Advanced Topics

Kevin Dockx

@KevinDockx | http://blog.kevindockx.com/

Token Expiration

Identity Tokens

- Sign in to the client
- Short lifetime
- Client implements its own expiration policy

Access Tokens

- Access resources (API)
- Longer lifetime
- Expiration policy controlled by the security token service

- **Requires the client to handle expired tokens**

Controlling Token Expiration

Learn how to control when a token expires

Refresh Tokens

A refresh token is used to achieve long-lived access
(avoiding redirection to the security token service)

Client requires offline_access scope & a flow that supports it
Supported by authorization code, hybrid and resource owner password credentials flows

Handling Expired Tokens with Refresh Tokens (MVC)

Learn how to work with refresh tokens for long-lived access

Handling Expired Tokens with Redirection (Angular)

Learn how to handle an expired token by redirecting to the security token service

Redirection vs Refresh Tokens

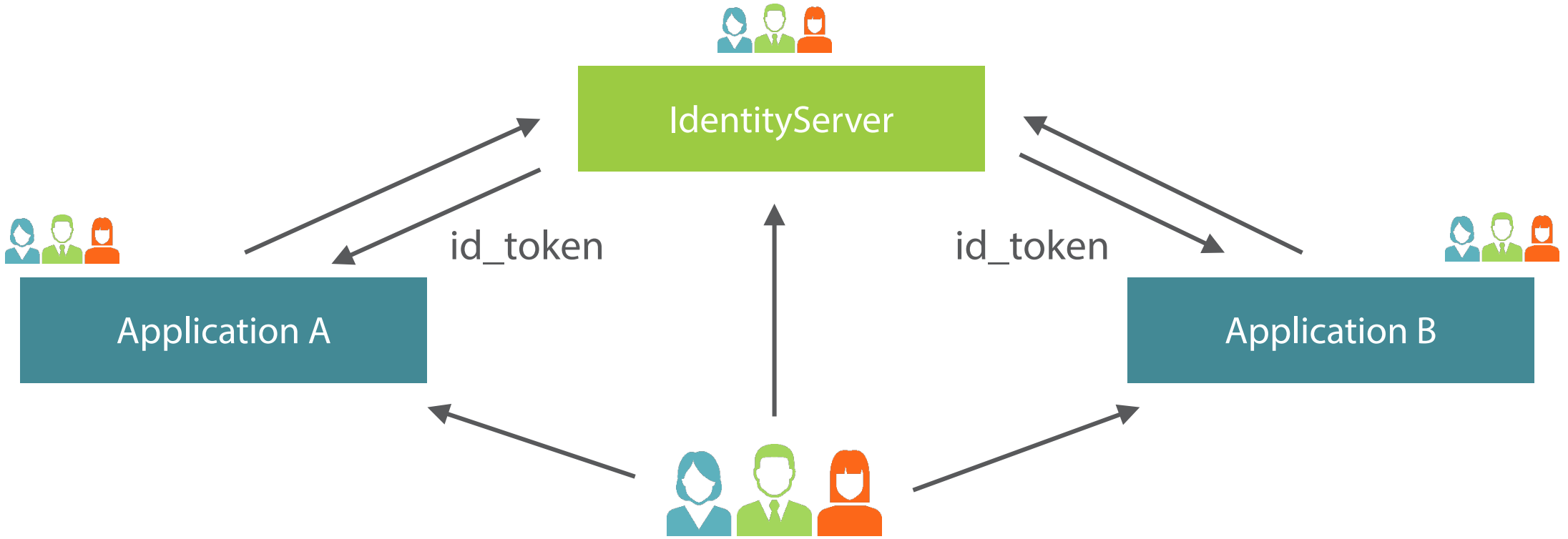
Redirection

- As long as we are signed in to the security token service, we don't have to provide our credentials
- Redirection is required
- Short lifetime

Refresh Tokens

- No need to be signed in to the security token service
- No redirection, happens in the background
- Much longer lifetime

Single Sign-on



"Given multiple applications, using the same set of credentials, a user should only provide these credentials once"

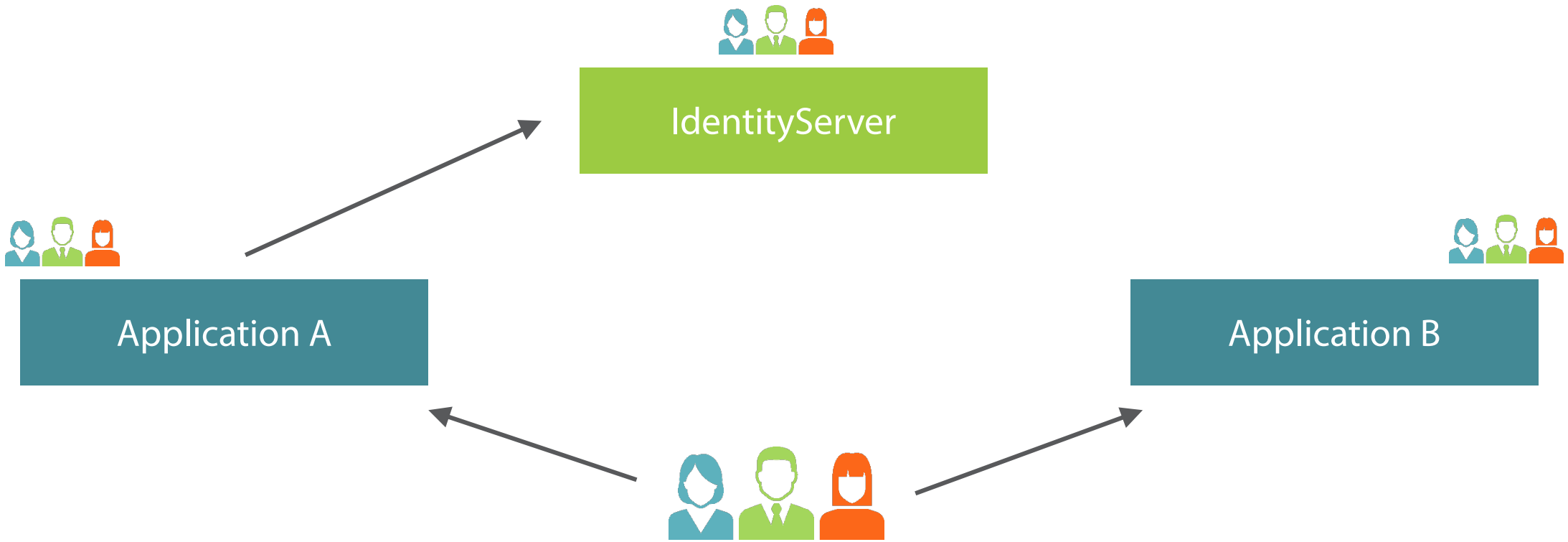
Single Sign-on Between MVC and Angular

Learn how to achieve single sign-on between our ASP.NET MVC and Angular clients

Redirecting to a Specific URI After Logging Out

Learn how to redirect to a specific URI after logging out of an ASP.NET MVC or Angular client

Single Sign-out



Related Specifications

OpenID Connect Session Management

Describes how to manage sessions for OpenID Connect, including when to log out the user

http://openid.net/specs/openid-connect-session-1_0.html

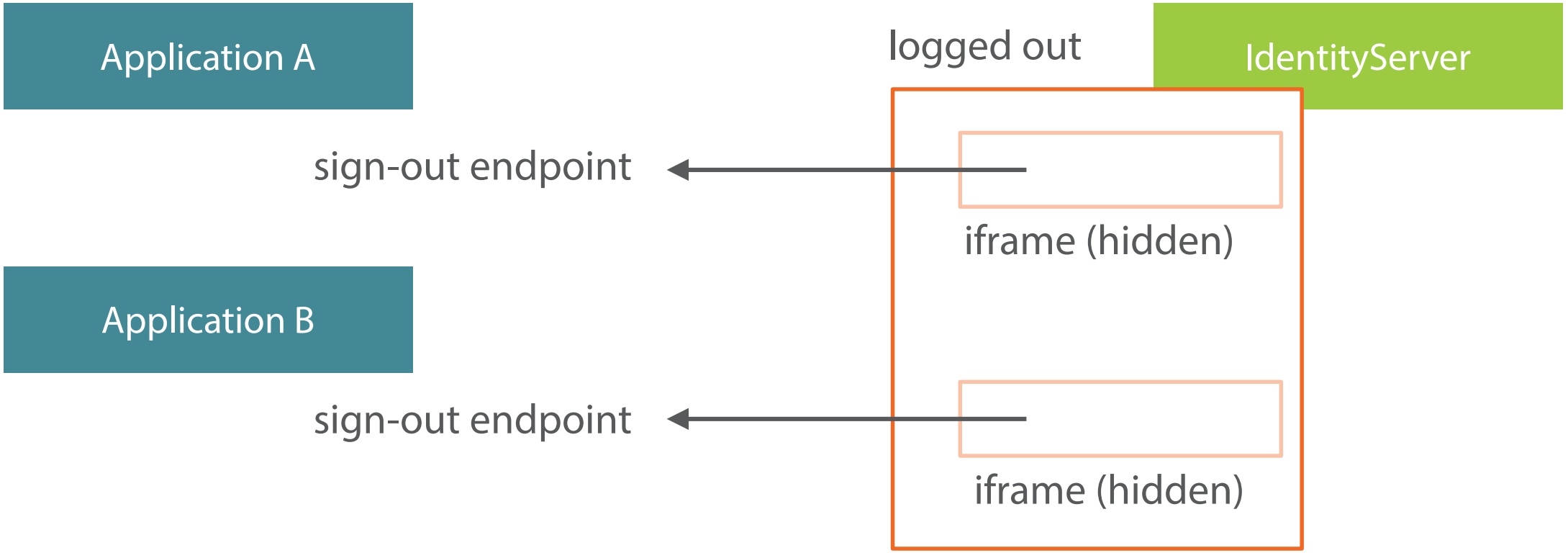
OpenID Connect HTTP-Based Logout

Defines an HTTP-based logout mechanism that does not need an OpenID provider iframe on relying party pages

http://openid.net/specs/openid-connect-logout-1_0.html

Early draft status…

Implementing Single Sign-out



Single Sign-out Between MVC and Angular

Learn how to achieve single sign-out between our ASP.NET MVC and Angular clients

Avoiding Visible Redirection in Angular with a Hidden Iframe

Client periodically checks validity of the access token

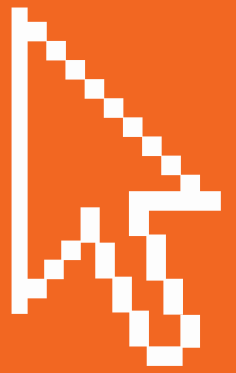
A token request is sent to the authorization endpoint, in a hidden iframe

Token(s) is/are delivered to a page in a hidden iframe

oidc_token_manager implements this functionality

Avoiding Visible Redirection in Angular with a Hidden Iframe

Learn how to request new tokens for an Angular client with a hidden iframe



Summary

	MVC 	Angular 
Sign in to the client	▪ Hybrid flow code id_token	▪ Implicit flow id_token
Access the API	▪ Hybrid flow code id_token token	▪ Implicit flow id_token token
Handle expired tokens	▪ Refresh token offline_access scope	▪ Authorization endpoint hidden iframe