Working effectively at scale

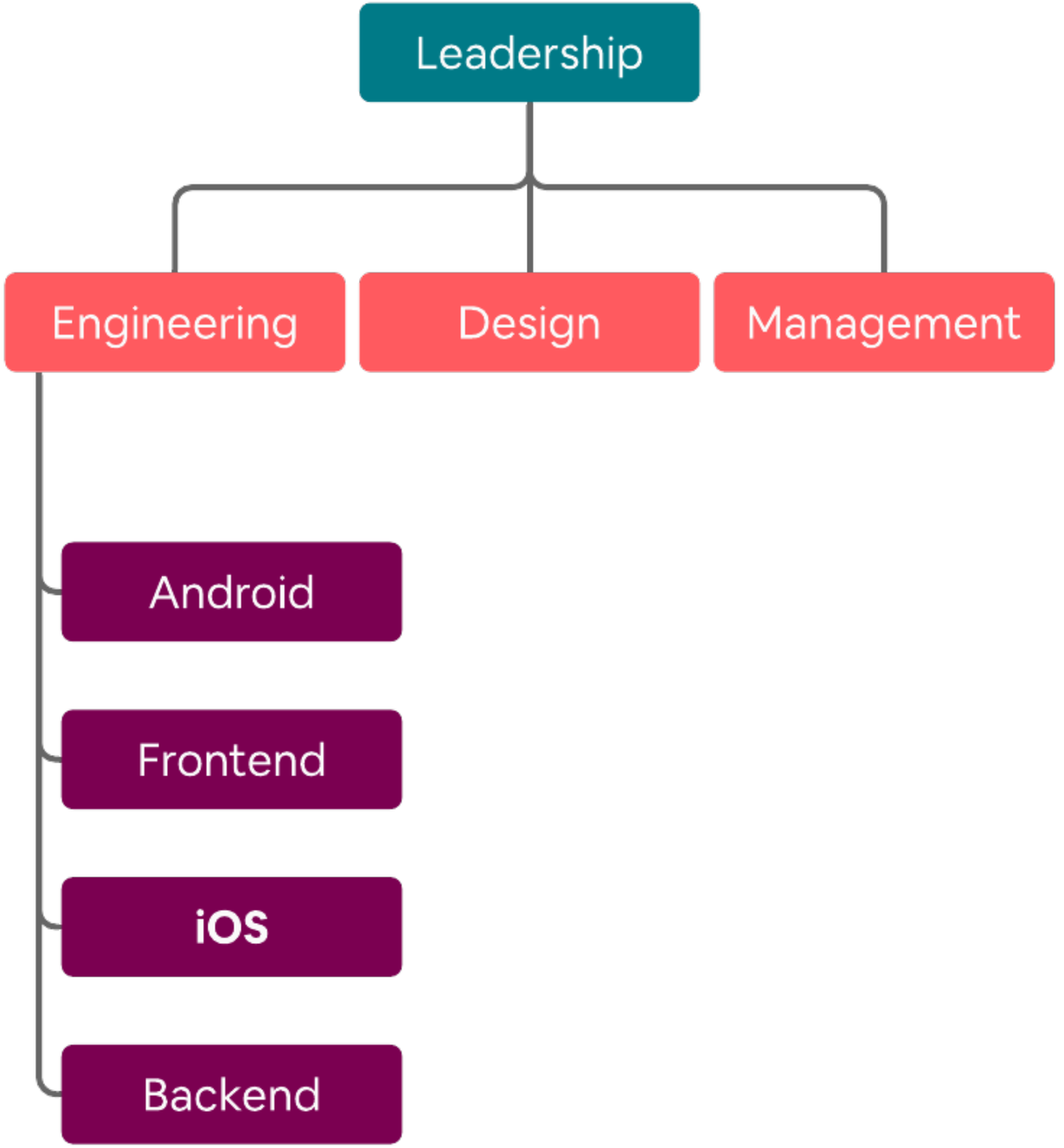
Francisco Díaz

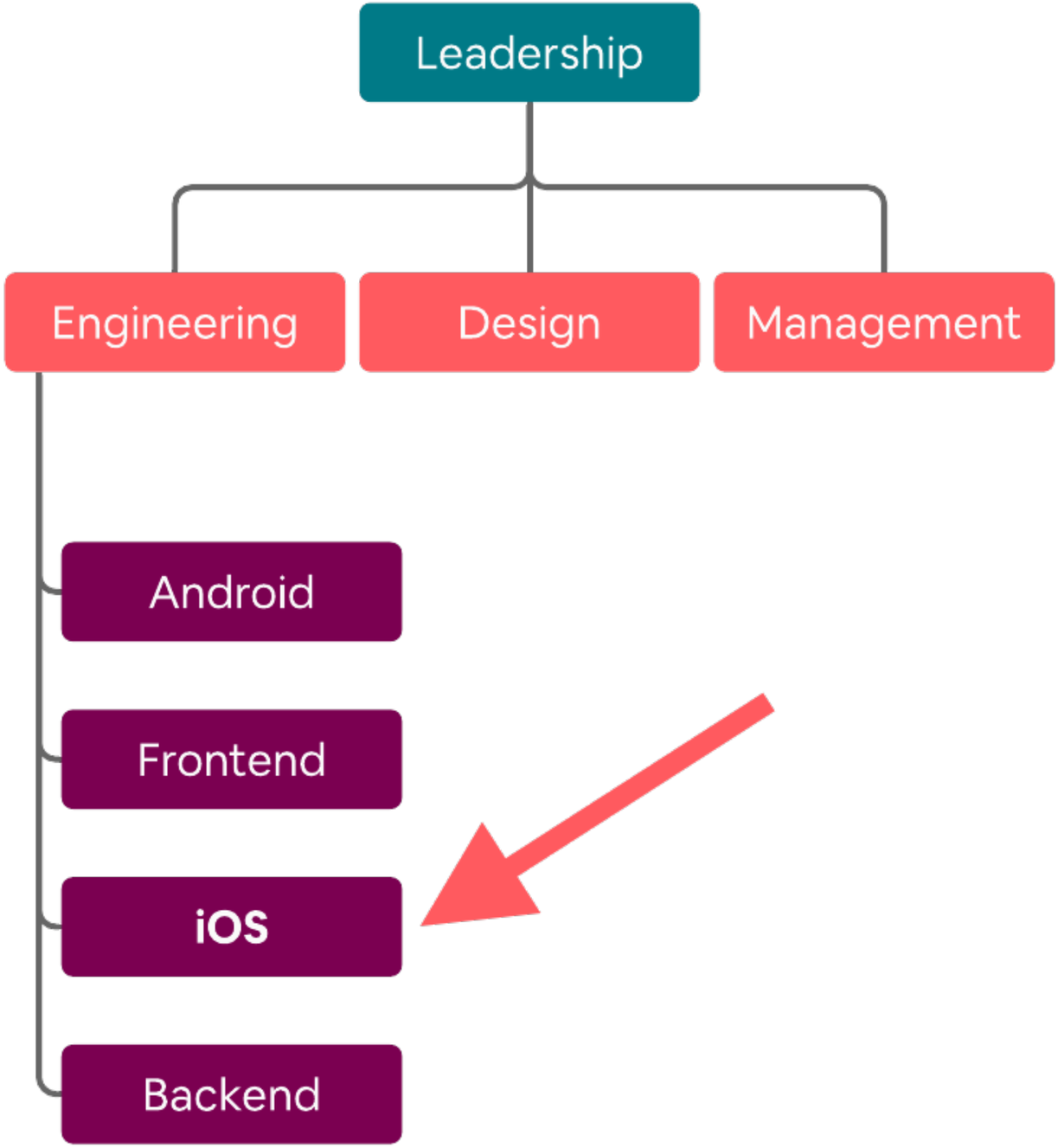
franciscodiaz.cl - @fco_diaz

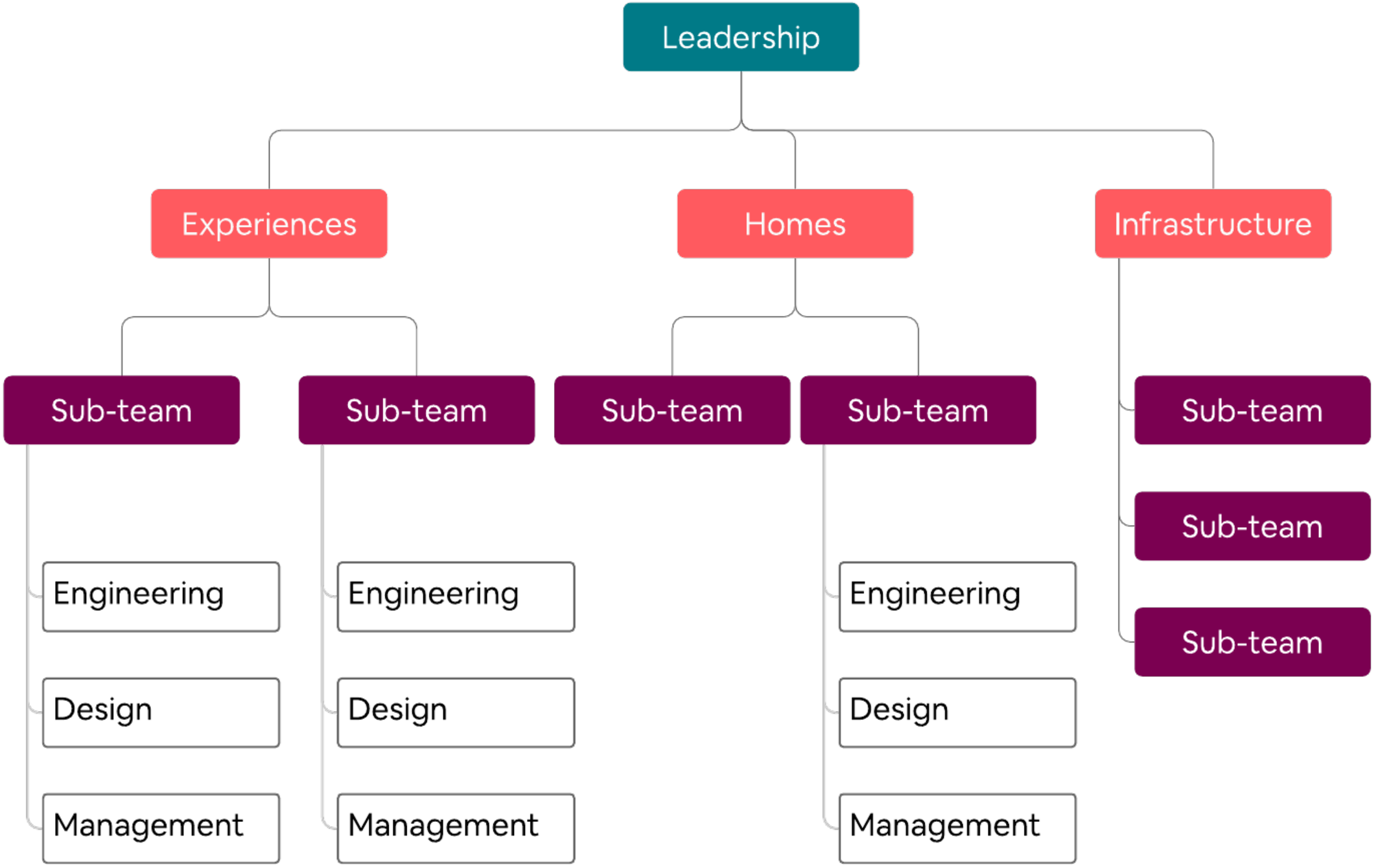
Startups

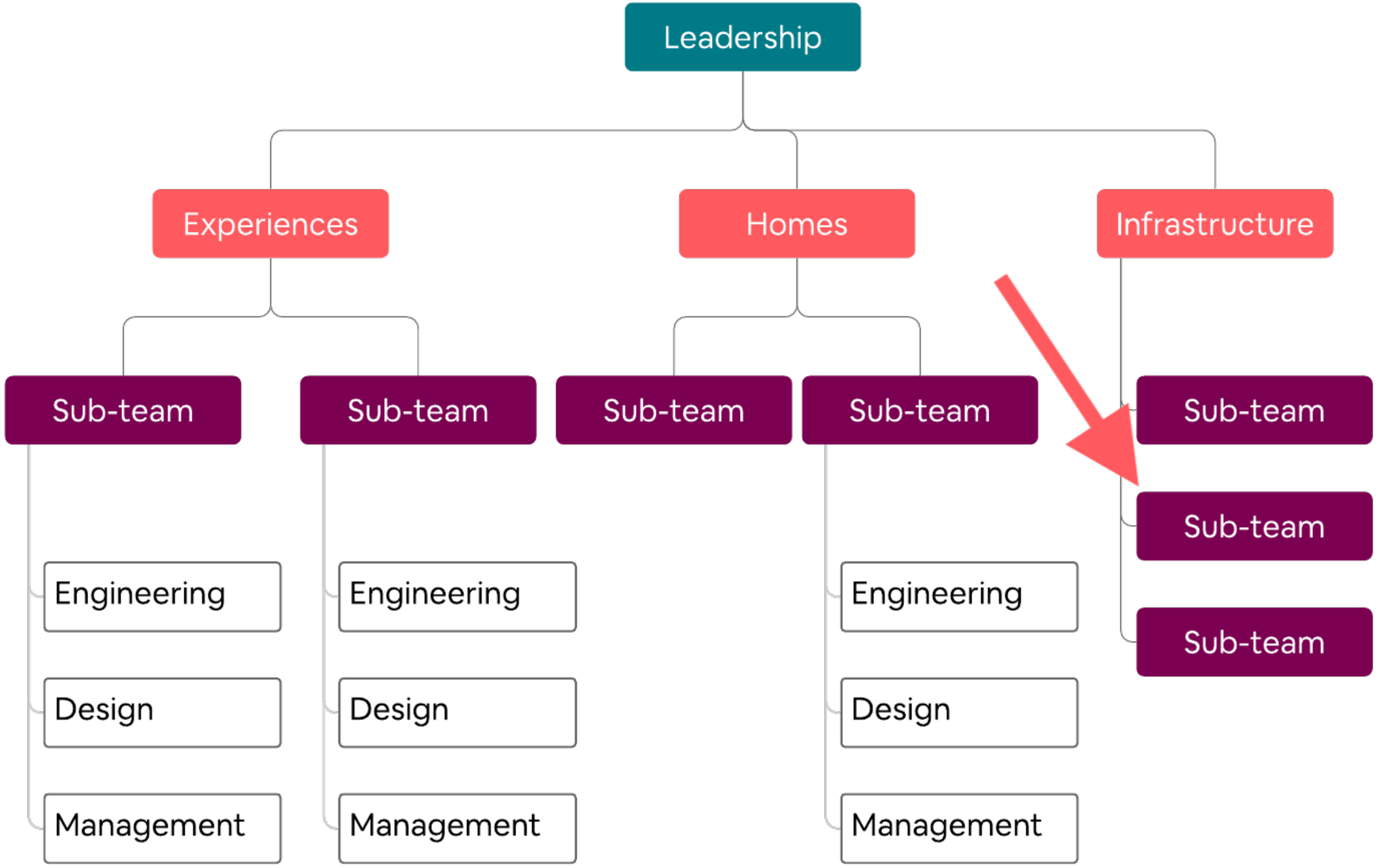
2011 - 2017

2017 - today

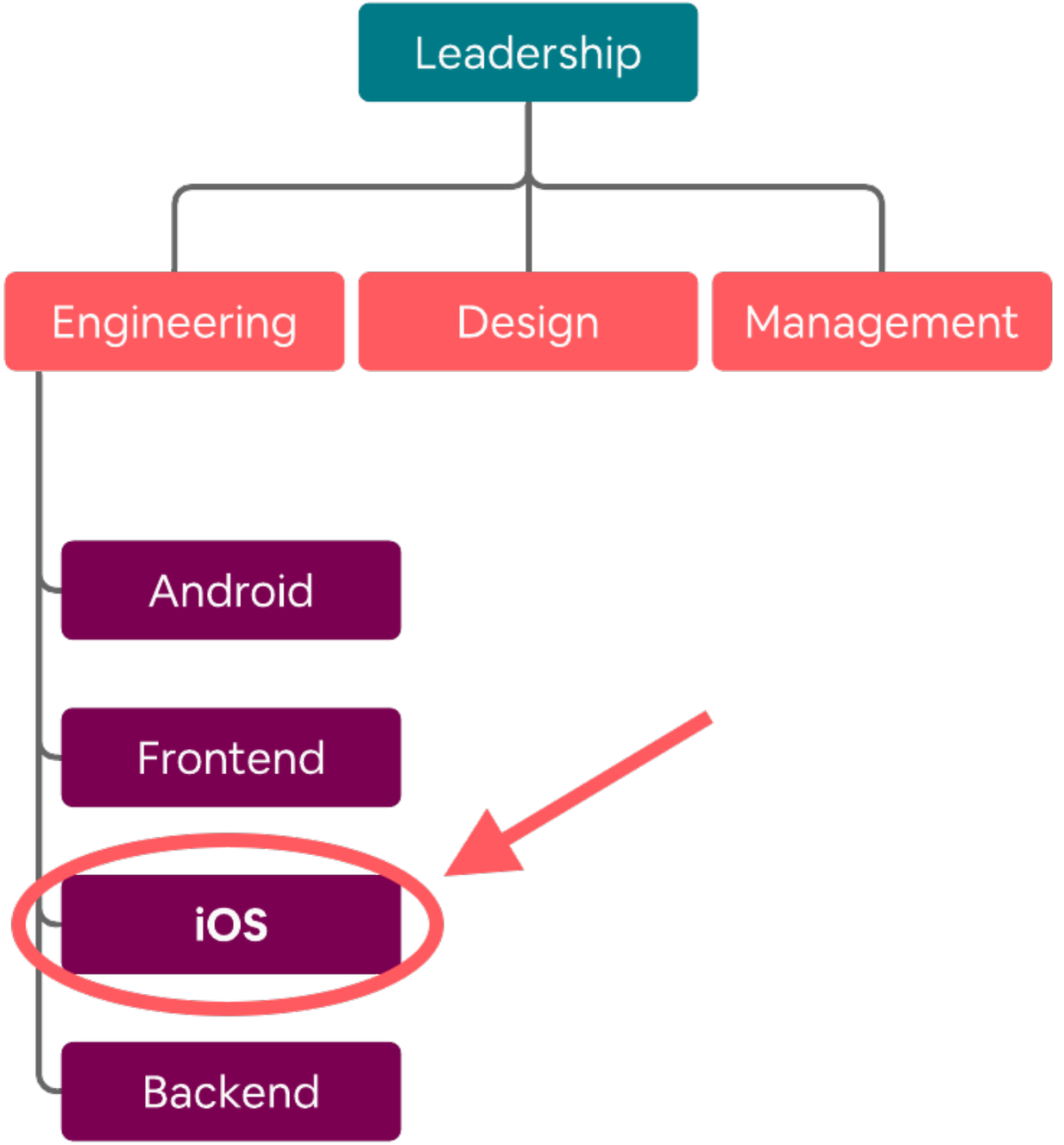

Leadership
Engineering
Design
Management
Android
Frontend
iOS
Backend

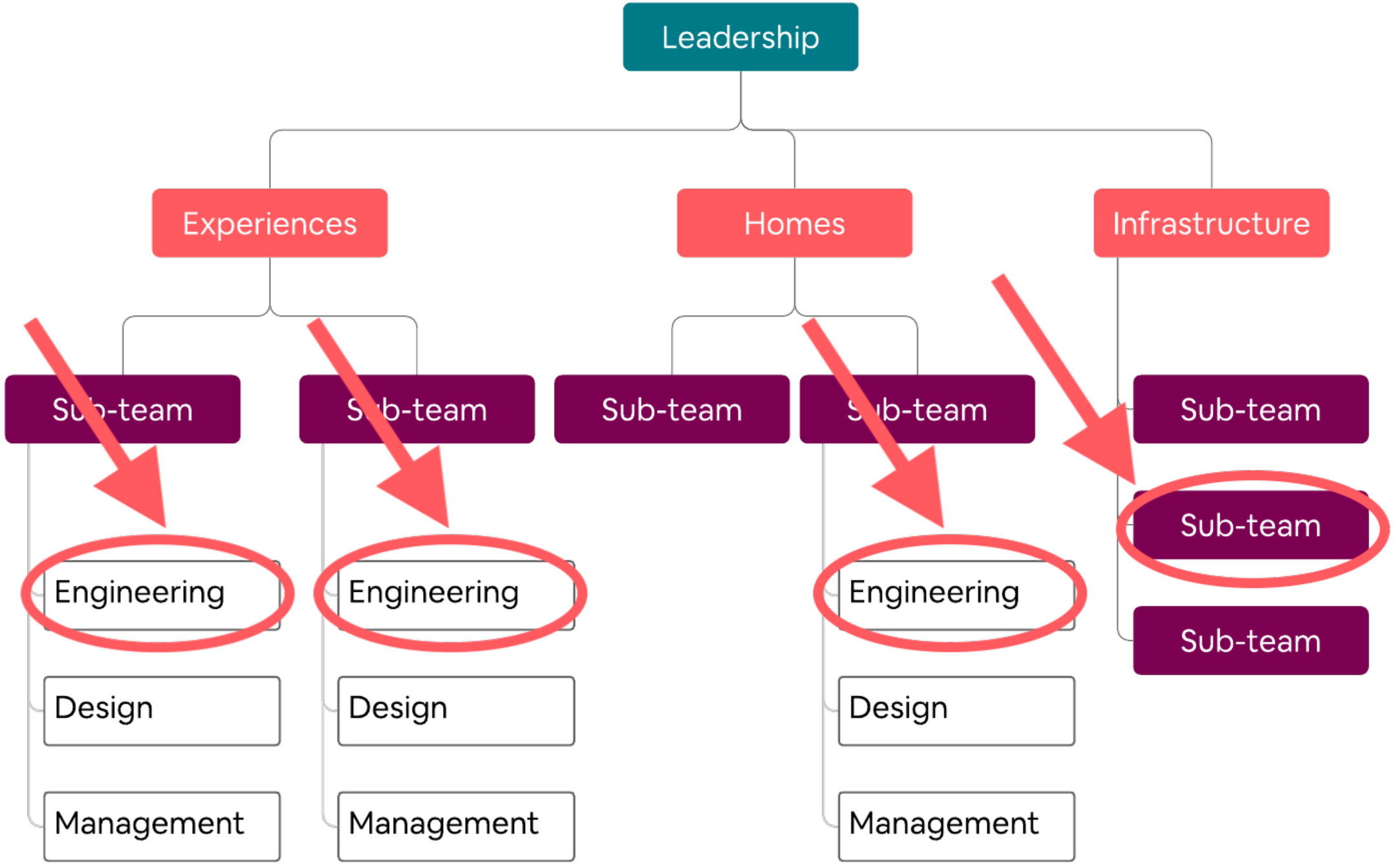

Leadership
Engineering
Design
Management
Android
Frontend
iOS
Backend


Leadership
Experiences
Homes
Infrastructure
Sub-team
Sub-team
Sub-team
Sub-team
Sub-team
Sub-team
Sub-team
Engineering
Design
Management
Engineering
Design
Management
Engineering
Design
Management


Leadership
Experiences
Homes
Infrastructure
Sub-team
Sub-team
Sub-team
Sub-team
Sub-team
Sub-team
Sub-team
Engineering
Design
Management
Engineering
Design
Management
Engineering
Design
Management


YEP
MERRY CHRISTM
DOUGHNUTS
DEADLIFTS


Leadership
Engineering
Design
Management
Android
Frontend
iOS
Backend


Leadership
Experiences
Homes
Infrastructure
Sub-team
Sub-team
Sub-team
Sub-team
Sub-team
Sub-team
Sub-team
Engineering
Engineering
Engineering
Design
Design
Design
Management
Management
Management

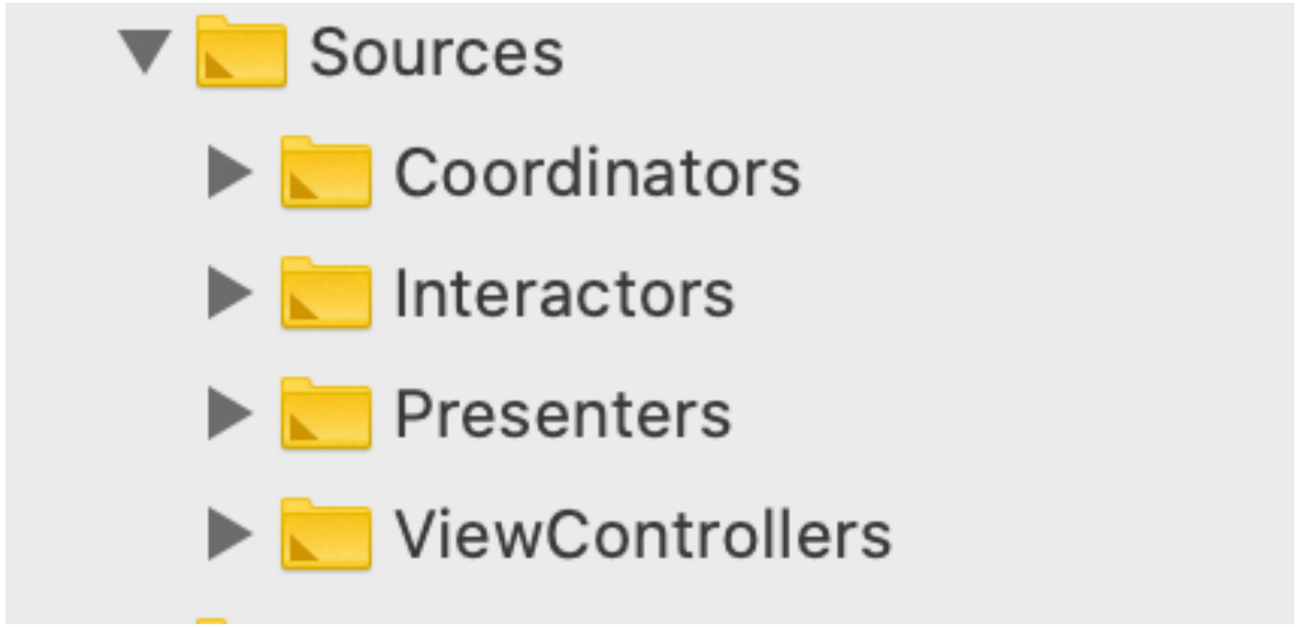

THIS GONE BE A LIL DIFFERENT

organizations … are constrained to produce designs which are copies of the communication structures of these organizations

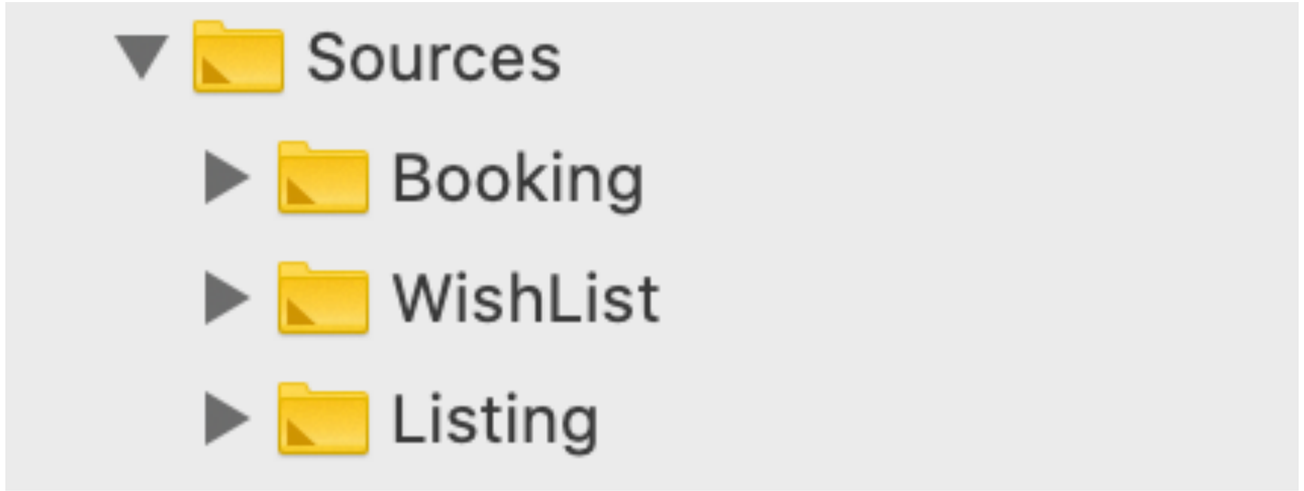
— Conway's law

How do you divide your codebase?

Architectural layer



User Flow



What about Airbnb?

```
commit d9fe9
Author: Andrew
Date:   Wed Jun 16 15:09:56 2010 -0700

    initial commit, all data structures created
```

1 million lines

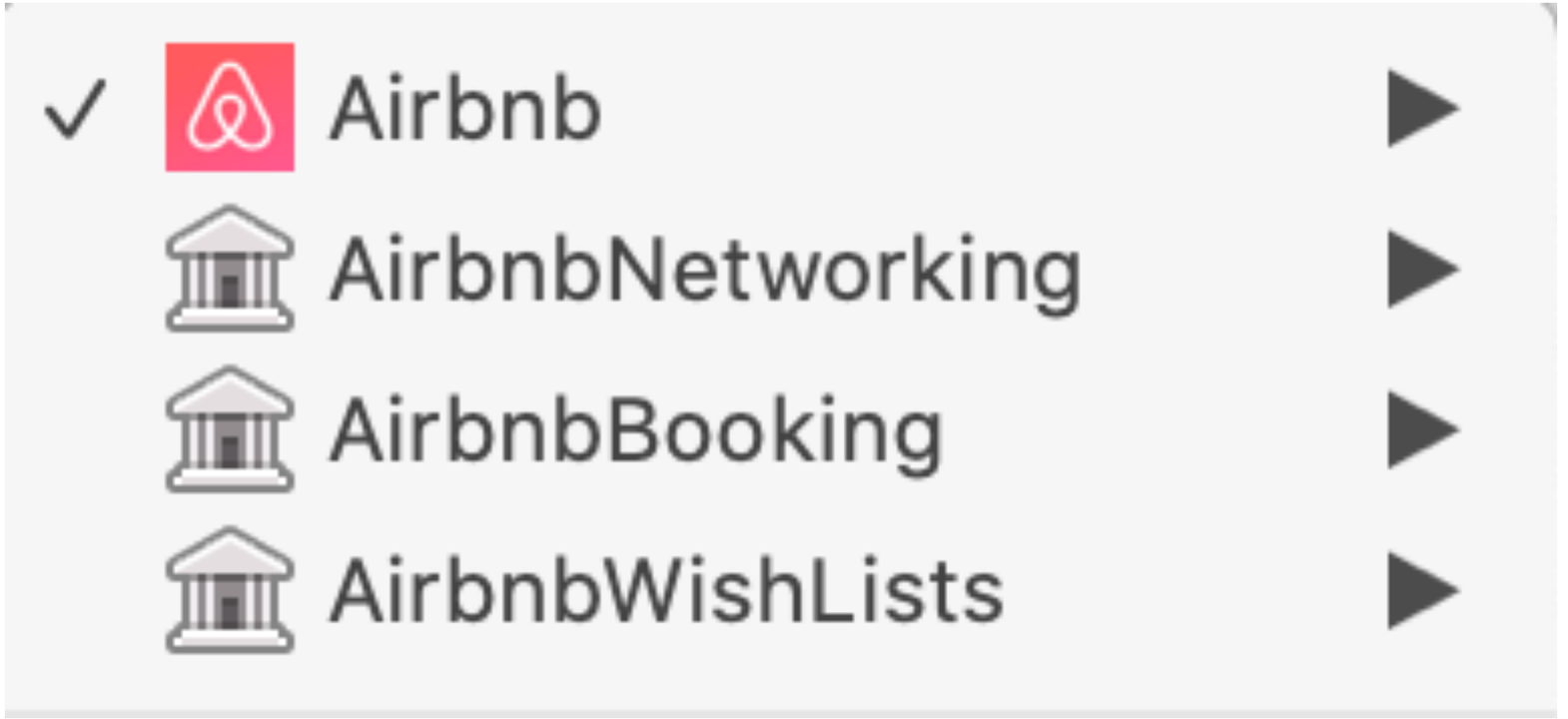
of Swift

~80 commits

on any given day to the repo (Android + iOS)

Bigger

buckets

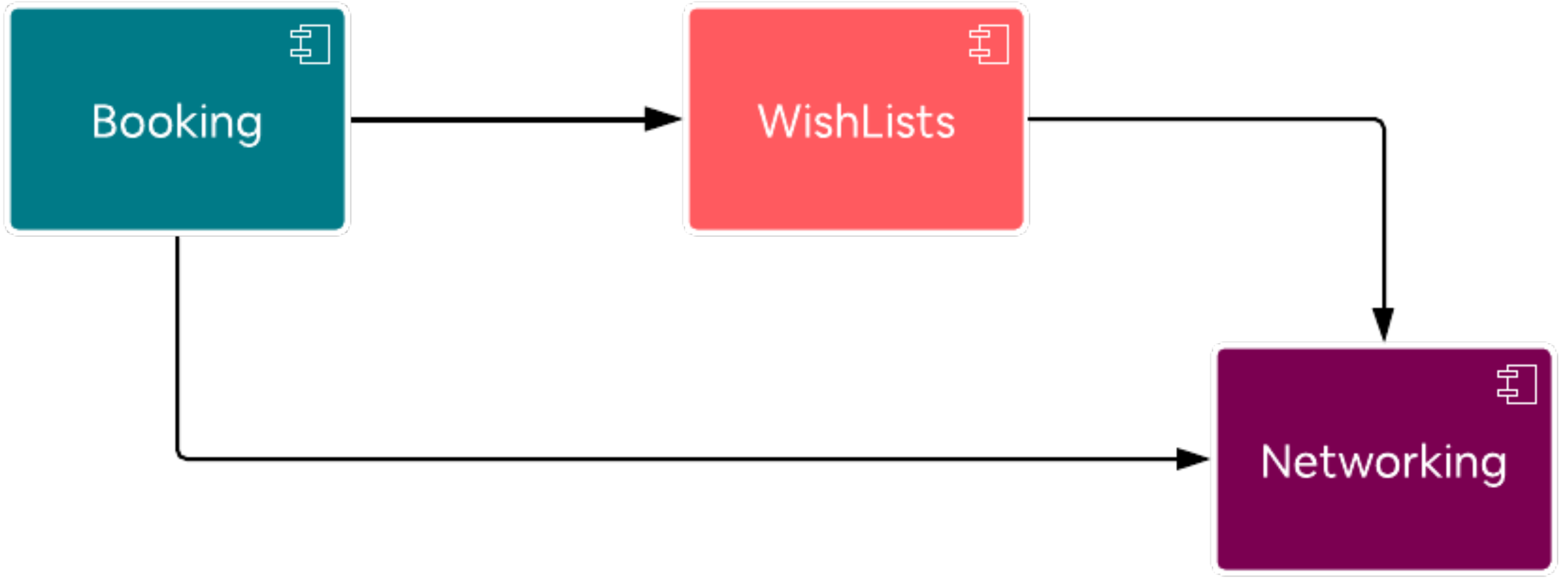

Airbnb
AirbnbNetworking
AirbnbBooking
AirbnbWishLists

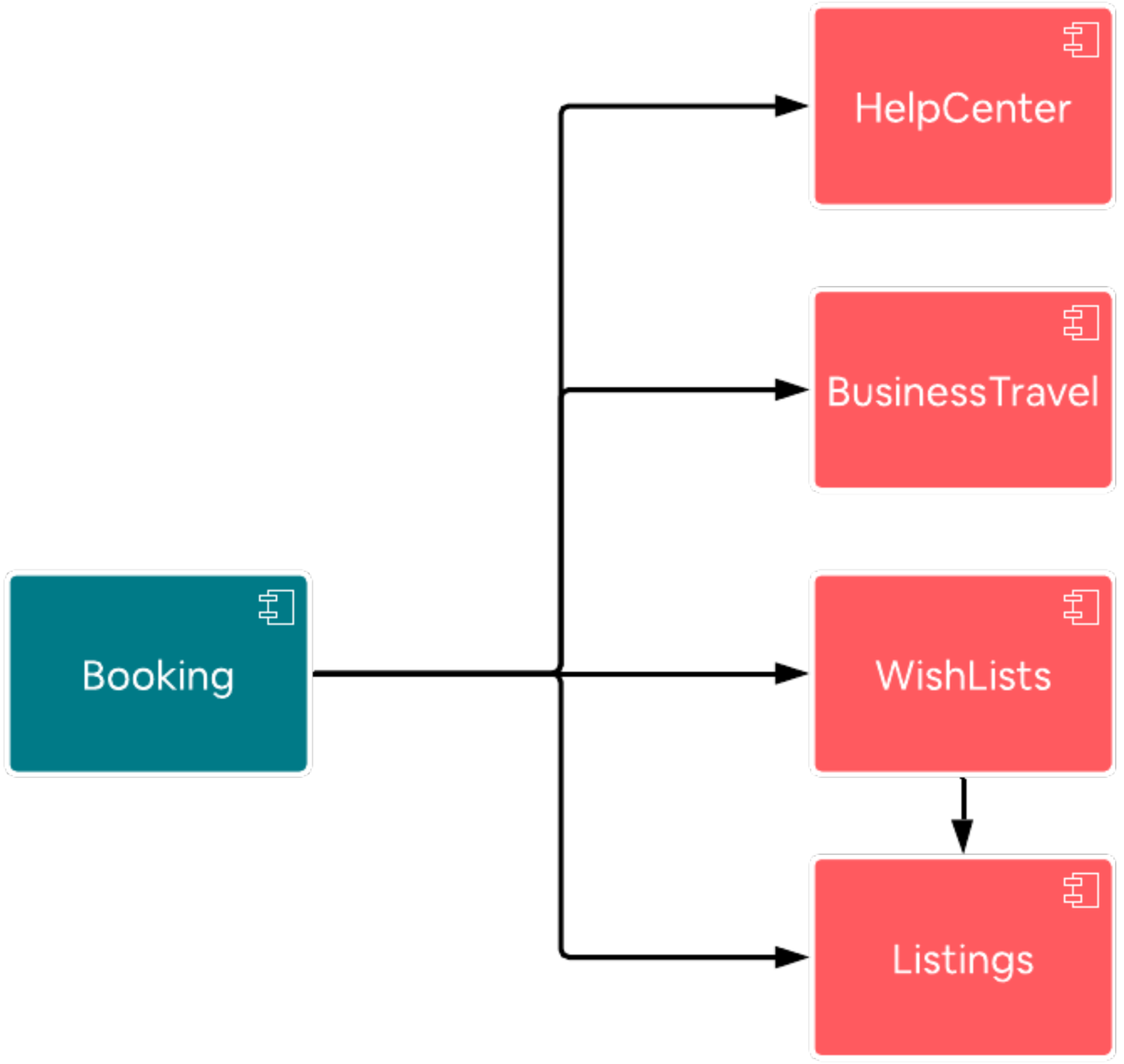
A user should be able to wishlist a listing from the booking flow

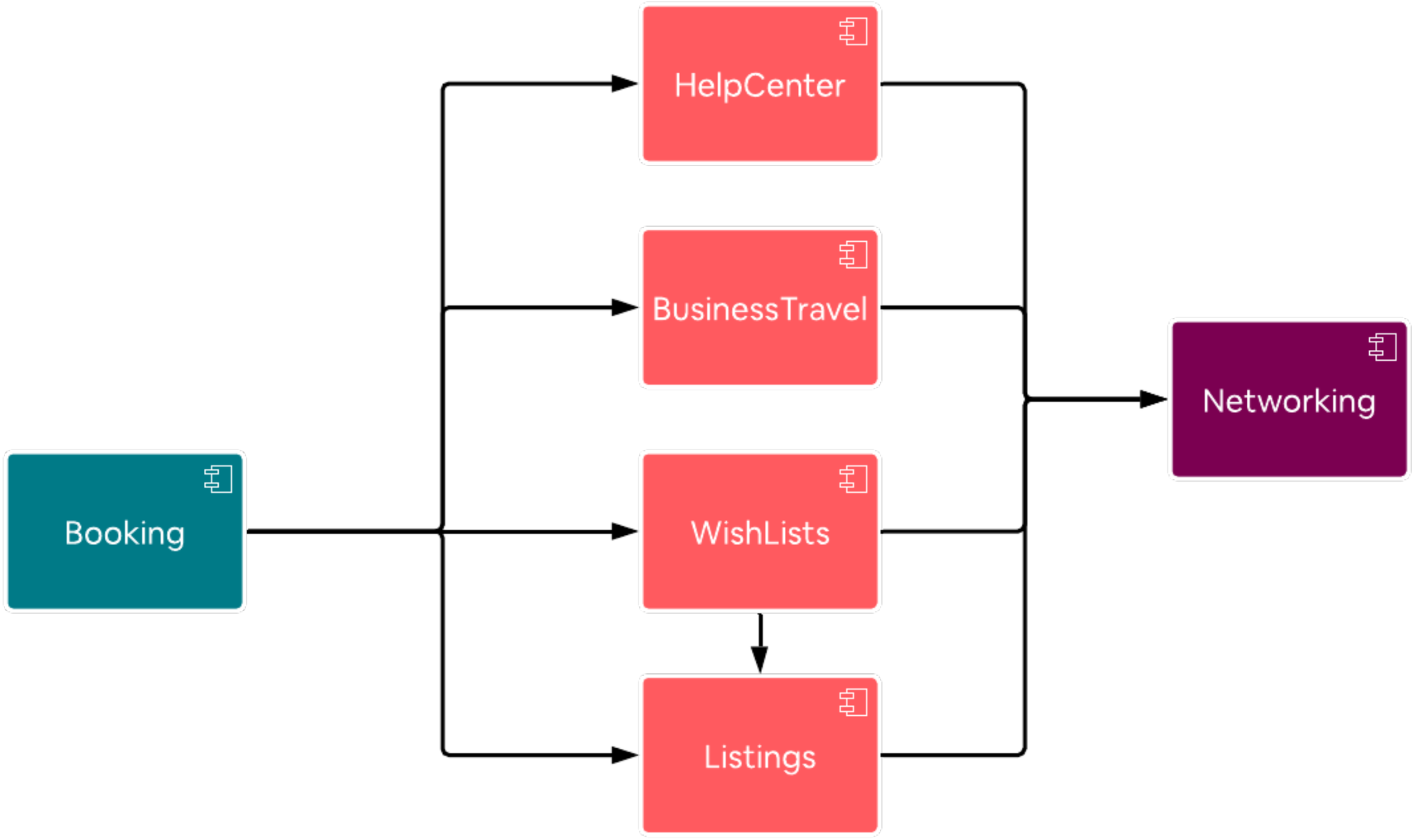
How do they

relate

with each other?


Booking
WishLists
Networking


HelpCenter
BusinessTravel
Booking
WishLists
Listings


HelpCenter
BusinessTravel
Networking
Booking
WishLists
Listings

50 min

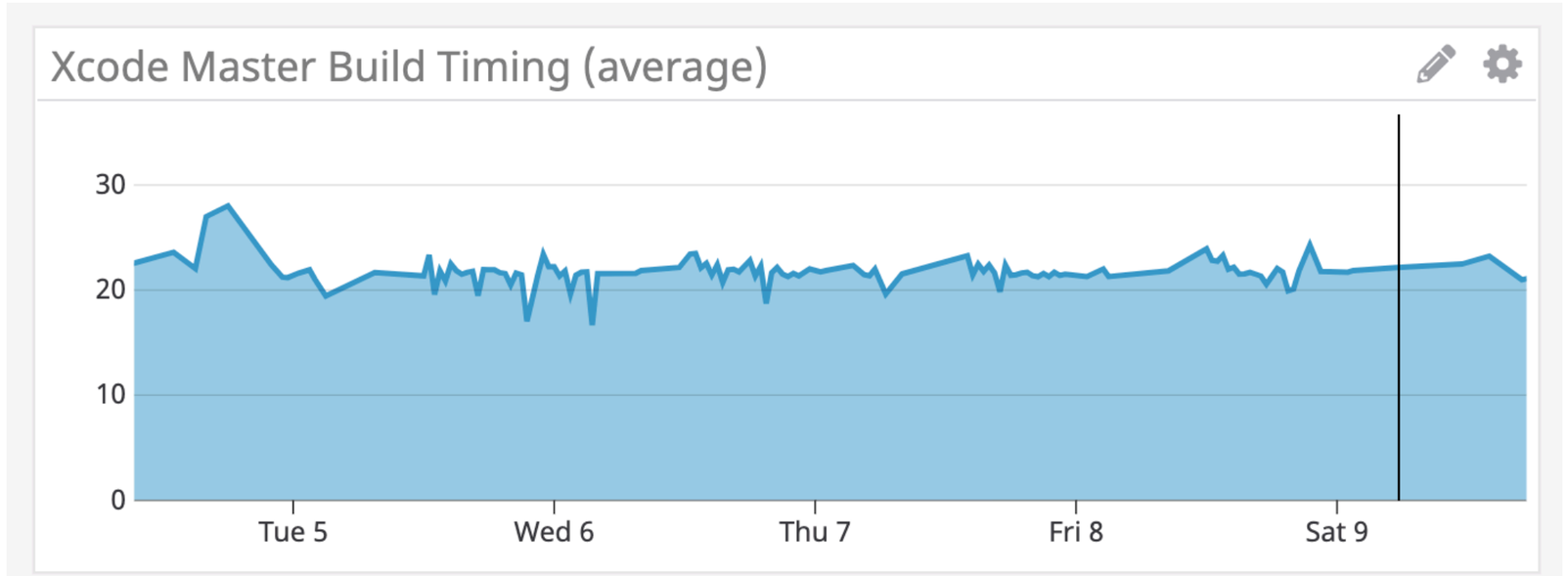
local clean builds

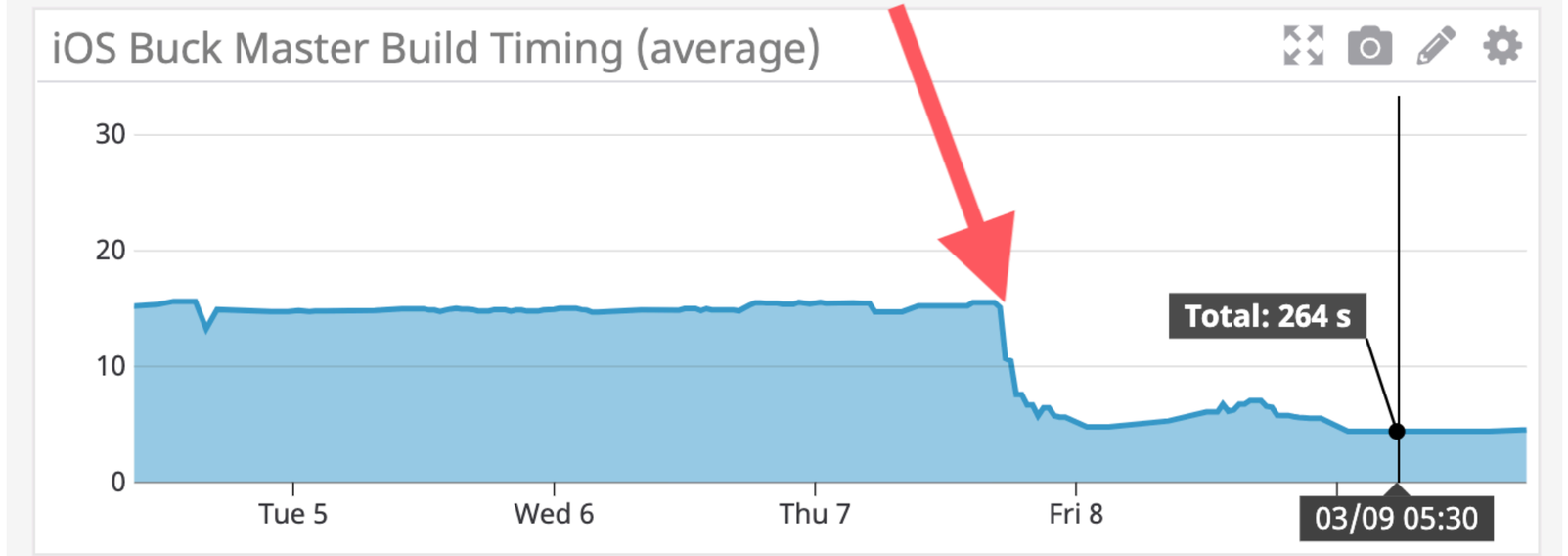
~30 min

Buck

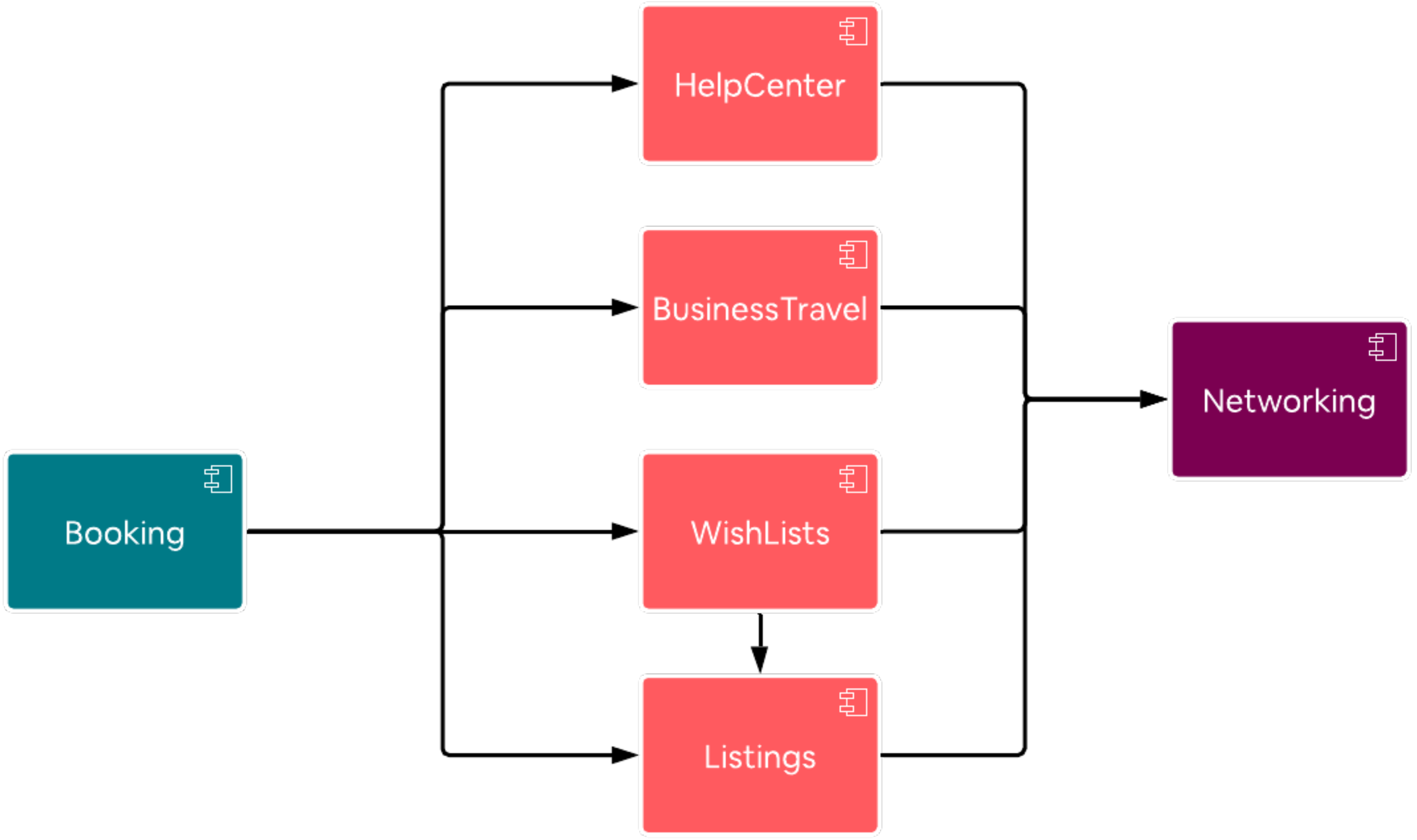
HTTP Cache

https://github.com/airbnb/BuckSample


Xcode Master Build Timing (average)
30
20
10
0
Tue 5
Wed 6
Thu 7
Fri 8
Sat 9

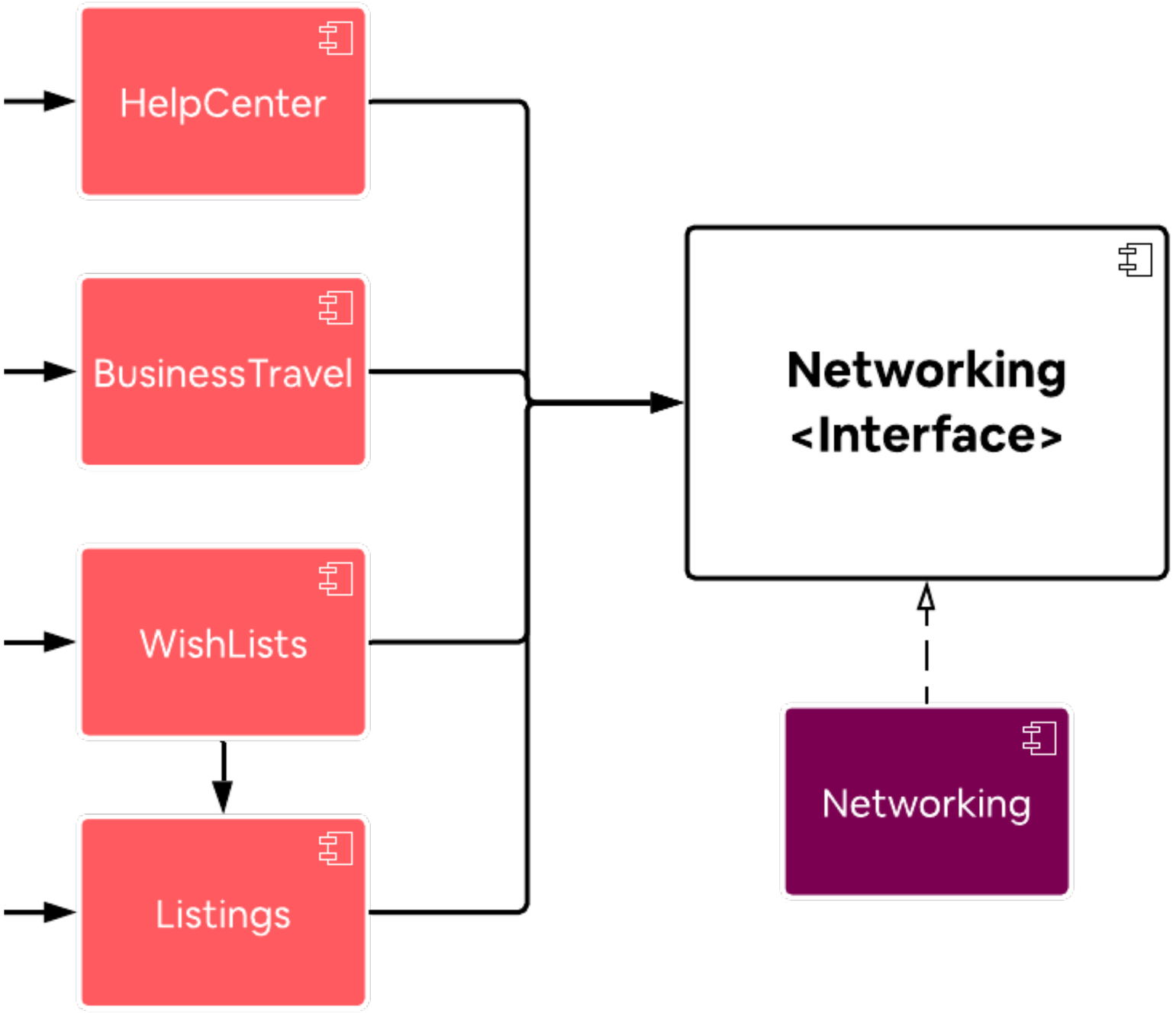

iOS Buck Master Build Timing (average)
30
20
10
0
Tue 5
Wed 6
Thu 7
Fri 8
Total: 264 s
03/09 05:30

~5 min

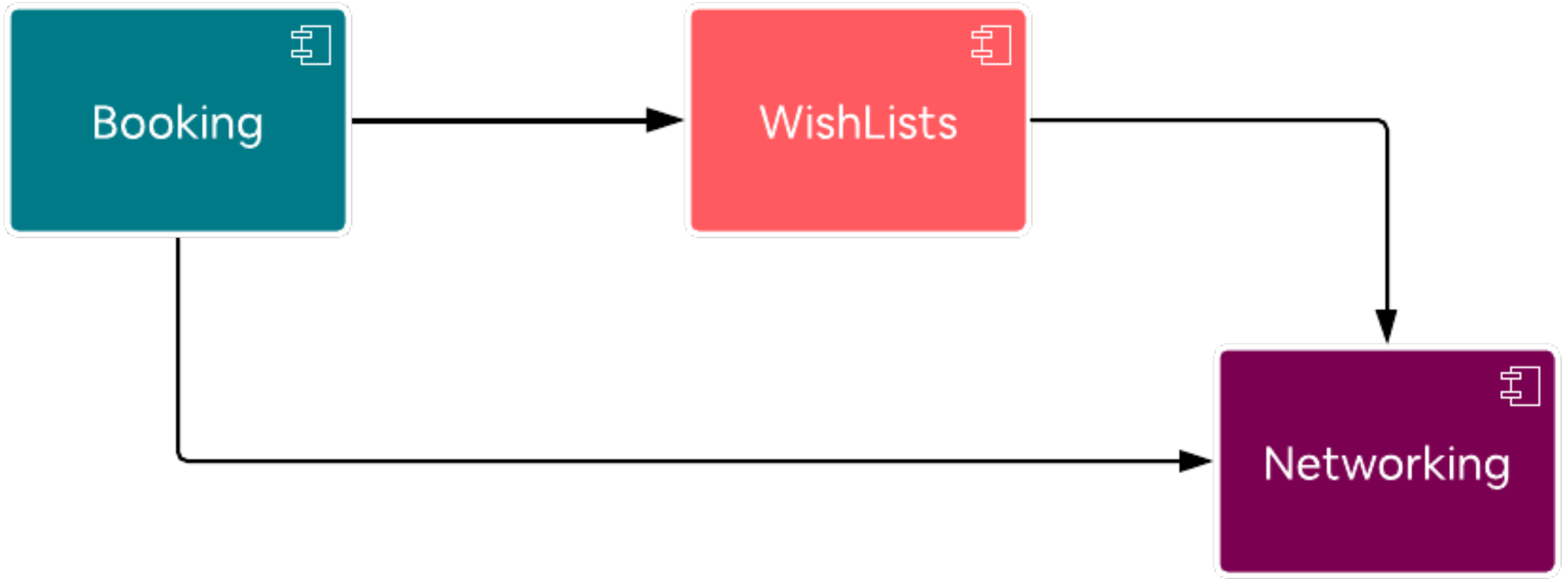

HelpCenter
BusinessTravel
Networking
Booking
WishLists
Listings

Dependency inversion

- **High-level modules should not depend on low-level modules**. Both should depend on abstractions.
- **Abstractions should not depend on details**. Details (concrete implementations) should depend on abstractions.


HelpCenter
BusinessTravel
WishLists
Listings
Networking
<Interface>
Networking

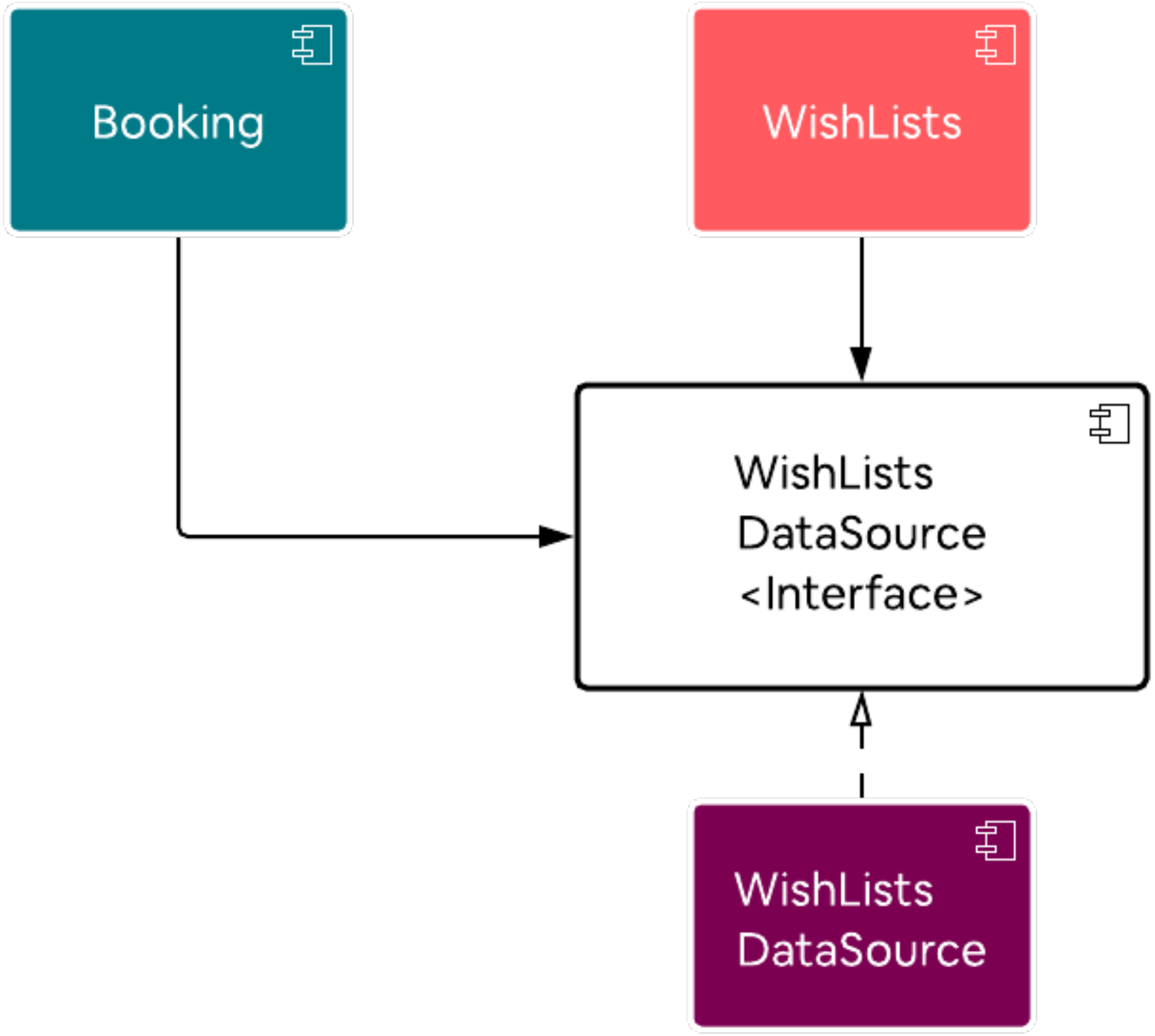
A user should be able to wishlist a listing from the booking flow


Booking
WishLists
Networking

Easy!

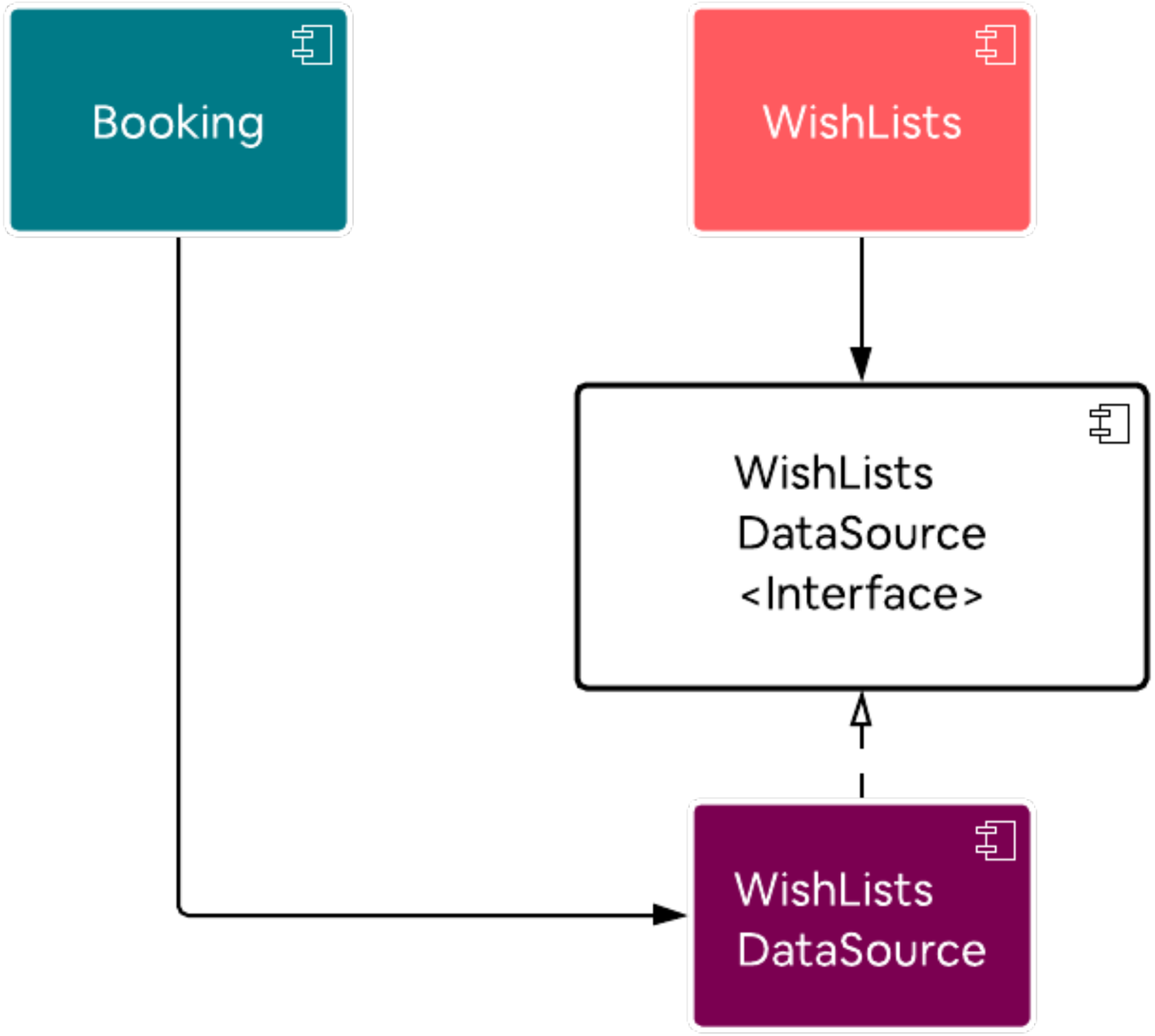
`WishListDataSource`

+ interface!


Booking
WishLists
WishLists
DataSource
<Interface>
WishLists
DataSource

`WishListDataSource`

is still visible


Booking
WishLists
WishLists
DataSource
<Interface>
WishLists
DataSource

Socializing

best practices

iOS developers

Automating

best practices

Groups
Modules

Groups

Modules

Module Types

Module Types

Feature + Interface

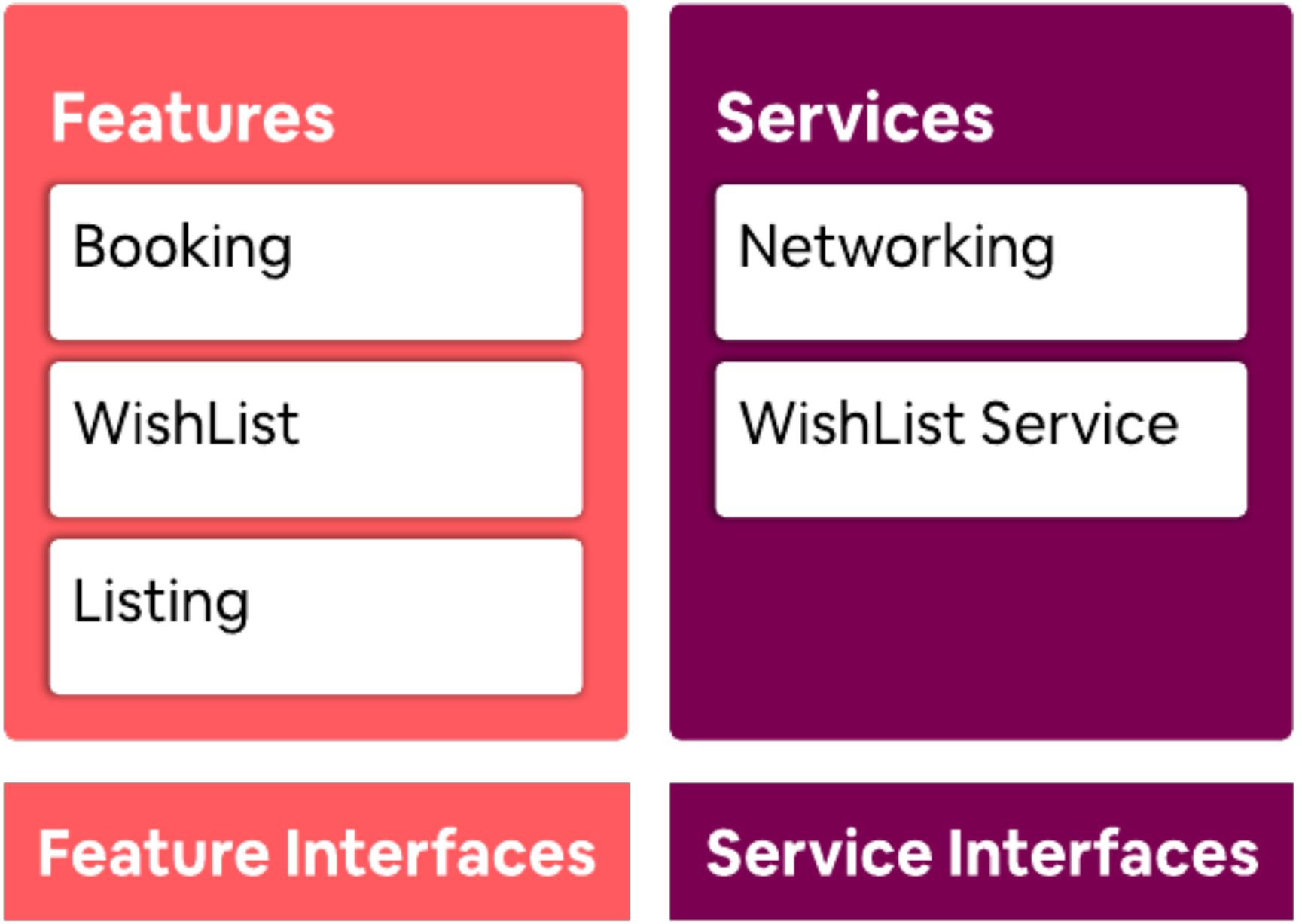
Service + Interface

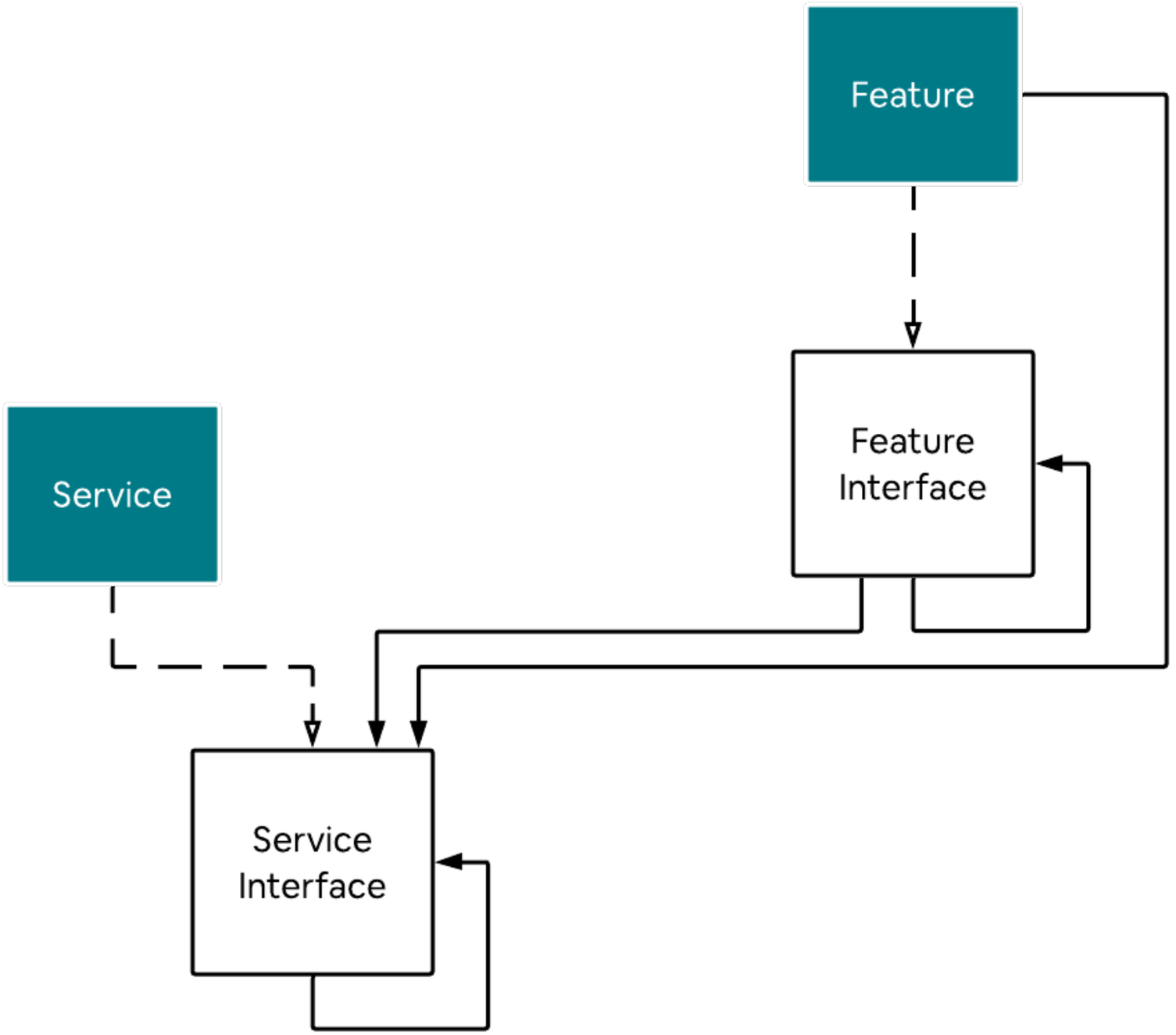
Feature

A screen or a flow in the app

Service

Manage shared state or resources


Features
Booking
WishList
Listing
Services
Networking
WishList Service
Feature Interfaces
Service Interfaces


Feature
Feature Interface
Service
Service Interface

How do we enforce these best practices?

/services
/service_interfaces
/features
/feature_interfaces

```
def service_interface(
        name,
        deps):

    max_visibility = [
        "//ios/feature_interfaces/...",
        "//ios/features/...",
        "//ios/service_interfaces/...",
        "//ios/services/...",
    ]
```

```
service_interface(
    name = "Networking",
    deps = [
        "//ios/service_interfaces/Logging",
    ],
)
```

```
feature(
    name = "Booking",
    deps = [
        "//ios/service_interfaces/Networking",
        "//ios/service_interfaces/WishListService",
    ],
)
```

iOS Platform

Module creation
needs to be easy

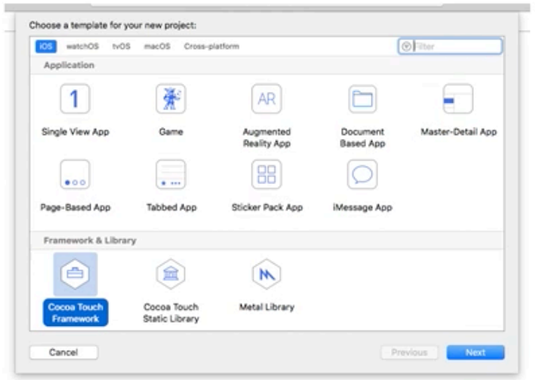
Can't find the perfect home for the new hotness you're cranking on? Your team's module getting crazy huge? If the answer to either of these or similar questions is yes, you're in the right place.

1. Let's walk through creating a new module, using the new `AirbnbWildcat` module as an example.

Creating Wildcat

Create a new framework (File → New → Project... → Cocoa Touch Framework).

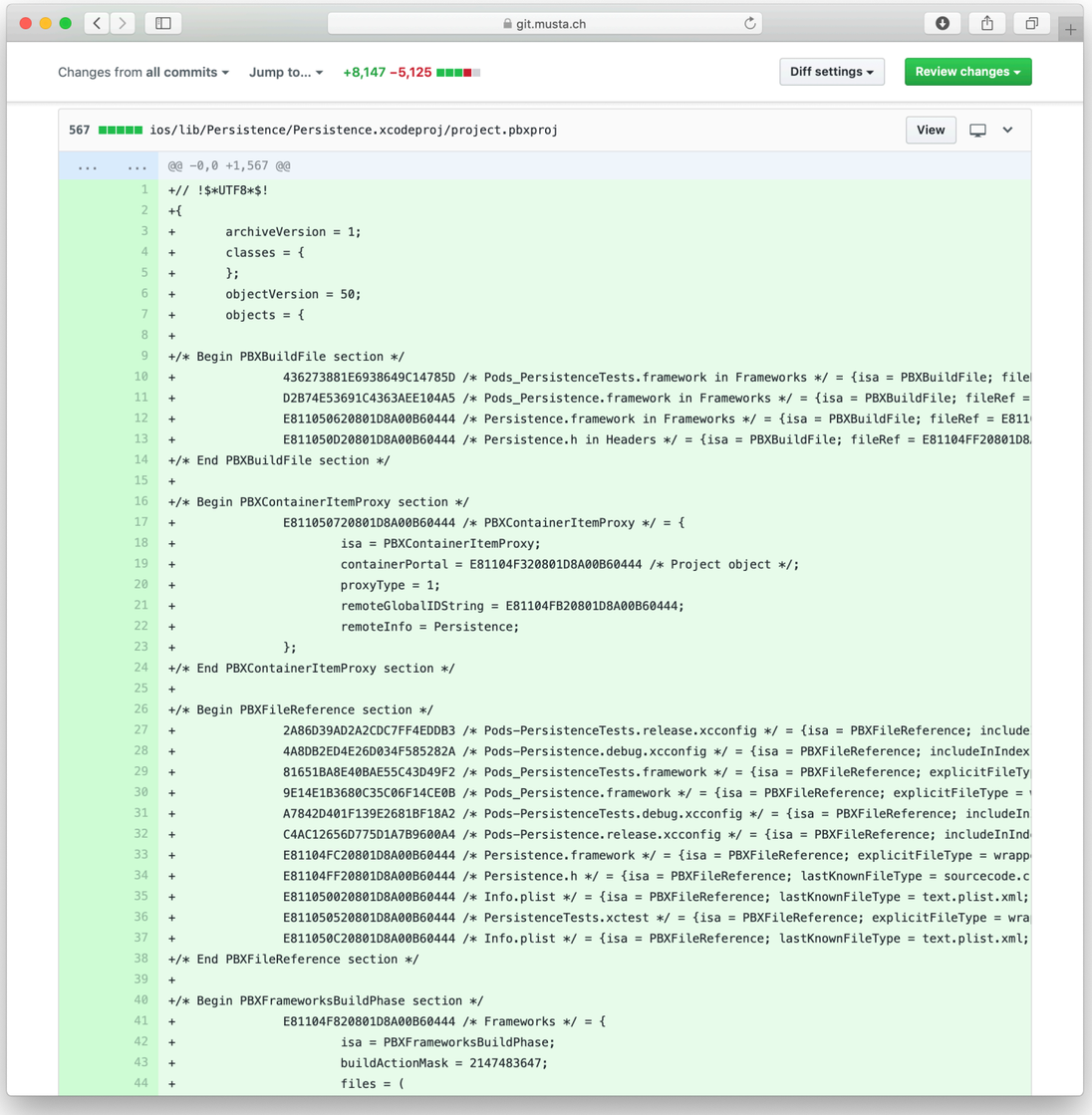
> *Currently all of our modules are frameworks.*



Field	Value	Additional commentary
Name	Wildcat	Don't prefix with Airbnb anymore
Team	None	This will be inherited from shared `.xcconfig` files, eg this
Organization Name	Airbnb, Inc.	
Organization Identifier	com.airbnb	
Language	Swift	🎉
Include Unit Tests	`true`	You know you should 😄

For the next dialog:

- Our modules live under **ios/lib/**


git.musta.ch
Changes from all commits
Jump to...
+8,147 −5,125
Diff settings
Review changes
567 ios/lib/Persistence/Persistence.xcodeproj/project.pbxproj
View
@@ −0,0 +1,567 @@
+// !$*UTF8*$!
+{
+ archiveVersion = 1;
+ classes = {
+ };
+ objectVersion = 50;
+ objects = {
+
+/* Begin PBXBuildFile section */
+ 436273881E6938649C14785D /* Pods_PersistenceTests.framework in Frameworks */ = {isa = PBXBuildFile; file
+ D2B74E53691C4363AEE104A5 /* Pods_Persistence.framework in Frameworks */ = {isa = PBXBuildFile; fileRef =
+ E811050620801D8A00B60444 /* Persistence.framework in Frameworks */ = {isa = PBXBuildFile; fileRef = E811
+ E811050D20801D8A00B60444 /* Persistence.h in Headers */ = {isa = PBXBuildFile; fileRef = E81104FF20801D8
+/* End PBXBuildFile section */
+
+/* Begin PBXContainerItemProxy section */
+ E811050720801D8A00B60444 /* PBXContainerItemProxy */ = {
+ isa = PBXContainerItemProxy;
+ containerPortal = E81104F320801D8A00B60444 /* Project object */;
+ proxyType = 1;
+ remoteGlobalIDString = E81104FB20801D8A00B60444;
+ remoteInfo = Persistence;
+ };
+/* End PBXContainerItemProxy section */
+
+/* Begin PBXFileReference section */
+ 2A86D39AD2A2CDC7FF4EDDB3 /* Pods-PersistenceTests.release.xcconfig */ = {isa = PBXFileReference; include
+ 4A8DB2ED4E26D034F585282A /* Pods-Persistence.debug.xcconfig */ = {isa = PBXFileReference; includeInIndex
+ 81651BA8E40BAE55C43D49F2 /* Pods_PersistenceTests.framework */ = {isa = PBXFileReference; explicitFileTy
+ 9E14E1B3680C35C06F14CE0B /* Pods_Persistence.framework */ = {isa = PBXFileReference; explicitFileType =
+ A7842D401F139E2681BF18A2 /* Pods-PersistenceTests.debug.xcconfig */ = {isa = PBXFileReference; includeIn
+ C4AC12656D775D1A7B9600A4 /* Pods-Persistence.release.xcconfig */ = {isa = PBXFileReference; includeInInd
+ E81104FC20801D8A00B60444 /* Persistence.framework */ = {isa = PBXFileReference; explicitFileType = wrapp
+ E81104FF20801D8A00B60444 /* Persistence.h */ = {isa = PBXFileReference; lastKnownFileType = sourcecode.c
+ E811050020801D8A00B60444 /* Info.plist */ = {isa = PBXFileReference; lastKnownFileType = text.plist.xml;
+ E811050520801D8A00B60444 /* PersistenceTests.xctest */ = {isa = PBXFileReference; explicitFileType = wra
+ E811050C20801D8A00B60444 /* Info.plist */ = {isa = PBXFileReference; lastKnownFileType = text.plist.xml;
+/* End PBXFileReference section */
+
+/* Begin PBXFrameworksBuildPhase section */
+ E81104F820801D8A00B60444 /* Frameworks */ = {
+ isa = PBXFrameworksBuildPhase;
+ buildActionMask = 2147483647;
+ files = (

```
rake
make:module
```

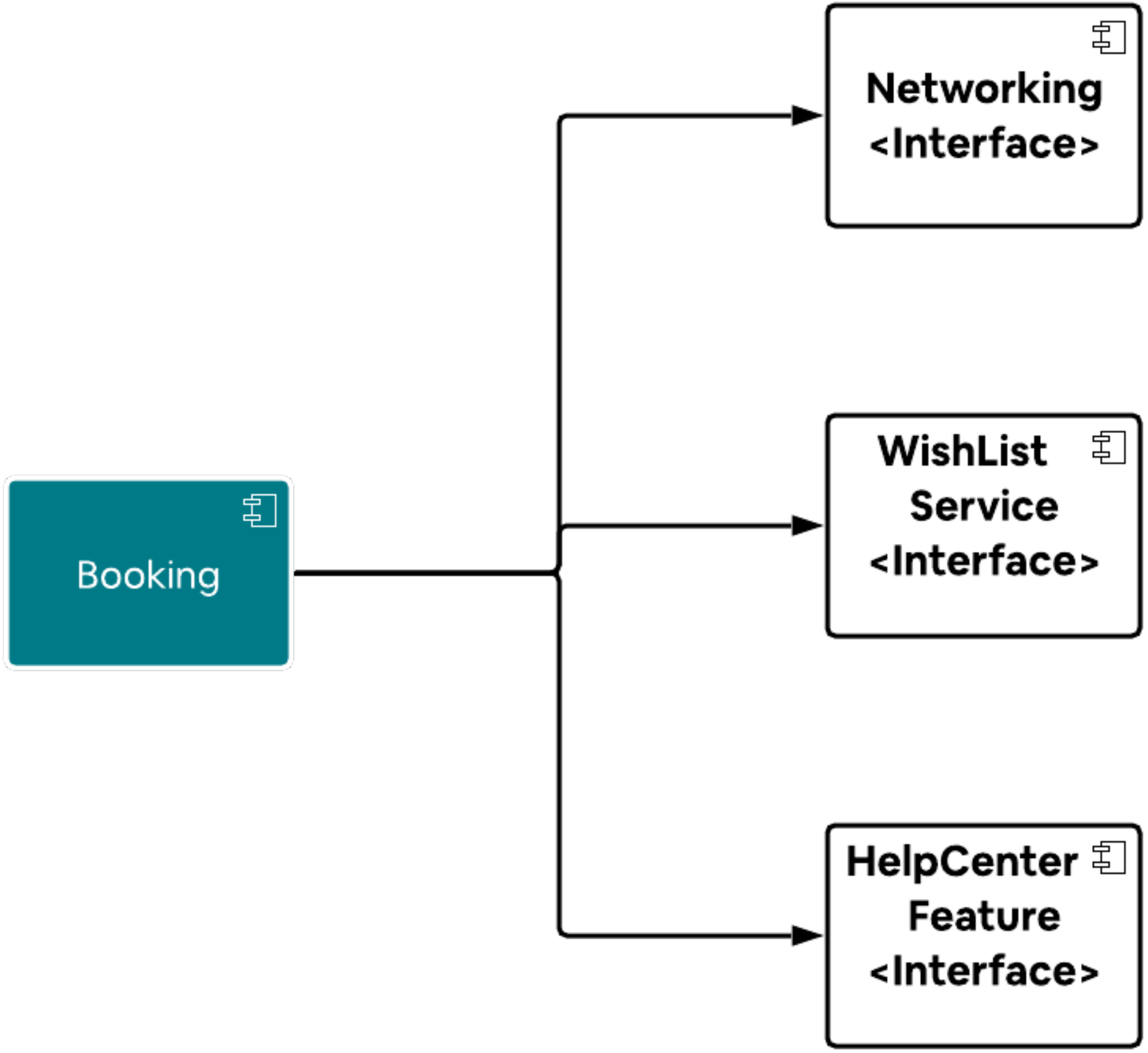
```
> Provide the type of module you want to create:
    1: Non Platform
    2: Feature
    3: Feature Interface
    4: Service
    5: Service Interface
4

> New module name:
Swiftable

> Provide a high level description of this module:
This is a module to present at Swiftable
```

Buck

Human readable dependencies

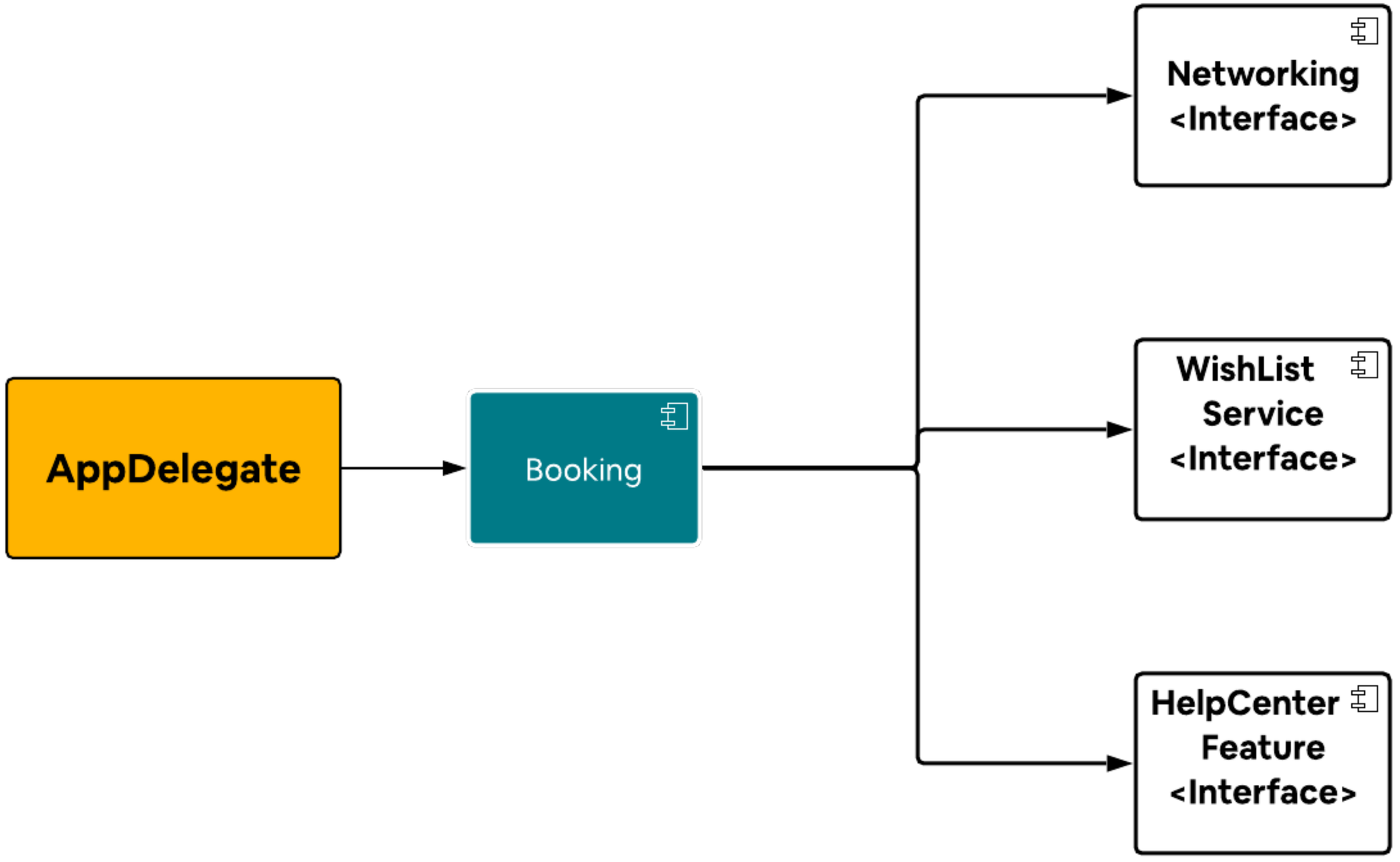
https://github.com/airbnb/BuckSample

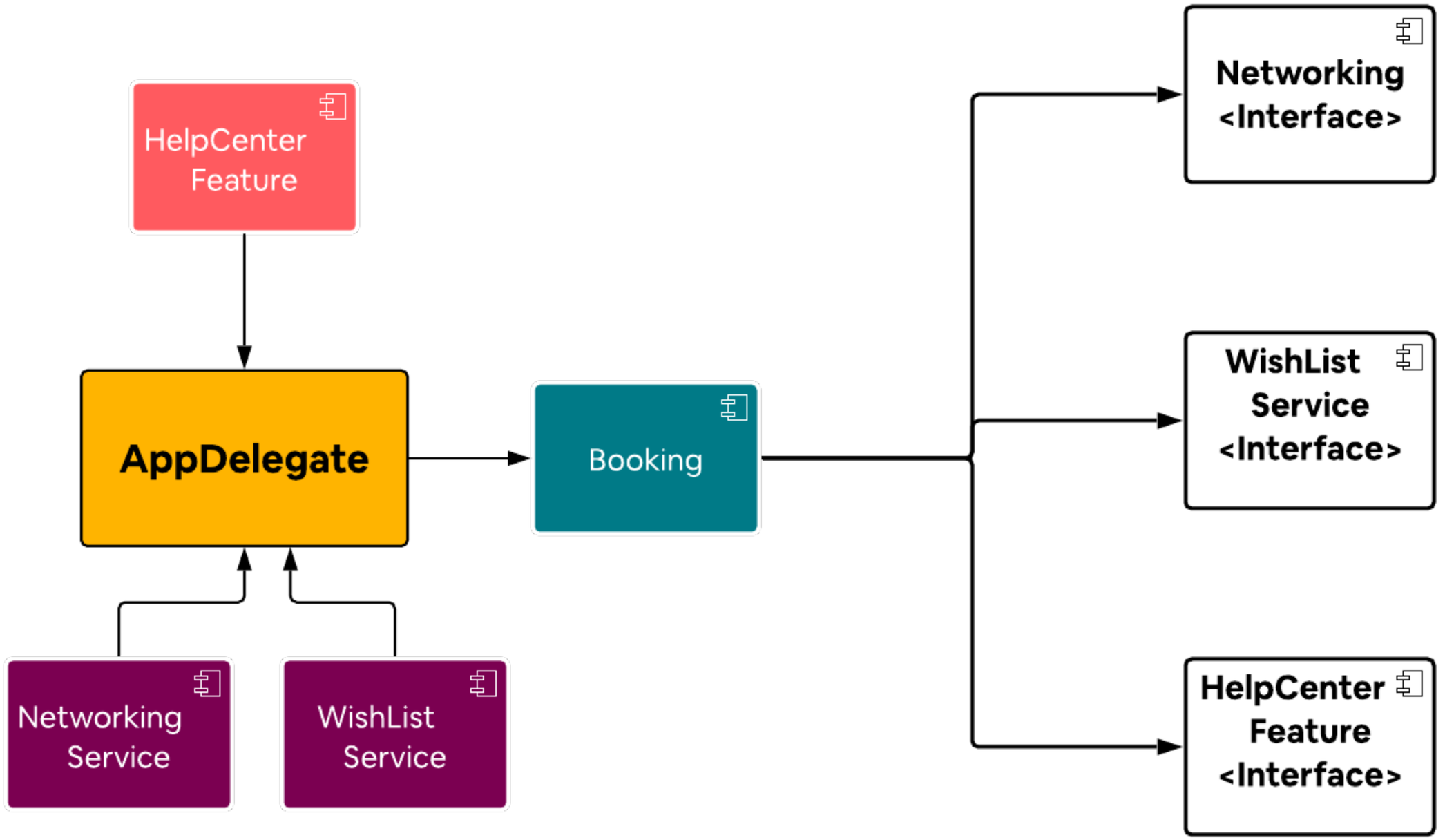
```
feature(
    name = "Booking",
    deps = [
        "//ios/service_interfaces/Networking",
        "//ios/service_interfaces/WishListService",
        "//ios/feature_interfaces/HelpCenter",
    ],
)
```

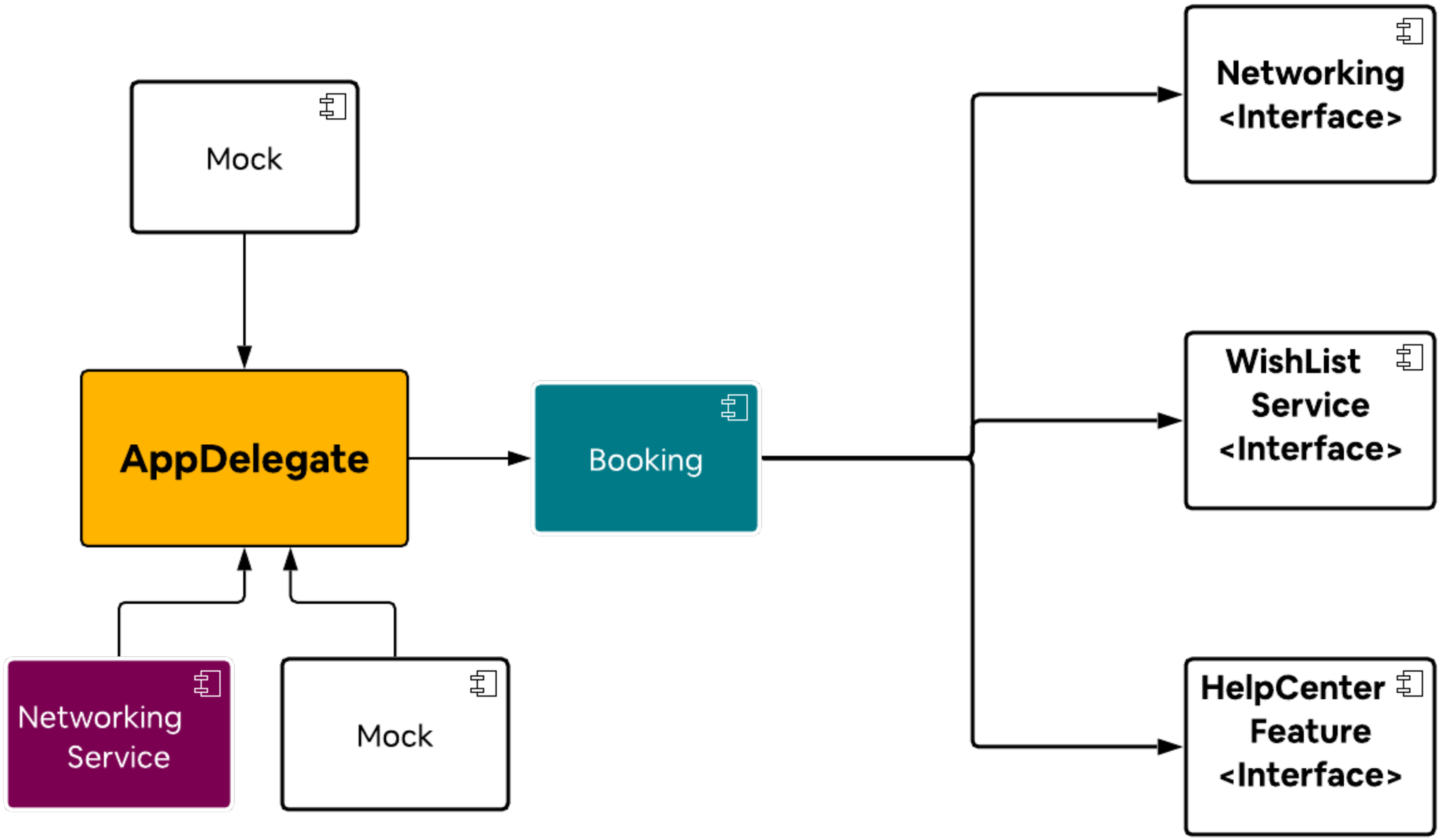

Networking
<Interface>
Booking
WishList
Service
<Interface>
HelpCenter
Feature
<Interface>

Feature

A screen or a flow in the app


AppDelegate
Booking
Networking
<Interface>
WishList
Service
<Interface>
HelpCenter
Feature
<Interface>


HelpCenter Feature
AppDelegate
Booking
Networking Service
WishList Service
Networking <Interface>
WishList Service <Interface>
HelpCenter Feature <Interface>


Mock
AppDelegate
Booking
Networking
<Interface>
WishList
Service
<Interface>
HelpCenter
Feature
<Interface>
Networking
Service
Mock

Dev Apps

~1 min

Dev Apps

Big buckets

Small

playgrounds

How do we get there?

~100 modules

One module type: `/libraries`

libraries/AirbnbBooking
libraries/AirbnbBusinessTravel
libraries/AirbnbHelpCenter
libraries/AirbnbListings
libraries/AirbnbNetworking
libraries/AirbnbWishLists
...

Before iOS Platform

/libraries

On the iOS Platform

/services
/service_interfaces
/features
/feature_interfaces

How to get everybody on the iOS Platform?

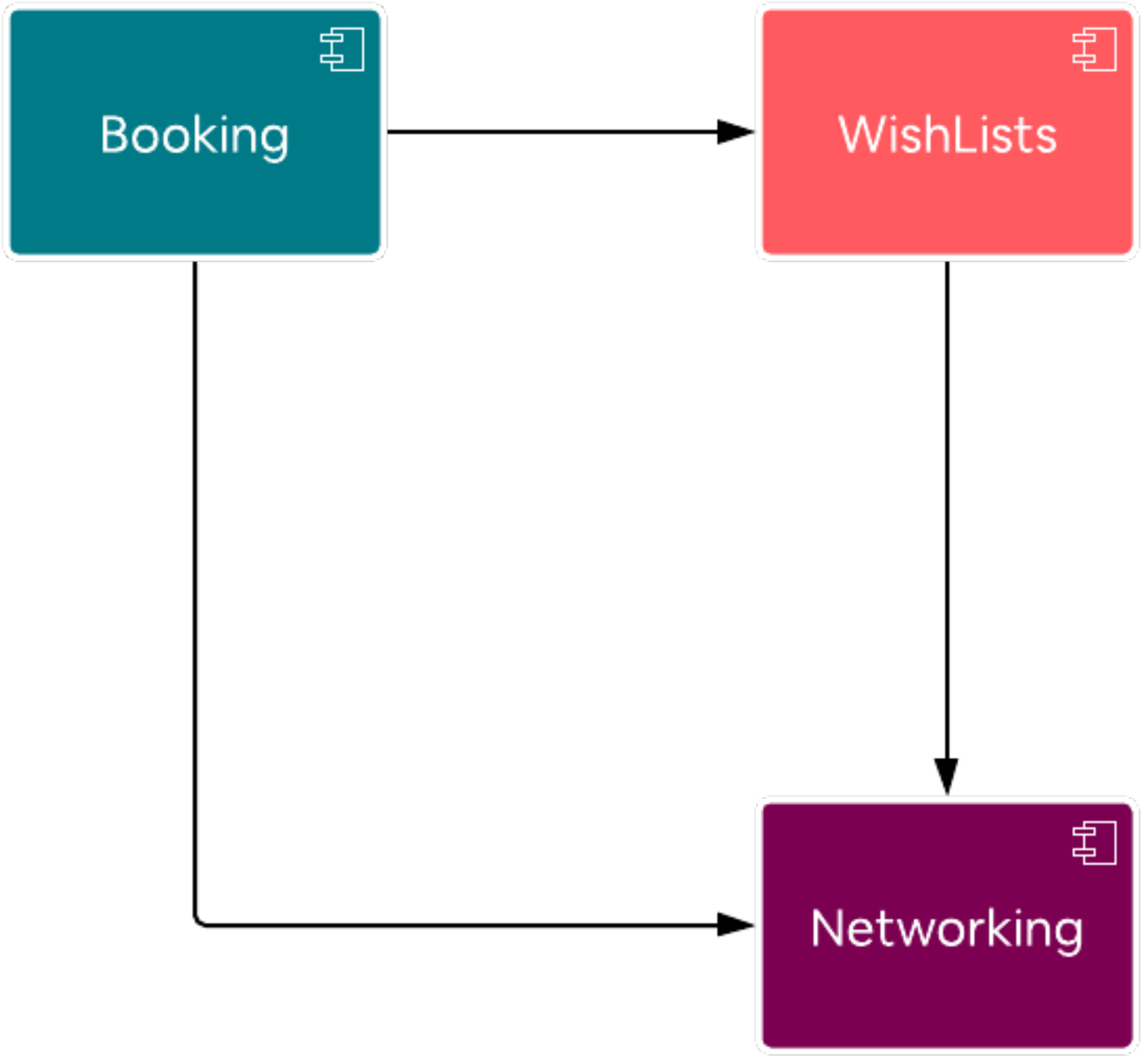
Remove `libraries/` and start over

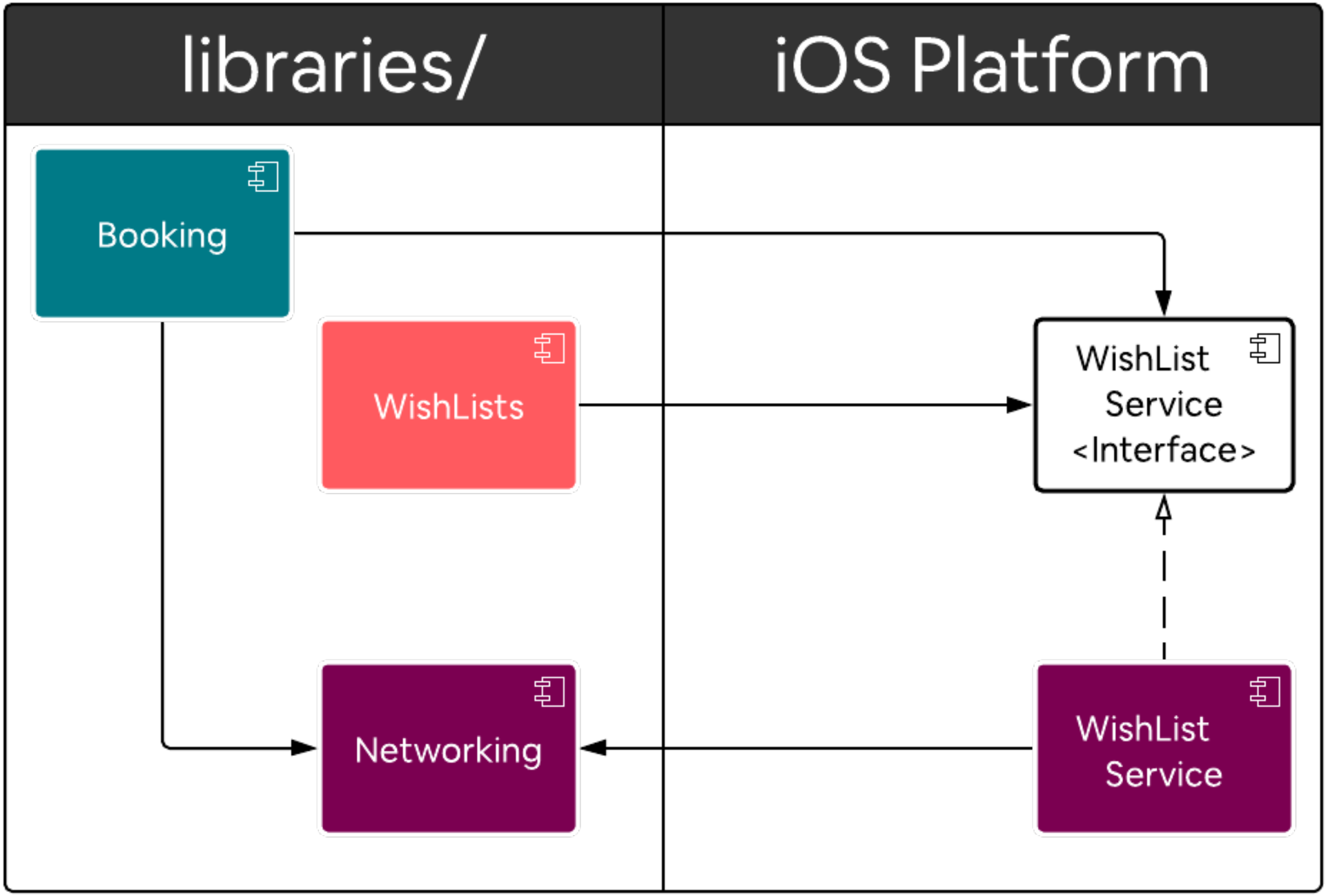
Progressively migrate

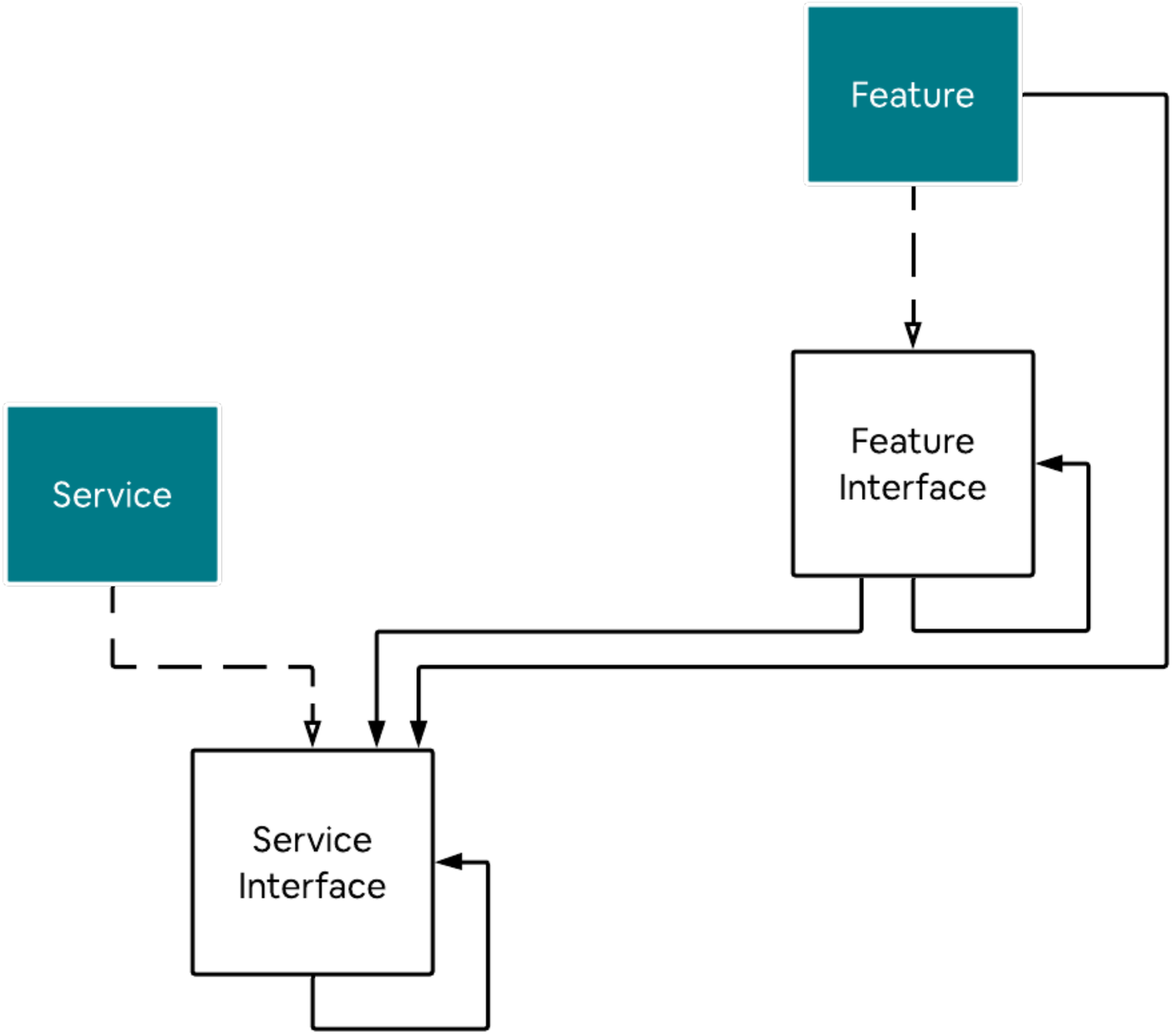
Let's migrate

WishLists Data Source

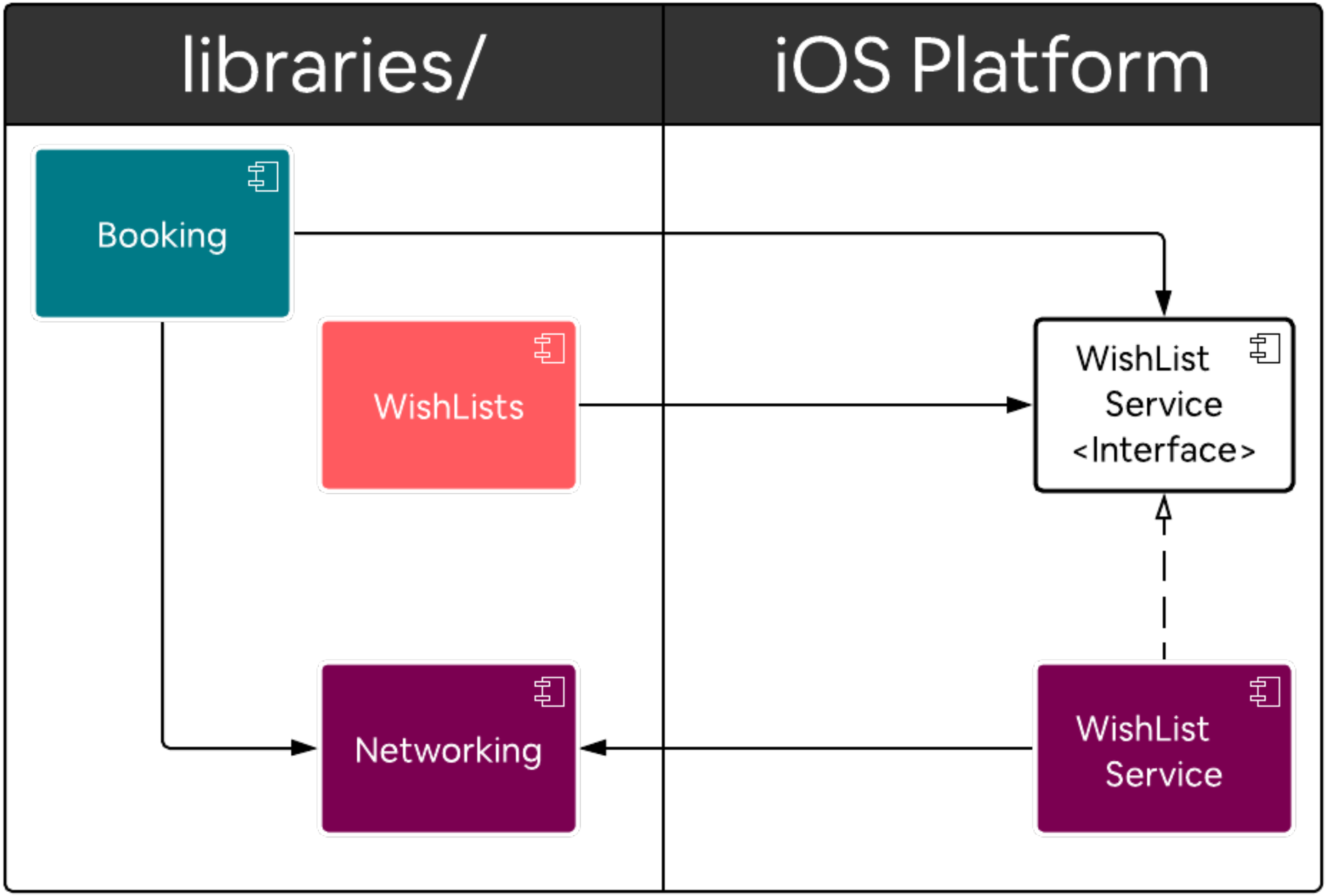
libraries/WishLists


Booking
WishLists
Networking


libraries/
iOS Platform
Booking
WishLists
Networking
WishList
Service
<Interface>
WishList
Service


Feature
Feature Interface
Service
Service Interface

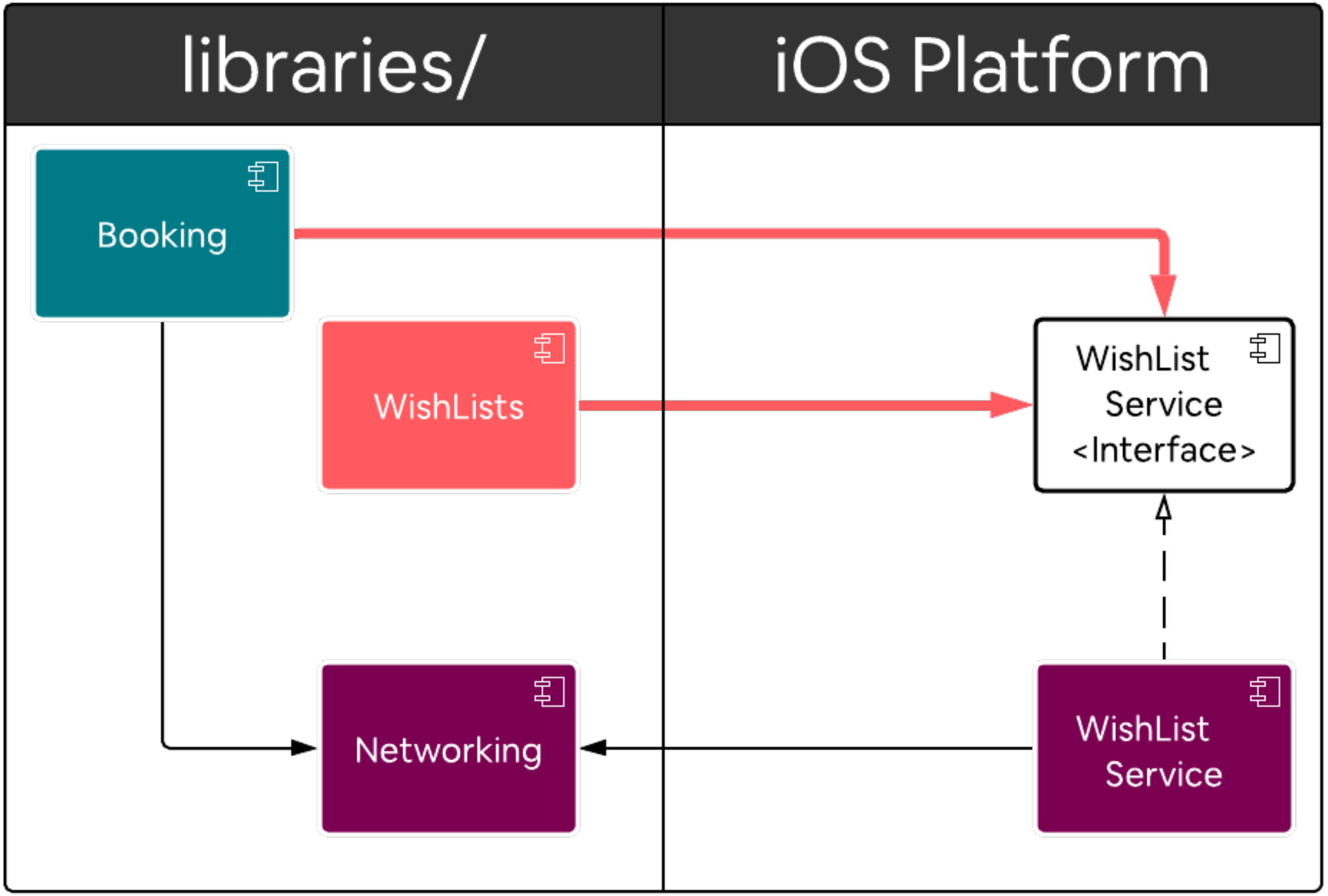
What are the dependency rules for `libraries/`?


libraries/
iOS Platform
Booking
WishLists
Networking
WishList
Service
<Interface>
WishList
Service

Inbound dependencies
Outbound dependencies

Inbound dependencies

Outbound dependencies


libraries/
iOS Platform
Booking
WishLists
Networking
WishList
Service
<Interface>
WishList
Service

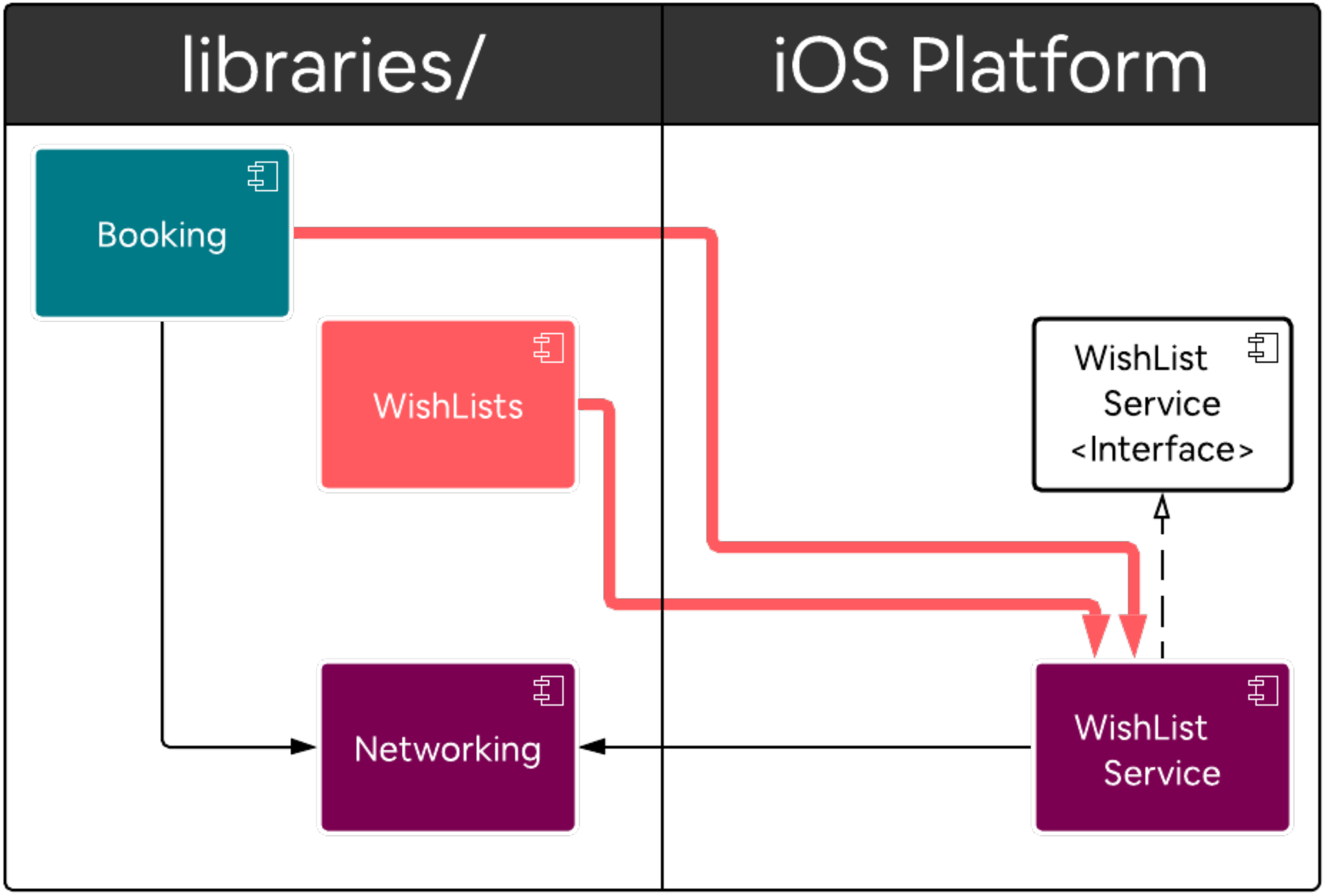
The interface module has

stricter rules

Migrate all the call sites

As the owner of WishLists

We don't control these

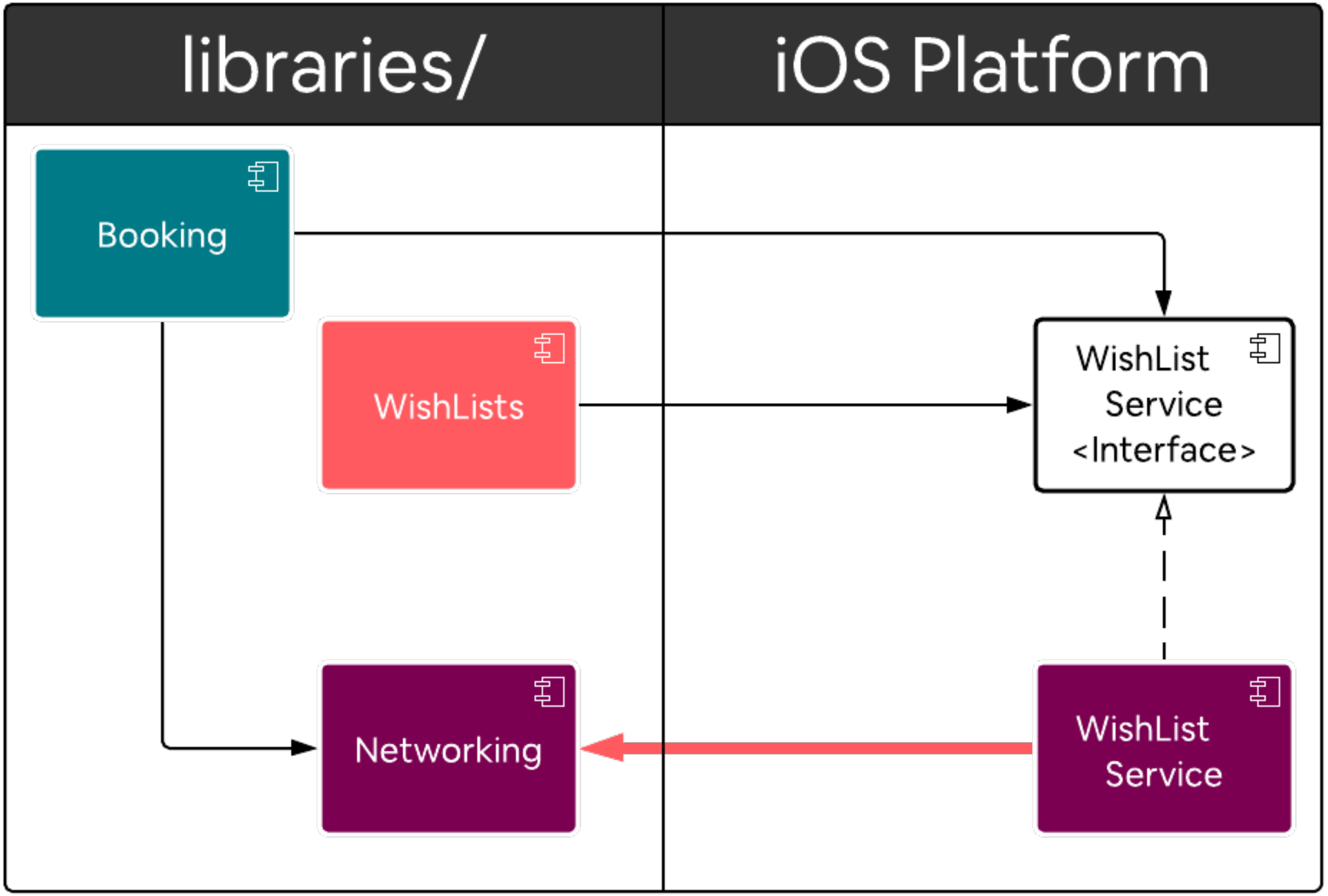

libraries/
iOS Platform
Booking
WishLists
Networking
WishList
Service
<Interface>
WishList
Service

Calculated

tech debt

Inbound dependencies

Outbound dependencies


libraries/
iOS Platform
Booking
WishLists
Networking
WishList
Service
<Interface>
WishList
Service

we know our usage

of Networking

We control our dependencies

Allow inbound dependencies from `libraries/`

Don't allow outbound dependencies to `libraries/`

`libraries/` has access to the iOS Platform

The iOS Platform doesn't have access to `libraries/`

Code on the iOS Platform has good boundaries

while we allow for
easier migration

```
def service_interface(
        name,
        visibility = []):
    max_visibility = [
        "//ios/feature_interfaces/...",
        "//ios/features/...",
        "//ios/service_interfaces/...",
        "//ios/services/...",
    ]

    add_visibility_for_legacy_module_structure(max_visibility)
```

```
def add_visibility_for_legacy_module_structure(visibility):
    visibility.extend([
        "//ios/libraries/...",
    ])
```

We started

migrating

from the bottom up

Try it

ourselves


Pilot
with others teams

Should I follow this?

Most likely:

NO

There's no
silver bullet

Adapt 

To recap:

1. Document best practices
2. Figure out where you're struggling
3. Automate those best practices

Gracias!

franciscodiaz.cl/talks