GNAP at IETF-110

March 9, 2021, 16:00 UTC

Chairs: Leif Johansson and Yaron Sheffer
Note taker(s): Kristina Yasuda, Jonathan Hammell

- Core draft update: changes since IETF109 (from -02 to -04) (Aaron Parecki)
    - everything captured in GitHub, 25 PRs in total
    - Editorial Changes
        - Removed closed issued from draft text (#150, #172)
        - Updated subject identifier info (#153, #177)
        - Minor typo fixes (#126, #179, #181)
        - Updated acknowledgements (#157)
        - Updated terminology (#29, #155)
            - Thanks to Fabien on moving that forward
    - Functional Changes
        - dropped redirect to a short URL (#139, #121, #53)
            - overlapped too much with regular redirect (QR codes, etc.)
            - reduced optional pieces
        - Dropped OIDC “claims” parameters (#140)
            - too far from GNAP goals, OIDC can specify that on top of GNAP
        - Made access tokens mandatory for continuation request (#129, #67)
            - two ways -> one way to do this, using access token
        - Changed access token request and response (#40, #39, #10, #13, #162)
            - bunch of issues, large syntactical change on how request and response are formatted
        - Refactor “key” information to new section (#152)
            - how key proofing works, still several options, broke into a separate section
        - Group interaction modes in the request (#122, #163)
            - how client can interact with a user: browser, redirect, etc. Grouped and labeled explicitly.

        - Dropped reading grant and token (#98, #99)
            - trying to simplify and remove optional things
            - no-one objected to removal of option
            - no concrete use-case
        - Access Token: New Request Syntax
            - Access Token as a container, aspects of the req.
            - Rich Object - full obj. that talks about that is asked
            - can be a simple string or rich object
            - corresponds to OAuth scopes
            - Reference - to be specific
            - Flags - used to be boolean
            - Labels - if asking for more than one access token
        - New Request syntax in interaction mode
            - works with browser or non-browser mode
            - how client starts and ends the interaction, grouped (new) in Start Modes and Finish Modes
            - UI hints meant for othe hints for the client

- Core draft roadmap: overview of next big topics (Justin Richer)

- On-device use cases and component definitions

    - Message signing mechanisms

        - work happening outside gnap WG
        - HTTP Message Signature Draft in progressing in HTTP WG
        - New ideas on using JOSE based on implementation experience from multiple sources
        - DPoP profile in FAPI WG in OIDF - will be coordinating
        - Open questions:
            - which ones will survive in the core document and which ones will be mandatory?
            - what is server and client do not support the same?
                - need to simplify
            - is anything MTI?
            - what are the failure states?

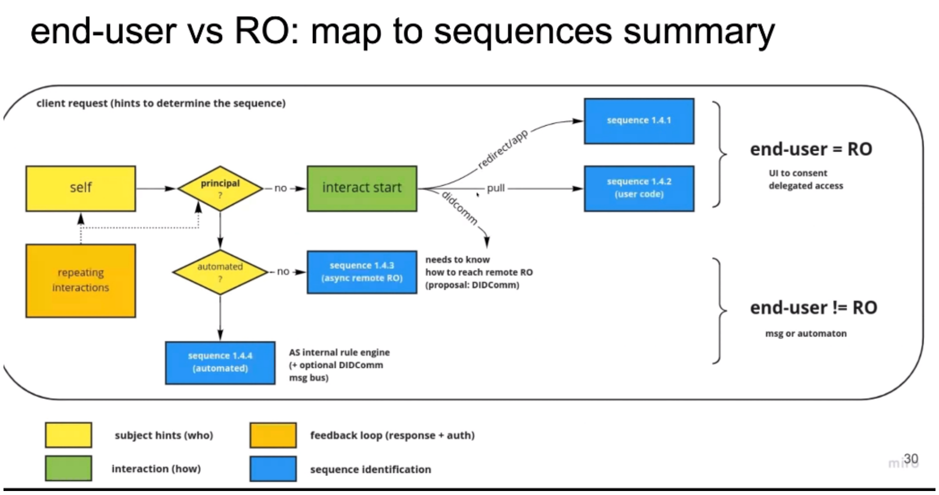
    - Key rotation (in the next few months)

    - imp as this relates to teh security model of the protocol
    - how do we allow different parties to rotate keys?
    - for client instances
      - clients use old keys? new keys? do we require AS to separate?
      - client management API?
    - for ongoing grants
      - Grant update API?
      - Does this also rotate key for client/token?
    - for access tokens
      - Part of token management API?
      - Could client instance use different keys for different tokens?

- Topic: Subject identifiers
  - Multi-user delegation (where user != RO)
  - Proposed next steps
    - Subject identifier (Fabien Imbault)
      - goal: provide PR soon; 10 % of all issues
      - subject or sessions related
      - #184 subject_types
        - subject_types (array of strings) in the request
          aubject_types_supported (array of strings) in the dictionary
        - when you make a request, you can request for sub or email
        - want to support a list of types
        - section 9, discovery
        - what Client knows about the user; you've got the req where you request more info; AS will respond with the info
      - #16, #42 Change examples - current based on email attribute
        - maybe email is not the best way to describe subject identifier, not encouraged
          - no way to make it unique, no info on the policy; do not want to tie to a delivery method
          - want more secure
        - account, alias, etc. supported
        - as_ref proposed - reference to the user what AS will try
      - #75 scope of subject identifiers
        - global or local identifier

  - to take from definition accepted by WG: “statement asserted locally by an AS about a subject”
- #171 subject identifiers as portable identifiers
  - related to OIDF Connect WG - how to make identifiers portable
  - two way to support: sec event alias or subject_types and assertion? (depends on what we support as types)
    - supports portability due to list of options related to the same identifier - can get from one identifier to another
  - portability might be beyond scope (more transactional), but GNAP can help support various types of identity systems (interoperability) - agnostic if OIDC or not, but support what exists
- #51 user reference as an assertion
  - opaque reference
  - could be used as in UMA (refre to that sub) or as in …?
  - label: as a reference os AS reference
  - unique locally
  - option 1: extention to secevents, extend sub type with a new label, reuse sub_identifier types;
  - option 2: GNAP-specific AS Reference; refer to internal value used for GNAP
    - currently this option in the draft
- #41 #43 assertions
  - OIDC ID tokens
  - need support for several assertions? as in array?
  - what types of assertions to support?
    - how would a registry work?
    - suggest to keep id_token (OIDC) since it is a use case we need
    - other option: DID, VCs, jwkthumbprint
    - remove SAMLv2… due to XML security - could be an extension
  - subject_type or assertion type? DID/jwkthumbprint
    - assertion makes more sense?
    - Is there a security issue in having a mapping between sub_ids and assertions?
      - could use more advanced types like Verifiable Credentials
  - #197 Requesting User Information
    - is the user the RO or end-user?

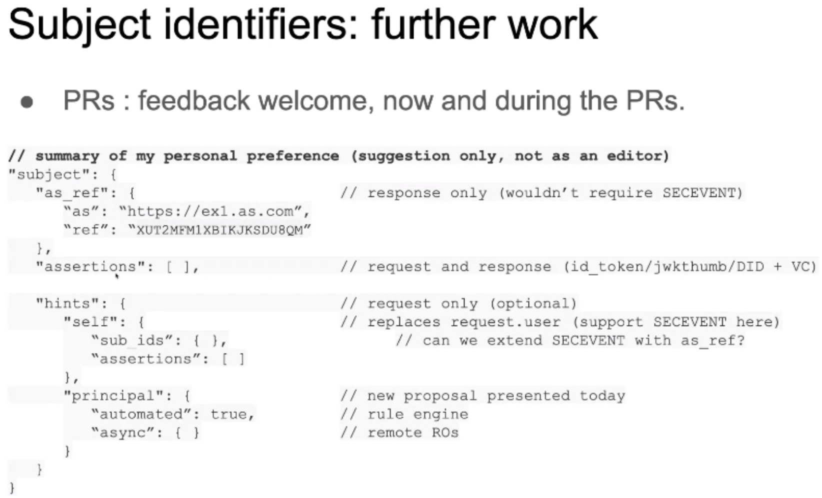
  - concrete text in section 2.4 under question regarding AS throwing an error if RO and end-user identifier does not match
- Personal thoughts – not raised as issues in GitHub yet:
  - RO = end-user in most cases, but in UMA2 they are different
    - creates a fragmented ecosystem
  - GNAP aims to solve this issue too via sequence diagrams 1.4.3, 1.4.4, but section 4 only covers UI interactions
    - innovative, but don’t cover case where RO != end-user
  - try to know when RO = User or not - usually a hardcoded assumption, can this be made the AS’s responsibility?
  - A complex application might require the user’s authentication to decide at run-time
  - How much do we expose that information to the client?
    - if anyone knows specific secret password, then AS should send any sensitive info about the RO
    - sequences 1.4.x, can try out further
  - intuition: user knows whether its his data or not.
  - find optional principle - determine who the resource owner is.
  - same structure, sub_types asked to AS, hints are described optionally
    - `self` request.user could be transferred here?
    - principal provides additional hints when the user is not the RO
    - AS is who makes the final decision
  - Example 1: 1.4.3 sequence (principal + async)
    - Parental control app where child will ask approval of his/her dad
    - within assertion, how do we know it’s a legitimate DID?
  - Example 2: 1.4.4 sequence (principal + automated)
    - Engine checks the age of the child who want to watch a film
    - assertion in the self-structure, ZKP of the age, you say it’s automated, running on AS
    - client will say to AS, please check automated rule to accept or deny the rule
  - composability
    - async could fail, automated rule engine could be limiting
    - could try fallback negotiation. e.g. child has been really good

- multiple ROs?
  - RO could be an array
  - want to manage resources managed between several parties. e.g. reach out to Mom
  - if multiple owners, need a mechanism to take a decision
- map to sequences summary
  - 

end-user vs RO: map to sequences summary

client request (hints to determine the sequence)
self
principal ?
no
interact start
redirect/app
pull
didcomm
sequence 1.4.1
sequence 1.4.2 (user code)
end-user = RO
UI to consent delegated access
repeating interactions
automated ?
no
sequence 1.4.3 (async remote RO)
needs to know how to reach remote RO (proposal: DIDComm)
end-user != RO
msg or automaton
sequence 1.4.4 (automated)
AS internal rule engine (+ optional DIDComm msg bus)
subject hints (who)
feedback loop (response + auth)
interaction (how)
sequence identification
30

- DIDComm delivery method
  - AS has its own DID
  - DID Comm for remote RO(1.4.3) and rule engine (1.4.4)
  - not mandatory, but considered as a technical framework
  - #197, #198 are related on terminology
  - in some cases subject will be a machine - rule engine from a company or a remote owner
  - DIDComm interact
    - Still need to decide how the interact would work (message format / query)
    - didcomm/didcomm_query: was discussed in XYZ, see issue #168
    - additional transport mechanism to reach out to remote owner
  - Role identification (RO / end-user)
    - does not remove the need for runtime authz - rather increases - AS decides what is happening
    - provides dynamic configuration for all sequences
    - light and composable, and allows to mix sync / async
    - clairfies when you have web interface and there doesn't
    - clarifies who's concerns

        - must deal with security and privacy considerations
    - further work on subject identifiers
        - summary of personal preference
        - 

Subject identifiers: further work

- PRs : feedback welcome, now and during the PRs.

```
// summary of my personal preference (suggestion only, not as an editor)
"subject": {
   "as_ref": {                           // response only (wouldn't require SECEVENT)
      "as": "https://ex1.as.com",
      "ref": "XUT2MFM1XBIKJKSDU8QM"
   },
   "assertions": [ ],                    // request and response (id_token/jwkthumb/DID + VC)

   "hints": {                            // request only (optional)
      "self": {                          // replaces request.user (support SECEVENT here)
         "sub_ids": { },                       // can we extend SECEVENT with as_ref?
         "assertions": [ ]
      },
      "principal": {                     // new proposal presented today
         "automated": true,              // rule engine
         "async": { }                    // remote ROs
      }
   }
}
```

- Discussion
    - Yaron Sheffer:SECEVENT: new version of subject_id moving to last call potentially
- Topic: On-device use cases and component definitions (SSI and GNAP)
    - Brought up a lot - how do we deal with SSI outside of the OAuth community
        - justin has been involved with W3 DID-COREC WG
        - work happening in OIDF Connect WG on Self-Issued OpenID Provider (SIOP)
            - SIOP uses implicit/hybrid flow that is now deprecated in OAuth 2.0
        - system used in a way not possible before
        - need ot look at the interconnections
    - Functions and responsibilities of the AS
        - known as Bring your AS model
        - Downsides of this approach
            - how does RS know when and how to trust access tokens issued on devices
        - client and Server both on the user's device usecase
            - trusting a new OAuth client is a problem
        - How do you discover an AS that is not reacheable from HTTP - client not a device, how RS gets introduced to this.
        - What happens if Bring your own AS happens
            - proxy AS being built - well understood pattern, but the problem shifted

    - what comes from the downstream
    - are we actually talking about the user bringing their own AS endpoint?
  - AS as Token Factory
    - who is making/authorizing the request
    - issue access token that can be used
    - aligned to gnap, OAuth 2
    - push back on the implied notion that AS has to have user login and user interact with it directly
    - new way for the user to present verifiable information to the third party - that could be a GNAP server
    - client start negotiation with AS, "hey I know how to present VC from the user, I can get you in touch you with the user"? user interacting with a separate party - as long as AS has a way to make that connection. not up to gnap to define those connections. I can get user in front of a webpage. when users
    - SIOP pretends it does redirect, while it is doing something else
    - we are not even inventing this - cloud based OAuth providers, federated login - no account at AS user is authenticating to. user is not even providing consent. AS is told through the assertions, trusts and issues a token
    - how OAuth 2.0 is written - assumptions are about redirecting the user, asking for codes, etc.
    - fantastic opportunity to codify what we are doing with AS - do not have to assume user interacts
    - we allow user to bing AS, but it is not all-mighty component that it was previously assumed to be.
    - who is the RO, how we interact with them
    - allows asynchronous user-focused interaction
    - Benefits of this approach
    - this is what I am asking for, this is who is allowed to say yer - can do direct interaction if the same as RO
    - in more advanced use-cases, more to negotiate send this communication across the wire
    - not changing how this protocol works; people already have weird add-ons
    - but shifting the way we speak about this is important
    - problems on the list of people applying terms from OAuth world in ways

      that was not intended
      - must be careful with the roles: can't say its client who gathers the consent
        - RS/AS trust boundary is well-understood
        - GNAP doesn't have to assume user logs in to AS
          - User might not even interact with AS during request
        - Extension points for interaction and claim formats
        - AS figures out who needs to approve based on what's being asked for through an assertion

- Possible Alignment with Self-Sovereign Identity (SSI)
  - Privacy-preserving use cases
- Discussion
  - Yaron: terminology, calling AS as a token factory, emphasis on the output rather than the behavior.
    Q: Is it even an AS if it doesn't enforce a policy?
    - Justin: Agreed. AS's job is to enforce the policy
      - cloud server policies can be very simple, not with all the user information
    - Aaron: Policy doesn't have to be pre-known user, can be just told about user from somewhere else
    - Justin: Cannot drop use case of HTTP redirects
  - Kristina: Is Self-Issued OpenID Provider in scope for GNAP?
    - Justin: Is SIOP in scope? As an individual, the goals and outcomes of SIOP should be in scope. It does not make sense to have the technology or even model of SIOP as part of GNAP. It starts with a redirect. Simplifies on how to work across diverse use cases. Choosing the grant type ahead of time make a lot of assumptions and in OAuth there is no way to switch the grant type. In the wild, try different flow when one fails. In GNAP, things can be presented to the AS and the AS can decide. Goal of allowing user to bring identity claims makes sense, but bringing AS with them should not be included (in personal opinion). SIOP can start with HTTP call (self-issued.me (http://self-issued.me) URLs). GNAP is not doing something that is radically different, but it is facilitated to support other types of interactions.
  - Justin: much of the framing language will need to be changed in the draft.

        - Yaron: was this asked on the list?
        - Justin: sent on Feb 17. Long thread, so will summarize on the list.
        - Fabien will summarize on the list for his topics.
        - Roman agrees with that approach.
- GNAP model & trust relationships: Privacy and security considerations (Denis Pinkas)
    - three main components of the model: client, AS, RS
    - important that the client is operated by the user, requiring privileges
    - not one AS, but several AS and a single RS can have relationships with several RS
    - what is important is that AS runs priviledges that can be an attribute or a right
    - to capture both access control lists and capacities.
    - no concept of the protected resources
    - Proposed definition of RS: server that provides operations to objects reference by specific application requests
    - additional concept of SERVICE
        - RS may publish to what type of SERVICE it applies
    - Scalability:
        - In OAuth: prior relationship between every RS and AS is not scalable
        - In GNAP: the RS trust a set of ASs for some sets fo privileges contained into an access token
            - no prior relationship needed
            - must clarify the trust relationship in the current draft
    - Two faces of the RO:
        - Old ISO document as Access Enforcement Function and Access Decision Function
        - access control rules are ACL based and/or capability based
        - token may not be necessary or rule may require multiple access tokens
        - combination of AS URL and operators and when those rules based on the control list, used ot present a method of the object
        - Capability based:
            - AS must cooperate with an RO
            - RO may be process, human being, or device (IoT case)
            - when AS delivers a capability, that cpability is analsed and filtered by the ADF of the RS
            - RO acting as ADF for an RS when both ACLs and/or capabilities are used
            - two kinds of RO - better to have separate terminology?

        - Privacy by Design
            - has to be taken into account prior to defining protocols
            - already too late for gnap? (privacy after design)
            - Two important privacy considerations are about:
                1. User choice and consent, and
                2. User notice
        - user choice and consent
            - user should be able to make a choice of AS before contacting AS
            - every end user should be able to agree on the values (rights attribute types) placed in the tokens
            - these two steps are unsupported/missing in the draft
        - end-user identifiers that need to be distinguished
        - Unlikability between RS user accounts versus Client collaboration attacks
            - Consequence: if an access token only contains one or more capabilities, client collaboration attacks will succeed.
        - End-user identifiers types
            - Consequence:
                - A RS should be able to indicate to a client which end-user identifier types may be presented to be able to perform an operation.
                - The client should be able to indicate these end-user identifier types to the selected AS, and
                - The RS must be able to distinguish between these end-user identifier types obtained from an AS.
        - liaison with SECEVENT WG should be established
        - Lief: Thanks for sending these topics to the editors. Please summarize on the mailing list

- Next steps
    - to keep up cadence of regular interims