

Matt Vaughn
@angularlicious
matt@angularlicio.us

Angular Architecture

Take Angular
Architecture to the
CLEANers

“

What is
architecture?

Architecture Examples



EFFECTIVE ARCHITECTURE

- ▶ Experience
 - ▷ Knowledge, understanding
 - ▷ Analysis, design, planning
 - ▷ Technical Leadership
- ▶ Essentials
 - ▷ Tools and materials
- ▶ Execution
 - ▷ Plan, recipes
 - ▷ Guard the code

“

Why does
architecture
matter?



CLEAN ARCHITECTURE

Use CLEAN code and architectural approaches.

CLEAN CODE

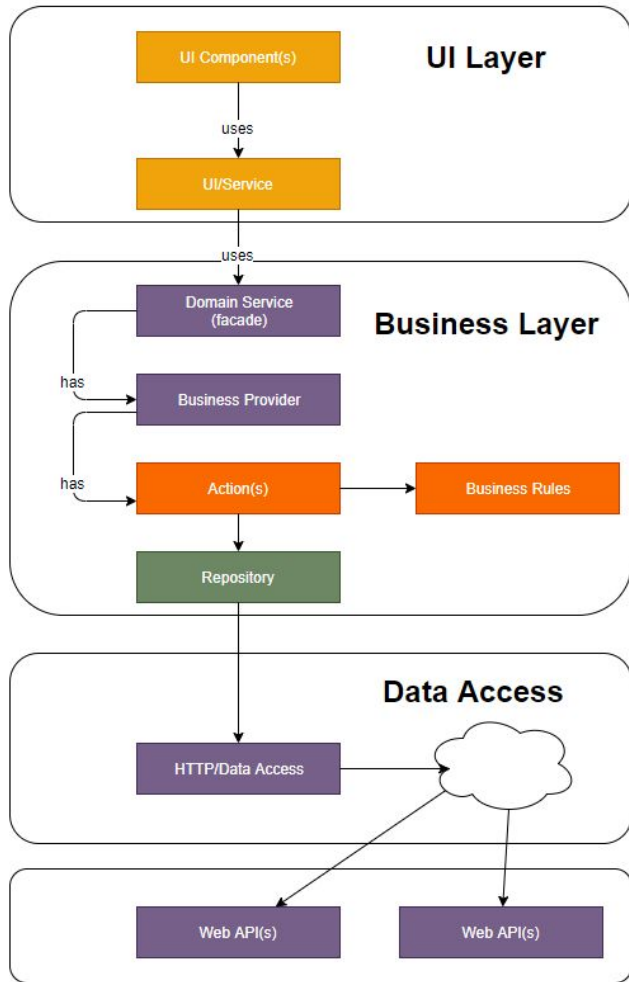
- ▶ Separation of Concerns
- ▶ Single Responsibility
- ▶ Clear boundaries between layers
- ▶ DRY Code
- ▶ Layered Architecture
- ▶ Design Patterns
- ▶ Object Oriented Practices

CLEAN ARCHITECTURE

- ▶ Layers
- ▶ Boundaries
- ▶ Inversion of Control (DI)
- ▶ *Testable*
- ▶ *Independent (frameworks, UI, database)*

Architecture HISTORY

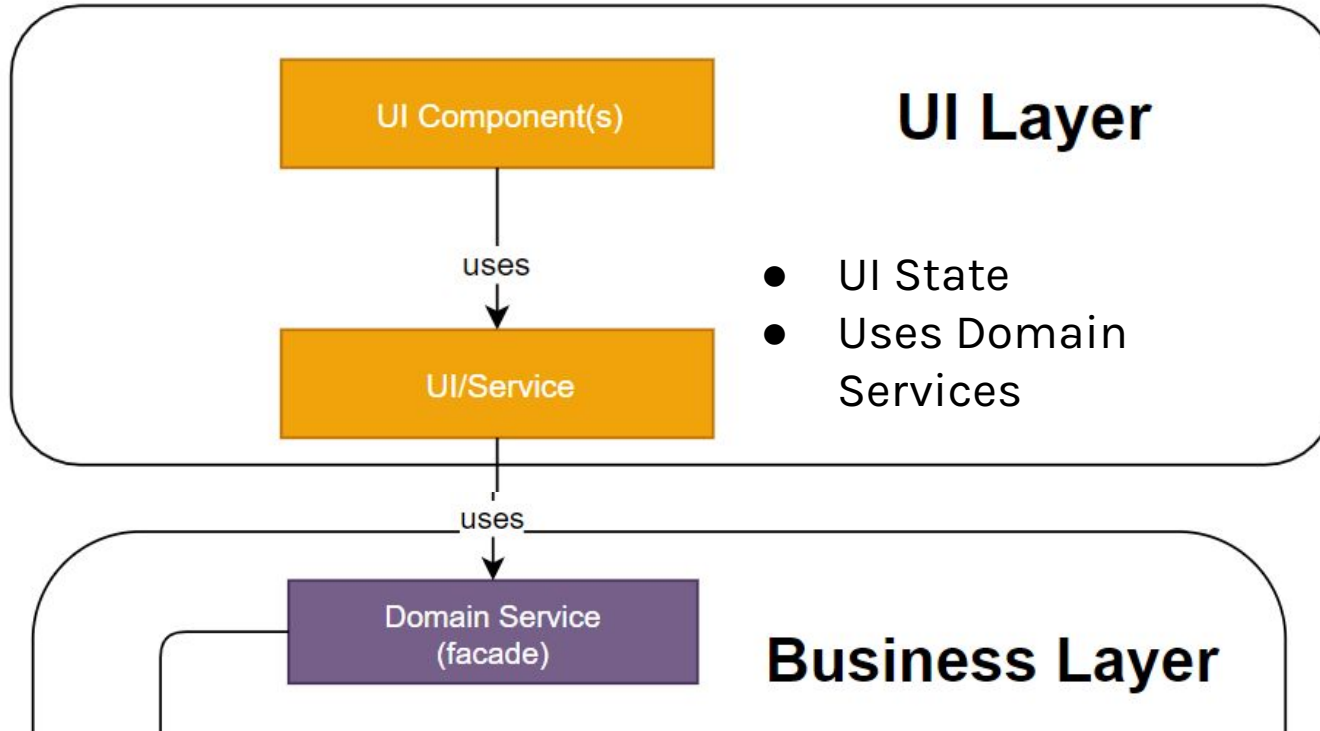
- ▶ Layers (n-tier)
- ▶ Hexagonal
- ▶ Clean
- ▶ Onion



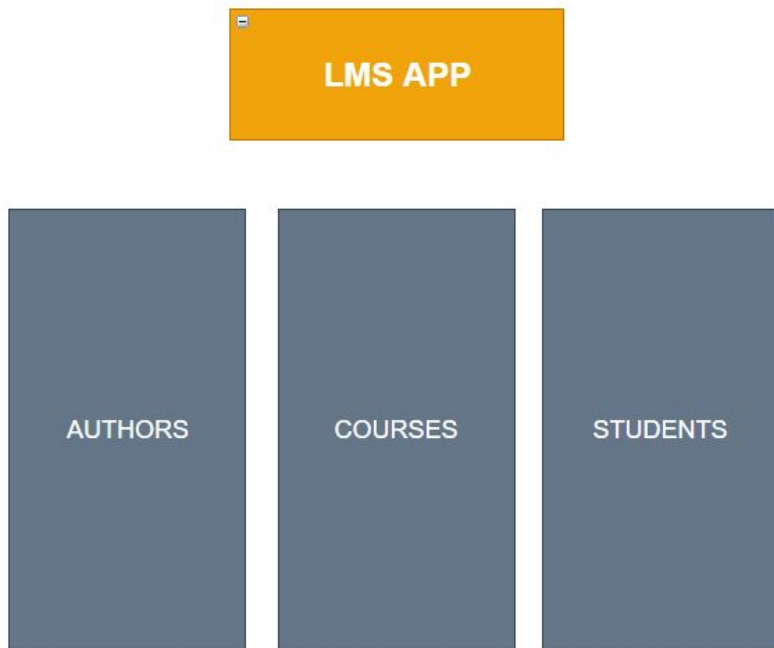
LAYERS SEPARATE CONCERNS

- ▶ UI Concerns
- ▶ Domain Services
- ▶ Business Logic Concerns
 - ▷ Actions
 - ▷ Rules
- ▶ Data Repository Concern
- ▶ Data Provider Concern/HTTP

UI CONCERNS

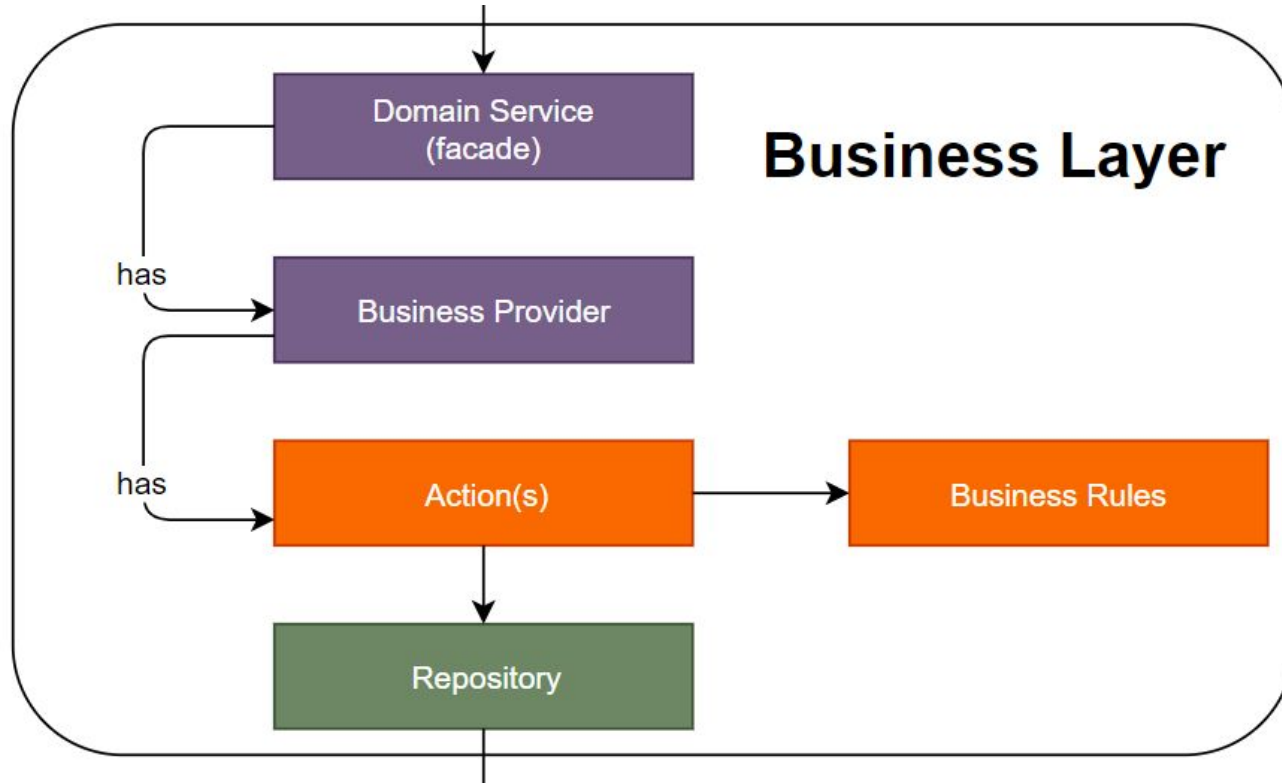


DOMAIN SERVICE CONCERNS

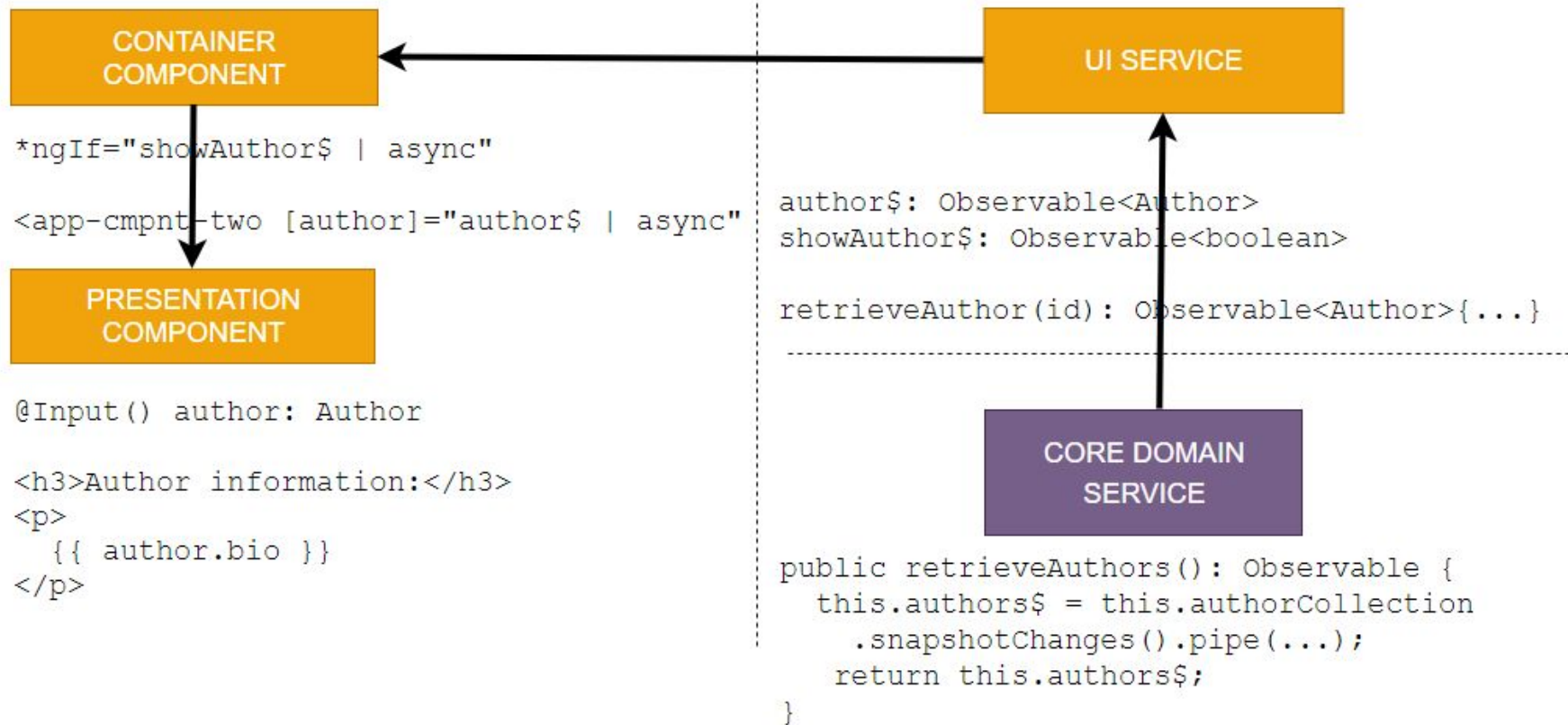


CORE DOMAIN LIBRARIES

DOMAIN SERVICE CONCERNS



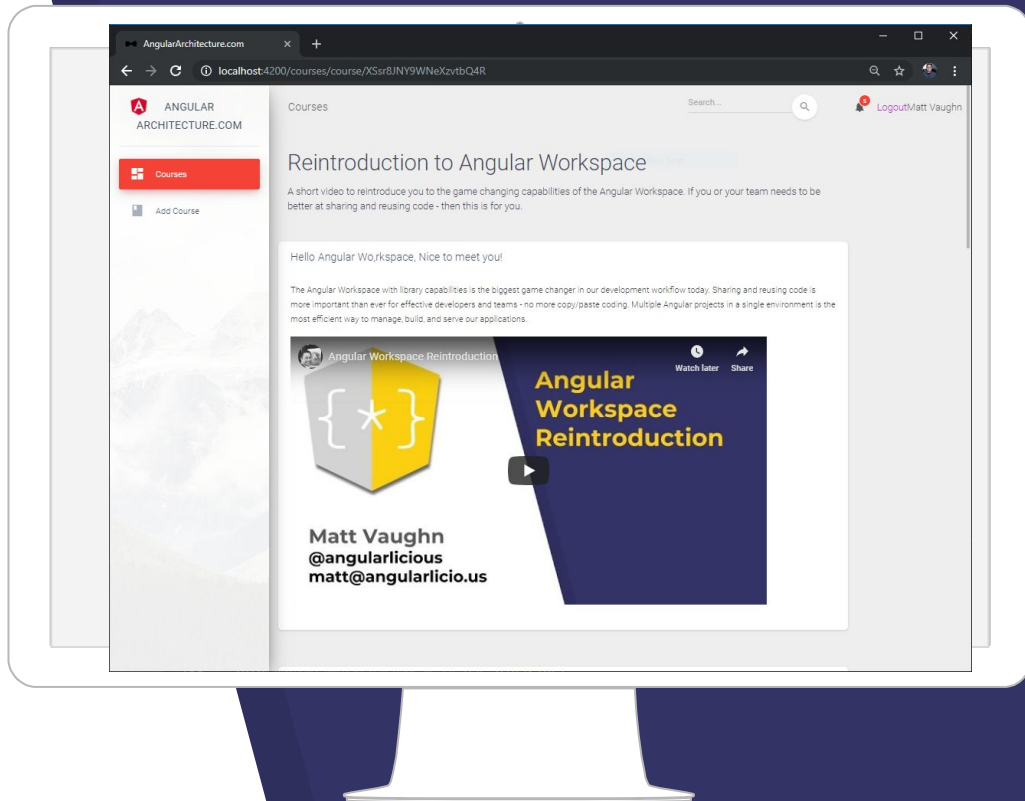
ONE-WAY DATA



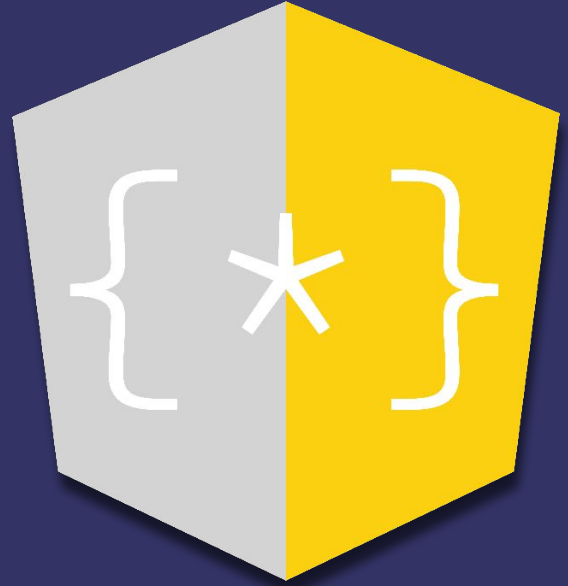


CODE REVIEW

Let's review a workspace with an application and several library projects.



Using Libraries



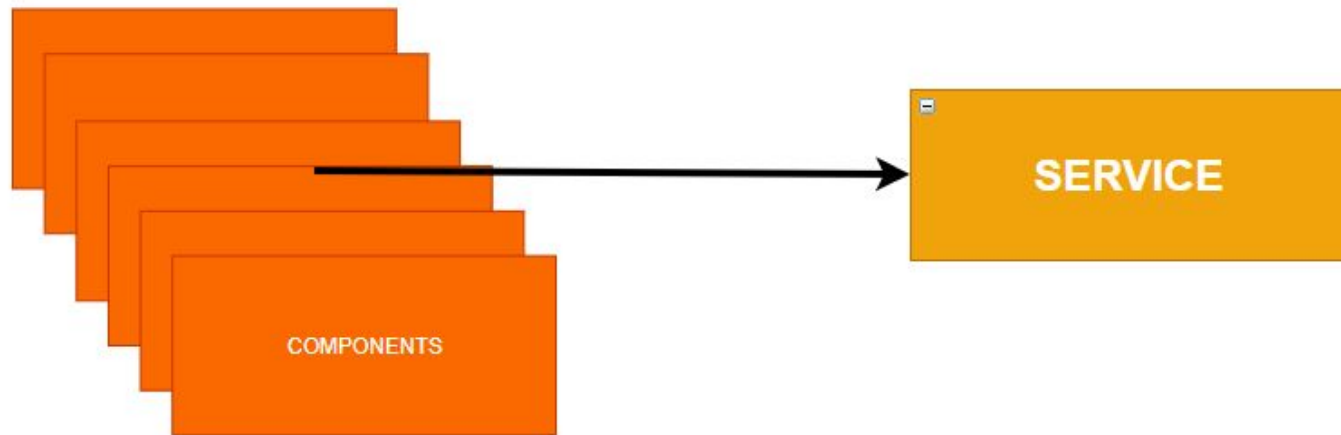
Is your Service OVERWORKED?



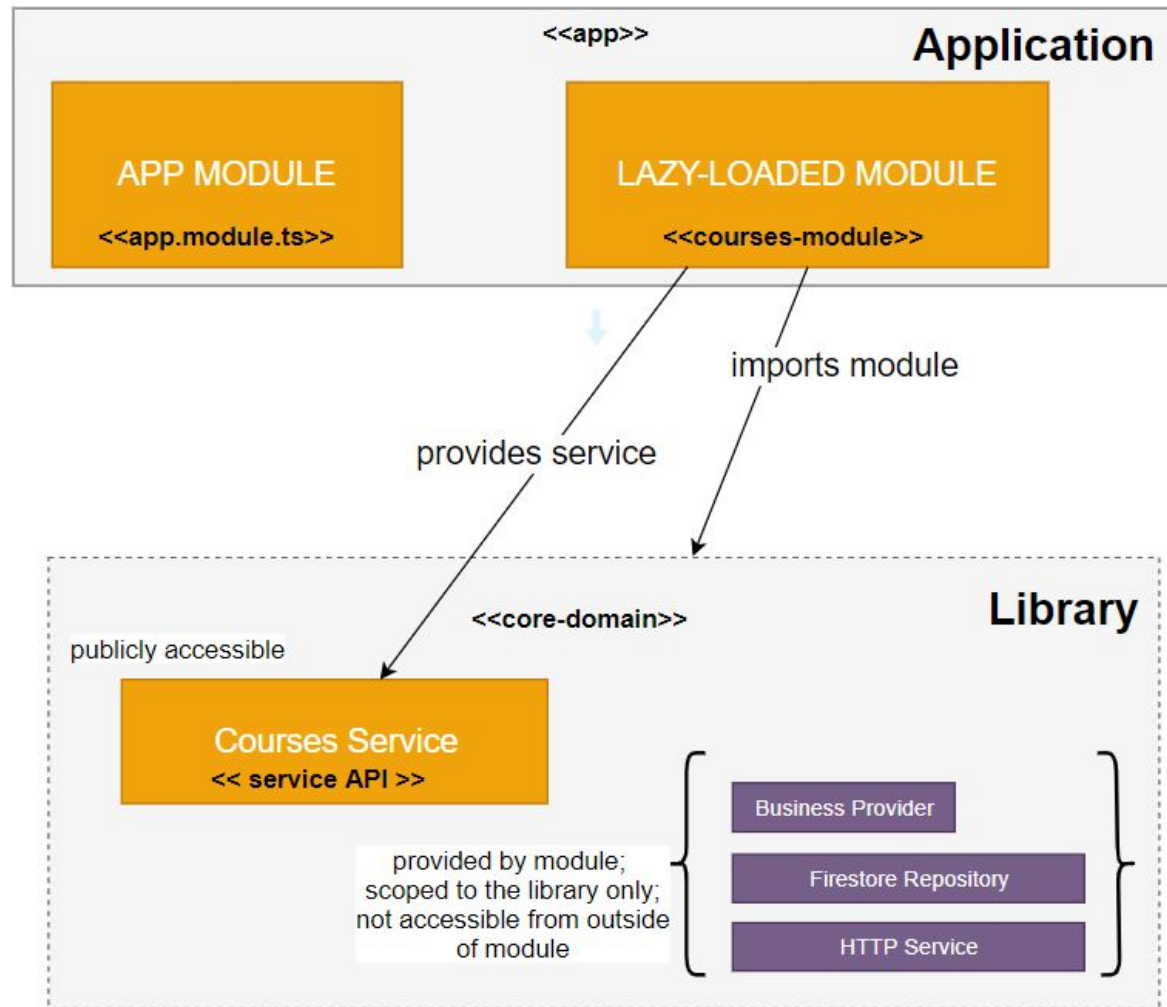
© BCCL 2019. ALL RIGHTS RESERVED.

SERVICE OVERLOAD

CODE...



Many services are loaded down because they have to manage state, provide API-like methods, perform business logic, validate input, process business rules, make HTTP requests, handle responses, handle HTTP errors, log events, etc.



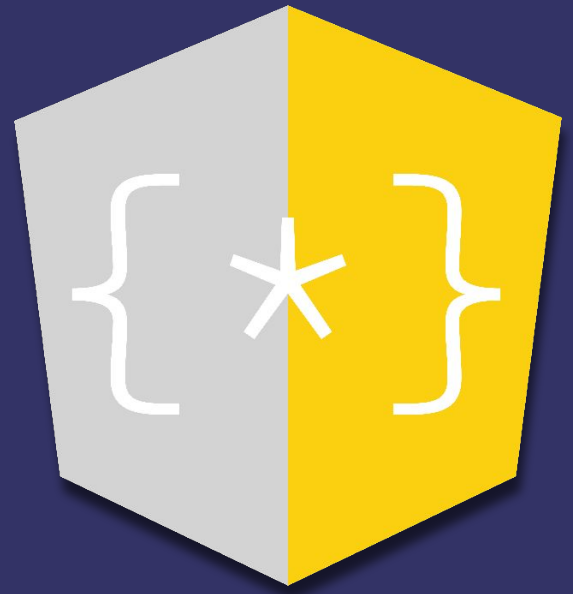
CUSTOM LIBRARIES

- Single Library Projects
- Angular Package Format
- Publishing and Consuming
- Versioning
- Not an efficient developer workflow

WHY LIBRARIES?

- Enables code reuse
- Maintenance
- Code Organization
- Publish and share

Library Types





DIFFERENT **LIBRARY** PROJECT TYPES

Cross-Cutting Concerns

Use to provide concerns that cross the boundaries of the application/domain: logging, error handling, configuration, etc.

Foundational

Use Object-Oriented programming to provide base classes for Angular Services, Components, Business Actions.

Components

A library for common components, pipes, and directives.

Feature Library

Use a library to encapsulate a feature library that can be used by multiple apps (security, shopping cart).

Core/Domain Library (Business Logic Layer)

Use a library to implement rich business logic layers that are accessed by service facades.

Framework

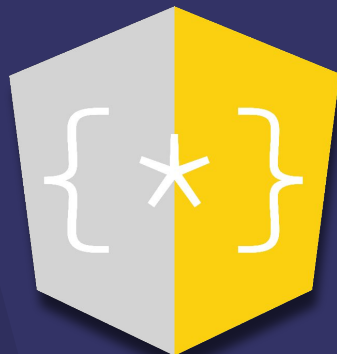
Create reusable frameworks to provide consistent implementation: rule engine, validators, business actions.

ANGULAR ARCHITECTURE

A Guide for Enterprise
Angular Applications



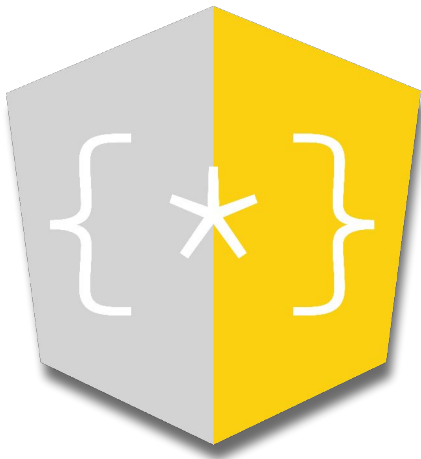
MATT VAUGHN



Free book for Denver Dev Day at:

BOOK: <http://bit.ly/DDD-ANGULAR-ARCHITECTURE>

Web: www.leanpub.com



@angularlicious

Email: matt@angularlicio.us

Web: www.angularlicio.us

Blog: www.medium.com/@angularlicious

Github: www.github/angularlicious



RESOURCES

All of the resources for this mini-workshop are available at:

- **GitHub/angularlicious**
- **Repo:**
 - **denver-dev-day-2019**