Development and Control of Distributed Applications

MDA-based Approach for Implementing Secure Mobile Agent Systems

Slim Kallel, Molka Rekik, Monia Loulou, and Ahmed Hadj Kacem

ReDCAD Lab.

University of Sfax, Tunisia

Slim.kallel@fsegs.rnu.tn

Workshop on
Agent Oriented Software Engineering
AOSE 2012

04.06.2012

Security policies and mobility concepts in MAS

- Security rules play a relevant role in MAS, but
  - They are often only an afterthought during the development process
  - Managing these rules in ad-hoc manner negatively affects the quality of the system

- Mobile Agents
  - Migrate through the network from one location to another
  - Be closer to services and resources to adequately perform its goals

Ensuring security rules

Ensuring that MAS exhibits security rules is crucial in many application domains

- Static verification
  - Most reliable means for this purpose
  - Not always feasible
- Runtime verification
  - A formal specification of the security properties to be verified
  - Mechanisms to verify the specified properties during program execution

Motivation

- The security rules are generally used only to control the resources access of mobile agents.

Motivation

- The security rules are generally used only to control the resources access of mobile agents.
- Lack of a holistic and generic approach for security policies in MAS
  - These policies are often ignored during the development process
  - The existing approaches do not consider the implementation on several mobile agents platforms; they take into account only one platform.

Motivation

- The security rules are generally used only to control the resources access of mobile agents.
- Lack of a holistic and generic approach for security policies in MAS
- The impact of security policies on modularity
  - The code implementing security policies is not modularized
  - The evolution problem

Our approach

MDS4MAS:

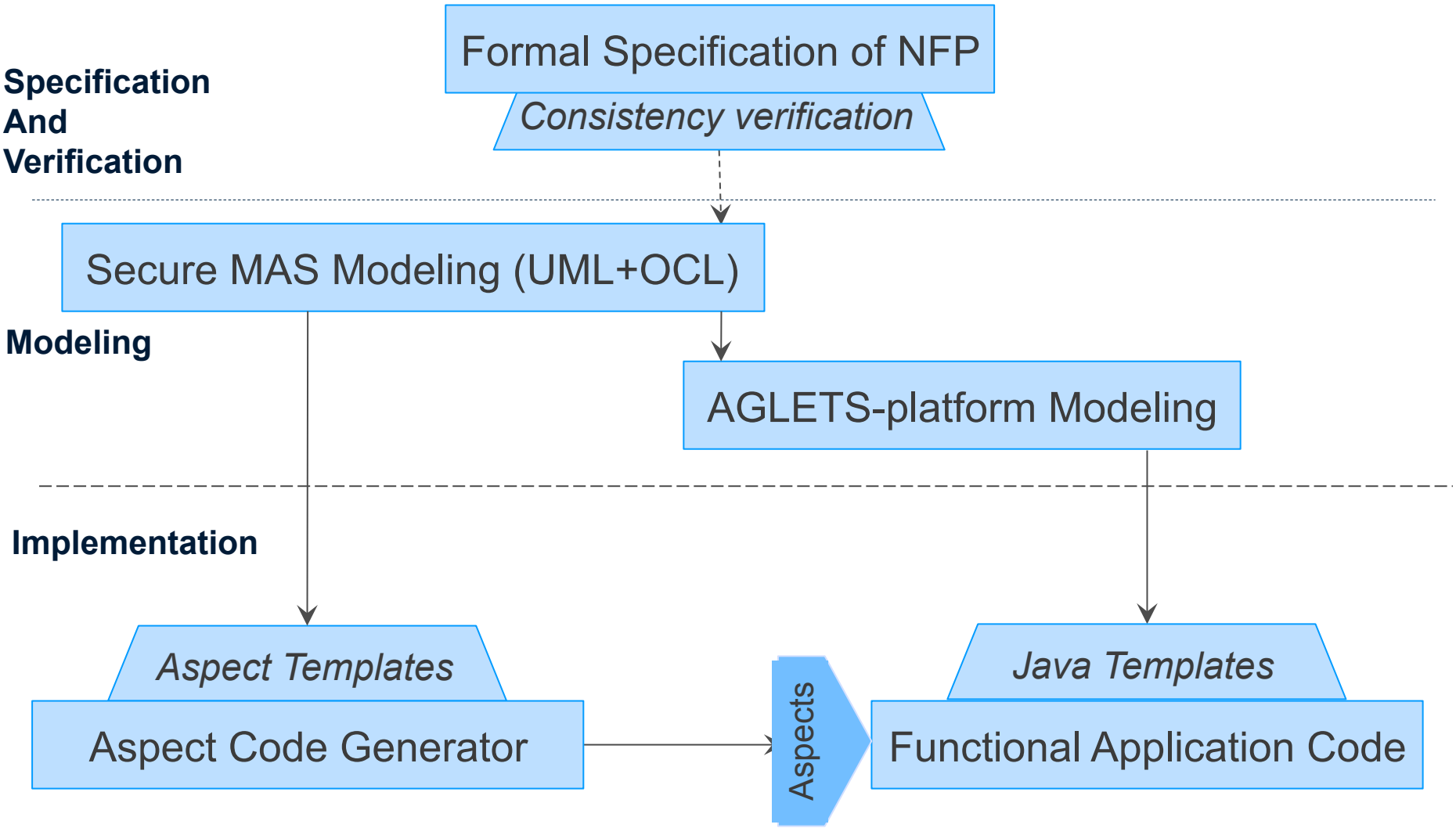
Model Driven Security for Mobile Agent Systems

Our approach

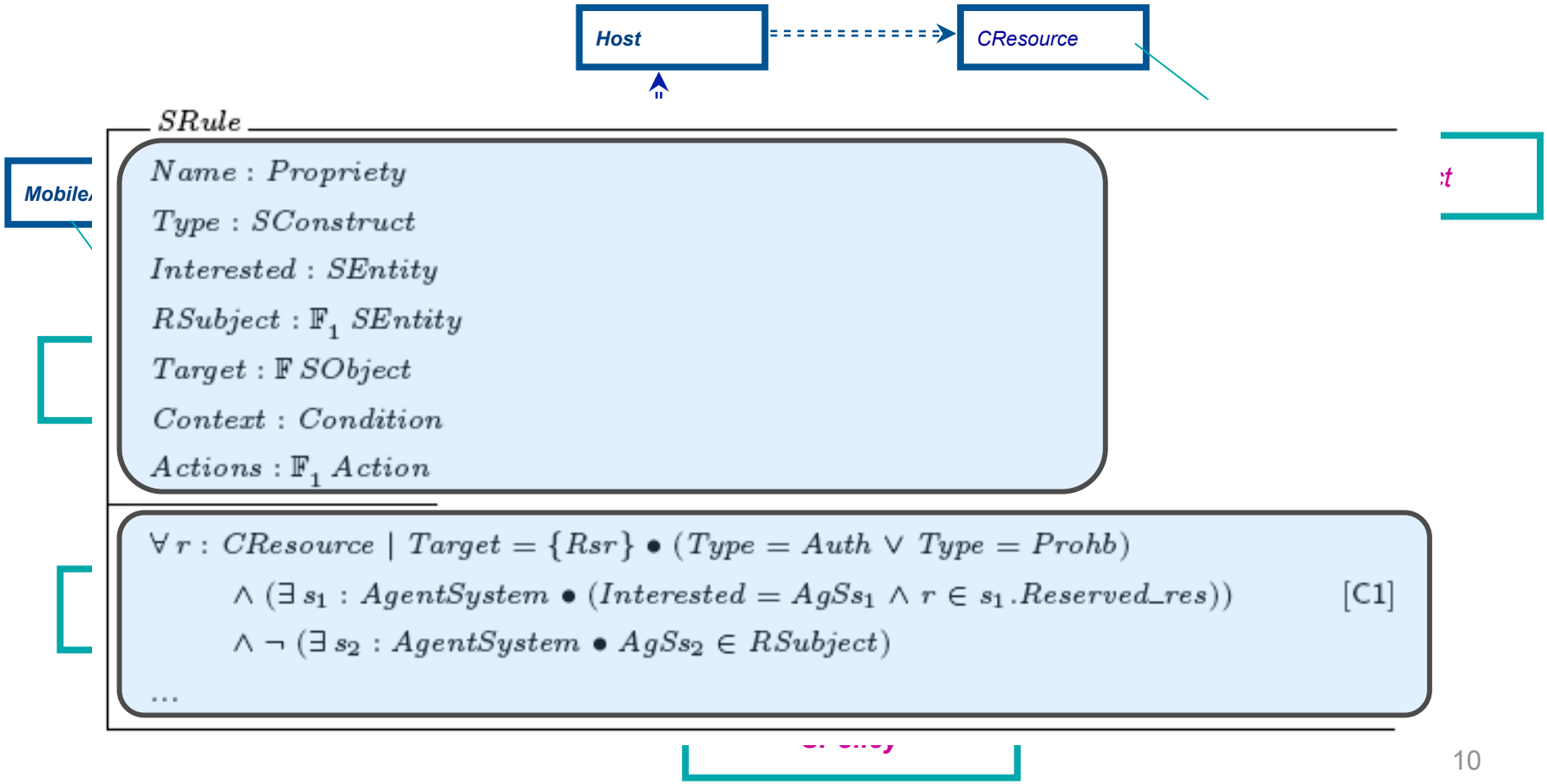
MDS4MbAS: An end-to-end approach for specifying and enforcing security policies in Mobile Agent Systems

- Combines formal methods, model driven architecture, and aspect-oriented programming
- Covers the whole development process, including
  - Formal specification and verification
  - Platform Independent and specific modeling
  - Runtime enforcement
- Bridges the gap between the development process steps

MDS4MAS Approach



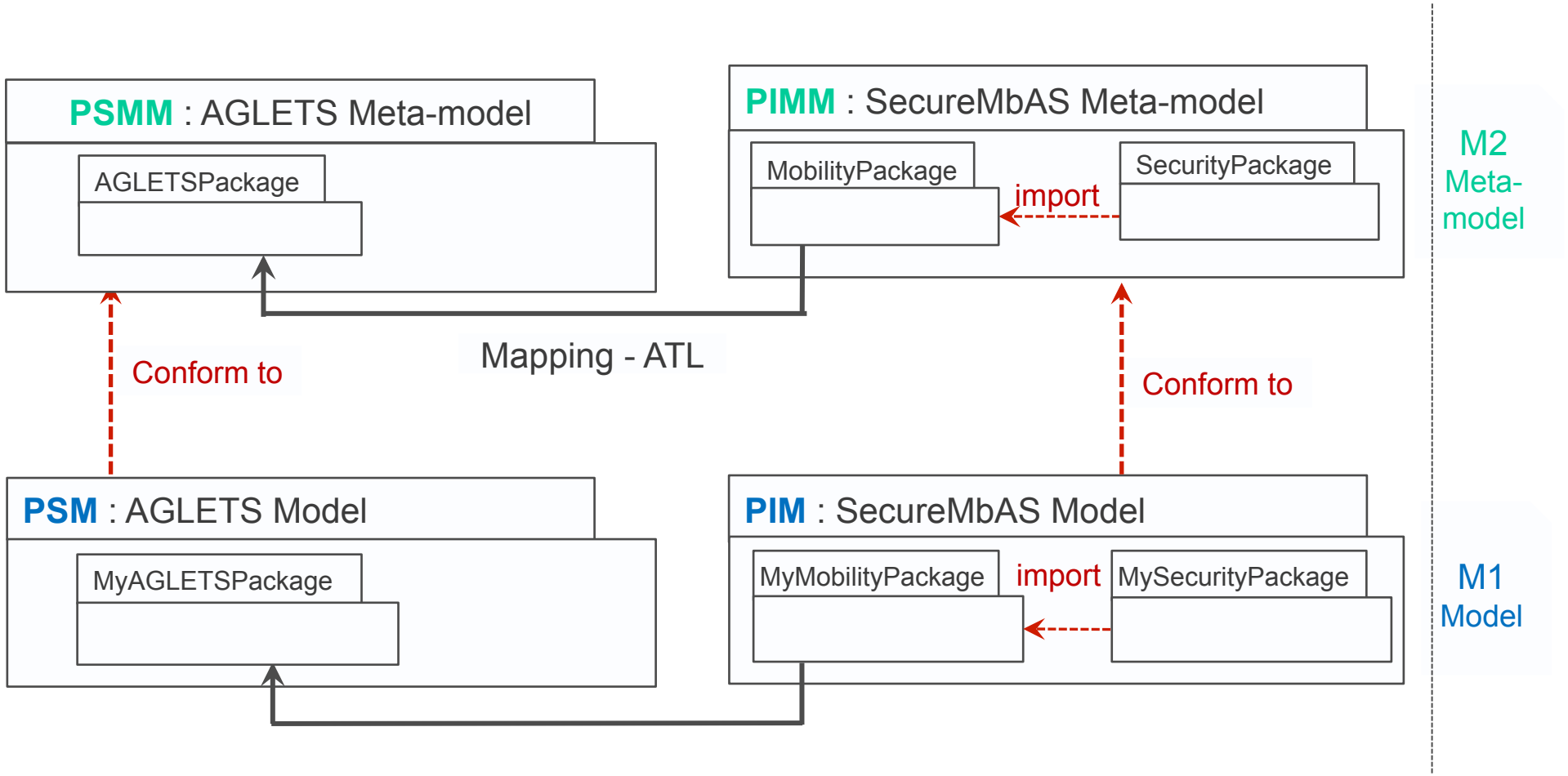
Formal Specification



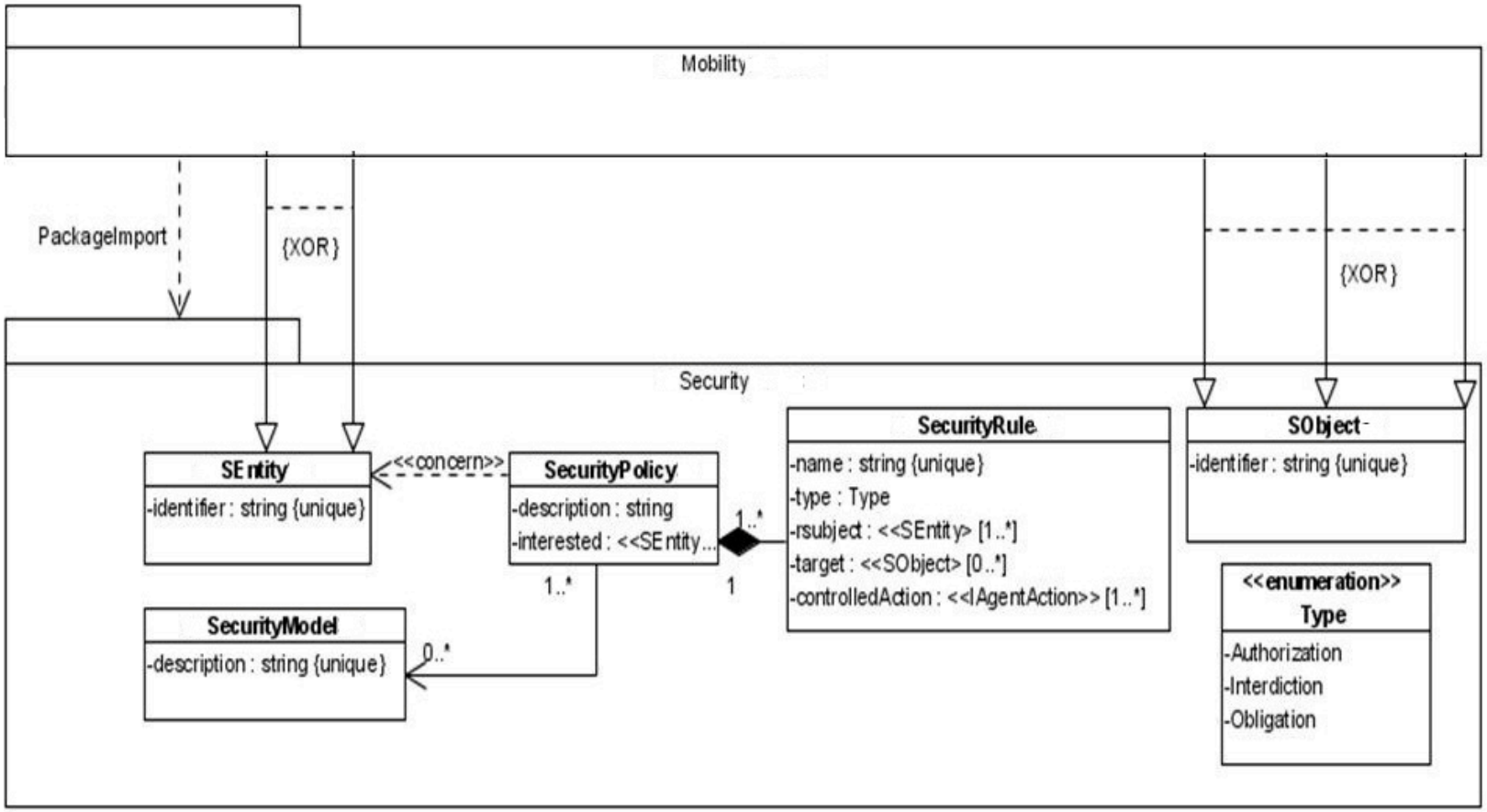
Formal Verification

- Using Z/EVES, as a semi-atomatic theorem prover, we formally verified
  - The consistency of Z specications
  - Other domain-specific properties.

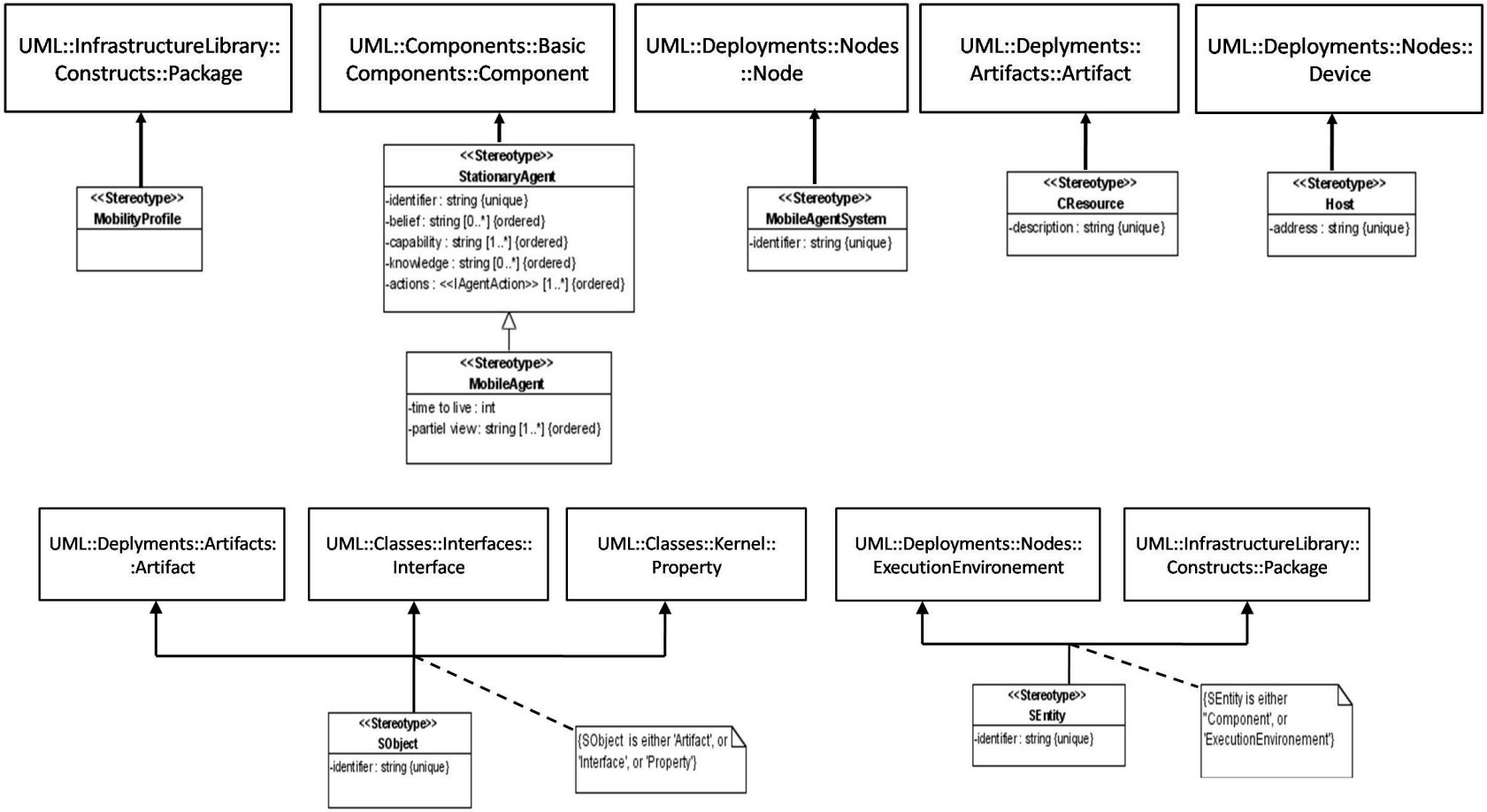
MDA based approach



PIMM

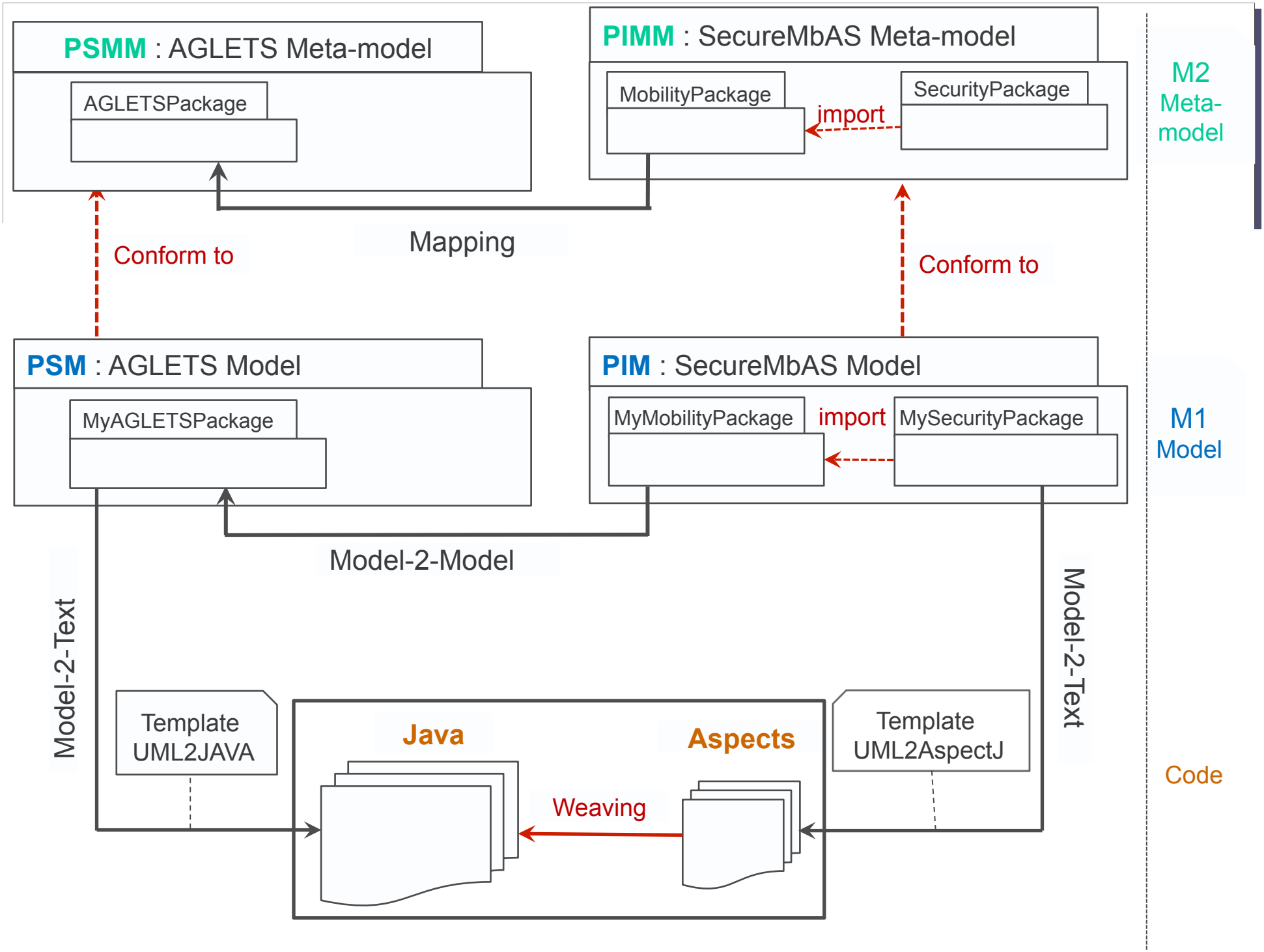

Mobility
PackageImport
{XOR}
{XOR}
Security
SEntity
-identifier : string {unique}
<<concern>>
SecurityPolicy
-description : string
-interested : <<SEntity...
1..*
1..*
1
SecurityModel
-description : string {unique}
0..*
SecurityRule
-name : string {unique}
-type : Type
-rsubject : <<SEntity> [1..*]
-target : <<SObject> [0..*]
-controlledAction : <<IAgentAction>> [1..*]
SObject
-identifier : string {unique}
<<enumeration>>
Type
-Authorization
-Interdiction
-Obligation

Part of the UML profile



Code generation process

- The code generation process is composed of two parts:
  - The generation of the functional code including the mobility concepts.
    - A Java code is automatically generated from the AGLETS specific model, which does not contain any security related code.
  - The generation of security code, which is obtained automatically from the OCL constraints describing the security constraints.
    - An aspectJ aspect is automatically generated from each OCL constraints


PSMM : AGLETS Meta-model
AGLETSPackage
PIMM : SecureMbAS Meta-model
MobilityPackage
import
SecurityPackage
M2
Meta-
model
Conform to
Mapping
Conform to
PSM : AGLETS Model
MyAGLETSPackage
PIM : SecureMbAS Model
MyMobilityPackage
import
MySecurityPackage
M1
Model
Model-2-Model
Model-2-Text
Model-2-Text
Template
UML2JAVA
Java
Aspects
Template
UML2AspectJ
Weaving
Code

Aspect Template

```
package MyAgletProfile;
public class Buyer extends Aglet
 {
   .......
   public void toBuy()
   {
    // Start of user code for
    operation toBuy
    // TODO should be
    implemented
     // End of user code
     }
  .......
 }
```

weaving

```
public aspect SR1{
  pointcut SR1pc() : execution (public *
    Buyer.toBuy(..));
  around () : SR1pc()
  {   ........
   if (allOCLConstraintsAreSatisfieted) {
      proceed();
    }
   else {
     System.out.println ("You can not
    execute this action ...");
    }
  }
}
```

pointcut

advice

Conclusion

- A framework for modeling mobile agent systems and their security policies.
  - A model specific to AGLETS platform automatically obtained.
  - A functional application code from the AGLETS model.
  - An AspctJ aspects to verify at runtime the specified security constraints.

Future Work

- Work on the MDS4MAS approach
  - Automatically generating UML model from Z formal specification
  - Verifying the model transformation
- Work on applying MDS4MAS to other MA platform
  - E.g, JADE
- Work on applying our approach for verifying and enforcing other properties of mobile agent systems
  - E.g., Temporal properties

Questions

Slim Kallel

Slim.kallel@fsegs.rnu.tn