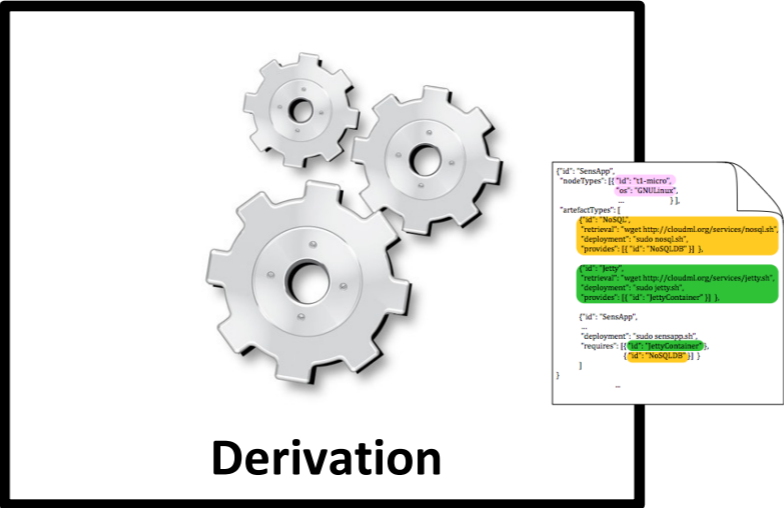
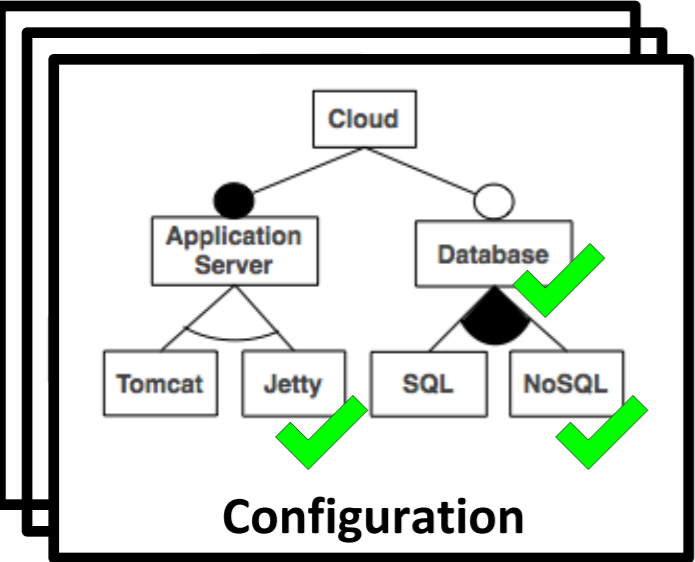
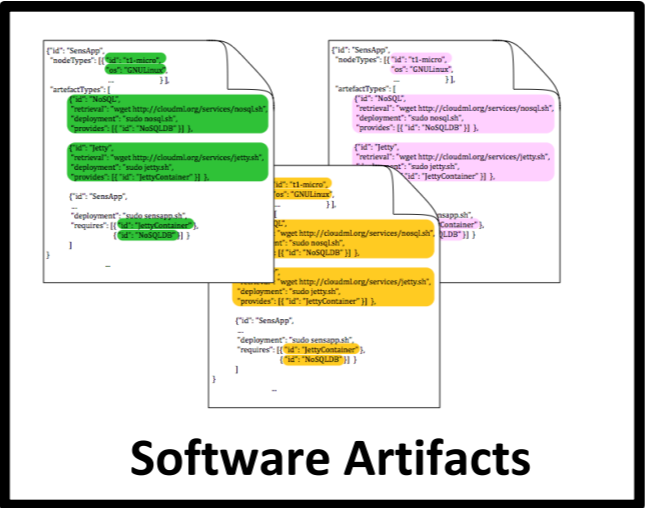
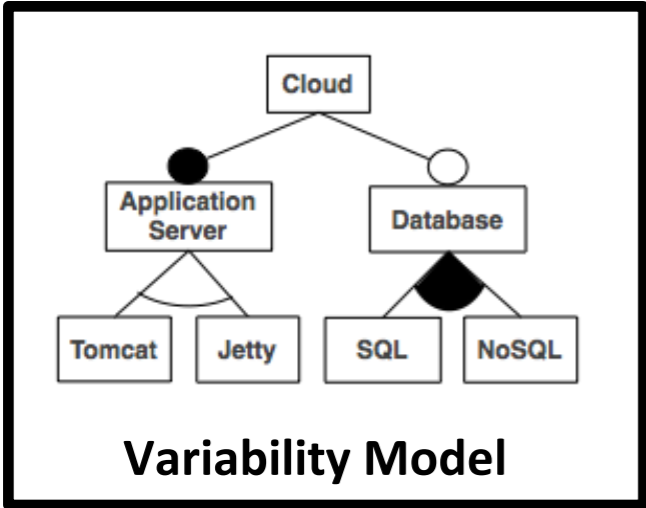


Evolution in Dynamic Software Product Lines

Clément Quinton Luciano Baresi

Domain Engineering

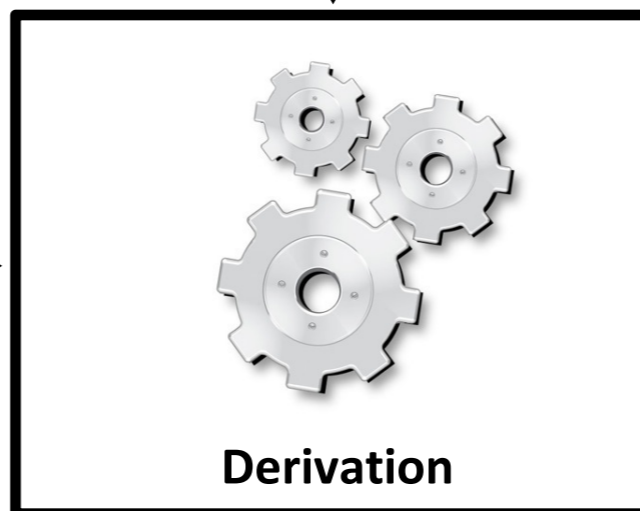
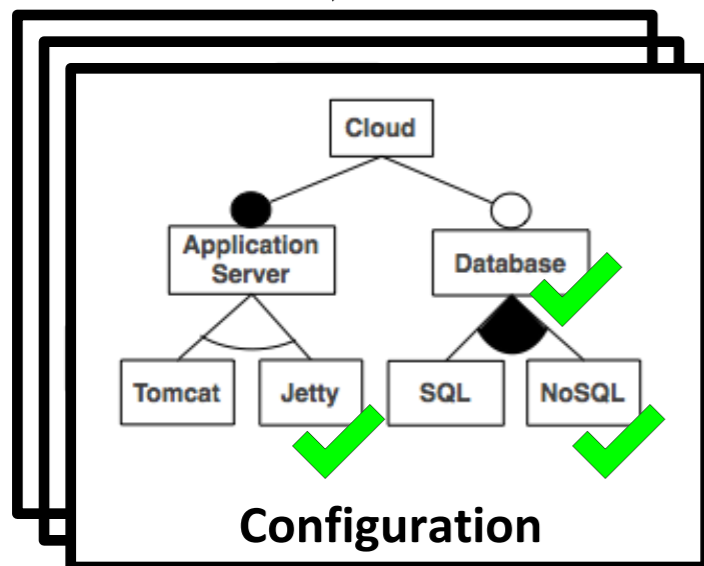
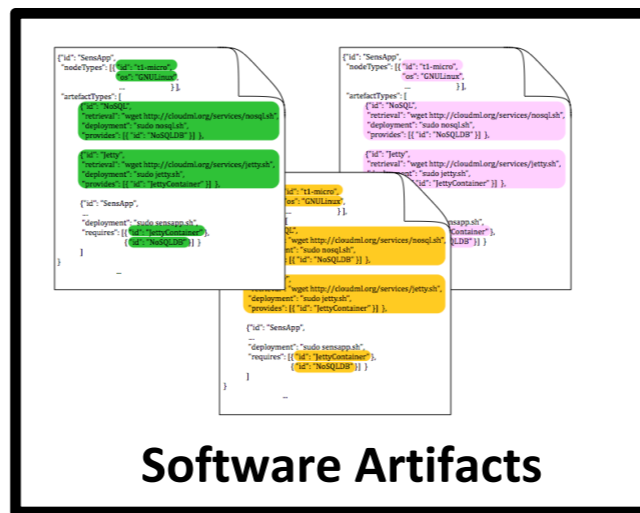
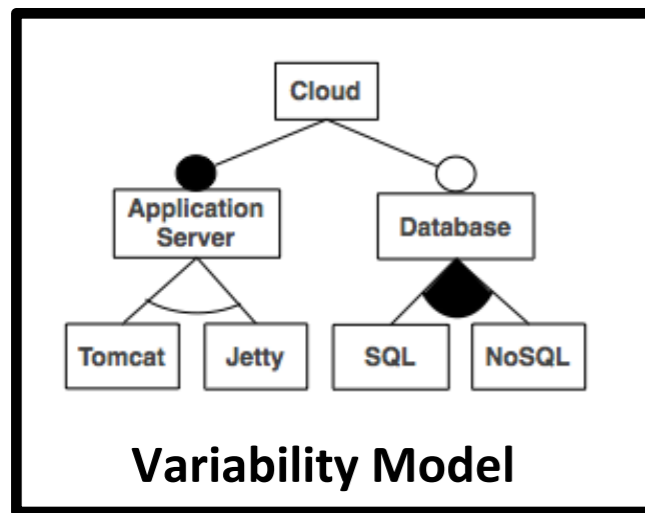


Application Engineering

Design Time

Runtime

Domain Engineering

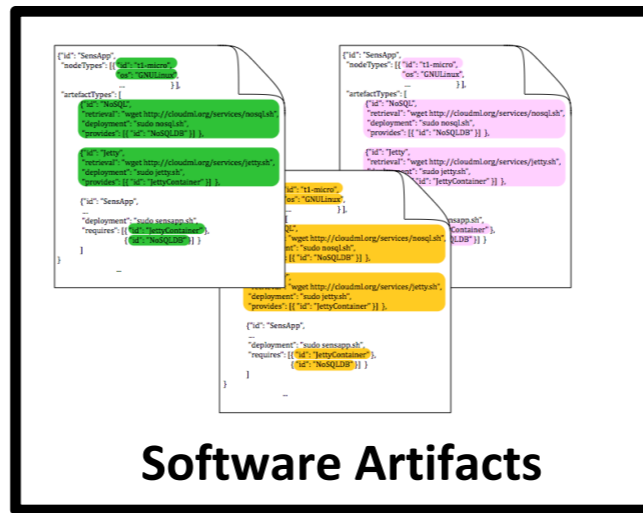
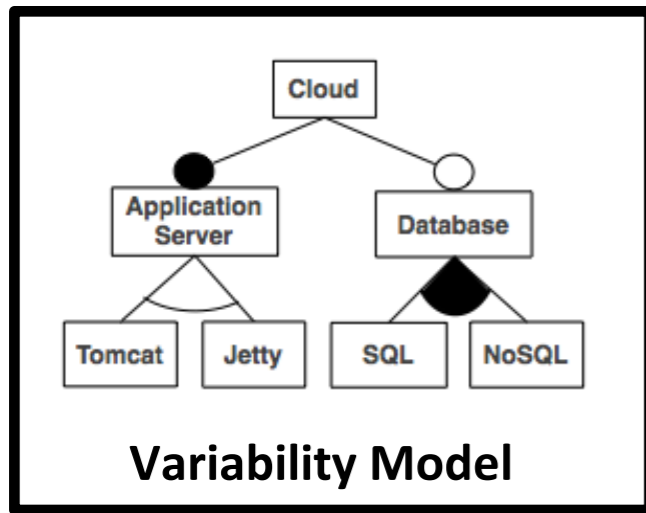


Application Engineering

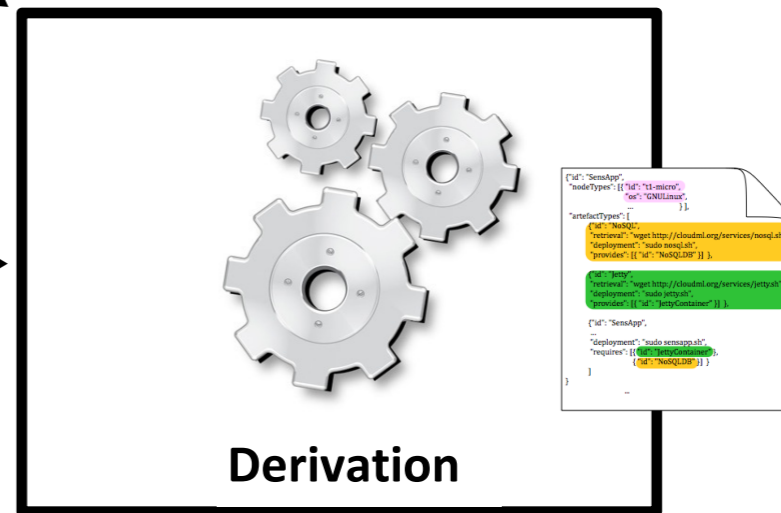
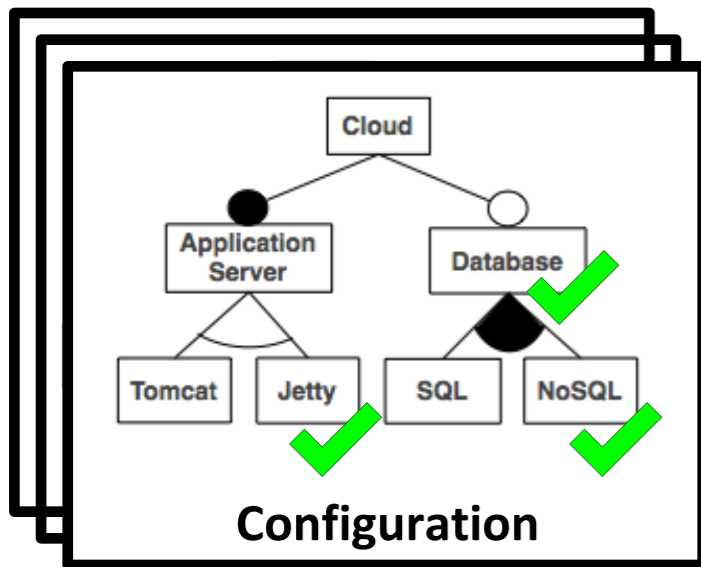
Design Time

Runtime

Domain Engineering



Adaptation
Reconfiguration

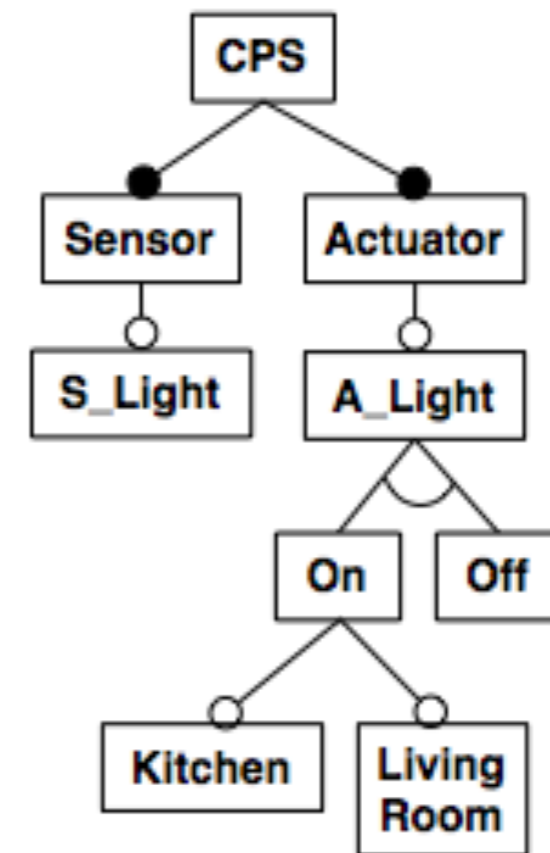
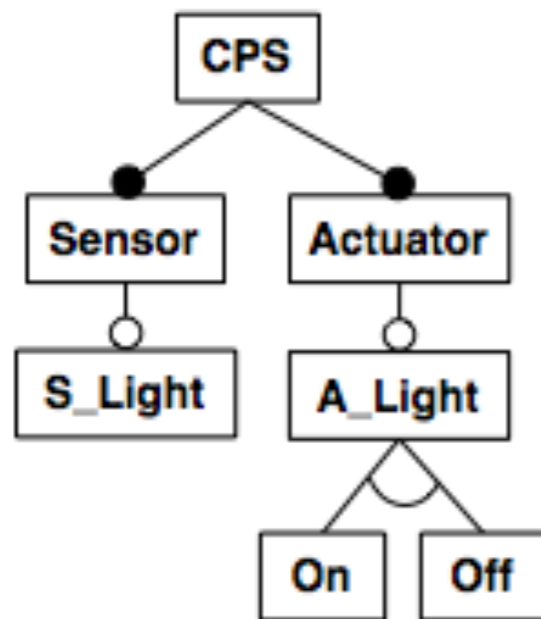


Application Engineering

Design Time

Runtime

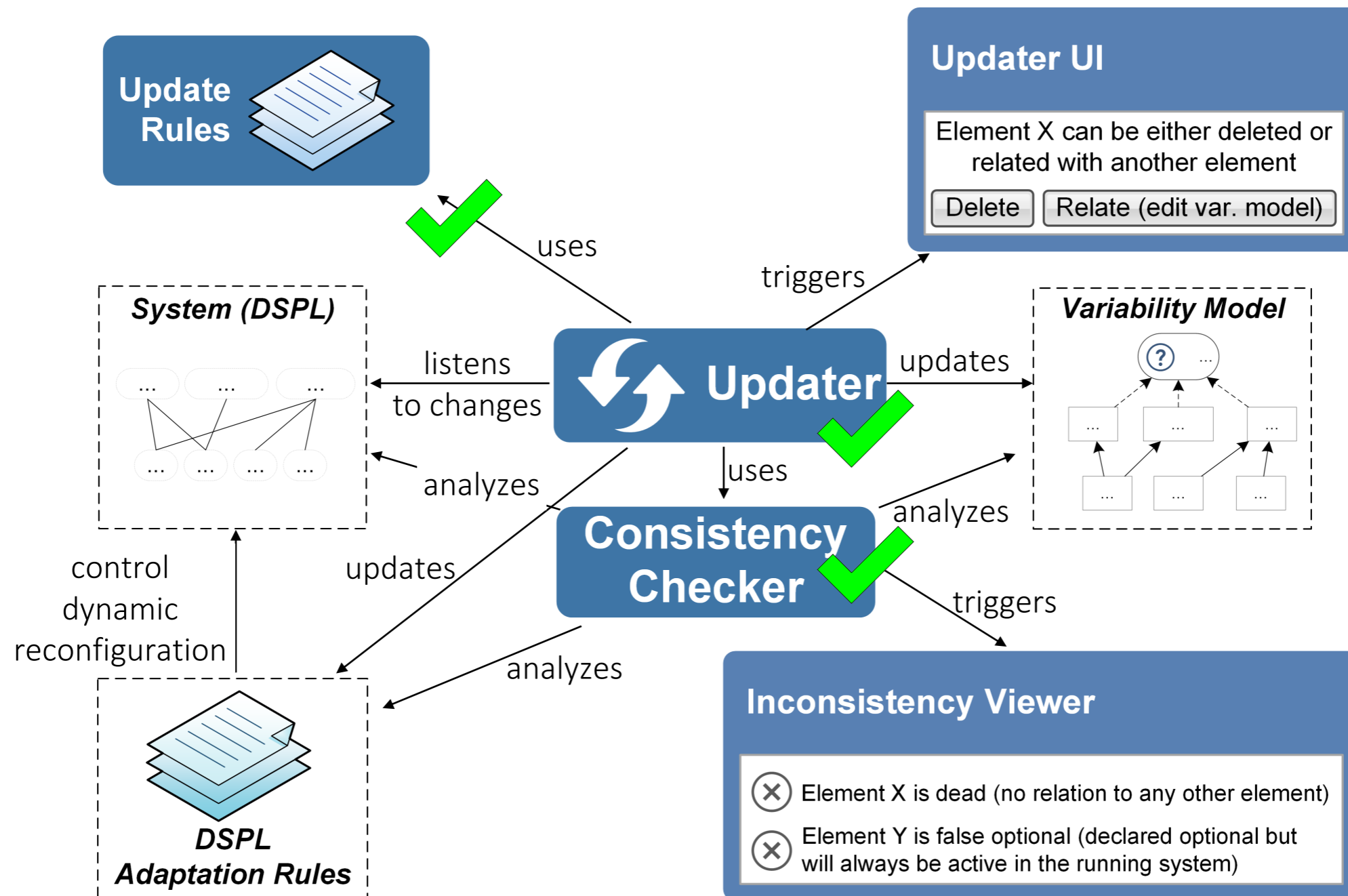
Consistency



luminosity < 40 lumens \rightarrow turnLightOn()
 (activates the *On* feature)

Automated evolution (based on rules, default solver choice)
Manual evolution (user choice)

Human in the loop!

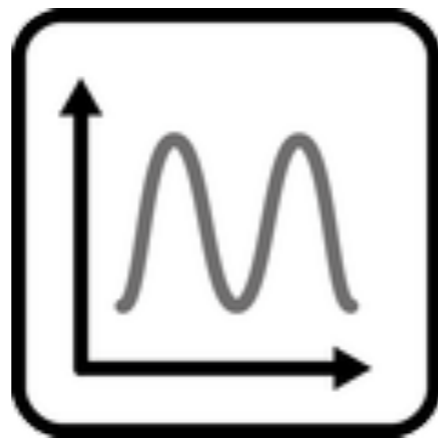


Learning and Evolution in Dynamic Software Product Lines

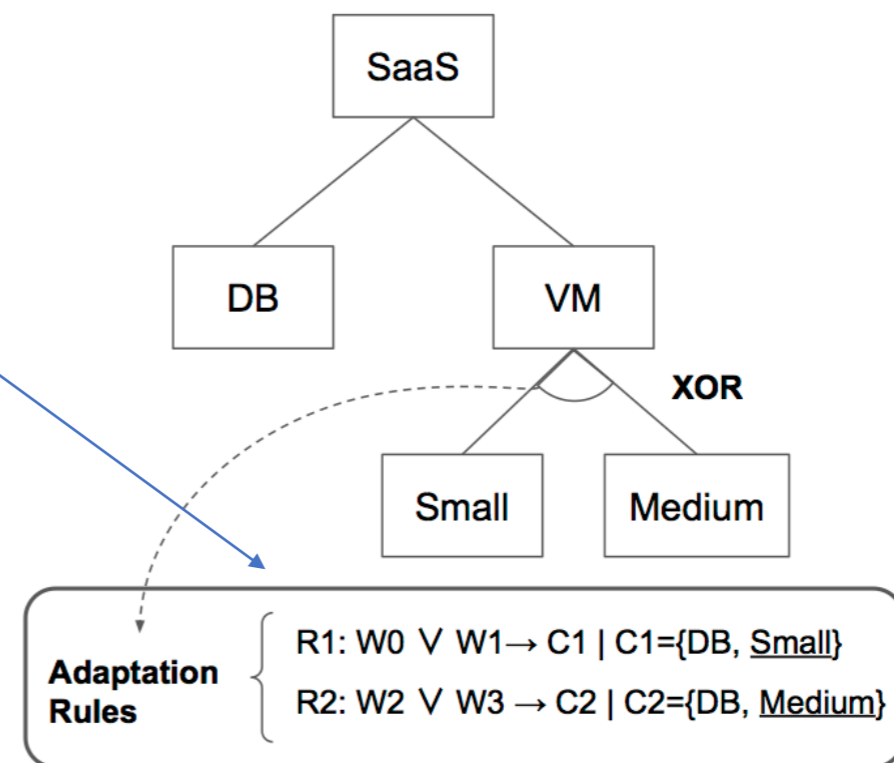
Clément Quinton and Luciano Baresi

Configuration vs. Adaptation Space

- Adaptation rules (policies, models, ...) define actual configurations that may be reached under certain context situations
→ define the DSPL's adaptation space
- Example: Simple Cloud System

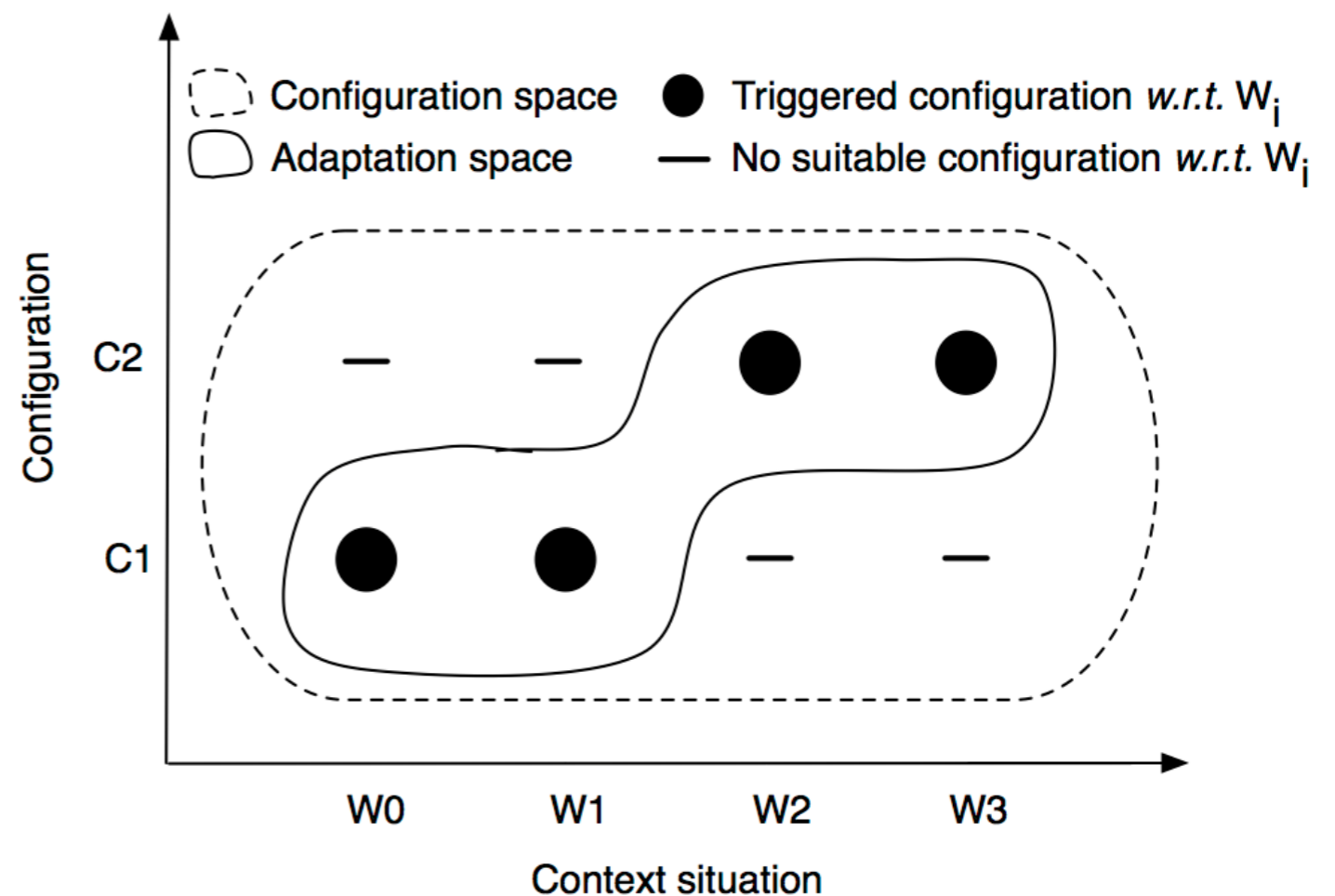
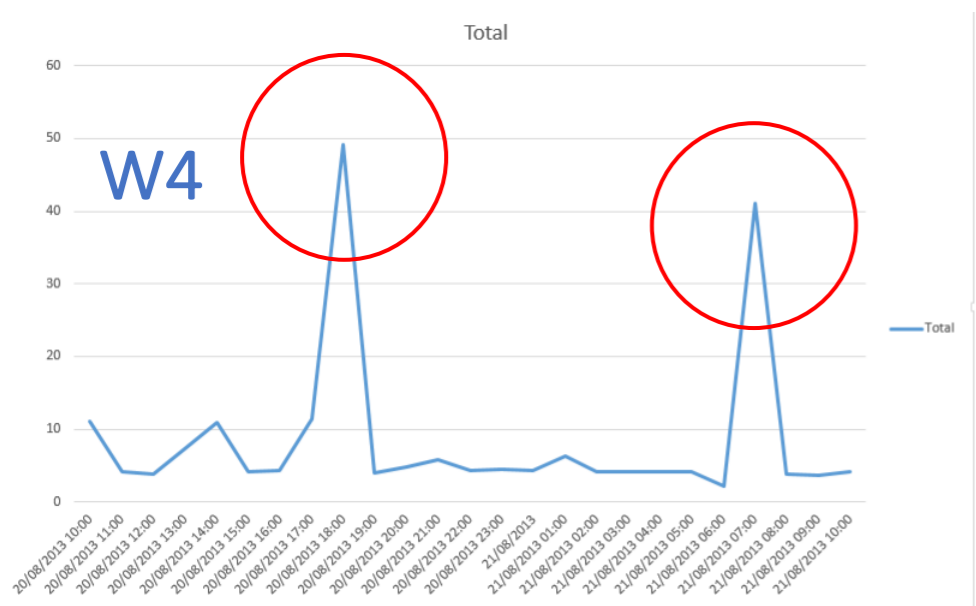


Anticipated
Workload W

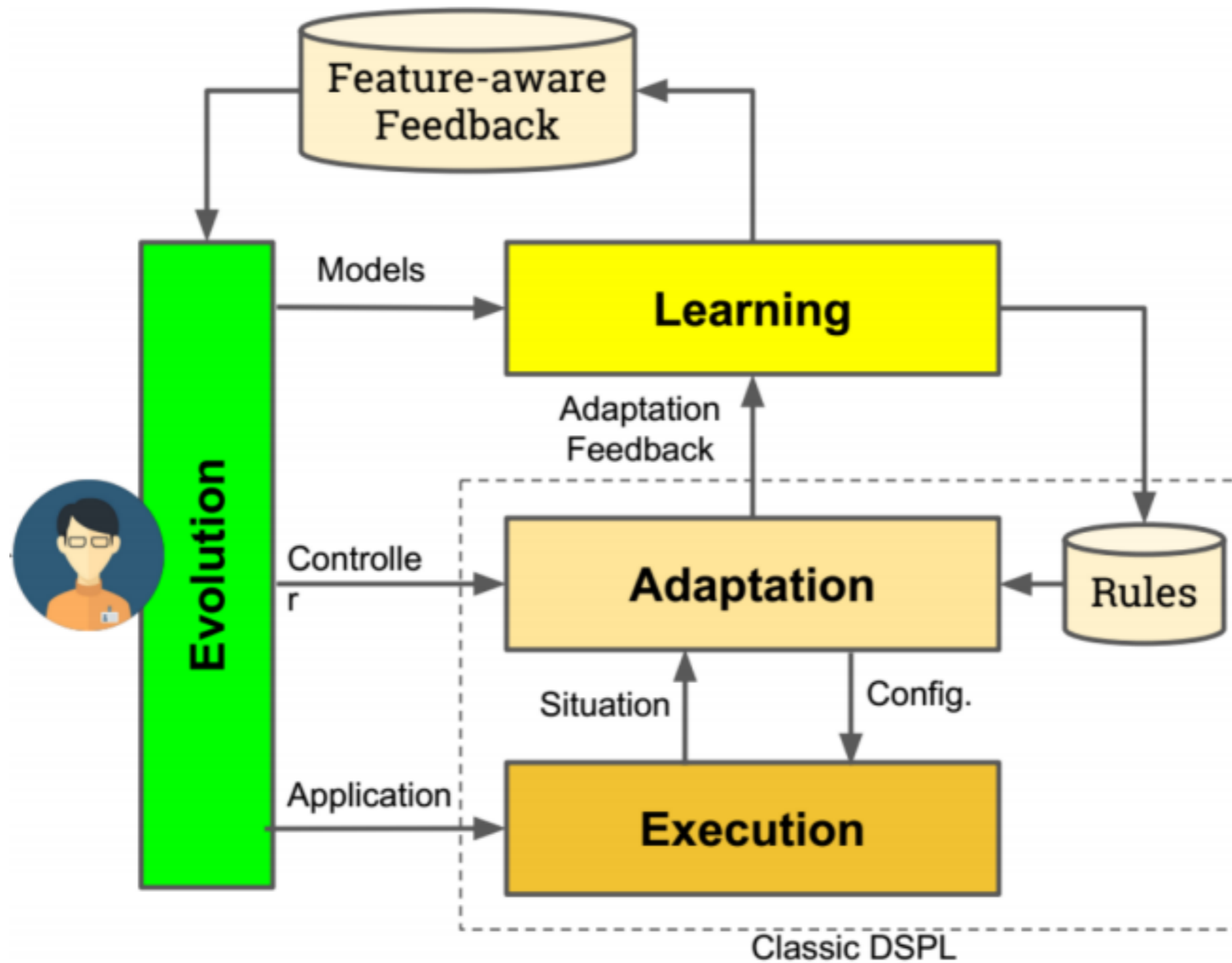


Limitations of DSPLs

- Fundamental assumption in DSPLs: Potential relevant context changes can be anticipated at design time (“known unknowns”)
- DSPLs limited in the presence of design time uncertainty!
- E.g., unanticipated context situations



Evolution & Learning For feature-oriented Adaptive systems



Questions?

luciano.baresi@polimi.it