Shibboleth Project June 2026 Webinar

Scott Cantor
scott@restingparrotsoftware.com
Shibboleth Consortium

Key Takeaways

- Major work now is on the SP re-development and OID Federation support. What time remains is IdP and plugin maintenance.
- The Windows installers for the IdP and Jetty are deprecated, will be discontinued when IdP V6 ships.
  - The IdP's standard installation process and the new Jetty plugin are (and will remain) fully supported on Windows.
- SP re-development on track to ship in early 2027 (late 2026 aspirational). The Alpha 2 Agent will be out shortly.
  - Current SP likely sunsets by end of 2028, certainly by mid-2029, but scrutiny from e.g., AI-assisted scanning could render dependencies non-viable much sooner.

Roadmap Status

Active Work

- IdP maintenance
- OpenID Federation development and documentation
- Service Provider 4 development and documentation
- Actively reviewing sustainable path to get our artifacts into Maven Central
- Prototyping Verified Credential issuance (hosting third party work)

Lagging

- IdP documentation redevelopment
- IdP User Interface enhancements

Identity Provider

- V5.2 branch on Spring 7.0 released (EOL July ’27)
- We will ship V5.3 on Spring 7.1 late this year to extend support into mid-2028. Given current development resources, we can’t devote the effort needed for V6.0 until SP4 is released.
- While there are a couple of significant bugs we can’t address until V6, we have no urgent blockers requiring it.
  - If you do have major blockers, please raise them.

Jetty Base Plugin

https://shibboleth.atlassian.net/wiki/x/EIAP6Q

- Appears stable/sound/fun/nutritious.
- Future work likely will be Docker examples/scripts wrapping the Jetty process, but this is very opinionated work, and our opinions won't match anyone else's, so probably best done by people themselves.
  - If there are desired enhancements or changes to facilitate this, please submit them.

OID Federation

- Fed support ”separate” from core OIDC/OAuth configuration for modularity, so new plugins are being developed:
  - oidfed-common – the generic features supporting trust resolution and policy enforcement, caching, etc.
  - OP oidfed plugin – Federated OP features such as automatic and explicit registration of clients
  - Work in the early stages on RP support both for proxying and in the forthcoming SP
- Docs TBD but some basic WIP material is posted at https://shibboleth.atlassian.net/wiki/x/KIBGDAE
- OP Fed support being tested in the eduGAIN OpenIDFed pilot

Service Provider

- Preparatory work done on deprecations and warnings for a 3.6.0 release in late 2026, likely the last planned V3 update
- V4 is quite far along and desperate for alpha testing and feedback, but not getting any (I mean that literally)
- New Agent and Hub documentation is separate, reflecting an expectation of different audiences
  - Approached with survey input on IdP in mind, broken into intro/intermediate/reference material

Service Provider 4 (Agent) Goals

- ✓ Shrink C++ code base into a single footprint without third-party dependencies (treating standard Linux packages as first-party)
- ✓ Reuse of well-tested SP code where applicable/possible even if not always ideal design
- ✓ Very simple agent configuration, primarily INI-based
- ✓ No/little awareness required of specific SSO protocols or settings
- ✓ No key handling
- ✓ SAML 2 and OIDC support (and OID Fed, though possibly post-1.0)
- ✓ 100% compatibility for most existing applications using SP V3

Service Provider 4 (Hub) Goals

- ✓ Java-based IdP plugins reusing existing IdP services and configuration
- ✓ Shared usage by large numbers of Agents with manageability in mind

  Scaling credentialing of Agents is very much an open question and likely will evolve over time based on feedback
- ✓ Minimize net-new configuration for majority of use cases/needs
- ✓ Provide scriptable customization features as with the IdP
- ✓ Unified SAML SP and OIDC RP support

Service Provider 4 SAML Testing

- Install Agent and IdP
- Add Jetty and shibd/shibd-saml plugins
- Download and enable Jetty – default of http://localhost/8080 works for a local Agent
- SAML Basics - https://shibboleth.atlassian.net/wiki/x/EAAJFAE
  - Self-assign an entityID
  - Generate keys and import saml-credentials.xml into agents.xml (idp/conf/sp/)
  - Add e.g. InCommon MDQ service to Hub config or perhaps a local metadata feed for an IdP
  - Set a few properties to point to IdP or Discovery Service
  - Share SP metadata with IdP(s)

Demos

Demo videos (publically) available of SAML and OpenID integrations of SP4 with Entra and Google

https://shibboleth.net/documents/2026-06-webinar/

Agents and Hub Configuration

- New service added for Hub is the Agent Resolver (all the old IdP services are generally used in some obvious way by Hub)
- Agents can contain “Applications”, which can contain RelyingParty Configurations (a la the existing IdP config)
  - Simple approach = simple, complex approaches = complex
- Work continues to streamline and automate Hub configuration, focusing on typical enterprise scenarios:
  - Auto-recognition of SAML and OpenID plugins and their contributed profiles and settings
  - Properties to configure global behavior where usable, metadata tags for per-IdP settings (not wired in yet, but will do that shortly)
  - Single location to define profile beans with custom settings when necessary to apply to all Agents

agents.xml Example from our Testbed

```
    <!-- Local Apache Agent -->
    <bean p:id="localhost" parent="shibboleth.sp.Agent" />

    <!-- Scott's laptop -->
    <bean p:id="sp.restingparrotsoftware.com" parent="shibboleth.sp.Agent"
            p:issuer="https://sp.restingparrotsoftware.com/sp"
            p:credentials="#{{ 'restingparrot': 'password' }}" />

...a few other team members' agents...

...lots of empty placeholders for customization we hope people avoid using...

    <!-- Still manual, hoping we can automate eventually. -->
    <import resource="saml-credentials.xml" />
```

Files Added by Hub

- sp/agents.xml – the Agent Resolver
- sp/sp.properties – general protocol-independent settings
- sp/saml-credentials.xml – new keys added for SAML
- sp/saml.properties – SAML-specific global settings
- sp/oidc-credentials.xml – new keys added for OIDC
- sp/oidc.properties – OIDC-specific global settings
- credentials/sp/ - suggested location for new keys

Agent Management

- The crux: how would a medium or large org manage hundreds or thousands of Agents?
  - Scott presumed this was a somewhat “solved” IAM problem but it doesn’t seem so.
- Hub supports use of LDAP/etc. service accounts for Agents, so organizations with scalable “service account management” should have obvious path.
- For others, this is a work in progress right now.

Scalable Agent Management Ideas

- A text-based “loader” for simple Agent definitions to avoid constant XML manipulation. Likely part of 1.0.
- Obvious option, implement database-backed Agent records allowing for simple → complex solutions over time to manage them. Less likely to be part of 1.0.
  - Eventually some kind of token-based enrollment process a la OpenID dynamic registration
- Non-goal – federated management; Hubs are NOT meant to be operated by third parties other than via outsourcing/SAAS arrangements.

Questions?