out) \times M_{dfo}$ |
| $M_{sm}$ | Sequential/multi-threaded program or single data transfer with the cost for machines performing the transfer (cost monitoring) | $t_e \times M_{cm} + size(out) \times M_{dfo} + size(in) \times M_{dfi}$ |
| $M_{se}$ | Sequential or multi-threaded program (cost estimation) | $f_{pi} \times M_{cm} + size(out) \times M_{dfo} + size(in) \times M_{dfi}$ where $f_{pi}$ is an estimated performance improvement function when $n$ expected threads to be used. $f_{pi}$ can be provided by performance prediction tools or scientists. In our case, currently, we use an ideal parallel performance improvement $f_{pi} = \frac{p}{n} \times t_e(p)$ where $p$ is the number of threads used to obtain $t_e(p)$. $p$ and $t_e(p)$ are known knowledge. |
| $M_{pm}$ | Parallel/MPI programs on multiple machines (cost monitoring) | $n \times M_{cm} \times t_e + size(out) \times M_{dfo} + size(in) \times M_{dfi}$ |
| $M_{pe}$ | Parallel/MPI programs on multiple machines (cost estimation) | $n \times M_{cm} \times f_{pi} + size(out) \times M_{dfo} + size(in) \times M_{dfi}$ where $f_{pi}$ is an estimated performance improvement function when $n$ processes are used. |
| $M_{wm}$ | Workflows (cost monitoring) | $\sum_{i=1}^{k} (size(in_i) \times M_{dfi}) + \sum_{i=1}^{l} (size(out_i) \times M_{dfo}) + \sum_{i=1}^{n} (M_{cm} \times t_e(machine_i))$ |
| $M_{we}$ | Workflows (cost estimation) | $\sum_{i=1}^{nwr} cost(wr_i)$. For a workflow region $wr_i$, $cost(wr_i) = \sum_{j=1}^{q} (cost(activity_j))$ where $cost(activity_j)$ is determined based on $M_{mp}$, $M_{sm}$, and $M_{sd}$, when the activity $activity_j$ is a parallel activity, sequential activity, or a data transfer activity, respectively. |

# Illustrative experiments

## Runtime analysis

Online Cost Monitoring an

Total cost= 5.005979133202549(EUR)
Computation cost=4.918044444444444 (EUR) for 12.94222222222
Computation: ec2-79-125-54-238.eu-west-1.compute.amazona
Activity: Fasta check U; Total usaged time=0.0(hours)
Activity: Run unaligned reads 1; Total usaged time=0.1777777
Activity: Check aligned reads 2; Total usaged time=0.0(hours)
Activity: Fasta check 2; Total usaged time=1.70666666666666
Activity: Unzip ref data; Total usaged time=0.0(hours)
Activity: Fasta check ref; Total usaged time=0.03555555555555
Activity: Unzip data; Total usaged time=0.0(hours)
Activity: Bowtie index; Total usaged time=0.14222222222222
Activity: Bowtie index U; Total usaged time=0.0(hours)
Activity: Run reads Brain1; Total usaged time=3.94666666666
Activity: Check aligned reads; Total usaged time=0.0(hours)
Activity: Check unaligned reads 1; Total usaged time=0.0(hour
Activity: Evaluate aligned results 1; Total usaged time=0.10666
Activity: Run reads Brain2; Total usaged time=3.59111111111
Activity: Fasta check; Total usaged time=1.137777777777777
Activity: Run unaligned reads 2; Total usaged time=0.21333333333333335(hours)
Activity: Check unaligned reads 2; Total usaged time=0.0(hours)
Activity: Evaluate aligned results 2; Total usaged time=1.2444444444444445(hours)
Activity: Unzip data 2; Total usaged time=0.0(hours)
Transfer cost= 0.087934688758105050(EUR) for 0.8793468875810504 (GB)
Transfer in: ec2-79-125-54-238.eu-west-1.compute.amazonaws.com: Total transferred size=0.8793468875810504(GB)
Activity: Download fasta data; Total transferred size=0.3350419262424114(GB)
Activity: Download fasta ref data; Total transferred size=0.02491060271859169(GB)
Activity: Download fasta U; Total transferred size=0.004382843151688576(GB)
Activity: Download Fast Data 2: Total transferred size=0.515011515468359(GB)

## Estimation

| Activities | Nr Machine | Transfer In (GB) | Transfer Out(GB) | Time (hours) | Storage (GB) | 1 Run Cost (EUR) | 30 Runs Cost (EUR) |
|---|---|---|---|---|---|---|---|
| Partial cloud-based resources | | | | | | | |
| DF1 | 1 | 0.43 | 0.43 | 0.122 | 0 | 0.139 | 4.18 |
| Preanalysis | 1 | 0 | 0 | 0.28 | | 0.053 | 1.596 |
| Analysis | 4 | 0 | 0 | 0.24 | | 0.182 | 5.472 |
| DF2 | 1 | 2.56 | 2.56 | 0.728 | | 0.83 | 24.887 |
| Total | | | | 1.37 | | 1.204 | 36.135 |
| Full cloud-based resources | | | | | | | |
| DF1 | 1 | 0.43 | 0.43 | 0.015 | 3 | 0.569 | 17.07 |
| Preanalysis | 1 | 0 | 0 | 0.28 | | 0.053 | 1.596 |
| Analysis | 4 | 0 | 0 | 0.24 | | 0.182 | 5.472 |
| DF2 | 1 | 2.56 | 2.56 | 0.091 | | 0.708 | 21.255 |
| Total | | | | 0.626 | | 1.513 | 45.393 |

Table 3: Example of estimated costs for running the Bones application with different possibilities. Costs for input data transfer/GB, output data transfer/GB, CPU instance/hour and storage/month are 0.1 EUR , 0.17 EUR, 0.19 EUR and 0.15 EUR. Data transfer within our cloud is 8MB/s and between our cloud and user's on-premise storage is 1MB/s.

## Cloud provider bill

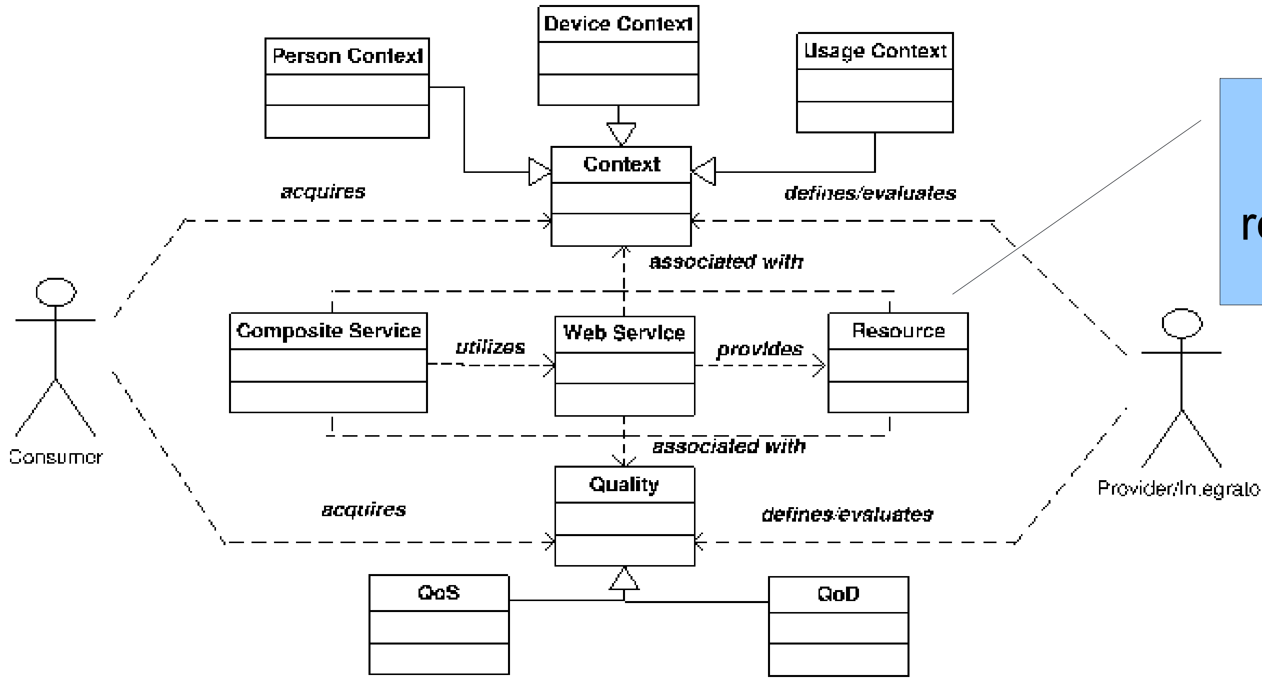
| Operation Name | Usage Type | Usage Value | Cost (EUR) |
|---|---|---|---|
| RunInstances | EU-BoxUsage:m1.large | 13 hours | 4,94 |
| RunInstances | EU-DataTransfer-In-Bytes | 1,043 GB | 0,104 |
| RunInstances | EU-DataTransfer-Out-Bytes | 4,48 GB | 0,672 |
| EBS:IO-Write | EU-EBS:VolumeIOUsage | 2083828 I/O calls | 0,229 |

H

Table 4: Operation name, usage types and usage values extracted from the Amazon billing report during the execution of the GSA workflow. We determined costs by using the following prices: 0.38 EUR per large instance (m1.large) instance-hour, 0.1 EUR per GB data transfer in, 0.150 EUR per GB data transfer out, 0.11 EUR per 1 million I/O requests.

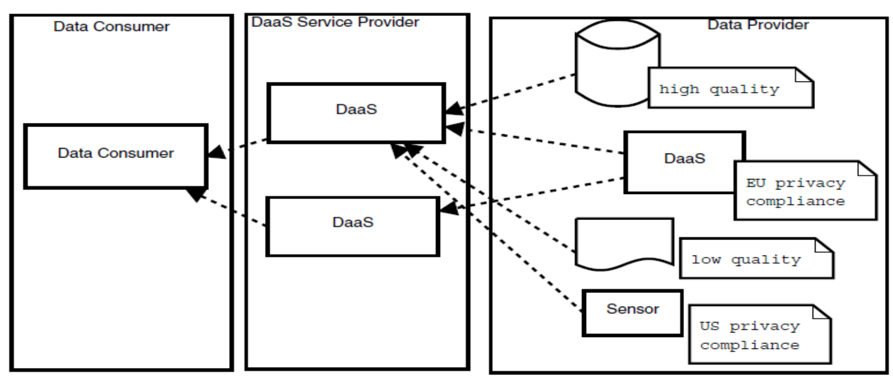
# Complex services and data composition/provisioning

Complex NFPs in clouds




Data resource

Elastic data provisioning



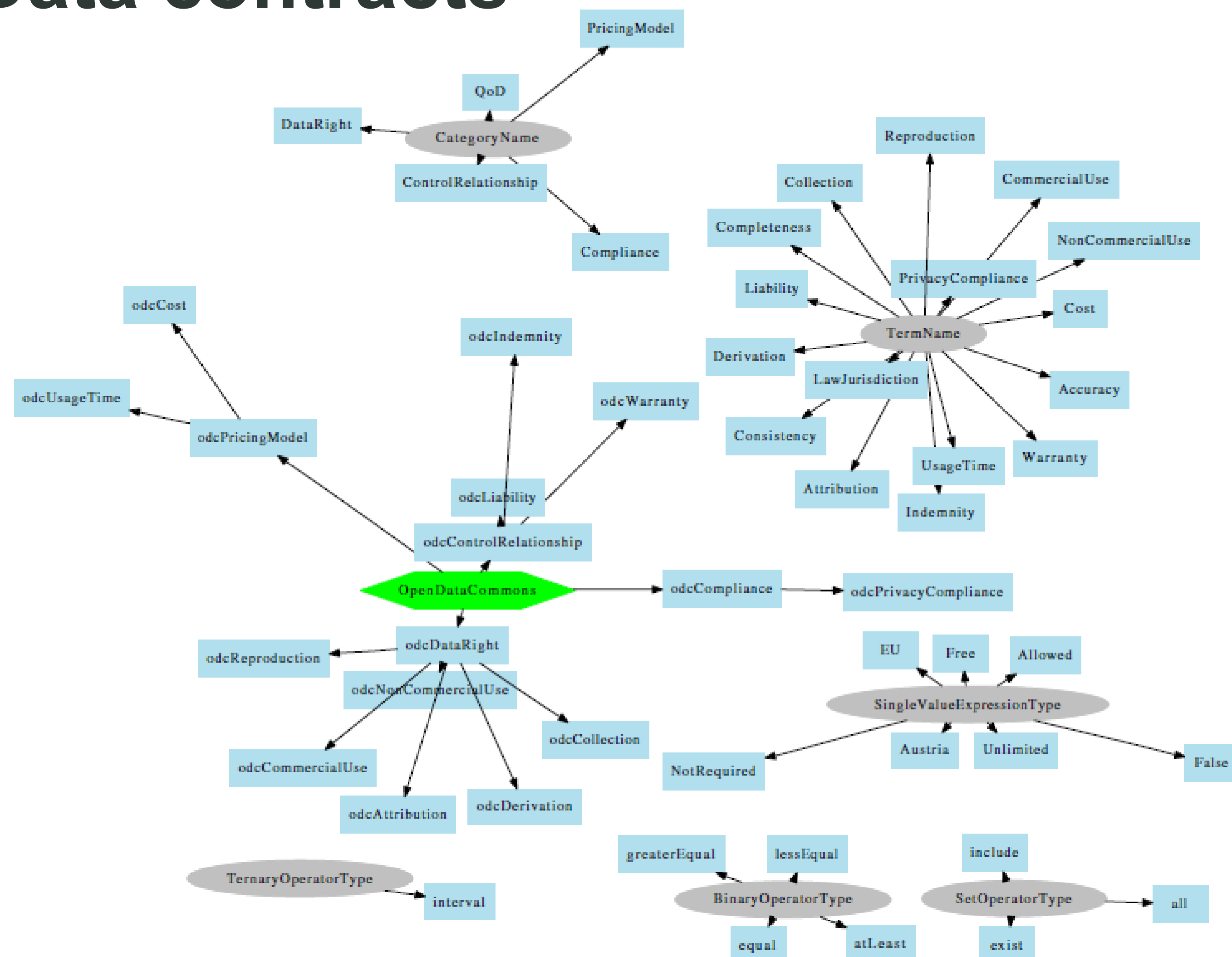



VITALAB Distributed Systems Group

# Service contract compatibilities for being elastic

- Three aspects – *service APIs*, *data APIs*, and *provided data concerns*;
  - a rich set of contract properties (e.g., *QoS, Data quality, Business, License* and *Context terms*);
  - several service contract specification languages (e.g., WSLA, WSOL, ODRL-S) together
  - QoS, Business, License and Context terms differently influence data/control flows of the service composition.
- So far research efforts are focused on service APIs and single contract specification

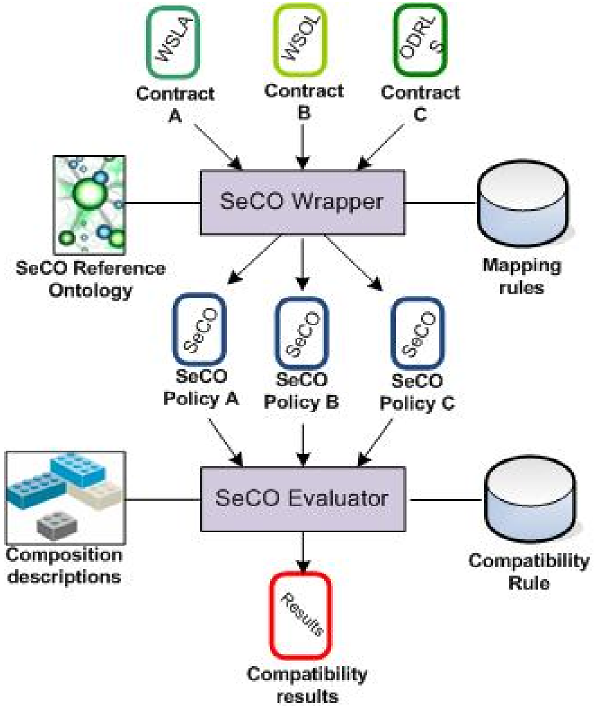
# Data contracts




Hong-Linh Truong, G.R. Gangadharan, Marco Comerio, Schahram Dustdar, Flavio De Paoli, "On Analyzing and Developing Data Contracts in Cloud-based Data Marketplaces", The 2011 Asia-Pacific Services Computing Conference (IEEE APSCC 2011), (c) IEEE Computer Society, December 12 - 15, 2011, Jeju, Korea

# Evaluating service contract compatibility

## Service Contract Mapping




## Data contract management

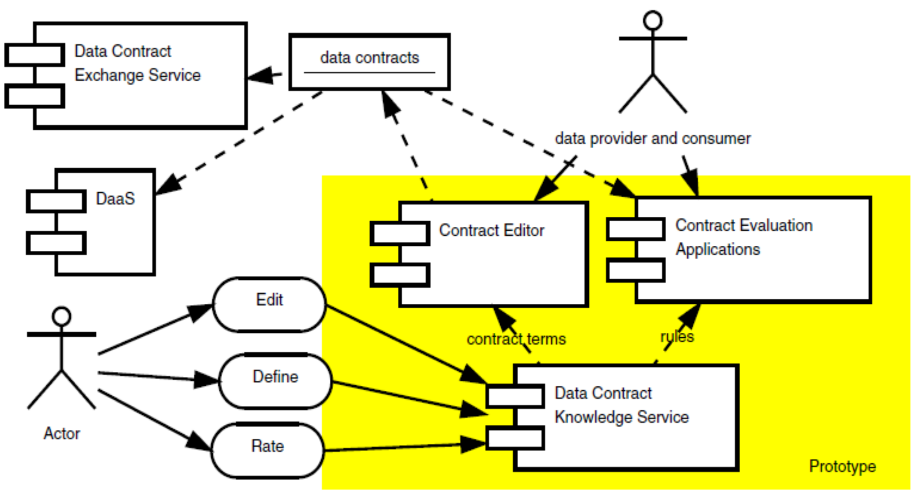
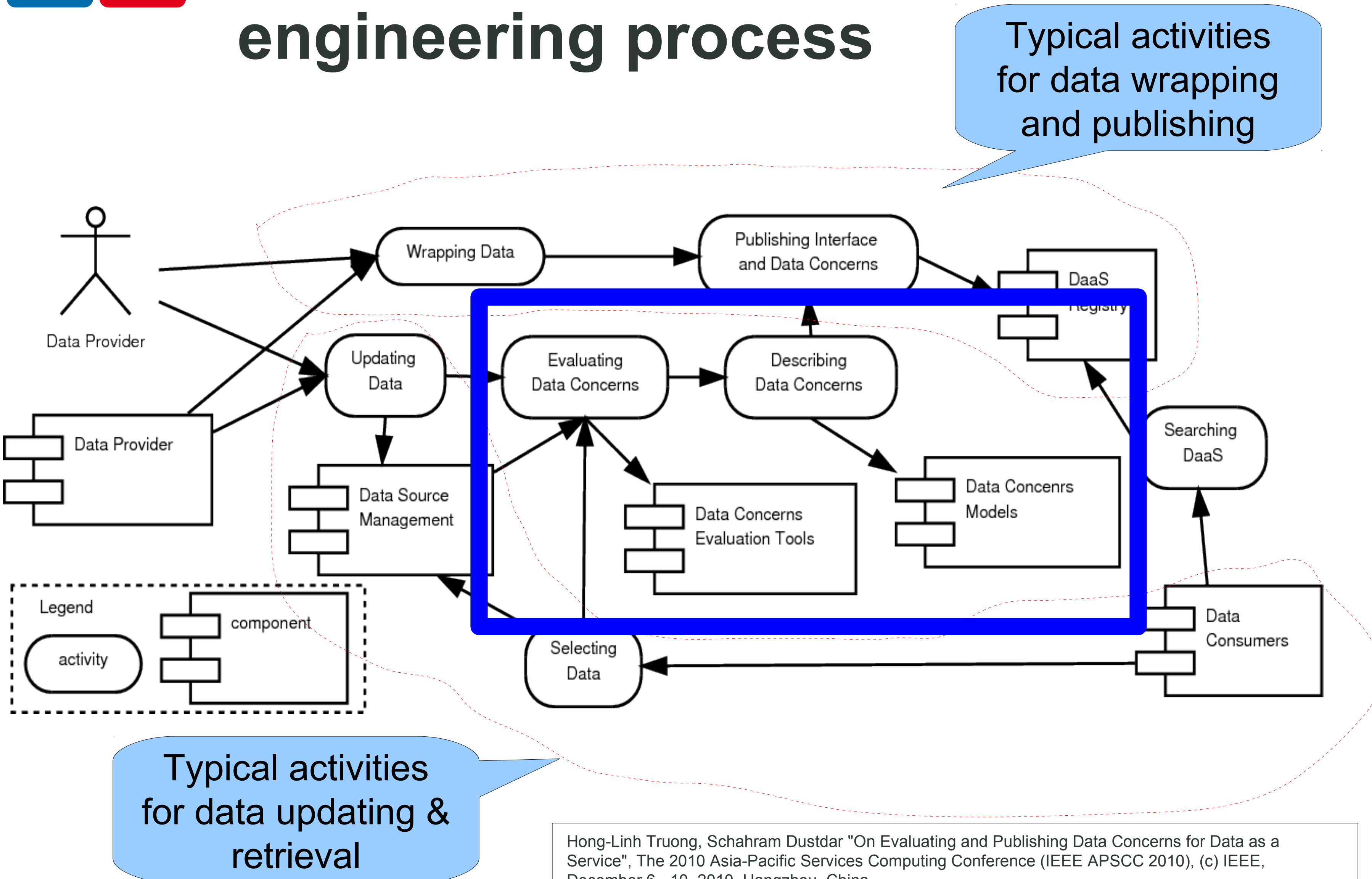



Fig. 3. Our prototype for data contract management

Marco Comerio, Hong-Linh Truong, Flavio De Paoli, Schahram Dustdar, " Evaluating Contract Compatibility for Service Composition in The SeCO2 Framework ", The 9th International Conference on Service Oriented Computing (ICSOC 2009), (c) Springer-Verlag, November 24 - 27, 2009, Stockholm, Sweden.

Hong-Linh Truong, G.R. Gangadharan, Marco Comerio, Schahram Dustdar, Flavio De Paoli, "On Analyzing and Developing Data Contracts in Cloud-based Data Marketplaces", The 2011 Asia-Pacific Services Computing Conference (IEEE APSCC 2011), (c) IEEE Computer Society, December 12 - 15, 2011, Jeju, Korea

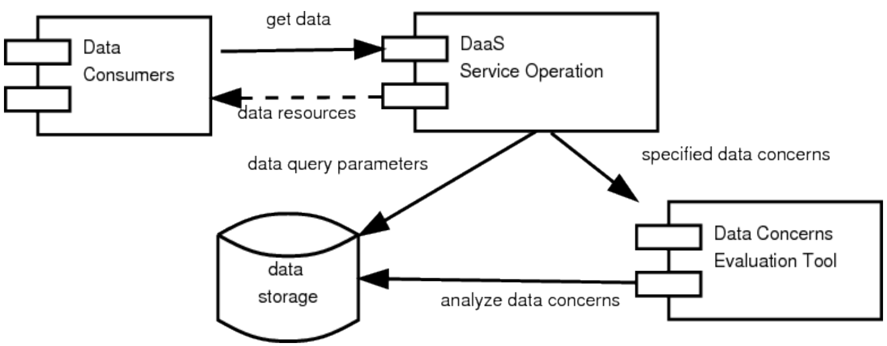
# Data concern-aware service engineering process



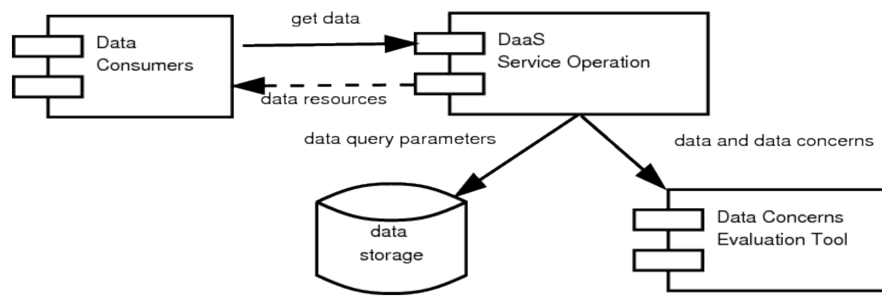

Hong-Linh Truong, Schahram Dustdar "On Evaluating and Publishing Data Concerns for Data as a Service", The 2010 Asia-Pacific Services Computing Conference (IEEE APSCC 2010), (c) IEEE, December 6 - 10, 2010, Hangzhou, China

# Evaluating data concerns (2)

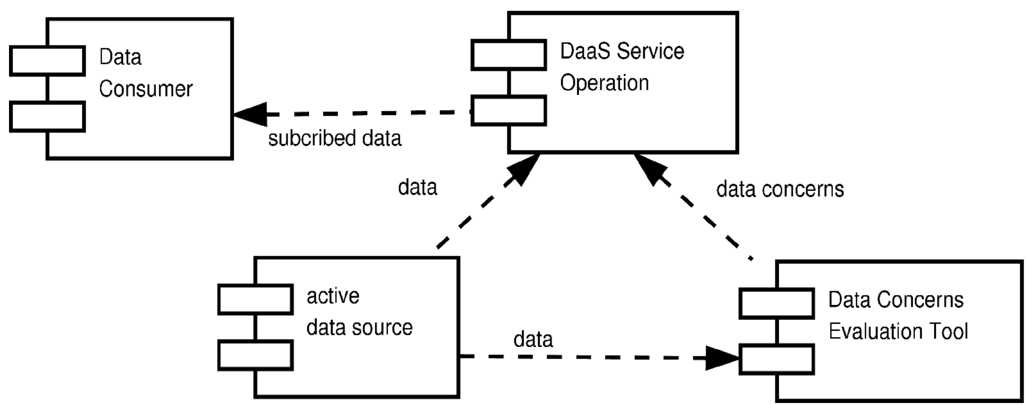
## Pull, pass-by-references




## Pull, pass-by-values




## Push, pass-by-values



# Hybrid resources needed for quality evaluation

- Challenges:
  - Subjective and objective evaluation
  - Long running processes
- Our approach
  - Different QoD measurements
  - Human and software tasks

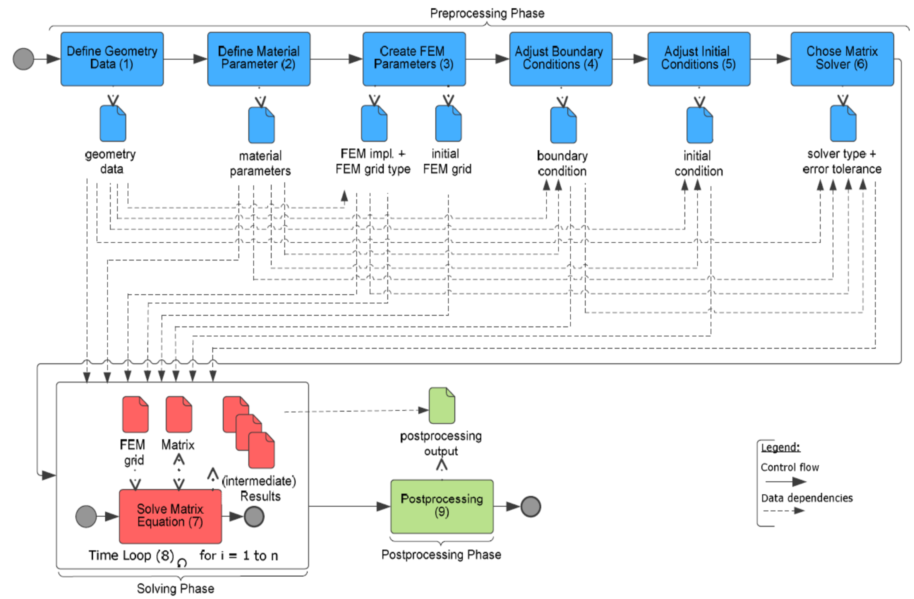
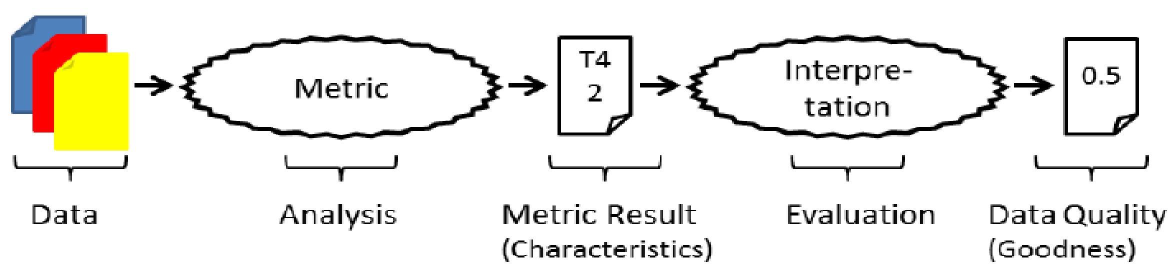


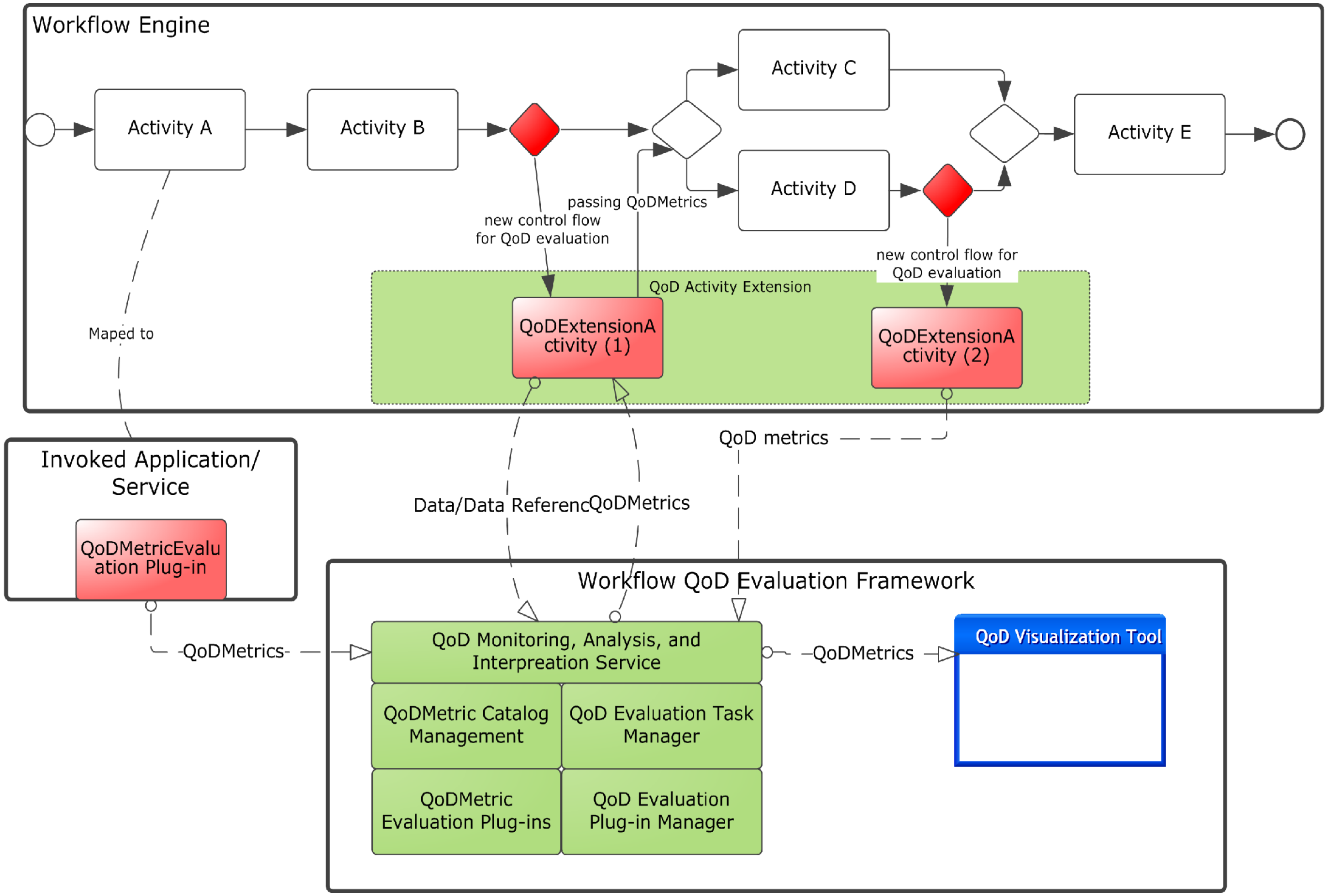

**Figure 3: Finite element method (FEM) based simulation workflow with data dependencies.**






VITALAB Distributed Systems Group

# Evaluating quality of data in workflows




Michael Reiter, Uwe Breitenbuecher, Schahram Dustdar, Dimka Karastoyanova, Frank Leymann, Hong-Linh Truong, A Novel Framework for Monitoring and Analyzing Quality of Data in Simulation Workflows, (c)IEEE Computer Society, The 7th IEEE International Conference on e-Science, 5-8 December, 2011, Stockholm, Sweden

# Open questions for elasticity in hybrid systems

- Software, data and human integration
- Monitor elastic processes in hybrid systems
  - On the fly monitoring provisioning and complex cost models from different layers
  - Monitoring of human-based computing elements
- Contract compatibility – no static
  - Dynamic contract evaluation for data, software-based services, and human-based computing elements
- Quality of data complex workflows
  - Evaluating quality of data of workflows with using hybrid processes of human + software
  - Elasticity of price and quality impact on cloud service and infrastructure provisioning

# Conclusions

- Elastic computing is needed in different aspects
  - Scaling software, services, and people in the same application
- But elasticity is not just “resource elasticity”
  - Resource, cost, quality, etc.
  - Hybrid systems of  software-based and human-based computing elements
  - Multiple clouds
- Several open research questions for realizing elastic processes of hybrid computing elements

# Thanks for your attention!

Hong-Linh Truong

Distributed Systems Group
Vienna University of Technology

Austria

truong@infosys.tuwien.ac.at
http://www.infosys.tuwien.ac.at