Demystifying API Security in the Application Network

September 2019

Steve Roberts

MuleSoft Solutions Architect

Security is hard, right?

Boundary & network controls (e.g. firewalls, whitelisting)

- APIs may move in the future, breaking security model.
- What is an “internal API” in Cloud?
- Risk of breach to trust boundary.

Zero-trust network (e.g. mTLS)

- Complex to set up and expensive to maintain.
- Untenable for unknown (public) clients.

Opaque identity tokens (e.g. Basic Auth)

- Expensive to validate in real-time.
- Cannot provide rich identity information.

Security is easy, right?

Cloud / iPaaS approaches require elevation of security controls to the application layer - abstract away from variable infrastructure.

- Each API takes control of its own security domain; it decides who to trust.
- Overall identity management remains centralised, with a lower management cost.
- Credential validation is a constant-time operation.

OAuth 2.0 & JWTs

Becoming the de-facto industry approach to API security

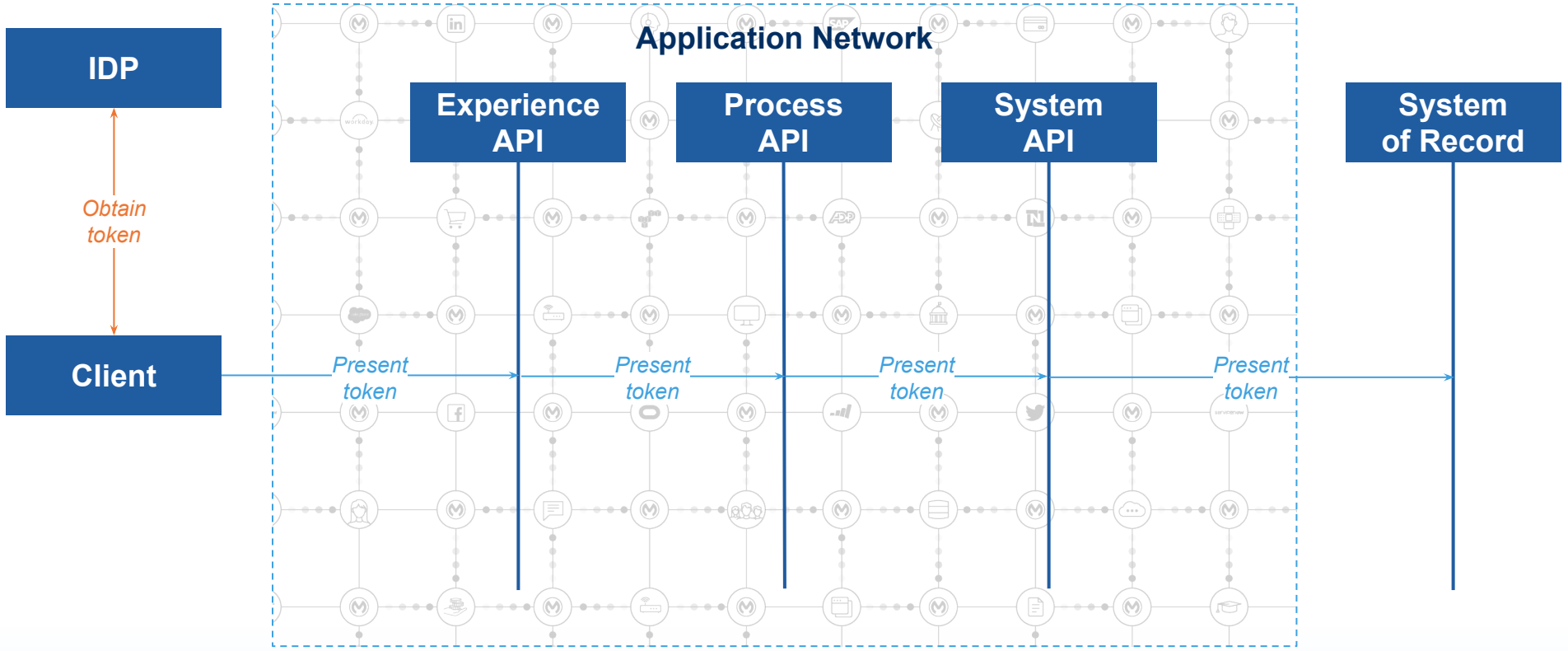
OAuth 2.0

- Defines common approach to 2- and 3-legged authentication.
- Usable for user and system based principals.
- Does NOT define how authentication should be performed, nor format of token.

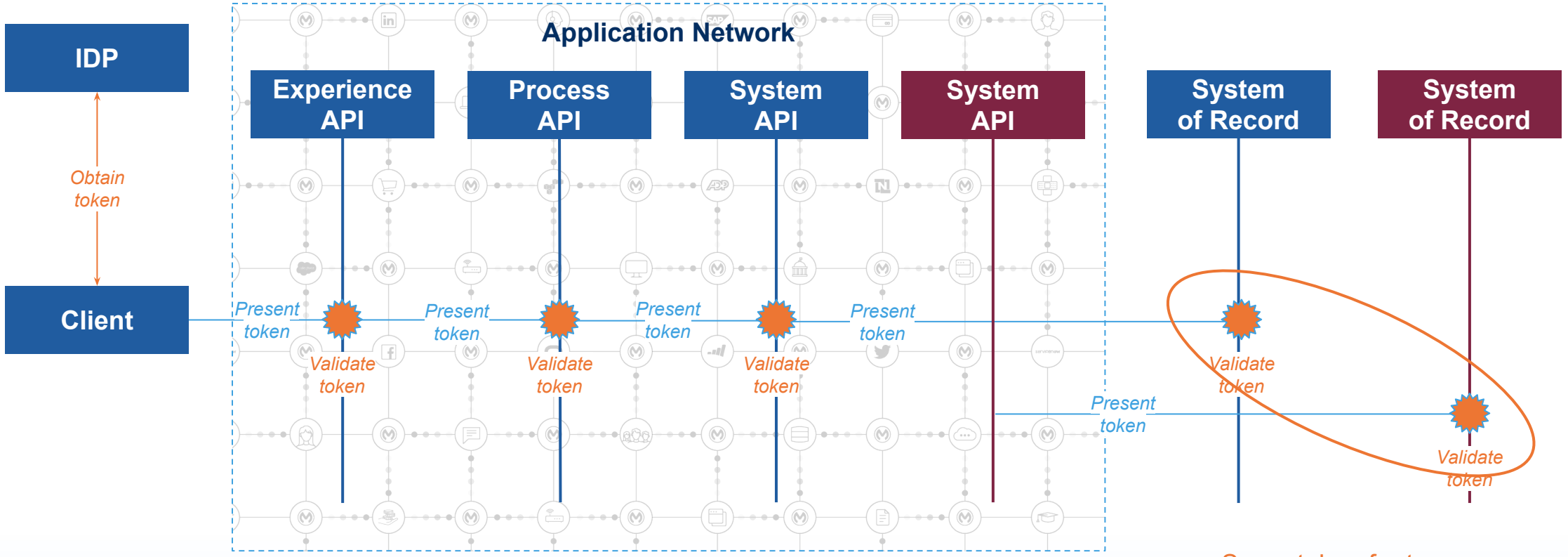
Java Web Tokens (JWTs)

- Cryptographically signed and transportable JSON object.
- Self-validating, can be validated “in-place”.
- Extensible and can include custom claims.
- Anypoint offers an out-of-the-box policy for validating JWTs.

So all we need is a token then?



So all we need is a token then?



Same token for two systems of record?
Conflict of concerns

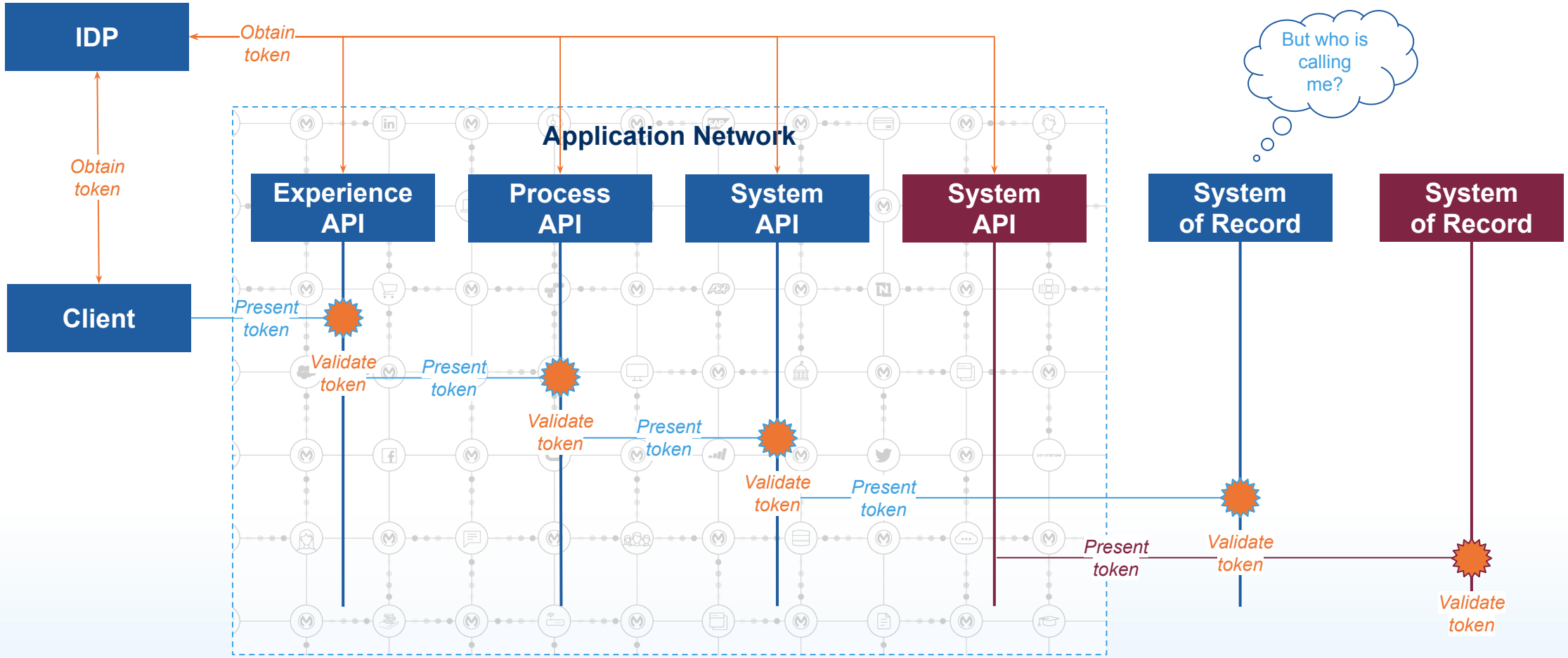
Not so fast!

Propagation of access tokens is a Bad Idea.

- Tokens should have explicit claims that detail how they are used and what they are valid for.
- We could add multiple audiences to a single token, but this could be many many systems in a complex application network.
- Also, future changes to data paths in the application network would require new audiences to be added to tokens.

But equally, using specific tokens between each component in the application network makes it harder to control authorization.

Each product needs a token



A Case of Trust and Identity

Trust

Only accepting requests from known and trusted clients.

ACCESS TOKEN

Identity

Understanding who we are making the request for.

ID TOKEN

OAuth2.0 is extended by OpenID Connect, which has us covered.

A Case of Trust and Identity

Access Token

Normally limited to the entitlements a principal owns.

May only contain a basic user ID (sub).

Keep targeted at a single audience.

ID Token

Contains information about the principal (system or user)

- e.g. Name, e-mail
- Custom claims; user preferences

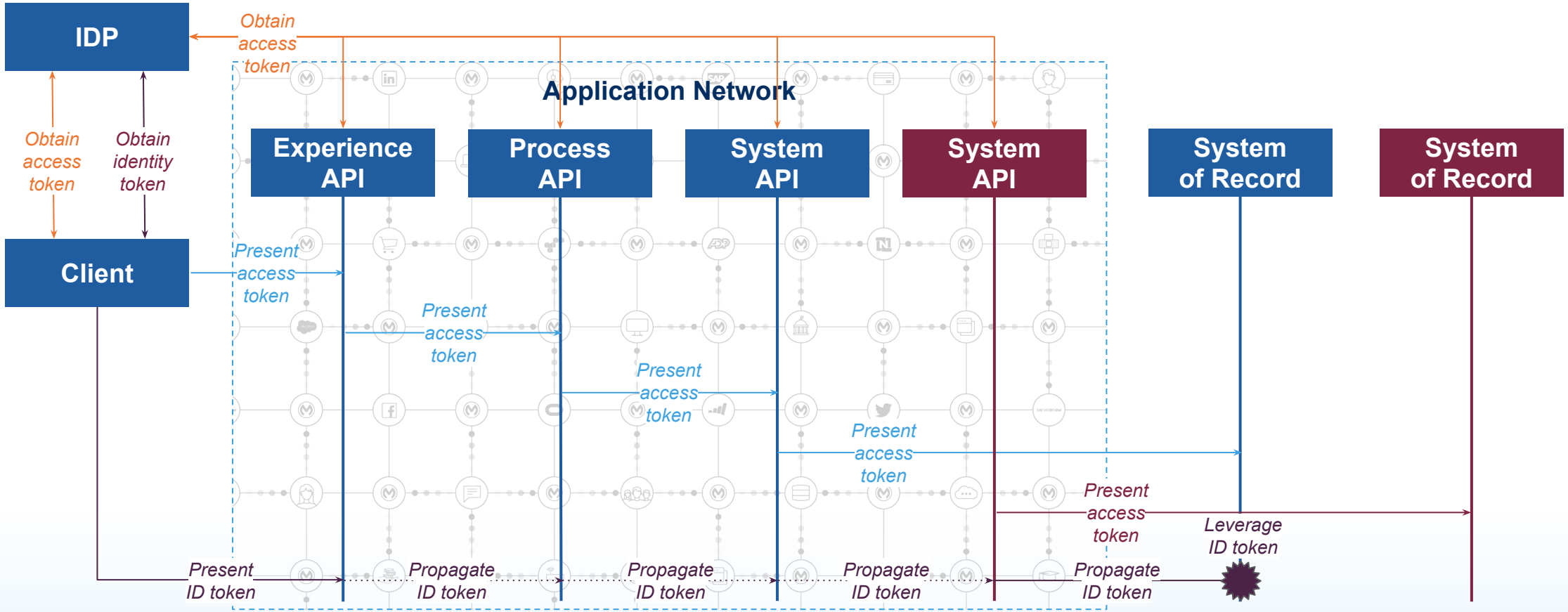
May be passed through the application network.

Does not specify what a user may access.

Both access tokens and ID tokens are JWTs

- Should be validated for authenticity.
- Have specified audience and validity.

Maintaining our identity



Objections & Alternatives

- Delegation or on-behalf-of (OBO) tokens
  - Allows APIs to obtain tokens using end-user identity.
  - ITEF RFC still in draft.
  - Not supported by all IDPs.
- Once issued, JWTs cannot be easily revoked.
  - Keep access token lifetime short.
  - Limit and validate audience claims.

Questions?

Steve Roberts

MuleSoft Solutions Architect

steve.roberts@mulesoft.com

@muleysteve