Modeling Elasticity Trade-Offs in Adaptive Mixed Systems

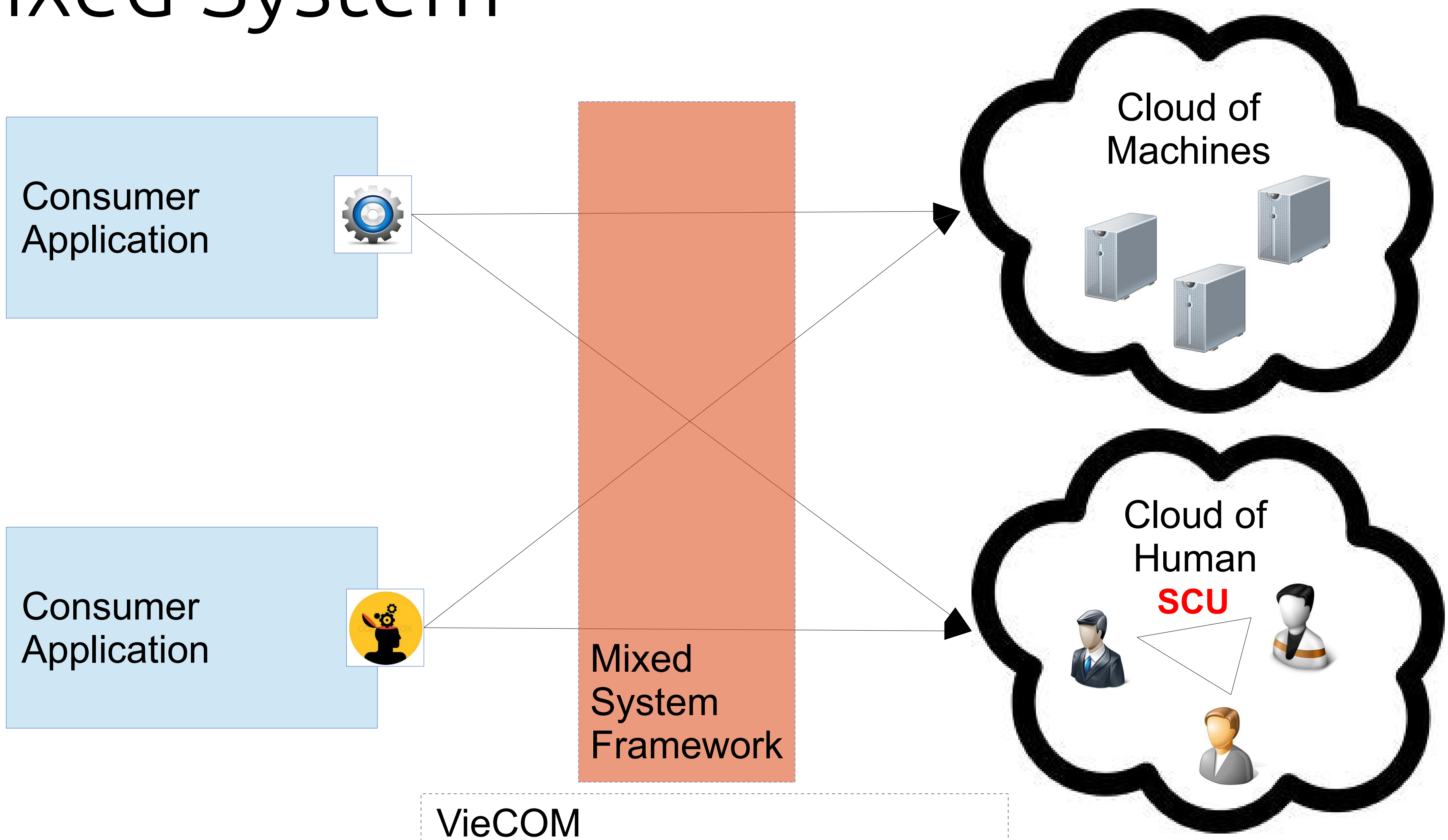
Muhammad Candra, Hong-Linh Truong, Schahram Dustdar

Distributed System Group
Vienna University of Technology

Outline

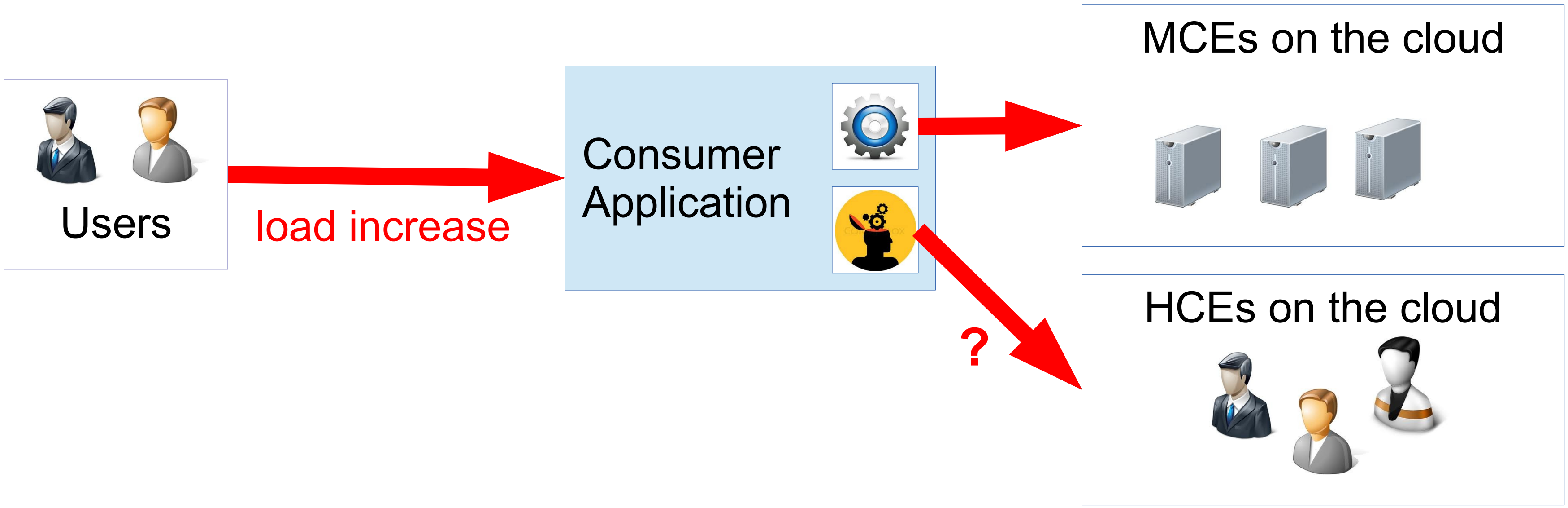
- Introduction
  - Mixed System
  - Elasticity
  - Motivation
- Elasticity Profile
  - Constructs
  - Binding
- Runtime Framework
  - Adaptive Mixed System Framework
- Example
- Conclusion & Future Work

Mixed System



VieCOM
Vienna Elastic Computing Model
- Virtualization
- SCU Management
- Quality Control Strategy
- **Elasticity**

Elasticity in Mixed System

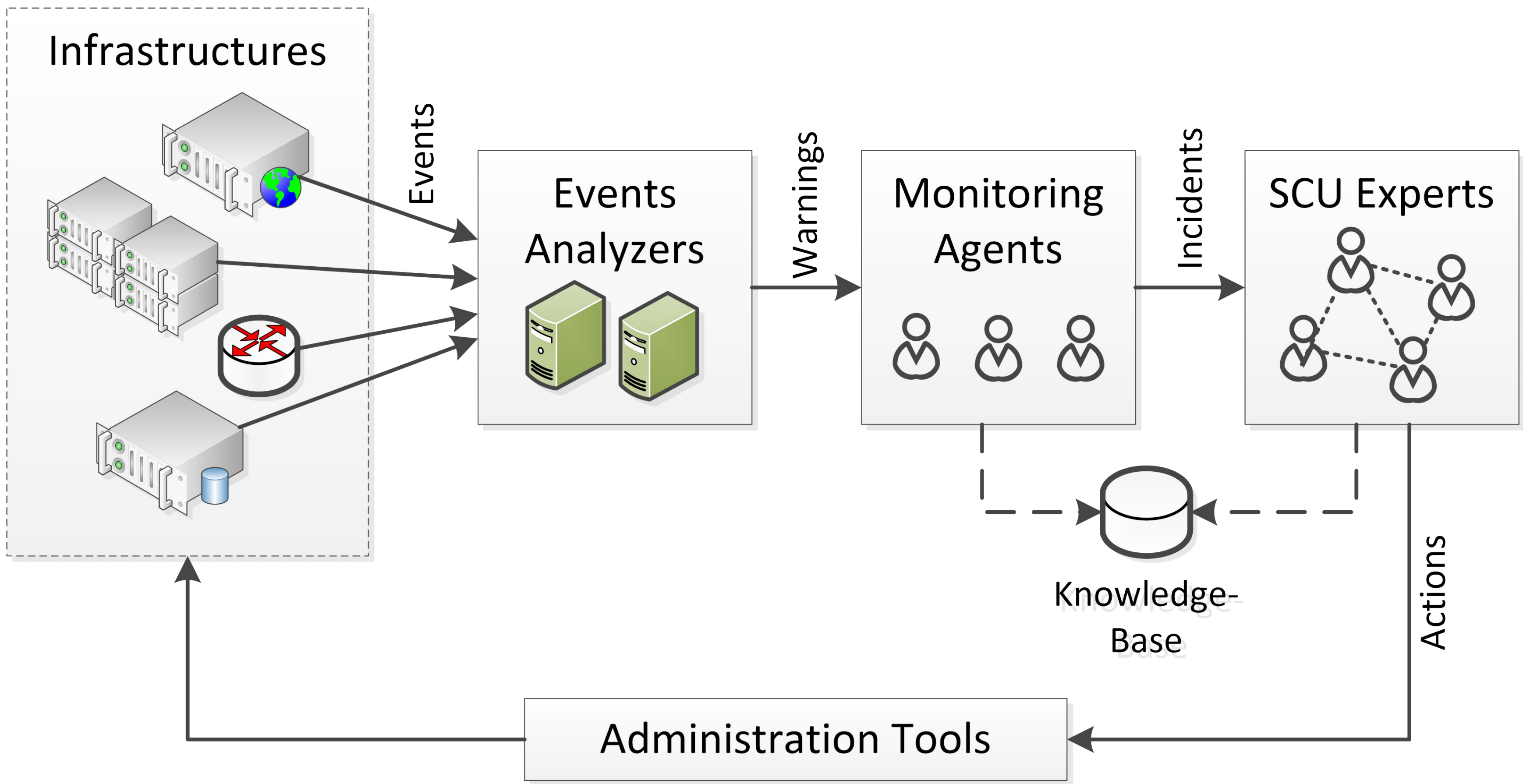


Elasticity dimension: Quality + Resources Scalability + Cost

- When the average utilization of the human workers on a running pool is above 8 hours per day, then additional workers must be assigned to the pool
- A human-task requester wants to pay a cheaper price if the worker takes more than 1 hour to finish the task.

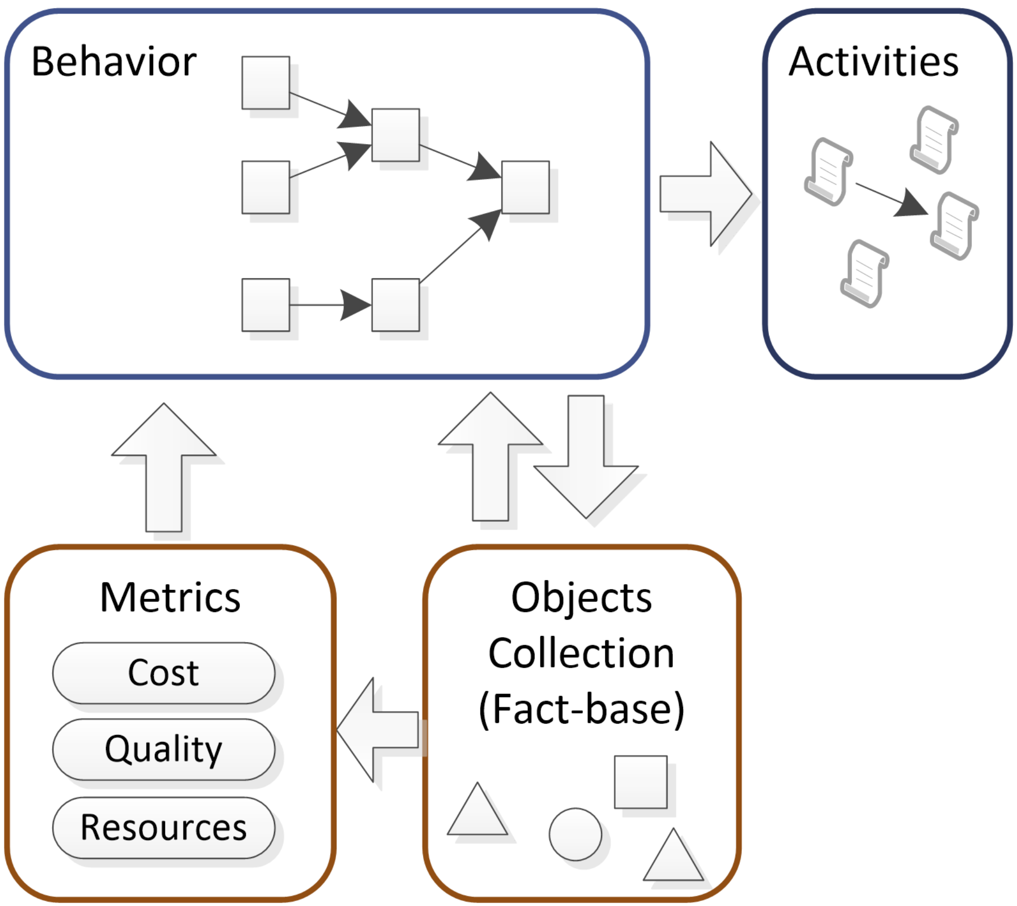
Motivation

SCU-based IT Infrastructure Monitoring and Management



We propose to model the behavior using ELASTICITY PROFILE

Constructs of Elasticity Profile



in Production Rule System:

Collection of Rules

Collection of Facts
(working memory)

Elasticity Profile

- Objects
  - Objects represent any component of a system or a process that can behave elastically
  - MCEs: machine instances, storages, etc.
  - HCEs: human workers, human-based tasks, etc.

- Metrics
  - Metrics represent the quality, resource, and cost properties of the objects.

Metric Dimension	*Machine Metrics*	*Human Metrics*
Resources	Number of resources, utilization, storage capacity, bandwidth capacity	Number of resources, utilization
Quality	Response time, throughput, availability	Response time, rating, availability, throughput, task acceptance rate
Cost	Cost / API calls, virtual instance / hours	Task price, hourly price

Elasticity Profile

- Behavior
  - Rules for defining adaptation strategy
  - Contains condition and consequence
- Activities
  - Assignment
  - Assertion
  - Invocation
  - Exception

$$\forall (worker, pool)$$
$$Worker(worker) \wedge ActivePool(pool) \wedge$$
$$IsMember(worker, pool) \wedge$$
$$HourUtilization(worker) >= 8$$
$$\Rightarrow AddWorker(pool)$$

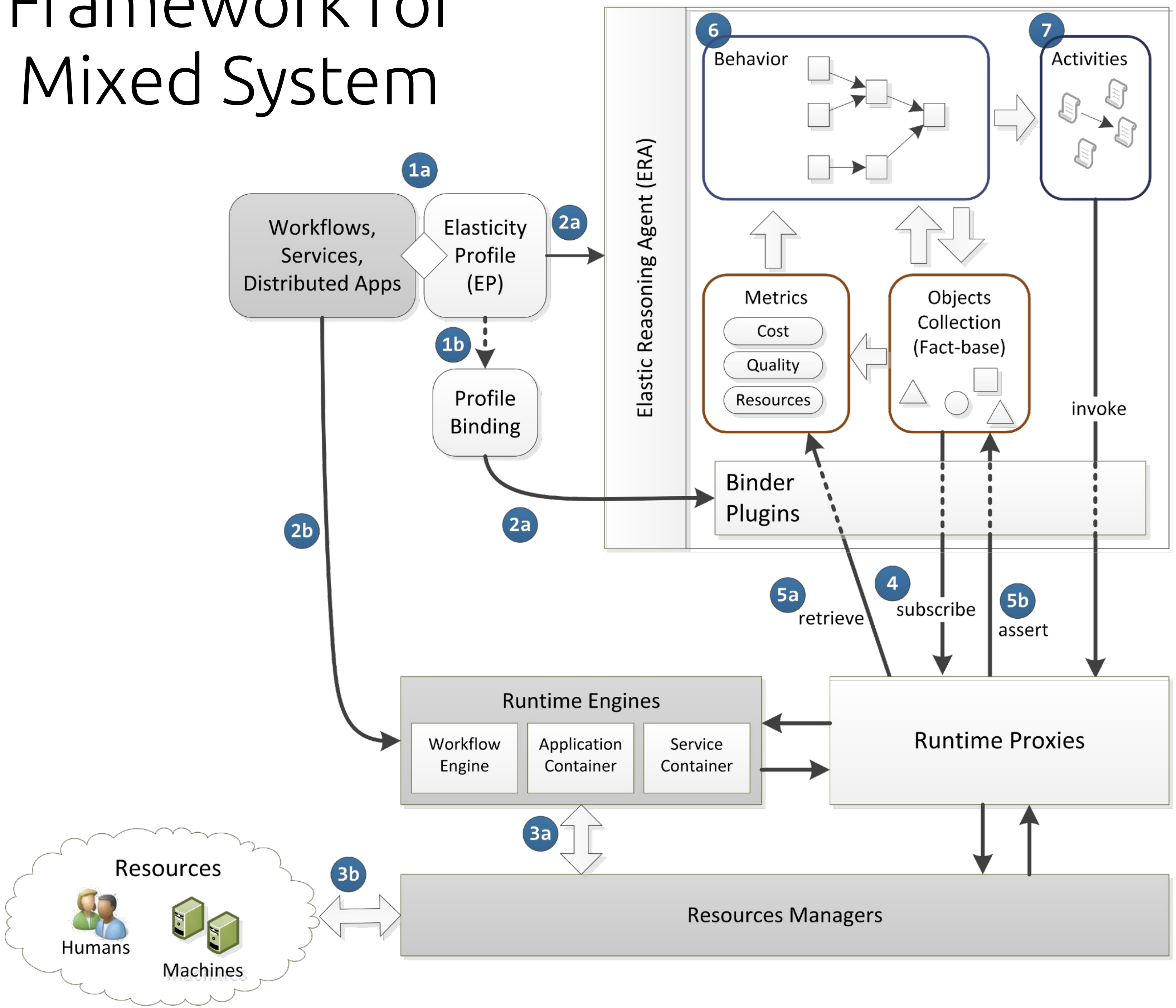
EP Grammar

```
⟨objects_statement⟩ ::= objects { ⟨objects_list⟩ } ;

⟨behavior_statement⟩ ::= behavior { ⟨implication_list⟩ } ;

⟨implication_list⟩ ::= ⟨implication⟩
    |  ⟨implication⟩ ; ⟨implication_list⟩

⟨implication⟩ ::= check [:⟨priority⟩] (⟨condition⟩) { ⟨consequences⟩ }

⟨consequences⟩ ::= ⟨consequence⟩ | ⟨consequence⟩; ⟨consequences⟩

⟨consequence⟩ ::= ⟨metric_identifier⟩ = ⟨value⟩
    |  assert ⟨instance_identifier⟩
    |  trigger ⟨action_identifier⟩(⟨value_list⟩)
    |  throw ⟨exception_identifier⟩(⟨value⟩)

⟨activities_statement⟩ ::= activities { ⟨activities_list⟩ } ;

⟨activities_list⟩ ::= ⟨activity⟩
    |  ⟨activity⟩, ⟨activities_list⟩

⟨activity⟩ ::= ⟨activity_identifier⟩(⟨activity_param_list⟩)
```

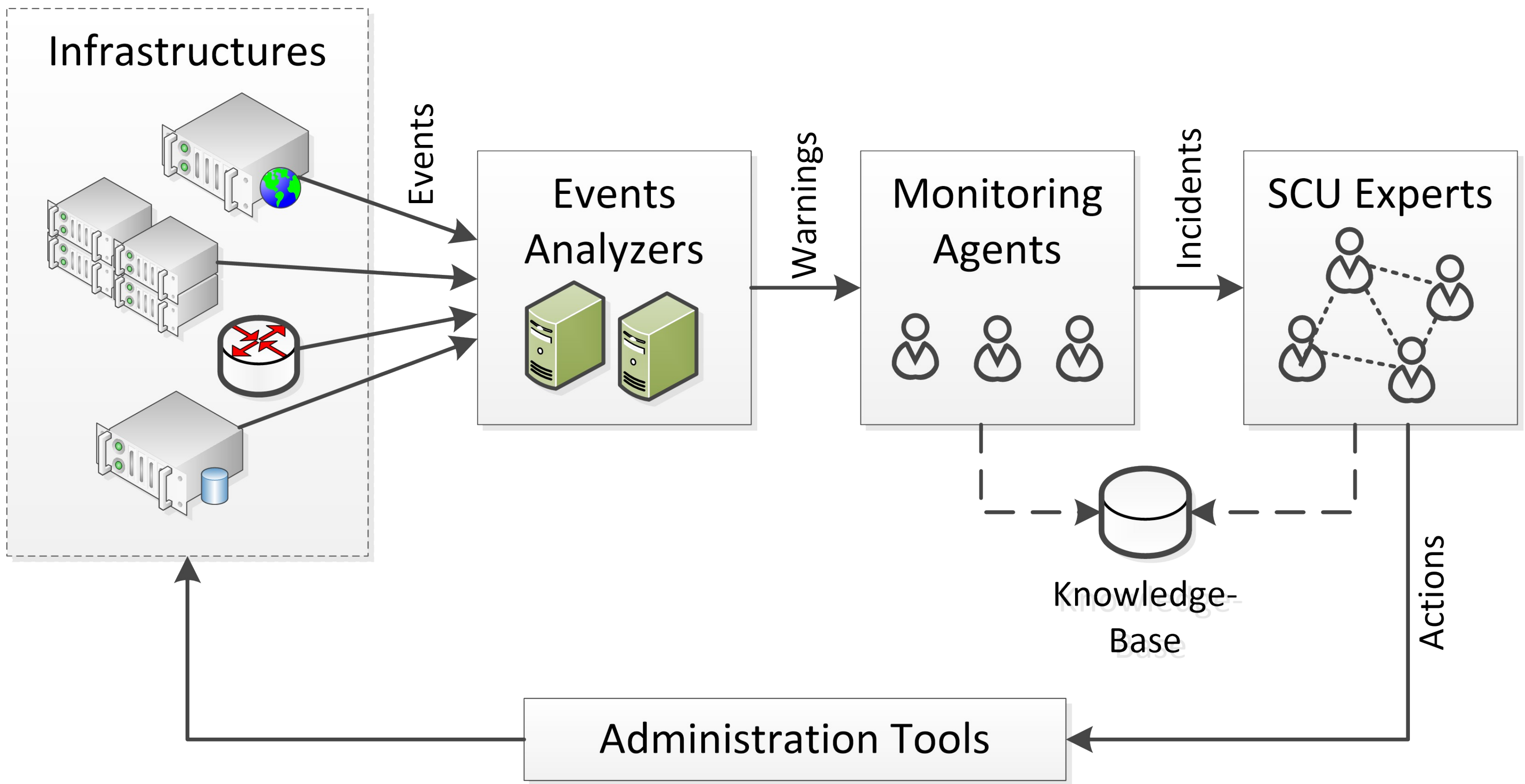
Binding

- Profile and runtime binding are separated
- Protocol: SOAP, RESTful, Java RMI
- Objects binding
  - Subscription to event notification
- Metrics binding
  - Remote getter and setter
- Activity binding
  - Remote method invocation

Runtime Framework for Adaptive Mixed System



Example



Infrastructures
Events
Events Analyzers
Warnings
Monitoring Agents
Incidents
SCU Experts
Knowledge-Base
Actions
Administration Tools

Example

```
profile SCU_IT_Management {
  objects {
    Customer, Event, Warning, Incident, Analyzer,
    MonitoringAgent, ExpertSCU
  };
  metrics {
    Customer has ServiceType, ...;
    Incident has Lifetime, ...;
    Analyzer has Utilization, Type, ...;
    ExpertSCU has ExpertiseLevel, ...;
    ...
  };
  actions {
    AddAnalyzer(ANALYZER_TYPE),
    ReduceAnalyzer(ANALYZER_TYPE),
    AddMonitoringAgent(),
    ReduceMonitoringAgent(),
    UpgradeSCU(ExpertSCU, EXPERTISE_TYPE),
    TimeoutException(Incident),
    ...
  };
```

Example

```
behavior {

  /* Dynamically scale analyzer for premium
     service based on the average utilization
     of the premium analyzers */
  check (Number(doubleValue > 0.8)
         from accumulate(
           Analyzer(Type==PREMIUM_MACHINE and
                    u:Utilization),
                    average(u))) {
    /* scale up */
    trigger AddAnalyzer(PREMIUM_MACHINE);
  };
  check (Number(doubleValue < 0.2)
         from accumulate(
           Analyzer(Type==PREMIUM_MACHINE and
                    u:Utilization),
                    average(u))) {
    /* scale down */
    trigger ReduceAnalyzer(PREMIUM_MACHINE);
  };
```

Example

```
/* Scale monitoring agent based on the number of
   queued warnings */
check (Number(intValue > 20)
        from accumulate(w:Warning(), count(w))) {
   /* scale up */
   trigger AddMonitoringAgent();
};
check (Number(intValue < 5)
        from accumulate(w:Warning(), count(w))) {
   /* scale down */
   trigger ReduceMonitoringAgent();
};

/* Upgrade SCU when the deadline is
   approaching*/
check (Incident(Lifetime > 2 * 3600 and
     getCustomer().ServiceType==PREMIUM and
     scu:getAssignedSCU()) and
     (scu.ExpertiseLevel < HIGH_EXPERTISE) ) {
   /* increasing expertise level,
      i.e., it will add more experts with
      higher expertise */
   trigger UpgradeSCU(scu, HIGH_EXPERTISE);
};
```

Conclusion

- Elasticity Profile
  - Constructs for modeling adaptation strategy in mixed systems
- Elasticity Framework
  - Mechanism for deploying and executing adaptation strategy

Future Works

- Part of VieCOM (Vienna Elastic Computing Model)
  - Quality Control Strategy for SCU
  - Discovery and negotiation on elastic human-based services

Thank you

Acknowledgment
The first author of this paper is financially supported by
the Vienna PhD School of Informatics
http://www.informatik.tuwien.ac.at/teaching/phdschool