Quick DER & LillyDAP

(and controlling your online identity)

Context → ARPA2

ARPA2.net wants to do a lot:

- Put users back in control of the Internet
- Open protocols connecting any2any

InternetWide.org funds development

Context → ARPA2 → Hosting

How can end users control online presence?

- Techies run a server machine
- Techies have their own domains
- Techies plug in their own servers

End users are the bait of a few silo's…

…they need better hosting facilities than now

Context → ARPA2 → Identity

Most vital part of online presence?

- Services, yes sure, but that's easy / local
- Setup and manage identities and *control it*
- Have a domain with users, groups, aliases, …
- And *Bring Your Own Identity (BYOID)*

That's our project **IdentityHub**

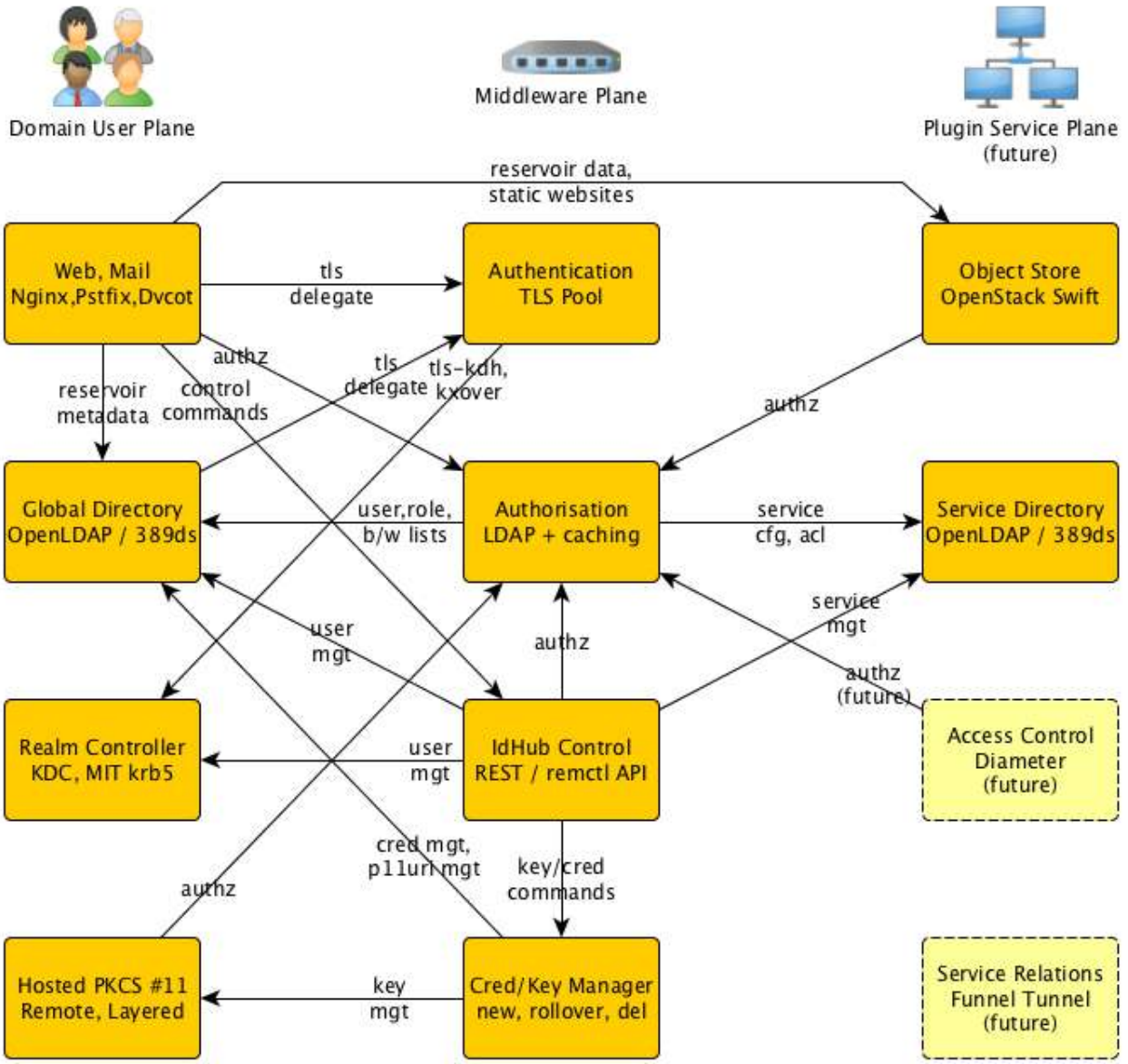
Context → ARPA2 → IdentityHub

How does one share identity *globally?*

- We'll need an *open standard*
- …and not Facebook, not Google+

- Publish X.509, OpenPGP, OpenSSH
- Host private credentials (in PKCS #11)
- HTTP is generic… LDAP is *just right*

Context → ARPA2 → IdentityHub



Bring Your Own IDentity

We *must* to rely on standards…

- Trust a domain via *DNSSEC*
- Locate its LDAP server via *DNS SRV*
- Validate its server cert via *DANE*
- Locate domain users with *LDAP attributes*
- Obtain *public-keys* over LDAP

…and they’re all there!

REST – / – LDAP

- Typed BLOB transport
- Know file locations
- Layout: non-standard
- Typing: non-standard
- Auth: Many, local prefs
- Ext: prone to clashes
- X.509,PGP,SSH: tbd

- Objects with Attributes
- Search by attributes
- Layout: Immaterial
- Typing: global, unique
- Auth: SASL
- Local ext: still unique
- X.509,PGP,SSH: done

Using LDAP

- Search under a [domain name] base:
  dc=example,dc=com
- Search for attributes of choice:
  uid=john
- Receive matching objects with their attributes:

  dn: uid=john,ou=People,dc=example,dc=com
  cn: John Doe
  uid: john
  mail: john.doe@example.com

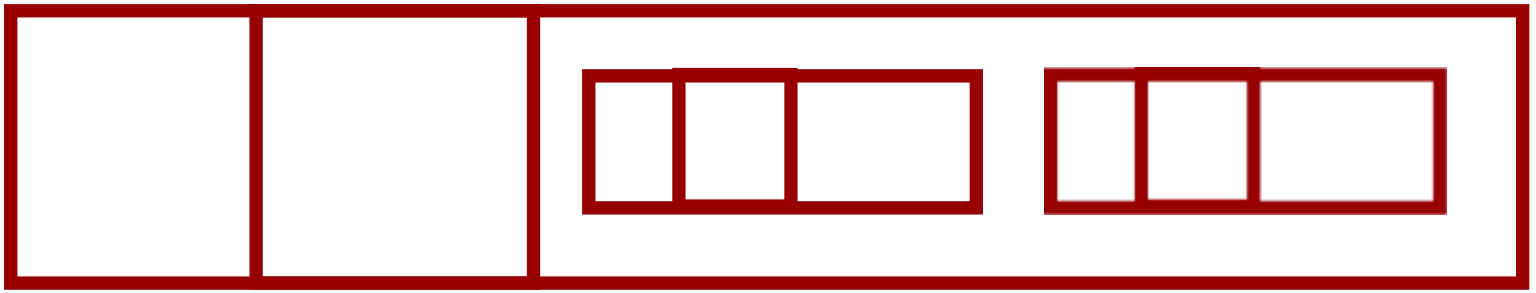
Advantages of LDAP

- Solid and fine data access
- The *only* open protocol for data
  - SQL is *not* a protocol
  - REST granularity *very* course (MIME-typed BLOBs)
- Excellent infrastructure
  - Proper IETF standards
  - Efficient, redundant, replicating storage servers
  - SRV records for domain linkage (ah!)
  - Supported by X.509, OpenPGP

(Current) Downsides of LDAP

- Complex servers
  - Designed for storage, not for dynamic data
  - No cgi-bin / python / … scripting
- Dreadful tooling
  - Well… integrated tools are actually quite cool
  - Some good web wrappers do exist
- Difficult to extend
  - Need internals of storage-based engines
- Enter **LillyDAP**

LDAP uses BER ≈ DER



- Tag, Length, Value
- Possibly nested inside Value
- DER is a canonical subset of BER

ARPA2 project

Not XML Schema?

Verbosity! Readability!

```
<xs:complexType name="PurchaseOrderType">
  <xs:sequence>
    <xs:element name="shipTo" type="USAddress"/>
    <xs:element name="billTo" type="USAddress"/>
    <xs:element ref="comment" minOccurs="0"/>
    <xs:element name="items"  type="Items"/>
  </xs:sequence>
  <xs:attribute name="orderDate" type="xs:date"/>
</xs:complexType>
```

Is ASN.1 better?

More terse, more readable:

```
PurchaseOrder ::= SEQUENCE {
    dateOfOrder    DATE,
    customer       CustomerInfo,
    items          ListOfItems
}
```

Then how about JSON?

Specs can be too terse:

(...)

JSON defines *no formal types*

Quick `n' Easy DER → API

- Quick DER is "just" a BER parser…

```
ok = der_unpack (&inder, stx, &cert, 1);
```

…and DER packer…

```
sz = der_pack (stx, &cert, &outbuf);
```

- Only ~1000 bytes in size
- Ports to the very small CPUs [tried ARM]

Quick `n' Easy DER → Data

- `ok = der_unpack (&inder, stx, &cert, 1);`
- `cert` points into `inder`, in an array of

```
struct dercursor {
    void *derptr;
    size_t derlen; }
```

- This array can be overlayed with a pointer to

```
struct Certificate {
     x509_tbsCertificate tbsCertificate;
    AlgorithIdentifier signatureAlgorithm;
     dercursor signatureValue; }
```

Quick `n' Easy DER → Compiler

- `asn2quickder` maps ASN.1 specifications into

  `struct x509Certificate { ... }`

  and syntax/parser bytecode.
- Quick DER comes with RFC includes:

  `#include <quick-der/rfc5280.h>`

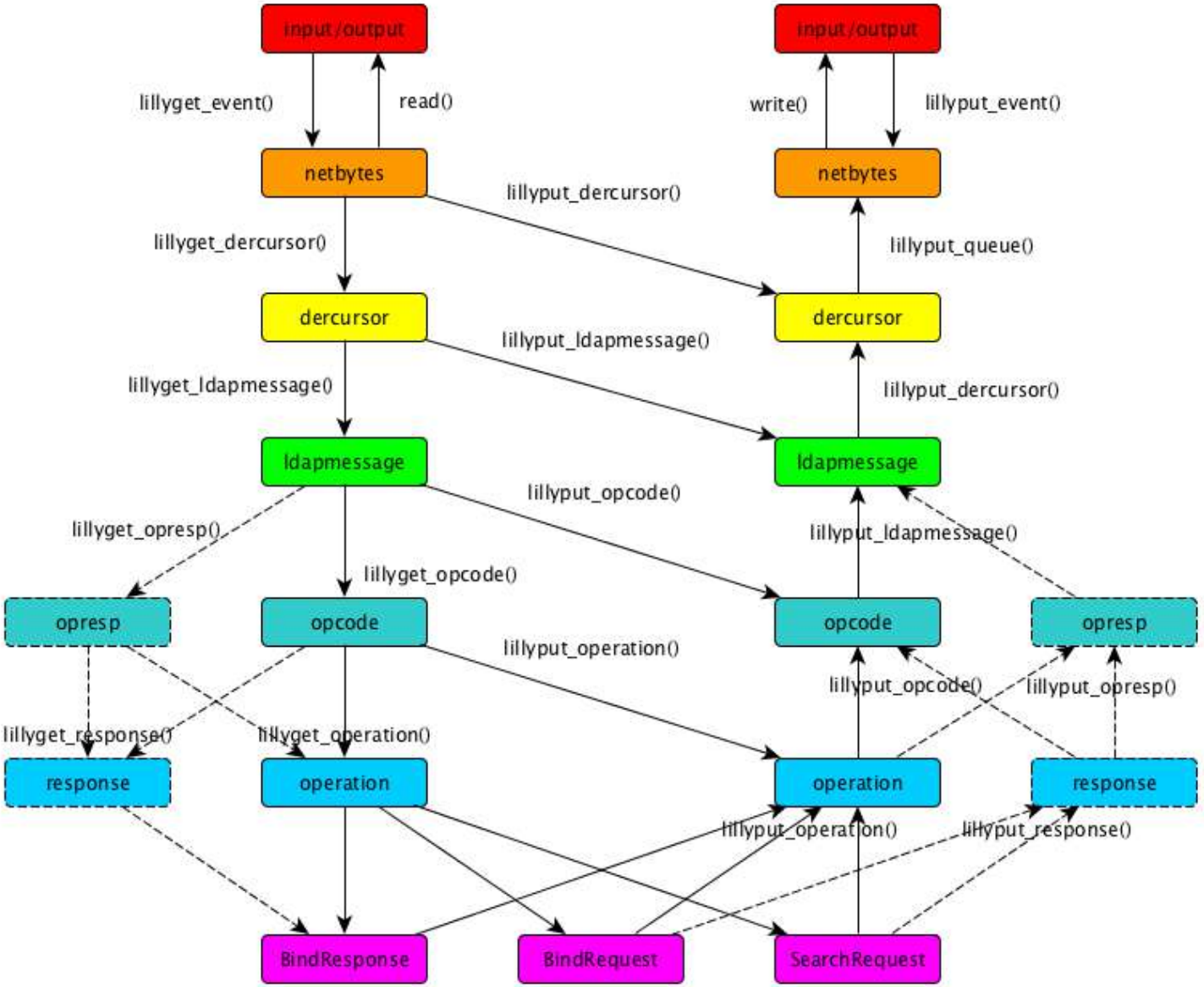
  `struct Certificate cert;`
- Whose dercursor you get to use as

  `cert.signatureValue.derptr/len`

LillyDAP: Little LDAP

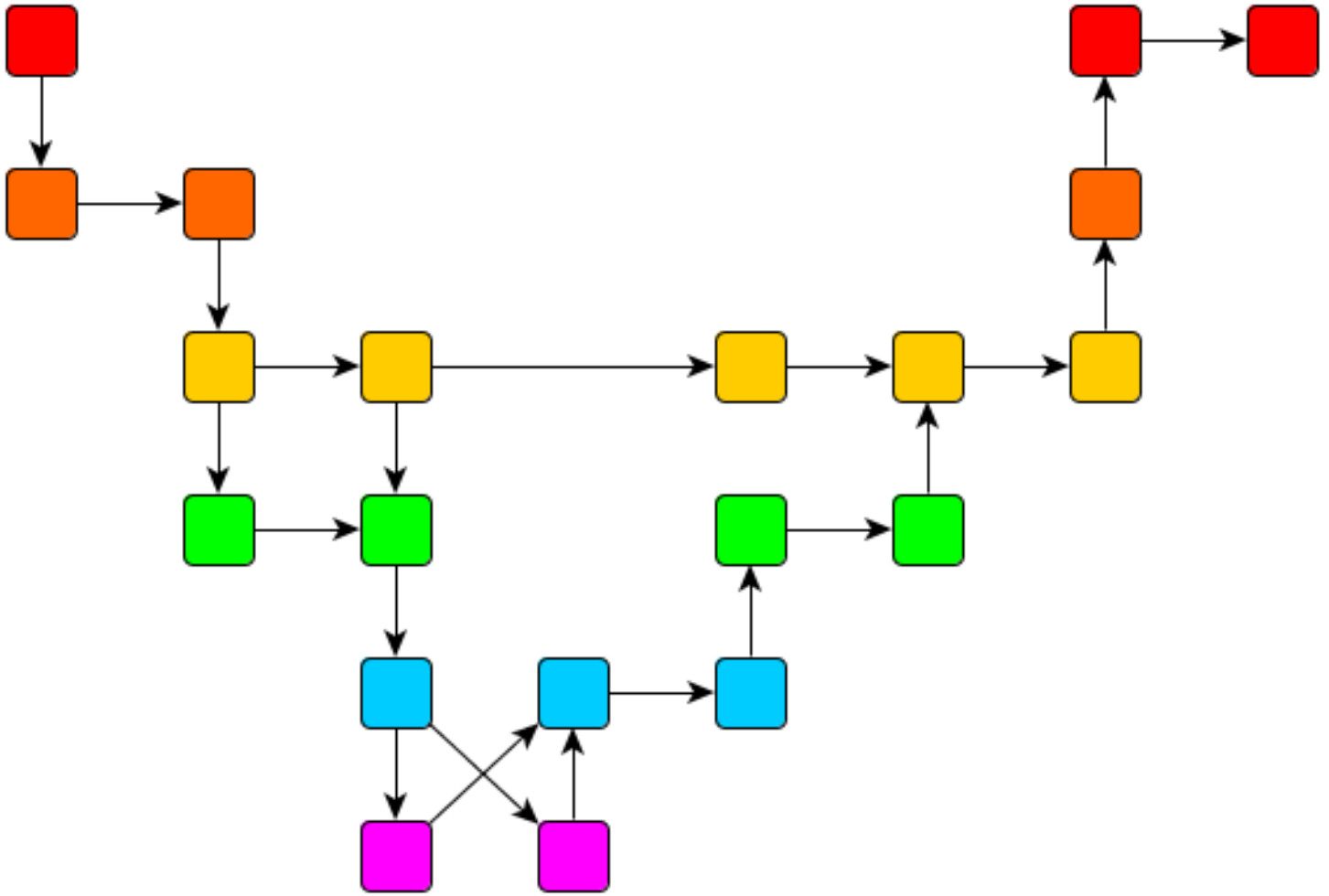
- LDAP builds on BER
- LillyDAP builds on Quick DER
  - Accept BER (liberal in what it accepts)
  - Generate DER (conservative in what it sends)
- LillyDAP is an async API for LDAP
  - Event-driven I/O of byte streams
  - Callbacks for LDAP chunks
  - Various levels of involvement
- LillyDAP lets you build your fantasies

LillyDAP: Little LDAP



ARPA2 project

LillyDAP: Couple as you Like



ARPA2 project

LillyDAP: Examples

- Filter queries
  - Anonymous queries only see certain data
  - Authenticated queries see it all
- Restructure the LDAP tree
  - `ou=People,dc=example,dc=com` goes to backend 1
  - `ou=Machines,ou=Automation` goes to backend 2
- Dynamic lookup of data
  - OpenPGP served from your `~/.openpgp`
  - X.509 from your `/etc/pki/cert`
- Client tools using LDAP

LillyDAP: LDAP made flexible

- LDAP can be used to interrogate dynamic data
- Much more semantics than REST, or JSON
- Extensible without name clashes
- Very small code base (also very young)

- LillyDAP can be to LDAP…

  …what Nginx is to the web!

ARPA2: Now starting

- ARPA2 relies on Quick DER & LillyDAP
  - New tools in our open source kit
- ARPA2 currently builds IdentityHub
  - Domains as a corner on the Internet
  - Control over users, groups, aliases, roles, …
  - Publish public keys over LDAP
  - Rely on existing open standards
- ARPA2 aims to integrate domains
  - *Bring Your Own Identity*

ARPA2: Now hiring

- ARPA2 is funded through InternetWide.org
  - Our external face: blog, funding
  - We actually have some developer funding
- ARPA2 is a network of open source developers
  - LDAP for public credentials (*BYOID*)
  - Authentication & Authorisation
  - User-controlled users, groups, …
  - Key management automated
  - Remote PKCS #11
  - Docker & AMQP 1.0

Further Reading

- https://github.com/vanrein/quick-der
- https://github.com/vanrein/lillydap

- http://internetwide.org (project blog)
- http://arpa2.net (individual projects)

- Rick van Rein <rick@openfortress.nl>