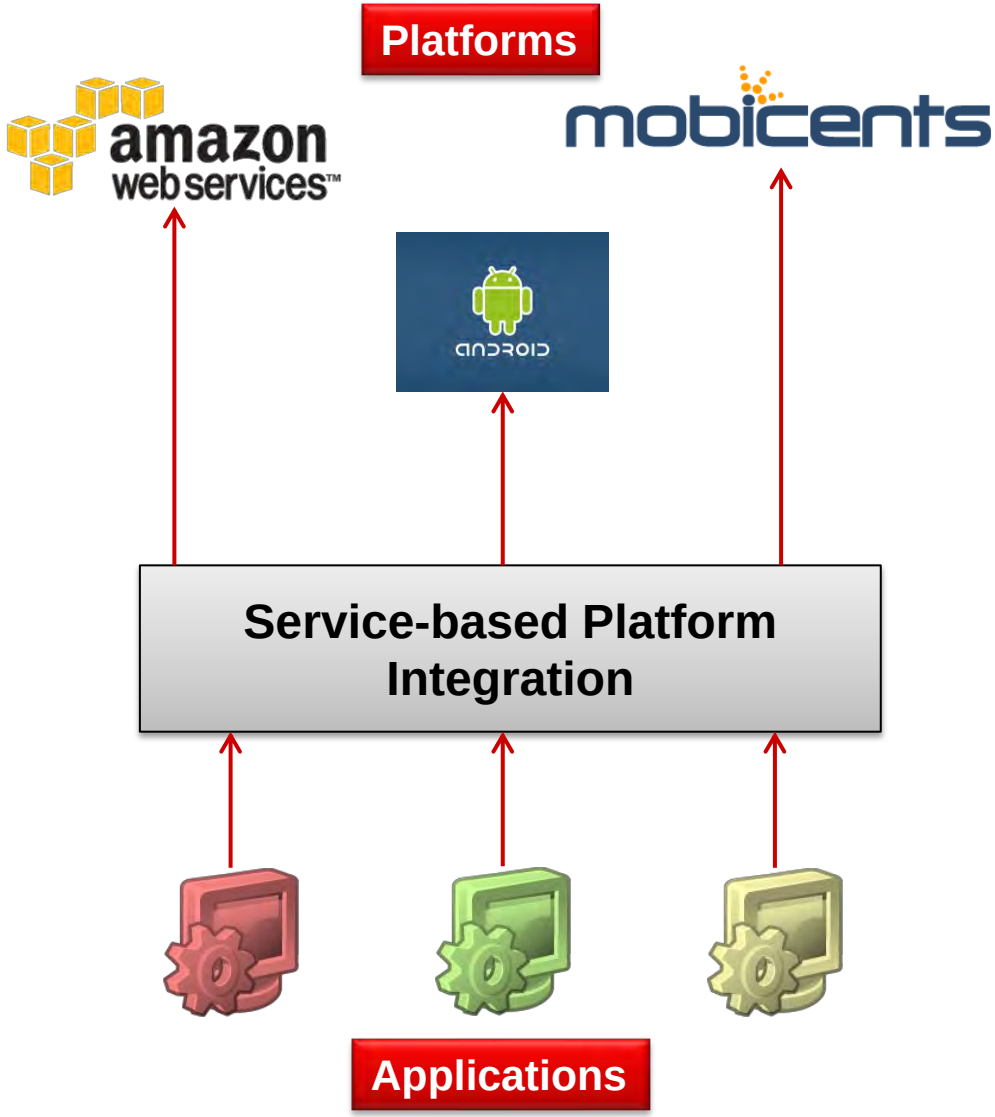
Architectural Decision Making for Service-Based Platform Integration: A Qualitative Multi-Method Study

Ioanna Lytra, Stefan Sobernig, Uwe Zdun

Faculty of Computer Science
University of Vienna, Austria

Institute for IS and New Media
WU Vienna, Austria

Architectural Decisions in Service-based Platform Integration



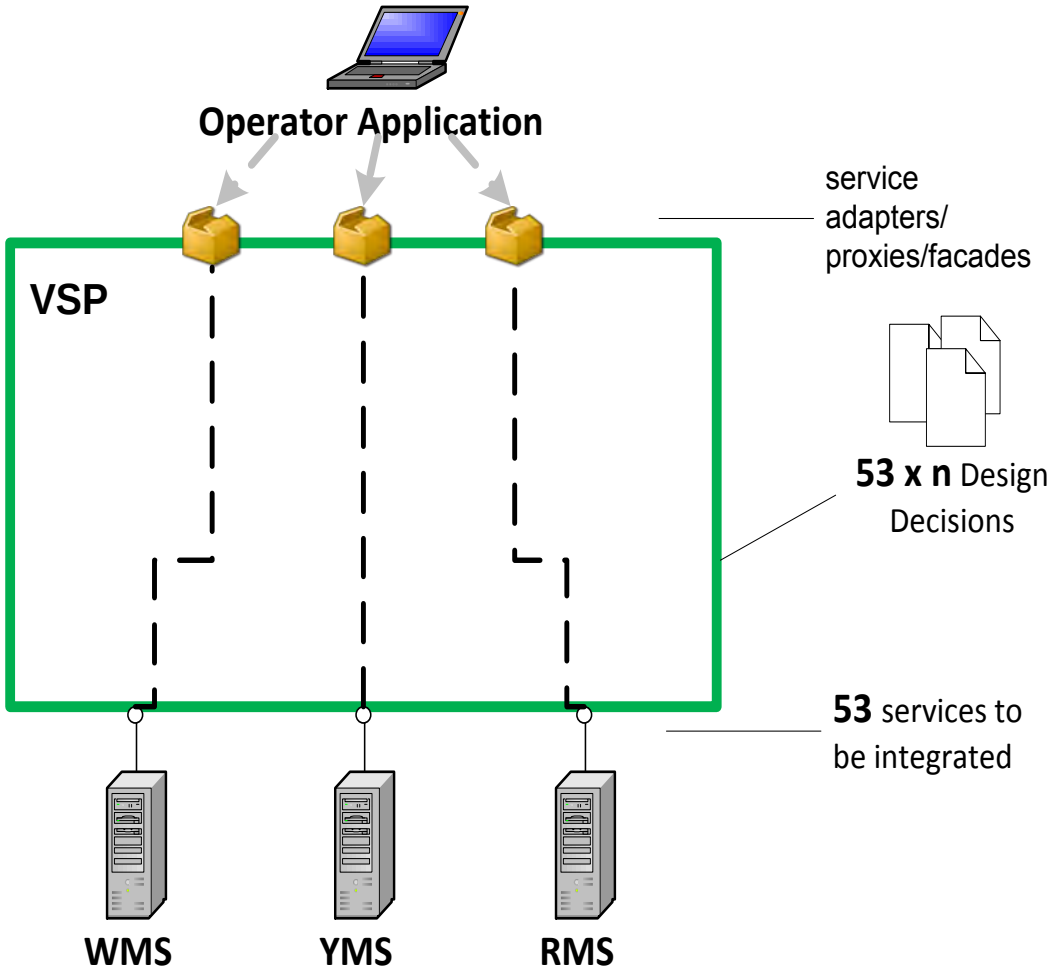
Software Platform: a collection of software sub-systems (e.g., communication middleware, databases) and interfaces which form an infrastructure for developing a set of related software applications

Research Questions

- *What are the recurring architectural design decisions on service-based platform integration documented by existing software patterns and pattern collections?*

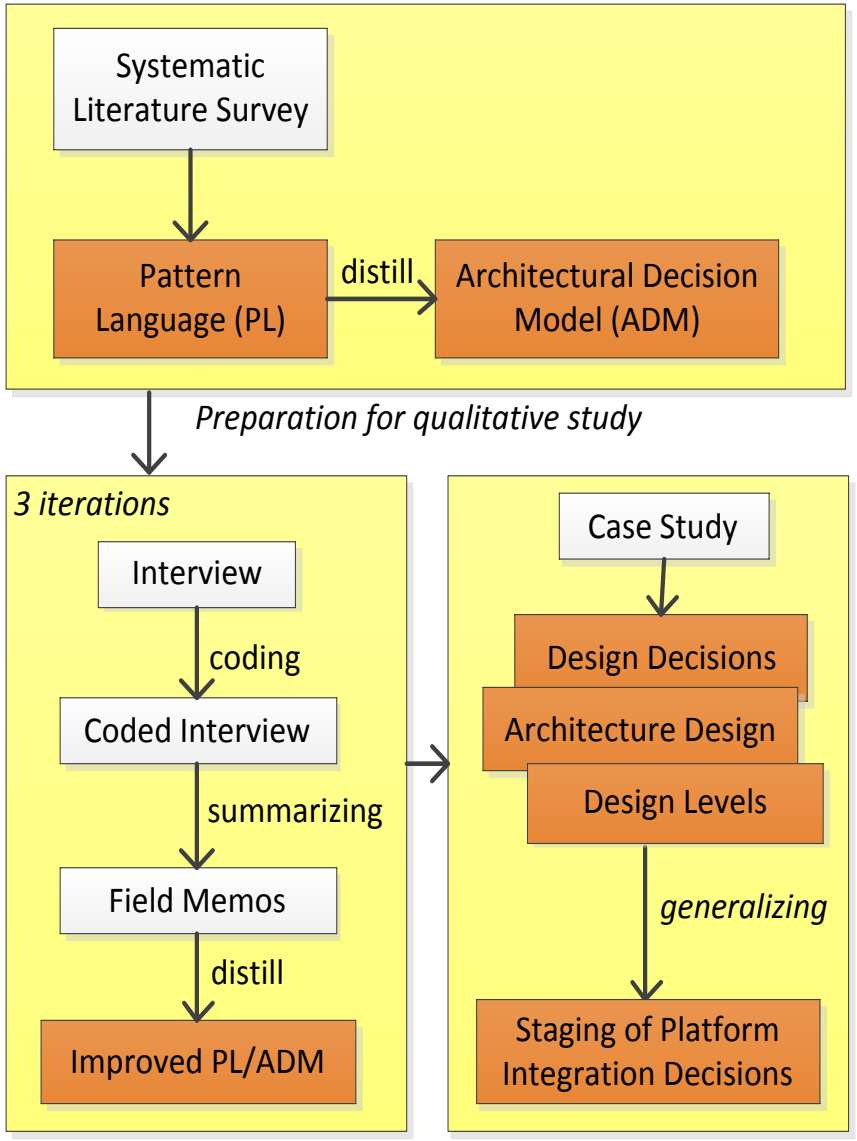
- *What are the levels of decision making when designing an architecture for service-based platform integration?*

Case Study



Virtual **S**ervice **P**latform: handles various integration aspects like interface adaptation between platforms, integration of service-based and non-service-based solutions, routing, enriching, aggregation, splitting of messages and events

Research Design



Systematic Literature Survey

- Inclusion and Exclusion Criteria: 33 conference proceedings of PLoP and EuroPLoP, 2 issues of TPLoP, 5 pattern collection and 25 pattern books (**402** patterns)
- Quality Assessment: reviewed patterns, referenced in SOA technical domain
- Pattern Extraction and Synthesis: **29** patterns selected + 11 patterns referenced

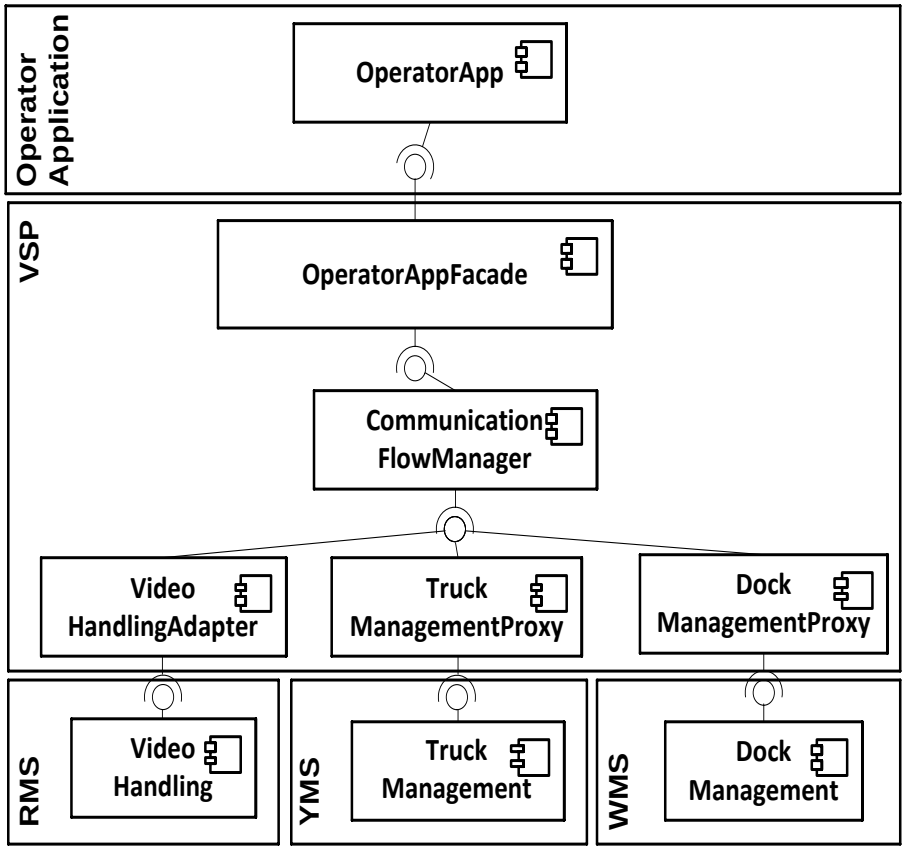
Interviews

- **3** interviews
- **9** experts
- **3** companies – **3** platforms

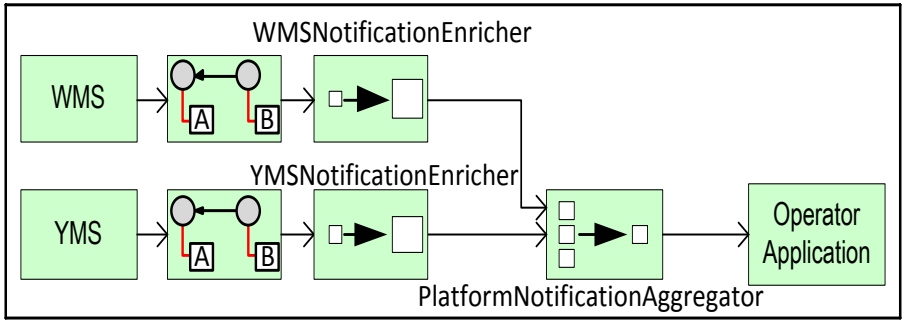
Interview Instrument (**4** categories, **29** questions)

Question	Type
Adaptation and Integration	
1.1 Can the services from the source platform be directly used in the VSP platform?	closed
Interface Design	
2.1 Have the platform services been exposed as services using standard interfaces/technologies?	closed
Communication Style	
3.1 How important is performance for the connection?	open

Case Study Design



Integration and Adaptation

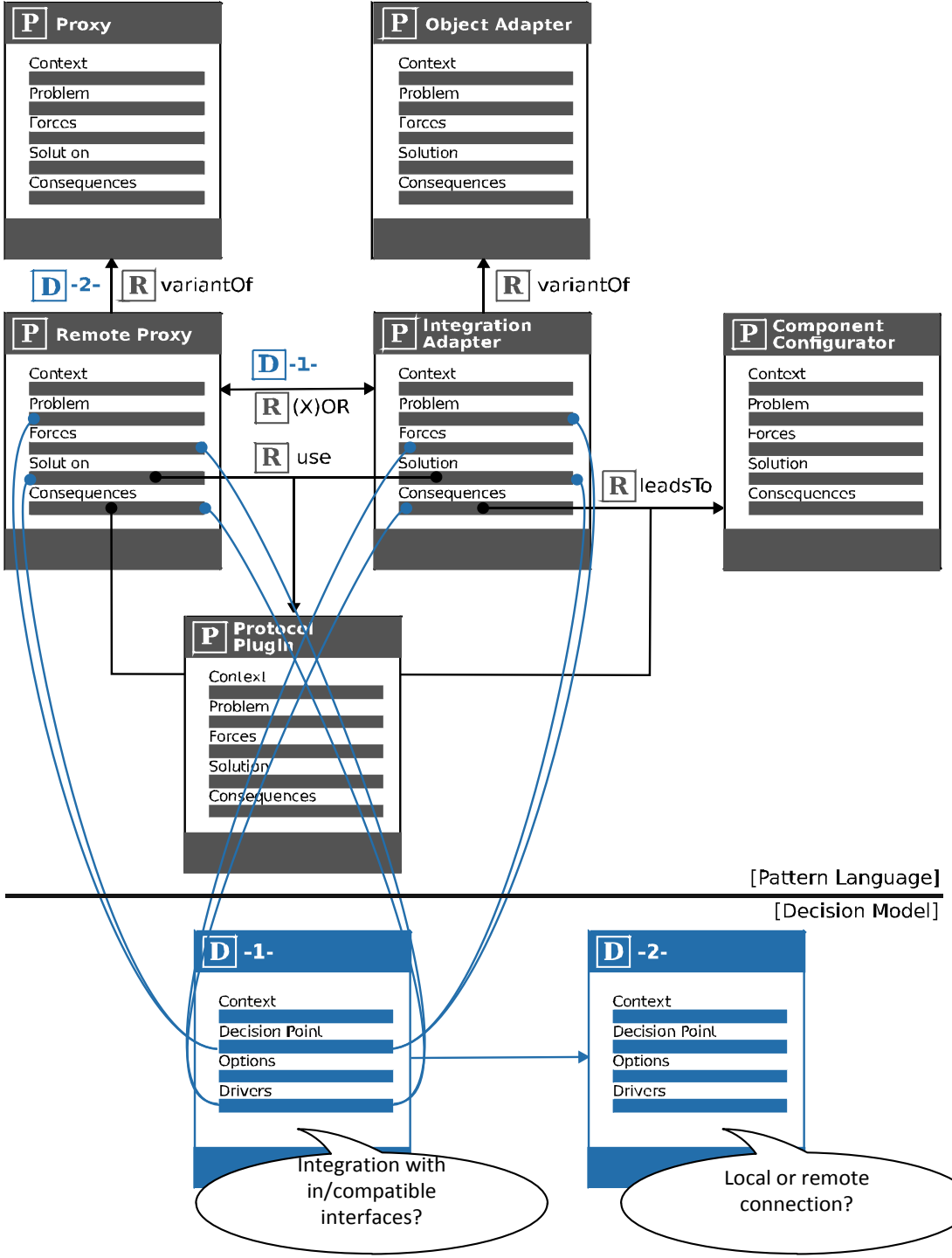


Communication Flow

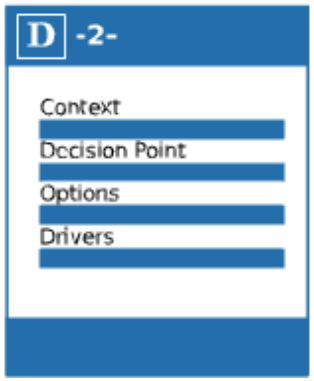
confirmatory:
to which extend our pattern language and our decision model correspond to the platform integration domain?

exploratory:
how is the decision making in platform integration being performed?

Relations between ADDs and Patterns



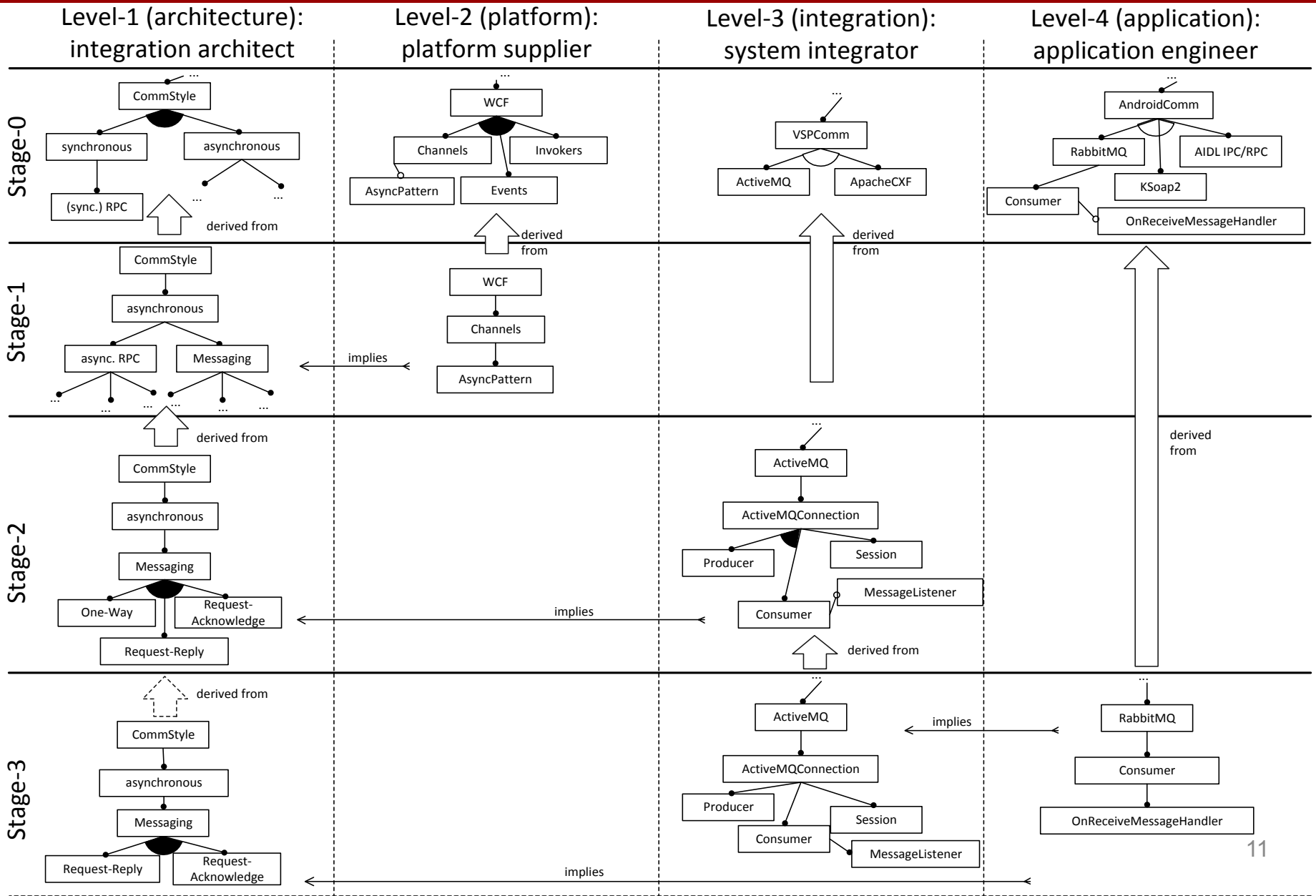
ADD Model



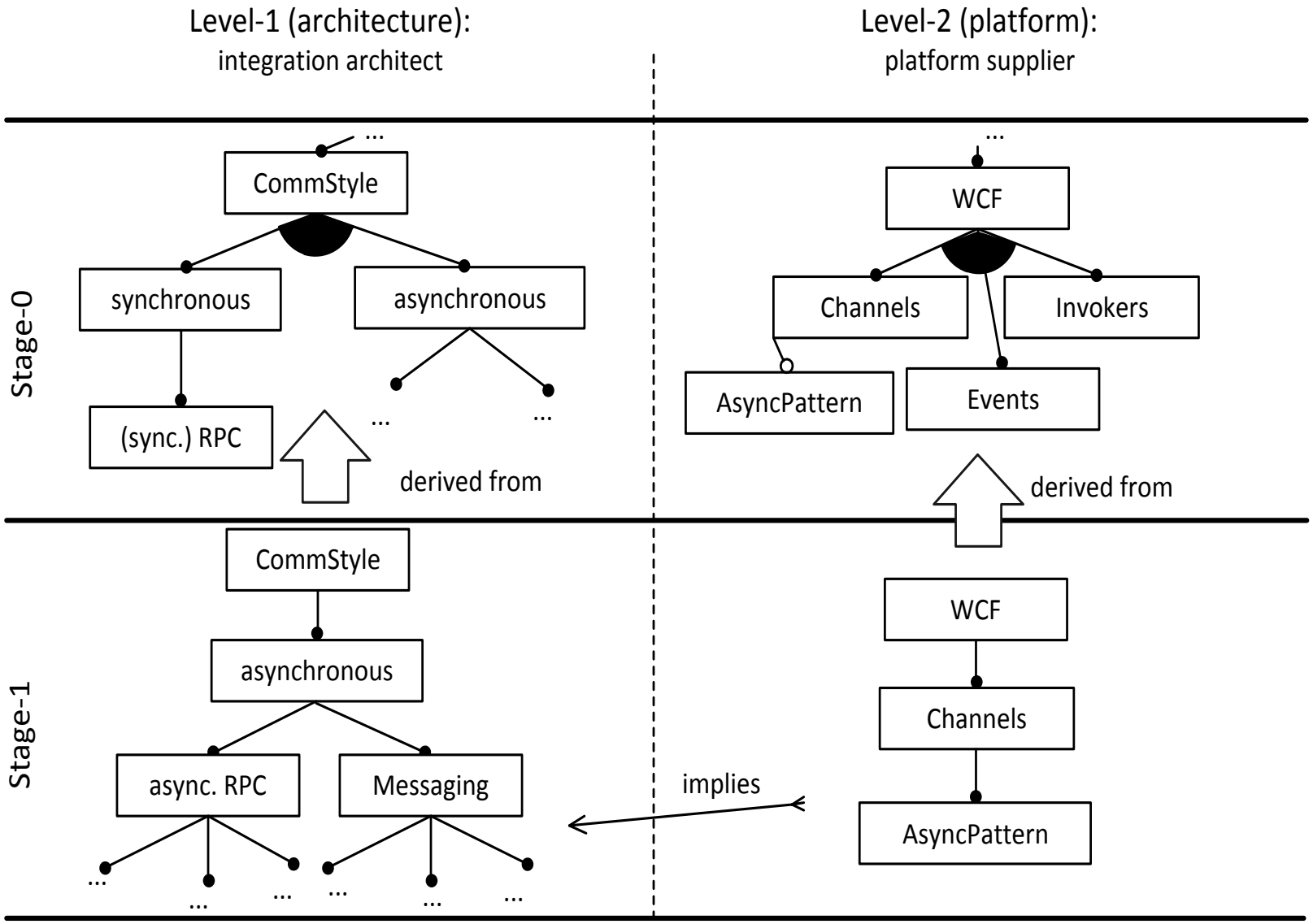
- Adaptation and Integration Patterns (6)
- Interface Design (6)
- Communication Style (8)
- Communication Flow (9)

Decision Point	Options and Patterns Dependencies
$D1$ – Which kind of component will be used for integrating the platform service into the service-based integration platform?	• None (direct calls from application to platform) • Integration component with same interface (select pattern PROXY or a PROXY variant) • Integration component with a different interface (select pattern ADAPTER or an ADAPTER variant)
$D2$ – Is the connection between platform and service-based integration platform a local or a remote connection?	• Local (Select local variant of PROXY or ADAPTER, as selected in other decisions) • Remote (Select remote variant of PROXY or ADAPTER, as selected in other decisions)

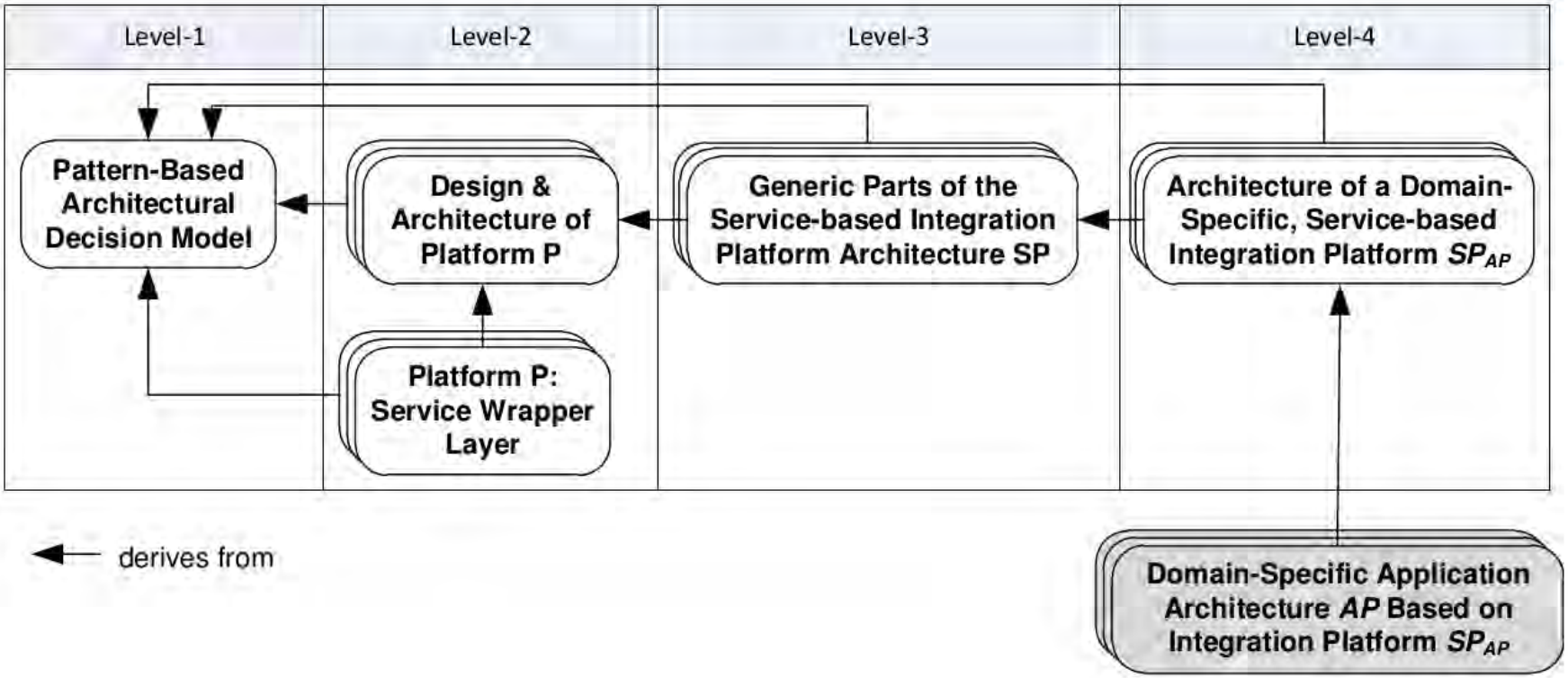
Exemplary Levels and Stages of Decision Making



Example for 2 Levels and 2 Stages



Artifacts in Levels of Decision Making



Limitations and Threats

- **Systematic Literature Review**: completeness, authors' bias
- **Interviews**: external and internal validity
- **Generalizability**: small sample but broad domain

Lessons Learned

1. Using software patterns facilitates iterative decision making.
2. Patterns are an important communication vehicle between interviewers and interviewees with different backgrounds.
3. Our research design should not impose design decisions onto the subjects.
4. The architecting process should be observed in the context of a real development project.

Conclusions

- **Architectural Decision Model**
  - Refine with further qualitative studies
  - Asses its cost-benefit balance
- **Decision stages and levels**
  - Tool support
  - How do they apply to other platform-like software development approaches?

Thank you for your attention!