

Drowsy Power Management

Matthew Lentz James Litton Bobby Bhattacharjee

University of Maryland



Background Energy Consumption

Mobile devices consume energy
without human-interaction

Many (periodic) **short-lived events** executed:



Fetch Remote Data

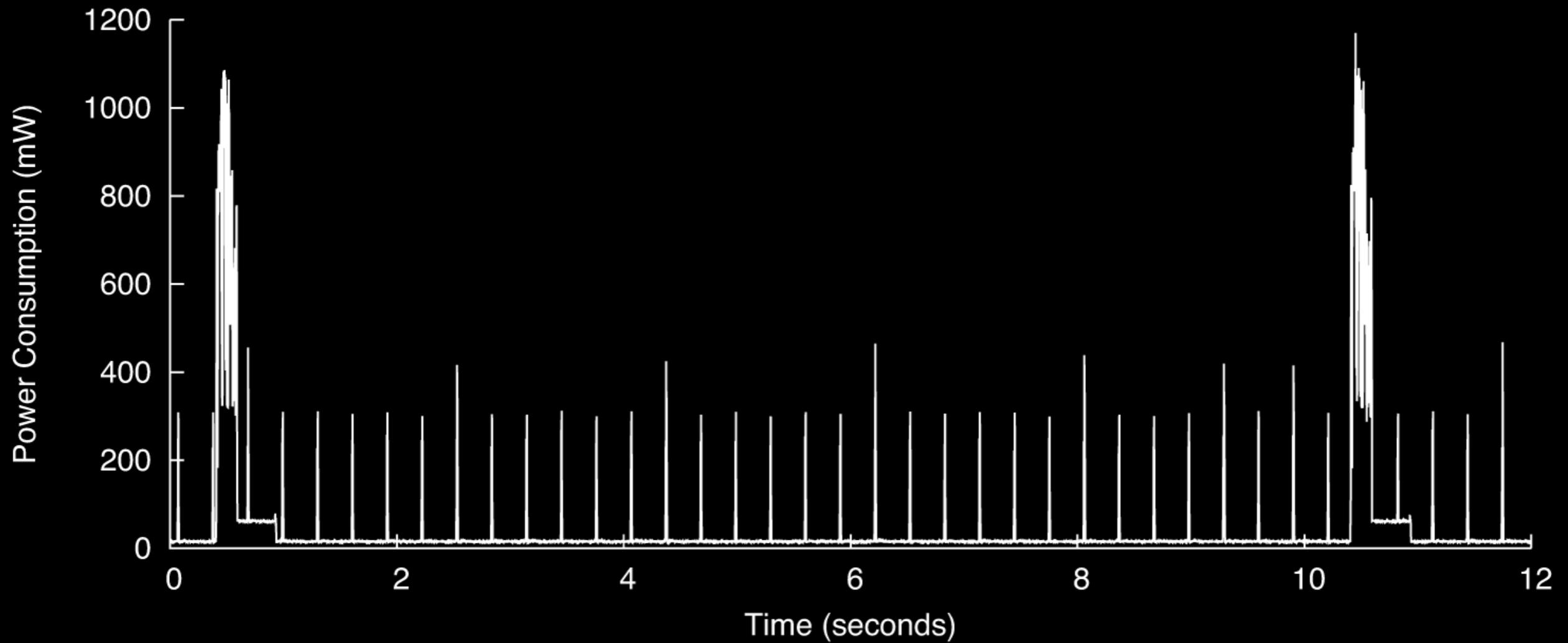
Communicate with Nearby Device(s)

Receive Push Notification

Sample Sensor(s)

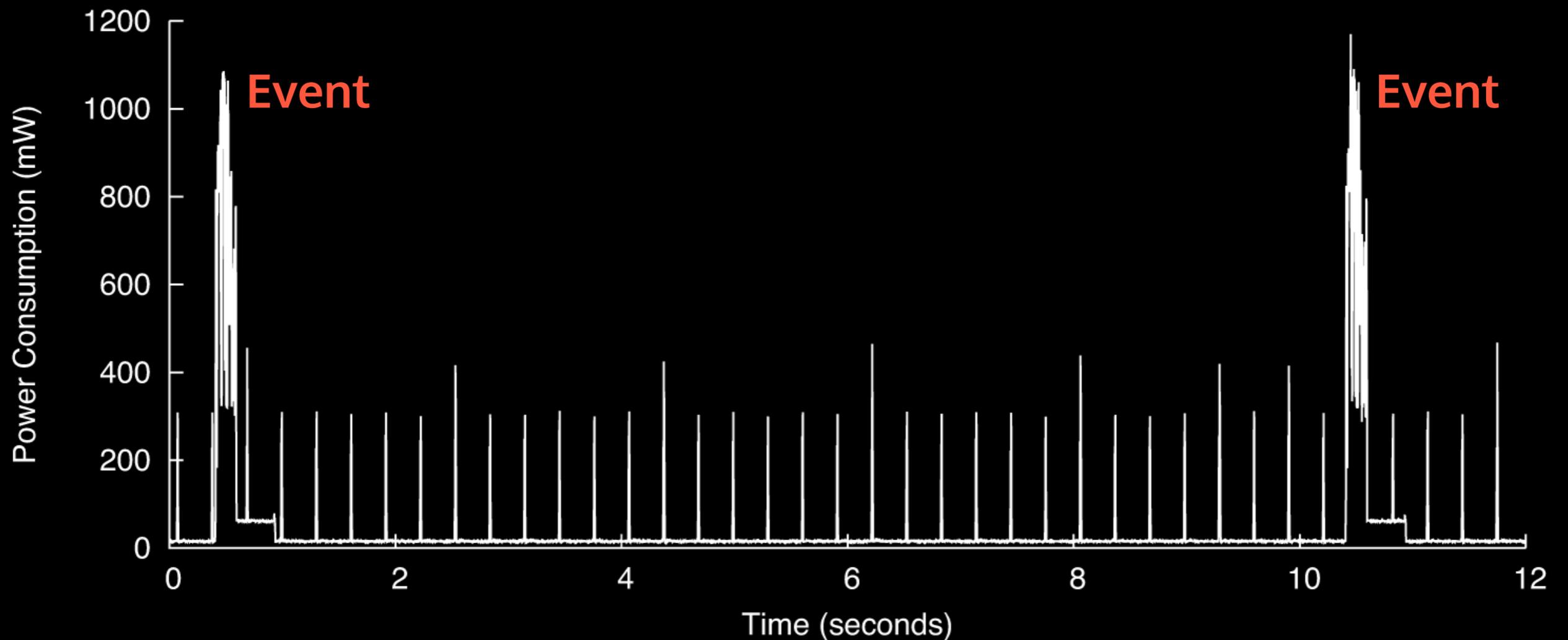
...

Power Trace with Events

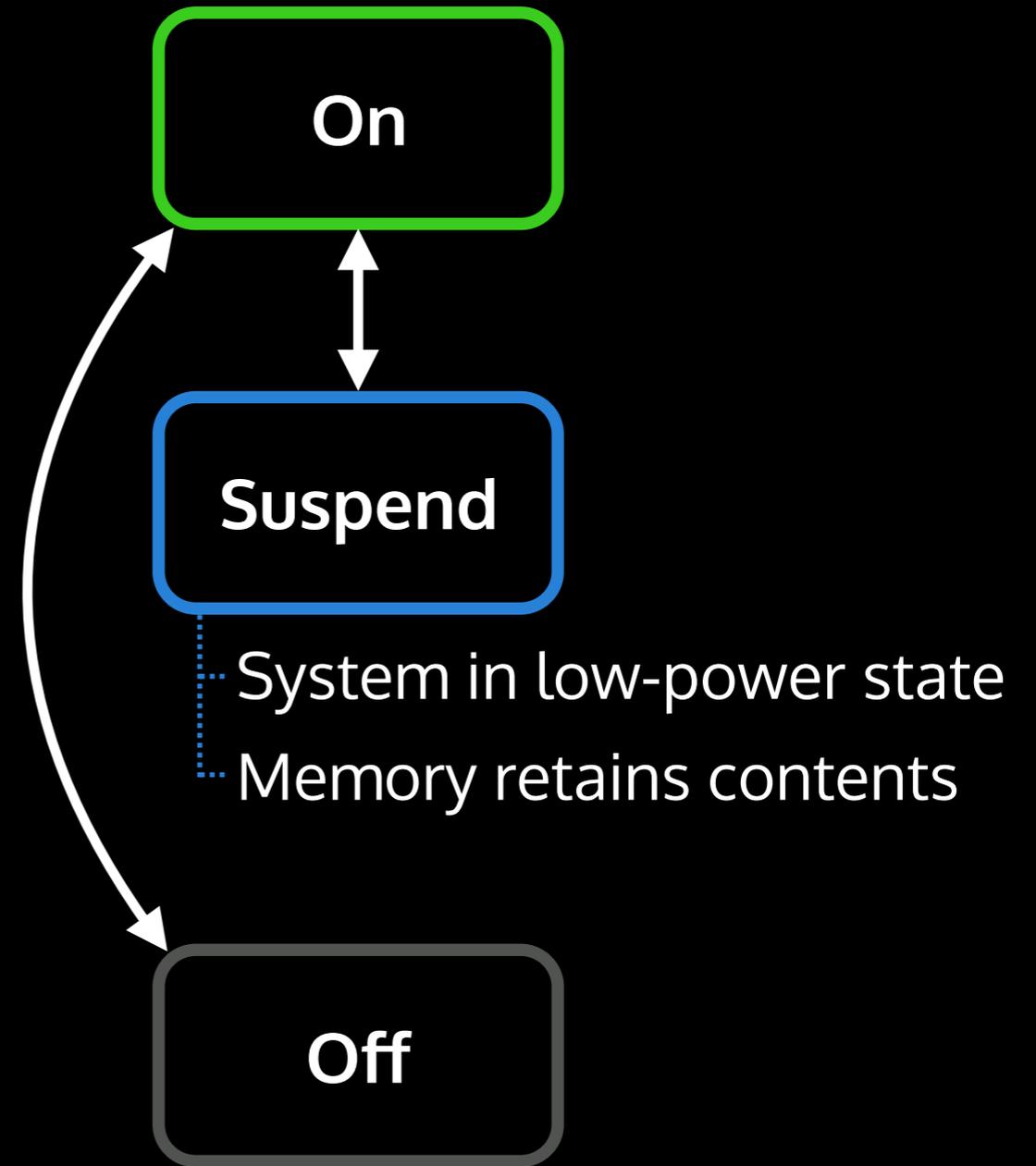
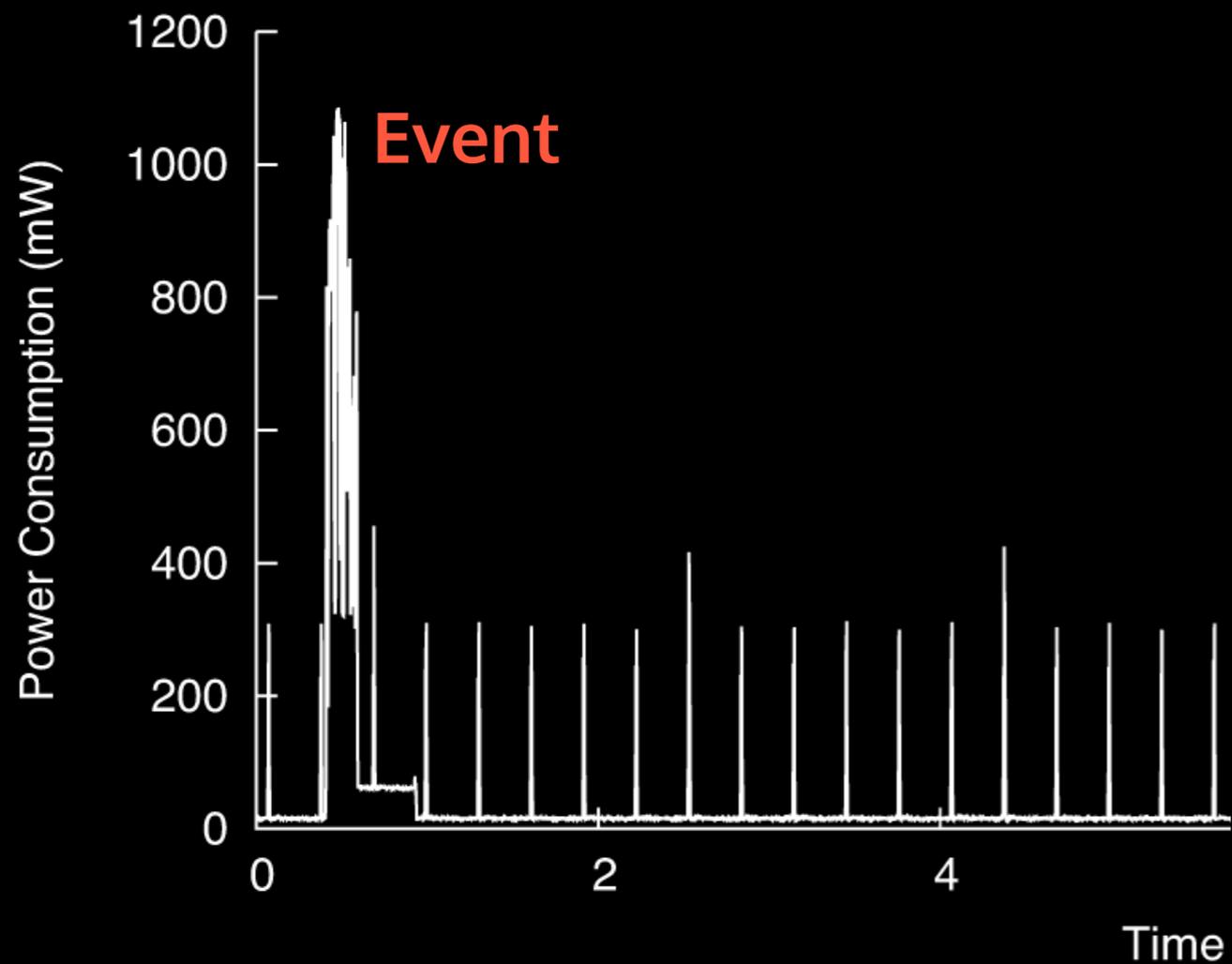


Power Trace with Events

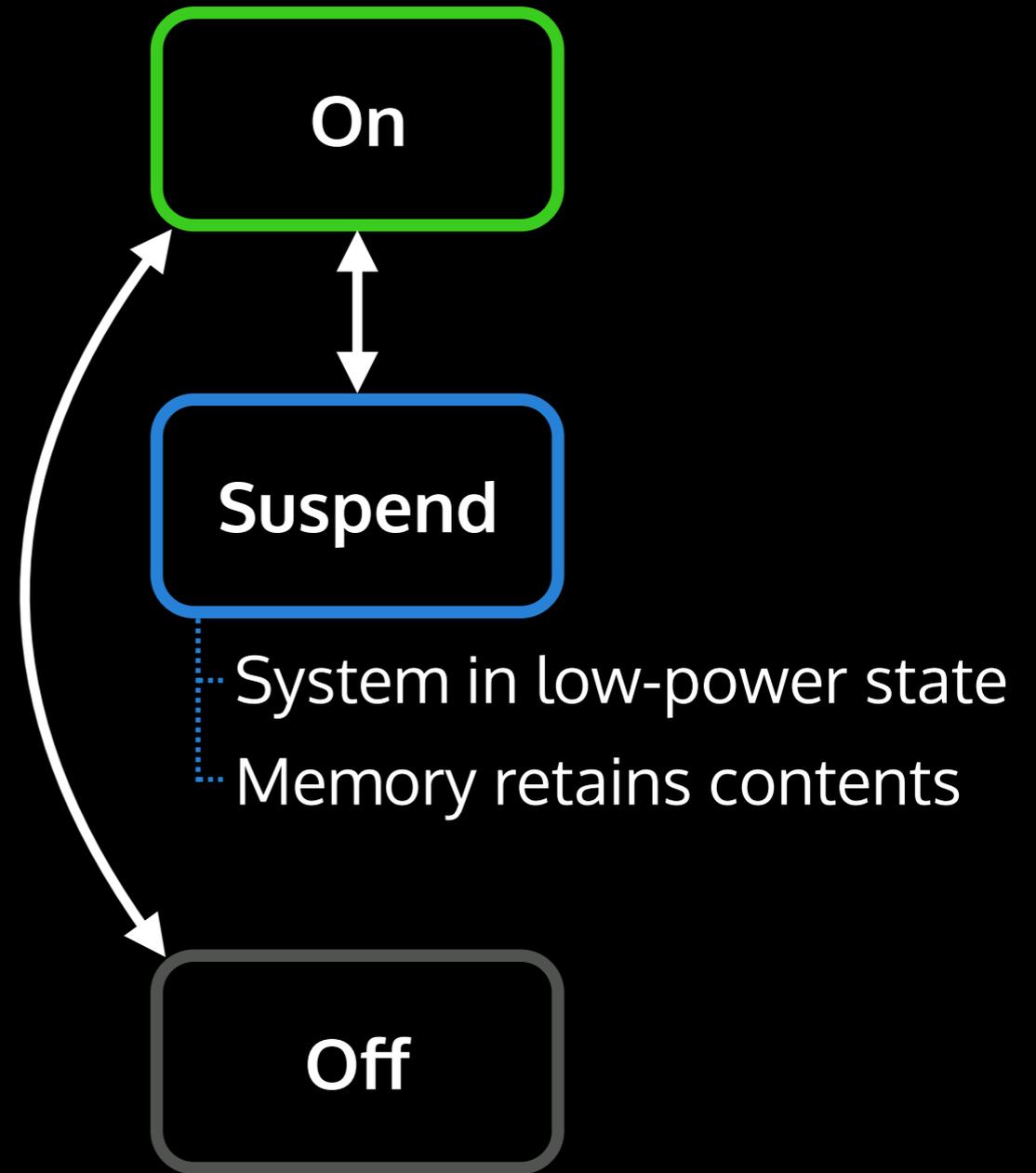
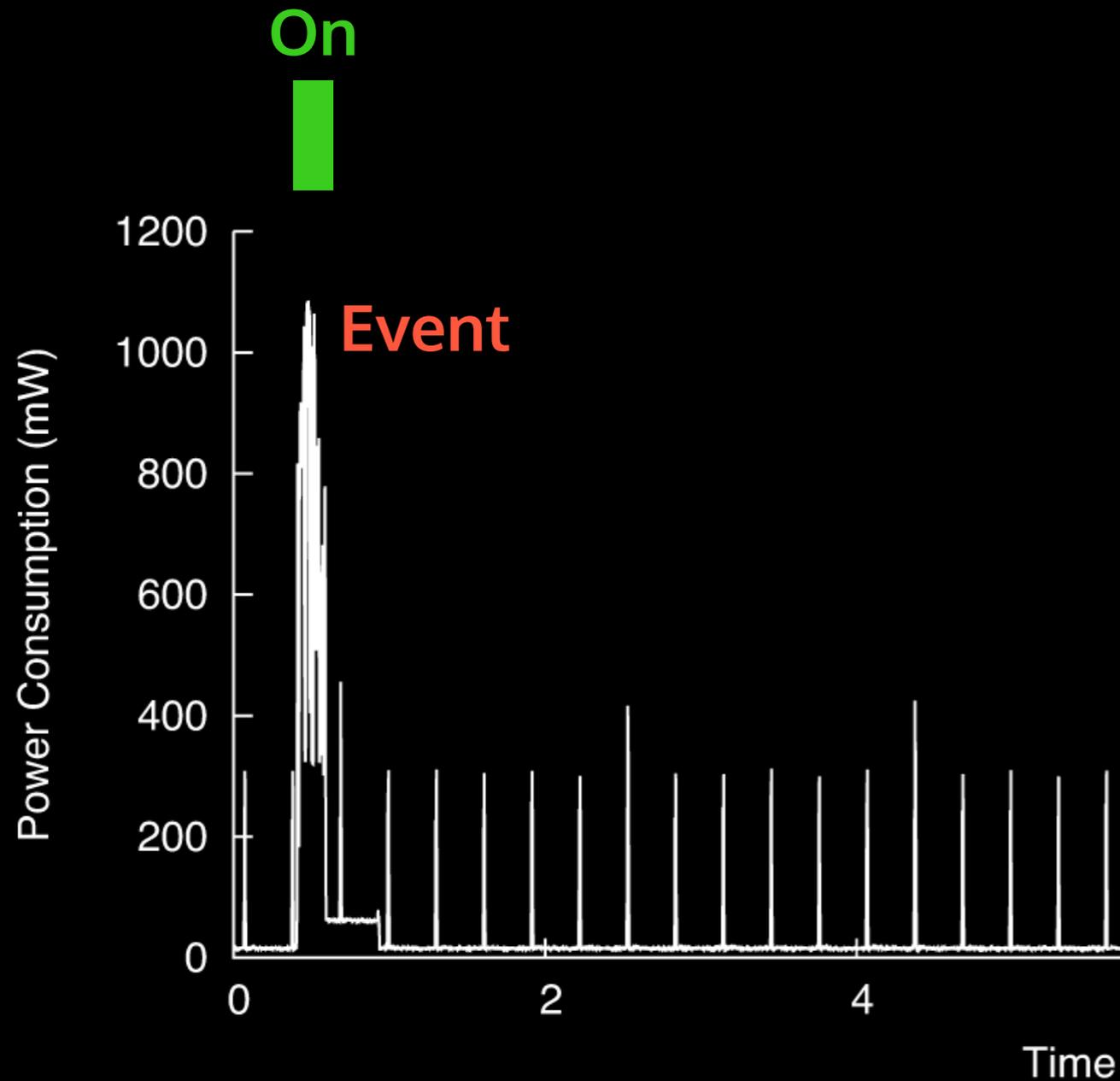
Event = Pull data from remote server over WiFi



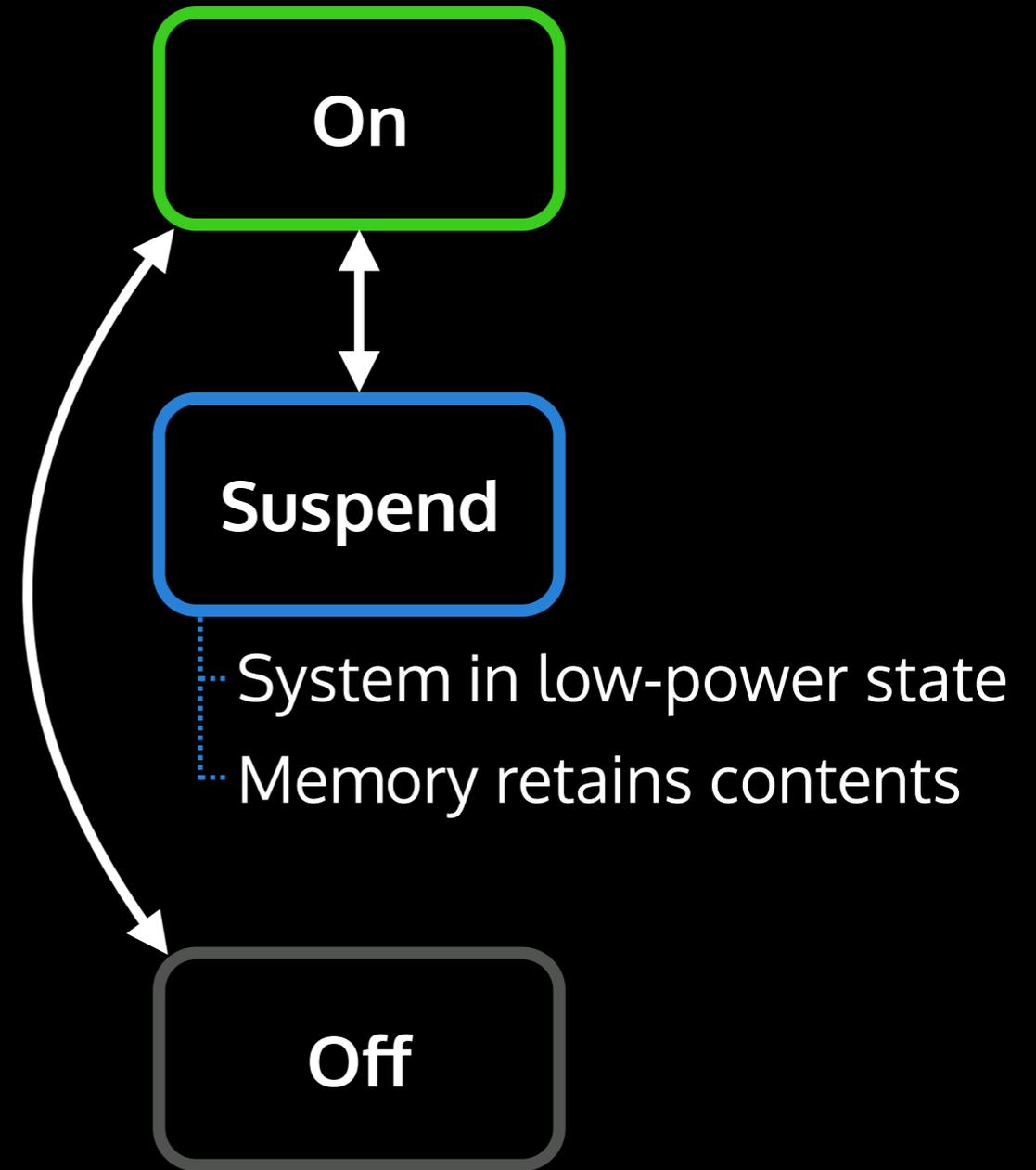
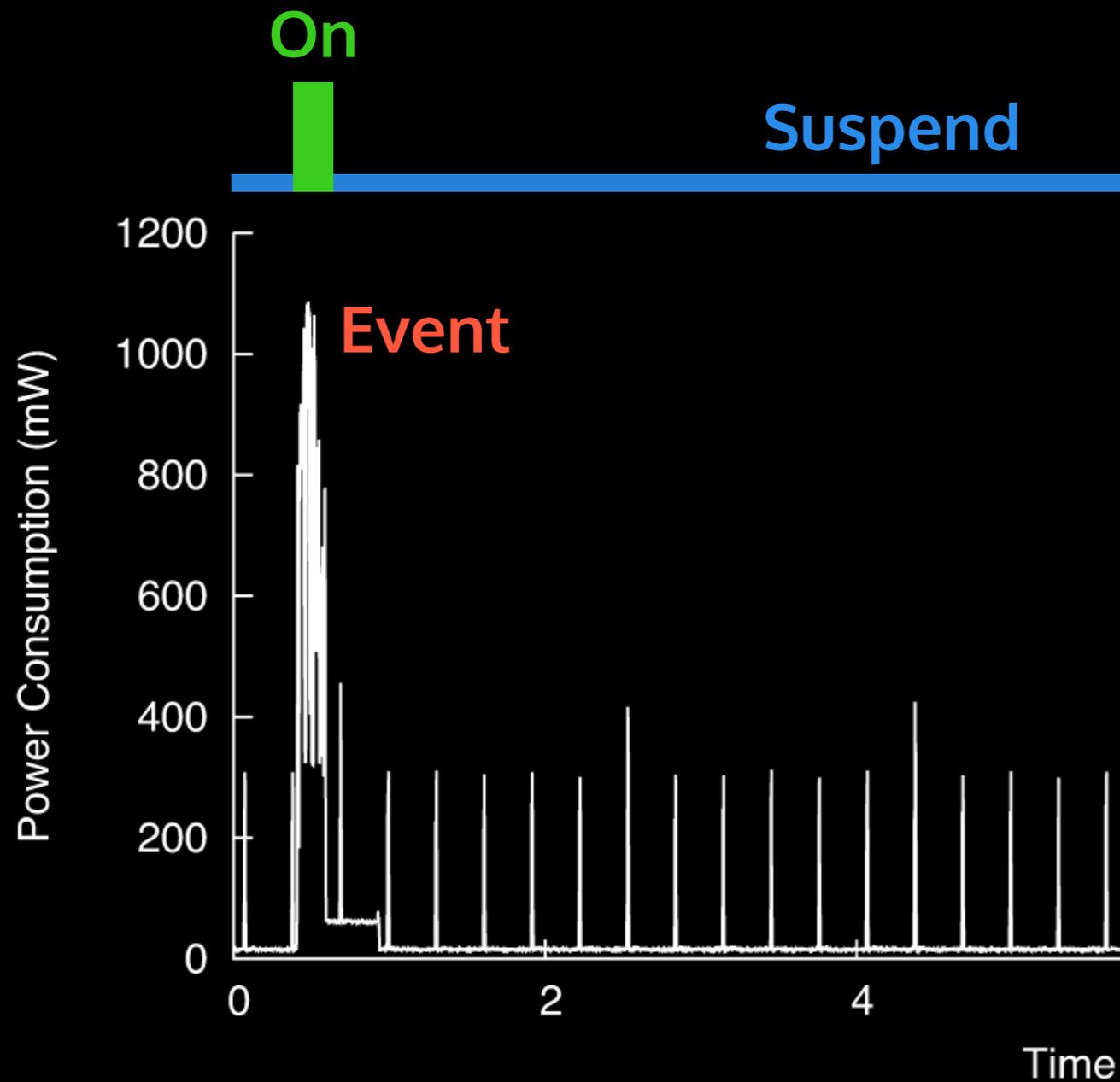
Power Management States



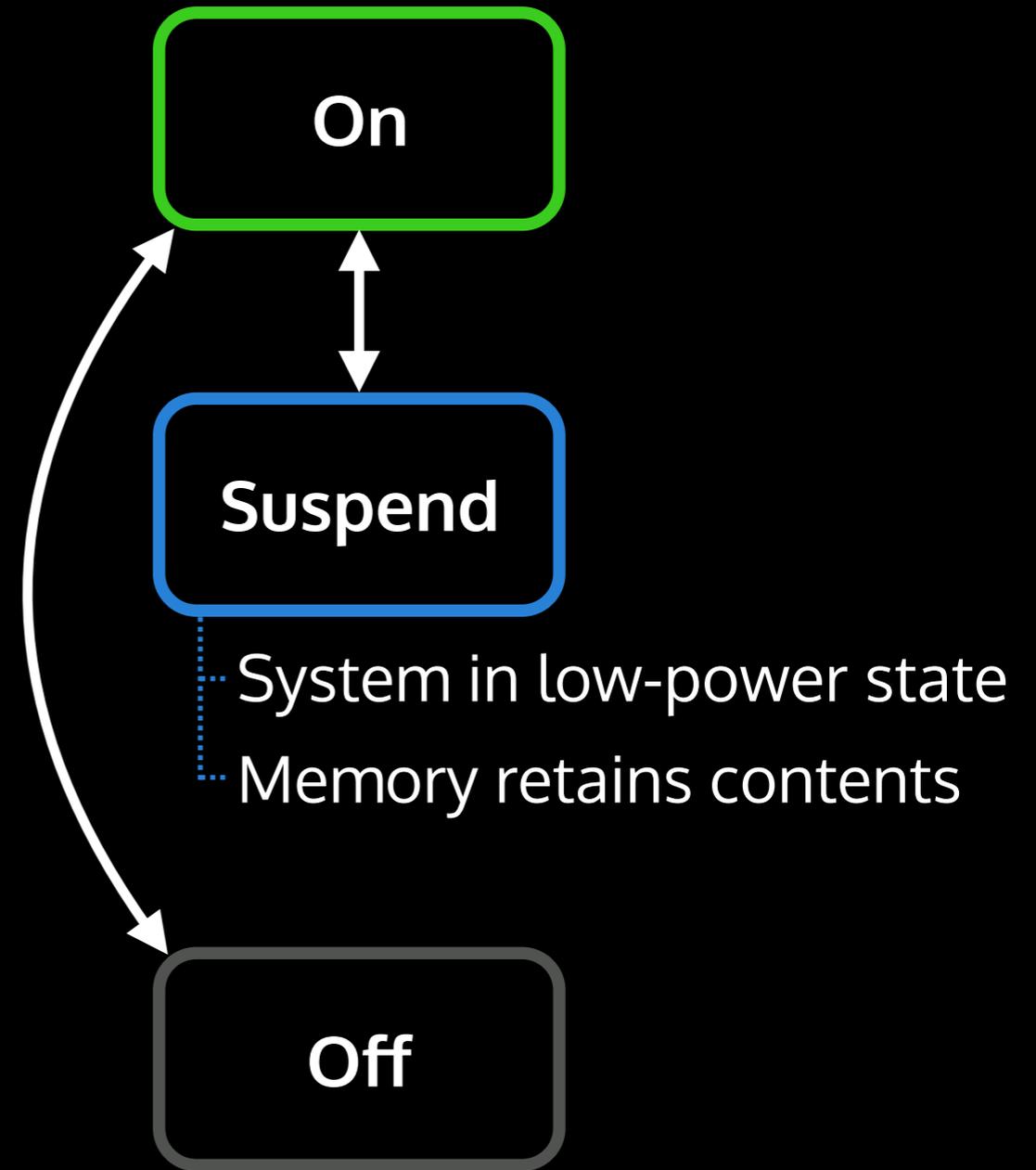
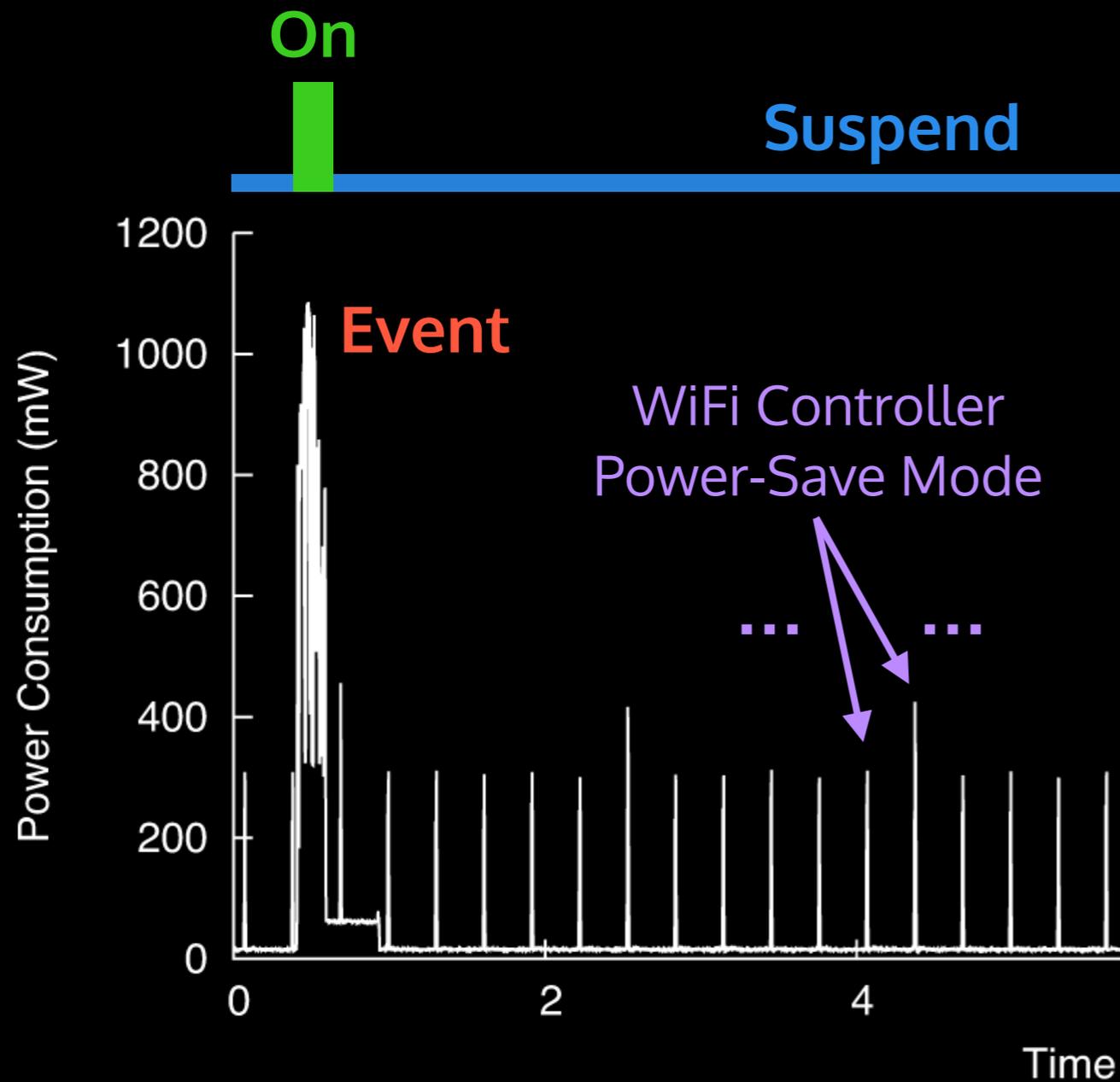
Power Management States



Power Management States



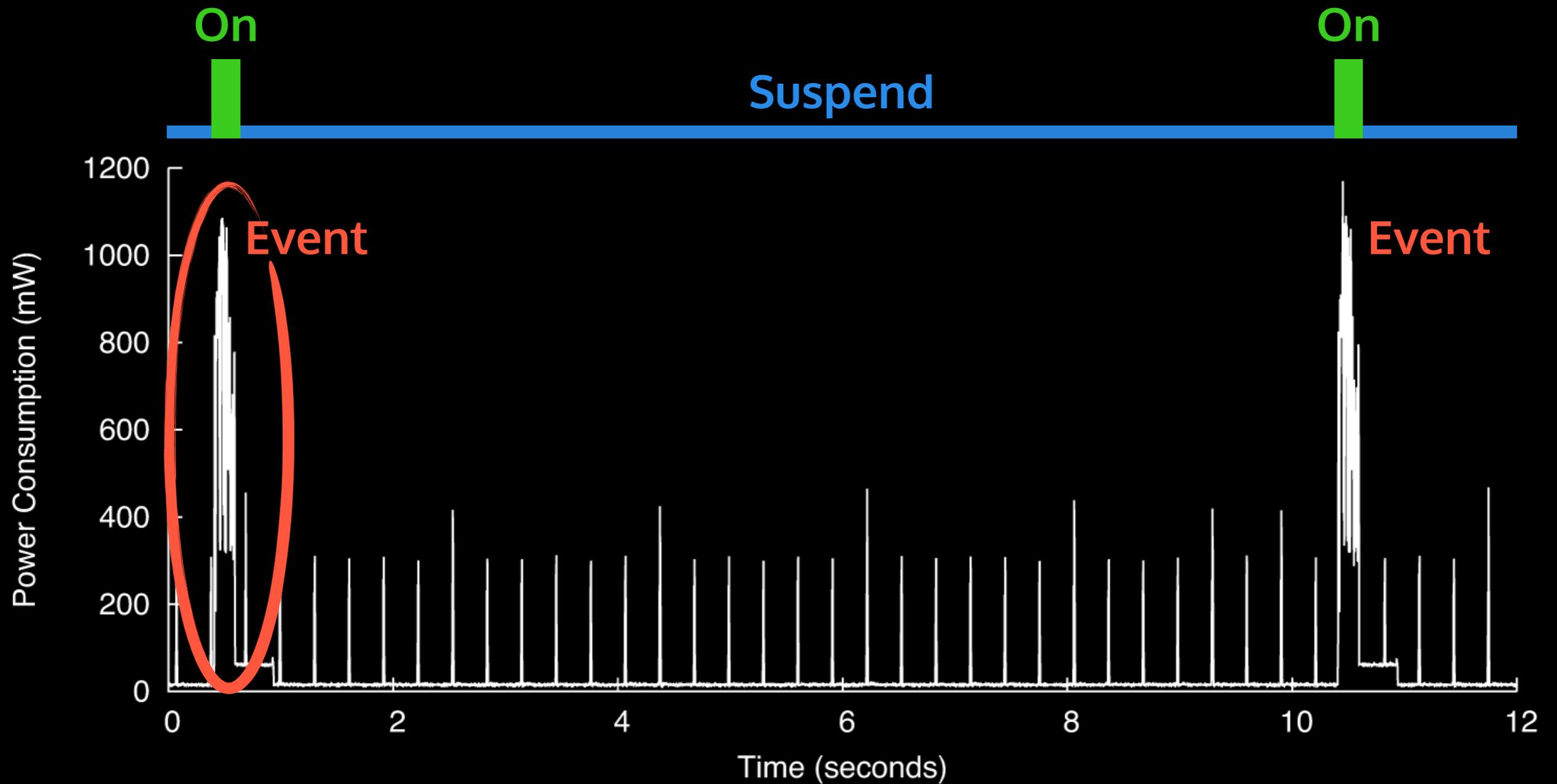
Power Management States



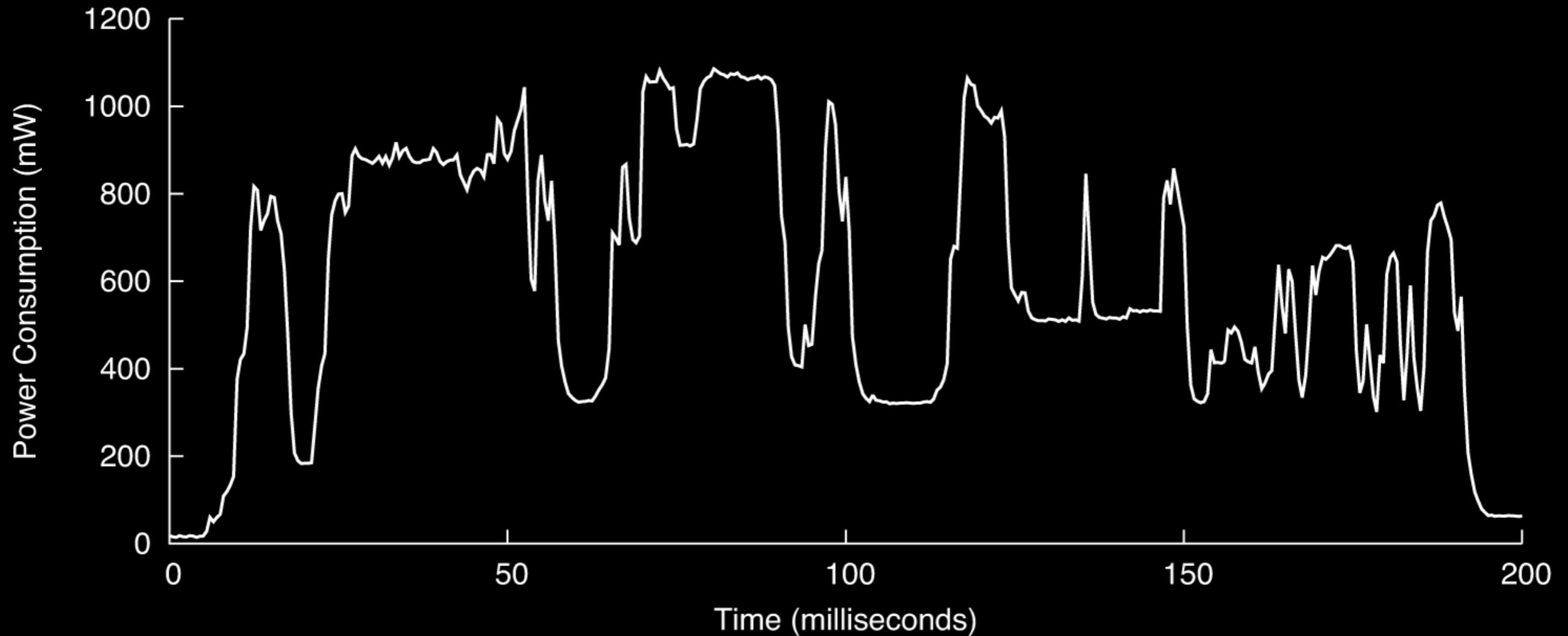
Power Trace with Events



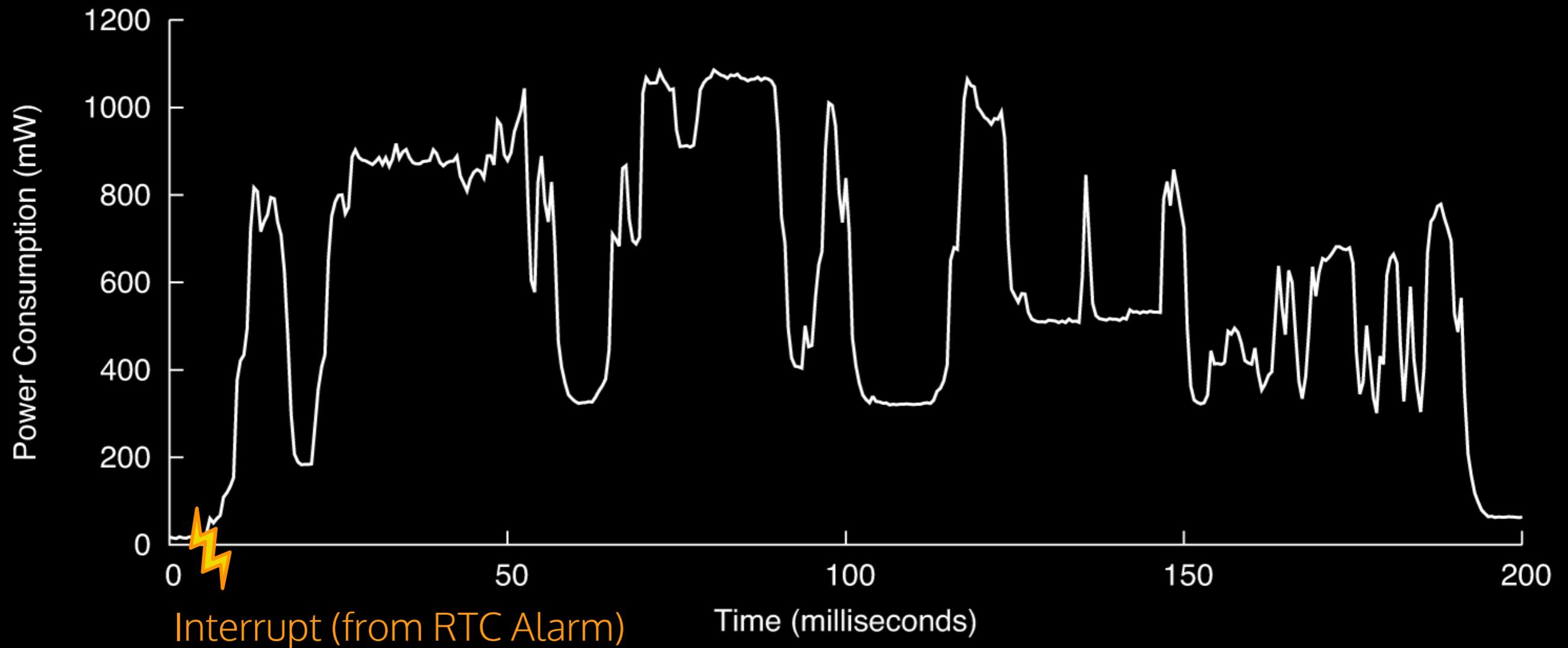
Power Trace with Events



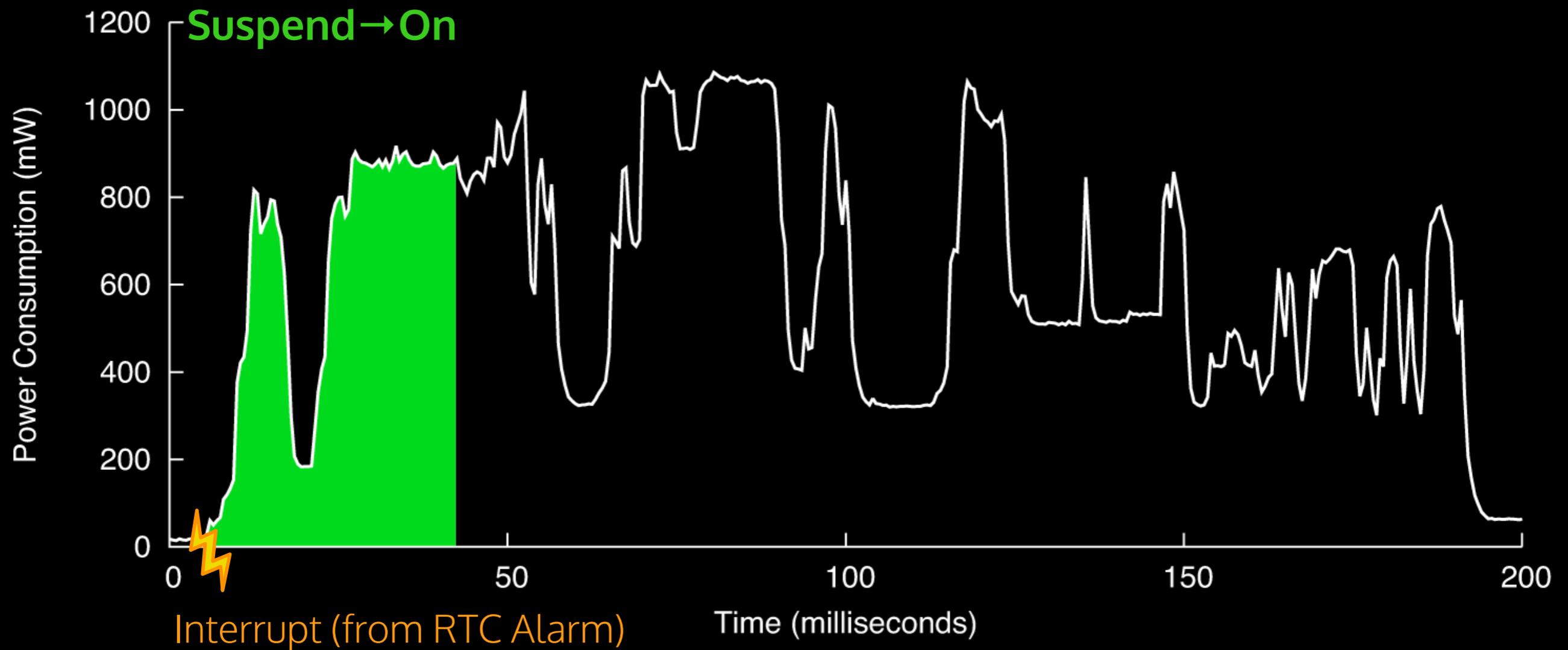
What Happens During a Wakeup?



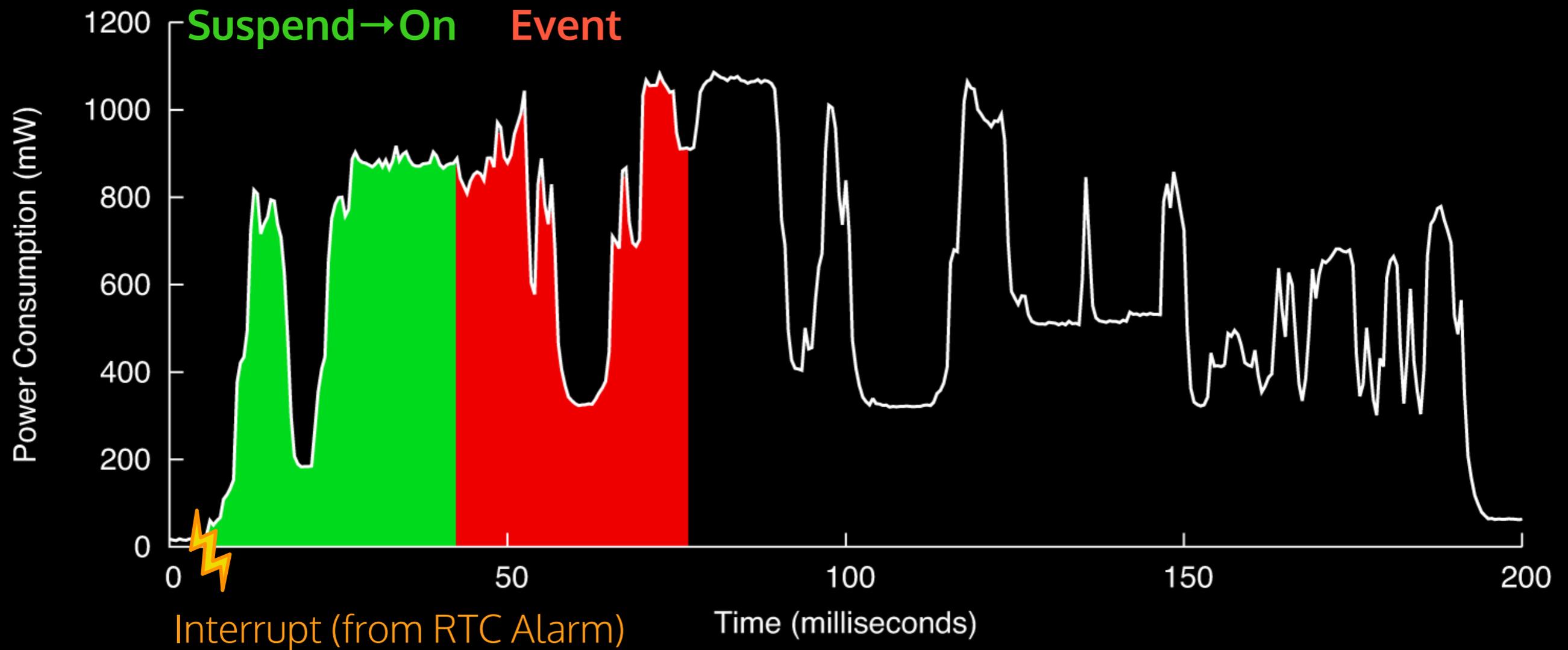
What Happens During a Wakeup?



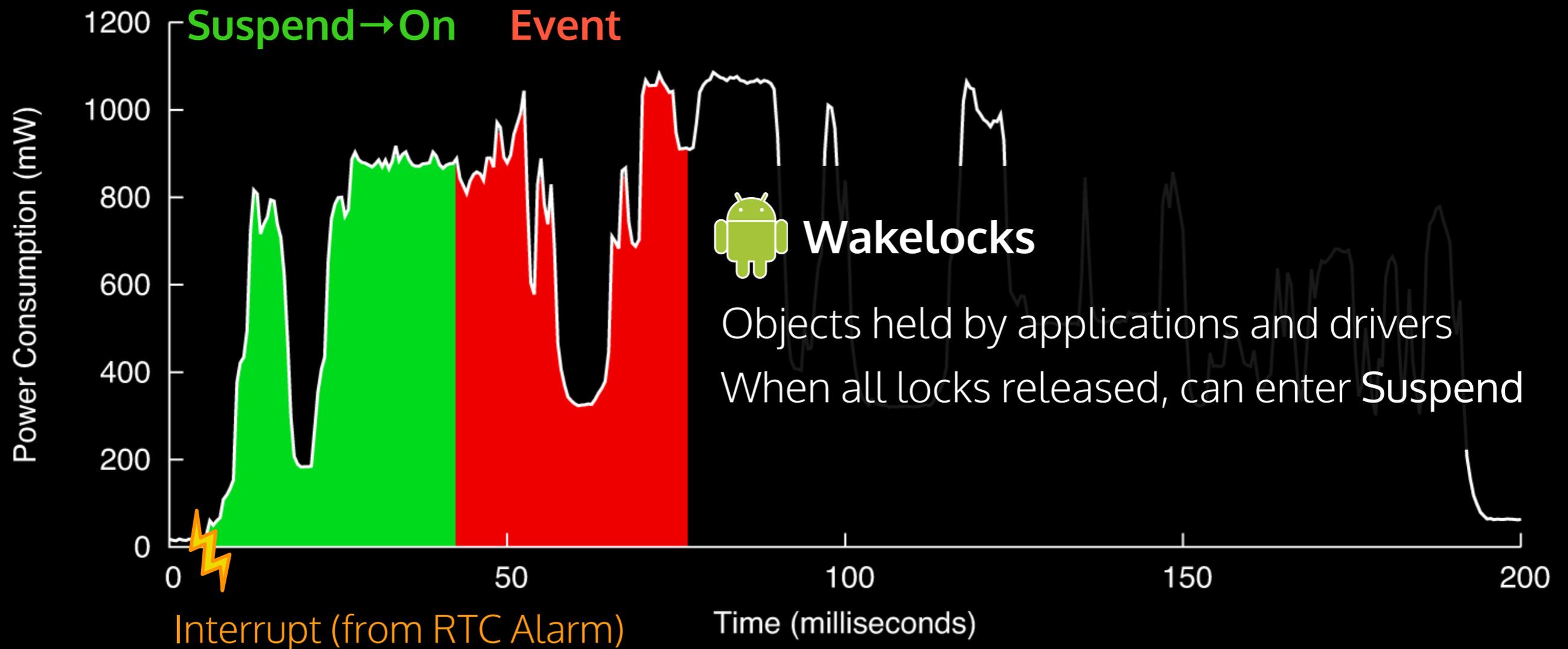
What Happens During a Wakeup?



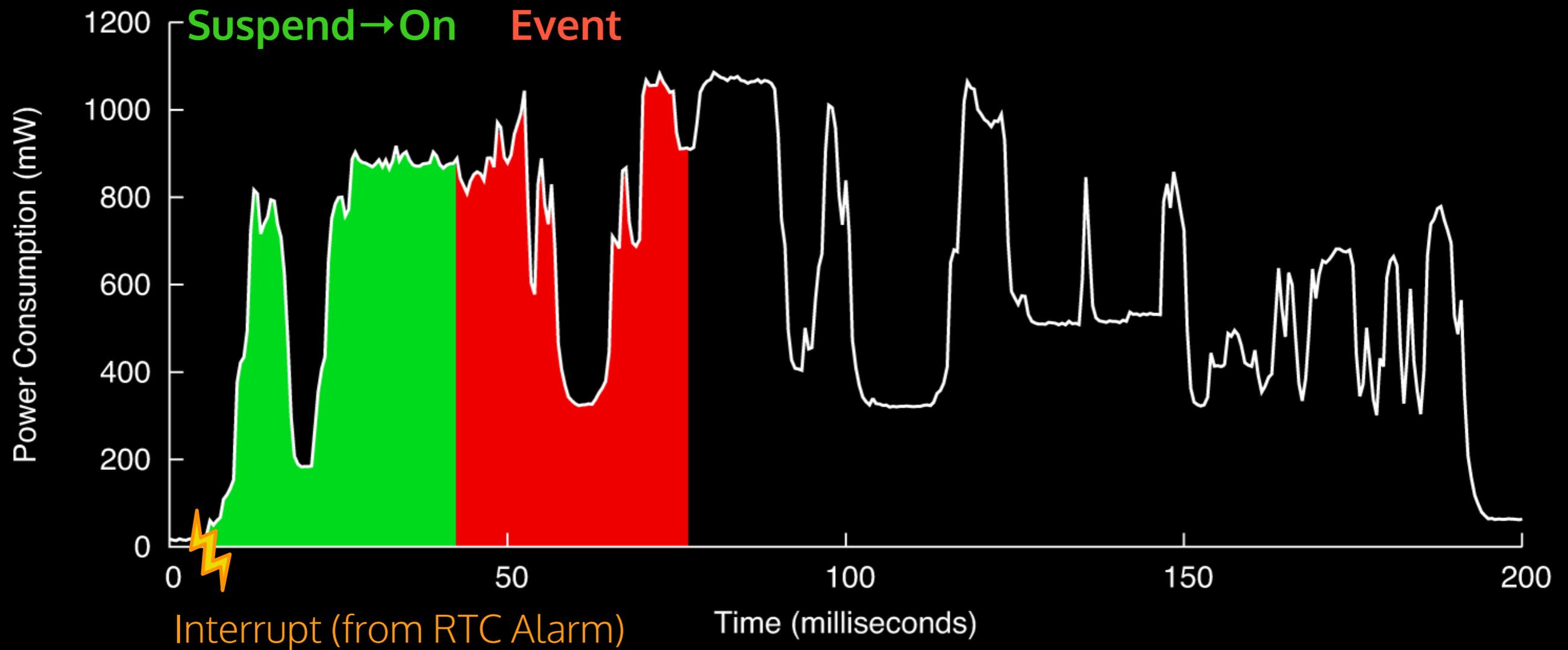
What Happens During a Wakeup?



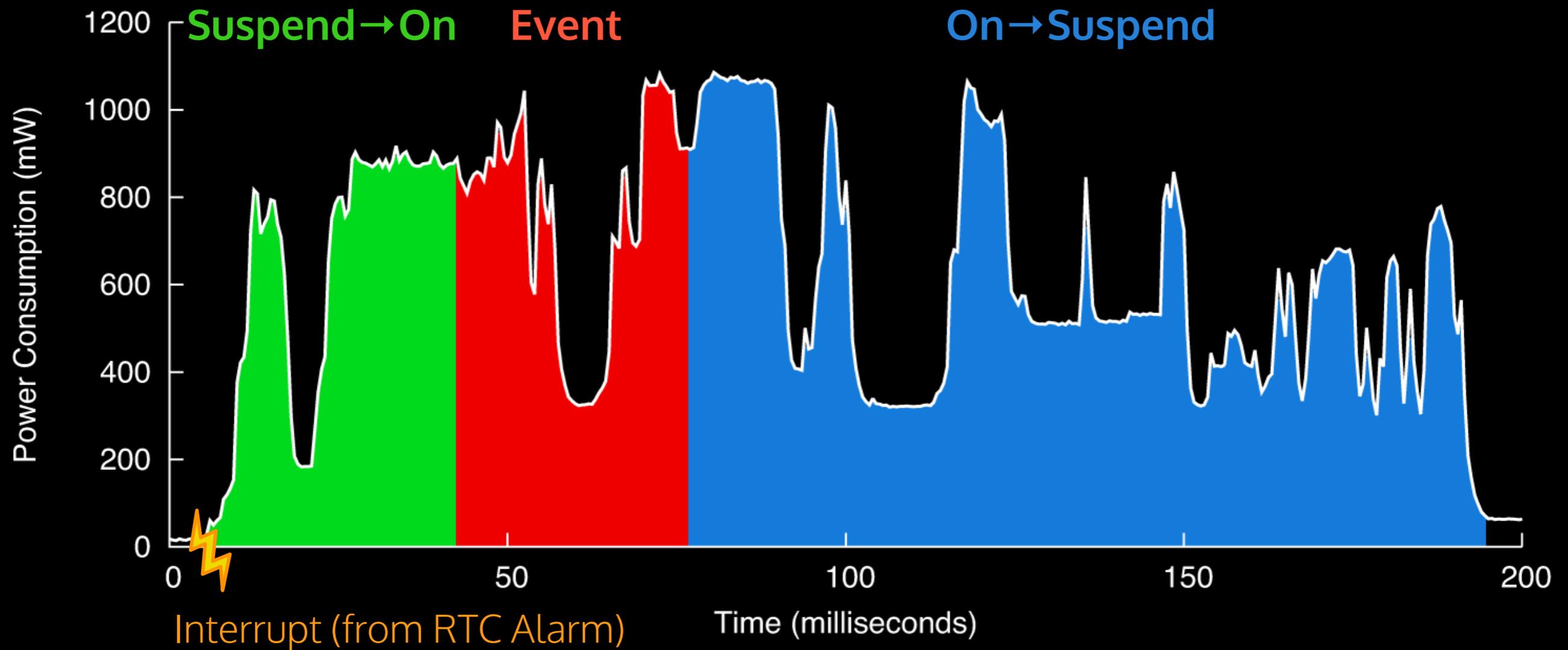
What Happens During a Wakeup?



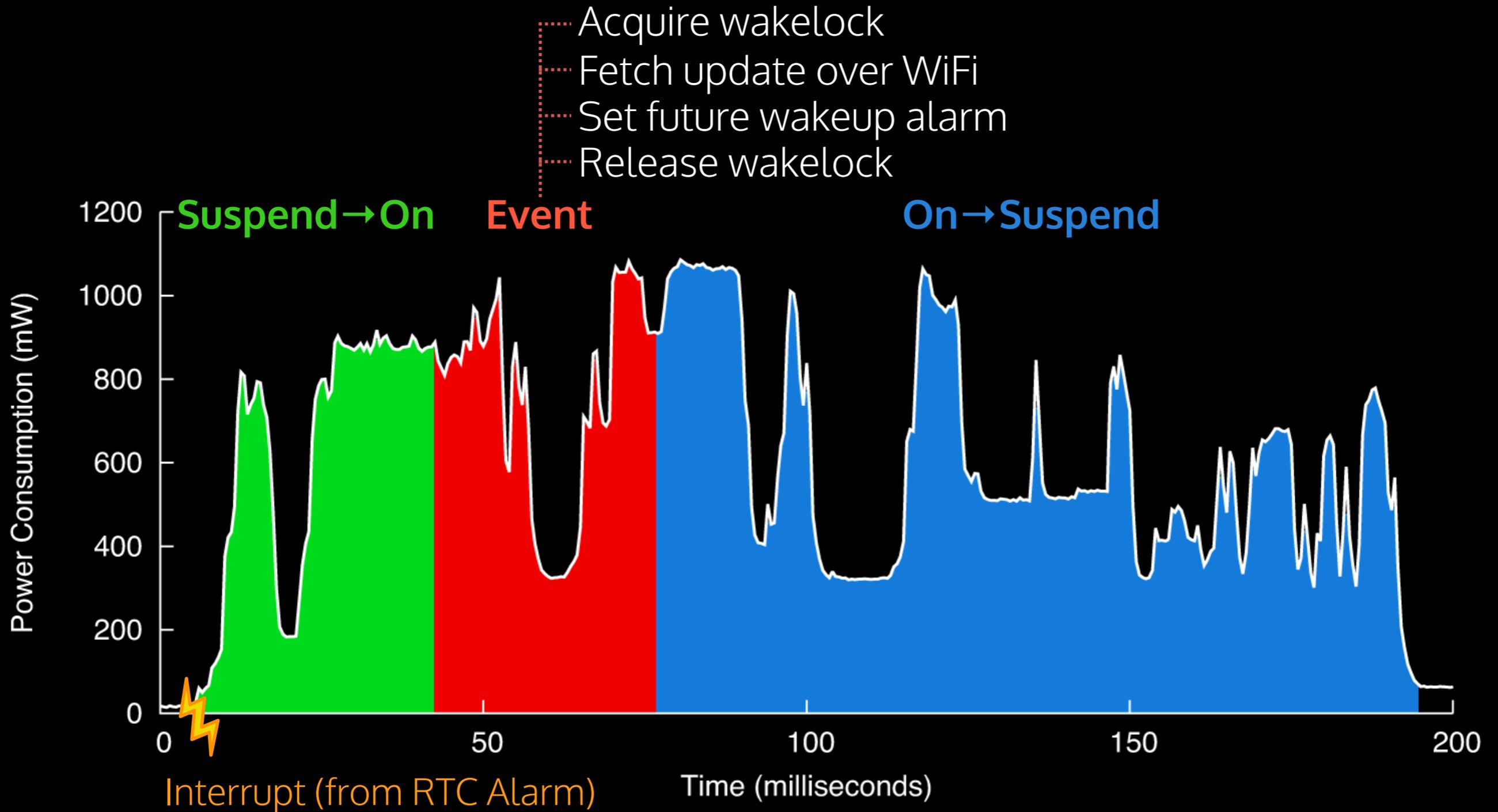
What Happens During a Wakeup?



What Happens During a Wakeup?

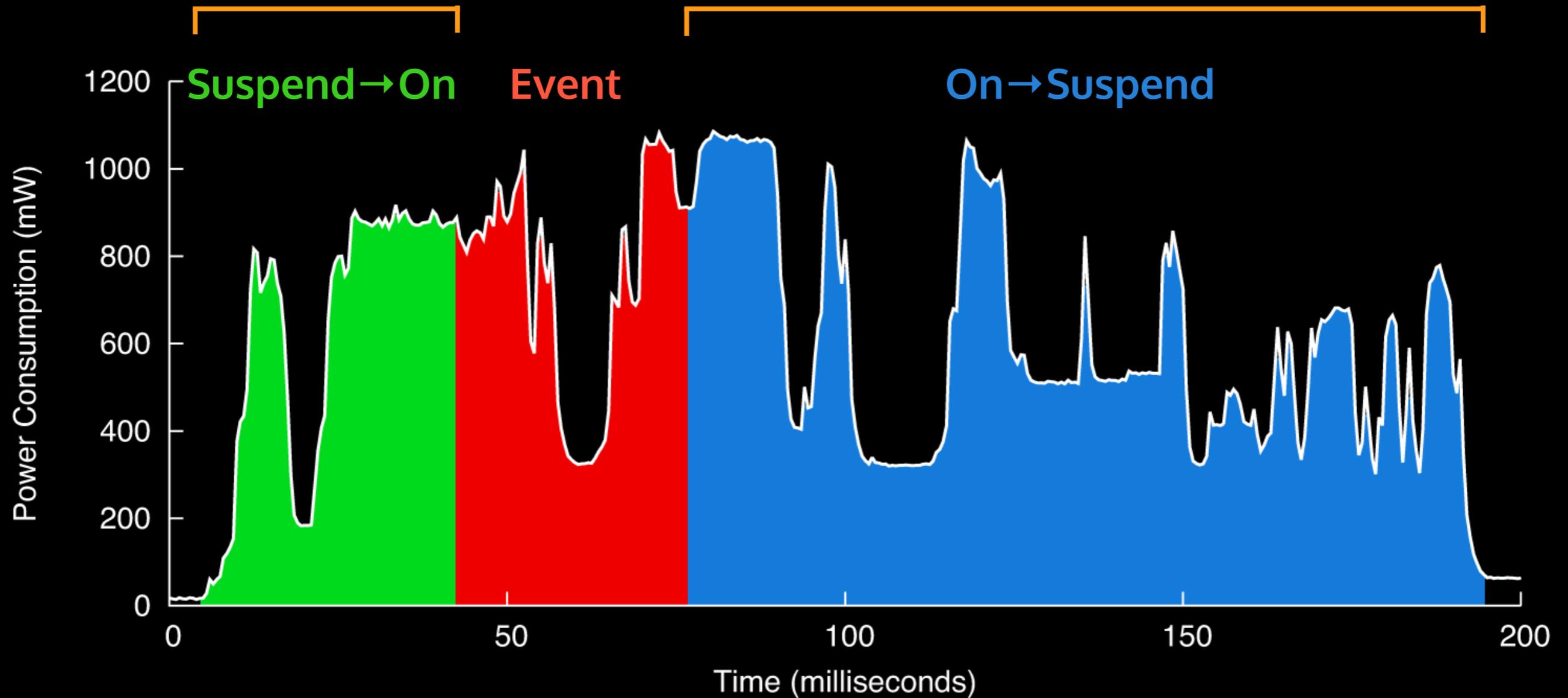


What Happens During a Wakeup?



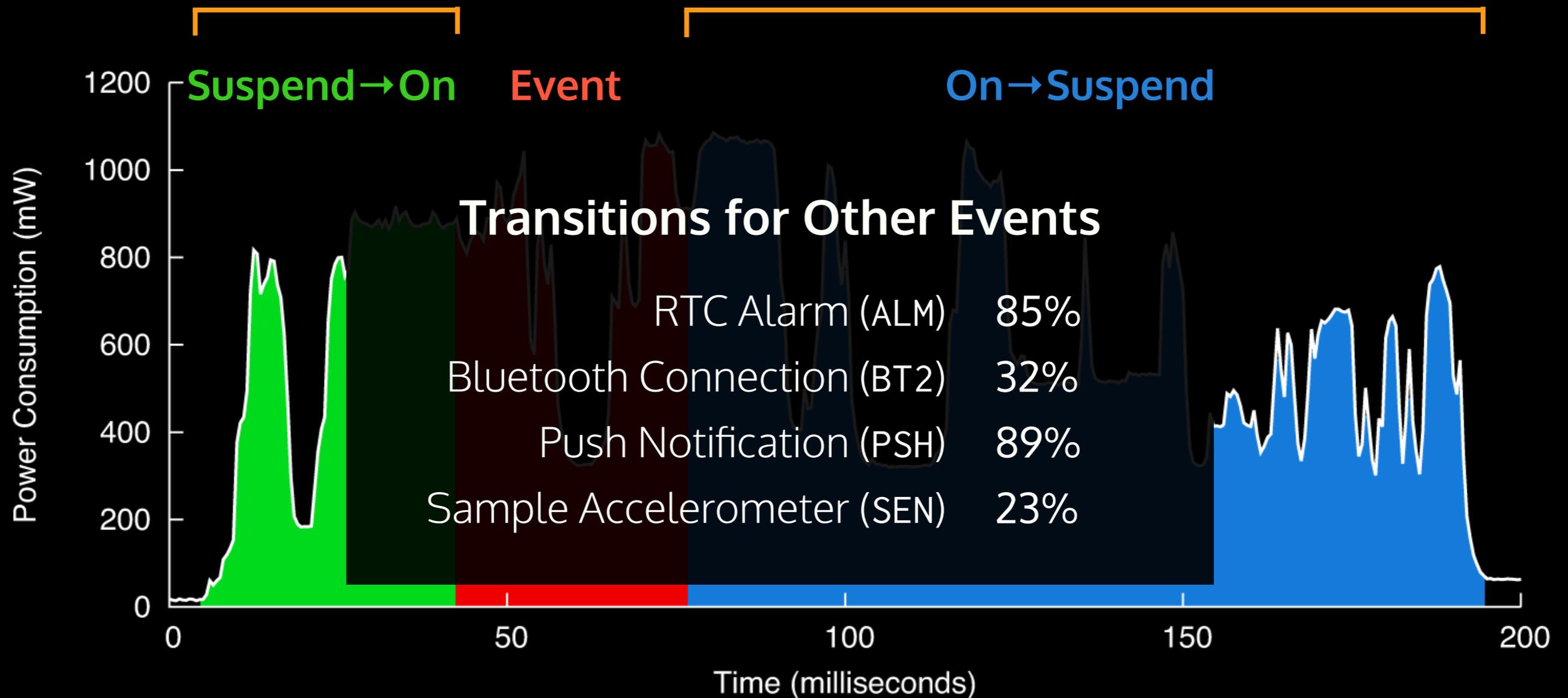
Transitions are Inefficient

Transitions account for 75% of total energy consumption!



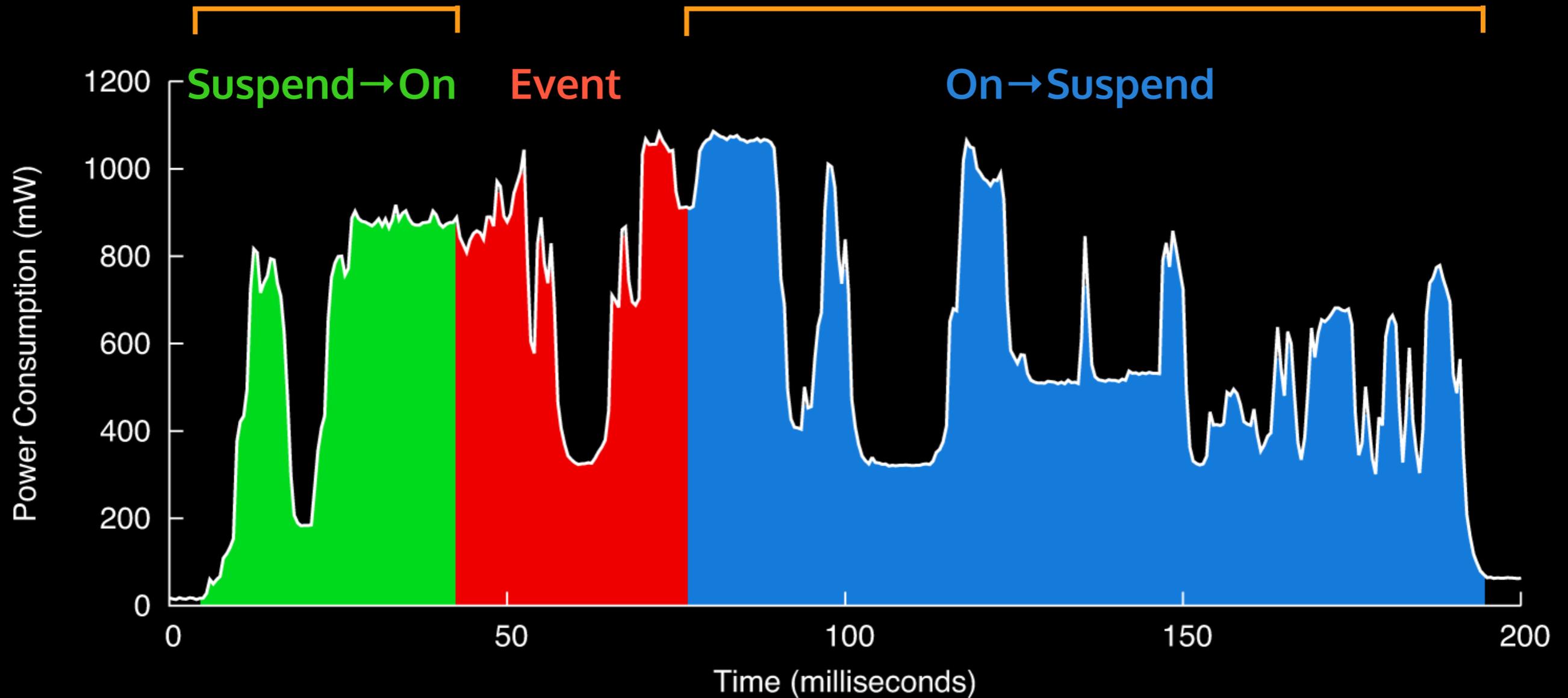
Transitions are Inefficient

Transitions account for 75% of total energy consumption!

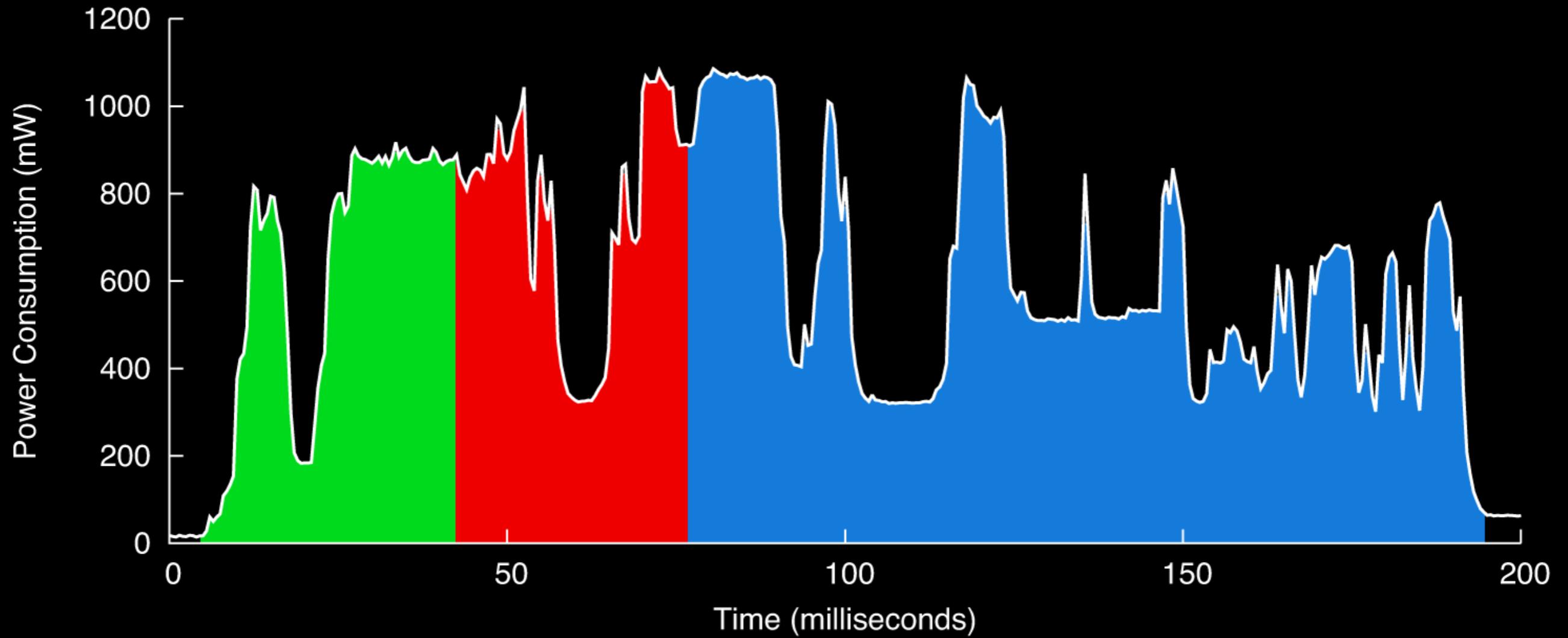


Transitions are Inefficient

Transitions account for 75% of total energy consumption!

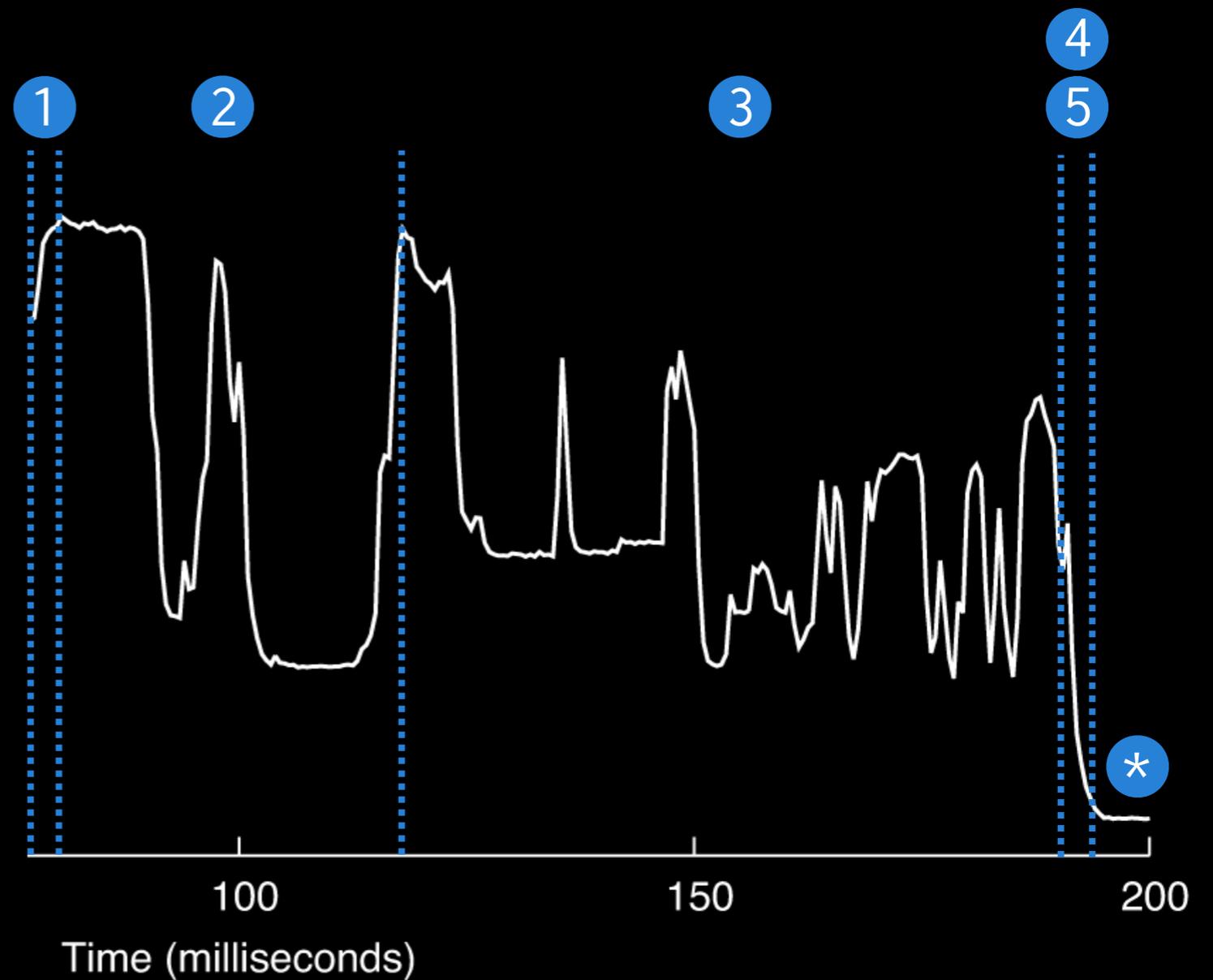


On → Suspend



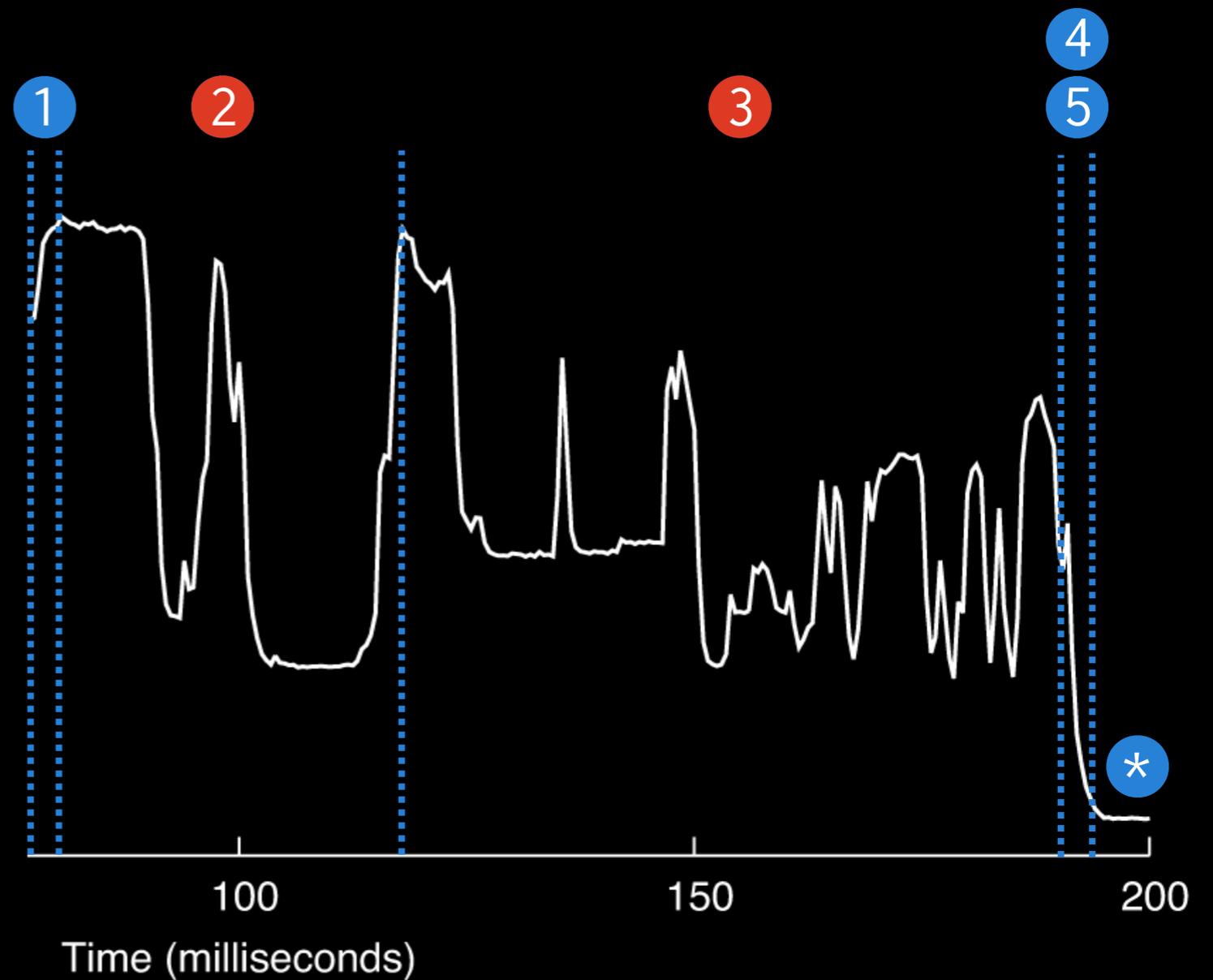
On → Suspend

- 1 Flush filesystem buffers
 - 2 Freeze all tasks
 - 3 Suspend all devices
 - 4 Disable non-boot CPUs
 - 5 Set RAM to self-refresh
- * Wait for interrupt

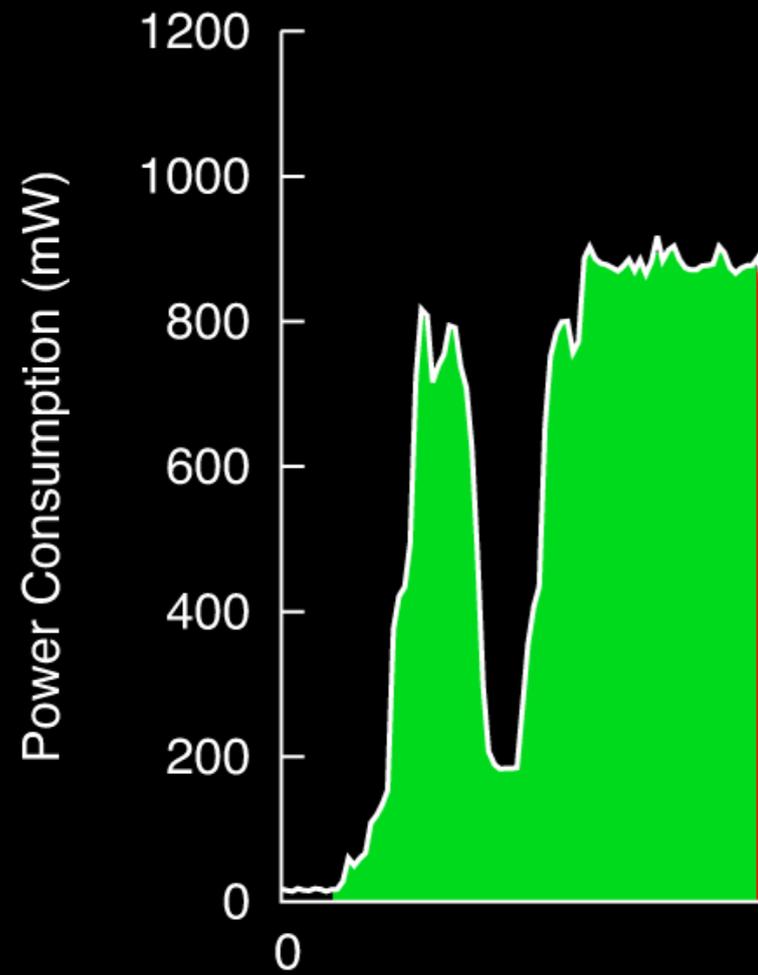


On → Suspend

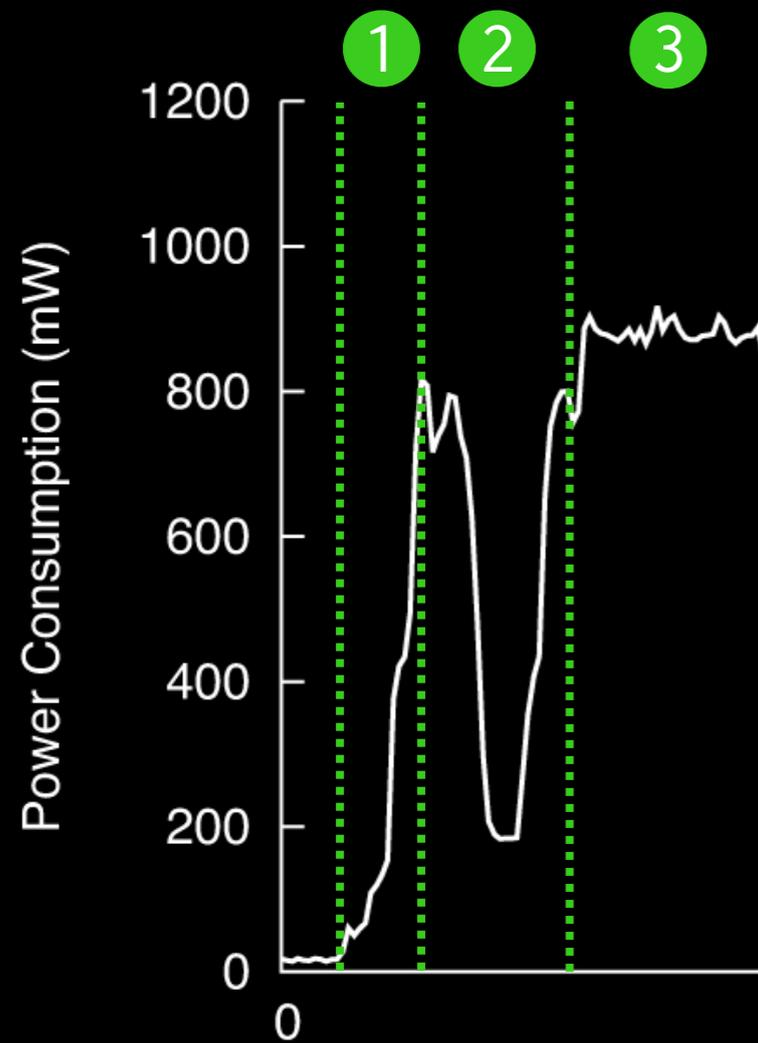
- 1 Flush filesystem buffers
- 2 Freeze all tasks
- 3 Suspend all devices
- 4 Disable non-boot CPUs
- 5 Set RAM to self-refresh
- * Wait for interrupt



Suspend → On

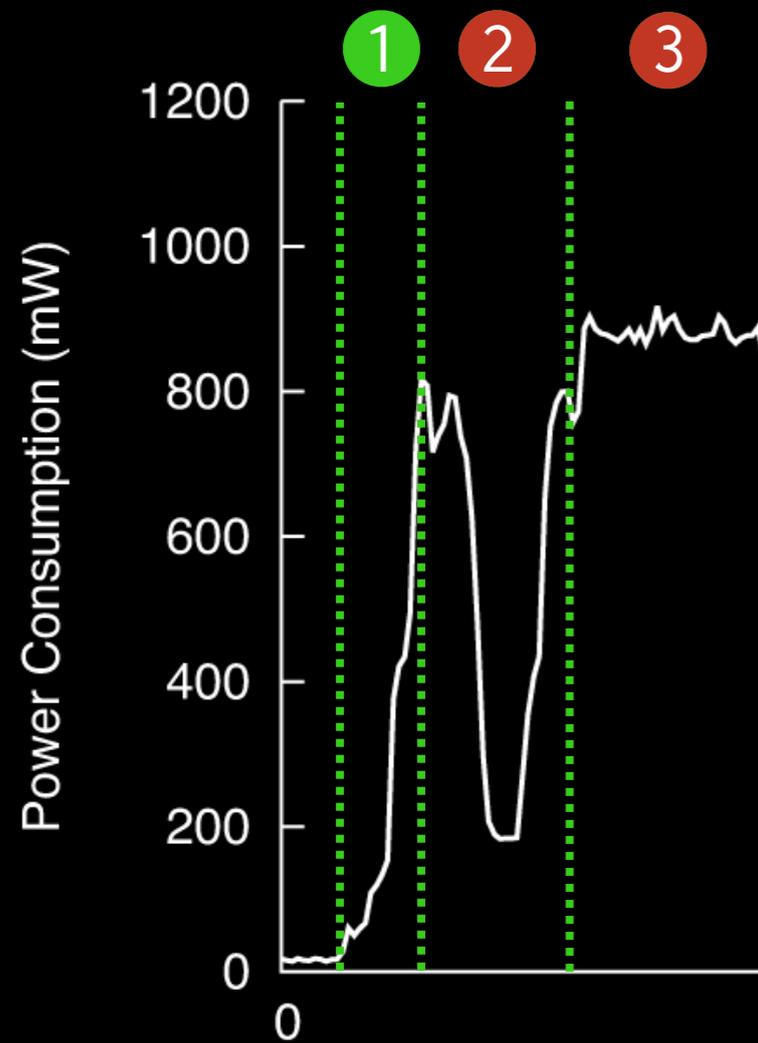


Suspend → On



- 1 Enable CPUs
- 2 Resume all devices
- 3 Thaw all tasks

Suspend → On



- 1 Enable CPUs
- 2 Resume all devices
- 3 Thaw all tasks

Drowsy Power Management

Wake up only what is necessary

Drowsy Power Management

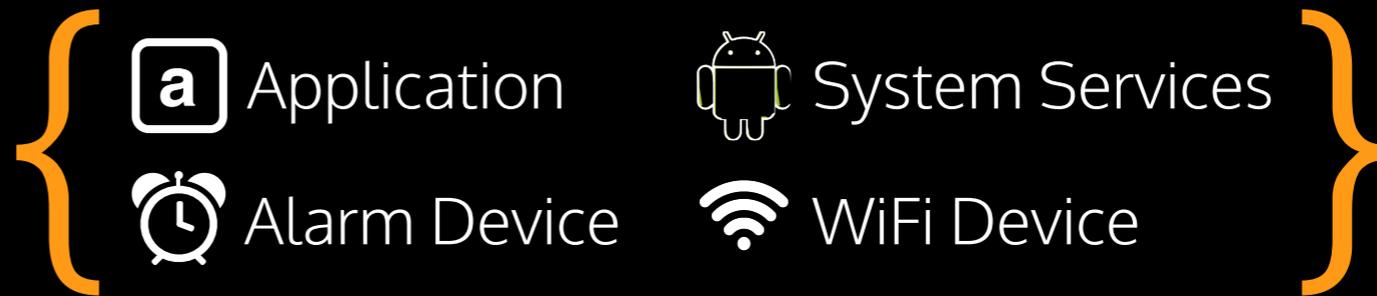
Wake up only what is necessary

Necessary
for Pull Data: { }

Drowsy Power Management

Wake up only what is necessary

Necessary
for Pull Data:



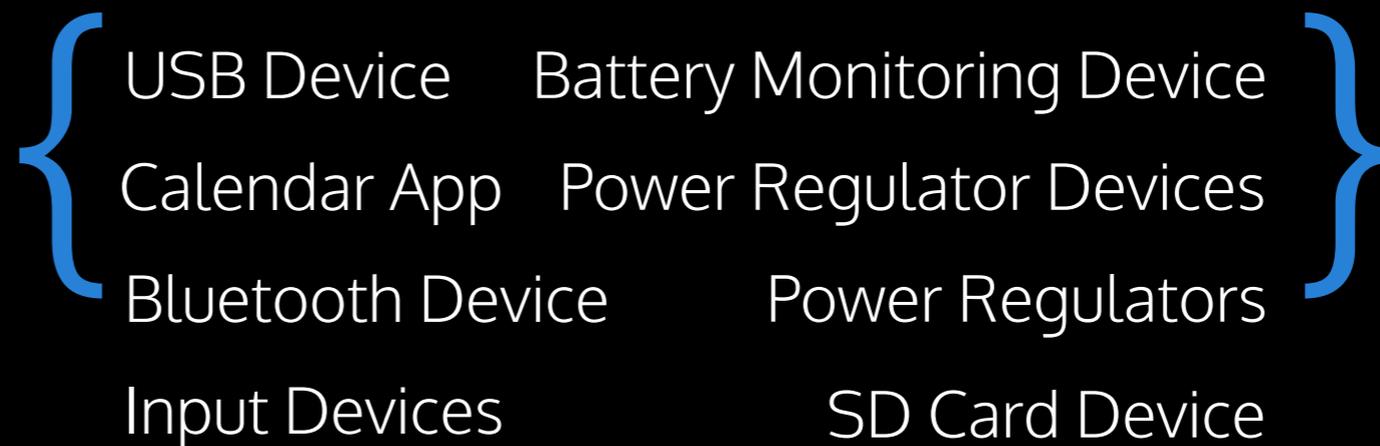
Drowsy Power Management

Wake up only what is necessary

Necessary
for Pull Data:



Unnecessary:



...

Drowsy Power Management

Construct **minimal “wake set”** of tasks and devices

Drowsy Power Management

Construct **minimal “wake set”** of tasks and devices

→ Smallest set that **maintains correct behavior**

Drowsy Power Management

Construct **minimal “wake set”** of tasks and devices

- Smallest set that **maintains correct behavior**
- Expand **on-demand** as event progresses

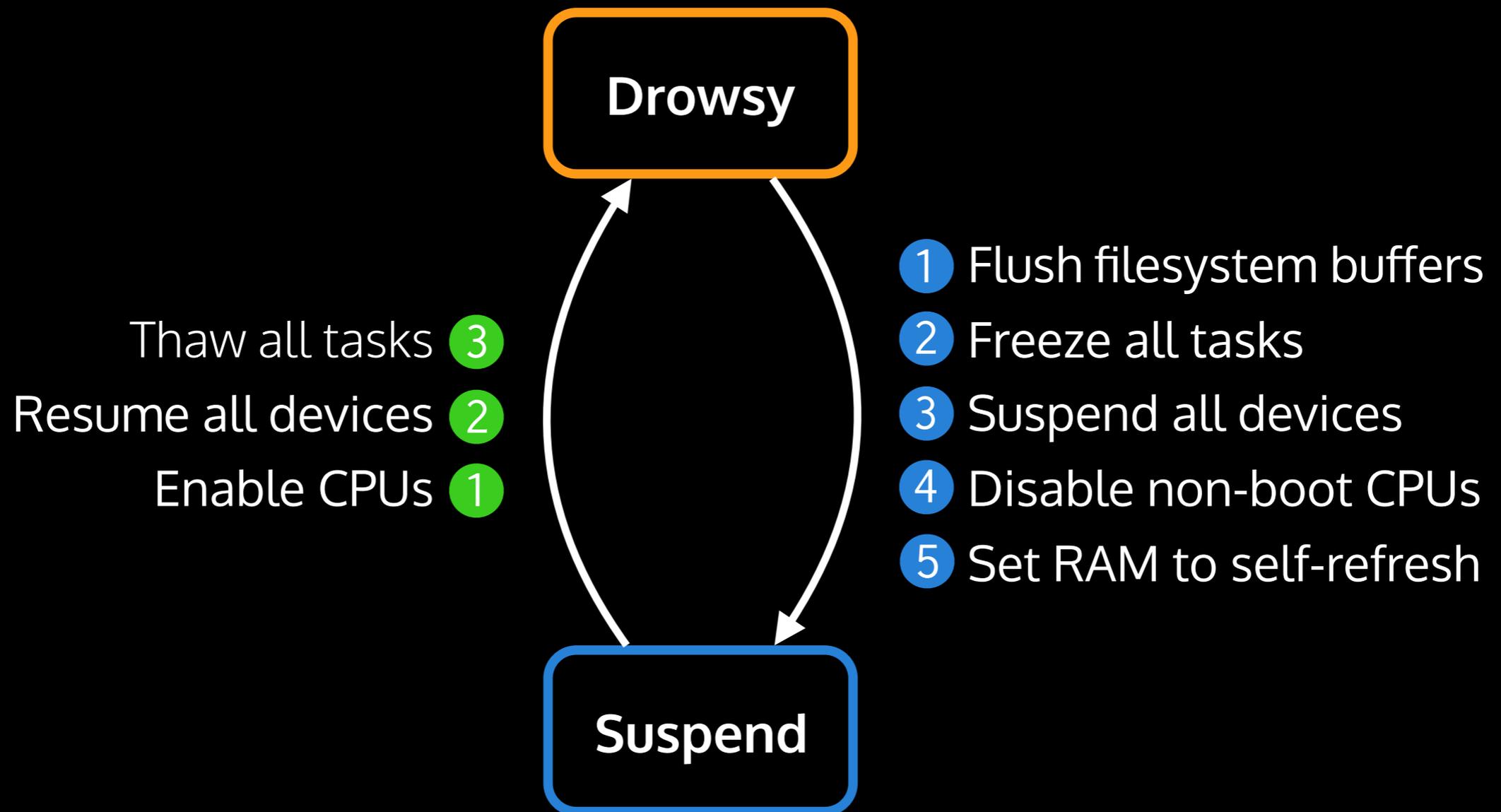
Drowsy Power Management

Construct **minimal “wake set”** of tasks and devices

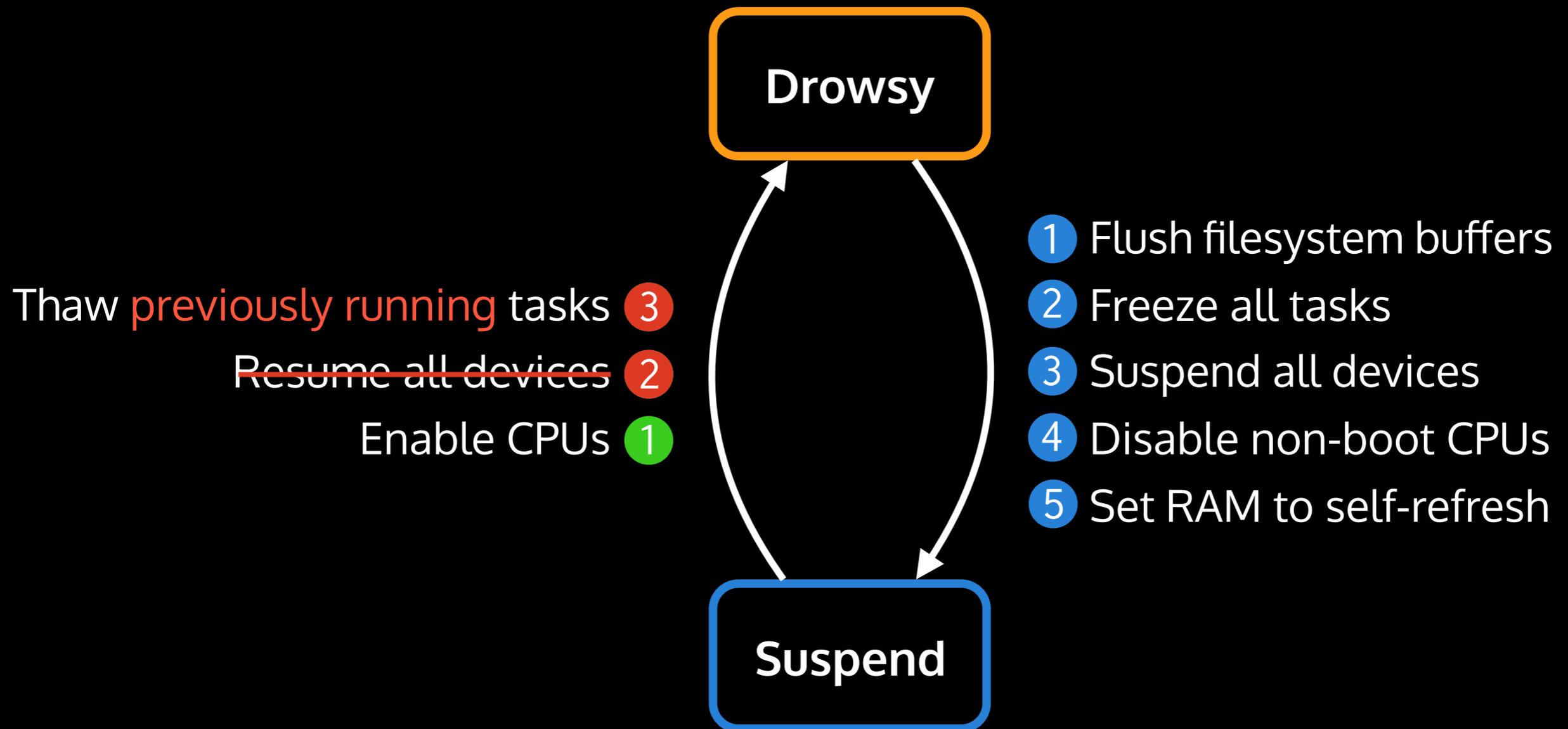
- Smallest set that **maintains correct behavior**
- Expand **on-demand** as event progresses

Constraint: No modifications to user-space

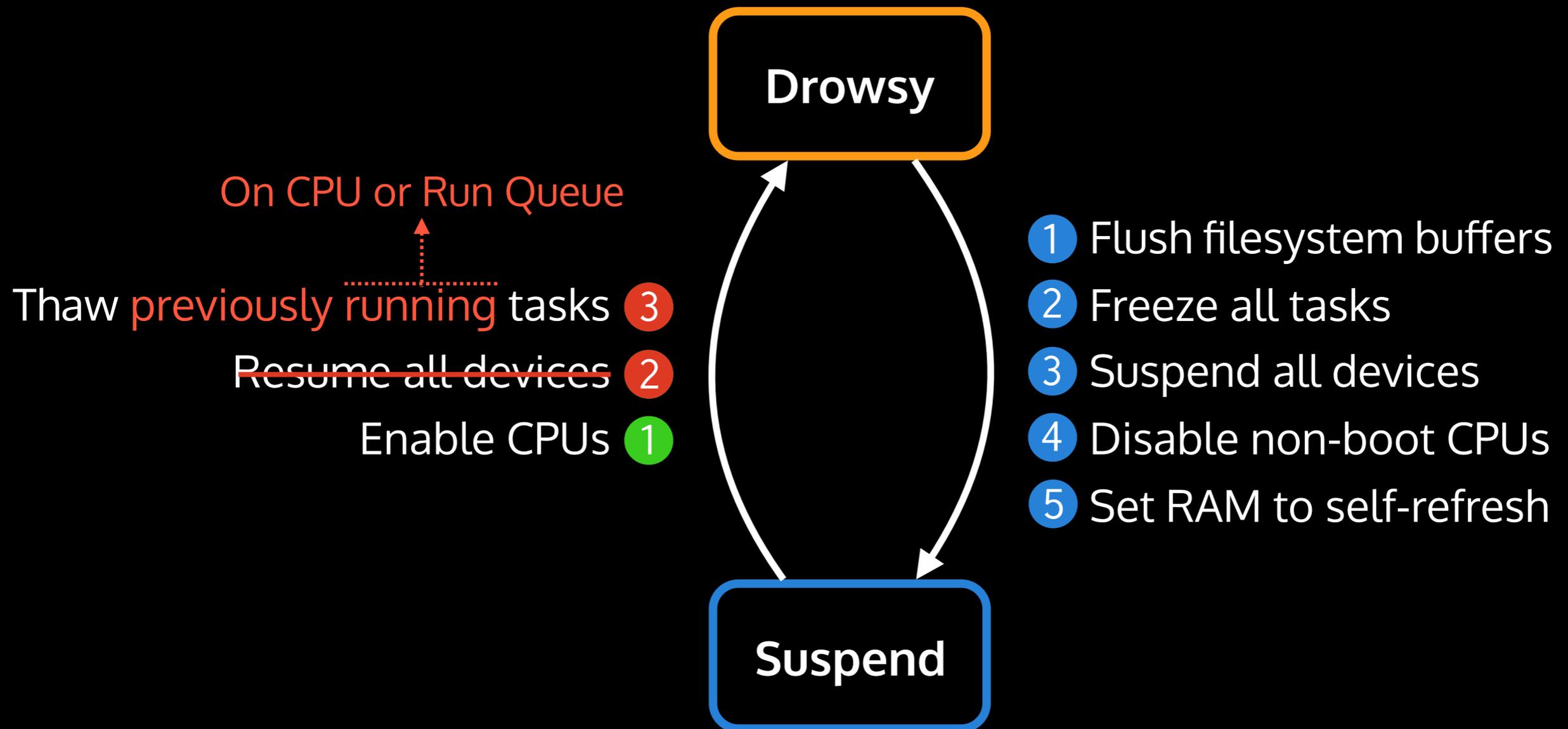
Transitions To/From Drowsy



Transitions To/From Drowsy



Transitions To/From Drowsy



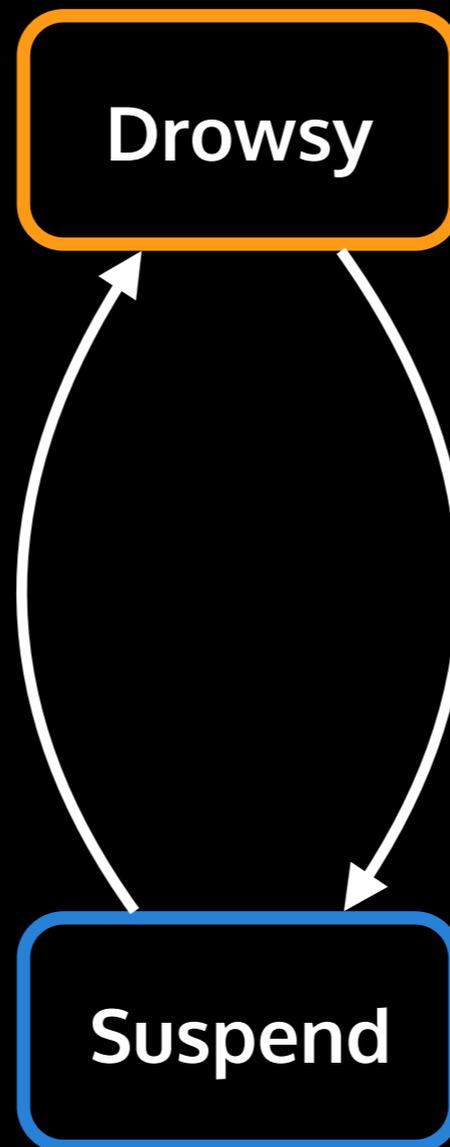
Transitions To/From Drowsy

Wake Set =
{ <Prev. Running Tasks> }

Thaw **previously running** tasks 3

~~Resume all devices~~ 2

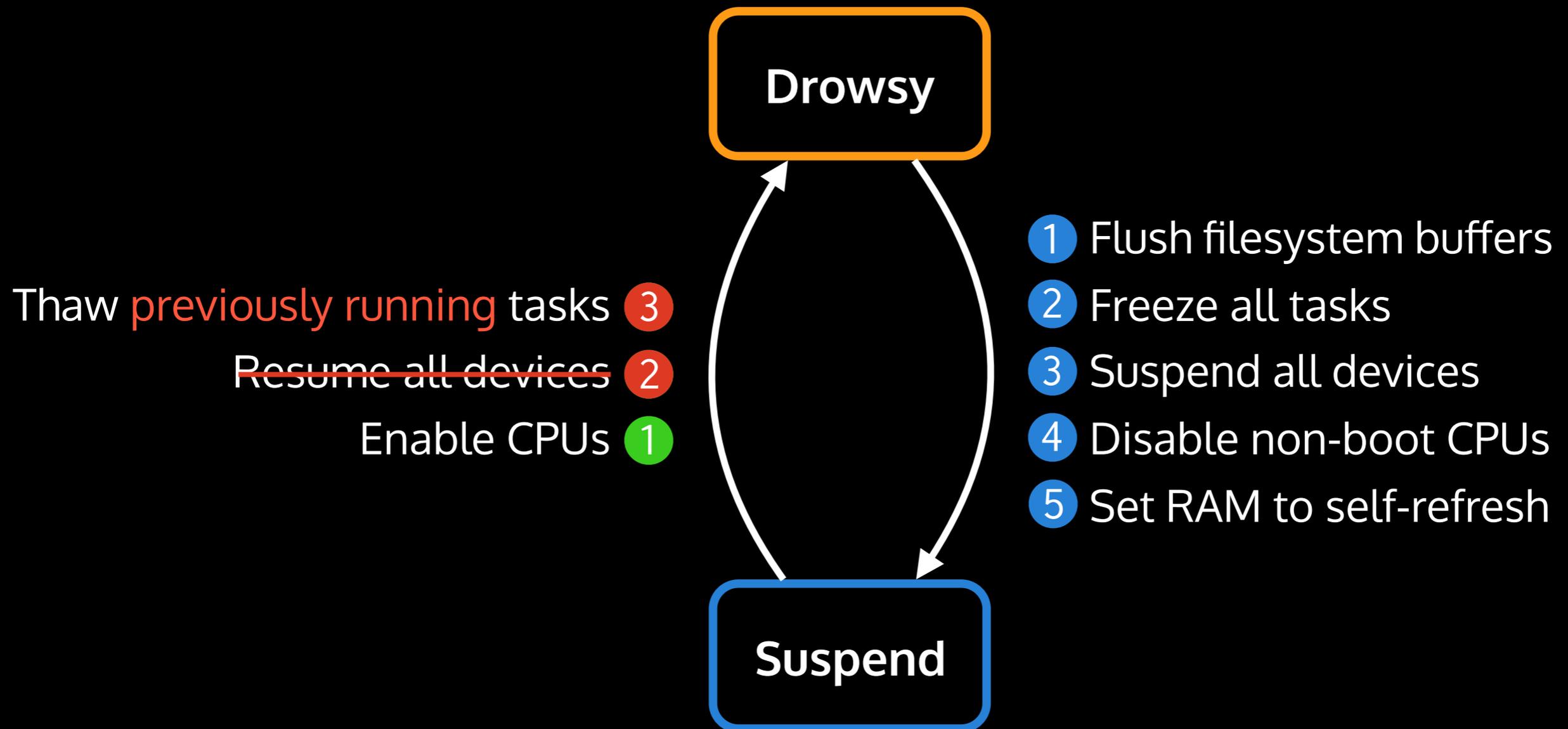
Enable CPUs 1



- 1 Flush filesystem buffers
- 2 Freeze all tasks
- 3 Suspend all devices
- 4 Disable non-boot CPUs
- 5 Set RAM to self-refresh

Transitions To/From Drowsy

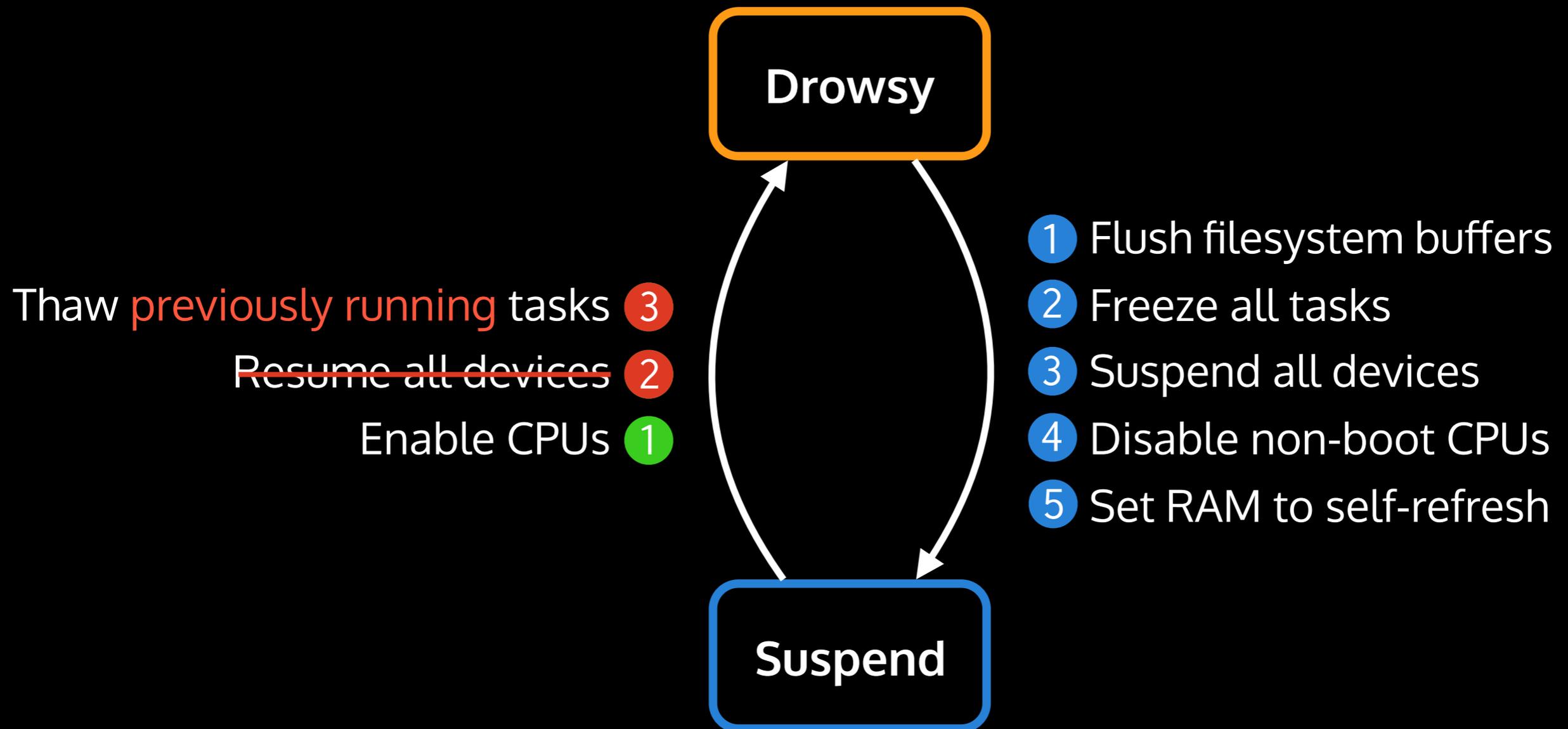
Wake Set =
{ <Prev. Running Tasks> }



Transitions To/From Drowsy

Wake Set =

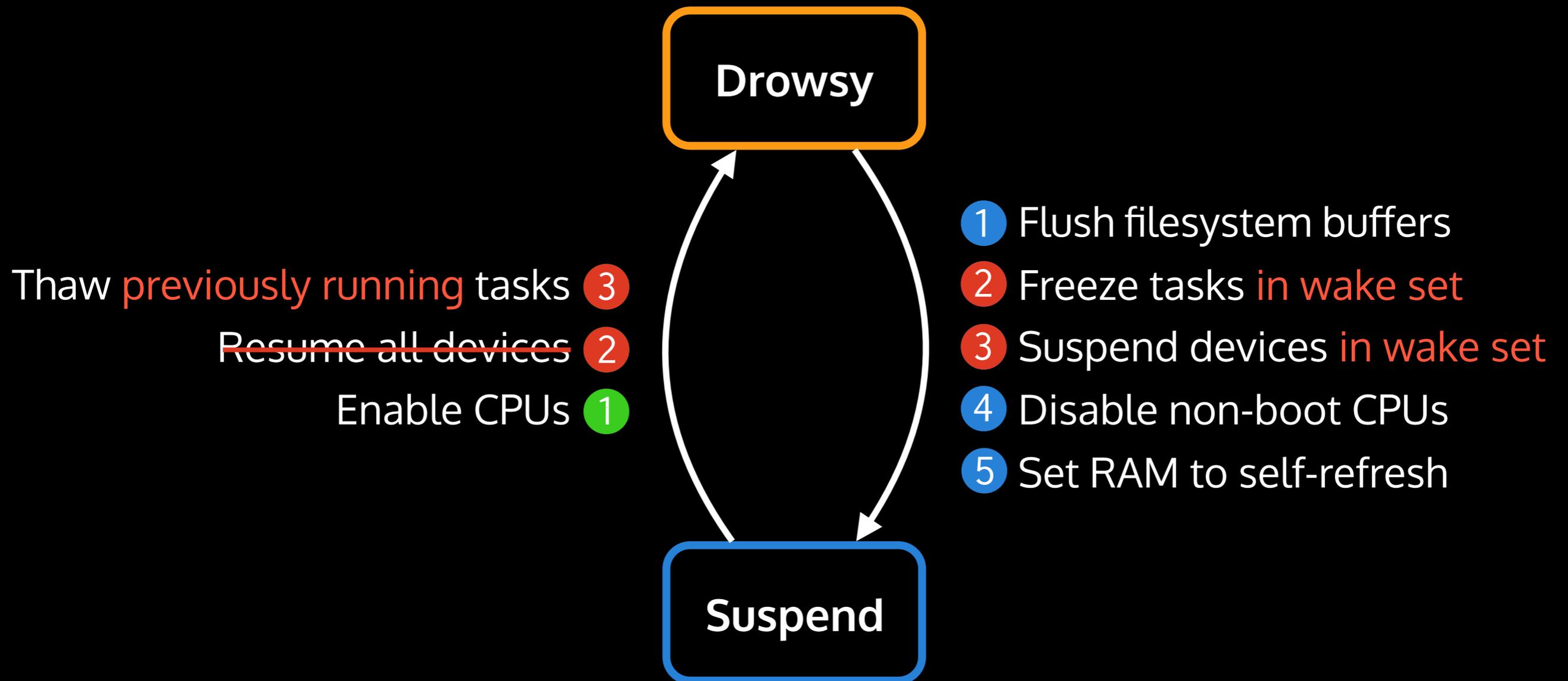
{ <Prev. Running Tasks>, Task, Device, ... }



Transitions To/From Drowsy

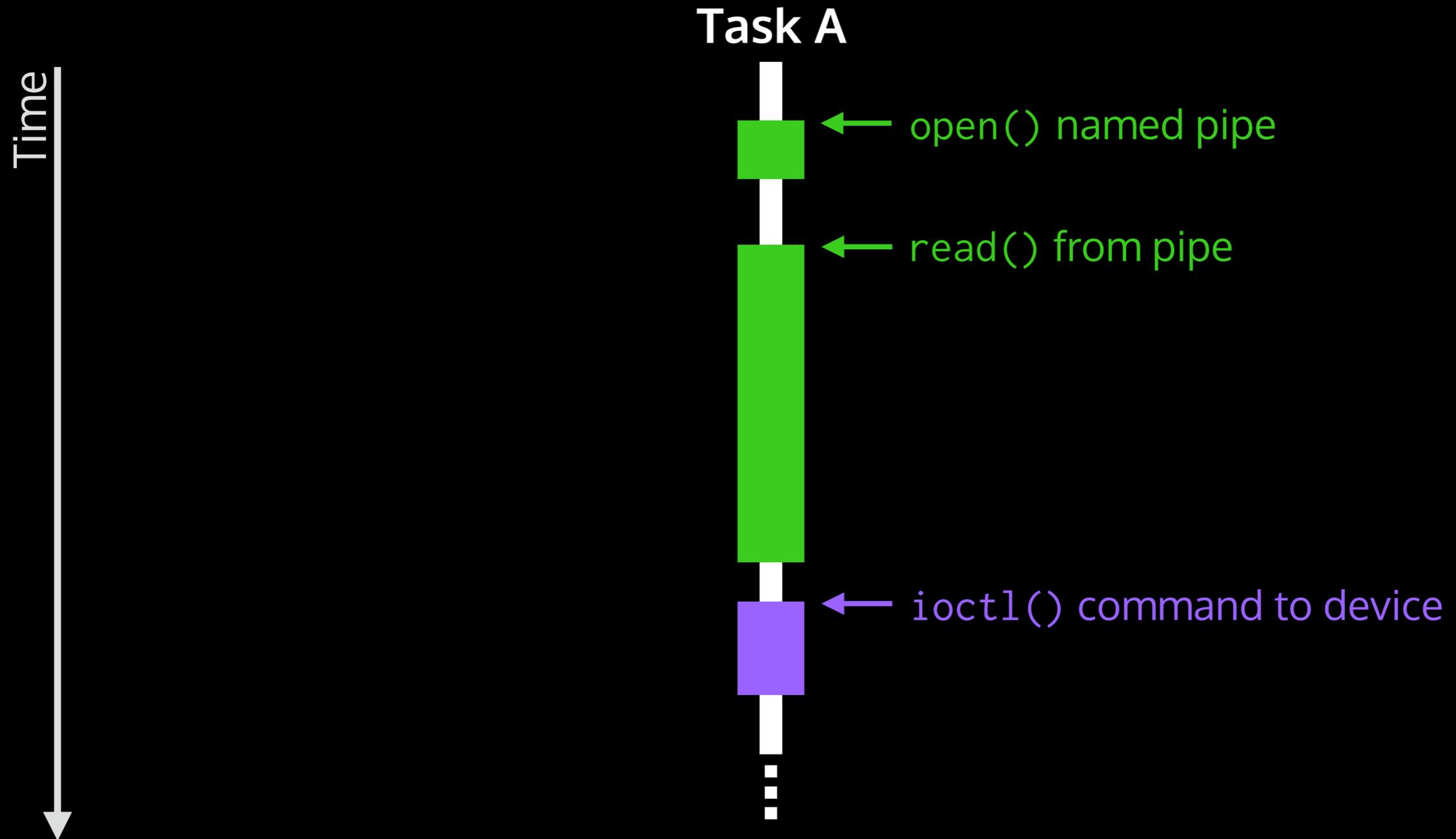
Wake Set =

{ <Prev. Running Tasks>, Task, Device, ... }



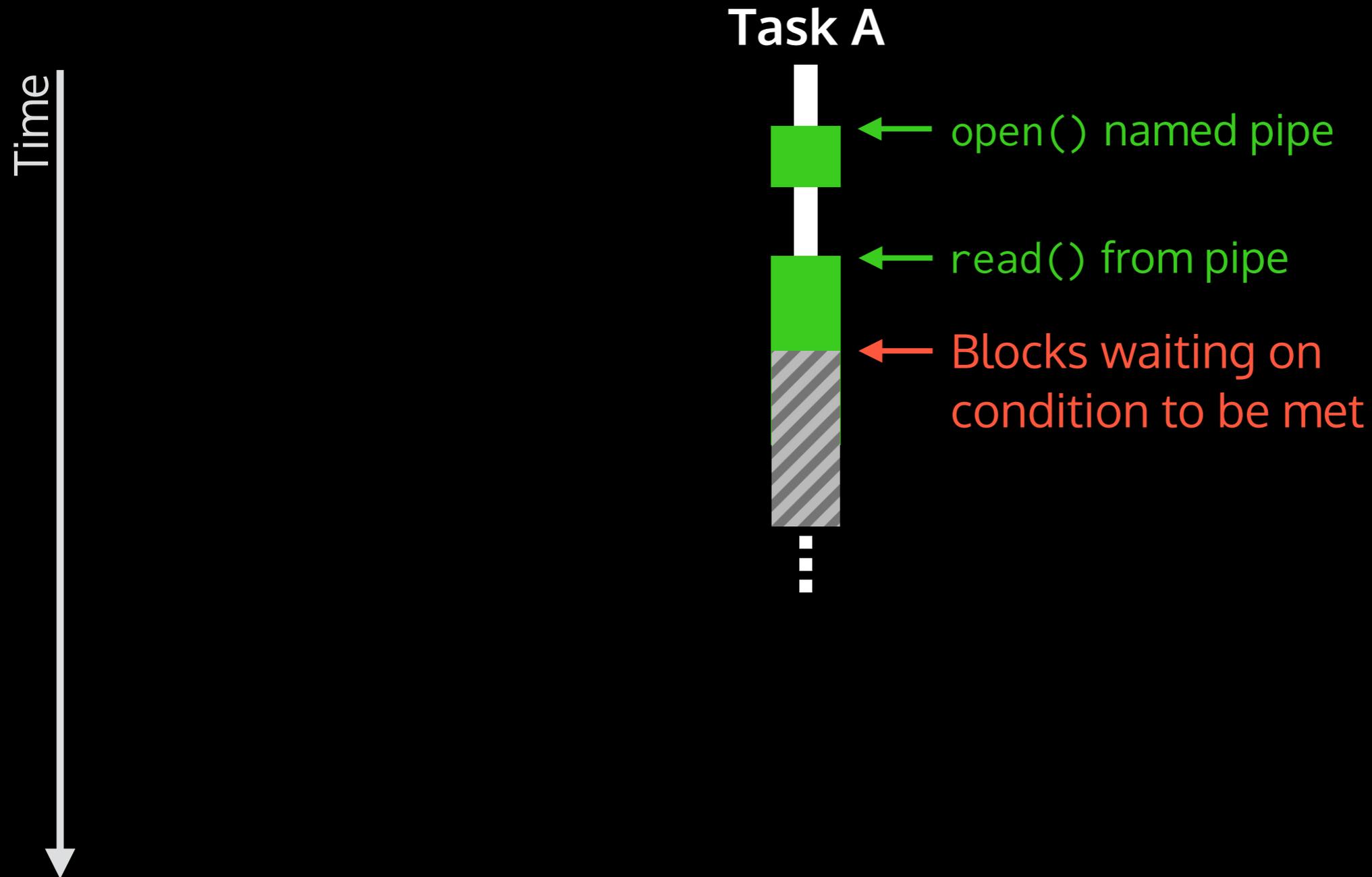
Constructing the Wake Set

Task States: Run (CPU) Run (I/O) Run (Device I/O)



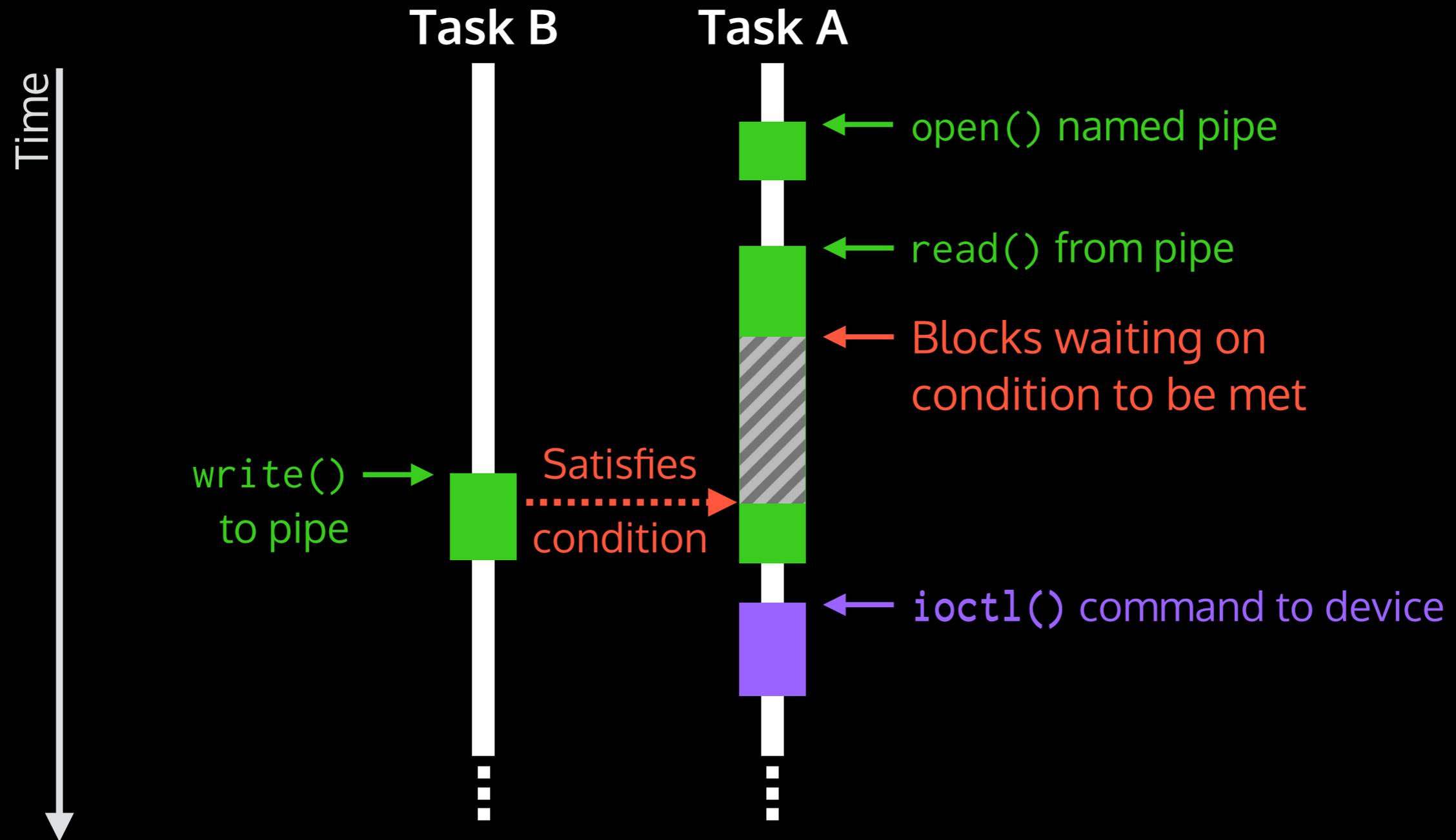
Constructing the Wake Set

Task States:  Run (CPU)  Run (I/O)  Run (Device I/O)  Blocked



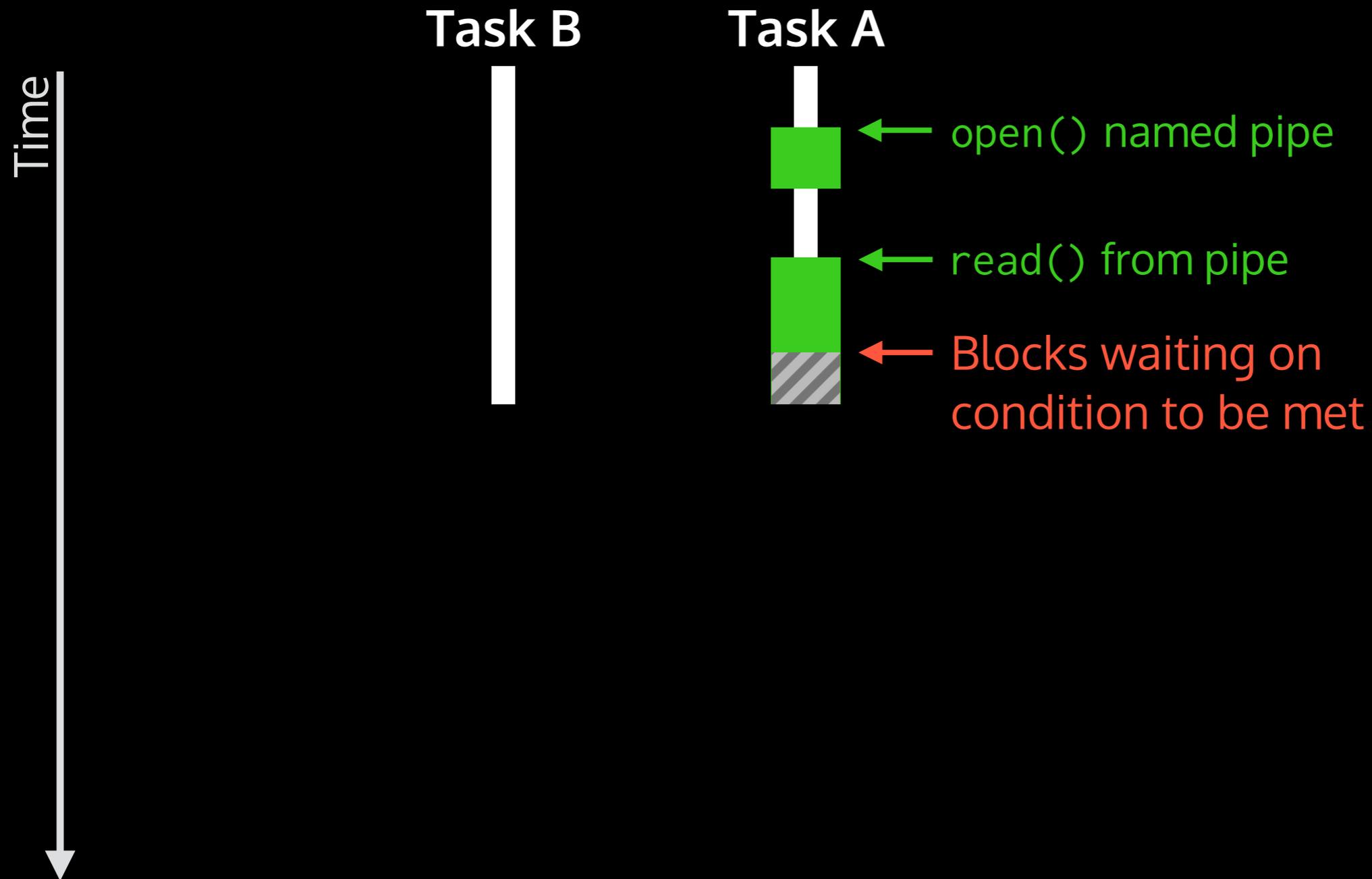
Constructing the Wake Set

Task States:  Run (CPU)  Run (I/O)  Run (Device I/O)  Blocked



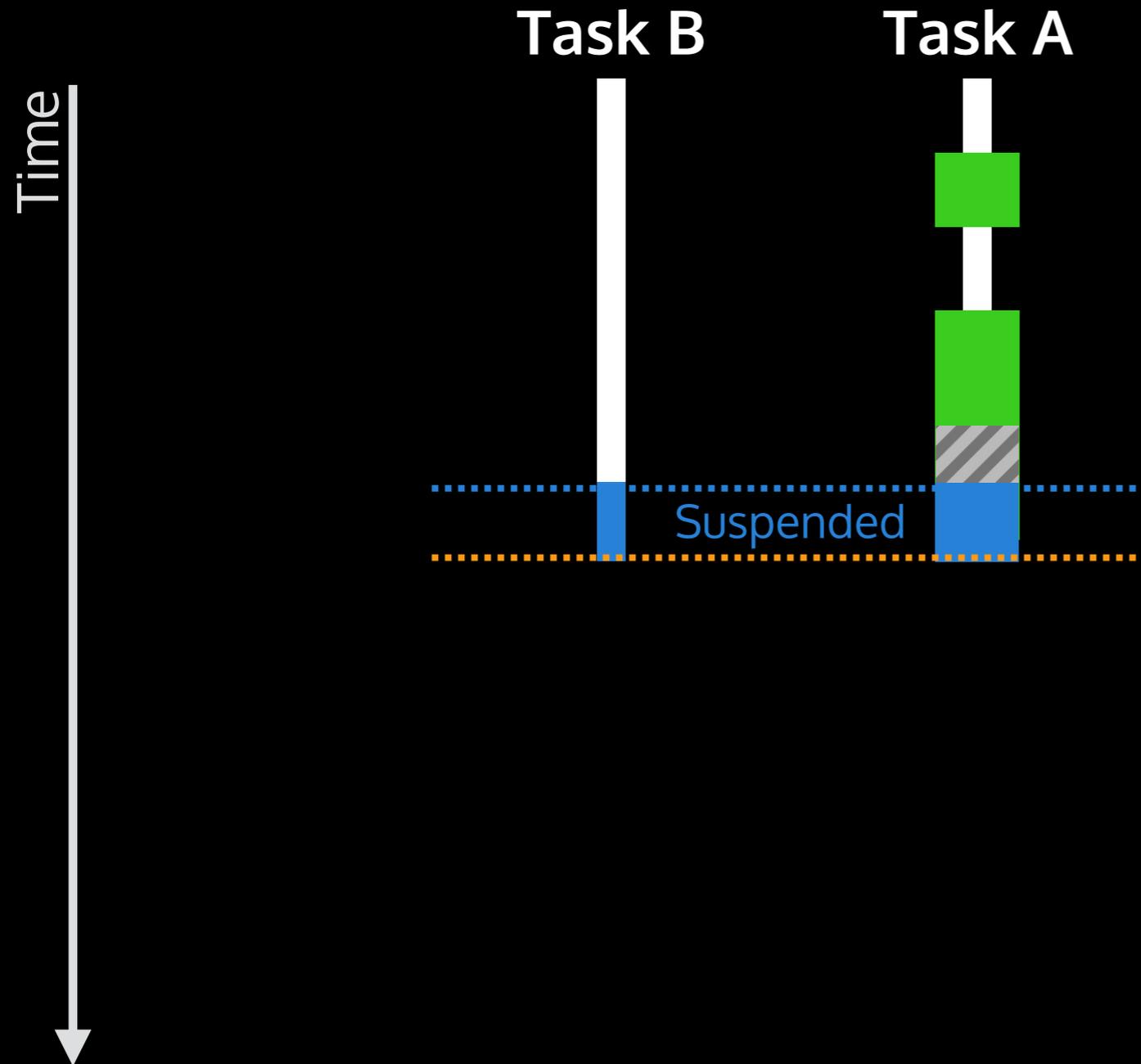
Constructing the Wake Set

Task States:  Run (CPU)  Run (I/O)  Run (Device I/O)  Blocked



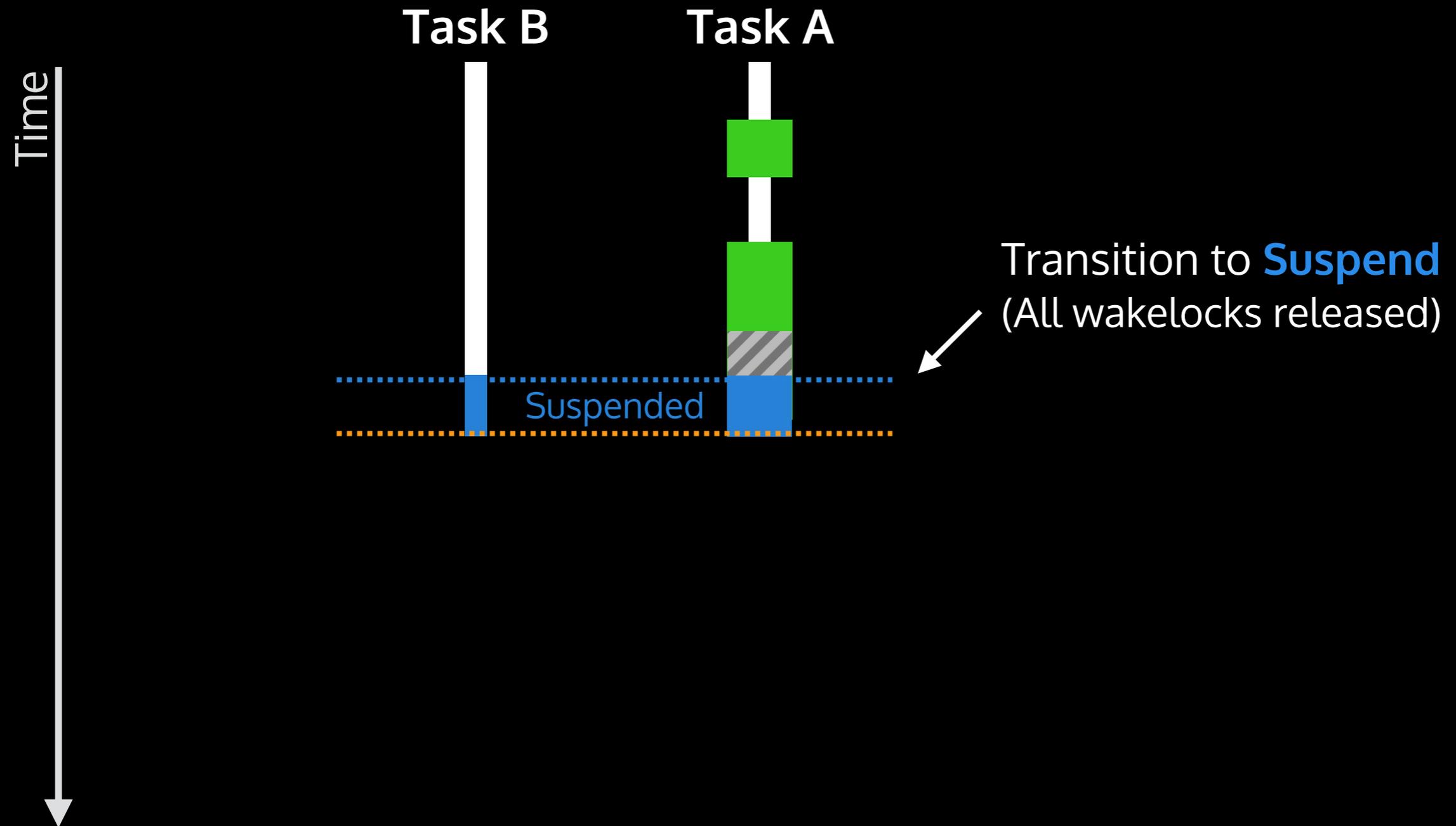
Constructing the Wake Set

Task States:  Run (CPU)  Run (I/O)  Run (Device I/O)  Blocked  Frozen



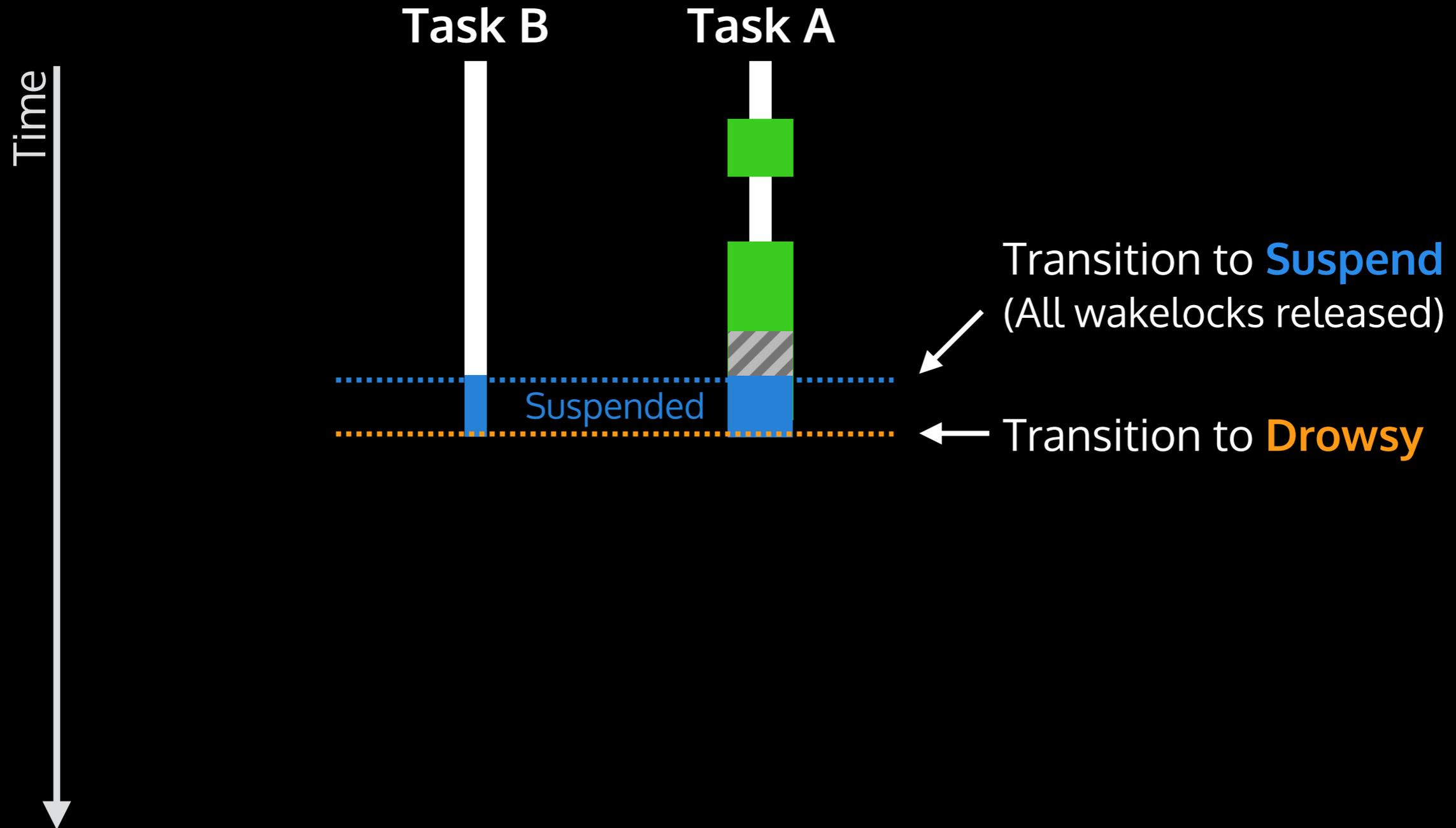
Constructing the Wake Set

Task States:  Run (CPU)  Run (I/O)  Run (Device I/O)  Blocked  Frozen



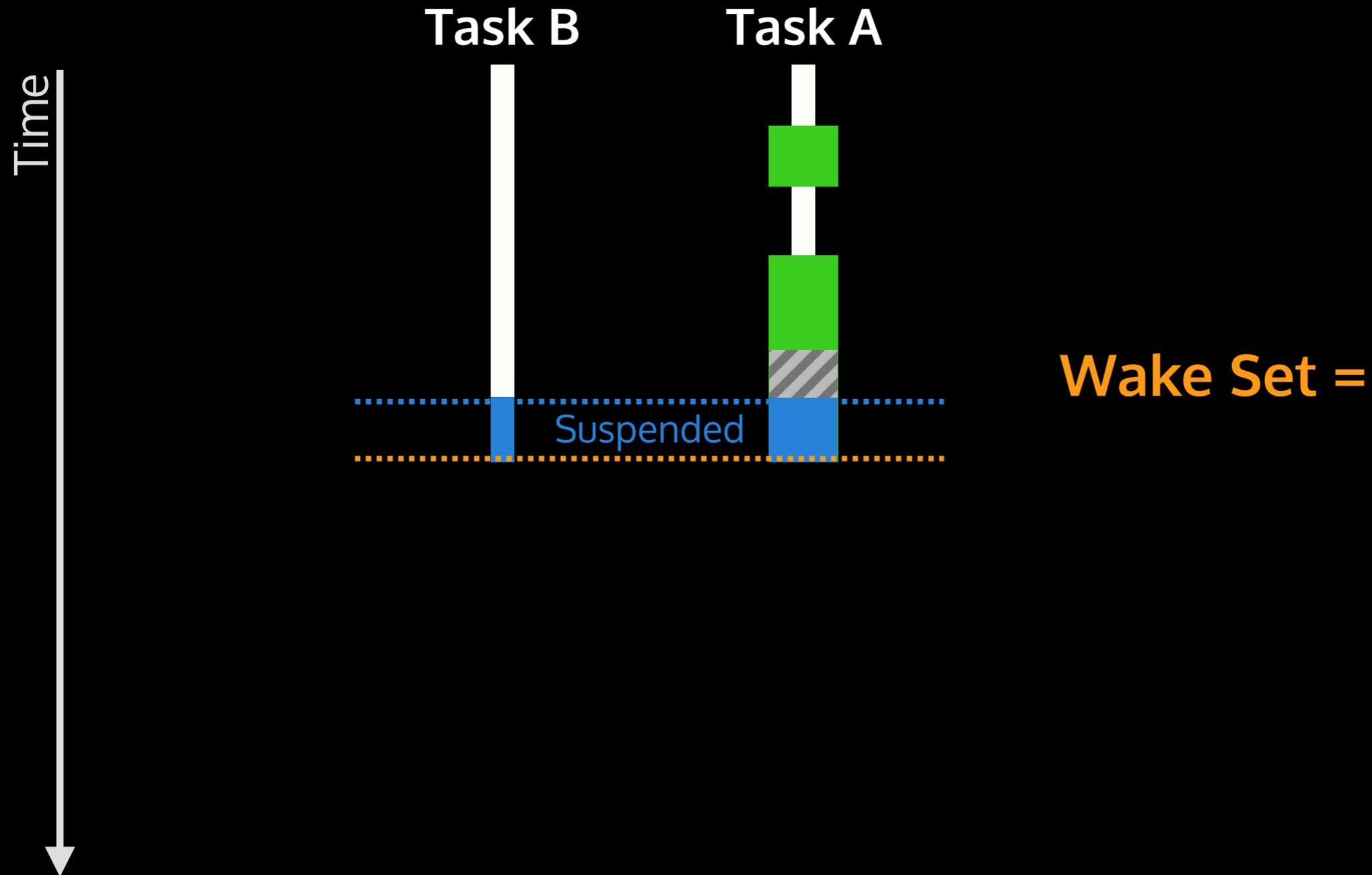
Constructing the Wake Set

Task States: Run (CPU) Run (I/O) Run (Device I/O) Blocked Frozen



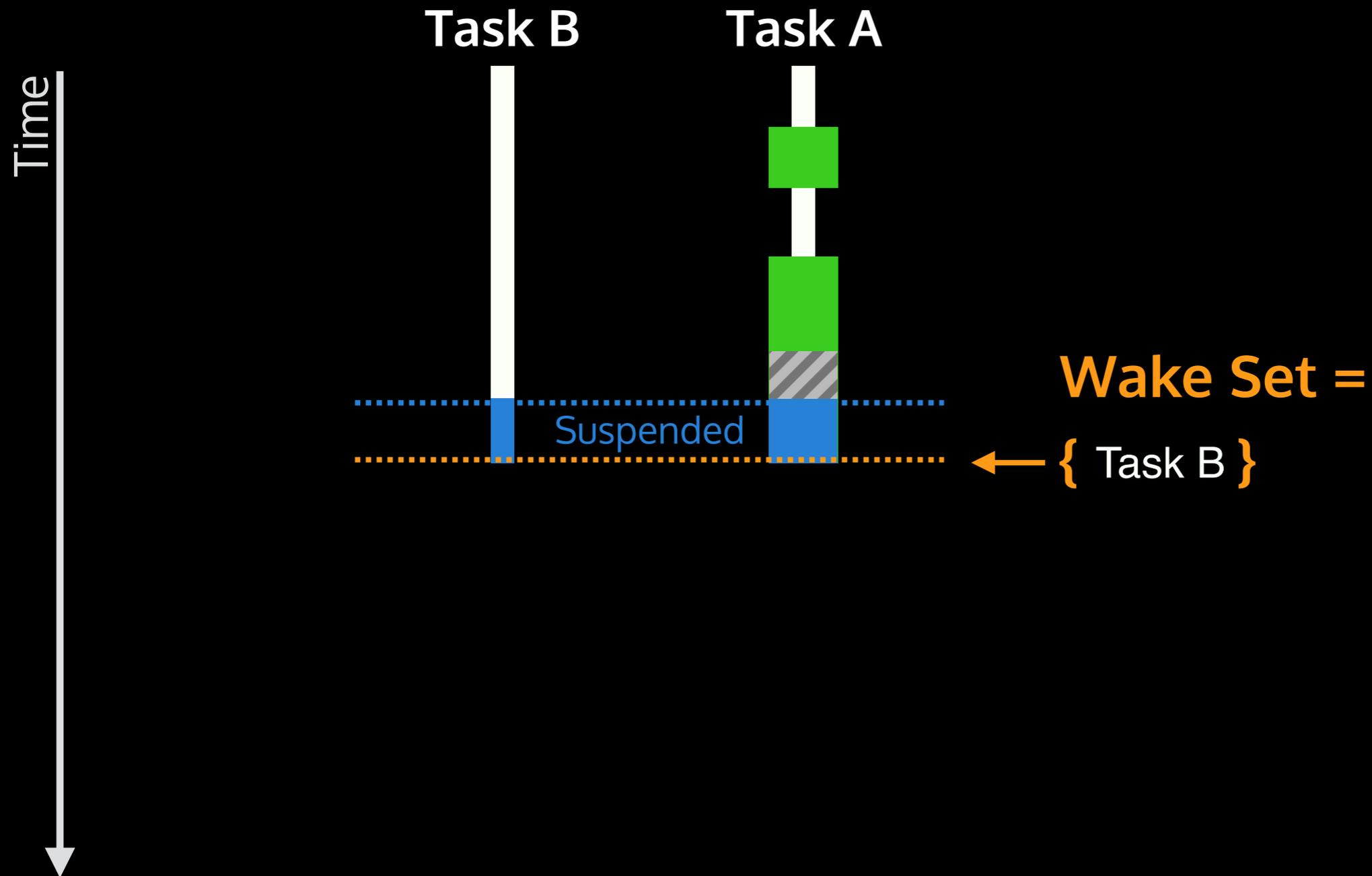
Constructing the Wake Set

Task States:  Run (CPU)  Run (I/O)  Run (Device I/O)  Blocked  Frozen



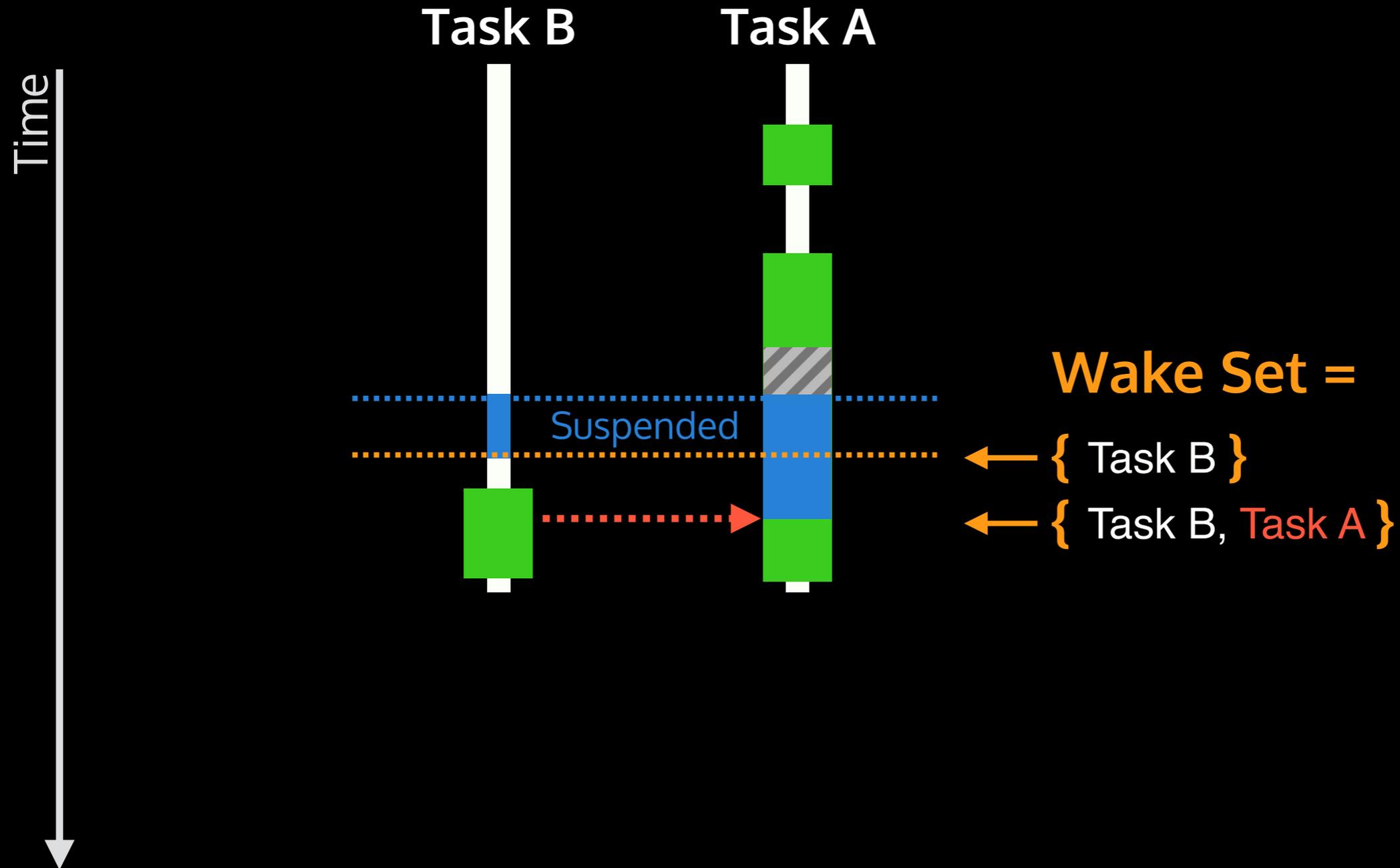
Constructing the Wake Set

Task States:  Run (CPU)  Run (I/O)  Run (Device I/O)  Blocked  Frozen



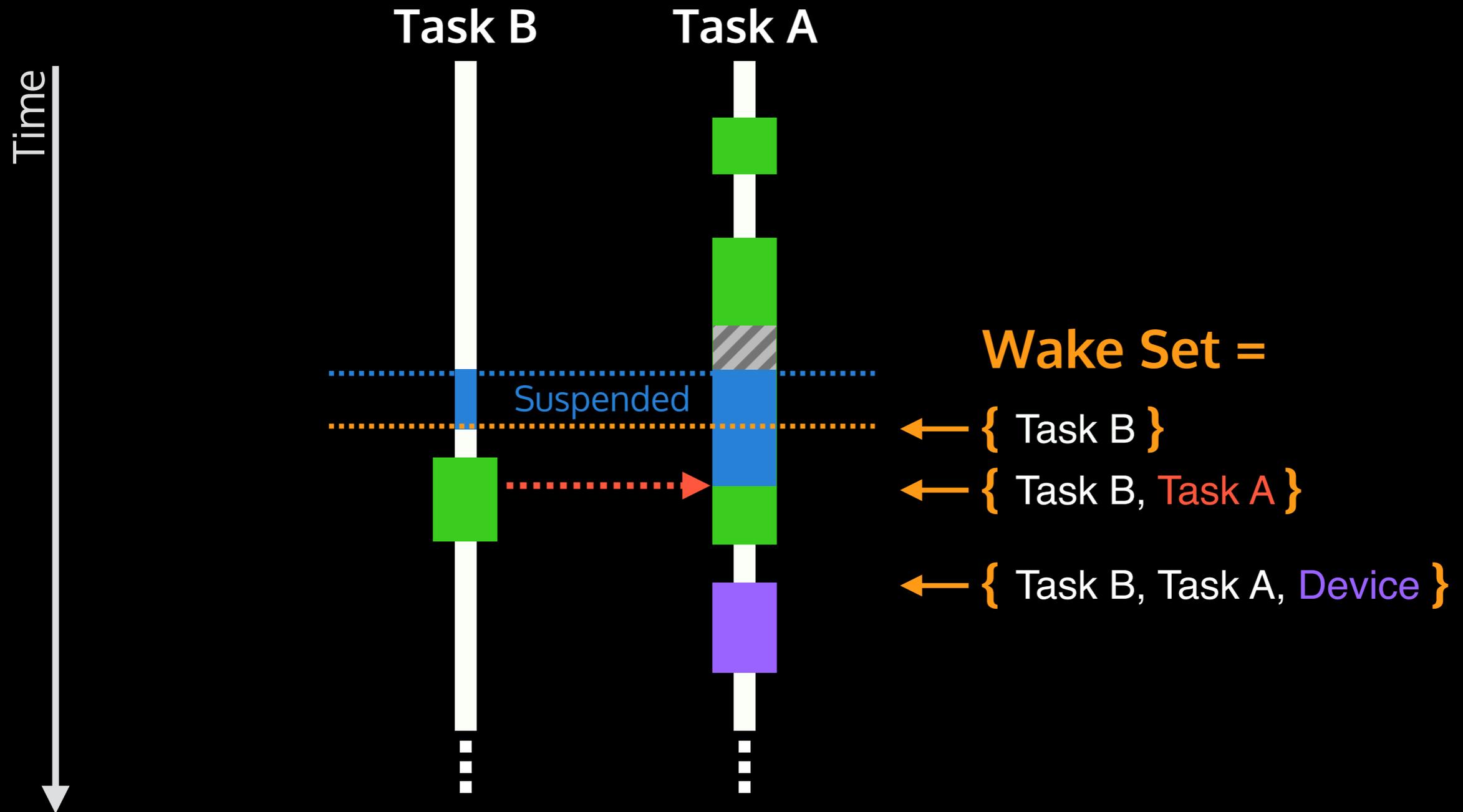
Constructing the Wake Set

Task States: Run (CPU) Run (I/O) Run (Device I/O) Blocked Frozen



Constructing the Wake Set

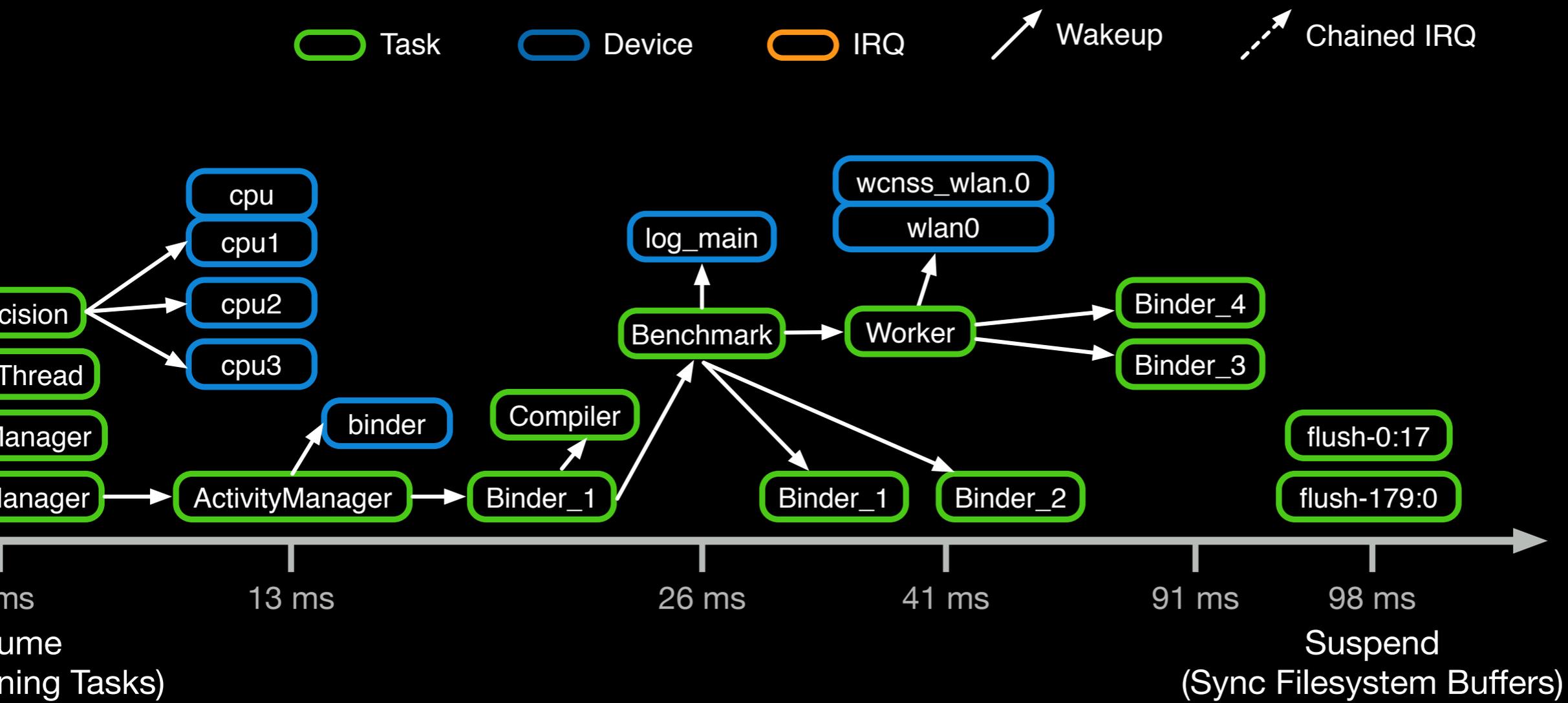
Task States: Run (CPU) Run (I/O) Run (Device I/O) Blocked Frozen



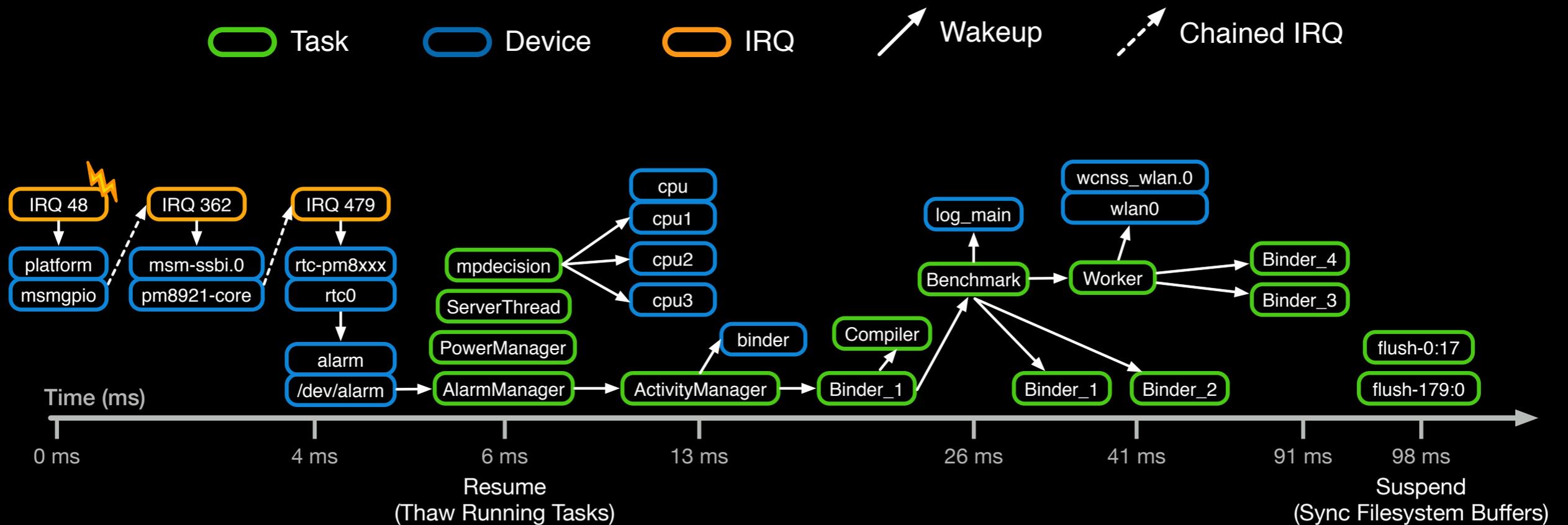
Drowsy Wake Set: Pull Data



Drowsy Wake Set: Pull Data



Drowsy Wake Set: Pull Data



15 tasks thawed & 16 devices resumed
1.8% 1.9%

Implementation



Implemented Drowsy within Android kernel

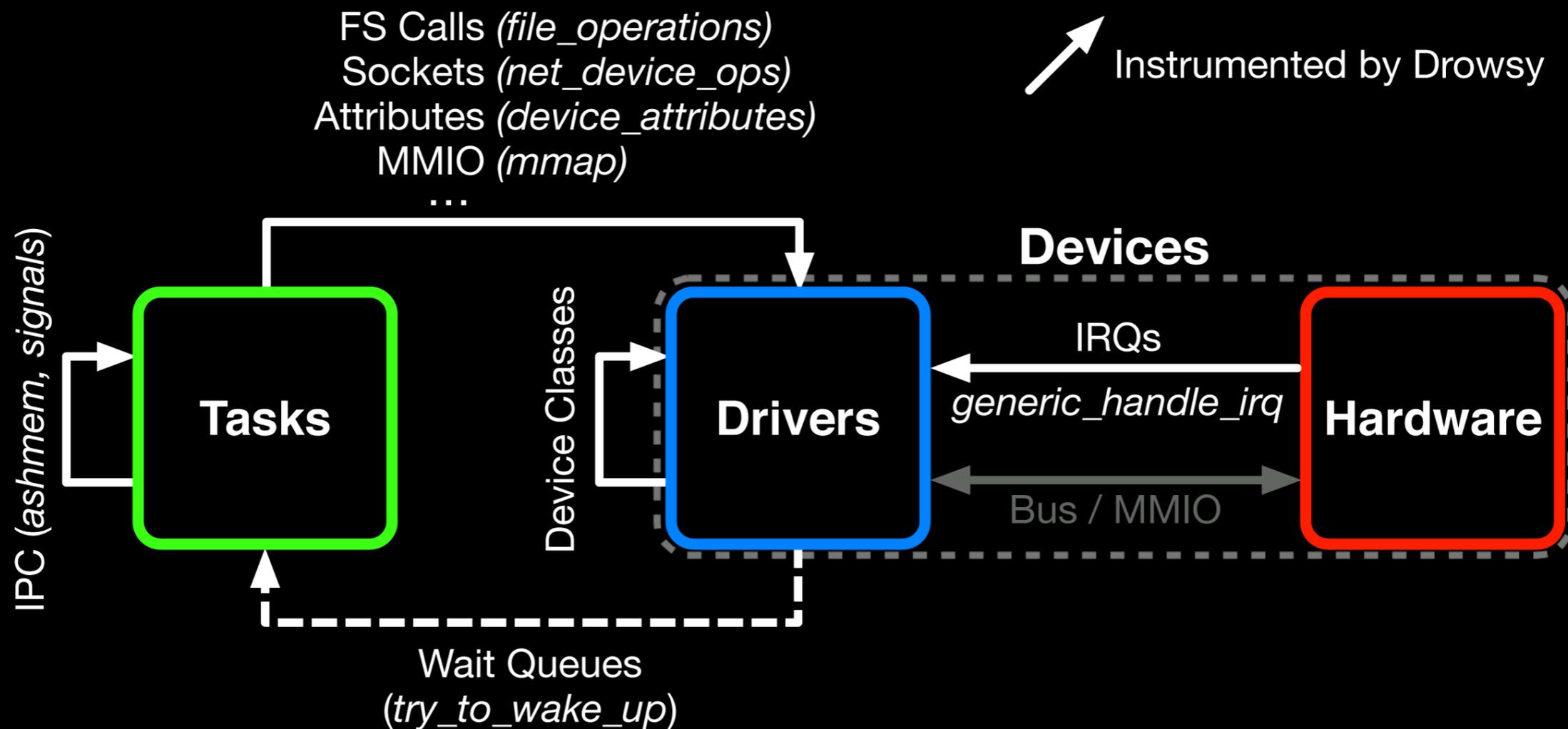
Platform: Google Nexus 4 (“Mako”)

Version: 4.2.2 (Fork of Linux 3.4)

SLOC: ~4,600

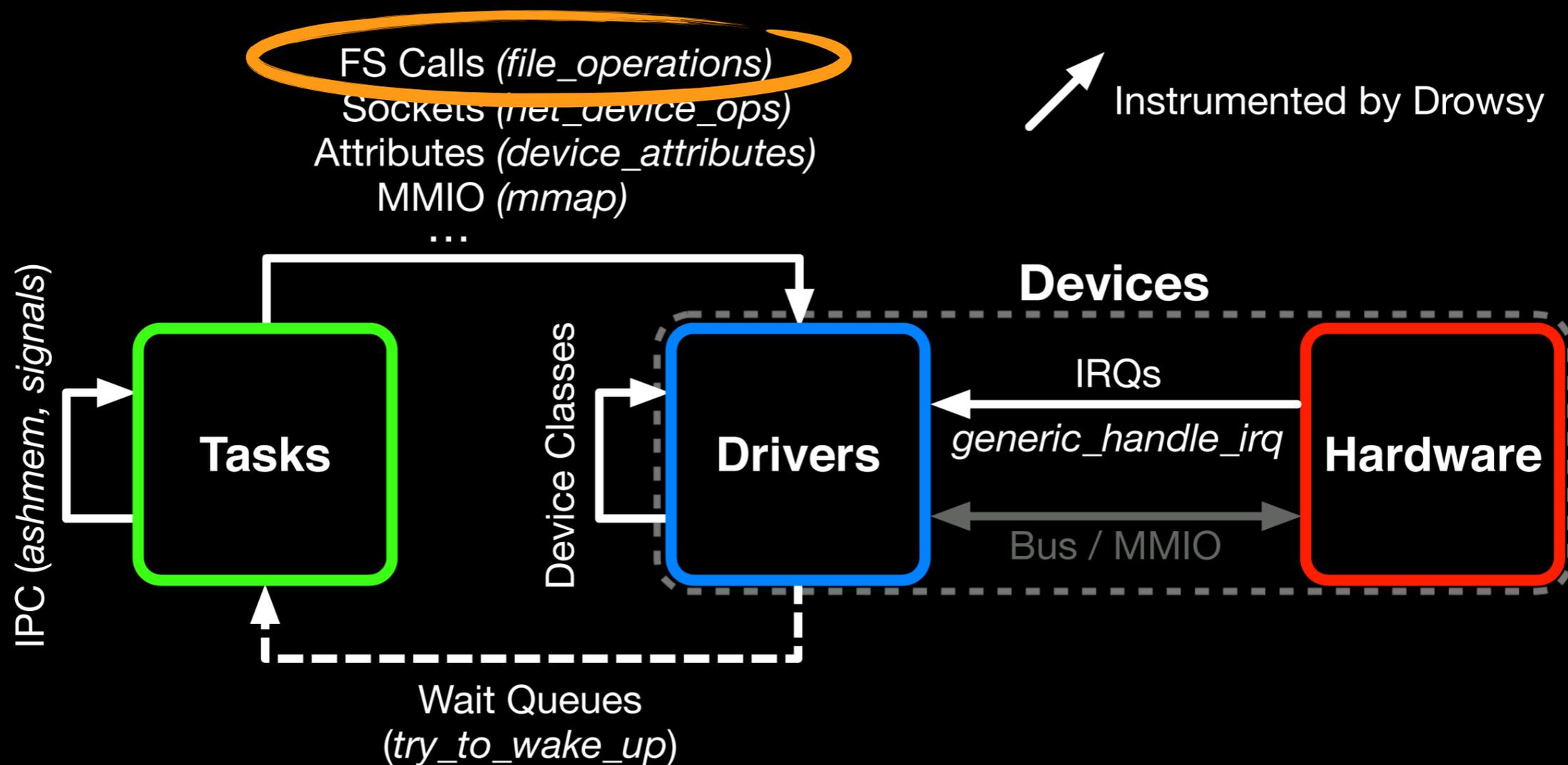
Instrumenting Android

For determining when to add tasks/devices to wake set

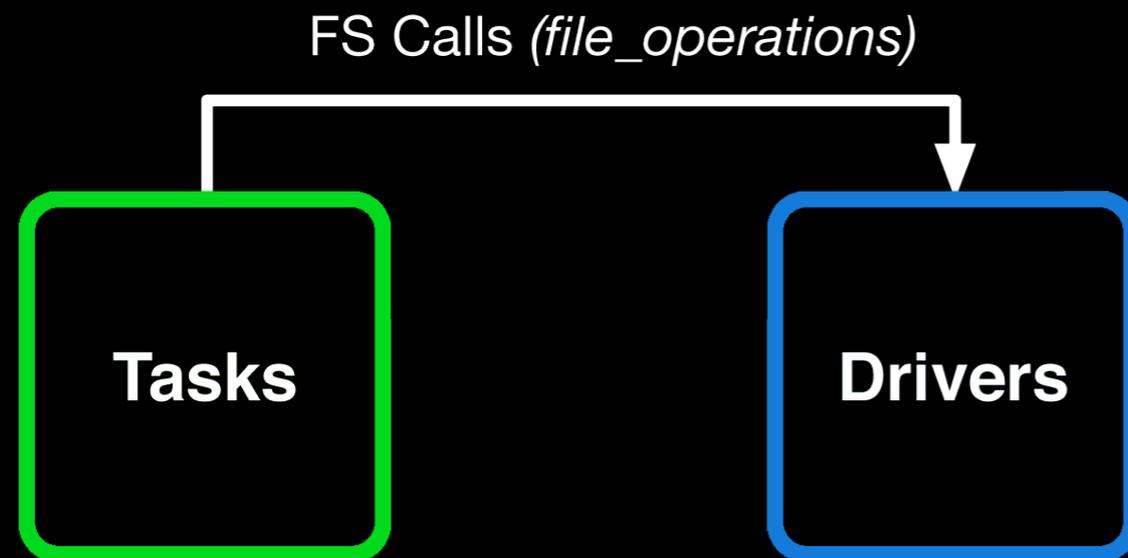


Instrumenting Android

For determining when to add tasks/devices to wake set

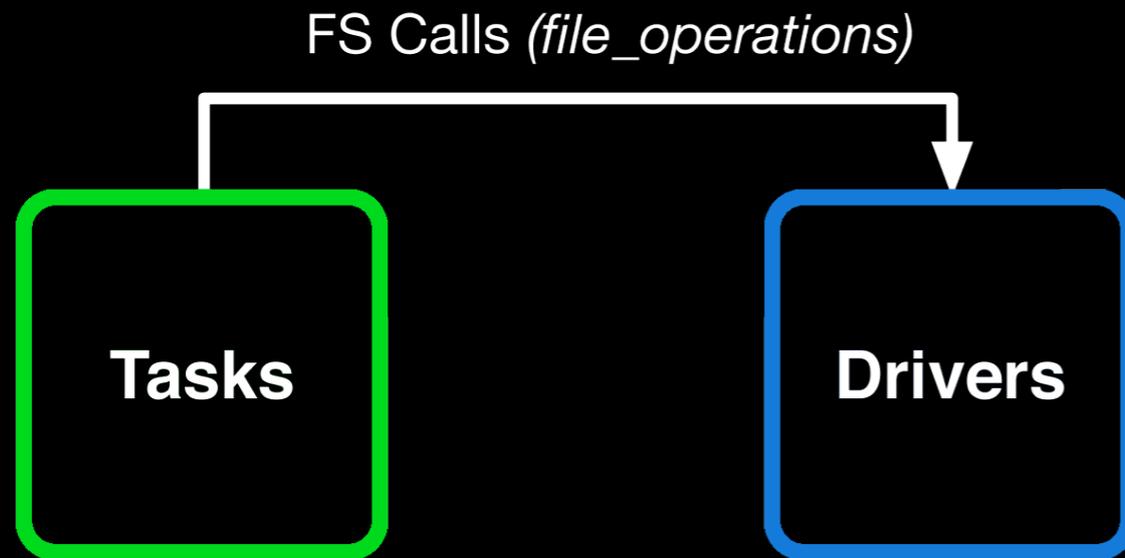


Example: file_operations



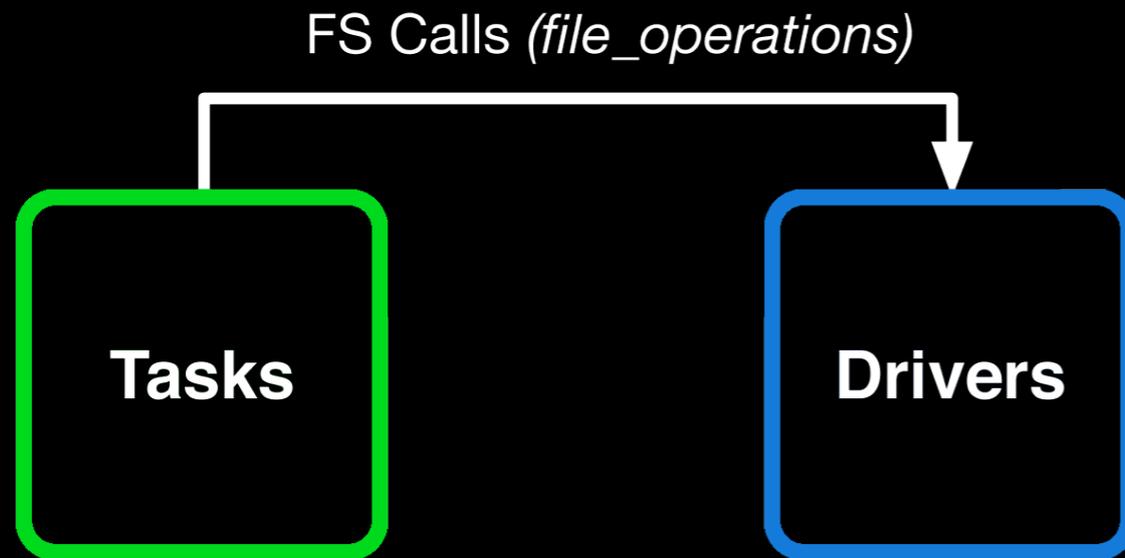
```
struct file_operations d_fops:  
    .open = &drv_open  
    .read = &drv_read  
    .write = &drv_write
```

Example: file_operations



```
struct file_operations d_fops:  
    .open = &drv_open  
    .read = &drv_read  
    .write = &drowsy_write  
    .write_impl = &drv_write
```

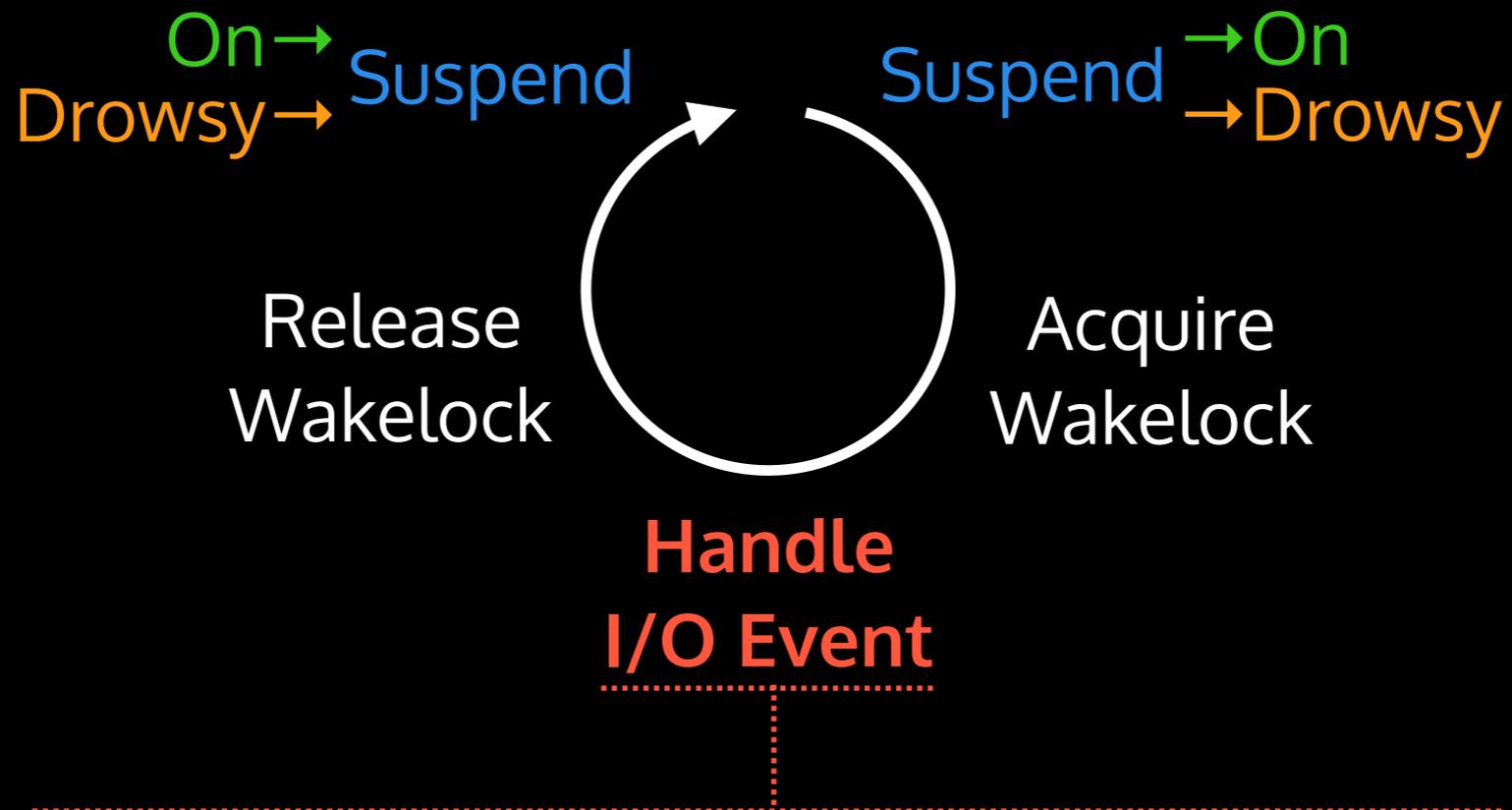
Example: file_operations



```
struct file_operations d_fops:  
    .open = &drv_open  
    .read = &drv_read  
    .write = &drowsy_write  
    .write_impl = &drv_write
```

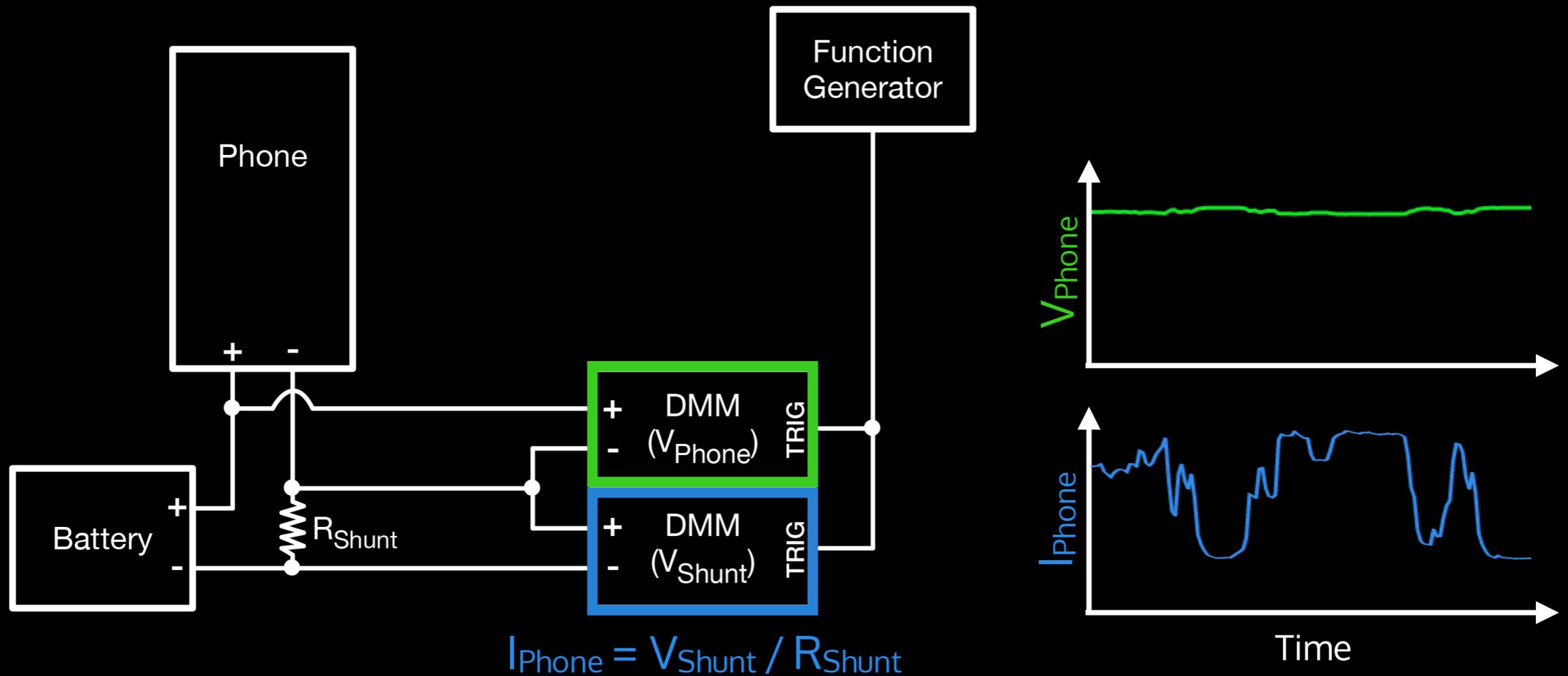
```
int drowsy_write(File *f, ...)  
    Device *d = fileToDevice(f)  
    if(d->state == Suspended)  
        resumeDevice(d)  
    f->f_ops.write_impl(f, ...)
```

Evaluation: Benchmarking

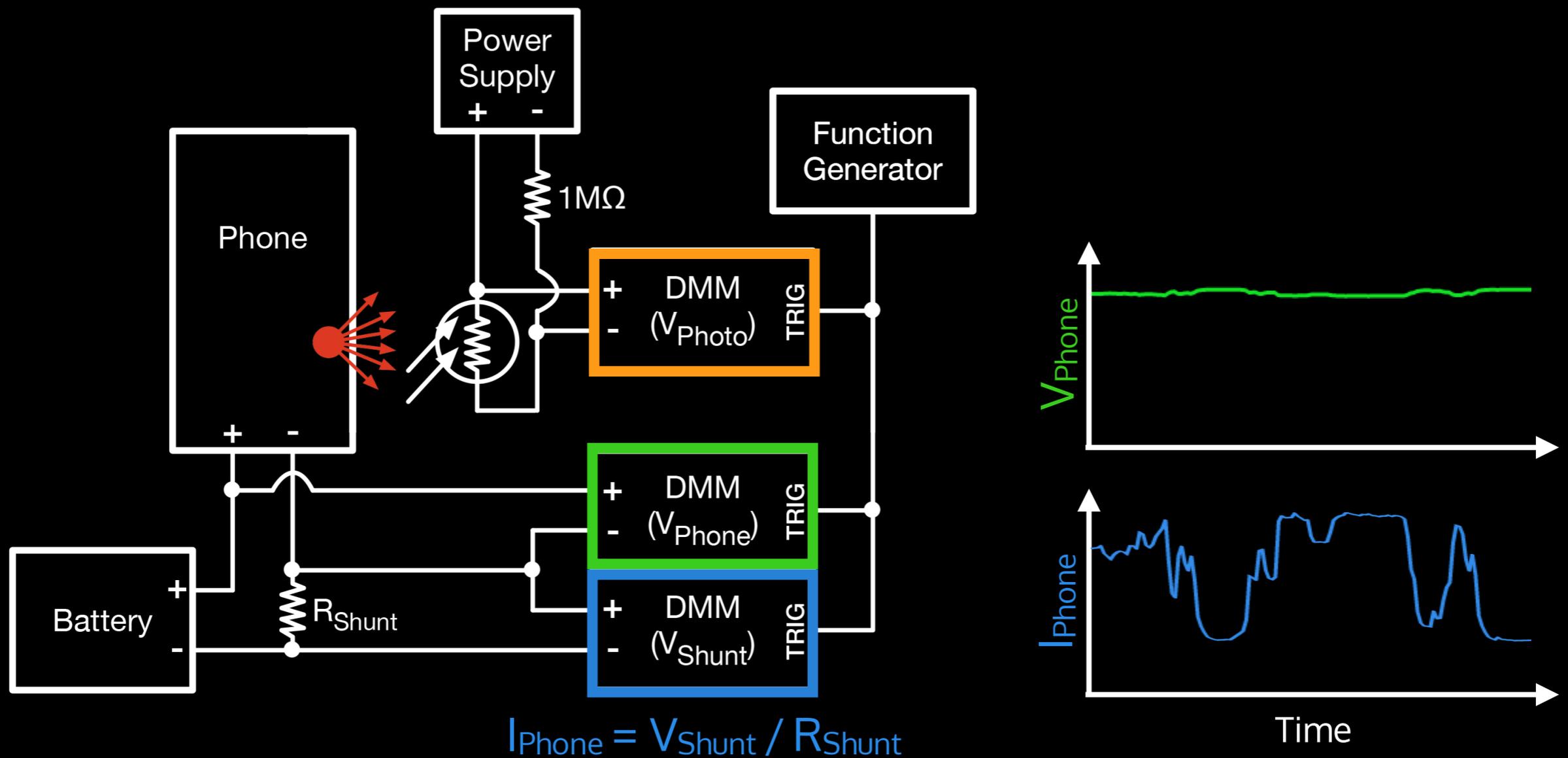


- ALM** Set a future wakeup alarm
- BT2** Accept incoming Bluetooth connection
- PUL** Fetch weather update (and set alarm)
- PSH** Receive incoming push notification
- SEN** Sample the accelerometer (and set alarm)

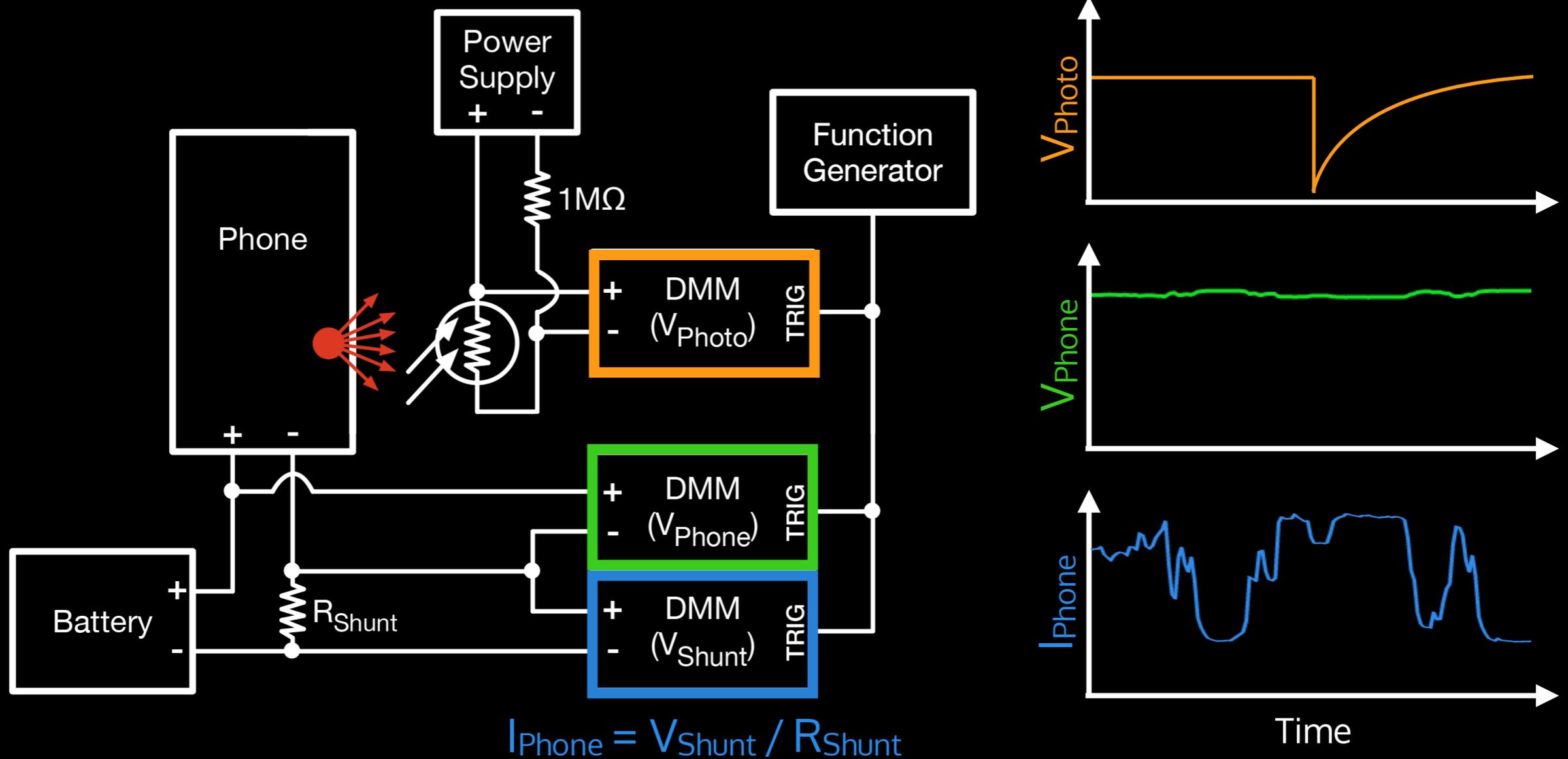
Evaluation: Measurement



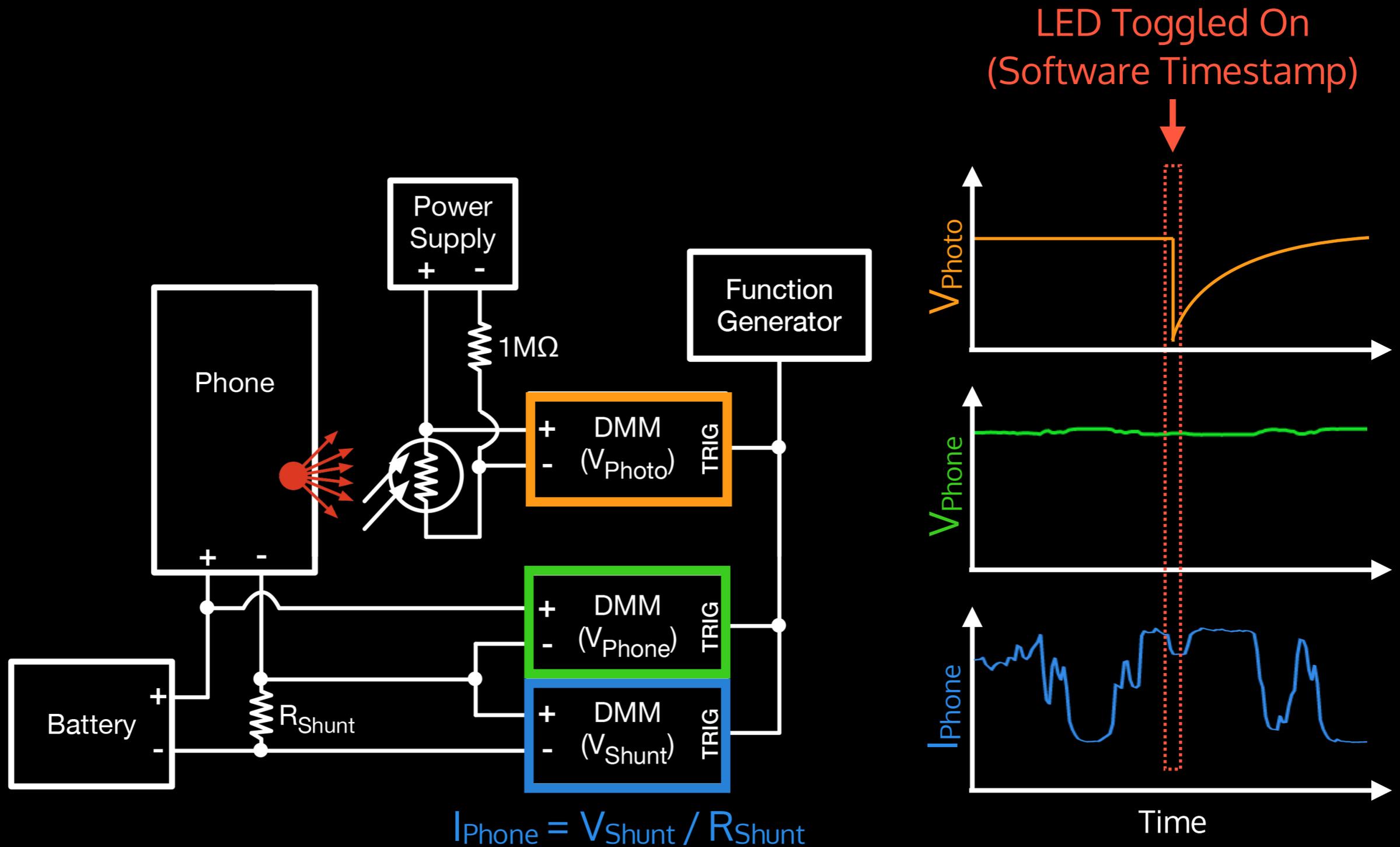
Evaluation: Measurement



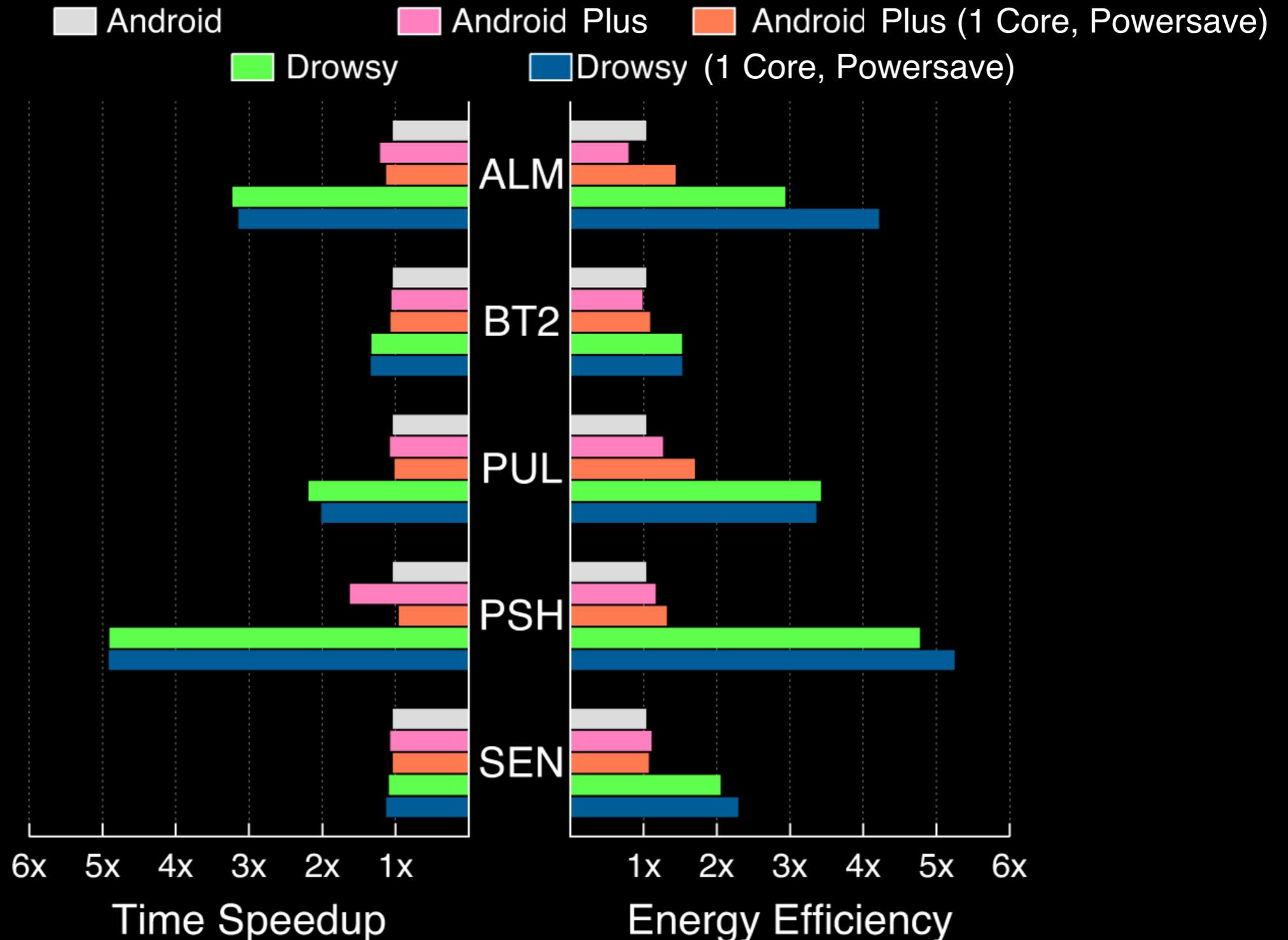
Evaluation: Measurement



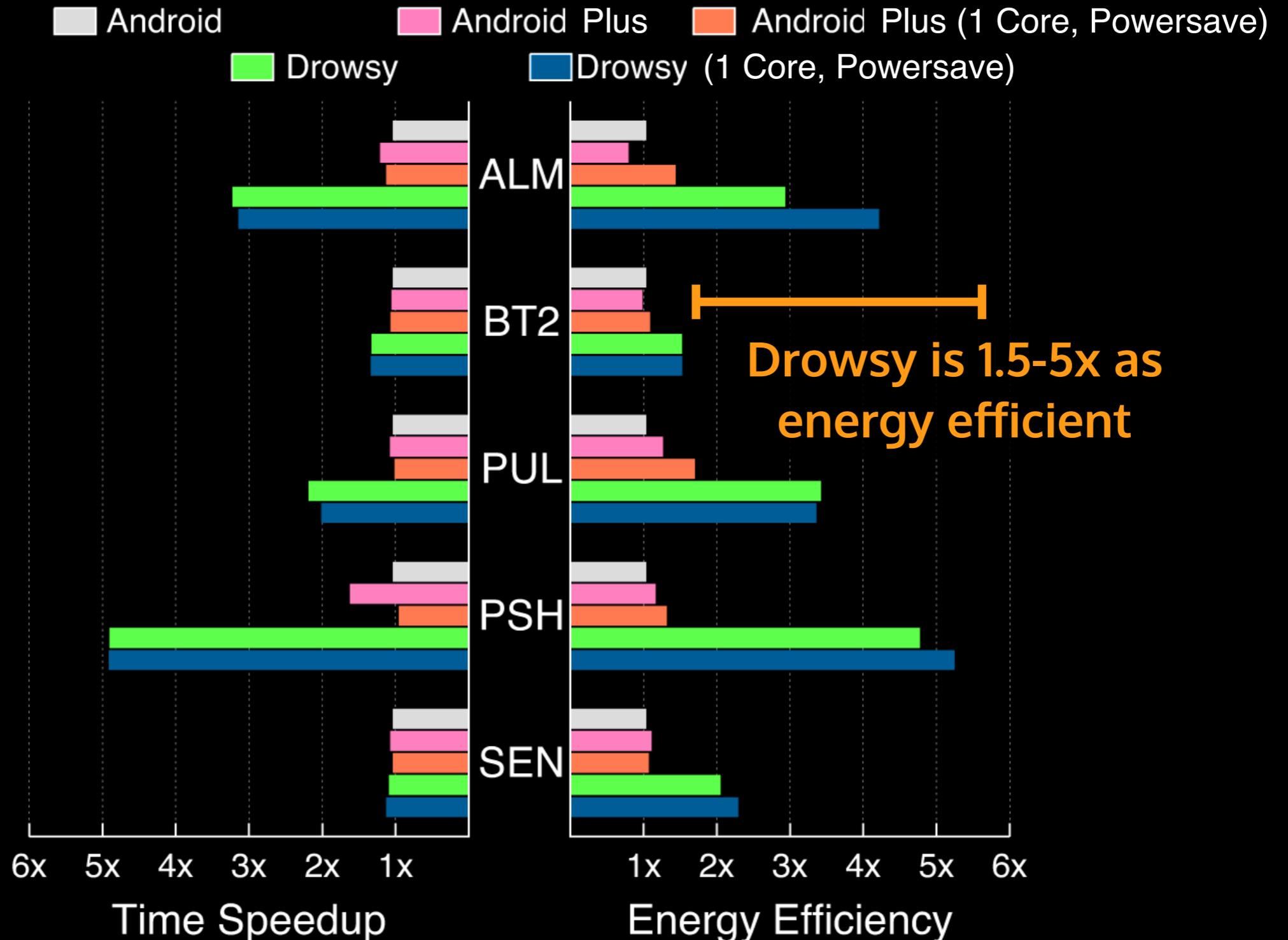
Evaluation: Measurement



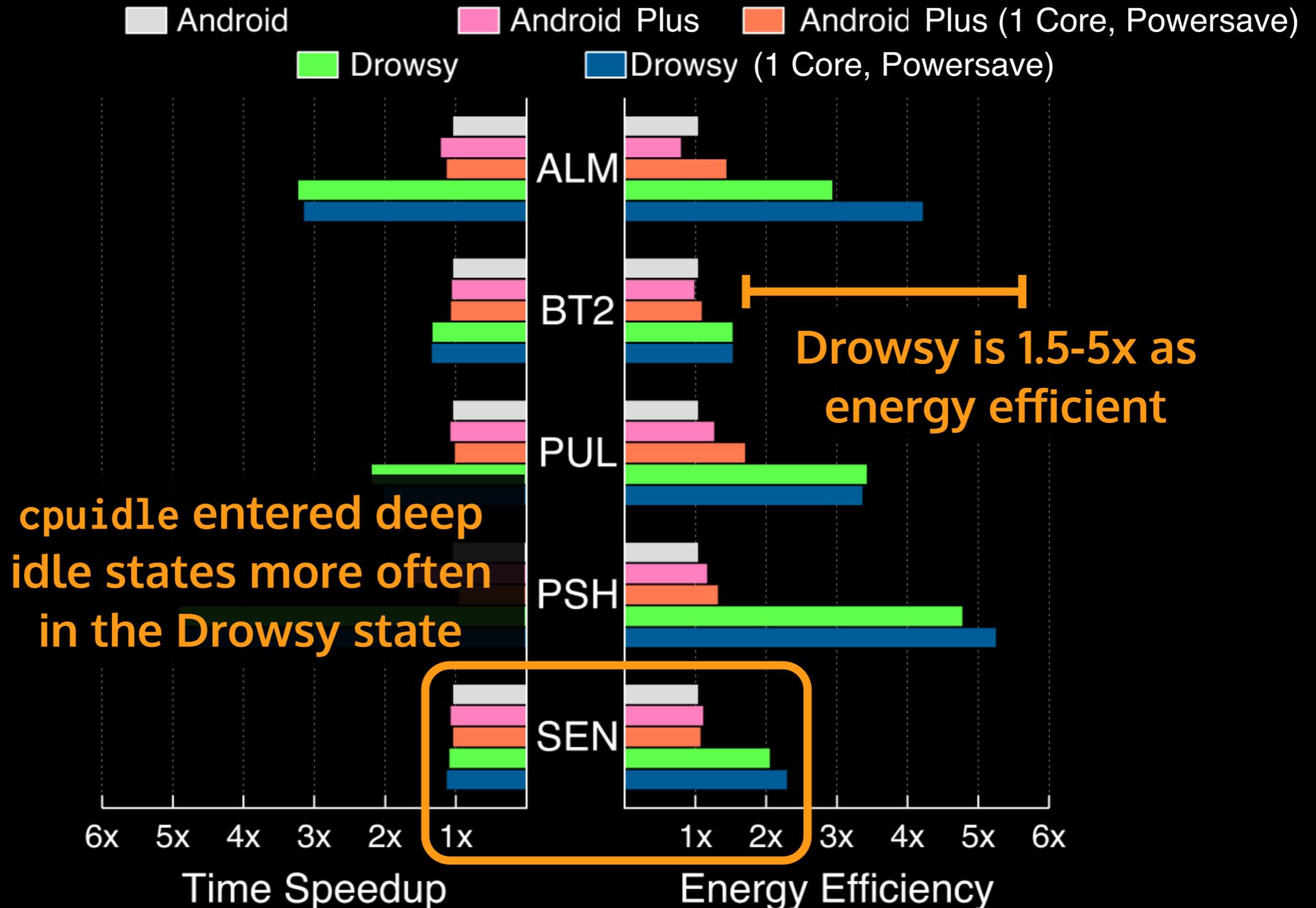
Improvement: Wakeup Events



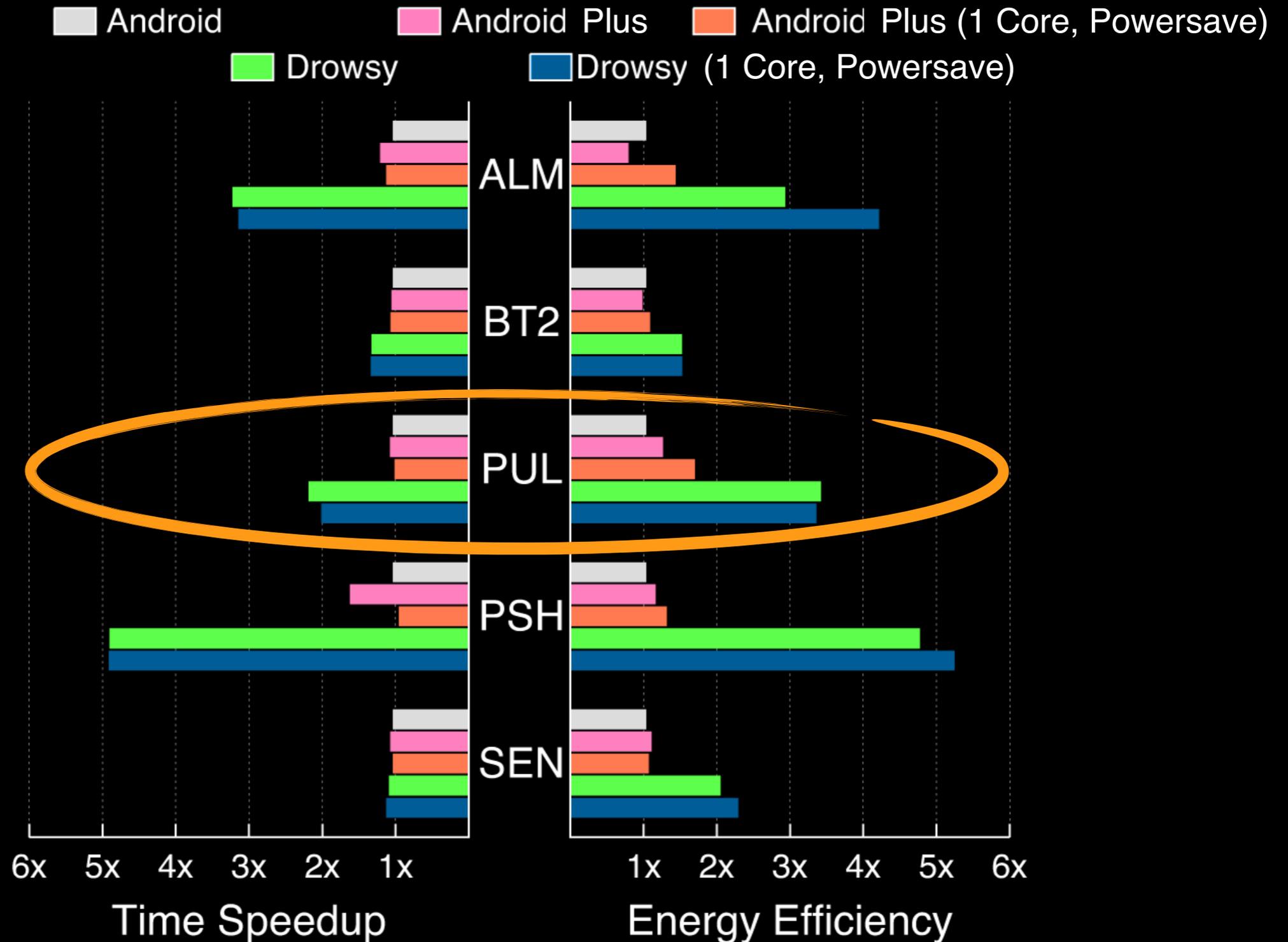
Improvement: Wakeup Events



Improvement: Wakeup Events

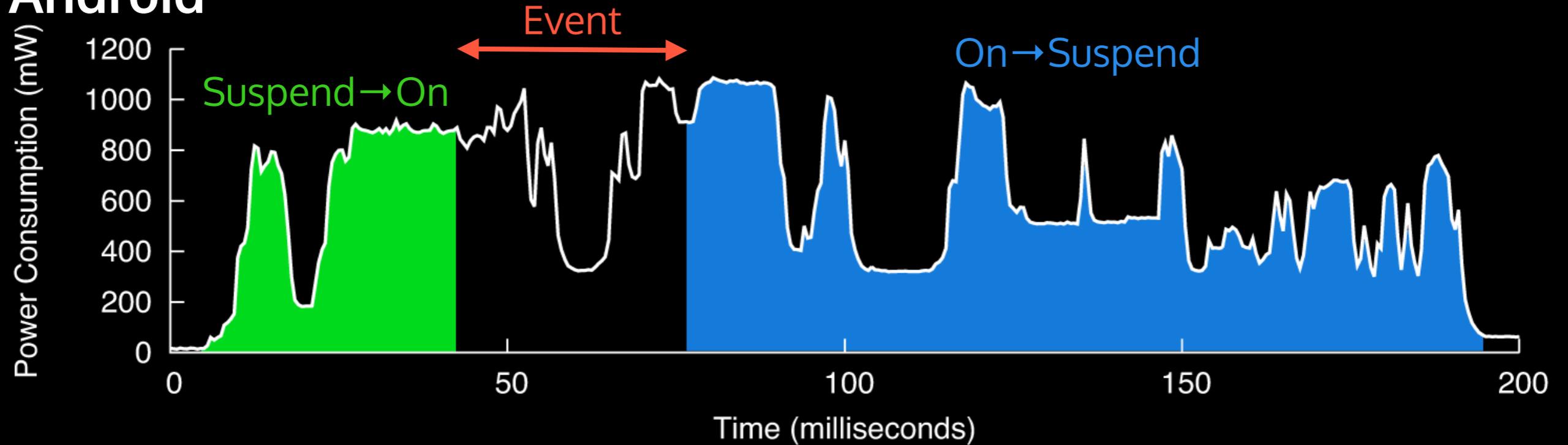


Improvement: Wakeup Events



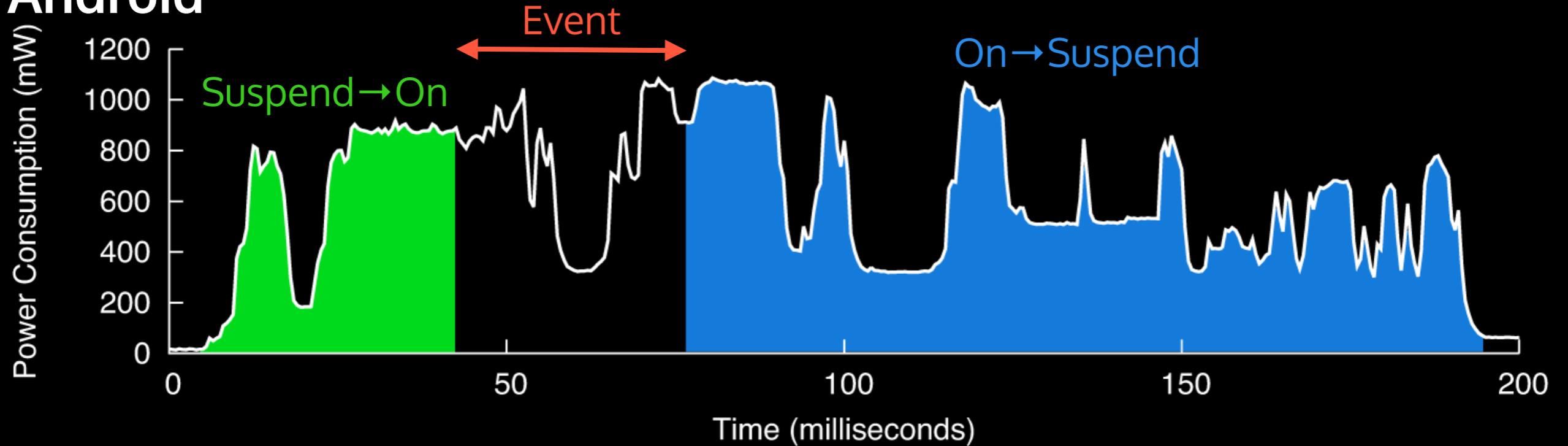
Wakeup Cycle: Pull Data

Android

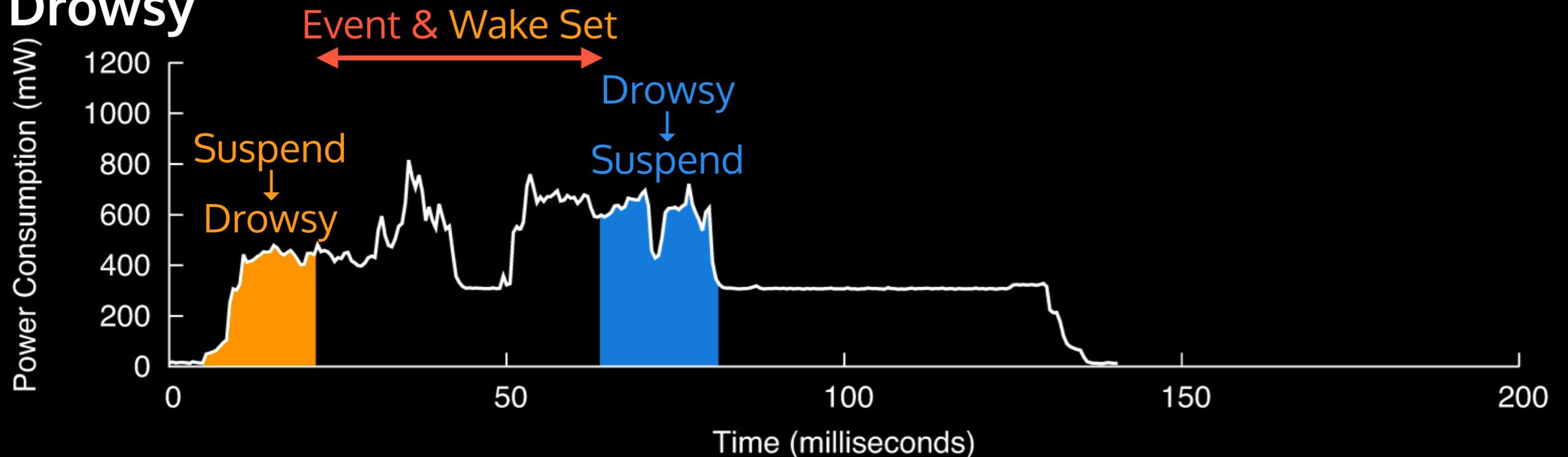


Wakeup Cycle: Pull Data

Android

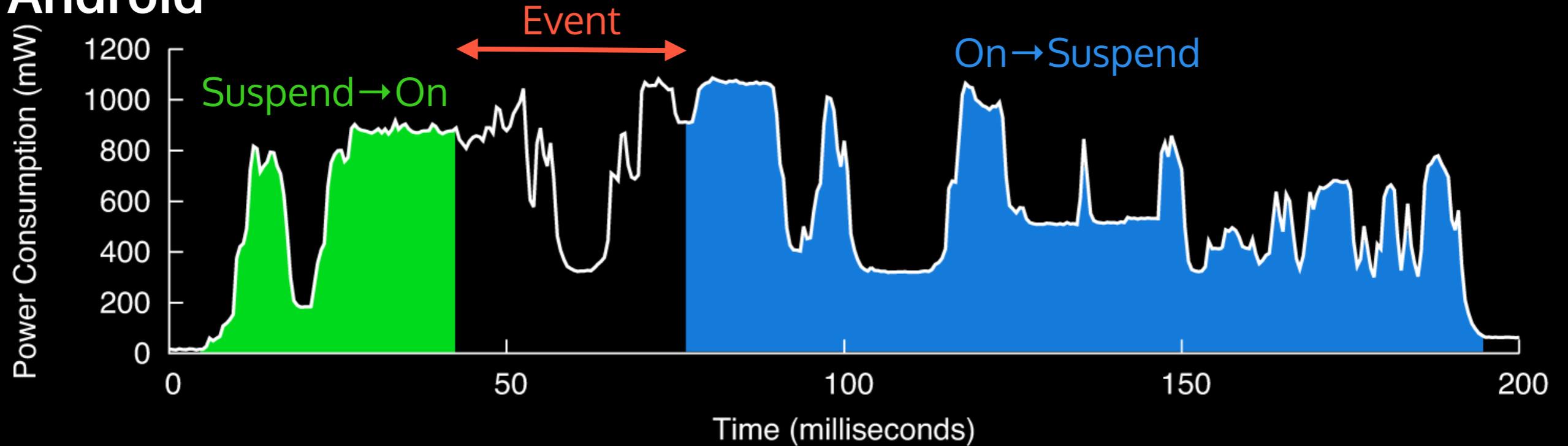


Drowsy

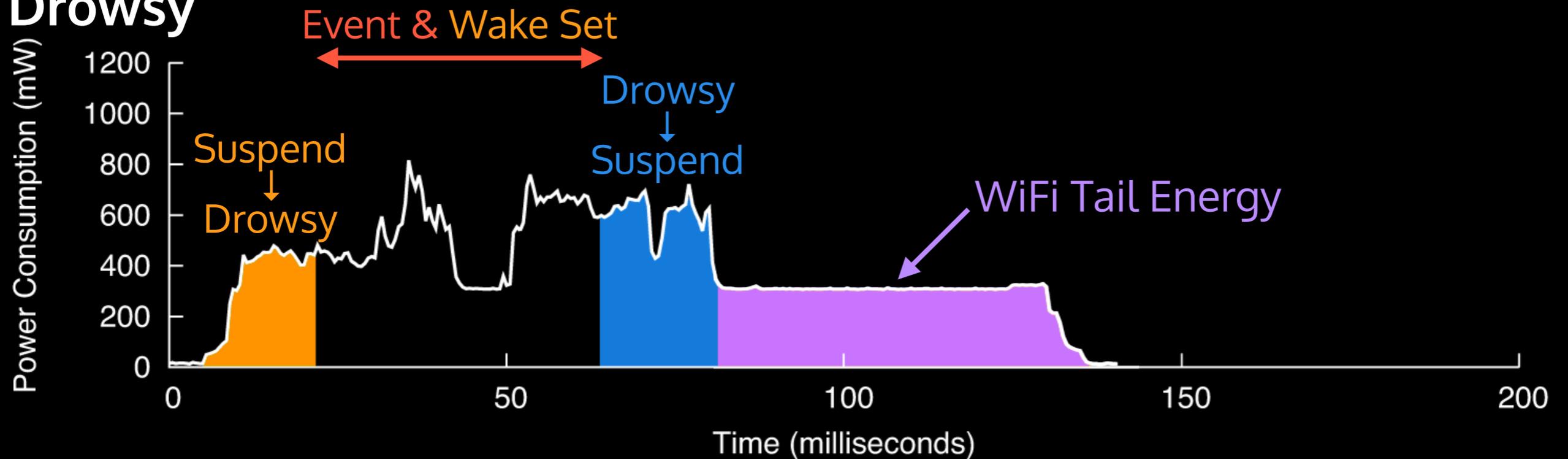


Wakeup Cycle: Pull Data

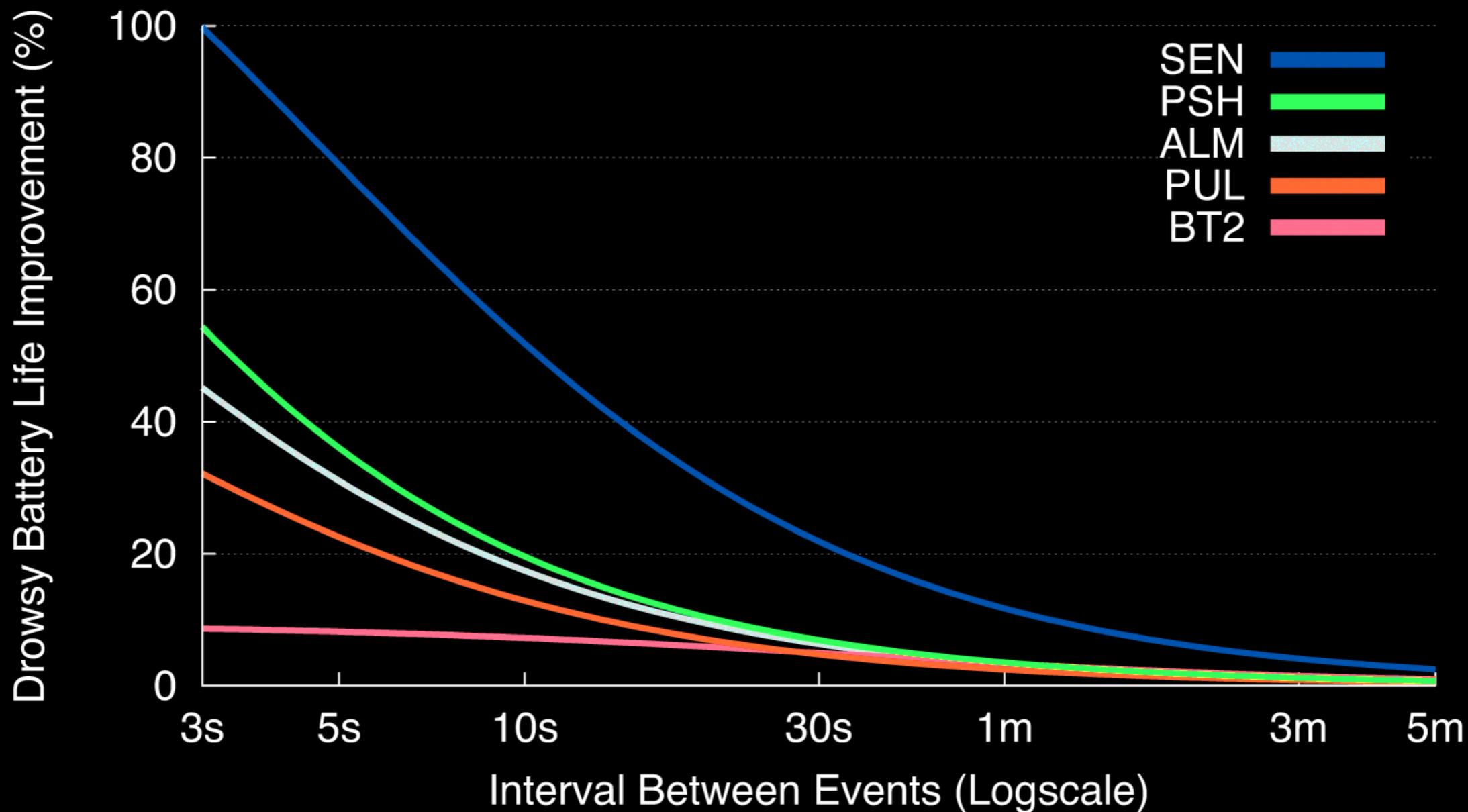
Android



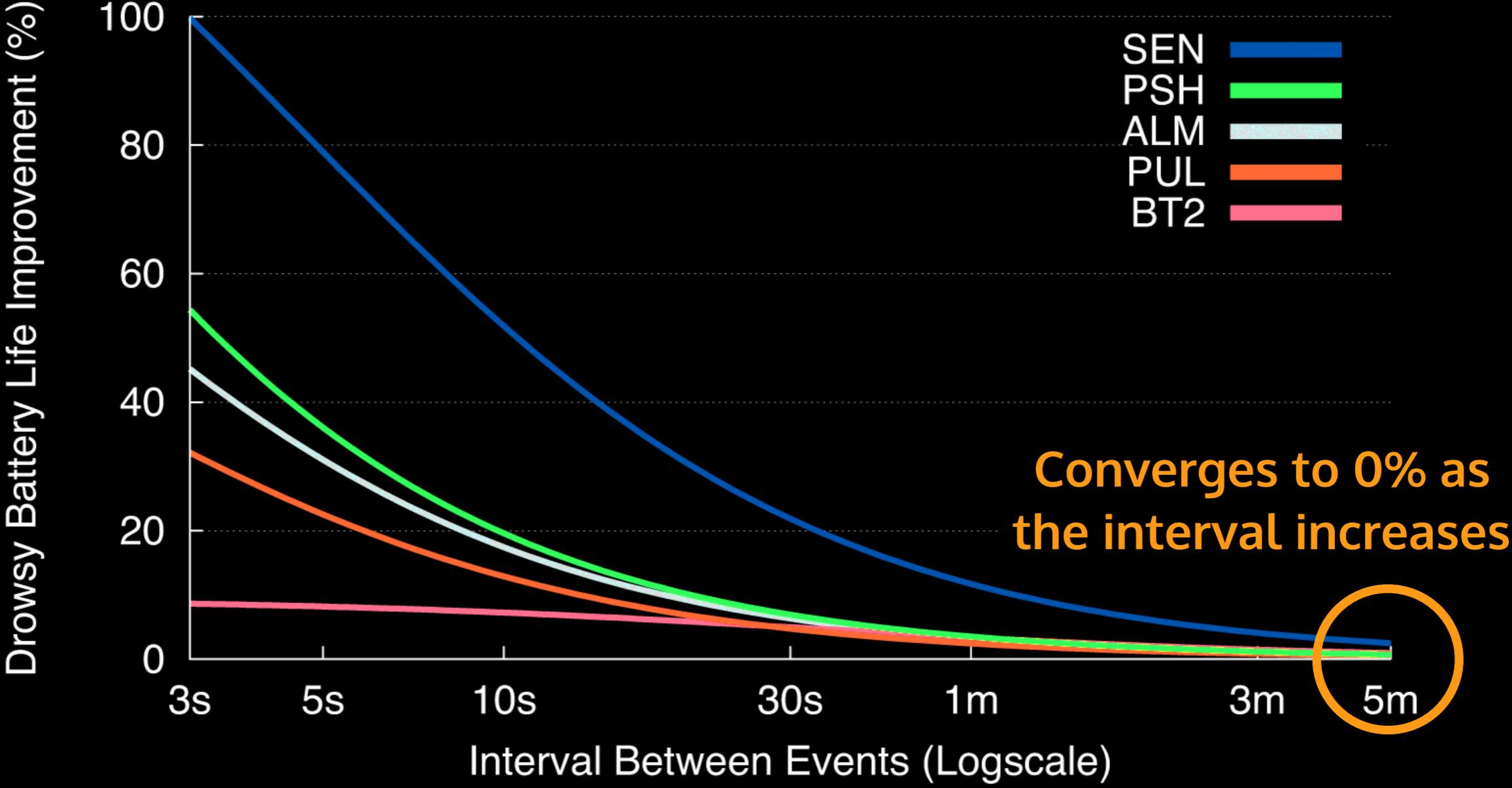
Drowsy



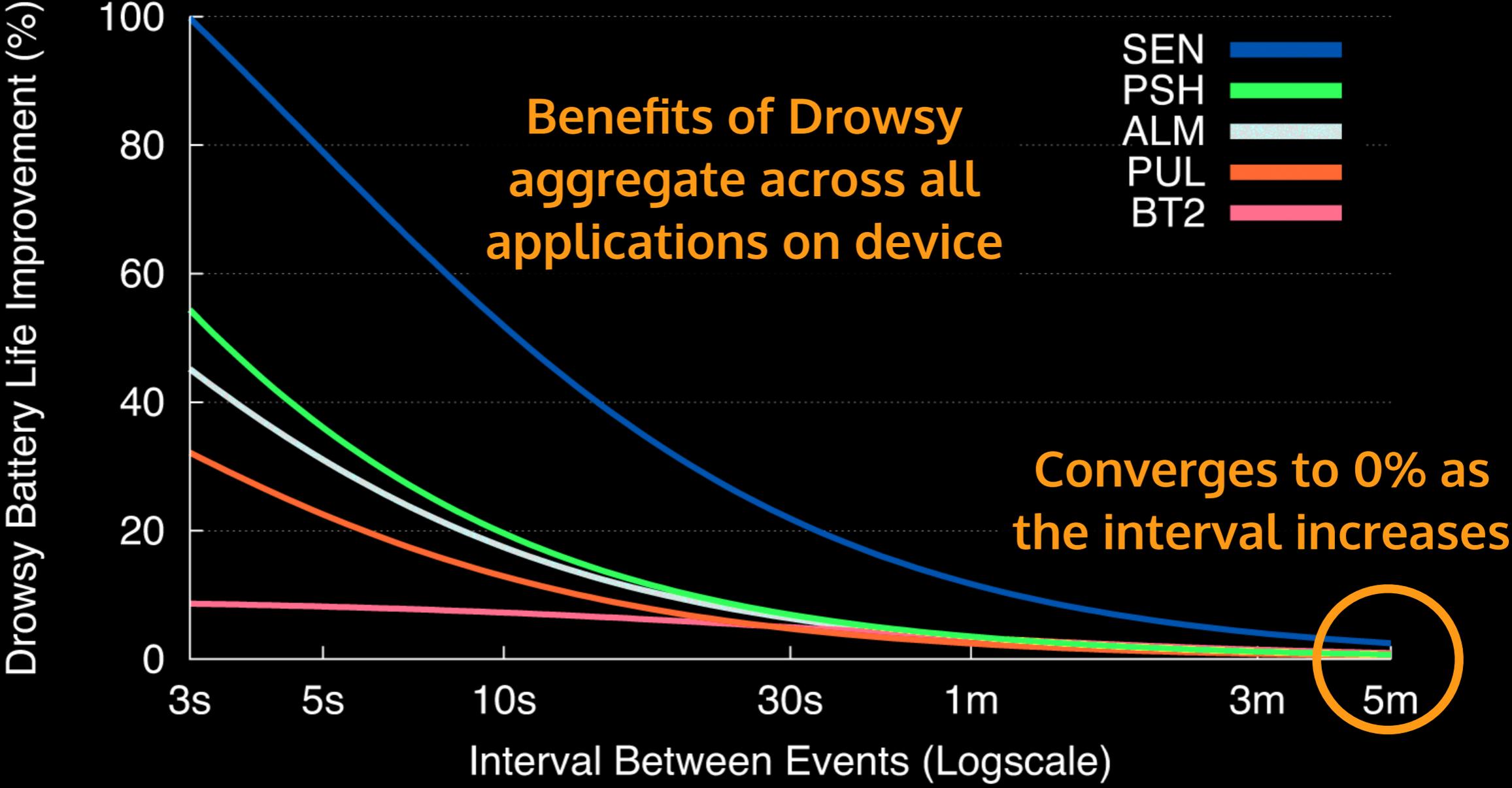
Improvement: Battery Life



Improvement: Battery Life



Improvement: Battery Life



Summary

Existing power management not optimized for short-lived events

Drowsy wakes up the minimal set of tasks and devices
is 1.5 - 5x as energy efficient for short-lived events

Source code is available at:

www.cs.umd.edu/projects/drowsy/

