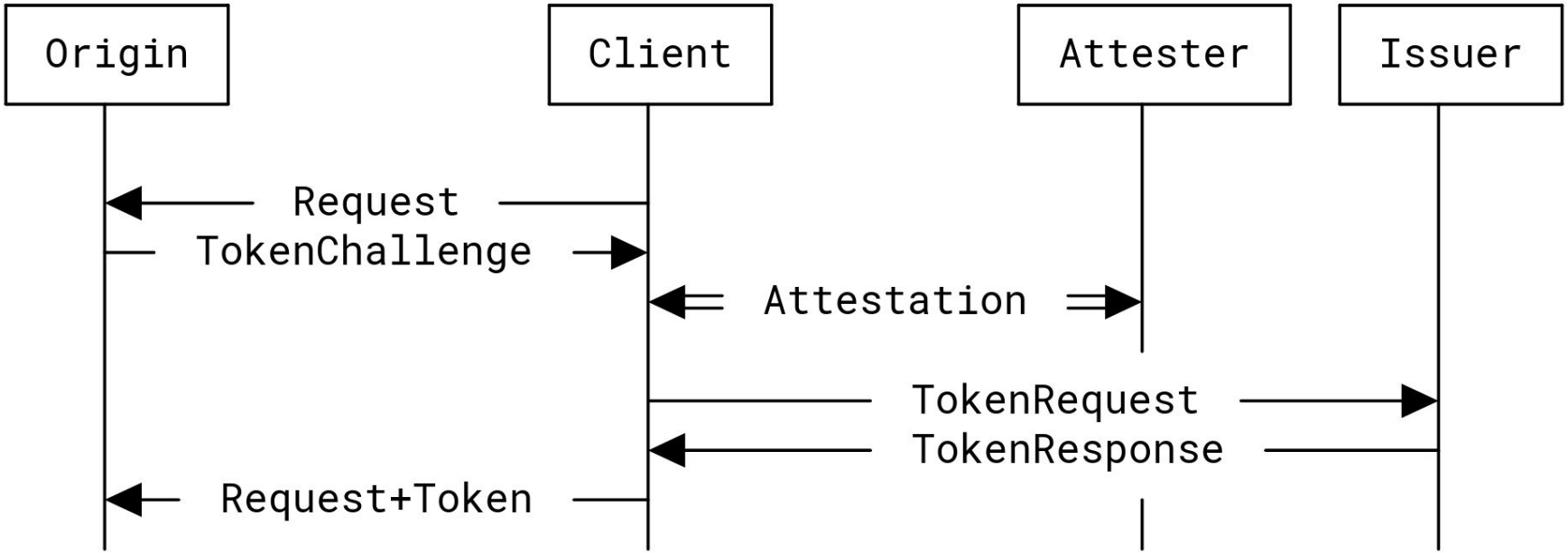
Privacy Pass Reverse Flow

`draft-meunier-privacypass-reverse-flow-04`

Thibault Meunier (Cloudflare)

Privacy Pass interim, May 2026, Online

Privacy Pass Architecture - RFC 9576



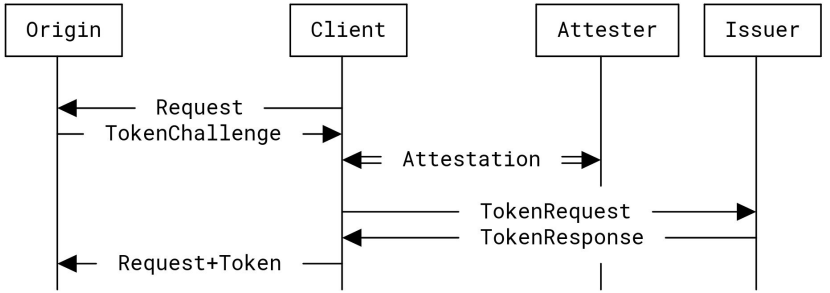
RFC 9576 - Limitations

1. If a client is misbehaving, how does an origin rate limit them?

2. If an origin has more resources, how can clients still contact them?

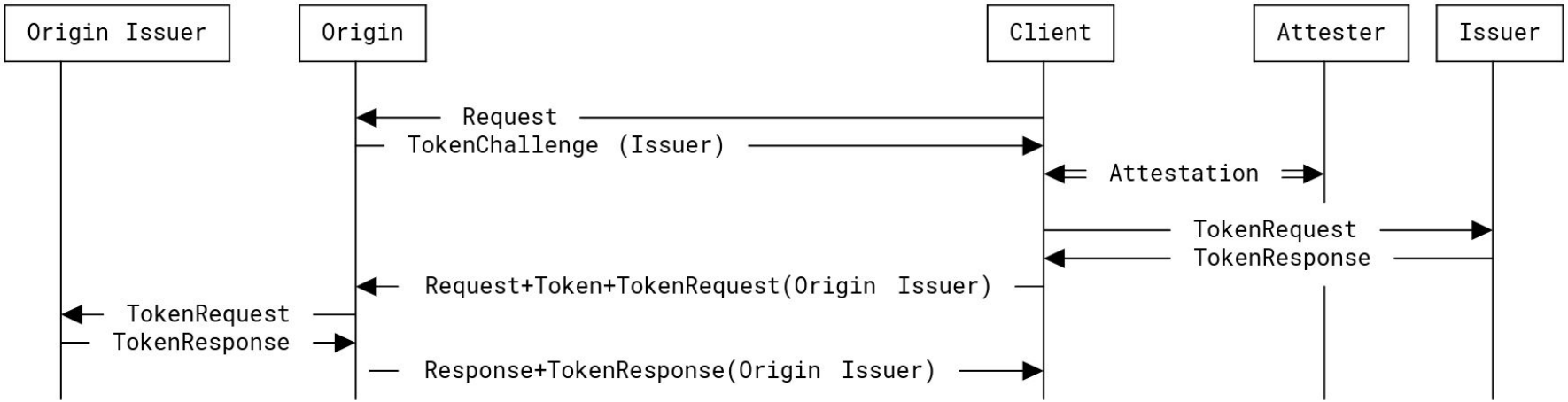
3. As an attester, how can I know if clients have been misbehaving?

4. As an origin, how do I limit requests going to the attester?



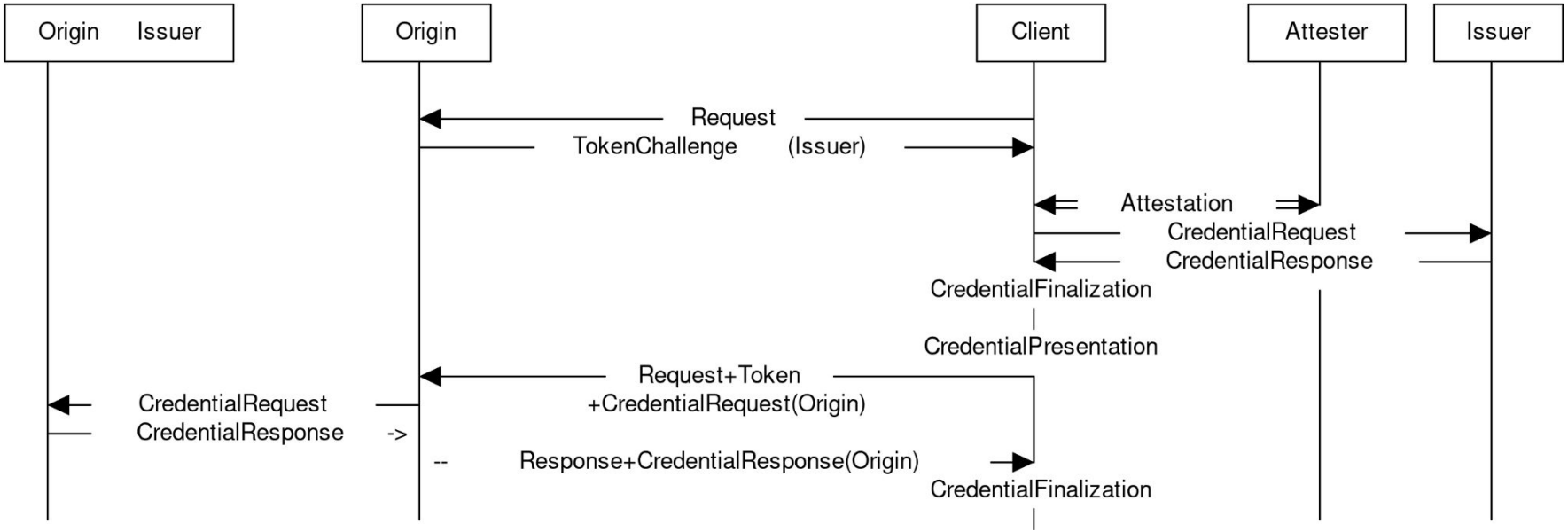
Reverse Flow -00

The first version uses the same architecture, and deploys it twice



Reverse Flow -04

The latest version uses the same architecture, with credentials, and deploys it twice



Difference since -00

1. Expand the motivation section

   Preserve rate-limit, bootstrap, credential type conversion

2. From tokens to credentials

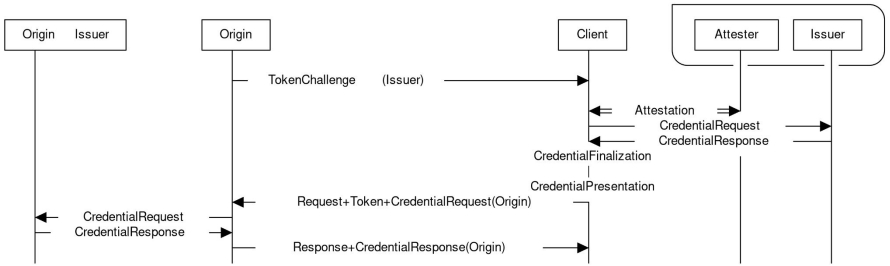
   Request, Response, Finalization, Presentation

3. Sharpen privacy guidance

   Metadata, bits per request, conversion

4. Concrete HTTP header

   PrivacyPass-Reverse. Used in ACT



Open questions

1. Do anonymous credentials work need architecture guidance in Privacy Pass?
2. Client state is not mentioned, but is key to new schemes
3. What is the right method to pass a CredentialRequest?
   HTTP header? body wrapping? others?
4. How to ensure privacy guarantees?

Draft: draft-meunier-privacypass-reverse-flow