Angular in Practice: Authentication and Authorization

The Importance of Using a BFF

Roland Guijt

Freelance consultant and trainer

@rolandguijt | roland.guijt@gmail.com

Assumed Knowledge

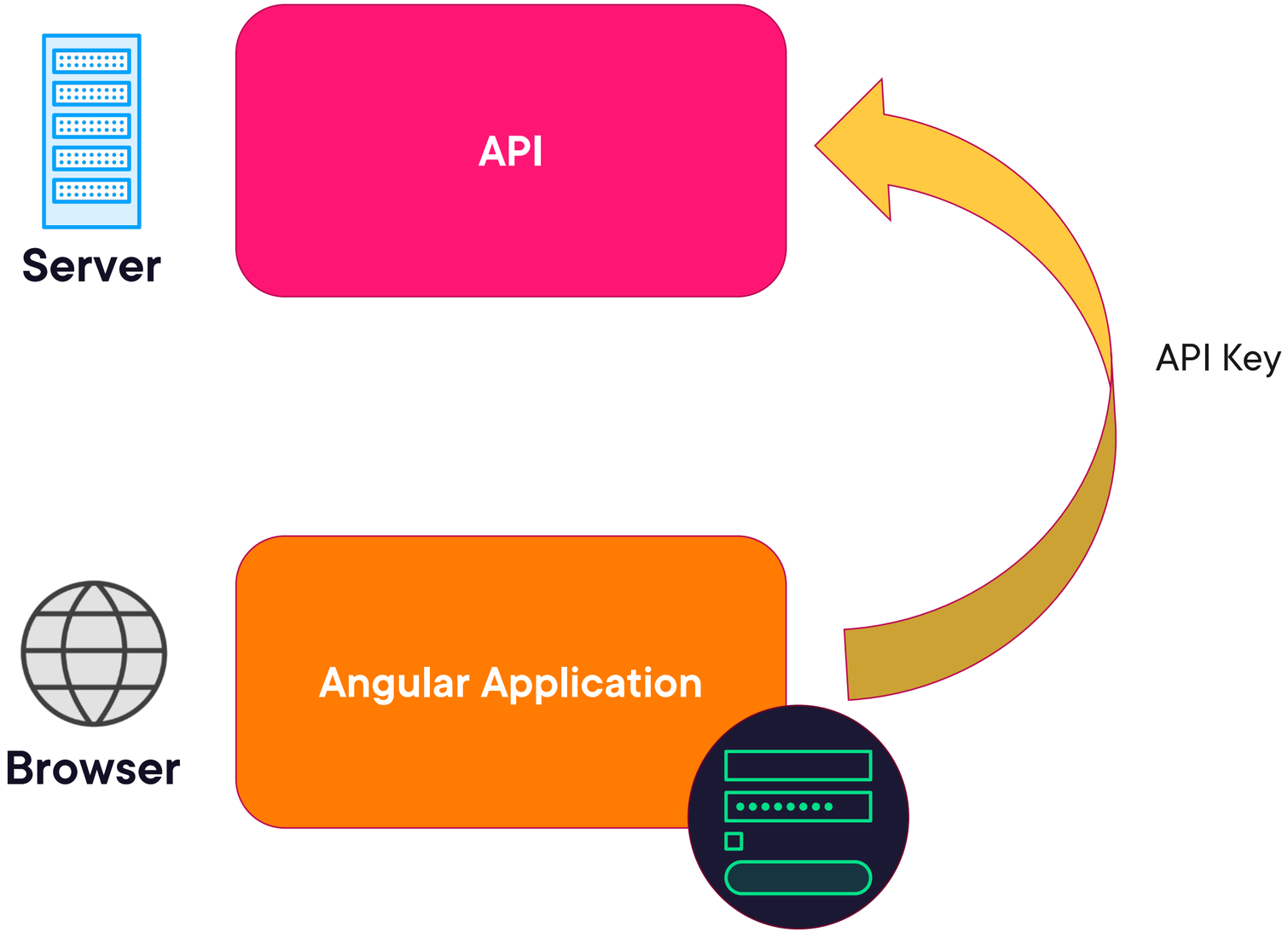
Angular

OAuth and OpenID Connect

Authentication and Authorization in Angular

https://github.com/RolandGuijt/ps-angularauthprac

Anti-pattern #1: Authentication by SPA



Problems

The browser can't be trusted with secrets

Browser context easy to read

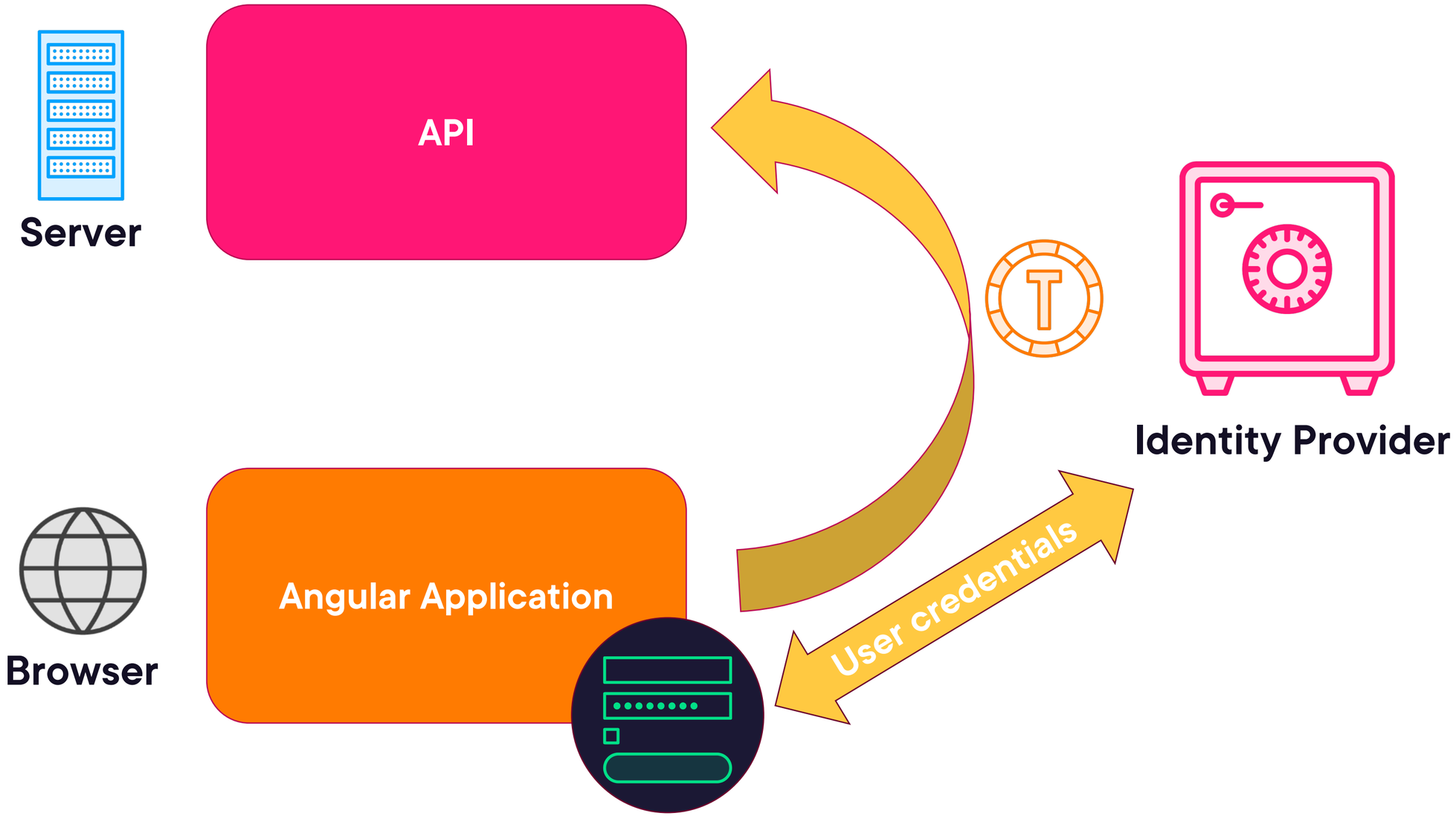
Browser susceptible to attacks

Encryption doesn't work

API key is static

API has no user context

Anti-pattern #2: Resource Owner Password Credentials



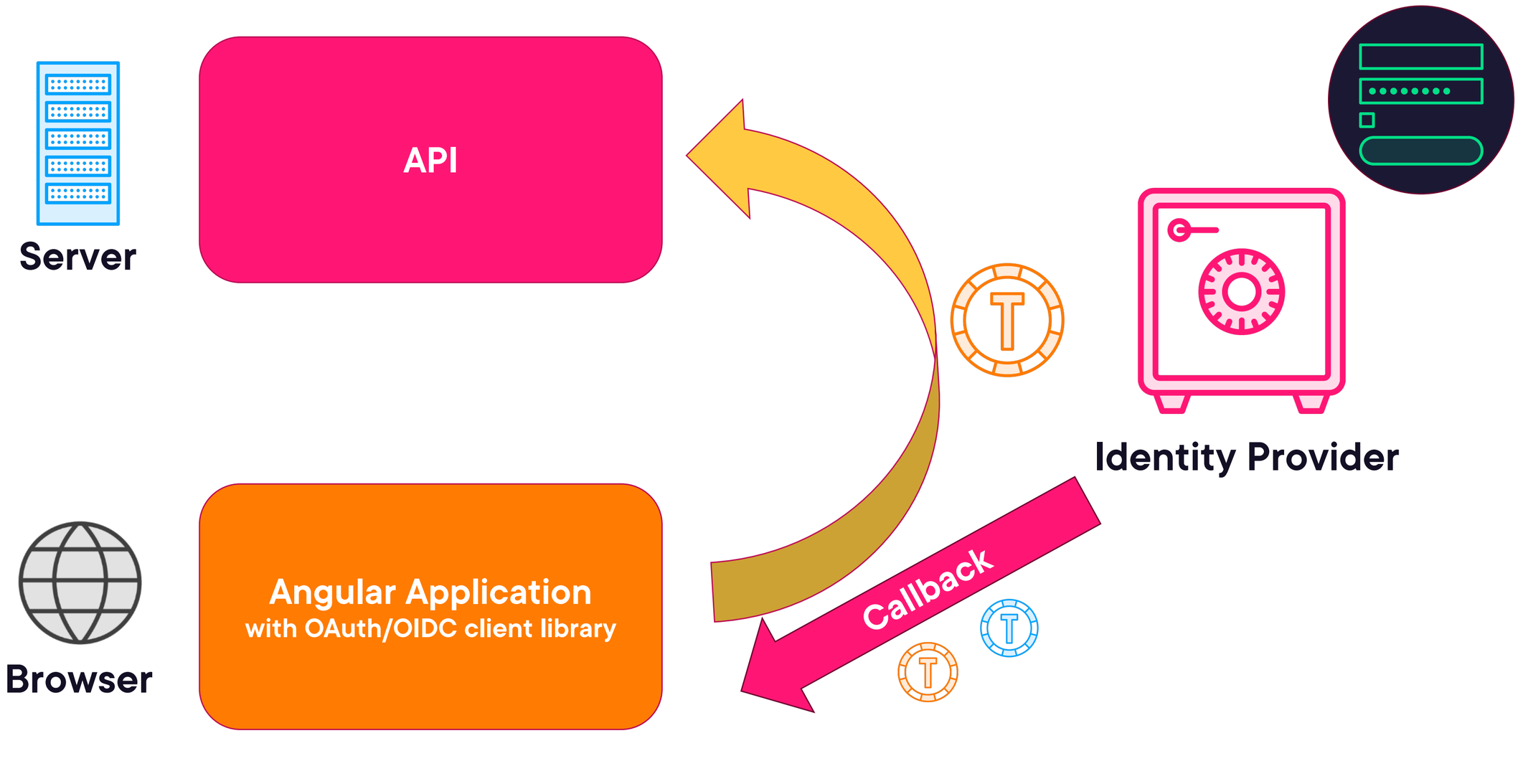
Problems

Access token: secret the browser can't keep

User secret exposed to browser

Limits the flexibility of authentication

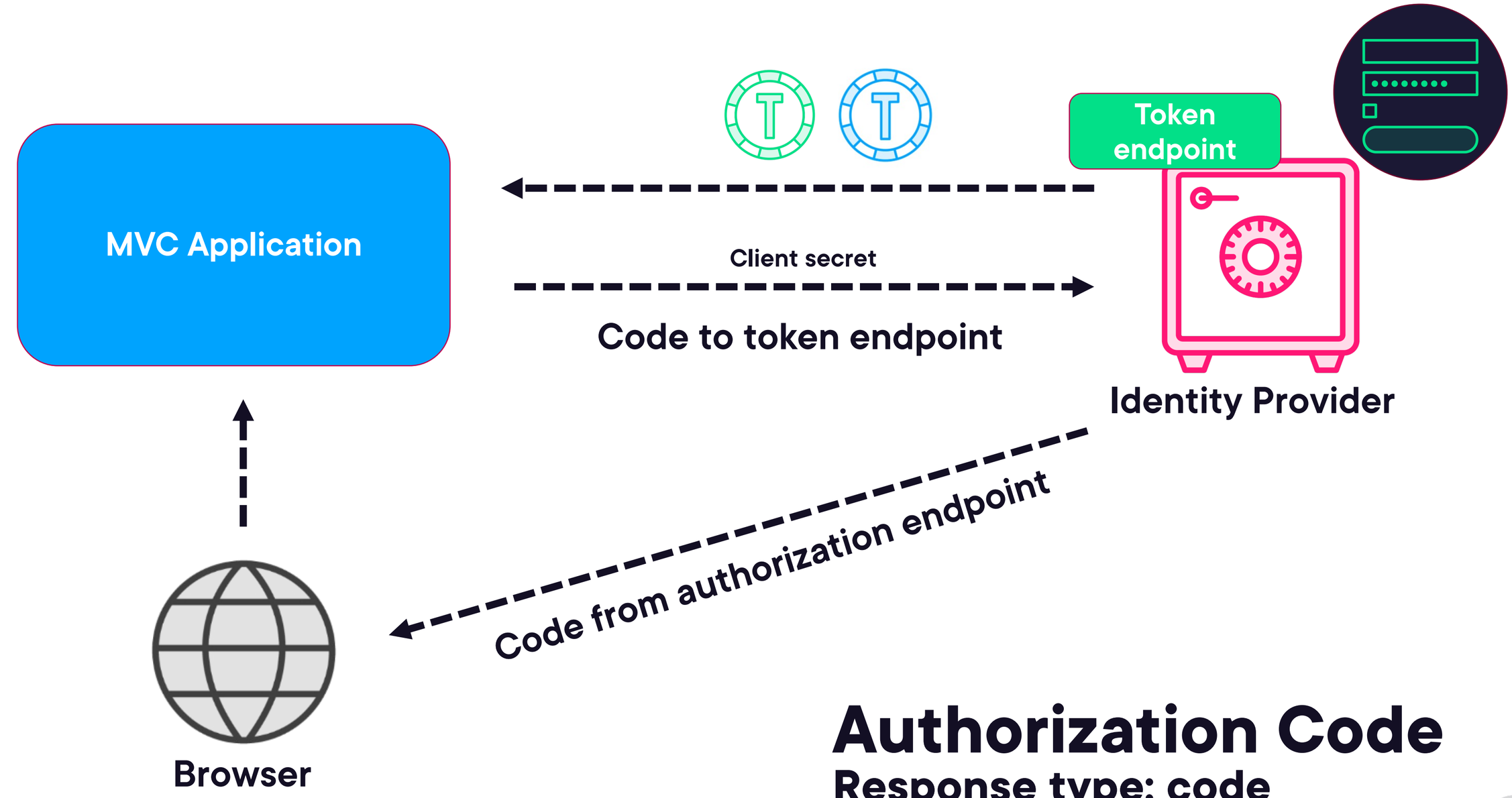
Anti-pattern #3: Implicit/Authorization Code Flow



Problems

Access token: secret the browser can't keep

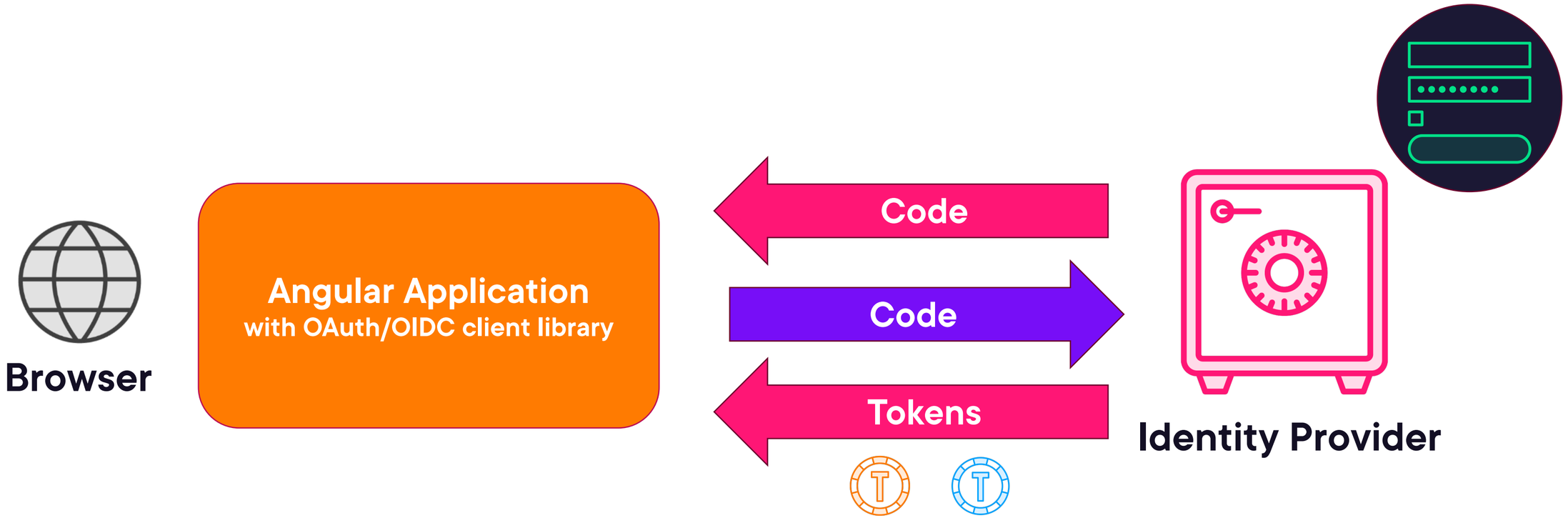
If refresh tokens are used problem gets worse



Authorization Code

Response type: code
Scope: openid

Anti-pattern #3: Implicit/Authorization Code Flow



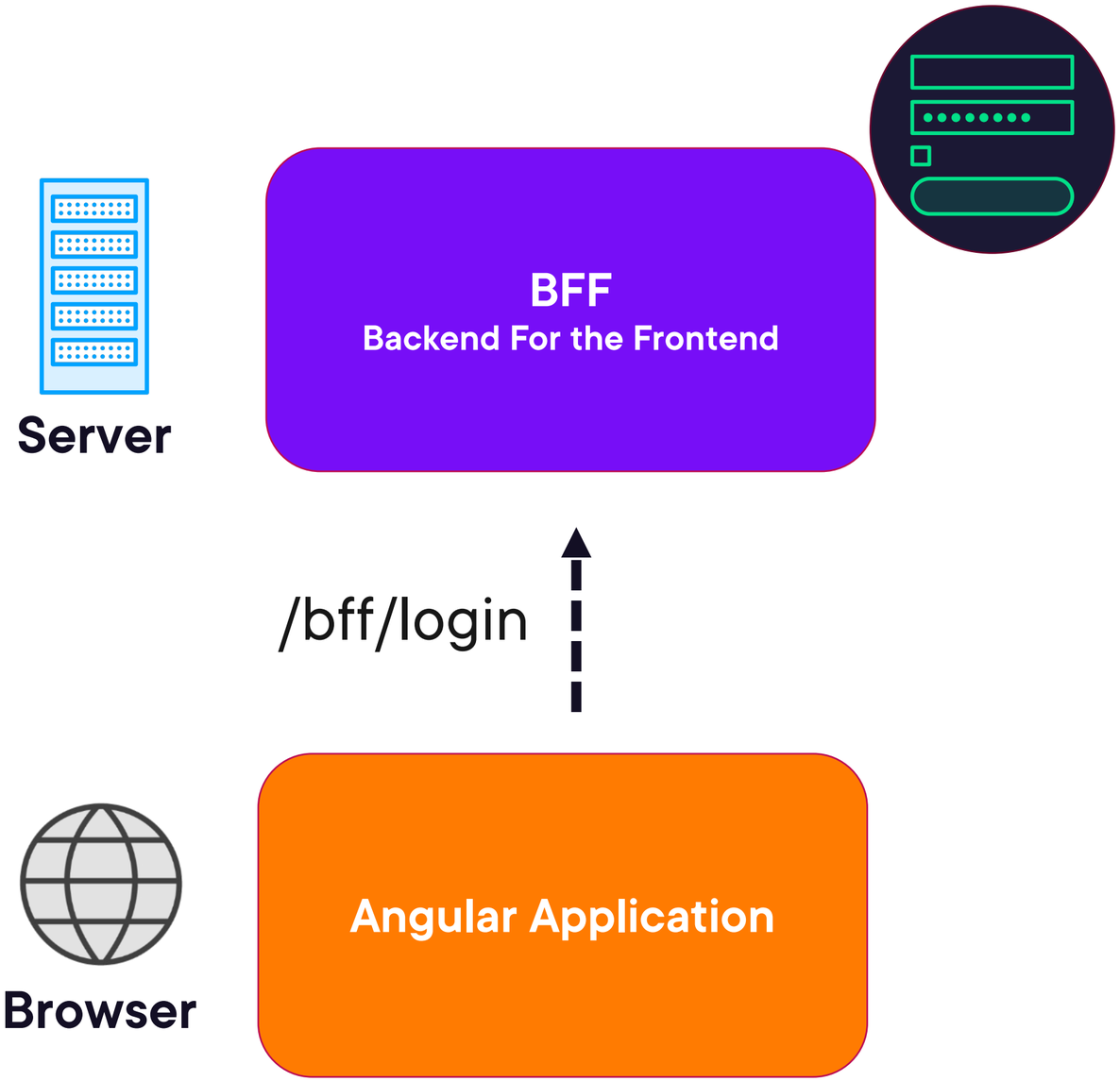
BFF - Backend for Frontend

OAuth 2.1

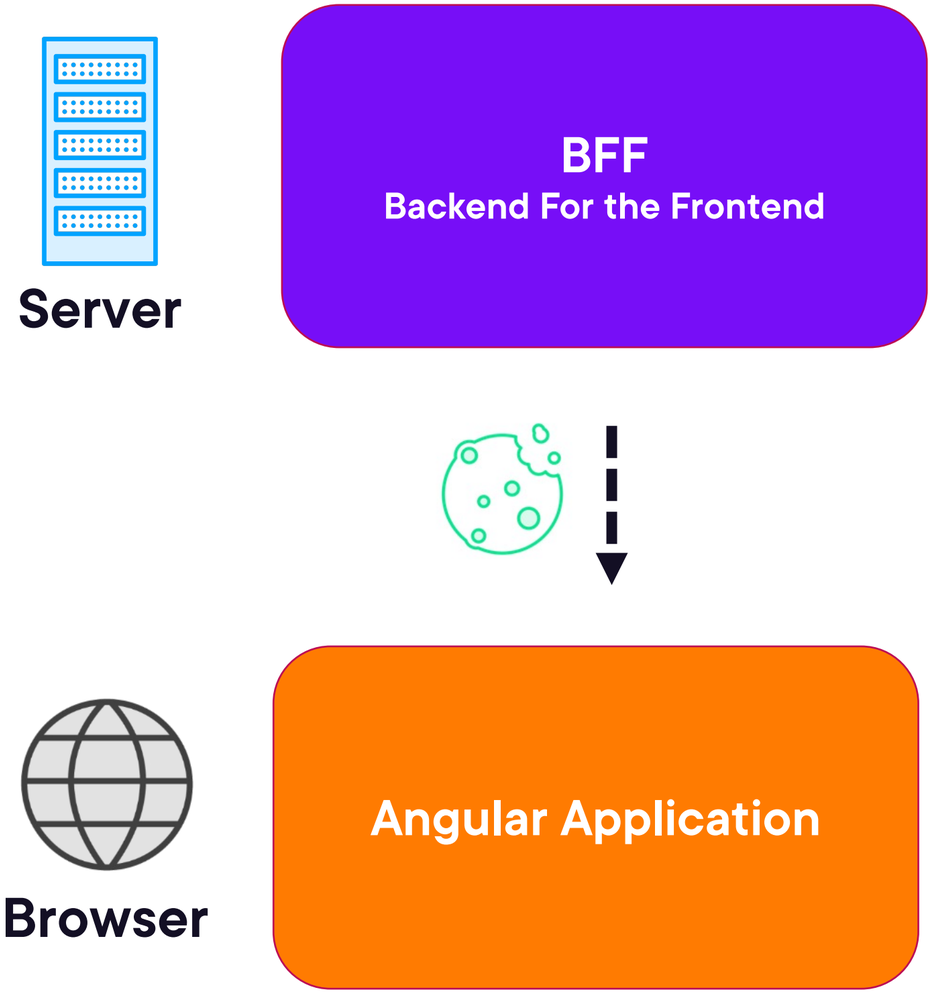
Implicit flow removed

Use BFF with SPAs

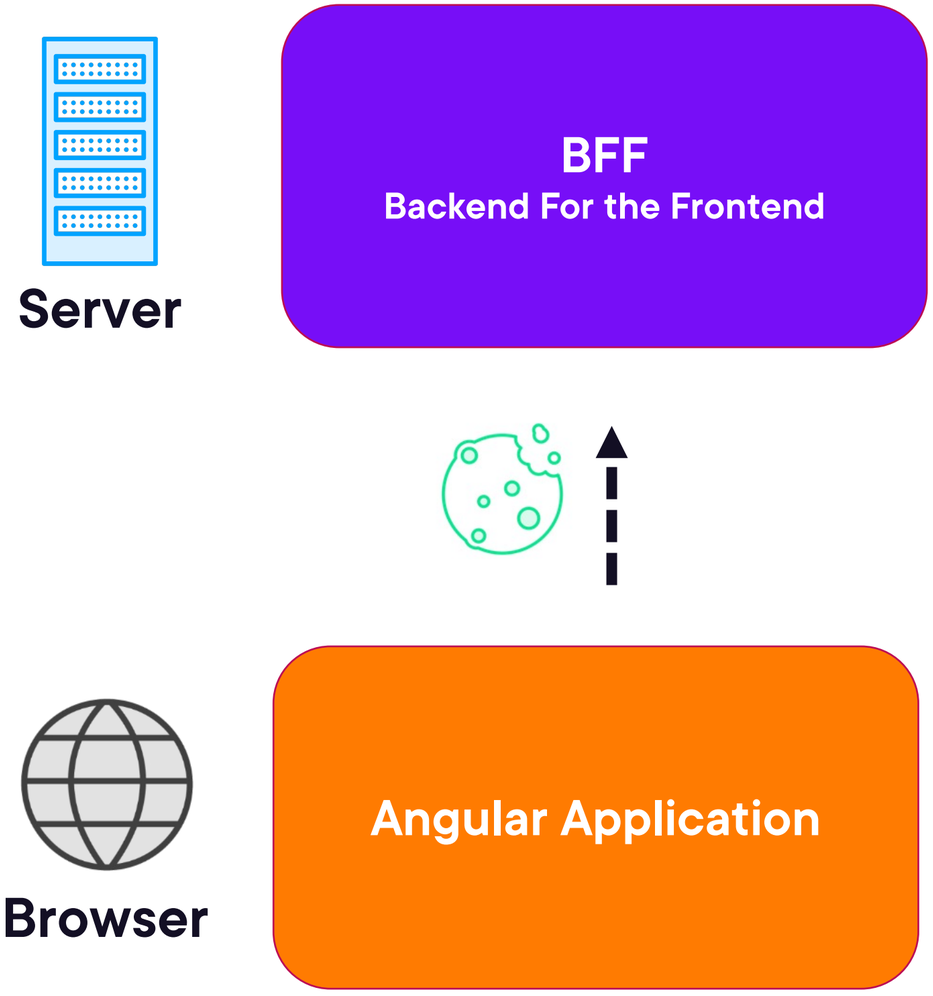
Cookie Authentication with BFF



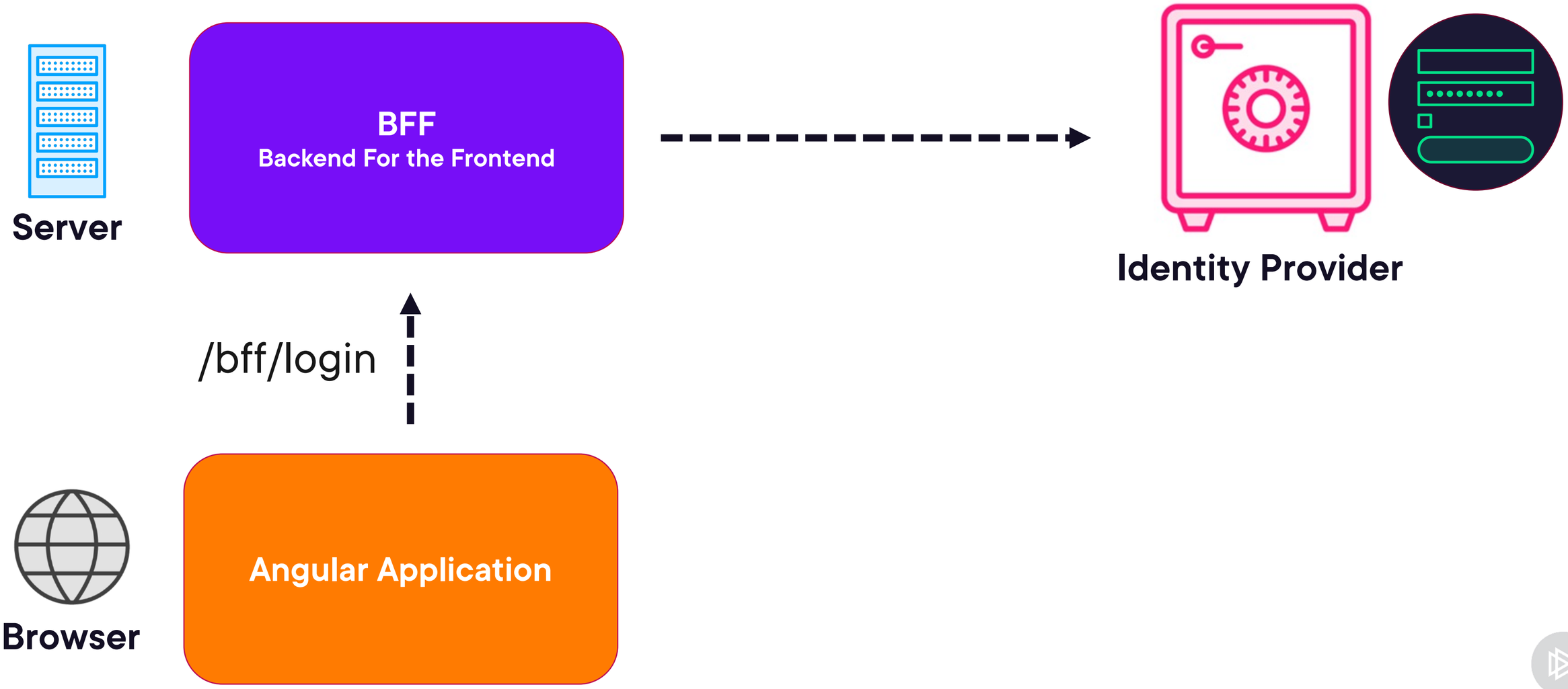
Cookie Authentication with BFF



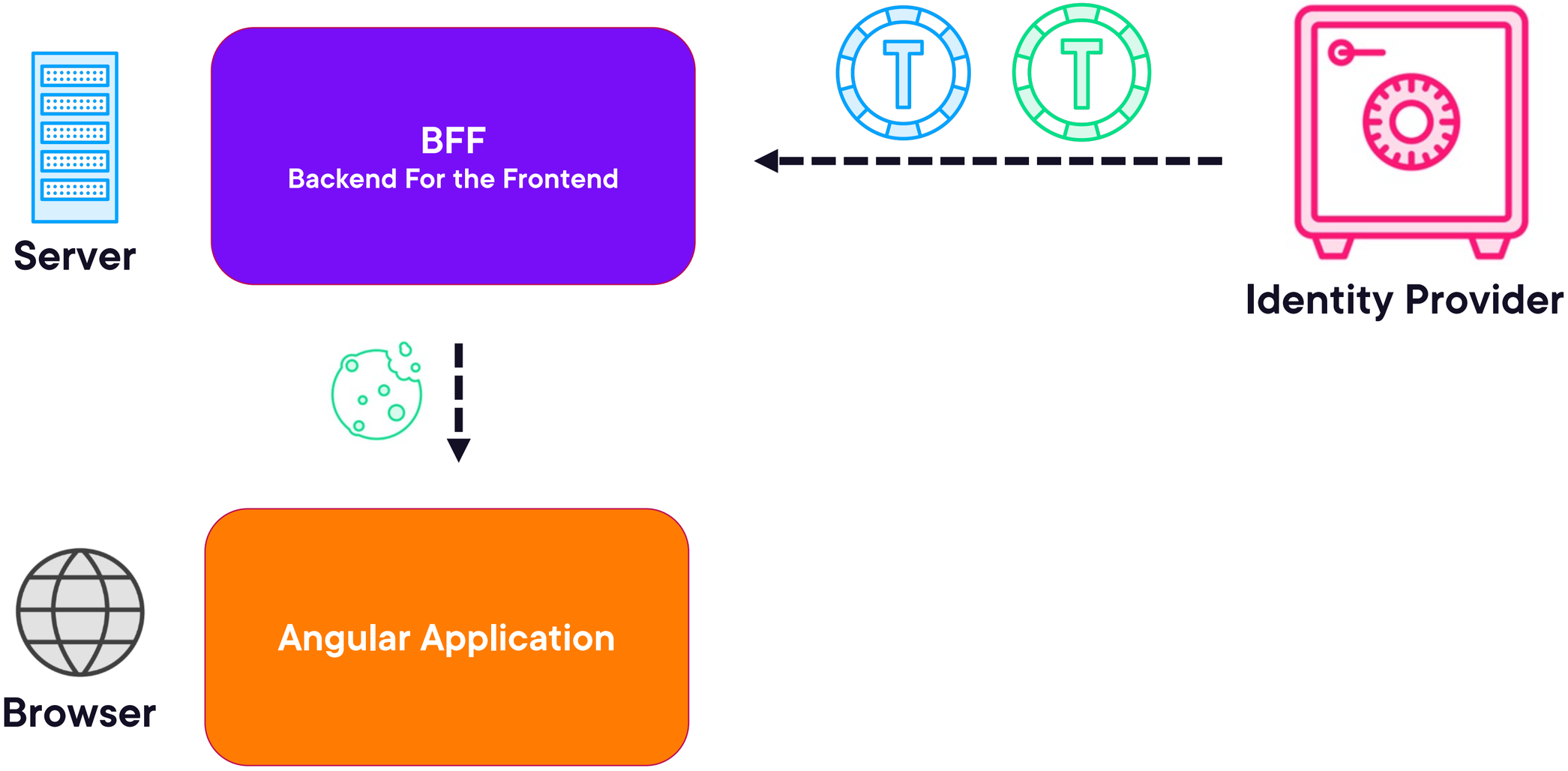
Cookie Authentication with BFF



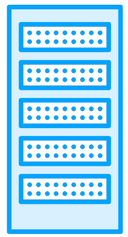
OpenID Connect with BFF



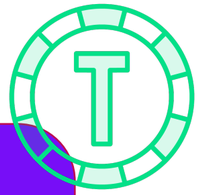
OpenID Connect with BFF



OpenID Connect with BFF



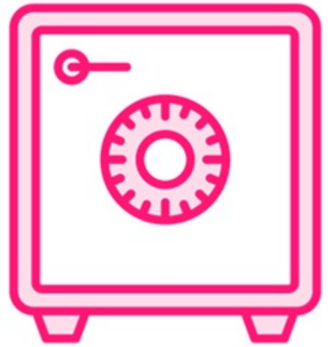
Server



BFF
Backend For the Frontend

Browser

Angular Application



Identity Provider

Session Cookie

Symmetric encryption

Only the BFF has the key

Not readable using javascript

BFF Endpoints

Login

Logout

- Deletes session cookie
- Deletes SSO cookie
- Trigger front/back channel logout

User

- Determine if user is logged in
- Get user claims

External APIs

Internal APIs

- Protected by session cookie

SameSite Cookies

Protect against CSRF

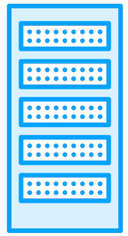
Only sent from the same site where it was set

For BFF: make the cookie usable only by the SPA

SameSite Cookies

https://bff.someserver.com

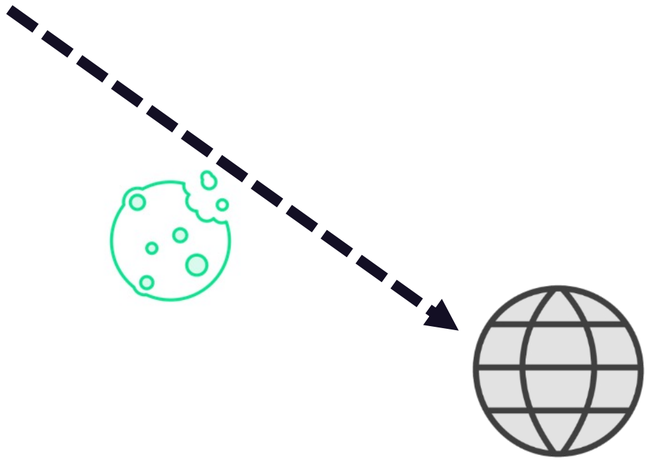
https://ng.mycdn.com



Server

BFF
Backend For the Frontend

Angular Application

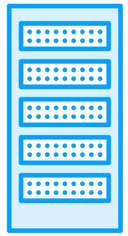


Browser: https://bff.someserver.com

SameSite Cookies

https://bff.someserver.com

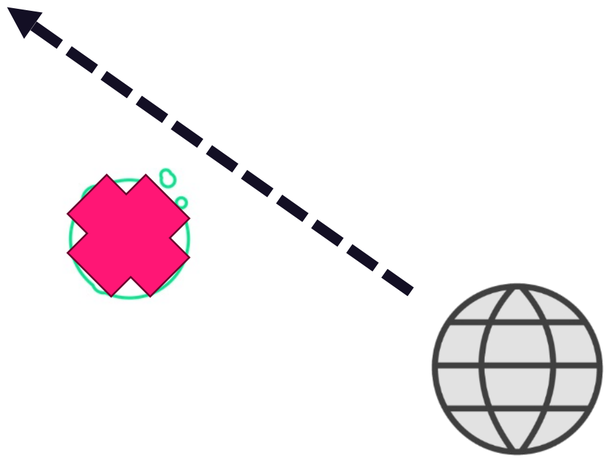
https://ng.mycdn.com



Server

BFF
Backend For the Frontend

Angular Application



Browser: https://ng.mycdn.com

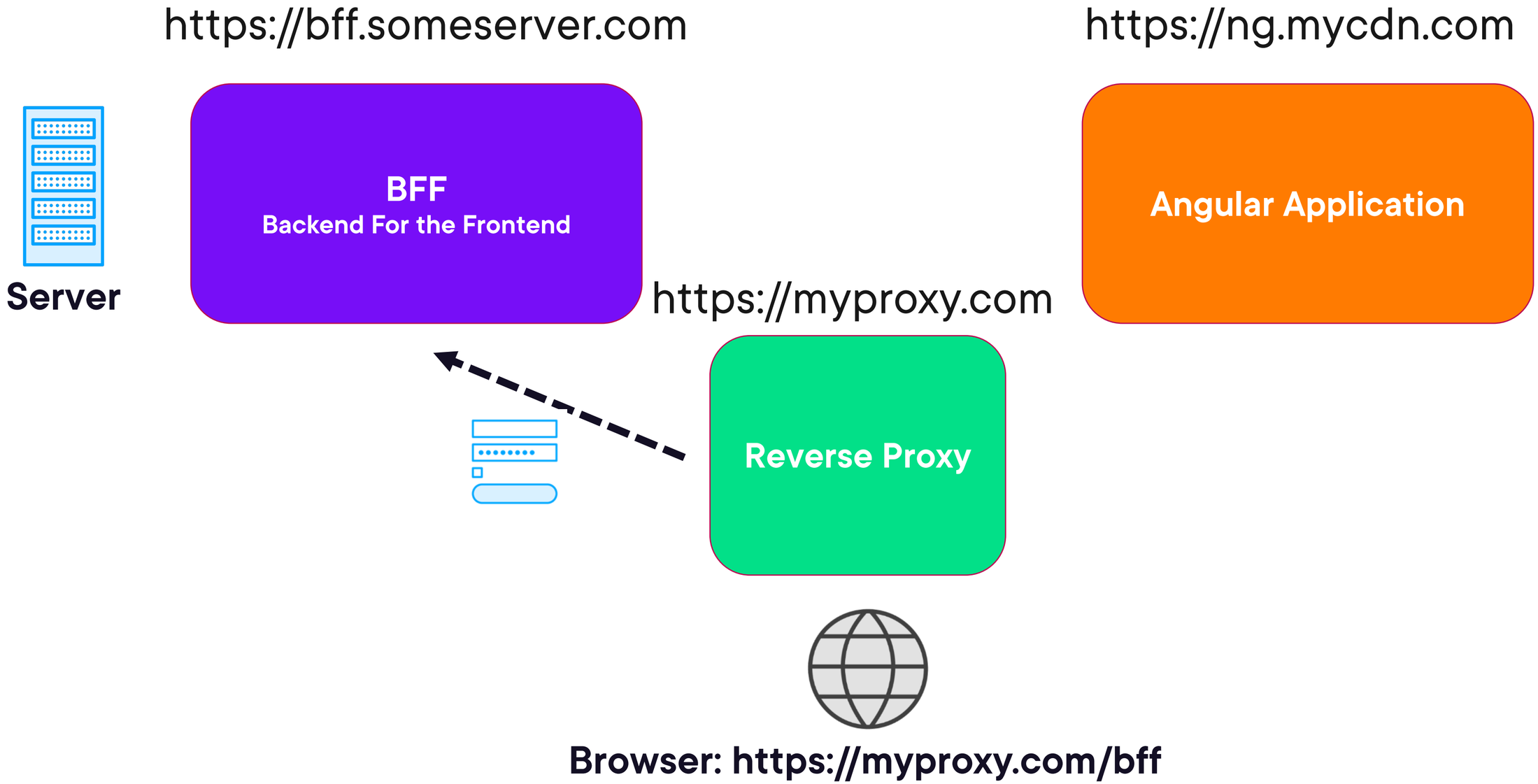
BFF and Angular Application on Same Site

Host Angular application from BFF

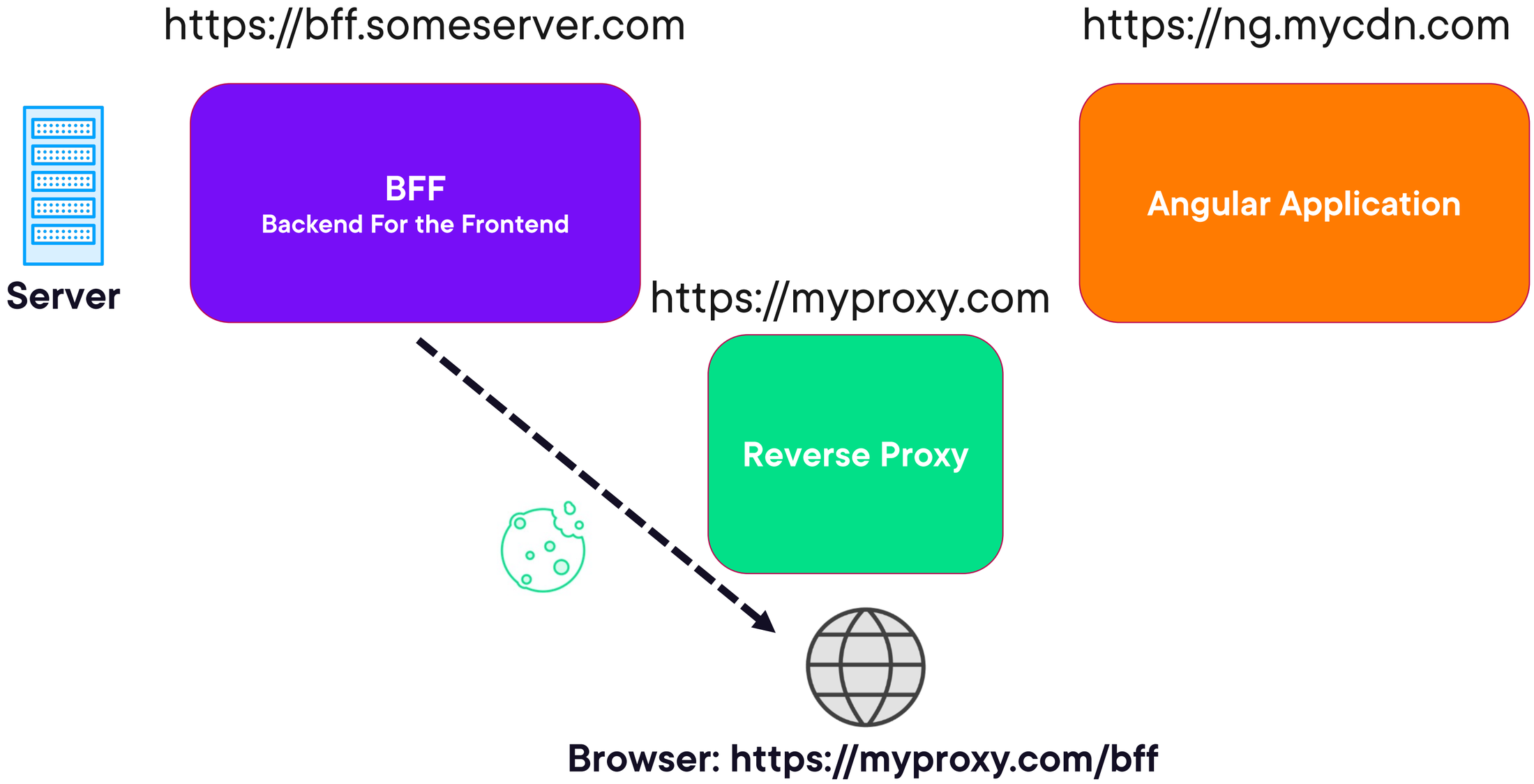
One deployable unit not practical at development and deployment time

Reverse proxy

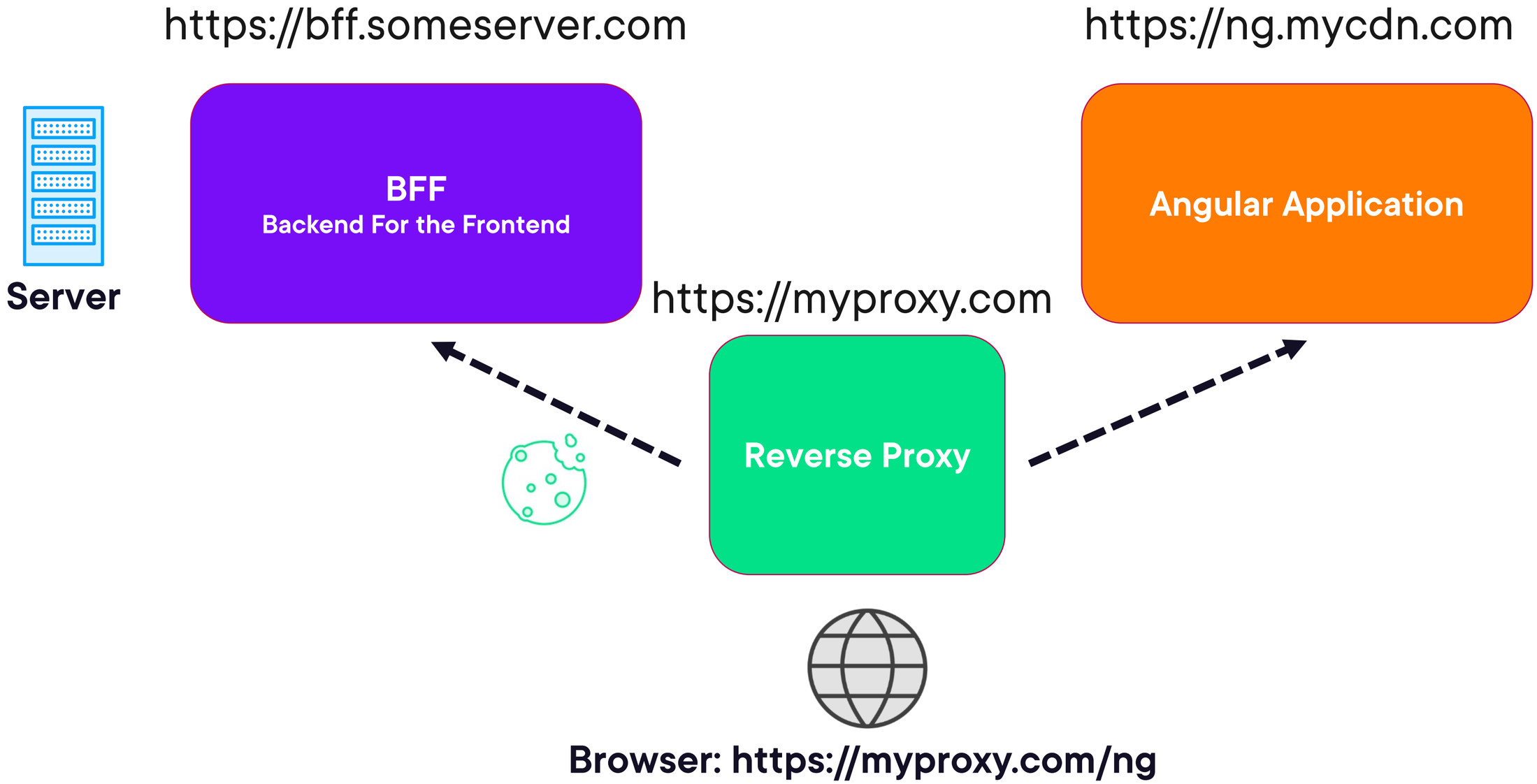
SameSite Cookies with Reverse Proxy



SameSite Cookies with Reverse Proxy



SameSite Cookies with Reverse Proxy



SameSite Cookie Modes

lax

strict

Up Next:

Setting Up a BFF with OAuth2 and OpenID Connect