Prometheus Overview

June 2011

Toby Blake <toby@inf.ed.ac.uk>

Account Management

- What is account management?
- What are we managing? Accounts, Identities?
- People and other Entities (machines, agents)
- AM system sits between central databases and our services
- Ensures changes flow from databases to services

Where We Were

- Central databases (IDMS, school db, LCFG, …)
- LDAP, KDC, AFS pts, AFS vldb, …
- Also, per-service databases (jabber, …)
- Centralised vs decentralised
- Some automation (db to ldap)
- Manual account creation
- Some adhoc manual updating

Problems

- Fragmented management
- Deletions often didn't occur
- Limited support for different account types (machines, lightweight accts)
- Difficult to extend for new services
- Difficult to detect problems
- Heavy support burden at times (new intake)
- Too centralised

Introducing Prometheus

- Simon wrote “Musings On Accounts” - September 2007
- Further design 2007 onwards
- Coding started 2009
- System went live August 2010
- All accounts created and modified automatically since then

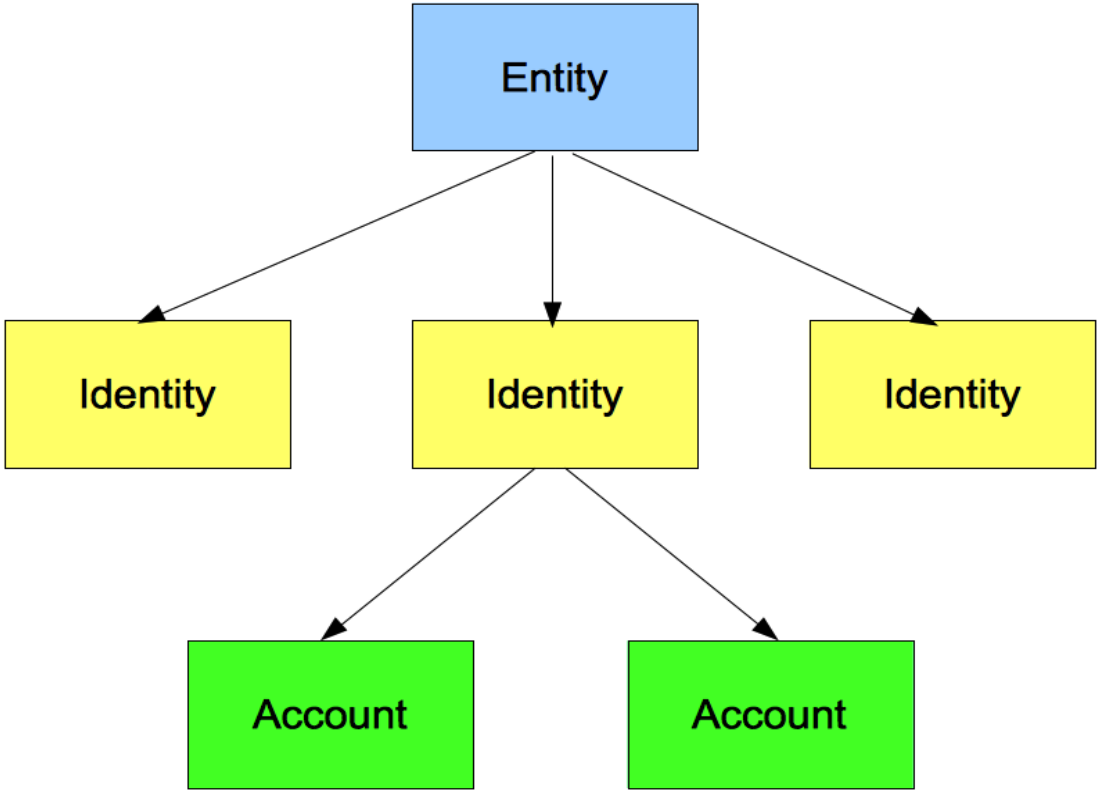
Roles/Entitlements

- We've used Roles and ~~Capabilities~~ Entitlements from the beginning
- Roles describe functions that a user performs
- Entitlements determine what a user can do
- Roles contain entitlements, negated entitlements and other roles
- Very powerful and flexible

Account/Identity Modelling

- Traditionally one person has one identity
- This isn't flexible enough
- We need and use multi-faceted identities
- Need to manage entitlements for these different identities
- Some need more access (me/admin), others need less (me/cron)
- Where do 'accounts' fit in?

Prometheus: Entity Tree Store



Prometheus: Datastores

- A datastore lets us talk to sources of data
- These can be databases, ldap, files, kdcs, almost anything
- Generally, they consist of stores of objects
- Provides a consistent interface to read, modify and delete objects
- Can provide access to all data or a filtered subset

Prometheus: Conduits

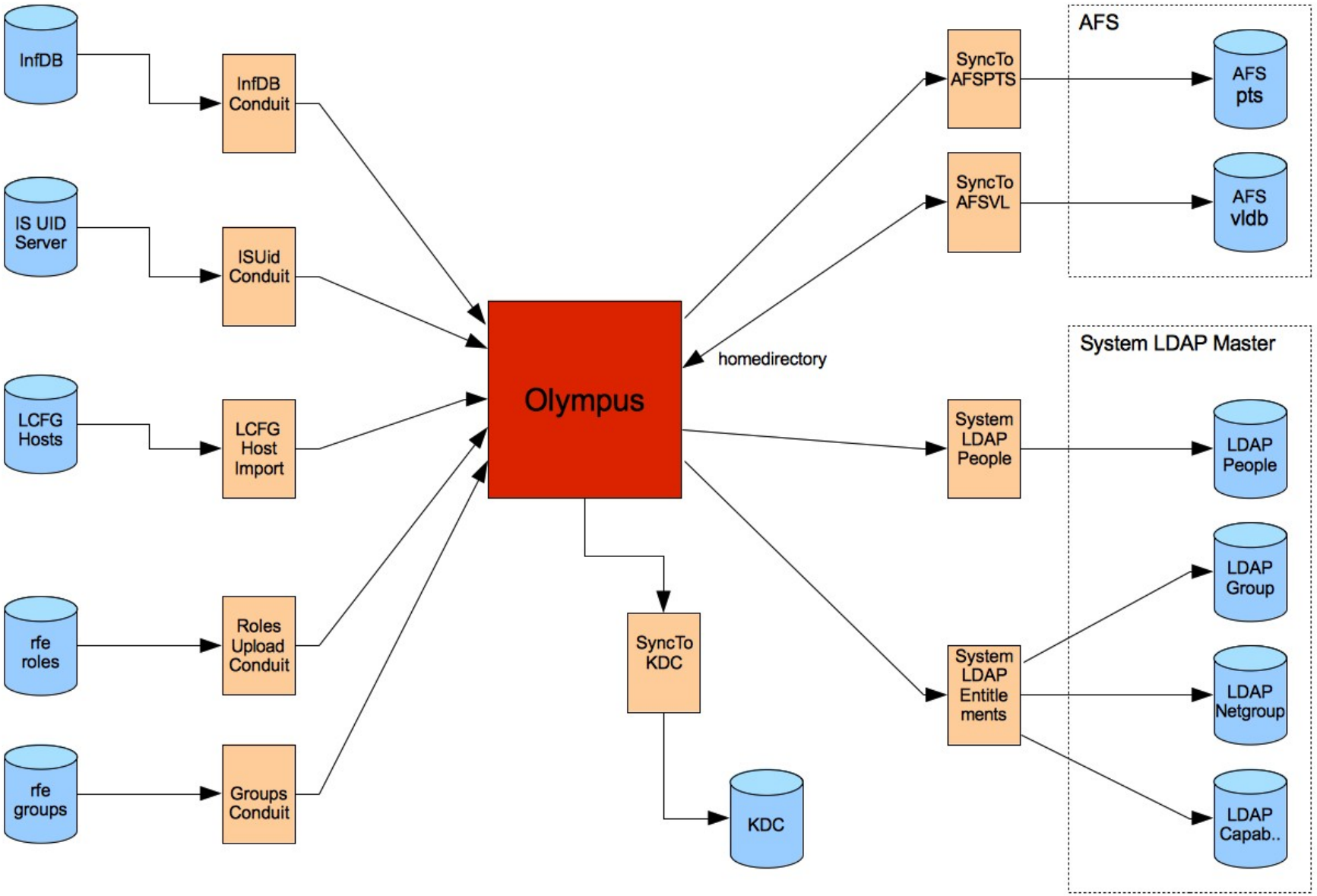
- Conduits connect datastores
- One-to-one is typical, but can also use a single store or many stores
- Conduits are responsible for flows of data
- They manage any necessary data transformation
- Currently run in a loop, but can be event-driven
- Can also report errors

Prometheus: Auditing

- How do we find out about data problems (e.g. uid mismatch)?
- Conduits do auditing as well as synchronisation
- The audit can do anything, but typically follows the same logic as the sync
- Reports the changes that a sync would make
- Invaluable in tidying up our data

Prometheus: Olympus

- Olympus is the name given to the set of stores representing prometheus's data
- Differentiates from Prometheus as a wider system
- Includes entity tree store, roles, groups, audit errors, conduit configuration, etc.
- All help in openldap directory


InfDB
InfDB Conduit
IS UID Server
ISUid Conduit
LCFG Hosts
LCFG Host Import
rfe roles
Roles Upload Conduit
rfe groups
Groups Conduit
Olympus
homedirectory
SyncTo KDC
KDC
SyncTo AFSPTS
SyncTo AFSVL
AFS
AFS pts
AFS vldb
System LDAP Master
System LDAP People
System LDAP Entitle ments
LDAP People
LDAP Group
LDAP Netgroup
LDAP Capab..

Where We Are Now

- Central store of data
- Decentralised management of stores
- Consistent data flow
- All user accounts automatically generated
- Support for multi-faceted identities
- Support for different account types
- Highly extensible architecture
- Auditing

Where We're Going

- Full automation of entire account lifecycle
- Users should be able to manage their IDs
- Users could bring up their own services
- Better audit/error management
- Data tidy-up
- Roles tidy-up
- More conduits
- General tidy-up, packaging, etc.

Technical Details

- All prometheus data stored in openldap directory
- lcfg-prometheus manages slapd
- LDAP schema for prometheus objects automatically generated
- All code written in OO Perl, using Moose
- Developed using git/gerrit
- Comprehensive test suite

Problems/Issues

- It's taken longer than originally anticipated
- Our data was/is a mess
- Main conduit run needs to be speeded up
  - delta syncrepl
  - Parallelisation
- Knowledge needs to be spread

More Information

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