Agenda

● Core draft update
  ○ Trust relationships
  ○ Security considerations
  ○ Privacy considerations

● Open Issues
  ○ Symmetric Crypto
  ○ SOLID use cases
  ○ End-user vs. RO
  ○ Generic HTTP access type

● What topics to focus on for IETF 112?
Draft Changes

- Collapse of “user_handle” into subject identifier constructs
- Trust Relationships
- Security Considerations
- Privacy Considerations
User Handle

- Use “subject information” opaque identifier instead of separate user handle
- Simplifies the protocol, uses constructs we already have

Response from AS:

```json
{
  "subject": [{
    "format": "opaque",
    "id": "XUT2FM1XBIKJKSDU8QM"
  }]
}
```

Request from Client Instance:

```json
{
  "user": "XUT2FM1XBIKJKSDU8QM"
}
(or)

{
  "user": [{
    "format": "opaque",
    "id": "XUT2FM1XBIKJKSDU8QM"
  }]
}
```
Trust Relationships

- Defined using promise theory (new informative reference)
  - allowing for a formal trust model, including threats
- New section 1.4 details the promises between end-user/RO, end-user/client, client/AS, RS/RO, AS/RO, AS/RS
- Refers to security and privacy considerations

### Proposal 1 (Trust)

An agent’s expectation that a promise will be kept. It may be assigned a value lying between 0 and 1, in the manner of a Bayesian probability.

### Proposal 2

Trust of an agent $S$ by another agent $R$ can exist if agent $R$ is informed that agent $S$ has made a promise to it in the past, or if the recipient of the promise $R$ is able to infer by indirect means that $S$ has made such a promise.
Security Considerations

● 21 Subsections, including:
  ○ TLS is required
  ○ You have to protect your keys and other artifacts
  ○ Bearer tokens cause problems
  ○ Use real crypto and randomization
  ○ Front-channel redirects are inherently susceptible to attack
  ○ You have to check all the hashes and signatures
  ○ Pre-registration doesn’t solve all the problems you think it does
  ○ MTLS doesn’t solve all the problems you think it does
  ○ Just because something is signed doesn’t mean you can trust it
Privacy Considerations

- Modeled after RFC6793
- Main topics:
  - Surveillance
    - Surveillance by the Client
    - Surveillance by the Authorization Server
  - Stored Data
  - Intrusion
  - Correlation
    - Correlation by Clients
    - Correlation by Resource Servers
    - Correlation by Authorization Servers
  - Disclosure in Shared References
Open Issues
Symmetric Cryptography

- Issue #299: Should we completely disallow symmetric cryptography?
- Reasons to disallow:
  - Symmetric crypto relies on keys being in the hands of both parties
  - Asymmetric crypto exists and is functional
- Reasons to allow:
  - Underlying crypto methods allow for symmetric cryptography
  - GNAP does not allow for symmetric key distribution
    - Only identifiers can get passed around
  - KMS and key derivation are safe practices
  - Post-quantum cryptography is largely symmetric
SOLID use case

- Client has access to provable claims about end-user
  - Can get these through a secondary AS
  - Backed by WebID trust in SOLID ecosystem
- Client presents claims to AS for access
  - AS maps claims about user to an RO and policy for an RS/resource set
  - AS probably doesn’t interact with end-user
- Dynamic discovery is necessary, pre-registration not tenable
- Application of consent-and-interaction phases?
  - Client tells AS that it can talk to EU’s server to get additional info if wanted
  - A kind of reciprocal GNAP?
Case 1: Client gets artifact from external service

- **end user**
- **client instance**
- **authorization server**
- **end user extension server**
- **resource server**

1. Request access, can do GNAP at euxs
2. Need claims about eu from euxs
3. Request claims about eu for as
4. Need to interact with eu
5. Interaction start
6. Interaction finish
7. Continue request
8. Claims about eu
9. Claims about eu
10. Token for eu based on claims
11. Access resources
12. Return
13. Go to euxs
Case 2: AS gets artifact from external service
Open Questions

● Case 1:
  ○ Presumes a verifiable artifact that client can carry to AS
  ○ Client could pre-load this artifact

● Case 2:
  ○ AS acts as client to external AS
  ○ AS can’t interact with end user normally
  ○ Is this any different from the mix-up attack we just patched against?
End-user vs. RO

● Two different roles for users in GNAP:
  ○ “End-user” uses the client software
  ○ “RO” controls access to the protected resources

● In OAuth they’re always the same person
  ○ GNAP interaction lets you connect the end-user to the AS so they can act as RO
  ○ GNAP doesn’t require end-user to be RO if AS can reach the RO (or their policy) somehow

● Subject information muddles this distinction
  ○ When the client is asking for subject info, it wants to know who the end-user is
  ○ If the RO isn’t the same as the end user, isn’t this an error?

● Draft text isn’t always clear about cases where end-user and RO are different
Generic HTTP Access Type

- The “access” object’s “type” field is up to the API being protected (AS/RS)
- What if we had a “generic HTTP” type?
  - Applicable to nearly all HTTP APIs out of the box
  - “actions” maps to verbs
  - “locations” maps to URLs (or templates)
  - “datatypes” maps to mime types
- Should we do this?
  - Do all RS’s need to understand these types now?
- If we do this, where?
  - Inside GNAP core
  - Inside GNAP-RS
  - In another extension in GNAP
  - In an external document (outside of IETF?)
Additional Topics for IETF 112?