Authentication with ASP.NET Core Identity

Roland Guijt

Freelance Trainer and Consultant | Microsoft MVP

@rolandguijt | roland.guijt@gmail.com

What is Identity?

Framework leveraging cookie authentication

Adds needed functionality around authentication

A Few ASP.NET Identity Features

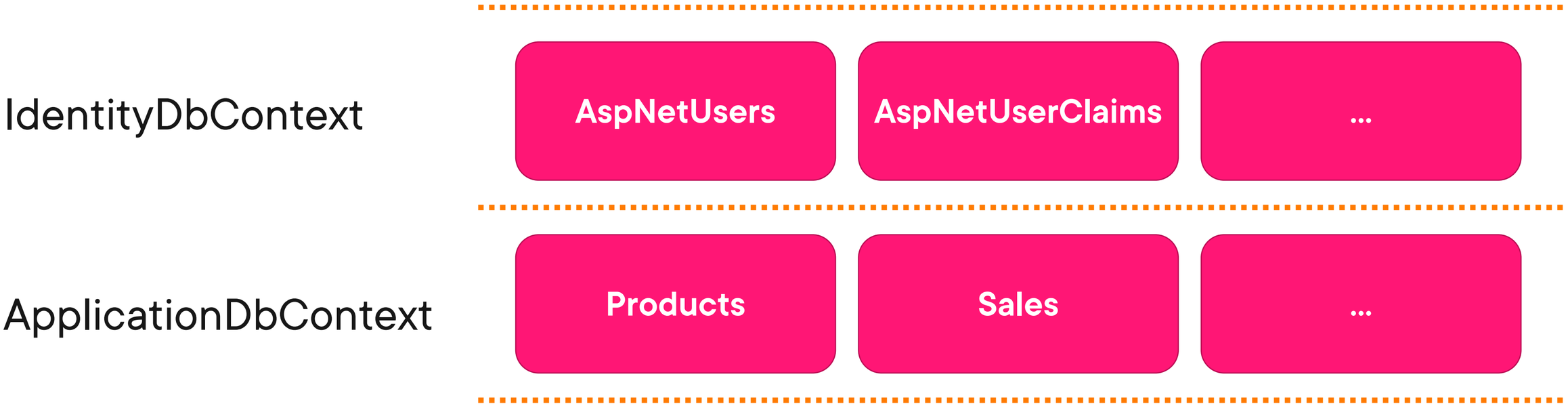
- Login and logout
- User registration
- Password management
- Two-factor authentication
- Account lockout
- User store

Course recommendation:

EF Core 6: The Big Picture

Julie Lerman

DbContexts in ASP.NET Core Identity



DbContexts in ASP.NET Core Identity

https://4sh.nl/configureidentity

Customizing Identity's UI

Require knowledge of Razor pages

Can be customized

But is it really needed?

_ViewStart.cshtml already sets the themed layout page

Some pages need more customization than others

Pages and Endpoints

If an endpoint for an Identity page exists in the project, that will be used

Identity uses the UI assembly as a fallback

Coming up: retrofitting Identity

Your Own User Class

- Create it first before retrofitting Identity
- Derive from IdentityUser
- Possibility to add extra properties
- Hard to add later

Please look at the code for the repositories and controllers in the provided solution if you're coding along.

SecurityStamp

Value stored in user table

Changes when credentials change or the account is locked

Stored as a claim in identity cookie

When new request occurs value for cookie is compared to database value

If no match: identity cookie rejected

Claims in UserClaims Table Or In User Type?

UserClaims

Easy

No need to add migration

No need to use IClaimsTransformation

Extra database query needed to get them

User type

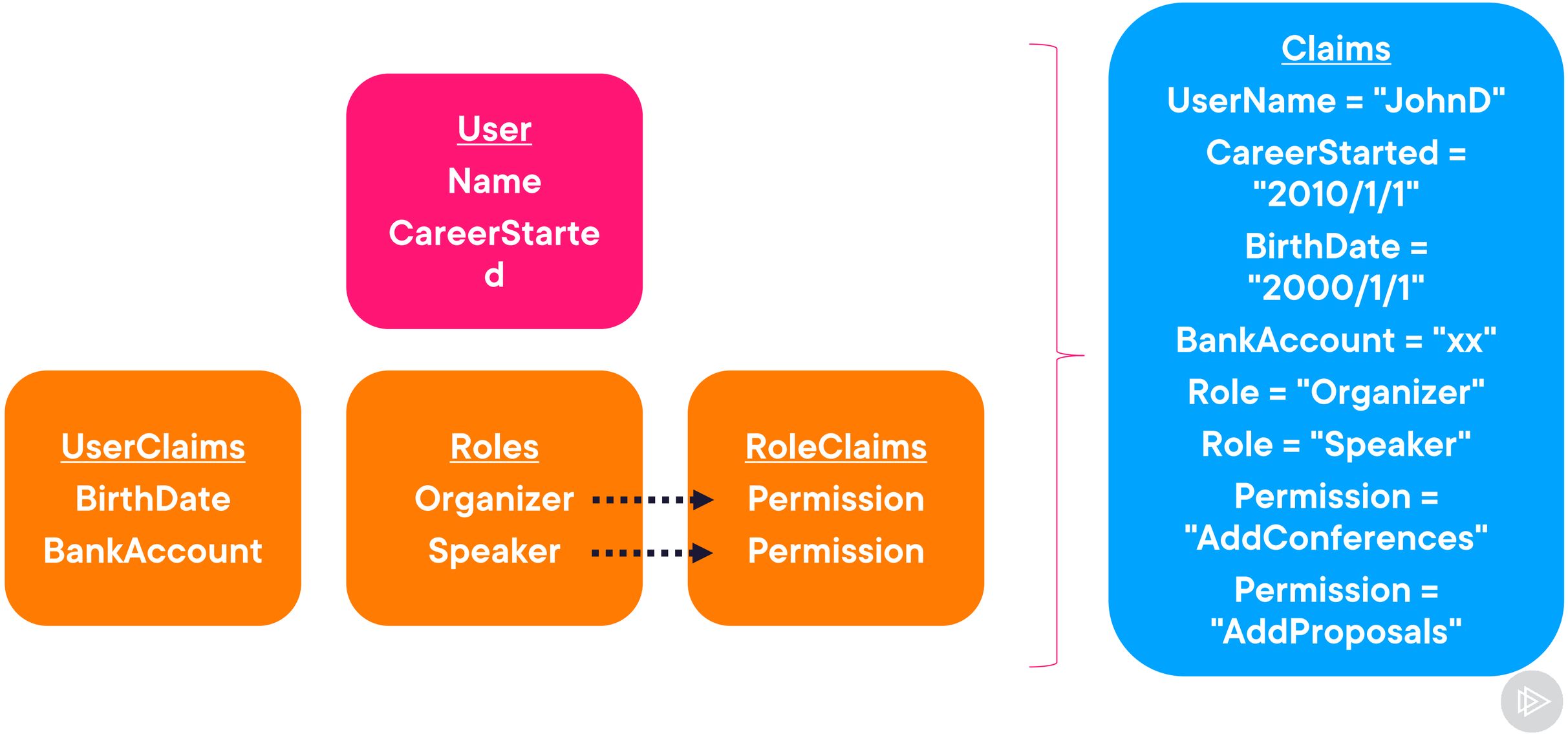
Claims directly accessible when using Identity's APIs

No additional queries needed

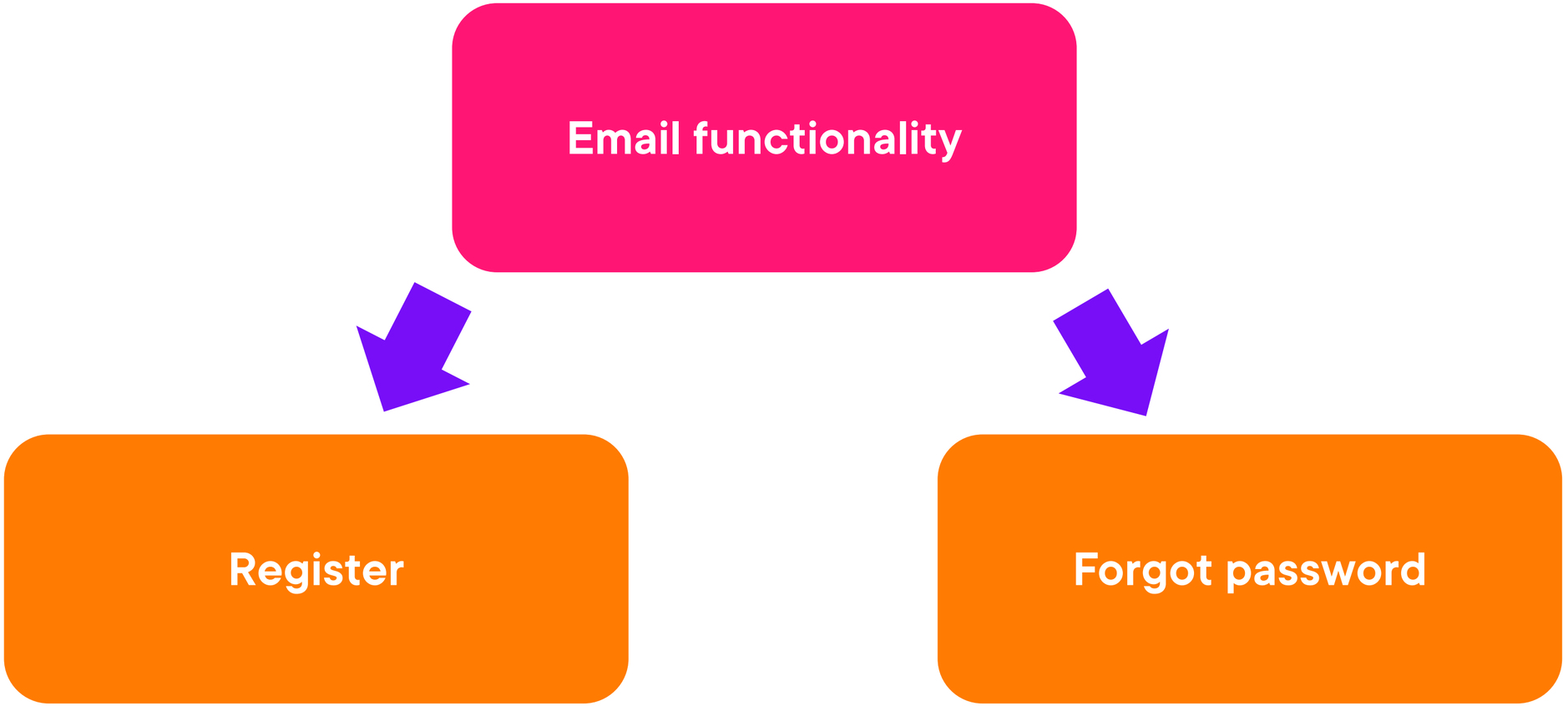
Role support is disabled by default

Role is a claim type

Claims and Roles

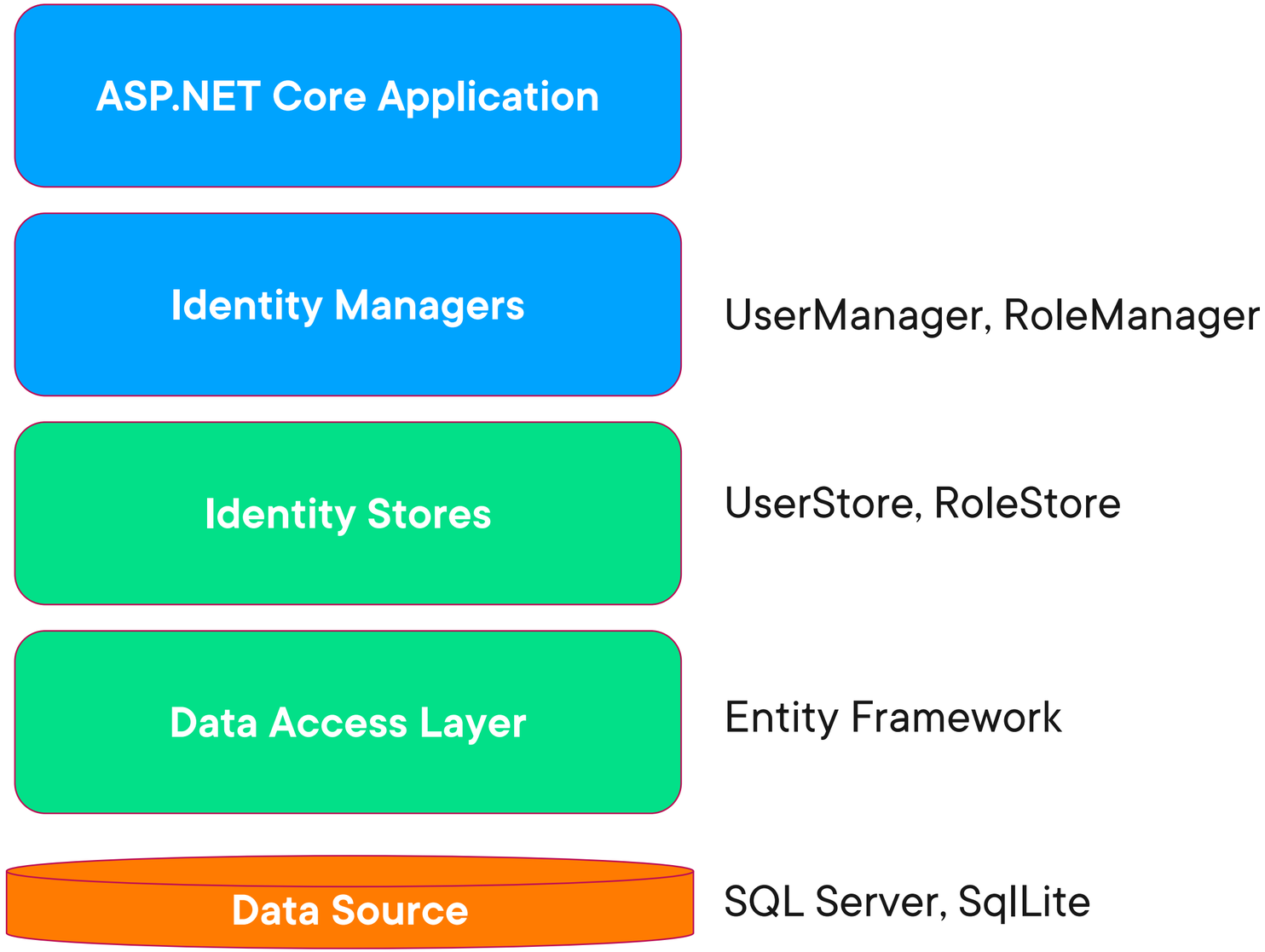


Use RoleManager<TRole> to manage roles


Email functionality
Register
Forgot password

https://4sh.nl/qrcodejs

Identity's Architecture



Examples of Identity's Types

PasswordValidator

PasswordHasher

RoleManager

RoleStore

EmailTokenProvider

SecurityStampValidator

UserValidator

SigninManager

UserStore

https://4sh.nl/identitycustom

Customizing Identity Recipe

Find the class involved

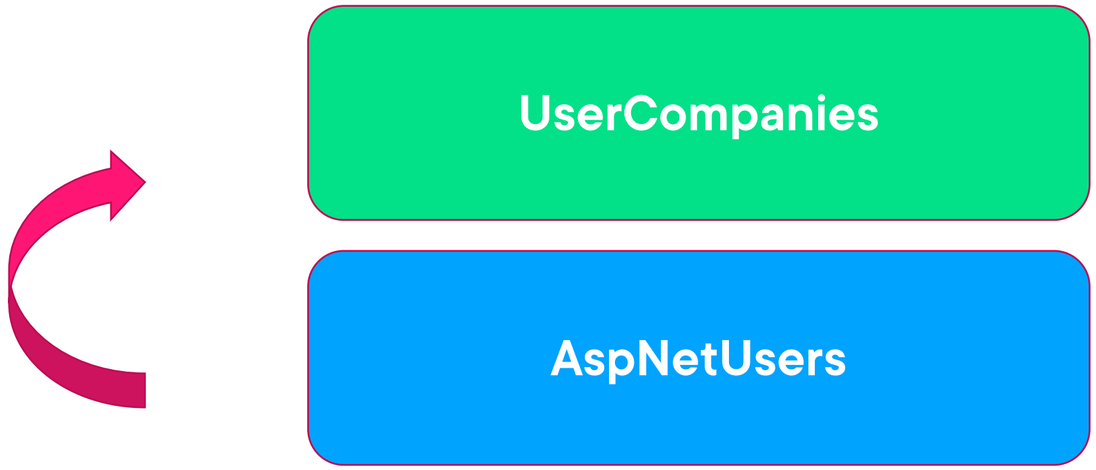
Derive from the existing class you would like to customize

Override the relevant methods

Use a method on Identity's builder object to register it

Or just register in the dependency injection container

Extensibility: An Example



Implementation Steps

Create entity

Add id and navigation property in ApplicationUser

Add to DbContext

Modify Identity's UserStore behavior

Course recommendation:

Secure User Account and Authentication Practices

Erik Dahl

Up Next:

Multi-Application Authentication with OpenID Connect