IETF 121
Dublin
November 2024

Aaron Parecki
Karl McGuinness

Identity Assertion Authorization Grant

https://datatracker.ietf.org/doc/draft-parecki-oauth-identity-assertion-authz-grant/
draft -02

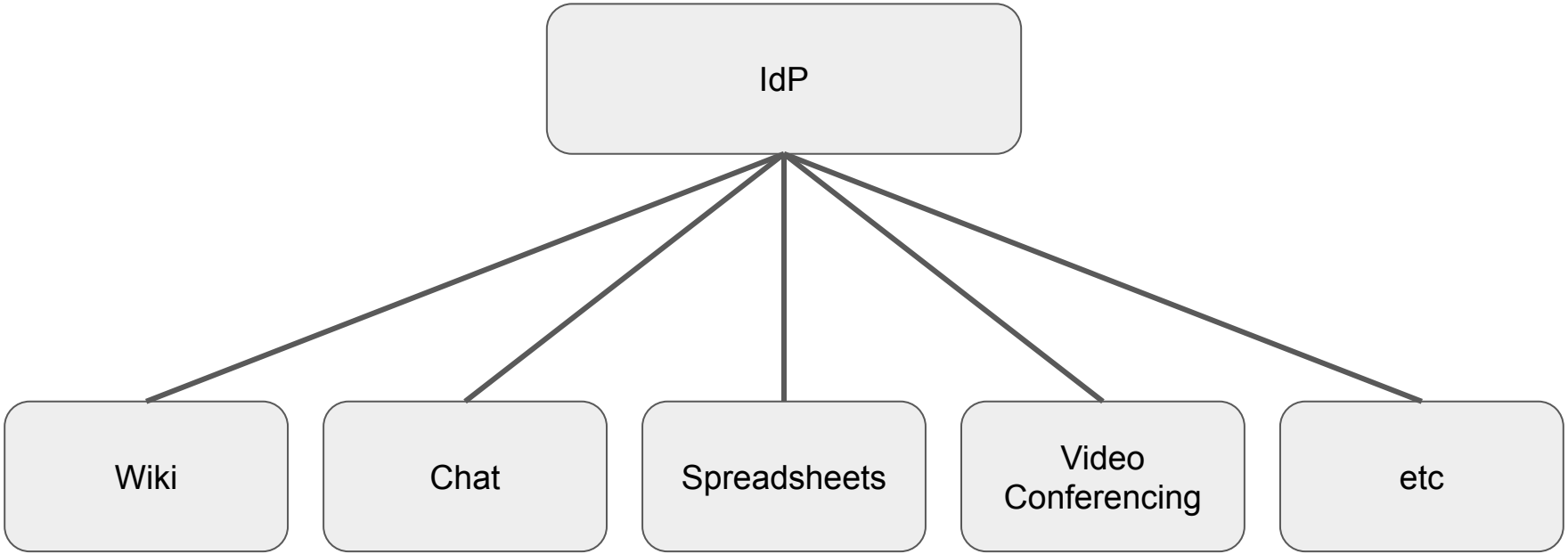
Identity Assertion Authorization Grant

https://datatracker.ietf.org/doc/draft-parecki-oauth-identity-assertion-authz-grant/

A profile of "OAuth Identity and Authorization Chaining Across Domains"

https://datatracker.ietf.org/doc/draft-ietf-oauth-identity-chaining/

Single Sign-on in an Enterprise



My Team Page

KM Created by Karl McGuinness
Just a moment ago • 1 min read • Analytics

This is a wiki page for my team that I share important links to content I want folks to discover

Zoom Recording

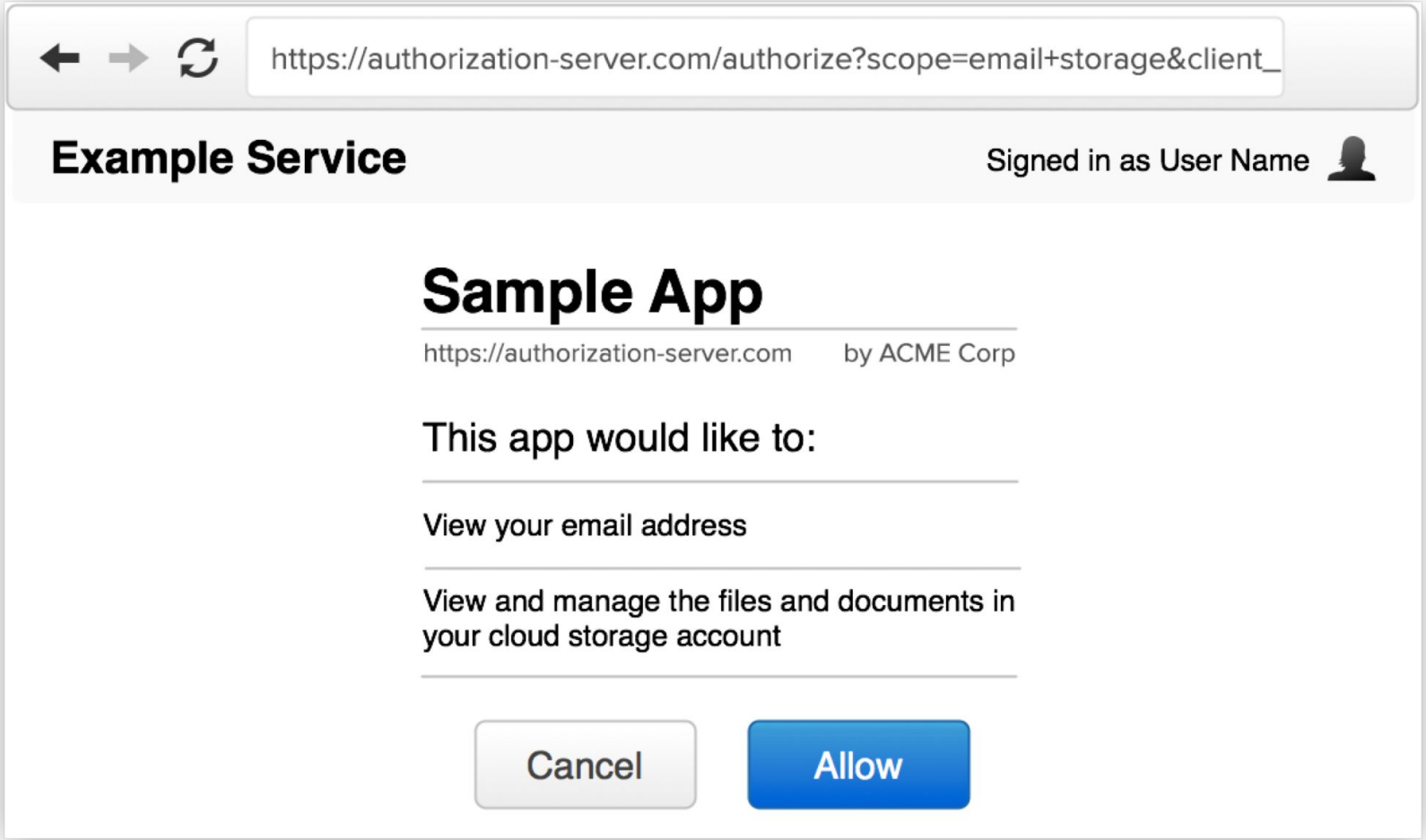
Video Conferencing, Web Conferencing, Webinars, Screen Sharing

Slack Thread

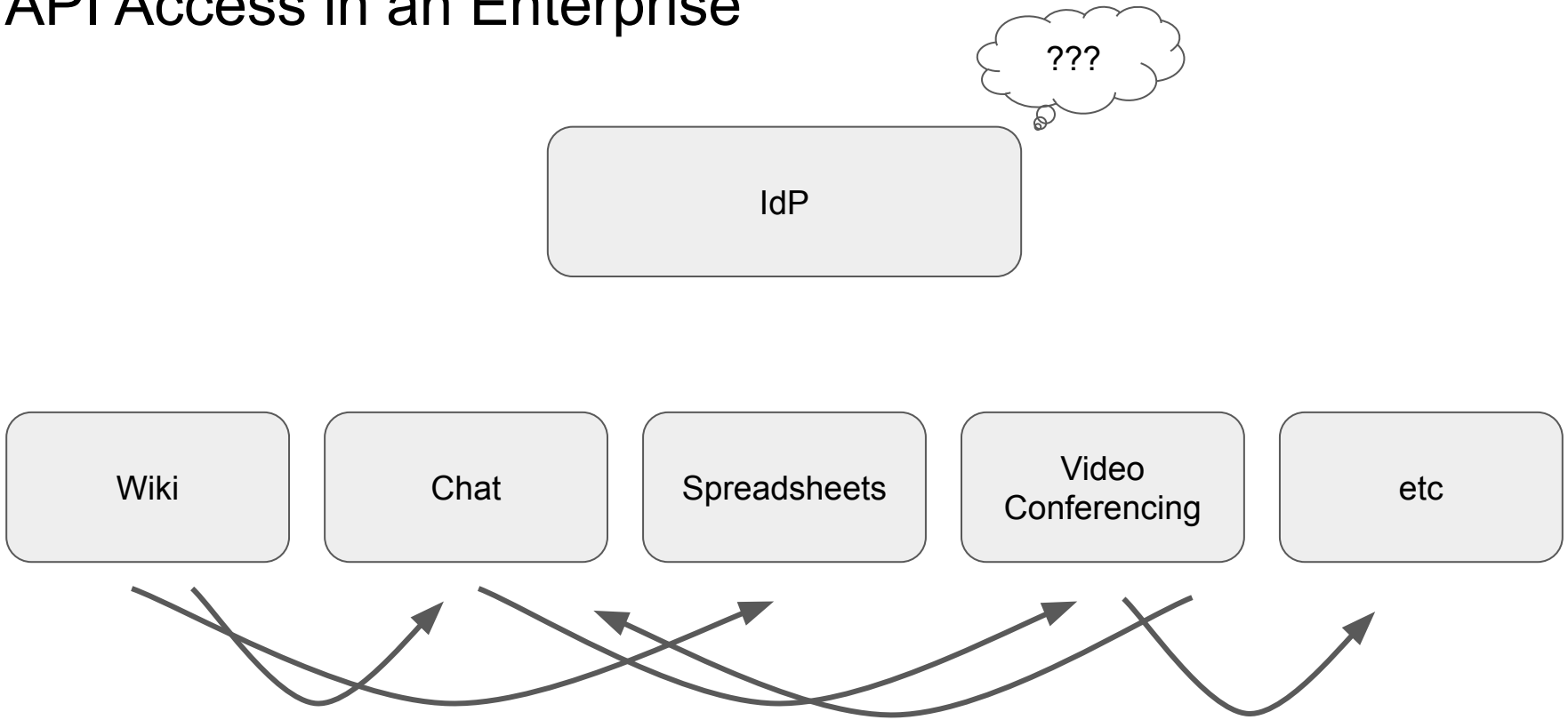
https://okta.slack.com/archives/C04NDRSF5DX/p1684935454741749 - Connect your Slack account

Figma

https://www.figma.com/file/BZexDbVItzH6BJLFkgGk0Z/Product-principles-sesh?type=whiteboard&node-id=0-1&t=4jP7GLJAYzv6mxCq-0 - Connect your Figma account


https://authorization-server.com/authorize?scope=email+storage&client_
Example Service
Signed in as User Name
Sample App
https://authorization-server.com
by ACME Corp
This app would like to:
View your email address
View and manage the files and documents in your cloud storage account
Cancel
Allow

API Access in an Enterprise



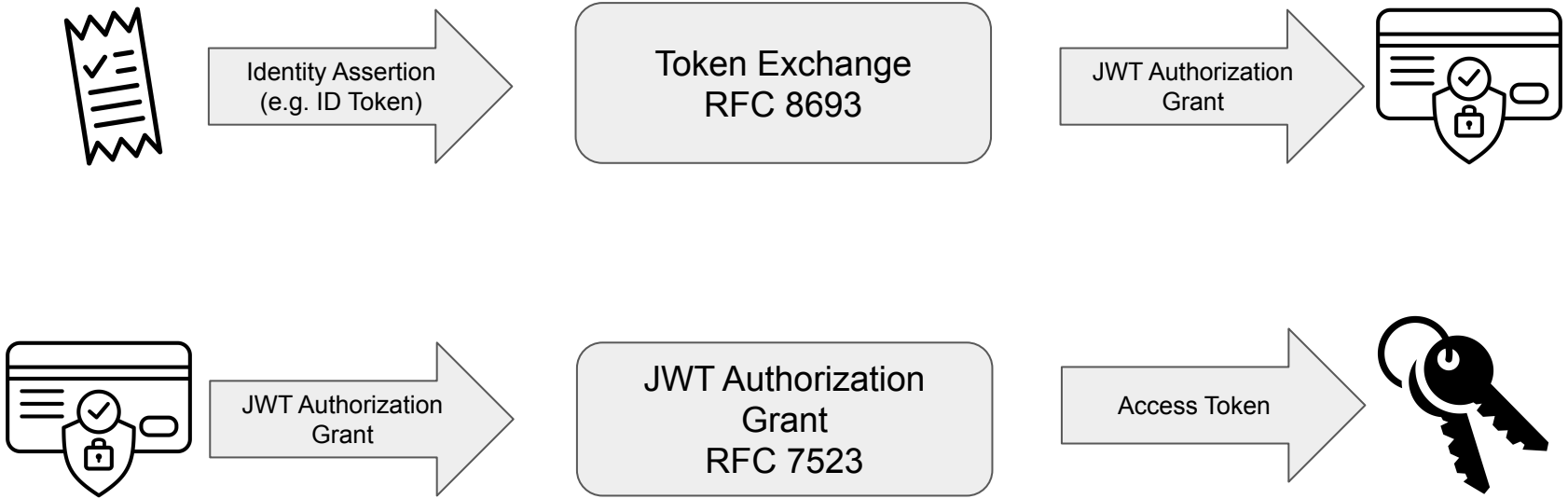
Identity Assertion Authorization Grant

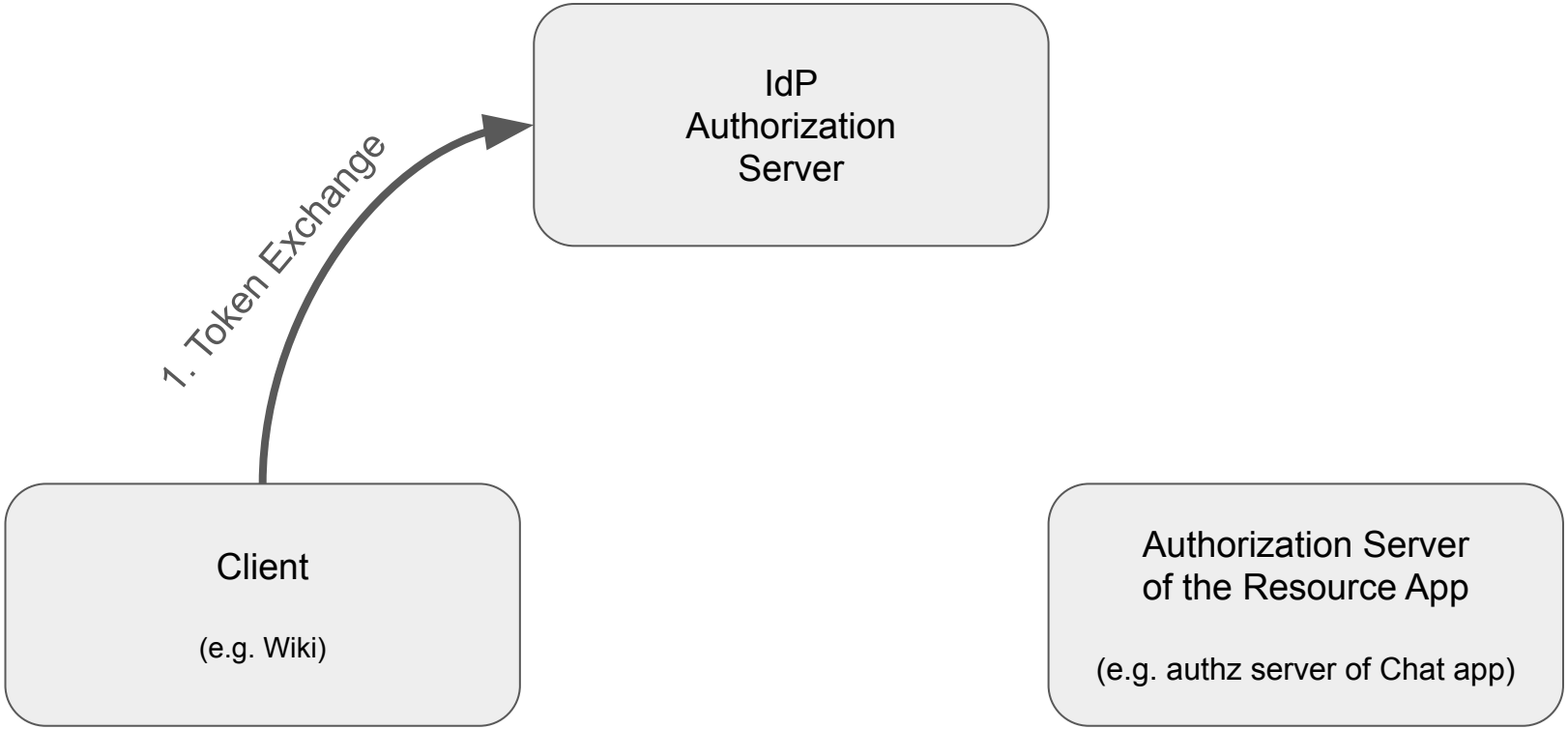
https://datatracker.ietf.org/doc/draft-parecki-oauth-identity-assertion-authz-grant/

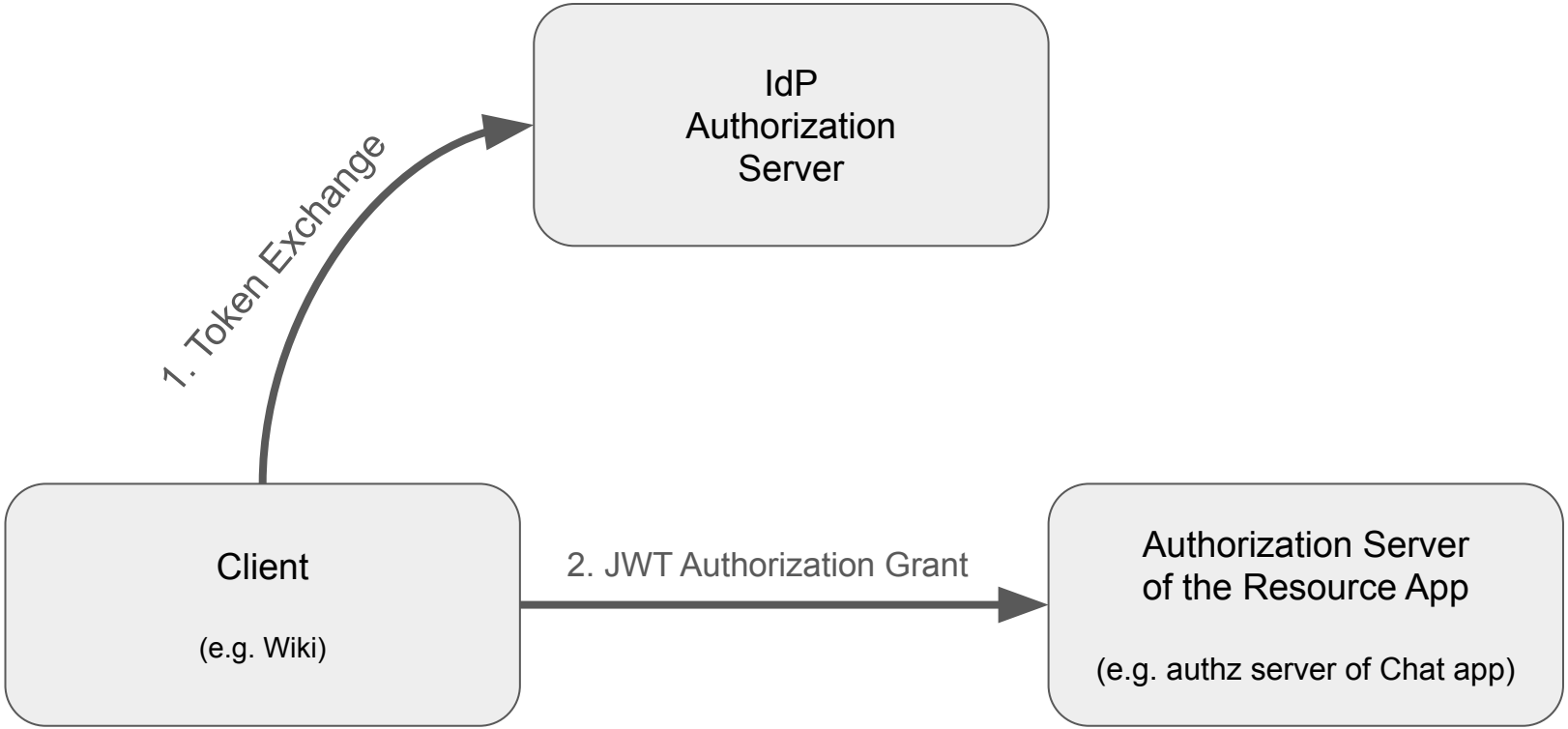
Goals:

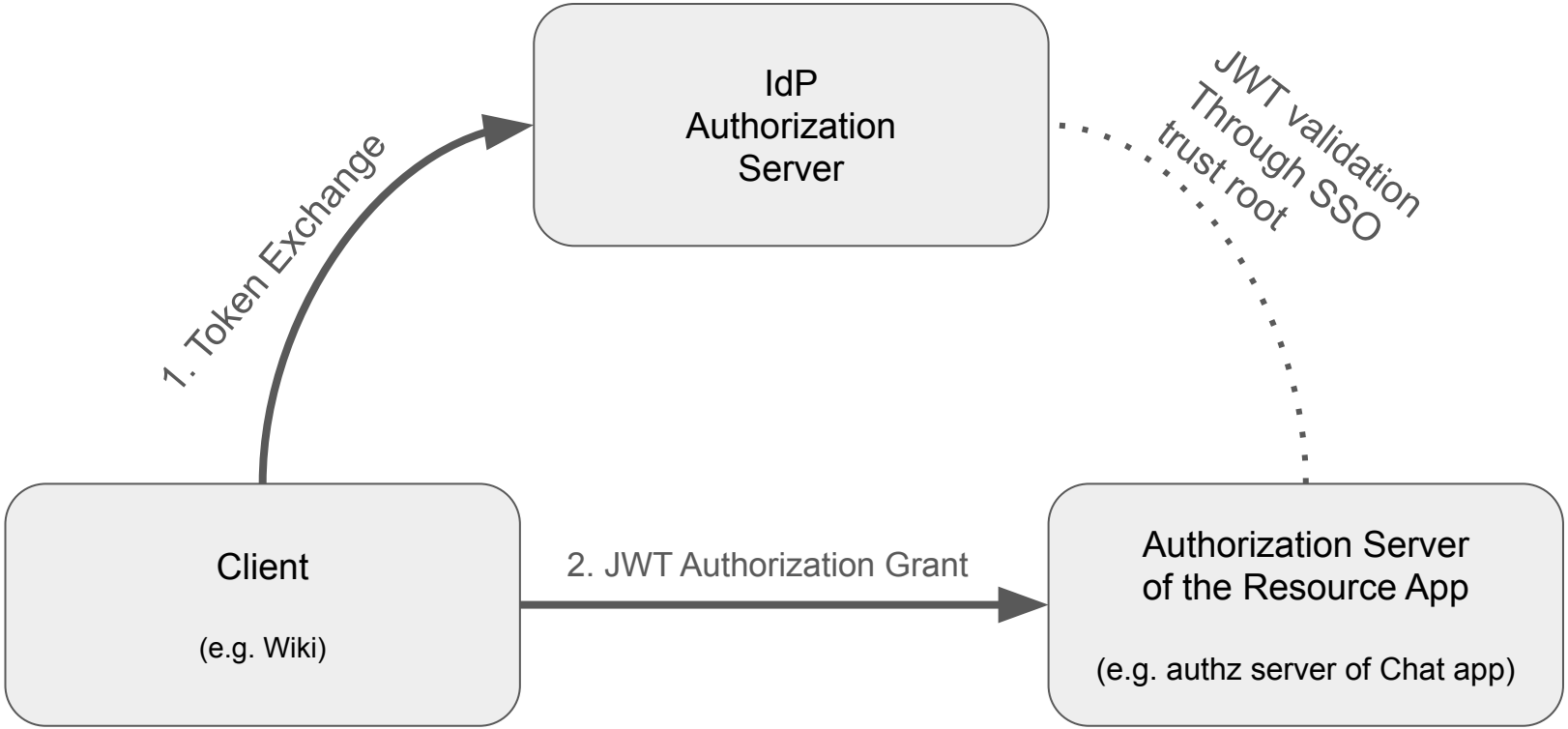
- Extend single sign-on to API access
- Reduce user friction
- Enable enterprise control of data sharing between apps

Building Blocks




IdP
Authorization
Server
1. Token Exchange
Client
(e.g. Wiki)
Authorization Server
of the Resource App
(e.g. authz server of Chat app)


IdP
Authorization
Server
1. Token Exchange
Client
(e.g. Wiki)
2. JWT Authorization Grant
Authorization Server
of the Resource App
(e.g. authz server of Chat app)


IdP
Authorization
Server
JWT validation
Through SSO
trust root
1. Token Exchange
Client
(e.g. Wiki)
2. JWT Authorization Grant
Authorization Server
of the Resource App
(e.g. authz server of Chat app)

Why Profile “Identity and Authorization Chaining” draft?

Identity and Authorization Chaining Across Domains	**Identity Assertion Authorization Grant**
How the initial token is obtained is out of scope	Initial token is obtained via an SSO flow
Input to token exchange is generic/flexible, can be any type of token	Only identity tokens are accepted (ID Token, SAML assertion)
Cross-domain trust relationship is out of scope	Cross-domain trust relationship is an SSO relationship from both domains to an identity provider
No additional semantics defined in the JWT Authorization Grant beyond what's in RFC7523	Requires certain claims in the JWT and defines their semantics

We need *interop* between all parties participating in this use case

Token Exchange Request (RFC 8693)

Client to Identity Provider

Token endpoint of target authorization server

Defined in this spec

ID Token

```
POST /oauth2/token HTTP/1.1
Host: acme.idp.example
Content-Type: application/x-www-form-urlencoded

grant_type=urn:ietf:params:oauth:grant-type:token-exchange
&requested_token_type=urn:ietf:params:oauth:token-type:id-jag
&resource=https://acme.chat.example/oauth2/token
&scope=chat.read+chat.history
&subject_token=eyJraWQiOiJzMTZ0cVNtODhwREo4VGZCXzdrSEtQ...
&subject_token_type=urn:ietf:params:oauth:token-type:id_token
&client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer
&client_assertion=eyJhbGciOiJSUzI1NiIsImtpZCI6IjIyIn0...
```

Token Exchange Response

From Identity Provider to Client

JWT Authorization Grant with defined claims and semantics

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "issued_token_type": "urn:ietf:params:oauth:token-type:id-jag",
  "access_token": "eyJhbGciOiJIUzI1NiIsI...",
  "token_type": "N_A",
  "scope": "chat.read chat.history",
  "expires_in": 300
}
```

JWT Authorization Grant

```
{
  "typ": "oauth-id-jag+jwt"
}
.
{
  "jti": "9e43f81b64a33f20116179",
  "iss": "https://acme.idp.example",
  "sub": "U019488227",
  "aud": "https://acme.chat.example/oauth2/token",
  "client_id": "f53f191f9311af35",
  "exp": 1311281970,
  "iat": 1311280970,
  "scope": "chat.read chat.history"
}
.
signature
```

JWT Assertion Authorization Grant (RFC 7523)

Client to Resource App's Authorization Server

```
POST /oauth2/token HTTP/1.1
Host: acme.chat.example
Authorization: Basic yZS1yYW5kb20tc2VjcmV0v3JOkF0XG5Qx2

grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer
assertion=eyJhbGciOiJIUzI1NiIsI...
```

JWT Authorization Grant

Token Response

From Resource App's Authorization Server to Client

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "token_type": "Bearer",
  "access_token": "2YotnFZFEjr1zCsicMWpAA",
  "expires_in": 86400,
  "refresh_token": "tGzv3JOkF0XG5Qx2TlKWIA",
}
```

No changes to existing access token response or access token format

Implementation Status

- Okta
  - Currently implemented in development branch of Okta IdP
  - In-progress work to ship in production
- Atlassian
  - Implemented a prototype against Okta
  - In-progress work to ship in production

We are in discussions with additional companies who are interested in building this as well.