REWERSE
reasoning on the web

Roles in Evolution

or: Evolutionary Development

Uwe Aßmann
Technische Universität Dresden
Chair of Software Engineering

CS Department in Dresden

OUTPUT Demo Day of the Department

REWERSE
reasoning on the web

Filters in Evolution

or: Evolutionary Development

Uwe Aßmann
Technische Universität Dresden
Chair of Software Engineering

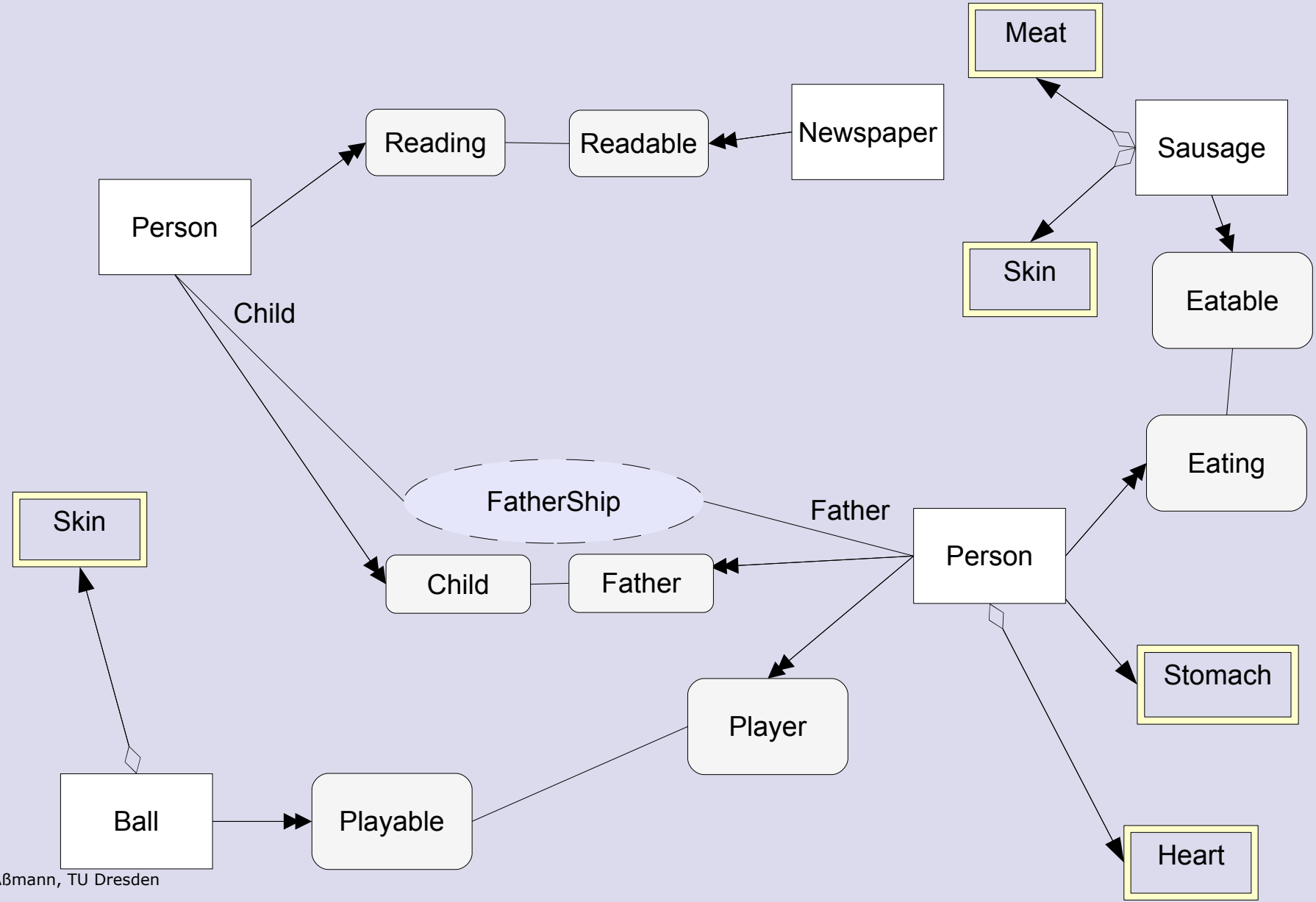
My Sabbatical

- Proposals, ...

Steven McMenamin

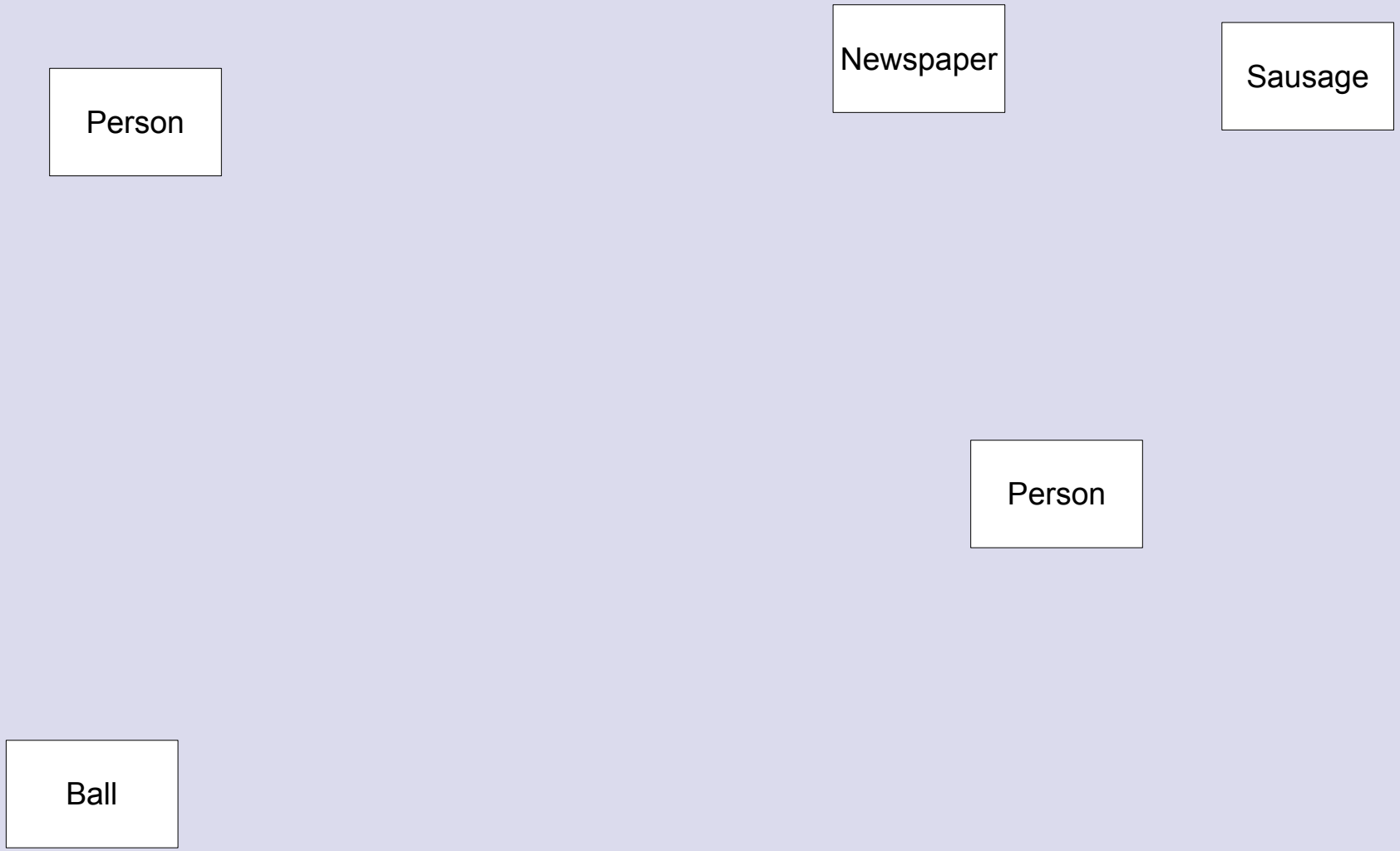
Slicing, e.g. with Role Models

An Analysis Model



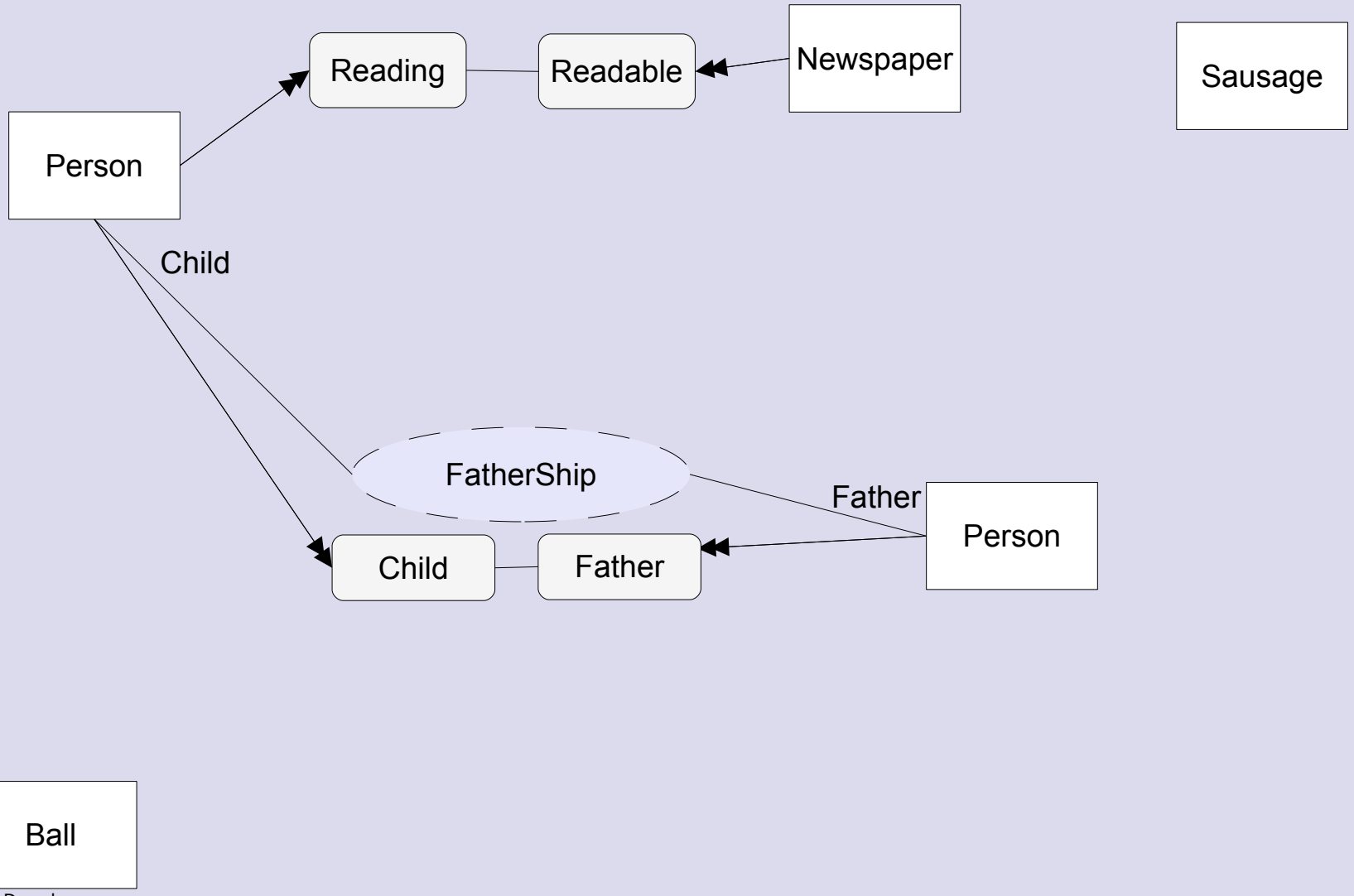
Refinement of Natural Objects by adding Collaborations

- Initial State: Identification of natural types of domain model



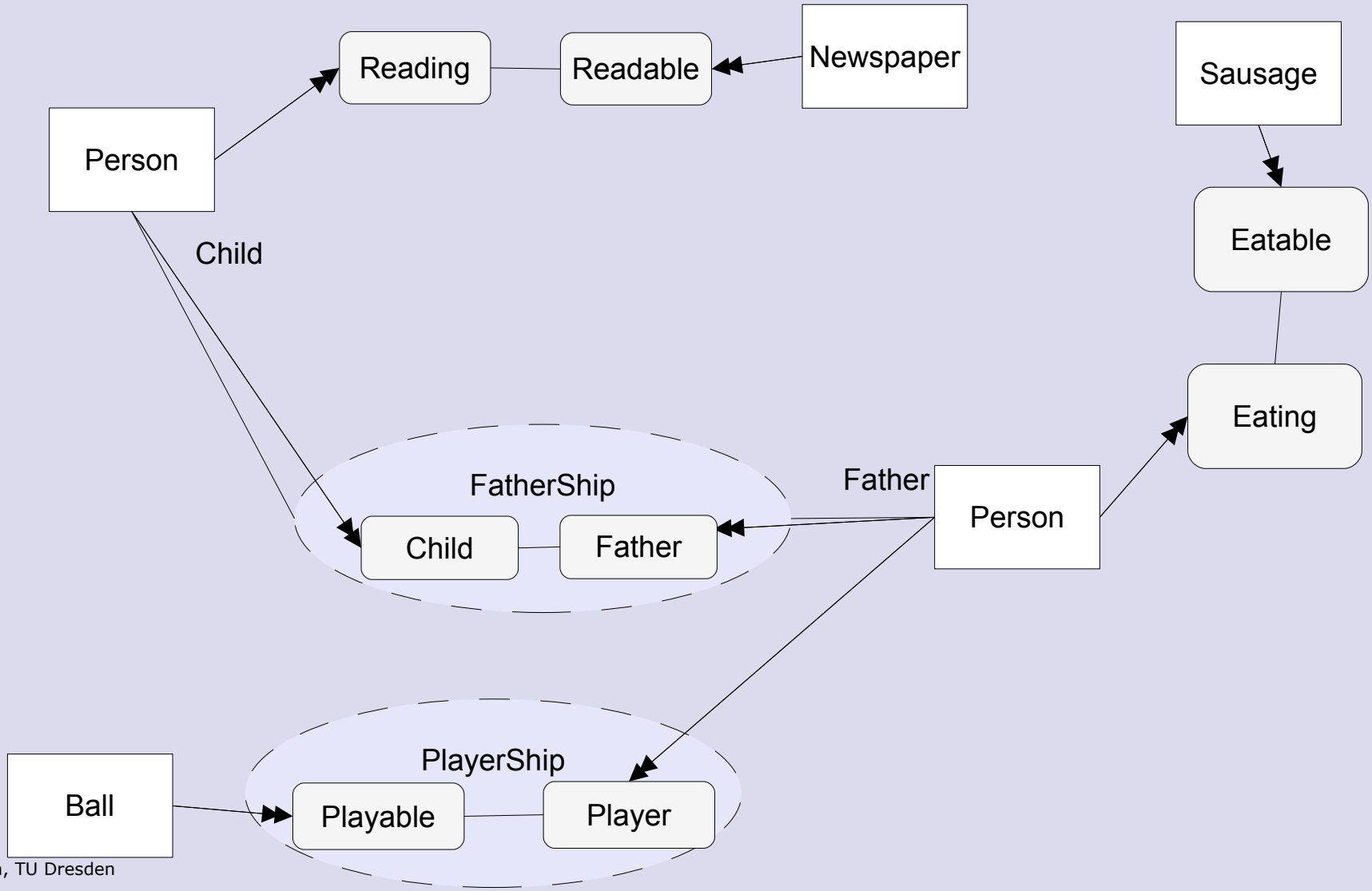
Refinement of Natural Objects by adding Collaborations

Extension by collaborations (connectors, teams)



Refinement of Natural Objects by adding Collaborations

Extension by collaborations (connectors, teams)



Implementation of Collaboration Schemas

Collaboration schemas are *type functors*

- Functions on types

Direct implementation

- Languages: ComposeJ, ObjectTeams, CeasarJ
- Mixin layers
- Role Object Pattern
- Semantic macros
- Generic templates (BETA, Compost)
- Aspects

Rewriting to standard languages

- Mapping (MDD process), e.g., with graph rewriting systems

REWERSE
reasoning on the web

Essence, Administration and Infrastructure

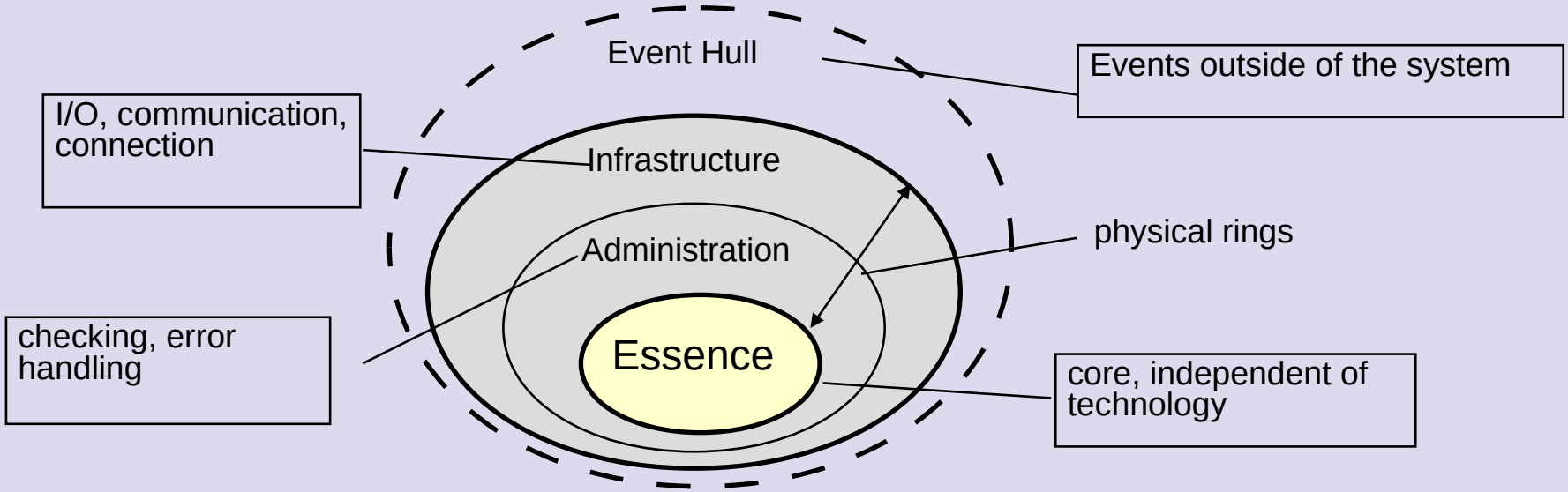
- **Structured Analysis of Second Generation**
- **McMenamin, S., Palmer, J.: Essential Systems Analysis, Yourdon Press, 1984.**

Essence Decomposition

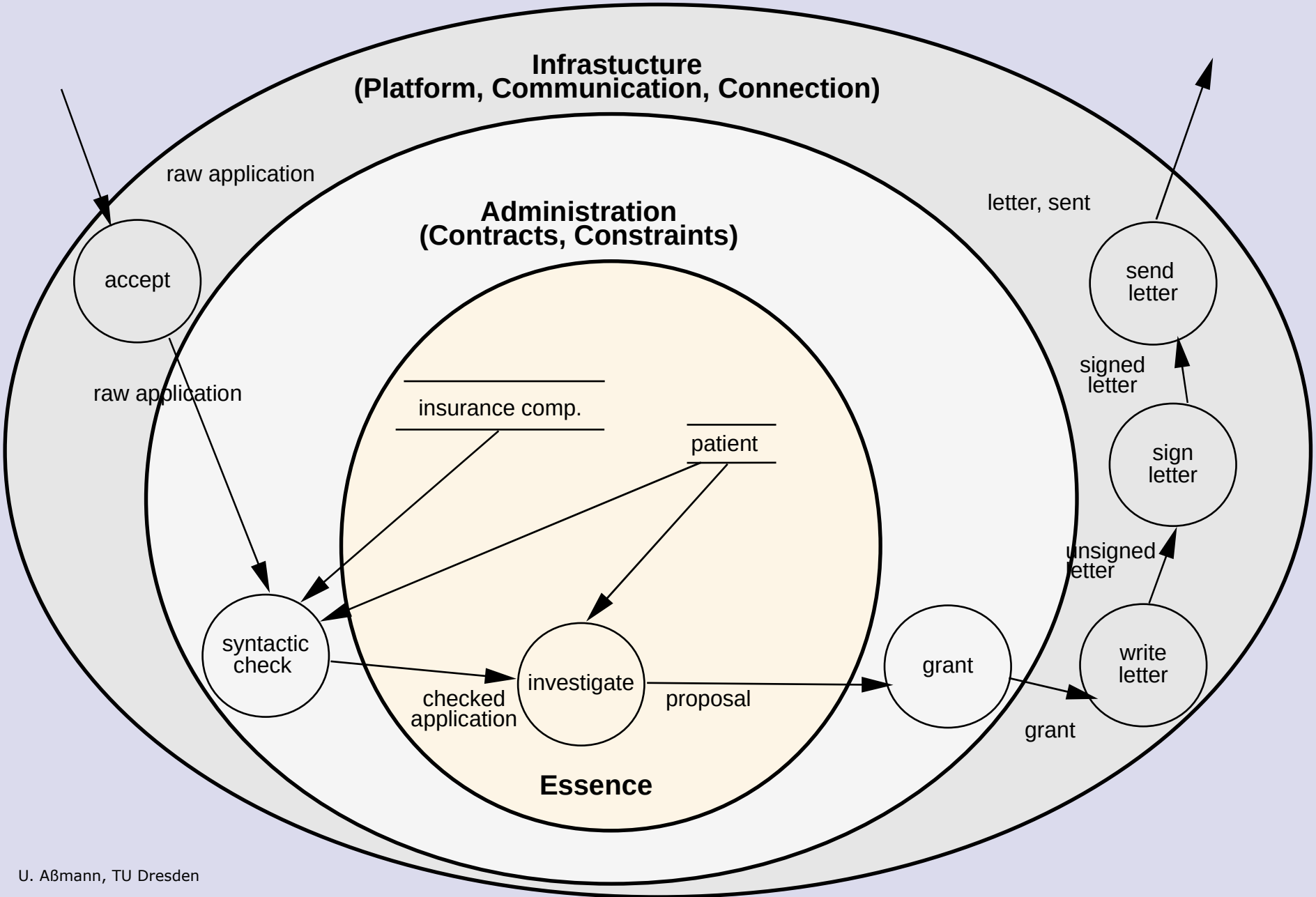
Decomposition for Essence identifies those parts of a system that is not dependent on any technology.

- assuming perfect technology

3 aspects: E-A-I



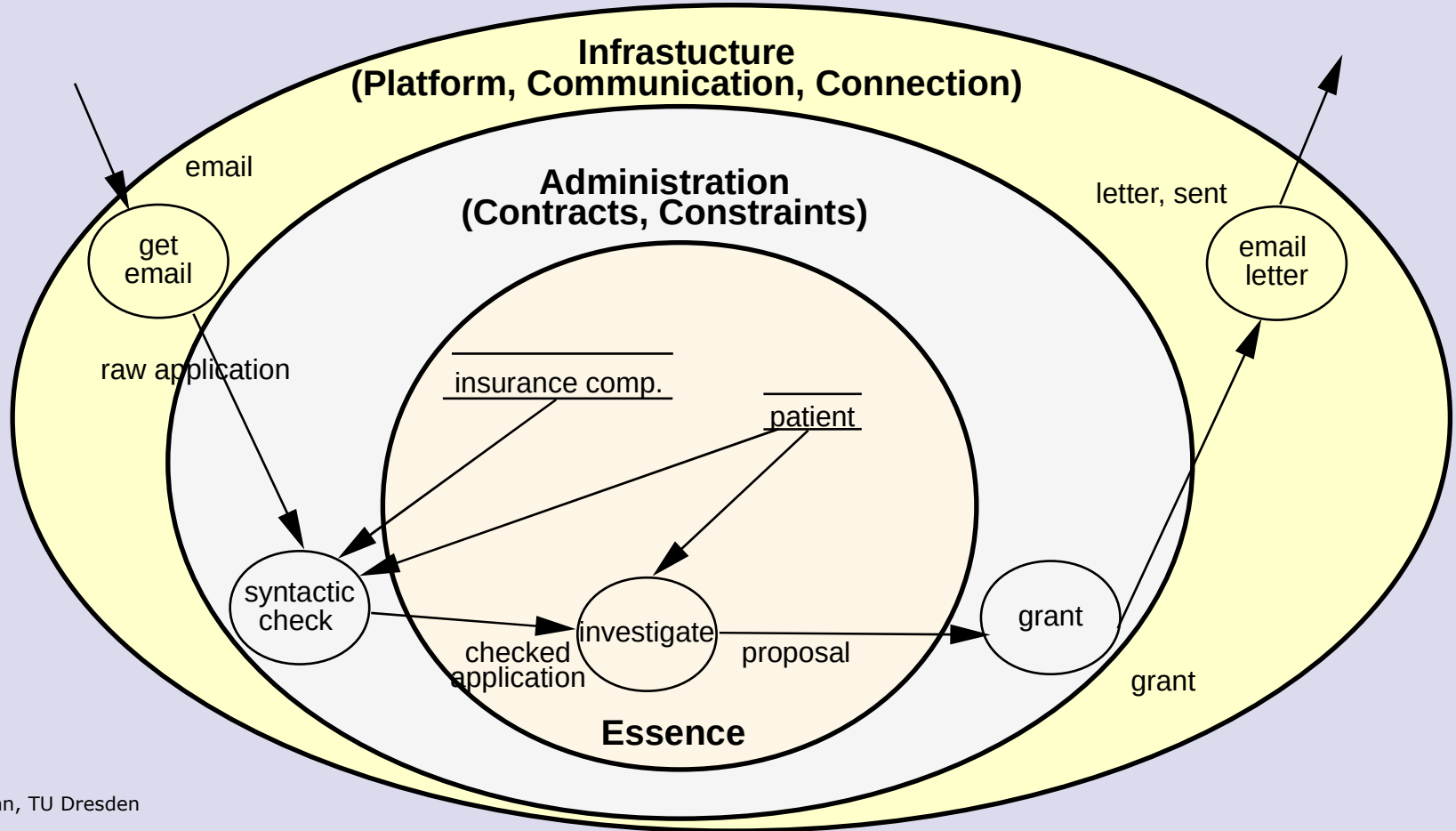
E-A-I of a Business Process



Use of E-A-I in Evolution

Administration and Infrastructure can change. Here: exchange infrastructure

That's how to evolve



E-A-I and MDA

Similar goals:

- Technology (platform) independence
- Reuse of essential parts
- Technology variation (platform change)
- Separation of concerns

REWERSE©
reasoning on the web

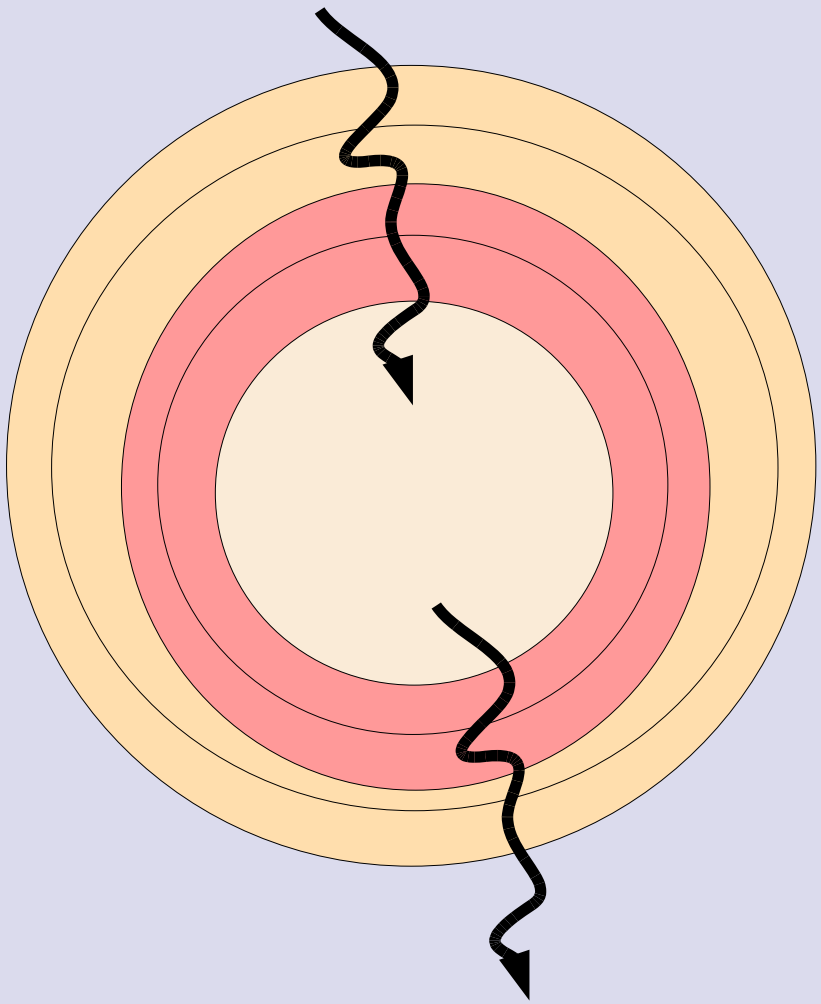
Essence, Administration and Infrastructure in Objects

- **How to enrich E-A-I with OO?**

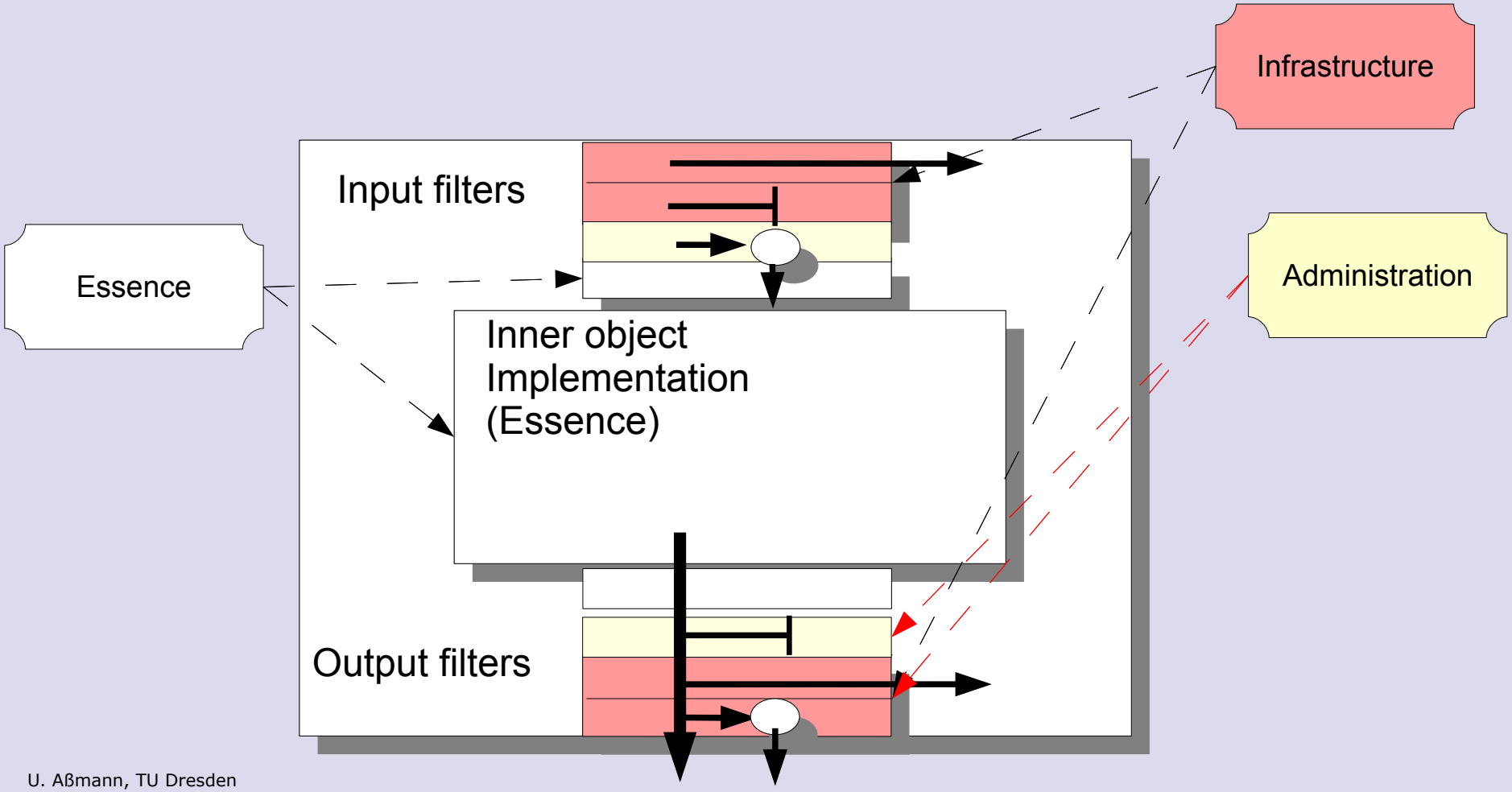
Combine Composition Filters with E-A-I

- Composition Filters (CF) wraps objects with *filters*
  - Messages flow through the filters
  - are accepted or rejected
  - are modified by them

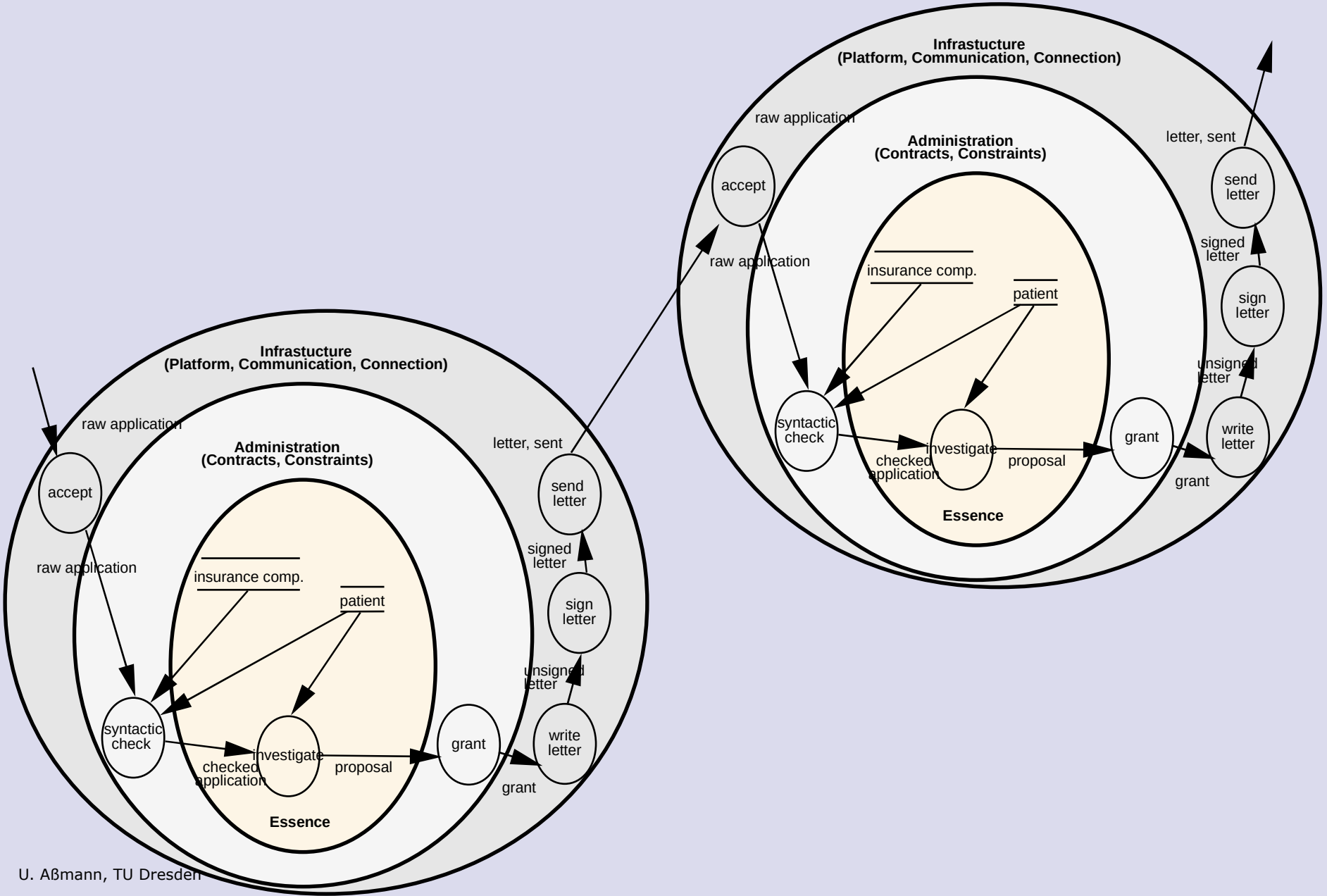
Composition Filters an ideal form of object-oriented E-A-I.



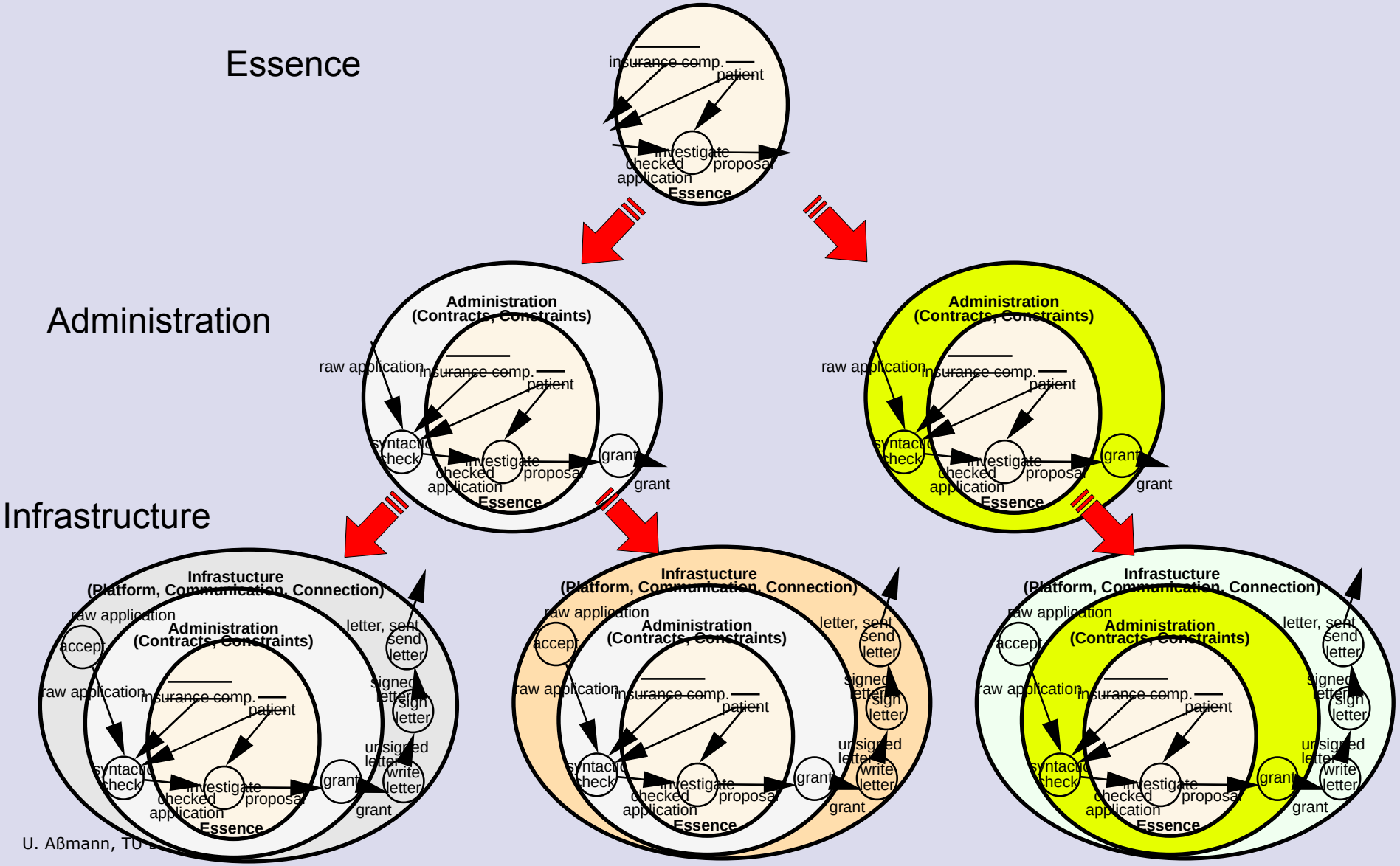
Filters can be related to E-A-I Concerns



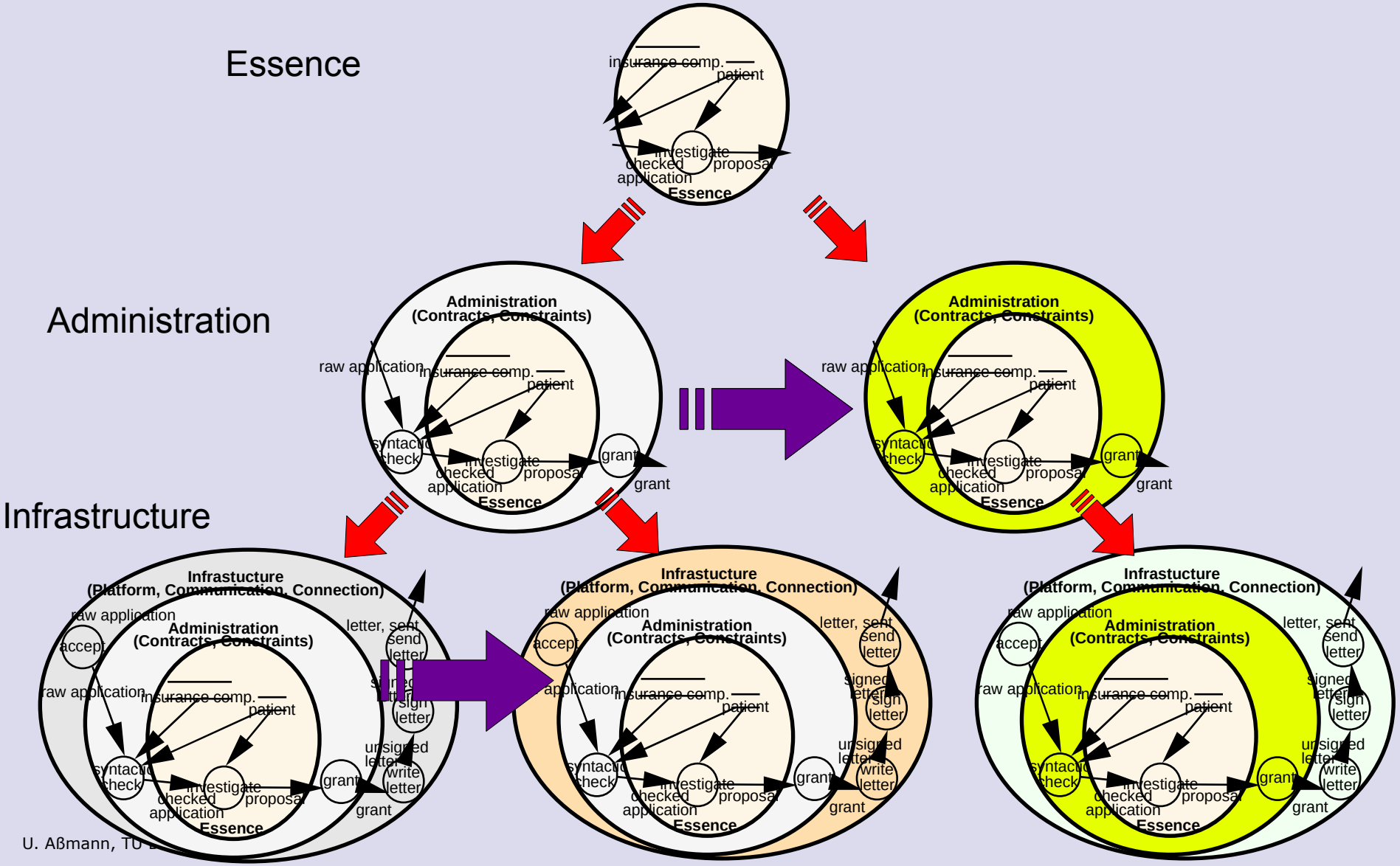
E-A-I of Composition Filters



MDA of Composition Filters, with E-A-I



Evolution of Composition Filters, with E-A-I

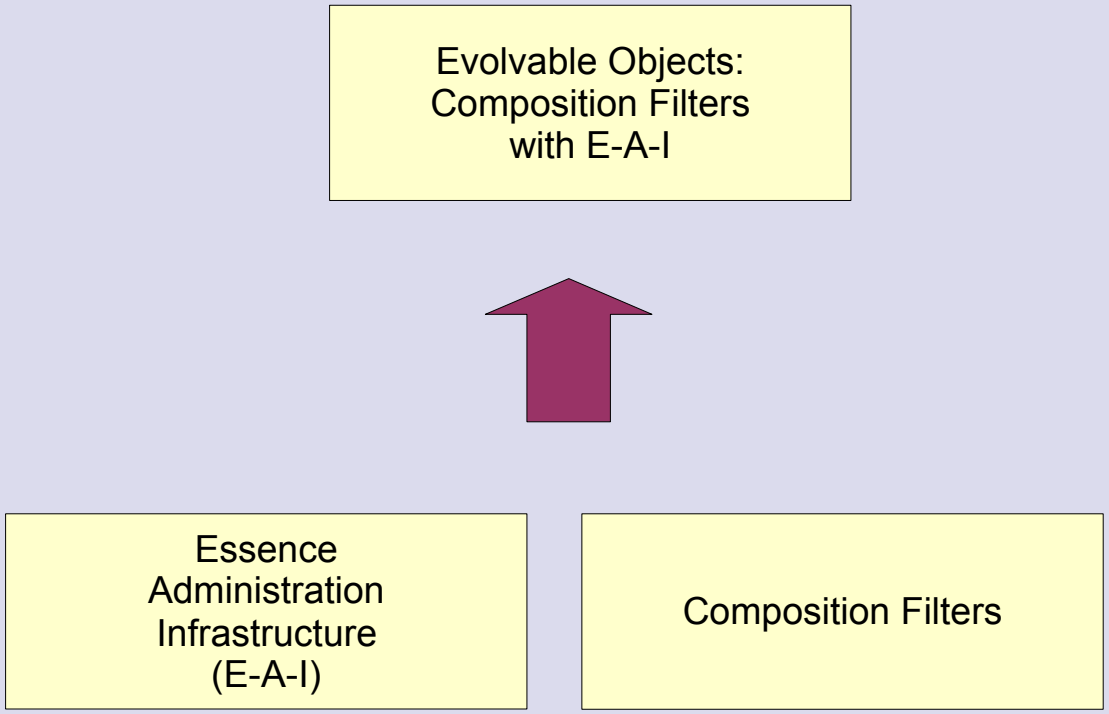


Evolution is...

the change of
Infrastructure and administration filters

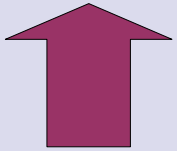
the change of
essence (filters and inner objects)

Evolvable Objects: Filters with Variability (E-A-I concerns)



Evolvable Objects: Filters with Variability (E-A-I concerns)

Evolvable Objects: Composition Filters with E-A-I



Essence Administration Infrastructure (E-A-I)

Composition Filters

REWERSE
reasoning on the web

The End

REWERSE Working Group I3 (*Composition and Typing*) **http://www.rewerse.net**

TUD ST group	**http://st.inf.tu-dresden.de**
Reuseware toolset	**http://www.reuseware.org**
ModelPlex project	**http://www.modelplex.org**
MOST project	**http://www.most-project.eu**