Microsoft

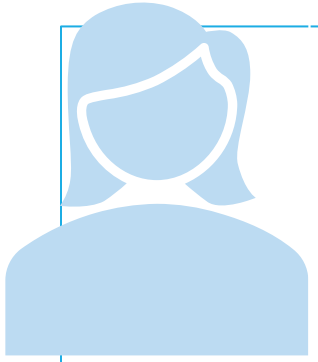
Microsoft Identity platform
Developer training

Microsoft

Microsoft identity platform
Developer Training

Adding Authentication in Your Apps

Identity



What is Identity?

- An Actor that can be Granted Access based on their Properties and Role in a System

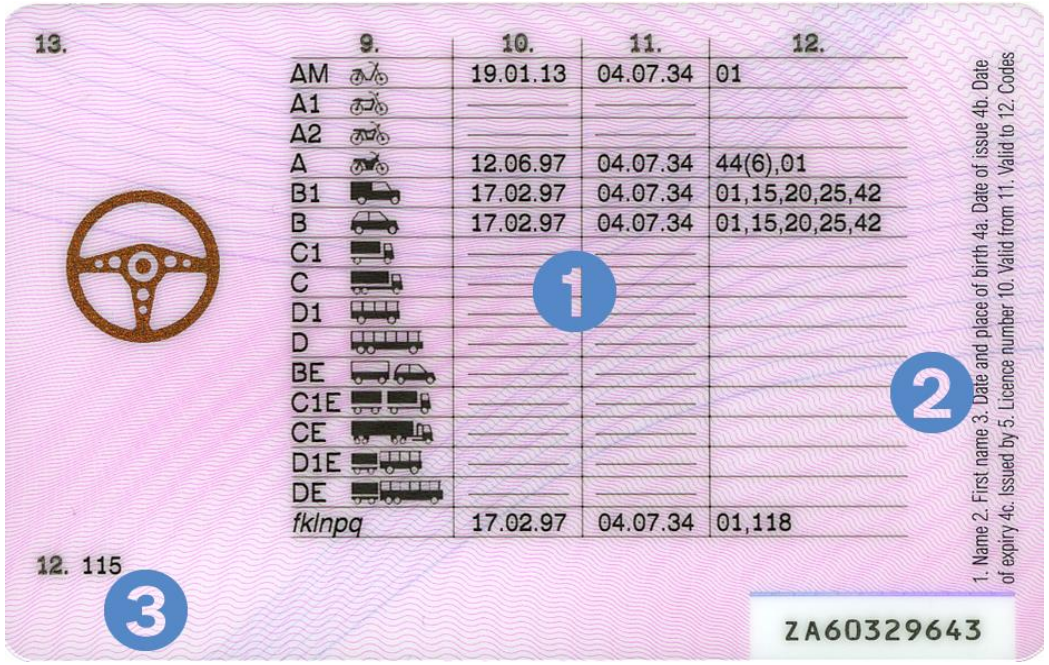
How is Identity handled?

- Identity is confirmed by an Authority that gives out a Trusted Token that can be used by the Actor to identify itself

Developer responsibility:

Deal with Tokens

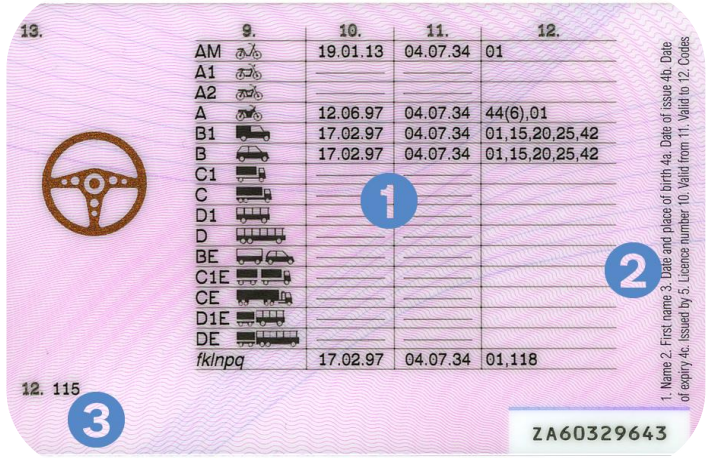
Token Example: The Driving License



License is given out by **Authority** (Government)
License is **Trusted** because of security means (Hard to reproduce/fake)
License contain **Claims** (ID, Name, Photo, Address)
License contains **Permissions** (Access to sit at driver seat of what type of vehicle)

Authentication versus Authorization

Authentication is done to verify the Actor (a person or system) is who it says it is.

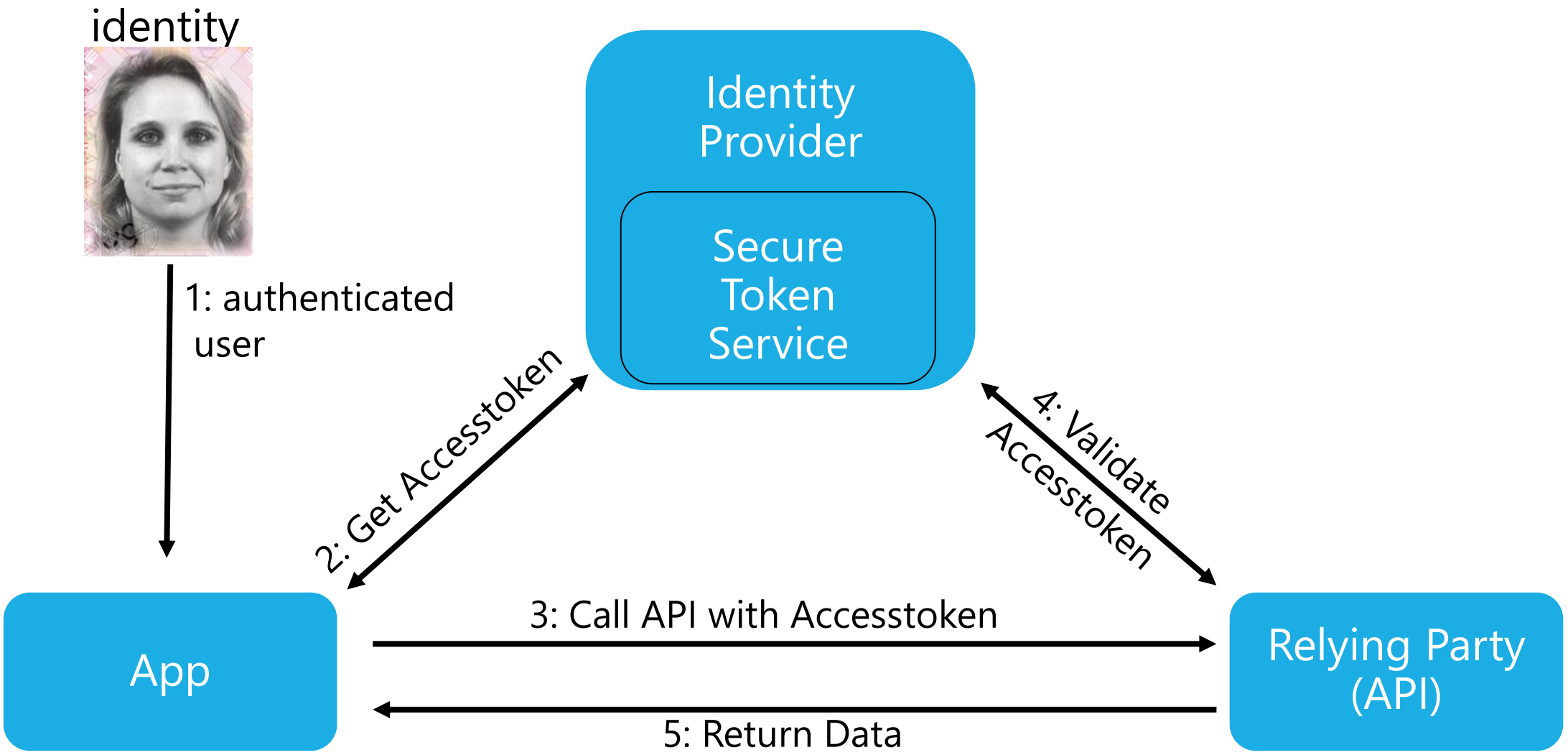


Authorization is to give an authenticated person access

Identity in an application

1. Identity	You or a daemon
2. Relying Party	The service you want to use
3. Identity Provider	The authority that can proof you are you
4. Secure Token Service	The service that hands out a token
5. IDToken	The thing that proofs you are you
6. AccessToken	The thing that proofs what permissions you have

Trust chain to retrieve tokens



Authentication in a Web Application

Browser

Web App

Identity Provider

User navigates to application

App checks if user is authenticated

App redirects browser to IDP

Browser navigates to IDP

IDP returns login experience

User enters credentials

IDP checks credentials

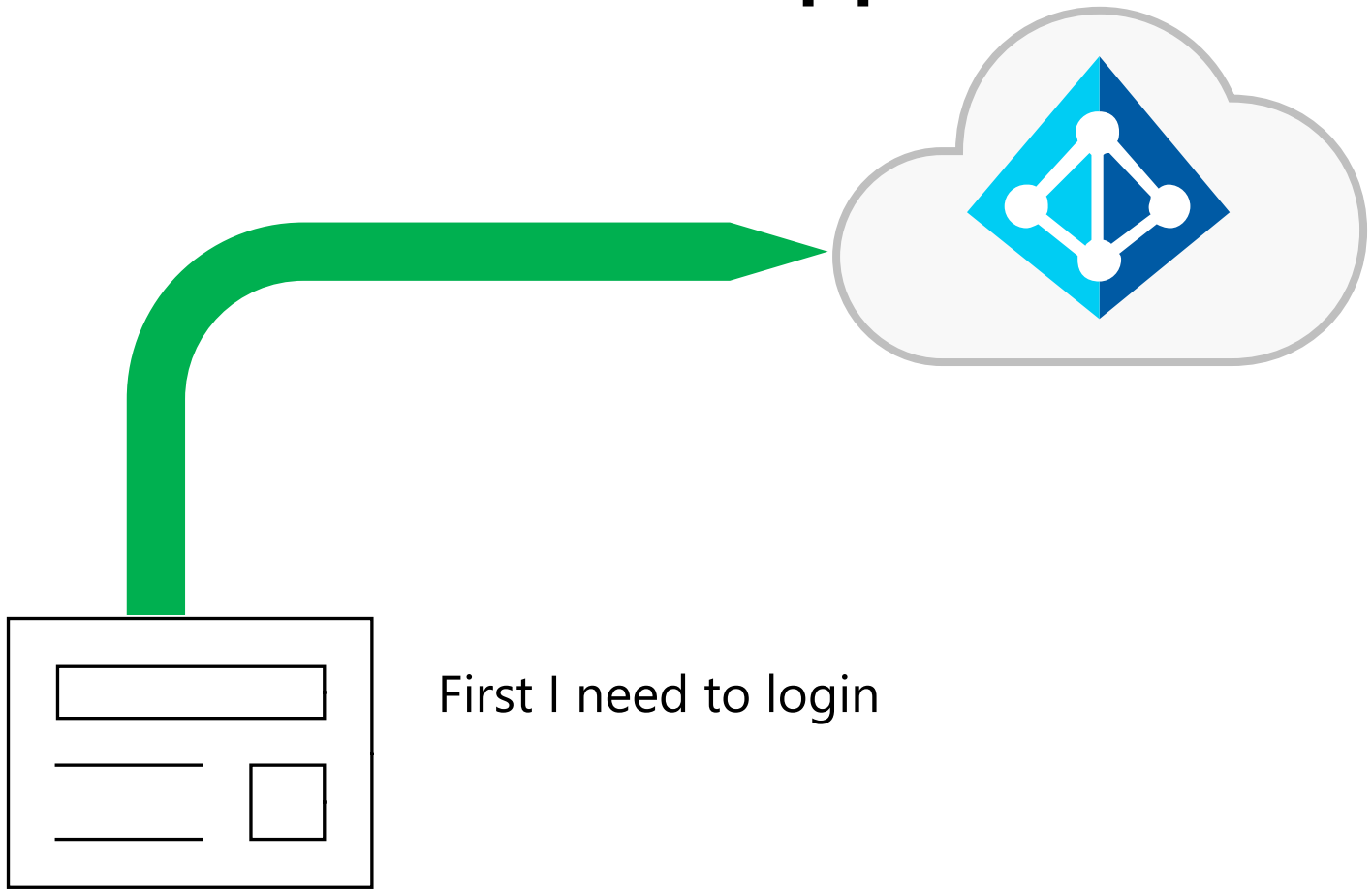
IDP redirects browser to redirectURI

Browser redirects to redirectURI

App verifies redirect URI
App return app screen with session cookie

Browser shows page

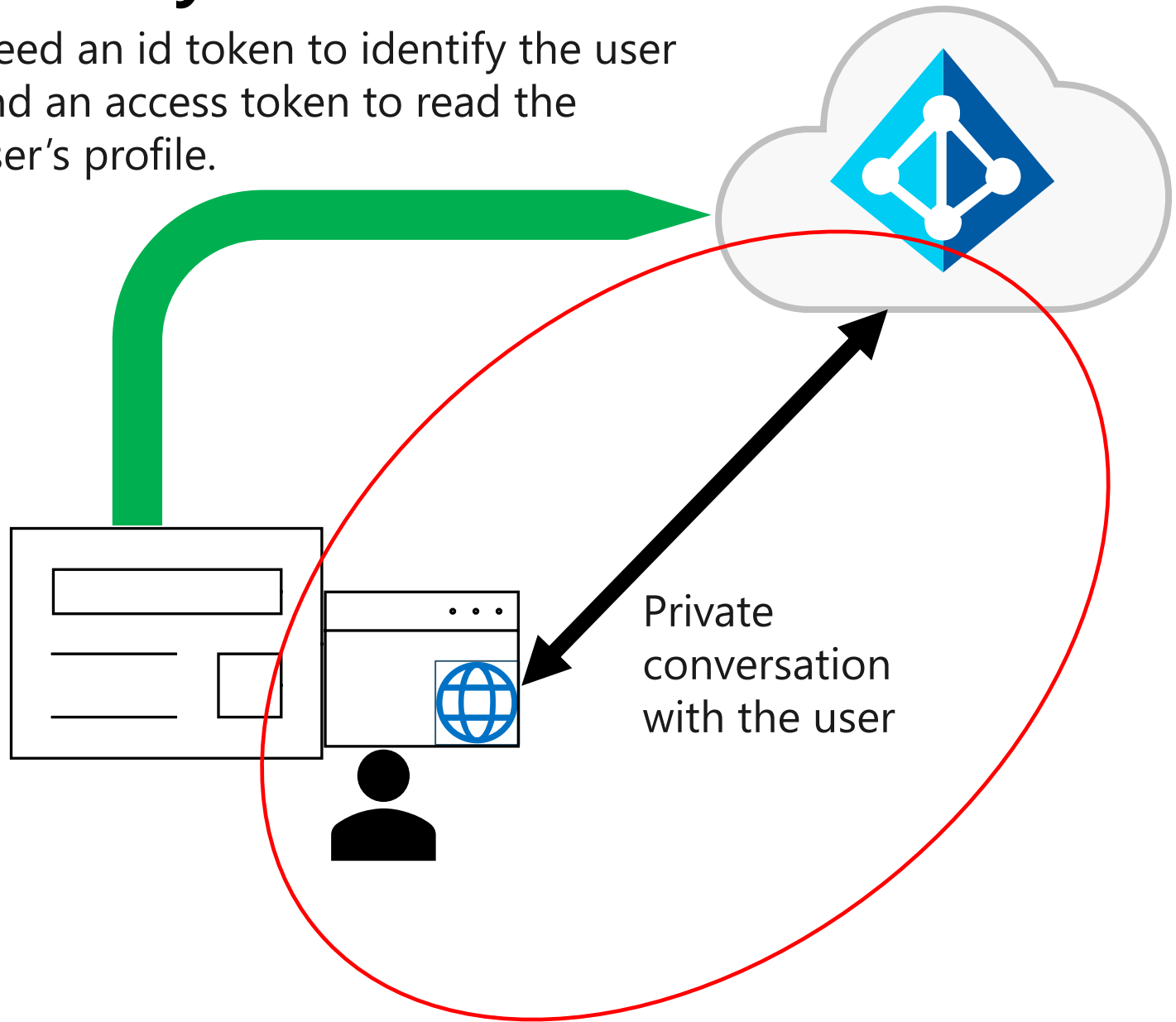
Let’s start with an app



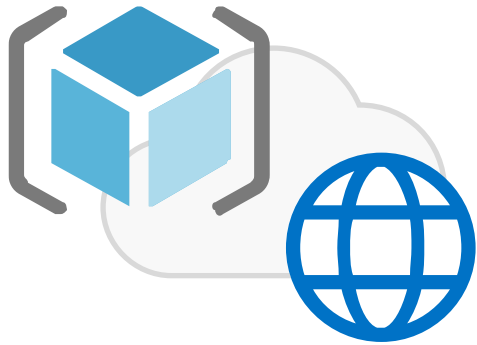
First I need to login

Identity Provider returns Id token

Need an id token to identify the user and an access token to read the user's profile.



Authorization Required

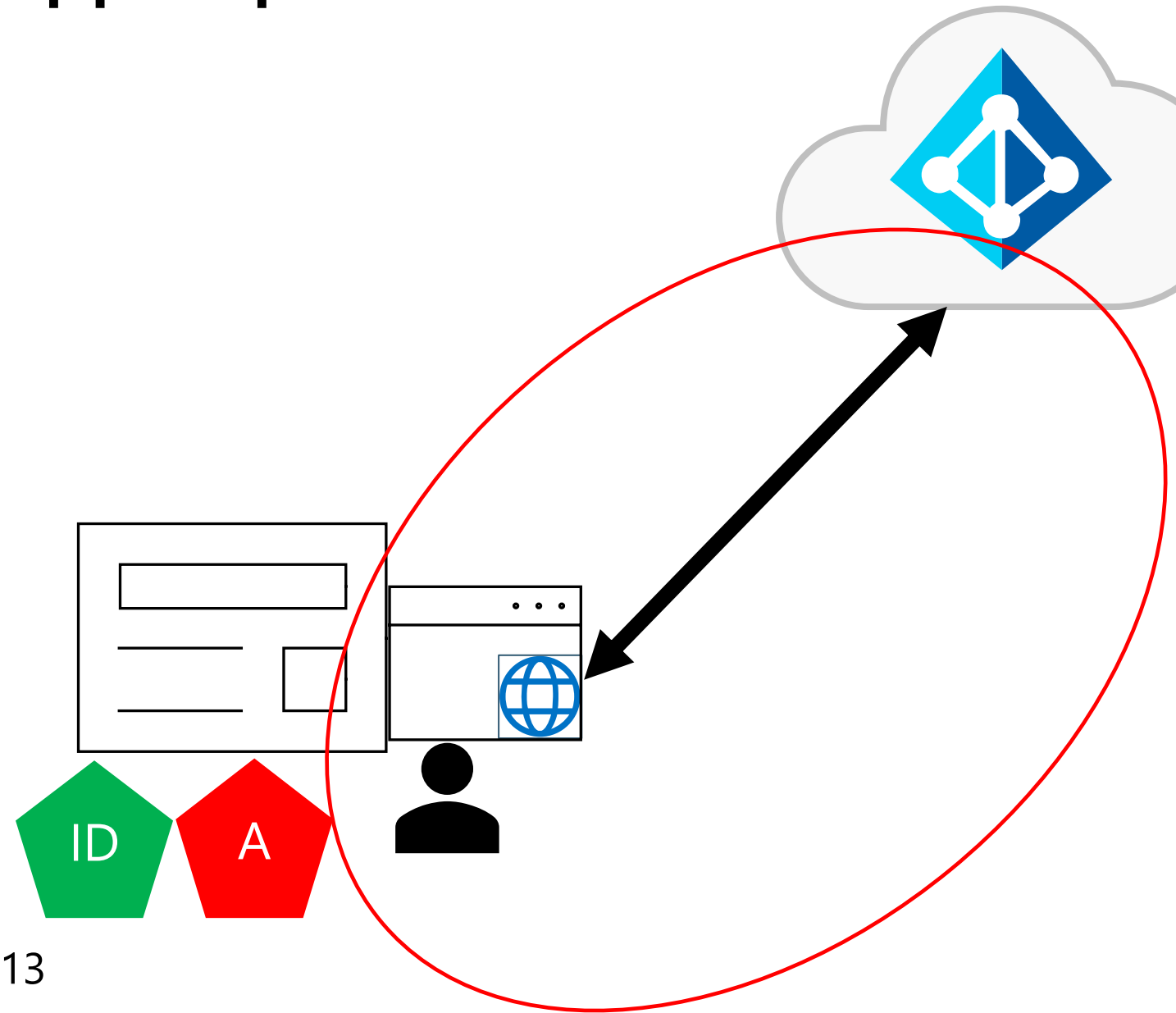


Relying Party (API)

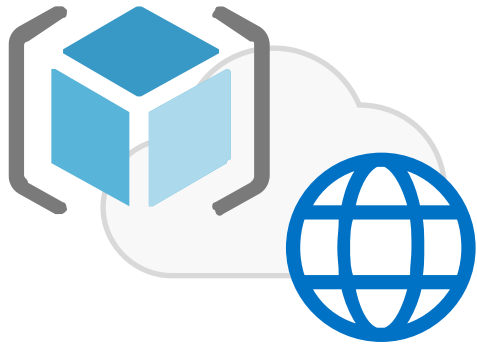
App receives an id_token

{ "typ": "JWT", "alg": "RS256", "kid": "1LTMzakihiRla_8z2BEJVXeWMqo" }.
{ "ver": "2.0",
"iss": "https://login.microsoftonline.com/3338040d-6c67-4c5b-b112-36a304b66dad/v2.0",
"aud": "6cb04018-a3f5-46a7-b995-940c78f5aef3",
"exp": 1536361411,"iat": 1536274711, "nbf": 1536274711,
"sub": "AAAAAAAAAAAAAAAAAAAAAIkzqFVrSaSaFHy782bbtaQ",
"name": "Abe Lincoln",
"preferred_username": "AbeLi@microsoft.com",
"oid": "00000000-0000-0000-66f3-3332eca7ea81",
"tid": "3338040d-6c67-4c5b-b112-36a304b66dad",
}
.[Signature]

App request and receives an access token



Authorization Required

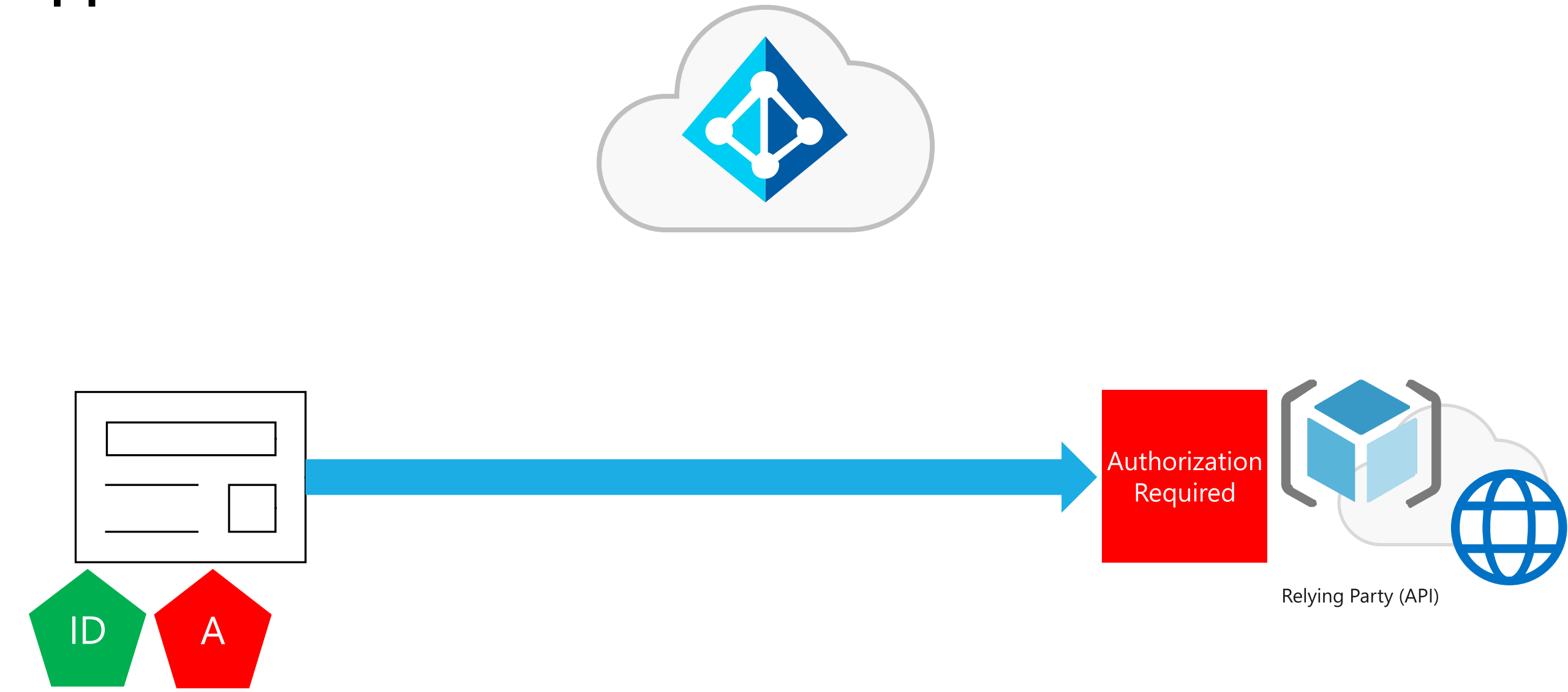


Relying Party (API)

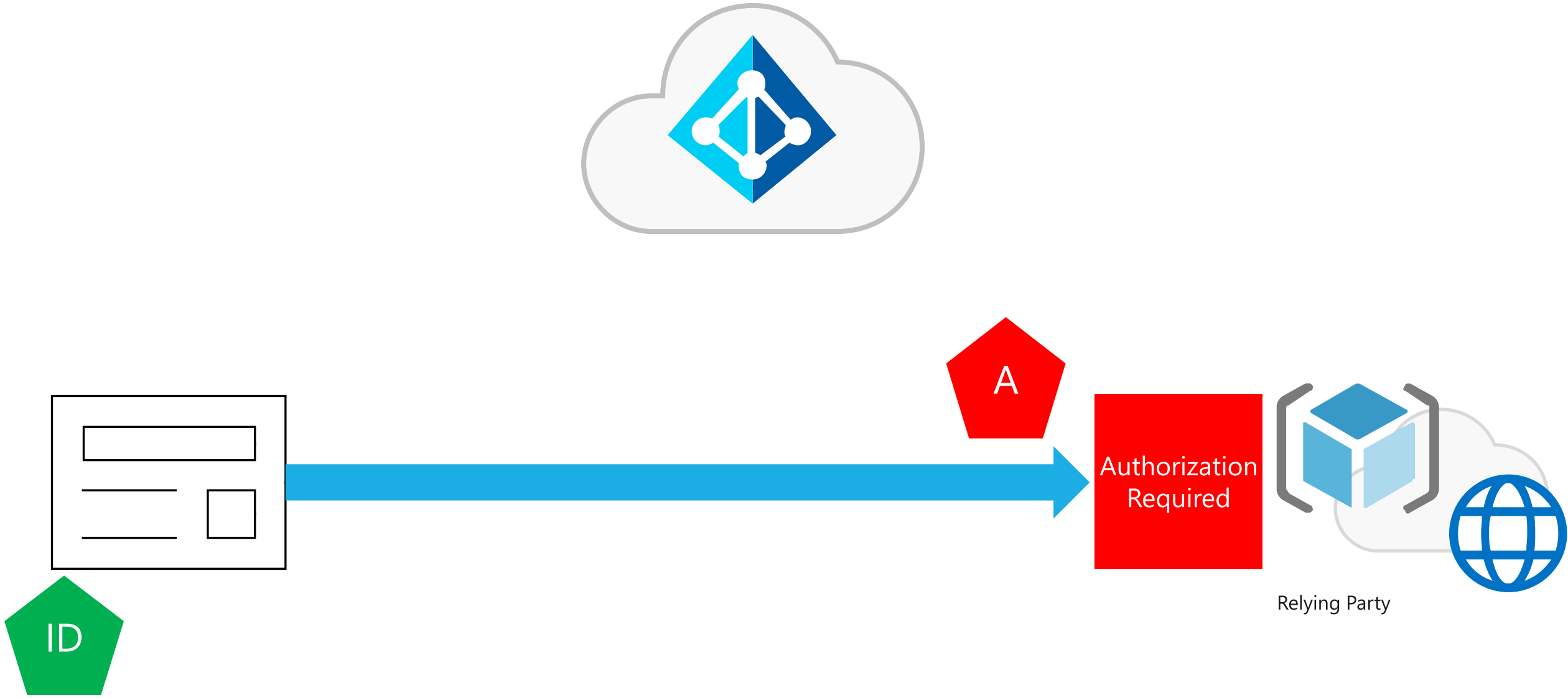
App receives an access token

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Imk2bEdrM0ZaenhSY1ViMkMzbkVRN3
N5SEpsWSJ9.eyJhdWQiOiI2ZTc0MTcyYi1iZTU2LTQ4NDMtOWZmNC1lNjZhMzliYjEyZTMi
LCJpc3MiOiJodHRwczovL2xvZ2luLm1pY3Jvc29mdG9ubGluZS5jb20vNzJmOTg4YmYtODZ
mMS00MWFmLTkxYWItMmQ3Y2QwMTFkYjQ3L3YyLjAiLCJpYXQiOjE1MzcyMzEwNDgsIm
5iZiI6MTUzNzIzMTA0OCwiZXhwIjoxNTM3MjM0OTQ4LCJhaW8iOiJBWFFBaS84SUFBQUF
0QWFaTG8zQ2hNaWY2S09udHRSQjdlQnE0L0RjY1F6amNKR3hQWXkvQzNqRGFOR3hYZ
DZ3TklJVkdSZ2hOUm53SjFsT2NBbk5aY2p2a295ckZ4Q3R0djMzMTQwUmlvT0ZKNGJDQ
0dWdW9DYWcxdU9UVDIyMjIyZ0h3TFBZUS91Zjc5UVgrMEtJaWpkcm1wNjlSY3R6bVE9PS
IsImF6cCI6IjZlNzQxNzJiLWJlNTYtNDg0My05ZmY0LWU2NmEzOWJiMTJlMyIsImF6cGFjciI
6IjAiLCJuYW1lIjoiQWJlIExpbmNvbG4iLCJvaWQiOiI2OTAyMjJiZS1mZjFhLTRkNTYtYWJkMS
03ZTRmN2QzOGU0NzQiLCJwcmVmZXJyZWRfdXNlcm5hbWUiOiJhYmVsaUBtaWNyb3Nv
ZnQuY29tIiwicmgiOiJJIiwic2NwIjoiYWNjZXNzX2FzX3VzZXIiLCJzdWIiOiJIS1pwZmFleVdhZ
GVPb3VZbGI0anJJLUtmZlRtMjIyWDVyclYzeERxZktRIiwidGlkIjoiNzJmOTg4YmYtODZmMS
00MWFmLTkxYWItMmQ3Y2QwMTFkYjQ3IiwidXRpIjoiZnFpQnFYTFBqMGVRYTgyUy1JWU
ZBQSIsInZlciI6IjIuMCJ9.pj4N-w_3Us9DrBLfpCt

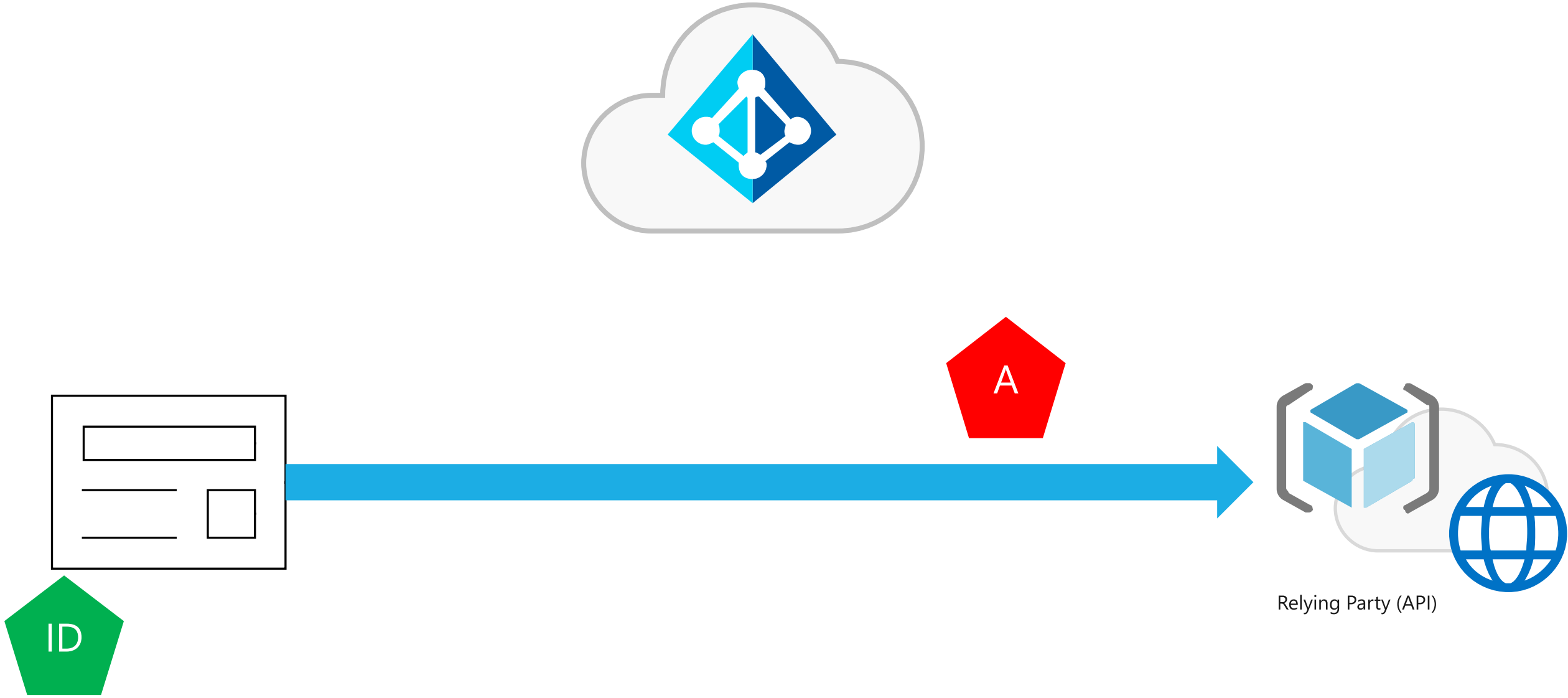
App has an authenticated user and a token to call API



Call the API



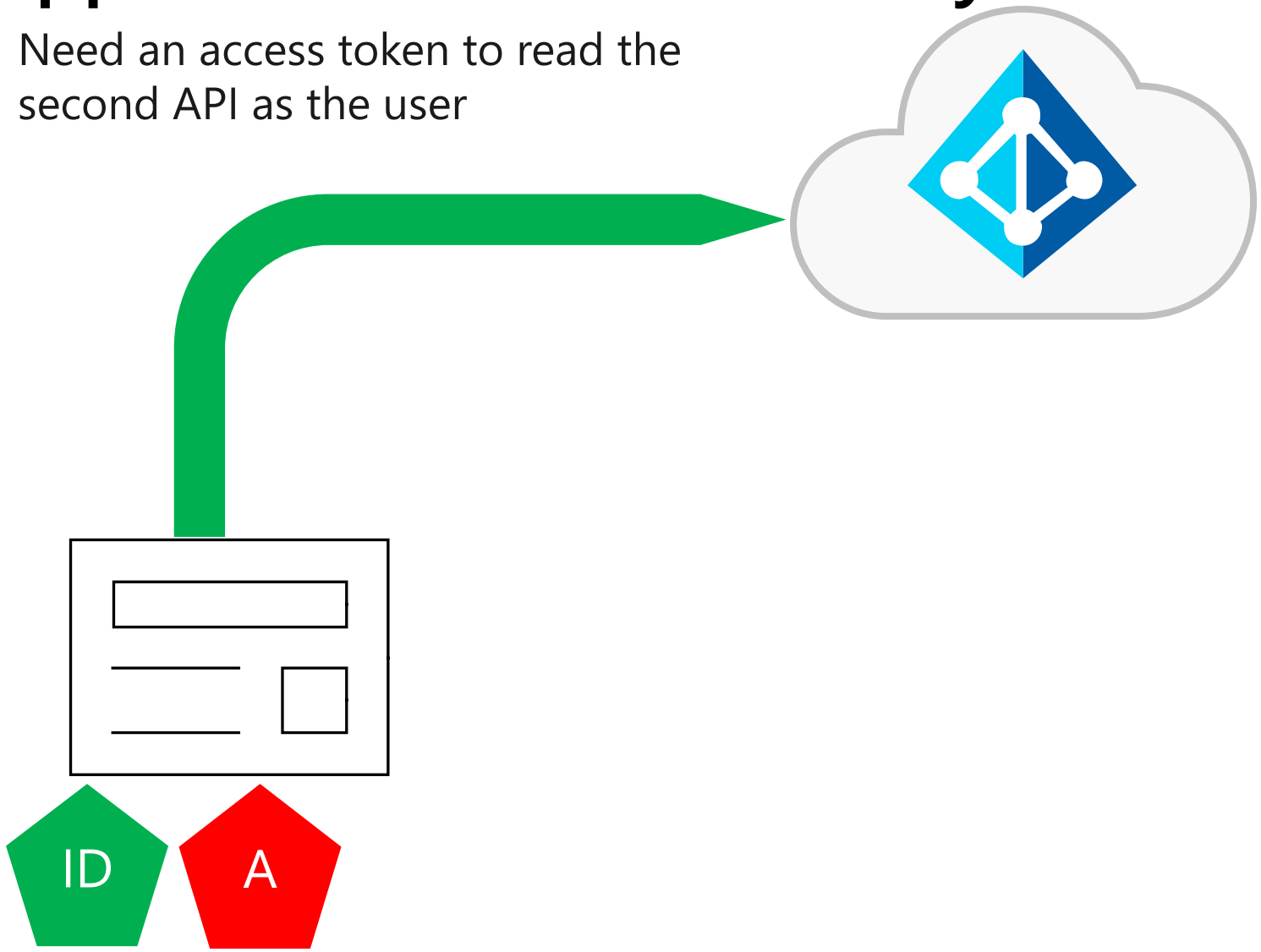
Call the API



But what if I need to call another API?

Apps asks Microsoft identity for another access token

Need an access token to read the second API as the user



Authorization Required

Permissions:
{Calendar.Update}

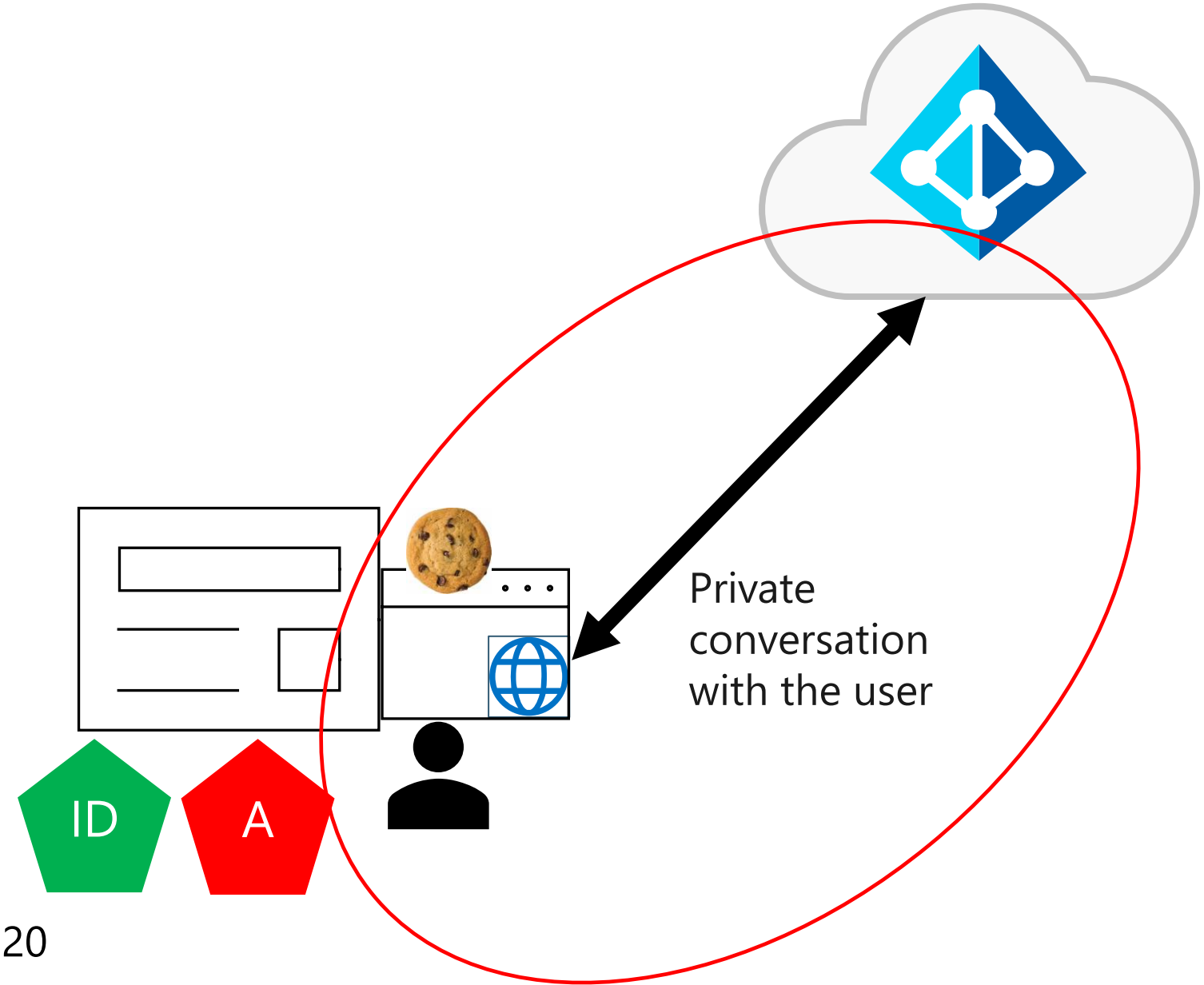
Relying Party 2 (API 2)

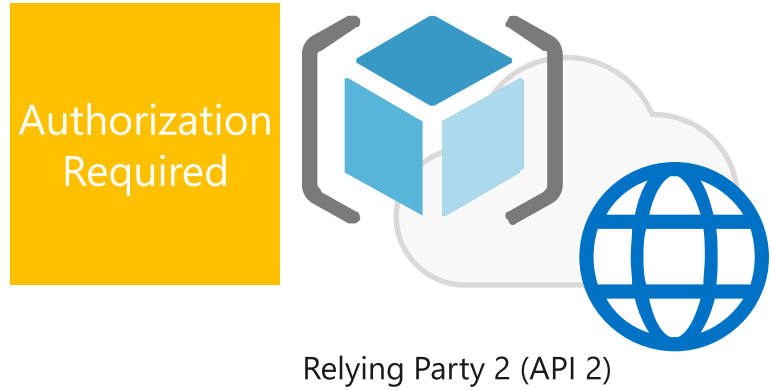
Authorization Required

Permissions:
{Mail.Read, Mail.Write}

Relying Party 1 (API 1)

The core of SSO

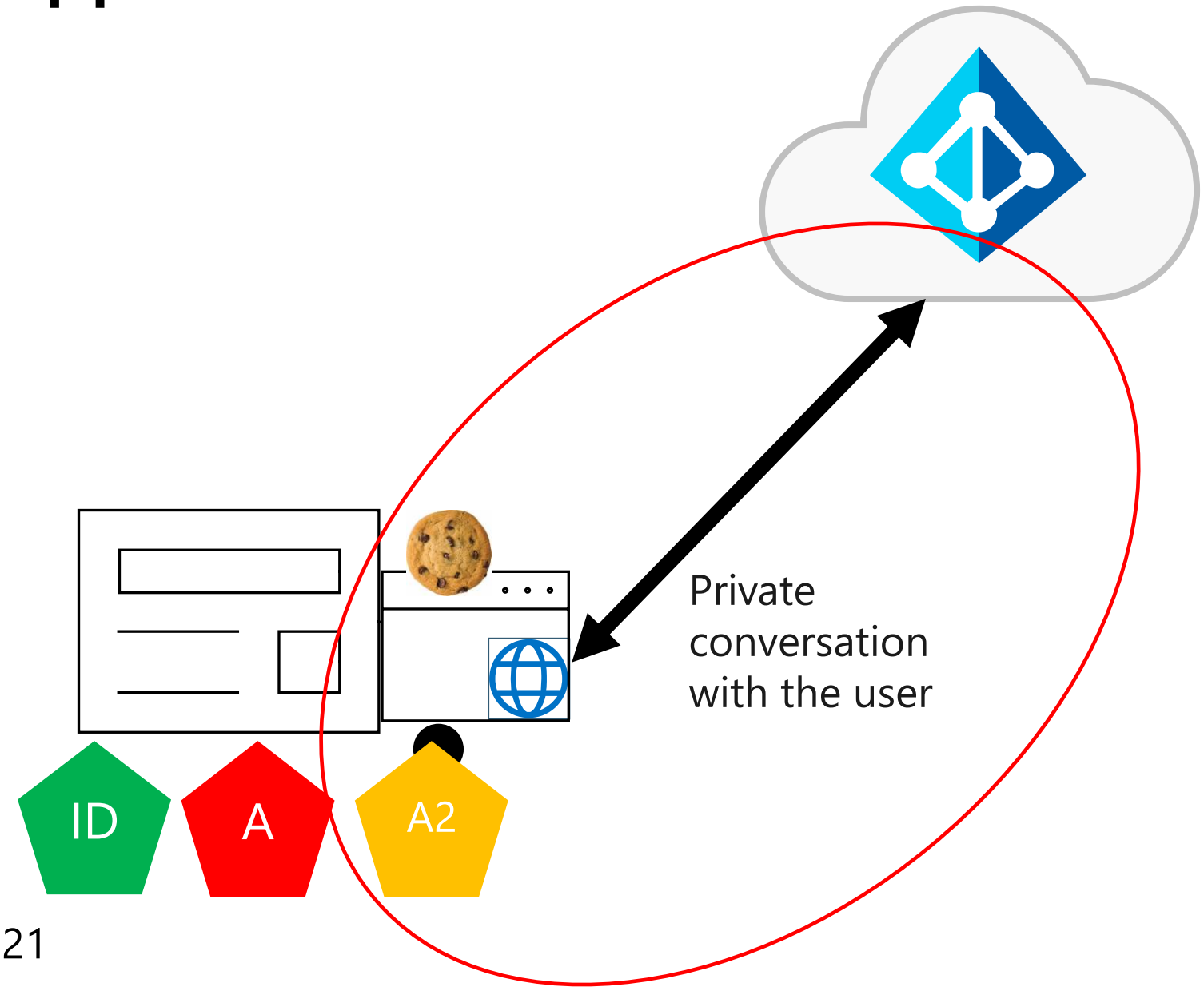




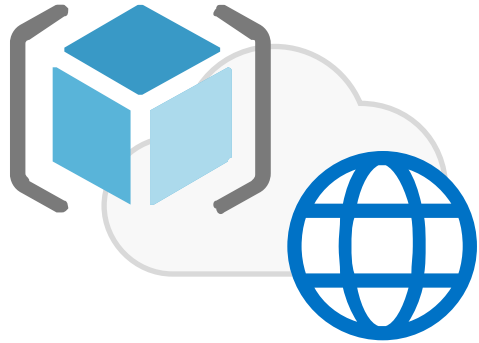
Relying Party 2 (API 2)

Relying Party (API 1)

App receives another access token

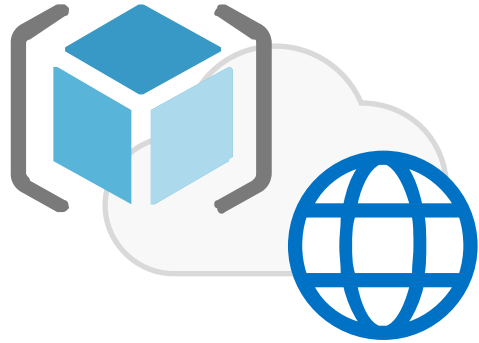


Authorization Required



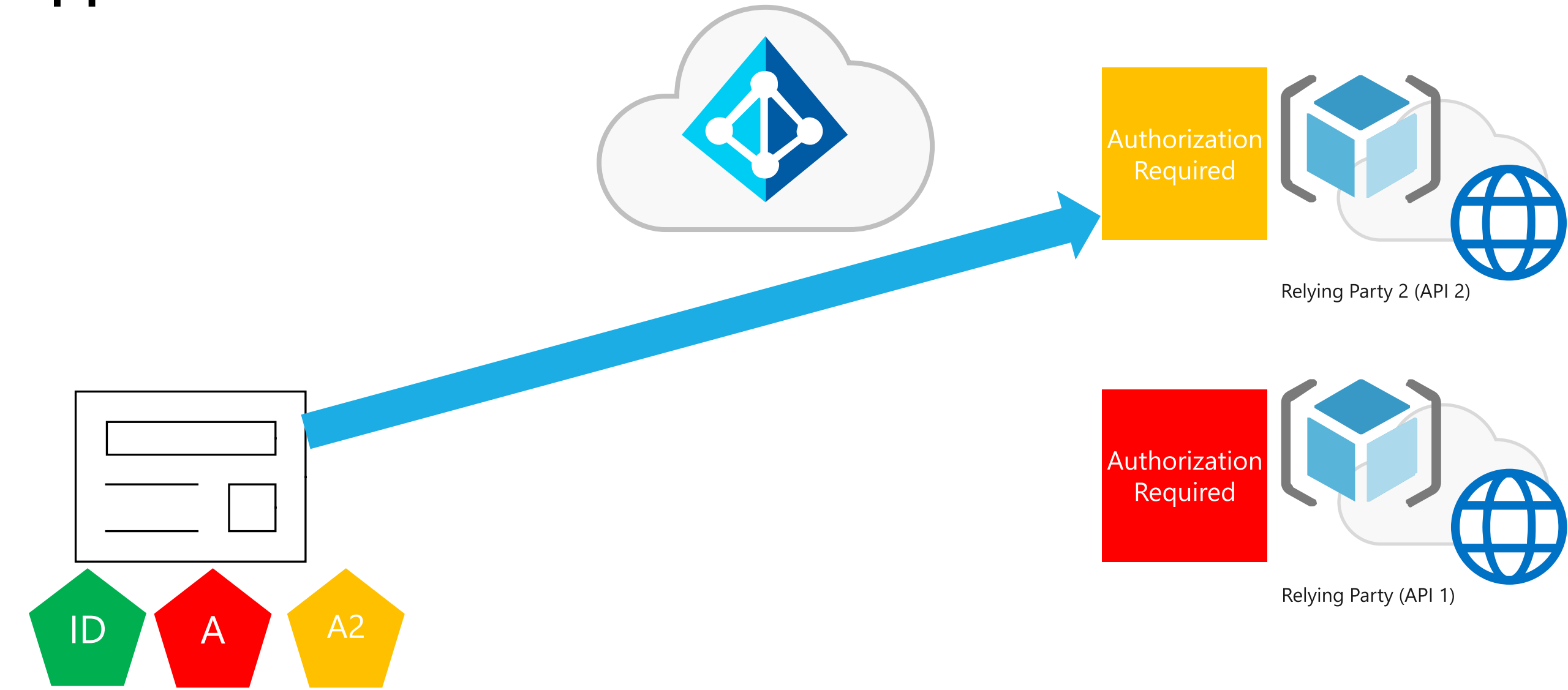
Relying Party 2 (API2)

Authorization Required

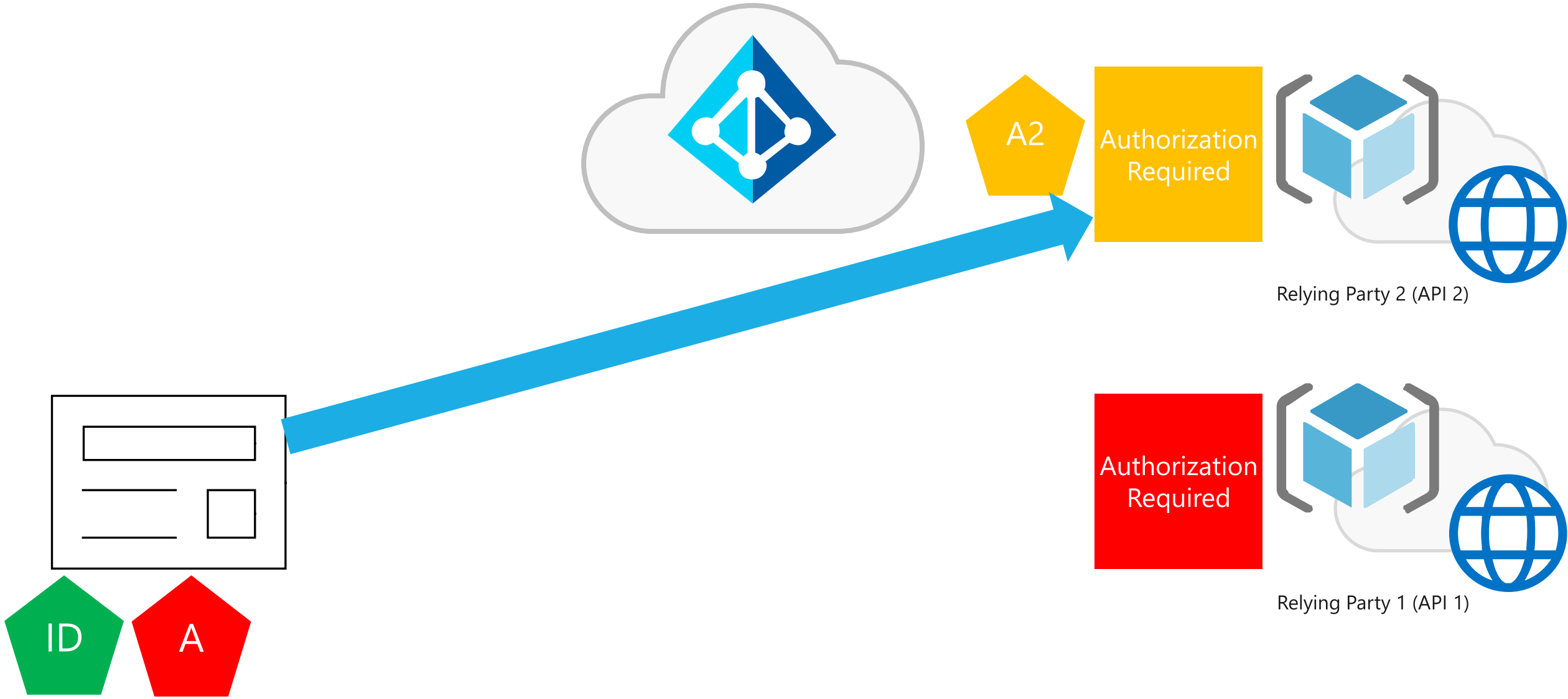


Relying Party (API)

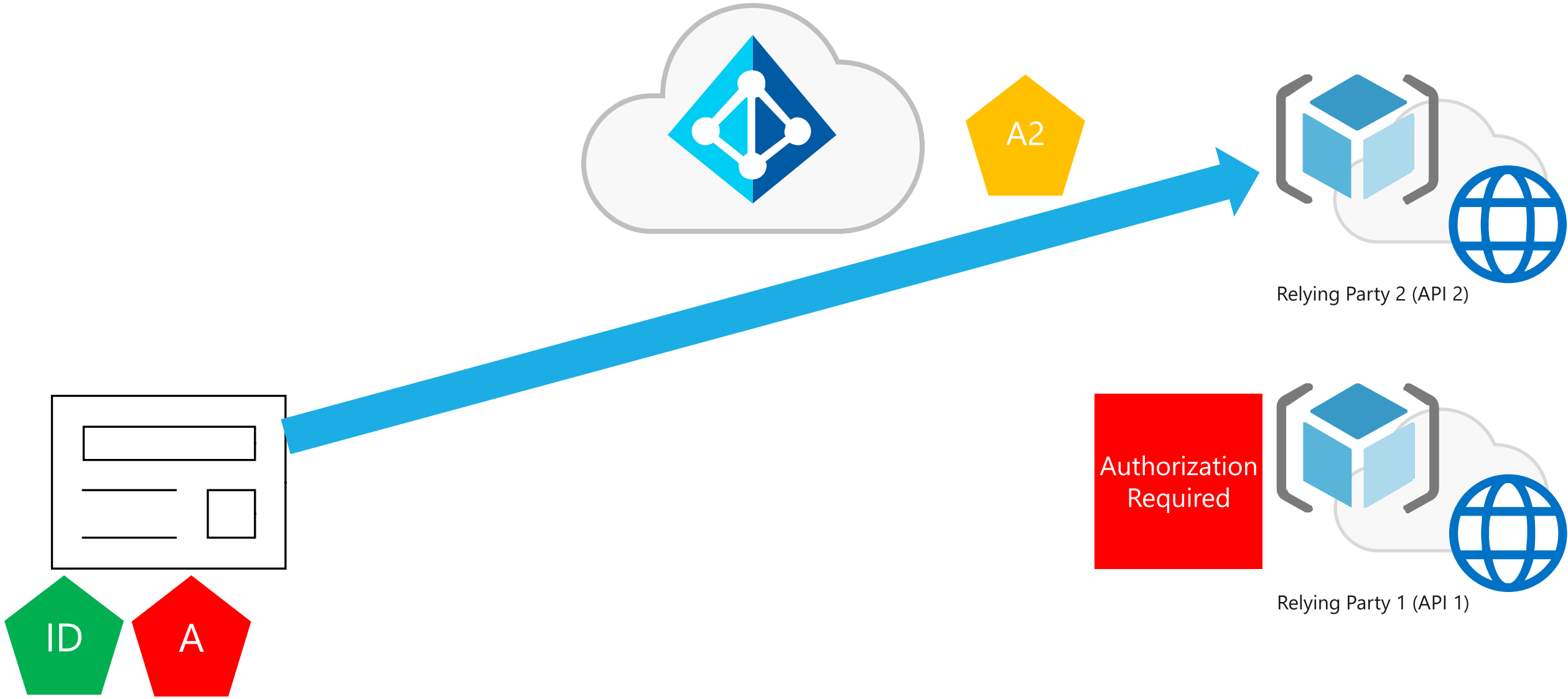
App has a token to call API



Call the API


A2
Authorization Required
Relying Party 2 (API 2)
Authorization Required
Relying Party 1 (API 1)
ID
A

Call the API


A2
Relying Party 2 (API 2)
Authorization Required
Relying Party 1 (API 1)
ID
A

Golden Rules for Tokens

1. Use tokens only for their intended function

   ID Tokens identify the user to the application

   Access tokens authorize the app for a range of operations for an API

2. Never mess with someone else's tokens

   Apps do not look at access tokens

   APIs never use ID Tokens for authorization

   APIs never accept Access Tokens for other APIs

3. Cache the tokens appropriately

   Do not invent the wheel. Use proved SDKs for your development stack

4. Obey the permissions given

   Access tokens contain scopes and roles. Do not allow more than the permissions given.

Demo: Authenticate & calling APIs from your app

“What about debugging?”

Do Services Authenticate?

Client Credentials

- Application can get access token without ID token
- Client Credentials identify the application
  - Secret Key
  - Certificate
  - Managed Identity for Azure Resources - Supporting Resources
- Avoid using a “Service Account”

- Use the Confidential Client in MSAL
  - MSAL .NET Daemon sample

Protocols do not guarantee security, Developers do!

- Modern protocols have flaws:
  - Token replay to client (implicit Grant) – Nonce
  - Token replay to server – Timestamps, session IDs
  - Token caching
  - Token lifetimes

- Make sure your developers are using a library or are fully invested in implementing the protocol and its recommendations
- Microsoft Libraries (MSAL) and ASP.Net middleware best choice when using Microsoft identity platform

USE MFA!

It is free!

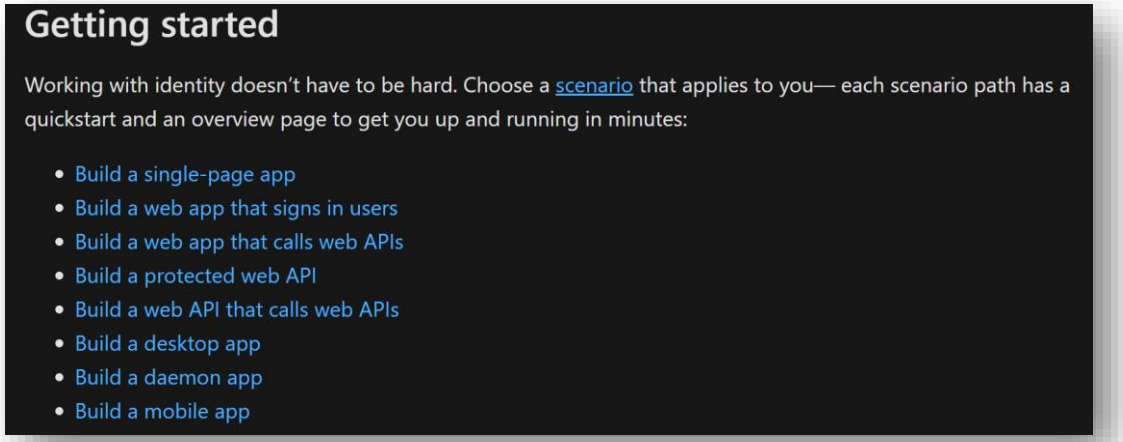
Let’s get coding

- Pick your tenant
  - Free Azure trial subscription (12 month)
  - MSDN
  - Dev/Test subscription under EA
- Some operations you will be doing
  - Adding apps
  - Adding/Updating/Deleting Users, Groups, etc.
  - Will need to be a global admin
- Pick your environment
  - Visual Studio Code
  - Visual Studio
  - Bring Your Own Environment

Coding Exercise

Adding Authentication to your App

Quick starts



- Go to: https://aka.ms/AuthLab
- Choose one of the following options:
  - Build a single-page app
  - Build a web app that signs in user (than choose your language)

- For ASP.NET if Roselyn error occurs:
  - Run Update-Package Microsoft.CodeDom.Providers.DotNetCompilerPlatform -r