情報通信研究機構
National Institute of Information and Communications Technology

Elastic Resource Adaptation in the OpenStack Platform

Pedro Martinez-Julia, Ved P. Kafle, Hiroaki Harai
ペドロ・マルティネスフリア、ベド カフレ、原井 洋明
Network Science and Convergence Device Technology Laboratory, Network System Research Institute
National Institute of Information and Communications Technology (NICT)
{pedro,kafle,harai}@nict.go.jp

IEICE Technical Committee on Network Virtualization (NV)

7 March 2018

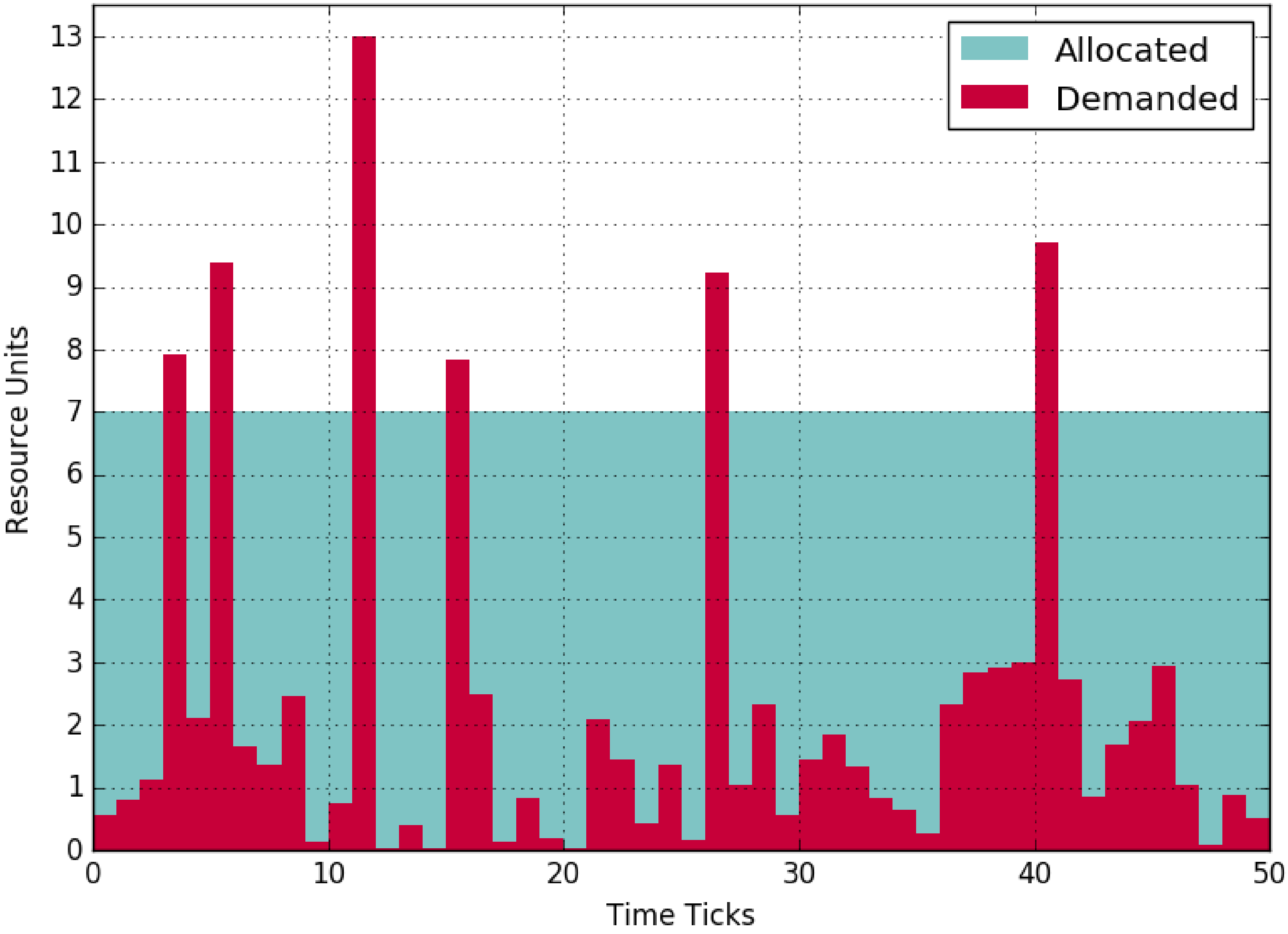
Outline

- Problem Statement:
  - Motivation and Research Topic
  - Use Case
- Solution:
  - Proposed Approach
  - Architecture Overview
  - Requirement Anticipation
  - Integration with ETSI-NFV-MANO.
- Conclusions & Future Work

Motivation and Research Topic (I)

Trivia:

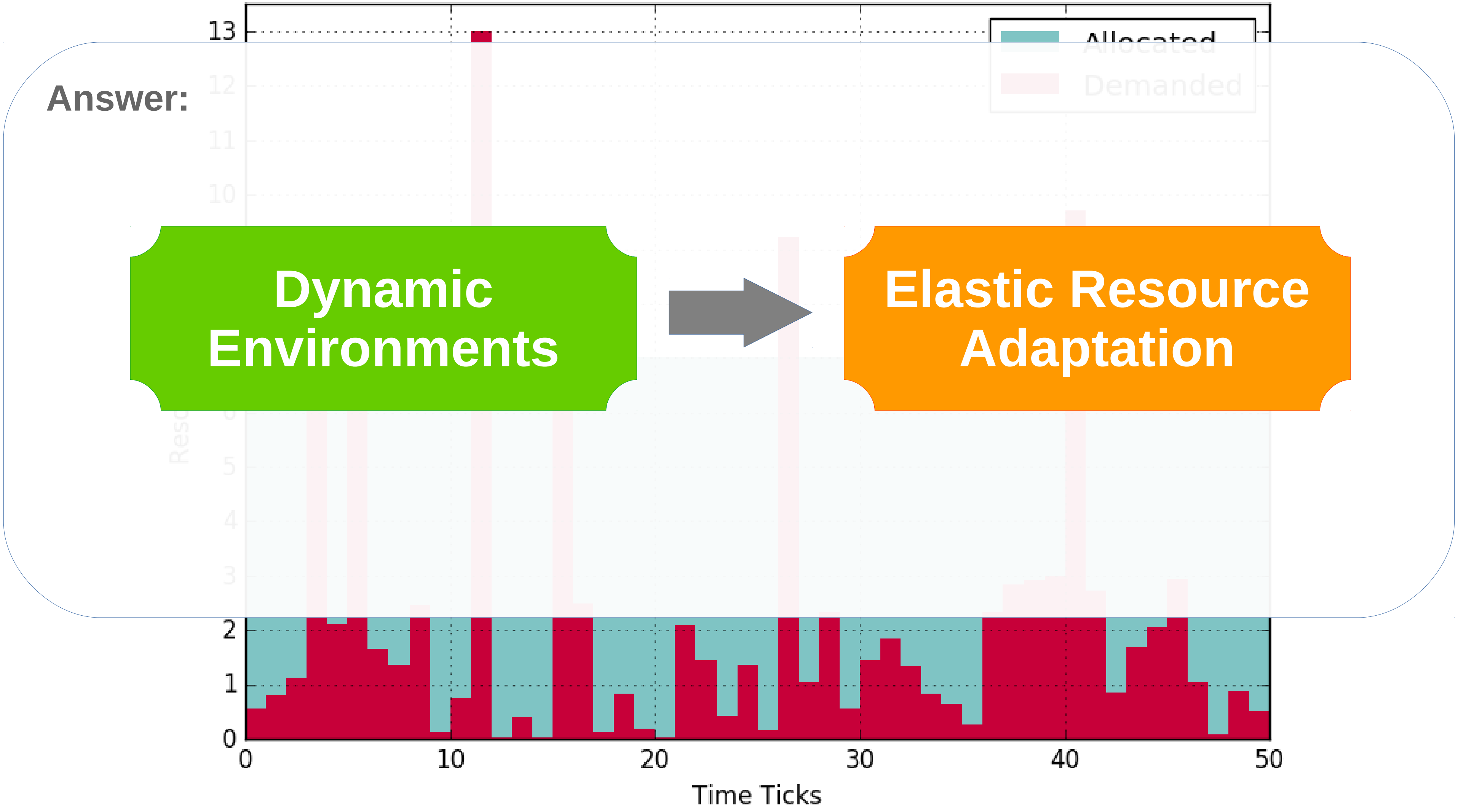
“High variation in resource demand” vs “Fixed resource allocation”.



Motivation and Research Topic (II)

Trivia:

“High variation in resource demand” vs “Fixed resource allocation”.



Answer:

Dynamic Environments → Elastic Resource Adaptation

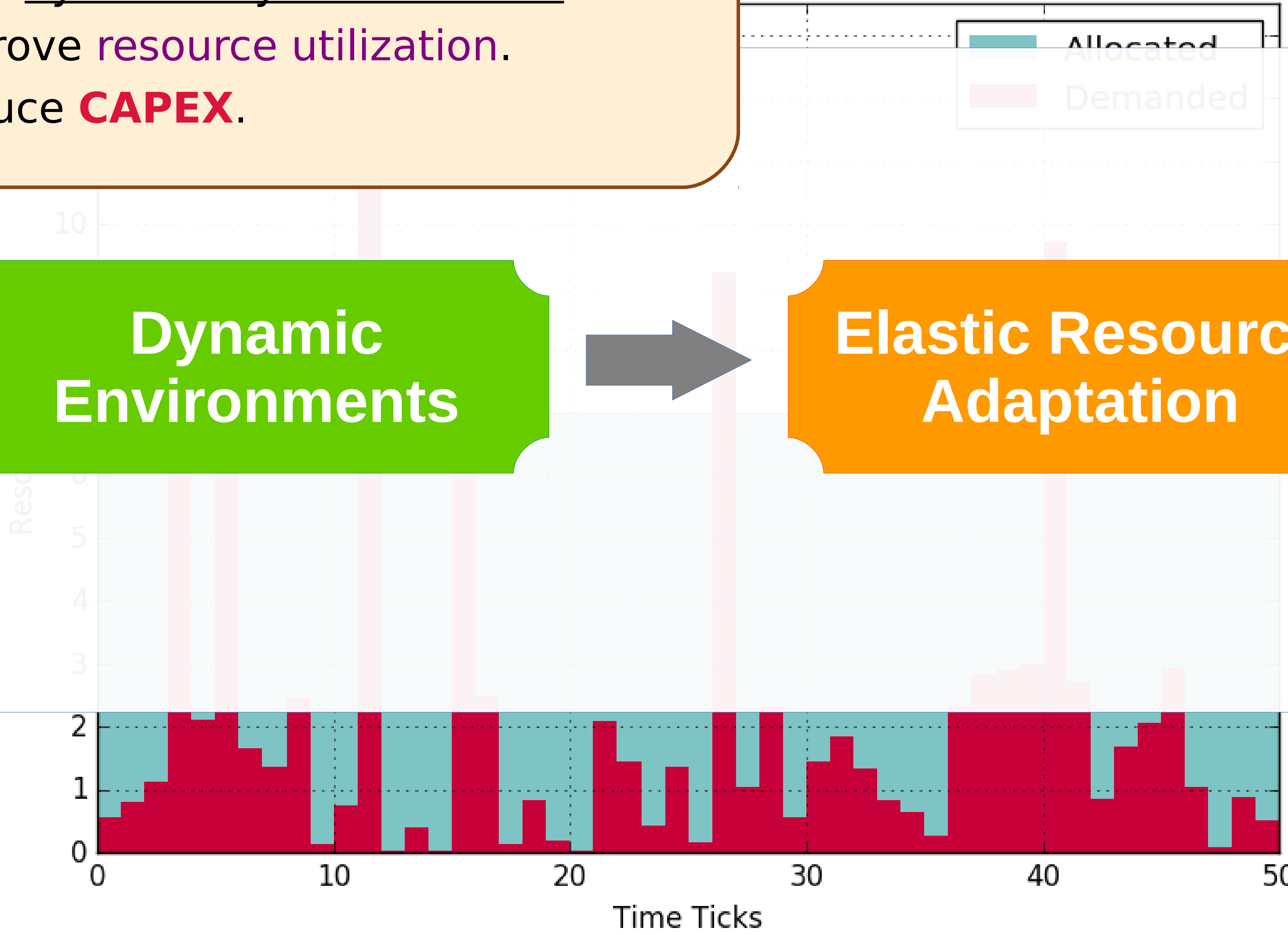
Motivation and Research Topic (III)

Trivia:

Virtual computer and network systems can be dynamically dimensioned to:

- Improve resource utilization.
- Reduce **CAPEX**.

vs “Fixed resource allocation”.



Dynamic Environments → **Elastic Resource Adaptation**

Motivation and Research Topic (IV)

Trivia:

vs “**Fixed resource allocation**”.

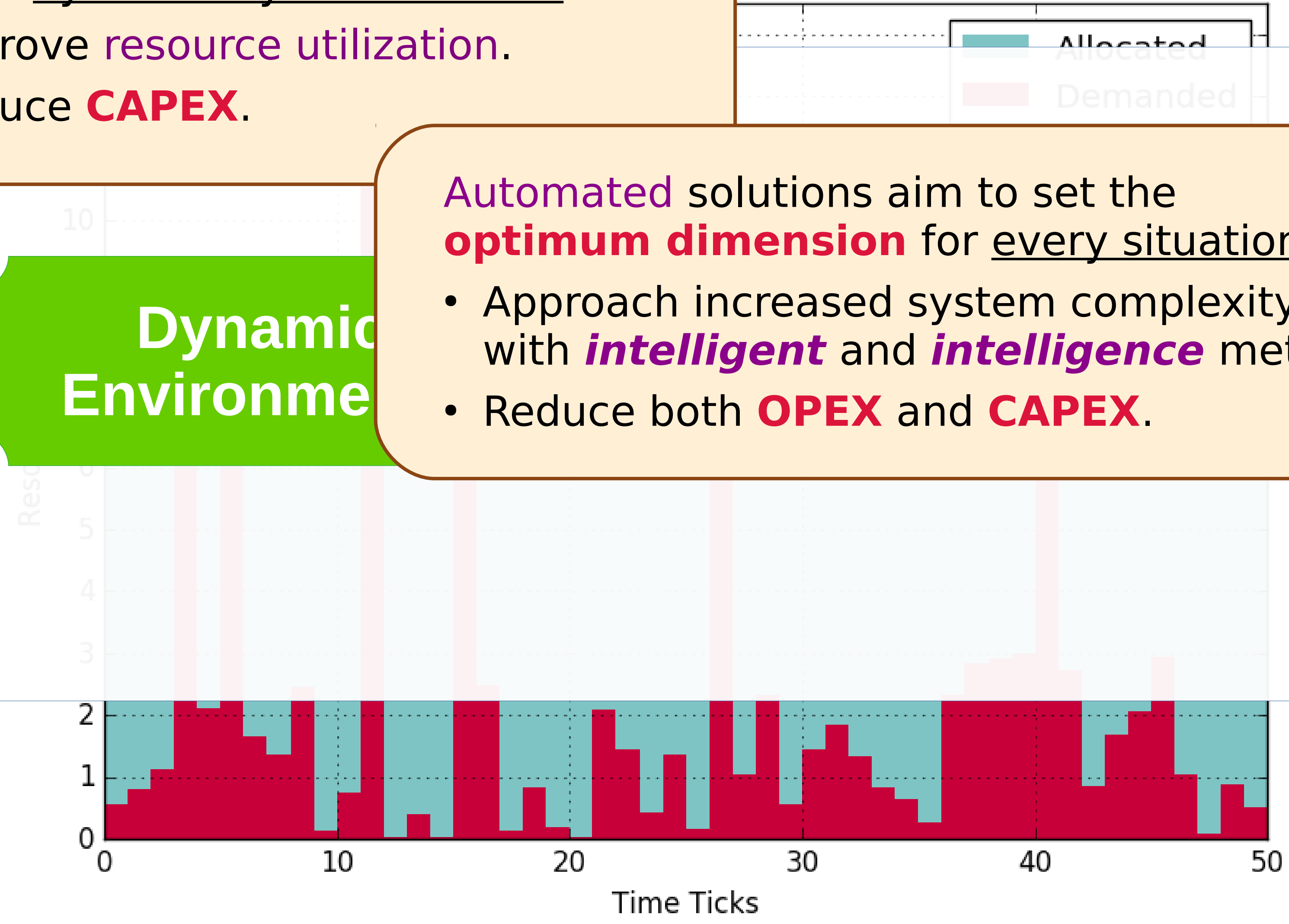
Virtual computer and network systems can be dynamically dimensioned to:

- Improve resource utilization.
- Reduce **CAPEX**.

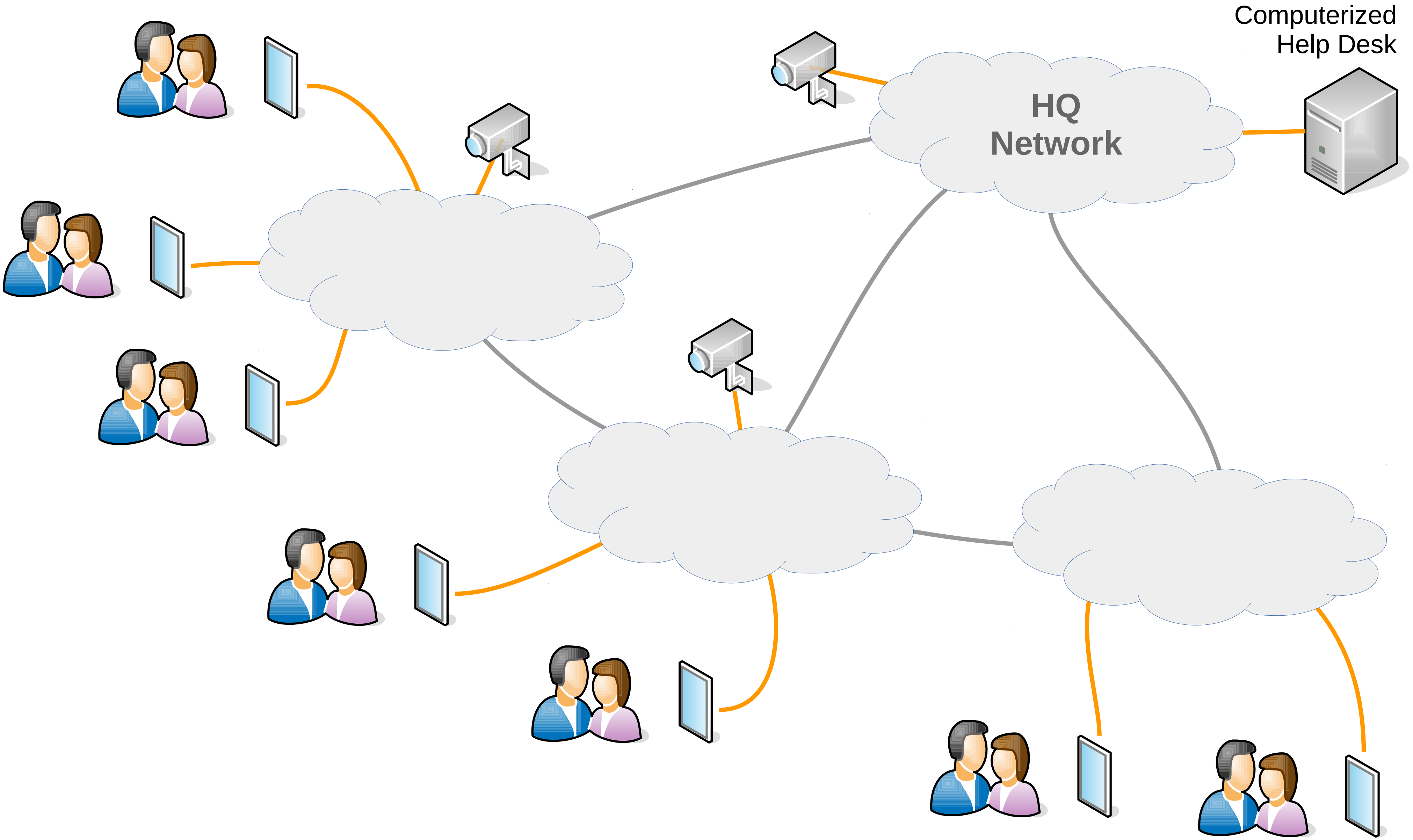
Dynamic Environme

Automated solutions aim to set the **optimum dimension** for every situation:

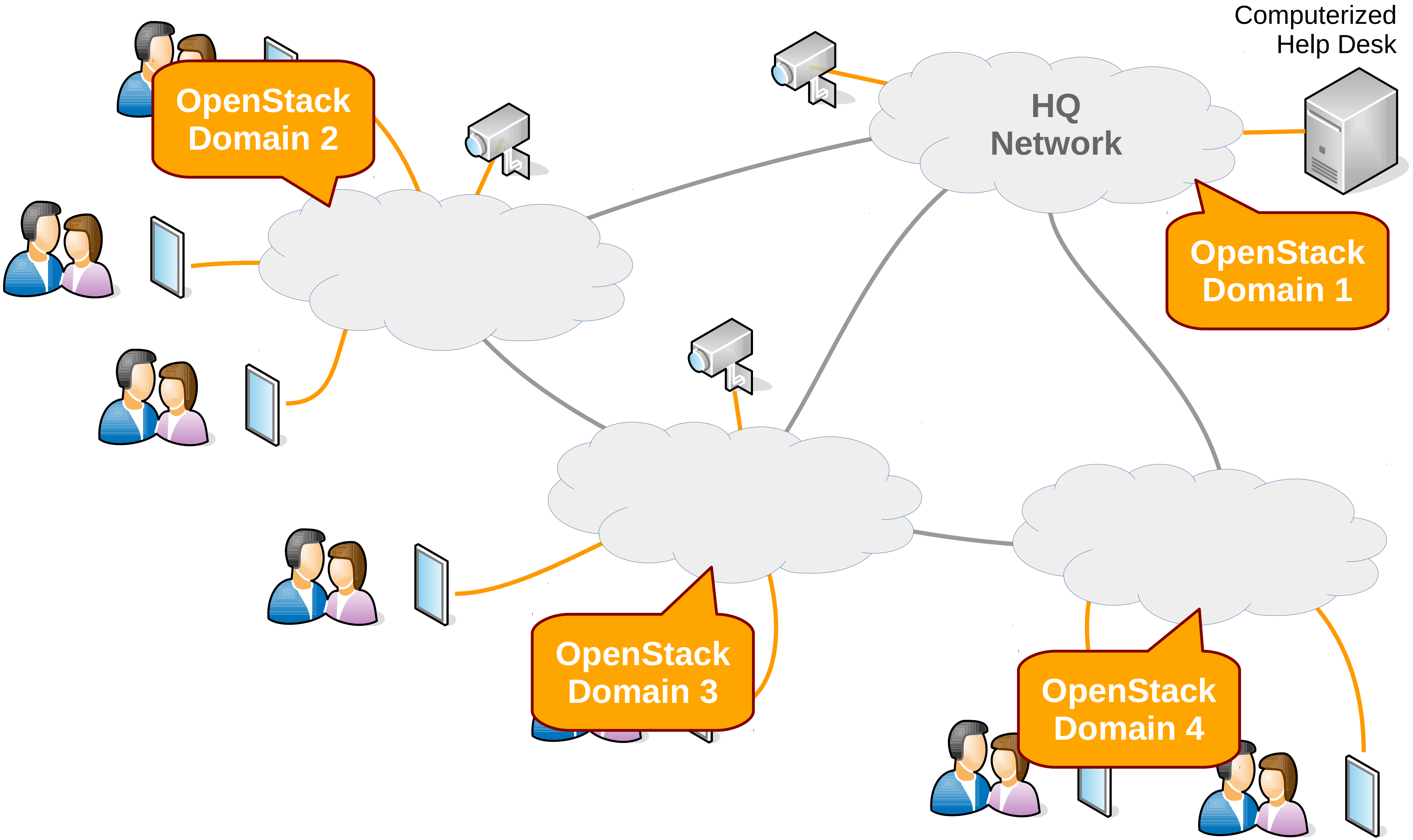
- Approach increased system complexity with ***intelligent*** and ***intelligence*** methods.
- Reduce both **OPEX** and **CAPEX**.



Use Case (I)



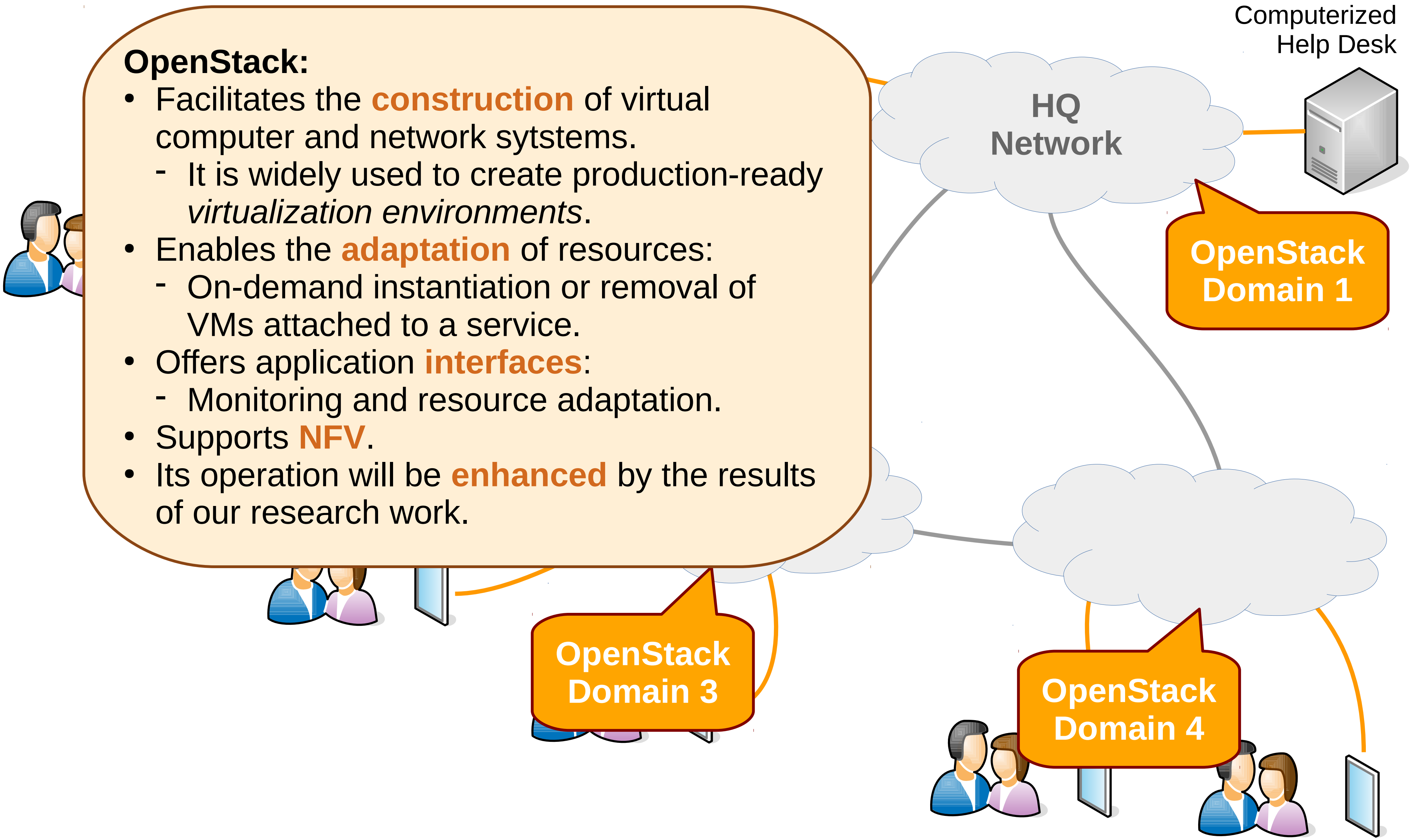
Use Case (II)


Computerized
Help Desk
HQ
Network
OpenStack
Domain 1
OpenStack
Domain 2
OpenStack
Domain 3
OpenStack
Domain 4

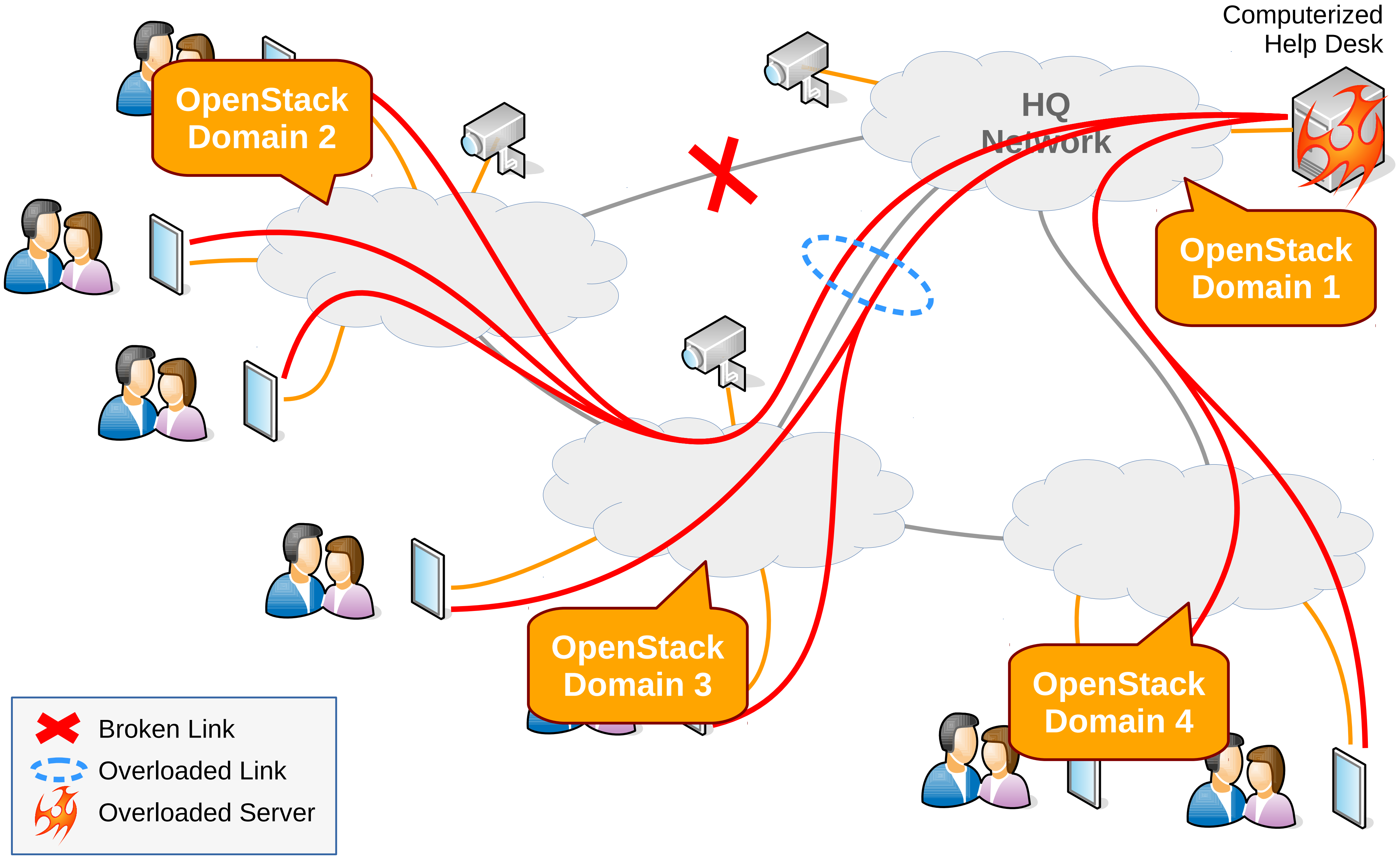
Use Case (III)

OpenStack:

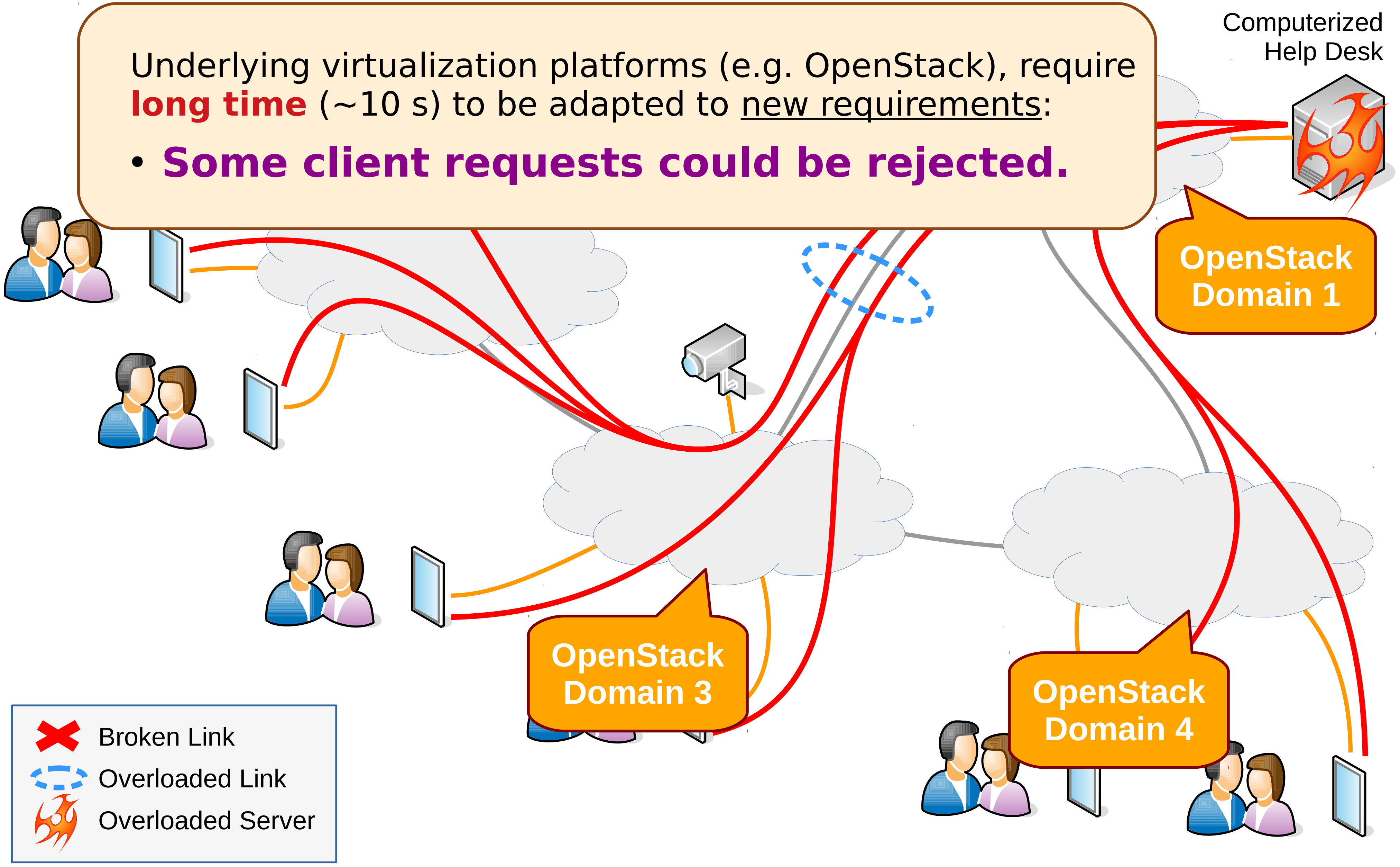
- Facilitates the **construction** of virtual computer and network sytstems.
  - It is widely used to create production-ready *virtualization environments*.
- Enables the **adaptation** of resources:
  - On-demand instantiation or removal of VMs attached to a service.
- Offers application **interfaces**:
  - Monitoring and resource adaptation.
- Supports **NFV**.
- Its operation will be **enhanced** by the results of our research work.



Use Case (IV)


Computerized Help Desk
HQ Network
OpenStack Domain 1
OpenStack Domain 2
OpenStack Domain 3
OpenStack Domain 4
Broken Link
Overloaded Link
Overloaded Server

Use Case (V)



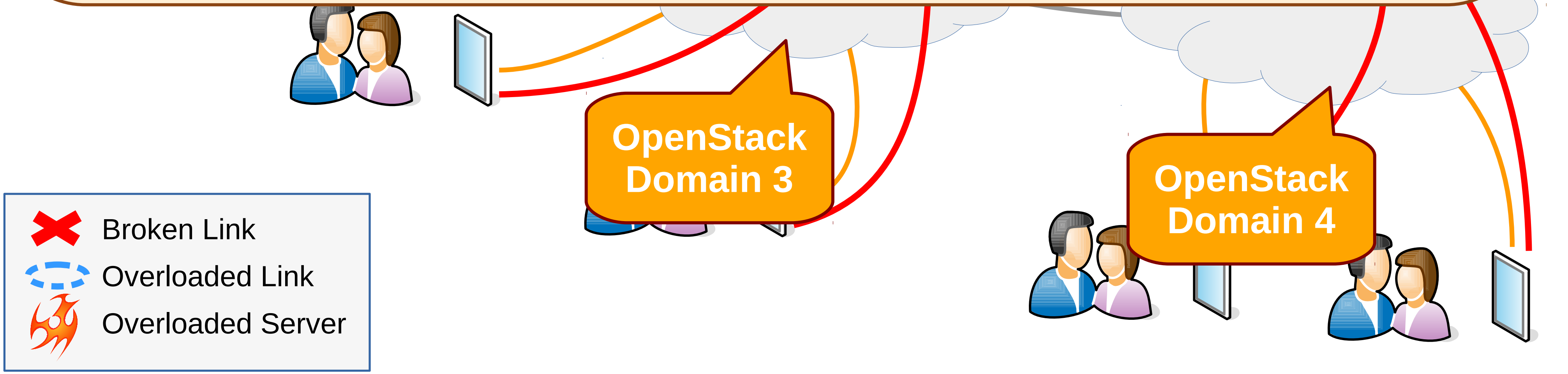
Underlying virtualization platforms (e.g. OpenStack), require **long time** (~10 s) to be adapted to new requirements:

- **Some client requests could be rejected.**

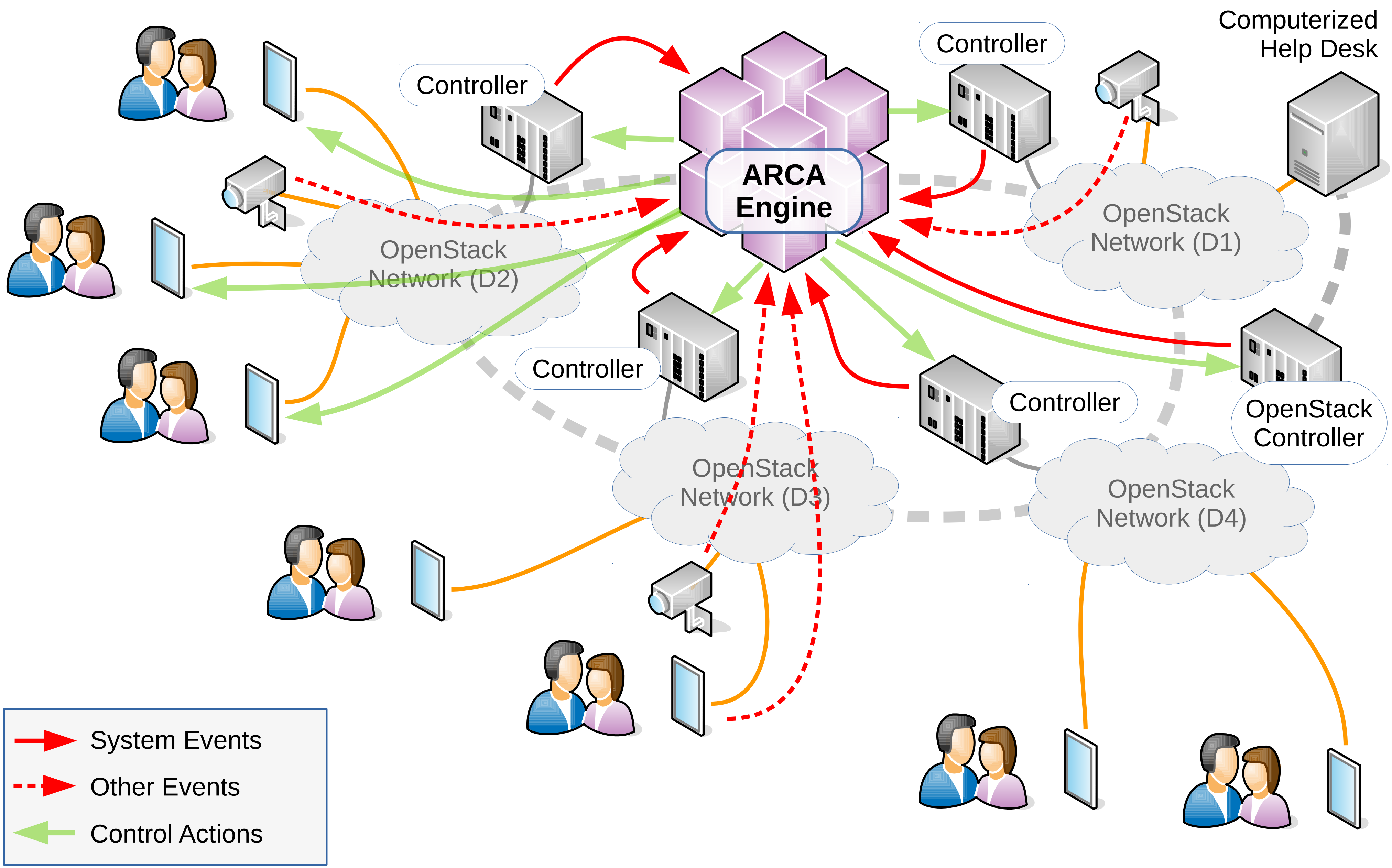
Use Case (VI)

Most changes in requirements are linked to events from outside the system:

- **User response can be derived from <u>event occurrence</u>.**
- **Required <u>resources</u> can be <u>anticipated</u> to reduce adaptation <u>delay</u> by noticing the events as soon as they occur.**
- **The system can be adapted <u>before</u> the client request <u>burst</u> actually <u>reaches</u> the <u>servers</u>.**



Proposed Approach (I)

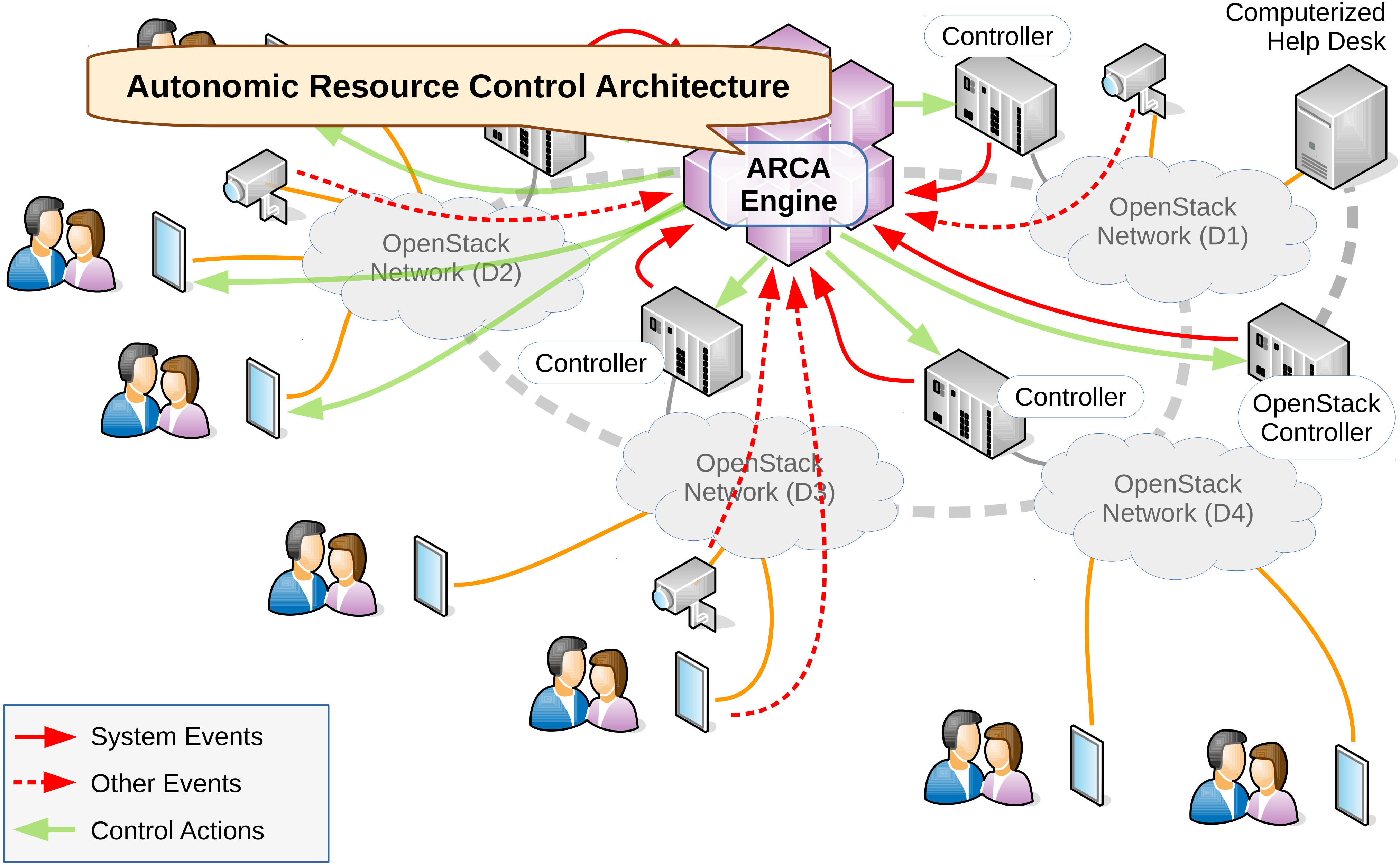


🡒 System Events

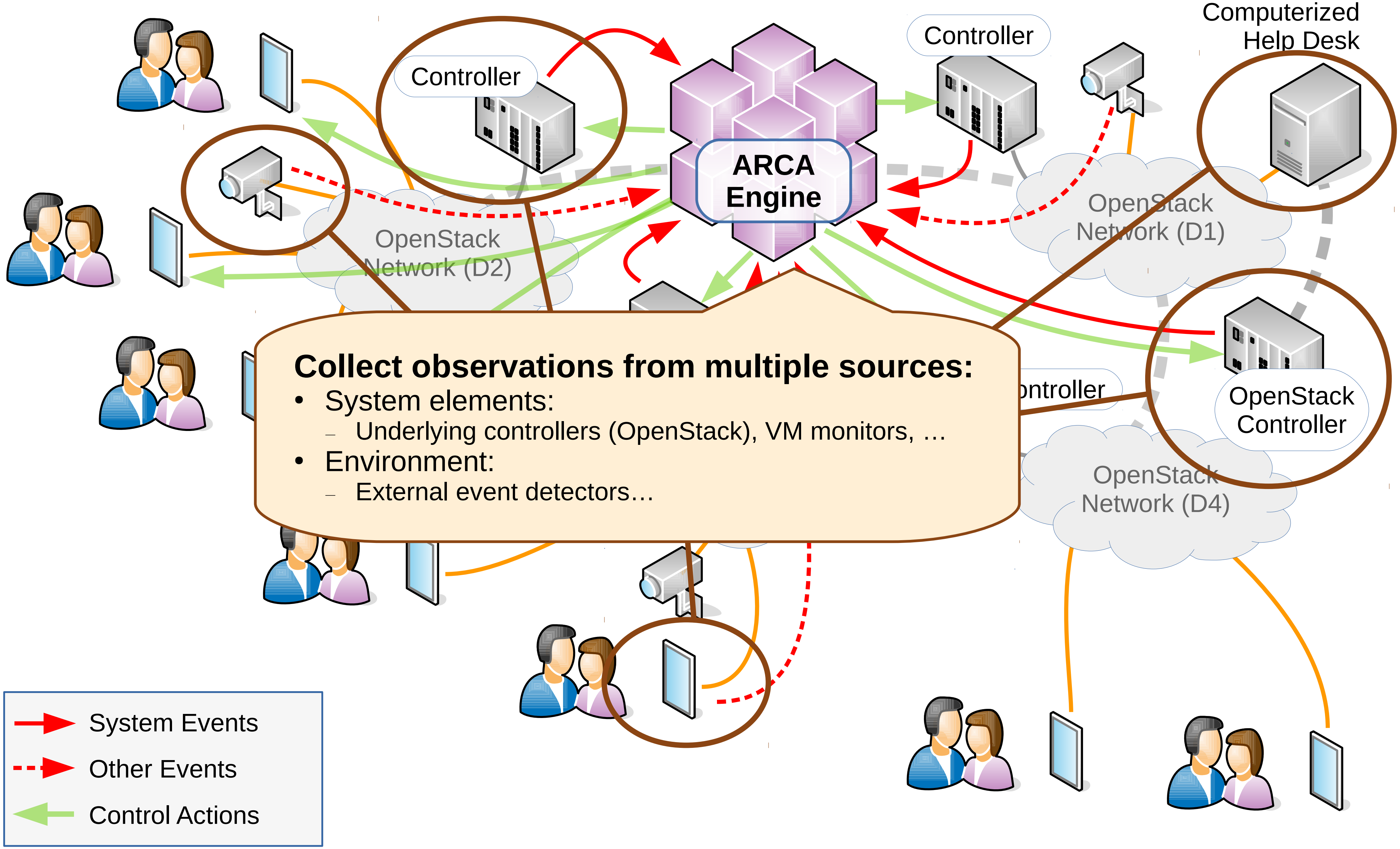
⇢ Other Events

🡐 Control Actions

Proposed Approach (II)



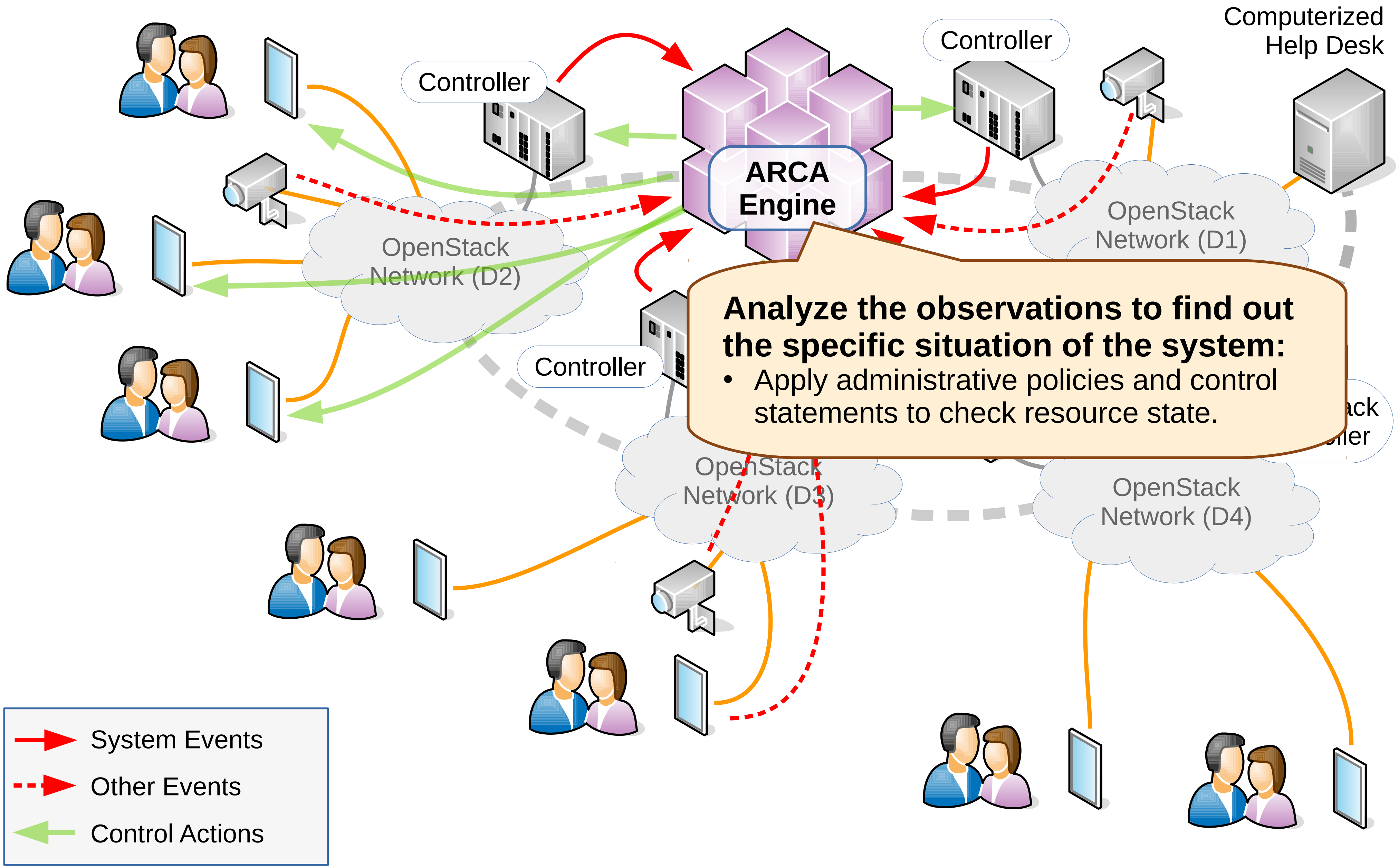
Proposed Approach (III)



Collect observations from multiple sources:

- System elements:
  - Underlying controllers (OpenStack), VM monitors, …
- Environment:
  - External event detectors…

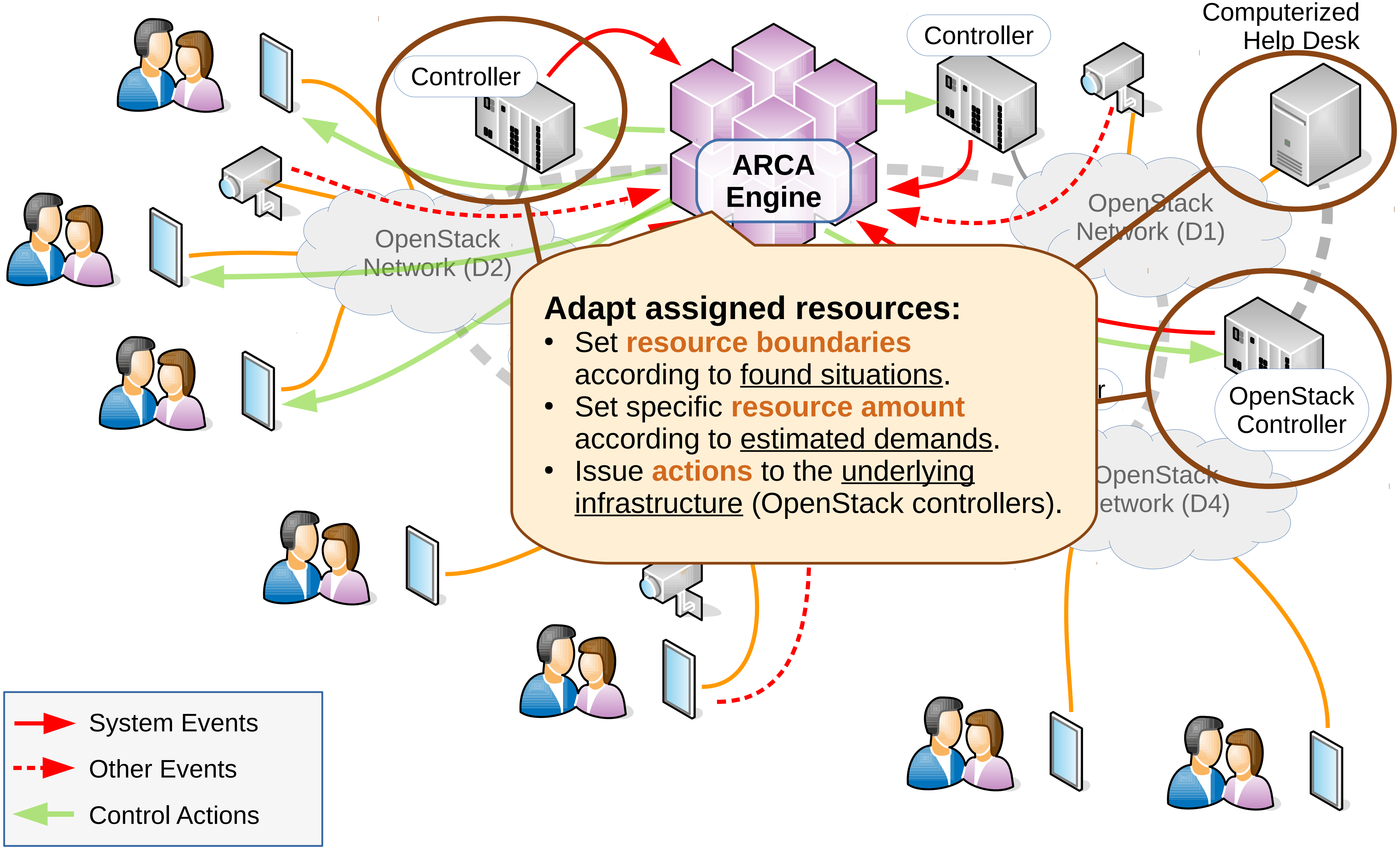
Proposed Approach (IV)



Analyze the observations to find out the specific situation of the system:

- Apply administrative policies and control statements to check resource state.

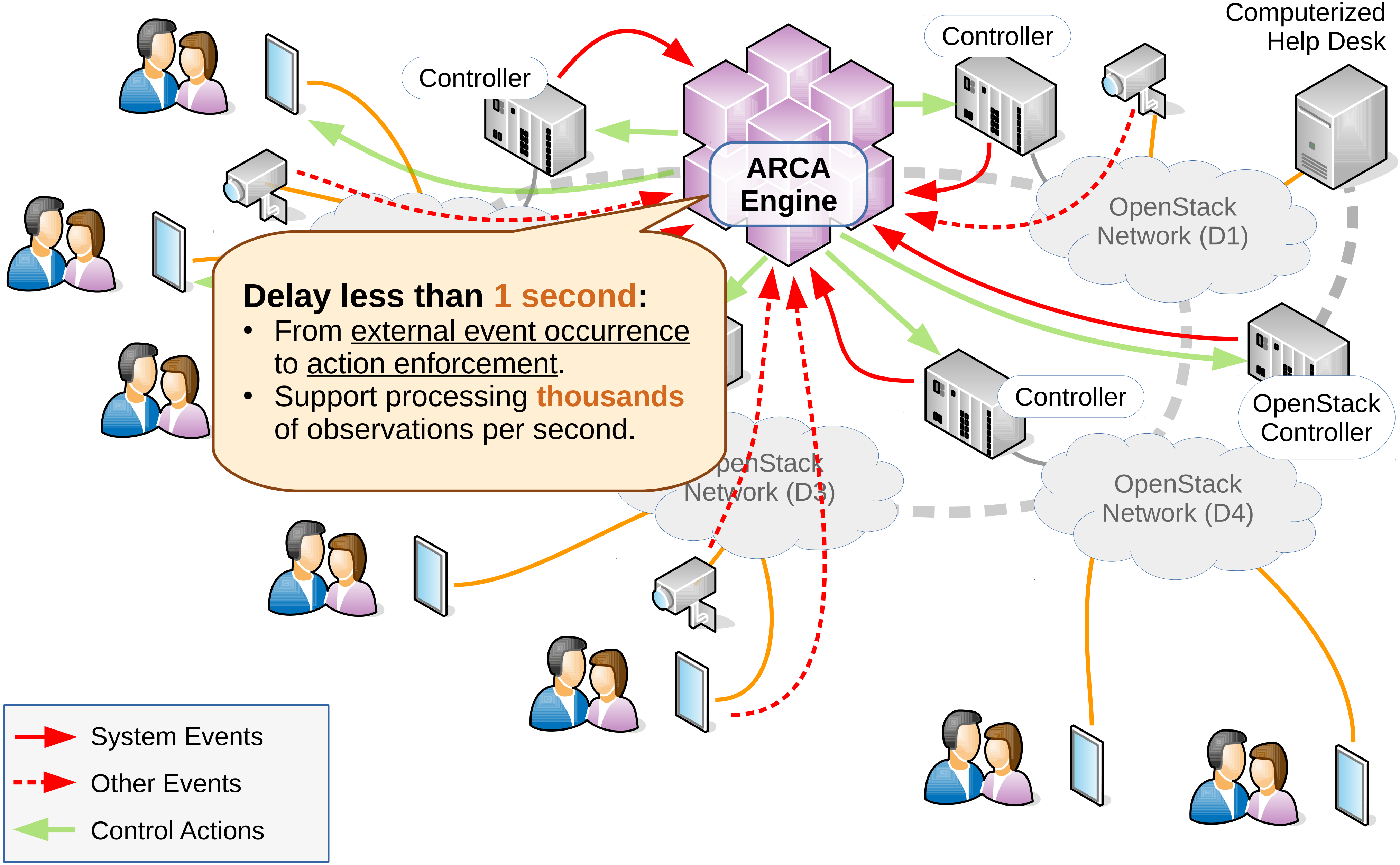
Proposed Approach (V)



Adapt assigned resources:

- Set **resource boundaries** according to found situations.
- Set specific **resource amount** according to estimated demands.
- Issue **actions** to the underlying infrastructure (OpenStack controllers).

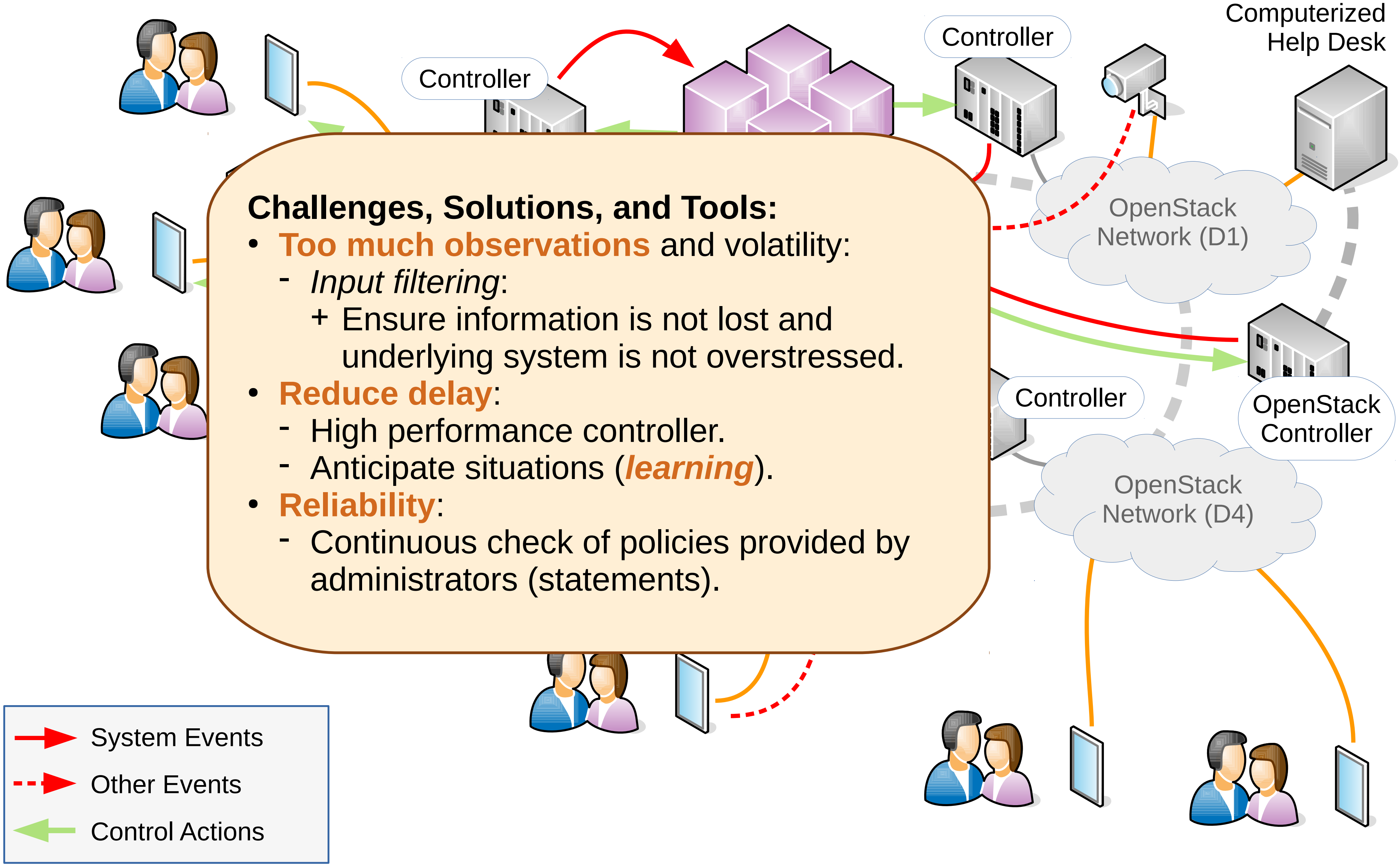
Proposed Approach (VI)



Delay less than 1 second:

- From external event occurrence to action enforcement.
- Support processing thousands of observations per second.

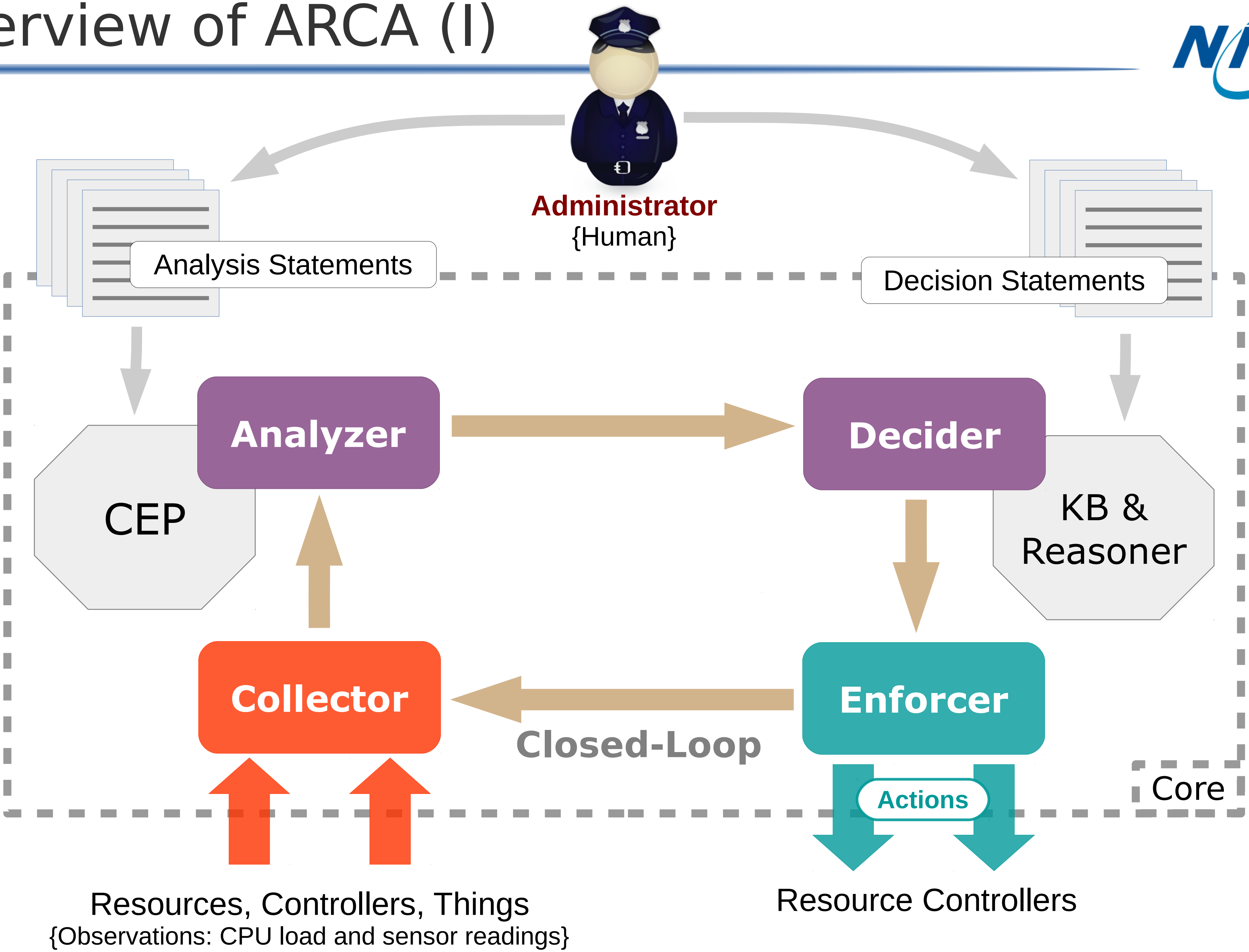
Proposed Approach (VII)



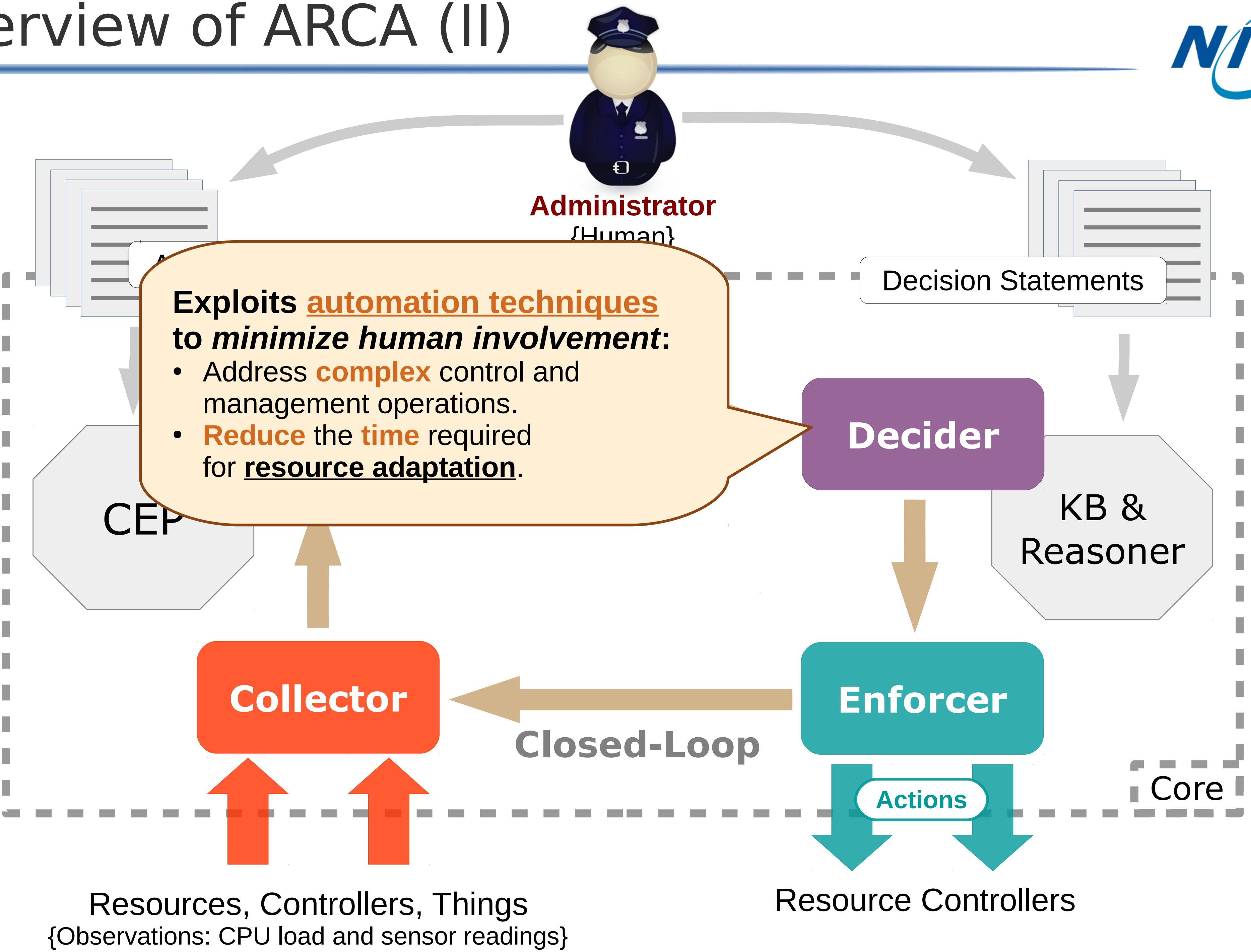
Challenges, Solutions, and Tools:

- **Too much observations** and volatility:
  - *Input filtering*:
    - + Ensure information is not lost and underlying system is not overstressed.
- **Reduce delay**:
  - High performance controller.
  - Anticipate situations (***learning***).
- **Reliability**:
  - Continuous check of policies provided by administrators (statements).

Overview of ARCA (I)



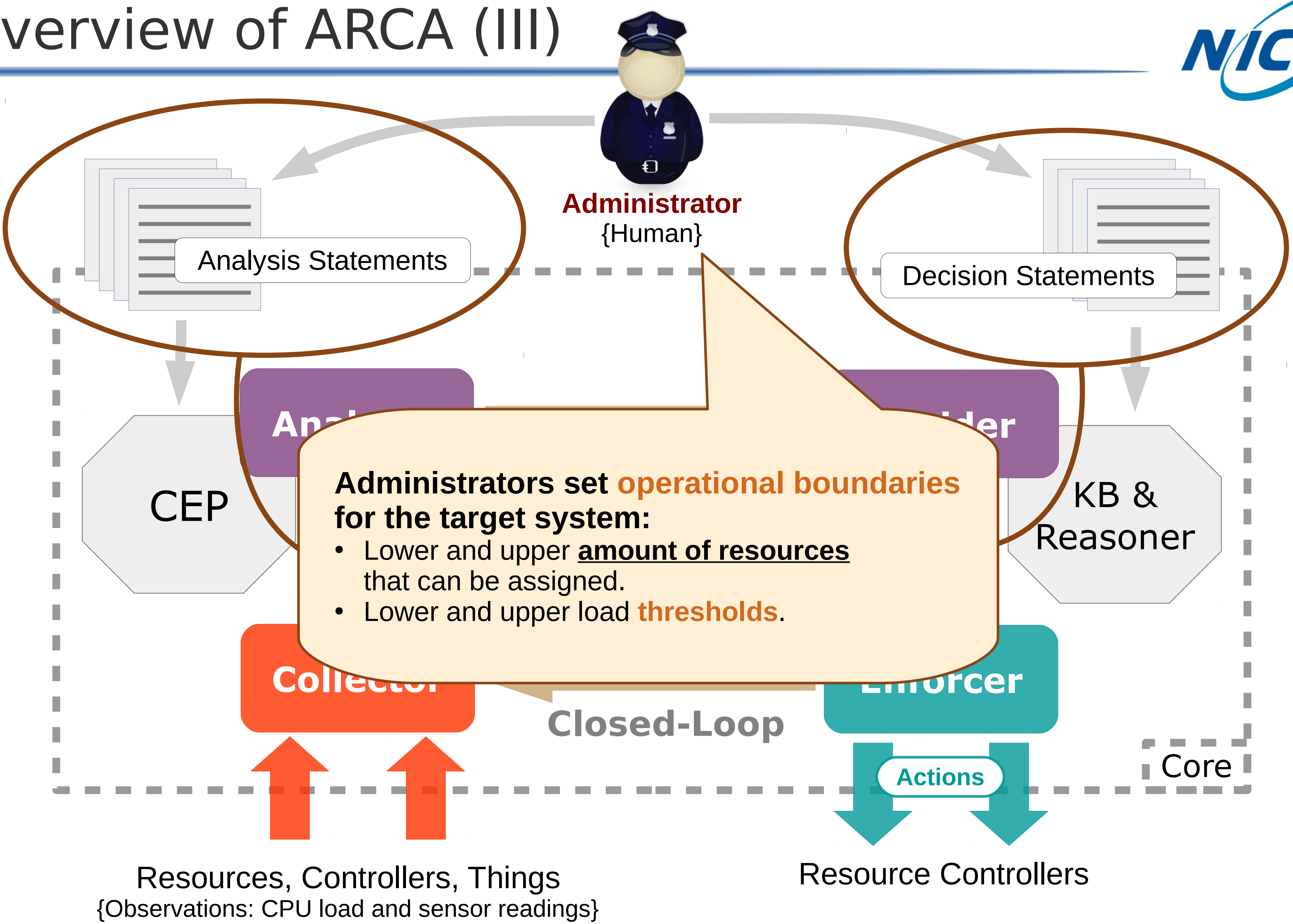
Overview of ARCA (II)



Exploits automation techniques to *minimize human involvement*:

- Address **complex** control and management operations.
- **Reduce** the **time** required for **resource adaptation**.

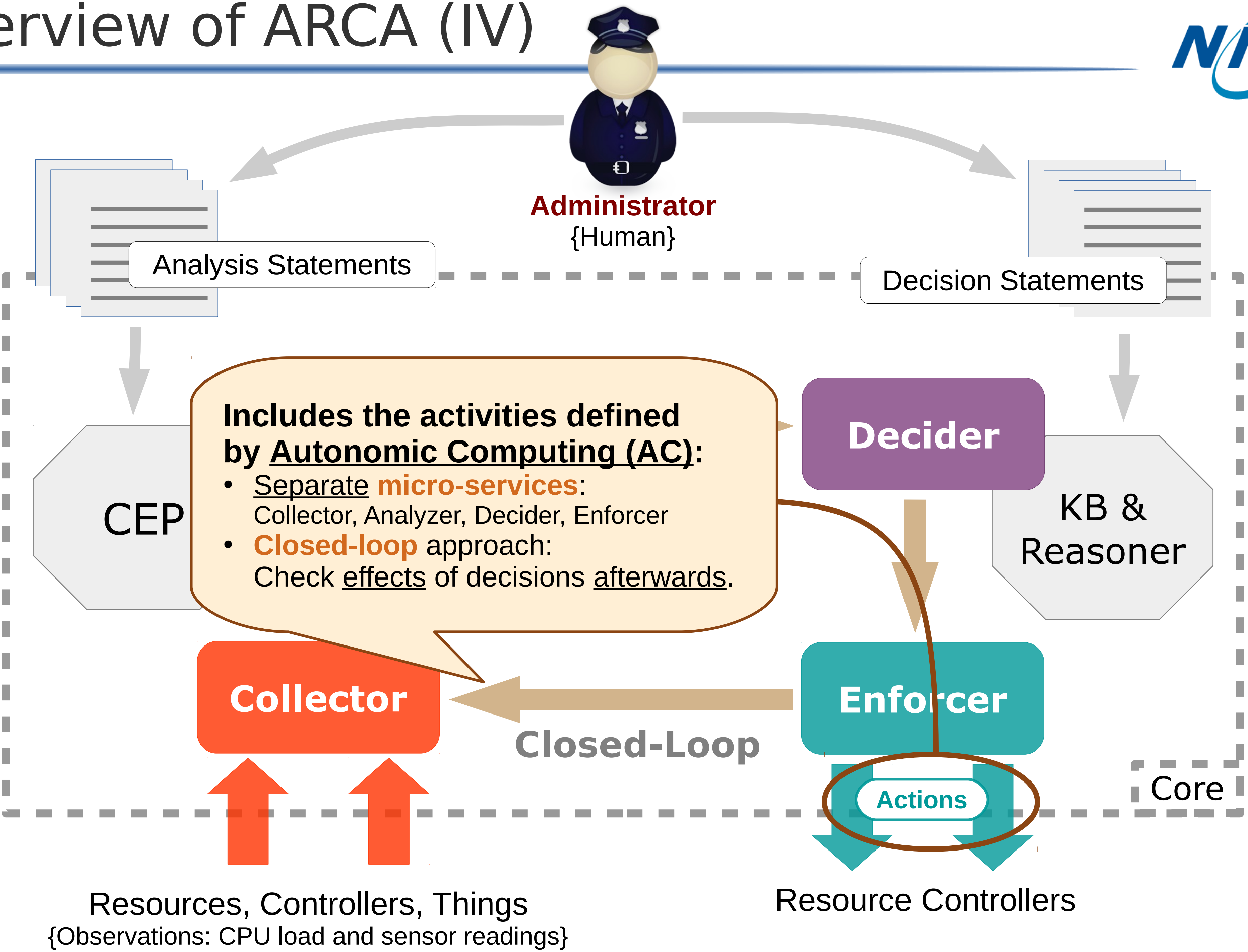
Overview of ARCA (III)



Administrators set operational boundaries for the target system:

- Lower and upper **amount of resources** that can be assigned.
- Lower and upper load **thresholds**.

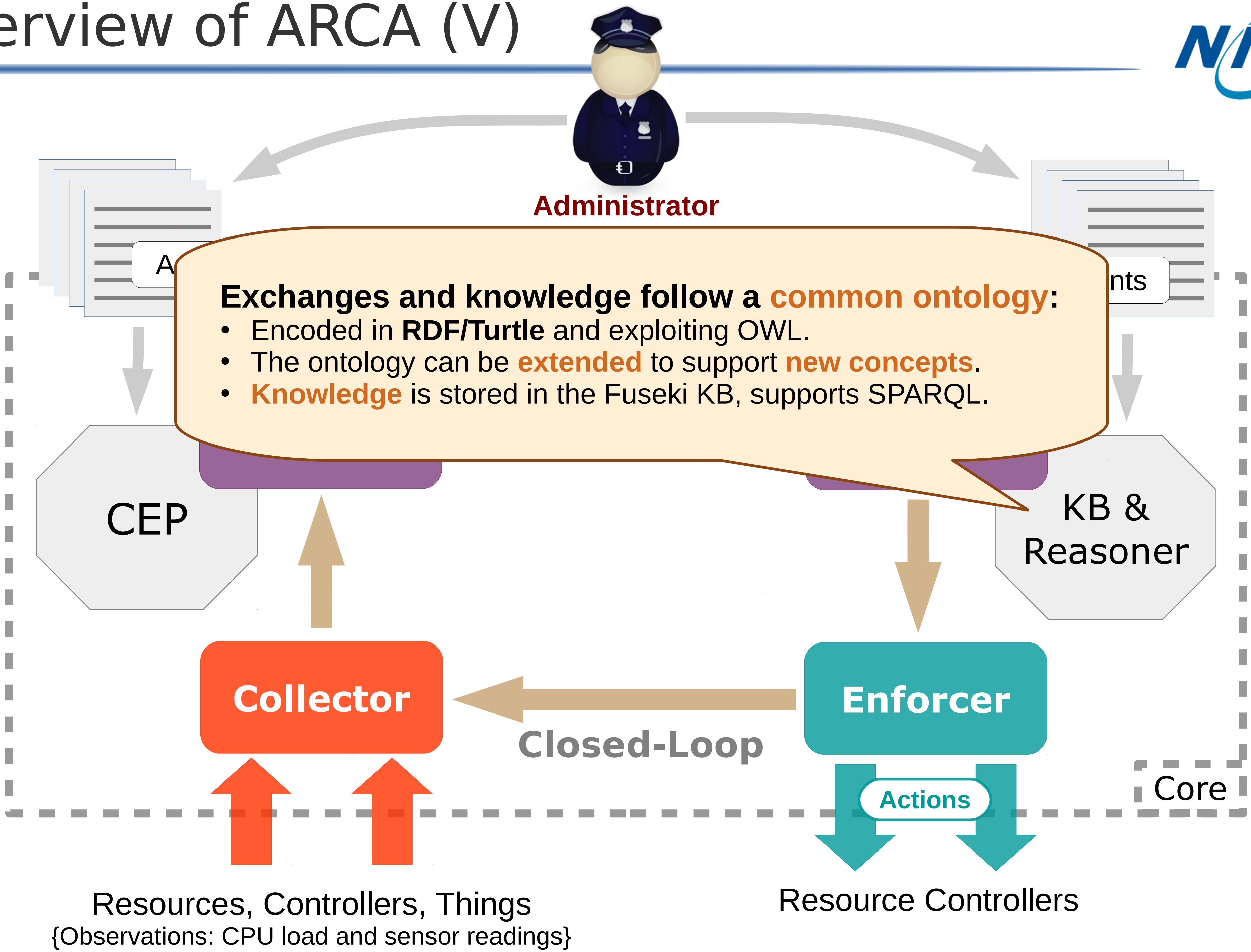
Overview of ARCA (IV)



Includes the activities defined by Autonomic Computing (AC):

- Separate **micro-services**: Collector, Analyzer, Decider, Enforcer
- **Closed-loop** approach: Check effects of decisions afterwards.

Overview of ARCA (V)



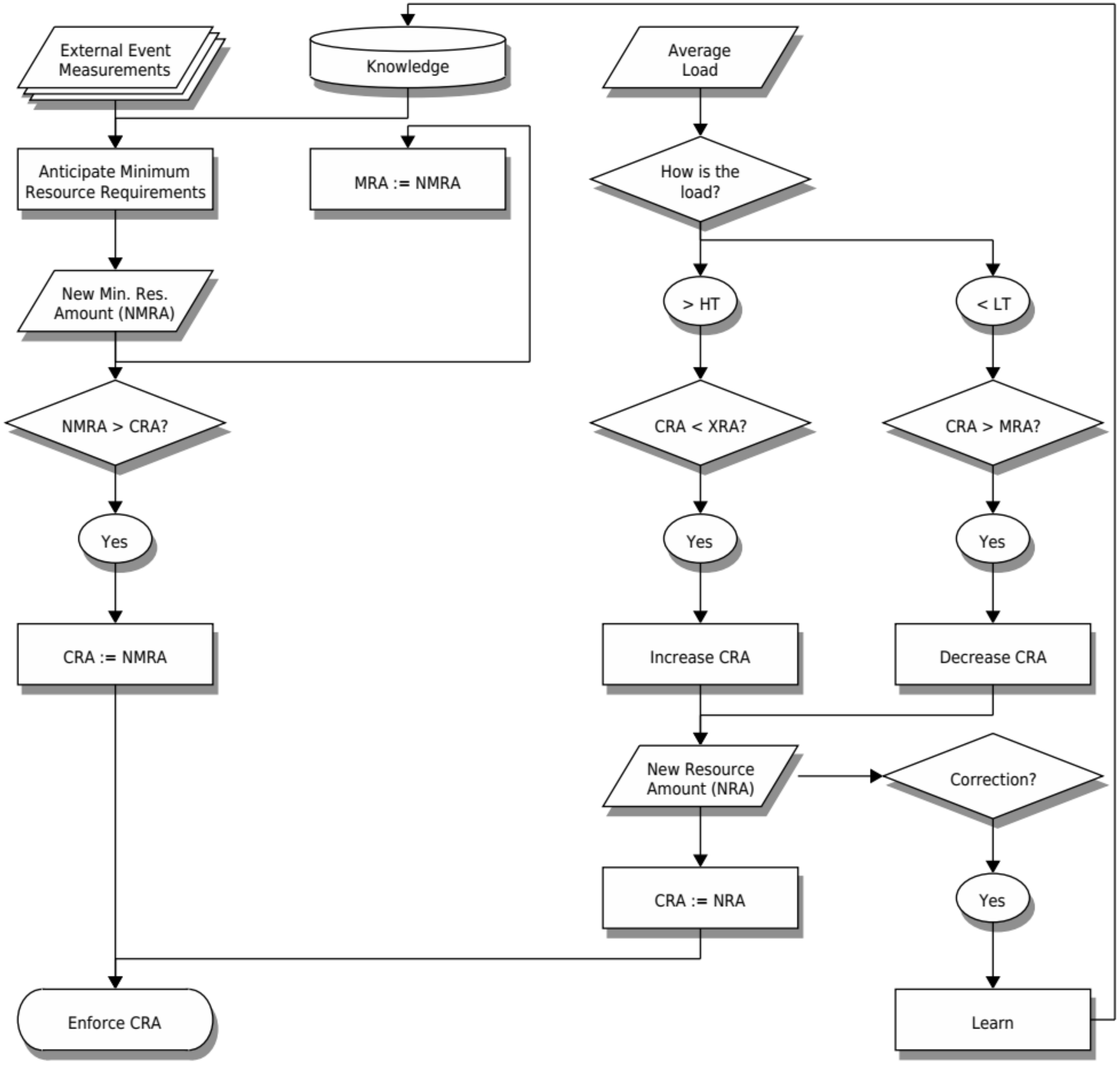
Exchanges and knowledge follow a common ontology:

- Encoded in **RDF/Turtle** and exploiting OWL.
- The ontology can be **extended** to support **new concepts**.
- **Knowledge** is stored in the Fuseki KB, supports SPARQL.

Resource Anticipation Strategy

- Functional and performance target:
  - **Anticipate** the amount of resources that a controlled system will require before it becomes effective.
- Involve **external event detectors**:
  - Physical: Things (IoT)
  - BigData
- Learn the event/reaction correlation:
  - Predict user behavior.
  - Correct mistaken predictions:
    - Improve and optimize learned model…
- Limit the memory used by the learning algorithm:
  - Keep only the most relevant vectors.
- **Fast adaptation** to big changes:
  - Discard old vectors when resizing.

Control Flow



- Two key controlled parameters:
  - Current Resource Amount (**CRA**).
  - Minimum Resource Amount (**MRA**).
- Two concurrent sub-routines:
  - **Anticipation**.
  - Threshold checking and **correction**.
- **Self-assessed** learning process:
  - Correcting learned data when finding mistakes…

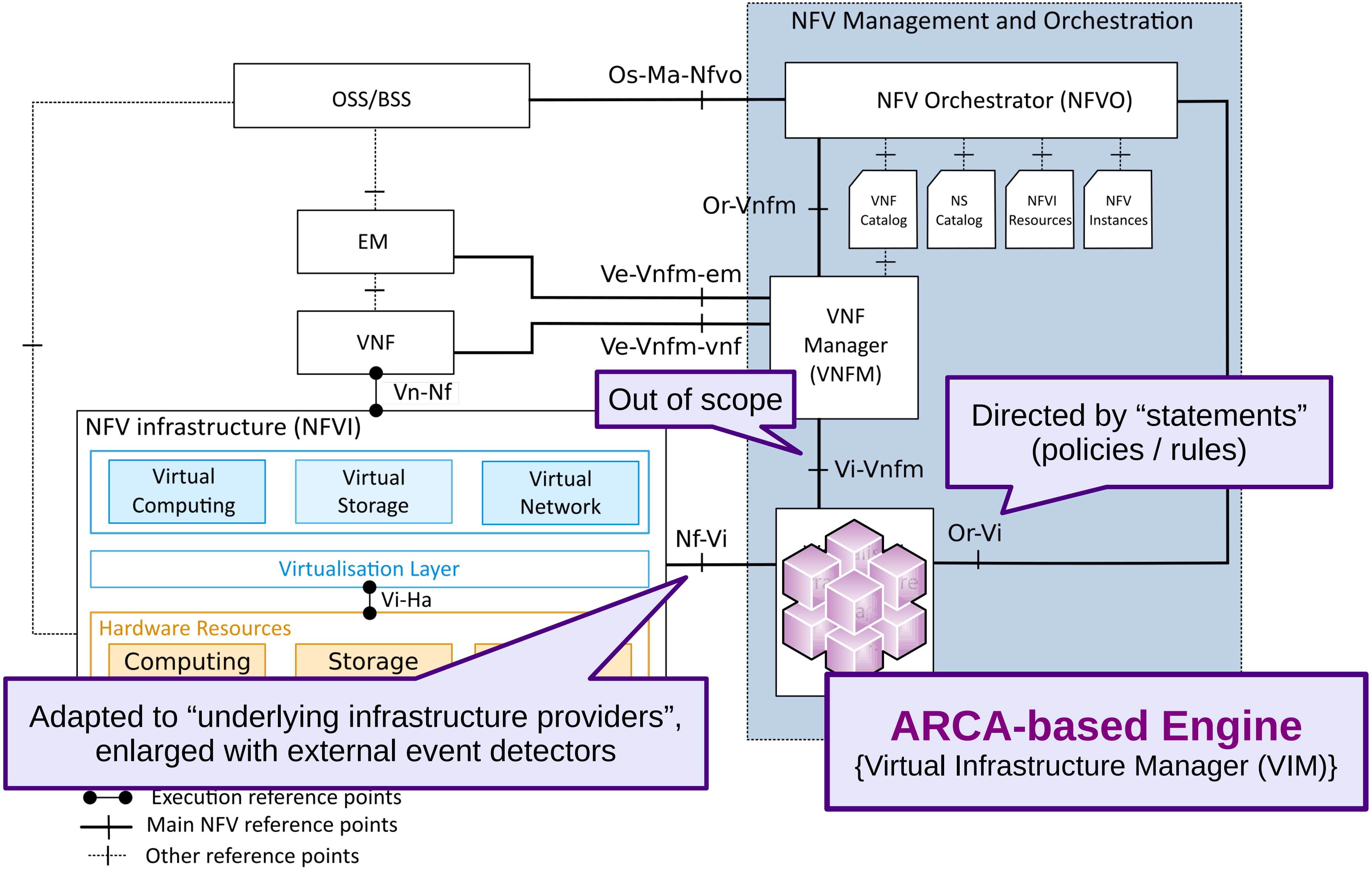
Algorithm (I)

```
procedure CONTROL(detectors, resources)
    $mra \leftarrow MIN\_RESOURCES$
    $cra \leftarrow mra$
    $anticipator \leftarrow$ LEARNERCREATE($MIN\_RESOURCES, MAX\_RESOURCES$)
    $ant\_severity, ant\_time, ant\_peak, ant\_peak\_rel\_time \leftarrow 0, 0, 0, 0$
    $severity, pseverity, load, drate \leftarrow 0, 0, 1, 0$
    $attl\_model \leftarrow$ LEARNERCREATE($MIN\_ATTL, MAX\_ATTL$)
    while TRUE do
        $severity \leftarrow$ COLLATESENSORREADINGS(COLLECT($detectors, SEVERITY$))
        $load \leftarrow$ CALCULATEAVGLOAD(COLLECT($resources, LOAD$))
        $drate \leftarrow$ CALCULATEAVGDROPRATE(COLLECT($resources, DROP\_RATE$))
        if $ant\_severity \neq 0$ then
            $demand \leftarrow cra * \frac{load + drate}{SERVER\_WORK\_QCAP}$
            if $demand > ant\_peak$ then
                $ant\_peak \leftarrow demand$
                $ant\_peak\_rel\_time \leftarrow (NOW - ant\_time) * 1.25$
            end if
            if $NOW - ant\_time >$ LEARNERGET($attl\_model, ant\_severity$) and $load < LT$ then
                LEARNERSET($anticipator, ant\_severity, \frac{ant\_peak}{HT}$)
                LEARNERSET($attl\_model, ant\_severity, ant\_peak\_rel\_time$)
                $ant\_severity \leftarrow 0$
                $ant\_time \leftarrow 0$
                $mra \leftarrow MIN\_RESOURCES$
            end if
        end if
```

Algorithm (II)

```
        if $severity \neq 0$ and $severity \neq pseverity$ then
            $ant\_severity \leftarrow severity$
            $ant\_time \leftarrow NOW$
            $ant\_peak \leftarrow 0$
            $mra \leftarrow \textsc{LearnerGet}(anticipator, severity)$
        end if
        $pseverity \leftarrow severity$
        $nra \leftarrow cra$
        if $load > HT$ or $cra < mra$ then
            $nra \leftarrow \textsc{Min}(\textsc{Max}(cra + INC, \frac{load * cra}{HT}, mra), MAX\_RESOURCES)$
        end if
        if $load < LT$ then
            $nra \leftarrow \textsc{Max}(nra - DEC, mra, MIN\_RESOURCES)$
        end if
        if $nra \neq cra$ and $\textsc{NoSideEffect}(nra)$ then
            $cra \leftarrow nra$
            $\textsc{Enforce}(cra)$
        end if
    end while
end procedure
```

Alignment With ETSI-NFV-MANO (I)



Out of scope

Directed by “statements” (policies / rules)

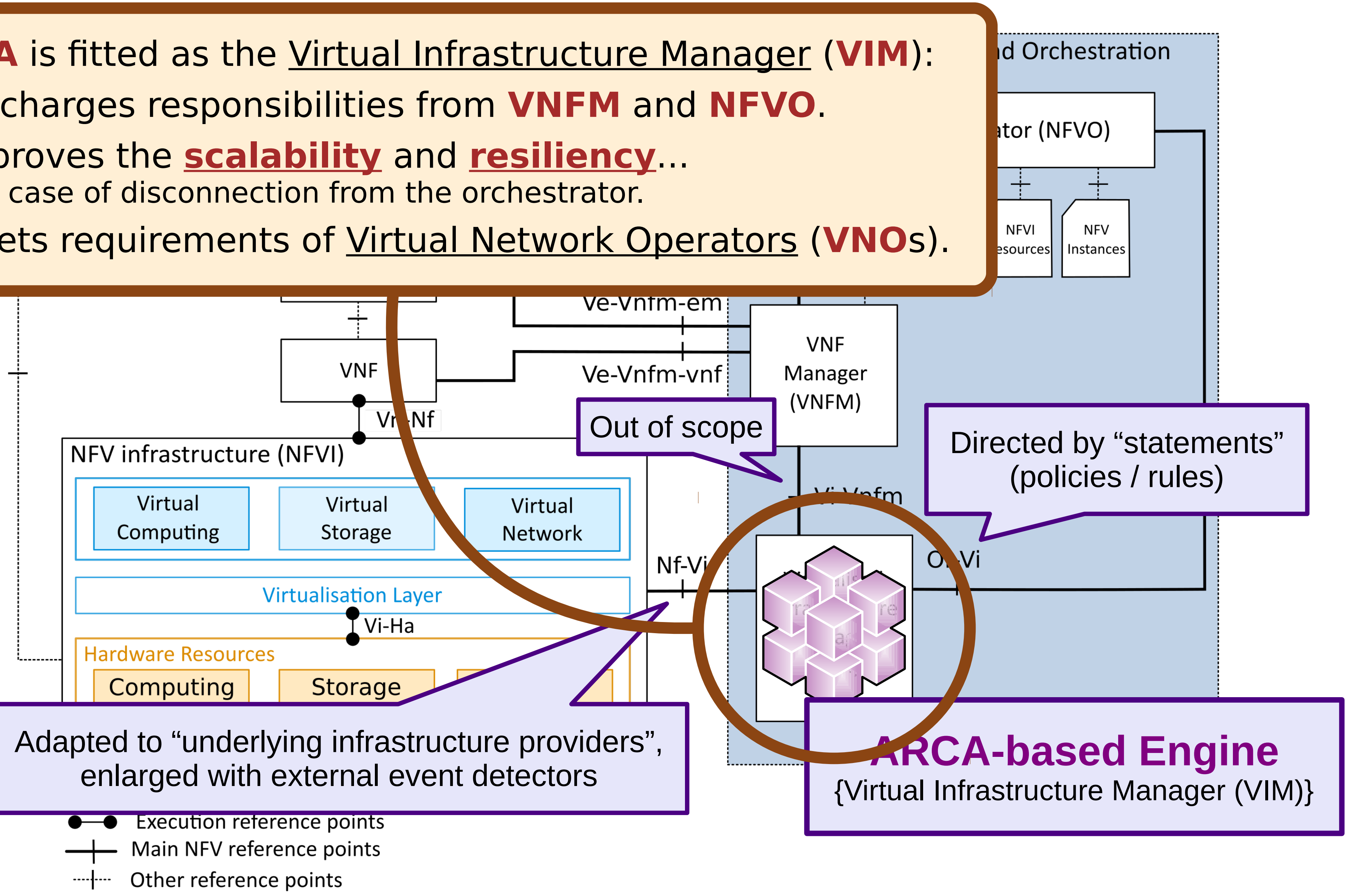
Adapted to “underlying infrastructure providers”, enlarged with external event detectors

ARCA-based Engine
{Virtual Infrastructure Manager (VIM)}

Alignment With ETSI-NFV-MANO (II)

ARCA is fitted as the Virtual Infrastructure Manager (**VIM**):

- Discharges responsibilities from **VNFM** and **NFVO**.
- Improves the **scalability** and **resiliency**...
  ...in case of disconnection from the orchestrator.
- Meets requirements of Virtual Network Operators (**VNO**s).



Out of scope

Directed by “statements” (policies / rules)

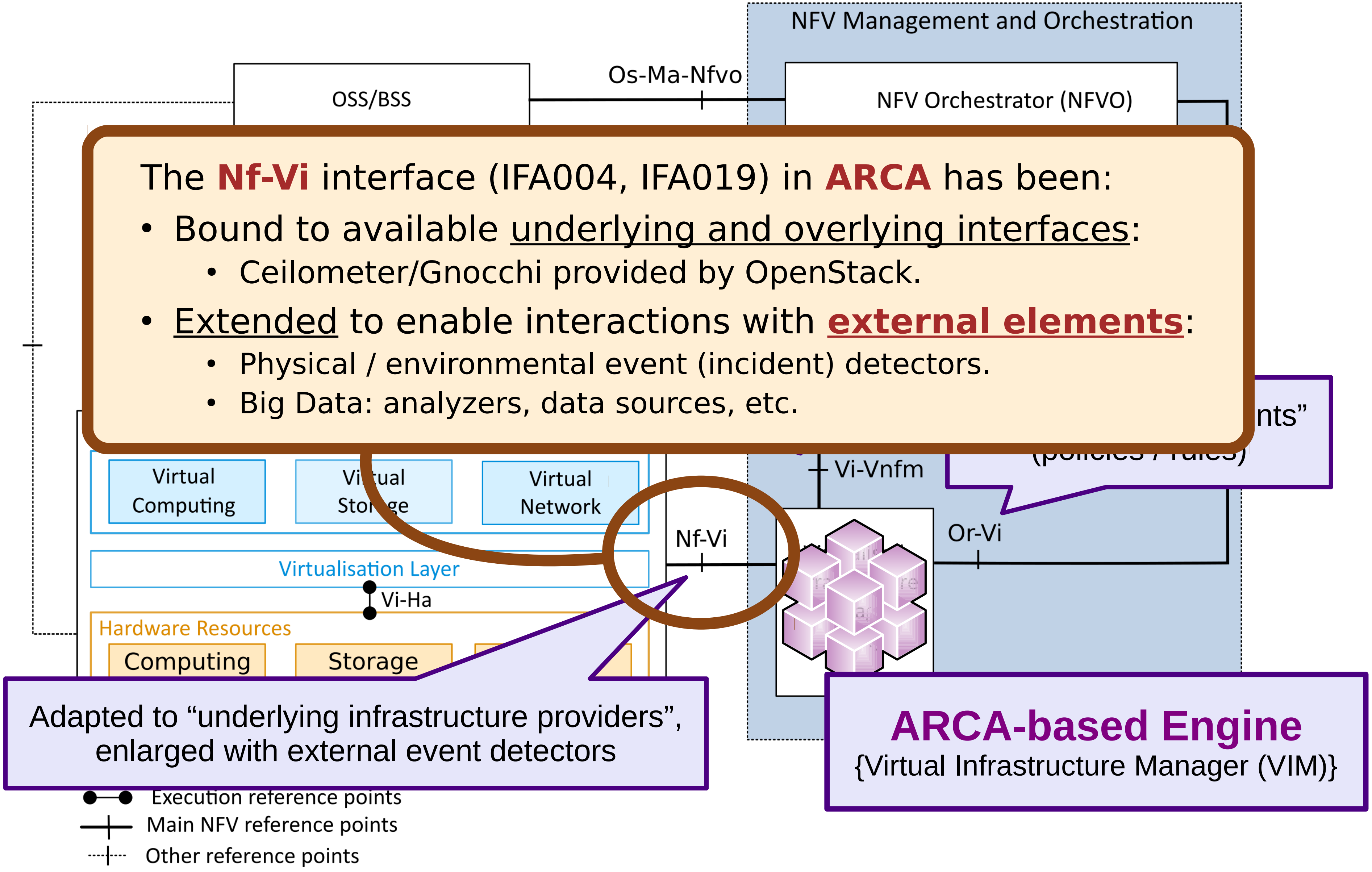
Adapted to “underlying infrastructure providers”, enlarged with external event detectors

ARCA-based Engine
{Virtual Infrastructure Manager (VIM)}

Alignment With ETSI-NFV-MANO (III)

The **Nf-Vi** interface (IFA004, IFA019) in **ARCA** has been:

- Bound to available underlying and overlying interfaces:
  - Ceilometer/Gnocchi provided by OpenStack.
- Extended to enable interactions with **external elements**:
  - Physical / environmental event (incident) detectors.
  - Big Data: analyzers, data sources, etc.



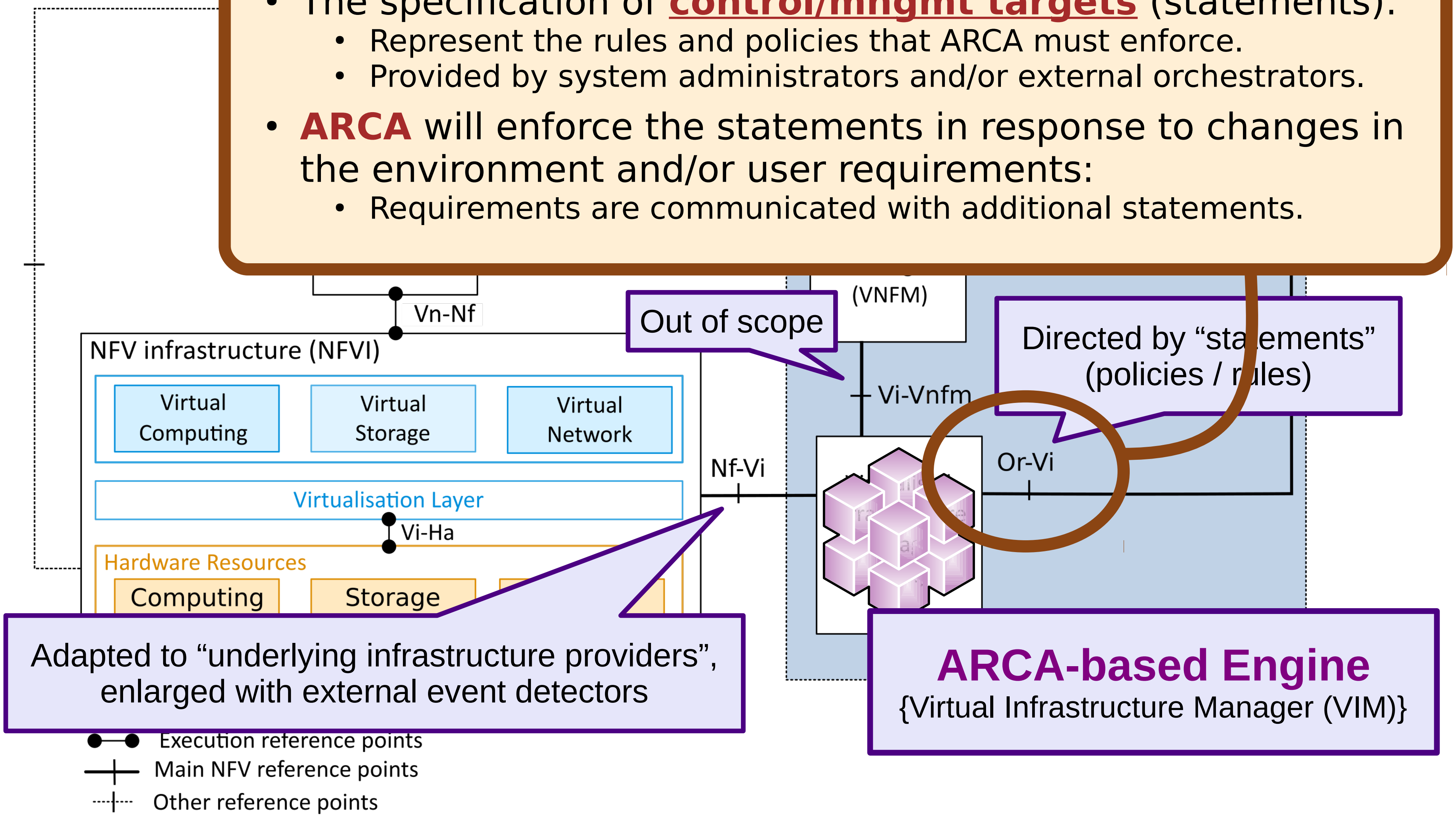
Adapted to “underlying infrastructure providers”, enlarged with external event detectors

ARCA-based Engine
{Virtual Infrastructure Manager (VIM)}

Alignment With ETSI-NFV-MANO (IV)

The **Or-Vi** interface (IFA005) is provided by:

- The specification of **control/mngmt targets** (statements):
  - Represent the rules and policies that ARCA must enforce.
  - Provided by system administrators and/or external orchestrators.
- **ARCA** will enforce the statements in response to changes in the environment and/or user requirements:
  - Requirements are communicated with additional statements.



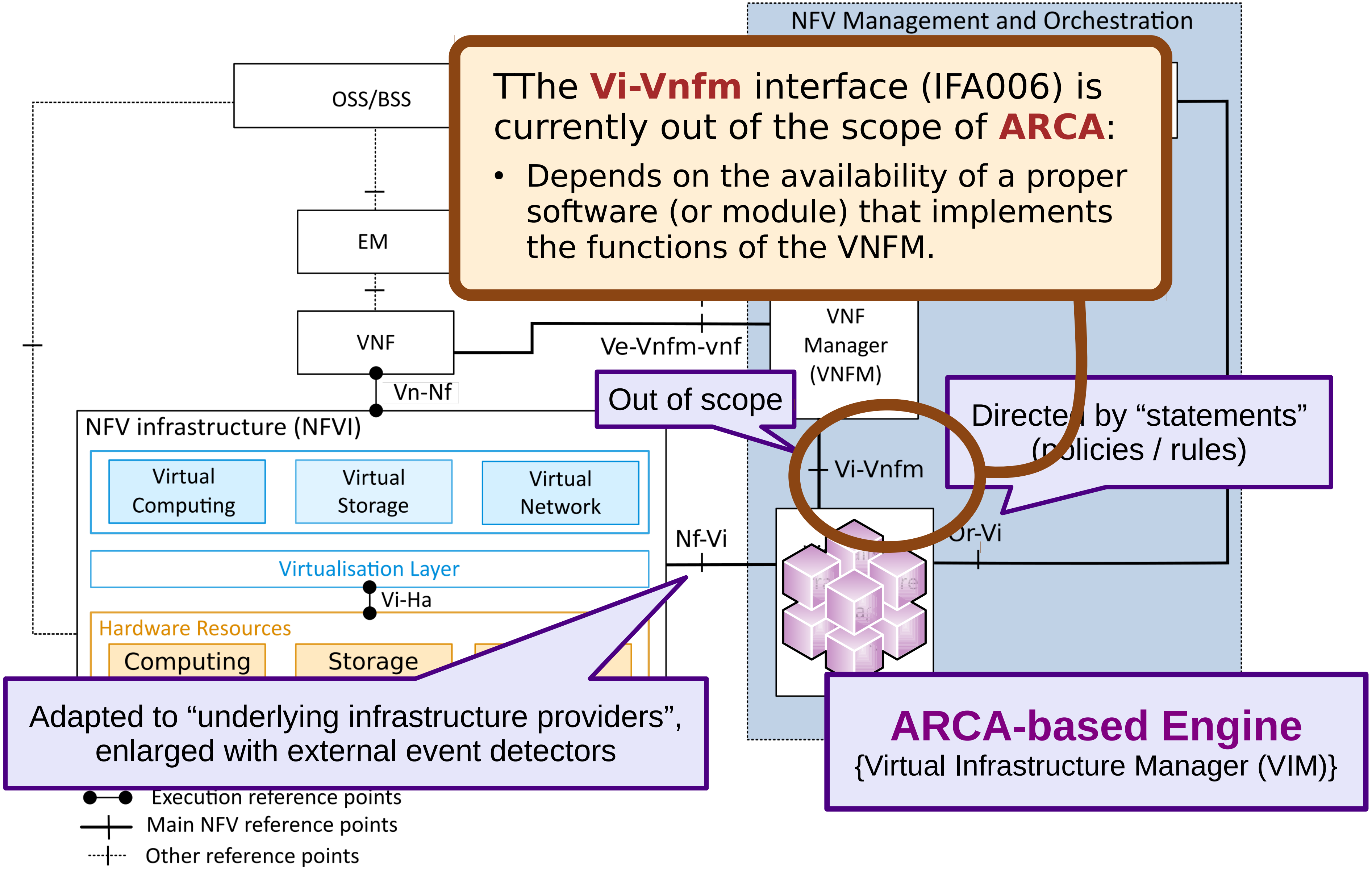
Out of scope

Directed by “statements” (policies / rules)

Adapted to “underlying infrastructure providers”, enlarged with external event detectors

ARCA-based Engine
{Virtual Infrastructure Manager (VIM)}

Alignment With ETSI-NFV-MANO (V)



TThe **Vi-Vnfm** interface (IFA006) is currently out of the scope of **ARCA**:

- Depends on the availability of a proper software (or module) that implements the functions of the VNFM.

Out of scope

Directed by "statements" (policies / rules)

Adapted to "underlying infrastructure providers", enlarged with external event detectors

ARCA-based Engine
{Virtual Infrastructure Manager (VIM)}

Conclusions & Future Work

- Designed **ARCA**:
  - To provide functions of the Virtual Infrastructure Manager (**VIM**) of **NFV-MANO**.
  - Extended **VIM interfaces** to meet requirements of the **real world**:
    - Sport events, TV shows, emergency scenarios...
  - Achieved good perfomance within an OpenStack-based deployment:
    - Detailed **overlying** and **underlying** infrastructures.
    - Reproduction of production-like environments to ensure transferable research results.
- SDN/NFV and OpenStack stakeholders will benefit from ARCA features:
  - **Efficient** use of **resources**:
    - Further reduce CAPEX and OPEX:
  - Benefit to both infrastructure providers and consumers.
- Next steps:
  - Keep **reducing** ARCA **response time**.
  - Increase complexity of the validation scenario:
    - Mix clients and servants in the same domains.
  - Align **ARCA**-based VNC to **additional equirements** from **NFV**/**SDN**.

Thanks for Your Attention

Q & A

– EOF –