

# Course Summary

---



**Dan Wahlin**

WAHLIN CONSULTING

@danwahlin [www.codewithdan.com](http://www.codewithdan.com)



# Angular Style Guide Highlights



**Coding conventions**

**Naming rules**

**Application structure**

**Organizing modules**

**Creating and using services**

**Lifecycle hooks**

# Organizing Features



**Use a feature-based approach to structure your code**

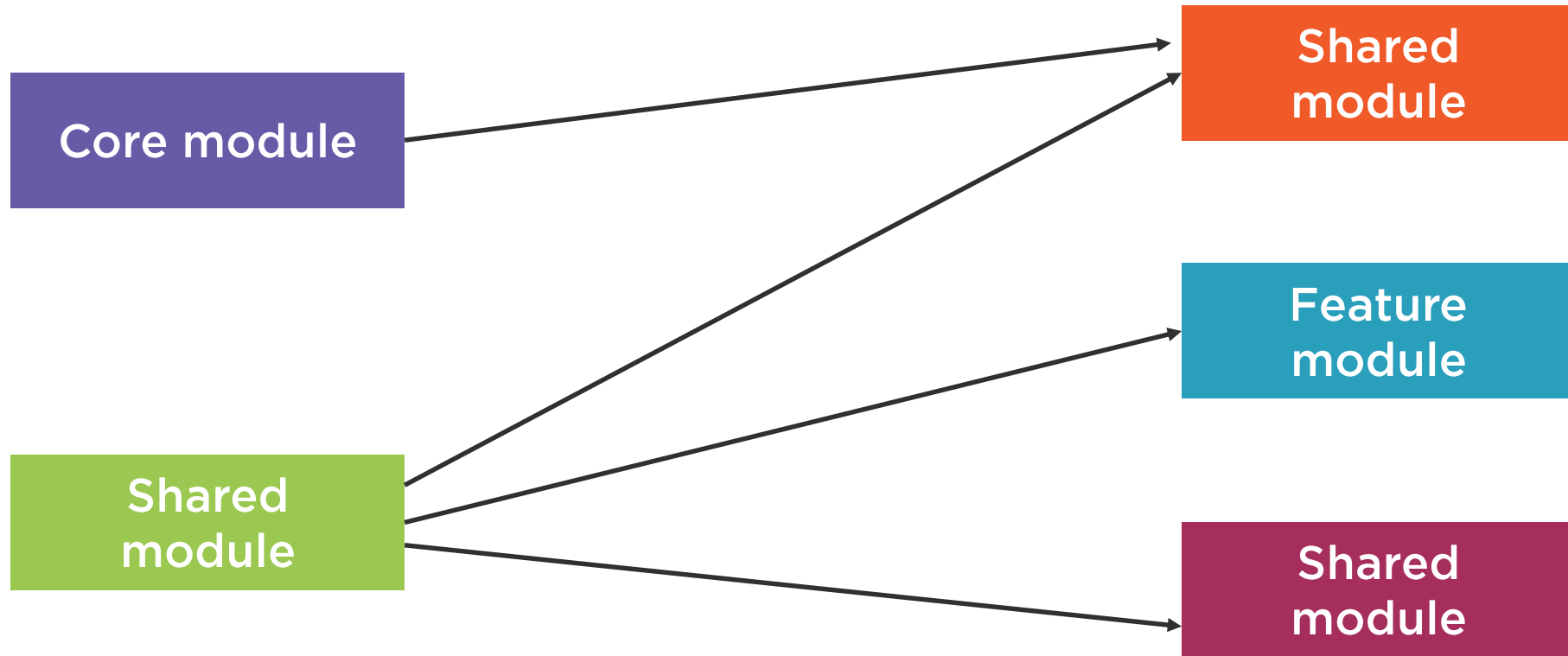
**Use the Angular CLI to generate initial feature folder/component**

**Minimum of one module per feature (as appropriate)**

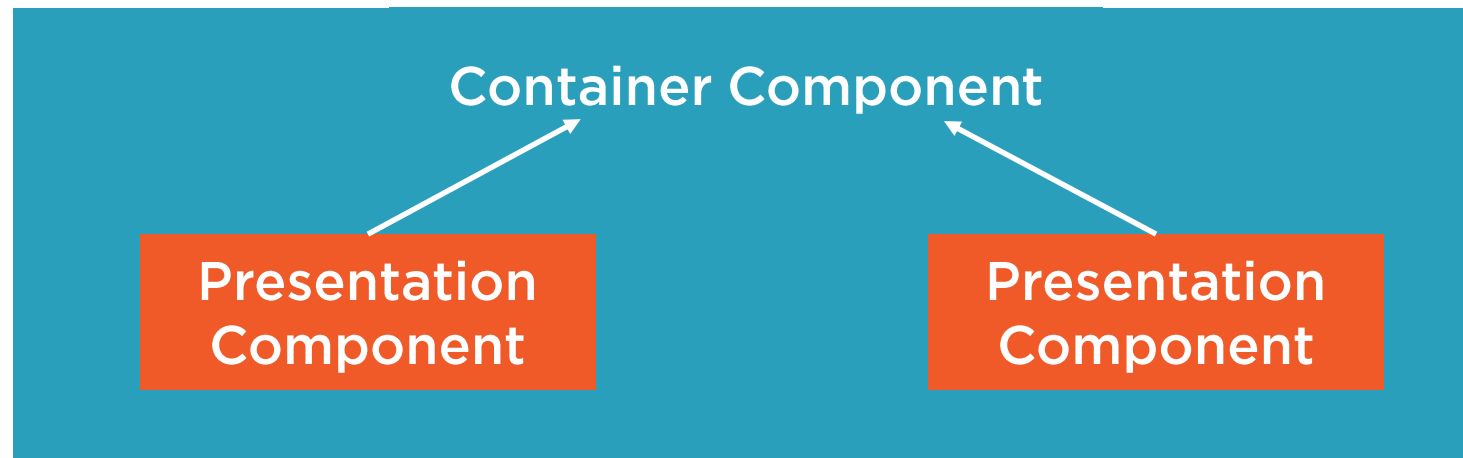
**Avoid deeply nested folders**

**Flatten feature hierarchies**

# Importing Core and Shared



# Container and Presentation Components

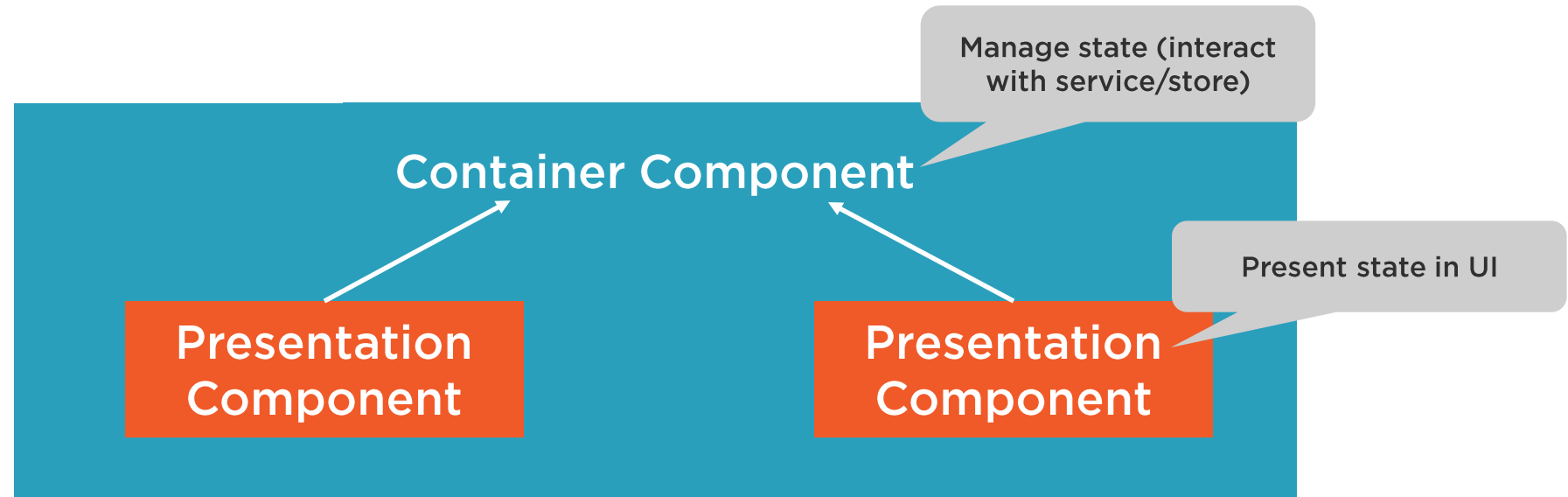


**Container component, retrieves state**

**Presentation component, presents/renders state**



# Container and Presentation Components

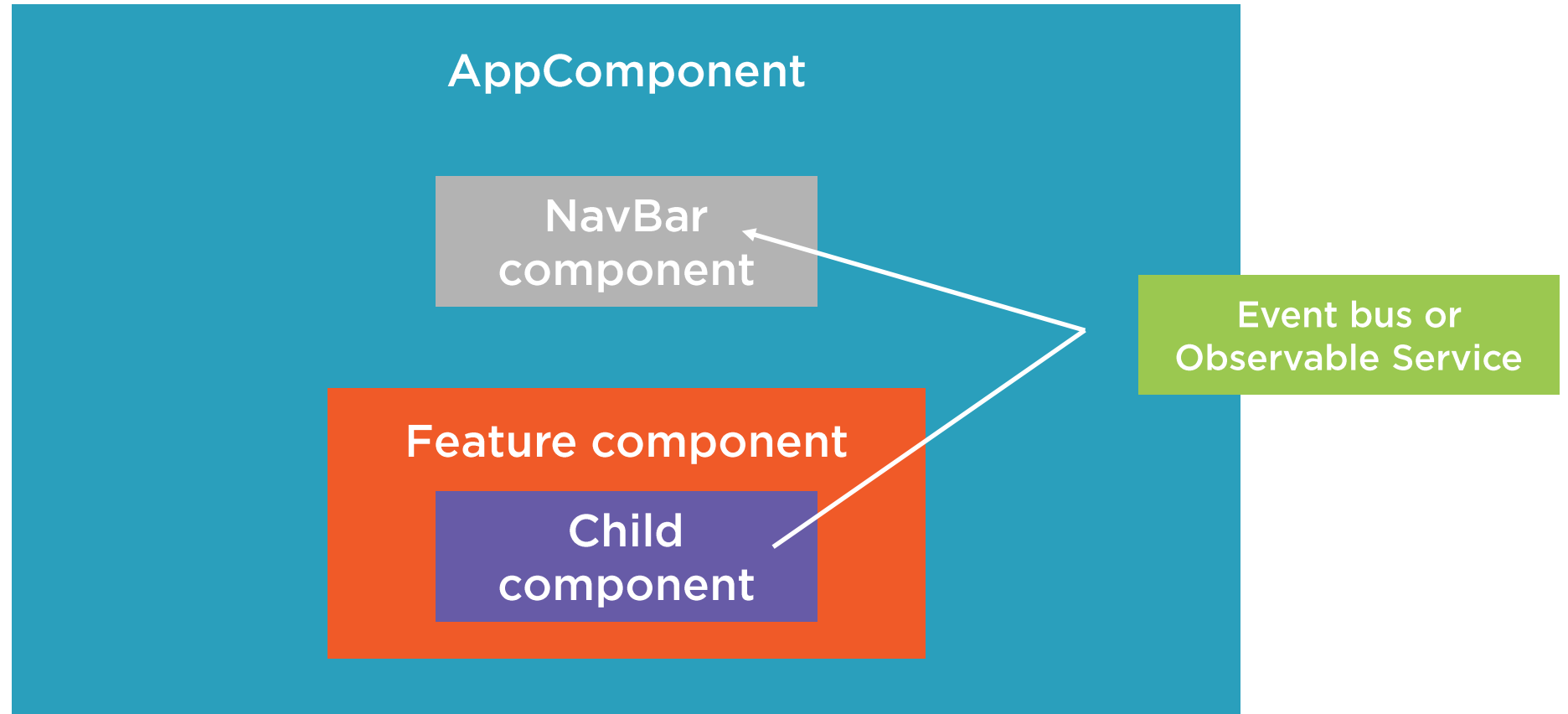


Container component, retrieves state

Presentation component, presents/renders state



# Component Communication Options



# Event Bus vs. Observable Service

## Event Bus

### Mediator Pattern

Angular service acts as the middleman between components

Components don't know where data is coming from by default

Loosely coupled

Relies on subject/observable

## Observable Service

### Observer Pattern

Angular service exposes observable directly to components

Components know where data is coming from

Not as loosely coupled as event bus

Relies on subject/observable





# Comparing State Management Options

Goal	NgRx	ngrx-data	Observable Store	Services
Provides store	✓	✓	✓	✗
Immutable data	✓	✓	✓	✗
Debugging	✓	✓	✗	✗
State history	✓	✓	✓	✗
Change notifications	✓	✓	✓	✗
Simple	✗	✓	✓	✓



Thank you!

