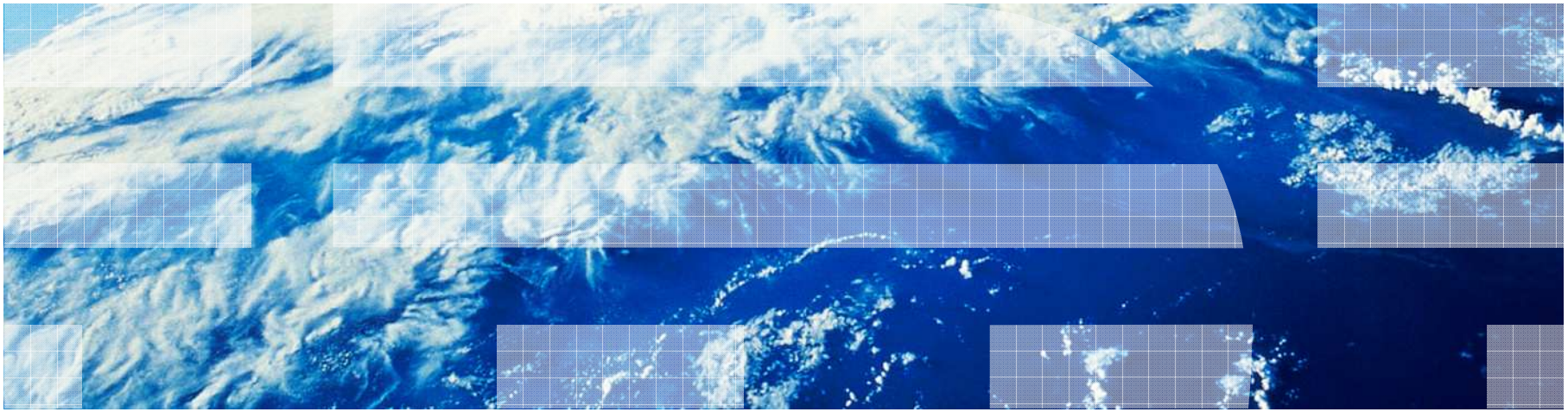
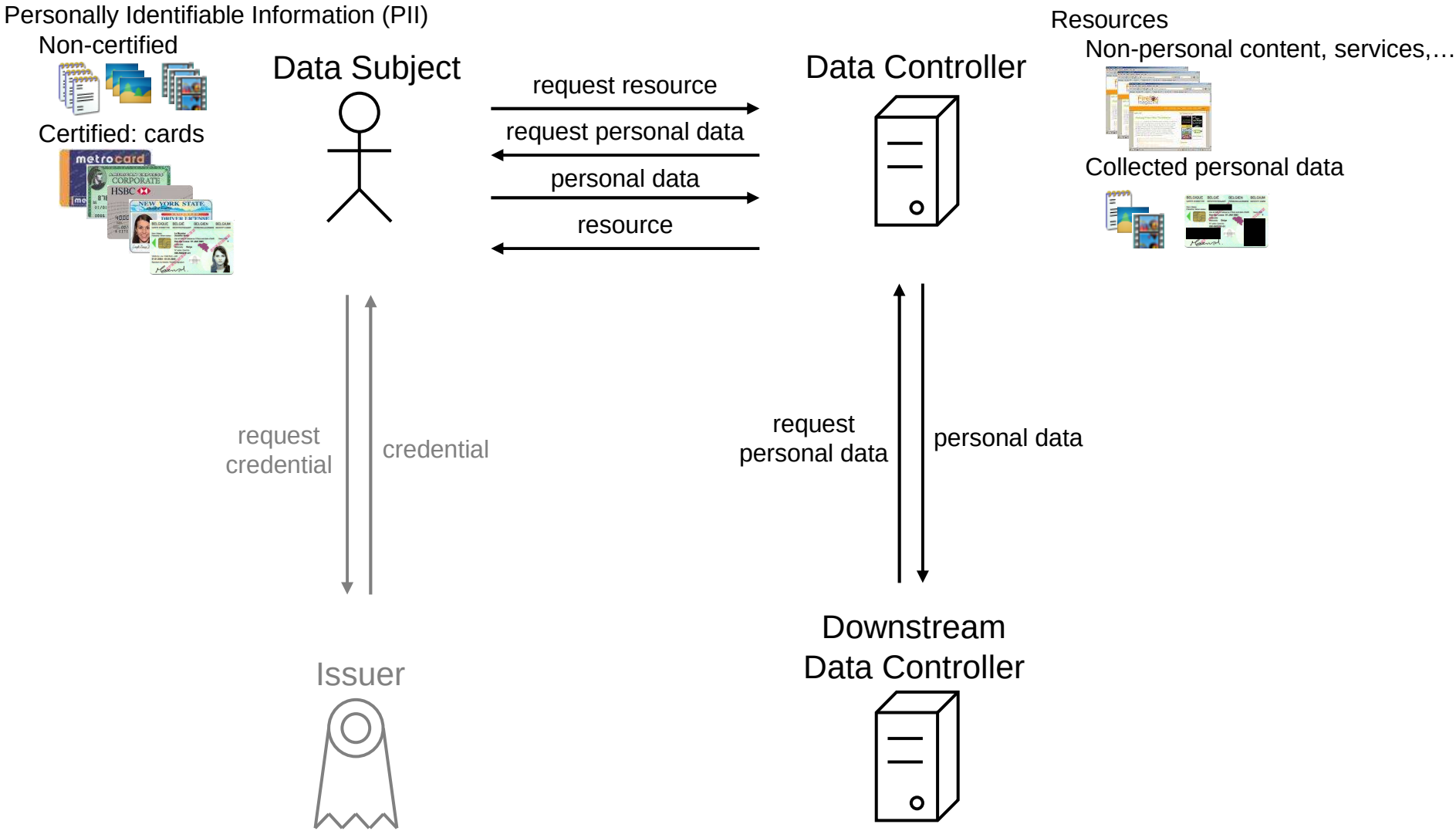
Gregory Neven, IBM Research – Zurich
joint work with Slim Trabelsi (SAP), Akram Njeh (SAP), Laurent Bussard (EMIC)

W3C Workshop on Privacy and Data Usage Control
October 4-5, 2010, Cambridge, MA, USA

The PrimeLife Policy Engine



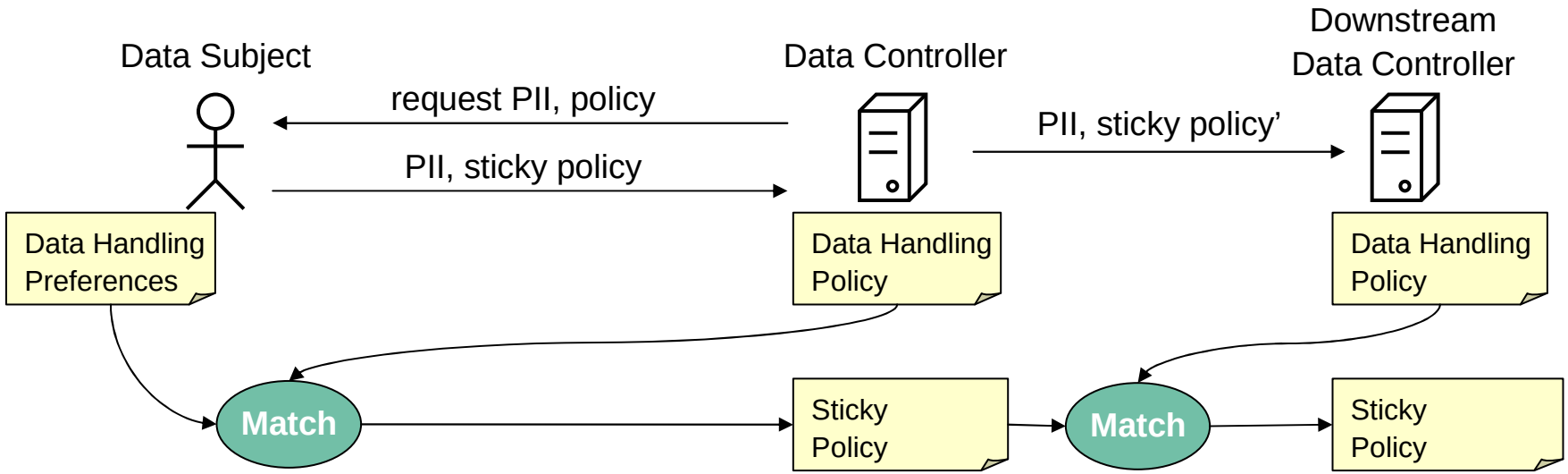
PrimeLife Policy Language (PPL) scenario



Main features of PPL

- Symmetrical architecture: (almost) same language/engine for DS and DC
- Privacy-friendly card-based access control
  - 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

Two-sided data handling “negotation”



Who imposes data handling policy onto whom?

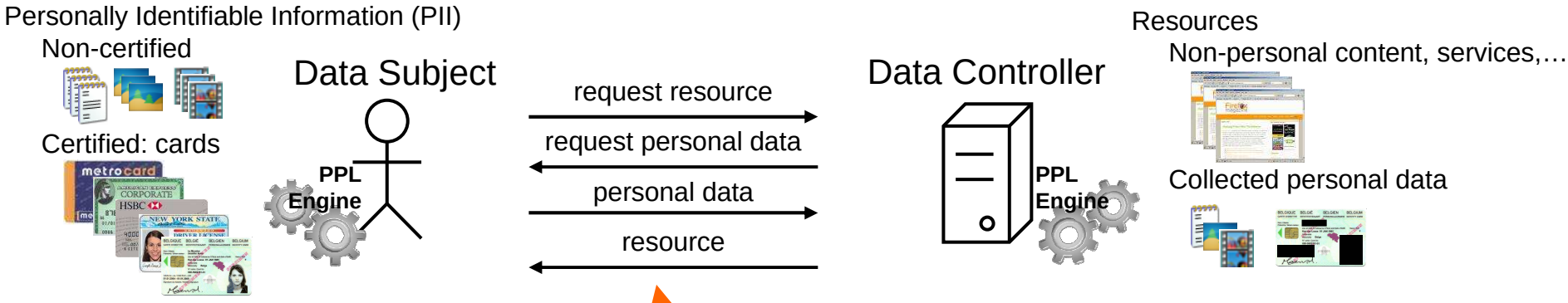
Two unilateral extremes:

- DC onto DS: take it or leave it → DS doesn’t even check policy
- DS onto DC: DC may lack incentive or infrastructure, preferences may uniquely identify DS

Tradeoff in PPL:

- DC proposes DHPolicy
- DS automatically checks whether matches DHPreferences, can take informed decision to overrule in case of mismatch

Symmetrical structure of PPL



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)

SAML

XACML

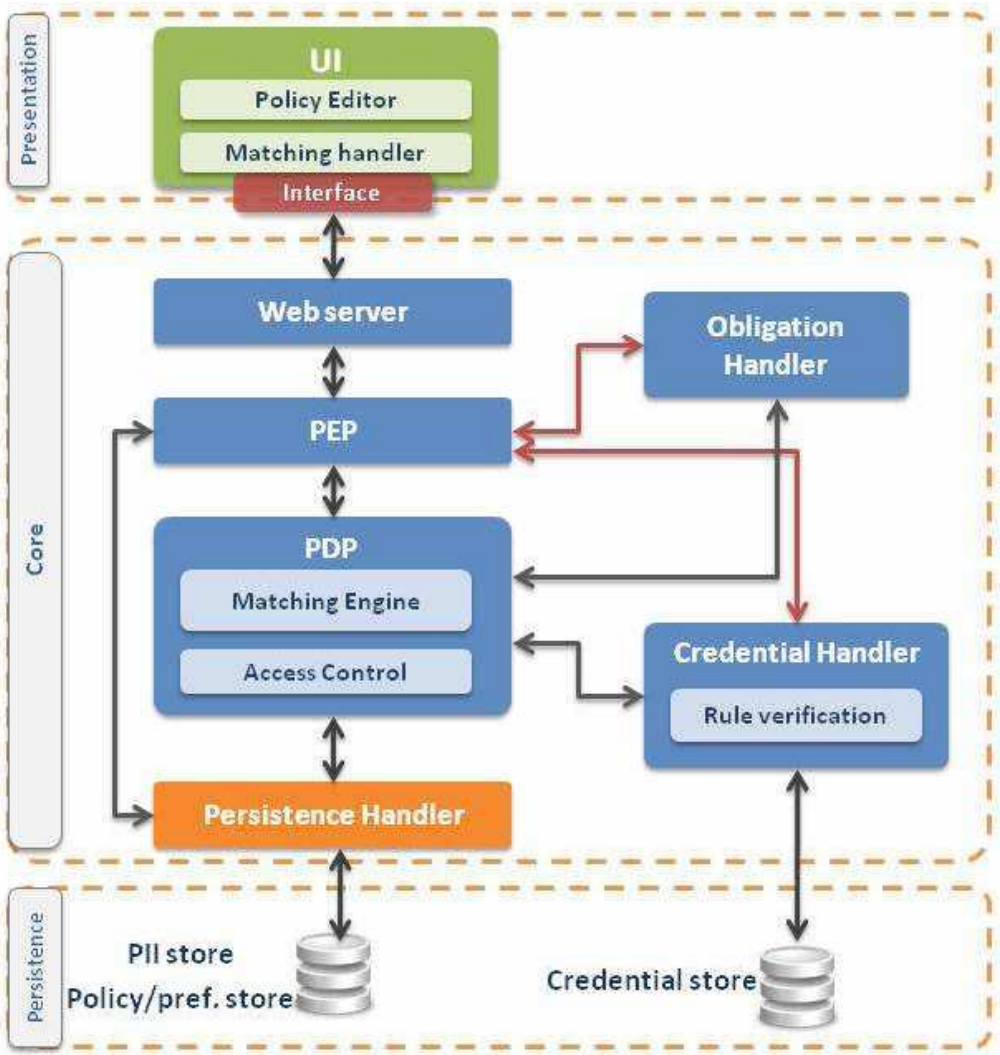
Specific Policy:
over specific resource (e.g. BuyService)
- **Access control policy (ACP):**
  who can access
  - cards to possess (e.g. ID card)
  - personal data to reveal (e.g. nationality)
  - conditions to satisfy (e.g. age>18)
- **Data handling policy (DHP):**
  how revealed personal data will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

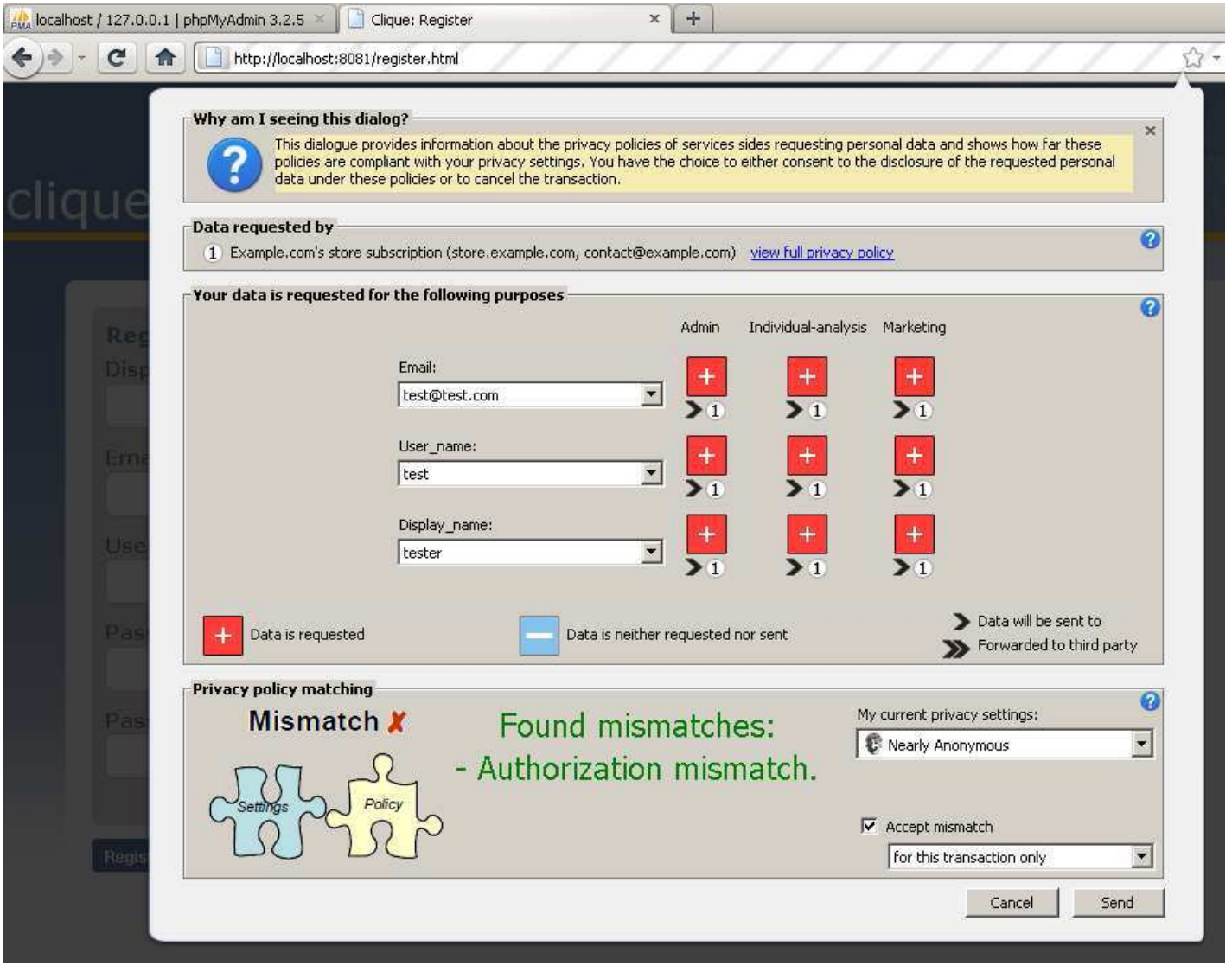
Authorizations and obligations

- 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

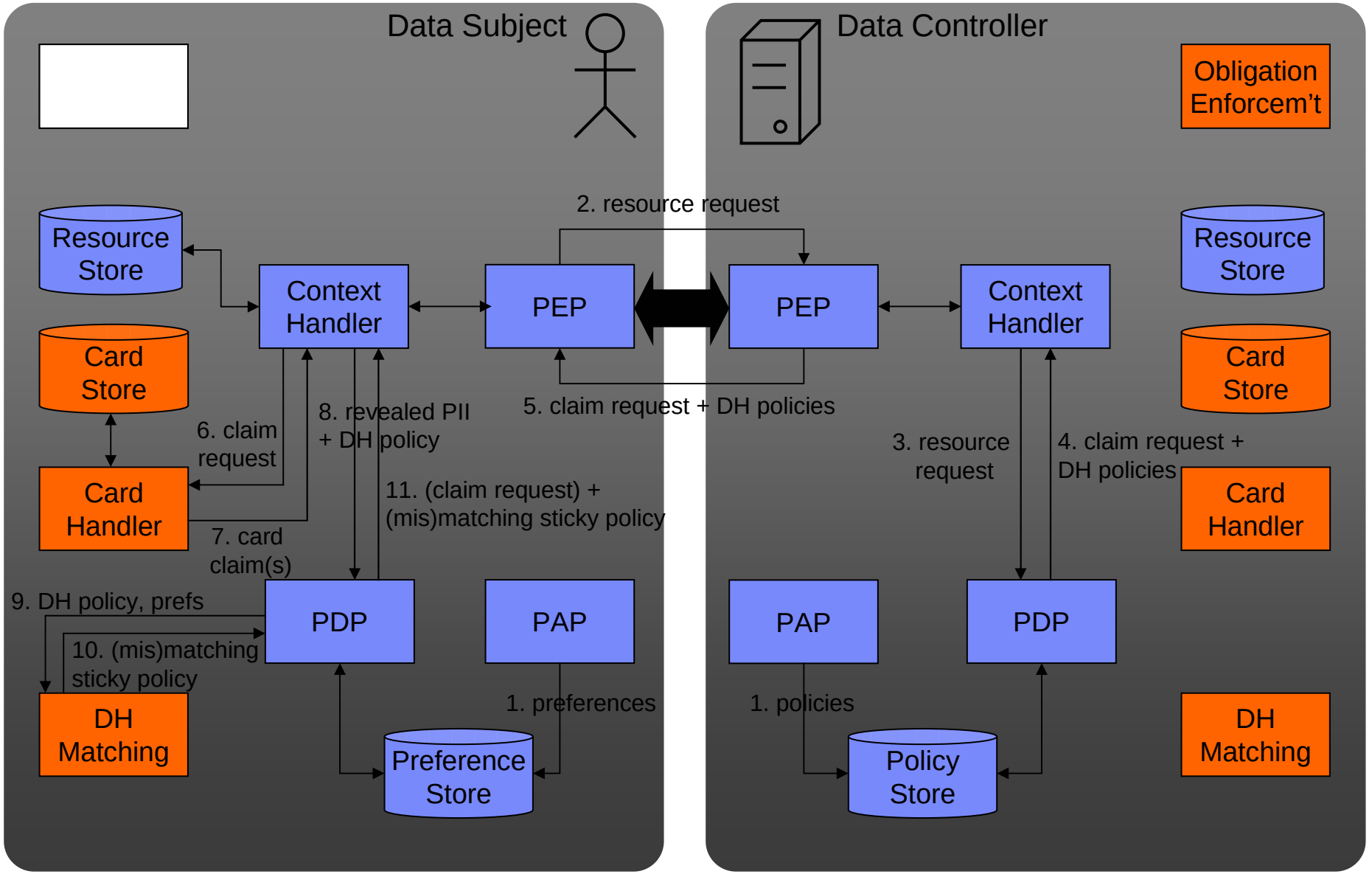
PPL layered architecture



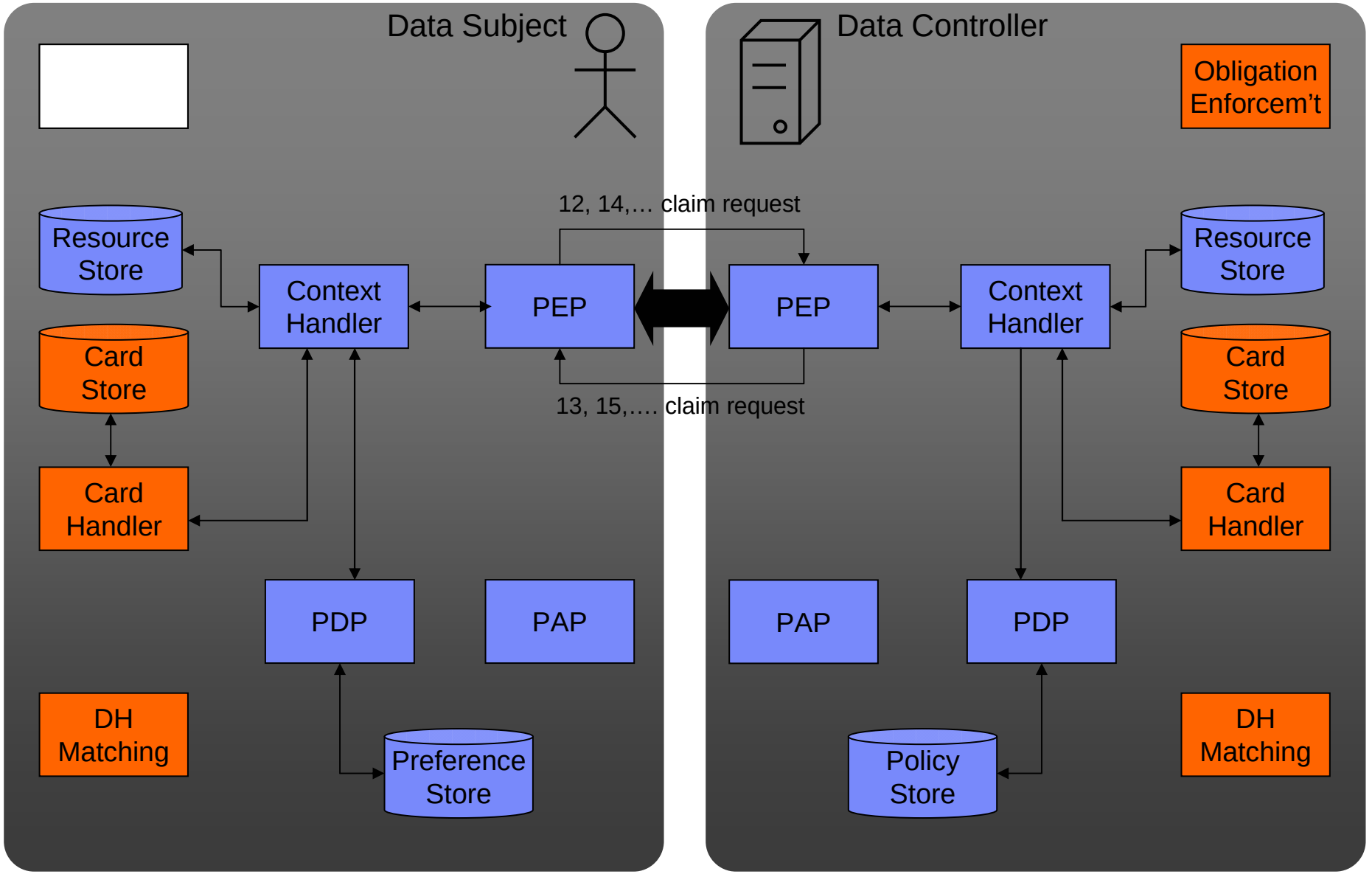
Send data dialog



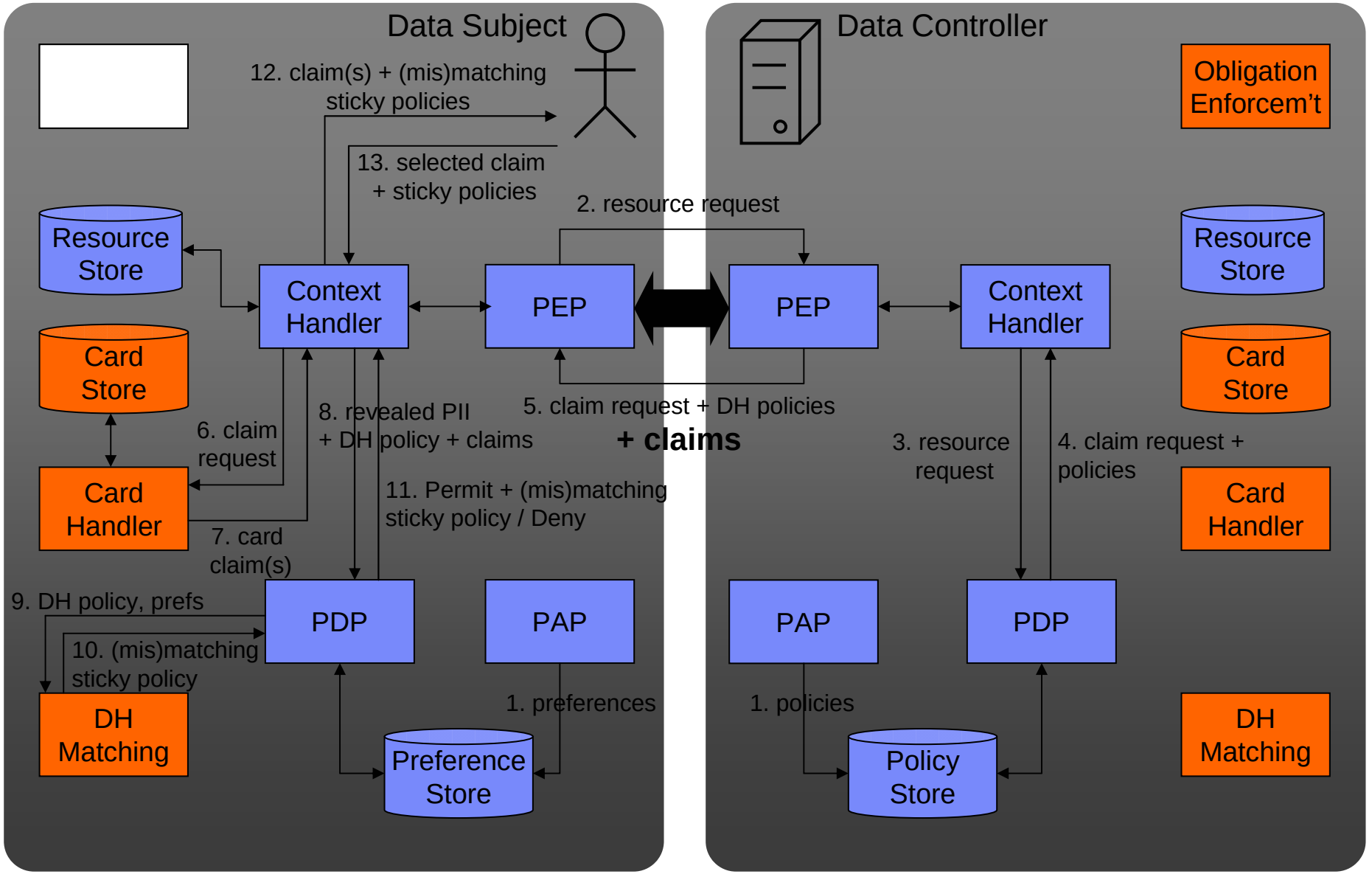
PPL symmetrical architecture & data flow



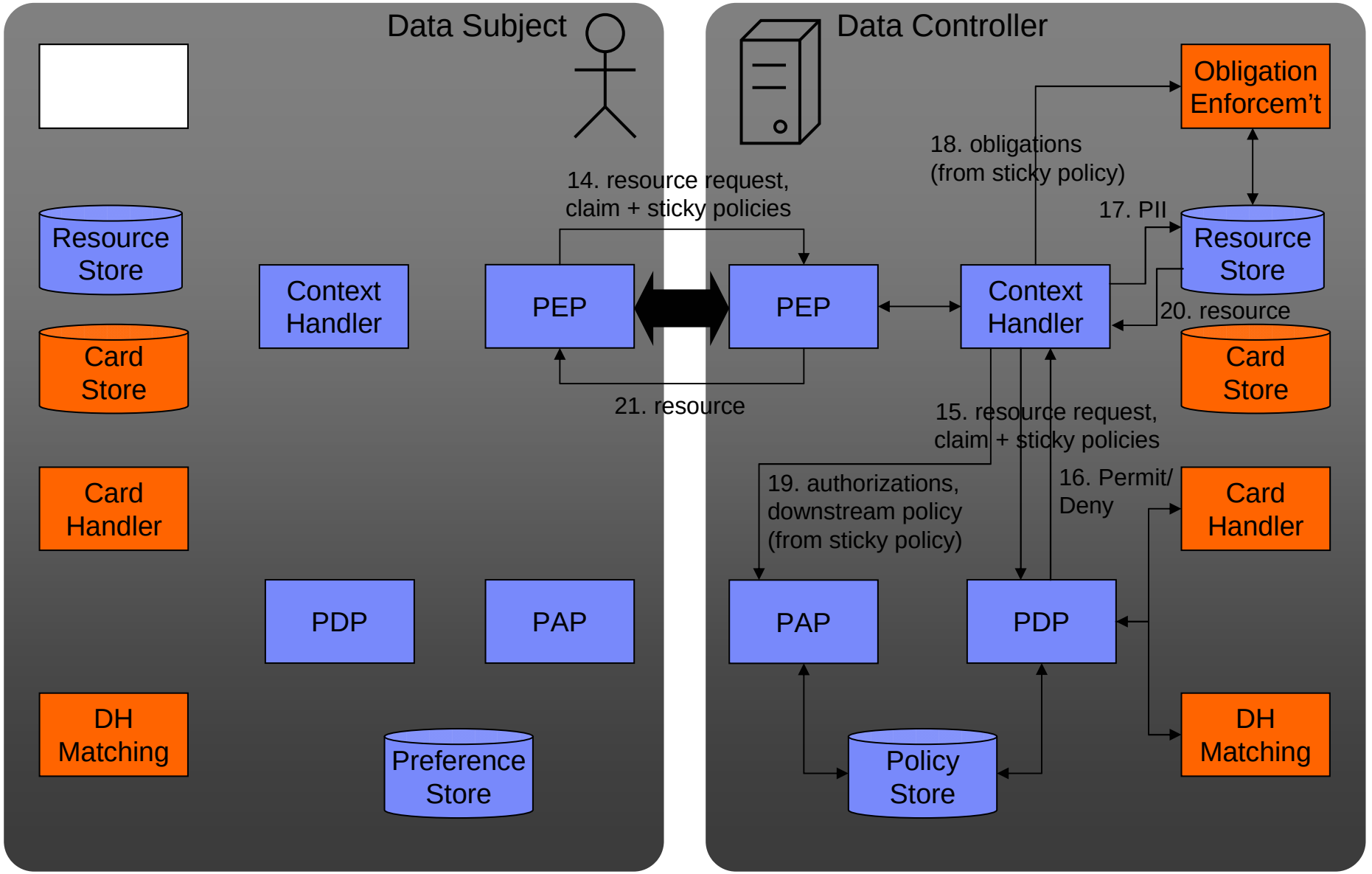
PPL symmetrical architecture & data flow



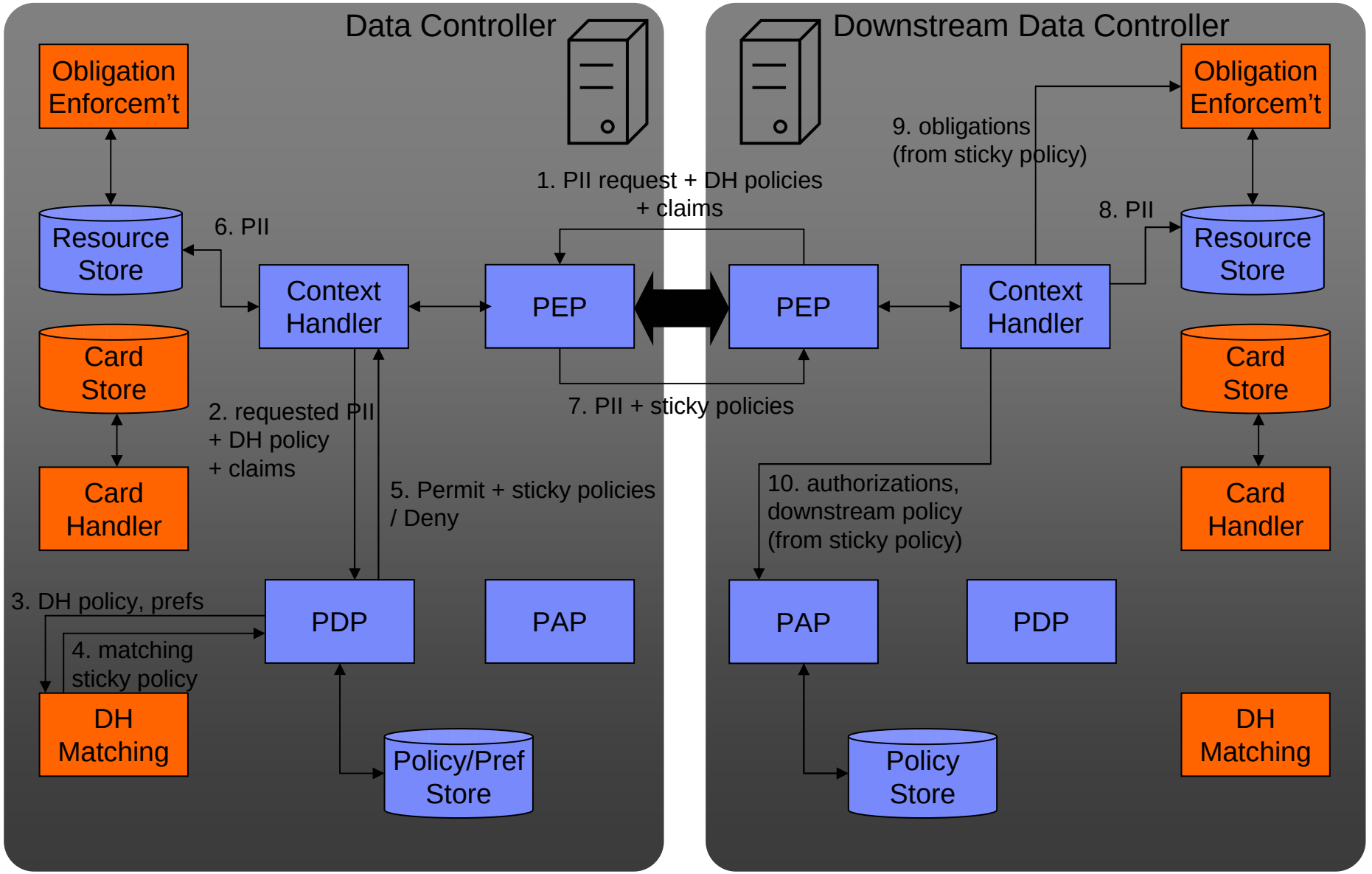
PPL symmetrical architecture & data flow



PPL symmetrical architecture & data flow



PPL downstream data flow



Conclusion: main features of PPL

- Symmetrical architecture: (almost) same language/engine for DS and DC
- Privacy-friendly card-based access control
  - 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