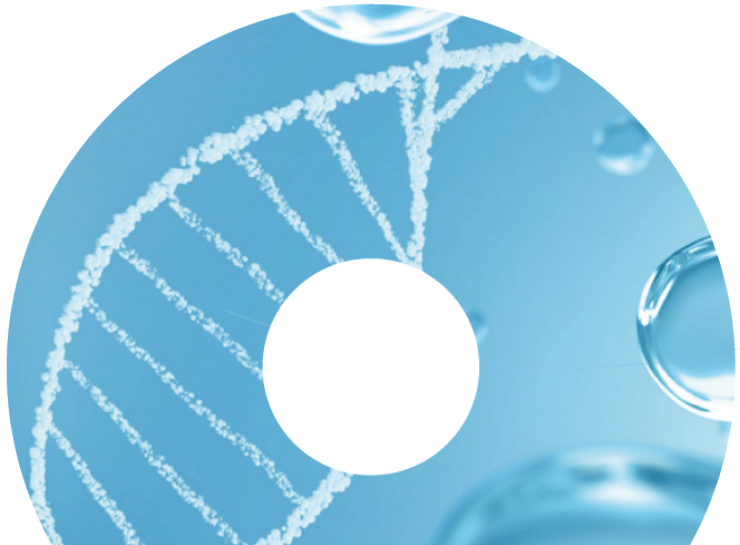
Evolución de un sistema de Autenticación y Autorización

ANGULAR + OAUTH + INTERSYSTEMS IRIS

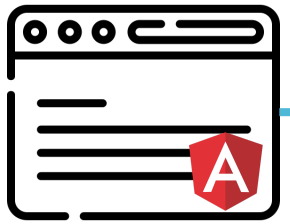
Lluís Vilalta Mallofré | CERBA

InterSystems Developer Community

cerba Internacional

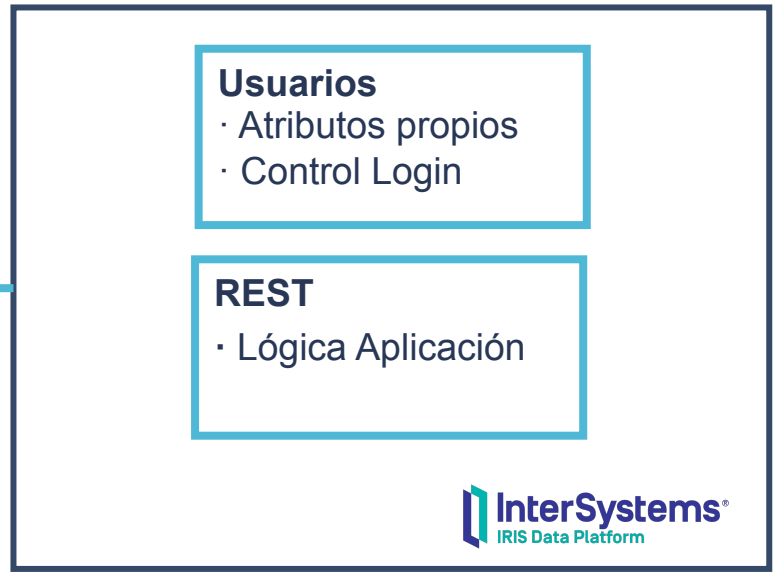


01| ESCENARIO INICIAL |



UI

- Angular8 (versión antigua)
- Logic Autenticación Basic



ACCESO SEGURO

- Sistema actual pensado para consumo interno o conexiones VPN.
- ¿Cómo permitir **acceso seguro** y controlado también desde fuera?

MANTENIMIENTO

- ¿Se puede cambiar el sistema **manteniendo la implementación actual** de servicios REST?

ESTÁNDAR

- ¿Es posible usar un **estándar**, pero manteniendo **particularidades** de control de usuarios?
- ¿El estándar puede utilizarse también en **Angular**?

02| CONCEPTOS|

ROLES

- **Servidor de Recursos** – servidor que tiene recursos protegidos (e.g. servicio REST)
- **Cliente** – aplicación que solicita acceso (limitado) a recursos protegidos (e.g. Angular)
- **Servidor de Autorización** – servidor responsable de generar tokens de acceso, con los que los clientes pueden acceder al servidor de recursos.

GRANT TYPES / FLUJOS DE TRABAJO

- **Tipos de acceso** (o flujos diferentes con los que OAuth puede trabajar)
- Hay tipos diferentes para cada caso (Client Credentials (conexiones de aplicación a aplicación , Authorization Code (con interacción de usuarios), etc.)

SCOPES

- Mecanismos para **limitar** el acceso. Ámbitos/Permisos a los que se quiere acceder.

03| OAuth|

InterSystems Developer Community

Encoded PASTE A TOKEN HERE


```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiIsImtpZCI6IjMifQ.eyJqdGkiOiJodHRwczovL3dlYnNlcnZlci9hdXRoc2VydmVyL29hdXRoMi5vd2V6U3pCMDRxSnVCam5KMVF4OHptcGJYRWMiLCJpc3MiOiJodHRwczovL3dlYnNlcnZlci9hdXRoc2VydmVyL29hdXRoMiIsInN1YiI6IkhkNlg2QU9KT1FZcU96WFhKY3pWQlZ2ZzdSTFBoN09ndGZaYXF0MnJfdGsiLCJleHAiOjE2OTMyMTA0MTMsImF1ZCI6IkhkNlg2QU9KT1FZcU96WFhKY3pWQlZ2ZzdSTFBoN09ndGZaYXF0MnJfdGsiLCJzY29wZSI6Im15L3Njb3BlIn0.d-3lV6MA9ot2fluzz6tVuwemdHGKa7-ZUh_tg2ZwF7OJWKxyWhH97H7aJDqwokJPOIyb7mSfET9m3cz9yAz6ANBfLbF8qa0O8dVaf24Ot3ls3fJU1rUe24EgkwpILyIdABfNIb4bsN-CUxoqCRaTkUH0menQBflCK4rDF1V0BqlzDZXJi_dQnW5tazgV-Y2fzab-LfdKunVHuKBElTB87FpH2jkDaIlNOk2KgUxcX54F5t08HmTf973NeoZ9dslXjJh1ULWQQyQ-JEMkwVDlhqApXiCtYo5EKd7OrILwvNrngmpxc0wu8EQ9rd_R61nQORgMr1WuMTnnrD3yMwPRVQ
```


Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

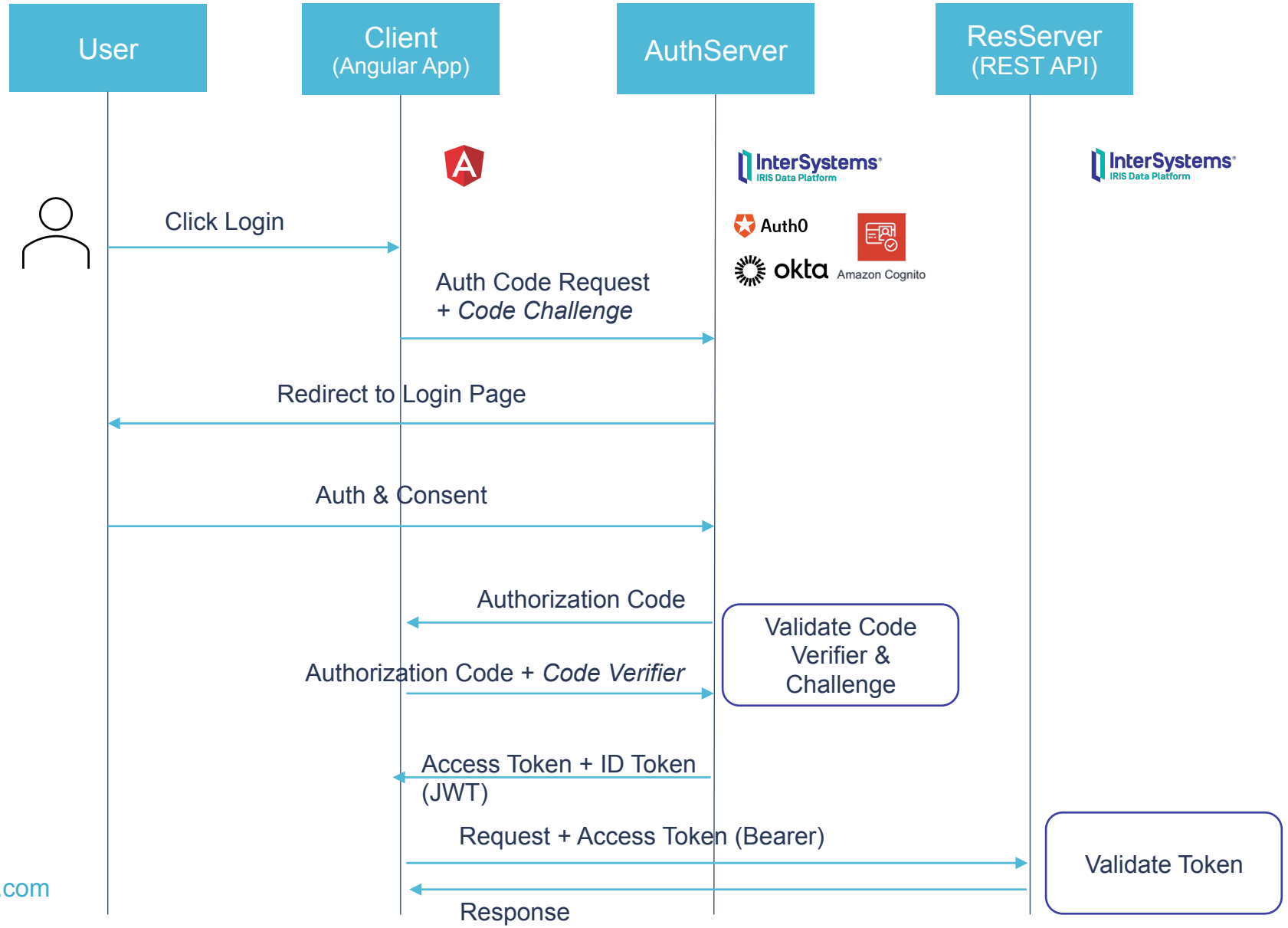

```
{
  "typ": "JWT",
  "alg": "RS512",
  "kid": "3"
}
```


PAYLOAD: DATA

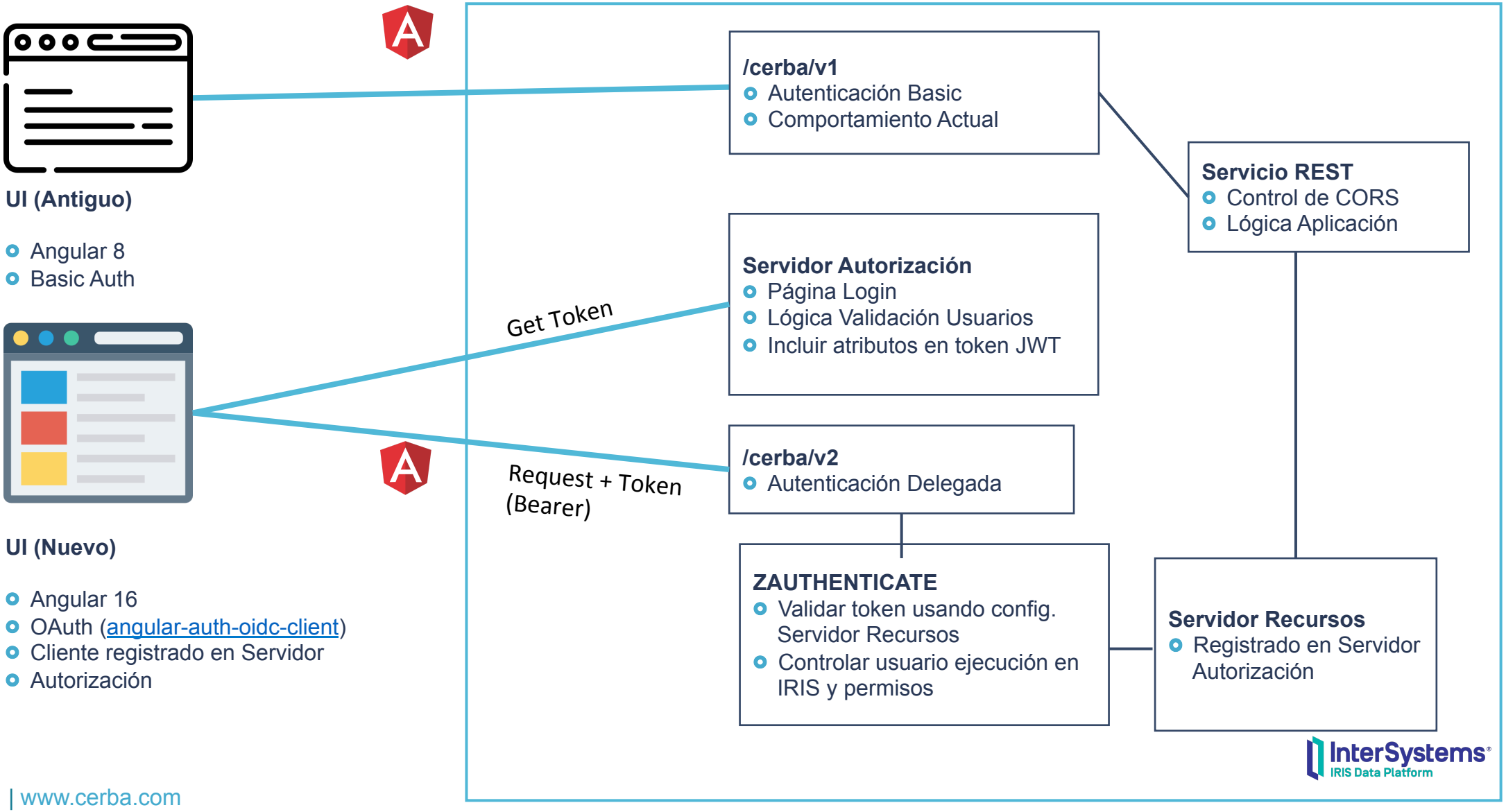

```
{
  "jti":
"https://webserver/authserver/oauth2.owezSzB04qJuBjnJ1Qx8zmpbXEc",
  "iss": "https://webserver/authserver/oauth2",
  "sub": "Hd6X6AOJOQYqOzXXJczVBVvg7RLPh7OgtfZaqt2r_tk",
  "exp": 1693210413,
  "aud": "Hd6X6AOJOQYqOzXXJczVBVvg7RLPh7OgtfZaqt2r_tk",
  "scope": "my/scope"
}
```


- Servidor Autorización puede añadir más atributos.

04| Authorization Code + PKCE|



05|Escenario Final|



06|angular-auth-oidc-client|

Servidor Autorización (IRIS)

Redirección (después login)

Client Id (Angular App)

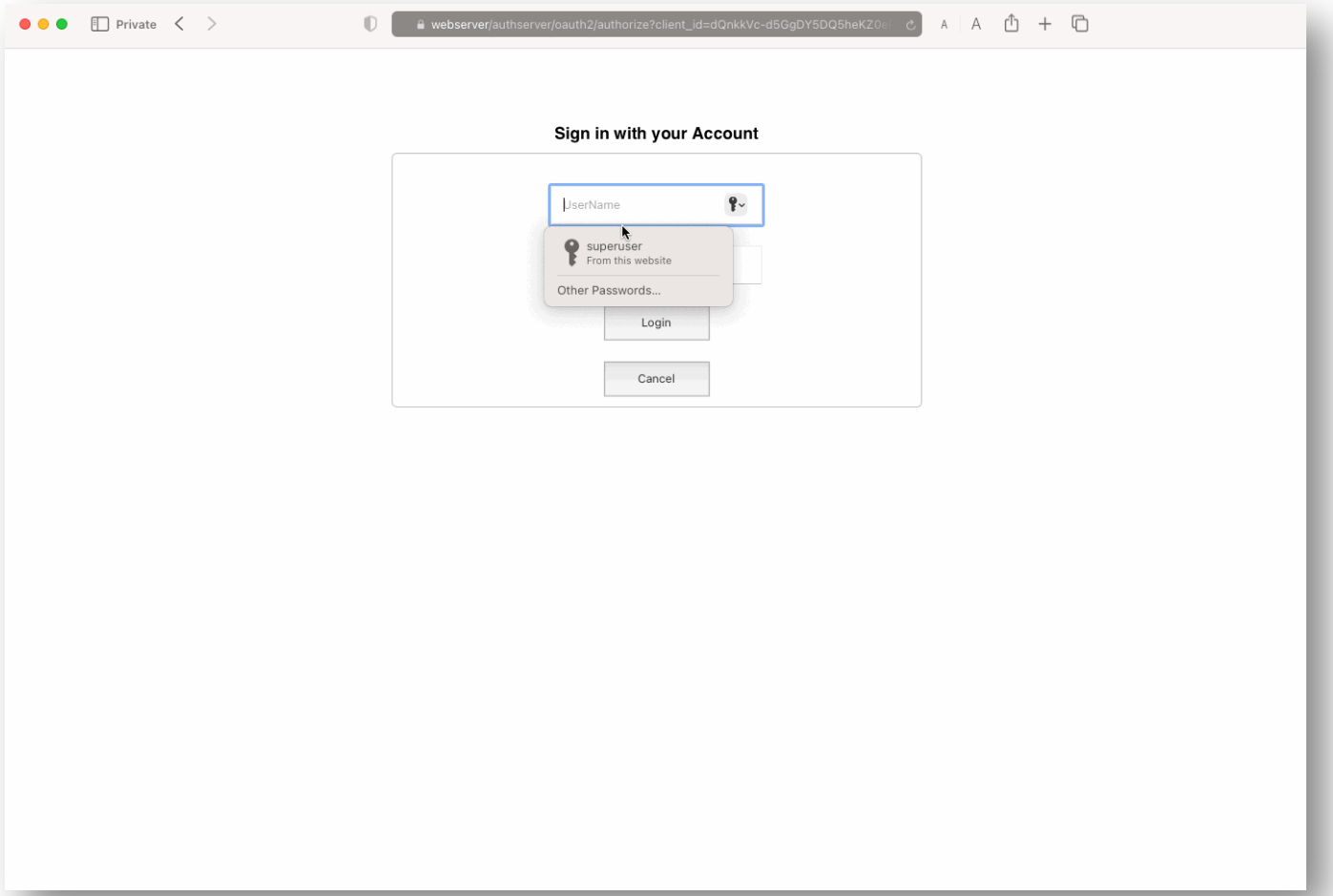
```
@NgModule({
    imports: [AuthModule.forRoot({
        config: {
            authority: 'https://webserver/authserver/oauth2/.well-known/openid-configuration',
            redirectUrl: environment.urlRedirect,
            postLogoutRedirectUri: environment.urlRedirect,
            clientId: 'dQnkkVc-d5GgDY5DQ5heKZ0ePbBlDTE_wGQcNh7KNzM',
            scope: 'openid profile email offline_access my/scope', // 'openid profile ' + your scopes
            responseType: 'code',
            logLevel: LogLevel.Debug,
            renewTimeBeforeTokenExpiresInSeconds: 10,
            secureRoutes: ['https://webserver/resserver/protected-resources/', 'https://webserver/resserver/protected-resources-delegated/'],

            /** token refresh */
            silentRenew: true,
            useRefreshToken: true,
            ignoreNonceAfterRefresh: true, // this is required if the id_token is not returned
            triggerRefreshWhenIdTokenExpired: false, // required when refreshing the browser if id_token is not updated after the first authentication
        }
      })],
    exports: [AuthModule],
})
```

Refresco token automático

Rutas Protegidas (envía el Token como Bearer automáticamente)

WORKSHOP-IRIS-OAUTH2

- https://openexchange.intersystems.com/package/workshop-iris-oauth2
- Configuración en IRIS como: cliente, servidor autorización, servidor recursos
- Client Credentials, Authorization Code, Authorization Code + PKCE
- Ejemplo Angular



¿Preguntas?

MUCHAS GRACIAS