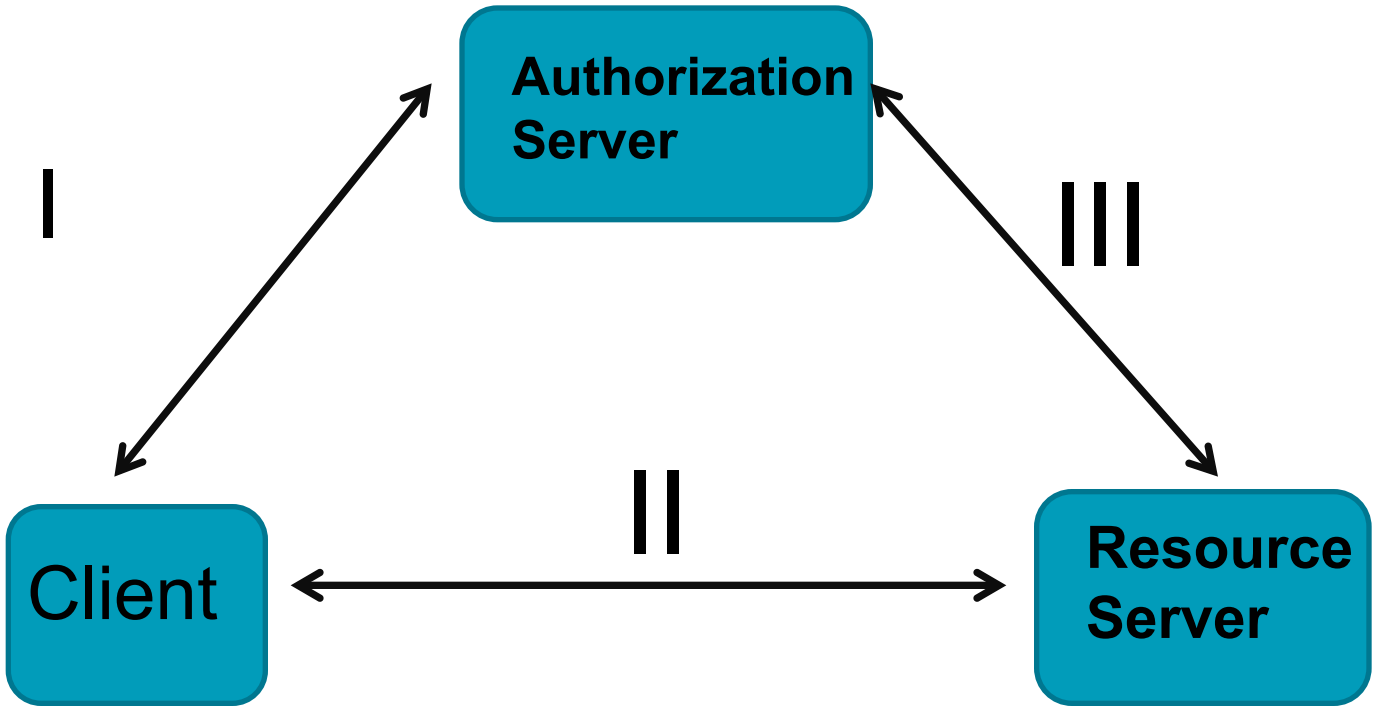
IETF OAuth Proof-of-Possession

Hannes Tschofenig

Status

- Finished various specifications, including
  - OAuth Core: RFC 6749
  - Bearer Tokens: RFC 6750
  - Security Threats: RFC 6819
- Discussion about an enhancement to Bearer Token security (now called “Proof-of-Possession”) since the early days of the working group.
- Design Team work late 2012/early 2013, which lead to requirements, use cases, and solution strawman proposals.
- Work on solution documents lead to new work items.

Architecture



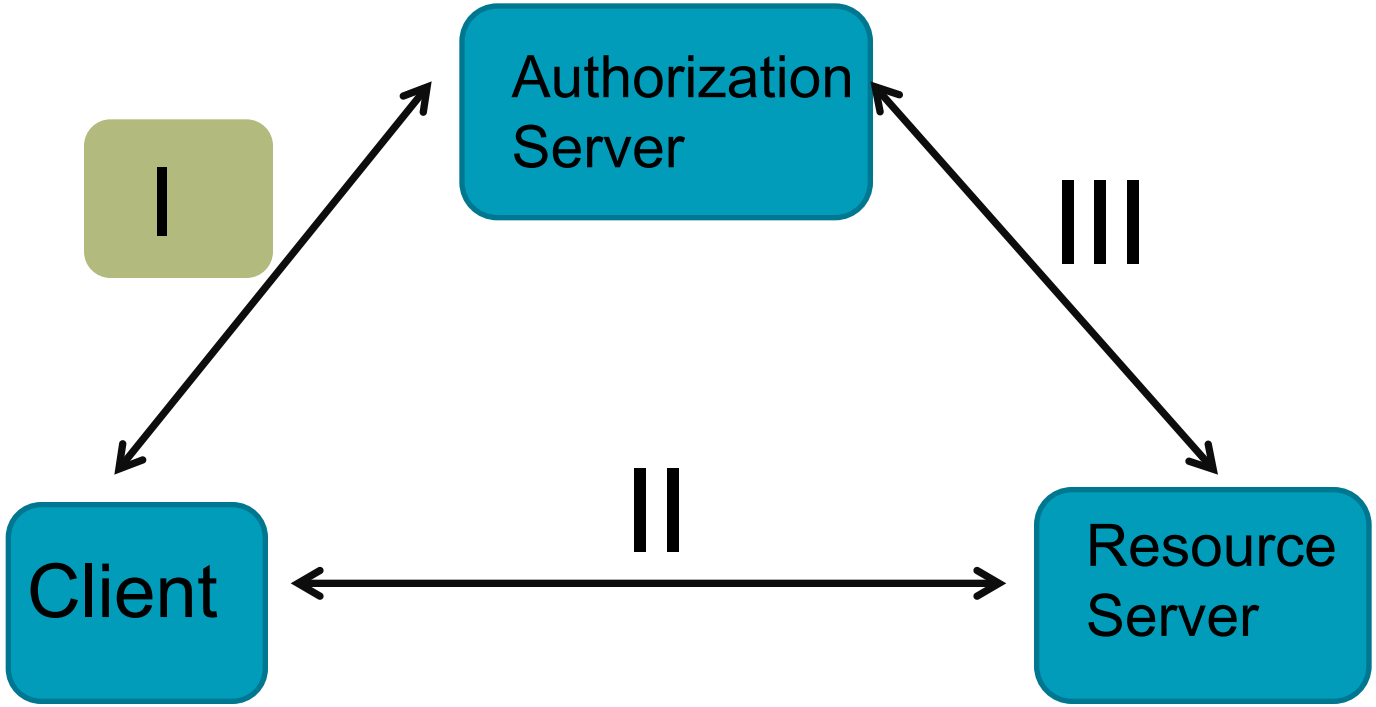
Relevant document:

http://datatracker.ietf.org/doc/draft-ietf-oauth-pop-architecture/

AS <-> Client Interaction

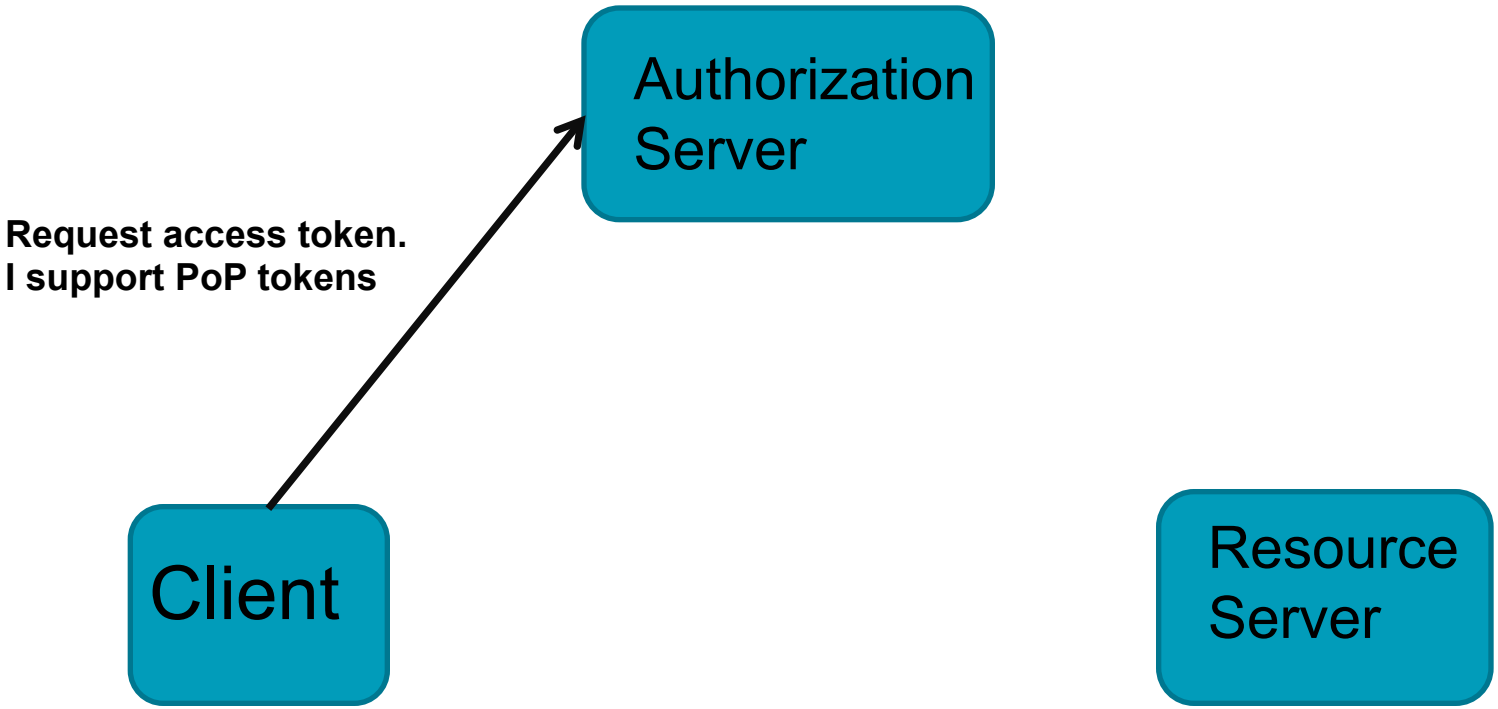
Variants:

- **Key Distribution at Access Token Issuance**
- **Key Distribution at Client Registration**



Relevant specifications:
http://datatracker.ietf.org/doc/draft-ietf-oauth-pop-key-distribution/
http://datatracker.ietf.org/doc/draft-ietf-oauth-proof-of-possession/

AS <-> Client Interaction
Example: Symmetric Key



AS <-> Client Interaction
Example: Symmetric Key

AS creates PoP-enabled access token

Authorization Server

Client

Resource Server

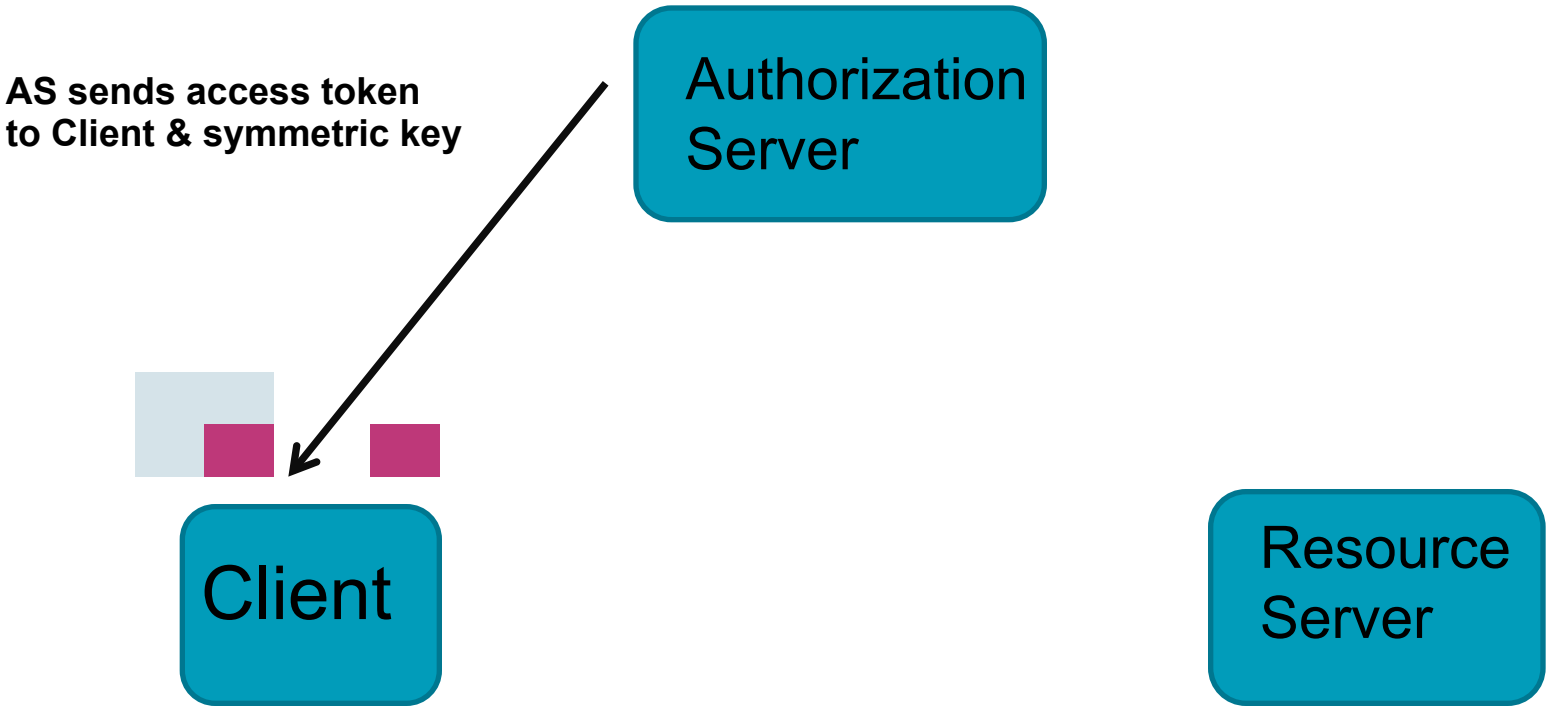
PoP Token: Symmetric Key Example


```
{
    "alg":"RSA1_5",
    "enc":"A128CBC-HS256",
    "cty":"jwk+json"
   }
{
    "iss": "https://server.example.com",
    "sub": "24400320",
    "aud": "s6BhdRkqt3",
    "nonce": "n-0S6_WzA2Mj",
    "exp": 1311281970,
    "iat": 1311280970,
    "cnf":{
     "jwk":
      "eyJhbGciOiJSU0ExXzUiLCJlbmMiOiJB
       MTI4Q0JDLUhTMjU2IiwiY3R5IjoiandrK
       ... (remainder of JWE omitted for brevity)"
     }
   }
```


Binds a symmetric key to the access token


```
{
 "kty":"oct",
 "alg":"HS256",
 "k":"ZoRSOrFzN_FzUA5XKM
      YoVHyzff5oRJxl-IXRtztJ6uE"
}
```

AS <-> Client Interaction
Example: Symmetric Key



AS <-> Client Interaction

- AS needs to bind a key to the access token.
  - Key can be an fresh and unique symmetric key, or
  - (ephemeral) public key
- This requires two extensions:
  - New elements within the JWT to include the (encrypted symmetric key) or the public key. JWT is also integrity protected.
  - Mechanism for conveying ephemeral key from AS to client and for client to provide directives to AS.
- Details in draft-ietf-oauth-pop-key-distribution
  - Transport symmetric key from AS to client.
  - Transport (ephemeral) asymmetric key from AS to client.
  - Transport public key from client to AS.
  - Algorithm indication

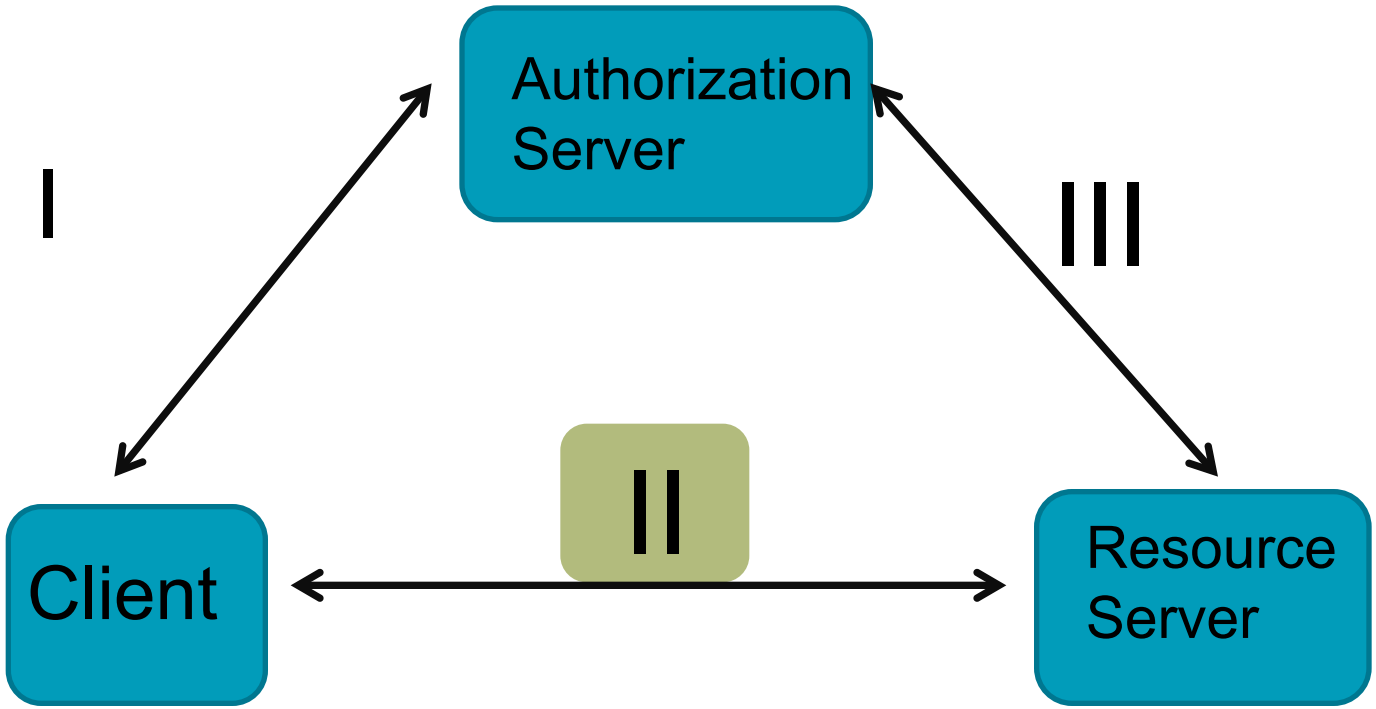
Dynamic Client Registration

- Attempt to simplify developer interaction with AS when they deploy client applications.
- Today, developers need to register various parameters (manually), such as
  - Authentication mechanism & client authentication credentials
  - Redirect URIs
  - Grant types
  - Meta data (client name, client logo, scopes, contact information, etc.)
- Also allows meta-data, including public keys, to be uploaded to AS.
- Two documents:
  - draft-ietf-oauth-dyn-reg
  - draft-ietf-oauth-dyn-reg-metadata
- WGLC in progress.

Client <-> RS Interaction

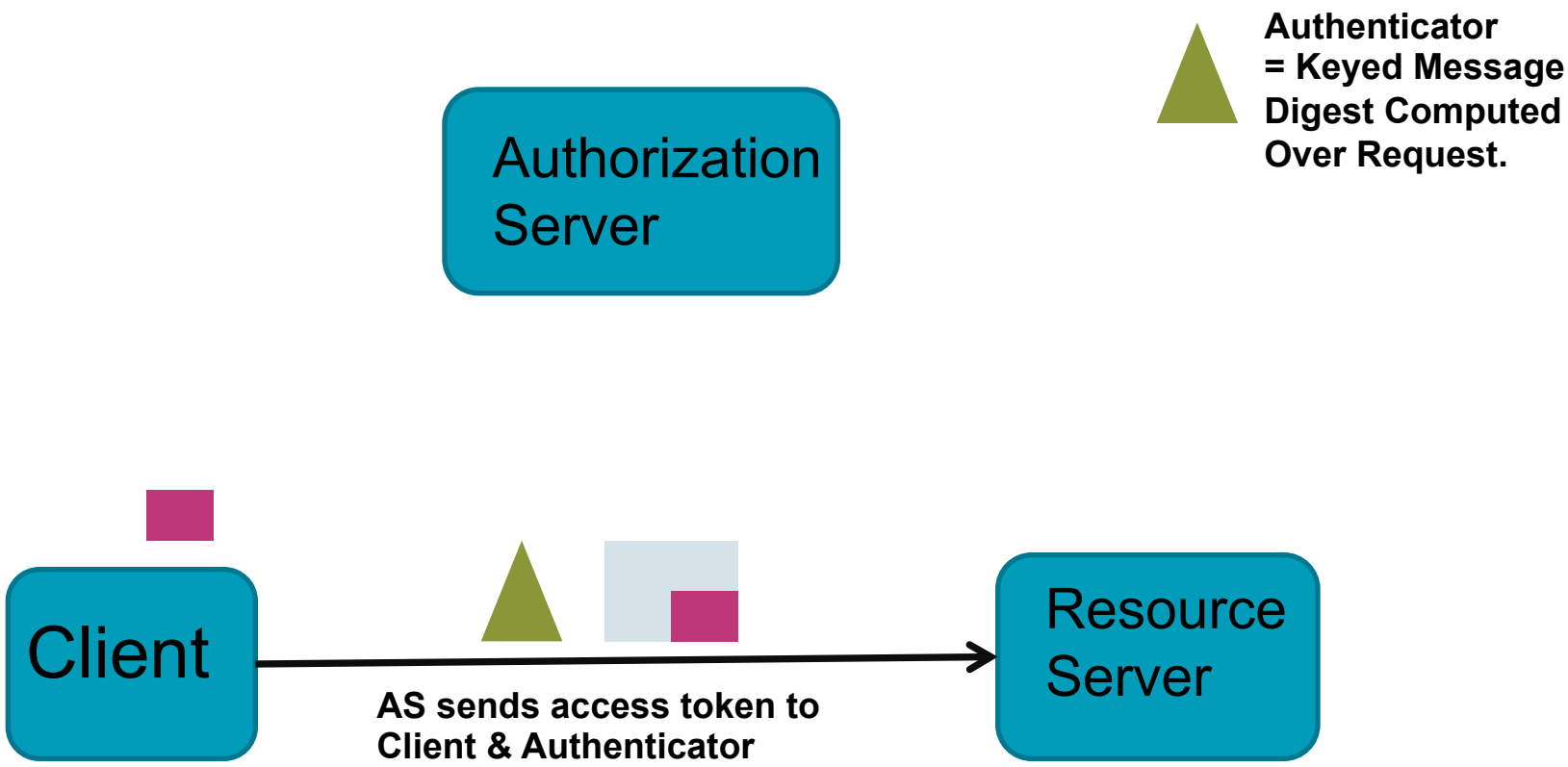
Building Blocks:

a) **Proof of possession of PoP key**
b) **Message integrity (+ Channel Binding)**
c) **RS-to-client authentication**

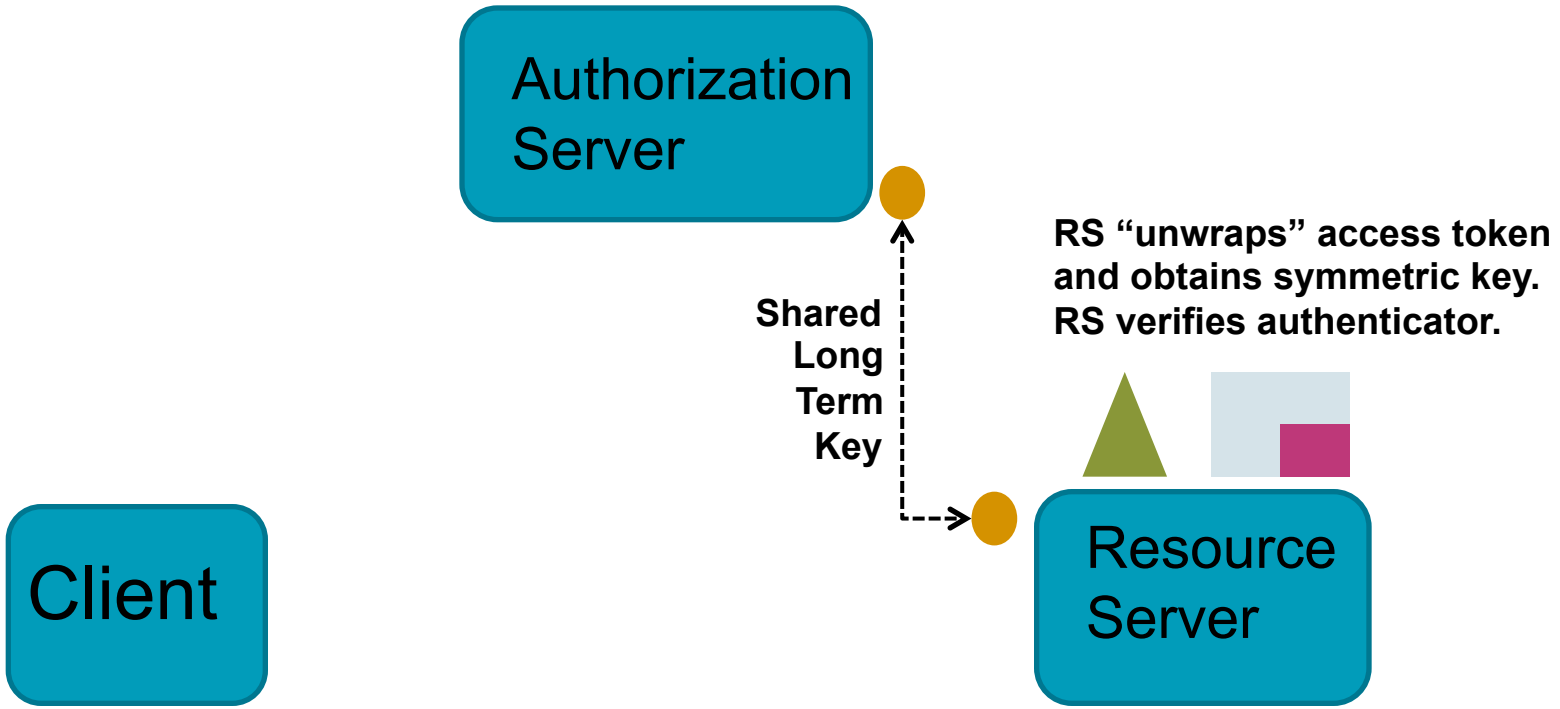


Relevant specification : http://datatracker.ietf.org/doc/draft-ietf-oauth-signed-http-request/

AS <-> Client Interaction
Example: Symmetric Key



AS <-> Client Interaction
Example: Symmetric Key

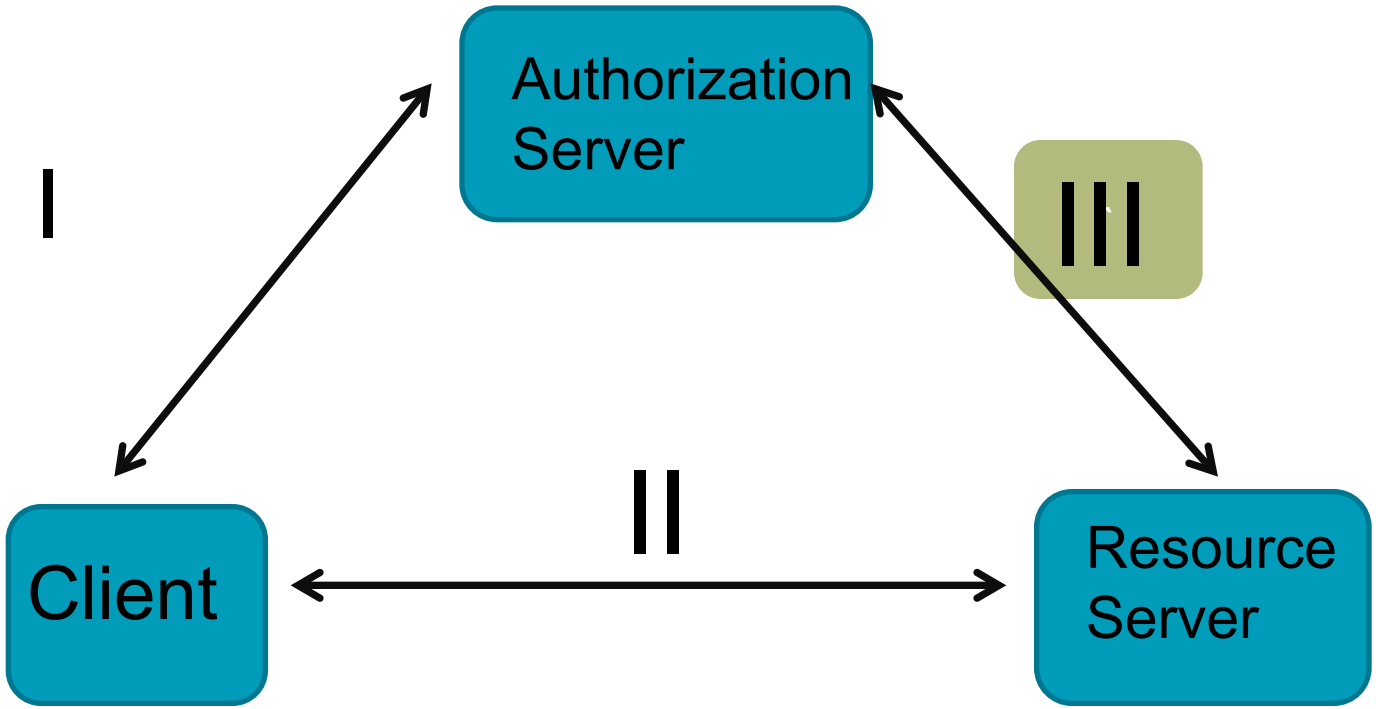


Channel Binding

- Channel bindings bind the application layer security to the underlying channel security mechanism.
- Various approaches for providing channel bindings:
  - PoP public key use in TLS (as described in HOTK draft)
  - **tls-unique**: TLS Finish message
  - **tls-server-end-point:** hash of the TLS server's certificate:
- Currently, no channel bindings described in <draft-ietf-oauth-signed-http-request>
- Be aware: New attacks have been identified with TLS-based channel bindings, see http://www.ietf.org/proceedings/89/slides/slides-89-tls-3.pdf

RS <-> AS Interaction [optional]

Variants:
a) Token introspection
b) Out-of-band



Relevant specification: http://datatracker.ietf.org/doc/draft-richer-oauth-introspection/

Next Steps

- Reviews for the document bundle needed.
- Open Issues will be added to the WG tracker.
- Main issues with the client<->resource server communication. Challenges:
  - Dealing with intermediaries modifying headers
  - Offering flexibility to developer
  - Reducing payload replicating
  - Minimizing canonicalization
  - Authentication of the server to the client
  - Channel binding functionality