GNAP Future Work
IETF 113

draft-ietf-gnap-core-protocol-09
draft-ietf-gnap-resource-servers-01

March 25, 2022

Justin Richer • Aaron Parecki • Fabien Imbault

Agenda

- Future work and roadmap

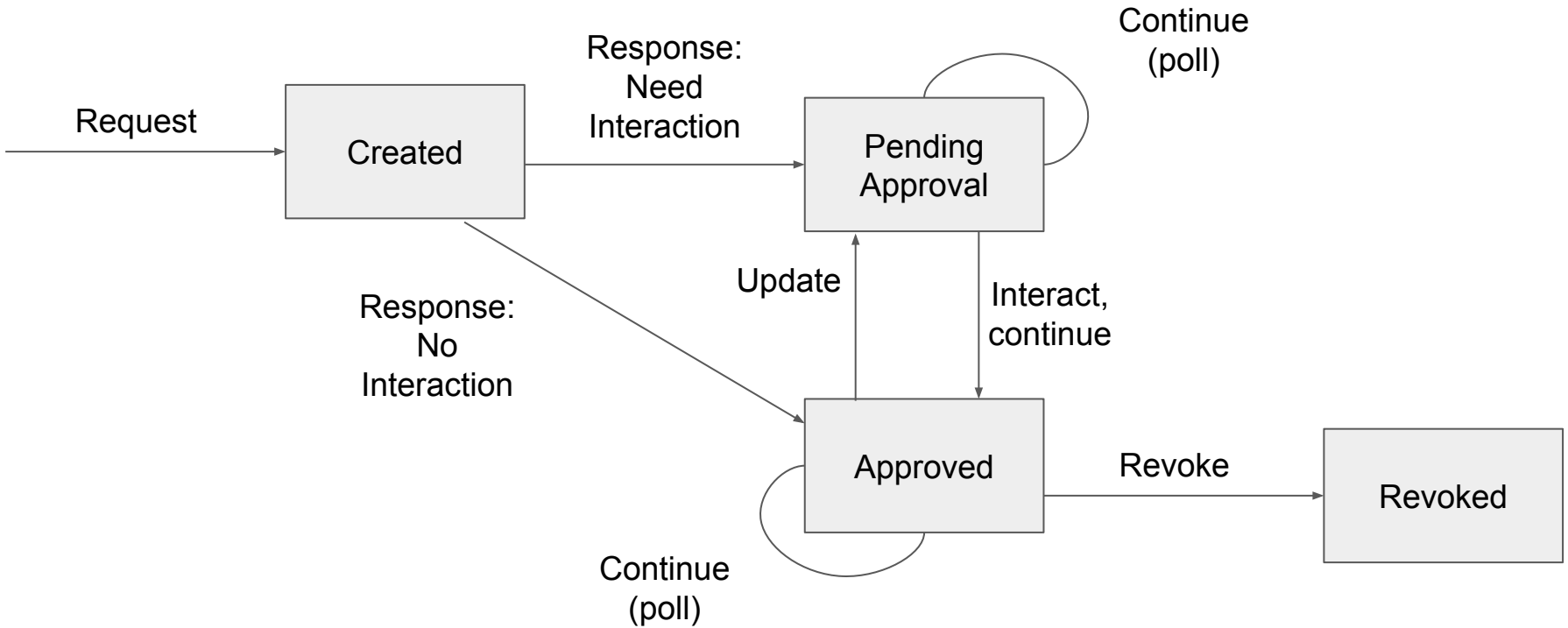
Draft Roadmap

- **Process the issue backlog**
  - https://github.com/ietf-wg-gnap/gnap-core-protocol/issues
  - https://github.com/ietf-wg-gnap/gnap-resource-servers/issues
- Clarity on what's allowed/not allowed at each step
- Key rotation
- Mandatory to Implement
- Extension discussion
  - IANA Registries
- What to do with JOSE
- Focus on the RS Draft

Grant Request Lifecycle

- Make lifecycle states explicit in the text
- Open questions:
  - Can you send “client” on a continuation request?
  - Can you send “interact_ref” multiple times?
  - Do you need to only use a “redirect” start method once, or can you do it multiple times?
- Editors have probable answers, will propose text to close these

Grant Request Lifecycle



Key Rotation Proposal

- WG feedback: feature is desirable
- Use different mechanisms for each presentation type
  - HTTPSig: multiple signatures
  - MTLS: PKI cert management
  - JOSE: wrapped JOSE objects
- Apply equally to each place that needs it
  - Client instance keys
  - Access token keys
- Reuse existing infrastructure and tooling where possible

Mandatory to Implement

- GNAP is very flexible (by design)
  - But most of the optional functions are negotiated at runtime
  - Always start the same way, can always get an answer (even if it's "no")
- What is the set of features/functions that are MTI
  - For an AS?
  - For a client instance?
  - For an RS?
- Should we have interoperability profiles (common combinations):
  - "Redirect-based web app"
  - "Mobile app with launch URL"
  - "Embedded device with polling"

Extensions

- What can be extended?
  - New fields in request and response
  - New data types for existing fields?
- Are extensions ignored if unknown?
- Ensure extensions don’t break the core
- Other general-purpose extension mechanisms:
  - End-user claim requests (VCs? OIDC?)
  - ‘access’ types (already discussed)
- Interaction start/finish mechanisms
  - And how they combine

JOSE

- Two JOSE-based key-proofing mechanisms kept in core
  - Detached JWS header
  - Attached JWS (replaces request body, when possible)
- Only JOSE dependencies in GNAP core
- Should these be their own spec?
- Could they be used outside of GNAP?

RS Draft: Future work

- Security/Privacy/Trust considerations
- Token model
  - Not a token format!

Open Discussion