

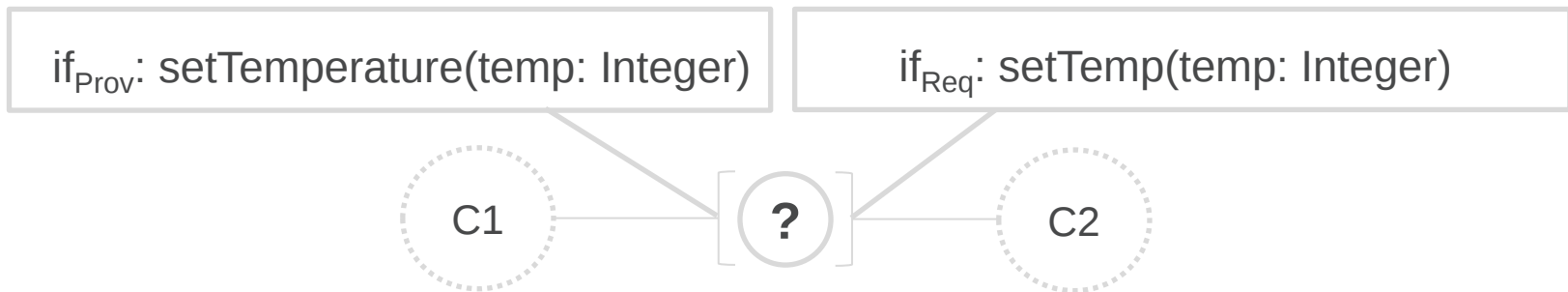


Knowledge-driven Architecture Composition

Case-based Formalization of Integration Knowledge to Enable
Automated Component Coupling

Fabian Burzlaff and Christian Bartelt – University of Mannheim (Germany)

1. Component Composition is a mature reserach area



Preventing Mismatch

Using a specialized
Architecture Framework
e.g. AutoSar

Detecting Mismatch

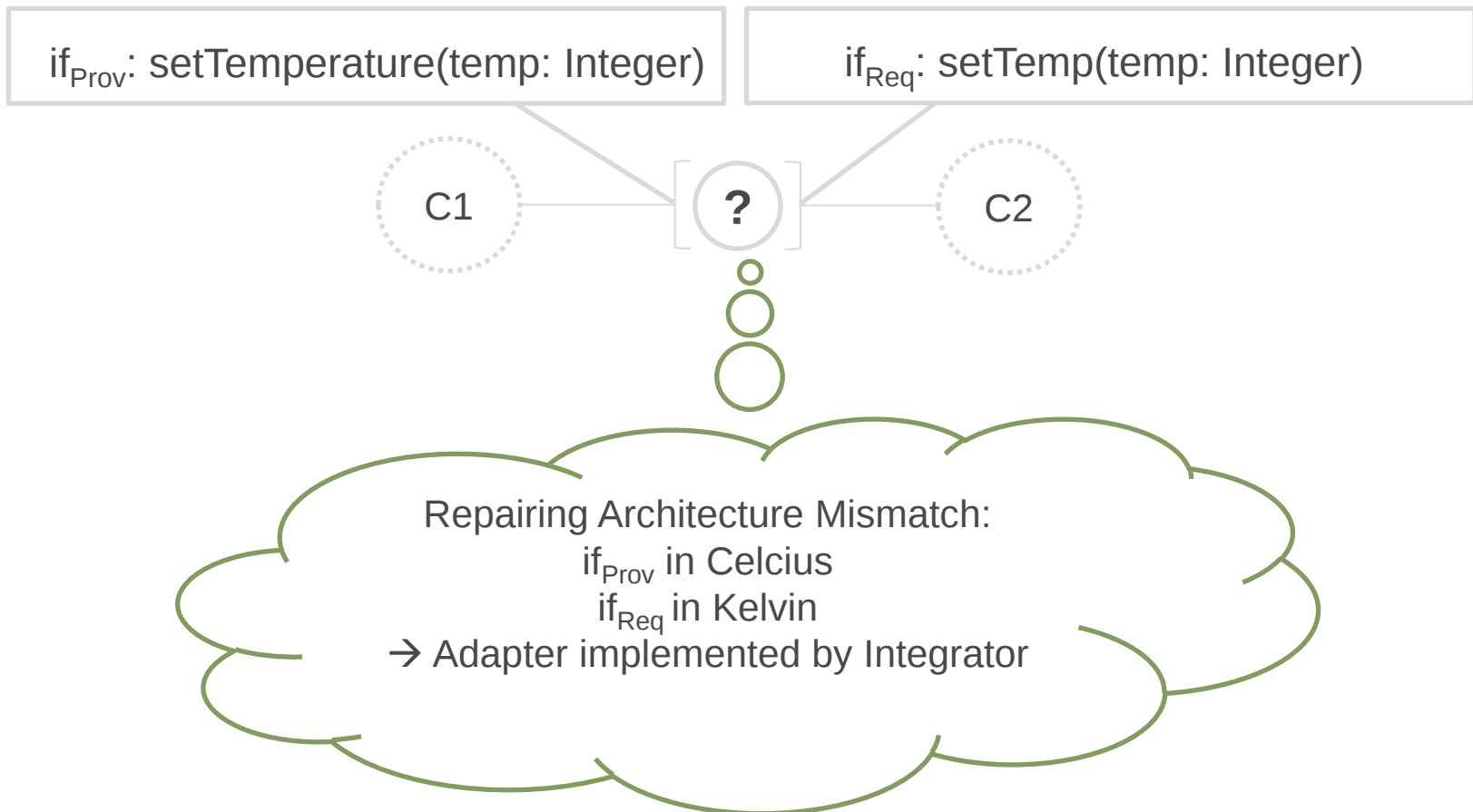
Using Interface
Description Languages
e.g. WSDL

Repairing Mismatch

Using Integration
mechanisms
e.g. individual Adapters

If components should be composed in an automated way, semantics must be formally described completely for each possible use case

1. Component Composition is a mature reserach area



2. Using formal semantic interface specifications for automated component composition

Domain Standard

1. Standardization board must agree on content
2. Component providers use this standard correctly
3. System operator can produce in a highly flexible way (Industrial IoT)



Automated Component Composition



Long standardization process

2. Using formal semantic interface specifications for automated component composition

Domain Standard

1. Standardization board must agree on content
2. Component providers use this standard correctly
3. System operator can produce in a highly flexible way (Industrial IoT)



Automated Component Composition



Long standardization process

No Domain Standard

1. Component provider writes formal interface spec.
2. Integrator performs adaptations
3. System operator must re-adapt systems when changing production



Fast Interface Specification



No automated Component Composition

2. Using formal semantic interface specifications for automated component composition

Domain Standard

1. Standardization board must agree on content
2. Component providers use this standard correctly
3. System operator can produce in a highly flexible way (Industrial IoT)



Automated Component Composition



Long standardization process

No Domain Standard

1. Component provider writes formal interface spec.
2. Integrator performs adaptations
3. System operator must re-adapt systems when changing production



Fast Interface Specification



No automated Component Composition

AS A CONSEQUENCE: Practicioners rely on informal semantic standardization and implement point-to-point connectors in the short-run (Hypothesis)



2. Currently, there is a dilemma between integration effort and automation based on formal specifications

Huge effort to create
complete semantic
specification

← Flexibility in Industrial IoT production scenarios →

Implementing a lot of
point-to-point
connectors

2. Currently, there is a dilemma between integration effort and automation based on formal specifications

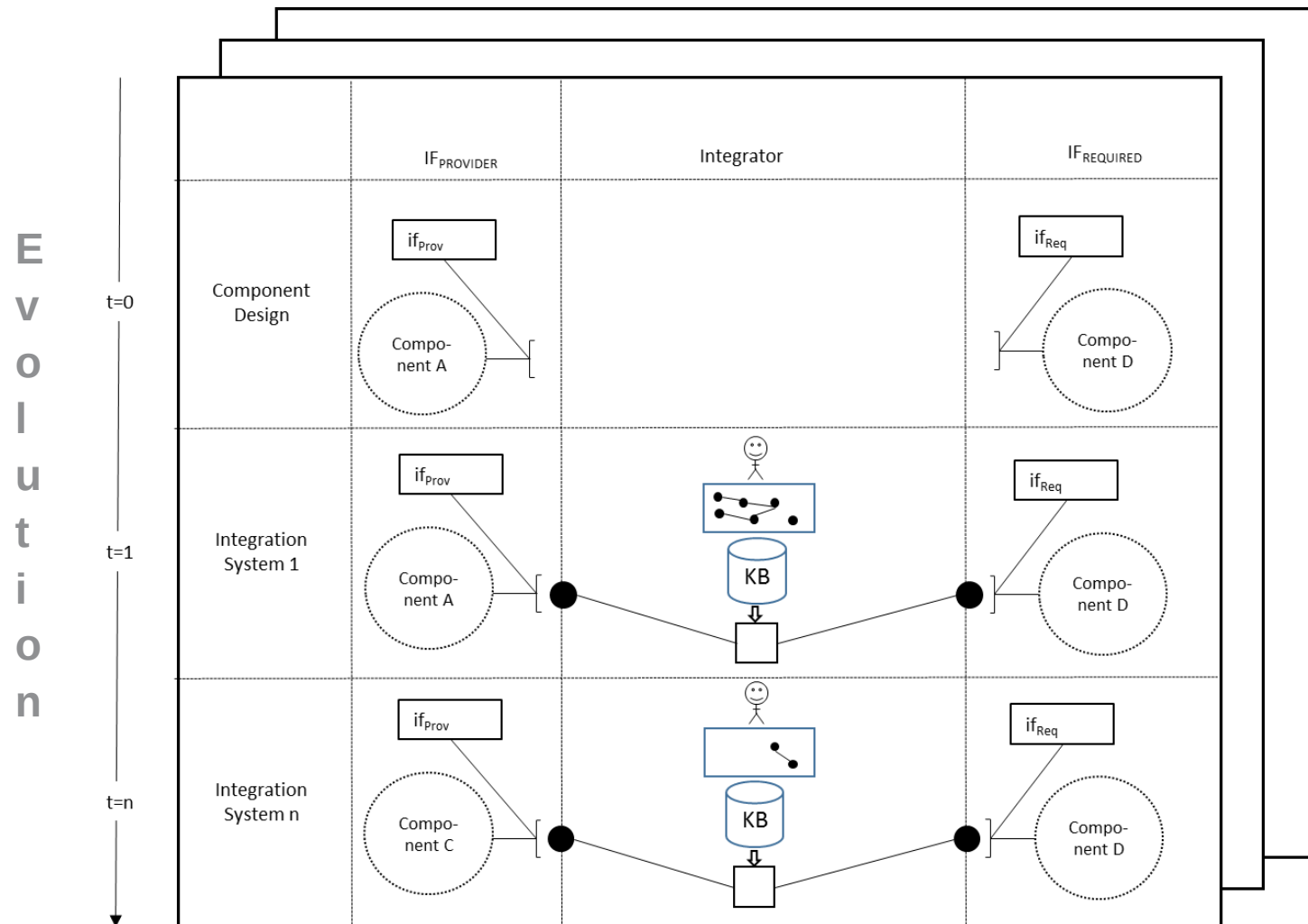
Huge effort to create
complete semantic
specification

← Flexibility in Industrial IoT production scenarios →

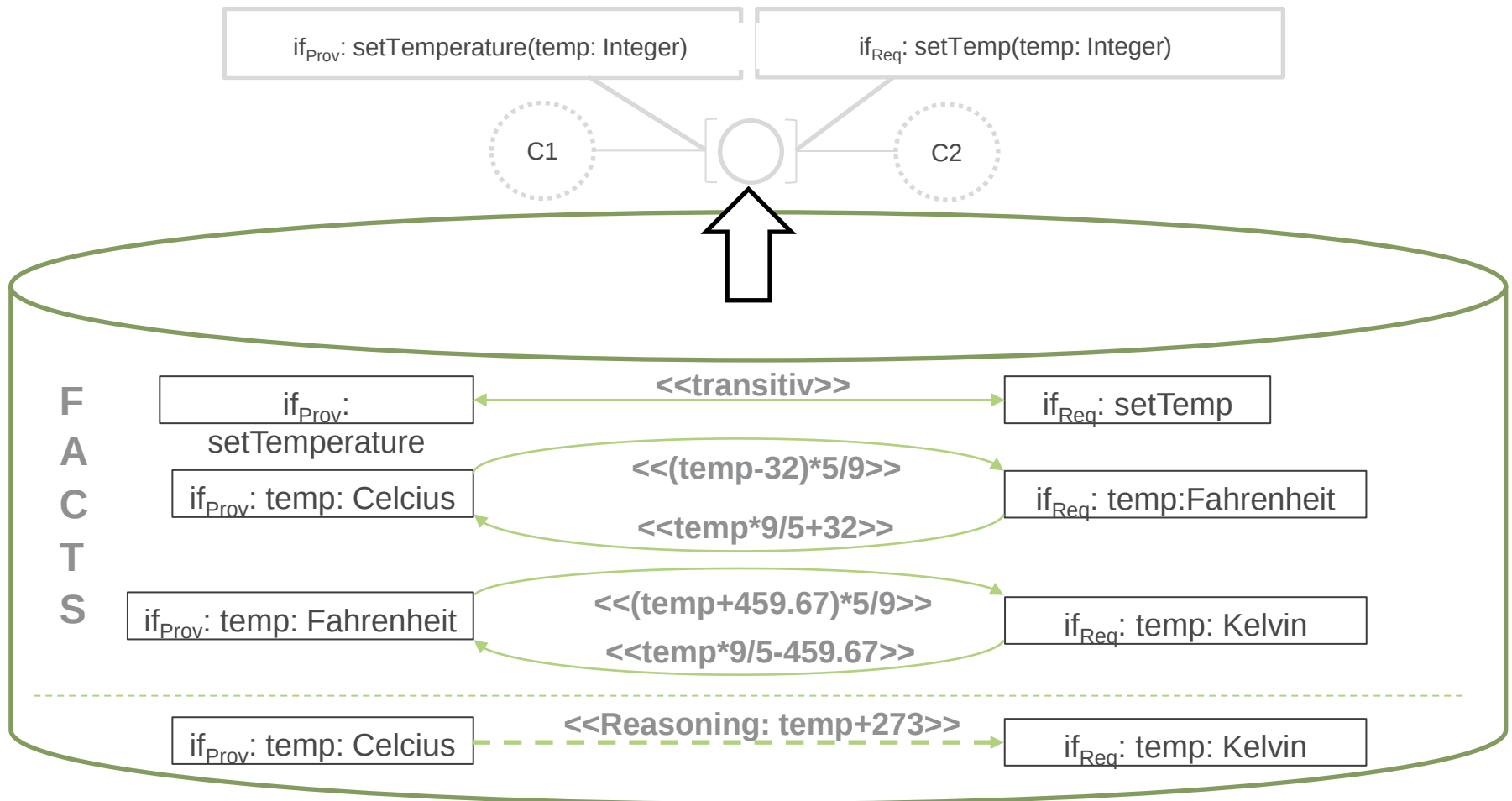
Implementing a lot of
point-to-point
connectors

How can we automate component integration tasks with only moderate formal semantic specification effort?

3. Case-based Formalization of Integration Knowledge



3. An illustrative example for formalizing integration knowledge



3. In-a-Nutshell: Knowledge-driven architecture composition

Aim

- Reduce Integration effort by case-based formalization of Integration knowledge
 - Novelty: Focus on Integrator instead of standardization process between component provider and requestor

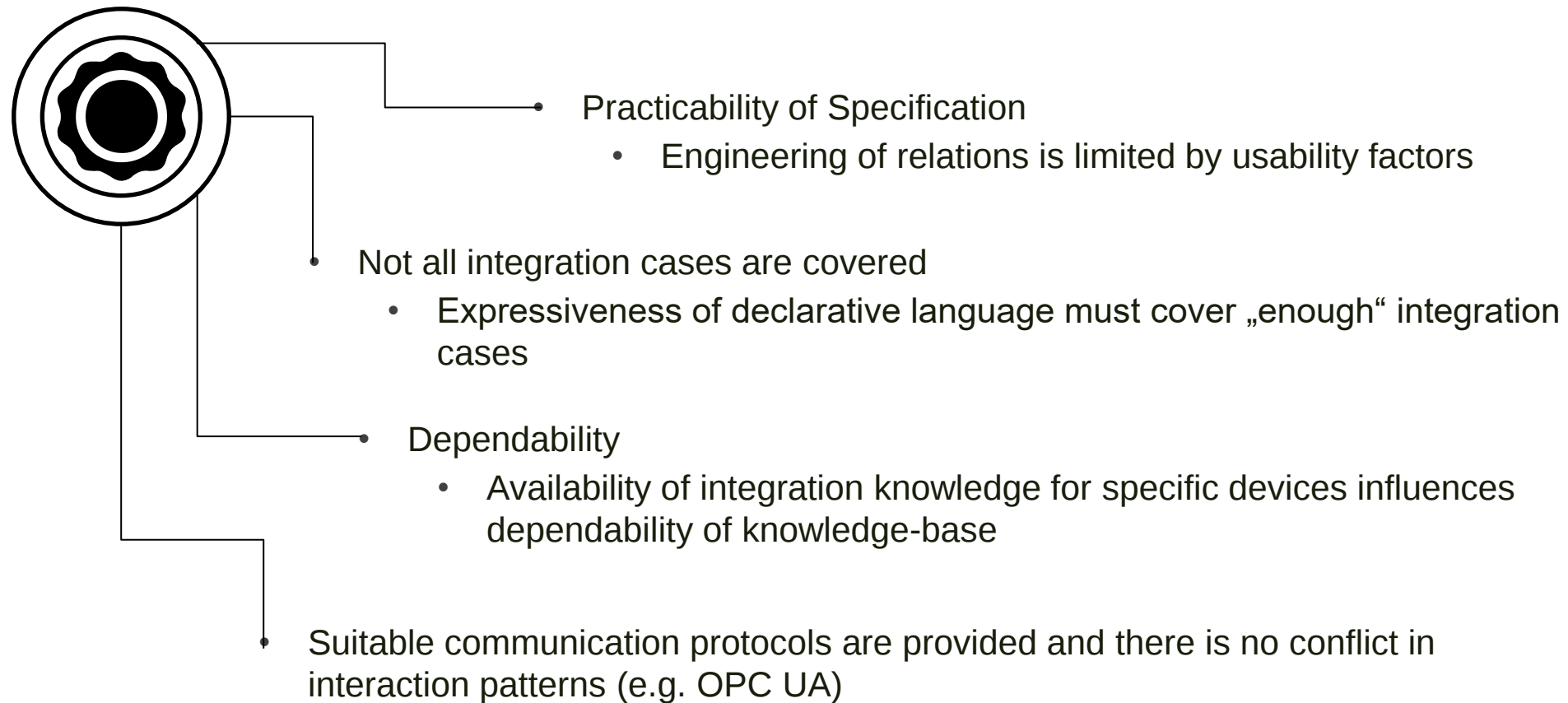
Method

- Reuseability of Component Composition Specification
 - Using a declarative languages that allows for integration knowledge management
- Need-based semantic composition specification done by integrator
 - Incomplete specification
 - Formal knowledge is only added if relevant for an integration case

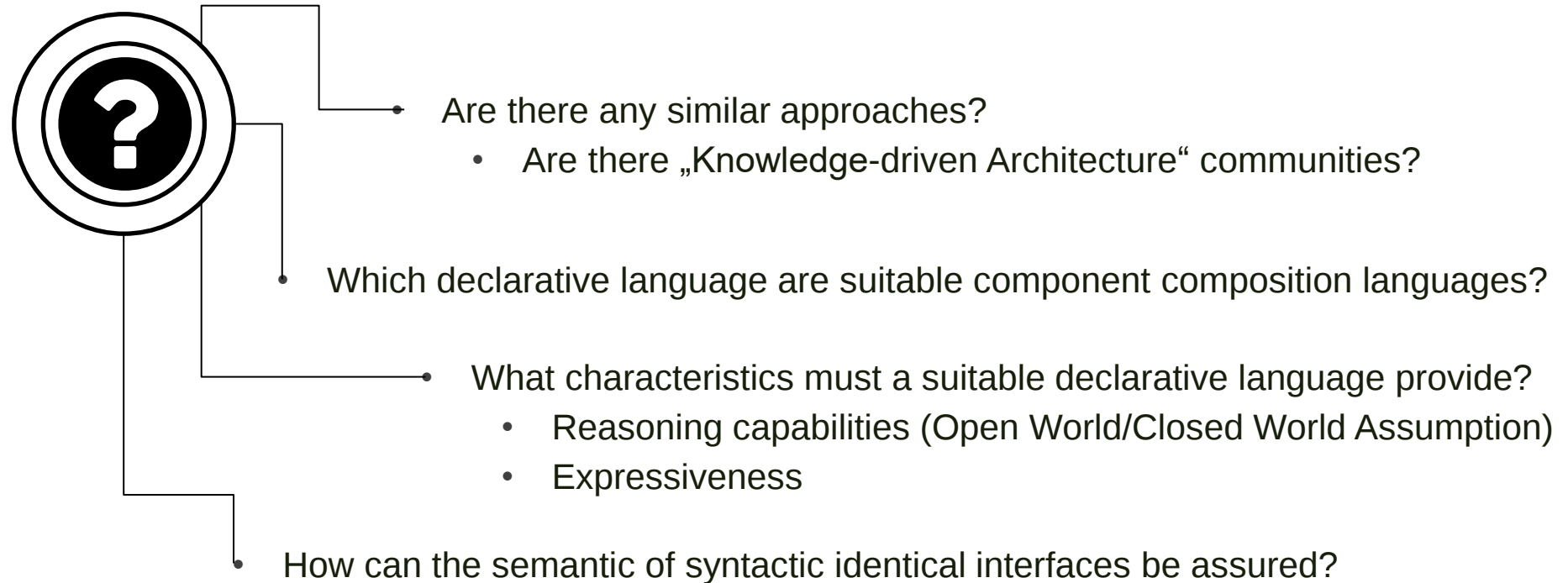
Result

- Dependable Automated Component Coupling based on formalized integration knowledge

4. Limitations of our approach



5. Open Questions





Thanks for listening!