ABFAB Architecture Overview

Josh Howlett, Sam Hartman, Hannes Tschofenig, Eliot Lear

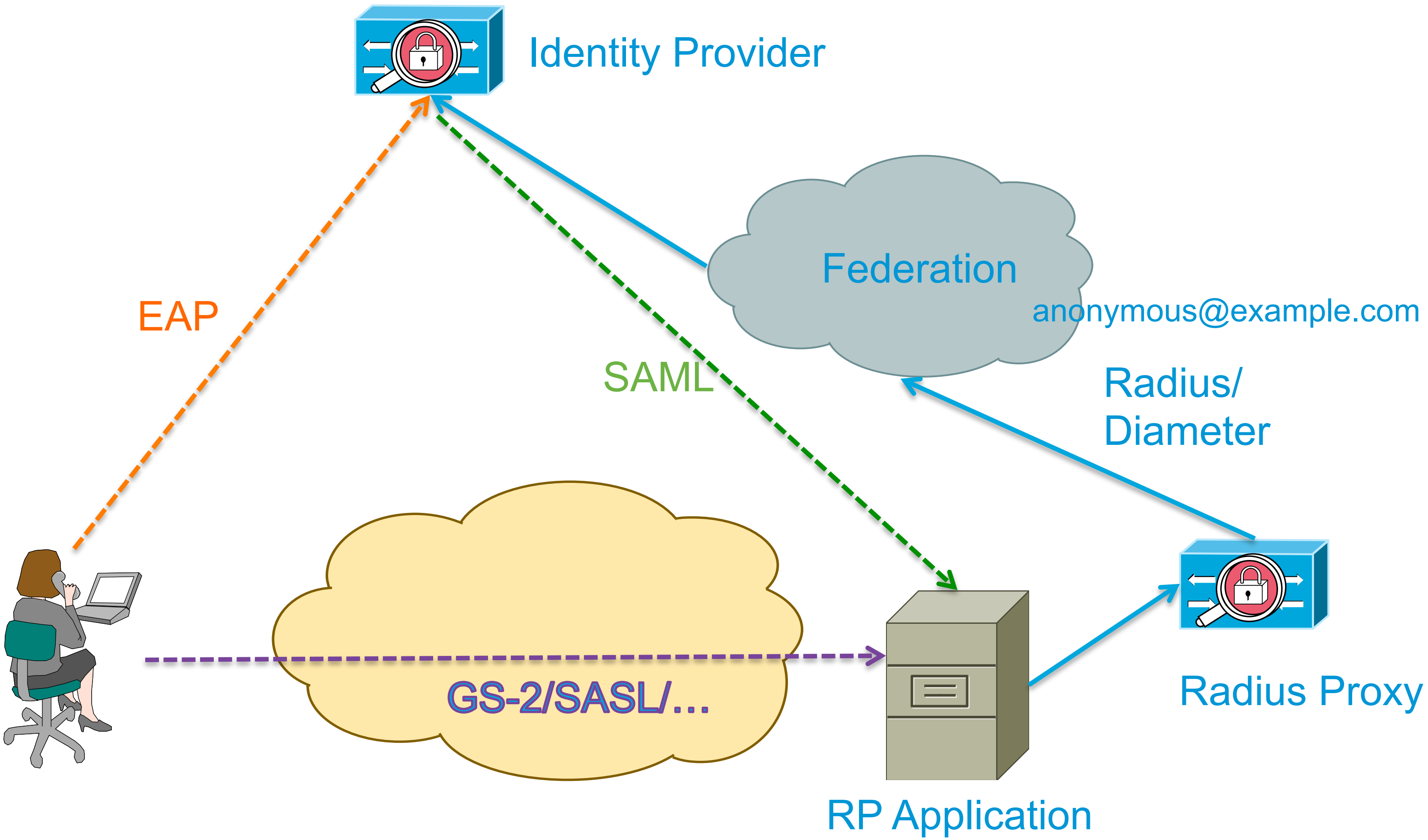
draft-ietf-lear-abfab-02

- Introduction
- Design Goals
- Major components
  - RADIUS/Diameter
  - GSS/GS2
  - EAP
  - SAML
- Discovery
- Trust
- Privacy Considerations
- Deployment Considerations
- Future Work

Design Goals

- Each party of a transaction will be authenticated, and the principal will be authorized for access to a specific resource.
- Means of authentication is to be decoupled so as to allow for multiple authentication methods.
- No sharing of long term private keys.
- Scale to large numbers of identity providers.
- Focus on non-web-based authentication.
- Stand on the shoulders of others (and not their backs).

Basic Components



Why GSS-API/SASL?

- Why GSS-API/SASL?

  We need a generalized application service interface

  Both GSS-API and SASL are there

- Why RADIUS?

  Need a AAA substrate that builds on existing trust relationships, where possible.

  Wide successful deployment

- Why EAP?

  We need a way to generalize **end-to-end** authentication mechanisms

  Lots of work has gone into EAP mechanisms.

- Why SAML?

  Need a way to frame and transport attribute assertions.

  SAML is widely deployed on the web.

From the draft…

```
Relying Party    Client App       IdP

         |              (1)             | Client App gets NAI (somehow)
         |               |              |
         |<-----(2)----->|              | Mechanism Selection
         |               |              |
         |<-----(3)-----<|              | NAI transmitted to RP
         |               |              |
         |<=====(4)====================>| Discovery
         |               |              |
         |>=====(5)====================>| Access request from RP to IdP
         |               |              |
         |               |< - - (6) - -<| EAP method to Principal
         |               |              |
         |               |< - - (7) - ->| EAP Exchange to authenticate
         |               |              | Principal
         |               |              |
         |               |           (8 & 9) Local Policy Check
         |               |              |
         |<====(10)====================<| IdP Assertion to RP
         |               |              |
         |               |              | (11) RP Processes results.
         |               |              |
         |>----(12)----->|              | Results to client app.
```

Discovery

- Based on Network Access Identifier (NAI) realm component [RFC4282]
- Realm = IdP
- Routing of request to IdP not in scope (right now)
  - Could be statically configured
  - AAA proxies
  - Trust Brokers
  - Global Credential
- Relying Party determines order of discovery when multiple federations exist
- IdP is usually billable party

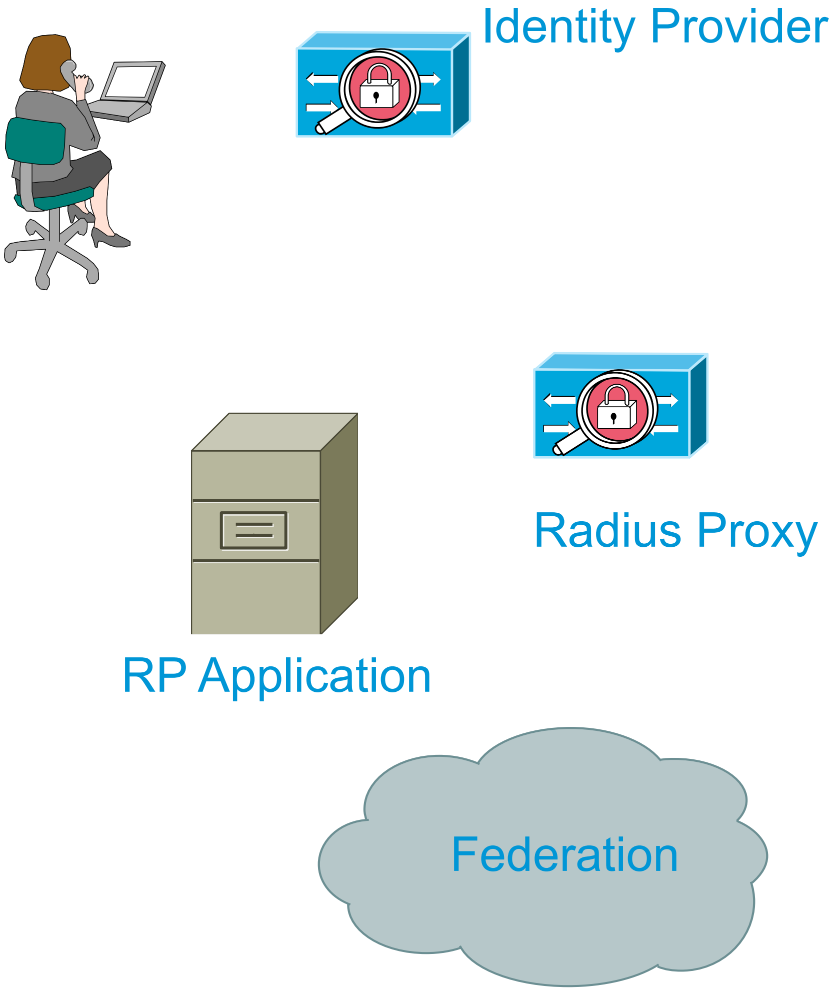
When the individual and identity are multiple federations

- The individual is represented by the IdP and must trust the IdP
- The Relying Party wants to maximize revenue.

  The model allows for the RP to order discovery.

  The model does NOT allow for the RP to see other than results indications and assertions from the IdP.
- The Federation wants to maximize revenue.

  Any discovery through a federation cannot provide the federation to claim it’s the only path.



Trust

- The principal

  trusts IdP to authorize and protect privacy

  trusts the relying party to deliver services

- The relying party

  trusts the federation to reach the IdP

  trusts the IdP to provide accurate authentication and attributes about the principal

- The IdP

  trusts the federation to authorize and convey RP communications

- The federation

  relies on no claims

Privacy Considerations

- Extensive discussion about sharing of principal information
  - Relationship between users and other entities
  - What data about the user is likely needed to be collected?
  - What is the identification protocol layer?
- Challenges
  - Federation agreements are often not transparent to all parties
  - Limited control available by principal

Future Work

- 3rd party attribute providers
  - Do we go through AAA infrastructure?
  - Use of HTTP?
  - Interactions with other infrastructure (OAUTH)?

Open Issues

- Which EAP mechanisms should be recommended (if any)?
  - Do we have a mandatory-to-implement mechanism?
- UI issues will impact us. Are they solved here?
- SAML exchanges need lots of tightening in the document
- More detail in the swimming lane diagram?
- Implementation guide needed?
  - Normative language in current version
- Security Considerations need to be written

And one last one…

There is an open question here as to the details; today RFC 5554 governs. We could use that and the current draft assumes we will. However in Beijing we became aware of some changes to these details that would make life much better for GSS authentication of HTTP. We should resolve this with kitten and replace this note with a reference to the spec we're actually following.

Next Steps

- Your turn: please read the draft
- WG draft?
- Split normative text out?