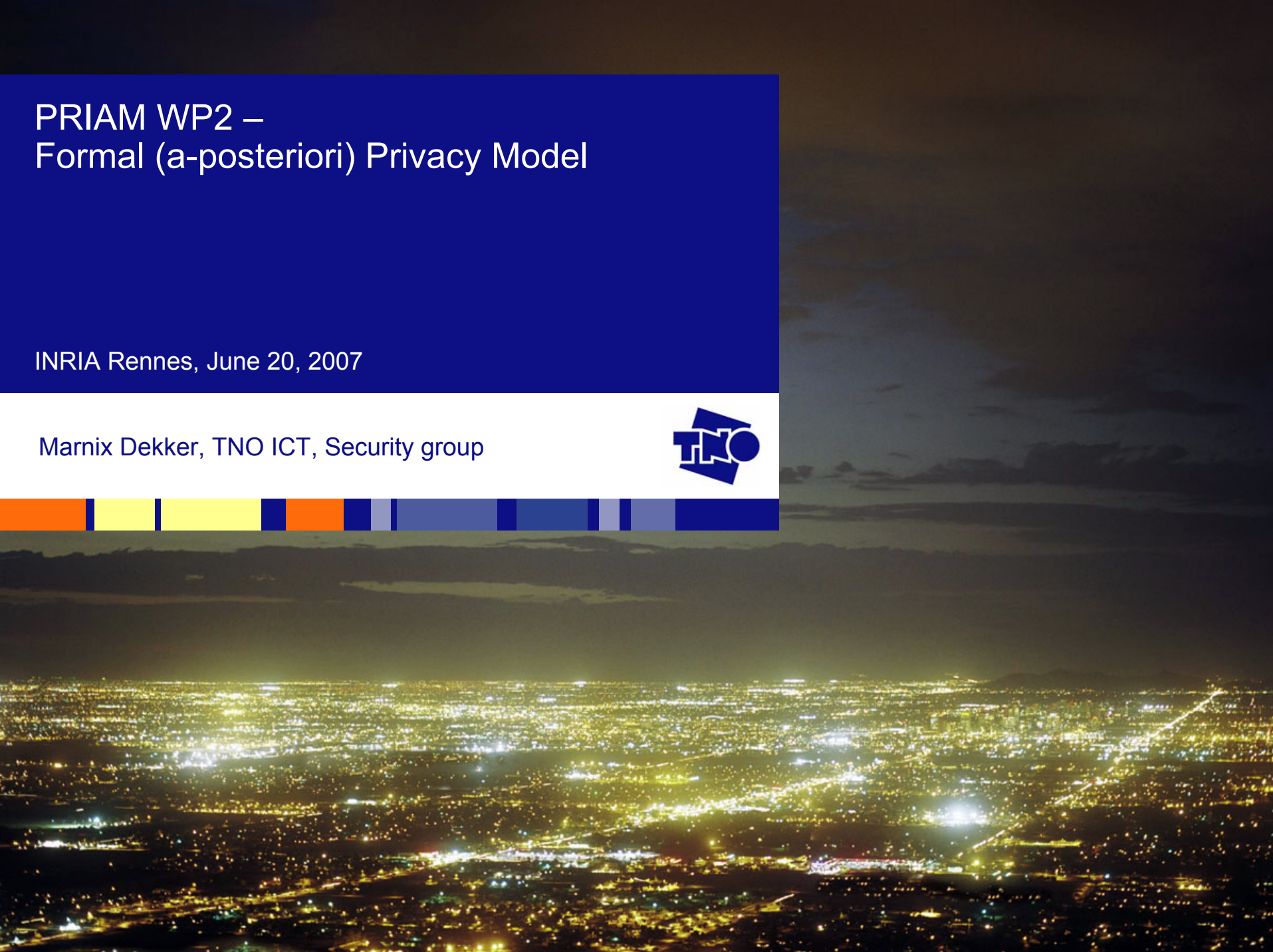
PRIAM WP2 – Formal (a-posteriori) Privacy Model

INRIA Rennes, June 20, 2007

Marnix Dekker, TNO ICT, Security group



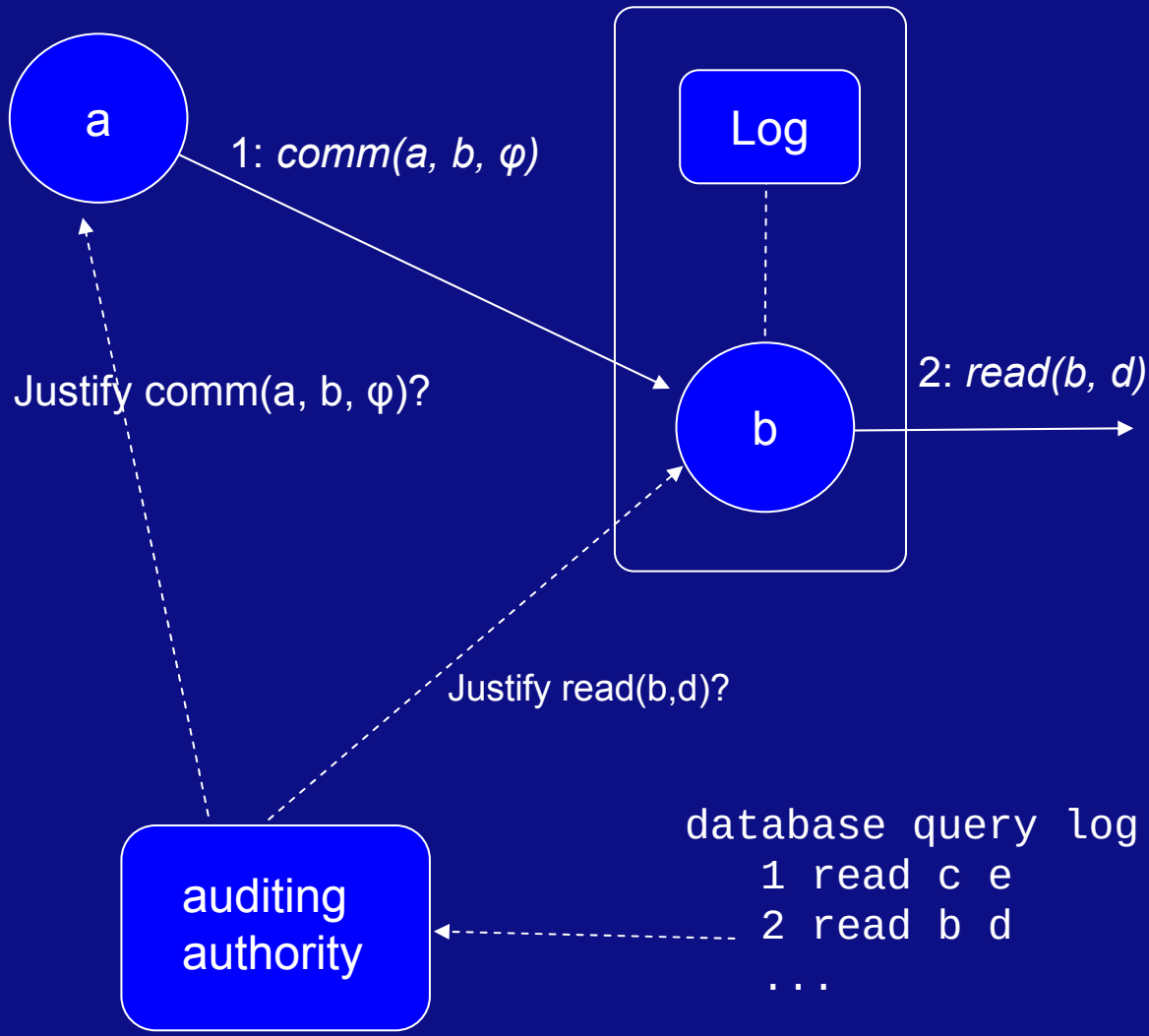
Talk outline

- Past work AC^2
  published in IEEE Policy 2005, VODCA 2006, IJIS 2007
  - Sketch
  - Sample use case
  - Technical details
  - Disadvantages wrt ambient scenario
- New framework for privacy in ambient technologies, hybrid (a-posteriori and a-priori) system, with a (standard) policy language and a formal proof system. To check system compliance, to allow safe user-overrides.

AC² Features and assumptions

- Decentralized architecture (e.g. no explicit prohibitions).
- Network of *peers*, trust relations are established.
- Discretionary: users can create data (then they *own* these data).
- Users can write and exchange policies regarding data.
- Agents can *misbehave*, compliance is checked a-posteriori.
- Users can securely log events, and context to help *justify* actions.
- Users can provide a (formal) *justification proof* to the authority, when audited.
- An *authority* (coalition or super-agent) can collect hard evidence.
- Users can be held *accountable* (contracts, bailsum).

AC² Sketch


a
1: comm(a, b, φ)
Log
b
2: read(b, d)
Justify comm(a, b, φ)?
Justify read(b,d)?
auditing authority
database query log
1 read c e
2 read b d
...

AC^2 Application scenarios

- Many peers, dynamic data usage, detailed policies
  - Collaborative work environments (SOX)
  - Electronic Health Record systems (HIPAA)
- Confidentiality is required, but unavailability is expensive (also for the data owner).
- Security measures must be verifiable, auditable, as the data is processed on another user's computer.

MEDICAL RECORDS:

From Clipboard To Point-and-Click

BY TRUDY E. BELL

CALL THEM ELECTRONIC charts or electronic medical records; whatever the name, the days of patients' medical conditions and diagnoses being written illegibly on paper and stored in manila folders are numbered. Medical records, according to plans under way, are going electronic.

To help make that happen, the IEEE has joined forces with the American Medical Association and eight other major nonprofit medical and engineering societies to form an umbrella consortium, the Biotechnology Council. The council's primary goal is nothing less than standardizing everything from medical terminology to networking protocols so that medical records can be stored electronically and sent instantly anywhere in the world—with absolute privacy, security, and understandability.

In a few months, the first fruits of the Biotechnology Council's efforts—the council passed its first anniversary in November—will ripen. Its first technical conference, the Distributed Diagnosis and Home Health Care Conference on remote-monitoring technologies and policies, is scheduled for 3 and 4 April in Washington, D.C. The council also plans to hold a workshop on what it terms Bio-

[Continued on page 17]

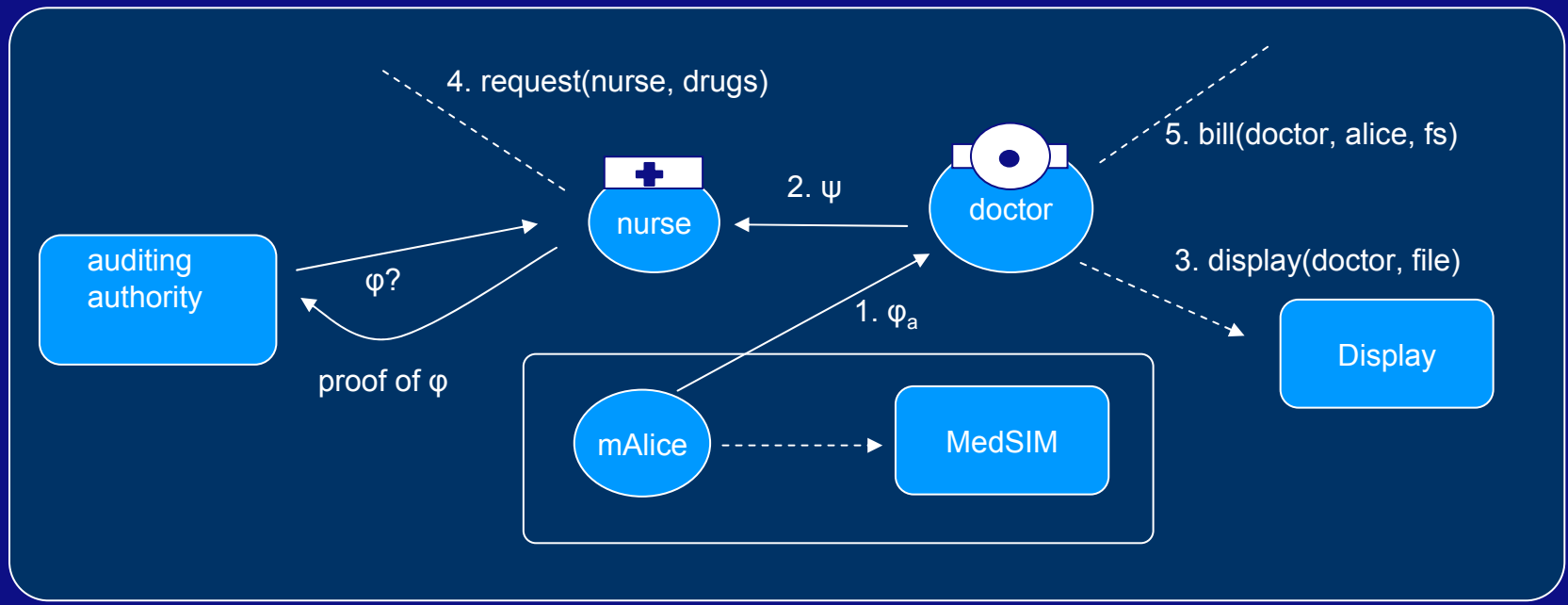
The days are numbered for storing medical records in paper folders, thanks partly to the IEEE's work on e-medical systems.

Example: Electronic Health Records (EHR)

- In 2002 a Privacy rule was attached to HIPAA (Health Insurance Portability an Accountability Act):
  - Only disclosure if permitted by HIPAA or if patient authorizes in writing.
  - Patients have the right to an accounting of disclosure of the past 6 years.
  - "...does not require that every risk of incidental disclosure be eliminated."
- Unavailability is expensive (doctor's wage, double exams, bad diagnoses, etc).
- Unexpected situations are common: Patients, and doctors are mobile. And medical care is often urgent.
- Decision procedure may be slow, since medical policies are complex, and the patient group is of order 10^6 (much larger than in e-commerce e.g.).

Advantages of a-posteriori control

- Audit trails are anyway required (HIPAA's *disclosure-accounting)*
- Doctors must already justify their actions, a-posteriori.
- Medical staff can continue their duties, leaving administrative details (such as obtaining authorizations, bills, certificates) for later.



Policy Language (Φ)

- We need some language to express facts, actions, and permissions (independent of the enforcement type).
  - SecPAL (2007, Microsoft Research)
  - ABLP (Abadi et al.)
  - Binder (DeTreville)
  - PCA (Appel & Felten, Bauer et al.)
- P3P, XRML, SPKI, XACML ... (ambiguous, unclear)
- Advantages of logic based languages
  - logic syntax ‘reads’ as natural language
  - unambiguous
  - less verbose
  - formal tools, techniques from logics (queries, consistency, decidability, ...)

AC² policy language

Users: a, b, ...
Data: d
Permissions and conditions: $p(a, d)$

$$\phi ::= p \mid owns(a, d) \mid \phi \rightarrow \phi \mid \phi \wedge \phi \mid \forall\phi \mid$$
$$says(a, b, \phi) \mid maySay(a, b, \phi) \mid$$
$$!\alpha \rightarrow \phi \mid ?\alpha \rightarrow \phi;$$

For example:
alice $maySay$ (bob $mayRead$ file) to bob.

AC² Semantics

$$\frac{\Gamma \vdash_a says(b, a, \phi) \qquad ?}{\Gamma \vdash_a \phi}\text{says\_e}$$

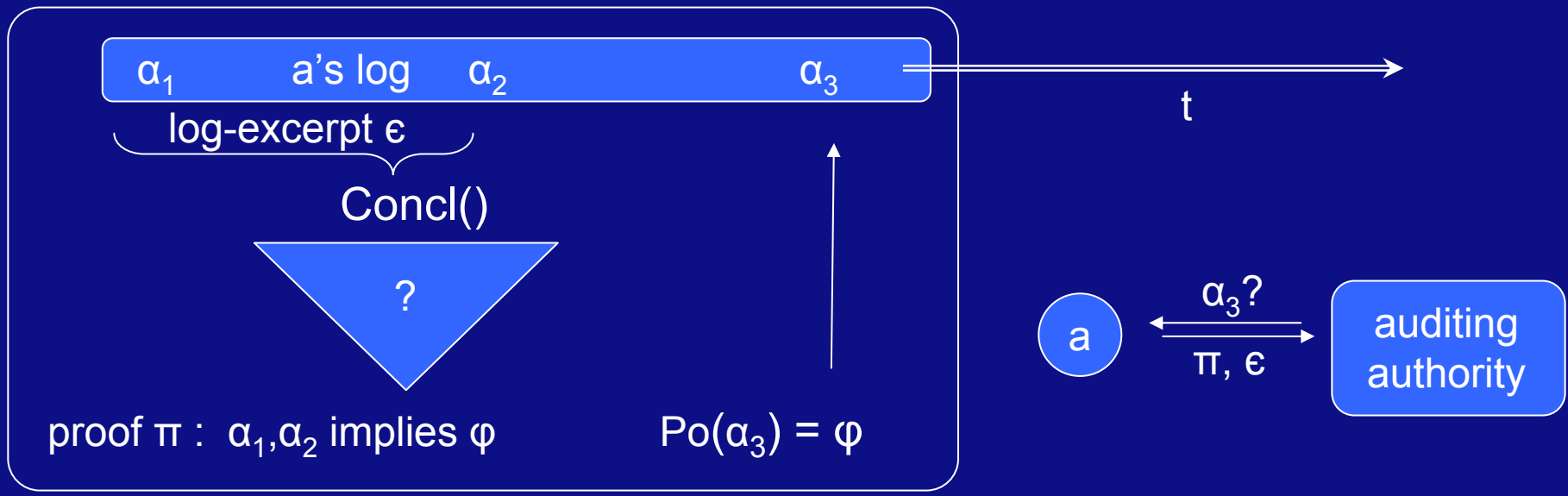
hand-off axiom (ABLP)
controls predicate (PCA)
trust management

$$\frac{\Gamma \vdash_a owns(a, d)}{\phi[d]}\text{owns\_e}$$

$$\frac{\vdash \phi \rightarrow \psi \qquad \Gamma \vdash_a maySay(b, c, \phi)}{\Gamma \vdash_a maySay(b, c, \psi)}\text{refine}$$

+ the introduction and elimination rules for *conjunction, implication*, *quantification*.

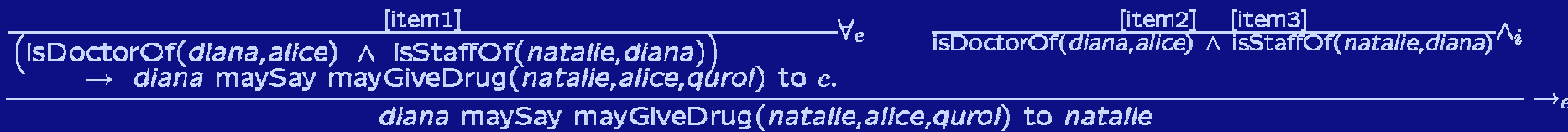
AC² From log and policy to justification



Conclusion (concl) function and *Proof-obligation (Po)* function are global and public. For example, *Po(read) = mayRead, concl(comm) = says, Po(comm) = maySay*

AC² Tools

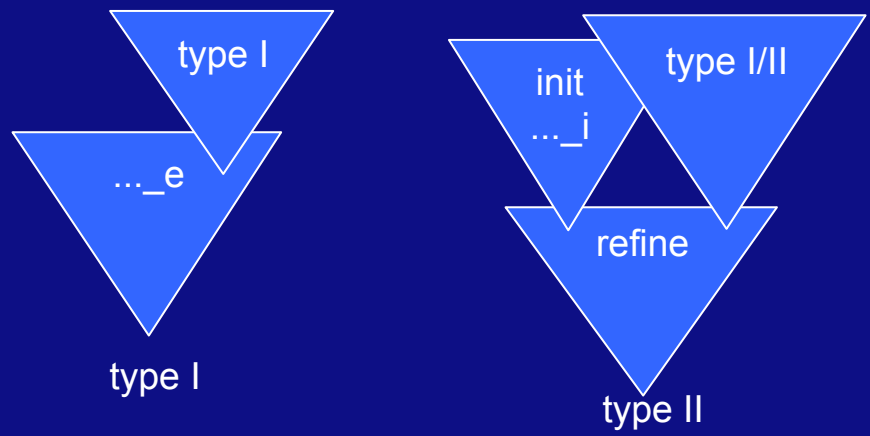
- Checking proofs is decidable (e.g. via Curry-Howard), easy like program-language type-checking.
- Twelf – A Type-theory proof-checker (front-end for the LF framework, types are inferred when possible).
- Compact proof-checker, verifiable by hand (15 lines of code).
- ASCII syntax for expressing, and remote checking of proofs.




```
thm:  (entail diana
   (cons item3 (cons item2 (cons item1)))
     (says diana (mayGiveDrug natalie alice qurol) natalie))=
(forall_1 (forall_1 (forall_1 (forall_1 (imp_1 (and_r init (perm_2 init))(refine init (map_cons map_nil)))))))).
```

Intermezzo: proof-system

- *maySay* distributes over connectives
- *owns* behaves like false
- alternatives (i.e. Li's Delegation Logic) are often based on Datalog and don't have *maySay*.
- Is our logic consistent/tractable/implementable? (proof-search)
- No normal forms:
  - Type I proofs lead to atomic predicates = a permission for an action.
  - Type II proofs lead to the maySay predicate = a permission to communicate a compound predicate.



AC² Proof finding

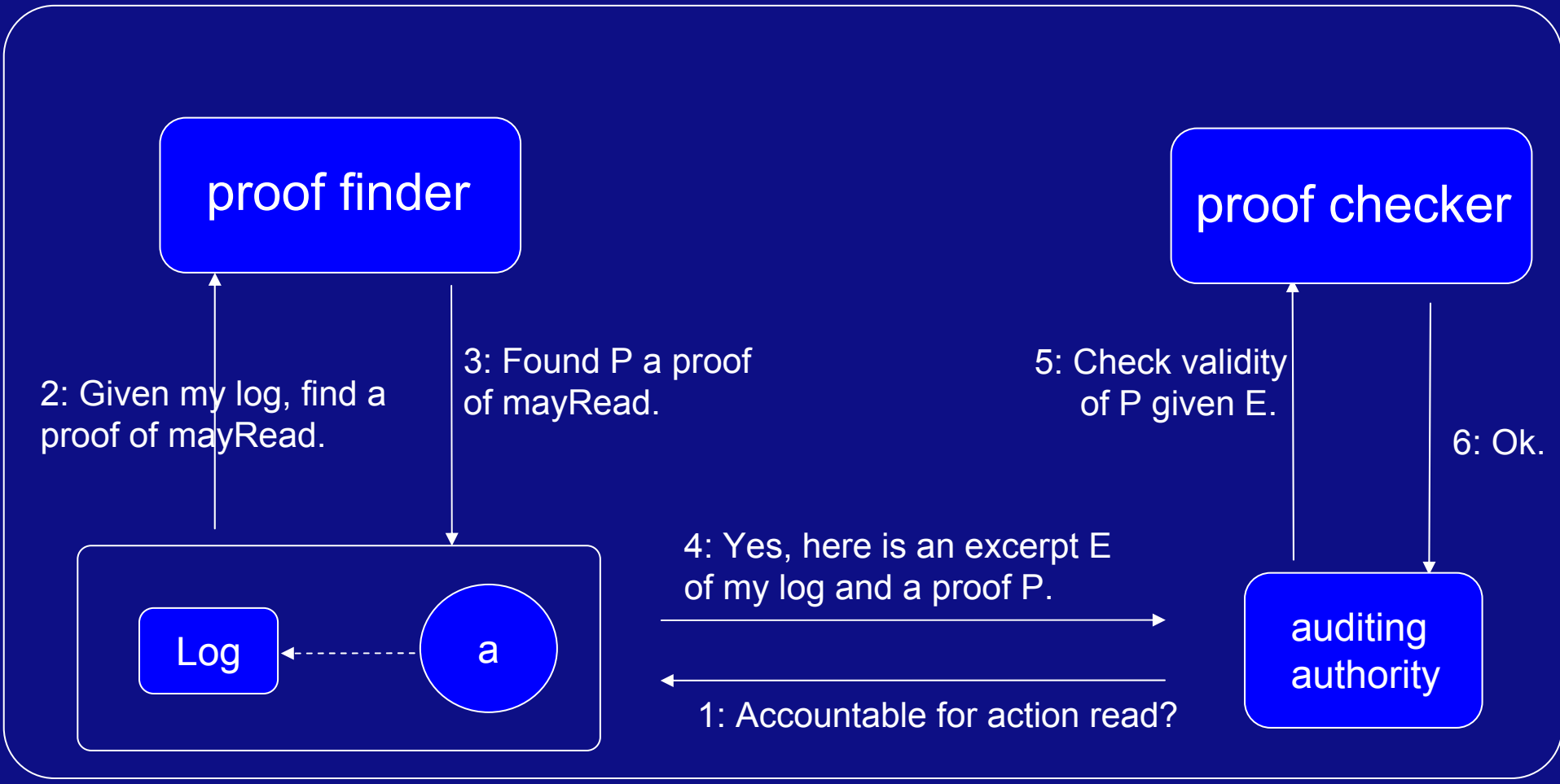
- Intuitionistic Sequent Calculus (Gentzen system, also known as LJ) allows mechanic (bottom up) proof finding.
- Proven soundness and completeness wrt the semantics (natural deduction rules).
- Proven cut-elimination (hence consistency, semi-decidability, and naive implementation in Prolog).

$$\frac{\Gamma_1, \phi_1 \vdash_a \psi}{\Gamma_1, (\phi_1 \wedge \phi_2);; \vdash_a \psi} \wedge L_1$$

```
%% twelf: and_l1 rule
and_l1: entail A (cons Phi Gamma1) Gamma2 Delta Chi ->
          entail A (cons (and Phi Psi) Gamma1) Gamma2 Delta Chi.

%prolog: and_l1 rule
entail(A,Gamma1,Gamma2,Delta,Psi,[Perms,' (and_l1',Pf,')',Bras]):-
                        perm([and(Phi1,_)],Tail,Gamma1,g1,Perms,Bras),
                        (A,[Phi1|Tail],Gamma2,Delta,Psi,Pf).
```

AC² Tool overview



The proof checker is small and fast (this is the TCB), written in Twelf (LF).
The proof finder is more complex (less safe), written in SWI Prolog.

Related work

A-posteriori
- E. Rissanen, Discretionary overriding of access control (2003)

Distributed access control systems
- Abadi et al., A calculus for access control in distributed systems (1993)
- Abadi, Logic in Access Control (2003)
- Appel and Felten, Proof Carrying Authentication (2003)
- Garg and Pfenning, Non-interference in constructive authorization logic. (2006)
- Abadi, Access Control in a Core Calculus of Dependency (2006)
- Fournet et al., A type discipline for authorization policies (2005)
- Becker, Fournet, Gordon, papers on SecPAL (2006, 2007)
- Issues
  - distributed implementation of consumables (linear logic)
  - consistency (non-interference)
  - classic or constructive logic
  - semantics

Disadvantages of the AC^2 wrt Ambient networks

- Audit trail may become large.
- Nothing can be prevented.
- Users are not ‘helped’ by the access control policy.
- Administrative actions are also checked a-posteriori, so rogue users (or virusses) can set off a cascade of ‘bad’ actions.
- Users must be accountable / is this realistic? (e.g. user-hacked cellphone).
- No distinction between systems and users.
- First order logic is semi-decidable (RBAC e.g. is decidable).
- Few built-in constructs
  - trust in users
  - device classes
  - context

New framework - Rough outline

- Distinction between user and device.
- Security decisions (proofs) are found and stored by device.
- Audit trails are cleaned (using the proofs) by a user.
- Override mechanism for trusted users (carte blanche).
- Policy language with built-in constructs and semantics for context and time, users and trust, devices and device states, objects and content.
- Model with built-in actions for private communication, broadcasting, processing and retaining data, processing and cleaning audit trails, a-posteriori/a-priori switch, user device interaction.
- Safety/Privacy queries: Can device process or leak private data without user consent. ???

marnix.dekker@tno.nl

Security Group, TNO ICT, Delft

Distributed and Embedded Systems group, University of Twente

papers available from http://cs.utwente.nl/~dekkermac