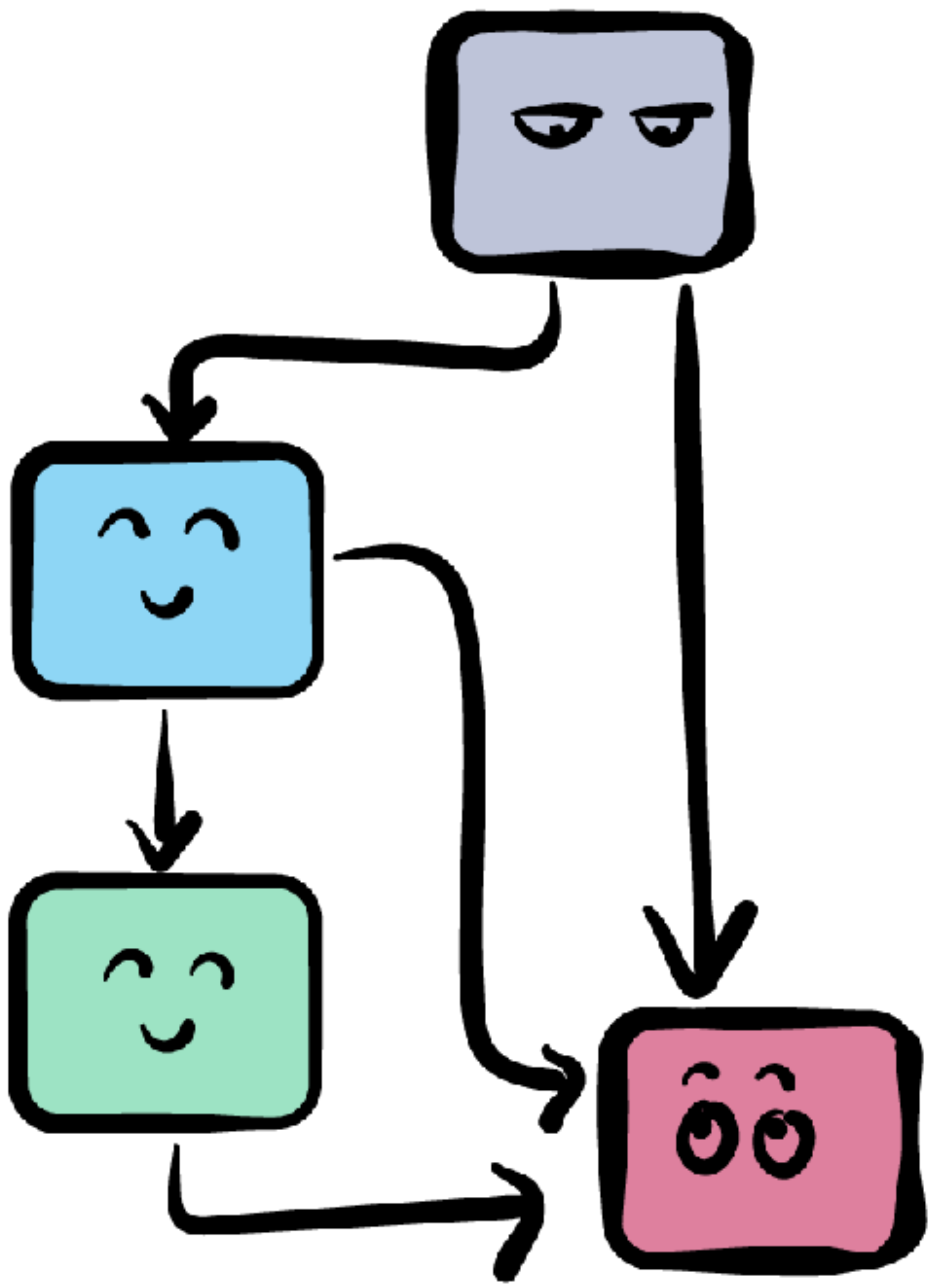
iOS Concurrency with GCD & Operations

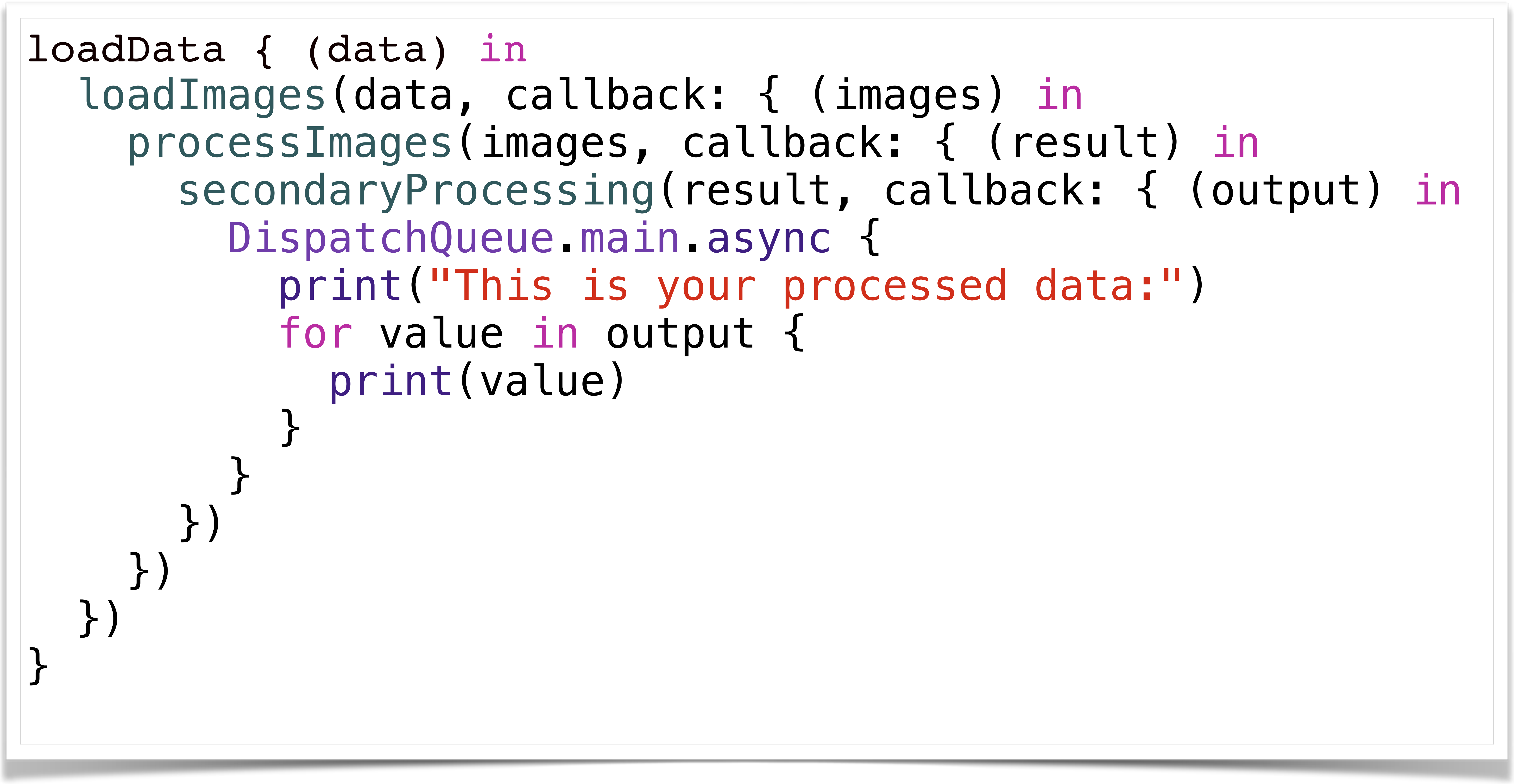


Part 3: Simple Use Cases

Already Asynchronous: no problem!

```
let downloadSession = URLSession(configuration: .ephemeral)
for url in urls {
  let _ = downloadSession.dataTask(with: url) { data, _, _ in
    self.image = UIImage(data: data!)
    DispatchQueue.main.async {
      // update UI with image
    }
  }
}
```

Already Asynchronous: no problem!



```swift
loadData { (data) in
  loadImages(data, callback: { (images) in
    processImages(images, callback: { (result) in
      secondaryProcessing(result, callback: { (output) in
        DispatchQueue.main.async {
          print("This is your processed data:")
          for value in output {
            print(value)
          }
        }
      })
    })
  })
}
```

MAKE SYNCHRONOUS ASYNCHRONOUS

```swift
open func dataTask(with url: URL, completionHandler: @escaping
  (Data?, URLResponse?, Error?) -> Swift.Void) -> URLSessionDataTask
```

```swift
public func asyncSyncFunction(_ input: inputType, runQueue: DispatchQueue,
  completionQueue: DispatchQueue,
  completion:@escaping (resultType, Error) -> ()) {
  runQueue.async {
    let result = syncFunction(input)
    completionQueue.async {
      completion(result, error)
    }
  }
}
```

Challenge Time!