Software Product Lines for Automatic Multi-Cloud Configuration

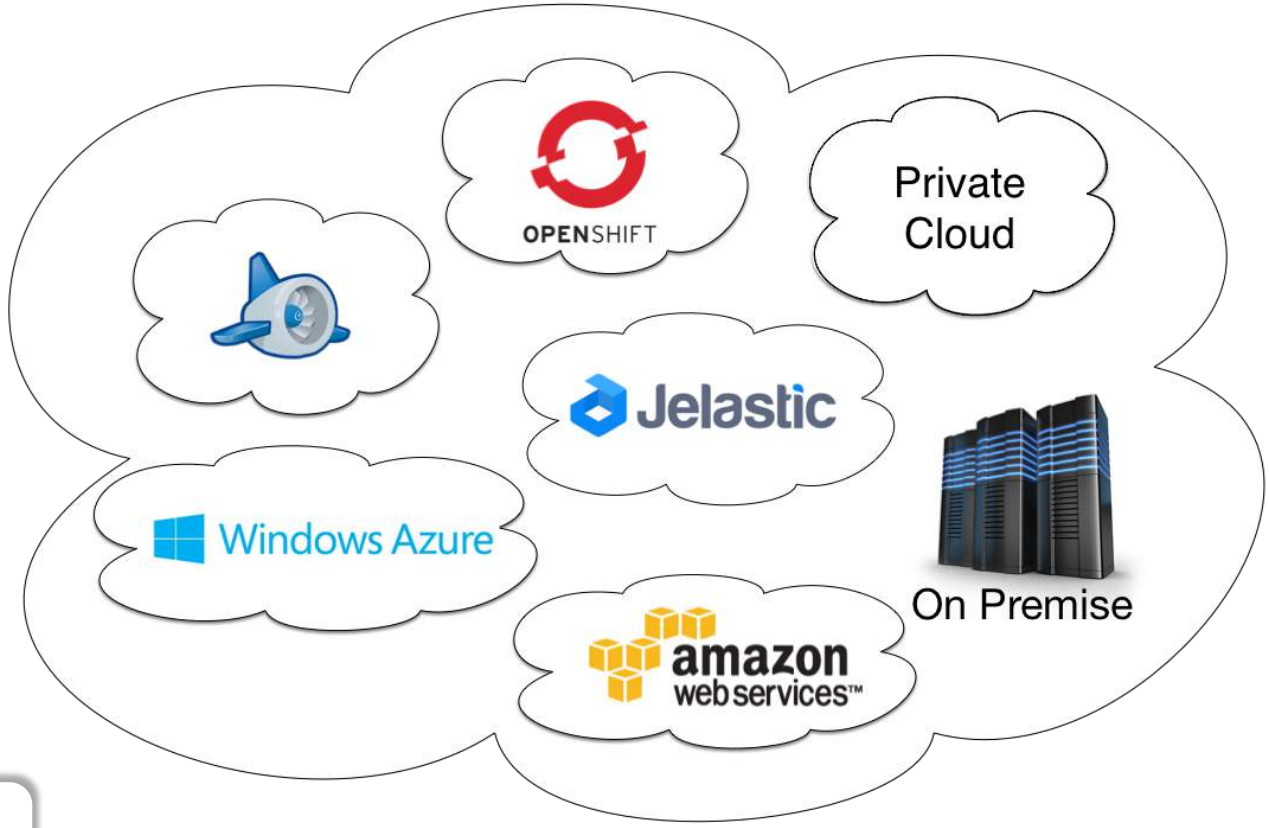
Gustavo Sousa
gustavo.sousa@inria.fr

Encadrants:
Walter Rudametkin
Laurence Duchien

Université Lille 1
CRIStAL UMR CNRS 9189
Inria Lille - Nord Europe
France

What is Multi-Cloud?

- Use of multiple, independent cloud providers
- Hybrid multi-cloud can involve **on-premise** infrastructure, **private** and **public clouds**

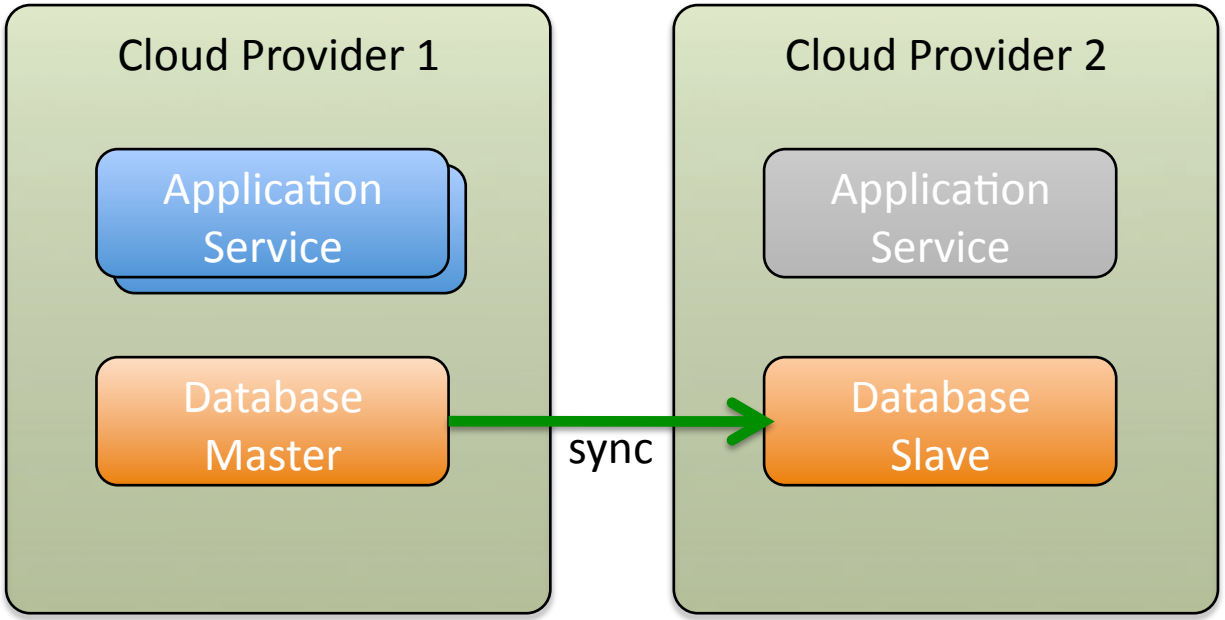


Why Multi-Cloud?

- Improve resilience
  - High Availability
  - Disaster Recovery
- Optimize costs and service quality
- Leverage cloud-specific features
- Leverage existing infrastructure
- Avoid vendor lock-in
- Reduce latency
- Comply with data regulation

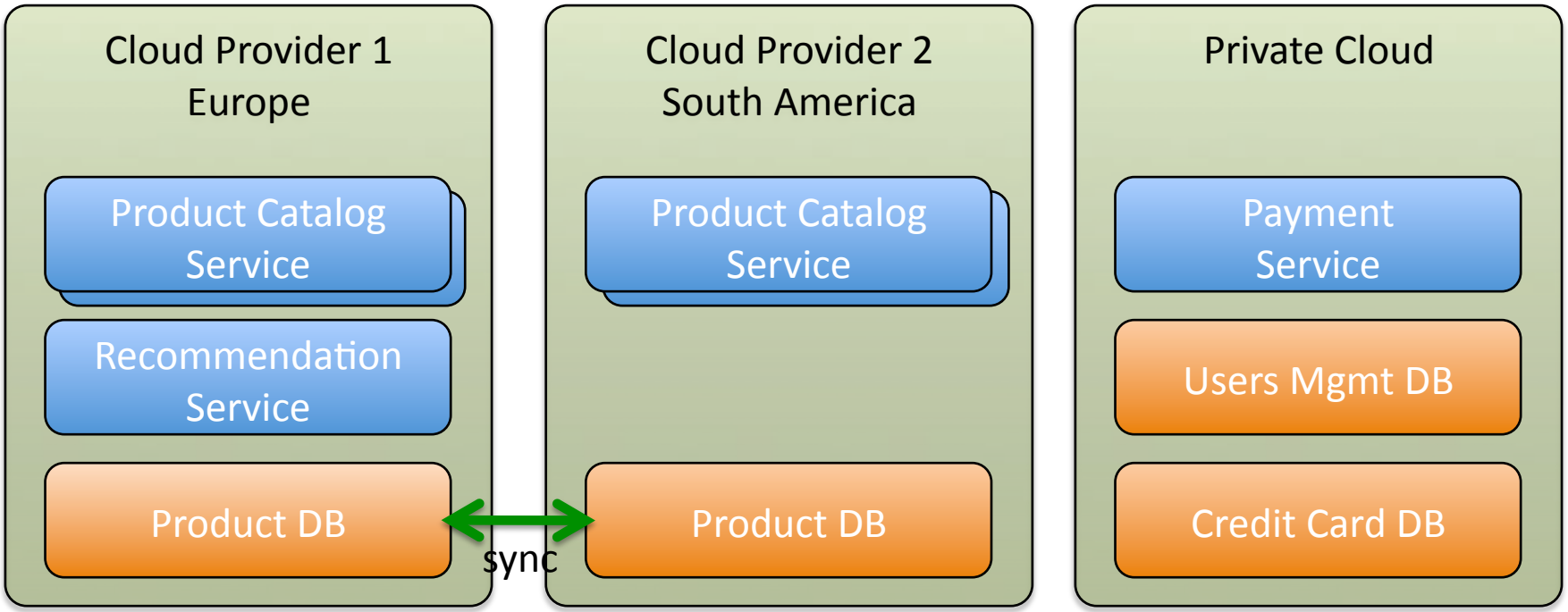
Scenario 1

- Avoid lock-in and improve resilience



Scenario 2

- Comply with data regulation and reduce latency



Multi-cloud complexity

- Introduces another dimension of complexity

- Multi-cloud systems deal with several aspects of the **application** and their **environment**
  - Provider selection
  - Resource provisioning
  - Application deployment
  - Monitoring
  - Self-adaptation
  - Migration
  - …

Multi-cloud complexity

- Introduces another dimension of complexity

- Multi-cloud systems deal with several aspects of the **application** and their **environment**
  - Provider selection
  - Resource provisioning
  - Application deployment
  - Monitoring
  - Self-adaptation
  - Migration
  - …

Challenges for Multi-Cloud Configuration

- Cloud provider heterogeneity
  - Service models (IaaS, PaaS, SaaS, DBaaS)
    - Each provider can have their own service model
  - Available features (monitoring, scaling, software packages, etc)
    - Equivalent features with different properties, measuring units, etc
  - Management APIs
  - Pricing model
  - Service level
    - Service quality is described in different terms, with different warranties

Challenges for Multi-Cloud Configuration

- Cloud provider heterogeneity
  - Service models (IaaS, PaaS, SaaS, DBaaS)
    - Each provider can have their own service model
  - Available features (monitoring, scaling, software packages, etc)
    - Equivalent features with different properties, measuring units, etc
  - Management APIs
  - Pricing model
  - Service level
    - Service quality is described in different terms, with different warranties
- Difficult to compare
  - Among themselves
  - Against application requirements

State of the art

- Many research projects, open source and commercial tools
  - mOSAIC, MODAClouds, PaaSage, soCloud, Aoleus, Optimis, Cloud4SOA, StratusLab, Tclouds
  - jclouds, δ-cloud and SimpleCloud
  - RightScale, Kaavo, Agility

Petcu, Dana. "Consuming resources and services from multiple clouds." Journal of Grid Computing 12.2 (2014): 321-345.

State of the art

- Propose a PaaS or middleware
  - Development should follow a pre-defined platform, component model or API
  - mOSAIC

    Di Martino, Beniamino, et al. "Building a mosaic of clouds." Euro-Par 2010 Parallel Processing Workshops. Springer Berlin Heidelberg, 2011.
  - soCloud

    Paraiso, Fawaz, et al. "A federated multi-cloud PaaS infrastructure." Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on. IEEE, 2012
  - Tclouds

    Verissimo, Paulo, Alysson Bessani, and Marcelo Pasin. "The TClouds architecture: Open and resilient cloud-of-clouds computing." Dependable Systems and Networks Workshops (DSN-W), 2012 IEEE/IFIP 42nd International Conference on. IEEE, 2012
  - Optimis

    Ferrer, Ana Juan, et al. "OPTIMIS: A holistic approach to cloud service provisioning." Future Generation Computer Systems 28.1 (2012): 66-77

State of the art

- Deal only with IaaS
  - jclouds, δ-cloud and SimpleCloud
  - RightScale, Kaavo, Agility
  - Aoleus

    Catan, Michel, et al. "Aeolus: Mastering the complexity of cloud application deployment." Service-Oriented and Cloud Computing. Springer Berlin Heidelberg, 2013. 1-3.

State of the art

- Do not support provider selection
  - Considering location constraints, redundancy and scalability rules
  - MODAClouds

    Ardagna, Danilo, et al. "Modaclouds: A model-driven approach for the design and execution of applications on multiple clouds." Proceedings of the 4th International Workshop on Modeling in Software Engineerin.
  - PaaSage

    Jeffery, Keith, Geir Horn, and Lutz Schubert. "A vision for better cloud applications." Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds.

State of the art

- Propose a PaaS or middleware
- Deal only with IaaS providers
- No support for provider selection

Requirements for Multi-Cloud Configuration Support

- Should work with existing frameworks
  - No need to use an specific API, middleware, component-model or PaaS implementation

Requirements for Multi-Cloud Configuration Support

- Should work with existing frameworks

- Users can specify the application's needs
  - Independent of cloud specific concepts
  - In terms of **required features**, **scalability** rules, **redundancy** and **location** constraints

Requirements for Multi-Cloud Configuration Support

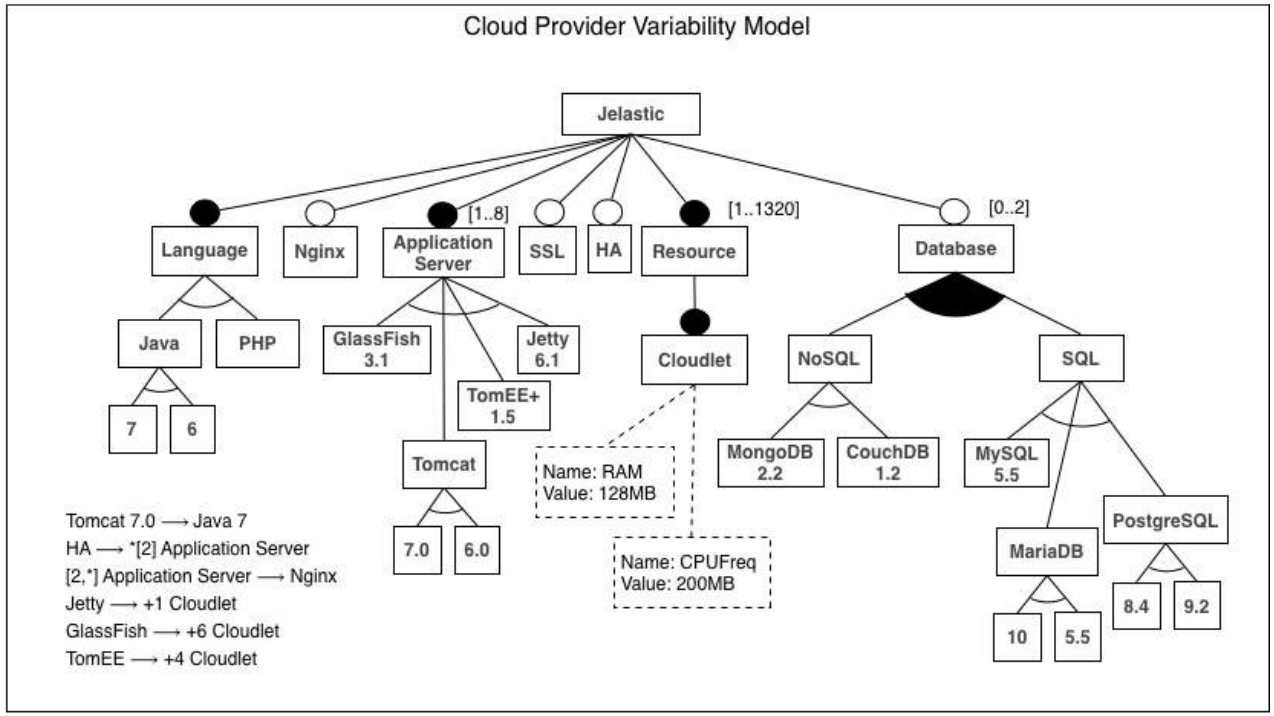
- Enable users to use any application development tools
- Users can specify the application's needs
- Automatic configuration
  - Select and configure cloud providers that comply with the set of requirements
  - Optimizing for costs

SALOON

- Uses **Software Product Line Engineering** principles to generate application configurations for a cloud provider
- SPLE is used to achieve software reuse across similar products
  - Feature Models describe variability
- SALOON uses Feature Models to describe variability between configurations **for a given cloud provider**
  - A valid cloud configuration is considered a product

SALOON Approach

- Cloud Provider configuration options are described as a **Feature Model** extended with **cardinalities** and **attributes**



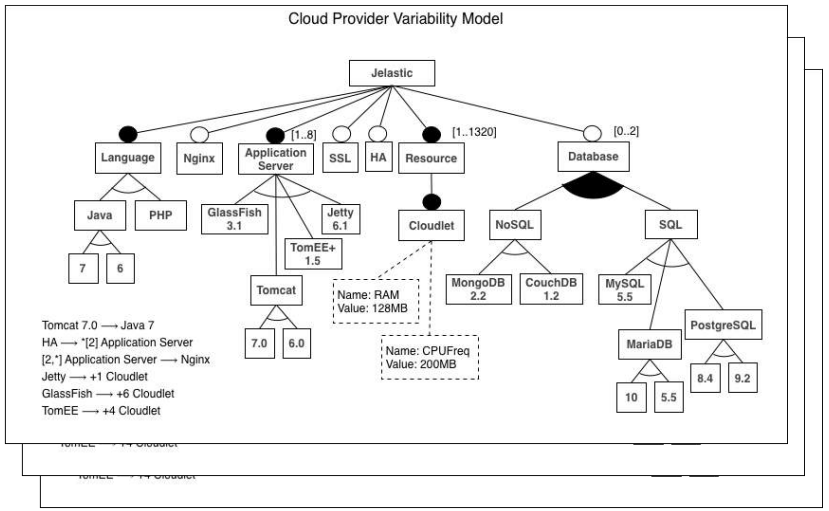
Quinton, Clément, Daniel Romero, and Laurence Duchien. "Automated Selection and Configuration of Cloud Environments Using Software Product Lines Principles." *IEEE CLOUD 2014.*

SALOON Approach

- Application requirements describe required features and their attributes

Application Requirements

Java
2 Jetty
1 Glassfish [CPU: 2GHz, RAM: 4GB]
MySQL [dbSize: 2GB]
MongoDB [dbSize: 10GB]



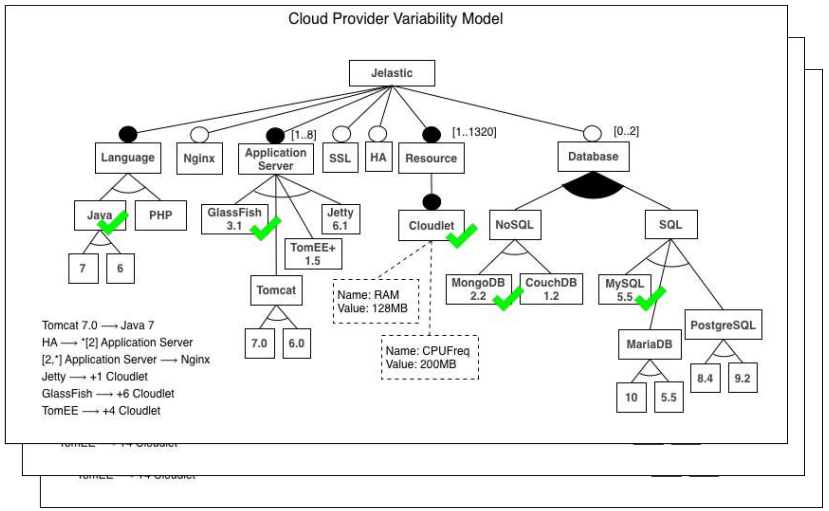
Quinton, Clément, Daniel Romero, and Laurence Duchien. "Automated Selection and Configuration of Cloud Environments Using Software Product Lines Principles." *IEEE CLOUD 2014.*

SALOON Approach

- Application requirements are mapped to features in each Cloud Provider Feature Model (product configuration)

Application Requirements

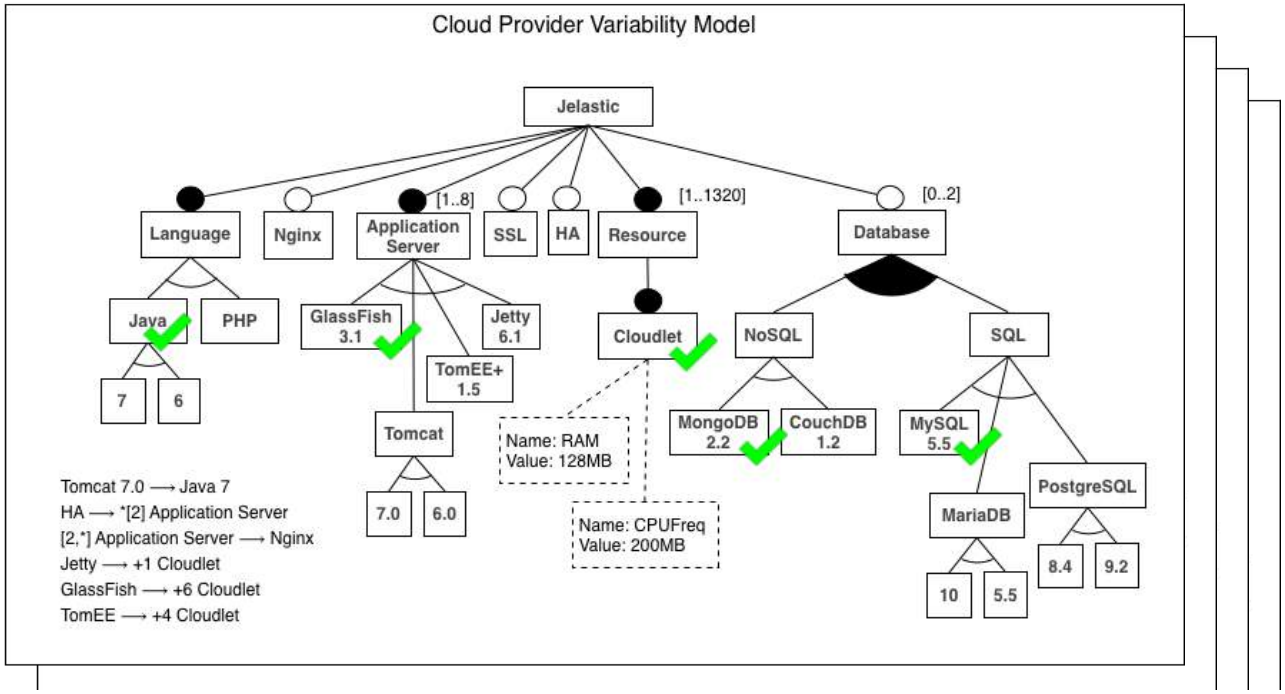
Java
2 Jetty
1 Glassfish [CPU: 2GHz, RAM: 4GB]
MySQL [dbSize: 2GB]
MongoDB [dbSize: 10GB]



Quinton, Clément, Daniel Romero, and Laurence Duchien. "Automated Selection and Configuration of Cloud Environments Using Software Product Lines Principles." *IEEE CLOUD 2014.*

SALOON Approach

- Product configurations are evaluated
- A list of valid configurations is obtained



Quinton, Clément, Daniel Romero, and Laurence Duchien. "Automated Selection and Configuration of Cloud Environments Using Software Product Lines Principles." *IEEE CLOUD 2014.*

SALOON limitations for Multi-Cloud

- Application requirements are described only in terms of required features, cardinalities and attributes
  - Does not consider multiple services and their relationship
- Requirements are evaluated against one cloud provider at a time
  - Does not consider using multiple clouds simultaneously
- Does not consider **redundancy**, **scalability** and **location** requirements
  - Motivation for multi-cloud

Research questions

GQ1. Can Software Product Lines and Search Based strategies be used to effectively and efficiently generate optimized multi-cloud configurations?

RQ1. How to describe an application's **multi-cloud environment requirements** independent of existing cloud providers?

intrinsic: required features and resources

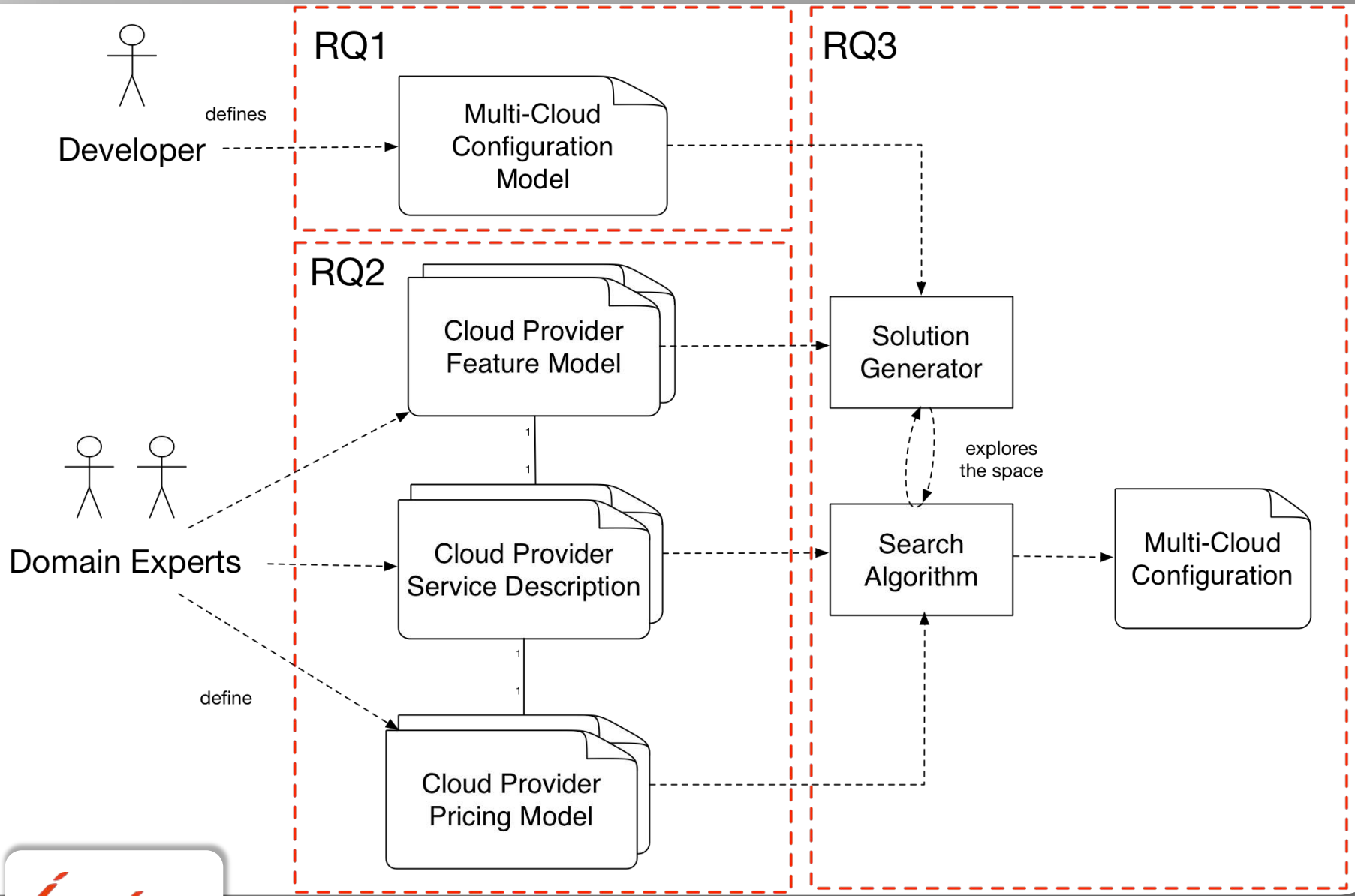
contextual: location constraints, scalability and redundancy rules

RQ2. How to abstract and describe the heterogeneity across different cloud providers to make them comparable?

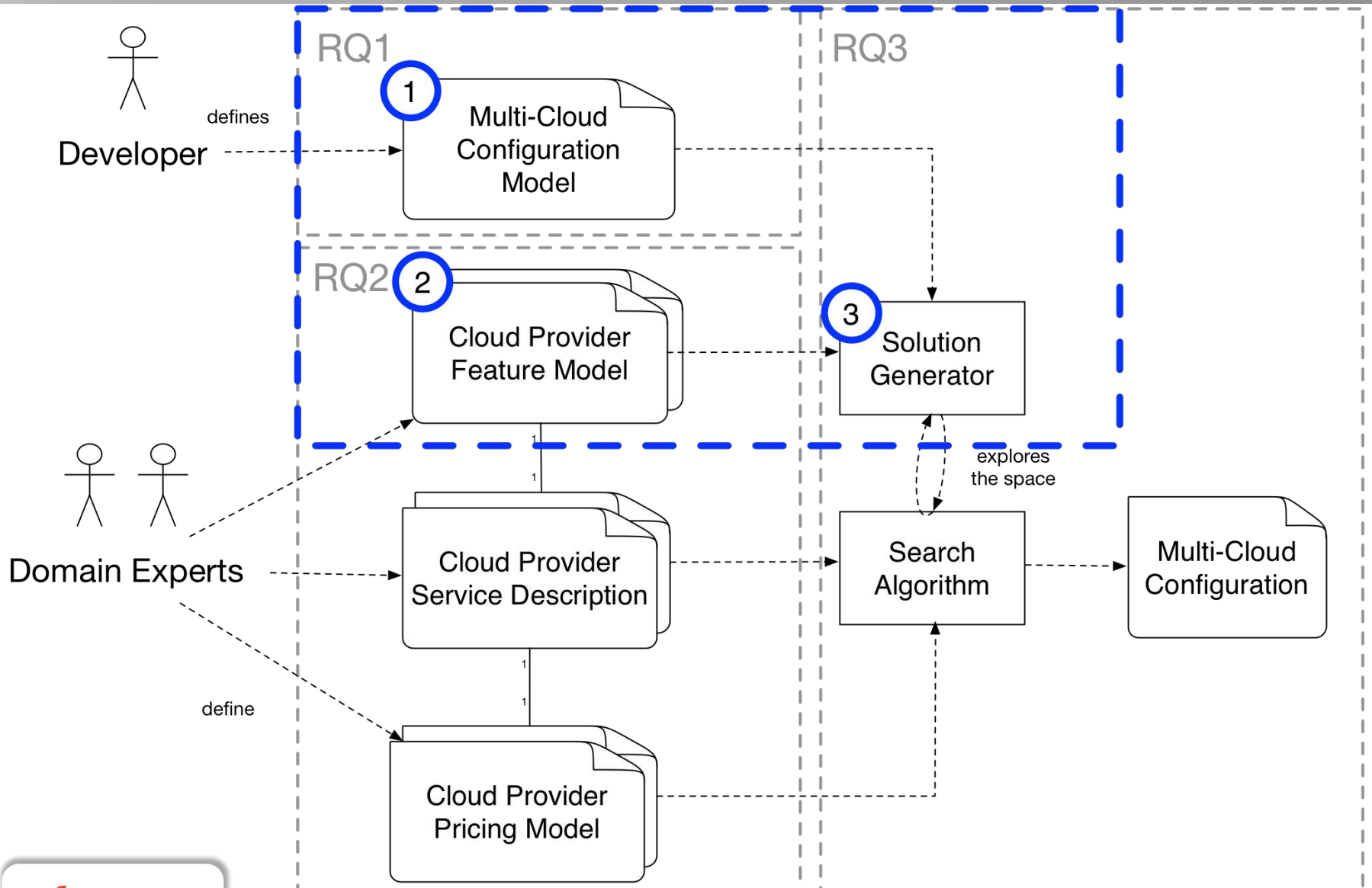
features, pricing model, service level

RQ3. How to build and optimize multi-cloud configurations?

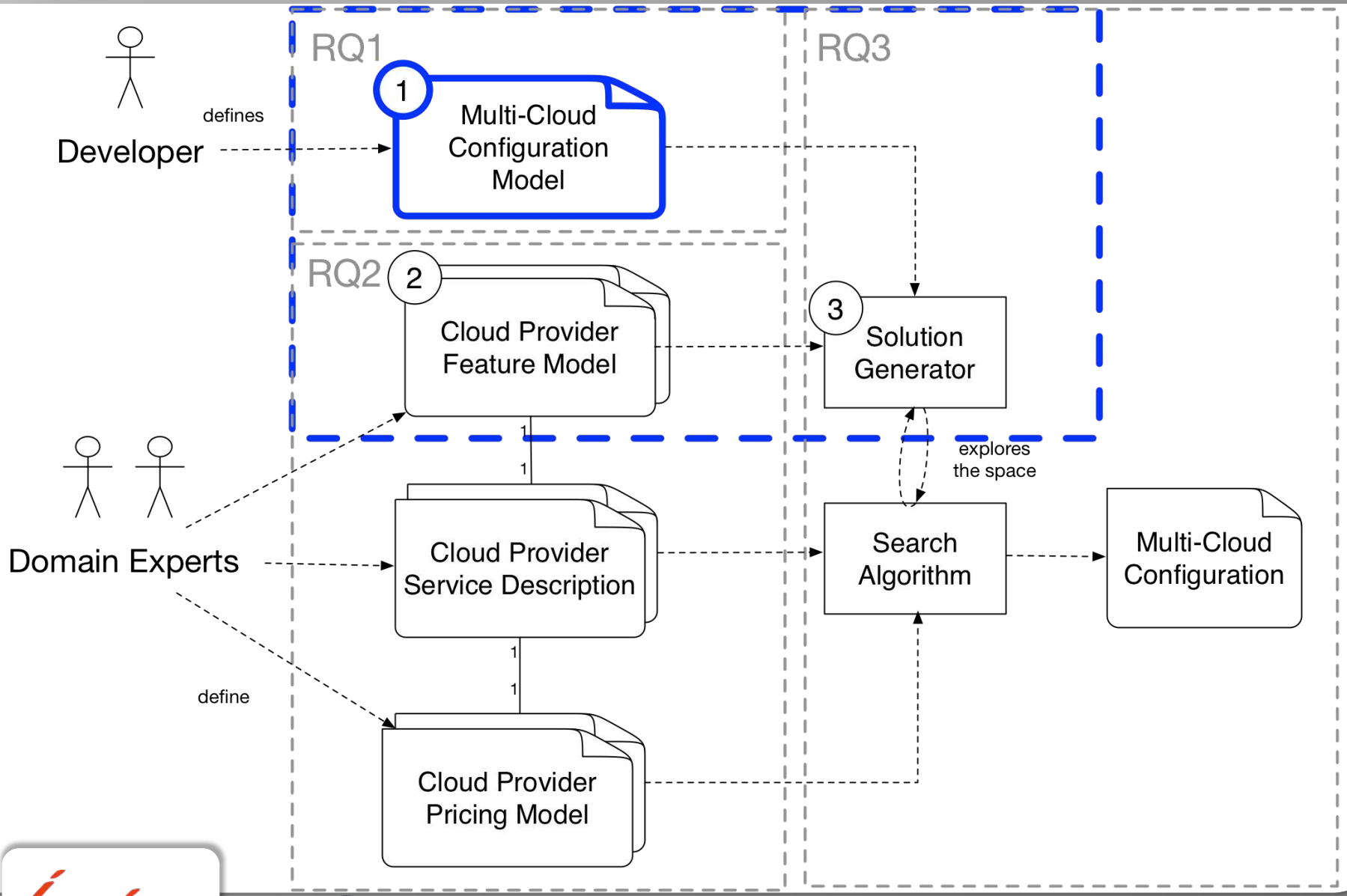
Conceptual overview



Topics


RQ1
RQ2
RQ3
1
Multi-Cloud Configuration Model
2
Cloud Provider Feature Model
3
Solution Generator
Developer
defines
Domain Experts
define
Cloud Provider Service Description
Cloud Provider Pricing Model
explores the space
Search Algorithm
Multi-Cloud Configuration
1
1
1
1

Multi-Cloud Configuration Model



Multi-Cloud Configuration Model

- Support multi-cloud motivation
  - Improve resilience
  - Optimize costs and service quality
  - Leverage cloud-specific features
  - Avoid vendor lock-in
  - …

Multi-Cloud Configuration Model

- Support multi-cloud motivation
- Support microservices architecture
  - Application composed of decoupled independent services

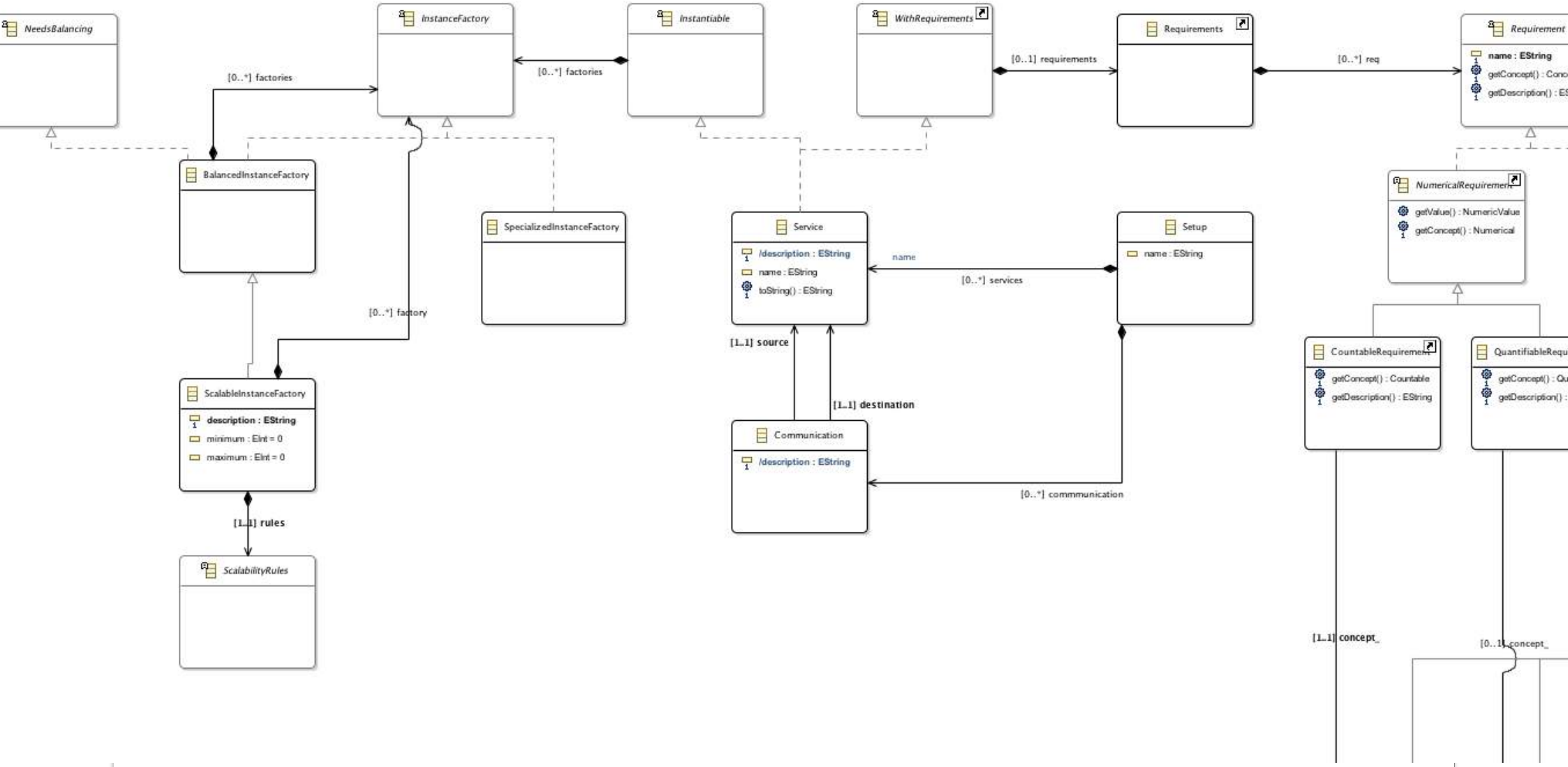
Multi-Cloud Configuration Model

- Support multi-cloud motivation
- Support microservices architecture
- Intrinsic requirements
  - Required features (frameworks, application servers, DBMS, etc)
  - Required resources (CPU, RAM, disk storage, networking)
  - Required communication

Multi-Cloud Configuration Model

- Support multi-cloud motivation
- Support microservices architecture
- Intrinsic requirements
- Contextual requirements
  - Redundancy
  - Location
  - Scalability

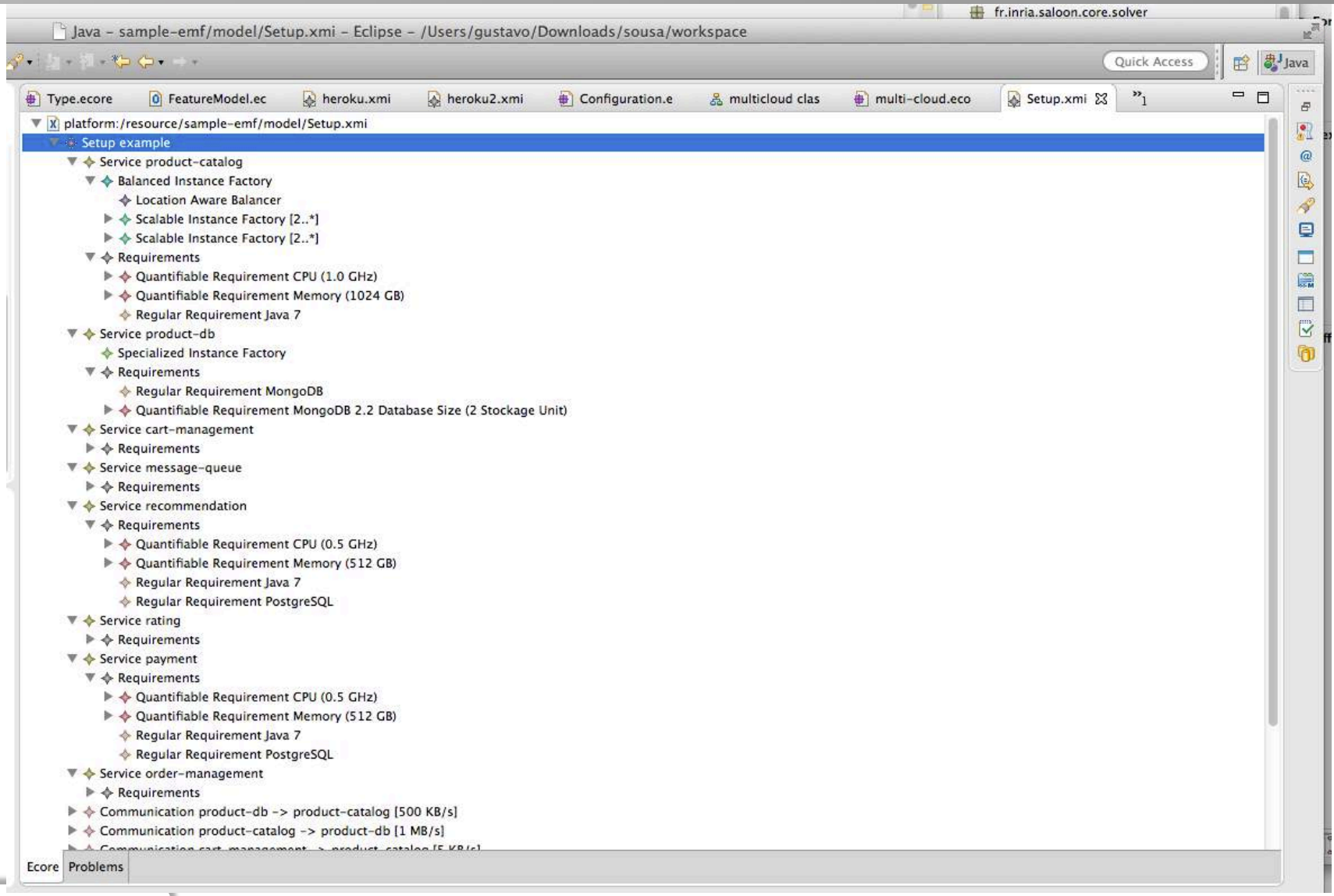
Proposed meta-model



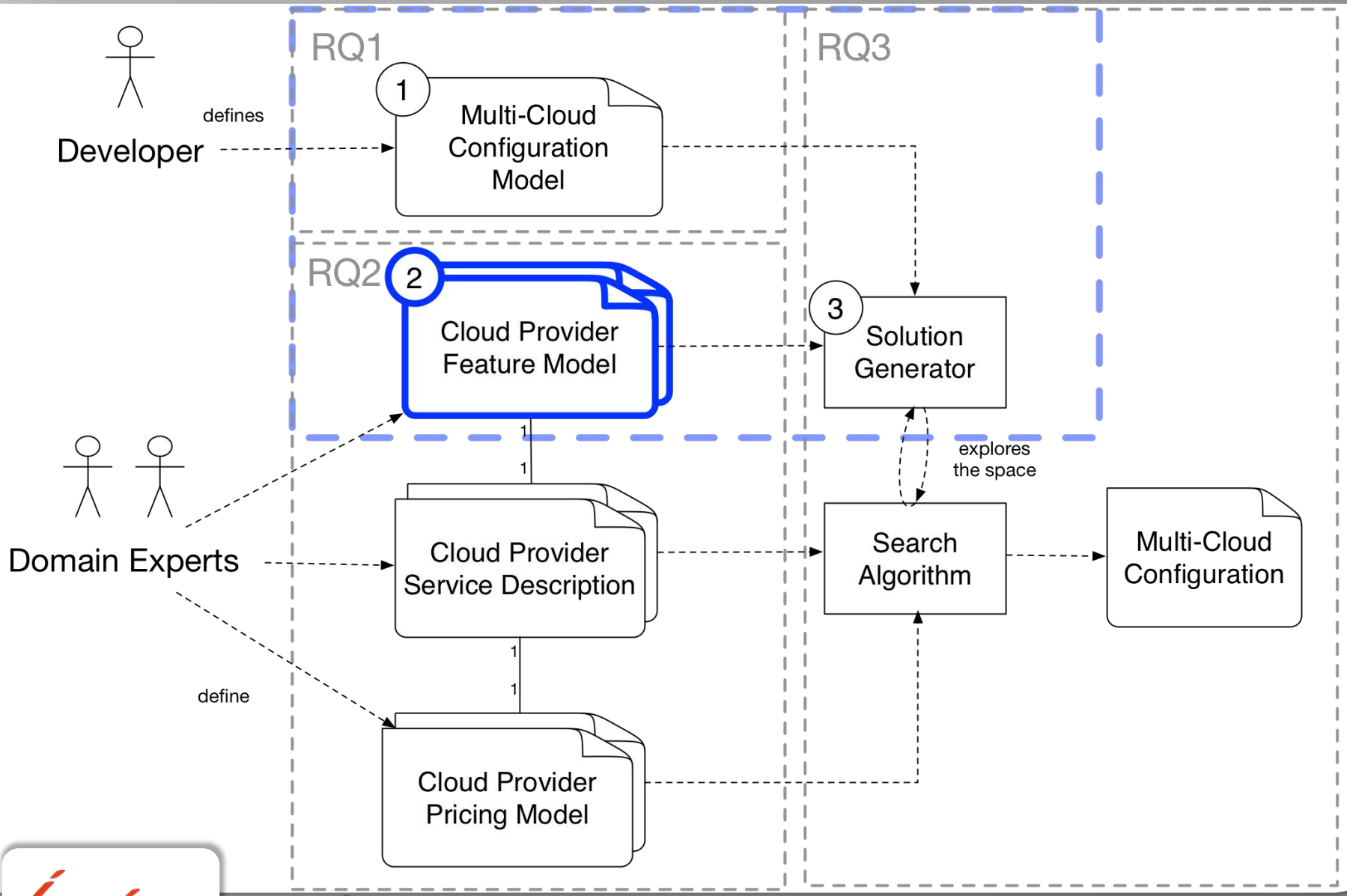
Multi-Cloud Configuration Model

- Current status
  - Initial meta-model designed
  - Intrinsic requirements mapped to SALOON ontology
  - Redundancy and location requirements
- Future work
  - Include scalability rules
  - Map requirements to FM configurations

Multi-Cloud Configuration Model



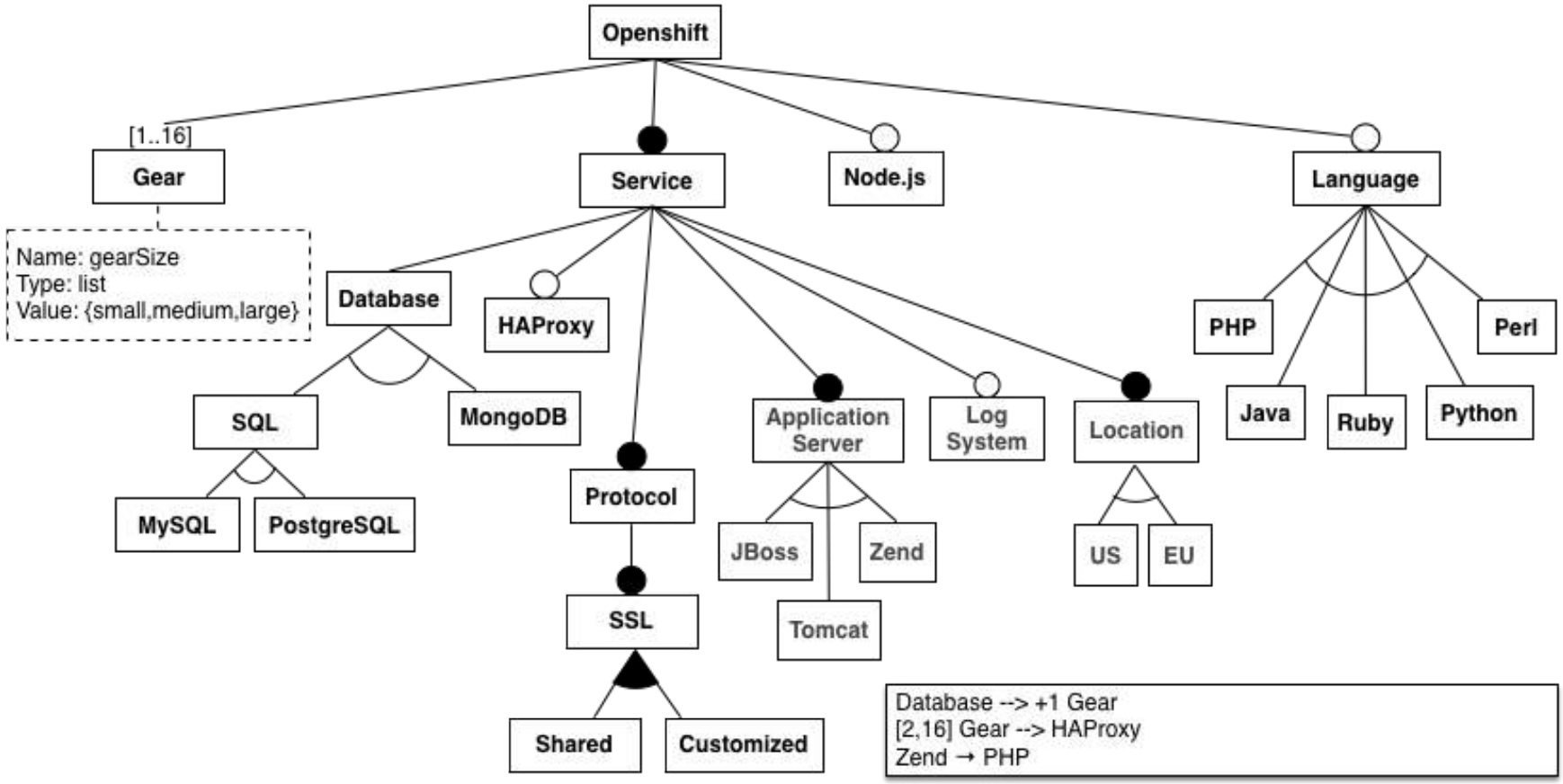
Cloud Provider Feature Model



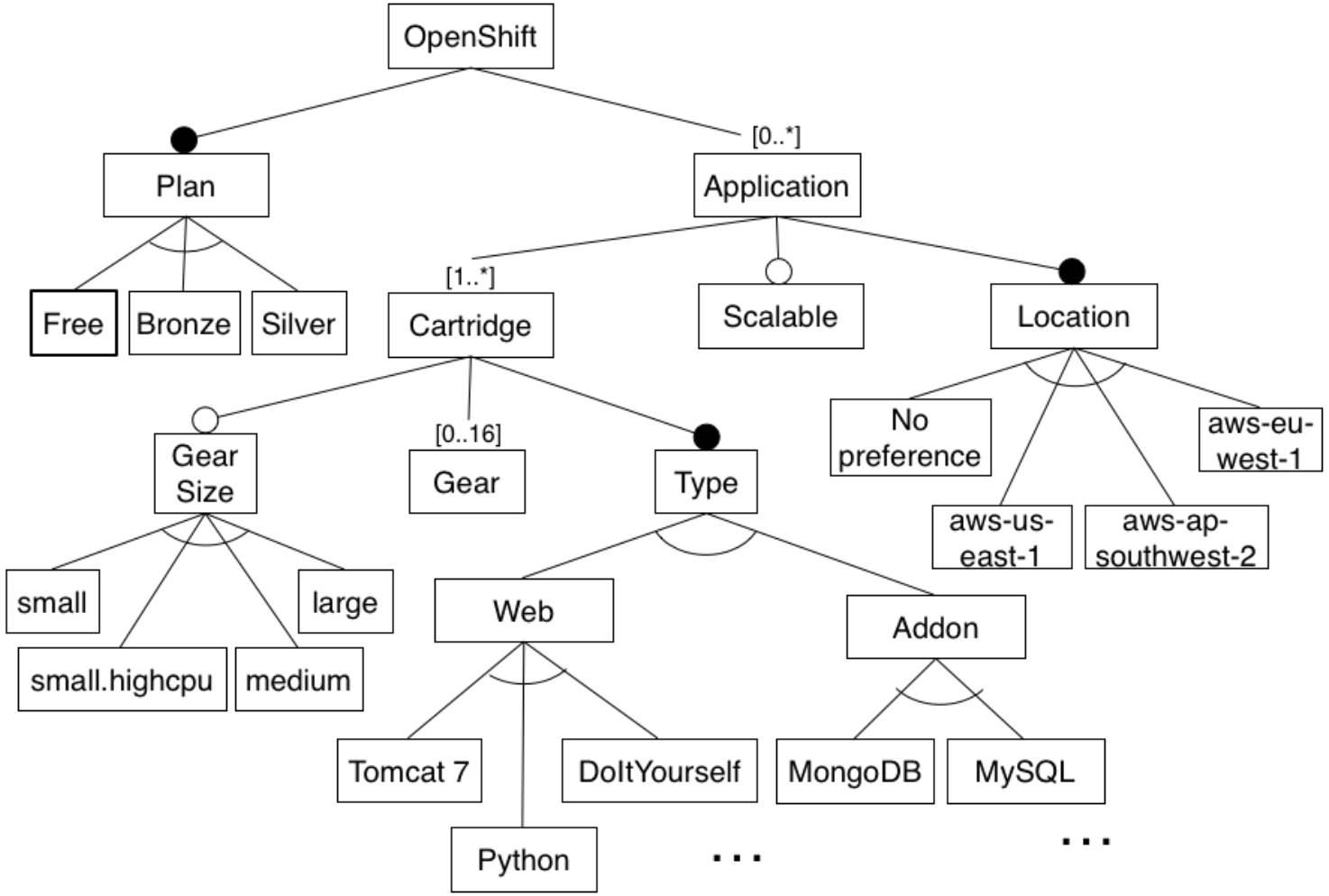
Cloud Provider Feature Model

- SALOON FMs only consider variability within one "application"
- Independent services are deployed in the cloud as individual "applications"
- Design of new FMs to enable description of microservice configurations

Single application FM for OpenShift



Multi-application FM for OpenShift

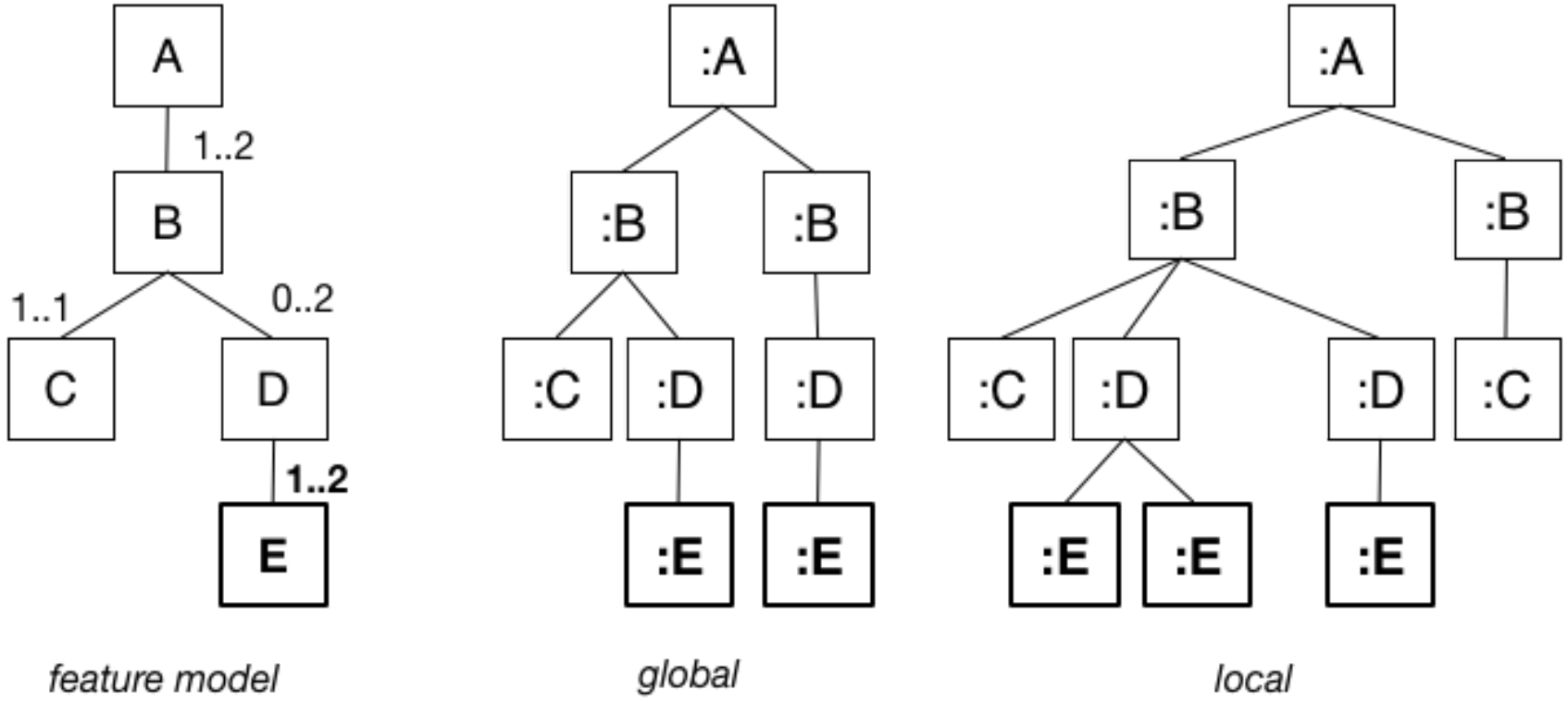


Problems found designing Cloud Provider FMs

- Limitations of constraints and cardinalities system

Limitations: Cardinalities

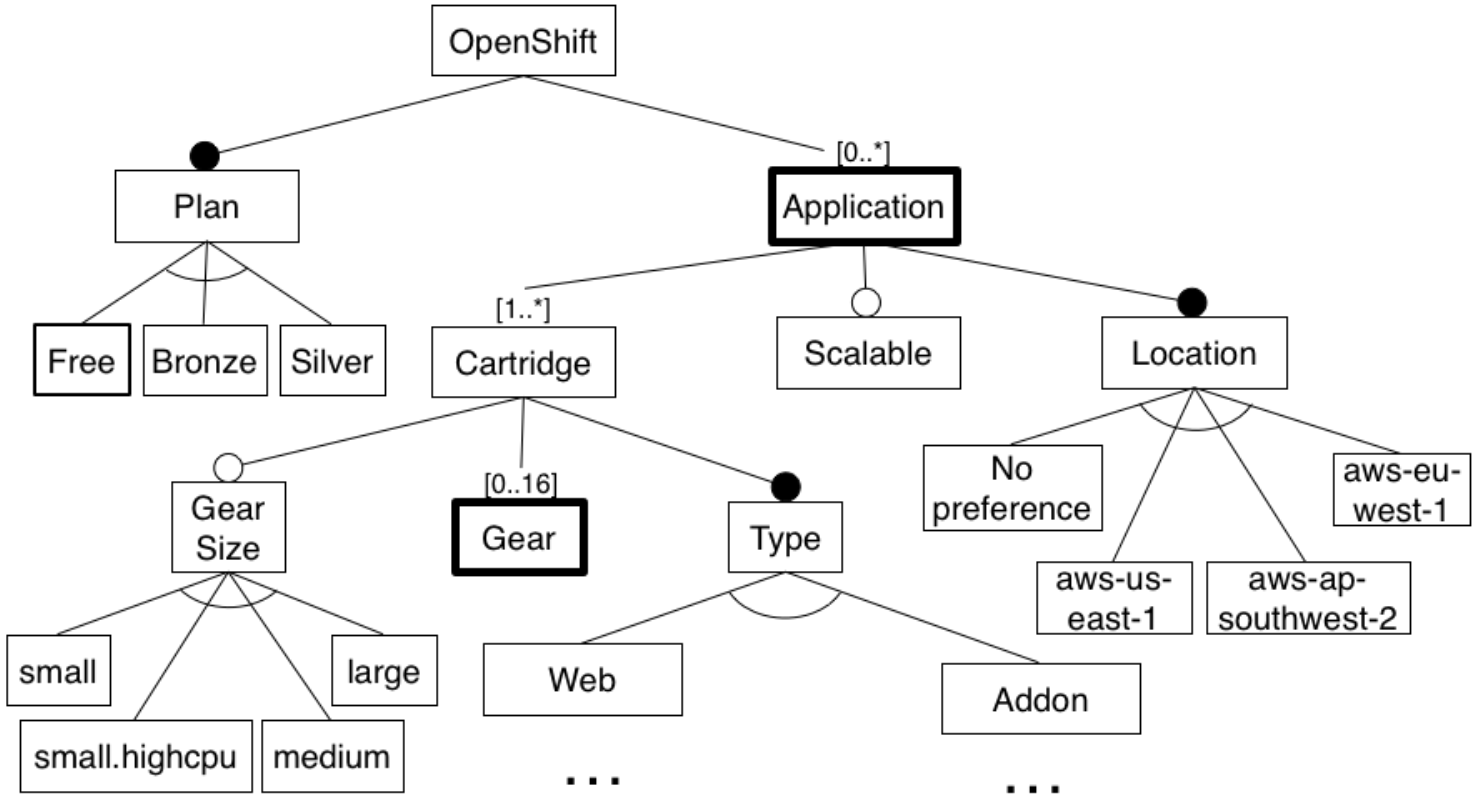
- Limitations of constraints and cardinalities system
  - Cardinalities interpreted either globally or locally



feature model *global* *local*

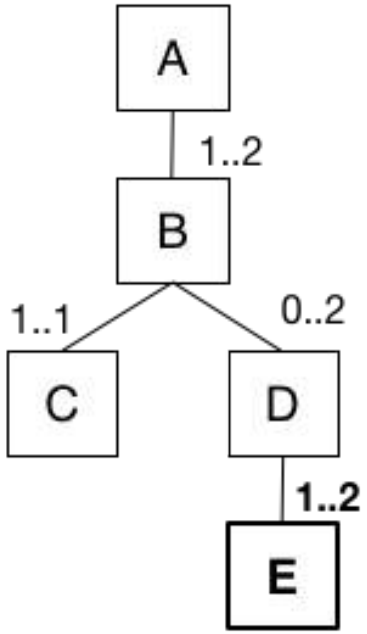
Limitations: Cardinalities

- `Cartridges` have `Gears`, but their number is related to an `Application`

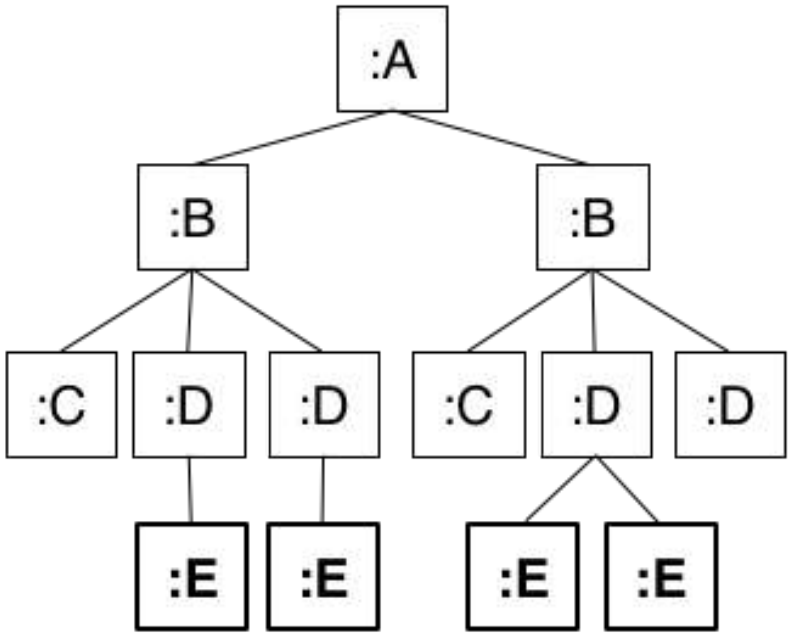


Proposal: Relative cardinalities

- A cardinality can be **local**, **global** or **relative** to any of its ancestors in the feature diagram tree



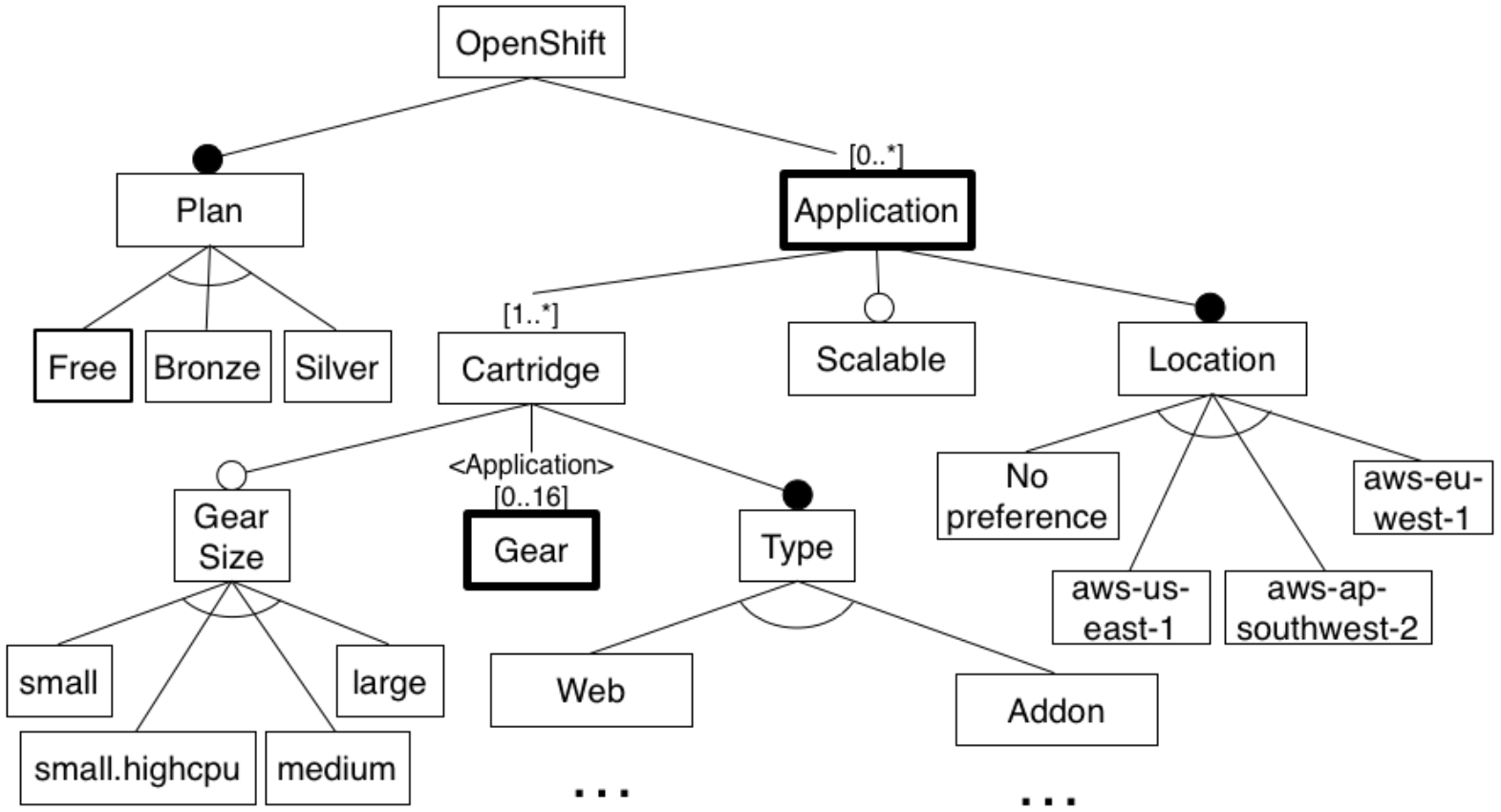
feature model



relative to feature B

Proposal: Relative cardinalities

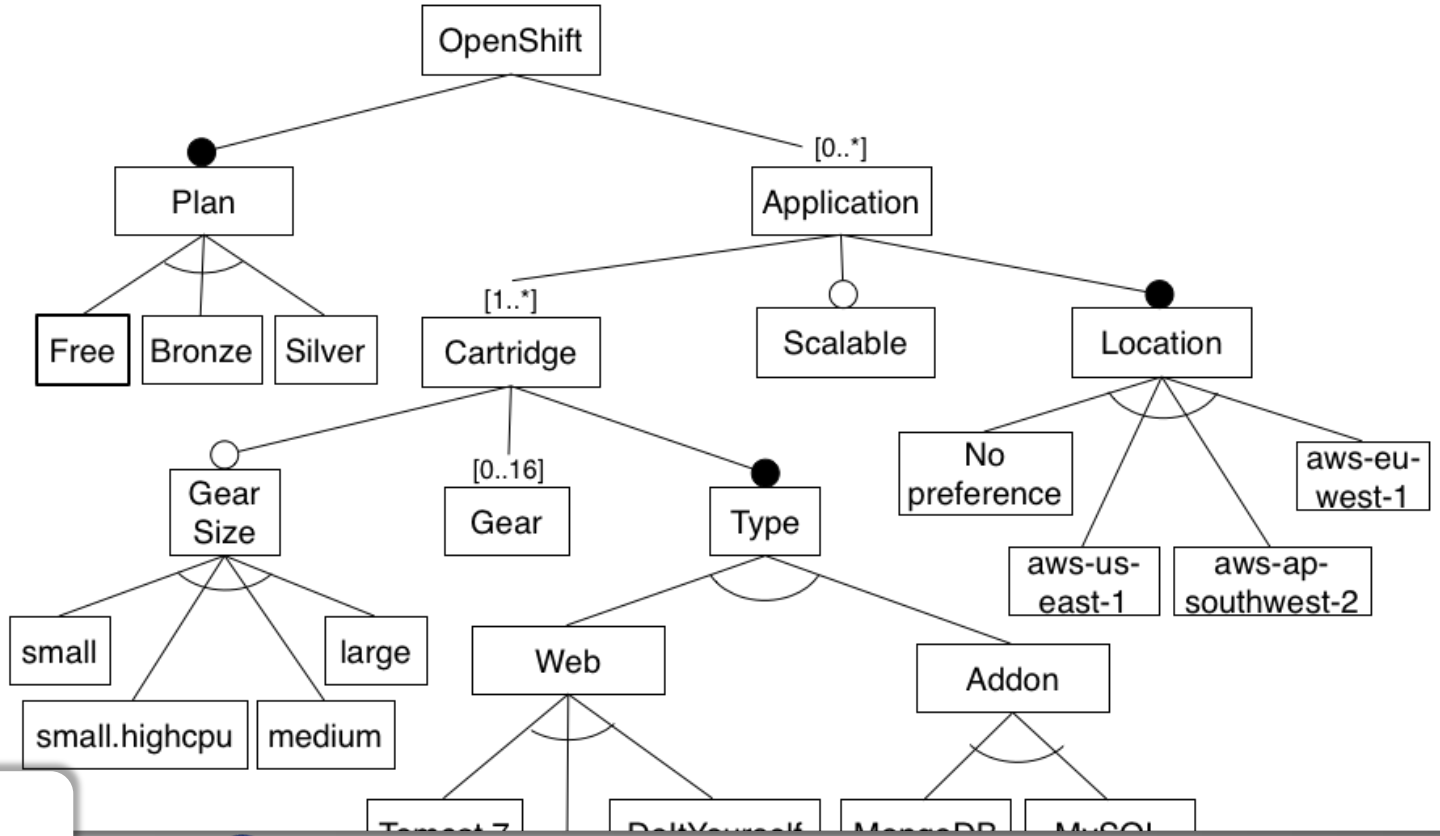
- A cardinality can be **local**, **global** or **relative** to any of its ancestors in the feature diagram tree



Consistency

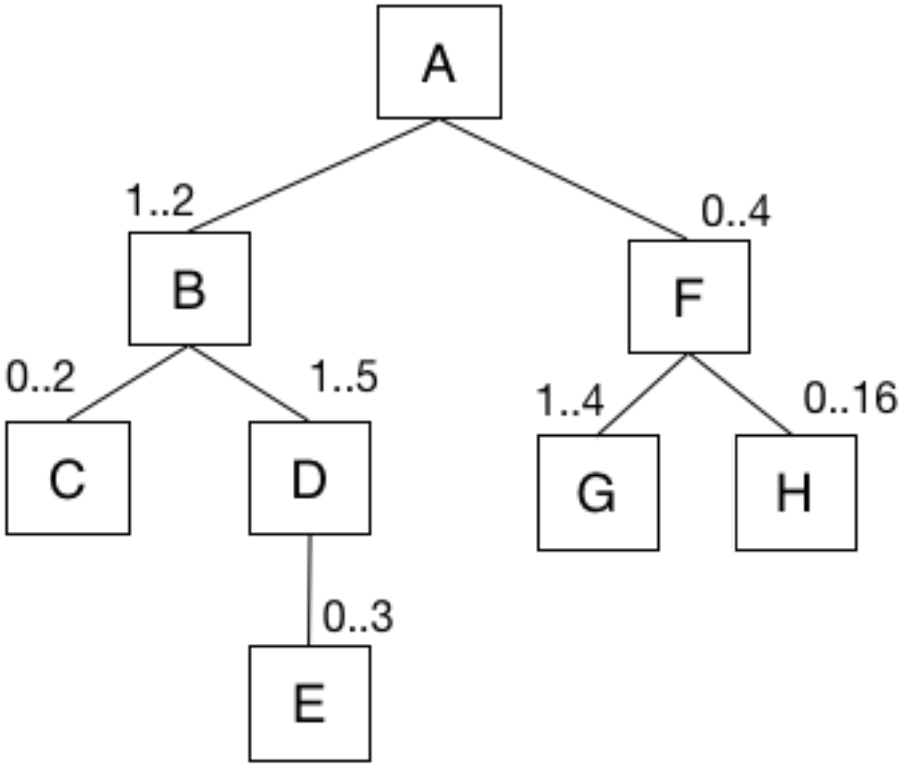
Limitations: Constraints

- When the `Free Plan` is chosen `Applications` can only be located in `aws-us-east-1` and all `Gears` should be `small`
- In non-`Scalable` applications, all `Addon` cartridges have 0 gears and all `Web` cartridges have exactly 1 gear



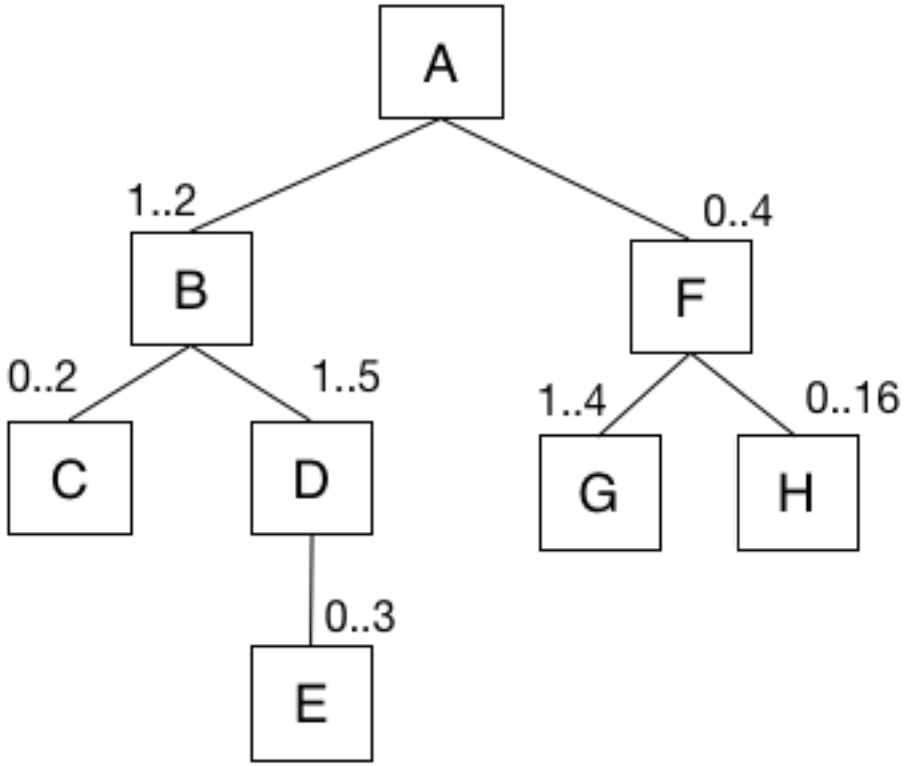
SALOON constraints

- [2..4] G → [8..16] H
- C' → +2 E



Limitations: Constraints

- For a given **B**, if it has at least one **C** then all its **D** instances should have at least one **E**

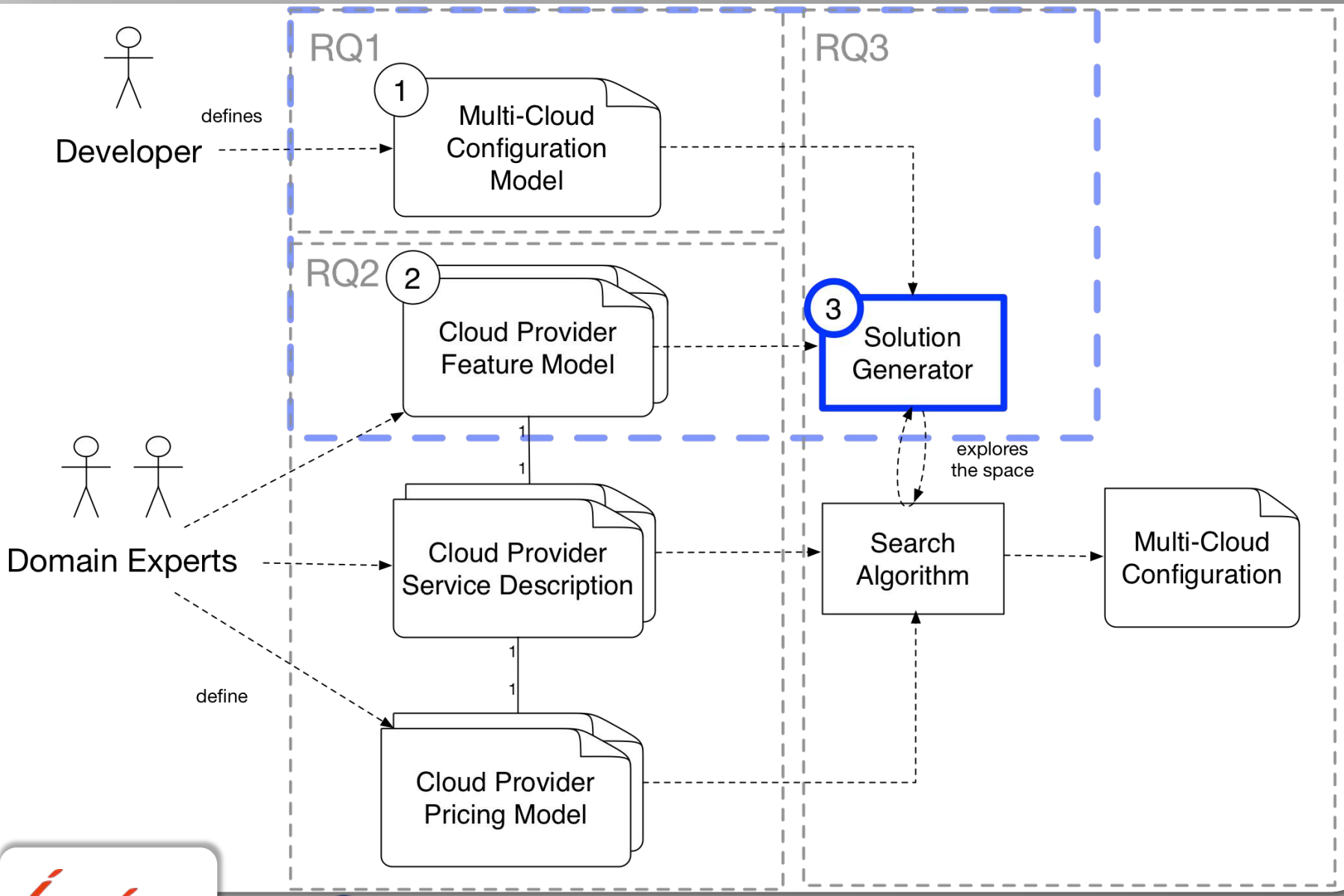


Cloud Provider Feature Model

- Current status
  - Designed multi-cloud provider FMs
  - Proposed relative cardinalities as a new FM construct

- Future work
  - Extend existing constraints for Cardinality-Based FMs
  - Consistency analysis for relative cardinalities and constraints

Overview


RQ1
RQ3
RQ2
1
Multi-Cloud Configuration Model
defines
Developer
2
Cloud Provider Feature Model
3
Solution Generator
explores the space
Domain Experts
Cloud Provider Service Description
Search Algorithm
Multi-Cloud Configuration
define
Cloud Provider Pricing Model

Solution Generator

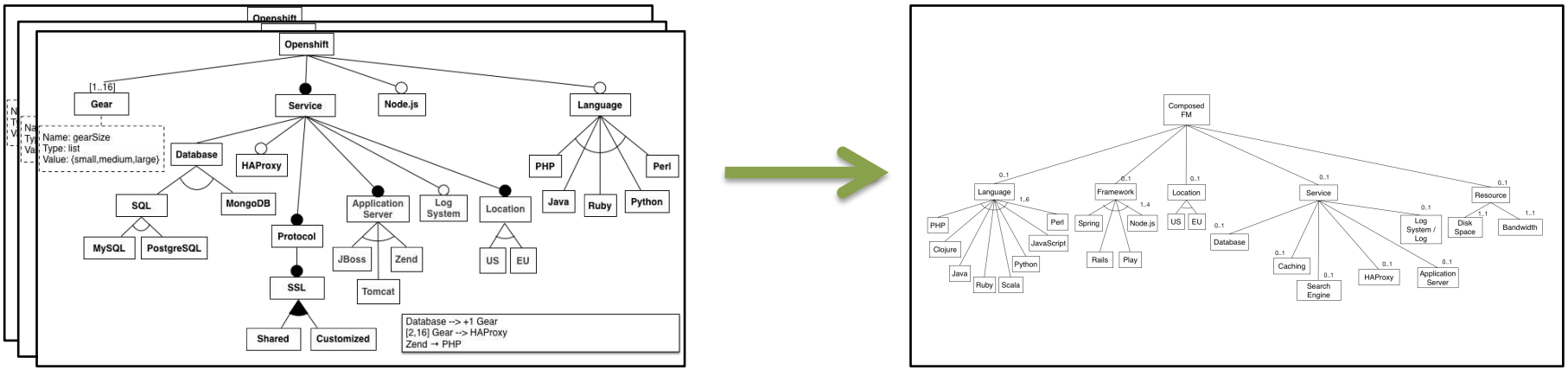
- Maps multi-cloud environment requirements to individual provider configurations
  - Provider selection complies with requirements
  - Provider configurations comply with the FM

- Similar to a multiple software product lines problem
  - Reuse of components from several suppliers (product lines) to generate new products

Hartmann, H.; Trew, T., "Using Feature Diagrams with Context Variability to Model Multiple Product Lines for Software Supply Chains," Software Product Line Conference, 2008. SPLC '08. 12th

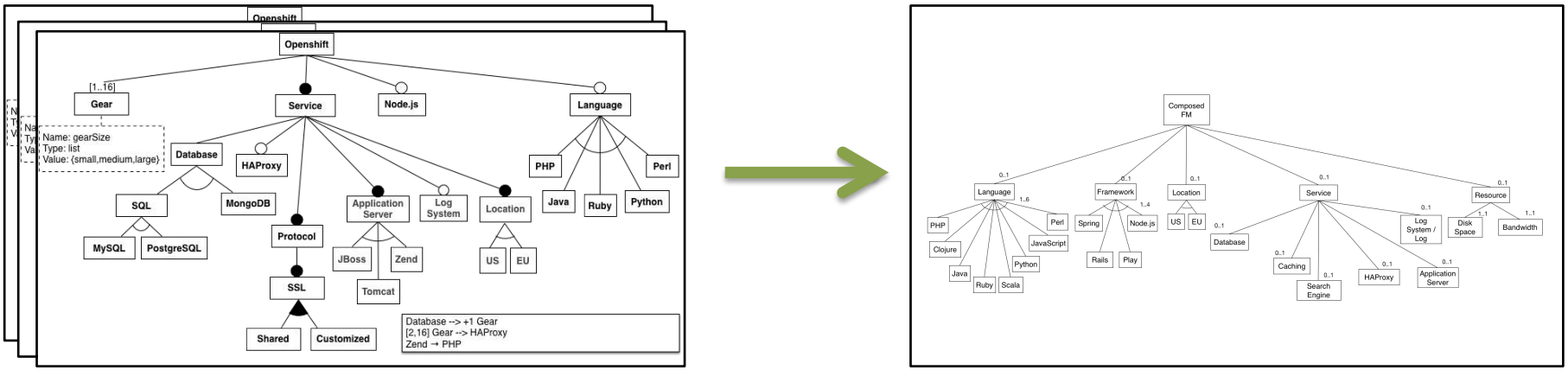
Solution Generator

- Feature model composition
  - Used in basic FMs



Solution Generator

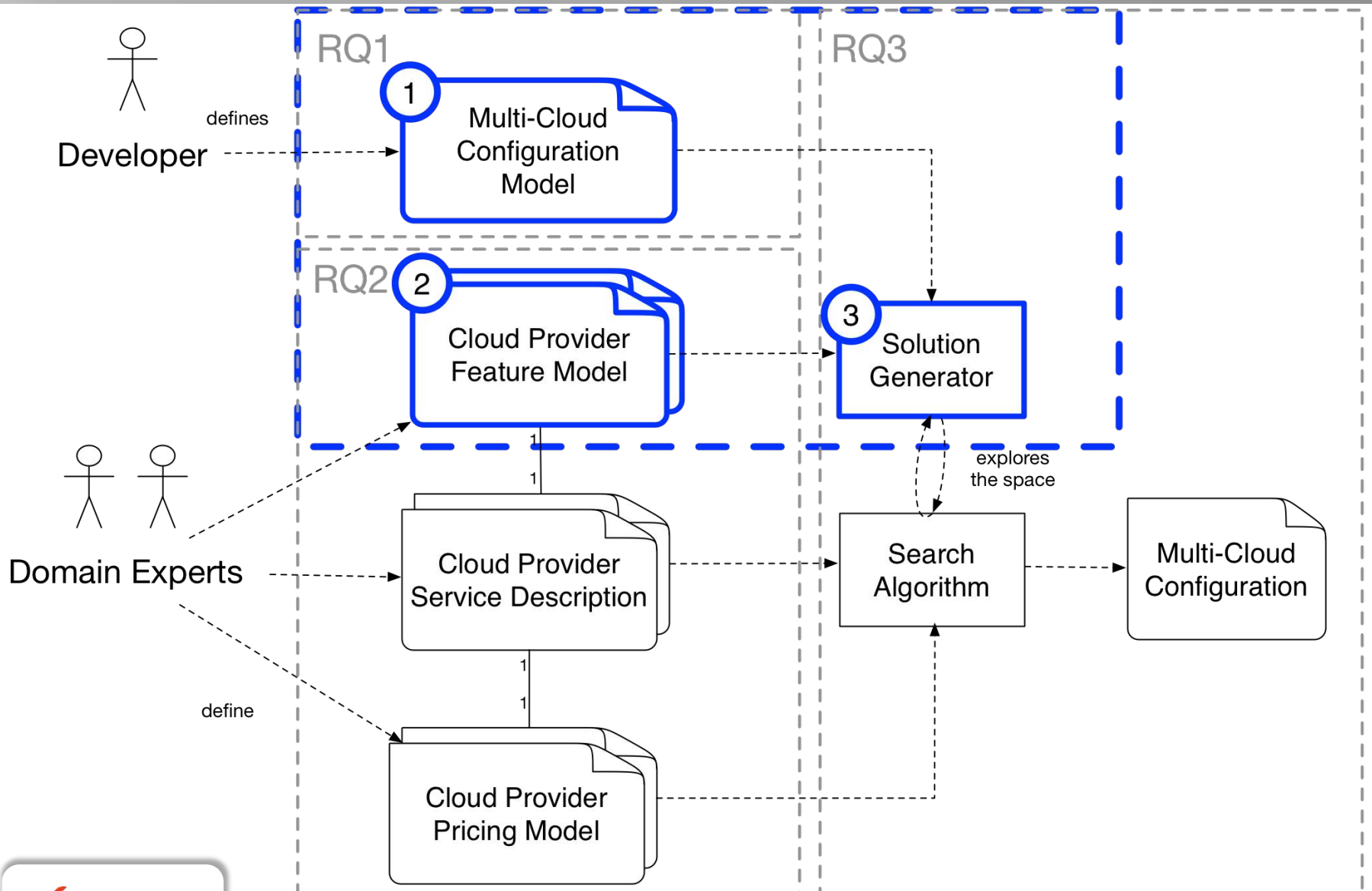
- Feature model composition
  - Used in basic FMs
  - Cardinality-based FM configurations include their hierarchy
  - Composing models showed to add extra complexity in the case of cloud provider feature models



Solution Generator

- Current status
  - Evaluated the use of feature model composition
- Future work
  - Investigate alternative methods for mapping multi-cloud requirements to multiple feature models

Conclusions



Conclusions

- Characterized the multi-cloud configuration problem
- Designed FMs to enable multi-application configurations
- Introduced the concept of relative cardinalities
- Identified issues with constraints in cardinality-based feature models

Future work

- Generate valid multi-cloud configurations
  - Abstract and describe differences between providers' features
  - Map provider-independent requirements to feature model configurations
- Model pricing and service level policies

- Identify appropriate search strategies for the optimizing multi-cloud configurations