

**Working
effectively
at scale**

**Francisco
Díaz**

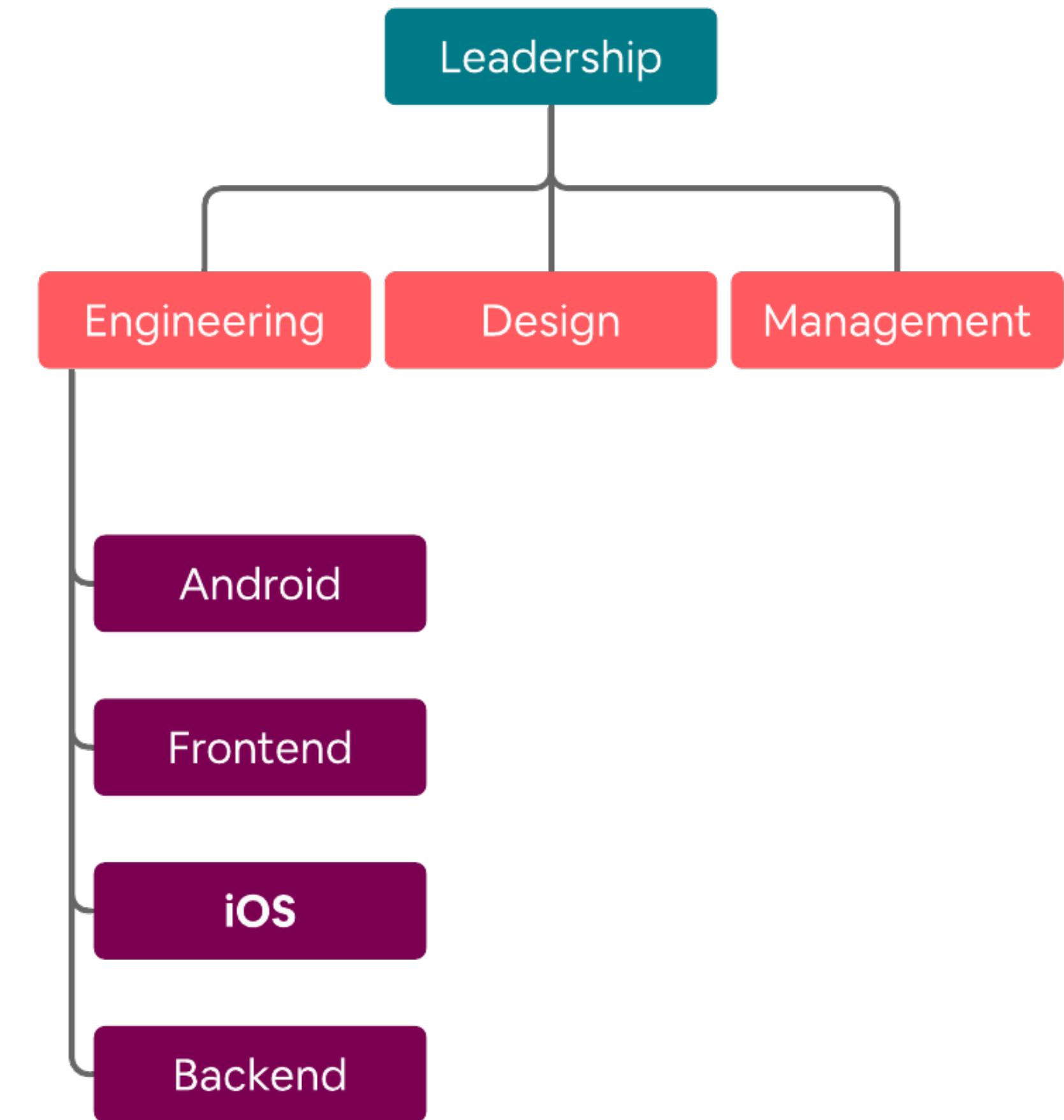
franciscodiaz.cl - @fco_diaz

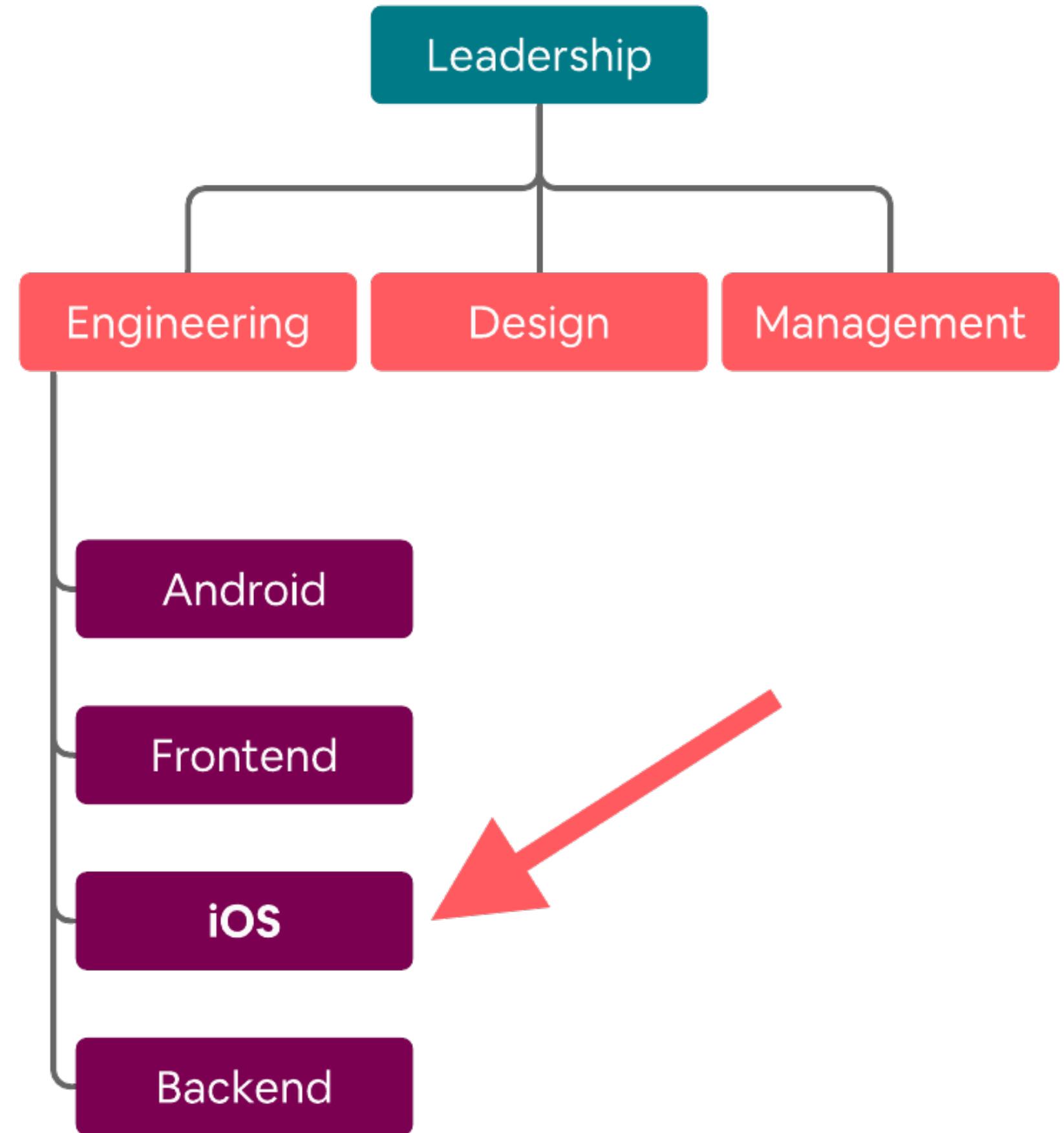
Startups

2011 - 2017

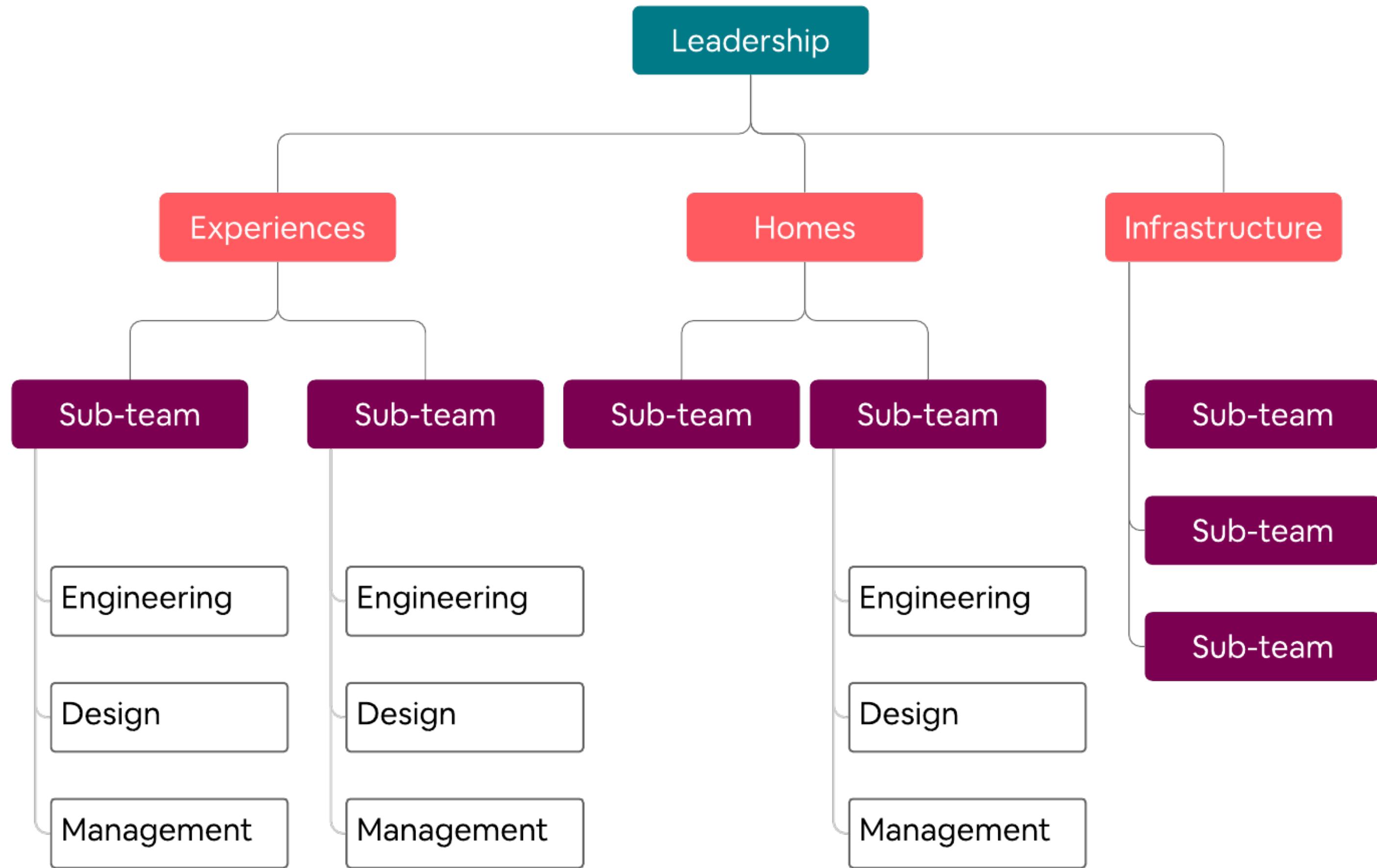
Airbnb

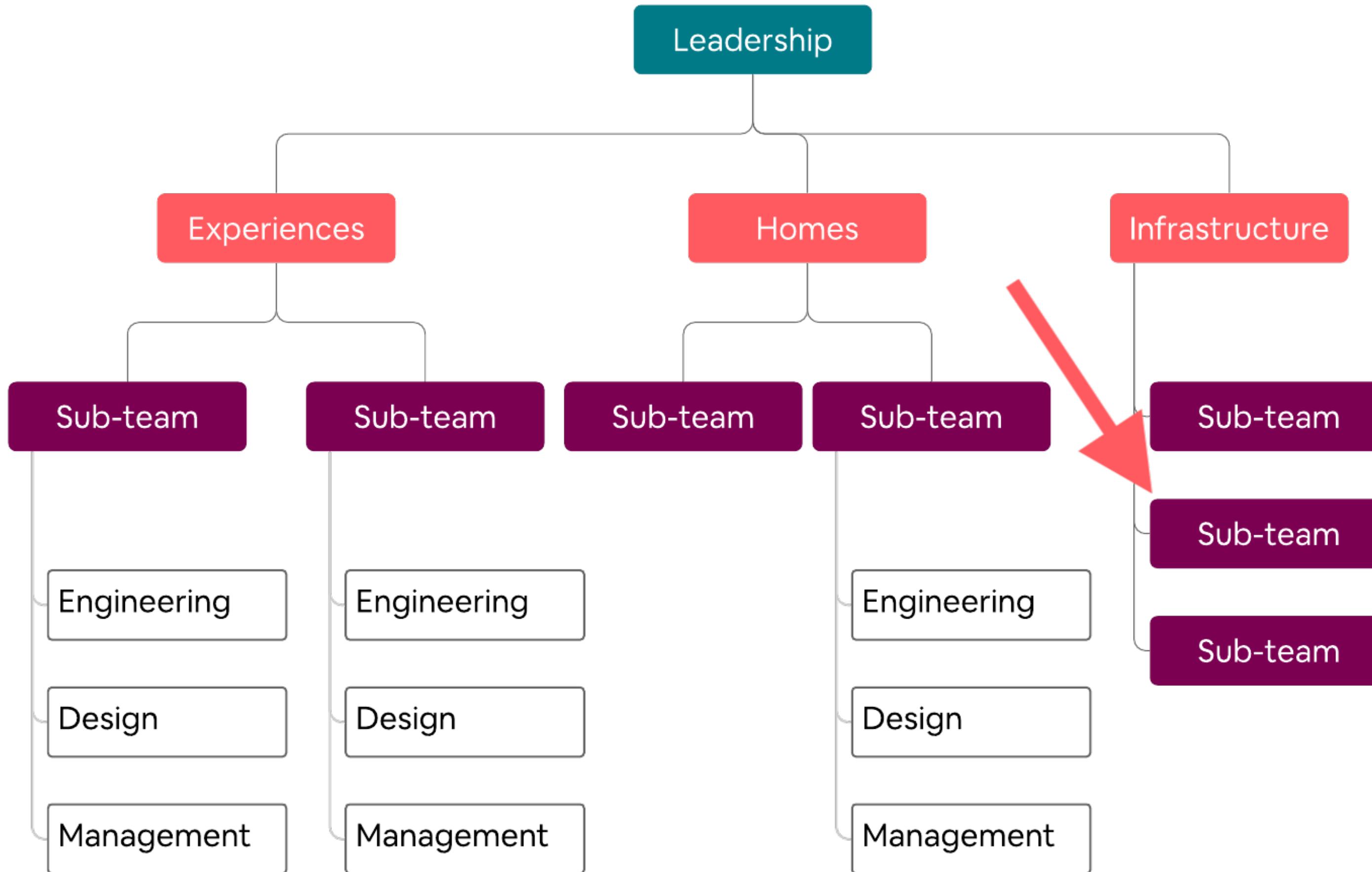
2017 - today



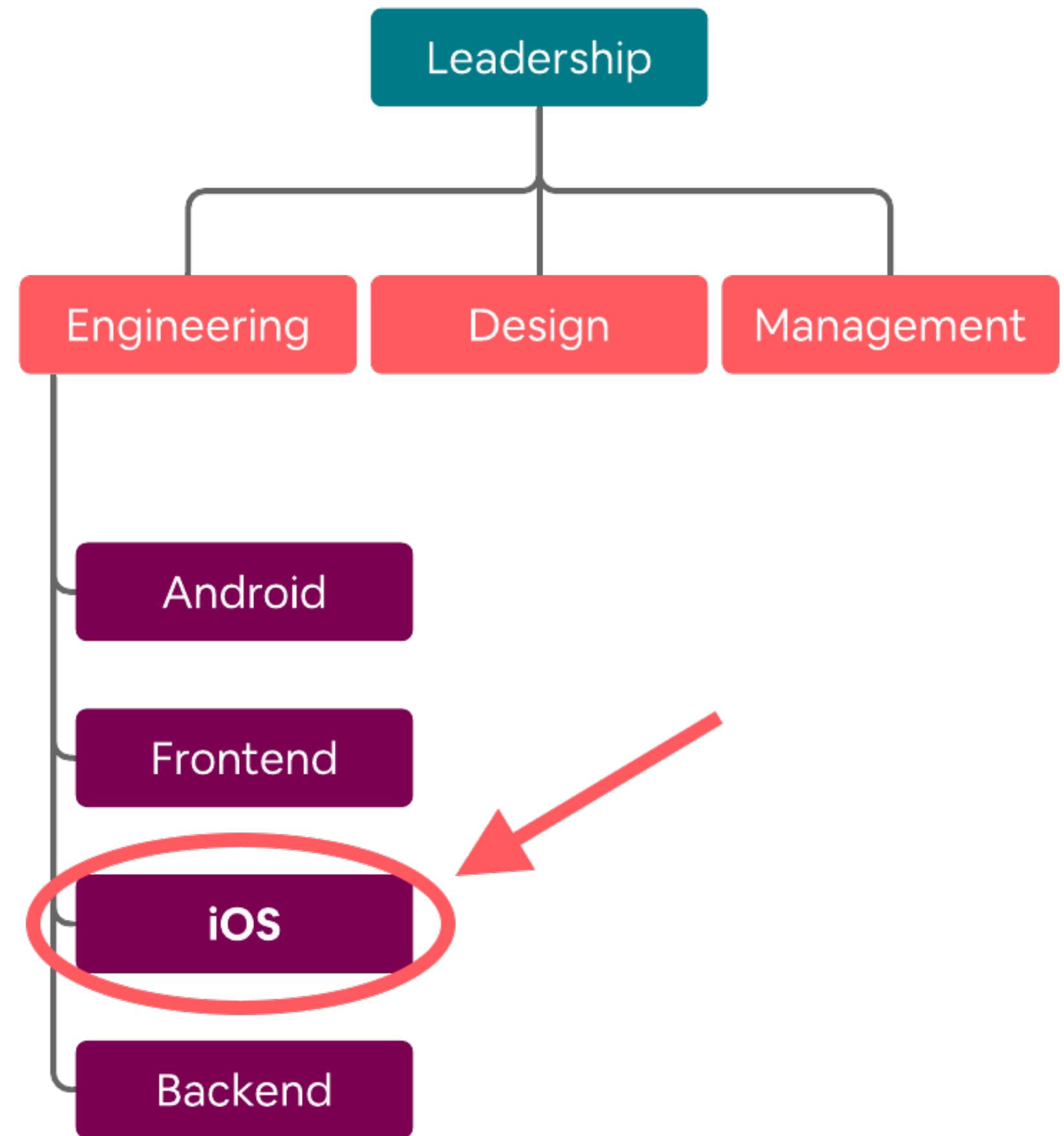


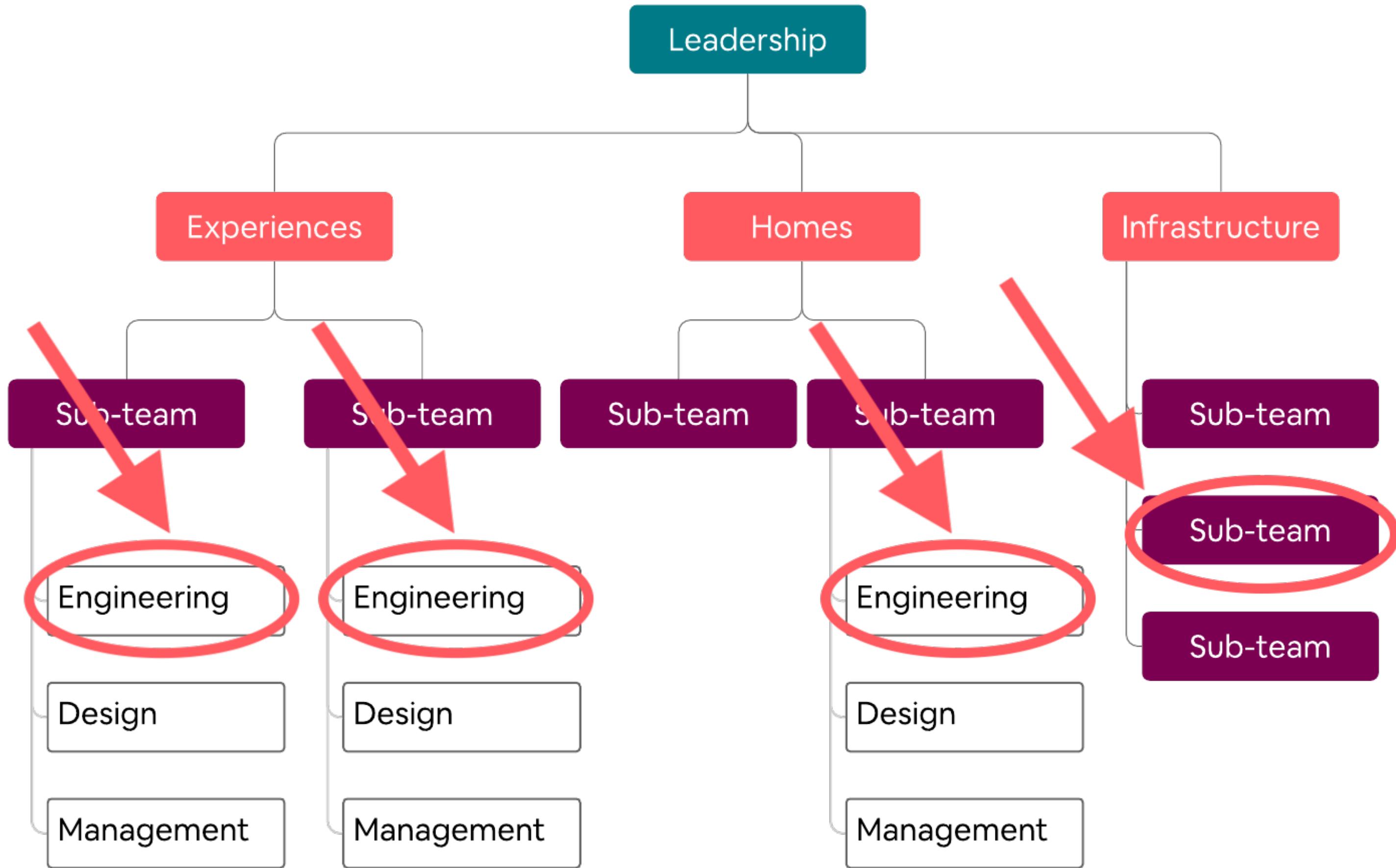














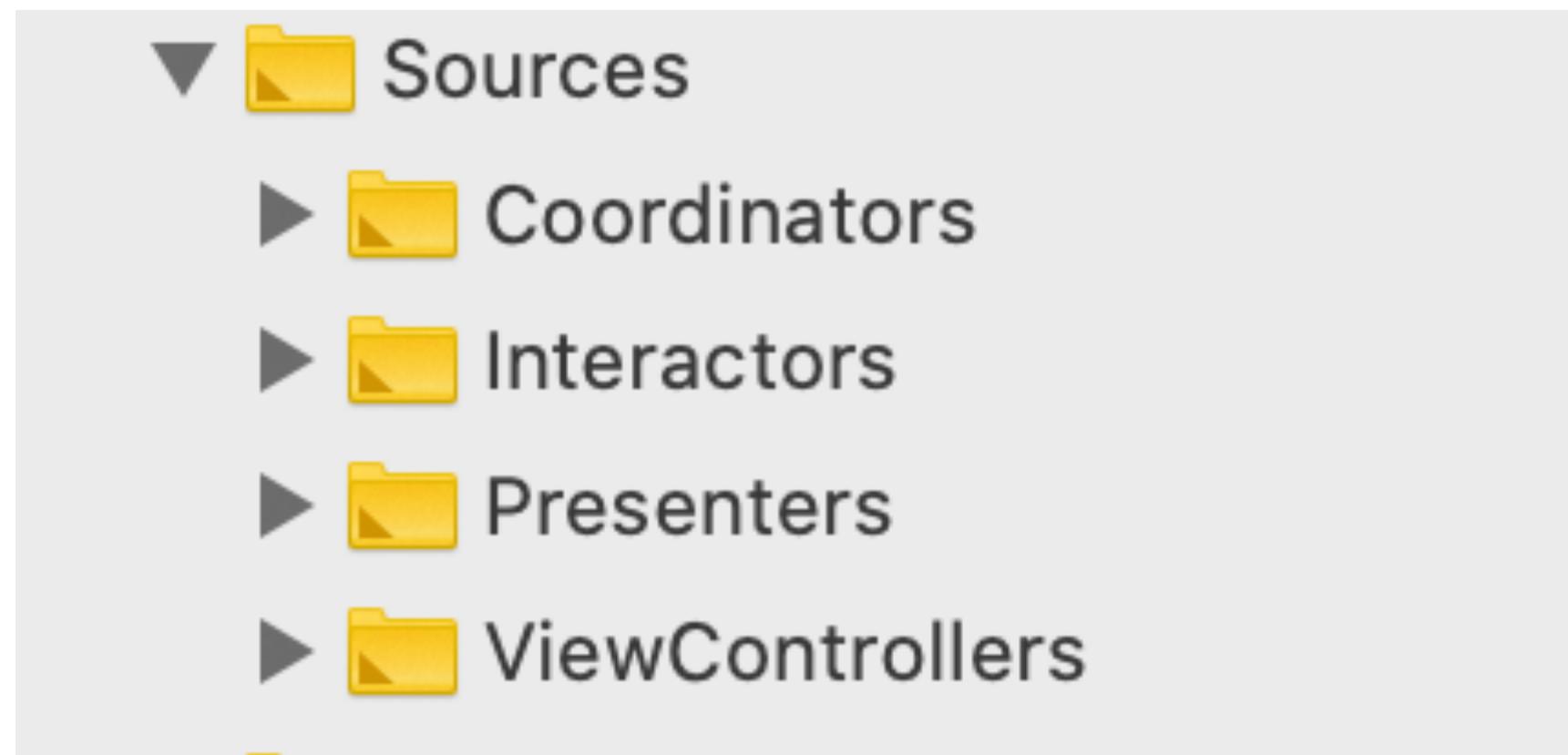
THIS GONE BE A LIL DIFFERENT

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

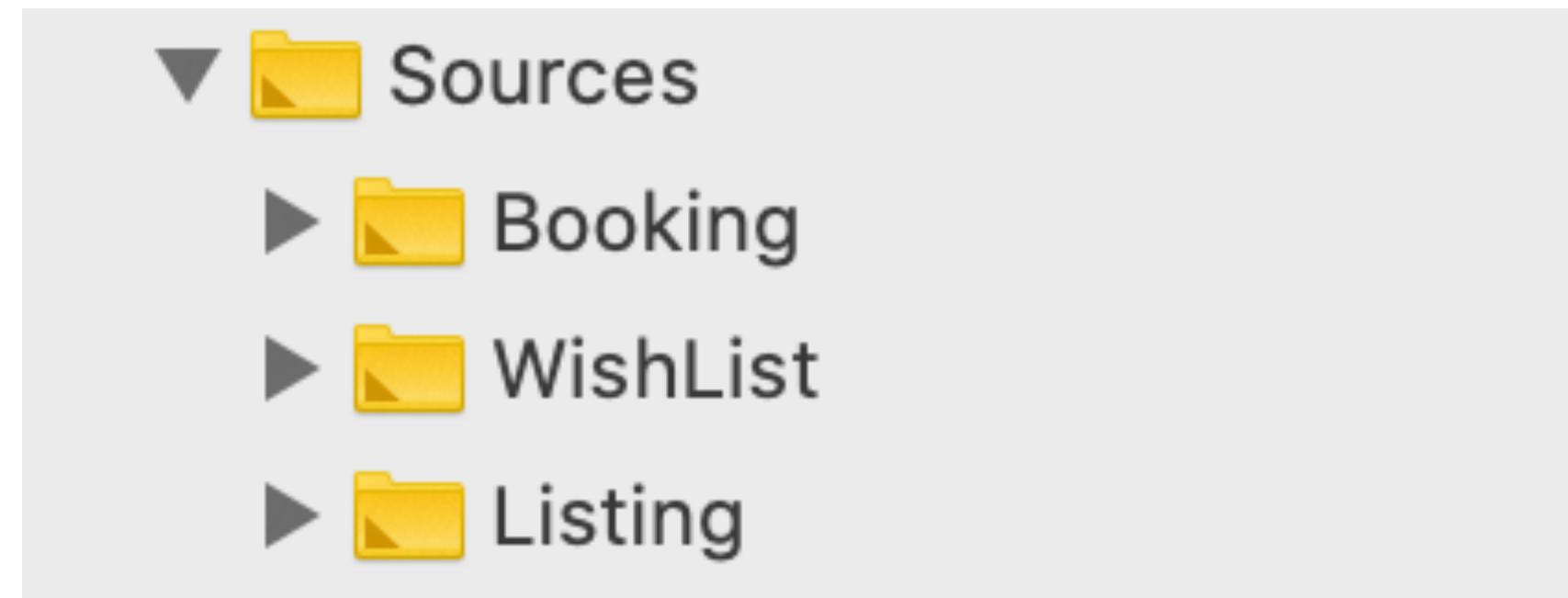
— Conway's law

**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

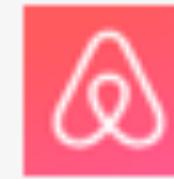
initial commit, all data structures created

1 million lines
of Swift

~80 commits

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

**Bigger
buckets**



Airbnb



AirbnbNetworking



AirbnbBooking



AirbnbWishLists

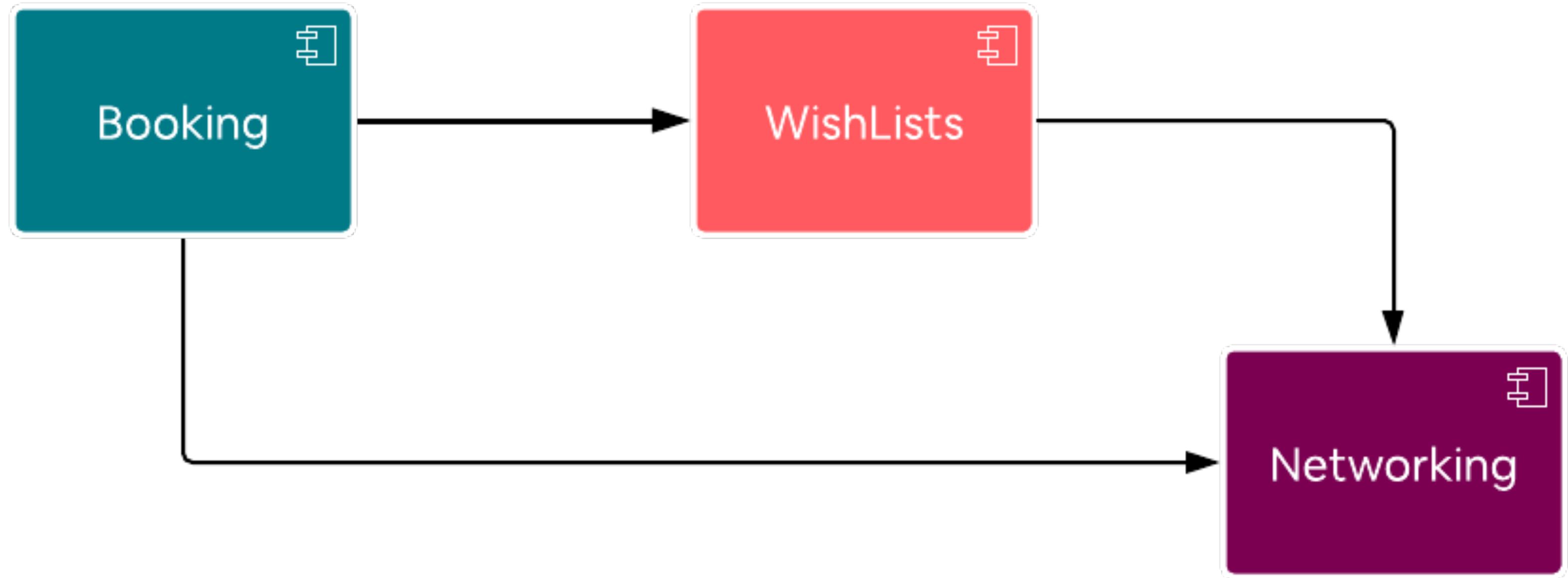


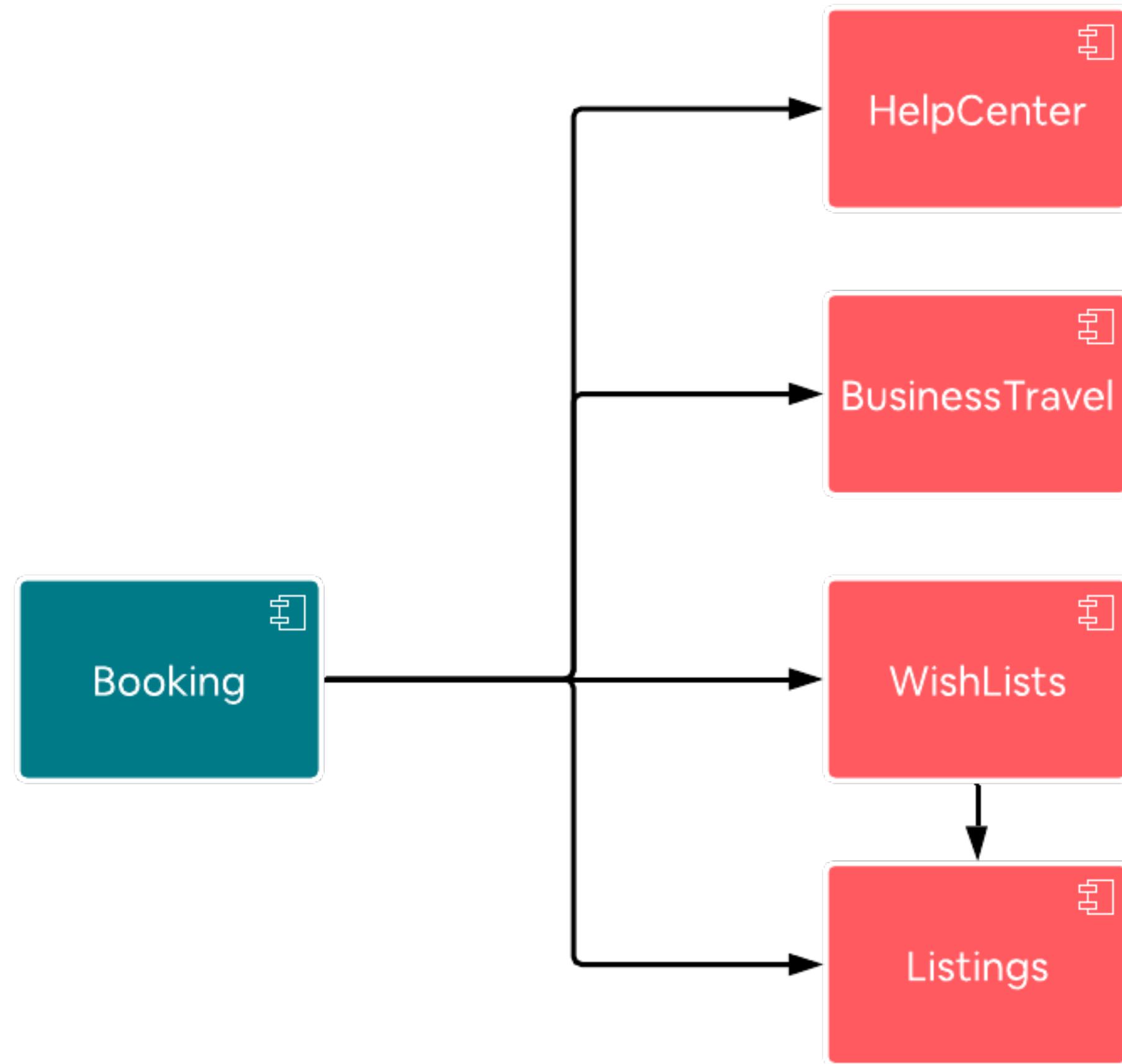
A user should be able to wishlist a listing from the booking flow

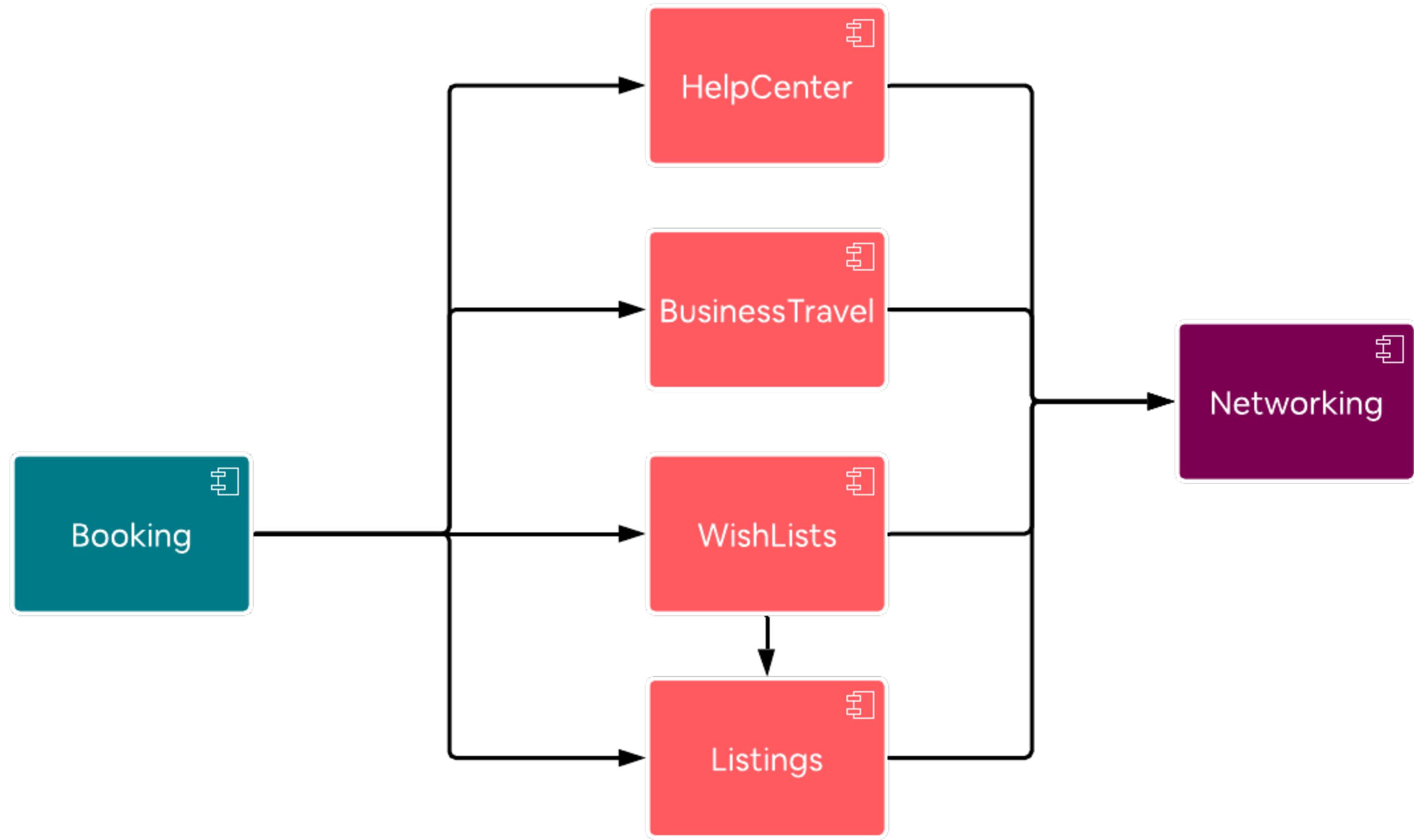
How do they

relate

with each other?







500 main

local clean builds

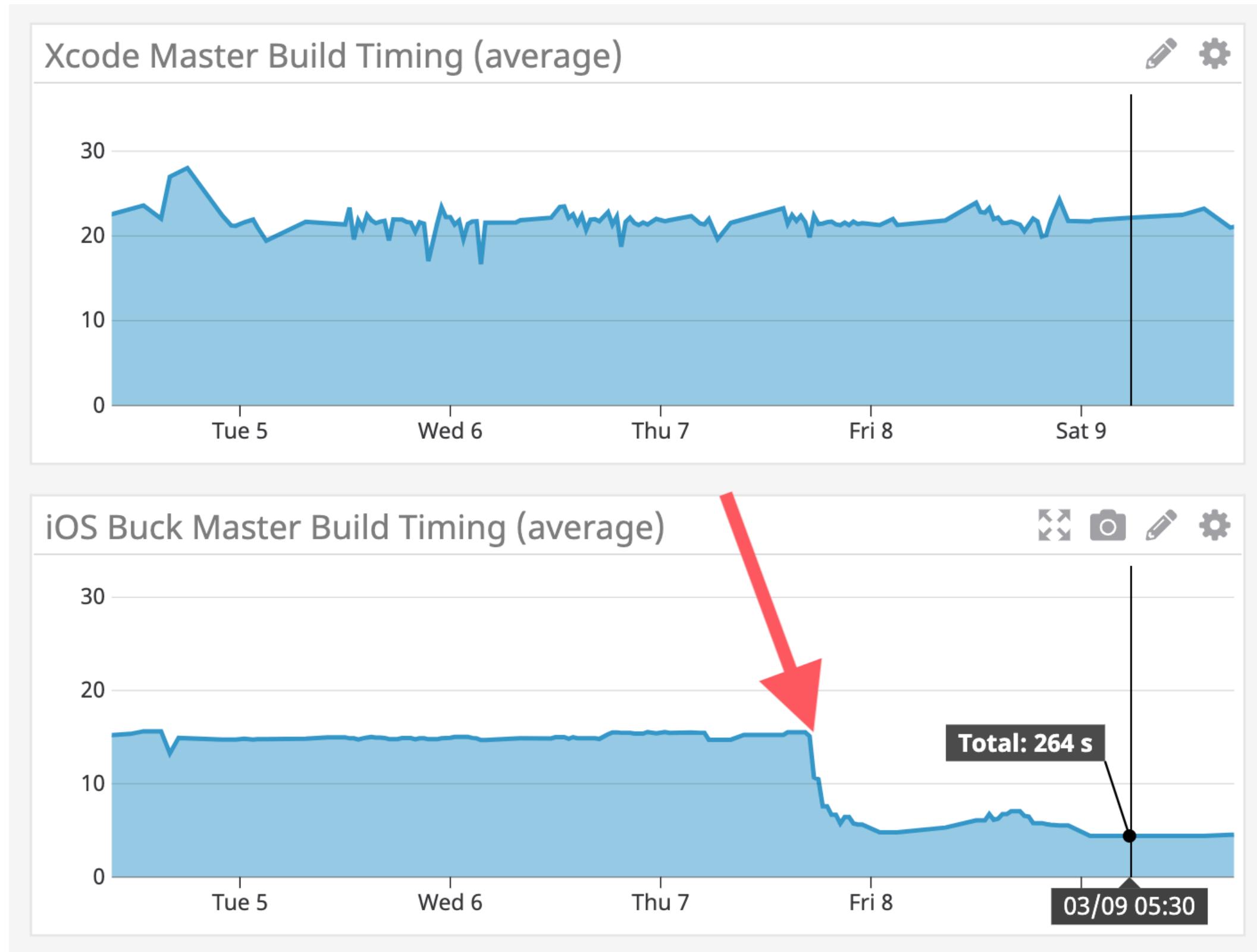
~30min



BUCK

HTTP Cache

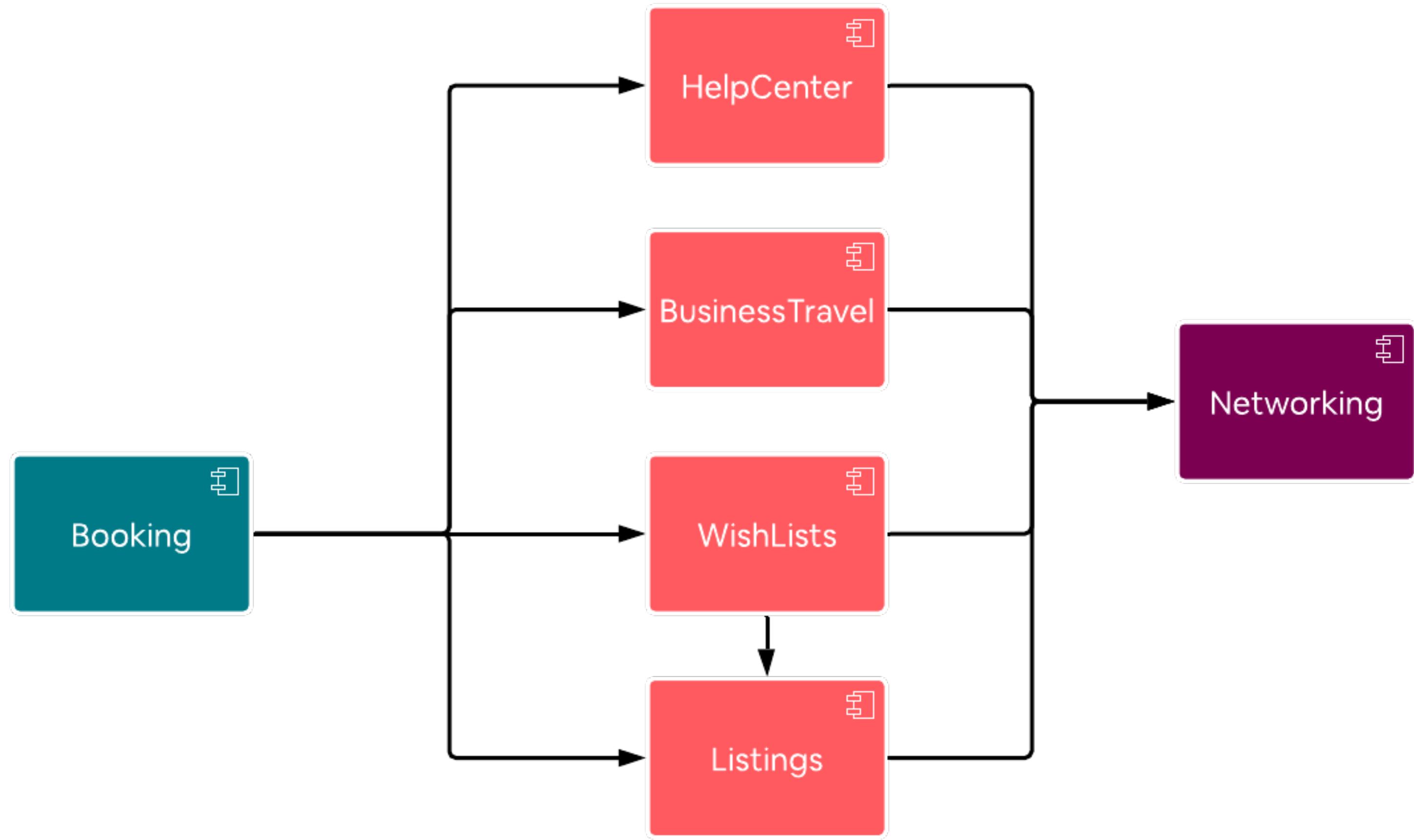
<https://github.com/airbnb/BuckSample>



~5min

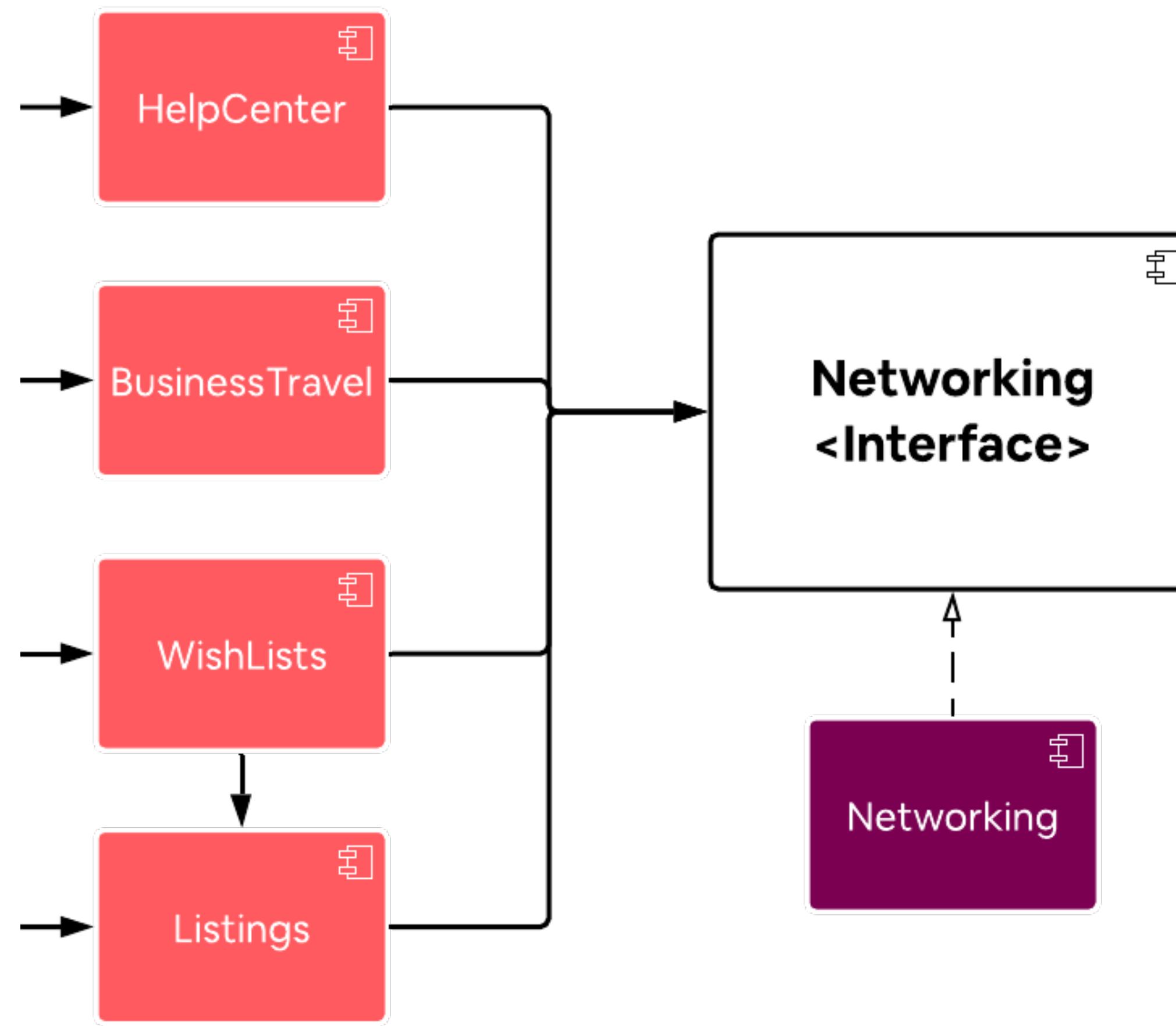




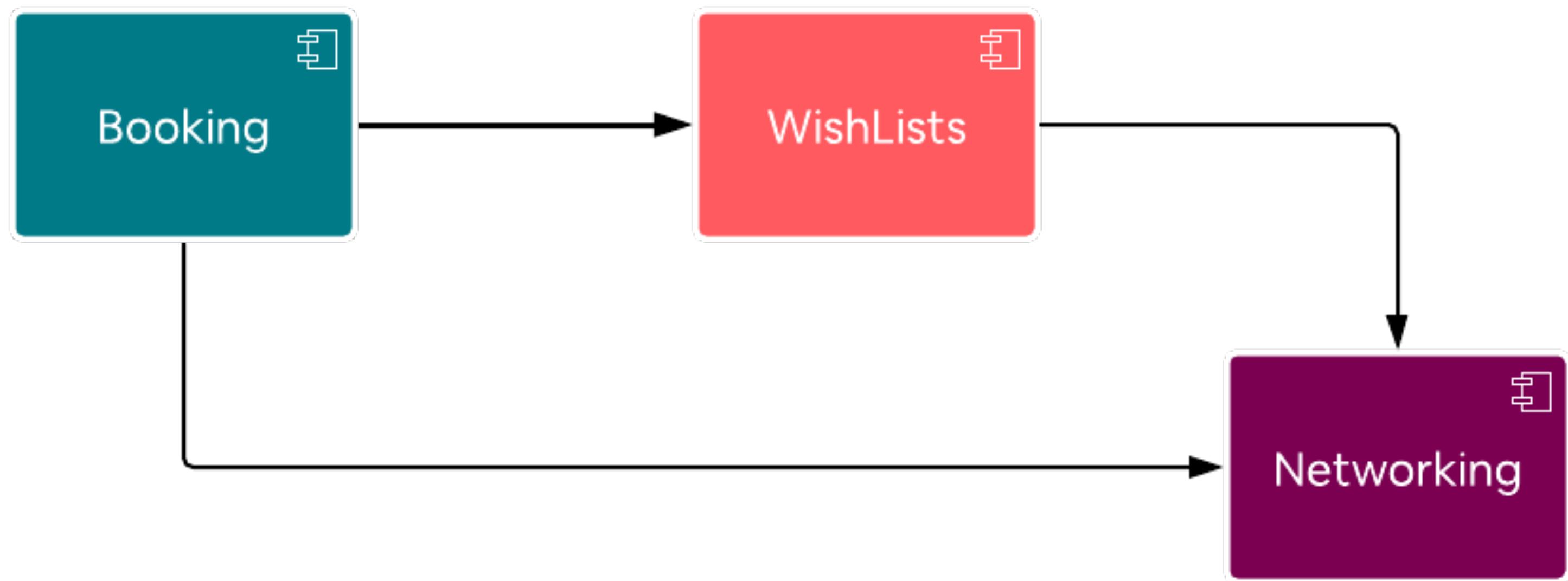


Dependency inversion

- ▶ **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.

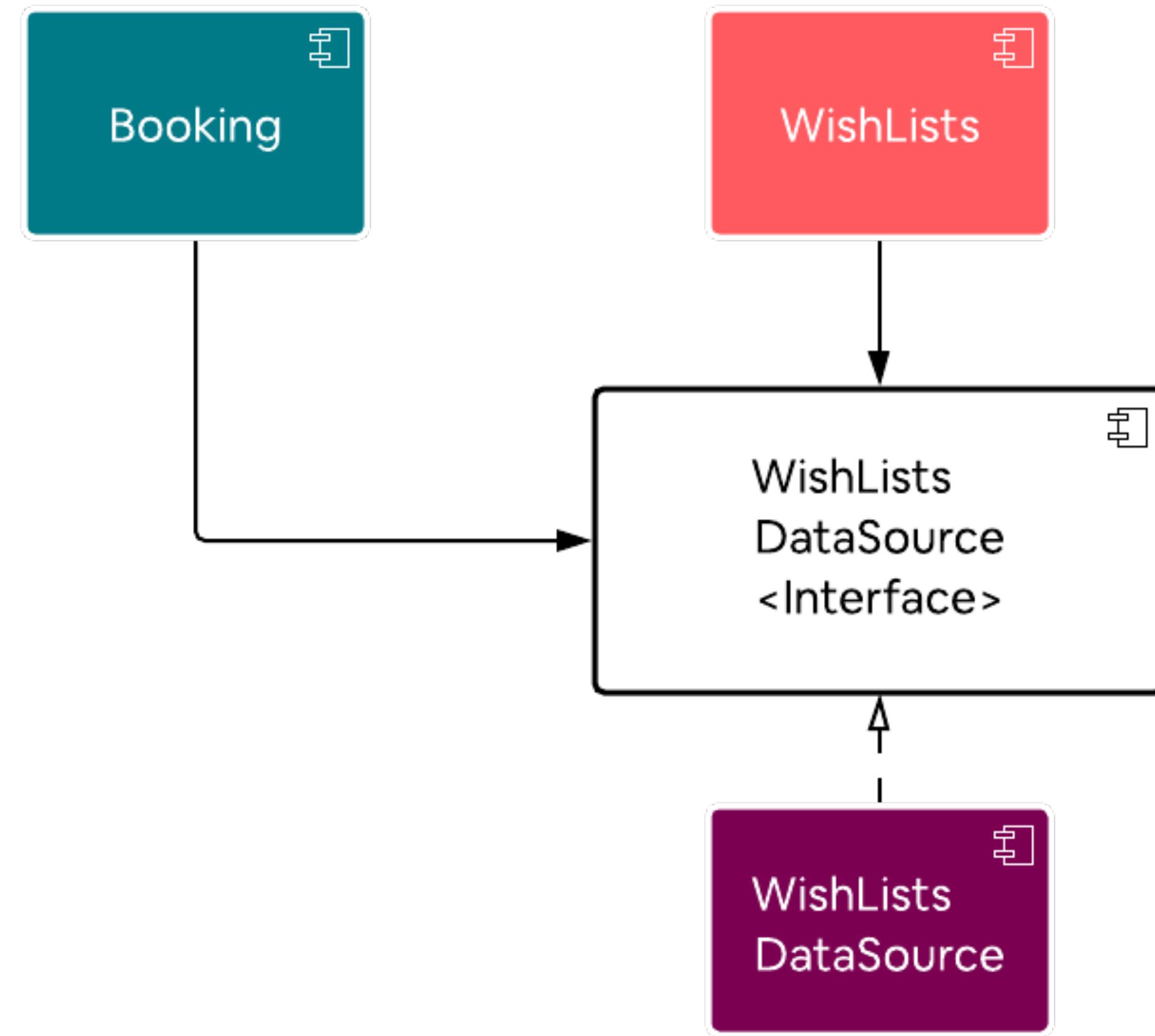


A user should be able to wishlist a listing from the booking flow

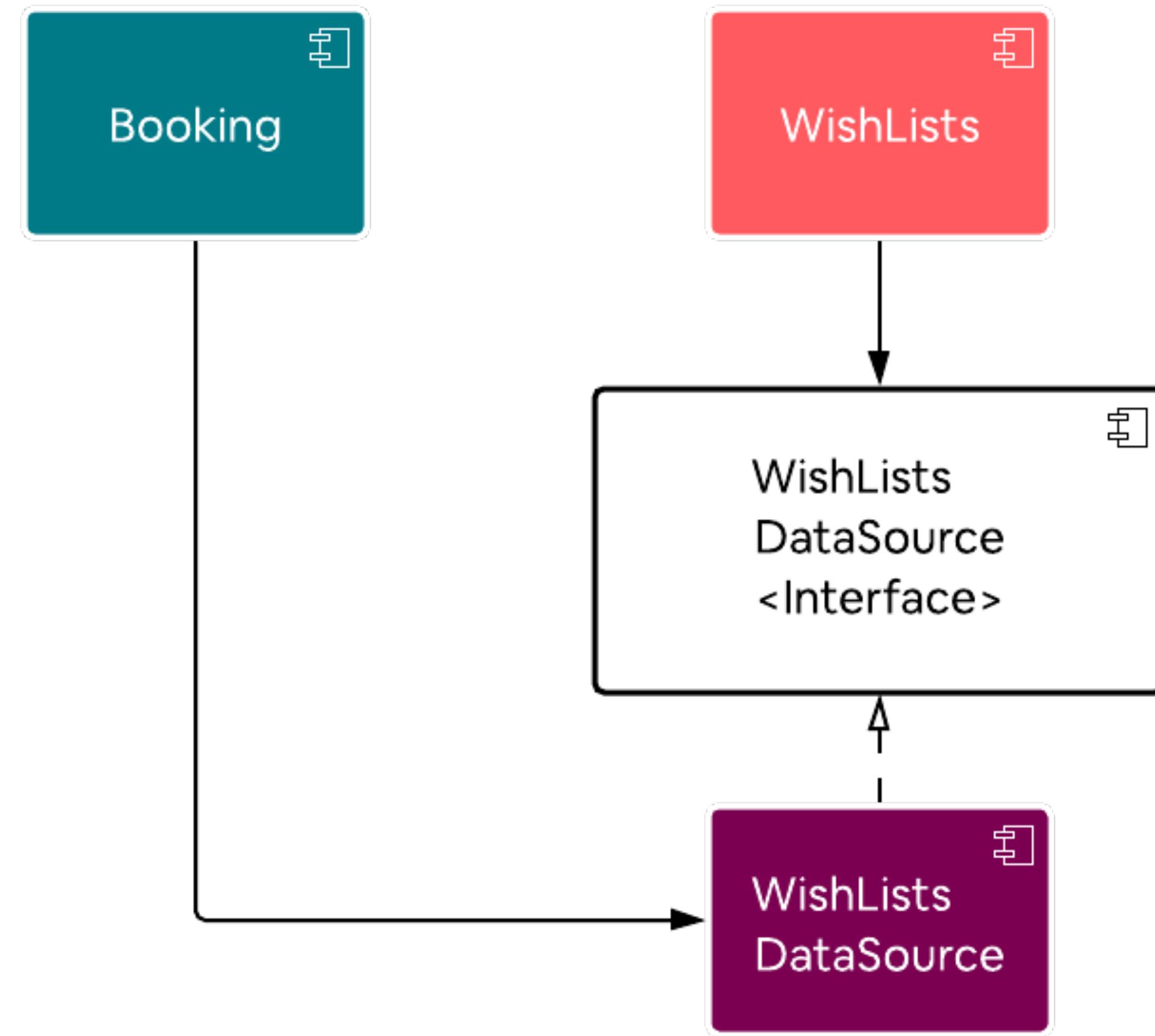


Easy!

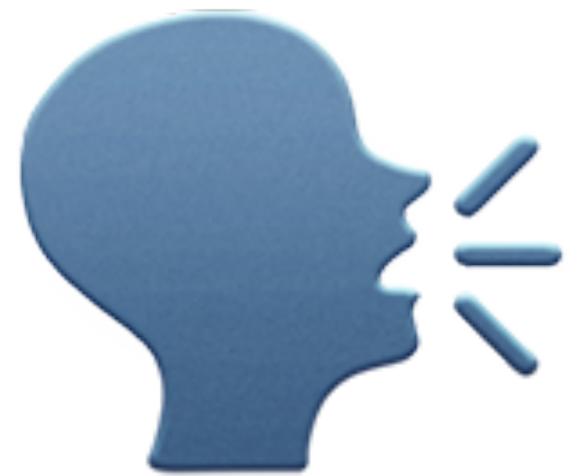
wishListDataSource
+ interface!



wishListDataSource
is still visible



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

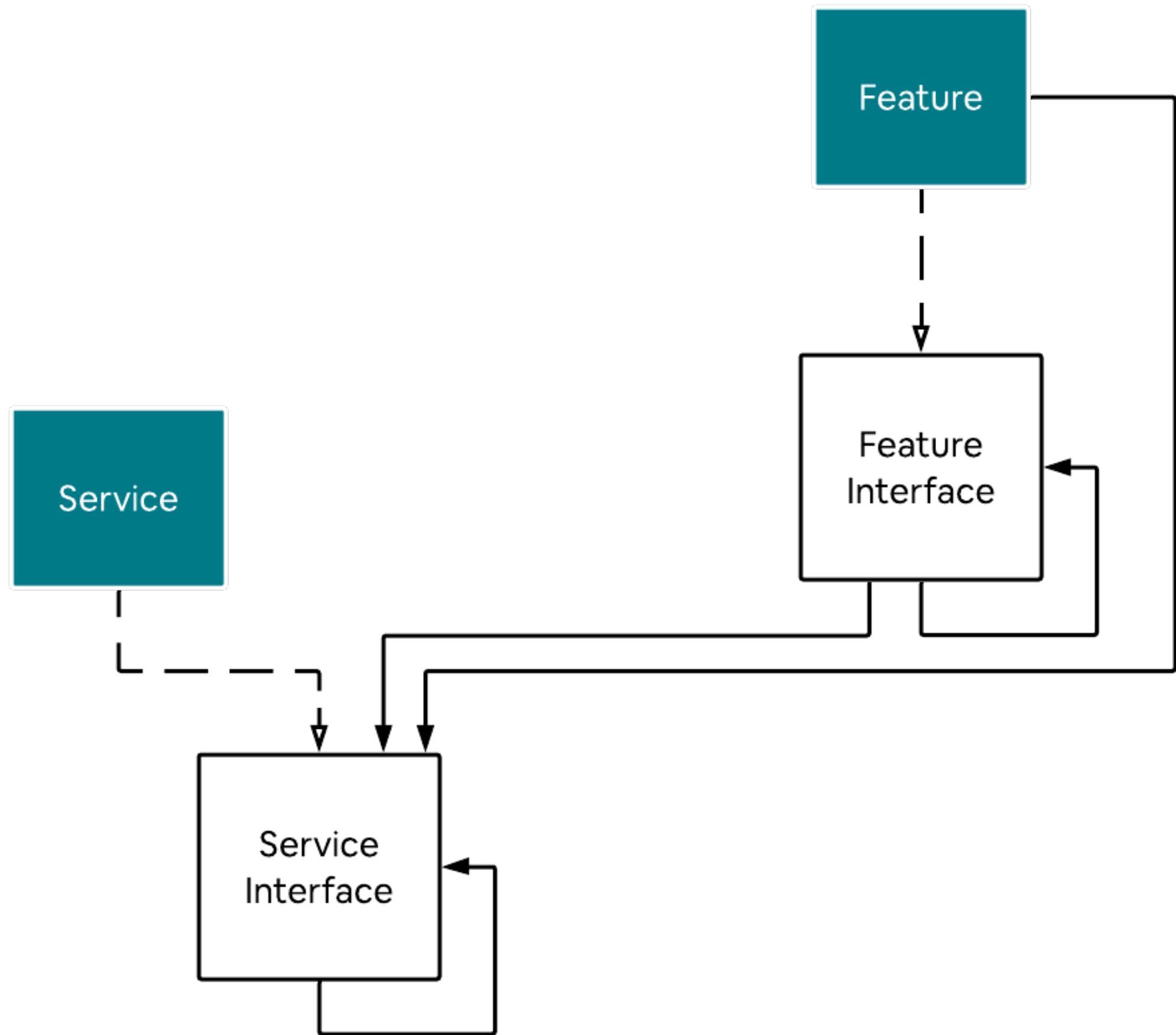
Feature Interfaces

Services

Networking

WishList Service

Service Interfaces



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****



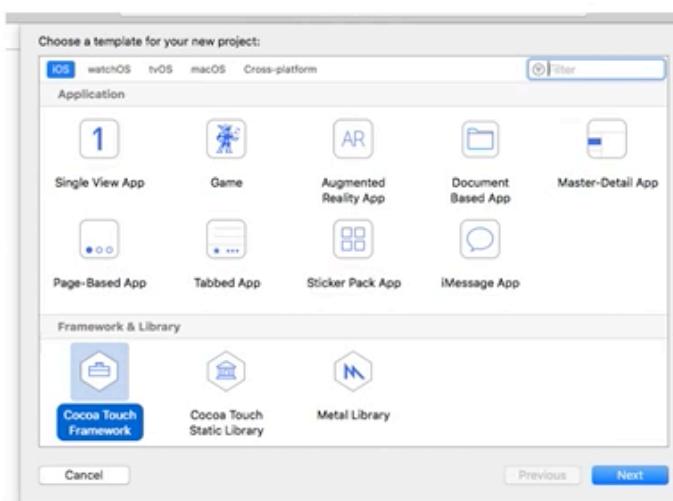
Can't find the perfect home for the new hoochiness you're cranking out? Your team's module getting crazy huge? If the answer to either of these or similar questions is yes, you're in the right place.

- 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/`

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 @@
1 +// !$*UTF8*$!
2 +{
3 +    archiveVersion = 1;
4 +    classes = {
5 +        };
6 +    objectVersion = 50;
7 +    objects = {
8 +
9 +/* Begin PBXBuildFile section */
10 +        436273881E6938649C14785D /* Pods_PersistenceTests.framework in Frameworks */ = {isa = PBXBuildFile; fileRef = D2B74E53691C4363AEE104A5 /* Pods_Persistence.framework in Frameworks */; fileRef = E811050620801D8A00B60444 /* Persistence.framework in Frameworks */; isa = PBXBuildFile; fileRef = E811050D20801D8A00B60444 /* Persistence.h in Headers */; isa = PBXBuildFile; fileRef = E81104FF20801D8A00B60444 /* Persistence.h in Headers */; };
11 +/* End PBXBuildFile section */
12 +
13 +/* Begin PBXContainerItemProxy section */
14 +        E811050720801D8A00B60444 /* PBXContainerItemProxy */ = {
15 +            isa = PBXContainerItemProxy;
16 +            containerPortal = E81104F320801D8A00B60444 /* Project object */;
17 +            proxyType = 1;
18 +            remoteGlobalIDString = E81104FB20801D8A00B60444;
19 +            remoteInfo = Persistence;
20 +
21 +        };
22 +/* End PBXContainerItemProxy section */
23 +
24 +/* Begin PBXFileReference section */
25 +        2A86D39AD2A2CDC7FF4EDDB3 /* Pods_PersistenceTests.release.xcconfig */ = {isa = PBXFileReference; includeInIndex = 1; };
26 +        4A8DB2ED4E26D034F585282A /* Pods_Persistence.debug.xcconfig */ = {isa = PBXFileReference; includeInIndex = 1; };
27 +        81651BAE840BAE55C43D49F2 /* Pods_PersistenceTests.framework */ = {isa = PBXFileReference; explicitFileType = wrapper.cframework; name = Pods_PersistenceTests.framework; };
28 +        9E14E1B3680C35C06F14CE0B /* Pods_Persistence.framework */ = {isa = PBXFileReference; explicitFileType = wrapper.cframework; name = Pods_Persistence.framework; };
29 +        A7842D401F139E2681BF18A2 /* Pods_PersistenceTests.debug.xcconfig */ = {isa = PBXFileReference; includeInIndex = 1; };
30 +        C4AC12656D775D1A7B9600A4 /* Pods_Persistence.release.xcconfig */ = {isa = PBXFileReference; includeInIndex = 1; };
31 +        E81104FC20801D8A00B60444 /* Persistence.framework */ = {isa = PBXFileReference; explicitFileType = wrapper.cframework; name = Persistence.framework; };
32 +        E81104FF20801D8A00B60444 /* Persistence.h */ = {isa = PBXFileReference; lastKnownFileType = sourcecode.c_header; name = Persistence.h; };
33 +        E811050020801D8A00B60444 /* Info.plist */ = {isa = PBXFileReference; lastKnownFileType = text.plist.xml; name = Info.plist; };
34 +        E811050520801D8A00B60444 /* PersistenceTests.xctest */ = {isa = PBXFileReference; explicitFileType = wrapper.carchive; name = PersistenceTests.xctest; };
35 +        E811050C20801D8A00B60444 /* Info.plist */ = {isa = PBXFileReference; lastKnownFileType = text.plist.xml; name = Info.plist; };
36 +/* End PBXFileReference section */
37 +
38 +/* Begin PBXFrameworksBuildPhase section */
39 +        E81104F820801D8A00B60444 /* Frameworks */ = {
40 +            isa = PBXFrameworksBuildPhase;
41 +            buildActionMask = 2147483647;
42 +            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:

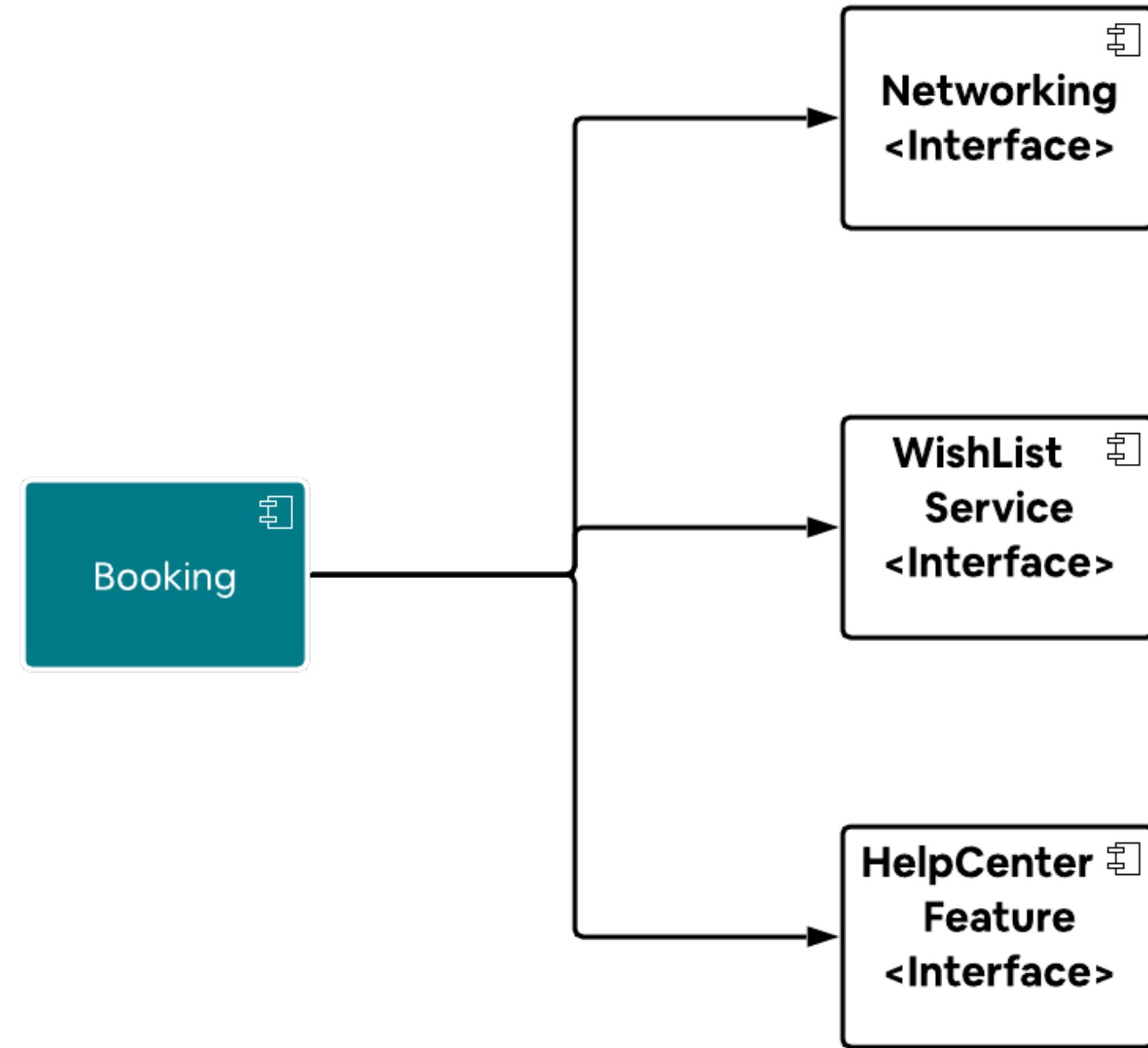
This is a module to present at Swiftable

BUCK

Human readable dependencies

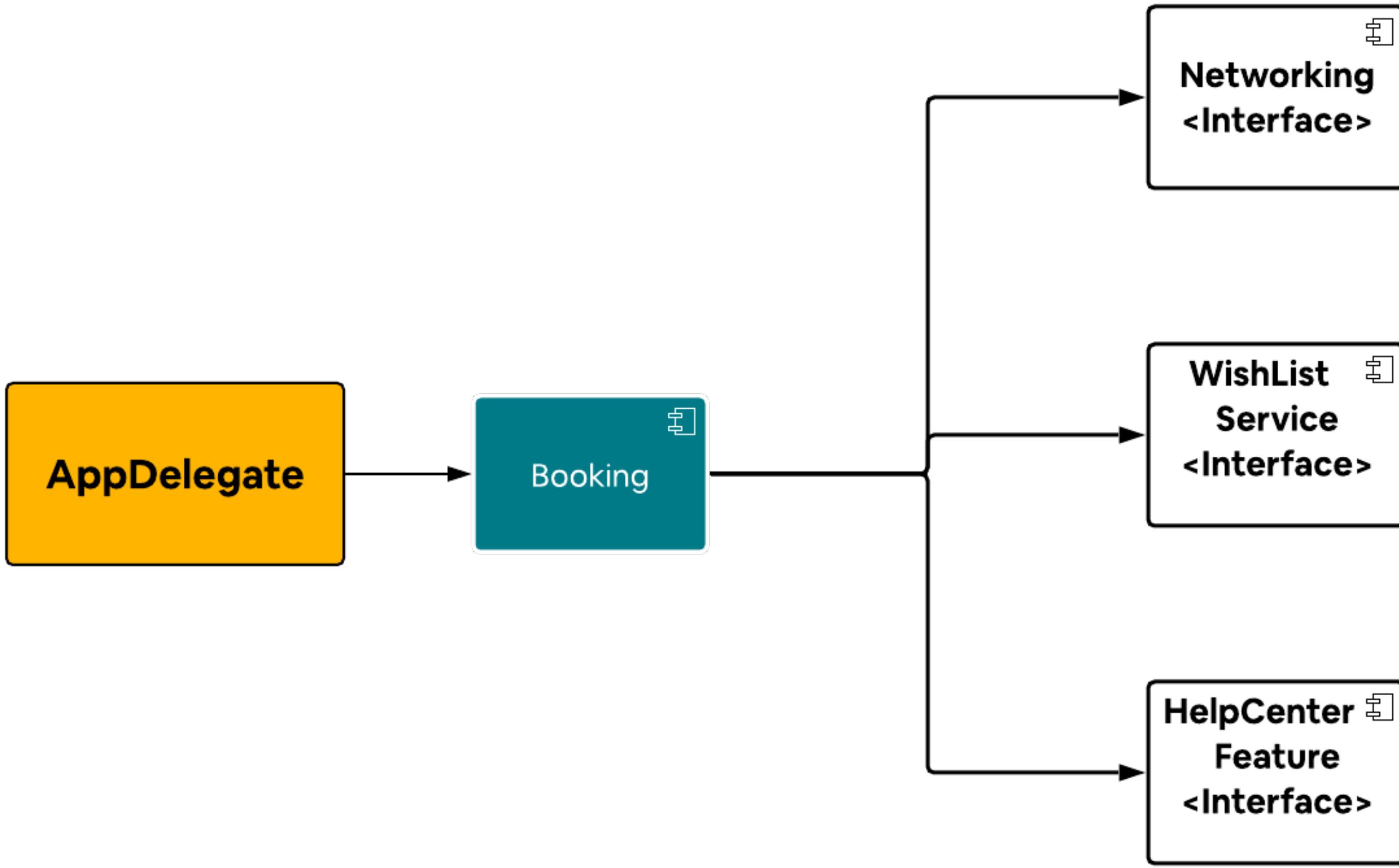
<https://github.com/airbnb/BuckSample>

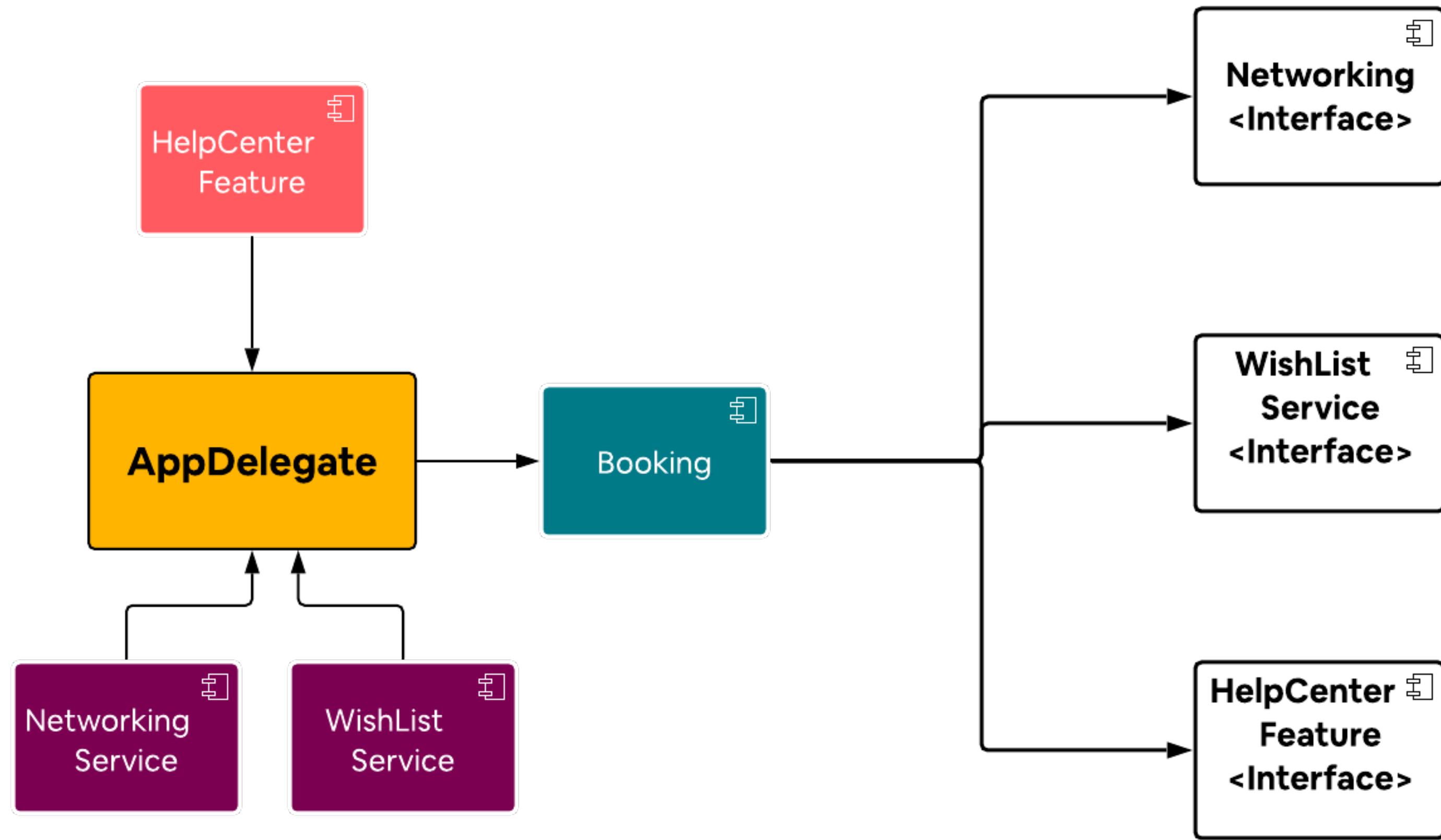
```
feature(  
    name = "Booking",  
    deps = [  
        "//ios/service_interfaces/Networking",  
        "//ios/service_interfaces/WishListService",  
        "//ios/feature_interfaces/HelpCenter",  
    ],  
)
```

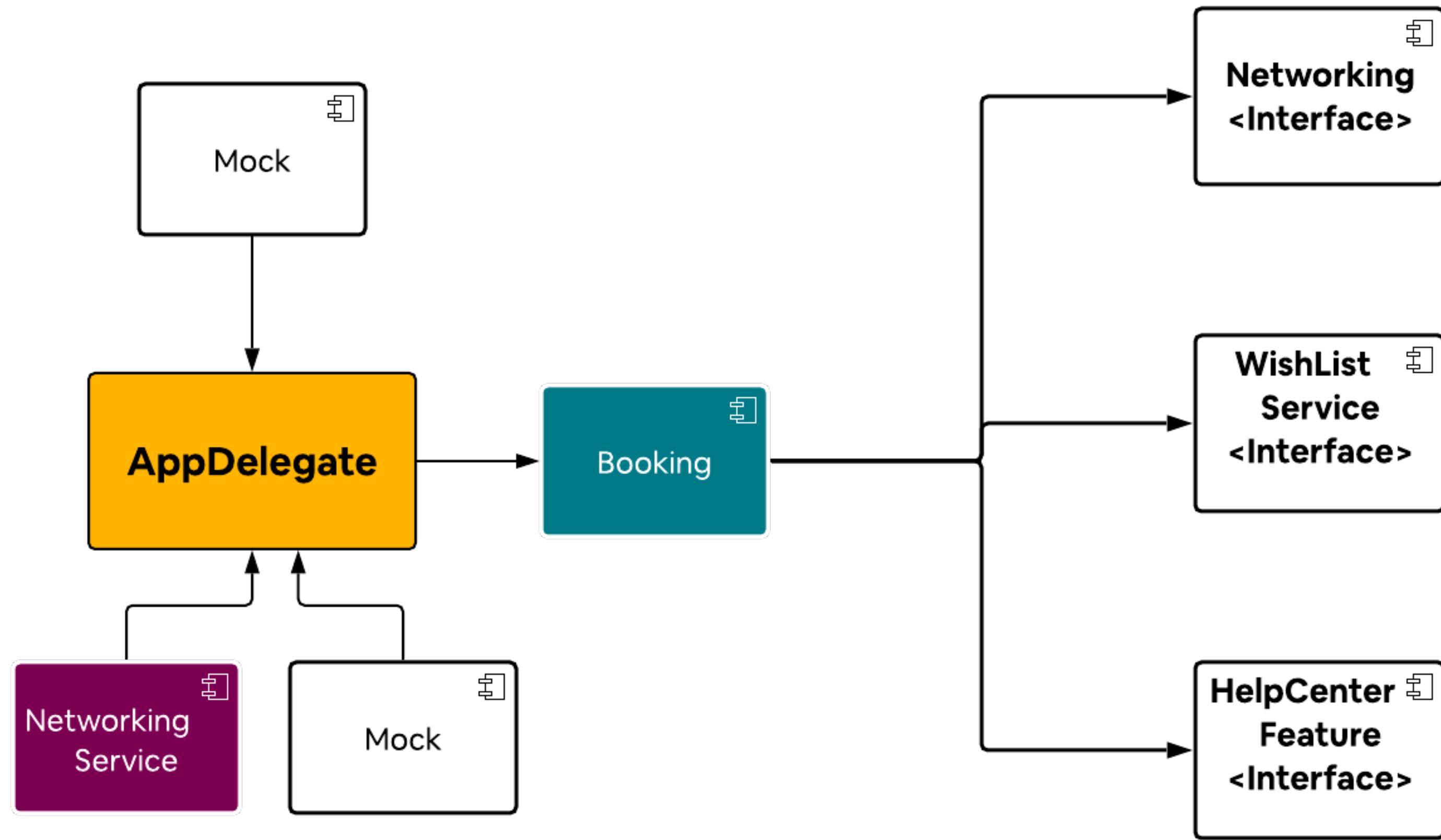


Feature

A screen or a flow
in the app







dev Apps

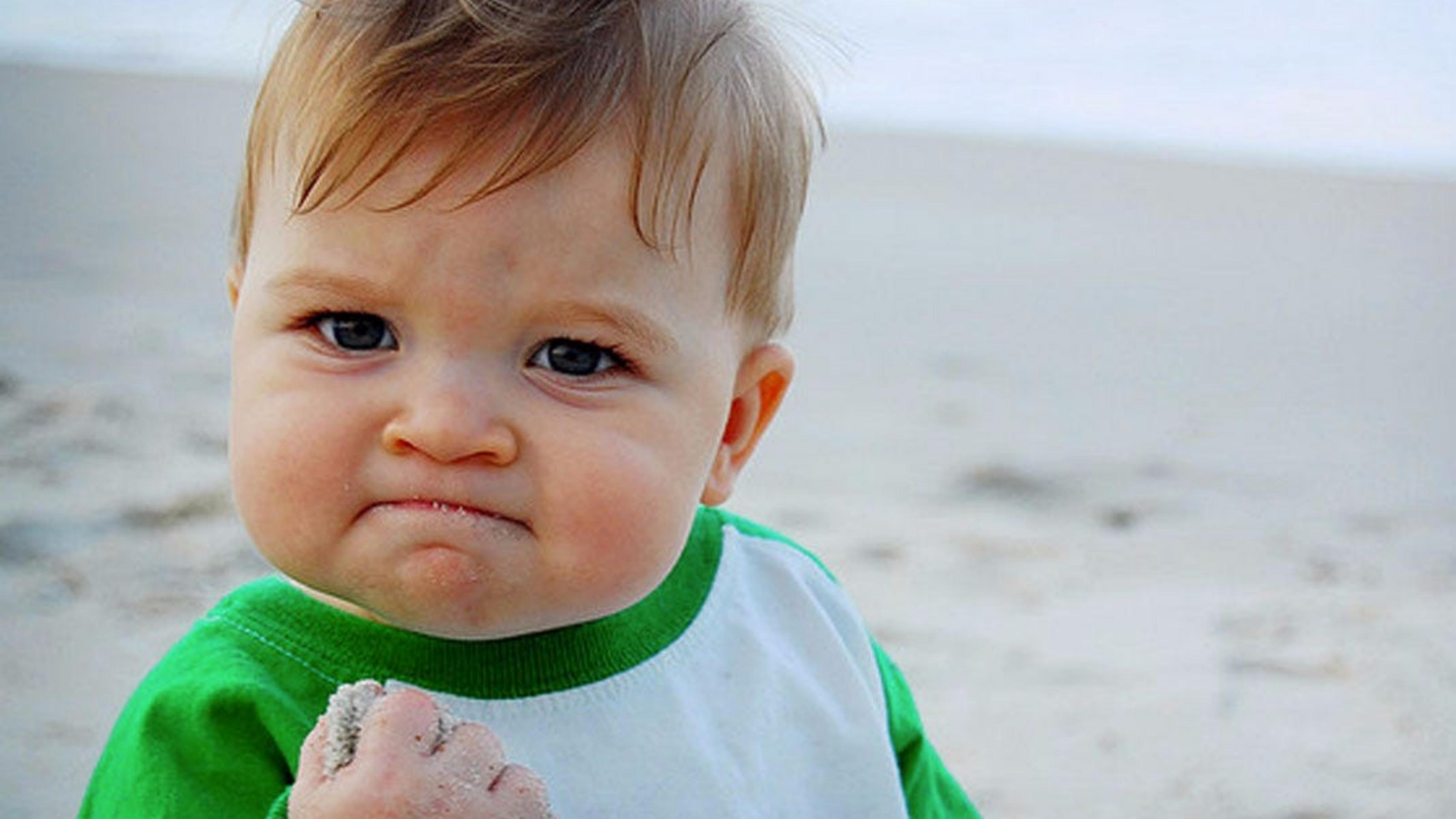
1 main

Dev Apps

Big buckets

sma

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**

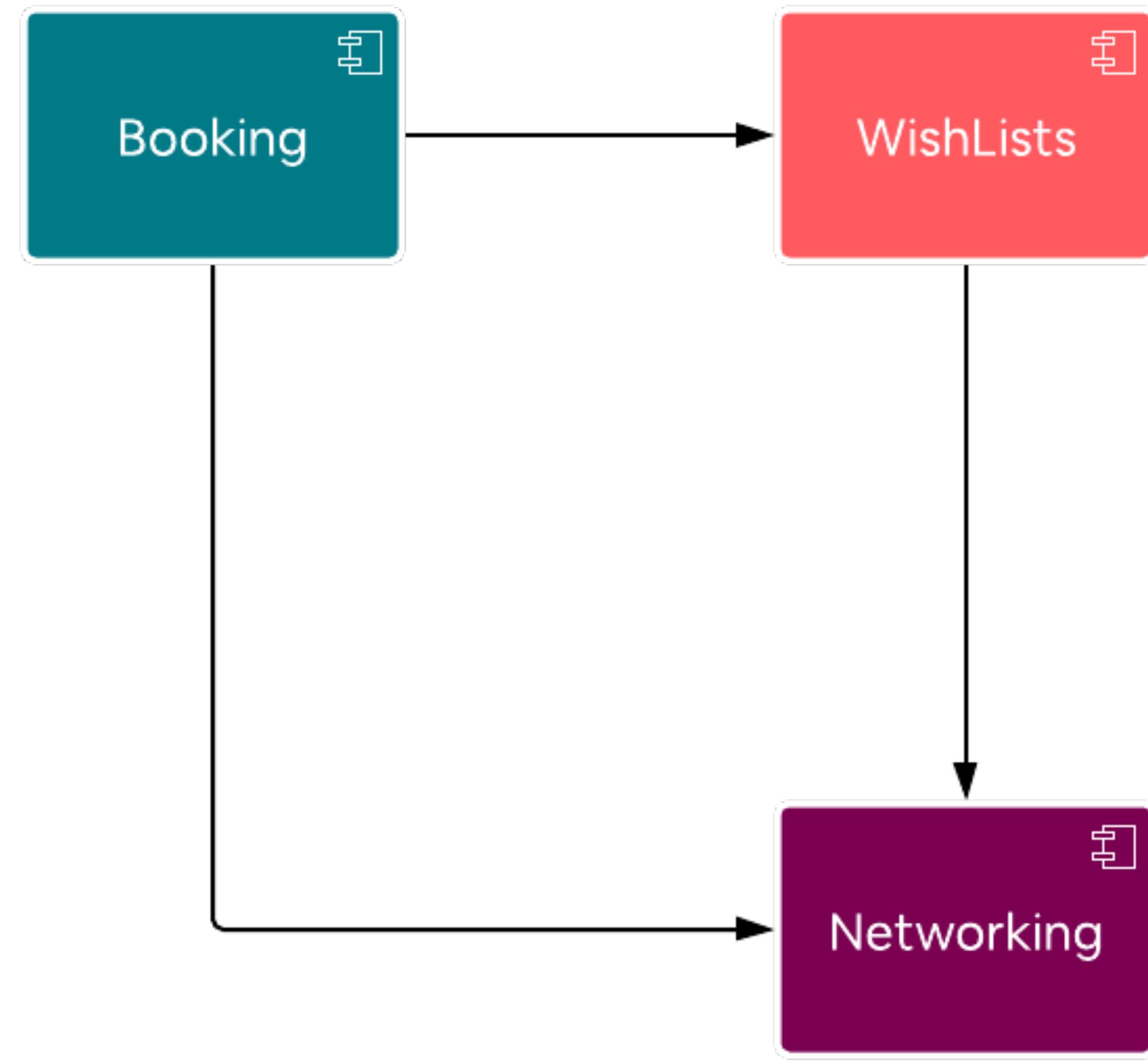


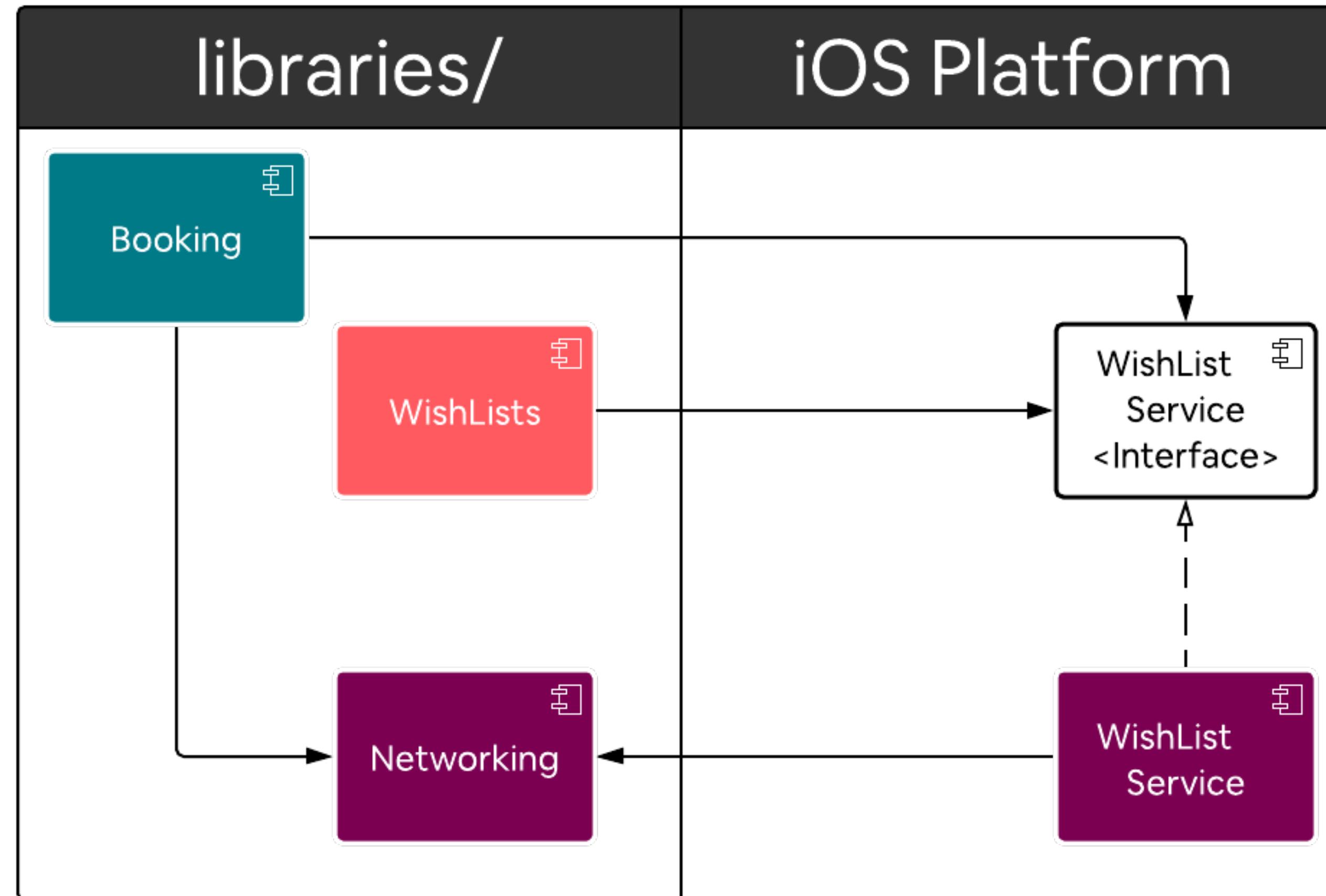
Progressively
migrate

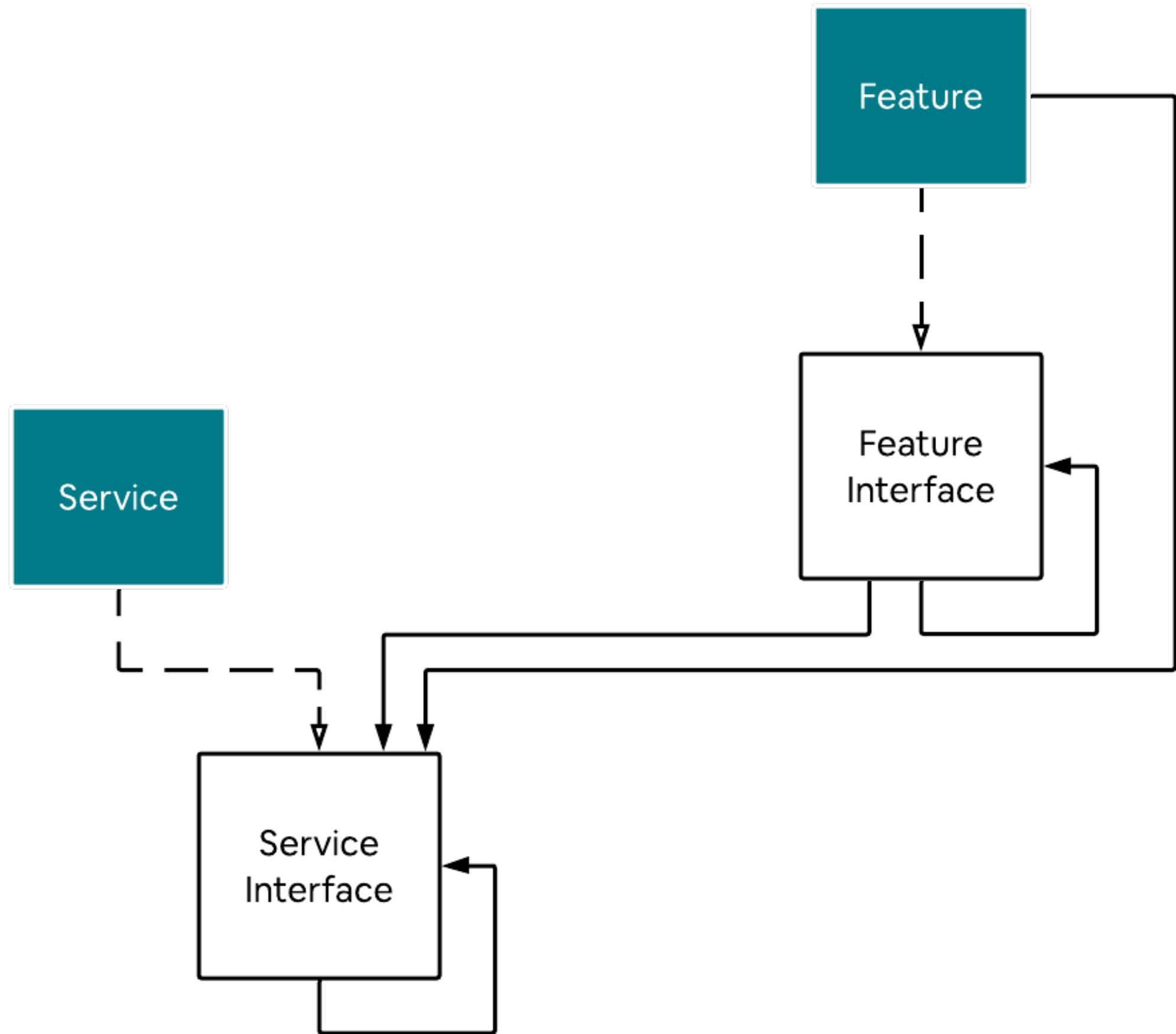
Let's
migrate

WishLists Data Source

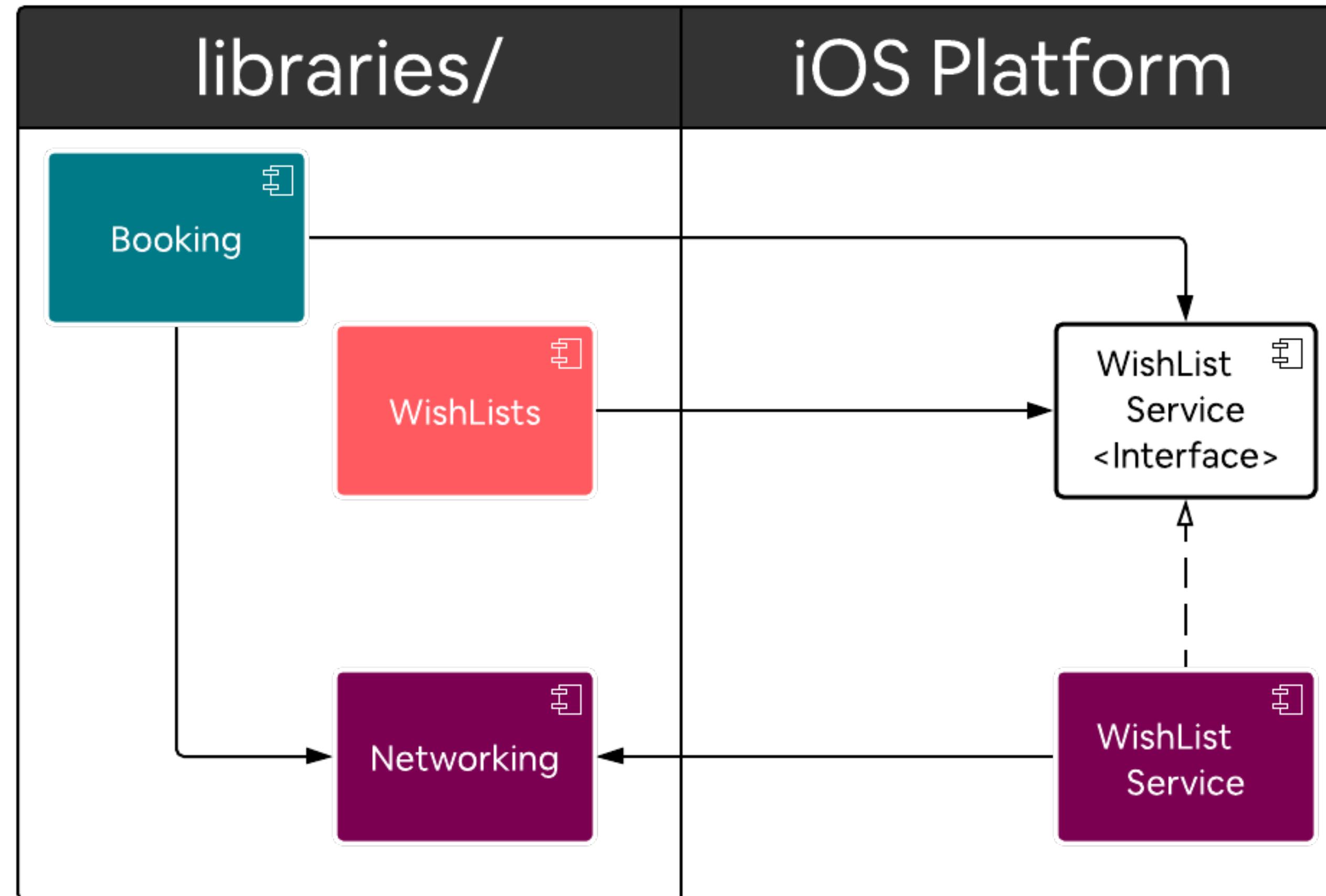
libraries/wishlists







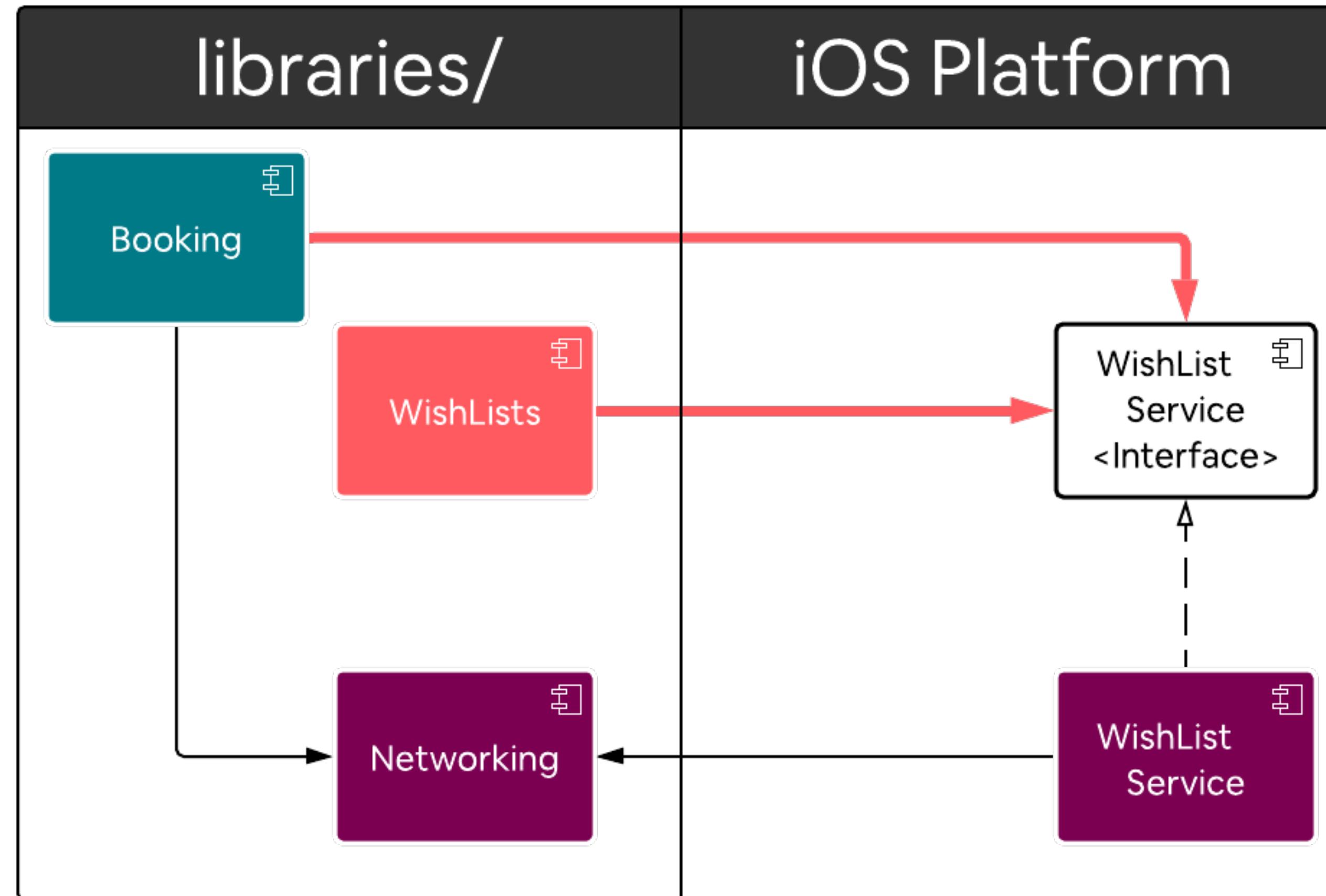
**What are the
dependency rules
for libraries/?**



Inbound dependencies
Outbound dependencies

Inbound dependencies

Outbound dependencies



**The interface module has
stricter rules**

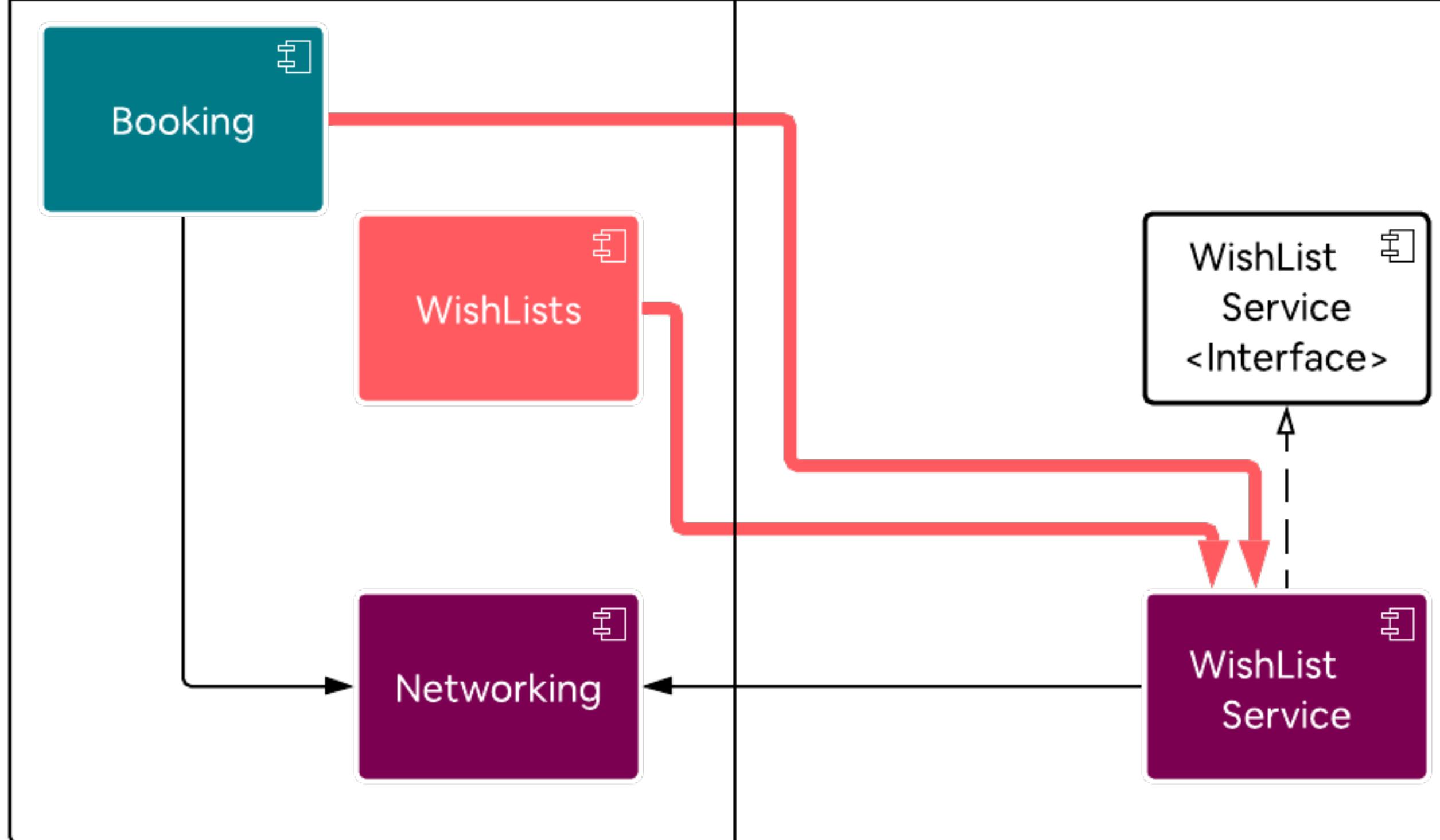
Migrate all the call
sites

As the owner of WishLists

We don't
control
these

libraries/

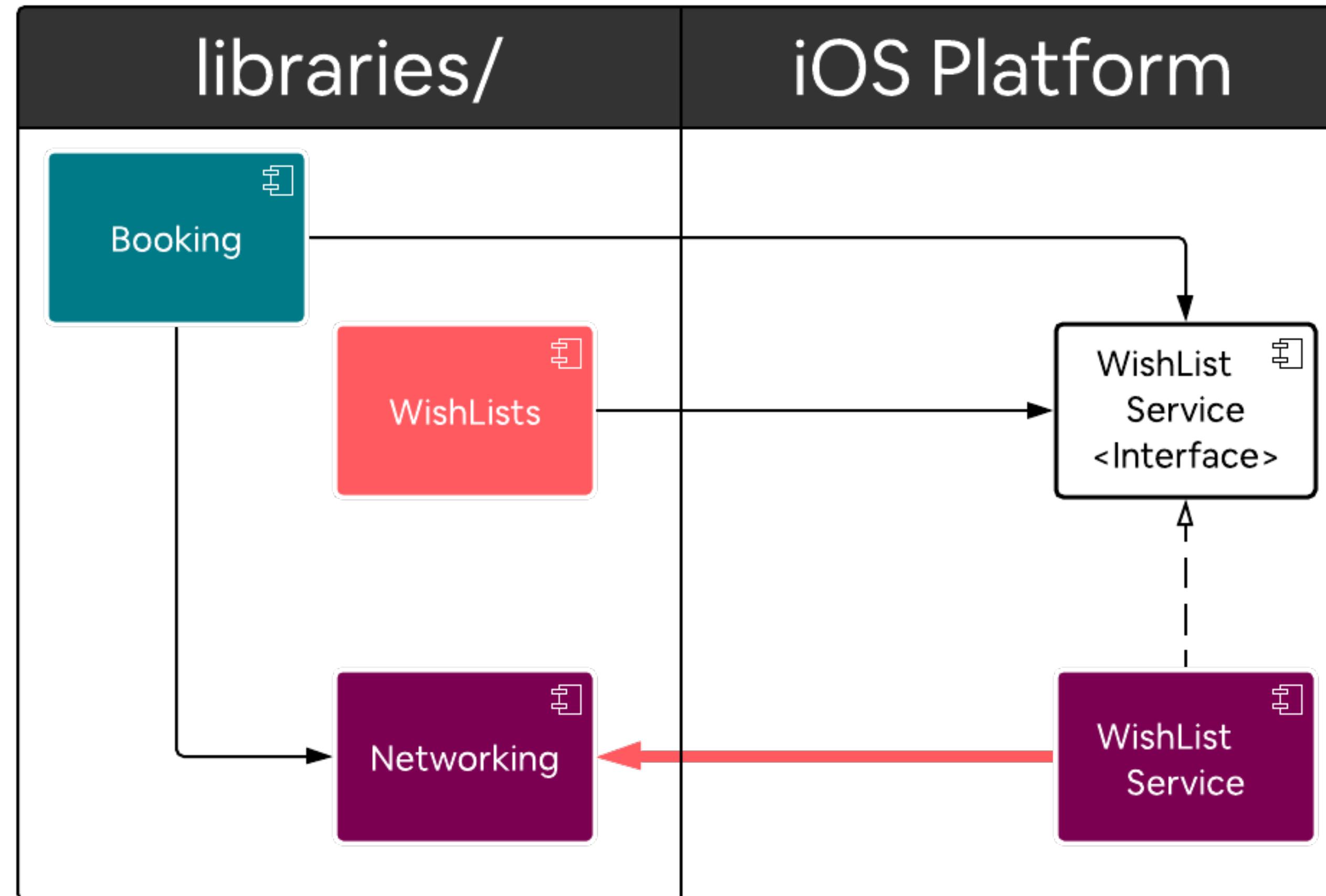
iOS Platform



**Calculated
tech debt**

Inbound dependencies

Outbound dependencies



**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

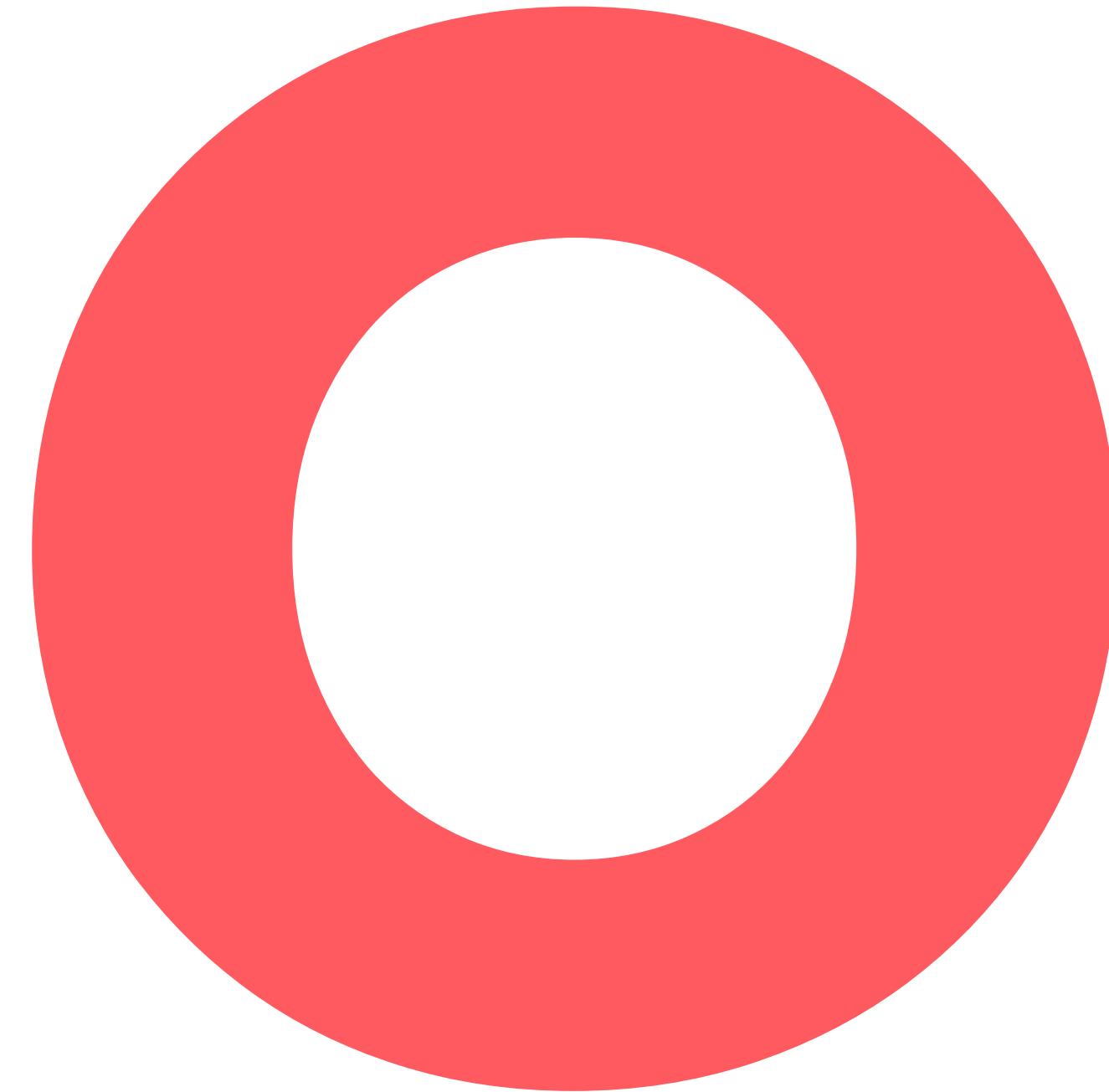
first

piilot

with others teams

Should I
implement
this?

Most likely



There's no
silver bullet

Adapt



Summary

1. Figure out where you're **struggling**
2. **Create and document** best practices
3. **Automate** best practices where needed

iGracias!

franciscodiaz.cl/talks