Authorization for IoT based on OAuth 2.0

draft-ietf-ace-oauth-authz-01

Göran Selander, Ericsson
ACE WG meeting, April 4, 2016

Authentication and Authorization for Constrained Environments

- This draft targets the ACE charter milestone:
  "Authentication and Authorization Solution"
- Previous versions shows that a profile OAuth 2.0,
  with some modifications, is feasible for IoT deployments
  in constrained environments [RFC7228]
- Recent work is about making the scope more precise
  → ACE Virtual Interim Meeting (March 2)
  → Followed by a discussion on the mailing list
- Purpose of this agenda item is to summarize the outcome and
  further progress the work

ACE solution summary

- /token and /introspect endpoints at the Authorization Server (AS)
  - /authz-info new endpoint at the Resource Server (RS)
- PoP tokens as access tokens
  - Client-RS authentication as new PoP method
- New claims enabling profiling of different deployments and new authorization information format
- Extensive use of CBOR/COSE, in particular CBOR Web Tokens

ACE Framework and ACE Profiles

› The ACE solution must support a variety of IoT deployment settings involving constrained and non-constrained devices (Use Cases RFC7744) while keeping the combinatorics down

› To accomodate this, we view the ACE solution as:

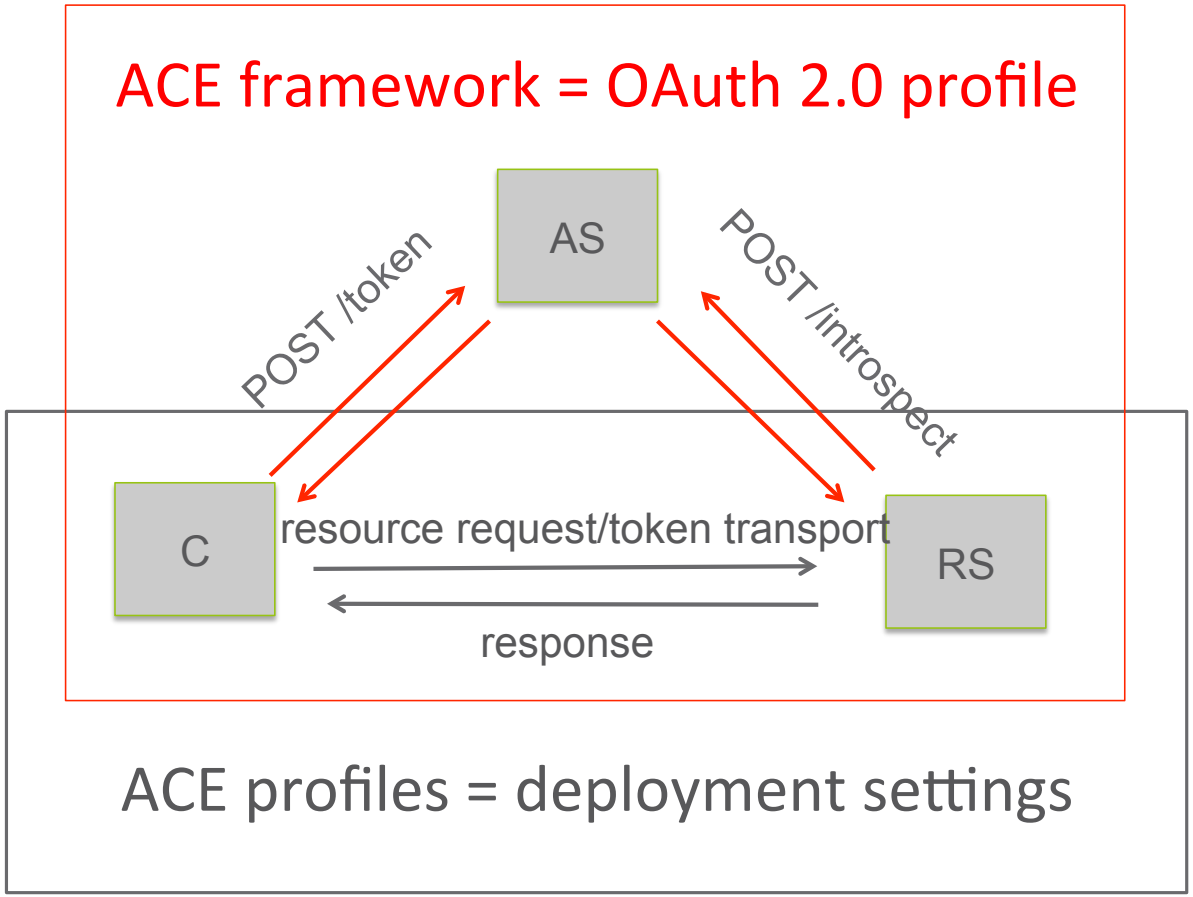
→ an ”ACE framework” defining an OAuth 2.0 profile, and

→ one or more ”ACE profiles” detailing certain deployments

› Different ACE profiles need not be interoperable, but the framework must be consistent with the different profiles

› A rich client may support multiple ACE profiles

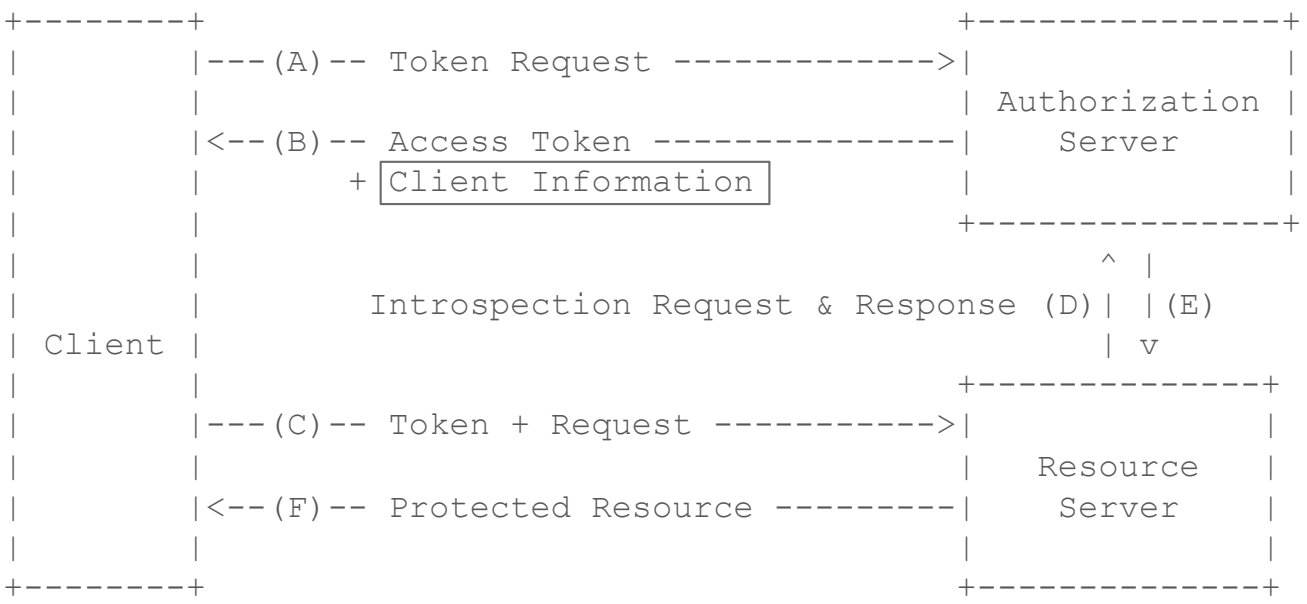
ACE Framework and ACE Profiles



Authentication, authorization and communication security **out of scope**

Authentication, authorization and communication security **in scope**

ACE Framework Basic Flow



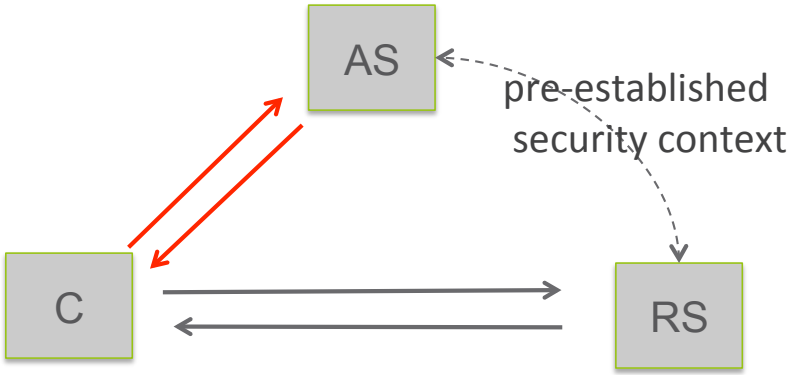
› Variations in the flow depend on

a) local token verification in RS or not (= introspection or not)

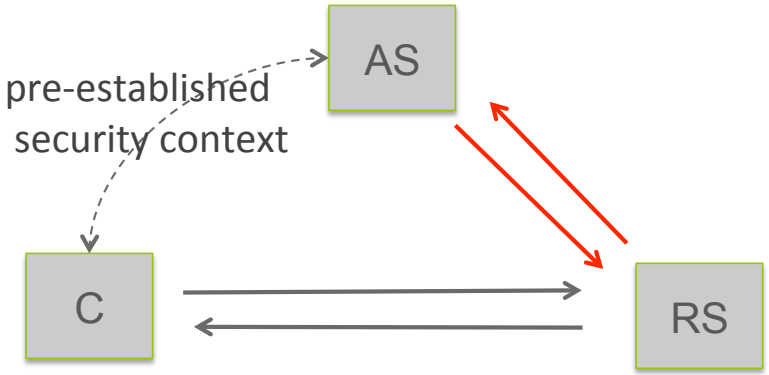
b) specifics of Client-Resource Server interaction (as detailed in a profile)

1. Deployment Options

RS verifies token locally

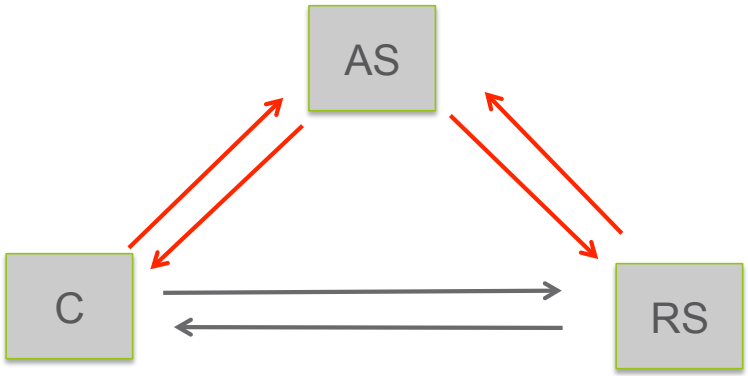


RS uses token introspection



- Agreed to merge examples with local token verification and introspection, respectively. “group communications” and “tokenless” may be separate drafts. Should we keep token introspection in this draft?
- **Proposal:** Keep introspection in this draft.
- Motivation: The same information elements needs to be transferred in both cases. Demonstrates support for multiple deployment scenarios.

2. Transport layer security or application layer security



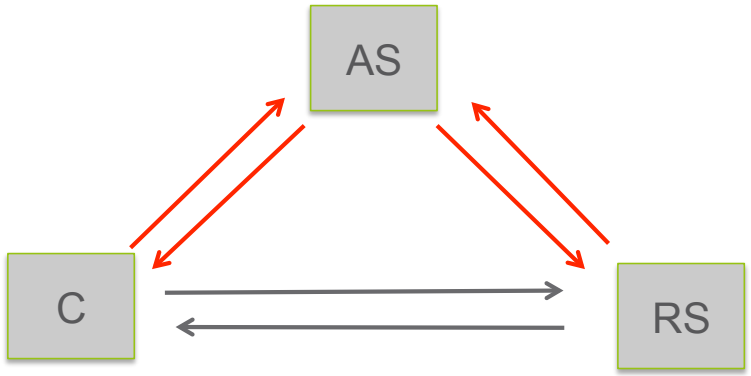
Interim discussion:

- 3 votes for both, different preferences of order.
- 2 votes for application layer security only
- 1 vote for transport layer security only

› The ACE framework should support the use of transport and/or application layer security. Should we include different security layer solutions in this draft?

› **Proposal:** Include profiles defining one transport layer security solution and one application layer security solution. The transport layer security solution is DTLS. The application layer solution should be COSE based.

3. Credential Options 1(2)



Interim discussion:

- No credentials are MTI
- At least use of RPK should be specified

AS supports key establishment via access token and client information.
Which credentials should be specified in this draft? Aspects to consider:

1. Transport of key/credential from AS to C and RS, respectively
2. Binding of key/credential to access token/rights

› **Proposal:** Define transport of RPK and PSK. (Motivation: PSK is favorable for constrained devices and feasible in the TTP setting)

› Define binding to RPK, PSK and Certificate. See next slide.

3. Credential options 2(2) -- sanity check

The ACE framework should support **first time** access as well as **repeated** requests

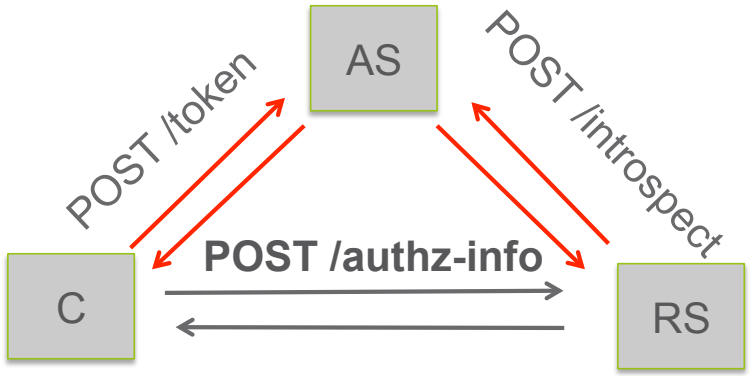
1. Client request access to a resource for the **first time**
   - Including authentication and key establishment of previously unknown client
2. Client request access to the **same resource** but requiring **different rights**
   - Using established keys without re-provisioning credential or re-authenticating
3. Client request access request to **the same RS**, but **different resources**
   - Without re-provisioning credential or re-authenticating

Proposal: The ACE framework should support the use of a **credential identifier** replacing the actual credential in the token request and in binding to the access rights

› Ex. 1: Client previously authenticated with RPK makes additional token request, **referencing RPK** used in the first place – instead of transporting public key again.

› Ex. 2: Client previously authenticated with certificate makes additional token request, **referencing a certificate** used to authenticate to the RS.

4. Token Transport

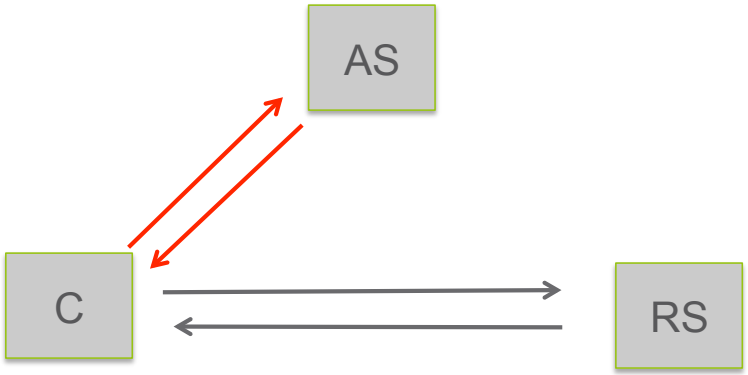


Interim discussion:
A number of options considered with different properties
(see Interim slides)

- Since the access token may contain information nessary for the C and RS to authenticate and establish communication security, it needs to be transported before or within the authentication protocol
- **Proposal:** Define only POST /authz-info in the ACE framework. Allow profiles to define alternatives in separate drafts

5. RS-synchronized time

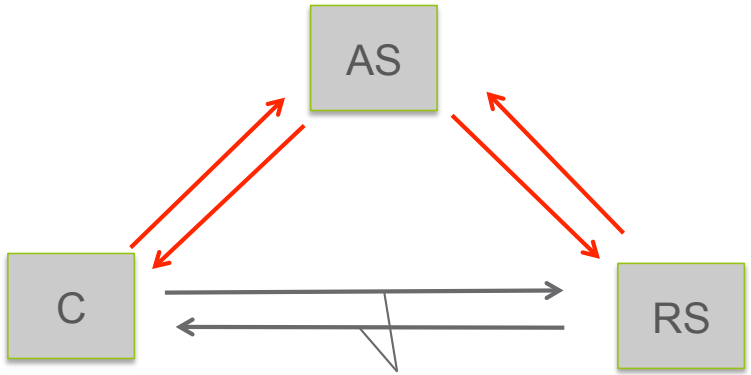
RS verifies token locally



Interim discussion:
One example of nonce-based freshness was presented and got support. Authenticated key exchange has been proposed on the mailing list.

- The RS needs to verify that request and authorization information is valid. in many cases is synchronized time sufficient. In case RS verifies token locally, and have no synchronized clock, additional mechanism is needed.
- **Proposal**: Define a nonce-based mechanism for aligning time in RS with time in AS in a separate draft

6. CoAP vs HTTP



CoAP or HTTP?

Interim discussion:

- 3 votes for CoAP only
- 2 votes for both

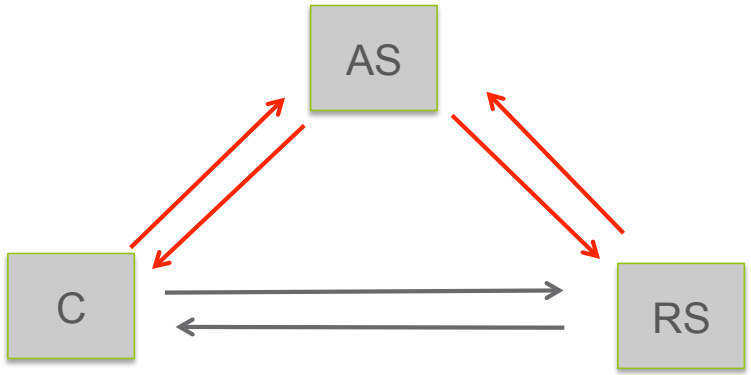
(The authors are divided here)

› Should we specify the solution in terms of CoAP and/or HTTP?

› Note that for a each protocol, different solutions need to be specified depending on the security protocol used.

› **Proposal:** The ACE framework should support the use of CoAP or HTTP. The CoAP profile should be included in this draft. This may be extend later on with HTTP or in a separate draft.

7. CBOR vs JSON, and COSE vs JOSE



Interim discussion:

- 3 votes for COSE/CBOR only
- 3 votes for both, different preferences of order

(The authors are divided here)

› Should we specify the solution in terms of CBOR or JSON?

› Note that not only CWT contains CBOR, e.g. also client information.

› **Proposal:** The ACE framework should support the use of CBOR/COSE and JSON/JOSE. A CBOR/COSE profile should be included. This may be extend later on with JSON, or in a separate draft. If CWT and JWT are exchangable without changing this draft, then both should be referenced.

Thank you!

Questions/comments?