

# Embedding GNAP

Justin Richer

IETF113, March 25, 2022

```
me.setHat(Hat.INDIVIDUAL)
```

# Embedding GNAP?

- Use case from BoF:
  - App calling a shopping API
  - RS realizes that user needs to update credit card info
  - RS needs to get user in front of an interaction page to update credit card info before going forward
- Use case from VC-API:
  - Wallet is presenting VC
  - Presentation API needs to gather some consent from user
  - Interaction could happen in a variety of ways
  - VC needs to be returned natively to the app

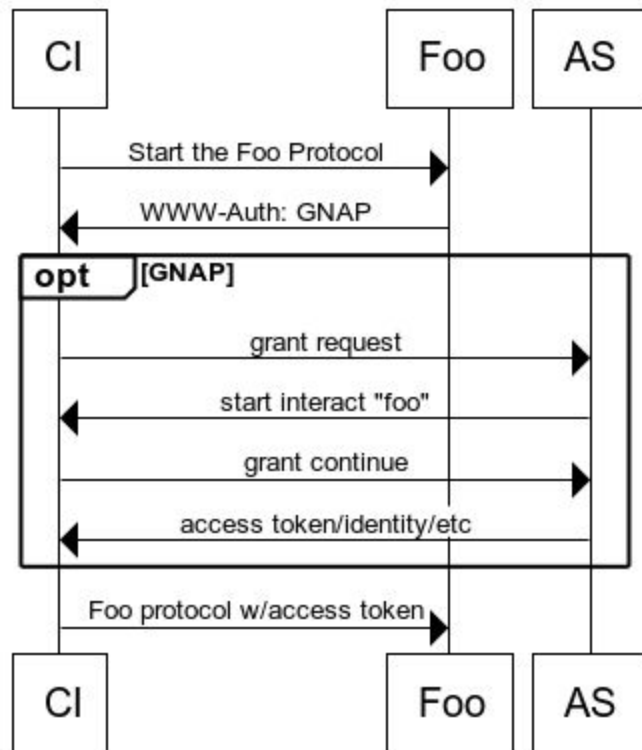
# Why?

- GNAP is never used completely on its own
  - Protecting an API
  - Getting subject information
  - Doing something else with an extension?
- How GNAP fits with other protocols is intentionally flexible
- Two models exist today
  - Traditional delegation (defined in core)
  - Embedding protocols inside GNAP (extension points defined in core)
- Is there an alternative model?
  - Embedding GNAP inside a protocol

# Traditional delegation

- Try the protocol and fail
- Go do security layer stuff
- Come back and try the protocol again

## Embed GNAP in Protocol (native)



# Traditional Delegation

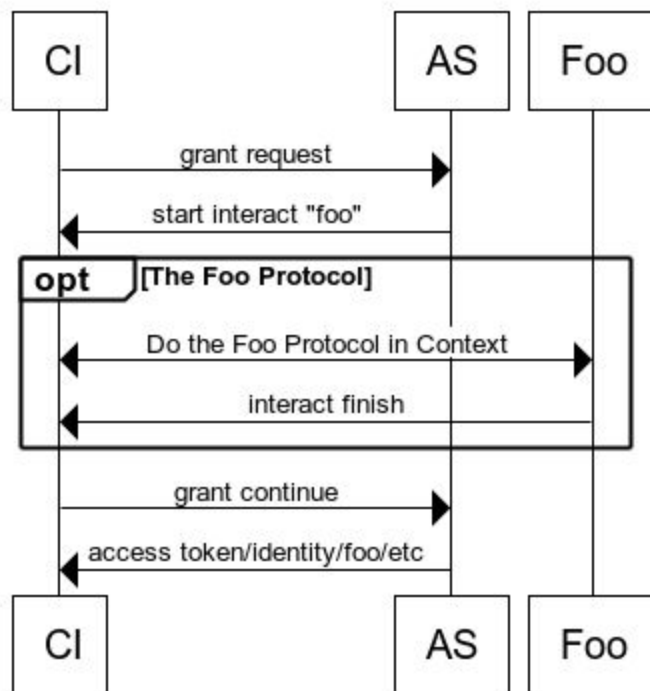
- Pros:
  - Separates layers
  - Separates concerns
  - Usable without modification to either GNAP or protocol
  - Can tie to multiple protocols at once
- Cons:
  - Chatty
  - Inefficient for simple cases

# Embedding the protocol inside of GNAP

- Start GNAP, but with extensions for the protocol
- During “interaction” phase, provide other information
- Return protocol response inside of GNAP response



## Embed protocol in GNAP



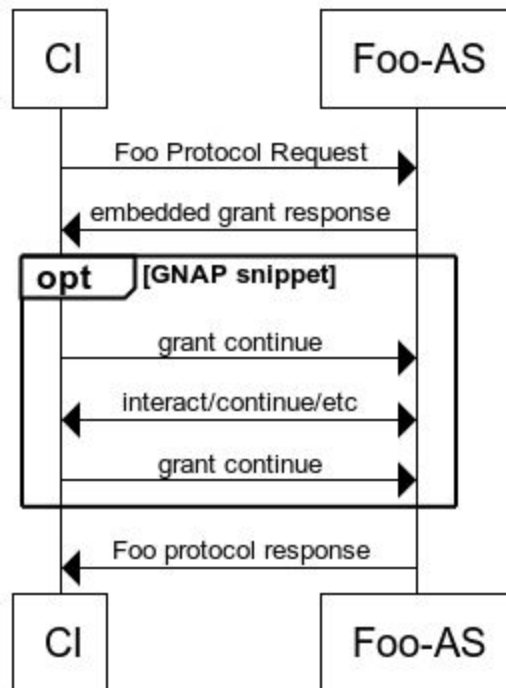
# Embedding the protocol inside of GNAP

- Pros:
  - Pattern used by OIDC on top of OAuth 2
  - GNAP has extension points to do this
  - Fits especially well for user information (authentication, identity, etc)
- Cons:
  - Specialized integration
  - Feels strange to start a security protocol when you're trying to just do your protocol
  - No longer speaking the protocol natively (especially for return)

# Embedding GNAP inside the protocol

- Try the protocol, and be told to start GNAP in-progress
- Skip the “request” phase and jump straight into interaction
- Instead of GNAP response, return protocol response

## Embed GNAP in Protocol (optimized)



# Embedding GNAP inside the protocol

- Pros:

- Efficient re-use of GNAP security structures (protocol doesn't have to re-invent them)
- Calling the protocol feels like normal until additional stuff is needed

- Cons:

- Specialized integration
- GNAP wasn't ever meant to start halfway through like this
- Every version of this is going to look different (based on the protocol it's in)
- Would be especially weird if the protocol is not HTTP/JSON based

# What do we do?

- Embedding protocols in GNAP:
  - Are the extension points sufficient?
  - Should we have at least one concrete extension before publication?
- Embedding GNAP in protocols:
  - Is there an extractable pattern we can write down?
  - Is there a specific application of this?
  - Does this need any changes to core?
  - Should we care at all?