Karlsruhe Institute of Technology

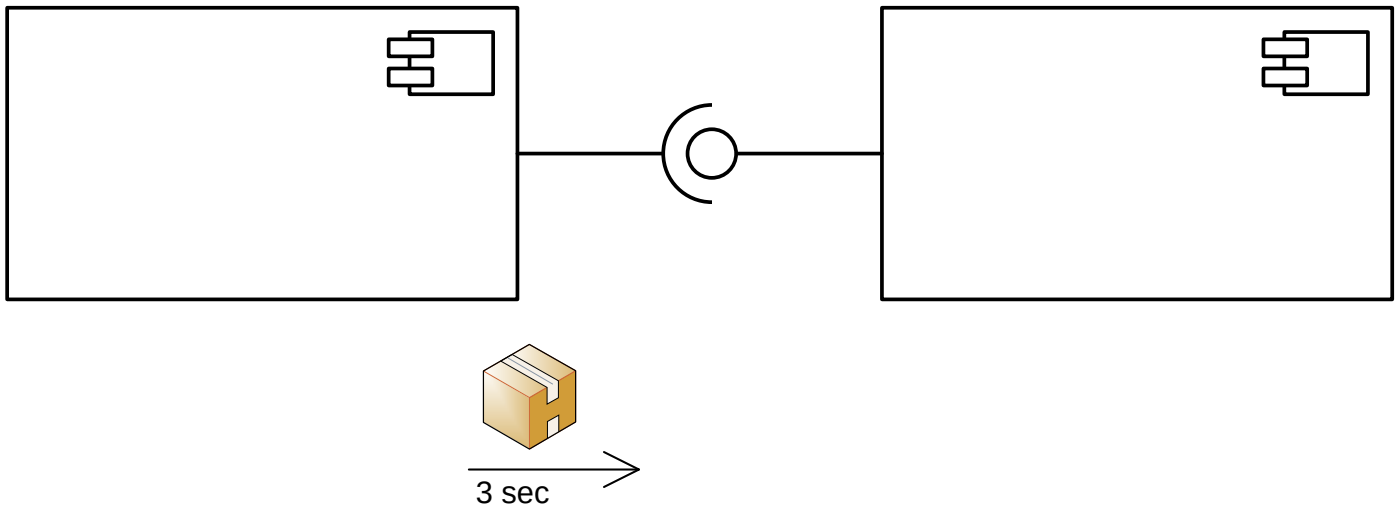
Performance Abstractions of Communication Patterns for Connectors

Study Thesis Final Presentation

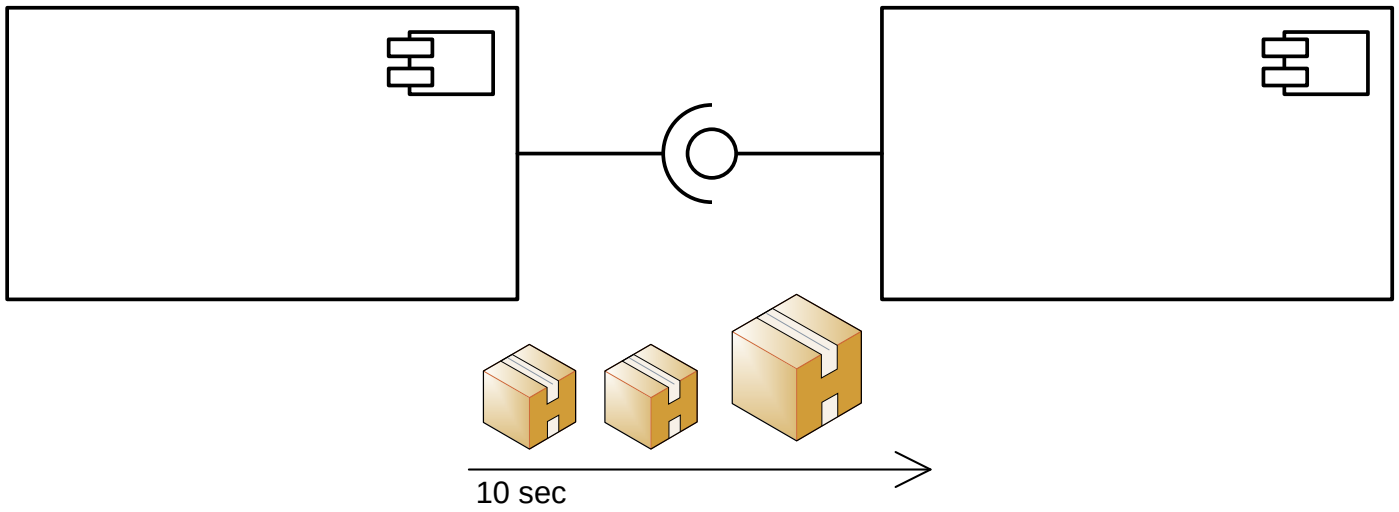
Misha Strittmatter | December 8, 2014

Chair for Software Design and Quality

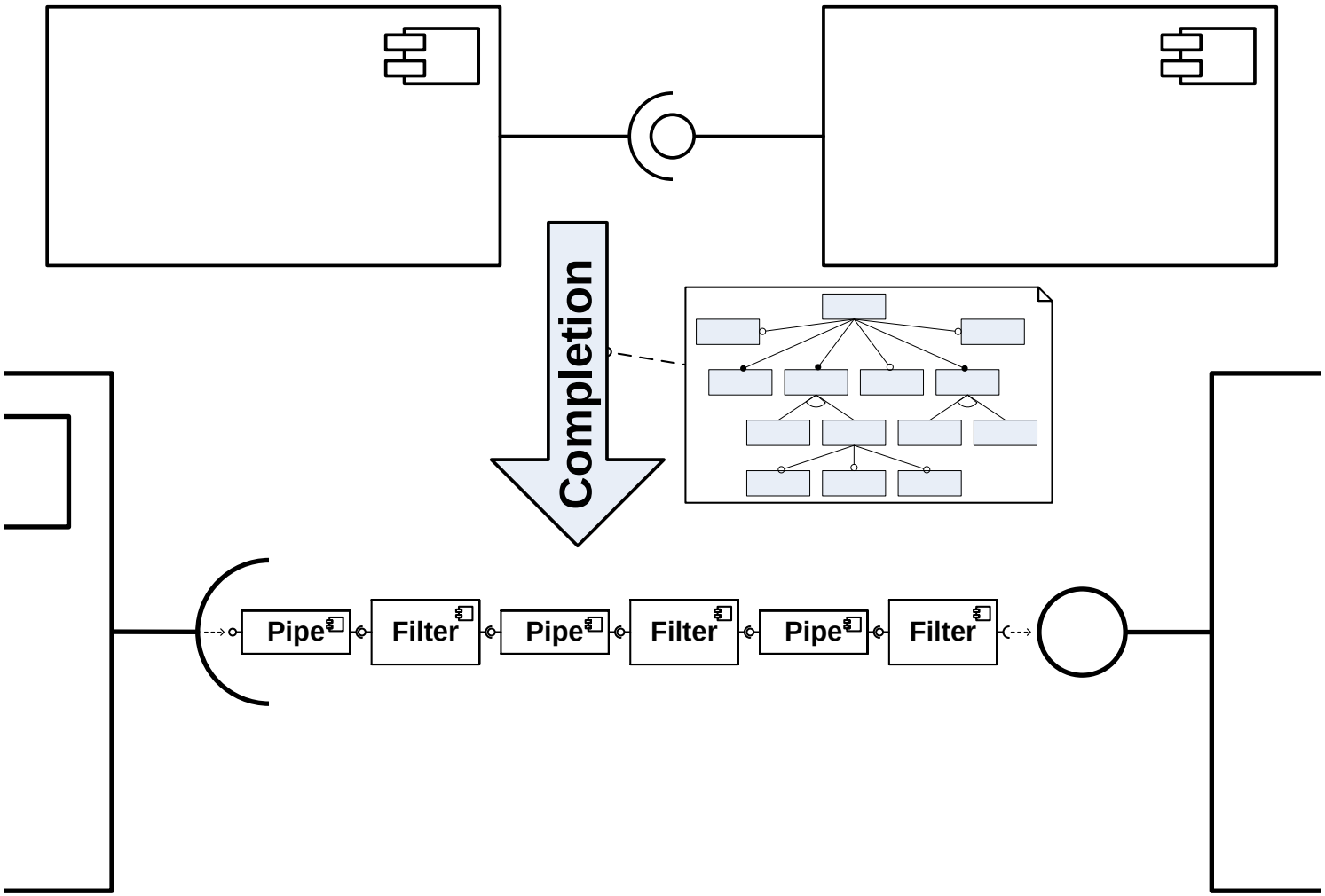
Motivation: Illustration



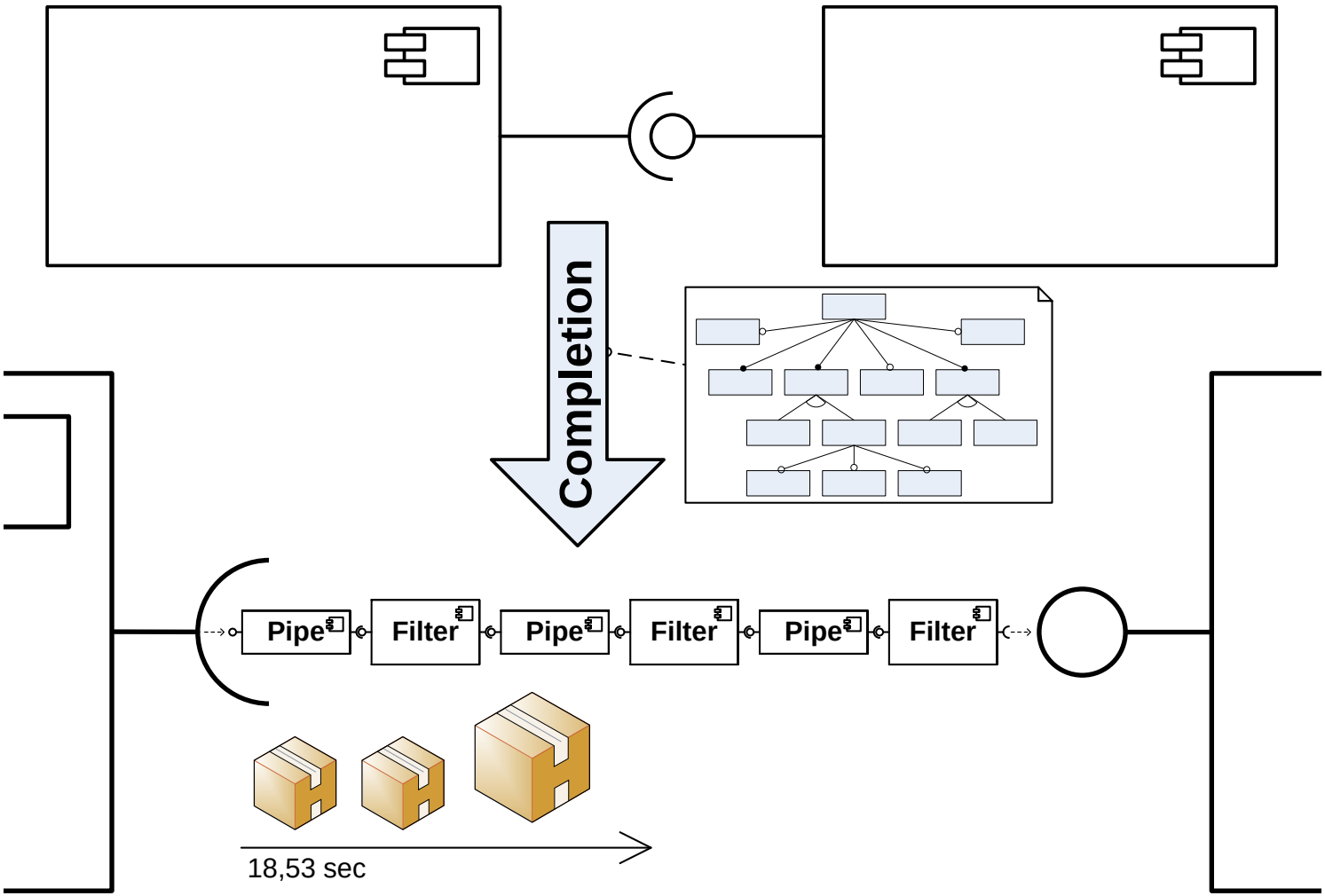
Motivation: Illustration



Motivation: Illustration

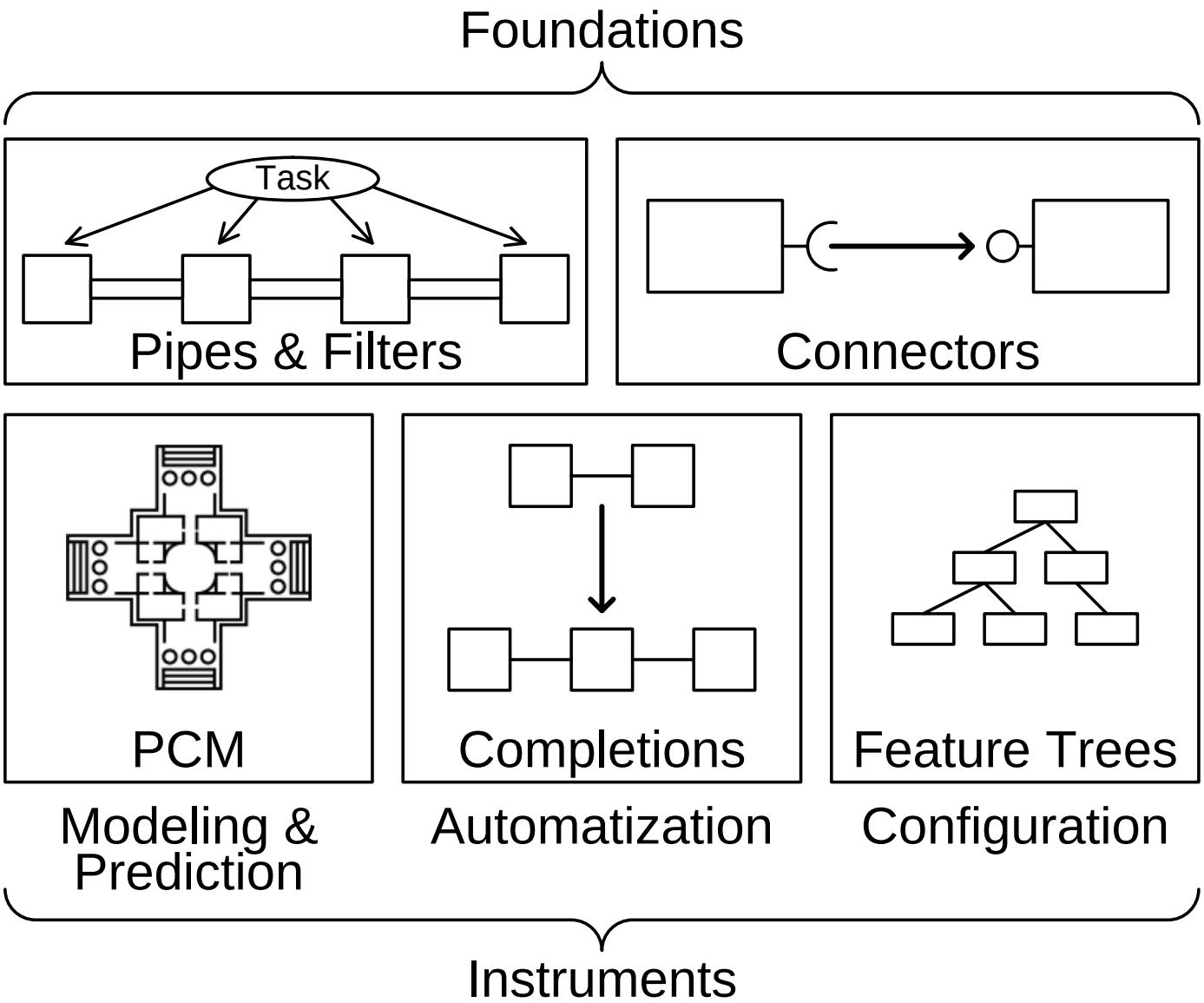


Motivation: Illustration



Motivation: PIBA

- Problem: Prediction and analysis considering non-functional properties of connectors
- Idea: Connector abstraction can be inserted by transformation
- Benefit:
  - Original model kept abstract
  - Simulations on intermediate model enriched with implementation details
  - (Coordination of other communication oriented completions)
- Action:
  - Research
  - Assert feasibility
  - Design basic constructs
  - Model arch and FMs
  - Map features to arch
  - Evaluate



[Bus07, BKR07, CE00]

Research: Related Work

Connectors

- Connector taxonomy [MMP00]
- **Composing connectors of elements [BP03]**

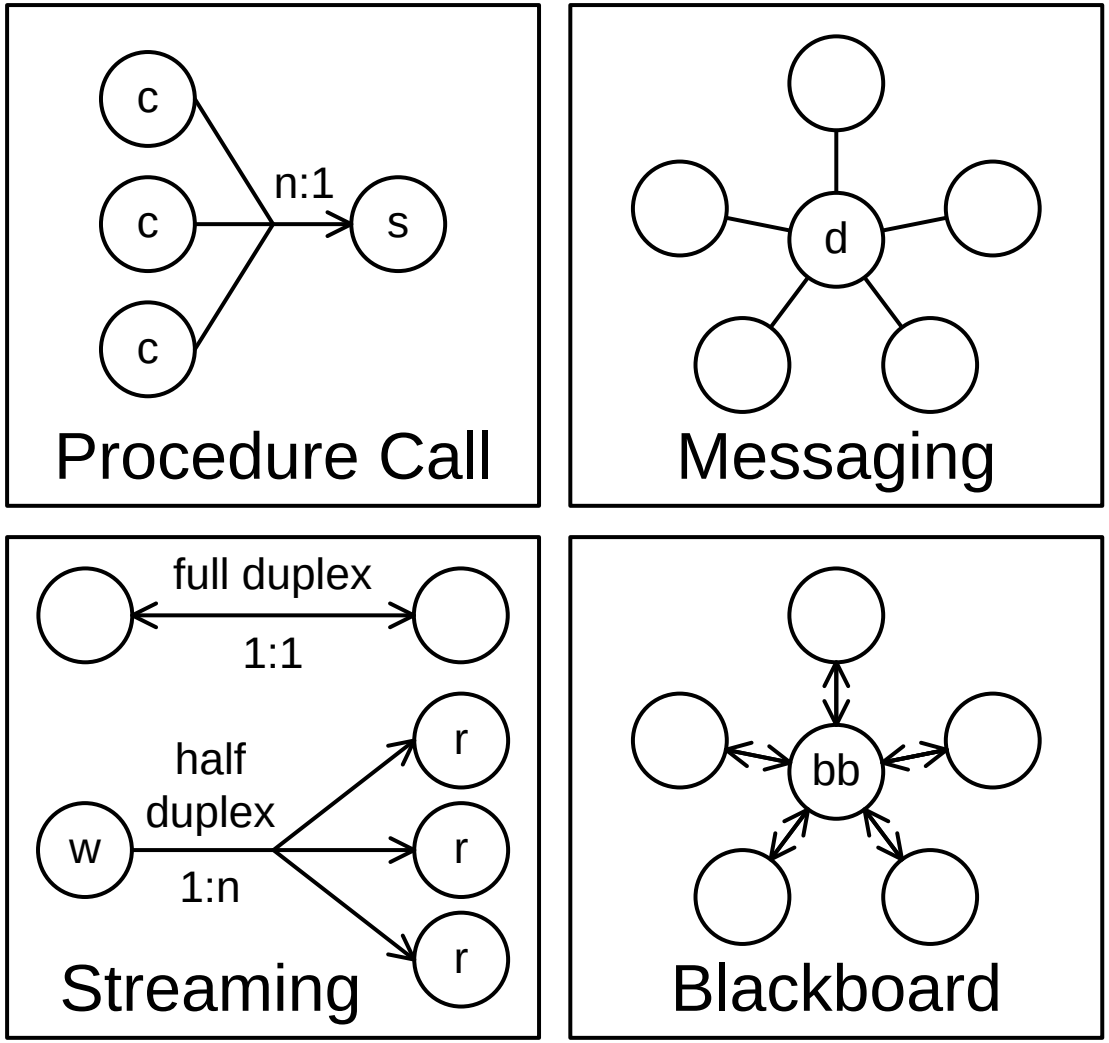
Completions

- Completion generation [KHB10], conflict resolution [KG09]
- Replication [Har10]
- Middleware completion [HFBR08]
- Encryption, Compression, Marshalling [Bec08]

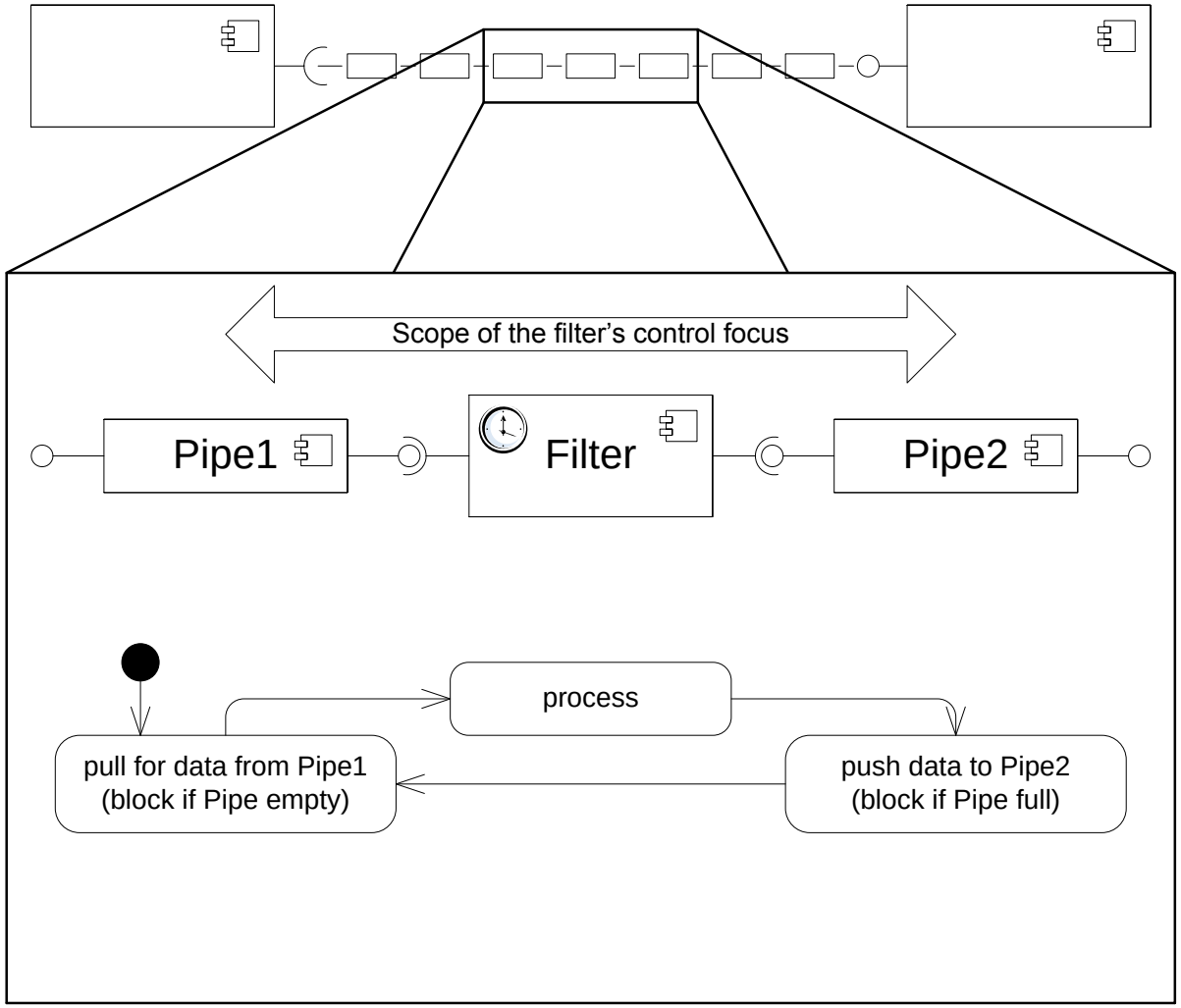
Component Systems

- Surveys [Koz10, LW05, KZM+10, Bur06]
- ROBOCOP [BMdW+04, Rob], OLAN [BBB+98], SOFA [PBJ98]

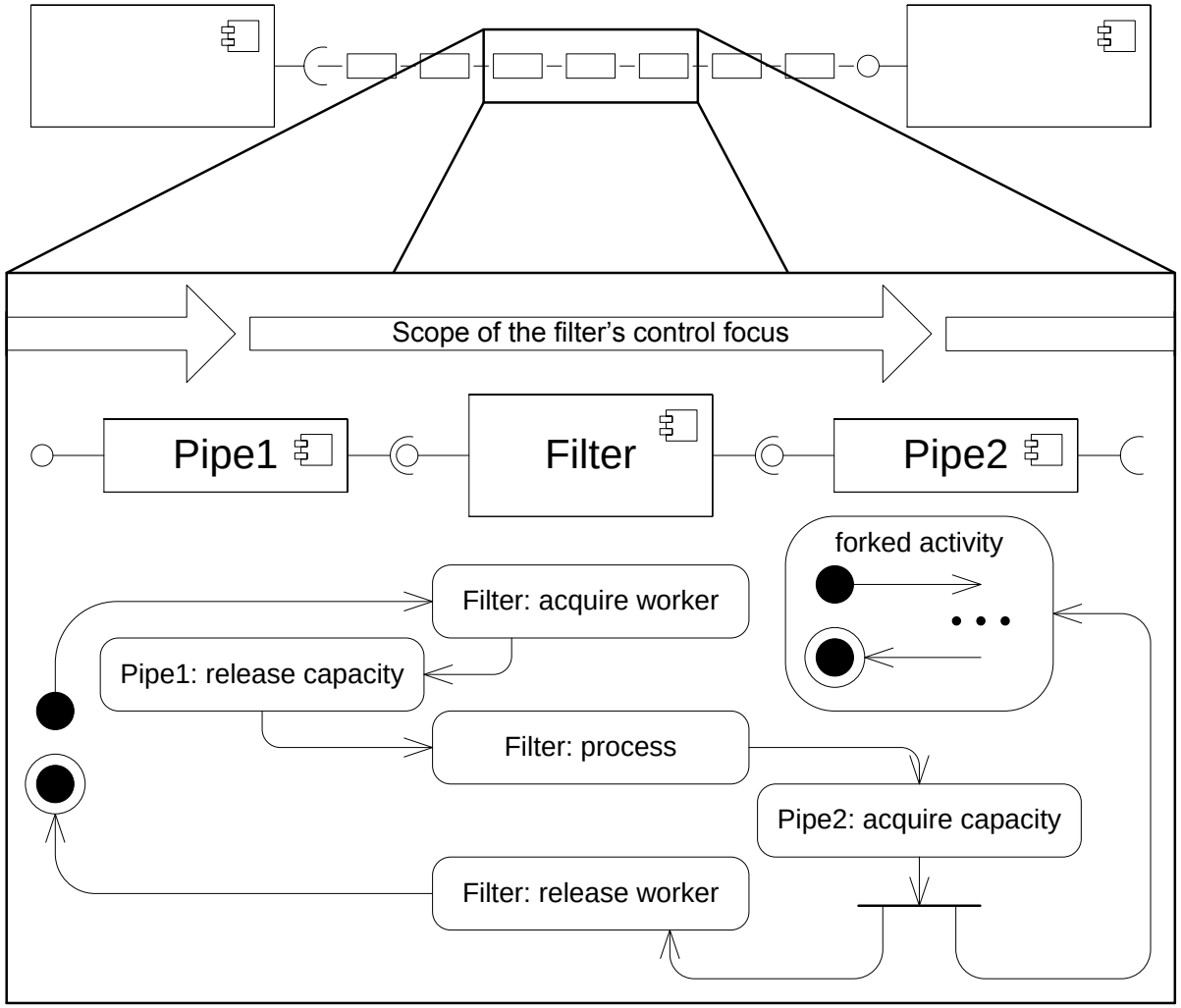
Research: Connector Topology



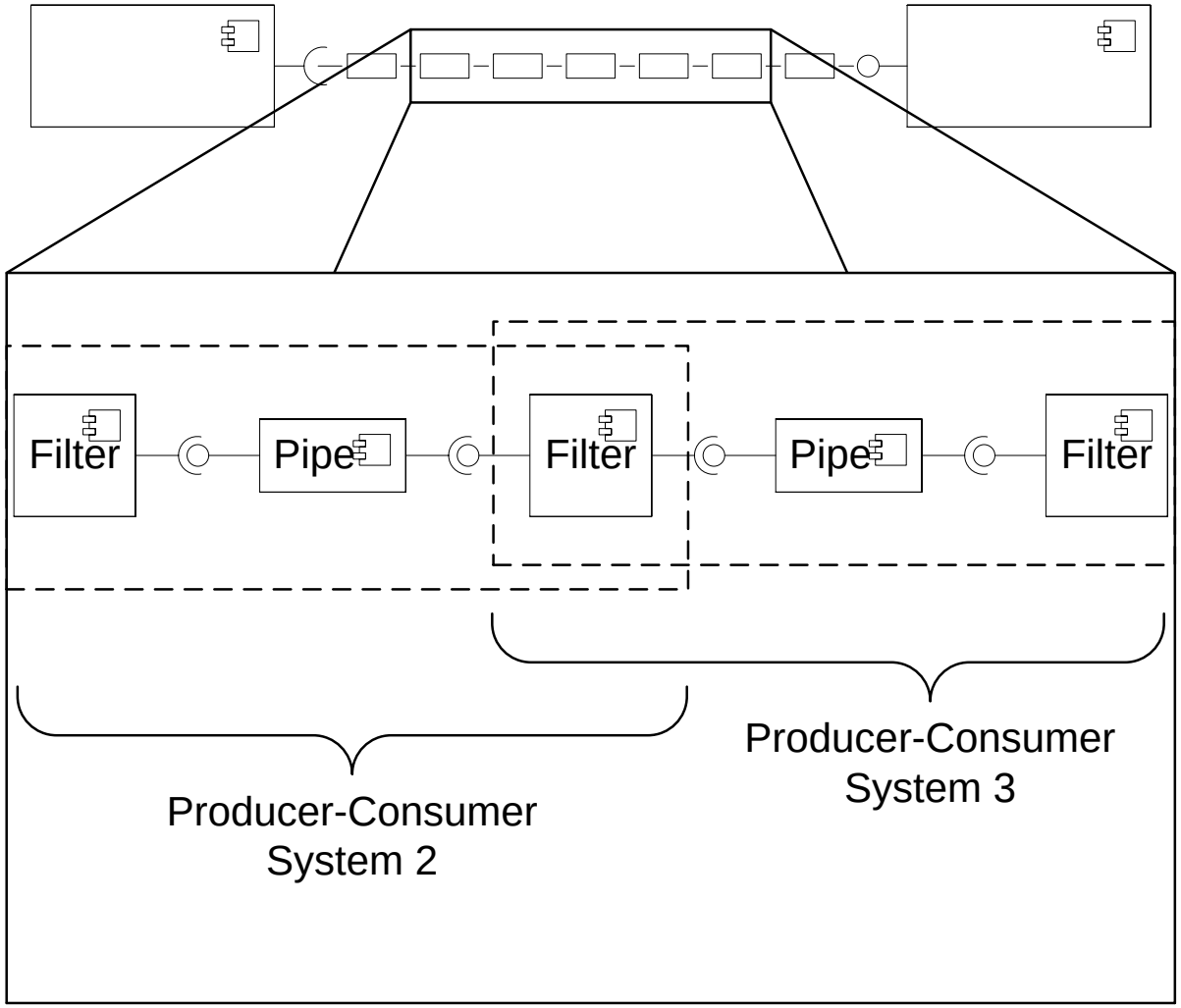
Contribution: Basic Design



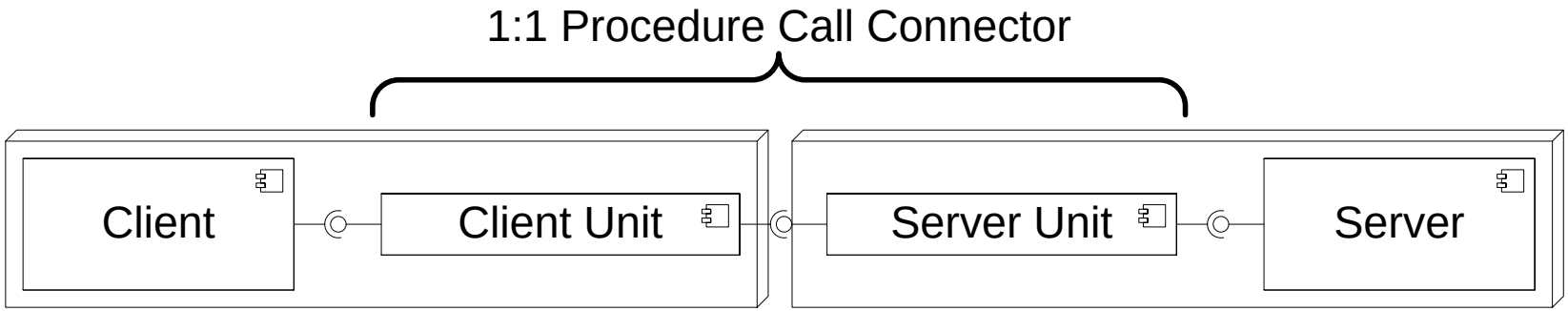
Contribution: Basic Design



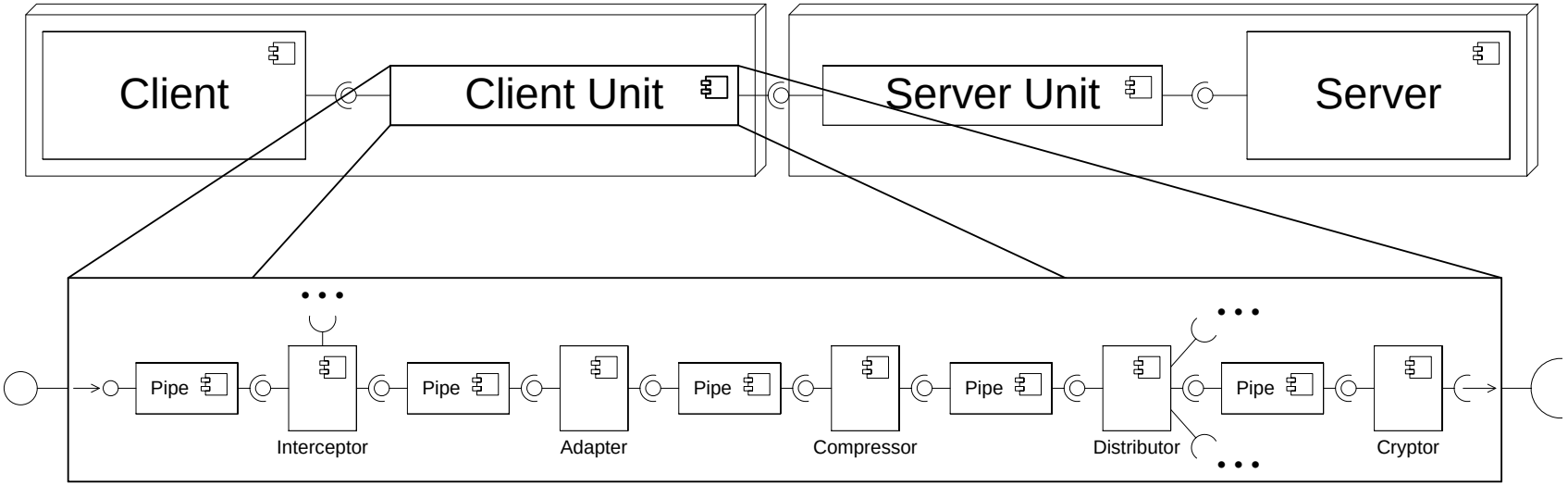
Contribution: Basic Design



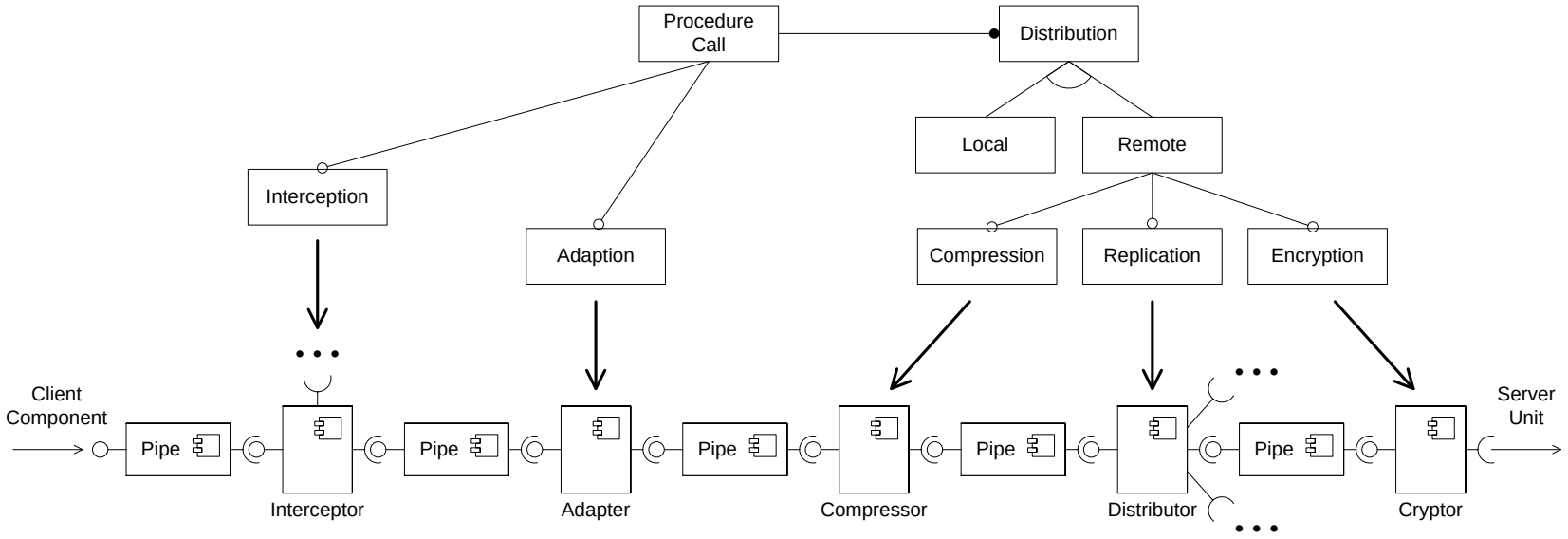
Contribution: Architecture Excerpt



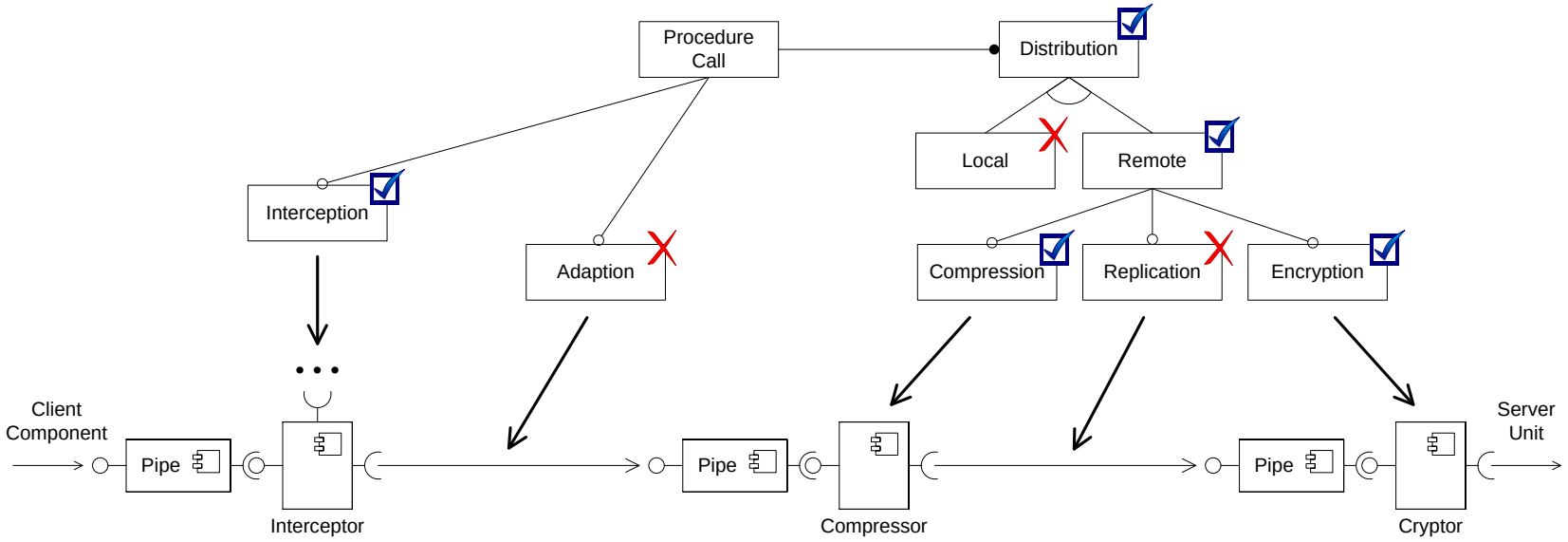
Contribution: Architecture Excerpt



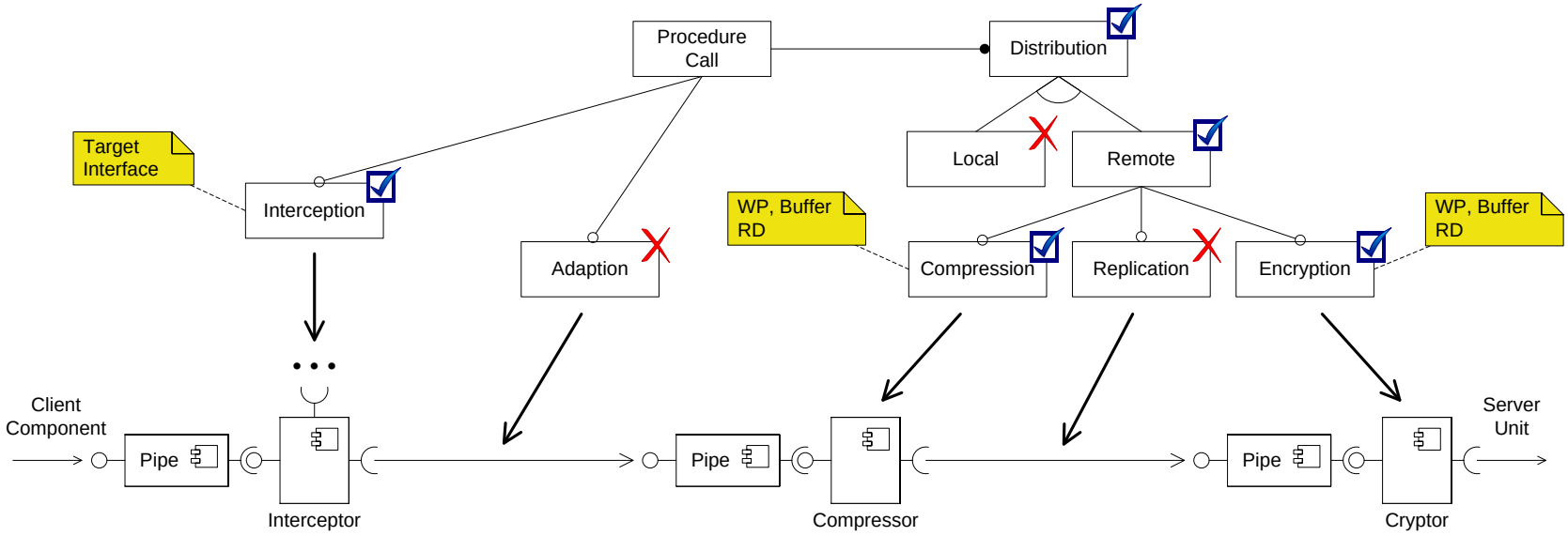
Contribution: Feature Arch Mapping



Contribution: Feature Selection



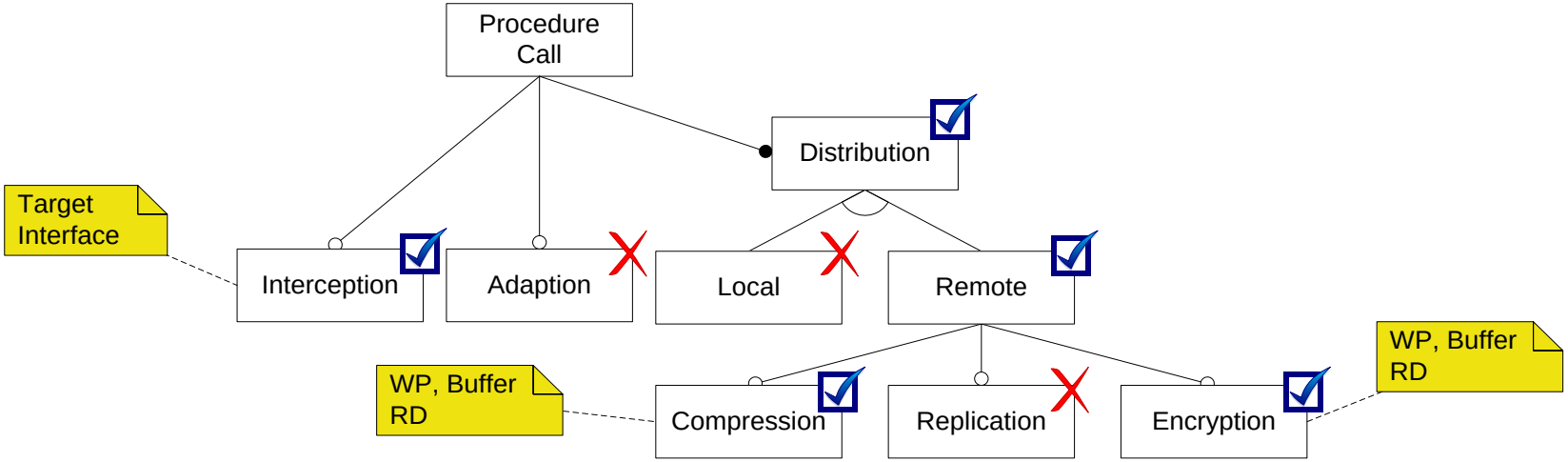
Contribution: Feature Annotation



Evaluation

We identified following effects:

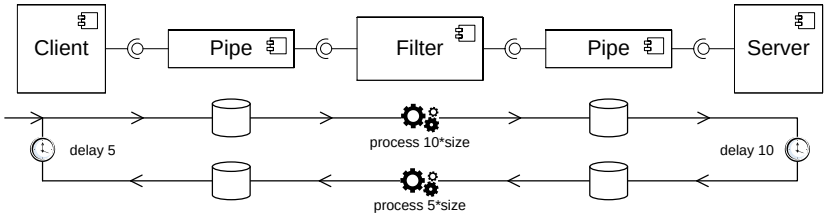
- **Feature configuration**
- Limitation of concurrency
- Piling chain effect
- Use of asynchronous calls

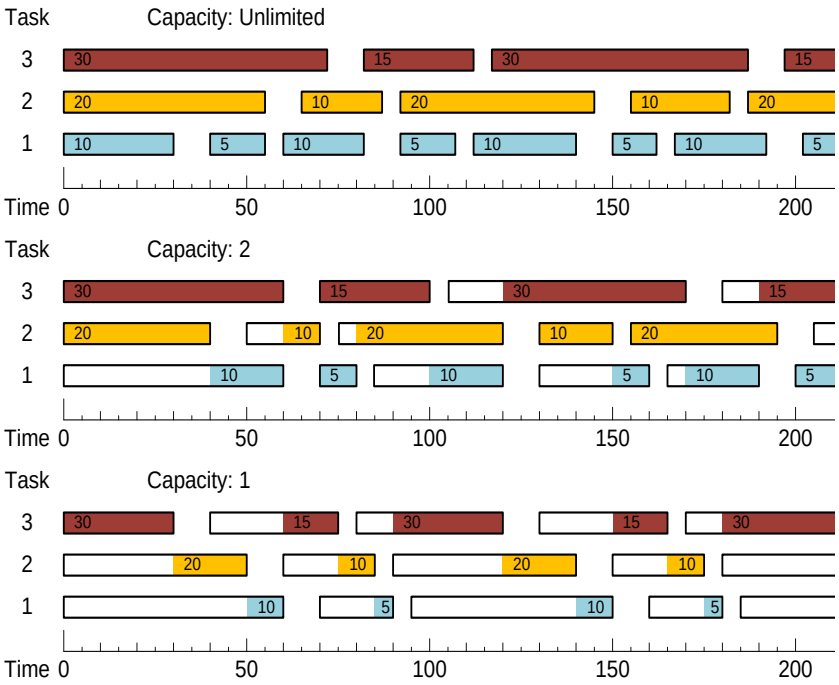


Evaluation

We identified following effects:

- Feature configuration
- **Limitation of concurrency**
- Piling chain effect
- Use of asynchronous calls

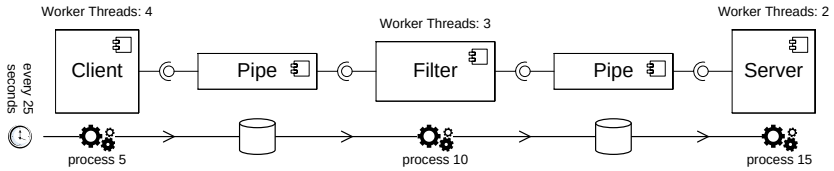


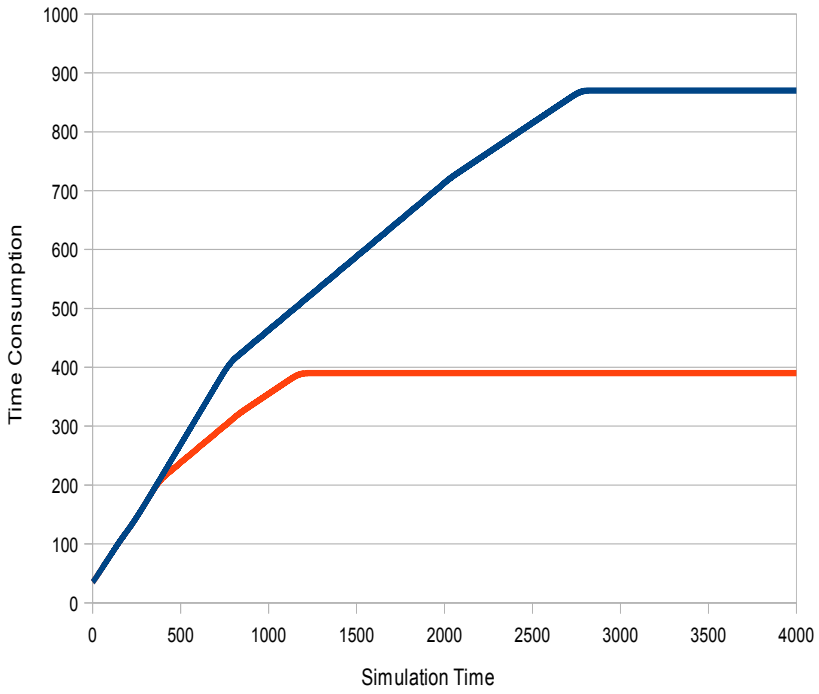


Evaluation

We identified following effects:

- Feature configuration
- Limitation of concurrency
- **Piling chain effect**
- Use of asynchronous calls

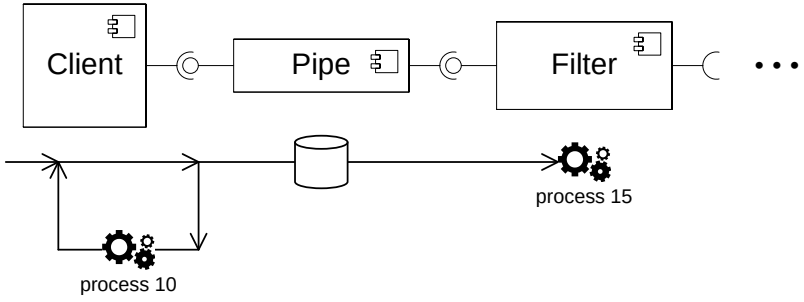


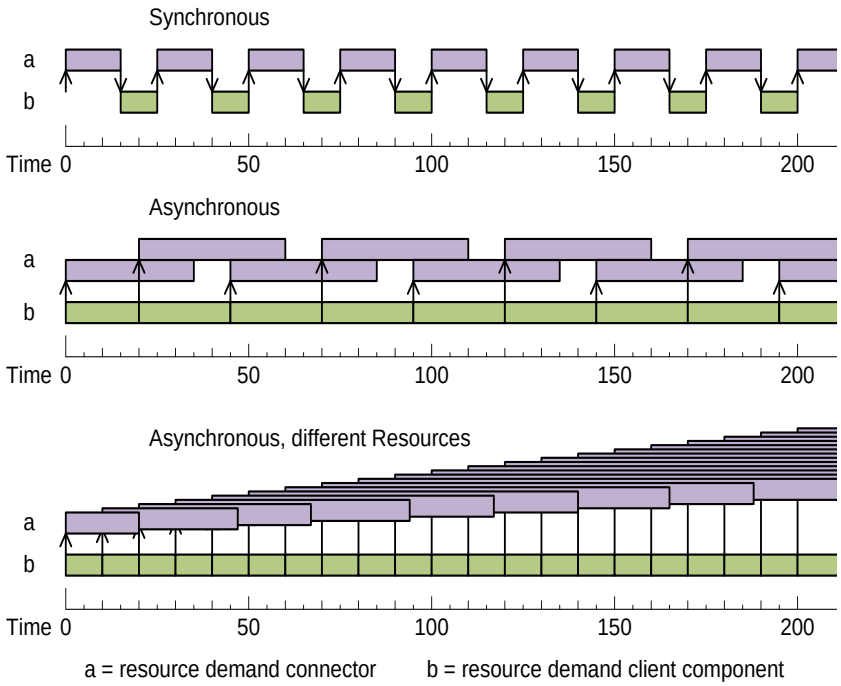


Evaluation

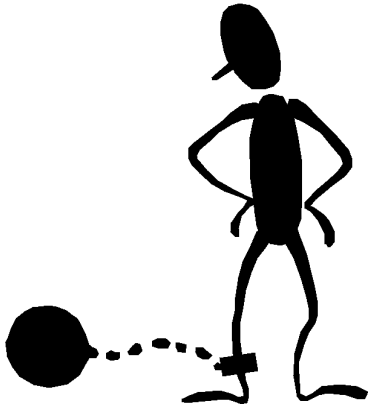
We identified following effects:

- Feature configuration
- Limitation of concurrency
- Piling chain effect
- **Use of asynchronous calls**





Restrictions



- Connector-use restricted to its communication method. Cannot be deployed in arbitrary context.
- Restricted to PCM $\rightarrow$ obvious and intended
- Remaining restrictions $\rightarrow$ future work (see next slide)

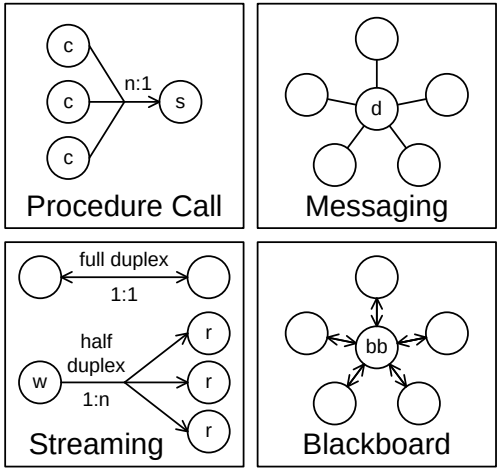
Conclusion

Feasibility assured:
active pulling, buffering, asynch, connectors

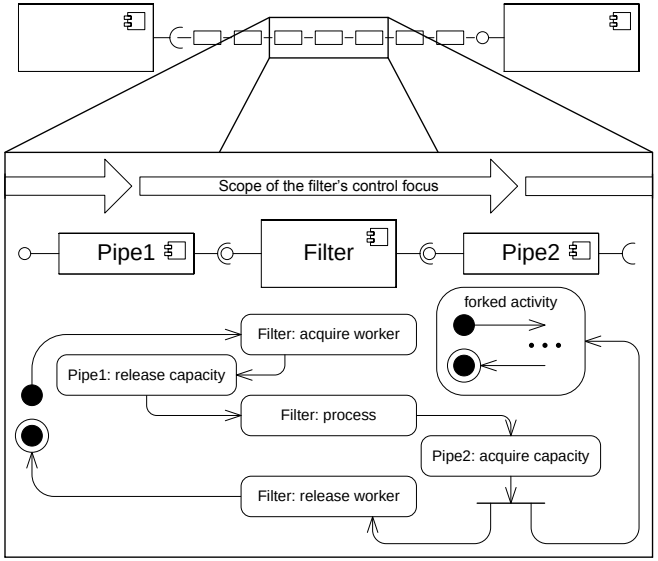
Future Work

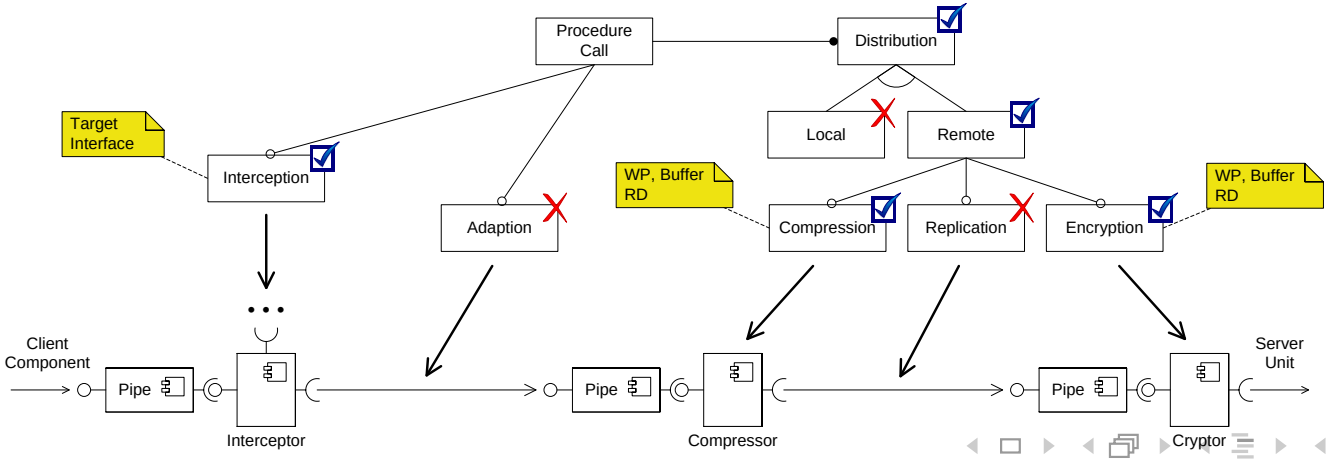
- Implementation (done: static prototype [Vog10], open: configurable completion)
- Needs data and incorporation of other completions
- Case study (evaluate concept)
- Find and validate connector configurations
- Refinement and expansion (architecture and feature model)
- Identify and implement new connectors
- Integration into a completion library or the PCM IDE

The End



Thanks for your attention!
Any questions?





References I

[BBB+98] R. Balter, L. Bellissard, F. Boyer, M. Riveill, and J.-Y. Vion-Dury.
Architecturing and configuring distributed application with olan.
In *Middleware '98: Proceedings of the IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 241–256, London, UK, 1998. Springer-Verlag.

[Bec08] Steffen Becker.
Coupled Model Transformations for QoS Enabled Component-Based Software Design.
PhD thesis, University of Oldenburg, Germany, March 2008.

[BKR07] Steffen Becker, Heiko Koziolek, and Ralf Reussner.
Model-based performance prediction with the palladio component model.
In *WOSP '07: Proceedings of the 6th international workshop on Software and performance*, pages 54–65, New York, NY, USA, 2007. ACM.

[BMdW+04] Egor Bondarev, Johan Muskens, Peter de With, Michel Chaudron, and Johan Lukkien.
Predicting real-time properties of component assemblies: A scenario-simulation approach.
EUROMICRO Conference, 0:40–47, 2004.

[BP03] Tomas Bures and Frantisek Plasil.
Composing connectors of elements.
Technical Report 3, Dep. of SW Engineering, Charles University, Prague, May 2003.

[Bur06] Tomas Bures.
Generating Connectors for Homogeneous and Heterogeneous Deployment.
PhD thesis, Charles University, Prague, September 2006.

[Bus07] Frank Buschmann.
Pattern-oriented software architecture, volume 1: A system of patterns.
Wiley, Chichester [u.a.], repr. edition, 2007.

References II

[CE00] Krzysztof Czarnecki and Ulrich Eisenecker.
Generative programming: methods, tools, and applications.
Addison Wesley, Boston [u.a.], 1. print. edition, 2000.

[Har10] Christian Harnisch.
Modeling parallel, component-based software architectures with design patterns.
Study thesis, Karlsruhe Institute of Technology (KIT), 2010.

[HFBR08] Jens Happe, Holger Friedrich, Steffen Becker, and Ralf H. Reussner.
A pattern-based performance completion for message-oriented middleware.
In *WOSP '08: Proceedings of the 7th international workshop on Software and performance*, pages 165–176, New York, NY, USA, 2008. ACM.

[KG09] Lucia Kapova and Thomas Goldschmidt.
Automated feature model-based generation of refinement transformations.
In *In Proceedings of the 35th EUROMICRO Conference on Software Engineering and Advanced Applications*, 2009.

[KHB10] Lucia Kapova, Jens Happe, and Steffen Becker.
Systematic pattern-based design of performance models for concurrent software systems.
Will be published, 2010.

[Koz10] Heiko Koziolek.
Performance evaluation of component-based software systems: A survey.
Performance Evaluation, 67(8):634 – 658, 2010.
Special Issue on Software and Performance.

[KZM+10] Lucia Kapova, Barbora Zimmerova, Anne Martens, Jens Happe, and Ralf H. Reussner.
State dependence in performance evaluation of component-based software systems.
In *Proceedings of the 1st Joint WOSP/SIPEW International Conference on Performance Engineering (WOSP/SIPEW '10)*, New York, NY, USA, 2010. ACM.

References III

[LW05] Kung-Kiu Lau and Zheng Wang.
A taxonomy of software component models.
EUROMICRO Conference, 0:88–95, 2005.

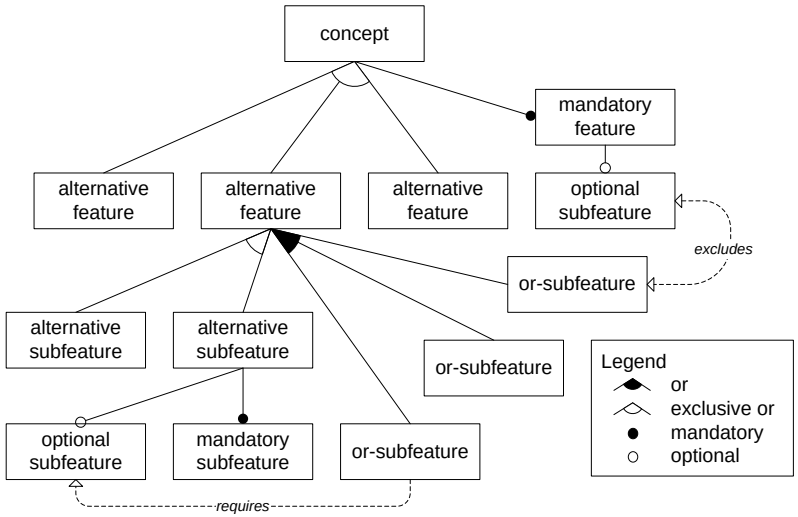
[MMP00] Nikunj R. Mehta, Nenad Medvidovic, and Sandeep Phadke.
Towards a taxonomy of software connectors.
In *ICSE '00: Proceedings of the 22nd international conference on Software engineering*, pages 178–187, New York, NY, USA, 2000. ACM.

[PBJ98] F. Plasil, D. Bálek, and R. Janecek.
SOFA/DCUP: Architecture for component trading and dynamic updating.
In *Fourth International Conference on Configurable Distributed Systems, 1998. Proceedings*, pages 43–51, 1998.

[Rob] Robocop: Robust open component based software architecture for configurable devices project.
Public deliverables.
`http://www.hitech-projects.com/euprojects/robocop/deliverables.htm`.

[Vog10] Christian Vogel.
Automated integration of pipes & filters connector configurations in PCM.
Study thesis proposal, Karlsruhe Institute of Technology (KIT), 2010.

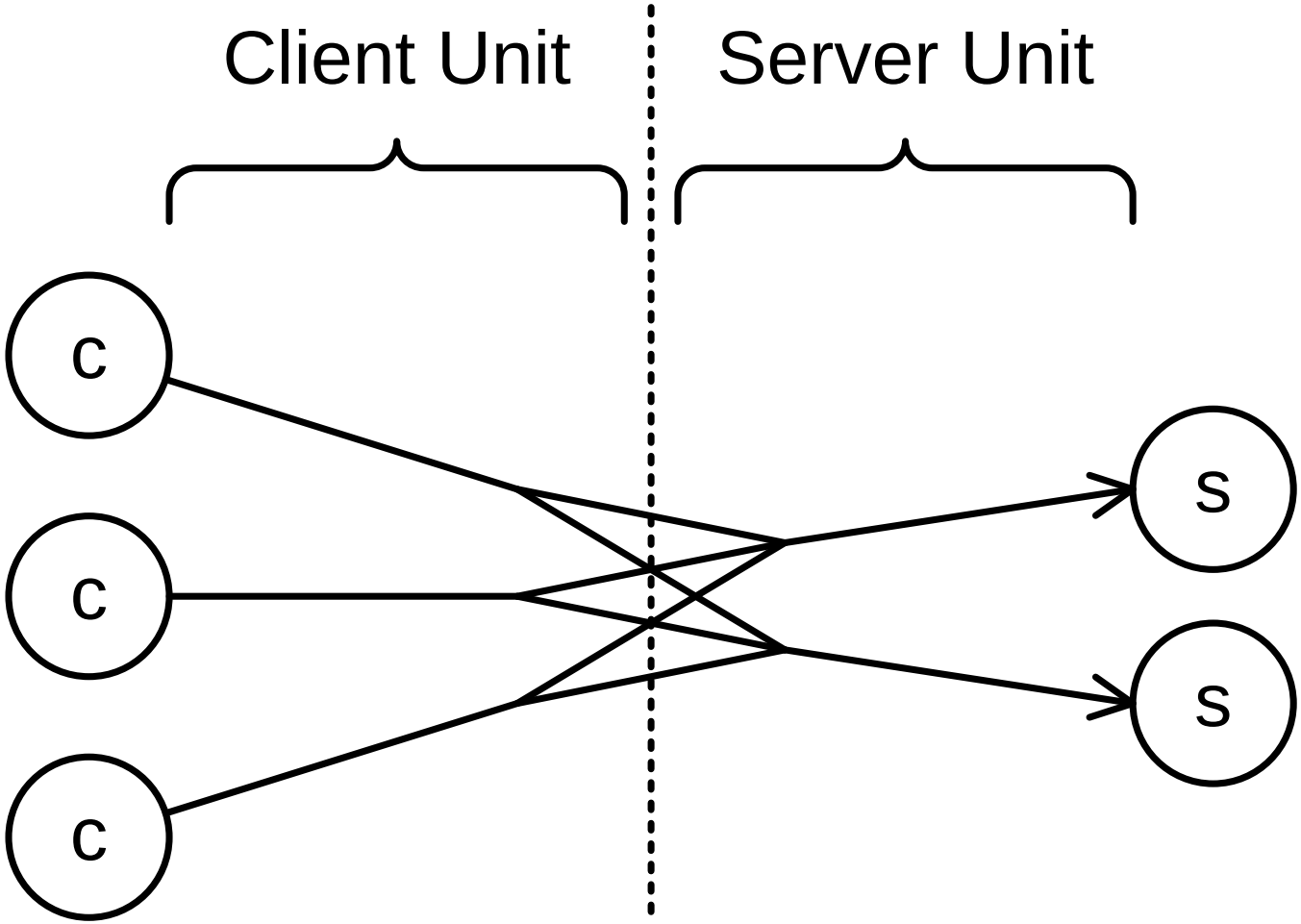
Appendix: Feature Diagram



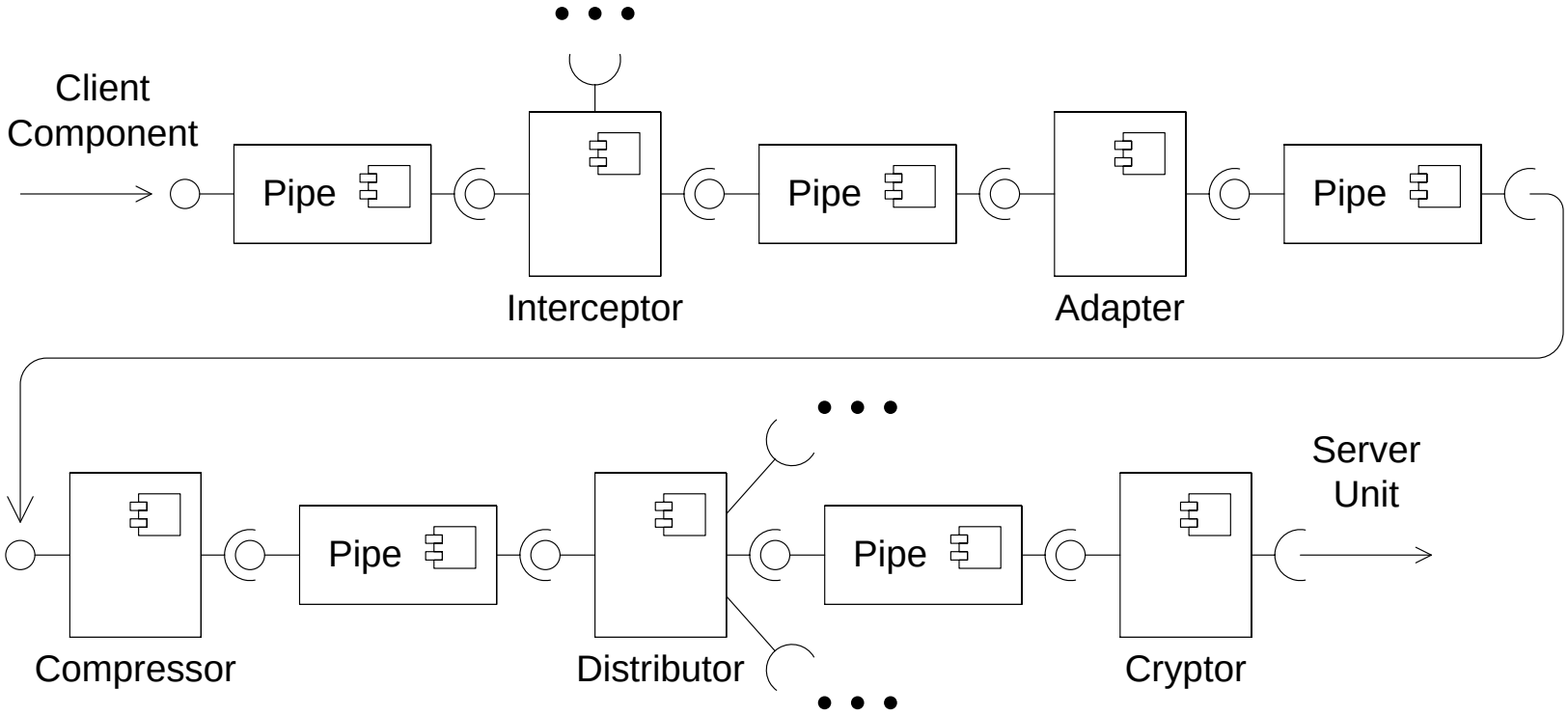
An instance of a feature diagram is a subset of nodes, selected after the following restrictions:

- The root is always included.
- Mandatory features have to be included, if their parent is included.
- Optional features can be included, if their parent is included.
- One or more features of an or-feature set have to be included, if their parent is included.
- Exactly one feature of an exclusive-or-feature set has to be included, if its parent is included.
- All requires- and excludes-constraints of a feature are satisfied.

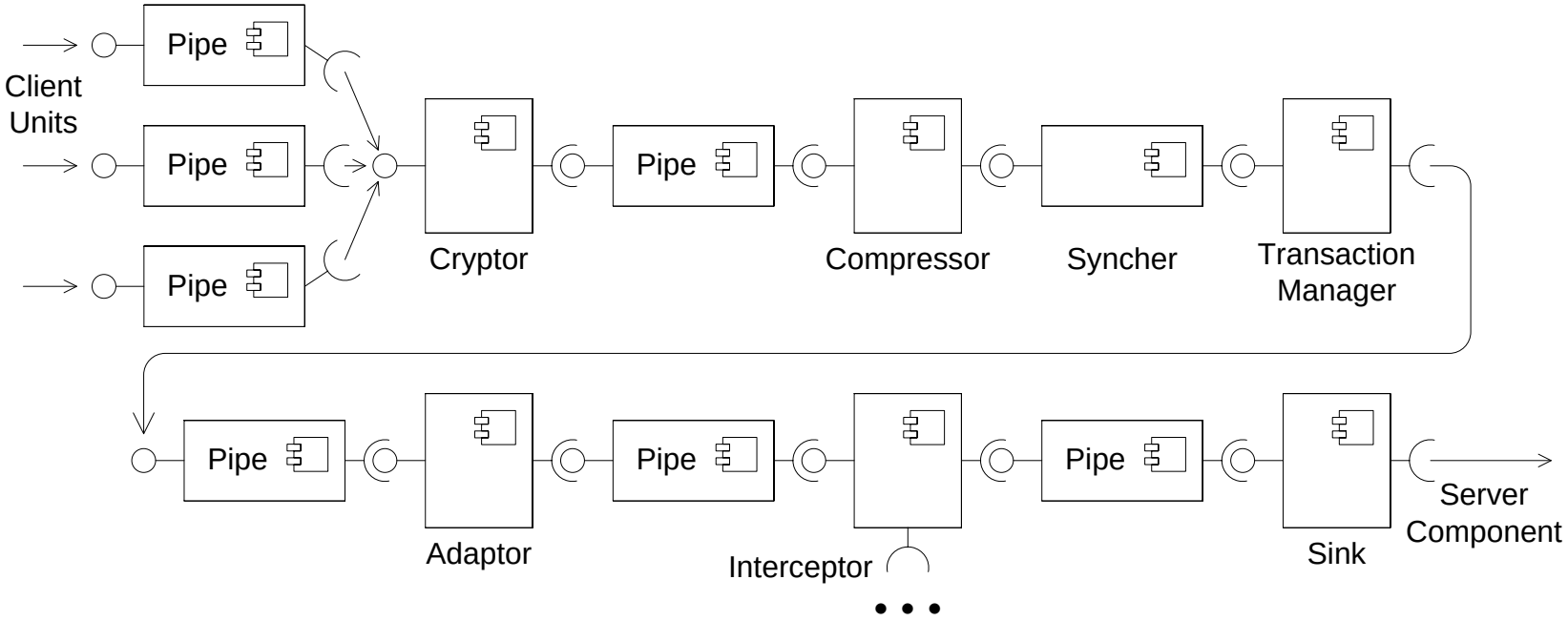
Appendix: Full Procedure Call Connector Topology



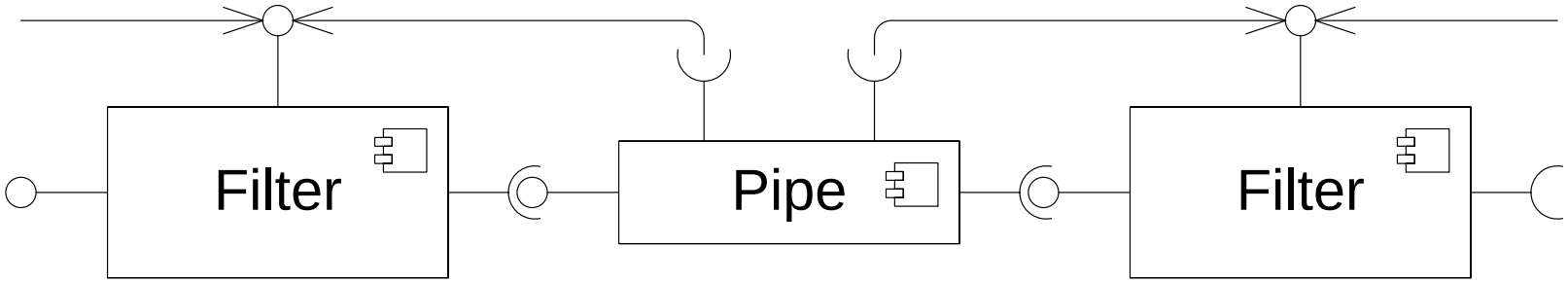
Appendix: Procedure Call Connector Client Unit Architecture



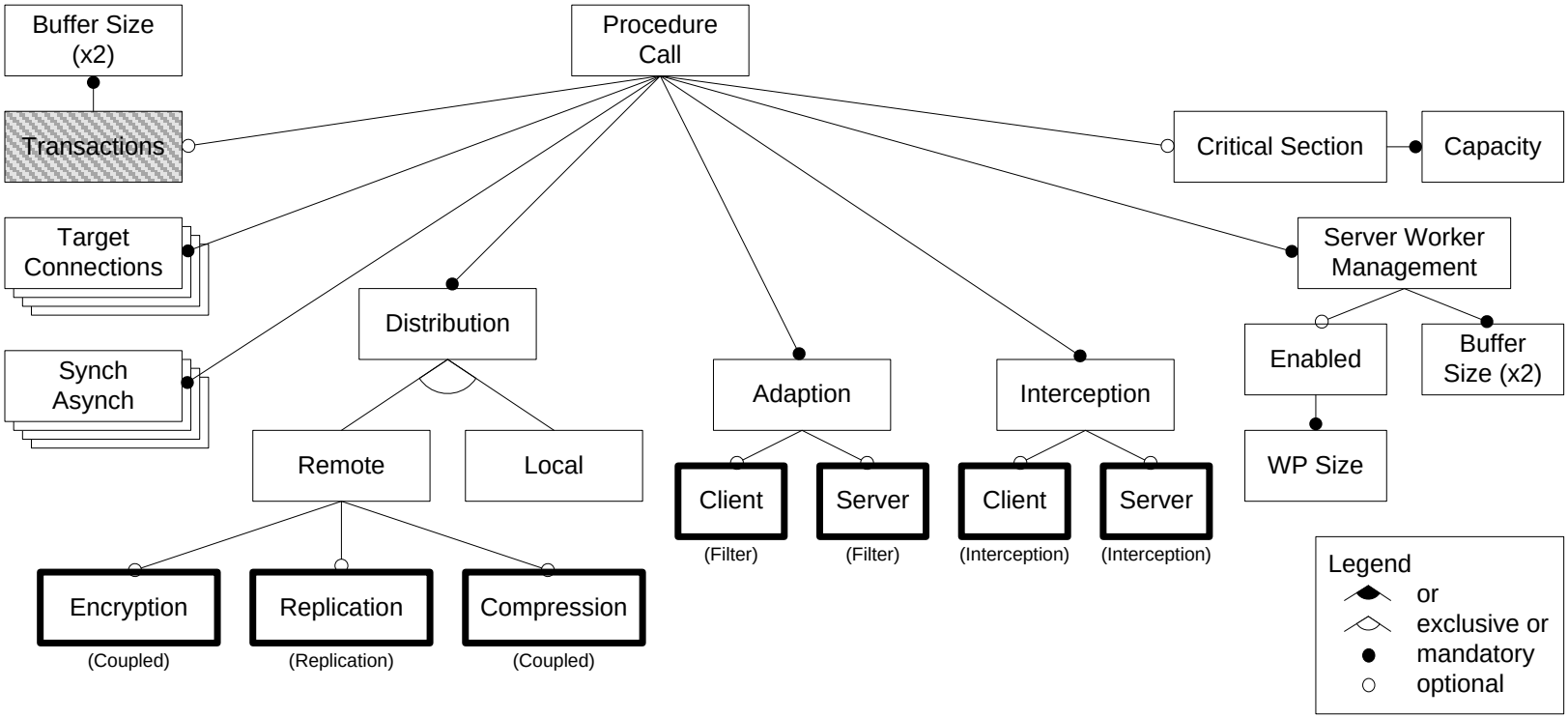
Appendix: Procedure Call Connector Server Unit Architecture



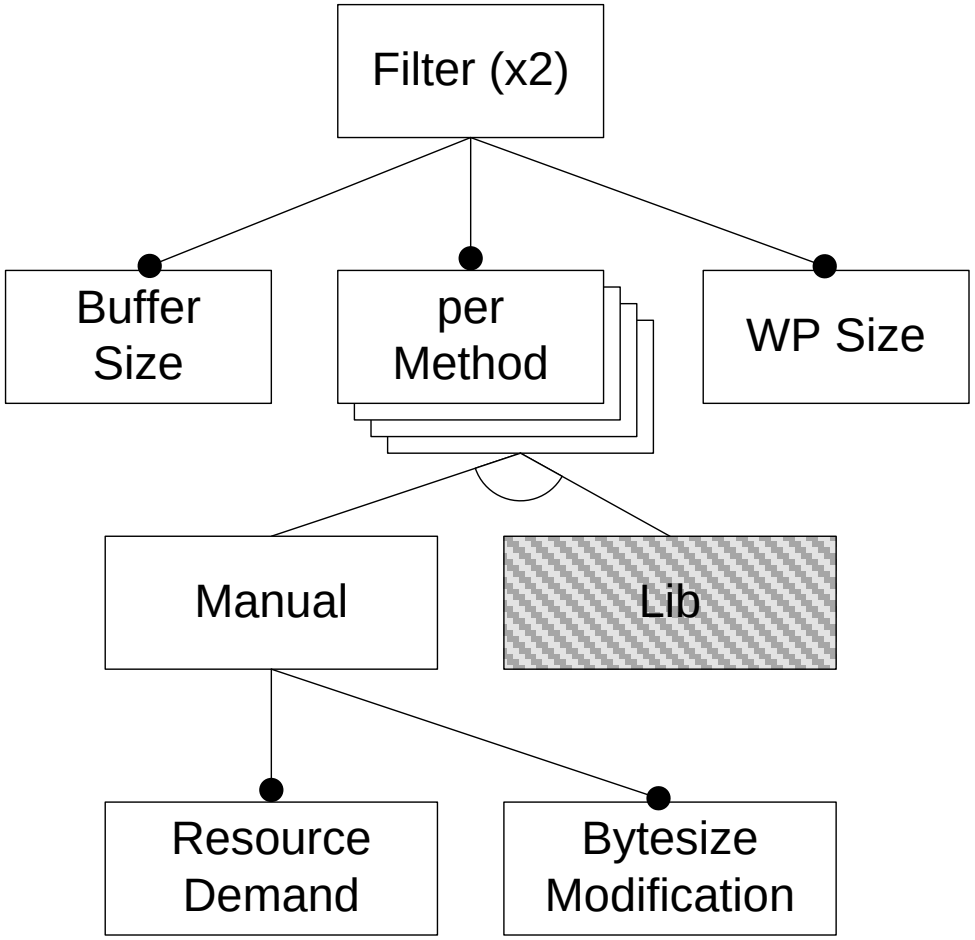
Appendix: Worker Management



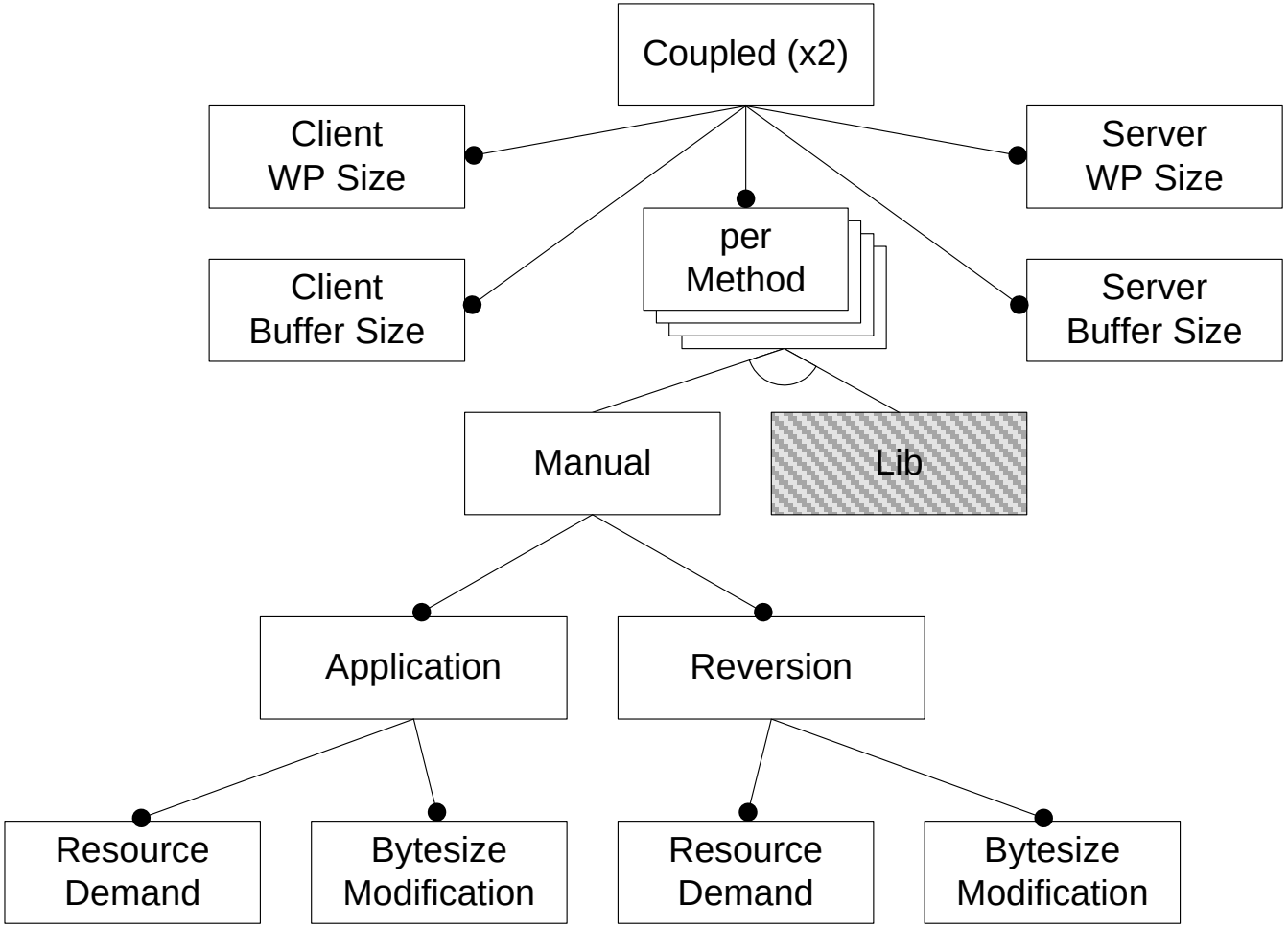
Appendix: Procedure Call Connector Feature Tree



Appendix: Filter Feature Tree



Appendix: Coupled Activity Feature Tree



Appendix: Interception & Replication Feature Tree