

The background of the slide is a wide-angle aerial photograph of a city during a vibrant sunset. The sky is filled with warm orange, yellow, and pink hues, transitioning into darker blues and purples at the top. Below, the city's grid-like street pattern is visible, with numerous buildings of varying heights and colors. In the foreground, there's a complex network of elevated highways and rail tracks. A large, semi-transparent dark blue rectangular box covers the lower third of the image, containing the title and subtitle text.

# Modular Project Architecture

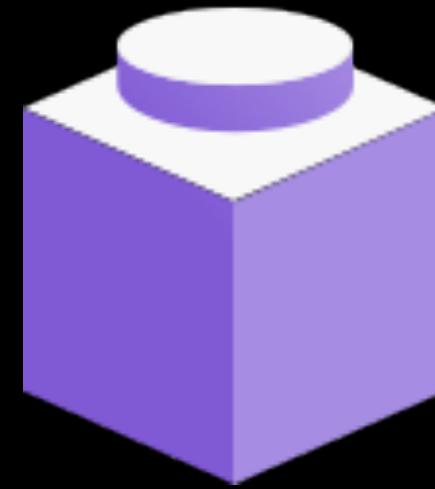
Deep Dish Swift 2023

Ben Scheirman





**NSScreencast (Video Tutorials)**  
<https://nsscreencast.com/>



**Combine Swift Course**  
<https://combineswift.com/>



**Side Mirror (macOS App)**  
<https://sidemirrorapp.com>



**Tonal Therapy (Tinnitus app)**  
<https://tonaltherapy.app/>



Easy Skeezy Date Formatting for Swift and Objective-C

DATE INPUT:

05/01/2023, 03:44 PM

(enter in ISO8601 format)

FORMAT TEXT

EEEE, MMM d, yyyy

LOCALE

en\_US\_POSIX



Result

Monday, May 1, 2023

Examples

Reference

Best Practices

About this site

Monday, May 1, 2023

EEEE, MMM d, yyyy

05/01/2023

MM/dd/yyyy

05-01-2023 15:44

MM-dd-yyyy HH:mm

May 1, 3:44 PM

MMM d, h:mm a

# Our Agenda

- Define what modularization is and why you should care
- Discuss the importance of understanding dependencies
- Show some tools and approaches to make this all work





# Now we have a Large Monolithic Project

- with longer build times



# SLOW BUILDS

# Now we have a Large Monolithic Project

- Cause longer build times
- We see git contention on hard-to-merge files



Hierarchical Flat < > project.pbxproj No Selection

```
Podfile Podfile.lock
Pods Manifest.lock
Pods.xcodeproj project.pbxproj
Target Support Files
Pods-TrakiTSwift
  Pods-TrakiTSwift-acknowledgements.markdown
  Pods-TrakiTSwift-acknowledgements.plist
  Pods-TrakiTSwift-frameworks.sh
  Pods-TrakiTSwift.debug.xcconfig
  Pods-TrakiTSwift.release.xcconfig
TrakiTSwift
  AppDelegate.swift
TrakiTSwift.xcodeproj project.pbxproj
```

c 186 21F8F2F1B5AB5302453100EE25F58EFD /\* ObserveOn.swift in Sources \*/ = {isa = PBXBuildFile; fileRef = B503D0C9B356C8866113F8747FE04CC2 /\* ObserveOn.swift \*/; };
c 187 2207C08298FD767301F65DD36F05B043 /\* Observable.swift in Sources \*/ = {isa = PBXBuildFile; fileRef = CEDA044890165F14ABBE7752A4D4E7BD /\* Observable.swift \*/; };
c 188 221DAE8B7764BA6A545214DE78A5E611 /\* RLMObjectBase.h in Headers \*/ = {isa = PBXBuildFile; fileRef = 44CA32F875BBB908A4C167CA82E4E3D1 /\* RLMObjectBase.h \*/; };
M 189 225262C43C46BCA8AF10A76B8988B732 /\* UIScrollView+Rx.swift in Sources \*/ = {isa = PBXBuildFile; fileRef = 5FA67D735FD8387BC547A370640D8D04 /\* UIScrollView+Rx.swift \*/; };
c 190 2283375C0C99789CF47EEDA0CE9D3D80 /\* ImageProcessor.swift in Sources \*/ = {isa = PBXBuildFile; fileRef = 0E639FA734425F51782ED1B11FE450F0 /\* ImageProcessor.swift \*/; };
c 191 228E8A3B9DD6DD73FAB125C50730C0DD /\* EasyPeasy-dummy.m in Sources \*/ = {isa = PBXBuildFile; fileRef = BEDA843ED4327ED572B36C6965133D10 /\* EasyPeasy-dummy.m \*/; };
c 192 2355B3C26660A94D184C2A4293FCA129 /\* Take.swift in Sources \*/ = {isa = PBXBuildFile; fileRef = FDCC208CFC71EE46B660A2E16F0603D5 /\* Take.swift \*/; };
c 193 2382F0F1376608160A043D9B3EB3D553 /\* UITabBarItem+Rx.swift in Sources \*/ = {isa = PBXBuildFile; fileRef = DB37F02DA0B658981FDFBEAF99FE7028 /\* UITabBarItem+Rx.swift \*/; };
c 194 23FB1BB3483395AA68A02768CDFD8D6A /\* BugfenderSDK-dummy.m in Sources \*/ = {isa = PBXBuildFile; fileRef = 2056E4753CB4431698818417E68F1715 /\* BugfenderSDK-dummy.m \*/; };
c 195 249EEF10BFAF8B4A52A0F8426867901A /\* RLMRealmConfiguration\_Private.h in Copy . Private Headers \*/ = {isa = PBXBuildFile; fileRef = D895A971E71903195D677961C79E07E8 /\* RLMRealmConfiguration\_Private.h \*/; };
c 196 24F0B89F85BA2BEF270AF5AEE803C057 /\* RLMObjectBase\_Private.h in Headers \*/ = {isa = PBXBuildFile; fileRef = D3C3A1F1018957EC02F7BD66DA78FF1E /\* RLMObjectBase\_Private.h \*/; };
c 197 2668A39759D5E3959F835D87D1D2E536 /\* Localize-Swift-dummy.m in Sources \*/ = {isa = PBXBuildFile; fileRef =

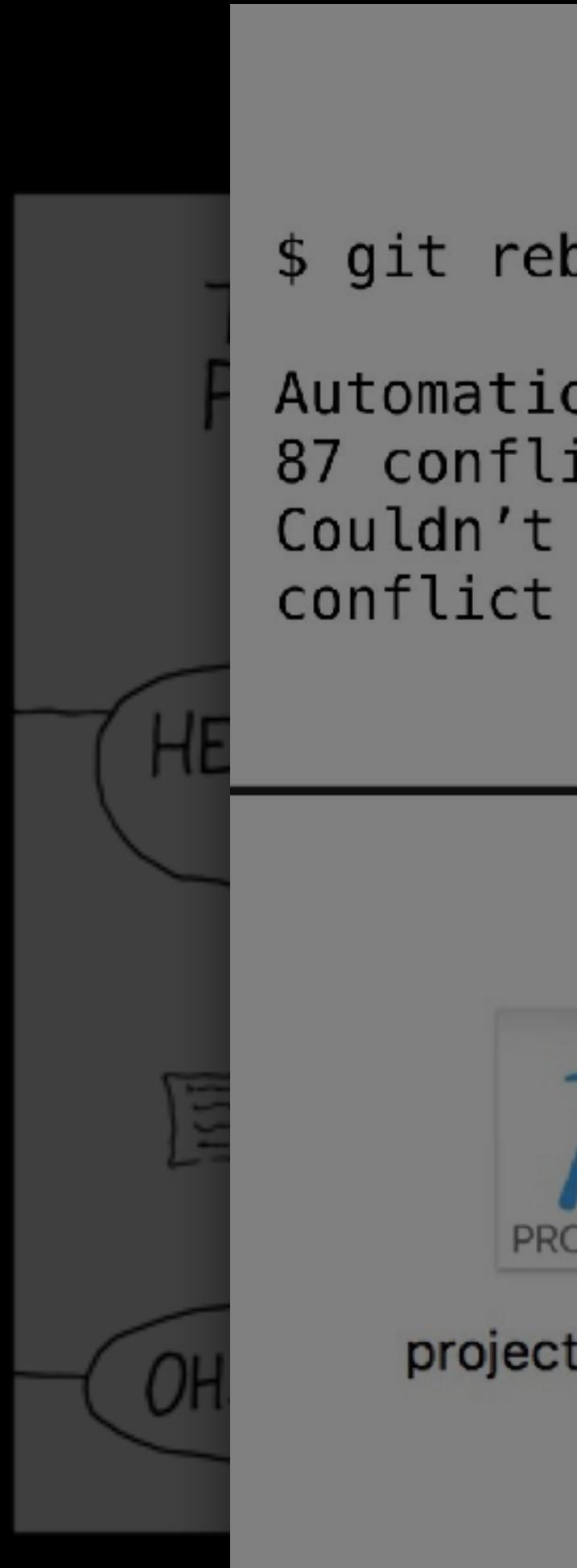
c 188 21F8F2F1B5AB5302453100EE25F58EFD /\* ObserveOn.swift in Sources \*/ = {isa = PBXBuildFile; fileRef = B503D0C9B356C8866113F8747FE04CC2 /\* ObserveOn.swift \*/; };
c 189 2207C08298FD767301F65DD36F05B043 /\* Observable.swift in Sources \*/ = {isa = PBXBuildFile; fileRef = CEDA044890165F14ABBE7752A4D4E7BD /\* Observable.swift \*/; };
c 190 221DAE8B7764BA6A545214DE78A5E611 /\* RLMObjectBase.h in Headers \*/ = {isa = PBXBuildFile; fileRef = 44CA32F875BBB908A4C167CA82E4E3D1 /\* RLMObjectBase.h \*/; };
c 191 225262C43C46BCA8AF10A76B8988B732 /\* UIScrollView+Rx.swift in Sources \*/ = {isa = PBXBuildFile; fileRef = 5FA67D735FD8387BC547A370640D8D04 /\* UIScrollView+Rx.swift \*/; };
c 192 2283375C0C99789CF47EEDA0CE9D3D80 /\* ImageProcessor.swift in Sources \*/ = {isa = PBXBuildFile; fileRef = 0E639FA734425F51782ED1B11FE450F0 /\* ImageProcessor.swift \*/; };
c 193 228E8A3B9DD6DD73FAB125C50730C0DD /\* EasyPeasy-dummy.m in Sources \*/ = {isa = PBXBuildFile; fileRef = BEDA843ED4327ED572B36C6965133D10 /\* EasyPeasy-dummy.m \*/; };
c 194 2355B3C26660A94D184C2A4293FCA129 /\* Take.swift in Sources \*/ = {isa = PBXBuildFile; fileRef = FDCC208CFC71EE46B660A2E16F0603D5 /\* Take.swift \*/; };
c 195 2382F0F1376608160A043D9B3EB3D553 /\* UITabBarItem+Rx.swift in Sources \*/ = {isa = PBXBuildFile; fileRef = DB37F02DA0B658981FDFBEAF99FE7028 /\* UITabBarItem+Rx.swift \*/; };
c 196 23FB1BB3483395AA68A02768CDFD8D6A /\* BugfenderSDK-dummy.m in Sources \*/ = {isa = PBXBuildFile; fileRef = 2056E4753CB4431698818417E68F1715 /\* BugfenderSDK-dummy.m \*/; };
c 197 249EEF10BFAF8B4A52A0F8426867901A /\* RLMRealmConfiguration\_Private.h in Copy . Private Headers \*/ = {isa = PBXBuildFile; fileRef = D895A971E71903195D677961C79E07E8 /\* RLMRealmConfiguration\_Private.h \*/; };
c 198 24F0B89F85BA2BEF270AF5AEE803C057 /\* RLMObjectBase\_Private.h in Headers \*/ = {isa = PBXBuildFile; fileRef = D3C3A1F1018957EC02F7BD66DA78FF1E /\* RLMObjectBase\_Private.h \*/; };
c 199 2668A39759D5E3959F835D87D1D2E536 /\* Localize-Swift-dummy.m in Sources \*/ = {isa = PBXBuildFile; fileRef =



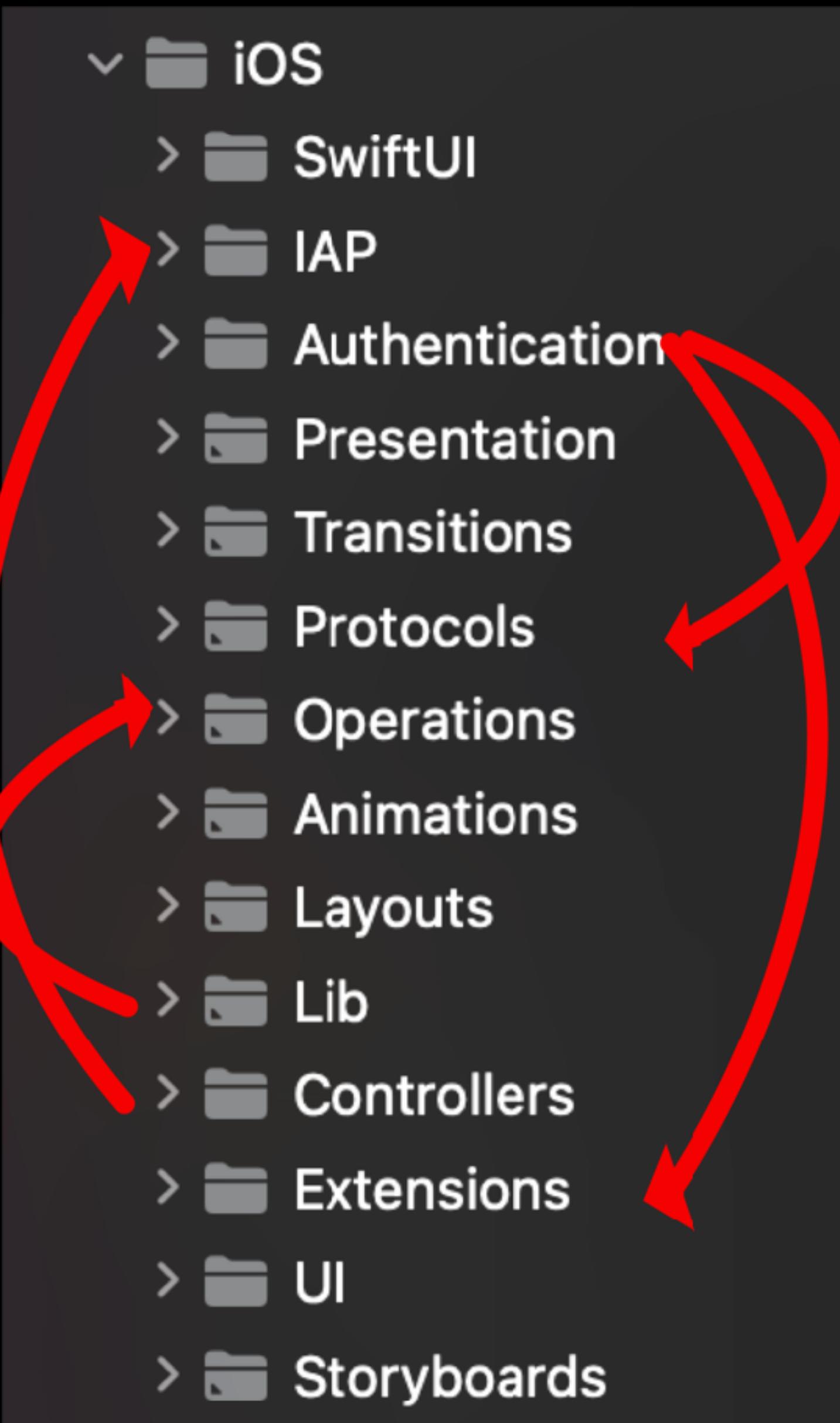
A medium shot of a man with dark hair and a beard, wearing a black t-shirt and white shorts, sitting on a wooden bench. He is looking down at a smartphone held in his hands. The background shows a brick wall and some foliage.

# Now we have a Large Monolithic Application

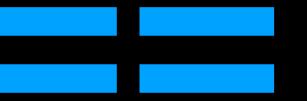
- Cause longer build times
- Produce git contention on hard-to-merge files
- We can't see our dependency problems



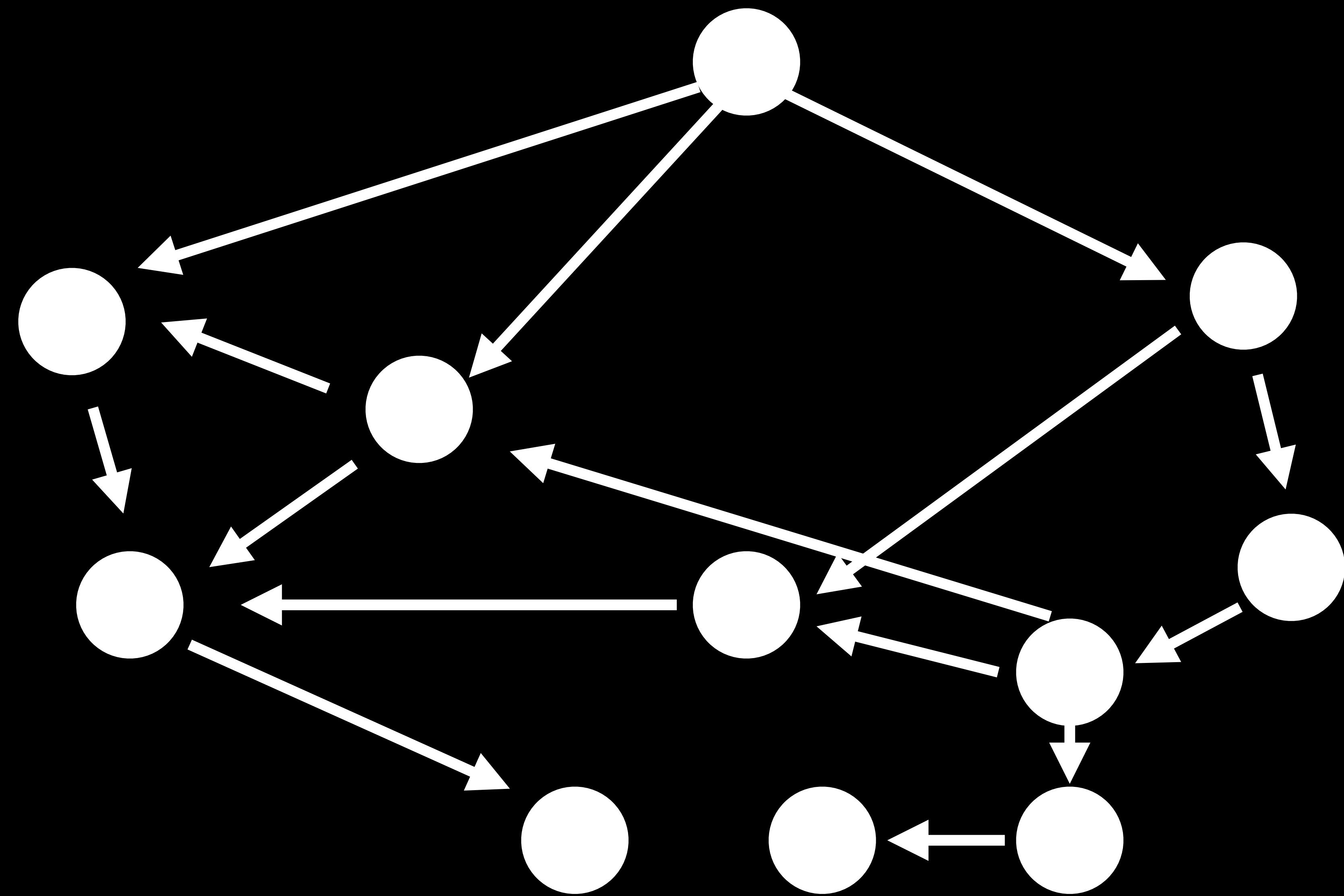
- ✓  iOS
- >  SwiftUI
- >  IAP
- >  Authentication
- >  Presentation
- >  Transitions
- >  Protocols
- >  Operations
- >  Animations
- >  Layouts
- >  Lib
- >  Controllers
- >  Extensions
- >  UI
- >  Storyboards

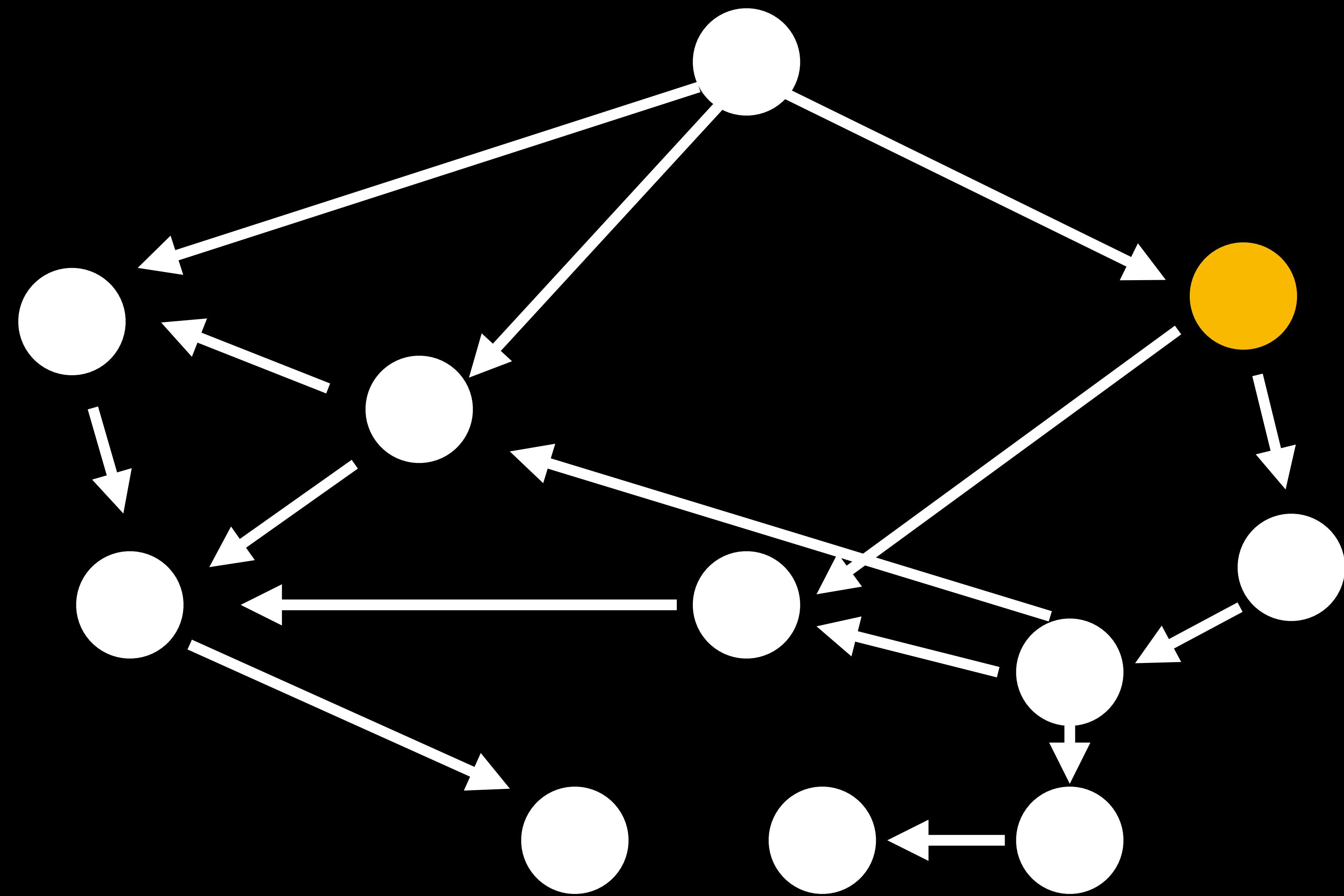


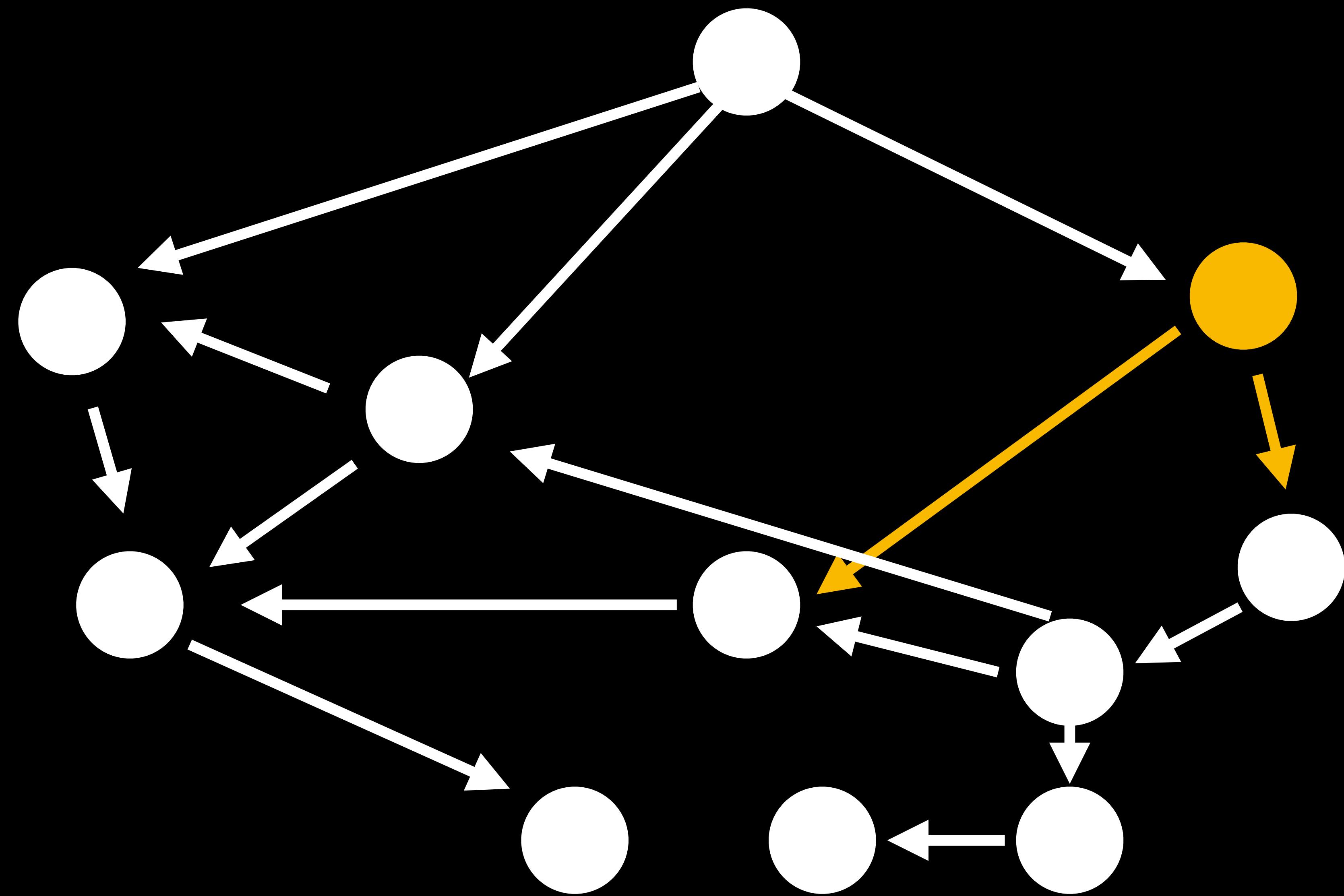
Tangled  
Dependencies

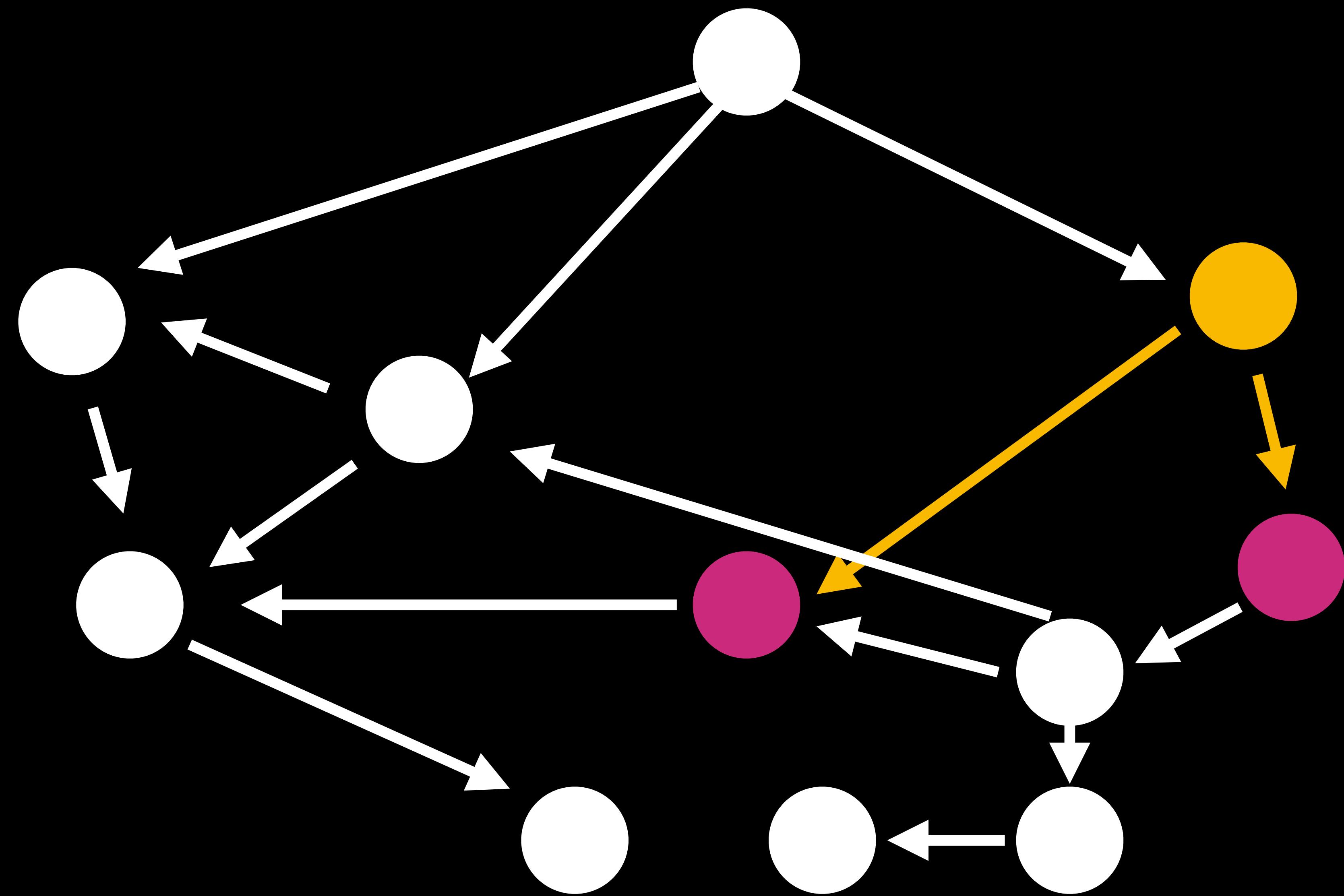


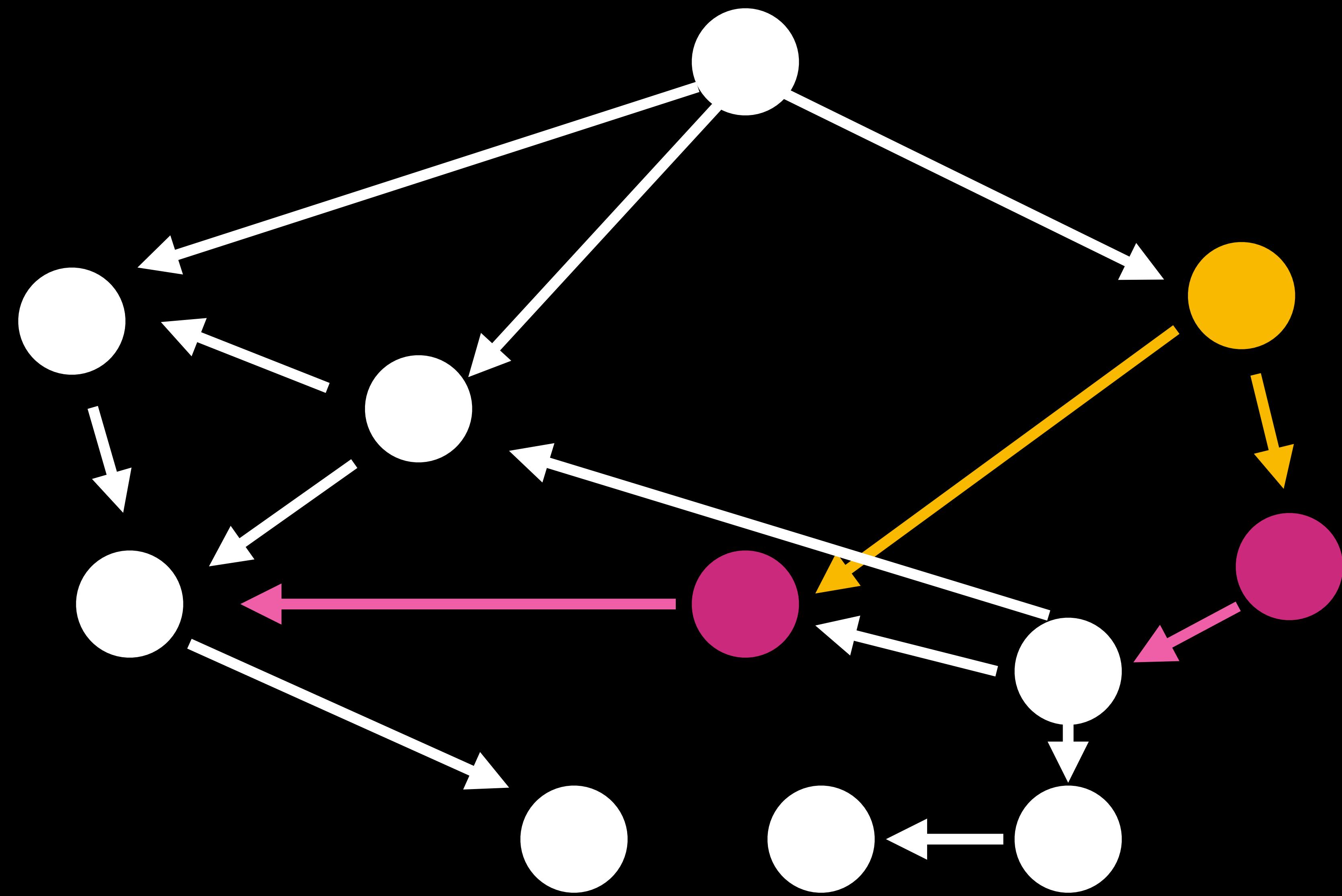
Resistance  
to changes

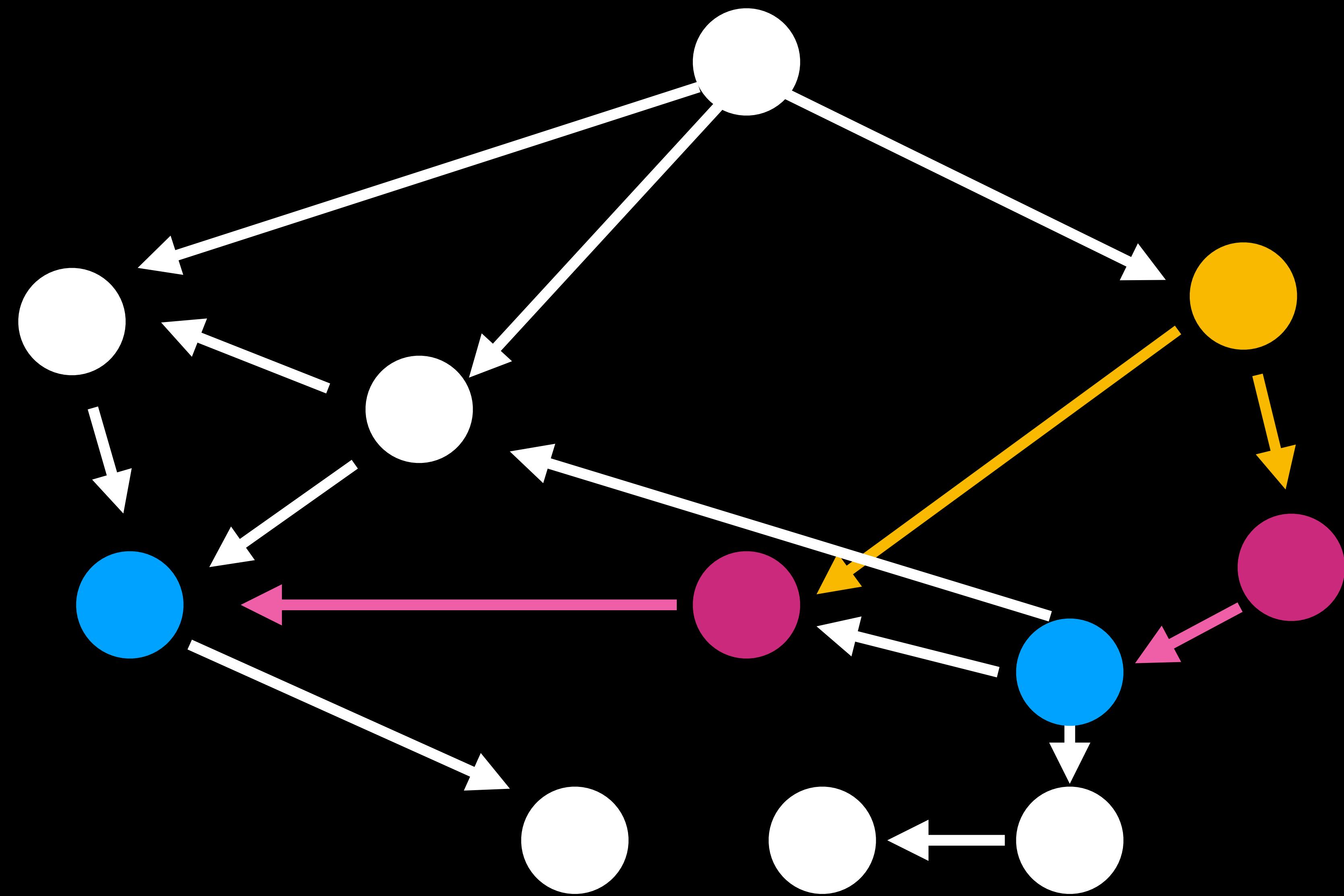


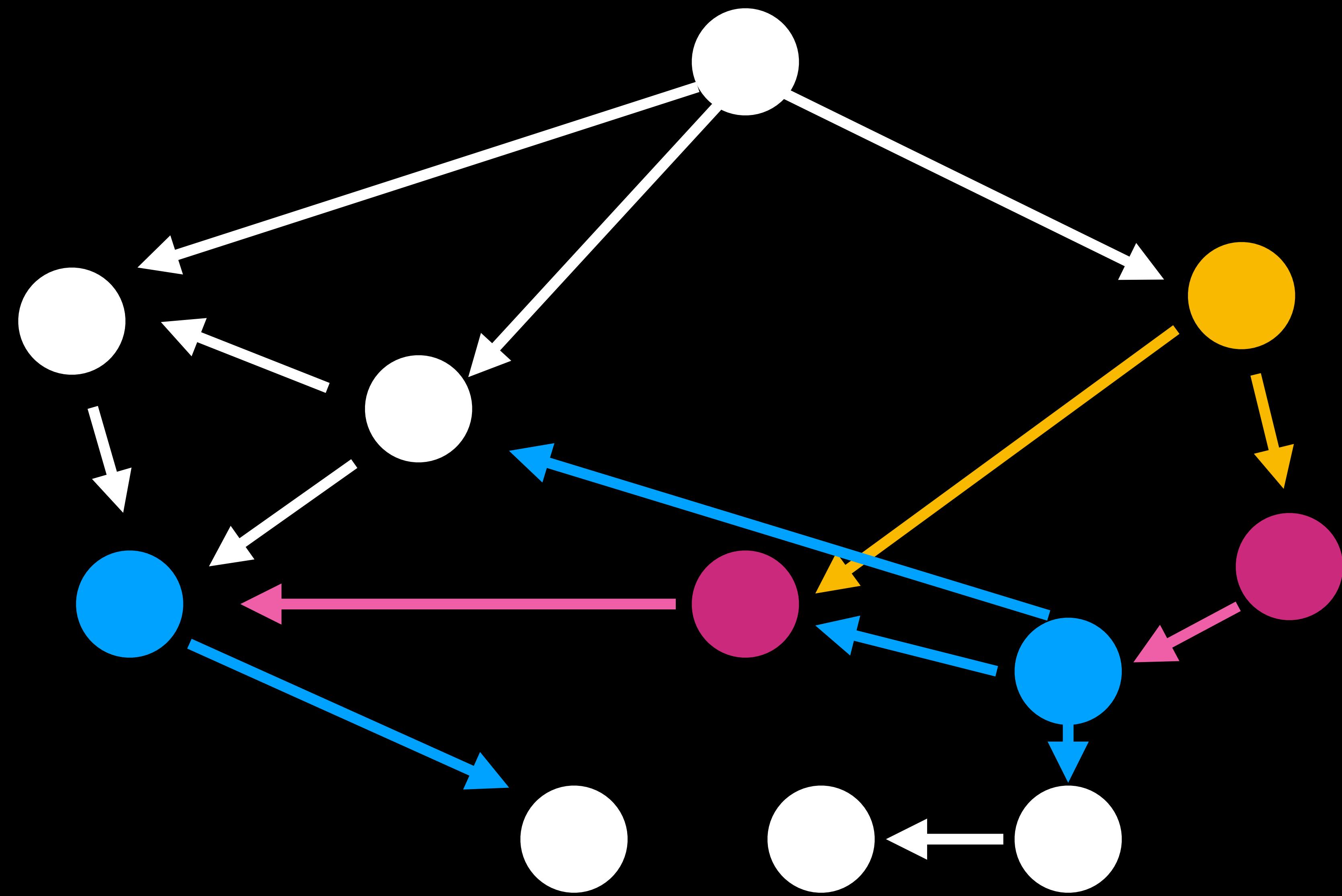


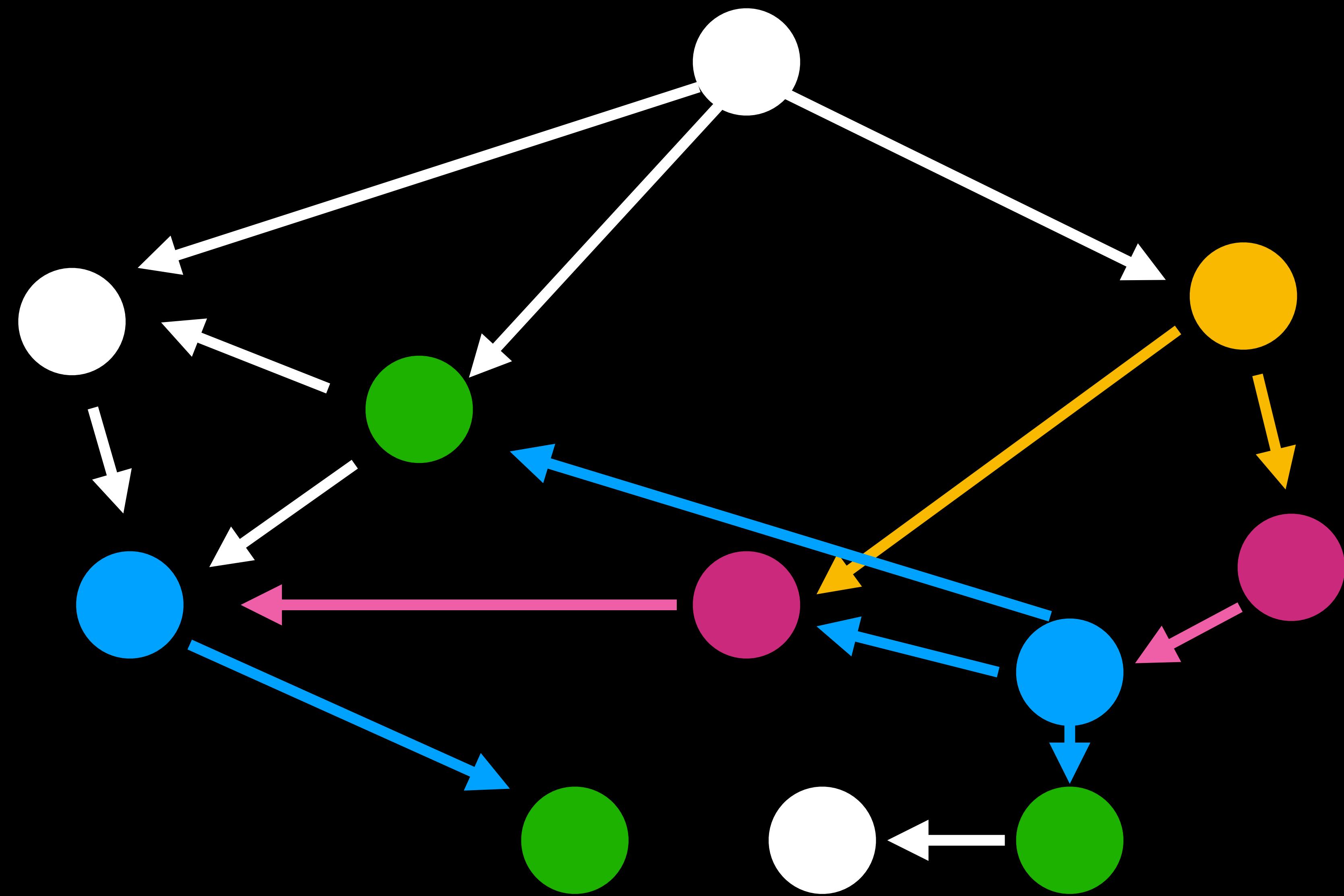


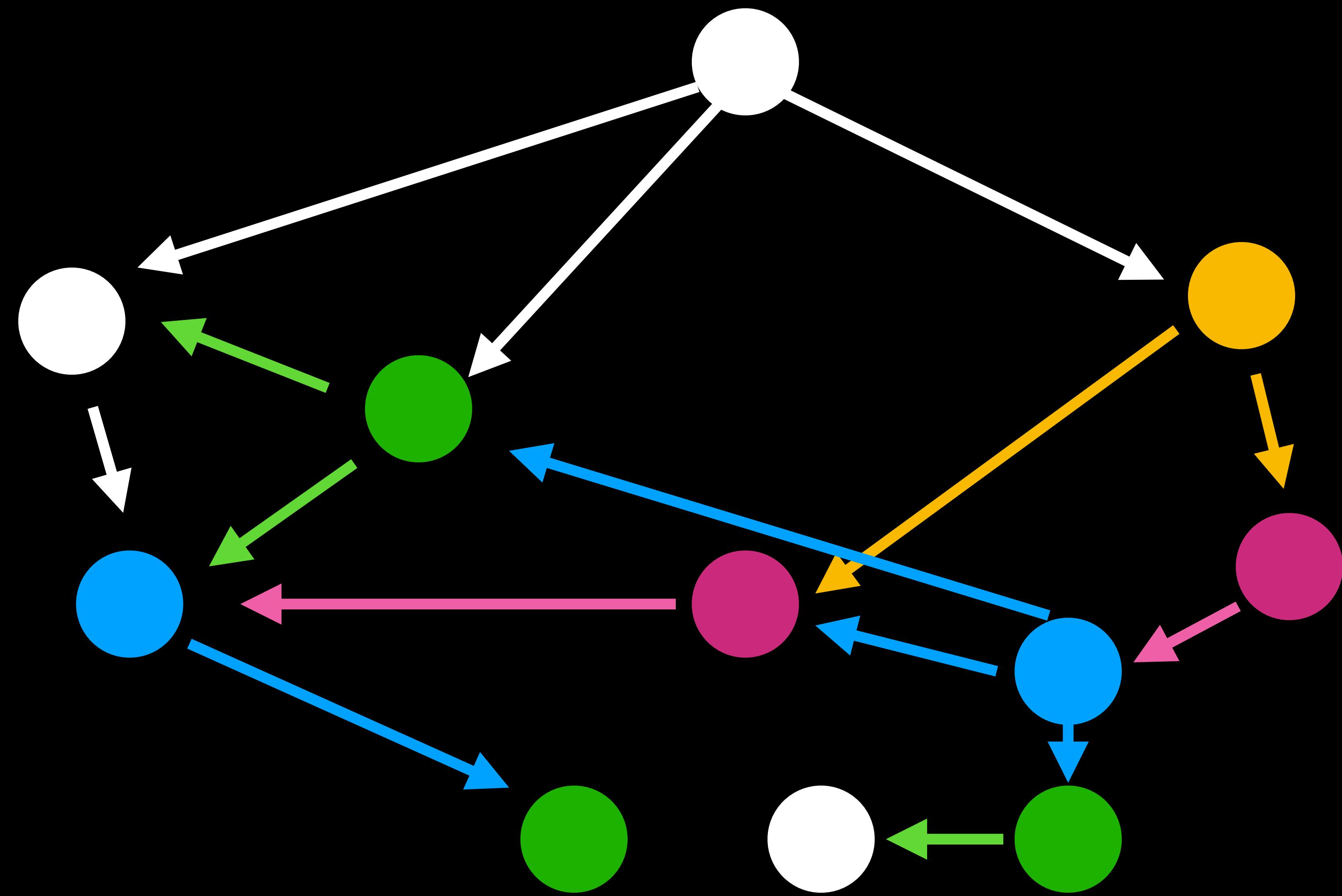


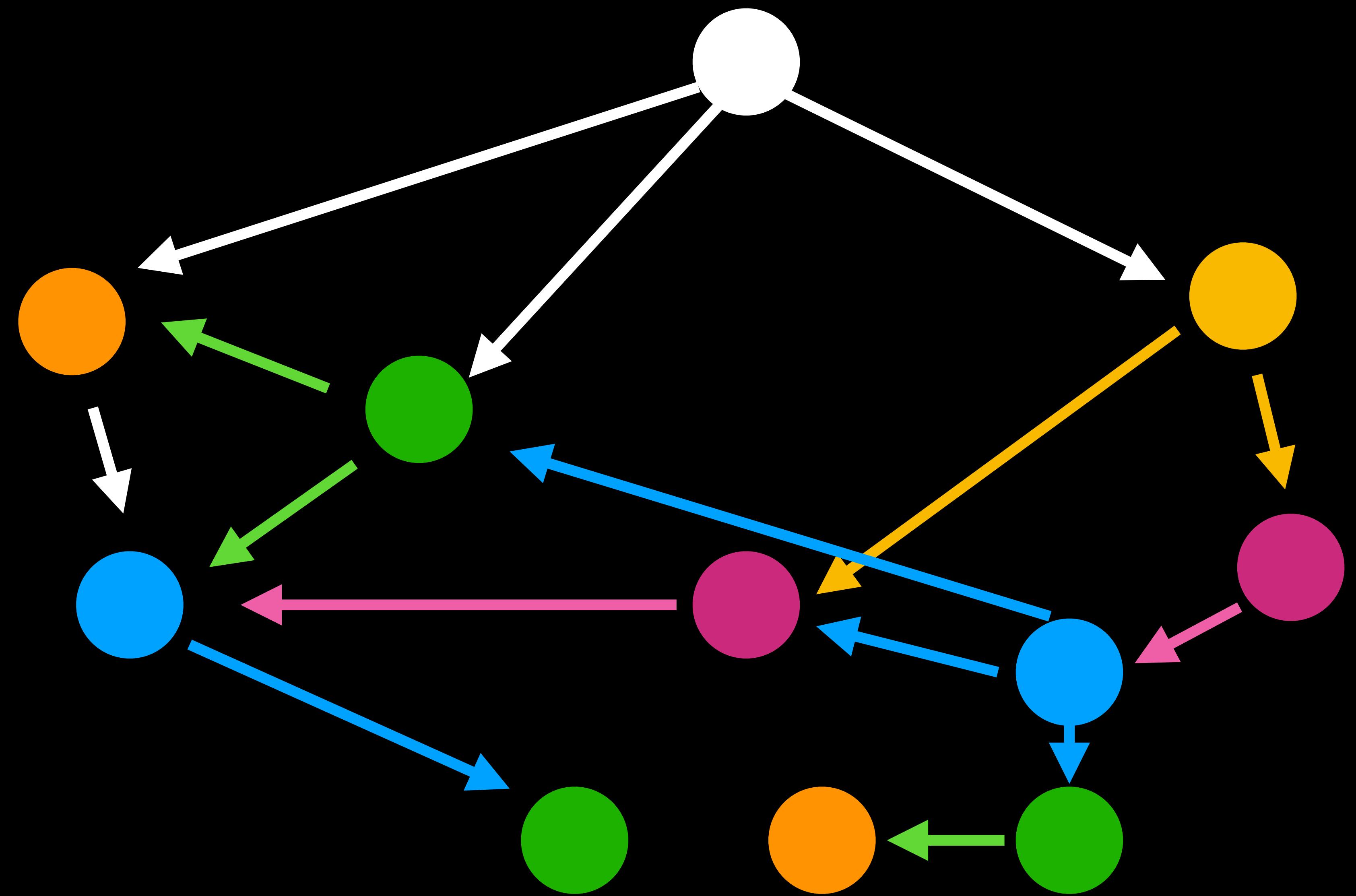


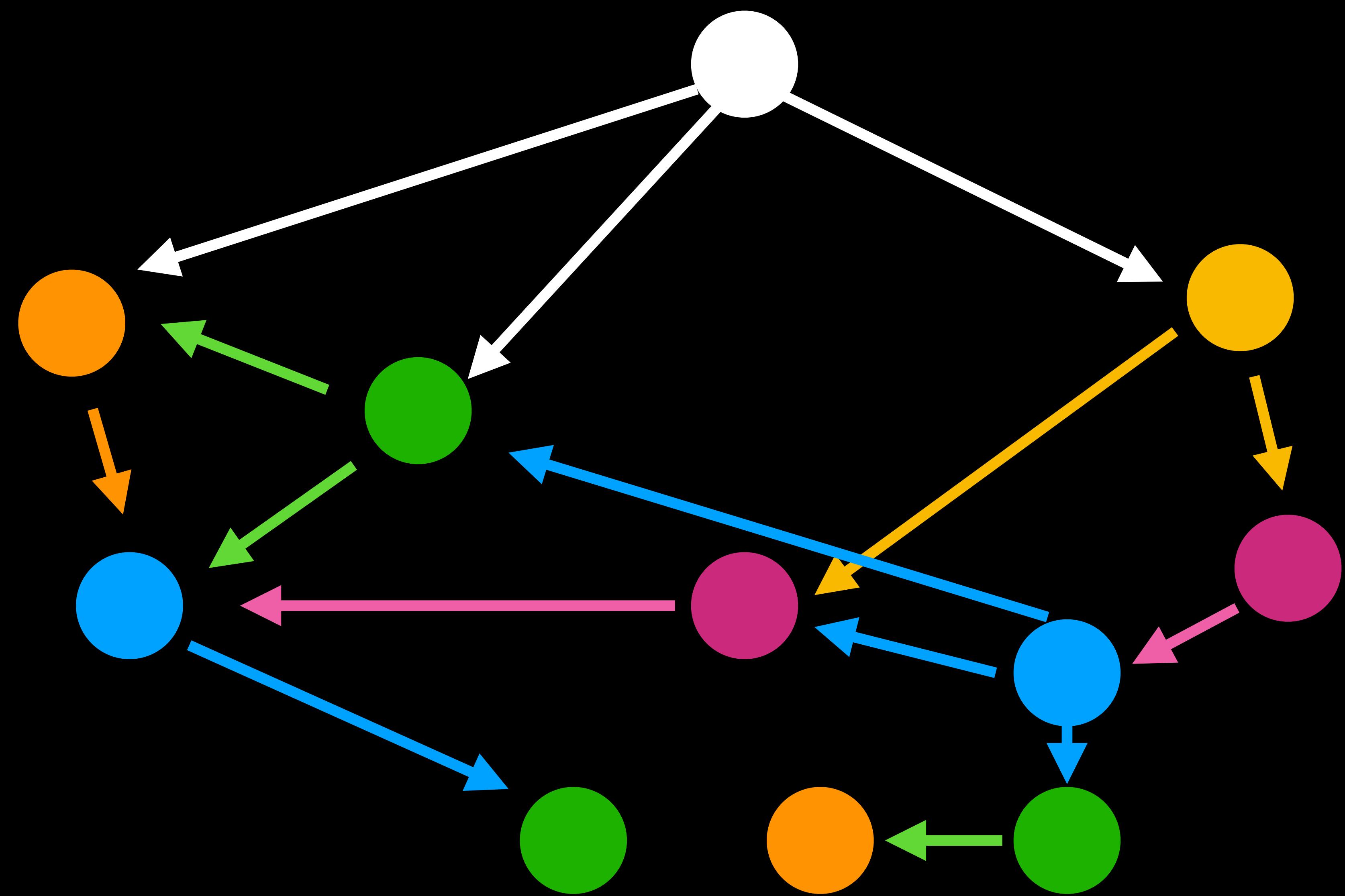


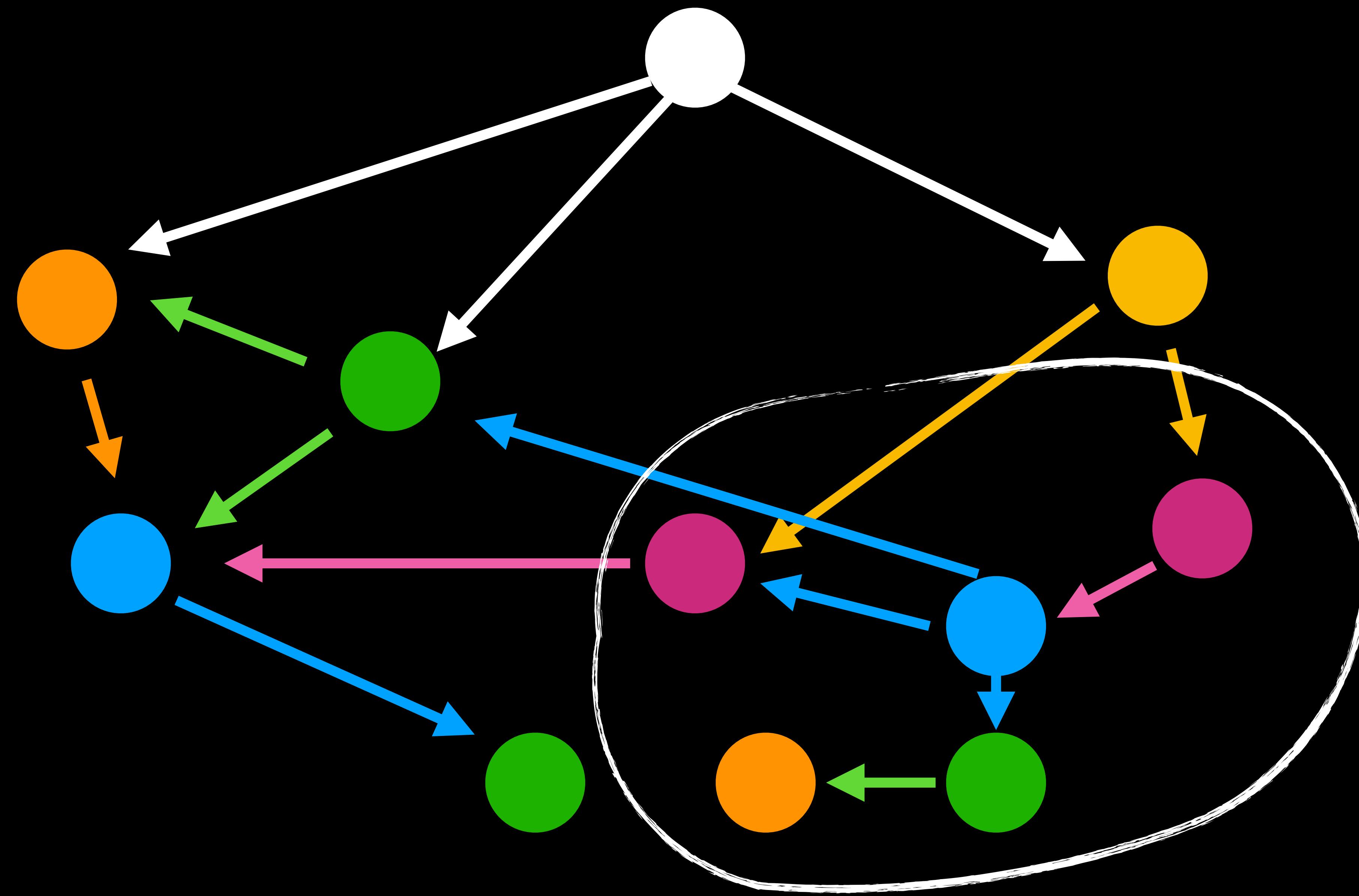


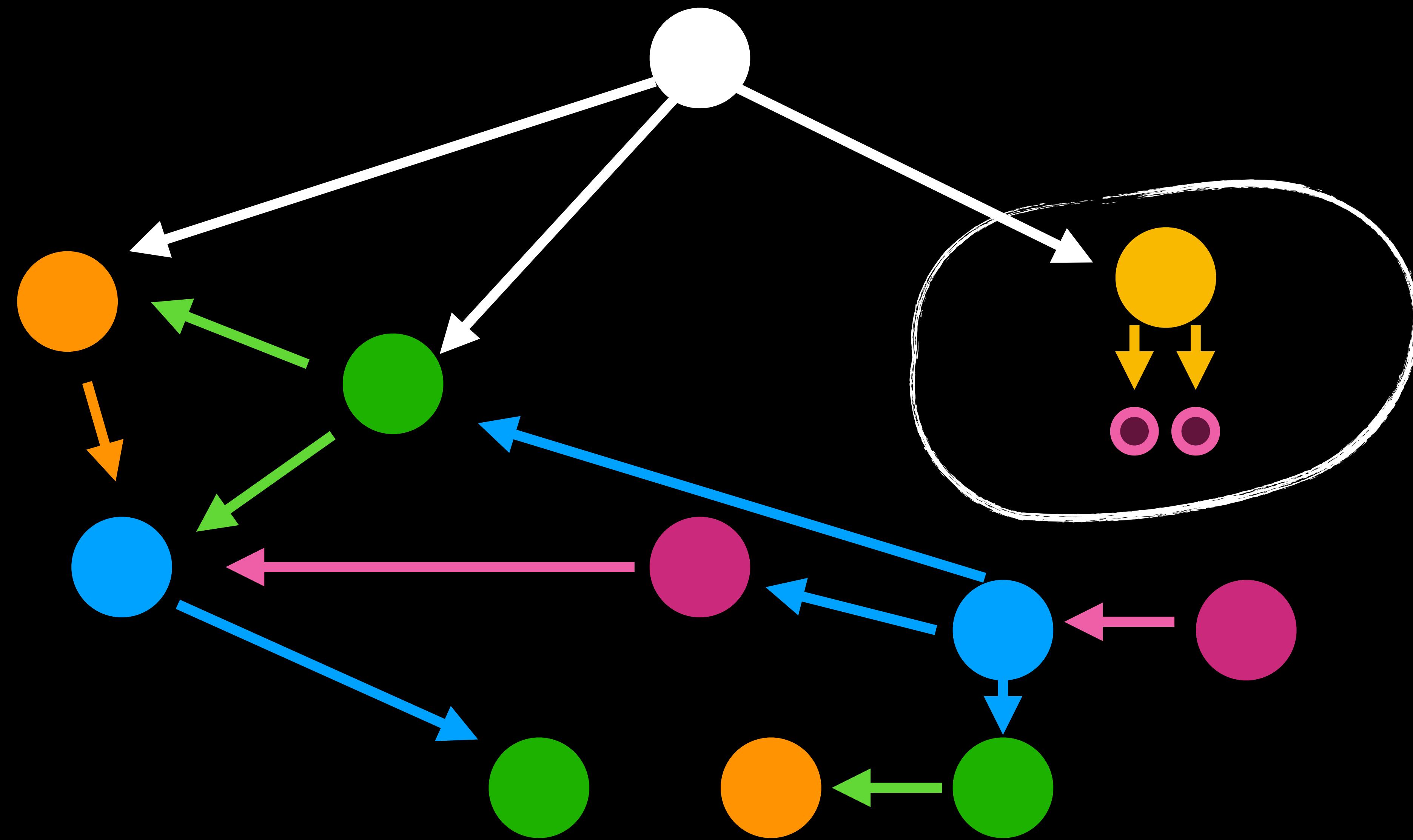












# Splitting up a Large Project into Modules

- More isolated, focused areas of code
- Faster Builds
- Fewer Dependencies
- Shorter Feedback Cycles
- Faster / Working Xcode Previews

# Start with “Easy” Modules

- The leaf nodes in your dependency graph
- Limited / No Internal Dependencies
- Limited / No External Dependencies

# Common Examples

- **Models**
- **Styleguide** (fonts, colors, etc)
- **Cross project utility code** (i.e. extensions)

# Basic Composition

- App Target
- UI
- Core

# More Advanced Composition

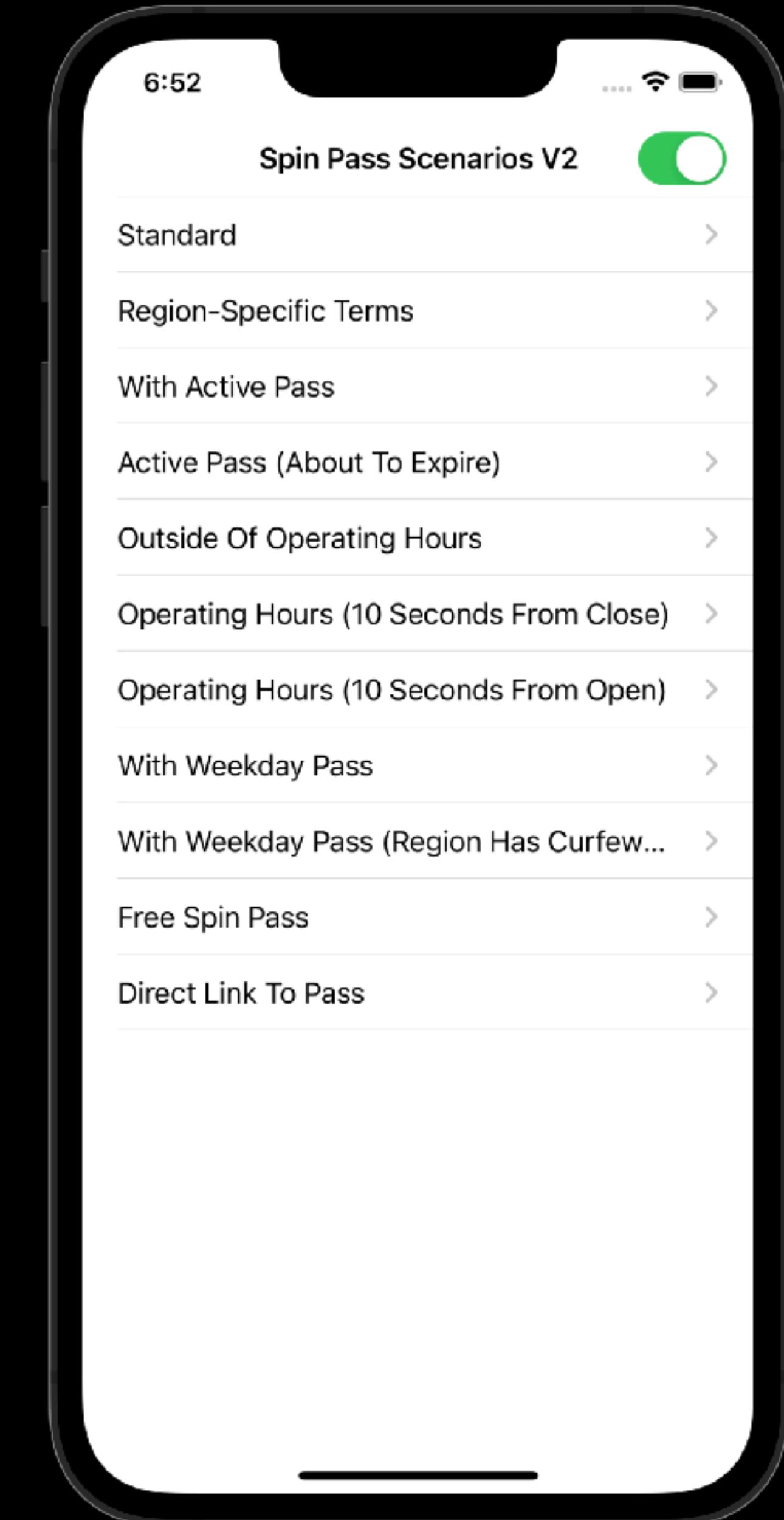
- iOS App
- watch App
- App Clip
- Models
- Persistence
- Utils
- Styleguide
- Auth
- Networking

# More Advanced Composition

- iOS App
- watch App
- App Clip
- Models
- Persistence
- Utils
- Styleguide
- Auth
- Networking
- **FeatureA**
- **FeatureB**

# Feature Modules

Example Apps offer a way to show off a feature in various states, isolated from concrete dependencies.



11:41



## Spin Pass Scenarios V2



- Standard >
- Region-Specific Terms >
- With Active Pass >
- Active Pass (About To Expire) >
- Outside Of Operating Hours >
- Operating Hours (10 Seconds From Close) >
- Operating Hours (10 Seconds From Open) >
- With Weekday Pass >
- With Weekday Pass (Region Has Curfew...) >
- Free Spin Pass >
- Direct Link To Pass >



# What is a clock?

`() -> Date`

```
struct Clock {  
    static var system: () -> Date {  
        Date.init  
    }  
  
    static func fixed(  
        hour: Int,  
        min: Int,  
        second: Int  
    ) -> () -> Date {  
        // ...  
    }  
}
```

```
var clock: () -> Date
```

```
clock = Clock.system
```

```
// or
```

```
clock = Clock.fixed(hour: 8, min: 29, second: 55)
```

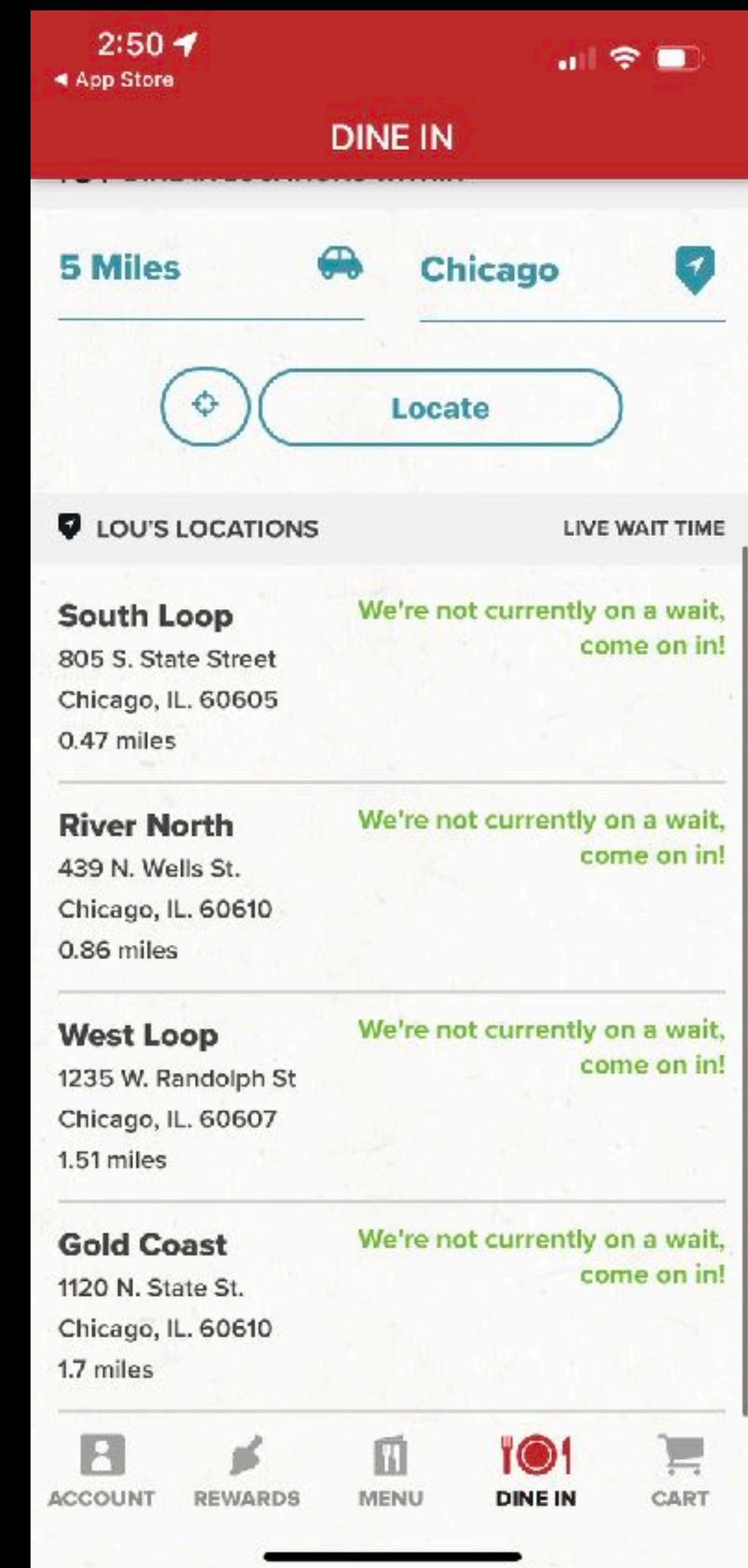
```
static func advancing(from fromDate: Date) -> () -> Date {  
}
```

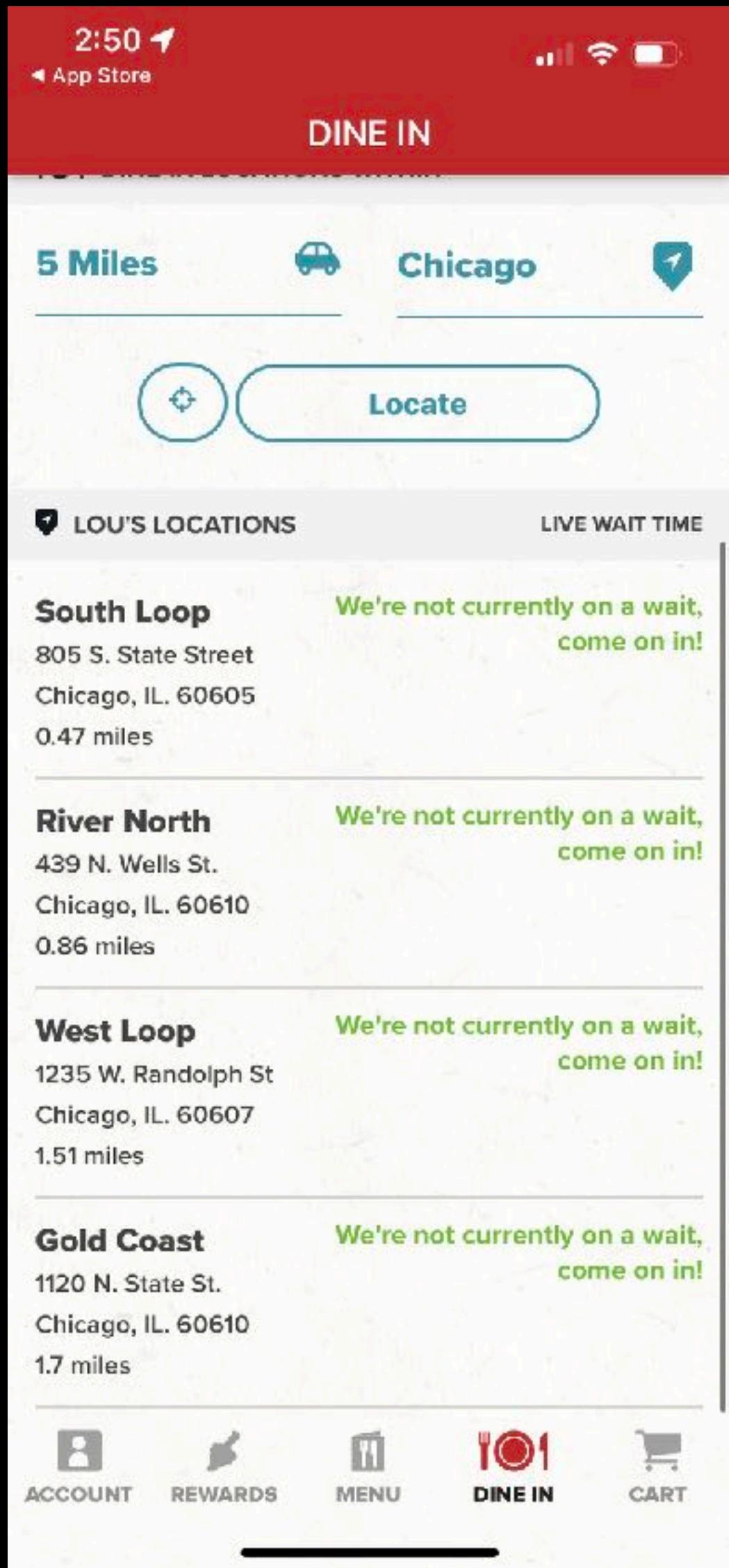
```
static func advancing(from fromDate: Date) -> () -> Date {  
    let startTime = Date()  
  
    }  
}
```

```
static func advancing(from fromDate: Date) -> () -> Date {  
    let startTime = Date()  
    return {  
  
    }  
}
```

```
static func advancing(from fromDate: Date) -> () -> Date {  
    let startTime = Date()  
    return {  
        let timeSinceCreate = Date().timeIntervalSince(startTime)  
  
    }  
}
```

```
static func advancing(from fromDate: Date) -> () -> Date {  
    let startTime = Date()  
    return {  
        let timeSinceCreate = Date().timeIntervalSince(startTime)  
        return fromDate.addingTimeInterval(timeSinceCreate)  
    }  
}
```





CoreLocation

URLSession

Date.init

LocationProvider

StoreFinderService

Clock

LocationsList

**CoreLocation**

**URLSession**

**Date.init**

**LocationProvider**

**StoreFinderService**

**Clock**

**StyleGuide**

**Delegate\***

**LocationsList**

# Taxonomy Is Important

## Foundations / Utilities

Combine operators

Autolayout helpers\*

Color functions\*

Logging utilities

## Component

Buttons

Date Picker

Size Picker

## Capability

Analytics

Telemetry

Network

Authentication

## Feature

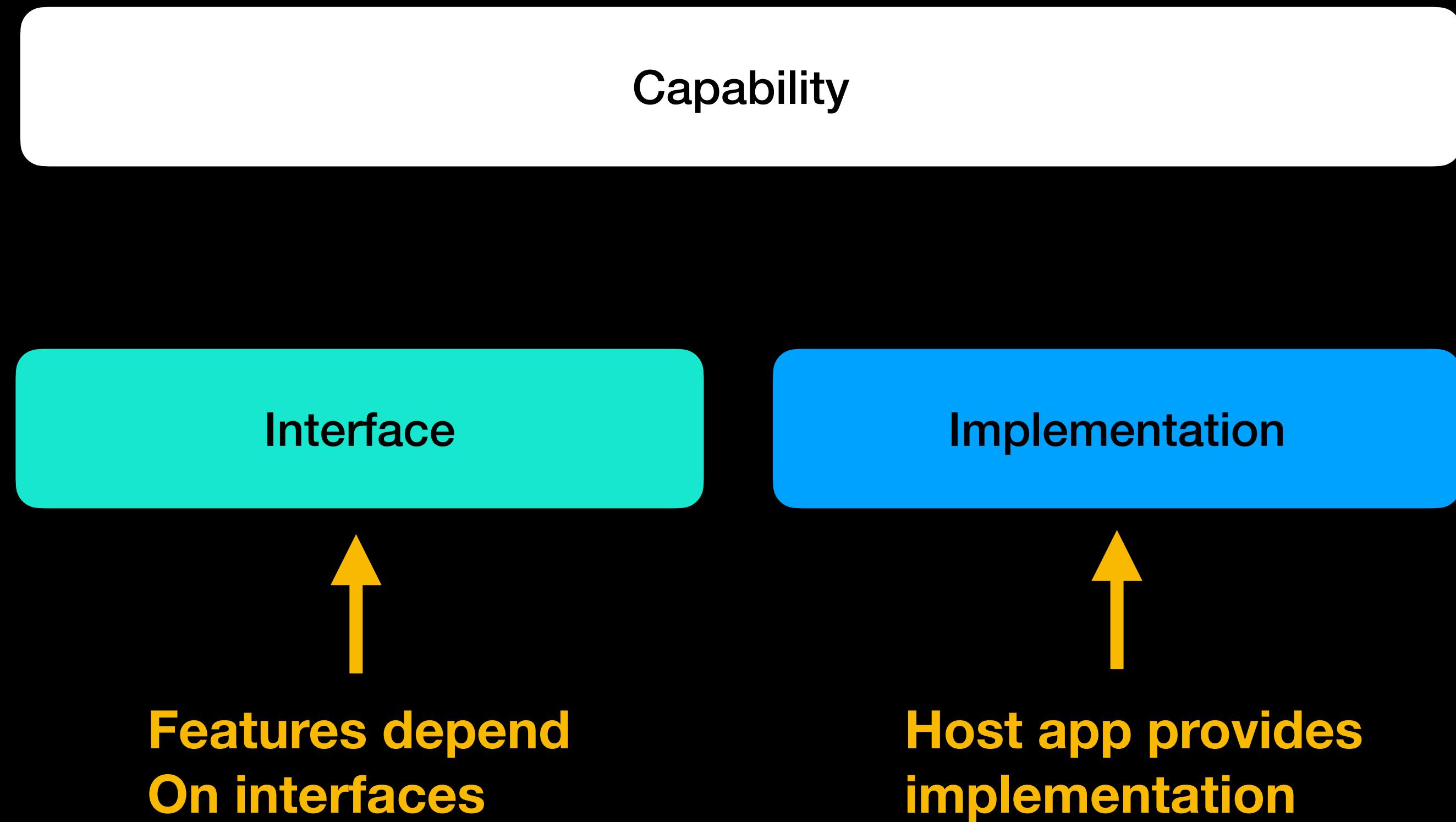
Stories

Live streams

Polls

Checkout

# Abstraction Is Critical



# Strategies and Tools

- Multiple Targets
- Nested Xcode Projects
- Swift Package Manager
- CocoaPods
- Xcodegen
- Tuist
- Bazel / Buck

```
import PackageDescription

let package = Package(
    name: "Models",
    platforms: [.iOS(.v15)],
    products: [
        .library(
            name: "Models",
            targets: ["Models"]),
    ],
    dependencies: [],
    targets: [
        .target(
            name: "Models",
            dependencies: []),
        .testTarget(
            name: "ModelsTests",
            dependencies: ["Models"]),
    ]
)
```

# Swift Package Manager

# Swift Package Manager

- Great if you already use SwiftPM
- Xcode can open Package.swift directly, no projects
- Requires familiarity of Package manifest api to define configuration, dependencies, etc

# CocoaPods

# CocoaPods

- Great if you already use Pods
- Define Pods for internal modules
- Path based dependencies for local modules

# Benefits of CocoaPods

- Podfile is ruby, so you can work around issues
- Very scriptable
- Can integrate with Xcodegen
- Lots of library support\*

# Challenges with CocoaPods

- If you have SPM dependencies you're in for some not-fun-work
- No Bundle.module
- pod lib create is antiquated
- Take care to specify local pods / podspecs or risk global search
- Duplicate Pods folders
- Industry is moving rapidly to SPM

# Downsides to modularization

- Requires some discipline
- Without automation, consistency is hard to ensure
- Easy to make breaking changes if you don't have CI running the entire build suite

# Thank you!



[@bens@mastodon.xyz](https://@bens@mastodon.xyz)



[@subdigital](https://@subdigital)

[nsscreencast.com](https://nsscreencast.com)

[combineswift.com](https://combineswift.com)

[benscheirman.com](https://benscheirman.com)

