



IBM Zurich Research Lab

Enterprise Privacy and Federated Identity Management

Michael Waidner
IBM Zurich Research Lab &
IBM Privacy Research Institute

April 2003

© 2003 IBM Corporation

Outline

1. Motivation
2. Enterprise Privacy Management
3. Federated Identity Management
4. Summary

Joint work of many people: *Paul Ashley, Endre Bangerter, Jan Camenisch, Satoshi Hada, Thomas Gross, Günter Karjoth, Michiharu Kudo, Anna Lysyanskaya, Birgit Pfitzmann, Calvin Powers, Matthias Schunter, Els Van Herreweghen, Michael Waidner.*

1. Motivation



What do people think about privacy?

- Privacy concerns are high, back to pre-9/11 level
- Consumers have clear preferences [Harris 2002]
 - ▶ Security procedures [90%], access control [84%], enforcement [>80%]
 - ▶ Assurance of real privacy practices [91%]
- Market for enterprise privacy management
 - ▶ Until 2002 general focus on client-side technology
 - ▶ Strong push for enterprise-side privacy technology
 - Ad-hoc identity management solutions with weak privacy
 - Integration of identity, privacy and access management
 - ▶ IBM Tivoli Privacy Manager only complete privacy management product [Steve Hunt, Giga, 02/2003]

Privacy-enhancing technologies



Client

- Trusted user device (!?)
- New processes
 - ▶ Local customization
- Privacy policies
 - ▶ Preferences
 - ▶ Negotiation
- Filtering and privacy violation detection
- Identity management
 - ▶ Many pseudonyms
 - ▶ Sharing of personal attributes
 - ▶ Trust establishment
- Customer privacy services
- User interface



Privacy-enabling Infrastructure



Organization

- Communication
 - ▶ End-to-end security
 - ▶ Anonymity
 - Trust
 - ▶ Certified attributes
 - ▶ Authentication
 - ▶ Identity
 - Convenience
 - ▶ SSO
 - ▶ Uncertified attributes
 - Payment and delivery
- Exploration of status quo
 - Process (re-)engineering
 - ▶ Data minimization paradigm
 - ▶ Anonymization techniques
 - Enterprise privacy policies
 - ▶ Creation & maintenance
 - ▶ Recording consent & negotiation
 - ▶ Authorization and enforcement
 - Identity management
 - ▶ Many pseudonyms
 - ▶ Sharing of personal attributes
 - ▶ Trust establishment
 - Customer privacy services
 - Auditing & violation detection

2. Enterprise Privacy Management

*Enterprise Privacy Authorization
Language (EPAL)*

Enterprise Privacy Management

- Enterprises want better privacy for their customers – but need support
- Making good privacy promises is easy and good for the business, keeping them is difficult but necessary ... and needs technology
- Privacy practices implementing the promises must be enforced & controlled
 - ▶ from access control to privacy authorization
 - ▶ enforcement on enterprise data systems
 - ▶ reporting back to data subjects
 - ▶ audit by independent third parties
- Compatibility with laws, regulations, and public promises
 - ▶ easy to understand and maintain by non-technical people
 - ▶ easy to derive new policies from existing ones (laws, corporate, sector, ...)
 - ▶ well-defined relation to P3P
- Requires a new language: EPAL, Enterprise Privacy Authorization Language

Syntactic elements of an EPAL policy

- EPAL definitions
 - ▶ Lists of hierarchies for data user, data category, purpose
 - ▶ Lists for action, container, condition, obligation
 - ▶ container provides an abstract definition of data to be evaluated by conditions
 - ▶ condition use XACML as language; used to check context, consent and other data subject properties

- EPAL policy is list of rules, sorted by priority
 - ▶ default ruling: allow, deny, not-applicable (for: w/in scope, but don't care)
 - ▶ Elements of a rule
 - allow (or deny or obligate)
 - data user⁺ du₁, du₂, ... e.g., "borderless-books"
 - action⁺ a₁, a₂, ... e.g., "read"
 - for purpose p₁, p₂, ... e.g., "book-of-the-month-club"
 - on data category⁺ dc₁, dc₂, ... e.g., "email"
 - under condition^{*} c1(container X₁, X₂, ...), c₂(...), ...
e.g., "/CustRecord/Consent/BookClub=True
&& /CustomerRecord/age>13"
 - yielding obligation^{*} o₁(), o₂(), ... e.g., "write audit"

- Plus management and version info, imported policies, scoping, ...

Semantics of EPAL: Authorization

- Policy maps any well-defined authorization request (data user, action, purpose, data category, container/s) to decision \in {allow, deny, not-applicable} + obligations
- Completion of rule set through inheritance
 - ▶ allow inherits down along hierarchies
 - ▶ deny inherits up and down
- Check rules in given order for applicability
 - ▶ rule covers request directly / by inheritance (efficiently via hash tables)
 - ▶ condition/s are satisfied
- Decision
 - ▶ First applicable deny/allow-rule decides + take rule's obligations + all from all obligate-rules on the way
 - ▶ If there is none then take default ruling + take all obligations from all obligate-rules

Semantics of EPAL: Auto

- Policy maps any well-
(data user, action, p)
to decision $\in \{\text{allow}, \text{d}\}$
- Completion of rule set
 - ▶ allow inherits down along
 - ▶ deny inherits up and down
- Check rules in given order for a
 - ▶ rule covers request directly / by inheritance (efficiently via hash tables)
 - ▶ condition/s are satisfied
- Decision
 - ▶ First applicable deny/allow-rule decides +
take rule's **obligation/s** + all from all obligate-rules on the way
 - ▶ If there is none then take default ruling +
take all obligations from all obligate-rules

EPAL gives abstract policies!

We also need deployment descriptions:

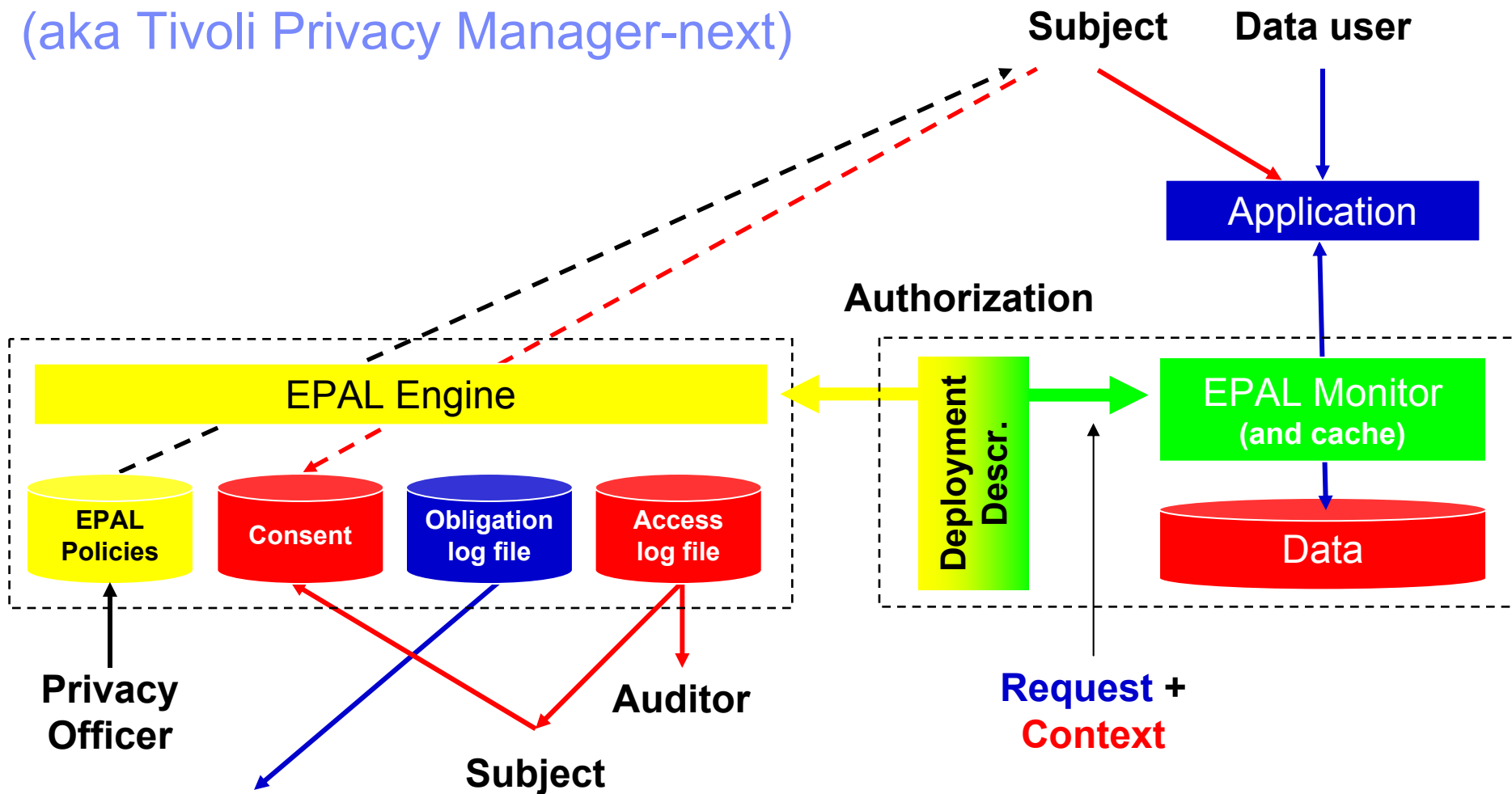
Storage locations \rightarrow data category

Storage locations \leftarrow container

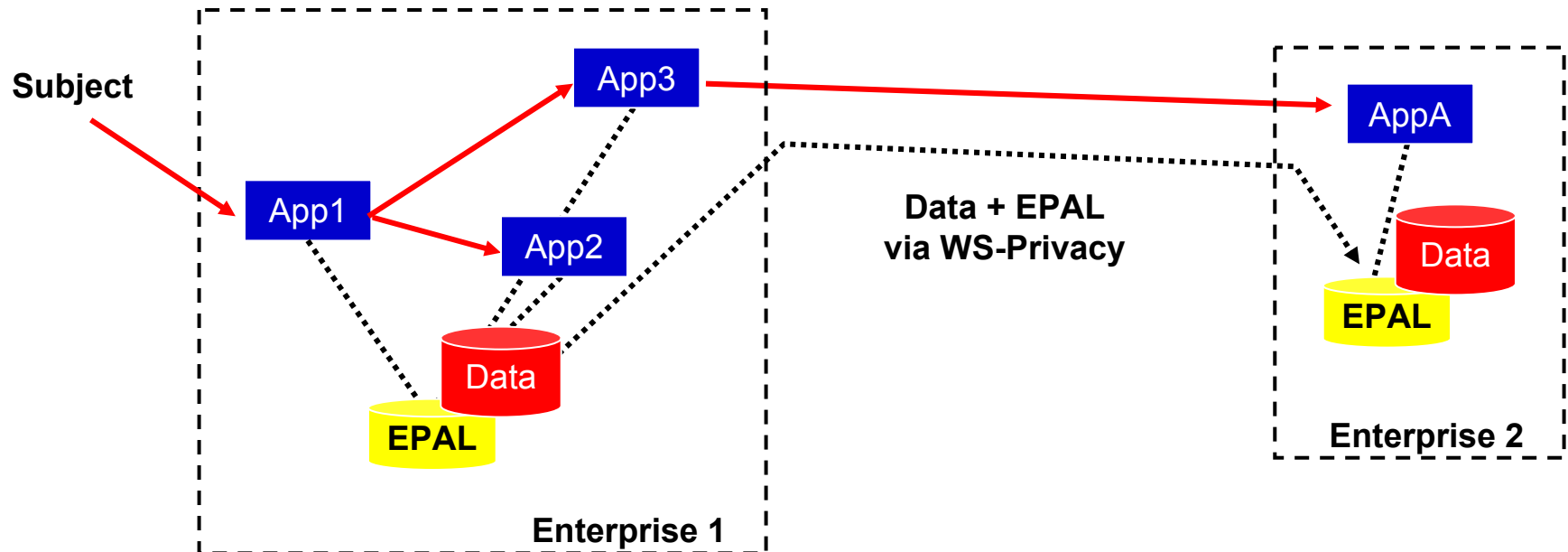
Application calls \rightarrow data user, purpose, operation

Some operations \leftarrow obligation

Enforcement architecture for EPAL (aka Tivoli Privacy Manager-next)



The next step: sticky-policy paradigm across domains



- Define “privacy boundaries” within the infrastructure that encompasses all the places where personal info is stored.
- It should always be possible to determine the applicable privacy policy that was in force when a particular piece of personal info was collected.

EPAL status and plans

- Language
 - ▶ Specification published March 2003, open for comments
 - ▶ Efficient evaluation
 - ▶ Formal standardization depends on feedback

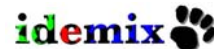
- Enforcement
 - ▶ Target: Tivoli Privacy Manager
 - ▶ Specific integration with WAS & DB2 (aka Hippocratic Database)
 - ▶ Sticky policy paradigm in a Web services context

- Various extensions
 - ▶ Individual-authored policies for collaborative applications
 - ▶ Stateful semantics
 - ▶ Refinement and composition of EPAL

3. Federated Identity Management: *Overview*

Federated Identity Management

- Enterprises want better privacy for their customers – but need support
- Vertical and horizontal integration, across trust domains, requires FIM
- Federated identity management
 - ▶ Get *uncertified* personal attributes to enterprise *B*
 - ▶ Get *certified* attributes from enterprise *A* to *B*, with user's consent
 - ▶ Special case: single sign-on across trust domains
- Essentially the MS Passport / Liberty space
- Our goal:
 - ▶ Full scalability, provable security, efficient (practical, not just polynomial)
 - ▶ Short-term deployable protocol with *better* privacy (optimal for uncertified attr.):
Browser-based Attribute Exchange (BBAE)
 - ▶ Mid-term deployable protocol with *maximum* privacy:
Identity Mix (idemix)

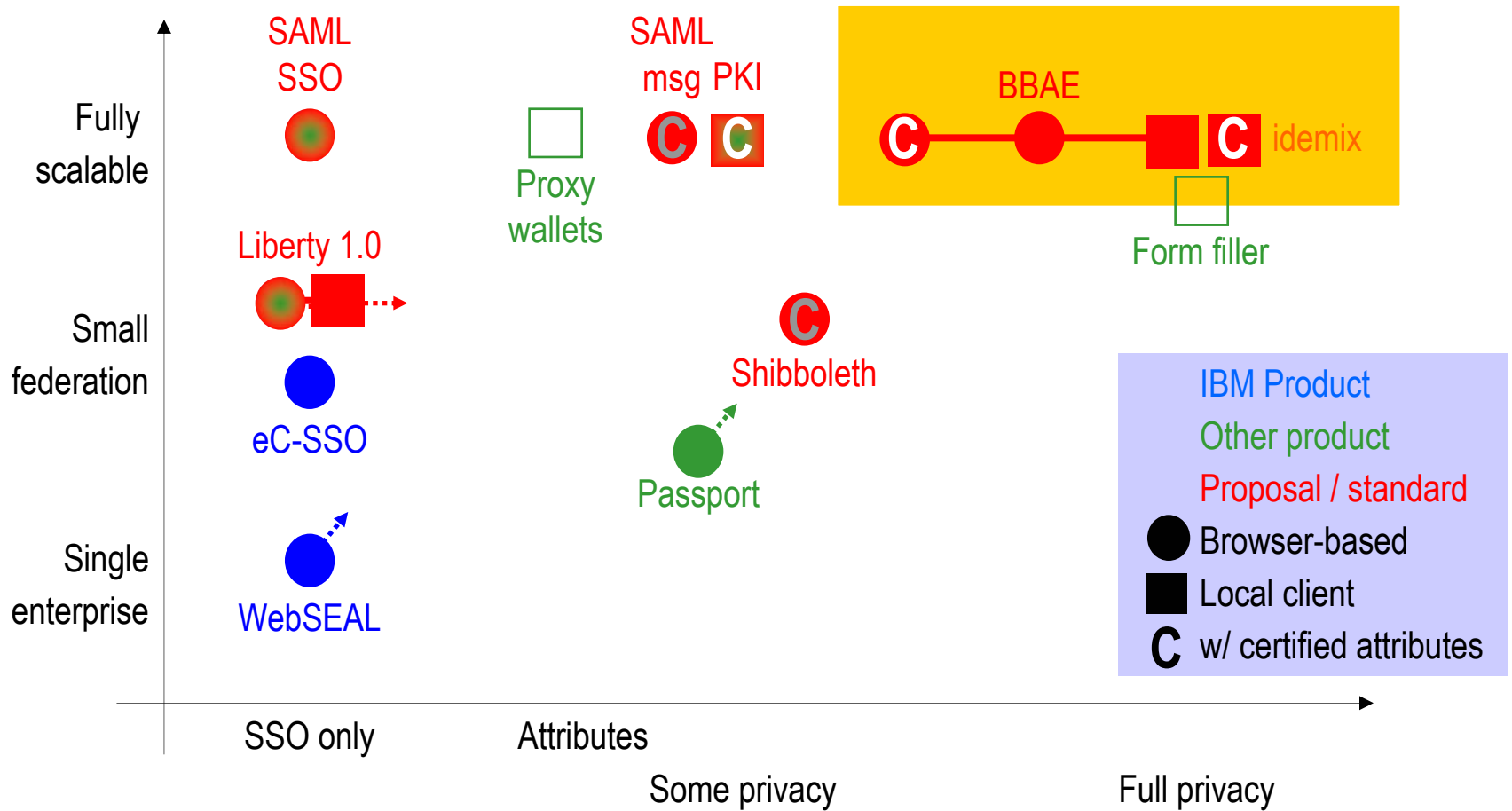


Privacy in federated identity management

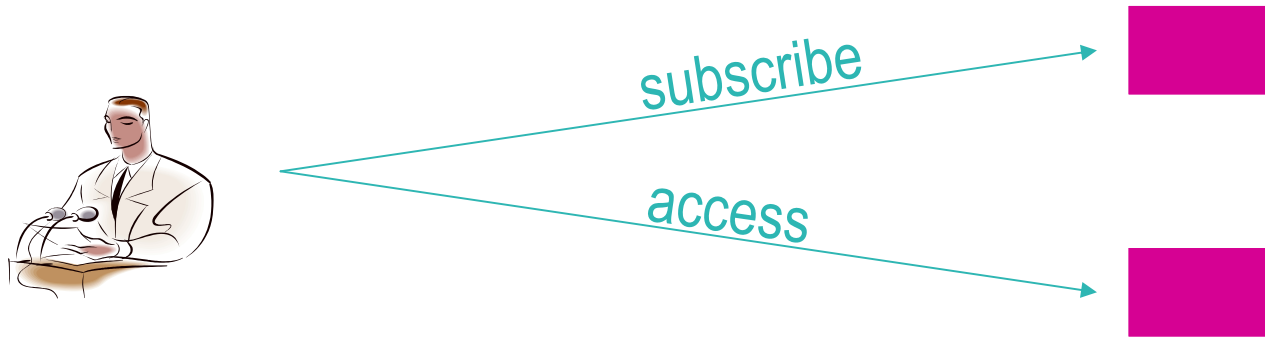
Attributes about P are only given to O, used there, or disclosed to others, with P's informed consent.

- Fundamental implications
 - ▶ Explicit privacy policy (modulo exceptions by law)
 - ▶ Avoid unnecessary identification: multiple pseudonyms per person
 - ▶ Avoid trust bottlenecks: multiple wallet holders, including local wallets
- Covers explicit attributes
 - ▶ Pseudonym, name, address, salary, blood group, ...
 - ▶ Facts, like “regular IBM employee” or “salary above 100'000 USD”
- Should also cover implicit attributes, as much as possible
 - ▶ Traffic patterns (e.g., browsing history)
 - ▶ Identifiable representation of non-identifiable attributes (e.g., attribute certificates)
 - ▶ Personal writing style

Existing proposals



idemix – hypothetical example



- Patent database

- ▶ Access requires paid subscription
- ▶ Subscription is proven by showing a valid certificate signed by the operator
- ▶ Enables operator to track the queries of each subscriber ...
- ▶ ... which many subscribers perceive as a breach of confidentiality

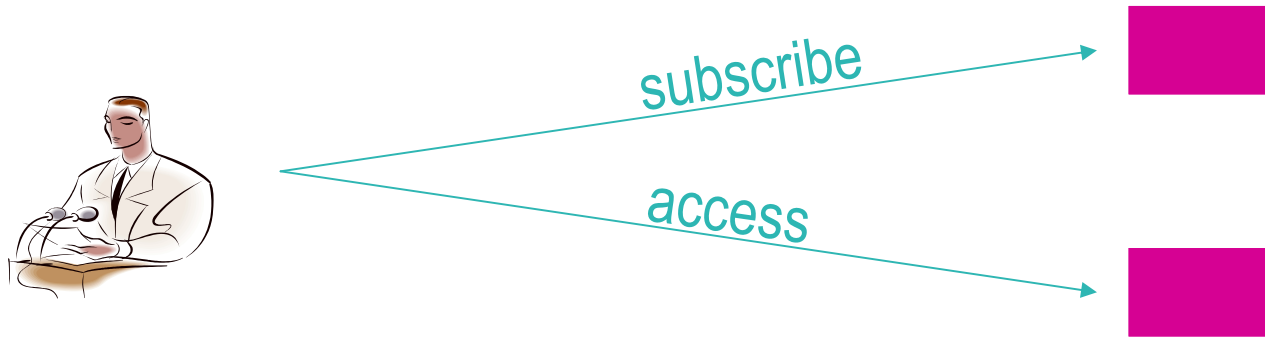
How idemix solves this problem

- Step 1: Pseudonyms
 - ▶ Organizations know individuals by pseudonyms only

 - Step 2: Control attributes
 - ▶ Only necessary attributes are shown

 - Step 3: Standardize attributes
 - ▶ Effective only if shown attributes don't identify individuals (rather an application requirement ...)
-
- Step 4: Prove knowledge of cert's
 - ▶ Certificates are kept secret, only their possession is shown (zero-knowledge proofs of knowledge)

idemix – hypothetical example (cont.)



- Patent database
 - ▶ Only possession of a valid subscription is proven
 - ▶ Certificate itself is never sent back to the organization O
 - ▶ O does not recognize repeated “shows” of the same certificate

idemix – features

- Security features
 - ▶ Unforgeability of credentials (based on SRSA)
 - ▶ Unlinkability of shows (based on DDH)
 - ▶ Prevention of credential sharing and pooling (based on DL)
- Optional features
 - ▶ Anonymity revocation
 - Local deanonymization: nym on which credential was issued
 - Global deanonymization: nym of user with a dedicated Root Authority
 - ▶ One-show credentials (e.g., for e-cash)
 - ▶ Credential revocation
- Reasonable performance for a PKI

Performance

- Test system
 - ▶ IBM Thinkpad T23 (1.3 MHz Pentium 3)
 - ▶ Debian Linux
 - ▶ Java 1.3.1 (Blackdown)
- Further optimizations (overlap of computation): expected speedup x2

	options	time (sec)
RegNym		0.1-0.2
GetCred	any	1.0-1.5
ShowCred	no	0.9-1.2
ShowCred	+ one-show	+0.2
ShowCred	+ expir. date	+0.2-0.4
ShowCred	+ local deanonym.	+0.3-0.5
ShowCred	+ global deanonym.	+0.3-0.5
ShowCred	all options on	1.9-2.4

4. Summary

Summary

- Privacy is a business issue, in need of policies and technologies
- Two important areas for the near future:
 - ▶ Enterprise privacy management
 - ▶ Federated identity management
- EPAL proposal for fine-grained privacy enterprise policies
- Federated identity management may become almost mandatory
 - ▶ Privacy is important: make it the normal case
- Efficient, user-friendly protocols possible
 - ▶ BBAE as short-term implementable alternative
 - ▶ idemix as mid-term full-privacy alternative
 - ▶ Compatible with Web services strategy
 - ▶ Integration with client-side technology (smartcard, TCPA) possible

For more information ...

- How to reach me
 - ▶ Notes Michael Waidner/Zurich/IBM@IBMCH
 - ▶ email wmi@zurich.ibm.com
 - ▶ Web <http://www.zurich.ibm.com/~wmi>

- Privacy at IBM
 - ▶ IBM privacy products and services (public):
<http://www.ibm.com/security/privacy>

 - ▶ IBM Privacy Research Institute (public):
<http://www.research.ibm.com/privacy>

 - ▶ Privacy at IBM (internal):
<http://w3.ibm.com/privacy>