

Running List: Collab Stuff

Framework – Services - Appliance

Next Steps - Appliance

Next Steps - Service

- Internet2 to put together a service instance in skunkworks, using previous code, Shib, Grouper, etc.
- Dutch to do the same, using...

Some Next Steps- Marketing/PR

- Graphics/slides for comanage as a lightweight collab platform linked to the waffles of federation... - Surfnet
- Graphics/slides for comanage refactored for enterprise or federation level deployment
- Video of the four types of users using comanage/apps
 - Ala <http://www.jisc.ac.uk/whatwedo/themes/accessmanagement/federation/animation.aspx>

Next Steps - Process

- Figure out groups – dev, -community, -??
- Figure out communication modes and times for each group
- Work out the international dimensions in dev and in apps

Next Steps- Framework

- Understand scope of framework – what concepts included/excluded
- Understand, for concepts within scope, the specifications linked to the concept
 - Specific instance – domestication? Is the externalization LDAP or SAML or...
- Application service registry issues (coordination of domestication)

More steps- framework

- Identify good candidates/work-arounds for the missing elements of the model, e.g. STS, provisioning
- Open up an account linking discussion

Current Comanage Materials

Positioning CManage

- Cmanage is not intended as an enterprise-class approach, though many enterprises and federations may well deploy large numbers of instances or a “refactored for industrial use” implementation
- Cmanage is intended as a collaboration-class approach that works well and sustainably with enterprise, federated and interfederated infrastructure
- Collaboration-class means lightweight in scope of services commonly managed (just IdM), minimal application requirements, easy implementation options (for example as a collaboration support appliance offered in a cloud), lack of enterprise oriented features (such as a full ESB), etc.
- Works well and sustainably with enterprise, federated and interfederated infrastructure means that Cmanage can easily and gracefully link Cmanage and federated accounts, work with data feeds from enterprise services, be refactored to leverage different types of infrastructure, etc.
- A lightweight collaboration support approach that integrates with deeper infrastructure

Types of folks

- sysadmin - install a new instance of the software, and configure it
- service mgr -- (the person managing a CO instance, which may contain multiple COs)
 - create a new VO, grant mgmt priv's to others
 - archive the content of a VO -- phase 2
 - remove a VO definition
- Collabadmin -- manages a single specific VO
 - manage the home page of the CO
 - enable a new service to support the work of the VO; use the mgmt GUI of the specific application to assign priv's within the application
 - manage a group (create a group, manage membership, delete the group)invite a person to join the VO (automatically adding person to some groups
 - collabadmin -- manage each service (these actions would be done within each service)
 - configure a service (eg this email list should be associated with ldap group X
 - manage permissions within a service (eg GROUP X can do ACTION Y)
- power user -- PI, with some enhanced functionality, a subset
- end user—
 - go to the "home page" of the CO—
 - navigate to services supporting the work of the CO-- (service specific;
 - uses GUI provided by application) various actions within the services supporting the work of the CO, depending on how privileges have been assigned

STS services

- {K, SAML} in, GridShib cert out
- Pubcookie in, SAML out
- Authn in, dedicated user/pwd out
- SAML token in, webcookie out

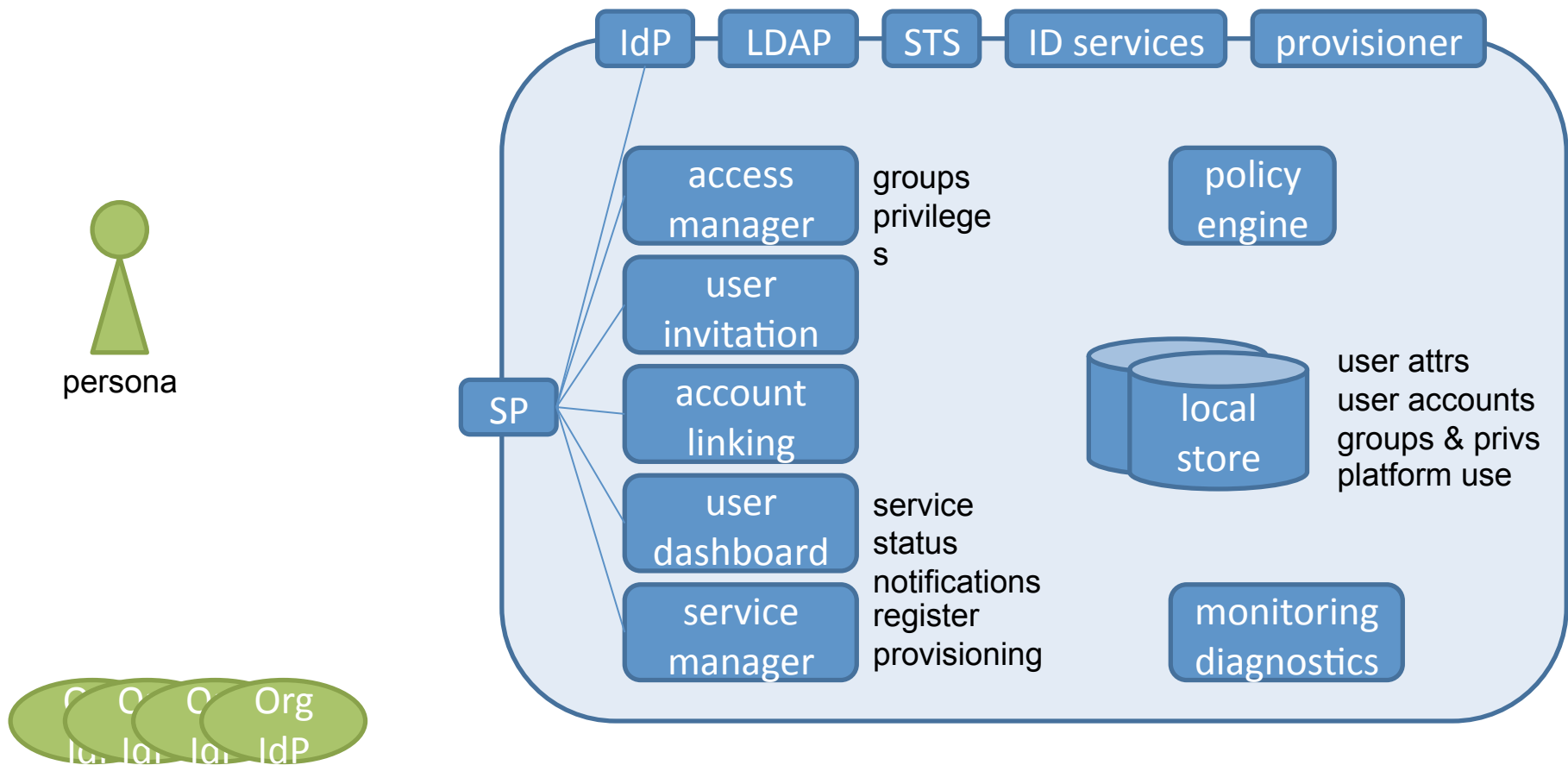
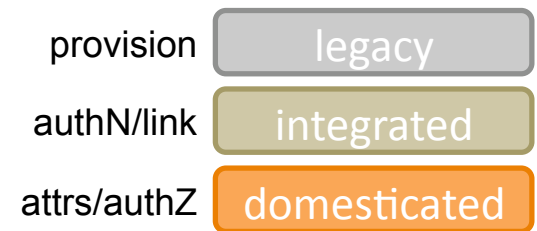
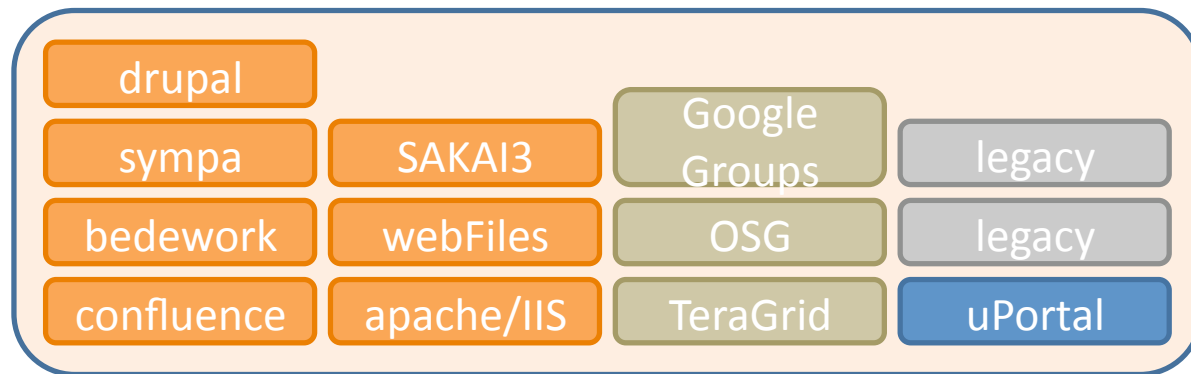
- Binding the framework to app development environments:
 - At what level does stuff need to be specified
 - Which development environments
.NET, php, Apache
 - Who will write the services

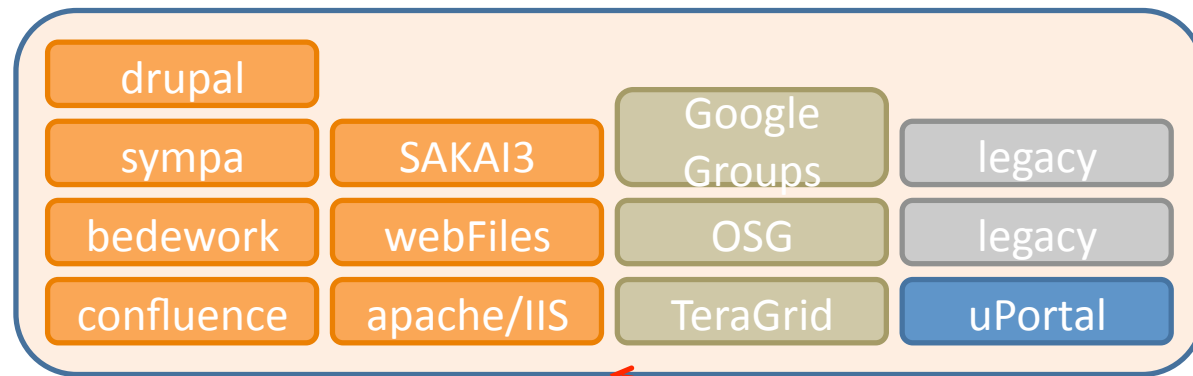
Framework -1

- Several different but consistent perspectives, for different audiences –
 - CIO (block functionality flows)
 - Apps developer – (API's, services, etc)
 - User (user workflows, for different types of users)
 - Others?
- Framework also has layers
 - Language and tech specs
 - Data and metadata specs (to follow later)
 - Others?

Block flow framework parts

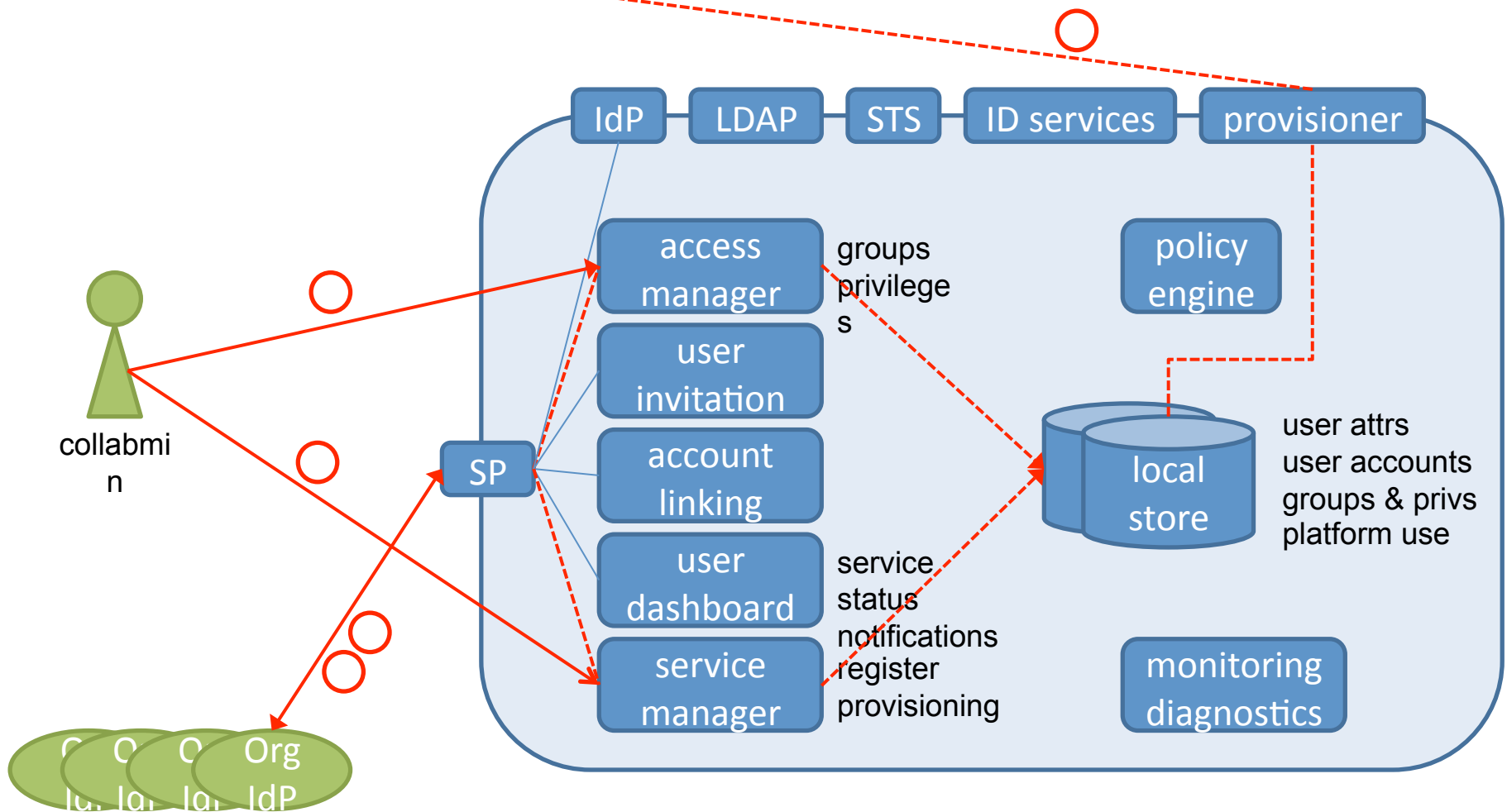
- A local datastore
- STS (security token service, aka credential convertor)
- Provisioning/deprovisioning into local store service
- An account linking mechanism
- Group and privilege manager (represent as unified for now)
- SP stub
- Local IdP
- Invitation engine
- Plug and play service for apps that want it
- Attribute services (?)
- Policy engine
- System monitoring and diagnostics
- User dashboard that includes a user collaboration data feed service





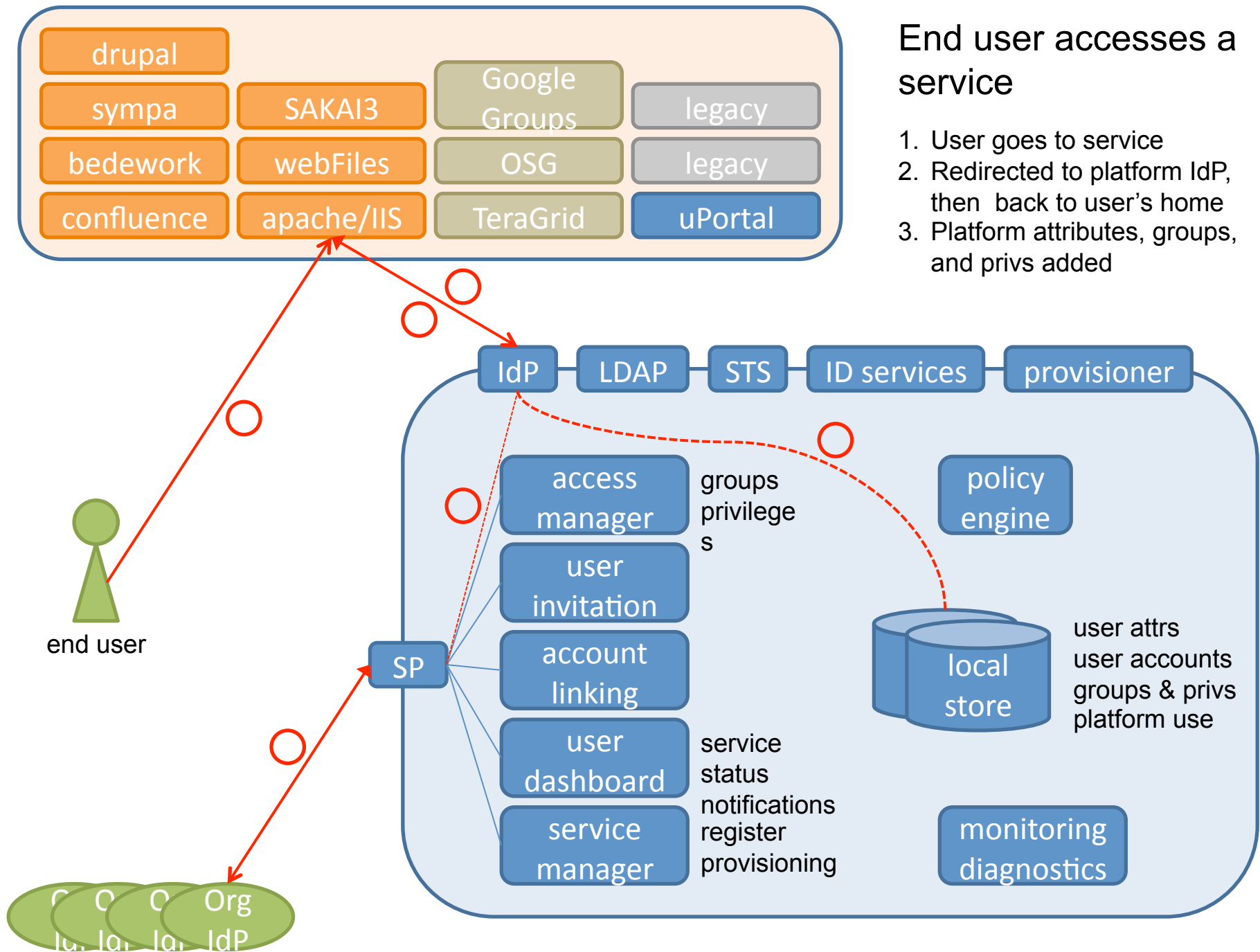
Collabmin adds a new CO to the platform

1. Create group, assign Admin to power user
2. Allocate service resources



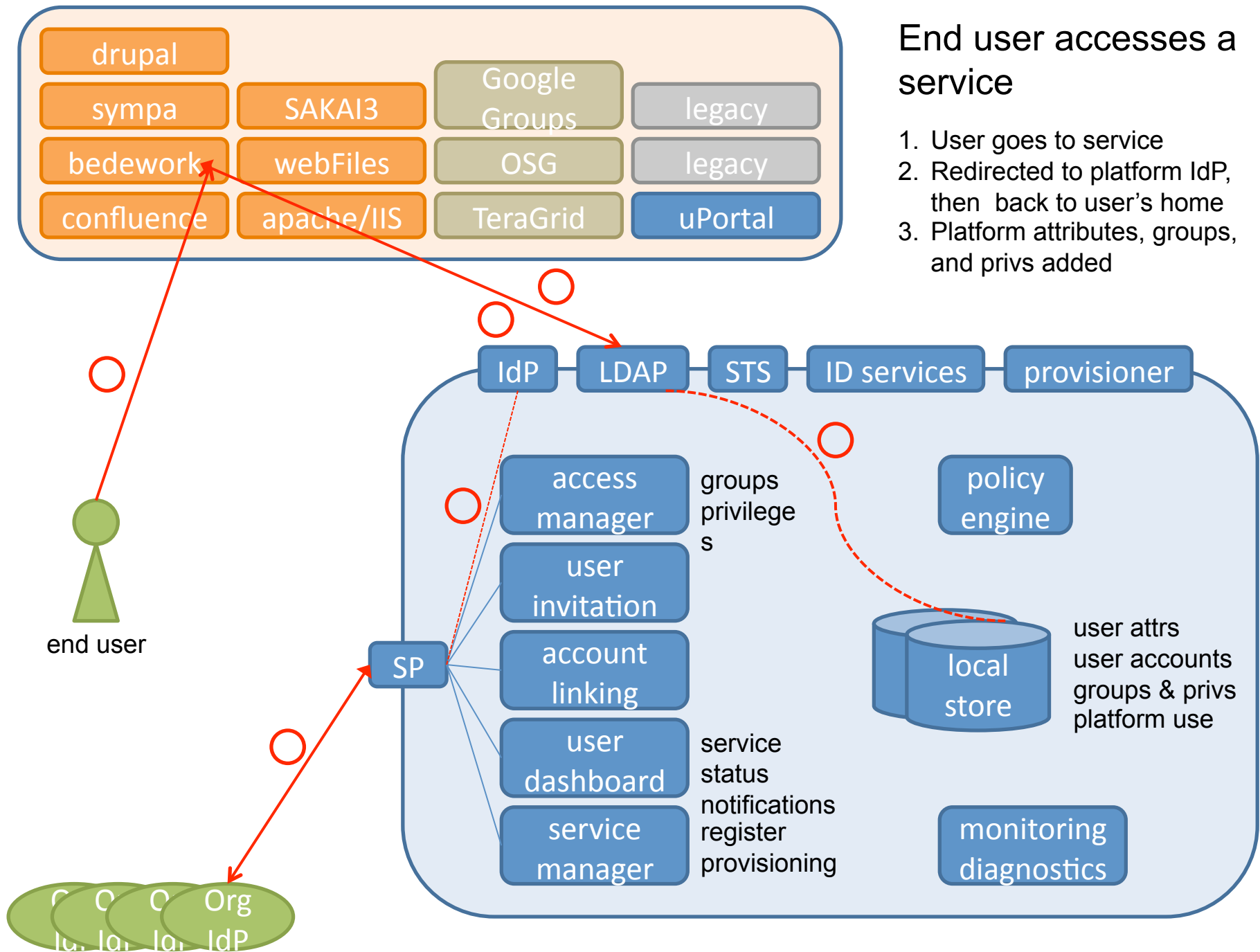
End user accesses a service

1. User goes to service
2. Redirected to platform IdP, then back to user's home
3. Platform attributes, groups, and privs added

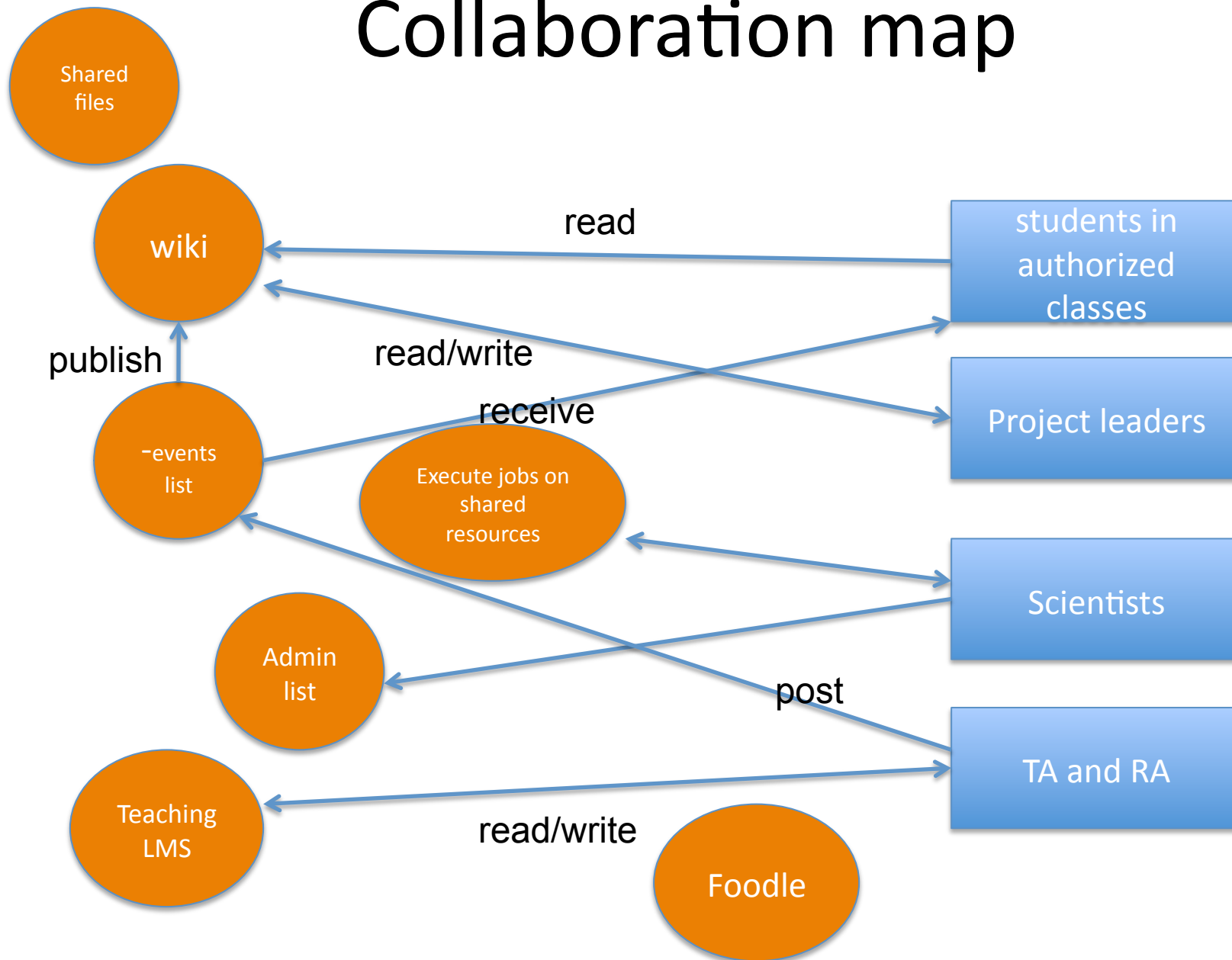


End user accesses a service

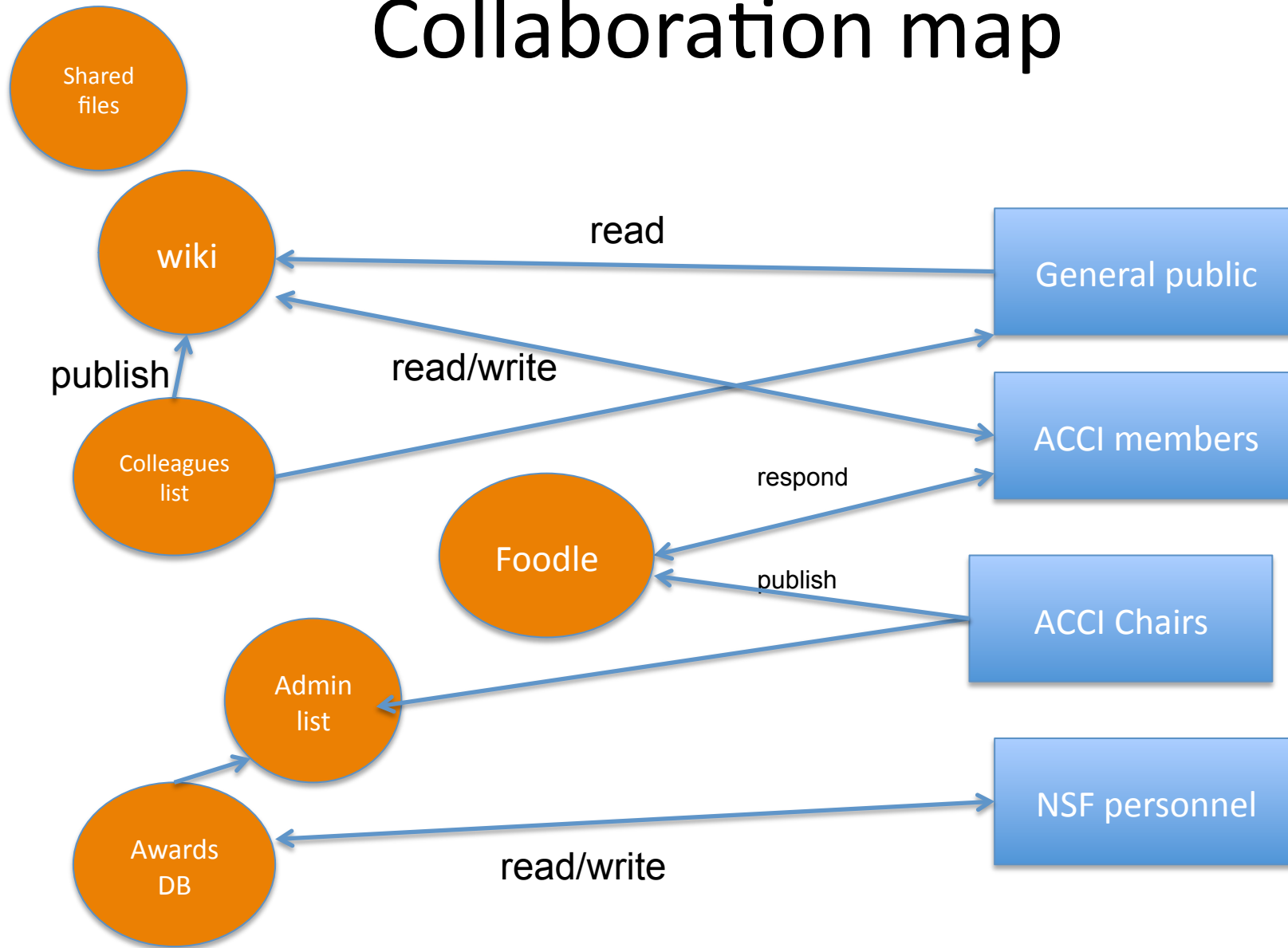
1. User goes to service
2. Redirected to platform IdP, then back to user's home
3. Platform attributes, groups, and privs added



Collaboration map



Collaboration map



App developer framework

- Two types
 - Stand-alone app
 - Apps written in an application development environment, e.g. .NET or Spring or...
- Make clear that app data stays in app, not in comanage
- Presents a set of services – which ones

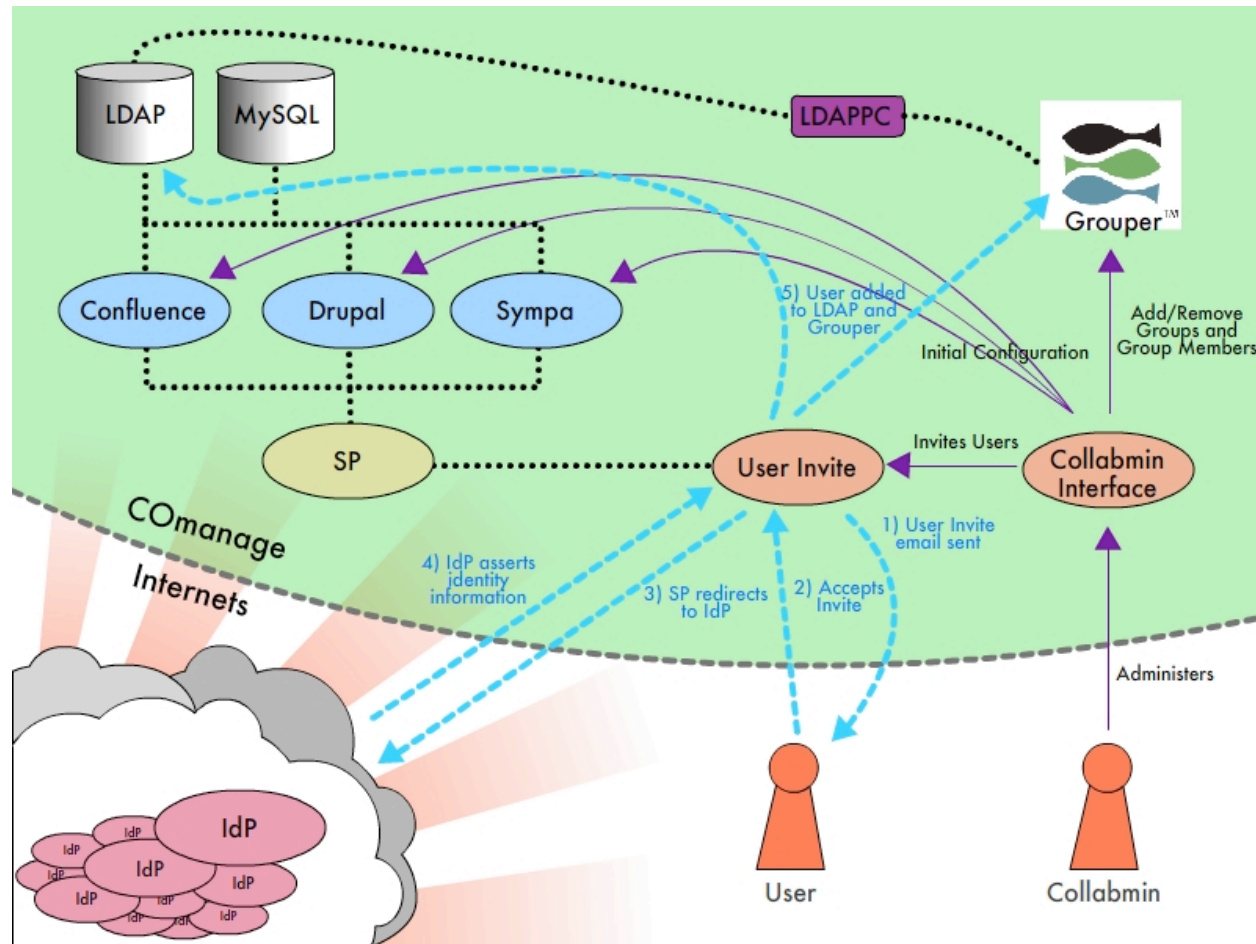
App developer framework

- Services provided are:
 - Authn
 - Authz (Y/N/?)
 - Attributes for app needs
 - Provisioning (?)
 - Some kind of monitoring
- Services explicitly not provided are:

How do apps get info

- Push into legacy apps
- Domesticated apps ask for it
- Domesticated apps need to speak LDAP or SAML or generic STS

Flows



Refactoring CManage

- Right word for the concept?
 - Unbundling, debinding, distributing
- What are likely refactorings?
- What connections need to be in place among refactored pieces

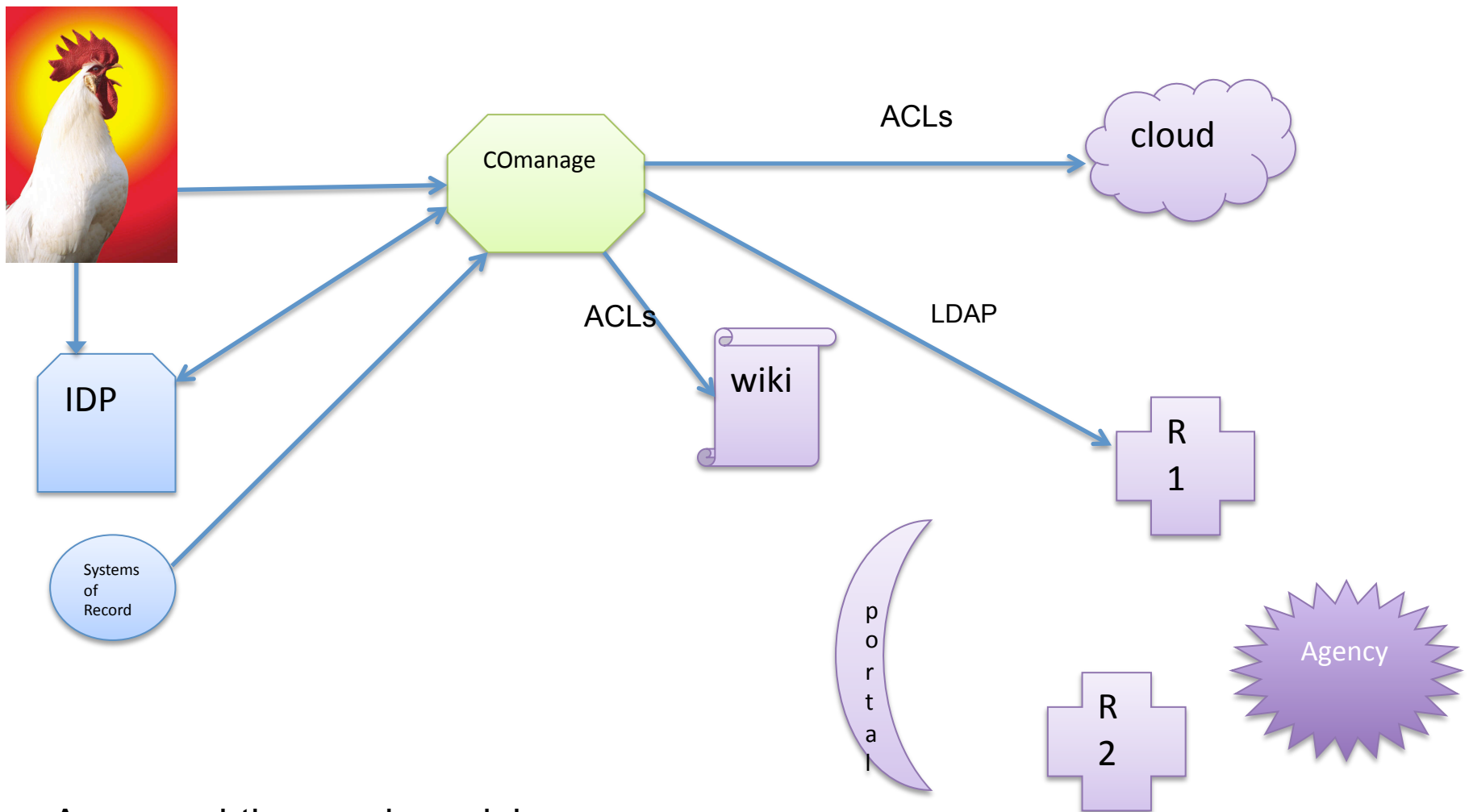
Parked issues

- Discussion of how to share the work of domesticating apps
- Cutover issues for existing VO's, and type of collabs to target for appliance, etc
- Domesticated Zimbra - a lot of us are interested in it and claim to have connections with the company
- How might the appliance and an RSS feed offer a "collaboration stream"
- Maintaining a base level appliance
- Setting a new time for the COmanage dev calls
- Assess the viability of the existing appliance code base

More parked issues

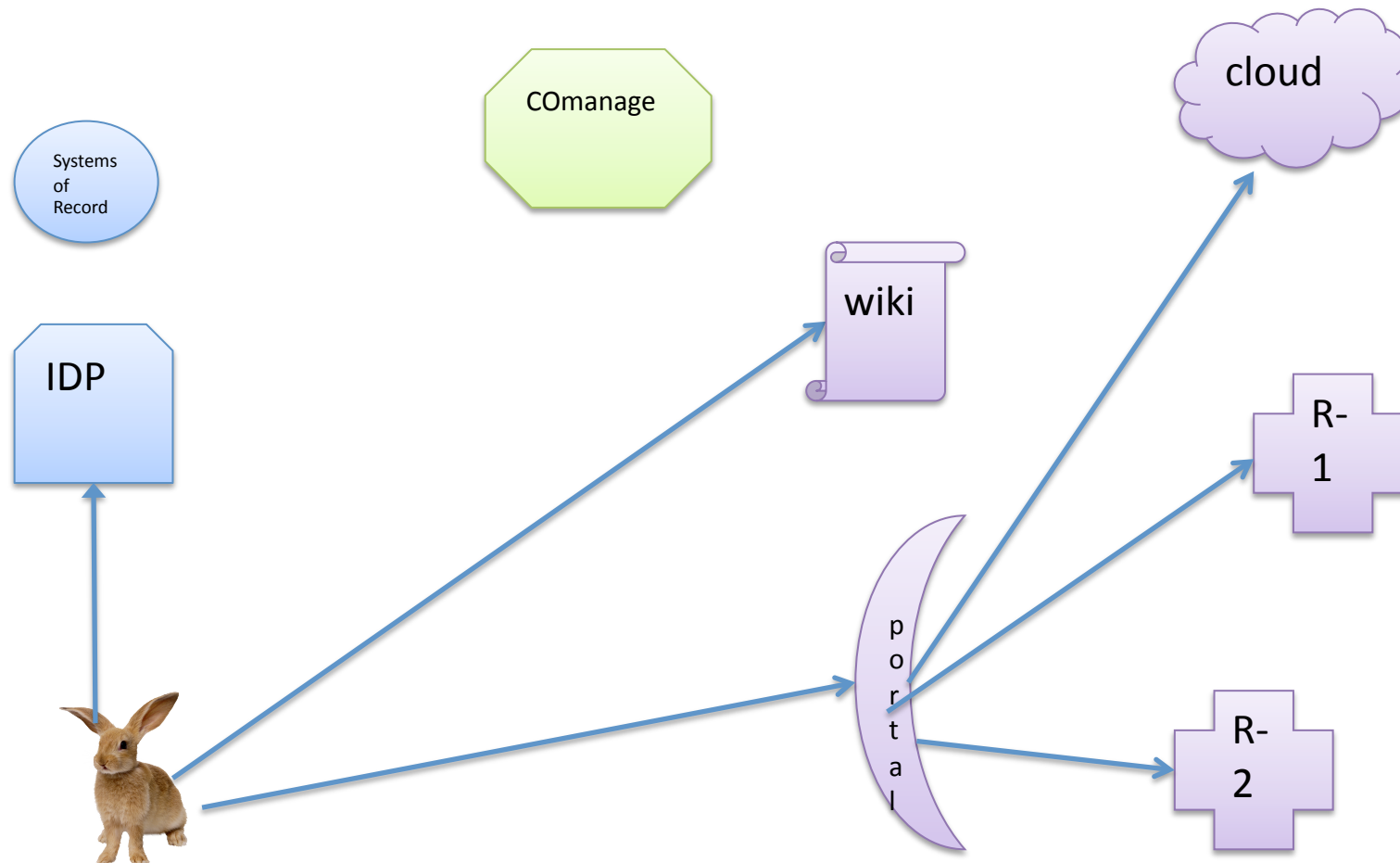
- VOMS comparison/integration
- Licensing issues
- Application check-in services
- Developing use cases
- Is the proper technical phrasing “claims-aware”, “STS aware”, externalized or something else
-

Attribute Flows- Scientist



A non-real-time push model

Attribute Flows - Student



A real-time pull model

Important Characteristics of Flows

- Across trust domains
- Semantic control enforced at COmanage

1. Rabbit accesses CO in usual fashion, login via Shib

2. Rabbit clicks button on CO front page to access Science Portal (flow shown)

