TECHNISCHE UNIVERSITÄT DRESDEN

Europa fördert Sachsen. ESF Europäischer Sozialfonds SACHSEN

Faculty of Computer Science, Institute of Software and Multimedia Technology, Chair of Software Technology

Dynamic Configuration Management of Cloud-based Applications

Julia Schroeter, Peter Mucha, Marcel Muth, Kay Jugel, Malte Lochau

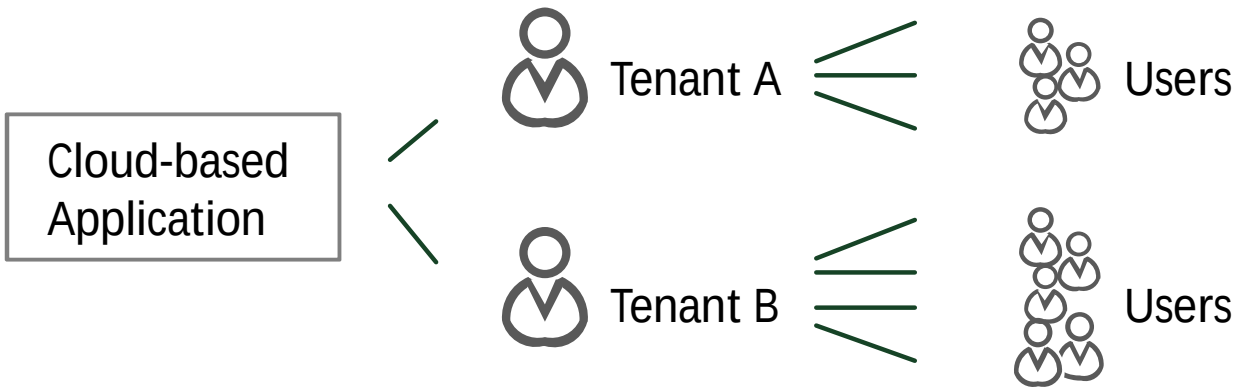
20.09.2012

Outline

1. Motivation
2. Characteristics of cloud-based applications
3. Dynamic configuration management
4. Conclusion and outlook

Motivation

- Cloud-based applications are often multi-tenant aware

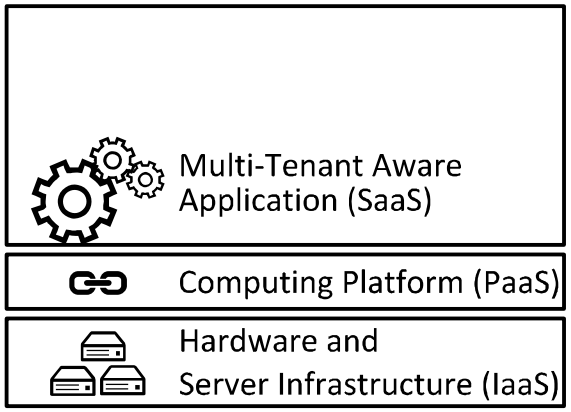


- Single-instance multi-tenancy
  - One application instance shared among various tenants
  - „One-size-fits-all“ paradigm

One size fits all?

- Tenants have different requirements
- Requirements may change
- Various stakeholders involved in provisioning application
- Need for efficient dynamic configuration management

Cloud-computing Stack [MG11]



- Software as a Service (SaaS)
  - Business application
- Platform as a Service (PaaS)
  - Multi-tenancy support
  - Load balancing
  - Persistence service
- Infrastructure as a Service (IaaS)
  - Storage
  - Computing capacity

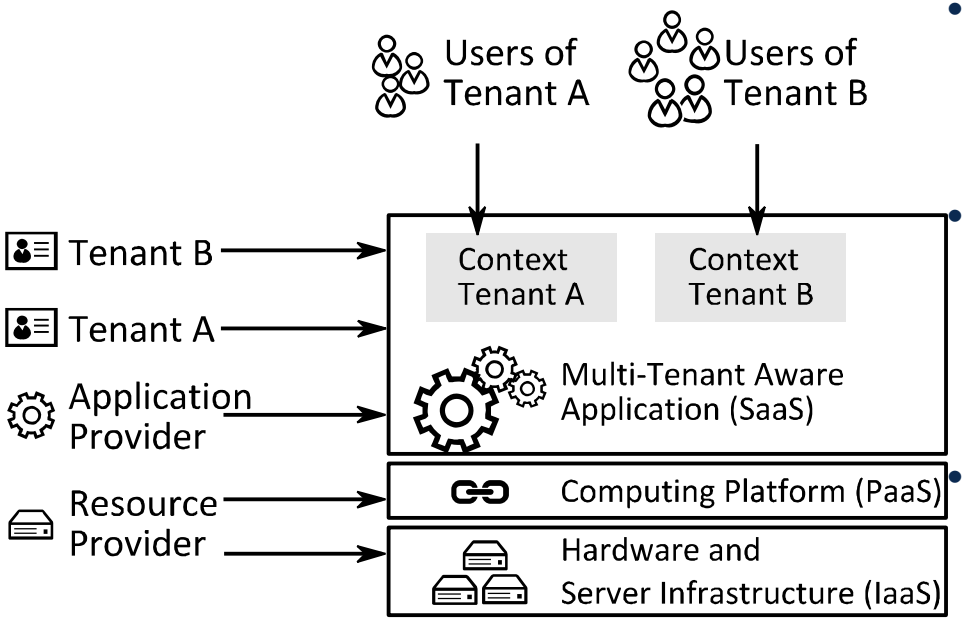
[MG11] *P. Mell and T. Grance*. The NIST definition of cloud computing. NIST Special Publication 800-145, National Institute of Standards and Technology, Information Technology Laboratory, 2011.

Case Study

- Indenica project [1]
- Create a virtual service platform (VSP)
- Create SaaS applications on top of VSP
- Various stakeholders involved

[1] www.indenica.eu

Stakeholders Involved in Configuration Process



- User
  - Various devices to access application
- Tenant
  - Functional requirements
  - Quality requirements (service level agreements)
- Application Provider
  - Provide application functionality
  - Platform pre-configuration
- Resource Provider
  - Infrastructure, platform pre-configuration

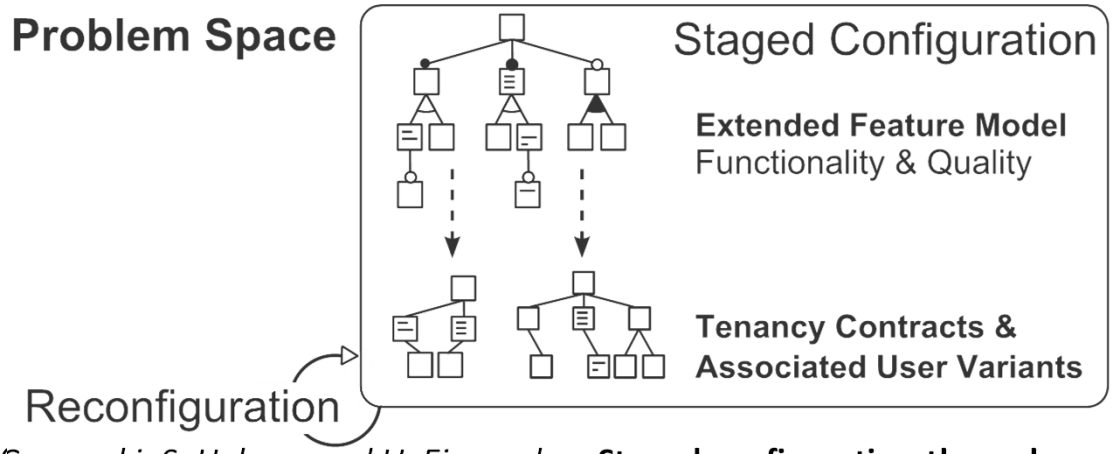
Characteristics of Cloud-based Applications

- Multi-tenancy aware application architecture
- Sharing of resources as well as the application instance
- Variable functionality and extra-functional qualities
- Runtime onboarding and decomissioning of tenants
- Not all tenants are known beforehand
- Various stakeholders involved in configuration process
- Change of a stakeholder‘s objectives

Dynamic Configuration Management

- Apply software product line (SPL) variability management

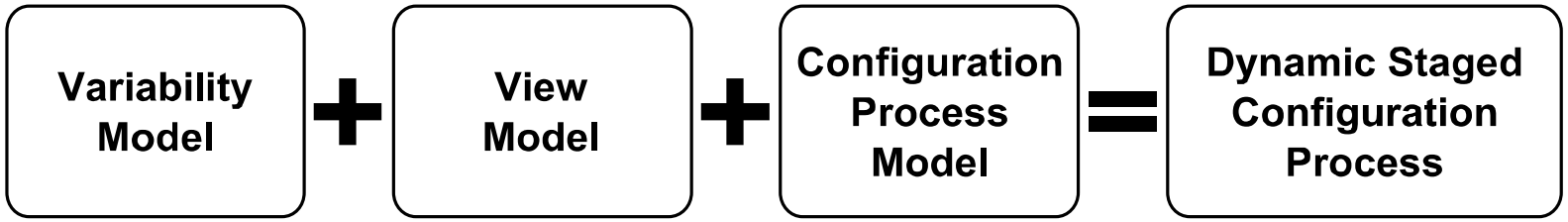
- Extend staged configuration proposed by Czarnecki et al.[CHE05]
  - Pre-configuration stages
  - Adding stakeholders at runtime to a stage
  - Reconfiguration on defined entry points



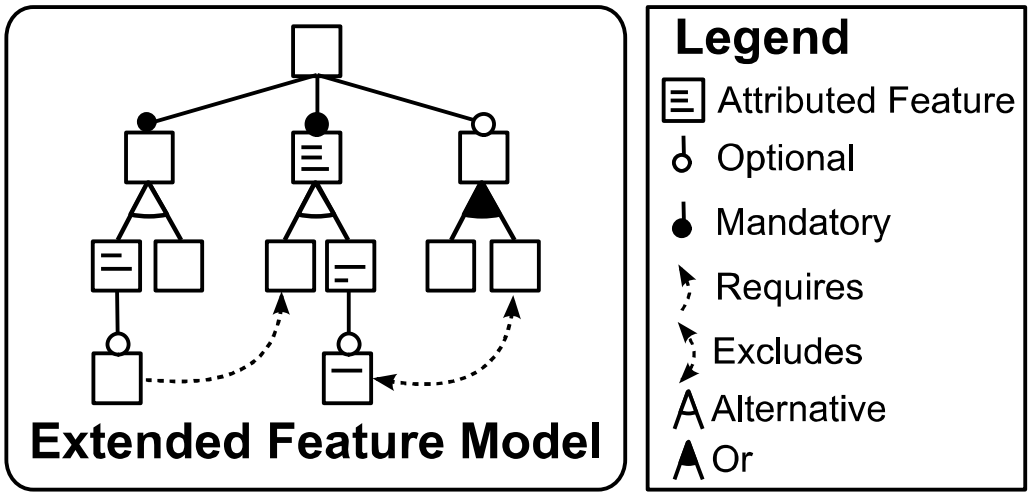
[CHE05] *K. Czarnecki, S. Helsen, and U. Eisenecker*. **Staged configuration through specialization and multi-level configuration of feature models.** Journal of Software Process: Improvement and Practice, 10(2):143–169, 2005

Dynamic Staged Configuration

- Model staged configuration process
- Separation of concerns
- Enable reuse



Variability Model



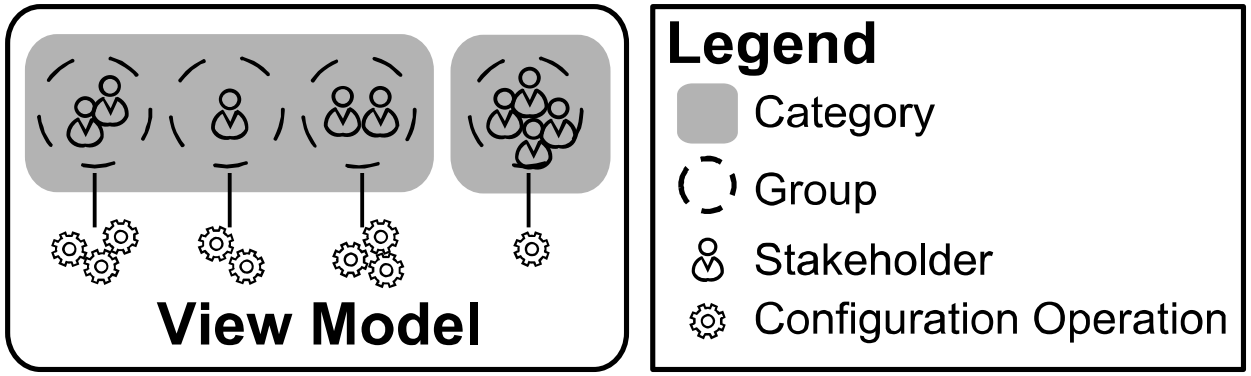
- Functionality represented as features
- Quality properties represented as attributes
- Cross-tree constraints among features and attributes

Configuration Operations on the Variability Model

- Atomic operations
  - Select feature
  - Deselect feature
  - Set attribute value
  - Limit an attribute domain

- Complex operations
  - Composed by multiple atomic operations

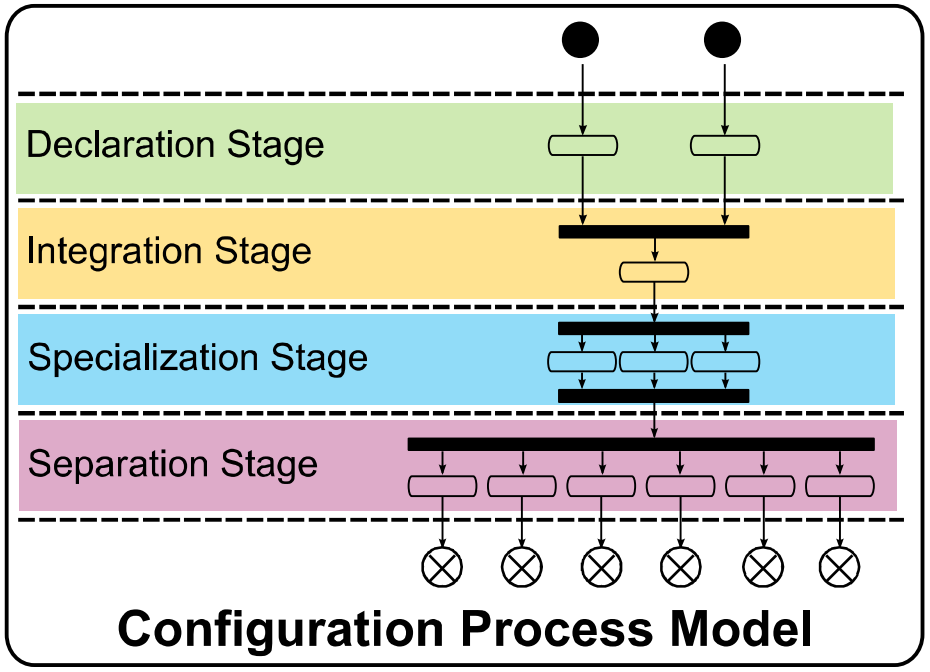
View Model



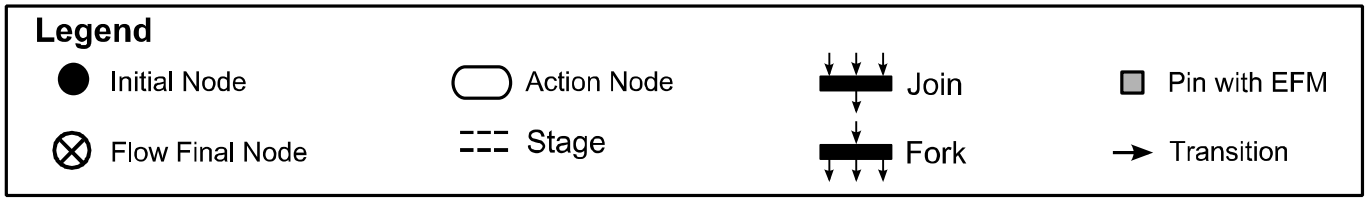
- Group stakeholders
- Assign configuration operations to (groups of) stakeholders
- Apply concepts of role based access control (RBAC) [FK92]

[FK92] *D. Ferraiolo and D. Kuhn*. Role based access control. Proceedings of the 15th National Computer Security Conference (NCSC '92), pages 554–563, 1992.

Configuration Process Model

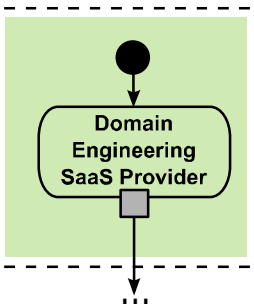


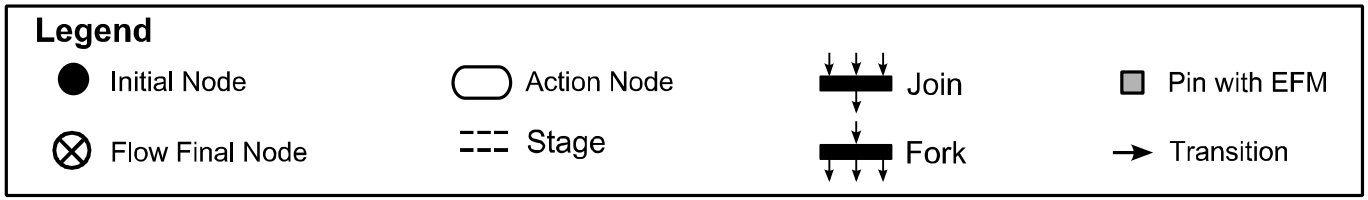
- Different stage types
- Free ordering of stage types
- Result of stages are pre-configurations with left variability
- Final results are complete configurations



Example for a Declaration Stage

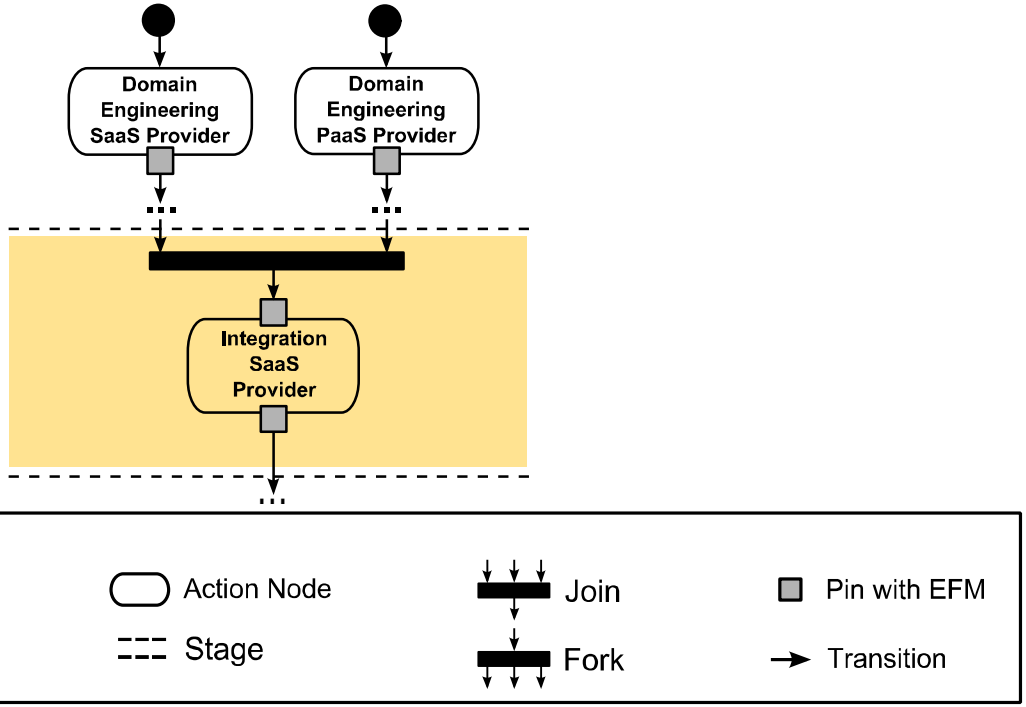
- Application declaration stage
  - Define application feature model
  - Done by application provider
  - A single feature model as output





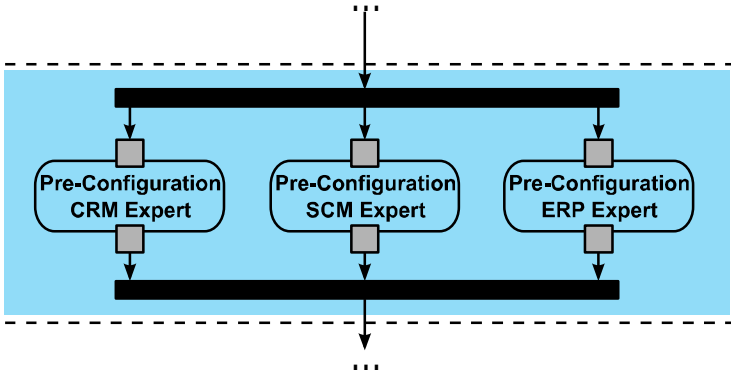
Example for an Integration Stage

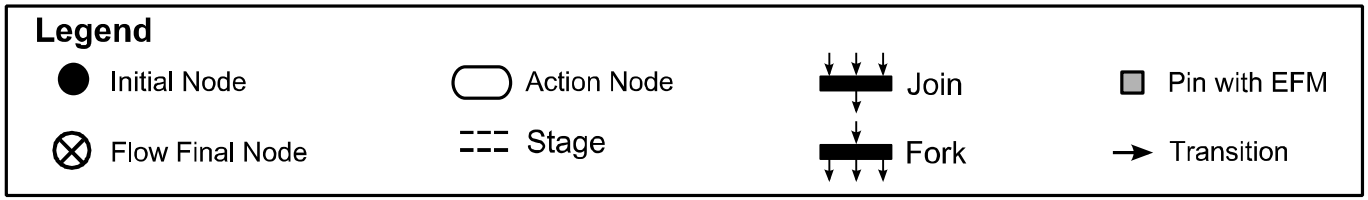
- Platform and application integration stage
  - Multiple feature models as input
  - One merged feature model as output



Example for a Specialization Stage

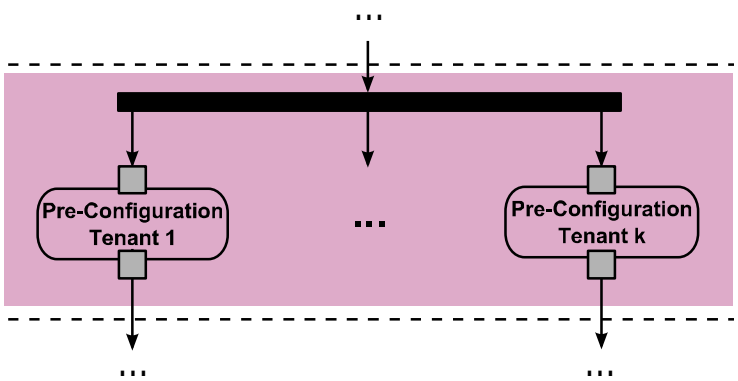
- Application specialization stage
  - One feature model as input
  - One feature model as output
  - Multiple configuration actions are performed in parallel and merged

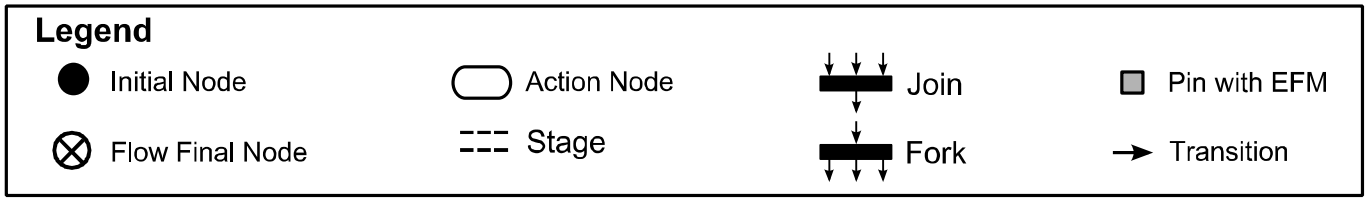




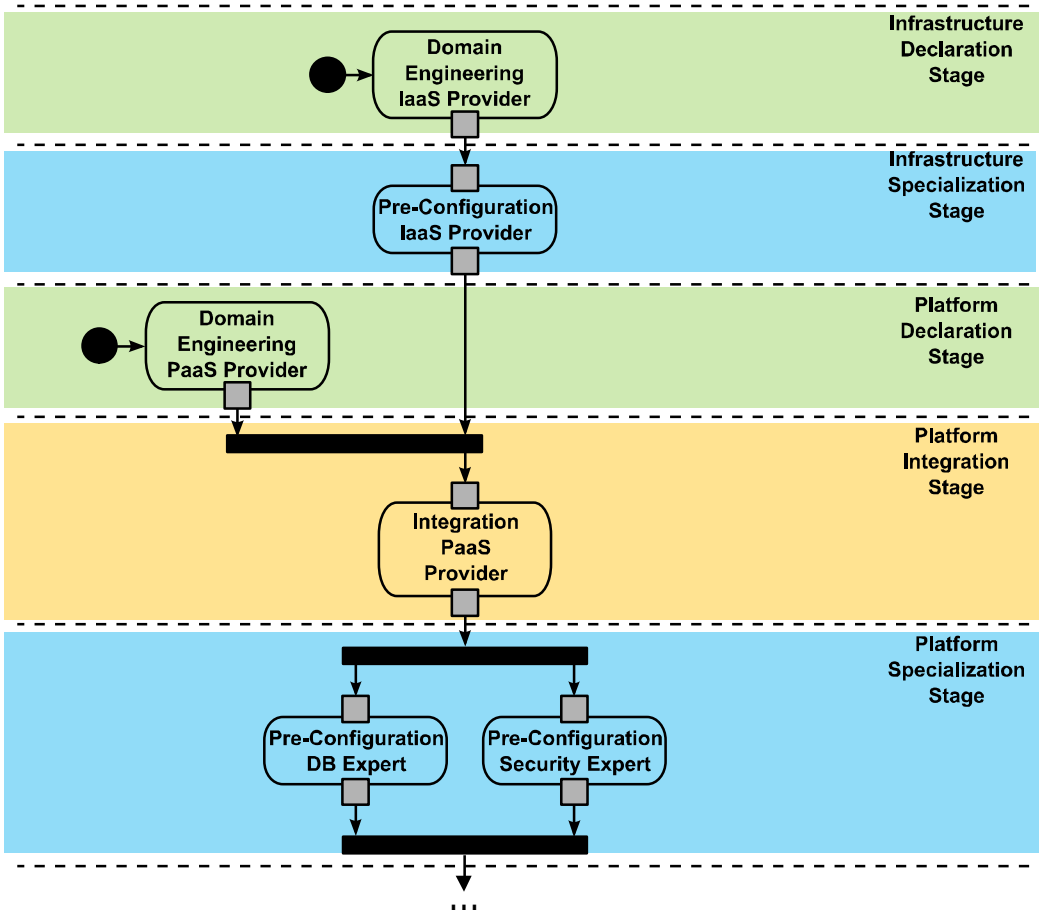
Example for a Separation Stage

- Tenant separation stage
  - One feature model as input
  - Multiple configuration actions are independently performed in parallel
  - Multiple independent feature models as output

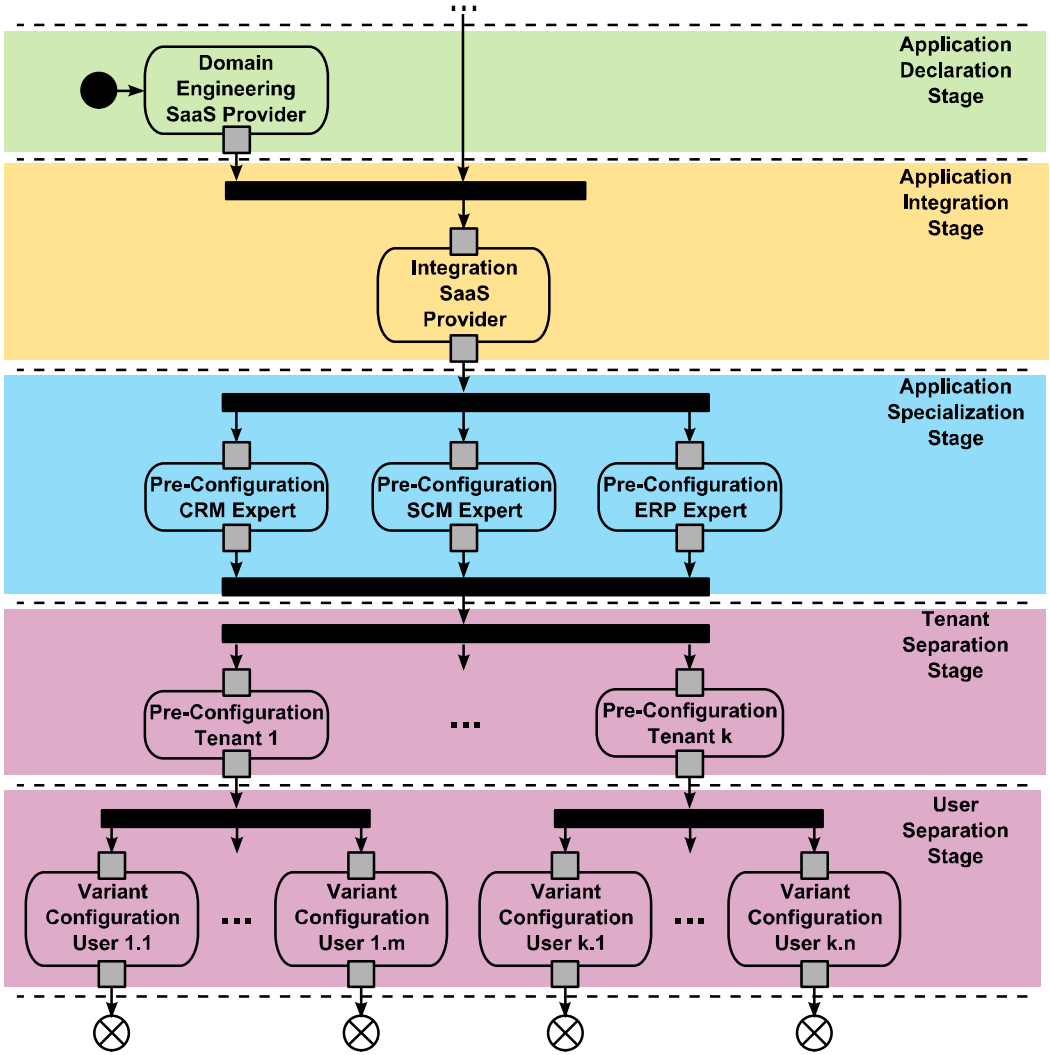




Case Study – Configuration Process Model

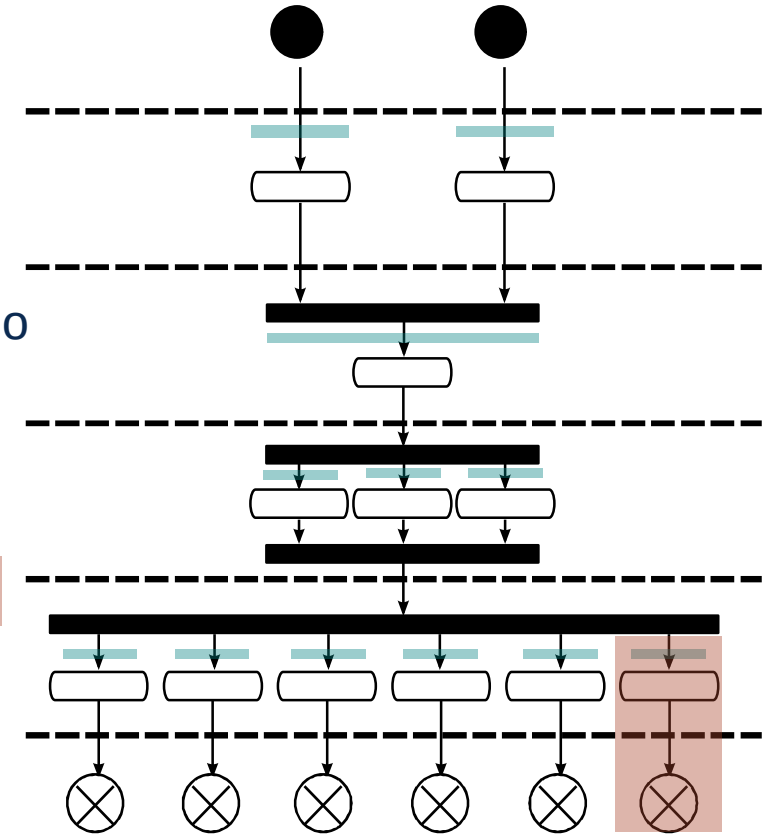


Case Study – Configuration Process Model



Dynamic Concepts

- Reconfiguration
  - On defined entry points
  - Changes are propagated to subsequent stages
- Add / remove stakeholder
  - At application runtime
  - Update view model
  - Add / remove actions in the configuration process



Conclusion and Outlook

- Support variability in cloud-based applications
- Staged configuration with various stakeholders involved
- Reconfiguration support to handle changing objectives
- Add and remove stakeholders dynamically

- Implementation of configuration management and the staged configuration process
- Tooling support for extended feature models needed
- Evaluation using different case studies

Thank you for attending. Any questions?

Contact

Julia Schroeter
Software Technology Group, TU Dresden
julia.schroeter@tu-dresden.de
www.juliaschroeter.de