



Enterprise Software Development

Join the Conversation #CleanArchitecture @dddsydney @jasongtau



NDC { Sydney }



Join the Conversation #CleanArchitecture @dddsydney @jasongtau



Clean Architecture

with ASP.NET Core 2.1

DDD Sydney - August 2018



Jason Taylor

SSW Solution Architect



@jasongtau



codingflow.net



github.com/jasongt

Join the Conversation #CleanArchitecture @dddsydney @jasongtau

Agenda

Architecture & Design

Domain Layer

Application Layer

Persistence Layer

Infrastructure Layer

Presentation Layer

Overview

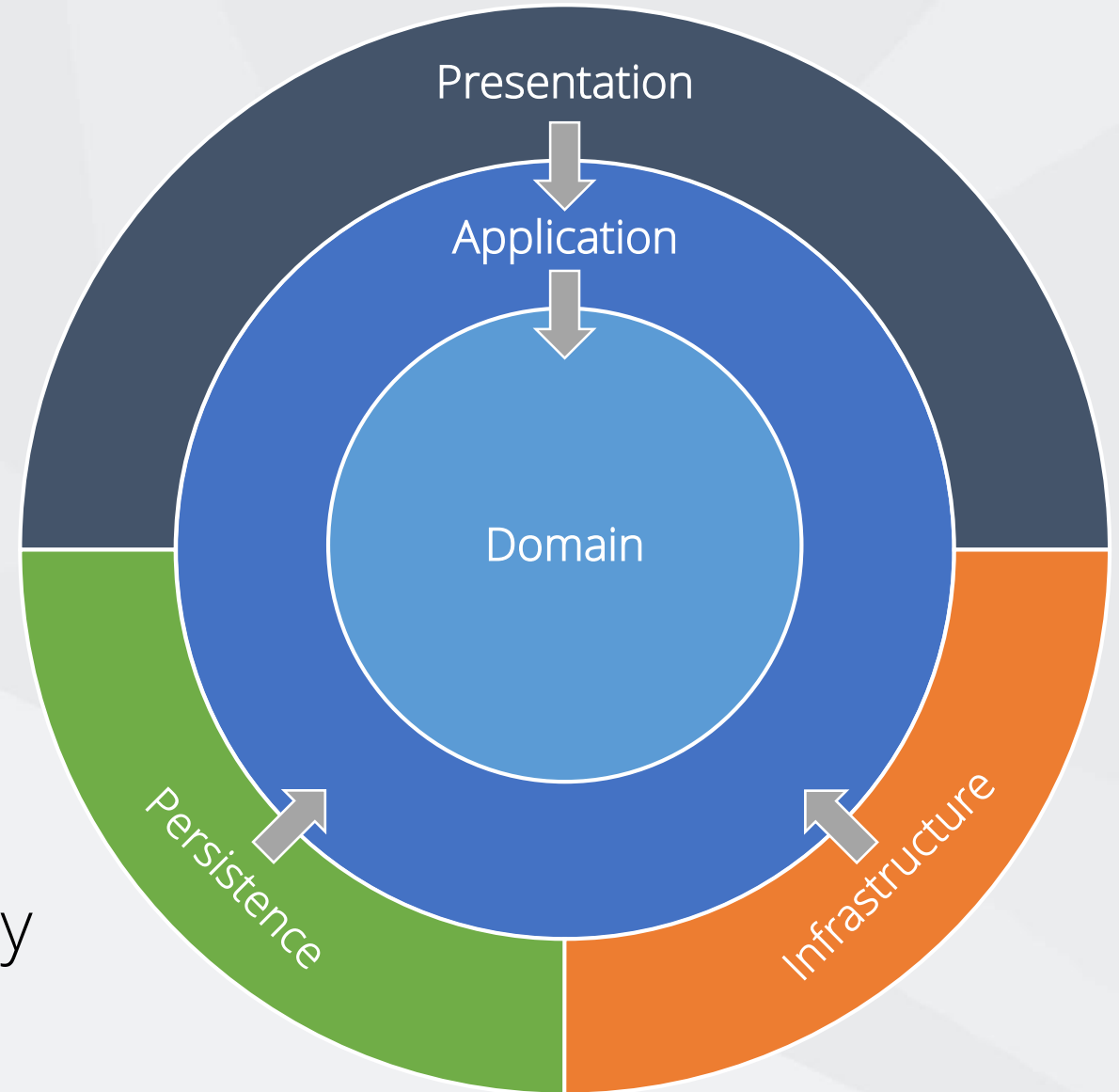
Independent of frameworks

Testable

Independent of UI

Independent of database

Independent of external agency



Northwind Traders



Cross Platform

.NET Core

Entity Framework Core

Code First

Data Seeding



Key Points

Domain contains enterprise-wide types and logic

Application contains application-specific types and logic

Infrastructure (including Persistence) contain all external concerns

Presentation and Infrastructure depend only on Application

Infrastructure and Presentation components can be replaced with minimal effort

Agenda

Architecture & Design

Domain Layer

Application Layer

Persistence Layer

Infrastructure Layer

Presentation Layer

Overview

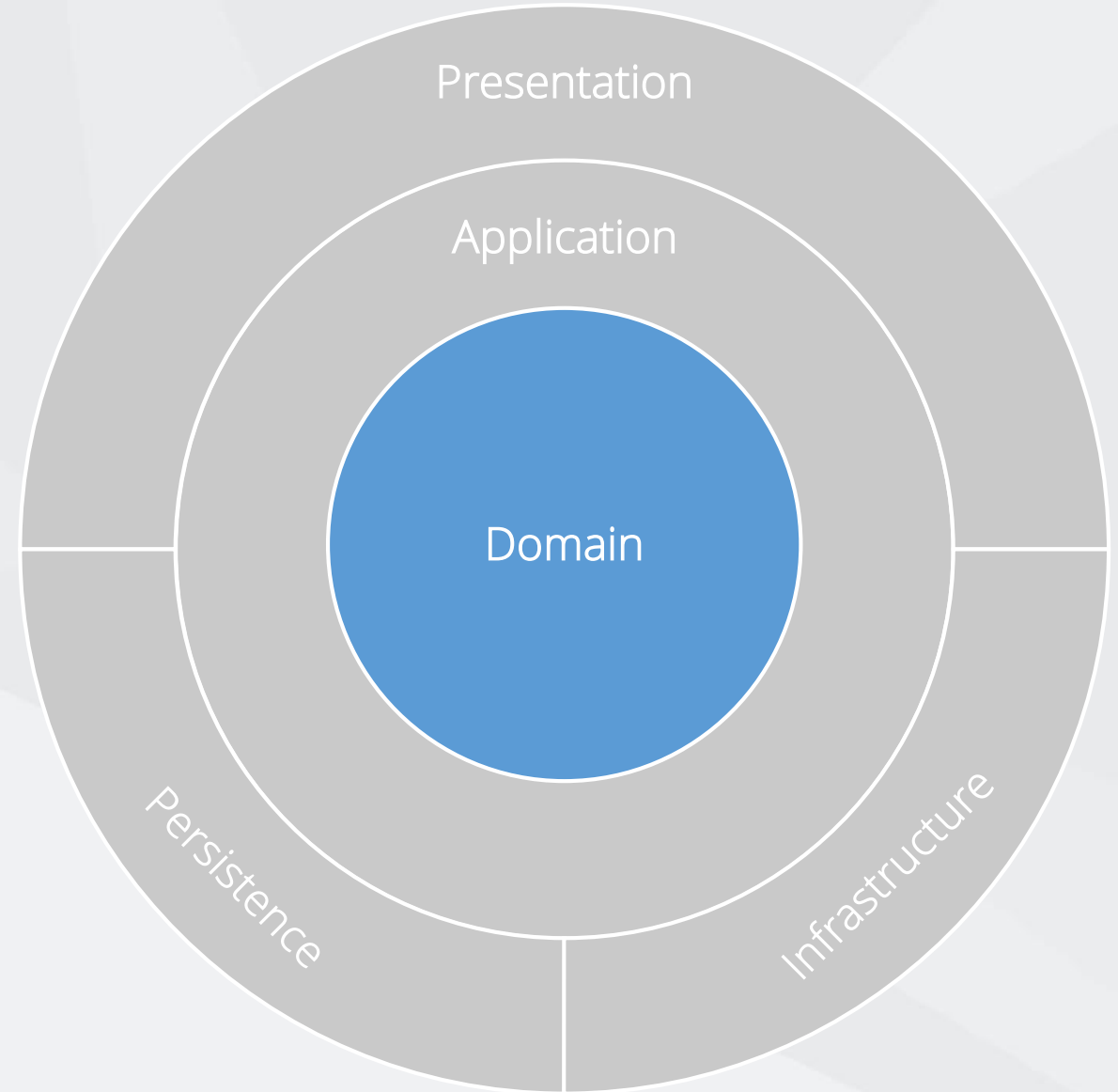
Entities

Value Objects

Enumerations

Logic

Exceptions



Demo

Reviewing the Domain layer



Key Points

Use data annotations sparingly

Use value objects when appropriate

Initialise all collections & use private setters

Create custom domain exceptions

Agenda

Architecture & Design

Domain Layer

Application Layer

Persistence Layer

Infrastructure Layer

Presentation Layer

Overview

Interfaces

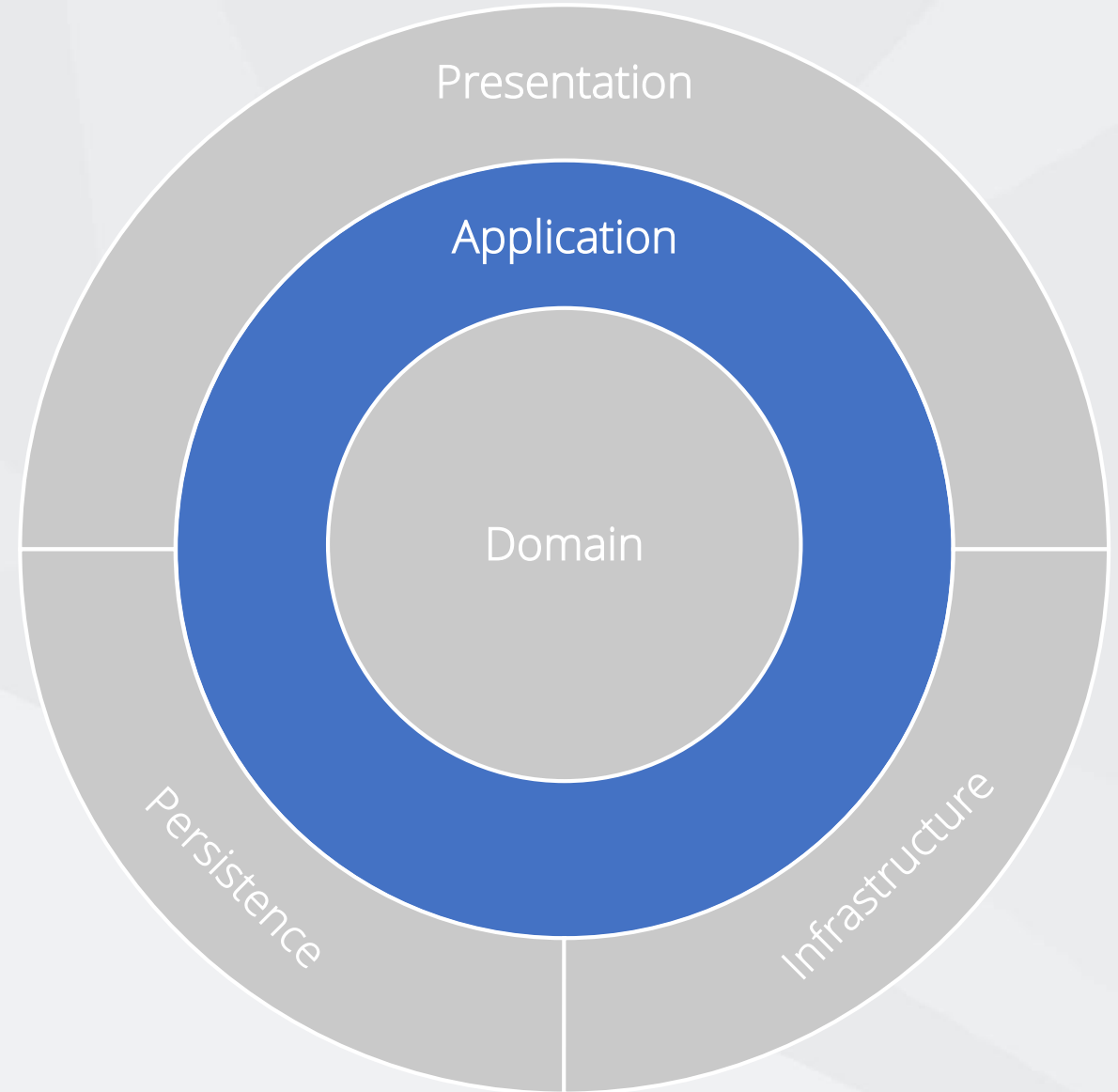
Models

Logic

Commands / Queries

Validators

Exceptions



CQRS

Command Query Responsibility Segregation

Separate reads (queries) from writes (commands)

Can maximise performance, scalability, and security

Can simplify your overall design

Easy to add new features, just add a new query or commands

Easy to maintain, changes only affect one command or query

MediatR + CQRS = ❤️

Define queries and commands as requests

Application layer is just a series of request / response objects

Ability to attach additional behaviour before and / or after each request, e.g. logging, validation, caching, authorisation and so on

Demo

Reviewing the Application layer

Key Points

Using CQRS + MediatR simplifies your overall design

Fluent Validation is useful for simple and complex validation scenarios

MediatR simplifies cross cutting concerns such as logging and validation

Independent of infrastructure and data access concerns

Agenda

Architecture & Design

Domain Layer

Application Layer

Persistence Layer

Infrastructure Layer

Presentation Layer

Overview

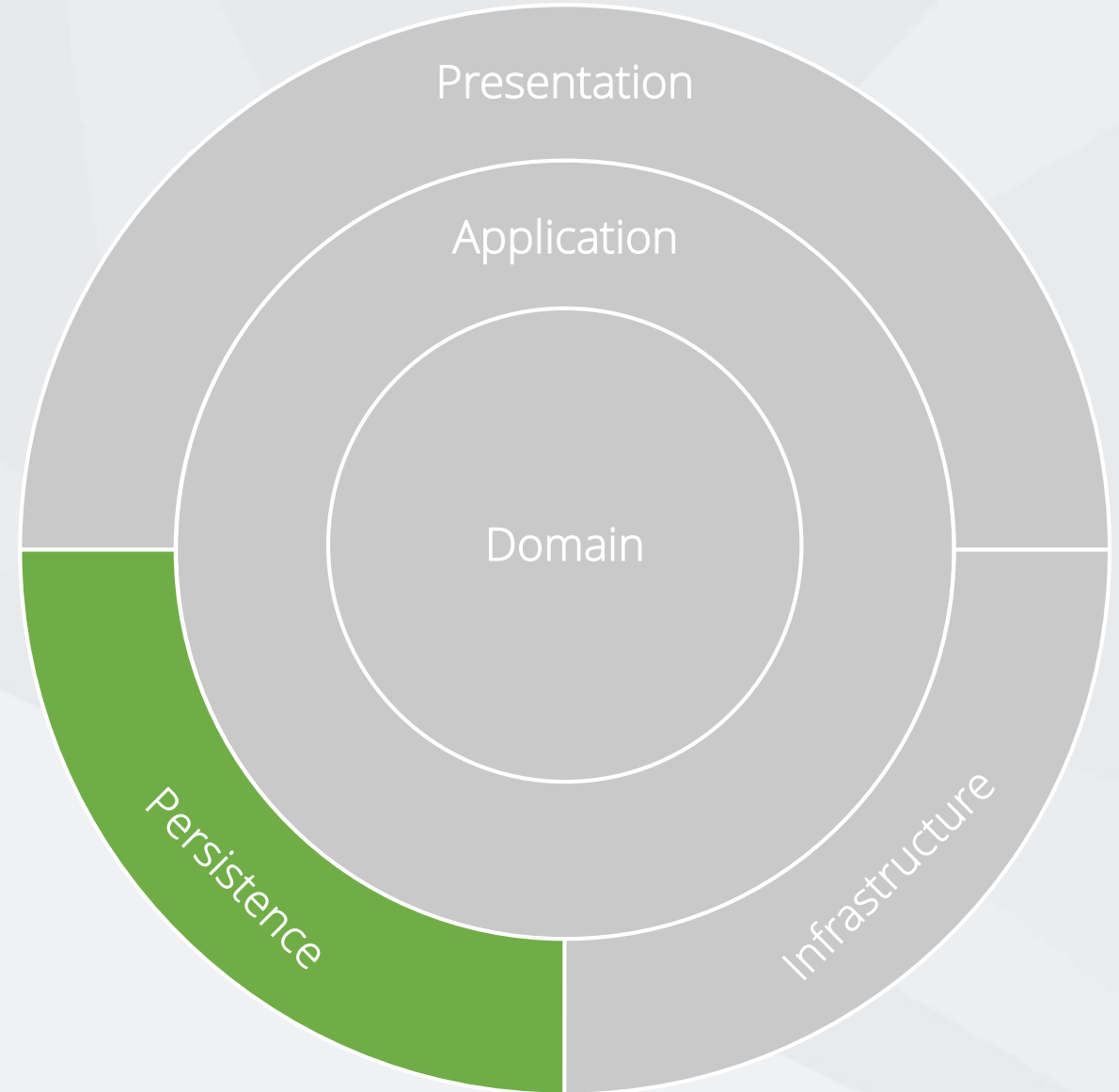
DbContext

Migrations

Configurations

Seeding

Abstractions



Unit of Work and Repository Patterns

Should we implement these patterns?



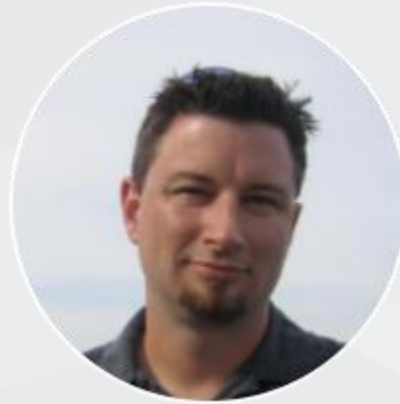
It isn't always the best choice, because:

- DbContext insulates your code from database changes
- DbContext acts as a unit of work
- DbSet acts as a repository
- EF Core has features for unit testing without repositories

What do the experts think?



I'm over Repositories, and definitely over abstracting your data layer.



No, you don't *need* a repository. But there are many benefits and you should consider it!



No, the repository/unit-of-work pattern isn't useful with EF Core.

Demo

Reviewing the Persistence layer

Key Points

Independent of the database

Prefer conventions over configuration

Use Fluent API Configuration over Data Annotations

Use an extension to automatically apply all entity type configurations

Agenda

Architecture & Design

Domain Layer

Application Layer

Persistence Layer

Infrastructure Layer

Presentation Layer

Overview

Implementations, e.g.

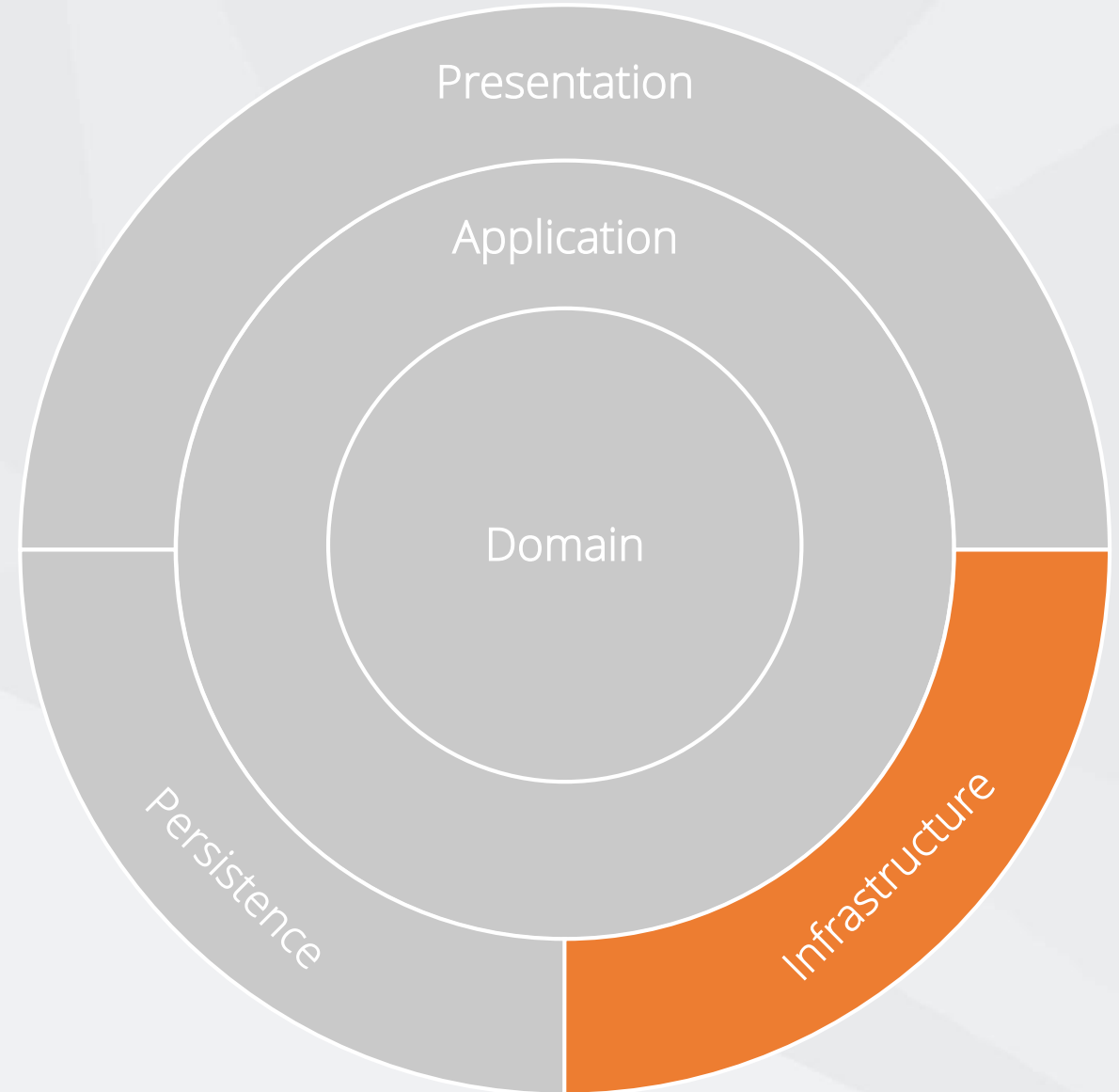
API Clients

File System

Email / SMS

System Clock

Anything external



Demo



Reviewing the Infrastructure layer

Join the Conversation #CleanArchitecture @dddsydney @jasongtau

Key Points

Contains classes for accessing external resources

Such as file systems, web services, SMTP and so on

Implements abstractions / interfaces defined within the
Application layer

No layers depend on Infrastructure layer, e.g. Presentation
layer

Agenda

Architecture & Design

Domain Layer

Application Layer

Persistence Layer

Infrastructure Layer

Presentation Layer

Overview

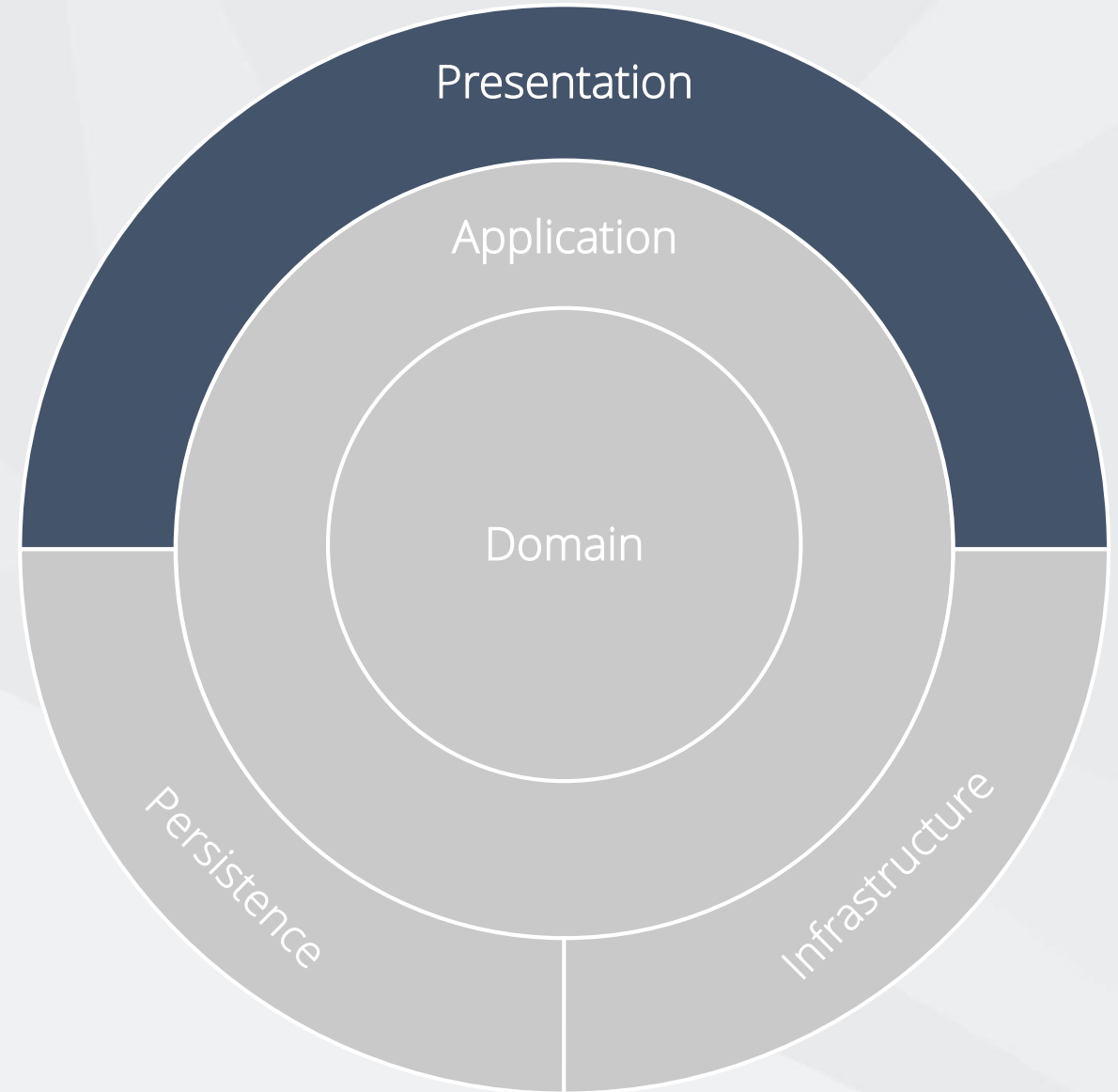
SPA – Angular or React

Web API

Razor Pages

MVC

Web Forms



Demo



Reviewing the Presentation layer

Join the Conversation #CleanArchitecture @dddsydney @jasongtau

Key Points

Controllers should not contain any application logic

Create and consume well defined view models

Utilising Open API bridges the gap between the frontend and backend

Recommend Resources

Join the Conversation #CleanArchitecture @dddsydney @jasongtau

2nd Edition
(ASP.NET Core 2 support)



Architecting Modern Web Applications with ASP.NET Core and Microsoft Azure



Steve Smith

Building Monoliths

Clean Architecture

Azure

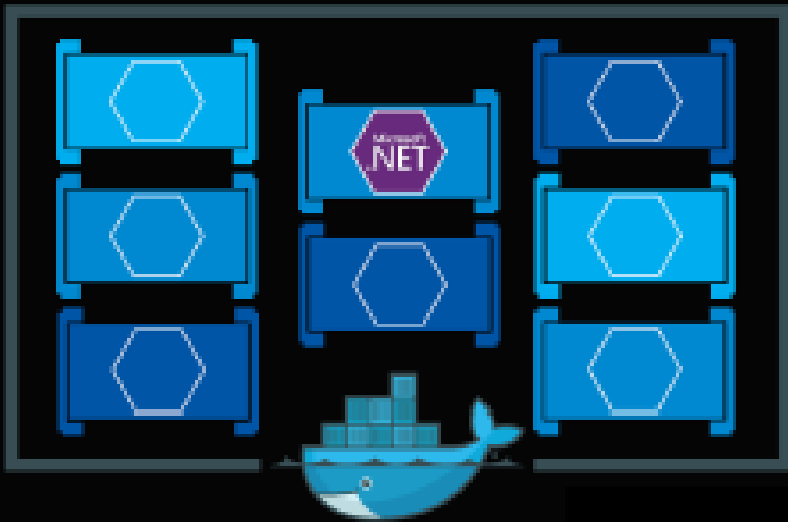
aka.ms/webappebook

Join the Conversation #CleanArchitecture @dddsydney @jasongtau

v2.1 Edition
(.NET Core 2.1 support)



.NET Microservices: Architecture for Containerized .NET Applications



Cesar de la Torre
Bill Wagner
Mike Rousos
Microsoft Corporation

Building Microservices

Microservices

Containers

DDD

Azure

aka.ms/microservicesesebook

Join the Conversation #CleanArchitecture @dddsydney @jasongtau

Robert C. Martin Series

Clean Architecture

A Craftsman's Guide to
Software Structure and Design

Robert C. Martin

Foreword by Kevlin Henney
Afterword by Jason Gorman



Clean Architecture

Robert C. Martin

bit.ly/clean-architecture-book

Join the Conversation #CleanArchitecture @dddsydney @jasongtau

DEV SUPERPOWERS
• TOUR •

.NET CORE

SYDNEY • BRISBANE • MELBOURNE • NOVEMBER 2018

DURATION

1 Day

PRICE

\$49 inc GST

Brisbane

MON, 26TH NOV 2018

[Book Now](#)

 SSW Brisbane

Melbourne

THU, 29TH NOV 2018

[Book Now](#)

 SSW Melbourne

Sydney

FRI, 30TH NOV 2018

[Book Now](#)

 SSW Sydney

About the presenters



Jason Taylor

Jason is a passionate solution architect with over 16 years professional experience.

He has worked with .NET since 2001, and currently specialises in teaching and developing enterprise applications utilising .NET Core, ASP.NET Core, EF

Sydney – Friday, 30th November - bit.ly/dotnet-tour

Next Step

Code & Slides

github.com/jasongt/northwindtraders



Thank you!

info@ssw.com.au

www.ssw.com.au

Sydney | Melbourne | Brisbane