Towards Forensic Data Flow Analysis of Business Process Logs

Rafael Accorsi, Claus Wonnemann, Thomas Stocker

University of Freiburg, Germany

accorsi@iig.uni-freiburg.de

IMF, Stuttgart 2011

Outline

1. ***BPSec*** group
2. Some problems for enterprise forensics
3. The **RecIF** approach and security model
4. Summary

BPSec Group

- „Business Process Security“
  - Focus: Security / Compliance
  - BMBF- und DFG-Projects
  - Four PhD candidates

- Approaches for
  - **Certification**
  - **Auditing**
  - **Simulation**

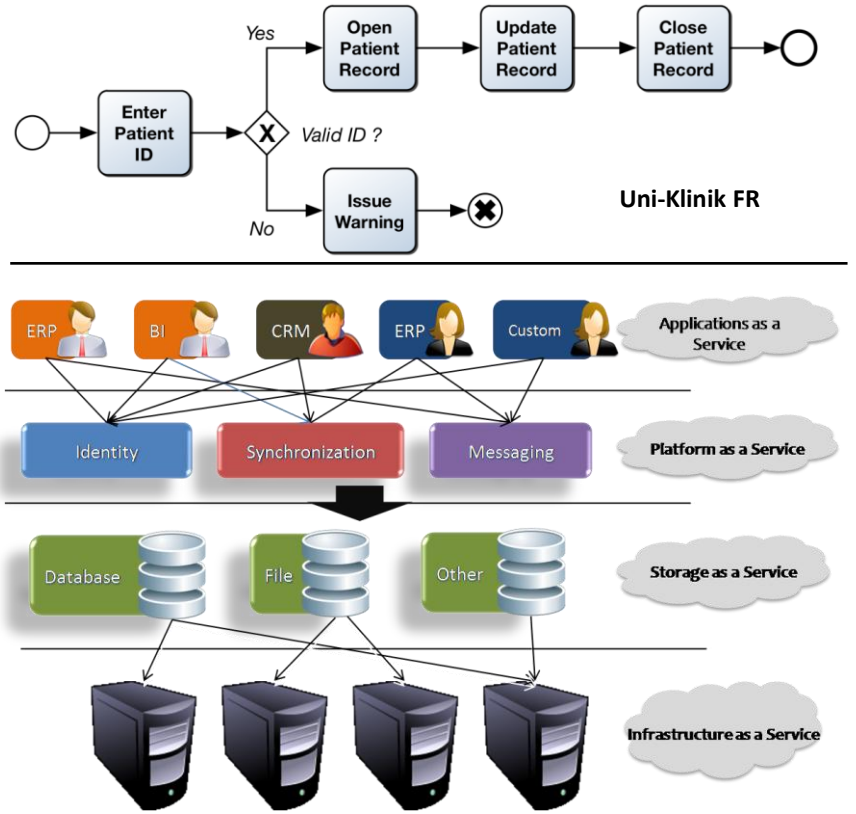
  of business processes and corresponding tool support

- Web: http://www.telematik.uni-freiburg.de/bpsec

  We are hiring!

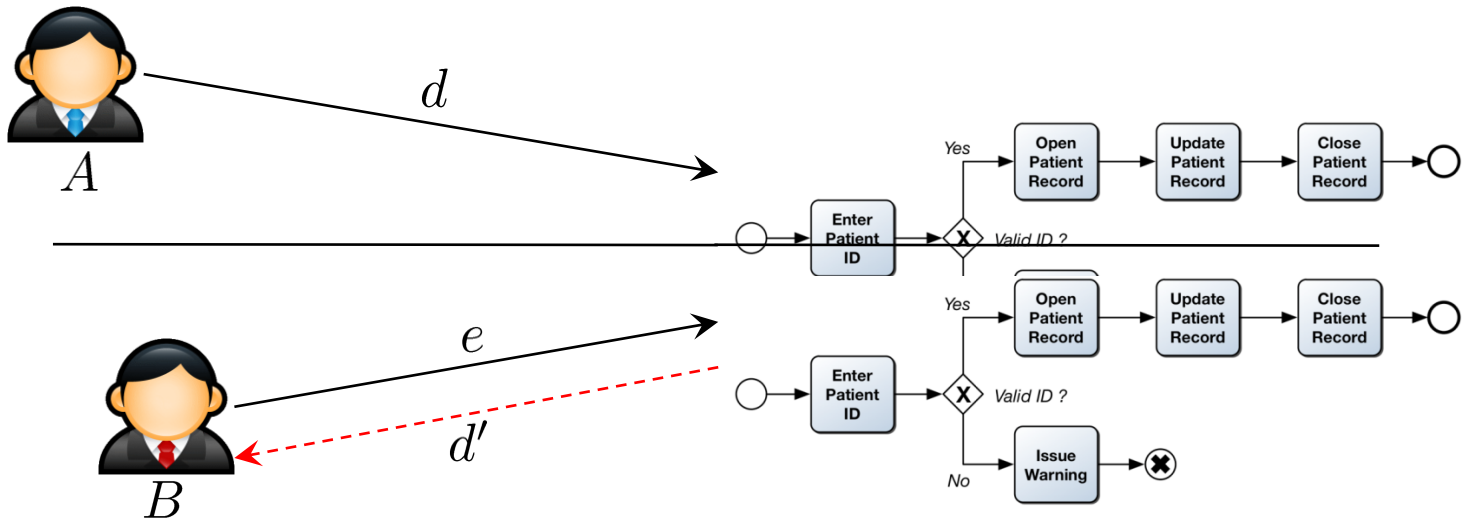
Process-aware Information Systems

- Business processes
  - Patterns for enterprise procedures (in IT-Systems → Workflows)
  - Specification in BPEL, BPMN, EPC, etc.



- **PAIS**
  - Software-layer for the management and execution of processes
  - Intra and cross-enterprise
  - Multi-tenancy
  - Dynamic and configurable

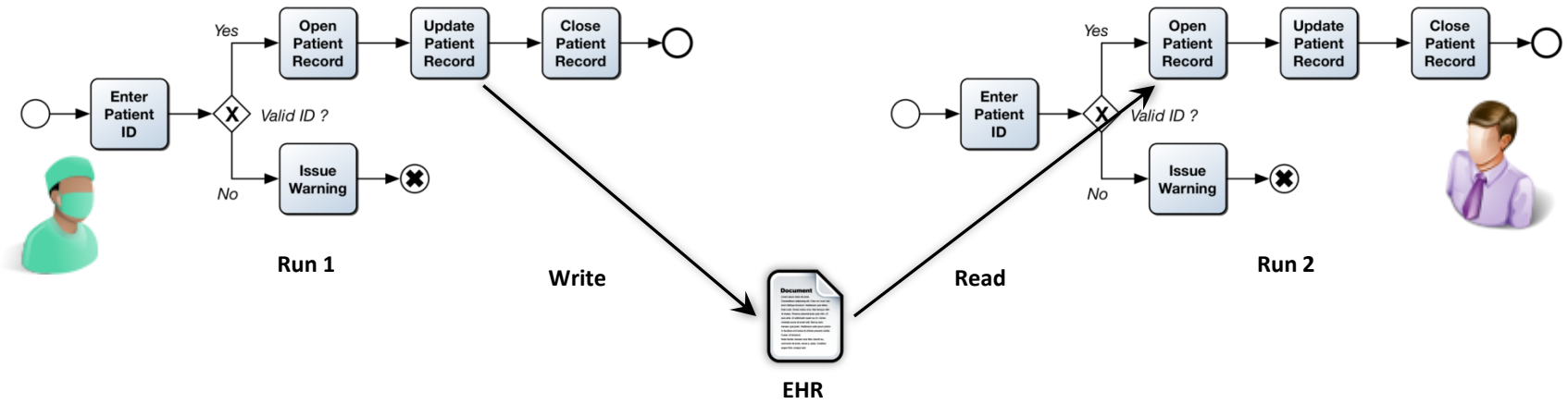
Security Requirements for Processes



- SLA demand isolation:
  - **Multi-party** : data flows only to authorized parties
  - **Multi-tenant** : A process instance does not influence another

- Further requirements
  - Sepation of duties, 4-eye principle, Chinese wall, etc

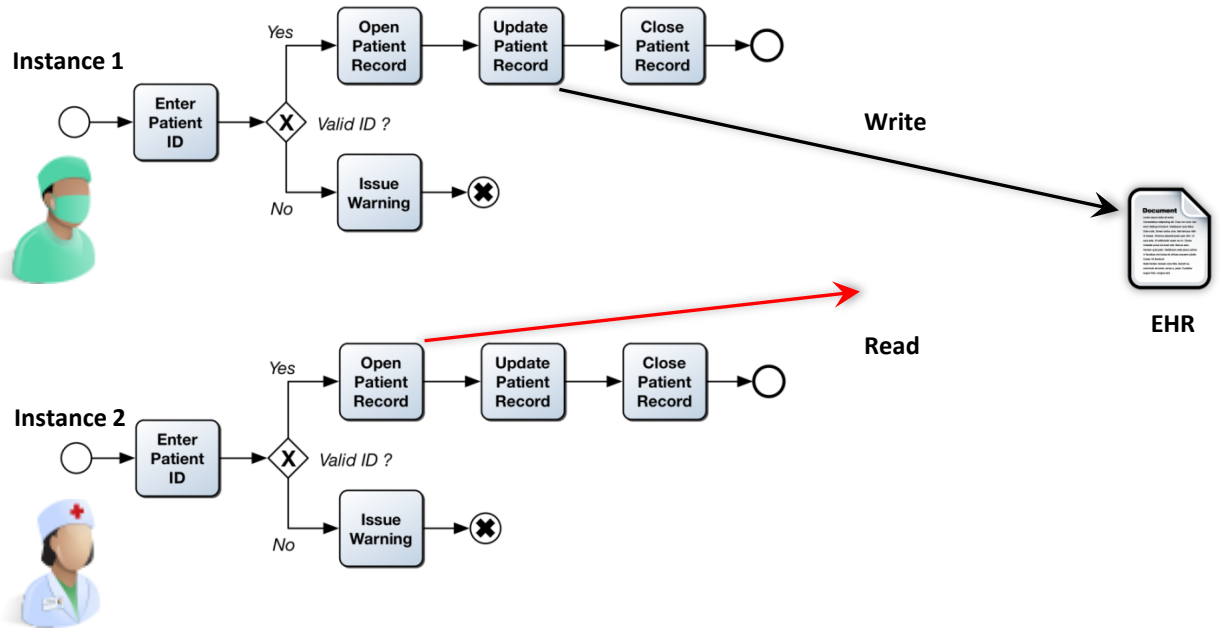
Problem 1: Chained Accesses



- Consequence of chained accesses: illegitimate data flows
  - Each access is legitimate
  - Their combination leads to a violation of the policy

- Administrative role/users can look at data
  - Security controls fail

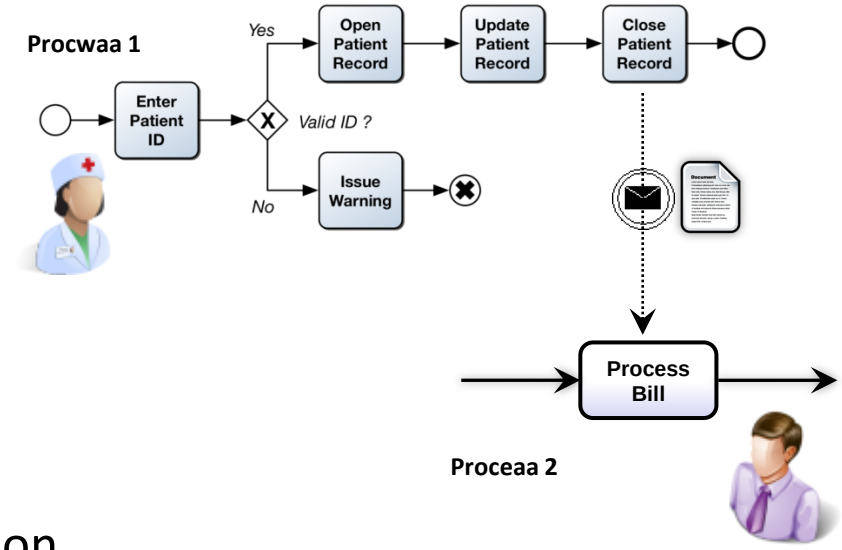
Problem 2: Concurrent Instances



- Consequence of concurrent instances
  - Instance 2 is deadlocked → Covert channel
  - **Information flow** between subjects

- Instance 2 can deduce information
  - Timing, execution parameter, subject identity , etc.

Problem 3: Causality



- Consequence of causal activities
  - Process 1 depends of Process 1
  - **Information flow** between processes
- Subject in Process 2 can derive information

- Are these problems forensically relevant?
  - TCSEC 70, SAS 70, ISO 17799, ISACA, usw.

- Is that so complicated to analyze these processes?

Typical Log File

File Edit Format View Help

```
#Software: BizAgi Web
#Version: 9.1.4.1002
#Platform: CLR 2.0.50727.4952
#OSVersion: Microsoft Windows NT 6.1.7600.0
#Machine: DEV-ROBBEN
#ProcessorCount: 4
#User Name: Classic .NET AppPool
#Domain: IIS APPPOOL
```

1

```
#Fields: date time session level module submodule message
```

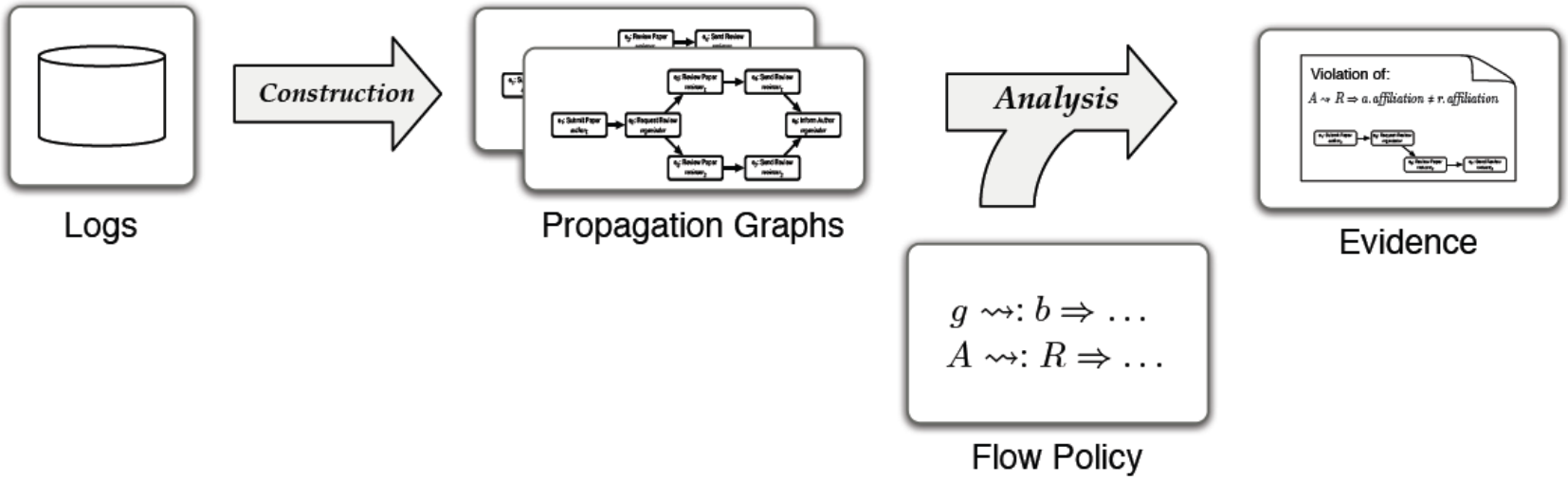
2

```
2010-11-23 16:15:01.217 58498953 INFO WORKFLOW---------- ---------- BEGIN: Create Process Instance
2010-11-23 16:15:01.338 58498953 INFO WORKFLOW---------- ---------- Begin transaction
2010-11-23 16:15:01.343 58498953 INFO WORKFLOW---------- ---------- Get Process Definition
2010-11-23 16:15:02.281 58498953 INFO WORKFLOW---------- ---------- Create Process: 551
2010-11-23 16:15:02.323 58498953 INFO WORKFLOW---------- ---------- Set case scope checkPoint
2010-11-23 16:15:02.664 58498953 INFO WORKFLOW---------- ---------- BEGIN: Executing task id=61 Name=St
2010-11-23 16:15:02.716 58498953 INFO WORKFLOW---------- ----------  Executing transition id=73 Name= Di
2010-11-23 16:15:02.727 58498953 INFO WORKFLOW---------- ----------  BEGIN: Executing task id=61 Name=S
2010-11-23 16:15:02.733 58498953 INFO WORKFLOW---------- ----------  END
2010-11-23 16:15:40.334 58498953 INFO WORKFLOW---------- ASSIGNMENT----------  possibleAssigneesIds id="1"

2010-11-23 16:15:40.342 58498953 INFO WORKFLOW---------- ASSIGNMENT----------  AllAssigneesIds id="1"

2010-11-23 16:15:40.576 58498953 INFO WORKFLOW---------- ---------- END
2010-11-23 16:15:40.585 58498953 INFO WORKFLOW---------- ---------- Commit data
2010-11-23 16:15:40.791 58498953 INFO WORKFLOW---------- ---------- Commit transaction
2010-11-23 16:15:40.798 58498953 INFO WORKFLOW---------- ---------- END
2010-11-23 16:15:40.828 58540828 INFO WORKFLOW---------- ---------- BEGIN: Create Process Instance
2010-11-23 16:15:40.838 58540828 INFO WORKFLOW---------- ---------- Begin transaction
2010-11-23 16:15:40.842 58540828 INFO WORKFLOW---------- ---------- Get Process Definition
2010-11-23 16:15:40.849 58540828 INFO WORKFLOW---------- ---------- Create Process: 552
2010-11-23 16:15:40.919 58540828 INFO WORKFLOW---------- ---------- Set case scope checkPoint
2010-11-23 16:15:40.940 58540828 INFO WORKFLOW---------- ---------- BEGIN: Executing task id=61 Name=St
2010-11-23 16:15:40.966 58540828 INFO WORKFLOW---------- ----------  Executing transition id=73 Name= Di
2010-11-23 16:15:40.970 58540828 INFO WORKFLOW---------- ----------  BEGIN: Executing task id=61 Name=S
2010-11-23 16:15:40.976 58540828 INFO WORKFLOW---------- ----------  END
```

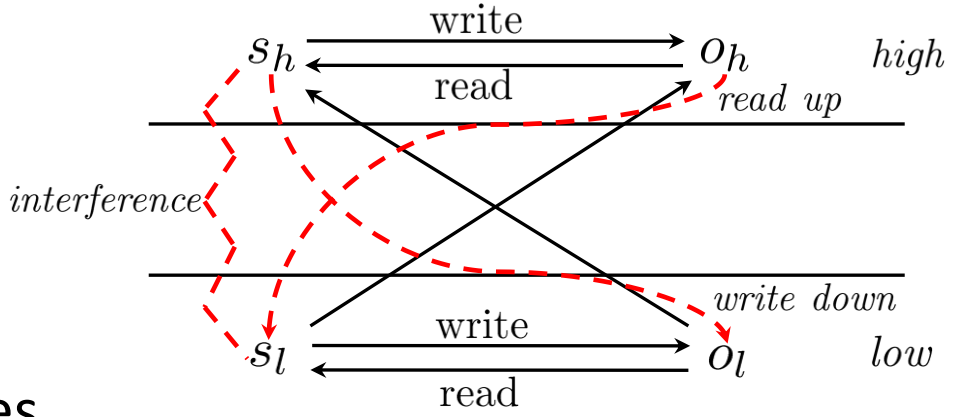
RecIF: Reconstructing Information Flows



- Reconstruction and analysis of data flows
  - Tackling Problem 1
  - Problem 2-3 require more expressive formalisms
- Propagation graphs: flow of data within an execution
- Use of flow policies and corresponding analysis

Multi Level Security Model (Denning 1976)

- System seen as security classes
  - *high*: confidential, *low*: public
- Capture both
  - Data flow
  - Information flow

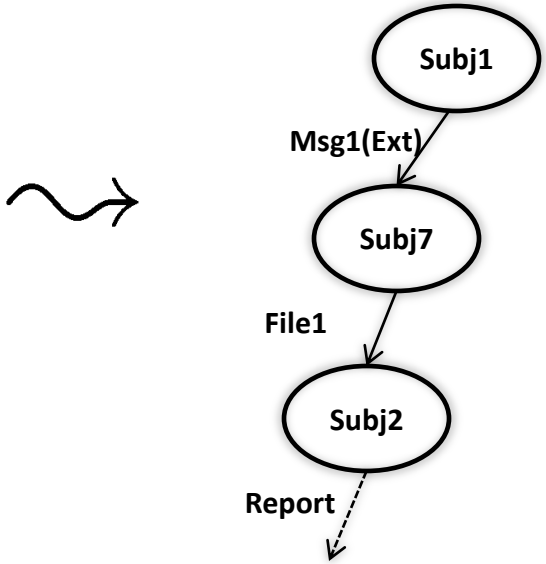


- Formalization of general policies
  - Description focuses on the relationship between classes
  - Not on the particular access rights and system specific aspects
  - Extensional and intensional specifications
- For RecIF: easier for investigators to formulate search criteria

Propagation graphs

Excerpt of a wf-log

Inst.ID	TStamp	Activity ID	Orig.	Input	Output
2	2010-4-23	Retr_Data	Subj1	Msg1(Ext)	File1
2	2010-4-23	Create_Rep	Subj7	File1	Report
2	2010-4-23	Publ_Rep	Subj2	Report	Web_Page

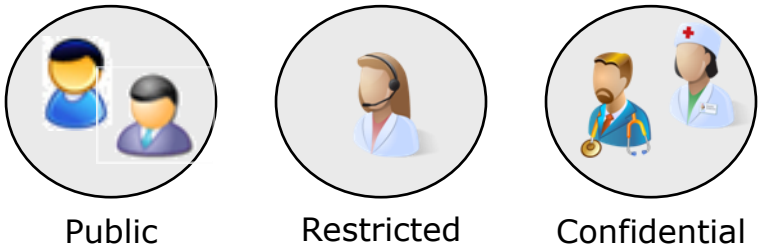


- Directed, labeled graph depicting the flow of data-items in a workflow execution
  - Nodes *V* denote subjects and the edges *E* denote flows
  - $PG = (V,E)$ *s. t.* $V = \{s \in S \mid S \in i^{W}_{i}\}$ and $E = \{(a,b) \in (A \times A) \mid a < b \wedge a.output \cap b.input \neq \{\}\}$
- Construction based upon normalized log files
- Each execution generates a PG
  - Redundant PGs are not added to the set of models

Dataflow policies

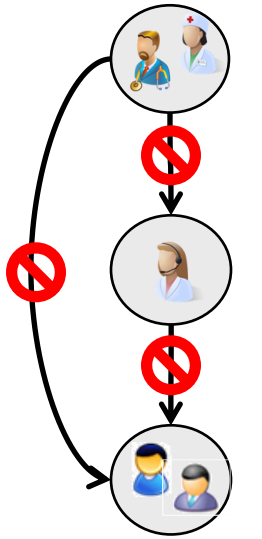
- The policy extensionally specifies:
  - The assignment of subjects and security classes
  - The allowed and forbidden dataflows

Exemplary security levels

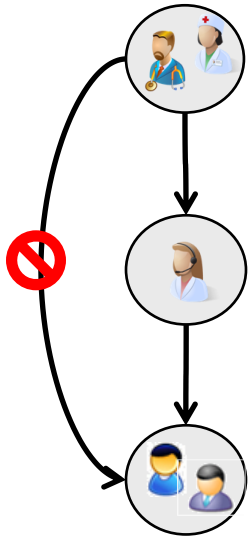


- Syntax $P = \{r_1, \ldots, r_n\}$:
  - r_i *Restriction* $\Rightarrow$ *Exception*
  - Restriction: flow relation source $\rightsquigarrow$ target
  - Exception: flows that contradict Restriction

- Trace-based semantics.
  - There is a dataflow from level L_1 to L_2 iff there is a data item modified in L_1 and subsequently read by L_2
  - Default-deny for non-specified settings



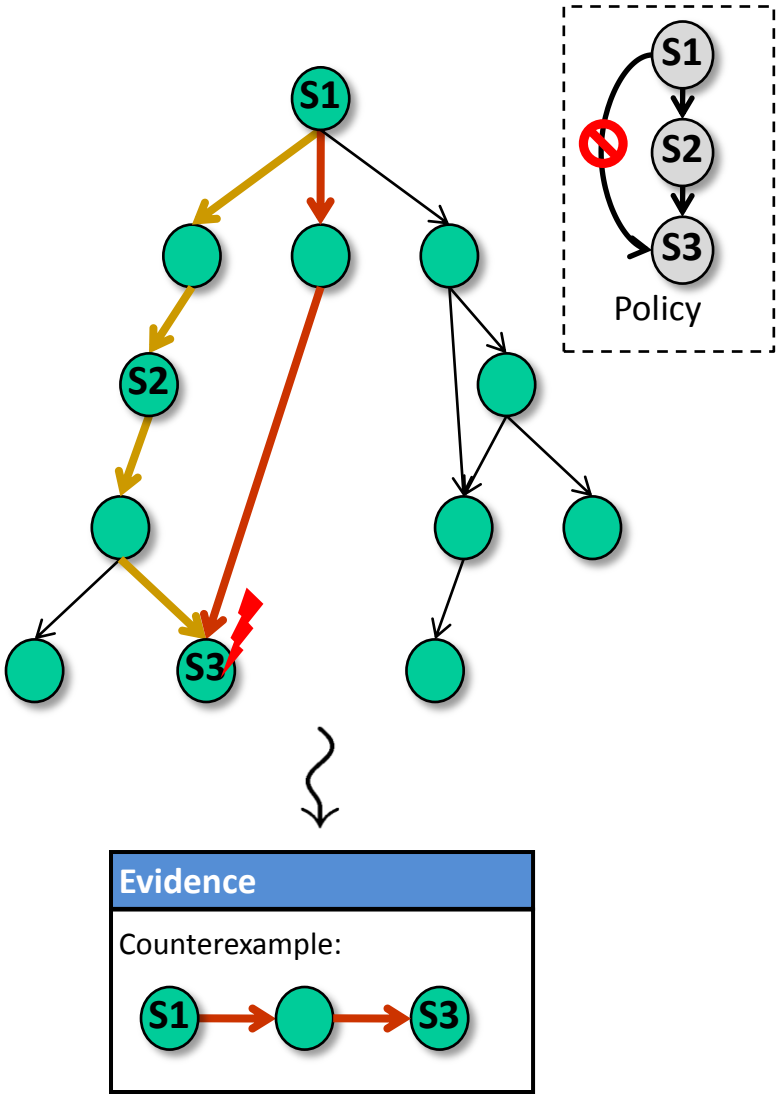
Strict hierarchies (e.g. Bell-LaPadula and Chinese-Wall)



Declassification (intransitive)

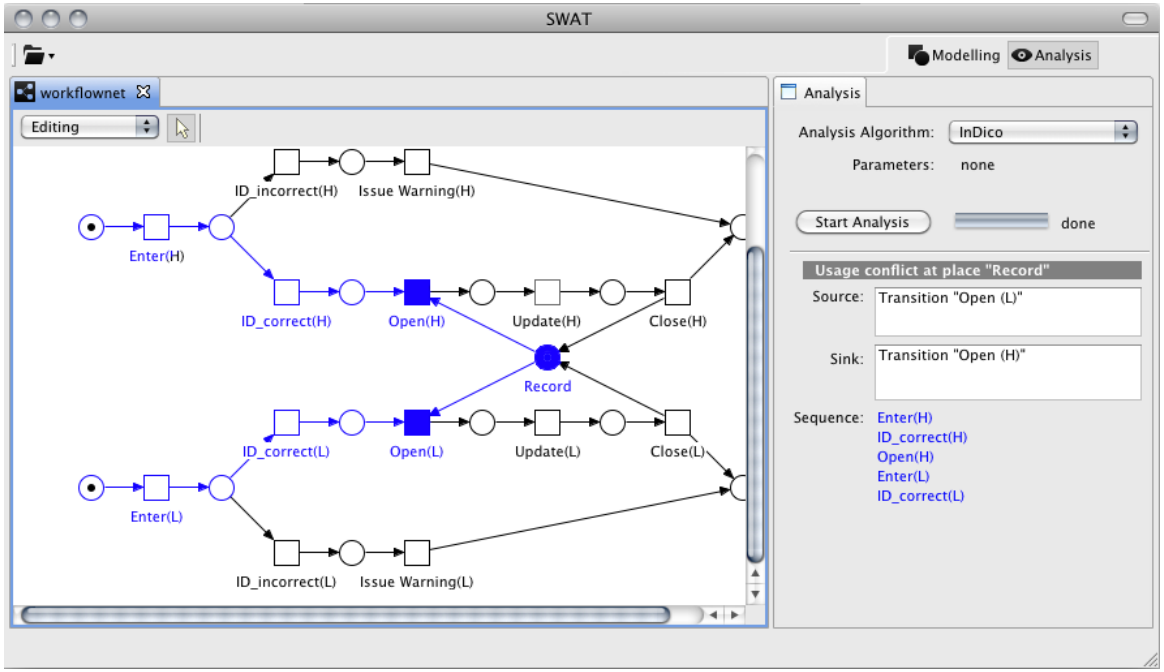
Evidence generation

- Compliance with policies reduced to a graph search problem
  - Analysis as depth-first search of PG against policies
  - Detects every dataflow violation
- Elimination of redundant graphs leads to performance optimizations
  - No loss of relevant traces
- Current limitations:
  - Excessive number of false positives
  - Bugs in reflexive/cyclic PG



Evaluation w/ SWAT: Security Workflow Analysis Toolkit

- Tool for workflow:
  - Modeling
  - Simulation
  - Security analyses



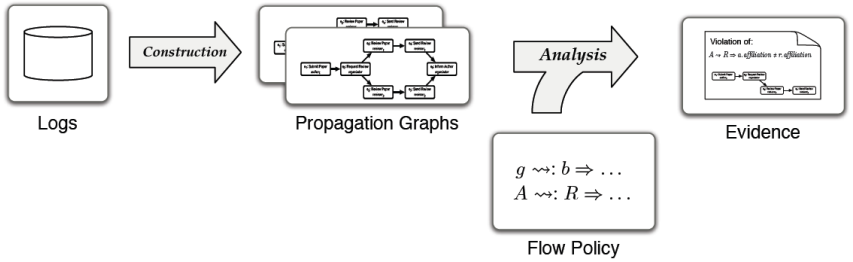
- IF-Audit tests:
  - Process w/ 15 activities
  - Log size 75K traces
  - Redundancy: 31%
  - Elapsed time: < 3 min

- Ongoing activities:
  - How expressive is the policy language?
  - How to derive them from the extensional policies?
  - Separation of duties, four-eye principle and delegation.
  - Further case studies.

Summary

- RecIF: Forensic data flow analysis of business process logs
  - Propagation graphs
  - Extensional policies

  Need for similar tools might grow!



- Evaluation:
  - Synthesized log files are not bee sufficient
  - Any candidates?

- Issues
  - Consideration of data flows only
  - Generality of policies is good, sometimes too good.
  - Propagation graphs: too simple?