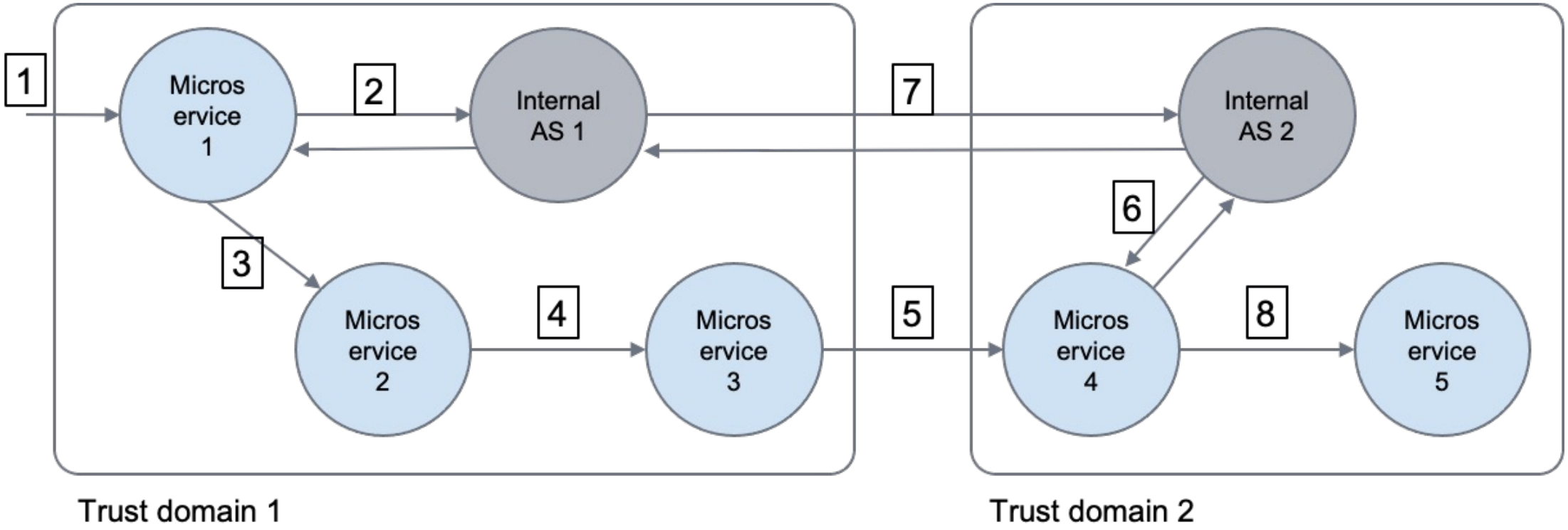
OAuth Identity Chaining

Kelley Burgin, MITRE

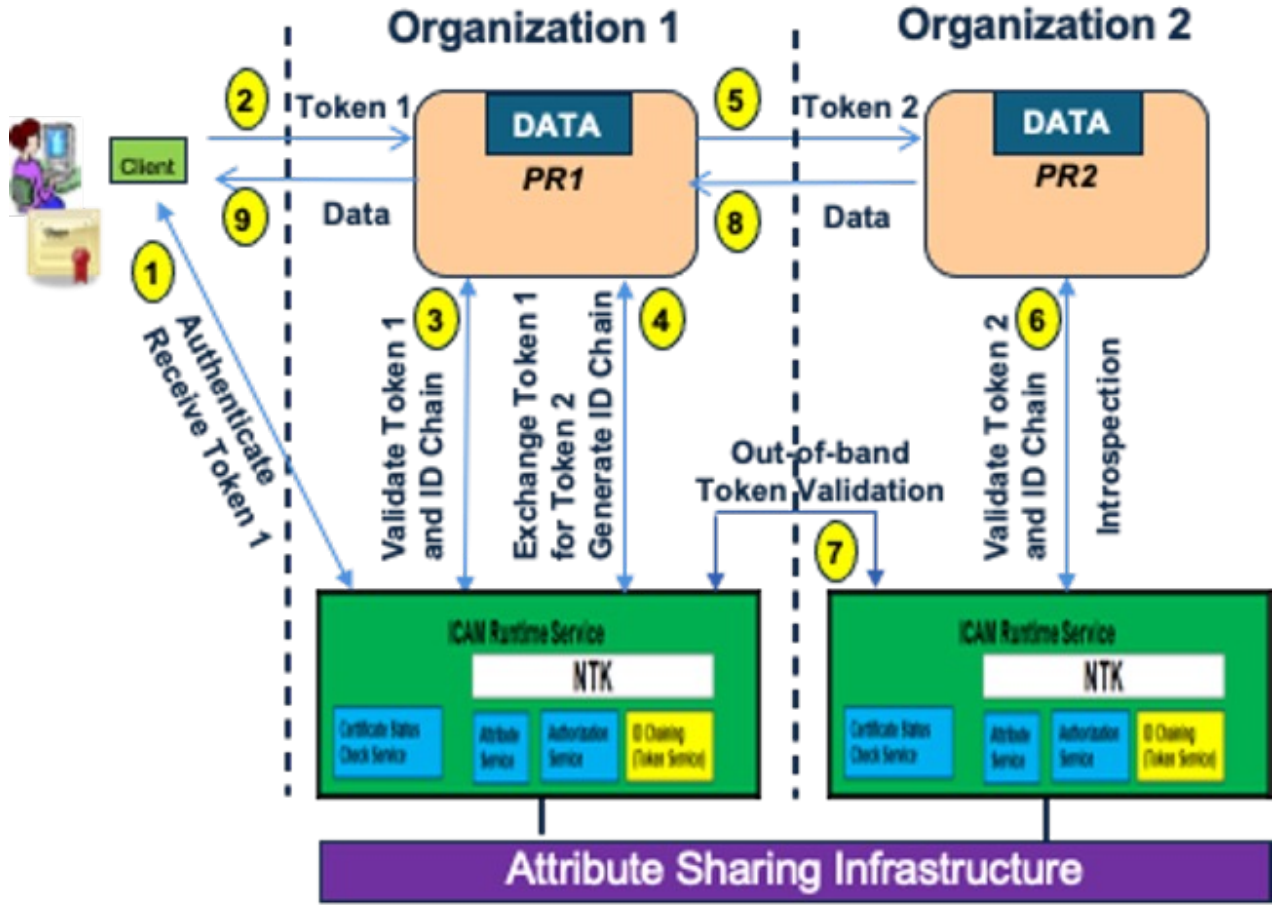
kburgin@mitre.org

November 2022

Proposed Solution Architecture



Proposed Solution Architecture (OAuth view)



AS to AS relationship
levied at the end

Problem (with new constraint)

- Use Case:
  - An OAuth client makes a request to a protected resource PR1, but PR1 needs to access a second PR2 in to answer the client's request.
    - If PR1 and PR2 are in the same "trust boundary", just an extension of Token Exchange (not discussed further)
    - If PR1 and PR2 are in different "trust boundaries", much more complex. Discussed here
- Problem with applying Token Exchange (different "trust boundaries")
  - How to obtain a token for PR1 to use at PR2.
  - Assumptions:
    - Clients authenticate to servers using mTLS, so "cnf" field is easy to fill by ASs
    - Access tokens are **sender-constrained** (and signed…)
    - We want additional logic in the ASs rather than the PRs

Solution

- The new **sender constrained** access token received by PR1 from token exchange (for use at PR2)
  - Has PR1 as the “client_id”
  - Is sender-constrained to PR1’s PKI certificate using “cnf” claim
  - Is audience constrained to PR2 using “aud” claim
  - Contains “act” claims that contain the “sub” and “iss” claims from previous tokens

Sender constrained

For verification by PR2

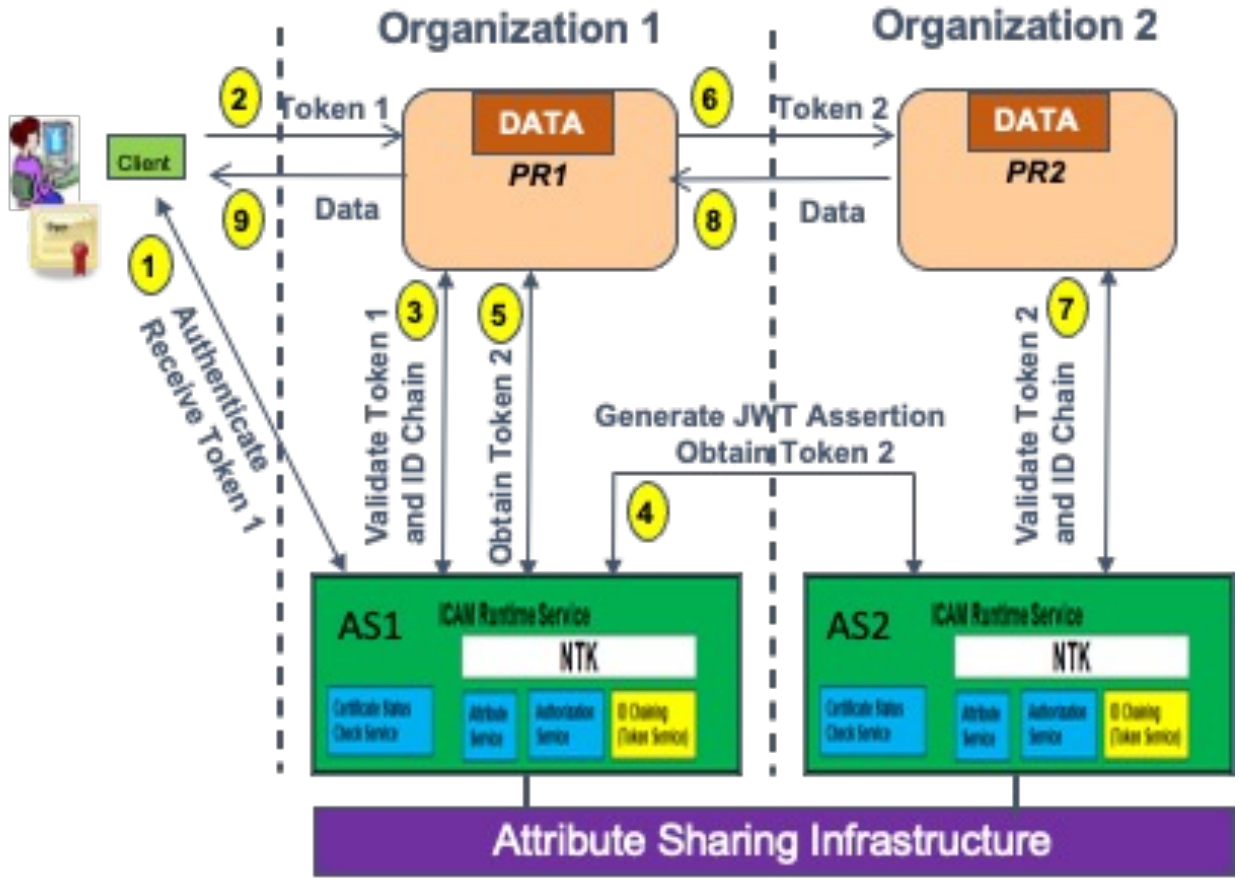
Alternate Solution (draft-burgin-jenkins-identity chaining)

Summary:

- PR1 performs token exchange with AS1
  - AS1 generates a JWT assertion that it uses to obtain the access token from AS2
- AS2 generates the token and returns it to AS1, who returns it to PR1 to complete the token exchange request

Problem:

- We need PR1 info, in this example, "`client_id`" and "`cnf`" fields in the token (sender constrained)
- So AS1 needs to pass these two bits of information to AS2 in its request to AS2 for the token



AS to AS relationship levied at the beginning

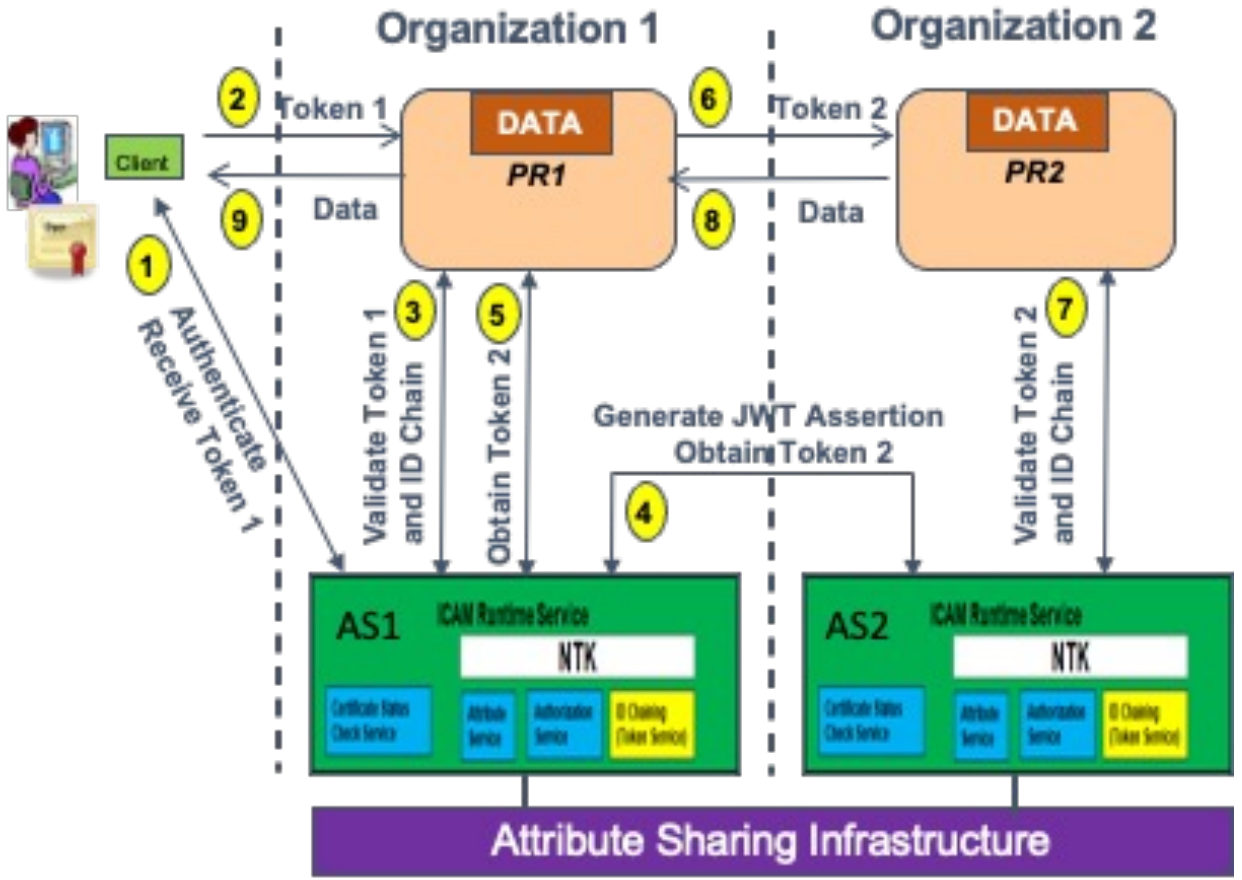
Alternate Solution [2]

Solution:

- Define a new private use OAuth claim

```
chained_id {
        "client_id": "PR1"
        "cnf": [Hash of PR1 PKI cert]
}
```

- AS1 includes "`chained_id`" in its token request to AS2
- AS2 includes "`client_id`" and "`cnf`" claims are populated with the values of PR1 obtained in the "`chained_id`" claim



Benefits

- Complete history included in "act" claims
- Iterated calls do not result in large final token
- Additional logic in the ASs, not the PRs