

# "GREEN" DEVELOPMENT: IS IT A THING?



**ALEKSANDRA KOMAGORKINA**



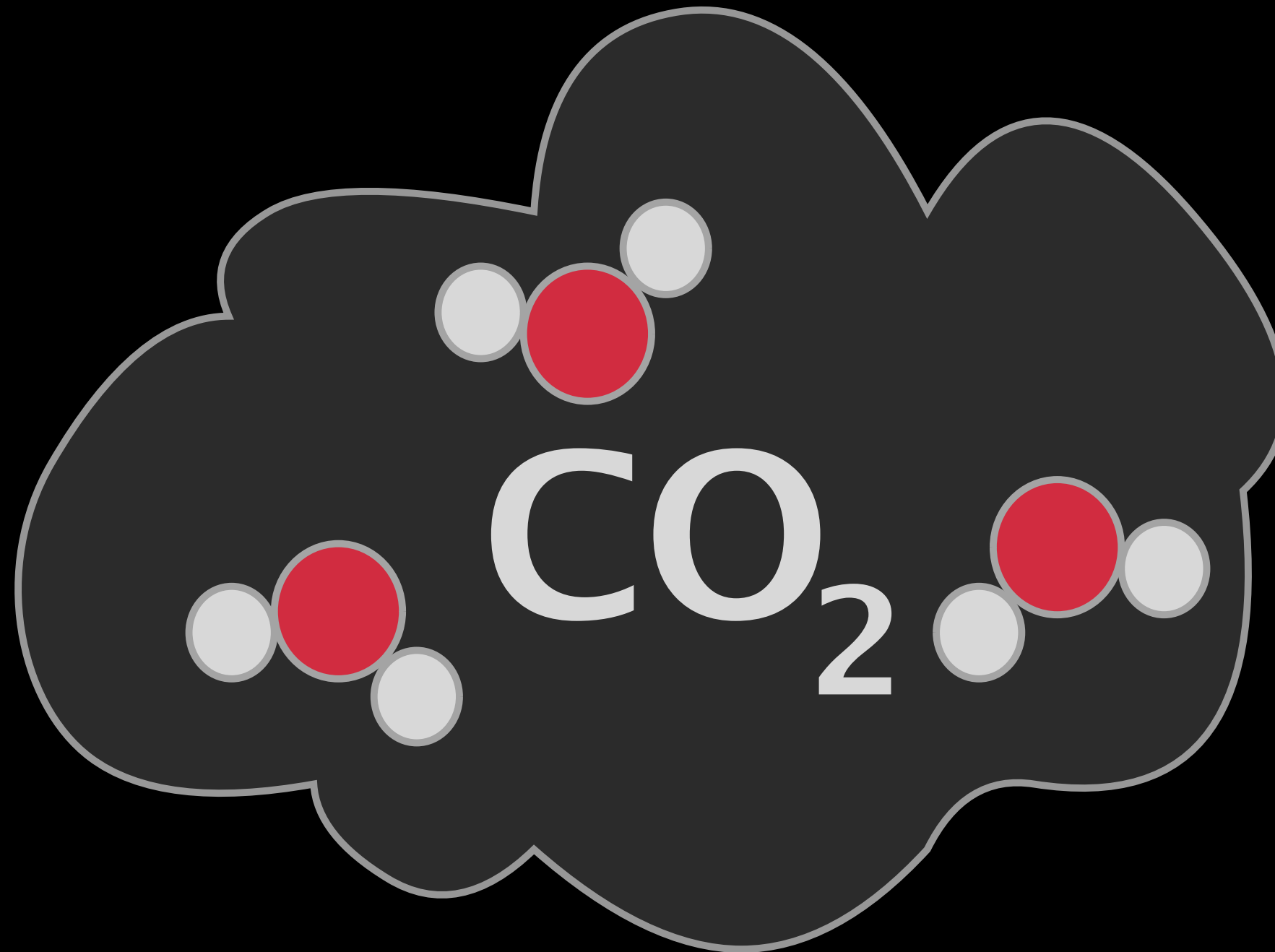
**@AKOMAGORKINA**

"Green development is a real estate development concept that carefully considers social and environmental impacts of development."

— Wikipedia



# WHAT ARE THESE EMISSIONS?



To obtain 1kWh from coal or fuel, 800g of CO<sub>2</sub> will be rejected in the atmosphere during combustion of fossil fuel

– J. Bernard, Sciences et vie 214 (2001) 68

**HOW MUCH ENERGY DO  
YOU NEED TO CHARGE  
YOUR IPHONE?**

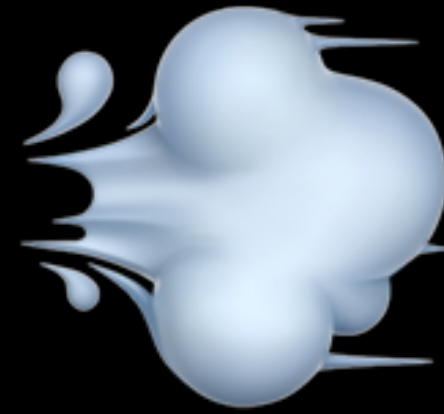
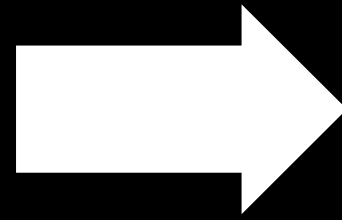
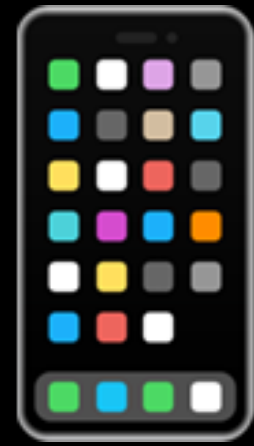
**iPhone 11 Pro Max battery holds a charge of \* 3500 mAh. If you fully drain and recharge your phone everyday, then over a year you would have to feed it about 5489 watt hours, or 5.5 kWh.**

**— Forbes**

**In January 2019, 900 million iPhones  
were in use worldwide**

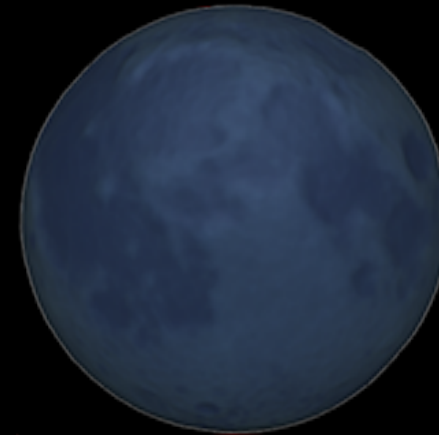
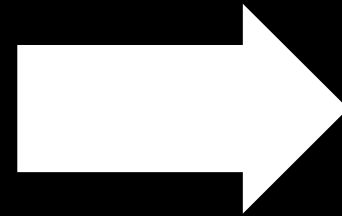
**— [macrumors.com](https://www.macrumors.com)**





**5.5 kWh**

**4.4 kg**



**0.9  
billion people**

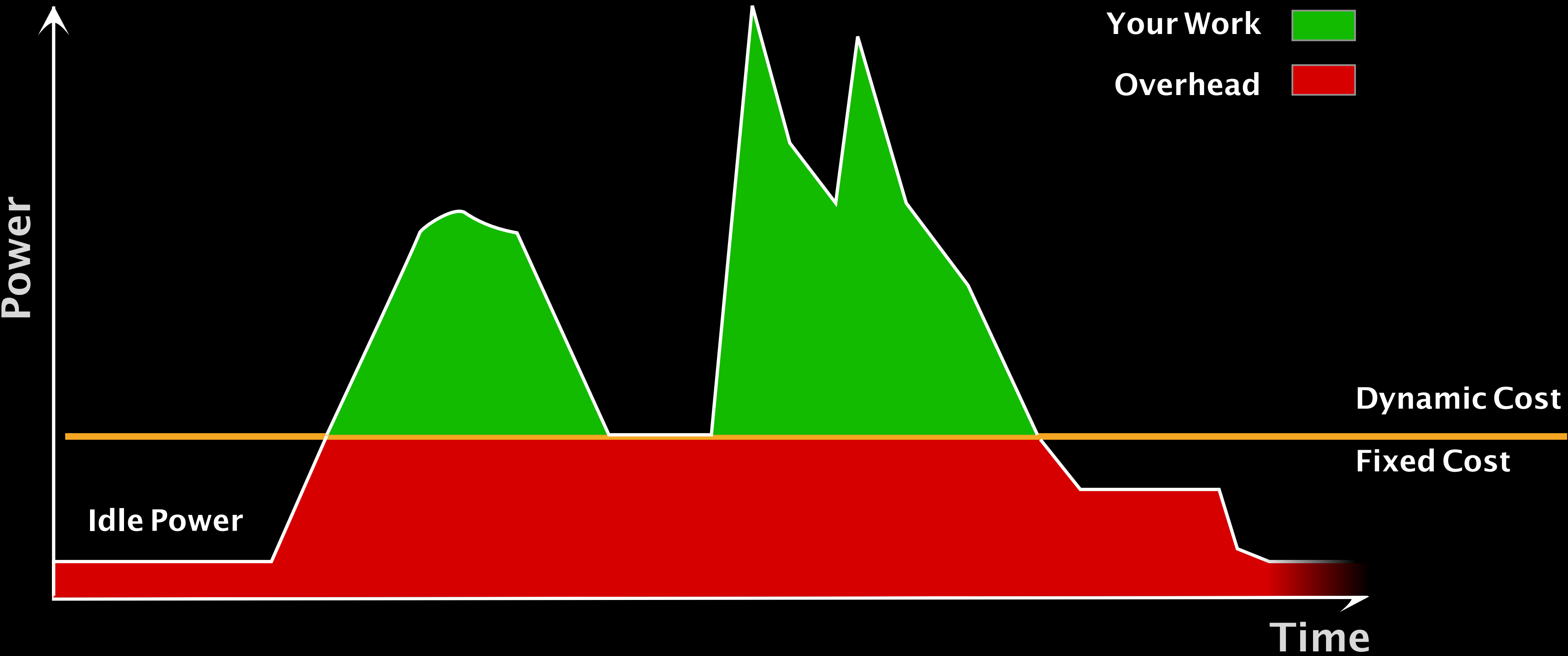
**3.96  
million tons**

# WHAT CAN WE DO?

**UNDERSTAND HOW  
ENERGY IS USED ON THE  
IPHONE**

# SPU

# DEVICE WAKE



# NETWORKING OPERATIONS

# GRAPHICS, ANIMATIONS, AND VIDEO

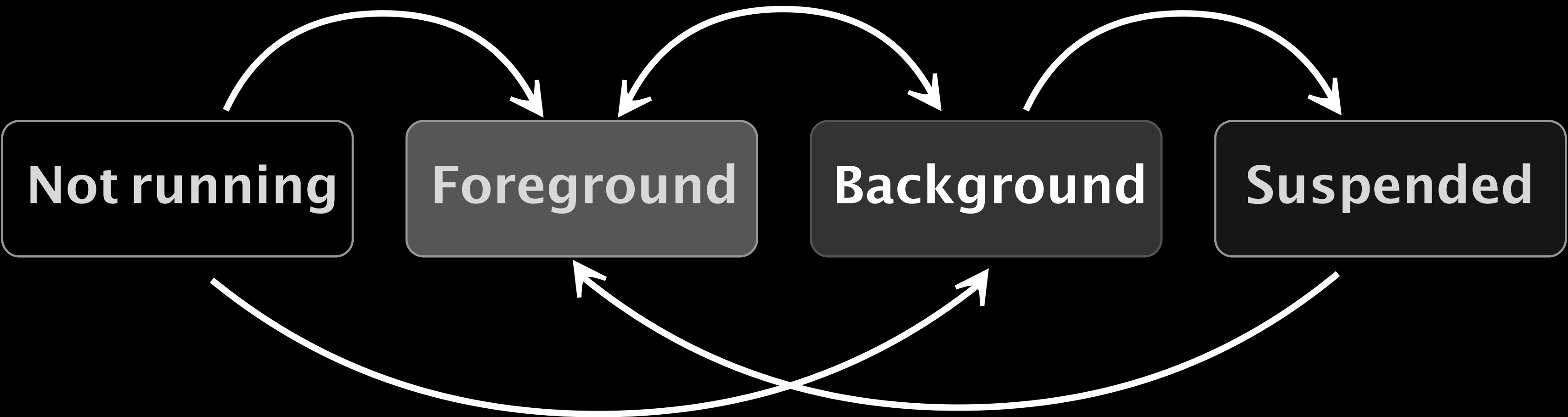


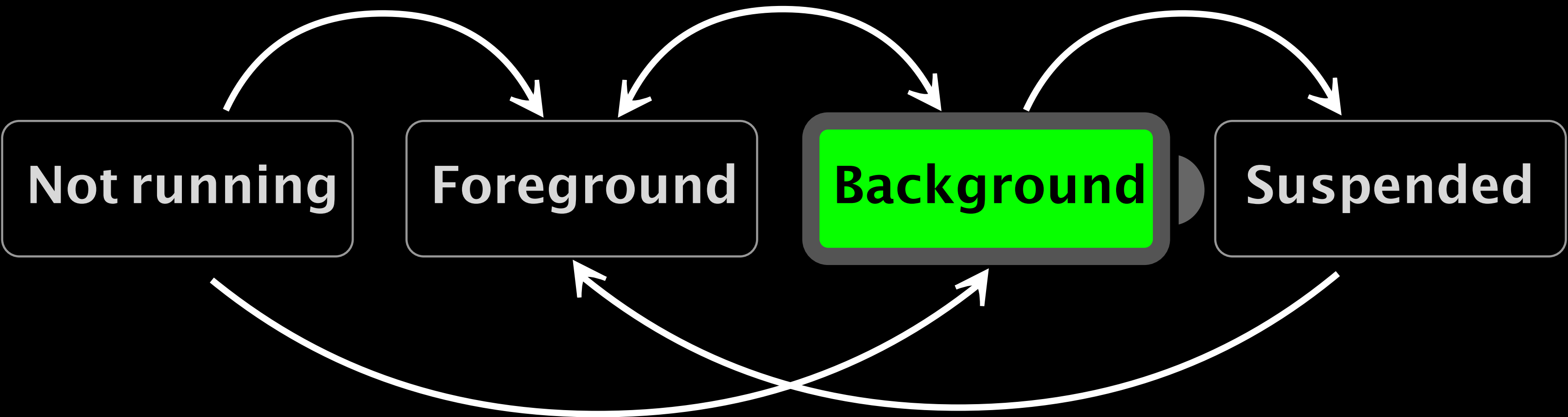
# LOCATION

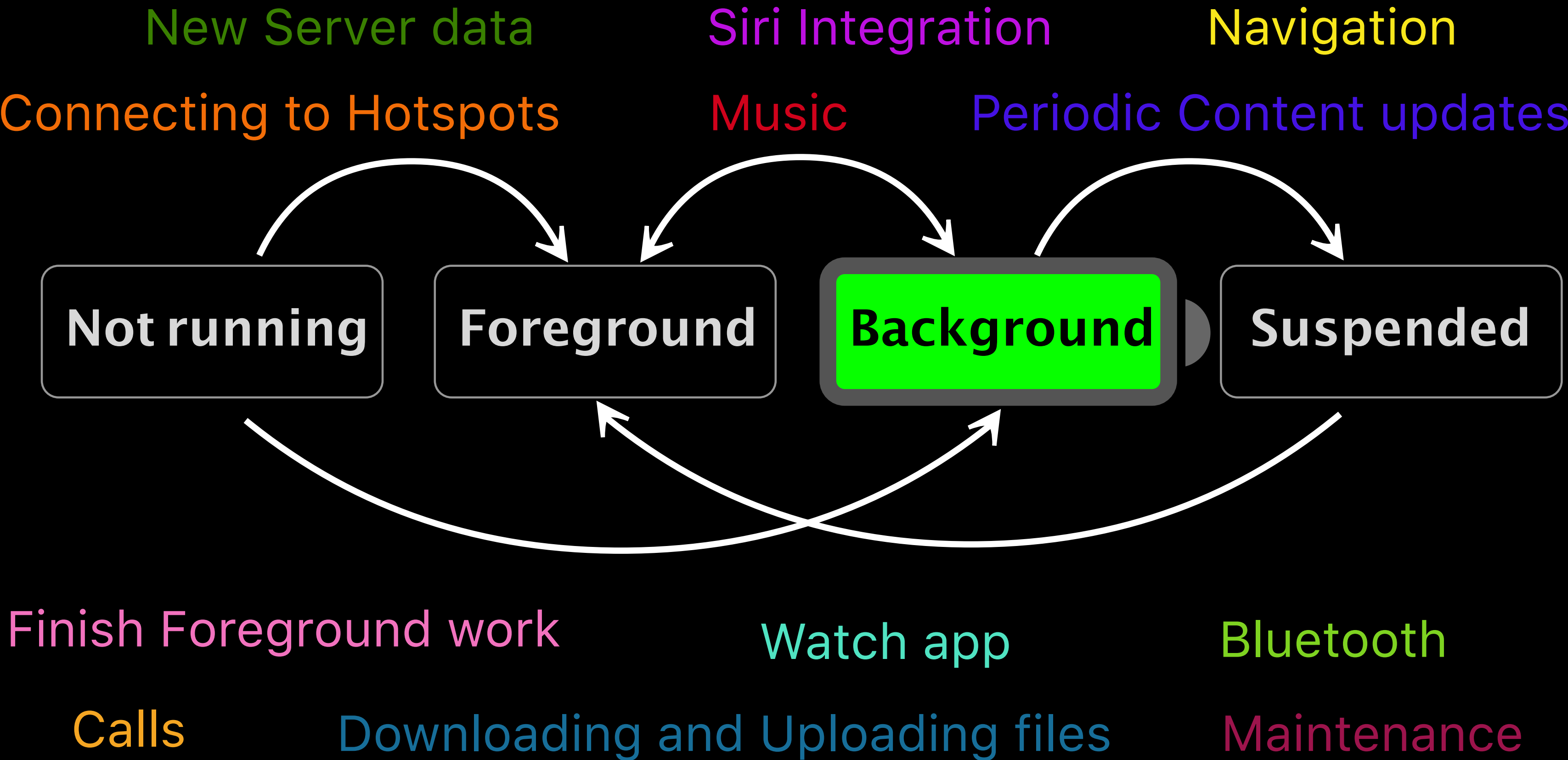
# MOTION

# BLUE TOOTH

# BACKGROUND







# ENERGY WASTE IN BACKGROUND

- ▶ Not notifying the system when background activity is complete
- ▶ Playing silent audio
- ▶ Performing location updates
- ▶ Interacting with Bluetooth accessories
- ▶ Downloads that could be deferred



# WHAT CAN WE DO?

# COMPLETE BACKGROUND TASKS

# BACKGROUND TASK COMPLETION

- ▶ **User expects immediate completion**
- ▶ **Protect completion**

**=> Give the app additional time to run in the background before being suspended**

# BACKGROUND TASK COMPLETION

```
UIApplication.beginBackgroundTask(expirationHandler:)
```

```
ProcessInfo.performExpiringActivity(withReason:using:)
```

# BACKGROUND TASK COMPLETION

```
// Guarding Important Tasks While App is Still in the Foreground
func send(_ message: Message) {
    let sendOperation = SendOperation(message: message)
    var identifier: UIBackgroundTaskIdentifier!
    identifier = UIApplication.shared.beginBackgroundTask(expirationHandler: {
        sendOperation.cancel()
        postUserNotification("Message not sent, please resend")
        // Background task will be ended in the operation's completion block below
    })
    sendOperation.completionBlock = {
        UIApplication.shared.endBackgroundTask(identifier)
    }
    operationQueue.addOperation(sendOperation)
}
```

**DEFER THE DOWNLOAD UNTIL THE BETTER TIME**

# DISCRETIONARY BACKGROUND URL SESSION

```
// Set up background URL session
let config = URLSessionConfiguration.background(withIdentifier: "com.app.attachments")
let session = URLSession(configuration: config, delegate: ..., delegateQueue: ...)

// Set discretionary
config.discretionary = true
```

# DISCRETIONARY BACKGROUND URL SESSION

```
// Set timeout intervals
config.timeoutIntervalForResource = 24 * 60 * 60
config.timeoutIntervalForRequest = 60

// Create request and task
var request = URLRequest(url: url)
request.addValue("...", forHTTPHeaderField: "...")
let task = session.downloadTask(with: request)

// Set time window
task.earliestBeginDate = Date(timeIntervalSinceNow: 2 * 60 * 60)

// Set workload size
task.countOfBytesClientExpectsToSend = 160
task.countOfBytesClientExpectsToReceive = 4096
task.resume()
```



# ALSO IN BACKGROUND: BACKGROUNDTASKS FRAMEWORK

 Background Processing Tasks

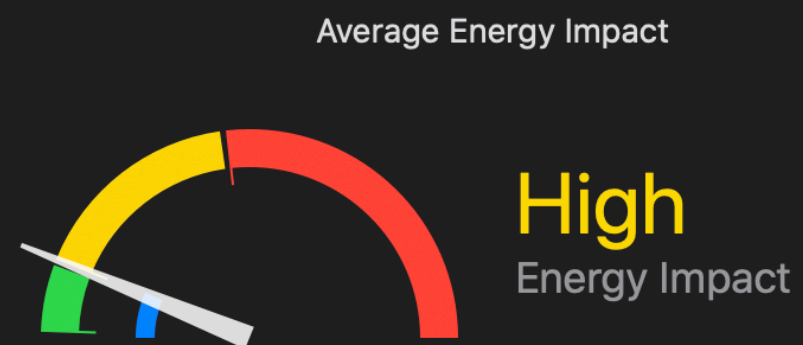
 Background App Refresh Task

MORE: ADVANCES IN APP BACKGROUND EXECUTION, WWDC2019

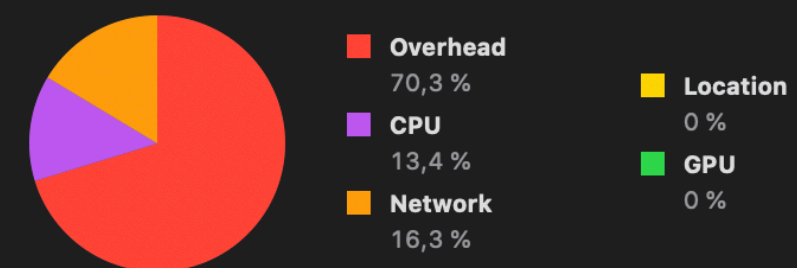
**IF YOU CAN MEASURE IT – YOU CAN MANAGE IT**

# MONITOR ENERGY USAGE IN THE DEBUGGER

## Energy

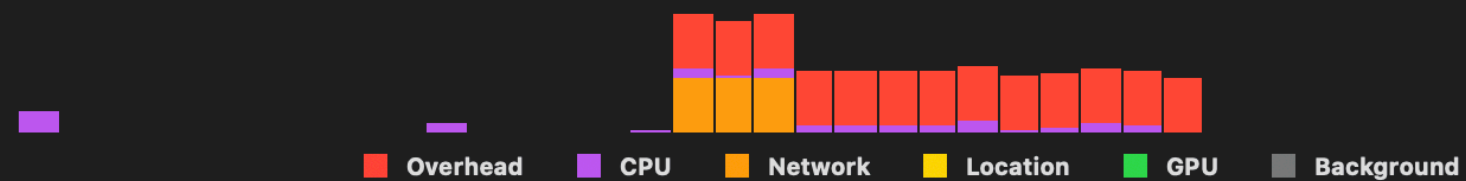


### Average Component Utilization



## Energy Impact

## Component Utilization



## Application State

Foreground

■ Background   ■ Foreground   ■ Suspended

## Thermal State

No device condition active

Nominal

■ Nominal   ■ Fair   ■ Serious   ■ Critical

## Overhead

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

## Energy

Average Energy Impact



**Low**  
Energy Impact

# XCODE INSTRUMENTS FOR OPTIMISATION

# MEASURE ENERGY IMPACT WITH NEW INSTRUMENTS:

- ▶ **XCTest Metrics**  
Performance of measure blocks
- ▶ **MetricKit**  
Framework for battery and performance metrics collection
- ▶ **Xcode Metrics Organizer**  
Aggregated battery, performance, and I/O metrics in Xcode

# COLLECTING METRICS USING XCTEST

```
// This test measures Launch Time
func testAppLaunchTime() {
    measure(metrics: [XCTOSSignpostMetric.applicationLaunch]) {
        XCUIApplication().launch()
    }
}
```



# ADOPTING METRICKIT TO RECEIVE METRICS

```
import MetricKit
// Conform to MXMetricManagerSubscriber protocol
extension AppDelegate: MXMetricManagerSubscriber {

    func subscribeToMetrics() { // Call from didFinishLaunching...
        let shared = MXMetricManager.shared
        shared.add(self)
    }

    // Receive daily metrics
    func didReceive(_ payloads: [MXMetricPayload]) {
        // Process metrics
    }
}
```

# WHAT ELSE CAN WE DO BETTER?

# ADOPT LOW DATA MODE

# **SUGGESTED TECHNIQUES FOR CONFORMING TO LOW DATA MODE:**

- ▶ **Reduce image quality**
- ▶ **Reduce pre-fetching (of unused or rarely used resources)**
- ▶ **Sync less often using locally cached data more heavily.**

# **SUGGESTED TECHNIQUES FOR CONFORMING TO LOW DATA MODE:**

- ▶ **Mark background tasks as discretionary**
- ▶ **Disable auto-play**
- ▶ **Do not block user-initiated work, even though low data mode is on.**

**AND WE CAN DO EVEN MORE**

9:41

1x



## Temperature




iPhone needs to cool down  
before you can use it.

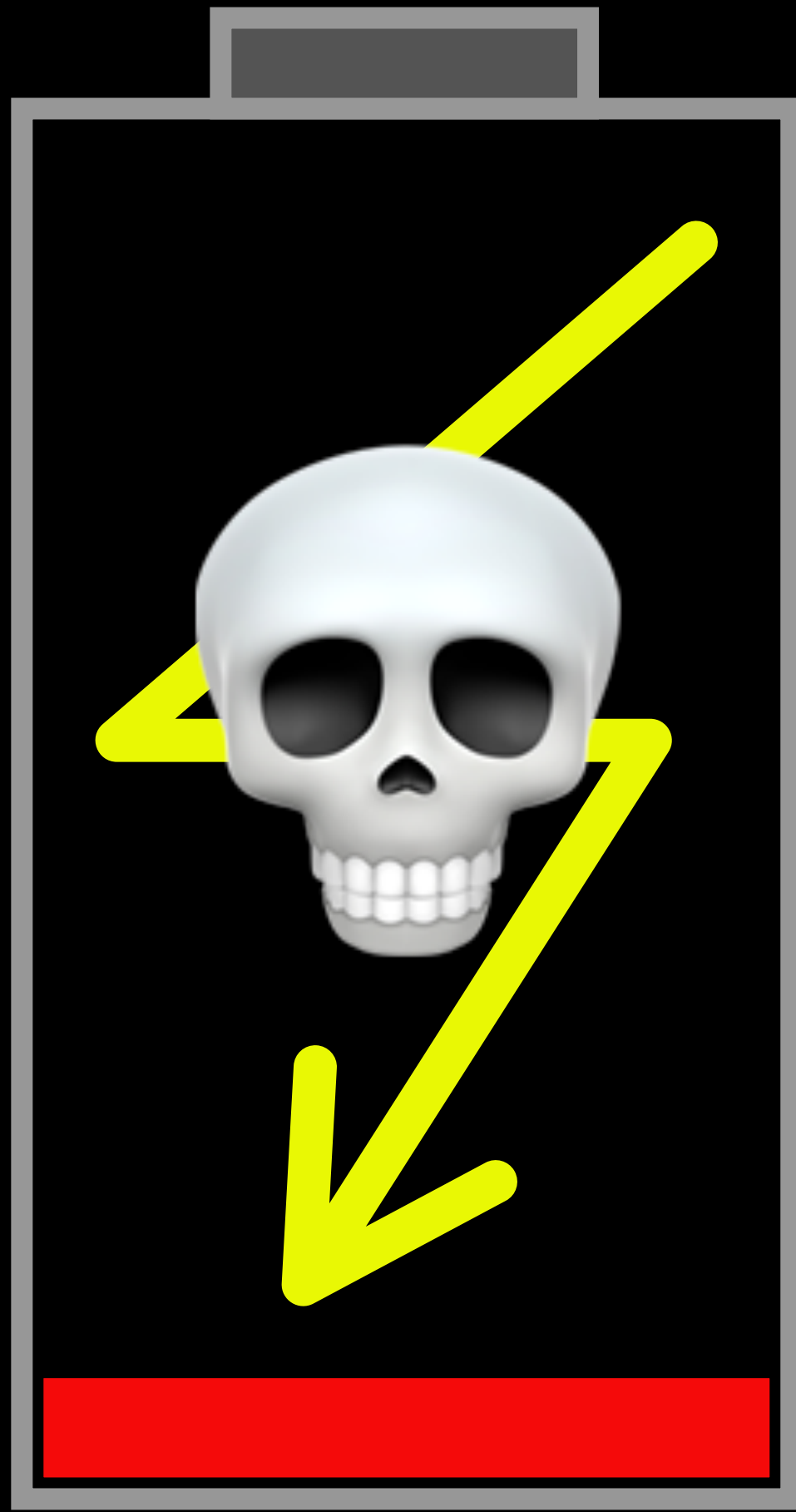
Emergency

# UNDERSTANDING DEVICE CONTIDIONS



# HOW DOES THE HEAT DAMAGE THE BATTERIES?

- ▶  Hot temperatures can cause permanent damage to batteries.
- ▶  Components like the voltage indicator can be affected by heat
- ▶  As batteries heat up, chemical reactions inside will also occur faster



SAVE  
NAME

# NEW TOOLS IN XCODE 11 FOR VARIOUS TEMPERATURES

# HOW TO WORK WITH THERMAL STATE CONDITIONS?

- ▶ **Register for**  
`ProcessInfo.thermalStateDidChangeNotification`
- ▶ **Use the `ProcessInfo.ThermalState` cases to react to thermal state changes**
- ▶ **Switch off background and unneeded functionality when thermal state is elevated**

# ACTIONS ON PROCESSINFO.THERMALSTATE

THERMAL STATE	RECOMMENDATIONS	SYSTEM ACTIONS
Nominal	No corrective action needed	
Fair	Slightly elevated thermal state, apps can proactively start energy-saving measures	Photos analysis pauses
Serious	System performance is impacted, reduce CPU, GPU, and I/O usage	ARKit and FaceTime reduce FPS rate, Restore from iCloud is paused
Critical	Reduce CPU, GPU, and I/O usage, and stop using peripherals such as camera	ARKit and FaceTime drop FPS rate

# SUBSCRIBE TO THERMAL STATE CONDITION CHANGES

```
NotificationCenter.default.addObserver(  
    self,  
    selector: #selector(reactToThermalStateChange(_:)),  
    name: ProcessInfo.thermalStateDidChangeNotification,  
    object: nil  
)
```

@objc

```
func reactToThermalStateChange(_ notification: Notification) {  
    print(ProcessInfo.processInfo.thermalState)  
}
```

```
var thermalState = ProcessInfo.ThermalState.nominal {  
    didSet {  
        switch thermalState {  
            case .nominal, .fair: // All good  
                configuration.userFaceTrackingEnabled = true  
                sceneView.renderMotionBlur = true  
            case .serious: // Something went wrong  
                configuration.userFaceTrackingEnabled = false  
                sceneView.renderMotionBlur = true  
            case .critical: // PANIC  
                configuration.userFaceTrackingEnabled = false  
                sceneView.renderMotionBlur = false  
        }  
    }  
}
```

# TO SUM UP

- ▶ **Think about Background**
- ▶ **Minimize Networking**
- ▶ **Measure as much as possible**



**WITH GREAT POWER  
COMES GREAT  
RESPONSIBILITY**

## FURTHER LINKS & READING

- ▶ [@StuFFmc at iOSDevUK 2019: Save the environment with Xcode](#)
- ▶ [@DonnyWals on Supporting Low Data Mode in your app](#)
- ▶ [WWDC2019: Improving Battery Life and Performance](#)
- ▶ [WWDC2019: Designing for Adverse Network and Temperature Conditions](#)
- ▶ [Energy Efficiency Guide for iOS Apps](#)

# THANK YOU!



 **@AKOMAGORKINA**