

Introduction to OpenID Connect and OAuth2 for ASP.NET Core

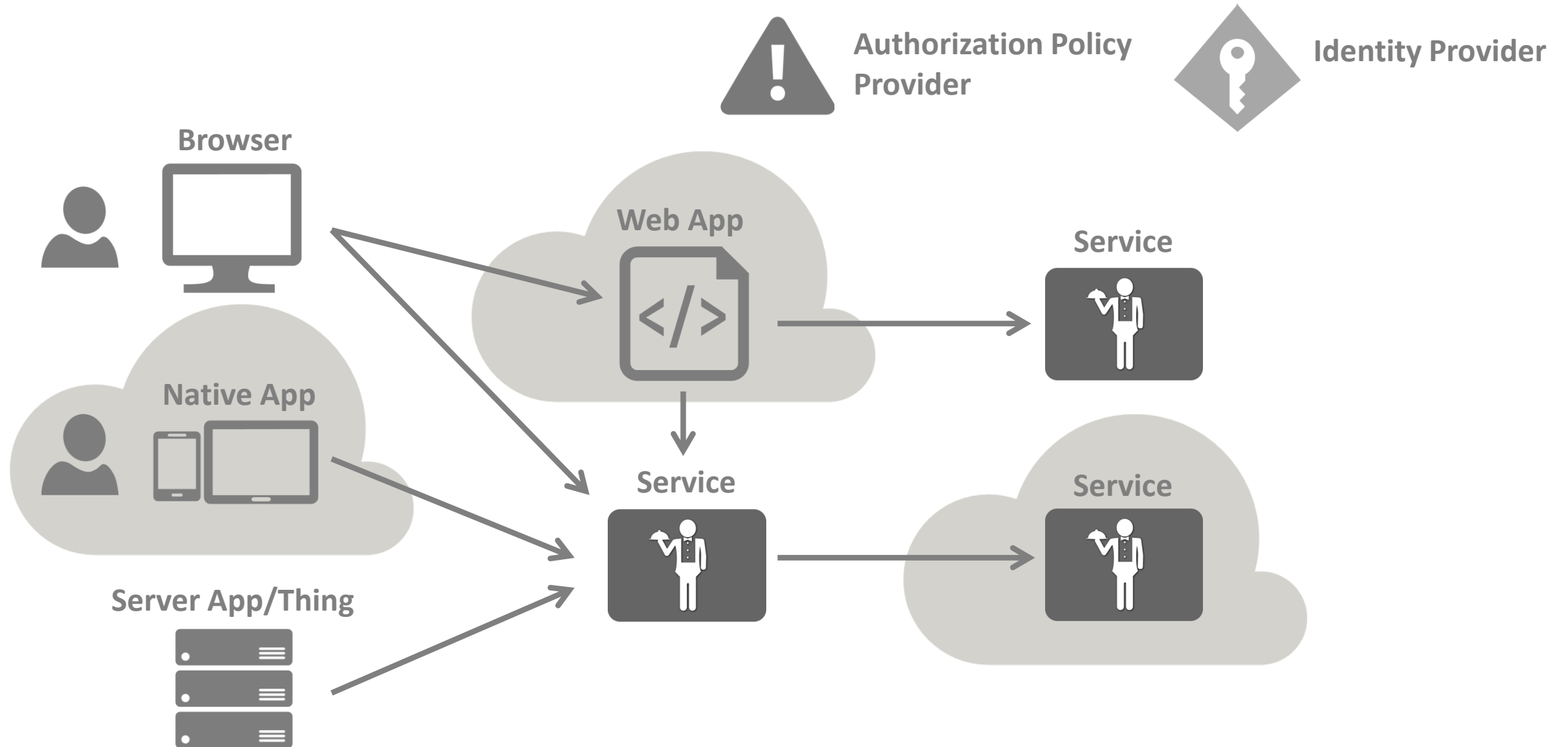
Brock Allen
Solliance, Inc.
@BrockLAllen



Code Again for
the First Time!



Modern Application Architecture



Agenda

- **Brief overview of ASP.NET Core authentication**
- **Centralized Authentication with OpenID Connect**
- **Protecting Web APIs and OAuth 2.0**

Authentication in ASP.NET Core

- **Combination of middleware and authentication handlers in DI**
 - middleware invokes handlers for request related processing
 - handlers can be also invoked manually
- **Handlers implement specific authentication methods**
 - Cookies for browser based authentication
 - Google, Facebook, and other social authentication
 - OpenId Connect for external authentication
 - JSON web token (JWT) for token-based authentication

Interacting with the authentication system

- **Extension methods on *HttpContext* call the *IAuthenticationService* in DI**

```
public static class AuthenticationHttpContextExtensions
{
    public static Task SignInAsync(this HttpContext context, ClaimsPrincipal principal) { }
    public static Task SignInAsync(this HttpContext context, string scheme, ClaimsPrincipal principal) { }

    public static Task SignOutAsync(this HttpContext context) { }
    public static Task SignOutAsync(this HttpContext context, string scheme) { }

    public static Task ChallengeAsync(this HttpContext context) { }
    public static Task ChallengeAsync(this HttpContext context, string scheme) { }

    public static Task ForbidAsync(this HttpContext context) { }
    public static Task ForbidAsync(this HttpContext context, string scheme) { }

    public static Task<AuthenticateResult> AuthenticateAsync(this HttpContext context) { }
    public static Task<AuthenticateResult> AuthenticateAsync(this HttpContext context, string scheme) { }
}
```

Setting up authentication

- **Global settings go into DI**
 - e.g. default schemes
- **Authentication middleware invokes handlers**

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddAuthentication(options =>
    {
        options.DefaultScheme = "Cookies";
    });
}

public void Configure(IApplicationBuilder app)
{
    app.UseAuthentication();
}
```

Cookie Authentication

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddAuthentication(defaultScheme: "Cookies")
        .AddCookie("Cookies", options =>
        {
            options.LoginPath = "/account/login";
            options.AccessDeniedPath = "/account/denied";

            options.Cookie.Name = "myapp";
            options.Cookie.Expiration = TimeSpan.FromHours(8);
            options.SlidingExpiration = false;
        });
}
```

Cookies: Logging in

- **SignInAsync issues cookie**
 - either using a named scheme, or default

```
var claims = new Claim[]
{
    new Claim("sub", "37734"),
    new Claim("name", "Brock Allen")
};

var ci = new ClaimsIdentity(claims, "password", "name", "role");
var cp = new ClaimsPrincipal(ci);

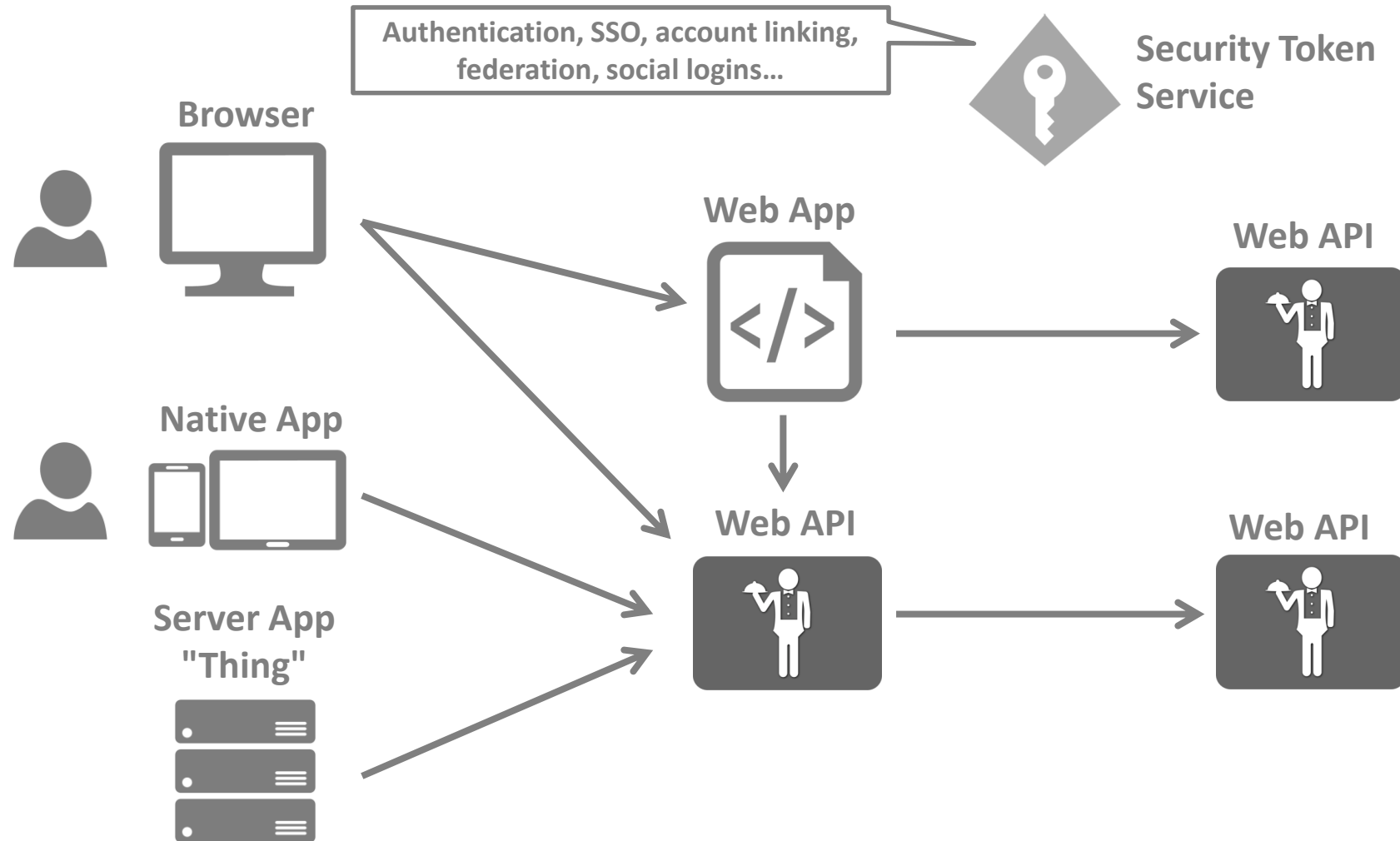
await HttpContext.SignInAsync(cp);
```


Cookies: Logging out

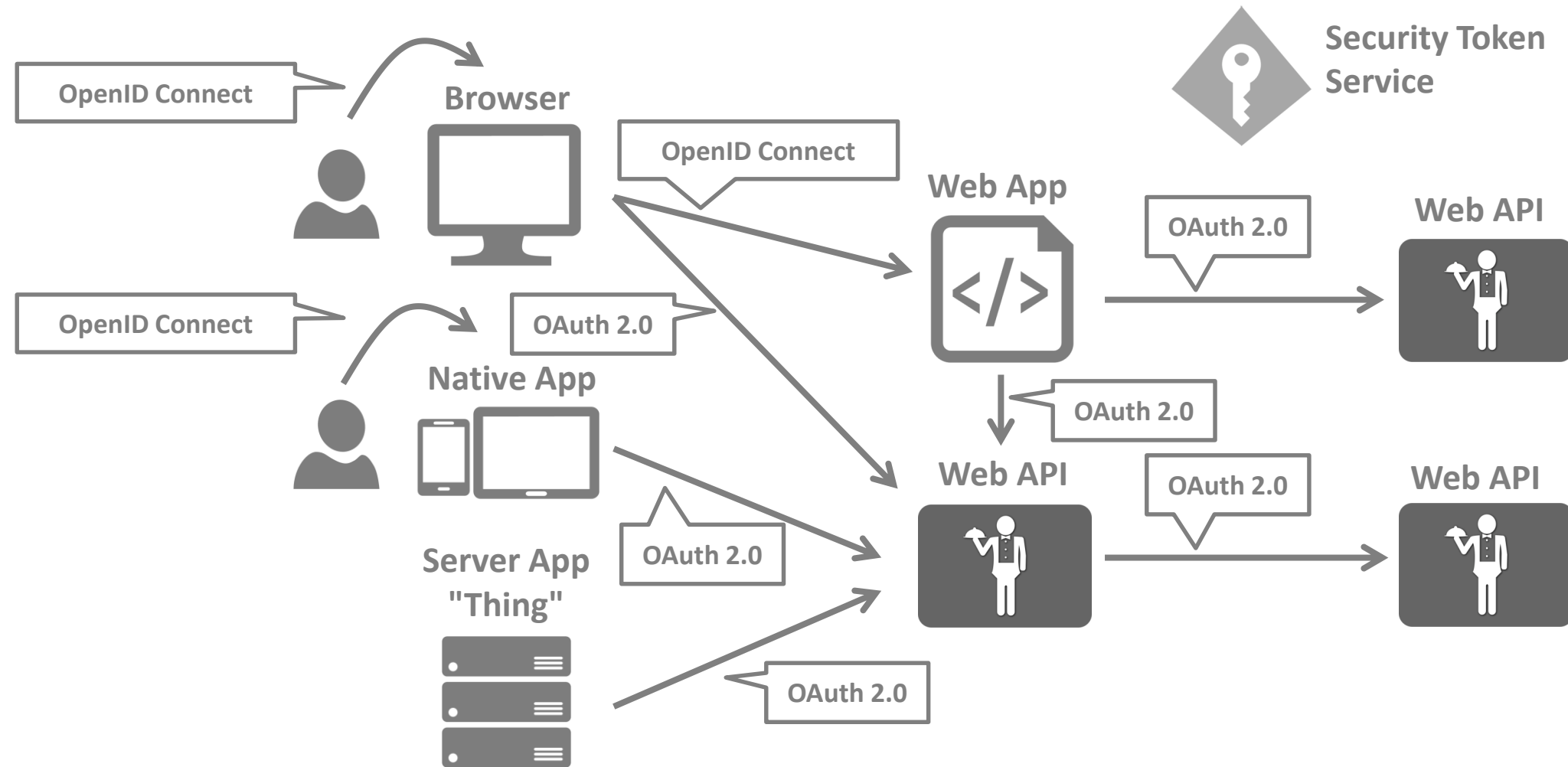
- **SignInAsync removes cookie**

```
await HttpContext.SignOutAsync();
```

The way forward...

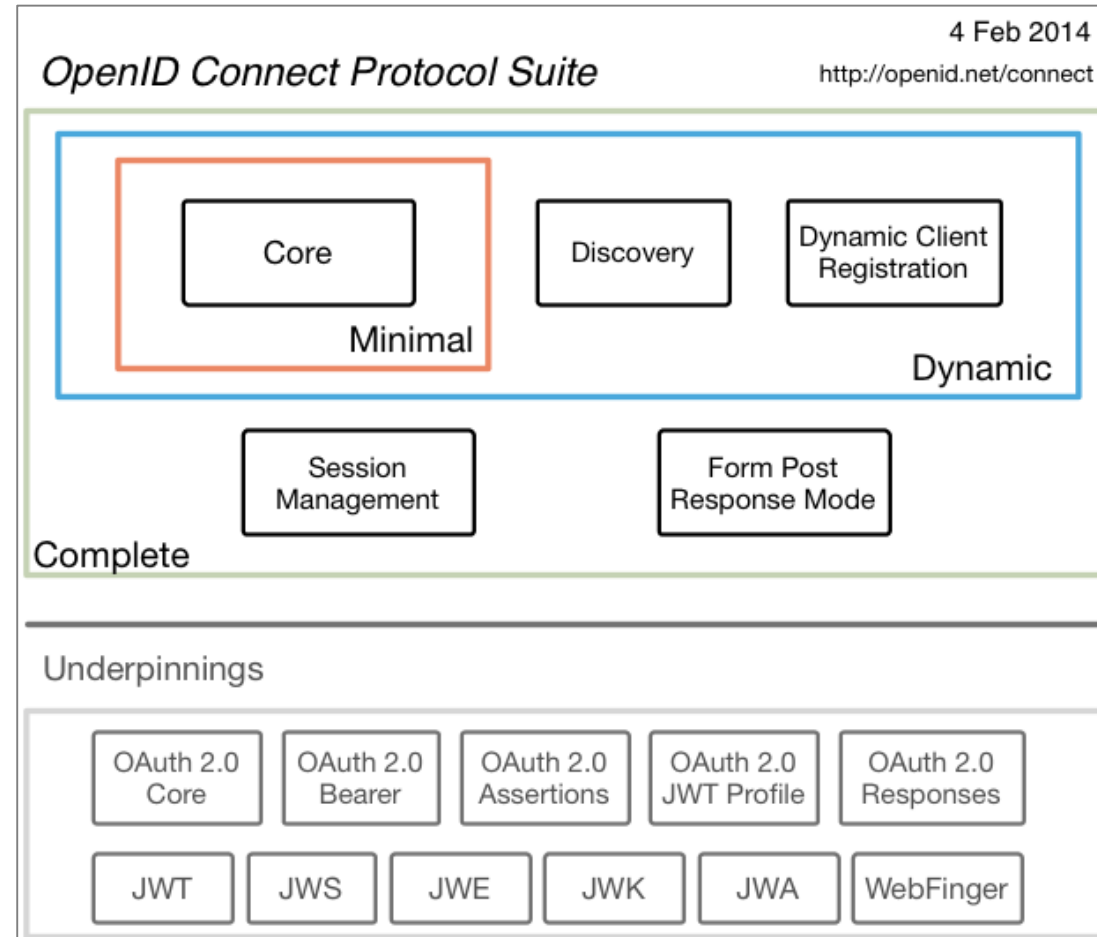


Security Protocols



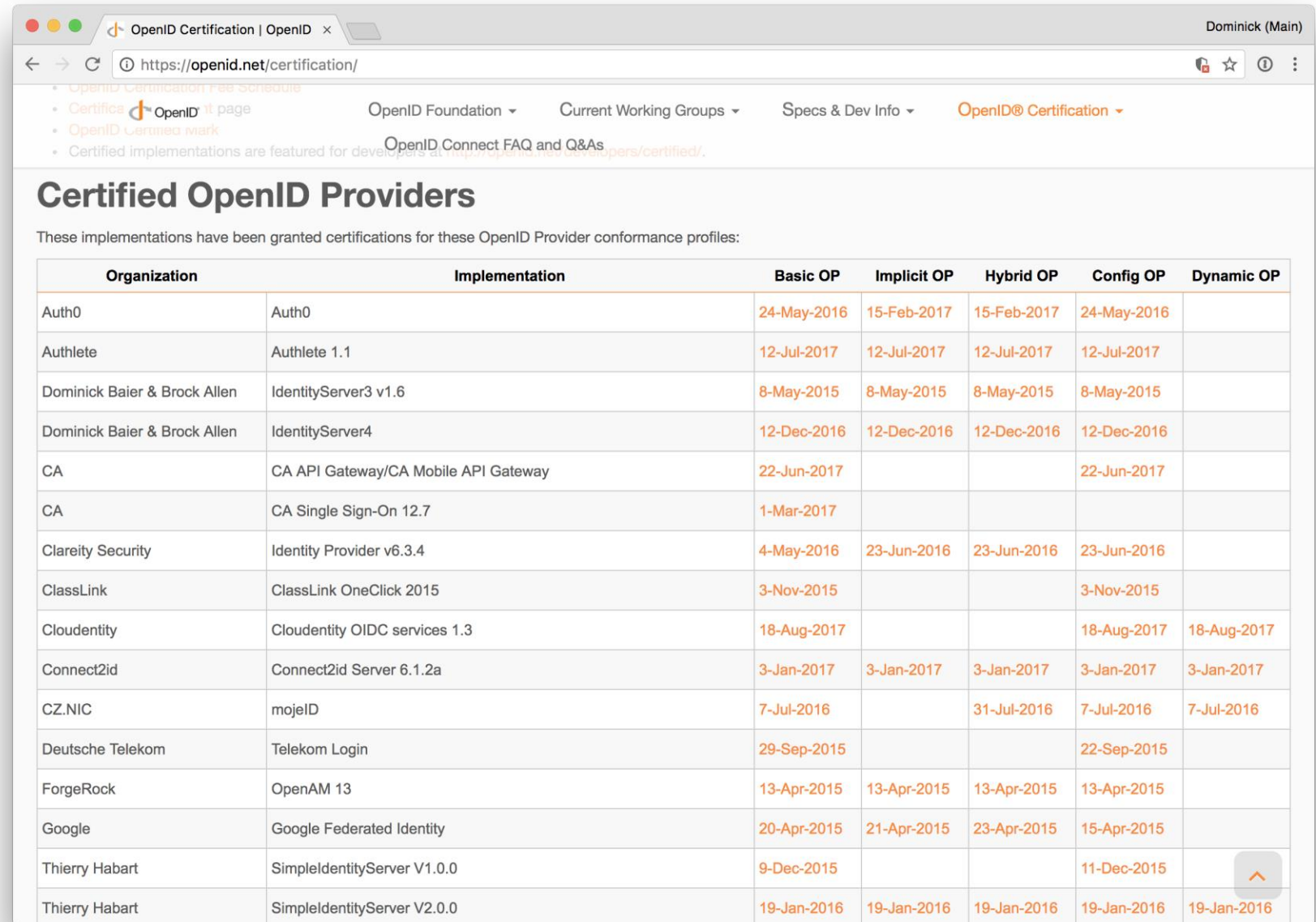


<http://openid.net/connect/>



OpenID Connect Certification

for providers and
client libraries




The screenshot shows a web browser window with the URL <https://openid.net/certification/>. The page title is "OpenID Certification | OpenID". The navigation bar includes links for "OpenID Foundation", "Current Working Groups", "Specs & Dev Info", and "OpenID® Certification". Below the navigation bar, there is a section titled "Certified OpenID Providers" with the text "These implementations have been granted certifications for these OpenID Provider conformance profiles:". Below this text is a table listing certified providers and their conformance profiles.

Organization	Implementation	Basic OP	Implicit OP	Hybrid OP	Config OP	Dynamic OP
Auth0	Auth0	24-May-2016	15-Feb-2017	15-Feb-2017	24-May-2016	
Authlete	Authlete 1.1	12-Jul-2017	12-Jul-2017	12-Jul-2017	12-Jul-2017	
Dominick Baier & Brock Allen	IdentityServer3 v1.6	8-May-2015	8-May-2015	8-May-2015	8-May-2015	
Dominick Baier & Brock Allen	IdentityServer4	12-Dec-2016	12-Dec-2016	12-Dec-2016	12-Dec-2016	
CA	CA API Gateway/CA Mobile API Gateway	22-Jun-2017			22-Jun-2017	
CA	CA Single Sign-On 12.7	1-Mar-2017				
Clareity Security	Identity Provider v6.3.4	4-May-2016	23-Jun-2016	23-Jun-2016	23-Jun-2016	
ClassLink	ClassLink OneClick 2015	3-Nov-2015			3-Nov-2015	
Cloudentity	Cloudentity OIDC services 1.3	18-Aug-2017			18-Aug-2017	18-Aug-2017
Connect2id	Connect2id Server 6.1.2a	3-Jan-2017	3-Jan-2017	3-Jan-2017	3-Jan-2017	3-Jan-2017
CZ.NIC	mojeID	7-Jul-2016		31-Jul-2016	7-Jul-2016	7-Jul-2016
Deutsche Telekom	Telekom Login	29-Sep-2015			22-Sep-2015	
ForgeRock	OpenAM 13	13-Apr-2015	13-Apr-2015	13-Apr-2015	13-Apr-2015	
Google	Google Federated Identity	20-Apr-2015	21-Apr-2015	23-Apr-2015	15-Apr-2015	
Thierry Habart	SimpleIdentityServer V1.0.0	9-Dec-2015			11-Dec-2015	
Thierry Habart	SimpleIdentityServer V2.0.0	19-Jan-2016	19-Jan-2016	19-Jan-2016	19-Jan-2016	19-Jan-2016

IdentityServer

GitHub, Inc. [US] | <https://github.com/IdentityServer>

This organization Search Pull requests Issues Gist

 **IdentityServer**
https://identityserver.io | identity@leastprivilege.com

Repositories People 6 Teams 6 Projects 0 Settings

Pinned repositories Customize pinned repositories

IdentityServer4
OpenID Connect and OAuth 2.0 Framework for ASP.NET Core
C# 995 303

IdentityServer3
OpenID Connect Provider and OAuth 2.0 Authorization Server Framework for ASP.NET 4.x/Katana
C# 1.8k 738

IdentityServer4.AccessTokenValidation
IdentityServer Access Token Validation for ASP.NET Core
C# 50 38

IdentityServer3.AccessTokenValidation
OWIN Middleware to validate access tokens from IdentityServer3
C# 57 70

IdentityServer4.Samples
Samples for IdentityServer4
JavaScript 213 200

IdentityServer3.Samples
Samples for IdentityServer v3
JavaScript 432 946

Search repositories... Type: All Language: All

IdentityServer4
OpenID Connect and OAuth 2.0 Framework for ASP.NET Core
security identity oauth2 dotnet aspnet-core
openid-connect identityserver4
C# 995 303 Updated 14 hours ago

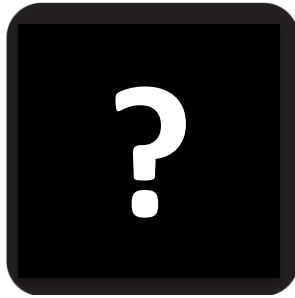
IdentityServer4.Quickstart.UI
Starter UI for in-memory IdentityServer4

Top languages
C# JavaScript CSS HTML

Most used topics
aspnet-core dotnet
identityserver4 oauth2
openid-connect



Endpoints



**Discovery
Endpoint**



**Authorize
Endpoint**

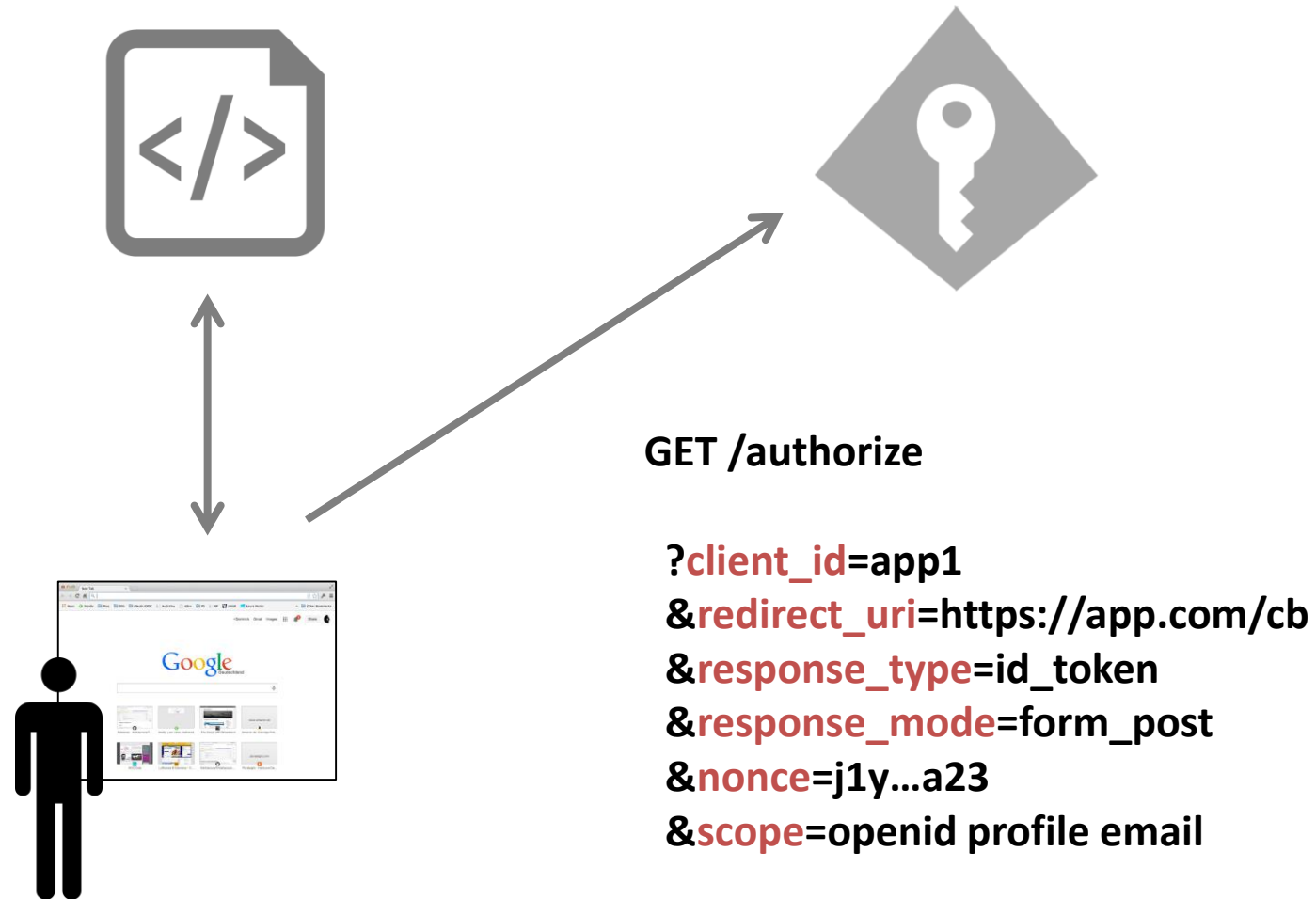


**Token
Endpoint**

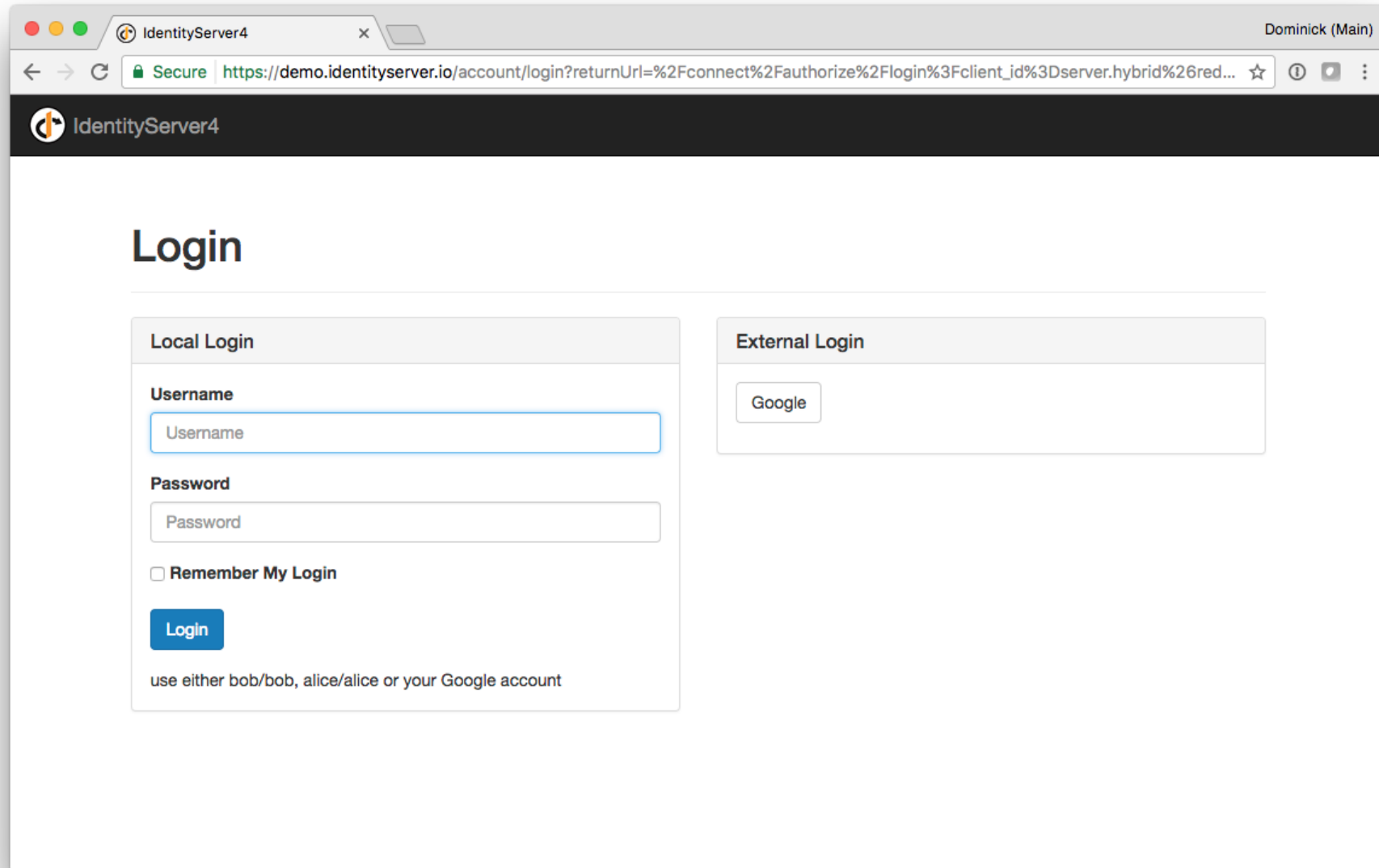
Flows

- **Implicit Flow**
 - browser-based applications
 - server web apps only using authentication / SPAs using APIs
- **Client Credentials Flow**
 - server to server API communication
 - headless devices / IoT
- **Hybrid Flow**
 - server web apps / native apps / mobile apps
 - using authentication and APIs

Implicit Flow for Web Applications



Authentication



The screenshot shows a web browser window with the title "IdentityServer4" and a user profile "Dominick (Main)" in the top right corner. The address bar shows a secure connection to "https://demo.identityserver.io/account/login?returnUrl=%2Fconnect%2Fauthorize%2Flogin%3Fclient_id%3Dserver.hybrid%26red...". The page header features the "IdentityServer4" logo and name. The main content area is titled "Login" and contains two panels: "Local Login" and "External Login". The "Local Login" panel includes input fields for "Username" and "Password", a "Remember My Login" checkbox, and a blue "Login" button. Below these fields is a note: "use either bob/bob, alice/alice or your Google account". The "External Login" panel contains a single "Google" button.

IdentityServer4

Dominick (Main)

Secure https://demo.identityserver.io/account/login?returnUrl=%2Fconnect%2Fauthorize%2Flogin%3Fclient_id%3Dserver.hybrid%26red...

IdentityServer4

Login

Local Login

Username

Password

☐ Remember My Login

Login

use either bob/bob, alice/alice or your Google account

External Login

Google

Consent

IdentityServer4 x Dominick (Main)

Secure https://demo.identityserver.io/consent?returnUrl=%2Fconnect%2Fauthorize%2Fconsent%3Fclient_id%3Dserver.hybrid%26redire... ☆ ⓘ ⌵

IdentityServer4 Dominick Baier ▾

Server-based Client (Hybrid) is requesting your permission

Uncheck the permissions you do not wish to grant.

Personal Information

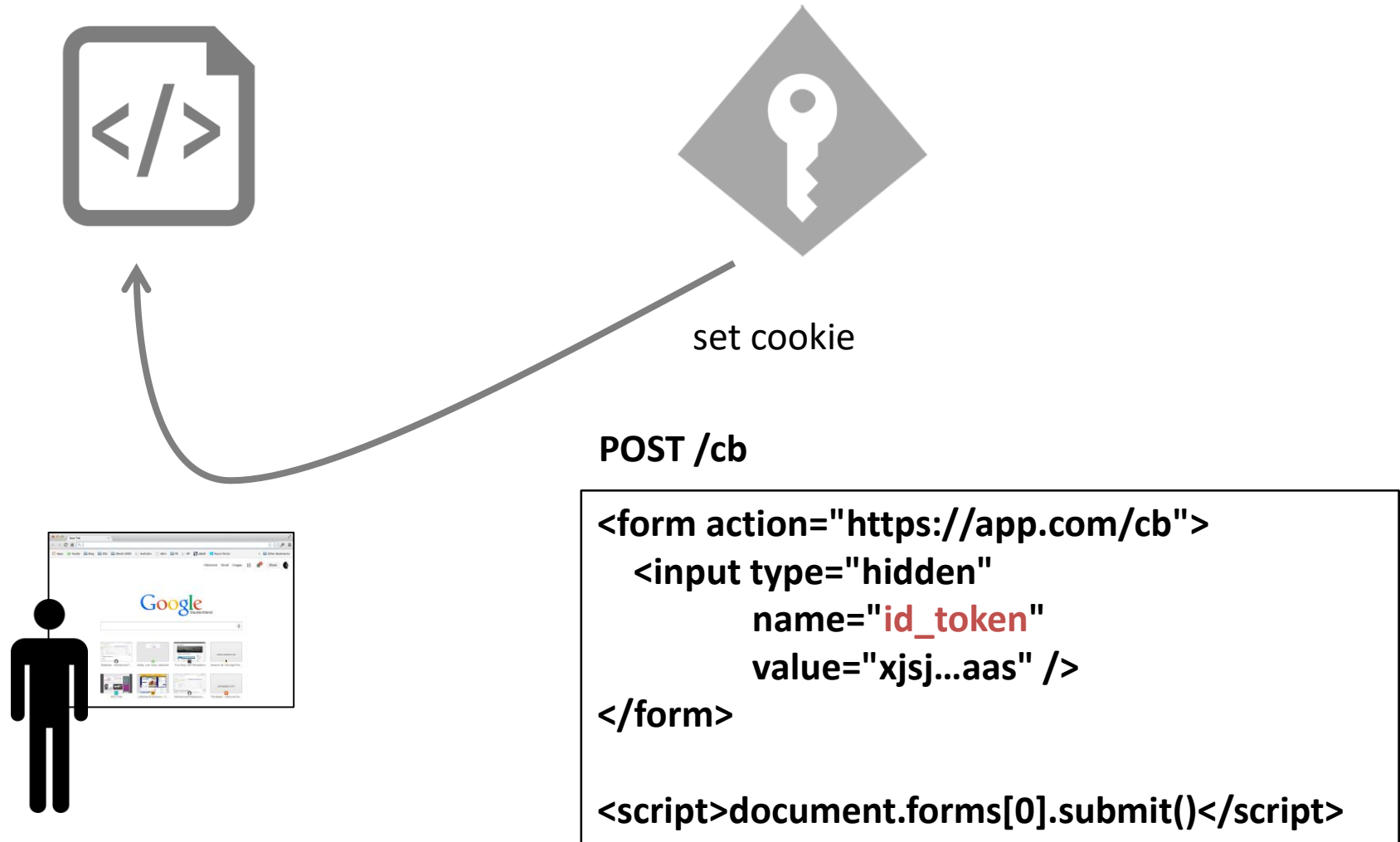
☒ Your user identifier *(required)*

☒ User profile ⓘ
Your user profile information (first name, last name, etc.)

☒ Your email address ⓘ

☒ Remember My Decision

Response



Identity Token

Header

```
{  
  "typ": "JWT",  
  "alg": "RS256",  
  "kid": "mj399j..."  
}
```

Payload

```
{  
  "iss": "https://issuer",  
  "exp": 1340819380,  
  "iat": 1340818761,  
  "aud": "app1",  
  "nonce": "j1y...a23",  
  "amr": [ "pwd" ],  
  "auth_time": 12340819300  
  
  "sub": "182jmm199",  
  "name": "Alice",  
}
```

eyJhbGciOiJIub251In0.eyJpc3MiOiJqb2UiLA0KICJleHAiOjEzMD.4MTkzODAsDQogImh0dHA6Ly9leGFt

Header

Payload

Signature

Discovery



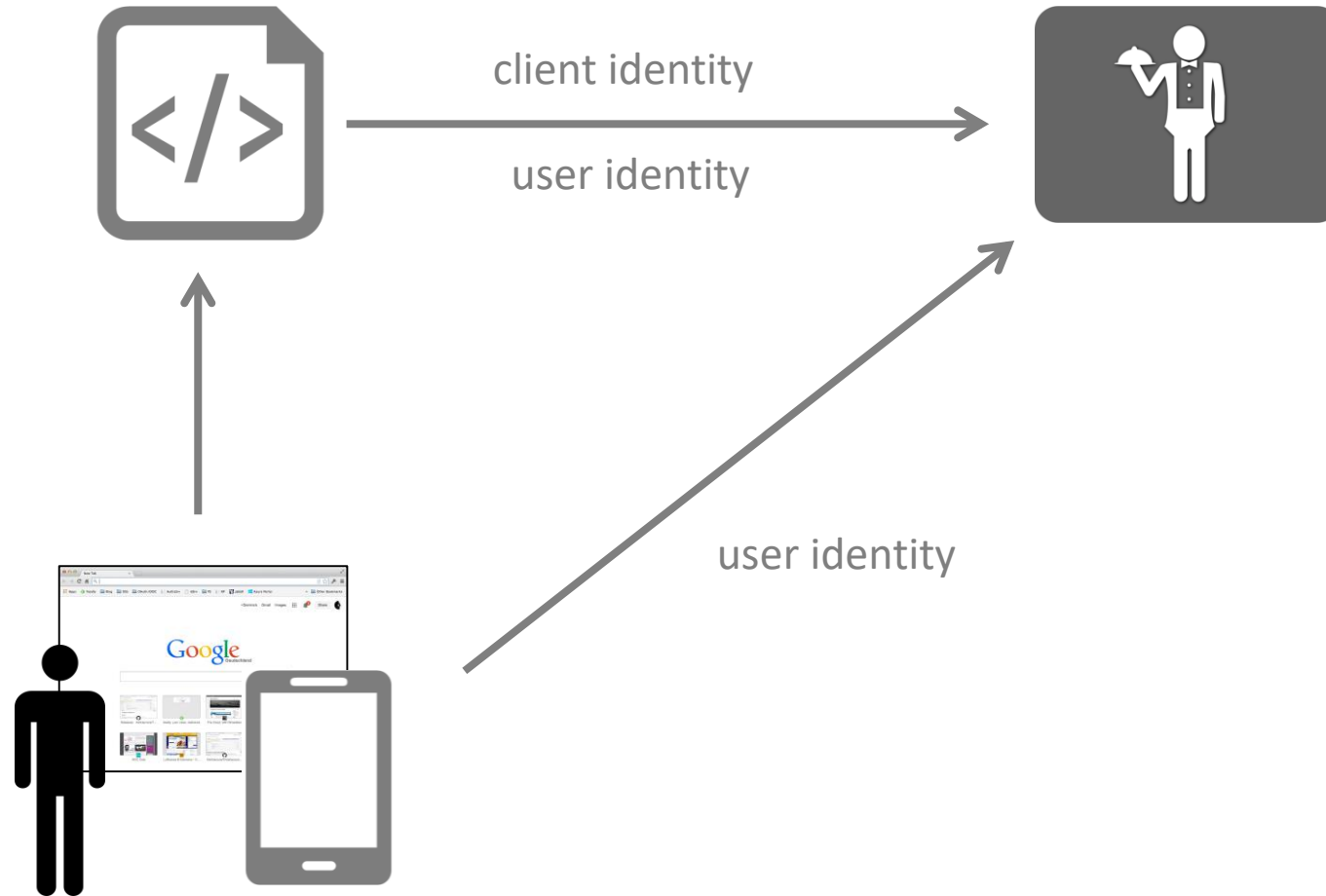
```
{
  "issuer": "https://demo.identityserver.io",
  "jwks_uri": "https://demo.identityserver.io/.well-known/openid-configuration/jwks",
  "authorization_endpoint": "https://demo.identityserver.io/connect/authorize",
  "token_endpoint": "https://demo.identityserver.io/connect/token",
  "userinfo_endpoint": "https://demo.identityserver.io/connect/userinfo",
  "end_session_endpoint": "https://demo.identityserver.io/connect/endsession",
  "check_session_iframe": "https://demo.identityserver.io/connect/checksession",
  "revocation_endpoint": "https://demo.identityserver.io/connect/revocation",
  "introspection_endpoint": "https://demo.identityserver.io/connect/introspect",
  "frontchannel_logout_supported": true,
  "frontchannel_logout_session_supported": true,
  "backchannel_logout_supported": true,
  "backchannel_logout_session_supported": true,
  "scopes_supported": [
    "openid",
    "profile",
    "email",
    "api",
    "offline_access"
  ],
  "claims_supported": [
    "sub",
    "name",
```

Connecting an MVC Client

```
services.AddAuthentication("Cookies")
    .AddCookie("Cookies", options =>
    {
        options.LoginPath = "/account/login";
        options.AccessDeniedPath = "/account/denied";
    })
    .AddOpenIdConnect("oidc", options =>
    {
        options.Authority = "https://demo.identityserver.io";
        options.ClientId = "mvc";

        options.TokenValidationParameters = new TokenValidationParameters
        {
            NameClaimType = "name",
            RoleClaimType = "role"
        };
    });
```

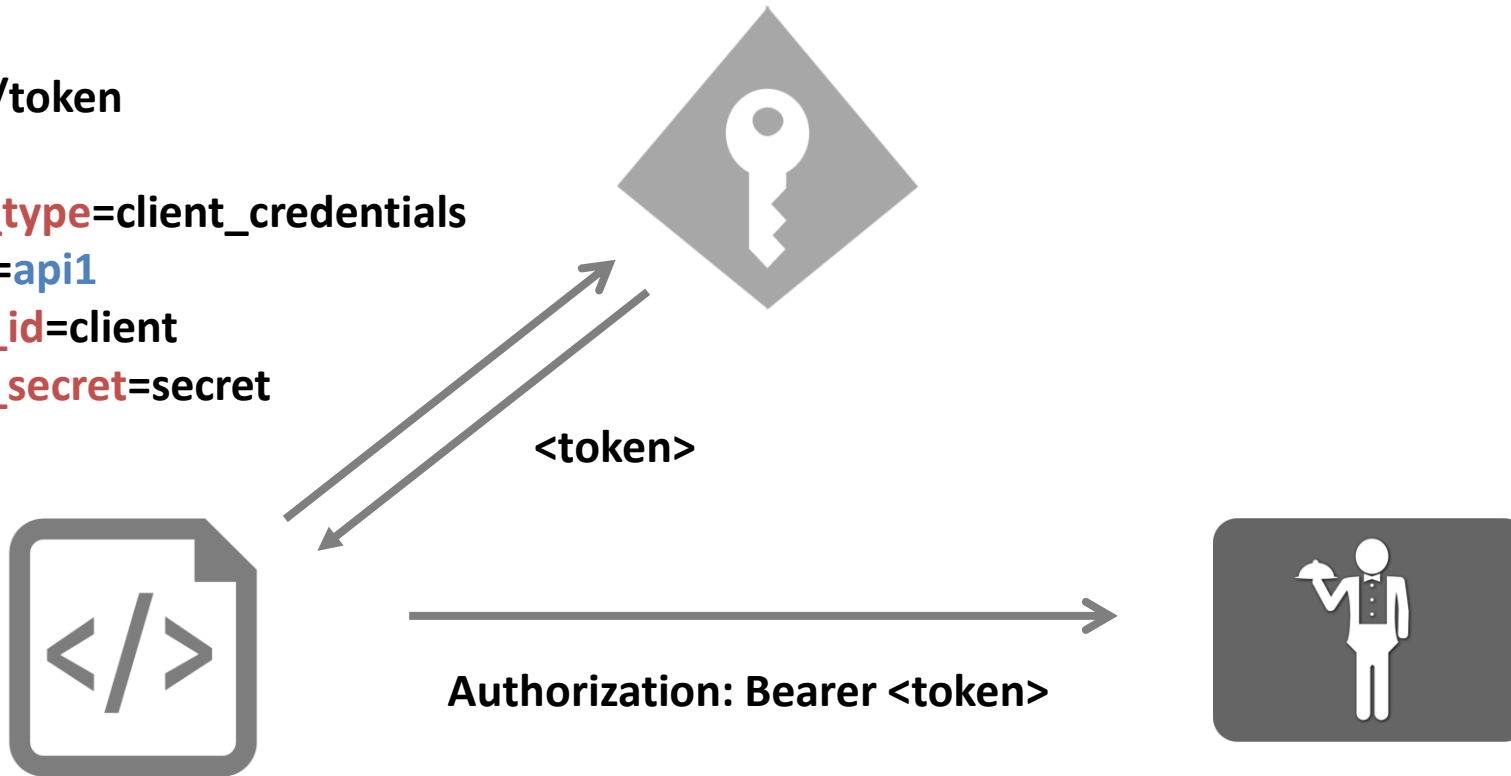
Accessing APIs



Calling an API using Client Identity

POST /token

grant_type=client_credentials
scope=api1
client_id=client
client_secret=secret



Access Token Validation

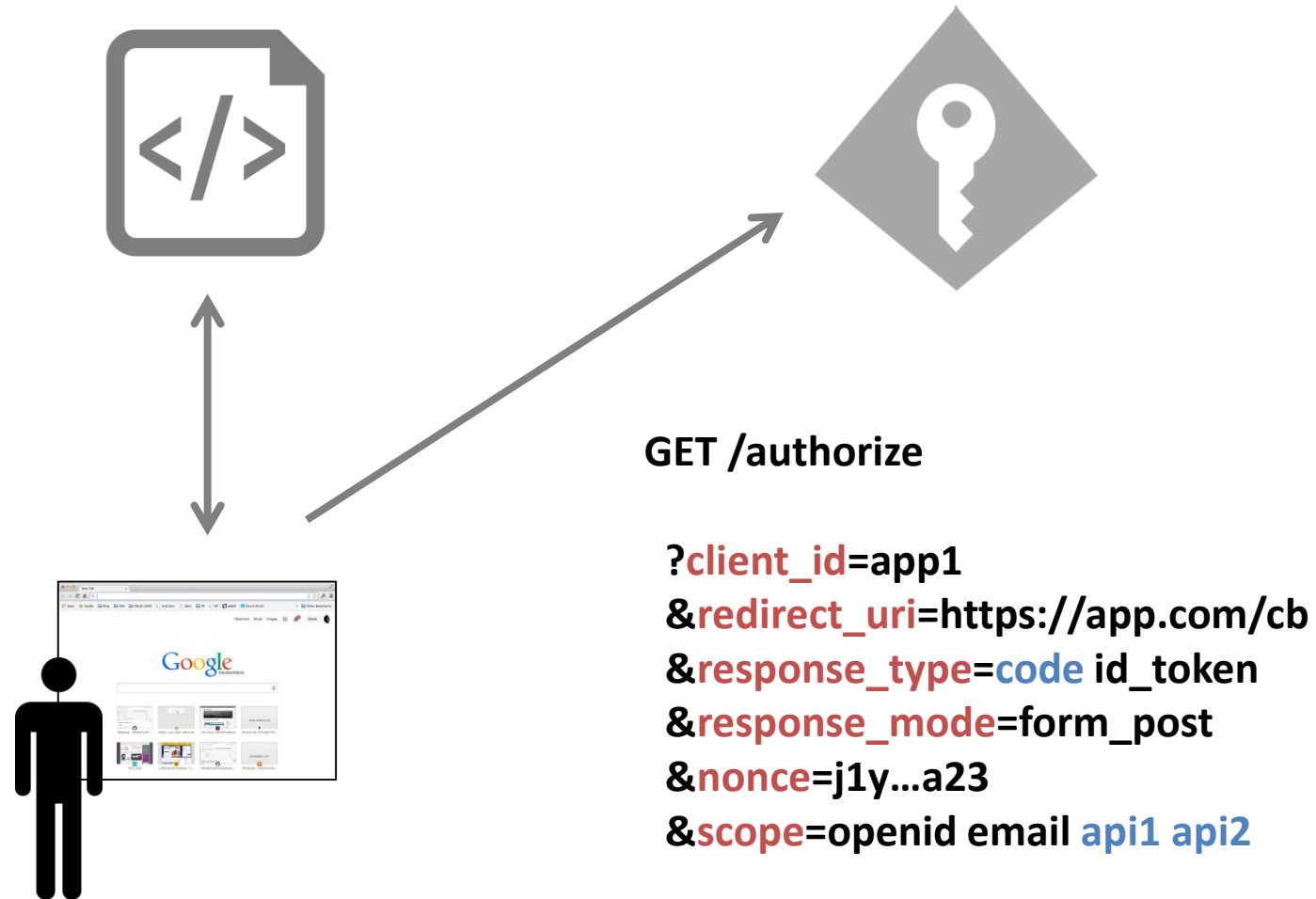
- **JWT bearer token authentication handler**

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddAuthentication("Bearer")
        .AddJwtBearer("Bearer", options =>
        {
            options.Authority = "https://demo.identityserver.io";
            options.Audience = "api1";
        });
}
```

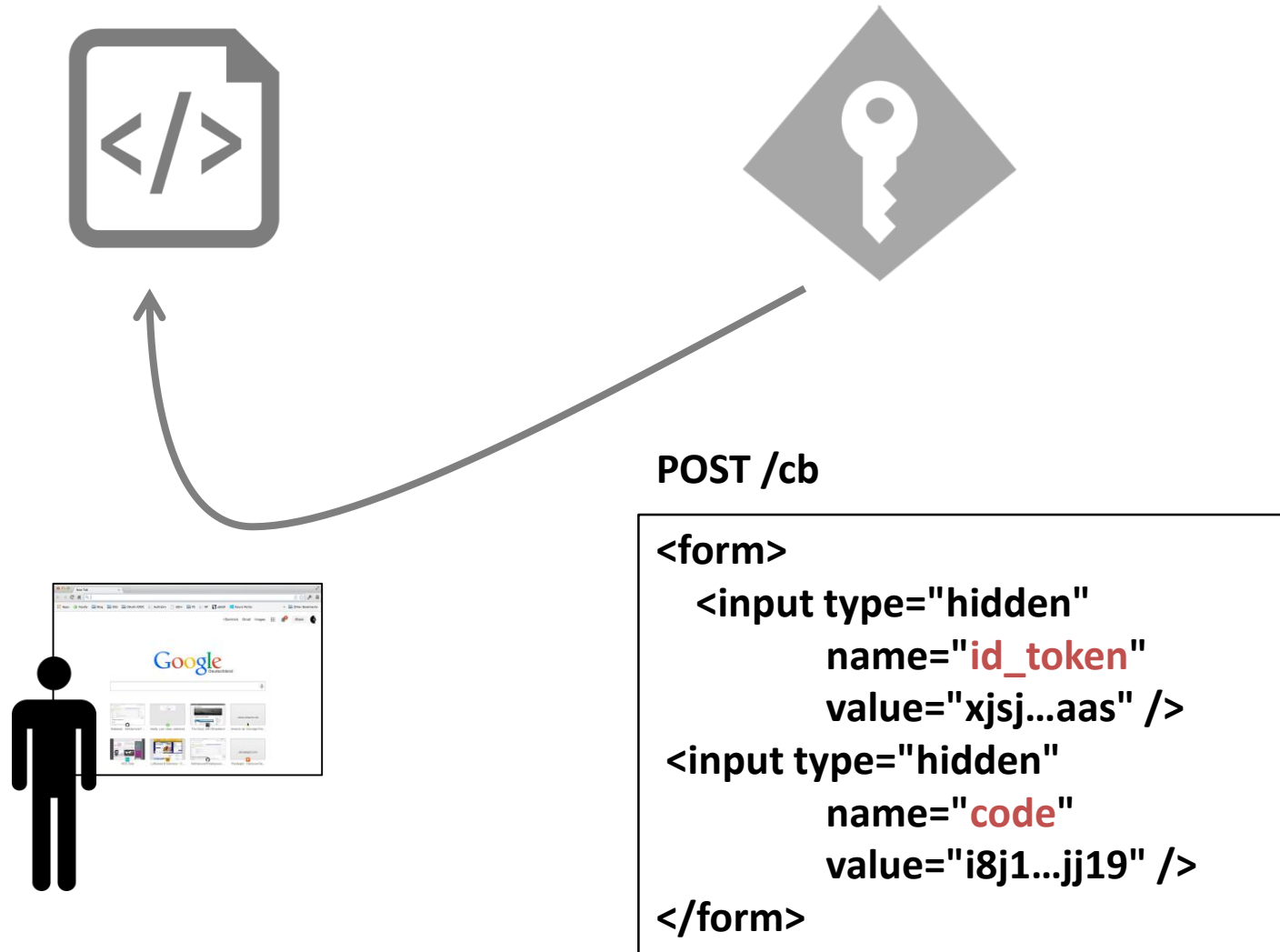
Web Applications using APIs

- **OpenID Connect Hybrid Flow combines**
 - user authentication (identity token)
 - access to APIs on behalf of user (access token)
- **Additional Security Features**
 - access tokens not exposed to the browser
 - (optional) long-lived API access

Hybrid Flow Request

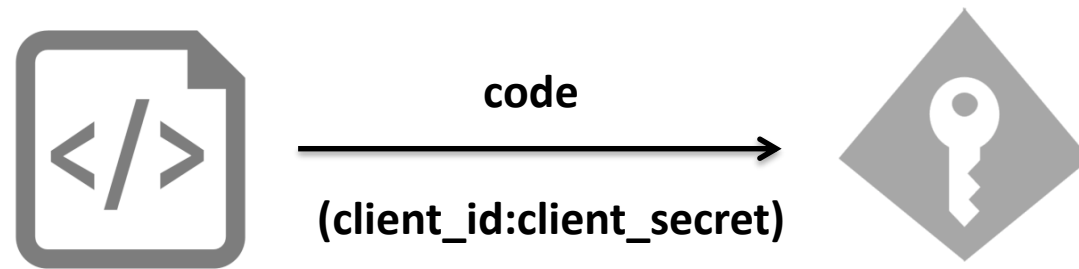


Hybrid Flow Response



Retrieving the Access Token

- **Exchange code for access token**
 - using client id and secret

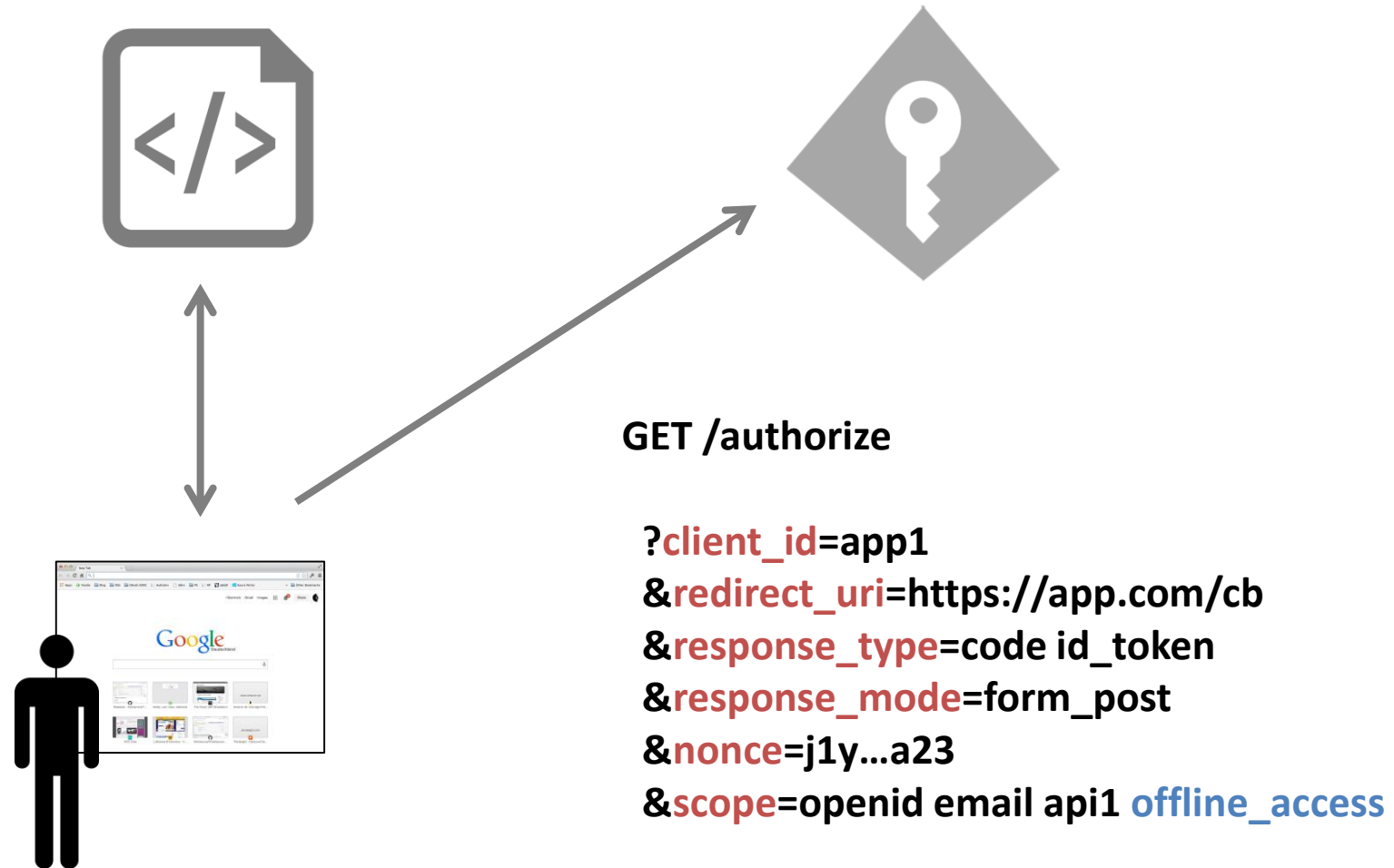


```
{  
  access_token: "xyz...123",  
  expires_in: 3600,  
  token_type: "Bearer"  
}
```

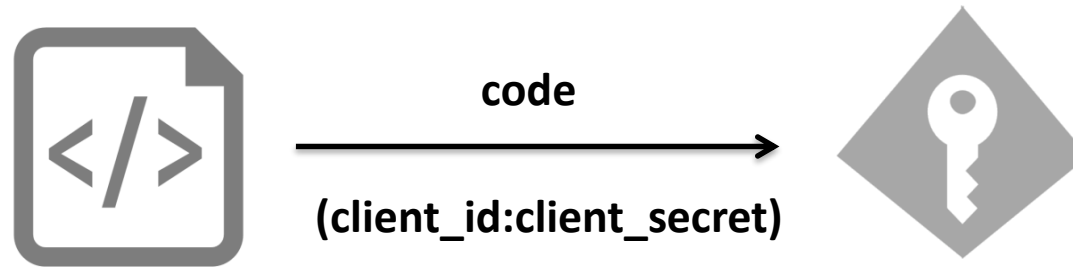
Access Token Lifetime Management

- **Access tokens have finite lifetimes**
 - requesting a new token requires browser round trip to authorization server
 - should be as short lived as possible
- **Refresh tokens allow renewal semantics**
 - no user interaction required
 - typically combined with a revocation feature

Requesting a Refresh Token

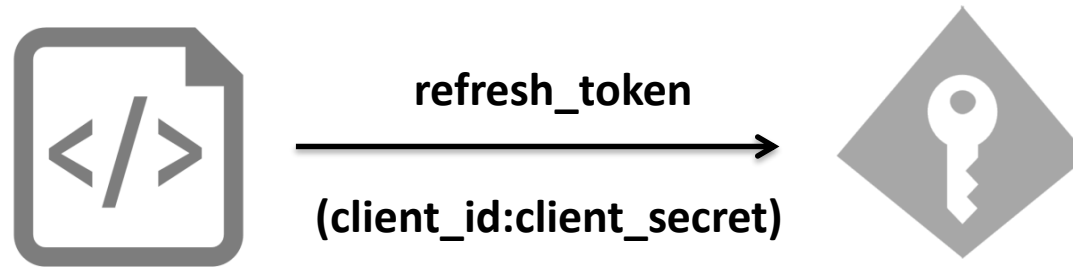


Retrieving the Access Token (w/ Refresh Token)



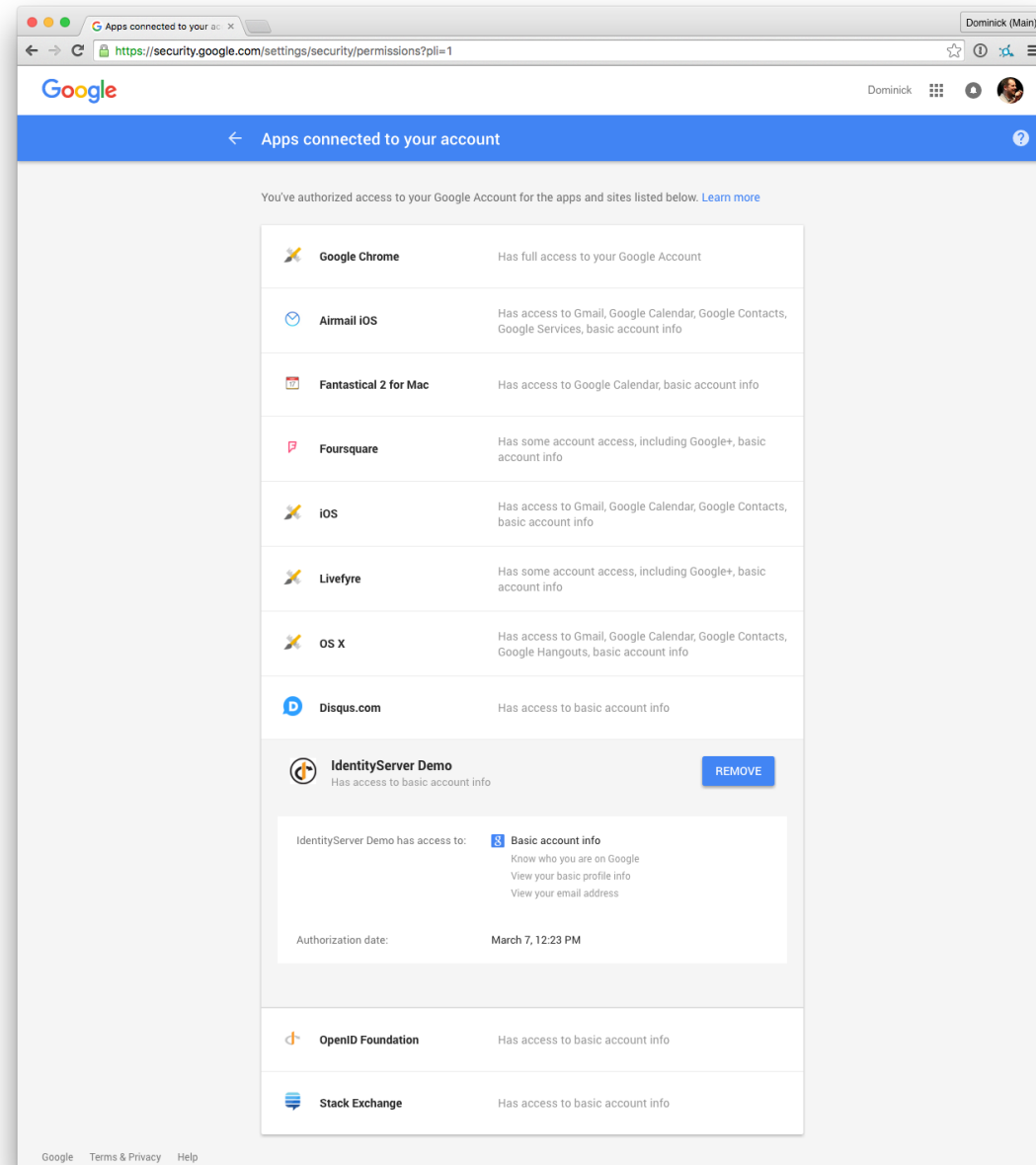
```
{  
  access_token: "xyz...123",  
  refresh_token: "jdj9...192j",  
  expires_in: 3600,  
  token_type: "Bearer"  
}
```

Refreshing an Access Token



```
{  
  access_token: "xyz...123",  
  refresh_token: "jdj9...192j",  
  expires_in: 3600,  
  token_type: "Bearer"  
}
```

Revocation



Token Revocation

- **Endpoint to programmatically revoke tokens (RFC 7009)**
 - reference tokens
 - refresh tokens



`/revoke?token=a19..18a`



Summary

- **ASP.NET Core is Microsoft's latest web platform**
 - implements (amongst other things) authentication, authorization, CORS...
- **Authentication handlers implement different authentication schemes**
- **OpenID Connect is for authentication and session management**
- **OAuth 2.0 is for (delegated) access to APIs**