

Automation and Simplification Through Declarative Performance Engineering

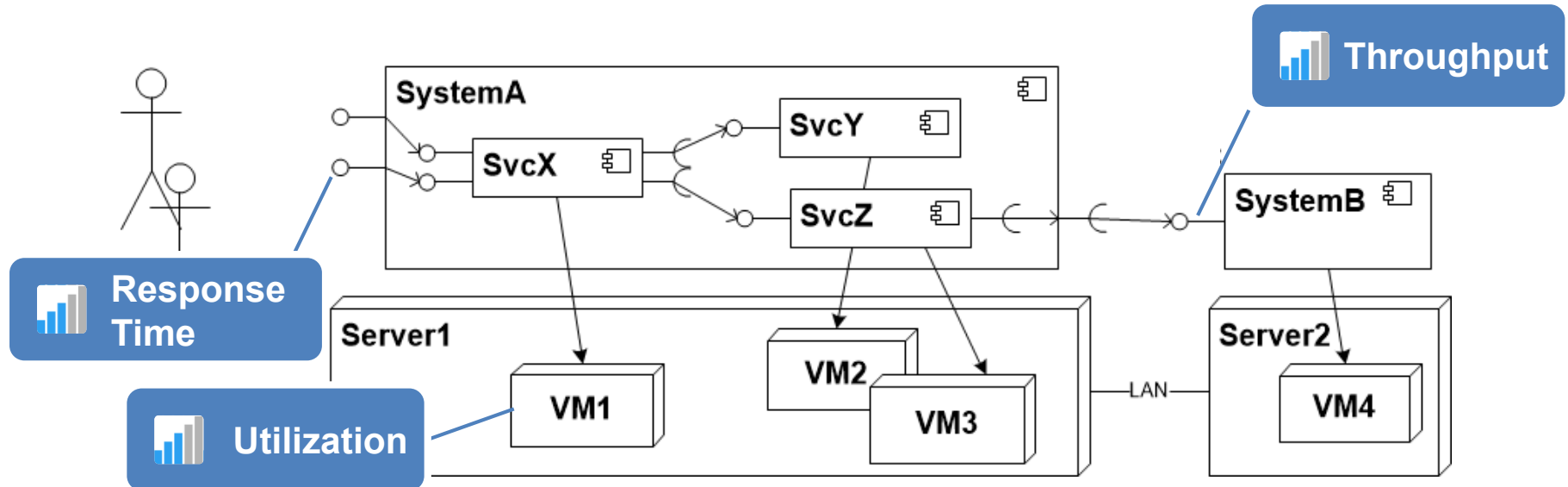
Jürgen Walter

Chair of Software Engineering
University of Würzburg

<http://se.informatik.uni-wuerzburg.de/>

Doctoral Colloquium June 14th 2016

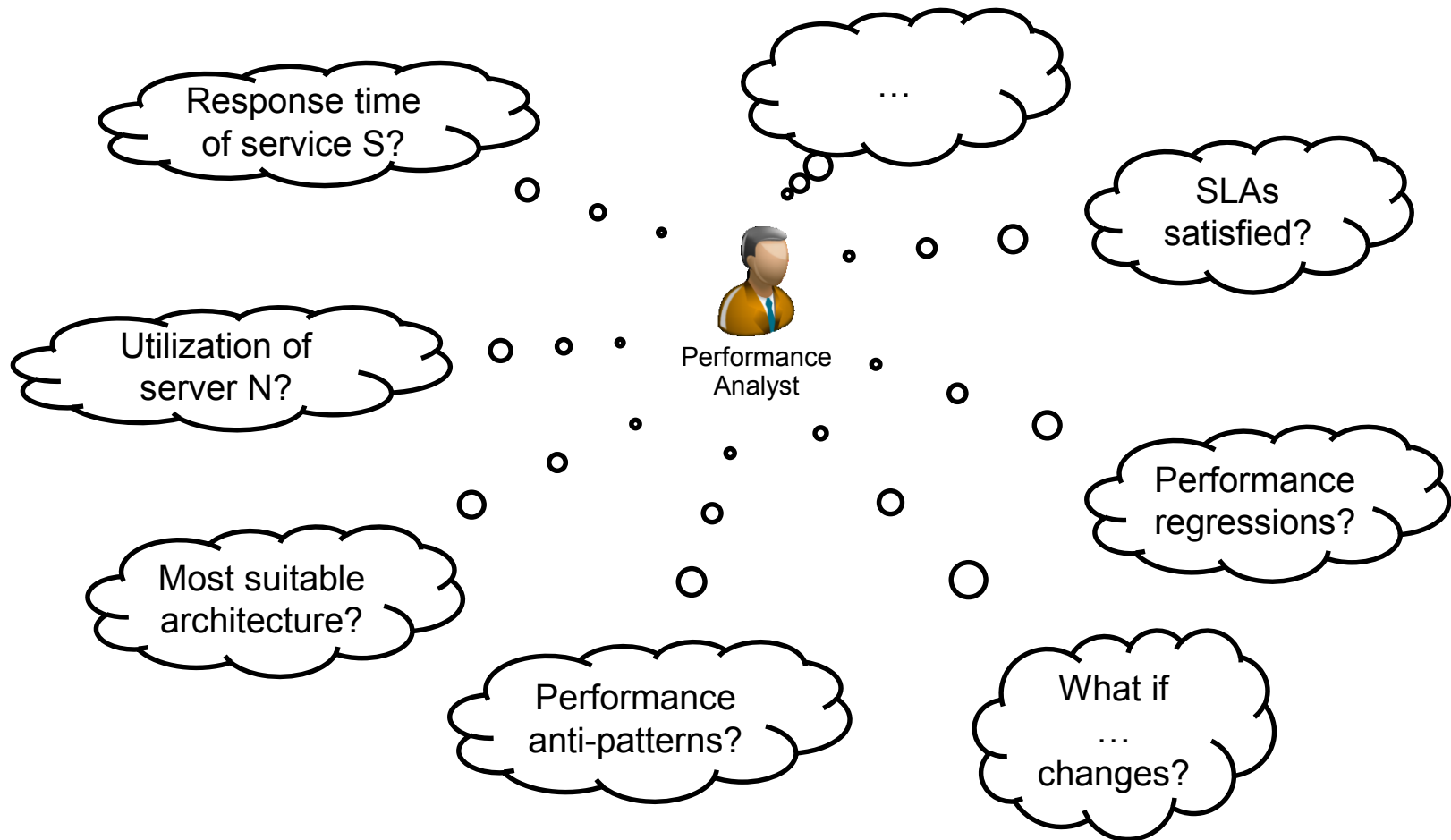
- Performance = timing behavior + resource usage



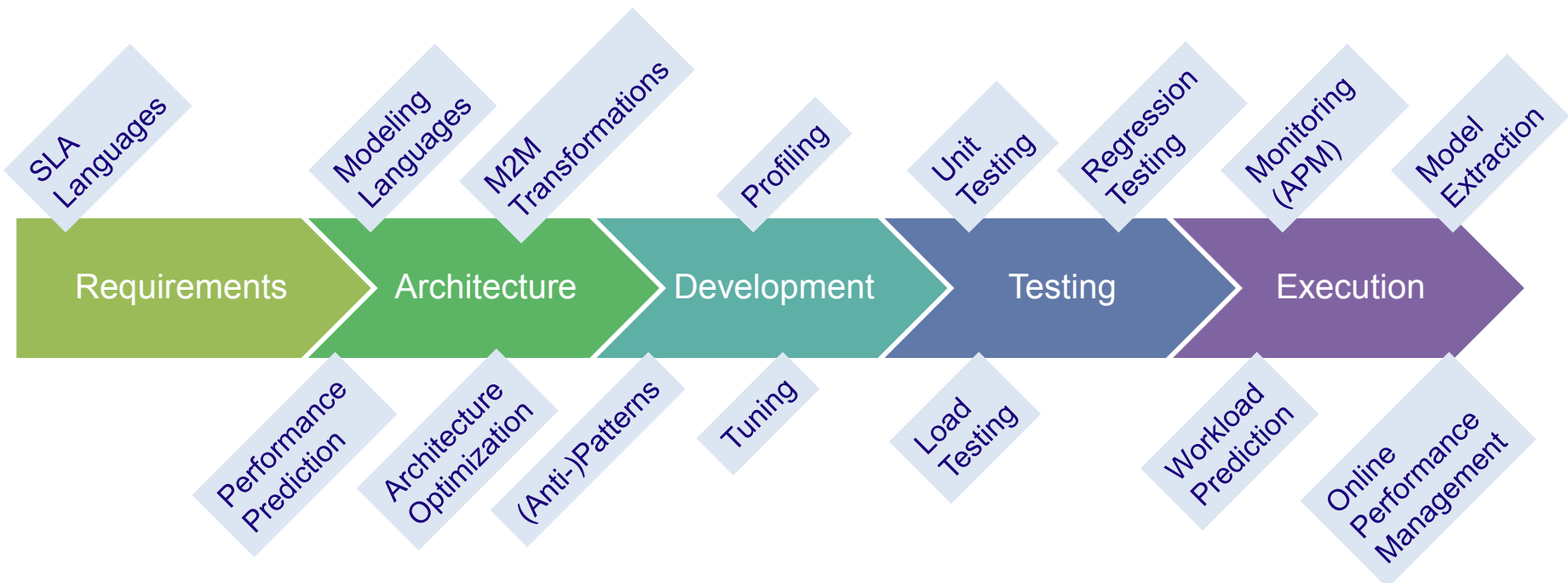
“the entire collection of **software engineering activities and related analyses** used throughout the **software development cycle**, which are directed to meeting **performance requirements**.”



Performance-Relevant Concerns Spanning the Software Lifecycle

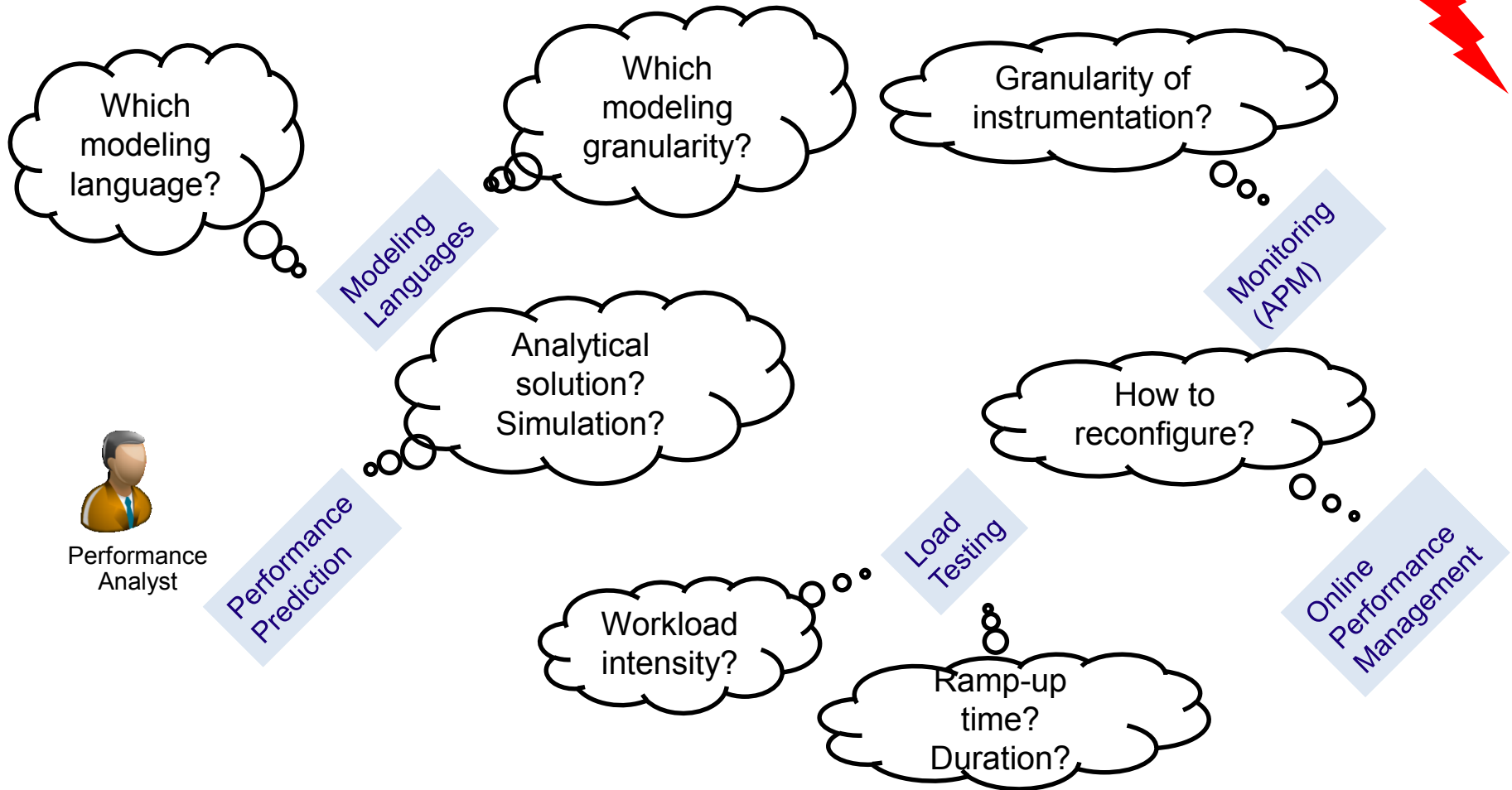


Extensive Body of Software Performance Engineering Exists



Brunnert, A., van Hoorn, A., Willnecker, F., Danciu, A., Hasselbring, W., Heger, C., Herbst, N., Jamshidi, P., Jung, R., von Kistowski, J., Koziol, A., Kroß, J., Spinner, S., Vögele, C., Walter, J. und Wert, A., eds . (2015) Performance-oriented DevOps: A Research Agenda Technical Reports of the SPEC Research Group

Problem Statement: Various Decisions to Apply SPE Correctly



Problem Statement: Various Decisions to Apply SPE Correctly

Performance Concerns



Performance Analyst

Software Performance Engineering

Challenges

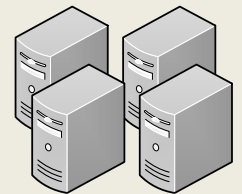
- Choice of solution strategy
- Parametrization of approach
- Result filtering and interpretation



Established Methods, Techniques, Tools

- Measurement-based
- Model-based

System



Performance
Concerns



Performance
Analyst

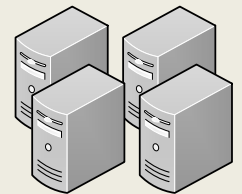
Software Performance Engineering

Established Methods, Techniques, and Tools

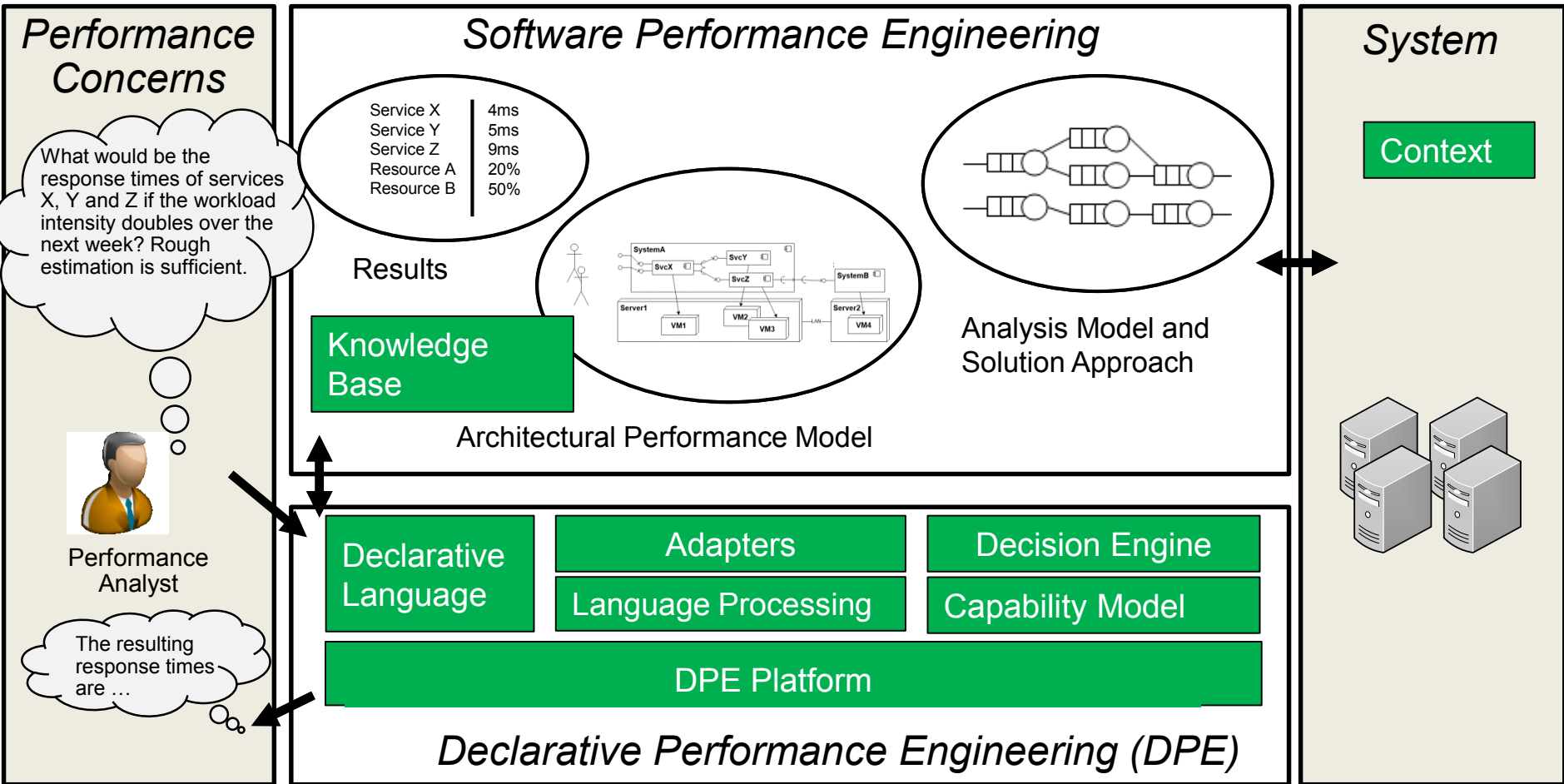
Asking “What?“,
Automating the “How?“

Declarative Performance Engineering (DPE)

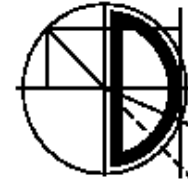
System



Jürgen Walter, Andre van Hoorn, Heiko Koziolk, Dusan Okanovic, and Samuel Kounev. Asking “What?“, Automating the “How?“: The Vision of Declarative Performance Engineering. In *Proceedings of the 7th ACM/SPEC International Conference on Performance Engineering (ICPE 2016)*, Delft, the Netherlands, March 12, 2016.

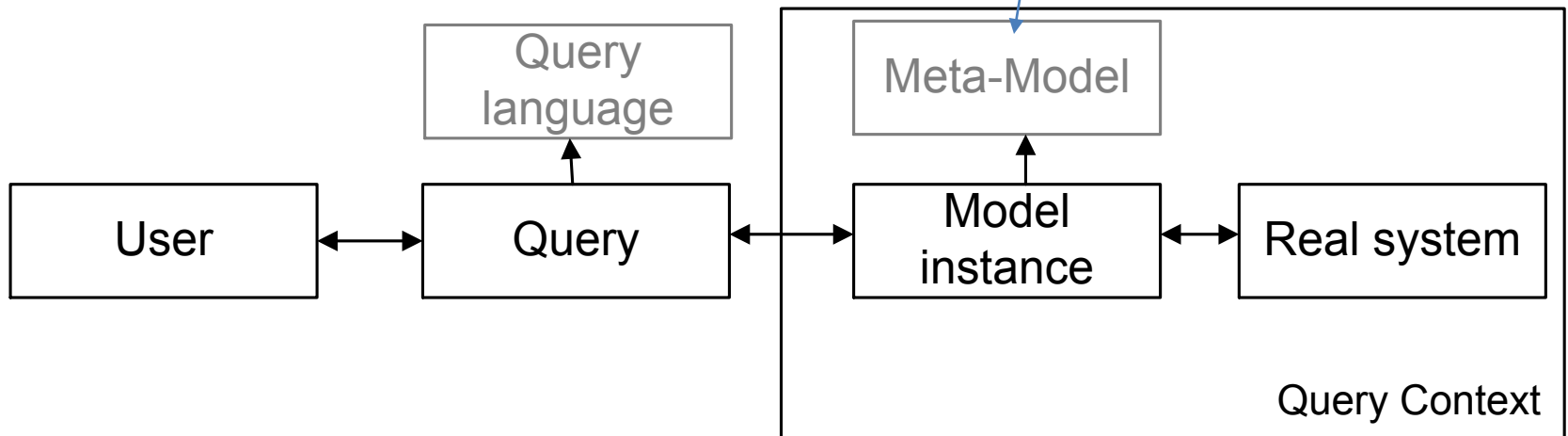


I say/define **what** i want to know,
the **how** will be automatically derived from what



<http://descartes.tools/dml>

(can be an arbitrary model)



- A **context** is a description of a system under test that uses and includes a (formalized) meta-model
 - **Context Model**... the model instance that describes the system under test
 - **Context Meta-Model** ... the meta-model to specify used to interpret the context model. Offers semantics for context model
 - ...the formal language in which the model is defined/specified/expressed

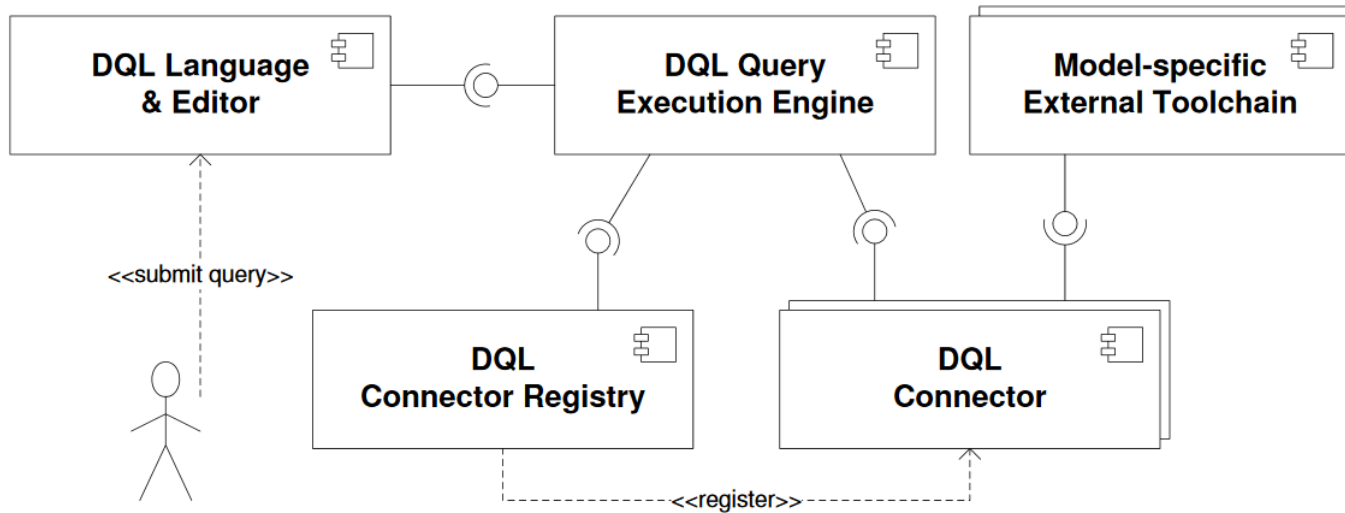
- A **Queryable Element (QE)** is an element of the meta-model for whose instances the respective a given set of queries is meaningful
 - an element of a scenario model
 - or the respective scenario meta-model
- **Queryable Element Context** is a connection of one or more objects of the MM, which refines the QE (e.g., containment)
 - Macht die Frage eindeutig interpretierbar
 - An objekt may be reference multiple times. If a QE is not identifiable then its embedding context has to be used additionally
 - One service may be instantiated multiple times
 - Solution: `SELECT AssemblyContext1.Service1 ...`
 - Solution: `SELECT AssemblyContext2.Service1 ...`

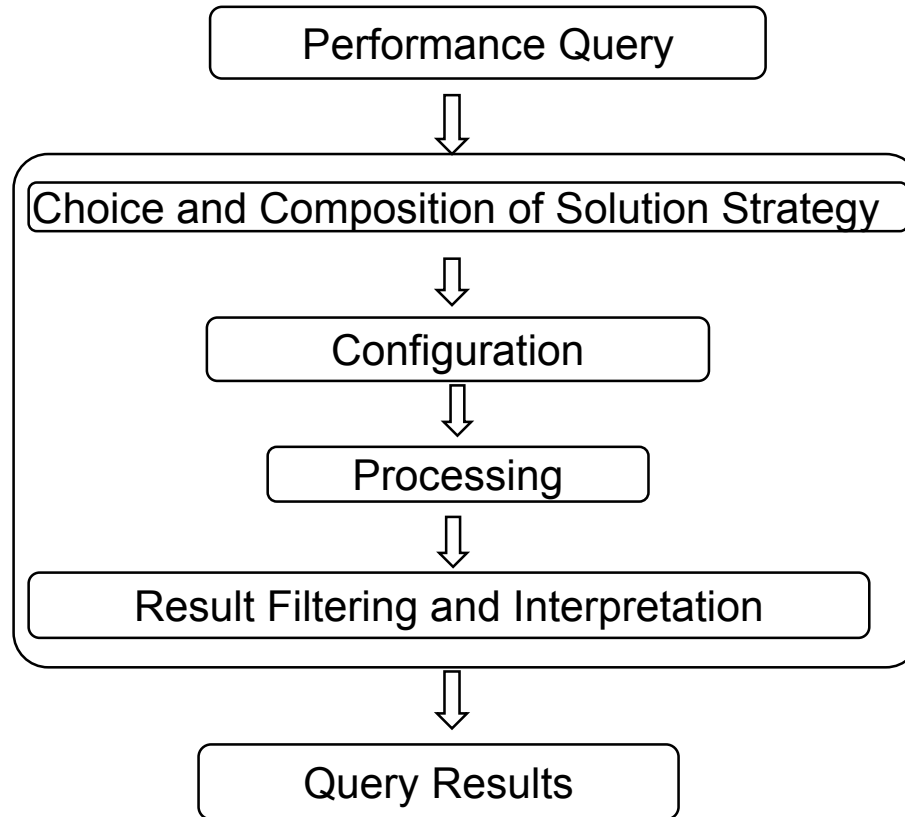
- SQL like interface for different Performance Engineering approaches



<http://descartes.tools/dql>

```
⇒ SELECT resource1.utilization, service1.avgResponseTime  
⇒ FOR RESOURCE 'id1' AS resource1,  
    SERVICE 'id2' AS service1  
    USING dml@'model';
```

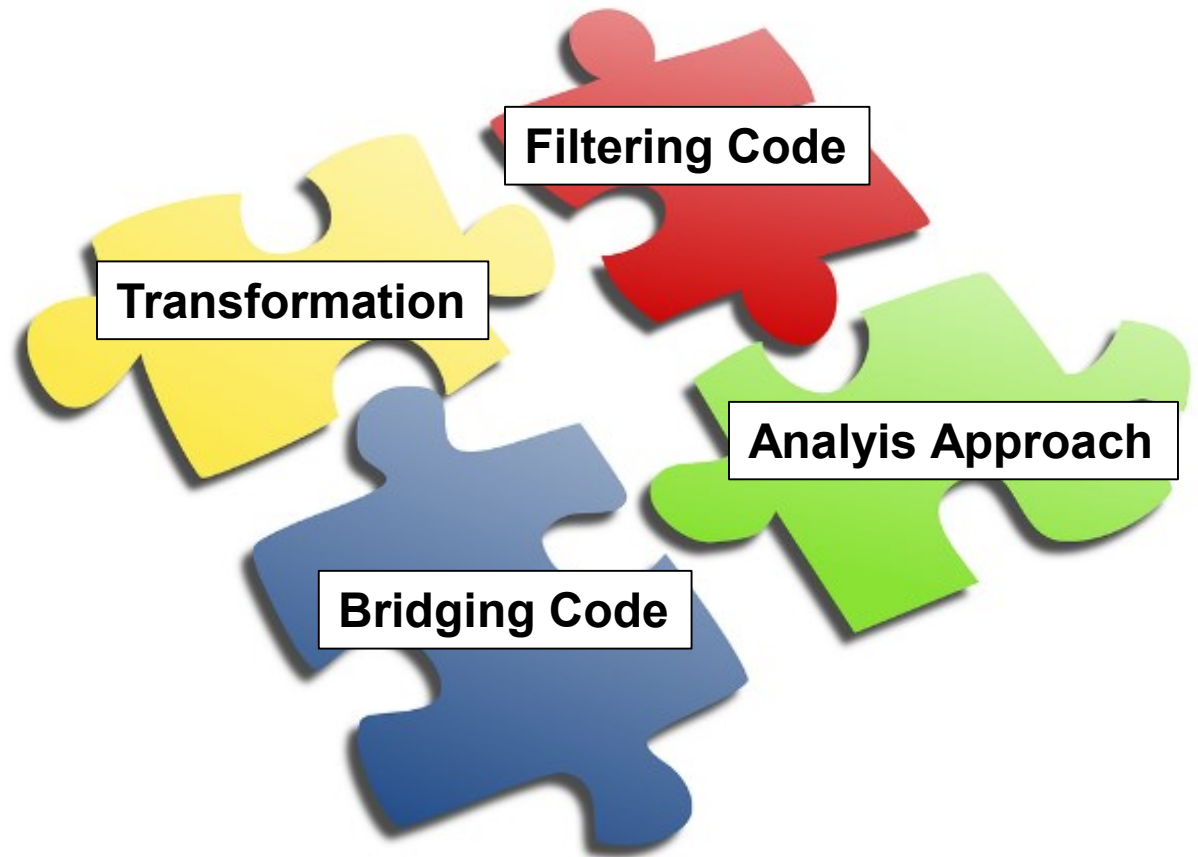




Note: Arrows depict dependencies or drives but do not imply strict ordering



Solution
Strategy
Expert



A **Solution Strategy Expert** chooses and composes model transformations and solution approaches to solve a query == solution receipt



Solution
Strategy
Expert



- Receipts can be **reused** if solution strategy is based on meta-model
- Receipts can be **formalized and implemented** in a solution strategy adapter (more on this later)
- ➔ Many possible solution strategies for one query that differ in speed, accuracy and provided statistic type
- ➔ Solution Strategies may not be able to answer questions for only a subset of queryable elements and respective metrics
- ➔ **Need for comparison** of different solution strategies



Solution
Strategy
Expert



- Solution expert implements a reusable strategy and describes its capabilities in formalized **Solution Strategy Capability Model** (using a **Solution Strategy Capability Meta-Model** that points to the *scenario meta-model*)
- Then, ---for a given query---- a **decision engine** may chose a suitable solution strategy based on a set of solution stratgey capability models

Expert System?!

- Is the envisioned decision support some kind of an expert system?
- Is it kind of an expert system but
 - Usually expert systems are just a type of database that directly map answers to a questions. Here, it is more complex

- Users wants to influence **costs** and **accuracy** of the query processing
 - If user requests a fast response, then the solution strategy shall adapt to this (usually to the cost of accuracy)
 - If user requests an accurate response, then the solution strategy shall adapt to this (usually to the cost of costs/time)
- Tradeoff example:
 - Reduce simulation/analysis/measurement time
 - Reduced warm-up period/Initialization bias
 - Switch analysis techniques: use analytical solvers or simulation

Quantitative case studies

Comparisons of
Performance and Accuracy

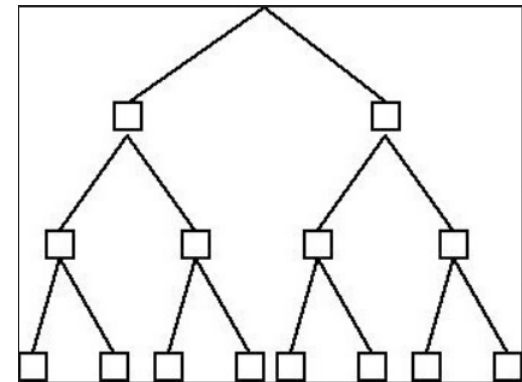
→ No automation



- Koziolk [Koz10]
- Banks et al. [BCN00]
- Balsamo et al. [BMIS04]

Tree-based decision support

→ Vague consideration of query
and context



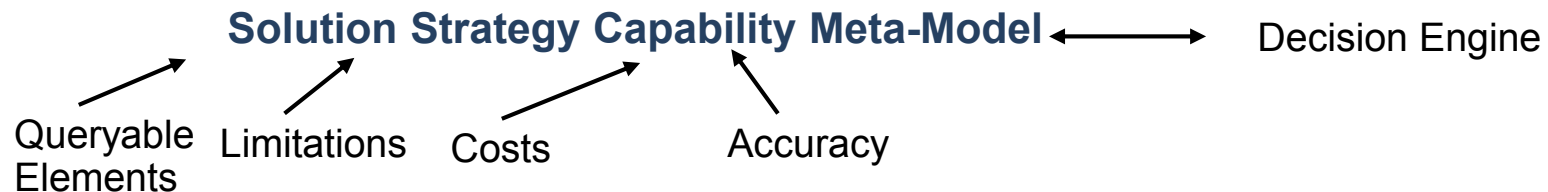
- Bolch et al. [BGdMT00]
- Brosig et al. [KHSB14]

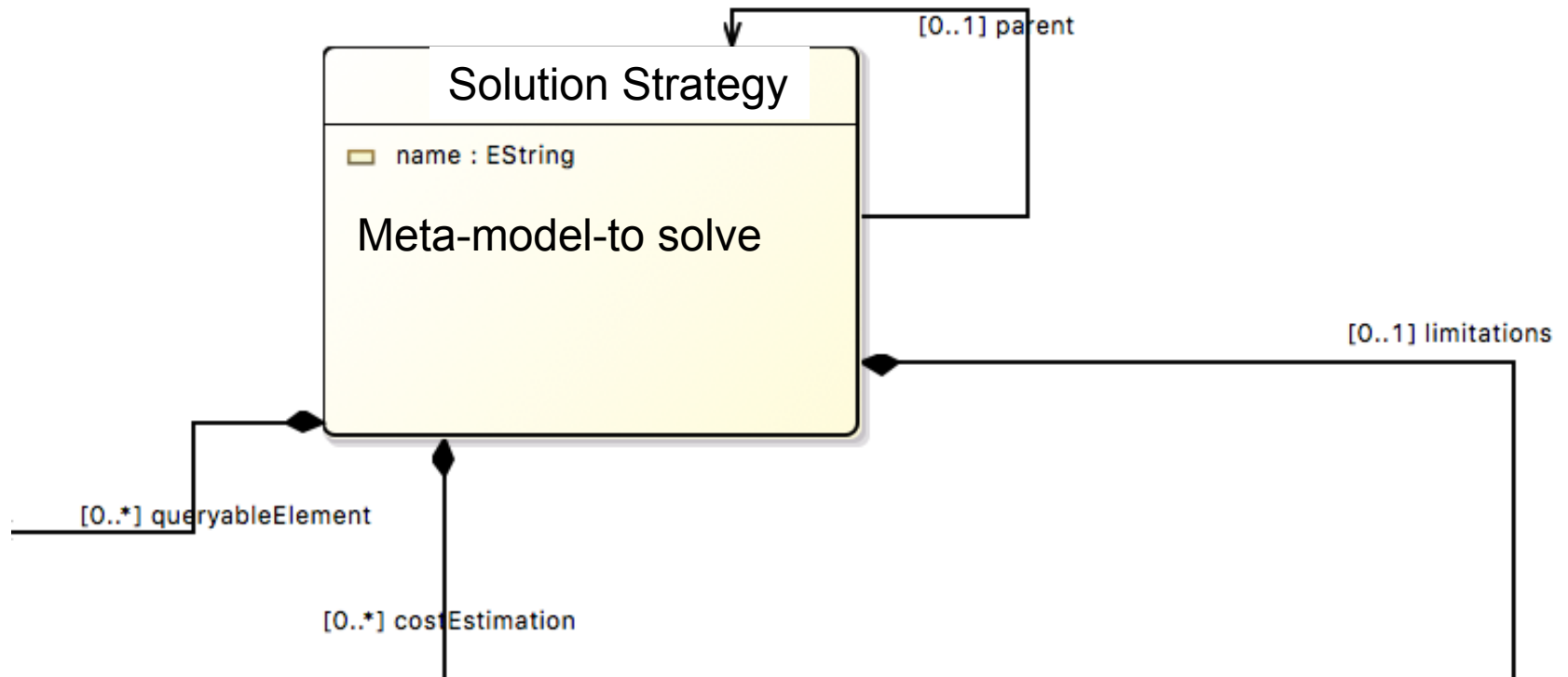
Decision support should consider

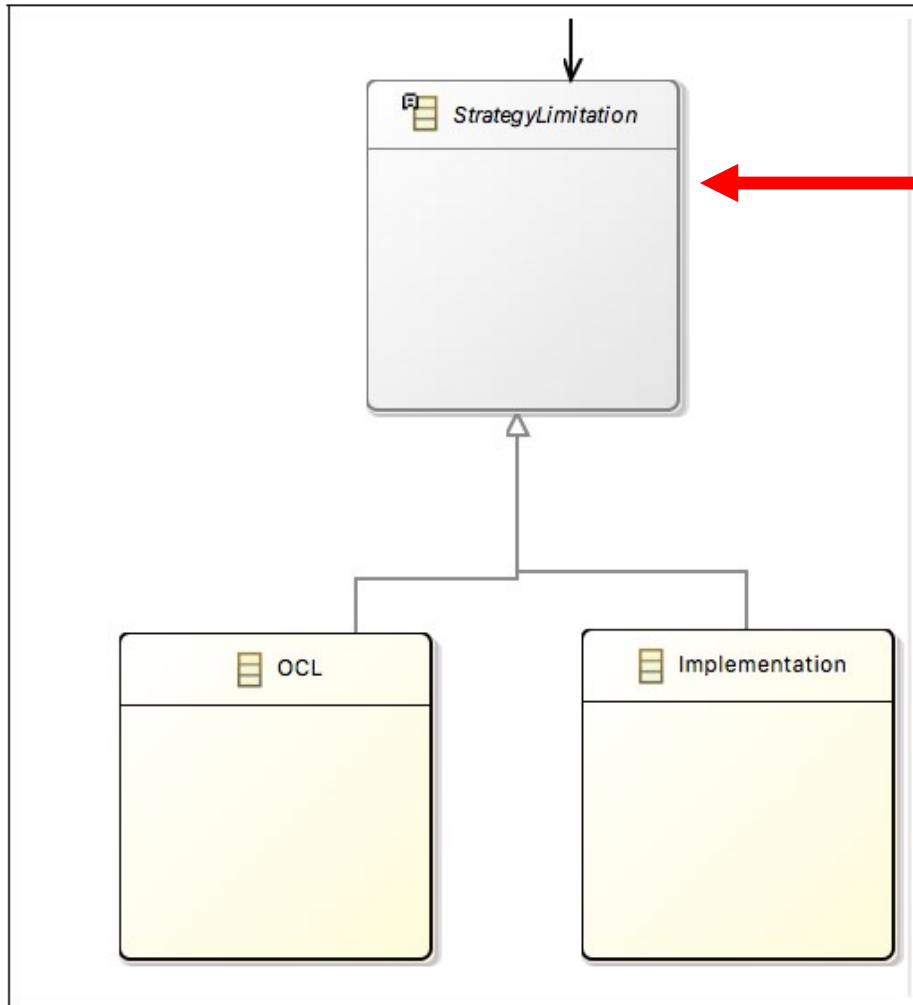
- model properties (model size, modeling deepness, number of users, ...)
- query properties (fast, accurat, system perturbation, ...)

How to compare solution strategies?

→ Split into decision engine and meta-model





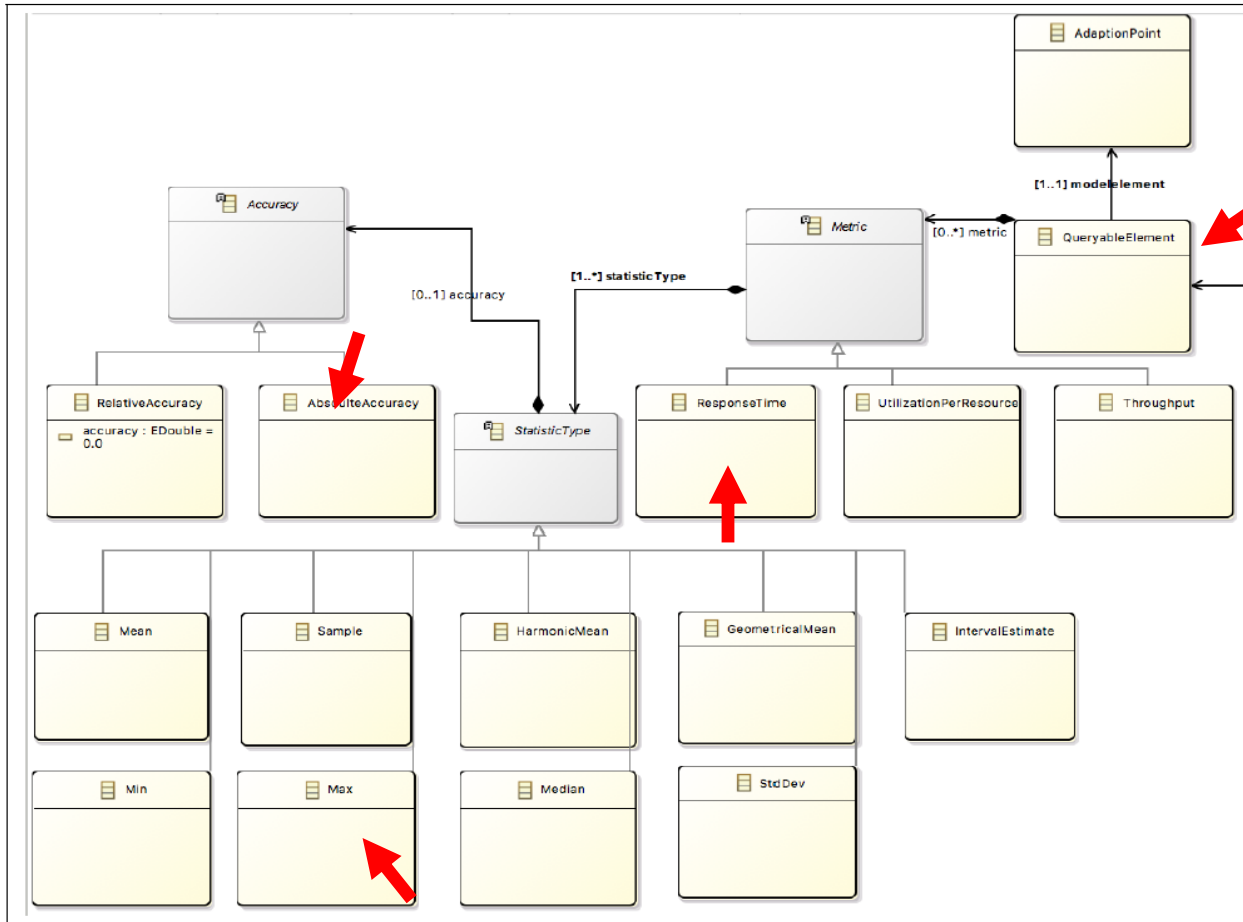


Will my analysis start and terminate for a given model/input?

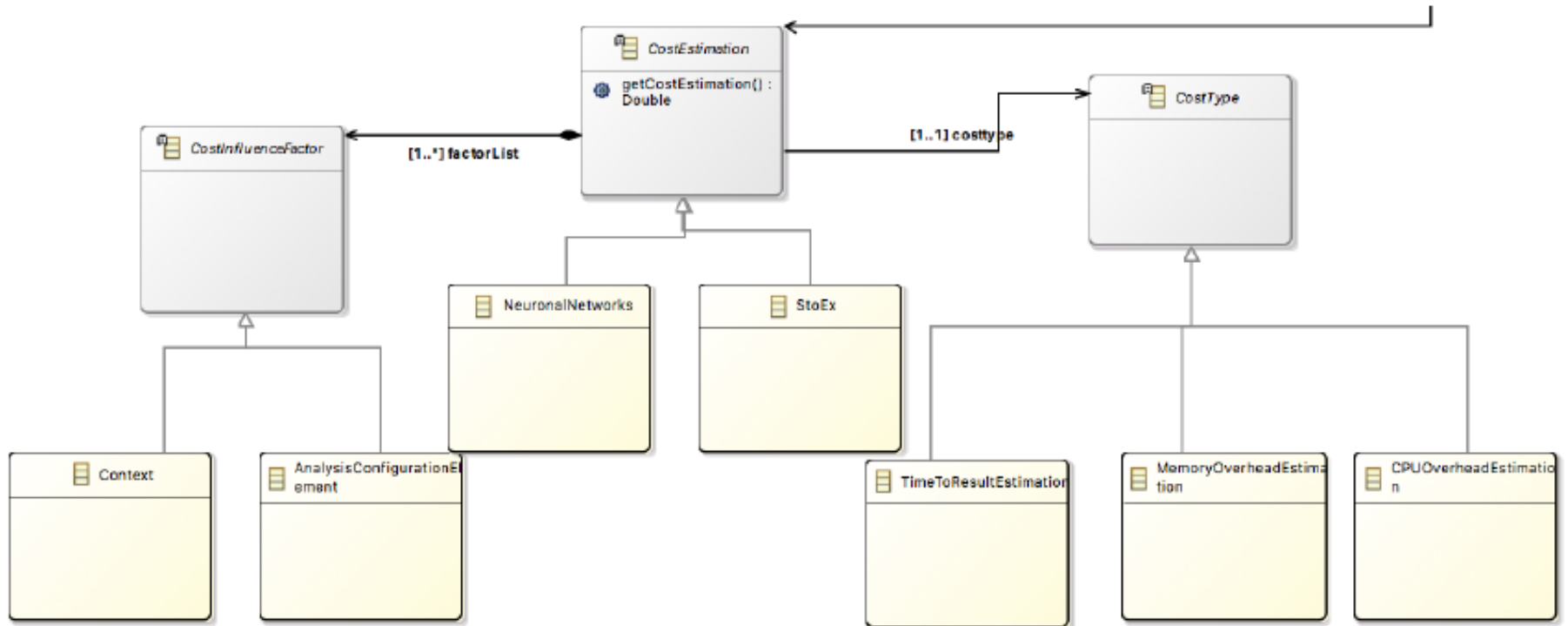
Limitation Repository that contains set of PredefinedStrategyLimitation for a certain meta-model to solve, e.g.

- Support for loops
- Support for open workloads

Queryable Element

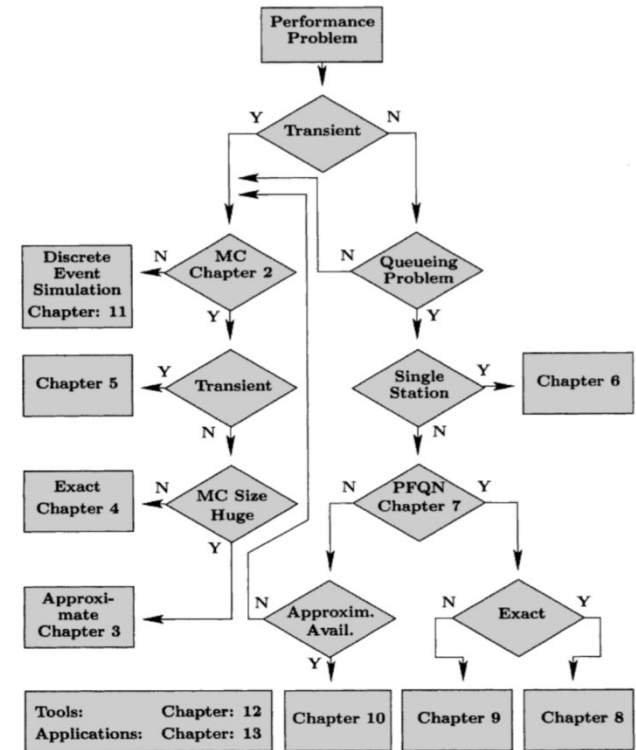
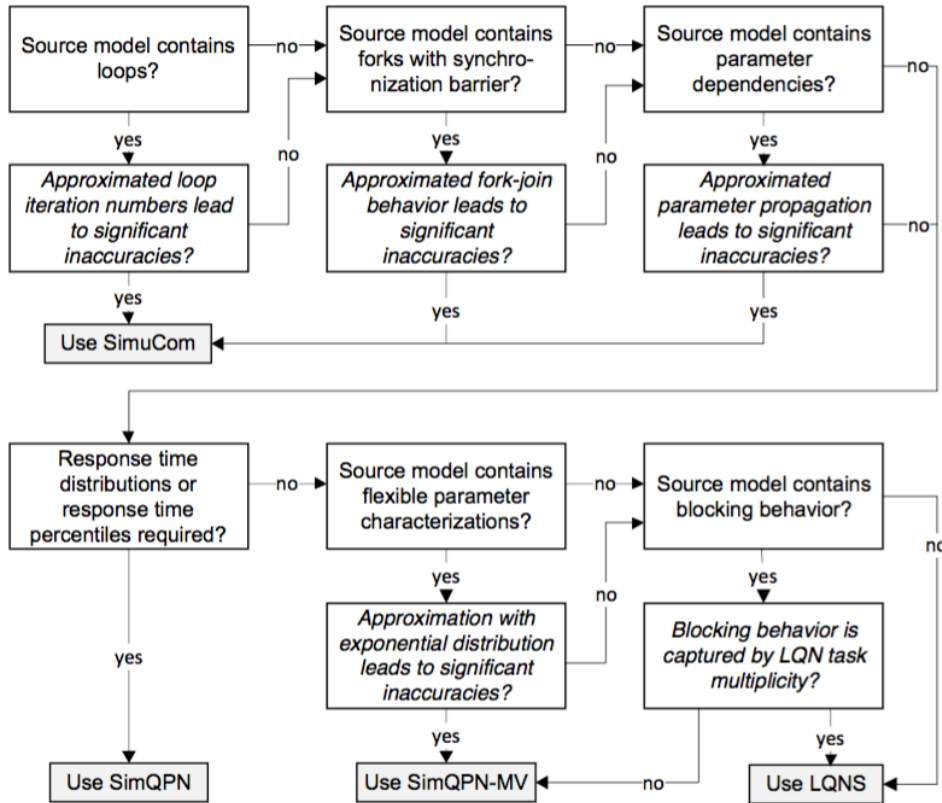


Is solution approach able to solve a given question?



Is solution approach capable to solve a given question?
➔ Cost functions

Example Trees

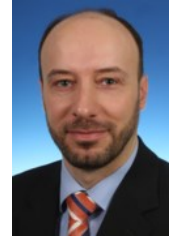


- Reduced Decision Logic
 - All Analysis approaches → Check Limitations
 - For remaining analysis approaches → Check support for the right queryable element
 - For remaining analysis approaches → Estimate costs and choose cheapest
- Capability Meta-Model provides a methodology for the comparison of solution strategies
- Simple addition of new comparisons attributes
- Simple addition of new solution strategies (as one does not rely on the knowledge of all other approaches)

- What if queries → query all changes and then do result filtering
- Extension and extendability to other query targets → System.bottlenecks
- Navigate through scenario model(e.g., affected subservices)
- Result presentation according to query
- Cost estimation for analysis approaches (machine learning techniques)
- Integration of goals to the DQL
 - Strategies Tactics Actions) (S/T/A) framework as one solution approach (applied for adaptation at runtime)

Project PIs

- Dr.-Ing. **André van Hoorn** (Prof.-Vertr.), University of Stuttgart
- Prof. Dr.-Ing. **Samuel Kounev**, University of Würzburg



Members

- Dr.-Ing. **Dušan Okanović**, University of Stuttgart
- Dipl.-Inform. **Jürgen Walter**, University of Würzburg



Associated Partners

- **Capgemini** Deutschland GmbH, Stuttgart, Germany



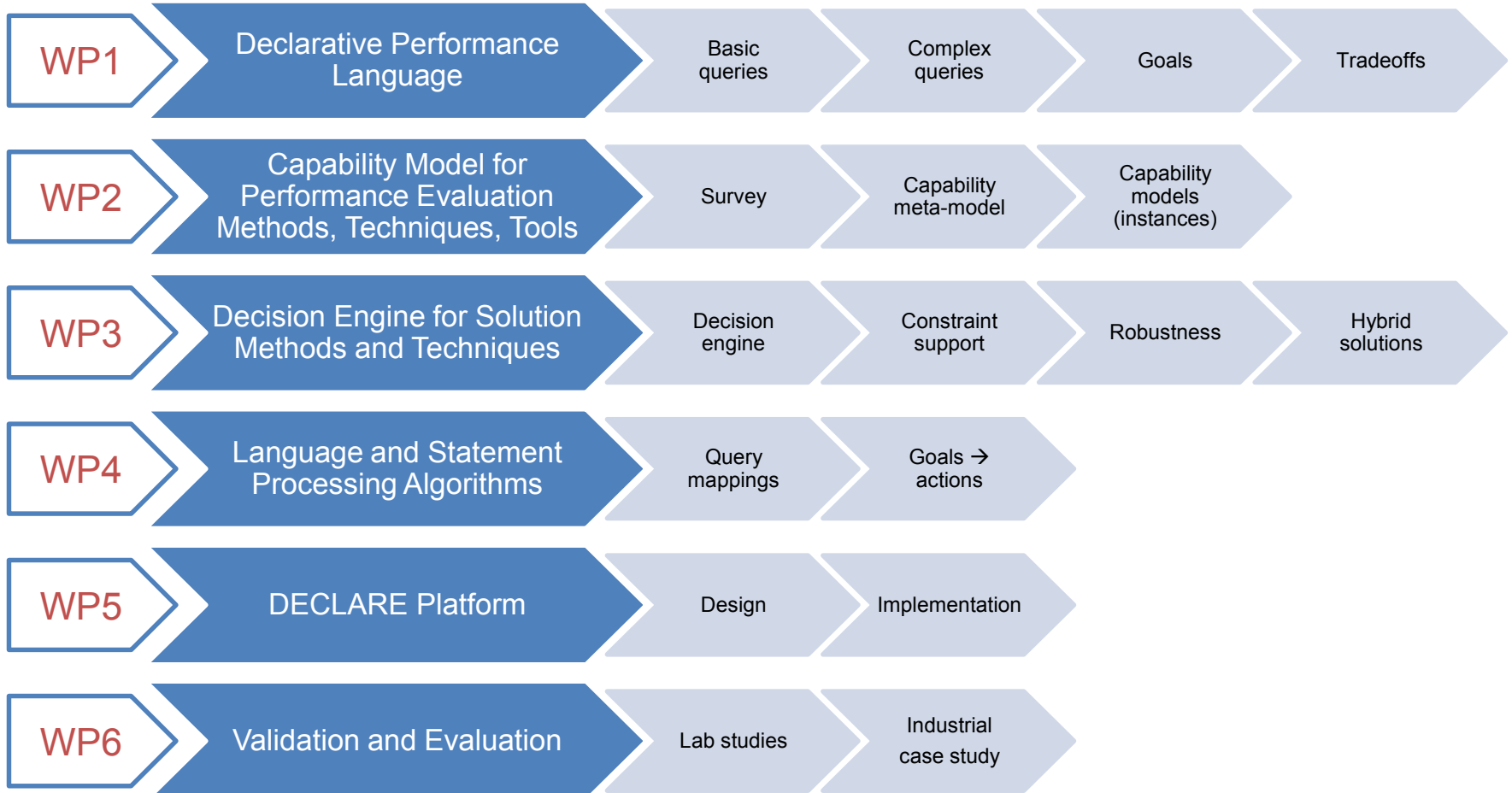
University of Stuttgart
Germany

Collaborators

- Research Group of the Standard Performance Evaluation Corporation (SPEC RG)









A **Scenario Model Builder** creates a scenario model that depicts a real world system.



<http://descartes.tools/pmX>

**CAN BE DONE MANUALLY OR
AUTOMATE USING**

1. $f (M, D, G) \rightarrow M'$ components + deployment
 2. $f' (M', D, G) \rightarrow M''$ control flow/interaction
 3. $f'' (M'', D, G) \rightarrow M'''$ resource demands
 4. $f''' (M''', G) \rightarrow$ target model
- $M \subset M' \subset M'' \subset M''' = PCM/DML$
 - D ... trace data / code /resource utilization
 - G ... goal specification (currently fixed and DQL in the future)

Declarative Performance Engineering

I say/define **what** i want to know,
the **how** will be automatically derived from what