

Authorization with Route Guards and Conditional Rendering



Roland Guijt

Freelance consultant and trainer

@rolandguijt | roland.guijt@gmail.com

Where to Implement Authorization

Identity provider is about authentication
Implement in all applications that receive claims/authorization data





Why write authorization rules multiple times?



**Implement authorization rules
on frontend and backend**





Authentication and Authorization in ASP.NET Core



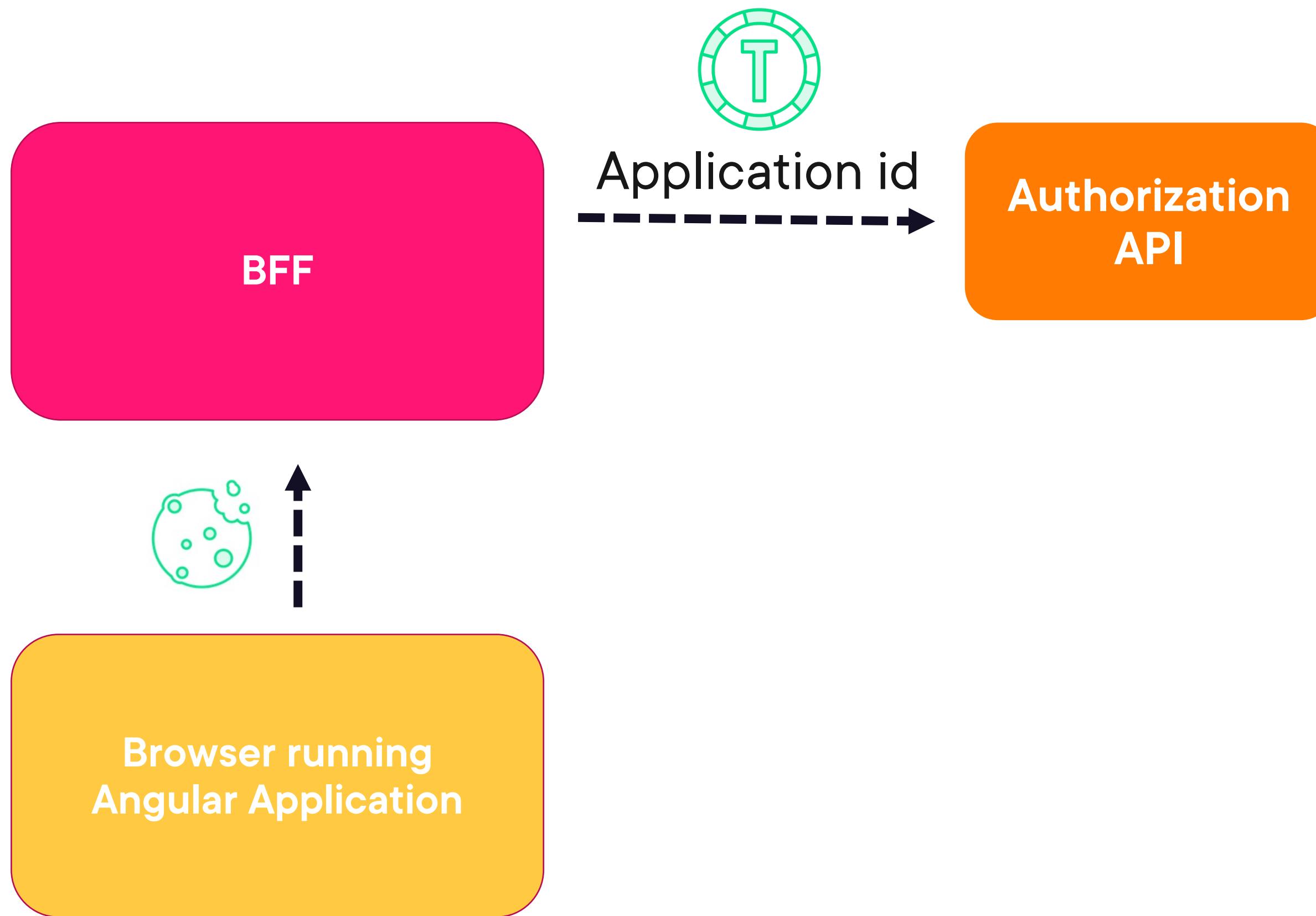
Authorization Data

Claims from the identity provider should be applicable to all identity provider clients

Other sources of authorization data should be created for application specifics



Authorization API



**Angular's route guards
protect individual routes by
calling functions**



Route Guards and Child Routes

```
{ path: 'product', component: ProductListComponent,
  canActivate: [() => inject(AuthService).authenticated],
  children: [
    { path: 'view/:id', component: ProductDetailsComponent },
    { path: 'edit/:id', component: ProductEditComponent },
    { path: 'add', component: ProductAddComponent },
  ],
}
```



Route Guards and Child Routes

```
{ path: 'product', component: ProductListComponent,
  canActivate: [() => inject(AuthService).authenticated],
  children: [
    { path: 'view/:id', component: ProductDetailsComponent,
      canActivate: [() => inject(AuthService).hasProductAccess()] },

    { path: 'edit/:id', component: ProductEditComponent,
      canActivate: [() => inject(AuthService).hasProductAccess()] },

    { path: 'add', component: ProductAddComponent,
      canActivate: [() => inject(AuthService).hasProductAccess()] },
  ],
}
```



Route Guards and Child Routes

```
{ path: 'product', component: ProductListComponent,
  canActivate: [() => inject(AuthService).authenticated],
  canActivateChild: [() => inject(AuthService).hasProductAccess()],
  children: [
    { path: 'view/:id', component: ProductDetailsComponent },
    { path: 'edit/:id', component: ProductEditComponent },
    { path: 'add', component: ProductAddComponent },
  ]
},
```



canMatch

```
[  
  { path: 'product', component: ProductListAdminComponent,  
    canMatch: [() => inject(AuthService).canShowAdminList()] },  
  
  { path: 'product', component: ProductListComponent,  
    canMatch: [() => inject(AuthService).authenticated] },  
  
  { path: 'product', component: ProductListAnonymousComponent },  
]
```



Thank you for watching!



Roland Guijt

Freelance consultant and trainer

@rolandguijt | roland.guijt@gmail.com