fortiss

Nordwijk, 22th October 2015

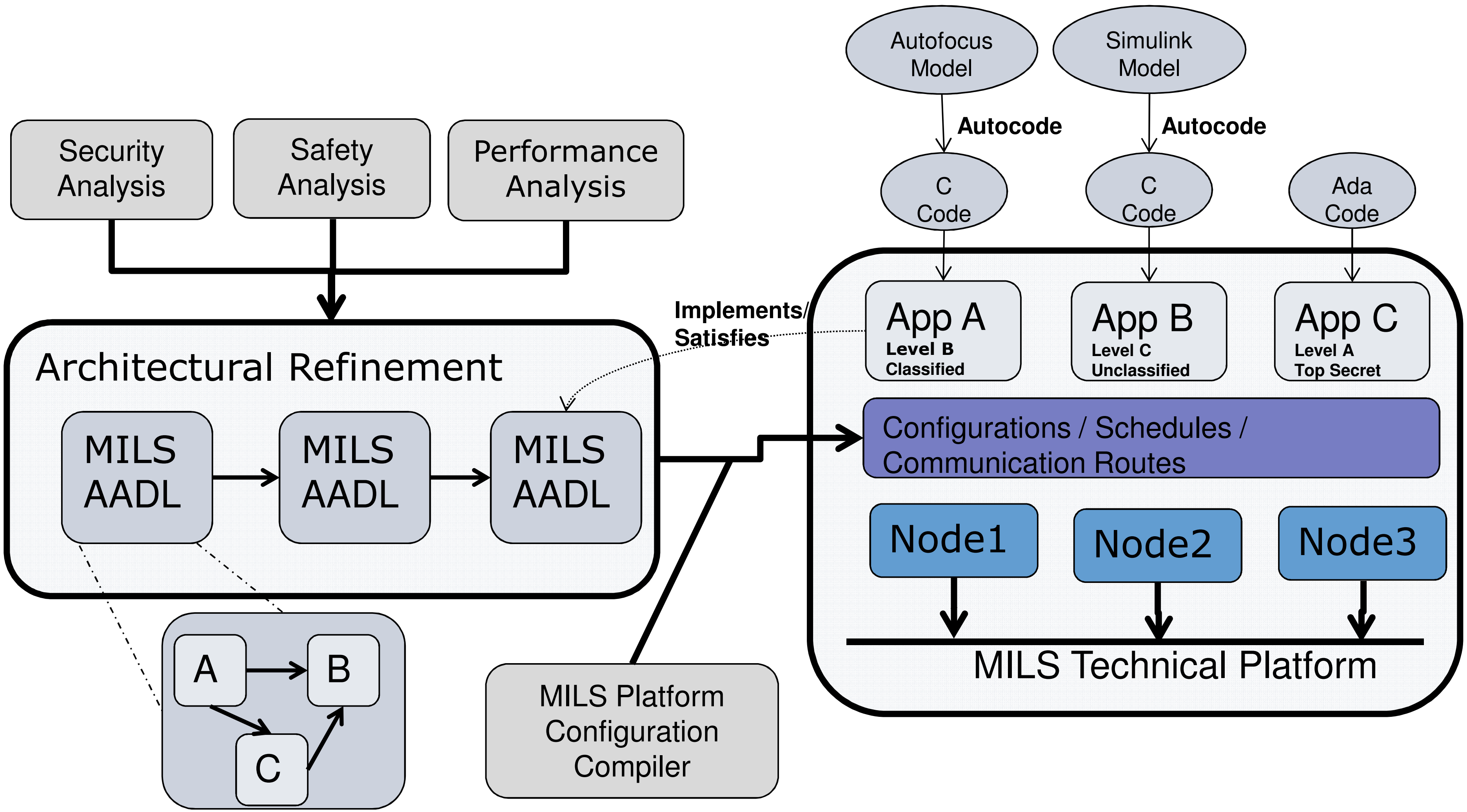
COMPASS in the D-MILS Project – An Experience Report

Dr. Harald Rueß

ruess@fortiss.org

fortiss GmbH
An-Institut Technische Universität München

MILS Architectural Design and Deployment



fortiss

MILS-AADL

MILS-AADL extends **COMPASS/SLIM** with

General language features

- Constant declarations
- Times specifications of system behavior supports time units
- *Pairs of values*
- **Key data types** *private key, public key (asymmetric), key (symmetric)*
- **Cryptographic operators** *decrypt, decrypt, verify*; algebraic specification
- **Discrete data types** *encrypted*, *signed*

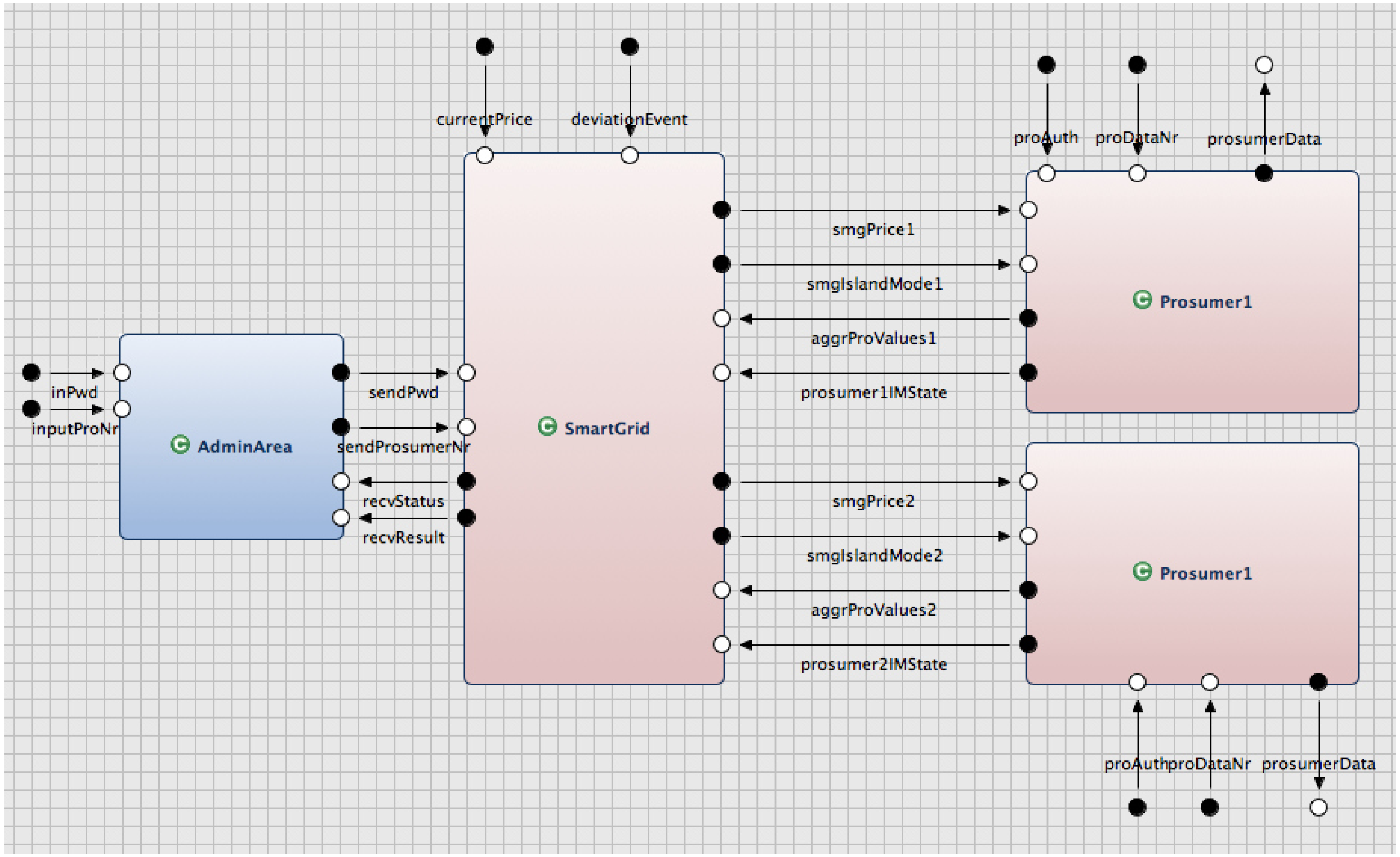
Nominal component specifications

- New component categories such as **network**, **node**
- Besides **event port** and **data port** also **event data port**
- Transmission of values along data parts uniformly represented by data flows
- Some keywords related to fault detection, isolation, and recovery omitted

Error model specifications

- Error states are now considered to be internal objects and are not visible to the environment

Case Study: Smart Microgrid in MILS-AADL



Clocks, continuous, modes, error modes,keys, encryption, ... (All is well)
Case study did not push probabilistic or real-time analysis to its limit

Logical specifications embedded in MILS-AADL

...

```
{nuXmv:
  LTLSPEC
    (G  ((data_#batteryStatus = full) ->
          ((data_#batteryStatus != full)
           V
           (!(event_charge_signal & (data_#chargingRate > 0)))
                 )));
 }
```

...

Pre-Xmas Wish List

Arguably the design of MILS-AADL is guided more by the capabilities of the underlying verification engines then by the needs for architectural modelling of complex systems

- Richer set of **discrete data types** needed; e.g. arrays and

```
datatype msg =
    atom(a: int)
    concat(m1, m2: msg)
    encr(m: msg, K: key)
```

- **Richer dynamics** needed
  - Currently only constant flows because of underlying decidability results
  - **But:** many real-world systems operate in complex environments described e.g. by means of non-linear stochastic differential equations
- **Temporal logic** deserves to be a first-class citizen in an architectural modeling language; e.g. contracts as integral part of component types / interfaces (a la Extended ML)
- **Refinement relation** e.g. on component types / interfaces
- Specification of **Dynamic and Adaptive Architectures**

An architectural language is only as good as its **supporting tools** (and tutorials and model repositories) and their **feedback** to the designer in already of **partial designs**

- Deadlock checking for infinite-state systems
- Fairness conditions (currently observed differences between simulator, nuXmv, and OCRA)
- (Symbolic/Concolic) Simulation / Exploration for partial designs
- Generation of e.g. certification documents
- Test-case generation
- Development environment: graphical editor, caching/Tracing of properties which have already been established
- **Analysis as a service** (instead of IDE) allows better integration in existing model-based tool chains

Harald Rueß

fortiss GmbH
Guerickestraße 25 · 80805 München · Germany

tel +49 89 3603522 0 **fax** +49 89 3603522 50

ruess@fortiss.org
www.fortiss.org