Towards evolving secured multi-model systems with model federation

Chahrazed Boudjemila

Fabien Dagnat

Salvador Martinez

MODELS 2023, ME, Västerås, Sweden

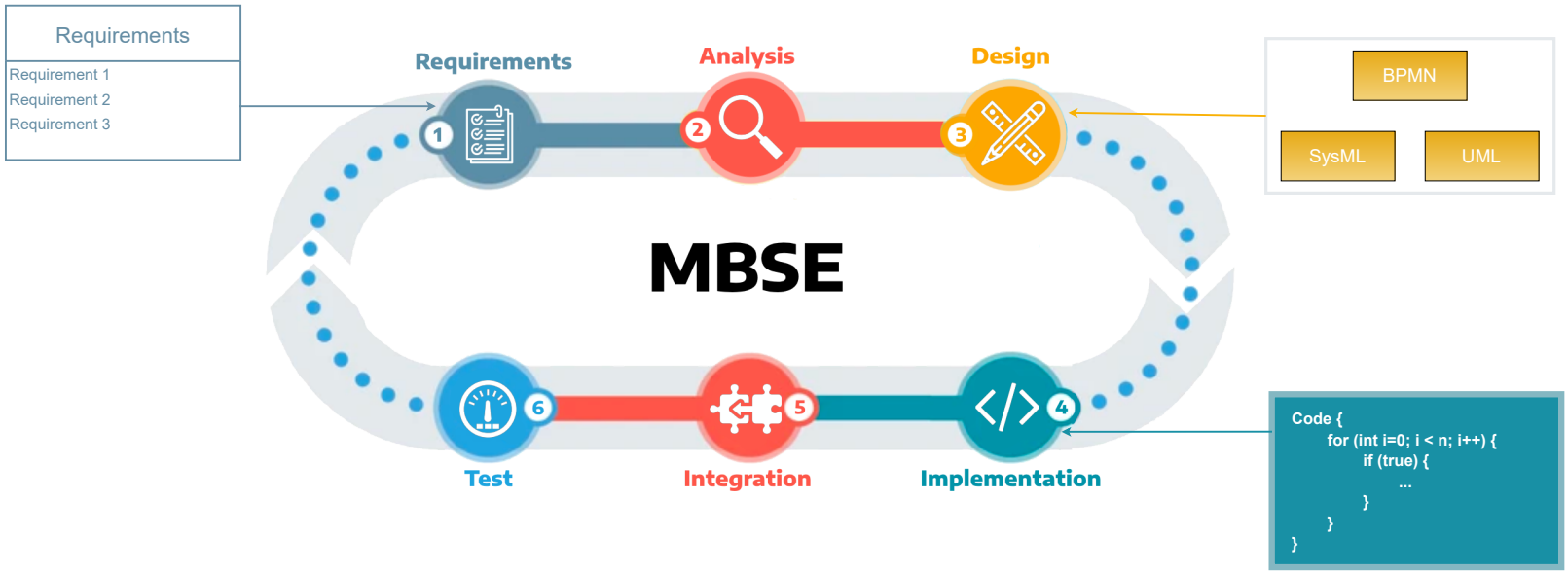
IMT-Atlantique

2023-10-02

Table of contents

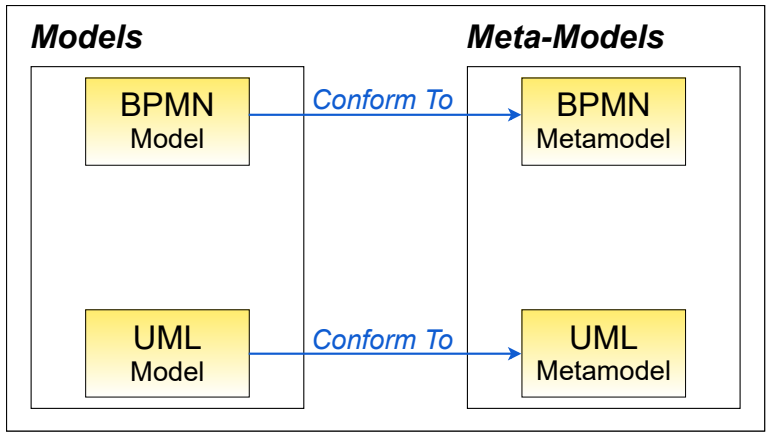
Introduction: Context

- Model-based system engineering (MBSE): Methodology dealing with complex systems that uses models in all engineering phases



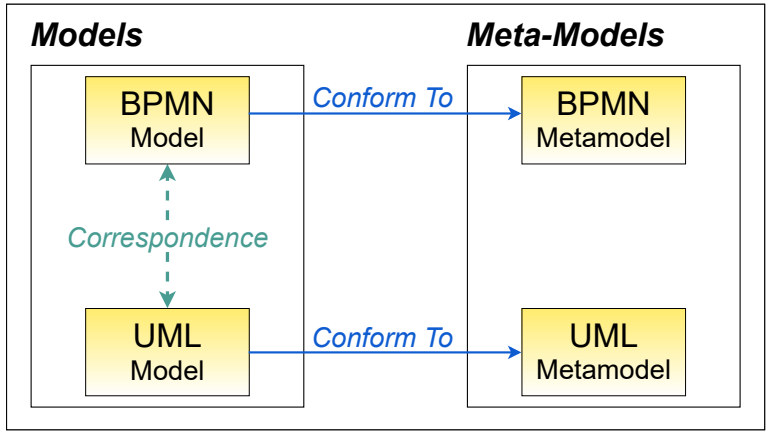
Introduction: Context

- Heterogeneous models
  - Models conform to different modeling languages



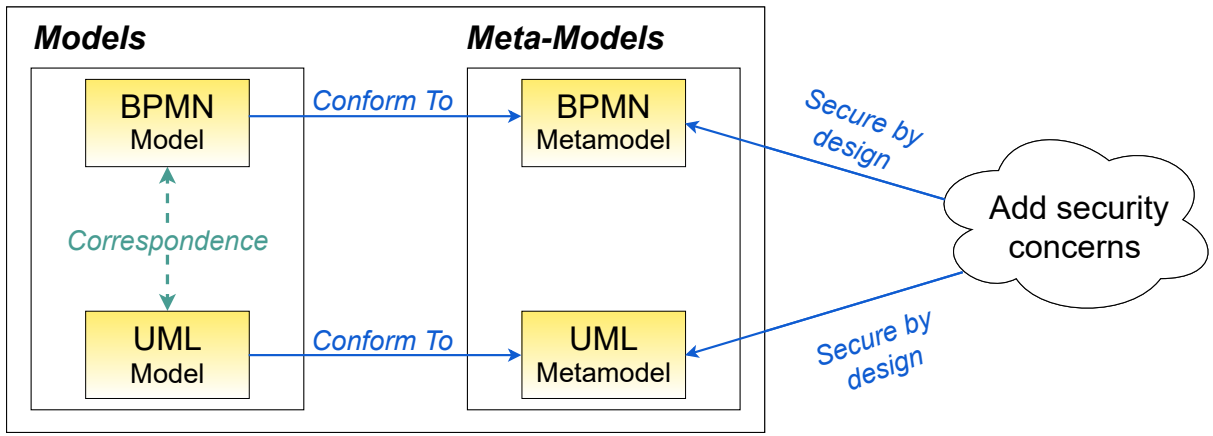
Introduction: Context

- Heterogeneous models
  - Models conform to different modeling languages
  - Models are dependent



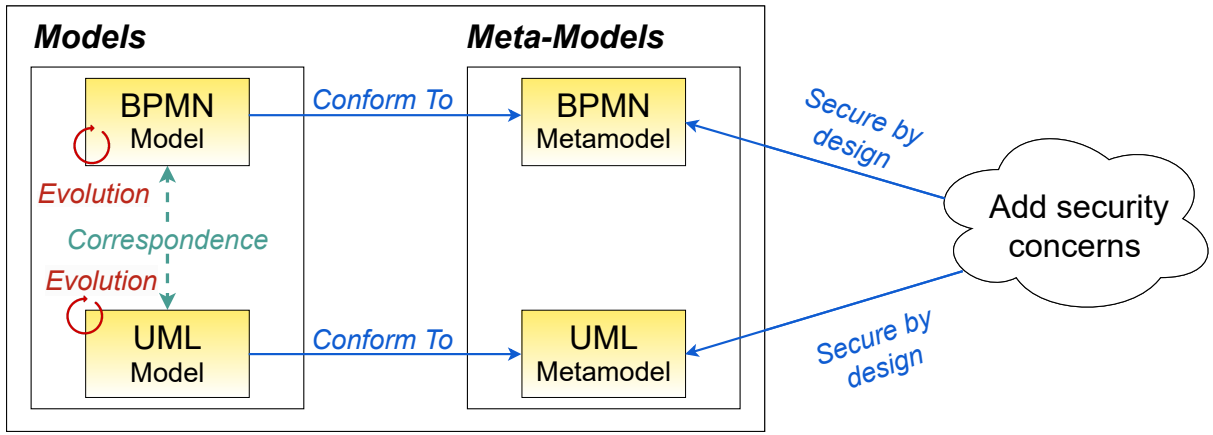
Introduction: Context

- Security: major concern in complex system
  - Incorporate security early in the system development process (security-by-design)



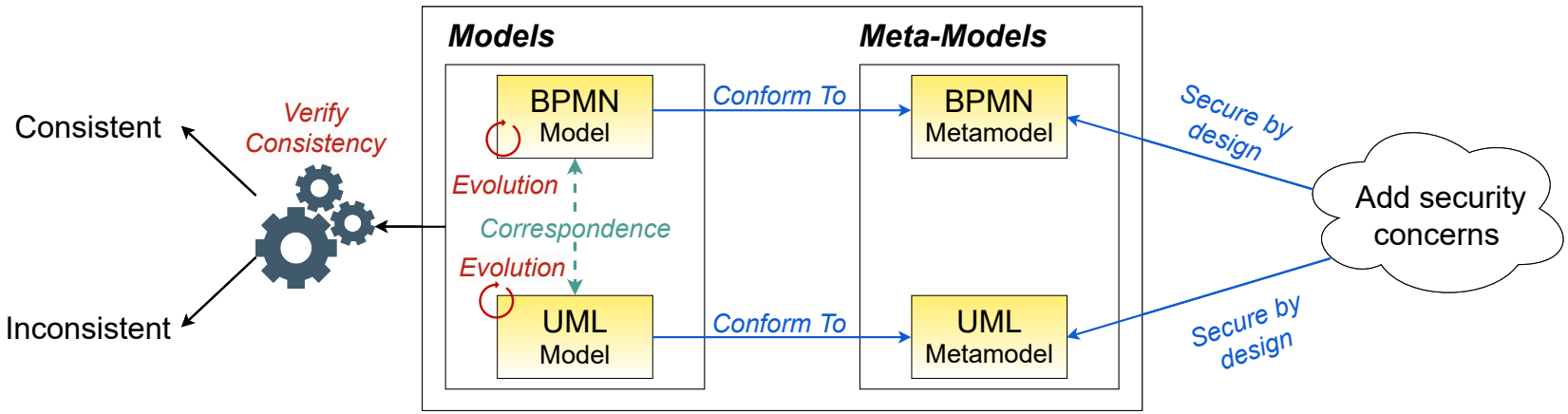
Introduction: Context

- System Models evolution
  - Models may change over time due to a variety of factors, such as changes in requirements and deployment
  - Changes may render models inconsistent which can lead to security issues



Introduction: Context

- System Models evolution
  - Models may change over time due to a variety of factors, such as changes in requirements and deployment
  - Changes may render models inconsistent which can lead to security issues



Problematic

- How to determine whether the security consistency of system models is preserved after their evolution?

Contribution

- Methodology
  - Enabling the secure consistent evolution of heterogeneous models

Contribution

- Methodology
  - Enabling the secure consistent evolution of heterogeneous models
  - Fits in Model-Based System Engineering (MBSE)

Contribution

- Methodology
  - Enabling the secure consistent evolution of heterogeneous models
  - Fits in Model-Based System Engineering (MBSE)
  - Follows the Security-by-Design paradigm

Contribution

- Methodology
  - Enabling the secure consistent evolution of heterogeneous models
  - Fits in Model-Based System Engineering (MBSE)
  - Follows the Security-by-Design paradigm
  - Based on the model federation approach

Methodology: process

Methodology: process

1- Design the security model federation

Methodology: process

1- Design the security model federation
2- Use the security model federation

Methodology: process

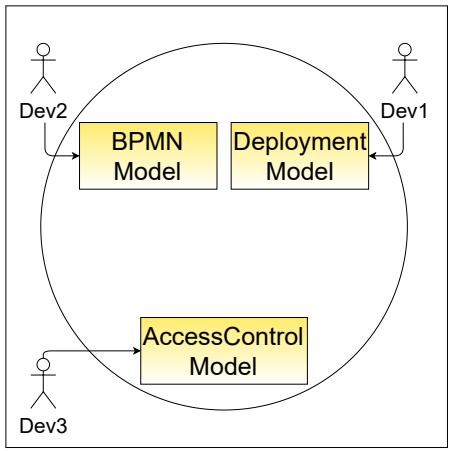
1- Designing the security model federation

- Design the secure system
- Build the security model federation

Methodology: process

1- Designing the security model federation

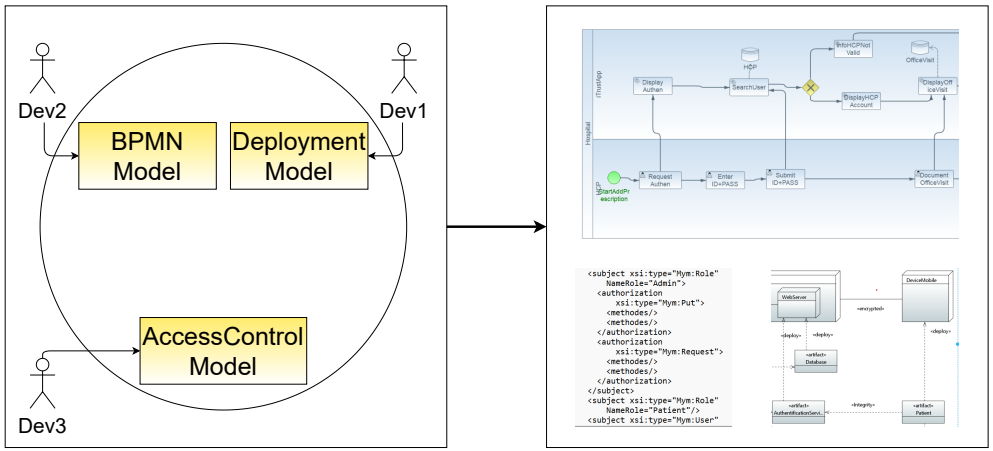
- Design the secure system
  - Select the appropriate modeling language and tool
  - Analyze the system's security requirements
  - Add security concerns to models



Methodology: process

1- Designing the security model federation

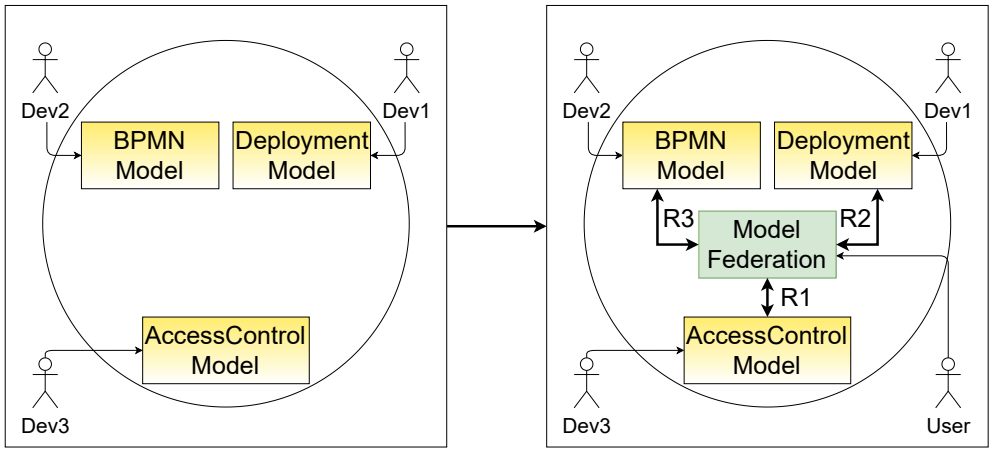
- Design the secure system
  - Select the appropriate modeling language and tool
  - Analyze the system's security requirements
  - Add security concerns to models



Methodology: process

1- Designing the security model federation

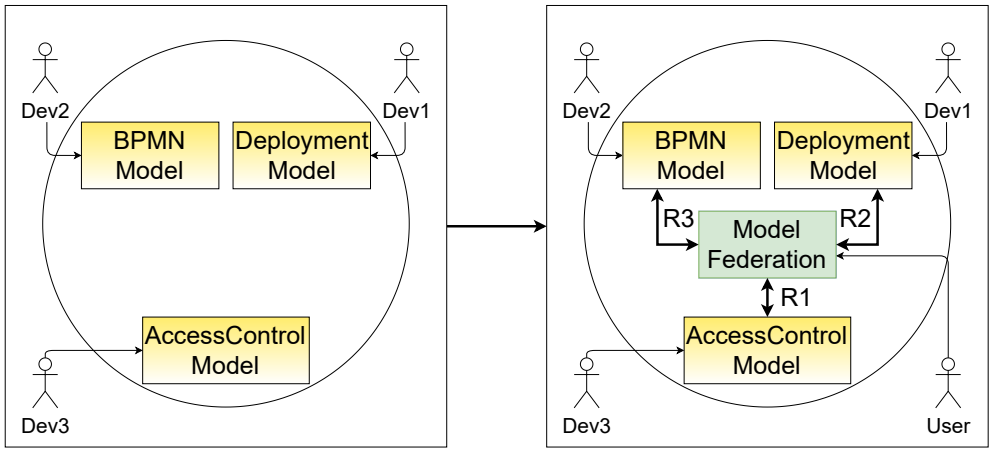
- Build the security model federation. The objective of this step is to: federate the models, define security consistency rules, and attach them to links



Methodology: process

1- Designing the security model federation

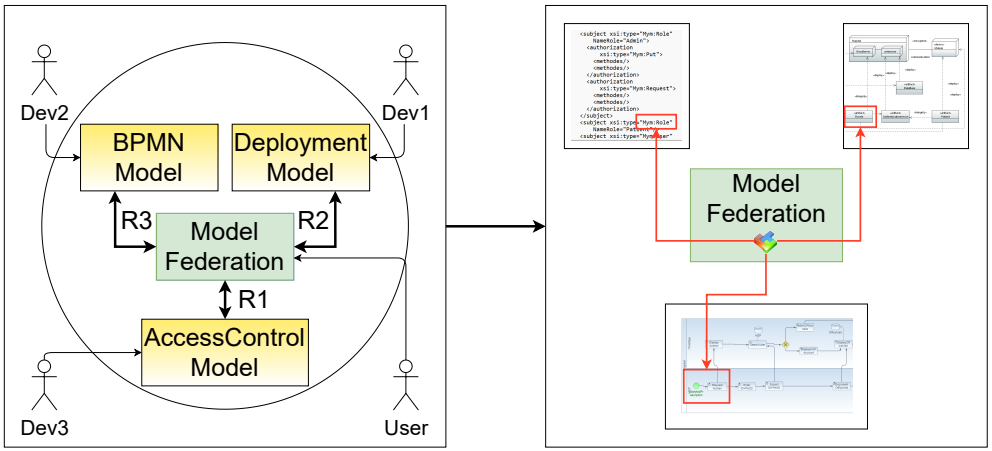
- Build the security model federation
  - Identify the dependencies



Methodology: process

1- Designing the security model federation

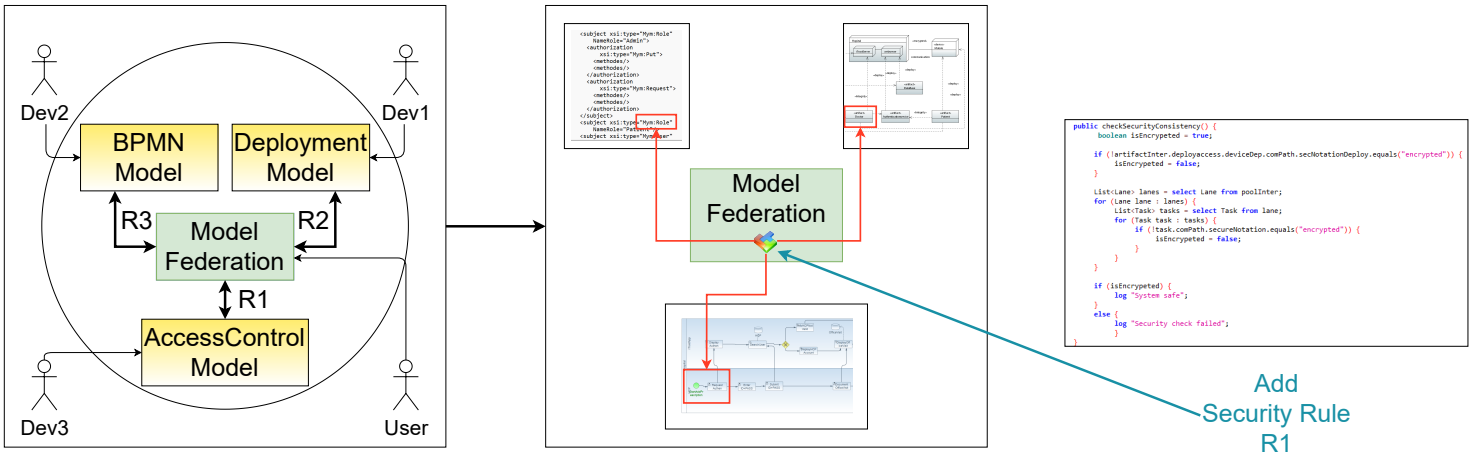
- Build the security model federation
  - Reify dependencies by creating explicit links between dependent modeling elements



Methodology: process

1- Designing the security model federation

- Build the security model federation
  - Define a set of security rules that represent the security consistency of models and attach them to links



Methodology: process

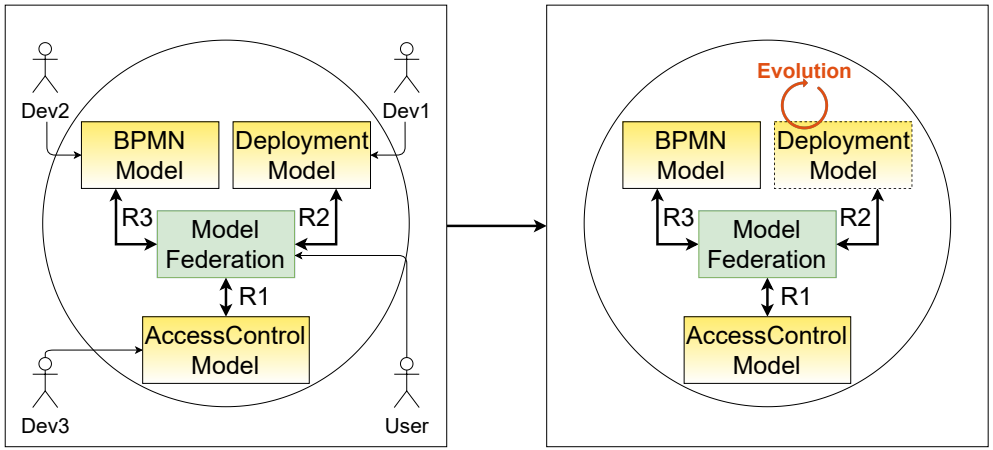
2- Using the security model federation

- Detect changes
- Analyze changes

Methodology: process

2- Using the security model federation

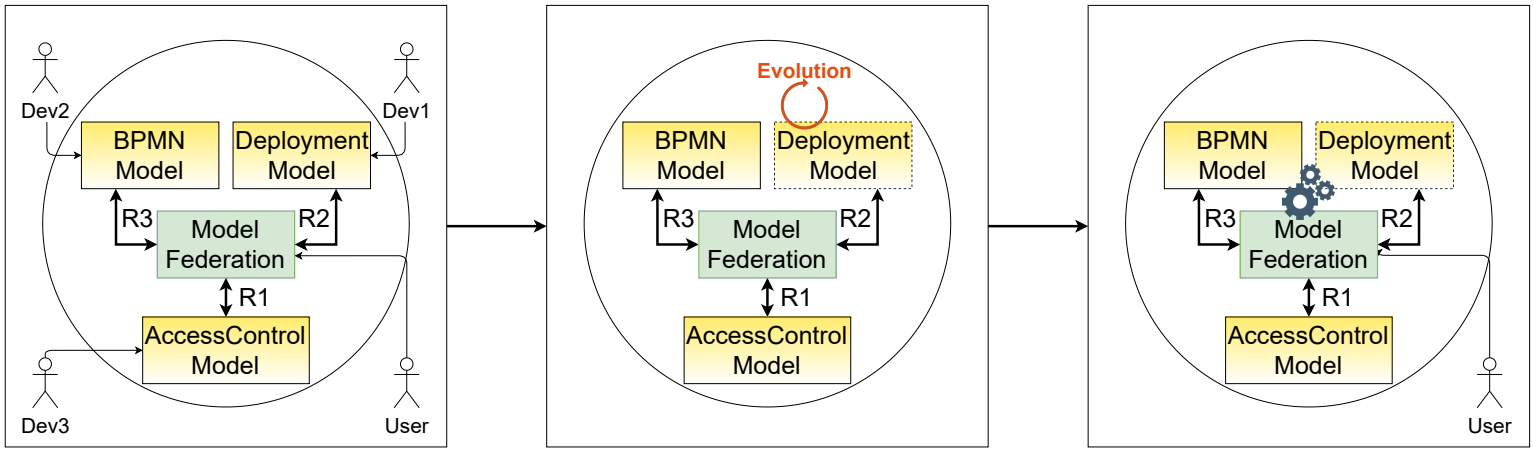
- Detect changes: detection may be performed manually (current) or automatically (future work)



Methodology: process

2- Using the security model federation

- Analyze changes: security rules attached to links that bind modified elements are (re)evaluated



Conclusion

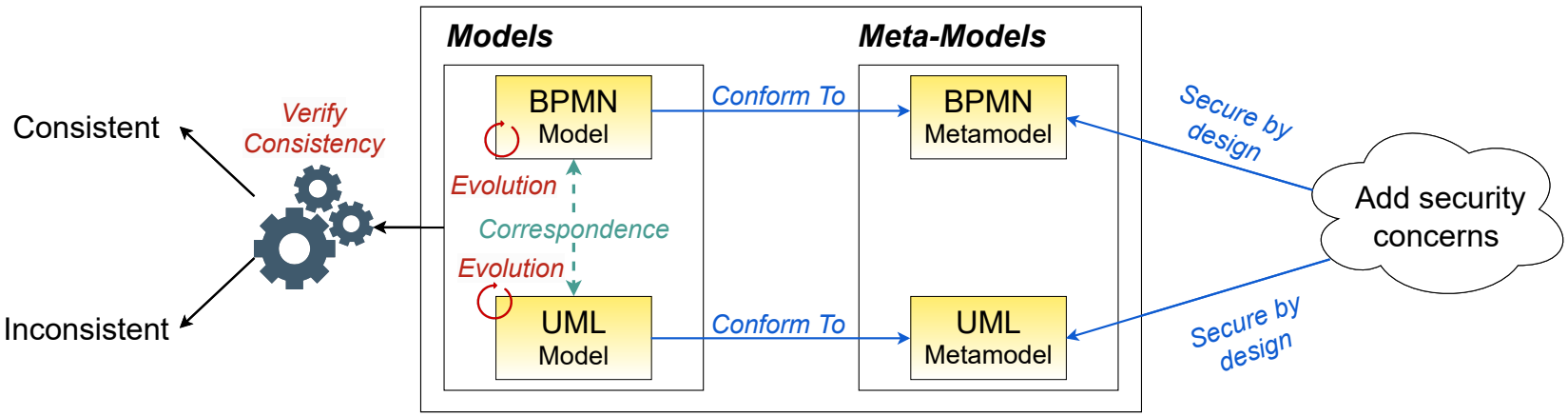
- We propose a methodology:
  - Enabling the secure evolution of models
  - Supporting heterogeneous models
  - Leveraging model federation

Conclusion: future work

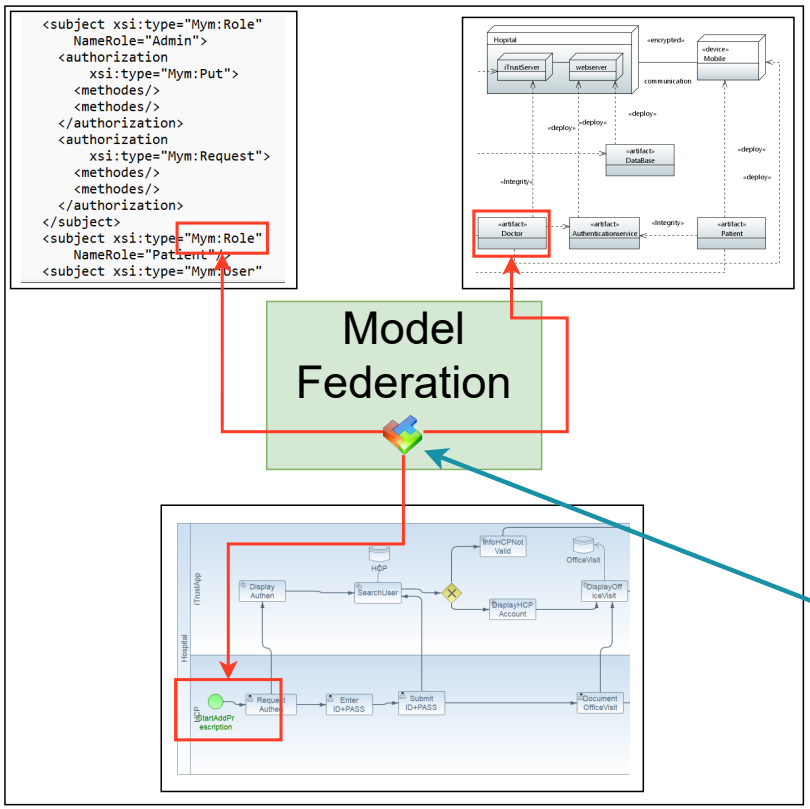
- Generalize a process that assists users in identifying security correspondences between heterogeneous models
- Include new types of models (at different abstraction levels)
- Automate the detection and evaluation of changes
- Validate our methodology on existing use cases

Conclusion

- We propose a methodology:
  - Enabling the secure evolution of models
  - Supporting heterogeneous models
  - Leveraging model federation



Example




```
public checkSecurityConsistency() {
      boolean isEncrypeted = true;

    if (!artifactInter.deployaccess.deviceDep.comPath.secNotationDeploy.equals("encrypted")) {
         isEncrypeted = false;
    }

    List<Lane> lanes = select Lane from poolInter;
    for (Lane lane : lanes) {
         List<Task> tasks = select Task from lane;
         for (Task task : tasks) {
              if (!task.comPath.secureNotation.equals("encrypted")) {
                   isEncrypeted = false;
              }
         }
    }

    if (isEncrypeted) {
         log "System safe";
    }
    else {
         log "Security check failed";
         }
}
```

Add
Security Rule
R1

- The model federation approach : its main objective the reification of a set of dependencies among a group of models. One notable feature of the model federa- tion approach is that federated models i.e., the models that participate in a given federation, can independently evolve in their original technical environment while adapters are used in order to connect them to the federation.
- OpenFlexo : framework provides us the infrastructure required to connect mul- tiple models while keeping each model in its technol- ogy space. Openflexo offers a Federation Model Language (FML) engine with an integrated model design environment which permits the design of models and their behaviour. Also, offers some mature technology adapters to connect external models to Openflexo. Among others we can find connectors to BPMN, excel, XML, EMF and OWL