

# Module 08:

## "Variations and Alternatives"



# Agenda

- ▶ Introduction
- ▶ **The Many Names of Clean Architecture**
- ▶ Vertical Slice Architecture
- ▶ Summary

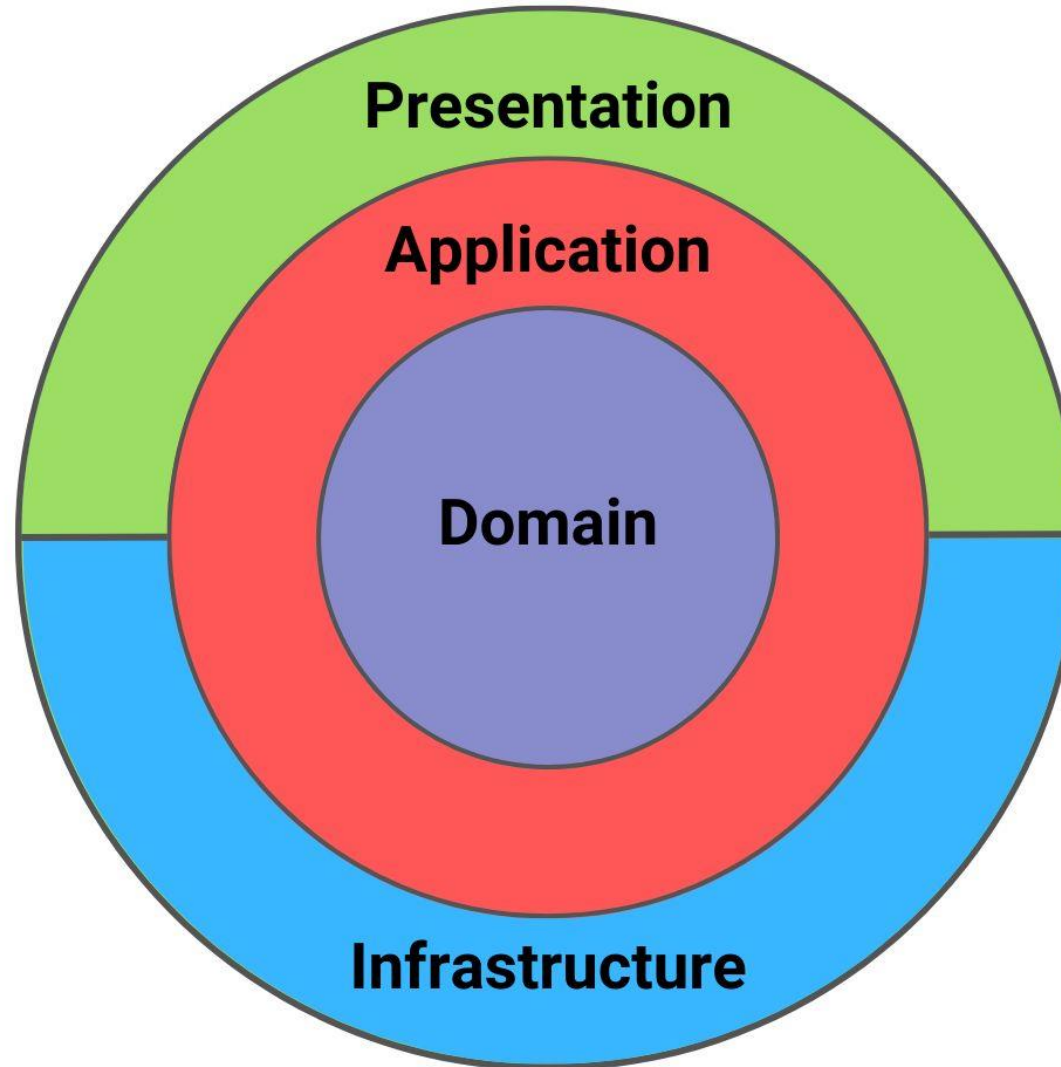


# Domain-centric Architectural Principles in Summary

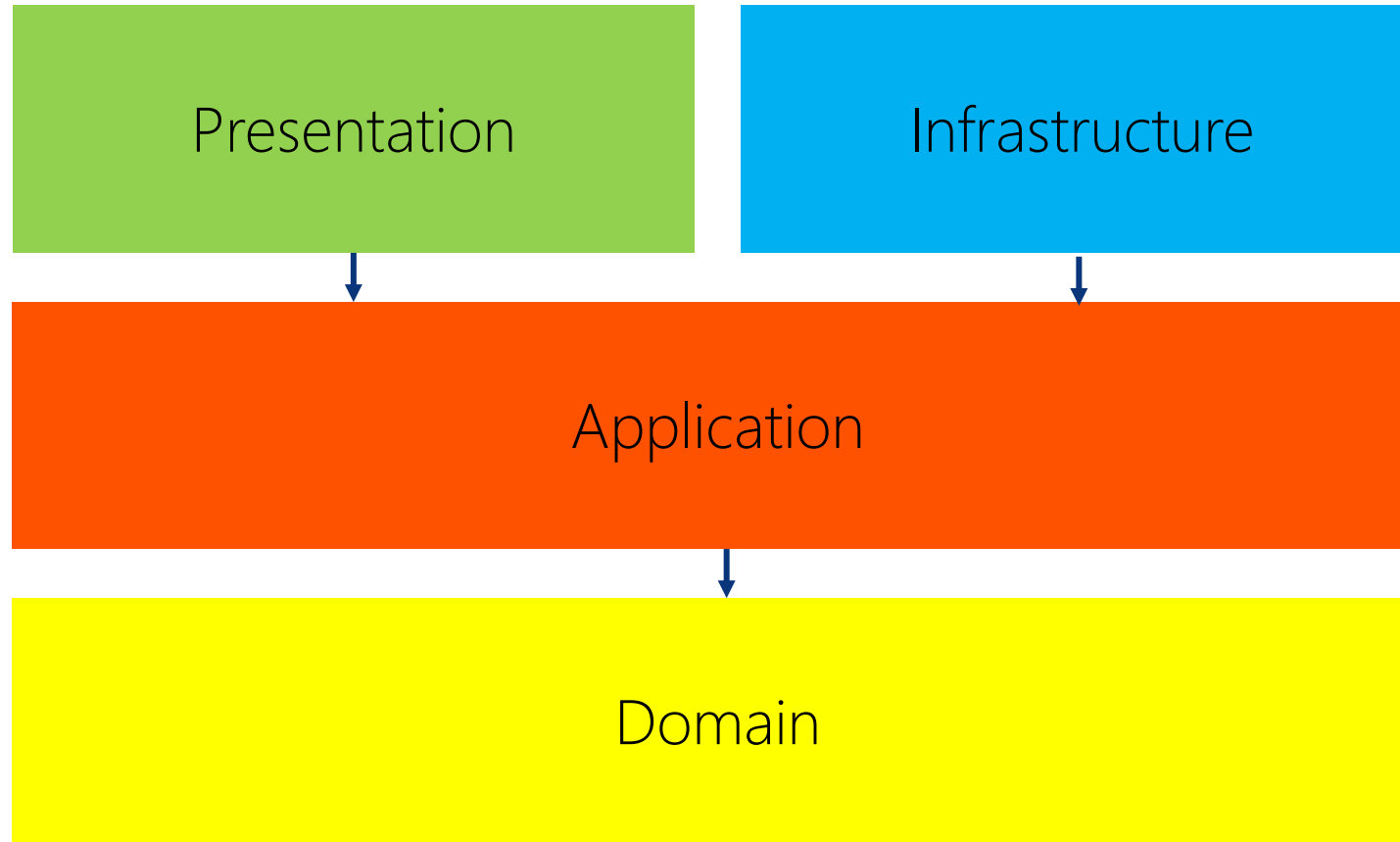
- ▶ Ports and Adapters a.k.a. Hexagonal Architecture
  - Alistair Cockburn (2005)
  - <https://alistaircockburn.us/hexagonal-architecture/>
- ▶ Onion Architecture
  - Jeffrey Palermo (2008)
  - <https://jeffreypalermo.com/2008/07/the-onion-architecture-part-1/>
- ▶ Clean Architecture a.k.a. Screaming Architecture
  - Robert C. Martin (2012)
  - <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>



# General Approach



# Clean Architecture



# Pros and Cons of Clean Architecture

## ▶ Pros

- Device independence
- Completely testable
- Domain focus

## ▶ Cons

- Harder to learn for developers
- Overkill for smaller services
- Onboarding



# Agenda

- ▶ Introduction
- ▶ The many names of Clean Architecture
- ▶ **Vertical Slice Architecture**
- ▶ Summary



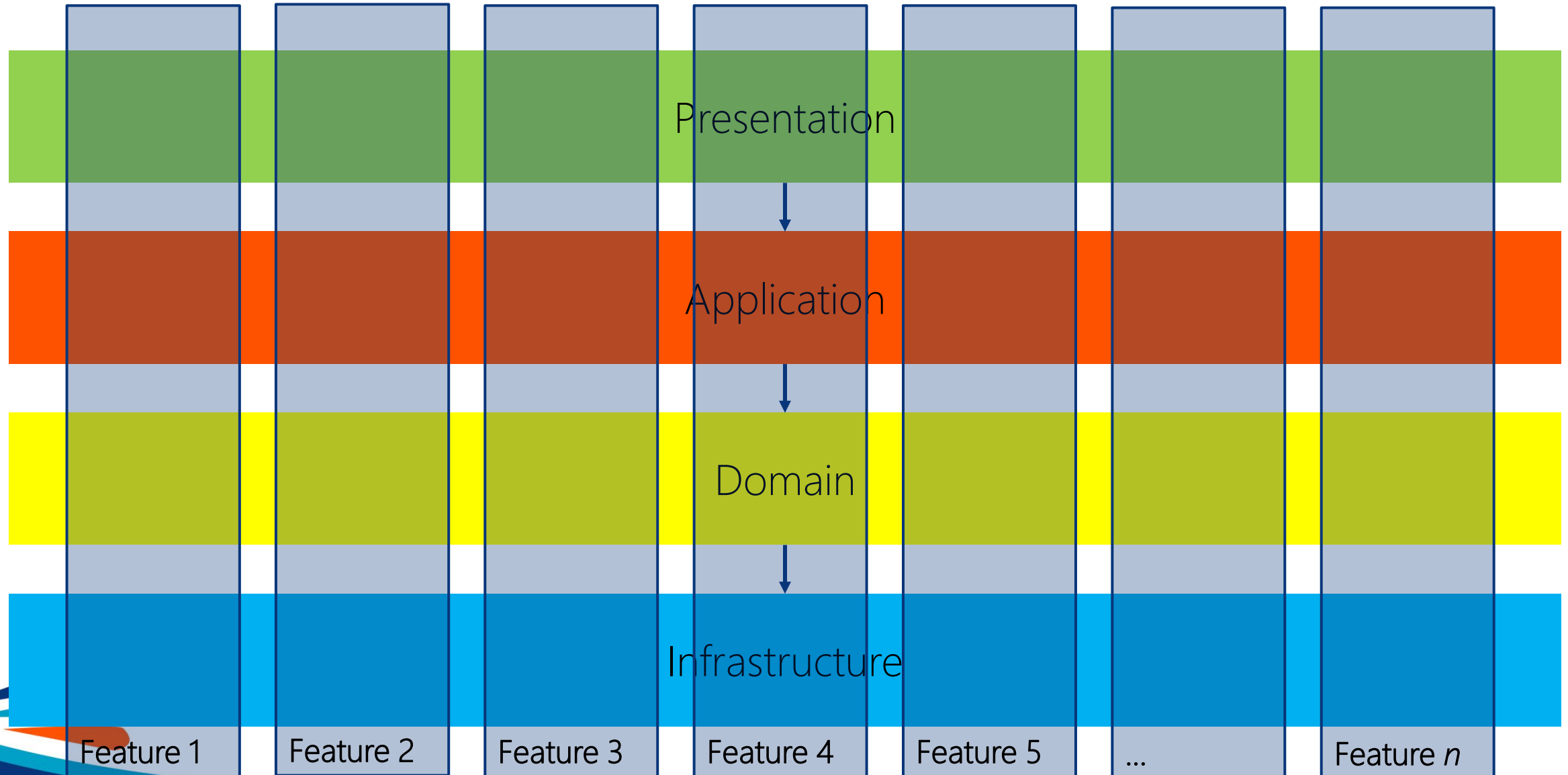
# What is Vertical Slices Architecture?

- ▶ Feature-centric architecture where each feature implementation has specific layers
- ▶ References between assemblies are "old school" layers
- ▶ Each feature implemented by a "slice"
  - is completely independent of other "slices"





# Vertical Slices Architecture



# Pros and Cons of Vertical Slice Architecture

## ► Pros

- Encapsulation
- Easier to learn for developers
- Isolation

## ► Cons

- Duplication
- Complexity
- Long-running tests



# So... Clean or Vertical Slice Architecture?!

- ▶ “Gradual” difference between the two
- ▶ Each slice is free to define structure as it sees fit
  - Some slices could be Clean Architecture!
- ▶ Choice might depend upon
  - Project size
  - Team skill set
  - Time to market
  - Testability
  - Complexity



