

TAS³ Trusted Architecture for Secure Shared Services (with Privacy)

Sampo Kellomäki (sampo@zxidp.org)

29. September, 2010, ICT, Brussels

05



TAS³ Intro

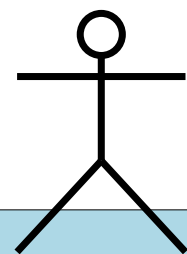
- Visit TAS³ booth in hall H (near Prime Life booth)
- Project runs until end of 2011
- Architecture
 - Identity Management, Authorization, and Audit plumbing
 - Holistic combination of existing technologies
- Protocol Profiles (SAML2, ID-WSF, XACML, ...)
- Reference Implementation in open source
 - zxid.org (Apache2 non-viral license)
- Vision of empowering users and building trust networks
 - Internet of Subjects Foundation
 - Competitive Services Market Place
 - Delegation
 - Trust scoring and trust building

Empowering user to take control of his data

- Fully Pair-wise pseudonymous design
 - Prevent correlation and collusion
- Model where user gives his data from his Personal Data Store
 - User well positioned to impose policies when releasing data
 - Only store data once, and in place that user chooses
- Personas, partial identities
- User self audit dashboard gives user visibility to use of his data
 - Independent means, to keep the service providers in check
- Digitally signed audit trail to ensure legal enforceability

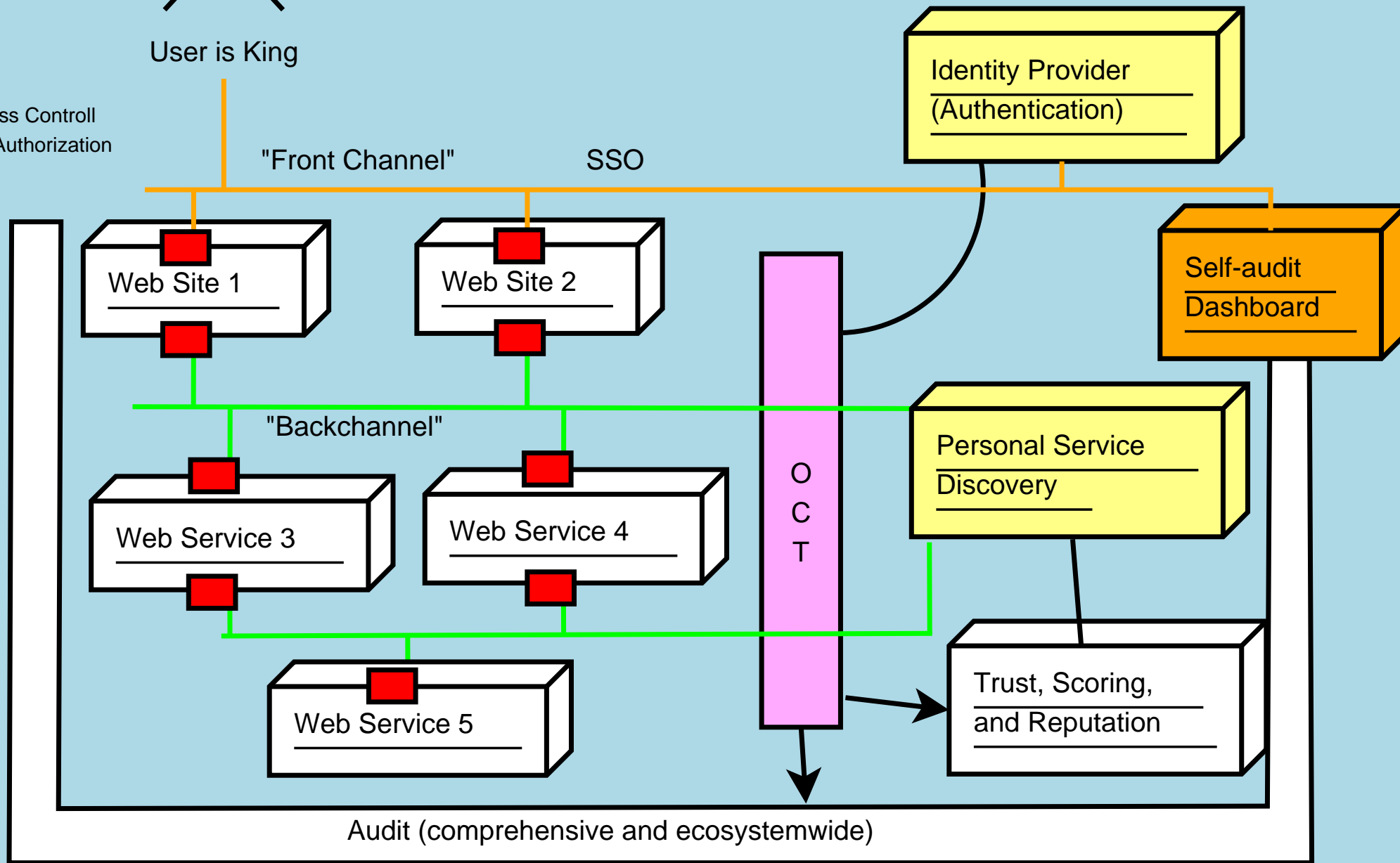


TAS³ Architecture Mini 2010

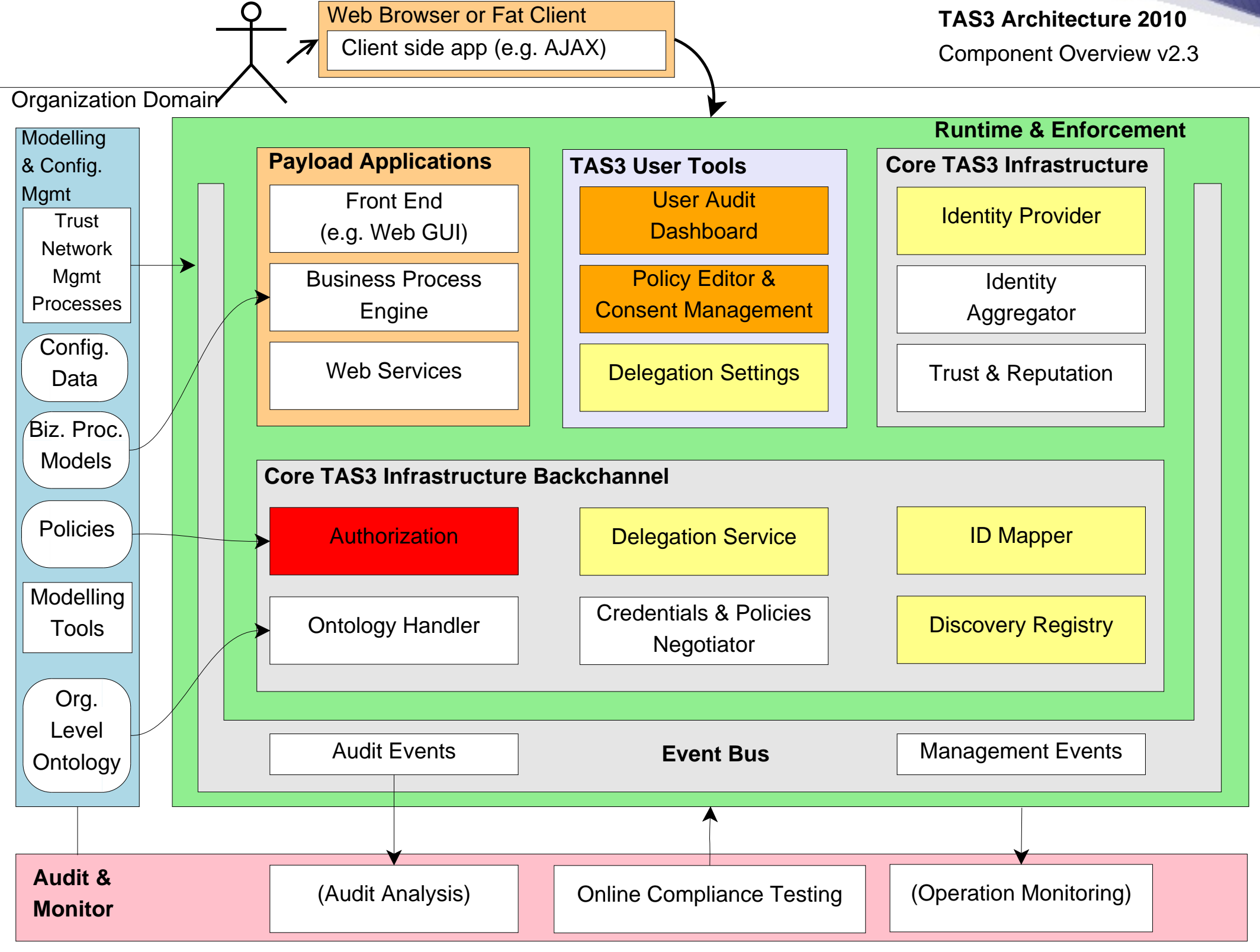


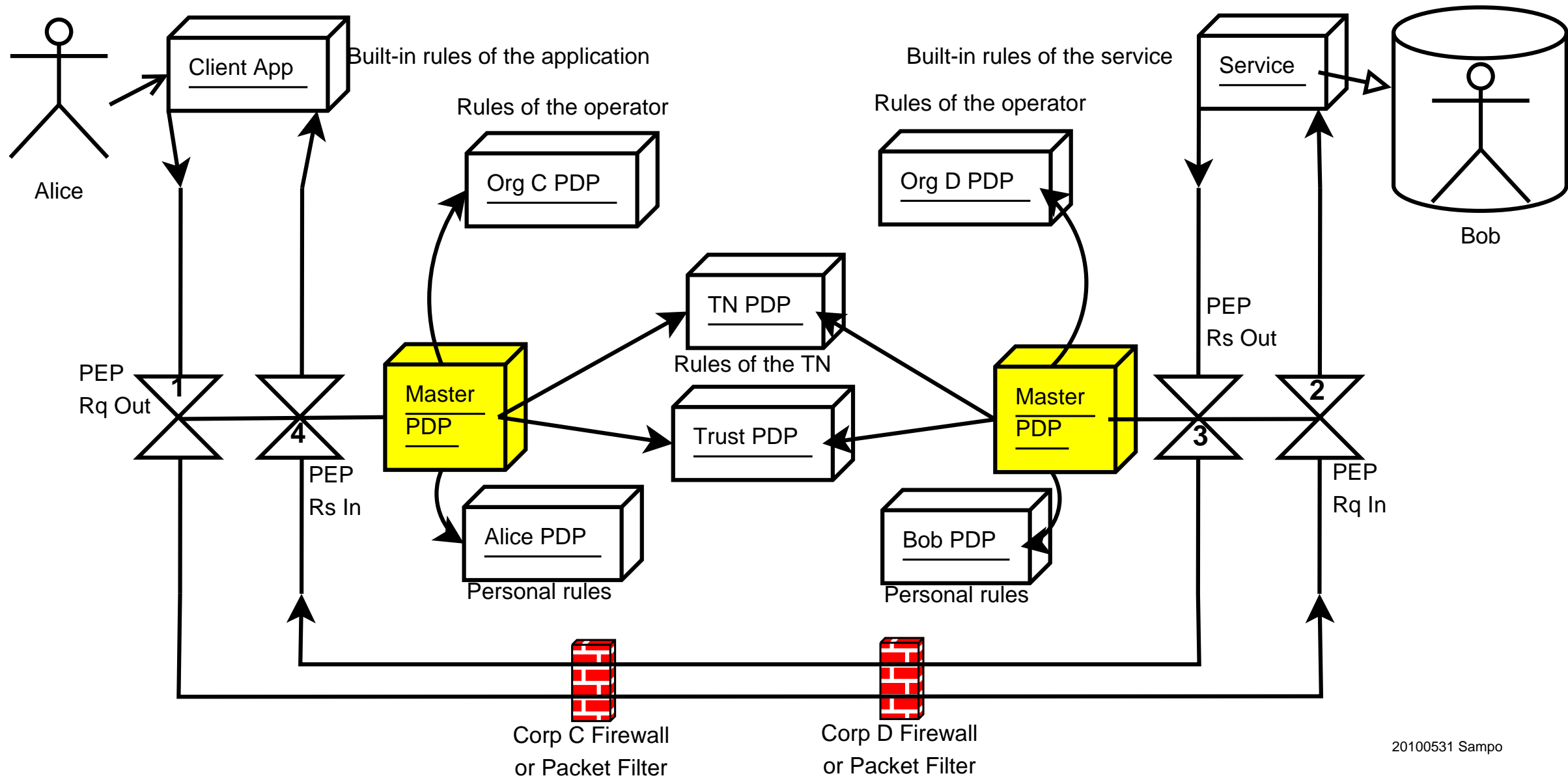
User is King

= Access Control and Authorization

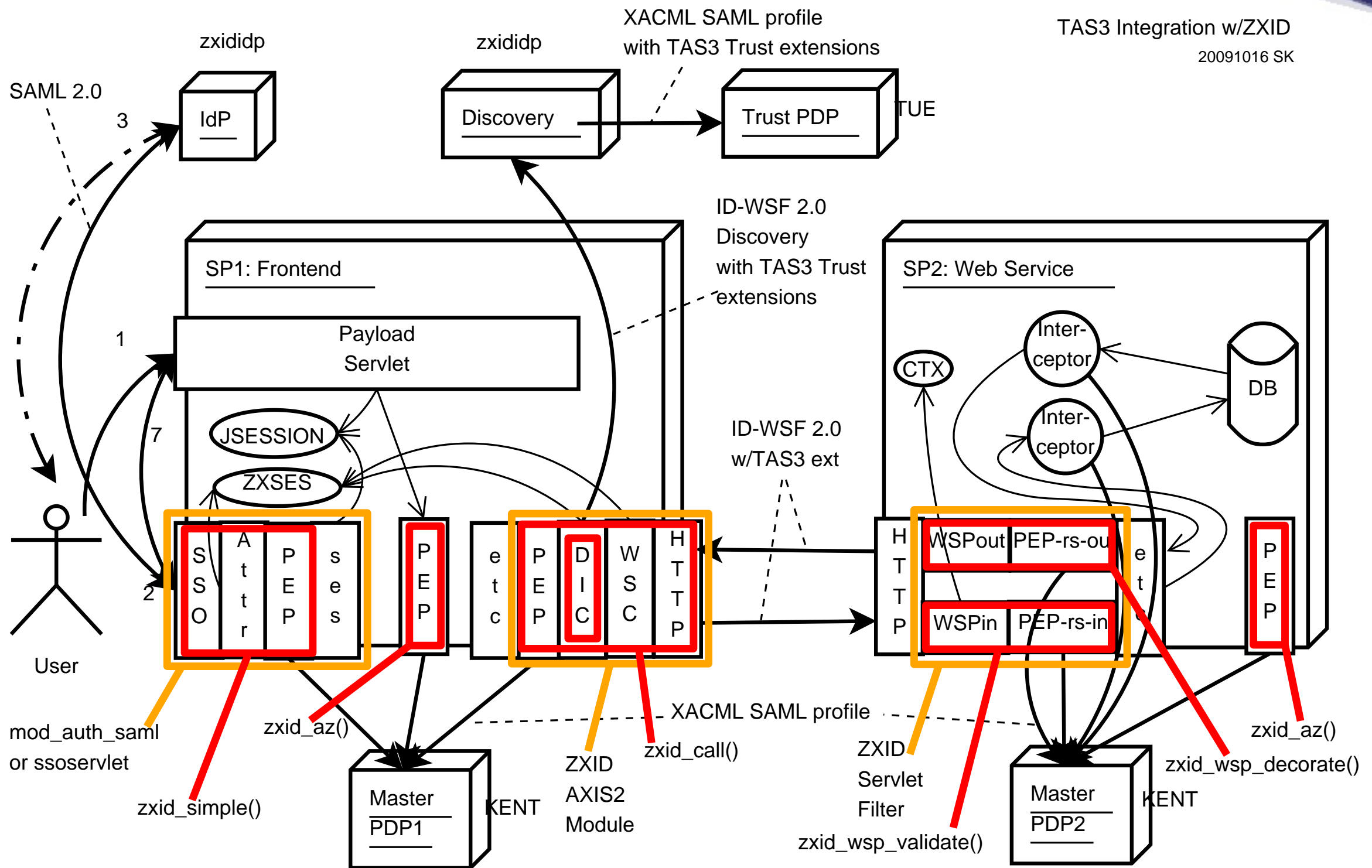


Governance & Interoperable Technology





20100531 Sampo



Promotors

TAS³ - Trusted Architecture for Securely Shareable Services

Core Security Architecture

IoS - Internet of Subjects Trust Convener and Ecosystem Builder

ZXID Reference implementation of the TAS³ Core Security Arch.



Core Standards

- OASIS SAML 2.0
- Liberty Alliance ID-WSF 2.0 & Data Services Template (DST) 2.1
- OASIS XACML 2.0 Access Control
- IoS and TAS³: Personal Data Store (PDS) Specification
- Sector specific data schemas
- Metadata standardization still TBD

IoS 7 Rules

1. Personal Control
2. Searchability
3. Instant Social Networking
4. Ubiquity
5. Symmetry
6. Minimization
7. Accountability



Big 4 of Privacy Protection (Seda *et al.*)

1. Awareness: Self audit (dashboard), Identity mirrors

2. Confidentiality

- Consent to release
- Reputation based screening, Trust and Privacy Negotiation
- Cryptographic protection
- Avoidance of correlation handles (prevent illicit collusion)

3. Control

- Intended purpose & Audience restrictions
- Sticky policies
- Policy enforcement & Audit

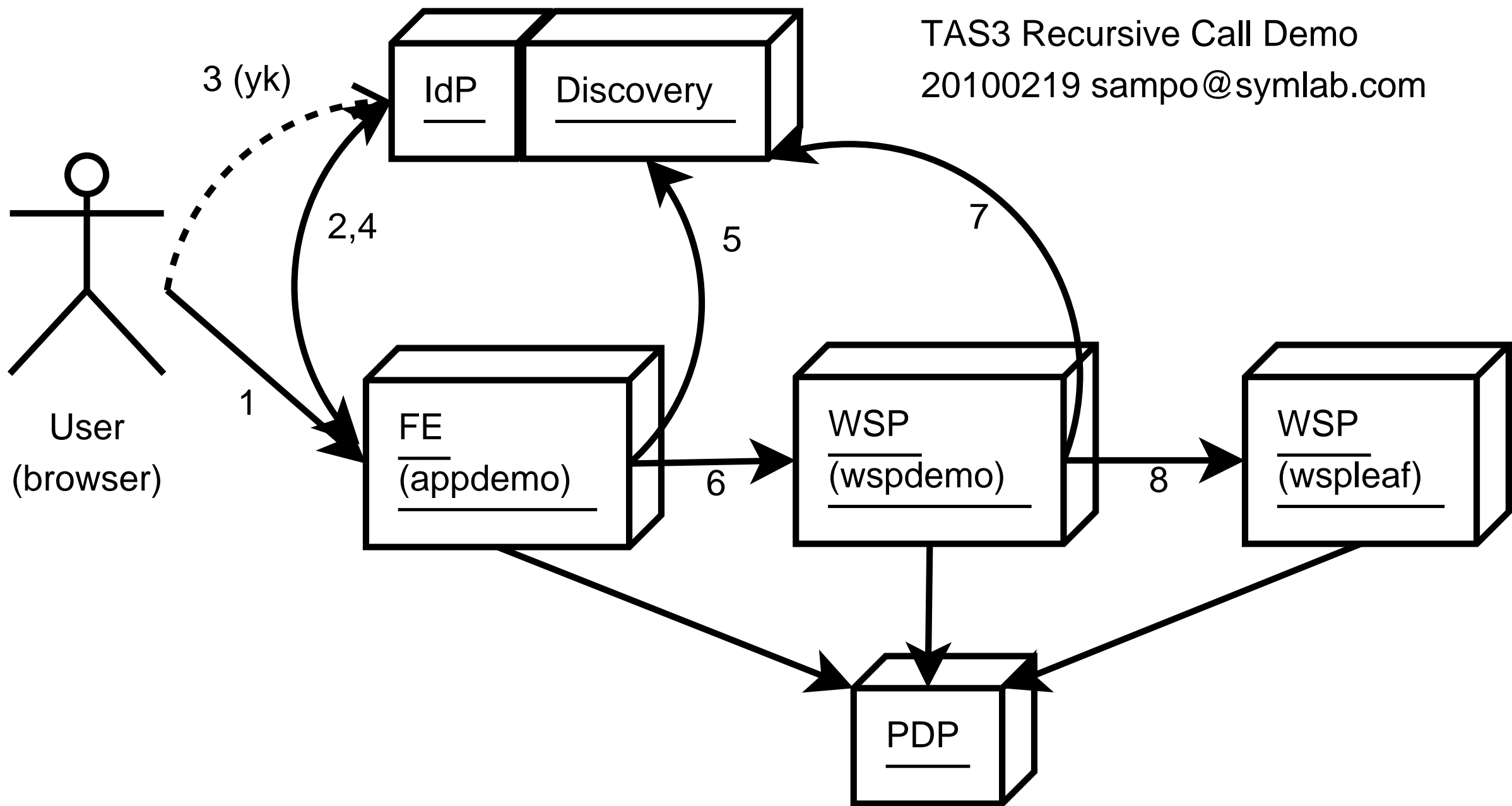
4. Practise

- Right to correct and delete, Right of response
- Trust and reputation feedback
- Send strong positive signal of your own

IoS Concepts

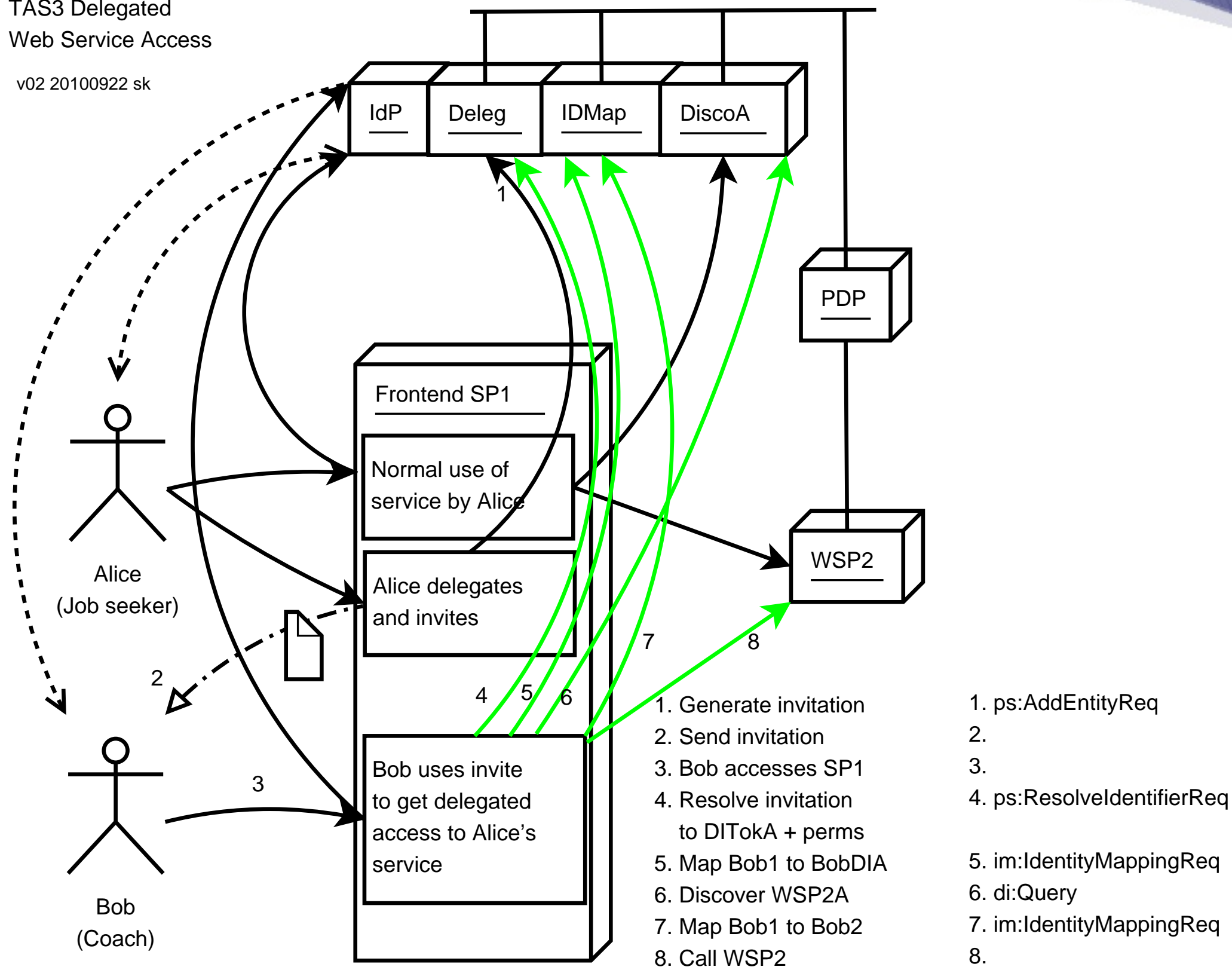
- IoS
 - IoS compliant Business Services
 - IoS Infrastructure
 - Dashboards
 - Shared WS: AIM, calendar, directories, harvesting, publication, ...
 - Personal data service(s) + dashboard (one or per service?)
 - Symmetry in providing services
 - Every user can become a Service Provider
- Personal - Communal - Public
- Separation of data from services
- Mostly pull and as-needed communication (minimization)

TAS3 Recursive Call Demo
20100219 sampo@symlab.com



TAS3 Delegated
Web Service Access

v02 20100922 sk



Delegation

1. Generate invitation

- Assign invitation ID for management of invitation
- Set up permissions for what resources invitee can access
 - The permissions can be keyed on invitee's identity, or
 - they can be keyed on the invitation ID

2. Send by out-of-band means, such as email or IM. The invitation will be formatted as a URL.

3. When Bob (being the invitee) clicks on the URL, he lands on Frontend site (alternatively Bob could land on WebGUI aspect of the Delegation server site)

- The site forces Bob to SSO (if this did not happen, invitation would be a bearer token)

4. The invitation is resolved to Discovery Token of Alice (the inviter)

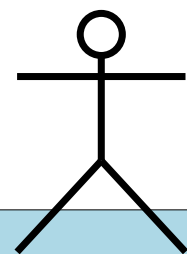
- The token contains as an attribute the invitation ID (the token is encrypted so that only the discovery service of Alice can open it, therefore the invitationID itself does not become a correlation handle).
 - Basically the discovery token of Alice would allow Bob to discover any service of Alice. As this is not desired, it is constrained by the permissions set at step 1.
 - Problem: how does SP1 accessed by Bob know where Alice's Delegation Service is located? This would be obvious if the URL points to the Delegation service of Alice.
5. For Bob to be able to call Alice's discovery service (next step), Bob needs to present his own identity token to DiscoA. This is obtained by calling Bob's ID Mapping service.
 6. Bob discovers Alice's WSP2. This is permitted by permissions.
 7. For Bob to be able to call Alice's WSP2 (next step), Bob needs

to present his own identity token at WSP2A. This is obtained by calling Bob's ID Mapping service.

8. Call to WSP2A is made with Alice's token from step 6 as TargetIdentity SOAP header and Bob's token from step 7 as wsse:Security/Token.

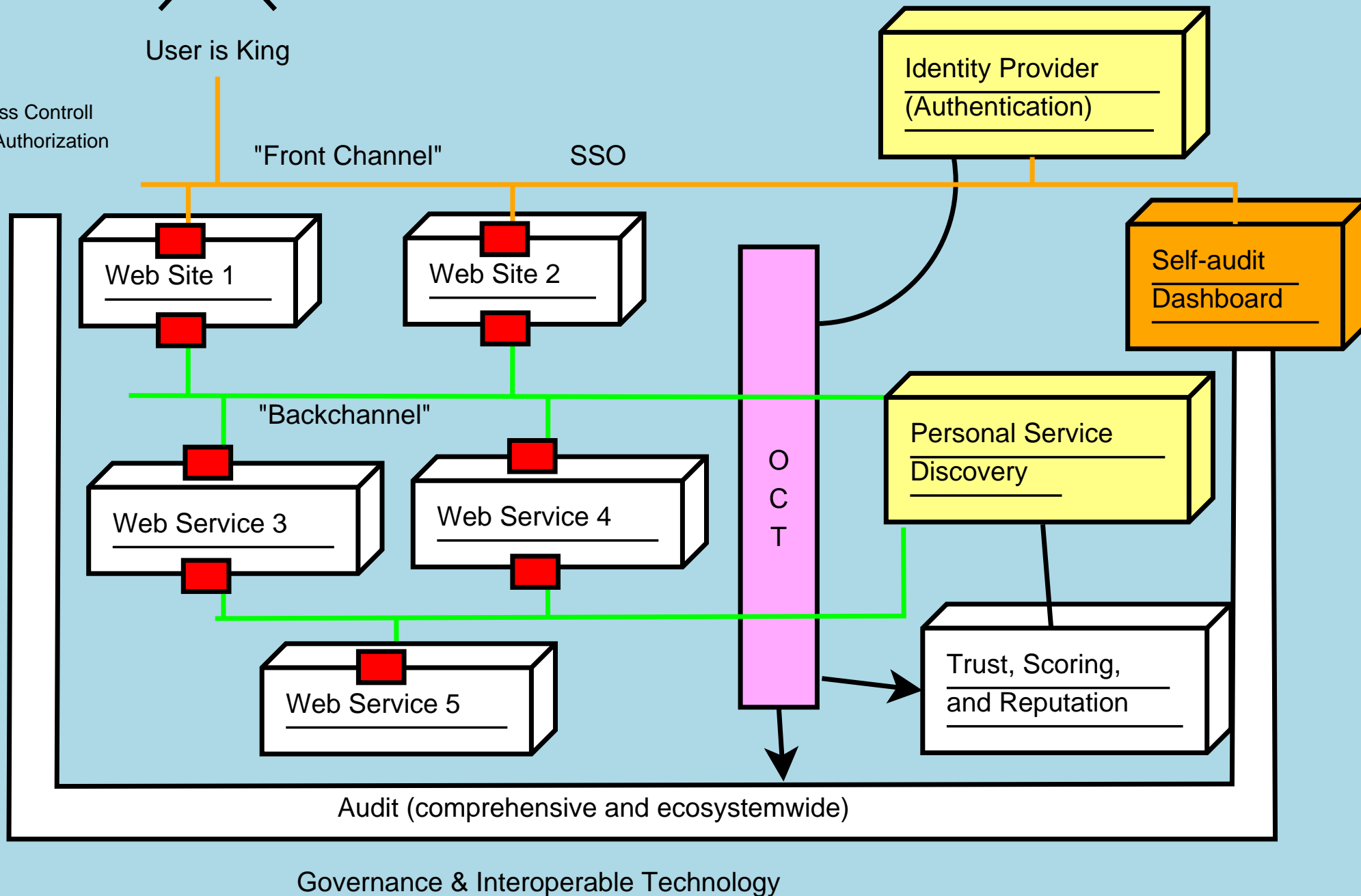
Ideally WSP2 would also have permissions indicating that the delegation from Alice to Bob is valid. This could be arranged by WSP2 making a call to Delegation service to confirm the delegation. Unfortunately such confirmation API is not specified by Liberty. We could invent an API. Another approach would be to at step 1 to provision the policies to PDP of WSP2.

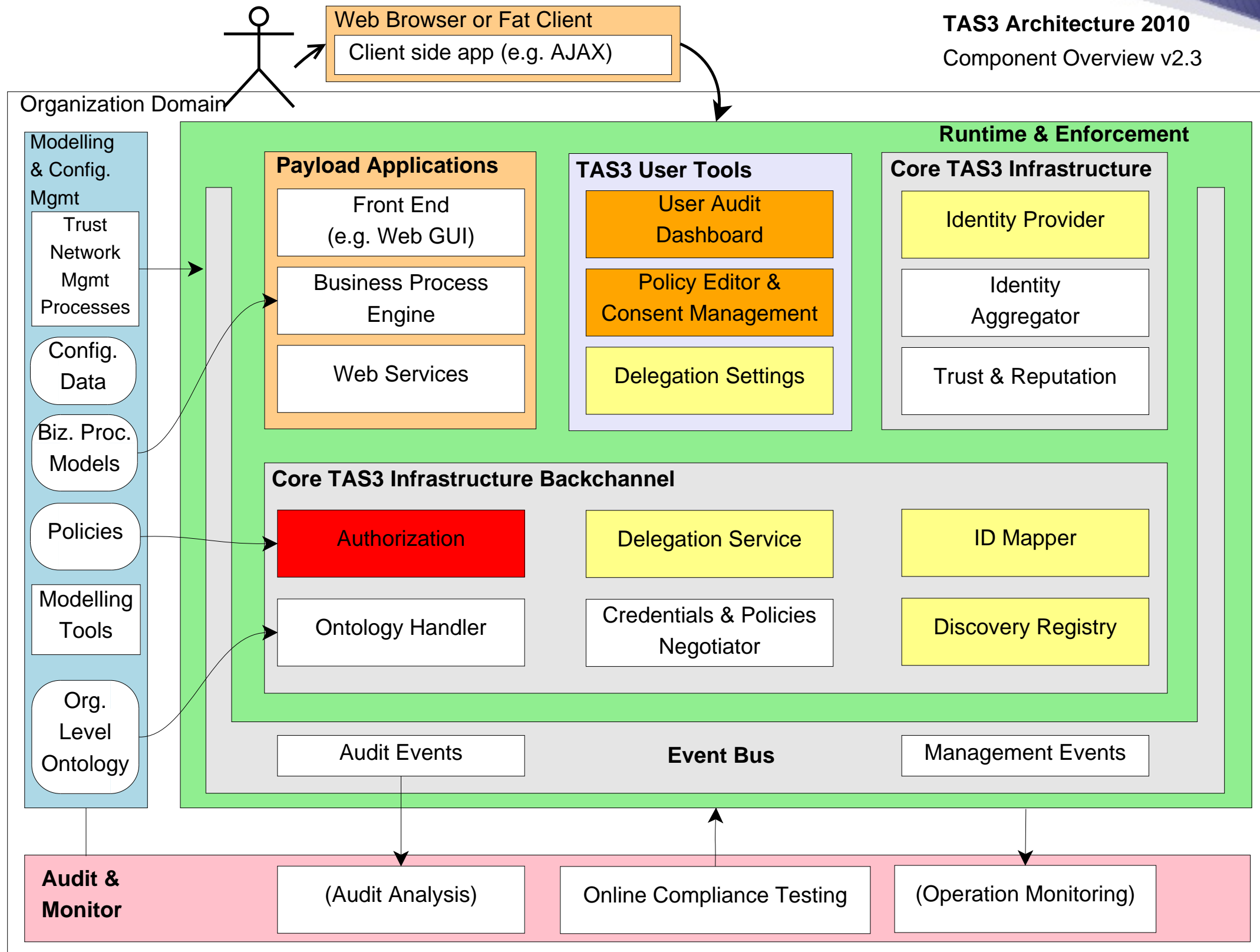
TAS³ Architecture Mini 2010

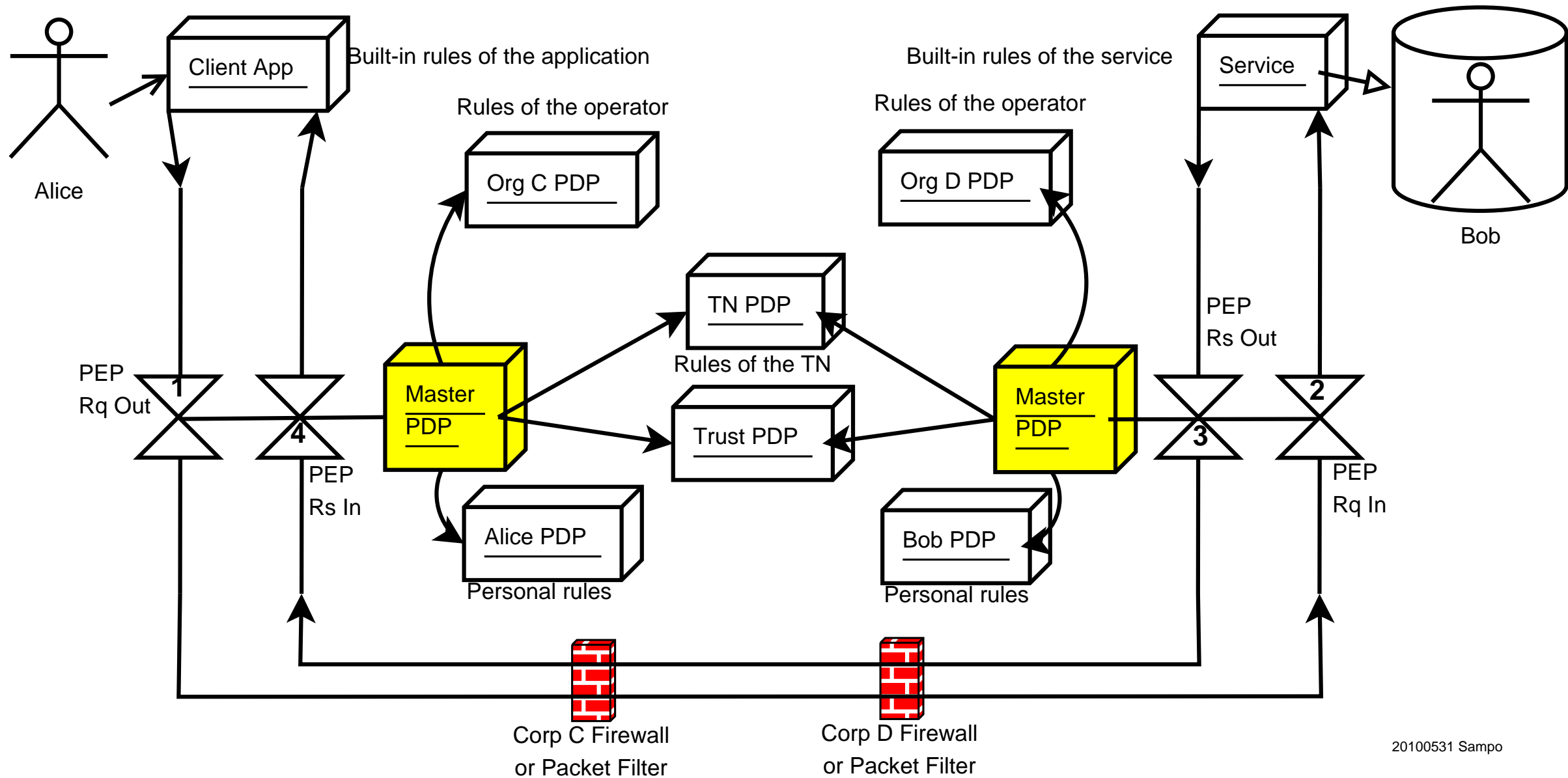


User is King

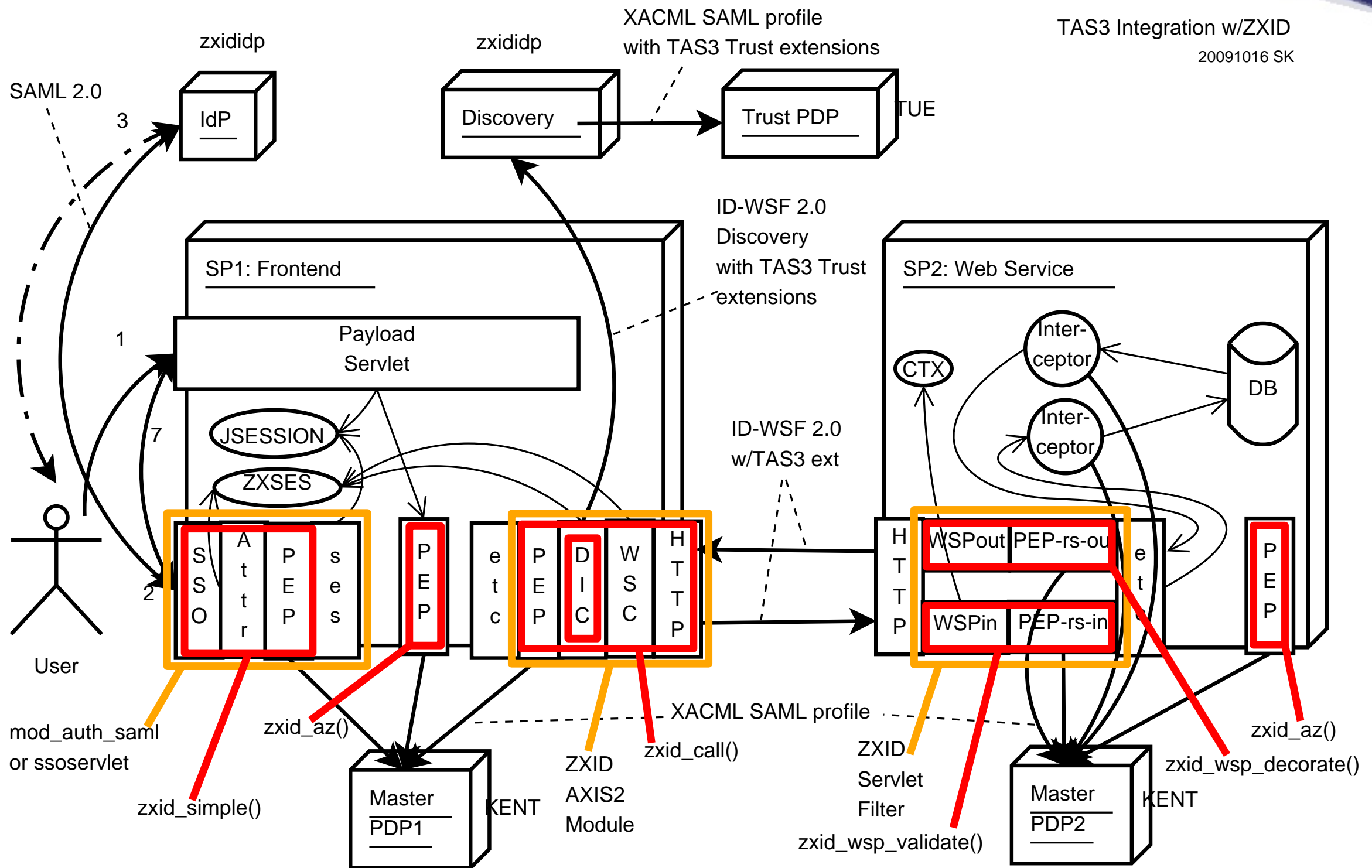
= Access Control and Authorization

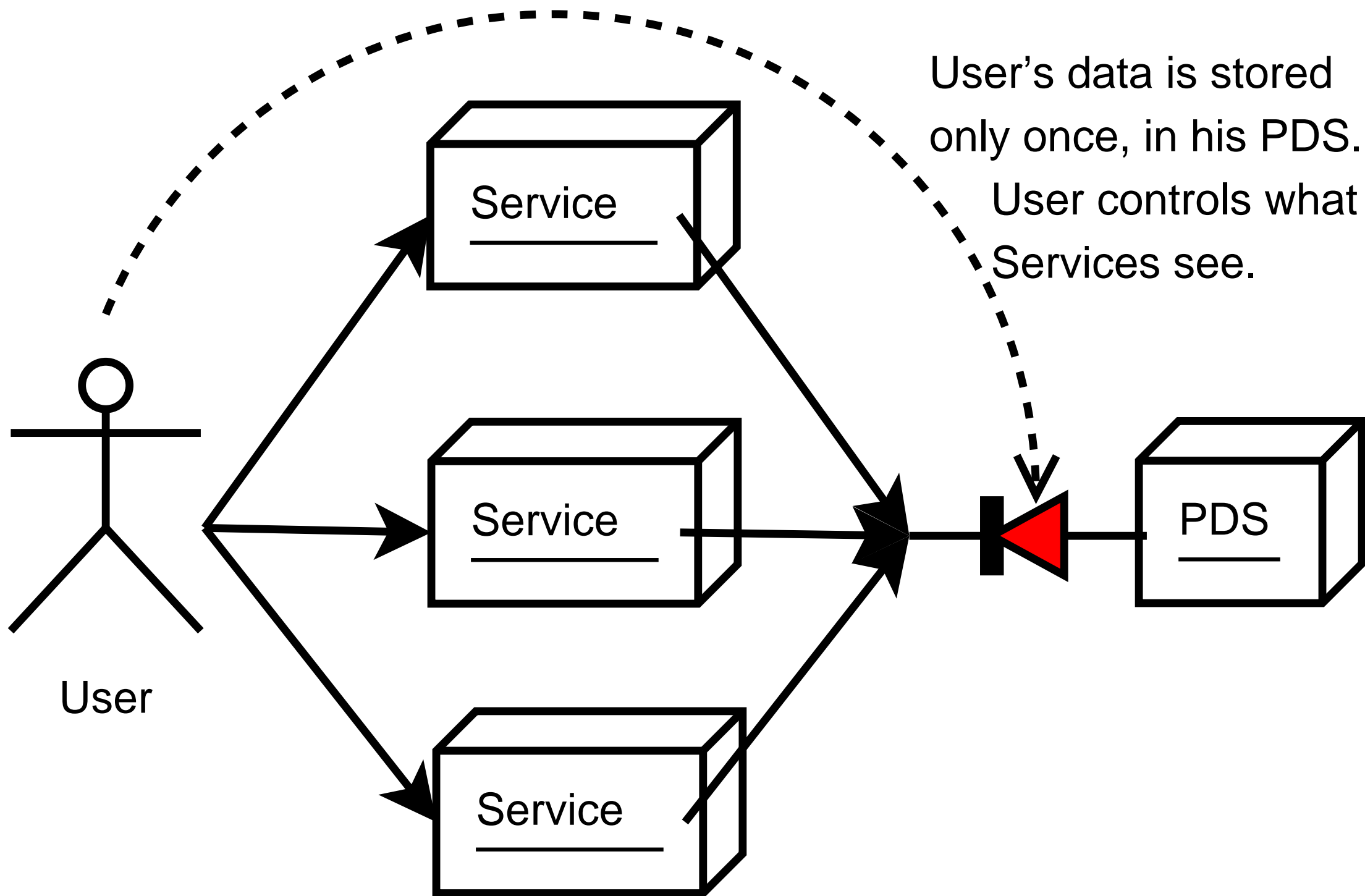


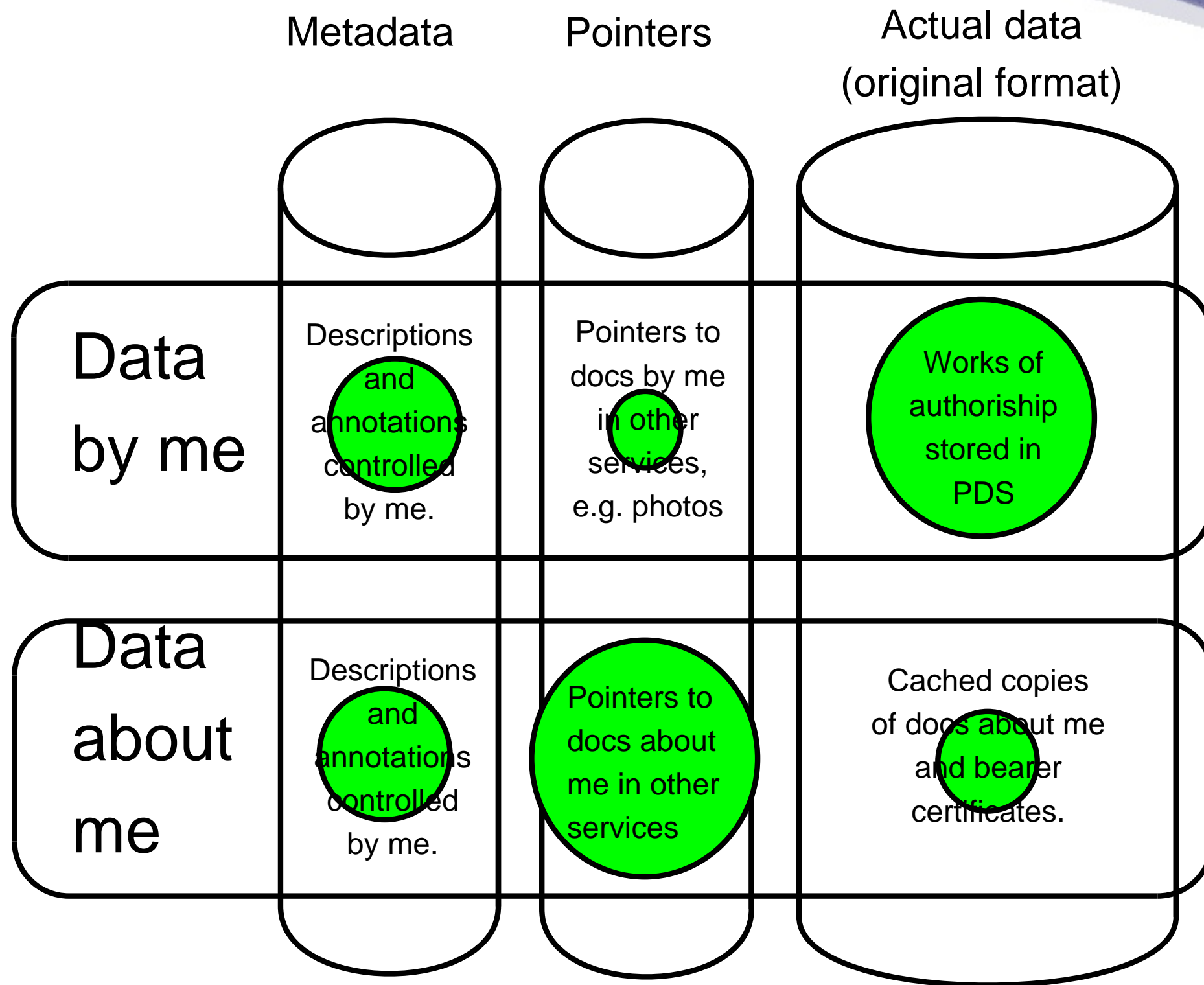


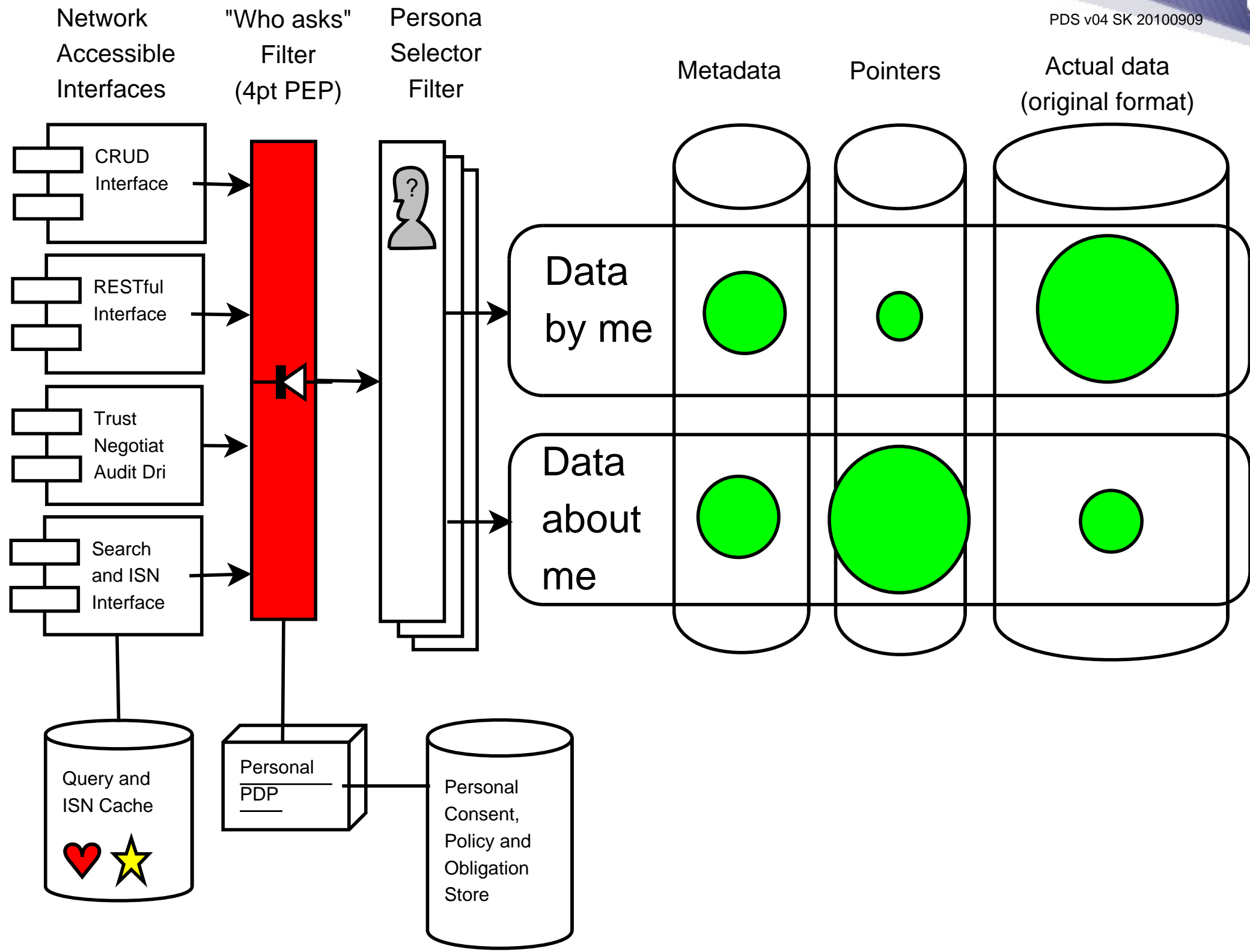


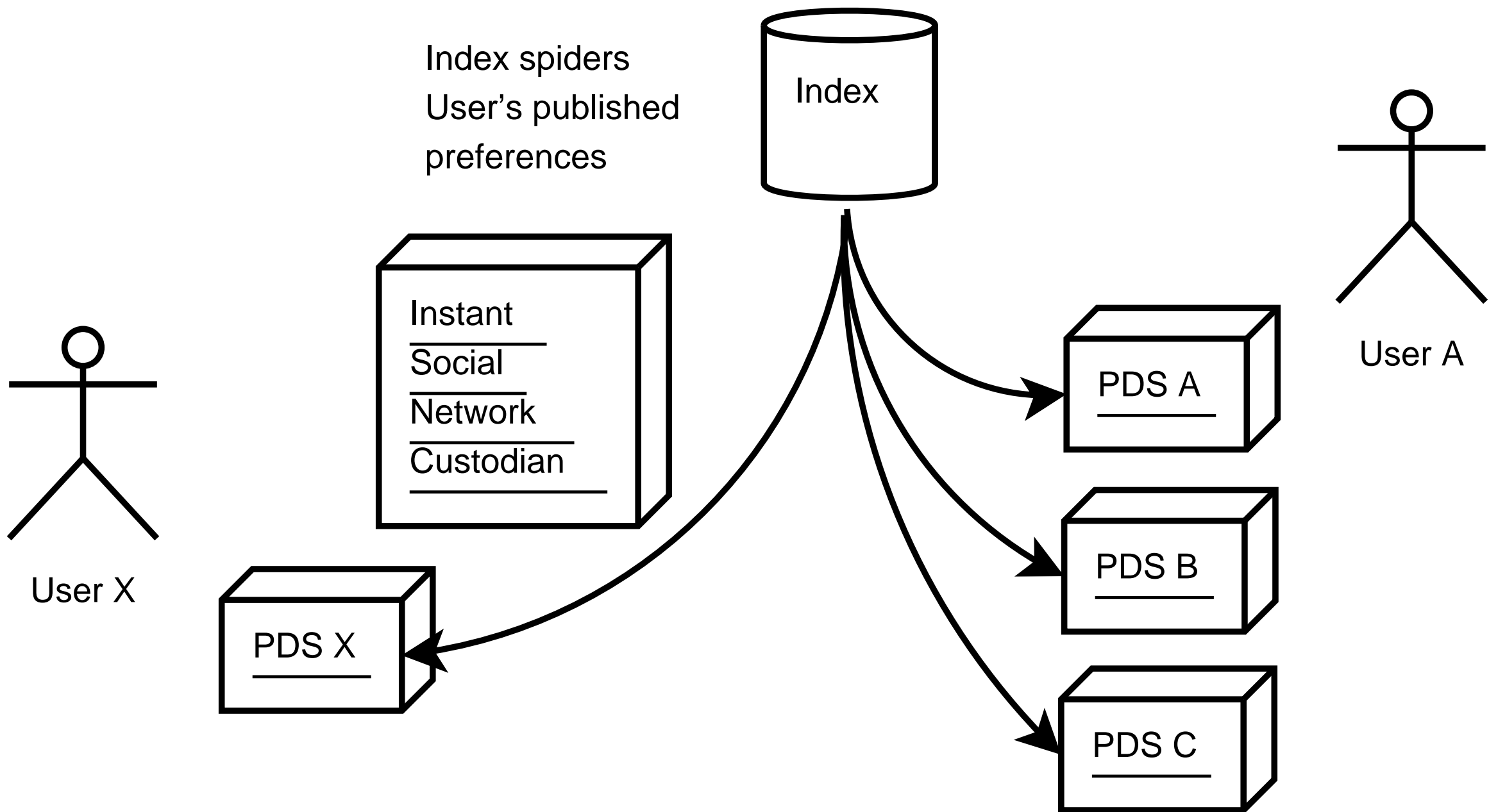
20100531 Sampo



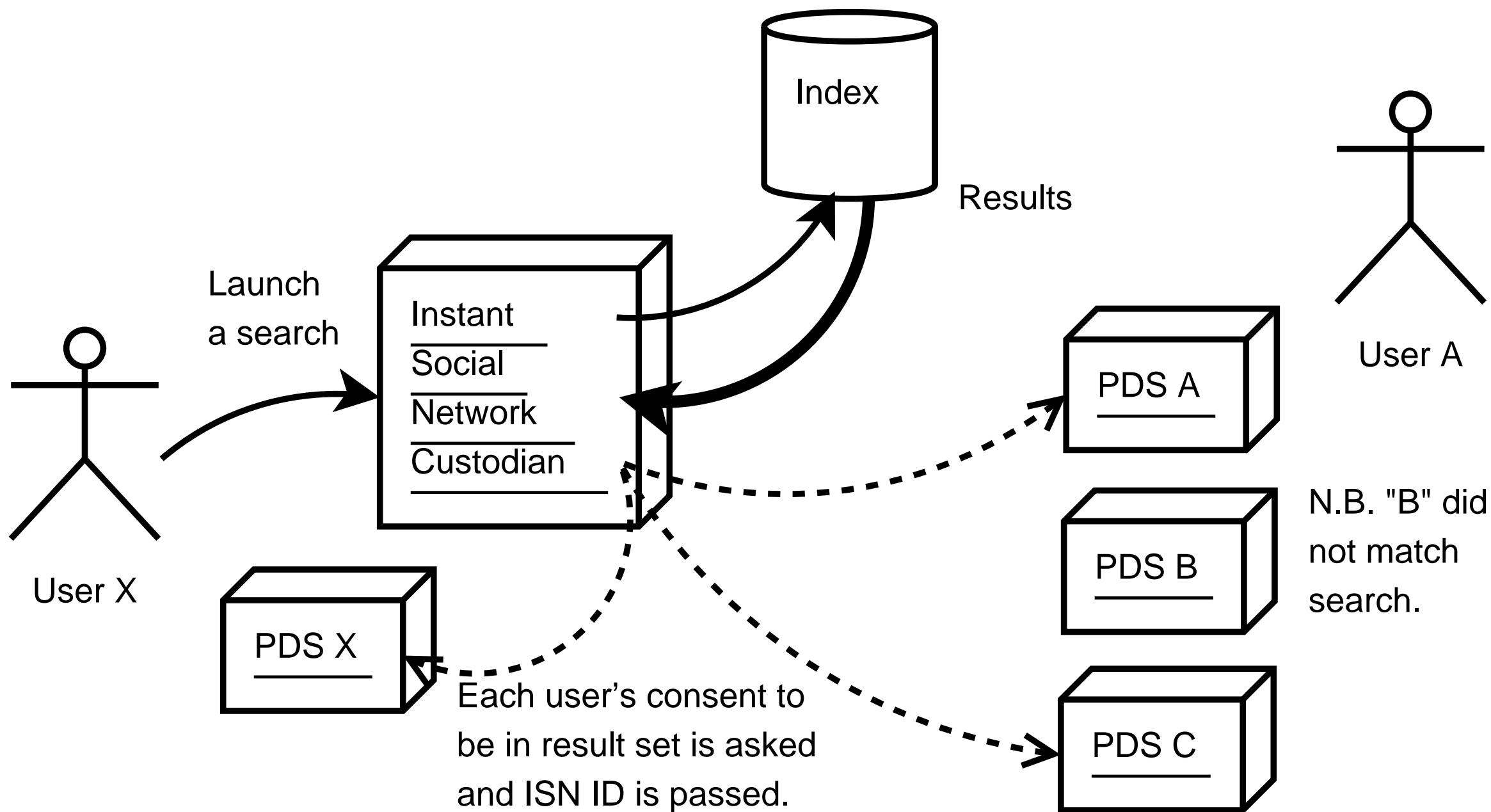


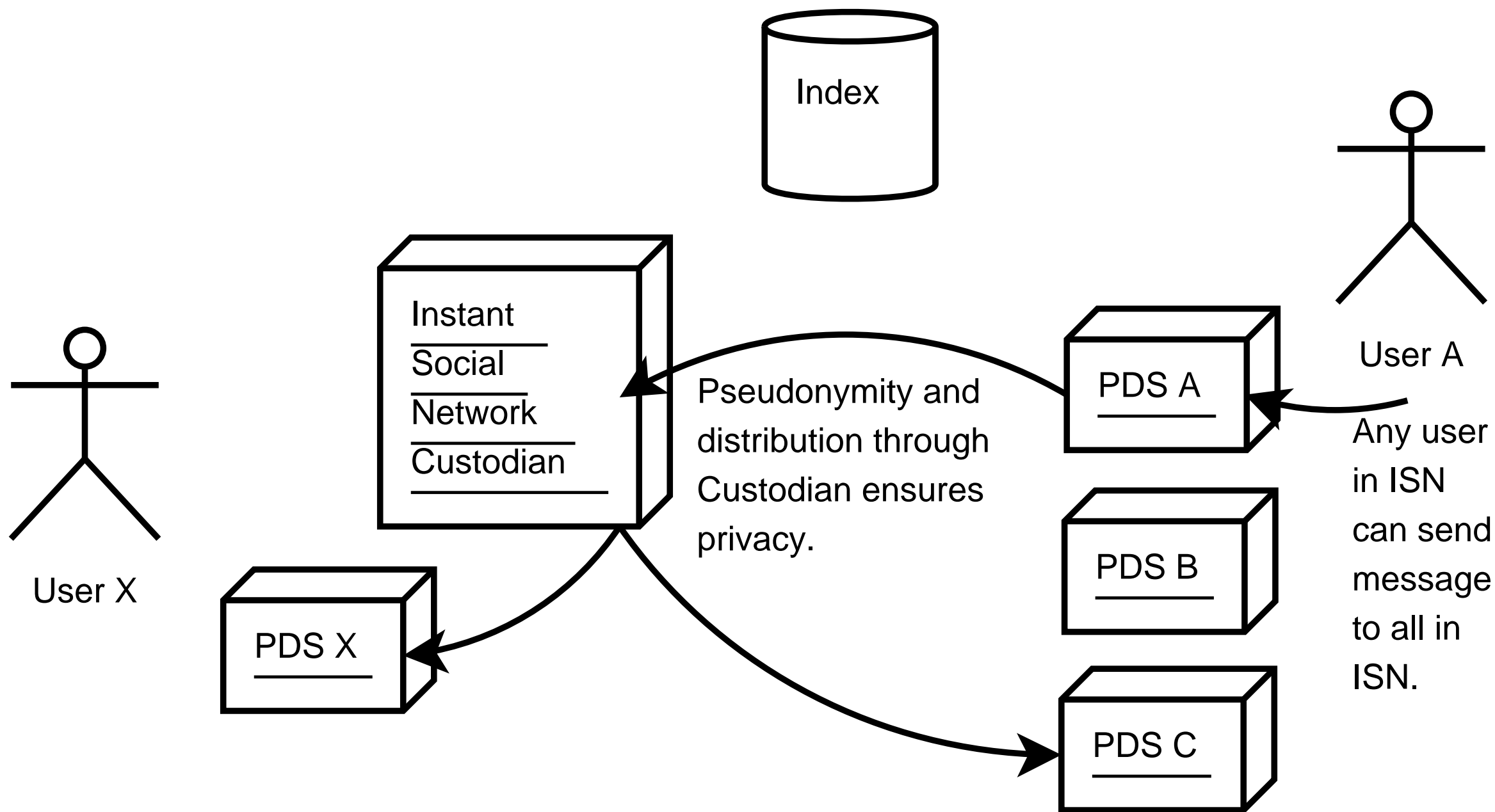




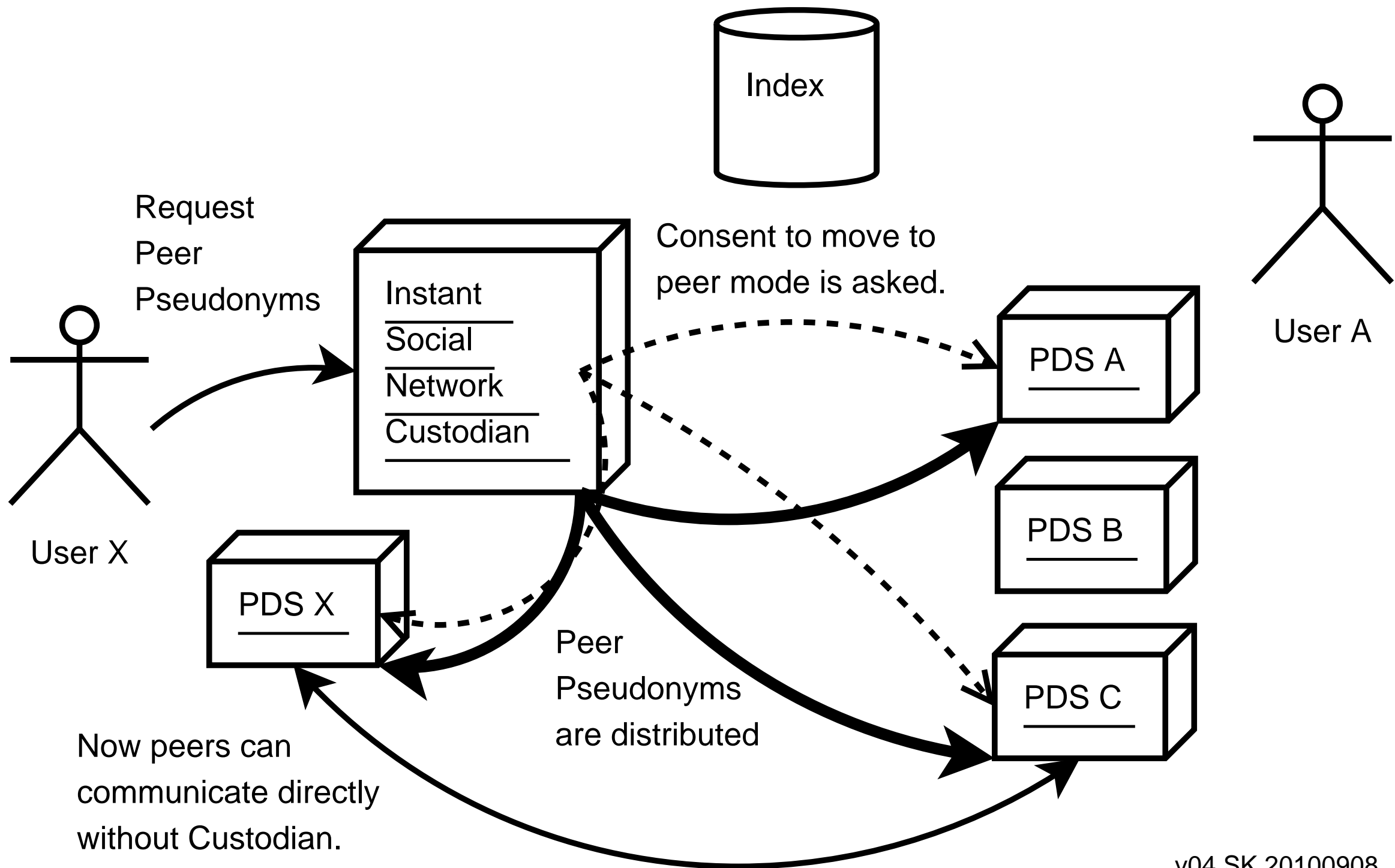


v04 SK 20100908





Any user in ISN can send message to all in ISN.



v04 SK 20100908

What is in Personal Data Store (PDS)?

- In
 - Core personal attribute data
 - cn / display name
 - language and other core preferences
 - core groups, tags, and roles
 - Age check?
 - Contact card, Shipping address / domicile
 - Personal documents at choice of user
 - Core social network (Social Data Store - SDS)
 - Contacts
 - Buddies and invitations and their permissions
 - Collaborative documents
 - Calendar data
 - Some audit records

- E-Portfolio / CV data
 - Degree certificates? Just references
- List of references to competencies
- Referees
- Personal Health Record? Copy of health records?
 - Possibility of managing personal doctor as member of your social network and keeping the records with him
- Fotos and videos
- Pointer to search, etc. Or discovery.
- Out (i.e. stored somewhere else)
 - Employee profile (maintained by employer's HR)
 - Per service preferences (maintained by each web site)
 - History or copy could be kept at PDS for backup
 - Shopping history (kept by each merchant), but copy could be kept at PDS for user's benefit

- Authorative health records
- Bookmarks
- Blogs

Services Provided by Personal Data Store (PDS)

- Attribute authority (for self asserted and long term signed credentials)
- Personal Data Broker
- Agent / Privacy Manager
- Audit Dashboard
- Persona switcher
- Index, search, interaction with harvesting, connecting to queries
- Pico payment processor
- Anonymous message router
- IdP / Authentication Provider?
- Discovery?
- Personal Policy Decision Point (PDP)?
 - Kantara User Managed Access (UMA)

Approaches for Personal Data Store

- Ideal architecture permits plurality of approaches
 - Not all approaches are acceptable to consumers of identity, thus flag the nature of data source (i.e. assurance level) so that self-asserted is readily identified and can be rejected.
- User must have choice (and competitive market of providers or approaches)
 - Discovery or bootstrapping will be the key enabler
- Every user can be a service provider: peer-to-peer (C2C, C2B, B2C)
- Managed model
- Personally owned model
- Network side (cf. virtual wallet) vs. user's desktop or device
- Roaming, multiaccess, simultaneous sessions and authorities

Variants of Personally owned model

- Personally operated model: run it literally on your own computer or smart phone
- Hosted model: it is as if you owned and operated it, but you buy it as a service (e.g. OVH root servers, Google Gear)
- Browser plug-ins or CardSpace
- Personal fat clients

Managed Model: Pros & Cons

- Pro
 - Easier for technically uninterested
 - Well managed, more secure
 - Convincing authentication and authority
 - Nannying: ability to prevent users from doing stupid things or at least advice them
 - Systematic disaster recovery
 - Cheaper per unit
 - Business model: pay for utility, clear promoter
 - Easier to arrange alternate revenue from searches and aggregations of data
 - User-not-present easy to support

- Contra

- Loss of control and lack of influence / bargaining power against too big providers
- Fat target and high impact of failure
- Capital intensive
- Offline use cases difficult to support

Personally Owned Model: Pros & Cons

- Pro
 - More tangible ownership and control of data
 - Offline use cases (except for rented/hosted cases)
- Contra
 - More difficult for technically uninterested (but rental/hosted approach can ease this)
 - Unconvincing authentication and authority
 - If you break it, you get to keep both pieces. Nobody to help.
 - No systematic disaster recovery
 - User-not-present difficult to support

User Centricity & Front Channel - Back Channel

- User centricity: user control. *Not about shifting bits through UA.*
- Front ch. doesn't really provide better guarantee than back ch.
 - User centricity requiring all traffic to pass through a user agent is a *flawed* notion and does not address deep web services reality
 - May be easier to arrange for user interaction from back channel
- Back channel is often a *really* required and undisputed part of architecture: not supporting it, will only serve to exclude PDS from those architectures.
 - User interaction from back channel: difficult, not impossible
 - Interaction Service can be used to contact the user from deep in the call chain.
 - (TAS³) business process aware Dashboard can be used to solicit user interaction and unblock a process that was stuck waiting for user input.

Available Standards and Stacks

- TAS³ (SAML2 + ID-WSF) (deploy per user, if desired)
 - Fully pair-wise pseudonymous privacy protection
- FOAF style
 - Built-in assumption of globally unique ID and correlation handle
- Liberty Advanced Client aims at providing truly pseudonymous IdP and services from personally owned devices
 - Also supports disconnected model
- Higgins work?
- Skunkworks and new developments?

How to harmonize these so that Managed and Personally Owned, all the way to on-device, models can co-exist?

- TAS³ decentralized + Liberty Advanced Client: an elegant solution

Applications

- Education
 - Mahara (work to separate database interface from rest of application / service)
 - Moodle (work to separate database interface from rest of application / service)
- Employment
 - Some matching / job seeker application, TBD
- Social networking
 - Wizi: ability to leverage core social network and profile
 - Nice iPhone app, good demo. But requires convincing CEO of a very busy company
 - Some sort of "contact kiss" application, TBD
- Other, Ideas?

Reality Check

- PDS and IoS infrastructure is a tall order, we can not have all of it on day one
- Initial core set of data?
- Initial core set features?
- Initial demonstration applications?
 1. Moodle vs. Dokear
 2. Mahara vs. Elgg
 3. Universal CV
 4. Wizi
 5. TAS³ and Kantara project web sites (Trac, Altassian Confluence)
 6. Web Mail (pdmail)
 7. Other?

PDS Data Priority List (London 2010)

1. Core contact card
2. E-Portfolio data
3. Audit records
4. Core social network
5. Core preferences, tags, and roles
6. Distribution of long term signed credentials from authoritative sources, age check
7. Advanced social data store
8. Personal and collaborative documents
9. Calendar data
10. Personal Health Record

PDS Feature Priority List (London 2010)

1. Discoverable, network side data store
2. IdP and Discovery support (even if not yet personally managed)
3. Audit dashboard
4. Agent / Privacy Manager / Personal Data Broker – first iteration
5. Index, search, interaction with harvesting, connecting to queries
6. Pico payment processor
7. Anonymous message router
8. Persona switcher
9. Personally owned PDS
10. Personal IdP, Discovery, service provider support
11. Better Audit dashb. / Agent / Privacy Mgr / Personal Data Broker
12. Personal Policy Decision Point

Requirements for PDS Software

We seek to convince *software developers* to implement PDS.

- Commercial (whether licensed or runs as SaaS model)
- Open source

Lets see what is included in such software...

1. Web Service

2. Web GUI

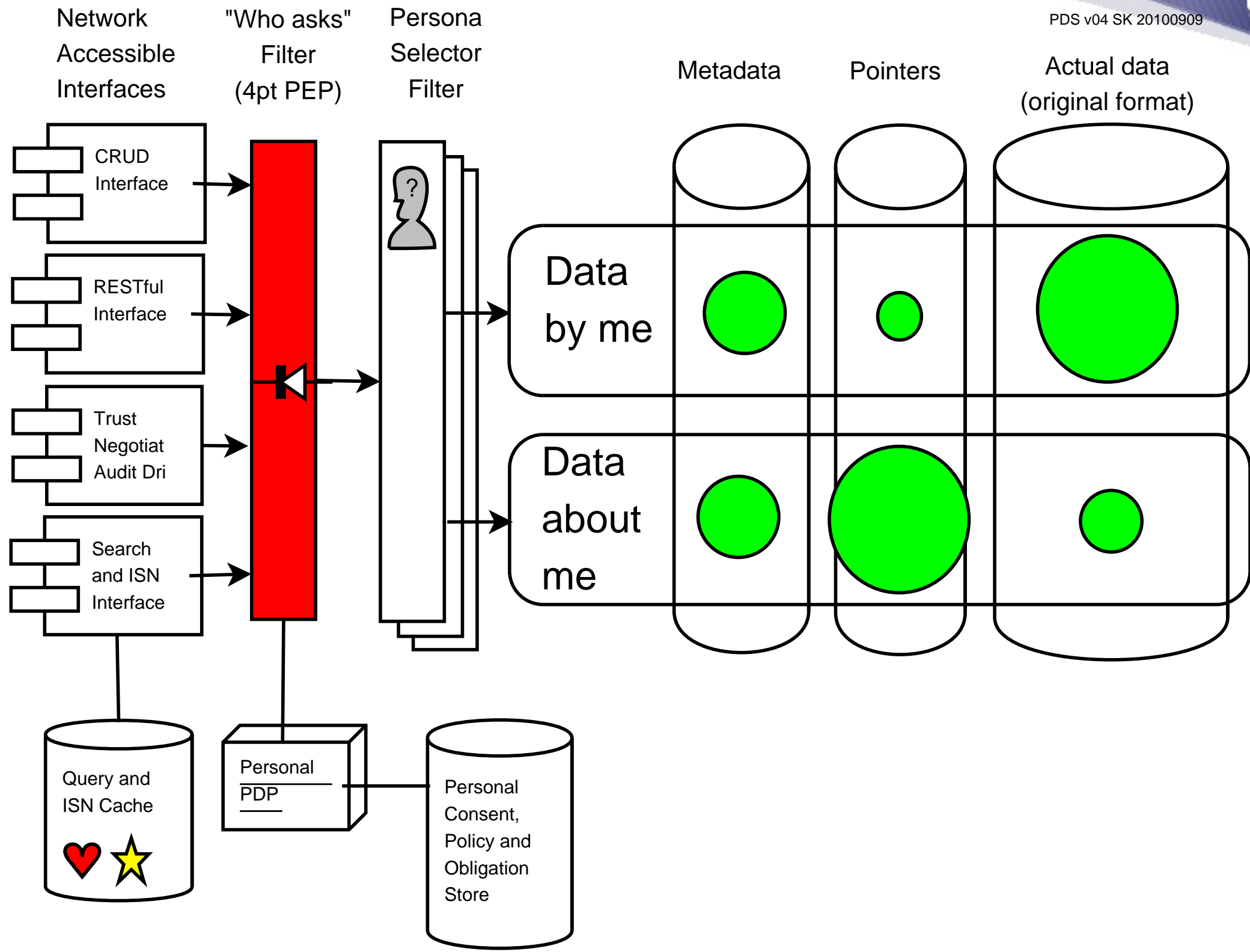
3. Supporting infrastructure such as

- Databases
- PEPs and PDPs
- Audit features

Much of this is needed to be a "TAS³ Web Service"

PDS Technical Properties: Scope

1. TAS³ web service, with full support for relevant TAS³ features
 - Data access using Liberty Data Services Template (DST 2.1)
 - Service Type "urn:ios:pds:2010-05:dst-2.1"
 - CRUD methods, box carrying, Subscriptions and Notifications
 - MTOM to preserve data in original format
 - Simple read-only data access (RESTful, SAML Attribute Query)
 - Distributed search responder (possibly part of R of CRUD)
 - Audit drill down as web service (to be specified)
 - Service Type "urn:tas3:audit:2010-06"
2. Web GUI (stand-alone, iFrame for data user, iFrame for Dashbrd)
 - At least basic privacy preferences management
 - Right-of-Access, Rectification, and Deletion
 - Audit drill down as GUI



GRAPHIC (ios-pds-db-struct-bg,fg,by,az,api,dash)

GRAPHIC (ios-pds-db-struct-bg,fg,by,az,api,dash,idp)

IoS PDS Special Requirement for ISN



To support *Instant Social Networking* (ISN) the PDS needs to provide:

- Special WAN indexable and anonymously (really anonymously, in some cases pseudonym may not be sufficient) searchable interface.
- If you are matched by a search, you gain equal rights to communicate with the other members of the result set (anonymously and progressively revealing details about yourself). This is *symmetry*.

"WAN indexable" means indexable by Google and similar services. This functionality is important for the business case of IoS, but is still in flux.

IoS Indexed, but, Distributed Search



One of the key elements of the business model of the Internet of Subjects is for the user to consent and accept to be found by searches of openended nature. The information you make available to such and other searches constitutes an important part of your "practise" of identity. We encourage legit players to strongly broadcast all their positive evidence.

PDS: TAS³ Binding Features

- Fully discovery based
- Fully pair-wise pseudonymous
- Both Requester Token and TargetIdentity token support
 - Foundation for delegation support
- UsageDirective header with SOL1 expressions
- Integrated to audit bus (messages TBD)
- 4 point PEP with external PDP capability
- SOAP w/XML-DSIG now
 - eventual RESTful binding w/Simple Sigs

PDS Data: Labeling

- *By Me*
 - Original data, or
 - Pointers to places where there is data by me
- *About Me*
 - Pointers to places where there is data about me
 - Copies of data, with signatures intact, about me
- Version control or history feature (need guidance from IoS steering group re how sophisticated)
- **Persona** Support (perhaps as branches in version control?)
- Resource granularity vs. subresource granularity
 - Labeling and data schema granularity directly determines the possible access control policy granularity

PDS Data: Format

1. Metadata: RDF (XRD?) w/Turtle or N3 serialization vs. JSON
 - TBD soon, please provide feedback and suggestions
2. Pointer: `< EPR of server + identity + Local pointer >`
 - EPR** (URL + token) allows locating the server on the net
 - Identity** a pair-wise persistent pseudonym, essential to prevention of correlation and emergence of GUID for the resource
 - Local pointer** allows multiple resources under one identity
3. Original data:
 - Copy of the data in original format, signatures intact
 - Pointer to original source is kept
 - MTOM binary clean enveloping in protocol: data and sigs intact

PDS Data: Schema and Data Vocabulary

- PDS and metadata are schema agnostic at basic layer (no bias to any particular schema)
- Metadata schema standardization desired
 - Common vocabularies are easiest way to have interoperability
 - Some common basis
- Recommend schema standards for some immediately pertinent datasets, e.g.
 - ePortfolios

PDS spec (WIP)

Detailed specification by Sampo *et al.* is available as

`draft-ios-pds-v01.pdf`



Thank You! PDS, IoS, TAS3, & ZXID bow to You

Sampo Kellomäki (sampo@zxidp.org)

+351-918.731.007

skype chat: sampo.kellomaki

