#WWDC19

# Optimizing App Launch

Spencer Lewson, Performance Engineer
Dan Sawada, Performance Engineer

What is launch

How to properly measure your launch

Use Instruments to profile your launch

Track your launch over time

App launch is a user experience interruption.


9:41 AM
FaceTime
Friday
31
Calendar
Photos
Camera
Mail
Clock
Maps
Weather
Reminders
Notes
Stocks
News
Books
App Store
Podcasts
TV
Health
Home
Wallet
Settings
StarSearcher

# 0.000


9:41 AM
FaceTime
Friday
31
Calendar
Photos
Camera
Mail
Clock
Maps
Weather
Reminders
Notes
Stocks
News
Books
App Store
Podcasts
TV
Health
Home
Wallet
Settings
StarSearcher

# 0.000

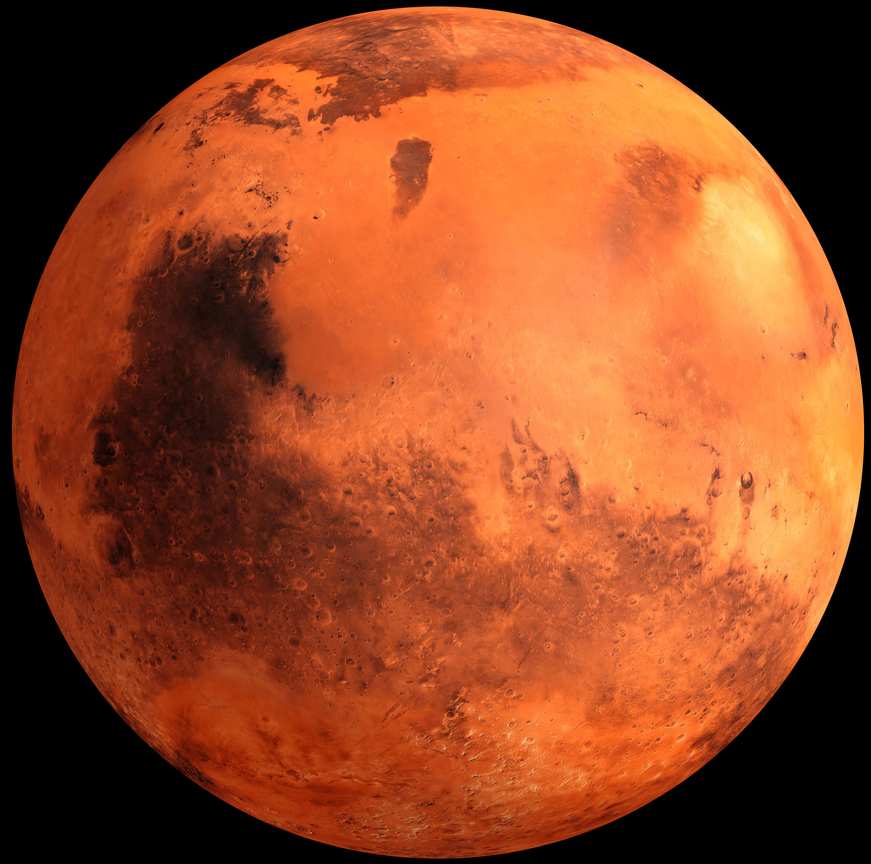
If we saved 1 millisecond per launch...

# 162 days

For each millisecond saved

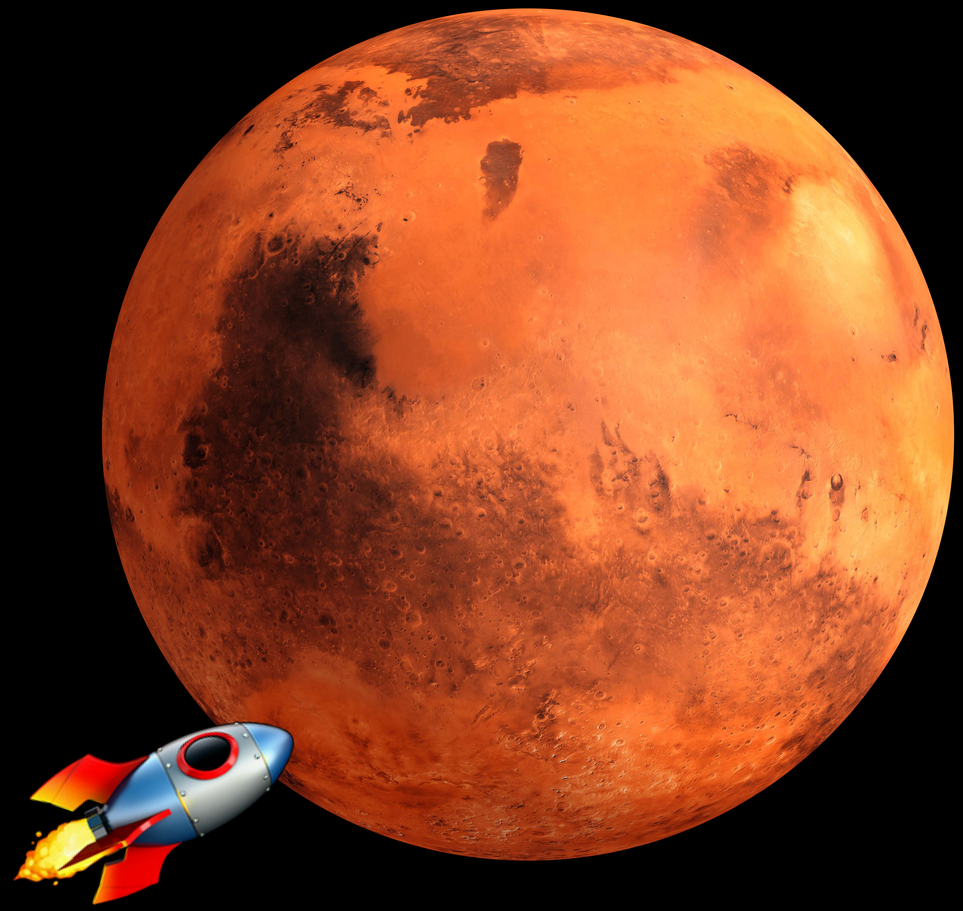
# 162 days

For each millisecond saved



# 162 days

For each millisecond saved



# Why Launch Is Important

# Why Launch Is Important

First experience with your app should be delightful

# Why Launch Is Important

First experience with your app should be delightful

Indicative of your code’s overall performance

# Why Launch Is Important

First experience with your app should be delightful

Indicative of your code’s overall performance

Impacts the system performance and battery

# Launch Types

Cold

Warm

Resume

# Launch Types

| Cold | Warm | Resume |
| --- | --- | --- |
| After reboot |  |  |
| App is not in memory | | |
| No process exists | | |

# Launch Types

| Cold | Warm | Resume |
| --- | --- | --- |
| After reboot | Recently terminated |  |
| App is not in memory | App is partially in memory | |
| No process exists | No process exists | |

# Launch Types

| Cold | Warm | Resume |
| --- | --- | --- |
| After reboot | Recently terminated | App is suspended |
| App is not in memory | App is partially in memory | App is fully in memory |
| No process exists | No process exists | Process exists |


Launch


Resume


Launch


Resume

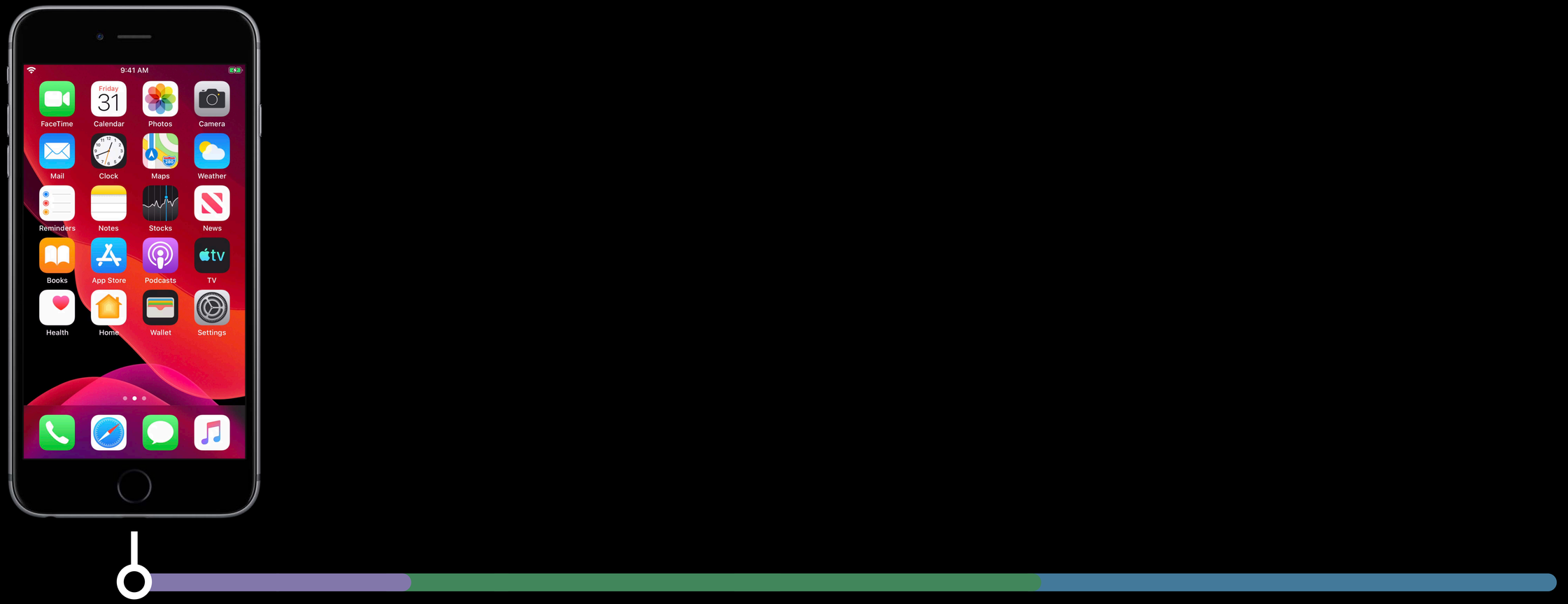

Launch


Resume

# 400 ms

To render first frame

# Phases of App Launch



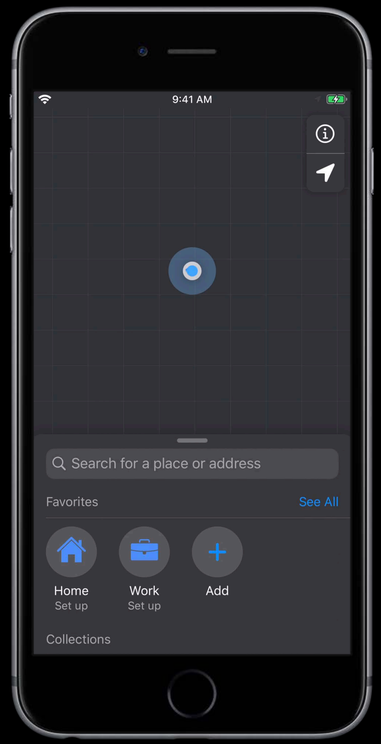
# Phases of App Launch


# Phases of App Launch

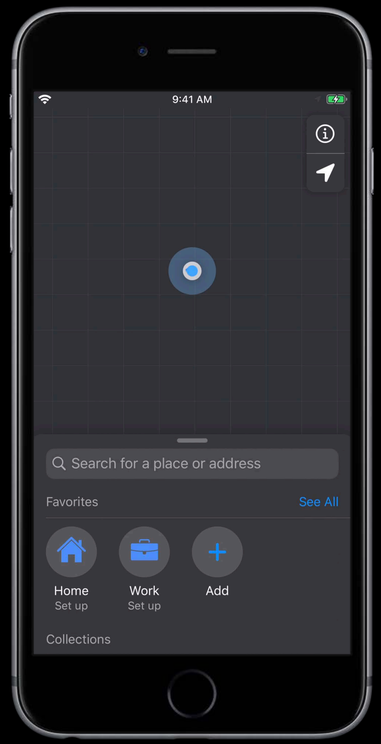
# Phases of App Launch


100

# Phases of App Launch




# Phases of App Launch




# Phases of App Launch


# Phases of App Launch


100

400

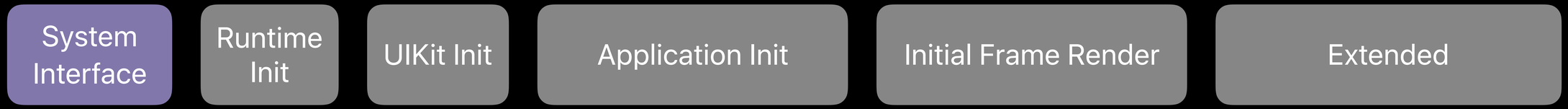
600

# Phases of App Launch

# Phases of App Launch

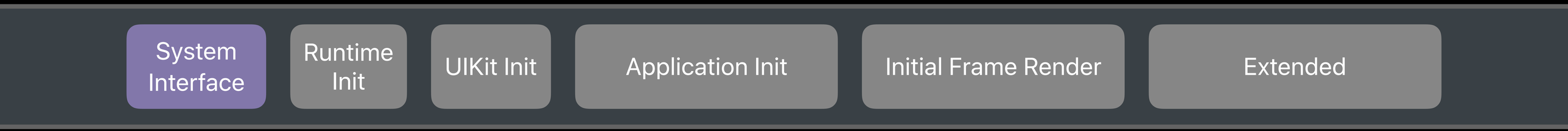
# System Interface

DYLD3



# System Interface

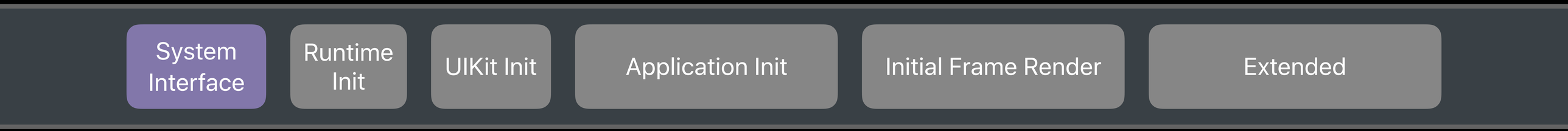
DYLD3



# System Interface

## DYLD3

Dynamic Linker loads shared libraries and frameworks



# System Interface

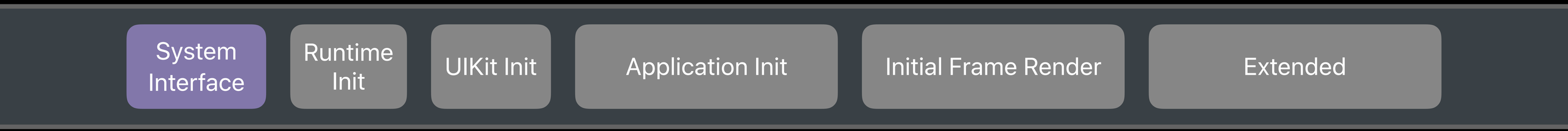
DYLD3

NEW

Dynamic Linker loads shared libraries and frameworks

Introduces caching of runtime dependencies to improve warm launch

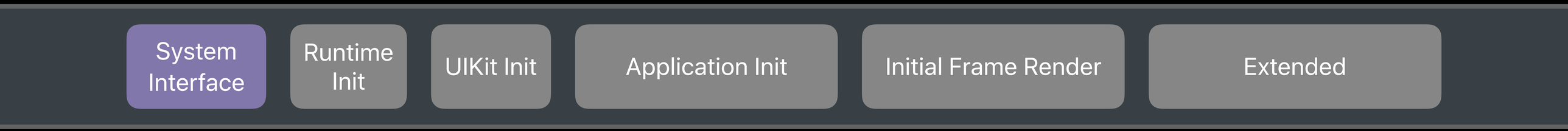
App Startup Time: Past, Present, and Future — WWDC 2017



# System Interface

DYLD3

NEW

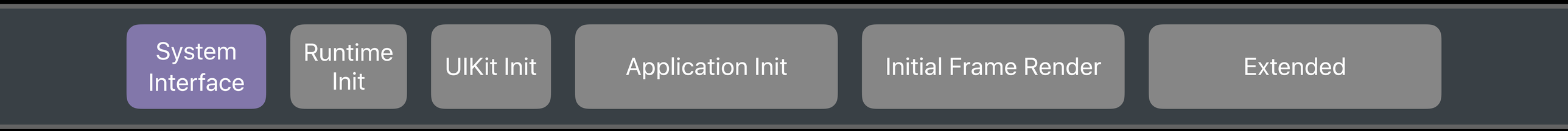


# System Interface

DYLD3

NEW

- Avoid linking unused frameworks

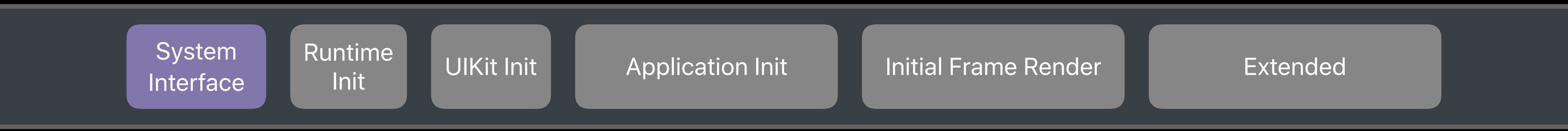


# System Interface

DYLD3

NEW

- Avoid linking unused frameworks
- Avoid dynamic library loading during launch

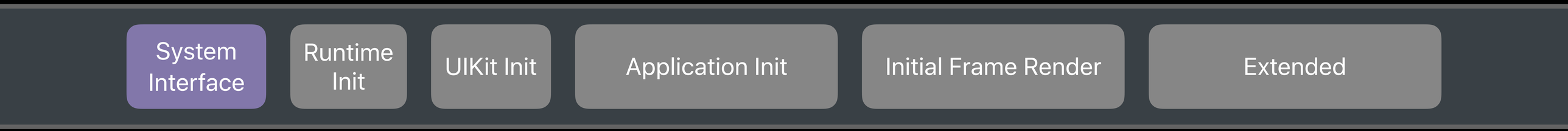


# System Interface

DYLD3

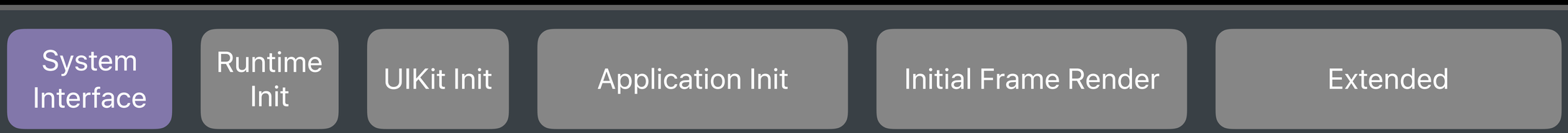
NEW

- Avoid linking unused frameworks
- Avoid dynamic library loading during launch
- Hard link all your dependencies



# System Interface

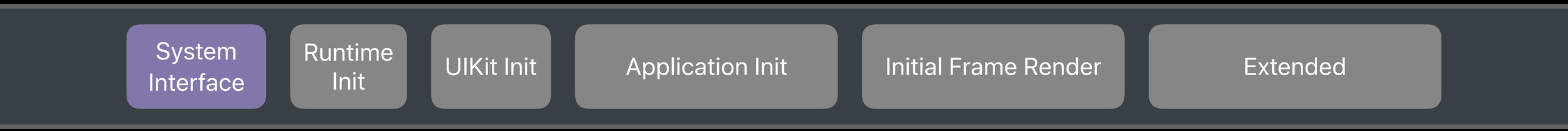
libSystem Init



# System Interface

## libSystem Init

Initializes the interfaces with low level system components

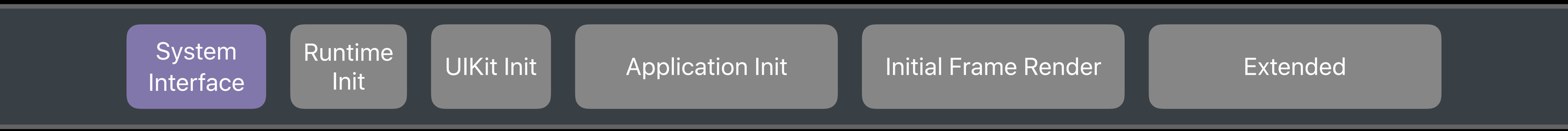


# System Interface

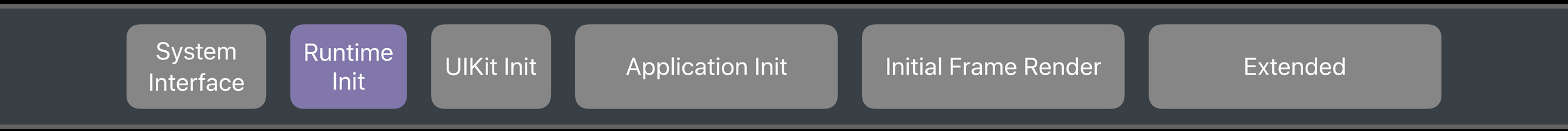
libSystem Init

Initializes the interfaces with low level system components

System side work with a fixed cost

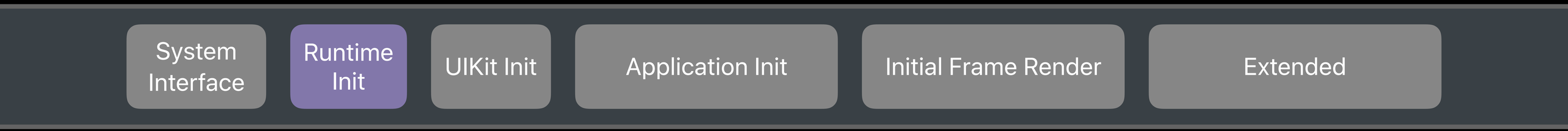


# Static Runtime Initialization



# Static Runtime Initialization

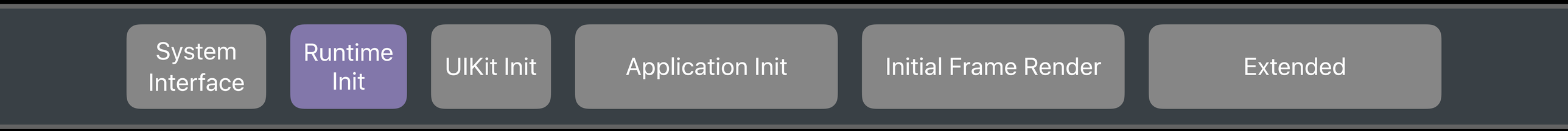
Initializes the language runtime



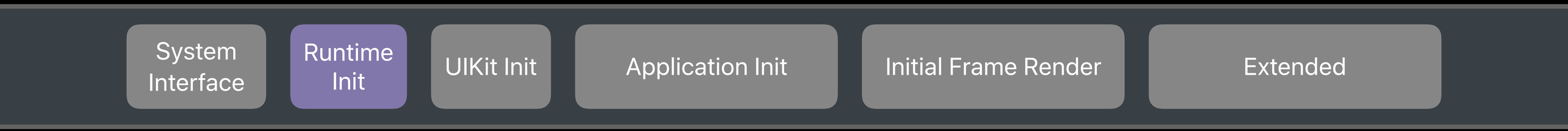
# Static Runtime Initialization

Initializes the language runtime

Invokes all class static load methods

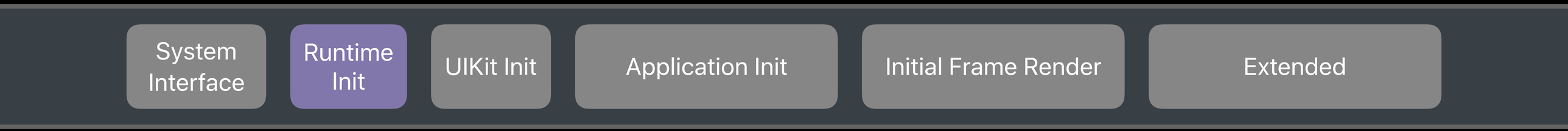


# Static Runtime Initialization



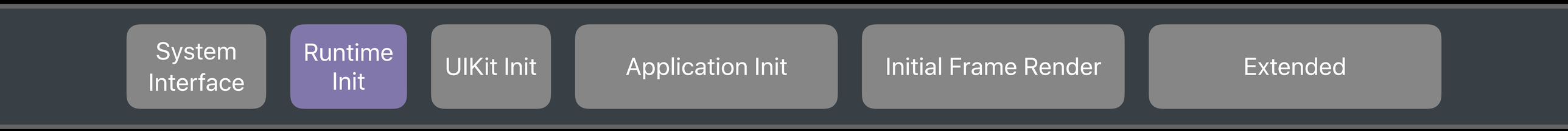
# Static Runtime Initialization

- Expose dedicated initialization API in frameworks



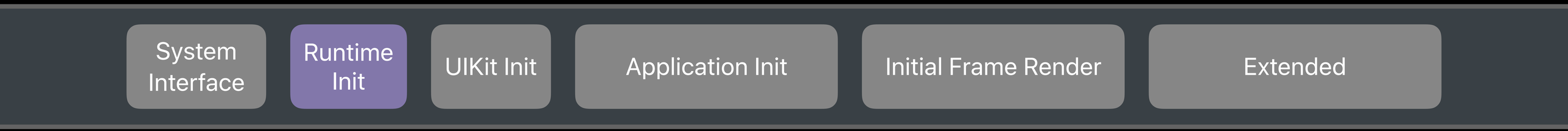
# Static Runtime Initialization

- Expose dedicated initialization API in frameworks
- Reduce impact to launch by avoiding `+[Class load]`

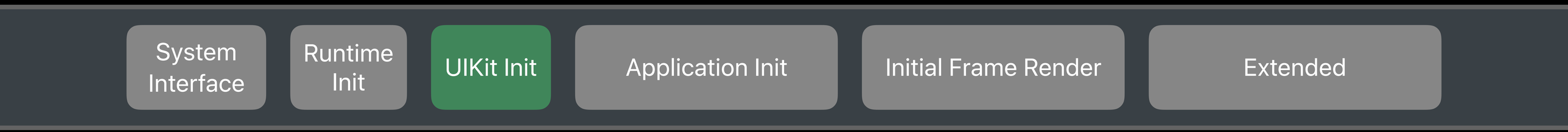


# Static Runtime Initialization

- Expose dedicated initialization API in frameworks
- Reduce impact to launch by avoiding `+[Class load]`
- Use `+[Class initialize]` to lazily conduct static initialization

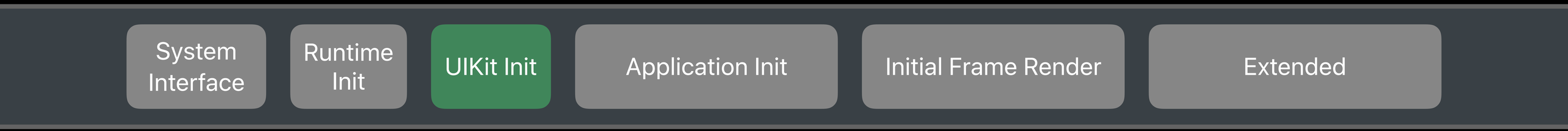


# UIKit Initialization



# UIKit Initialization

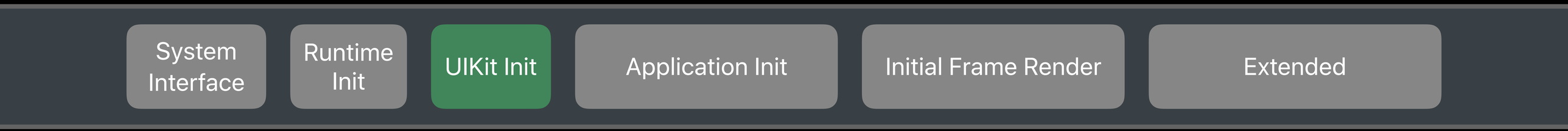
Instantiates the `UIApplication` and `UIApplicationDelegate`



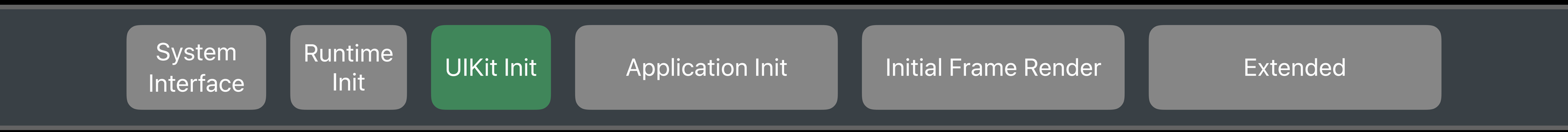
# UIKit Initialization

Instantiates the `UIApplication` and `UIApplicationDelegate`

Begins event processing and integration with the system

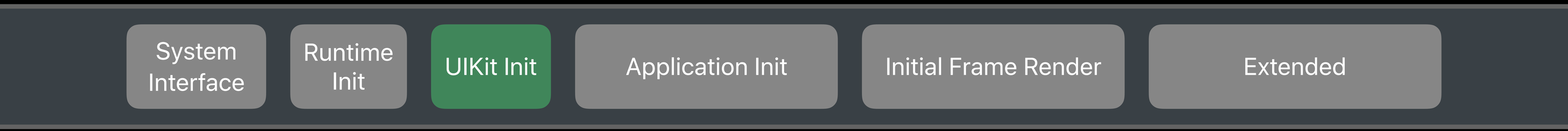


# UIKit Initialization



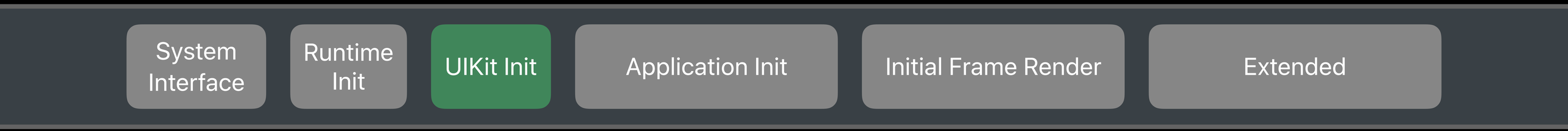
# UIKit Initialization

- Minimize work in `UIApplication` subclass



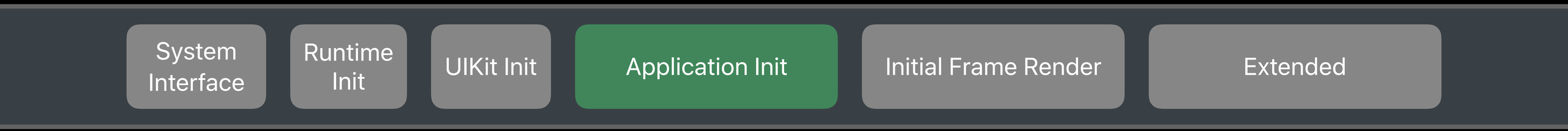
# UIKit Initialization

- Minimize work in `UIApplication` subclass
- Minimize work in `UIApplicationDelegate` initialization



# Application Initialization

## Lifecycle Callbacks

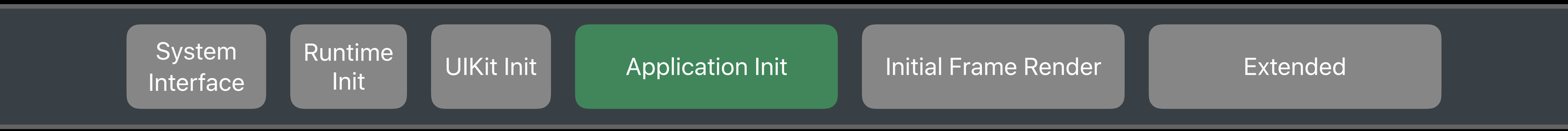


# Application Initialization

## Lifecycle Callbacks

Invokes UIApplicationDelegate app lifecycle callbacks

```
application:willFinishLaunchingWithOptions:
application:didFinishLaunchingWithOptions:
```



# Application Initialization

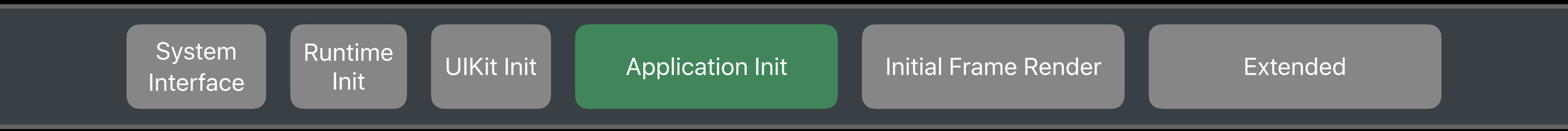
## Lifecycle Callbacks

### Invokes UIApplicationDelegate app lifecycle callbacks

```
application:willFinishLaunchingWithOptions:
application:didFinishLaunchingWithOptions:
```

### Invokes UIApplicationDelegate UI lifecycle callbacks

```
applicationDidBecomeActive:
```



# Application Initialization

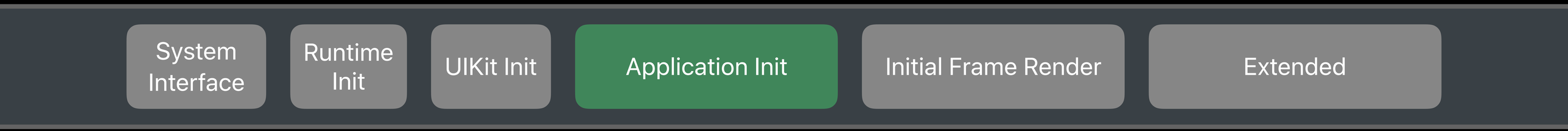
## Lifecycle Callbacks

### Invokes UIApplicationDelegate app lifecycle callbacks

```
application:willFinishLaunchingWithOptions:
application:didFinishLaunchingWithOptions:
```

### Invokes UIApplicationDelegate UI lifecycle callbacks

```
applicationDidBecomeActive:
```



# Application Initialization

## Lifecycle Callbacks

NEW

### Invokes UIApplicationDelegate app lifecycle callbacks

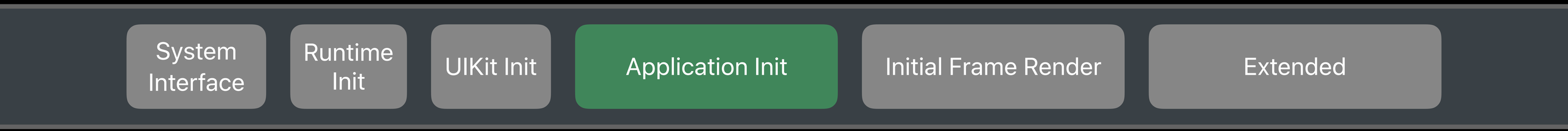
```
application:willFinishLaunchingWithOptions:
application:didFinishLaunchingWithOptions:
```

### Invokes UIApplicationDelegate UI lifecycle callbacks

```
applicationDidBecomeActive:
```

### Invokes UISceneDelegate UI lifecycle callbacks for each scene

```
scene:willConnectToSession:options:
sceneWillEnterForeground:
sceneDidBecomeActive:
```



# Application Initialization

## Lifecycle Callbacks

NEW

### Invokes UIApplicationDelegate app lifecycle callbacks

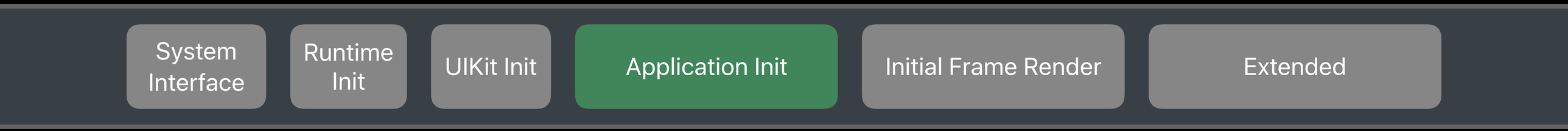
```
application:willFinishLaunchingWithOptions:
application:didFinishLaunchingWithOptions:
```

### Invokes UIApplicationDelegate UI lifecycle callbacks

```
applicationDidBecomeActive:
```

### Invokes UISceneDelegate UI lifecycle callbacks for each scene

```
scene:willConnectToSession:options:
sceneWillEnterForeground:
sceneDidBecomeActive:
```



# Application Initialization

## Lifecycle Callbacks

NEW

```
application:willFinishLaunchingWithOptions:
application:didFinishLaunchingWithOptions:
```

```
applicationDidBecomeActive:
```

```
scene:willConnectToSession:options:
sceneWillEnterForeground:
sceneDidBecomeActive:
```

System Interface | Runtime Init | UIKit Init | Application Init | Initial Frame Render | Extended

# Application Initialization

## Lifecycle Callbacks

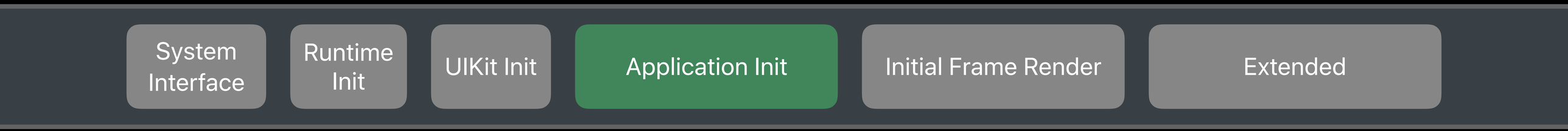
NEW

- Defer unrelated work

```
application:willFinishLaunchingWithOptions:
application:didFinishLaunchingWithOptions:
```

```
applicationDidBecomeActive:
```

```
scene:willConnectToSession:options:
sceneWillEnterForeground:
sceneDidBecomeActive:
```



# Application Initialization

## Lifecycle Callbacks

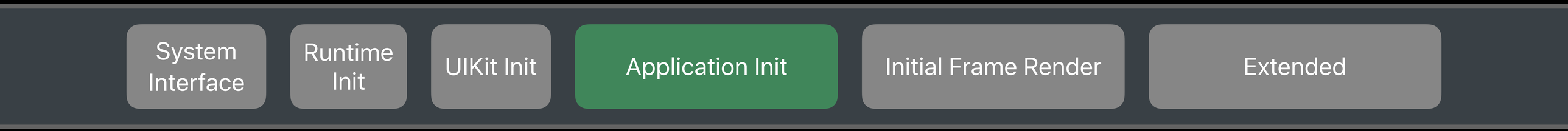
NEW

- Defer unrelated work
- Share resources between scenes

```
application:willFinishLaunchingWithOptions:
application:didFinishLaunchingWithOptions:
```

```
applicationDidBecomeActive:
```

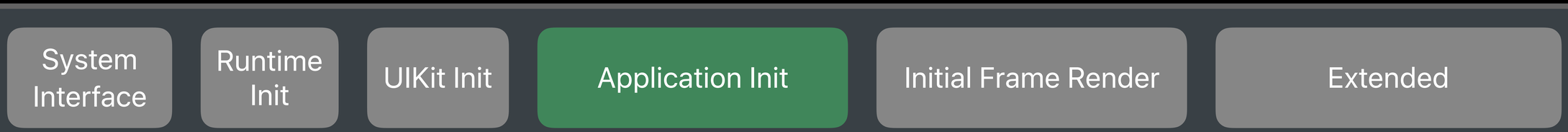
```
scene:willConnectToSession:options:
sceneWillEnterForeground:
sceneDidBecomeActive:
```



# Application Initialization

## Lifecycle Callbacks

| | |
|---|---|
| Multitasking and the Application Lifecycle | WWDC 2019 |
| Getting the Most out of Multitasking | WWDC 2019 |



# First Frame Render

System Interface | Runtime Init | UIKit Init | Application Init | Initial Frame Render | Extended

# First Frame Render

Creates, performs layout for, and draws views

```
loadView
viewDidLoad
layoutSubviews
```

# First Frame Render

Creates, performs layout for, and draws views

Commits and renders first frame

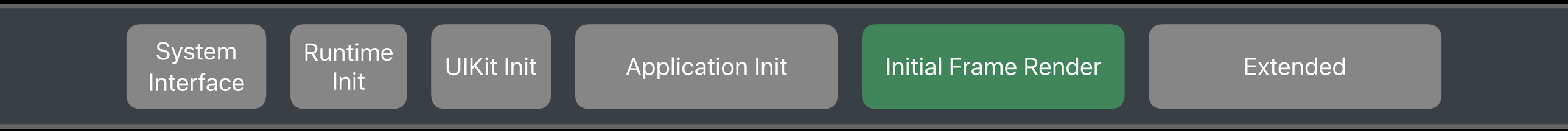
```
loadView
viewDidLoad
layoutSubviews
```

# First Frame Render

System Interface | Runtime Init | UIKit Init | Application Init | Initial Frame Render | Extended

# First Frame Render

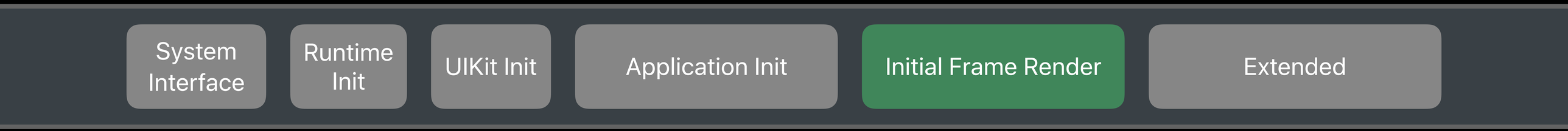
- Flatten view hierarchies and lazily load views



# First Frame Render

- Flatten view hierarchies and lazily load views
- Optimize auto layout usage



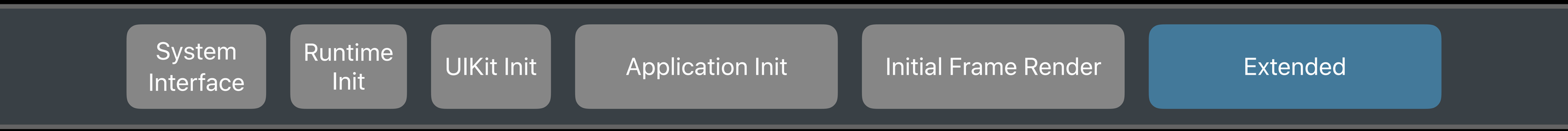


# Extended

System Interface | Runtime Init | UIKit Init | Application Init | Initial Frame Render | Extended

# Extended

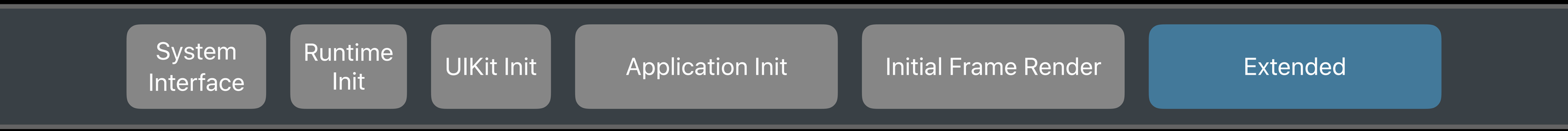
App-specific period after first frame



# Extended

App-specific period after first frame

Displays asynchronously loaded data

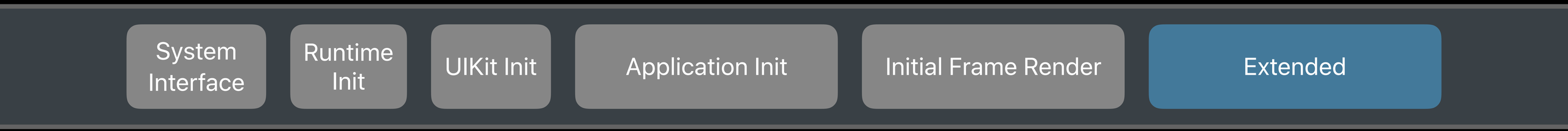


# Extended

App-specific period after first frame

Displays asynchronously loaded data

App should be interactive and responsive



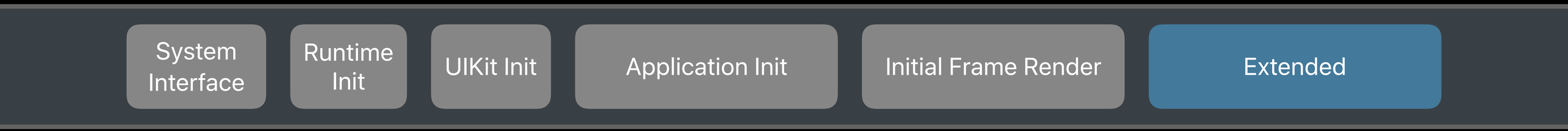
# Extended

System Interface | Runtime Init | UIKit Init | Application Init | Initial Frame Render | Extended

# Extended

- Leverage os_signpost to measure work

Measuring Performance Using Logging WWDC 2018



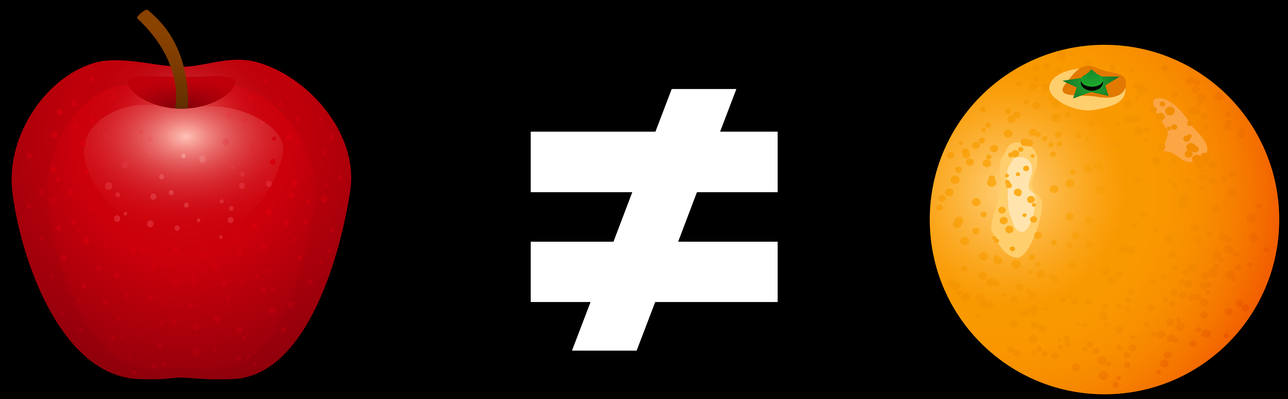
What is launch

How to properly measure your launch

Use Instruments to profile your launch

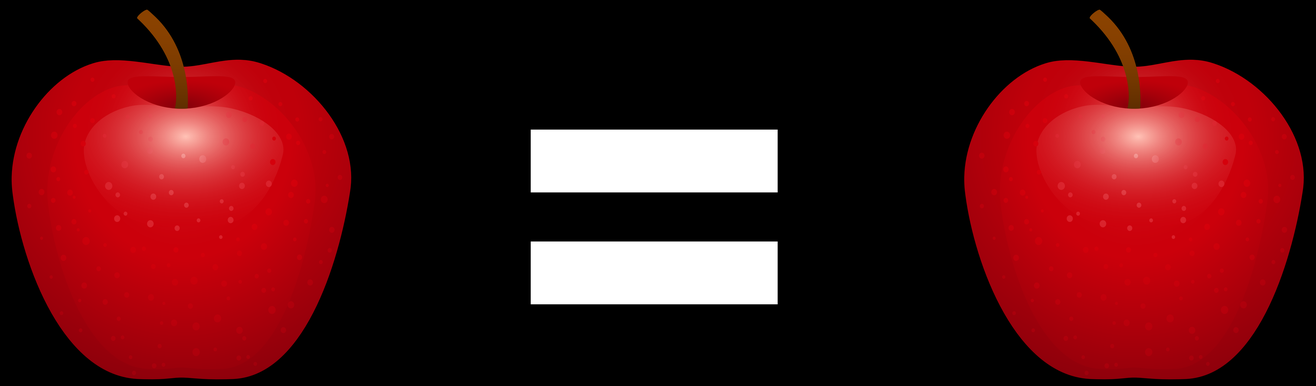
Track your launch over time

# Trading Representativeness for Consistency



Comparison requires controlled variables

# Trading Representativeness for Consistency



Comparison requires controlled variables

# Trading Representativeness for Consistency

# Trading Representativeness for Consistency

Remove sources of variance to produce more consistent results

# Trading Representativeness for Consistency

Remove sources of variance to produce more consistent results

May result in launch times that are not representative

# Trading Representativeness for Consistency

Remove sources of variance to produce more consistent results

May result in launch times that are not representative

Use consistent results to evaluate progress

# Test in a Clean and Consistent Environment

# Test in a Clean and Consistent Environment

- Reboot then let system quiesce for 2–3 minutes

# Test in a Clean and Consistent Environment

- Reboot then let system quiesce for 2–3 minutes
- Enable Airplane Mode or mock the network

# Test in a Clean and Consistent Environment

- Reboot then let system quiesce for 2–3 minutes
- Enable Airplane Mode or mock the network
- Use unchanging or no iCloud Account

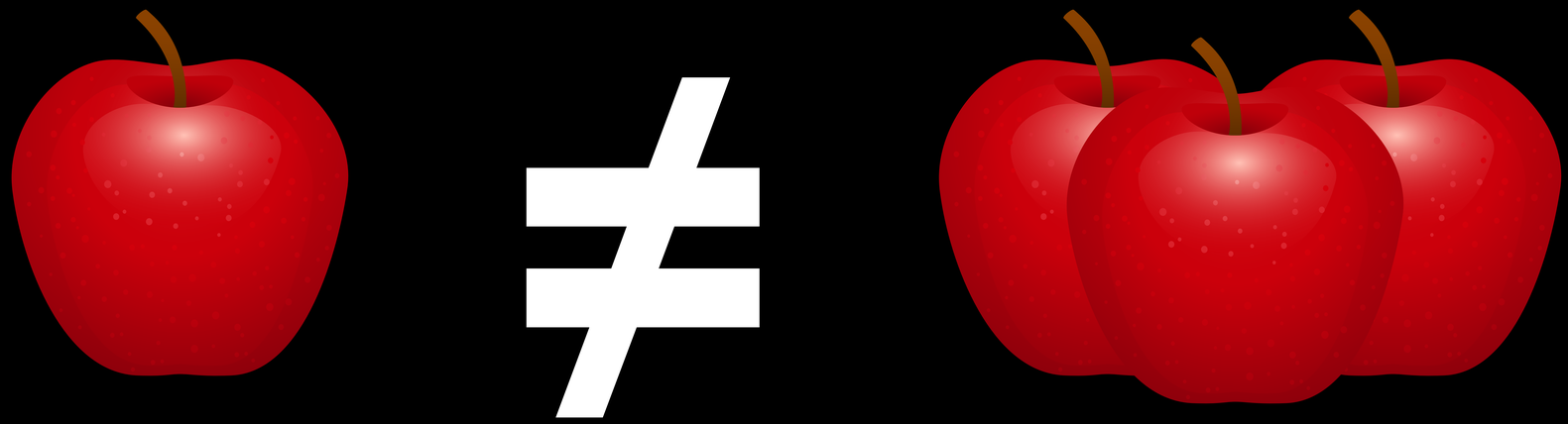
# Test in a Clean and Consistent Environment

- Reboot then let system quiesce for 2–3 minutes
- Enable Airplane Mode or mock the network
- Use unchanging or no iCloud Account
- Use release build of your app

# Test in a Clean and Consistent Environment

- Reboot then let system quiesce for 2–3 minutes
- Enable Airplane Mode or mock the network
- Use unchanging or no iCloud Account
- Use release build of your app
- Measure warm launches

# Test with Representative Data



Small versus Large Datasets

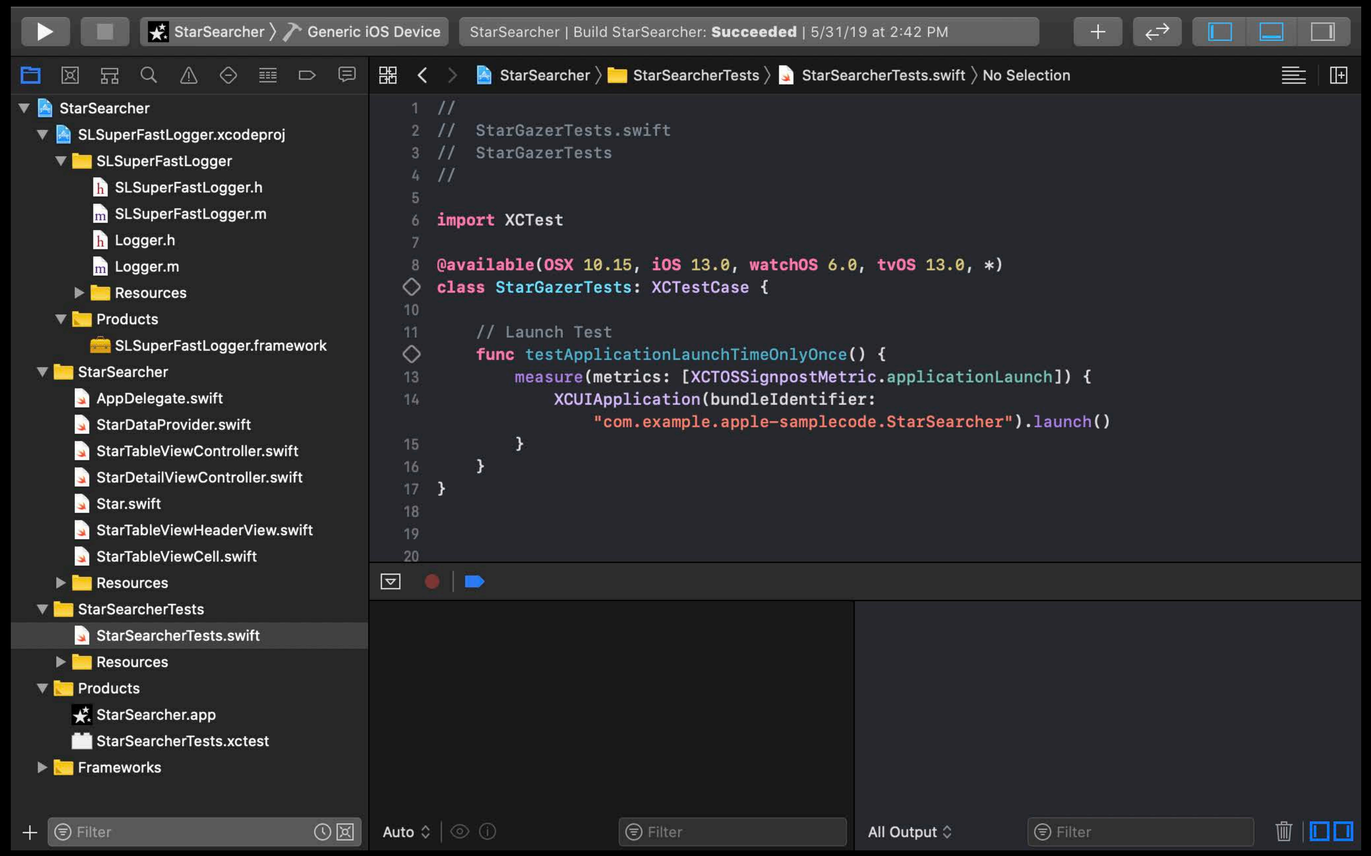
# Target Older and Newer Devices

Older devices

Newer devices

# Measuring Launch with XCTest

NEW



What is launch

How to properly measure your launch

Use Instruments to profile your launch

Track your launch over time

# Tips and Tricks

1. Minimize
2. Prioritize
3. Optimize

# 1. Minimize Work

# 1. Minimize Work

- Defer work unrelated to first frame

# 1. Minimize Work

- Defer work unrelated to first frame
- Move blocking work off main thread

# 1. Minimize Work

- Defer work unrelated to first frame
- Move blocking work off main thread
- Reduce memory usage

# 2. Prioritize Work

# 2. Prioritize Work

- Identify the right QoS for your task

# 2. Prioritize Work

- Identify the right QoS for your task
- Utilize scheduler optimizations for app launch

# 2. Prioritize Work

- Identify the right QoS for your task
- Utilize scheduler optimizations for app launch
- Preserve the priority with the right primitives

# 3. Optimize Work

# 3. Optimize Work

- Simplify or limit existing work

# 3. Optimize Work

- Simplify or limit existing work
- Optimize algorithms and data structures

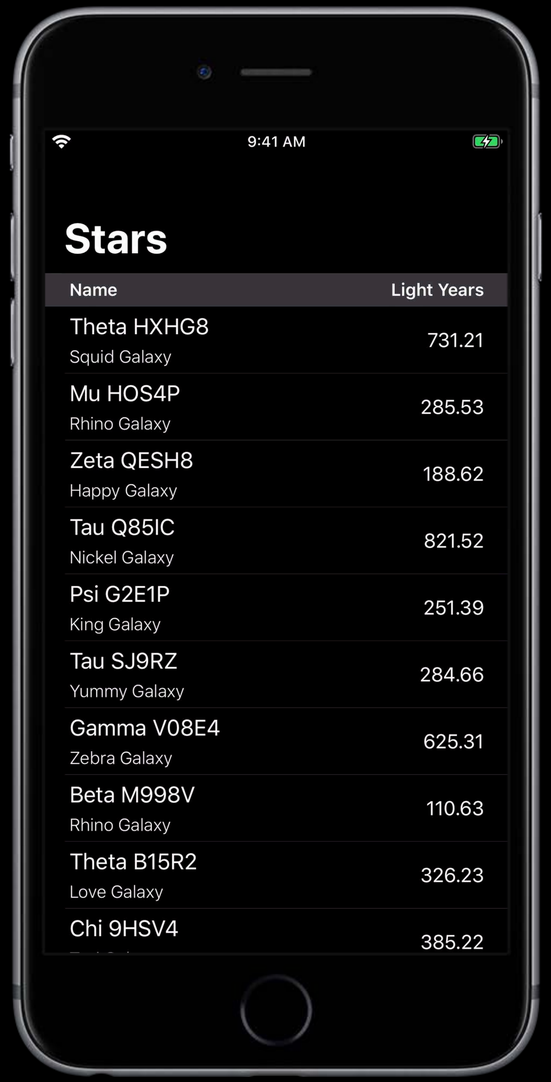
# 3. Optimize Work

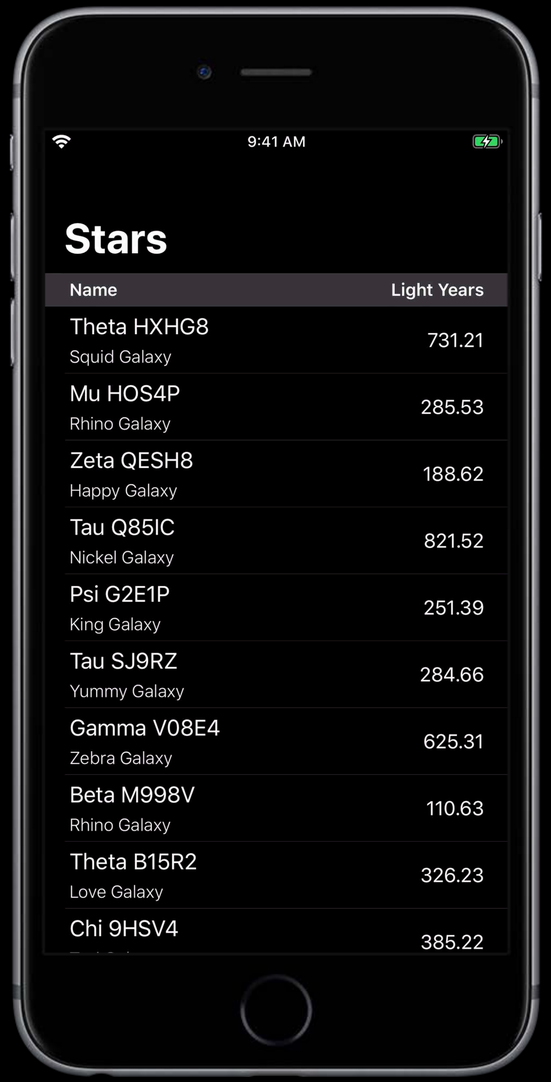
- Simplify or limit existing work
- Optimize algorithms and data structures
- Cache resources and computations

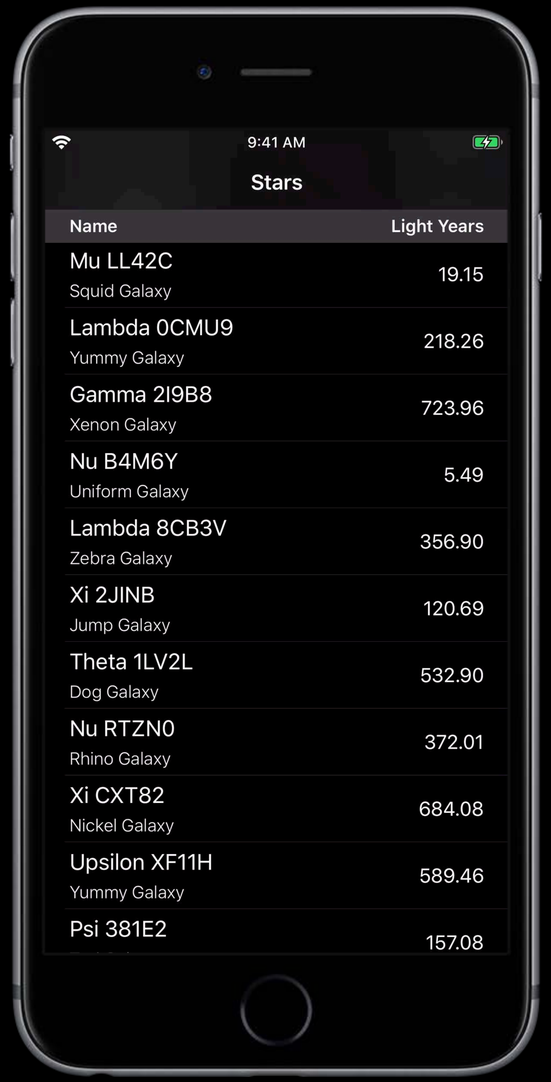
# *Demo*

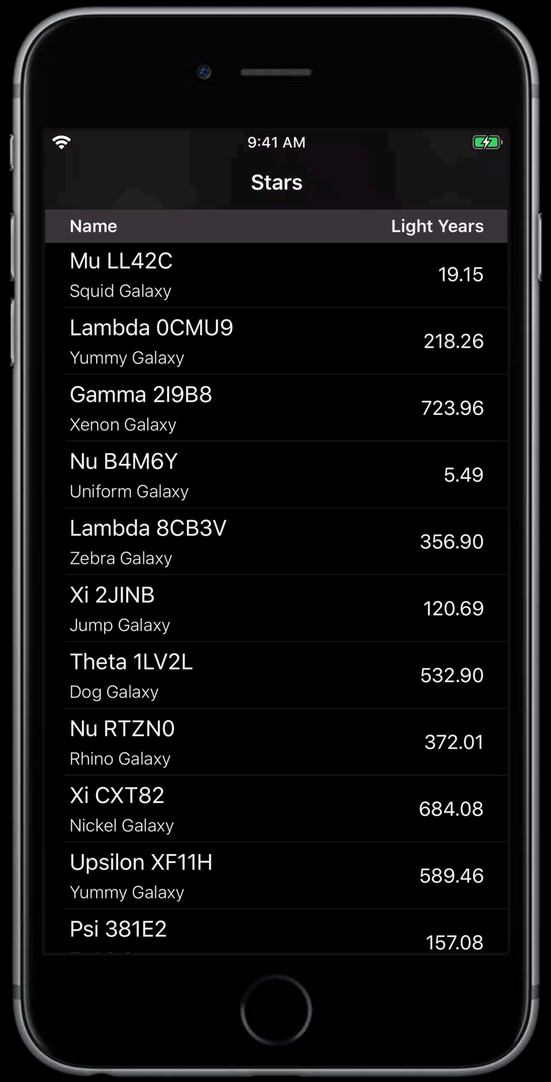
## Using the App Launch Template

Dan Sawada, Performance Engineer


9:41 AM
Stars
Name
Light Years
Theta HXHG8
Squid Galaxy
731.21
Mu HOS4P
Rhino Galaxy
285.53
Zeta QESH8
Happy Galaxy
188.62
Tau Q85IC
Nickel Galaxy
821.52
Psi G2E1P
King Galaxy
251.39
Tau SJ9RZ
Yummy Galaxy
284.66
Gamma V08E4
Zebra Galaxy
625.31
Beta M998V
Rhino Galaxy
110.63
Theta B15R2
Love Galaxy
326.23
Chi 9HSV4
385.22


9:41 AM
Stars
Name
Light Years
Theta HXHG8
Squid Galaxy
731.21
Mu HOS4P
Rhino Galaxy
285.53
Zeta QESH8
Happy Galaxy
188.62
Tau Q85IC
Nickel Galaxy
821.52
Psi G2E1P
King Galaxy
251.39
Tau SJ9RZ
Yummy Galaxy
284.66
Gamma V08E4
Zebra Galaxy
625.31
Beta M998V
Rhino Galaxy
110.63
Theta B15R2
Love Galaxy
326.23
Chi 9HSV4
385.22


9:41 AM
Stars
Name
Light Years
Mu LL42C
Squid Galaxy
19.15
Lambda 0CMU9
Yummy Galaxy
218.26
Gamma 2I9B8
Xenon Galaxy
723.96
Nu B4M6Y
Uniform Galaxy
5.49
Lambda 8CB3V
Zebra Galaxy
356.90
Xi 2JINB
Jump Galaxy
120.69
Theta 1LV2L
Dog Galaxy
532.90
Nu RTZN0
Rhino Galaxy
372.01
Xi CXT82
Nickel Galaxy
684.08
Upsilon XF11H
Yummy Galaxy
589.46
Psi 381E2
157.08


9:41 AM
Stars
Name
Light Years
Mu LL42C
Squid Galaxy
19.15
Lambda 0CMU9
Yummy Galaxy
218.26
Gamma 2I9B8
Xenon Galaxy
723.96
Nu B4M6Y
Uniform Galaxy
5.49
Lambda 8CB3V
Zebra Galaxy
356.90
Xi 2JINB
Jump Galaxy
120.69
Theta 1LV2L
Dog Galaxy
532.90
Nu RTZN0
Rhino Galaxy
372.01
Xi CXT82
Nickel Galaxy
684.08
Upsilon XF11H
Yummy Galaxy
589.46
Psi 381E2
157.08


9:41 AM
Stars
Gamma 2I9B8
Located at the Xenon Galaxy...
723.96 Light Years Ahead...


9:41 AM
Stars
Gamma 2I9B8
Located at the Xenon Galaxy...
723.96 Light Years Ahead...


9:41 AM
FaceTime
Friday
31
Calendar
Photos
Camera
Mail
Clock
Maps
Weather
Reminders
Notes
Stocks
News
Books
App Store
Podcasts
TV
Health
Home
Wallet
Settings
StarSearcher

# 0.000


9:41 AM
FaceTime
Friday
31
Calendar
Photos
Camera
Mail
Clock
Maps
Weather
Reminders
Notes
Stocks
News
Books
App Store
Podcasts
TV
Health
Home
Wallet
Settings
StarSearcher

# 0.000

# *Demo*

# System Improvements

GCD thread policy tunings

Sandbox improvements

Constant classes

Improved scheduler

Reduced ARC overhead

Runtime syscall elimination

Trampoline-less launch

App Packaging

Hot/cold code splitting

Elimination of priority inversions

Cached runtime dependencies

Auto Layout optimizations

kevent improvements

Static initializer reduction

Darwin notification optimizations

Adaptive spinning for contended runtime locks

# System Improvements

GCD thread policy tunings

Sandbox improvements

Constant classes

Improved scheduler

Reduced ARC overhead

Runtime syscall elimination

Trampoline-less launch

App Packaging

Hot/cold code splitting

Elimination of priority inversions

Cached runtime dependencies

Auto Layout optimizations

kevent improvements

Static initializer reduction

Darwin notification optimizations

Adaptive spinning for contended runtime locks

What is launch

How to properly measure your launch

Use Instruments to profile your launch

Track your launch over time

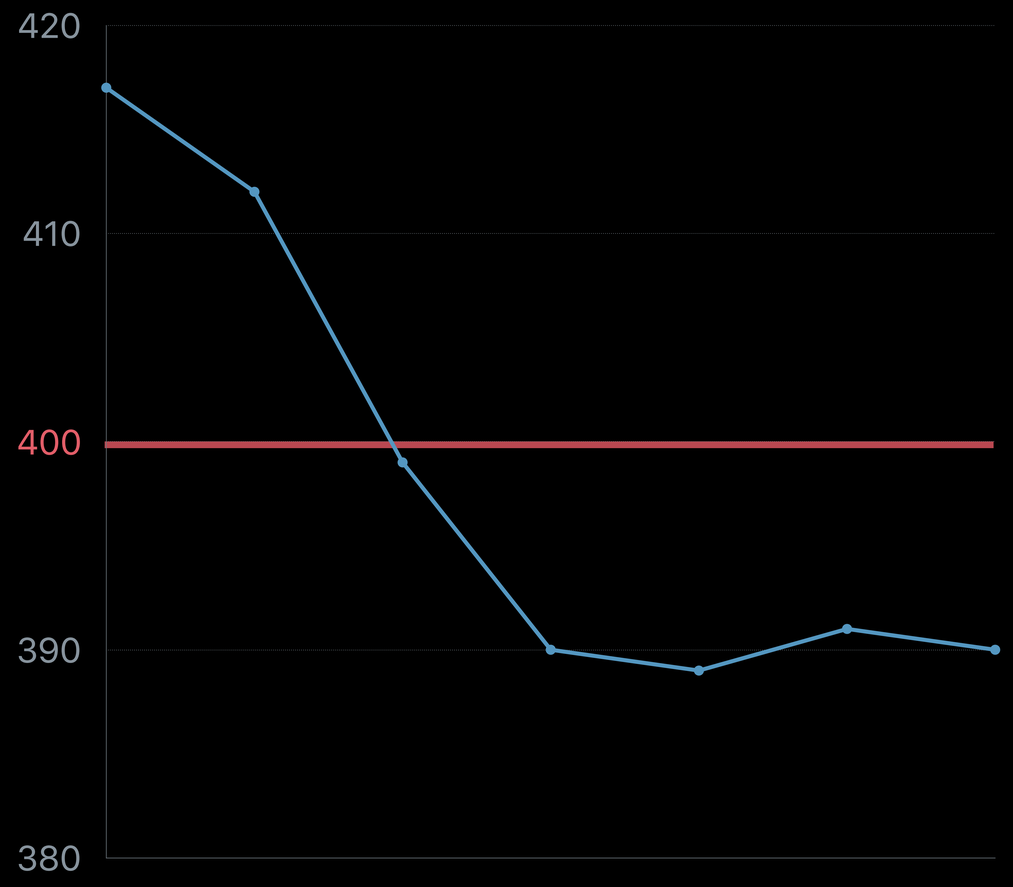
# Track Your Launch Over Time

# Track Your Launch Over Time

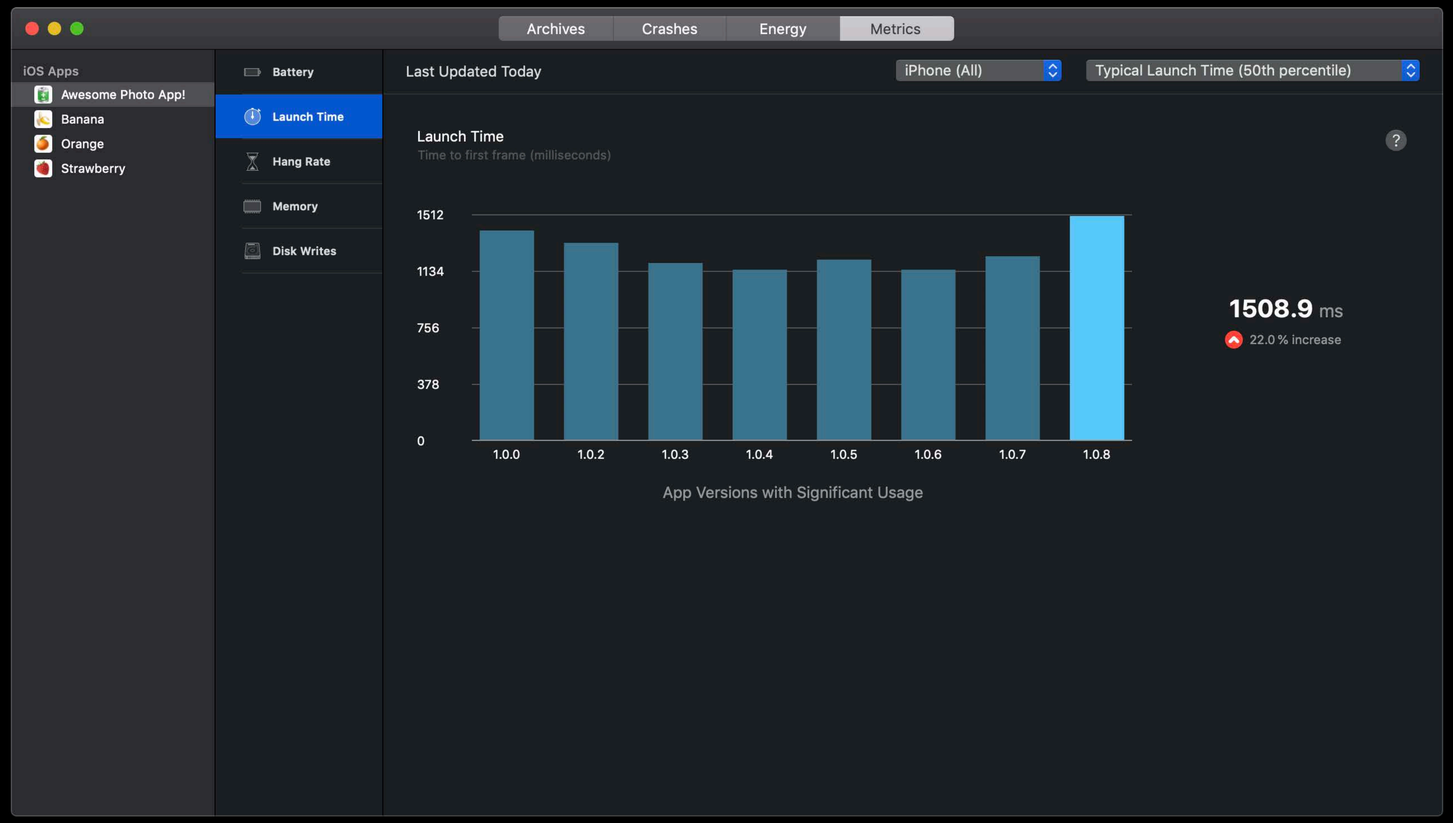
- Make performance a development-time priority

# Track Your Launch Over Time

- Make performance a development-time priority
- Plot it and have a target



# Monitor Customer Launches with Xcode Organizer



# Adopt MetricKit for More Statistics

# Adopt MetricKit for More Statistics

Collect custom power and performance metrics

# Adopt MetricKit for More Statistics

Collect custom power and performance metrics

Aggregated results delivered every 24 hours

# Adopt MetricKit for More Statistics

Collect custom power and performance metrics

Aggregated results delivered every 24 hours

# Summary

# Summary

- Start understanding your launch today

# Summary

- Start understanding your launch today
- Measure — don’t estimate — performance

# Summary

- Start understanding your launch today
- Measure — don’t estimate — performance
- Track performance in all phases of development

# More Information

developer.apple.com/wwdc19/423

| | |
|---|---|
| Improving Battery Life and Performance | WWDC 2019 |
| High Performance Auto Layout | WWDC 2018 |
| Modernizing Grand Central Dispatch Usage | WWDC 2017 |


WWDC19