ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Angular Testing

5 - Testing Strategies

Two Competing Schools of Unit Testing

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Differences

London ~ Unit Test

- Unit is a **class**
- Mock everything except the class
  - Very tightly coupled to implementation
- Disadvantages
  - No refactoring
  - Lots of code for mocking
  - No interplay testing
- Advantages
  - Edge cases, finding bugs, exploratory
  - Great code quality (FP)
  - Fast

Detroit (Chicago) ~ Integration Test

- Unit is a **behaviour**
- Mock out-of-system dependencies
  - Runs against an API (UI)
- Advantages
  - Great for refactoring
  - Efficient (coverage)
- Disadvantages
  - Large setup required
  - Slow
  - Hard (Async, Change Detection, DOM,...)
  - Code Quality is of no concern

It is not Unit vs. Integration

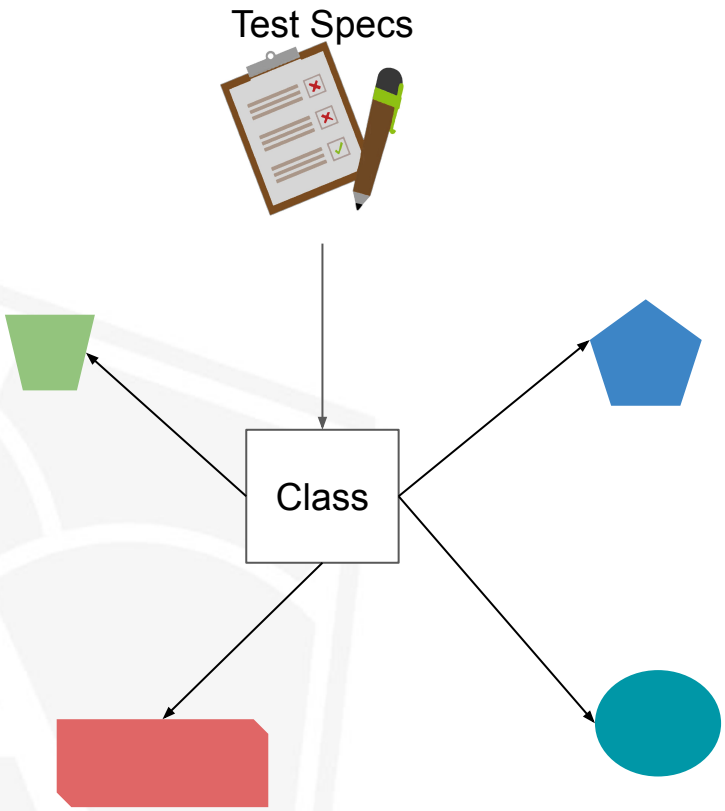
It is about the right balance

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Removing
Unit Tests???

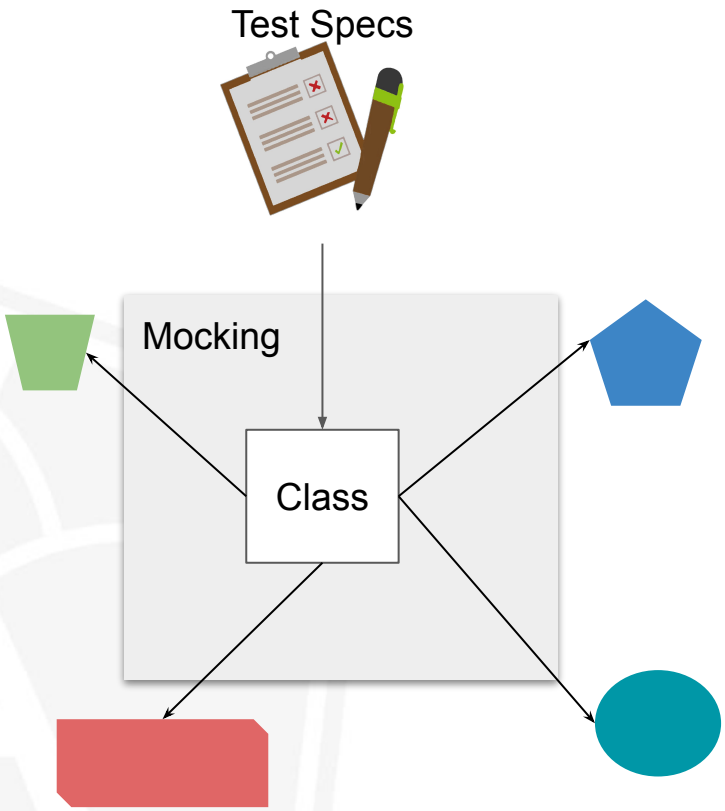
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

London Style



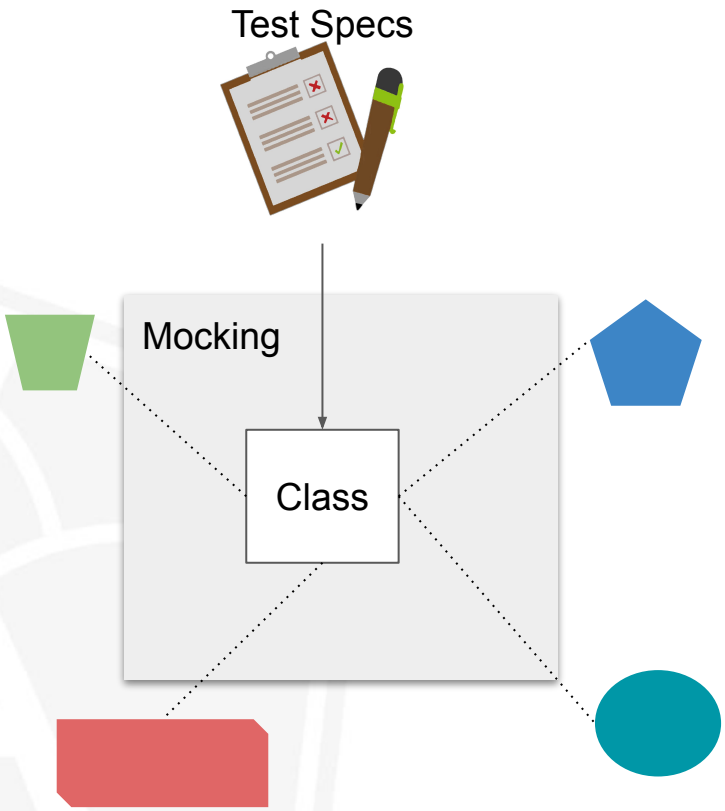
ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

London Style



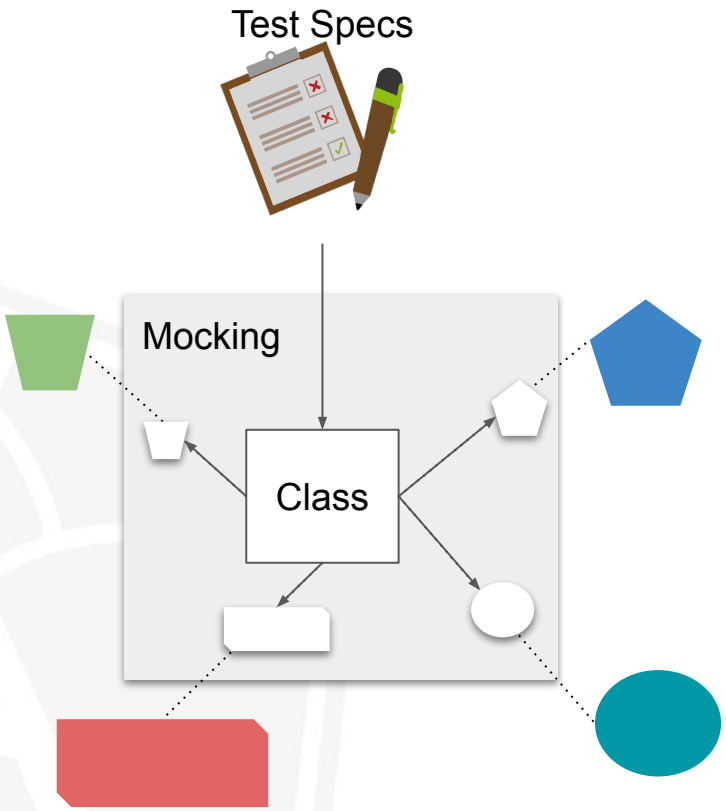
ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

London Style



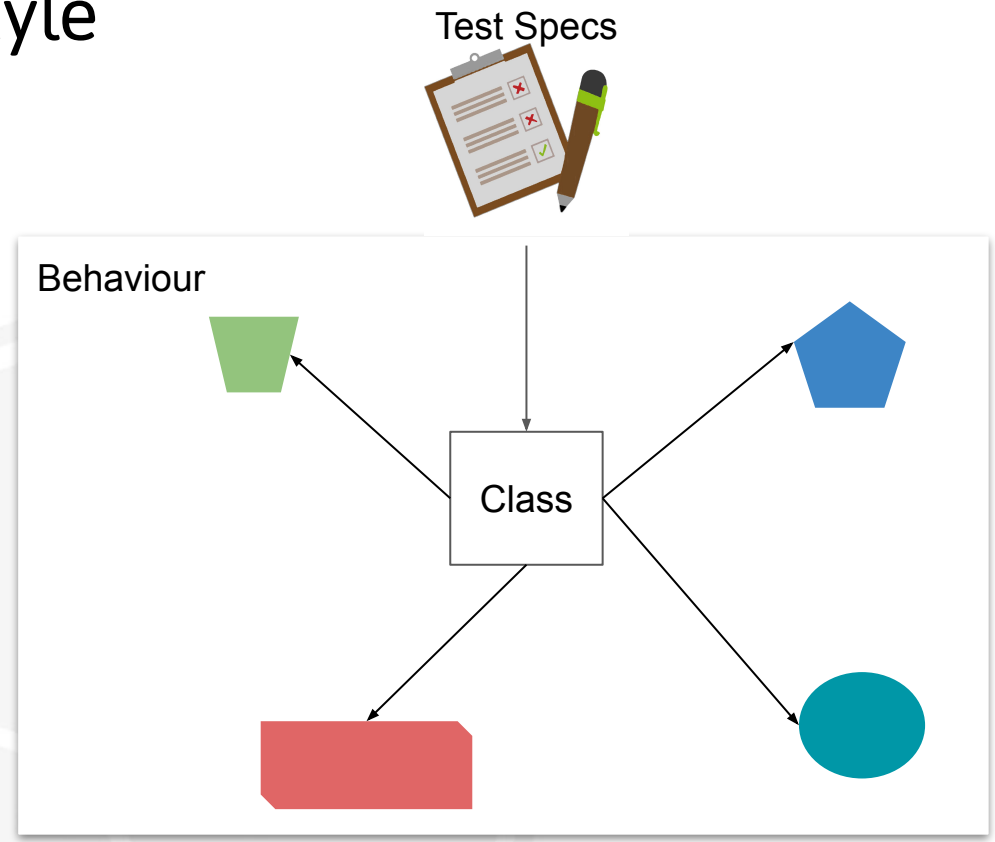
ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

London Style



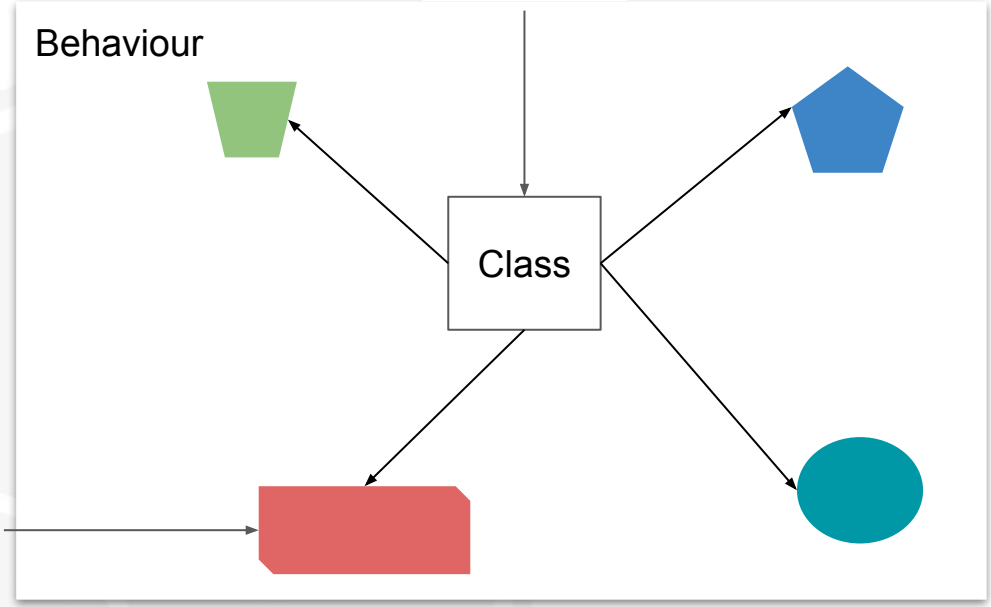
ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Detroit Style



ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

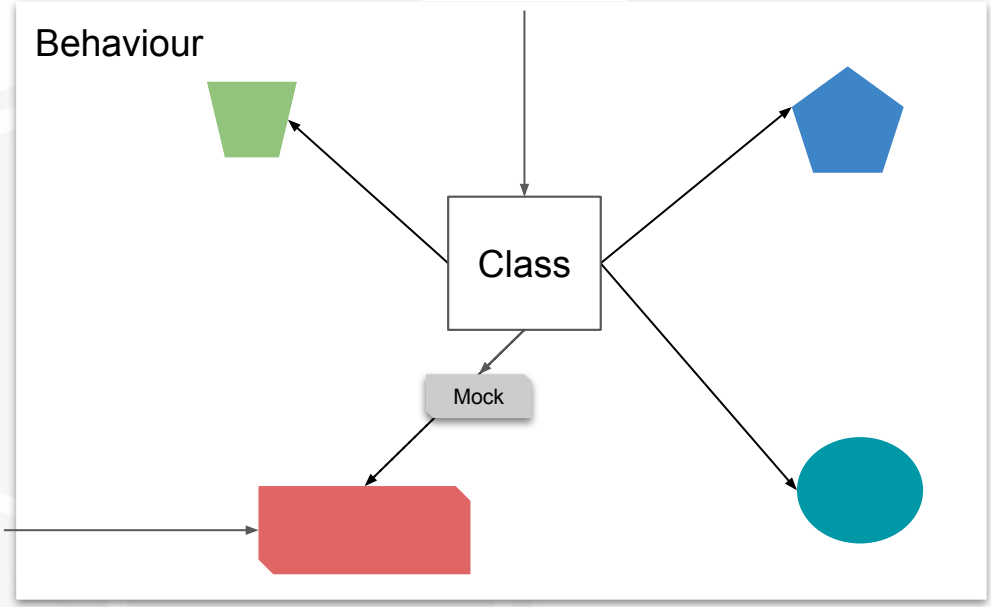
Detroit Style



Out-of-System Dependency
e.g. HttpClient

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

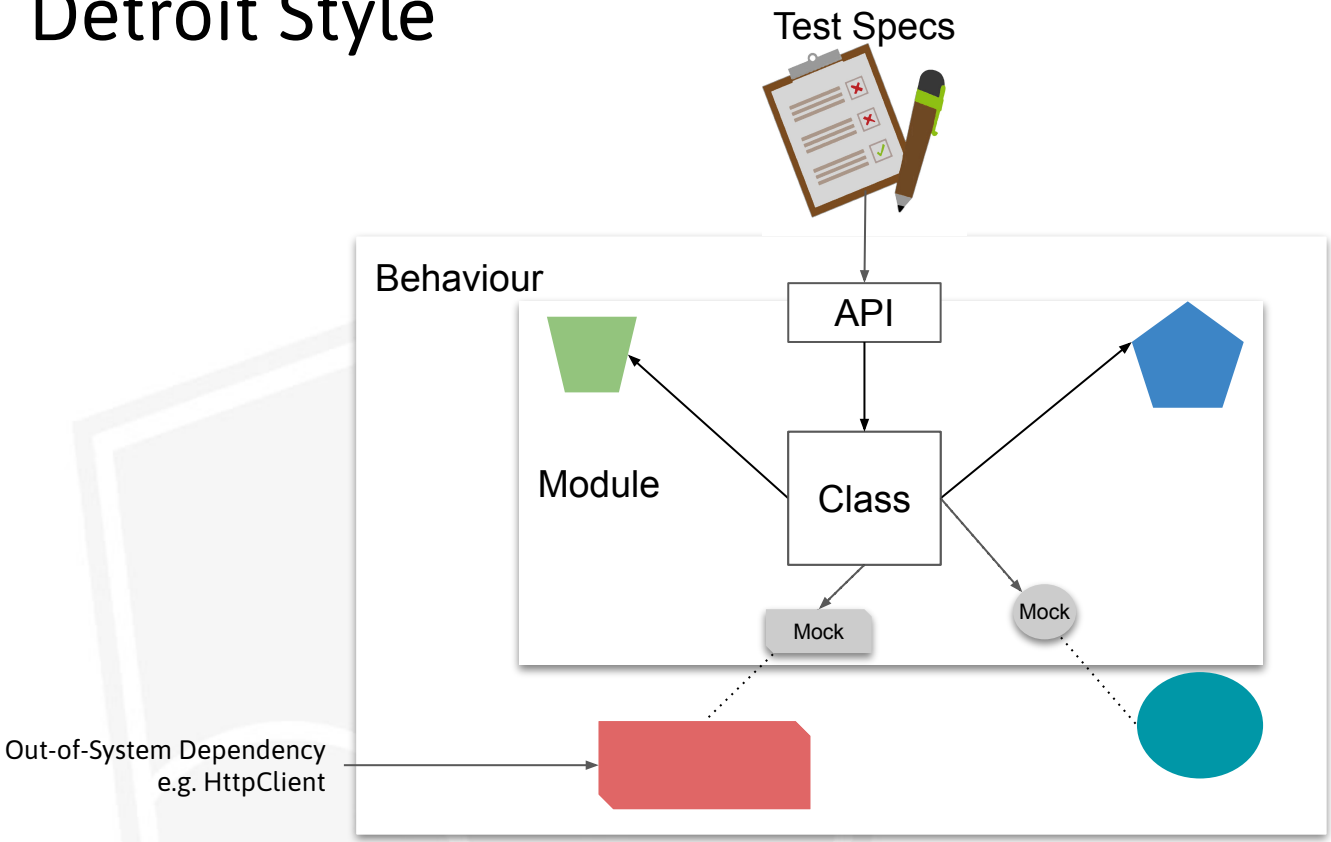
Detroit Style



Out-of-System Dependency
e.g. HttpClient

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Detroit Style



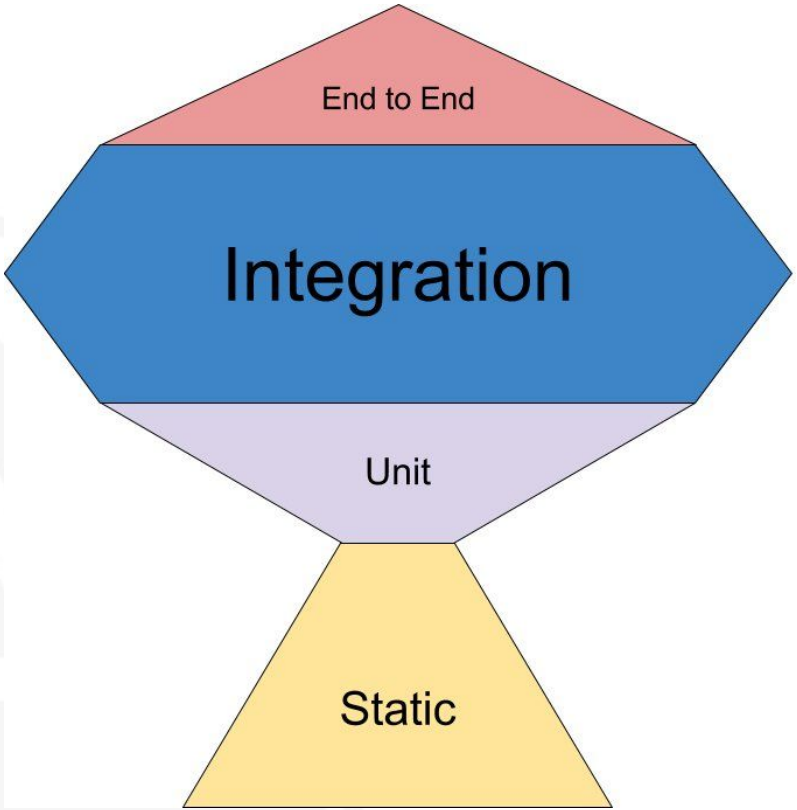
ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Criterias

- Speed
  - Execution
  - Writing & Maintaining
  - CI & Local Setup
- Timing
- Industry
- Effectiveness
- Application Type

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Testing According to ROI



Kent C. Dodds: https://twitter.com/kentcdodds/status/960723172591992832

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Application Types

1: Anemic

- Most parts of data processing (unit test) done in backend
- Frontend as "proxy" → less logic
- Integration is King

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

2: Autonomous

- Backend acts as Store
- Lots of Logic in Frontend
- Unit Tests & Integration Tests are critical

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

3: Complex UI

- ViewState in different variations
- Go for Component Tests

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

4: UI Library

- Library Vendor
- Storybook
  - Visual Regression
  - Cypress

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

5: Too big to Test

- "I can only run unit tests because of build time"
- Architectural Engine
- Split Logic and UI from each other
- Functional Style
- Example: RulesEngine for Workflow
  - Unit Tests for RulesEngine
  - Integration Tests for Execution into UI

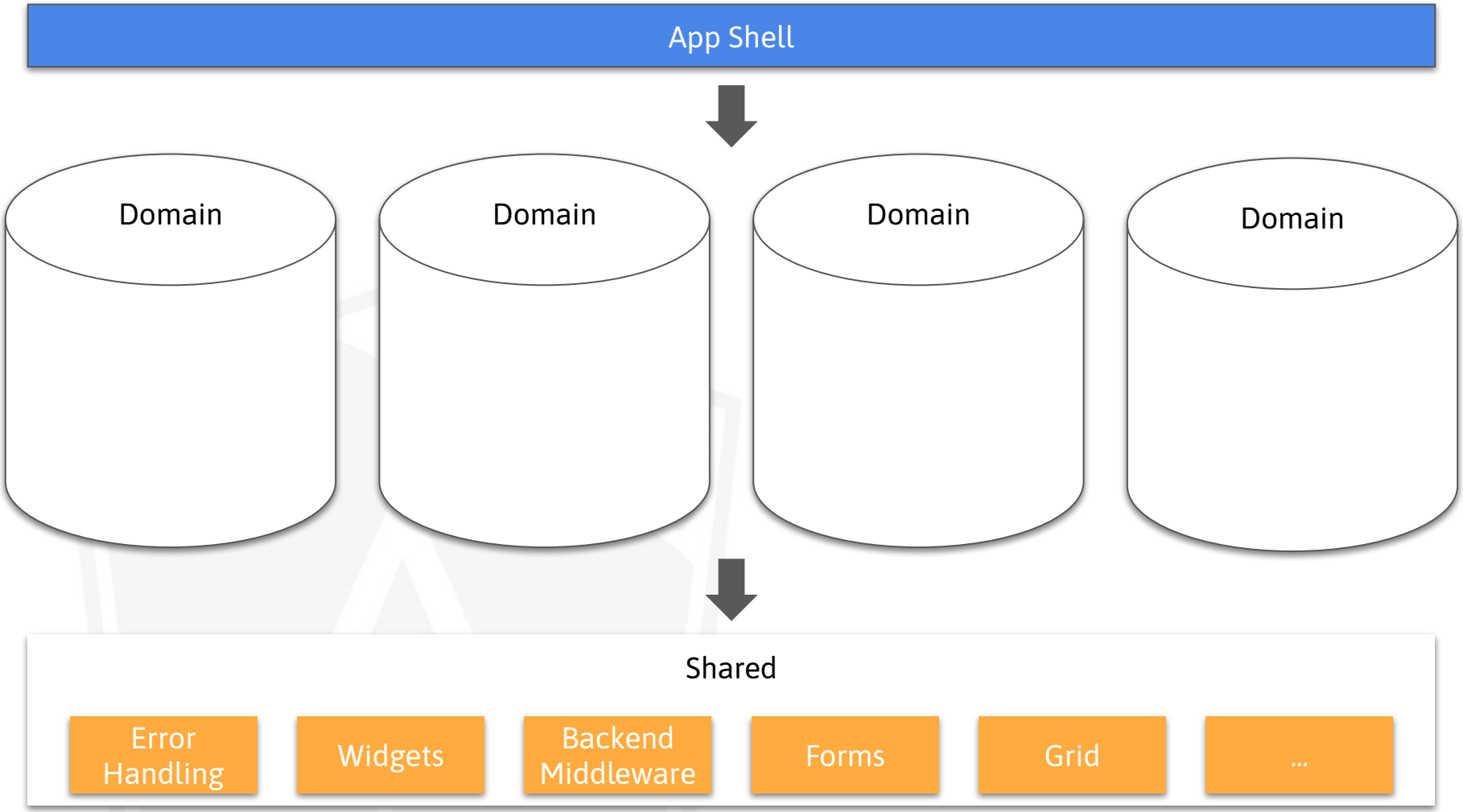
ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

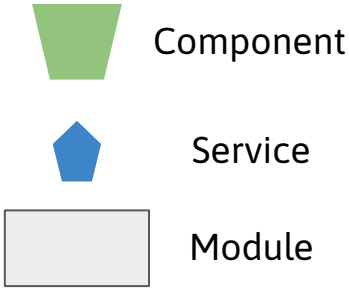
Trust your instincts!

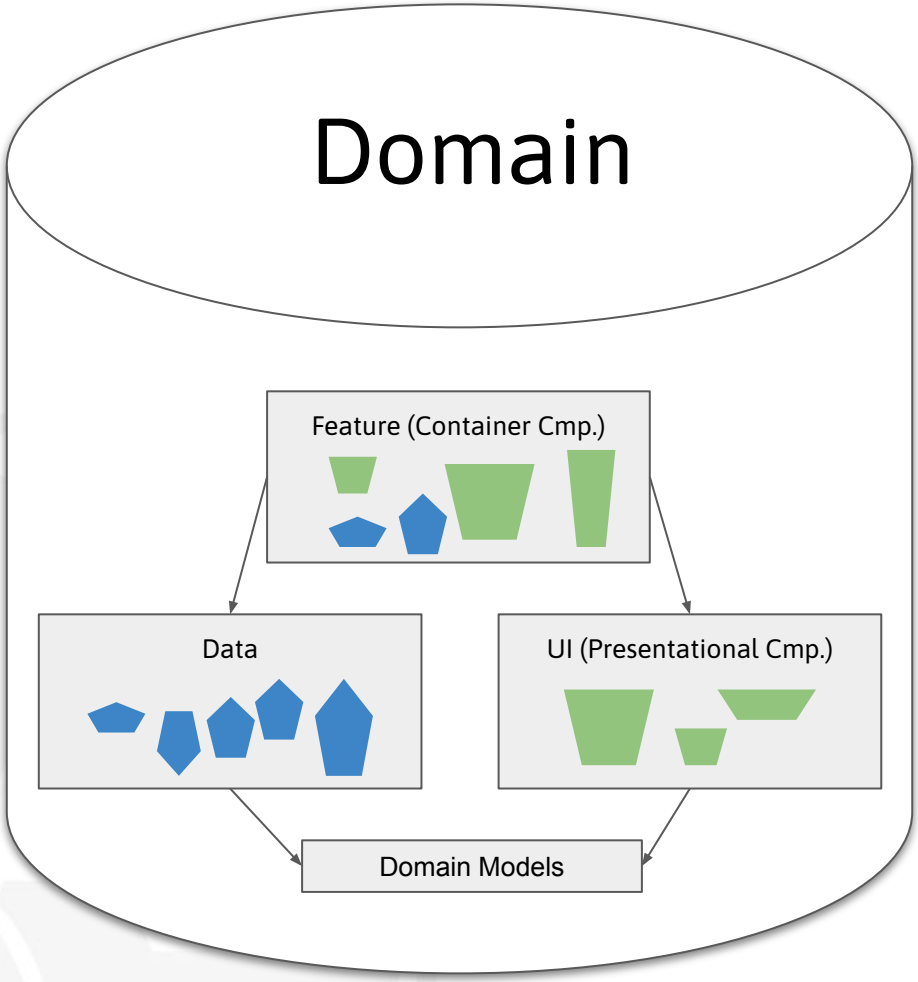
- *"It doesn't feel right"*
- *"What are we actually testing here? If a function is called? Really?!"*
- *"I don't see any value in testing."*
- *"I never discovered a bug with my tests."*
- *"I am wasting my time with writing tests instead of producing "real" code."*

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Testable Architecture


App Shell
Domain
Domain
Domain
Domain
Shared
Error Handling
Widgets
Backend Middleware
Forms
Grid
...





ANGULAR ARCHITECTS
INSIDE KNOWLEDGE