



E-Privacy – Privacy in the Electronic Society

Privacy Policy Languages *Enterprise Privacy Architectures*

Günter Karjoth

Spring term 2009

Potential semantic inconsistencies in P3P policies

- multiple retention values apply to one data item;
- a statement has conflicting purposes and retention values;
 - purpose historical with retention no-retention
- a statement has conflicting purposes and recipients;
 - all purposes but only recipient delivery
- a statement has conflicting purposes and data items
 - purpose contact but no data items physical or online

What is the difference ?

Policy 1:

```
stmt( data: {#user.home-info.telecom,  
            #user.bdate(optional)},  
      purpose: {individual-analysis,  
               telemarketing(opt-in)},  
      recipient: {ours},  
      retention: {stated-purpose} )
```

Policy 2:

```
stmt( data: {#user.home-info.telecom,  
            #user.bdate(optional)},  
      purpose: {individual-analysis},  
      recipient: {ours},  
      retention: {stated-purpose})  
stmt( data: {#user.home-info.telecom,  
            #user.bdate(optional)},  
      purpose: {telemarketing(opt-in)},  
      recipient: {ours},  
      retention: {stated-purpose} )
```

Policy 3:

```
stmt( data: {#user.home-info.telecom},  
      purpose: {individual-analysis,  
               telemarketing(opt-in)},  
      recipient: {ours},  
      retention: {stated-purpose} )  
stmt( data: {#user.bdate(optional)},  
      purpose: {individual-analysis,  
               telemarketing(opt-in)},  
      recipient: {ours},  
      retention: {stated-purpose} )
```

Modelling P3P Policies

<STATEMENT>

<PURPOSE><current/><contact required="opt-in"/> </PURPOSE>

<RECIPIENT><ours/></RECIPIENT>

<RETENTION><business-practices/></RETENTION>

<DATA-GROUP>

<DATA ref="#customer.home-info.postal"/>

<DATA ref="#dynamic.miscdata"/>

<CATEGORIES><demographic/><physical/></CATEGORIES> </DATA>

</DATA-GROUP>

</STATEMENT>

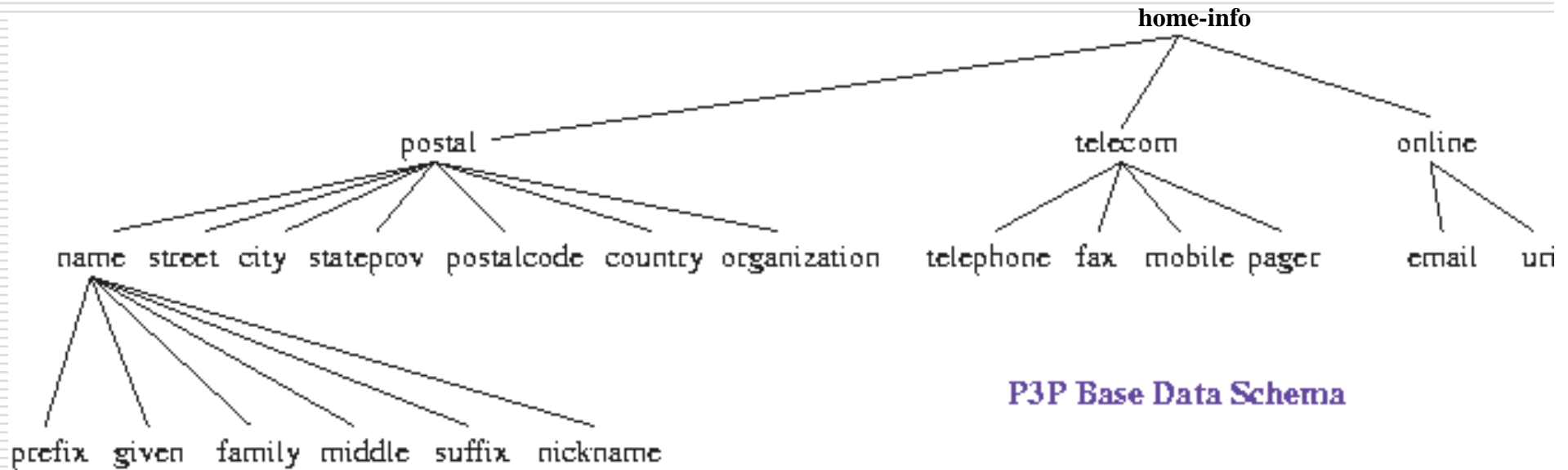
- A P3P statement σ can be interpreted as a function $P[\sigma]$, from a data hierarchy D to subsets of actions $\wp(A)$.
- P3P has a union semantics:

$$P[p]d = \bigcup_{\sigma \in p} P[\sigma]d$$

because it permits an action if permitted by at least one statement.

- How can you verify that a service provider has promised **not** to perform action a on data d ?

P3P Base Data Structures



•user

•user.home-info

telemarketing

•user.home-info.postal

•user.home-info.postal.name

contact (but not telemarketing)

•user.home-info.postal.name.family

The model

- A – set of all potential actions (including purpose, recipient, retention, etc.)
- D – set of data objects, ordered by \leq_D , such that if $d_1 \leq_D d_2$ then d_1 is a component of d_2 .
- A privacy policy, p , is a function from D to subsets of A : $p(d) \subseteq \wp(A)$
 - a service provider who performs action a on data d has violated p if $a \notin p(d)$
 - $a \in p(d)$ does not imply action a may be performed on data d because the provider effectively performs a on all $d' \leq_D d$

Perspectives and Modalities

- **Consumers** want privacy policies that permit **as little use** of their personal data **as possible**.
 - use an upper bound in case of incomplete or conflicting information
 - ◇ - ability to perform an action using some parts of data

- **Service providers** want privacy policies that permit **as much use** of their customers' personal data **as is useful**.
 - use an lower bound
 - □ - ability to perform an action using all parts of data

Perspectives and Modalities (cont'd)

Notice that a single detailed policy allowing some but not all data be used for telemarketing,

- a privacy-conscious **consumer** would like the policy be *summarized* (using the \diamond modality) as “allows telemarketing” so the consumer can reject the policy, whereas
- a **service provider** who has committed to the policy must *summarize* the policy (using the \square modality) as “prohibits telemarketing” in order to avoid violating the policy when it is enforced using the policy summary.

$$p_s \sqsubseteq p \sqsubseteq p_c$$

Kripke model

The Kripke model for policy p is $(D, \leq_D, \text{ and } \Vdash_p)$, where for all $d \in D$ and all $a \in A$,

$$d \Vdash_p a \iff a \in p(d).$$

We extend the \Vdash relation to modal formulae in the standard manner:

$$d \Vdash_p \varphi_1 \wedge \varphi_2 \iff d \Vdash_p \varphi_1 \text{ and } d \Vdash_p \varphi_2$$

$$d \Vdash_p \neg \varphi \iff d \not\Vdash_p \varphi$$

$$d \Vdash_p \Box \varphi \iff (\forall d' \leq_D d) (d' \Vdash_p \varphi)$$

and adopt the convention that $\Diamond \varphi \equiv \neg \Box \neg \varphi$.

Enforcement

Intuitively, one policy entails another if the former policy makes a stronger promise!

Given privacy policies p and q with a common data hierarchy D , p **entails** q , written $p \sqsubseteq q$, if for all $d \in D$,

$$p(d) \subseteq q(d)$$

Intuitively, one policy enforces another if any action performed under the former is announced under the latter.

Given policies p and q with data hierarchy D_p and D_q , respectively, and $D_p \subseteq D_q$, q **enforces** p if and only if

$$d \Vdash_q \varphi \Rightarrow d \Vdash_p \varphi$$

for all $d \in D_p$ and all positive simple, modal formulae φ .

Privacy Preferences

- f is a privacy preference if it is a unary predicate on policies
- privacy preference f is robust if, for all policies $p \sqsubseteq p'$, $f(p') \Rightarrow f(p)$
 - a robust preference that accepts a policy will also accept a more restrictive policy
 - is an upper bound on policy permissiveness

Example: A preference to block web sites that use home telephone numbers for telemarketing is robust, whereas a preference to block web sites that *do not use* home telephone numbers for telemarketing is not robust.

A P3P Preference Exchange Language (APPEL)

An APPEL preference consists of a set of rules, each giving a judgment (**block**, **limited**, or **request**) and a condition under which to issue that judgment:

- rules are processed in order, halting at the first rule that matches
- a rule matches if its pattern (expression) is satisfied by the policy
- every APPEL expression has a connective attribute that defines the logical operators between its subexpressions:
or, and, non-or, non-and, or-exact, and-exact

Example:

1. Requests for **personal information** which will be given out to **3rd parties** should be **rejected**.
2. The user **does not mind** revealing **click-stream** and **user agent** information to sites that collect no other information. However, she insists that the service provides some form of **assurance**.
3. All **other requests** for data transfer should result in a **warning** (indicating a conflict with her privacy preferences).

APPEL rule

```
<appel:RULE behavior="limited" prompt="yes"
  description="Warning! Data may be shared.">
  <p3p:POLICY>
    <p3p:STATEMENT>
      <p3p:RECIPIENT appel:connective="or" >
        <p3p:same/>
        <p3p:other-recipient/>
        <p3p:public/>
        <p3p:unrelated/>
      </p3p:RECIPIENT>
    </p3p:STATEMENT>
  </p3p:POLICY>
</appel:RULE>
```

pattern {

description

connective

- or
- and
- non-or
- non-and
- and-exact
- or-exact

Behavior

- request
- block
- limited

Example Ruleset

```
<appel:APPEL xmlns:APPEL="http://www.w3.org/2001/02/APPELv1">  
  <appel:RULESET crtddb="W3C" crtndon="1999-11-03T09:21:32-05:00">
```

```
    <appel:RULE behavior="block"  
      description="Service collects identifiable data for 3rd parties">  
      <POLICY><STATEMENT>  
        <DATA-GROUP><CATEGORIES connective="or">  
          <physical/><demographic/><userid/>  
        </CATEGORIES></DATA-GROUP>  
        <RECIPIENT connective="or">  
          <same/><other-recipient/><public/><delivery/><unrelated/>  
        </RECIPIENT>  
      </STATEMENT></POLICY>  
    </appel:RULE>
```

```
    <appel:RULE behavior="request"  
      description="Service only collects clickstream data">  
      <POLICY><STATEMENT>  
        <DATA-GROUP connective="or-exact">  
          <DATA name="Dynamic.HTTP.UserAgent"/>  
          <DATA name="Dynamic.ClickStream.Server"/>  
        </DATA-GROUP>  
      </STATEMENT>  
      <DISCLOSURE discURI="*" />  
      <DISPUTES-GROUP><DISPUTES org="*" /></DISPUTES-GROUP>  
    </POLICY>  
  </appel:RULE>
```

```
  <appel:RULE behavior="limited" description="Suspicious Policy. Beware!">  
    <appel:OTHERWISE/>  
  </appel:RULE>
```

An Example Preference

```
<appel:RULESET>
  <appel:RULE behavior="block">
    <POLICY>
      <STATEMENT>
        <PURPOSE connective="or">
          <contact/>
          <telemarketing/>
        </PURPOSE>
      </STATEMENT>
    </POLICY>
  </appel:RULE>

  <appel:RULE behavior="request">
    <appel:OTHERWISE/>
  </appel:RULE>
</appel:RULESET>
```

Block sites whose policies indicate that the information collected can be used for *contact* or *telemarketing*.

A non-robust APPEL preference

```
<appel:RULE behavior="block"  
<POLICY>  
  <STATEMENT connective="or">  
    <PURPOSE connective="non-and">  
      <telemarketing/>  
    </PURPOSE>  
  </STATEMENT>  
</POLICY>  
</appel:RULE>
```

The connective *non-and* causes the rule to fire for policies that do not disclose the *telemarketing* purpose.

XPref [Agrawal *et al*, 2003]

- 😊 uses minimal subset of XPath
- 😊 provides significant expressive power
- 😊 there is an APPEL to XPath translation

- 😞 can express non-robust preferences
- 😞 using XPref robustly is non-intuitive

Approximations to a robust XPref preference

```
<RULE behavior="block" (A)
  condition= "/POLICY/STATEMENT/DATA-GROUP/*
  [ name(.) = "DATA" and @ref = "user.home-info.postal" ]" />
```

```
<RULE behavior="block" (B)
  condition= "/POLICY/STATEMENT/DATA-GROUP/*
  [ name(.) = "DATA" and ( @ref = "user.home-info.postal" or
                           @ref = "user.home-info" or
                           @ref = "user.home" ) ]" />
```

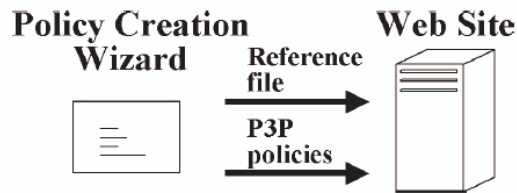
```
<RULE behavior="block" (C)
  condition= "/POLICY/STATEMENT/DATA-GROUP/*
  [ name(.) = "DATA" and
    ( starts-with(@ref = "user.home-info.postal") or
      @ref = "user.home-info" or
      @ref = "user.home" ) ]" />
```

Block services that use my home address:

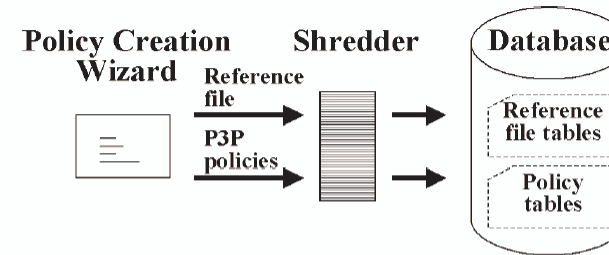
`#user.home-info.postal` \Vdash $\diamond a$

Implementing P3P Using Database Technology

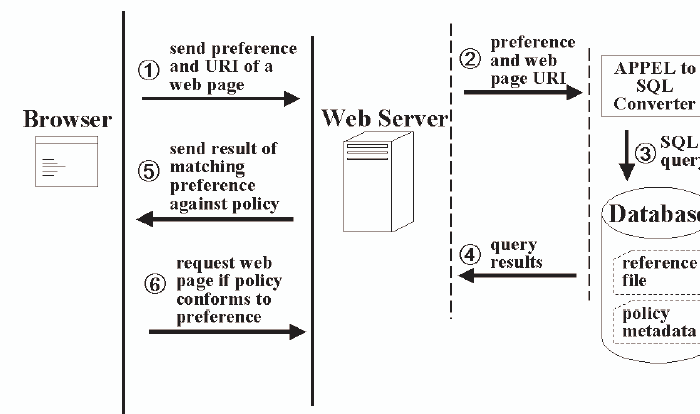
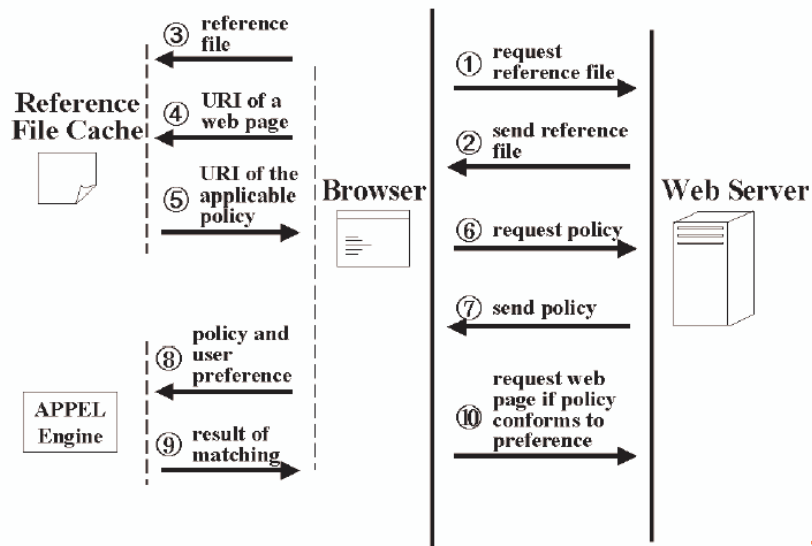
[Agrawal et al, 2003]



client-centric



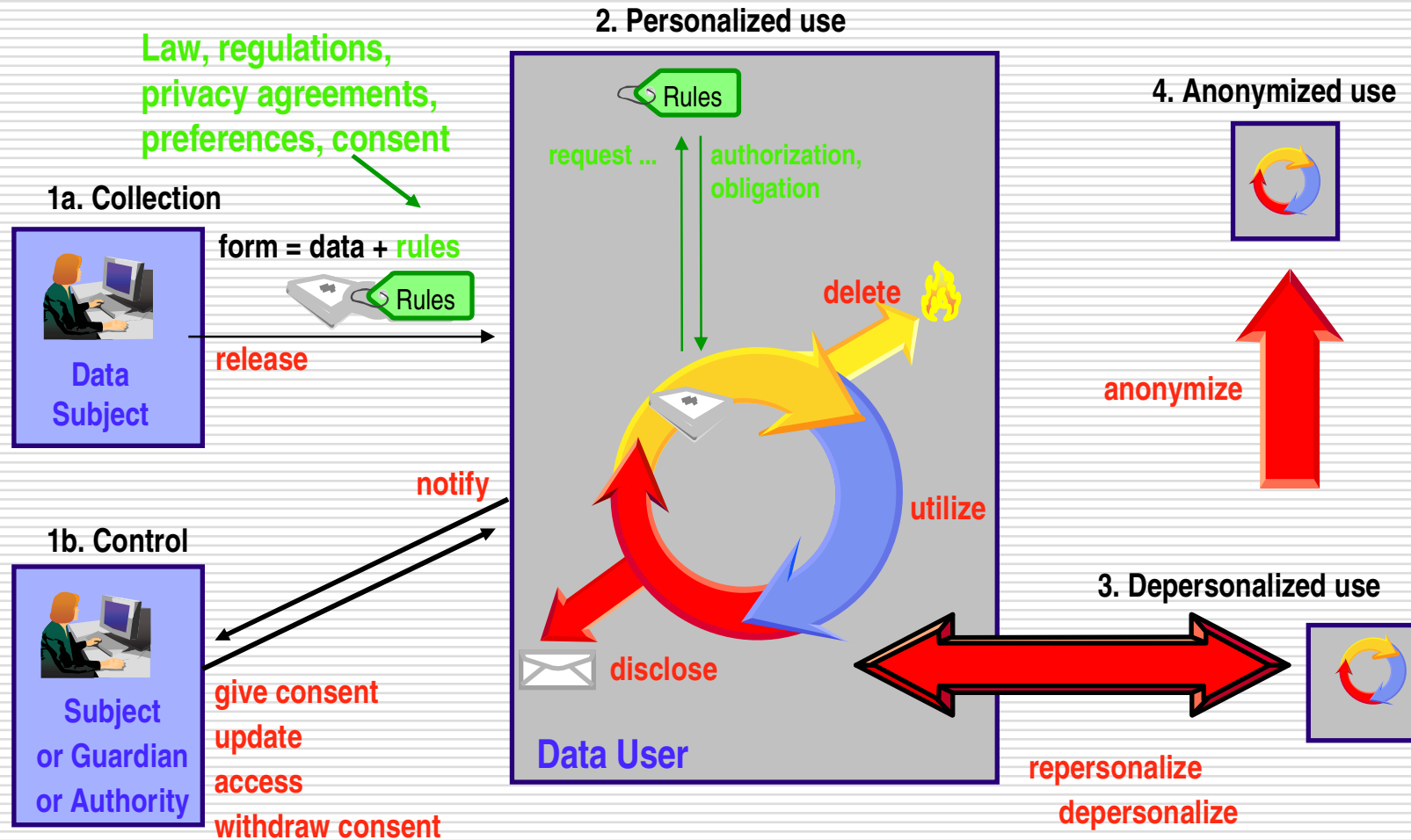
server-centric



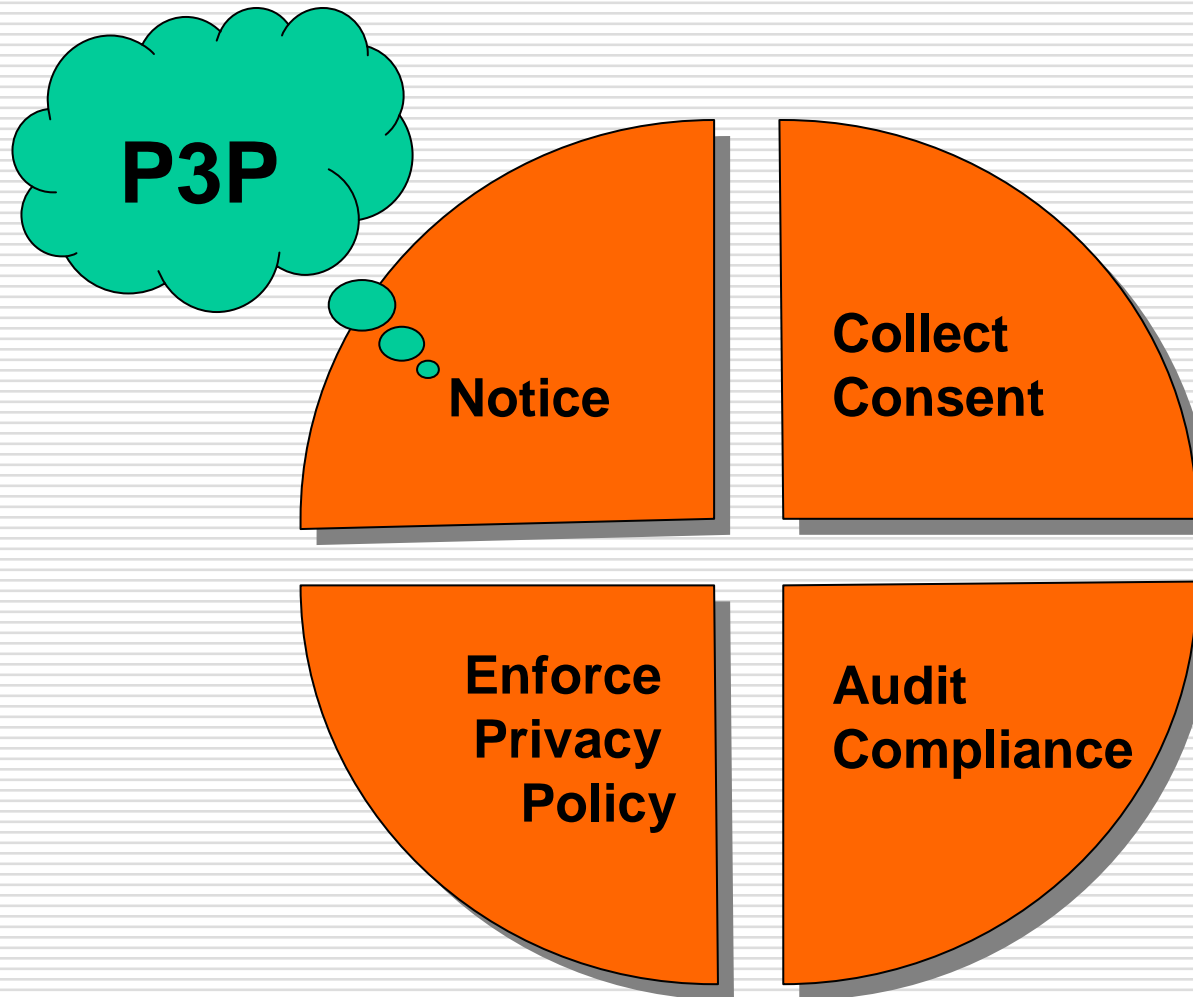
→ 15-30 times faster

→ consumer must trust server

EPA: Privacy-aware business process design

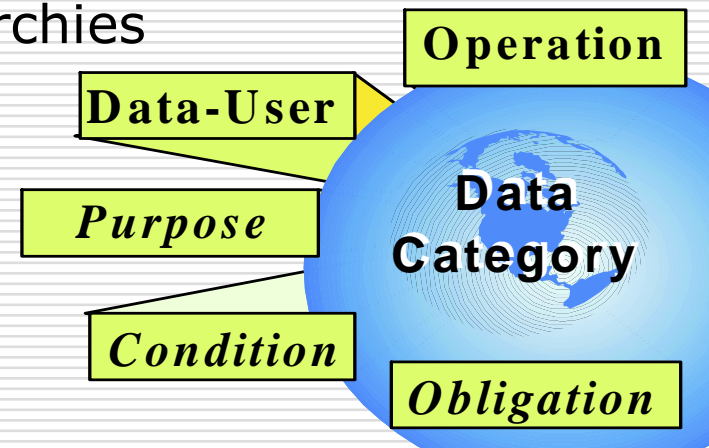


„The Privacy Pie“



Inside Business: Enriched Privacy Policy

- Vocabulary defines scope:
 - Data, users, and purposes as hierarchies
 - Operations, obligations as lists



- Rules authorize access:

A [user category] should be [allowed or denied] the ability to perform [action] on [data category] for [purpose] under [condition] yielding an obligation to [obligation].

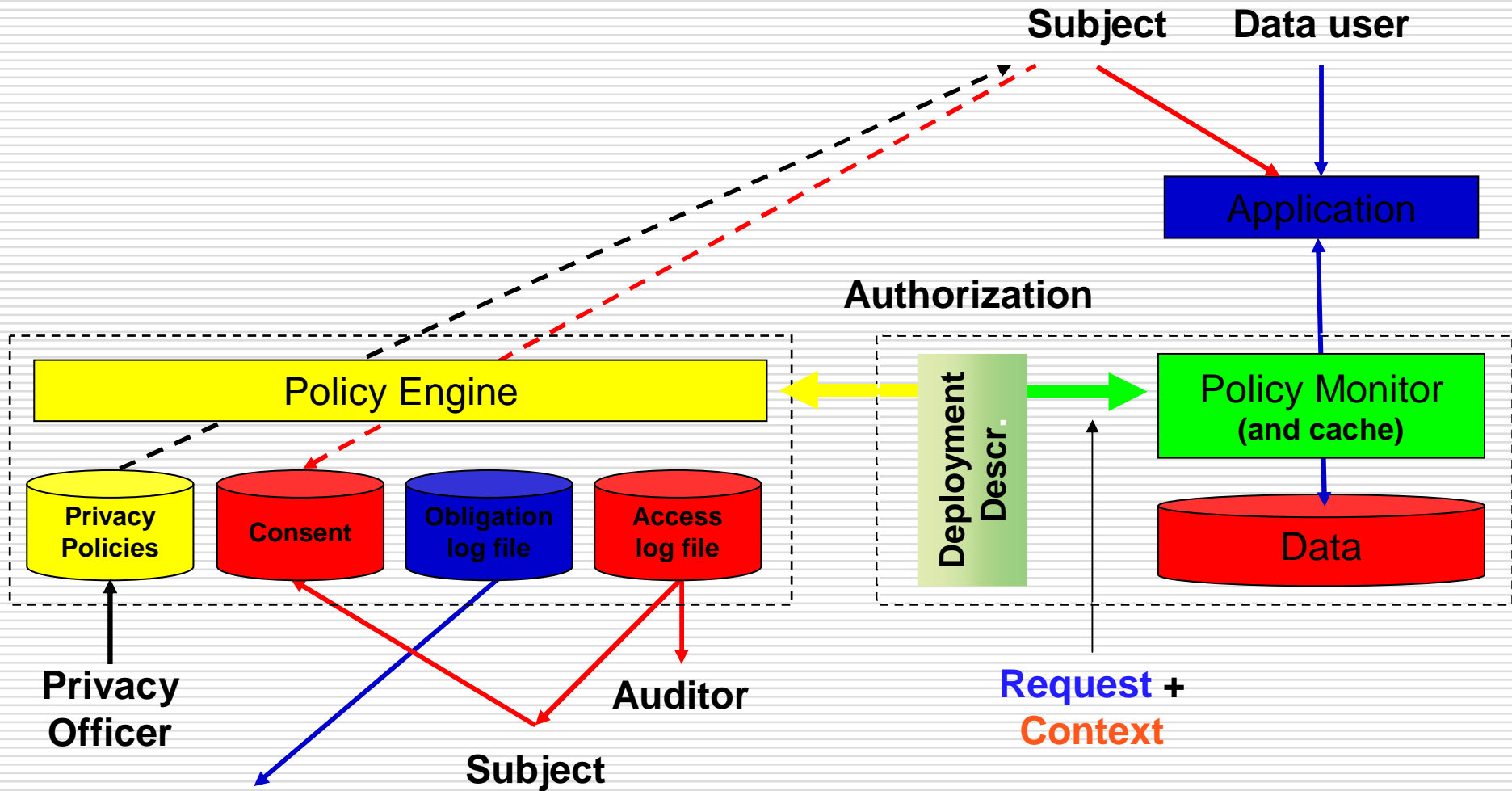
Example:

“A nurse is allowed to perform write on GeneralMedicalInformation for MedicalFollowUp under InCharge(patient) yielding an obligation to notify(physicianInCharge).”

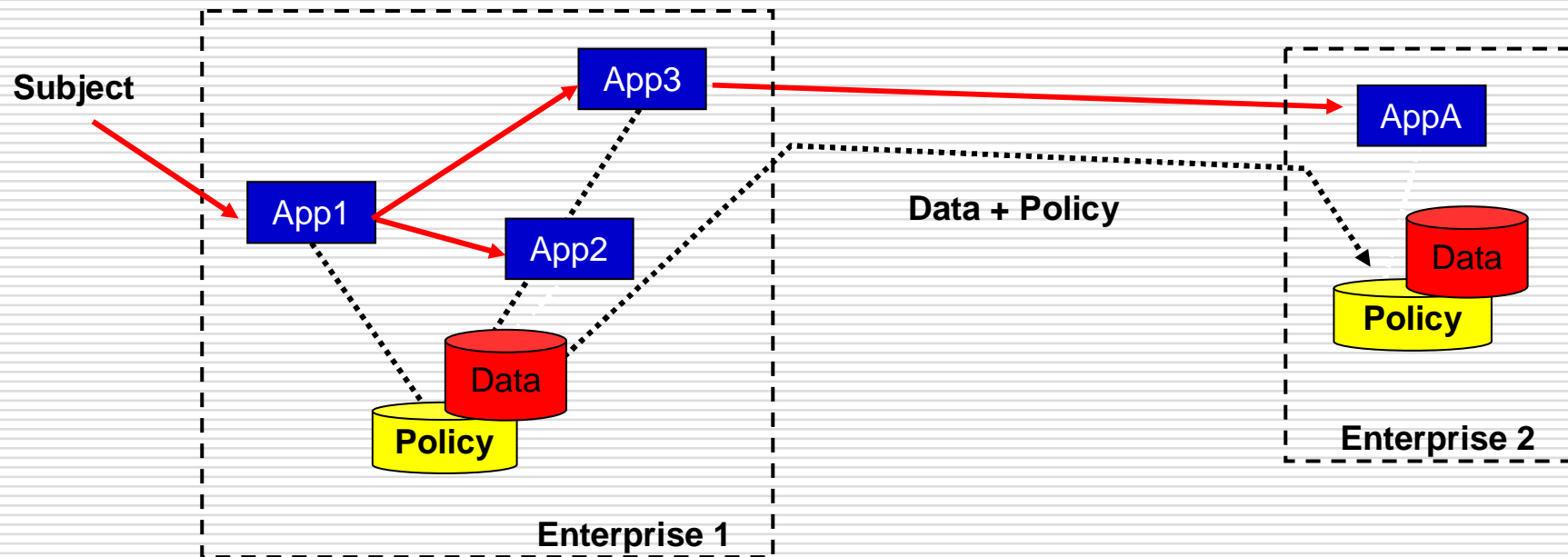
Choices for Enforcing Privacy

- **Do nothing and pray**
- **Coding privacy policy into applications**
 - cost of coding and maintenance becomes prohibitive
 - time to change to a new policy is far too large.
 - each of the applications has to be modified for each policy change
 - difficult reporting and auditing
- **Centralized Enforcement Infrastructure**
 - centralized consent and policy management
 - centralized auditing and reporting
 - distributed enforcement

Enforcement Architecture

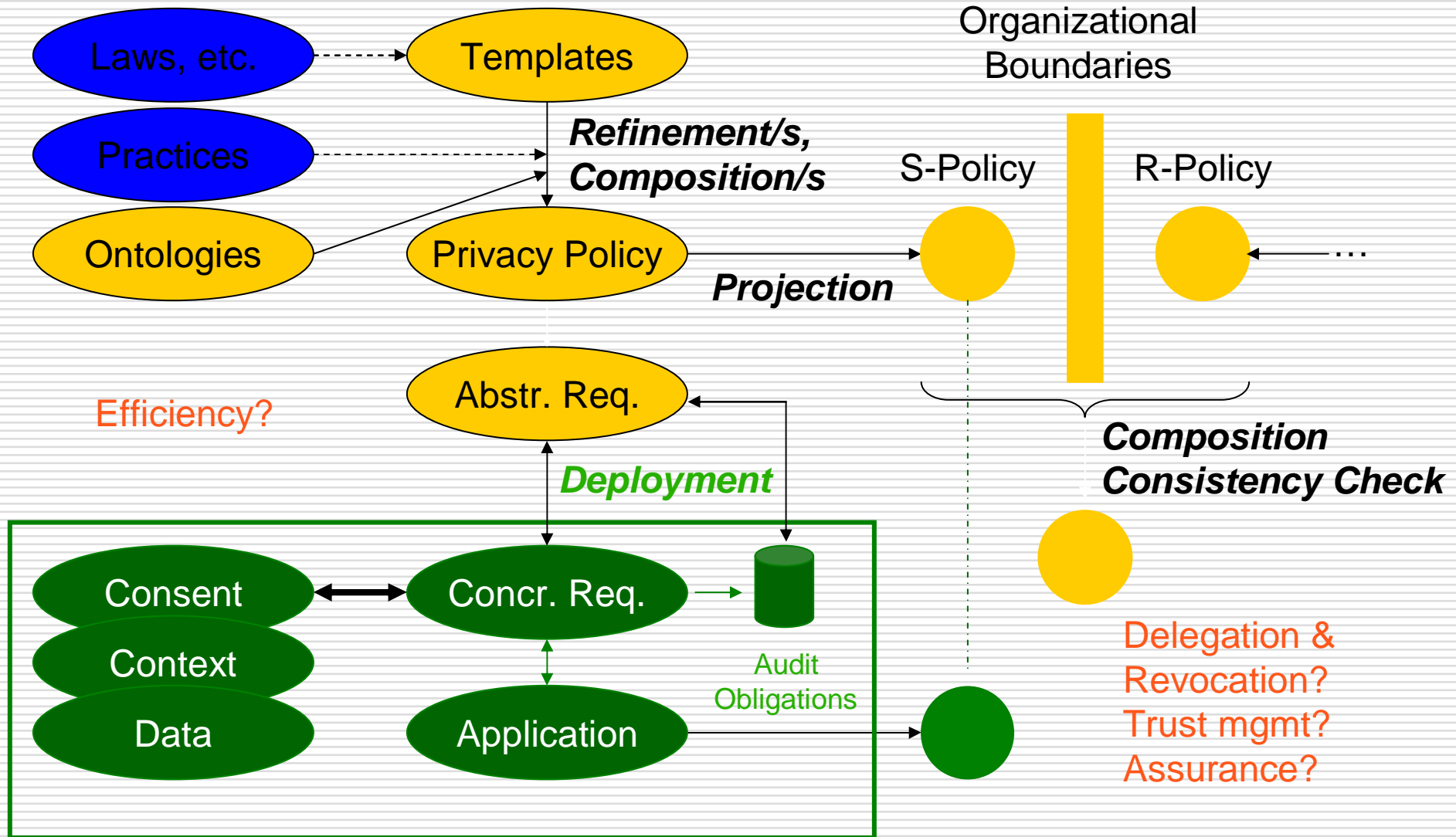


Sticky-policy paradigm across domains



- 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.

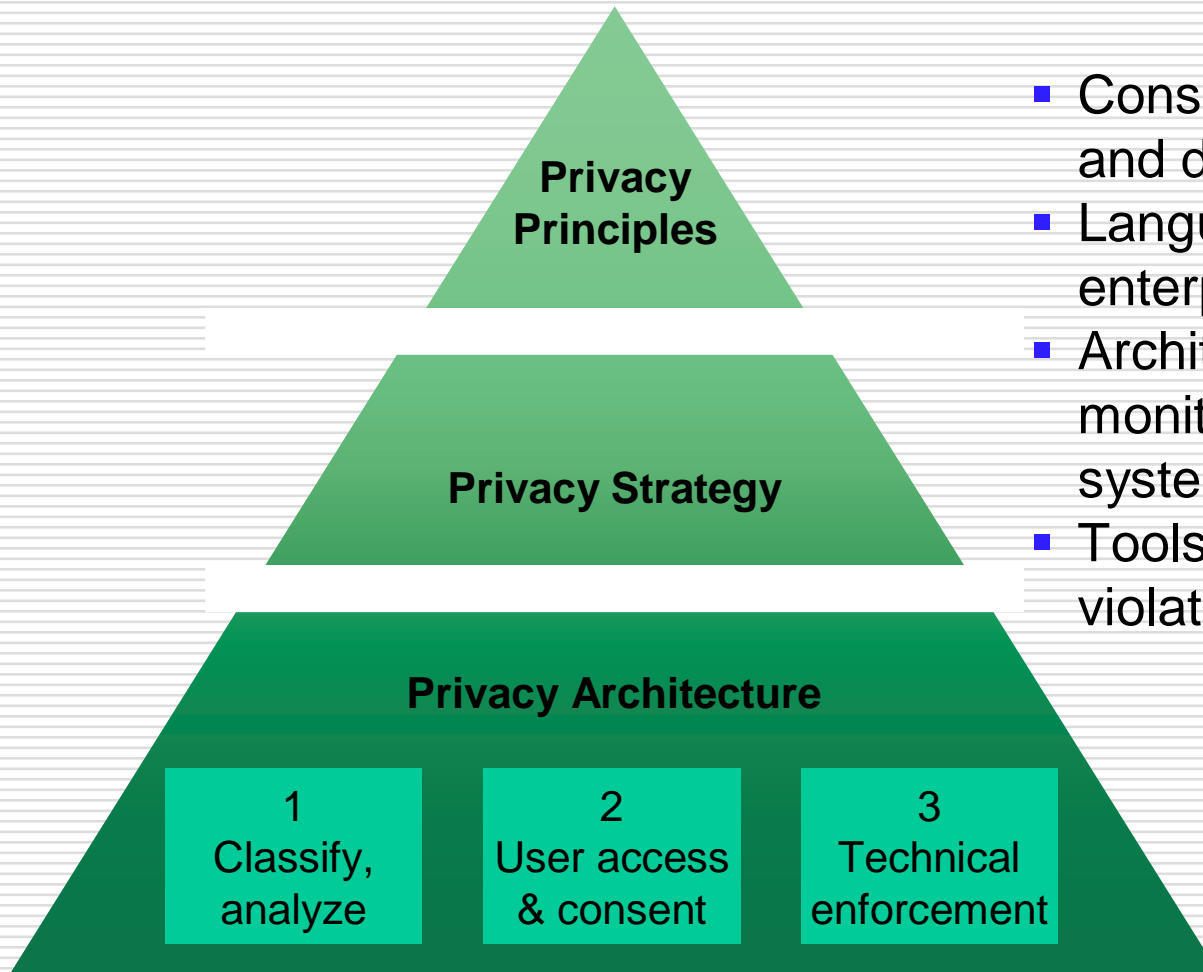
What can be done with a policy?



Desirable properties of a privacy language

- guaranteed consistency
 - guaranteed safety
 - admitting local reasoning
 - closed under combination
-
- all four goals cannot be achieved simultaneously in any language with minimum level of expressiveness
 - sequential semantics forces consistency

IBM Enterprise Privacy Architecture (EPA)



- Consulting method for analysis and design
- Language for describing enterprise privacy policies
- Architecture for enforcing (or monitoring) policies on data systems
- Tools for detecting privacy violations

<http://www.zurich.ibm.com/security/enterprise-privacy/>

Reference: Government of Alberta

Summary

- ❑ current preference languages work only on the syntax of privacy policies
- ❑ consumers and service providers have distinct perspectives
- ❑ privacy promises must separate these modalities to have a sound link between promises and enforcement
- ❑ P3P policies enforce their compact policies

References

- R. Agrawal, J. Kiernan, R. Srikant and Y. Xu: "An XPath-based Preference Language for P3P." 12th WWW Conference, 2003
- R. Agrawal, J. Kiernan, R. Srikant and Y. Xu: "Implementing P3P Using Database Technology." 19th Int'l. Conf. on Data Engineering (ICDE), 2003
- A. Barth and J.C. Mitchell: "Enterprise Privacy Promises and Enforcement." WITS'05, 2005.
- T. Yu, N. Li, and A.I. Antón: "*A Formal Semantics for P3P*". *ACM Workshop on Secure Web Services (SWS)*, 2004.
- G. Karjoth and M. Schunter: "A Privacy Policy Model for Enterprises." 15th IEEE Computer Security Foundations Workshop, 271-281, 2002.
www.csl.sri.com/programs/security/csfw/csfw15/slides/karjoth.pdf