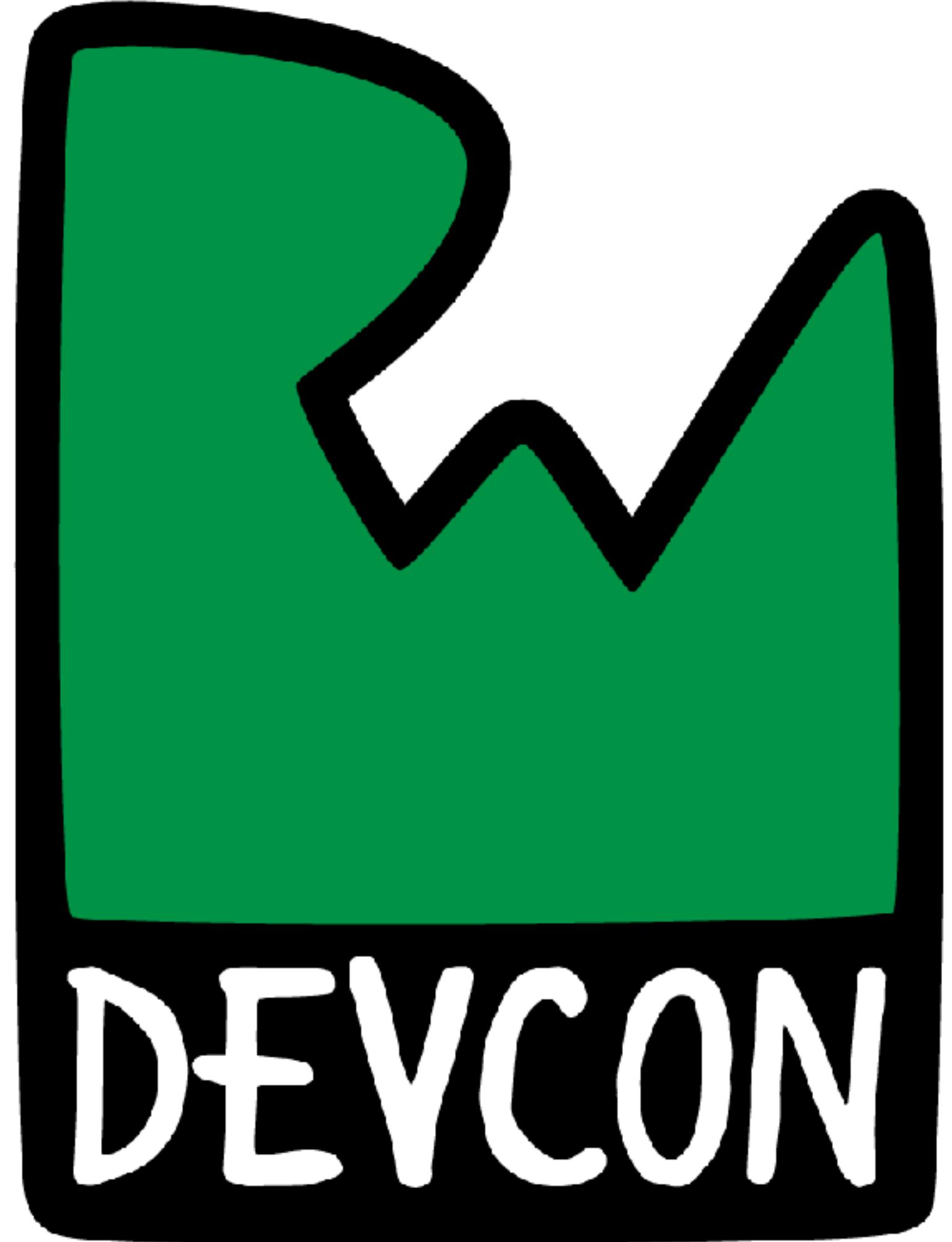


Session 3: Architecting Modules



WHAT IS A MODULE

OVERVIEW

- ⚙️ Part 1: What, Why, and How of Modules
- ⚙️ Part 2: Reuse and Performance
- ⚙️ Part 3: Insulating 3rd Party Modules





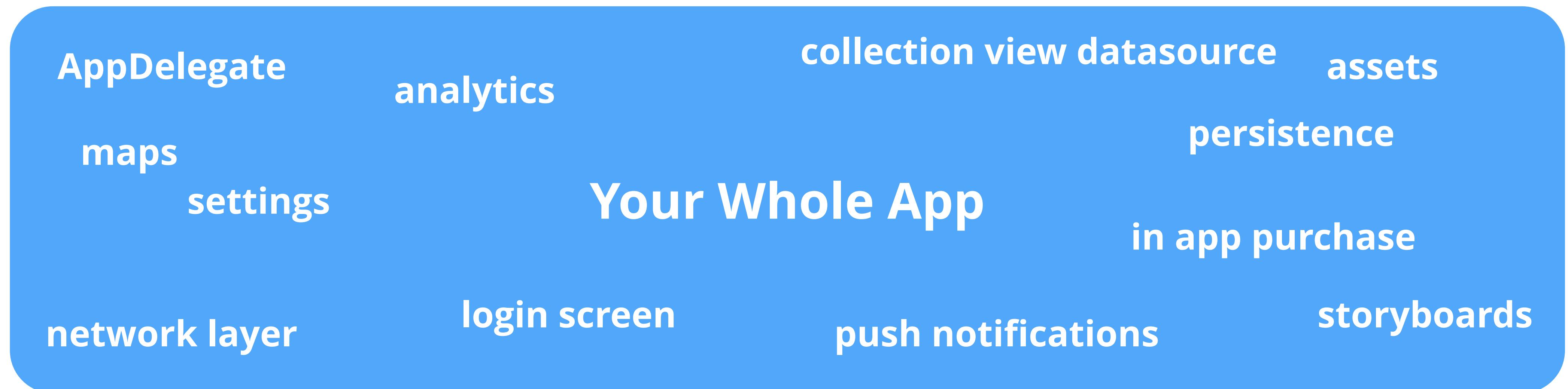
R
W

WHAT IS A MODULE?

“A *module* is a single unit of code distribution”



MODULE DIAGRAM

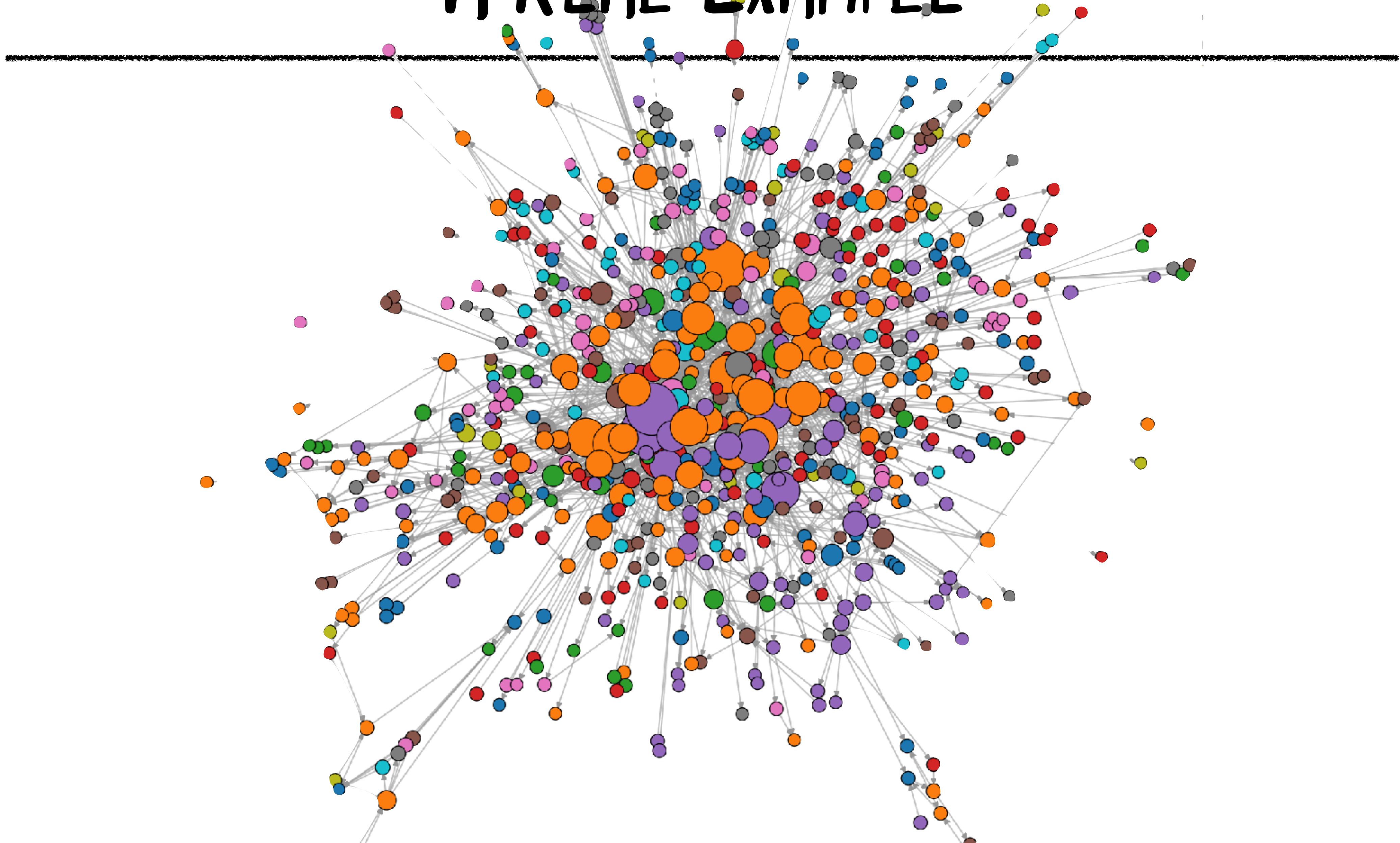


```
@testable import YourWholeApp
```

Unit Tests



A REAL EXAMPLE



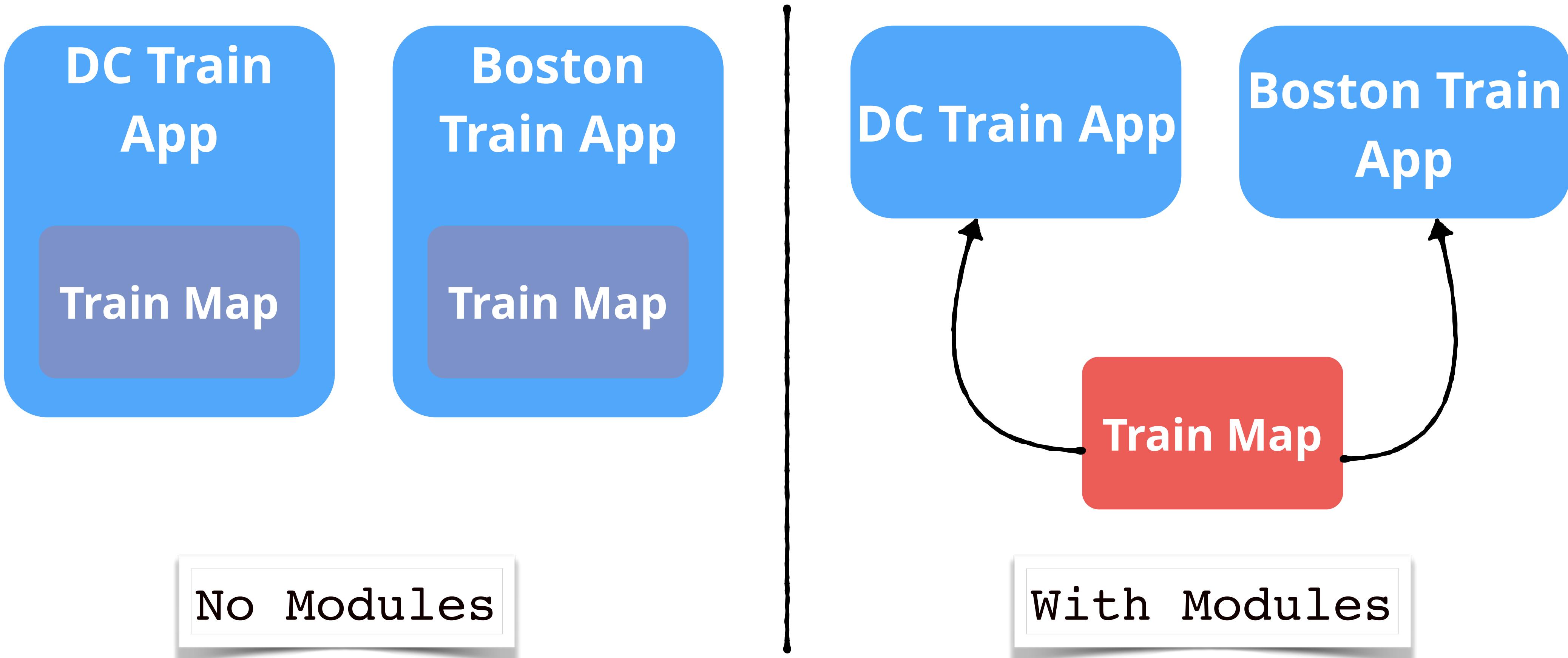
R
W

THE “ILITIES”

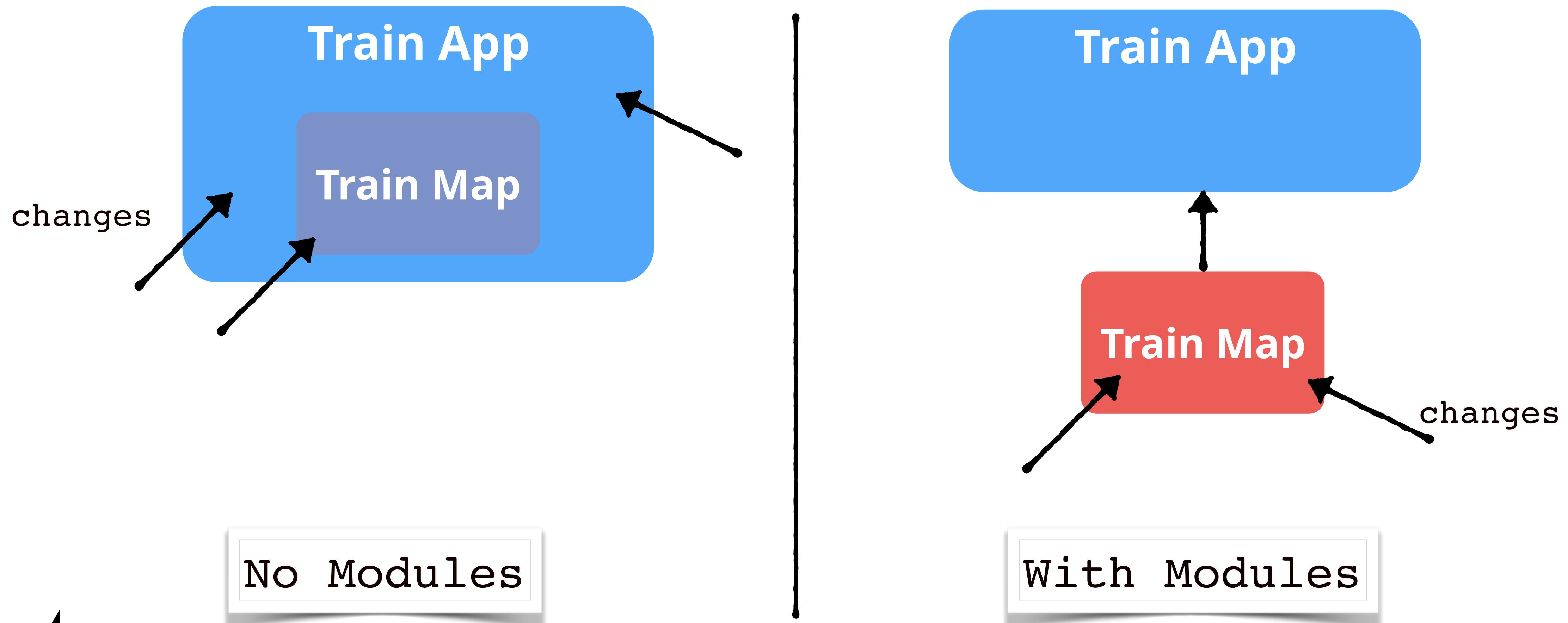
- ⚙️ Reusability
- ⚙️ Maintainability
- ⚙️ Readability
- ⚙️ Testability
- ⚙️ Scalability



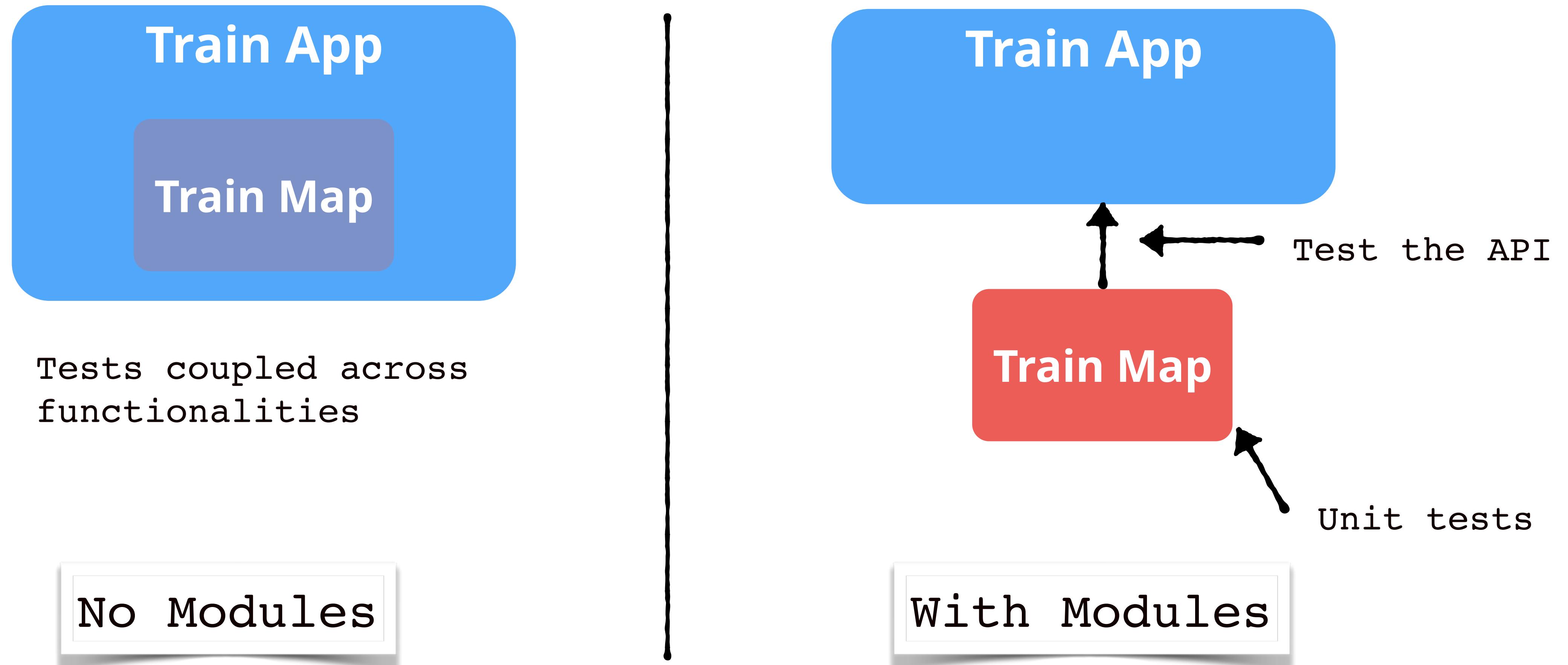
REUSABILITY



MAINTAINABILITY



TESTABILITY



How To MODULE

Transportation App

Interface

Model

Network

Demo 1



How To MODULE

Transportation App

Service
Map

Ticketing

Scheduling

Demo 2



How To MODULE

Transportation App

Monetization
Team

API Team

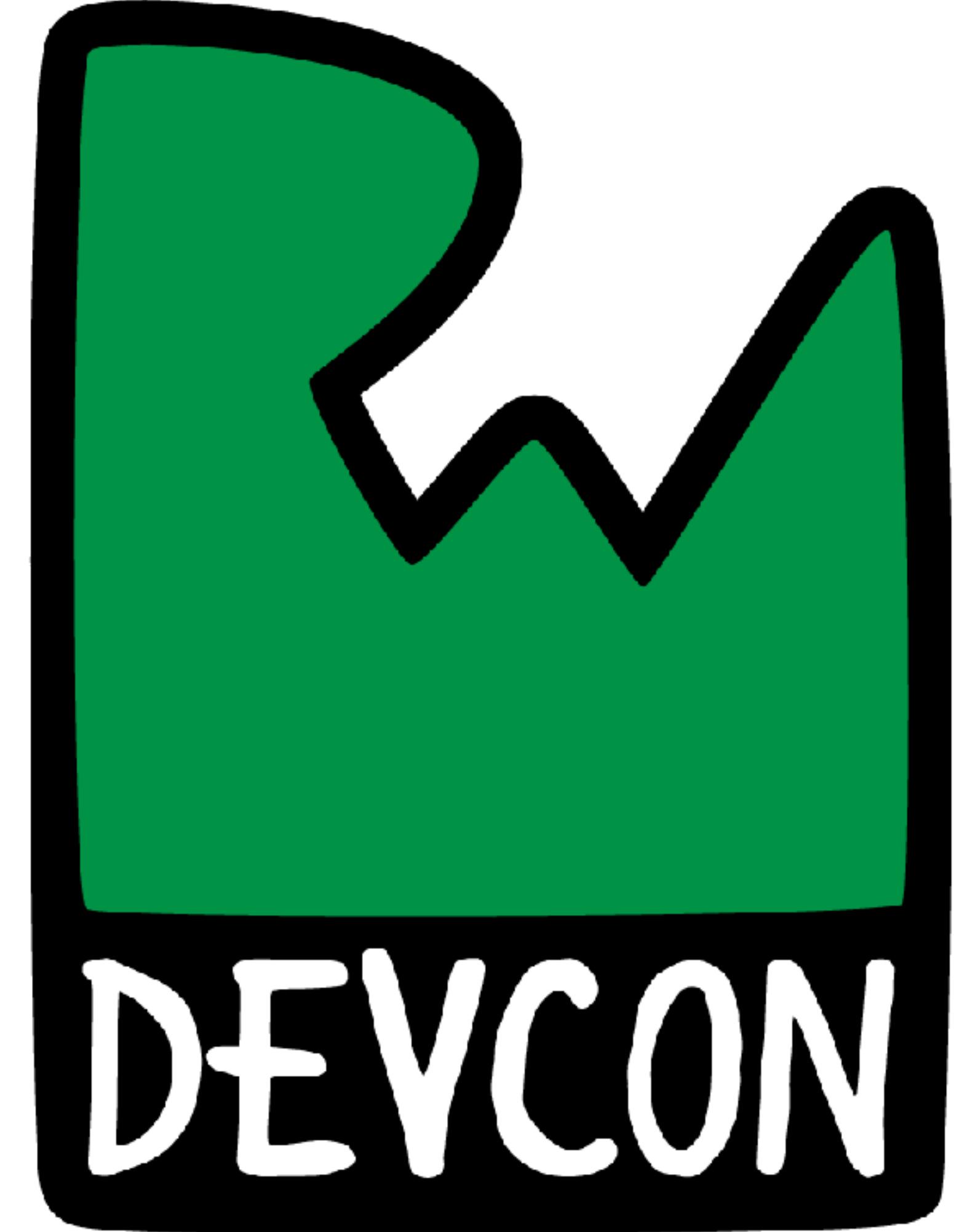
Customer
Service



DEMO 1



Session 3: Architecting Modules



REUSABILITY AND PERFORMANCE

FRAMEWORKS ALL THE WAY DOWN



R
W

REUSABILITY

Train Schedule App

Today Widget

Ticket
Purchasing
Service

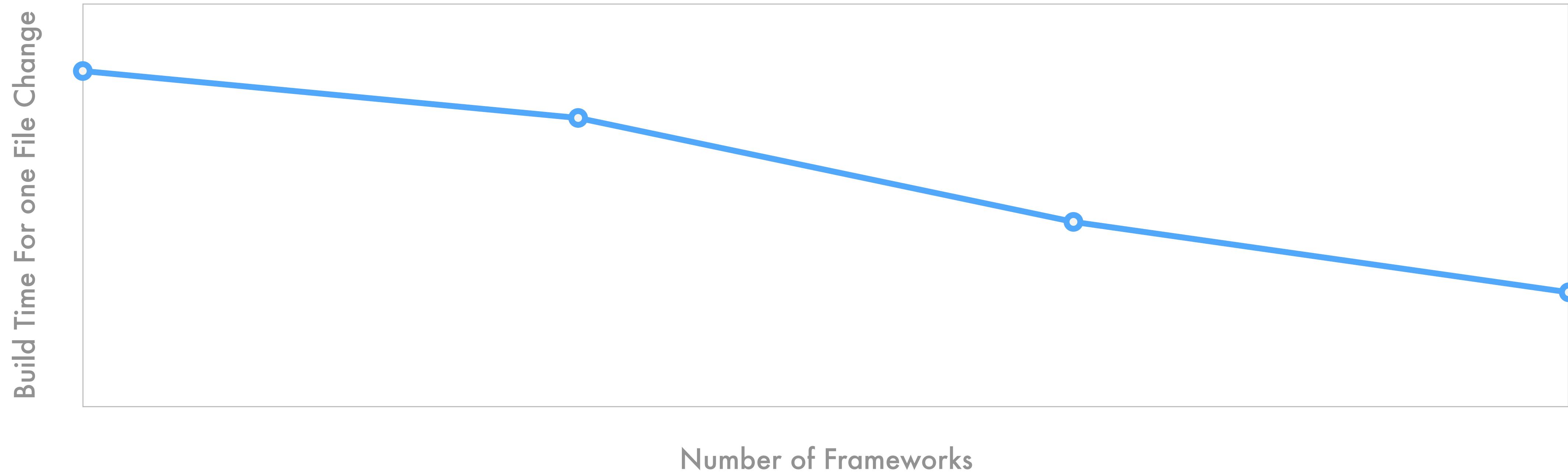
Train Schedule
Service

Handy Utilities



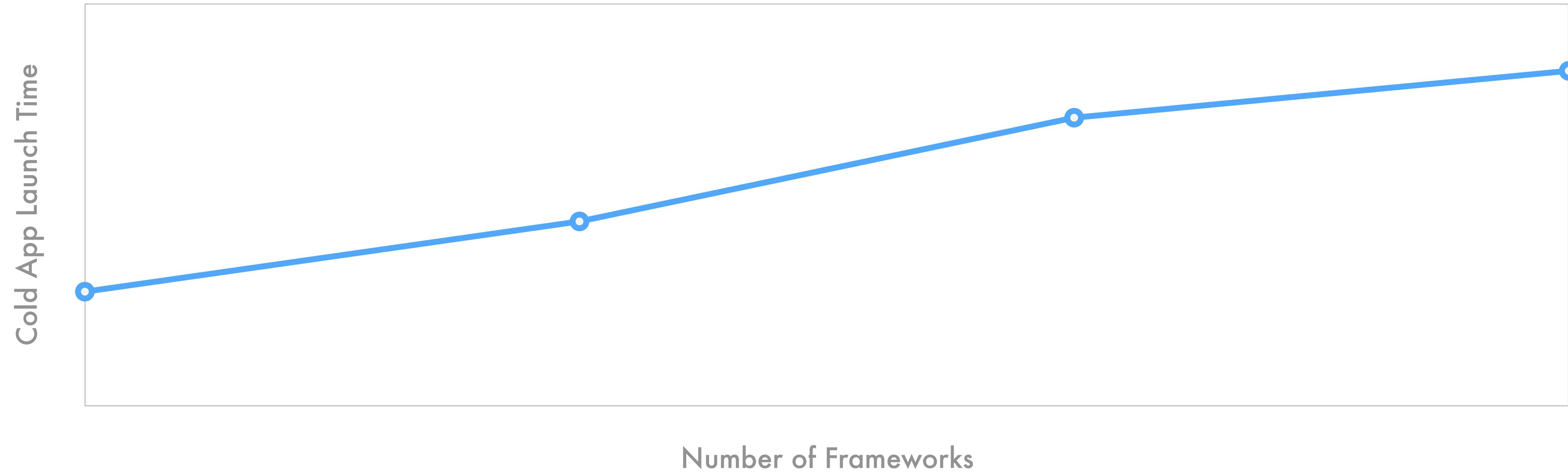
HIDDEN BENEFITS

- ▶ Improved Build & Test Time



HIDDEN COSTS

- ▶ Increased Dynamic Library Loading Time

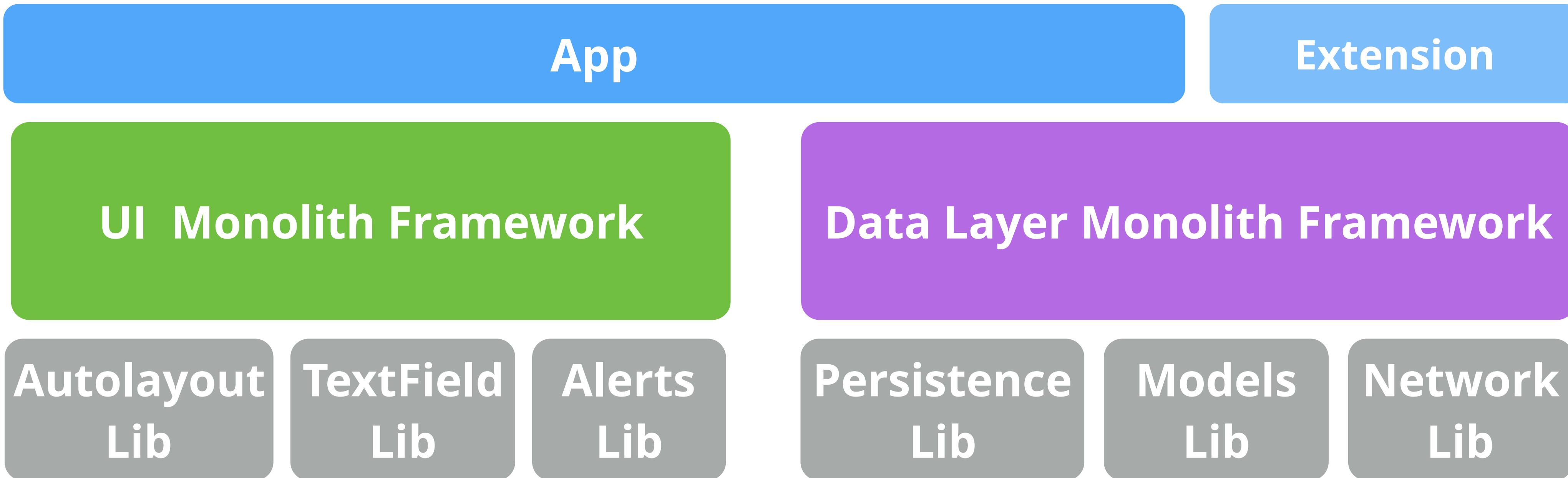


STATIC VS DYNAMIC

- ⚙️ XCode 9 Supports Swift in Static Libraries
- ⚙️ Static libraries trade file size and flexibility for improved performance.



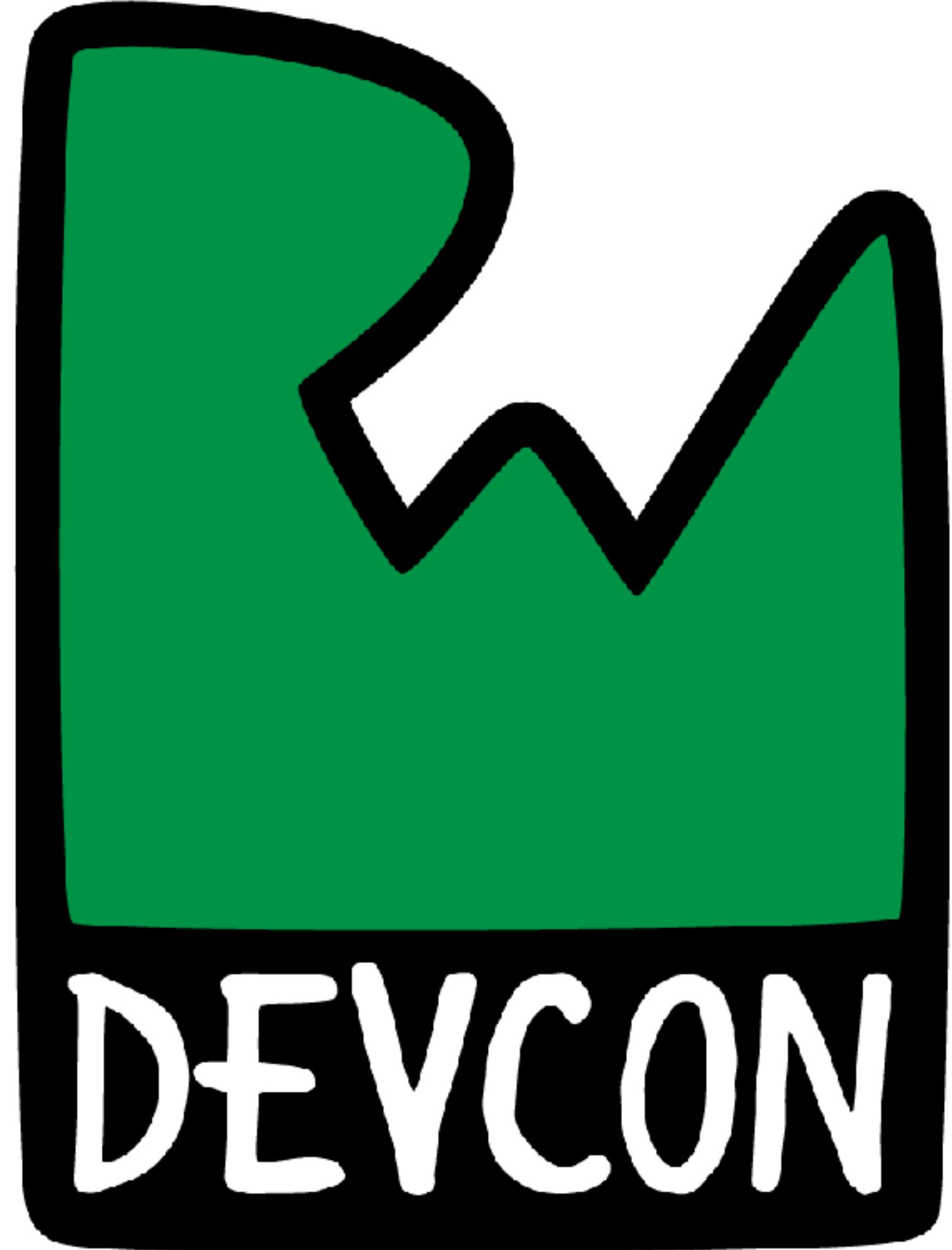
BLEND FOR FLEXIBILITY



DEMO 2



Session 3: Architecting Modules



INSULATING AGAINST CHANGES

USING THIRD PARTY MODULES

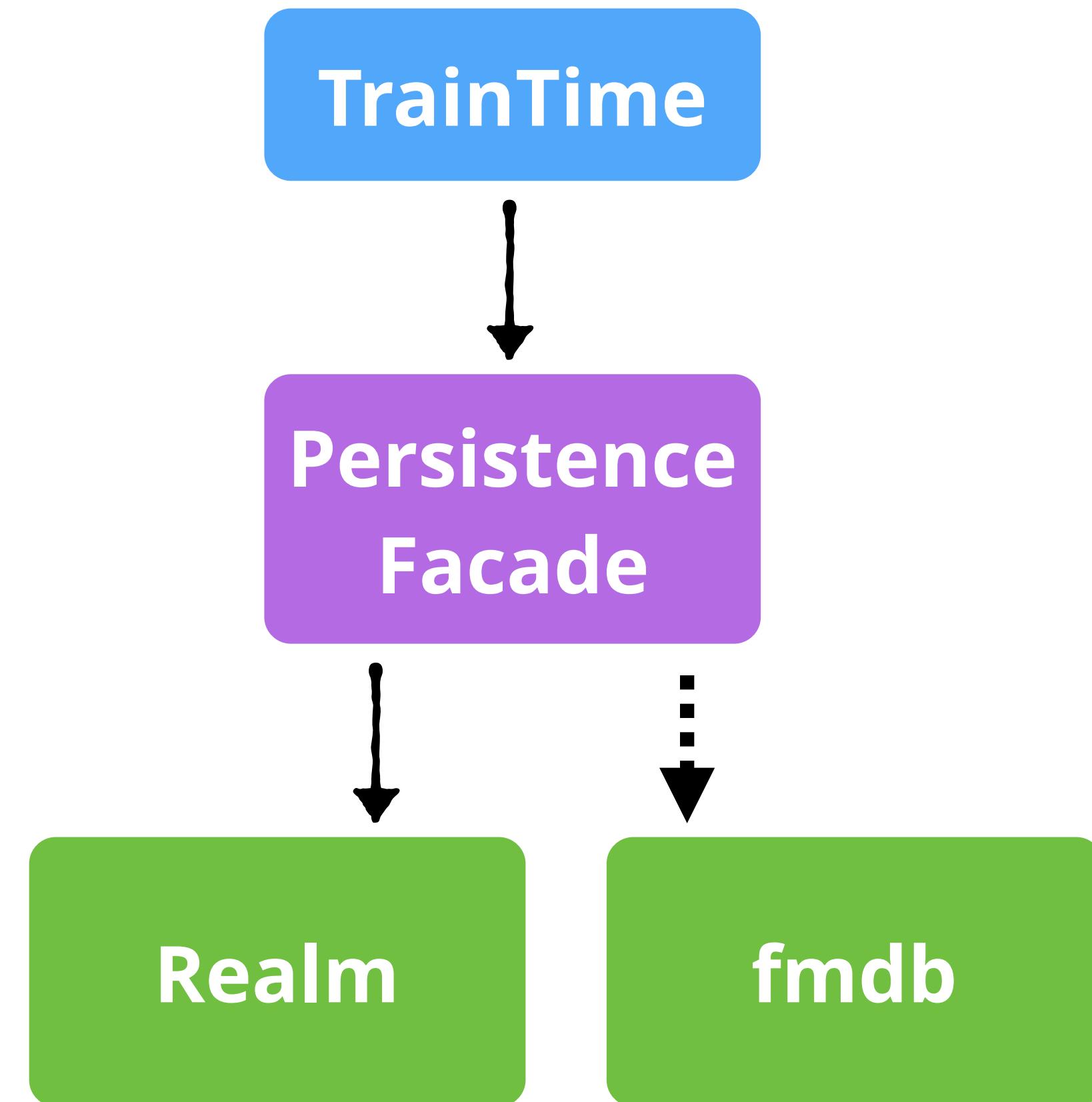
Deprecated Code



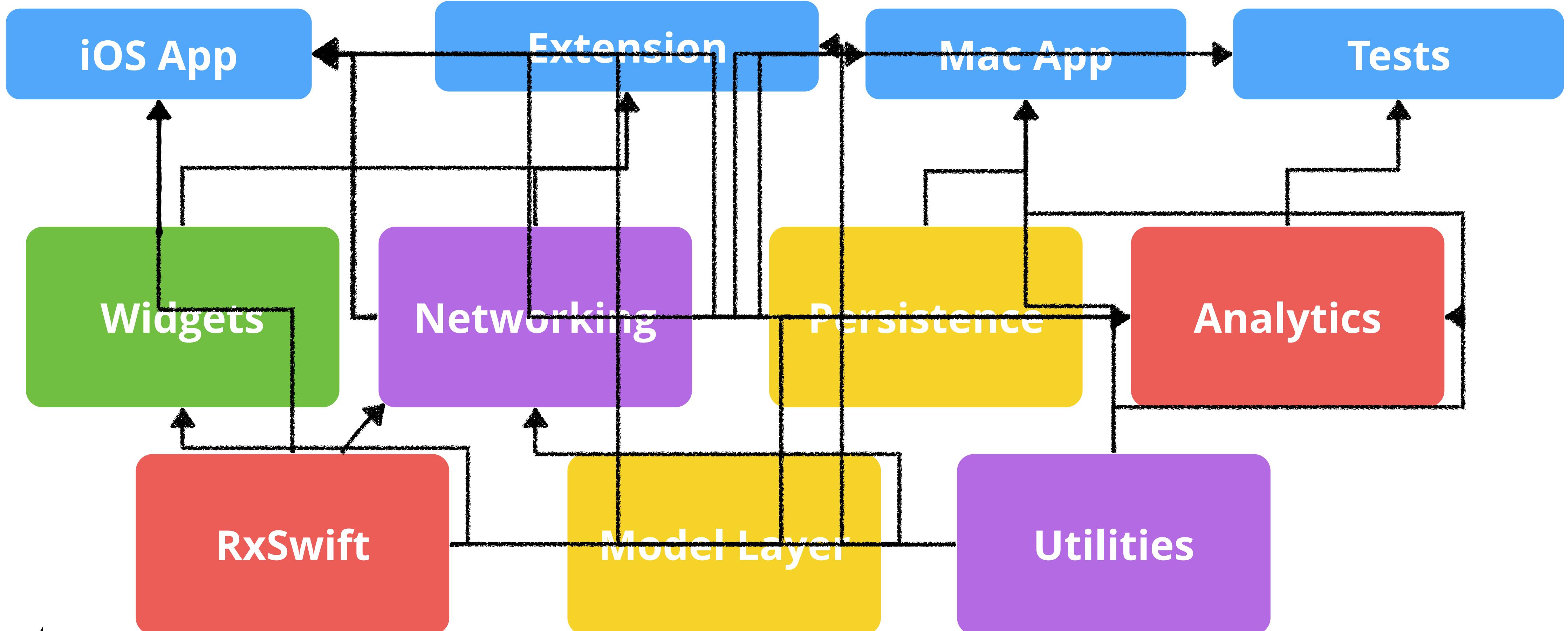
Swift 5

API Changes

BRIDGES, FACADES, ADAPTERS AND PROXIES



SCALABILITY AND DEPENDENCIES



BE A THIRD PARTY

- ⚙️ Cocoapods
- ⚙️ Carthage
- ⚙️ Swift Package Manager (SPM)



DEMO 3



Session 3: Architecting Modules



RW CONCLUSION

WHAT You LEARNED

⚙️ Demo 1: How to Module

TrainTime

Info
Service

User
Service

Helpers



WHAT You LEARNED

- ⚙️ **Demo 2:** Reusability & Static Swift Modules
 - ⚙️ Today Widget
 - ⚙️ Static Libraries



WHAT You LEARNED

- ⚙️ **Demo 3:** Protecting From Outside Changes
- ⚙️ Reusable Persistence Facade



PARTING WARNING

“Eternal vigilance is the price of modularization”



WHERE To Go FROM HERE?

- ⚙️ Creating and Distributing Frameworks
- ⚙️ Creating a Framework
- ⚙️ Building Reusable Frameworks (RWDevCon '17)
- ⚙️ Twitter: @TheMikeKatz



RESOURCES

- ⚙️ Dependency Visualizer
- ⚙️ <https://github.com/PaulTaykalo/object-dependency-visualizer>

