Engineering Human-based Services in Hybrid Computing Systems

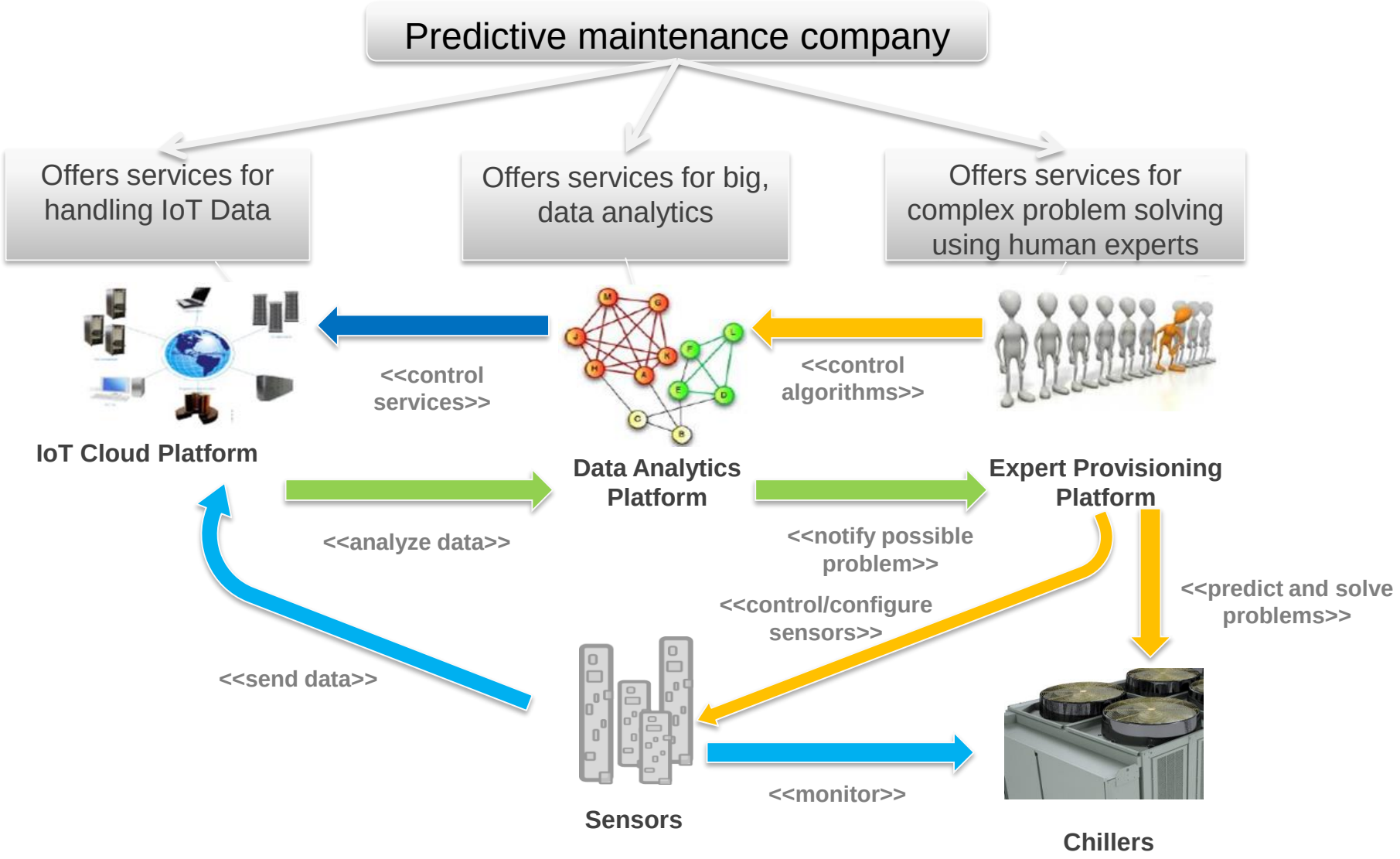
Hong-Linh Truong
Distributed Systems Group, TU Wien

truong@dsg.tuwien.ac.at
http://dsg.tuwien.ac.at/staff/truong
@linhsolar

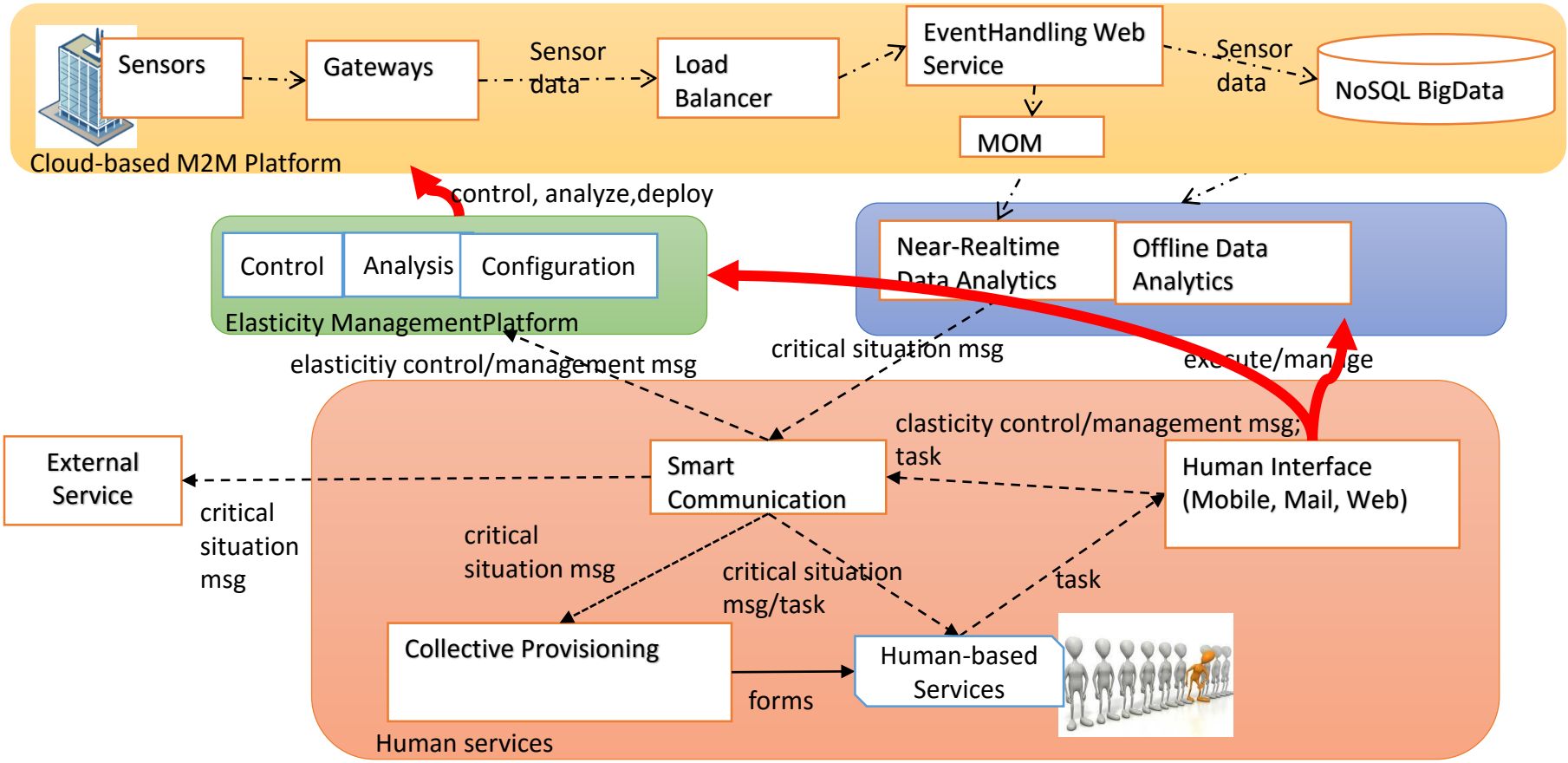
What this lecture is about?

- Motivating scenarios
- Human service units
- Provisioning and employing human service units – frameworks

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>>
<<predict and solve problems>>
<<control/configure sensors>>
<<send data>>
<<monitor>>
Sensors
Chillers

Integrated systems of software, things and people services



Hybrid intelligence

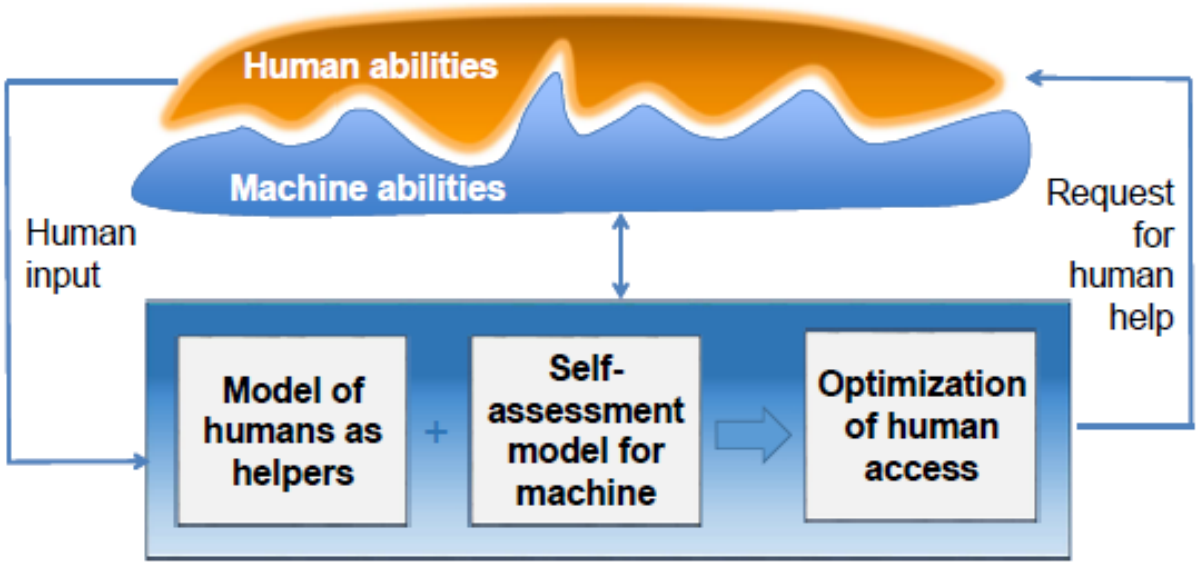
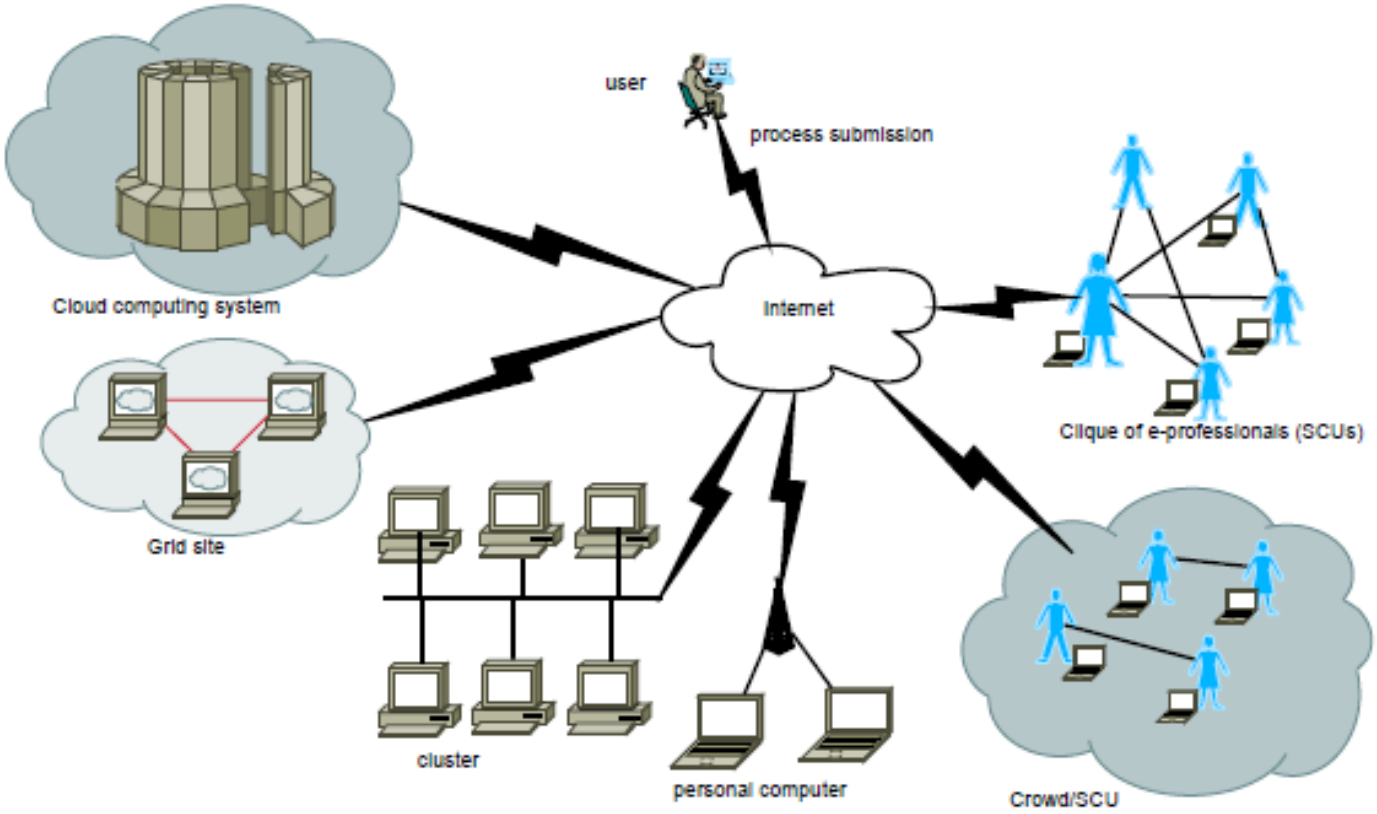


Figure 1: Reasoning capabilities for hybrid intelligence.

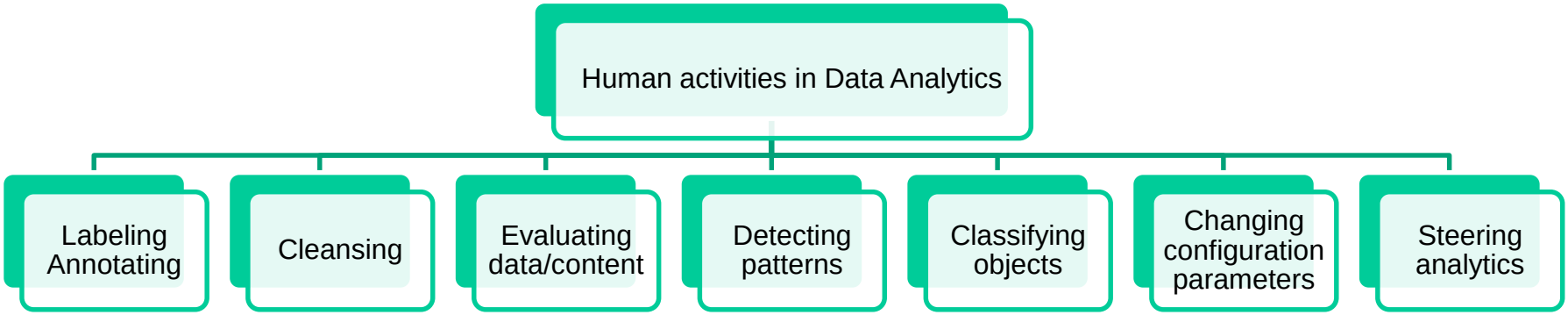
Source: Ece Kamar. 2016. Directions in hybrid intelligence: complementing AI systems with human intelligence. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (IJCAI'16), Gerhard Brewka (Ed.). AAAI Press 4070-4073. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/11/hi.pdf

Human-based services for solving complex problems (2)



But how to program human-based services and software-based services together?

Example: some common tasks in data analytics



Human service units in data analytics -- functions

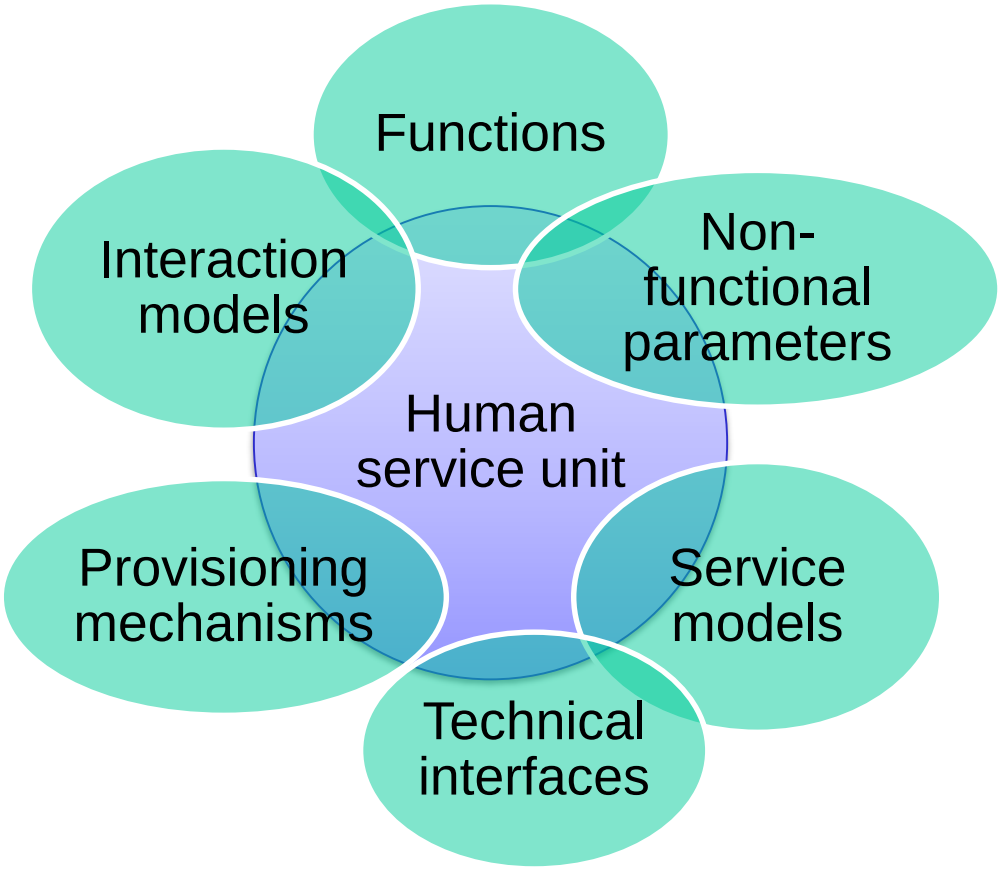
- Evaluating: is the quality of picture good?
- Classifying: is it a man‘s or a woman‘s picture?
- Detecting: any unidentified object in a picture?
- Labeling: adding location information of a picture
- Cleansing: remove duplicated pictures
- Steering: the quality of picture is bad, should we continue to merge it with others?

How to model such functions for human units ?
E.g., with WSDL or REST?

HUMAN SERVICE UNITS

Human service units

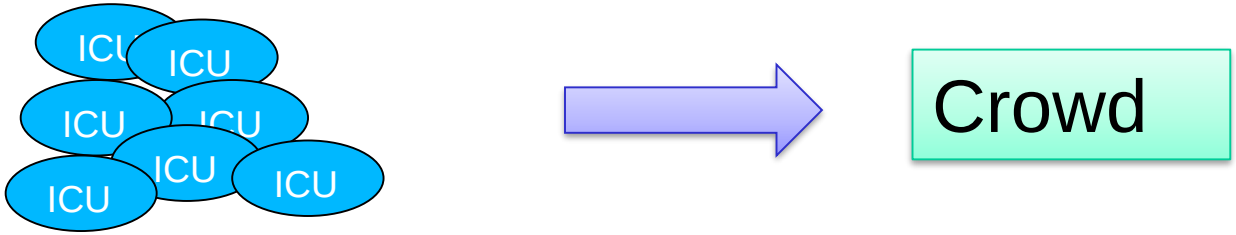
Human acting as a „service unit“



Forms of human service

- Individual Compute Unit
  - An individual is treated like „a processor“ or “functional unit“. A service can wrap human capabilities to support the communication and coordination of tasks
- Hybrid Compute Unit (Collective)
  - A set of people and software that are initiated and provisioned as a service for solving tasks
- Web services interfaces can be built
- Different pricing models and different quality models

Human service units – provisioning mechanisms (1)



- An infrastructure can be introduced for accessing many ICUs in a crowd
  - Allow people to register their service unit capabilities
  - Facilitate communication, task bidding, retrieval and result delivery
  - Act like a marketplace: multiple providers and multiple consumers

Human service units – provisioning mechanisms (2)

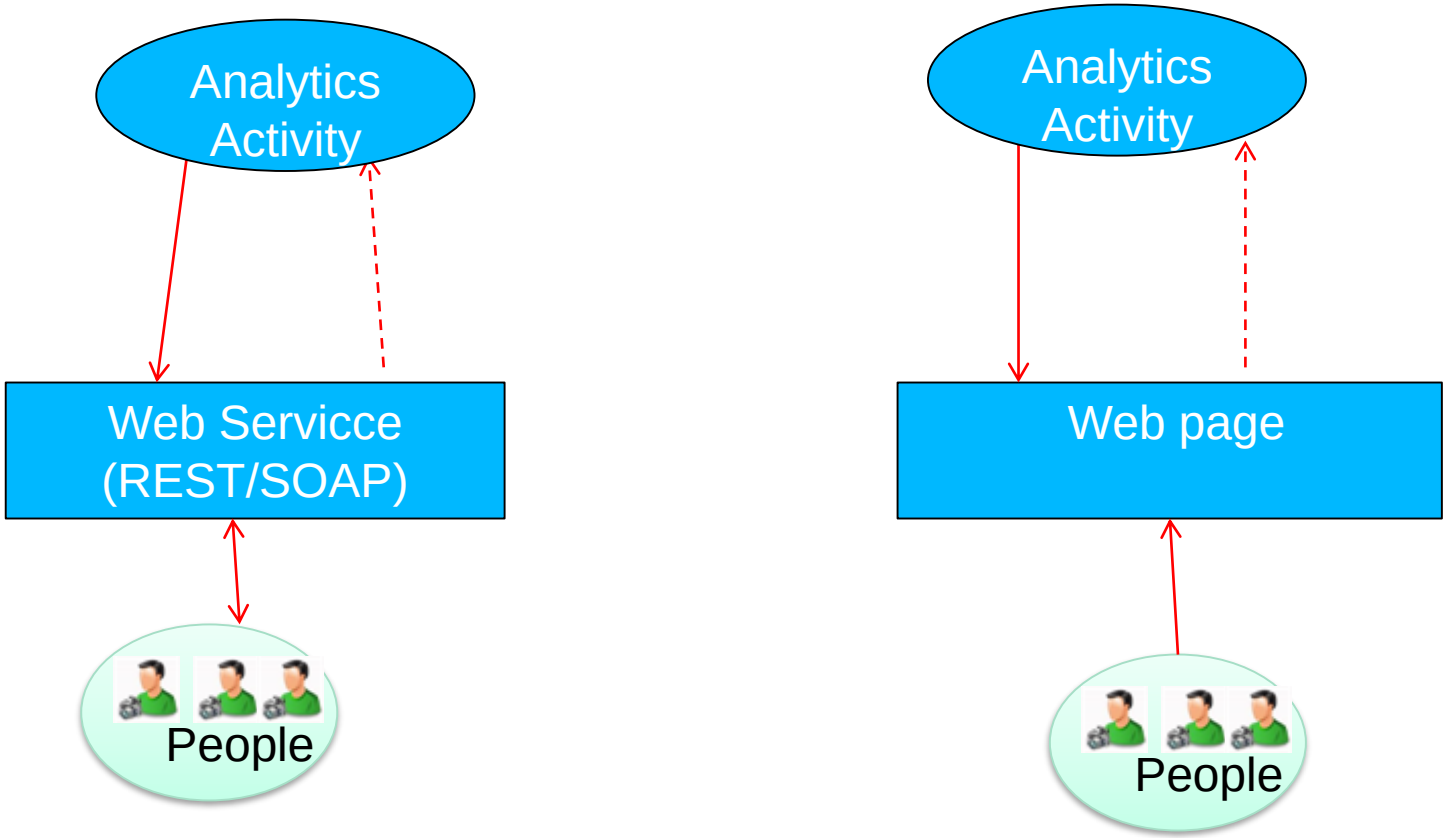
- An „infrastructure-as-a-service“ for ICUs
  - Facilitate communication, task retrieval and result delivery
  - Single ICUaaS provider and multiple consumers

MTurk as an ICU provider

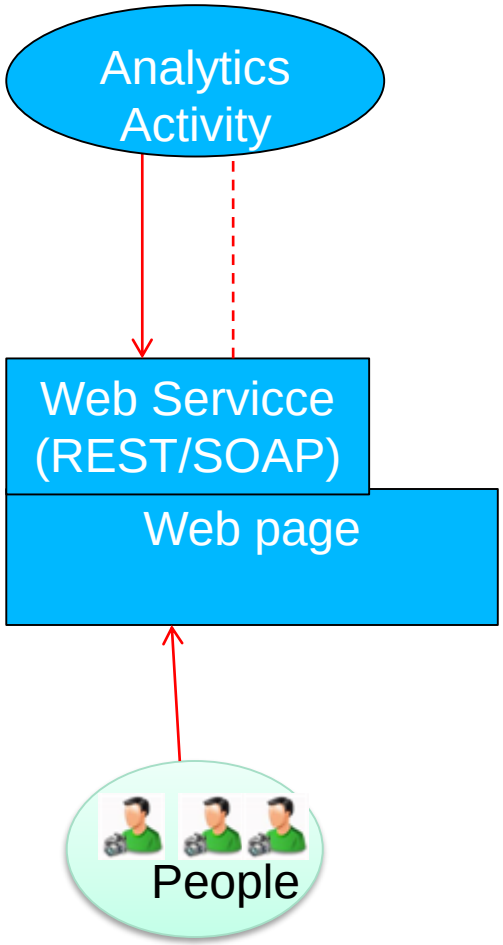
Human service units – provisioning mechanisms (3)

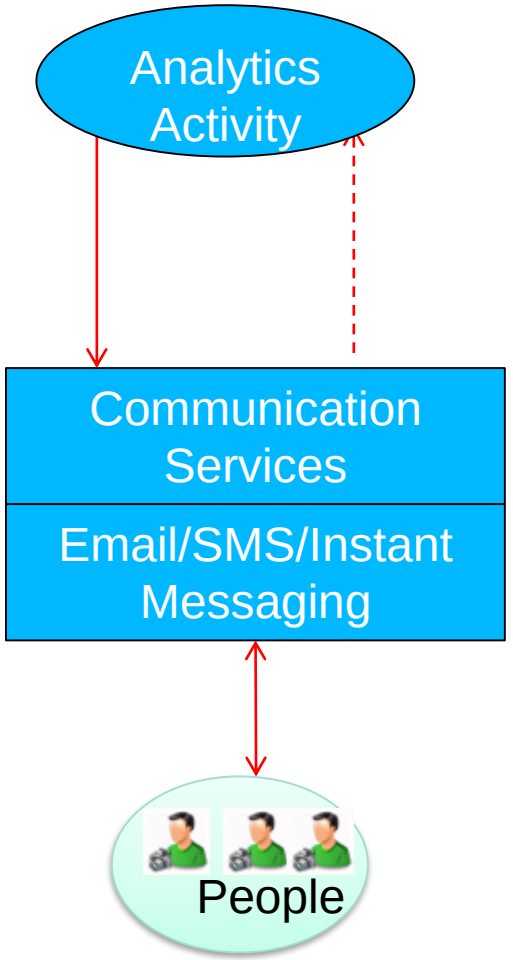
- An „infrastructure-as-a-service“ for SCUs
  - Facilitate communication, task retrieval and result delivery
  - Single SCUaaS provider and multiple consumers

Human service units – technical interfaces (1)

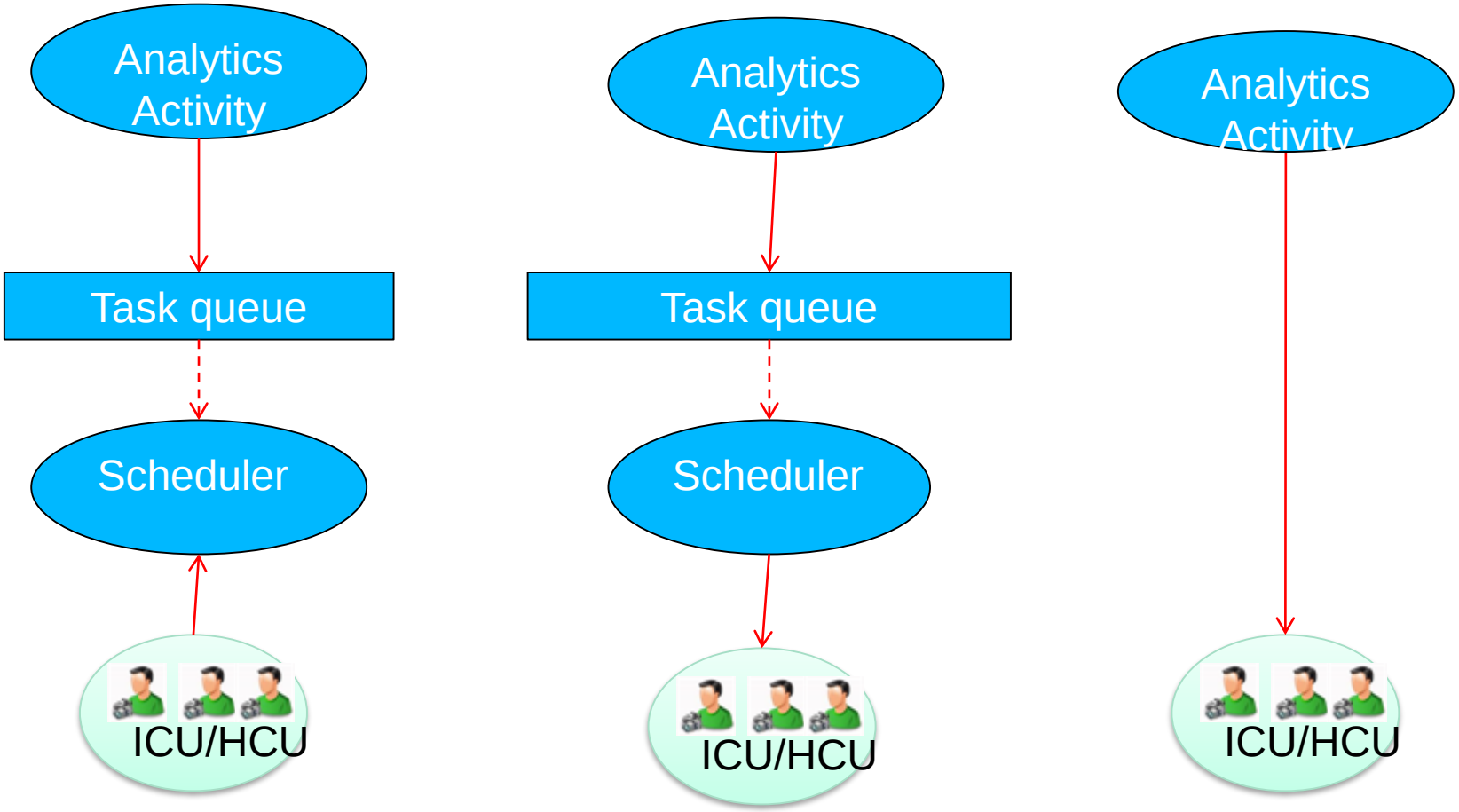


Human service units – technical interfaces (2)

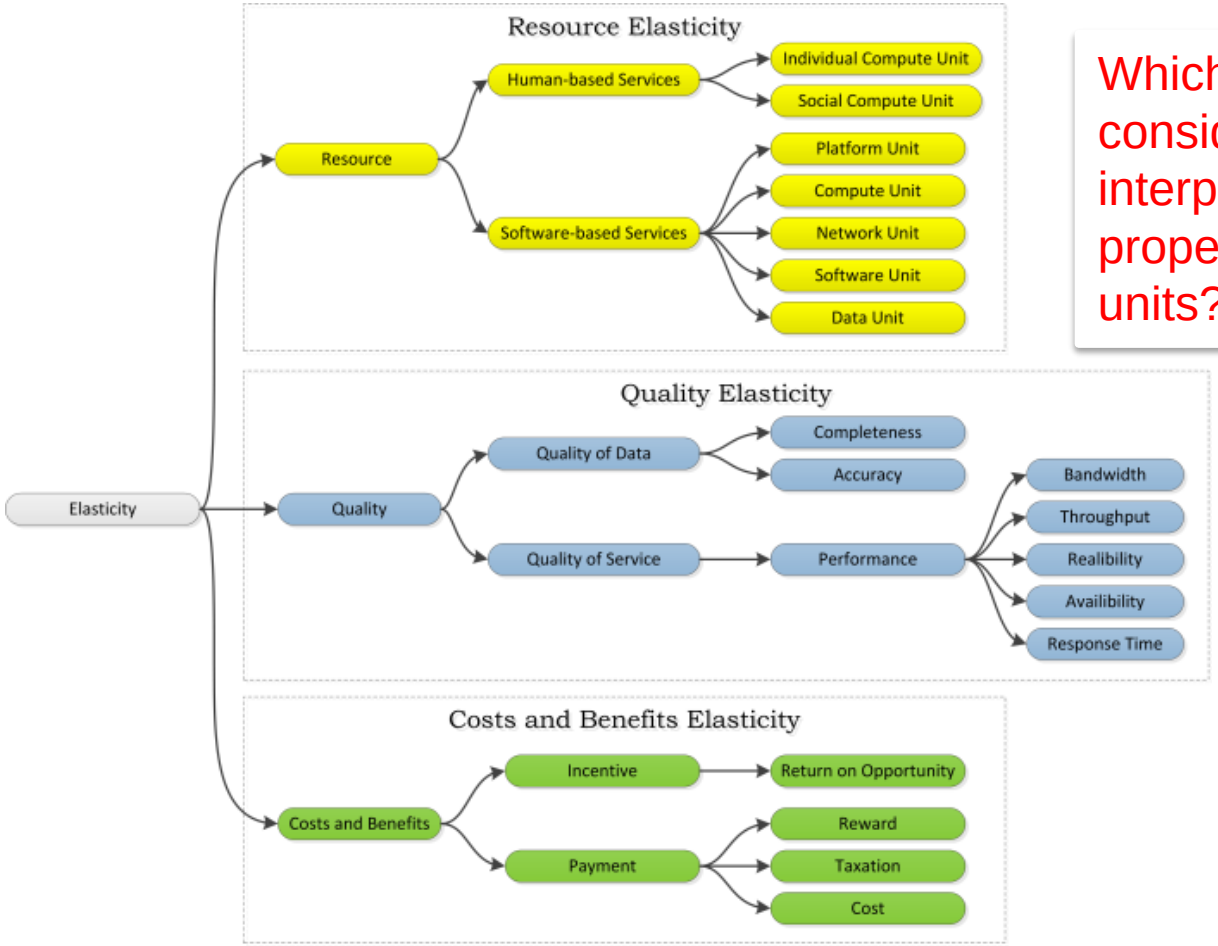




Human service units – interaction model



Human service units -- NfPs



Which are important considerations when interpreting non-functional properties for human service units?

Incorporating human units into complex processes

- How to provision and employ human compute units?
- How to select human units?
- Where to place human units in data analytics and why?
- How to monitor and test human units in data analytics?

PROVISIONING AND EMPLOYING HUMAN SERVICE UNITS-- SOME FRAMEWORKS

Approaches

- Software perform task routing and management
- Software perform the work and invoke human only needed

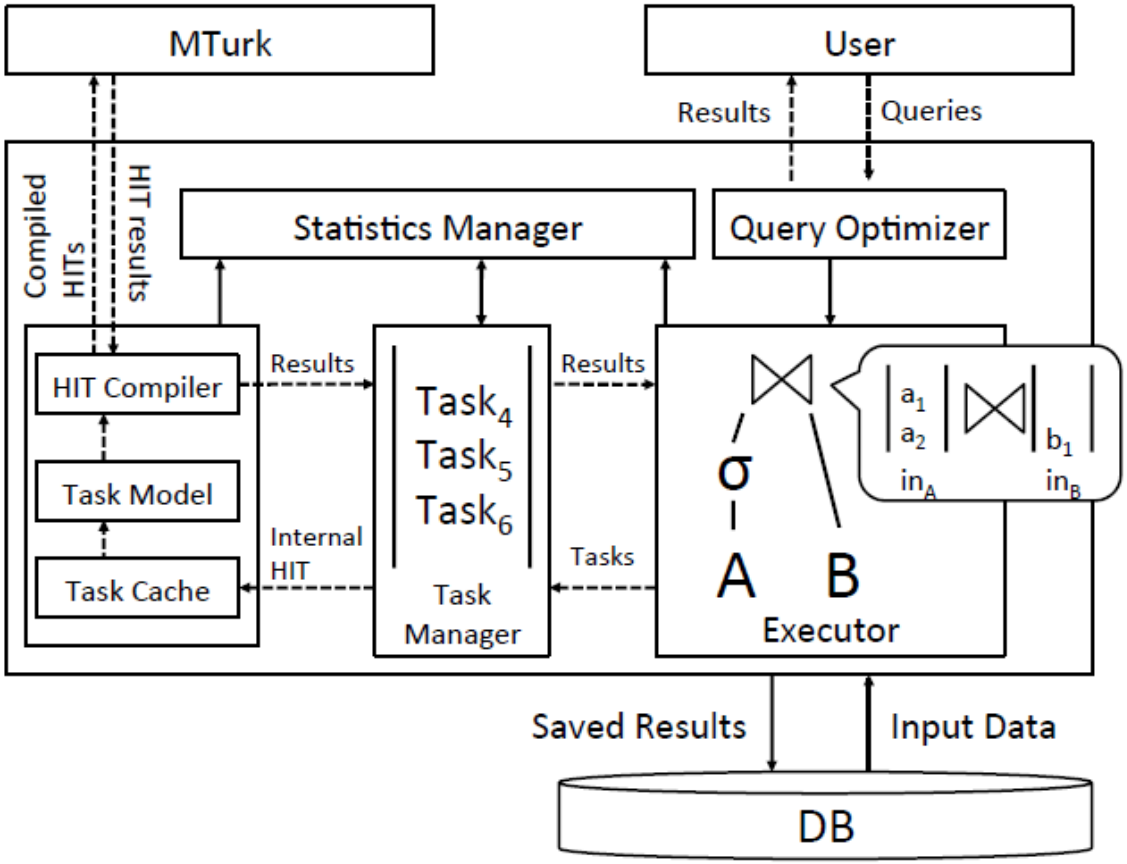
Qurk system architecture (1)

```
SELECT c.name
FROM celeb c JOIN photos p
ON samePerson(c.img,p.img)
AND POSSIBLY gender(c.img) = gender(p.img)
AND POSSIBLY hairColor(c.img) = hairColor(p.img)
AND POSSIBLY skinColor(c.img) = skinColor(p.img)
```

```
TASK gender(field) TYPE Generative:
    Prompt: "<table><tr> \
              <td><img src='%s'> \
              <td>What is this person's gender? \
            </table>", tuple[field]
    Response: Radio("Gender",
               ["Male","Female",UNKNOWN])
    Combiner: MajorityVote
```

Source: Adam Marcus, Eugene Wu, David Karger, Samuel Madden, and Robert Miller. 2011. Human-powered sorts and joins. Proc. VLDB Endow. 5, 1 (September 2011), 13-24.

Qurk system architecture (2)



Source: Adam Marcus, Eugene Wu, David Karger, Samuel Madden, and Robert Miller. 2011. Human-powered sorts and joins. Proc. VLDB Endow. 5, 1 (September 2011), 13-24.

Jabberwocky approach (1)

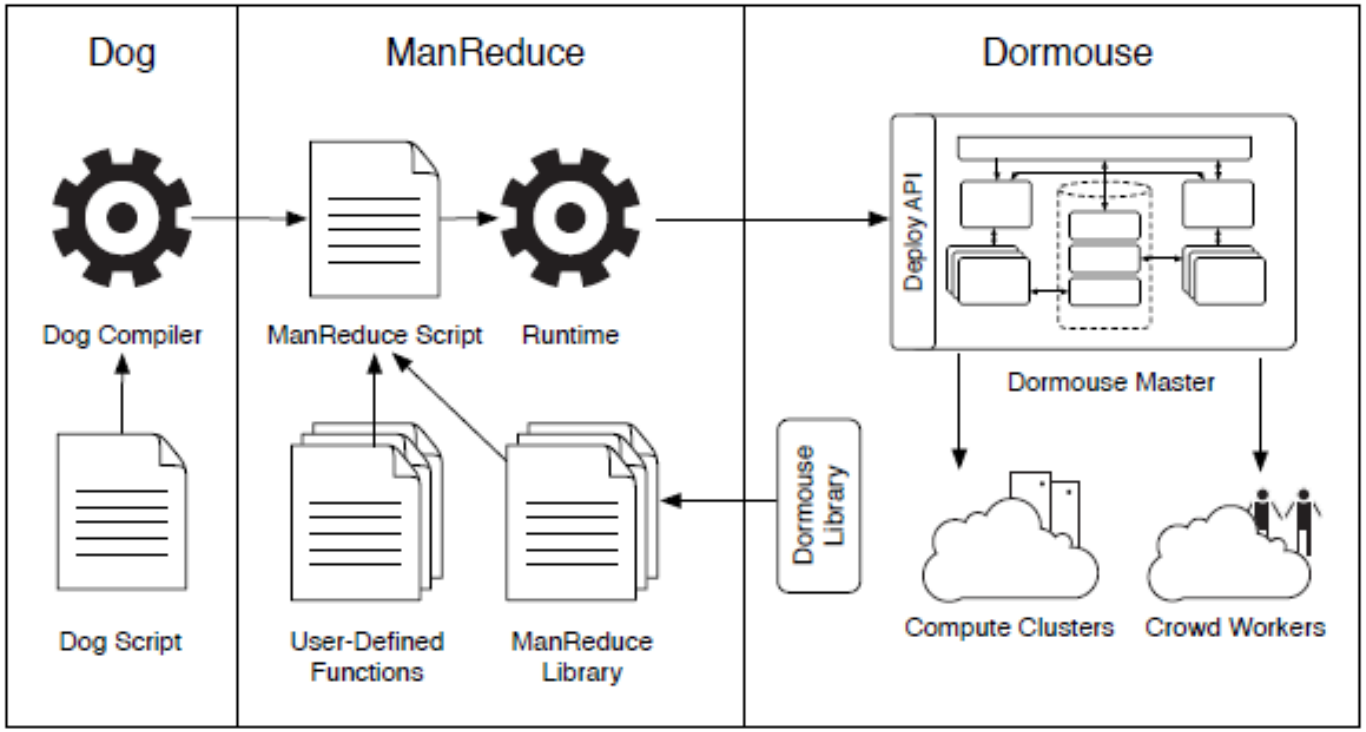


Figure 1: Overview of Jabberwocky

Source: Salman Ahmad, Alexis Battle, Zahan Malkani, Sepandar D. Kamvar: **The jabberwocky programming environment for structured social computing**. UIST 2011: 53-64

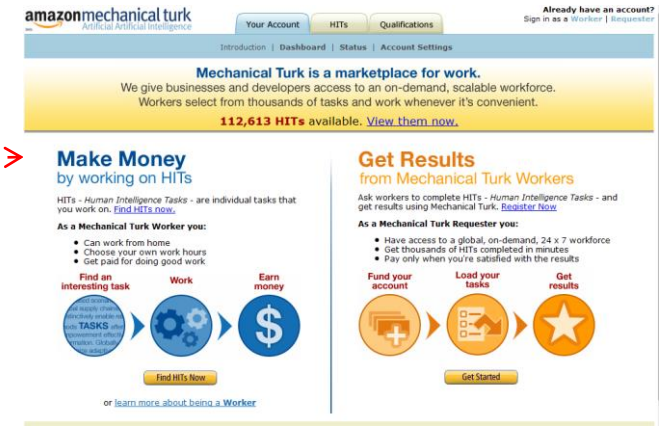
Jabberwocky approach (2)

```
1  map :name => :extract_disease_facts do |key,
       value|
2    facts = RiskExtractor.extract (value)
3
4    for fact in facts do
5      emit (fact["disease"], fact["risk_factor"
         ])
6    end
7
8  end
9
10 reduce :name => :summarize do |key, values|
11
12   task = SummarizeFacts.prepare
13     :task_name => "Summarize disease risks:
         #{key}"
14   task.facts = values
15
16   task.ask do |answer|
17     emit (key, answer)
18   end
19
20 end
```

Source: Salman Ahmad, Alexis Battle, Zahan Malkani, Sepandar D. Kamvar: **The jabberwocky programming environment for structured social computing**. UIST 2011: 53-64

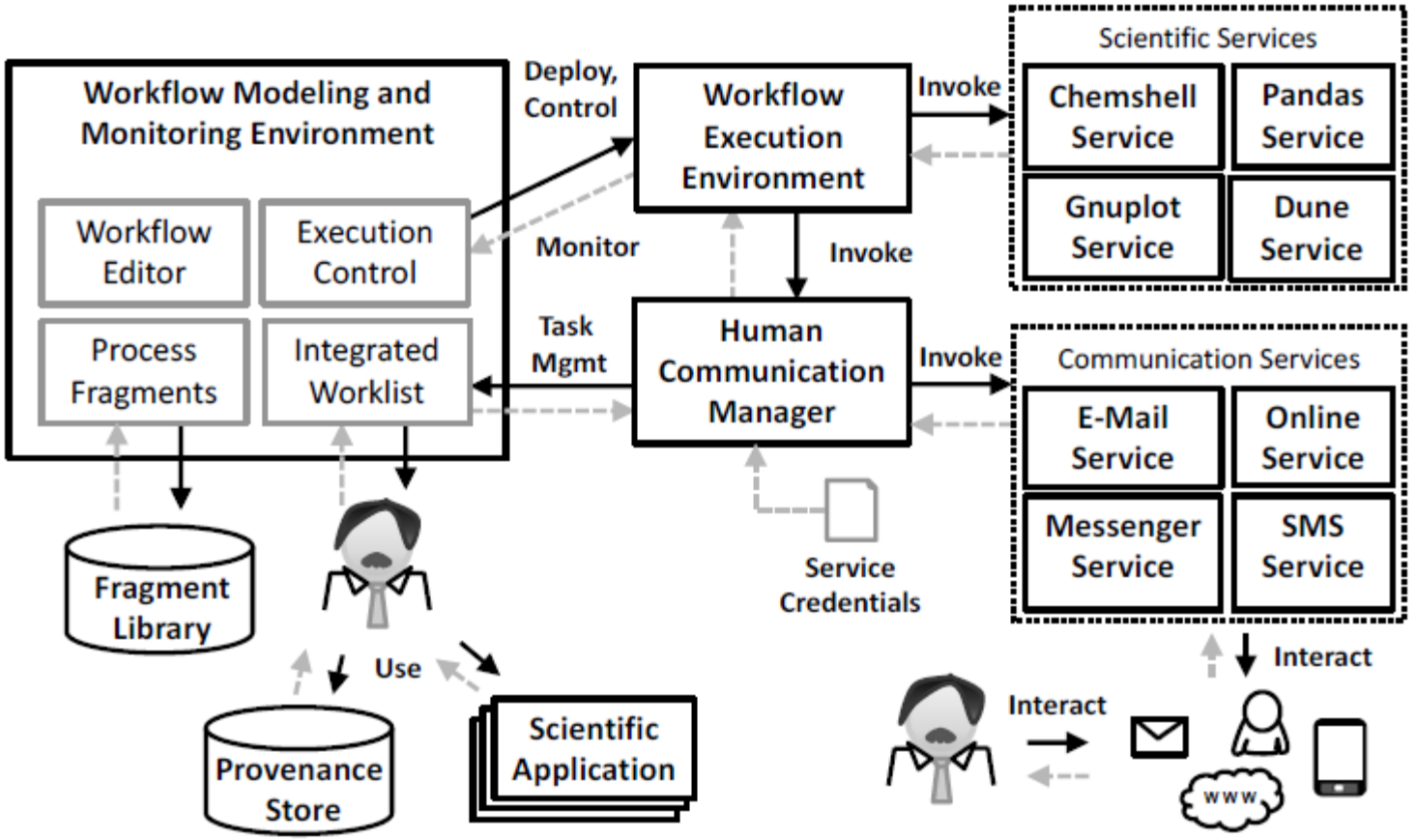
Automan approach

```
1  import edu.umass.cs.automan.adapters.MTurk._
2
3  object SimpleProgram extends App {
4    val a = MTurkAdapter { mt =>
5      mt.access_key_id = "XXXX"
6      mt.secret_access_key = "XXXX"
7    }
8
9    def which_one() = a.RadioButtonQuestion { q =>
10     q.budget = 8.00
11     q.text = "Which one of these does not belong?"
12     q.options = List(
13       a.Option('oscar, "Oscar the Grouch"),
14       a.Option('kermit, "Kermit the Frog"),
15       a.Option('spongebob, "Spongebob Squarepants"),
16       a.Option('cookie, "Cookie Monster"),
17       a.Option('count, "The Count")
18     )
19   }
20
21   println("The answer is " + which_one()())
22 }
```



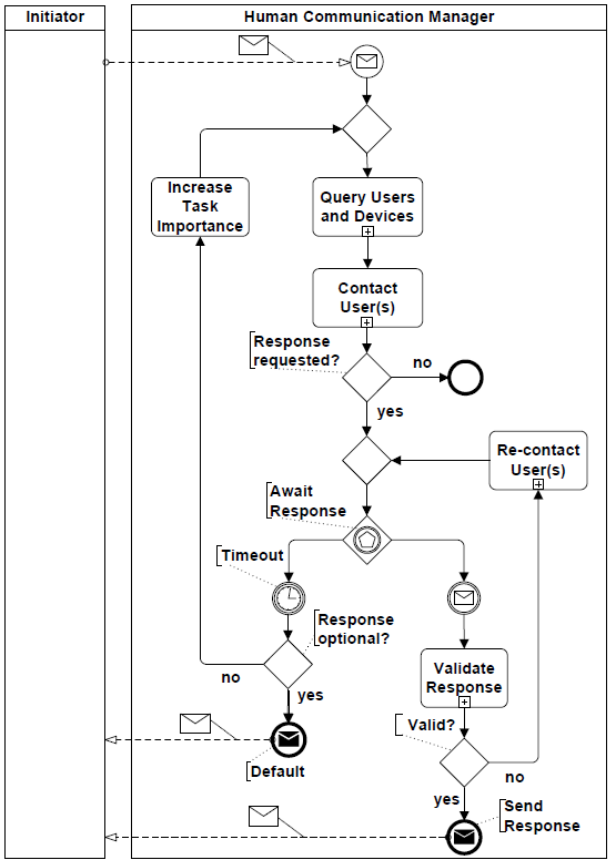
Source: Daniel W. Barowy, Charlie Curtsinger, Emery D. Berger, Andrew McGregor: **AutoMan: a platform for integrating human-based and digital computation**. OOPSLA 2012: 639-654

SW4H approach (1)



Karastoyanova, Dimka; Dentsas, Dimitrios; Schumm, David; Sonntag, Mirko; Sun, Lina; Vukojevic, Karolina: Service-based Integration of Human Users in Workflow-driven Scientific Experiments. In: Proceedings of the 8th IEEE International Conference on eScience (eScience 2012

SW4H approach (2)



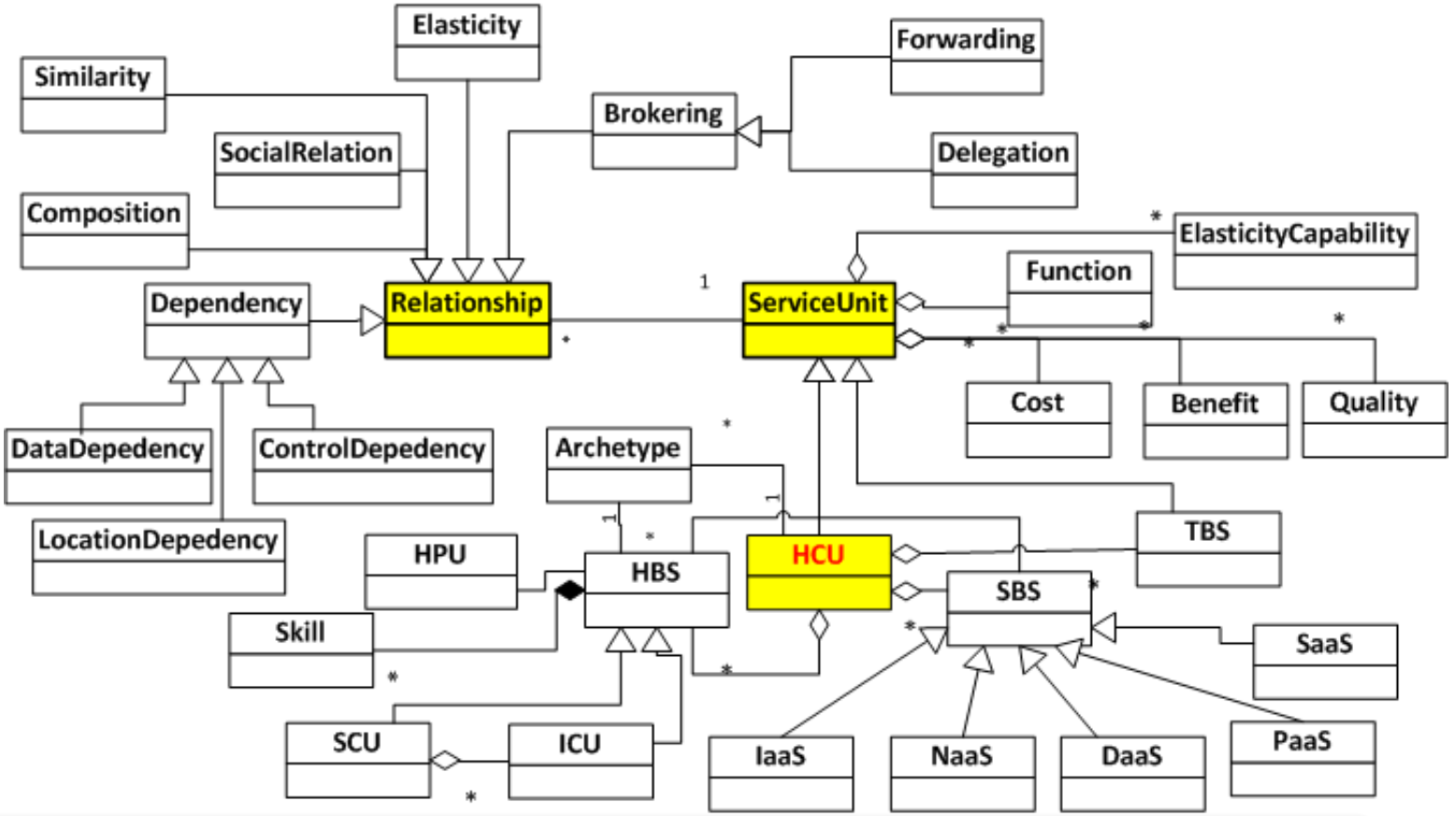
- Similar concepts in collaborative working environments but integrated into workflows
- Do not discuss about where and how to select human units

Karastoyanova, Dimka; Dentsas, Dimitrios; Schumm, David; Sonntag, Mirko; Sun, Lina; Vukojevic, Karolina: Service-based Integration of Human Users in Workflow-driven Scientific Experiments. In: Proceedings of the 8th IEEE International Conference on eScience (eScience 2012

Viecom - Hybrid compute units

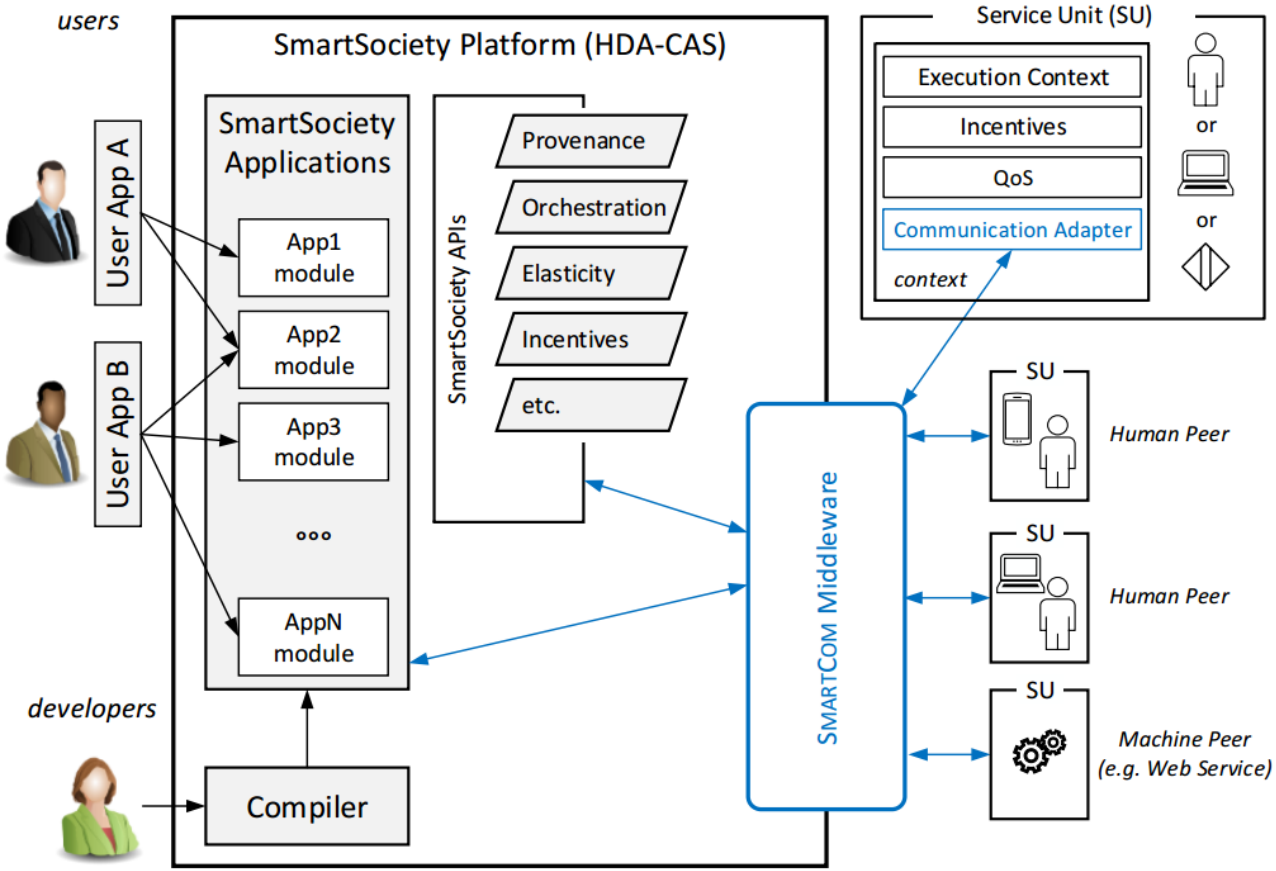
Hybrid compute unit (HCU): a set of service units includes software-based services, human-based services and things-based services *that can be provisioned, deployed and utilized as a collective* on-demand based on different quality, pricing and incentive models.

Hybrid compute unit design – fundamental elements



Hong-Linh Truong, Hoa Khanh Dam, Aditya Ghose, Schahram Dustdar "Augmenting Complex Problem Solving with Hybrid Compute Units",9th International Workshop on Engineering Service-Oriented Application (WESOA's 2013), In conjunction with ICSOC 2013, Dec 2, 2013, Berlin, Germany, (c)Springer-Verlag

Highlights: Virtualizing Communication

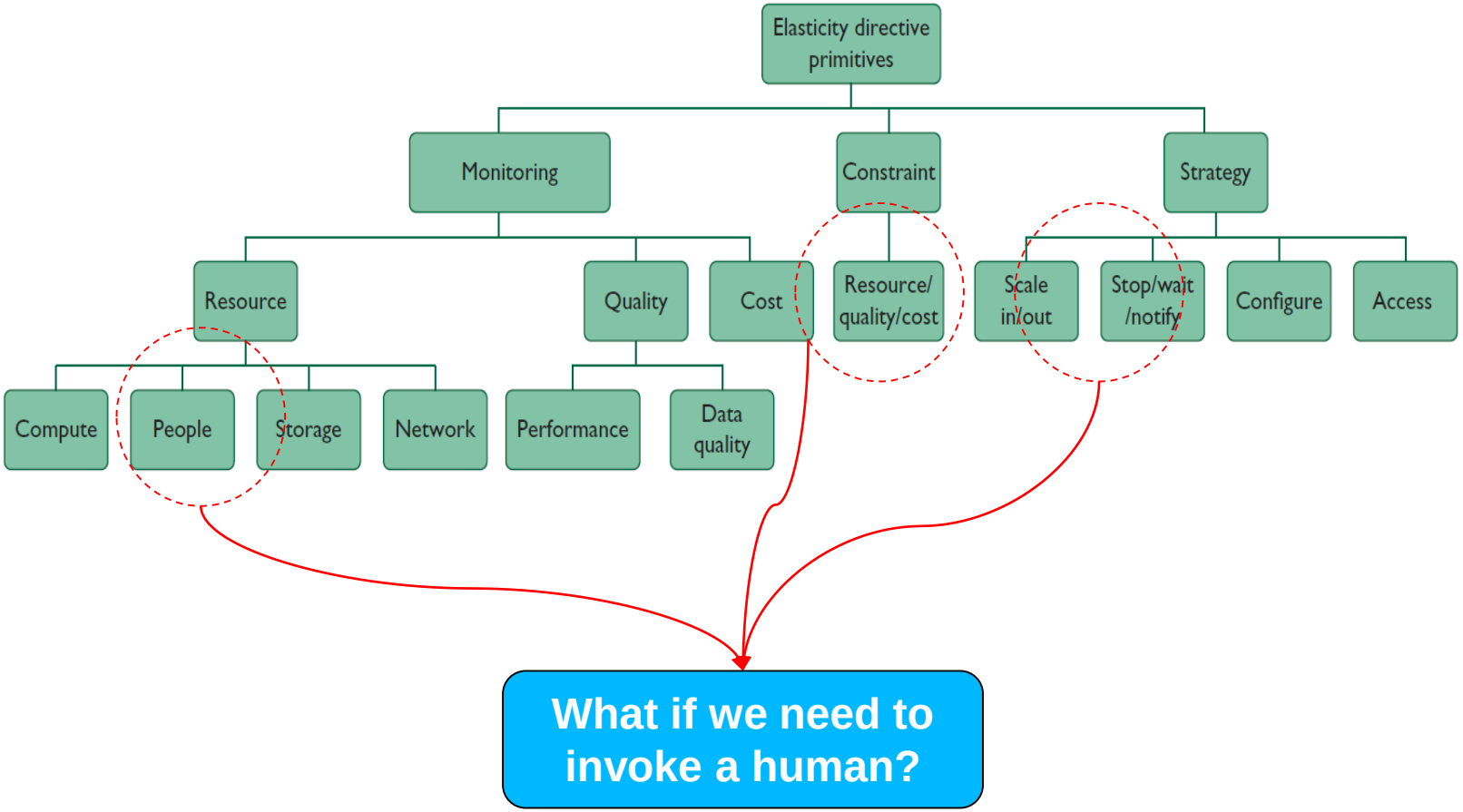


- Extensible architecture
  - Adapters for: email, Dropbox, REST, Android
- Integrated with WP4,6,8;
  API access for WP5,2
- Open source and documentation:
  - https://github.com/tuwiendsg/SmartCom

P. Zeppezauer, O. Scekic, H.-L. Truong and S. Dustdar, "Virtualizing Communication for Hybrid and Diversity-Aware Collective Adaptive Systems," *10th International Workshop on Engineering Service-Oriented Applications (WESOA'14@ICSOC),* Paris, 2014.

Zeppezauer, Virtualizing Communication for Hybrid and Diversity-aware Collective Adaptive Systems, Master thesis, Dec 2014.

Specifying and controling elasticity of human-based services



SYBL extension

Notification description

```
Notification := notificationID:NOTIFY Role WHEN ComplexCondition
                          : notify(NotificationType, message)
Role := ROLE(Responsability1, Responsability2), Role |
        ROLE (Responsability1, Responsability2)|
        RoleX, Role | RoleX
NotificationType := NOTIFICATION | ERROR | WARNING
```

Notification directive example

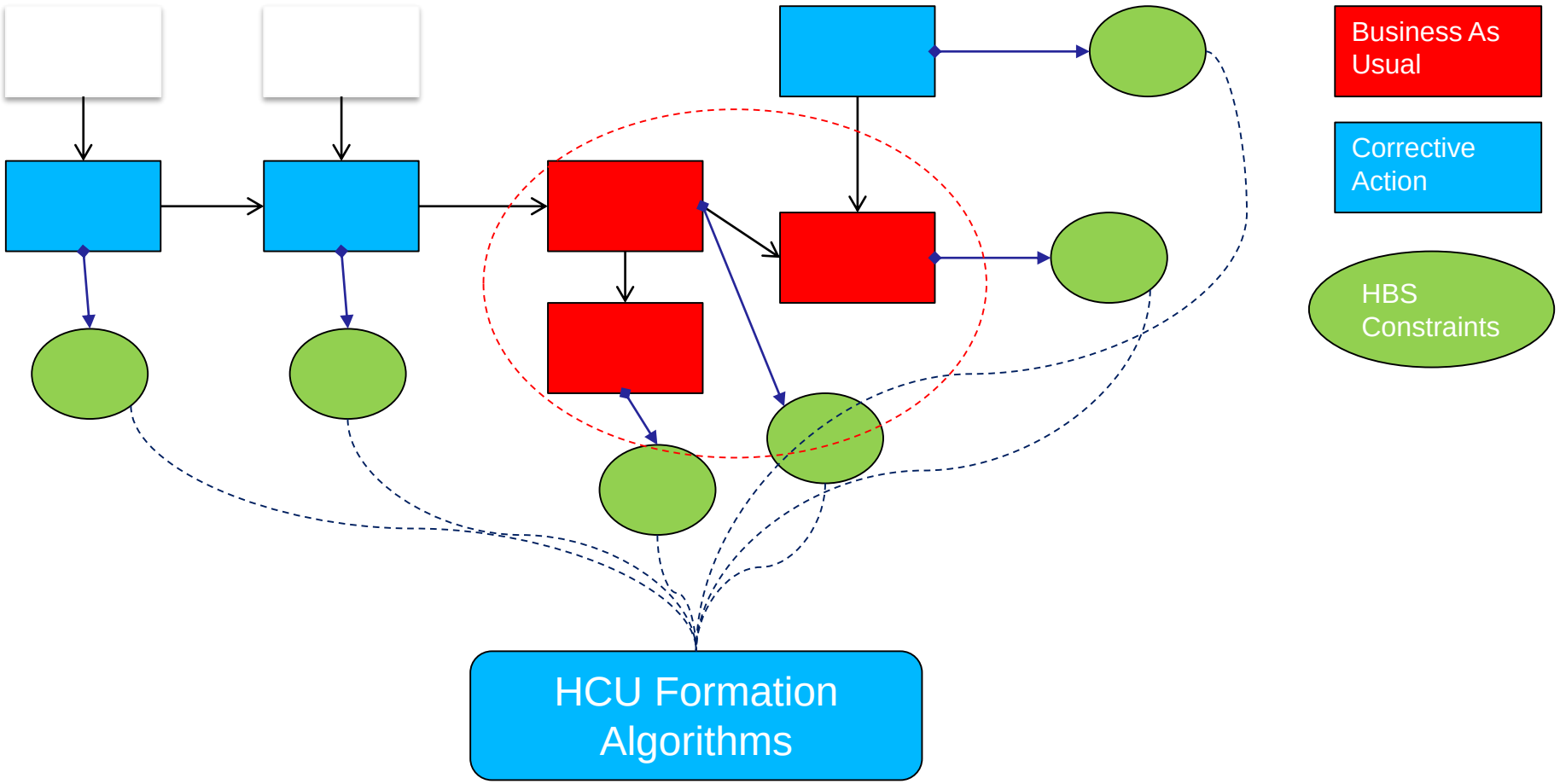
```
No1:NOTIFY OperationsManager WHEN responseTime > 1.2 s :
  notify(WARNING, "Response time exceeds 1.2 s")
```

Georgiana Copil, Hong Linh Truong, Schahram Dustdar: Supporting Cloud Service Operation Management for Elasticity. ICSOC 2015: 123-138

Selecting human units

- Do not select at all
  - Let human units bid the tasks
    - E.g., in crowdsourcing platforms
- Static/fix mapping
  - E.g., using static information for human-task mapping
- Simple selection techniques
  - Using the requirement of the task to find the suitable human units based on their capabilities
- Complex selection techniques
  - Utilizing complex dependency graphs to find suitable human units

Selecting HCU based on task graphs



Hong Linh Truong, Schahram Dustdar, Kamal Bhattacharya: Programming Hybrid Services in the Cloud. ICSOC 2012: 96-110

Placement techniques for human units

- Usually at design time the developer/designer decides
  - Where to put human units
  - Where some triggers should be put in order to invoke human units if needed
- At runtime
  - Find suitable human units
  - Invoke human units
- Placement of human units
  - Application-specific
  - Needs automatic algorithms and supporting tools

Eloquent: AI + Human Tasks

- https://www.eloquent.ai/
- Combine AI with human for “task-oriented dialog AI”
- In the domain of customer support

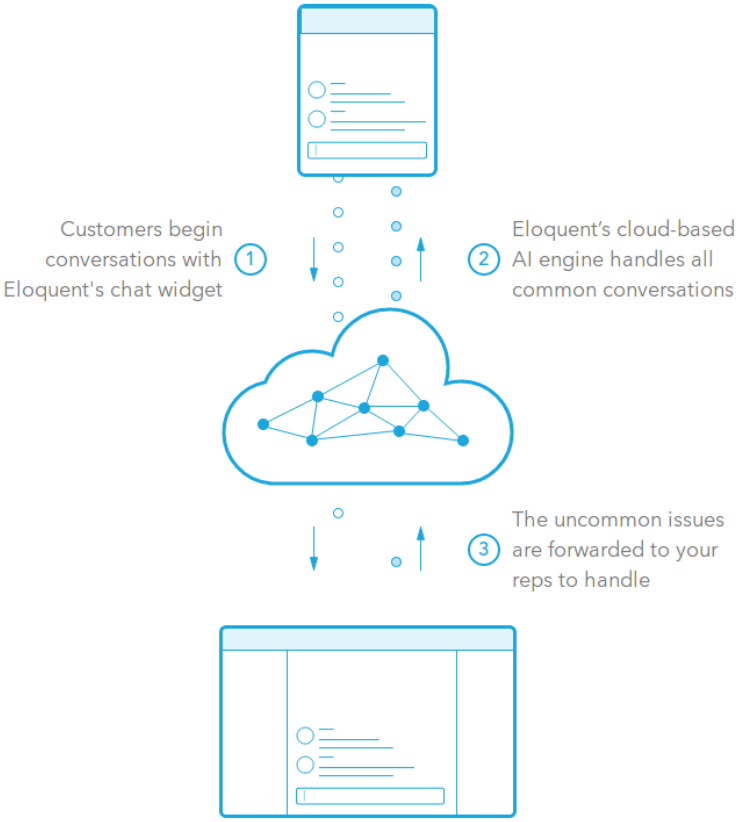


Figure source: https://www.eloquent.ai/

Exercises

- Read mentioned papers
- Analyze pros and cons of existing frameworks for data analytics
- Survey existing algorithms for matching human units to data analytics tasks
- Examine requirements for locating places for human units and implement some algorithms
- Examine monitoring techniques for cloud of human compute units

Thanks for your attention

Hong-Linh Truong
Distributed Systems Group, TU Wien
truong@dsg.tuwien.ac.at
http://dsg.tuwien.ac.at/staff/truong