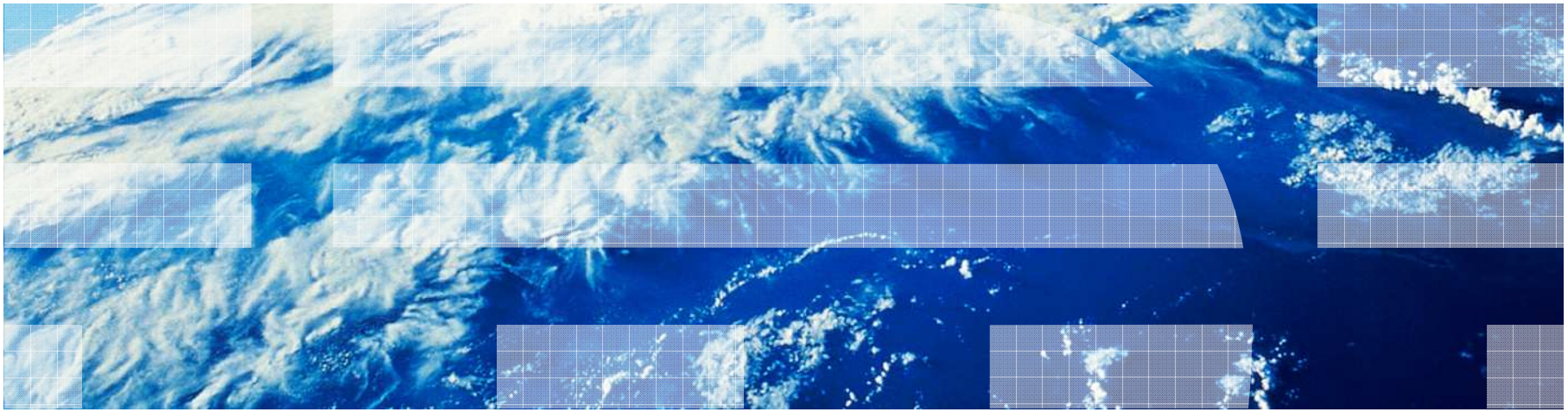
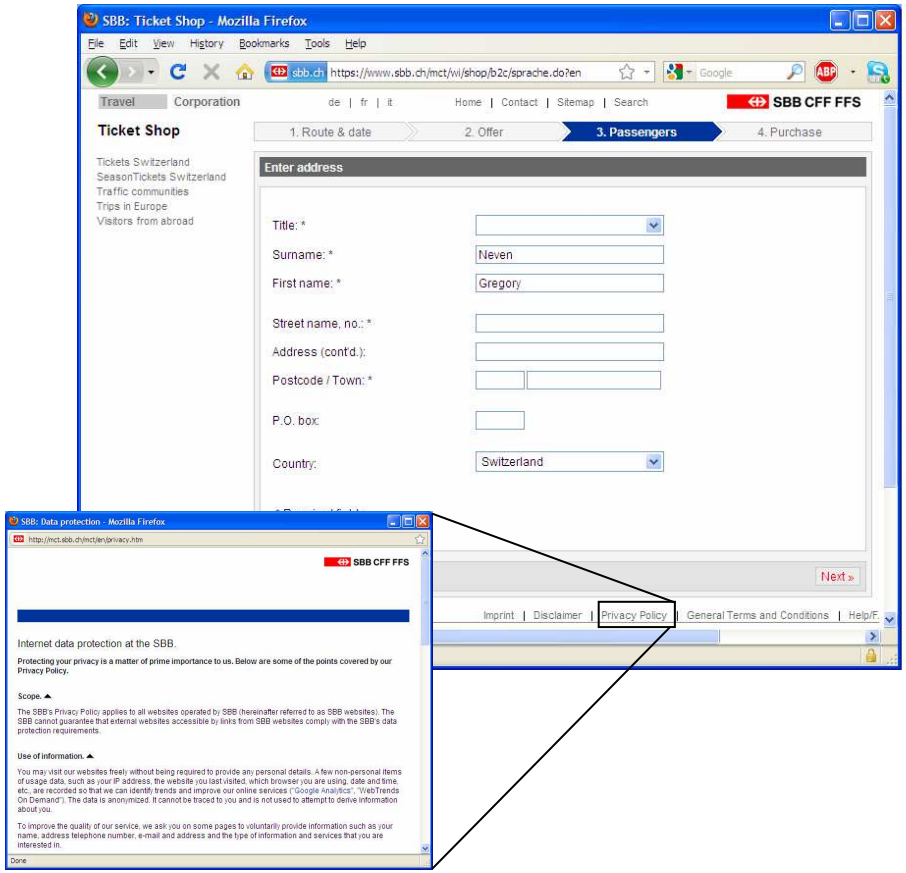
Gregory Neven, IBM Research – Zurich
PrimeLife/IFIP Summer School, August 2-6, 2010, Helsingborg (Sweden)

PPL & Friends:

Privacy Policies in PrimeLife



The whole idea

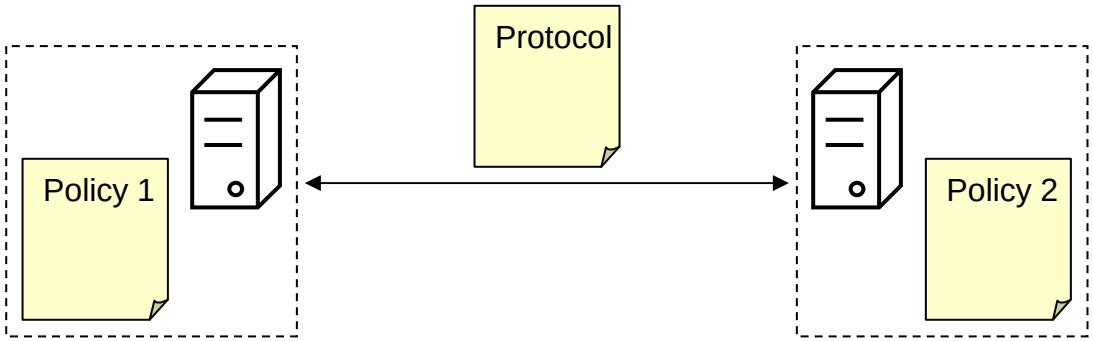


- Forms & natural-language privacy policies are so 20th century
- Want automation for
  - Requesting attribute values
  - Filling out attribute values
  - 3rd party attestation of attributes
  - Interpreting privacy policies

Policy languages

- Need for standardized, machine-interpretable languages to express
  - what information server needs
  - how server will treat this information
  - what information user is willing to give away to whom
  - how user expects her information to be treated

- Policy languages vs. protocol languages



Overview of this talk

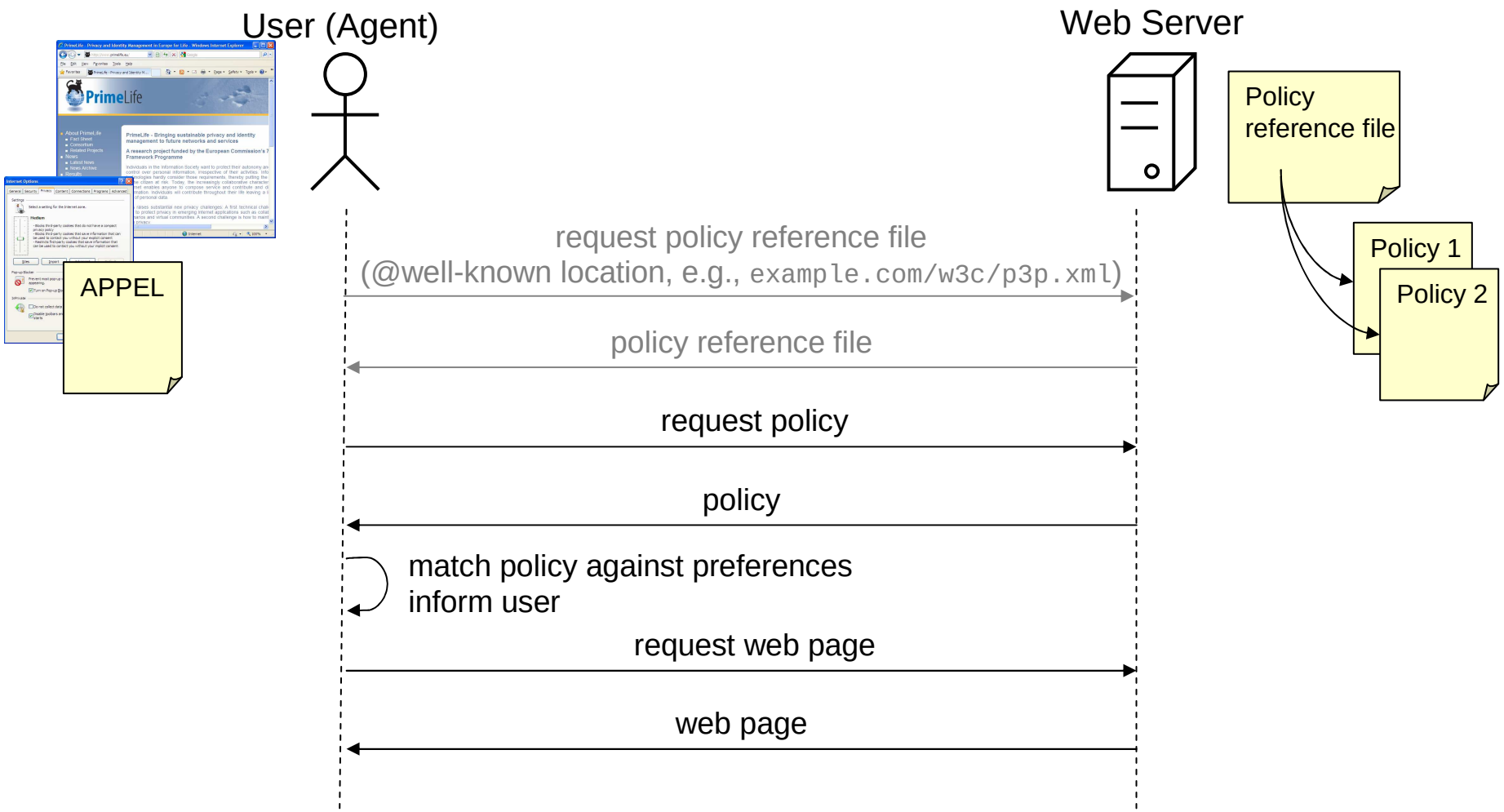
- Privacy policy language: P3P
- Access control policy language: XACML
- Attribute & identity claims protocol language: SAML

- PrimeLife Policy Language (PPL)

P3P

- Platform for Privacy Preferences (P3P)
- Developed by World Wide Web Consortium (W3C)
- Standard protocol language for sites to describe their privacy policies
- Enables development of user agents that
  - match P3P policies against user's preferences
    - Internet Explorer 6: cookies preferences tab
    - AT&T/CMU Privacy Bird: preferences in APPEL language
  - summarize/visualize P3P policies
- Critiques
  - almost exclusively used for cookies management
  - too complex to create/read P3P policy
  - lack of enforcement, consumer/industry/browser support

P3P data flow



P3P policy format

Legal entity & contact info of web server

Ability for users to access their own data

Applicable law, court, and sanctions in case of disputes

Advantages to user of accepting policy

Indicates no identifiable data will be collected

Purposes of data collection and usage

Recipients of collected data

Kind of retention: none, business-practice, indefinite, …

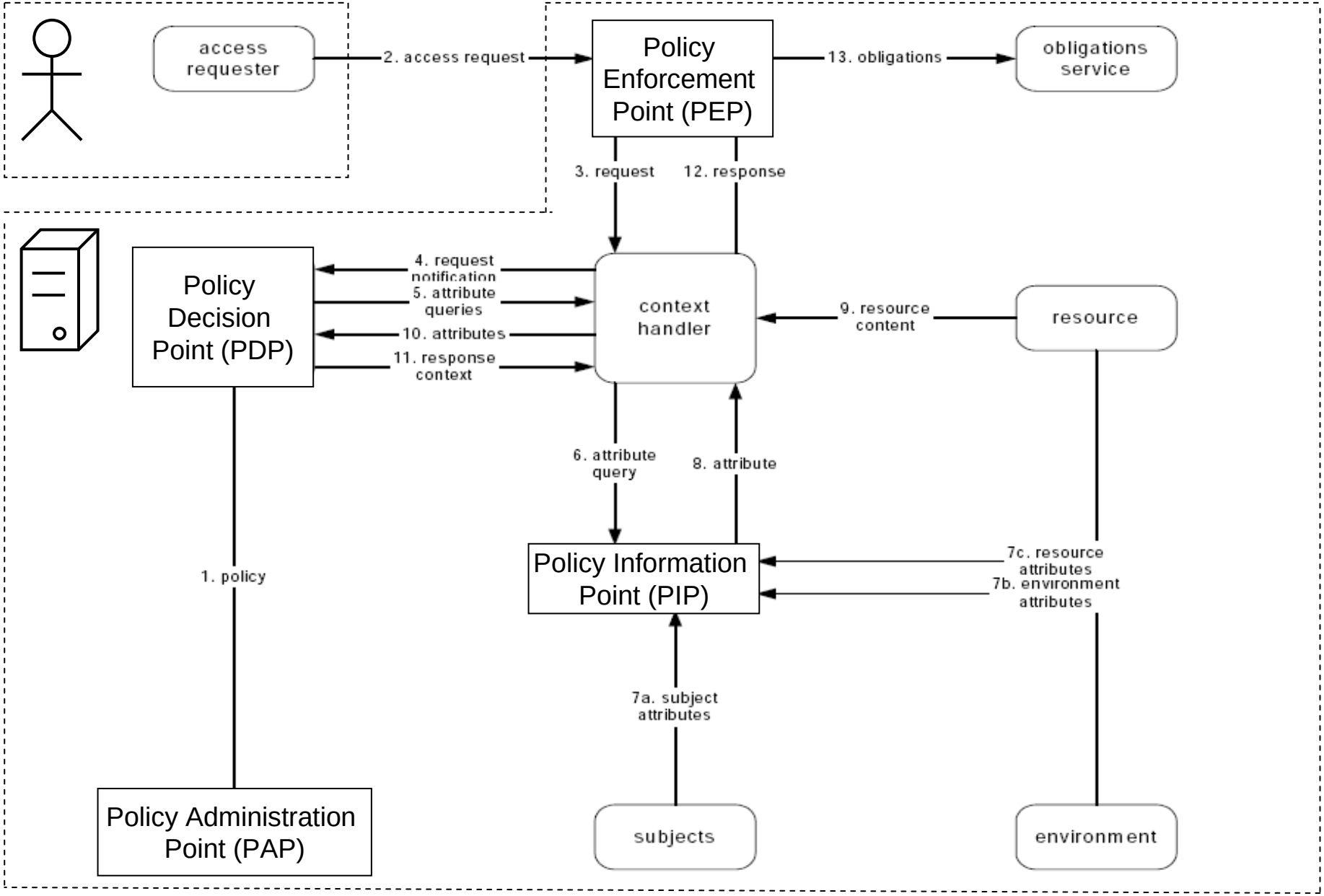
Which (categories of) data will be transferred or inferred

```
<POLICY>
  <ENTITY>
  <ACCESS>
  <DISPUTES-GROUP>
    <DISPUTES>
      <REMEDIES>
  <STATEMENT>
    <CONSEQUENCE>
    <NON-IDENTIFIABLE>
    <PURPOSE>
    <RECIPIENT>
    <RETENTION>
    <DATA-GROUP>
      <DATA>
        <CATEGORIES>
```

XACML

- eXtensible Access Control Markup Language, developed by OASIS
- Industry standard specifying
  - XML-based access control policy language
  - XML-based access request/response protocol language
  - processing model
- Access decisions based on attributes of
  - Subject (e.g., username, role)
  - Protected resource (e.g., file name, URL, content,…)
  - Action (e.g., read, write,…)
  - Environment (e.g., date, time,…)
- Extension points: can define new attributes, data types, functions, obligations, rule/policy combining algorithms

XACML data flow



XACML policy format

Built-in rule/policy combining algorithms:

- Permit overrides
- Deny overrides
- First applicable
- Only one applicable

“Permit” or “Deny”

Restricts the applicability of this rule

Restrictions on subject who performs request

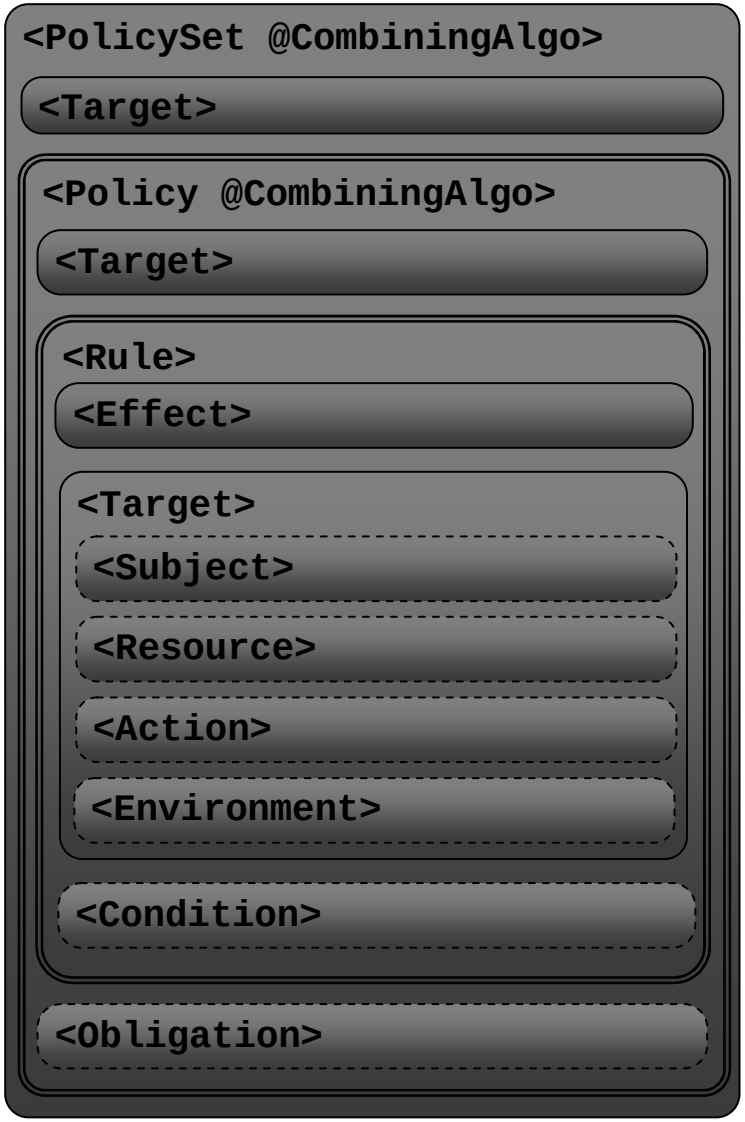
Restrictions on protected resource

Restrictions on requested action

Restrictions on environment attributes

Arbitrary other restrictions (boolean function over atts)

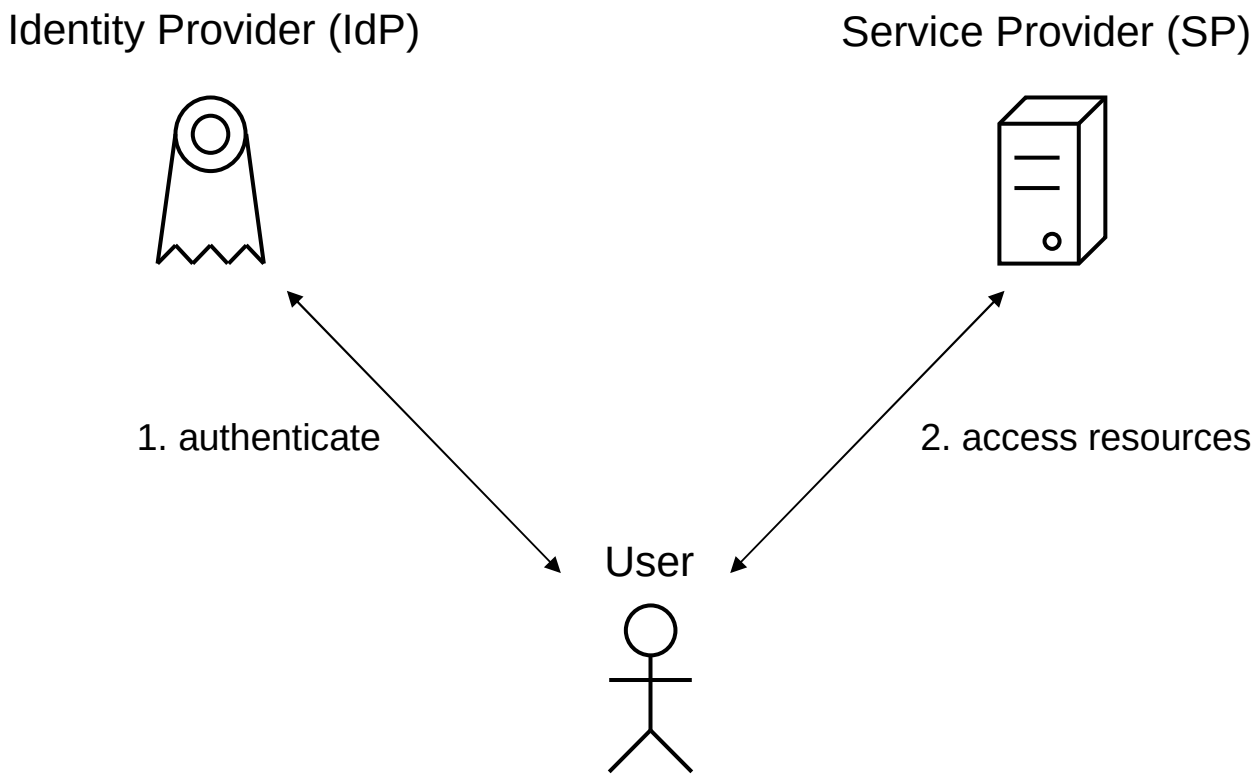
Obligations PEP has to adhere to when access granted



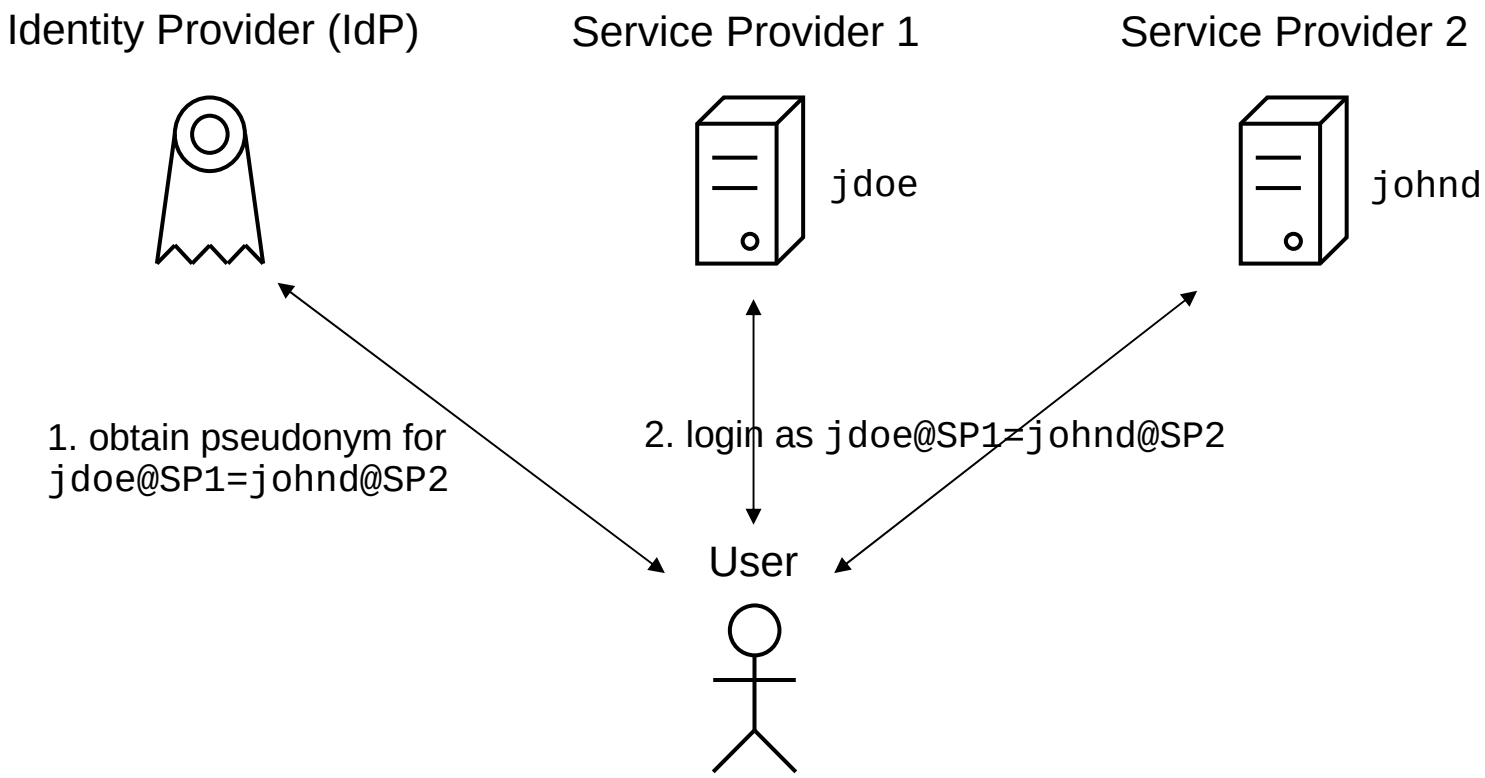
SAML

- Security Assertion Markup Language, by OASIS
- Protocol language for communicating
  - user authentication information
  - user attribute information
  - authorization decision information
- Main use cases
  - Single sign-on (SSO)
  - Identity federation
  - Attribute provision
- SAML profile of XACML
  - Translate SAML attributes into XACML attributes
  - Use SAML to carry XACML policies, authorization decisions,…

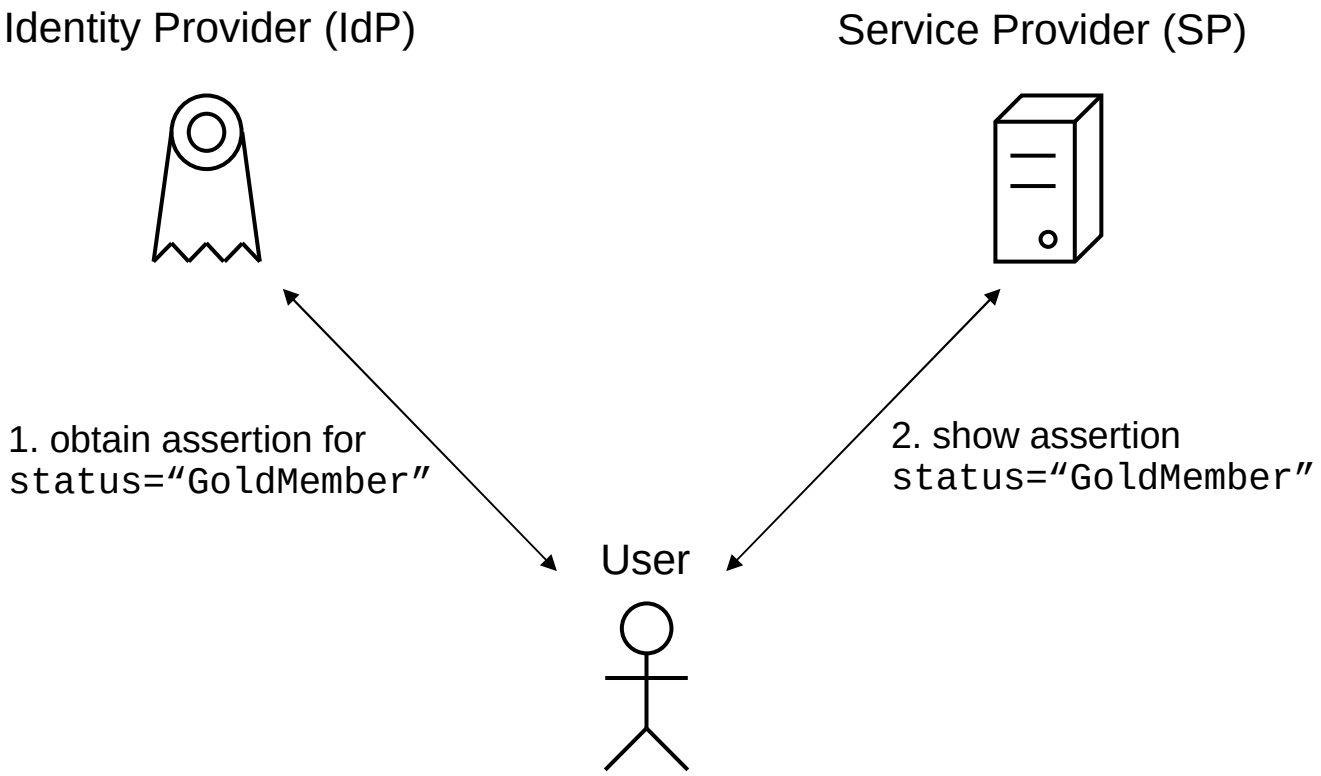
SAML single sign-on



SAML identity federation



SAML attribute provision



SAML assertion format

Assertion with unique identifier and time of issuing

SAML authority making the claim

XML signature of assertion by issuer

Principal that is subject of the assertion

Constraints on use of assertion (time, audience,…)

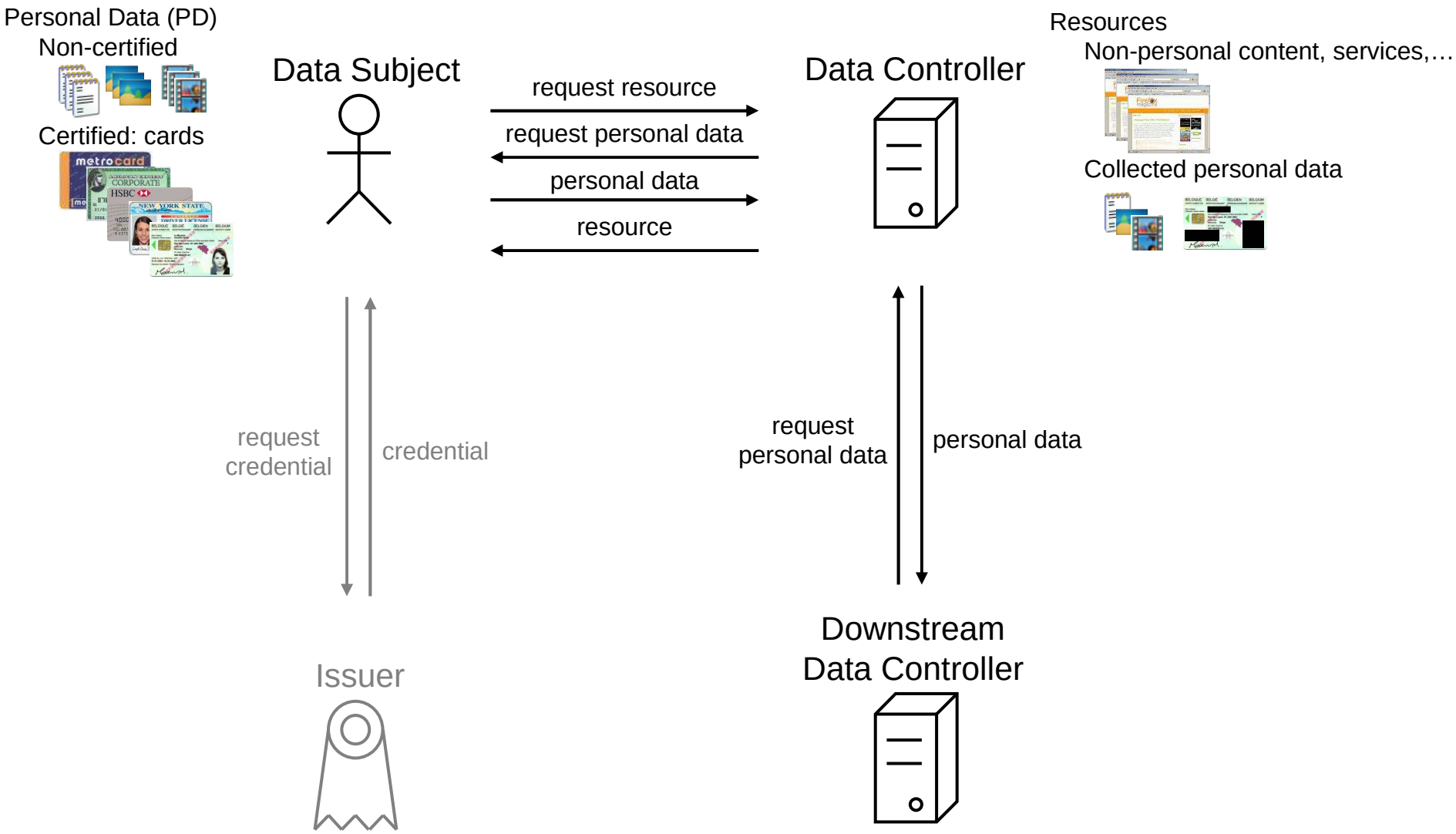
Abstract statement type

Statement that (and how) user authenticated at IdP

Statement that subject has specified attribute values.

```
<Assertion @ID @IssueInstant>
  <Issuer>
  <ds:Signature>
  <Subject>
  <Conditions>
  <Statement>
  <AuthnStatement>
  <AttributeStatement @Name>
    <AttributeValue>
```

PrimeLife Policy Language (PPL) scenario



Main features of PPL

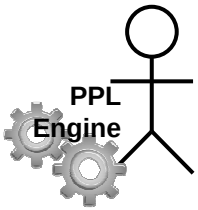
- Card-based access control
  - attributes grouped in cards
  - technology independence
  - privacy friendly
    - reveal attributes vs. prove conditions
    - support anonymous credentials (Identity Mixer, U-Prove)
- Integrated data handling
  - two-sided detailed data handling preferences/policies
  - automated matching procedure
  - extensible vocabularies
  - downstream usage
- Policy sanitization
- Based on existing standards: XACML & SAML

Structure of PPL

Personal Data (PD)
Non-certified

Certified: cards

Data Subject



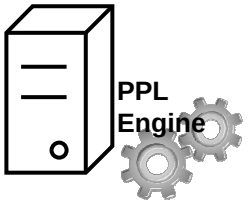
request resource →

← request personal data

personal data →

← resource

Data Controller



Resources
Non-personal content, services,…

Collected personal data

SAML

Specific Policy:
over specific personal data (e.g. birth date)
- **Access control policy (ACP):**
  who can access (e.g. PrivacySeal silver)
- **Data handling preferences (DHPrefs):**
  how is to be treated when revealed
  - **Authorizations** (e.g. marketing purposes, forwarded to PrivacySeal gold)
  - **Obligations** (e.g. delete after ≤2y)

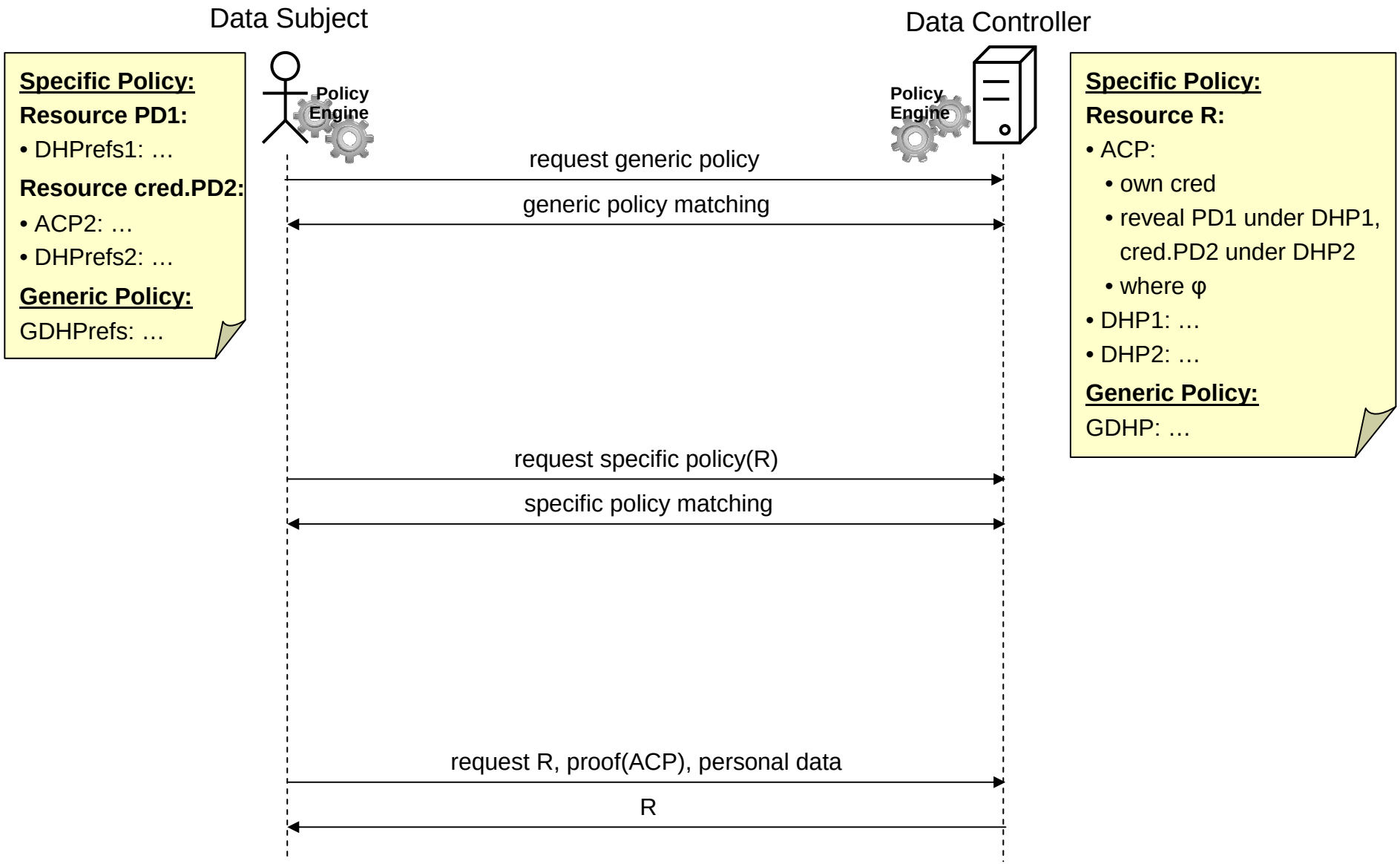
Generic Preferences:
DHPrefs over implicitly revealed personal data (e.g. IP address, cookies,…)
- **Authorizations** (e.g. admin purposes)
- **Obligations** (e.g. delete after ≤2y)

XACML

Specific Policy:
over specific resource (e.g. BuyService)
- **Access control policy (ACP):**
  who can access
  - cards to possess (e.g. ID card)
  - personal data to reveal (e.g. nationality)
  - conditions to satisfy (e.g. age>18)
- **Data handling policy (DHP):**
  how revealed personal data will be treated
  - **Authorizations** (e.g. marketing purposes)
  - **Obligations** (e.g. delete after 1y)

Generic Policy:
DHP over implicitly revealed personal data (e.g. IP address, cookies,…)
- **Authorizations** (e.g. admin purposes)
- **Obligations** (e.g. delete after 1y)

Interaction overview



Card abstraction

- Card contains
  - list of attribute-value pairs
  - *pre-evidence*: technology-specific meta-data to
    - protect attribute integrity
    - prove card ownership
- Card *issuer* vouches for attributes wrt *owner* (identity/authority)
- Hierarchy of card *types:* define attributes contained
- Instantiating technologies: X.509, SAML, CardSpace, OpenID, Kerberos, trusted LDAP, Identity Mixer, U-Prove,…

Card-based access control example

- Policy: requirements on owned cards, e.g.,

  own p::Passport **issued-by** admin.ch, fgov.be, governo.it

  own c::Creditcard **issued-by** visa.com, amex.com

  reveal c.number, c.expdate

  where p.name = c.name ^ p.bdate < today-18Y

  ^ c.expdate > today ^ p.expdate > today+1M

- Authentication = *claim* over owned cards + *evidence*, e.g.,

  own p::Passport **issued-by** admin.ch

  own c::Creditcard **issued-by** visa.com

  reveal c.number = “1234567890”

  reveal c.expdate = “31/12/2012”

  where p.name = c.name ^ p.bdate < 05/05/1992 ^ p.expdate > 05/06/2010

Features of card-based access control

Access control requirements language supporting

- Privacy preservation
  - for user: *minimal* claim to be disclosed

    (selectively) reveal attribute ↔ predicate satisfied
  - for server: “sanitize” sensitive policies
- Multi-card claims

  but prevent “card mixing” through reference pointer to individual cards
- Technology independence

  but supporting advanced features, esp. anonymous credentials

Authorizations and obligations

- General principle: provide
  - wrapper for user-extensible vocabularies
  - basic pre-defined vocabulary
- Authorizations
  - “use for purpose”
    - user-extensible ontology of purposes,
    - basic pre-defined ontology available
  - “forward under policy” = downstream access control
- Obligations
  - general structure: **do** action **when** trigger (**from** start **to** end)
  - pre-defined actions:
    - delete data
    - anonymize data
    - notify data subject
    - write to (secure) log
  - pre-defined triggers:
    - at time, periodic
    - data access, data deletion
    - data loss, obligation violation
    - aliens landing on earth

Obligation & authorization matching

automated matching of **any** two data handling preferences/policies via **"less permissive than"** relation (≤) defined on

- authorizations, e.g.

  use for {delivery} ≤ use for {delivery,marketing}

- triggers, e.g.

  trigger at 2010/01/01 ≤ trigger at 2010/12/31

- actions, e.g.

  delete firstname, lastname ≤ delete firstname

- obligations

  action, trigger, validity

  $$o_1=(a_1,t_1,v_1) \leq o_2=(a_2,t_2,v_2) \Leftrightarrow (a_1 \leq a_2) \wedge (t_1 \leq t_2) \wedge (v_1 \leq v_2)$$

- sets of authorizations and obligations

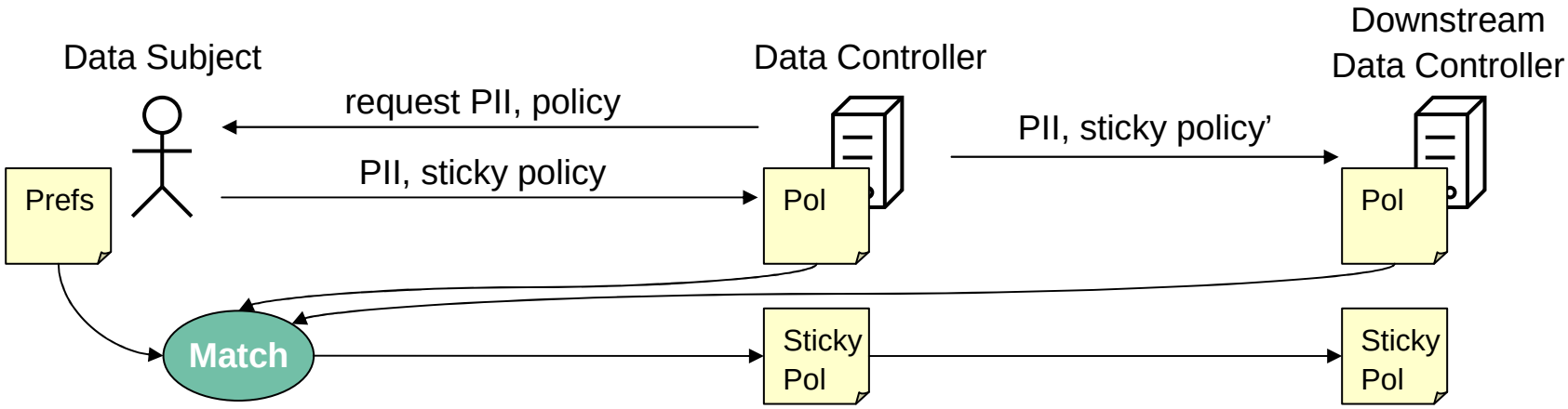
  $$O_1 \leq O_2 \Leftrightarrow \forall o_1 \in O_1 \; \exists o_2 \in O_2 : o_1 \leq o_2$$

- data handling policies

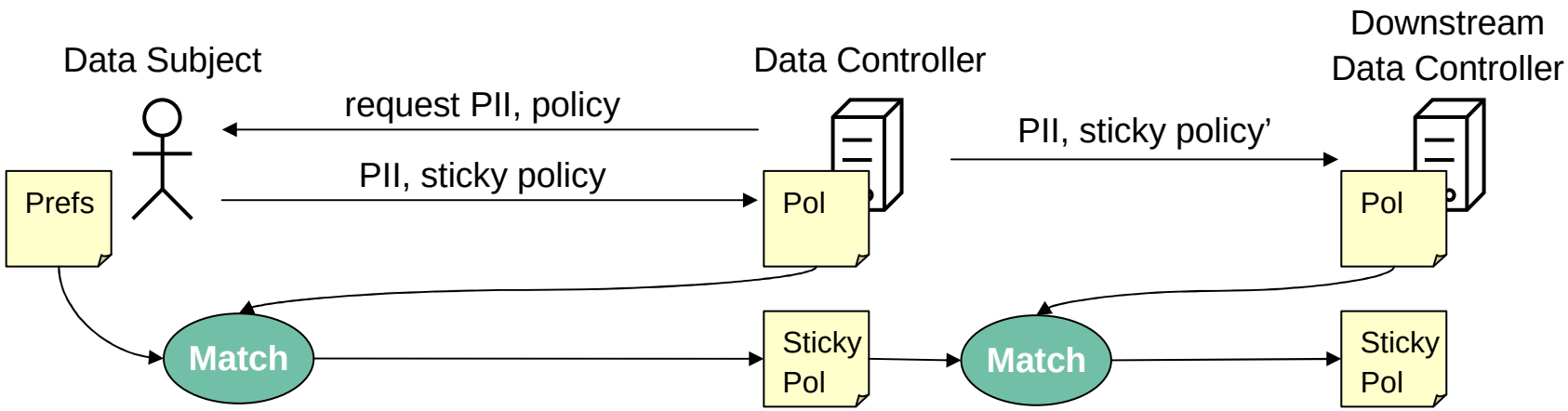
  $$P_1 = (A_1,O_1) \leq P_2 = (A_2,O_2) \Leftrightarrow A_1 \leq A_2 \wedge O_1 \leq O_2$$

Downstream usage matching

- Proactive matching



- Lazy matching



Obligation matching prototype

Privacy Policy Demo

View Load M Grammars FORMULA

Alice Service1 Service2

Matching Privacy

Personal Data

PII Id	PII Type	PII Value
0	email	Alice@Contoso.co
1	corporateE	Alice@Adatum.cor
2	phone	+49 12 34 56 78 9
3	phone	+33 12 34 56 78 9
4	xRay	X-Rax: jdsgvjksahc

Privacy Preferences

DSL XML Object Model

```
Plain English Preferences:

if CAx says <Service> is a
BookingService
{
  email:
    <Service> must delete it within
365 days
    <Service> can use it for
{ "statistics", "contact" }
    <Service> can send it
downstream

  phone:
    <Service> must delete it within
30 days
    <Service> can use it for
{ "emergency", "cancellation"}
}
```

Privacy Policy

DSL XML Object Model

```
BookingService
//CAy says <Service> is a Hospital

a1 : email:
        I will delete this PII within 180
days
        // I will notify <User> by email
at <Provided email> when it is used
for "primaryService"
        I may use this PII for
{"statistics", "contact"}
        I may send it to Service2

a2 : phone:
        I will delete this Pii within 30
days
        // I will log when it is used for
"contact"
        I may use this PII for
"cancellation"// or "contact"
```

Privacy Policy

DSL XML Object Model

```
Plain English Policy:

CAx says <Service> is a
BookingService
//CAy says <Service> is a Hospital

a1 : email:
        I will delete this PII within 180
days
        // I will notify <User> by email
at <Provided email> when it is used
for "primaryService"
        I may use this PII for
{"statistics", "contact"}
        // Can I send it to services
certified as "TrustedStatisticsService"
by "TTP" ?

a2 : phone:
        I will delete this Pii within 30
```

Basic | User Perspective (Privacy Matching) | Service Perspective (Privacy Enforcement) | Auditor Perspective (Privacy Checking)

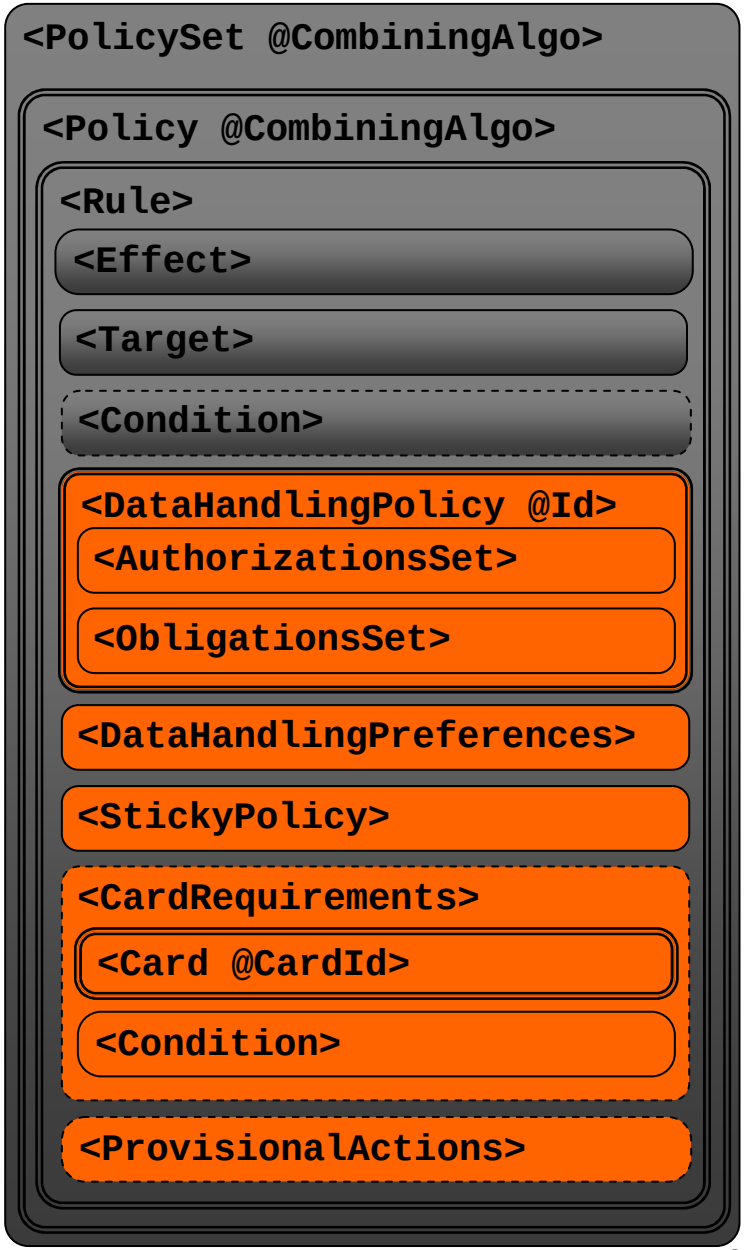
```
OK: Privacy Preferences do match policy (Pref ≥ Pol)

Loading file:///D:/SDs/SD/main/projects/WorkflowSecurity/SecPALforPrivacy/code/ObligationMatching/Matching/data/withDs/modelTest.4ml

Loading file:///D:/SDs/SD/main/projects/WorkflowSecurity/SecPALforPrivacy/code/ObligationMatching/Matching/data/withDs/domainTest.4ml

-- Evaluating M ...
  conforms = True
```

PPL policy format



Proposed data handling policies for revealed attributes

Requested authorizations

Promised obligations

Preferences how target resource should be treated

Agreed-upon sticky policy for target resource

Card-based access control for target resource

Cards to be presented

Required condition over card attributes

Actions to be performed, e.g., reveal attribute under referenced DHP, sign statement, limited spending,…

PPL claims format

One assertion per card, plus cross-card assertion

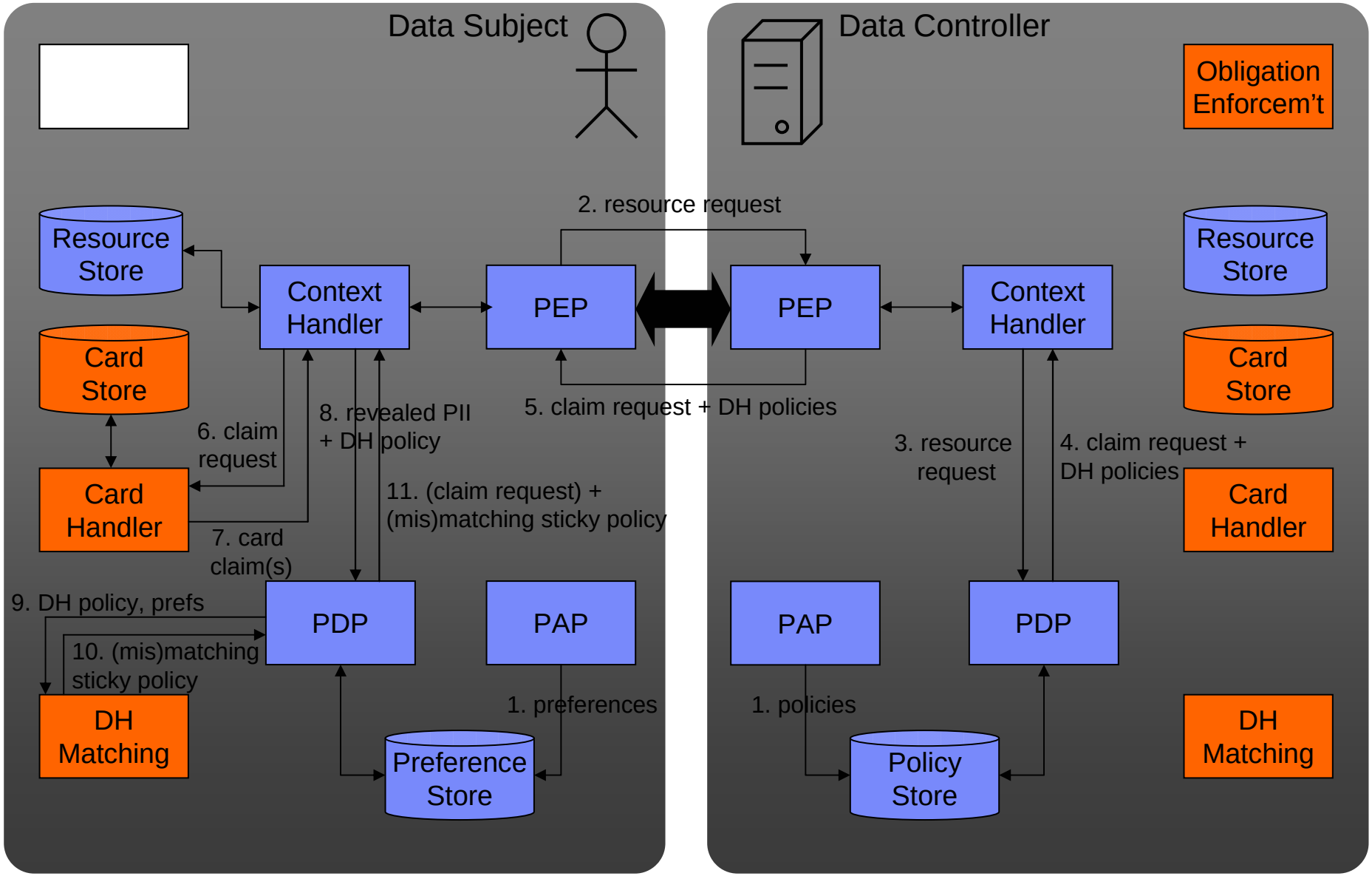
Reference to sticky policy associated to attribute value

New statement type to carry sticky policies

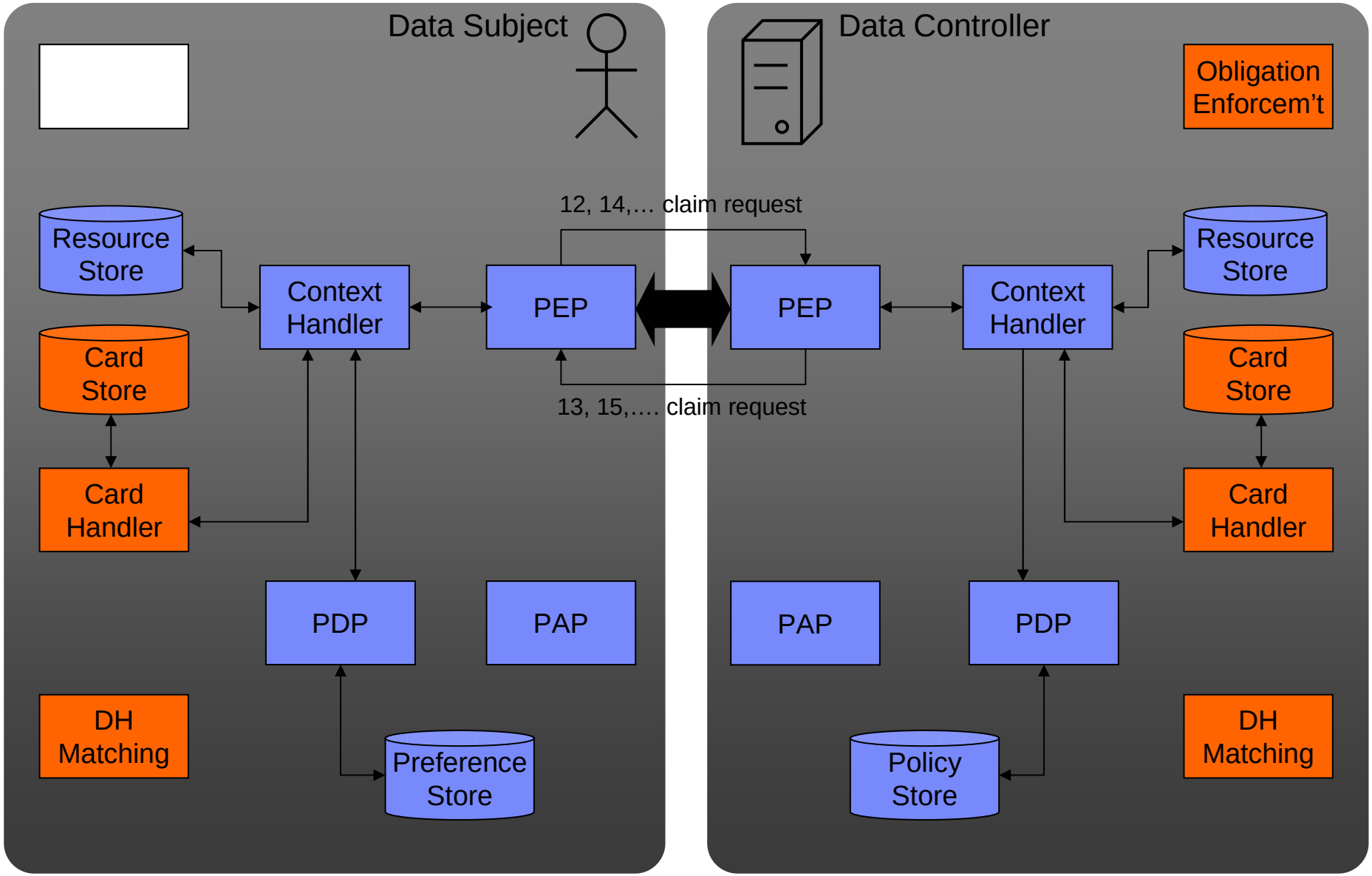
New statement type to carry conditions over attributes

New statement type to carry other (non-XML-signature) types of card evidence

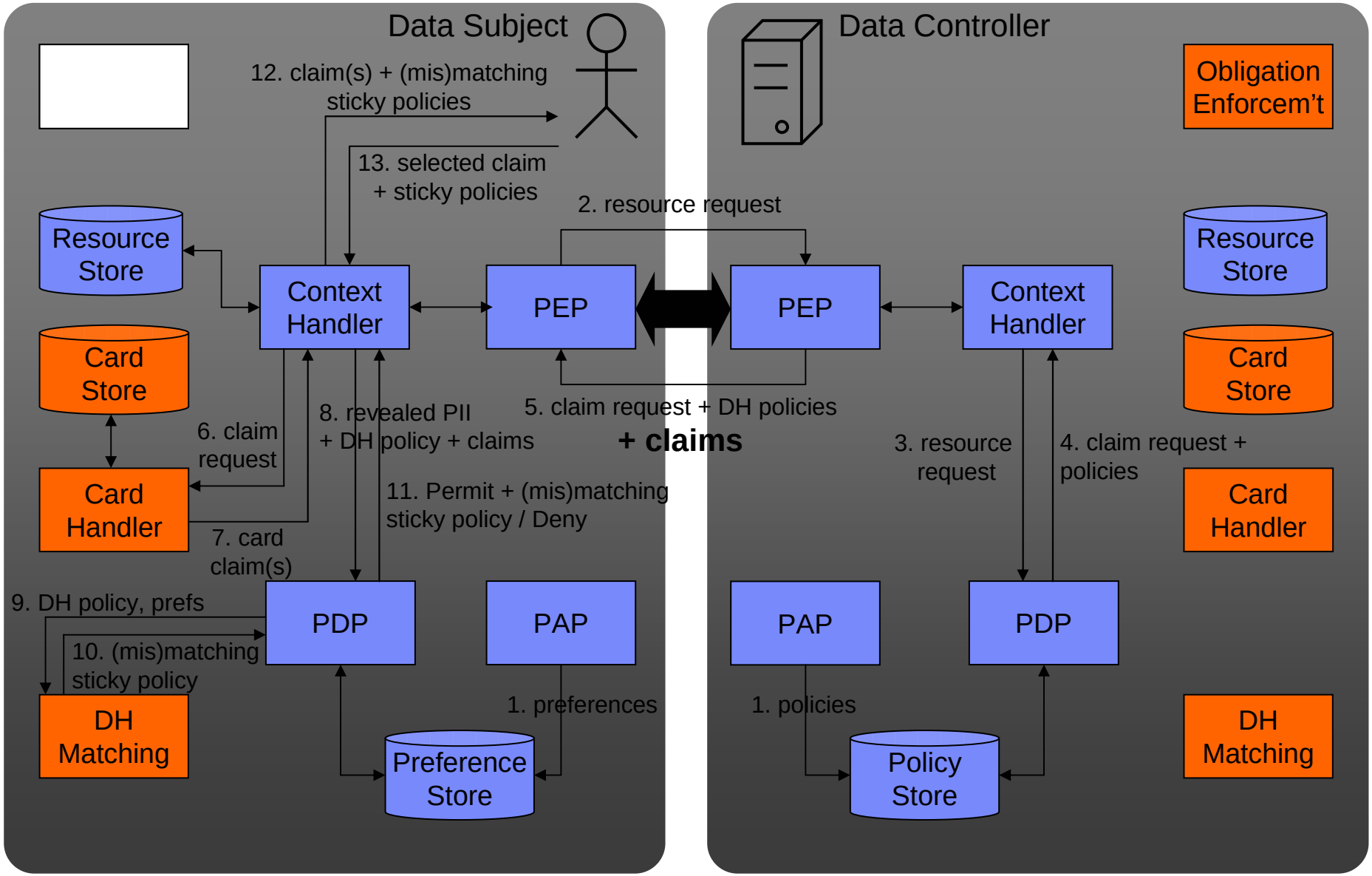
PPL data flow



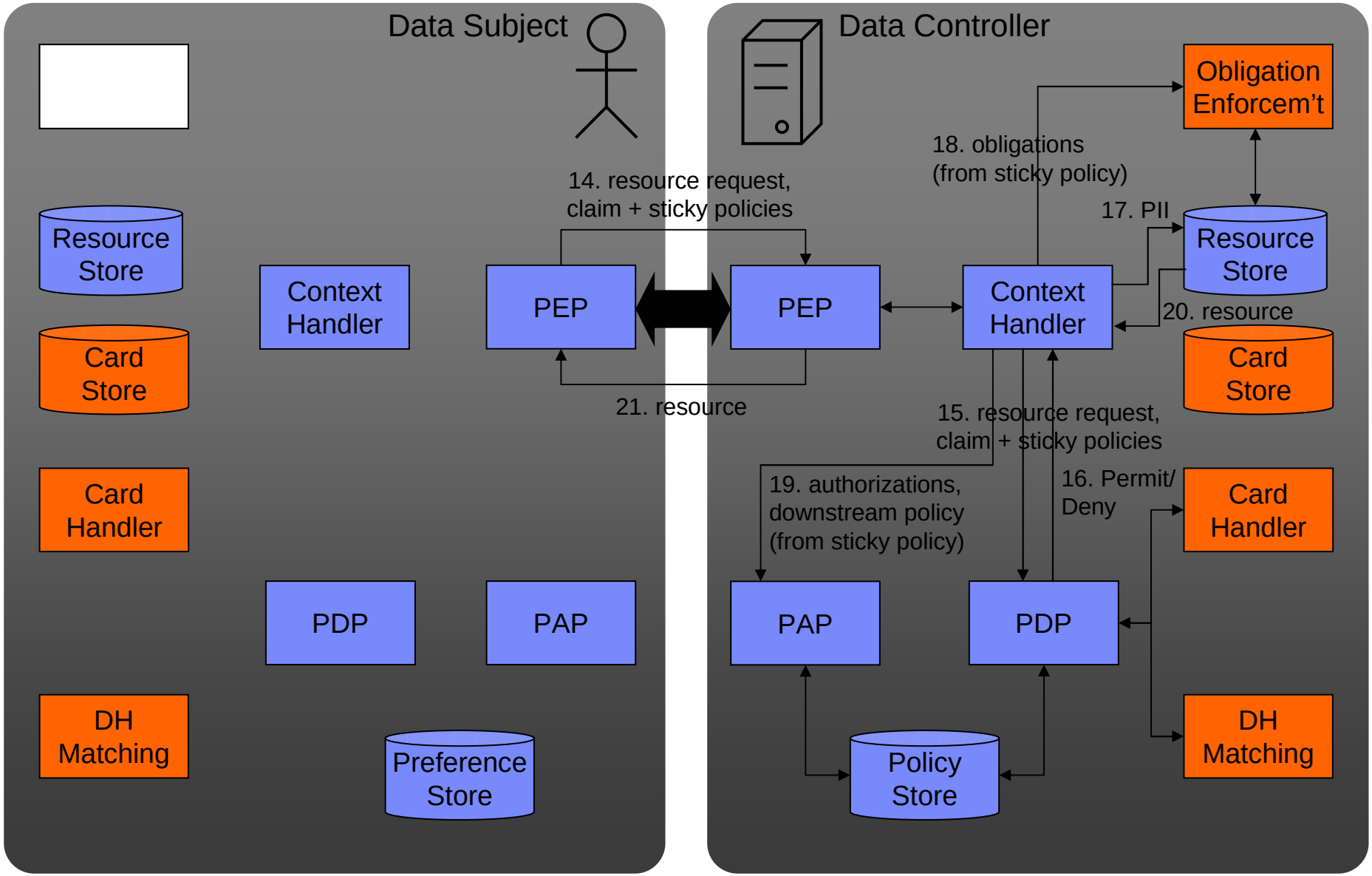
PPL data flow



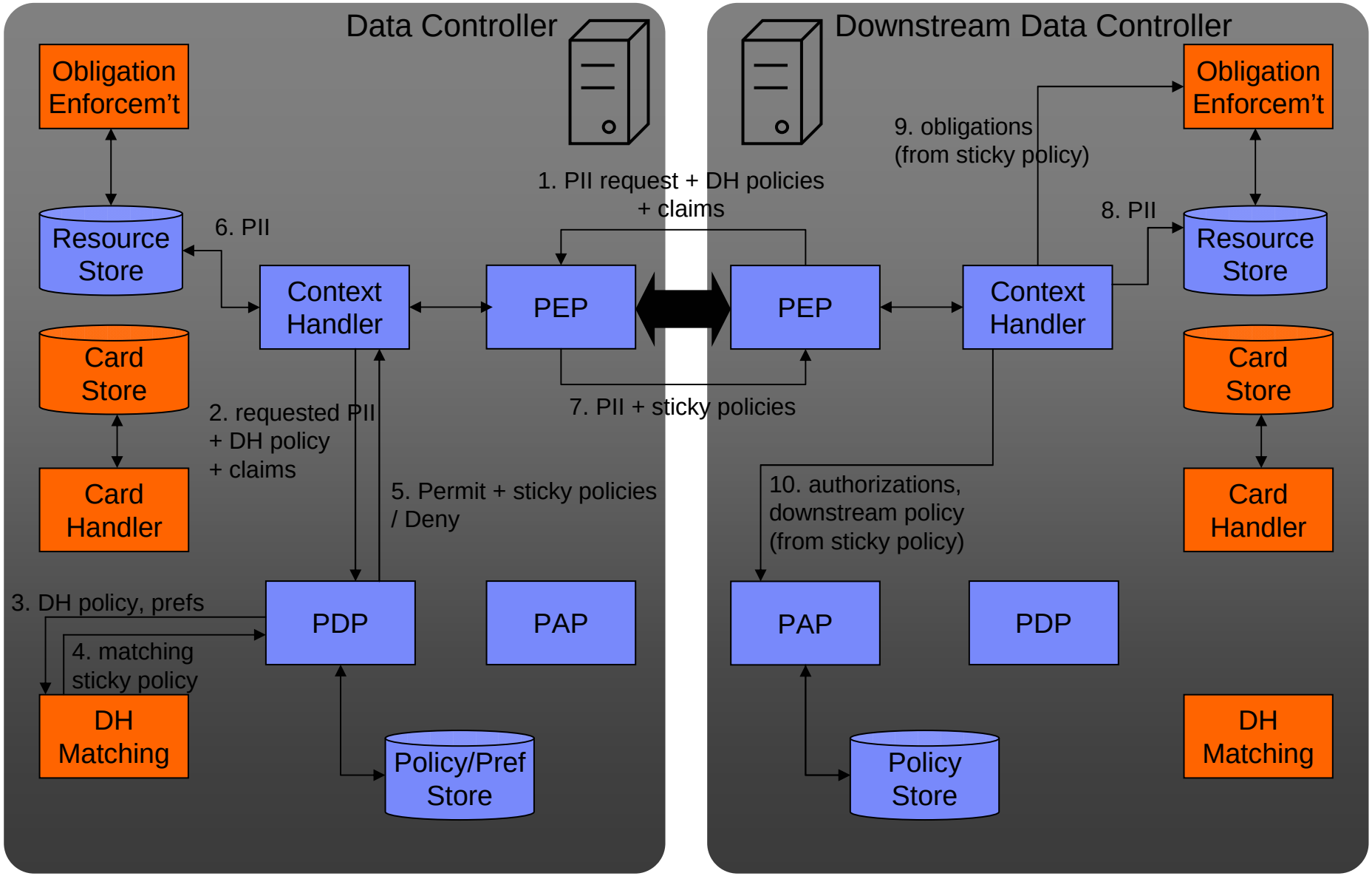
PPL data flow



PPL data flow



PPL data flow (downstream)



Conclusion: main features of PPL

- Card-based access control
  - attributes grouped in cards
  - technology independence
  - privacy friendly
    - reveal attributes vs. prove conditions
    - support anonymous credentials (Identity Mixer, U-Prove)
- Integrated data handling
  - two-sided data handling preferences/policies
  - automated matching procedure
  - extensible vocabularies
  - downstream usage
- Policy sanitization
- Based on existing standards: XACML & SAML

W3C Workshop on Privacy and data usage control

- http://www.w3.org/2010/policy-ws/
- October 4-5, Boston
- Position paper deadline: September 10