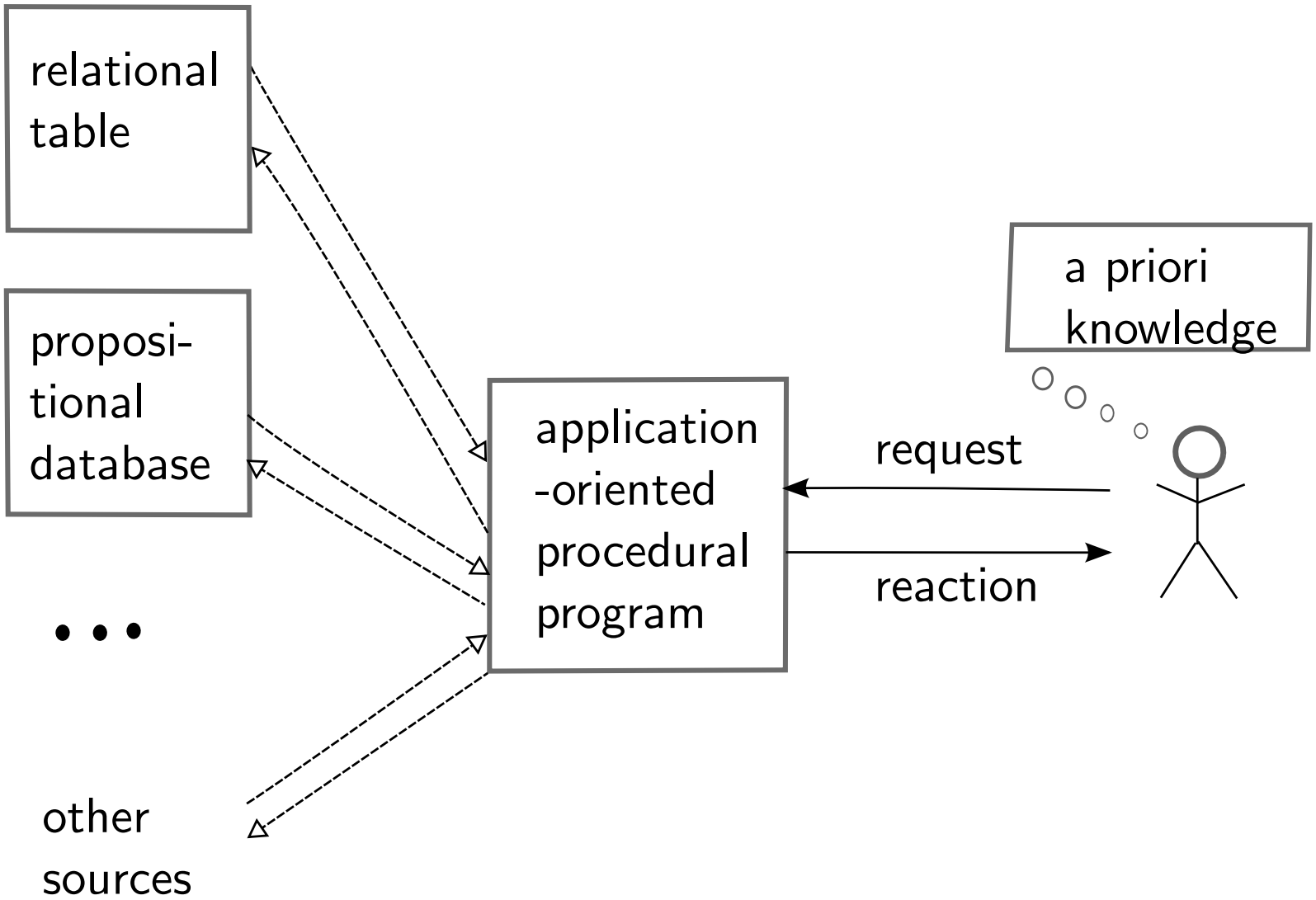
Inference-proof program-based mediation of data sources

Joachim Biskup *Cornelia Tadros*

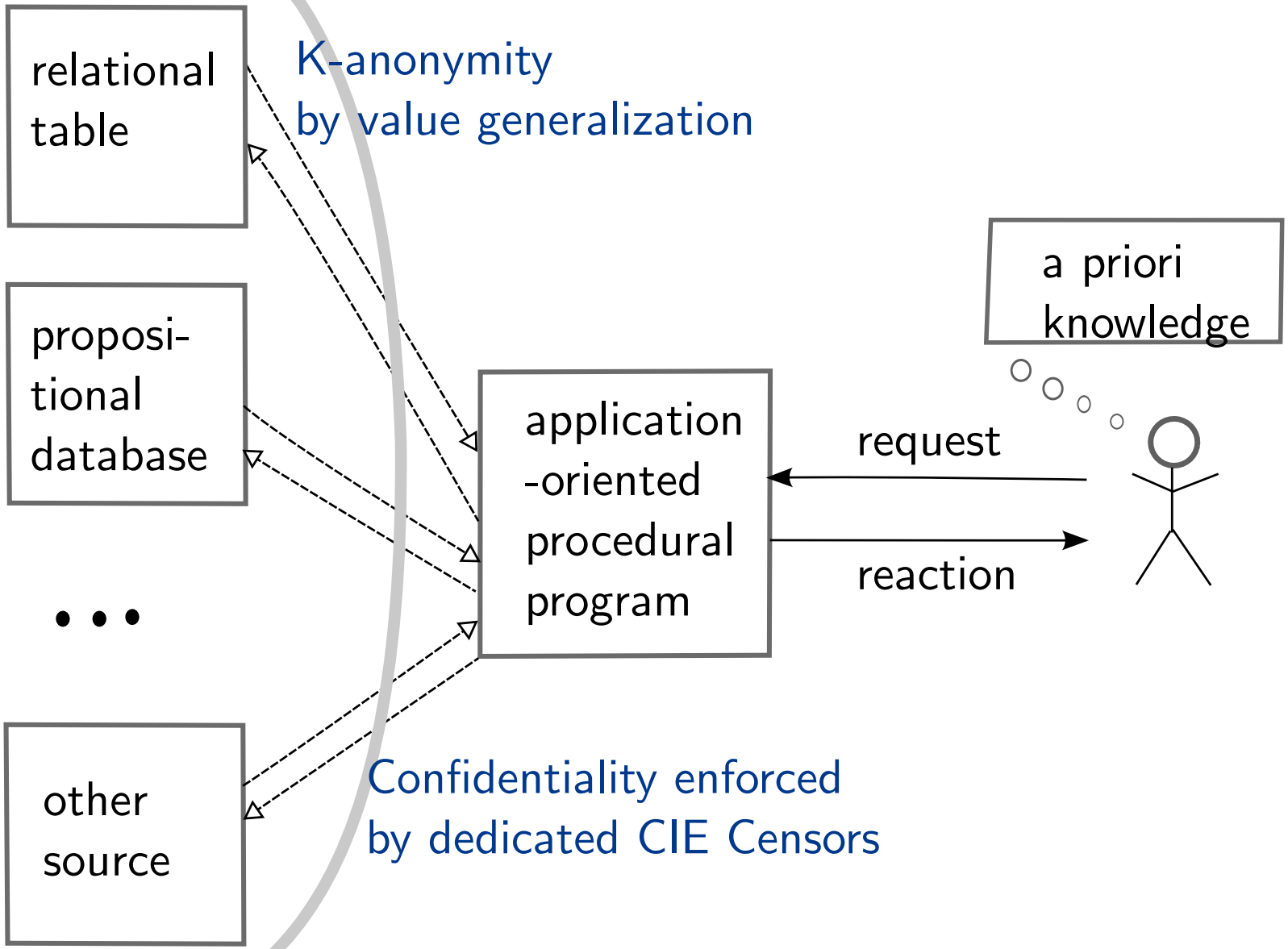
TU Dortmund
Faculty of Computer Science
Information Systems and Security – ISSI

17. Feb. 2016

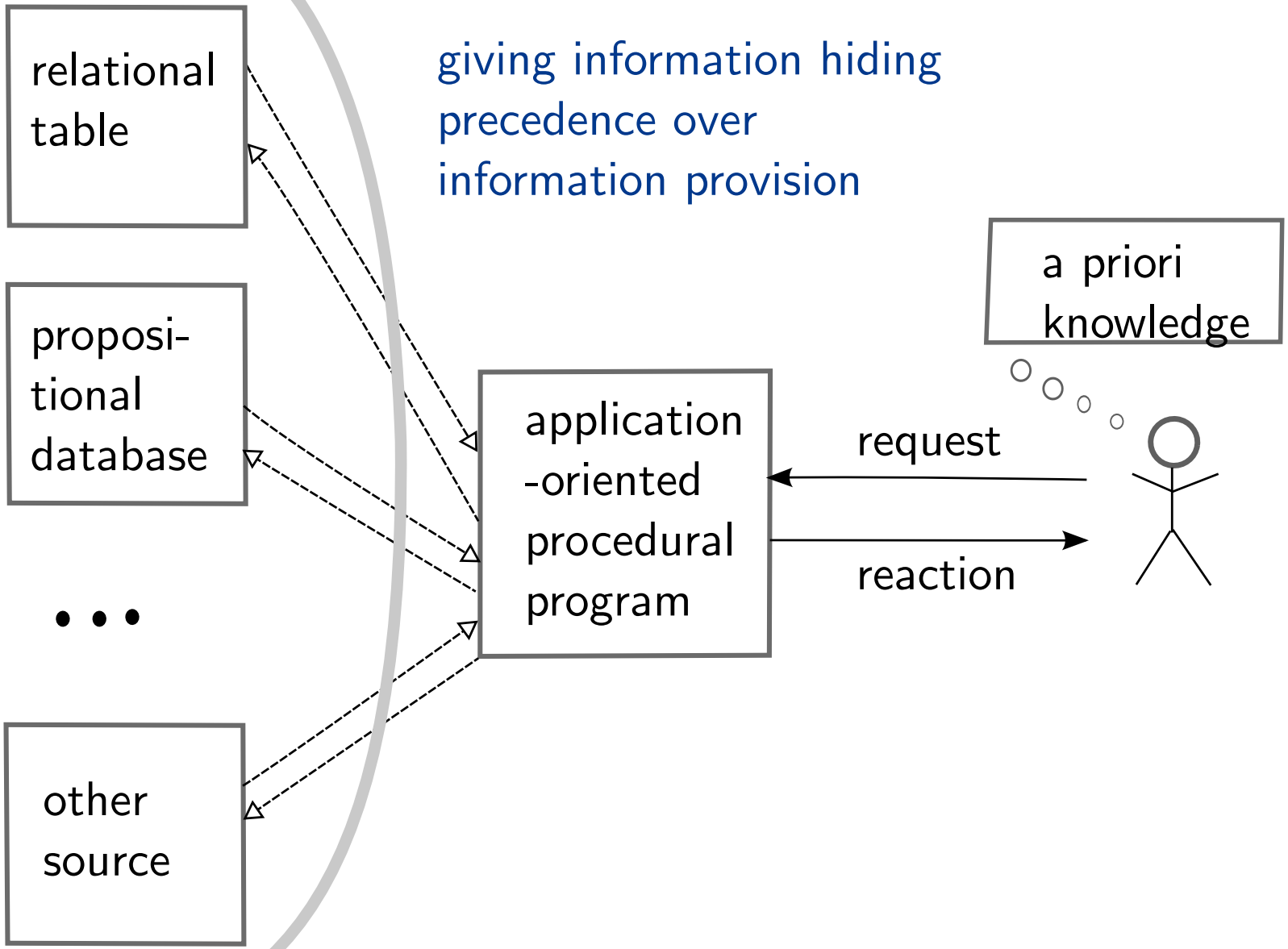
A Challenging Scenario



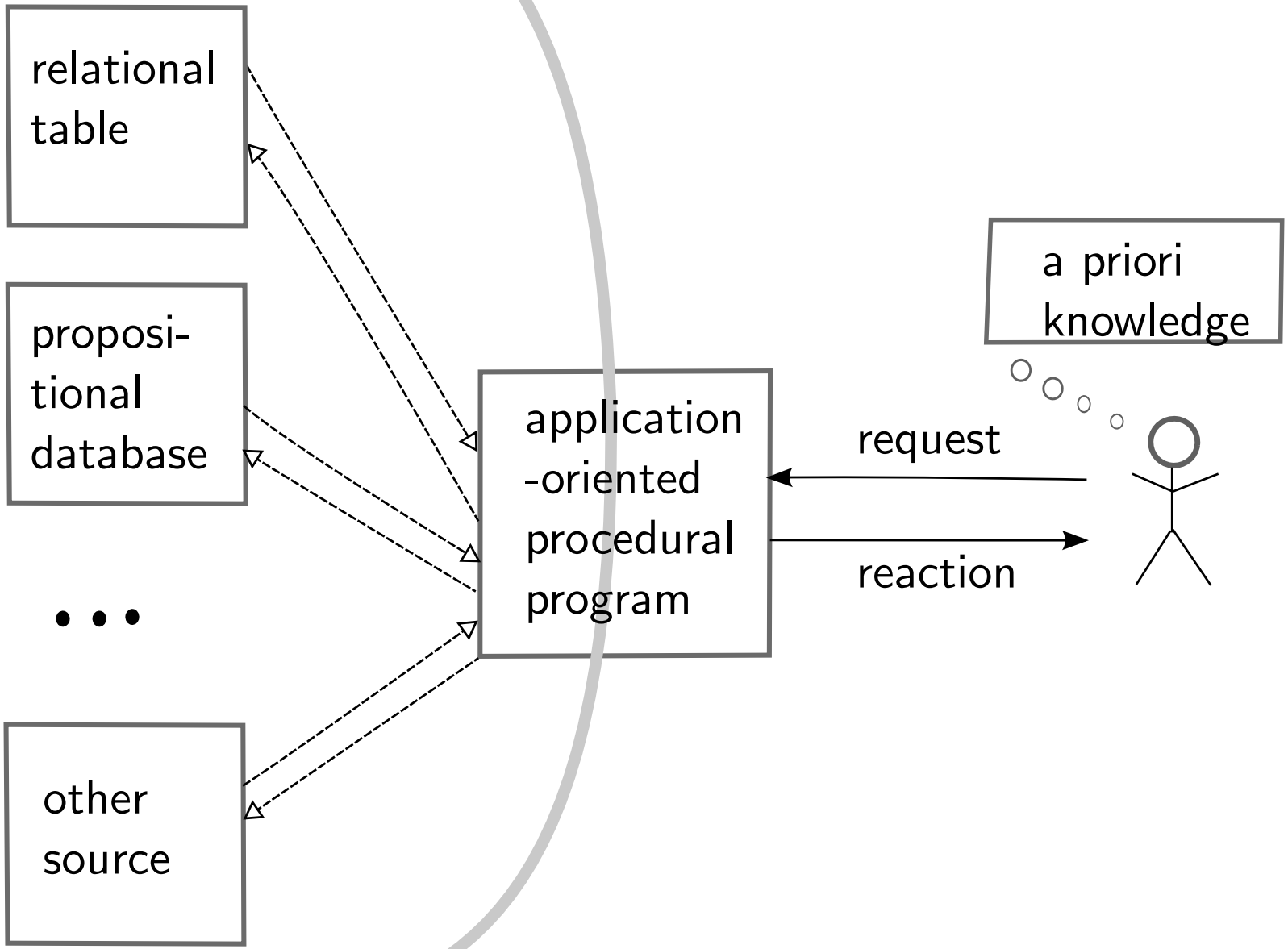
A Straightforward Solution



A Straightforward Solution



A Solution to Balance the Hiding/Provision Trade-Off



Existing Technologies

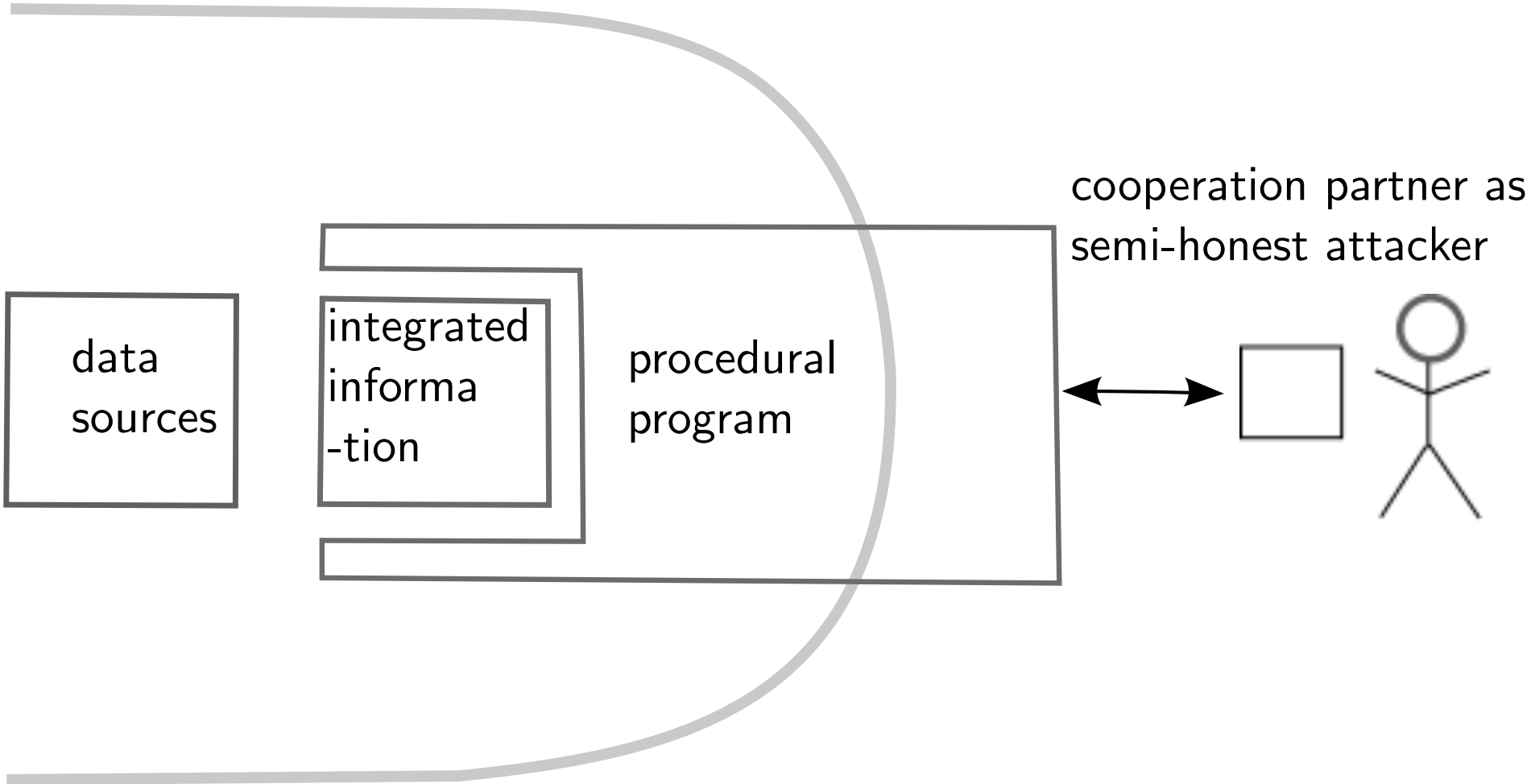
- Logic-oriented belief forming
- Adversarial reasoning
- Information system integration and mediation
- Logic-based inference control
- Language-based information flow control and declassification

Talk Based on

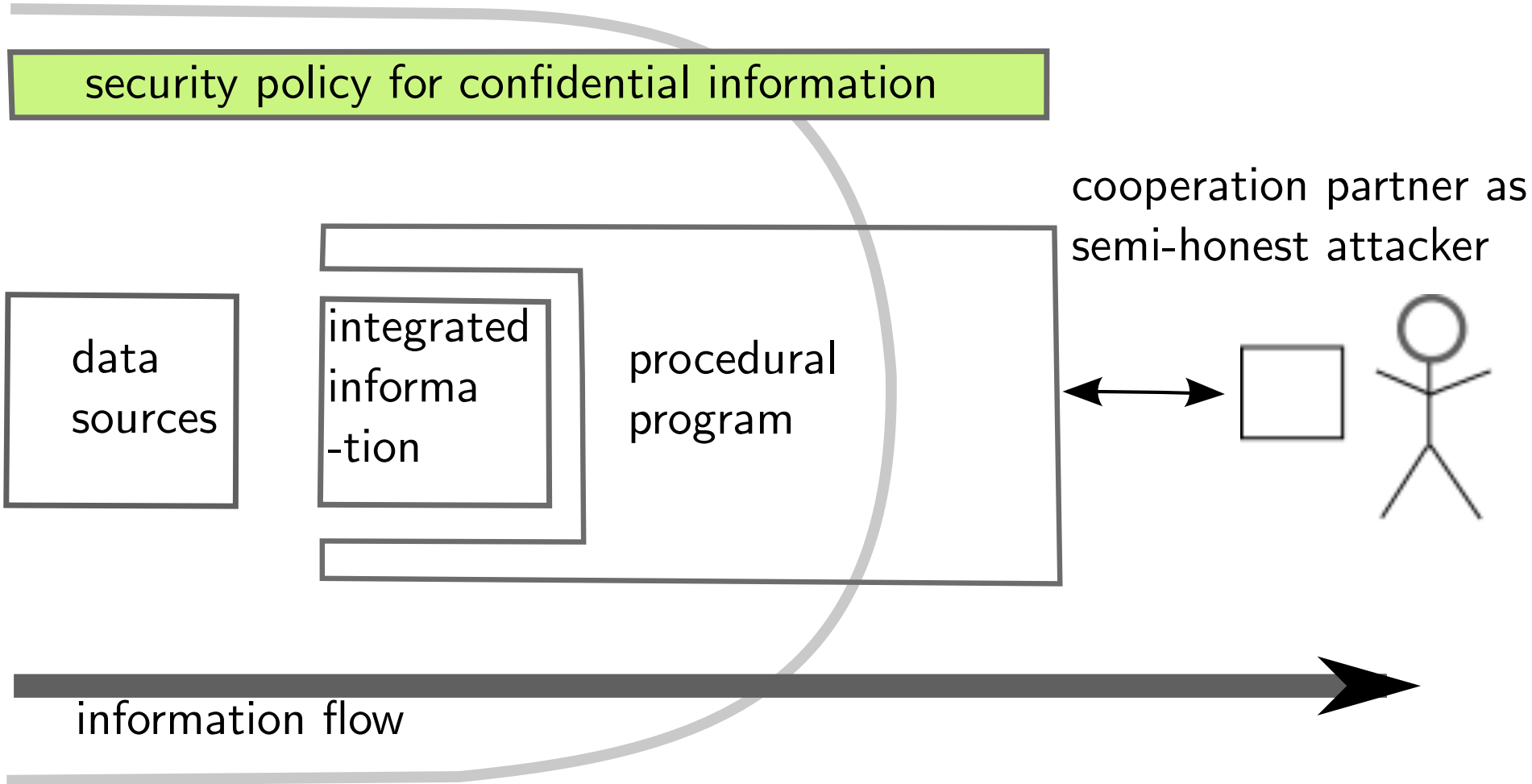
- Joachim Biskup, Cornelia Tadros:
  Constructing Inference-Proof Belief Mediators.
  Published in Data and Applications Security and Privacy, 2015.
- Joachim Biskup, Cornelia Tadros:
  Confidentiality Enforcement by Hybrid Control of Flows
  from Abstract Information States
  through Program Execution via Declassification.

Overview

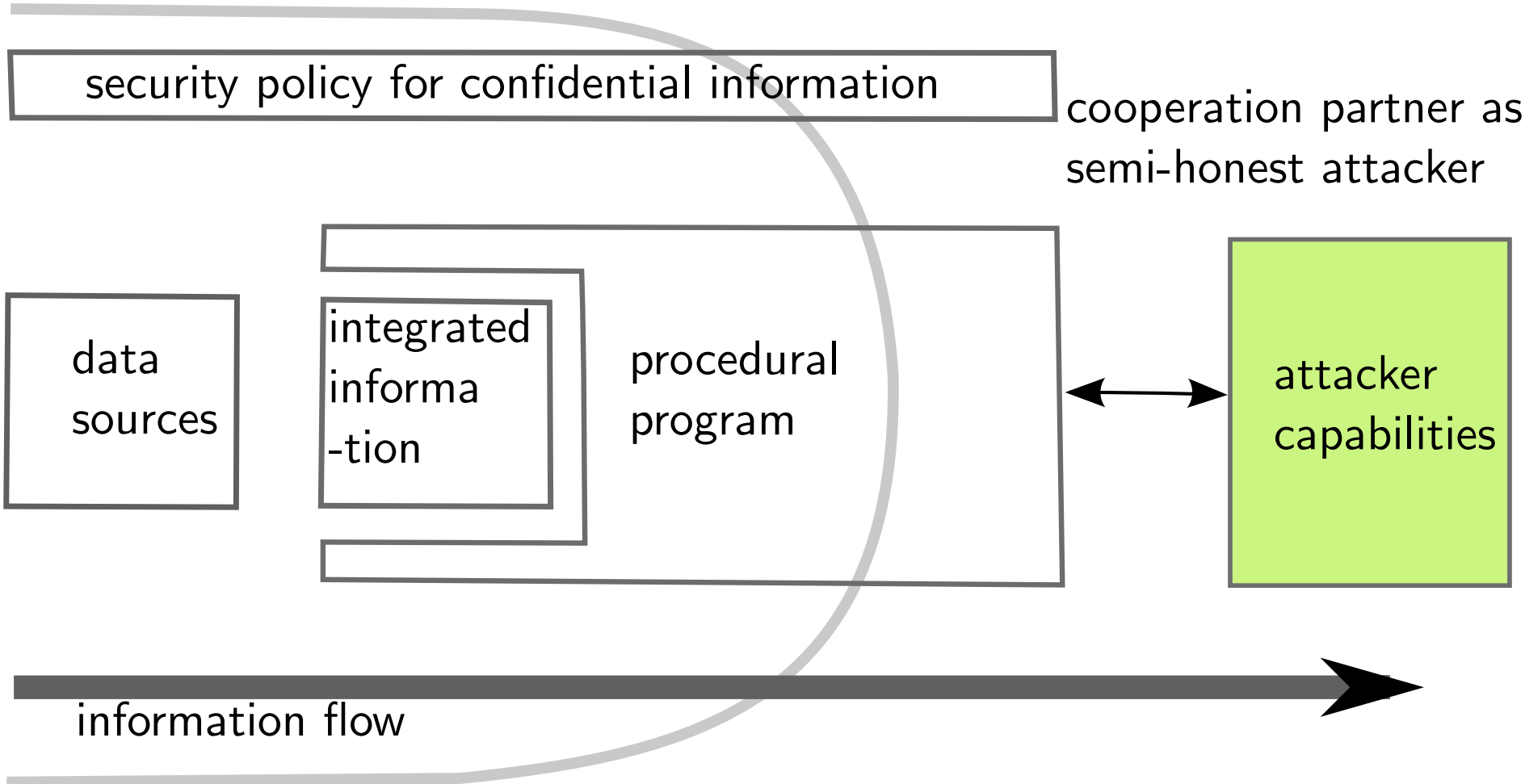
Inference-Proof Information Mediator



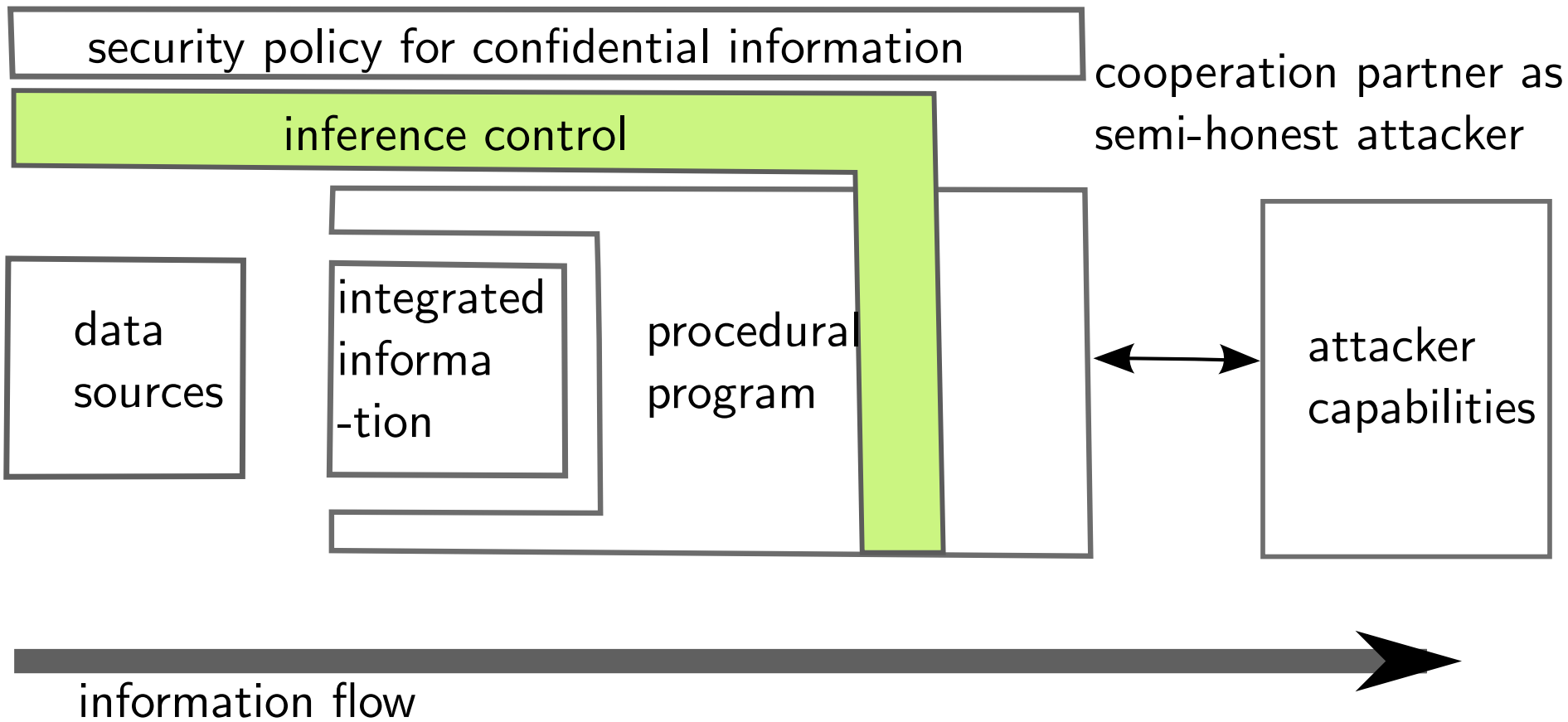
Inference-Proof Information Mediator



Inference-Proof Information Mediator



Inference-Proof Information Mediator

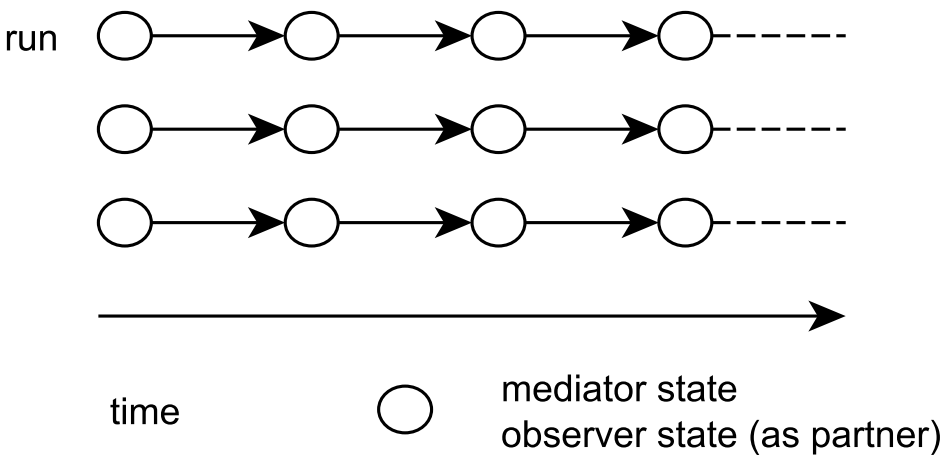


Main Requirements for the Mediator's Construction

- Confidentiality Policy:
  confidential pieces of information
  as sets $\mathcal{S}$ of abstract integrated information states
- Semantics:
  if actual integrated information state *ibs*
  is contained in such an $\mathcal{S}$,
  the cooperation partner must not know this
- Narrower Scenario
  according to assumptions

Semantics of the Confidentiality Policy

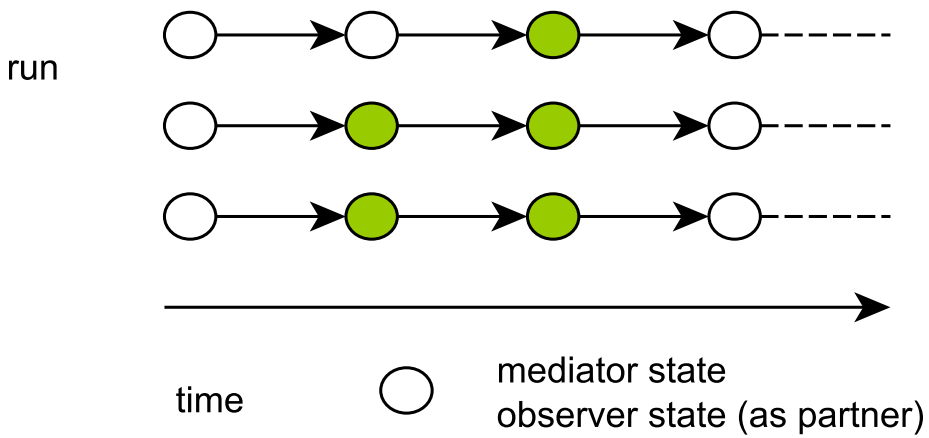
- semantics based on an abstract system model,
  here Runs & Systems



- mediator's functionality
  defined by runs
- observer models partner
  as an attacker,
  a skeptical reasoner

Semantics of the Confidentiality Policy

- policy protects confidential information against skeptical inferences by $\mathcal{K}$

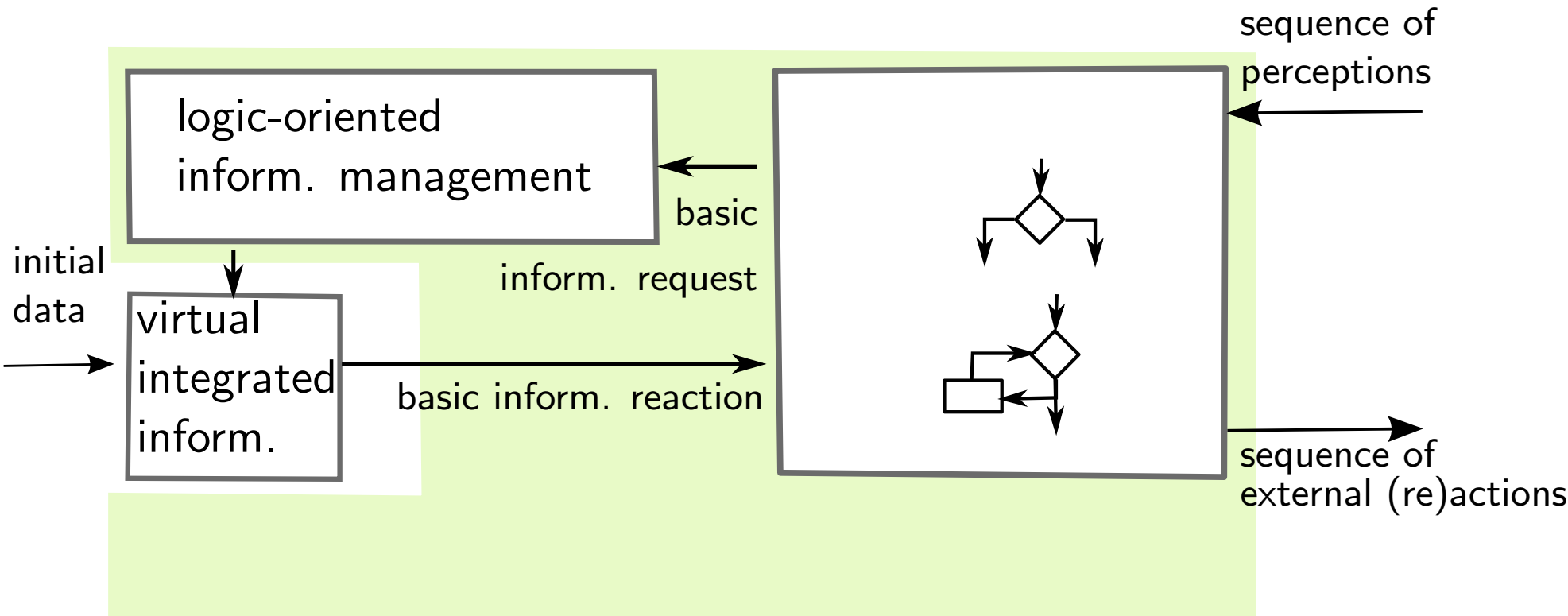


- $\mathcal{K}$ reasons on observations and background
- $\mathcal{K}$ models knowledge as a set of possible situations

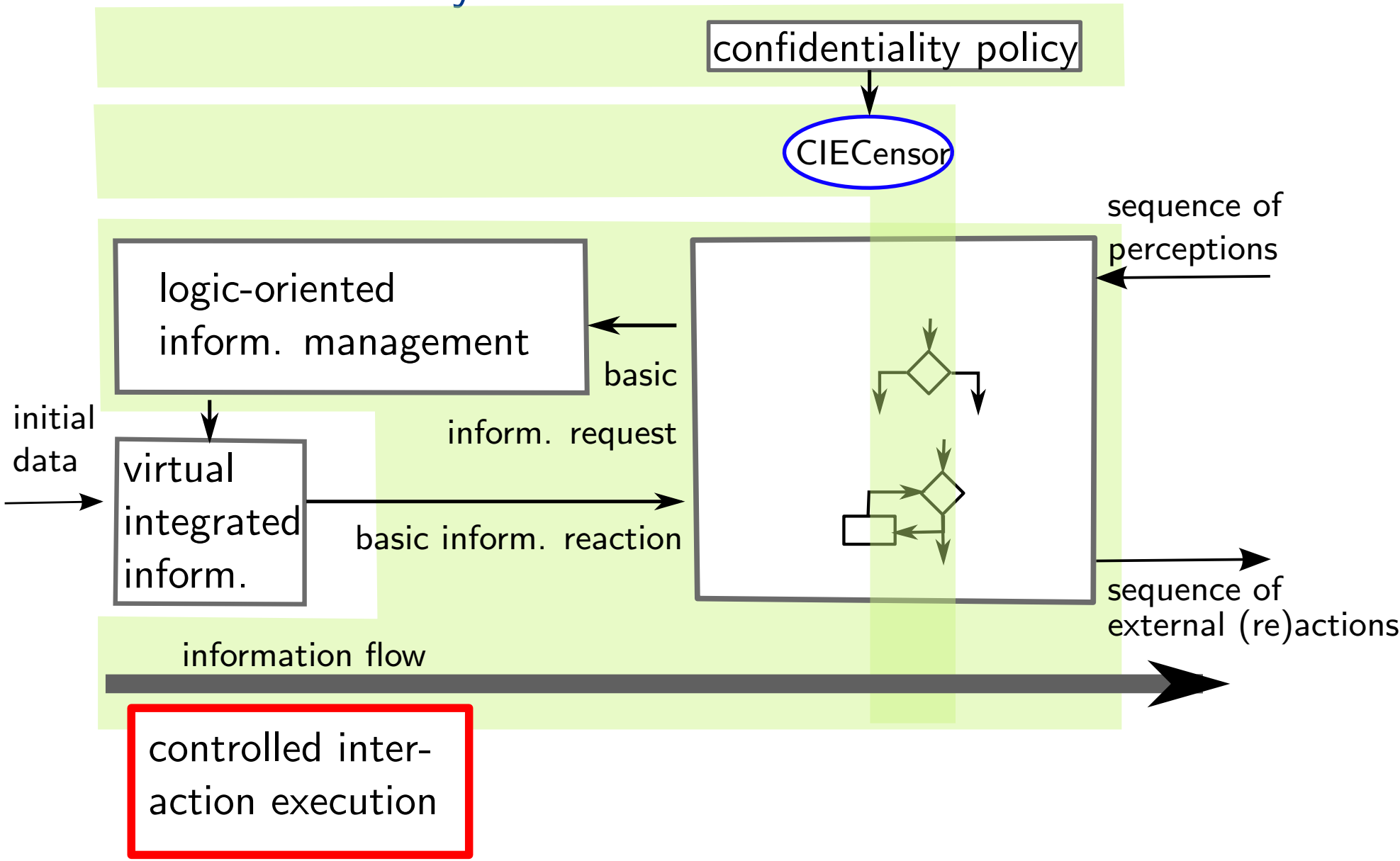
Property (Confidentiality Preservation)

For all runs r and times t and for all $\mathcal{S} \in$ pol it holds $\mathcal{K}(r,t) \not\subseteq \mathcal{S}$.

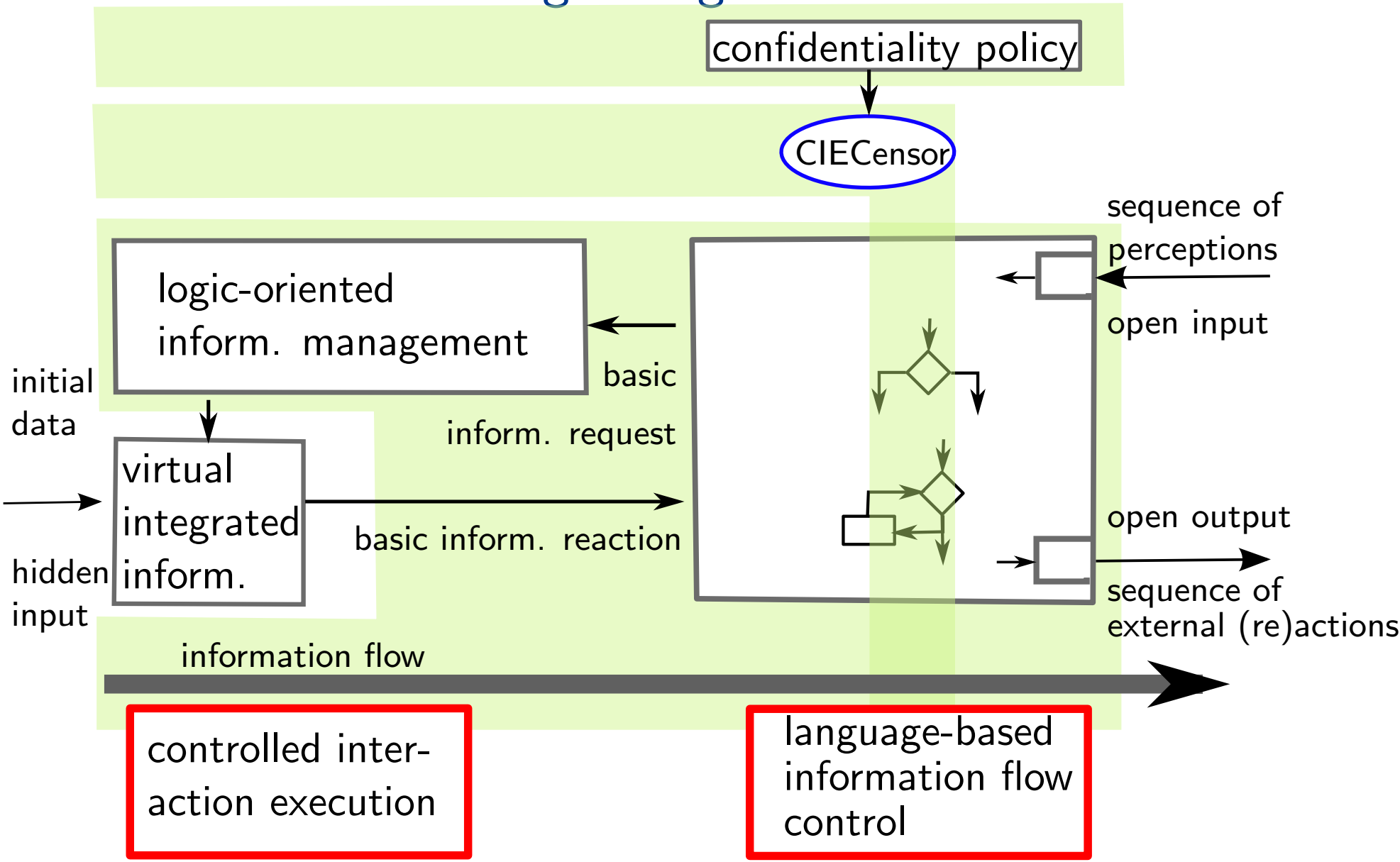
Basic Information Requests on Virtually Integrated Information



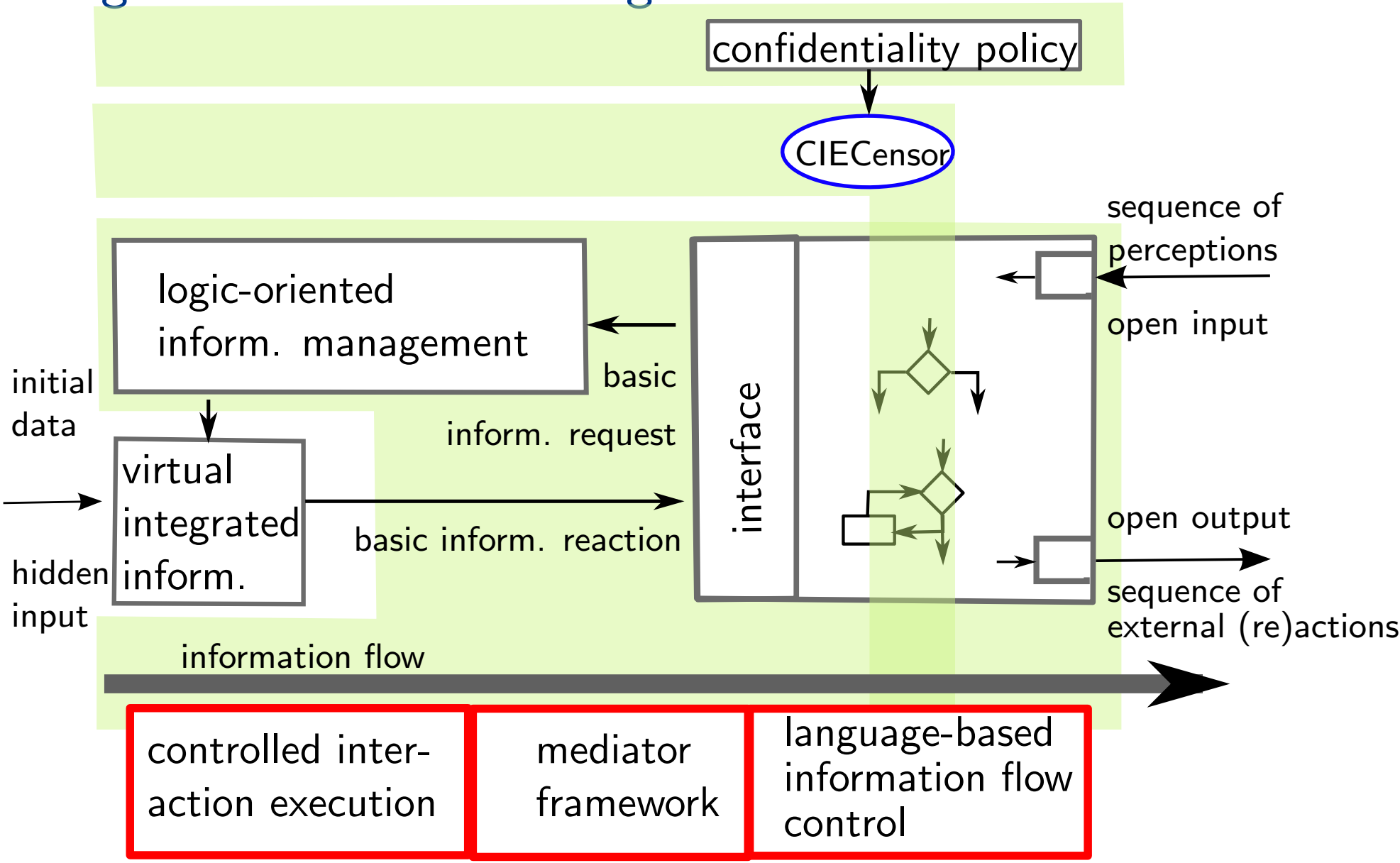
Inference Control by Means of a CIE Censor



Information Flow through Program Execution



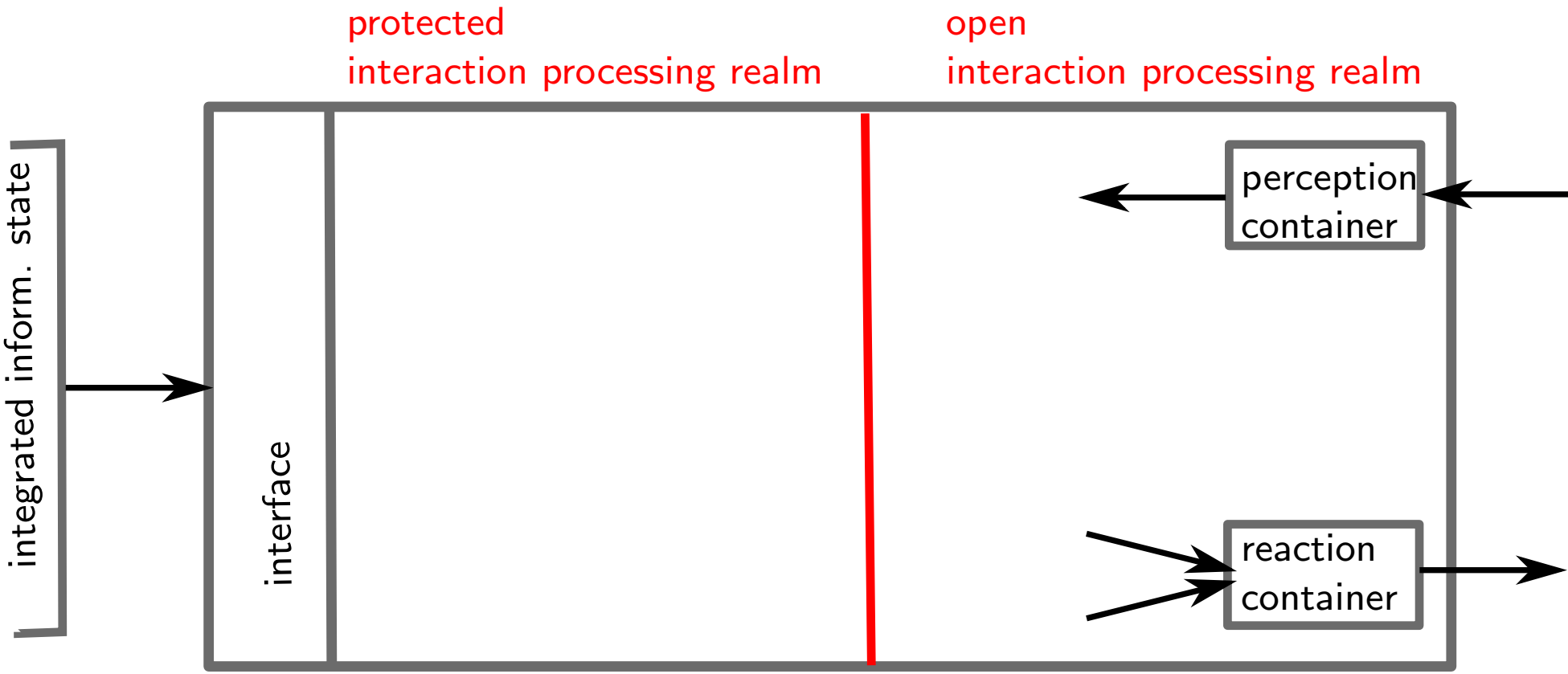
Integrated-Information-Program Interface for the Mediator



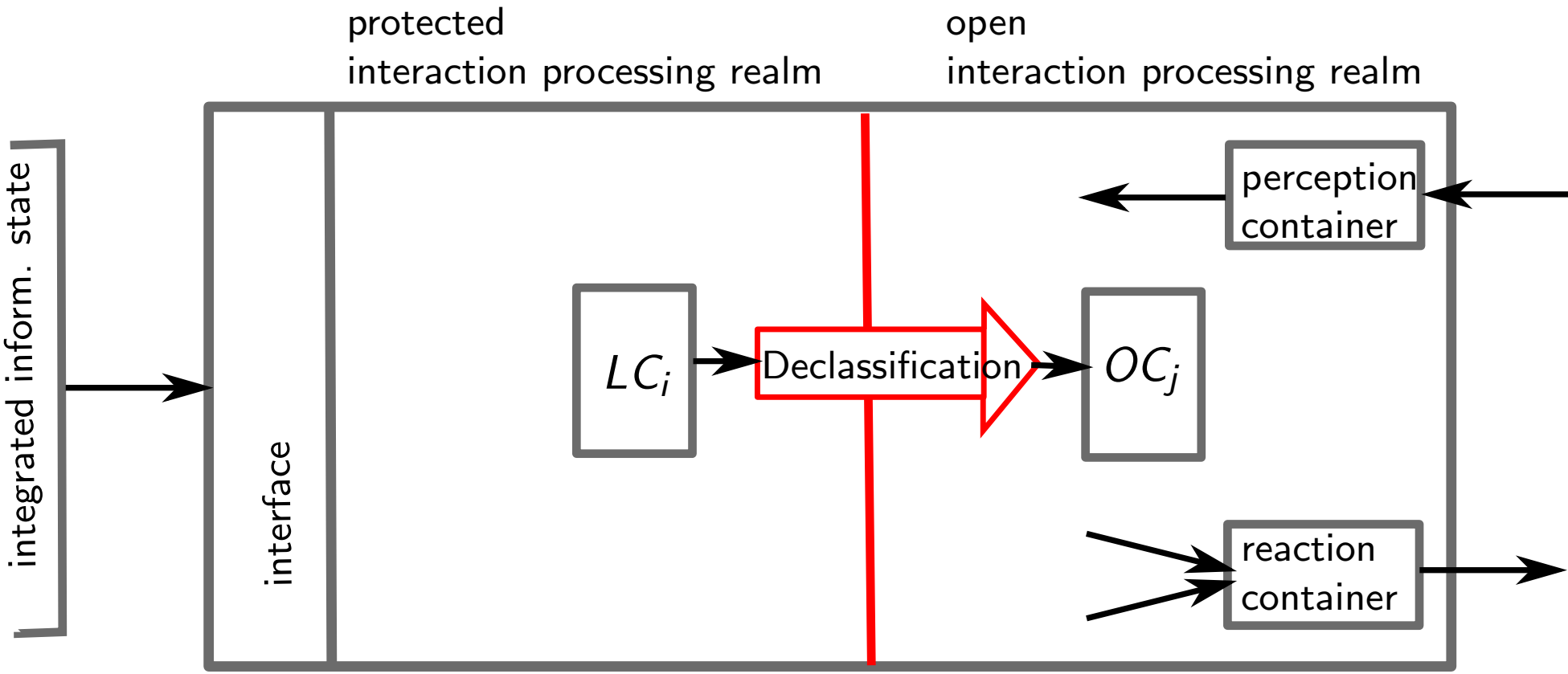
Mediator Framework for Unified Inference Control

1. Isolation by typing

1. Isolation by typing

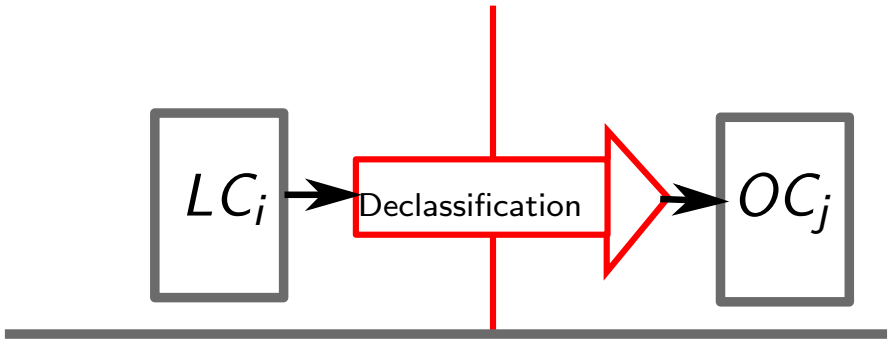


2. Declassification

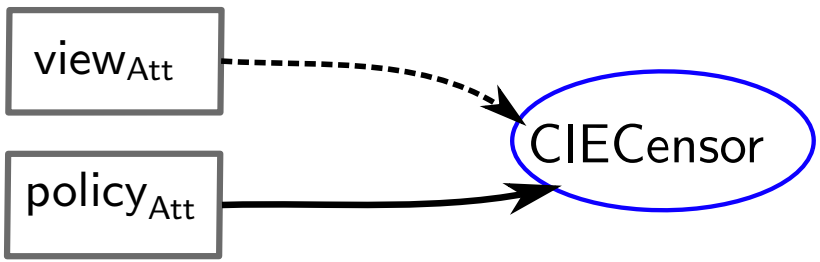


- the placement of declassification assignments is a policy for information provision
- it mainly trades off ressource efficiency for information provision or vice versa

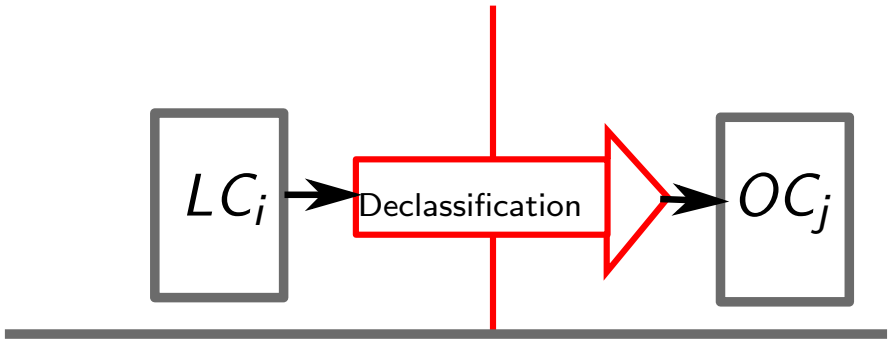
3. History-aware policy compliance



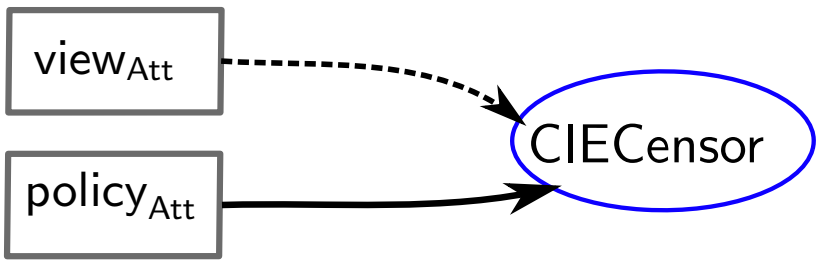
- Policy: sets $\mathcal{S} \subseteq \mathcal{IBS}$ of integrated inform. states



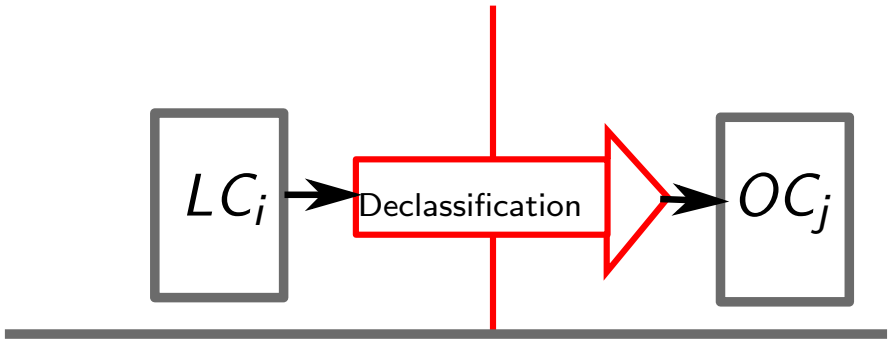
3. History-aware policy compliance

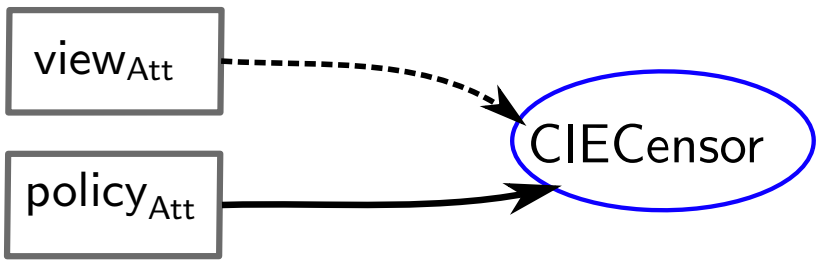


- Policy: sets $\mathcal{S} \subseteq \mathcal{IBS}$ of integrated inform. states
- History: previous view $\text{view}_{\text{Att}} \subseteq \mathcal{IBS}$



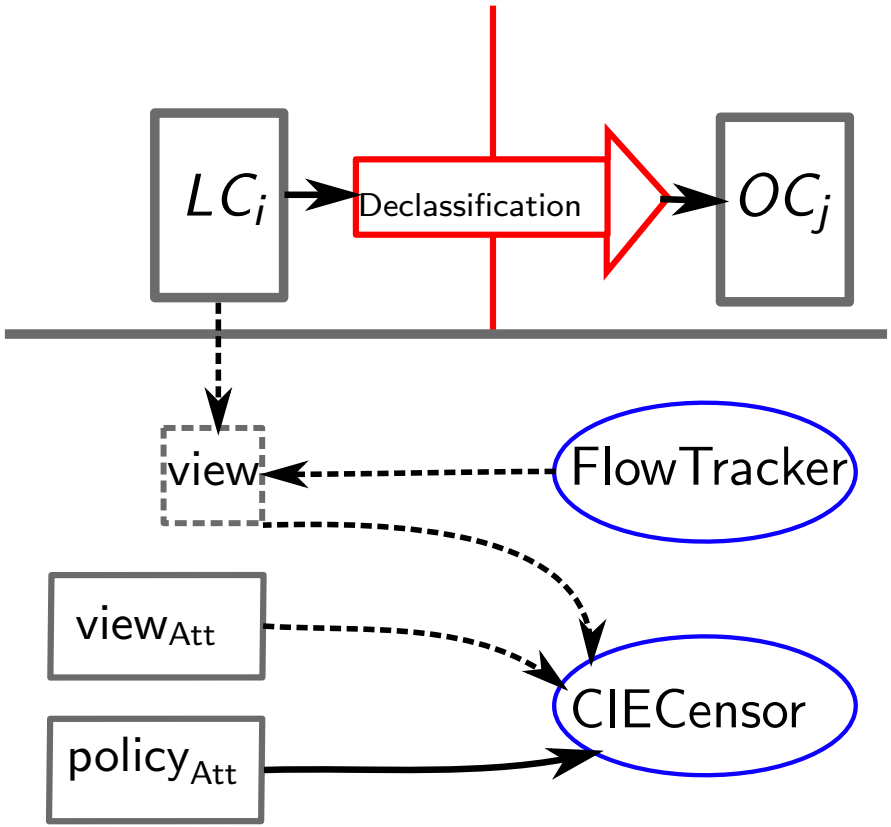
3. History-aware policy compliance





- Policy: sets $\mathcal{S} \subseteq \mathcal{IBS}$ of integrated inform. states
- History: previous view $\text{view}_{\text{Att}} \subseteq \mathcal{IBS}$
- Initial Policy Compliance: $\text{view}_{\text{Att}} \not\subseteq \mathcal{S}$ for all $\mathcal{S} \in \text{policy}_{\text{Att}}$

4.Need of Local Flow Tracking



- Policy: sets $\mathcal{S} \subseteq \mathcal{IBS}$ of integrated inform. states
- History: previous view $\text{view}_{\text{Att}} \subseteq \mathcal{IBS}$
- Initial Policy Compliance: $\text{view}_{\text{Att}} \not\subseteq \mathcal{S}$ for all $\mathcal{S} \in \text{policy}_{\text{Att}}$

5. Identifying implicit flows

Goal

Find alternative execution paths

to hide execution paths

originating from confidential pieces of integrated information

by making them indistinguishable from the alternatives

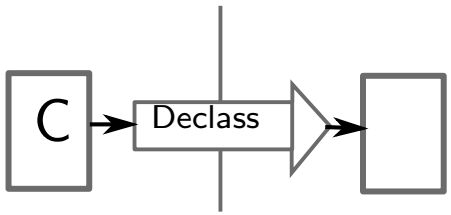
Approach

Identify and represent such candidates of alternative paths

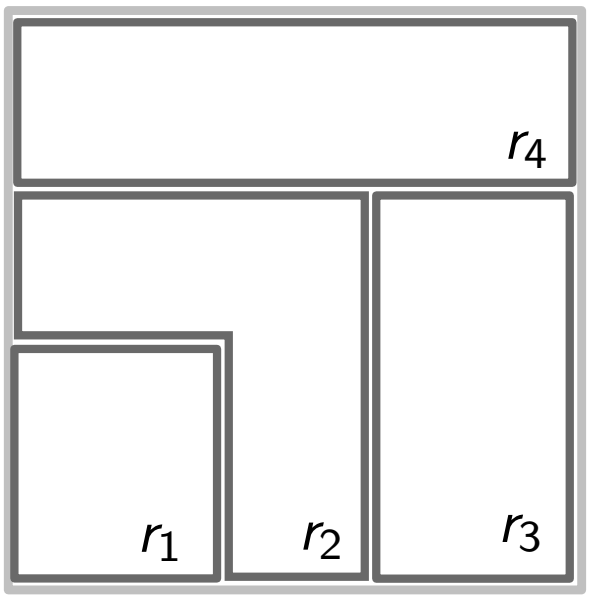
by symbolic execution of protected realm commands

during program execution

6. Determining local flows by FlowTracker

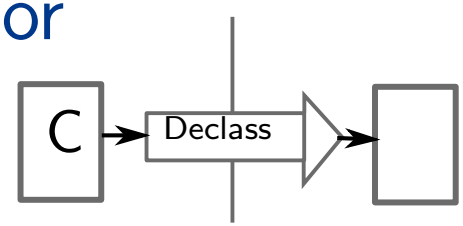


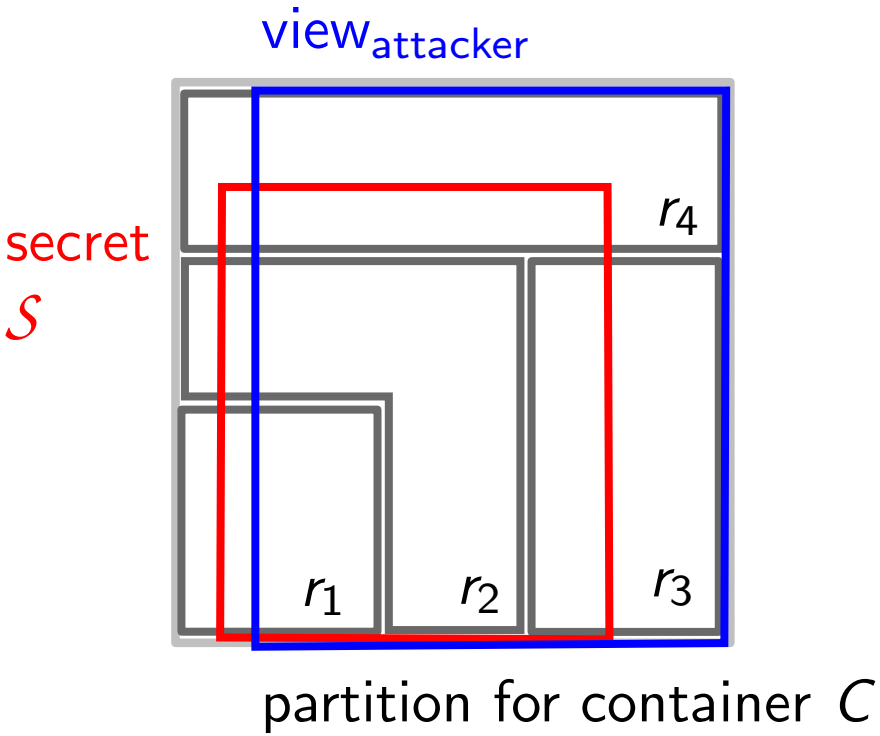
as partition of set of integrated inform. states



- represents indistinguishable execution paths originating from
  the respective inform. states
  leading to a value r_i of a container c
- initialized for basic inform. reactions
- refined by FlowTracker using
  a precomputed symbolic expression for c

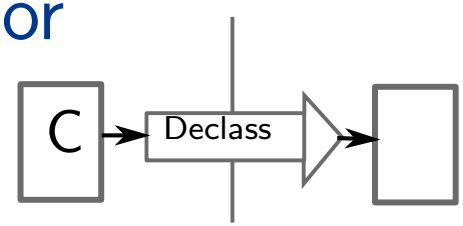
7. Evaluation of harmlessness by CIECensor

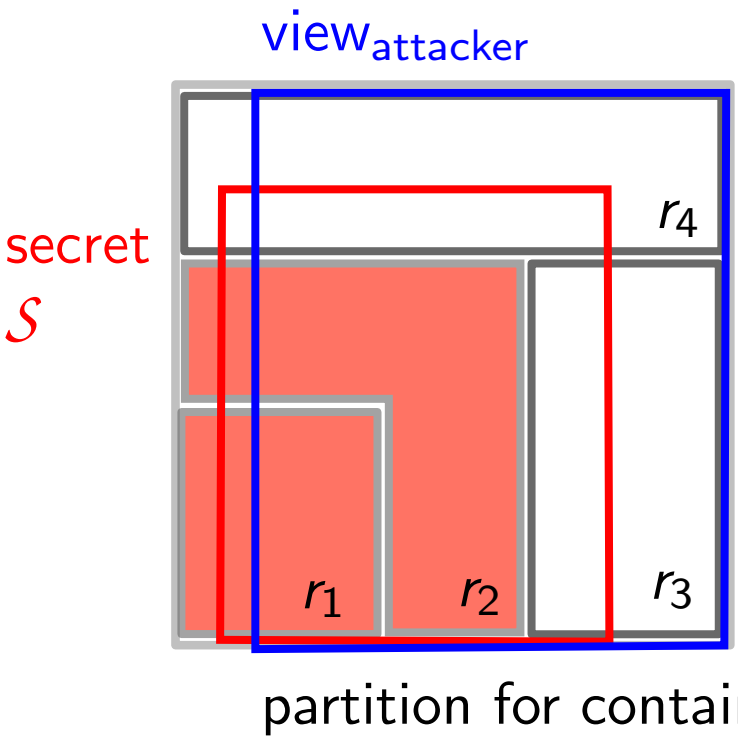




partition for container C

7. Evaluation of harmlessness by CIECensor



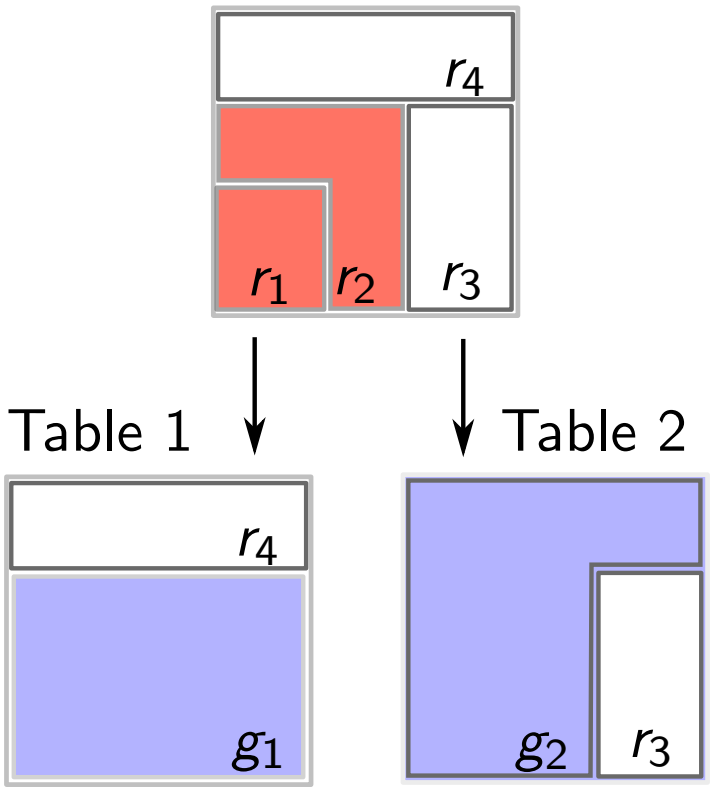


identifying

harmful unions of blocks

$(B_{r_1} \cup B_{r_2}) \cap \text{view}_{\text{attacker}} \subseteq \mathcal{S}$

8. Filtering and modifying by generalization



using a distortion table:

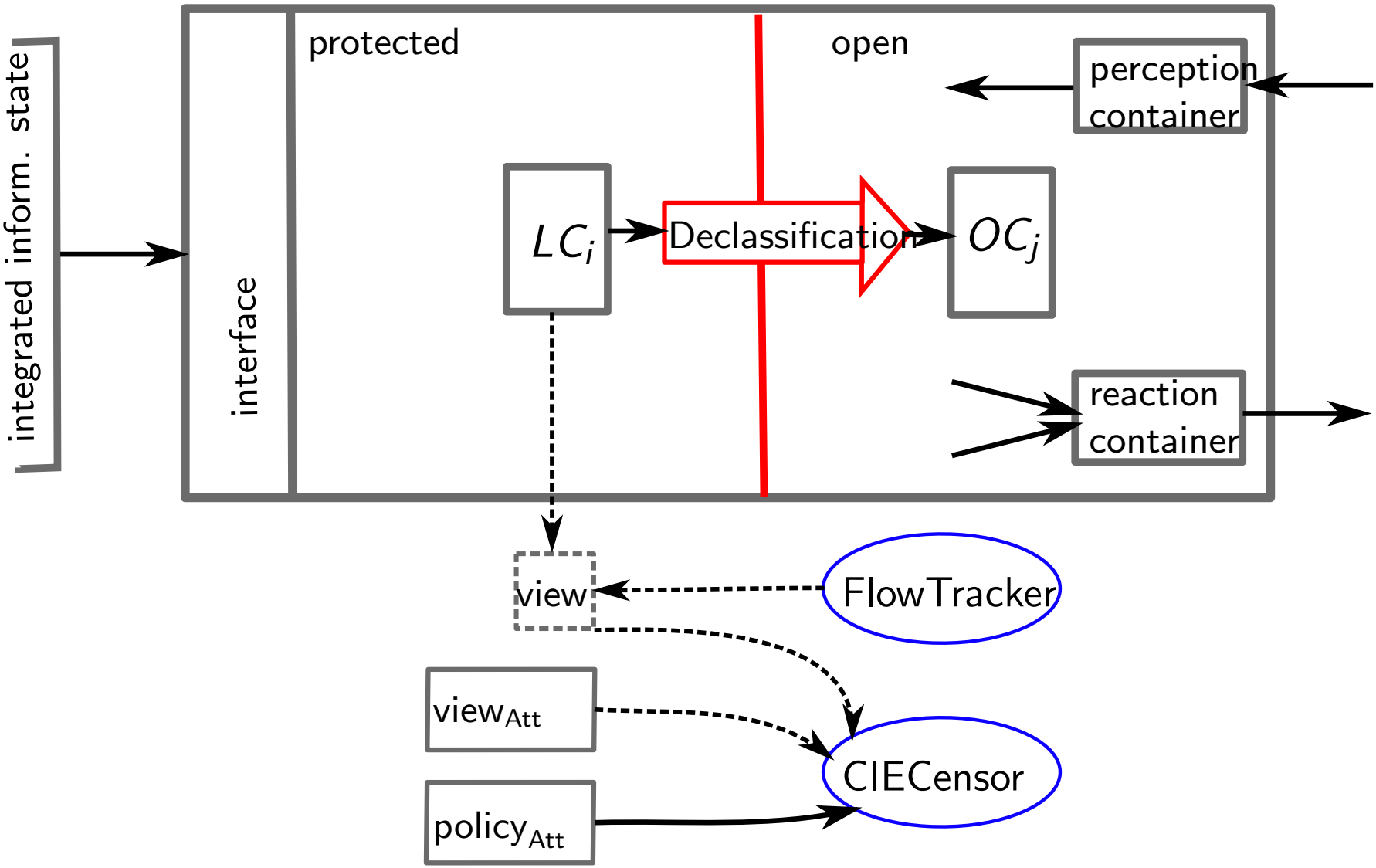
set of indices per harmful union of blocks

$\times$

value

$\rightarrow$ value/generalized value

The Complete Framework



Conclusion

Design, Formalization and Verification of a Mediator Framework

- for a holistic control based on a unified view on heterogenous sources by virtual integration

Design, Formalization and Verification of a Mediator Framework

- for a holistic control based on a unified view on heterogenous sources by virtual integration
- adhering to requirements in line with the agent paradigm

Design, Formalization and Verification of a Mediator Framework

- for a holistic control based on a unified view on heterogenous sources by virtual integration
- adhering to requirements in line with the agent paradigm
- with a integrated-inform.-program interface providing basic belief requests for a unified inference control based on

Design, Formalization and Verification of a Mediator Framework

- for a holistic control based on a unified view on heterogenous sources by virtual integration
- adhering to requirements in line with the agent paradigm
- with a integrated-inform.-program interface providing basic belief requests for a unified inference control based on
  - controlled interaction execution

Design, Formalization and Verification of a Mediator Framework

- for a holistic control based on a unified view on heterogenous sources by virtual integration
- adhering to requirements in line with the agent paradigm
- with a integrated-inform.-program interface providing basic belief requests for a unified inference control based on
  - controlled interaction execution
  - language-based information flow control

Design, Formalization and Verification of a Mediator Framework

- for a holistic control based on a unified view on heterogenous sources by virtual integration
- adhering to requirements in line with the agent paradigm
- with a integrated-inform.-program interface providing basic belief requests for a unified inference control based on
  - controlled interaction execution
  - language-based information flow control
- constructed using

Design, Formalization and Verification of a Mediator Framework

- for a holistic control based on a unified view on heterogenous sources by virtual integration
- adhering to requirements in line with the agent paradigm
- with a integrated-inform.-program interface providing basic belief requests for a unified inference control based on
  - controlled interaction execution
  - language-based information flow control
- constructed using
  - security typing,

Design, Formalization and Verification of a Mediator Framework

- for a holistic control based on a unified view on heterogenous sources by virtual integration
- adhering to requirements in line with the agent paradigm
- with a integrated-inform.-program interface providing basic belief requests for a unified inference control based on
  - controlled interaction execution
  - language-based information flow control
- constructed using
  - security typing,
  - declassification,

Design, Formalization and Verification of a Mediator Framework

- for a holistic control based on a unified view on heterogenous sources by virtual integration
- adhering to requirements in line with the agent paradigm
- with a integrated-inform.-program interface providing basic belief requests for a unified inference control based on
  - controlled interaction execution
  - language-based information flow control
- constructed using
  - security typing,
  - declassification,
  - flow tracking inspired by symbolic execution

Design, Formalization and Verification of a Mediator Framework

- for a holistic control based on a unified view on heterogenous sources by virtual integration
- adhering to requirements in line with the agent paradigm
- with a integrated-inform.-program interface providing basic belief requests for a unified inference control based on
  - controlled interaction execution
  - language-based information flow control
- constructed using
  - security typing,
  - declassification,
  - flow tracking inspired by symbolic execution
  - and generalization for policy compliance