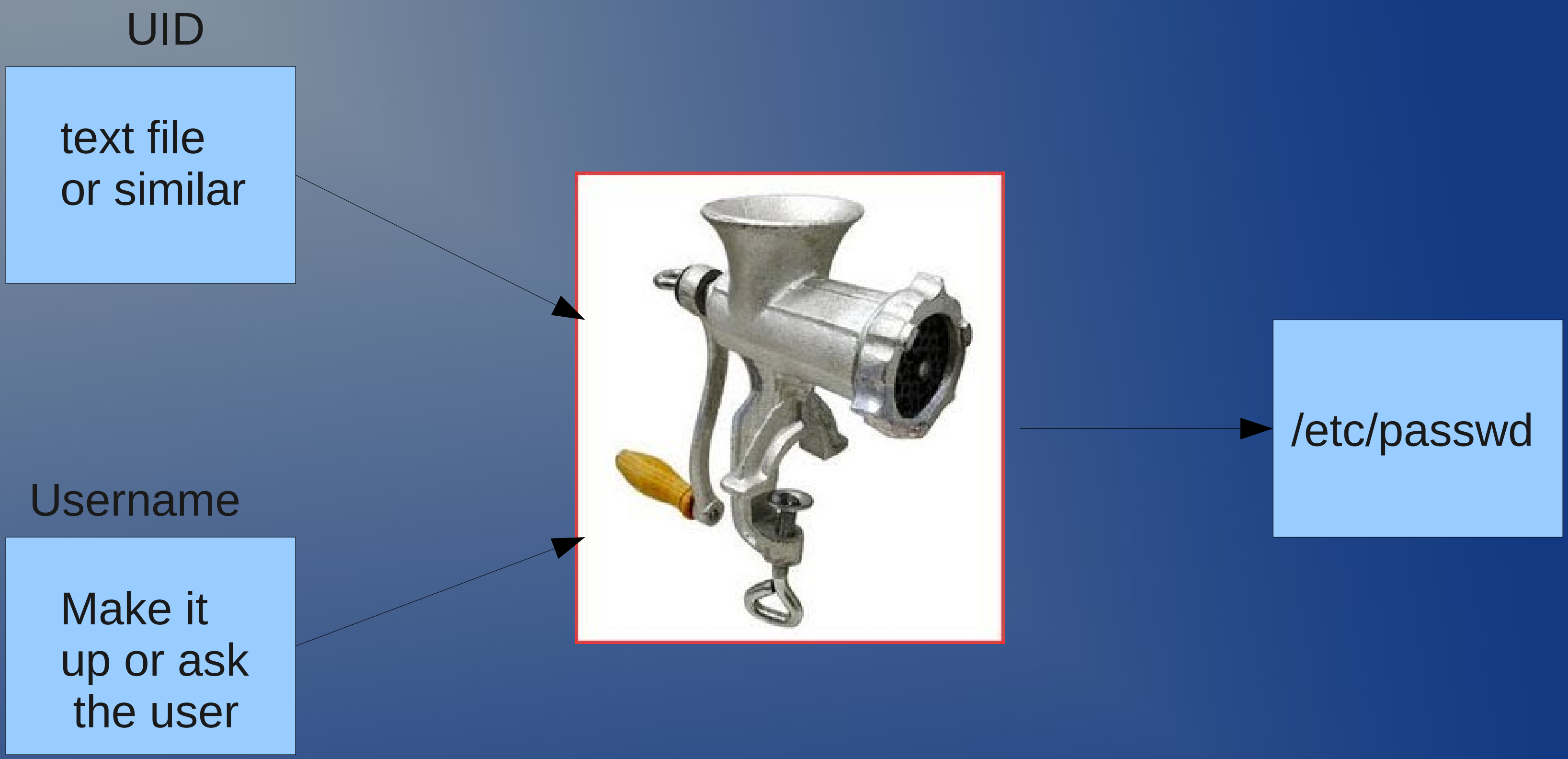
The Handle Turns Itself...

Adventures In Account Management

Toby Blake
Craig Strachan
School of Informatics
University of Edinburgh

The Good Old Days



UID

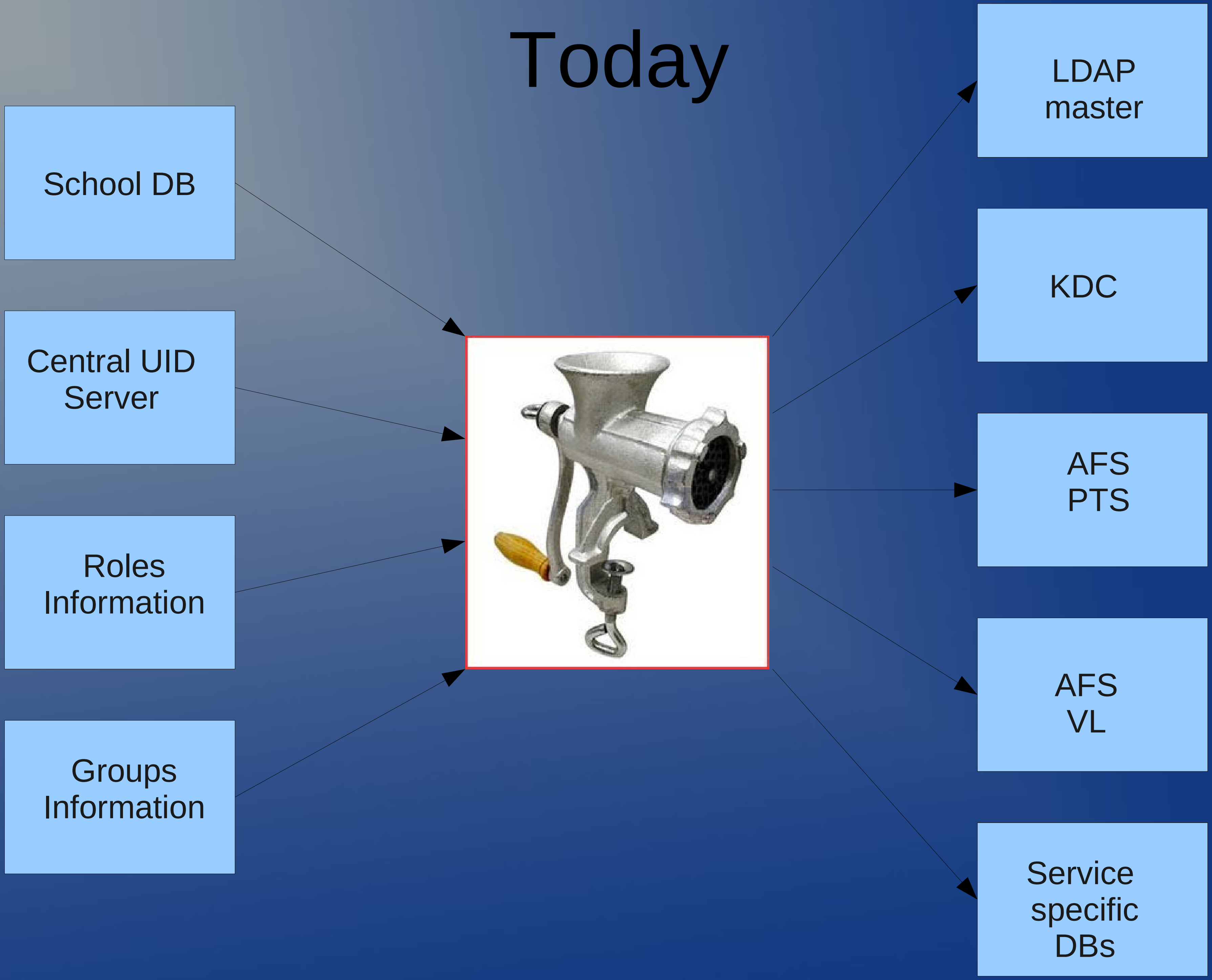
text file or similar

Username

Make it up or ask the user

/etc/passwd

Today



School DB

Central UID Server

Roles Information

Groups Information

LDAP master

KDC

AFS PTS

AFS VL

Service specific DBs

So Let's Automate Things!

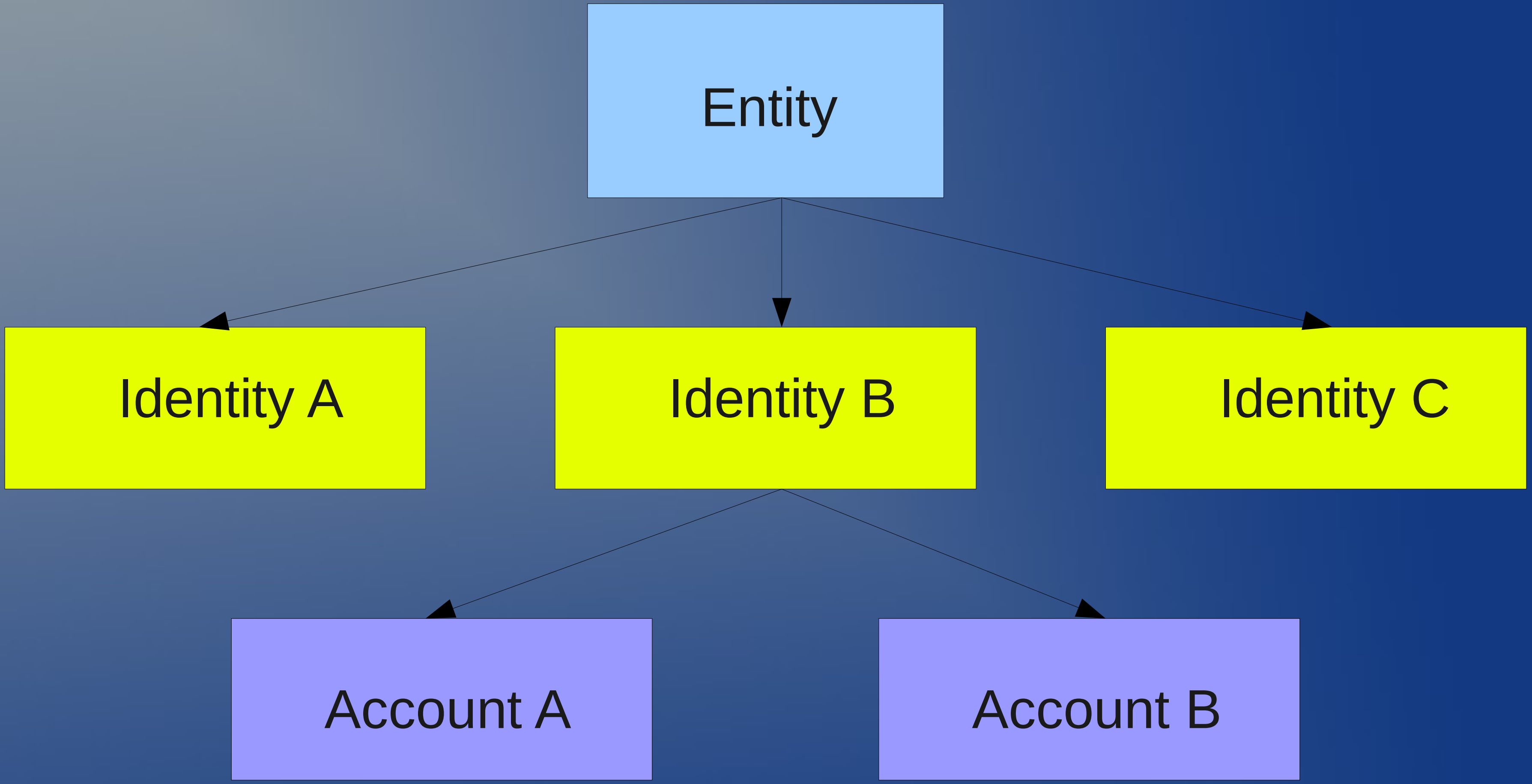
- Over the years, various scripts and manual procedures were written
- This led to different mechanisms to update different things running in different places and managed in different ways
- One of the initial issues when writing the new system was to work out just what needed to happen where it needed to happen and when it needed to happen
- This was difficult to extend and made managing the account lifecycle (archiving/deleting etc) impossible
- Something better was needed leading to...

Prometheus!

Prometheus

- Written in OO Perl using Moose
- Uses LDAP for its central database (Olympus)
- Models *entities* which can have *identities* which in turn can have *accounts*
- Makes heavy use of *roles* and *entitlements*
- Does its work using *stores* and *conduits*

The Entity Model



Entity

Identity A

Identity B

Identity C

Account A

Account B

Roles and Entitlements

- A *role* is a function that an entity fulfils or a position it holds:
  - person
  - staff member
  - computing officer
- An *entitlement* is something an entity can do:
  - log into a School machine
  - access the staff-only part of the School web site
  - become root on School machines
- Roles contain entitlements, negative entitlements and other roles

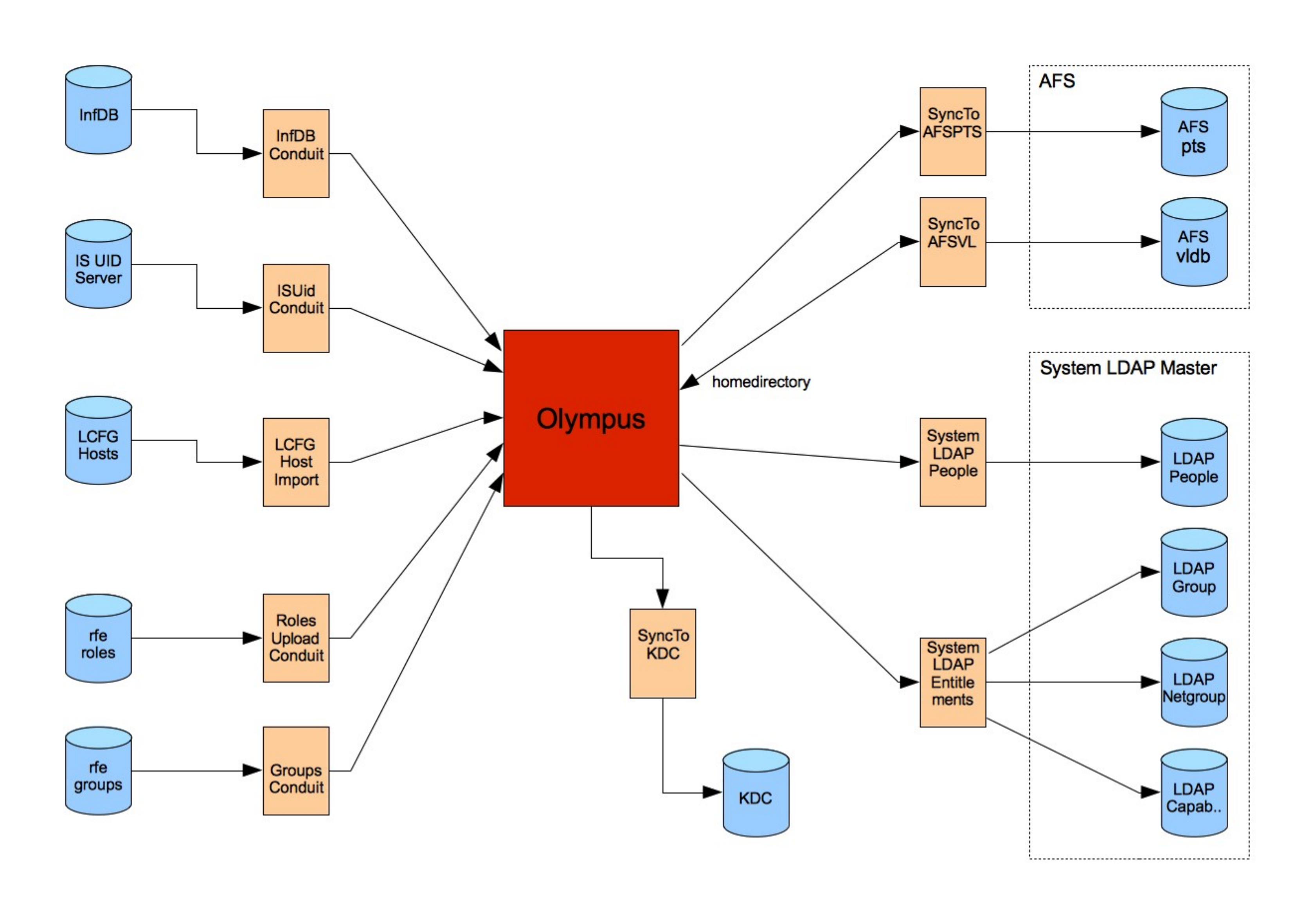
The Store

- A store is an abstraction of a source of data
  - LDAP database
  - KDC
  - External datasource
- A store provides methods to
  - enumerate the entire contents of the store
  - fetch, add, update and delete individual store objects
- A store may also do other service specific tasks
  - the KDC store manages Kerberos password setting/modification

The Conduit

- Manages the flow of data between stores
- Two modes of operation:
  - audit – reports what changes a sync would make and any issues found with the data
  - sync – actually changes the data
- Conduits can do other things
  - the AFS VL conduit decides which server and partition a new user volume should be created on
- Conduits can be run individually but in practice are normally run in a continual loop
- A conduit run can also be triggered by an event of some sort.

In all its glory



The Kerberos Store and Conduit

- Store uses Authen::Krb5::Admin to communicate with KDC
- Conduit interates through all identities and makes any necessary additions and modifications to the KDC
- Should do deletions as well but not yet
- Benefits – new principals are created with ~allow_tix. The principal isn't activated until the password has been set with prometheus tools.
  - at start of term students can set password using web interface – much better than old system
  - If necessary, can do blanket or selective disabling of principals until password has been changed. This has been necessary in the past.

AFS Stores and Conduits

- Stores and conduits for managing AFS PTS and VL databases
- Mostly uses AFS::* perl modules but resorts to shell commands in some places
- PTS entries and user volumes are created automatically for identities which qualify.
- server/partition is allocated by VL conduit from hand maintained pool of available partitions.
- roles/entitlements control which kind of partition volume is created on
- VL conduit takes care of creating all enclosing volumes, mountpoints etc

AFS Stores and Conduits – Improvements and Additions

- Quotas currently are managed automatically but outwith Prometheus. Would be simple to add using new store/conduit
- Better way of managing partition pools
- Automatic load-balancing of servers by moving volumes around whilst meeting partition requirements etc

And so the handle turns by itself

- All the constituent parts of our user accounts are now generated automatically once the system has the information it needs
- For example, every August data for hundreds of new student accounts appears in the various source databases
    - Prometheus takes the information from these source databases and creates all the necessary entries in the downstream databases for these accounts
    - The students are mailed automatically to tell them that their accounts are ready
    - The students set their initial password via a password portal using Prometheus tools
    - All of this happens without significant sysadmin input.

The future

- Full management of the lifecycle of an account
- extensions and additions (AFS quotas, mailing list …)
- Web interface
- refinements/enhancements to roles and entitlements
- ...

More Information

https://wiki.inf.ed.ac.uk/DICE/PrometheusOverview

Questions?