Authentication and Authorization in ASP.NET Core

Roland Guijt

MVP | CONSULTANT | TRAINER | AUTHOR

@rolandguijt rolandguijt.com

The Plan

- Authentication
  - ASP.NET Core Identity
  - Token service
- Authorization

Overview

Authentication

ASP.NET Core Identity
Identity Server

Authorization

ASP.NET Core
Authorization

Part One: Authentication with ASP.NET Core Identity

ASP.NET Core Identity

A framework that helps you to manage and authenticate users per application

Features

Login and logout

Register

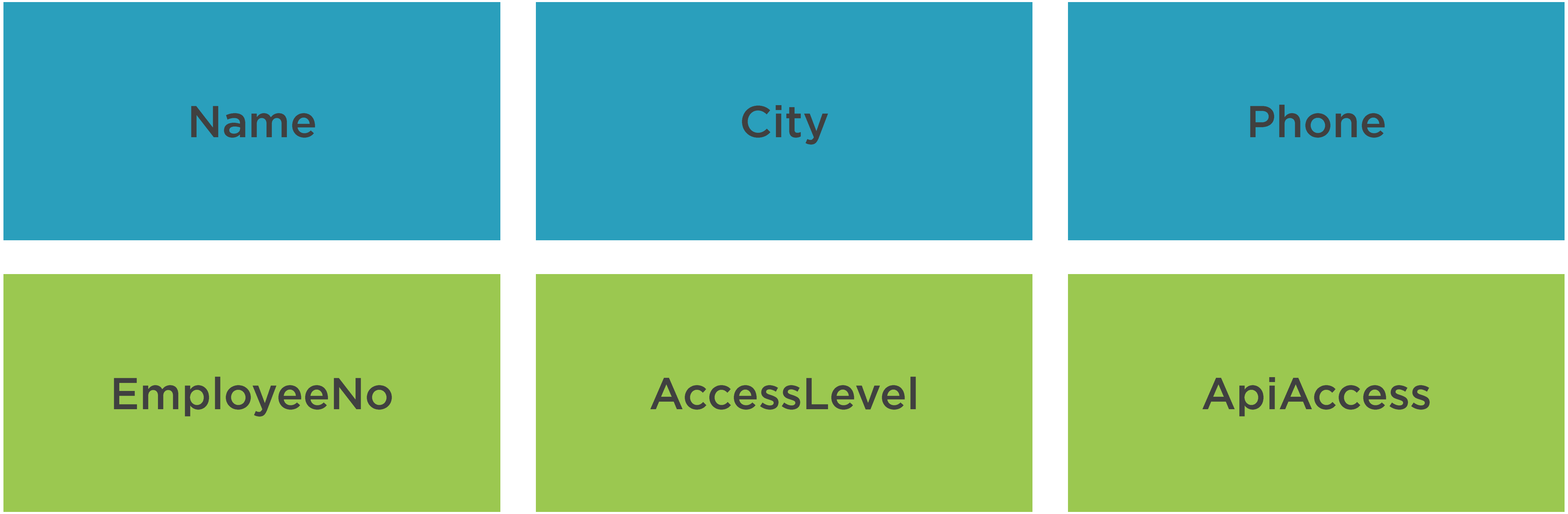
Third party logins

Password management

Lockout

Two-factor authentication

Claims-based Identity



Name

City

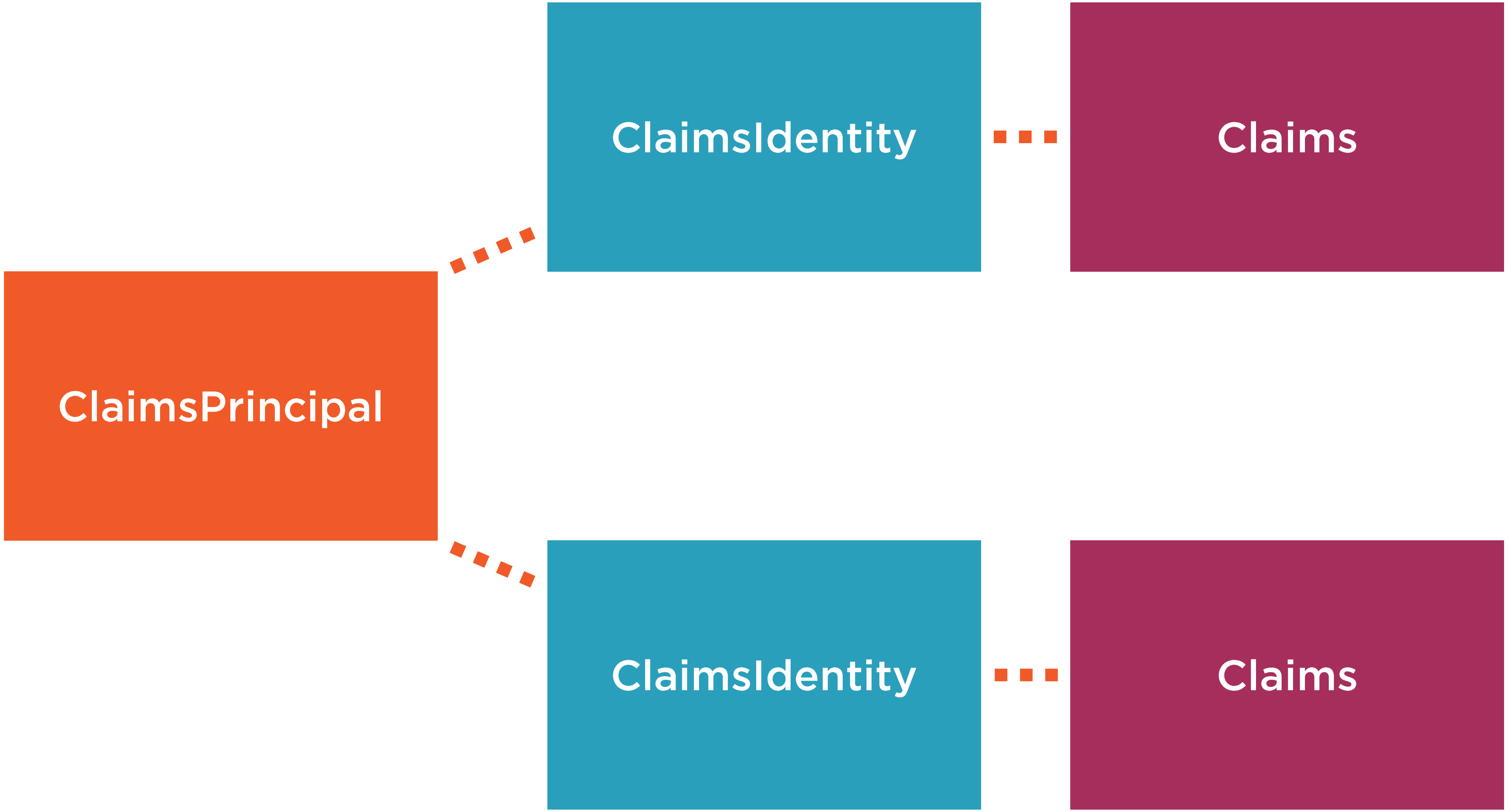
Phone

EmployeeNo

AccessLevel

ApiAccess

Claims-based Identity

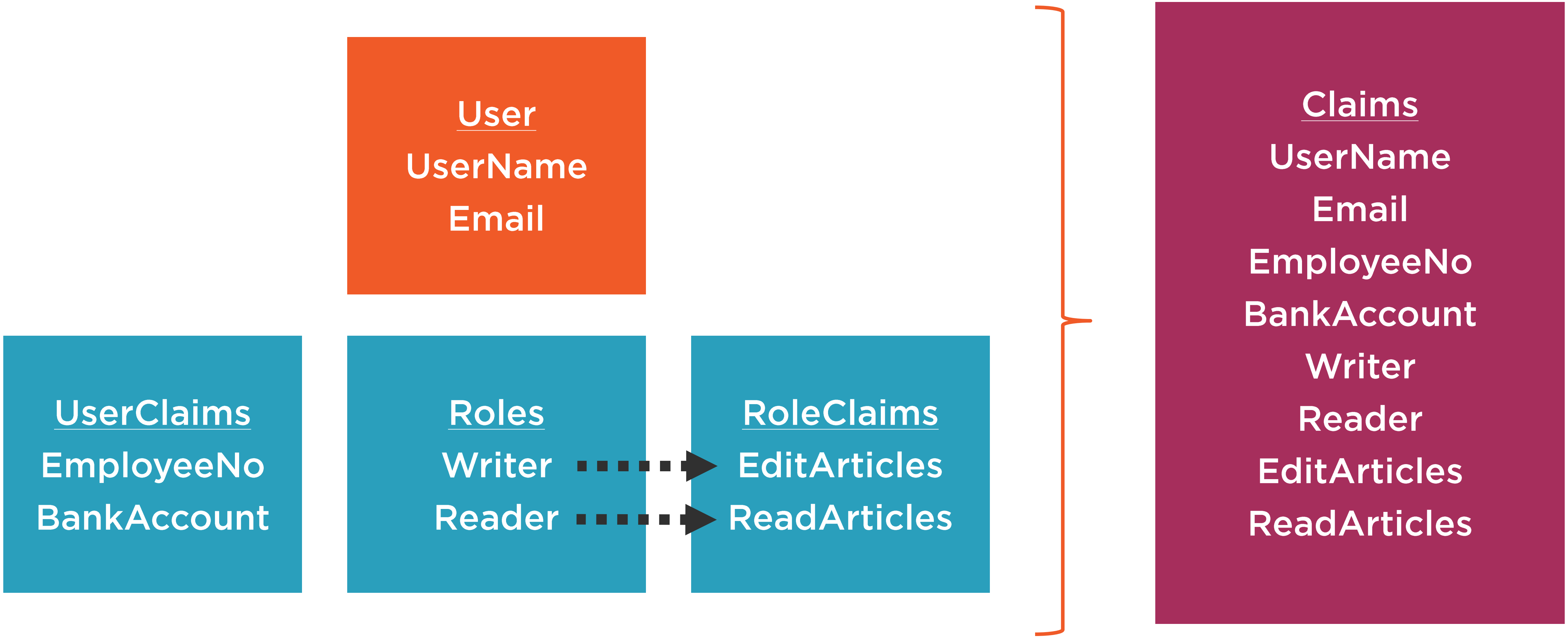


@rolandguijt

Demo

Show me the code

List Of Claims



User
UserName
Email

UserClaims
EmployeeNo
BankAccount

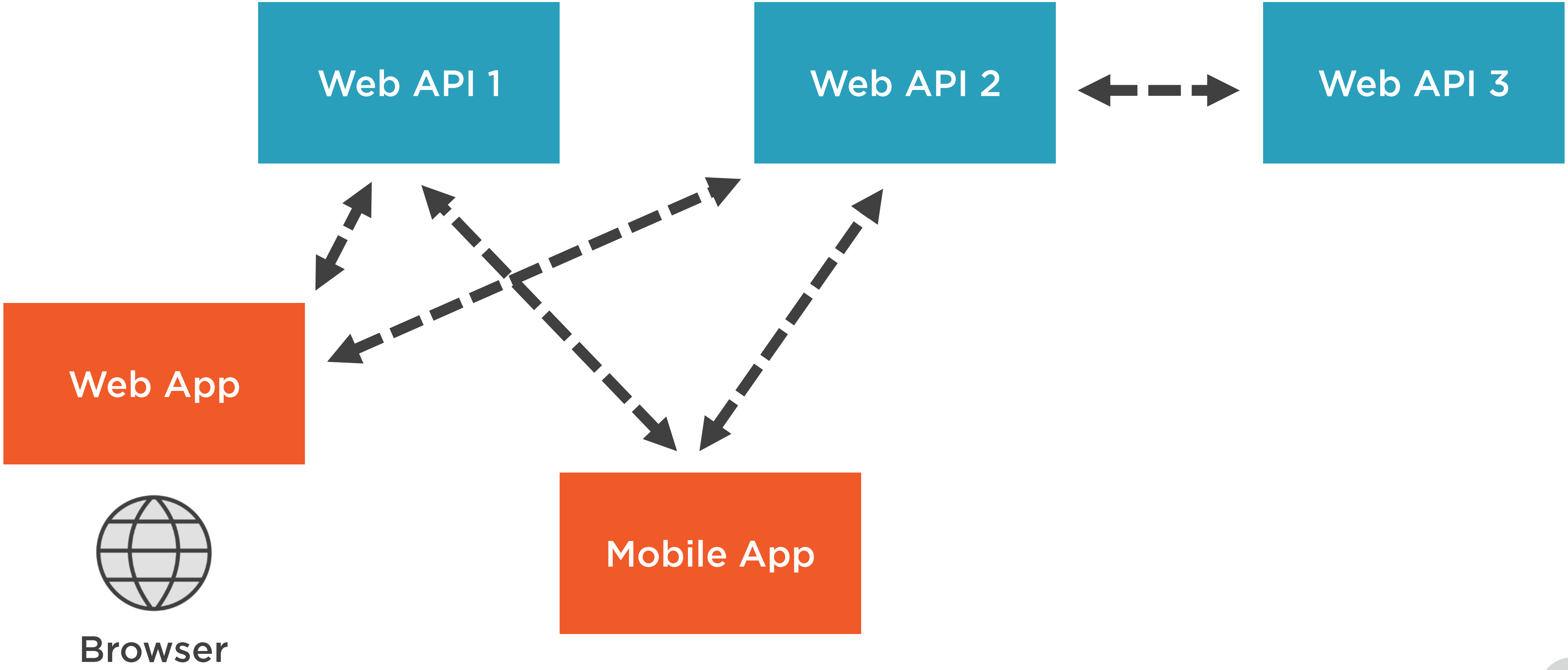
Roles
Writer
Reader

RoleClaims
EditArticles
ReadArticles

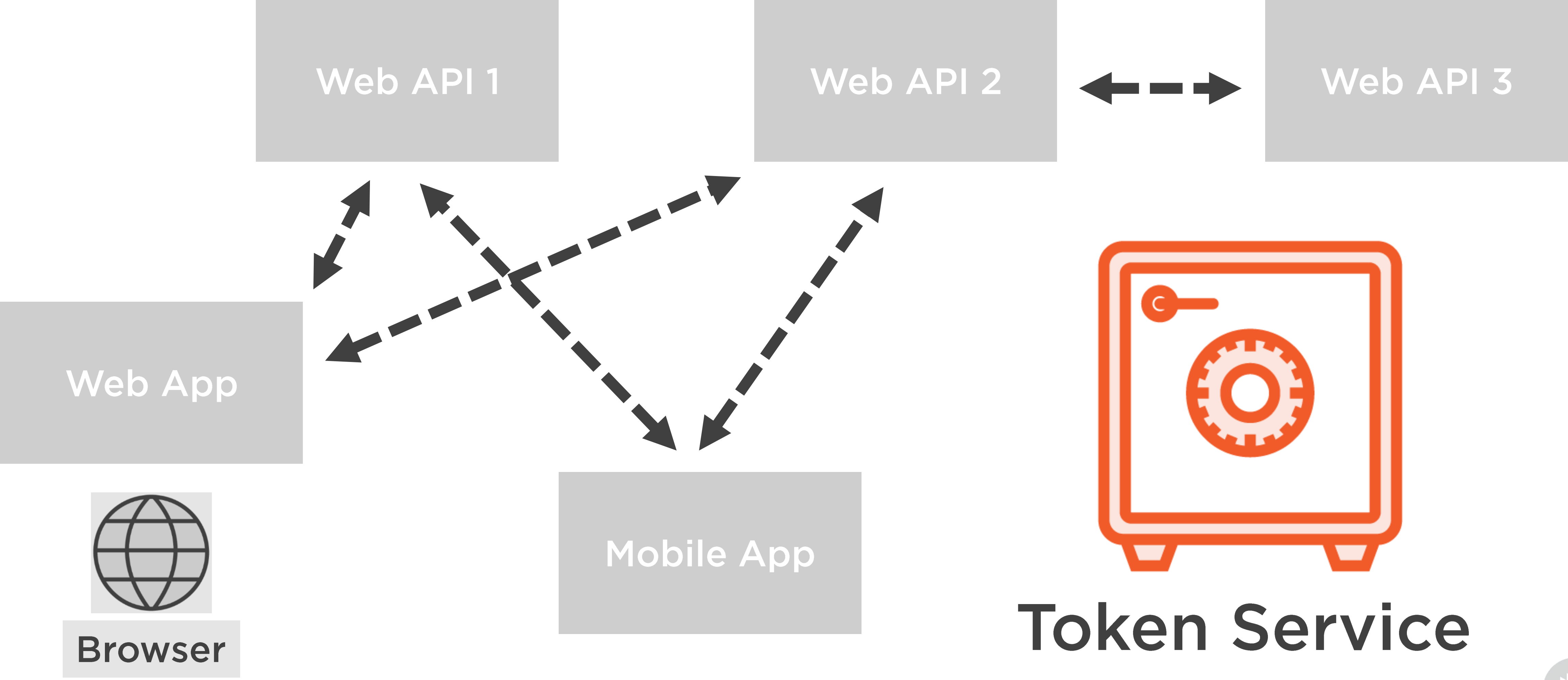
Claims
UserName
Email
EmployeeNo
BankAccount
Writer
Reader
EditArticles
ReadArticles

Part Two: Authentication with a Token Service Implementing IdentityServer

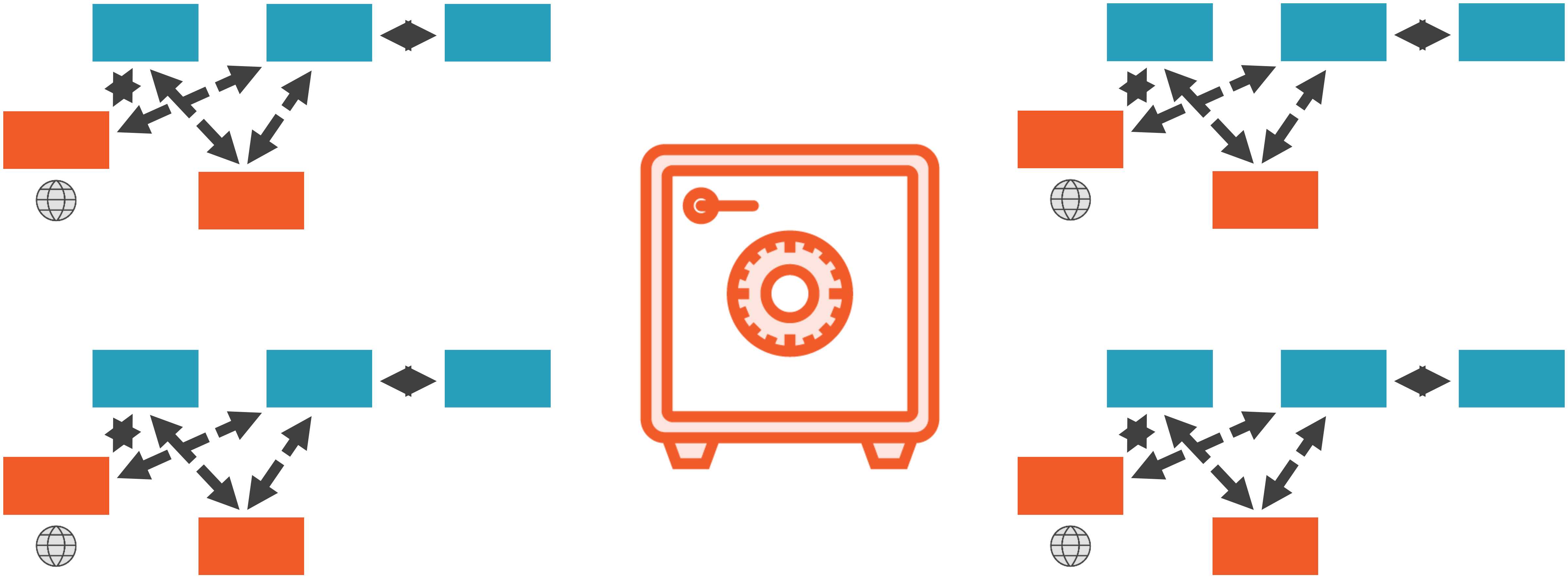
A Typical Modern Application



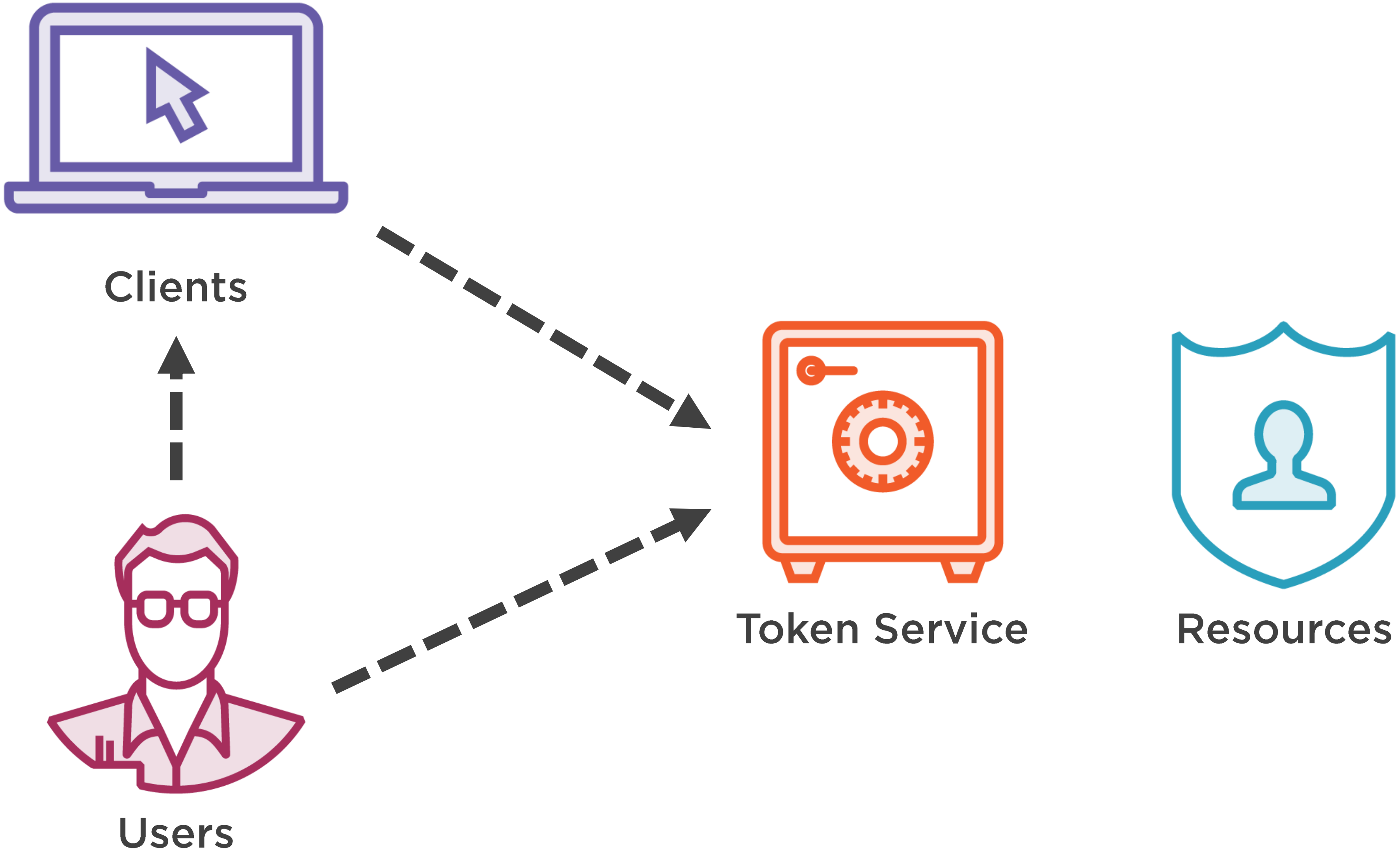
The Token Service



One Token Service To Rule Them All



Concepts



Clients

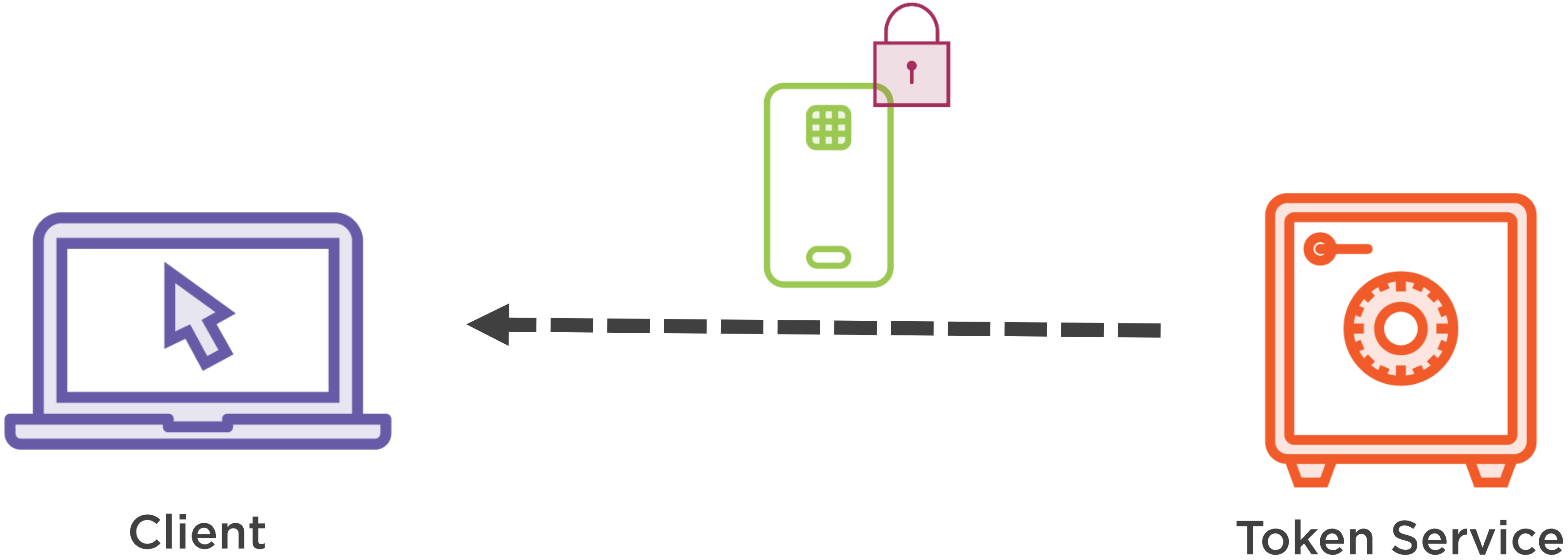
Users

Token Service

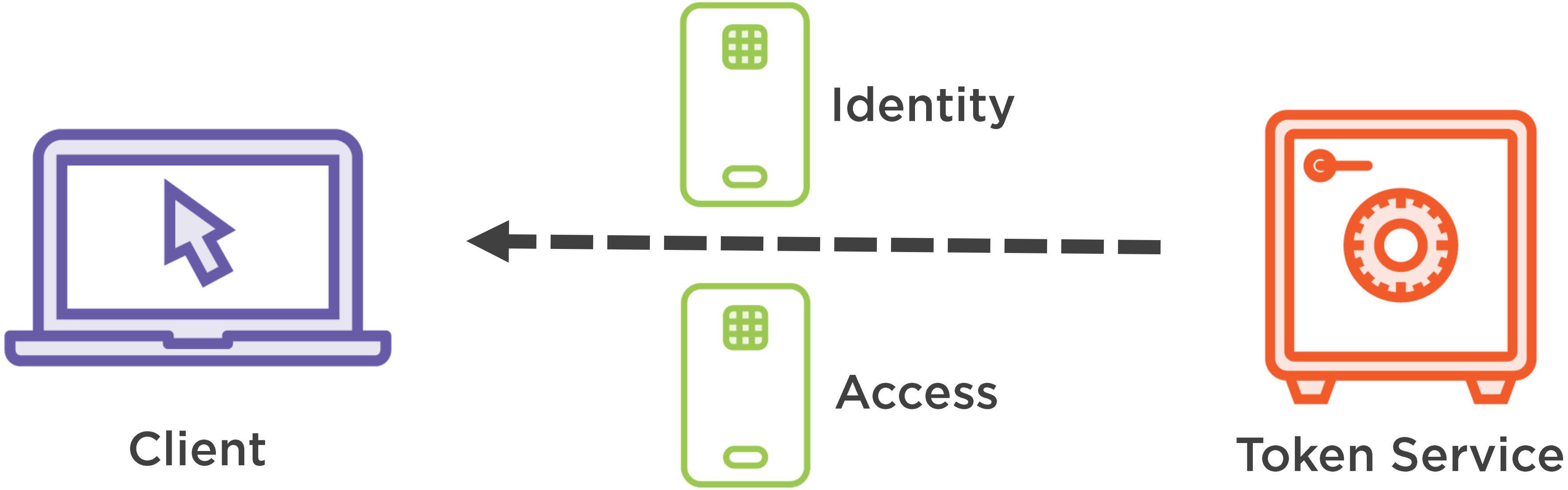
Resources

@rolandguijt

Tokens



Tokens



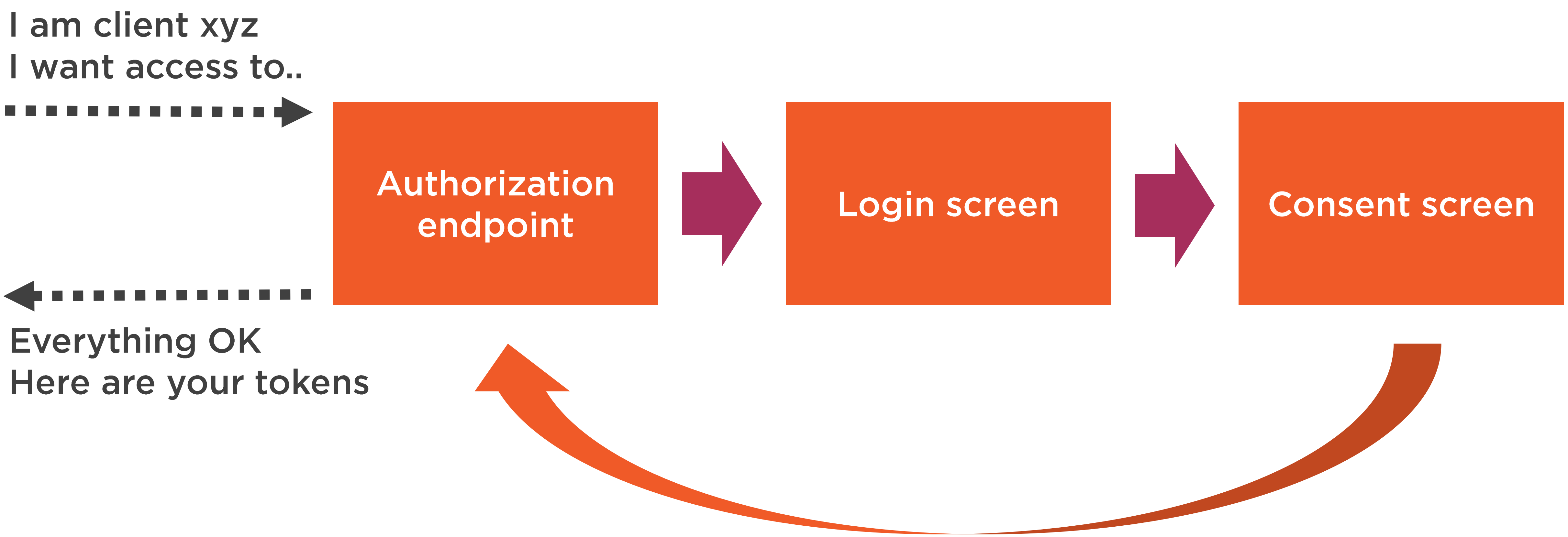
The JWT Token

```
{
  "typ": "JWT",
  "alg": "HS256"
}
{
  "iss": “tokenservice.com",
  "exp": 1300819380,
  “sub": “12xde8aacxzz”,
  "name": “Roland Guijt",
  “role": “Admin”
}
```

Standards

Identity	OpenID Connect (OIDC)
Access	OAuth2

Authenticating with a Token Service



I am client xyz
I want access to..

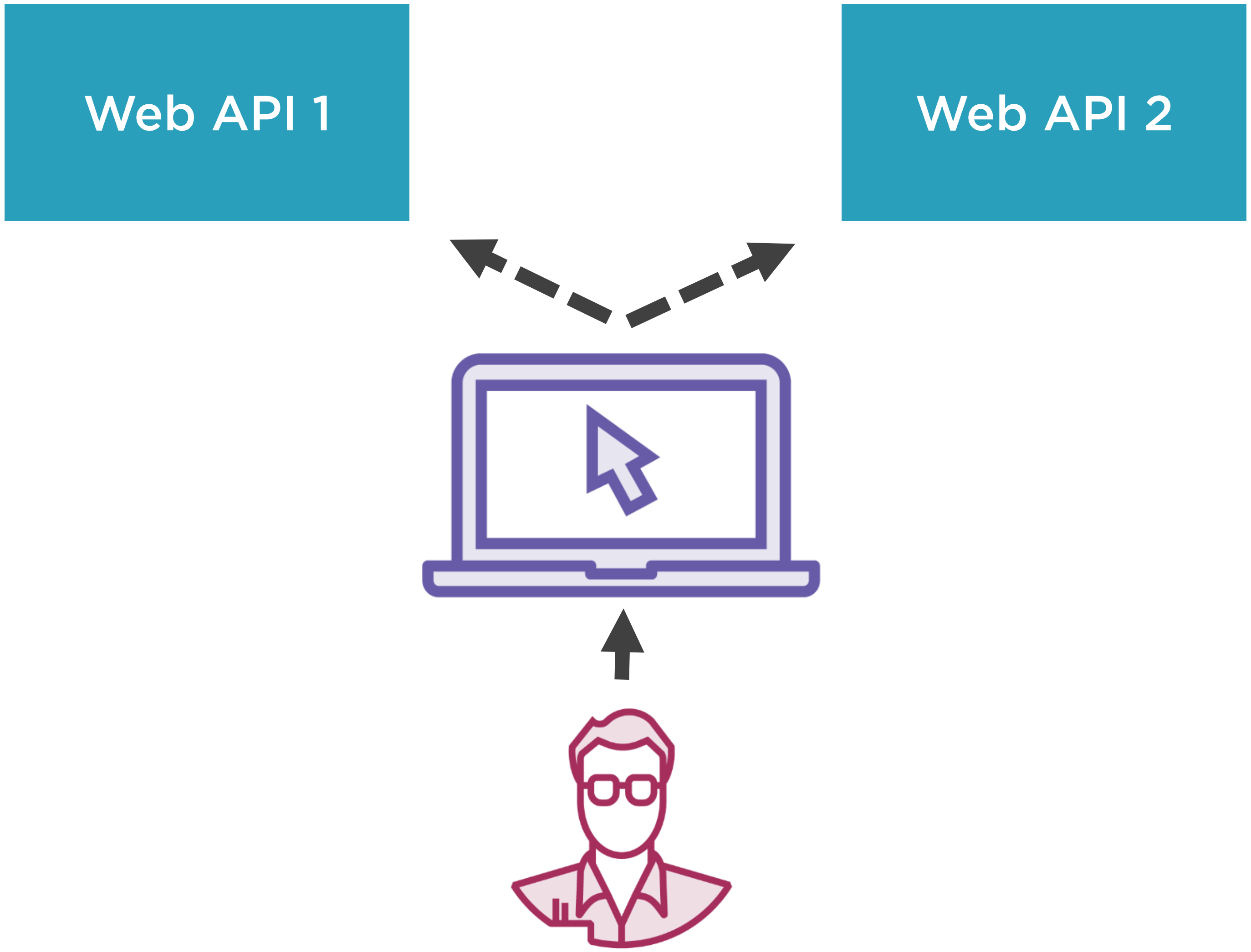
Authorization endpoint

Login screen

Consent screen

Everything OK
Here are your tokens

Tokens: Typical Application Scenario



Demo

Show me the code

Demo

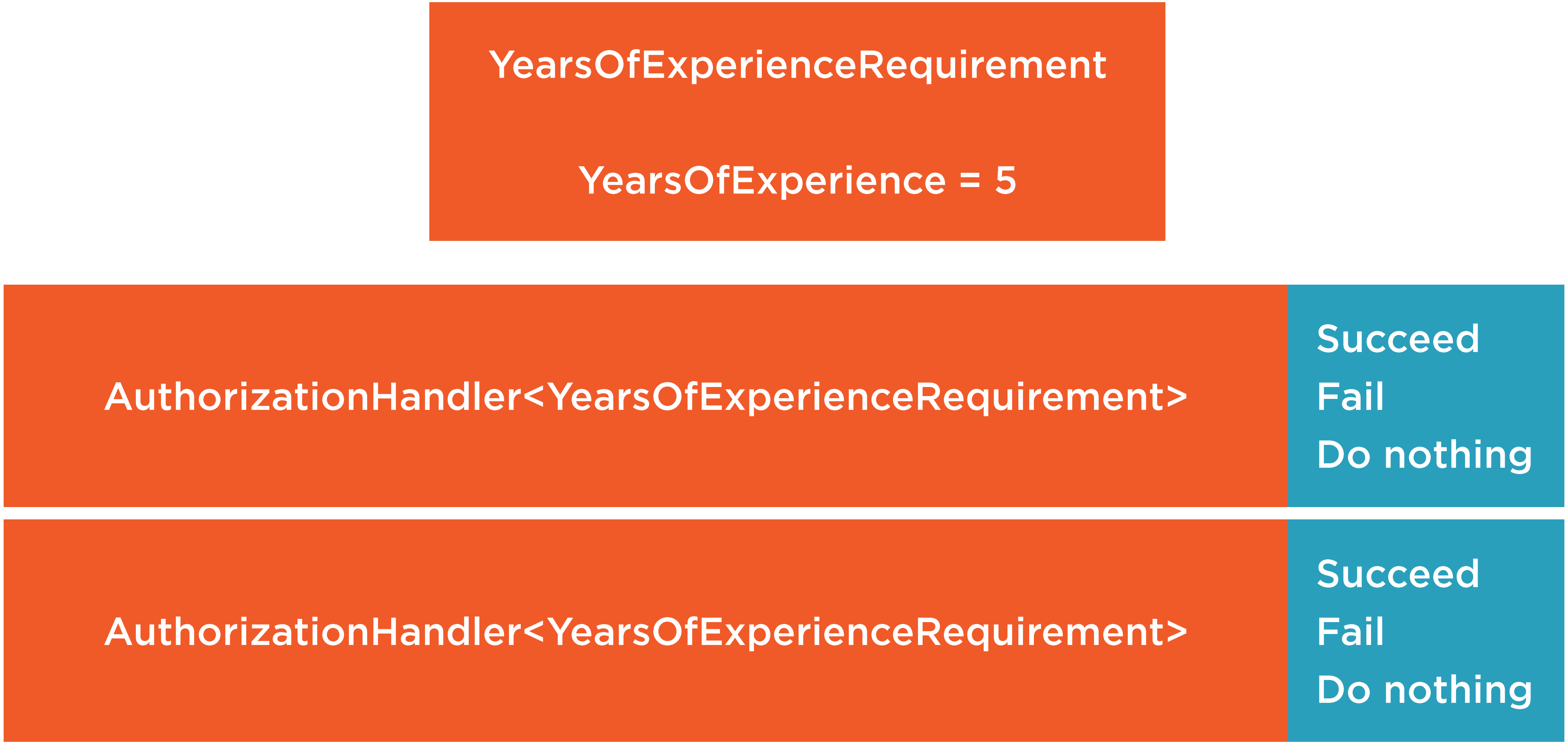
Show me the code

Part Three: Authorization

Demo

Show me the code

Requirements and Handlers



YearsOfExperienceRequirement

YearsOfExperience = 5

AuthorizationHandler<YearsOfExperienceRequirement>

Succeed

Fail

Do nothing

AuthorizationHandler<YearsOfExperienceRequirement>

Succeed

Fail

Do nothing

Authorization Data

Identity Token is about the user's identity

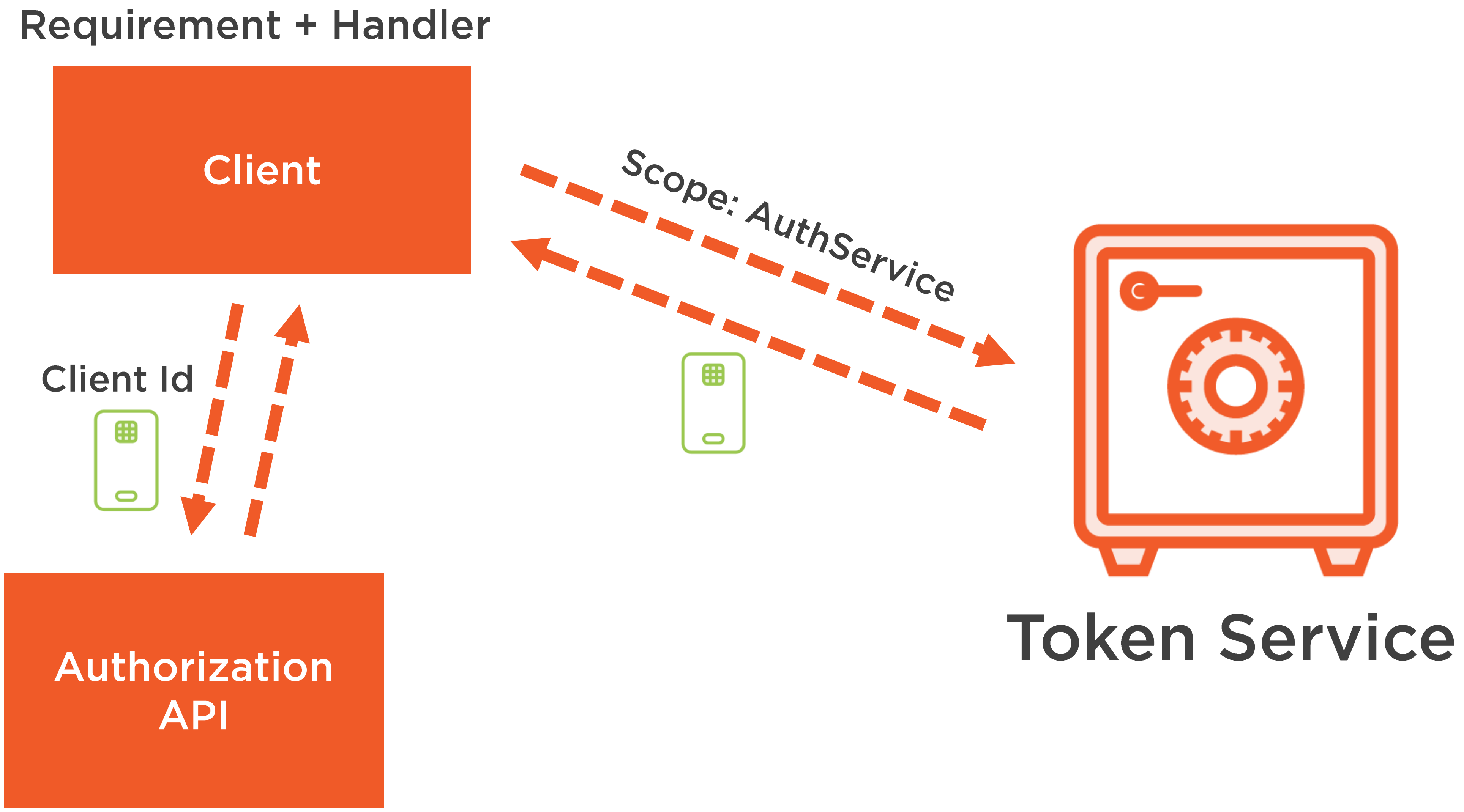
Not Authorization Data

Bloated identity token could cause problems

Context for each client or API is different

Suggestion: Create Authorization API

Architecture With Authorization API



Requirement + Handler

Client

Client Id

Scope: AuthService

Authorization API

Token Service

Take Aways

Choose your authentication method wisely

Use policies to authorize the user

Think about where the authorization data should come from