

#WWDC18

What's New in Energy Debugging

Phillip Azar, Apple/Battery Life

David Choi, Apple/Battery Life





Review general battery life concepts

Review tools for energy debugging

New features for energy debugging

Review general battery life concepts

Review tools for energy debugging

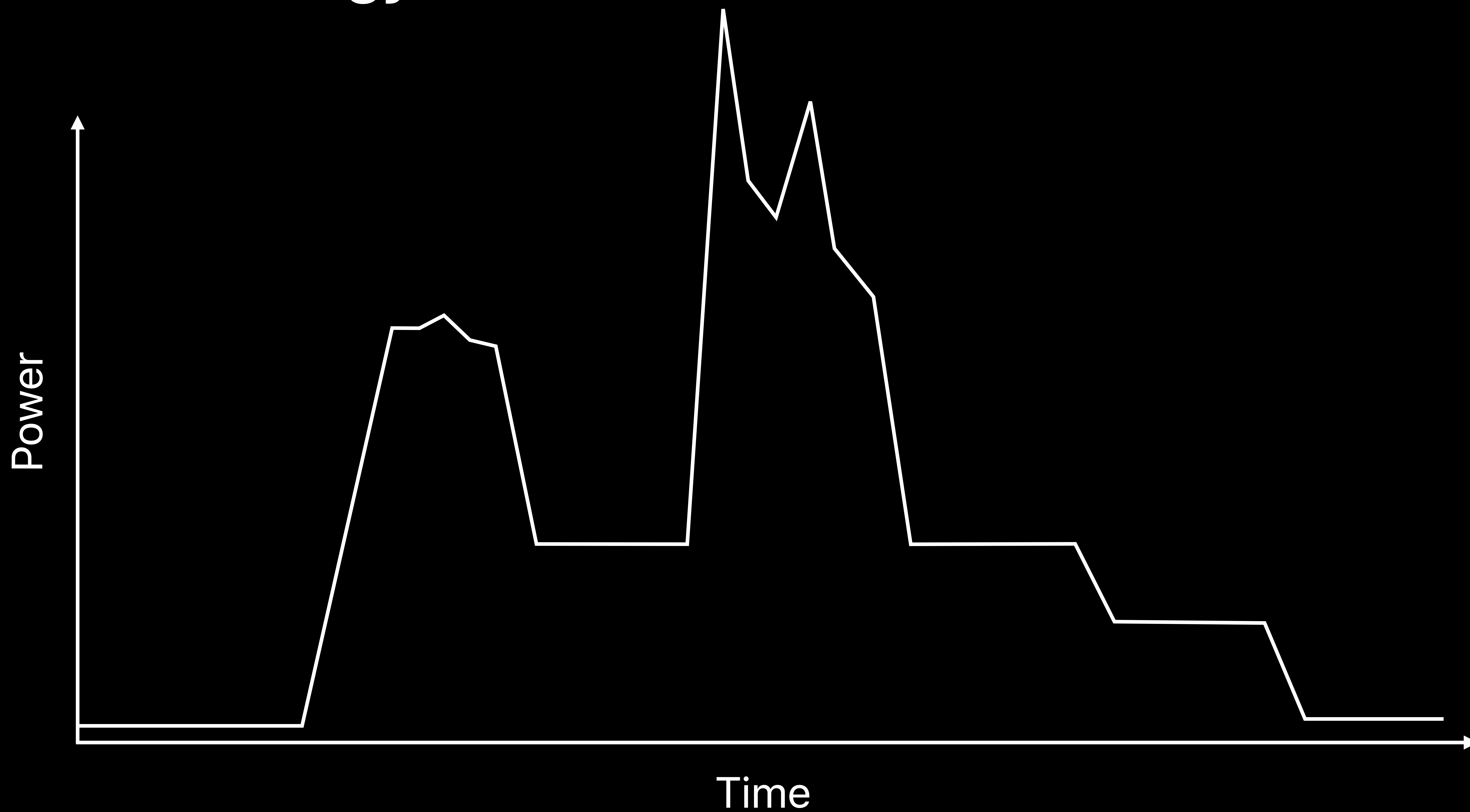
New features for energy debugging

What Is Energy?

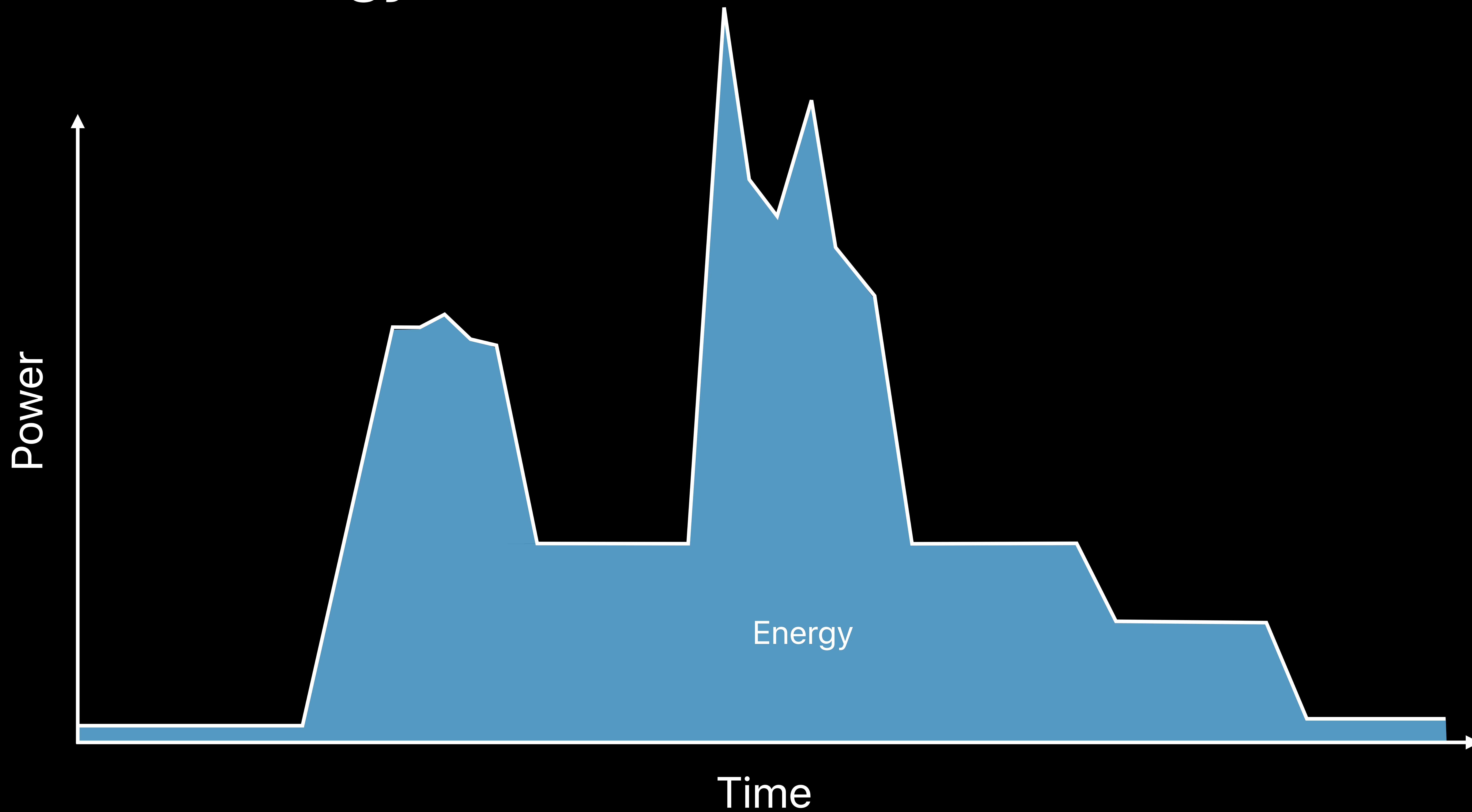
What Is Energy?

Power X Time

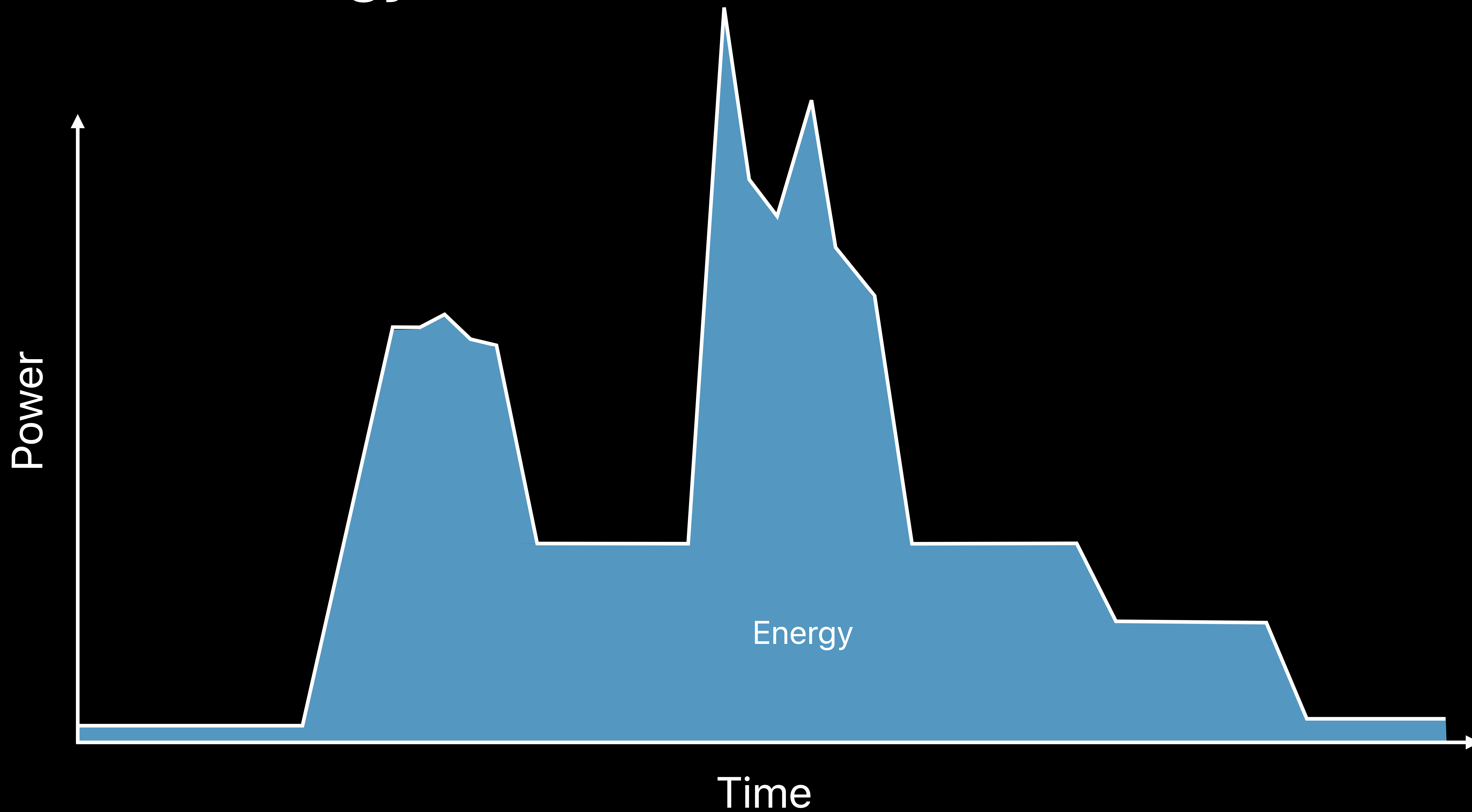
What Is Energy?



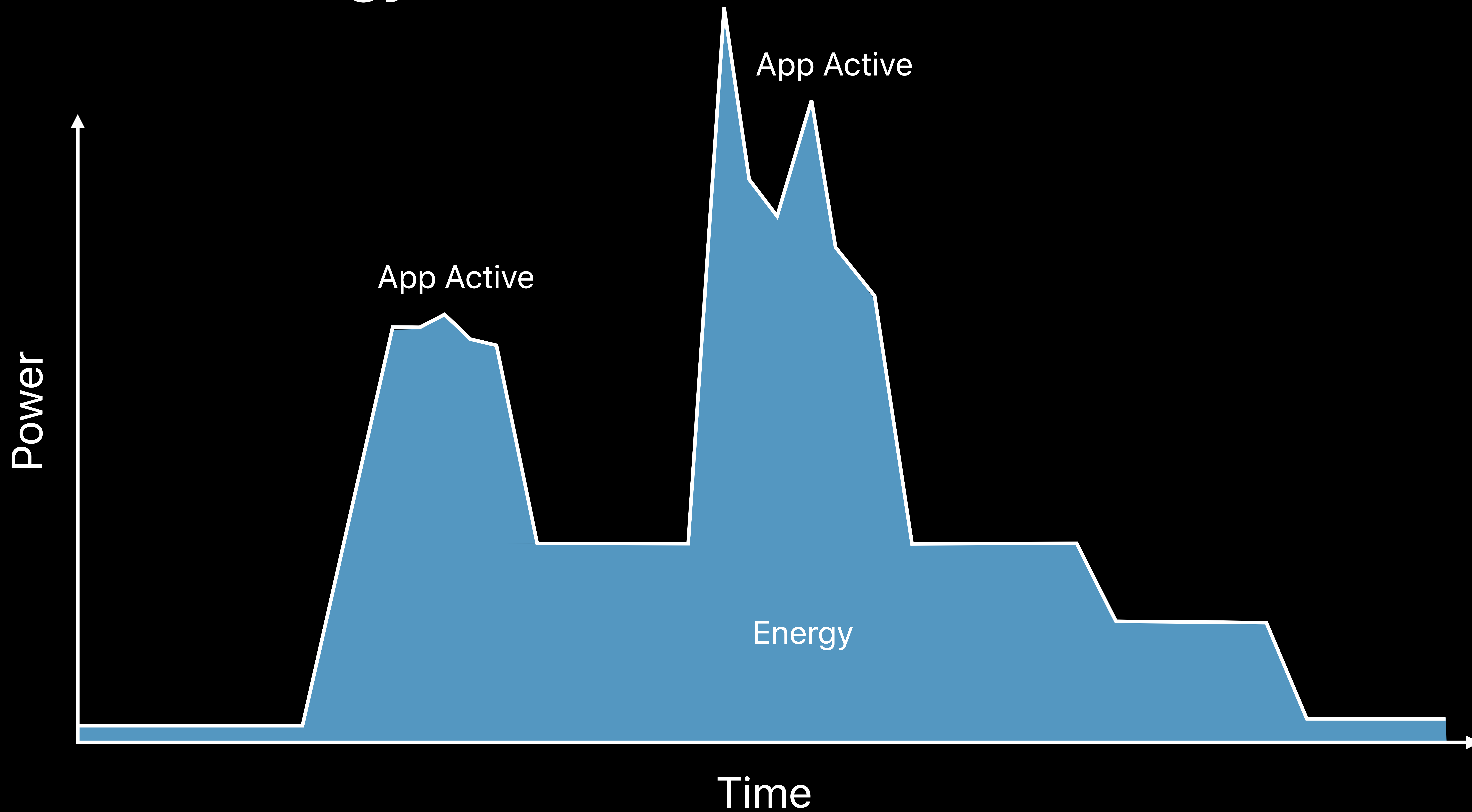
What Is Energy?



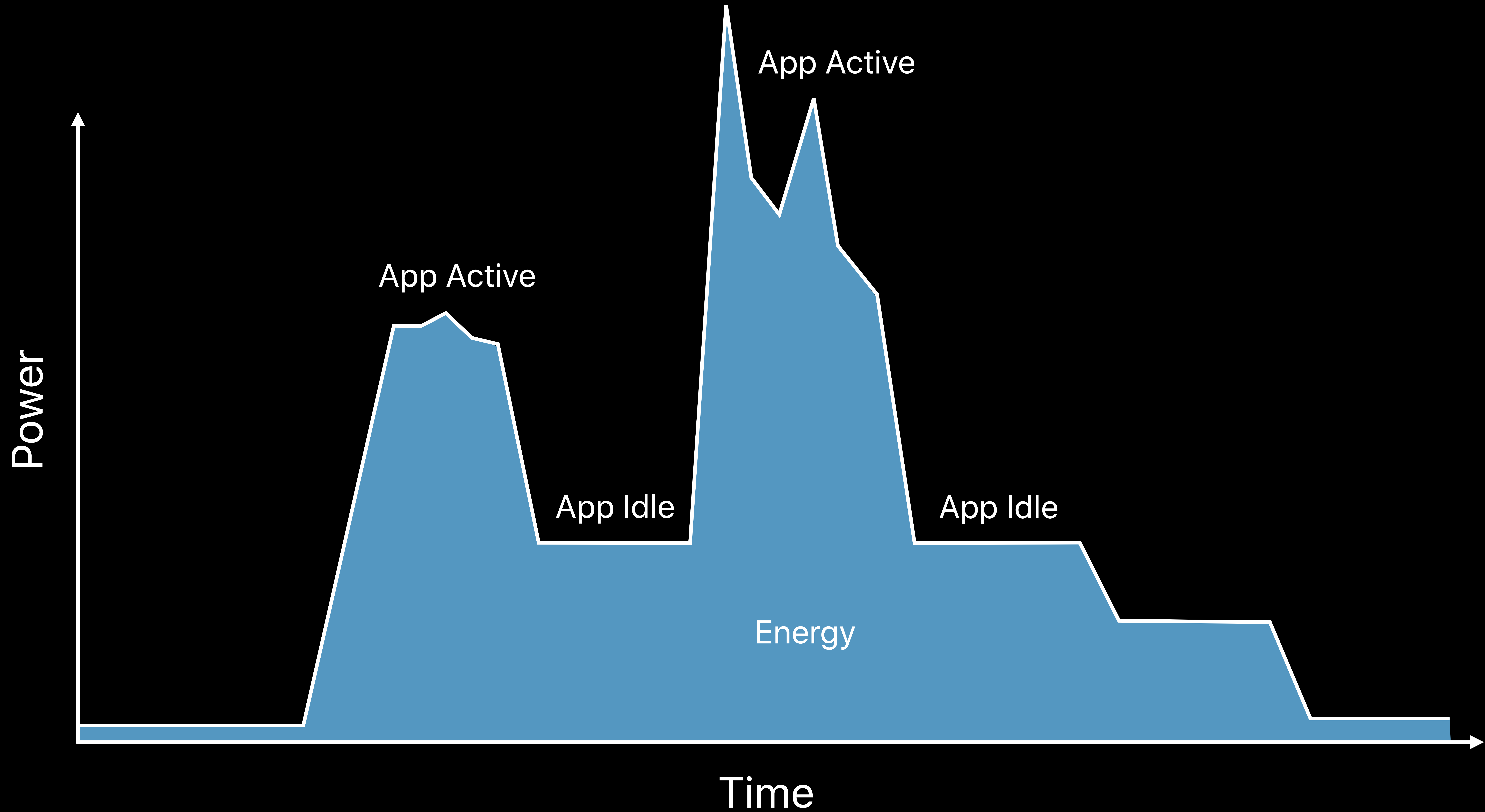
What Is Energy?



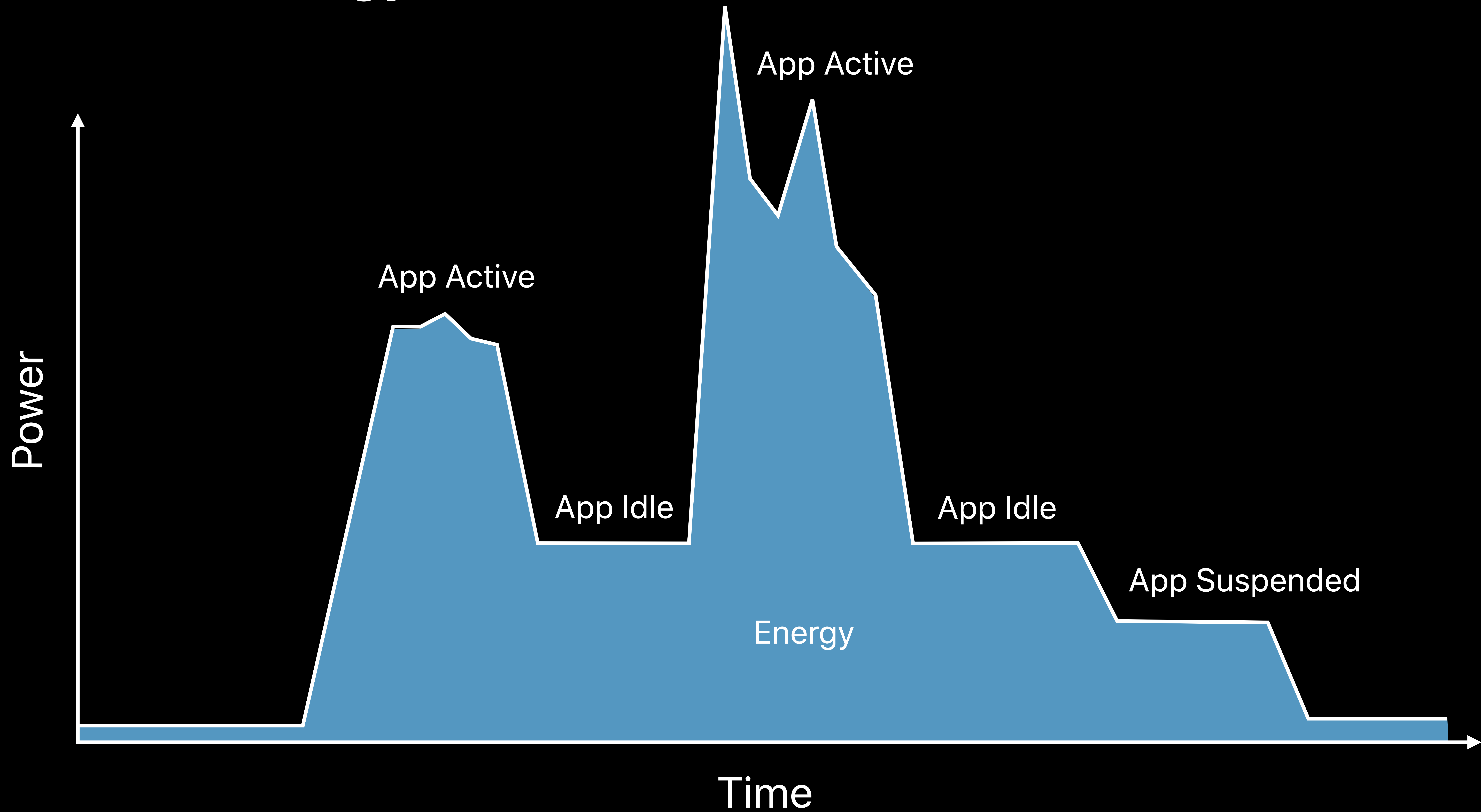
What Is Energy?



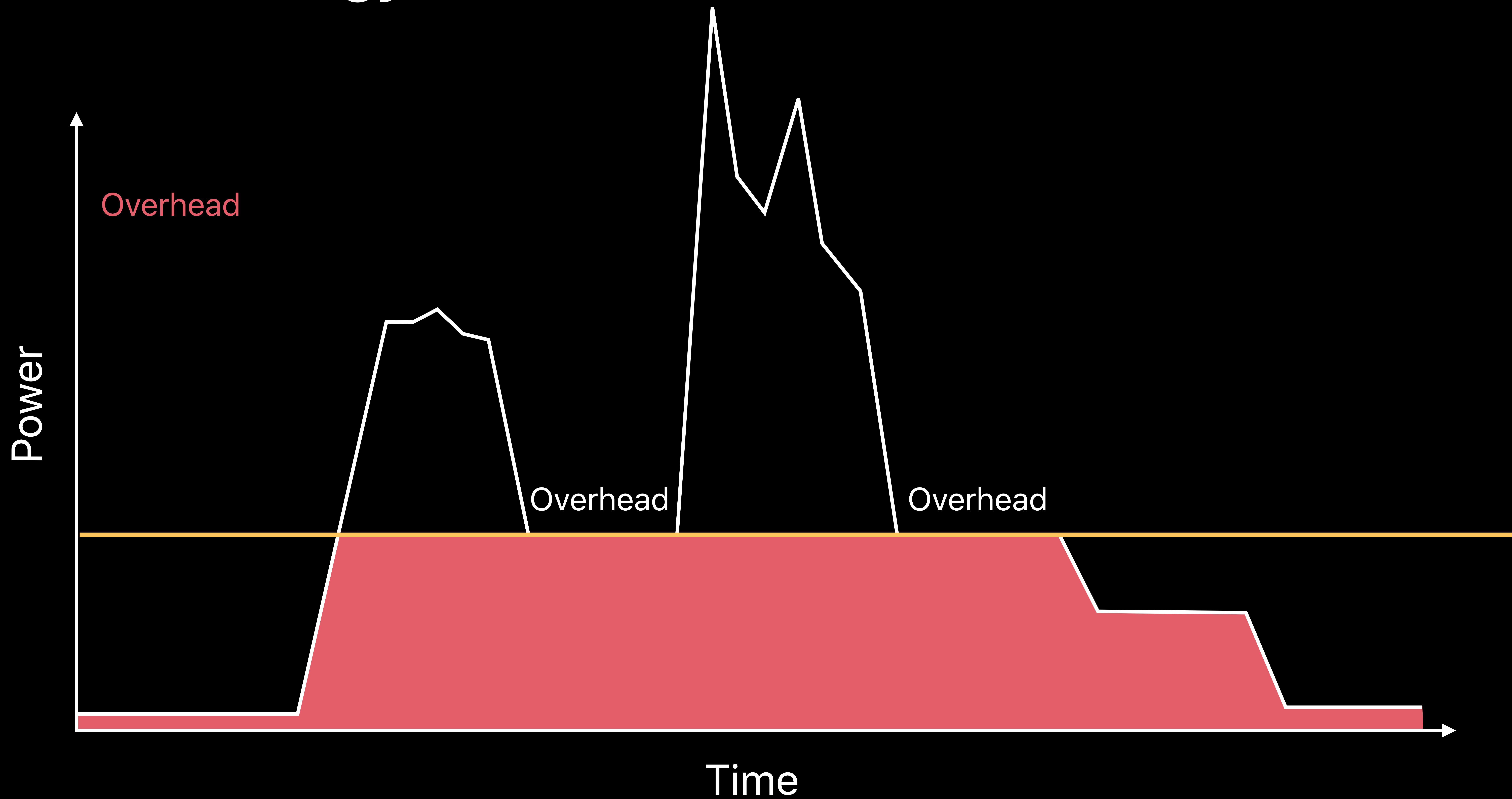
What Is Energy?



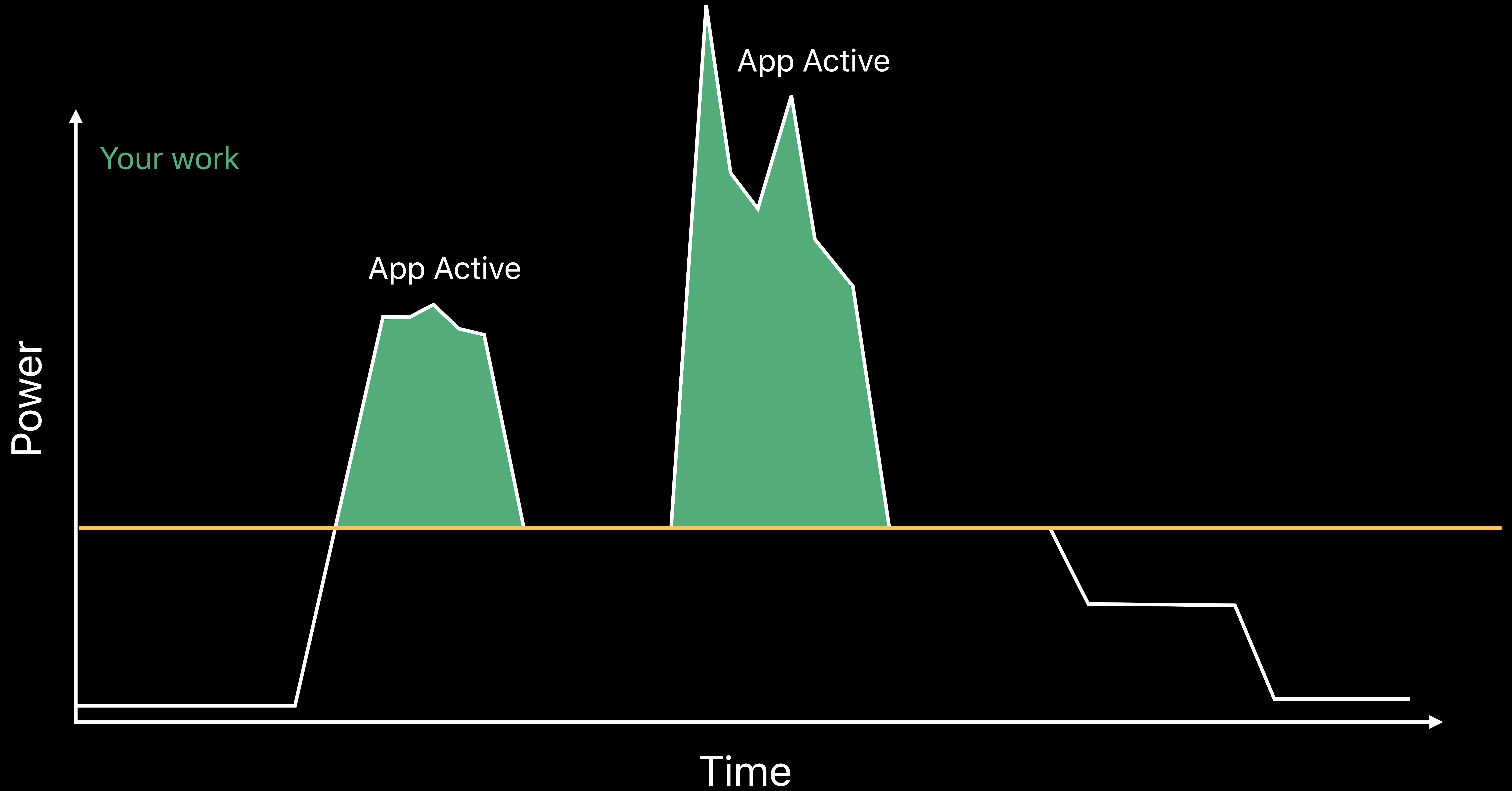
What Is Energy?



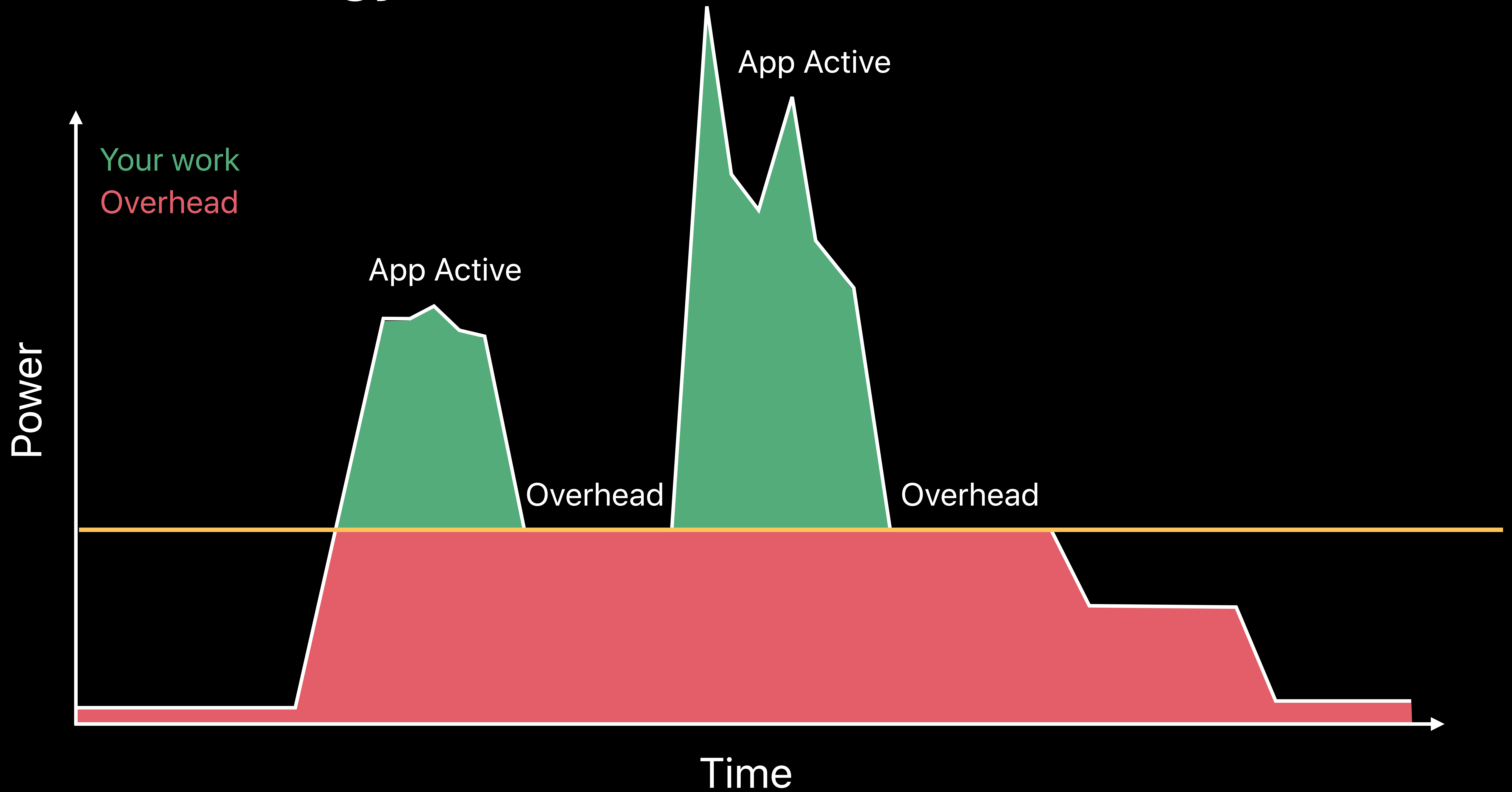
What Is Energy?



What Is Energy?



What Is Energy?



What Consumes Energy?

What Consumes Energy?



Processing



Networking



Location



Graphics

A dark gray rounded square icon with a lighter gray border. Inside the square, the letters "CPU" are written in a bold, white, sans-serif font.

CPU

Processing



Processing

The "work horse"



Processing

The "work horse"

Workload-dependent



Processing

The “work horse”

Workload-dependent

More operations by your app = more energy





Networking



Networking

Cellular, Wi-Fi, Bluetooth



Networking

Cellular, Wi-Fi, Bluetooth

Traffic-dependent



Networking

Cellular, Wi-Fi, Bluetooth

Traffic-dependent

More network requests = more energy





Location



Location

GPS, WiFi, Cellular



Location

GPS, WiFi, Cellular

Accuracy, frequency-dependent



Location

GPS, WiFi, Cellular

Accuracy, frequency-dependent

More time spent tracking location =
more energy





Graphics



Graphics

Animations and UI



Graphics

Animations and UI

Complexity-dependent



Graphics

Animations and UI

Complexity-dependent

More rendering = more energy





Processing



Networking



Location



Graphics



Processing



Networking



Location

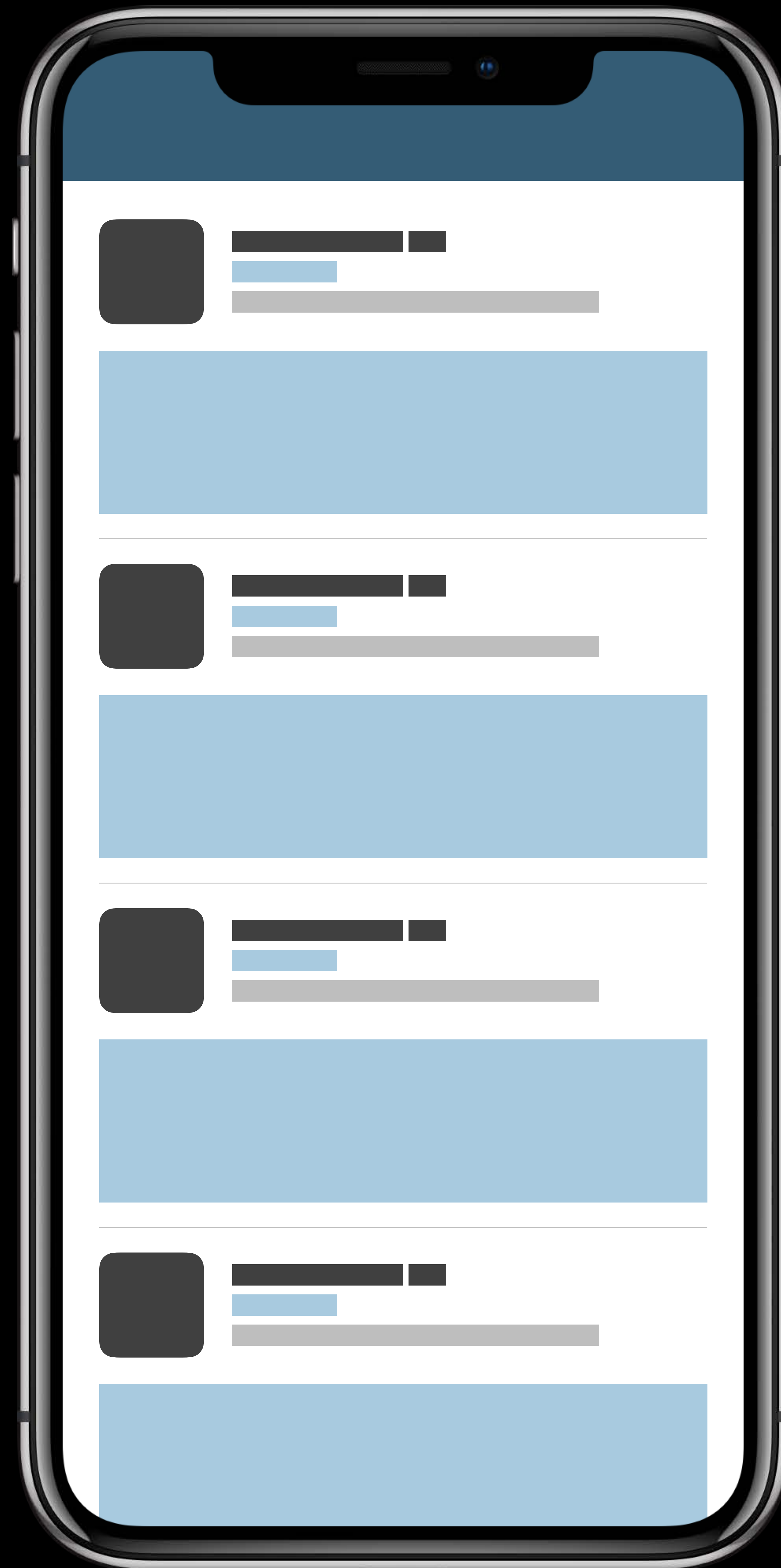


Graphics

More work = More energy

**How Do We Optimize
Energy Consumption?**

Foreground



Foreground



Foreground

Focus on **providing value to the user**



Foreground

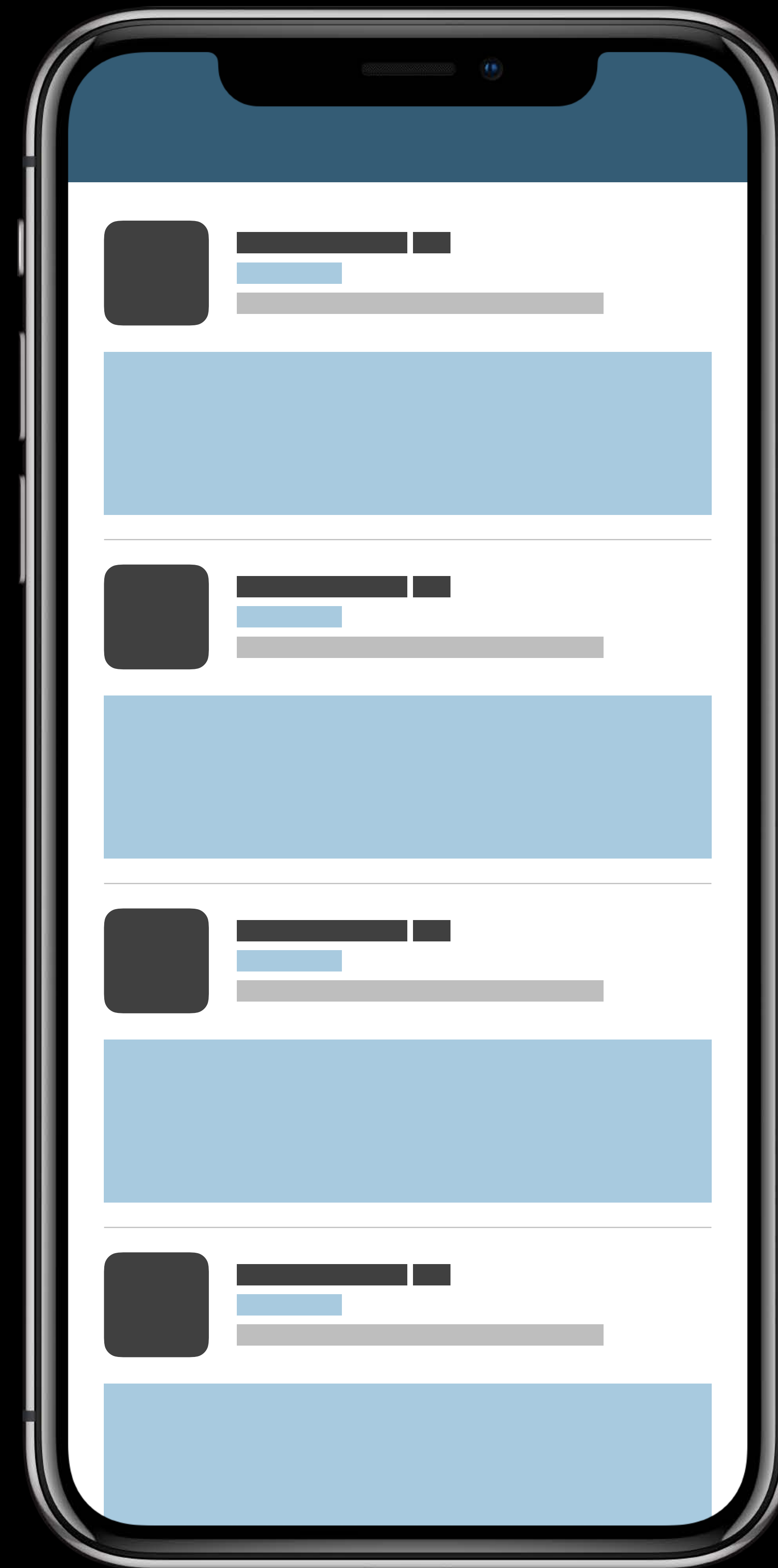
Focus on **providing value to the user**

- **Do required work only**



Example

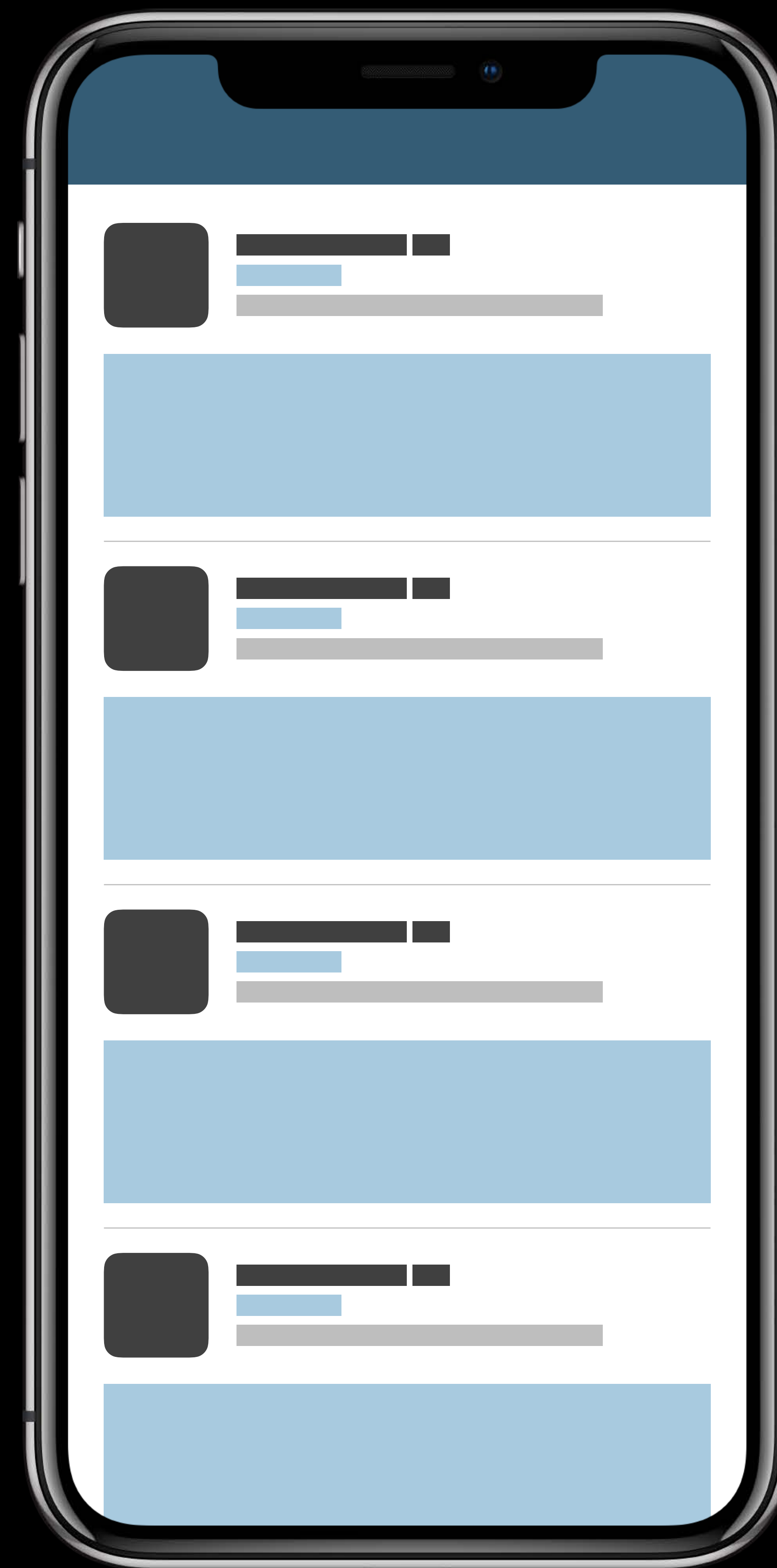
Media app feed



Example

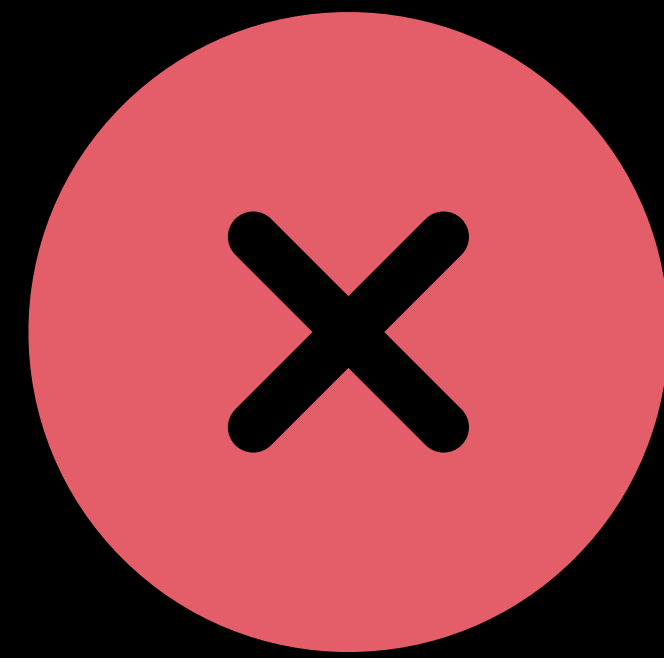
Media app feed

Timer based feed refresh

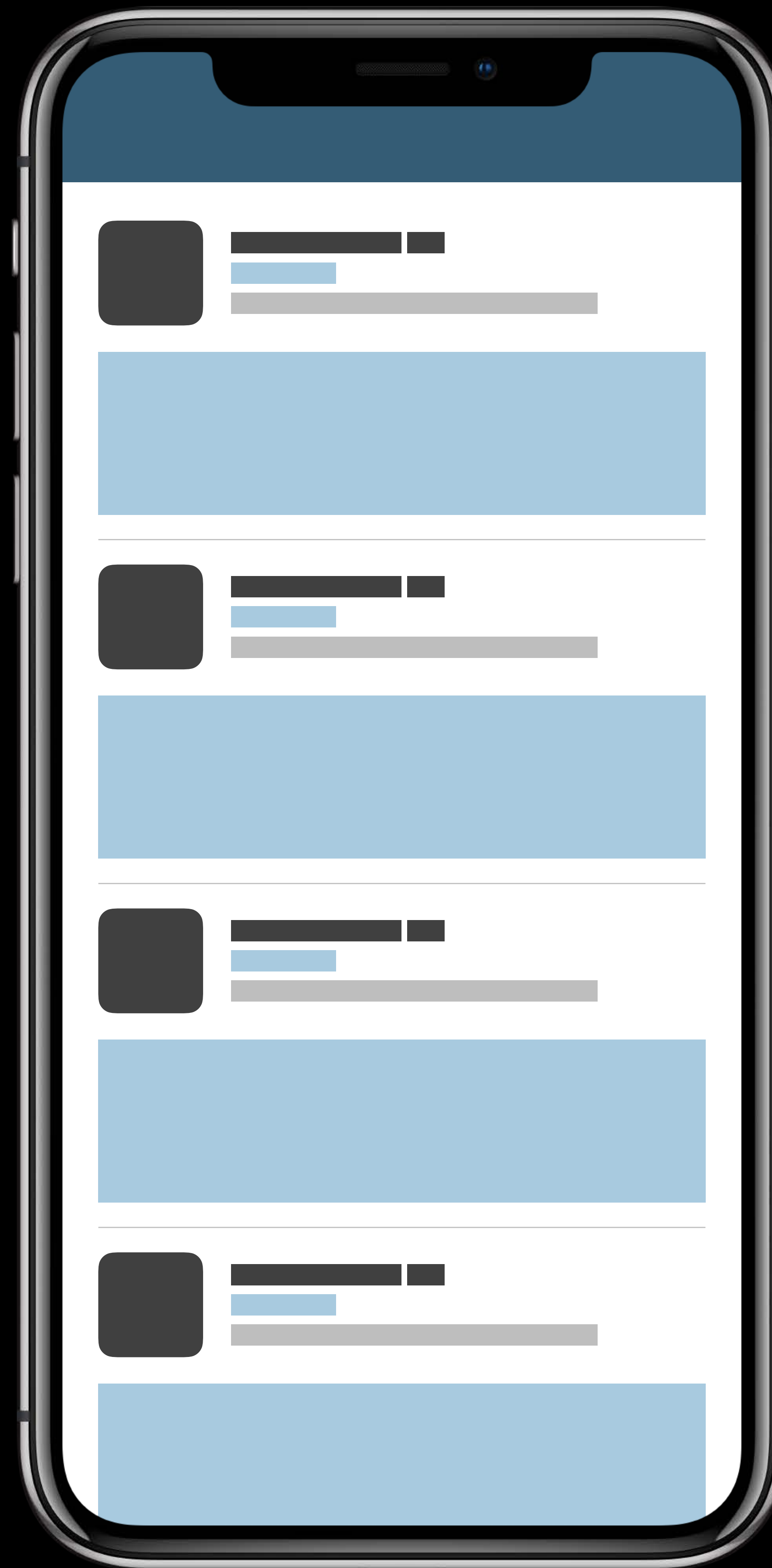


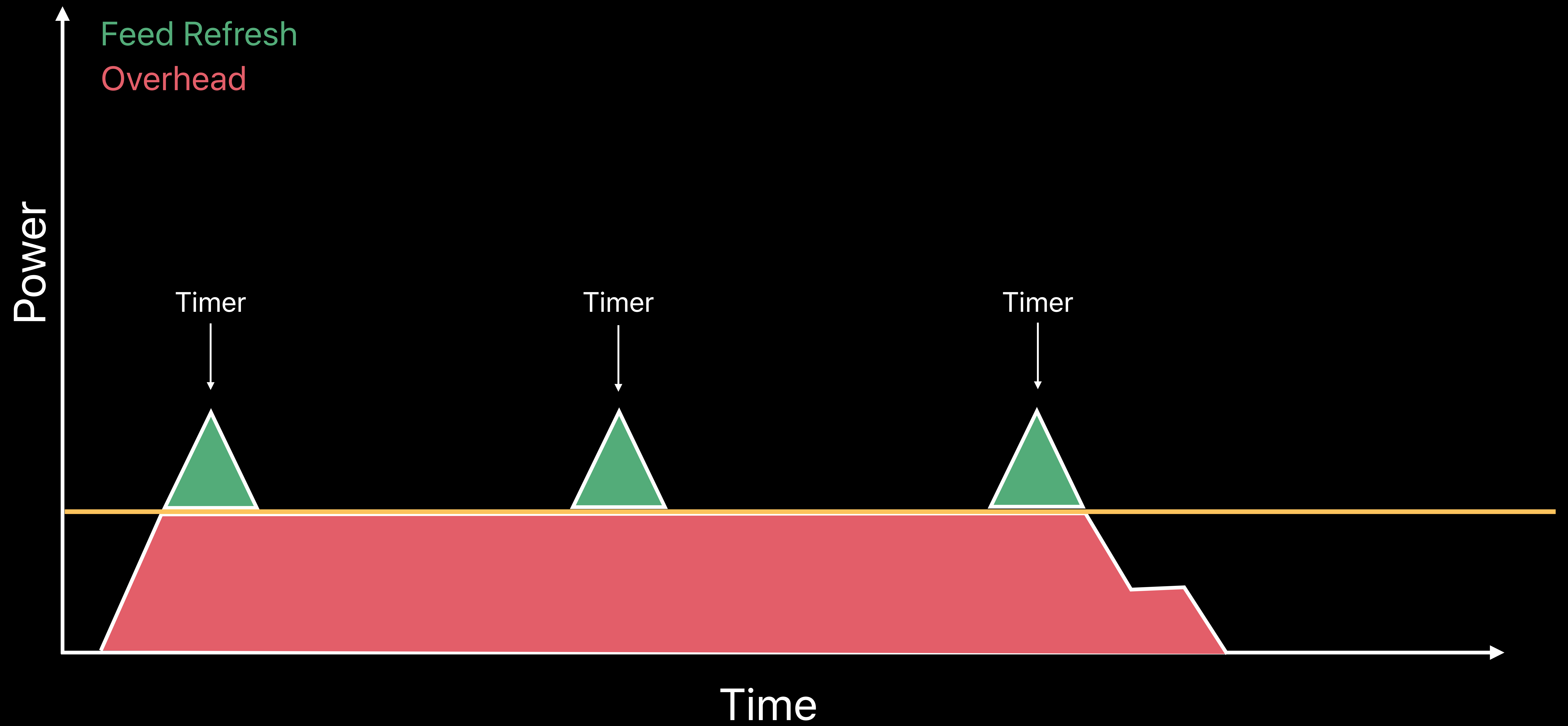
Example

Media app feed



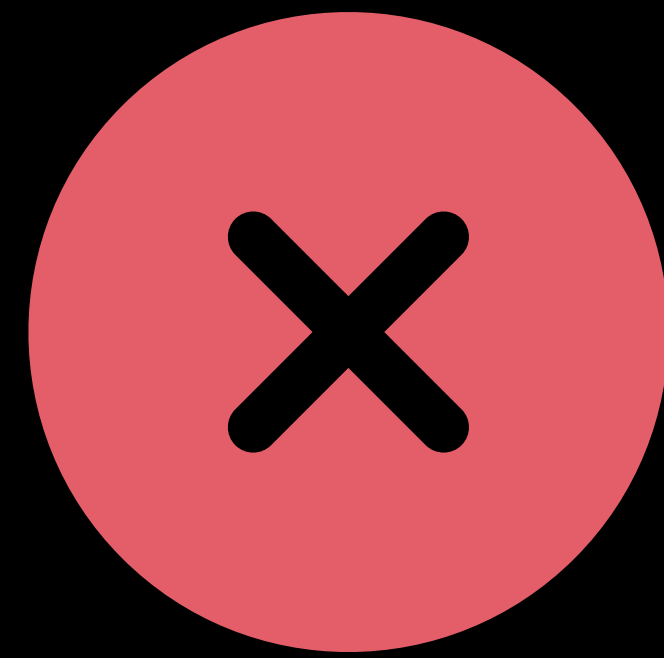
Timer based feed refresh



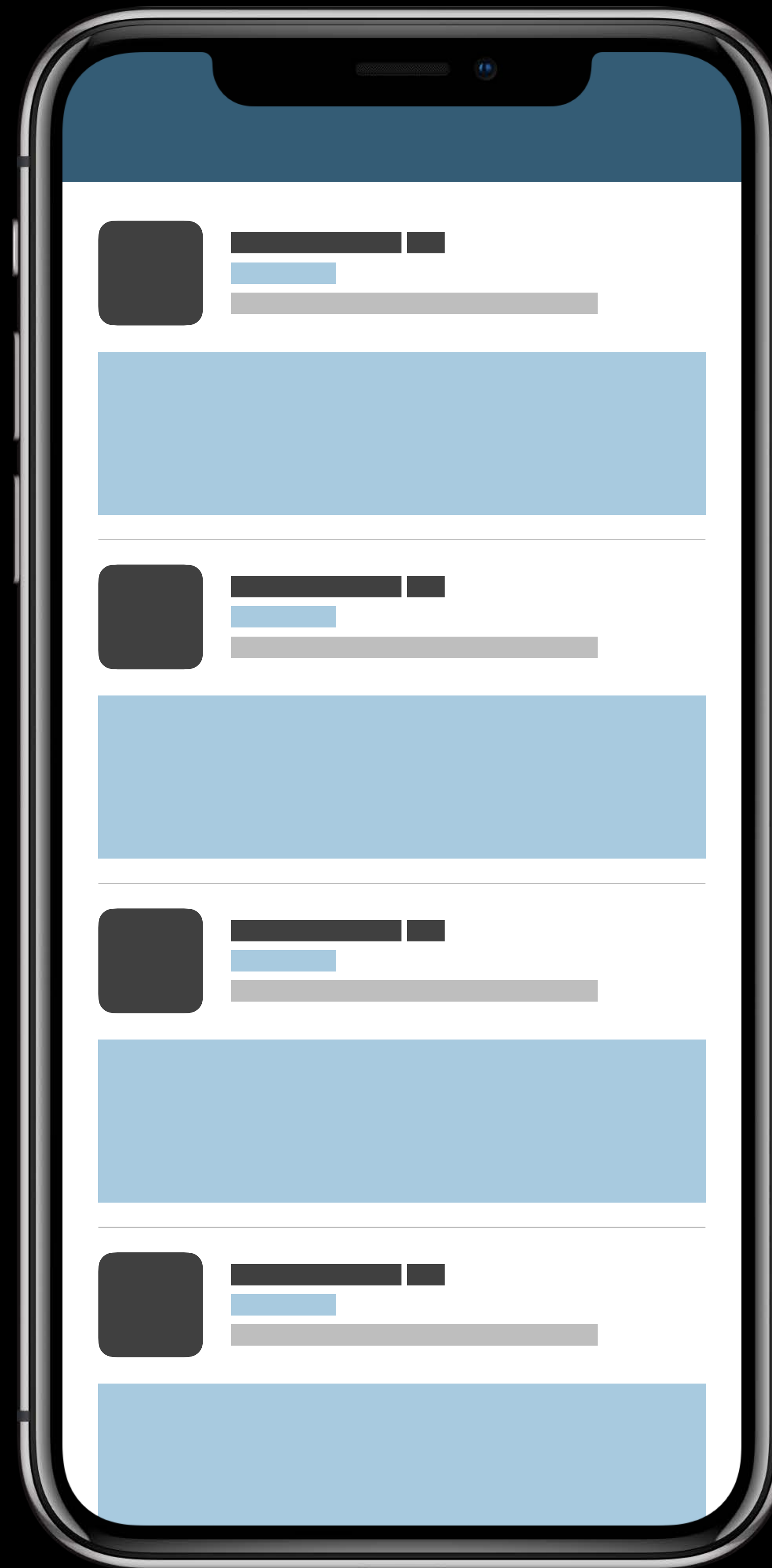


Example

Media app feed

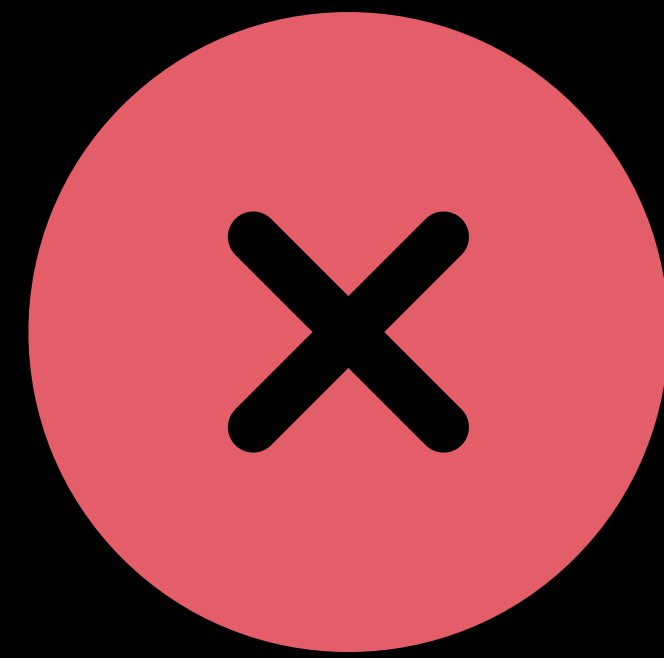


Timer based feed refresh

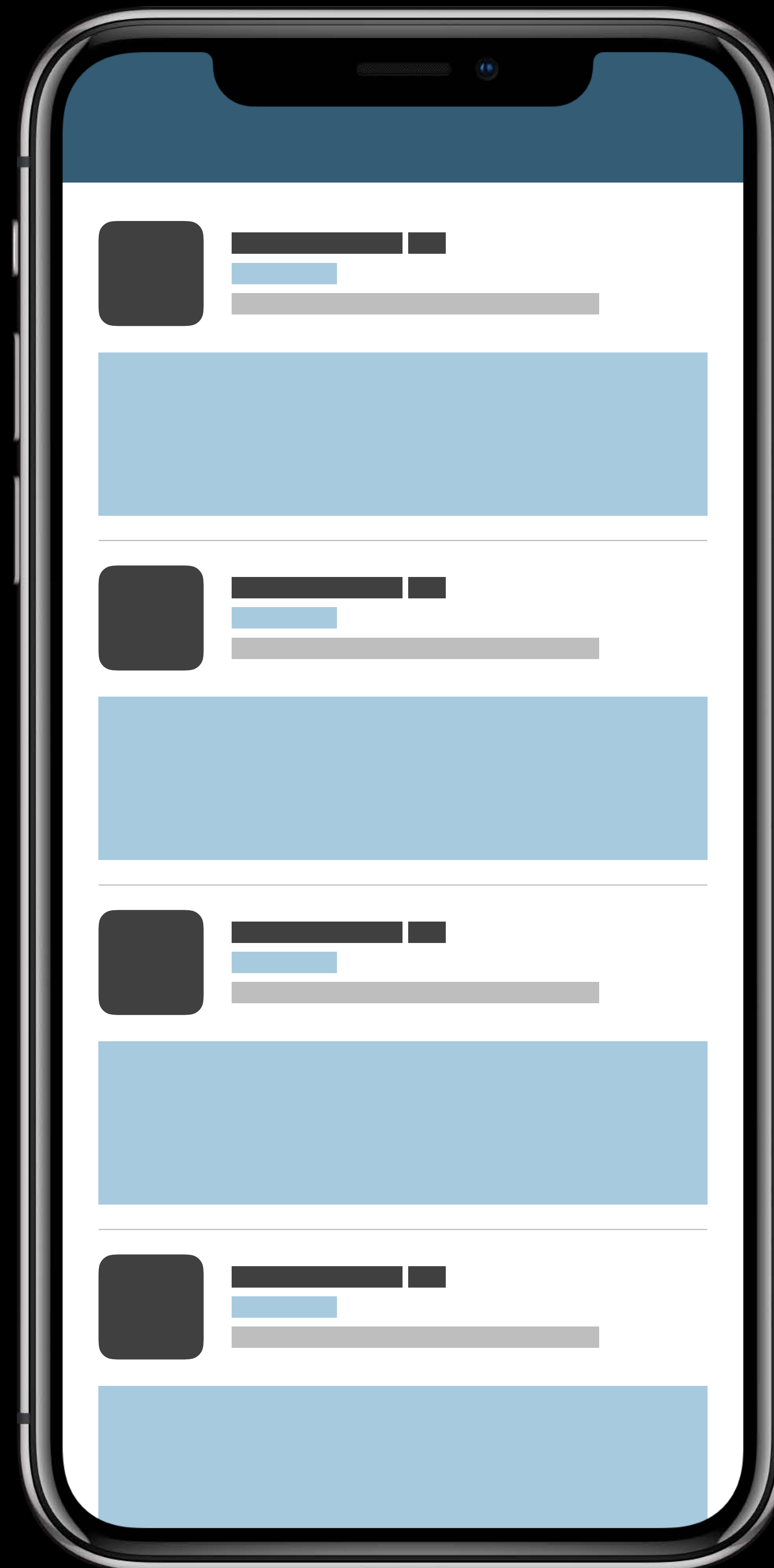


Example

Media app feed



Timer based feed refresh

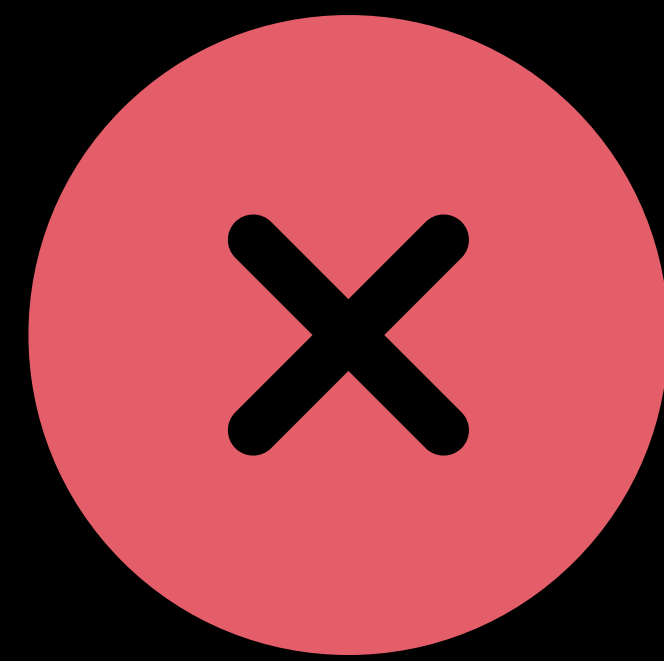


Feed refresh on demand

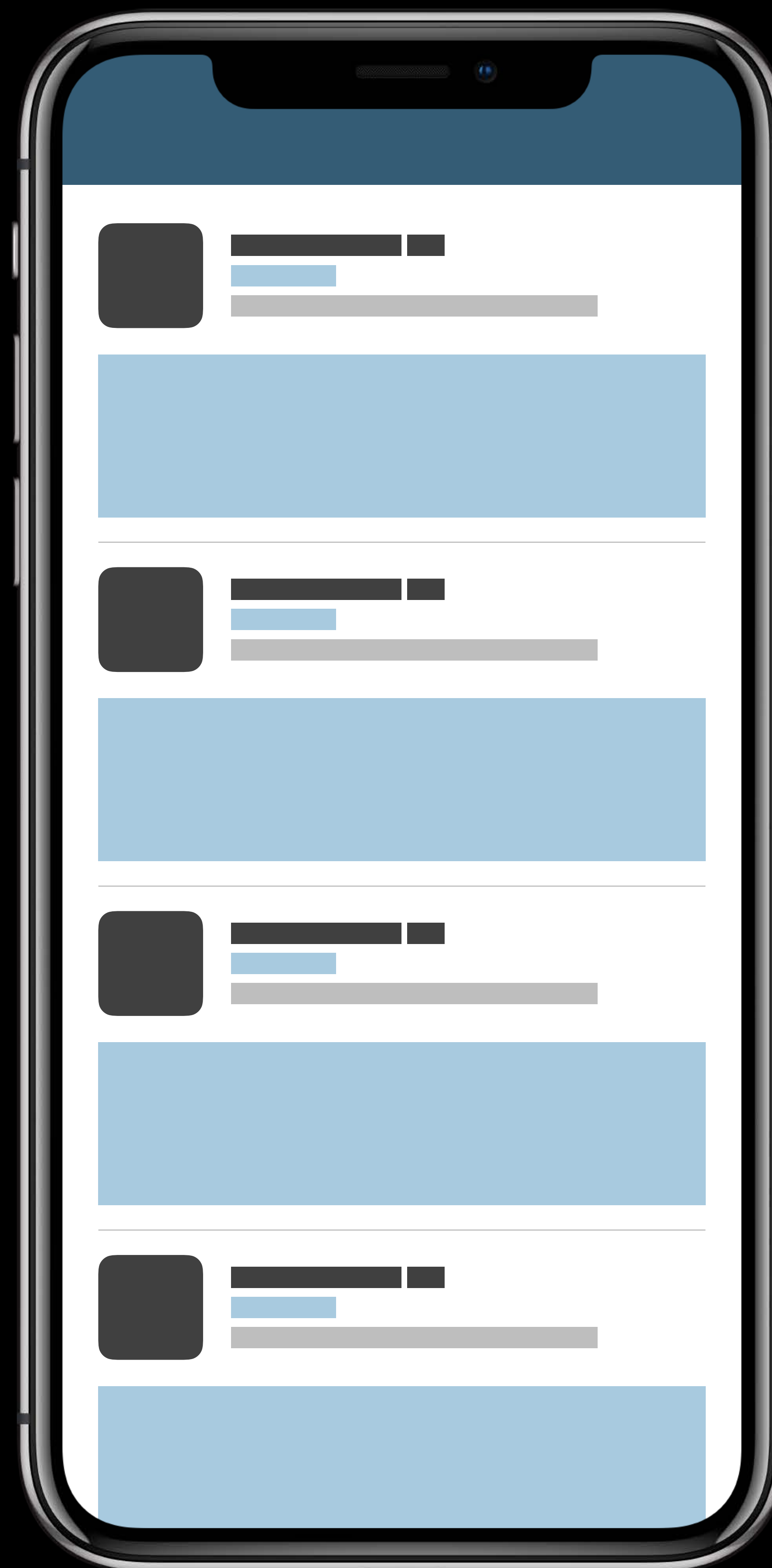
- Notification
- User interaction

Example

Media app feed

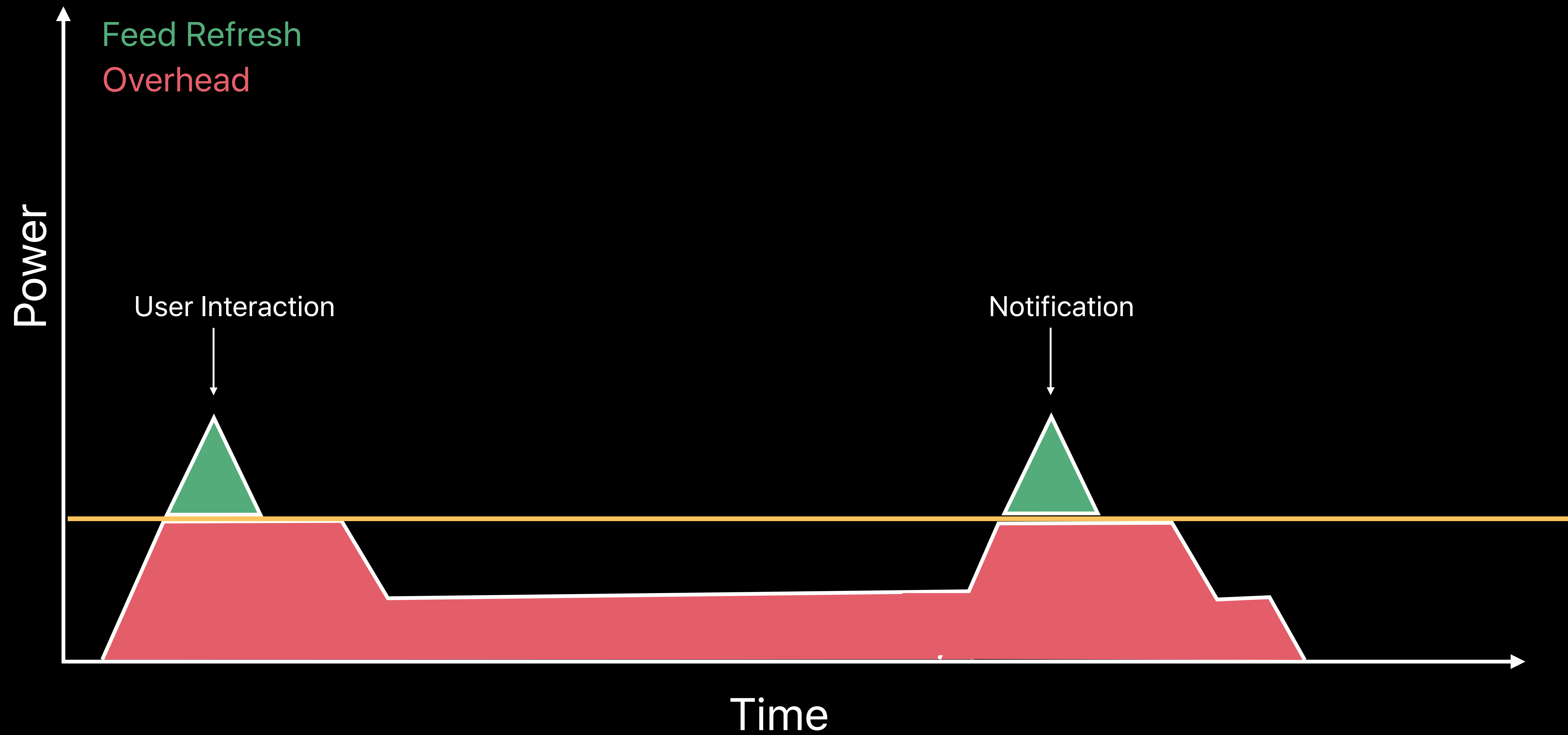


Timer based feed refresh



Feed refresh on demand

- Notification
- User interaction



Foreground

Focus on **providing value to the user**

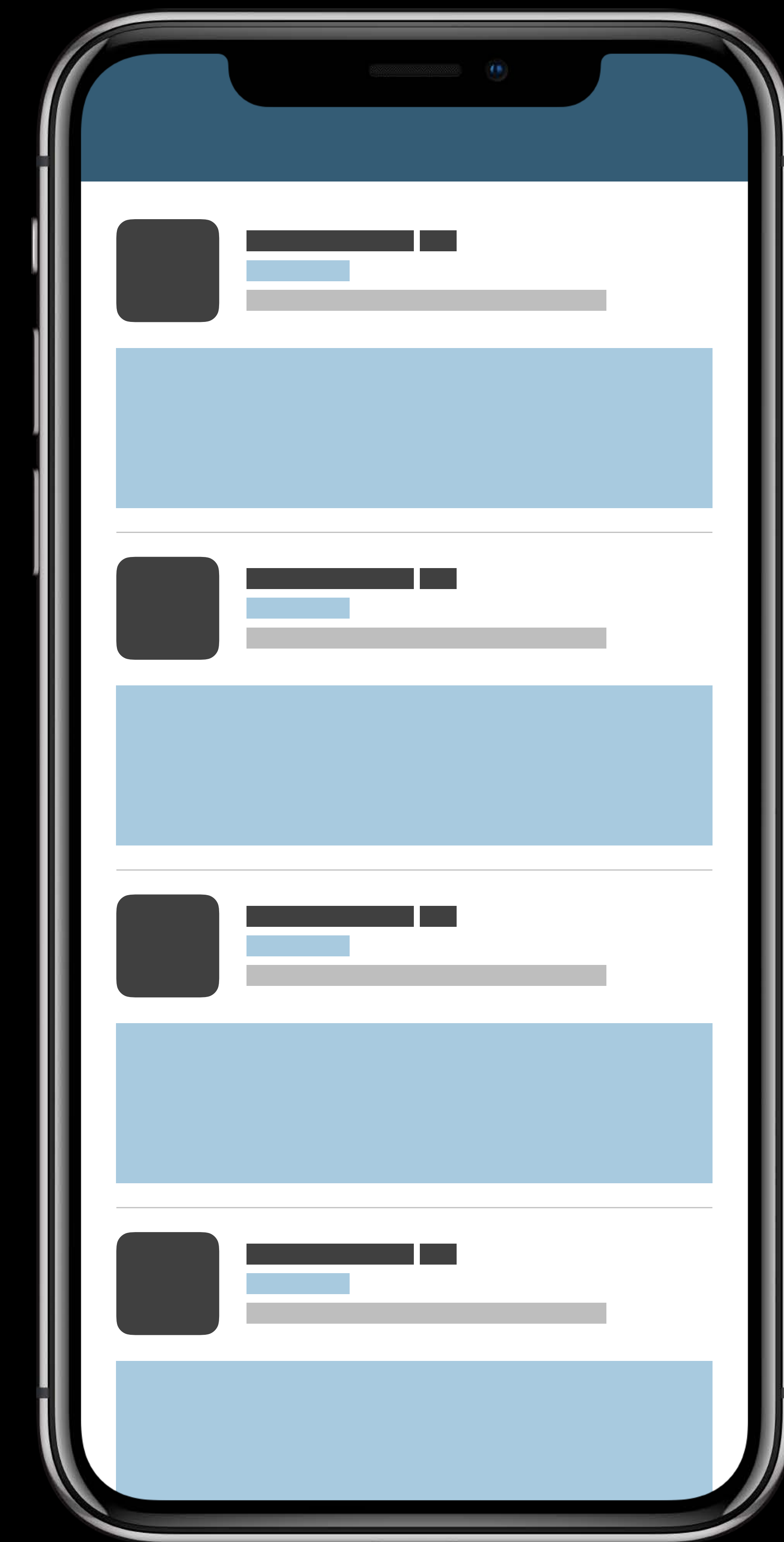
- Do work when requested



Foreground

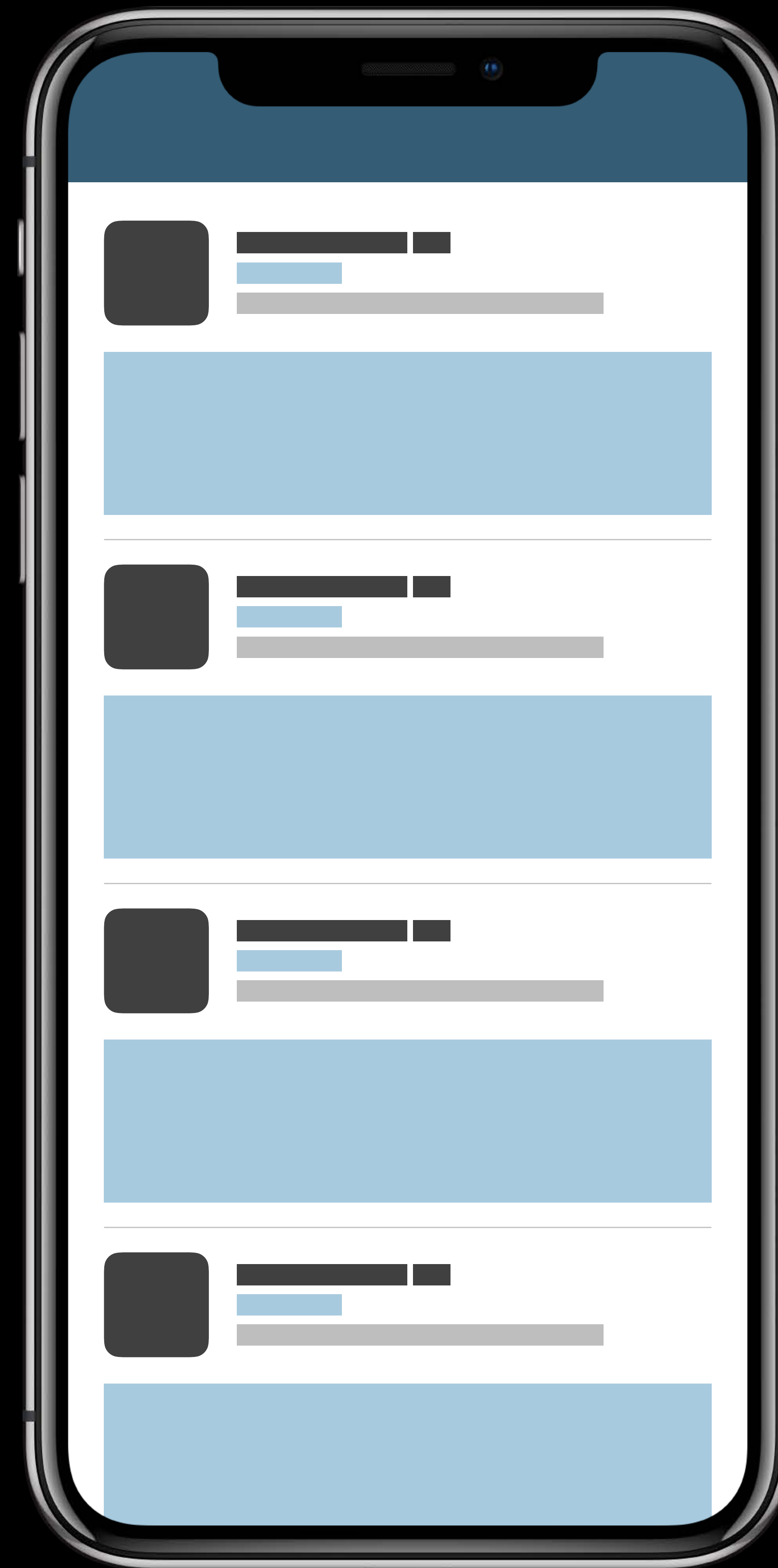
Focus on **providing value to the user**

- Do work when requested
- **Minimize complex UI**



Example

Video player app



Example

Video player app



Example

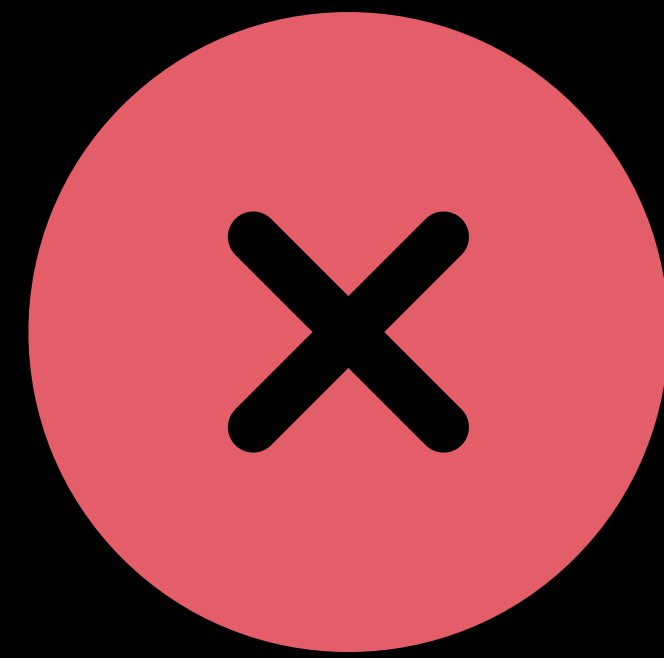
Video player app

Persistent UI controls



Example

Video player app

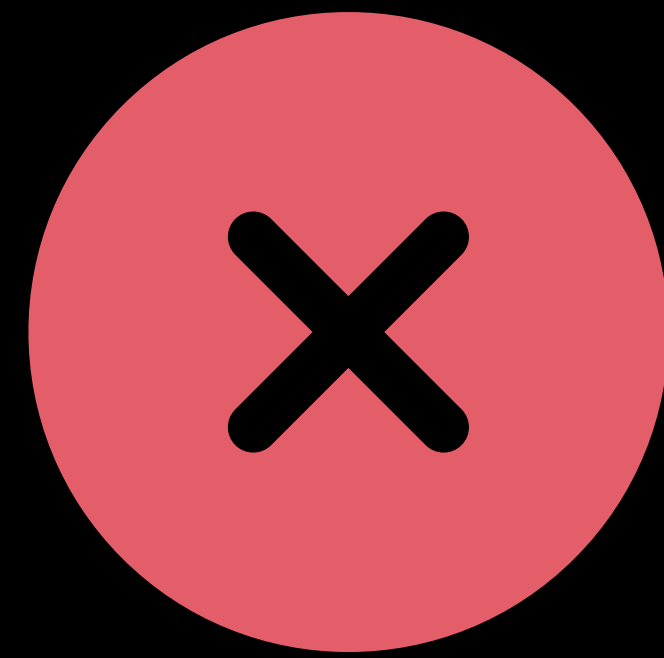


Persistent UI controls



Example

Video player app

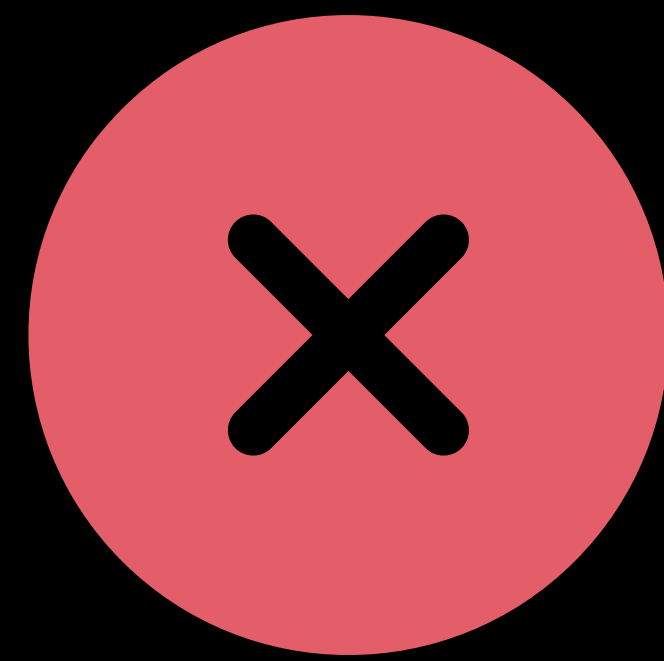


Persistent UI controls



Example

Video player app



Persistent UI controls

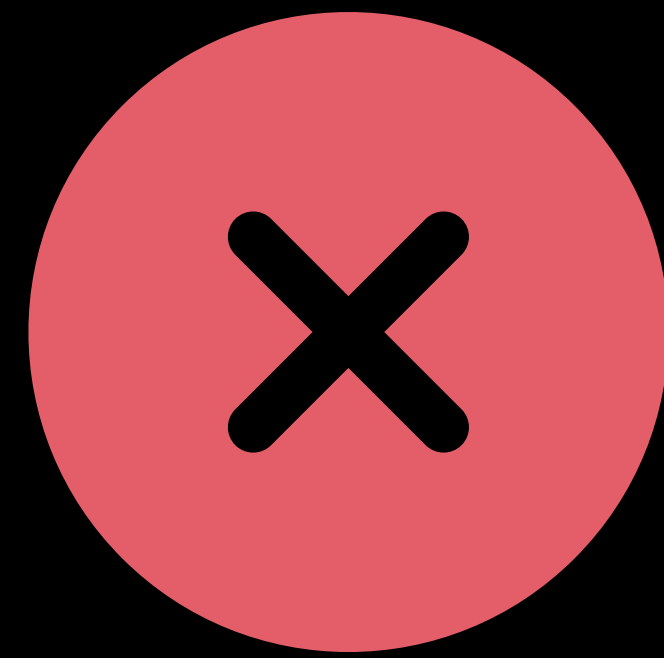


Auto dismissed UI controls

- No user interaction
- Timeout based

Example

Video player app



Persistent UI controls



Auto dismissed UI controls

- No user interaction
- Timeout based

Background



Background



Background

Focus on **minimizing workload**



Background

Focus on **minimizing workload**

- **Coalesce tasks**



Example

App analytics



Example

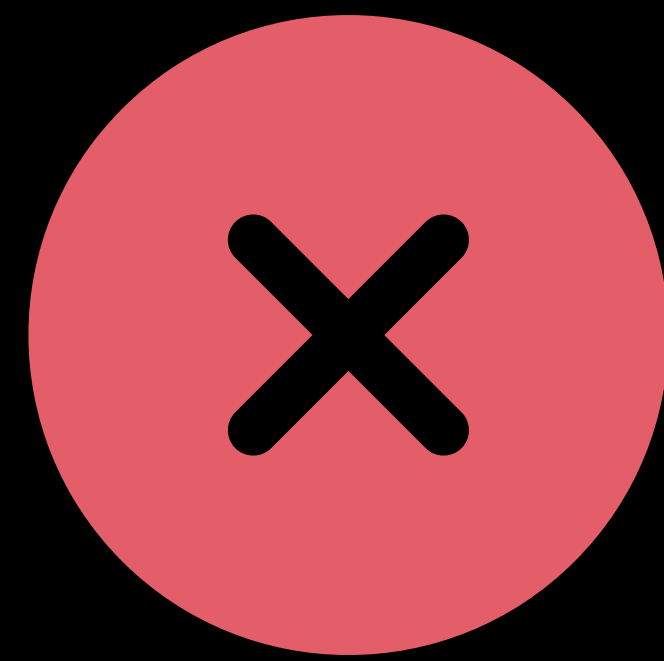
App analytics

Sending immediately



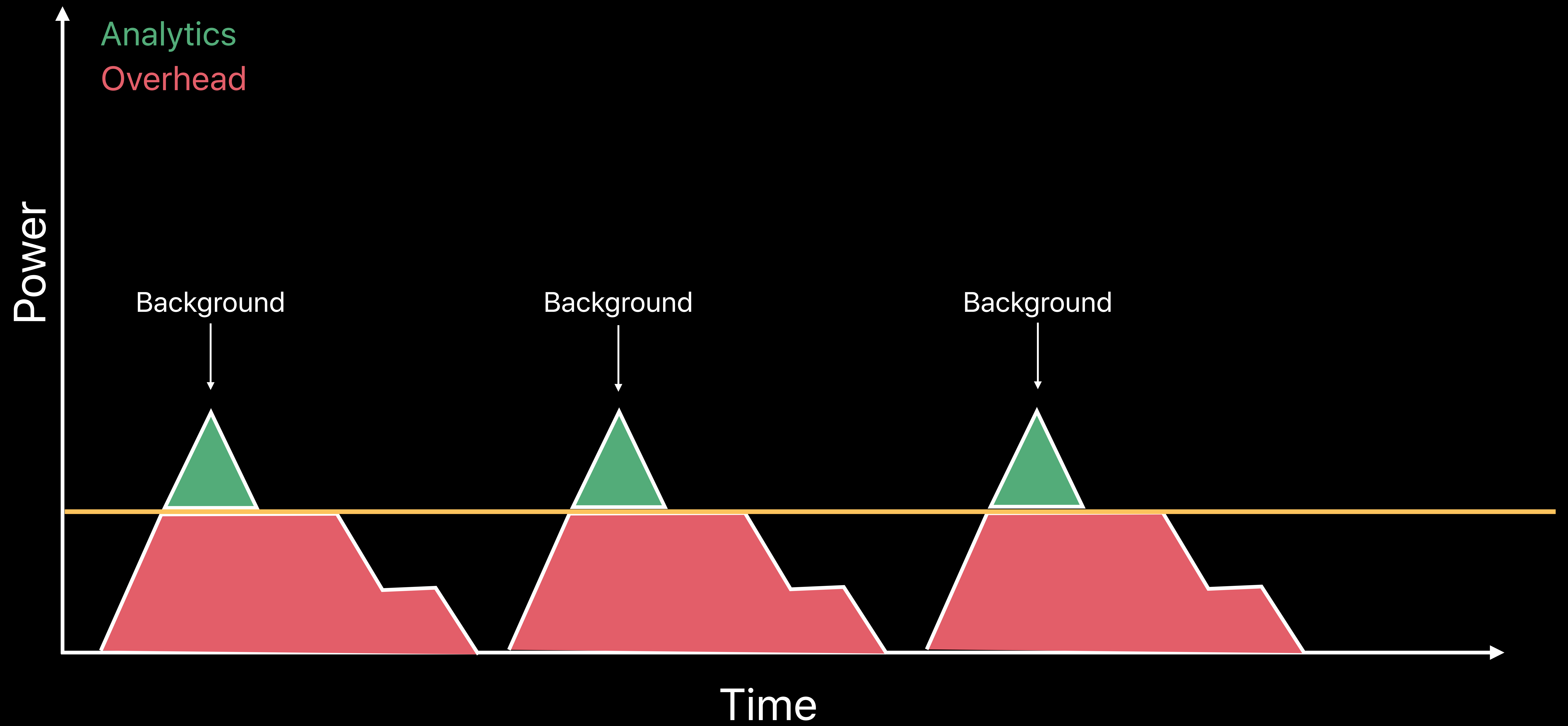
Example

App analytics



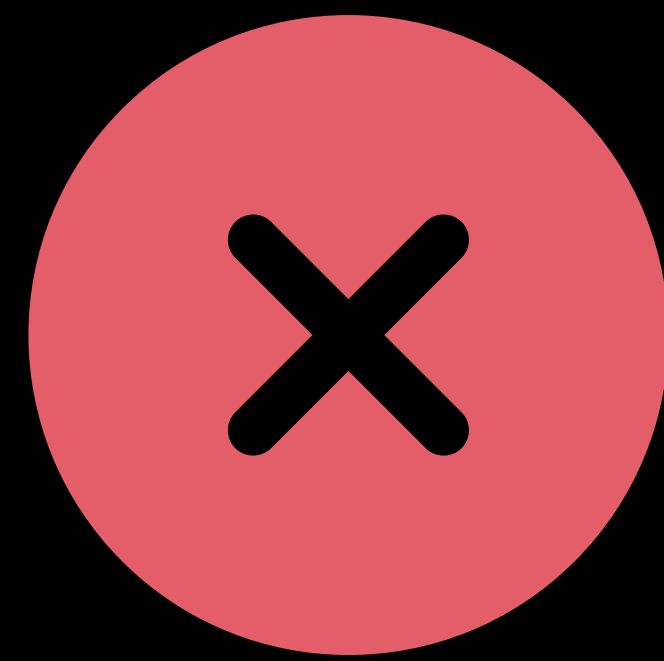
Sending immediately





Example

App analytics

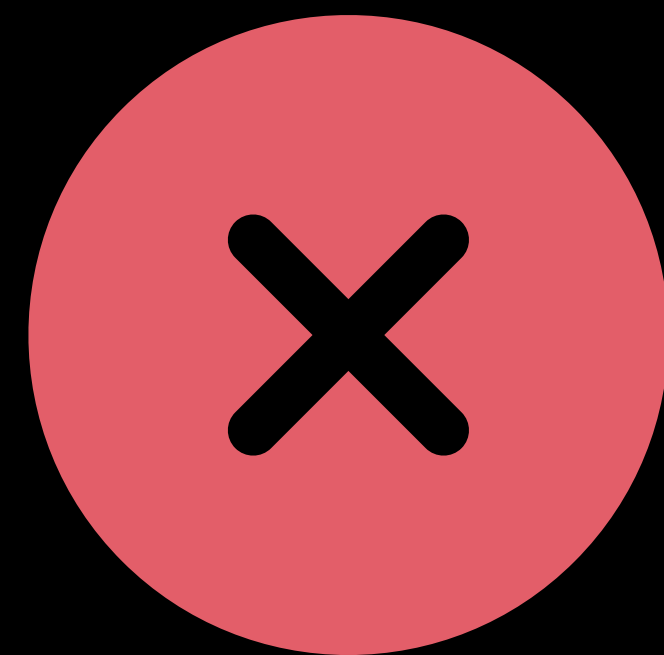


Sending immediately



Example

App analytics



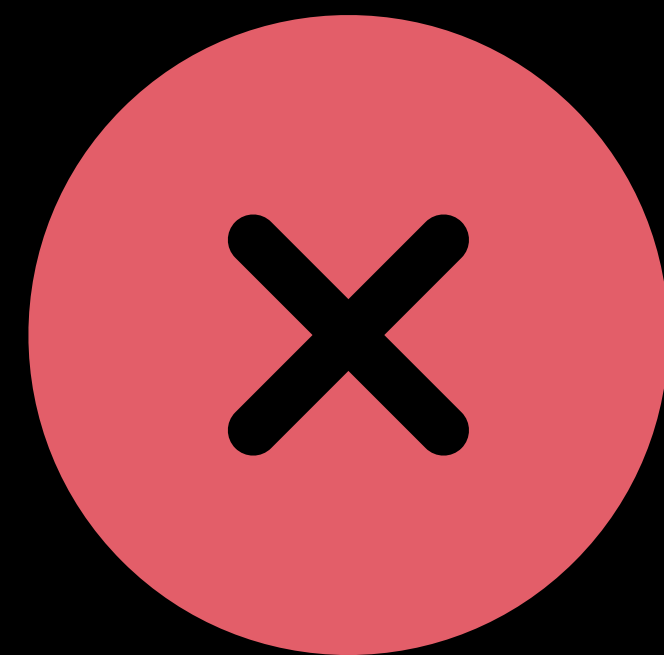
Sending immediately



Sending deferred batches

Example

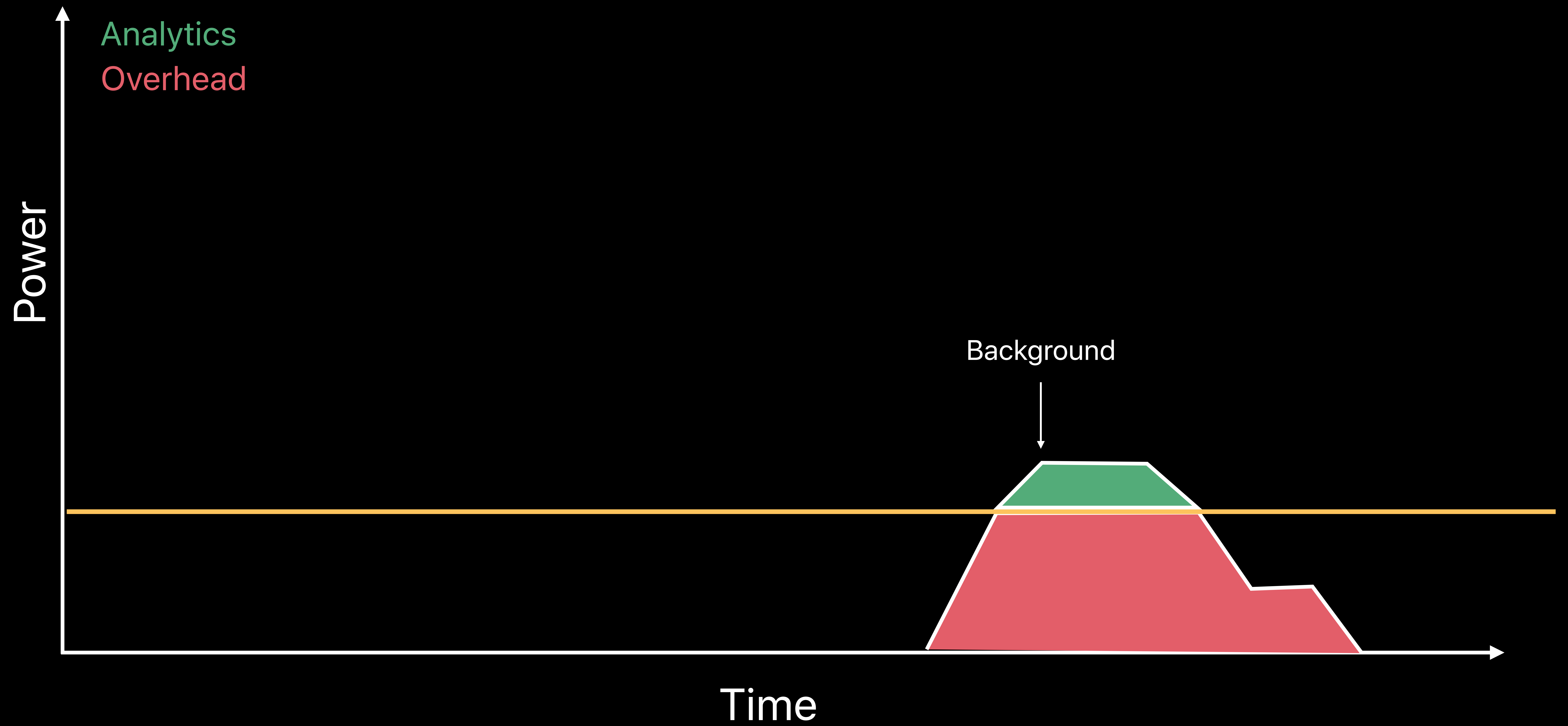
App analytics



Sending immediately



Sending deferred batches



Background

Focus on **minimizing workload**

- Coalesce tasks



Background

Focus on **minimizing workload**

- Coalesce tasks
- **End tasks quickly**



Example

Background modes



Example

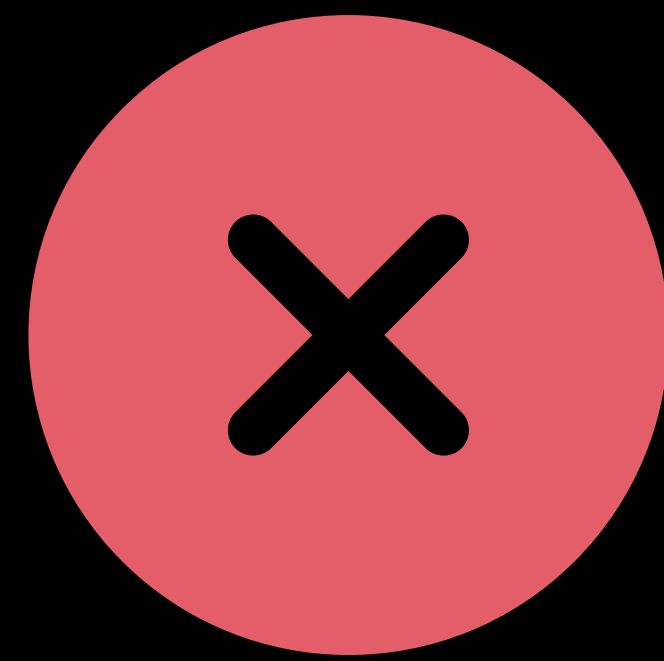
Background modes

Lettings tasks expire



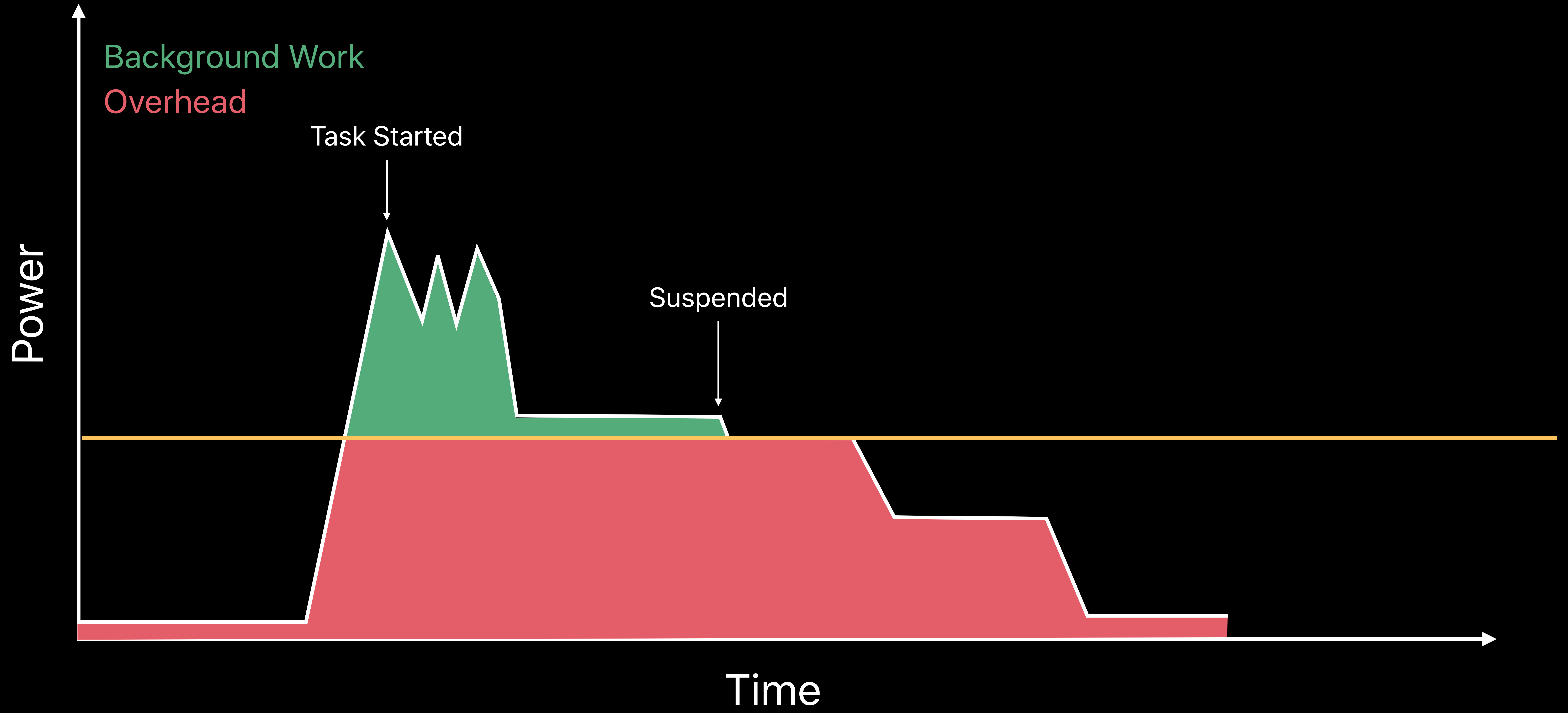
Example

Background modes



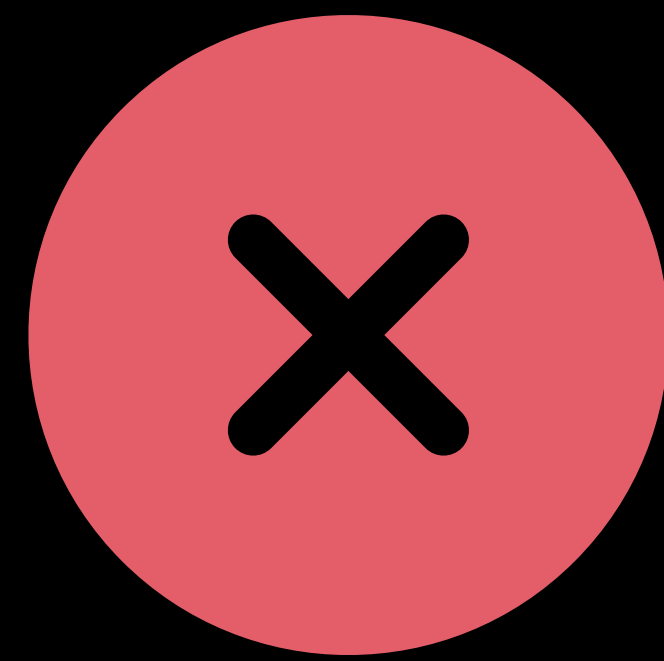
Lettings tasks expire





Example

Background modes

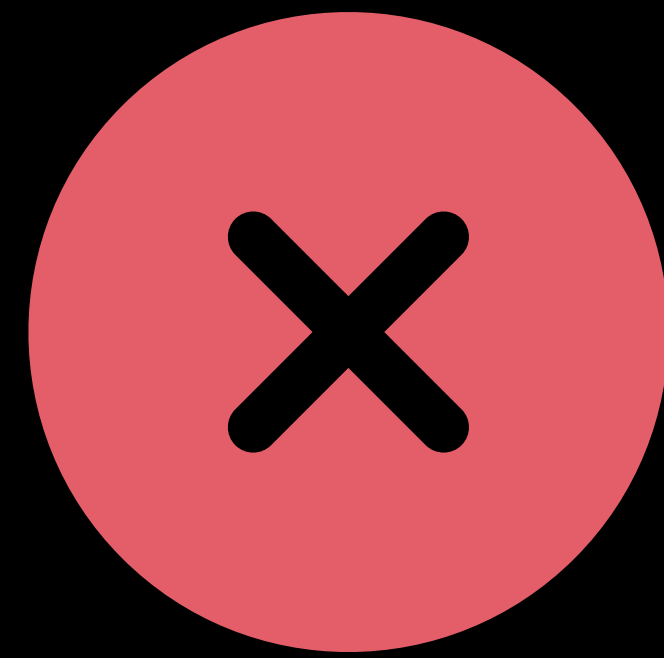


Lettings tasks expire



Example

Background modes



Lettings tasks expire

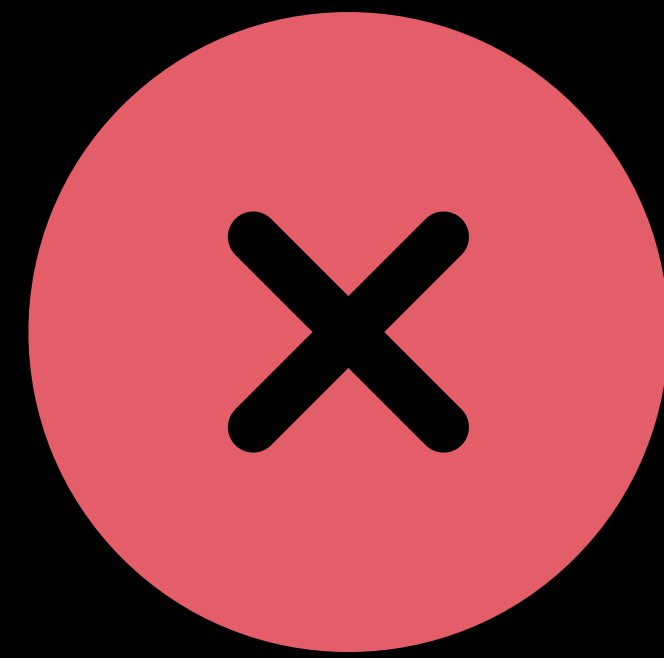


Calling completion handlers

- UIBackgroundTask
- VoIP
- Etc

Example

Background modes

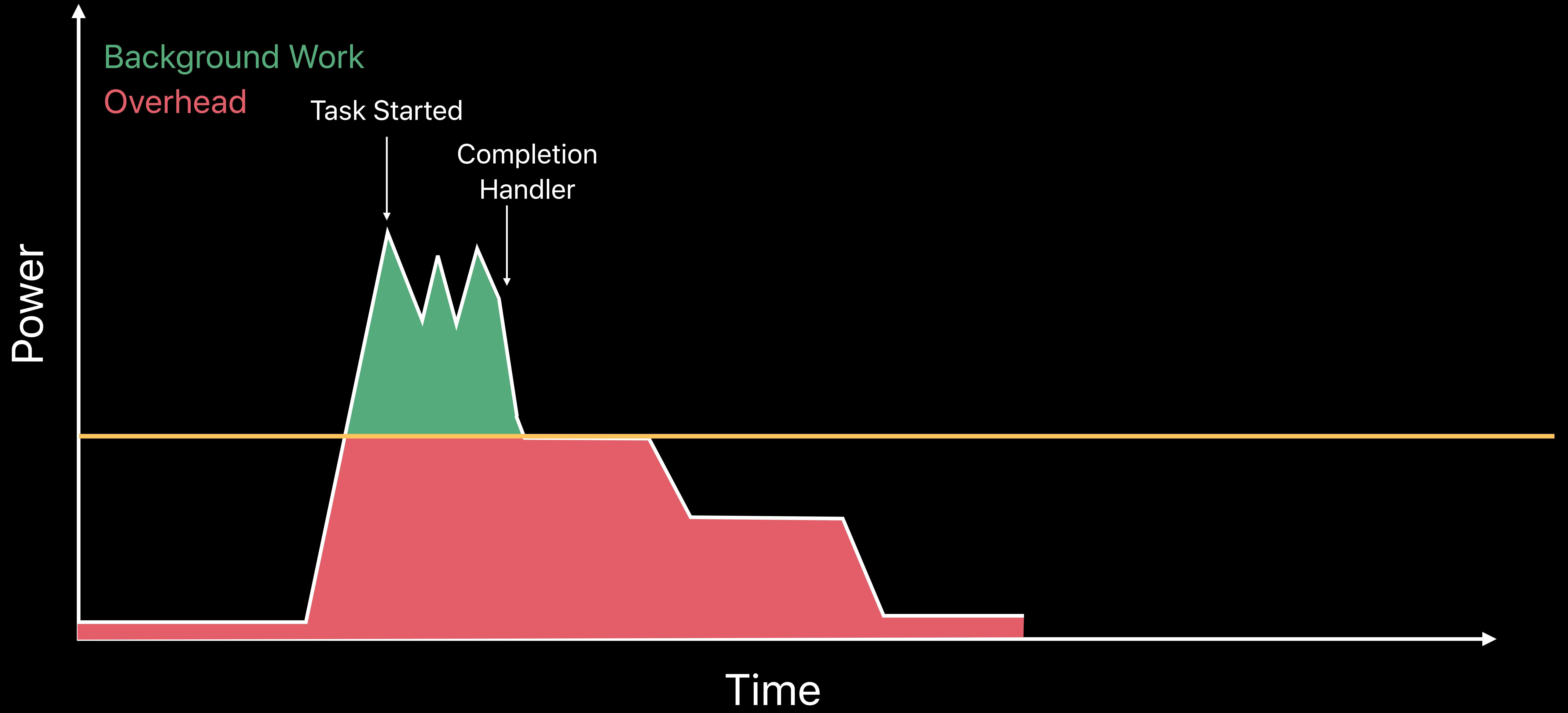


Lettings tasks expire



Calling completion handlers

- UIBackgroundTask
- VoIP
- Etc





Processing



Networking



Location



Graphics



Processing



Networking



Location



Graphics

<https://developer.apple.com/videos/play/wwdc2017/238/>

Review general battery life concepts

Review tools for energy debugging

Review new features for energy debugging

Tools for Energy Debugging

Tools for Energy Debugging



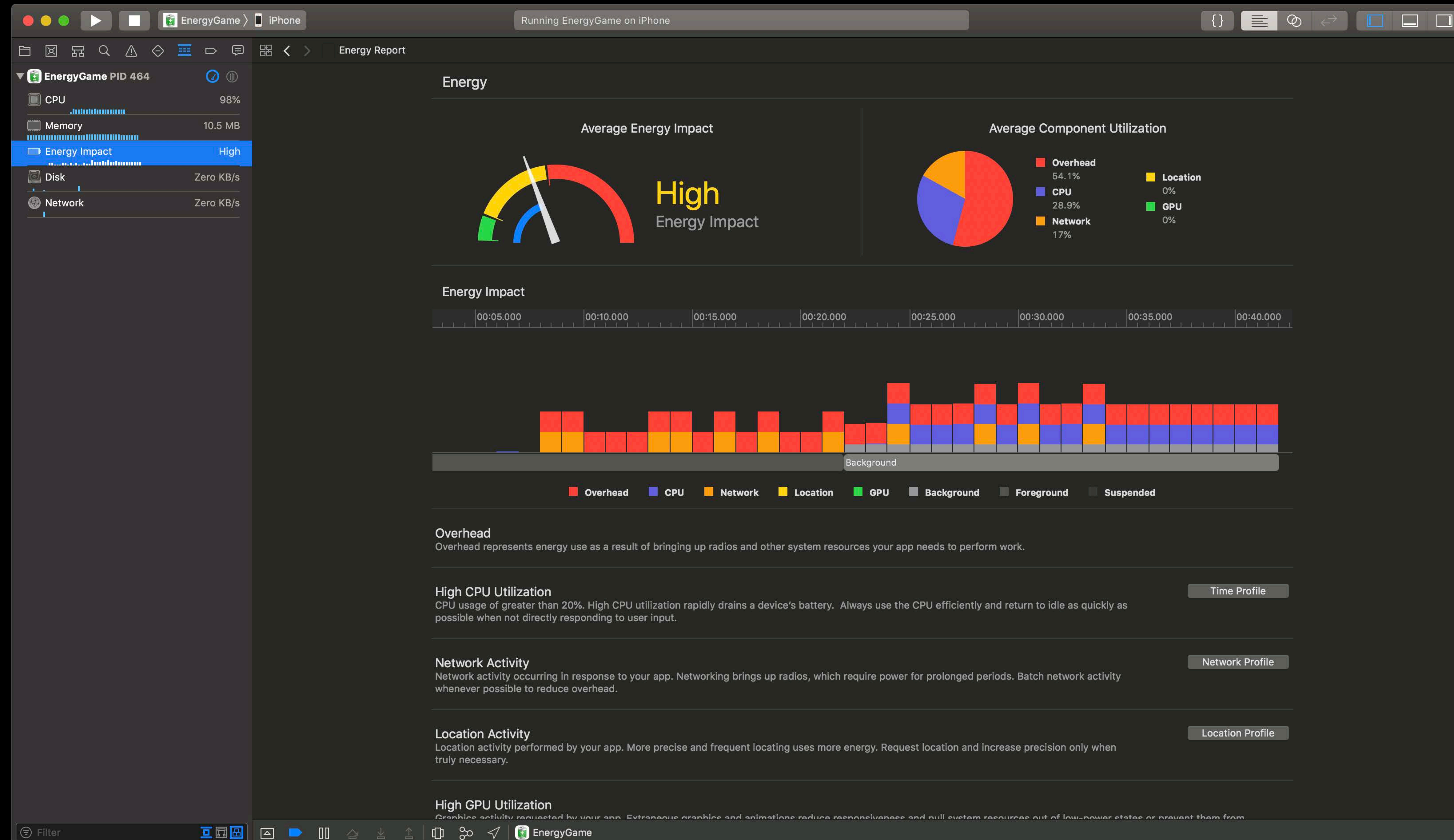
Energy Gauges

Tools for Energy Debugging



Energy Gauges

Energy Gauges



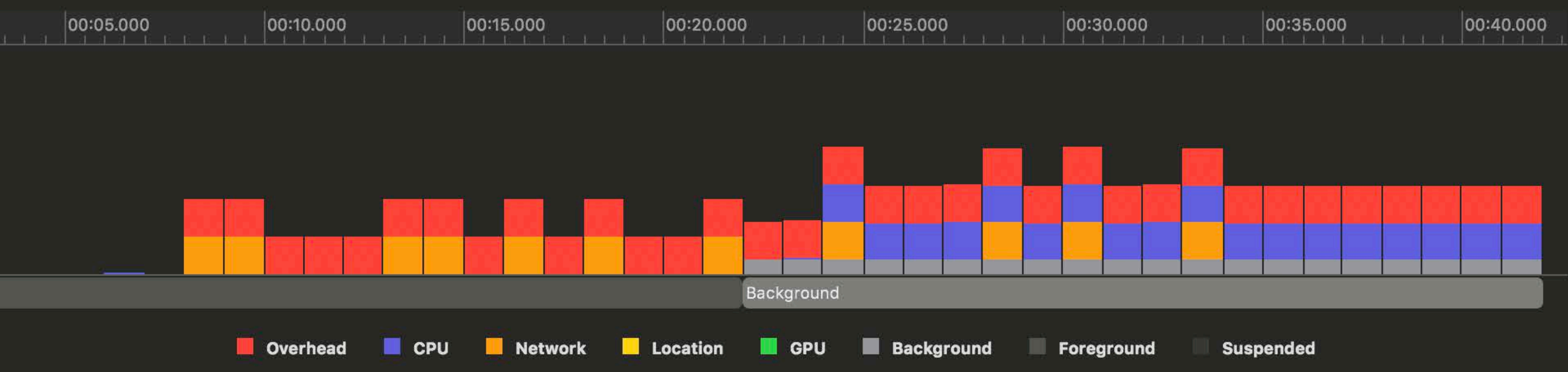
EnergyGame PID 464

- CPU 98%
- Memory 10.5 MB
- Energy Impact High**
- Disk Zero KB/s
- Network Zero KB/s

Energy



Energy Impact



Overhead

Overhead represents energy use as a result of bringing up radios and other system resources your app needs to perform work.

High CPU Utilization

CPU usage of greater than 20%. High CPU utilization rapidly drains a device's battery. Always use the CPU efficiently and return to idle as quickly as possible when not directly responding to user input.

Time Profile

Network Activity

Network activity occurring in response to your app. Networking brings up radios, which require power for prolonged periods. Batch network activity whenever possible to reduce overhead.

Network Profile

Location Activity

Location activity performed by your app. More precise and frequent locating uses more energy. Request location and increase precision only when truly necessary.

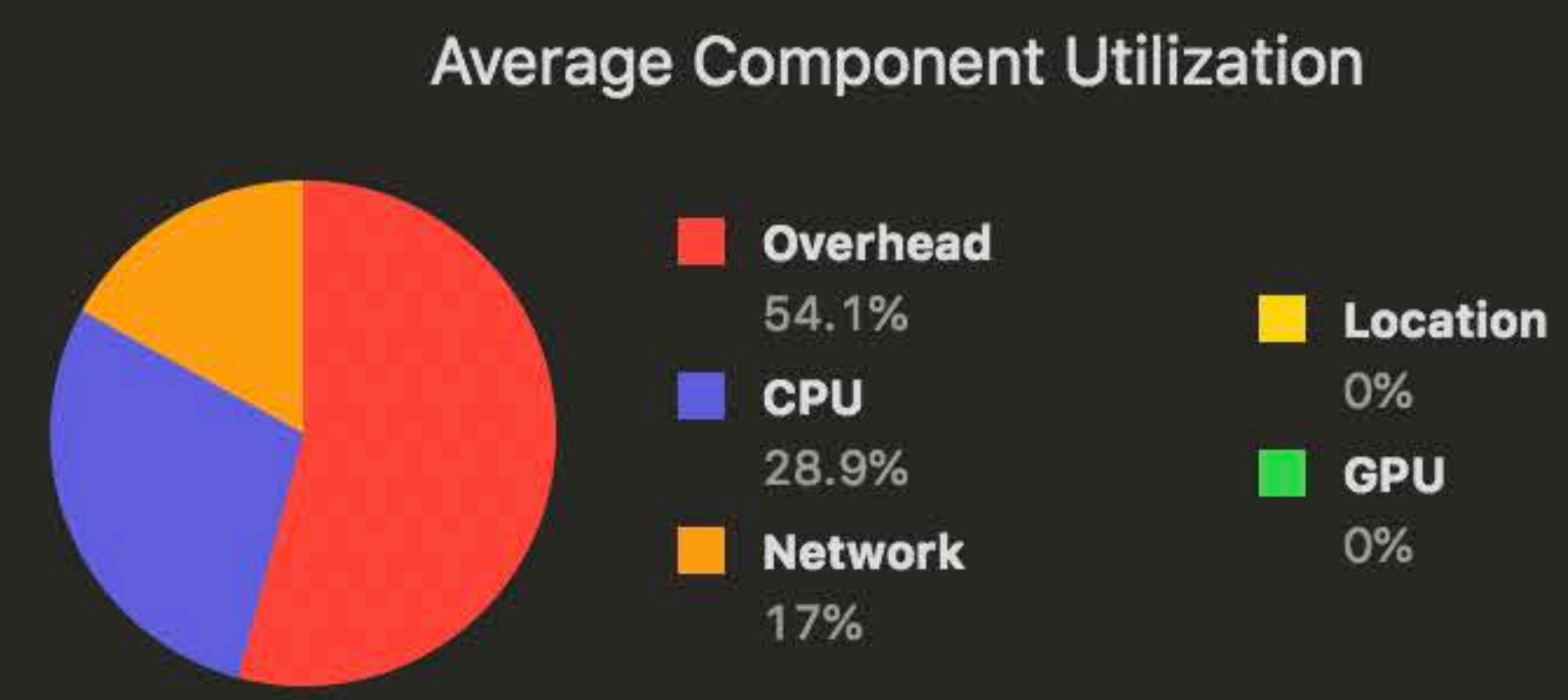
Location Profile

High GPU Utilization

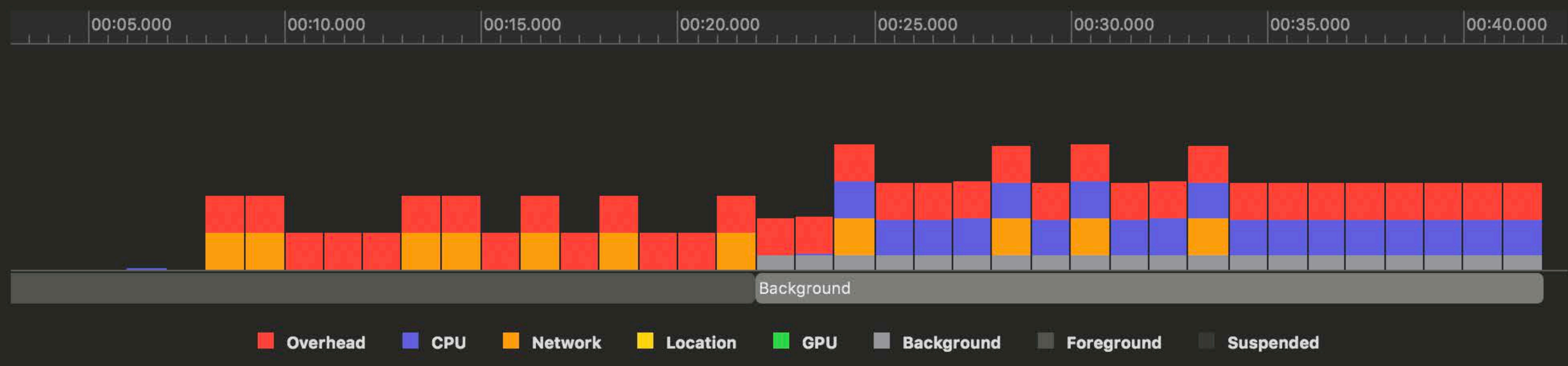
Graphics activity requested by your app. Extraneous graphics and animations reduce responsiveness and pull system resources out of low-power states or prevent them from

EnergyGame PID 464	98%
CPU	98%
Memory	10.5 MB
Energy Impact	High
Disk	Zero KB/s
Network	Zero KB/s

Energy



Energy Impact



Overhead

Overhead represents energy use as a result of bringing up radios and other system resources your app needs to perform work.

High CPU Utilization

CPU usage of greater than 20%. High CPU utilization rapidly drains a device's battery. Always use the CPU efficiently and return to idle as quickly as possible when not directly responding to user input.

Time Profile

Network Activity

Network activity occurring in response to your app. Networking brings up radios, which require power for prolonged periods. Batch network activity whenever possible to reduce overhead.

Network Profile

Location Activity

Location activity performed by your app. More precise and frequent locating uses more energy. Request location and increase precision only when truly necessary.

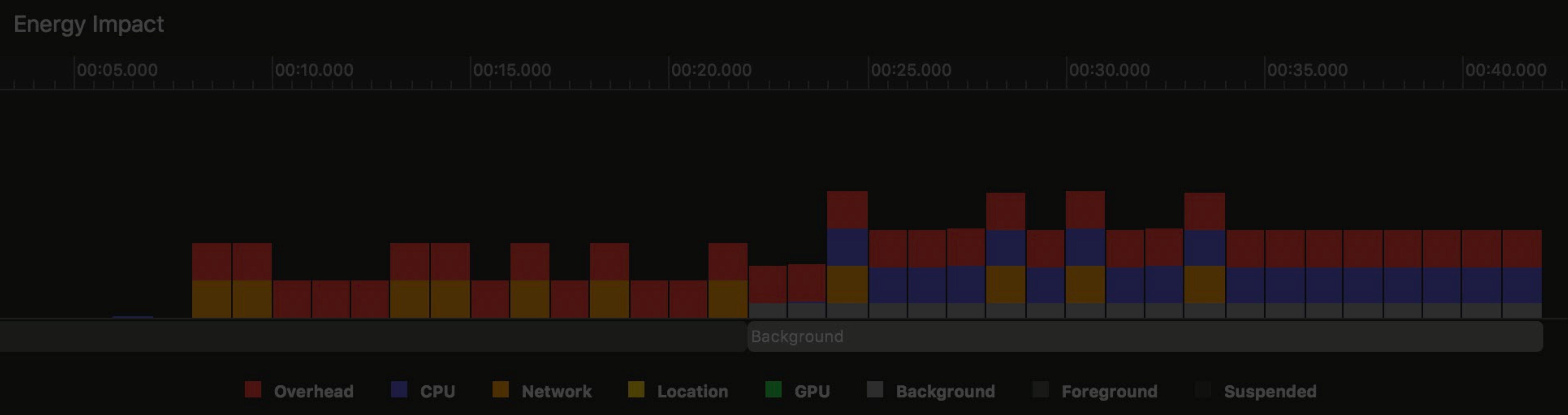
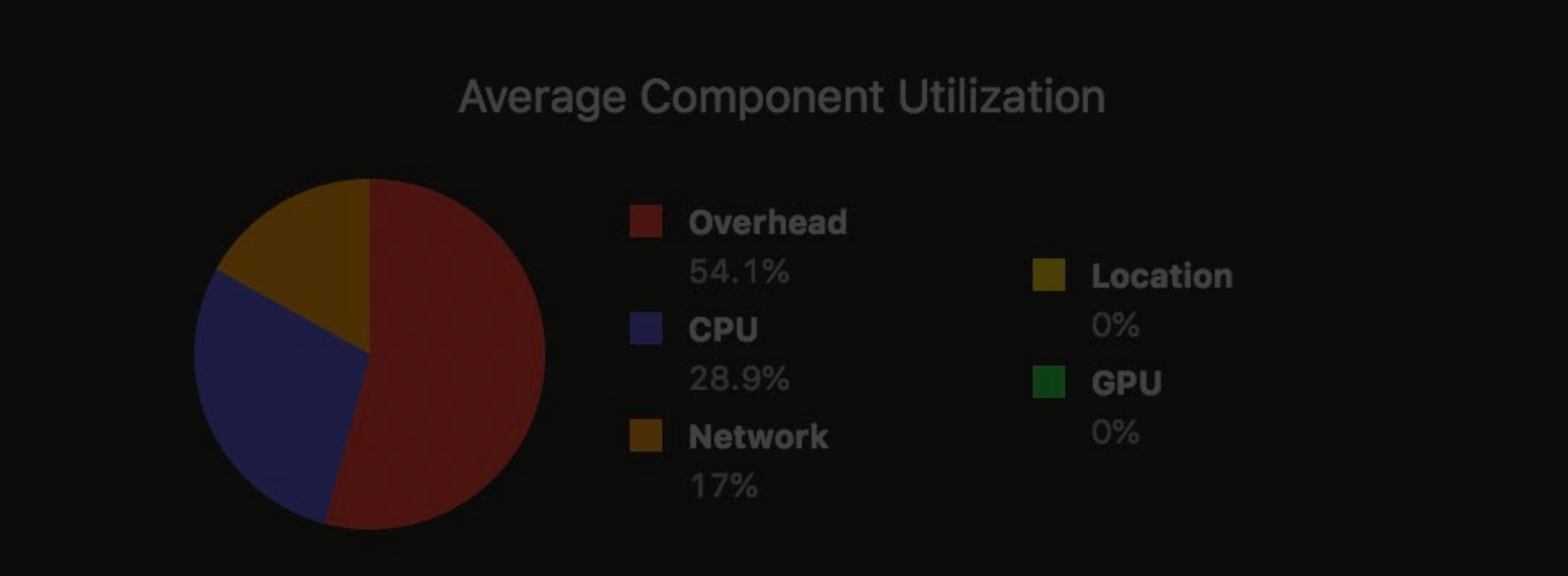
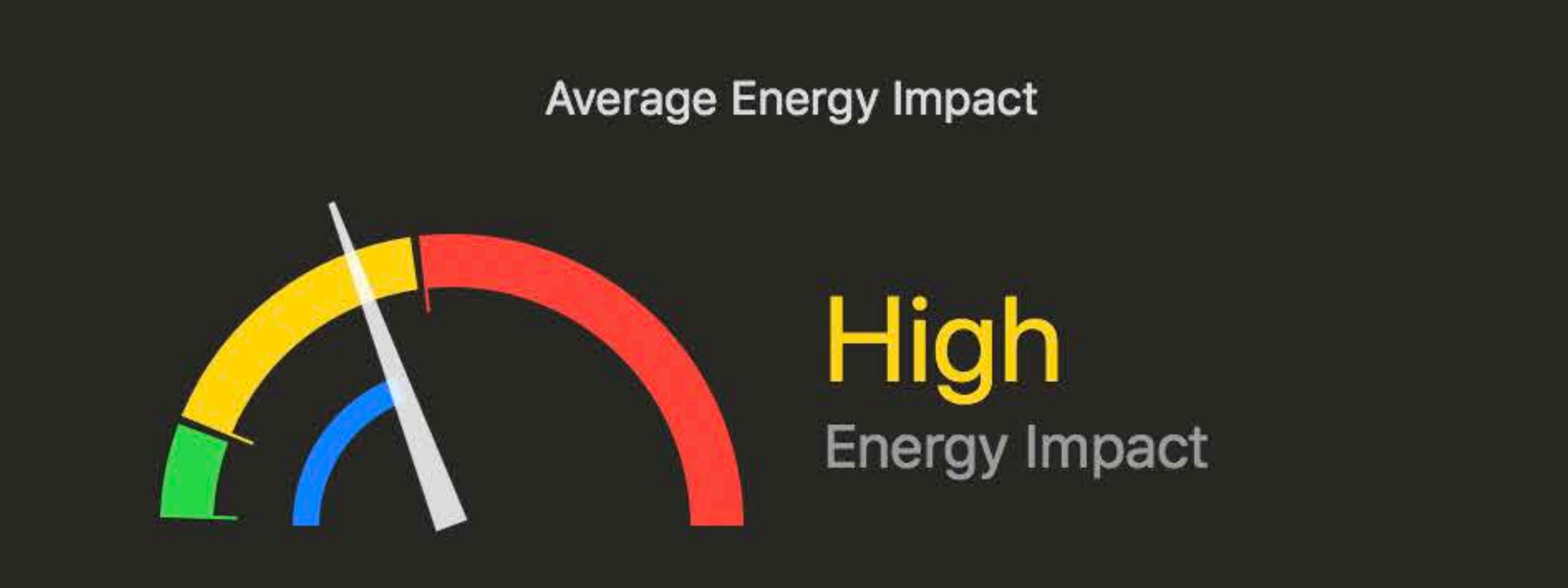
Location Profile

High GPU Utilization

Graphics activity requested by your app. Extraneous graphics and animations reduce responsiveness and pull system resources out of low-power states or prevent them from

EnergyGame PID 464	98%
CPU	98%
Memory	10.5 MB
Energy Impact	High
Disk	Zero KB/s
Network	Zero KB/s

Energy



Overhead
Overhead represents energy use as a result of bringing up radios and other system resources your app needs to perform work.

High CPU Utilization
CPU usage of greater than 20%. High CPU utilization rapidly drains a device's battery. Always use the CPU efficiently and return to idle as quickly as possible when not directly responding to user input.

[Time Profile](#)

Network Activity
Network activity occurring in response to your app. Networking brings up radios, which require power for prolonged periods. Batch network activity whenever possible to reduce overhead.

[Network Profile](#)

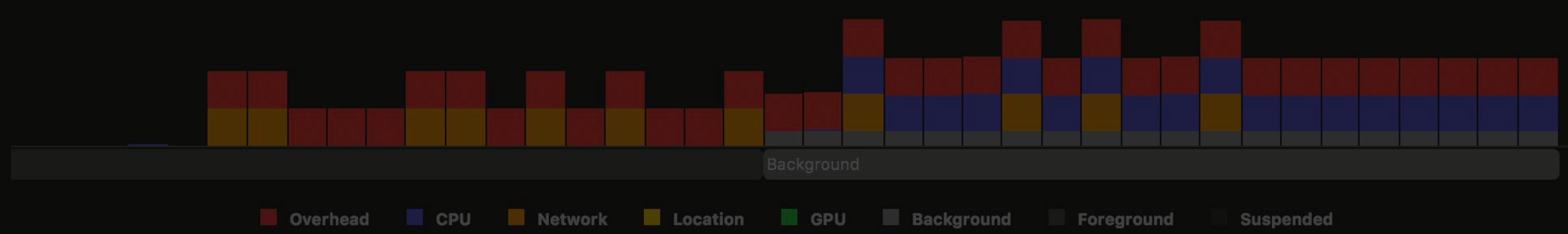
Location Activity
Location activity performed by your app. More precise and frequent locating uses more energy. Request location and increase precision only when truly necessary.

[Location Profile](#)

High GPU Utilization
Graphics activity requested by your app. Extraneous graphics and animations reduce responsiveness and pull system resources out of low-power states or prevent them from

EnergyGame PID 464

- CPU 98%
- Memory 10.5 MB
- Energy Impact High**
- Disk Zero KB/s
- Network Zero KB/s

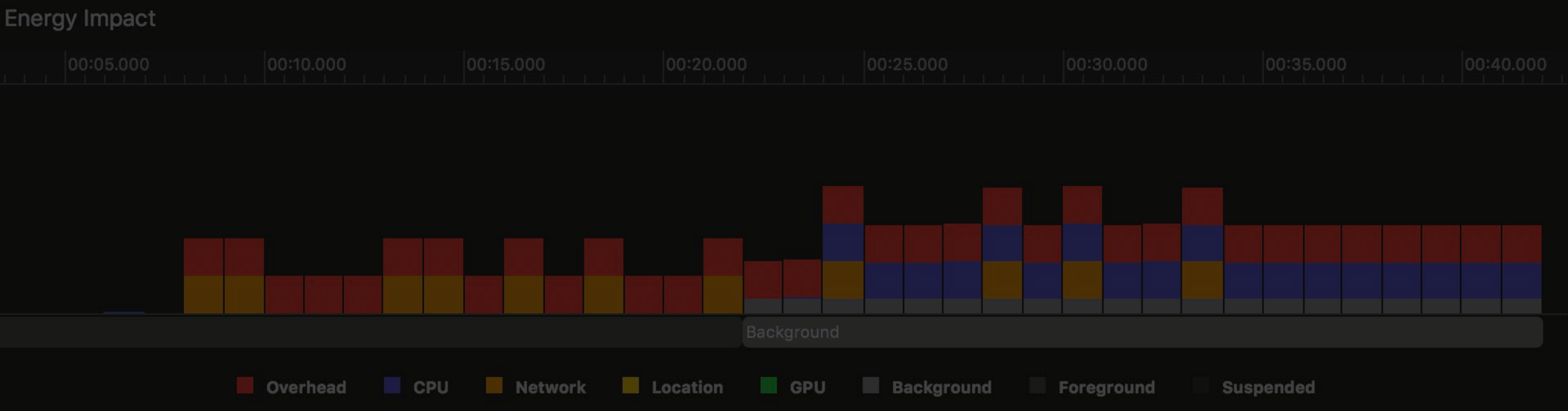
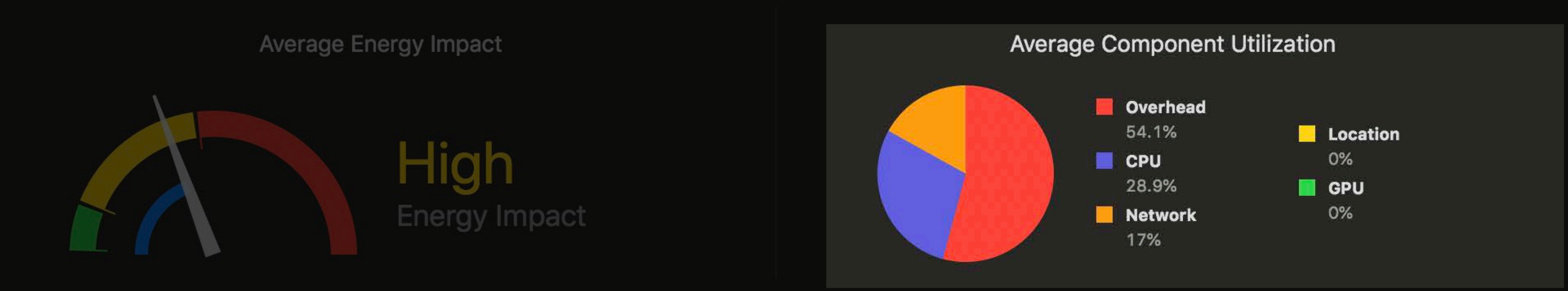


- Overhead**
Overhead represents energy use as a result of bringing up radios and other system resources your app needs to perform work.
- High CPU Utilization**
CPU usage of greater than 20%. High CPU utilization rapidly drains a device's battery. Always use the CPU efficiently and return to idle as quickly as possible when not directly responding to user input. [Time Profile](#)
- Network Activity**
Network activity occurring in response to your app. Networking brings up radios, which require power for prolonged periods. Batch network activity whenever possible to reduce overhead. [Network Profile](#)
- Location Activity**
Location activity performed by your app. More precise and frequent locating uses more energy. Request location and increase precision only when truly necessary. [Location Profile](#)
- High GPU Utilization**
Graphics activity requested by your app. Extraneous graphics and animations reduce responsiveness and pull system resources out of low-power states or prevent them from...

Energy Report

EnergyGame PID 464	98%
CPU	98%
Memory	10.5 MB
Energy Impact	High
Disk	Zero KB/s
Network	Zero KB/s

Energy



Overhead
Overhead represents energy use as a result of bringing up radios and other system resources your app needs to perform work.

High CPU Utilization
CPU usage of greater than 20%. High CPU utilization rapidly drains a device's battery. Always use the CPU efficiently and return to idle as quickly as possible when not directly responding to user input.

[Time Profile](#)

Network Activity
Network activity occurring in response to your app. Networking brings up radios, which require power for prolonged periods. Batch network activity whenever possible to reduce overhead.

[Network Profile](#)

Location Activity
Location activity performed by your app. More precise and frequent locating uses more energy. Request location and increase precision only when truly necessary.

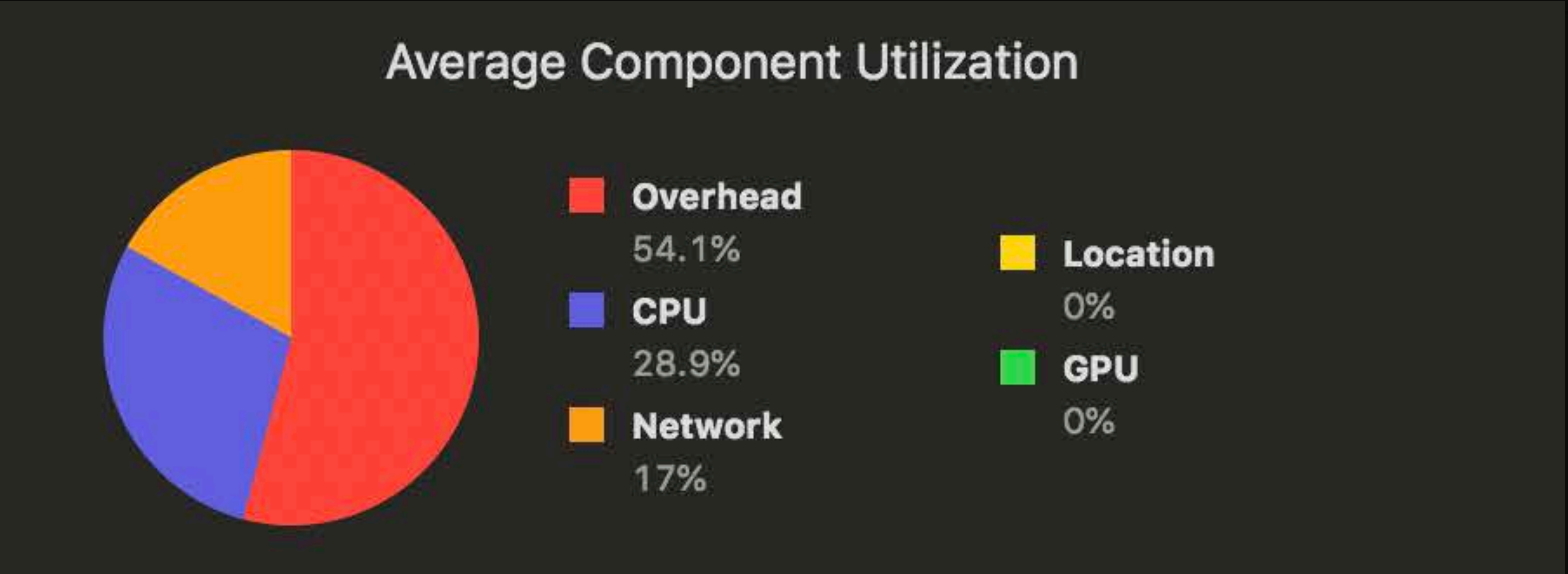
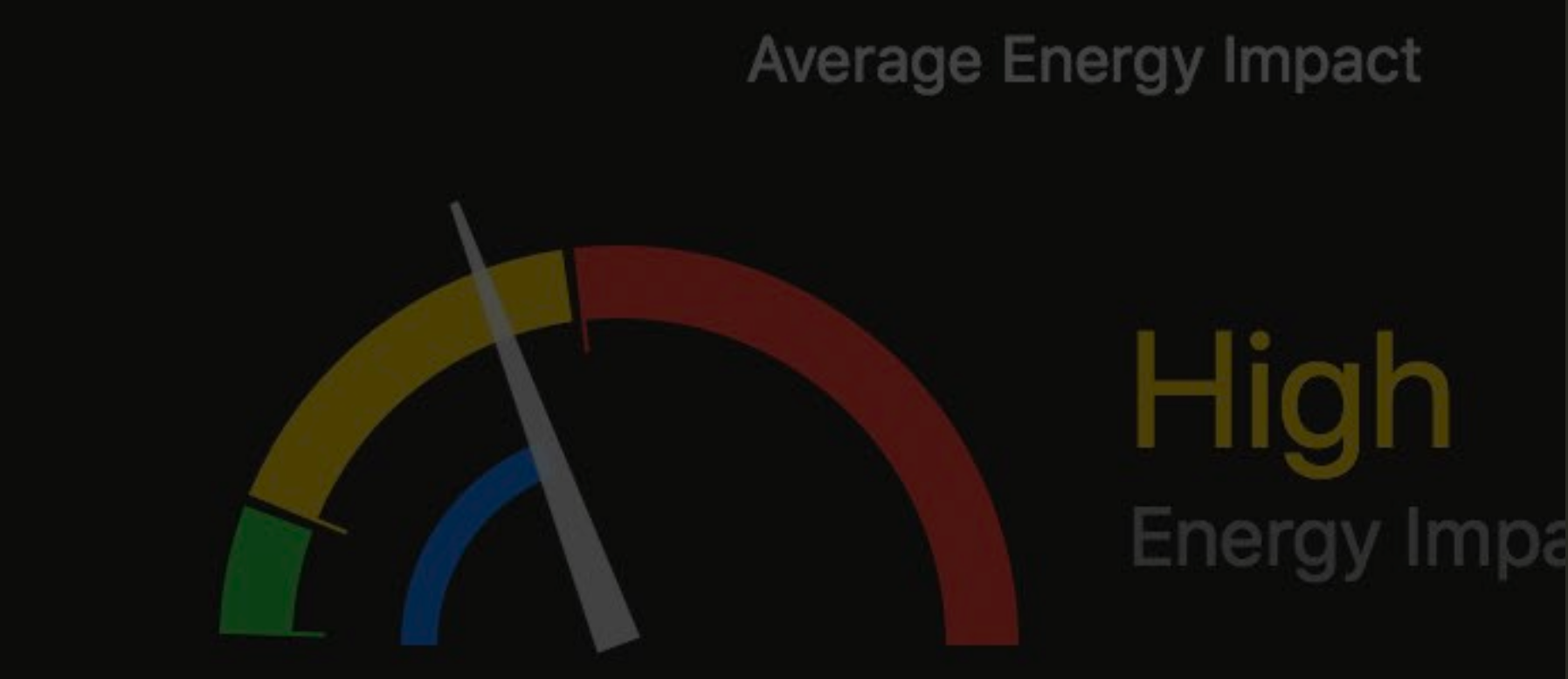
[Location Profile](#)

High GPU Utilization
Graphics activity requested by your app. Extraneous graphics and animations reduce responsiveness and pull system resources out of low-power states or prevent them from

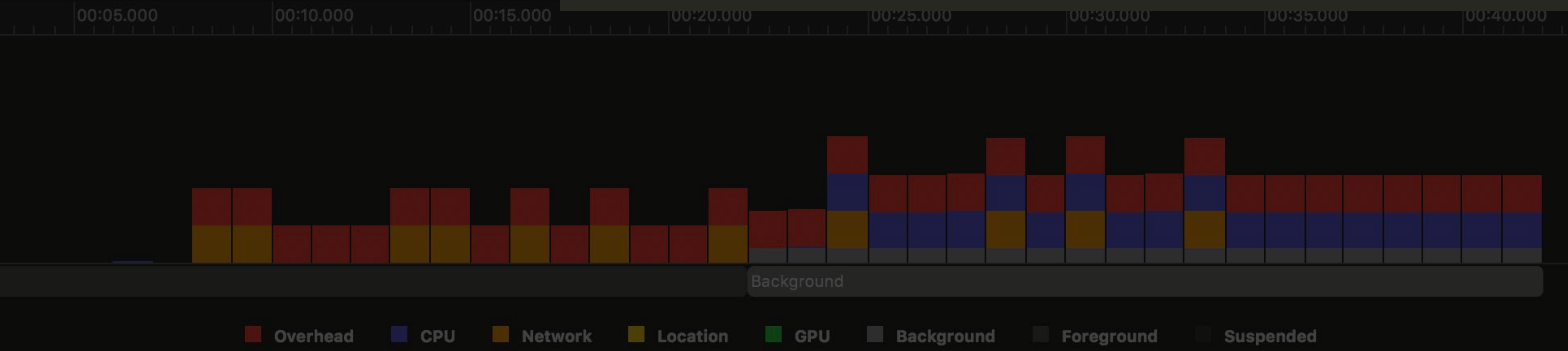
Energy Report

EnergyGame PID 464	98%
CPU	98%
Memory	10.5 MB
Energy Impact	High
Disk	Zero KB/s
Network	Zero KB/s

Energy



Energy Impact



Overhead

Overhead represents energy use as a result of bringing up radios and other system resources your app needs to perform work.

High CPU Utilization

CPU usage of greater than 20%. High CPU utilization rapidly drains a device's battery. Always use the CPU efficiently and return to idle as quickly as possible when not directly responding to user input.

Time Profile

Network Activity

Network activity occurring in response to your app. Networking brings up radios, which require power for prolonged periods. Batch network activity whenever possible to reduce overhead.

Network Profile

Location Activity

Location activity performed by your app. More precise and frequent locating uses more energy. Request location and increase precision only when truly necessary.

Location Profile

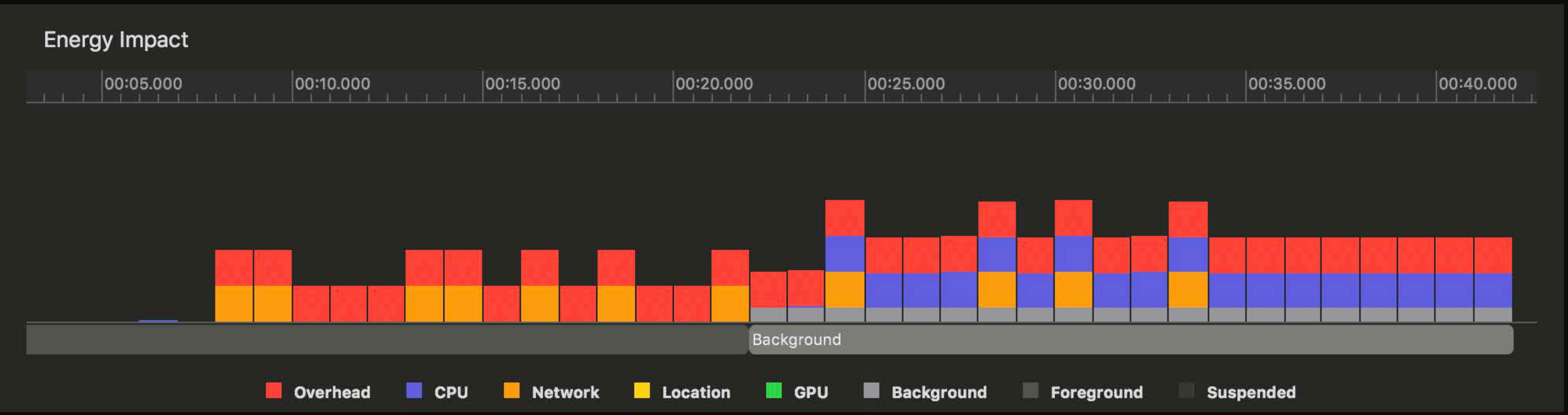
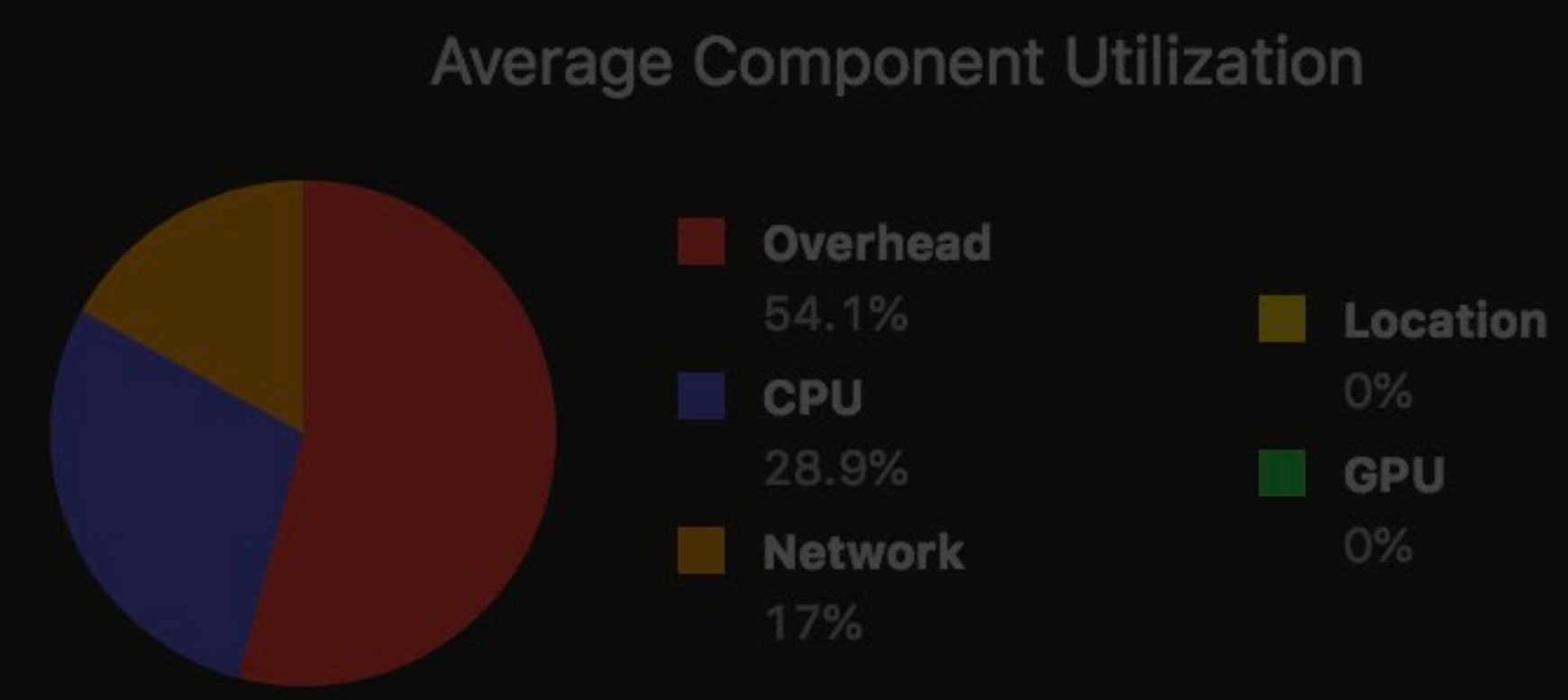
High GPU Utilization

Graphics activity requested by your app. Extraneous graphics and animations reduce responsiveness and pull system resources out of low-power states or prevent them from

Energy Report

EnergyGame PID 464	98%
CPU	98%
Memory	10.5 MB
Energy Impact	High
Disk	Zero KB/s
Network	Zero KB/s

Energy



Overhead

Overhead represents energy use as a result of bringing up radios and other system resources your app needs to perform work.

High CPU Utilization

CPU usage of greater than 20%. High CPU utilization rapidly drains a device's battery. Always use the CPU efficiently and return to idle as quickly as possible when not directly responding to user input.

Time Profile

Network Activity

Network activity occurring in response to your app. Networking brings up radios, which require power for prolonged periods. Batch network activity whenever possible to reduce overhead.

Network Profile

Location Activity

Location activity performed by your app. More precise and frequent locating uses more energy. Request location and increase precision only when truly necessary.

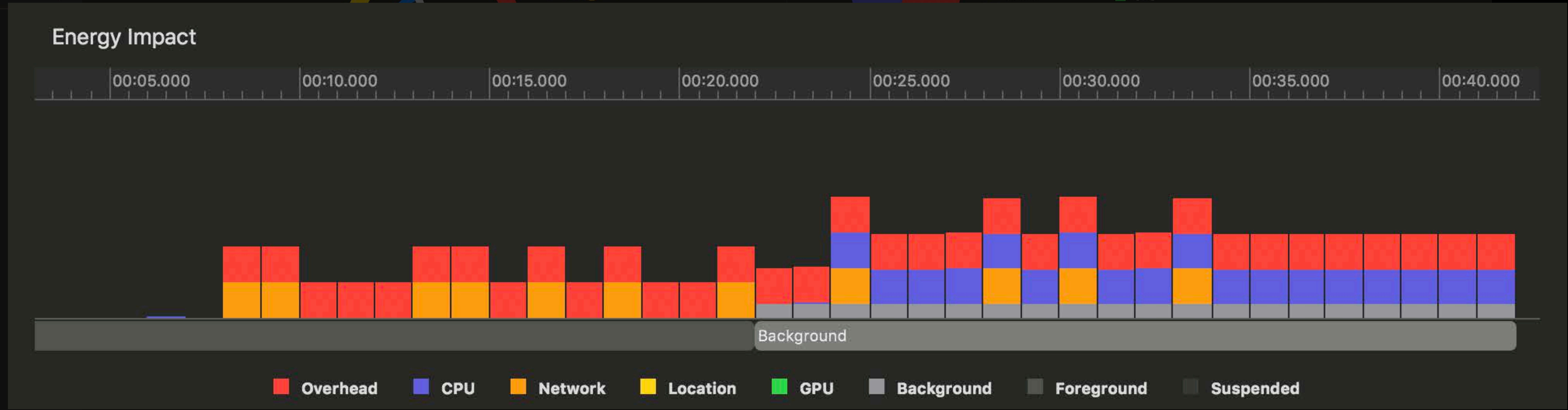
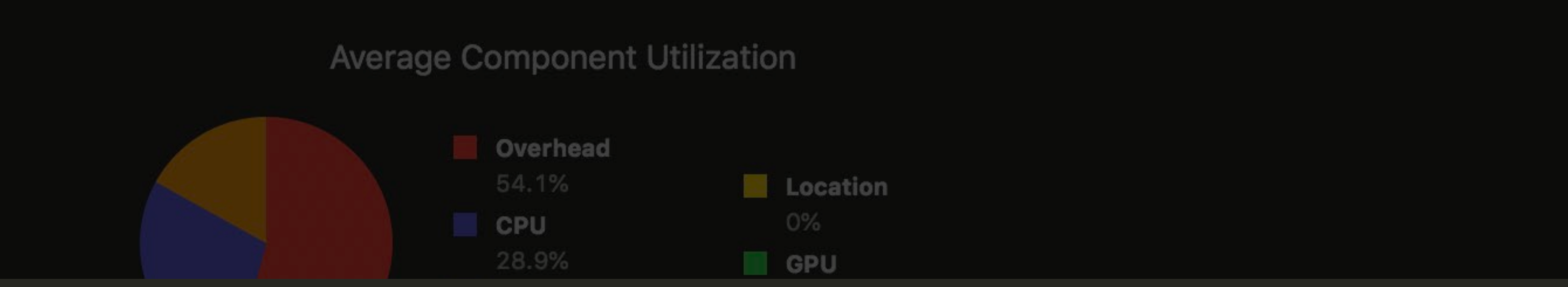
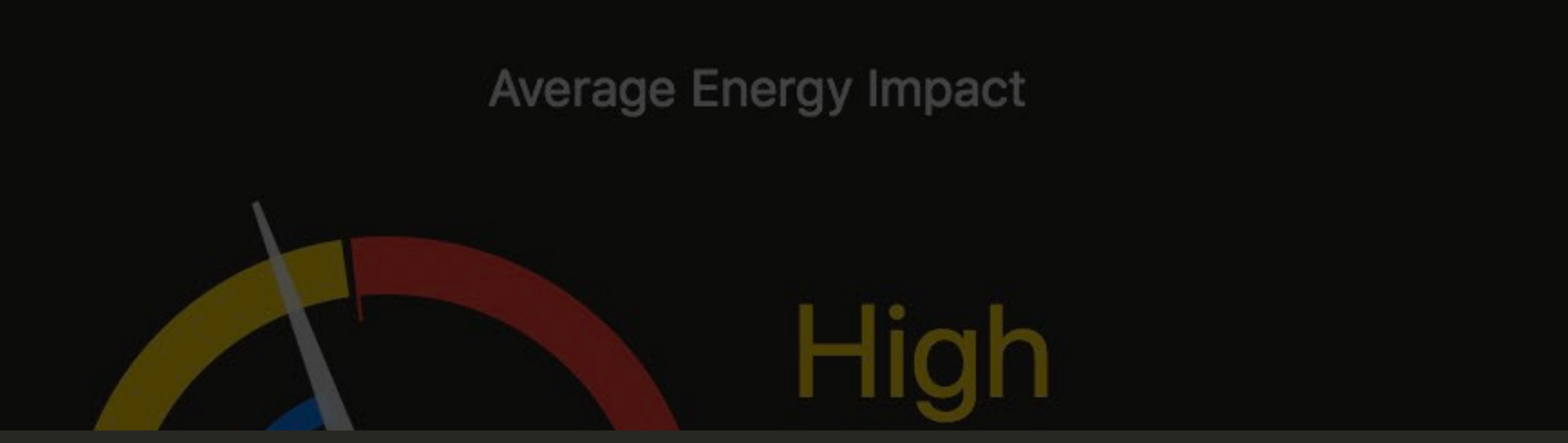
Location Profile

High GPU Utilization

Graphics activity requested by your app. Extraneous graphics and animations reduce responsiveness and pull system resources out of low-power states or prevent them from

EnergyGame PID 464	98%
CPU	98%
Memory	10.5 MB
Energy Impact	High
Disk	Zero KB/s
Network	Zero KB/s

Energy



- Overhead
- CPU
- Network
- Location
- GPU
- Background
- Foreground
- Suspended

- High CPU Utilization**
CPU usage of greater than 20%. High CPU utilization rapidly drains a device's battery. Always use the CPU efficiently and return to idle as quickly as possible when not directly responding to user input.
- Network Activity**
Network activity occurring in response to your app. Networking brings up radios, which require power for prolonged periods. Batch network activity whenever possible to reduce overhead.
- Location Activity**
Location activity performed by your app. More precise and frequent locating uses more energy. Request location and increase precision only when truly necessary.
- High GPU Utilization**
Graphics activity requested by your app. Extraneous graphics and animations reduce responsiveness and pull system resources out of low-power states or prevent them from

- Time Profile
- Network Profile
- Location Profile

Energy Gauges Are Great for...

Energy Gauges Are Great for...

High level energy characterization

Rapid profiling

Overhead

Overhead represents energy use as a result of bringing up radios and other system resources your app needs to perform work.

High CPU Utilization

CPU usage of greater than 20%. High CPU utilization rapidly drains a device's battery. Always use the CPU efficiently and return to idle as quickly as possible when not directly responding to user input.

Time Profile

Network Activity

Network activity occurring in response to your app. Networking brings up radios, which require power for prolonged periods. Batch network activity whenever possible to reduce overhead.

Network Profile

Location Activity

Location activity performed by your app. More precise and frequent locating uses more energy. Request location and increase precision only when truly necessary.

Location Profile

High GPU Utilization

Graphics activity requested by your app. Extraneous graphics and animations reduce responsiveness and pull system resources out of low-power states or prevent them from powering down all together, resulting in significant energy use. Make updates to visible content only, reduce the use of opacity, and prefer lower, consistent frame rates when performing animations.

Background State

Your app is in a background state, keeping the system awake. Even an idle background app uses energy. If your app requires background operations, use deferral APIs that let the system schedule the work efficiently and wake to run your app only when necessary. Otherwise, reduce activity immediately when placed in the background and notify the system once the activity is complete.

Foreground State

Your app is in the foreground. Use recommended APIs, batch and reduce network operations, and avoid unnecessary updates to the user interface. Strive to make your app absolutely idle when it's not responding to user input.

Overhead

Overhead represents energy use as a result of bringing up radios and other system resources your app needs to perform work.

High CPU Utilization

CPU usage of greater than 20%. High CPU utilization rapidly drains a device's battery. Always use the CPU efficiently and return to idle as quickly as possible when not directly responding to user input.

Network Activity

Network activity occurring in response to your app. Networking brings up radios, which require power for prolonged periods. Batch network activity whenever possible to reduce overhead.

Location Activity

Location activity performed by your app. More precise and frequent locating uses more energy. Request location and increase precision only when truly necessary.

High GPU Utilization

Graphics activity requested by your app. Extraneous graphics and animations reduce responsiveness and pull system resources out of low-power states or prevent them from powering down all together, resulting in significant energy use. Make updates to visible content only, reduce the use of opacity, and prefer lower, consistent frame rates when performing animations.

Background State

Your app is in a background state, keeping the system awake. Even an idle background app uses energy. If your app requires background operations, use deferral APIs that let the system schedule the work efficiently and wake to run your app only when necessary. Otherwise, reduce activity immediately when placed in the background and notify the system once the activity is complete.

Foreground State

Your app is in the foreground. Use recommended APIs, batch and reduce network operations, and avoid unnecessary updates to the user interface. Strive to make your app absolutely idle when it's not responding to user input.

Time Profile

Network Profile

Location Profile

Overhead

Overhead represents energy use as a result of bringing up radios and other system resources your app needs to perform work.

High CPU Utilization

CPU usage of greater than 20%. High CPU utilization rapidly drains a device's battery. Always use the CPU efficiently and return to idle as quickly as possible when not directly responding to user input.

Network Activity

Network activity occurring in response to your app. Networking brings up radios, which require power for prolonged periods. Batch network activity whenever possible to reduce overhead.

Location Activity

Location activity performed by your app. More precise and frequent locating uses more energy. Request location and increase precision only when truly necessary.

High GPU Utilization

Graphics activity requested by your app. Extraneous graphics and animations reduce responsiveness and pull system resources out of low-power states or prevent them from powering down all together, resulting in significant energy use. Make updates to visible content only, reduce the use of opacity, and prefer lower, consistent frame rates when performing animations.

Background State

Your app is in a background state, keeping the system awake. Even an idle background app uses energy. If your app requires background operations, use deferral APIs that let the system schedule the work efficiently and wake to run your app only when necessary. Otherwise, reduce activity immediately when placed in the background and notify the system once the activity is complete.

Foreground State

Your app is in the foreground. Use recommended APIs, batch and reduce network operations, and avoid unnecessary updates to the user interface. Strive to make your app absolutely idle when it's not responding to user input.

Time Profile

Network Profile

Location Profile

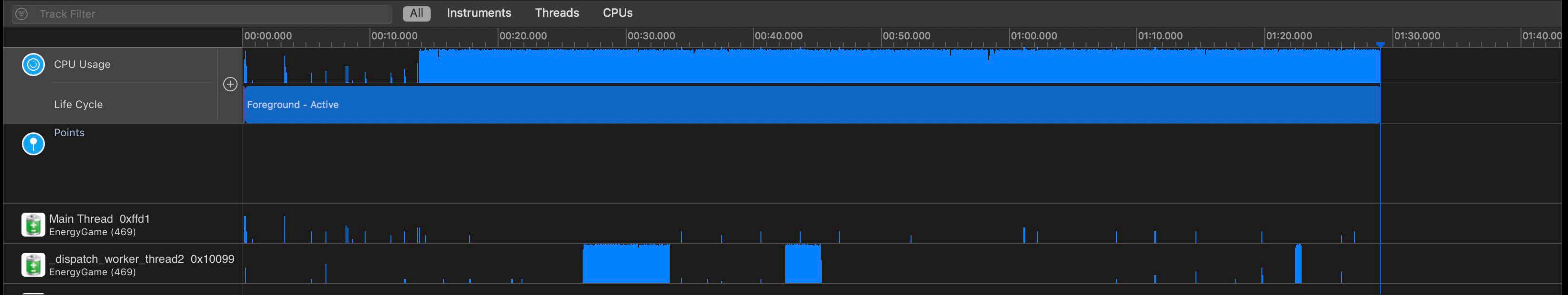
Time Profiler

The screenshot displays the Xcode Instruments interface for a Time Profiler session. The top section shows the application 'EnergyGame' running on an iPhone (12.0). The main area features a timeline with various instrument tracks: CPU Usage (showing high activity), Life Cycle (Foreground - Active), Points, Main Thread (0xffd1), and a dispatch worker thread (0x10099). The bottom section shows the Time Profiler profile with a call tree. The selected item is a closure in static ViewController.heavyTimer(). The Heaviest Stack Trace on the right shows the call path from EnergyGame through dispatch and system libraries to cryptographic functions.

Weight	Self Weight	Symbol Name
1.24 min	100.0%	EnergyGame (469)
25.70 s	34.4%	_dispatch_workloop_worker_thread 0x1009b
25.70 s	34.4%	_pthread_wqthread libsystem_pthread.dylib
25.66 s	34.3%	_dispatch_worker_thread2 libdispatch.dylib
25.66 s	34.3%	_dispatch_root_queue_drain libdispatch.dylib
25.66 s	34.3%	_dispatch_source_invoke\$VARIANT\$armv81 libdispatch.dylib
25.65 s	34.3%	_dispatch_continuation_pop\$VARIANT\$armv81 libdispatch.dylib
25.65 s	34.3%	_dispatch_client_callout libdispatch.dylib
25.65 s	34.3%	thunk for @escaping @callee_guaranteed () -> () EnergyGame
25.65 s	34.3%	closure #1 in static ViewController.heavyTimer() EnergyGame
19.46 s	26.0%	503.00 ms arc4random libsystem_c.dylib
6.86 s	9.1%	6.86 s os_unfair_lock_lock\$VARIANT\$armv81 libsystem_platform.dylib
5.67 s	7.5%	24.00 ms generate libcorecrypto.dylib
2.61 s	3.4%	5.00 ms drbg_update libcorecrypto.dylib
803.00 ms	1.0%	508.00 ms drbg_update libcorecrypto.dylib
281.00 ms	0.3%	281.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib
14.00 ms	0.0%	14.00 ms memset_s libsystem_c.dylib
799.00 ms	1.0%	0 s ccaes_arm_encrypt_key libcorecrypto.dylib
785.00 ms	1.0%	785.00 ms aes_key_expansion libcorecrypto.dylib
11.00 ms	0.0%	11.00 ms ccaes_arm_encrypt_key libcorecrypto.dylib
3.00 ms	0.0%	3.00 ms ccaes_arm_encrypt_key128 libcorecrypto.dylib
347.00 ms	0.4%	40.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib
198.00 ms	0.2%	198.00 ms ccaes_arm_encrypt_ecb libcorecrypto.dylib
80.00 ms	0.1%	80.00 ms Decrypt_Main_Loop_End libcorecrypto.dylib
26.00 ms	0.0%	26.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib

Heaviest Stack Trace

- 74693.0 EnergyGame (469)
- 25695.0 _dispatch_workloop_worker_th
- 25695.0 _pthread_wqthread
- 25659.0 _dispatch_worker_thread2
- 25659.0 _dispatch_root_queue_drain
- 25656.0 _dispatch_source_invoke\$VAR
- 25652.0 _dispatch_continuation_pop\$
- 25652.0 _dispatch_client_callout
- 25652.0 thunk for @escaping @callee_
- 25649.0 closure #1 in static ViewConti
- 19459.0 arc4random
- 6860.0 os_unfair_lock_lock\$VARIANT\$



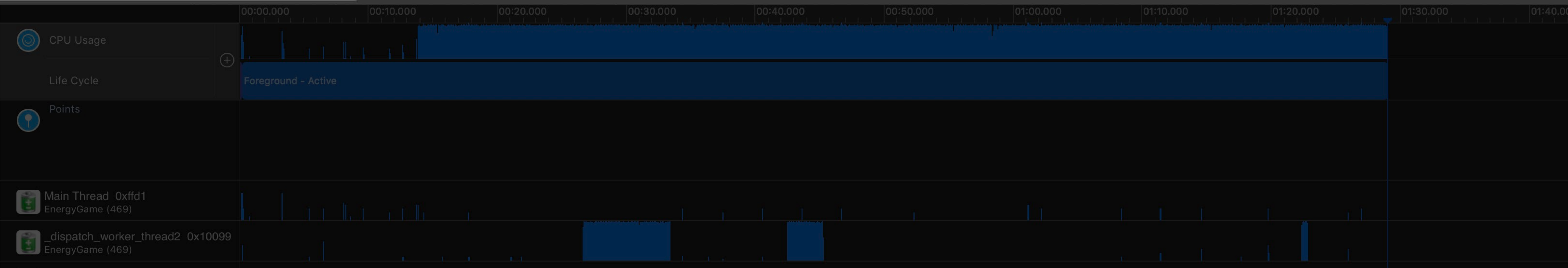
Time Profiler > Profile > Root

Weight	Self Weight	Symbol Name
1.24 min 100.0%	0 s	EnergyGame (469)
25.70 s 34.4%	0 s	_dispatch_workloop_worker_thread 0x1009b
25.70 s 34.4%	0 s	_thread_wqthread libsystem_pthread.dylib
25.66 s 34.3%	0 s	_dispatch_worker_thread2 libdispatch.dylib
25.66 s 34.3%	0 s	_dispatch_root_queue_drain libdispatch.dylib
25.66 s 34.3%	0 s	_dispatch_source_invoke\$VARIANT\$armv81 libdispatch.dylib
25.65 s 34.3%	0 s	_dispatch_continuation_pop\$VARIANT\$armv81 libdispatch.dylib
25.65 s 34.3%	0 s	_dispatch_client_callout libdispatch.dylib
25.65 s 34.3%	0 s	think for @escaping @callee_guaranteed () -> () EnergyGame
25.65 s 34.3%	0 s	closure #1 in static ViewController.heavyTimer() EnergyGame
19.46 s 26.0%	503.00 ms	arc4random libsystem_c.dylib
6.86 s 9.1%	6.86 s	os_unfair_lock_lock\$VARIANT\$armv81 libsystem_platform.dylib
5.67 s 7.5%	24.00 ms	generate libcorecrypto.dylib
2.61 s 3.4%	5.00 ms	drbg_update libcorecrypto.dylib
803.00 ms 1.0%	508.00 ms	drbg_update libcorecrypto.dylib
281.00 ms 0.3%	281.00 ms	ccaes_vng_ctr_crypt libcorecrypto.dylib
14.00 ms 0.0%	14.00 ms	memset_s libsystem_c.dylib
799.00 ms 1.0%	0 s	ccaes_arm_encrypt_key libcorecrypto.dylib
785.00 ms 1.0%	785.00 ms	aes_key_expansion libcorecrypto.dylib
11.00 ms 0.0%	11.00 ms	ccaes_arm_encrypt_key libcorecrypto.dylib
3.00 ms 0.0%	3.00 ms	ccaes_arm_encrypt_key128 libcorecrypto.dylib
347.00 ms 0.4%	40.00 ms	ccaes_vng_ctr_crypt libcorecrypto.dylib
198.00 ms 0.2%	198.00 ms	ccaes_arm_encrypt_ecb libcorecrypto.dylib
80.00 ms 0.1%	80.00 ms	Decrypt_Main_Loop_End libcorecrypto.dylib
26.00 ms 0.0%	26.00 ms	ccaes_vng_ctr_crypt libcorecrypto.dylib

Heaviest Stack Trace

74693.0	EnergyGame (469)
25695.0	_dispatch_workloop_worker_th
25695.0	_pthread_wqthread
25659.0	_dispatch_worker_thread2
25659.0	_dispatch_root_queue_drain
25656.0	_dispatch_source_invoke\$VAR
25652.0	_dispatch_continuation_pop\$
25652.0	_dispatch_client_callout
25652.0	think for @escaping @callee_
25649.0	closure #1 in static ViewConti
19459.0	arc4random
6860.0	os_unfair_lock_lock\$VARIANT\$

Track Filter: All Instruments Threads CPUs

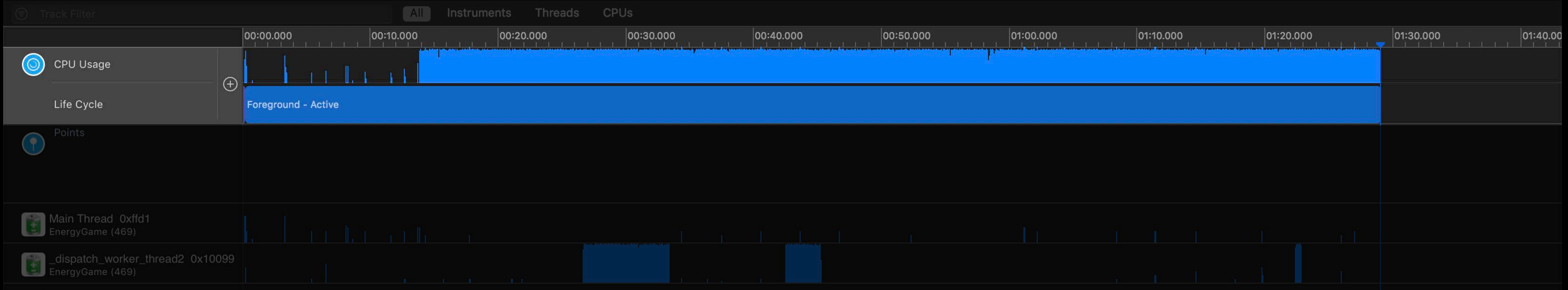


Time Profiler > Profile > Root

Weight	Self Weight	Symbol Name
1.24 min	100.0%	0 s EnergyGame (469)
25.70 s	34.4%	0 s _dispatch_workloop_worker_thread 0x1009b
25.70 s	34.4%	0 s _pthread_wqthread libsystem_pthread.dylib
25.66 s	34.3%	0 s _dispatch_worker_thread2 libdispatch.dylib
25.66 s	34.3%	0 s _dispatch_root_queue_drain libdispatch.dylib
25.66 s	34.3%	0 s _dispatch_source_invoke\$VARIANT\$armv81 libdispatch.dylib
25.65 s	34.3%	0 s _dispatch_continuation_pop\$VARIANT\$armv81 libdispatch.dylib
25.65 s	34.3%	0 s _dispatch_client_callout libdispatch.dylib
25.65 s	34.3%	0 s thunk for @escaping @callee_guaranteed () -> () EnergyGame
25.65 s	34.3%	0 s closure #1 in static ViewController.heavyTimer() EnergyGame
19.46 s	26.0%	503.00 ms arc4random libsystem_c.dylib
6.86 s	9.1%	6.86 s os_unfair_lock_lock\$VARIANT\$armv81 libsystem_platform.dylib
5.67 s	7.5%	24.00 ms generate libcorecrypto.dylib
2.61 s	3.4%	5.00 ms drbg_update libcorecrypto.dylib
803.00 ms	1.0%	508.00 ms drbg_update libcorecrypto.dylib
281.00 ms	0.3%	281.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib
14.00 ms	0.0%	14.00 ms memset_s libsystem_c.dylib
799.00 ms	1.0%	0 s ccaes_arm_encrypt_key libcorecrypto.dylib
785.00 ms	1.0%	785.00 ms aes_key_expansion libcorecrypto.dylib
11.00 ms	0.0%	11.00 ms ccaes_arm_encrypt_key libcorecrypto.dylib
3.00 ms	0.0%	3.00 ms ccaes_arm_encrypt_key128 libcorecrypto.dylib
347.00 ms	0.4%	40.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib
198.00 ms	0.2%	198.00 ms ccaes_arm_encrypt_ecb libcorecrypto.dylib
80.00 ms	0.1%	80.00 ms Decrypt_Main_Loop_End libcorecrypto.dylib
26.00 ms	0.0%	26.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib

Heaviest Stack Trace

74693.0	EnergyGame (469)
25695.0	_dispatch_workloop_worker_th
25695.0	_pthread_wqthread
25659.0	_dispatch_worker_thread2
25659.0	_dispatch_root_queue_drain
25656.0	_dispatch_source_invoke\$VAR
25652.0	_dispatch_continuation_pop\$
25652.0	_dispatch_client_callout
25652.0	thunk for @escaping @callee_
25649.0	closure #1 in static ViewConti
19459.0	arc4random
6860.0	os_unfair_lock_lock\$VARIANT\$



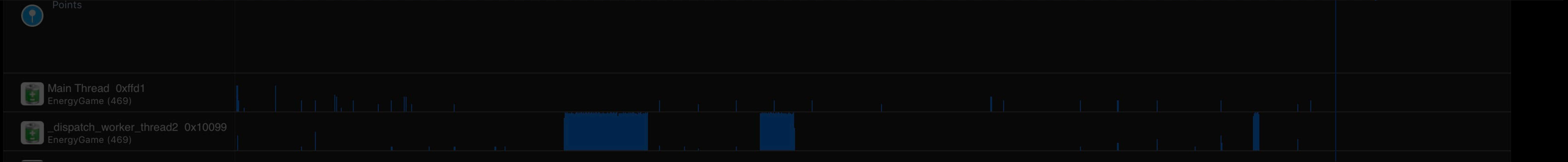
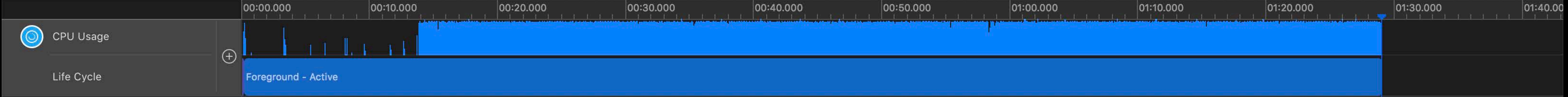
Time Profiler Profile Root

Weight	Self Weight	Symbol Name
1.24 min	100.0%	0 s EnergyGame (469)
25.70 s	34.4%	0 s _dispatch_workloop_worker_thread 0x1009b
25.70 s	34.4%	0 s _pthread_wqthread libsystem_pthread.dylib
25.66 s	34.3%	0 s _dispatch_worker_thread2 libdispatch.dylib
25.66 s	34.3%	0 s _dispatch_root_queue_drain libdispatch.dylib
25.66 s	34.3%	0 s _dispatch_source_invoke\$VARIANT\$armv81 libdispatch.dylib
25.65 s	34.3%	0 s _dispatch_continuation_pop\$VARIANT\$armv81 libdispatch.dylib
25.65 s	34.3%	0 s _dispatch_client_callout libdispatch.dylib
25.65 s	34.3%	0 s thunk for @escaping @callee_guaranteed () -> () EnergyGame
25.65 s	34.3%	0 s closure #1 in static ViewController.heavyTimer() EnergyGame
19.46 s	26.0%	503.00 ms arc4random libsystem_c.dylib
6.86 s	9.1%	6.86 s os_unfair_lock_lock\$VARIANT\$armv81 libsystem_platform.dylib
5.67 s	7.5%	24.00 ms generate libcorecrypto.dylib
2.61 s	3.4%	5.00 ms drbg_update libcorecrypto.dylib
803.00 ms	1.0%	508.00 ms drbg_update libcorecrypto.dylib
281.00 ms	0.3%	281.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib
14.00 ms	0.0%	14.00 ms memset_s libsystem_c.dylib
799.00 ms	1.0%	0 s ccaes_arm_encrypt_key libcorecrypto.dylib
785.00 ms	1.0%	785.00 ms aes_key_expansion libcorecrypto.dylib
11.00 ms	0.0%	11.00 ms ccaes_arm_encrypt_key libcorecrypto.dylib
3.00 ms	0.0%	3.00 ms ccaes_arm_encrypt_key128 libcorecrypto.dylib
347.00 ms	0.4%	40.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib
198.00 ms	0.2%	198.00 ms ccaes_arm_encrypt_ecb libcorecrypto.dylib
80.00 ms	0.1%	80.00 ms Decrypt_Main_Loop_End libcorecrypto.dylib
26.00 ms	0.0%	26.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib

Heaviest Stack Trace

74693.0	EnergyGame (469)
25695.0	_dispatch_workloop_worker_th
25695.0	_pthread_wqthread
25659.0	_dispatch_worker_thread2
25659.0	_dispatch_root_queue_drain
25656.0	_dispatch_source_invoke\$VAR
25652.0	_dispatch_continuation_pop\$
25652.0	_dispatch_client_callout
25652.0	thunk for @escaping @callee_
25649.0	closure #1 in static ViewConti
19459.0	arc4random
6860.0	os_unfair_lock_lock\$VARIANT\$

Track Filter All Instruments Threads CPUs

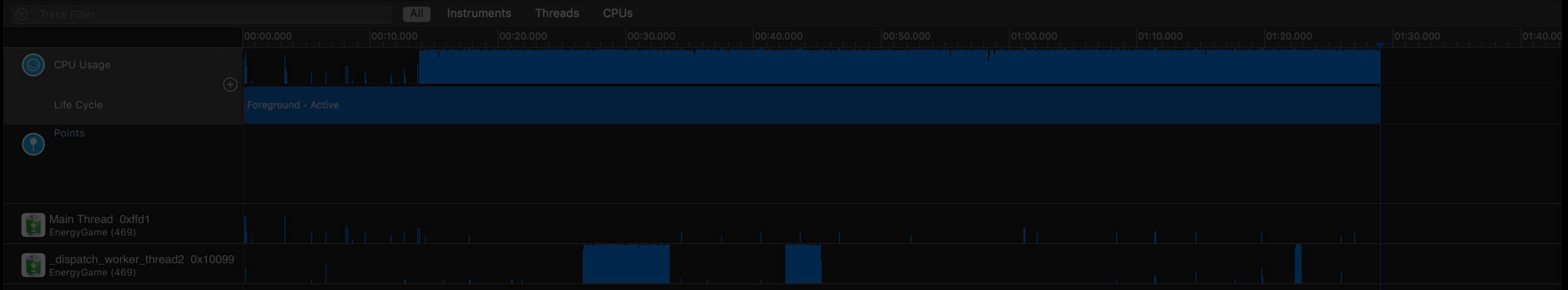


Time Profiler Profile Root

Weight	Self Weight	Symbol Name
1.24 min 100.0%	0 s	EnergyGame (469)
25.70 s 34.4%	0 s	_dispatch_workloop_worker_thread 0x1009b
25.70 s 34.4%	0 s	_thread_wqthread libsystem_thread.dylib
25.66 s 34.3%	0 s	_dispatch_worker_thread2 libdispatch.dylib
25.66 s 34.3%	0 s	_dispatch_root_queue_drain libdispatch.dylib
25.66 s 34.3%	0 s	_dispatch_source_invoke\$VARIANT\$armv81 libdispatch.dylib
25.65 s 34.3%	0 s	_dispatch_continuation_pop\$VARIANT\$armv81 libdispatch.dylib
25.65 s 34.3%	0 s	_dispatch_client_callout libdispatch.dylib
25.65 s 34.3%	0 s	thunk for @escaping @callee_guaranteed () -> () EnergyGame
25.65 s 34.3%	0 s	closure #1 in static ViewController.heavyTimer() EnergyGame
19.46 s 26.0%	503.00 ms	arc4random libsystem_c.dylib
6.86 s 9.1%	6.86 s	os_unfair_lock_lock\$VARIANT\$armv81 libsystem_platform.dylib
5.67 s 7.5%	24.00 ms	generate libcorecrypto.dylib
2.61 s 3.4%	5.00 ms	drbg_update libcorecrypto.dylib
803.00 ms 1.0%	508.00 ms	drbg_update libcorecrypto.dylib
281.00 ms 0.3%	281.00 ms	ccaes_vng_ctr_crypt libcorecrypto.dylib
14.00 ms 0.0%	14.00 ms	memset_s libsystem_c.dylib
799.00 ms 1.0%	0 s	ccaes_arm_encrypt_key libcorecrypto.dylib
785.00 ms 1.0%	785.00 ms	aes_key_expansion libcorecrypto.dylib
11.00 ms 0.0%	11.00 ms	ccaes_arm_encrypt_key libcorecrypto.dylib
3.00 ms 0.0%	3.00 ms	ccaes_arm_encrypt_key128 libcorecrypto.dylib
347.00 ms 0.4%	40.00 ms	ccaes_vng_ctr_crypt libcorecrypto.dylib
198.00 ms 0.2%	198.00 ms	ccaes_arm_encrypt_ecb libcorecrypto.dylib
80.00 ms 0.1%	80.00 ms	Decrypt_Main_Loop_End libcorecrypto.dylib
26.00 ms 0.0%	26.00 ms	ccaes_vng_ctr_crypt libcorecrypto.dylib

Heaviest Stack Trace

74693.0	EnergyGame (469)
25695.0	_dispatch_workloop_worker_th
25695.0	_pthread_wqthread
25659.0	_dispatch_worker_thread2
25659.0	_dispatch_root_queue_drain
25656.0	_dispatch_source_invoke\$VAR
25652.0	_dispatch_continuation_pop\$
25652.0	_dispatch_client_callout
25652.0	thunk for @escaping @callee_
25649.0	closure #1 in static ViewConti
19459.0	arc4random
6860.0	os_unfair_lock_lock\$VARIANT\$

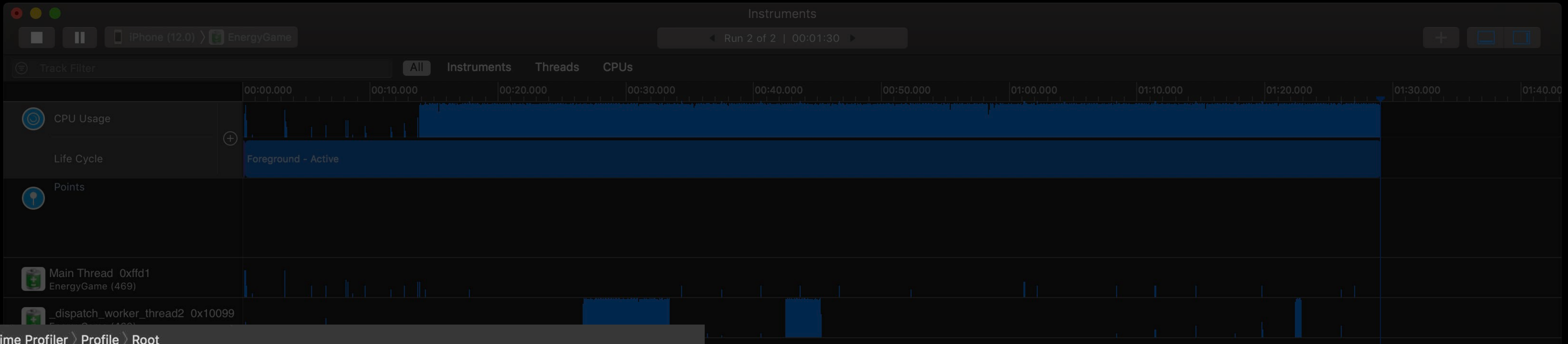


Time Profiler Profile Root

Weight	Self Weight	Symbol Name
1.24 min 100.0%	0 s	EnergyGame (469)
25.70 s 34.4%	0 s	_dispatch_workloop_worker_thread 0x1009b
25.70 s 34.4%	0 s	_dispatch_wqthread libsystem_pthread.dylib
25.66 s 34.3%	0 s	_dispatch_worker_thread2 libdispatch.dylib
25.66 s 34.3%	0 s	_dispatch_root_queue_drain libdispatch.dylib
25.66 s 34.3%	0 s	_dispatch_source_invoke\$VARIANT\$armv81 libdispatch.dylib
25.65 s 34.3%	0 s	_dispatch_continuation_pop\$VARIANT\$armv81 libdispatch.dylib
25.65 s 34.3%	0 s	_dispatch_client_callout libdispatch.dylib
25.65 s 34.3%	0 s	thunk for @escaping @callee_guaranteed () -> () EnergyGame
25.65 s 34.3%	0 s	closure #1 in static ViewController.heavyTimer() EnergyGame
19.46 s 26.0%	503.00 ms	arc4random libsystem_c.dylib
6.86 s 9.1%	6.86 s	os_unfair_lock_lock\$VARIANT\$armv81 libsystem_platform.dylib
5.67 s 7.5%	24.00 ms	generate libcorecrypto.dylib
2.61 s 3.4%	5.00 ms	drbg_update libcorecrypto.dylib
803.00 ms 1.0%	508.00 ms	drbg_update libcorecrypto.dylib
281.00 ms 0.3%	281.00 ms	ccaes_vng_ctr_crypt libcorecrypto.dylib
14.00 ms 0.0%	14.00 ms	memset_s libsystem_c.dylib
799.00 ms 1.0%	0 s	ccaes_arm_encrypt_key libcorecrypto.dylib
785.00 ms 1.0%	785.00 ms	aes_key_expansion libcorecrypto.dylib
11.00 ms 0.0%	11.00 ms	ccaes_arm_encrypt_key libcorecrypto.dylib
3.00 ms 0.0%	3.00 ms	ccaes_arm_encrypt_key128 libcorecrypto.dylib
347.00 ms 0.4%	40.00 ms	ccaes_vng_ctr_crypt libcorecrypto.dylib
198.00 ms 0.2%	198.00 ms	ccaes_arm_encrypt_ecb libcorecrypto.dylib
80.00 ms 0.1%	80.00 ms	Decrypt_Main_Loop_End libcorecrypto.dylib
26.00 ms 0.0%	26.00 ms	ccaes_vng_ctr_crypt libcorecrypto.dylib

Heaviest Stack Trace

74693.0	EnergyGame (469)
25695.0	_dispatch_workloop_worker_th
25695.0	_pthread_wqthread
25659.0	_dispatch_worker_thread2
25659.0	_dispatch_root_queue_drain
25656.0	_dispatch_source_invoke\$VAR
25652.0	_dispatch_continuation_pop\$
25652.0	_dispatch_client_callout
25649.0	thunk for @escaping @callee_
19459.0	arc4random
6860.0	os_unfair_lock_lock\$VARIANT\$



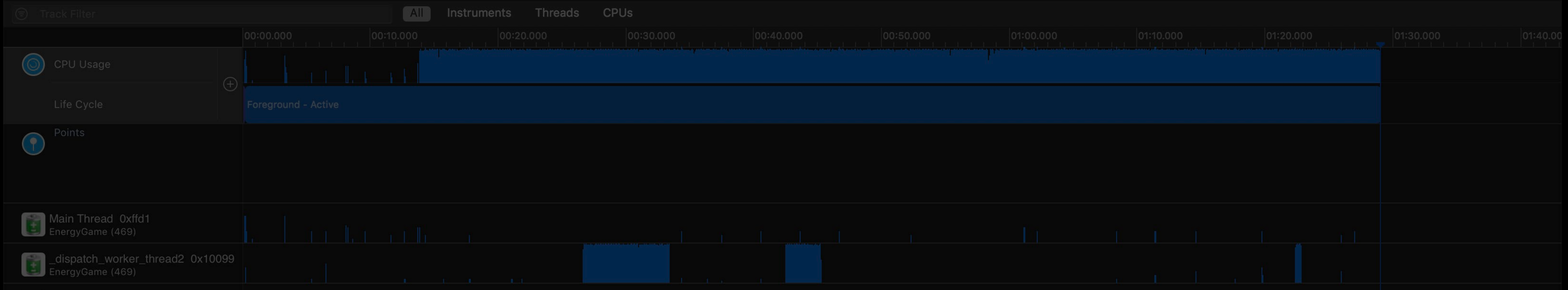
Time Profiler > Profile > Root

Weight	Self Weight	Symbol Name
1.24 min	100.0%	0 s EnergyGame (469)
25.70 s	34.4%	0 s _dispatch_workloop_worker_thread 0x1009b
25.70 s	34.4%	0 s _pthread_wqthread libsystem_pthread.dylib
25.66 s	34.3%	0 s _dispatch_worker_thread2 libdispatch.dylib
25.66 s	34.3%	0 s _dispatch_root_queue_drain libdispatch.dylib
25.66 s	34.3%	0 s _dispatch_source_invoke\$VARIANT\$armv81 libdispatch.dylib
25.65 s	34.3%	0 s _dispatch_continuation_pop\$VARIANT\$armv81 libdispatch.dylib
25.65 s	34.3%	0 s _dispatch_client_callout libdispatch.dylib
25.65 s	34.3%	0 s think for @escaping @callee_guaranteed () -> () EnergyGame
25.65 s	34.3%	0 s closure #1 in static ViewController.heavyTimer() EnergyGame
19.46 s	26.0%	503.00 ms arc4random libsystem_c.dylib
6.86 s	9.1%	6.86 s os_unfair_lock_lock\$VARIANT\$armv81 libsystem_platform.dylib
5.67 s	7.5%	24.00 ms generate libcorecrypto.dylib
2.61 s	3.4%	5.00 ms drbg_update libcorecrypto.dylib
803.00 ms	1.0%	508.00 ms drbg_update libcorecrypto.dylib
281.00 ms	0.3%	281.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib
14.00 ms	0.0%	14.00 ms memset_s libsystem_c.dylib
799.00 ms	1.0%	0 s ccaes_arm_encrypt_key libcorecrypto.dylib
785.00 ms	1.0%	785.00 ms aes_key_expansion libcorecrypto.dylib
11.00 ms	0.0%	11.00 ms ccaes_arm_encrypt_key libcorecrypto.dylib
3.00 ms	0.0%	3.00 ms ccaes_arm_encrypt_key128 libcorecrypto.dylib
347.00 ms	0.4%	40.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib
198.00 ms	0.2%	198.00 ms ccaes_arm_encrypt_ecb libcorecrypto.dylib
80.00 ms	0.1%	80.00 ms Decrypt_Main_Loop_End libcorecrypto.dylib
26.00 ms	0.0%	26.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib

Input Filter: Involves Symbol | Call Tree | Call Tree Constraints | Data Mining

Heaviest Stack Trace

74693.0	EnergyGame (469)
25695.0	_dispatch_workloop_worker_th
25695.0	_pthread_wqthread
25659.0	_dispatch_worker_thread2
25659.0	_dispatch_root_queue_drain
25656.0	_dispatch_source_invoke\$VAR
25652.0	_dispatch_continuation_pop\$
25652.0	_dispatch_client_callout
25652.0	think for @escaping @callee_
25649.0	closure #1 in static ViewConti
19459.0	arc4random
6860.0	os_unfair_lock_lock\$VARIANT\$



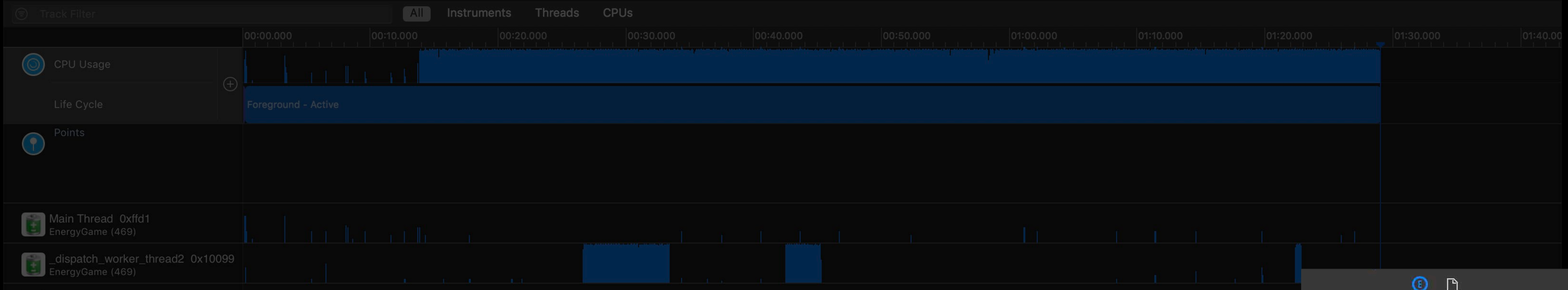
Time Profiler Profile Root

Weight	Self Weight	Symbol Name
1.24 min	100.0%	0 s EnergyGame (469)
25.70 s	34.4%	0 s _dispatch_workloop_worker_thread 0x1009b
25.70 s	34.4%	0 s _pthread_wqthread libsystem_pthread.dylib
25.66 s	34.3%	0 s _dispatch_worker_thread2 libdispatch.dylib
25.66 s	34.3%	0 s _dispatch_root_queue_drain libdispatch.dylib
25.66 s	34.3%	0 s _dispatch_source_invoke\$VARIANT\$armv81 libdispatch.dylib
25.65 s	34.3%	0 s _dispatch_continuation_pop\$VARIANT\$armv81 libdispatch.dylib
25.65 s	34.3%	0 s _dispatch_client_callout libdispatch.dylib
25.65 s	34.3%	0 s thunk for @escaping @callee_guaranteed () -> () EnergyGame
25.65 s	34.3%	0 s closure #1 in static ViewController.heavyTimer() EnergyGame
19.46 s	26.0%	503.00 ms arc4random libsystem_c.dylib
6.86 s	9.1%	6.86 s os_unfair_lock_lock\$VARIANT\$armv81 libsystem_platform.dylib
5.67 s	7.5%	24.00 ms generate libcorecrypto.dylib
2.61 s	3.4%	5.00 ms drbg_update libcorecrypto.dylib
803.00 ms	1.0%	508.00 ms drbg_update libcorecrypto.dylib
281.00 ms	0.3%	281.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib
14.00 ms	0.0%	14.00 ms memset_s libsystem_c.dylib
799.00 ms	1.0%	0 s ccaes_arm_encrypt_key libcorecrypto.dylib
785.00 ms	1.0%	785.00 ms aes_key_expansion libcorecrypto.dylib
11.00 ms	0.0%	11.00 ms ccaes_arm_encrypt_key libcorecrypto.dylib
3.00 ms	0.0%	3.00 ms ccaes_arm_encrypt_key128 libcorecrypto.dylib
347.00 ms	0.4%	40.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib
198.00 ms	0.2%	198.00 ms ccaes_arm_encrypt_ecb libcorecrypto.dylib
80.00 ms	0.1%	80.00 ms Decrypt_Main_Loop_End libcorecrypto.dylib
26.00 ms	0.0%	26.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib

Heaviest Stack Trace

- 74693.0 EnergyGame (469)
- 25695.0 _dispatch_workloop_worker_th
- 25695.0 _pthread_wqthread
- 25659.0 _dispatch_worker_thread2
- 25659.0 _dispatch_root_queue_drain
- 25656.0 _dispatch_source_invoke\$VAR
- 25652.0 _dispatch_continuation_pop\$1
- 25652.0 _dispatch_client_callout
- 25652.0 thunk for @escaping @callee_
- 25649.0 closure #1 in static ViewConti
- 19459.0 arc4random
- 6860.0 os_unfair_lock_lock\$VARIANT\$

Input Filter Involves Symbol Call Tree Call Tree Constraints Data Mining



Time Profiler Profile Root

Weight	Self Weight	Symbol Name
1.24 min	100.0%	0 s EnergyGame (469)
25.70 s	34.4%	0 s _dispatch_workloop_worker_thread 0x1009b
25.70 s	34.4%	0 s _pthread_wqthread libsystem_pthread.dylib
25.66 s	34.3%	0 s _dispatch_worker_thread2 libdispatch.dylib
25.66 s	34.3%	0 s _dispatch_root_queue_drain libdispatch.dylib
25.66 s	34.3%	0 s _dispatch_source_invoke\$VARIANT\$armv81 libdispatch.dylib
25.65 s	34.3%	0 s _dispatch_continuation_pop\$VARIANT\$armv81 libdispatch.dylib
25.65 s	34.3%	0 s _dispatch_client_callout libdispatch.dylib
25.65 s	34.3%	0 s think for @escaping @callee_guaranteed () -> () EnergyGame
25.65 s	34.3%	0 s closure #1 in static ViewController.heavyTimer() EnergyGame
19.46 s	26.0%	503.00 ms arc4random libsystem_c.dylib
6.86 s	9.1%	6.86 s os_unfair_lock_lock\$VARIANT\$armv81 libsystem_platform.dylib
5.67 s	7.5%	24.00 ms generate libcorecrypto.dylib
2.61 s	3.4%	5.00 ms drbg_update libcorecrypto.dylib
803.00 ms	1.0%	508.00 ms drbg_update libcorecrypto.dylib
281.00 ms	0.3%	281.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib
14.00 ms	0.0%	14.00 ms memset_s libsystem_c.dylib
799.00 ms	1.0%	0 s ccaes_arm_encrypt_key libcorecrypto.dylib
785.00 ms	1.0%	785.00 ms aes_key_expansion libcorecrypto.dylib
11.00 ms	0.0%	11.00 ms ccaes_arm_encrypt_key libcorecrypto.dylib
3.00 ms	0.0%	3.00 ms ccaes_arm_encrypt_key128 libcorecrypto.dylib
347.00 ms	0.4%	40.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib
198.00 ms	0.2%	198.00 ms ccaes_arm_encrypt_ecb libcorecrypto.dylib
80.00 ms	0.1%	80.00 ms Decrypt_Main_Loop_End libcorecrypto.dylib
26.00 ms	0.0%	26.00 ms ccaes_vng_ctr_crypt libcorecrypto.dylib

Heaviest Stack Trace

- 74693.0 EnergyGame (469)
- 25695.0 _dispatch_workloop_worker_thread
- 25695.0 _pthread_wqthread
- 25659.0 _dispatch_worker_thread2
- 25659.0 _dispatch_root_queue_drain
- 25656.0 _dispatch_source_invoke\$VAR
- 25652.0 _dispatch_continuation_pop\$1
- 25652.0 _dispatch_client_callout
- 25652.0 think for @escaping @callee_
- 25649.0 closure #1 in static ViewConti
- 19459.0 arc4random
- 6860.0 os_unfair_lock_lock\$VARIANT\$

Other Profiling Templates and Instruments

Other Profiling Templates and Instruments

Energy diagnostics profiling template

Activity monitor profiling template

Core animation profiling template

GPU driver profiling template

Location energy instrument

Network profiling template

Instruments Are Great for...

Instruments Are Great for...

Root cause analysis

In depth profiling

Instruments Are Great for...

Root cause analysis

In depth profiling

Untethered profiling

Working Demo

Energy Debugging

Working Demo

Takeaways

Energy Gauges for rapid iteration

Instruments for root cause analysis

Make energy a top priority

You've shipped your app.
Whats next?

NEW

Review general battery life concepts

Review tools for energy debugging

New features for energy debugging

New Features for Energy Debugging

David Choi, Battery Life

Now that You've Shipped Your App...

How do you know if your customers are experiencing energy issues?

How do you debug a customer energy issue?



A New Way of Debugging Energy Issues



NEW

Xcode Energy Logs

Xcode Energy Organizer

Xcode Energy Logs

NEW

Reports high CPU energy events
Points out energy hotspots in code
Data from TestFlight and App Store



When Are Xcode Energy Logs Generated?

NEW

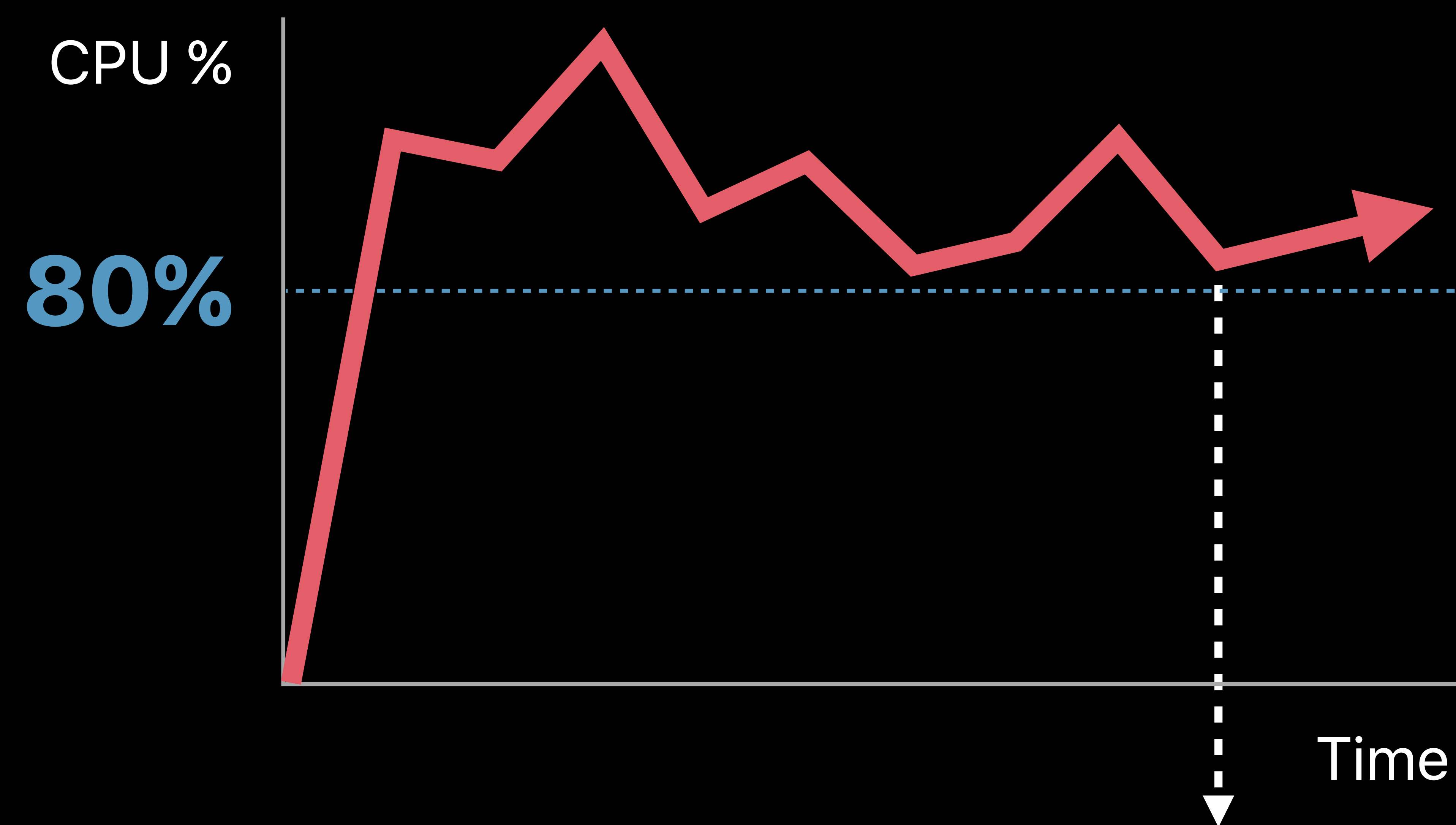


High CPU Energy Events

- 80% CPU spin over 3 minutes in foreground
- 80% CPU spin over 1 minutes in background

When Are Xcode Energy Logs Generated?

NEW



High CPU Energy Events

- 80% CPU spin over 3 minutes in foreground
- 80% CPU spin over 1 minutes in background

Xcode Energy Log

Xcode Energy Logs: Impact on User Battery Life

Up to 1% battery drop

Xcode Energy Logs: Impact on User Battery Life

Up to 1% battery drop



8 minutes
of talk time



6 minutes
of browsing



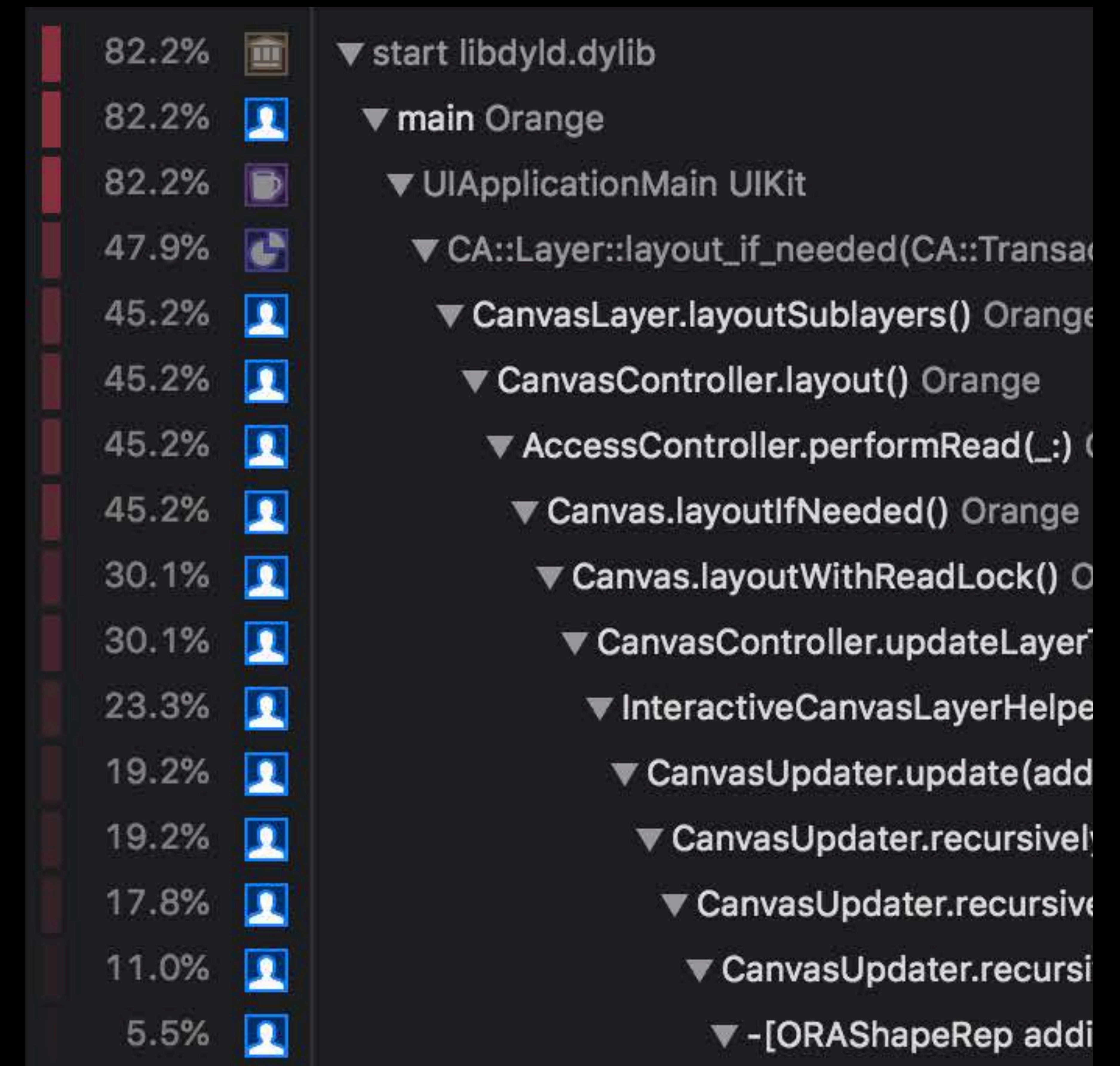
30 minutes
of music

What's in an Energy Log?

Energy condition that triggered the report

Device type and app build number

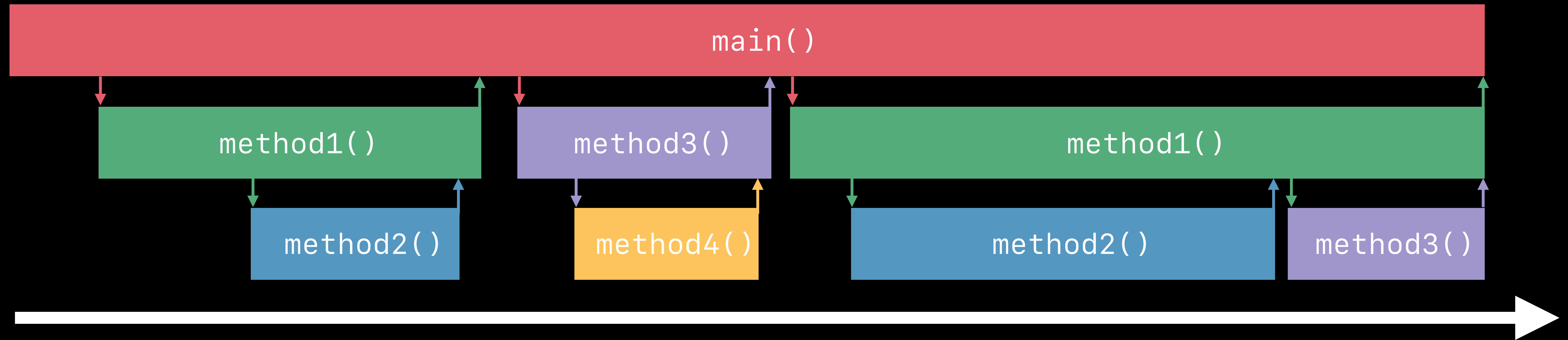
Weighted call graph that shows energy hotspots



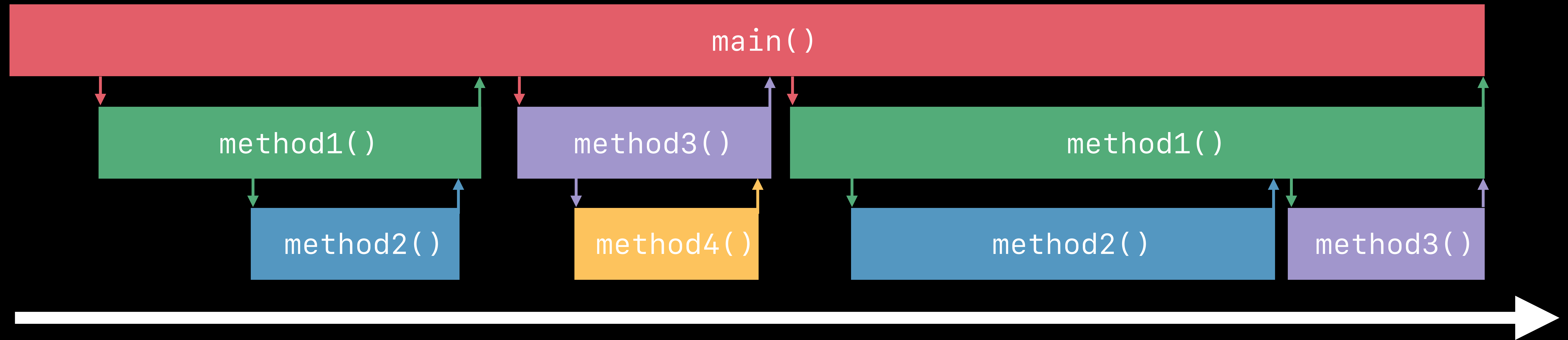
Xcode Energy Log: Weighted Call Graph



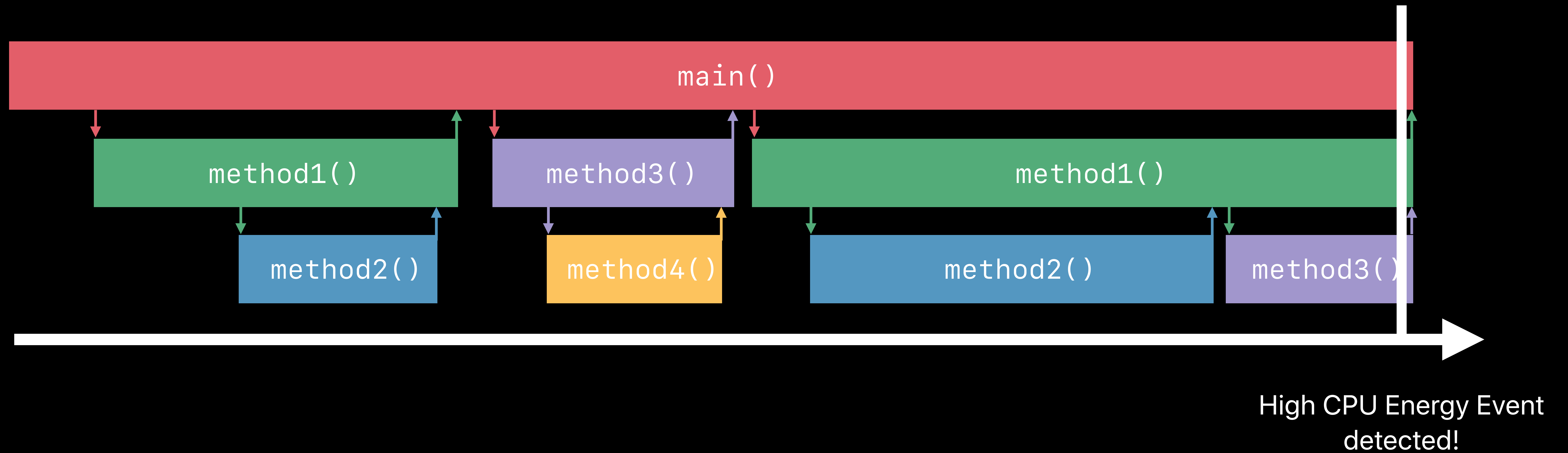
Xcode Energy Log: Weighted Call Graph



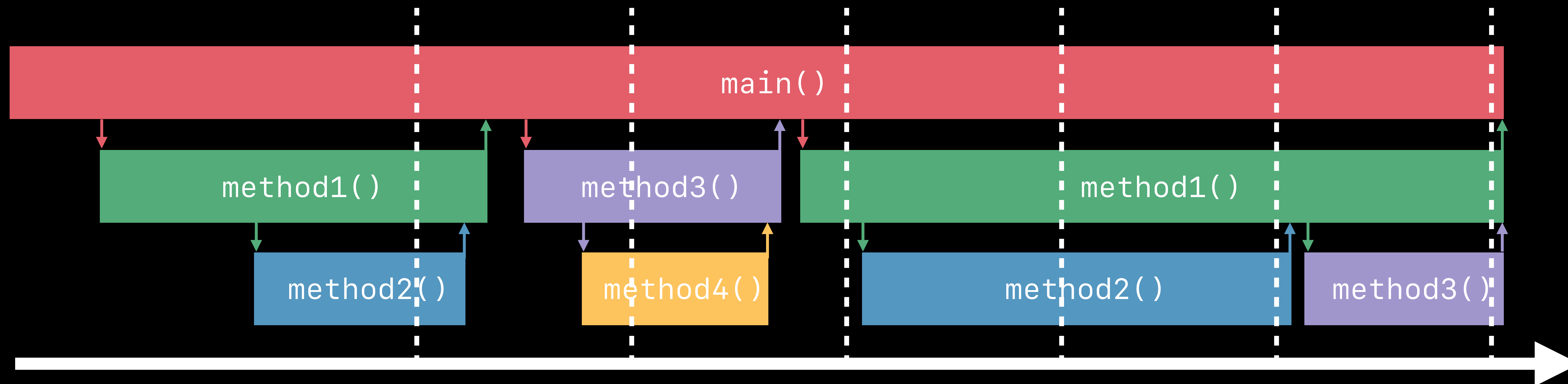
Xcode Energy Log: Weighted Call Graph



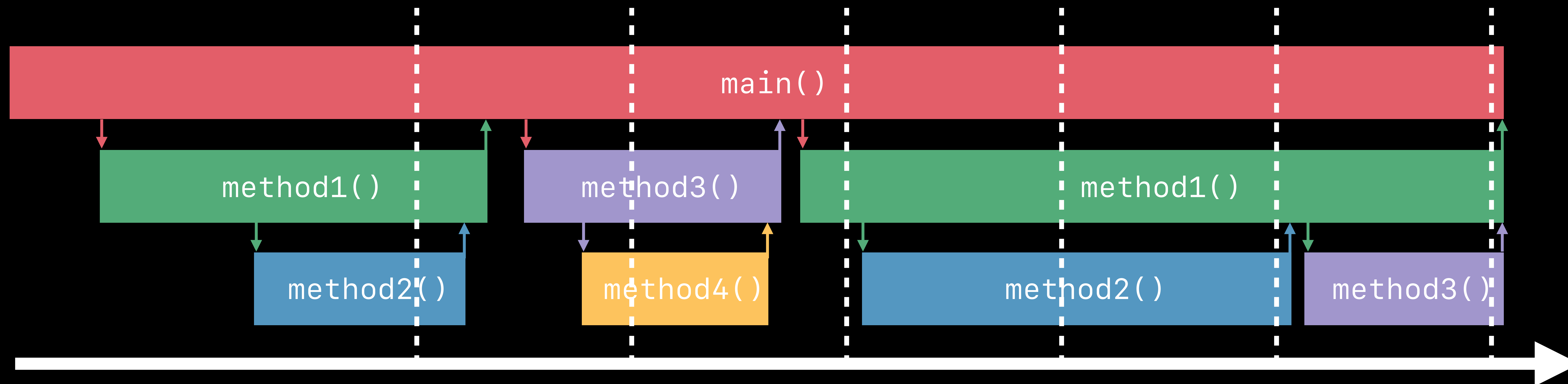
Xcode Energy Log: Weighted Call Graph



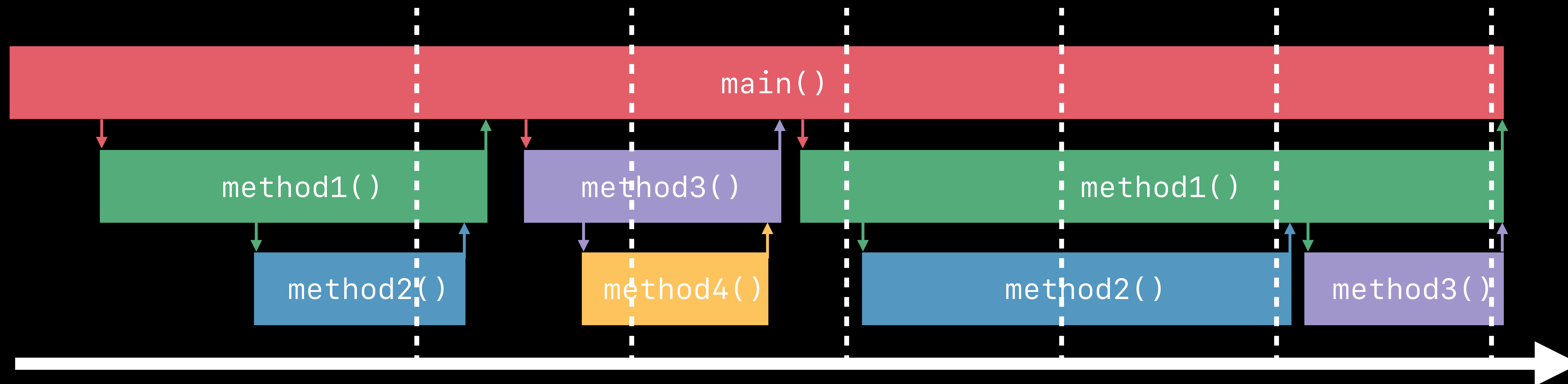
Xcode Energy Log: Weighted Call Graph



Xcode Energy Log: Weighted Call Graph

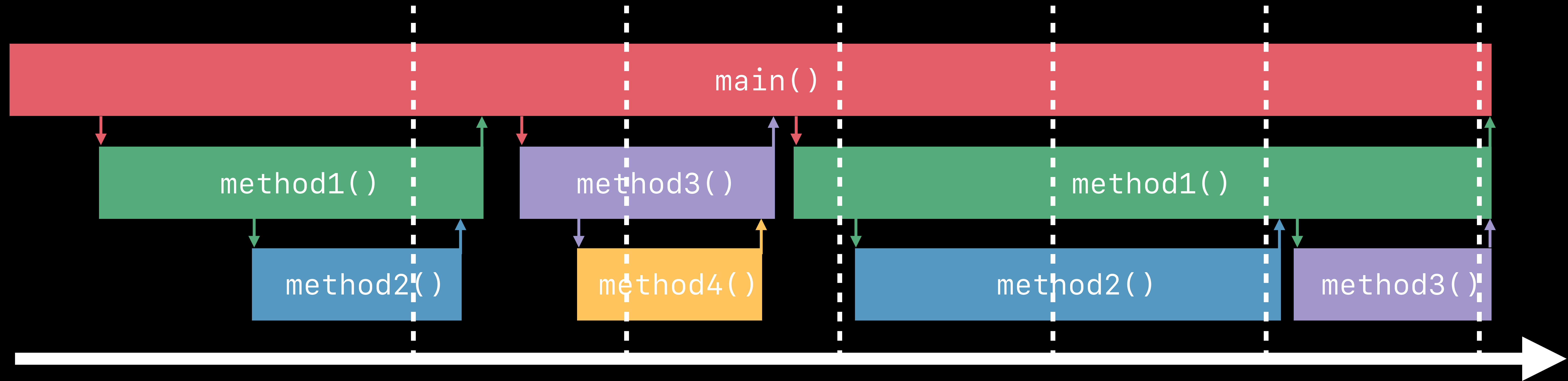


Xcode Energy Log: Weighted Call Graph



```
main()  
method1()  
method2()
```

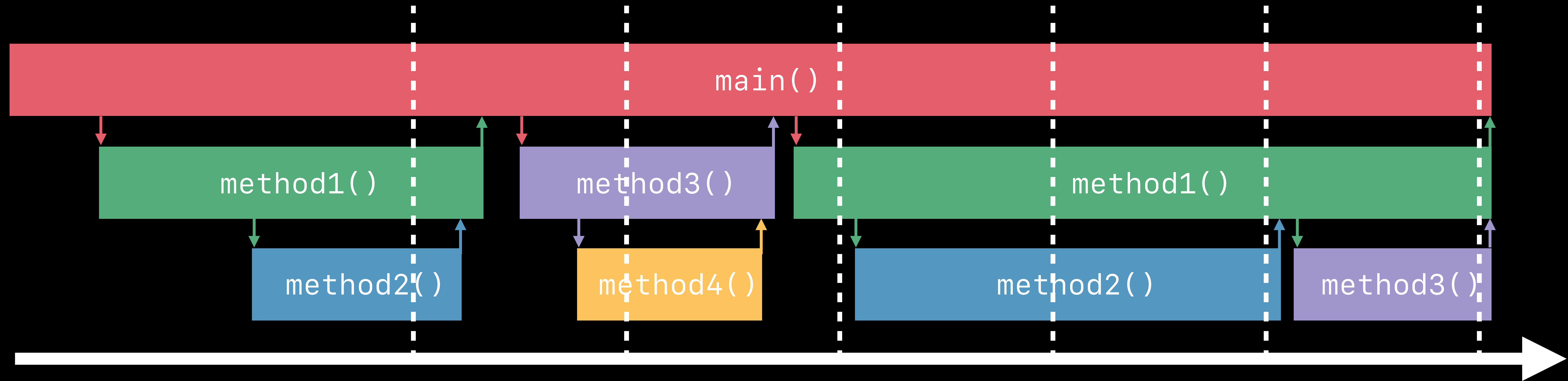
Xcode Energy Log: Weighted Call Graph



main()
method1()
method2()

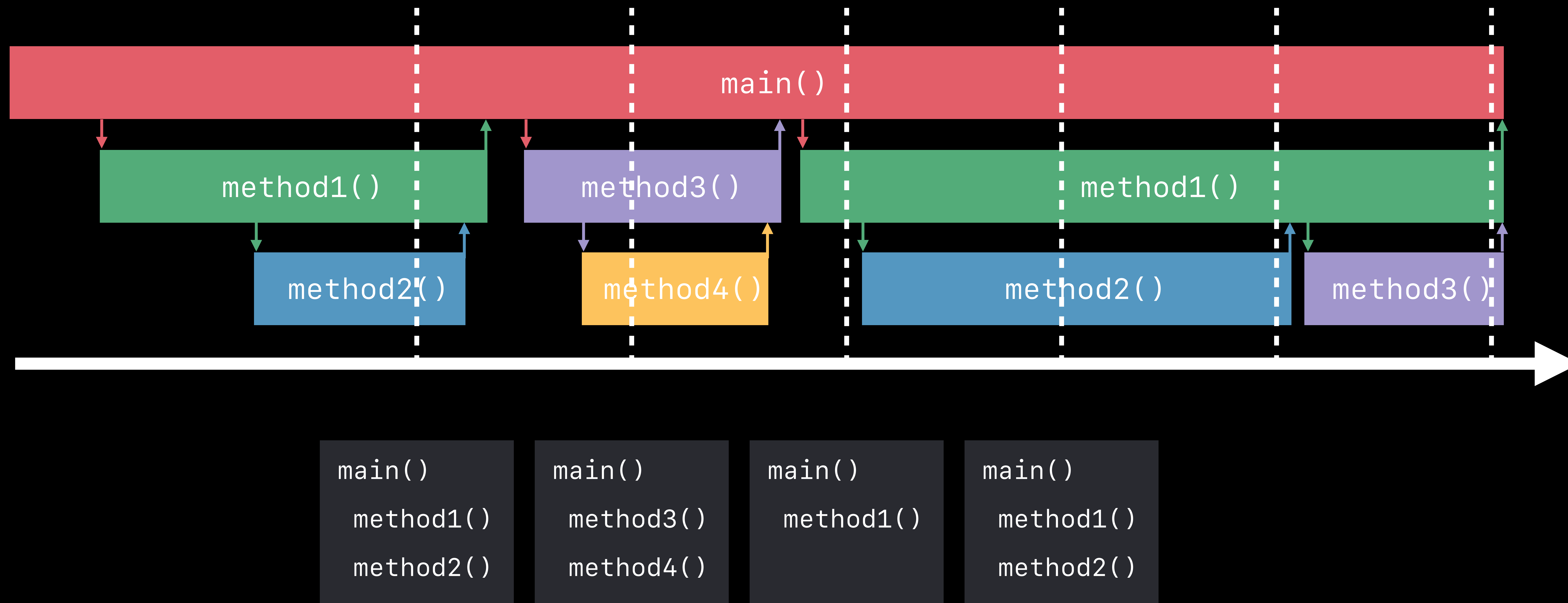
main()
method3()
method4()

Xcode Energy Log: Weighted Call Graph

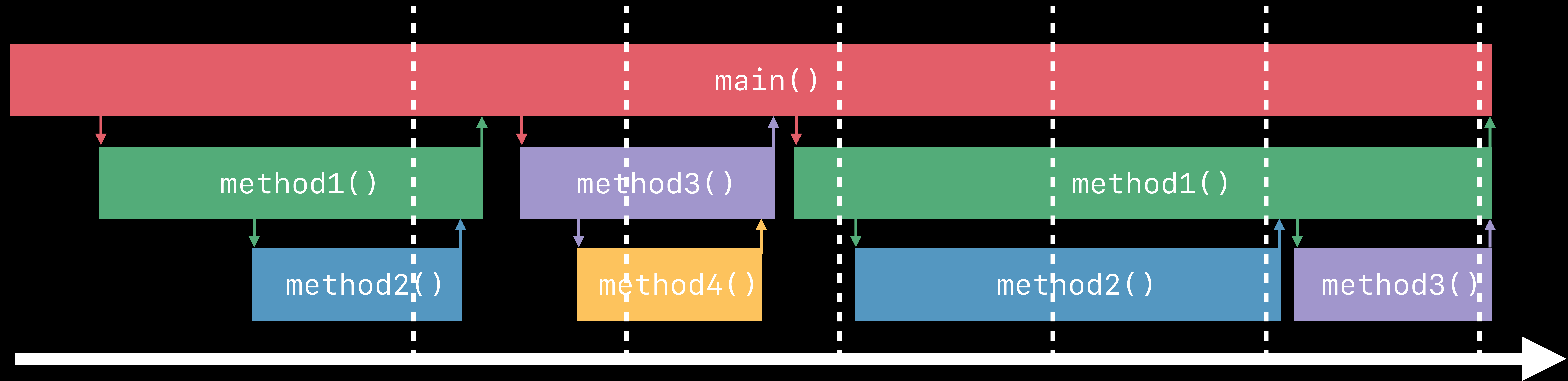


<code>main()</code>	<code>main()</code>	<code>main()</code>
<code>method1()</code>	<code>method3()</code>	<code>method1()</code>
<code>method2()</code>	<code>method4()</code>	

Xcode Energy Log: Weighted Call Graph



Xcode Energy Log: Weighted Call Graph



```
main()  
method1()  
method2()
```

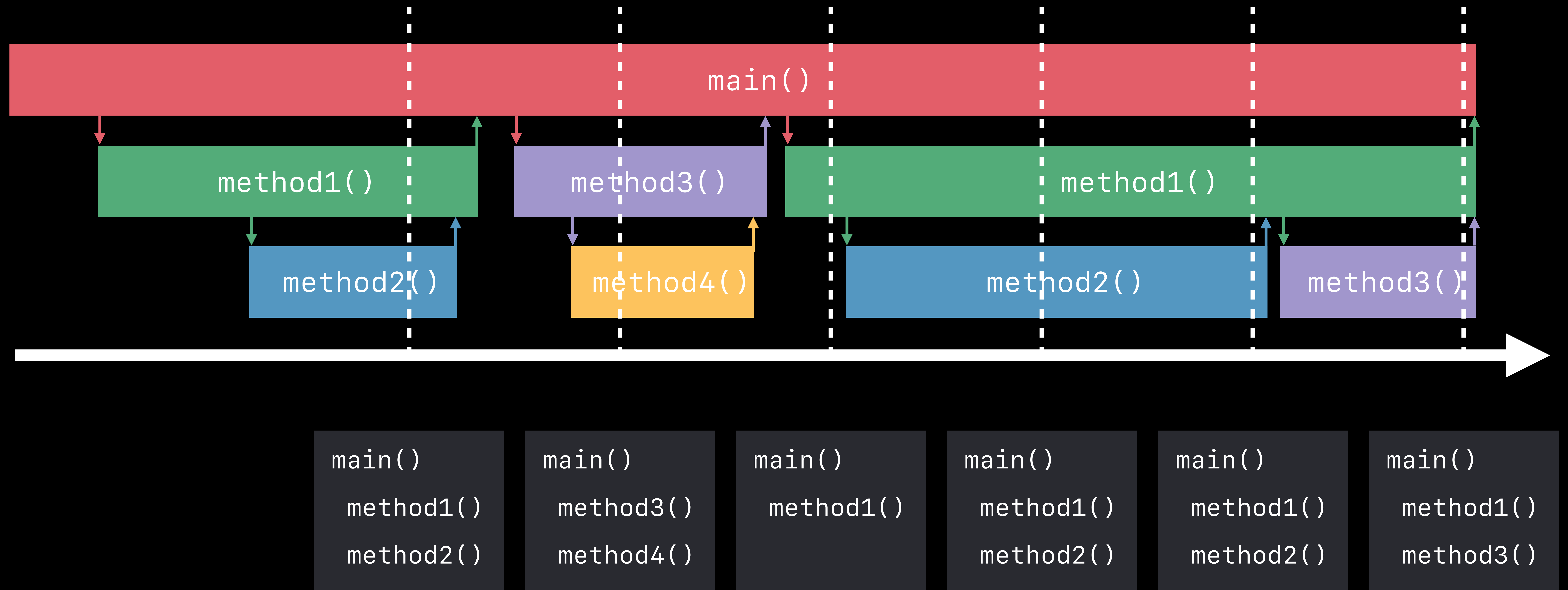
```
main()  
method3()  
method4()
```

```
main()  
method1()
```

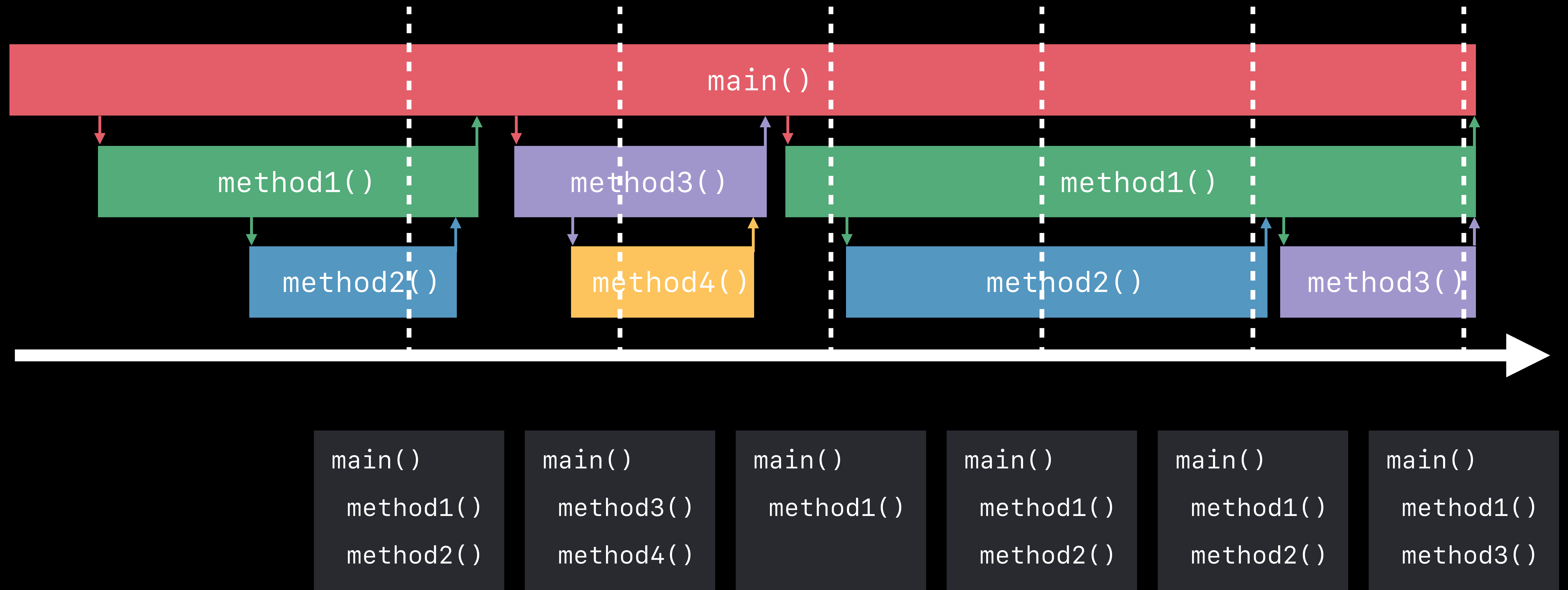
```
main()  
method1()  
method2()
```

```
main()  
method1()  
method2()
```

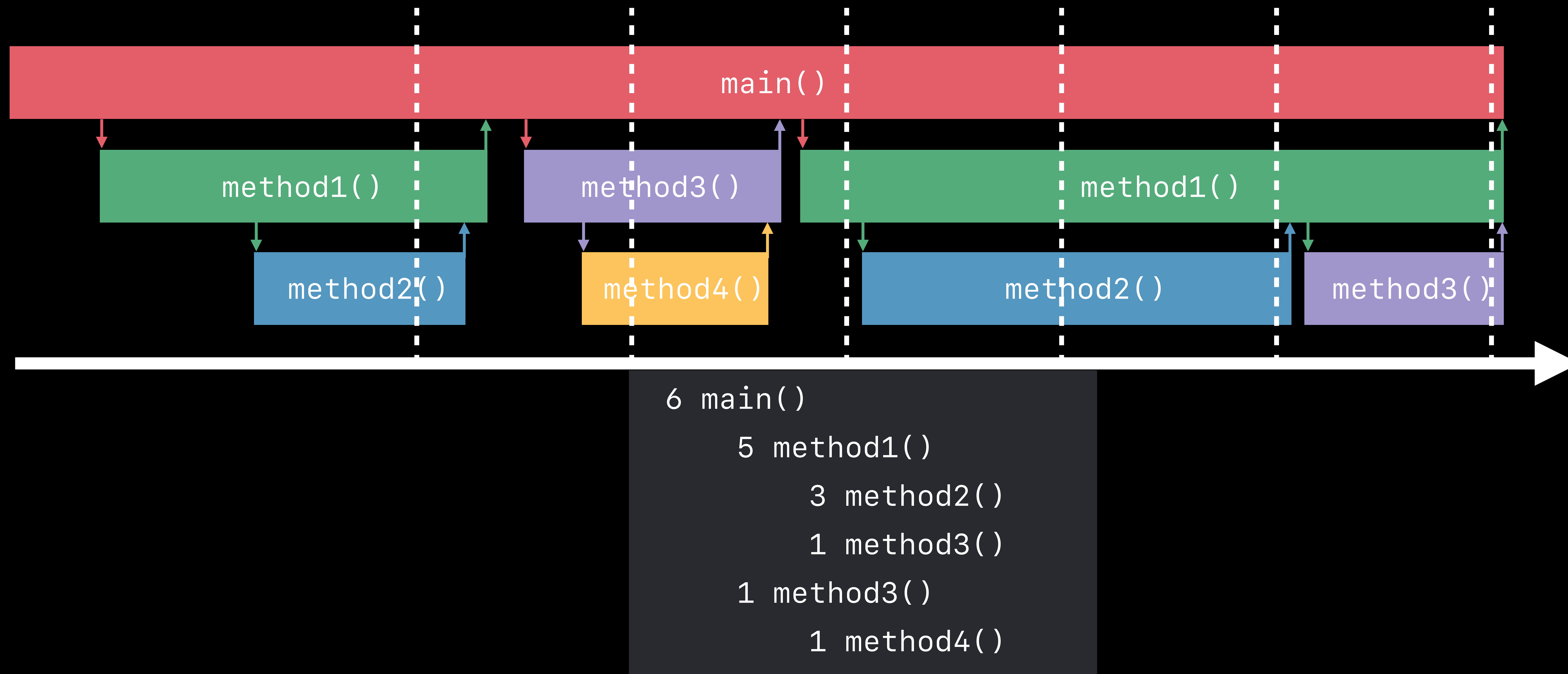
Xcode Energy Log: Weighted Call Graph



Xcode Energy Log: Weighted Call Graph



Xcode Energy Log: Weighted Call Graph



Weighted Call Graph

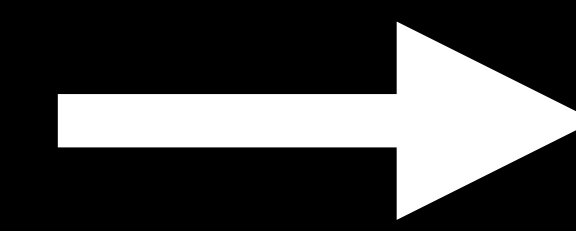
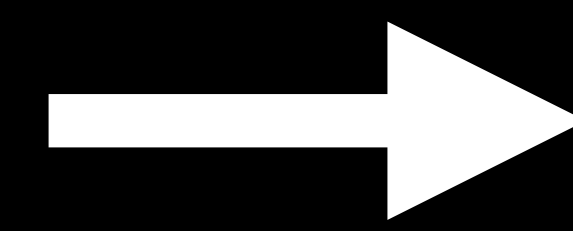
Collection of backtrace samples

Aggregated by sample counts

More samples mean heavily executed code

```
6 main()  
  5 method1()  
    3 method2()  
    1 method3()  
1 method3()  
  1 method4()
```

How to Access Energy Logs?



Xcode Energy Organizer

NEW

The screenshot displays the Xcode Energy Organizer interface, which provides a detailed breakdown of energy usage for an application. The interface is divided into several sections:

- App Store / Build 17:** Lists various energy-consuming tasks with their respective device counts and percentages. The top item is "Orange: Canvas.layoutIfNeeded() + 324" on 128 devices, accounting for 82.2% of the energy usage.
- Energy Usage Summary:** Shows the total energy usage for the selected task, "Orange: Canvas.layoutIfNeeded()", as 128 devices.
- Energy Log Details:** Provides specific details for the selected task, including the binary version (Orange 3.0 (17)), the device type (iPad Pro (10.5-inch) (Model...)), and the CPU usage type (High CPU Usage). It also notes a detail of 144 seconds CPU time over 178 seconds (80% CPU average), exceeding a limit of 80% CPU over 180 seconds.
- Last 2 Weeks:** A bar chart showing energy usage trends over time, with a total of 128 devices. The chart includes a legend for device types: iPad (68%), iPod (13%), and iPhone (19%).
- Notes:** A section for adding notes, currently showing "No Notes".
- Previous Top Energy Logs:** A list of other high-energy tasks, such as "OrangeKit: CanvasScheme.setUp() + 248" on 23 devices.

Xcode Energy Organizer

NEW

Test Flight and App Store iOS apps

The screenshot displays the Xcode Energy Organizer interface. On the left, a sidebar lists apps categorized by platform:

- iOS Apps**
 - Banana
 - Orange
 - Strawberry
- macOS Apps**
 - Pineapple
- tvOS Apps**
 - Kiwi

The main area shows a detailed energy log for an app named "Orange". The log is organized into columns: "Build", "Energy Usage", and "Energy Log". The "Energy Usage" column shows a total of 324 units for the selected build. The "Energy Log" column displays a hierarchical tree of energy-consuming operations, such as "start libdyld.dylib", "main Orange", and "Canvas.layoutIfNeeded()", with their respective energy costs and percentages.

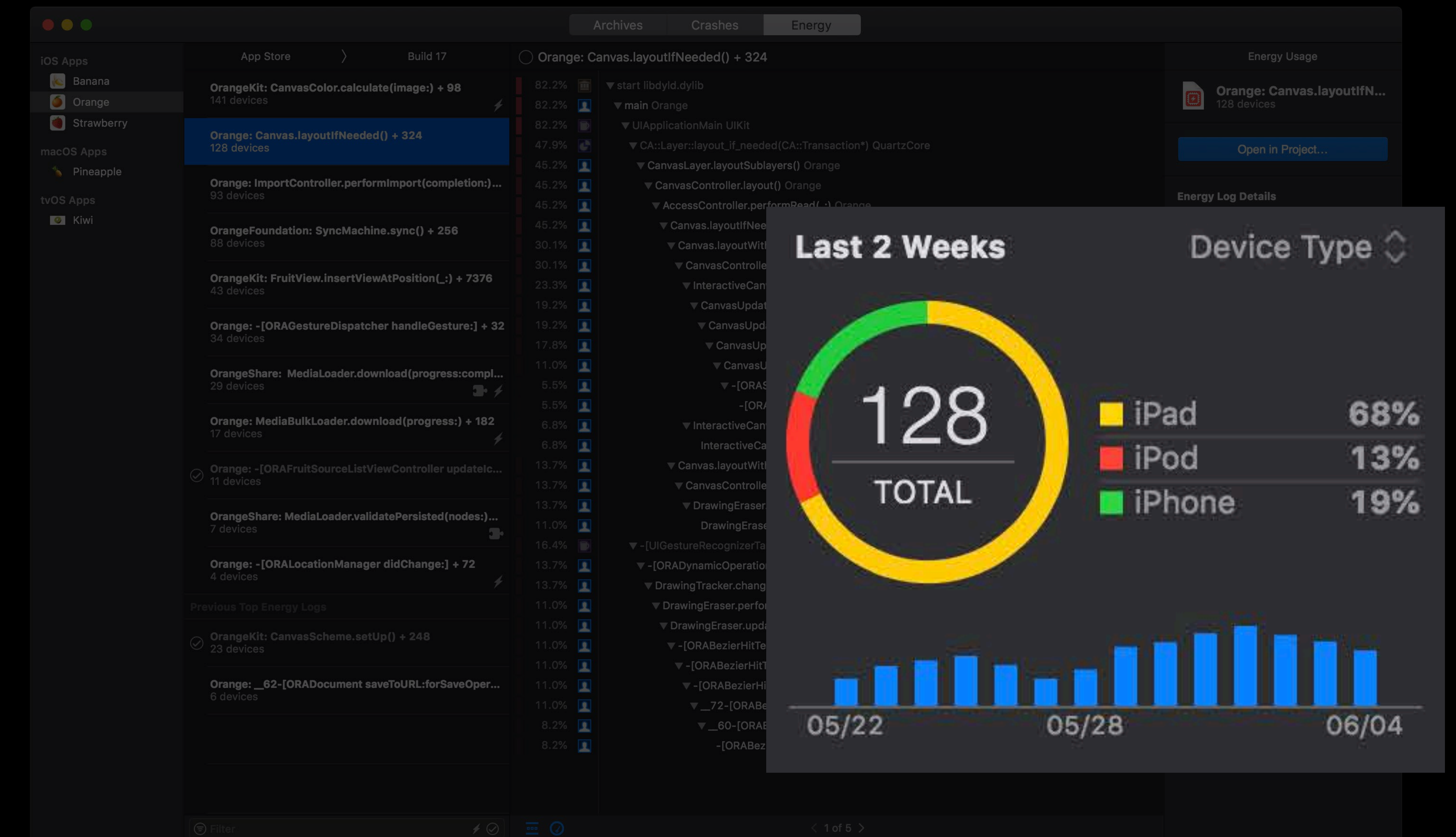
On the right side, there is a summary panel for the selected energy log entry, "Orange: Canvas.layoutIfNeeded() + 324". It includes a bar chart showing energy usage over time, a "Last 2 Weeks" summary with a total of 128 units, and a breakdown by device type: iPad (68%), iPhone (13%), and iPod (19%).

Xcode Energy Organizer

NEW

Test Flight and App Store iOS apps

Recent statistics



Xcode Energy Organizer

NEW

Test Flight and App Store iOS apps

Recent statistics

List of top issues

The screenshot displays the Xcode Energy Organizer interface. At the top, it shows 'App Store' and 'Build 17'. Below this is a list of energy issues:

- OrangeKit: CanvasColor.calculate(image:) + 98 (141 devices)
- Orange: Canvas.layoutIfNeeded() + 324 (128 devices)** (highlighted in blue)
- Orange: ImportController.performImport(completion:) ... (93 devices)
- OrangeFoundation: SyncMachine.sync() + 256 (88 devices)
- OrangeKit: FruitView.insertViewAtPosition(:) + 7376 (43 devices)
- Orange: -[ORAGestureDispatcher handleGesture:] + 32 (34 devices)
- OrangeShare: MediaLoader.download(progress:compl... (29 devices)

The right-hand side of the interface shows a detailed view of the top issue, 'Orange: Canvas.layoutIfNeeded()'. It includes an 'Energy Usage' section with a bar chart showing 'Last 2 Weeks' data. The chart shows a total of 128 devices, with 68% on iPad, 13% on iPod, and 19% on iPhone. Below the chart is a 'Notes' section with a 'Mark As Resolved' button.

Xcode Energy Organizer

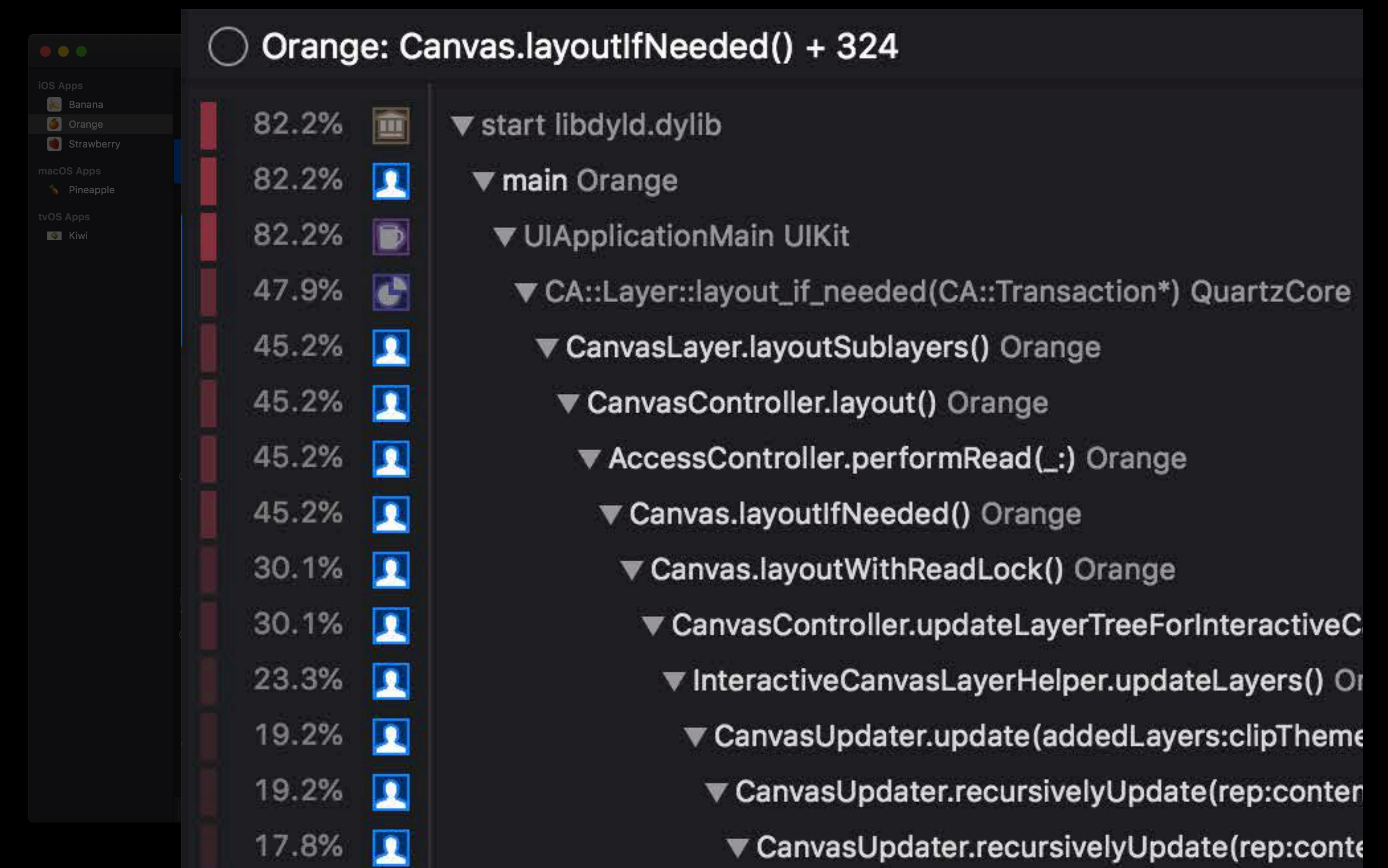
NEW

Test Flight and App Store iOS apps

Recent statistics

List of top issues

Weighted call graph



Xcode Energy Organizer

NEW

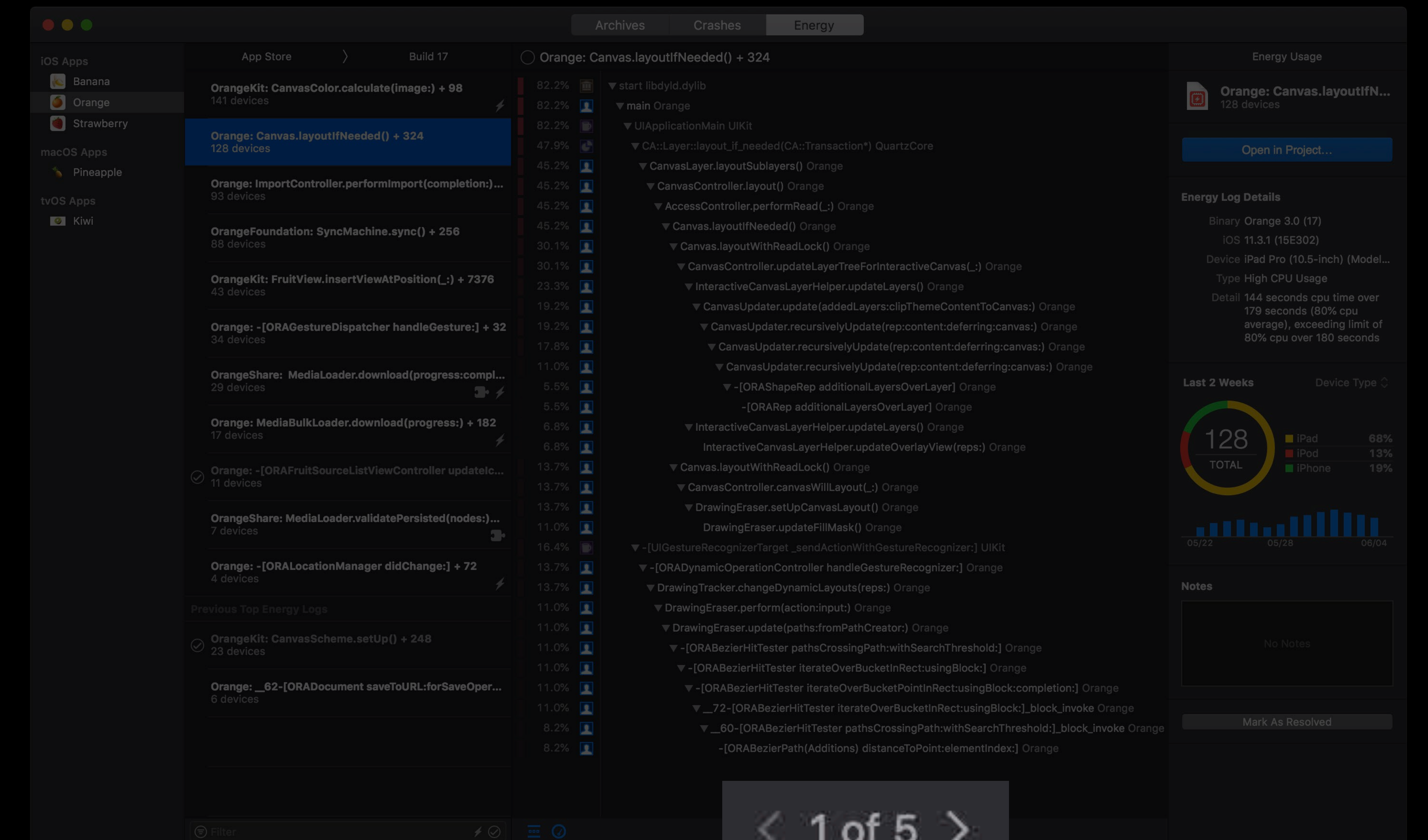
Test Flight and App Store iOS apps

Recent statistics

List of top issues

Weighted call graph

Page through logs



Xcode Energy Organizer

NEW

Test Flight and App Store iOS apps

Recent statistics

List of top issues

Weighted call graph

Page through logs

Open in Project

The screenshot displays the Xcode Energy Organizer interface. On the left, a sidebar lists various apps like Banana, Orange, Strawberry, etc. The main area shows a list of energy issues, such as 'OrangeKit: CanvasColor.calculate(image): + 98' and 'Orange: Canvas.layoutIfNeeded(): + 324'. A 'Weighted call graph' is visible, showing a hierarchical tree of method calls. On the right, a summary panel shows 'Last 2 Weeks' energy usage with a '128 TOTAL' indicator and a bar chart. A blue button labeled 'Open in Project...' is overlaid on the interface.

Demo

Xcode Energy Organizer

Energy Organizer

Demo takeaways

Discover top energy issues from the field

View energy hotspots with weighted call graphs

“Open in Project” to debug code

Summary

Think about energy use in design

Use energy gauges and instruments

Explore the new Xcode Energy Organizer

More Information

<https://developer.apple.com/go/?id=ios-energy-efficiency-guide>

<https://developer.apple.com/videos/play/wwdc2017/238/>

 **WWDC18**