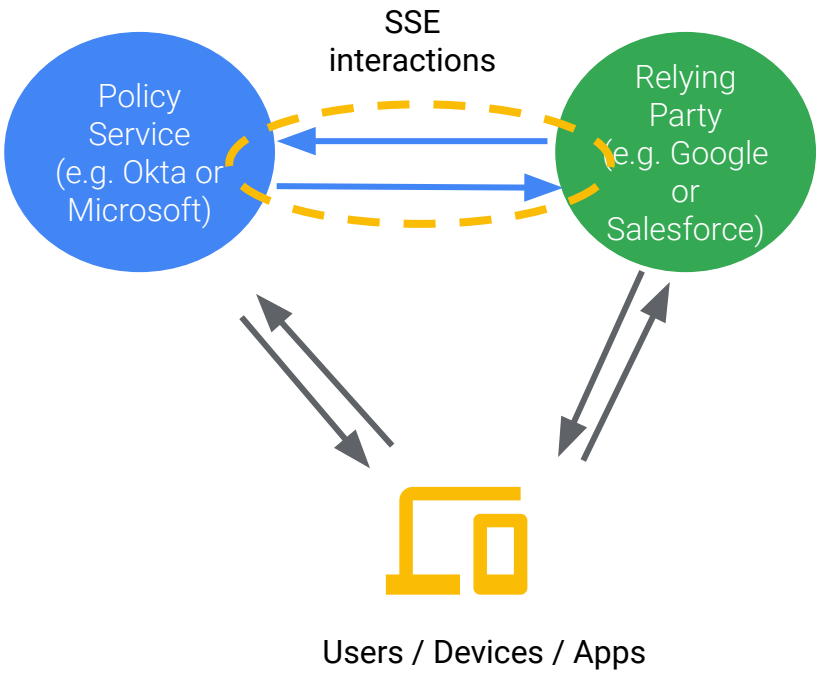
Shared Signals and Events

Summary and Status - Spring 2021

Atul Tulshibagwale
(Google)

The Shared Signals and Events Framework

- Synchronizes distributed state relating to shared principals
  - Tenants
  - OUs
  - Users
  - Sessions
  - Devices
  - Applications
  - ...more in future
- Based on asynchronous publish and subscribe of Security Event Tokens (SETs)
- Defines an event stream management API to manage streams between Transmitters and Receivers
- Defines a structure for subject identifiers



SSE Framework: The Event Stream Management API

- Transmitter Configuration Metadata
  - GET - receiver gets current information
  - POST - receiver specifies its information
- Stream Status
  - GET - gets transmitter status
  - POST - Receiver updates status

SSE Framework: The Event Stream Management API (Continued)

- Add Subject - Receiver adds or removes subject in the stream
- Remove Subject - Receiver removes subject from the stream
- Stream Verification
    - POST requests verification
    - Verification Event - Transmitter sends event that Receiver should use to verify liveness
- Stream Updated Event - Transmitter updates stream status

SSE Framework: Simple Subjects

- Simple Subject - Single subject identifier

Simple Subject Example:

```
"subject": {
  "format": "email",
  "email": "user@example.com"
},
```

SSE Framework: Complex Subjects

- Complex Subject - Multiple subject identifiers all describing the same entity (e.g. user, device, session, application, etc.)

Complex Subject Example:

```
"subject": {
  "session": {
    "format": "opaque",
    "id": "dMTlD|16002.16|16008.16"
  },
  "user": {
    "format": "iss_sub",
    "iss":
"https://idp.ex.com/12/",
    "sub":
"dMTlD|16002.16|16008.16"
  },
  "tenant": {
    "format": "opaque",
    "id": "123456789"
  }
```

Applications of Shared Signals and Events

Continuous Access Evaluation ***Profile*** (CAEP)

- Conveys state changes relating to access to resources
- Enables peers to tune access based on updated state
- Expected to be frequent, normal events
- Event types include:
  - Credential change
  - Token claims change
  - Level of Assurance change
  - Device compliance change

RISC

- Enables providers to prevent attackers from compromising linked accounts
- RISC helps enables coordination in restoring accounts in the event of compromise
- Expected to exceptional, rare events
- Event types include:
  - Account purged
  - Credential change required
  - Identifier recycled

Current Status

- Specification
  - CAEP work merged into master branch
  - Will be requesting to start the comment period for making it an implementers' draft

- Industry adoption
  - Microsoft, Sailpoint, Thales and Google working on implementations

Google Adoption

- Operating a production “Cross-Account Protection” service based on RISC
- Working on proof-of-concept CAEP implementation with third-parties on different use-cases

Microsoft Adoption

- Implemented CAEP Event Types with SET Push across many Microsoft services
- Begun working with a few third-parties for signal sharing around session revocation

SailPoint Adoption

- Developed and released an open source Java library:
- SailPoint Java library @ GitHub
    - sailpoint-oss/openid-sse-model

Thales Adoption

- Working on Proof-of-Concept with third-party partner
- Use-case:
  - Suspicious activity at SP1
  - User logged out, session ended by SP1
  - SP1 sends CAEP event to IdP
  - User sent to IdP, and is asked to authenticate again (SSO session terminated after CAEP event)
  - IdP sends CAEP event to SP2
  - User logged out, session ended by SP2