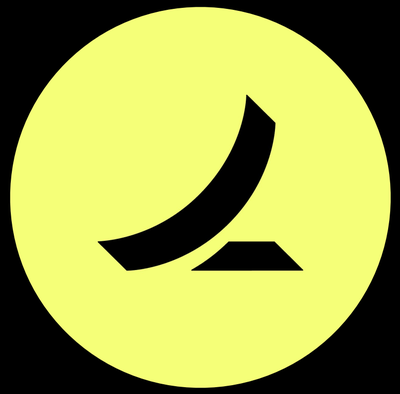
Beyond Xcode

Batteries Not Included

hats 🎩

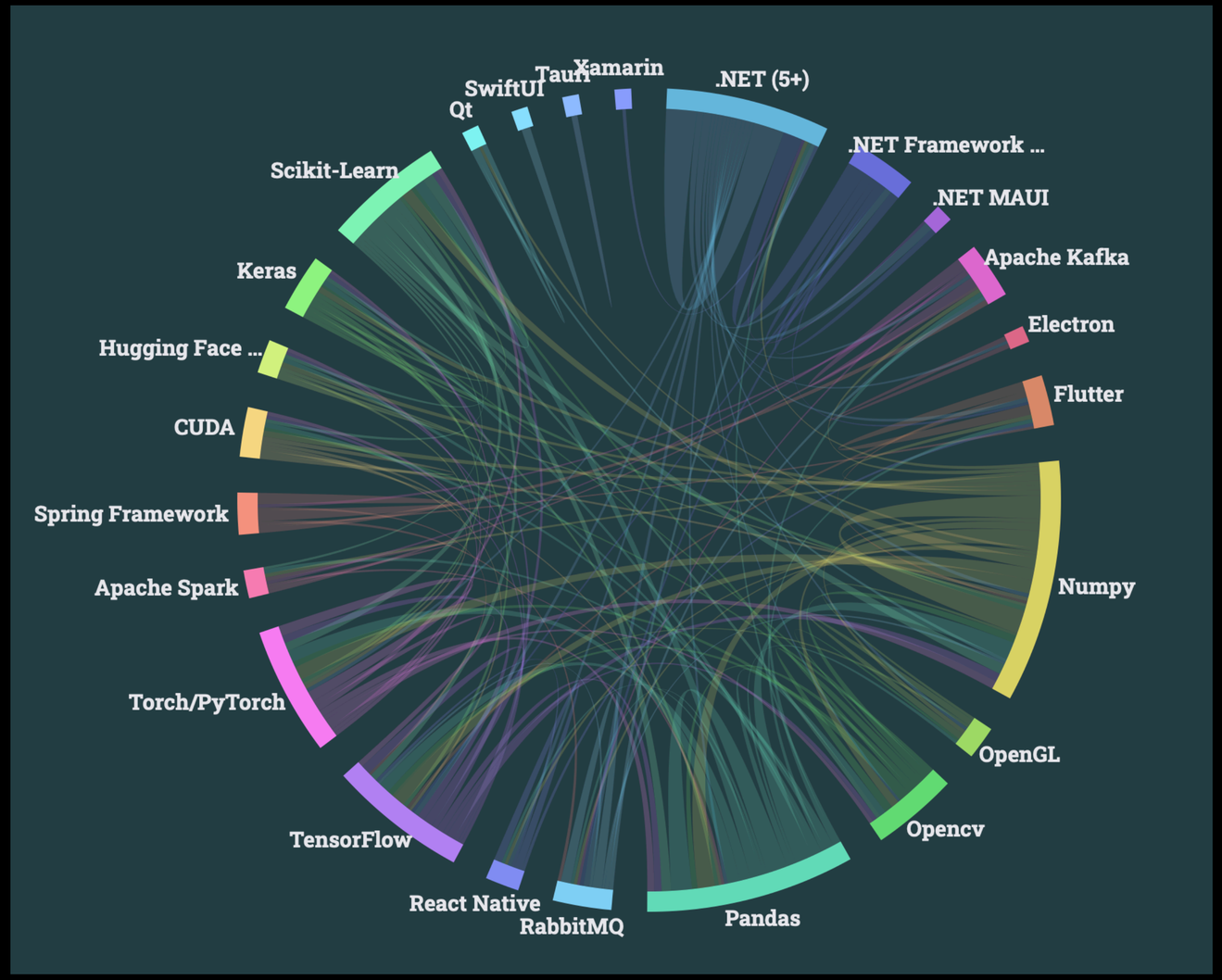
Computer Science
University of Waterloo



iOS Developer
Ramp

Maintainer
Theos

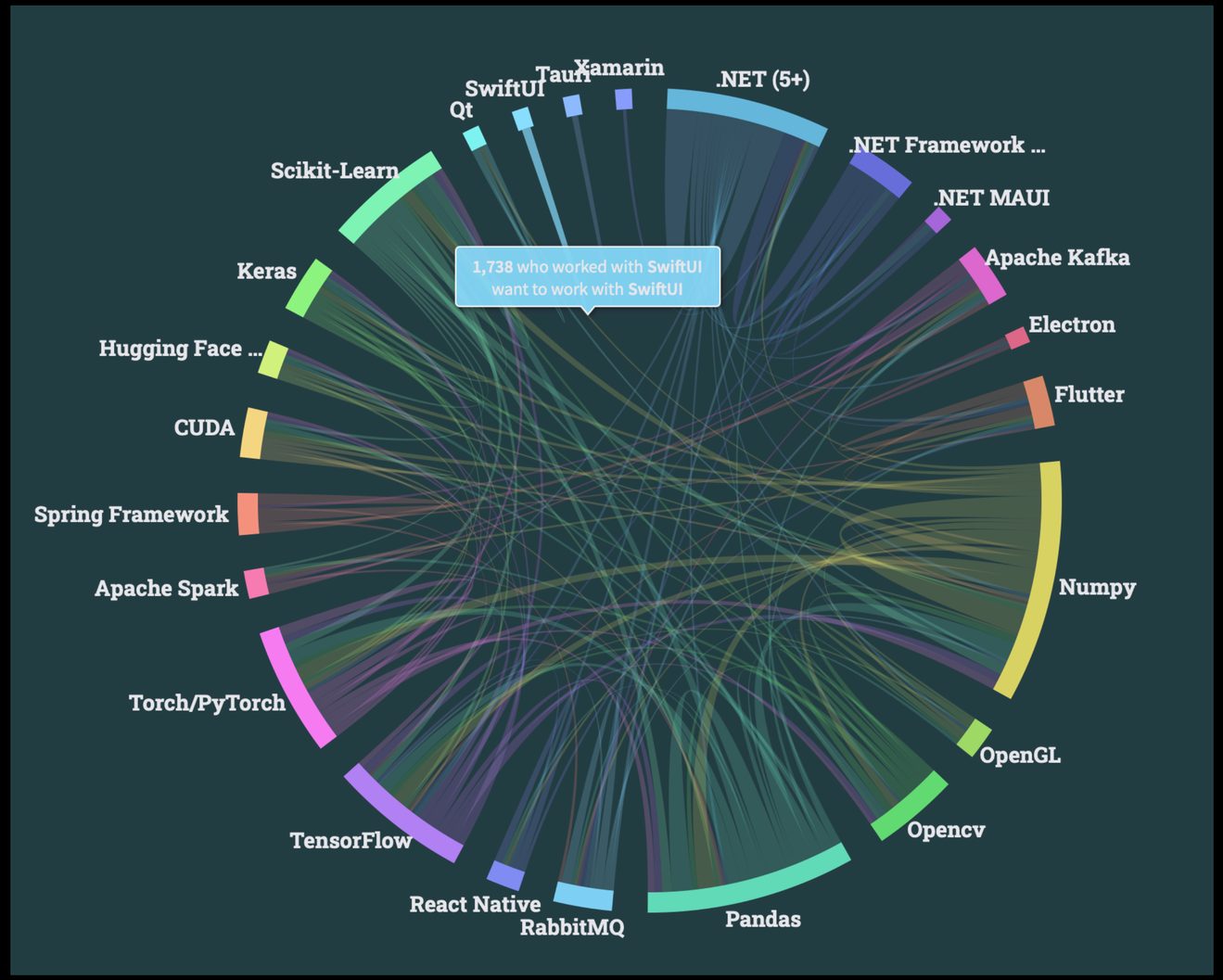
we ❤️ swift(ui)



Which [non-web] ***frameworks and libraries*** *have you done extensive development work in over the past year, and which do you want to work in over the next year?*

(StackOverflow 2023 Developer Survey)

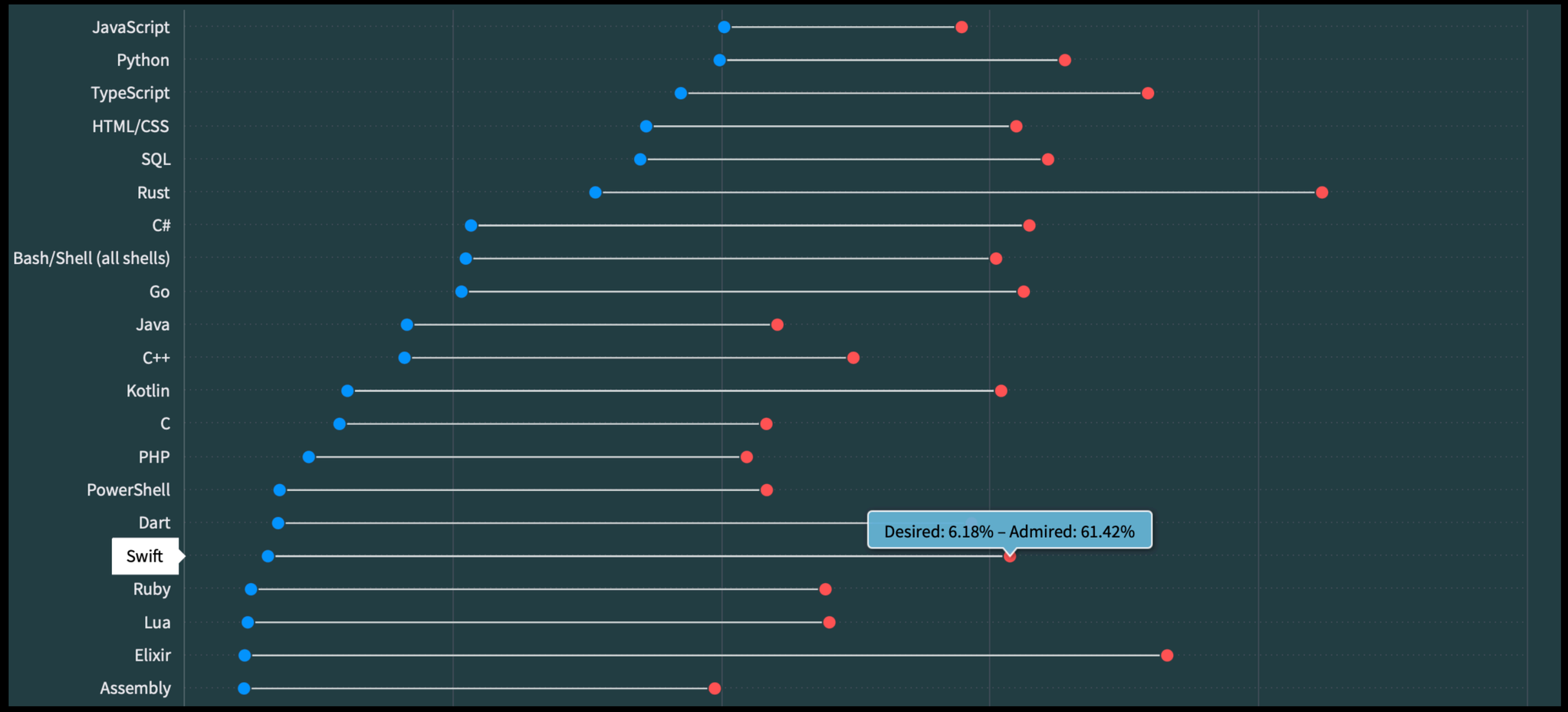
we ❤️ swift(ui)



*Which [non-web] **frameworks and libraries** have you done extensive development work in over the past year, and which do you want to work in over the next year?*

(StackOverflow 2023 Developer Survey)

…but we're siloed



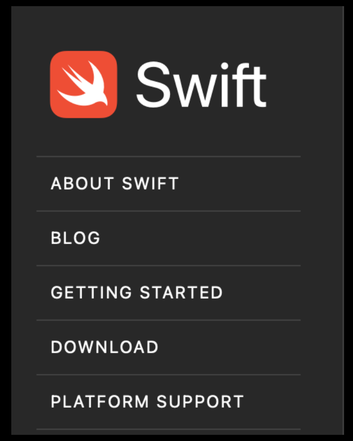
*Which **programming, scripting, and markup languages** have you done extensive development work in over the past year, and which do you want to work in over the next year?* (StackOverflow 2023 Developer Survey)

why?

“Swift is for Apple devices.”

why?

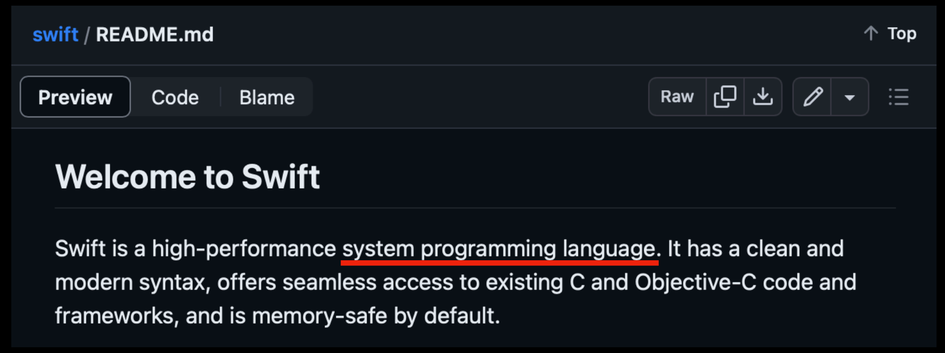
“Swift is for Apple devices.”



Welcome to Swift.org

Welcome to the Swift community. Together we are working to build a programming language to empower everyone to turn their ideas into apps on any platform.

Announced in 2014, the Swift programming language has quickly become one of the fastest growing languages in history. Swift makes it easy to write software that is incredibly fast and safe by design. Our goals for Swift are ambitious: we want to make programming simple things easy, and difficult things possible.



In which fields would you like to see Swift in the future?

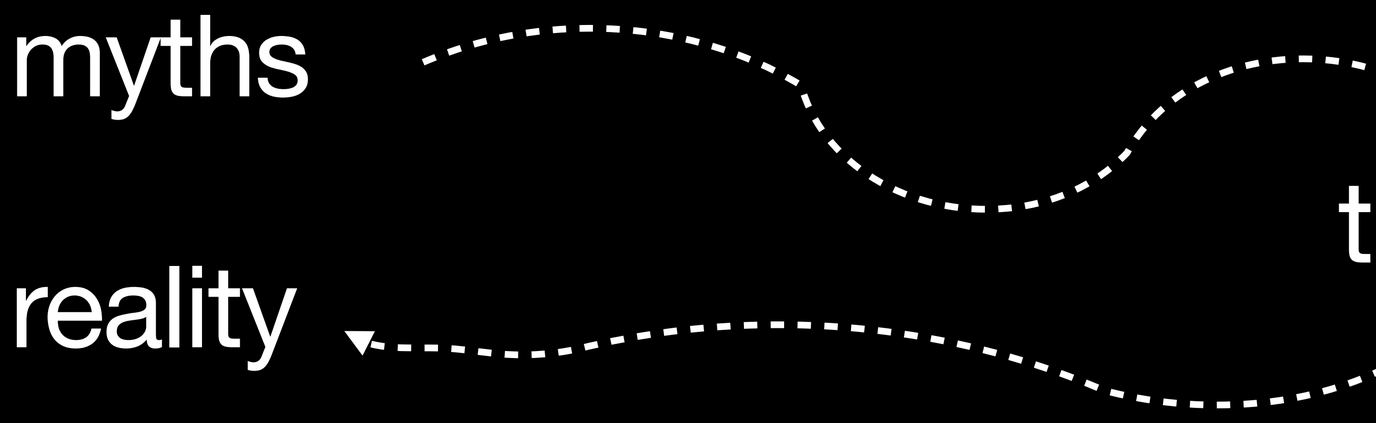
(1:15:32)

> **Lattner:** My goal for Swift has always been and still is total world domination. It’s a modest goal. *[Applause]*

the plan

myths
↓
reality

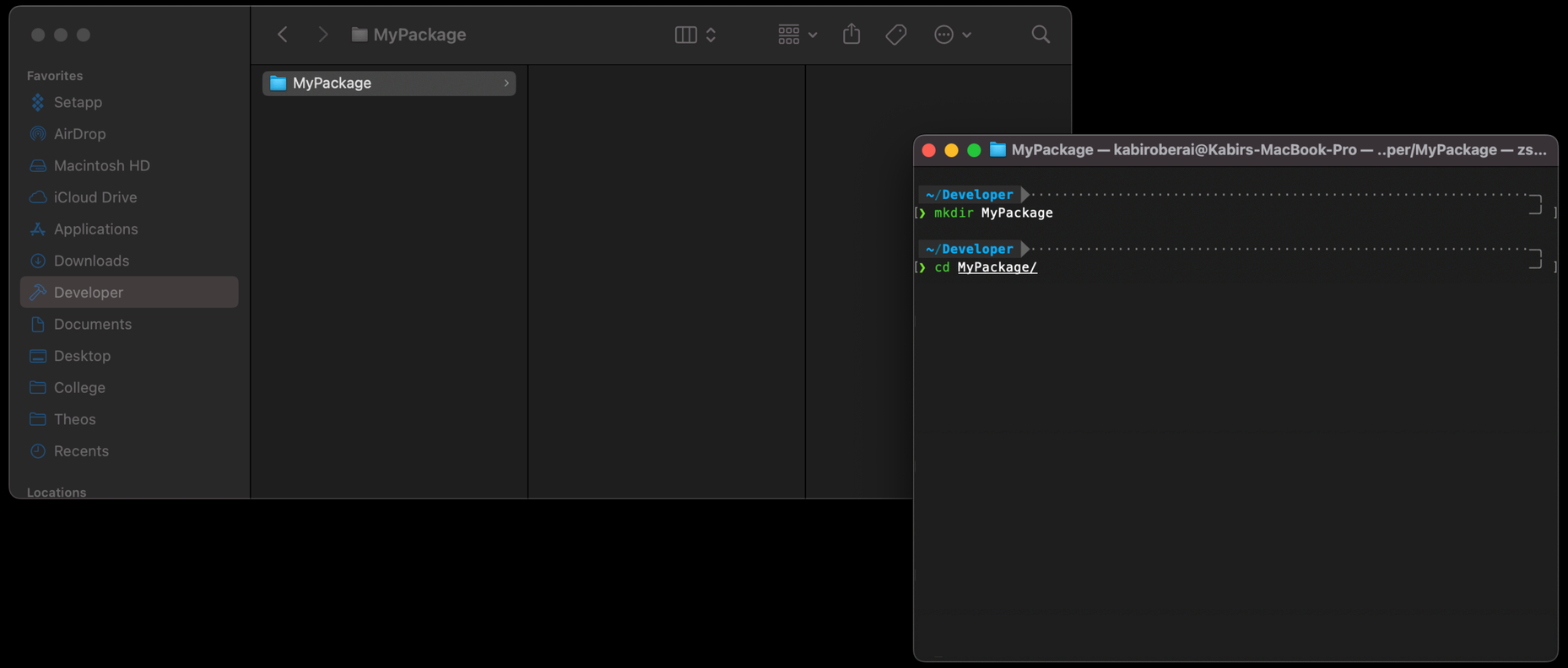
the plan



myth 1: *Xcode is the only decent way to build Swift code.*

myth 1: *Xcode is the only decent way to build Swift code.*

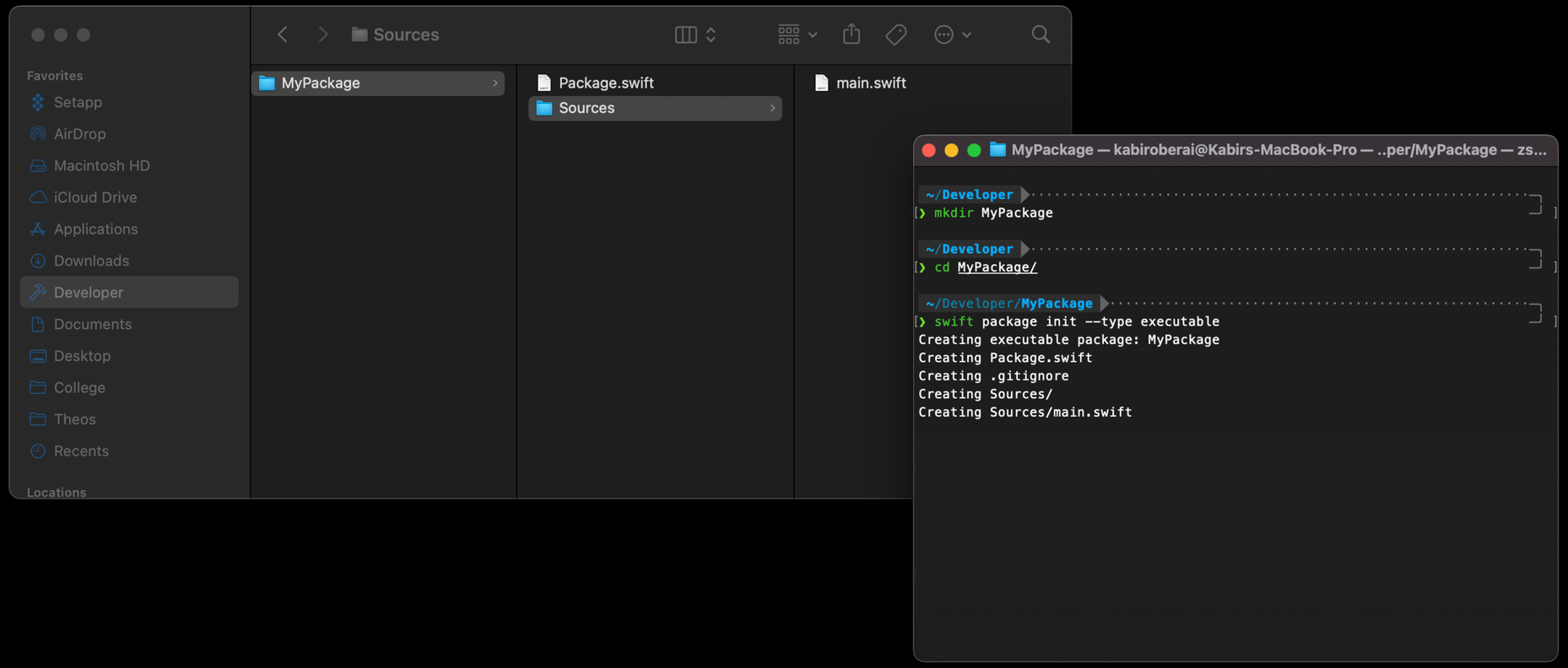
reality: SwiftPM is your friend on the command line 🤝



https://swift.org/getting-started/#using-the-package-manager

myth 1: *Xcode is the only decent way to build Swift code.*

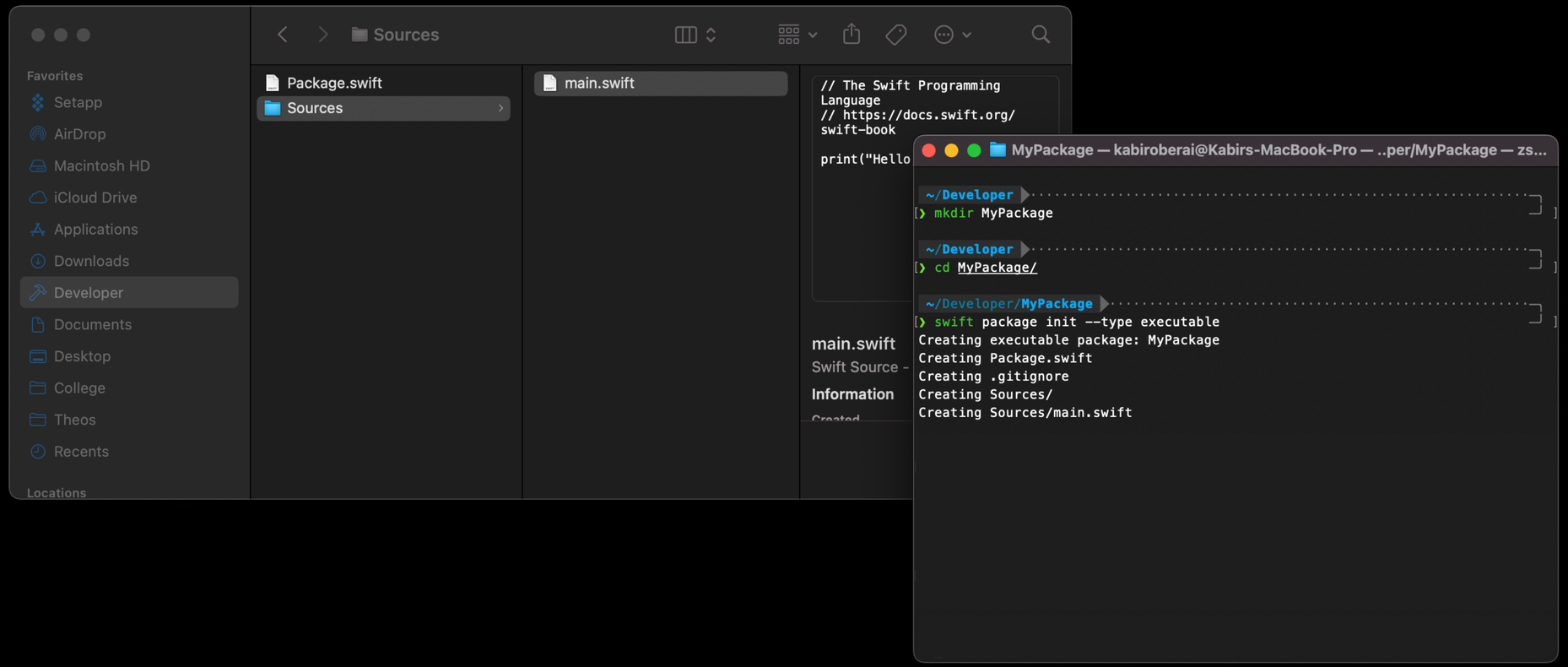
reality: SwiftPM is your friend on the command line 🤝



https://swift.org/getting-started/#using-the-package-manager

myth 1: *Xcode is the only decent way to build Swift code.*

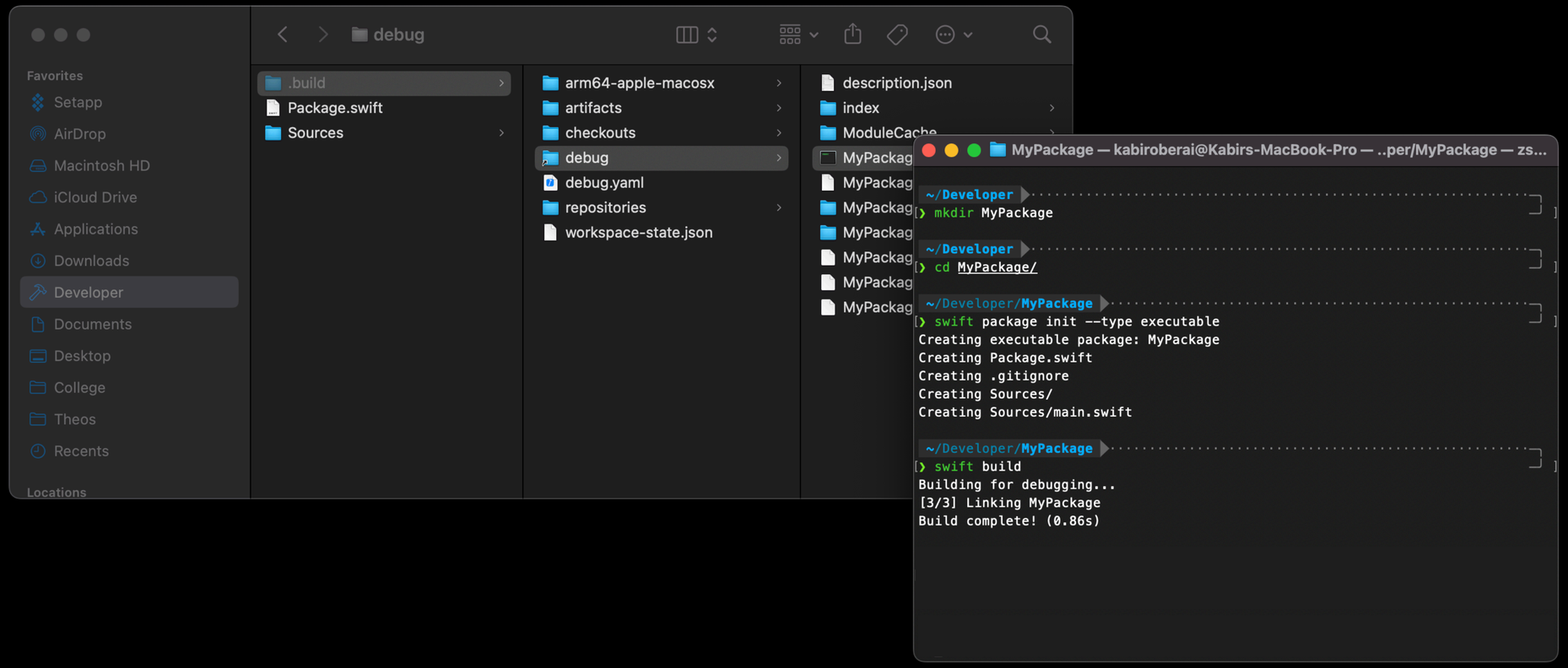
reality: SwiftPM is your friend on the command line 🤝



https://swift.org/getting-started/#using-the-package-manager

myth 1: *Xcode is the only decent way to build Swift code.*

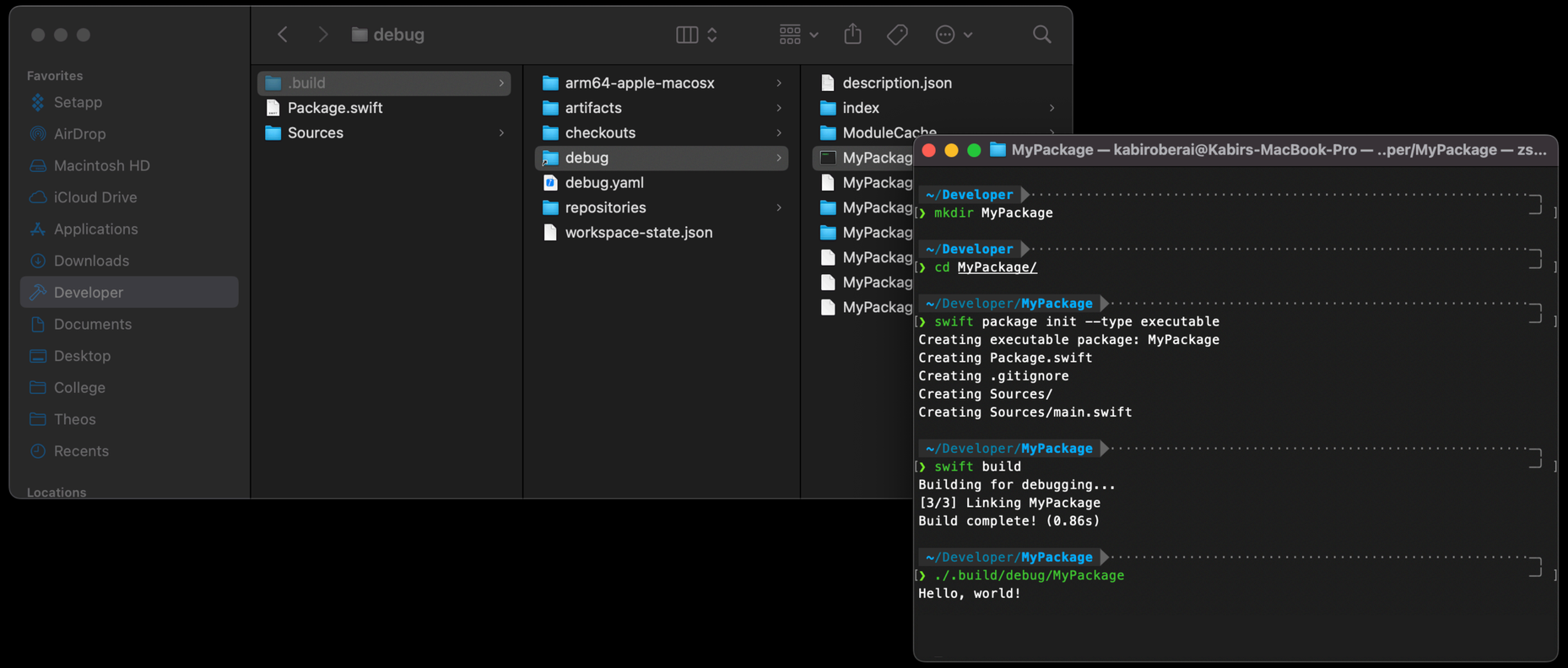
reality: SwiftPM is your friend on the command line 🤝



https://swift.org/getting-started/#using-the-package-manager

myth 1: *Xcode is the only decent way to build Swift code.*

reality: SwiftPM is your friend on the command line 🤝

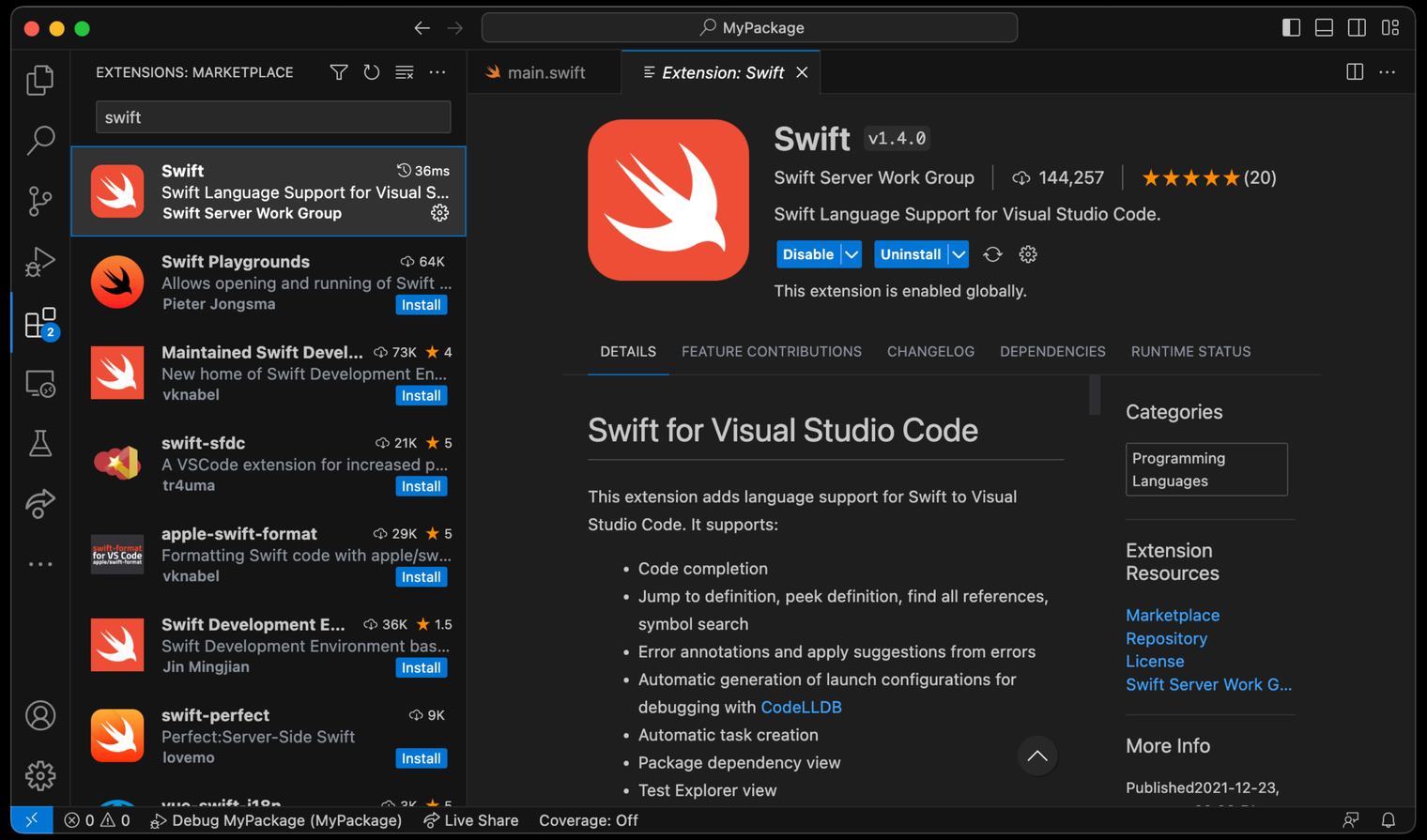


https://swift.org/getting-started/#using-the-package-manager

myth 2: *Xcode is the only decent way to edit Swift code.*

myth 2: *Xcode is the only decent way to edit Swift code.*

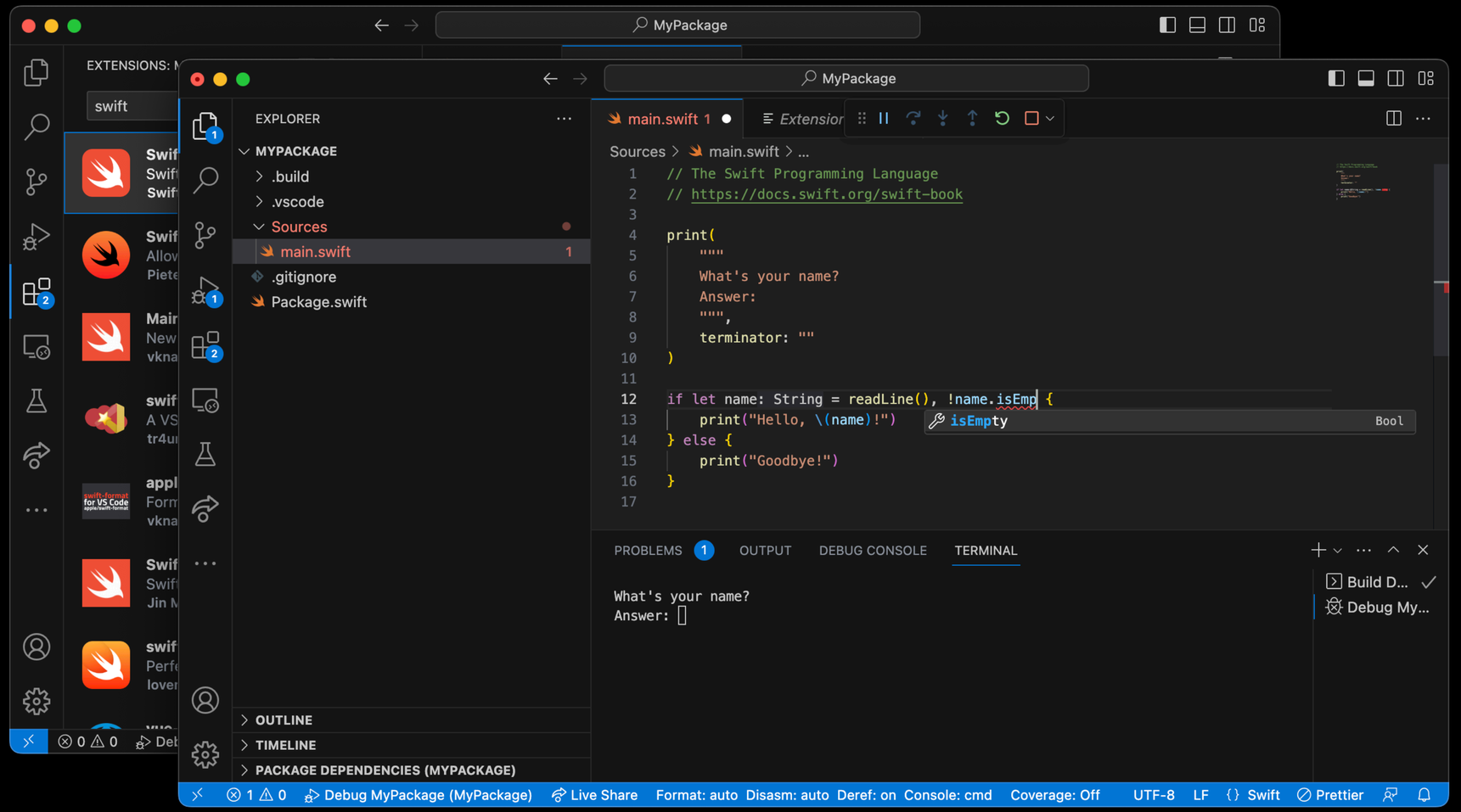
reality: SourceKit-LSP supports your favorite IDEs 🔍



apple/sourcekit-lsp

myth 2: *Xcode is the only decent way to edit Swift code.*

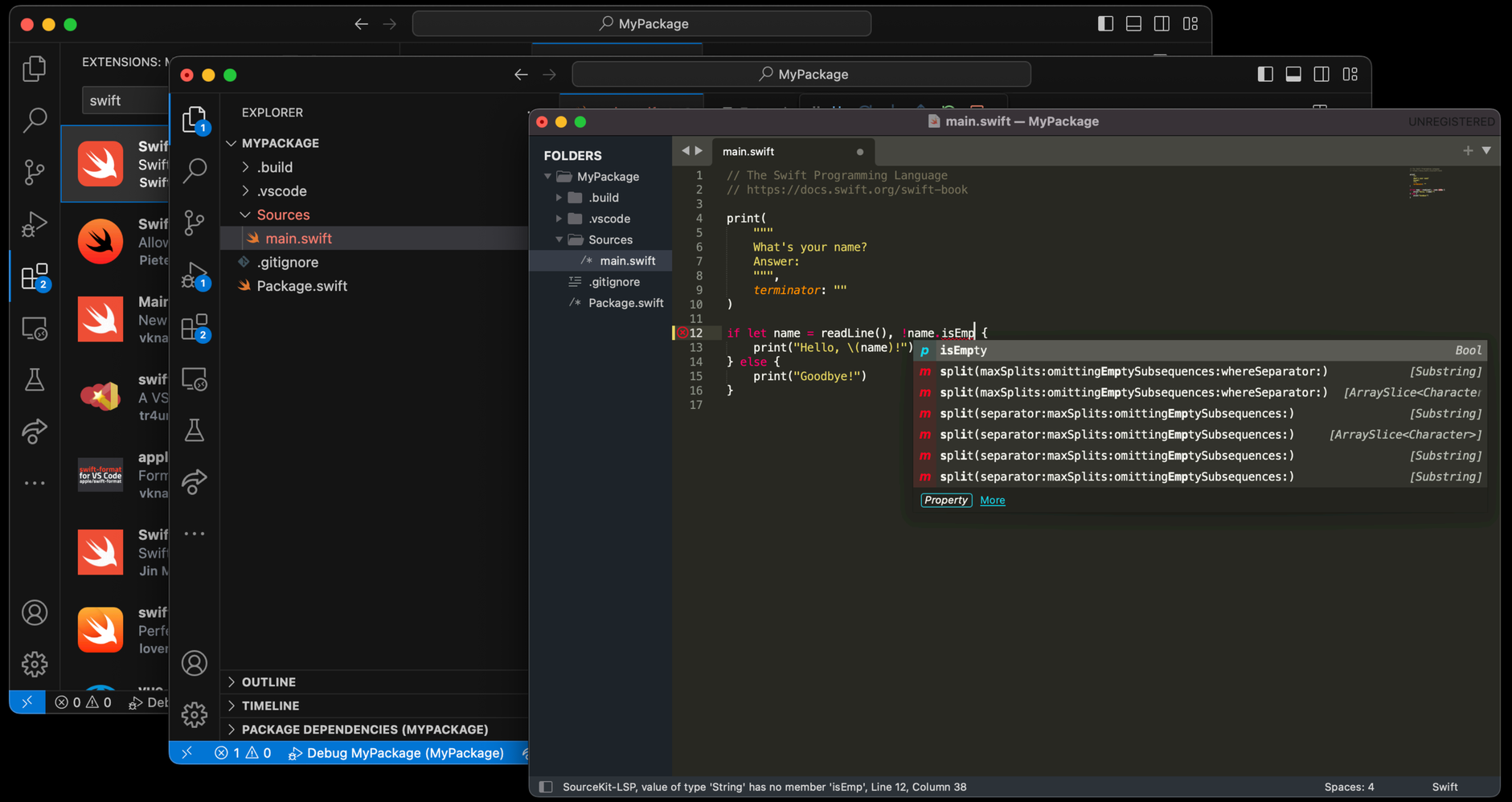
reality: SourceKit-LSP supports your favorite IDEs 🔍



apple/sourcekit-lsp

myth 2: *Xcode is the only decent way to edit Swift code.*

reality: SourceKit-LSP supports your favorite IDEs 🔍

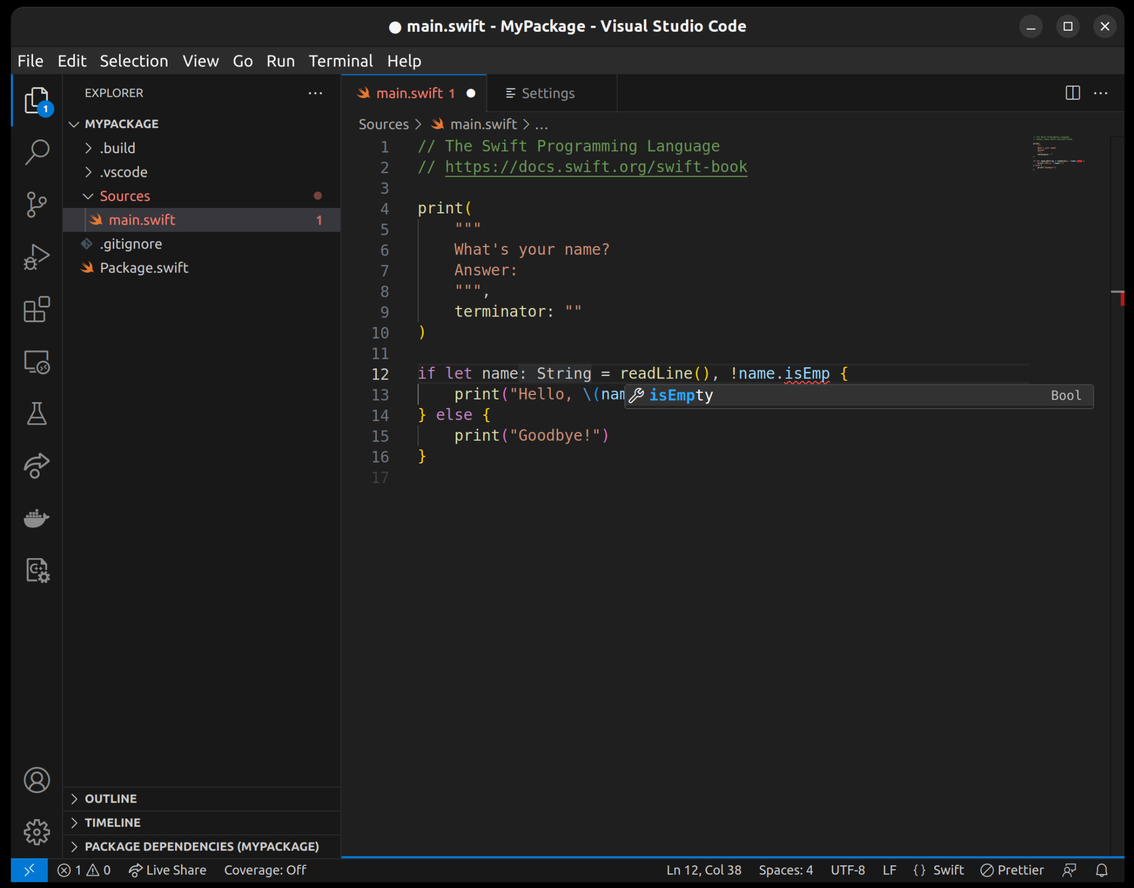


apple/sourcekit-lsp

myth 3: *You can only write (edit+build) Swift code on macOS*

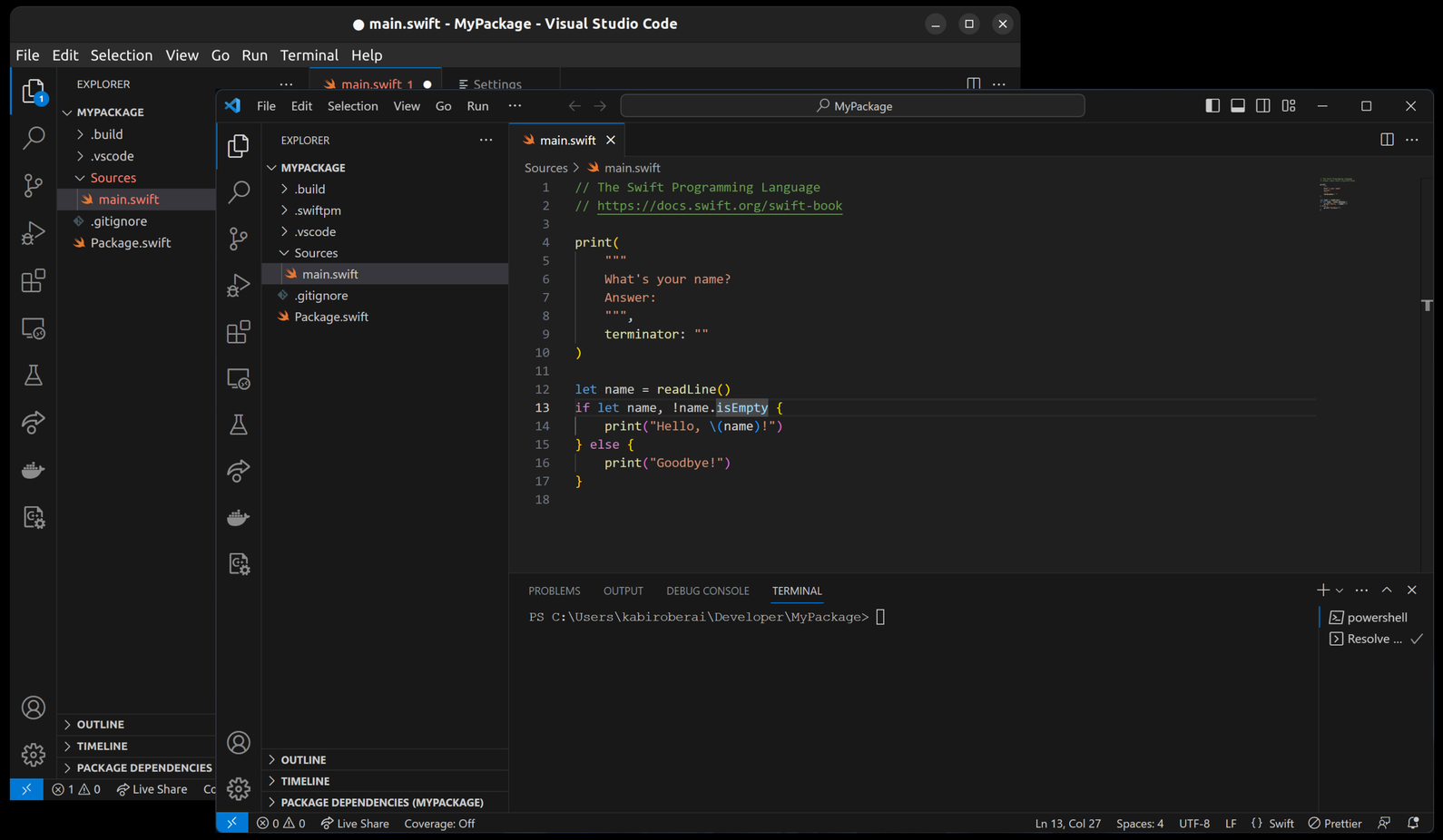
myth 3: *You can only write (edit+build) Swift code on macOS*

reality: You can write Swift code on Linux,



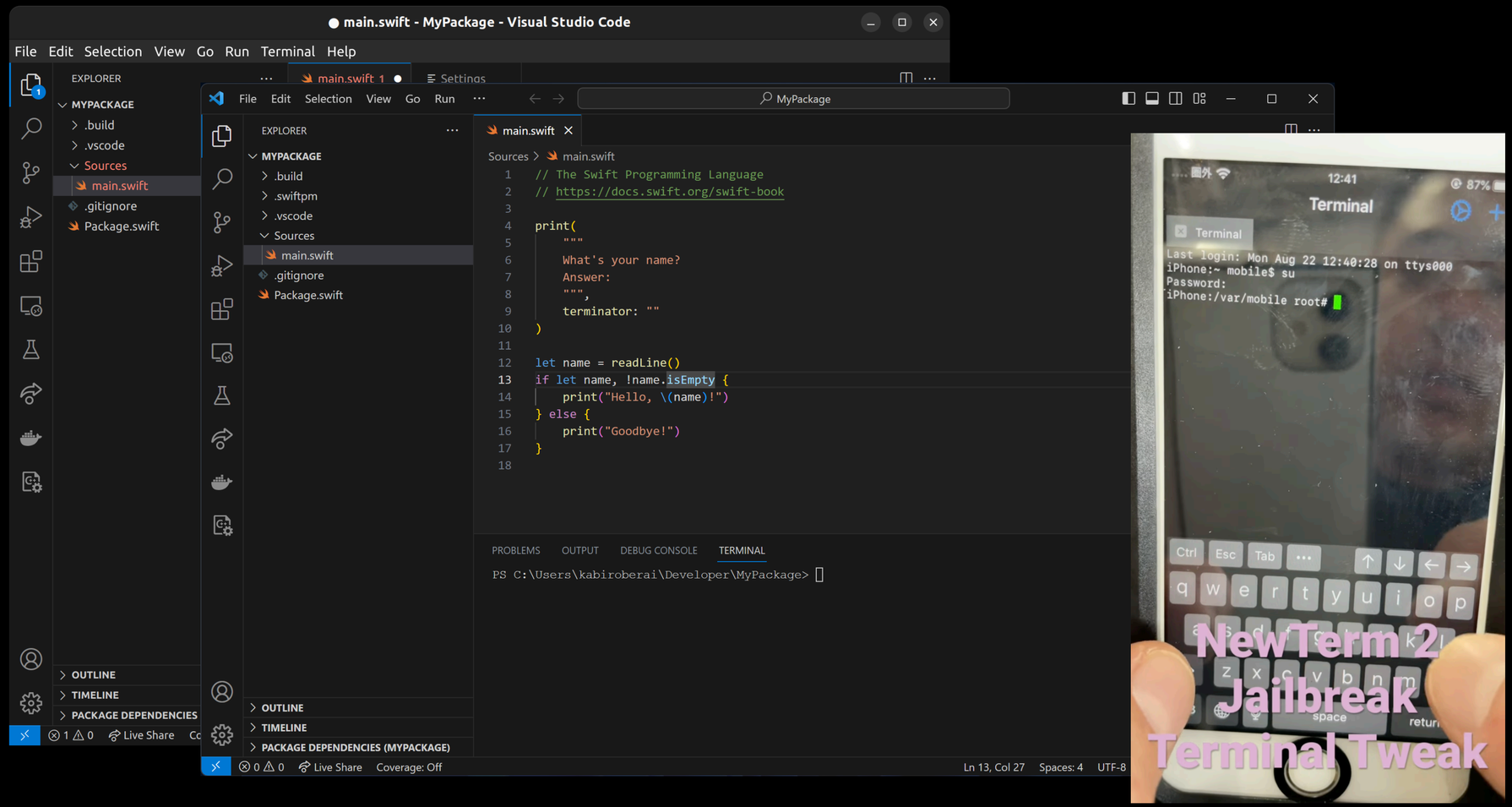
myth 3: *You can only write (edit+build) Swift code on macOS*

reality: You can write Swift code on Linux, Windows,



myth 3: *You can only write (edit+build) Swift code on macOS*

reality: You can write Swift code on Linux, Windows, and your (jail)broken iPhone 5 from 2012 🌐

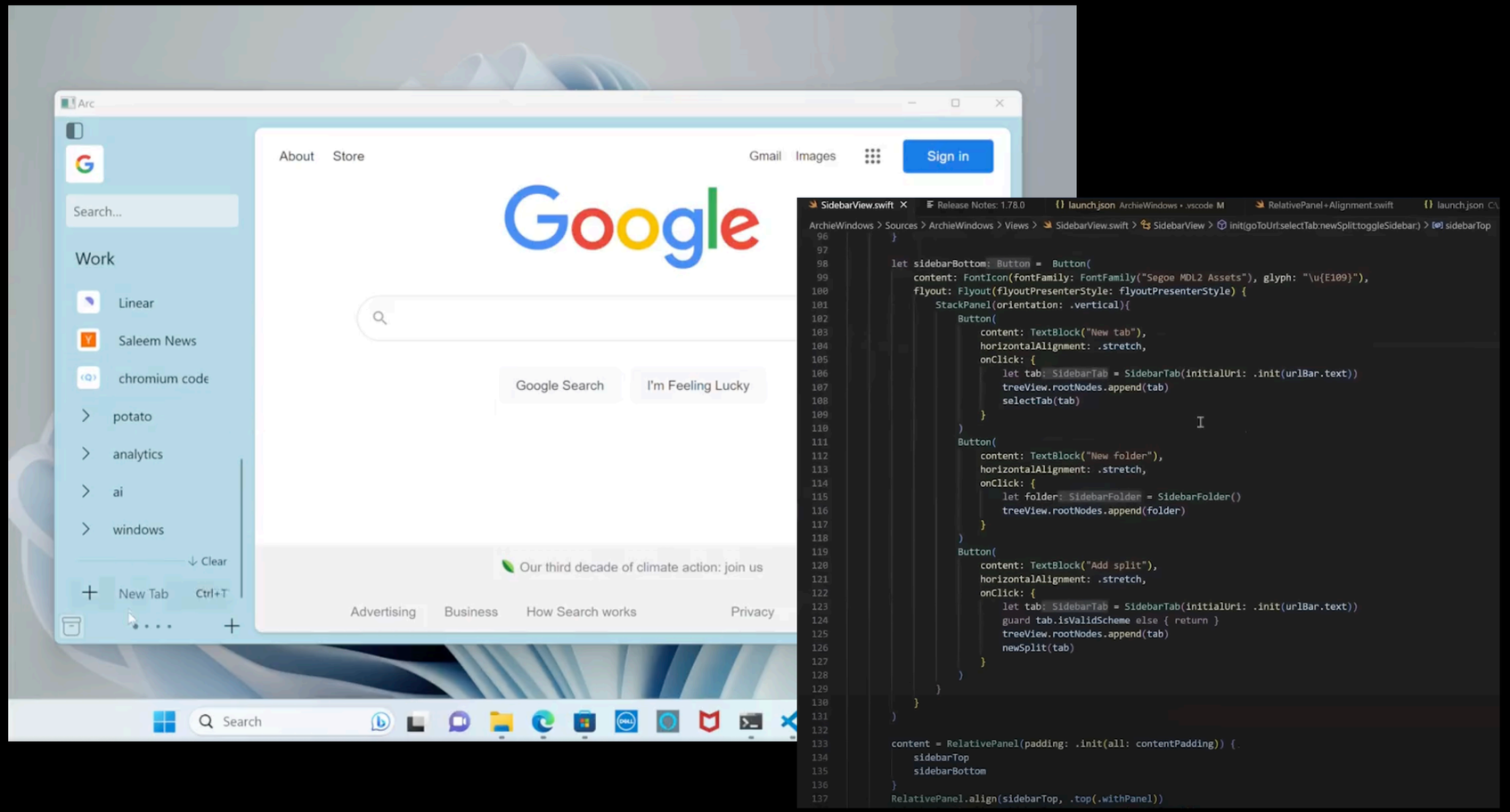


iPhone image: applemacjp

myth 4: *Swift on Linux and Windows is only for servers.*

myth 4: *Swift on Linux and Windows is only for servers.*

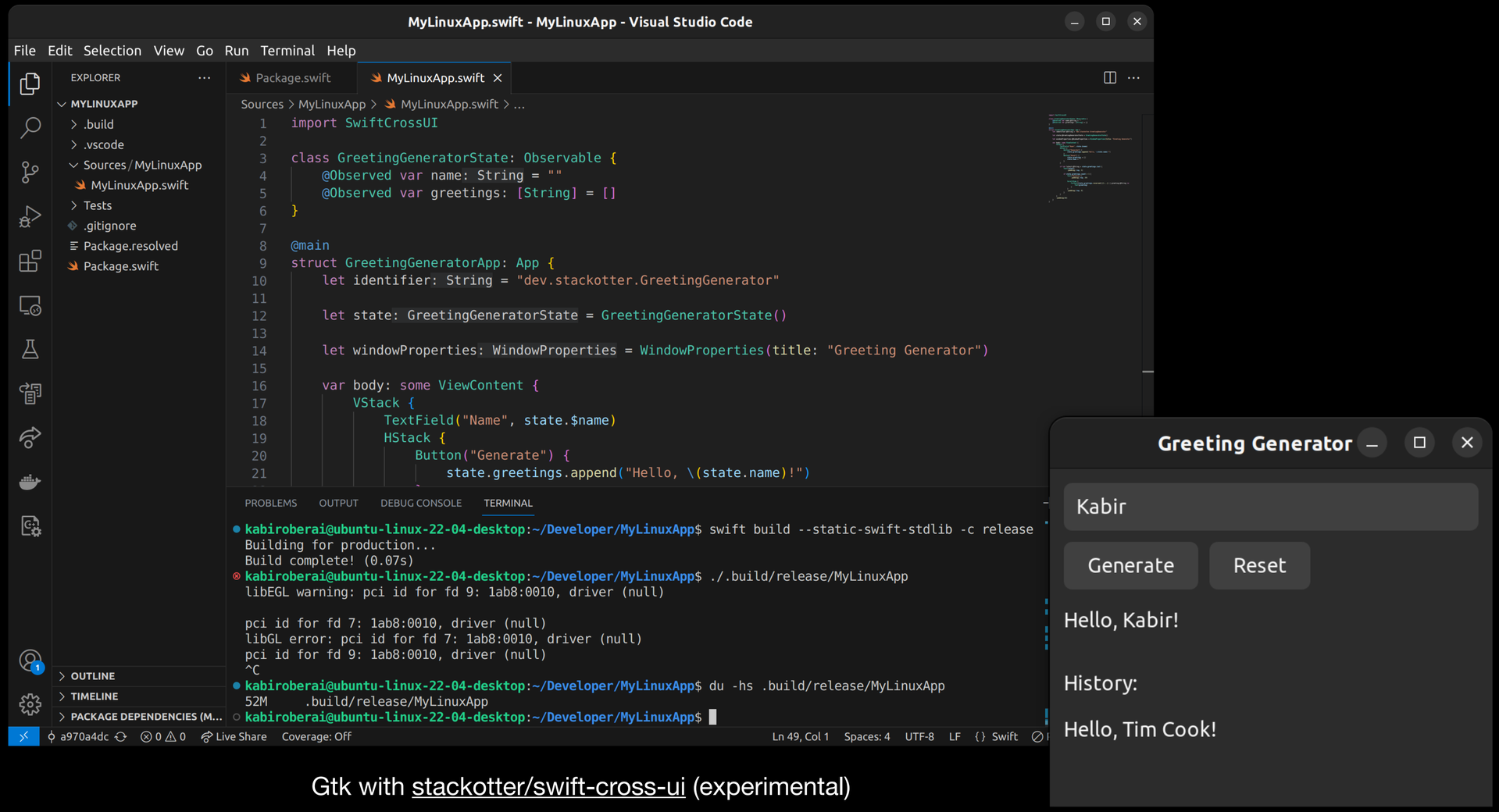
reality: Consumer apps are coming 👨‍👨‍👧



WinUI+Swift via The Browser Company

myth 4: *Swift on Linux and Windows is only for servers.*

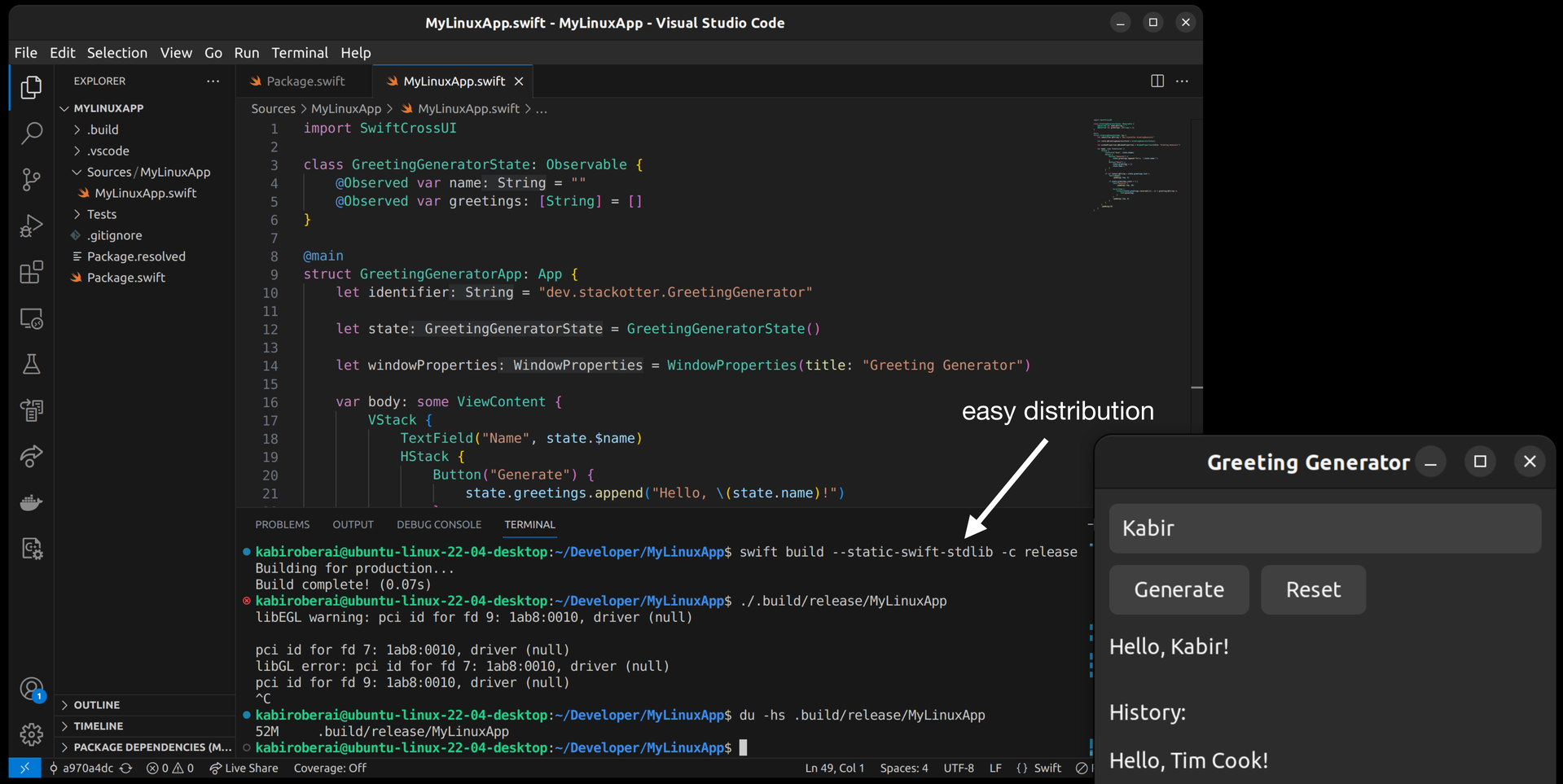
reality: Consumer apps are coming 👨‍👨‍👧



Gtk with stackotter/swift-cross-ui (experimental)

myth 4: *Swift on Linux and Windows is only for servers.*

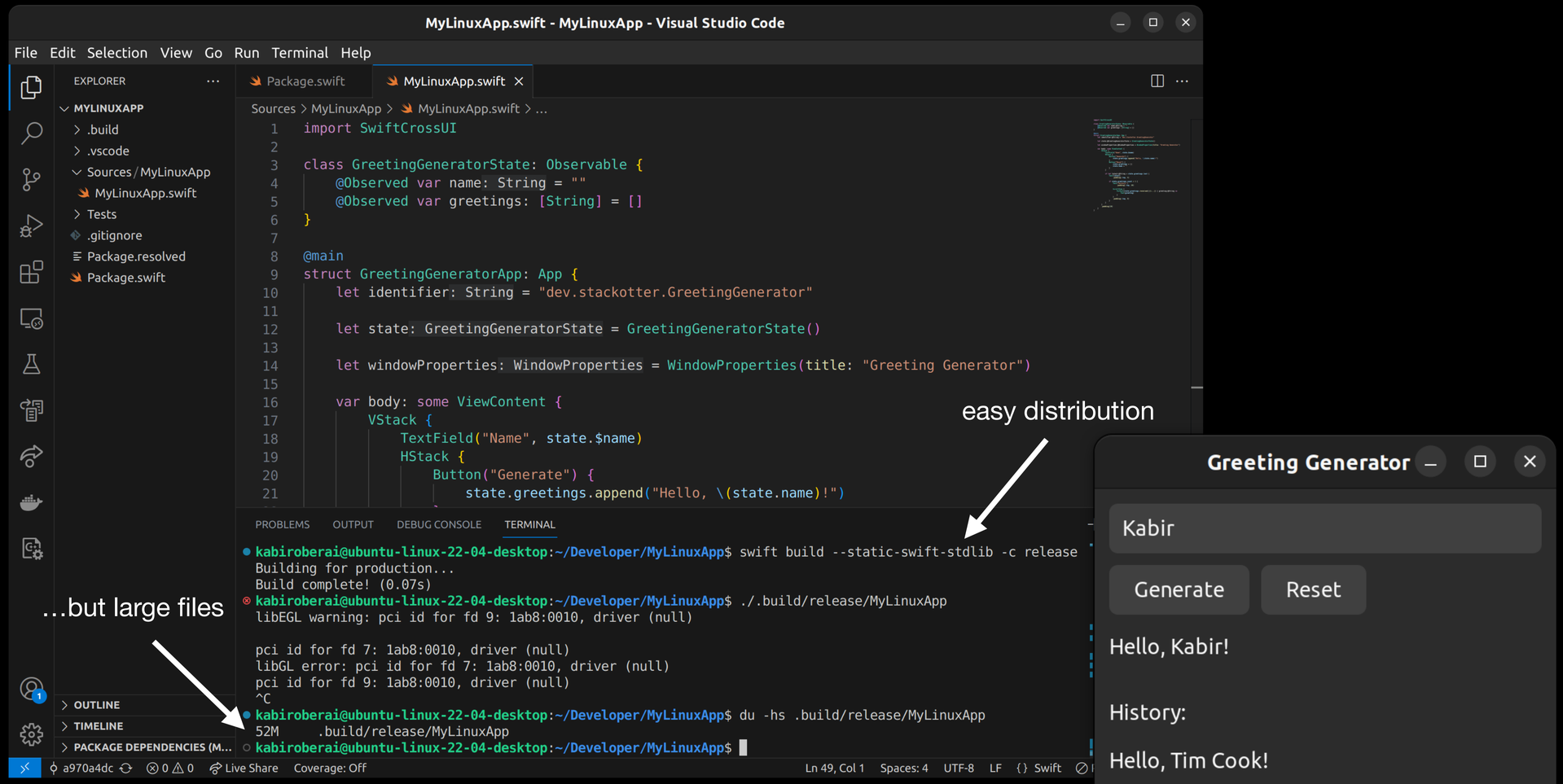
reality: Consumer apps are coming 👨‍👨‍👧



Gtk with stackotter/swift-cross-ui (experimental)

myth 4: *Swift on Linux and Windows is only for servers.*

reality: Consumer apps are coming 👨‍👨‍👧

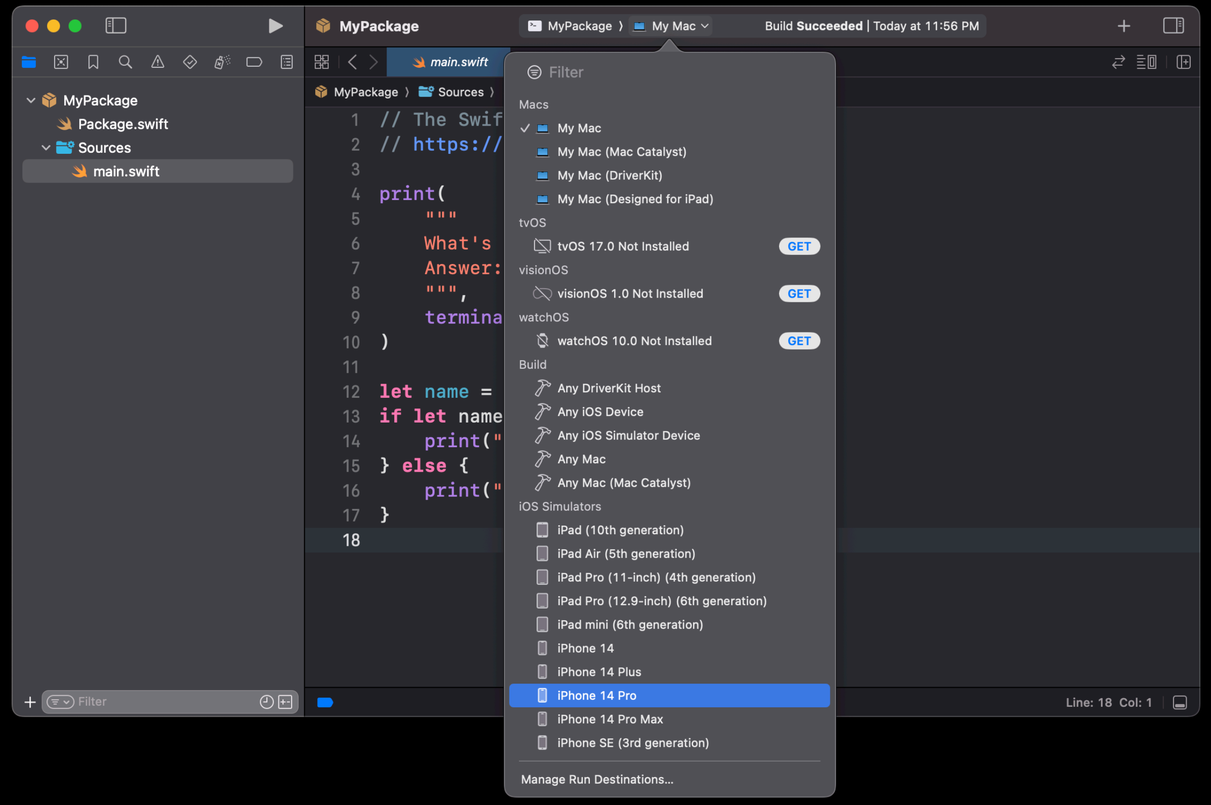


Gtk with stackotter/swift-cross-ui (experimental)

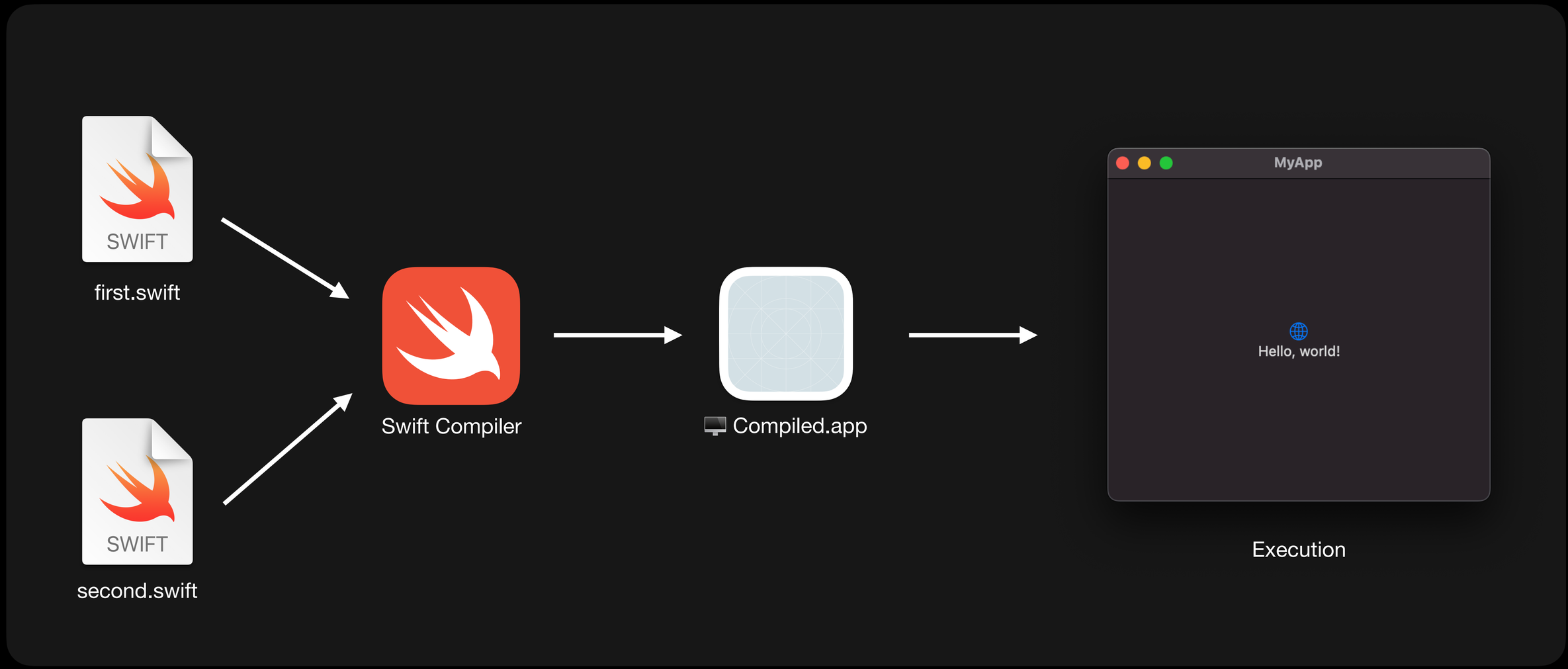
myth 5: *You can only build Swift code for your own platform.*

myth 5: *You can only build Swift code for your own platform.*

reality: Swift is a **cross-compiler** 🎭

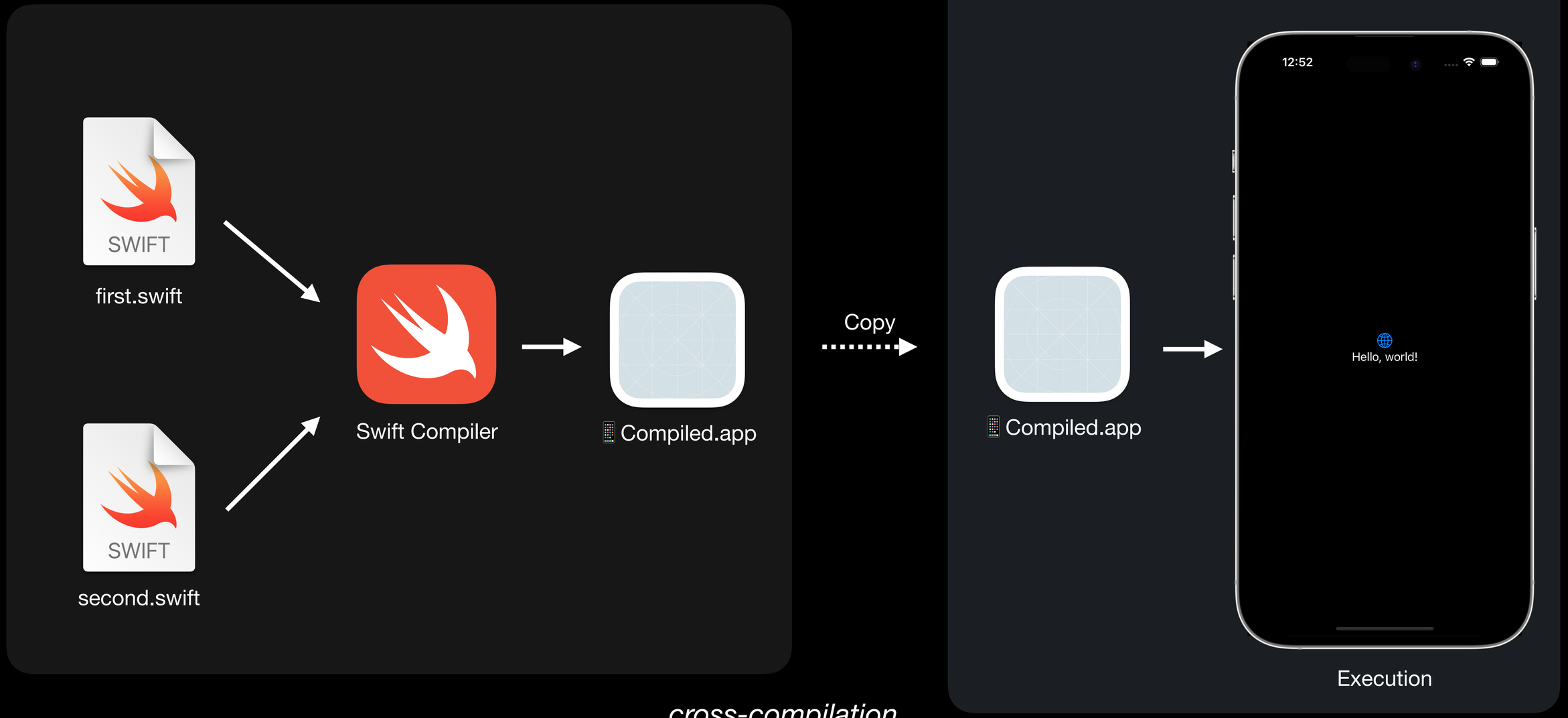


interlude: **theory time!**



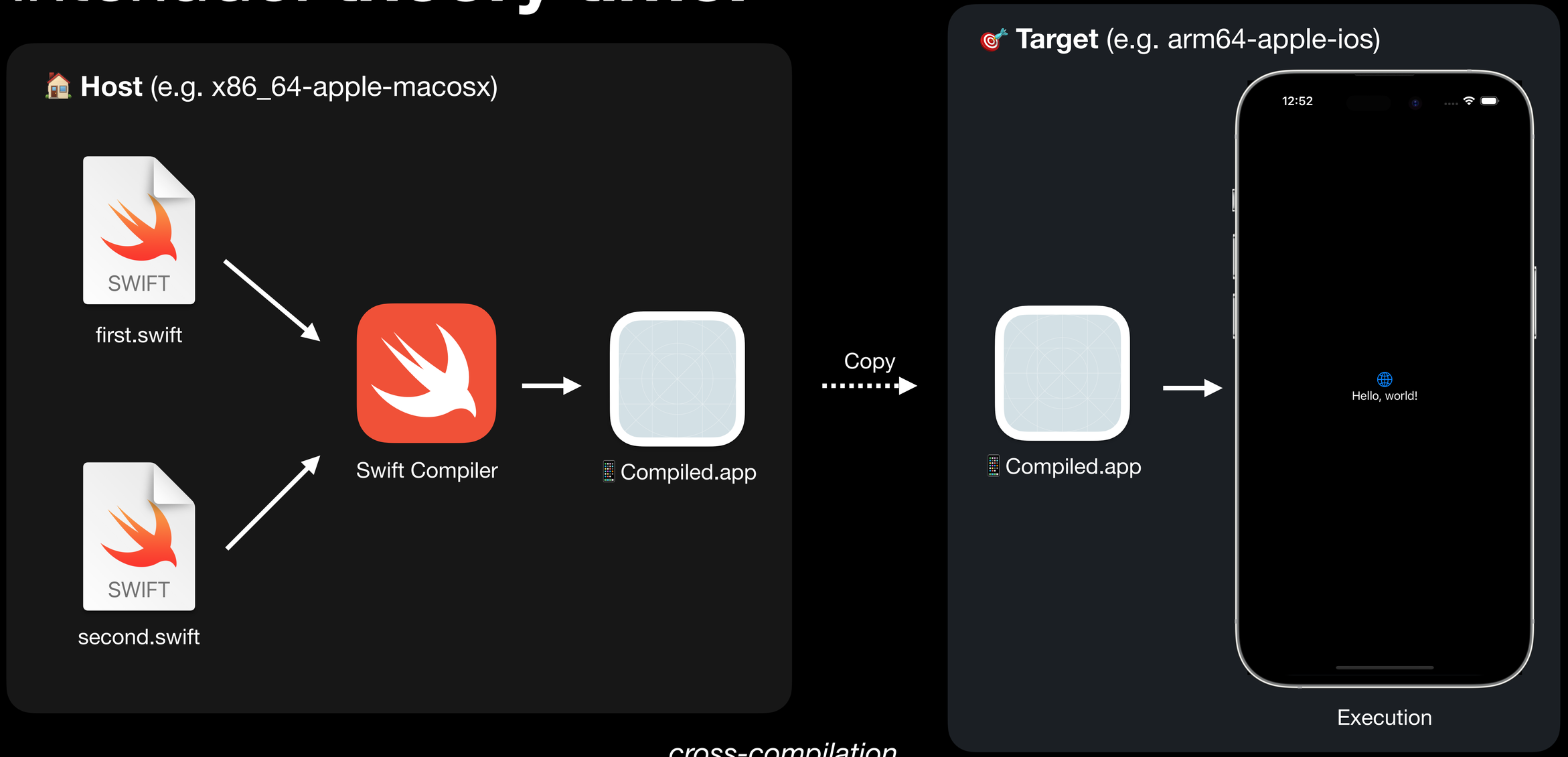
typical build process

interlude: **theory time!**



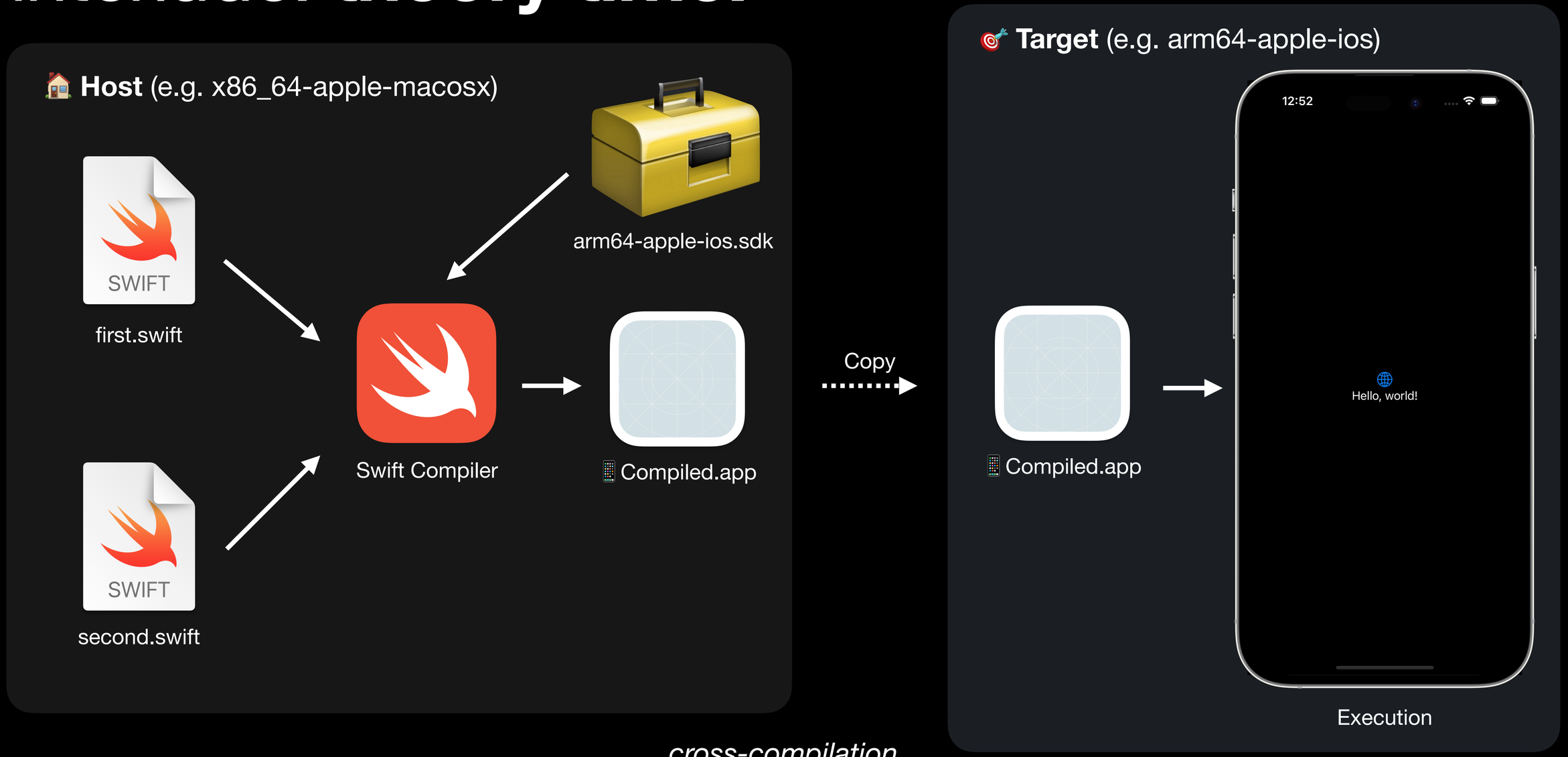
cross-compilation

interlude: **theory time!**



cross-compilation

interlude: **theory time!**



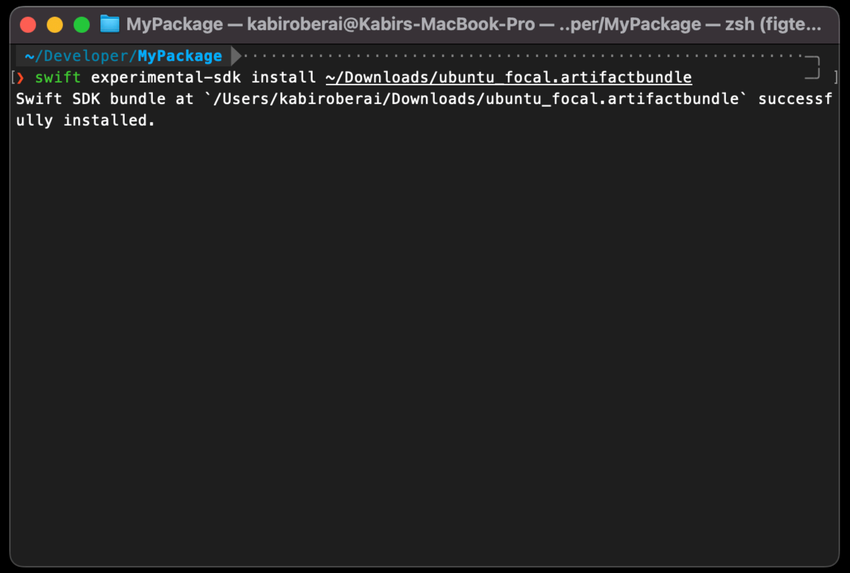
cross-compilation

myth 5: *You can only build Swift code for your own platform.*

reality: Swift is a **cross-compiler** 🎭

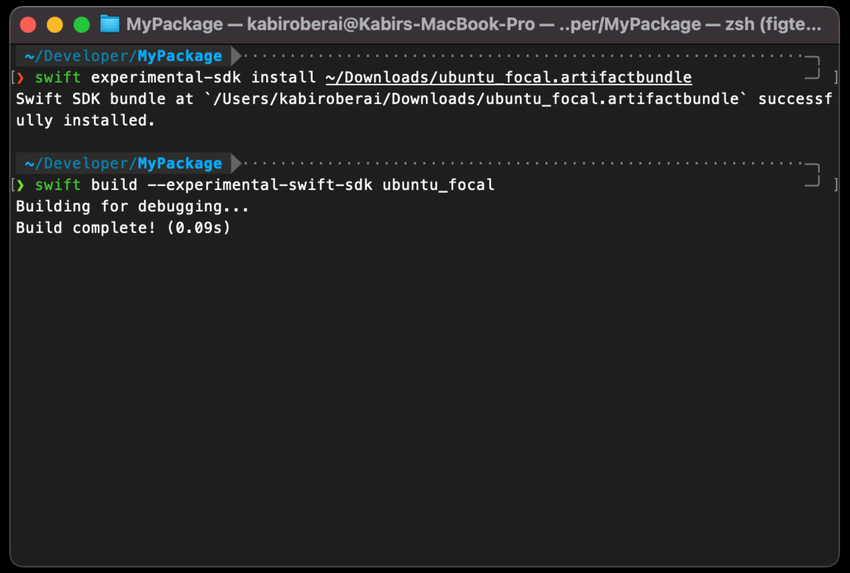
myth 5: *You can only build Swift code for your own platform.*

reality: Swift is a **cross-compiler** 🎭 and SwiftPM is too!



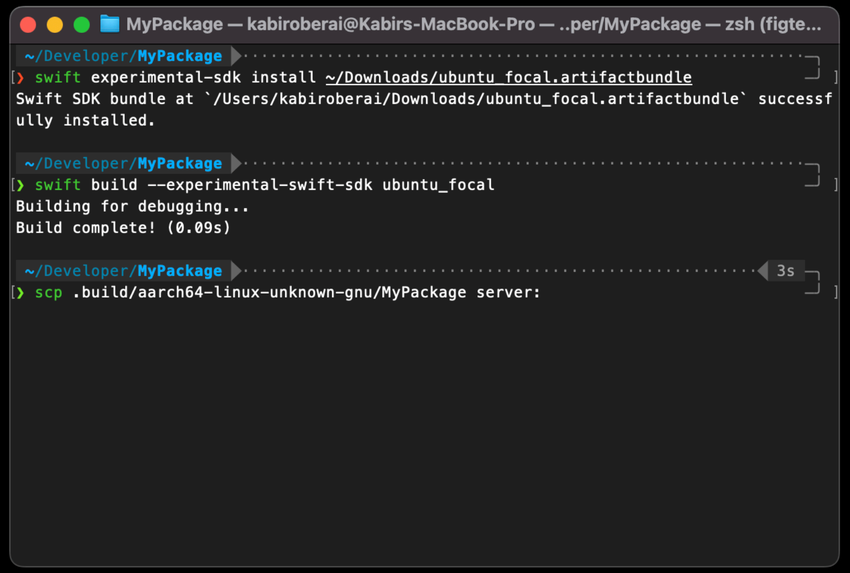
myth 5: *You can only build Swift code for your own platform.*

reality: Swift is a **cross-compiler** 🎭 and SwiftPM is too!



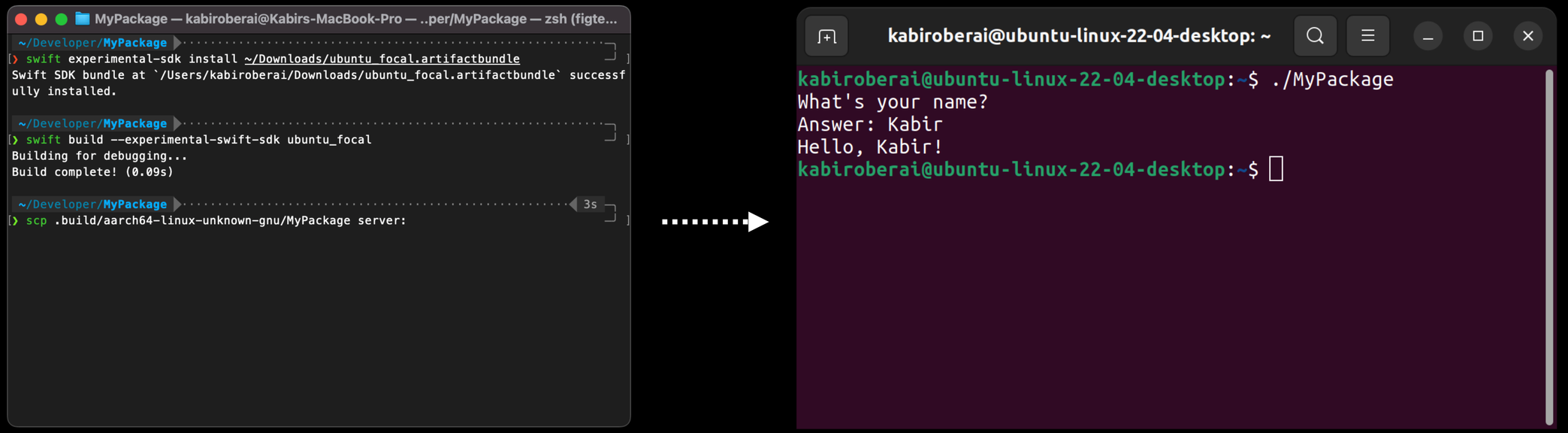
myth 5: *You can only build Swift code for your own platform.*

reality: Swift is a **cross-compiler** 🎭 and SwiftPM is too!



myth 5: *You can only build Swift code for your own platform.*

reality: Swift is a **cross-compiler** 🎭 and SwiftPM is too!

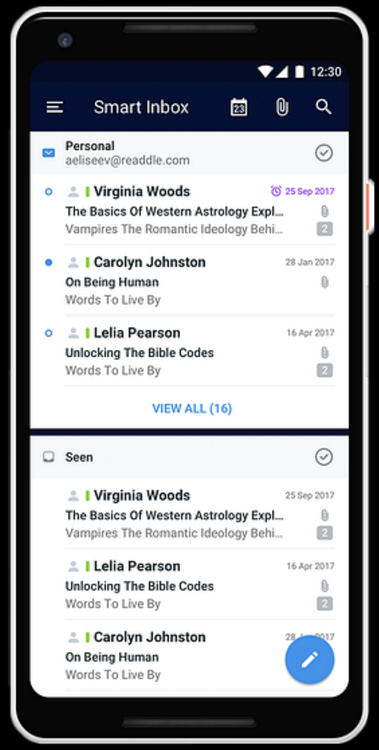


SE-0387: Swift SDKs for Cross Compilation

myth 6: *Swift only targets Darwin, Linux, and Windows.*

myth 6: *Swift only targets Darwin, Linux, and Windows.*

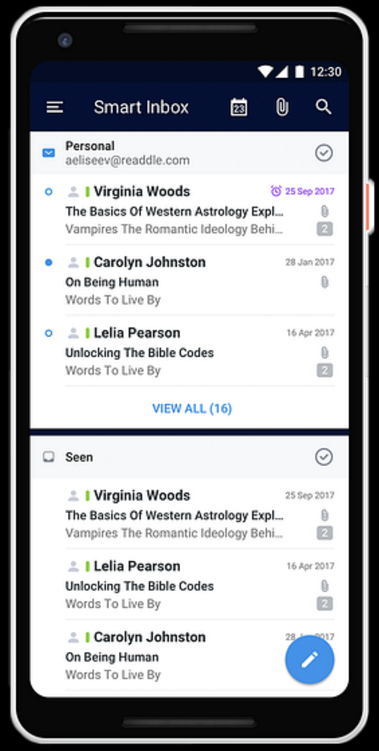
reality: Many other platforms are supported 🤖



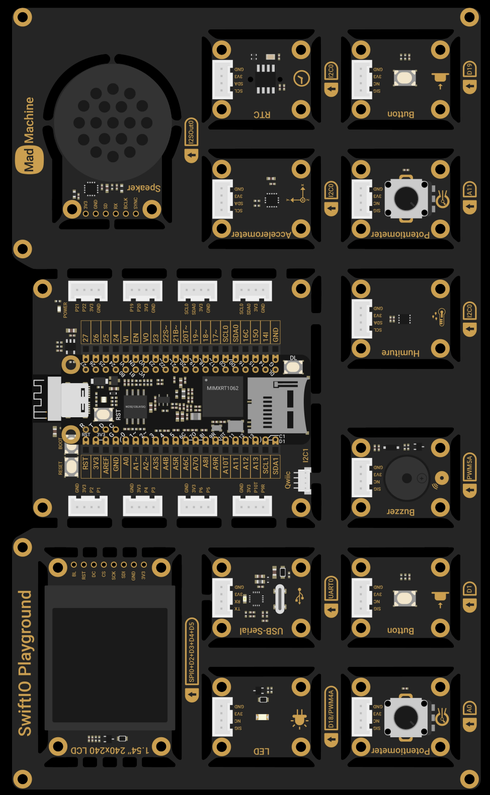
Spark for Android (Readdle)

myth 6: *Swift only targets Darwin, Linux, and Windows.*

reality: Many other platforms are supported 🤖



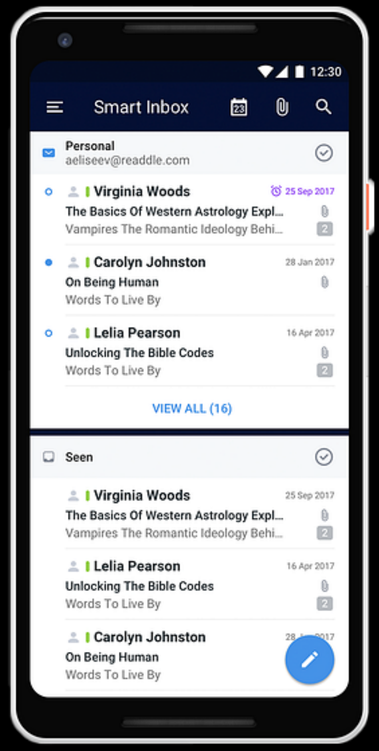
Spark for Android (Readdle)



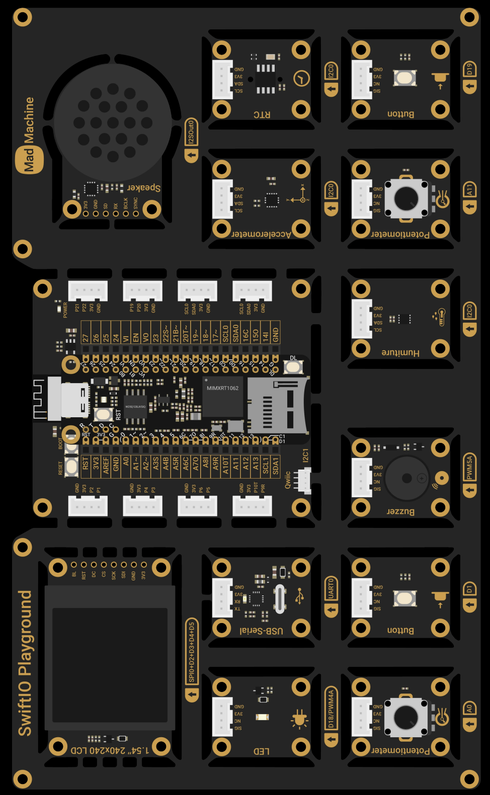
embedded/bare metal (MadMachine)

myth 6: *Swift only targets Darwin, Linux, and Windows.*

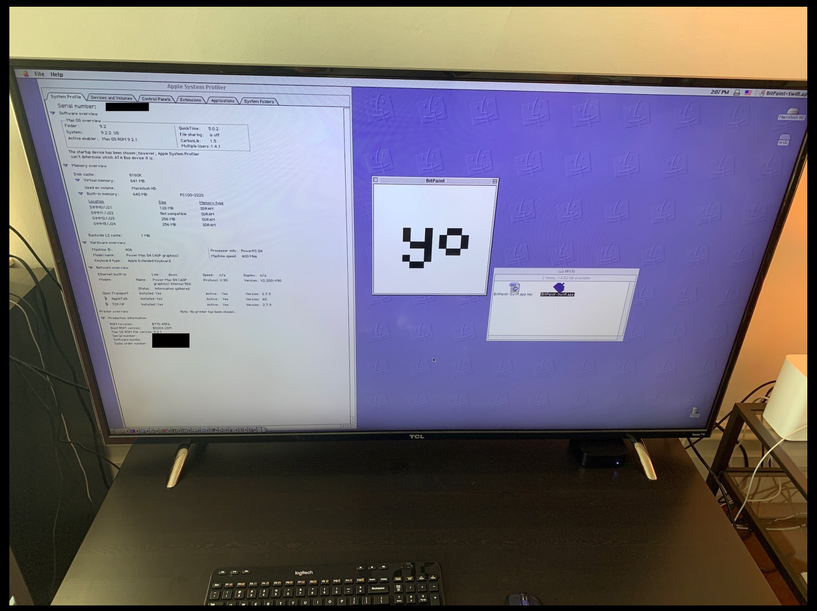
reality: Many other platforms are supported 🤖



Spark for Android (Readdle)



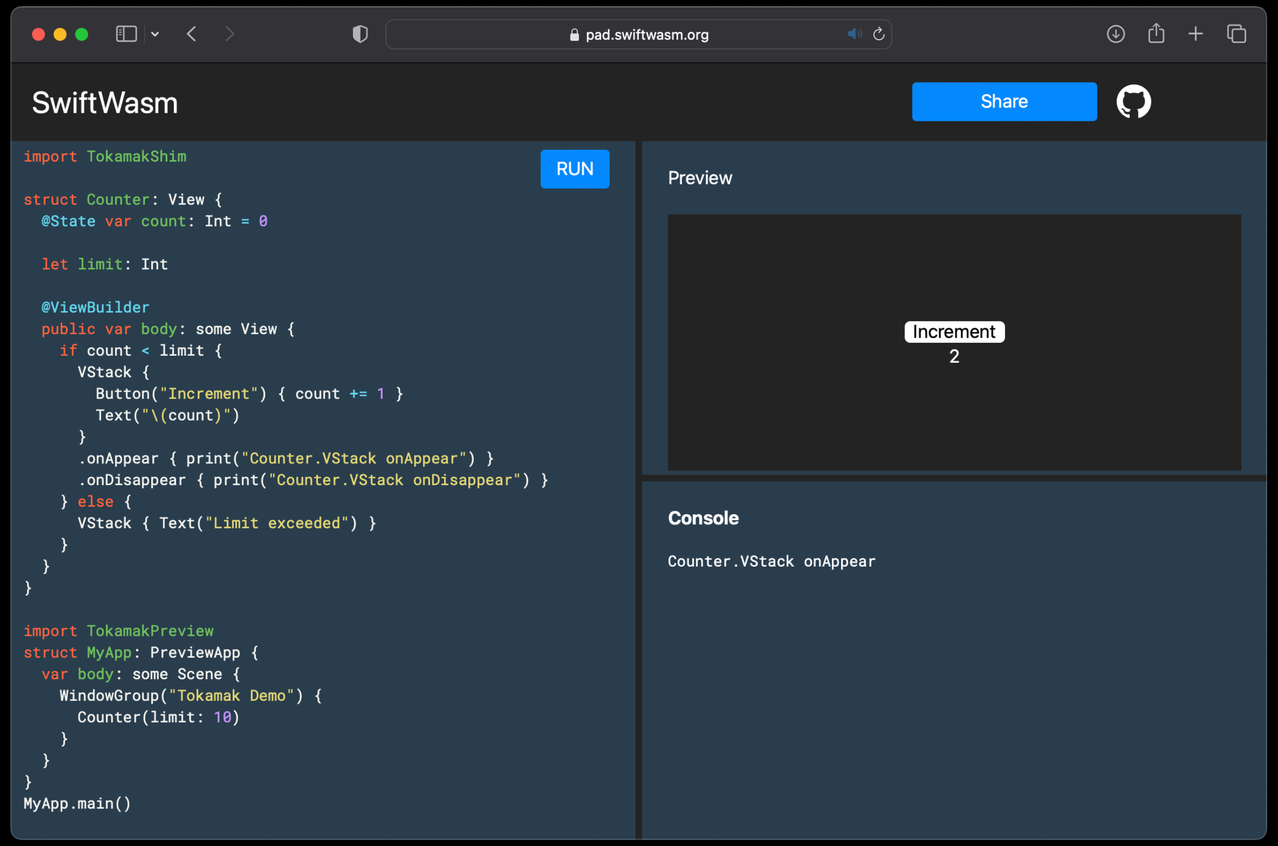
embedded/bare metal (MadMachine)



and… Mac OS 9 (Jordan Rose)

myth 6: *Swift only targets Darwin, Linux, and Windows.*

reality: Many other platforms are supported 🤖



Swift in WebAssembly with SwiftWasm and Tokamak

myth 7: *When choosing a cross-platform language, it's either/or.*

myth 7: *When choosing a cross-platform language, it's either/or.*

reality: Swift interoperates with more languages than you think 👄

rust-bindgen

built-in

PythonKit

(…and more)

Kotlin Native

built-in (5.9+)

myth 7: *When choosing a cross-platform language, it's either/or.*

reality: Swift interoperates with more languages than you think 👄

rust-bindgen

built-in

PythonKit

(…and more)

Kotlin Native

built-in (5.9+)

JavaScriptKit

myth 7: *When choosing a cross-platform language, it's either/or.*

reality: Swift interoperates with more languages than you think 👄

rust-bindgen

built-in

PythonKit

JavaScriptCore

(…and more)

Kotlin Native

built-in (5.9+)

JavaScriptKit

myth 7: *When choosing a cross-platform language, it's either/or.*

reality: Swift interoperates with more languages than you think 👄

rust-bindgen

built-in

Node.js/Electron

PythonKit

JavaScriptCore

(…and more)

Kotlin Native

built-in (5.9+)

JavaScriptKit

myth 7: *When choosing a cross-platform language, it's either/or.*

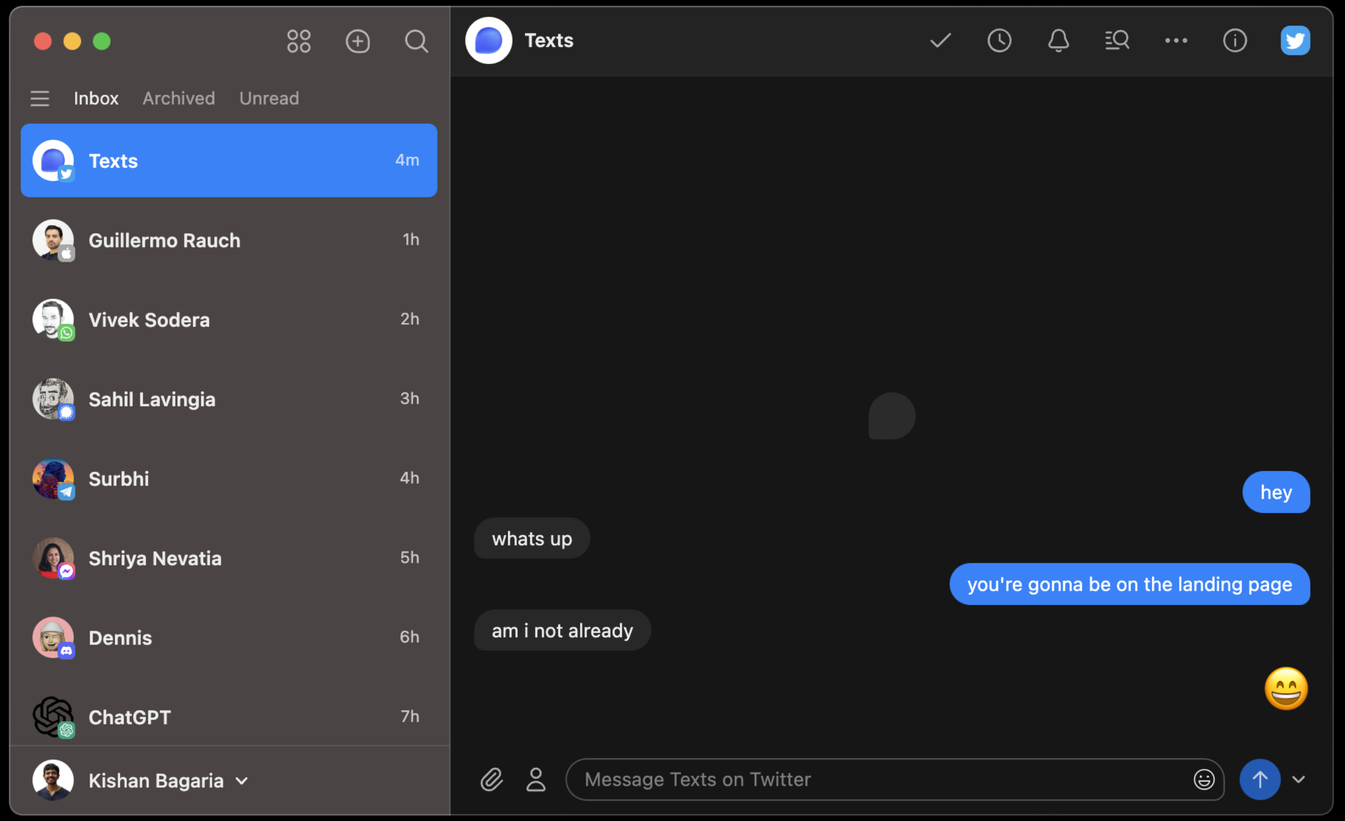
reality: Swift interoperates with more languages than you think 👄

kabiroberai/NodeSwift

myth 7: *When choosing a cross-platform language, it's either/or.*

reality: Swift interoperates with more languages than you think 👄

kabiroberai/NodeSwift

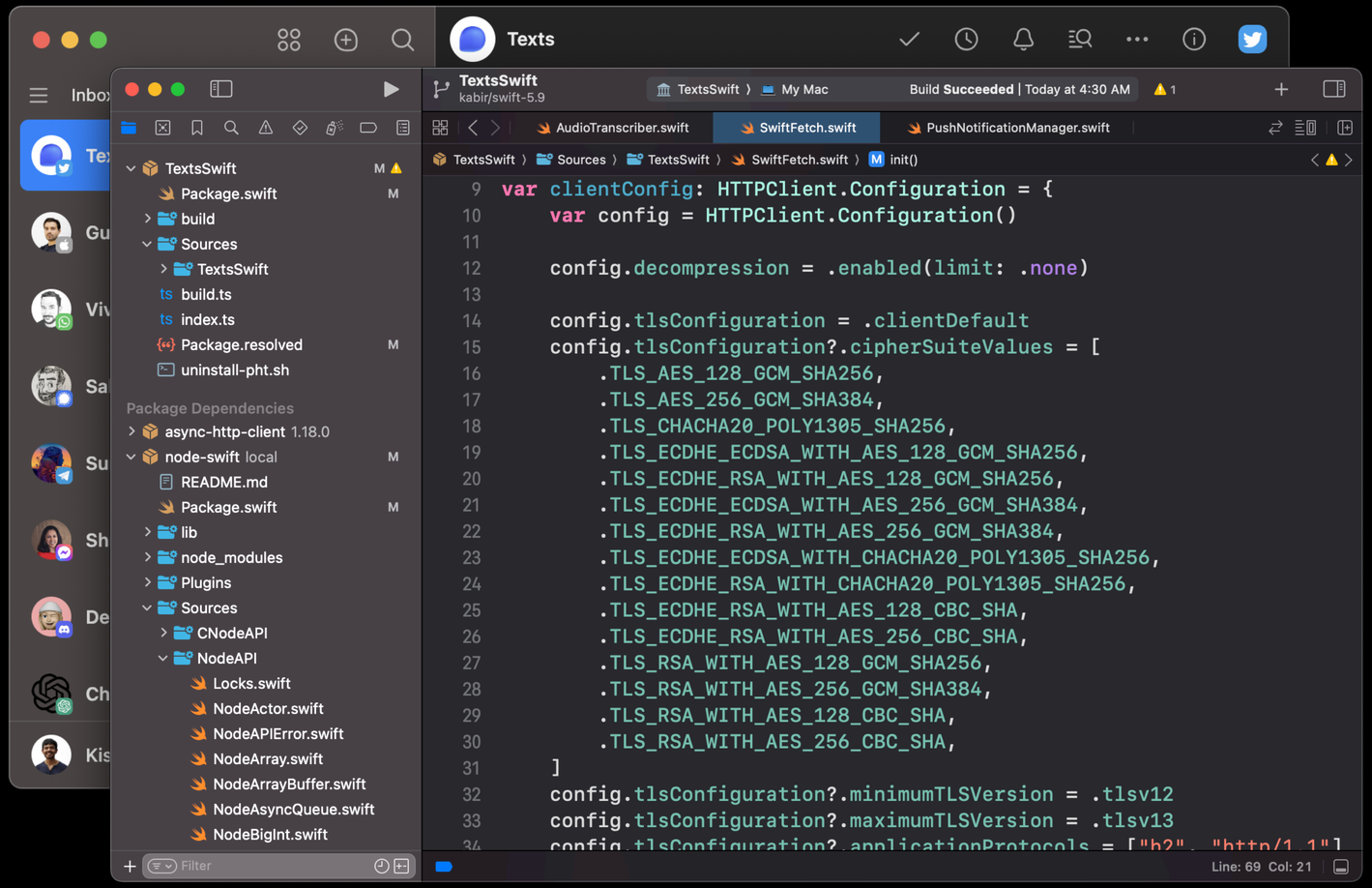


Texts

myth 7: *When choosing a cross-platform language, it's either/or.*

reality: Swift interoperates with more languages than you think 👄

kabiroberai/NodeSwift



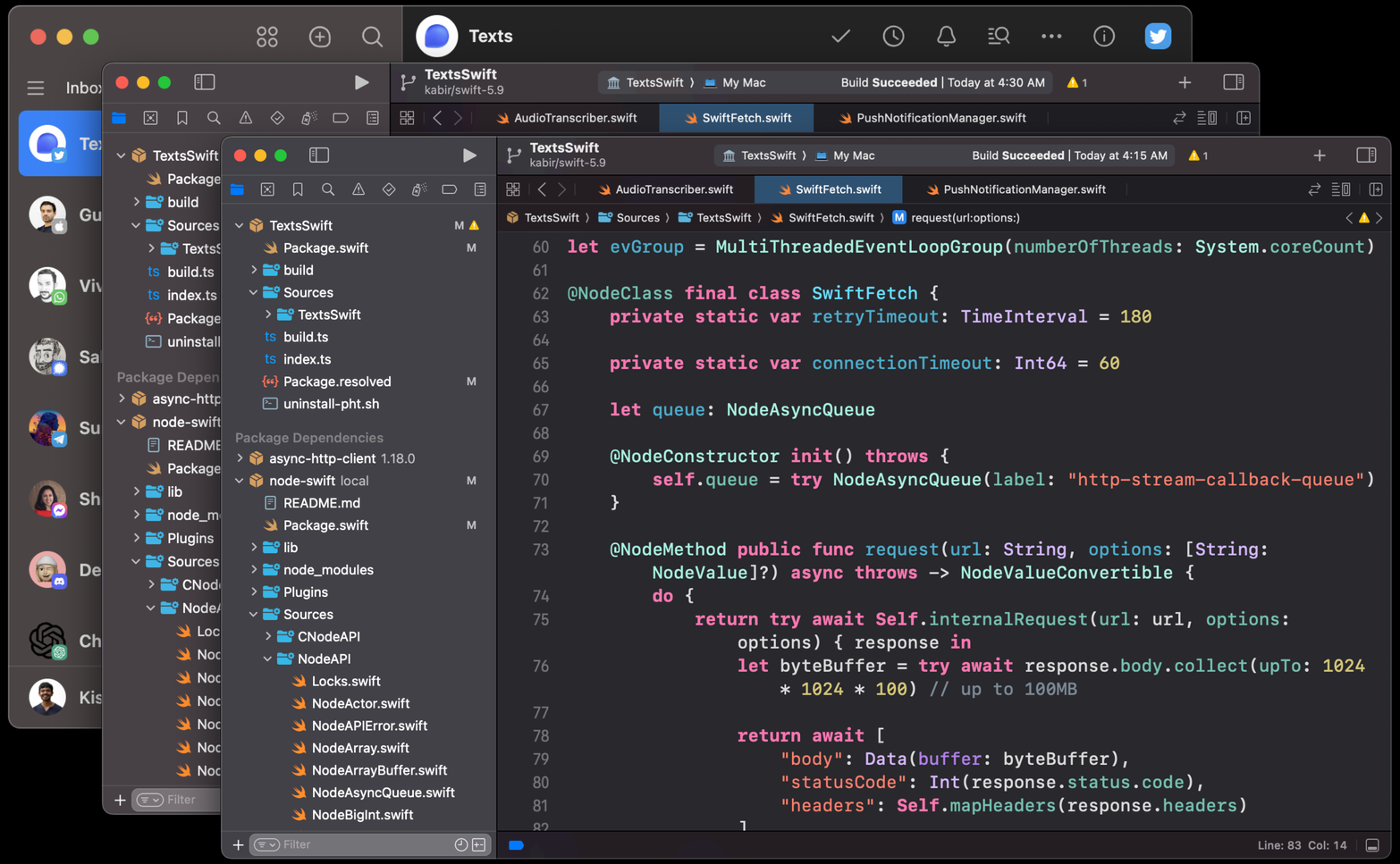

```swift
var clientConfig: HTTPClient.Configuration = {
    var config = HTTPClient.Configuration()

    config.decompression = .enabled(limit: .none)

    config.tlsConfiguration = .clientDefault
    config.tlsConfiguration?.cipherSuiteValues = [
        .TLS_AES_128_GCM_SHA256,
        .TLS_AES_256_GCM_SHA384,
        .TLS_CHACHA20_POLY1305_SHA256,
        .TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
        .TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
        .TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
        .TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
        .TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256,
        .TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256,
        .TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
        .TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
        .TLS_RSA_WITH_AES_128_GCM_SHA256,
        .TLS_RSA_WITH_AES_256_GCM_SHA384,
        .TLS_RSA_WITH_AES_128_CBC_SHA,
        .TLS_RSA_WITH_AES_256_CBC_SHA,
    ]
    config.tlsConfiguration?.minimumTLSVersion = .tlsv12
    config.tlsConfiguration?.maximumTLSVersion = .tlsv13
```

Texts

myth 7: *When choosing a cross-platform language, it's either/or.*

reality: Swift interoperates with more languages than you think 👄

kabiroberai/NodeSwift



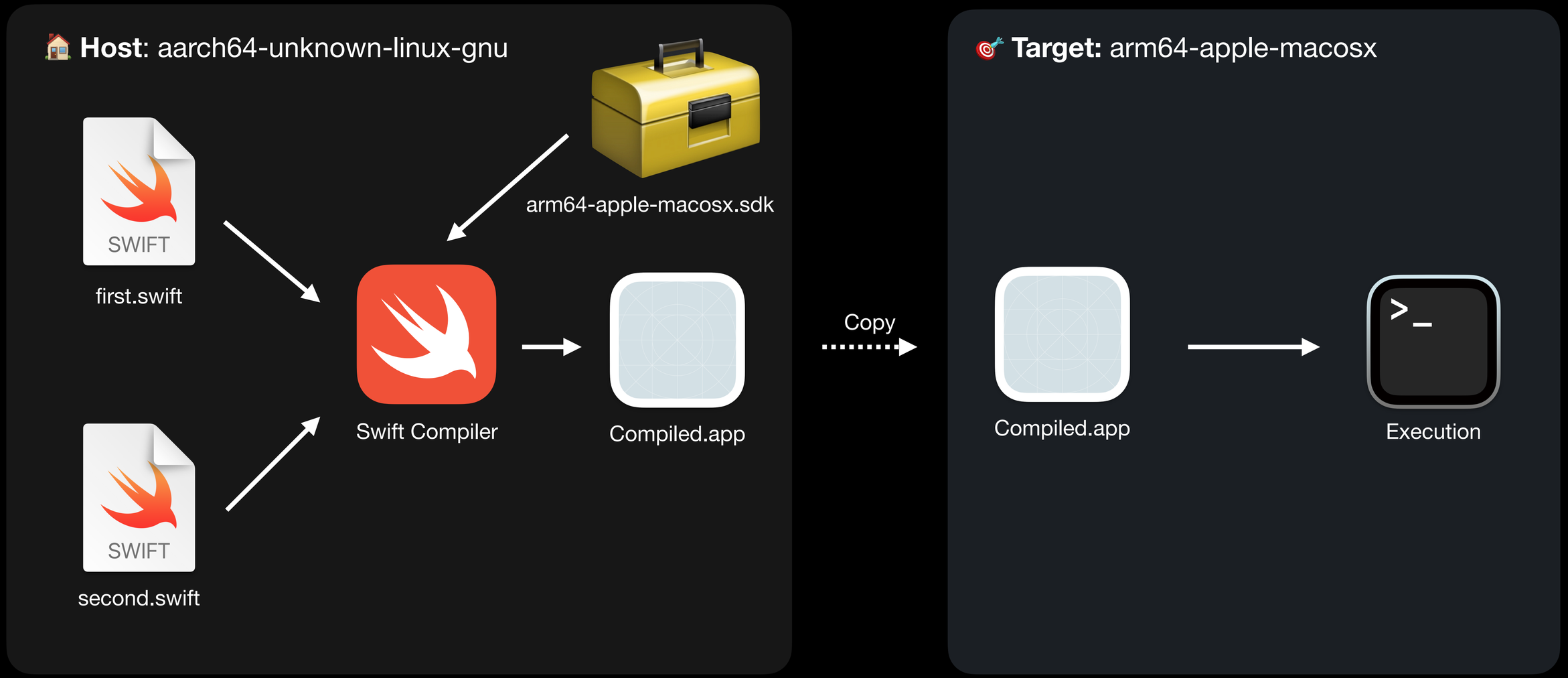

```swift
let evGroup = MultiThreadedEventLoopGroup(numberOfThreads: System.coreCount)

@NodeClass final class SwiftFetch {
    private static var retryTimeout: TimeInterval = 180

    private static var connectionTimeout: Int64 = 60

    let queue: NodeAsyncQueue

    @NodeConstructor init() throws {
        self.queue = try NodeAsyncQueue(label: "http-stream-callback-queue")
    }

    @NodeMethod public func request(url: String, options: [String:
        NodeValue]?) async throws -> NodeValueConvertible {
        do {
            return try await Self.internalRequest(url: url, options:
                options) { response in
                let byteBuffer = try await response.body.collect(upTo: 1024
                    * 1024 * 100) // up to 100MB

                return await [
                    "body": Data(buffer: byteBuffer),
                    "statusCode": Int(response.status.code),
                    "headers": Self.mapHeaders(response.headers)
```

Texts

interlude

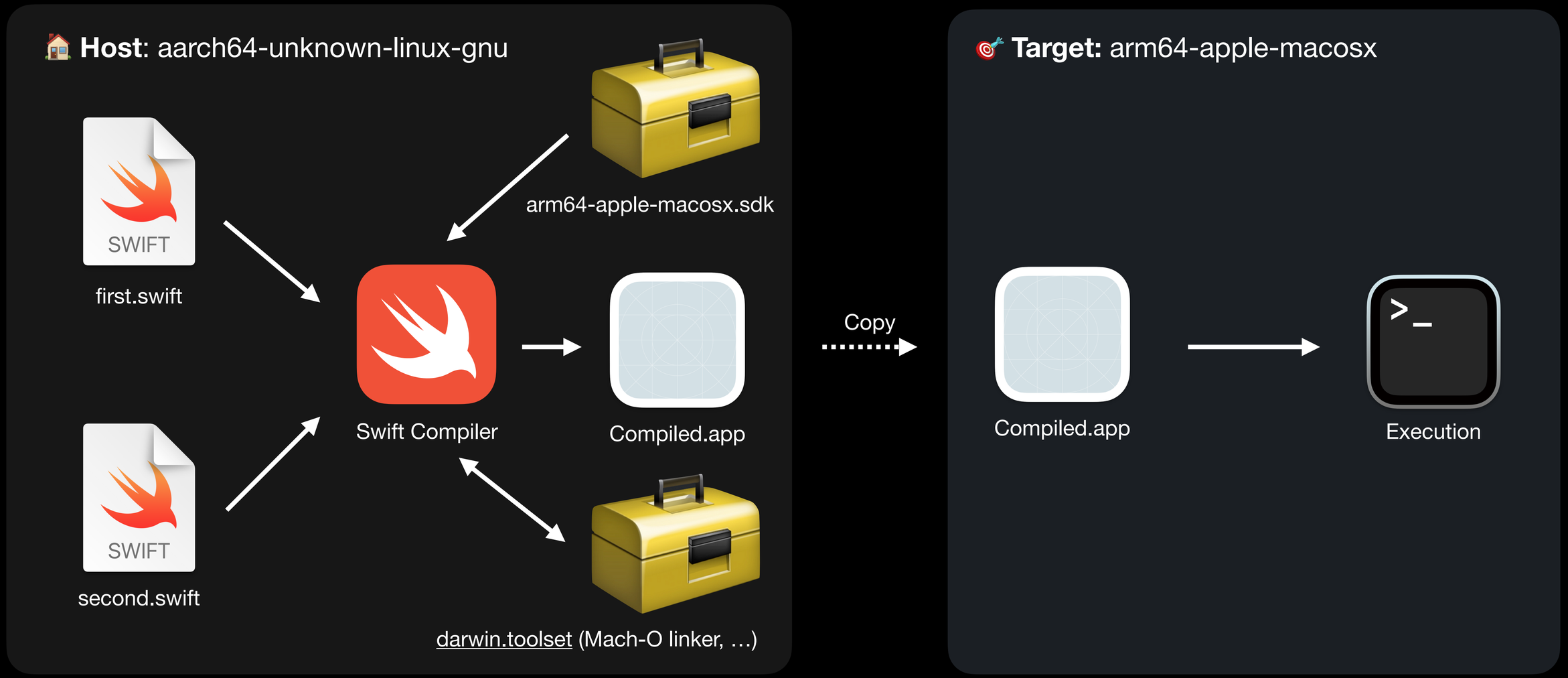
host = **Linux**
target = **macOS**?

interlude: **more theory** 👨‍🏫



from Linux to macOS?

interlude: **more theory** 👨‍🏫

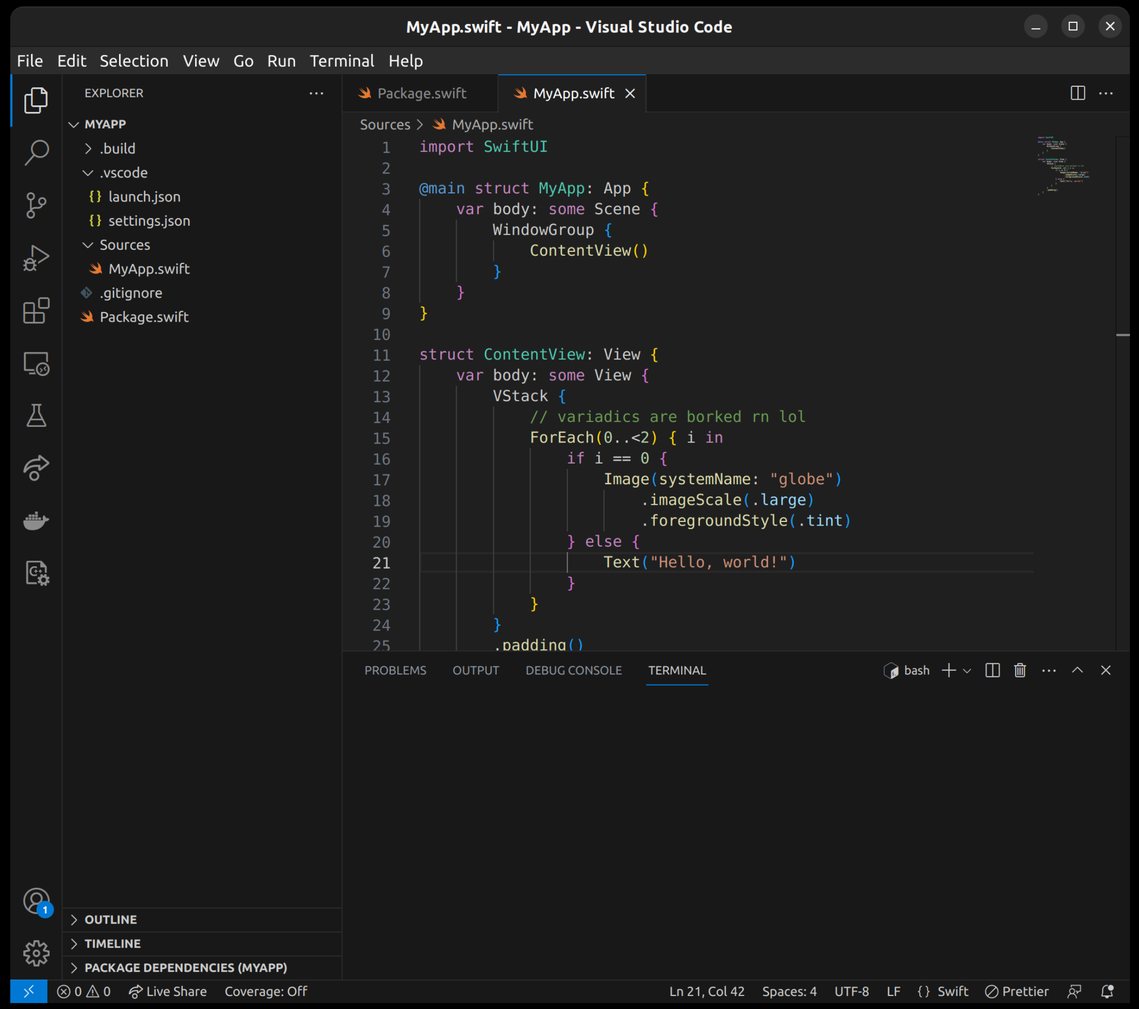


from Linux to macOS?

myth 8: *macOS Swift apps can only be built on macOS*

myth 8: *macOS Swift apps can only be built on macOS*

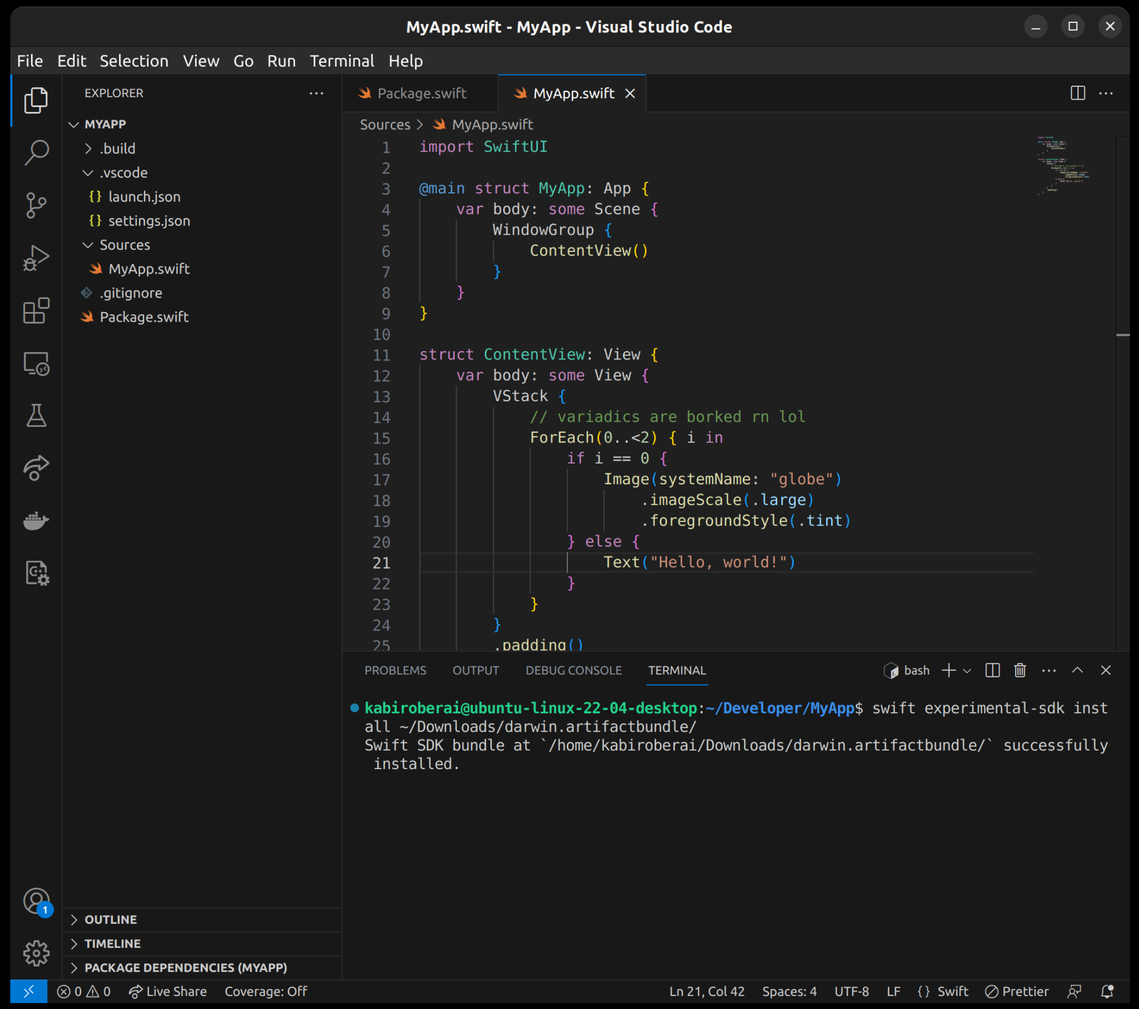
reality: Cross-compilation doesn't discriminate ⚖️



kabiroberai/swift-sdk-darwin

myth 8: *macOS Swift apps can only be built on macOS*

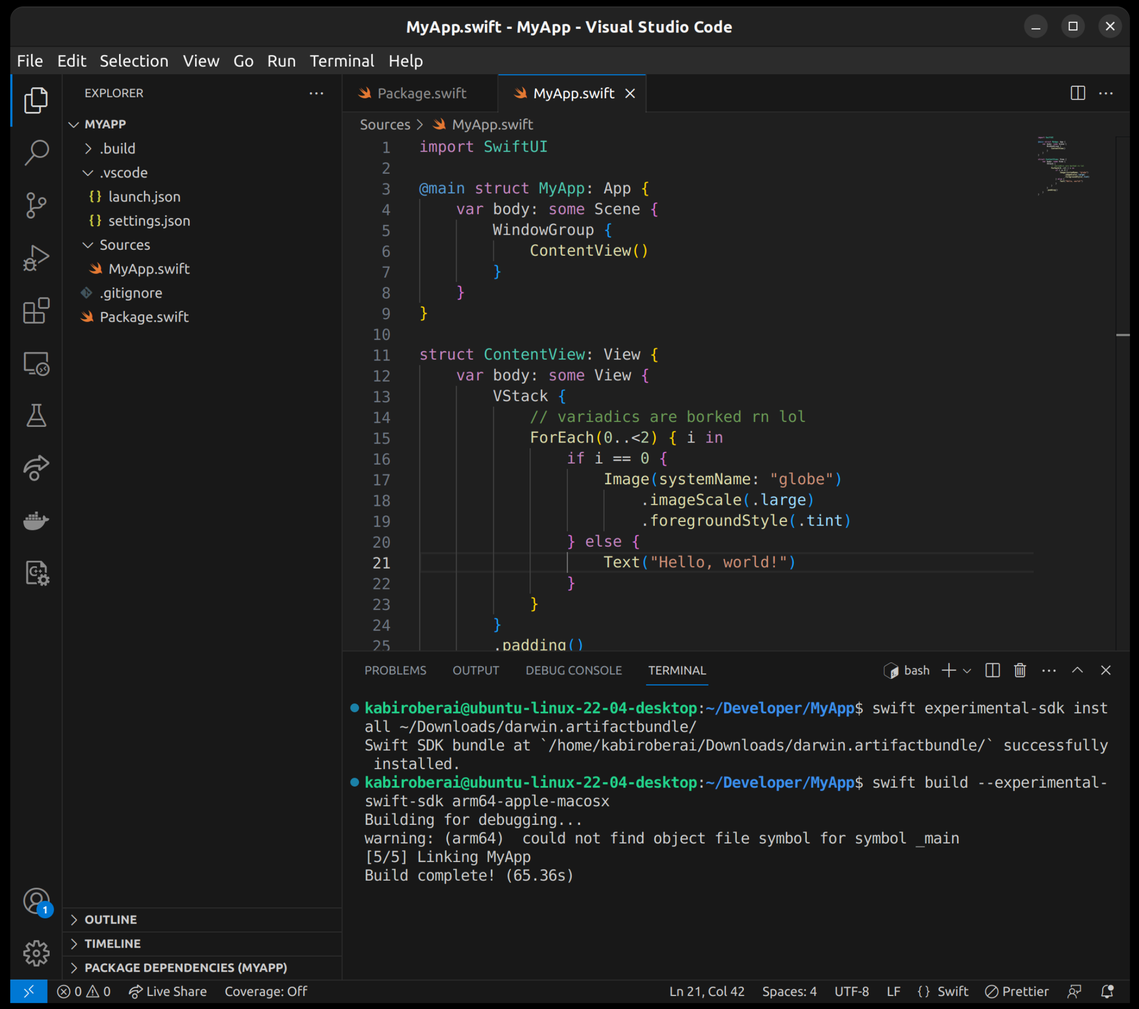
reality: Cross-compilation doesn't discriminate ⚖️



kabiroberai/swift-sdk-darwin

myth 8: *macOS Swift apps can only be built on macOS*

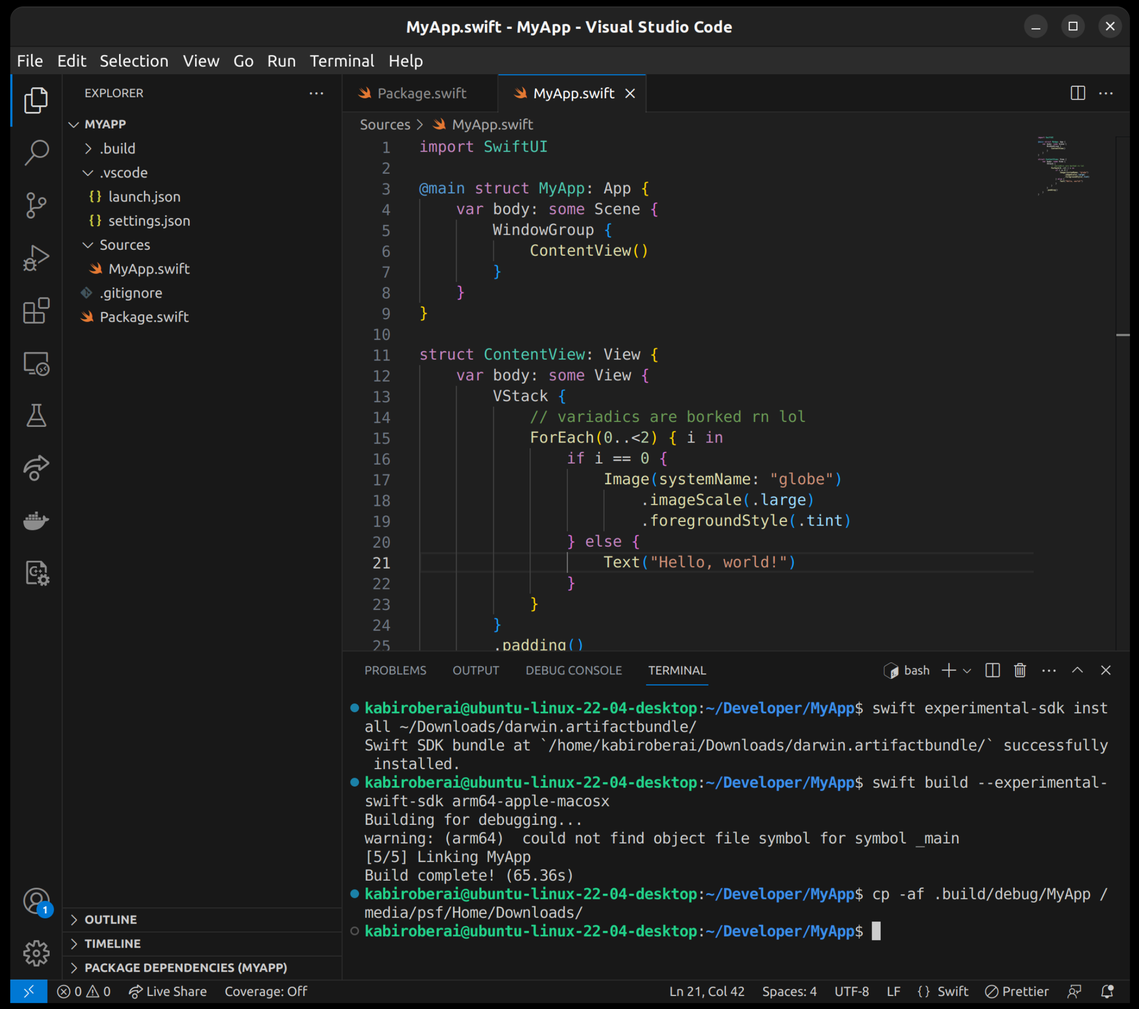
reality: Cross-compilation doesn't discriminate ⚖️



kabiroberai/swift-sdk-darwin

myth 8: *macOS Swift apps can only be built on macOS*

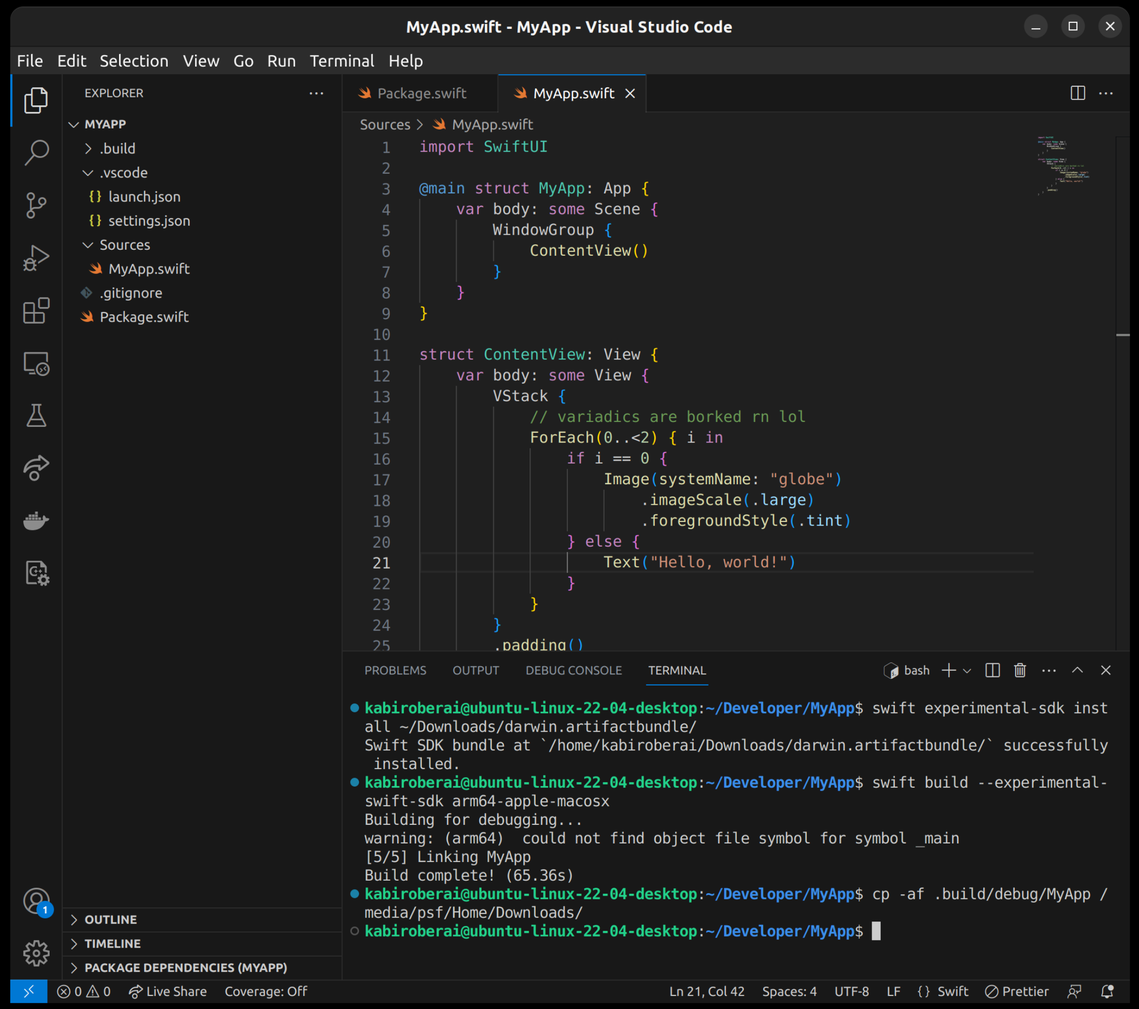
reality: Cross-compilation doesn't discriminate ⚖️

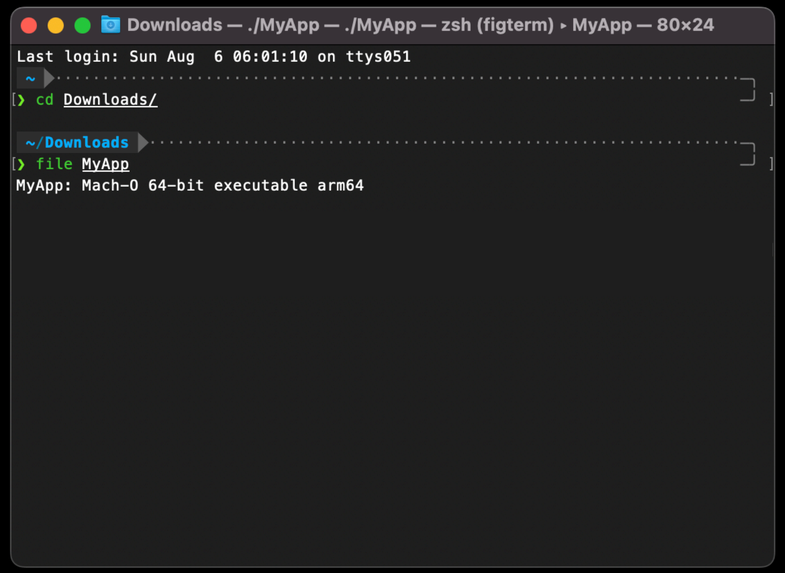


kabiroberai/swift-sdk-darwin

myth 8: *macOS Swift apps can only be built on macOS*

reality: Cross-compilation doesn't discriminate ⚖️

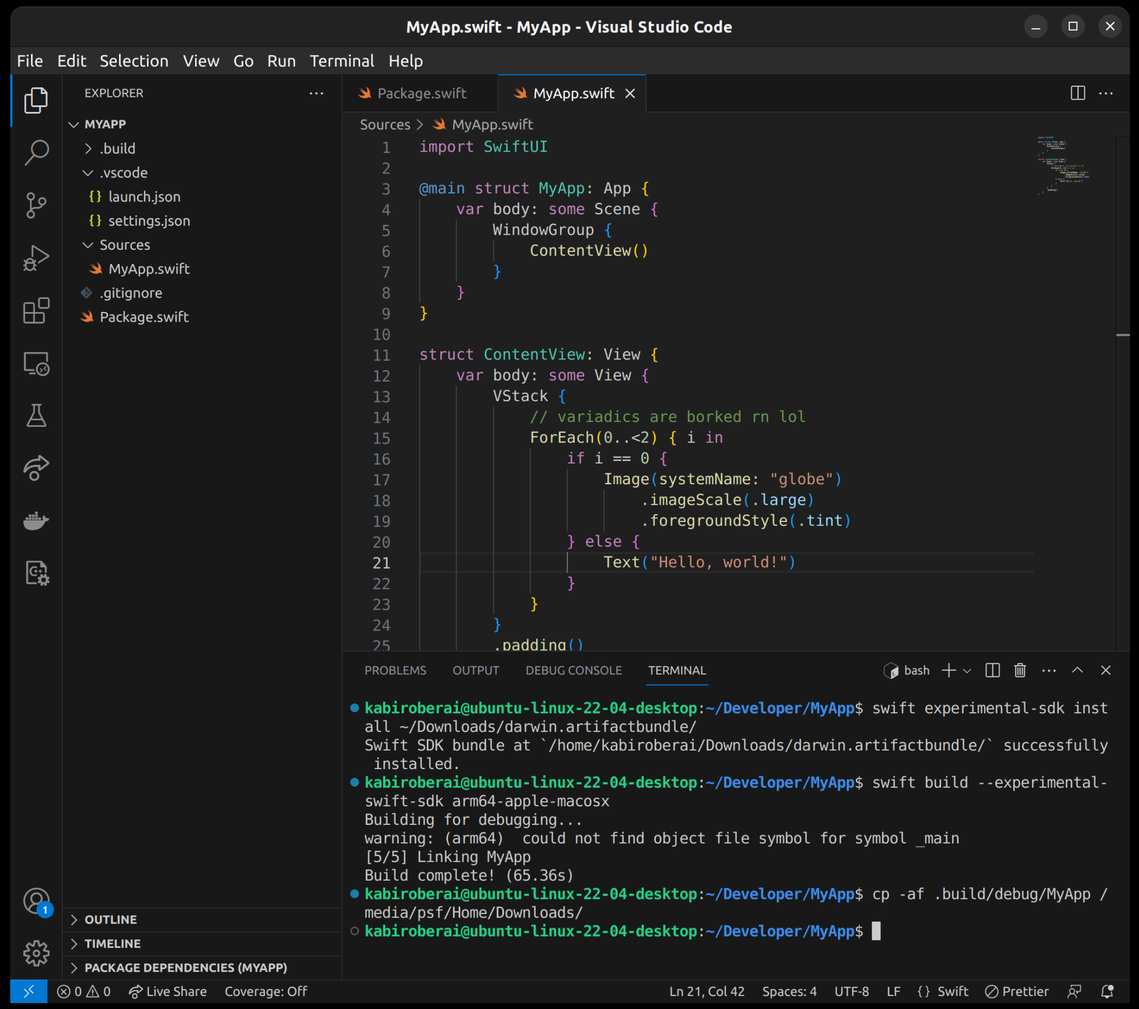




kabiroberai/swift-sdk-darwin

myth 8: *macOS Swift apps can only be built on macOS*

reality: Cross-compilation doesn't discriminate ⚖️

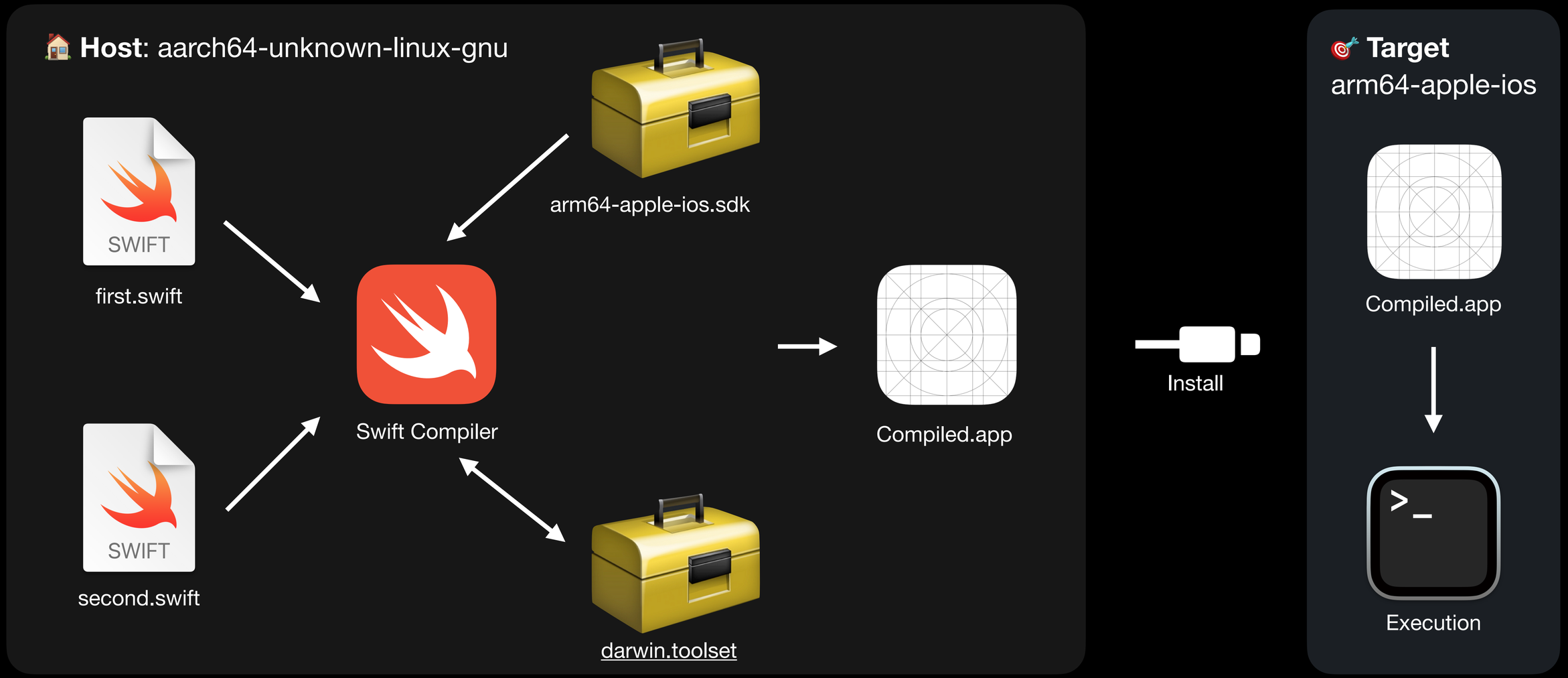


kabiroberai/swift-sdk-darwin

interlude

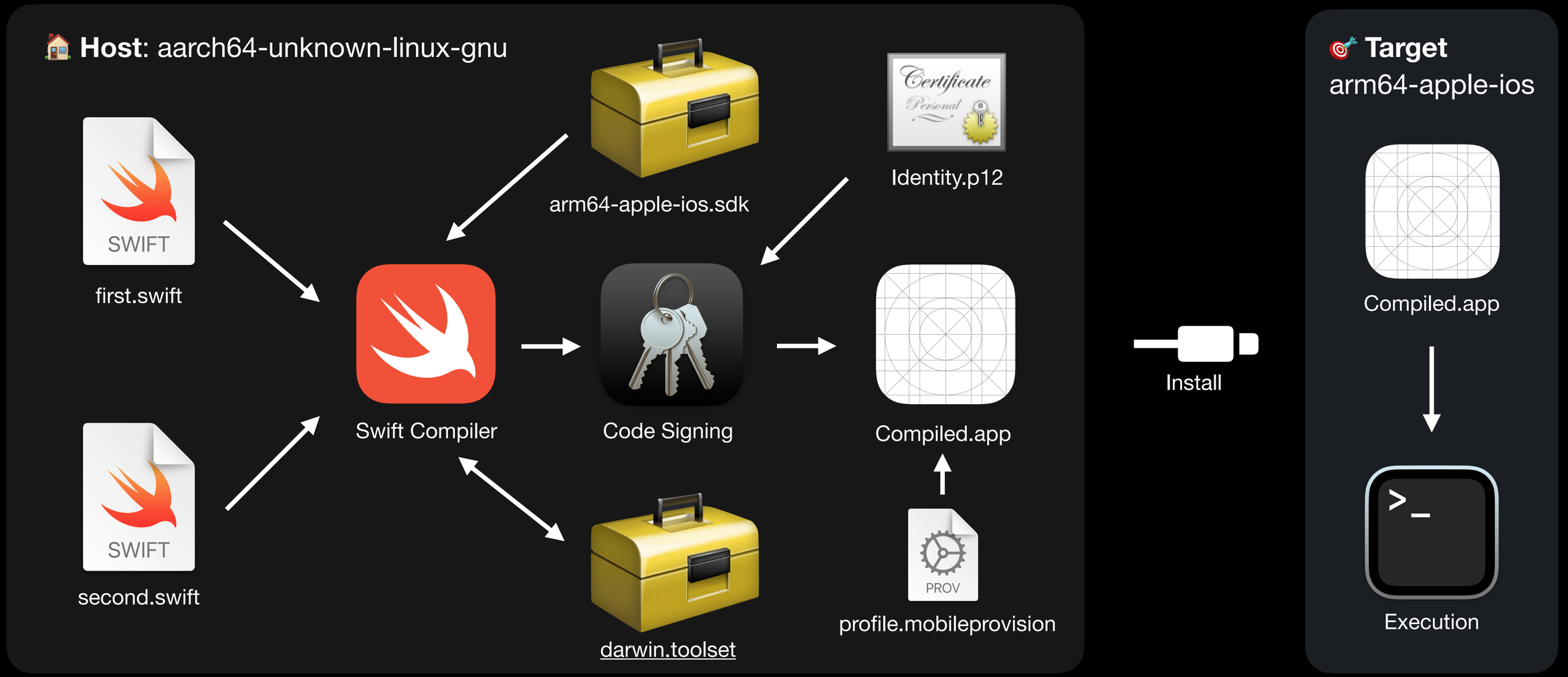
host = **Linux**
target = **iOS**?!

interlude: **even more theory** 🧠



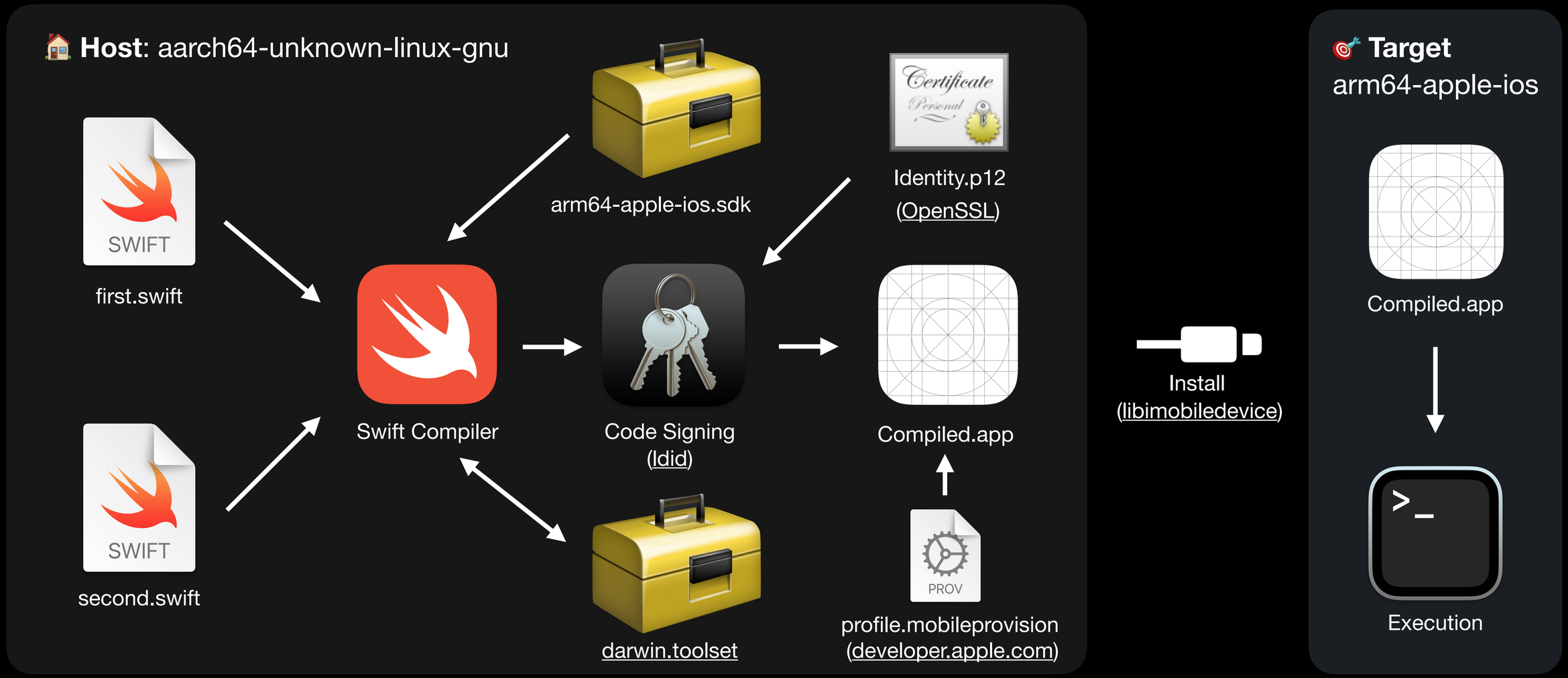
from Linux to iOS

interlude: **even more theory** 🧠



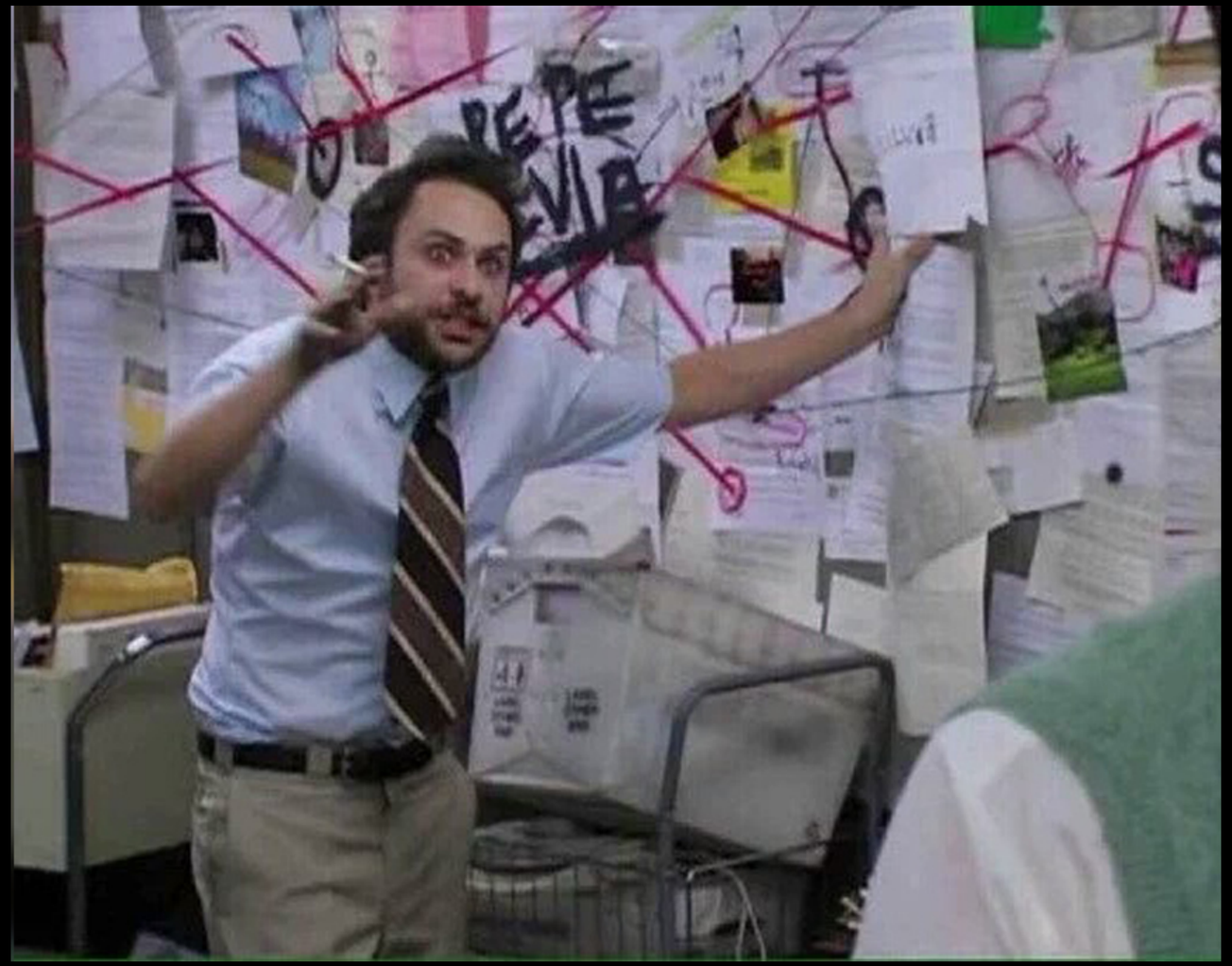
from Linux to iOS

interlude: **even more theory** 🧠



from Linux to iOS

and now,,



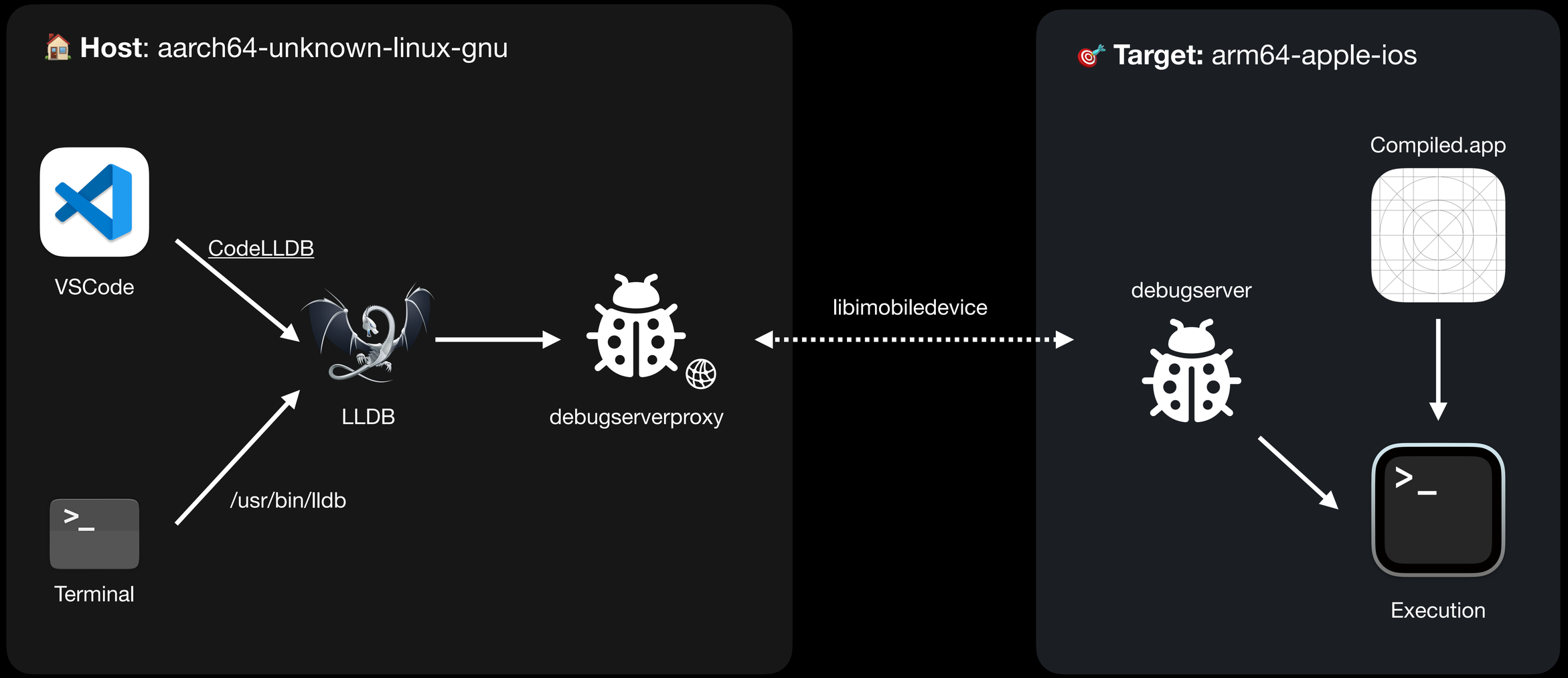
myth 9: *iOS Swift apps can only be built on macOS*

myth 9: *iOS Swift apps can only be built on macOS*

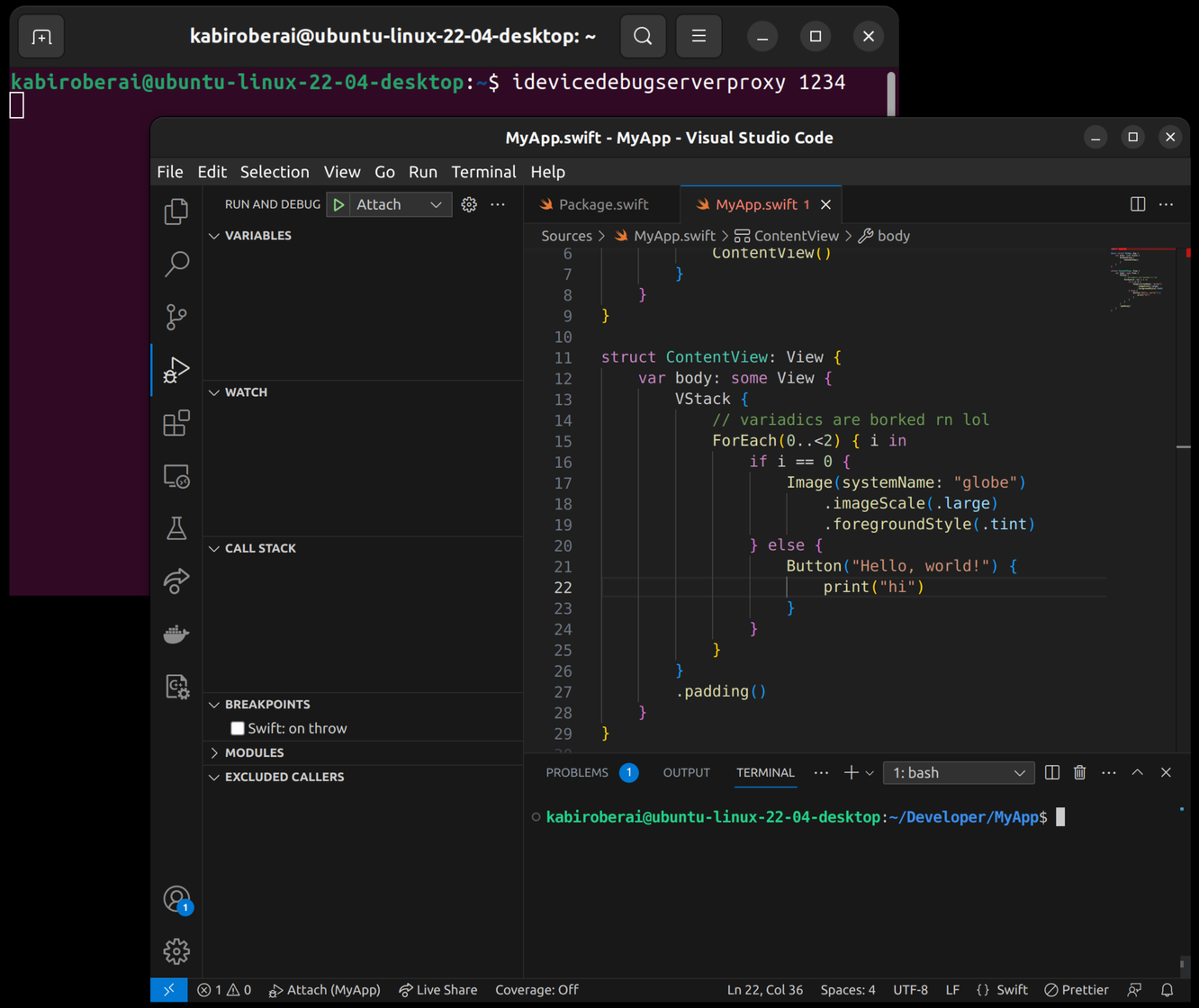
reality: Cross-compilation REALLY doesn’t discriminate 😇

demo 📸

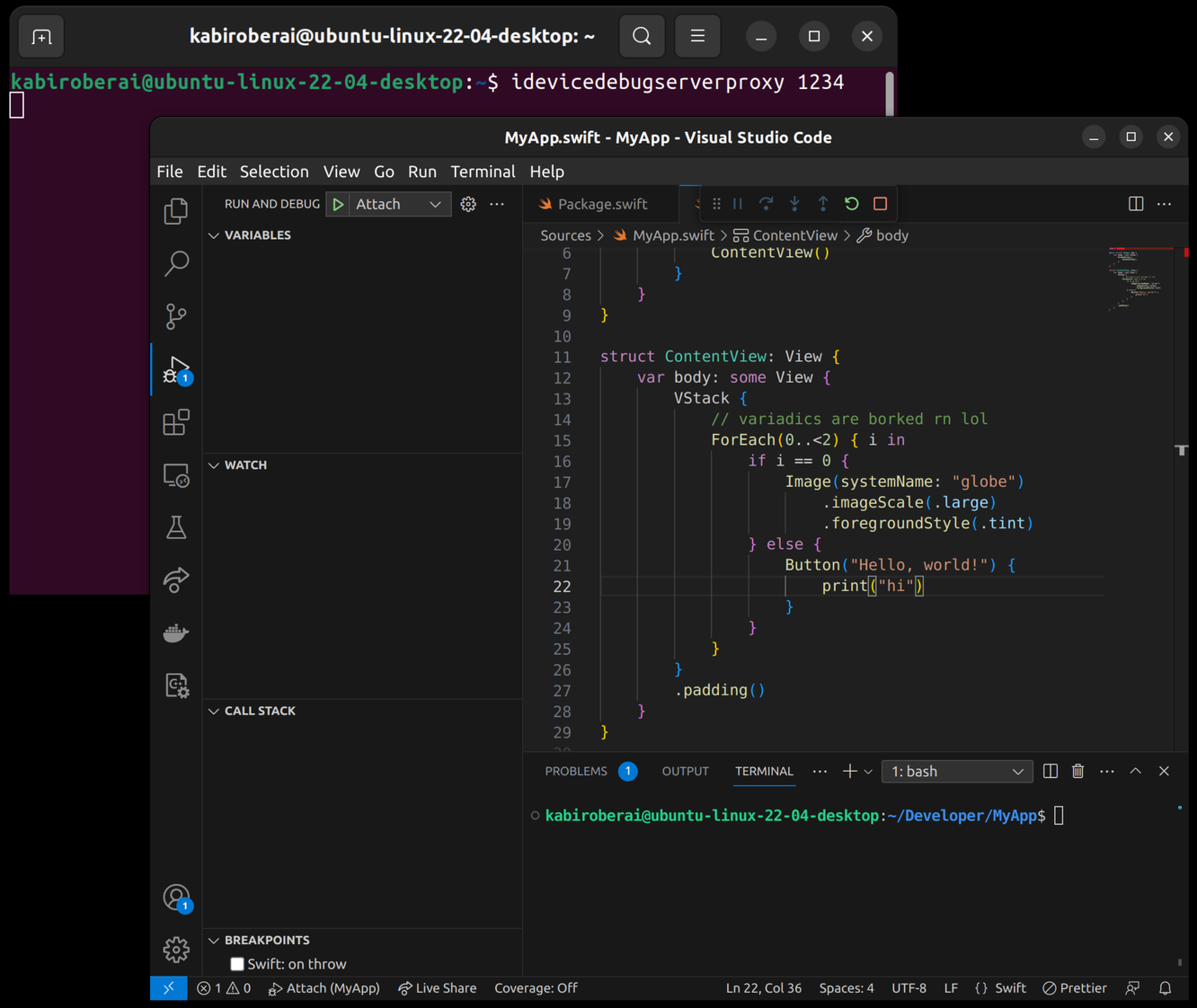
bonus: **remote debugging**



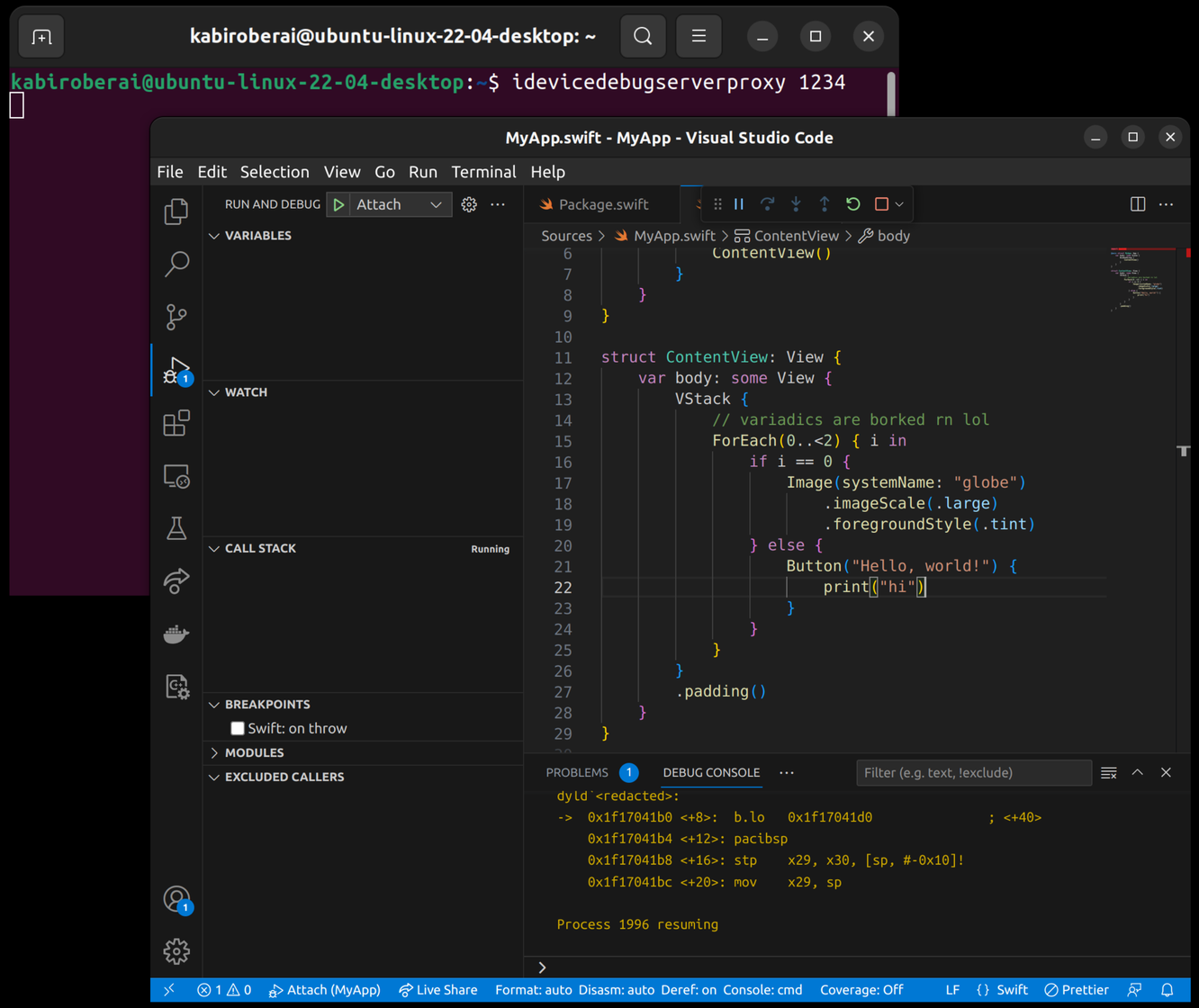
bonus: **remote debugging**

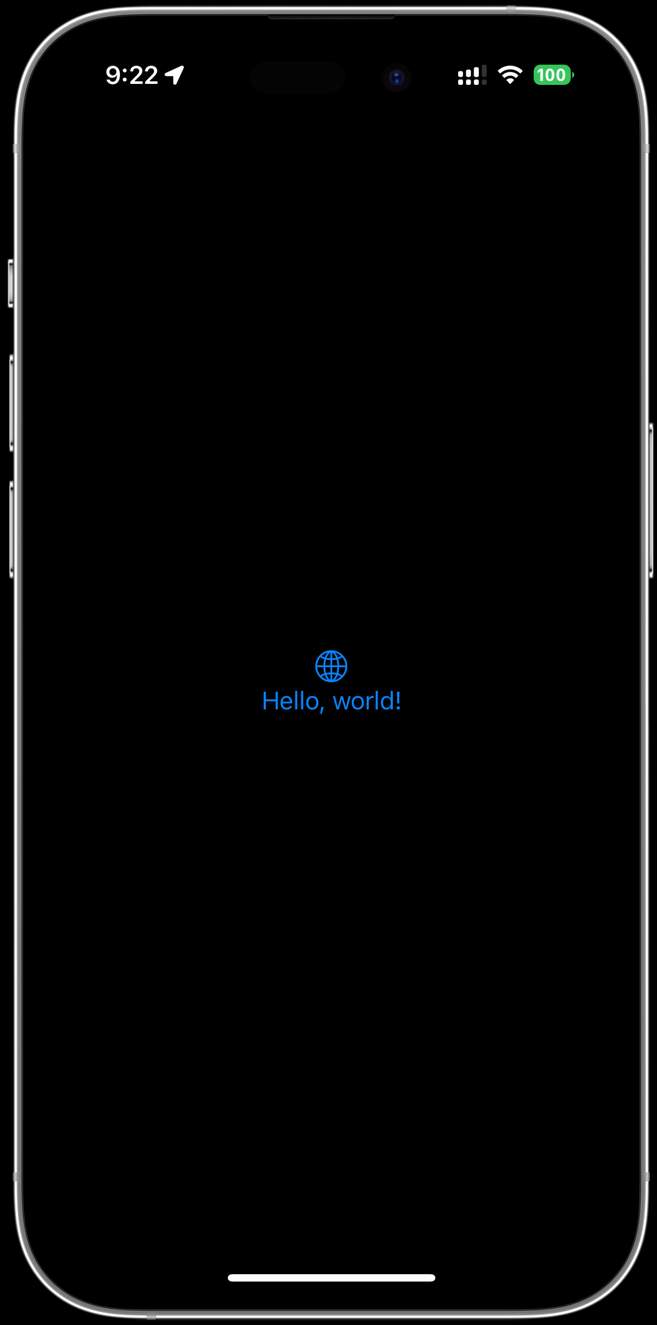


bonus: **remote debugging**

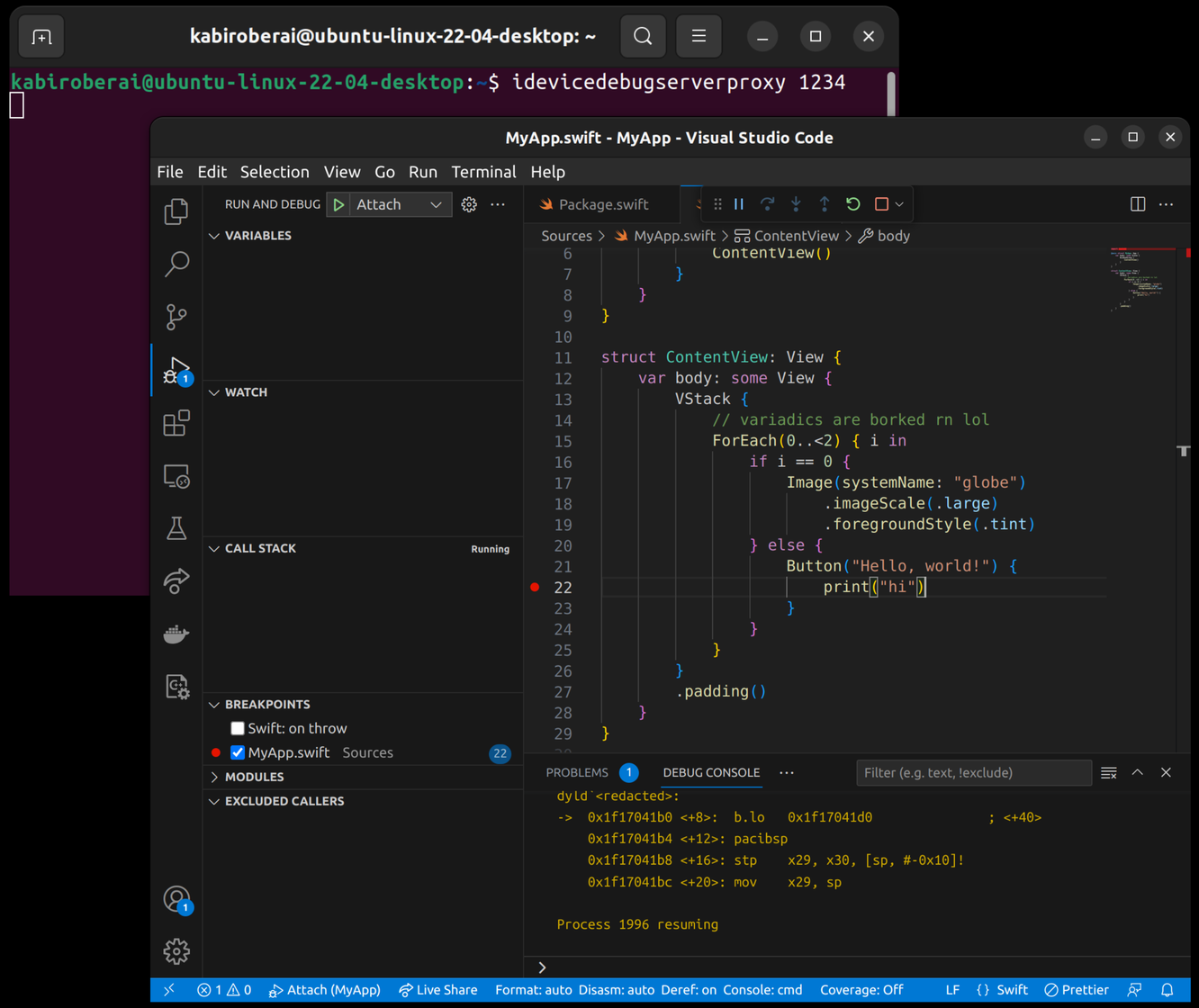


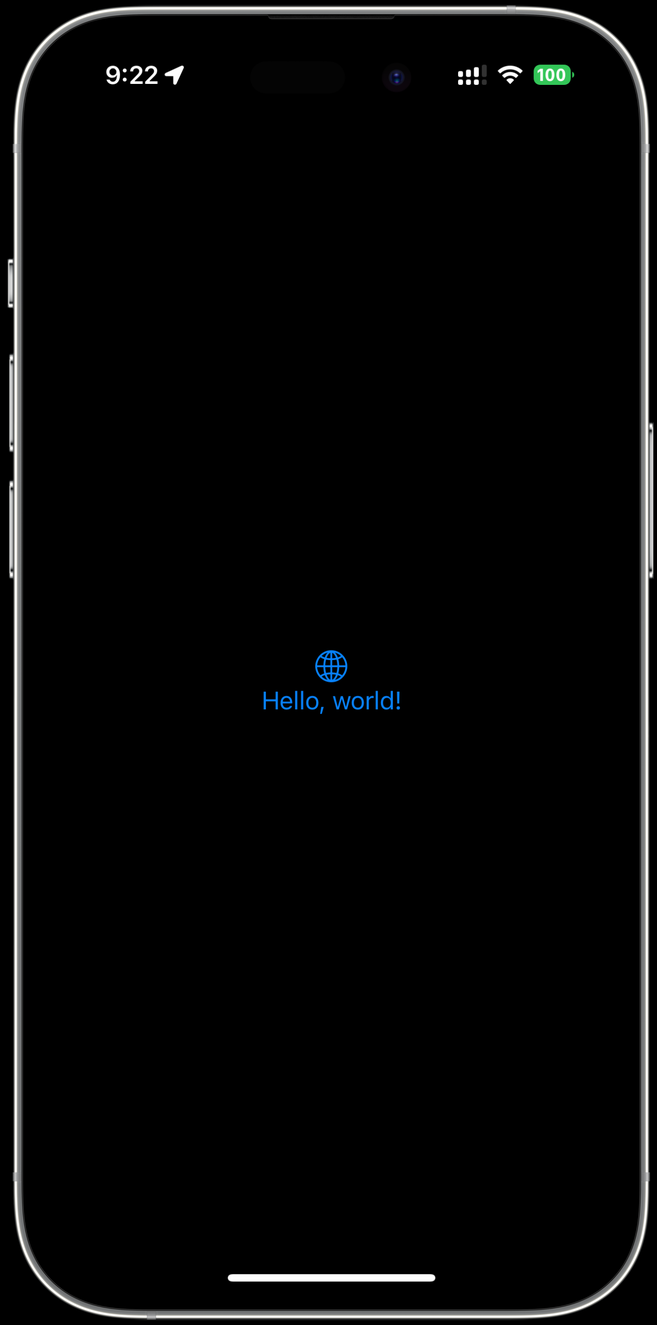
bonus: **remote debugging**



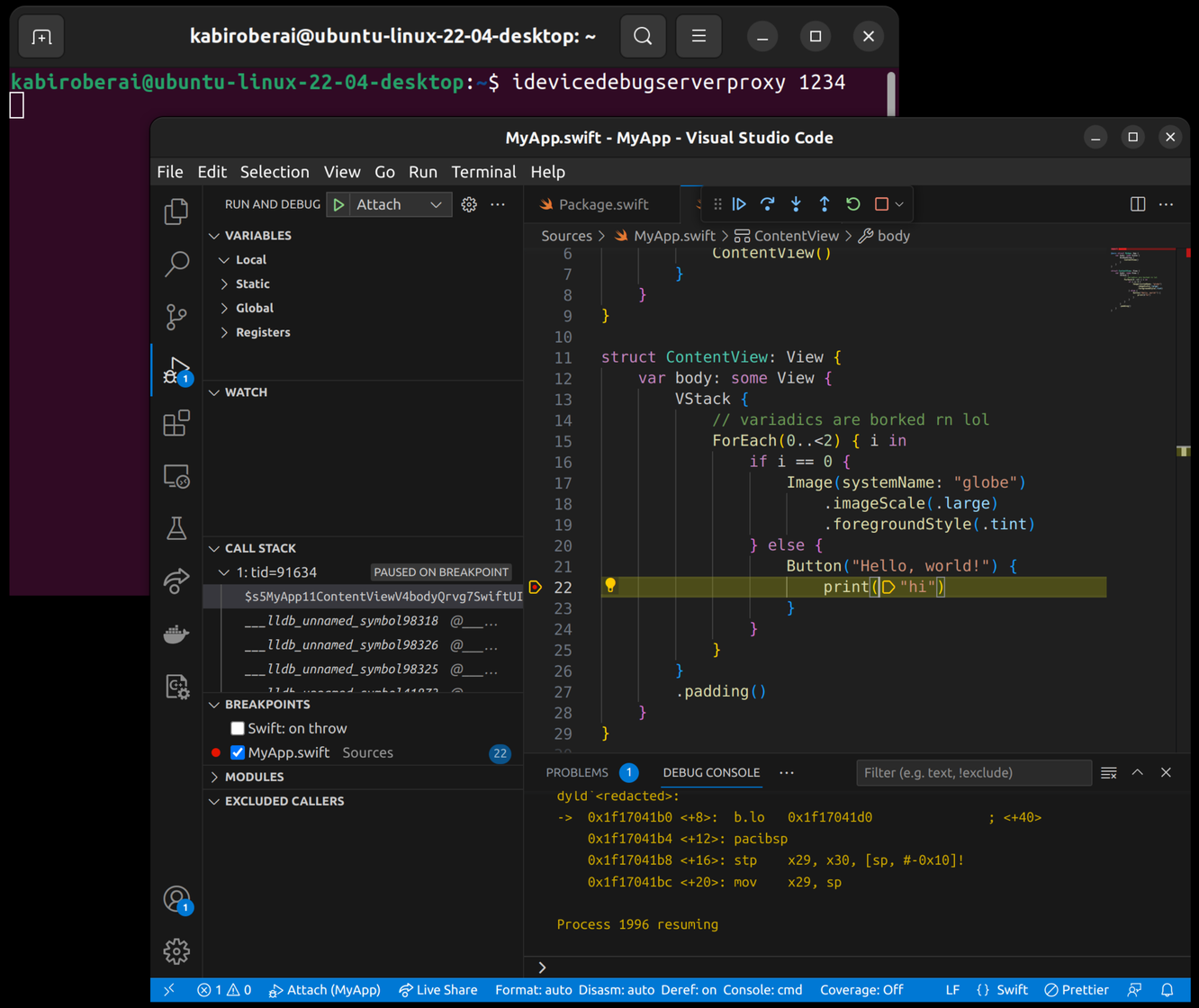


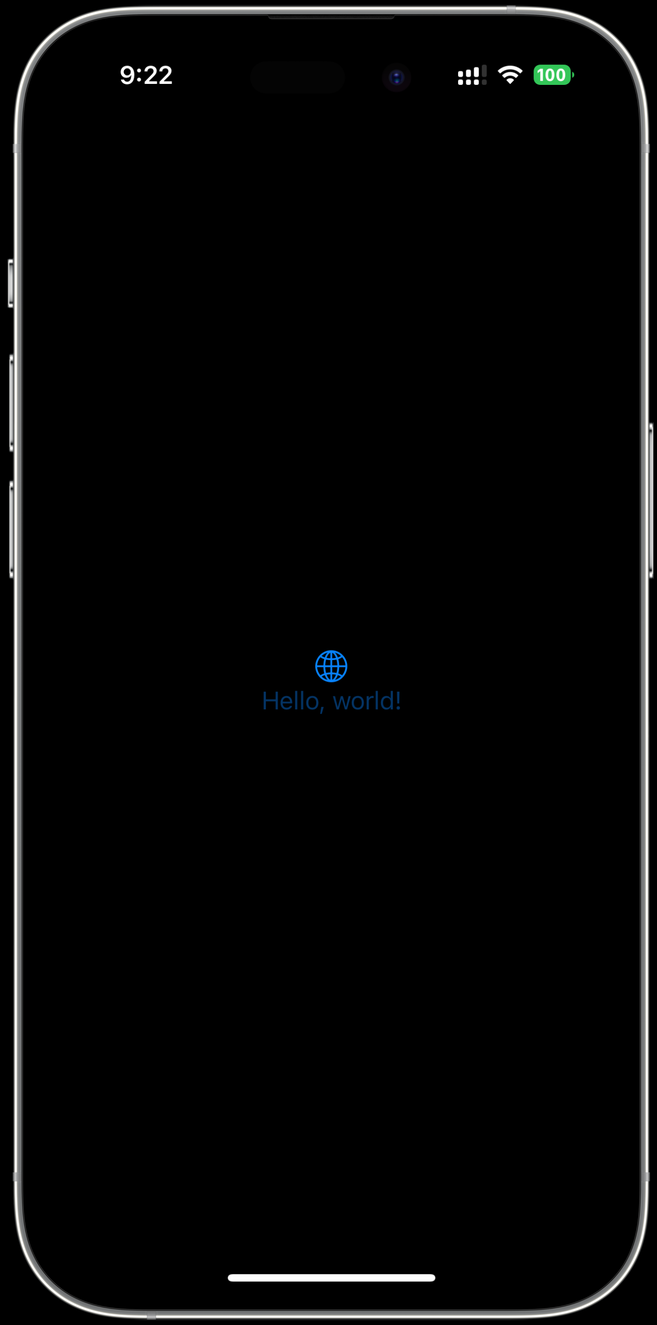
bonus: **remote debugging**





bonus: **remote debugging**





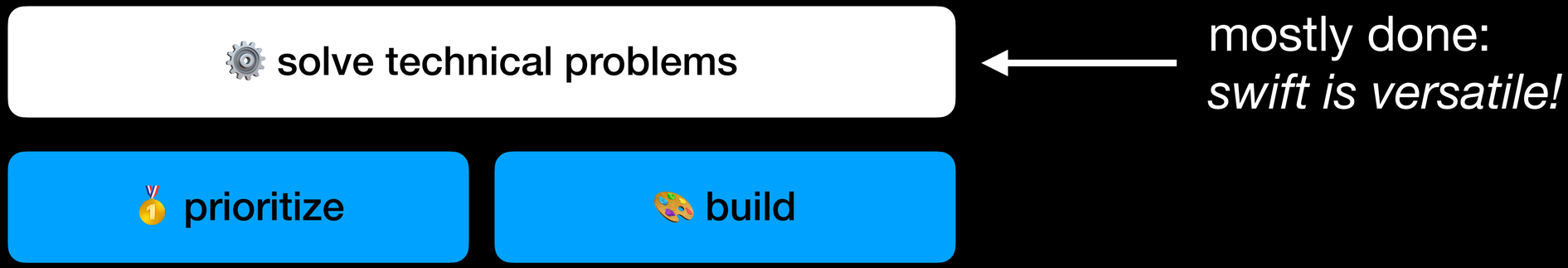
conclusion
three simple steps to world domination

⚙️ solve technical problems

🥇 prioritize

🎨 build

conclusion
three simple steps to world domination



conclusion
three simple steps to world domination

⚙️ solve technical problems ← mostly done: *swift is versatile!*

🥇 prioritize | 🎨 build ← **your turn** 🚀

get in touch!

@kabiroberai
/in/kabiroberai
me@kabiroberai.com

slides & code
ober.ai/swiftto