Federated Identity Providers

and the Ipsilon project

Simo Sorce
Sr. Princ. Sw. Engineer, Red Hat
2015/02/06

What is Federation ?

In a nutshell:

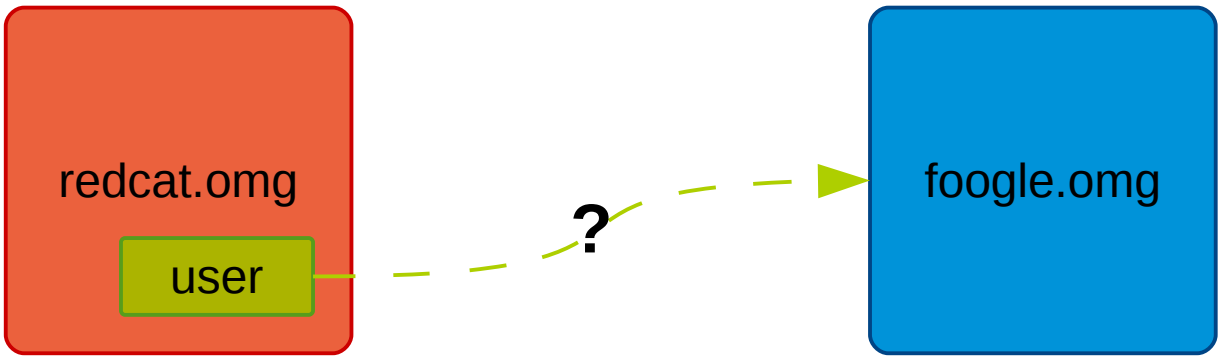
Dealing with users that you do not control on your own.

To do that you need to trust a third party

Trusting a third party

An organization wants to offer services to another which “owns” the users identities.

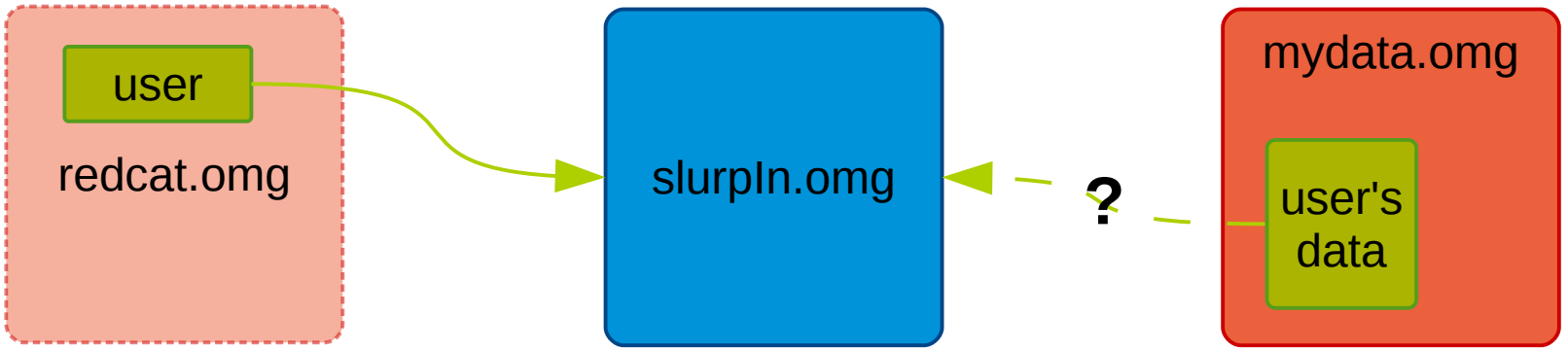
- User's org controls what is disclosed about the user
- User does not need to know additional credentials
- Third party does not need full view of the users store



Trusting a third-third party

Federation is also used when another party need access to data *on the user's behalf*.

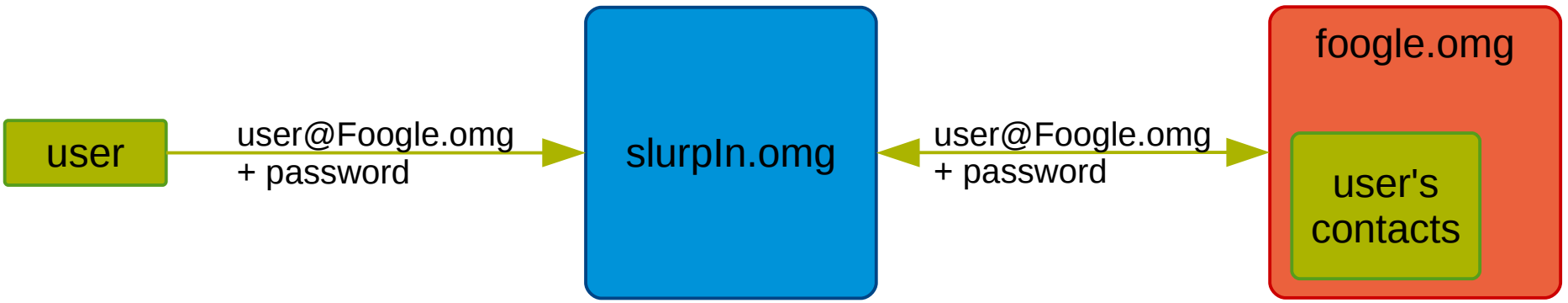
- Also know as *delegation*.
- The third party only get access to specific user data
- The user/org. is in control of the permissions granted



Not Federation

Surrendering credentials is **not** federation.

- User's org. has no control, breach of privacy.
- User has no control on what the 3rd party will actually do with the credentials.
- 3rd party has liabilities it shouldn't want.
- No Single-sign-on.



Federation protocols

Most federation protocols are web/HTTP oriented

- Some authentication flows depend on a user sitting in front of a browser
- Non-interactive modes are available in some cases
- Delegation modes are non interactive (but may depend on interactive modes for setting up the delegation)

To name a few:

- SAML, OpenId, OpenId Connect, Persona, ...

How does it work ?

Glossary

Identity Provider

- Server that authenticate users
- Or provides enough data to verify an authentication assertion

Service Provider / Relaying Party

- Server that needs authentication by a third party Identity Provider
- The system the user is trying to access

    (directly or indirectly like in the delegation case)

SAML

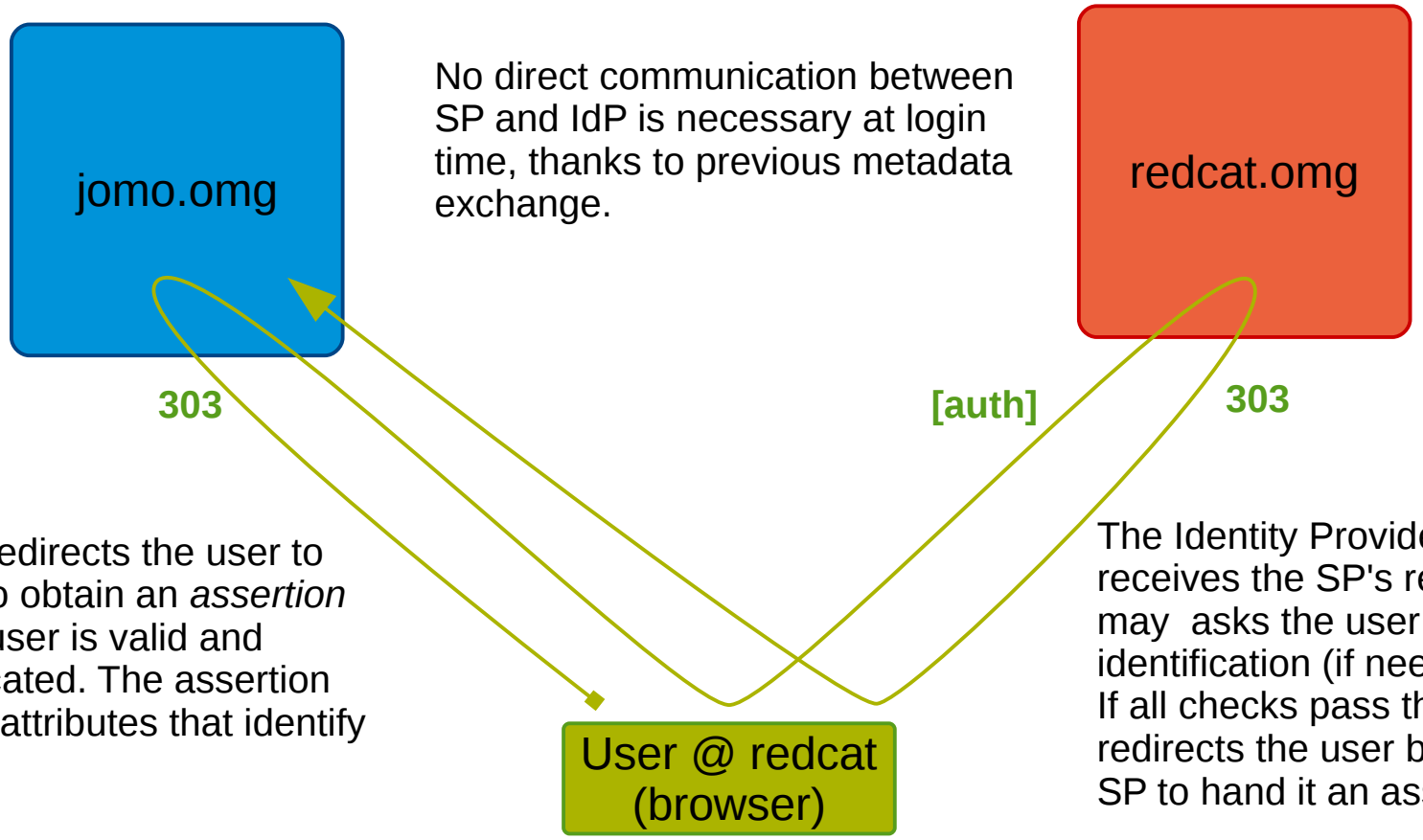
Key aspects:

- Requires agreement between parties
  - exchange of metadata and public keys
- The Identity Provider can choose what data to send
  - third parties receive *assertions* with *attributes*
  - Data can be encrypted
- Single-sign-on friendly
  - Support also single-logout and administrative logout
- Enterprise oriented
  - Based on XML and SOAP on top of HTTP
  - Spec by OASIS

OASIS
Advancing open standards for the information society

SAML

Example auth flow:



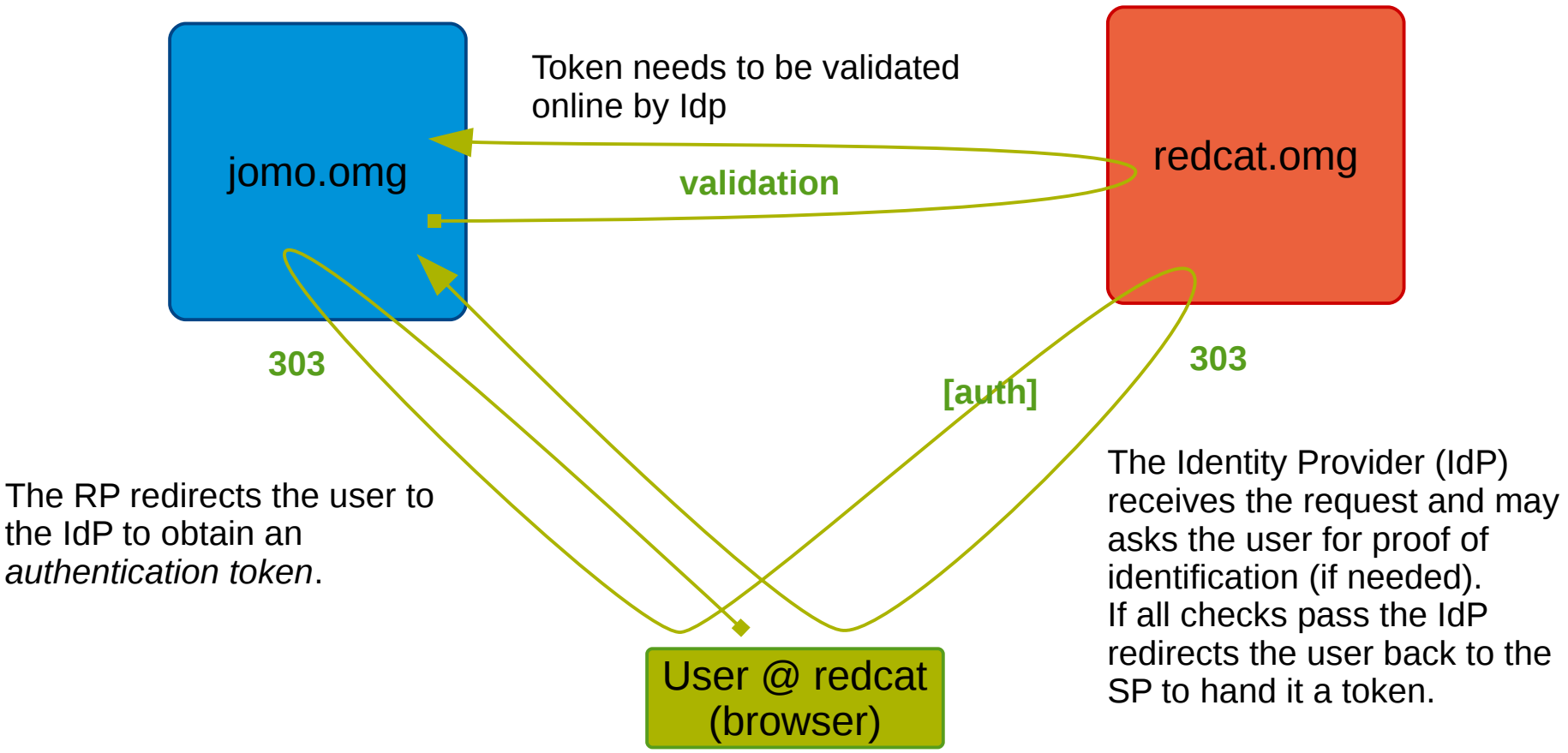
OpenID Connect

Key aspects:

- Supports user-driven consent
  - Users may be allowed to tell the IdP to trust arbitrary third-parties (Idp does not need to trust the RP)
  - Users can be allowed to decide whether to allow or deny authentication requests and what data to send
- Completely different from OpenID 1.0/2.0
- Consumer Oriented
  - Based on REST, JSON and Oauth 2.0

OpenID Connect

Example auth flow:



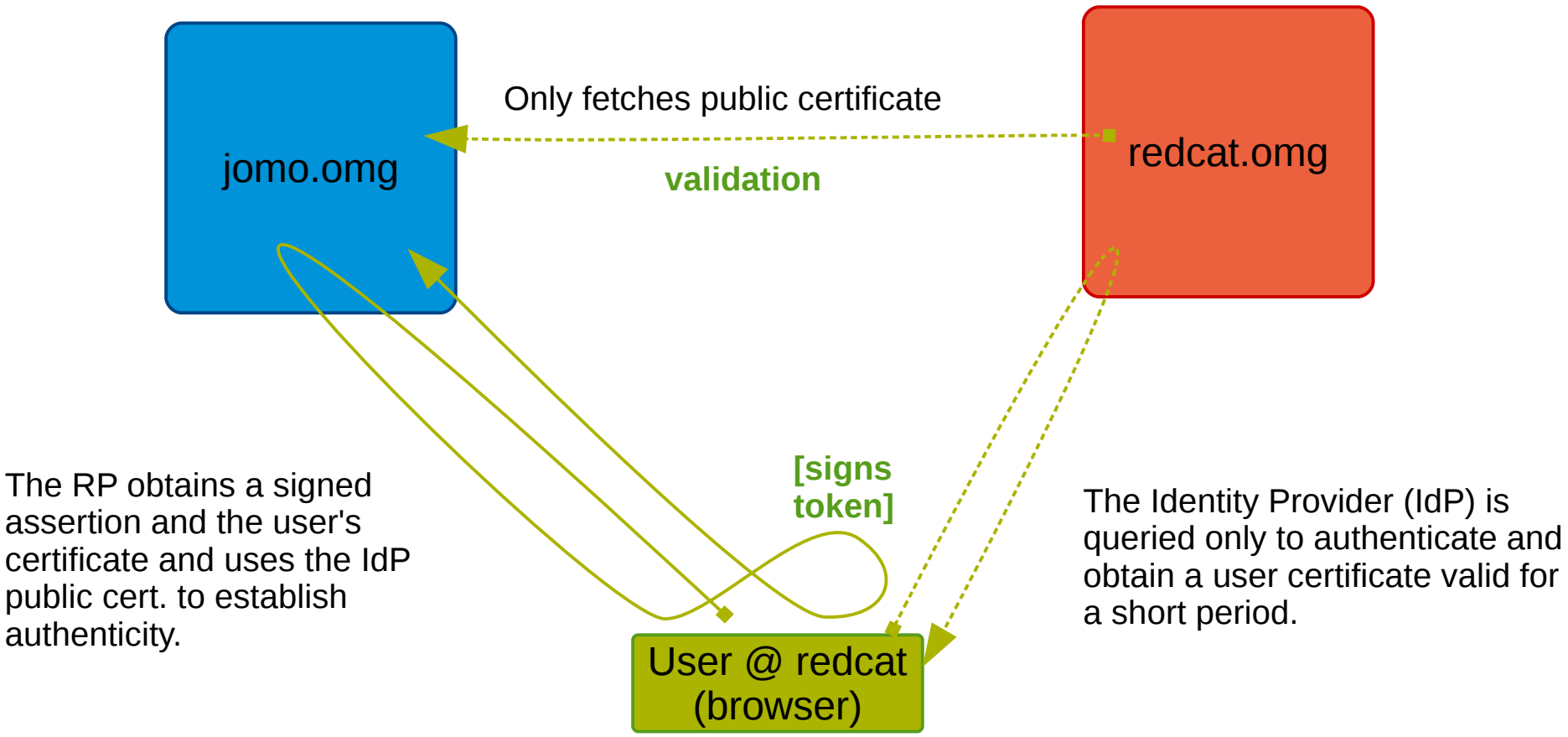
Persona

Key aspects:

- Privacy oriented
  - The Idp doesn't get to know each and every user's move
- Requires a browser plugin or some complex javascript
- Based on email address for identity
  - requires a public website to host the Idp public certificate
  - Uses crypto to generate custom user certificates
- Uses custom public/private key protocol
  - The protocol is called BrowserID

Persona

Example auth flow:

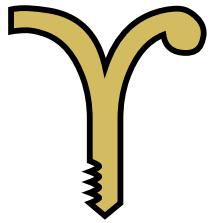


The RP obtains a signed assertion and the user's certificate and uses the IdP public cert. to establish authenticity.

The Identity Provider (IdP) is queried only to authenticate and obtain a user certificate valid for a short period.

The Ipsilon project

Ipsilon



A pluggable Identity Provider

Supports multiple authentication methods

Supports multiple Federation protocols

Provides tools for easy installation, configuration and management

Not an Identity Management server

Ipsilon

The server is built in python

- Best run in mod_wsgi
- Standalone mode via cherrypy
- Plugins are “drop-in”

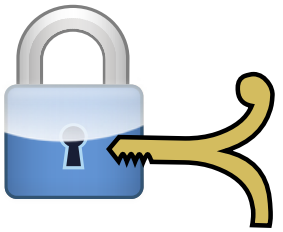
Clients available for apache

- Native C modules
  - mod_auth_mellon (for SAML)

Merged with FedOAuth

FedOauth

- Current Fedora authentication system
- Implemented OpenID
- Written in python too



Merged into Ipsilon

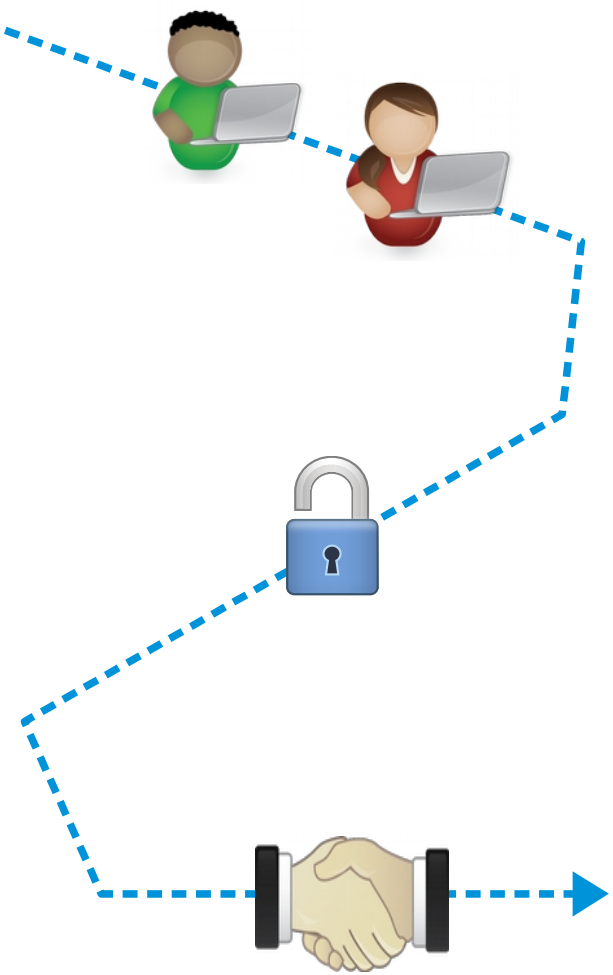
- Merge complete in December 2014
- Rolling the merged Ipsilon project in Fedora Infrastructure right now

Ipsilon authentication

Supports authentication via

- any apache module
- direct LDAP binds
- Kerberos
- Chaining to other IdP
- IPA / AD / etc...

- Supports fetching info via
  - LDAP
  - Nsswitch
  - Other IdP

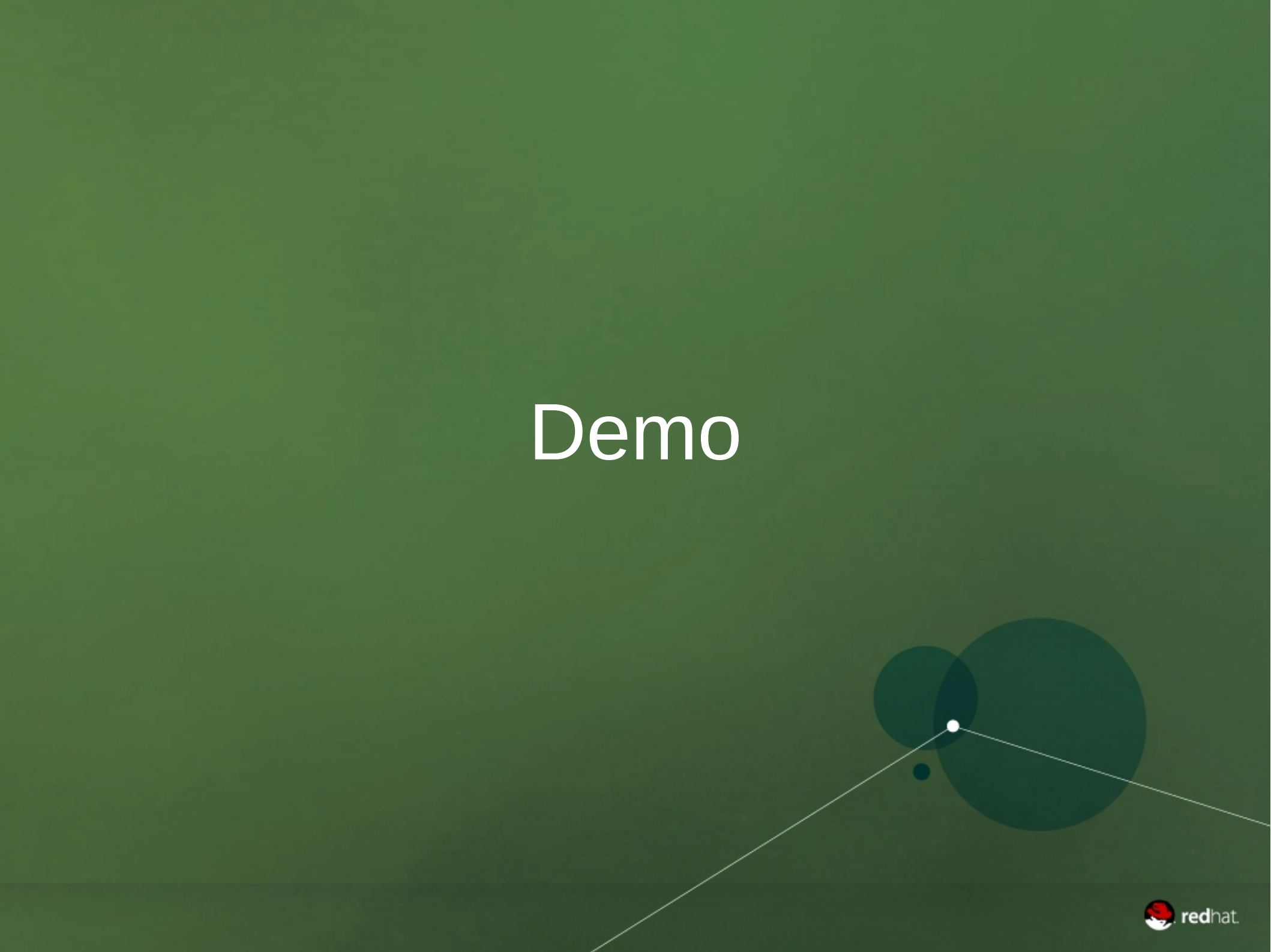


Ipsilon protocol support

Federation protocols:

- SAML
  - Uses lasso/xmlsec1 libraries
  - Main focus when project was started
  - ECP profile in the making
- OpenID
  - Ported over from FedOauth
  - For Fedora Infrastructure support
- Persona
  - Ported over from FedOauth
- OpenID Connect (next)

Demo



Ipsilon roadmap

Integration with FreeIPA should be seamless

- Automatic configuration during setup

REST API

- For all admin operations
- For SAML SP registration

Protocols:

- Improve SAML support
- OpenID Connect
- More auth/info plugins
- kx509 ?

Questions ?

Project points of contact:

http://fedoraproject.org/ipsilon

#ipsilon on Freenode

Feedback about this talk: http://devconf.cz/f/24