Coordinated Run-time Adaptation of Variability-intensive Systems

An Application in Cloud Computing

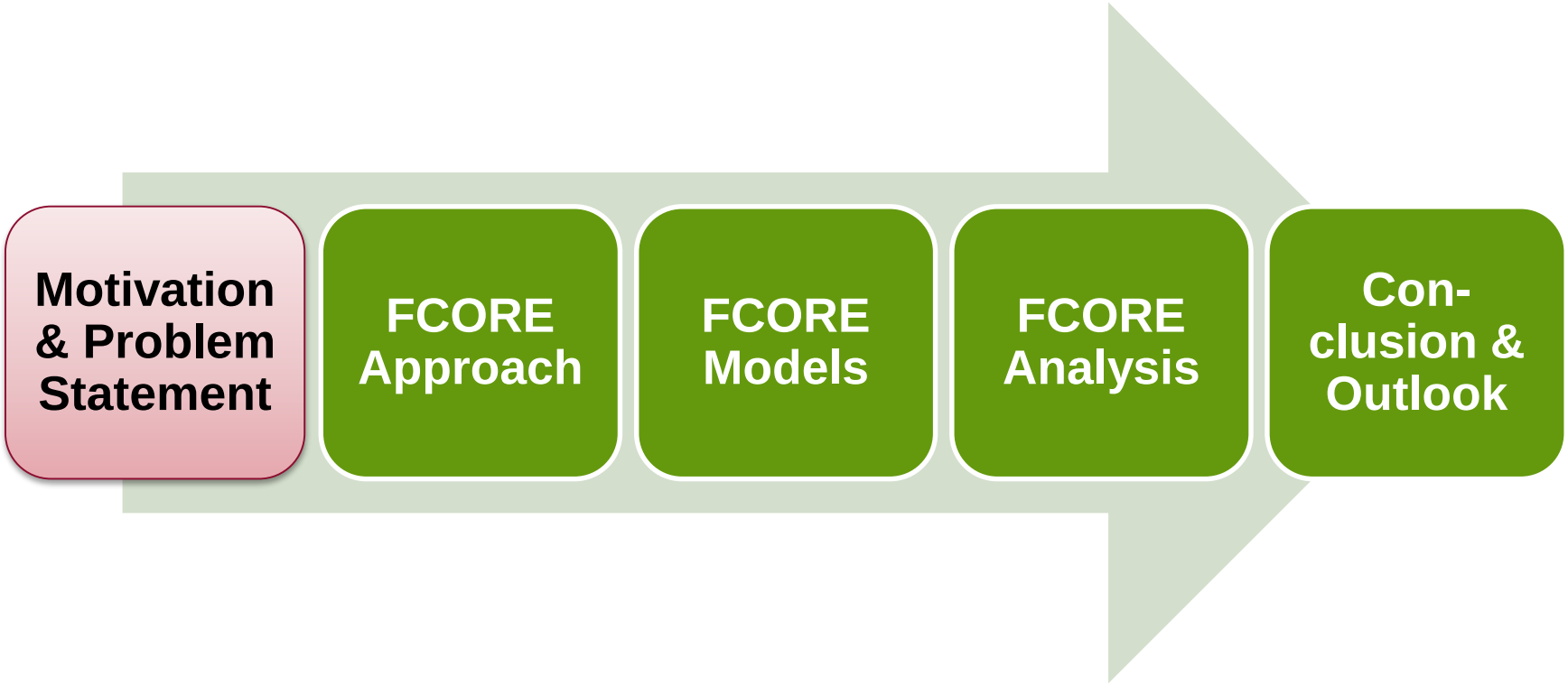
Andreas Metzger†, Andreas Bayer†, Daniel Doyle*, Amir Molzam Sharifloo†, Klaus Pohl†, Florian Wessling†

† paluno (The Ruhr Institute for Software Technology), University of Duisburg Essen, Germany

* Intel, Ireland

paluno
The Ruhr Institute for Software Technology

UNIVERSITÄT
DUISBURG
ESSEN


Motivation & Problem Statement
FCORE Approach
FCORE Models
FCORE Analysis
Con-clusion & Outlook

Motivation
Adaptive Software Systems

$$M, W \models R ?$$

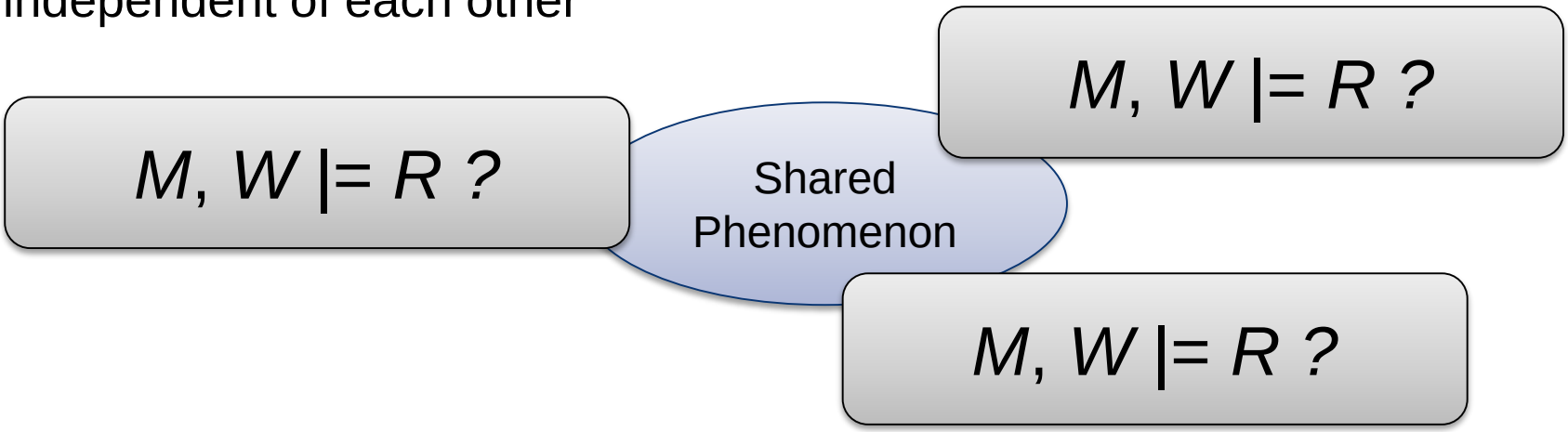
- Adaptive software systems can modify own structure and behavior at run time to cope with dynamic changes in…
  - *M* = machine (software) → self-healing
  - *W* = world (context) → context-aware
  - *R* = changing requirements → ???

- @runtime: adaptive systems monitor changes in *M* and *W* or even *R* directly
  - $M, W \not\models R$ (requirements violation) → self-modification

Motivation

Coordination among Adaptive Systems

- Distributed systems (e.g., cloud systems or cyber-physical systems) orchestrate many adaptive sub-systems
- Each sub-system may perform adaptations simultaneously and independent of each other

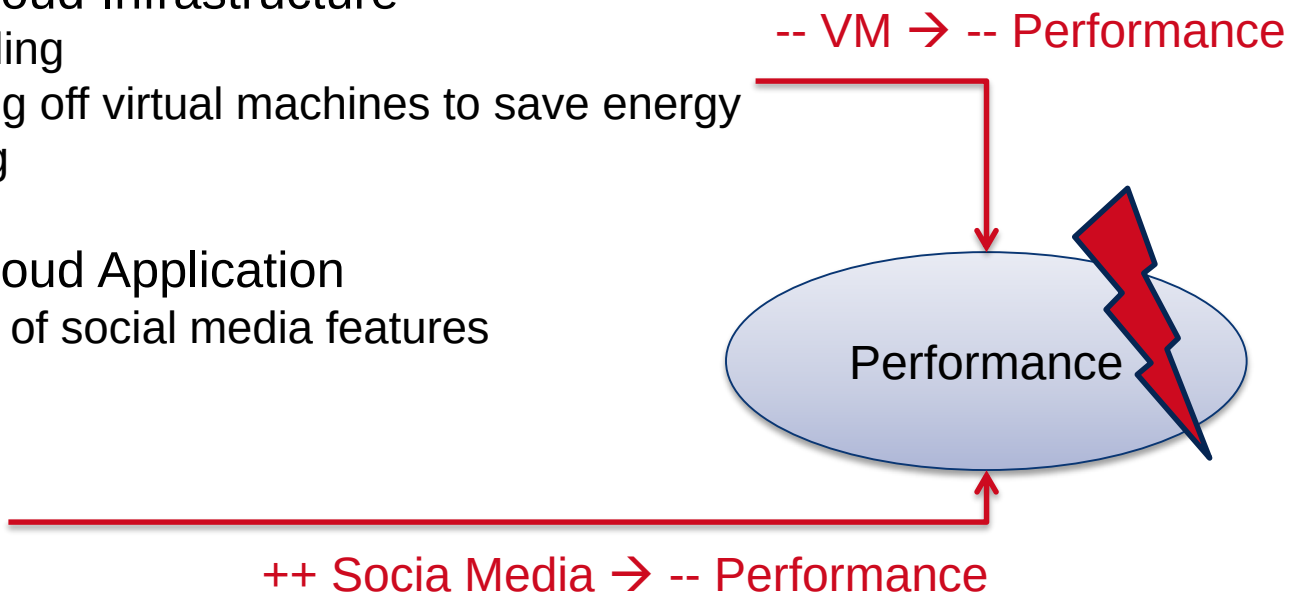


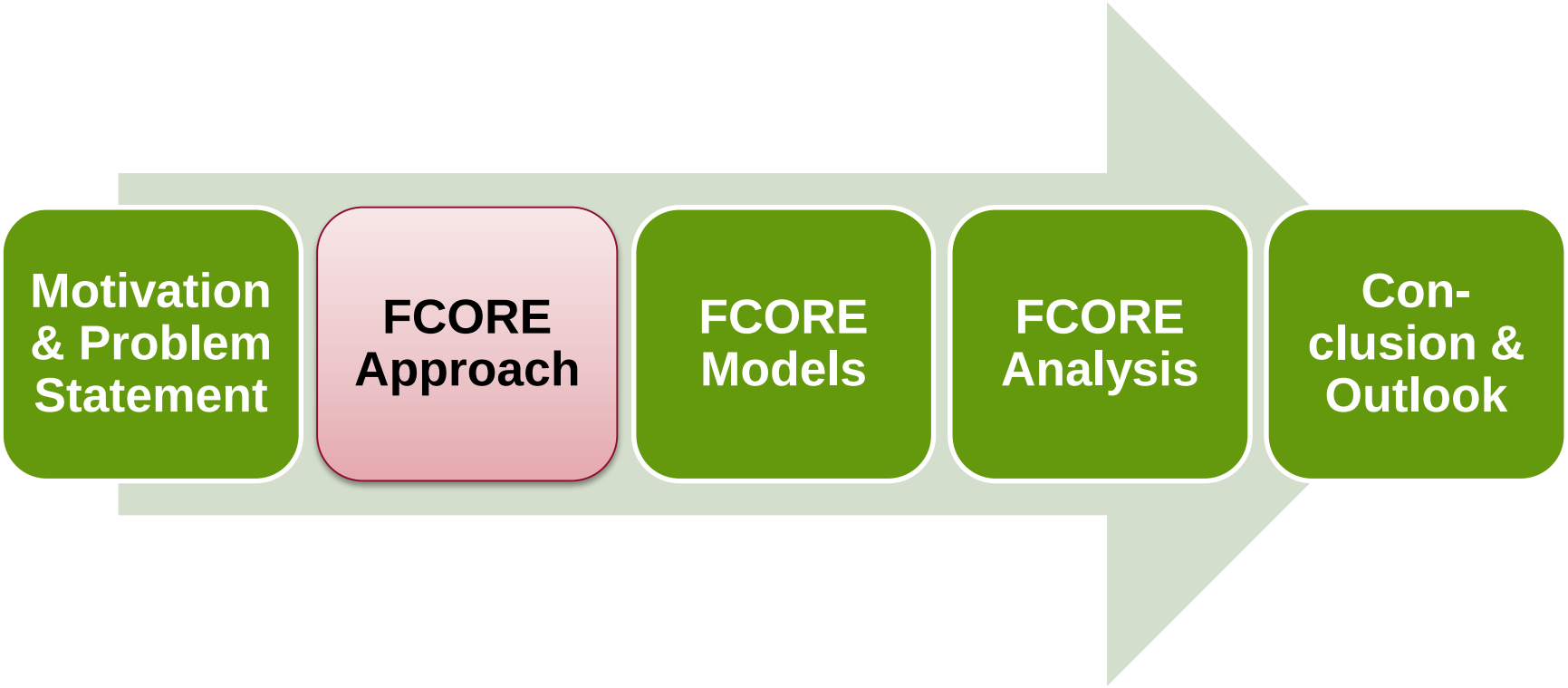
- However, adaptations may affect shared phenomena, thus:
  - **Conflicts** between adaptations may occur
  - **Synergies** among adaptations may be missed

Motivation

Use Case: Conflicts in Adaptive Cloud Systems

- „AdvancedTV“ Use Case from EU Project Cloud Wave
  - Cloud application that offers services in parallel to running TV programme
- Two adaptive systems:
  - Cloud Infrastructure (IaaS: CPU, RAM, …)
  - Cloud Application (SaaS)
- Adaptations of Cloud Infrastructure
  - Horizontal Scaling
    - E.g., turning off virtual machines to save energy
  - Vertical Scaling
  - …
- Adaptations of Cloud Application
  - Different levels of social media features
    - No
    - Partial
    - Unlimited
  - …

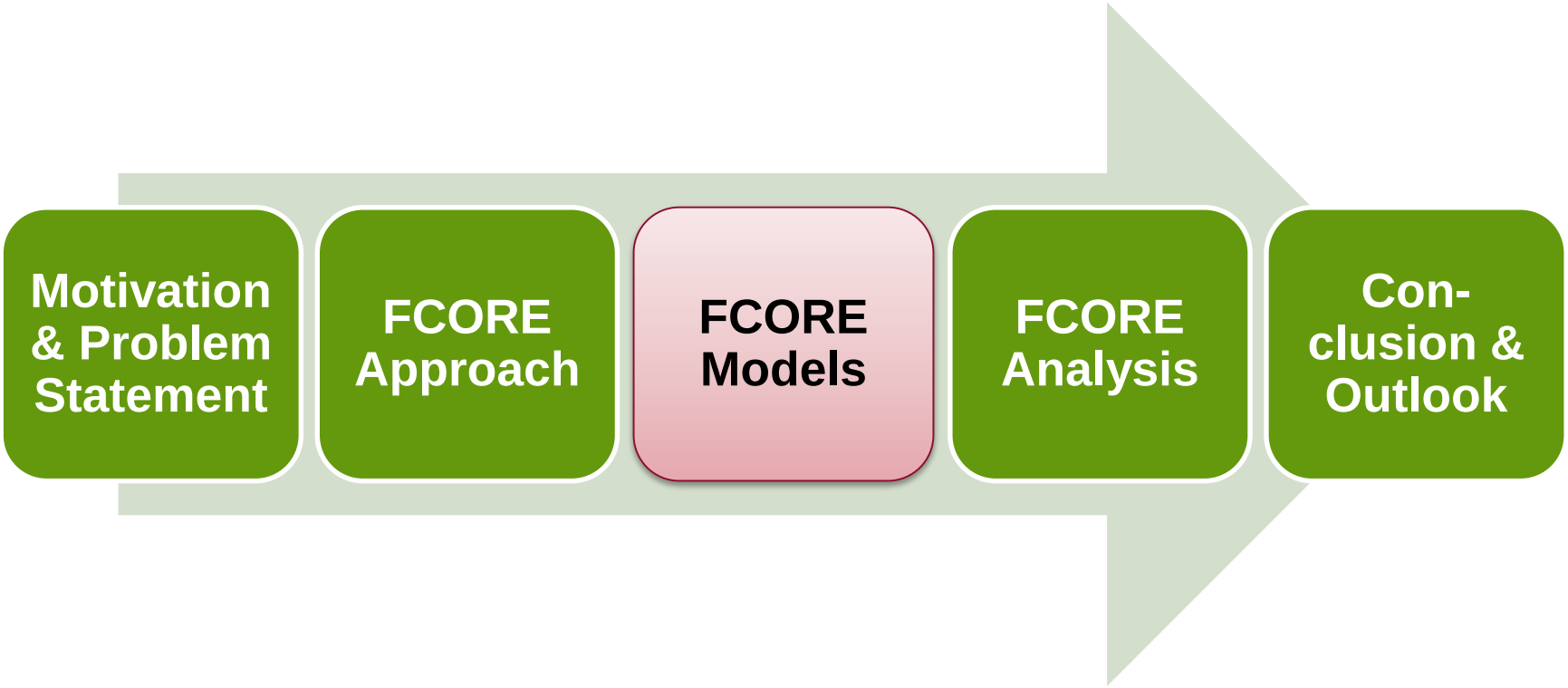



Motivation & Problem Statement
FCORE Approach
FCORE Models
FCORE Analysis
Con- clusion & Outlook

FCORE Approach
Main Ideas and Challenges

- **Explicitly model adaptations and dependencies among systems during design time**
  - Challenge 1: Developers must model adaptations of their systems
    → **Sufficiently compact, yet expressive modeling technique**
  - Challenge 2: Systems developed by different developers/organizations
    → **Suitably (small) common denominator to describe dependencies among systems**

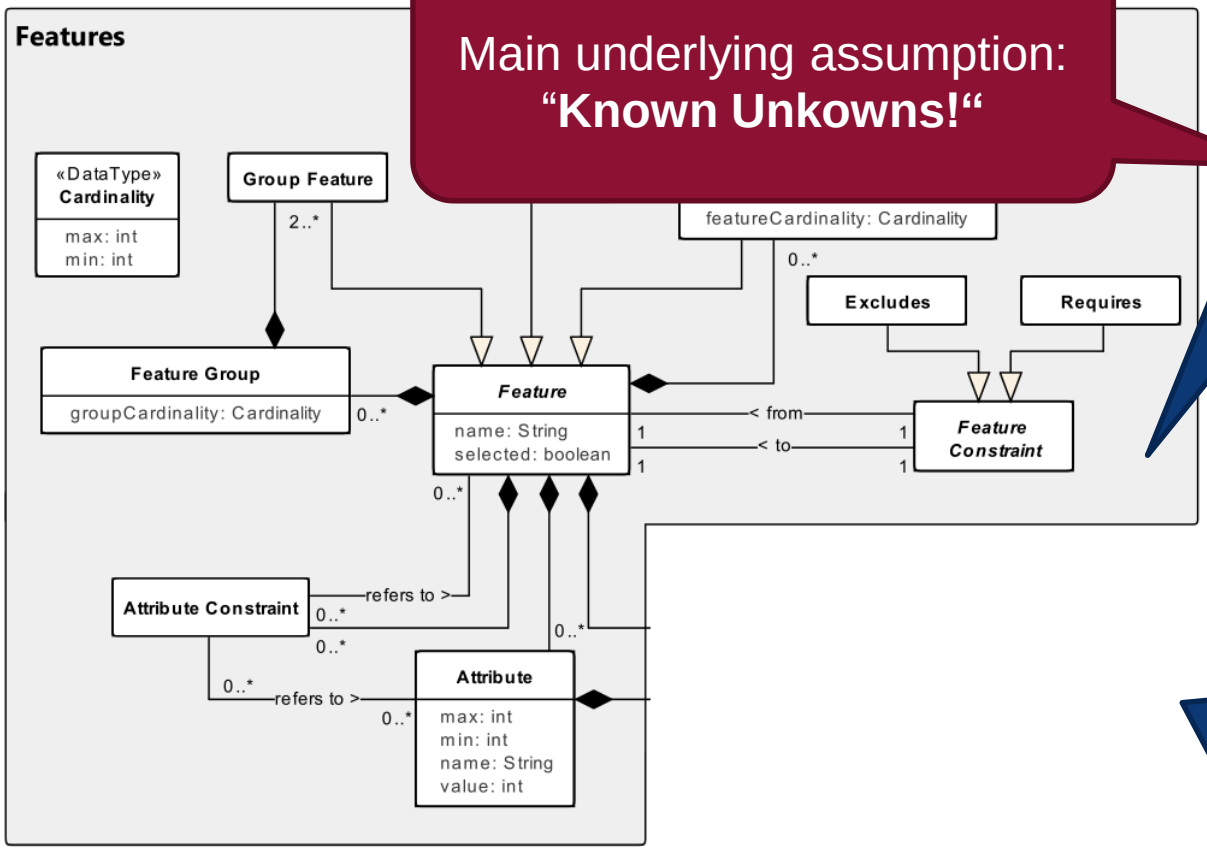
- **Analyze models at run time to determine conflicts and identify optimizations (synergies)**
  - Challenge 3: Self-adaptation at run time must be fast enough to be effective (otherwise may be too late)
    → **Efficient model analysis during system execution**


Motivation & Problem Statement
FCORE Approach
FCORE Models
FCORE Analysis
Con-clusion & Outlook

FCORE Models

Main Underlying Concepts

- **FCORE = DSPL Feature Model + Goal Models**



Main underlying assumption:
"Known Unkowns!"

Concepts from Dynamic Software Product Lines Feature Models to describe **adaptations**

Concepts from Goal Models to describe **dependencies** via shared phenomena

FCORE Models

Why DSPL Models? (Challenge 1)

- **DSPLs can build on proven engineering foundations of SPLs!**

- DSPL extend existing software product line engineering approaches by moving their capabilities to run time
- Variability binding is postponed to run time, allowing a DSPL to activate or deactivate certain features
- Configurations of a DSPL are expressed in terms of a product line variability model, usually a feature model

Classical SPL	Dynamic SPL
variability describes different pos- sible software systems	variability describes different possible configurations (i.e., adaptations) of the same system

FCORE Models
Which Kinds of DSPL Models? (Challenges 1&3)

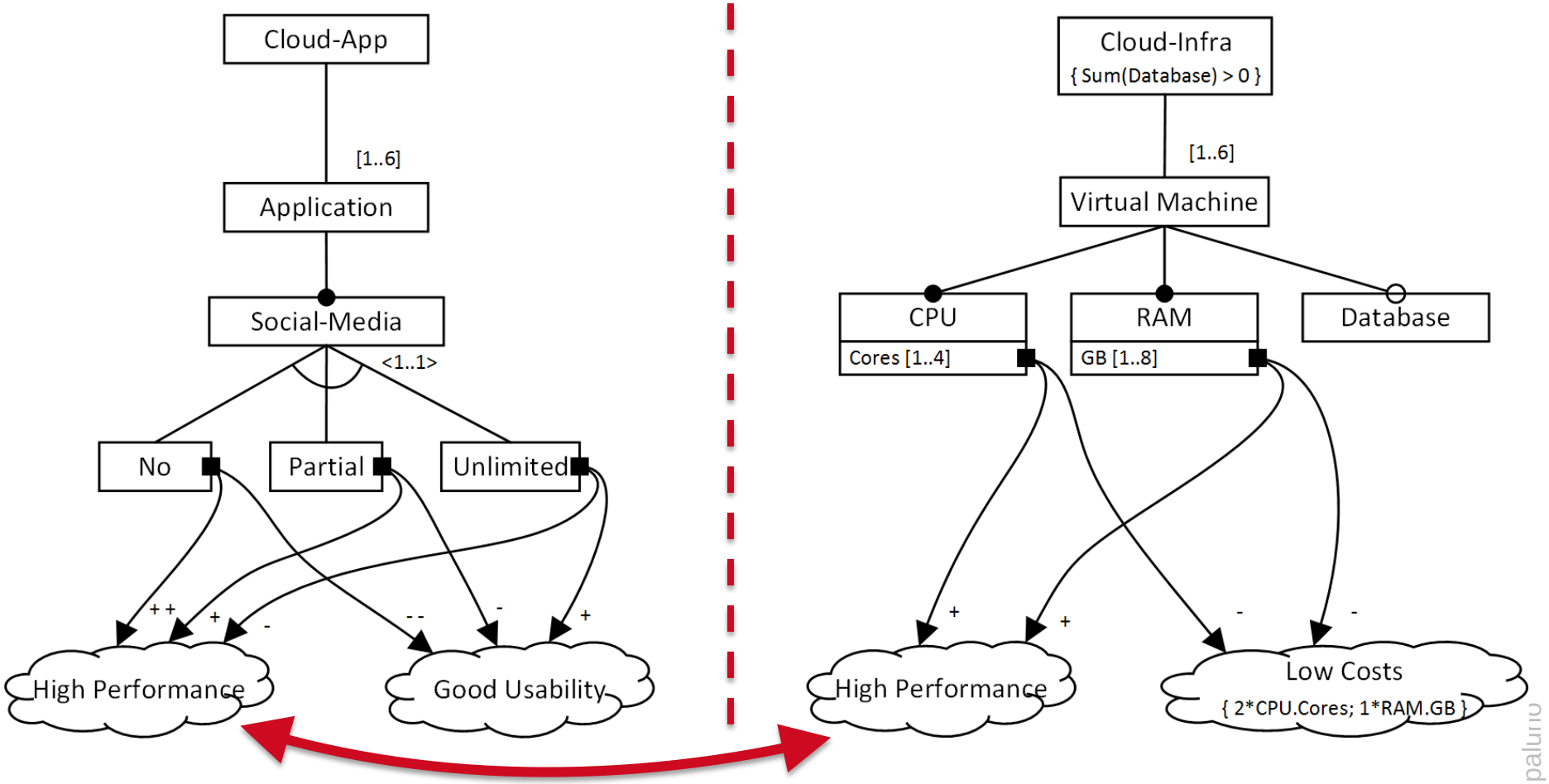
Approach	Expressiveness	Analysis
Basic-FM	High redundancy in models (replication of FM sub-trees) Cardinalities only 1..1 / 0..n	SAT solver
Cardinality-Based FM	Alternative-Groups Cardinalities n..m Feature-Cardinalities n..m (i.e., instantiation of features)	SAT solver
Extended-FM	Feature-Attributes (Integer, Enumeration, …)	CSP solver

FCORE Models
Why Goal Models? (Challenge 2)

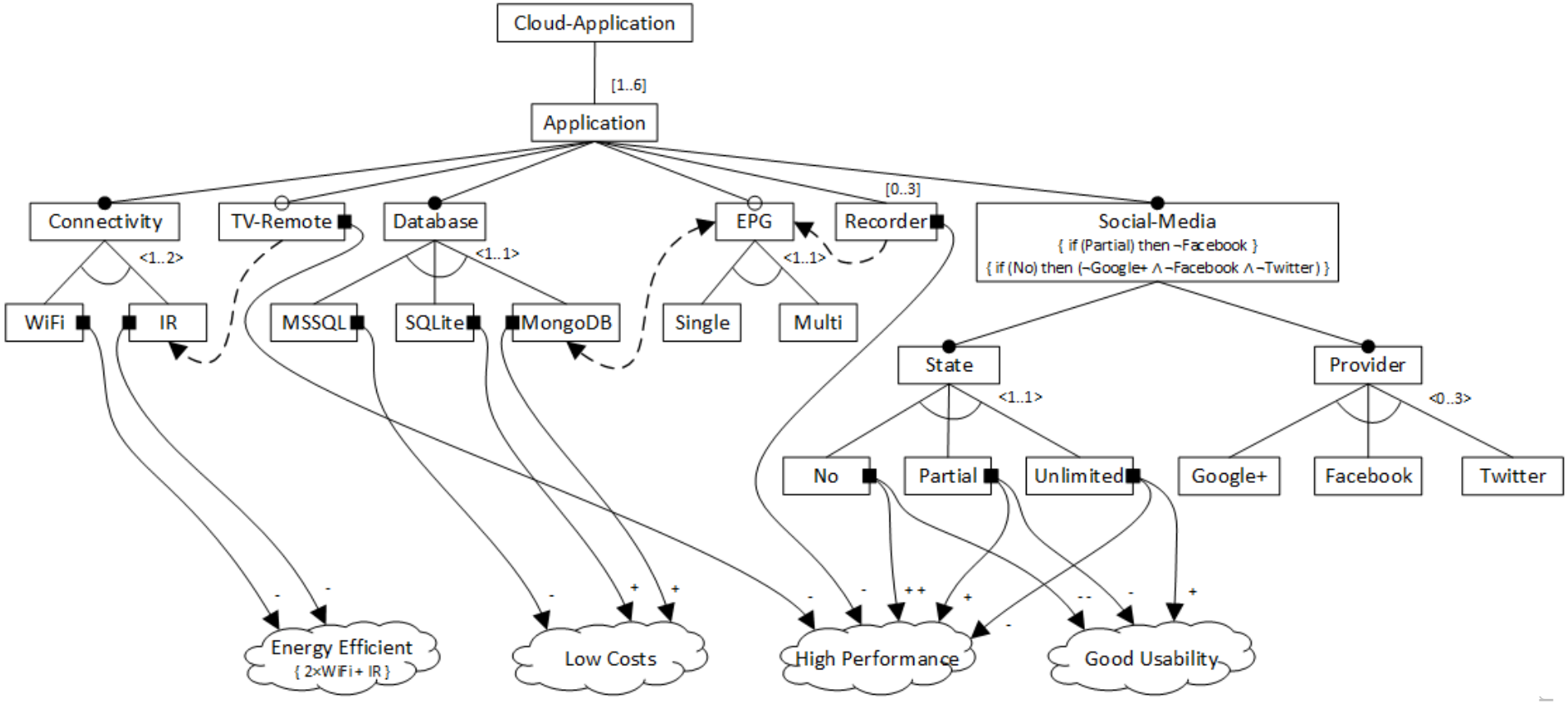
- **Soft Goals provide high-level of abstraction to describe influences of features (~ “tasks”) on goal satisfaction**

- Well-known from requirements engineering
- Defining dependencies among systems requires **agreeing on a set of shared soft goals**
  - E.g., in cloud computing, these soft goals may be derived from standardized QoS catalogues for SLAs

FCORE Models
CloudWave Use Case: Simplified

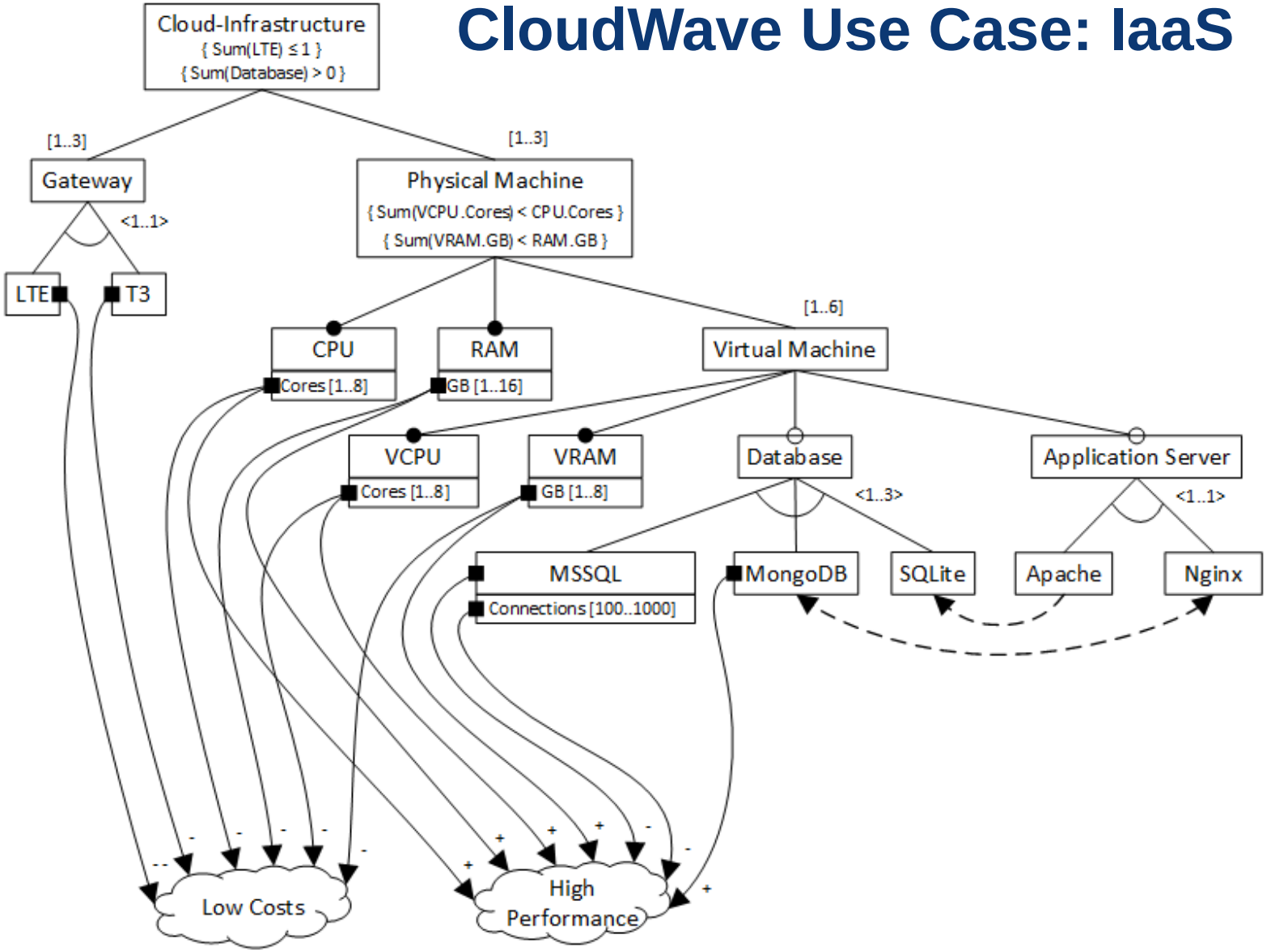


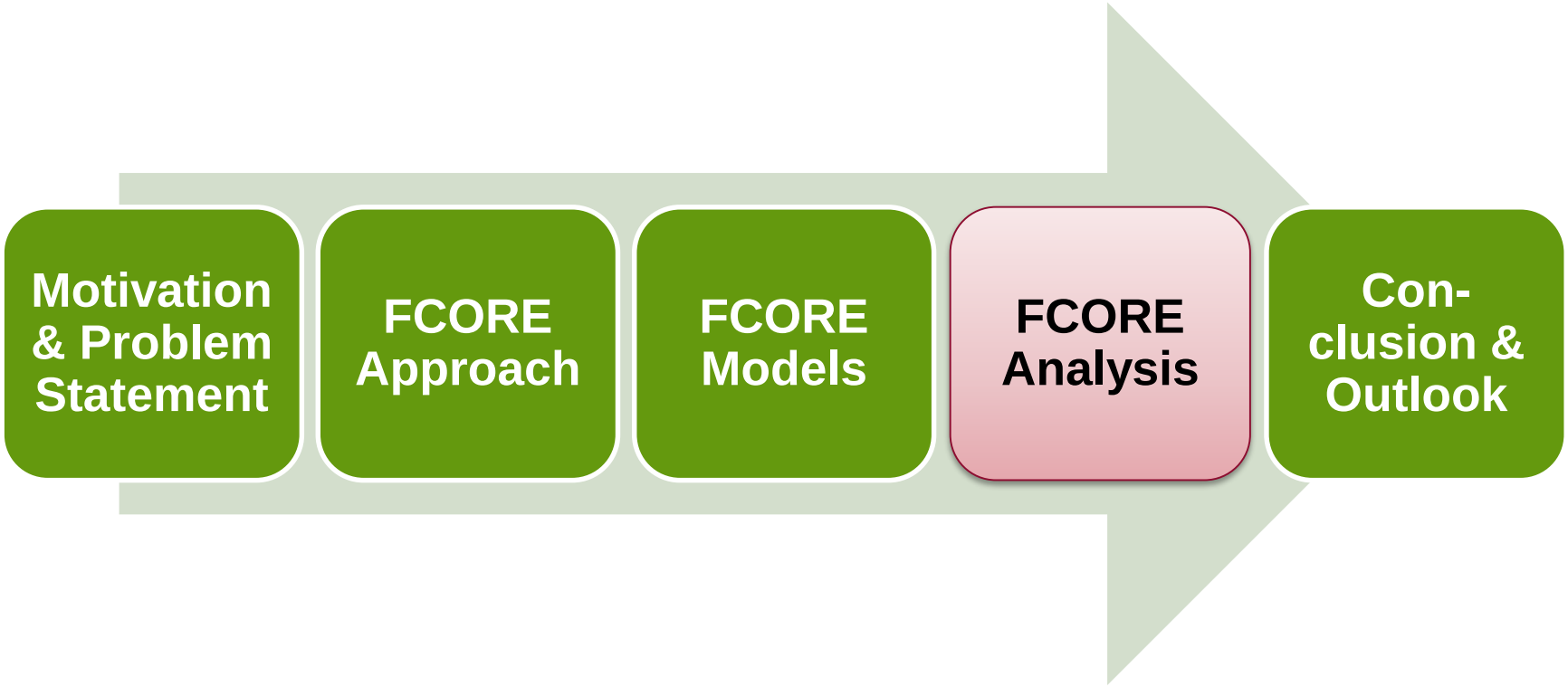
FCORE Models
CloudWave Use Case: SaaS



FCORE Models

CloudWave Use Case: IaaS




Motivation & Problem Statement
FCORE Approach
FCORE Models
FCORE Analysis
Con-clusion & Outlook

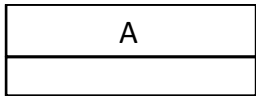
FCORE Analysis
Main Underlying Strategy (Challenge 3)

- Formalize FCORE Model as CSP (justification see above)
- Perform automated reasoning on formalization
- Two main usages:
  - **FCORE Filter:** Validity check of given configurations (= detecting conflicts)
    - E.g., 1 CPU + Unlimited Social Media → violation of high performance
  - **FCORE Search:** Search for configurations with high goal satisfaction (= exploiting synergies)
    - E.g., 6 CPUs + No Social Media → high performance + low costs

FCORE Analysis
Formalization: Features

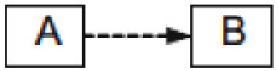
= Feature selected

- Feature



$$A \in \{0, 1\}$$

- Requires-Relation

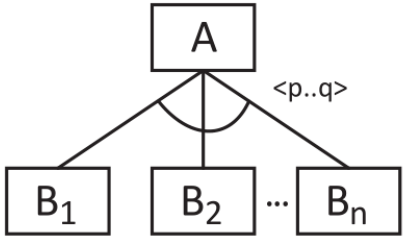


$$\text{if } (B = 0)\ A = 0$$

- Excludes-Relation

$$\text{if } (A = 1)\ B = 0$$

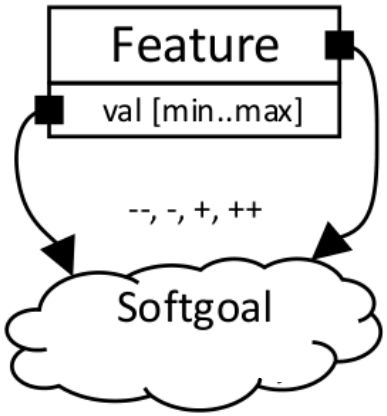
- Feature Group



$$\text{if } (A = 0) \quad \sum_{x=1}^{n} B_x = 0$$
$$\text{else} \quad p \leq \sum_{x=1}^{n} B_x \leq q$$

FCORE Analysis
Formalization: Goals

- Softgoals and Attributes



$$contribution = \text{“+”}: \quad v'_x := \frac{v_x - min + 1}{max - min + 1} \quad v'_x := 0, \text{ if } Feature = 0$$

$$contribution = \text{“−”}: \quad v'_x := 1 - \frac{v_x - min + 1}{max - min + 1} \quad v'_x := 1, \text{ if } Feature = 0$$

$$sgVal := \frac{k_1 \cdot v'_1 + k_2 \cdot v'_2 + \ldots + k_n \cdot v'_n}{\sum_{x=1}^{n}(k_x)}$$

Softgoal satisfaction:
sgVal = [-1.0, +1.0]

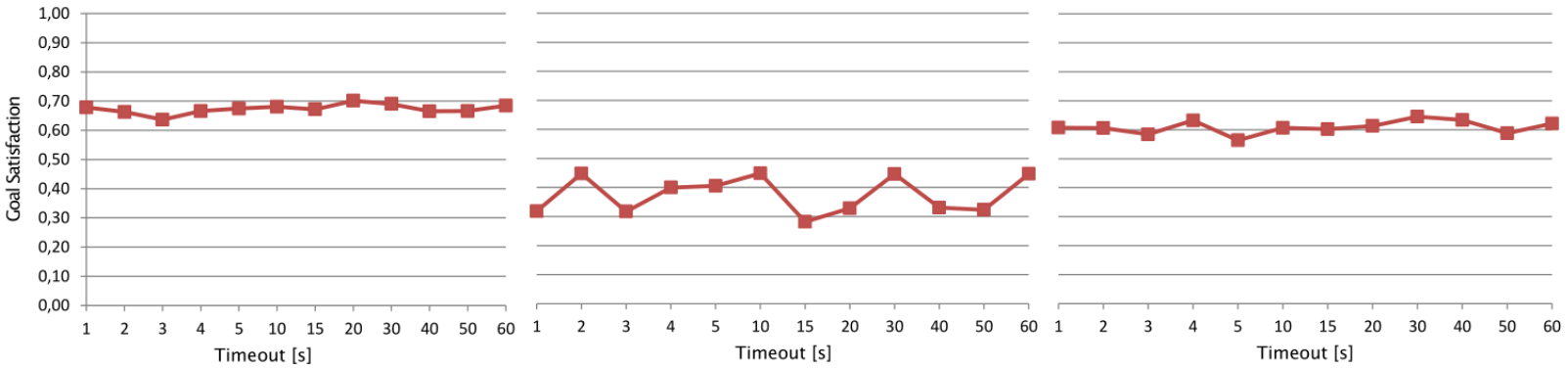
$$contribution = ++ \Rightarrow sgVal := +1.0$$

$$contribution = -- \Rightarrow sgVal := -1.0$$

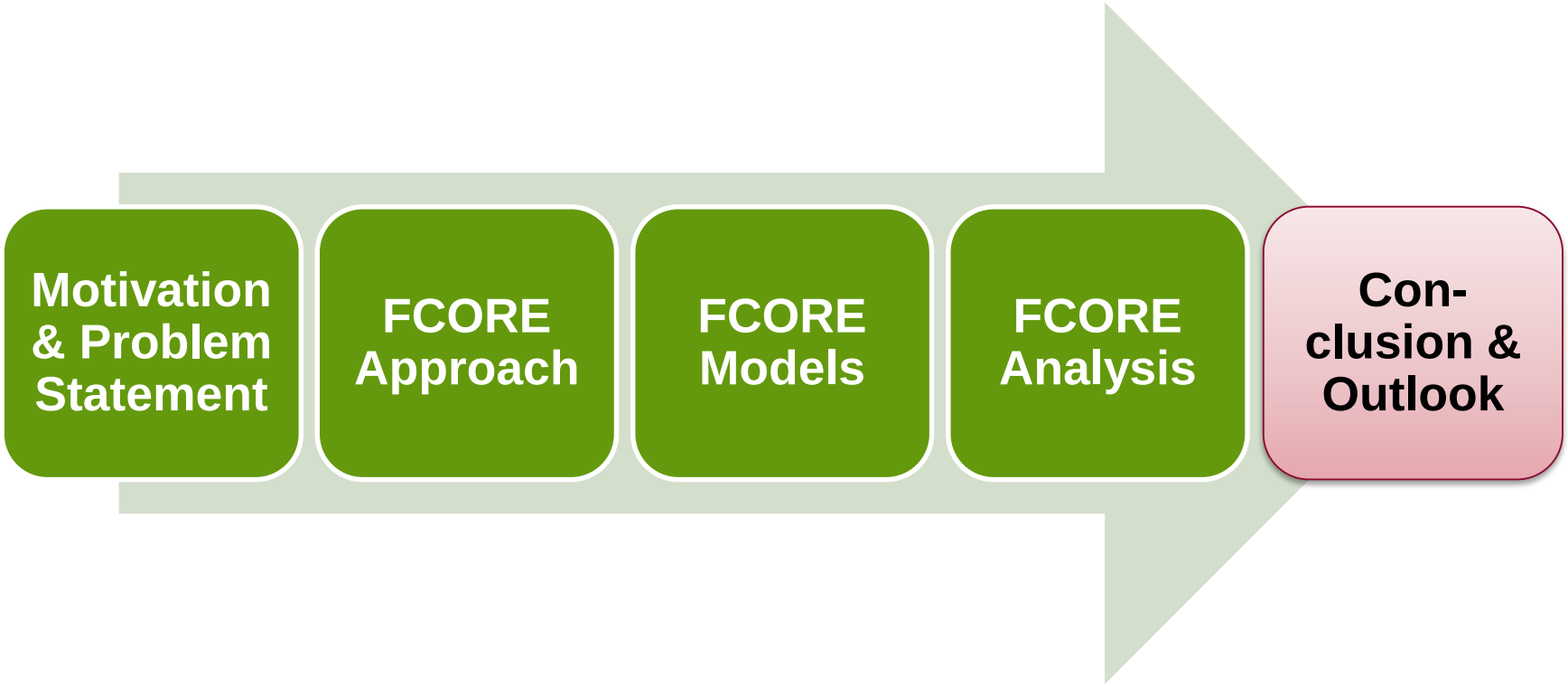
FCORE Analysis
Performance (Challenge 3)

- **FCORE Filter**
  - No performance issues
  - Just compute goal satisfaction for given configuration
  - Ca. 2ms for cloud use case

- **FCORE Search**
  - CSP to find optimal configurations (maximize *sgVal*)
  - Experimental results for cloud use case



(a) SaaS model (b) IaaS model (c) Combined model

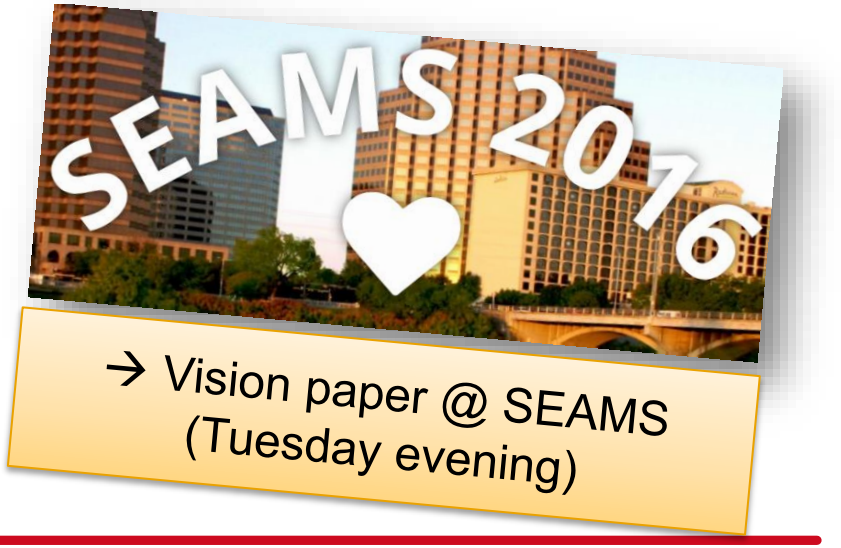

Motivation & Problem Statement
FCORE Approach
FCORE Models
FCORE Analysis
Con-clusion & Outlook

Conclusion and Outlook

- **Concluded:** FCORE as an approach for coordinating among adaptive, variability intensive systems
  - Building on DSPLs
  - Offering Modelling + Analysis
  - Exemplified for the case of cloud computing

- **Ongoing:**
  - Implementation as part of CloudWave Adaptation Engine (jointly with IBM and intel)

- **Future:**
  - Handling "**Unknown Unknowns"**: Extending DSPLs with dynamic **learning** and **evolution**



Agile Service Engineering for the Future Internet

The research leading to these results has received funding from the European Union's Seventh Framework Programme FP7/2007-2013 under grant agreement 610802 (CloudWave)

http://www.cloudwave-fp7.eu/

Thank You!