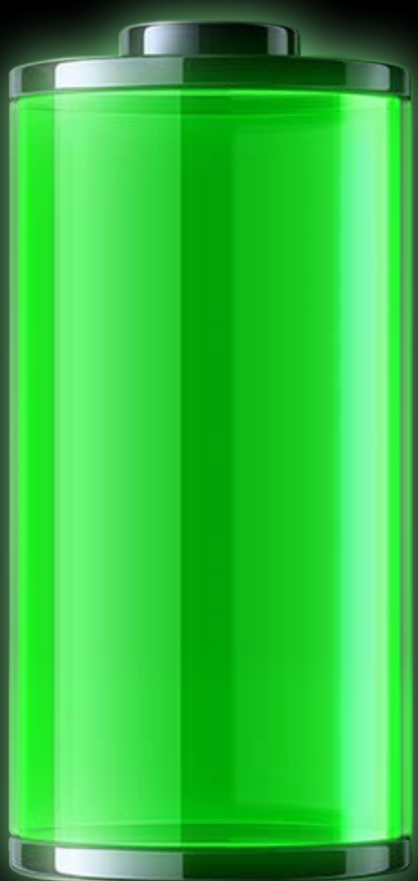
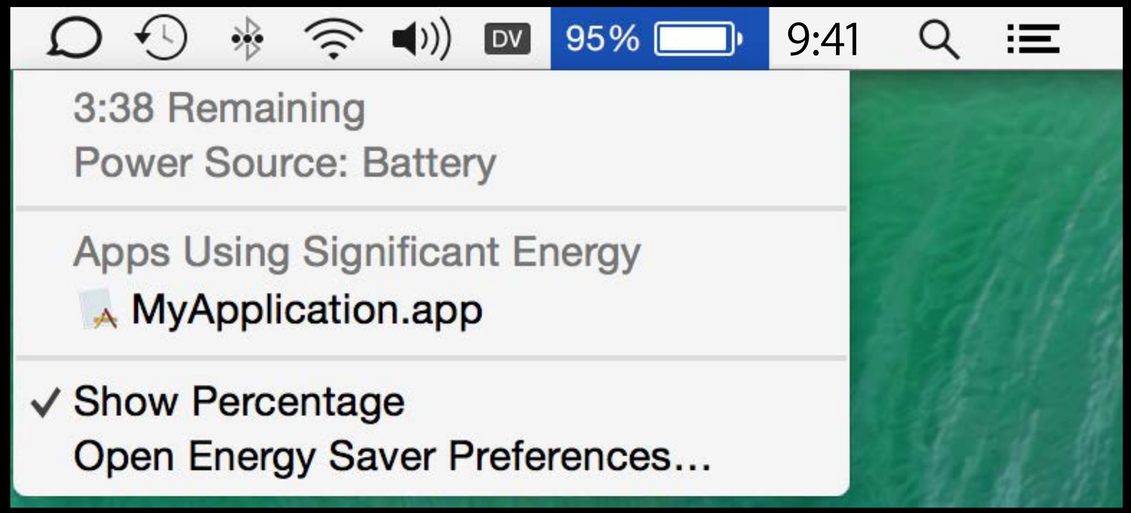
Writing Energy Efficient Code

Part 1

Session 710

Anthony Chivetta
OS X Performance & Power




DV
95%
9:41
3:38 Remaining
Power Source: Battery
Apps Using Significant Energy
MyApplication.app
Show Percentage
Open Energy Saver Preferences…


9:41 AM
100%
Battery Usage
BATTERY USAGE
Last 24 Hours
Last 7 Days
Safari 24%
Facebook 19%
Phone 14%
Mail
Background Activity 13%
Maps
Location 11%
Pandora
Audio 9%
Messages 6%
No Cell Coverage 4%
Shows proportion of battery used by each app when iPhone is not charging.

What You'll Learn

What You'll Learn

Power and energy concepts

What You'll Learn

Power and energy concepts

Improving your energy use

What You'll Learn

Power and energy concepts

Improving your energy use

- Do it never

What You'll Learn

Power and energy concepts

Improving your energy use

- Do it never
- Do it at a better time

What You'll Learn

Power and energy concepts

Improving your energy use

- Do it never
- Do it at a better time
- Do it more efficiently

What You'll Learn

Power and energy concepts

Improving your energy use

- Do it never
- Do it at a better time
- Do it more efficiently
- Do it less

What You'll Learn

Power and energy concepts

Improving your energy use

- Do it never
- Do it at a better time
- Do it more efficiently
- Do it less

Part 2

What You'll Learn

Power and energy concepts

Improving your energy use

- Do it never
- Do it at a better time
- Do it more efficiently
- Do it less

Part 2

- Networking

What You'll Learn

Power and energy concepts

Improving your energy use

- Do it never
- Do it at a better time
- Do it more efficiently
- Do it less

Part 2

- Networking
- Location

What You'll Learn

Power and energy concepts

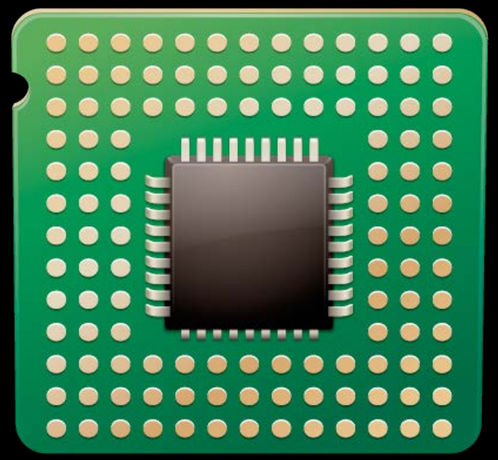
Improving your energy use

- Do it never
- Do it at a better time
- Do it more efficiently
- Do it less

Part 2

- Networking
- Location
- Sleep/Wake

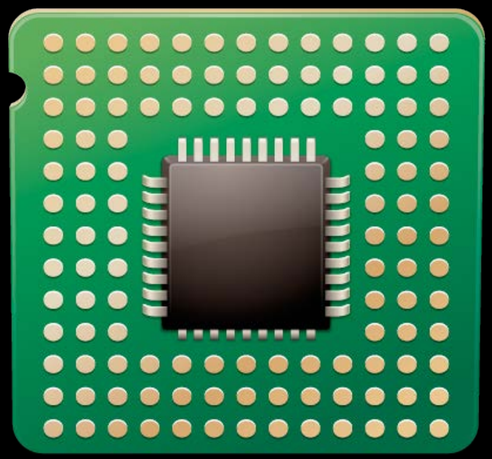
What Uses Energy?



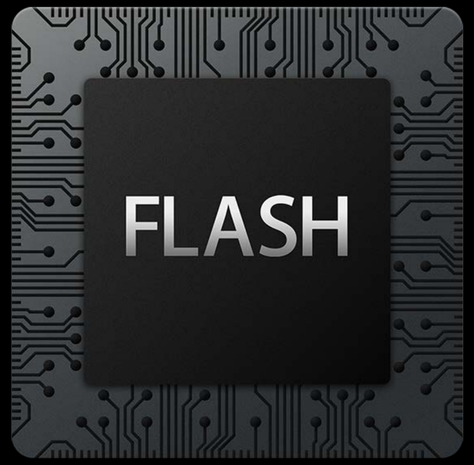
CPU

(not an exhaustive list)

What Uses Energy?



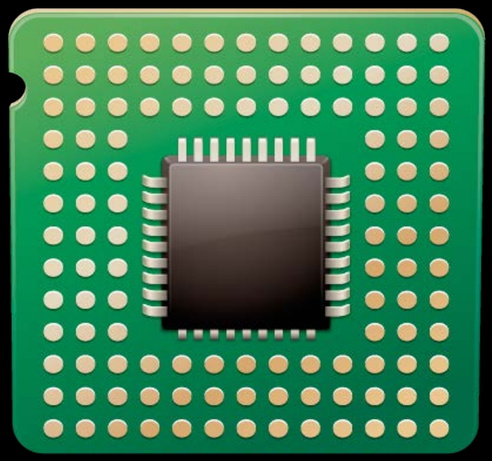
CPU



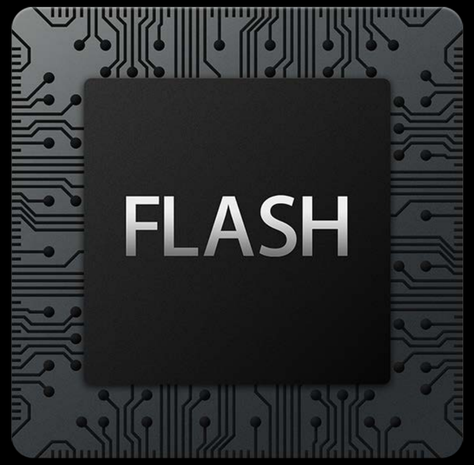
Storage

(not an exhaustive list)

What Uses Energy?



CPU

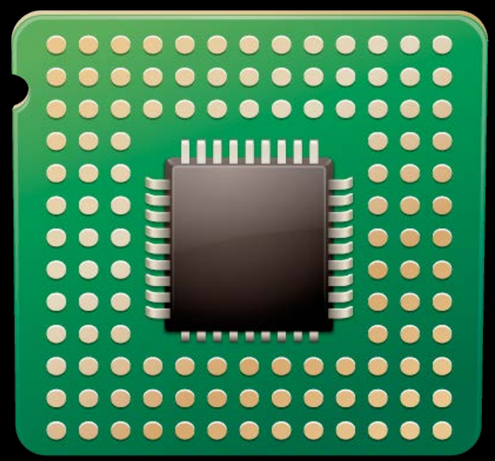


Storage

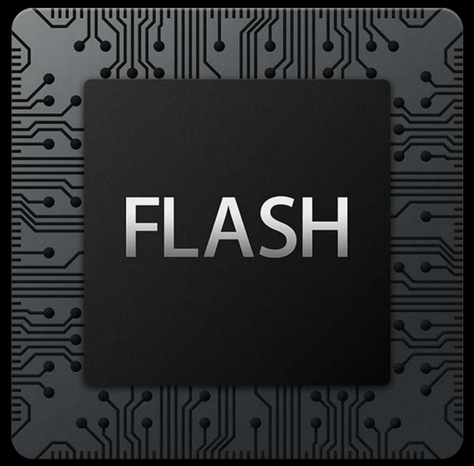
Networking

(not an exhaustive list)

What Uses Energy?



CPU



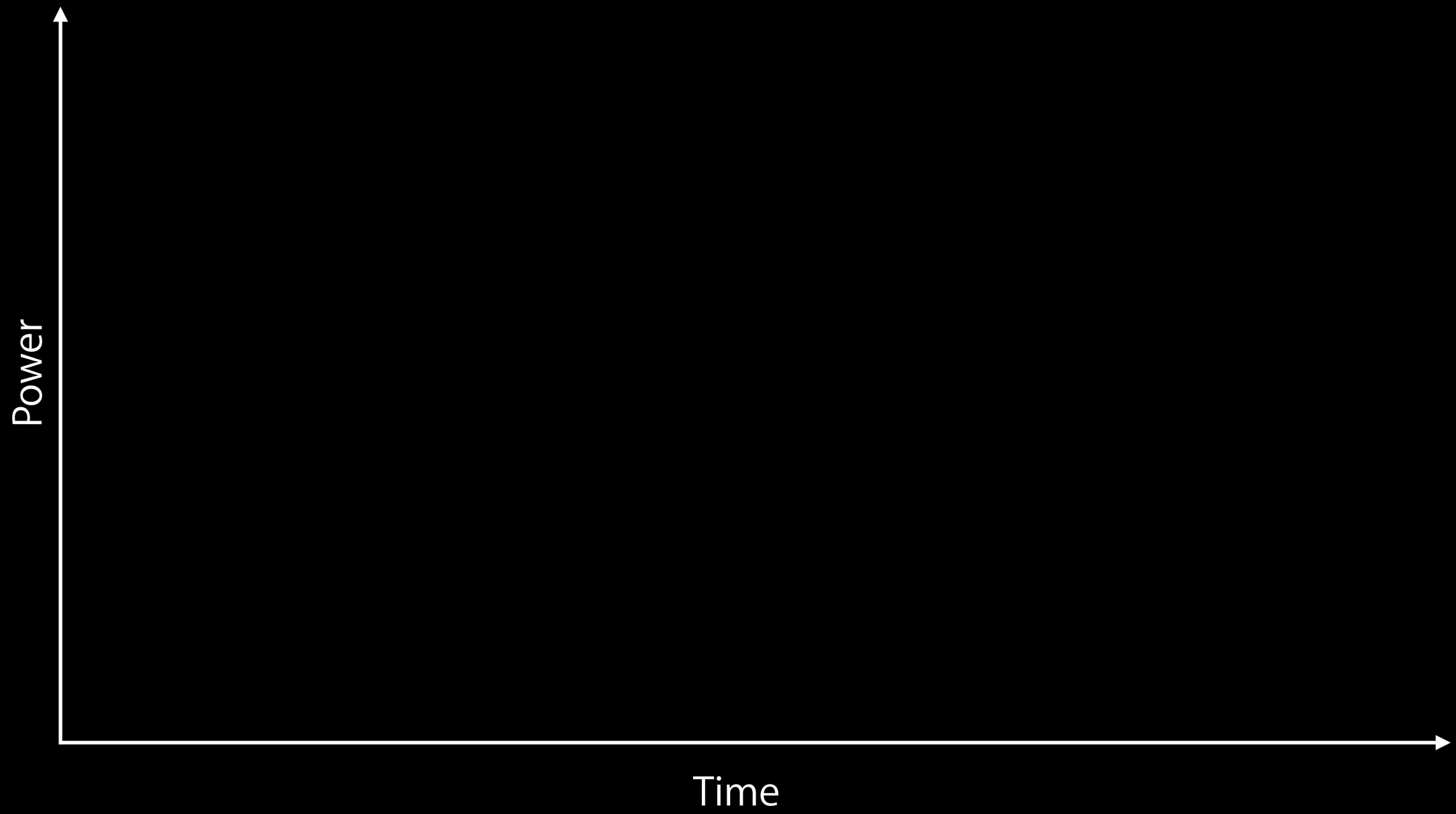
Storage

Networking

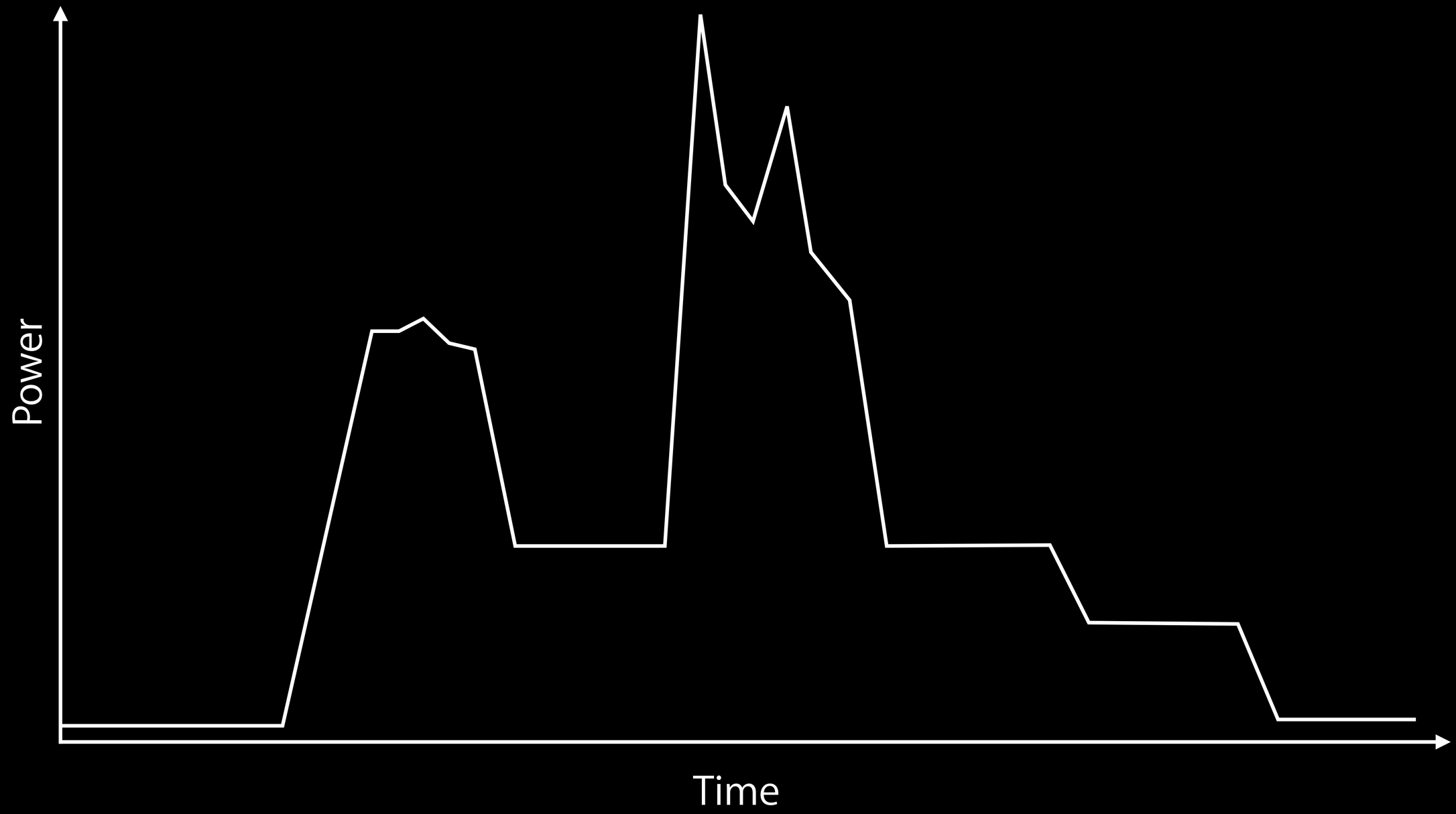
Graphics

(not an exhaustive list)

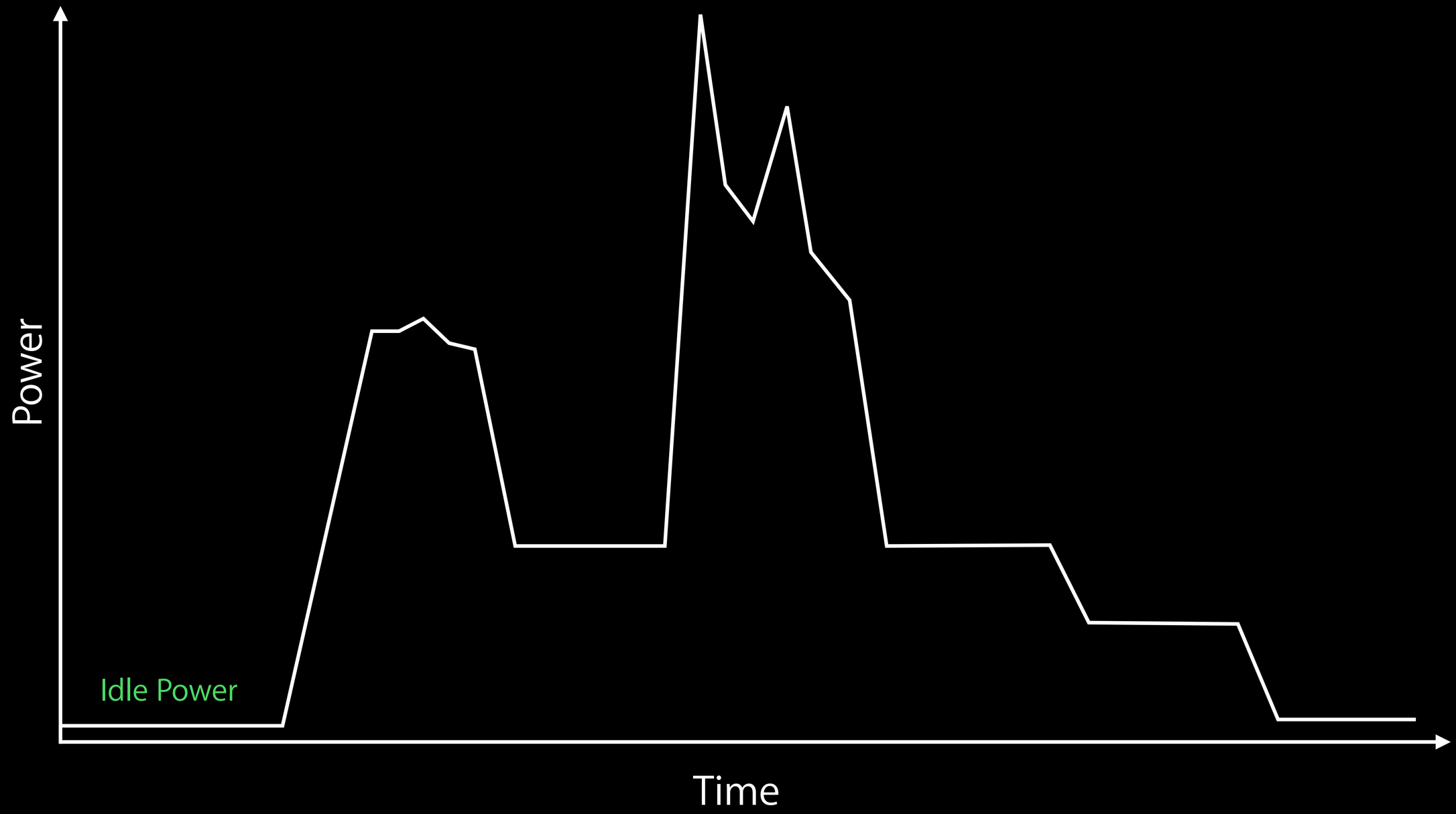
Fixed Costs



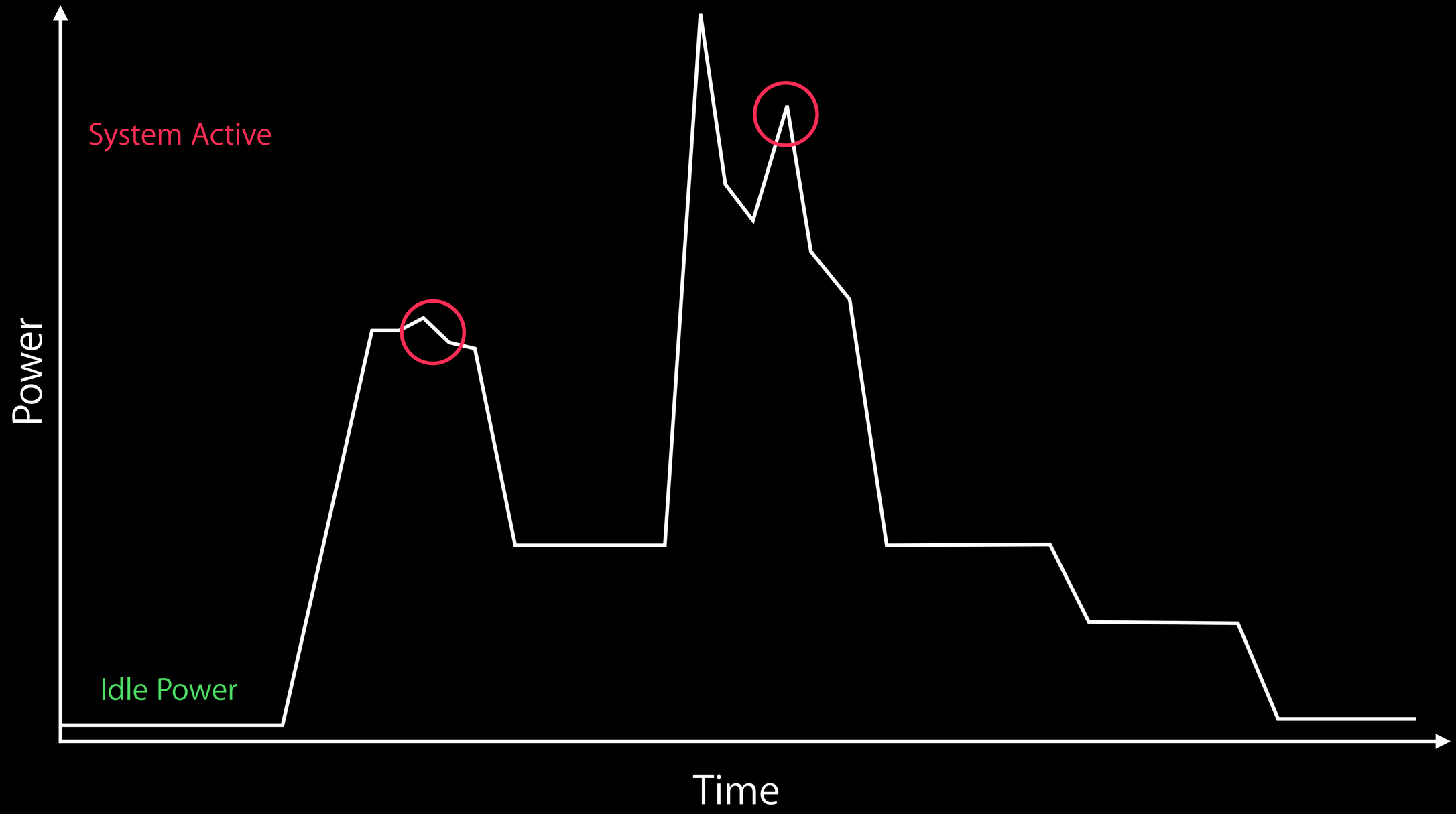
Fixed Costs



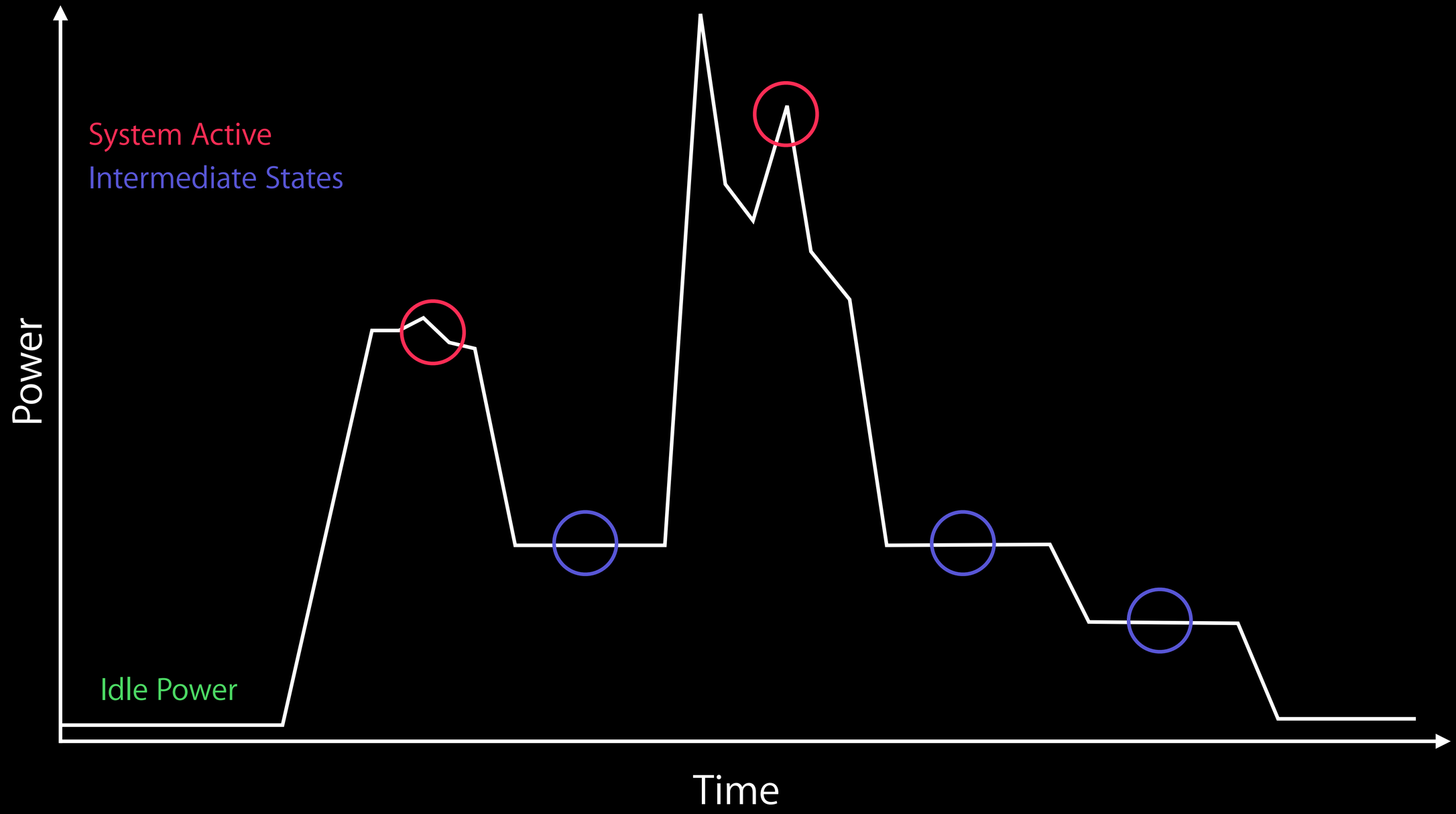
Fixed Costs



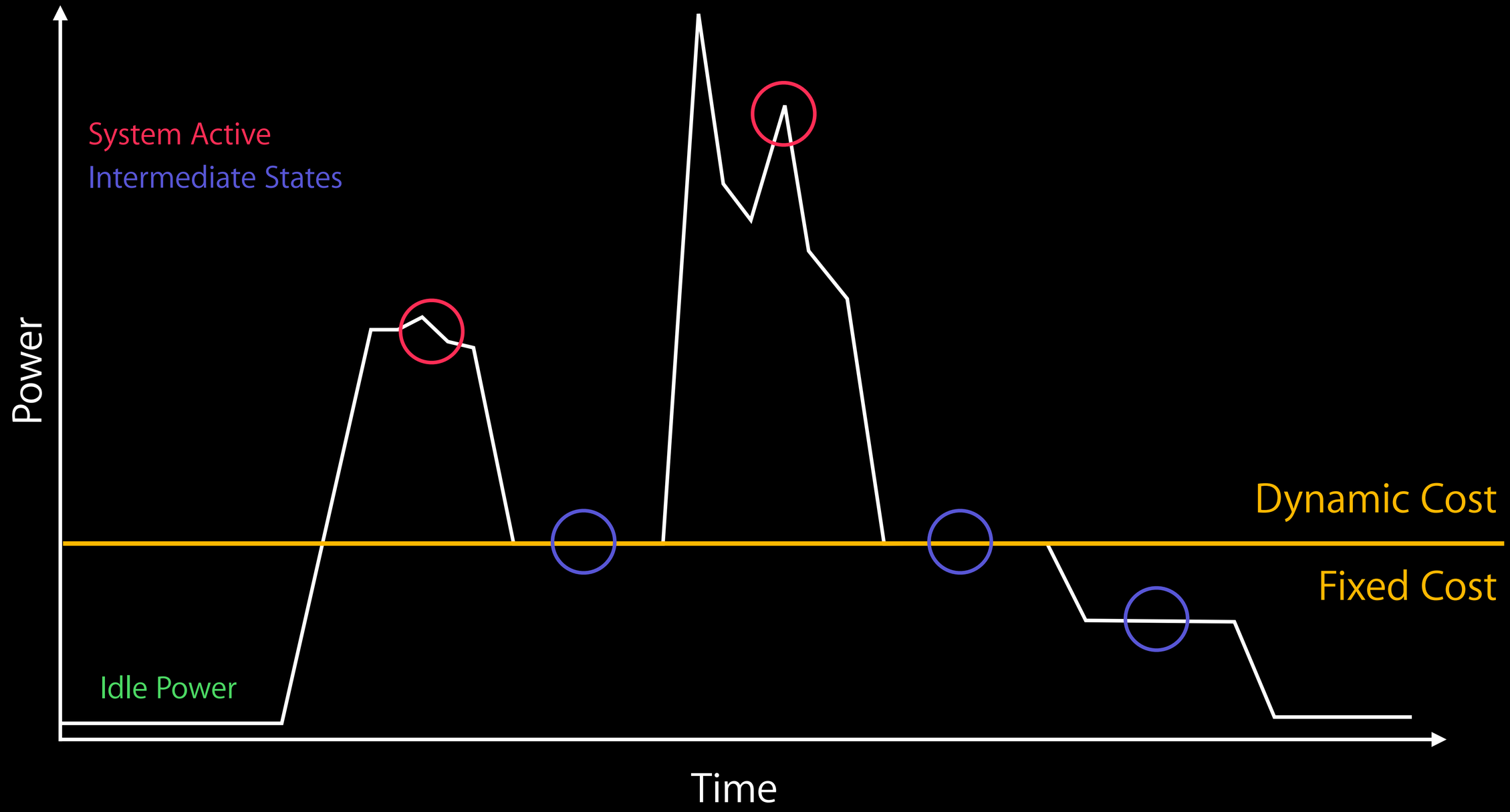
Fixed Costs



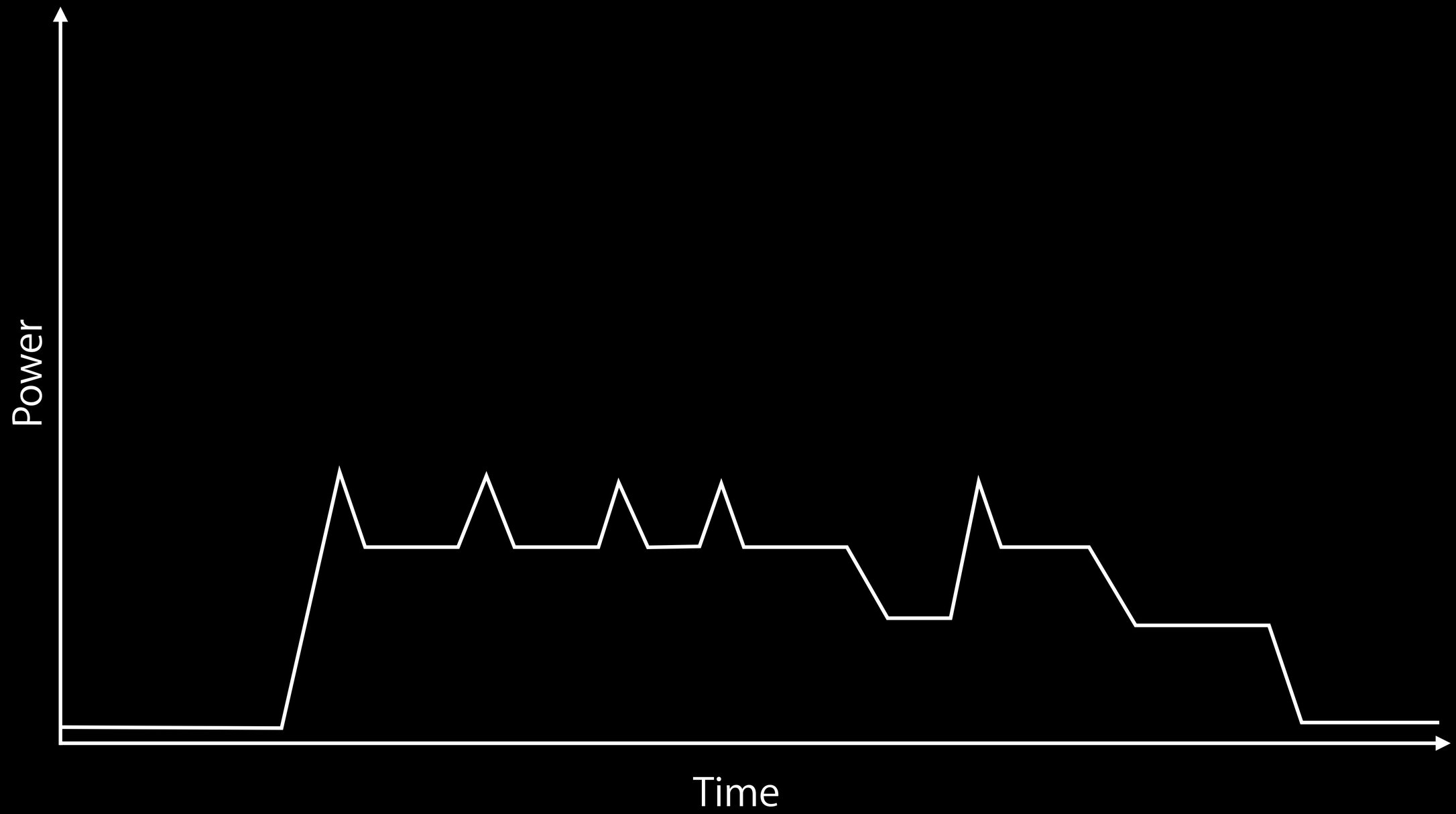
Fixed Costs



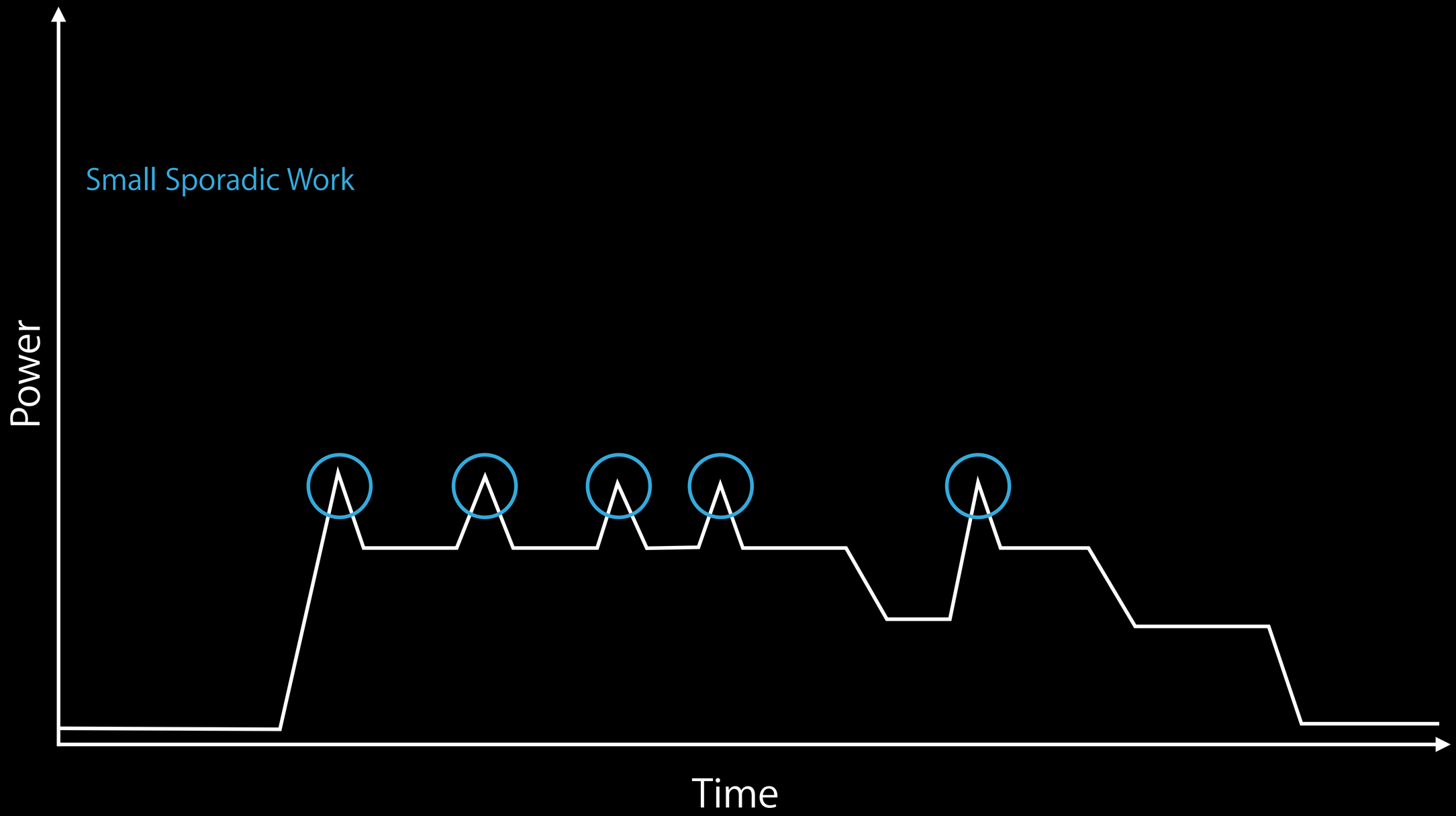
Fixed Costs


System Active
Intermediate States
Power
Dynamic Cost
Fixed Cost
Idle Power
Time

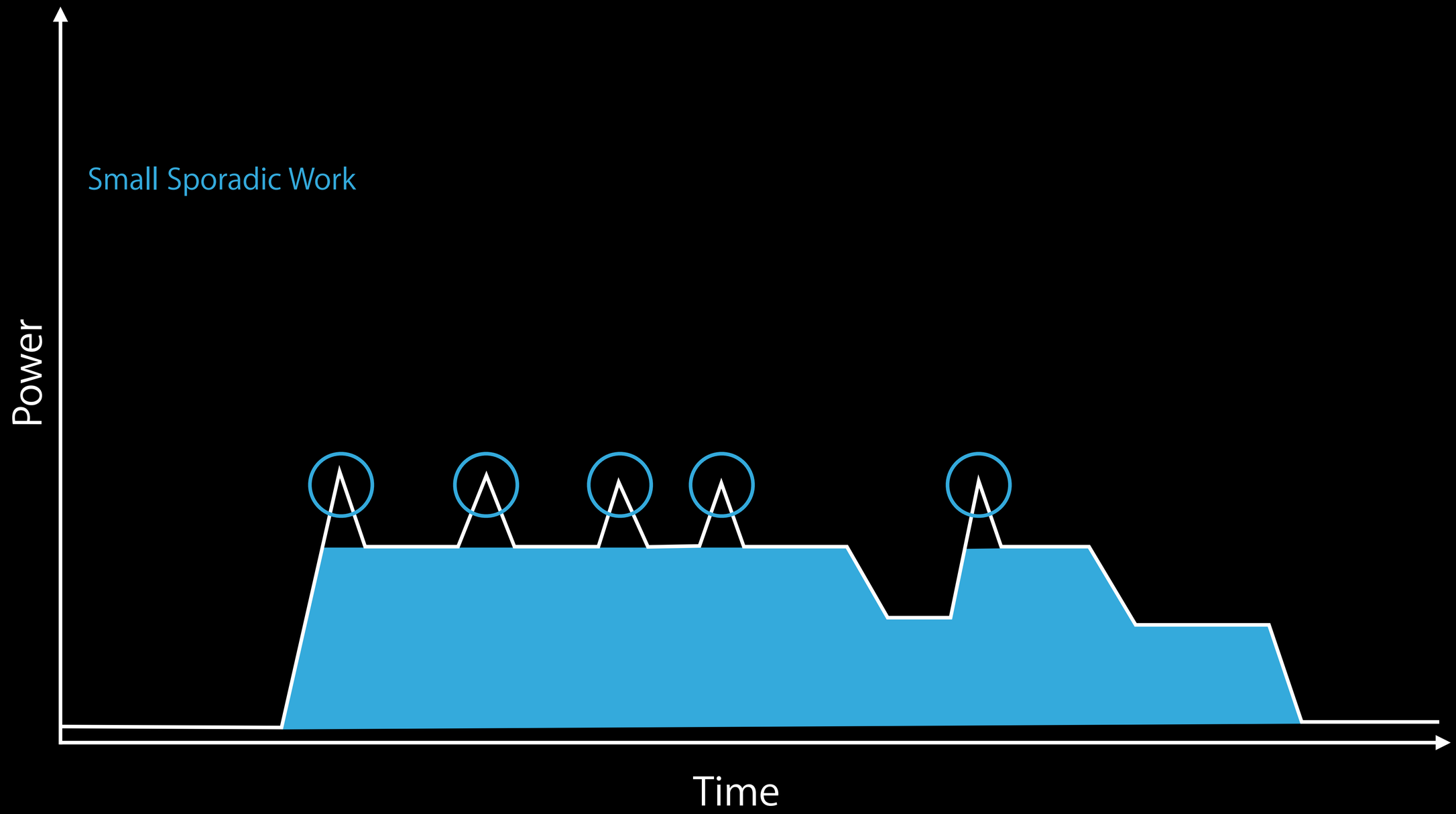
Fixed Costs



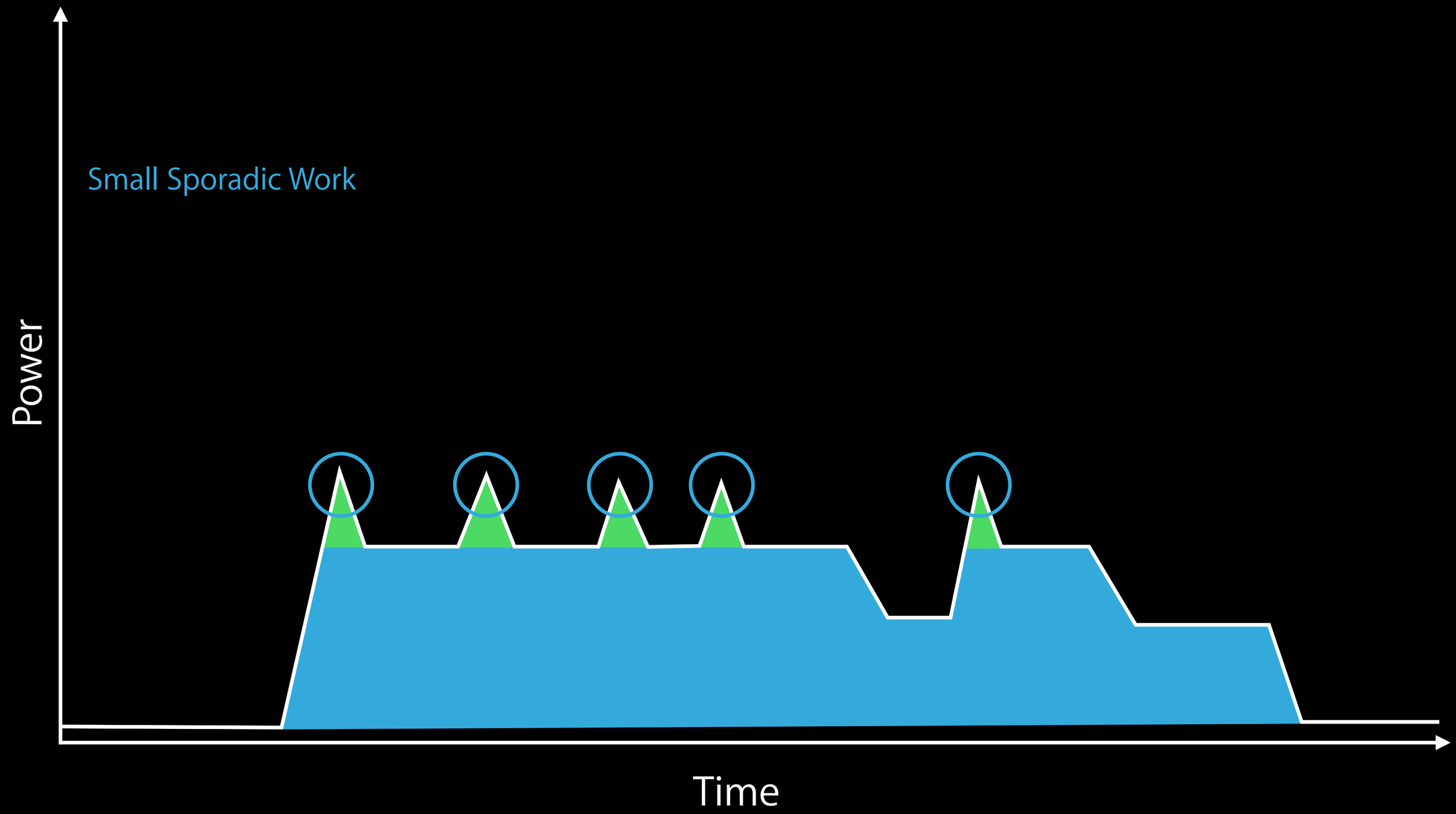
Fixed Costs



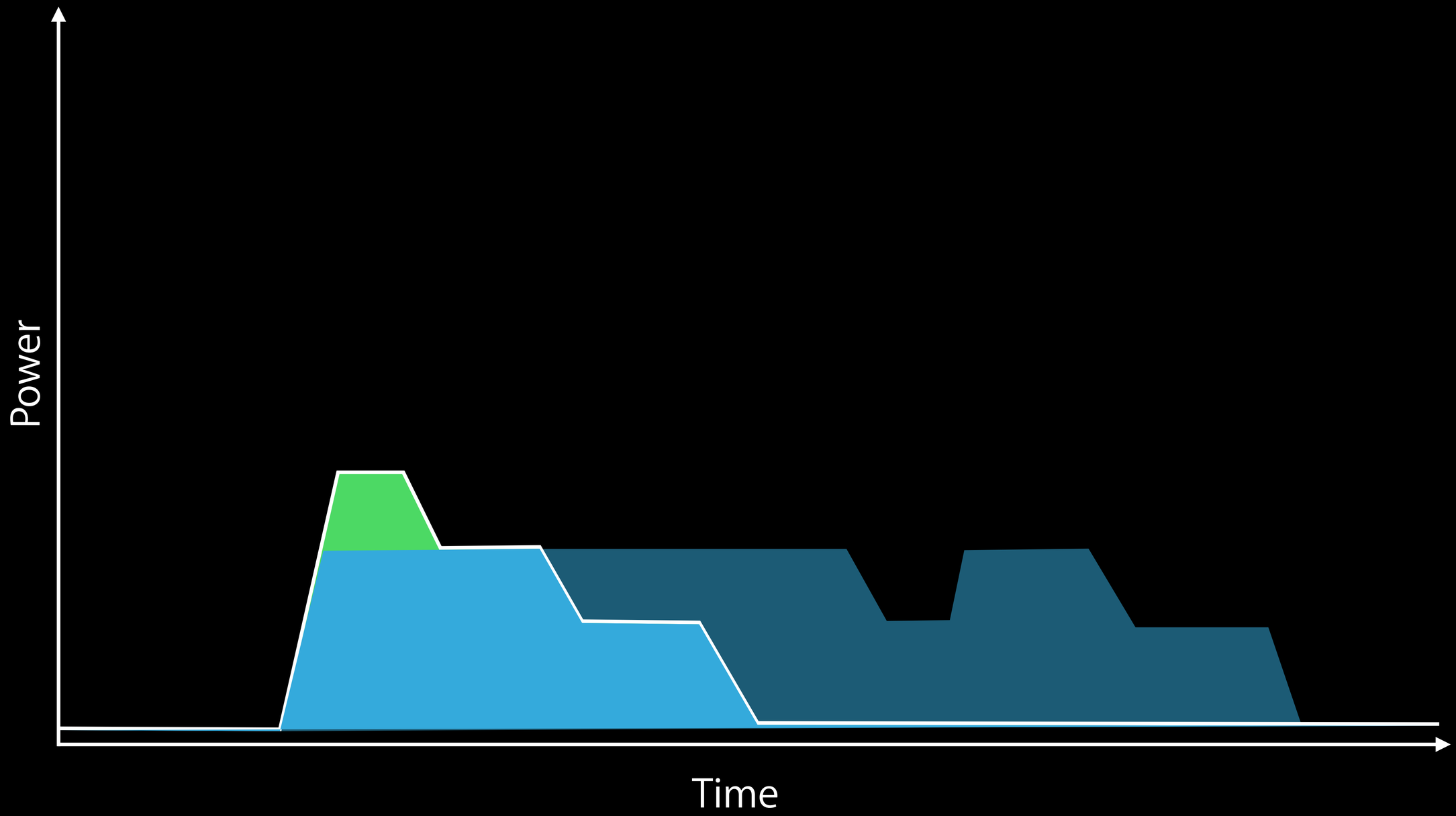
Fixed Costs



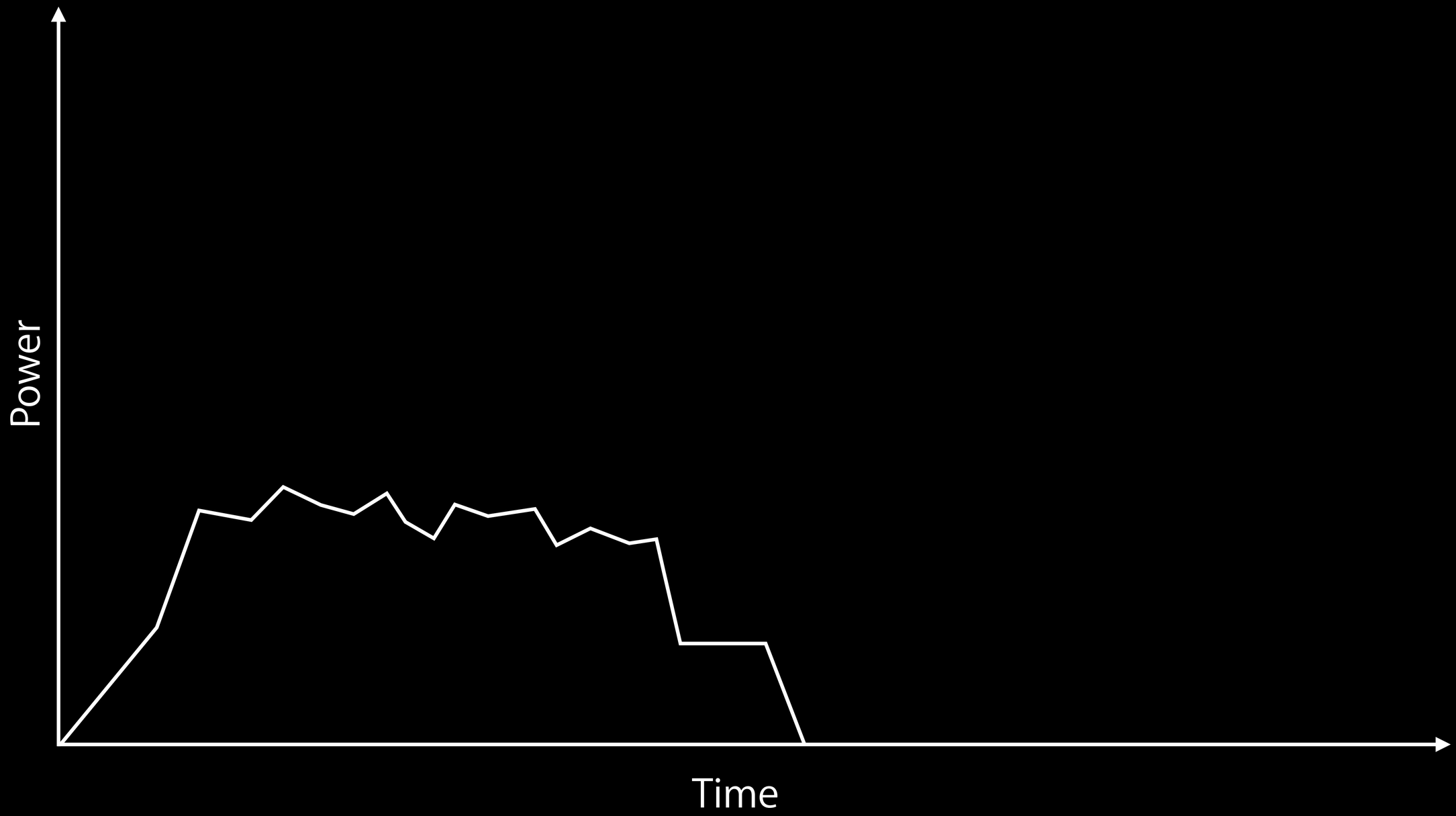
Fixed Costs



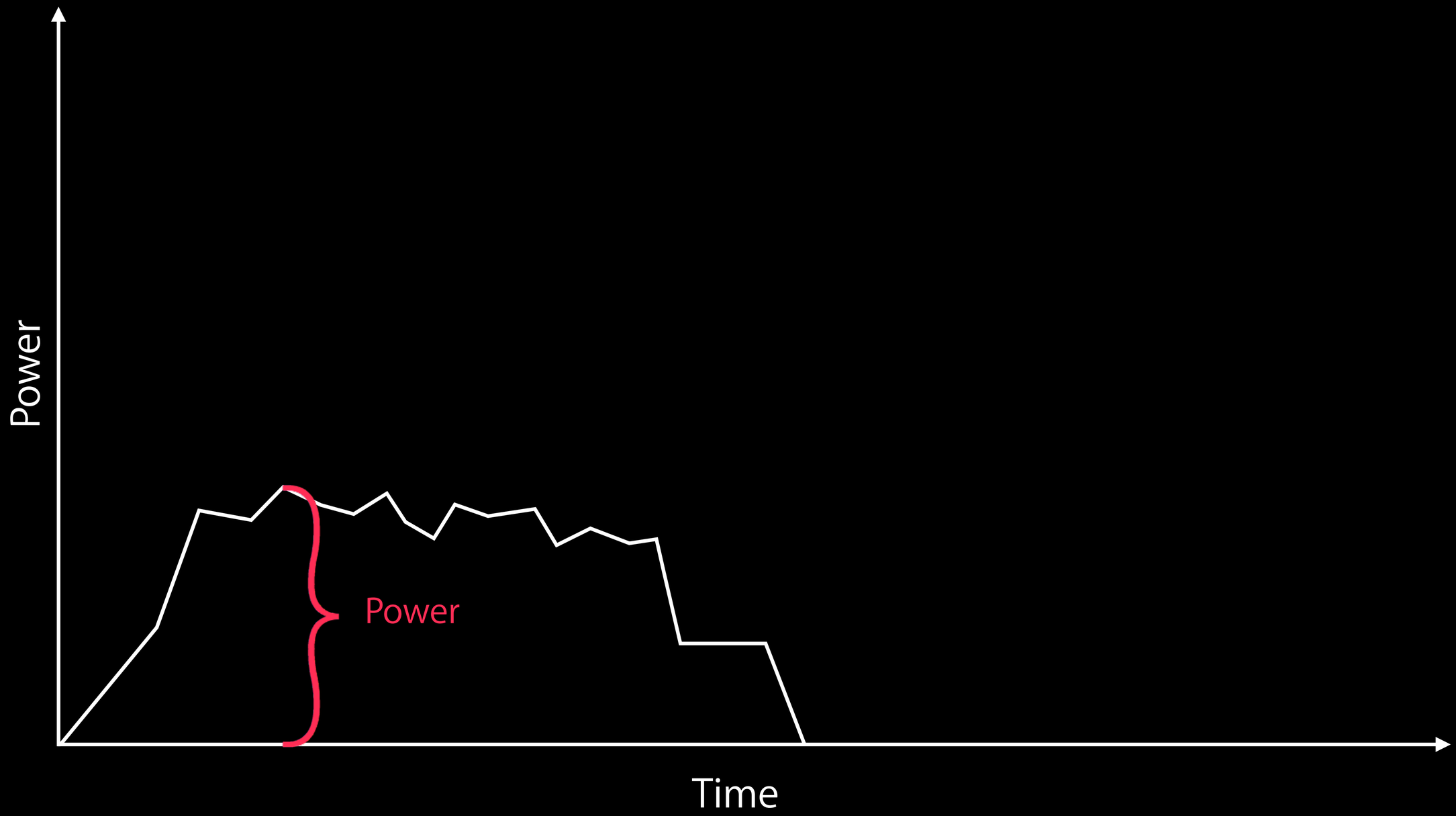
Fixed Costs



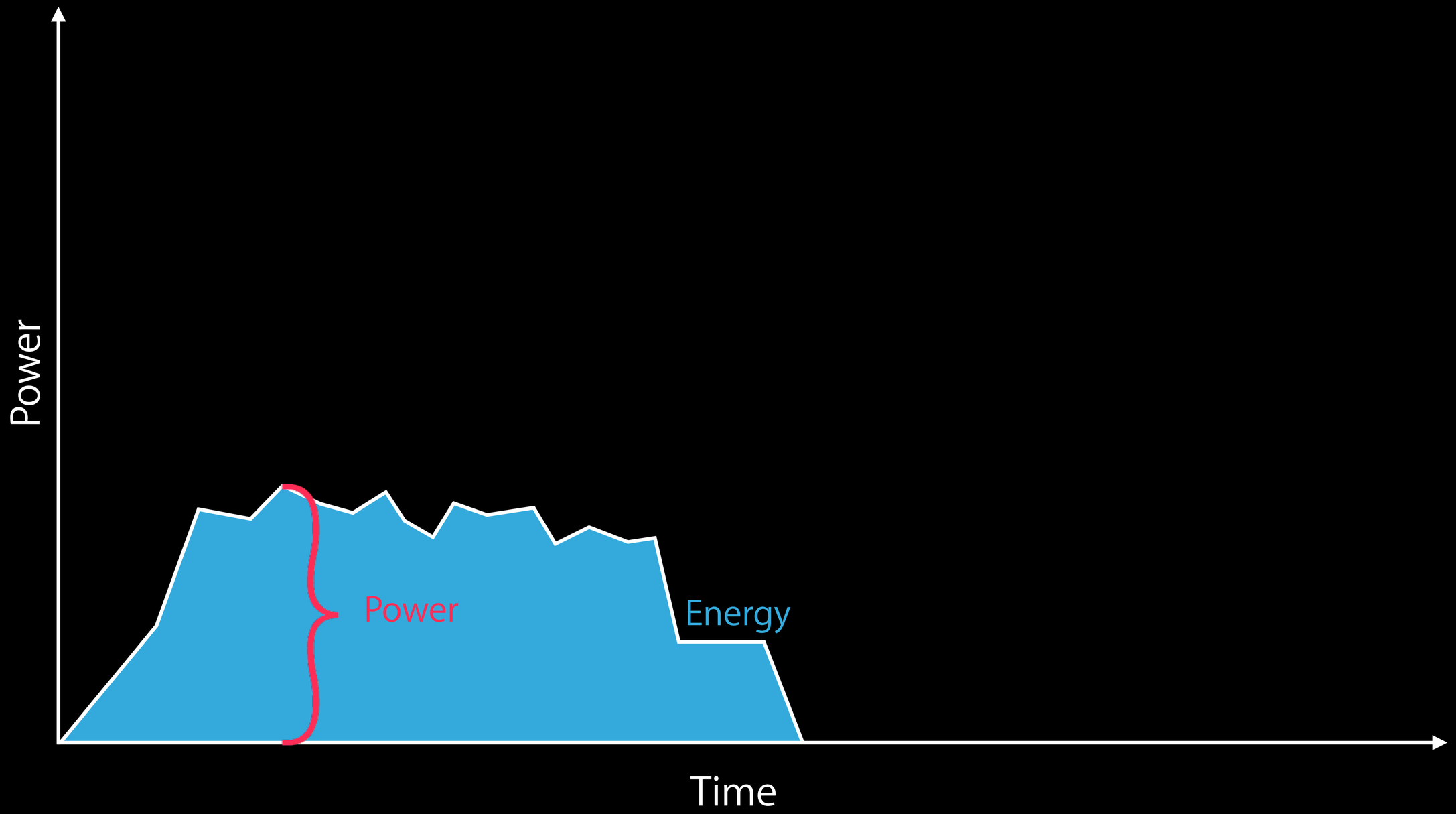
Energy and Power



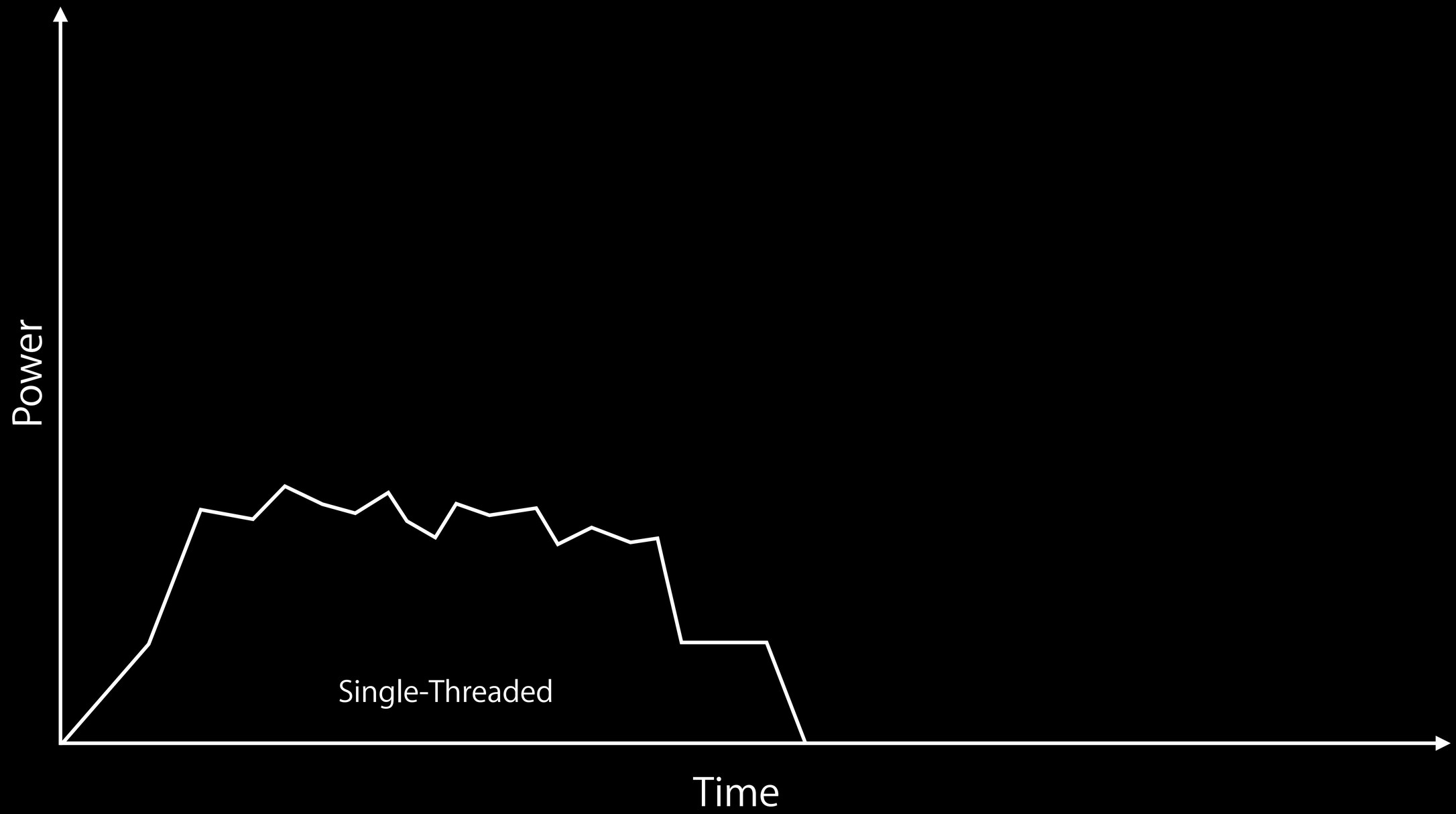
Energy and Power



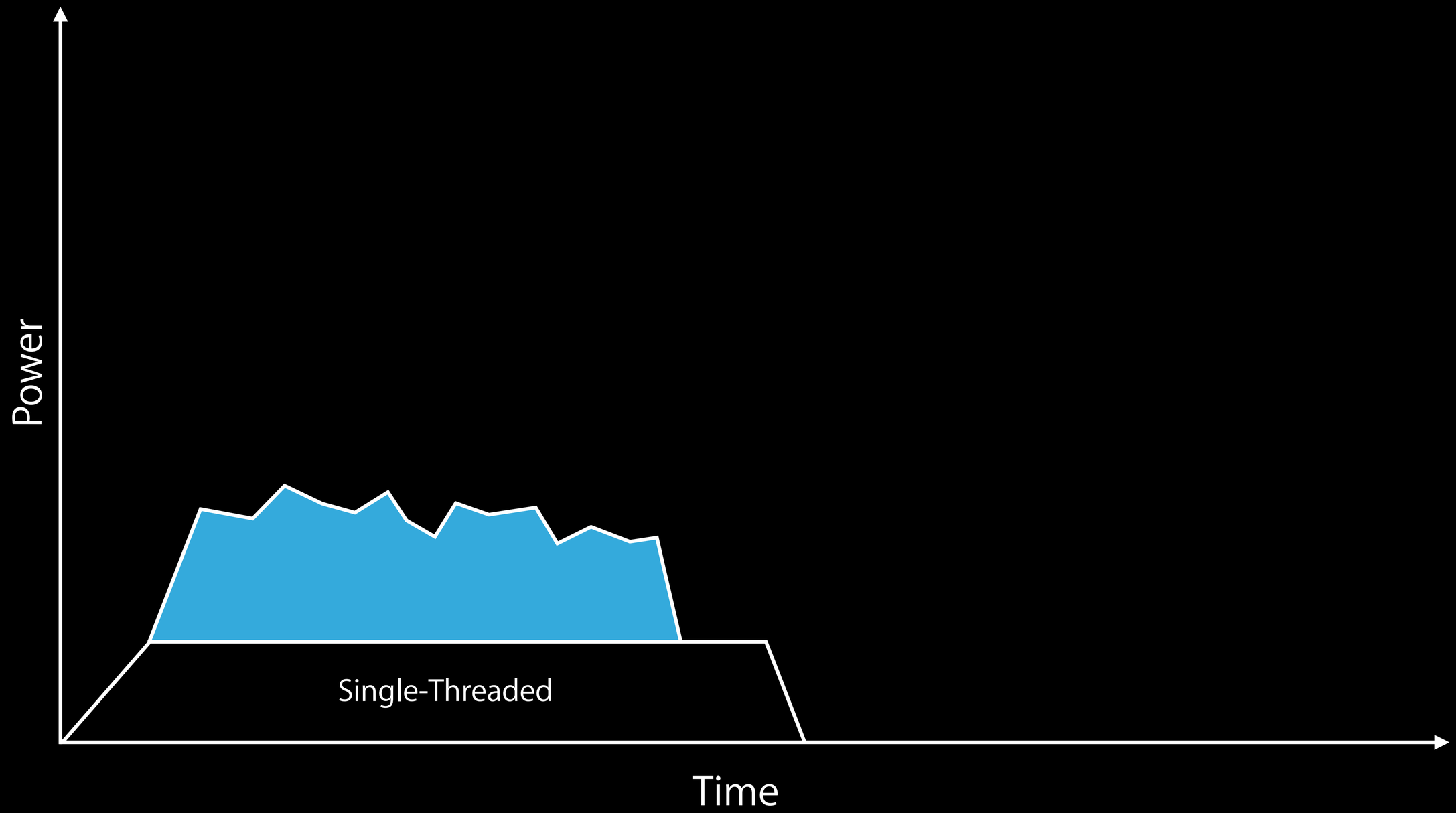
Energy and Power



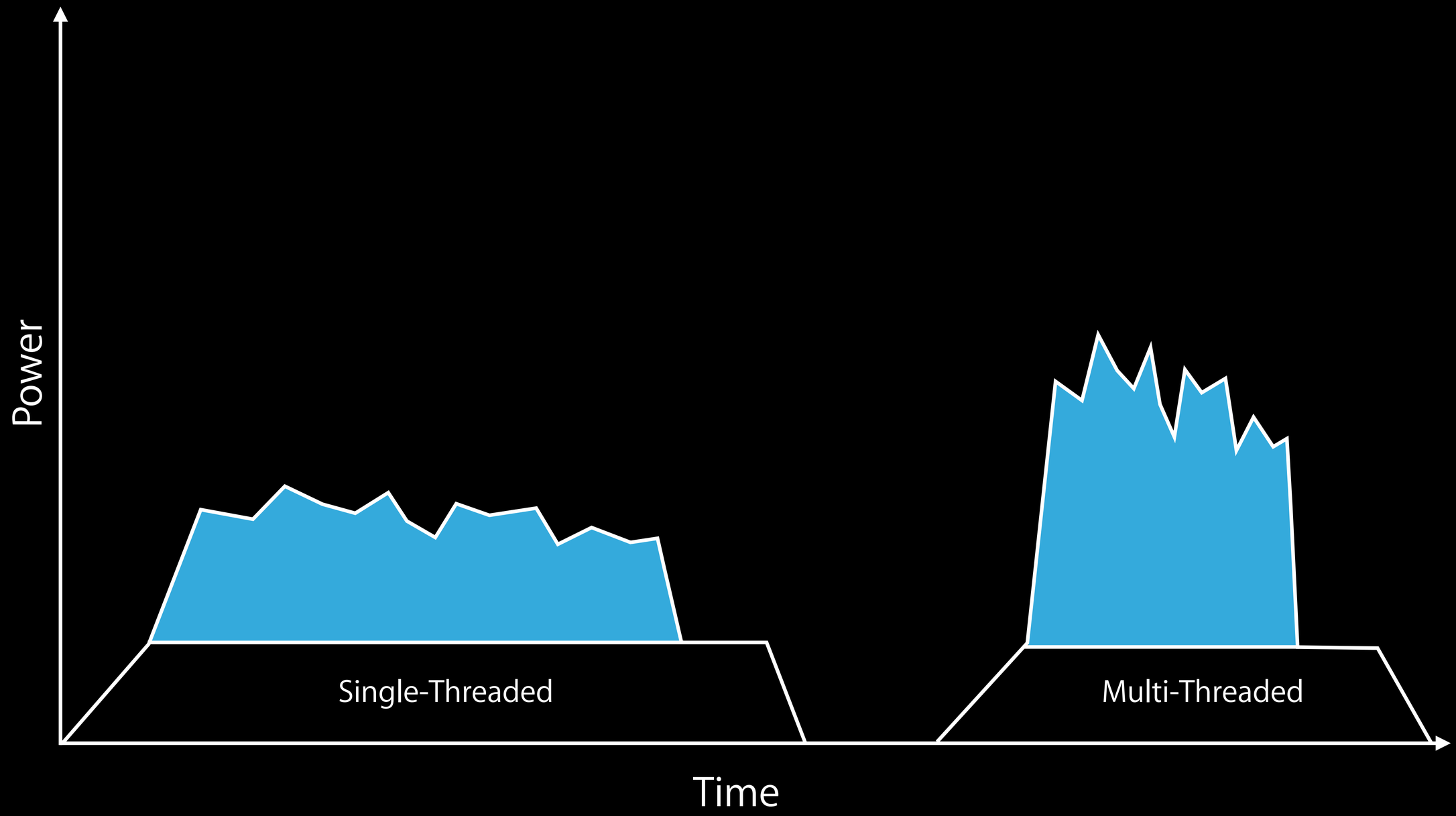
Trading Power for Energy



Trading Power for Energy



Trading Power for Energy



Power Fundamentals

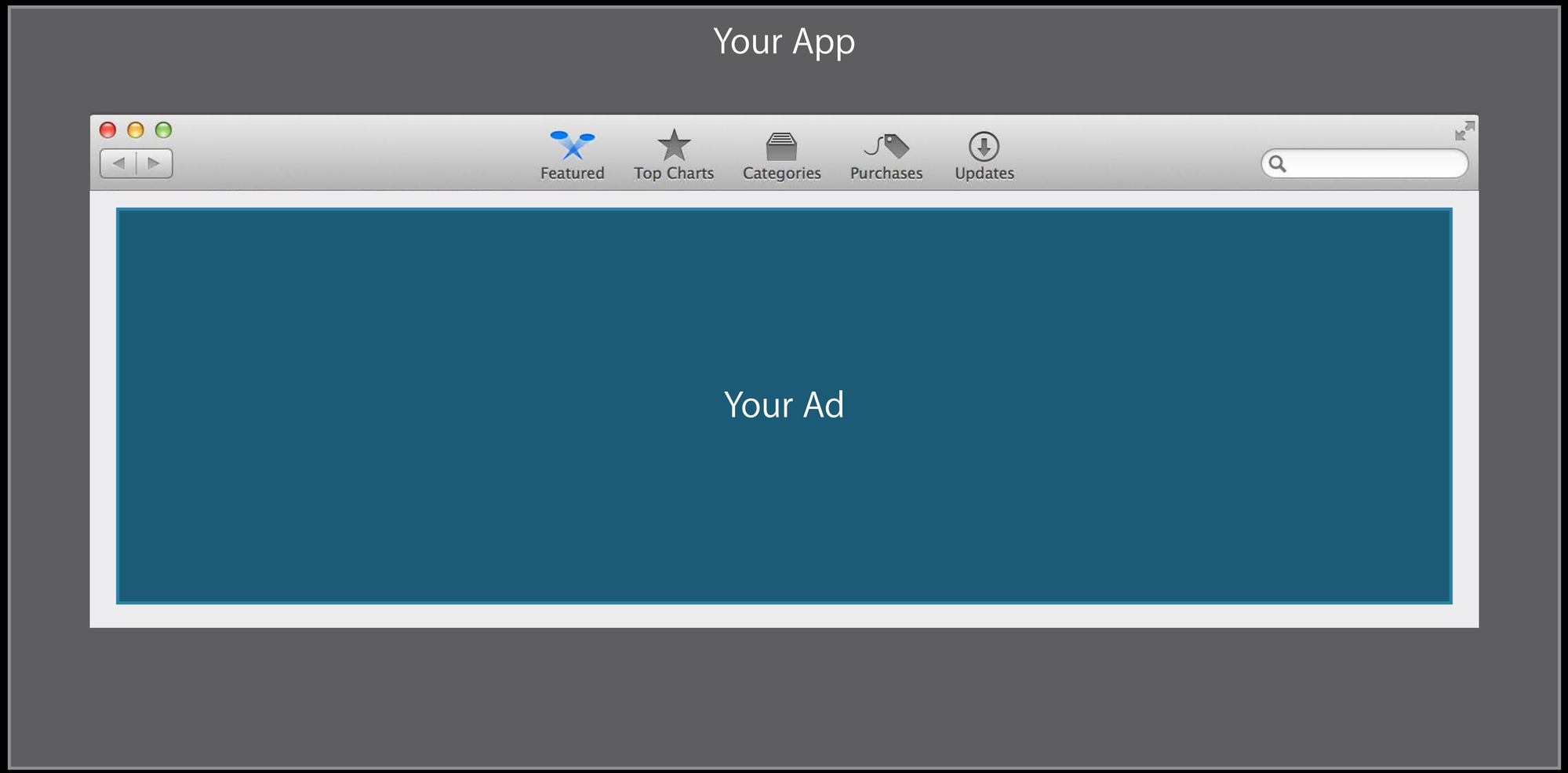
Work has a fixed cost

- For small workloads, this can dominate
- For intensive workloads, dynamic cost will dominate

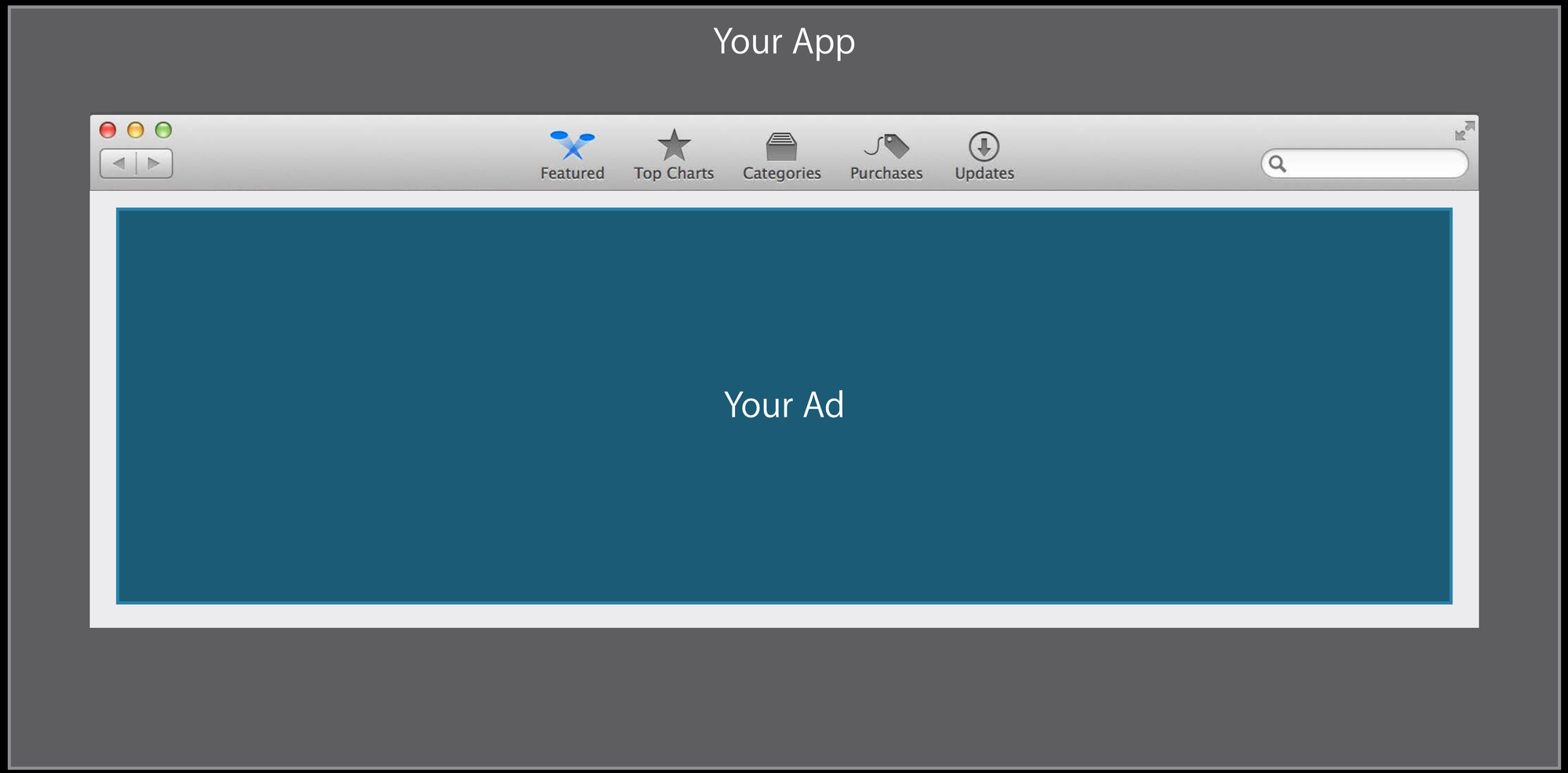
Better performance = less energy

Do It Never

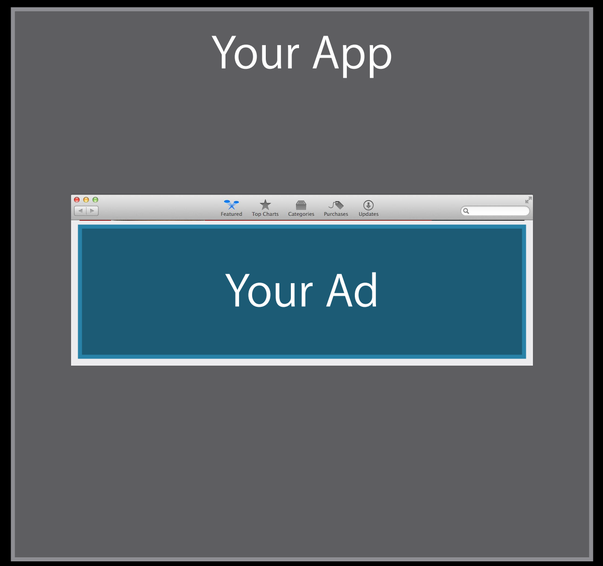
Avoid Unneeded Work



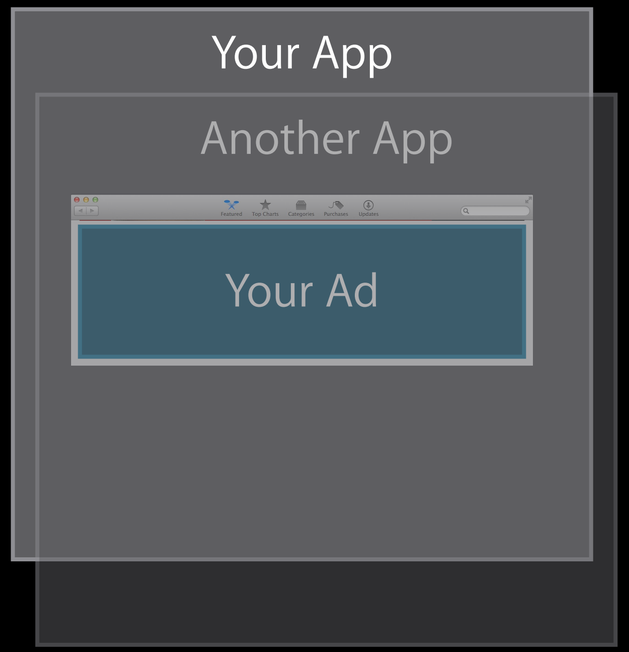
Avoid Unneeded Work



Avoid Unneeded Work



Avoid Unneeded Work



Active App Transitions

iOS

```
- (void)applicationDidResignActive: (UIApplication *)application {
  // Pause animations and UI updates
}

- (void)applicationDidBecomeActive: (UIApplication *)application {
  // Resume animations and UI updates
}
```

Or, listen for the `UIApplicationWillResignActiveNotification` notification.

Active App Transitions

OS X

```
- (void)applicationDidResignActive:(NSNotification *)aNotification {
  // Pause animations and UI updates
}

- (void)applicationDidBecomeActive:(NSApplication *)aNotification {
  // Resume animations and UI updates
}
```

Or, listen for the `NSApplicationDidResignActiveNotification` notification.

Occlusion Notifications

Occlusion Notifications

Occlusion notifications indicate visibility of windows or applications

Occlusion Notifications

Occlusion notifications indicate visibility of windows or applications

For applications, implement delegate method

```
- (void)applicationDidChangeOcclusionState:(NSNotification *)notification
```

- Or check

```
if ([NSApp occlusionState] & NSApplicationOcclusionStateVisible)
```

Occlusion Notifications

Occlusion notifications indicate visibility of windows or applications

For applications, implement delegate method

```
- (void)applicationDidChangeOcclusionState:(NSNotification *)notification
```

- Or check

```
if ([NSApp occlusionState] & NSApplicationOcclusionStateVisible)
```

For windows, implement delegate method

```
- (void)windowDidChangeOcclusionState:(NSNotification *)notification
```

- Or check

```
if ([window occlusionState] & NSWindowOcclusionStateVisible)
```

App Nap

App Nap

App Nap reduces an inactive app's energy use

App Nap

App Nap reduces an inactive app's energy use

App Nap relies on heuristics

App Nap

App Nap reduces an inactive app's energy use

App Nap relies on heuristics

You are the authoritative source for what work is important

App Nap

App Nap reduces an inactive app's energy use

App Nap relies on heuristics

You are the authoritative source for what work is important

In a well-behaved app, App Nap should never be in effect during work

App Nap

App Nap reduces an inactive app's energy use

App Nap relies on heuristics

You are the authoritative source for what work is important

In a well-behaved app, App Nap should never be in effect during work

```
-NSProcessInfo (void)performActivityWithOptions:(NSActivityOptions)options
    reason:(NSString *)reason
    usingBlock:(void (^)(void))block
```

Avoiding Unnecessary Work

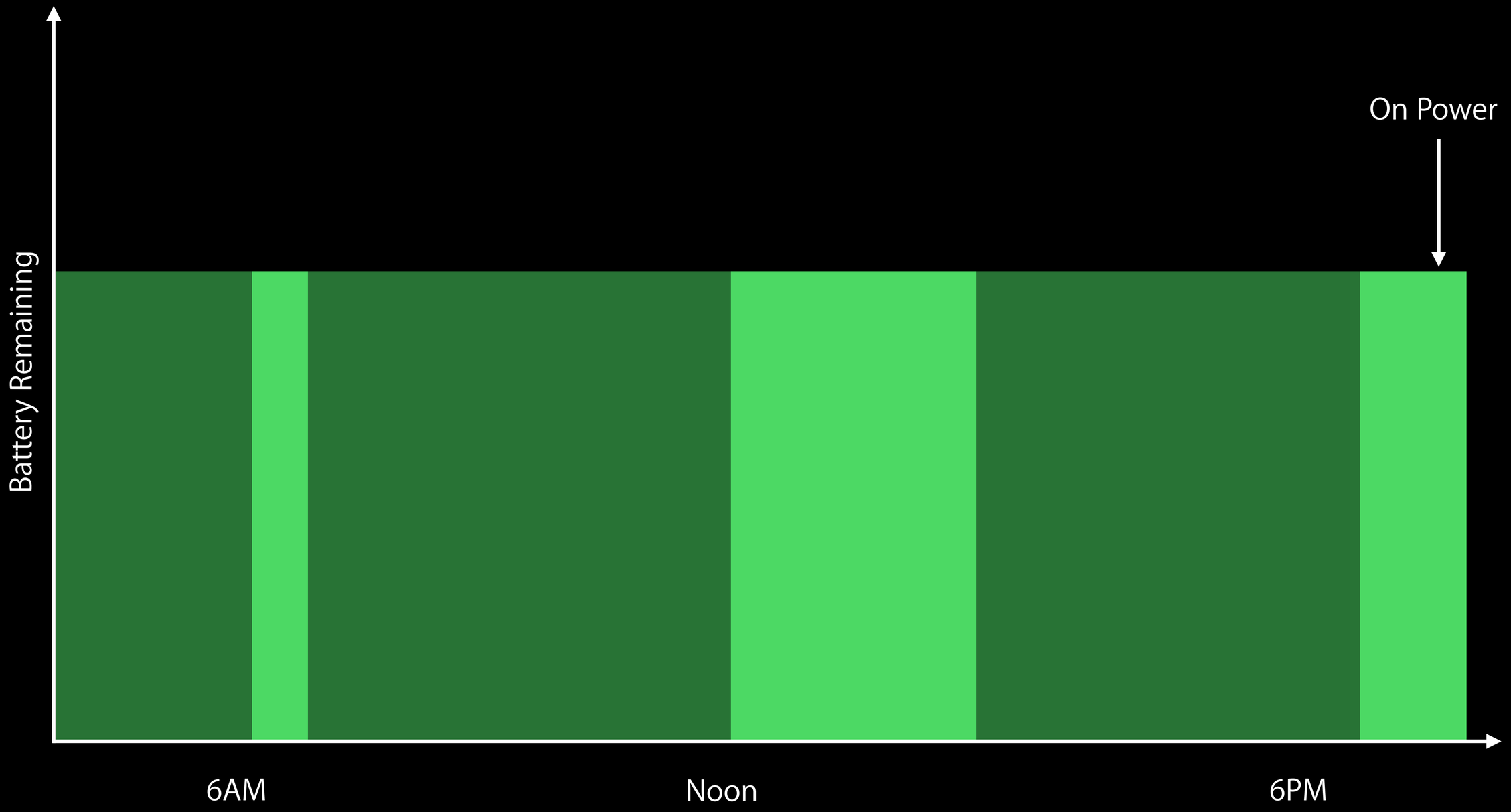
Monitor app state to know when work isn’t visible

Avoid updating UI until the user can see the results

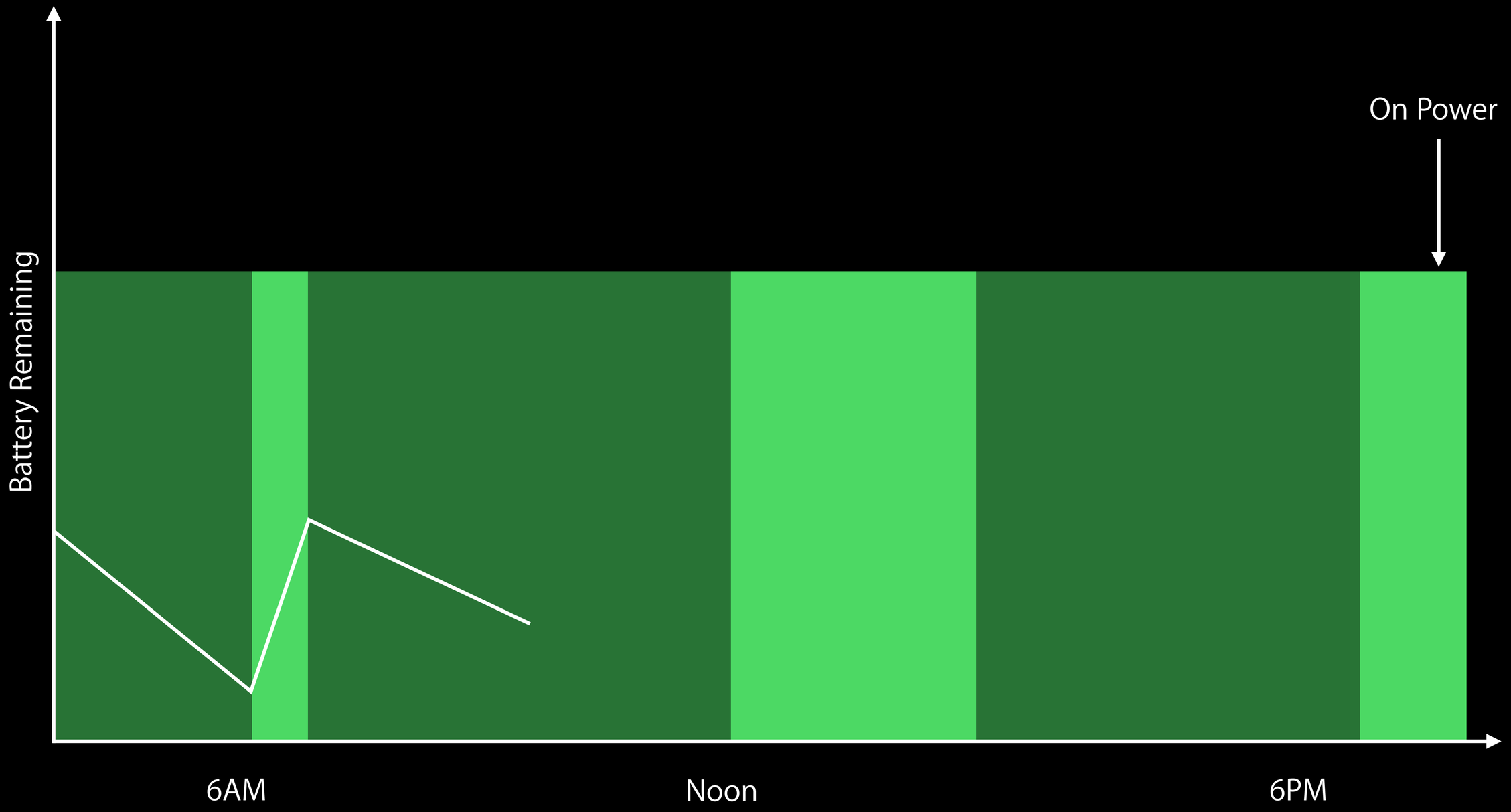
Nap yourself when not in use, so App Nap doesn’t have to

Do It at a Better Time

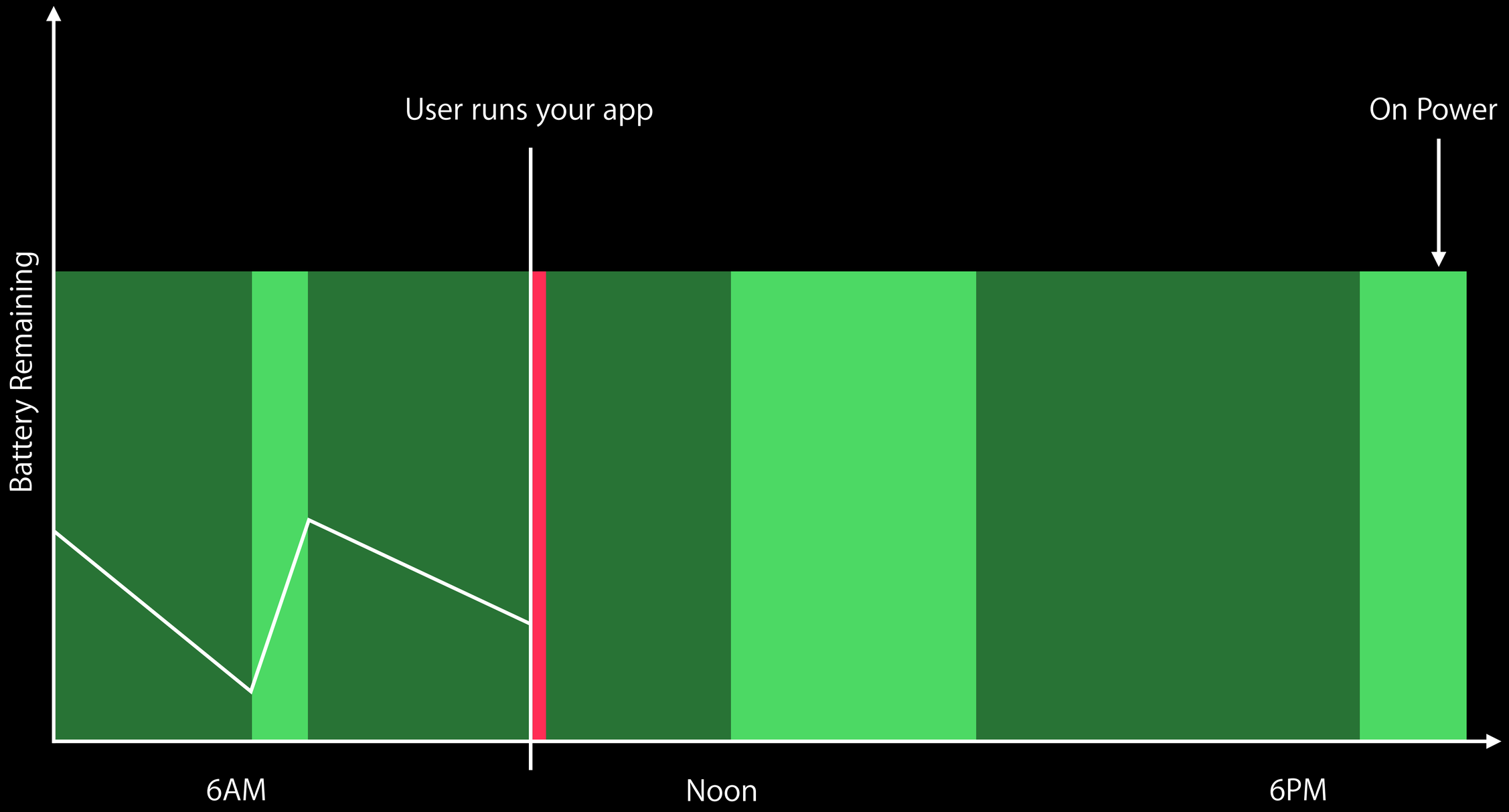
Do It at a Better Time



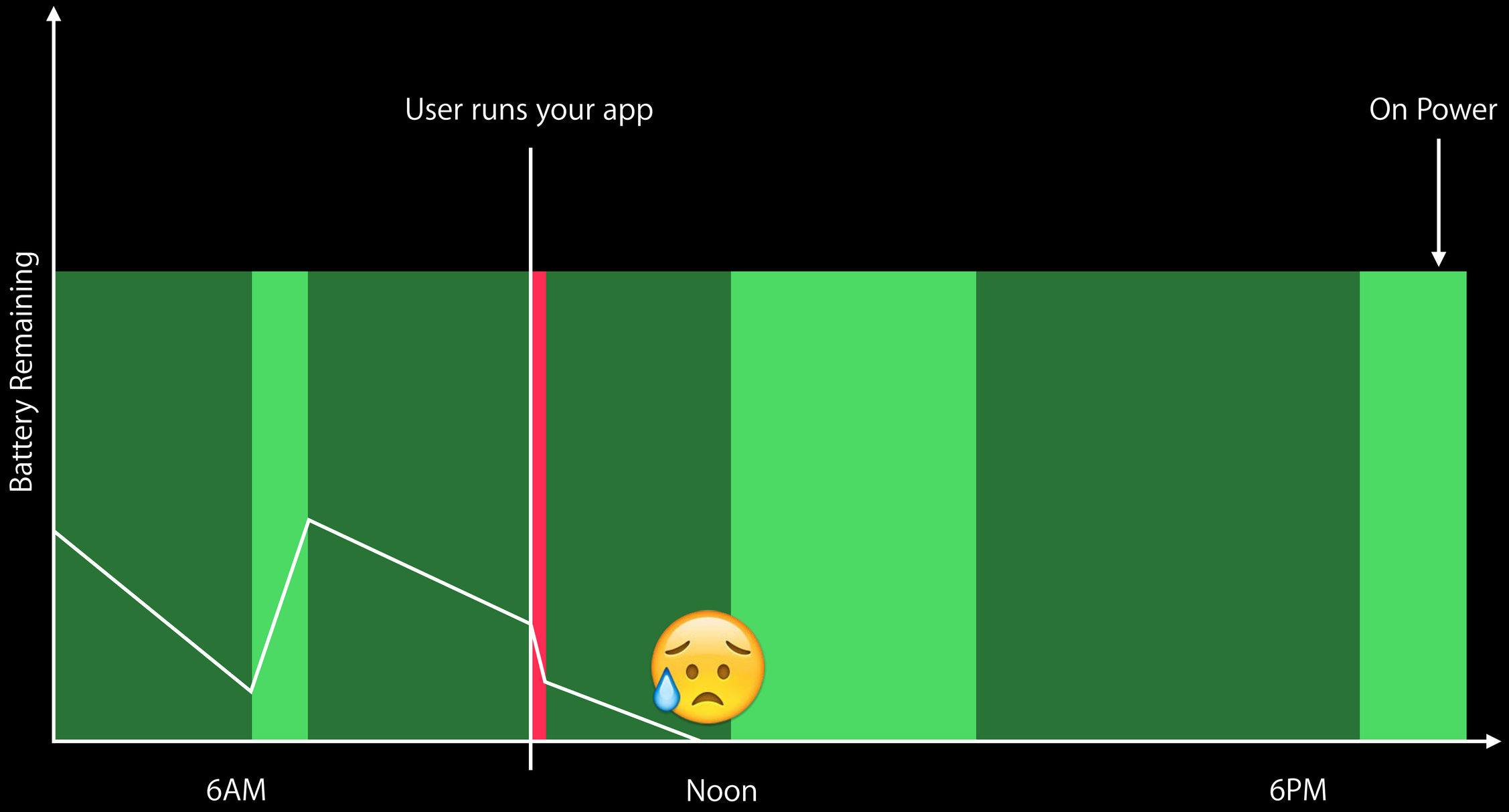
Do It at a Better Time



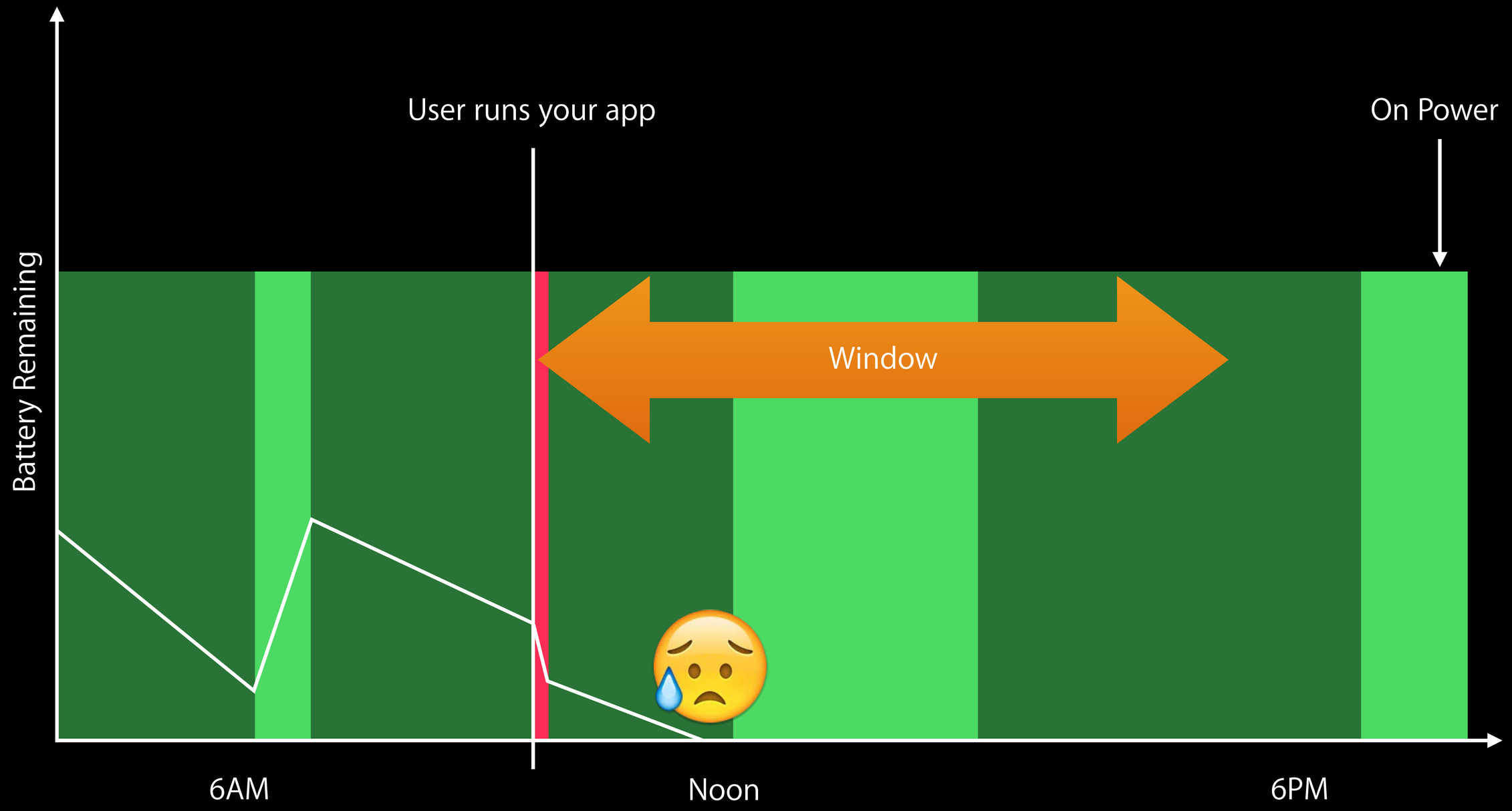
Do It at a Better Time



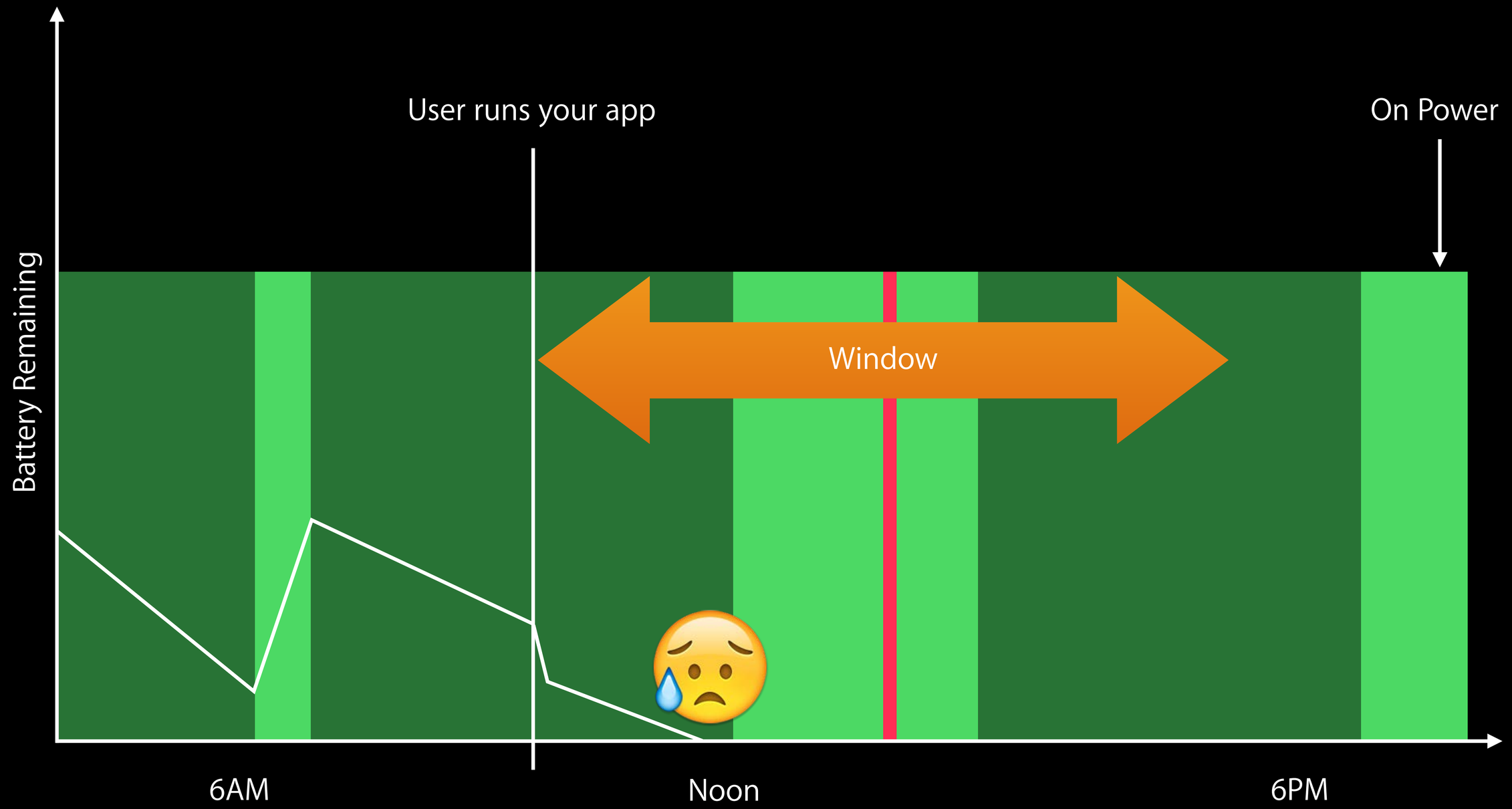
Do It at a Better Time



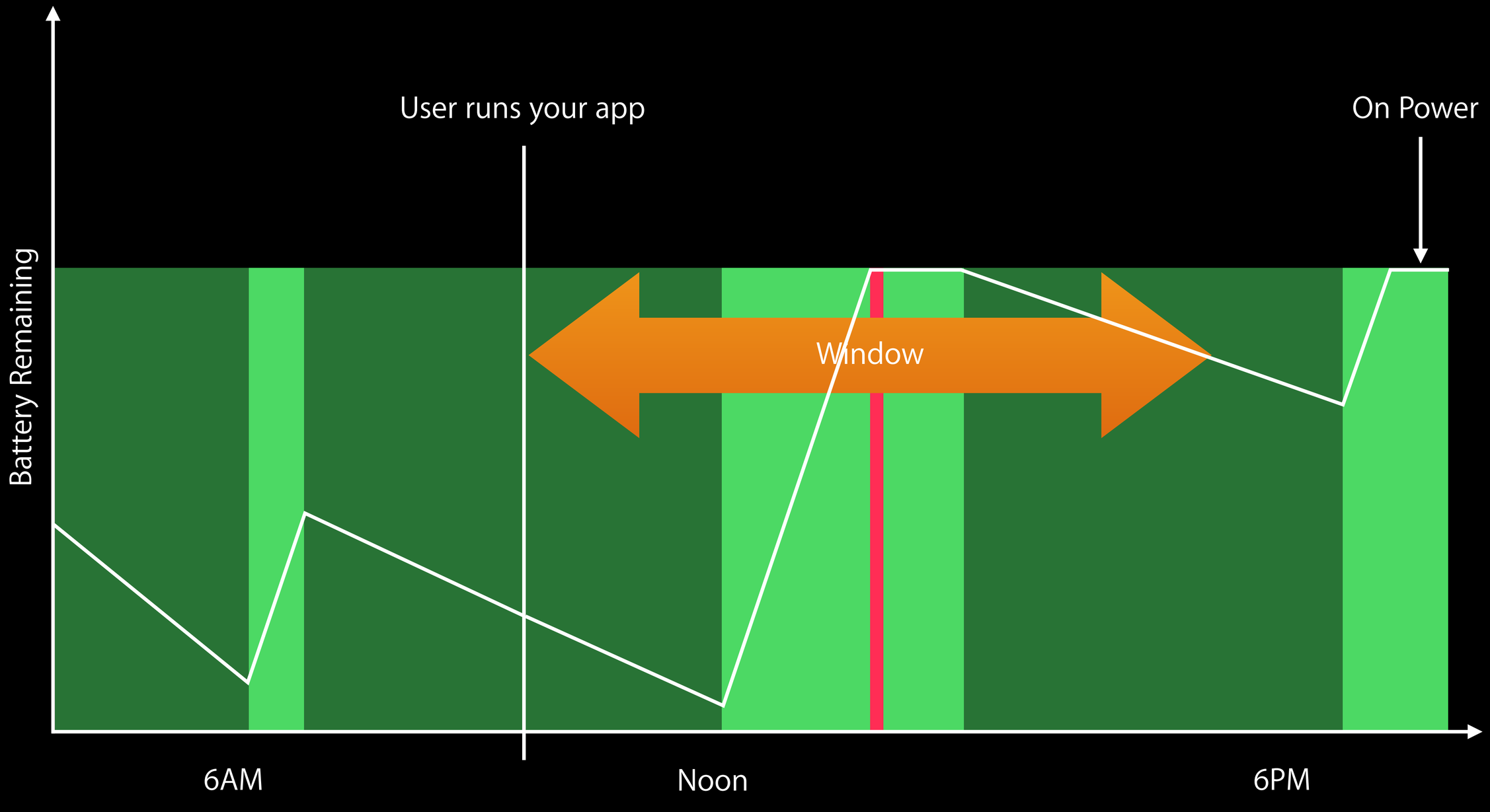
Do It at a Better Time



Do It at a Better Time



Do It at a Better Time



NSBackgroundActivityScheduler

NSBackgroundActivityScheduler

New API in OS X Yosemite

NSBackgroundActivityScheduler

New API in OS X Yosemite

Allows scheduling arbitrary tasks for a good time in the future

NSBackgroundActivityScheduler

New API in OS X Yosemite

Allows scheduling arbitrary tasks for a good time in the future

Supports repeating or non-repeating activities

NSBackgroundActivityScheduler

New API in OS X Yosemite

Allows scheduling arbitrary tasks for a good time in the future

Supports repeating or non-repeating activities

Can be used to schedule

- Periodic content fetch
- Update install
- Garbage collection and data maintenance tasks
- Automatic saves or backups

Creating a Scheduler

```
activity = [[NSBackgroundActivityScheduler alloc]
    initWithIdentifier:@“com.apple.sample-app.MyActivity”];
```

Each activity must have an identifier

Identifier should be in reverse-DNS style

Name should be unique, but the same across app runs

- “com.example.MyApp.updatecheck”

Specifying Scheduling Properties

```
// Activity will fire in the next 10 minutes
activity.tolerance = 10 * 60;
```

Specifying Scheduling Properties

```
// Activity will fire in the next 10 minutes
activity.tolerance = 10 * 60;

// Activity will fire between 15 and 45 minutes from now
activity.interval = 30 * 60;
activity.tolerance = 15 * 60;
```

Specifying Scheduling Properties

```
// Activity will fire in the next 10 minutes
activity.tolerance = 10 * 60;

// Activity will fire between 15 and 45 minutes from now
activity.interval = 30 * 60;
activity.tolerance = 15 * 60;

// Activity will fire once each hour
activity.repeats = YES
activity.interval = 60 * 60;
```

Scheduling Your Work

Scheduling Your Work

```
[activity
   scheduleWithBlock:^(NSBackgroundActivityCompletionHandler completion){
   // do the work


   completion(NSBackgroundActivityResultFinished);
}];
```

Scheduling Your Work

Deferring in-progress work

```
[activity
   scheduleWithBlock:^(NSBackgroundActivityCompletionHandler completion){
   for ( /* each item of work */) {
      if (activity.shouldDefer){
         completion(NSBackgroundActivityResultDeferred);
         return;
      }
      // do item of work
   }
   completion(NSBackgroundActivityResultFinished);
}];
```

Scheduling Your Work

Deferring in-progress work

```
[activity
   scheduleWithBlock:^(NSBackgroundActivityCompletionHandler completion){
   for ( /* each item of work */) {
      if (activity.shouldDefer){
         completion(NSBackgroundActivityResultDeferred);
         return;
      }
      // do item of work
   }
   completion(NSBackgroundActivityResultFinished);
}];
```

NSBackgroundActivityScheduler

You specify scheduling requirements for work

NSBackgroundActivityScheduler

You specify scheduling requirements for work

System select the best time to perform that work

NSBackgroundActivityScheduler

You specify scheduling requirements for work

System select the best time to perform that work

Support for repeating tasks without drift

NSBackgroundActivityScheduler

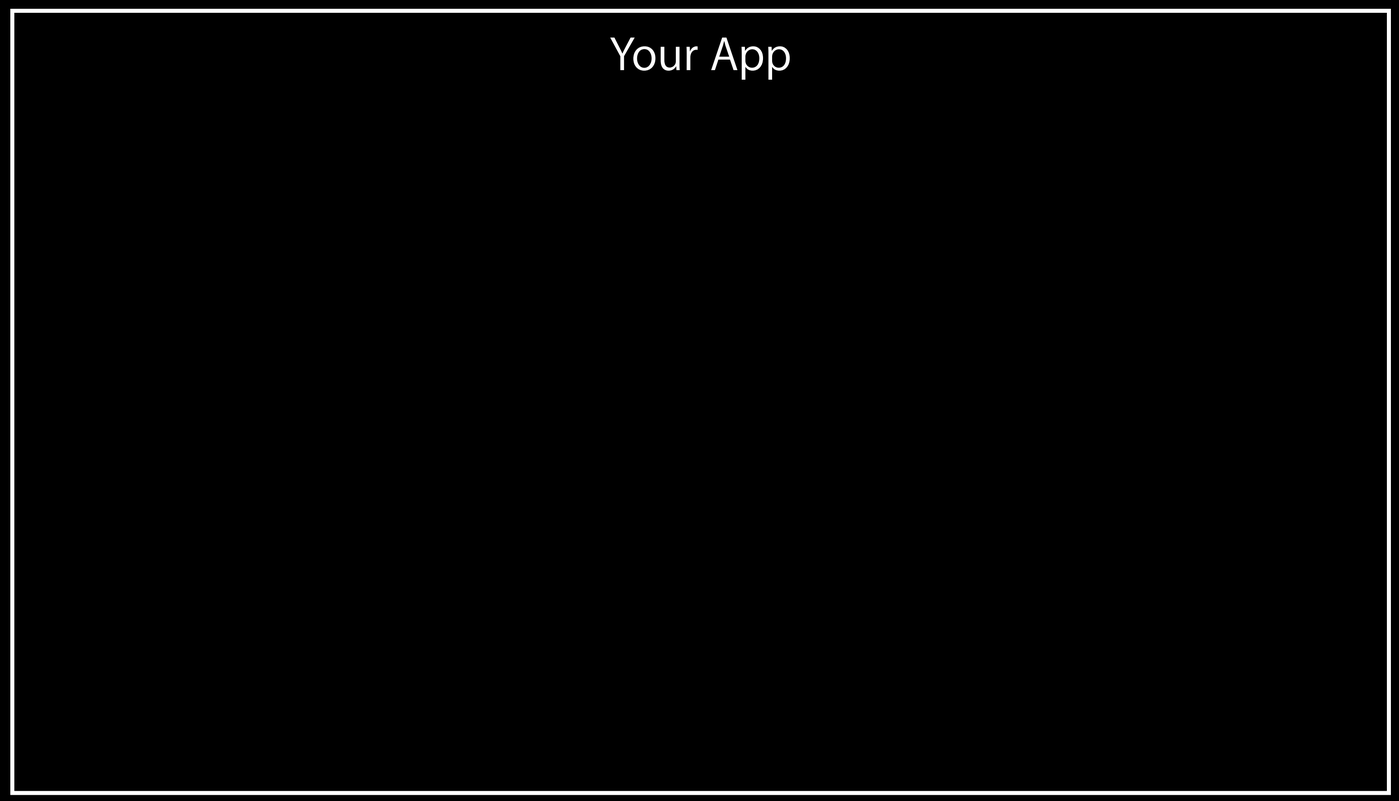
You specify scheduling requirements for work

System select the best time to perform that work

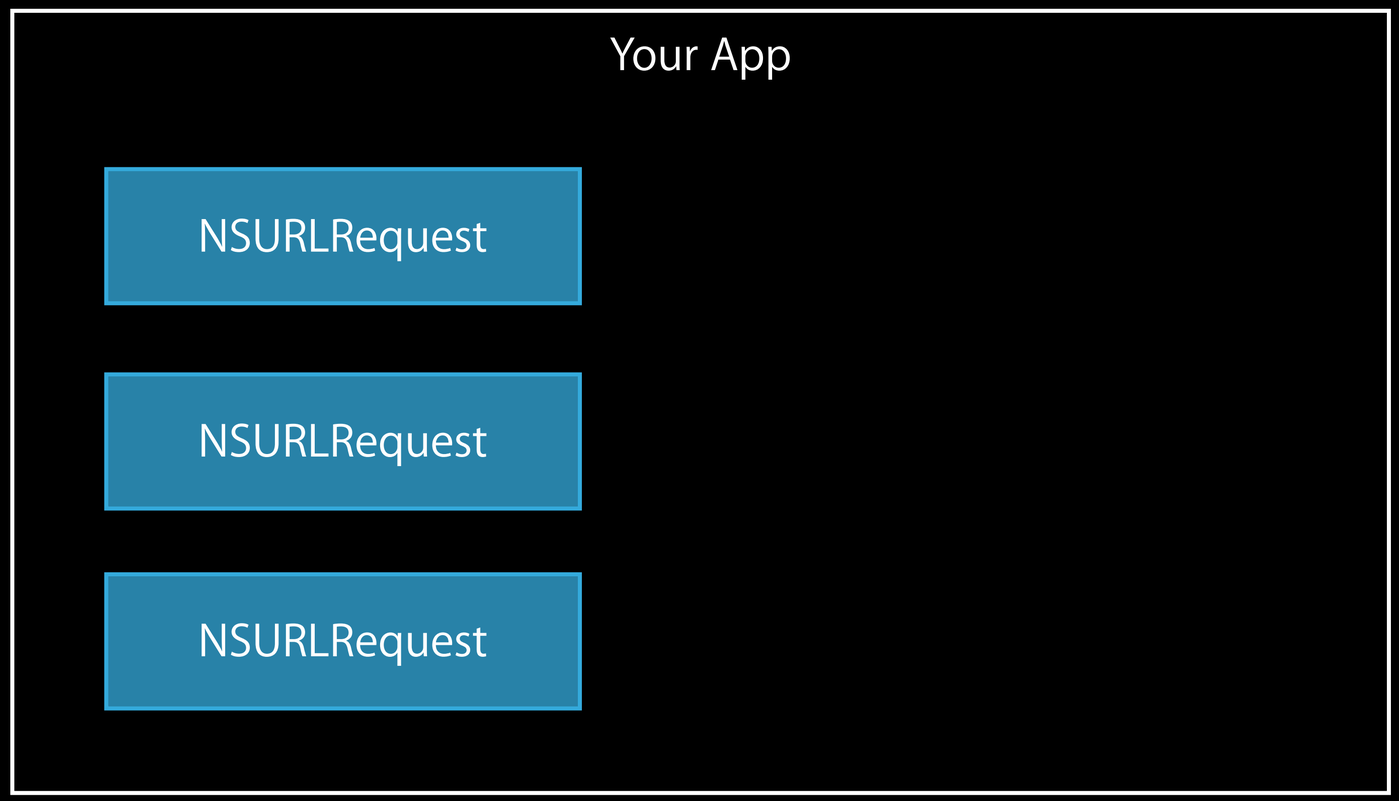
Support for repeating tasks without drift

Available in OS X Yosemite or with the xpc_activity C API in 10.9

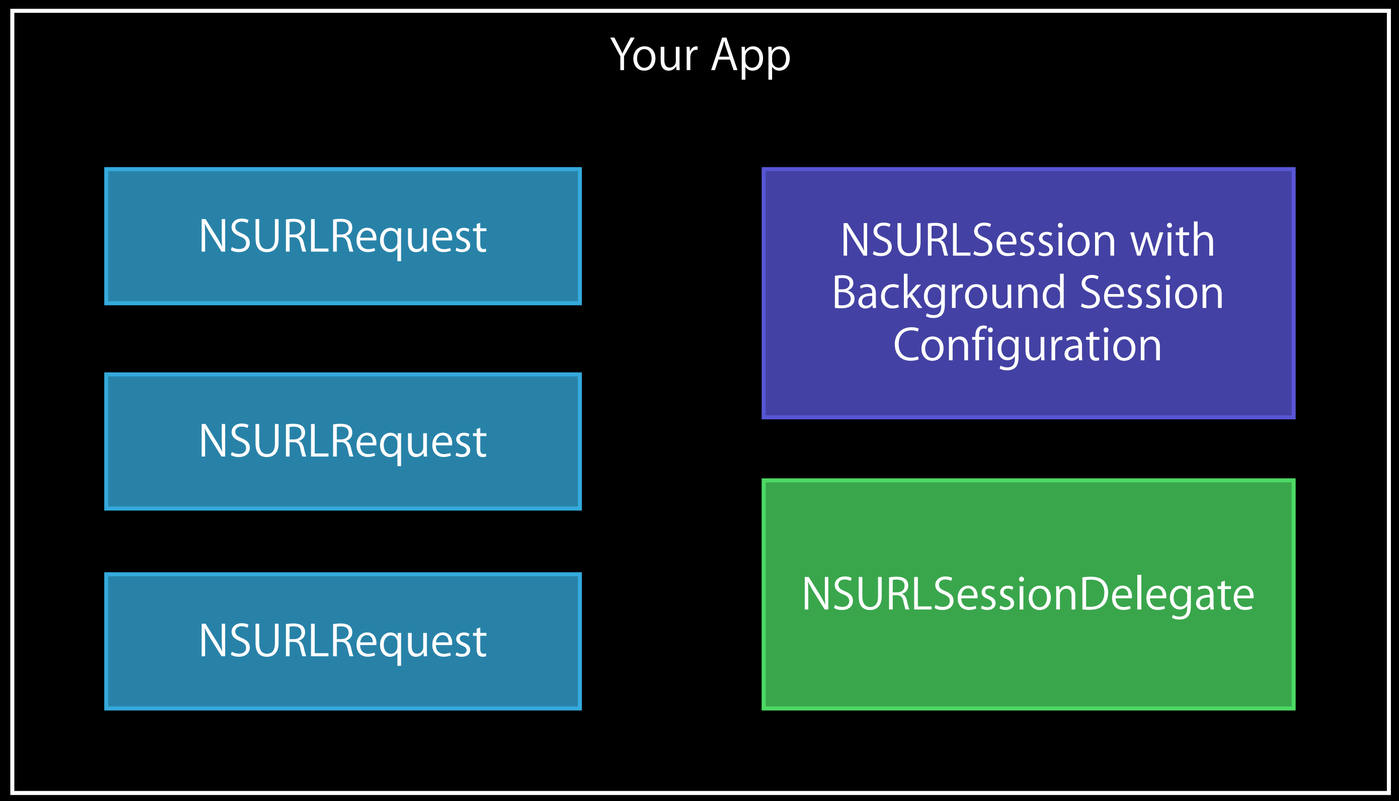
NSURLSession
Background Session



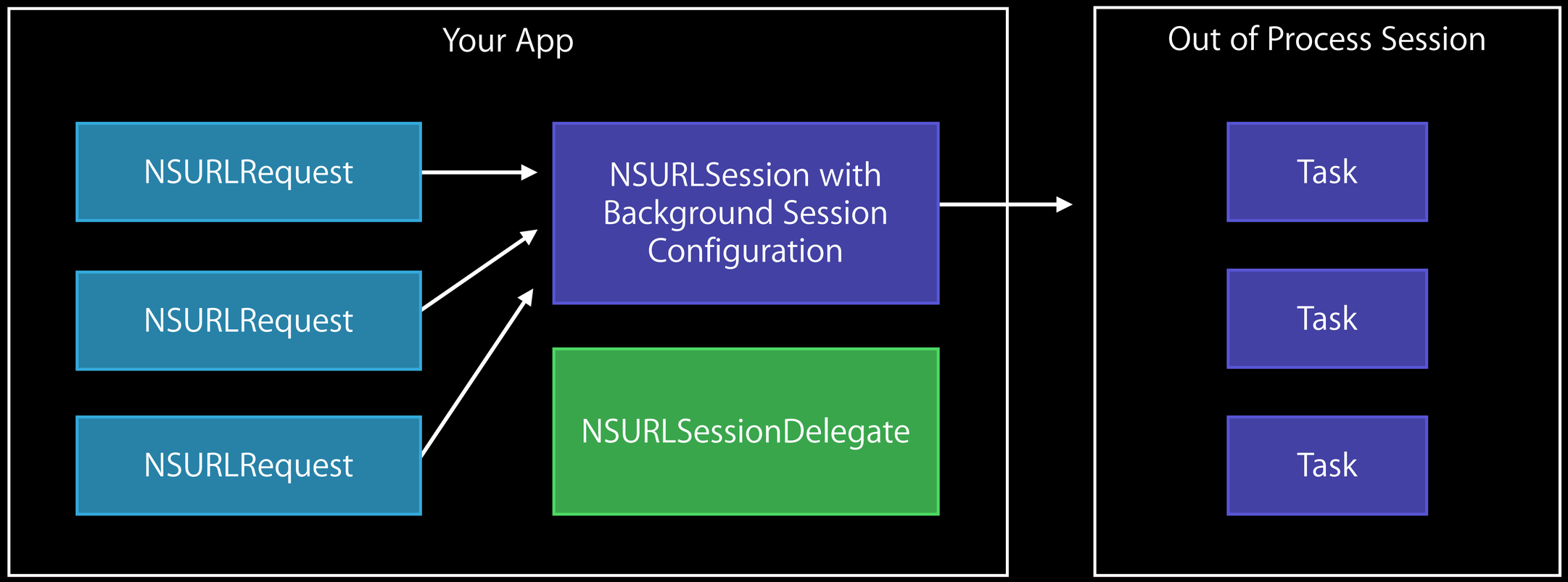
NSURLSession
Background Session



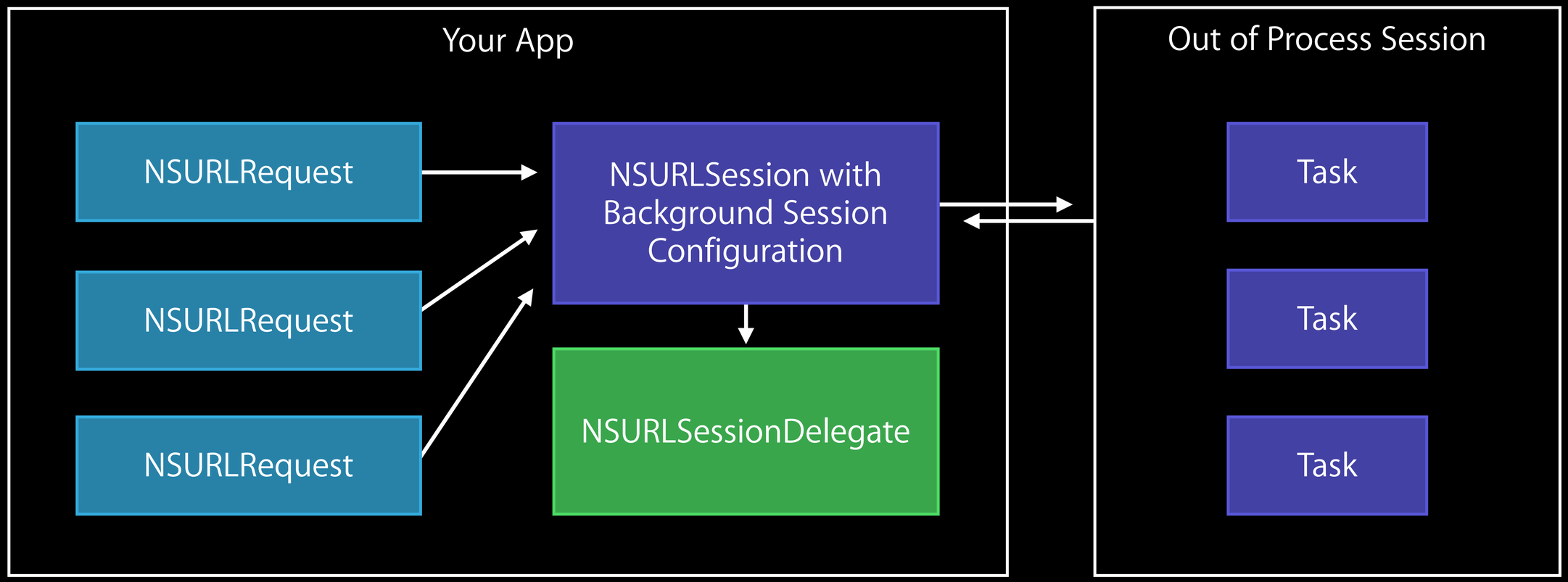
NSURLSession
Background Session



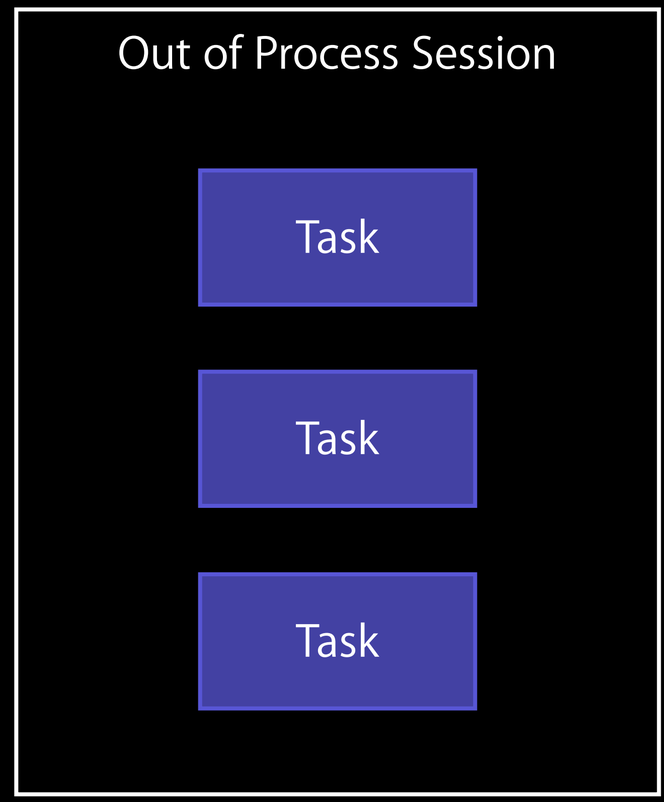
NSURLSession Background Session



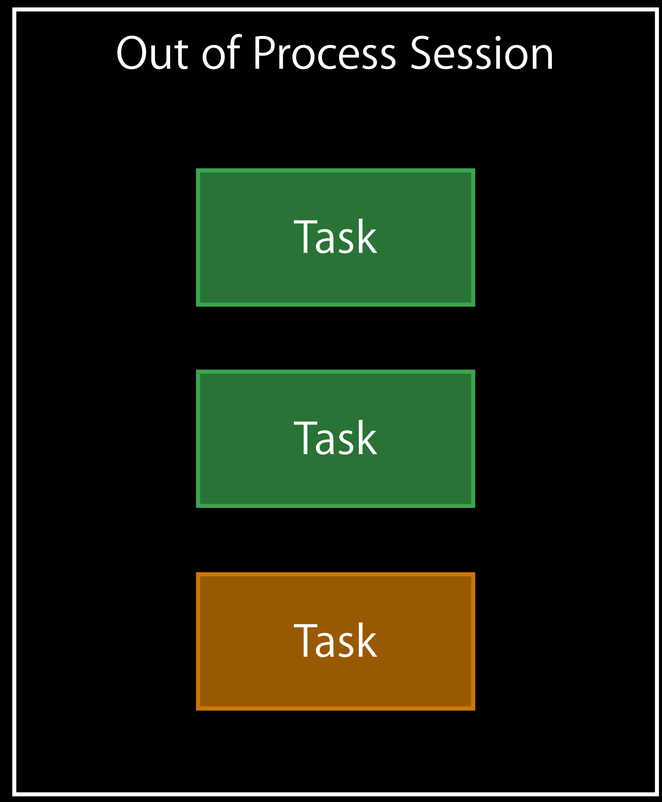
NSURLSession
Background Session



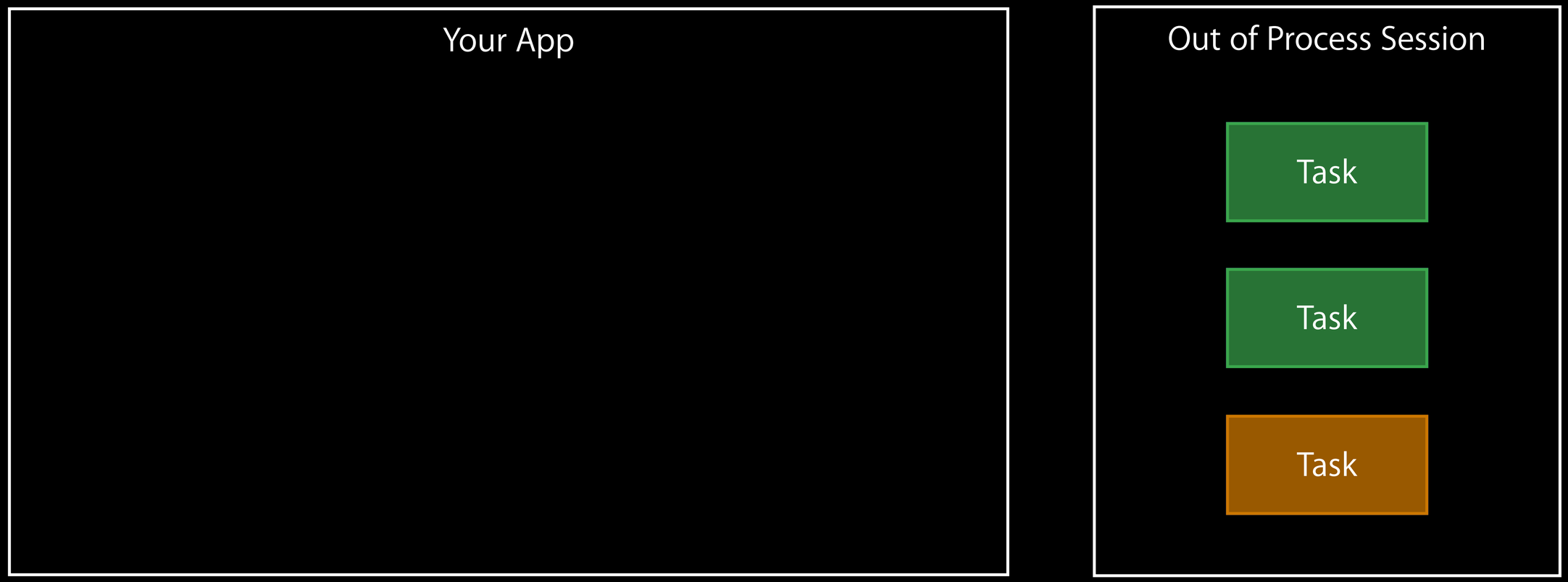
NSURLSession
Background Session



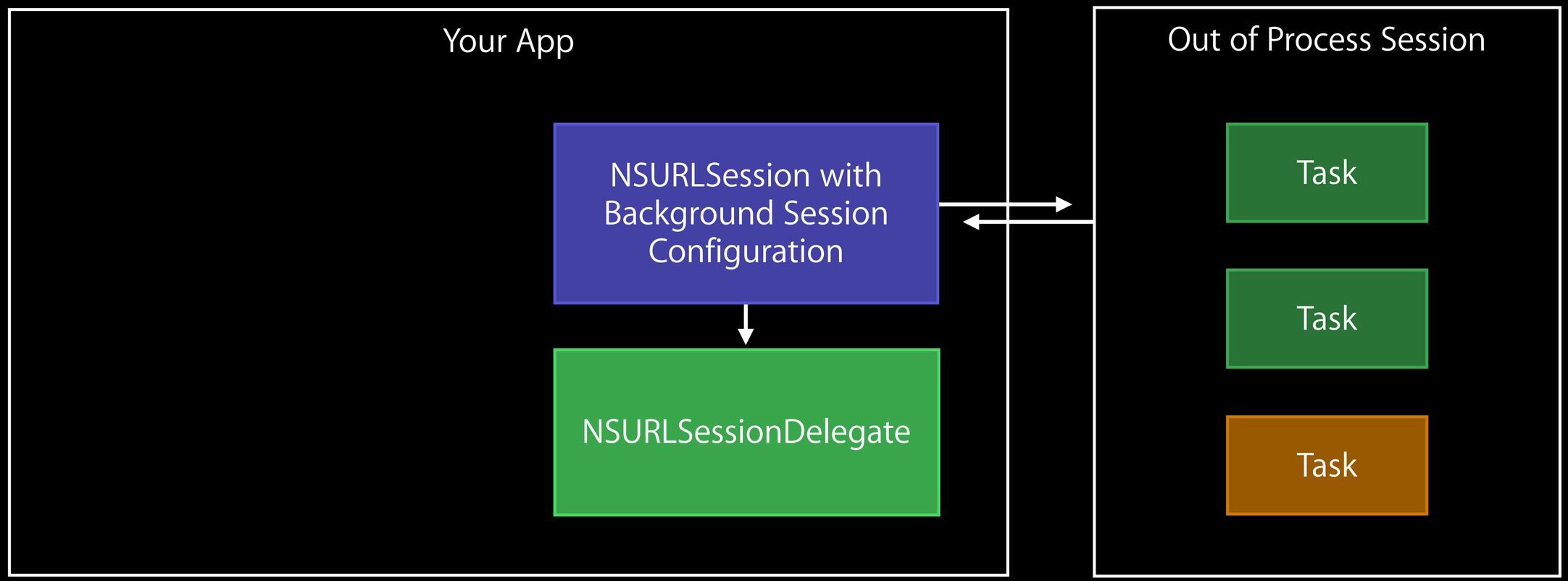
NSURLSession
Background Session



NSURLSession
Background Session



NSURLSession Background Session




```
+ backgroundSessionConfigurationWithIdentifier:
“com.example.MySessionIdentifier”
```

Discretionary Tasks

Discretionary Tasks

```
configuration.discretionary = YES;
```

Discretionary Tasks


```
configuration.discretionary = YES;
```

Discretionary sessions available in iOS 7.0 and OS X Yosemite

Discretionary Tasks


```
configuration.discretionary = YES;
```

Discretionary sessions available in iOS 7.0 and OS X Yosemite

Automatically picks the best time to do work

Discretionary Tasks


```
configuration.discretionary = YES;
```

Discretionary sessions available in iOS 7.0 and OS X Yosemite

Automatically picks the best time to do work

Provides bandwidth monitoring and automatic retry

Discretionary Tasks

```
configuration.discretionary = YES;
```

Discretionary sessions available in iOS 7.0 and OS X Yosemite

Automatically picks the best time to do work

Provides bandwidth monitoring and automatic retry

Scheduling window can be adjusted with

```
configuration.timeoutIntervalForResource = 24*60*60; // 1 day
```

- If this time elapses, an error will be thrown
- Should be >12 hours

Related Session

• What's New in Foundation Networking	Nob Hill	Tuesday 3:15PM

Do It More Efficiently

Resource Management Properties

Responsiveness

Efficiency

Resource Management Properties

Responsiveness

CPU Scheduler Priority

I/O Priority

Efficiency

Resource Management Properties

Responsiveness

CPU Scheduler Priority

I/O Priority

Efficiency

Timer Coalescing

Throughput/Efficiency CPU Hints

Quality of Service Classes

UI	User Interactive	Main thread, animations
IN	User Initiated	Immediate results
UT	Utility	Long-running tasks
BG	Background	Not user visible

Choosing a QoS Class

User Interactive

Is this work actively involved in updating the UI?
e.g., main thread, animations, input event processing

Choosing a QoS Class

UI	User Interactive	**Is this work actively involved in updating the UI?** e.g., main thread, animations, input event processing
IN	User Initiated	**Is this work required to continue user interaction?** e.g., loading active content

Choosing a QoS Class

UI	User Interactive	**Is this work actively involved in updating the UI?** e.g., main thread, animations, input event processing
IN	User Initiated	**Is this work required to continue user interaction?** e.g., loading active content
UT	Utility	**Is the user aware of the progress of this work?** e.g., long-running jobs with progress indicators

Choosing a QoS Class

	User Interactive	**Is this work actively involved in updating the UI?** e.g., main thread, animations, input event processing
	User Initiated	**Is this work required to continue user interaction?** e.g., loading active content
	Utility	**Is the user aware of the progress of this work?** e.g., long-running jobs with progress indicators
	Background	**Can this work be deferred to a better time?** e.g., if so, use `NSBackgroundActivityScheduler`

Choosing a QoS Class

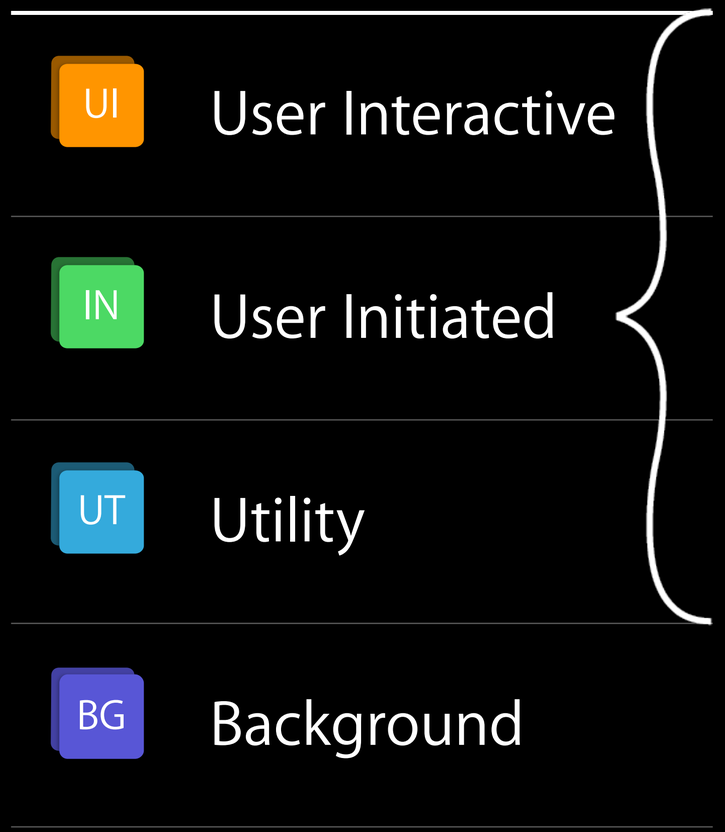
User Interactive

User Initiated

Utility

Background

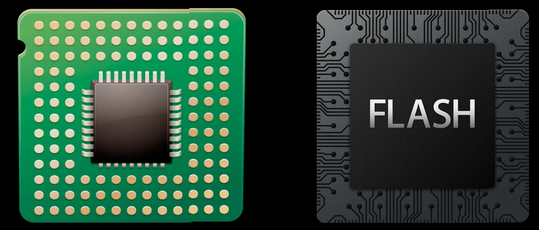
Choosing a QoS Class



Choosing a QoS Class

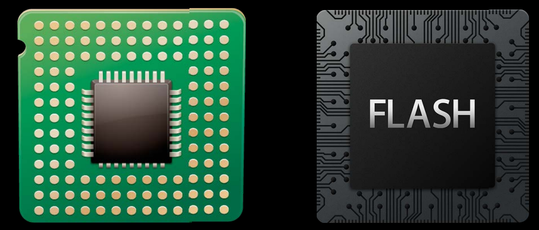
UI	User Interactive	Is it okay for User Interactive work to happen before my work?
IN	User Initiated	Is it okay for this work to compete with other User Initiated work?
UT	Utility	Is it okay for my work to take precedence over Utility work?
BG	Background	

Effects of QoS



Background

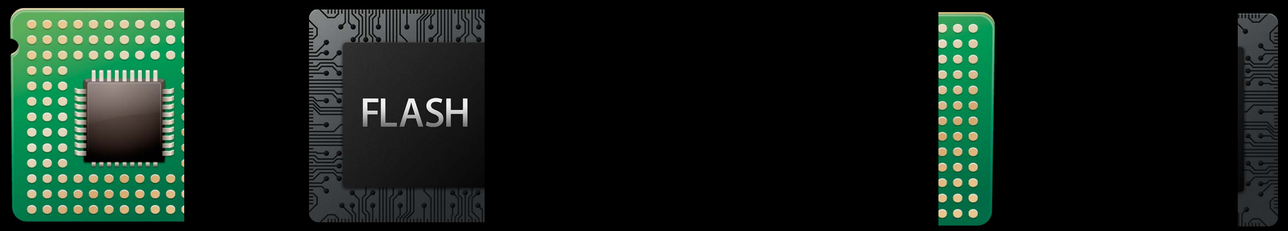
Effects of QoS



User Initiated

Background

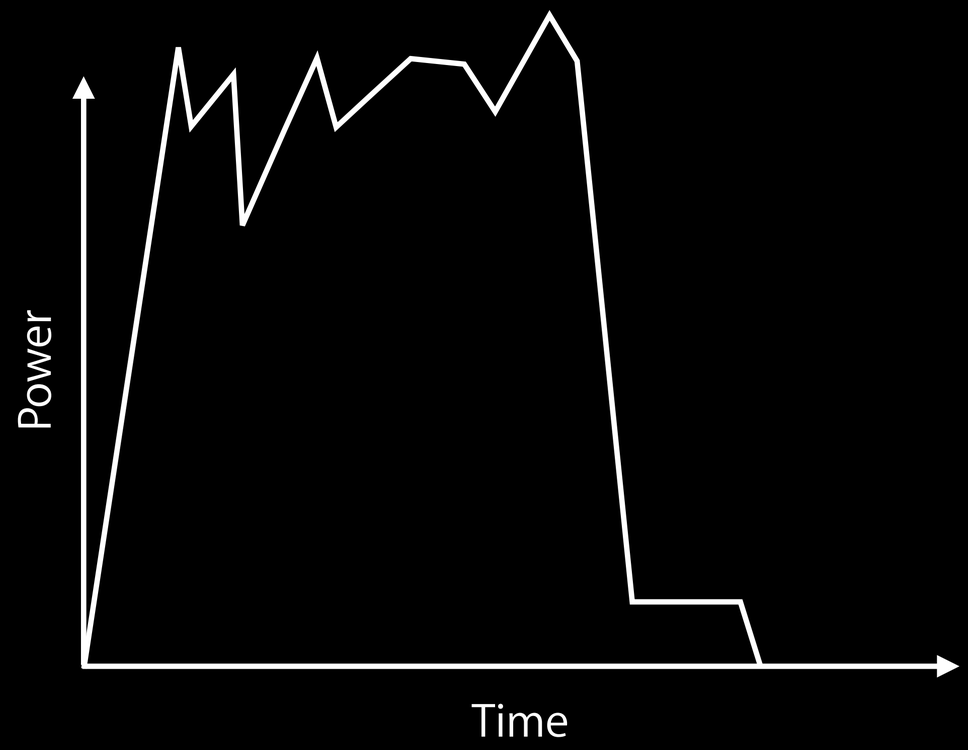
Effects of QoS



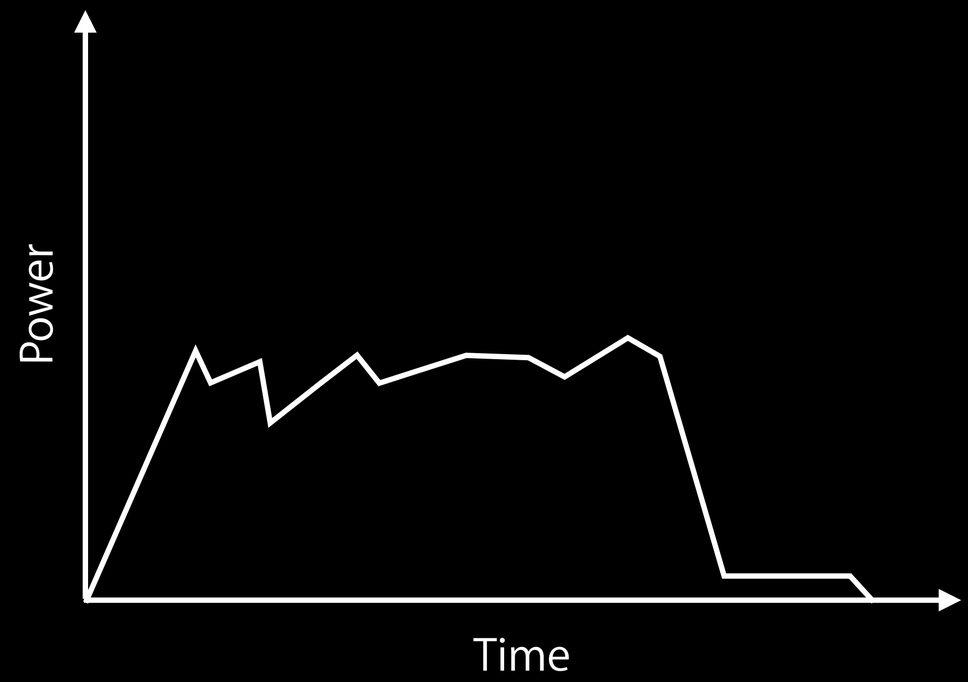
User Initiated

Background

Effects of QoS

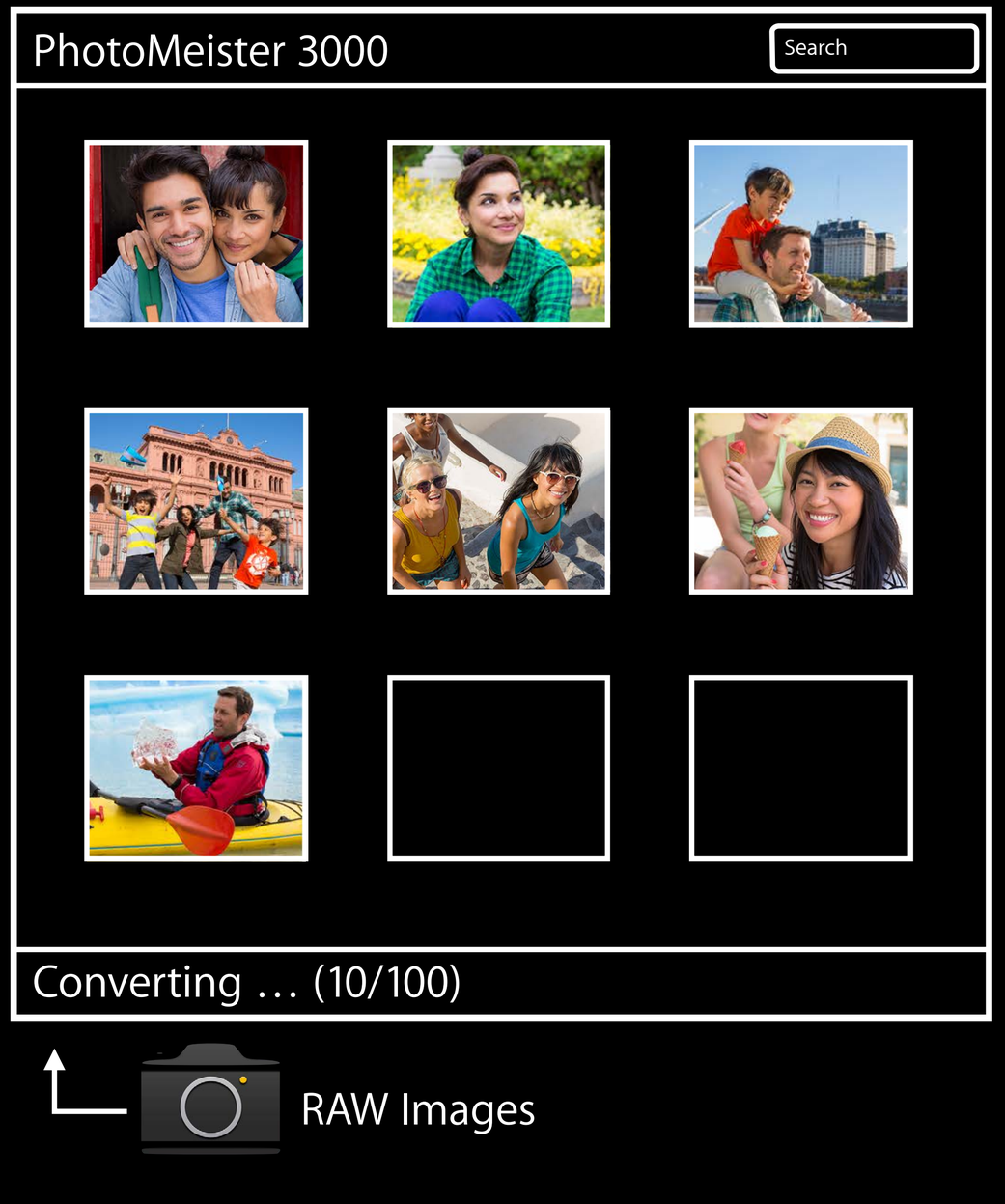


User Initiated



Background

Adoption Case Study



Adoption Case Study

User Interactive

- Main Thread (automatic)

Adoption Case Study

User Interactive

- Main Thread (automatic)

User Initiated

- Thumbnail generation
- Image load (on click)

Adoption Case Study

User Interactive

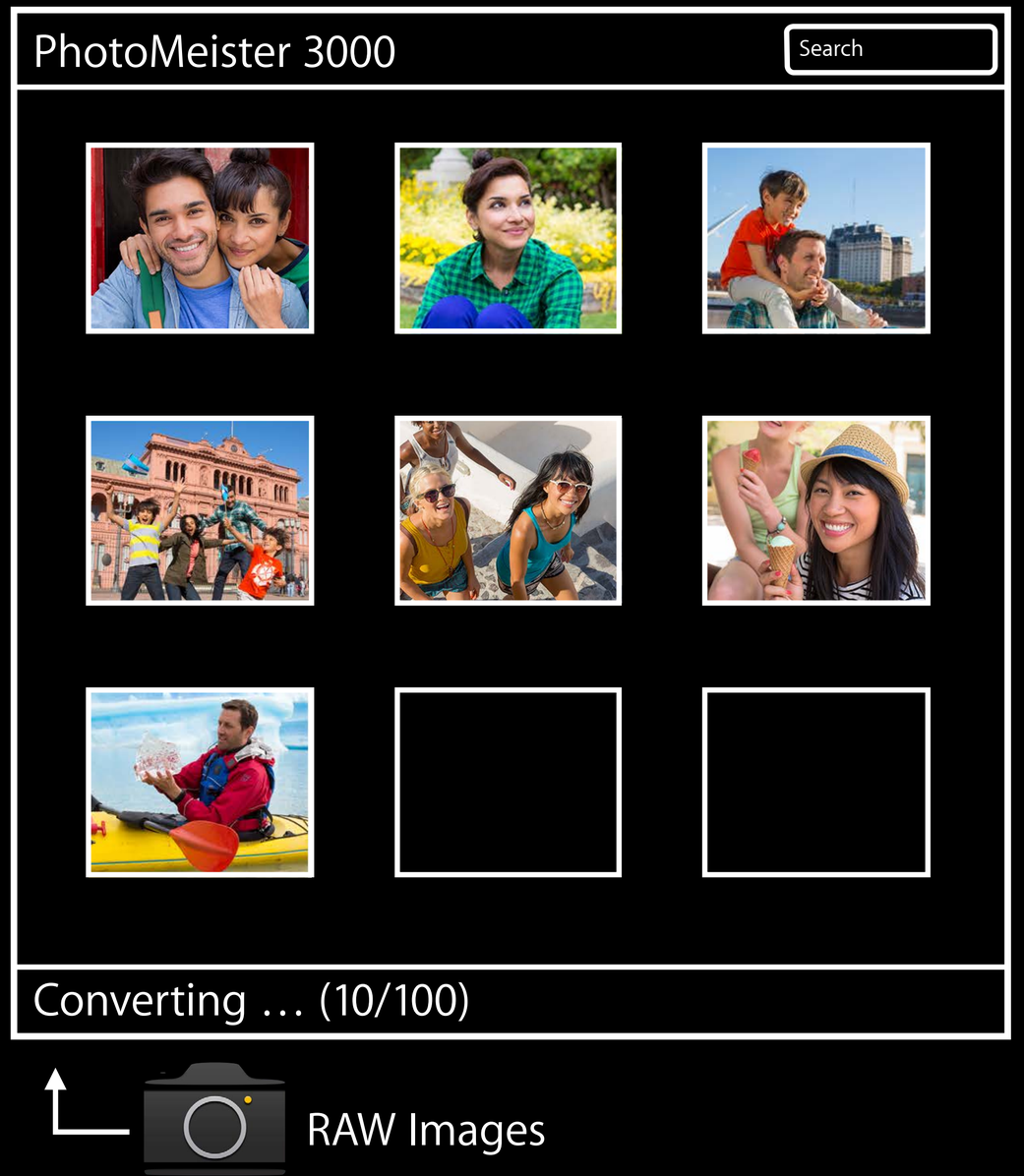
- Main Thread (automatic)

User Initiated

- Thumbnail generation
- Image load (on click)

Utility

- Image import and conversion



Adoption Case Study

User Interactive

- Main Thread (automatic)

User Initiated

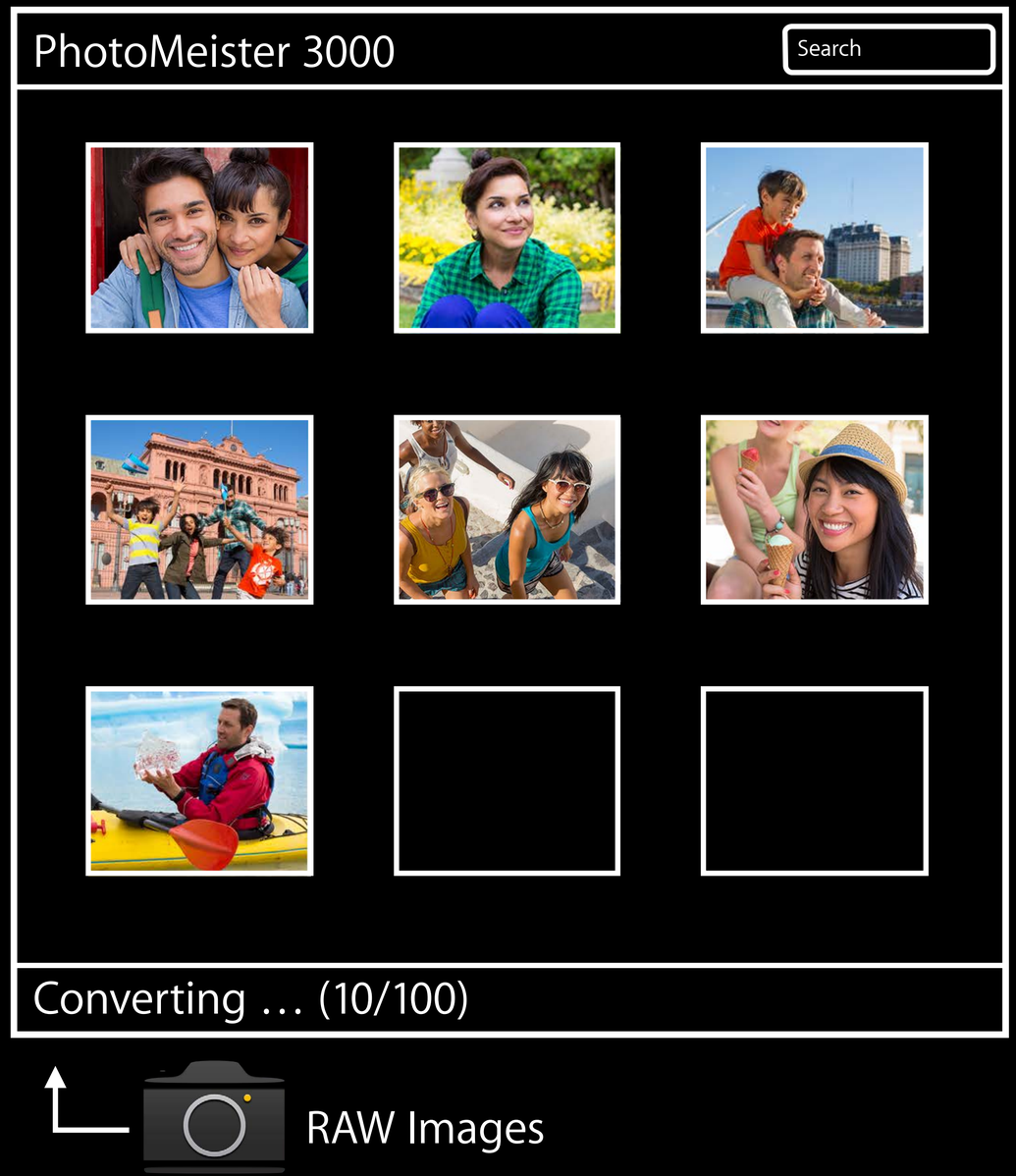
- Thumbnail generation
- Image load (on click)

Utility

- Image import and conversion

Background

- Search indexing



Queue Structure

Main Thread
(User Interactive)

Queue Structure

Main Thread
(User Interactive)

Queue Structure

Main Thread
(User Interactive)

NSOperation(Queue) and QoS

NSOperation(Queue) and QoS

NSOperation and NSOperationQueue now have a qualityOfService property:

```
operation.qualityOfService = NSQualityOfServiceUtility;
```

NSOperation(Queue) and QoS

NSOperation and NSOperationQueue now have a qualityOfService property:

```
operation.qualityOfService = NSQualityOfServiceUtility;
```

If set on both the operation and queue, the higher will be used

NSOperation(Queue) and QoS

NSOperation and NSOperationQueue now have a qualityOfService property:

```
operation.qualityOfService = NSQualityOfServiceUtility;
```

If set on both the operation and queue, the higher will be used

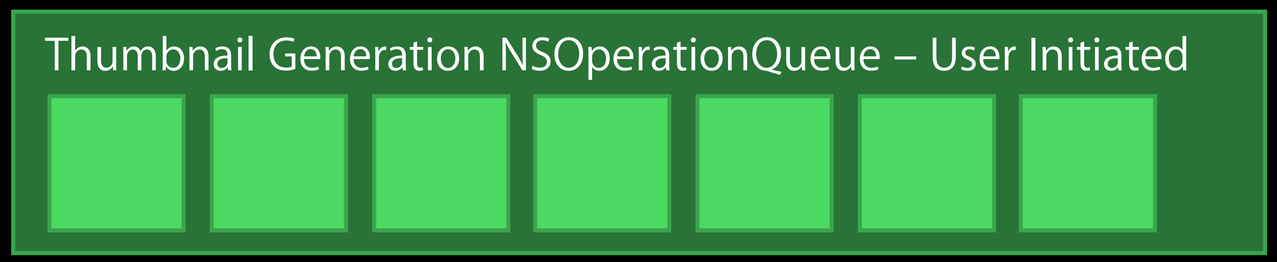
If not set, NSOperations will infer a QoS from the environment when possible

QoS Application

Main Thread
(User Interactive)

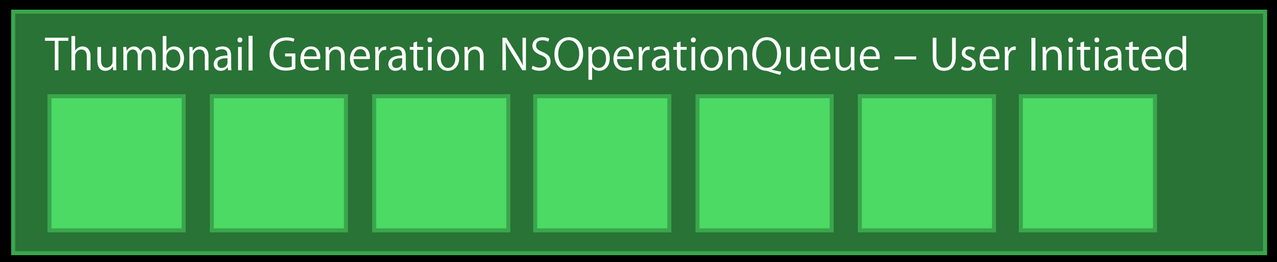
QoS Application

Main Thread
(User Interactive)



QoS Application

Main Thread
(User Interactive)



QoS Is Not Static

The logical QoS of an operation may change over time

- e.g., when user requests the result of work that's already happening a lower QoS

QoS Is Not Static

The logical QoS of an operation may change over time

- e.g., when user requests the result of work that's already happening a lower QoS

With NSOperation, the QoS of an operation can be promoted by:

- Enqueueing a higher QoS operation on the same queue
- Using addDependency: with a higher QoS operation
- `waitUntilFinished:` or `waitUntilAllOperationsAreFinished:` from a higher QoS thread

Promotion

Main Thread

Image Click Event

Image Conversion NSOperationQueue – Utility

Promotion

Main Thread

Image Click Event

```
// Find operation for image
```

Image Conversion NSOperationQueue – Utility

Promotion

Main Thread

Image Click Event

```
// Find operation for image
operation.queuePriority =
   NSOperationQueuePriorityVeryHigh;
```

Image Conversion NSOperationQueue – Utility

Promotion

Main Thread

Image Click Event

```
// Find operation for image
operation.queuePriority =
   NSOperationQueuePriorityVeryHigh;
operation.qualityOfService =
   NSQualityOfServiceUserInitiated;
```

Image Conversion NSOperationQueue – Utility

Promotion

Main Thread

Image Click Event

```
// Find operation for image
operation.queuePriority =
   NSOperationQueuePriorityVeryHigh;
operation.qualityOfService =
   NSQualityOfServiceUserInitiated;
```

Image Conversion NSOperationQueue – Utility

Your Turn


Feed Reader 9000
Search
News
Neat Things
Cats
Kittens
26 ways…
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut volutpat accumsan tellus, eu imperdiet est elementum sed. Sed tincidunt, nibh ut consectetur posuere, est metus.
Mauris in elementum orci, varius est. Sed fringilla velit a vestibulum sagi.
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Updating Feeds…

Your Turn

Loading content

Your Turn

Loading content

- User Initiated

Your Turn

Loading content

- User Initiated

Image pre-fetching

Your Turn

Loading content

- User Initiated

Image pre-fetching

- Background

Your Turn

Loading content

- User Initiated

Image pre-fetching

- Background

Fetching new feeds

Your Turn

Loading content

- User Initiated

Image pre-fetching

- Background

Fetching new feeds

- Requested by user—User Initiated

Your Turn

Loading content

- User Initiated

Image pre-fetching

- Background

Fetching new feeds

- Requested by user—User Initiated
- Automatic—Utility

Your Turn

Loading content

- User Initiated

Image pre-fetching

- Background

Fetching new feeds

- Requested by user—User Initiated
- Automatic—Utility

Search indexing

Your Turn

Loading content

- User Initiated

Image pre-fetching

- Background

Fetching new feeds

- Requested by user—User Initiated
- Automatic—Utility

Search indexing

- Background

Debugging QoS

Debugging QoS

Set breakpoints to confirm requested QoS

Debugging QoS

Set breakpoints to confirm requested QoS

Use powermetrics to confirm which QoS is in use

Debugging QoS

Set breakpoints to confirm requested QoS

Use powermetrics to confirm which QoS is in use

Use spindump to determine the QoS code is executing with

MyApplication 〉 My Mac 64–bit

Running MyApplication : MyApplication

1

CPU Report

MyApplication
PID 7632, Paused

CPU 7%

Memory 12.7 MB

Energy Impact Low

Disk Zero KB/s

Network Zero KB/s

Thread 1
Queue: com.apple.main-thread (serial)
- 0 mach_msg_trap
- 11 NSApplicationMain
- 12 main
- 13 start

Thread 2
Queue: com.apple.libdispatch-manager (serial)

Thread 5
Queue: com.apple.root.utility-qos (concurrent)
- 0 __semwait_signal
- 2 usleep
- 3 __45-[AppDelegate applicationDidFini…
- 4 _dispatch_call_block_and_release
- 9 start_wqthread

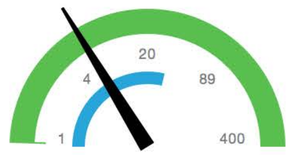
Enqueued from com.apple.main-thread (Thr…
- 0 _dispatch_async_f_slow
- 1 -[AppDelegate applicationDidFinishLa…
- 2 __CFNOTIFICATIONCENTER_IS_CALL…
- 18 NSApplicationMain
- 19 main
- 20 start

Thread 7

Thread 9

Thread 10

Percentage Used



7%

Usage Comparison

MyApplication
7%

Other Processes
70%

Free
323%

Usage over Time

Duration: 45 sec
High: 31%
Low: 0%

Threads

MyApplication 〉 Thread 5 〉 0 __semwait_signal

(lldb)

Auto

All Output

MyApplication 〉 My Mac 64–bit | Running MyApplication : MyApplication | 1

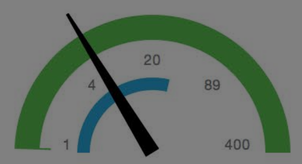
CPU Report

MyApplication
PID 7632, Paused

CPU	7%
Memory	12.7 MB
Energy Impact	Low
Disk	Zero KB/s
Network	Zero KB/s

- **Thread 1** Queue: com.apple.main-thread (serial)
  - 0 mach_msg_trap
  - 11 NSApplicationMain
  - 12 main
  - 13 start
- **Thread 2** Queue: com.apple.libdispatch-manager (serial)
- **Thread 5** Queue: com.apple.root.utility-qos (concurrent)
  - 0 __semwait_signal
  - 2 usleep
  - 3 __45-[AppDelegate applicationDidFini…
  - 4 _dispatch_call_block_and_release
  - 9 start_wqthread
  - Enqueued from com.apple.main-thread (Thr…
  - 0 _dispatch_async_f_slow
  - 1 -[AppDelegate applicationDidFinishLa…
  - 2 __CFNOTIFICATIONCENTER_IS_CALL…
  - 18 NSApplicationMain
  - 19 main
  - 20 start
- **Thread 7**
- **Thread 9**
- **Thread 10**

Percentage Used



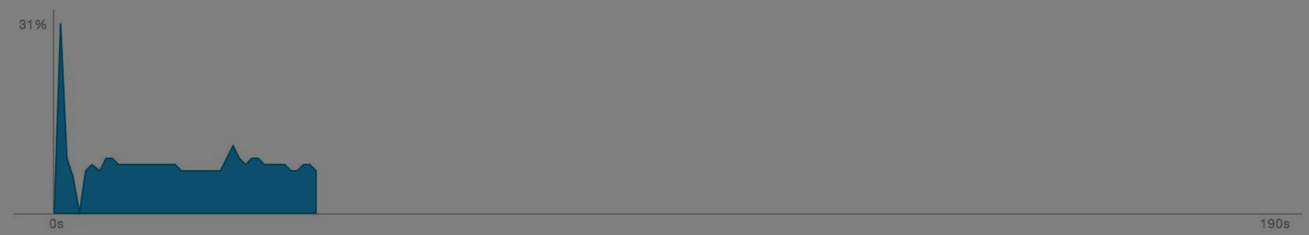
7%

Usage Comparison

- **MyApplication** 7%
- **Other Processes** 70%
- **Free** 323%

Usage over Time

Duration: 45 sec
High: 31%
Low: 0%



Threads

MyApplication 〉 Thread 5 〉 0 __semwait_signal

(lldb)

Auto | All Output

MyApplication 〉 My Mac 64–bit

Running MyApplication : MyApplication

1

CPU Report

MyApplication
PID 7632, Paused

CPU 7%

Memory 12.7 MB

Energy Impact Low

Disk Zero KB/s

Network Zero KB/s

Thread 1
Queue: com.apple.main-thread (serial)
- 0 mach_msg_trap
- 11 NSApplicationMain
- 12 main
- 13 start

Thread 2
Queue: com.apple.libdispatch-manager (serial)

Thread 5
Queue: com.apple.root.utility-qos (concurrent)
- 0 __semwait_signal
- 2 usleep
- 3 __45-[AppDelegate applicationDidFini…
- 4 _dispatch_call_block_and_release
- 9 start_wqthread

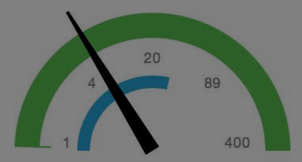
Enqueued from com.apple.main-thread (Thr…
- 0 _dispatch_async_f_slow
- 1 -[AppDelegate applicationDidFinishLa…
- 2 __CFNOTIFICATIONCENTER_IS_CALL…
- 18 NSApplicationMain
- 19 main
- 20 start

Thread 7

Thread 9

Thread 10

Percentage Used



7%

Usage Comparison

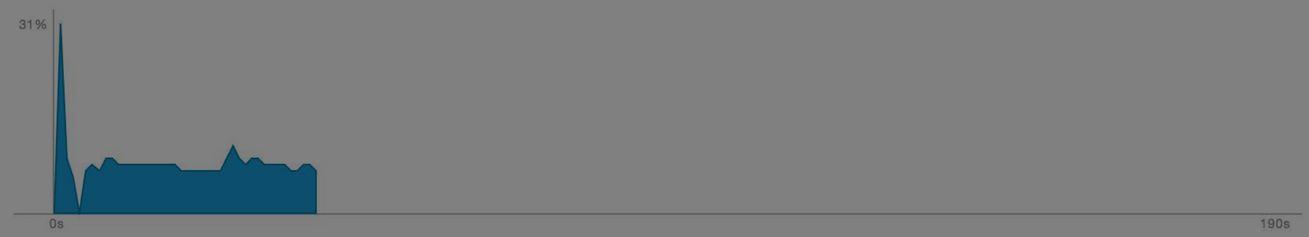
MyApplication
7%

Other Processes
70%

Free
323%

Usage over Time

Duration: 45 sec
High: 31%
Low: 0%



Threads

MyApplication 〉 Thread 5 〉 0 __semwait_signal

(lldb)

Auto

All Output

MyApplication 〉 My Mac 64–bit

Running MyApplication : MyApplication

1

CPU Report

MyApplication
PID 7632, Paused

CPU 7%

Memory 12.7 MB

Energy Impact Low

Disk Zero KB/s

Network Zero KB/s

Thread 1
Queue: com.apple.main-thread (serial)
- 0 mach_msg_trap
- 11 NSApplicationMain
- 12 main
- 13 start

Thread 2
Queue: com.apple.libdispatch-manager (serial)

Thread 5
Queue: com.apple.root.utility-qos (concurrent)
- 0 __semwait_signal
- 2 usleep
- 3 __45-[AppDelegate applicationDidFini…
- 4 _dispatch_call_block_and_release
- 9 start_wqthread

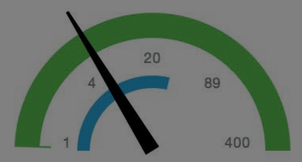
Enqueued from com.apple.main-thread (Thr…
- 0 _dispatch_async_f_slow
- 1 -[AppDelegate applicationDidFinishLa…
- 2 __CFNOTIFICATIONCENTER_IS_CALL…
- 18 NSApplicationMain
- 19 main
- 20 start

Thread 7

Thread 9

Thread 10

Percentage Used



7%

Usage Comparison

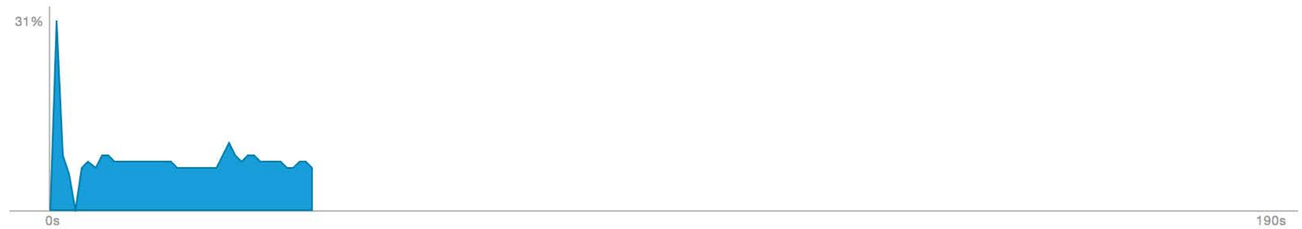
MyApplication
7%

Other Processes
70%

Free
323%

Usage over Time

Duration: 45 sec
High: 31%
Low: 0%



Threads

MyApplication 〉 Thread 5 〉 0 __semwait_signal

(lldb)

Auto

All Output

MyApplication 〉 My Mac 64–bit

Running MyApplication : MyApplication

1

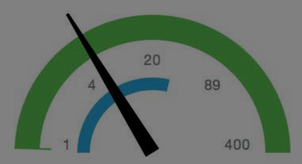
CPU Report

MyApplication
PID 7632, Paused

CPU	7%
Memory	12.7 MB
Energy Impact	Low
Disk	Zero KB/s
Network	Zero KB/s

- **Thread 1** Queue: com.apple.main-thread (serial)
  - 0 mach_msg_trap
  - 11 NSApplicationMain
  - 12 main
  - 13 start
- **Thread 2** Queue: com.apple.libdispatch-manager (serial)
- **Thread 5** Queue: com.apple.root.utility-qos (concurrent)
  - 0 __semwait_signal
  - 2 usleep
  - 3 __45-[AppDelegate applicationDidFini…
  - 4 _dispatch_call_block_and_release
  - 9 start_wqthread
  - Enqueued from com.apple.main-thread (Thr…
  - 0 _dispatch_async_f_slow
  - 1 -[AppDelegate applicationDidFinishLa…
  - 2 __CFNOTIFICATIONCENTER_IS_CALL…
  - 18 NSApplicationMain
  - 19 main
  - 20 start
- **Thread 7**
- **Thread 9**
- **Thread 10**

Percentage Used



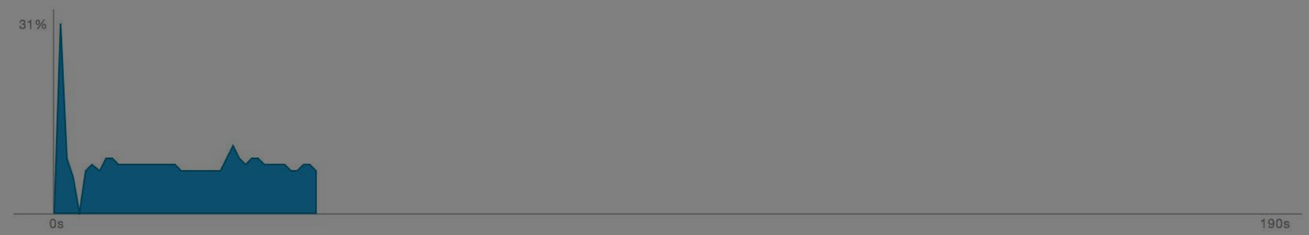
7%

Usage Comparison

- **MyApplication** 7%
- **Other Processes** 70%
- **Free** 323%

Usage over Time

Duration: 45 sec
High: 31%
Low: 0%



Threads

Thread 1	User Interactive
Thread 2	User Interactive
Thread 3	QoS Unavailable
Thread 5	Utility
Thread 7	User Interactive
Thread 9	Default

MyApplication 〉 Thread 5 〉 0 __semwait_signal

(lldb)

Auto

All Output

QoS Accounting

```
% sudo powermetrics --show-process-qos --samplers tasks
```

QoS Accounting

```
% sudo powermetrics --show-process-qos --samplers tasks
*** Sampled system activity (Wed May 28 00:03:12 2014 -0700) (5006.36ms
elapsed) ***

*** Running tasks ***

Name                                ID     CPU ms/s  User%
MyApplication                       8424   88.89     94.16

    Deadlines (<2 ms, 2-5 ms)  Wakeups (Intr, Pkg idle)
    227.89  0.00               228.89  165.98

    QOS (ms/s) Default Maint   BG      Util    Lgcy    U-Init  U-Intr
               0.00    0.00    0.04    88.64   0.03    0.00    0.17
```

QoS Accounting

```
% sudo powermetrics --show-process-qos --samplers tasks
*** Sampled system activity (Wed May 28 00:03:12 2014 -0700) (5006.36ms
elapsed) ***

*** Running tasks ***
```

Name	ID	CPU ms/s	User%
MyApplication	8424	88.89	94.16

```
    Deadlines (<2 ms, 2-5 ms)  Wakeups (Intr, Pkg idle)
    227.89  0.00               228.89  165.98

    QOS (ms/s) Default Maint   BG      Util    Lgcy    U-Init  U-Intr
               0.00    0.00    0.04    88.64   0.03    0.00    0.17
```

QoS Accounting

```
% sudo powermetrics --show-process-qos --samplers tasks
*** Sampled system activity (Wed May 28 00:03:12 2014 -0700) (5006.36ms
elapsed) ***

*** Running tasks ***

Name                                ID     CPU ms/s  User%
MyApplication                       8424   88.89     94.16

    Deadlines (<2 ms, 2-5 ms)  Wakeups (Intr, Pkg idle)
    227.89  0.00               228.89  165.98

    QOS (ms/s) Default Maint   BG      Util    Lgcy    U-Init  U-Intr
               0.00    0.00    0.04    88.64   0.03    0.00    0.17
```

QoS Accounting

```
% sudo powermetrics --show-process-qos --samplers tasks
*** Sampled system activity (Wed May 28 00:03:12 2014 -0700) (5006.36ms
elapsed) ***

*** Running tasks ***

Name                                ID     CPU ms/s  User%
MyApplication                       8424   88.89     94.16

    Deadlines (<2 ms, 2-5 ms)  Wakeups (Intr, Pkg idle)
    227.89  0.00               228.89  165.98

    QOS (ms/s) Default Maint   BG      Util    Lgcy    U-Init  U-Intr
               0.00    0.00    0.04    88.64   0.03    0.00    0.17
```

Sampling QoS

```
% sudo spindump –timeline MyApplication
```

Sampling QoS

```
% sudo spindump -timeline MyApplication
…
Thread 0x6bb7a      DispatchQueue 6     1000 samples (1-1000) priority 16-20      cpu time 1.488s
<thread QoS utility, priority 20>
1000  start_wqthread + 13 (libsystem_pthread.dylib + 6657) [0x7fff82e73a01] 1-1000
  1000  _pthread_wqthread + 663 (libsystem_pthread.dylib + 15313) [0x7fff82e75bd1] 1-1000
    1000  _dispatch_worker_thread3 + 79 (libdispatch.dylib + 72233) [0x10002fa29] 1-1000
      1000  _dispatch_root_queue_drain + 1408 (libdispatch.dylib + 17968) [0x100022630] 1-1000
…
            4    __45-[AppDelegate applicationDidFinishLaunching:]_block_invoke + 181
(AppDelegate.m:25 in MyApplication + 4837) [0x1000012e5] 1-4
<thread QoS background>
            2    __45-[AppDelegate applicationDidFinishLaunching:]_block_invoke + 217
(AppDelegate.m:28 in MyApplication + 4837) [0x1000012e5] 1-4
```

Sampling QoS

```
% sudo spindump -timeline MyApplication
…
Thread 0x6bb7a      DispatchQueue 6     1000 samples (1-1000) priority 16-20      cpu time 1.488s
<thread QoS utility, priority 20>
1000  start_wqthread + 13 (libsystem_pthread.dylib + 6657) [0x7fff82e73a01] 1-1000
  1000  _pthread_wqthread + 663 (libsystem_pthread.dylib + 15313) [0x7fff82e75bd1] 1-1000
    1000  _dispatch_worker_thread3 + 79 (libdispatch.dylib + 72233) [0x10002fa29] 1-1000
      1000  _dispatch_root_queue_drain + 1408 (libdispatch.dylib + 17968) [0x100022630] 1-1000
…
            4    __45-[AppDelegate applicationDidFinishLaunching:]_block_invoke + 181
(AppDelegate.m:25 in MyApplication + 4837) [0x1000012e5] 1-4
<thread QoS background>
            2    __45-[AppDelegate applicationDidFinishLaunching:]_block_invoke + 217
(AppDelegate.m:28 in MyApplication + 4837) [0x1000012e5] 1-4
```

Sampling QoS

```
% sudo spindump -timeline MyApplication
…
Thread 0x6bb7a      DispatchQueue 6     1000 samples (1-1000) priority 16-20      cpu time 1.488s
<thread QoS utility, priority 20>
1000  start_wqthread + 13 (libsystem_pthread.dylib + 6657) [0x7fff82e73a01] 1-1000
  1000  _pthread_wqthread + 663 (libsystem_pthread.dylib + 15313) [0x7fff82e75bd1] 1-1000
    1000  _dispatch_worker_thread3 + 79 (libdispatch.dylib + 72233) [0x10002fa29] 1-1000
      1000  _dispatch_root_queue_drain + 1408 (libdispatch.dylib + 17968) [0x100022630] 1-1000
…
            4    __45-[AppDelegate applicationDidFinishLaunching:]_block_invoke + 181
(AppDelegate.m:25 in MyApplication + 4837) [0x1000012e5] 1-4
<thread QoS background>
            2    __45-[AppDelegate applicationDidFinishLaunching:]_block_invoke + 217
(AppDelegate.m:28 in MyApplication + 4837) [0x1000012e5] 1-4
```

Quality of Service

Specify the responsiveness and energy requirements of work

Available in both Foundation and C APIs

Classify long-running or resource-intensive operations in your existing code

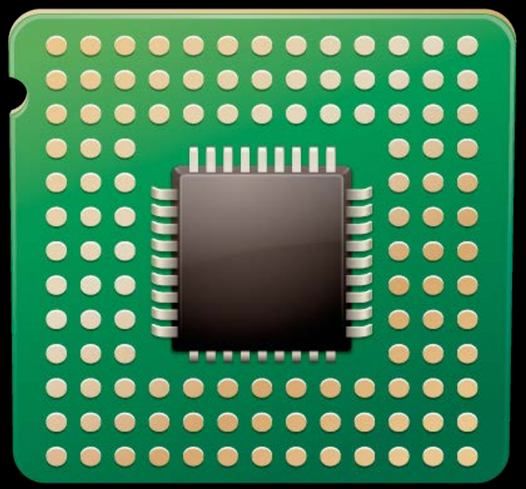
Aim for >90% of time at Utility or below when the user is inactive

Related Session

•	Power, Performance, and Diagnostics: What's New in GCD and XPC	Russian Hill	Thursday 2:00PM

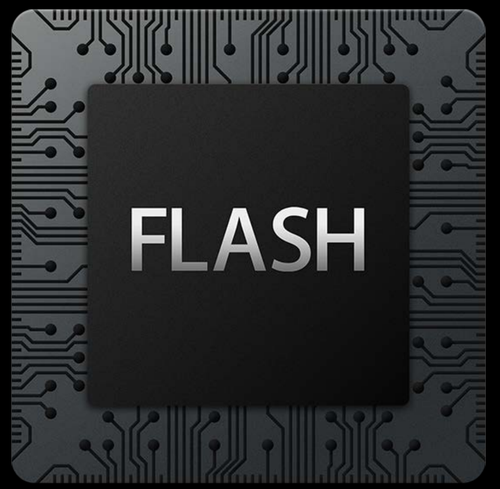
Do It Less

Use Less…



CPU

Graphics



Storage

My Mac 64–bit | Running MyApplication : MyApplication

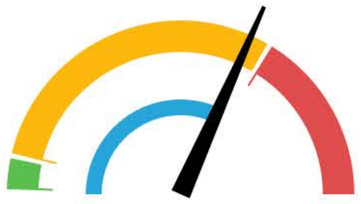
Energy Report

MyApplication
PID 9197, Running

CPU	95%
Memory	10.4 MB
Energy Impact	High
Disk	Zero KB/s
Network	Zero KB/s

Energy

Utilization



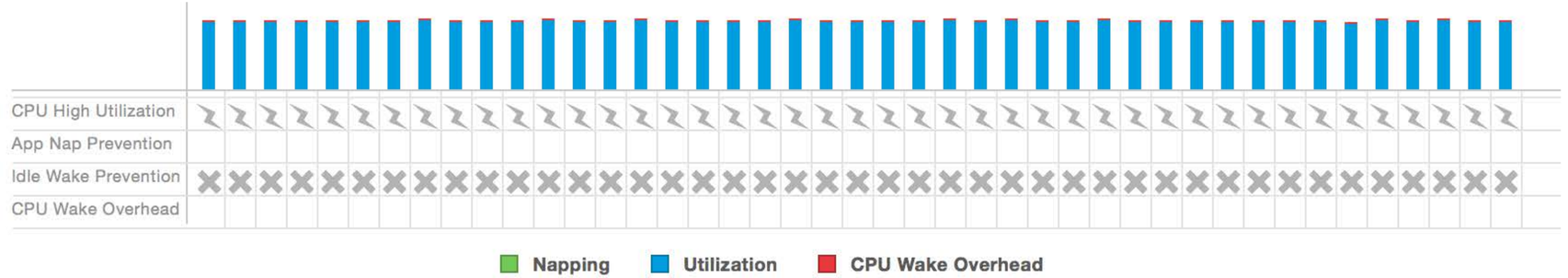
High
Energy Impact

Wakes and CPU

33 Wakes
Last Second

32 average per second

Energy Impact



MyApplication

My Mac 64–bit | Running MyApplication : MyApplication

Energy Report

MyApplication
PID 9197, Running

- **CPU** 95%
- **Memory** 10.4 MB
- **Energy Impact** High
- **Disk** Zero KB/s
- **Network** Zero KB/s

Energy

Utilization

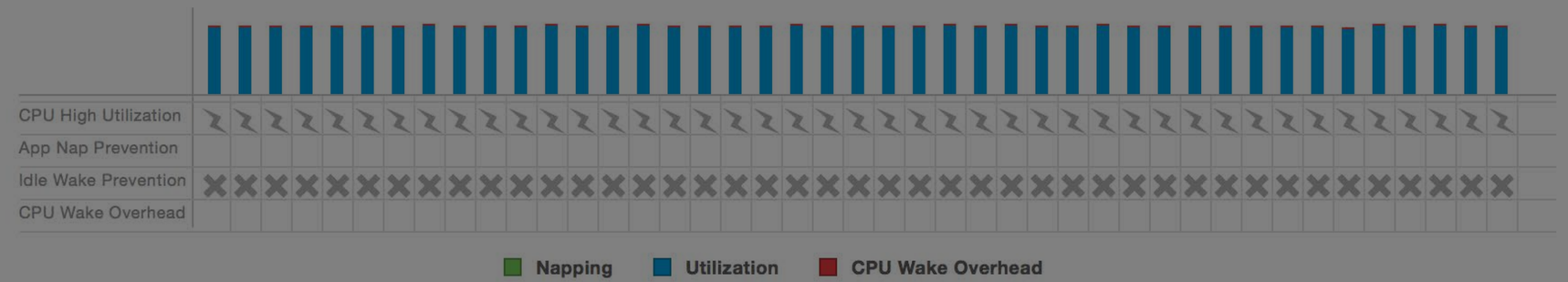
High
Energy Impact

Wakes and CPU

33 Wakes
Last Second

32 average per second

Energy Impact



MyApplication

My Mac 64–bit | Running MyApplication : MyApplication

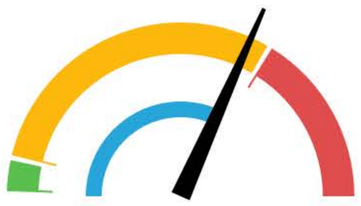
Energy Report

MyApplication
PID 9197, Running

CPU	95%
Memory	10.4 MB
Energy Impact	High
Disk	Zero KB/s
Network	Zero KB/s

Energy

Utilization



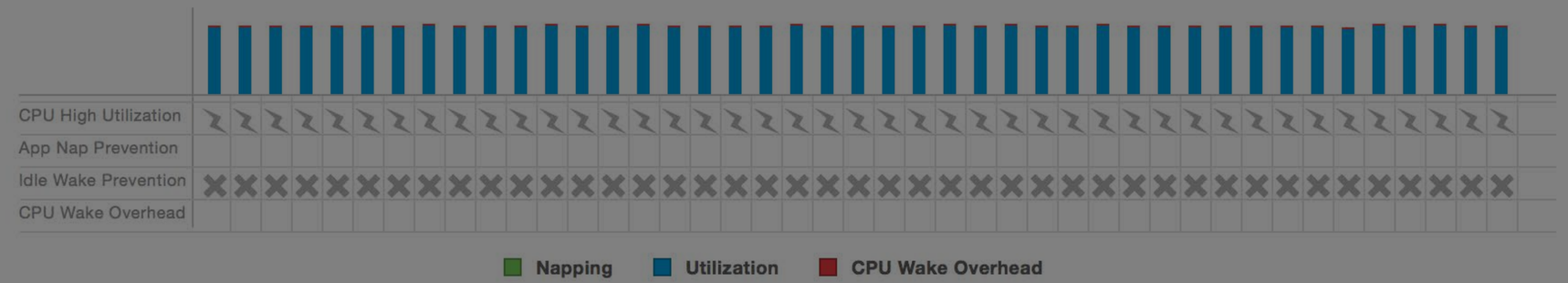
High
Energy Impact

Wakes and CPU

33 Wakes
Last Second

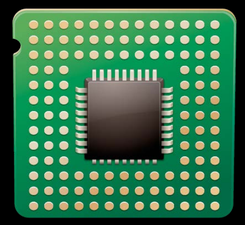
32 average per second

Energy Impact



MyApplication

Reduce CPU Use

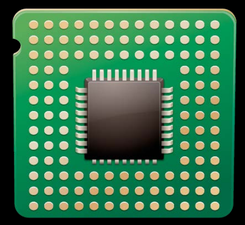


1% CPU

10% CPU

100% CPU

Reduce CPU Use



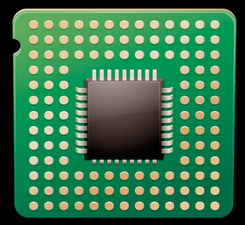
1% CPU

10% higher power draw

10% CPU

100% CPU

Reduce CPU Use



1% CPU

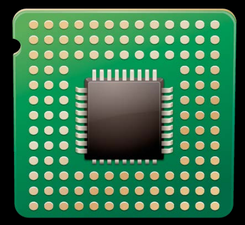
10% higher power draw

10% CPU

2x power draw

100% CPU

Reduce CPU Use



1% CPU

10% higher power draw

10% CPU

2x power draw

100% CPU

10x power draw

MyApplication 〉 My Mac 64–bit

Running MyApplication : MyApplication

1

CPU Report

MyApplication
PID 7632, Paused

CPU 7%

Memory 12.7 MB

Energy Impact Low

Disk Zero KB/s

Network Zero KB/s

Thread 1
Queue: com.apple.main-thread (serial)
- 0 mach_msg_trap
- 11 NSApplicationMain
- 12 main
- 13 start

Thread 2
Queue: com.apple.libdispatch-manager (serial)

Thread 5
Queue: com.apple.root.utility-qos (concurrent)
- 0 __semwait_signal
- 2 usleep
- 3 __45-[AppDelegate applicationDidFini…
- 4 _dispatch_call_block_and_release
- 9 start_wqthread

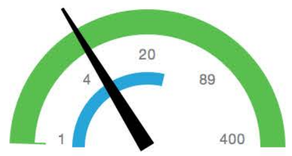
Enqueued from com.apple.main-thread (Thr…
- 0 _dispatch_async_f_slow
- 1 -[AppDelegate applicationDidFinishLa…
- 2 __CFNOTIFICATIONCENTER_IS_CALL…
- 18 NSApplicationMain
- 19 main
- 20 start

Thread 7

Thread 9

Thread 10

Percentage Used



7%

Usage Comparison

MyApplication
7%

Other Processes
70%

Free
323%

Usage over Time

Duration: 45 sec
High: 31%
Low: 0%

Threads

MyApplication 〉 Thread 5 〉 0 __semwait_signal

(lldb)

Auto

All Output

MyApplication 〉 My Mac 64–bit

Running MyApplication : MyApplication

1

CPU Report

MyApplication
PID 7632, Paused

CPU 7%

Memory 12.7 MB

Energy Impact Low

Disk Zero KB/s

Network Zero KB/s

Thread 1
Queue: com.apple.main-thread (serial)
- 0 mach_msg_trap
- 11 NSApplicationMain
- 12 main
- 13 start

Thread 2
Queue: com.apple.libdispatch-manager (serial)

Thread 5
Queue: com.apple.root.utility-qos (concurrent)
- 0 __semwait_signal
- 2 usleep
- 3 __45-[AppDelegate applicationDidFini…
- 4 _dispatch_call_block_and_release
- 9 start_wqthread

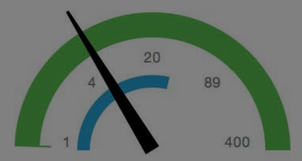
Enqueued from com.apple.main-thread (Thr…
- 0 _dispatch_async_f_slow
- 1 -[AppDelegate applicationDidFinishLa…
- 2 __CFNOTIFICATIONCENTER_IS_CALL…
- 18 NSApplicationMain
- 19 main
- 20 start

Thread 7

Thread 9

Thread 10

Percentage Used



7%

Usage Comparison

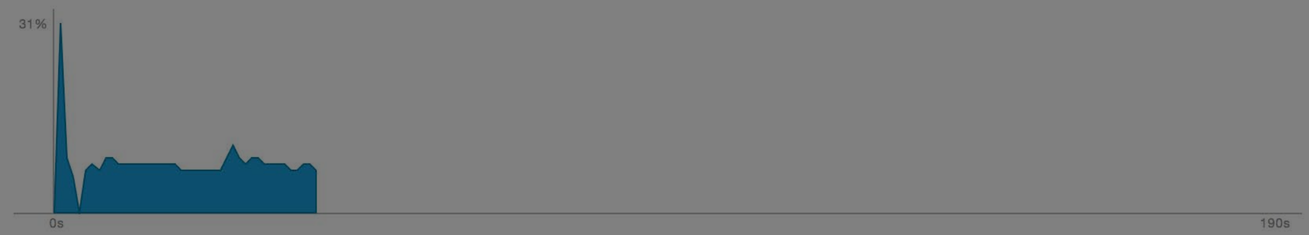
MyApplication
7%

Other Processes
70%

Free
323%

Usage over Time

Duration: 45 sec
High: 31%
Low: 0%



Threads

MyApplication 〉 Thread 5 〉 0 __semwait_signal

(lldb)

Auto

All Output

MyApplication 〉 My Mac 64–bit

Running MyApplication : MyApplication

1

CPU Report

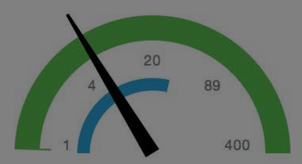
MyApplication
PID 7632, Paused

CPU	7%
Memory	12.7 MB
Energy Impact	Low
Disk	Zero KB/s
Network	Zero KB/s

- **Thread 1**
  Queue: com.apple.main-thread (serial)
  - 0 mach_msg_trap
  - 11 NSApplicationMain
  - 12 main
  - 13 start
- **Thread 2**
  Queue: com.apple.libdispatch-manager (serial)
- **Thread 5**
  Queue: com.apple.root.utility-qos (concurrent)
  - 0 __semwait_signal
  - 2 usleep
  - 3 __45-[AppDelegate applicationDidFini…
  - 4 _dispatch_call_block_and_release
  - 9 start_wqthread

  Enqueued from com.apple.main-thread (Thr…
  - 0 _dispatch_async_f_slow
  - 1 -[AppDelegate applicationDidFinishLa…
  - 2 __CFNOTIFICATIONCENTER_IS_CALL…
  - 18 NSApplicationMain
  - 19 main
  - 20 start
- **Thread 7**
- **Thread 9**
- **Thread 10**

Percentage Used



7%

Usage Comparison

- **MyApplication** 7%
- **Other Processes** 70%
- **Free** 323%

Usage over Time

Duration: 45 sec
High: 31%
Low: 0%

Threads

MyApplication 〉 Thread 5 〉 0 __semwait_signal

(lldb)

Auto

All Output

Instruments1

MacBook Pro 〉 StockWatcher.app

Run 1 of 1 00:00:18

All Cores All Processes / Threads

Instruments

Time Profiler

Time Profiler 〉 Call Tree 〉 Call Tree

Involves Symbol

Running Time		Self	Symbol Name
1906.0ms	97.7%	0.0	▼Main Thread 0x6adef
1894.0ms	97.1%	0.0	▼start libdyld.dylib
1894.0ms	97.1%	0.0	▼NSApplicationMain AppKit
1827.0ms	93.6%	0.0	▼-[NSApplication run] AppKit
1819.0ms	93.2%	0.0	▼-[NSApplication nextEventMatchingMask:untilDate:inMode:dequeue:] AppKit
1816.0ms	93.1%	0.0	▼_DPSNextEvent AppKit
1811.0ms	92.8%	0.0	▼_BlockUntilNextEventMatchingListInModeWithFilter HIToolbox
1811.0ms	92.8%	1.0	▼ReceiveNextEventCommon HIToolbox
1793.0ms	91.9%	0.0	▼RunCurrentEventLoopInMode HIToolbox
1783.0ms	91.4%	0.0	▼CFRunLoopRunSpecific CoreFoundation
1782.0ms	91.3%	0.0	▼__CFRunLoopRun CoreFoundation
1702.0ms	87.2%	0.0	▼__CFRunLoopDoObservers CoreFoundation
1699.0ms	87.1%	0.0	▼__CFRUNLOOP_IS_CALLING_OUT_TO_AN_OBSERVER_CALLBACK_FUNCTION__ CoreF
1645.0ms	84.3%	0.0	▼__83-[NSWindow _postWindowNeedsDisplayOrLayoutOrUpdateConstraintsUnlessPosting
1645.0ms	84.3%	0.0	▼_handleWindowNeedsDisplayOrLayoutOrUpdateConstraints AppKit
1622.0ms	83.1%	0.0	▼-[NSWindow displayIfNeeded] AppKit
1622.0ms	83.1%	0.0	▼-[NSView displayIfNeeded] AppKit
1608.0ms	82.4%	0.0	▶-[NSView _displayRectIgnoringOpacity:isVisibleRect:rectIsVisibleRectForView:] Ap
14.0ms	0.7%	0.0	▶-[NSView _sendViewWillDrawInRect:clipRootView:] AppKit
20.0ms	1.0%	0.0	▶_CFAutoreleasePoolPop CoreFoundation
2.0ms	0.1%	0.0	▶CA::Transaction::commit() QuartzCore
1.0ms	0.0%	1.0	-[NSWindow isVisible] AppKit
50.0ms	2.5%	0.0	▶__35-[NSWindow _postInvalidCursorRects]_block_invoke3230 AppKit
2.0ms	0.1%	0.0	▶__38-[NSApplication setWindowsNeedUpdate:]_block_invoke2481 AppKit
1.0ms	0.0%	0.0	▶-[NSWindow(NSDrag) _registerDragTypes:] AppKit

Sample Perspective

- All Sample Counts
- Running Sample Times

Call Tree

- [x] Separate by Thread
- [ ] Invert Call Tree
- [ ] Hide Missing Symbols
- [ ] Hide System Libraries
- [ ] Flatten Recursion
- [ ] Top Functions

Call Tree Constraints

- [ ] Count 0 ∞
- [ ] Time (ms) -∞ ∞

Data Mining

Symbol Library Restore

Performance Unit Tests

Performance Unit Tests

Additions to XCTestCase API in Xcode 6

Performance Unit Tests

Additions to XCTestCase API in Xcode 6

Helps you find performance regressions

Performance Unit Tests

Additions to XCTestCase API in Xcode 6

Helps you find performance regressions

```
- (void)testSomething
{
    [self measureBlock:^{
        // ...snip..
    }];
}
```

Related Sessions

• Testing in Xcode 6	Marina	Thursday 9:00AM
• Continuous Integration with Xcode 6	Marina	Thursday 2:00PM

Reduce CPU Use

Reduce CPU Use

CPU use has a huge dynamic range in power

Reduce CPU Use

CPU use has a huge dynamic range in power

Monitor CPU use with Xcode debug gauge

Reduce CPU Use

CPU use has a huge dynamic range in power

Monitor CPU use with Xcode debug gauge

Profile with Instruments

Reduce CPU Use

CPU use has a huge dynamic range in power

Monitor CPU use with Xcode debug gauge

Profile with Instruments

Prevent regressions with performance unit tests

Minimize Timers

Minimize Timers

-[NSObject performSelector:withObject:afterDelay:]

pthread_cond_timedwait()

CFRunLoopTimer

NSTimer

Grand Central Dispatch timers

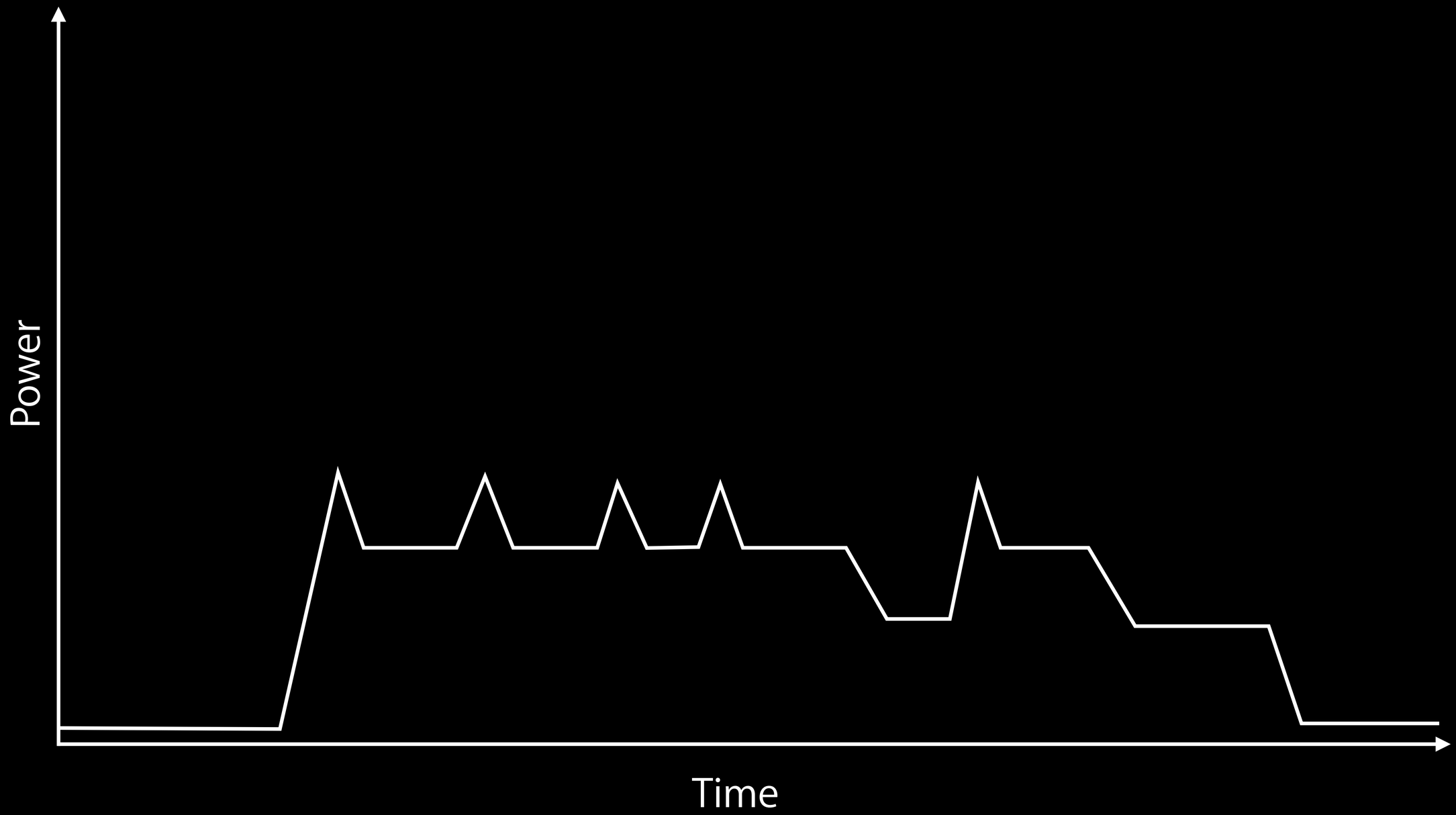
sleep()

select()

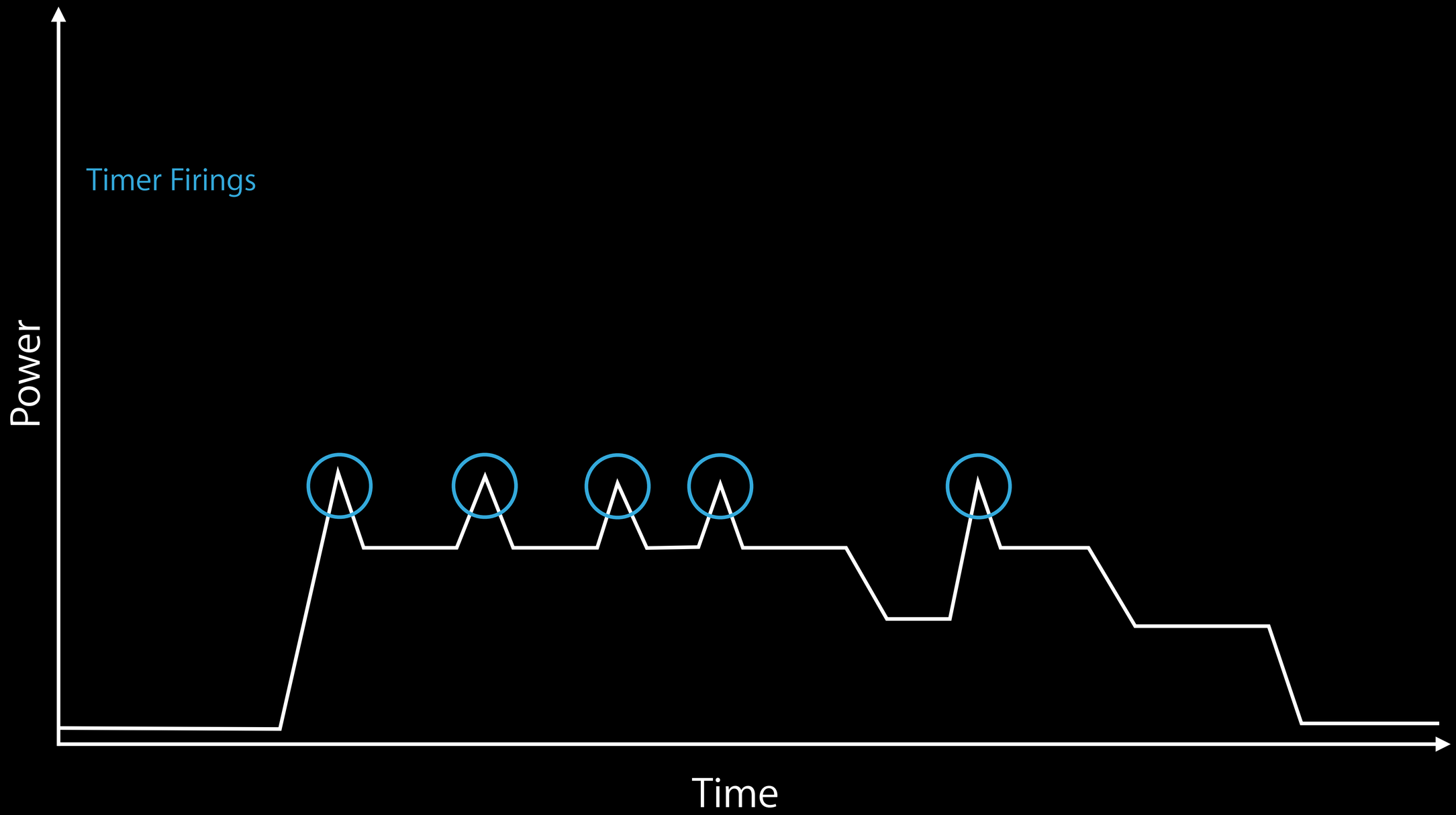
CVDisplayLink

dispatch_semaphore_wait()

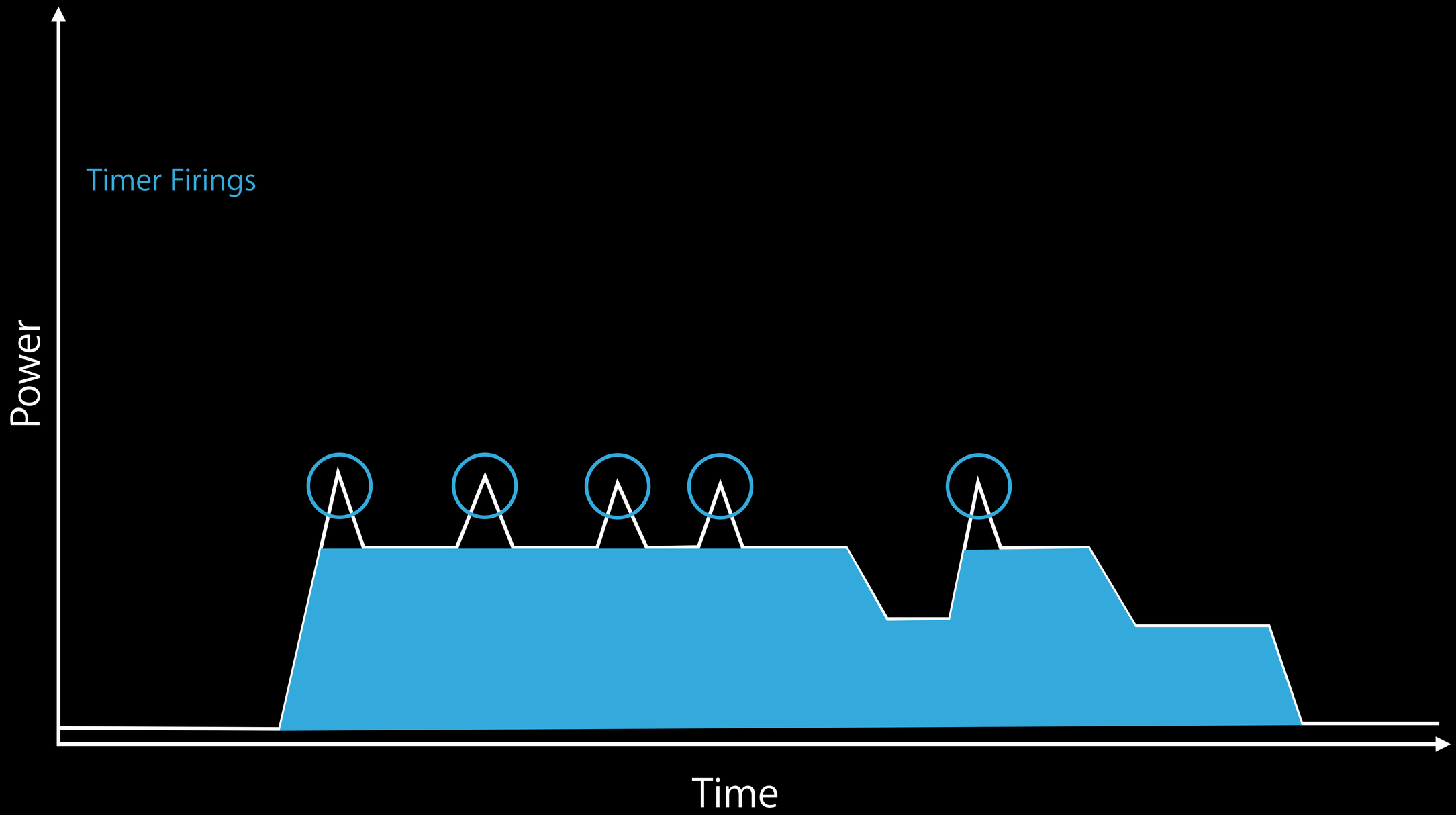
Timer Fixed Cost



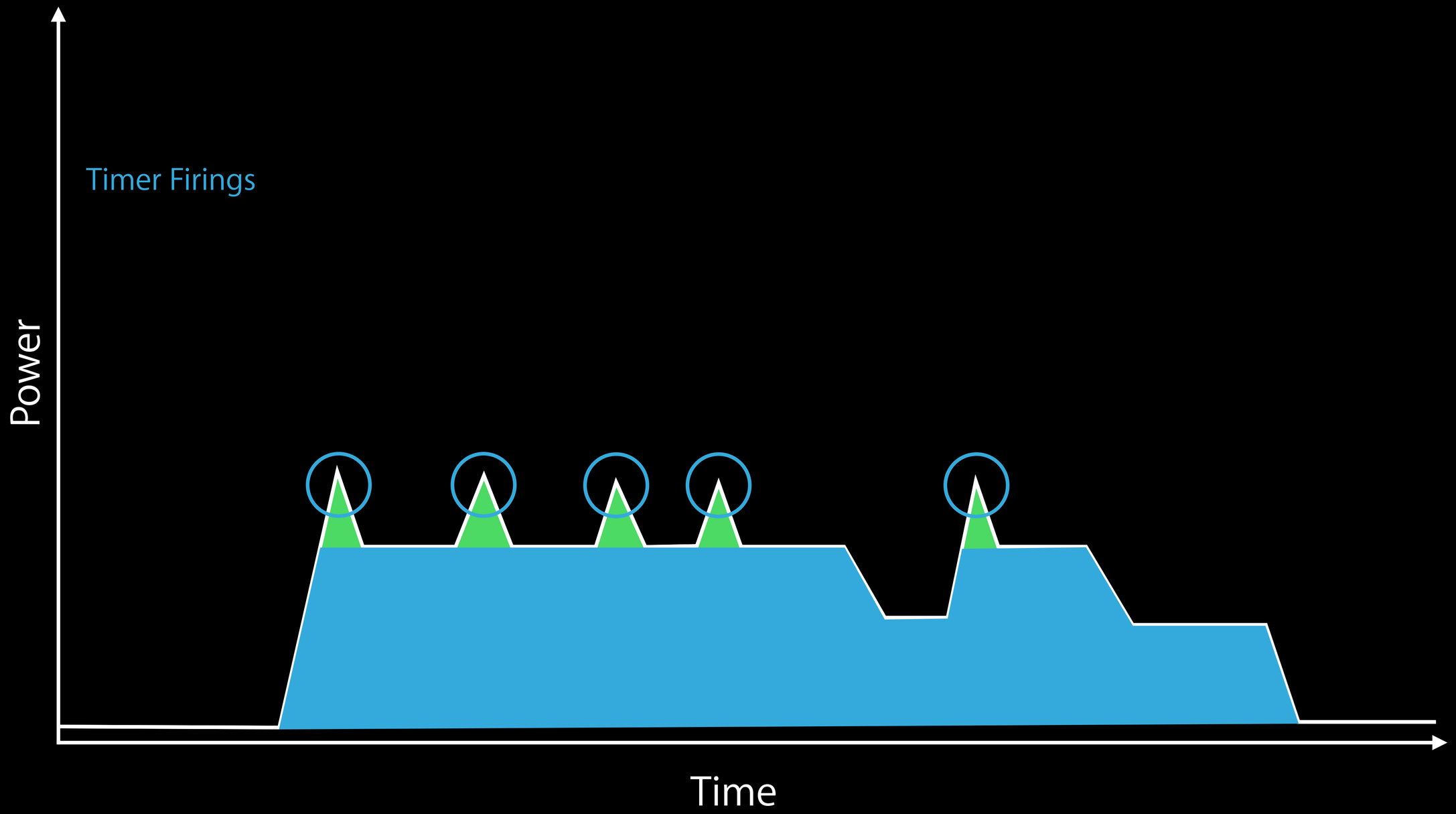
Timer Fixed Cost



Timer Fixed Cost



Timer Fixed Cost



My Mac 64–bit | Running MyApplication : MyApplication

Energy Report

MyApplication
PID 9197, Running

CPU	95%
Memory	10.4 MB
Energy Impact	High
Disk	Zero KB/s
Network	Zero KB/s

Energy

Utilization

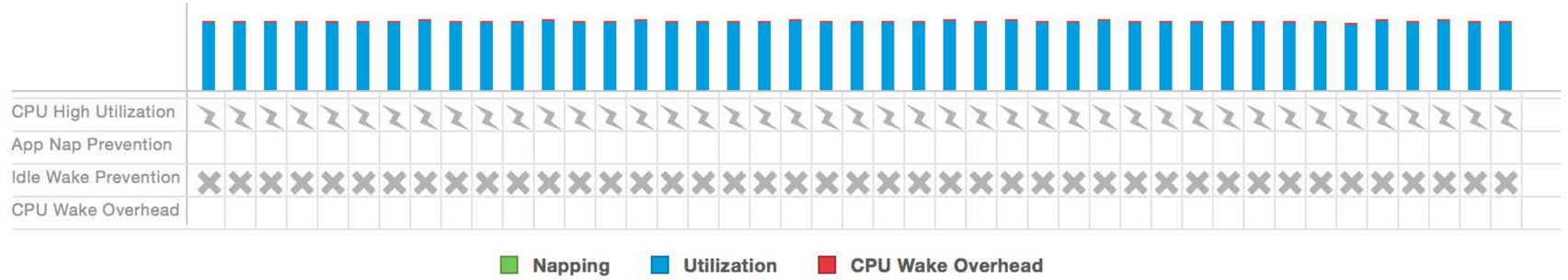
High
Energy Impact

Wakes and CPU

33 Wakes
Last Second

32 average per second

Energy Impact



MyApplication

My Mac 64–bit | Running MyApplication : MyApplication

Energy Report

MyApplication
PID 9197, Running

- **CPU** 95%
- **Memory** 10.4 MB
- **Energy Impact** High
- **Disk** Zero KB/s
- **Network** Zero KB/s

Energy

Utilization

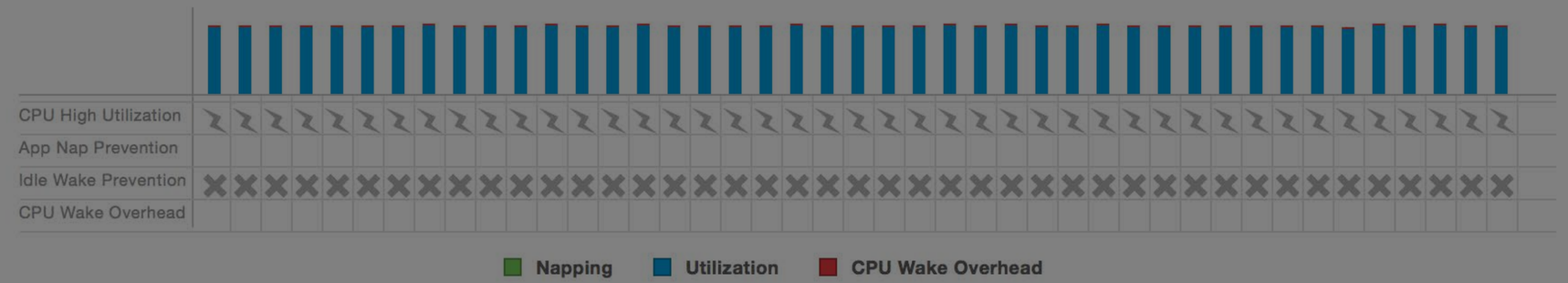
High
Energy Impact

Wakes and CPU

33 Wakes
Last Second

32 average per second

Energy Impact



MyApplication

My Mac 64–bit | Running MyApplication : MyApplication

Energy Report

MyApplication
PID 9197, Running

CPU	95%
Memory	10.4 MB
Energy Impact	High
Disk	Zero KB/s
Network	Zero KB/s

Energy

Utilization

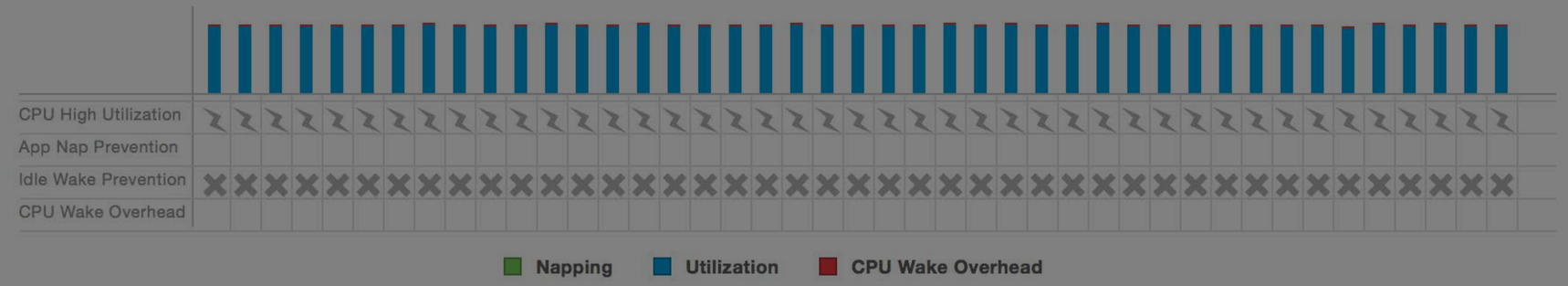
High
Energy Impact

Wakes and CPU

33 Wakes
Last Second

32 average per second

Energy Impact



MyApplication

Diagnosing Timer Issues

timerfires

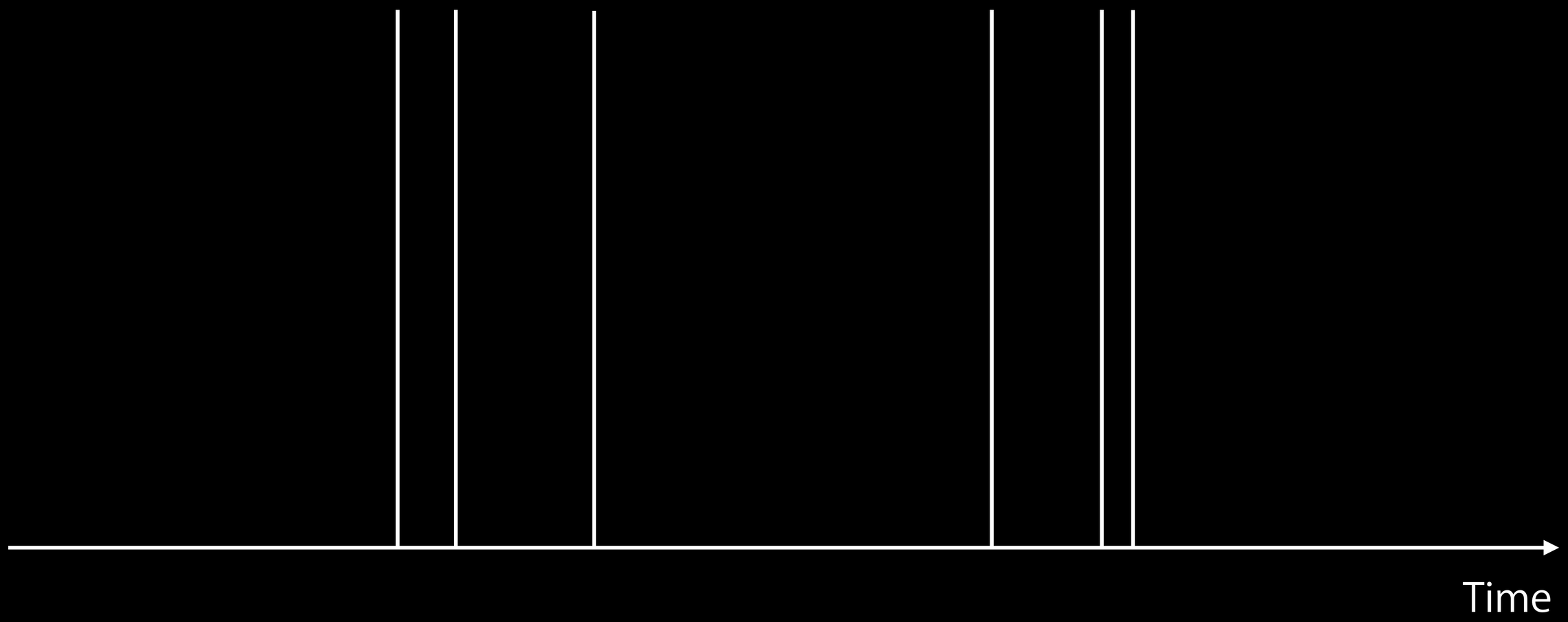
```
$ sudo timerfires -p MyApplication -s -g
```

Diagnosing Timer Issues

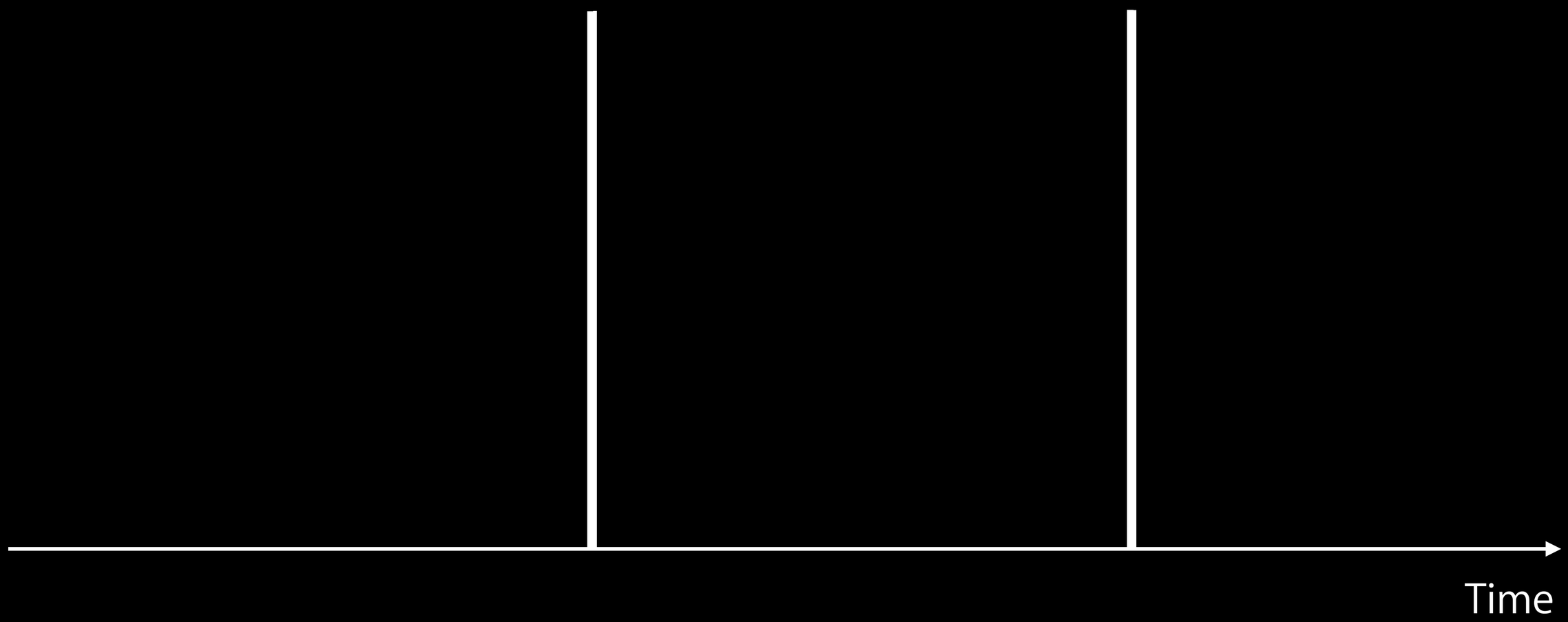
timerfires

```
$ sudo timerfires -p MyApplication -s -g
  COUNT   PID   PROCESS  TYPE      TIMER ROUTINE
     555  1603  MyApp    dispatch  MyApp`-[AppModel updateWidgets:]
     735  1603  MyApp    CF        MyApp`-[AppDelegate timerFired:]
    1127  1603  MyApp    sleep
          libsystem_kernel.dylib`__semwait_signal+0xa
          libsystem_c.dylib`usleep+0x36
          MyApp`-[AppModel pollForChange]+0x1a
          libdispatch.dylib`_dispatch_call_block_and_release+0xc
```

Timer Coalescing



Timer Coalescing



Specify Timer Tolerance

Specify Timer Tolerance

```
[myTimer setTolerance:60.0];

CFRunLoopTimerSetTolerance(myTimer, 60.0);

dispatch_source_set_timer(my_timer, DISPATCH_TIME_NOW,
                          30 * NSEC_PER_SEC, 60 * NSEC_PER_SEC);
```

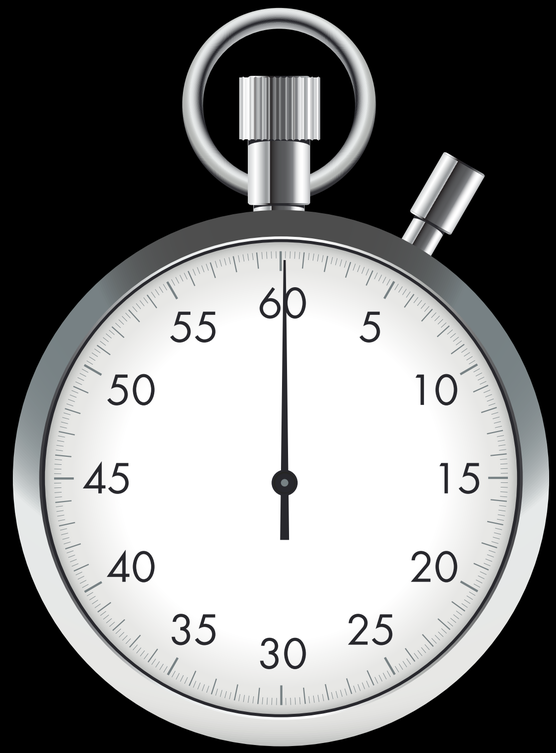
Minimize Timers

Be mindful of wakeup overhead

Monitor for wakeups

Debug with timerfires

Specify timer tolerance



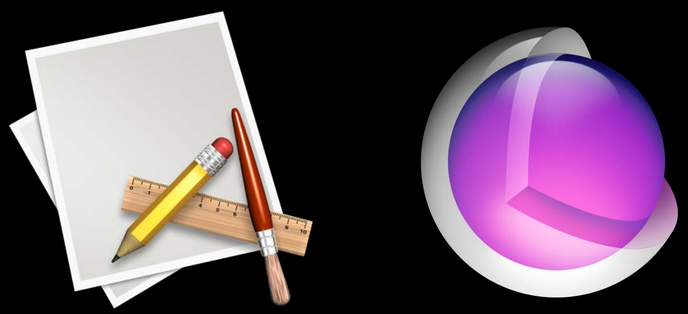
Related Session

• Energy Best Practices	WWDC 2013

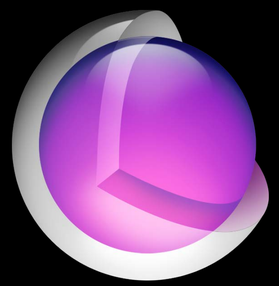
Efficient Graphics

Efficient Graphics

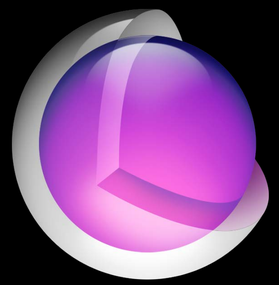
Efficient Graphics



Efficient Graphics



Efficient Graphics



Limit Screen Updates

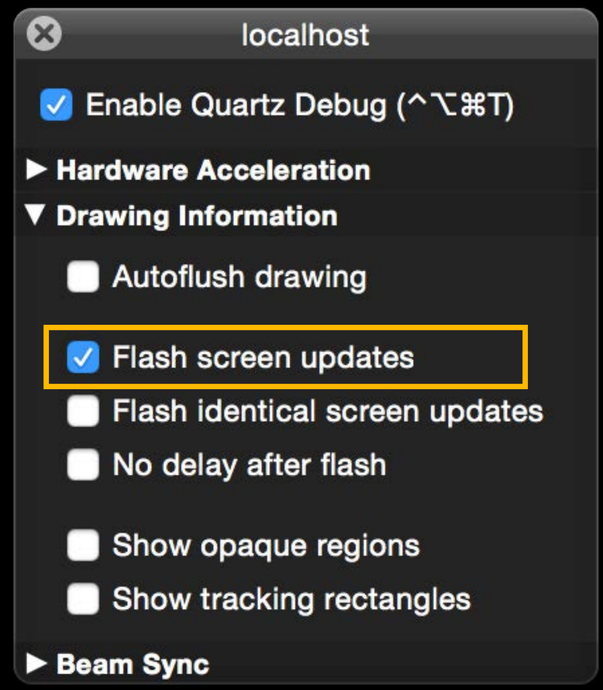
Avoid extraneous screen updates

Unnecessary drawing kicks graphics hardware out of low-power modes

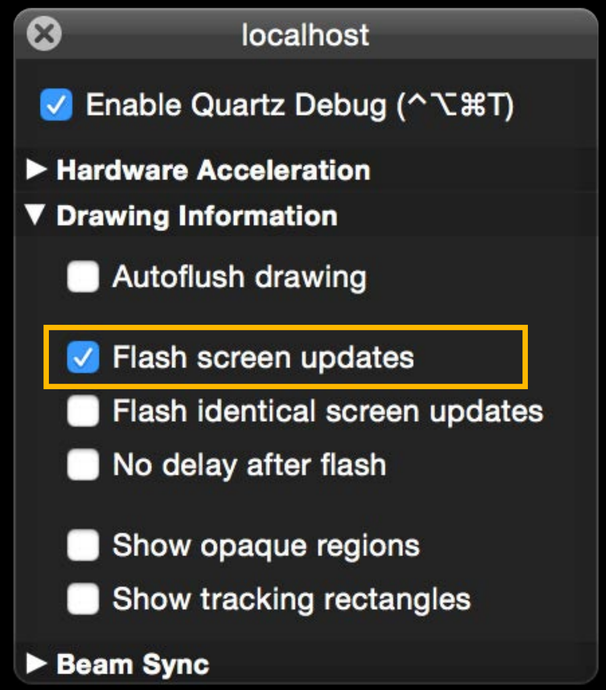
Drawing more content than needed causes extra power draw to update the screen

- Use `needsToDrawRect:` or `getRectsBeingDrawn:count:` methods to fine-tune drawing

Flash Screen Updates



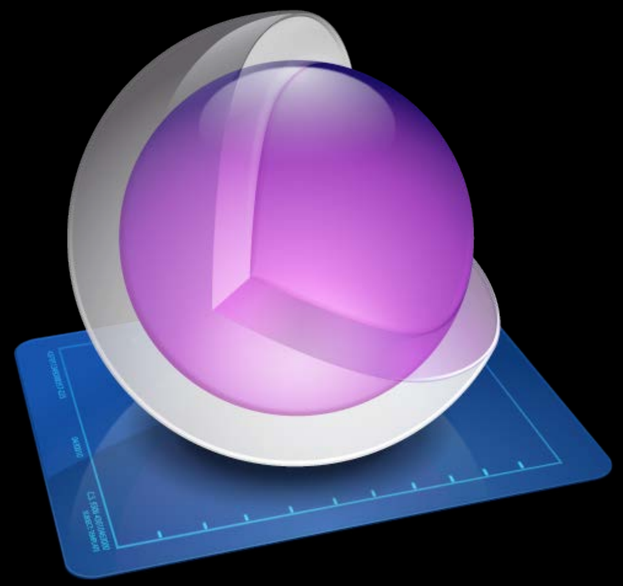
Flash Screen Updates



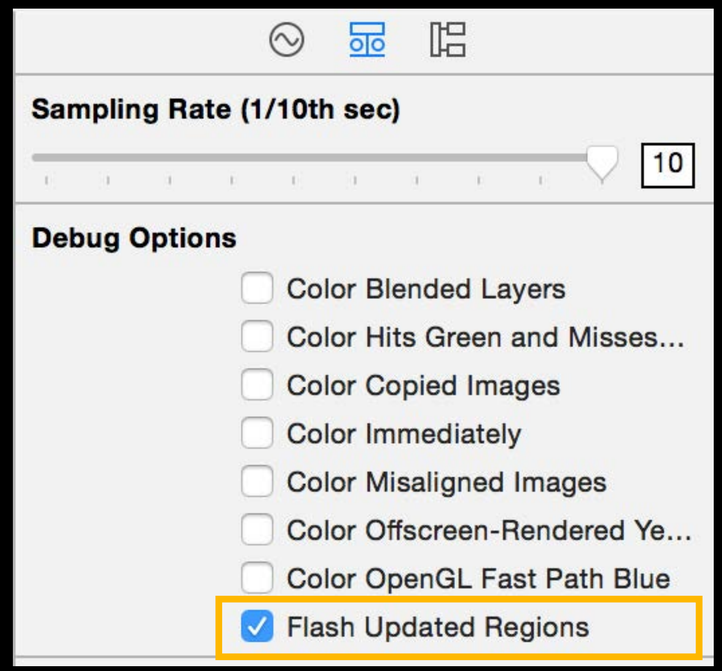
Flash Screen Updates

Graphics Tools for Xcode: http://developer.apple.com

Flash Screen Updates



Core Animation



Visual Effects

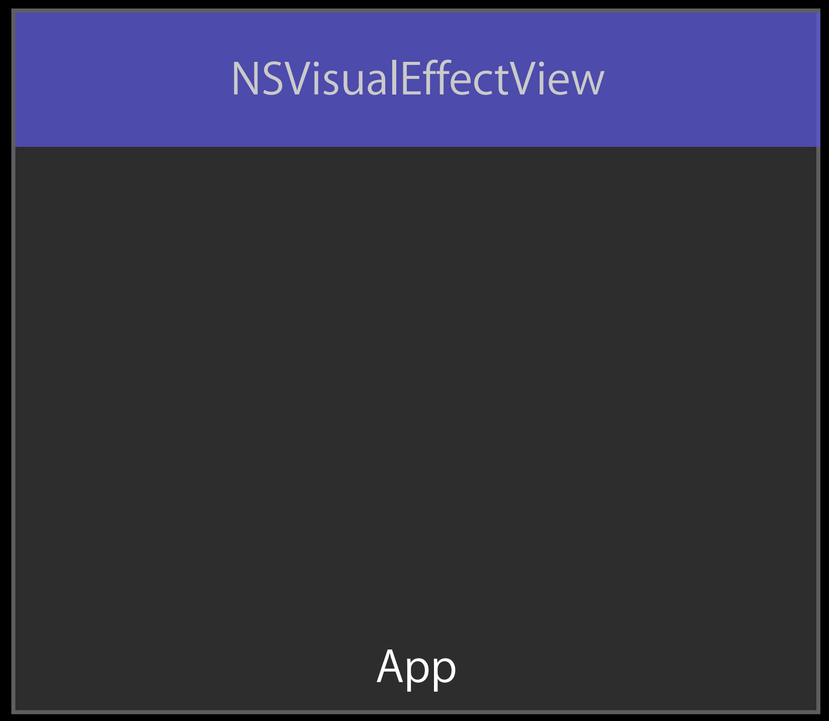
Translucent blurs have an energy cost

Avoid placing over updating elements

Visual Effects

Translucent blurs have an energy cost

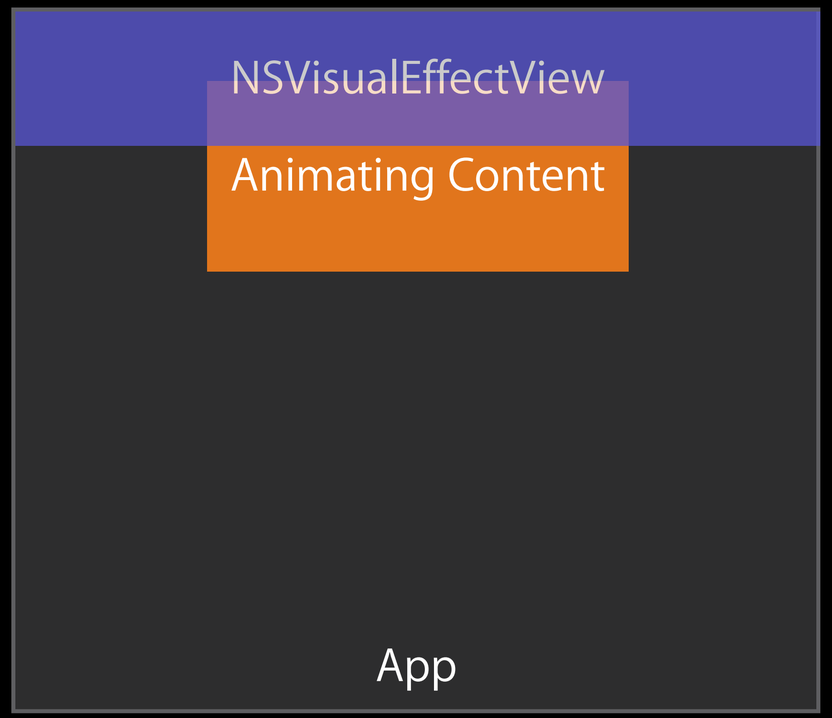
Avoid placing over updating elements



Visual Effects

Translucent blurs have an energy cost

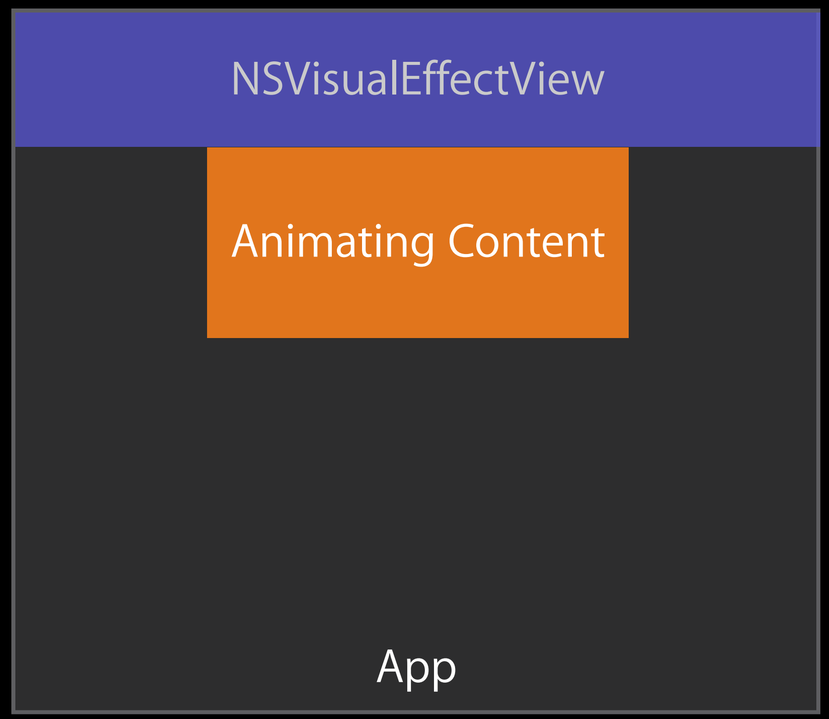
Avoid placing over updating elements



Visual Effects

Translucent blurs have an energy cost

Avoid placing over updating elements



Efficient Graphics

Draw minimally and efficiently

Monitor drawing with Quartz Debug or Instruments

Avoid blurs on updating content

Flash Power

Writes to Flash are much more energy hungry than reads

- Write the minimum content necessary
- Do writes in aggregate for better power efficiency

Flash Power

Writes to Flash are much more energy hungry than reads

- Write the minimum content necessary
- Do writes in aggregate for better power efficiency

Any I/O will pull device out of low-power states

- Use caching to your advantage

Do It Less

Profile and monitor CPU

Reduce timers

Be efficient in the use of graphics

Minimize I/O

Summary

Improving your app's energy consumption improves user experience

Summary

Improving your app's energy consumption improves user experience

Continuously monitor your app's energy and resource consumption

Summary

Improving your app's energy consumption improves user experience

Continuously monitor your app's energy and resource consumption

Look for ways to

- Do it never—Respond to changes in active state

Summary

Improving your app's energy consumption improves user experience

Continuously monitor your app's energy and resource consumption

Look for ways to

- Do it never—Respond to changes in active state
- Do it at a better time—Let the system schedule work

Summary

Improving your app's energy consumption improves user experience

Continuously monitor your app's energy and resource consumption

Look for ways to

- Do it never—Respond to changes in active state
- Do it at a better time—Let the system schedule work
- Do it more efficiently—Specify Quality of Service Classes on your work

Summary

Improving your app's energy consumption improves user experience

Continuously monitor your app's energy and resource consumption

Look for ways to

- Do it never—Respond to changes in active state
- Do it at a better time—Let the system schedule work
- Do it more efficiently—Specify Quality of Service Classes on your work
- Do it less—Optimize and improve your resource use

Summary

Improving your app's energy consumption improves user experience

Continuously monitor your app's energy and resource consumption

Look for ways to

- Do it never—Respond to changes in active state
- Do it at a better time—Let the system schedule work
- Do it more efficiently—Specify Quality of Service Classes on your work
- Do it less—Optimize and improve your resource use

And stick around for Part 2

More Information

Paul Danbold
Core OS Evangelist
danbold@apple.com

Energy Best Practices
WWDC 2013

Building Resource Efficient Apps
WWDC 2013

Apple Developer Forums
http://devforums.apple.com

Related Sessions

What's New in Foundation Networking	Nob Hill	Tuesday 3:15PM
Improving Your App with Instruments	Marina	Tuesday 4:30PM
Writing Energy Efficient Code, Part 2	Russian Hill	Wednesday 11:30AM
Testing in Xcode 6	Marina	Thursday 9:00AM
Fix Bugs Faster Using Activity Tracing	Russian Hill	Thursday 11:30AM
Continuous Integration with Xcode 6	Marina	Thursday 2:00PM
Power, Performance, and Diagnostics: What's New in GCD and XPC	Russian Hill	Thursday 2:00PM

Labs

• Power and Performance Lab	Core OS Lab B	Wednesday 2:00PM
• Instruments Lab	Tools Lab B	Thursday 9:00AM
• Power and Performance Lab	Core OS Lab A	Thursday 3:15PM

WWDC14