Swift on Windows

Saleem Abdulrasool (@compnerd)

Swift on Windows

Saleem Abdulrasool (@compnerd)

Quid Celeri?

Quid Celeri?

What is Swift?

```
import WinRT

@main
struct Asynchronous {
  public static func main() async throws {
    try RoInitialize()
    let controller =
        try Windows.System.DispatcherQueueController.CreateOnDedicatedThread()
    try await controller.ShutdownQueue()
  }
}
```

Source Code: https://github.com/compnerd/SwiftWinRT

Quid Celeri?

What is Swift?

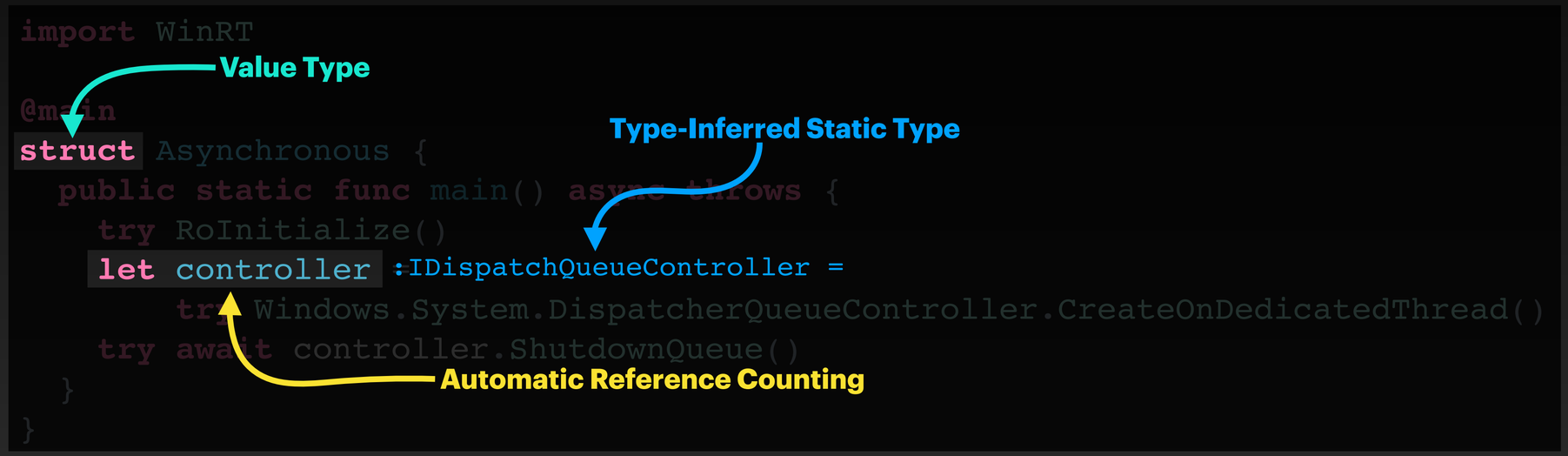
- Modern

Source Code: https://github.com/compnerd/SwiftWinRT

Quid Celeri?

What is Swift?

- Modern
- Type Safe



Source Code: https://github.com/compnerd/SwiftWinRT

Quid Celeri?

What is Swift?

- Modern
- Type Safe
- Multi-paradigm
- General Purpose

```
import WinRT

@main
struct Asynchronous {
  public static func main() async throws {
    try RoInitialize()
    let controller =
        try Windows.System.DispatcherQueueController.CreateOnDedicatedThread()
    try await controller.ShutdownQueue()
  }
}
```

Source Code: https://github.com/compnerd/SwiftWinRT

Quid Celeri?

What is Swift?

- Modern
- Type Safe
- Multi-paradigm
- General Purpose

```
import WinRT

@main
struct Asynchronous {
  public static func main() async throws {
    try RoInitialize()
    let controller =
        try Windows.System.DispatcherQueueController.CreateOnDedicatedThread()
    try await controller.ShutdownQueue()
  }
}
```

Source Code: https://github.com/compnerd/SwiftWinRT

- Application Programming

Quid Celeri?

What is Swift?

- Modern
- Type Safe
- Multi-paradigm
- General Purpose

```
import WinRT

@main
struct Asynchronous {
  public static func main() async throws {
    try RoInitialize()
    let controller =
        try Windows.System.DispatcherQueueController.CreateOnDedicatedThread()
    try await controller.ShutdownQueue()
  }
}
```

Source Code: https://github.com/compnerd/SwiftWinRT

- Application Programming
- Systems Programming

Wrong Channel?

Wrong Channel?

Swift on Windows is a LLVM Distribution

Wrong Channel?

Swift on Windows is a LLVM Distribution

- LLVM Based

Wrong Channel?

Swift on Windows is a LLVM Distribution

- LLVM Based
- Supports Multiple Languages
  - Swift
  - C
  - C++
  - Objective-C

Wrong Channel?

Swift on Windows is a LLVM Distribution

- LLVM Based
- Supports Multiple Languages
  - Swift
  - C
  - C++
  - Objective-C
- Intended to be used as a complete toolchain

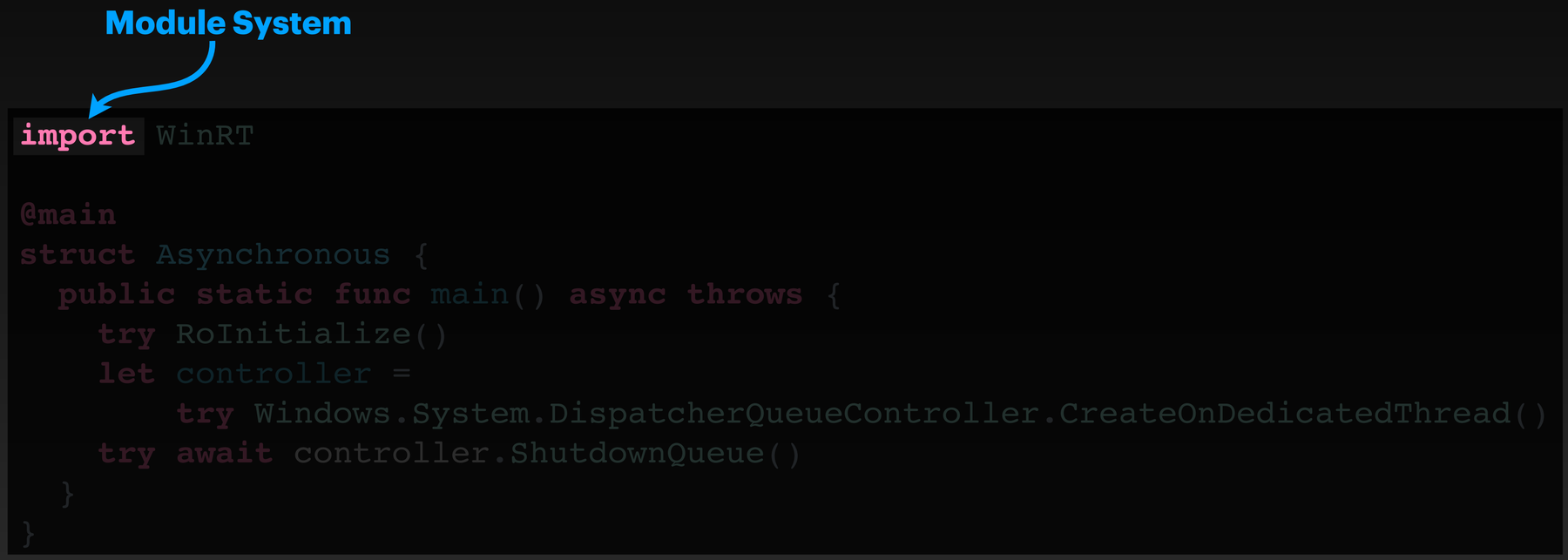
One More Thing ...

FFI

Quid Celeri?

What is Swift?

- Modern
- Type Safe
- Multi-paradigm
- General Purpose



Source Code: https://github.com/compnerd/SwiftWinRT

- Application Programming
- Systems Programming

One More Thing ...

FFI

One More Thing ...
FFI

- Compile Time FFI

One More Thing ...
FFI

- Compile Time FFI
- Embedded Clang

One More Thing ...
FFI

- Compile Time FFI
- Embedded Clang
- clang modules

Recipes

Recipes

Toolchain Hosts and Targets

Recipes

Toolchain Hosts and Targets

Host

- Windows
  - X64
  - ARM64

Recipes

Toolchain Hosts and Targets

Host

- Windows
  - X64
  - ARM64

Target

- Windows
  - X86, X64, ARM, ARM64
- Android
  - X86, X64, ARM, ARM64
- Linux
  - X86, X64, ARM, ARM64
- WASI
  - WASM

Elastic Sky

Elastic Sky

Build Statistics

Elastic Sky
Build Statistics

- Azure Pipelines

Elastic Sky
Build Statistics

- Azure Pipelines
- ~30 min to clone sources (~5.5 GB)

Elastic Sky
Build Statistics

- Azure Pipelines
- ~30 min to clone sources (~5.5 GB)
- Toolchain is ~10 GB w/o Debug Info

Elastic Sky
Build Statistics

- Azure Pipelines
- ~30 min to clone sources (~5.5 GB)
- Toolchain is ~10 GB w/o Debug Info
  - ~50 GB for a Release Build with Debug Info

Elastic Sky
Build Statistics

- Azure Pipelines
- ~30 min to clone sources (~5.5 GB)
- Toolchain is ~10 GB w/o Debug Info
  - ~50 GB for a Release Build with Debug Info
- SDK is ~500 MB for a Release Build

Elastic Sky
Build Statistics

- Azure Pipelines
- ~30 min to clone sources (~5.5 GB)
- Toolchain is ~10 GB w/o Debug Info
  - ~50 GB for a Release Build with Debug Info
- SDK is ~500 MB for a Release Build
- Developer Tools are ~200 MB for a Release Build

Elastic Sky
Build Statistics

- Azure Pipelines
- ~30 min to clone sources (~5.5 GB)
- Toolchain is ~10 GB w/o Debug Info
  - ~50 GB for a Release Build with Debug Info
- SDK is ~500 MB for a Release Build
- Developer Tools are ~200 MB for a Release Build
- ~8-hours for a single stage, non-LTO, non-PGO build

Elastic Sky
Build Statistics

- Azure Pipelines
- ~30 min to clone sources (~5.5 GB)
- Toolchain is ~10 GB w/o Debug Info
  - ~50 GB for a Release Build with Debug Info
- SDK is ~500 MB for a Release Build
- Developer Tools are ~200 MB for a Release Build
- ~8-hours for a single stage, non-LTO, non-PGO build
- Unified Installer ~500 MB

Flight Manifest

Flight Manifest

Windows Swift Toolchain Distribution Contents

Flight Manifest

Windows Swift Toolchain Distribution Contents

- Toolchain

Flight Manifest

Windows Swift Toolchain Distribution Contents

- Toolchain
  - clang, clang++, clang-cl, swift, ar, as, ld, db

Flight Manifest

Windows Swift Toolchain Distribution Contents

- Toolchain
  - clang, clang++, clang-cl, swift, ar, as, ld, db
  - nm, objdump, c++filt, ...

Flight Manifest

Windows Swift Toolchain Distribution Contents

- Toolchain
  - clang, clang++, clang-cl, swift, ar, as, ld, db
  - nm, objdump, c++filt, ...
  - mt, rc, ...

Flight Manifest

Windows Swift Toolchain Distribution Contents

- Toolchain
  - clang, clang++, clang-cl, swift, ar, as, ld, db
  - nm, objdump, c++filt, ...
  - mt, rc, ...
- SDK
  - Swift runtime/standard library, libdispatch, Foundation, XCTest

Flight Manifest

Windows Swift Toolchain Distribution Contents

- Toolchain
  - clang, clang++, clang-cl, swift, ar, as, ld, db
  - nm, objdump, c++filt, ...
  - mt, rc, ...
- SDK
  - Swift runtime/standard library, libdispatch, Foundation, XCTest
- Developer Tools
  - Package Manager, clangd, LSP

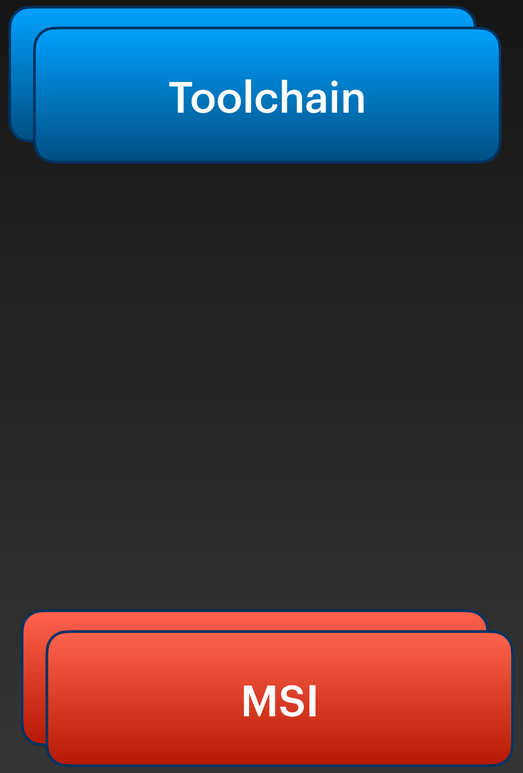
Oscillations

Oscillations

Building in Phases

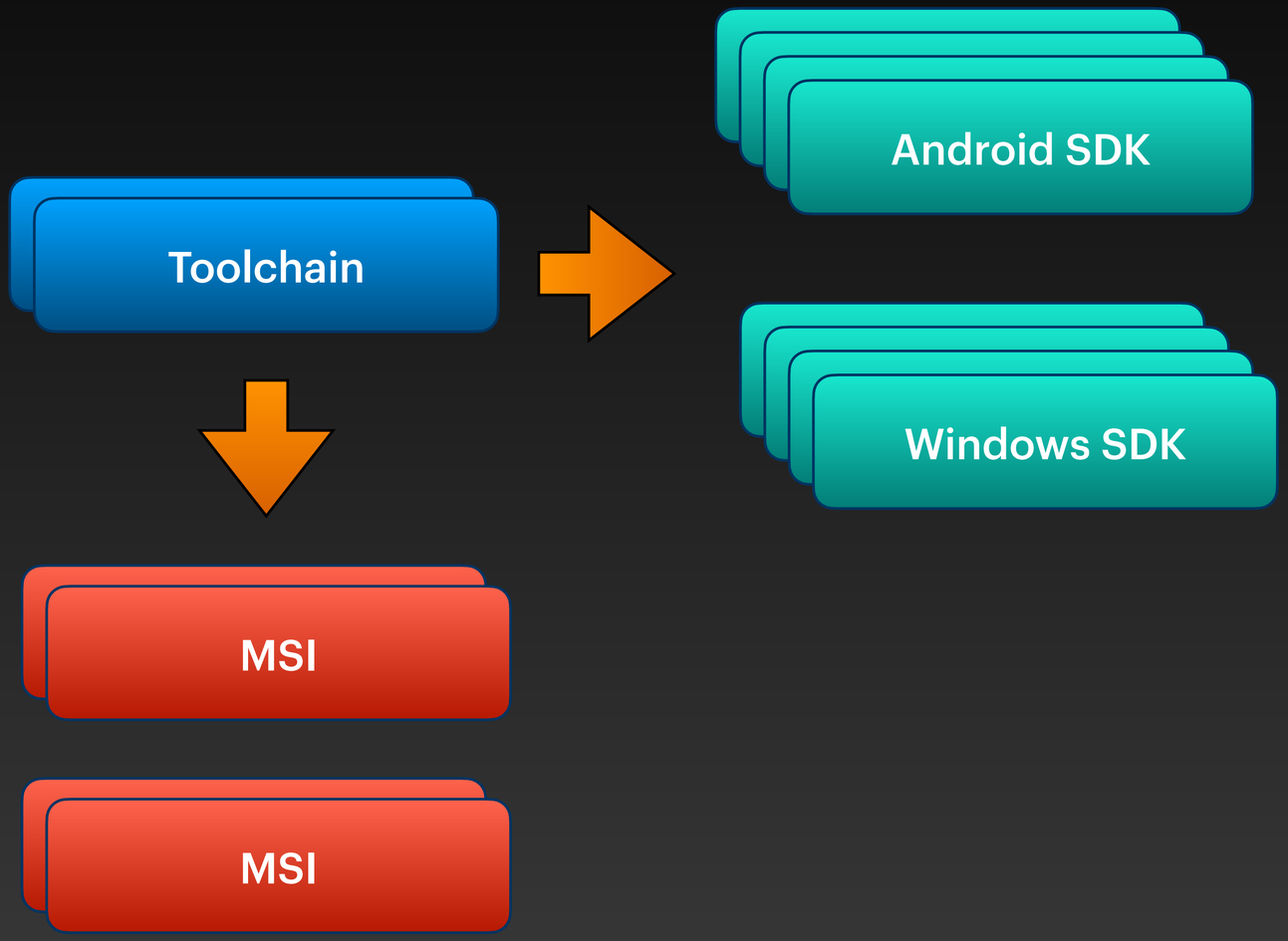
Oscillations

Building in Phases



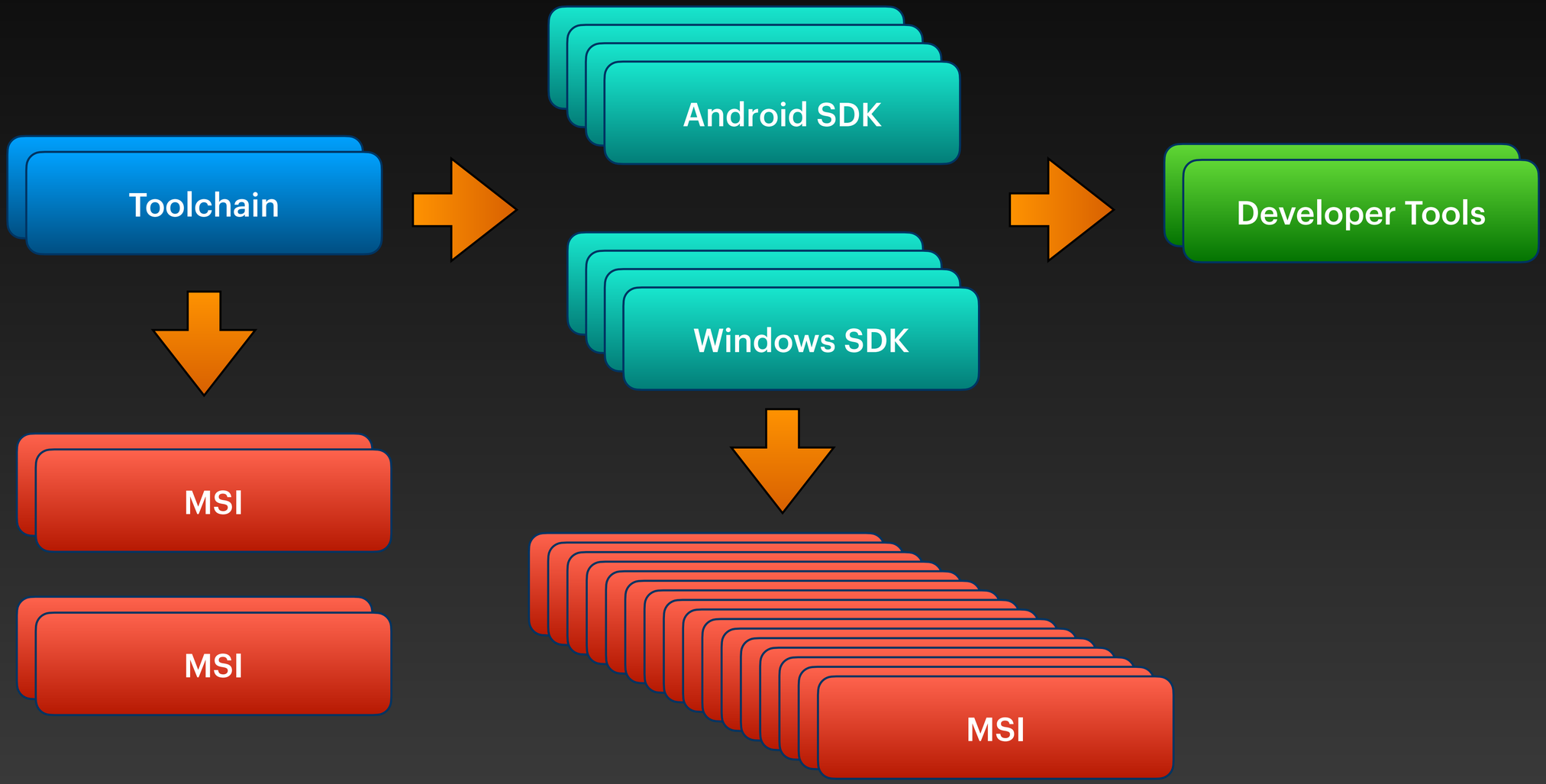
Oscillations

Building in Phases



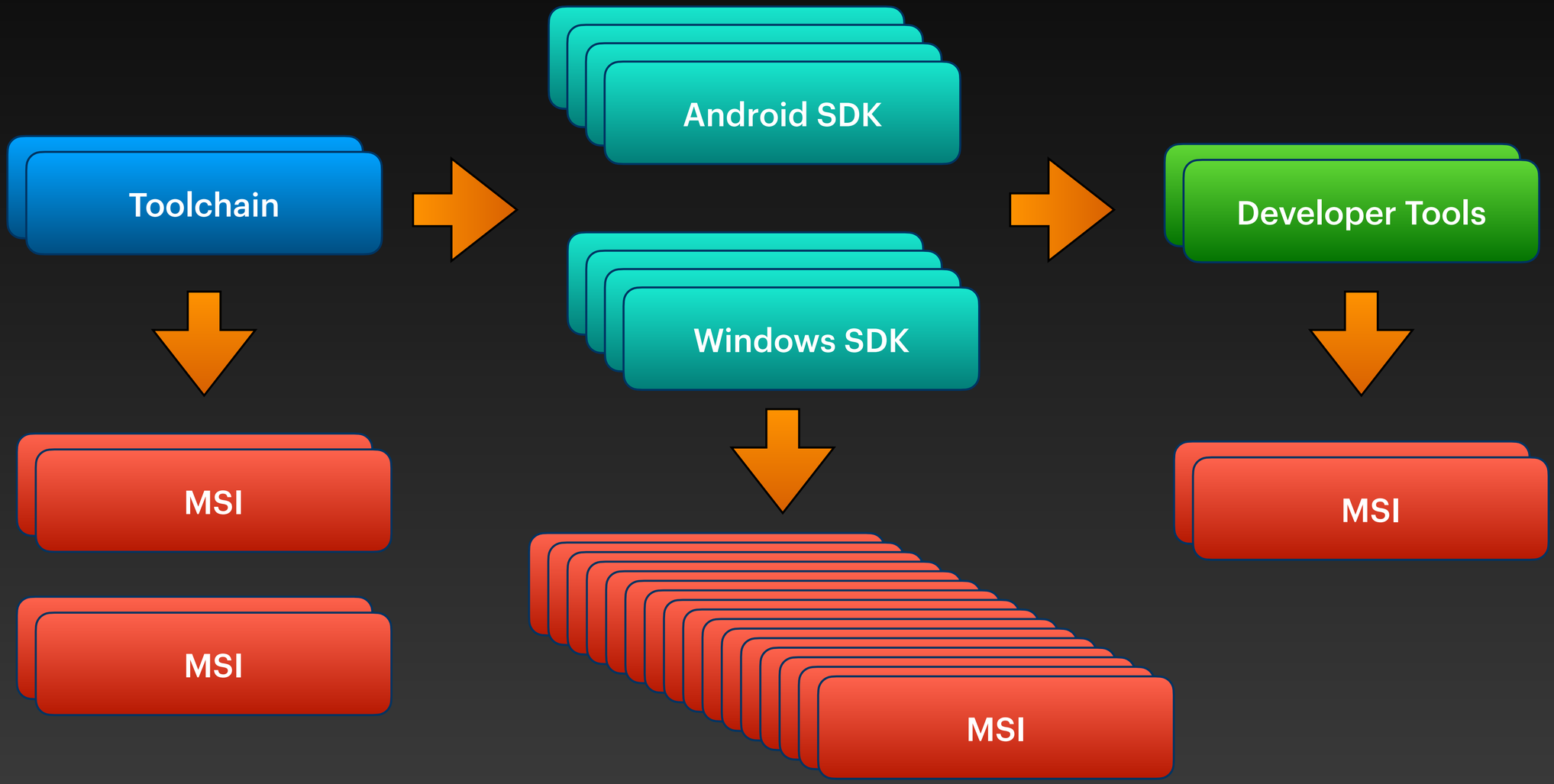
Oscillations

Building in Phases



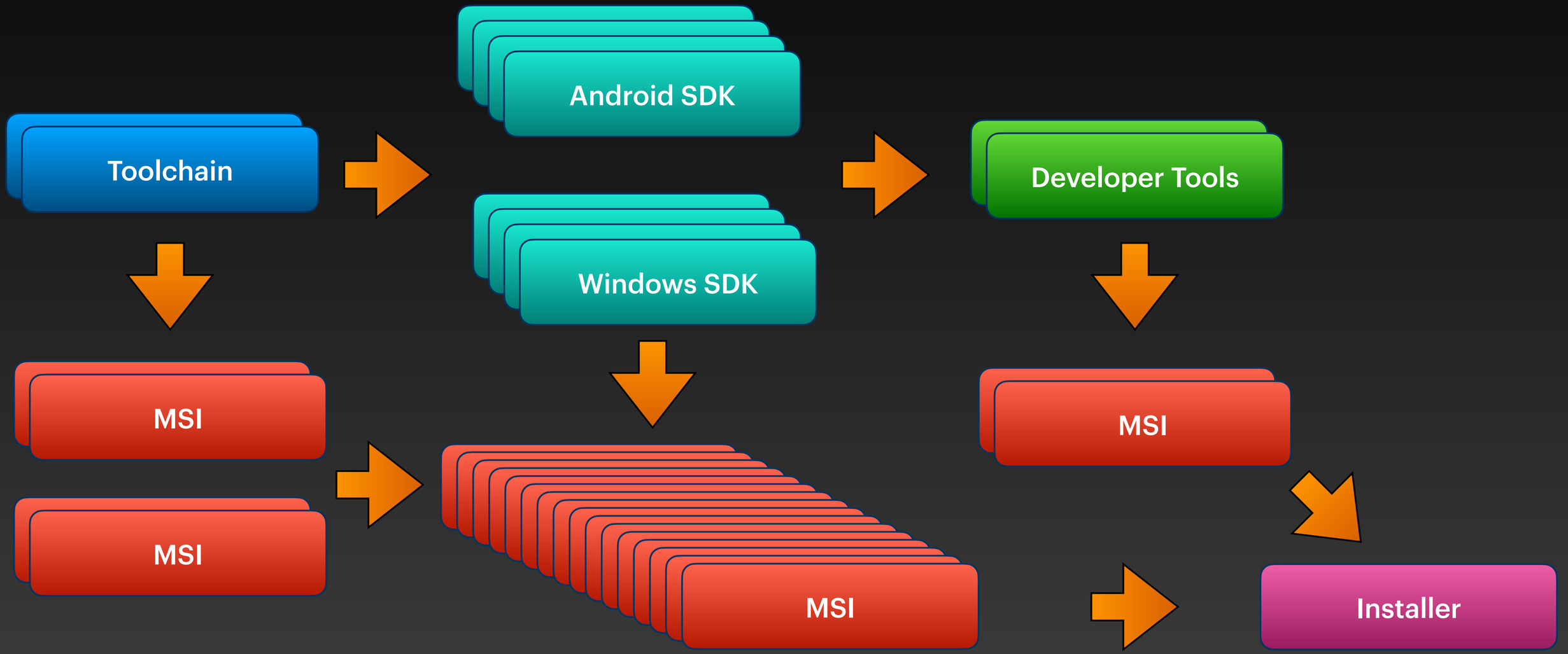
Oscillations

Building in Phases



Oscillations

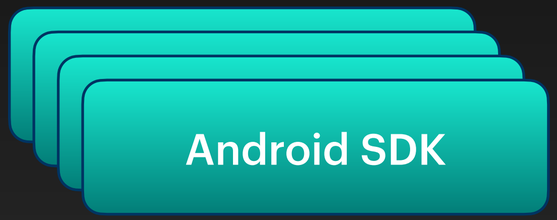
Building in Phases

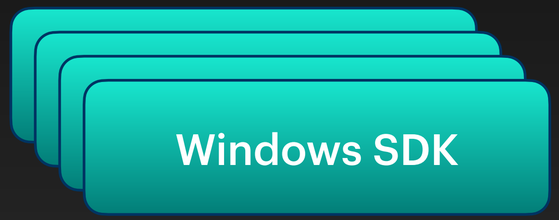


Build All The Things!

Build All The Things!

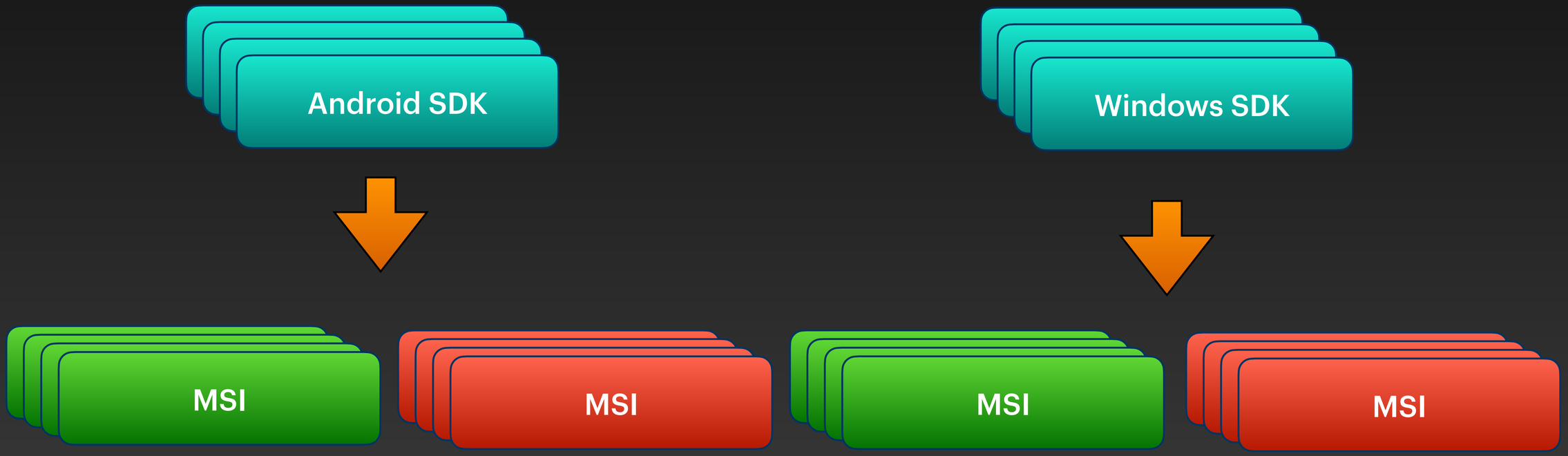
Packaging Explosion





Build All The Things!

Packaging Explosion



Not So Quick

Not So Quick

Complications

Not So Quick
Complications

- C/C++ SDK provided by the platform

Not So Quick
Complications

- C/C++ SDK provided by the platform
  - WinSDK
  - NDK

Not So Quick
Complications

- C/C++ SDK provided by the platform
  - WinSDK
  - NDK
- WinSDK has depends on MSVC Tool Set

Not So Quick
Complications

- C/C++ SDK provided by the platform
  - WinSDK
  - NDK
- WinSDK has depends on MSVC Tool Set
- clang modules are defined inline

Not So Quick
Complications

- C/C++ SDK provided by the platform
  - WinSDK
  - NDK
- WinSDK has depends on MSVC Tool Set
- clang modules are defined inline
- Post-facto modifications

Ignore Your Target Audience

Ignore Your Target Audience

Toolchain Flexibility

Ignore Your Target Audience

Toolchain Flexibility

- Uniformity across platforms

Ignore Your Target Audience

Toolchain Flexibility

- Uniformity across platforms
  - Installation Root (%SystemDrive%)

Ignore Your Target Audience

Toolchain Flexibility

- Uniformity across platforms
  - Installation Root (%SystemDrive%)
  - Encapsulate Unix layout

Ignore Your Target Audience

Toolchain Flexibility

- Uniformity across platforms
  - Installation Root (%SystemDrive%)
  - Encapsulate Unix layout
  - Inspired by macOS

Ignore Your Target Audience

Toolchain Flexibility

- Uniformity across platforms
  - Installation Root (%SystemDrive%)
  - Encapsulate Unix layout
  - Inspired by macOS
- Built for flexibility, not familiarity

Ignore Your Target Audience

Toolchain Flexibility

- Uniformity across platforms
  - Installation Root (%SystemDrive%)
  - Encapsulate Unix layout
  - Inspired by macOS
- Built for flexibility, not familiarity
- Setup for split, parallel toolchain installations

Ignore Your Target Audience

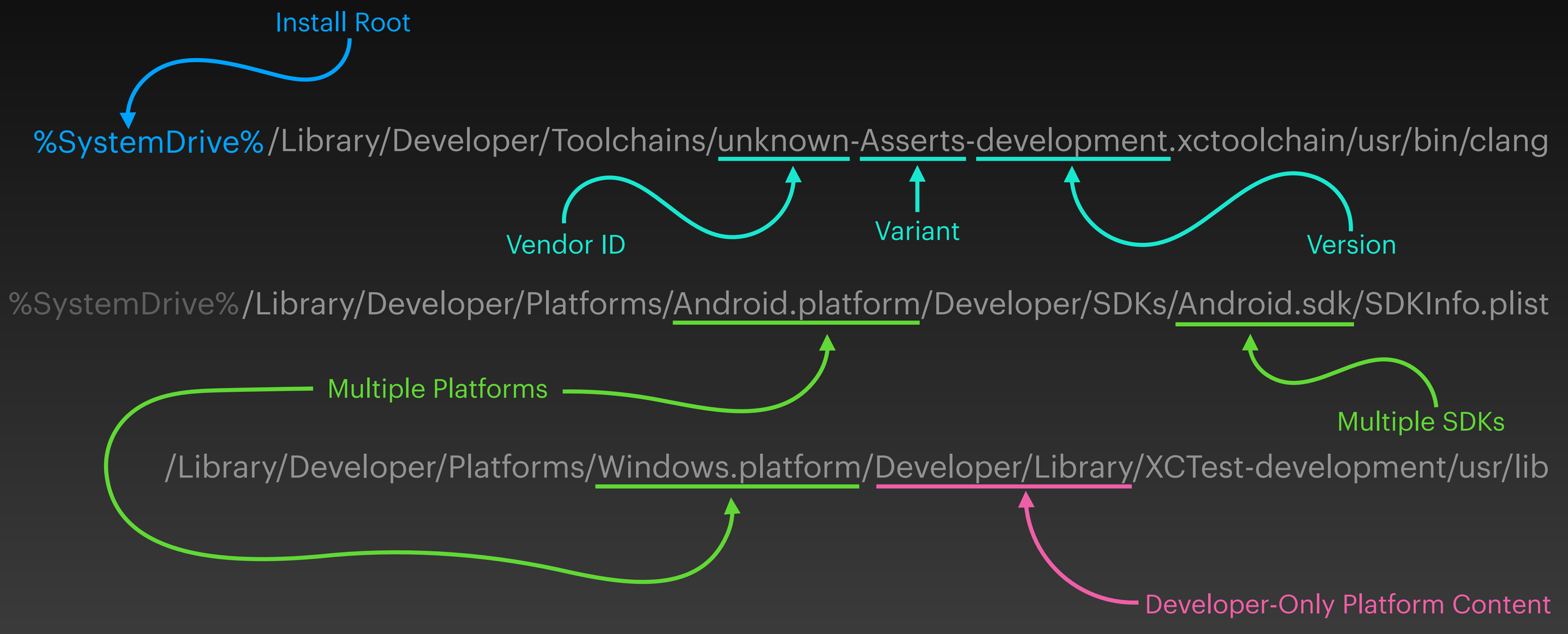
Toolchain Flexibility

- Uniformity across platforms
  - Installation Root (%SystemDrive%)
  - Encapsulate Unix layout
  - Inspired by macOS
- Built for flexibility, not familiarity
- Setup for split, parallel toolchain installations
- Multiple parallel SDK installation

5 seconds of Fame

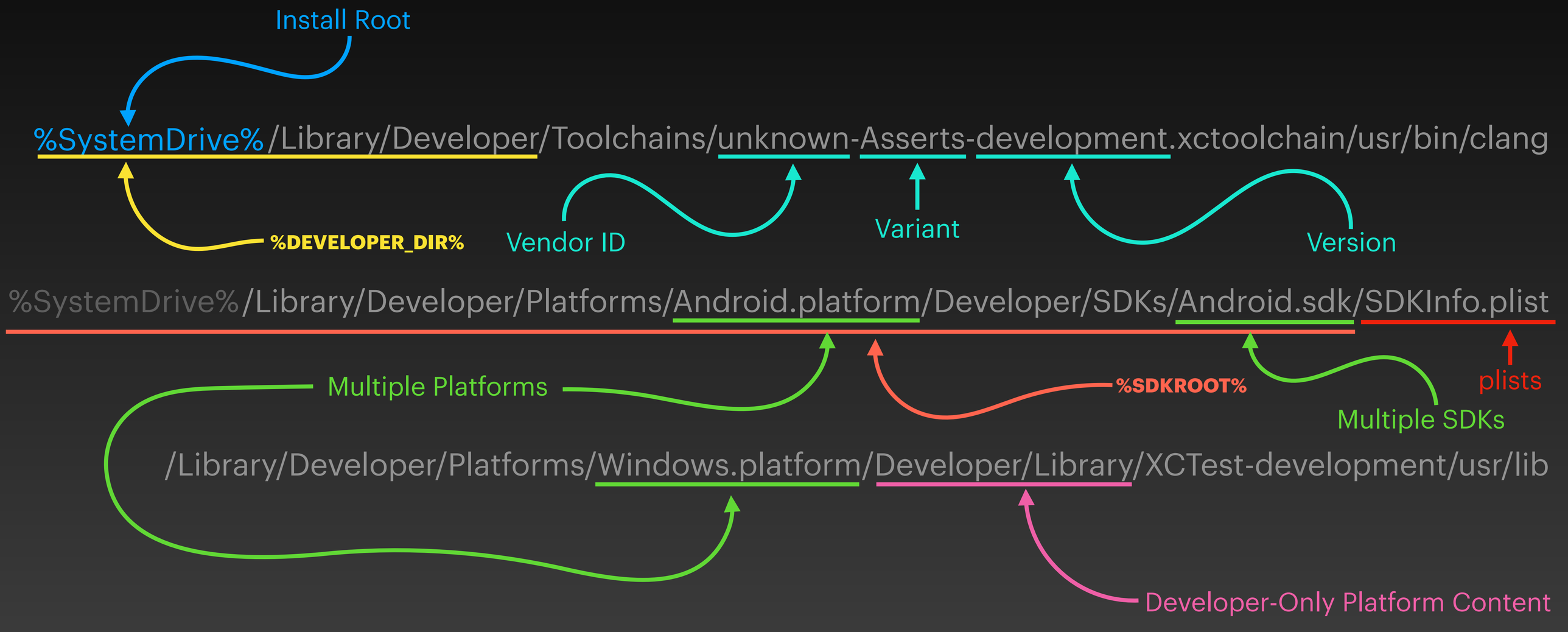
5 seconds of Fame

The Actual Layout



5 seconds of Fame

The Actual Layout



Something’s Amiss

Something's Amiss

Missing Pieces

Something's Amiss

Missing Pieces

- mt is incomplete

Something's Amiss

Missing Pieces

- mt is incomplete
- Fallback to MSVC Tools

The Great Depression

The Great Depression

LLVM Won’t Help

The Great Depression
LLVM Won’t Help

- Non-executable components

The Great Depression
LLVM Won’t Help

- Non-executable components
  - runtimes build

The Great Depression
LLVM Won’t Help

- Non-executable components
  - runtimes build
    - VsDevCmd

The Great Depression
LLVM Won’t Help

- Non-executable components
  - runtimes build
    - VsDevCmd
    - Runtime Dependencies

The Great Depression
LLVM Won’t Help

- Non-executable components
  - runtimes build
    - VsDevCmd
    - Runtime Dependencies
    - CMake Dependency

The Great Depression
LLVM Won’t Help

- Non-executable components
  - runtimes build
    - VsDevCmd
    - Runtime Dependencies
    - CMake Dependency
- Custom Package Management

The Great Depression
LLVM Won’t Help

- Non-executable components
  - runtimes build
    - VsDevCmd
    - Runtime Dependencies
    - CMake Dependency
- Custom Package Management
- LLVM is not a cross-compiling build setup

Schooling

Schooling

Lessons Learnt

Schooling
Lessons Learnt

- Prefer MSVC over LLVM

Schooling
Lessons Learnt

- Prefer MSVC over LLVM
- Integrate into LLVM’s build if possible

Schooling
Lessons Learnt

- Prefer MSVC over LLVM
- Integrate into LLVM’s build if possible
- Compose CMake caches for external builds

Schooling
Lessons Learnt

- Prefer MSVC over LLVM
- Integrate into LLVM’s build if possible
- Compose CMake caches for external builds
- Overly verbose distribution cache

Thank You

winget install swift

https://youtu.be/ZjlXa1NIfJc