goto; copenhagen

GOTO Copenhagen 2019
Conference Nov. 18 - 20

Scaling up an iOS codebase

Tjeerd in ’t Veen

goto; copenhagen

GOTO Copenhagen 2019

Conference Nov. 18 - 20

Click ‘Rate Session’ to rate session and ask questions.

Follow us @gotocph

goto; copenhagen

GOTO Copenhagen 2019

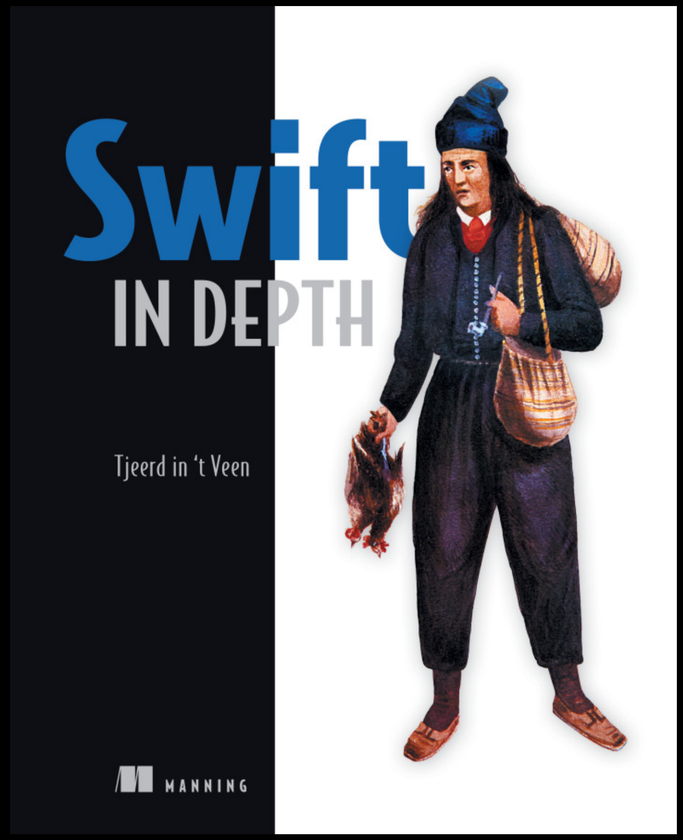
Conference Nov. 18 - 20

Did you **remember to rate** the previous session ?

Follow us @gotocph

Tjeerd in ’t Veen
@tjeerdintveen

ING



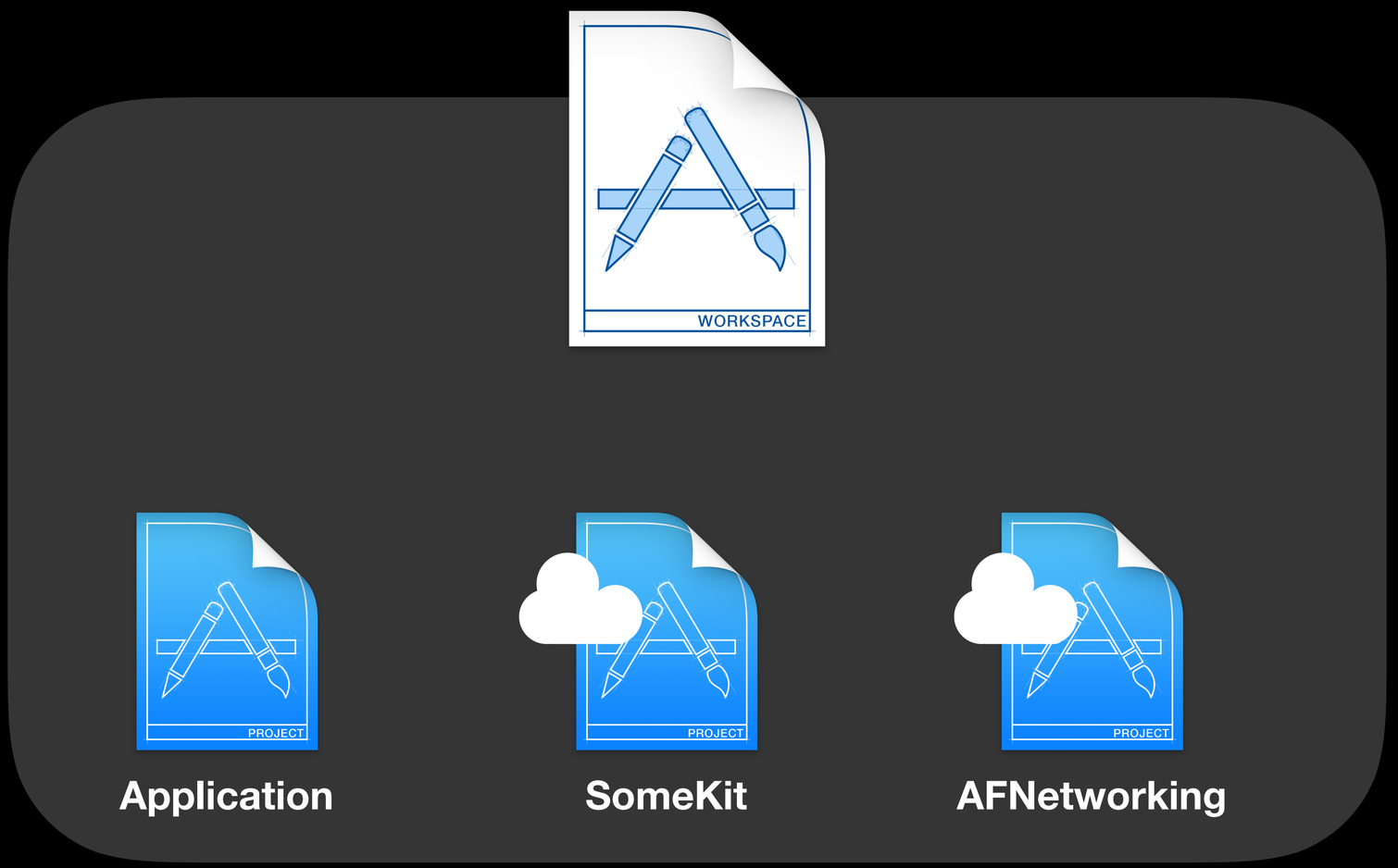
Why this talk

What we'll cover

- Handling a growing codebase
- Thinking in modules
- Versioning in practice
- Preventing source-breaking changes
- Handling dependency hell
- Organizational challenges
- Package managers


PROJECT


WORKSPACE


WORKSPACE
PROJECT
PROJECT
PROJECT
Application
SomeKit
AFNetworking

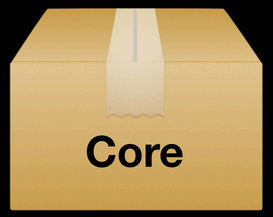
Application

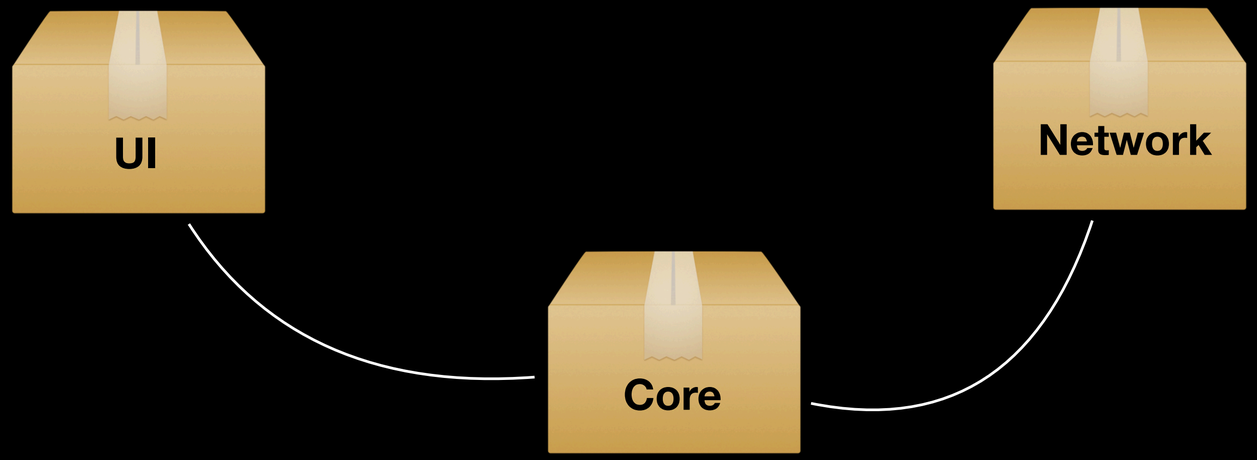
Features

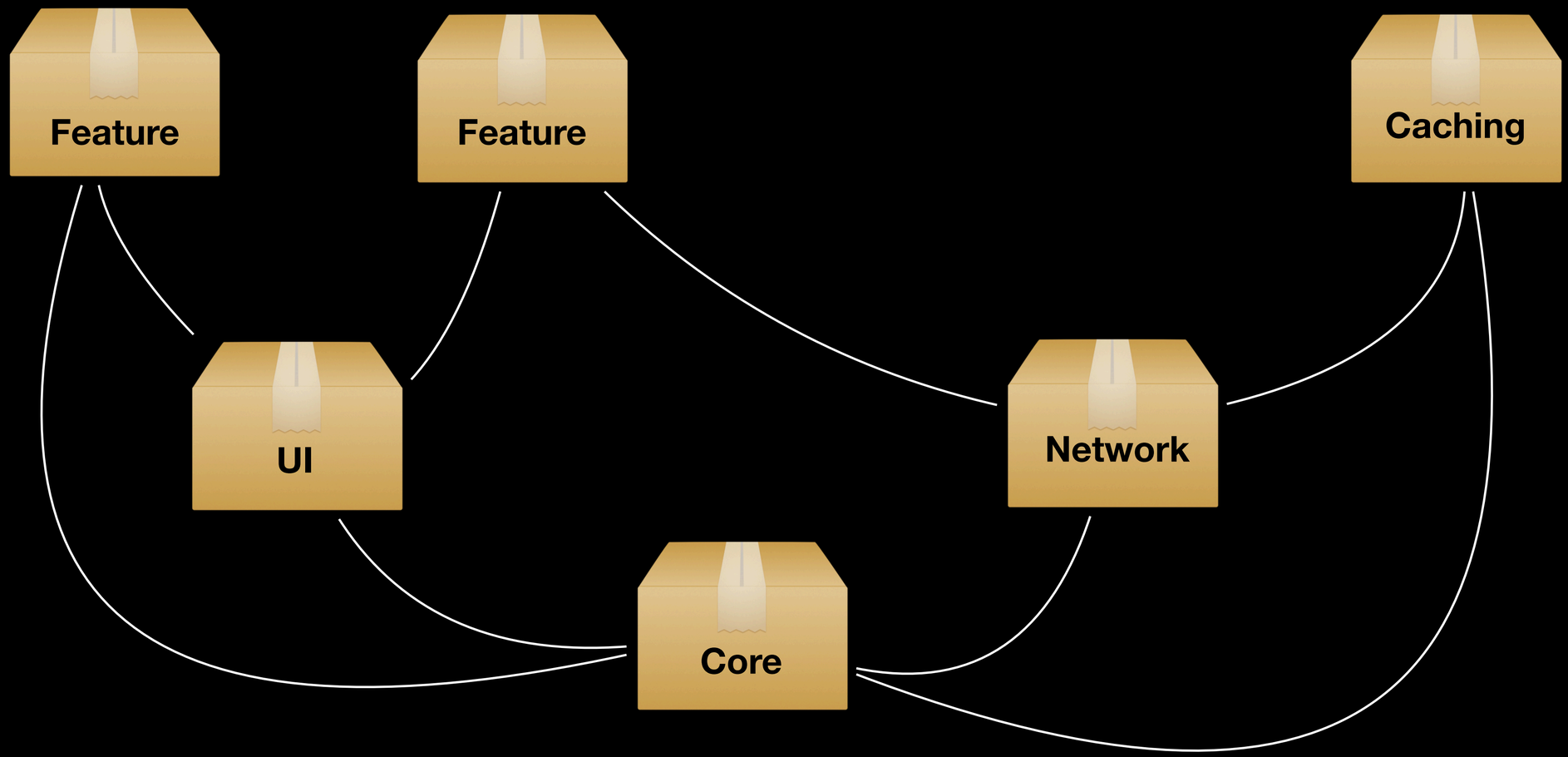
UI library

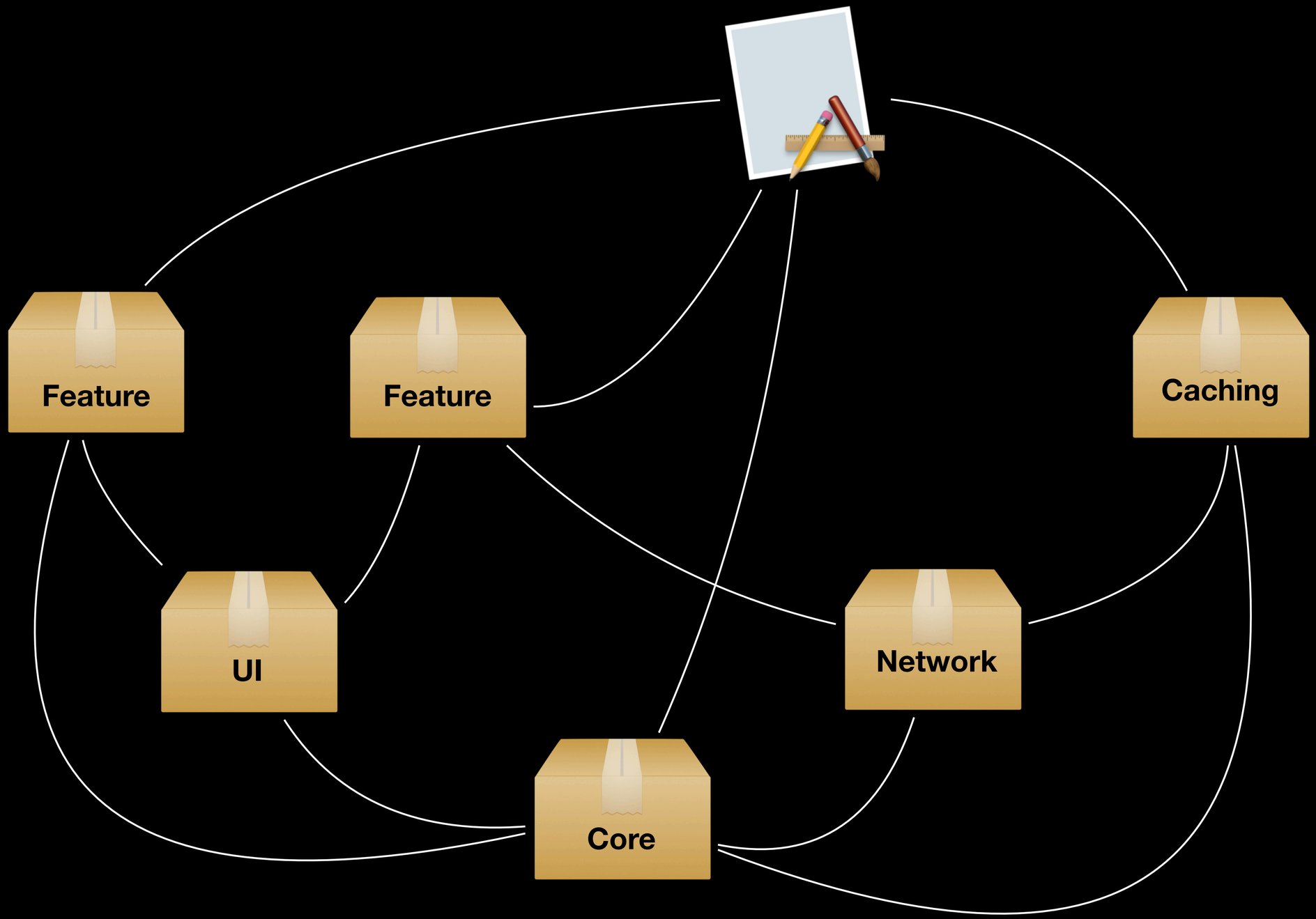
Networking

Core library


Core


UI
Network
Core


Feature
Feature
Caching
UI
Network
Core

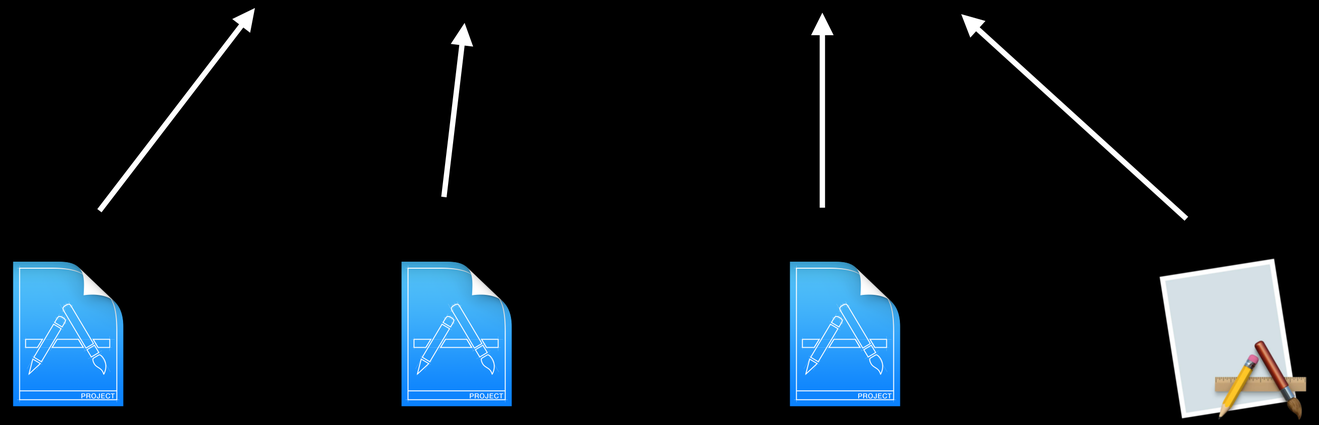

Feature
Feature
Caching
UI
Network
Core

The extraction process

- Cut the wires to the application
- Review the public API

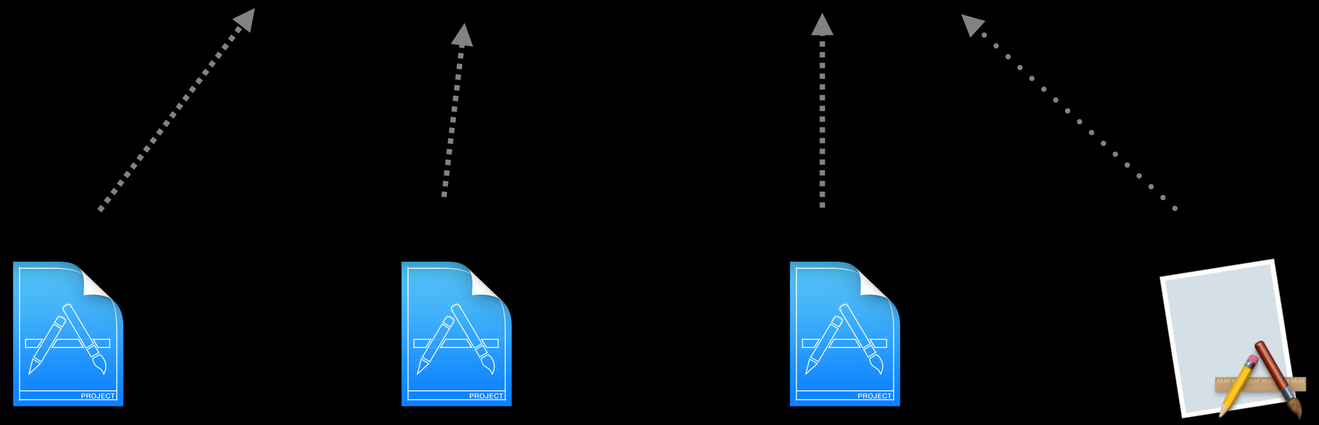
Access levels


```
public func calculatePlan() -> WorkoutPlan {
    ...
}
```



Access levels

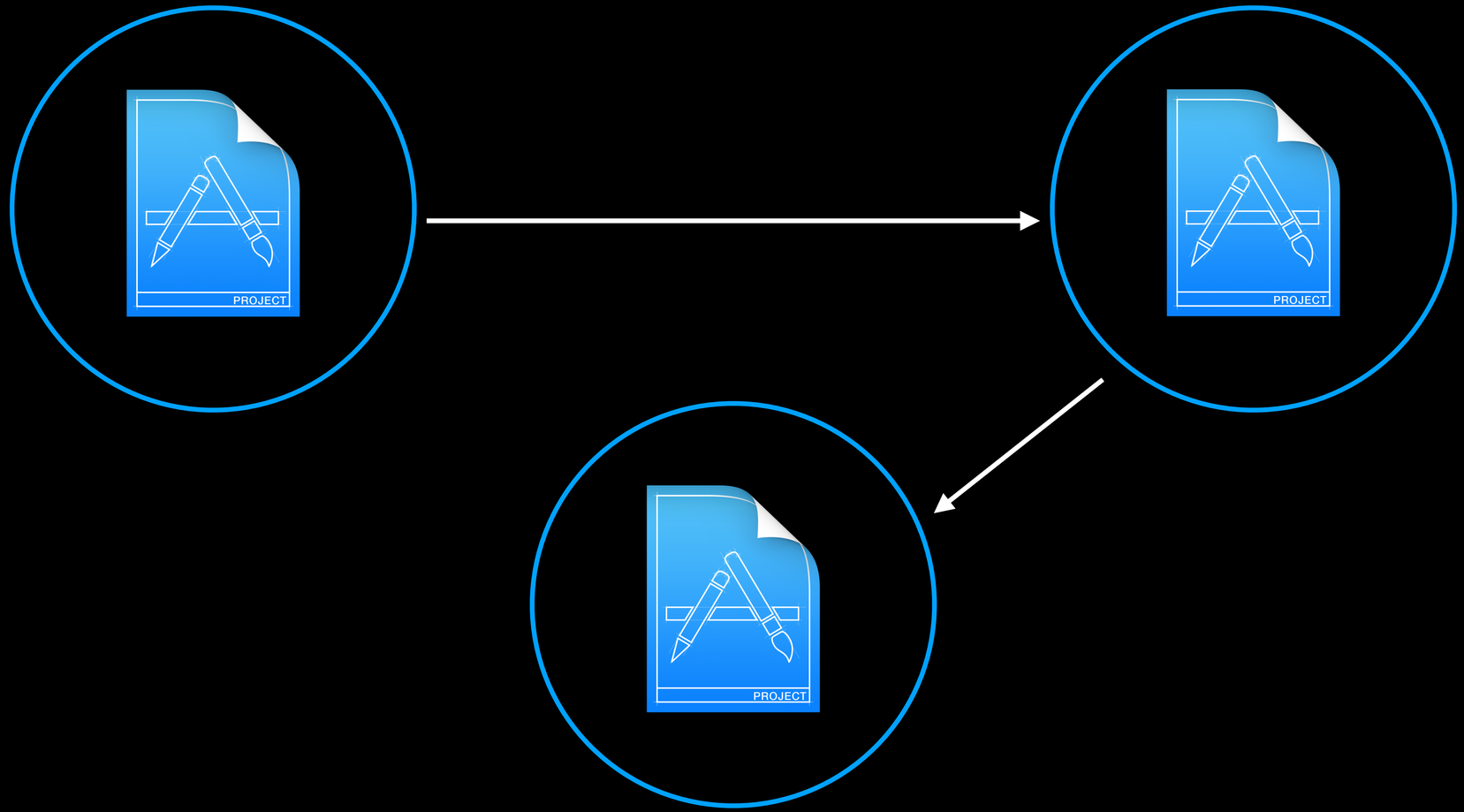

```
internal func calculatePlan() -> WorkoutPlan {
    ...
}
```



Access levels

```
@testable import WorkoutKit

let plan = calculatePlan()
```

Minimize public API



The extraction process

- Cut the wires to the application
- Review the public API
- Test public API
- Add documentation

A healthy module (checklist)

- ✓ CHANGELOG.md
- ✓ README.md with quick start guide
- ✓ Intuitive api
- ✓ Public API is tested
- ✓ Sample app for examples and UITests
- ✓ A “how to add issues or fixes” guide
- ✓ Doc comments (Quick help)

UI elements

```
public enum SettingsUIElements {
   static let profileButton = XCUIApplication().app.buttons["Profile"]
   static let aboutButton = XCUIApplication().app.buttons[“About"]
   …
}
```

Using local modules

Pros

- ✓ Hard boundaries between code
- ✓ Smaller scope of reasoning
- ✓ Better access level control
- ✓ Easier testing
- ✓ Compile times are lower

Cons

- – Code is fragmented
- – Still monolithic

Inter-workspace

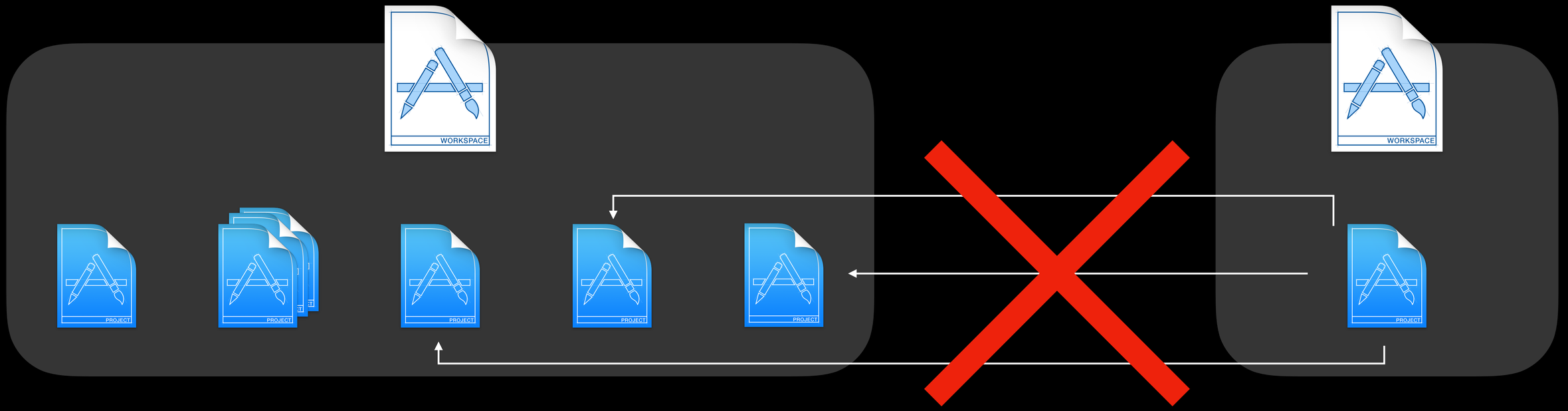
Application

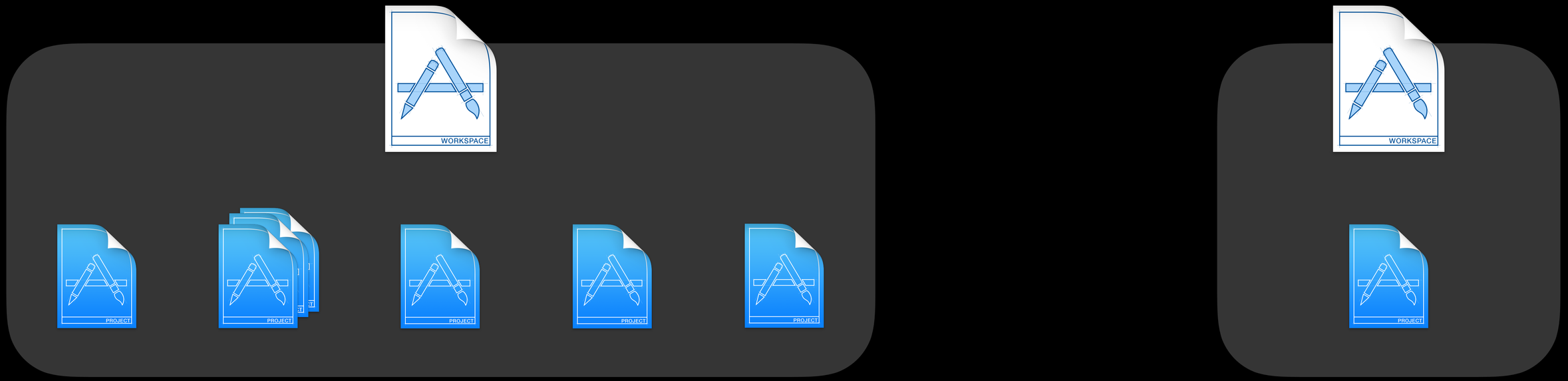
Features

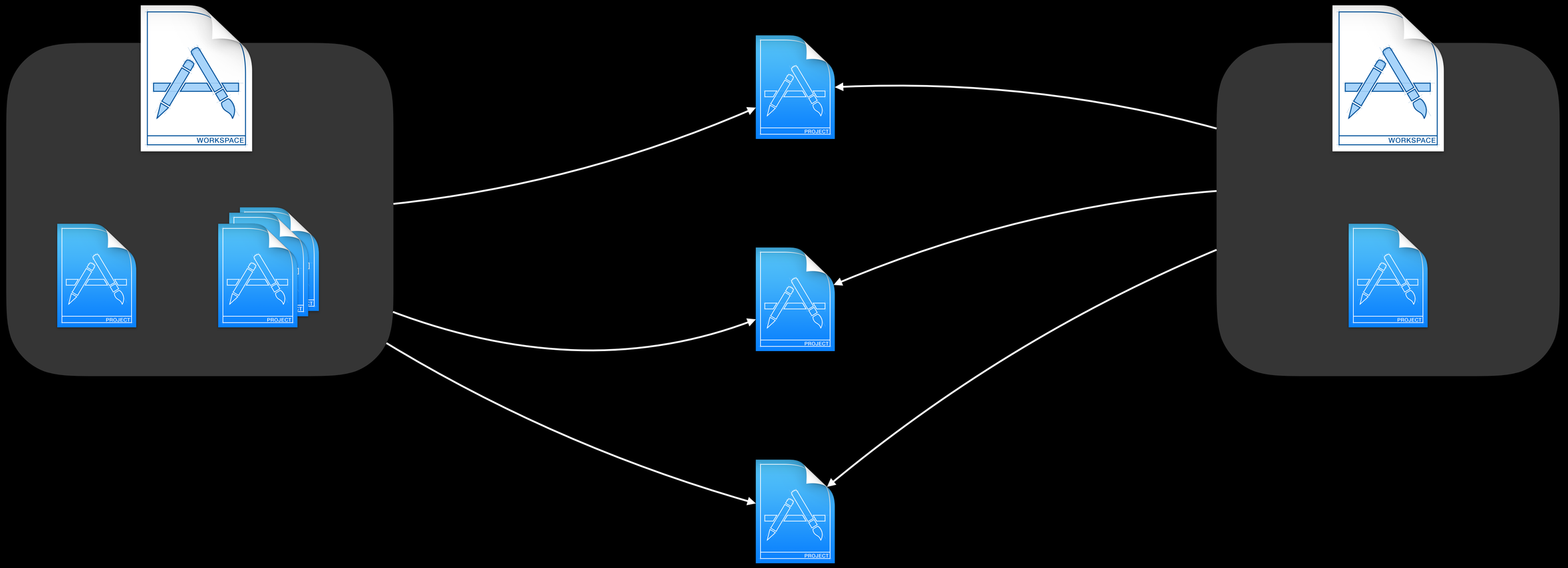
UI library

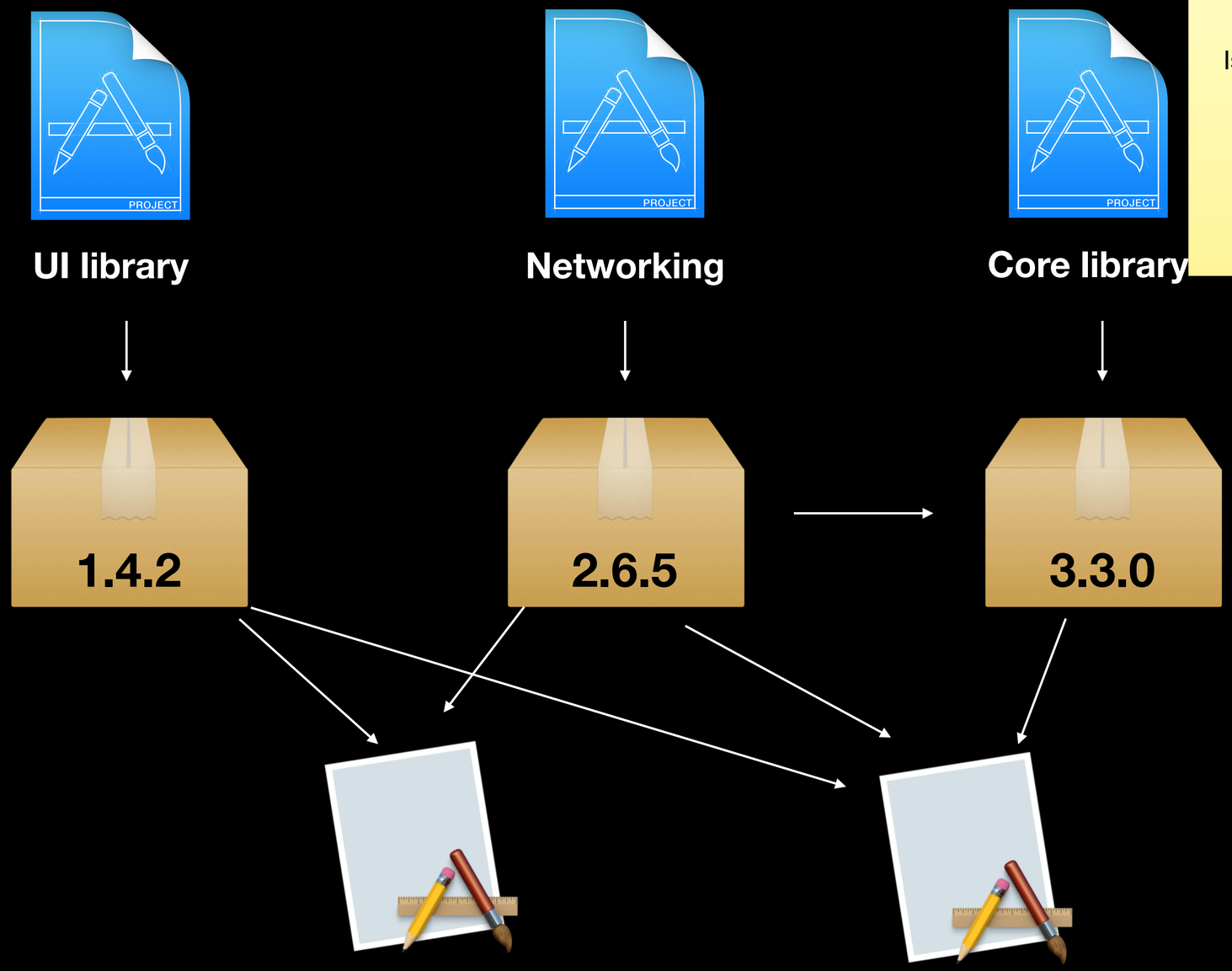
Networking

Core library


WORKSPACE
WORKSPACE
PROJECT
PROJECT
PROJECT
PROJECT
PROJECT
PROJECT


WORKSPACE
PROJECT
PROJECT
PROJECT
PROJECT
PROJECT
WORKSPACE
PROJECT


WORKSPACE
PROJECT
PROJECT
PROJECT
PROJECT
PROJECT
WORKSPACE
PROJECT



Is app icon correct?

Semver
Major.Minor.Patch

Semver

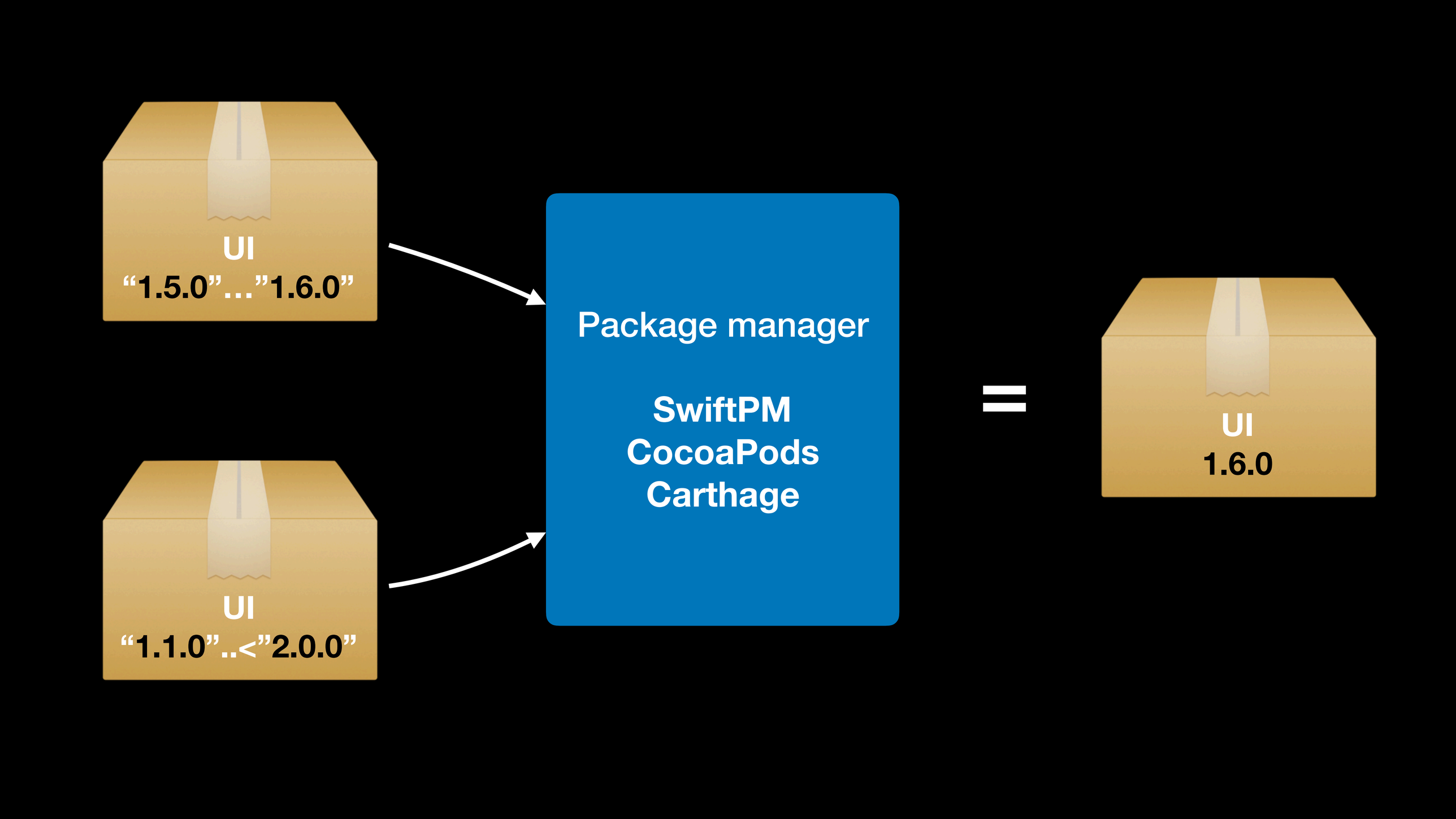
3.4.7

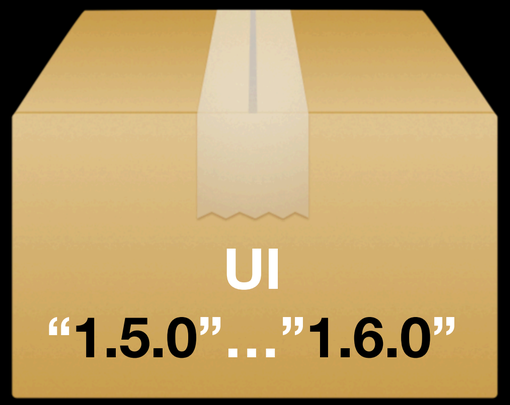
Semver

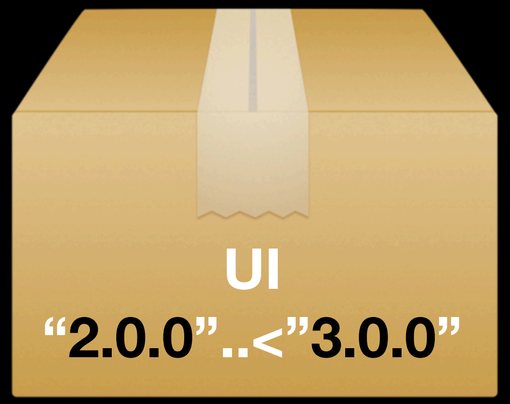
3.4.7

Semver

3.4.7

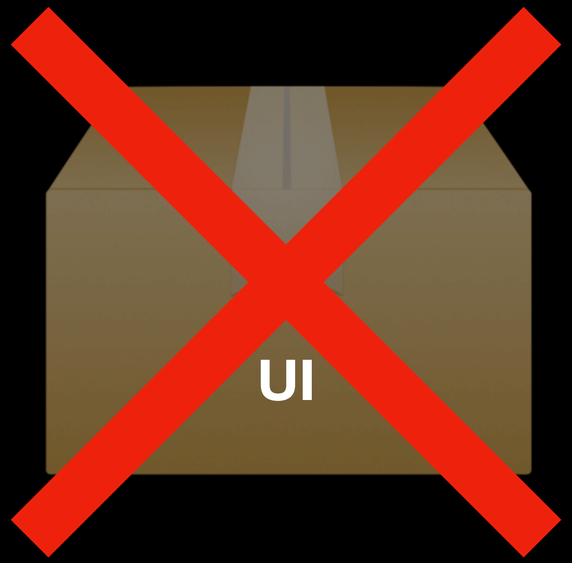

UI
“1.5.0”…”1.6.0”
UI
“1.1.0”..<”2.0.0”
Package manager
SwiftPM
CocoaPods
Carthage
=
UI
1.6.0

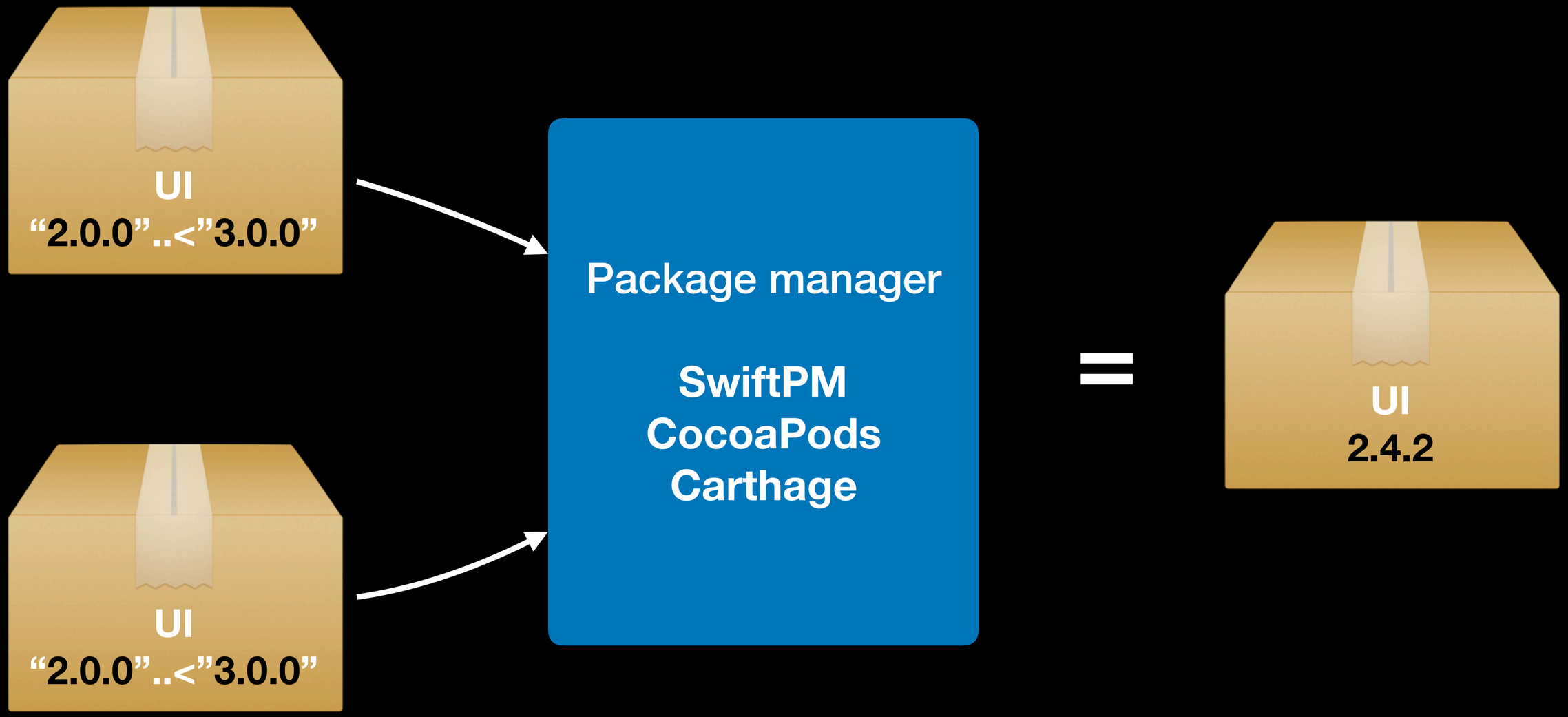




Package manager

SwiftPM
CocoaPods
Carthage

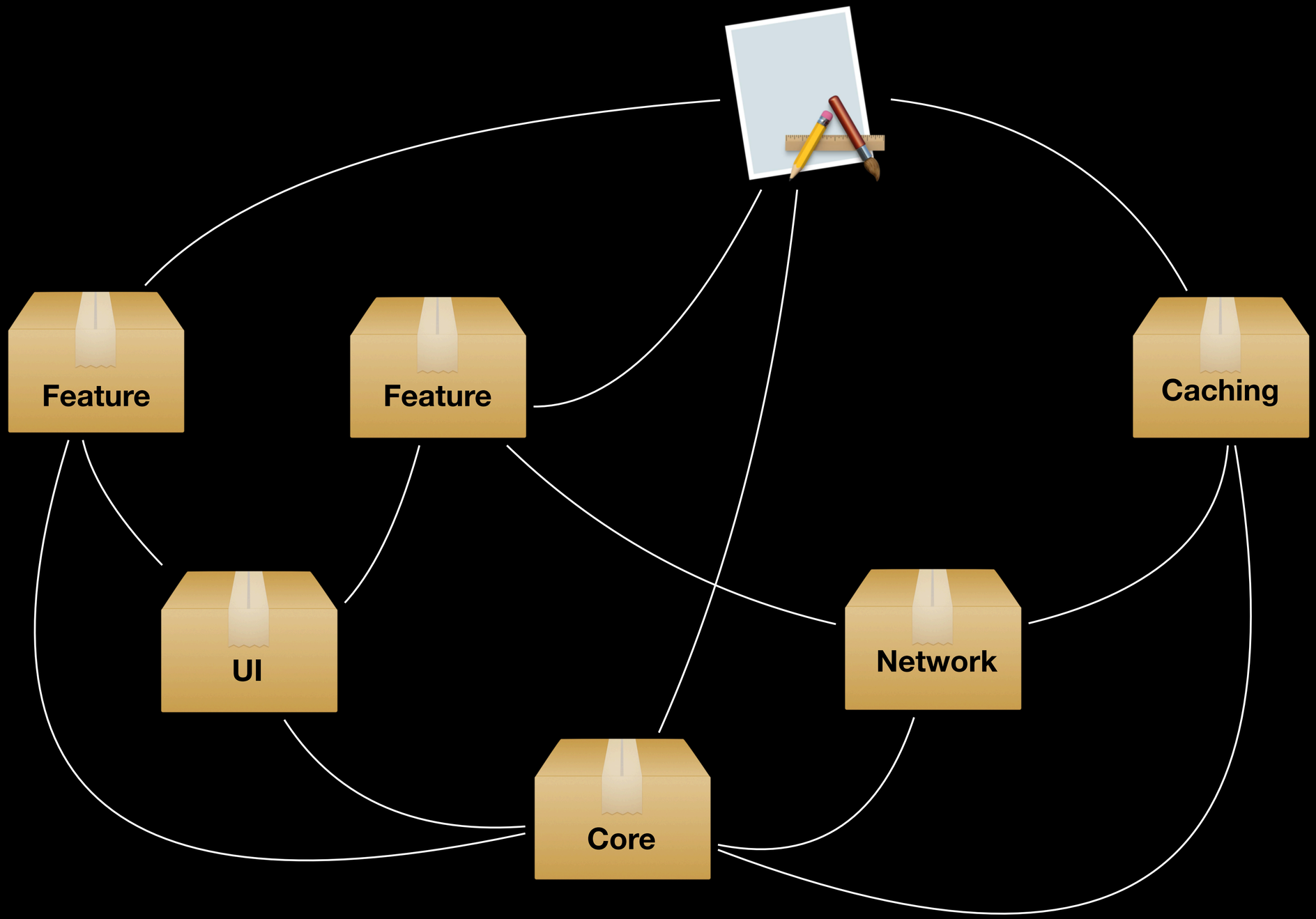


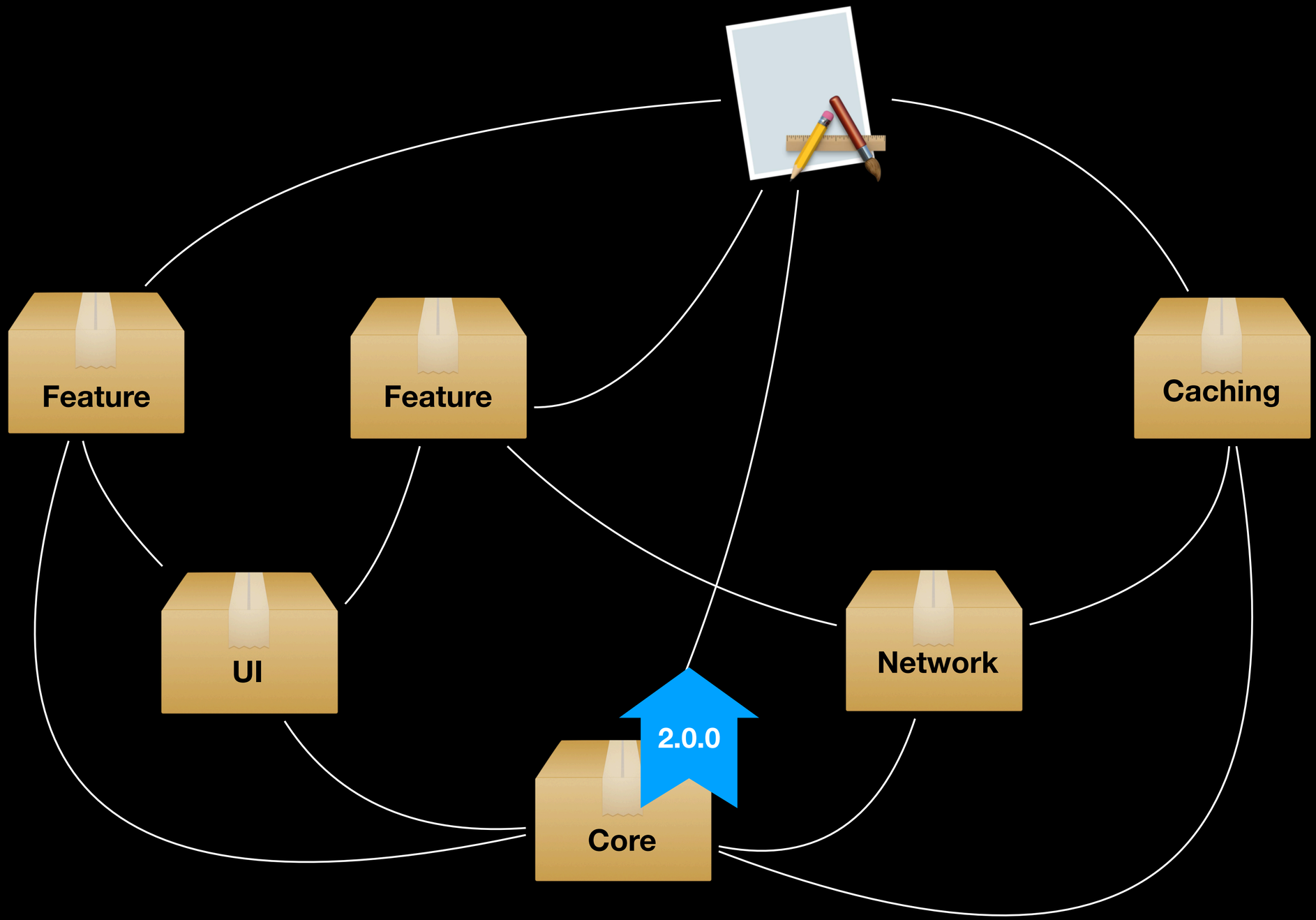

UI
“2.0.0”..<”3.0.0”
UI
“2.0.0”..<”3.0.0”
Package manager
SwiftPM
CocoaPods
Carthage
=
UI
2.4.2

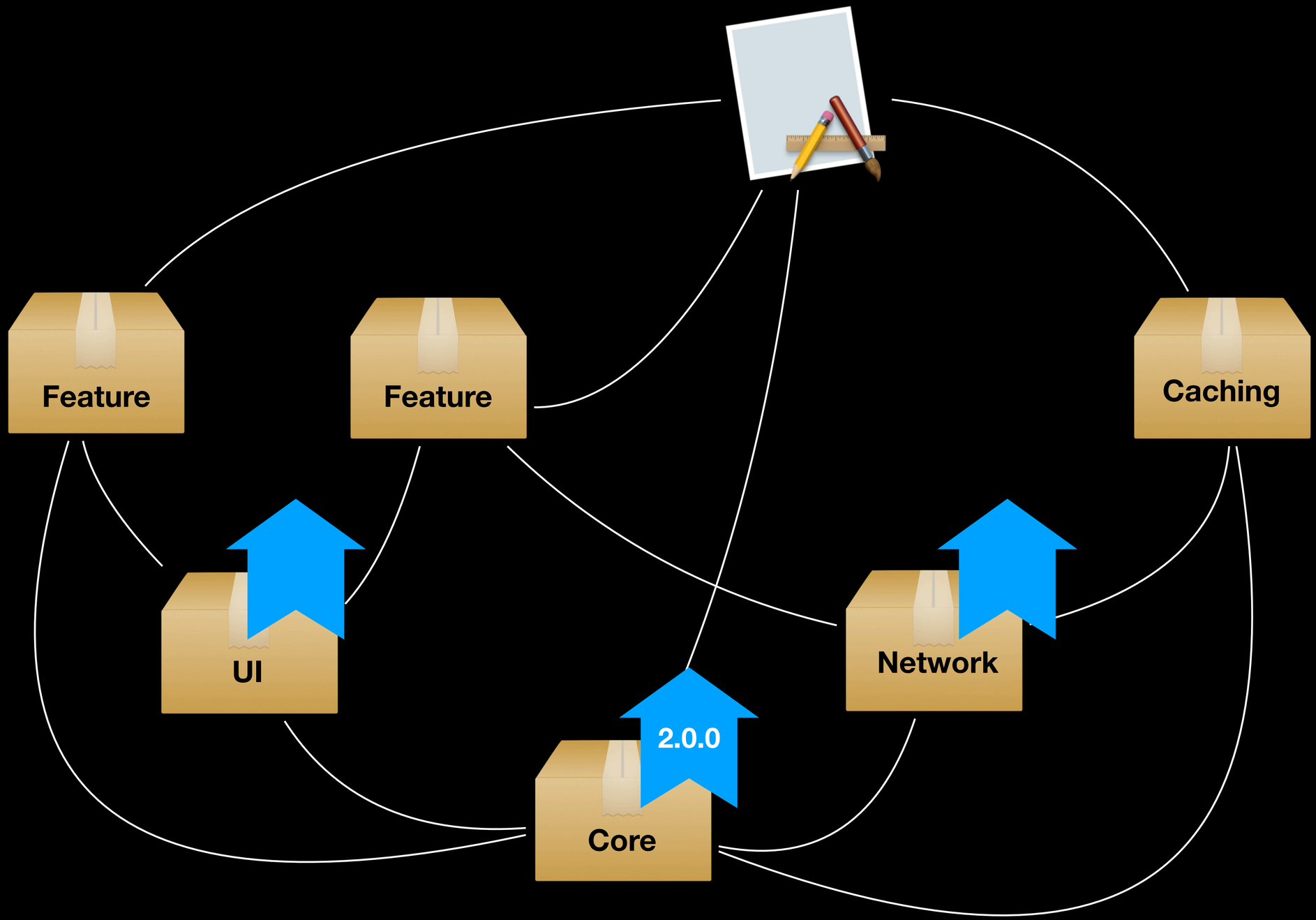
Package.resolved
Cartfile.resolved
Podfile.lock

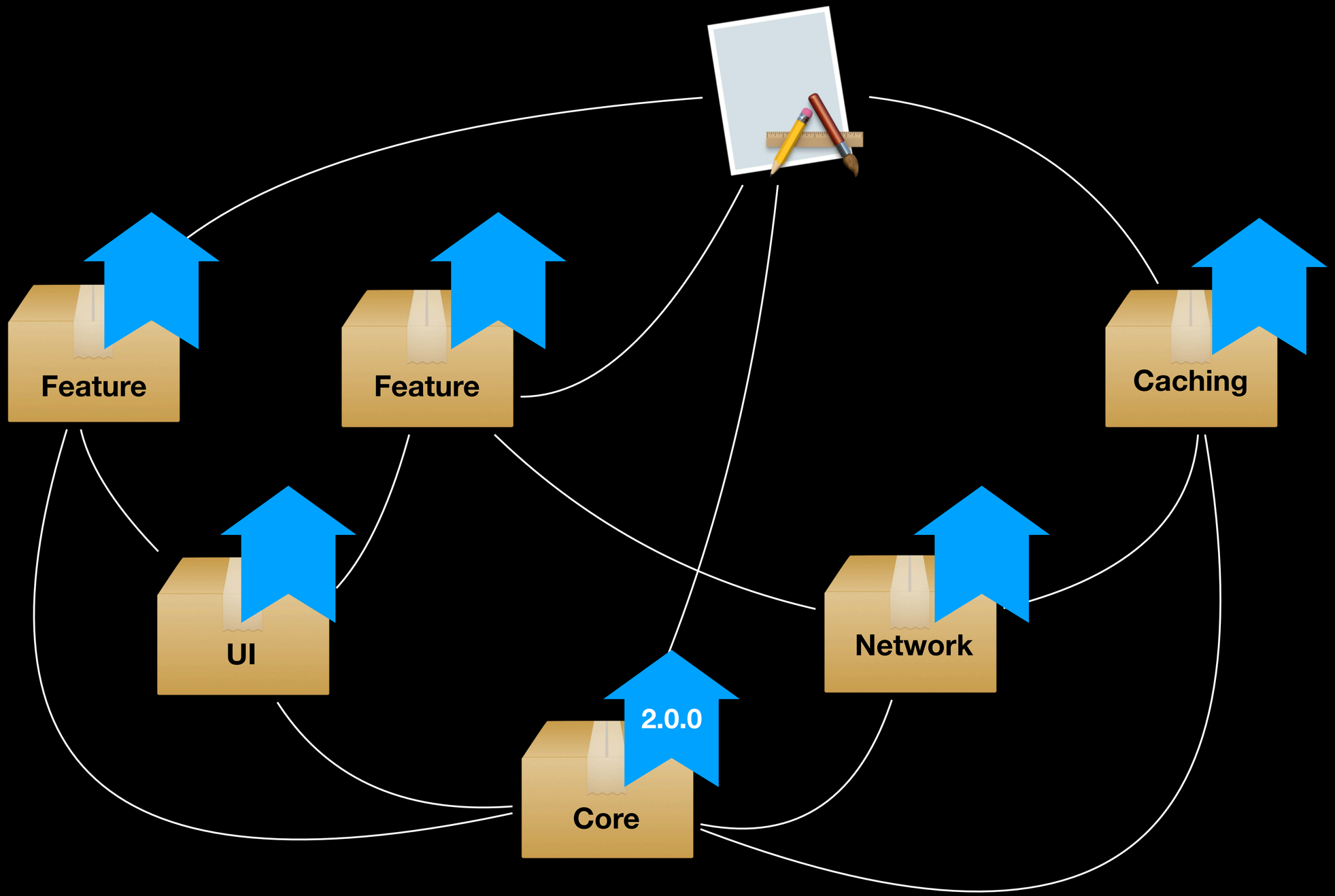
Feature A “1.2.6”
Feature B “2.3.1”
Feature C “5.2.2”
Feature D “1.0.0”
Feature E “1.2.3”
UI “4.3.3”
Networking “3.12.0”
Core “1.2.1”

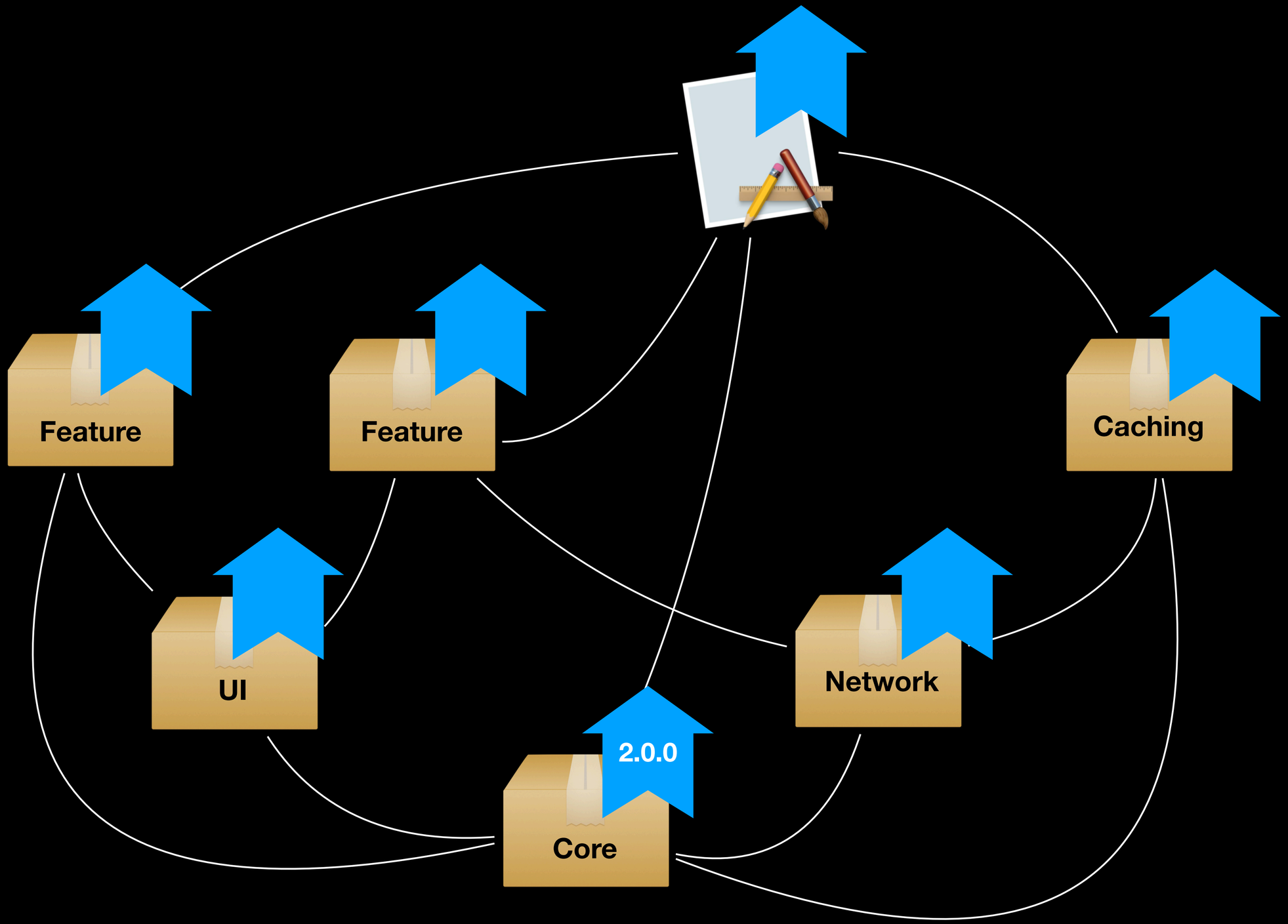
The impact of majors

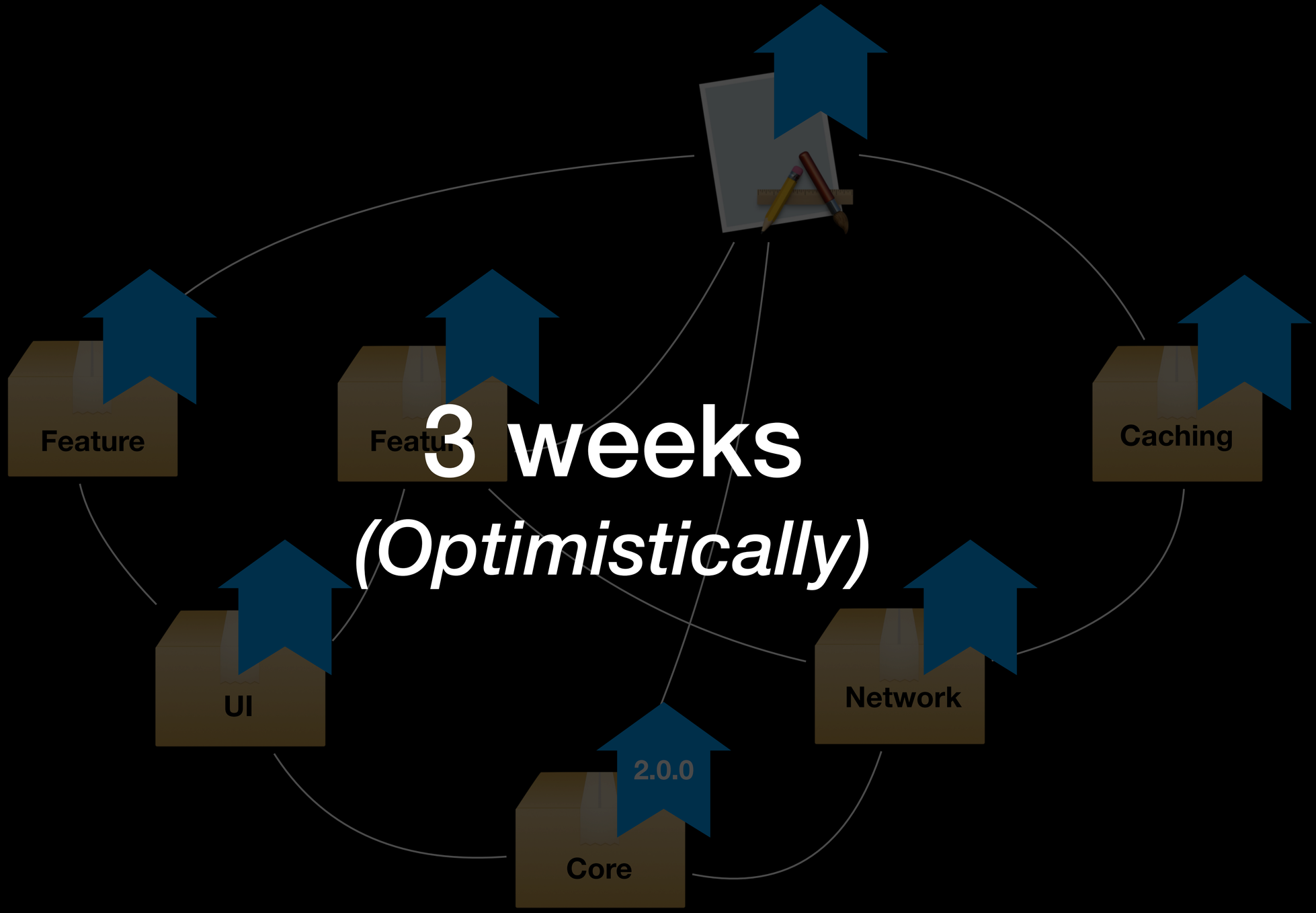

Feature
Feature
Caching
UI
Network
Core


Feature
Feature
Caching
UI
Network
2.0.0
Core


Feature
Feature
Caching
UI
Network
2.0.0
Core


Feature
Feature
Caching
UI
Network
2.0.0
Core


Feature
Feature
Caching
UI
Network
2.0.0
Core



3 weeks

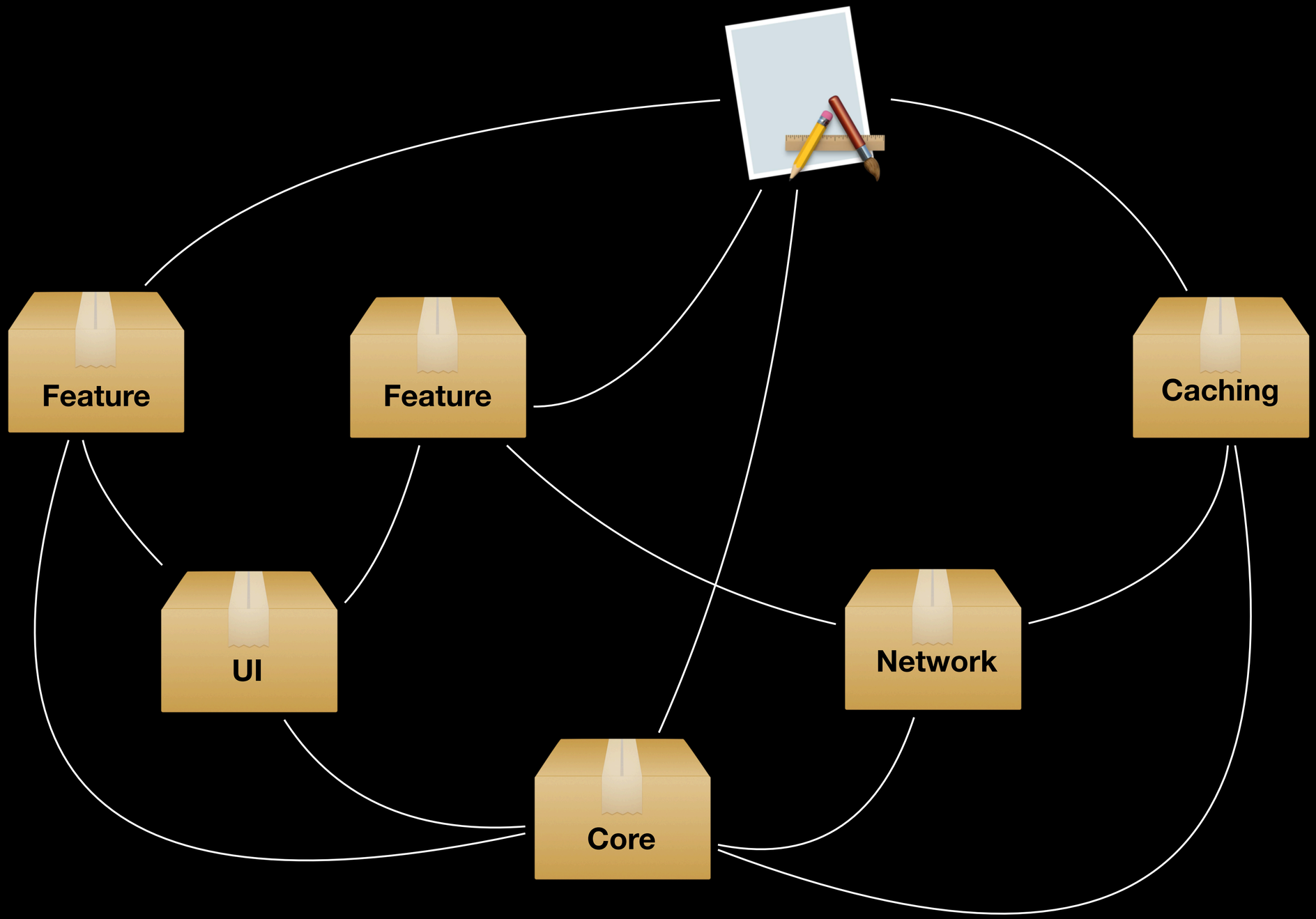
(Optimistically)

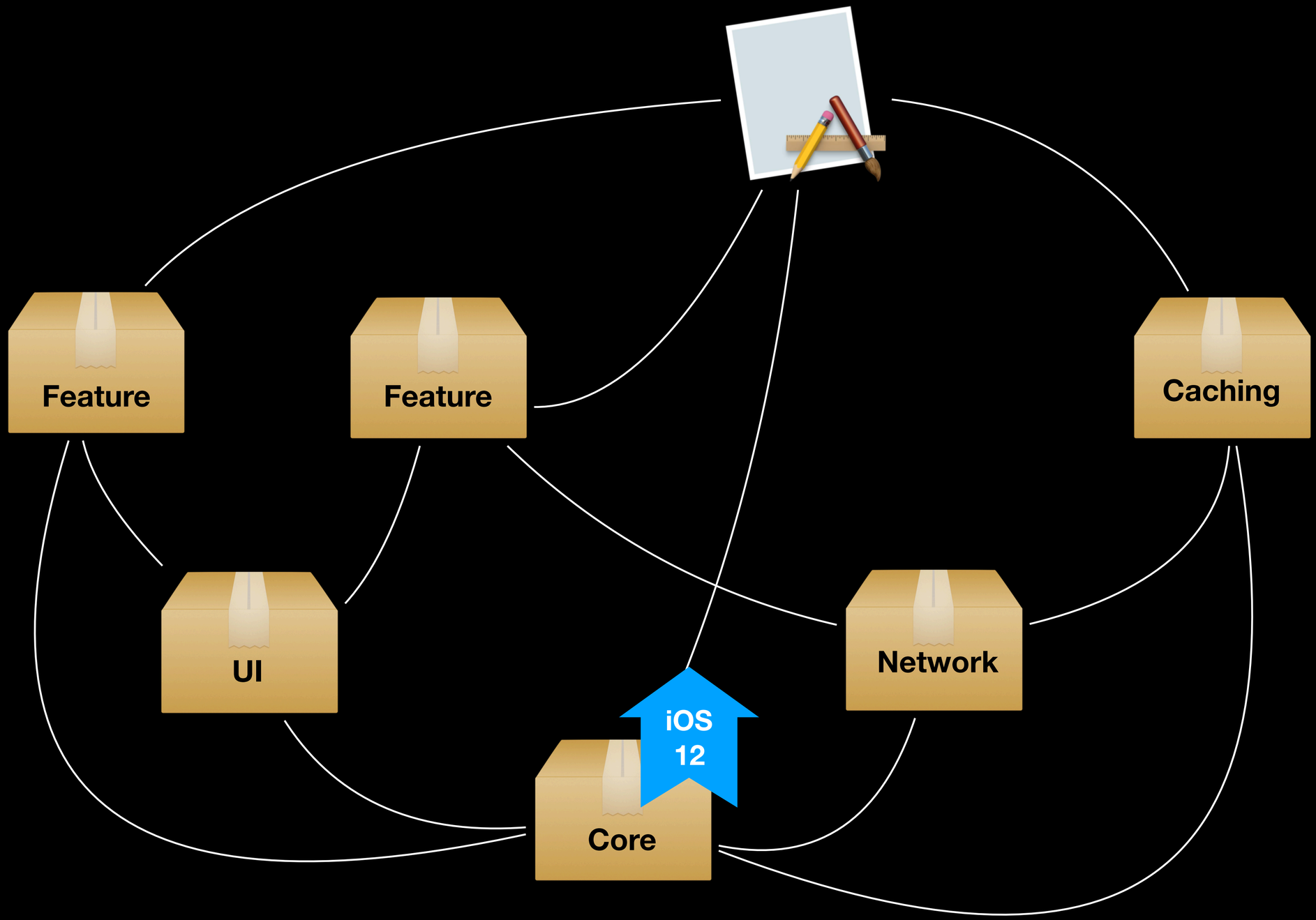
Major releases are semantics

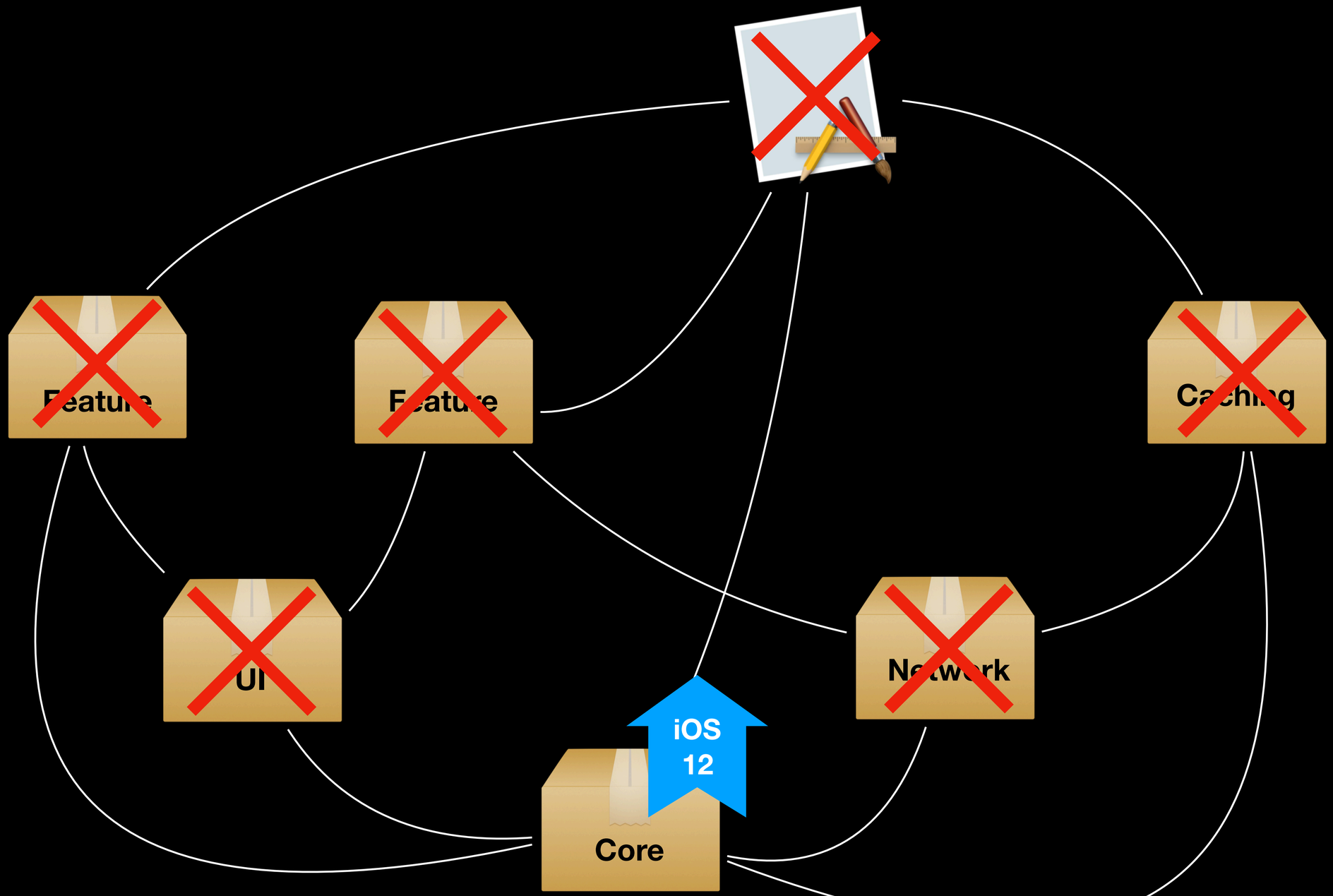
~~Major releases are semantics~~

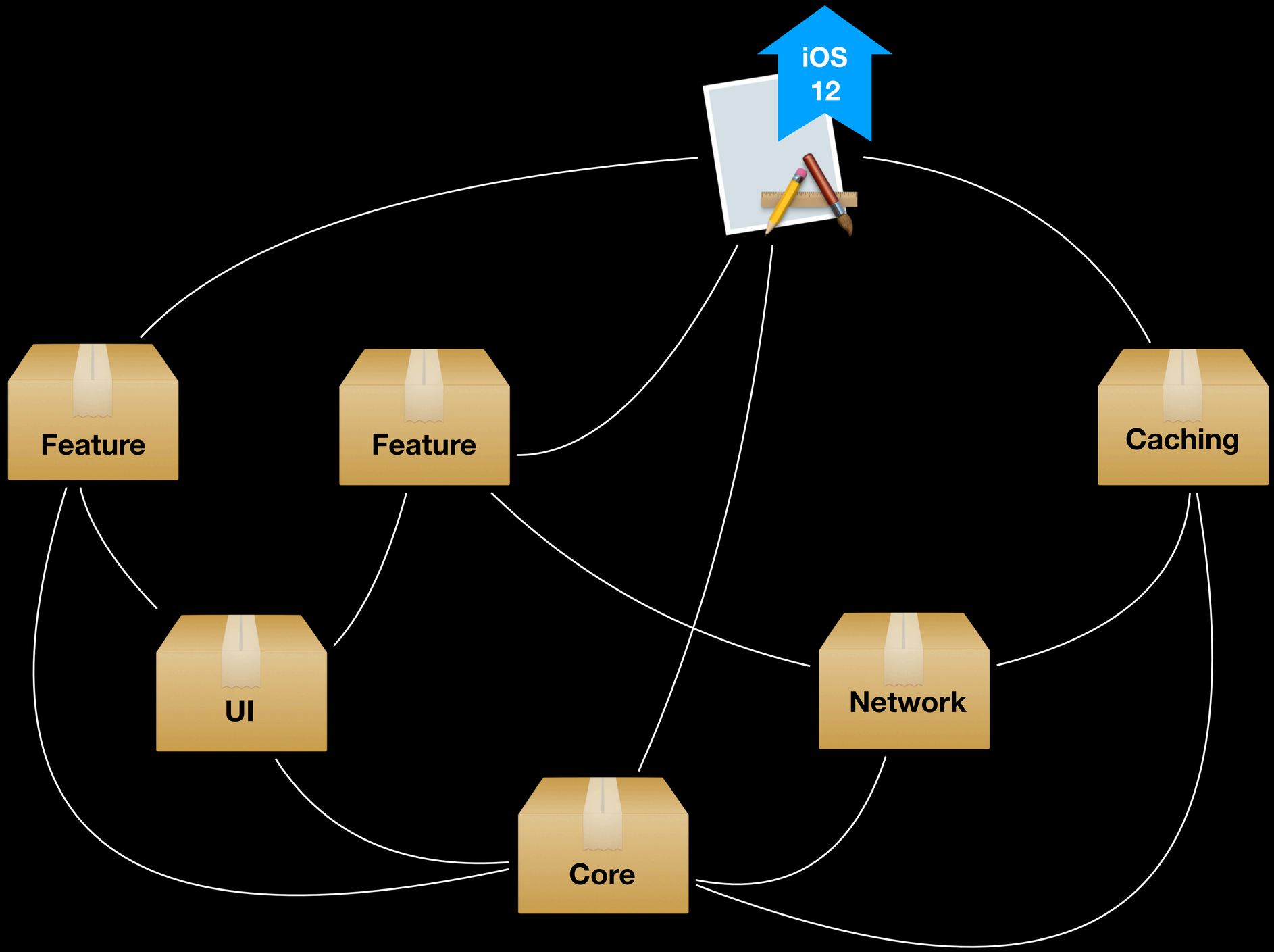
Major releases are an organizational challenge

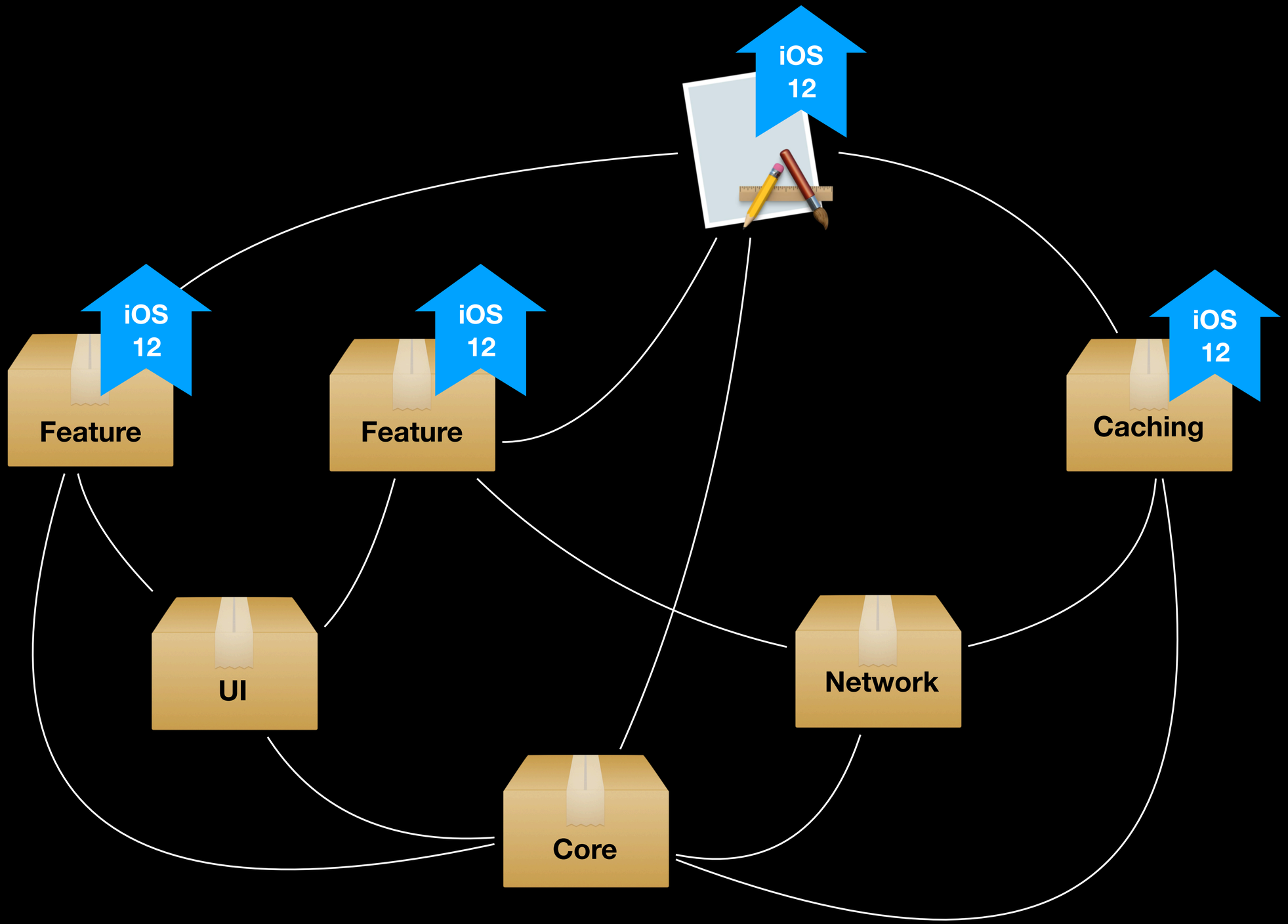
Avoiding majors

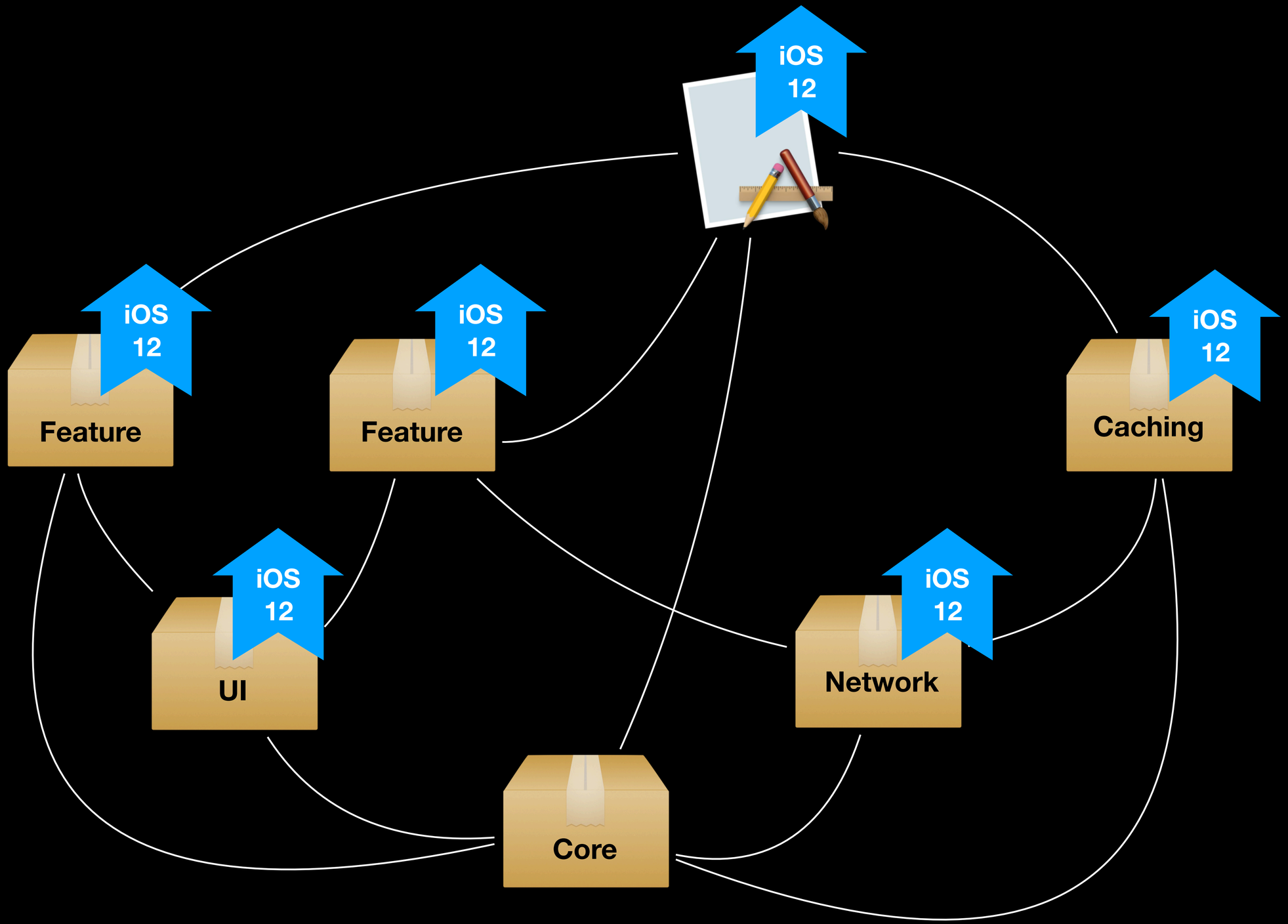

Feature
Feature
Caching
UI
Network
Core

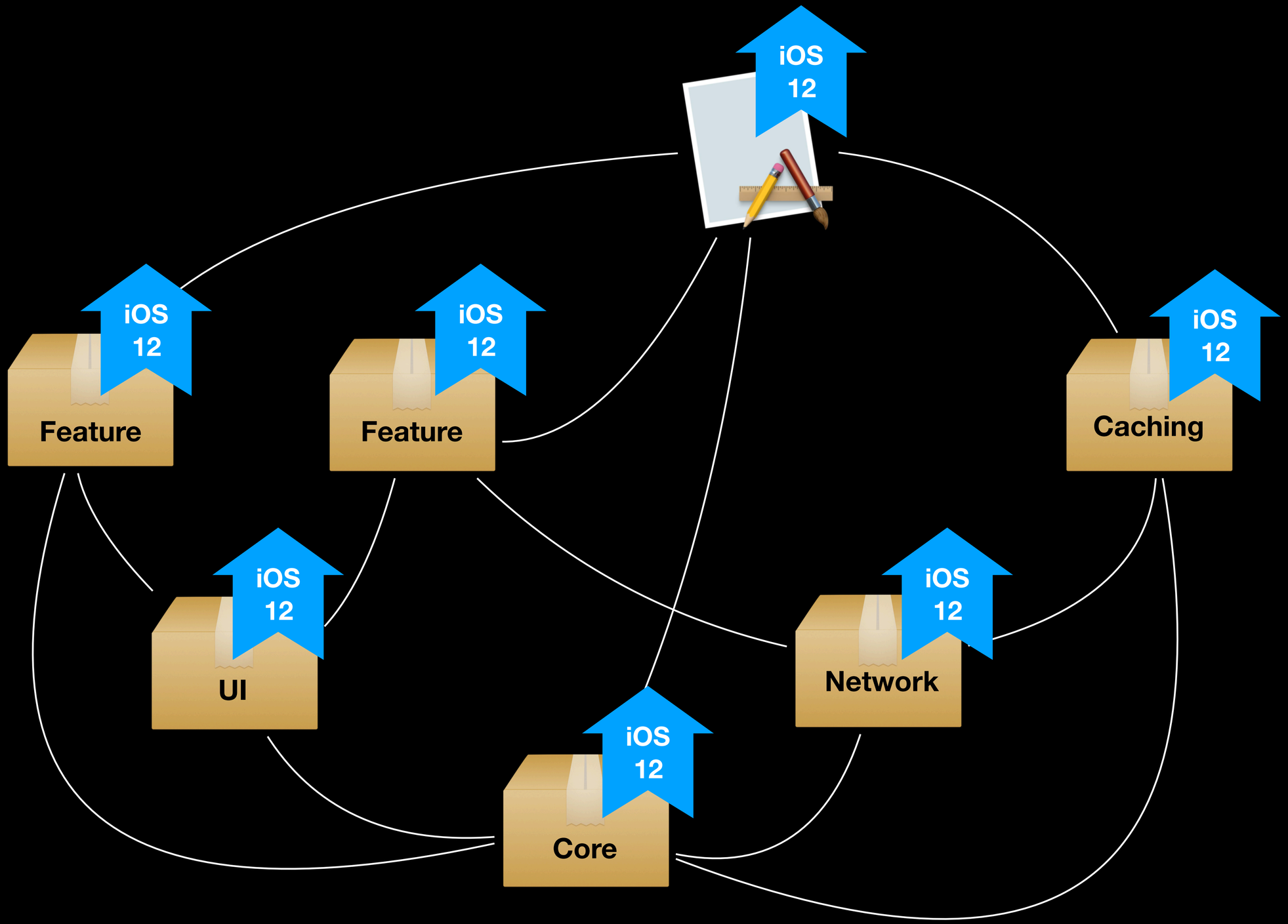

Feature
Feature
Caching
UI
Network
iOS
12
Core


Feature
Feature
Caching
UI
Network
iOS
12
Core


iOS
12
Feature
Feature
Caching
UI
Network
Core


iOS 12
iOS 12
Feature
iOS 12
Feature
iOS 12
Caching
UI
Network
Core


iOS 12
iOS 12
Feature
iOS 12
Feature
iOS 12
Caching
iOS 12
UI
iOS 12
Network
Core


iOS 12
Feature
iOS 12
Feature
iOS 12
Caching
iOS 12
UI
iOS 12
Network
iOS 12
Core

```
public enum Cheese {
    case brie
    case gouda
    case camembert
}
```

```
public enum Cheese {
    case brie
    case gouda
    case camembert
    case cheddar
}
```

```
func eatCheese(_ cheese: Cheese) {
    switch cheese {
    case .brie: print("Yummy brie")
    case .gouda: print("Loving me some gouda!")
    case .camembert: print("Tasty camembert")
    }
}
```

```
func eatCheese(_ cheese: Cheese) {
    switch cheese {
    case .brie: print("Yummy brie")
    case .gouda: print("Loving me some gouda!")
    case .camembert: print("Tasty camembert")
    @unknown default: print("I'll eat anything")
    }
}
```

```
func eatCheese(_ cheese: Cheese) {
    switch cheese {
    case .brie: print("Yummy brie")
    case .gouda: print("Loving me some gouda!")
    case .camembert: print("Tasty camembert”)
    @unknown default: print("I'll eat anything")
    }
}
```

Switch must be exhaustive

```
func eatCheese(_ cheese: Cheese) {
    switch cheese {
    case .brie: print("Yummy brie")
    case .gouda: print("Loving me some gouda!")
    case .camembert: print("Tasty camembert”)
    case .cheddar: print("Mmmmm cheddar”)
    @unknown default: print("I'll eat anything")
   }
}
```

```
open class SheetViewController: UIViewController {
    public var containerView: UIView

    public override func viewDidLoad() {
        super.viewDidLoad()

        ...
    }

    ...
}
```

```
open class SheetViewController: UIViewController {
    public var containerView: UIView

    public override func viewDidLoad() {
        super.viewDidLoad()
        containerView.translatesAutoresizingMaskIntoConstraints = false
        …
    }

    …
}
```

syntax highlighting

Better example? Color change can still be major

```
public class SheetViewController: UIViewController {
    private var containerView: UIView

    public override func viewDidLoad() {
        super.viewDidLoad()
        containerView.translatesAutoresizingMaskIntoConstra
        …
    }

    …
}
```

Better example? Color change can still be major

Start with a beta

0.2.3

Deprecations

```
public func fetchWorkouts(_ user: User) -> [Workout]
```

~~`public func fetchWorkouts(_ user: User) -> [Workout]`~~

```
public func fetchWorkouts<I: Identifiable>(_ id: I) -> [Workout]
```

```
@available(*, deprecated, renamed: "fetchWorkouts(id:)")
public func fetchWorkouts(_ user: User) -> [Workout]

public func fetchWorkouts<I: Identifiable>(_ id: I) -> [Workout]
```

Maintain deprecated code to ease migrations

Escape hatch

```
public func fetchArticles(_ onComplete: @escaping (Result<[Article], Error>) -> Void)

public func fetchUsers(_ onComplete: @escaping (Result<[User], Error>) -> Void)

public func fetchComments(article: Article, onComplete: @escaping (Result<[Comment], Error>) -> Void)
```

Escape hatch

```
public func fetchArticles(_ onComplete: @escaping (Result<[Article], Error>) -> Void)

public func fetchUsers(_ onComplete: @escaping (Result<[User], Error>) -> Void)

public func fetchComments(article: Article, onComplete: @escaping (Result<[Comment], Error>) -> Void)

public func fetchData(_ request: URLRequest, onComplete: @escaping (Result<Data, Error>) -> Void)
```

Secret majors

TODO Add description of w
people are

Subtle changes

```
public func eraseAllData(_ removeBackUp: Bool = false) {
  ...
}
```

Subtle changes

```
public func eraseAllData(_ removeBackUp: Bool = false) {
  ...
}
```

```
public func eraseAllData(_ removeBackUp: Bool = true) {
  ...
}
```

Protocols

```
public protocol MapType {
    var coordinates: CLLocation { get }
}
```

Protocols

```
public protocol MapType {
    var coordinates: CLLocation { get }
    var elevations: [Int] { get }
}
```

Protocols

```
public protocol MapType {
    var coordinates: CLLocation { get }
    var elevations: [Int] { get }
}

extension MapType {
    var elevations: [Int] { return [] }
}
```

Protocols

```
public protocol MapType {
    var coordinates: CLLocation { get }
    var elevations: [Int] { get }
}

extension MapType {
    var elevations: [Int] { return [] }
}
```

Protocols

```
public protocol MapType {
    var coordinates: CLLocation { get }
    var altitudes: [Int] { get }
}

extension MapType {
    var elevations: [Int] { return [] }
}
```

Optionals

```
struct User {
    let name: String
    let birthDate: Date
}
```

Optionals

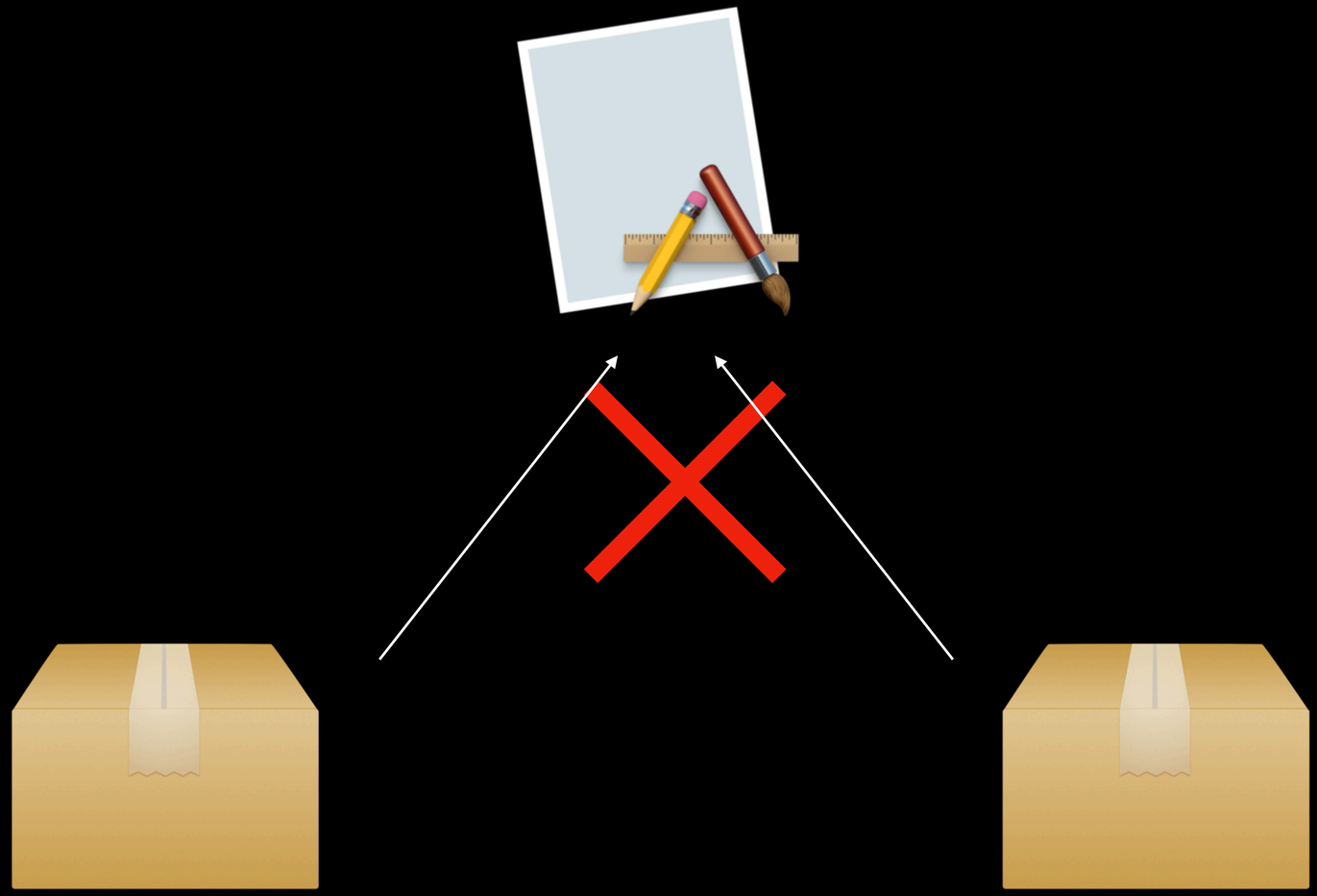
```
struct User {
    let name: String
    let birthDate: Date?
}
```

Xcode / iOS specific code

```
if #available(iOS 13, *) {
    viewController.isModalInPresentation = false
}
```

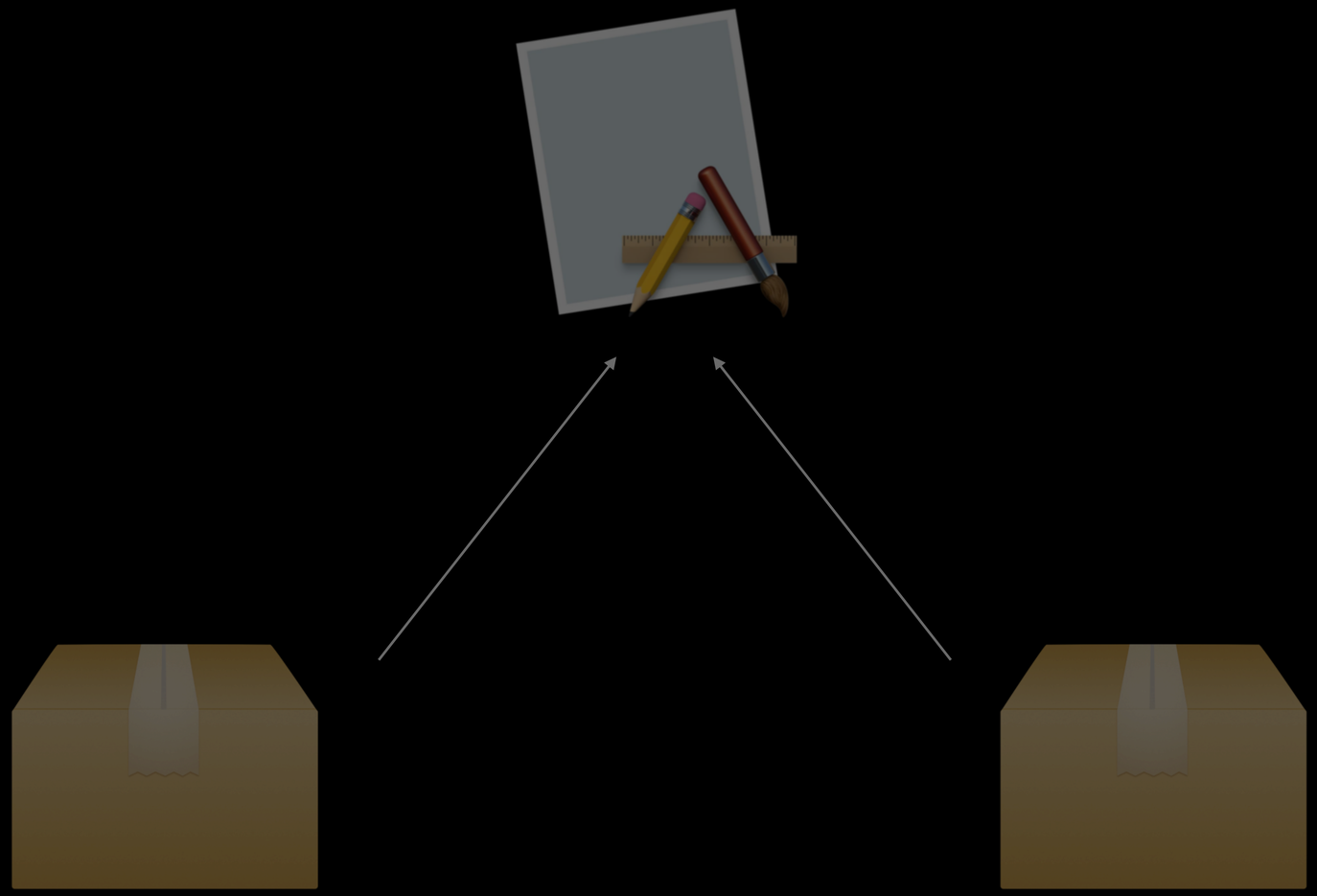
Xcode / iOS specific code

```
if #available(iOS 13, *) {
    #if swift(>=5.1)
    viewController.isModalInPresentation = false
    #endif
}
```



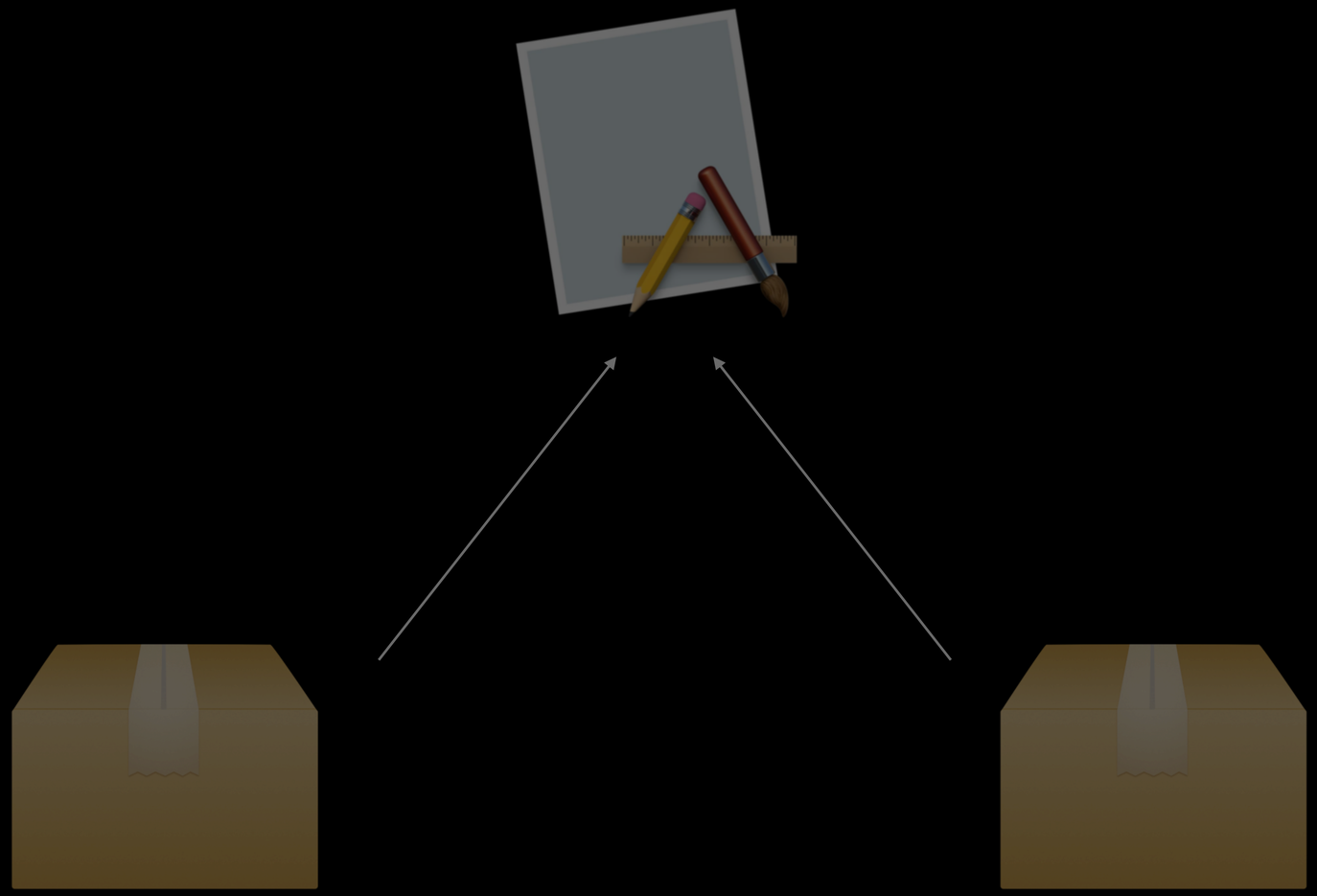
```
public extension UIViewController {
  func trackPage(name: String) {
    …
  }
}
```

```
public extension UIViewController {
  func trackPage(name: String) {
    …
  }
}
```



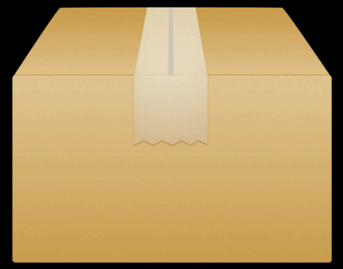
```
public extension UIViewController {
  func trackPage(name: String) {
    …
  }
}
```

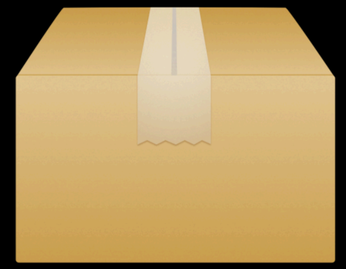
```
public extension UIViewController {
  func trackPage(name: String) {
    …
  }
}
```



```
private extension UIViewController {
  func trackPage(name: String) {
    …
  }
}
```

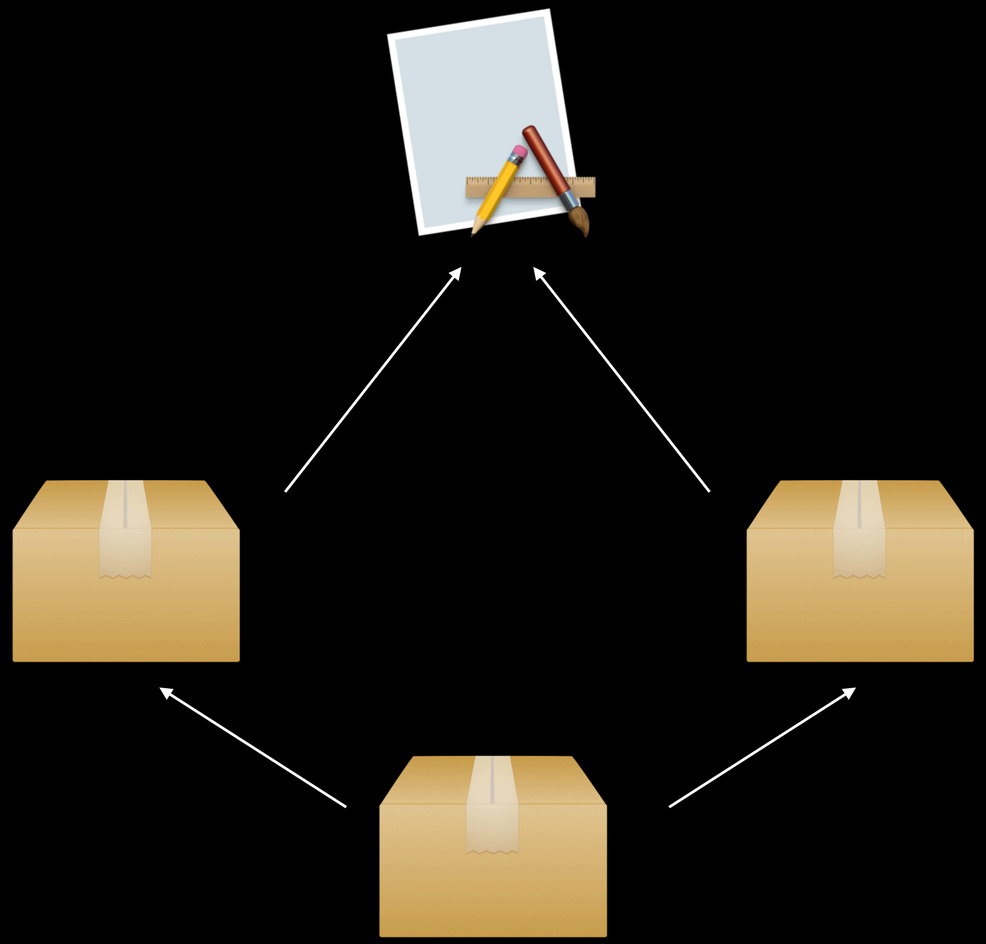
```
internal extension UIViewController {
  func trackPage(name: String) {
    …
  }
}
```





```
public extension UIViewController {
  func trackPage(name: String) {
    …
  }
}
```

```
public extension UIViewController {
  func trackPage(name: String) {
    …
  }
}
```



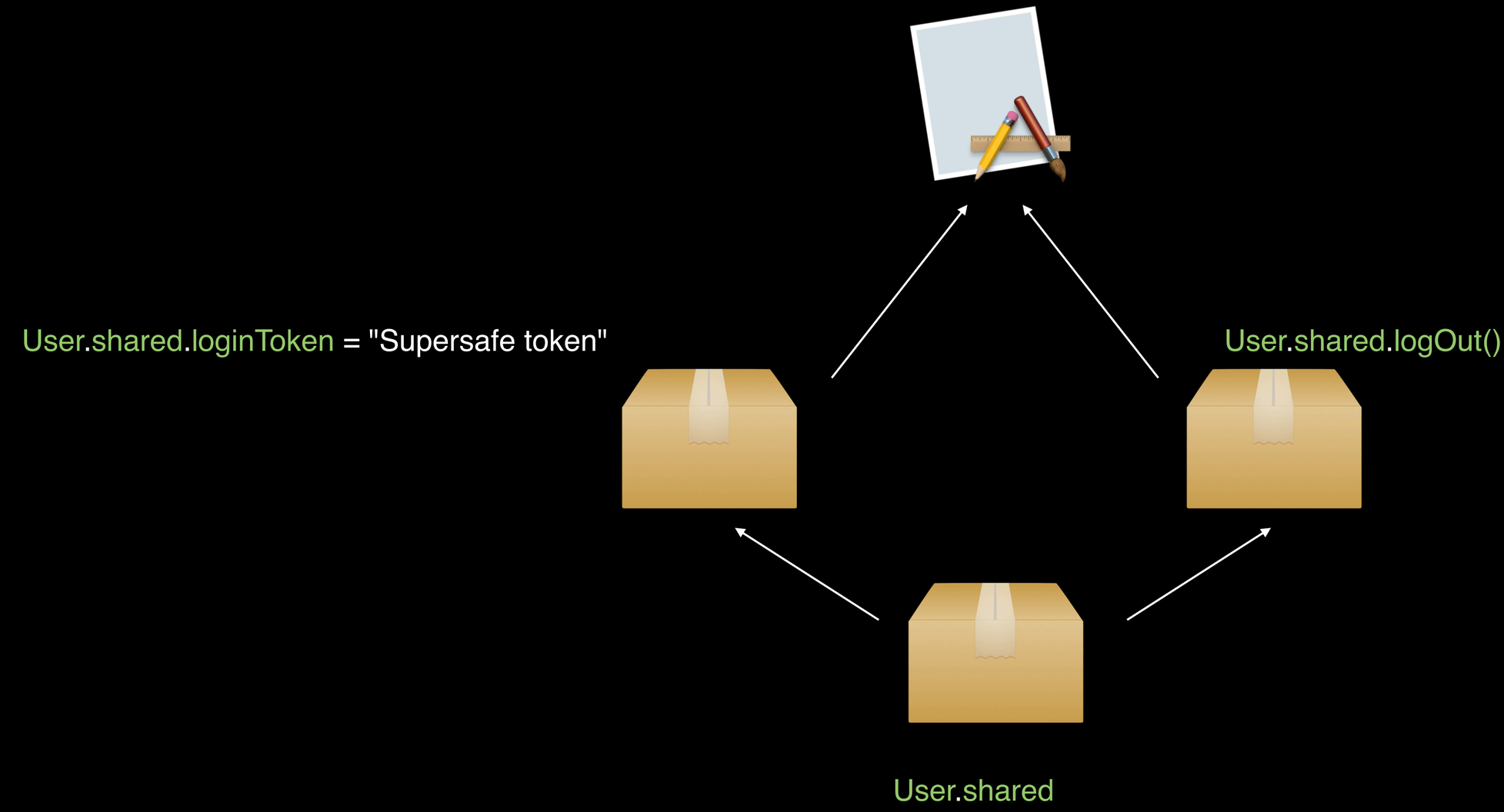
```
public extension UIViewController {
  func trackPage(name: String) {
    …
  }
}
```

```
public extension UIViewController {
  func trackPage(name: String) {
    …
  }
}
```

```
public protocol Analytics {
  func trackPage(name: String)
}

extension Analytics where Self: UIViewController {
  func trackPage(name: String) {
    …
  }
}
```

```
public protocol Analytics {
  func trackPage(name: String)
}

extension Analytics where Self: UIViewController {
  func trackPage(name: String) {
     …
  }
}

final class ArticlesViewController: Analytics {
  func viewDidAppear(_ animated: Bool) {
     super.viewDidAppear(animated)
     self.trackPage(name: “articles")
   }
  …
}
```

Singletons


User.shared.loginToken = "Supersafe token"
User.shared.logOut()
User.shared

“Can others update and compile without changing code?”

“Can others update and compile without changing code?”

No? Major
Yes? Minor or Patch
(probably)

Making majors less impactful

- Avoid majors in the first place (if possible)
- Plan the release with coworkers
- Do the work for others
- Write a migration guide
- Release minor / patch changes before a major
- Test your updated version before making a release

~~Making majors less impactful~~

Show empathy

Remote modules

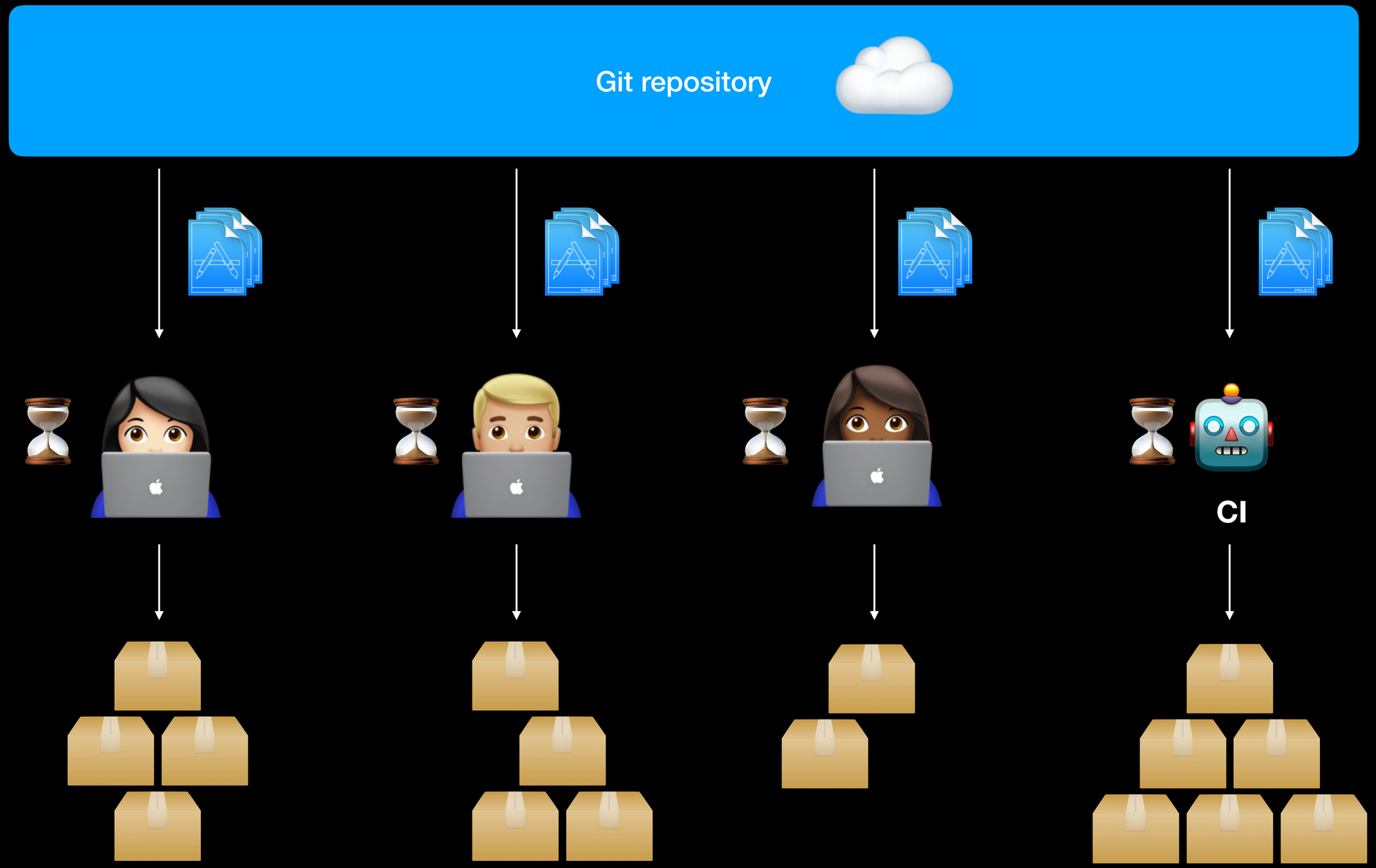
Pros

- ✓ Support multiple workspaces
- ✓ Granular control with versioning

Cons

- – Complexity rises
- – More ceremony
- – Difficult to version correctly

Package managers


Git repository
CI

apple.com

Tilpas din 16" MacBook Pro – space grey

2,3 GHz 8-core Intel Core i9-processor (9. generation), Turbo Boost op til 4,8 GHz

16 GB 2666 MHz DDR4-hukommelse

AMD Radeon Pro 5500M med 4 GB GDDR6-hukommelse

SSD-lager på 1 TB

16" Retina-skærm med True Tone

Fire Thunderbolt 3-porte

Touch Bar og Touch ID

Baggrundsbelyst tastatur – dansk

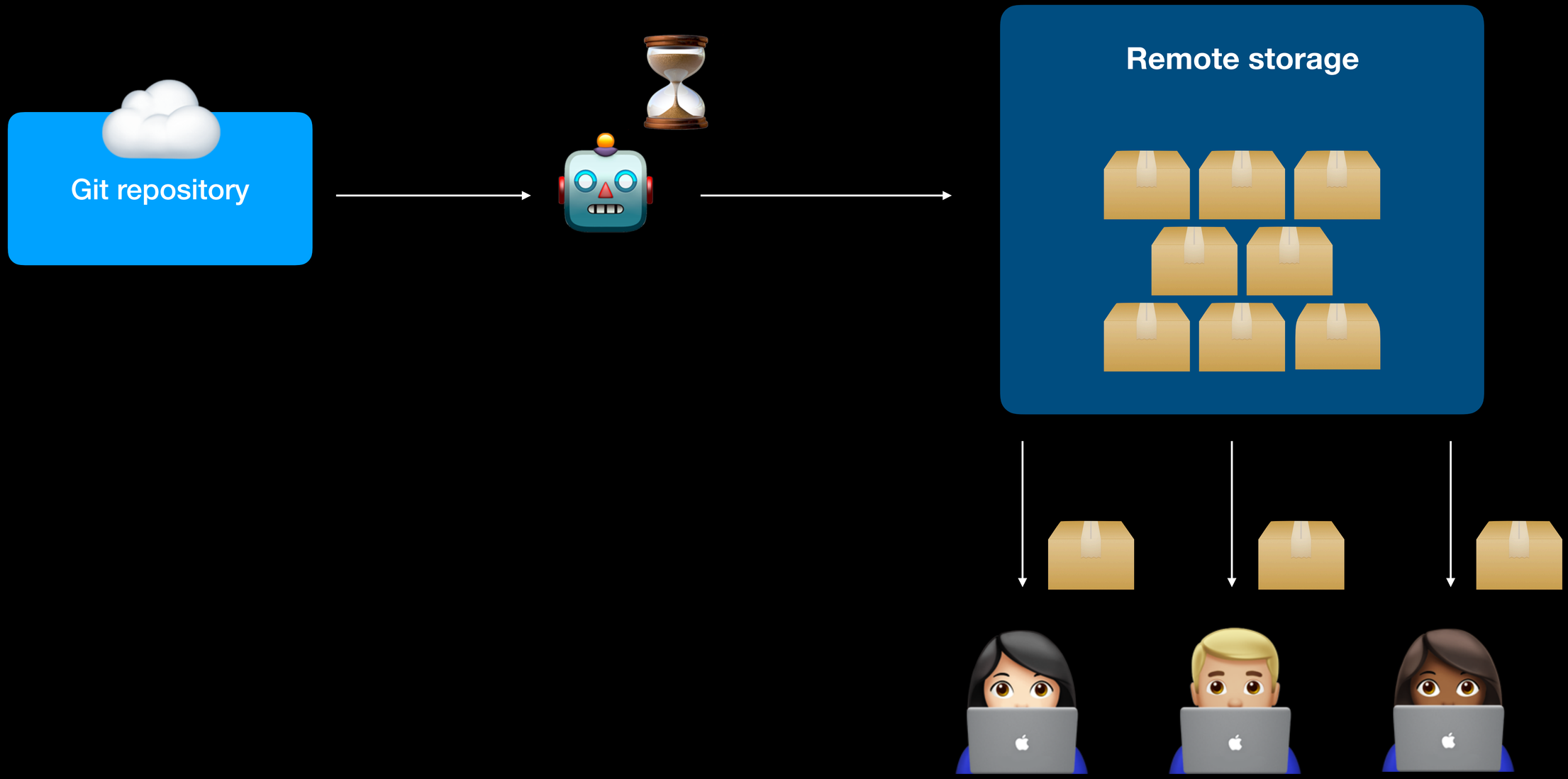
Afsendelse:
5-7 hverdage
Gratis levering
Se leveringsdatoer

24.999,00 kr.

Læg i Shoppingpose

First enterprise requirement

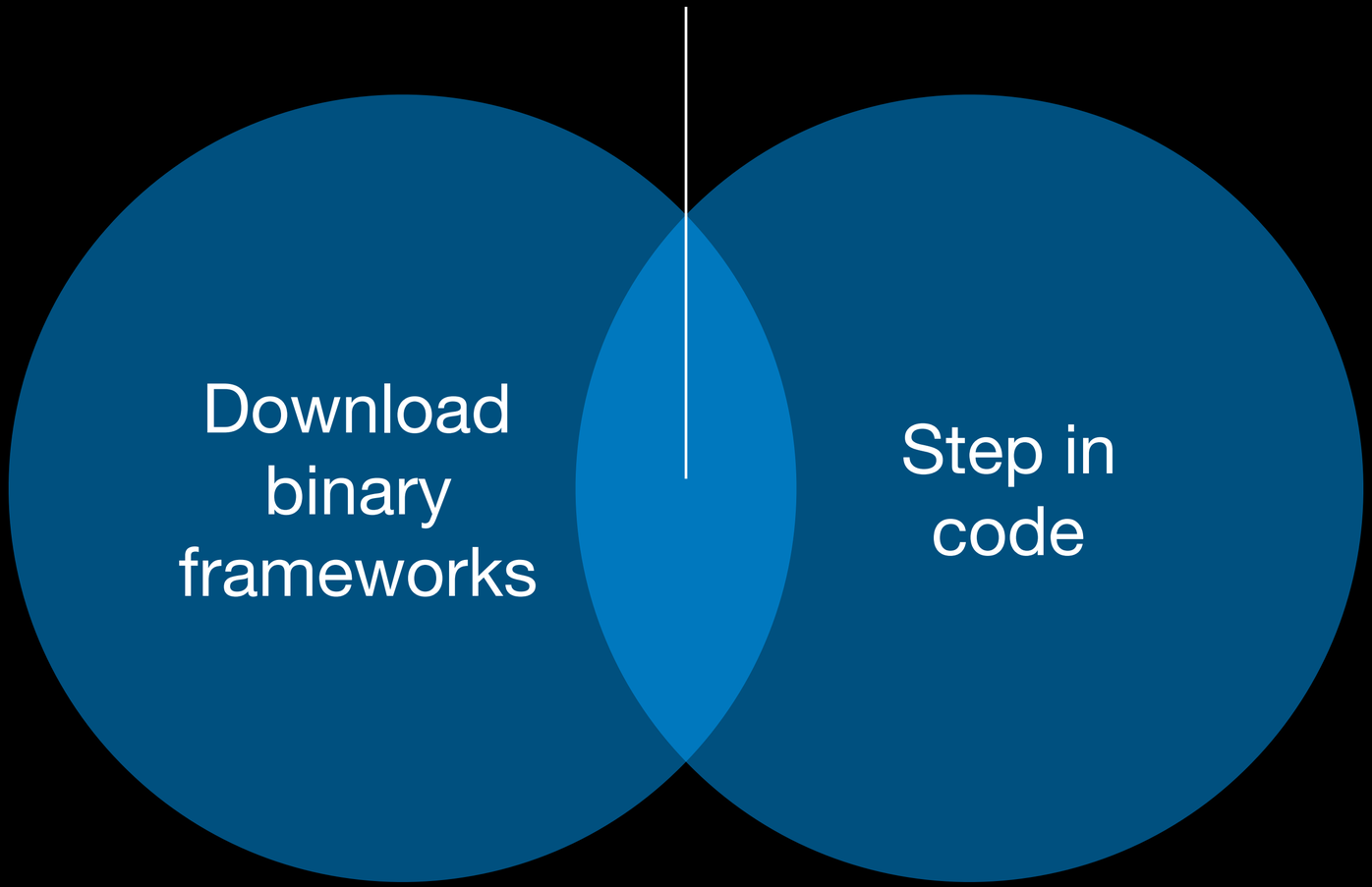
Binary support


Git repository
Remote storage

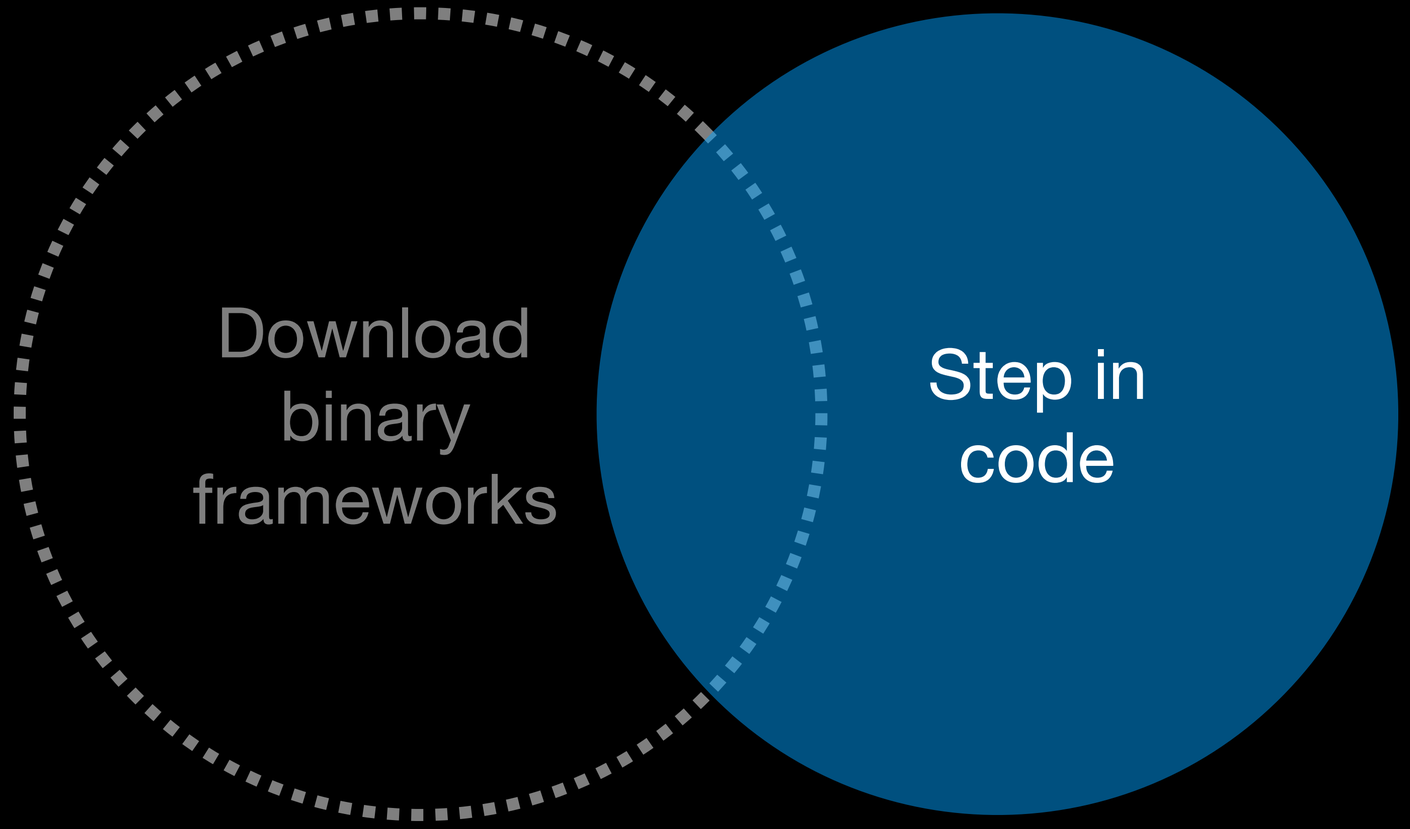
Second requirement

Step-in code

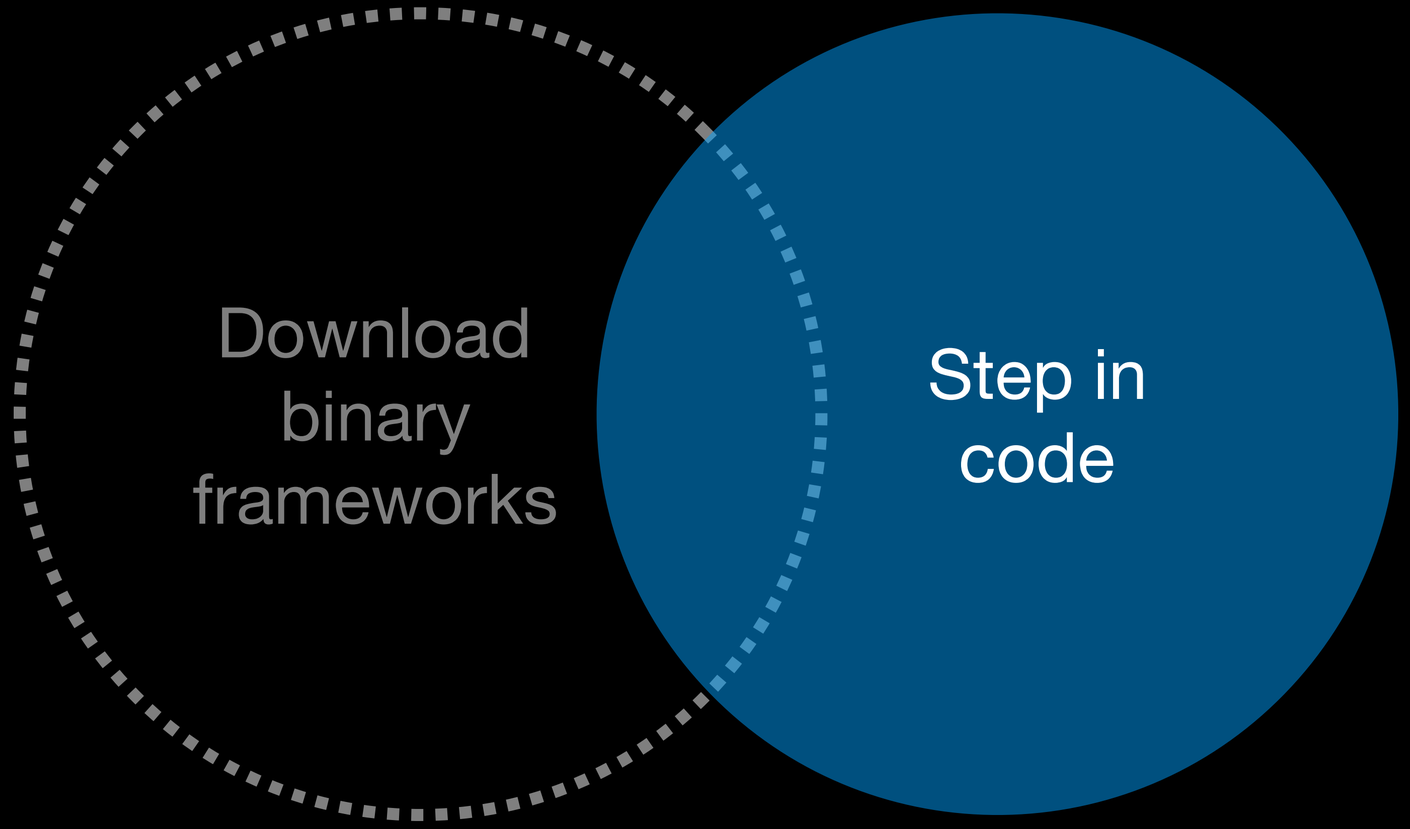
The sweet spot



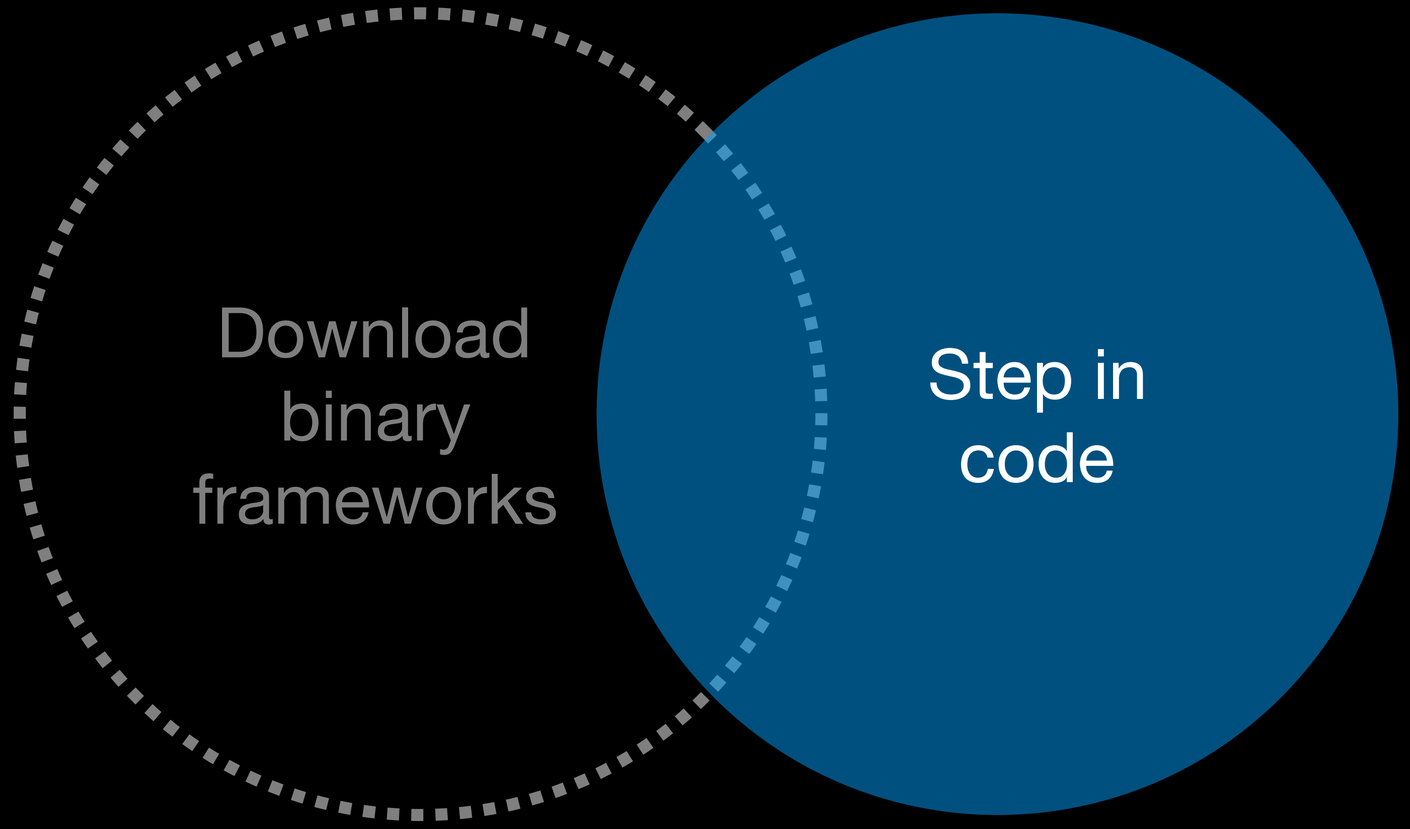
Swift PM



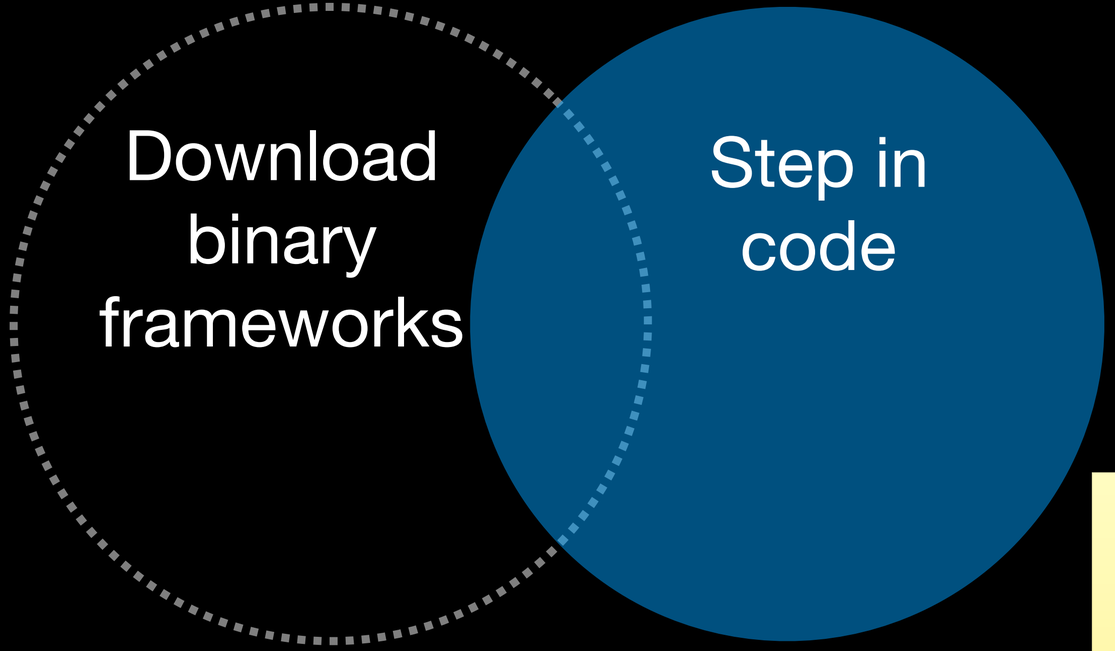
CocoaPods



CocoaPods+binary

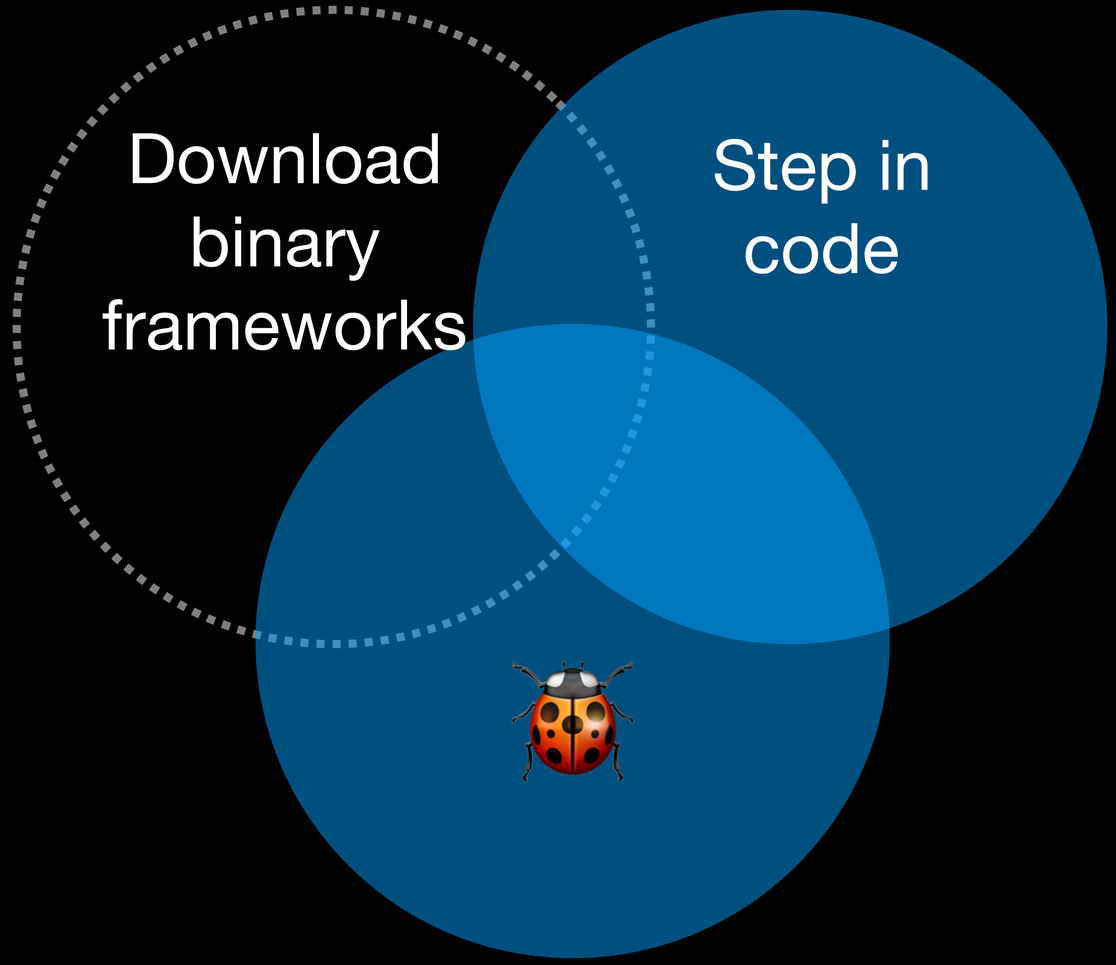


Carthage

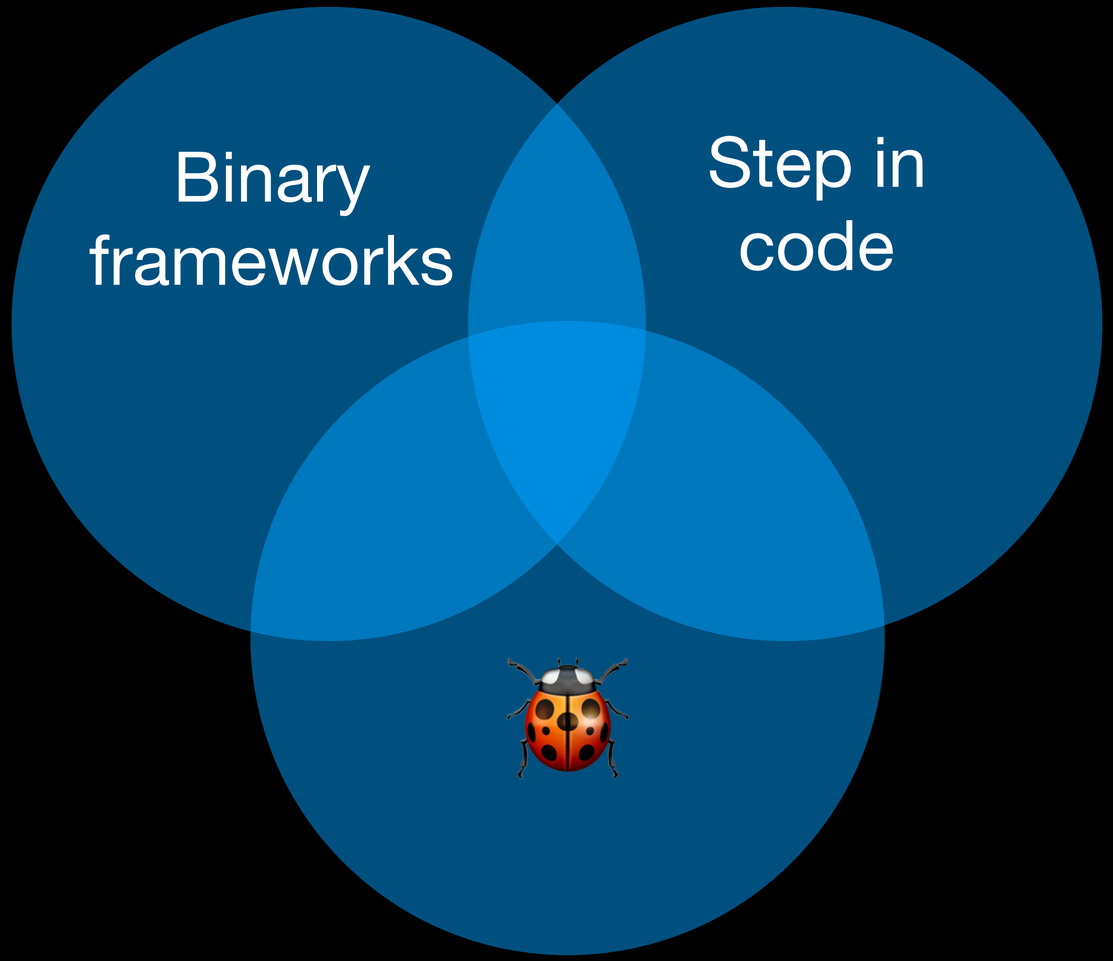


So carthage builds are not self hosted

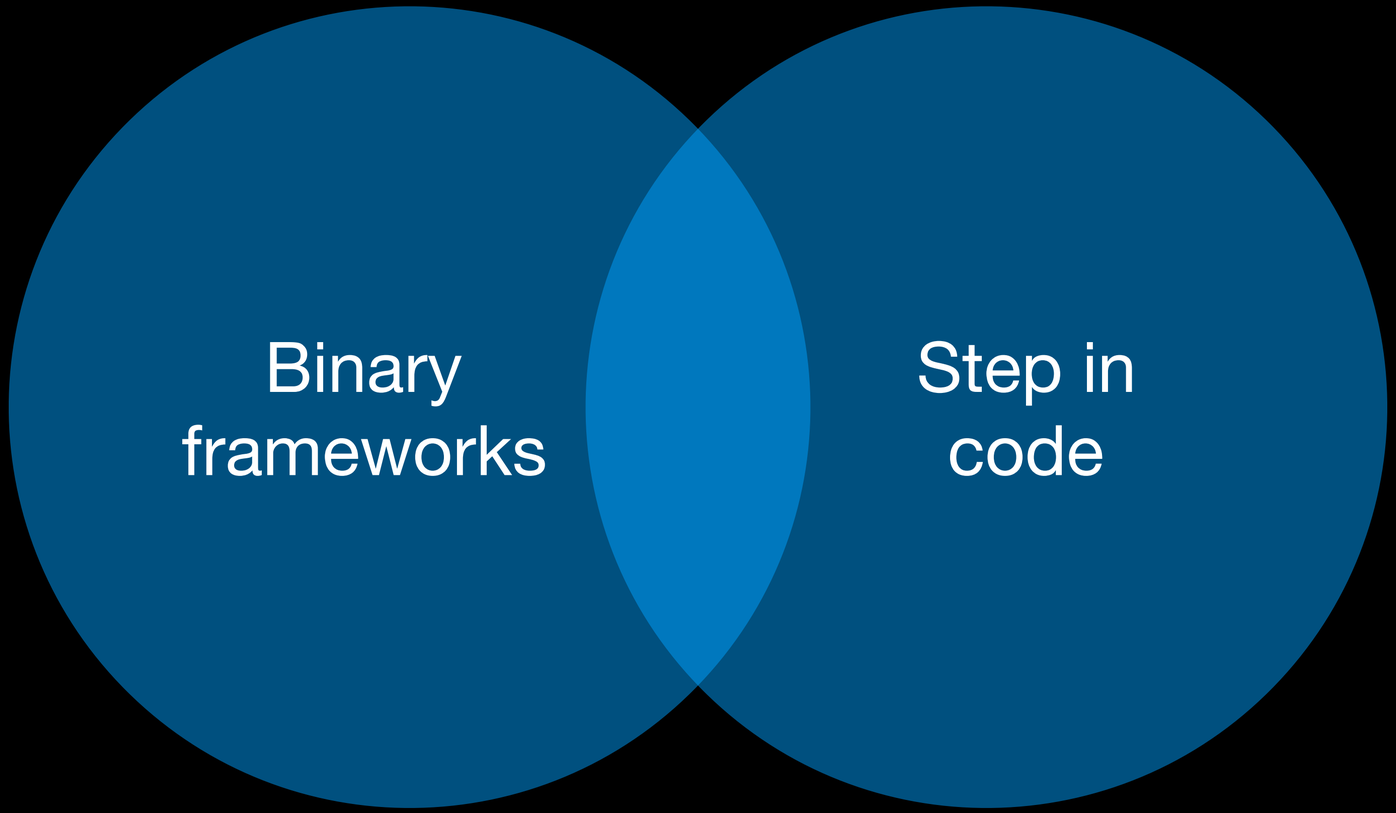
Carthage



Carthage+Rome



Carthage++

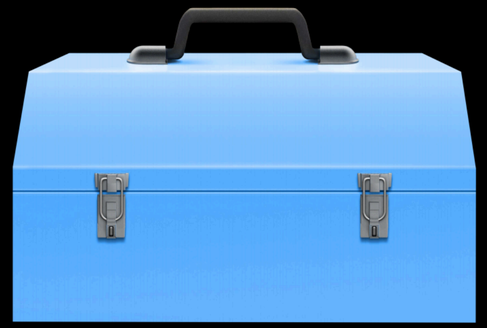


Carthage++

Make link unclickable

soperations/Carthage

XCFrameworks



What we covered

- Deciding between local and remote modules
- Dependency management is complicated
- Semantic versioning isn’t just semantics
- Dealing with major releases
- Keeping projects stable
- Package managers

@tjeerdintveen

tjeerd@swiftindepth.com

Make link unclickable

40% discount code on everything

wgotocph19

https://www.manning.com/books/swift-in-depth

goto; copenhagen

GOTO Copenhagen 2019

Conference Nov. 18 - 20

Please

Remember to rate this session

Thank you!

Follow us @gotocph

goto;
copenhagen

Follow us @gotocph