One-time Tokens

via Macaroons

Neil Madden, IETF 121, Dublin

2024–11–07

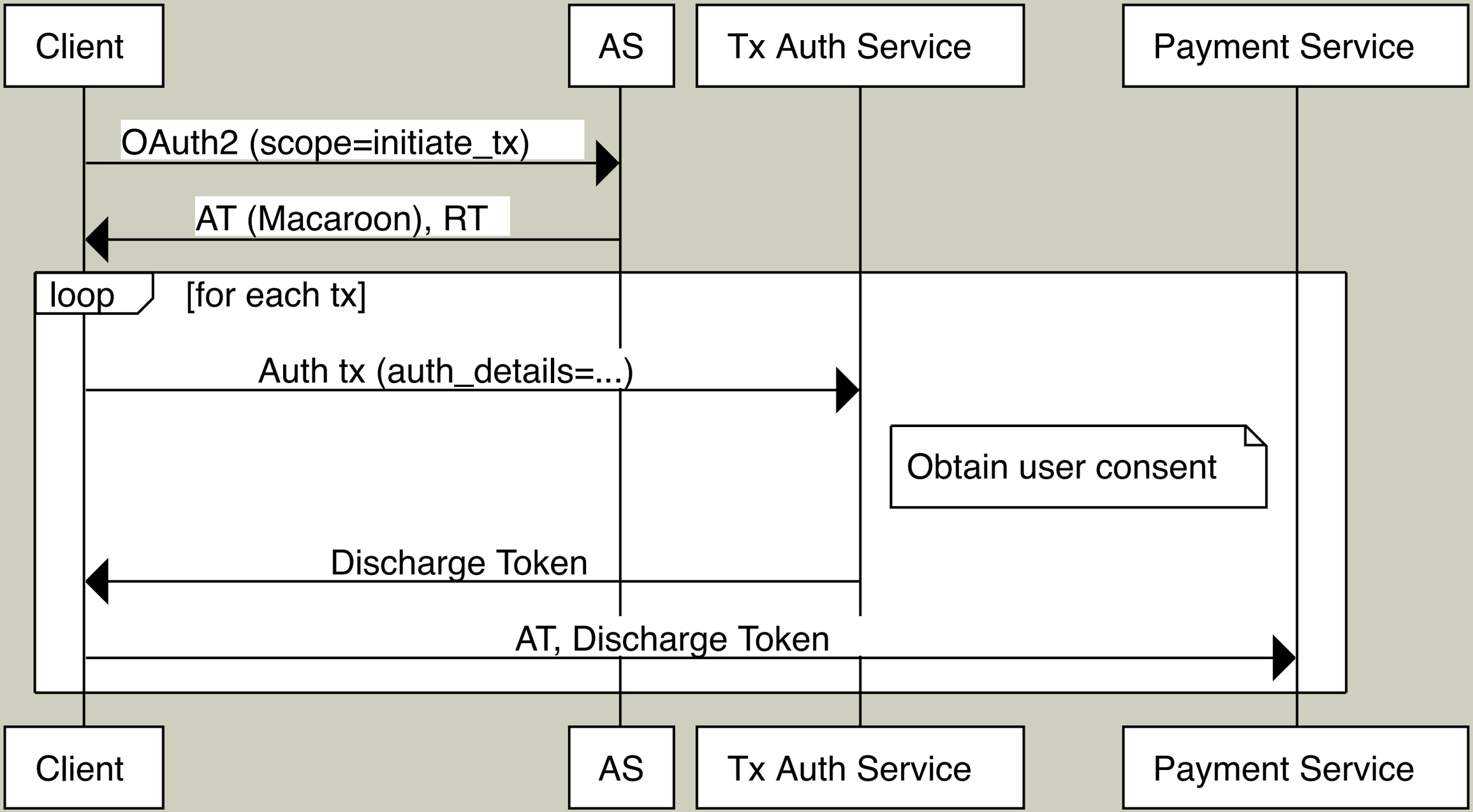
Patent disclosure

Aspects of this talk were patented by ForgeRock (now Ping) in US Patent **US11997207B2**

Macaroon 3rd-party caveats for transaction auth

- Initial OAuth2 flow for long-lived AT/RT as normal
  - Authorizes client to *initiate* transactions (eg start a payment)
- AT is a Macaroon with a “3rd-party caveat”
- Client must go to transaction auth service to retrieve “discharge token” to satisfy 3PC
  - Discharge token is bound to specific transaction

Transaction Auth



www.websequencediagrams.com

Again, with less pâtisserie

- Third-party caveats are like “cnf”, discharge tokens are like DPoP proofs
- But rather than the client producing the proof, it has to get it from a separate service (“third party”)
- The “cnf” is essentially a random symmetric key encrypted with PK of discharge service
- Discharge token is HMAC’ed with that key

One-time use without (additional) state

- In our example, the transaction auth service is presented with the `authorization_details` for this specific transaction
  - Often contains a unique key already (eg Stripe's idempotency key)
- The discharge token is cryptographically bound to that specific transaction:
  - Contains (a hash of?) the auth_details
- Can also sometimes use ETag/If-Matches

Summary

- Although flows look similar superficially, one-time auth and normal OAuth are very different
  - See Dmitry's slides
- Permission to *initiate* transactions is itself a privilege: should require consent
- Layer one-time auth *on top of* normal OAuth long-lived grants, not *instead of*