

# Integrating an Online Configuration Checker with Existing Management Systems: Application to CIM/WBEM Environments

Ludi Akue, Emmanuel Lavinal,  
Thierry Desprats, Michelle Sibilla

University of Toulouse – IRIT Lab. – France

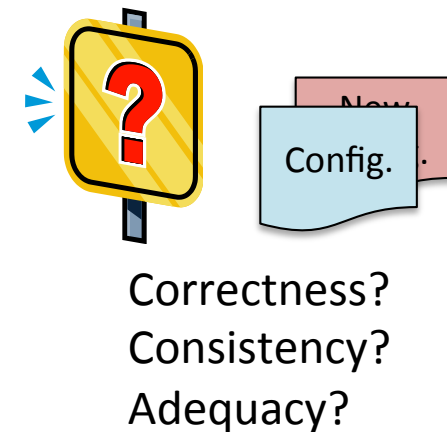
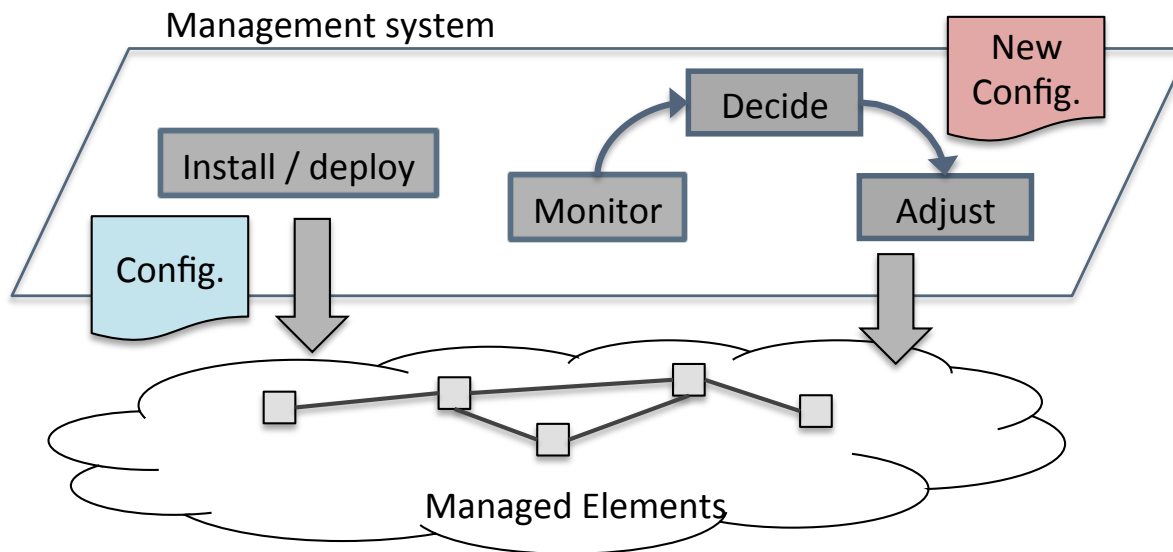
# Outline

---

- Configuration verification requirement in NSM
- Framework for runtime configuration verification
  - MeCSV metamodel
  - Configuration constraints checker
- Application of the framework in CIM/WBEM environments
- Experiments on the CIM System Virtualization Profile

# Configuration verification requirement

- Configuration and reconfiguration are fundamental activities in network and service management (the system's behavior is controlled by its configuration)



**→ Need to verify configurations before their application to guarantee the system's safe and correct operation**

# Configuration verification

---

- Currently, most related work address
  - Use-case specific verification [SANchk08]
  - Structural integrity verification [Hinrich04, Warren06, PoDIM07]
- **Structural verification** deals with type conformance, range of authorized values, syntactic correctness...
- Examples of structural verification:
  - The “host name” configuration parameter must not be empty
  - An IPv4 address must have 4 bytes
- Structural configuration verification is independent from a runtime

$\text{Conf}_x$  is structurally valid  $\Rightarrow$   $\text{Conf}_x$  is valid  $\forall$   $\text{RT}_i$

# Configuration verification (cont.)

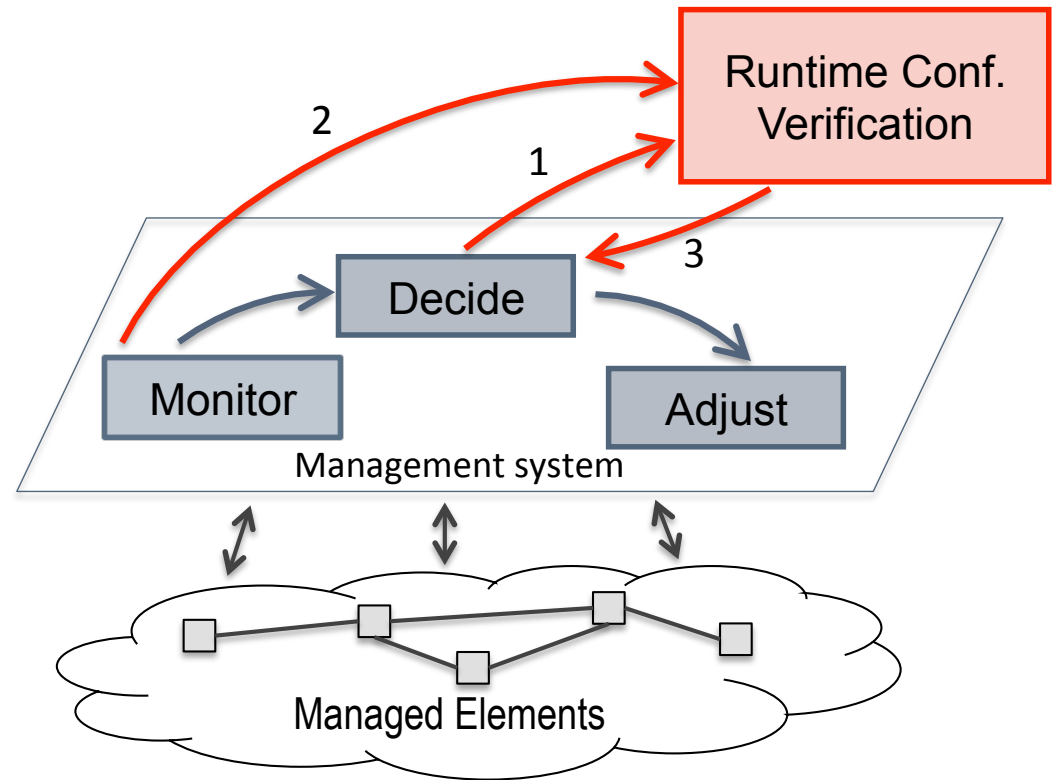
---

- In today's dynamic environments, configurations are highly dependent on the operational conditions
- We argue that an **online configuration verification** capability is required to assess the configuration's compliance regarding the current runtime context
- Examples of online/operational verification:
  - The server hosting the HTTP service must be in a running state
  - When applying the backup configuration, the current number of clients must be less than 50% of the maximum number of clients
- **Online configuration verification depends on a runtime**

$\text{Conf}_x$  is operationally valid in  $\text{RT}_y \not\Rightarrow \text{Conf}_x$  is valid  $\forall \text{RT}_i$

# General approach for an online configuration verification

- Enhancing existing management systems with a verification functionality
- Independent from a specific configuration language or platform
- Enable an online configuration verification (coupled with a runtime monitoring)



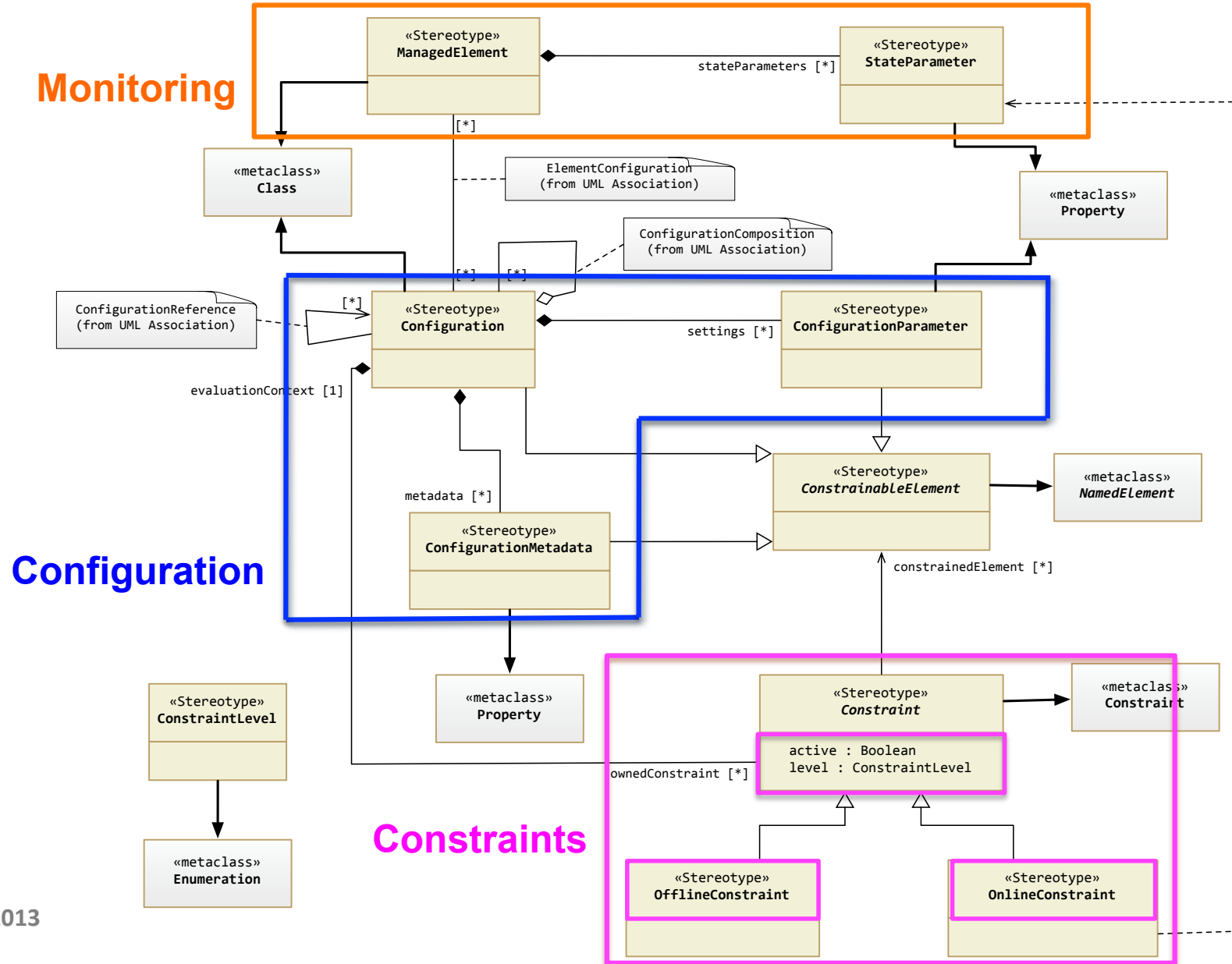
→ Approach based on a dedicated high-level language: MeCSV metamodel

# MeCSV: Metamodel for Configuration Specification & Validation

---

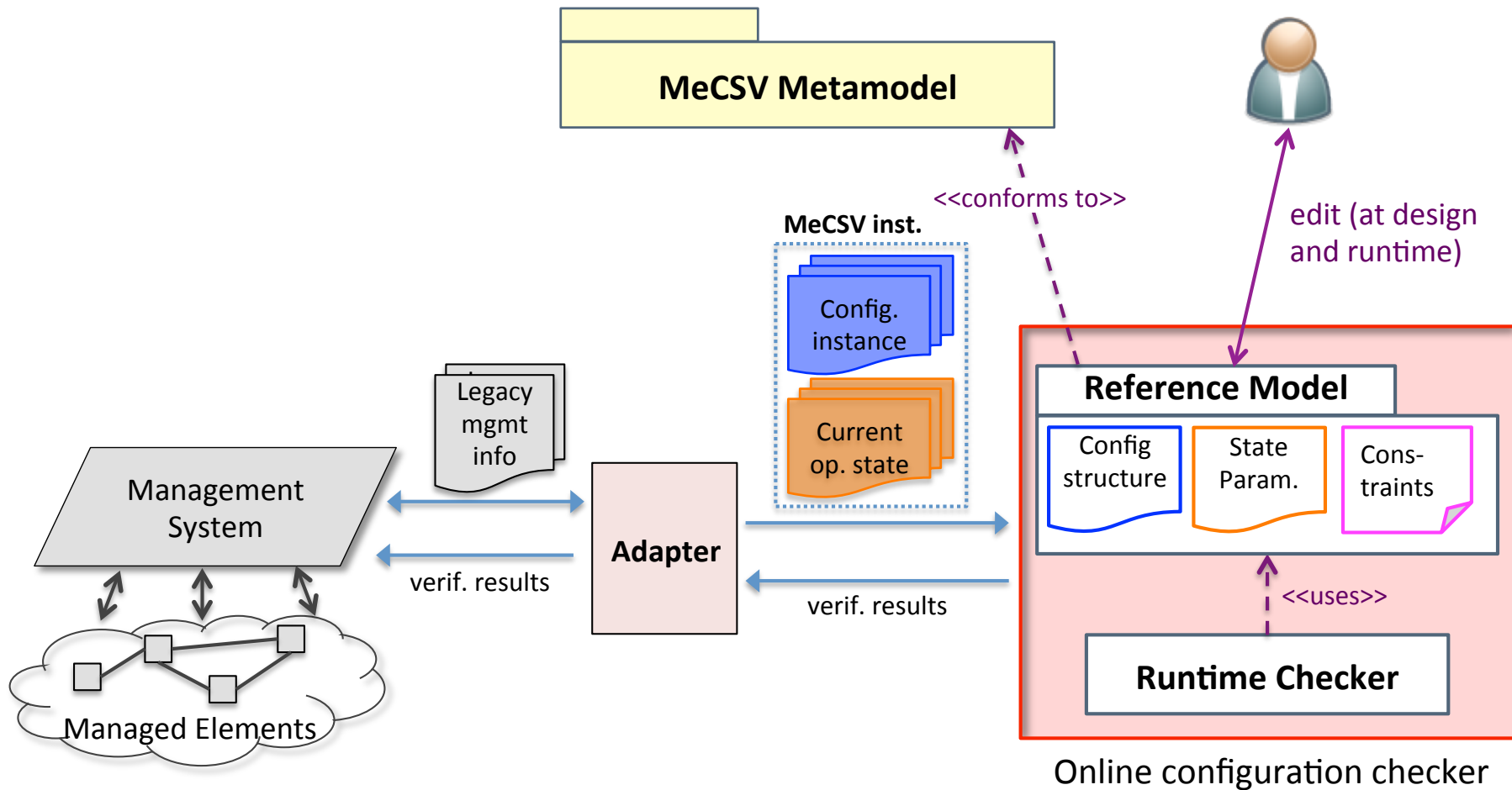
- MeCSV metamodel [L. Akue et al., VALID 2012]
  - High-level language dedicated to configuration verification
  - Simple but sufficient to express offline and online constraints
- Built around 3 capabilities :
  - **Configuration description:** the ability to define a configuration model for a given system
  - **Constraint expression:** the ability to express constraints that configurations must respect for both structural and operational validity
  - **Monitored data representation:** the ability to represent relevant operational state data that influence the operational validity of a configuration

# MeCSV UML Profile



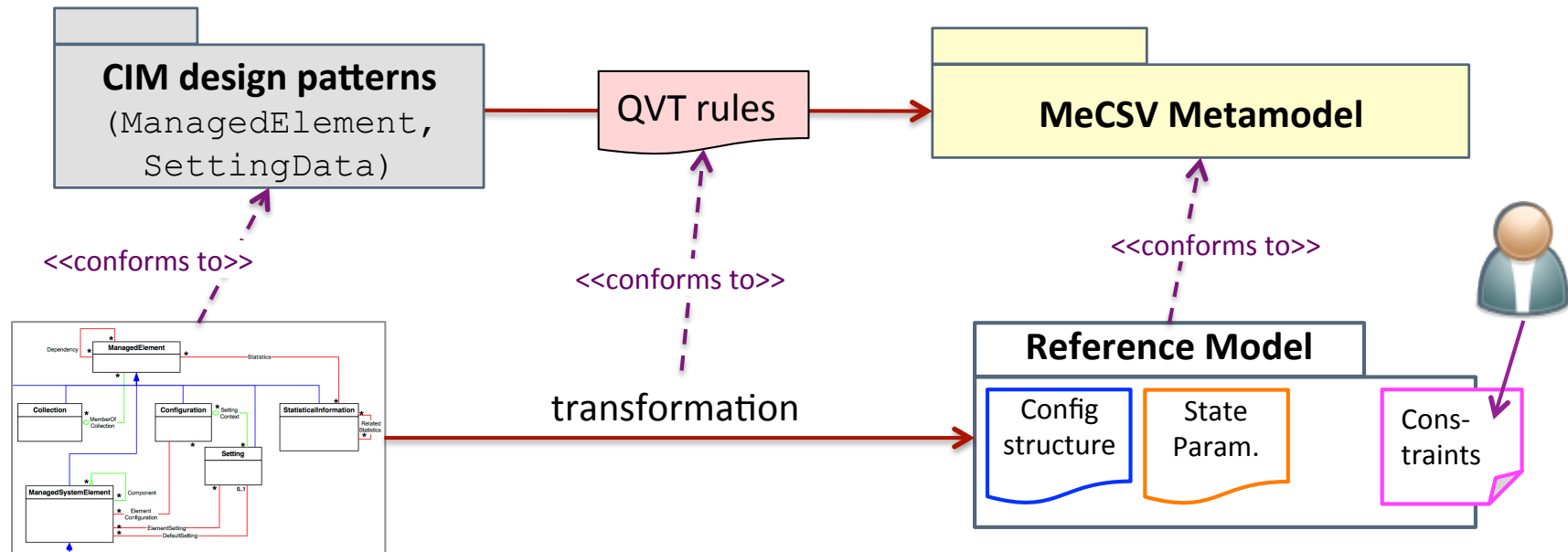


# Framework for online configuration verification



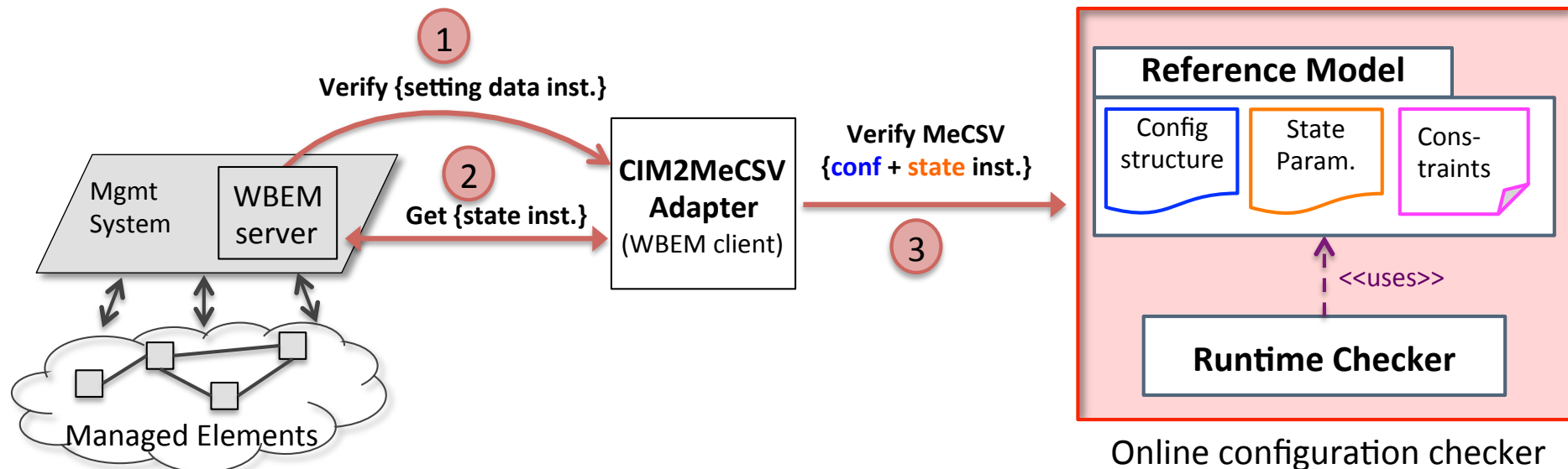
# Application to CIM/WBEM standards

- **Design time:** generating part of the MeCSV reference model from CIM schemas
  1. Usage of the DMTF's CIM\_SettingData configuration pattern
  2. Definition of transformation rules (based on the QVT transformation language) between CIM settings data and the MeCSV metamodel



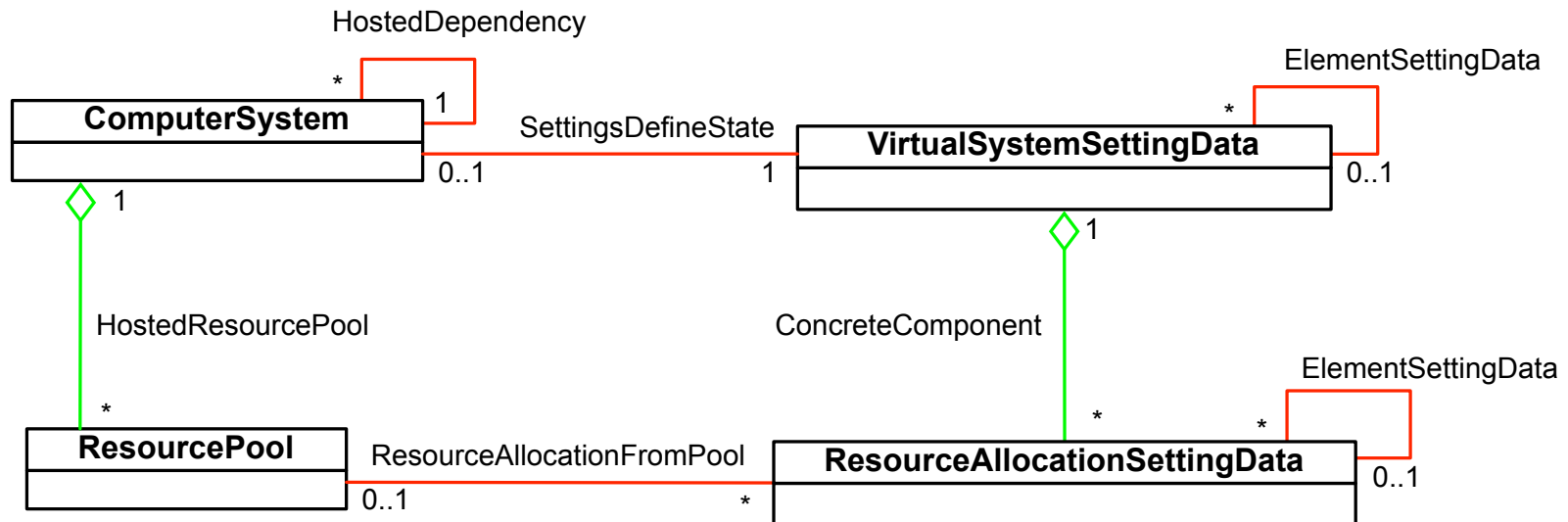
# Application to CIM/WBEM standards (cont.)

- **Runtime:** building a CIM2MeCSV adapter capable of
  - Mapping at runtime CIM instances to MeCSV instances
  - Gathering from the managed system the state parameters' current values (used for online constraints evaluation)



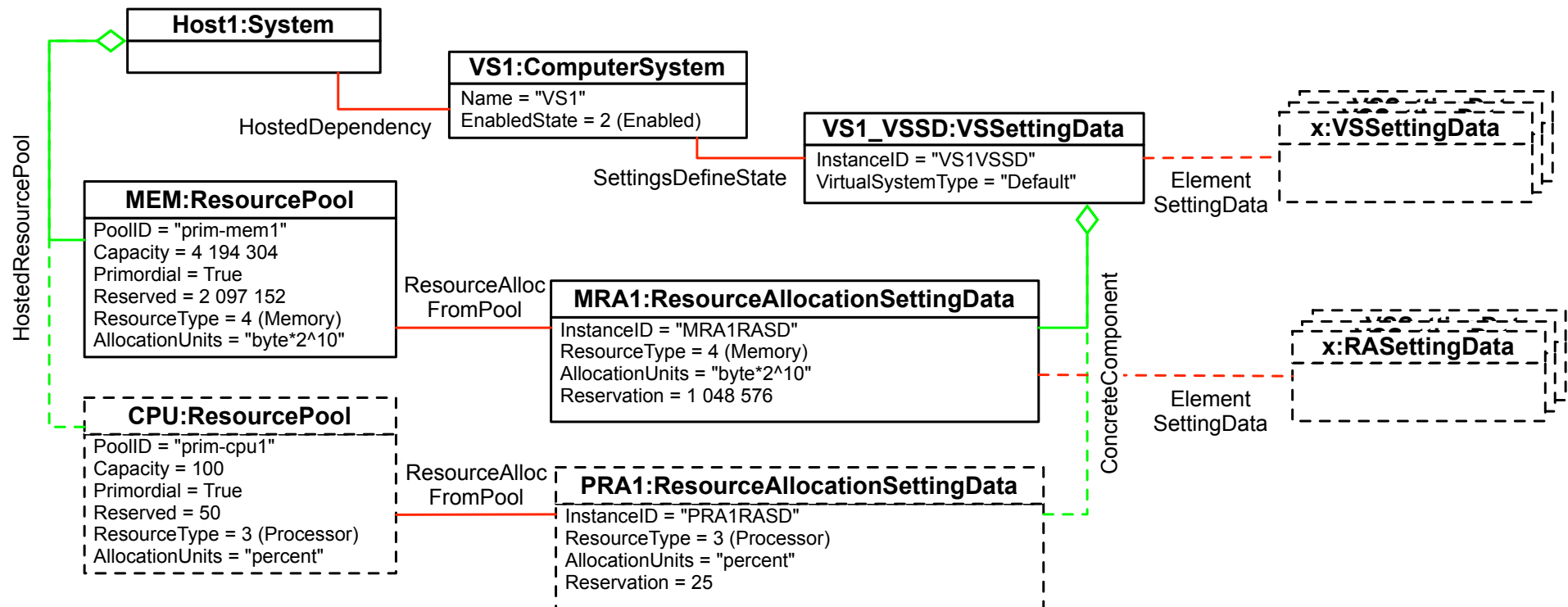
# CIM System Virtualization Profile Case Study

- System virtualization profile (simplified class diagram)



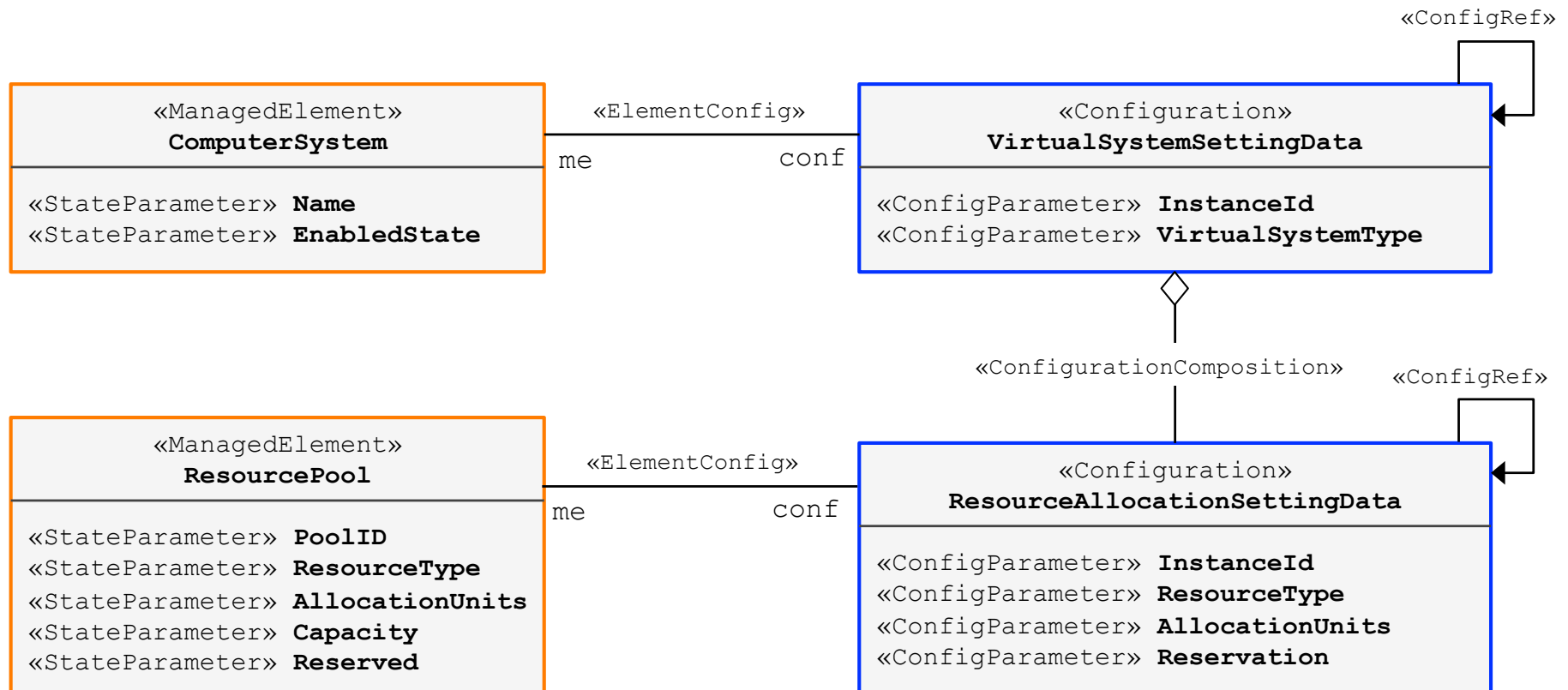
# CIM System Virtualization Profile Case Study

- Example



# Experimentation: design time

- MeCSV reference model generation (from CIM VS profile)



# Experimentation: design time

---

- Adding constraints to the MeCSV reference model

## Offline constraints examples

- Every `VirtualSystemSD` instance must have a unique id
- Every `VirtualSystemSD` instance must include a sub configuration `ResourceAllocationSD` of type “Processor”, “Memory” and “Disk”
- Every `ResourceAllocationSD` instance must have a `Reservation > 0`

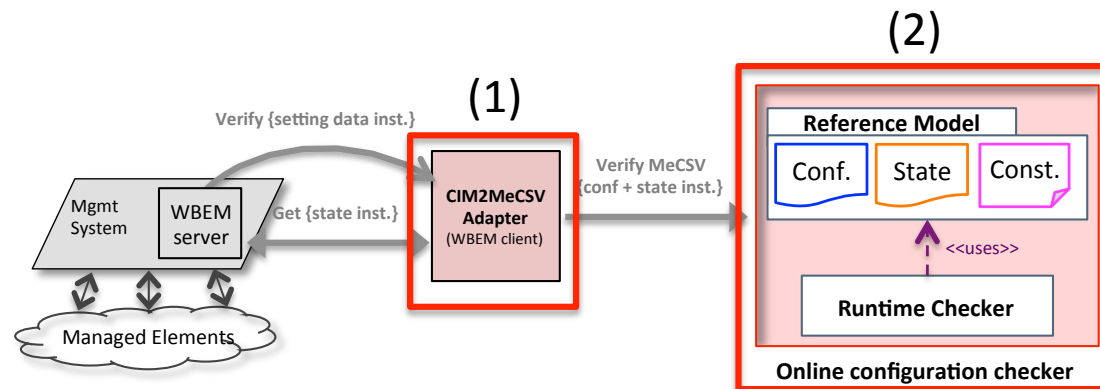
## Online constraints examples

- The `VirtualSystemSD`'s “state” configuration must be associated to a computer system instance in the `enabled` state
- The sum of the all the `ResourceAllocationSD` `Reservations` must not exceed the `Capacity` of the corresponding `ResourcePool`

# Experimentation: runtime

- Goals

1. Test the CIM2MeCSV adapter and measure its overhead (transformation and integration time)
2. Test the online configuration checker and measure the verification time



- Evaluation according to different configuration sizes and number of constraints

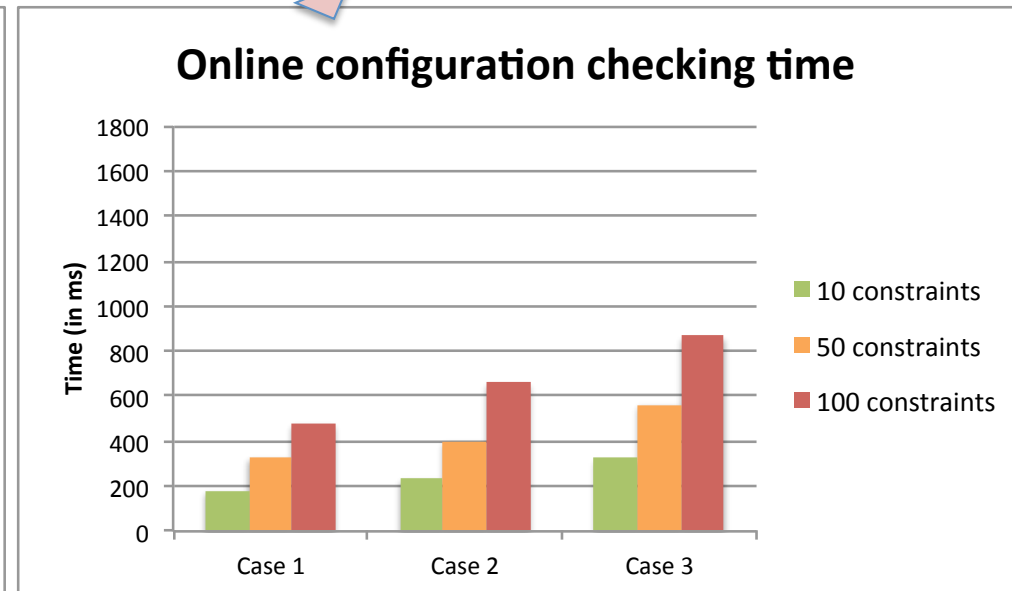
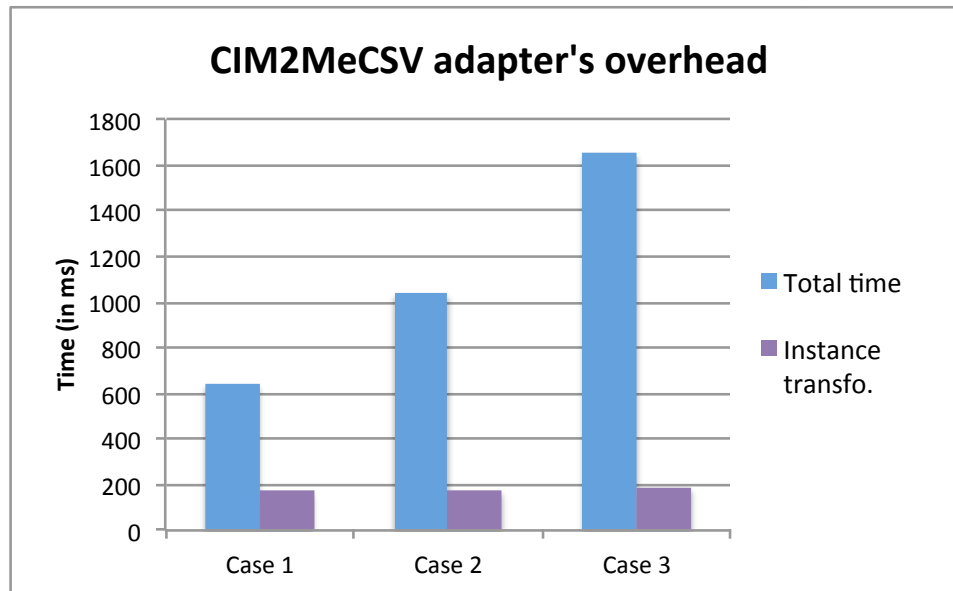
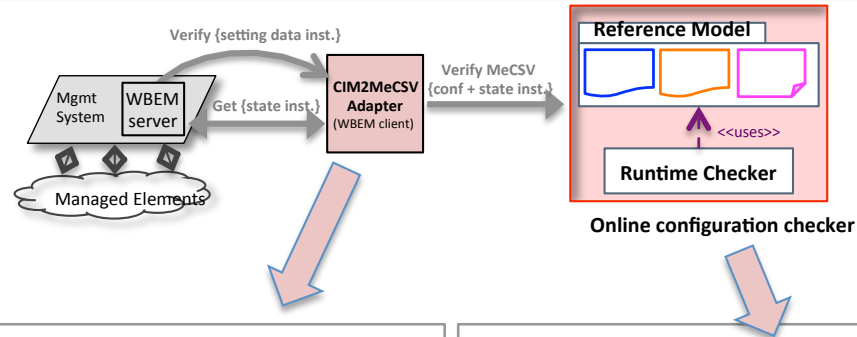


# Experimentation: runtime – evaluation setup

---

- CIM instances provided by OpenPegasus WBEM server (default repository provider)
- 3 scenarios
  - **Case 1:** 1 host, 1 VM, ~ 180 config param. & 140 state param.
  - **Case 2:** 1 host, 2 VM, ~ 360 config param. & 140 state param.
  - **Case 3:** 1 host, 3 VM, ~ 800 config param. & 140 state param.
- 3 sets of OCL constraints
  - 10 constraints (~ 70% offline & 30% online)
  - 50 constraints (~ 70% offline & 30% online)
  - 100 constraints (~ 70% offline & 30% online)

# Experimentation: runtime – evaluation results



- **Case 1:** 1 host, 1 VM, ~ 180 config param. & 140 state param.
- **Case 2:** 1 host, 2 VM, ~ 360 config param. & 140 state param.
- **Case 3:** 1 host, 3 VM, ~ 800 config param. & 140 state param.

# Conclusion and future work

---

- Framework for online configuration verification based on a dedicated (minimal) metamodel named MeCSV
  - Provide online constraints verification for configuration applicability
  - Enrich existing mgmt systems with a configuration verification capability
- Application to CIM/WBEM environments
  - Mapping rules from CIM modeling patterns to MeCSV constructs
  - Runtime adapter to retrieve state parameters from a WBEM server
- Future work
  - Improve the mapping between standard management information models and MeCSV constructs (in particular for the verification results)
  - Ease the specification of constraints (derivation from requirements?)

**Thank you!**

**SVM**  **2013**  
**ZURICH**