Continuous Deployment of Multi-clouds Applications

Nicolas Ferry, Hui Song, Franck Chauvel, Arnor Solberg

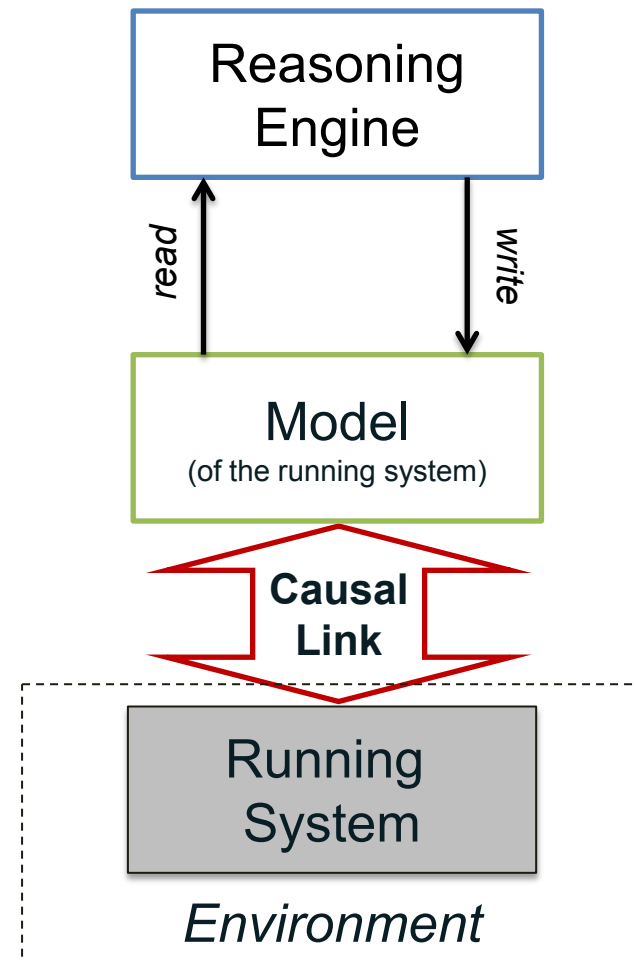**SINTEF**

QUDOS 2015

SINTEF

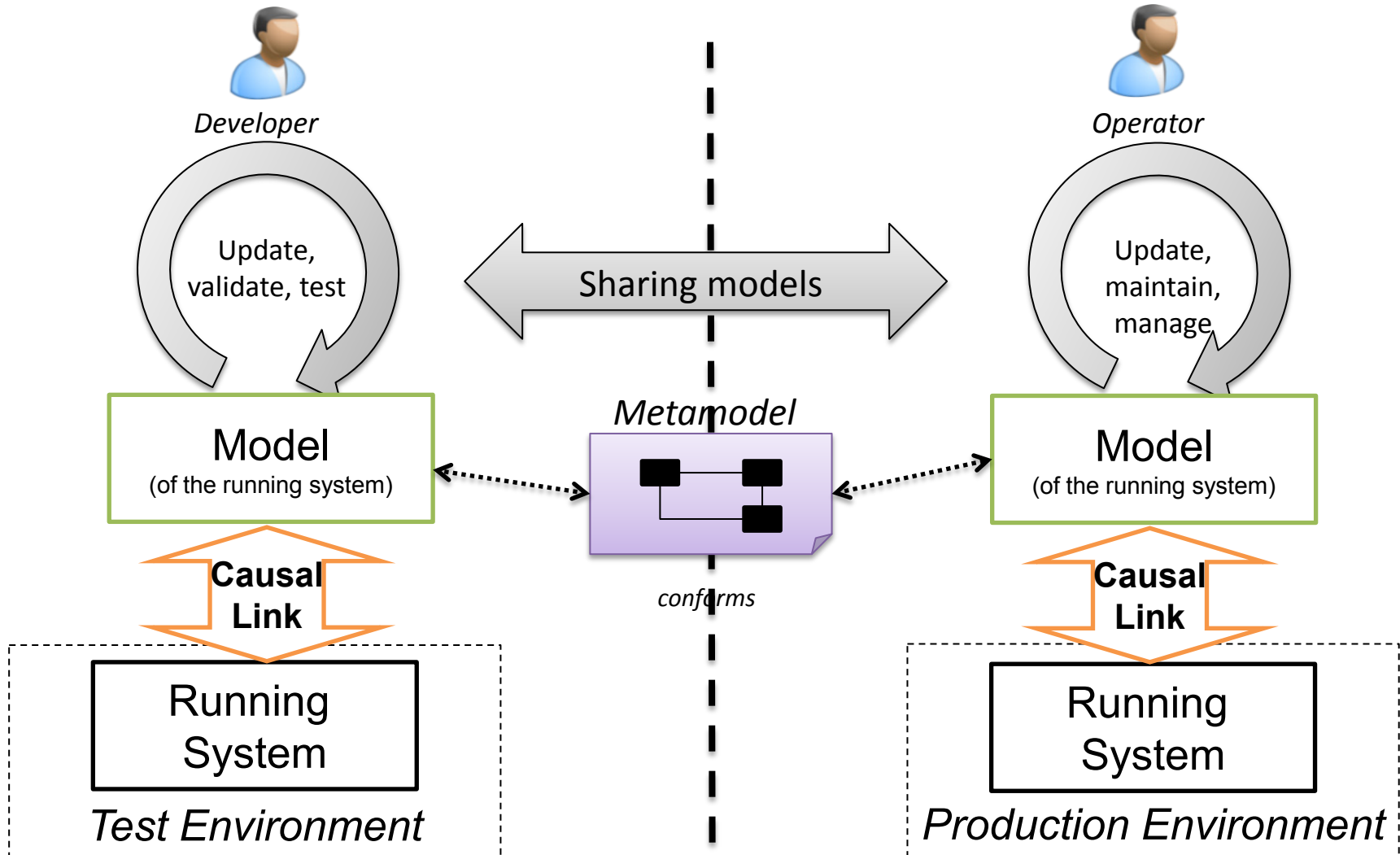Technology for a better society

The Cloud Modelling Framework

- Two main components:
  - A **modelling environment** with a tool-supported domain-specific modelling language (**CloudML**) to model the provisioning and deployment of multi-cloud systems
  - A **models@run-time environment** for enacting the provisioning, deployment and adaptation of these systems
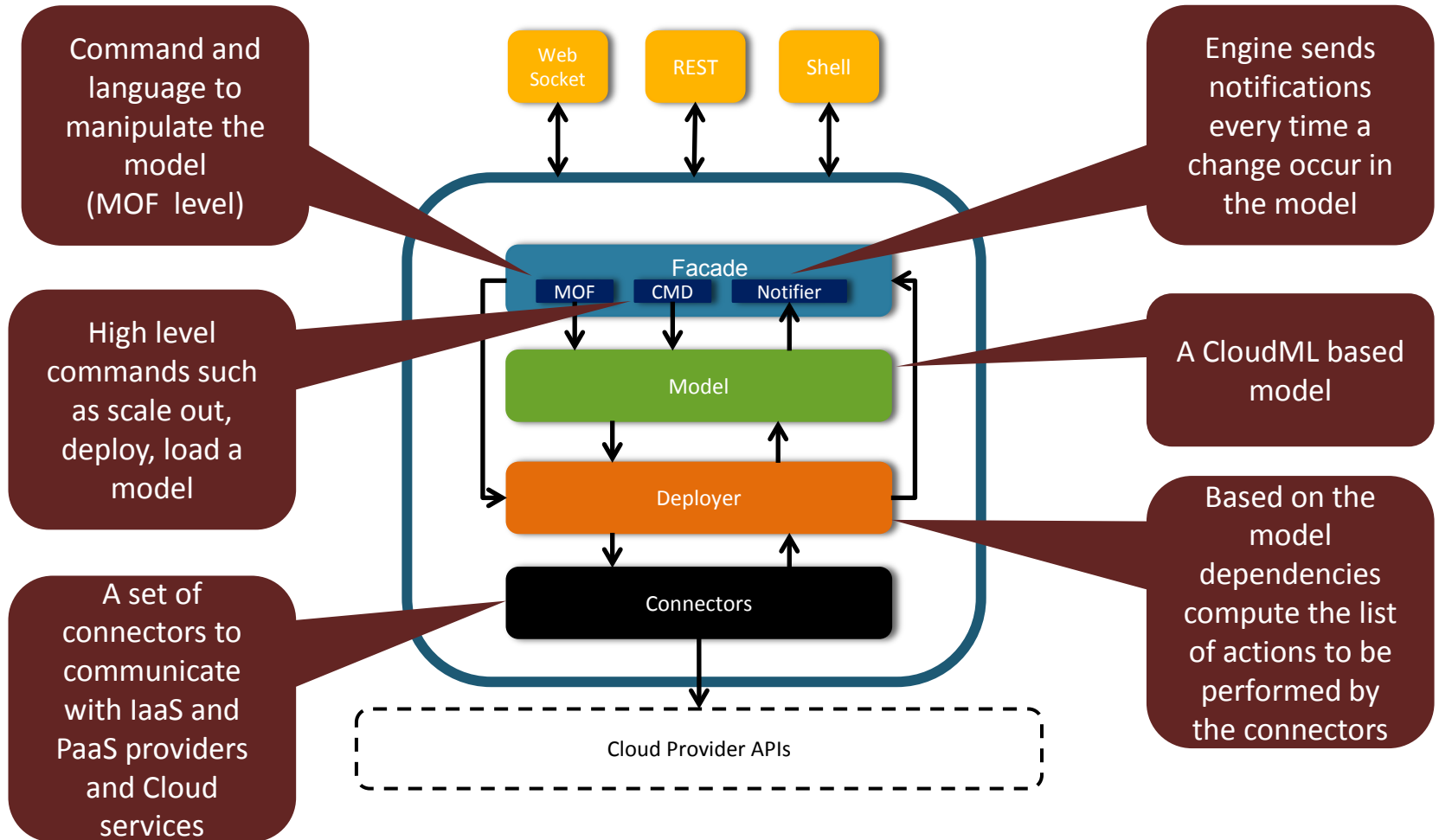
The models@runtime pattern

- Architectural pattern for DAS

- Using a Causal Link

- Benefits
  - Continuous design & validation
  - Separation of Concerns
  - Reuse of MDE tools

Models@runtime to support DevOps

Developer

Operator

Update, validate, test

Sharing models

Update, maintain, manage

Model
(of the running system)

Metamodel

Model
(of the running system)

Causal Link

conforms

Causal Link

Running System

Running System

Test Environment

Production Environment

The Models@Runtime environment

Command and language to manipulate the model (MOF level)

Engine sends notifications every time a change occur in the model

High level commands such as scale out, deploy, load a model

A CloudML based model

Web Socket

REST

Shell

Facade

MOF   CMD   Notifier

Model

Deployer

Connectors

Cloud Provider APIs

A set of connectors to communicate with IaaS and PaaS providers and Cloud services

Based on the model dependencies compute the list of actions to be performed by the connectors

SINTEF

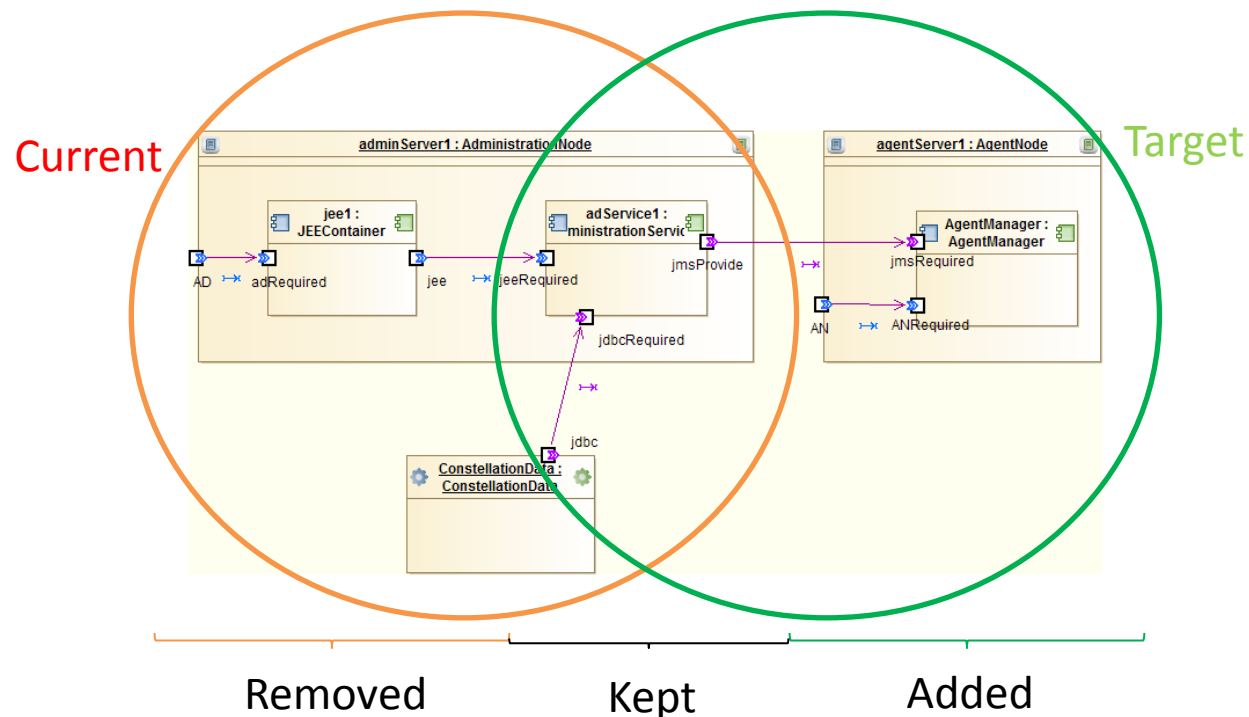Technology for a better society

Models@Runtime core architecture
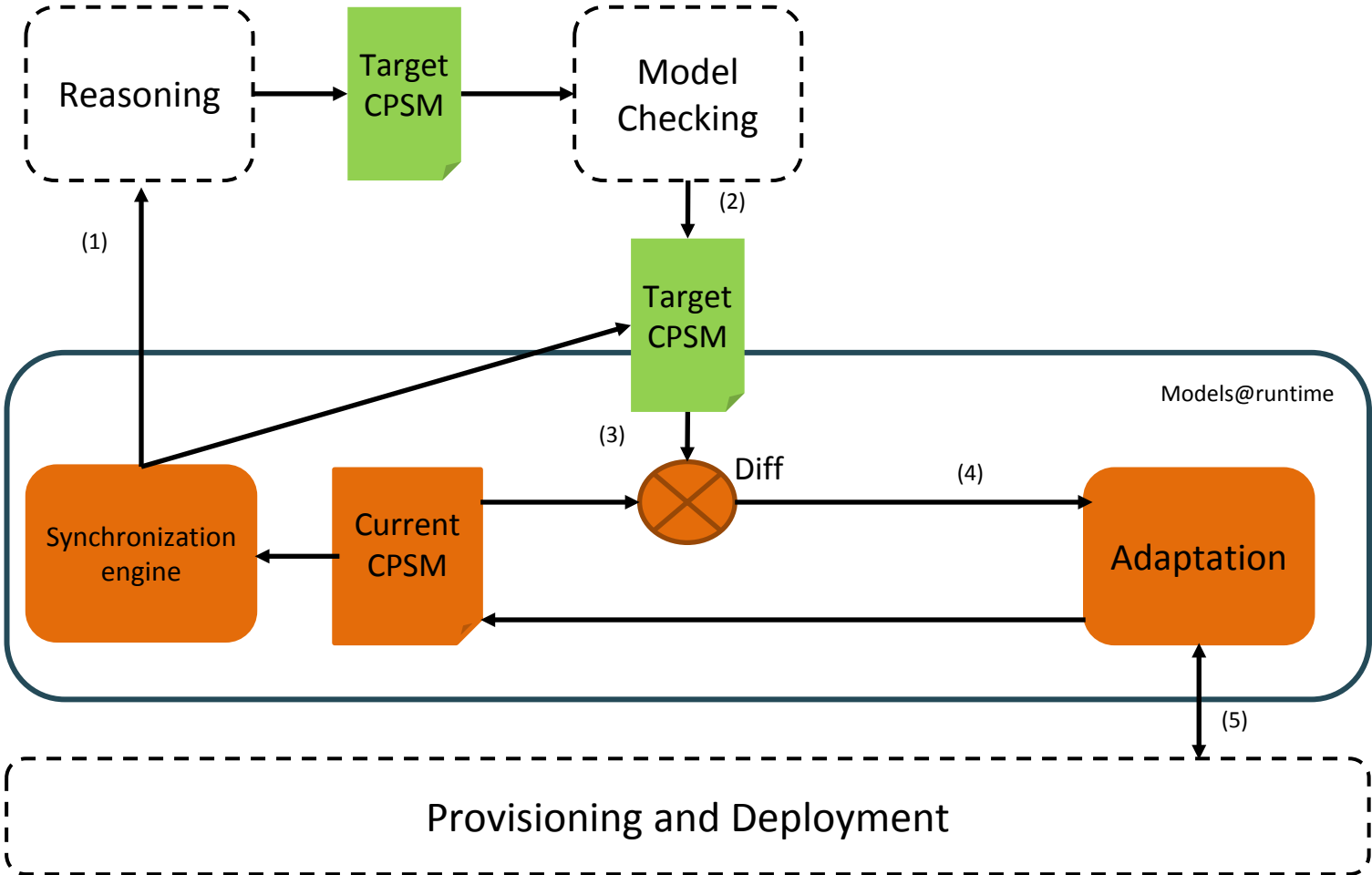
Comparison engine

- Priority is given to the target model
- Comparison is not only about the IDs

High level operations

- **Bursting** can be triggered by:
  - Providing a **new deployment** model
  - Dynamically, using the **scale out** command
- The **scale out** command:
  1. Updates the target model by replicating the VM and its hosted components
  2. Creates an image of the VM to be scaled
  3. Reconfigures all components after scaling
  4. Restarts the new components

Demo

Conclusion

- Towards a unique language for developers and operators

- Facilitate interactions between Dev & Ops for better quality

- The models@runtime pattern can be applied to other type of models (e.g., QoS models)

Thank you !

?
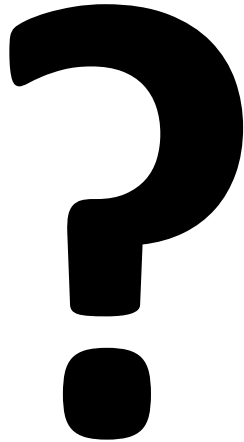
Contact: *nicolas.ferry@sintef.no*

Prepared questions

- How the Model@runtime engine could be extended to support automated testing and how the comparison engine can be beneficial?

- Continuous delivery and management tools can be a threat to your project!
  - Considering a team that fully relies on these tools: what if the tool chain is broken or one of the tool fails; what about their security etc. The impact of such failure can be dramatic.
  - In the early stage of a project, QA should also focus on analysing the risks of adopting typical "DevOps" solutions and prepare back up plans!