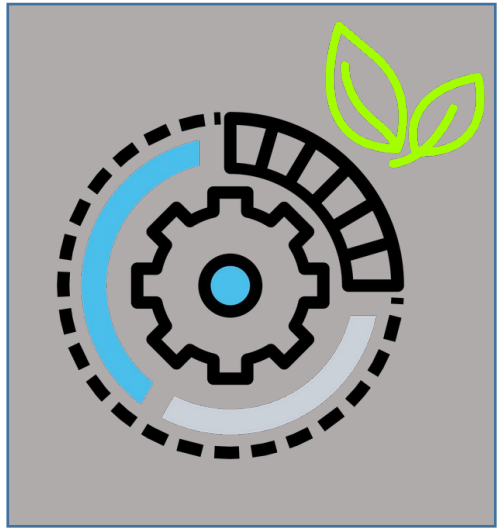
A Case for Lease-Based, Utilitarian Resource Management on Mobile Devices

Yigong Hu, Suyi Liu, Peng Huang
Johns Hopkins University

ASPLOS 2019

Mobile Computing Requires Energy-Aware Apps

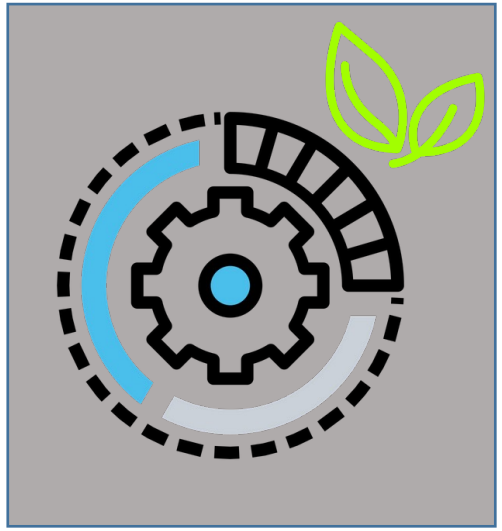


Design mechanisms to optimize system energy usage

Apps need to be energy-aware too!

Energy-aware adaptation for mobile applications [SOSP'99]
ECOSystem: Managing Energy as a First Class Operating System Resource [ASPLOS '02]
Energy Management in Mobile Devices with the Cinder Operating System [EuroSys '11]
Drowsy Power Management [SOSP '15]
CALOREE: Learning Control for Predictable Latency and Low Energy [ASPLOS '18]

Mobile Computing Requires Energy-Aware Apps

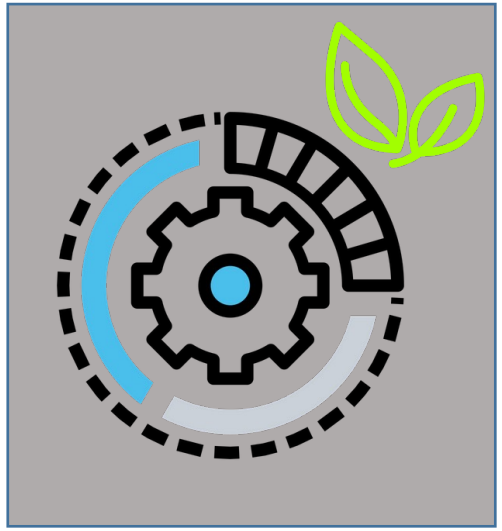


Design mechanisms to optimize system energy usage

Apps need to be energy-aware too!

Energy-aware adaptation for mobile applications [SOSP'99]
ECOSystem: Managing Energy as a First Class Operating System Resource [ASPLOS '02]
Energy Management in Mobile Devices with the Cinder Operating System [EuroSys '11]
Drowsy Power Management [SOSP '15]
CALOREE: Learning Control for Predictable Latency and Low Energy [ASPLOS '18]

Mobile Computing Requires Energy-Aware Apps



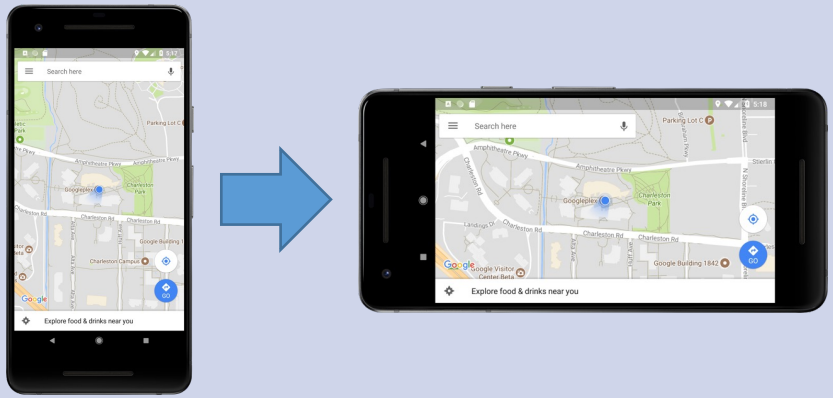
Design mechanisms to optimize system energy usage

Apps need to be energy-aware too!

Energy-aware adaptation for mobile applications [SOSP'99]
ECOSystem: Managing Energy as a First Class Operating System Resource [ASPLOS '02]
Energy Management in Mobile Devices with the Cinder Operating System [EuroSys '11]
Drowsy Power Management [SOSP '15]
CALOREE: Learning Control for Predictable Latency and Low Energy [ASPLOS '18]

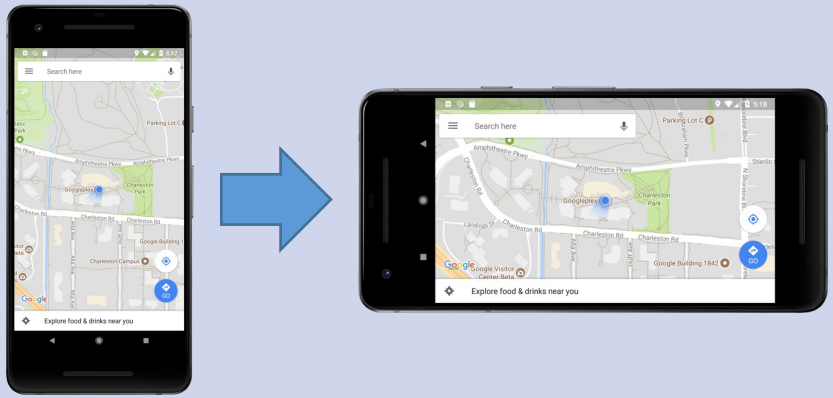
Energy-Aware Programming Is Challenging

- Reasoning about code and testing app is hard

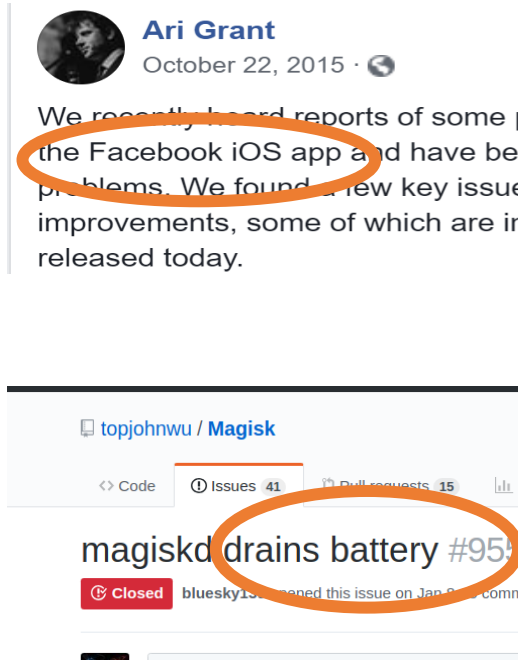
User Interaction	Environment Conditions	Runtime Changes
 Click Bottom	 Network Disconnect	 Orientation Change

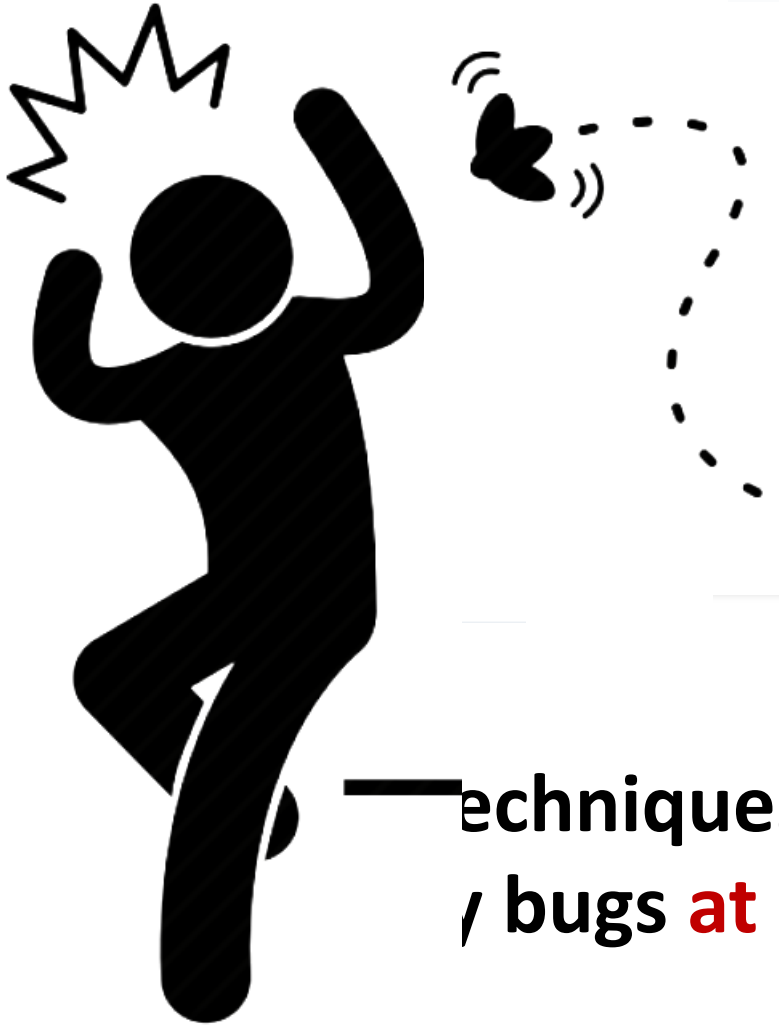
Energy-Aware Programming Is Challenging

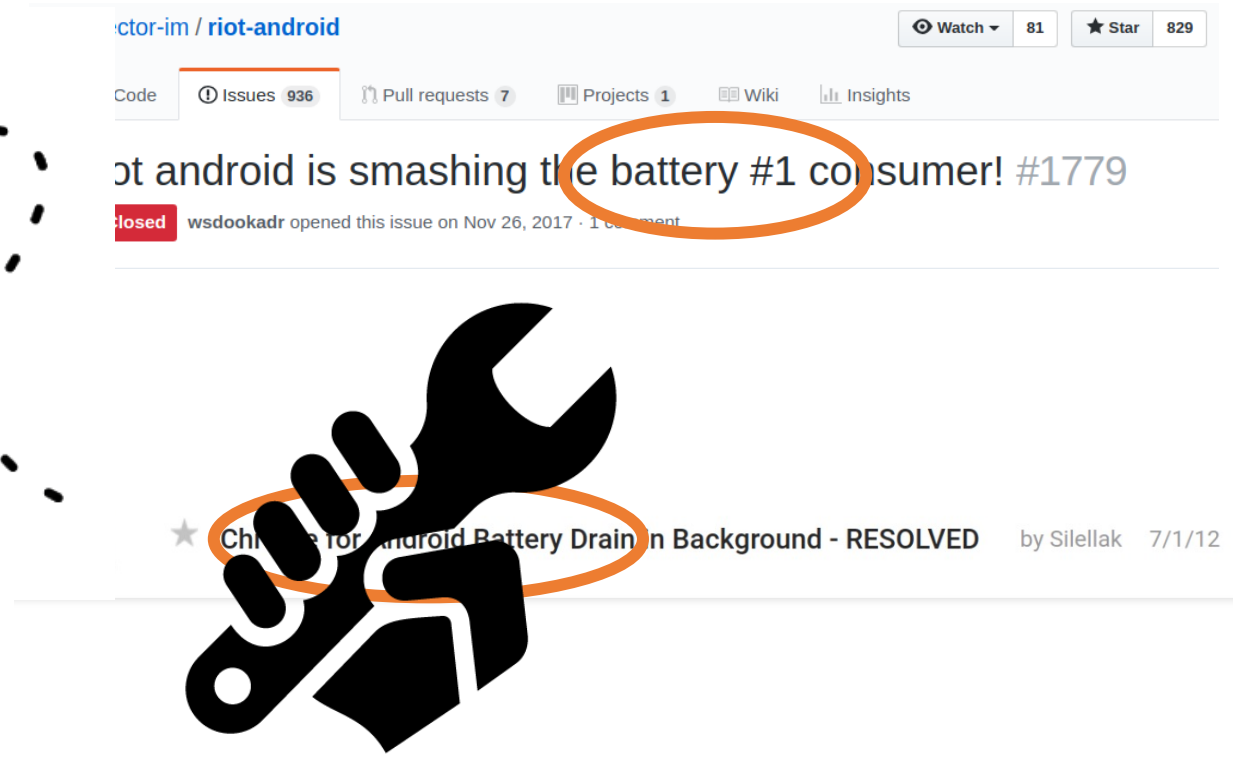
- Reasoning about code and testing app is hard

User Interaction	Environment Conditions	Runtime Changes
 Click Bottom	 Network Disconnect	 Orientation Change

Apps with Energy Bugs Are Common

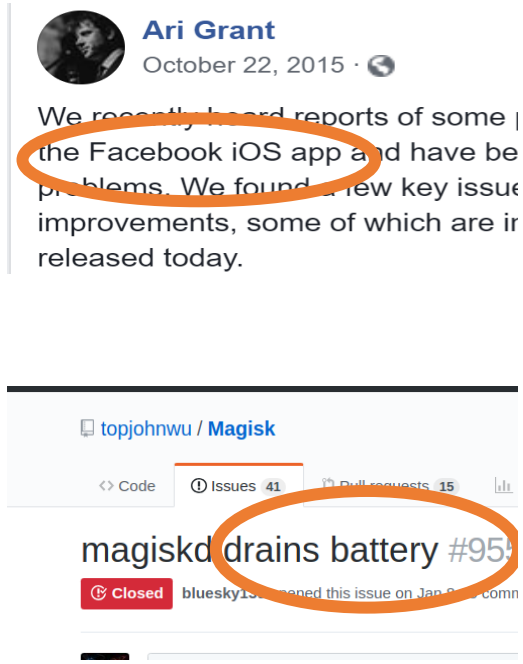


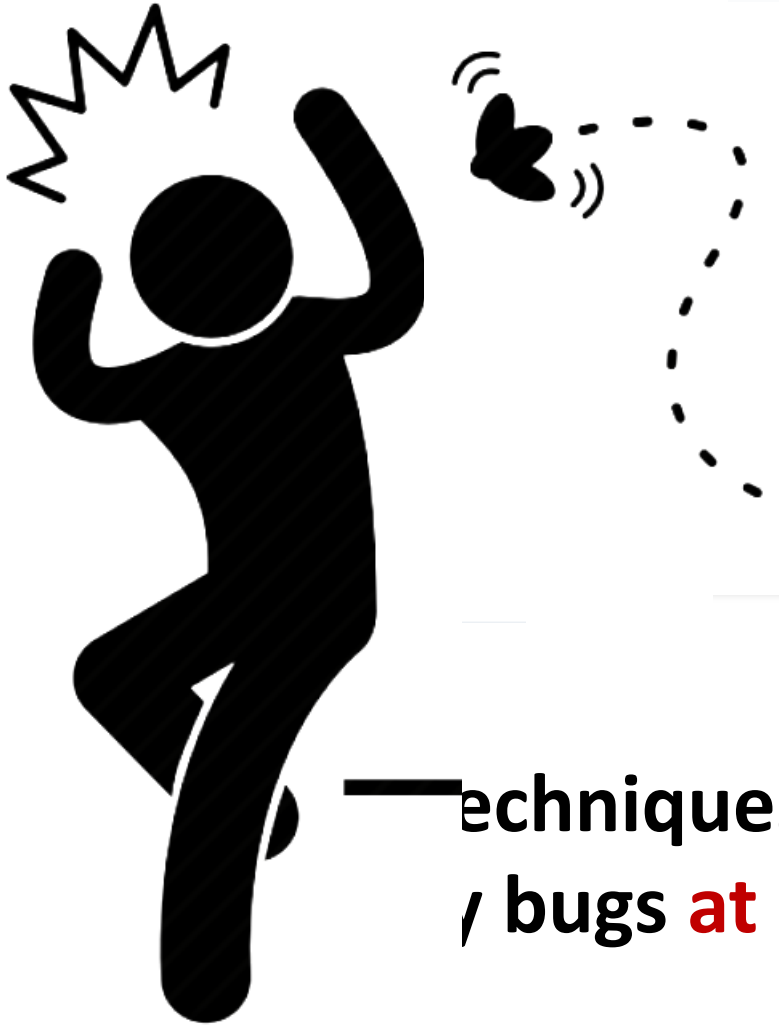


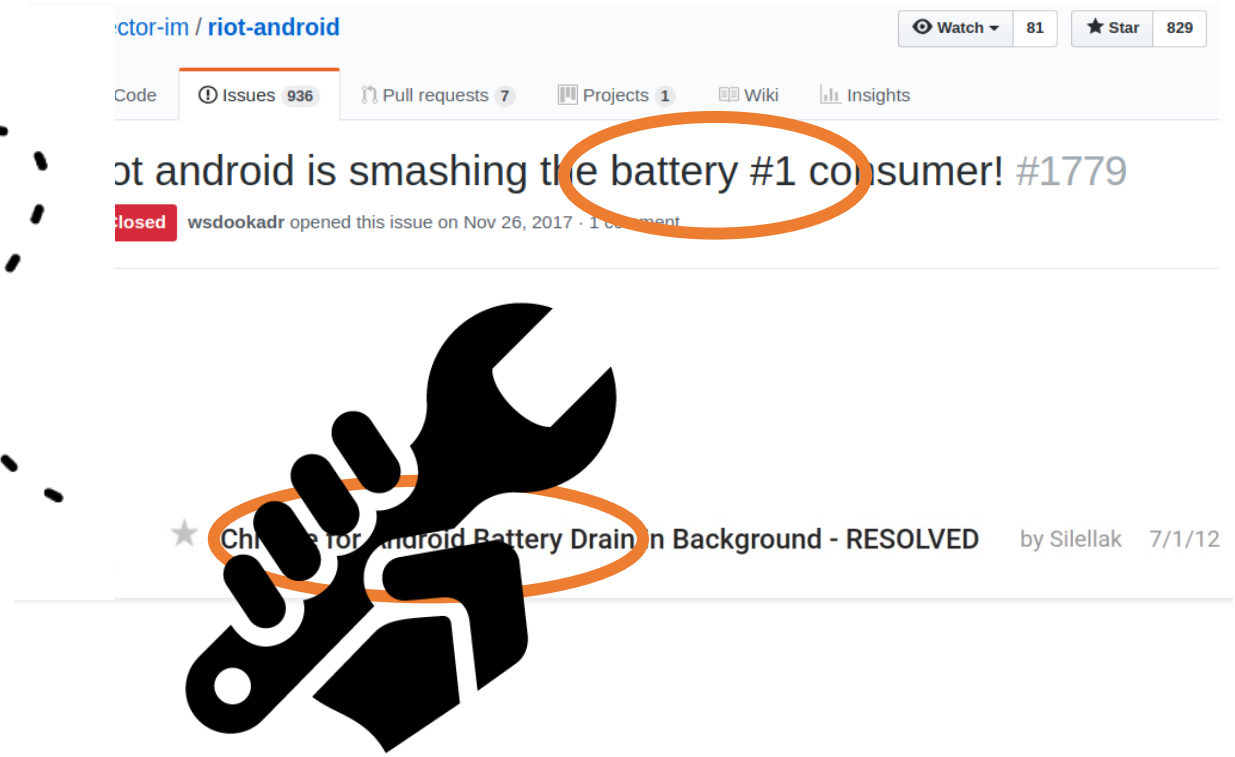


echniques to mitigate
bugs **at runtime**

Apps with Energy Bugs Are Common

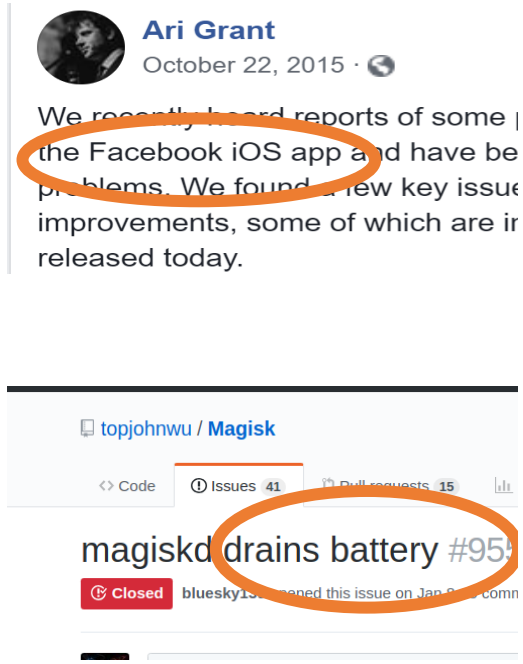


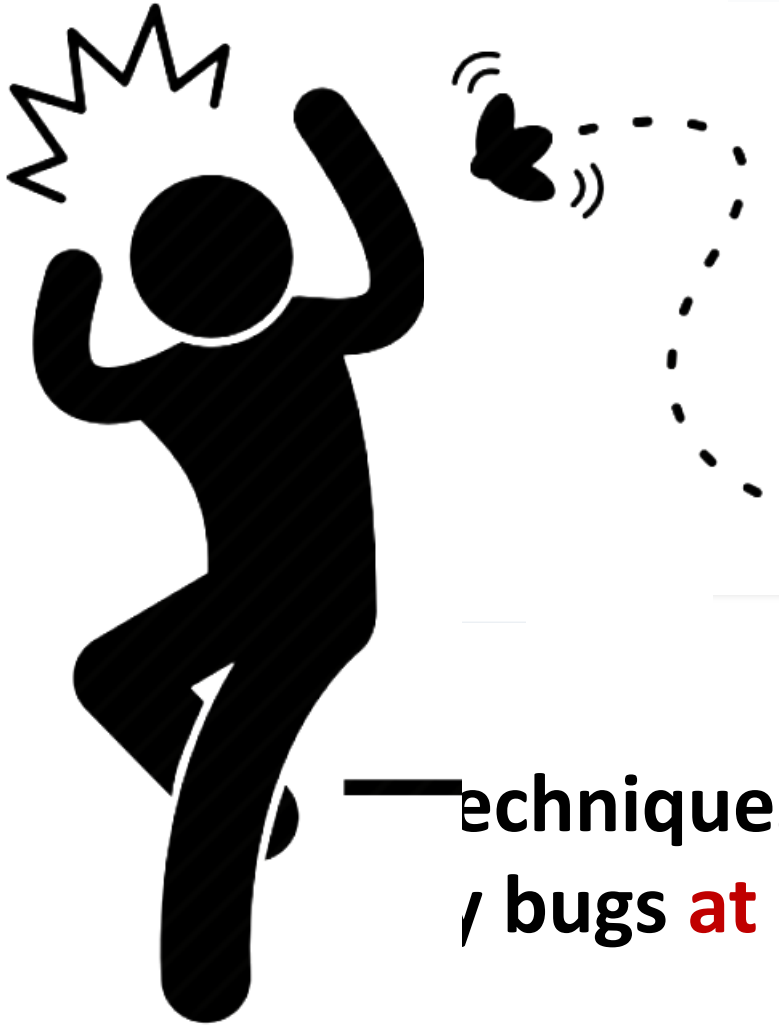


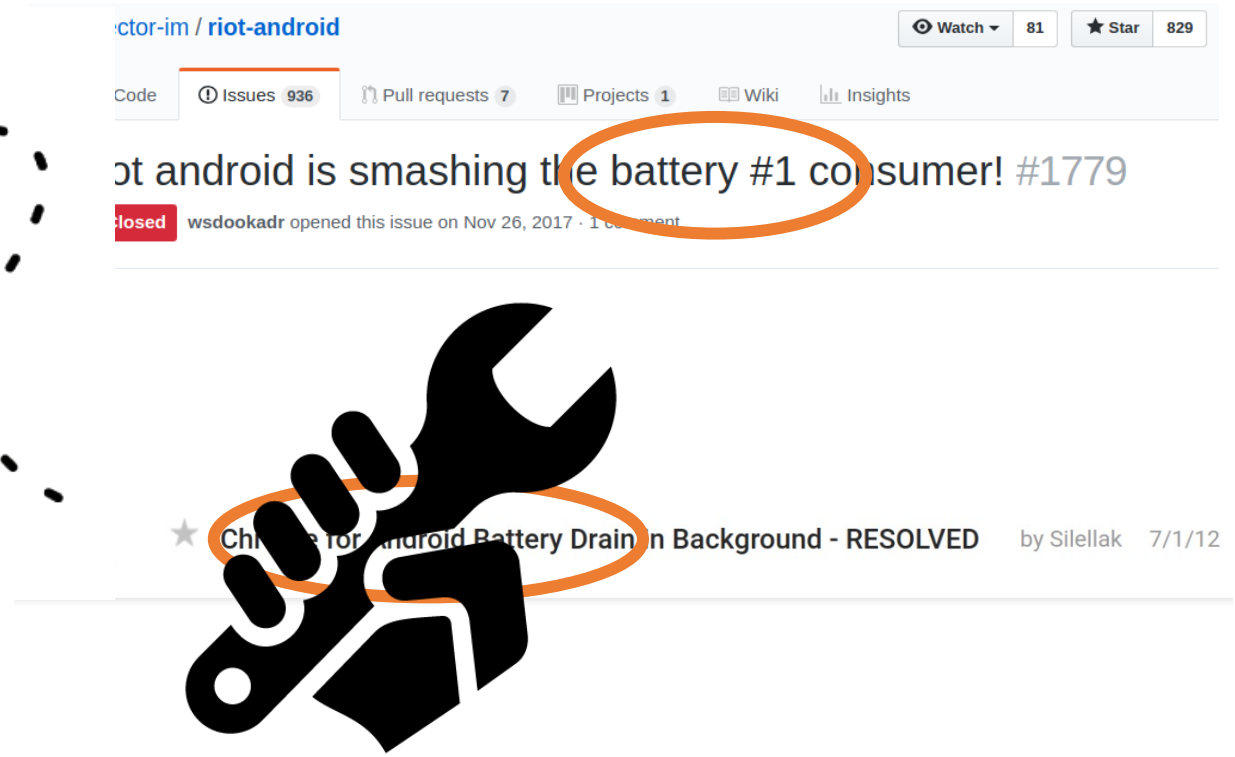


echniques to mitigate
bugs **at runtime**

Apps with Energy Bugs Are Common

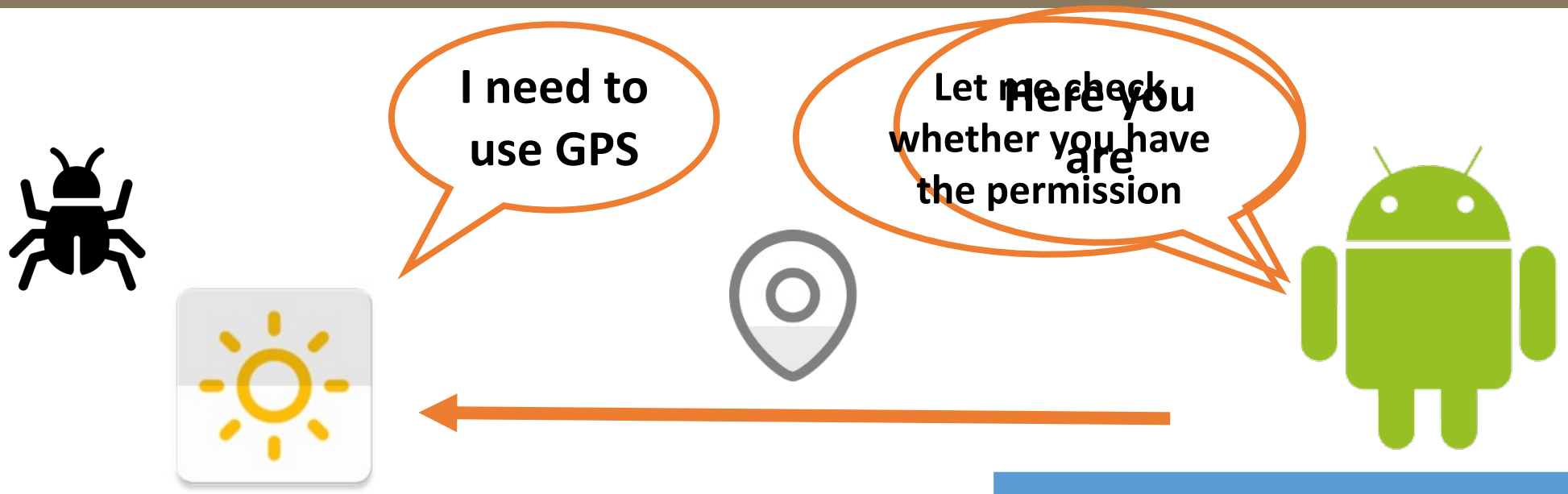






echniques to mitigate
bugs **at runtime**

Default Mobile Resource Management Is Libertarian



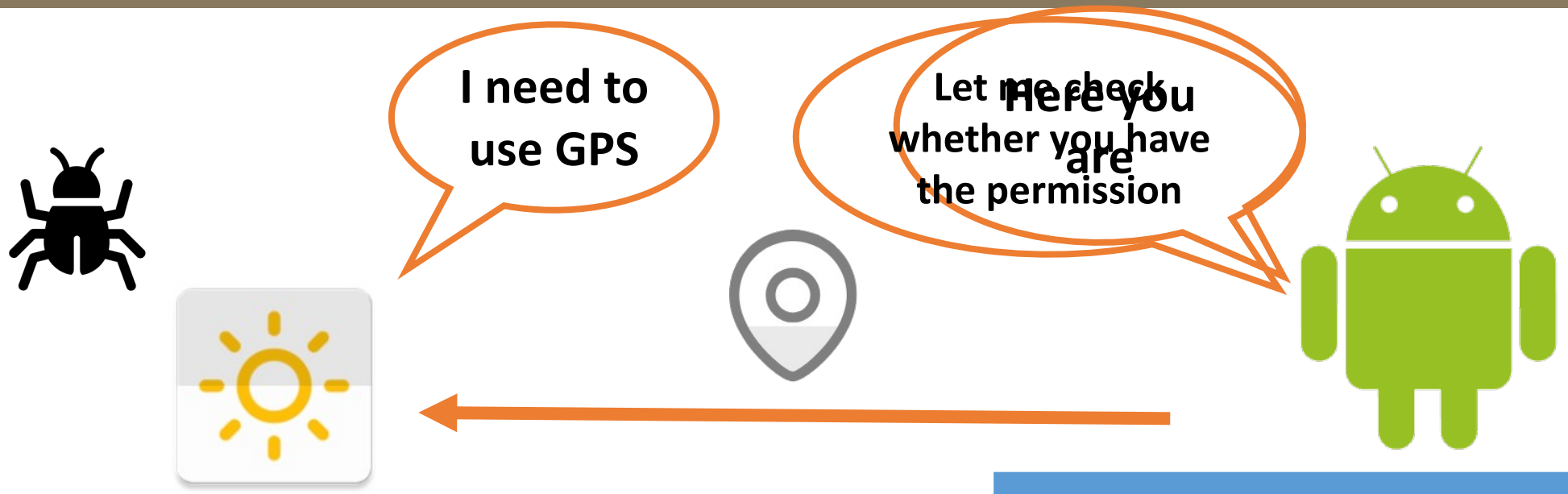
✓ App Usability

✗ Energy Efficiency

Give app freedom

- Allocate resource if permission and sanity checks pass
- Revoke resource only when the app releases it

Default Mobile Resource Management Is Libertarian



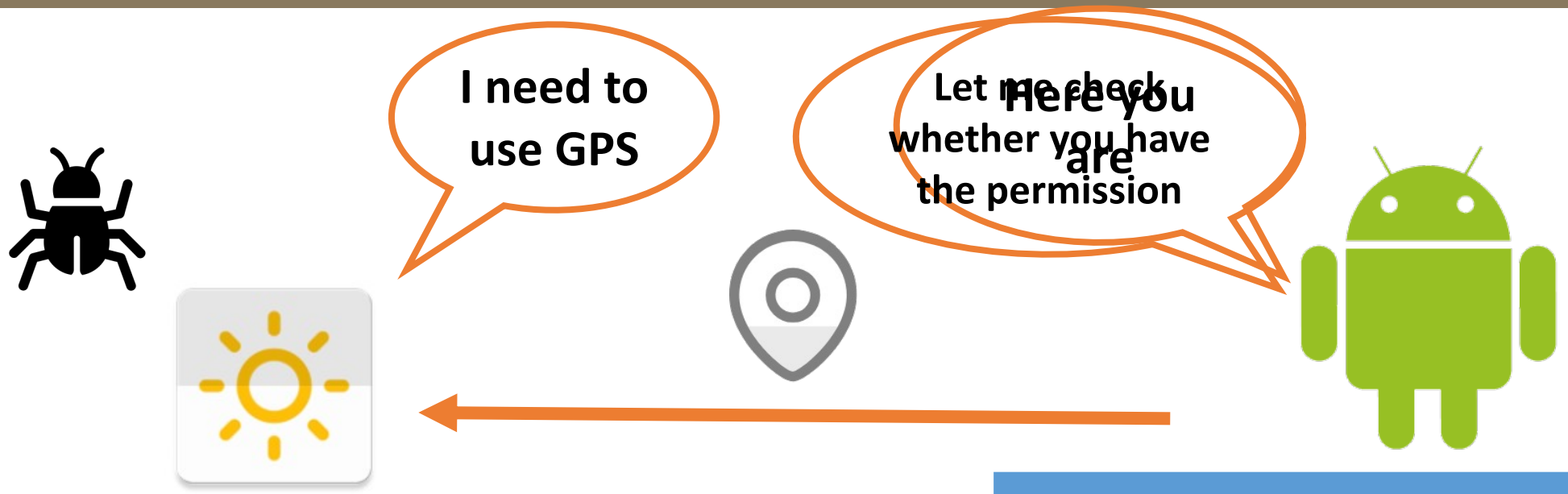
✓ App Usability

✗ Energy Efficiency

Give app freedom

- Allocate resource if permission and sanity checks pass
- Revoke resource only when the app releases it

Default Mobile Resource Management Is Libertarian



✓ App Usability

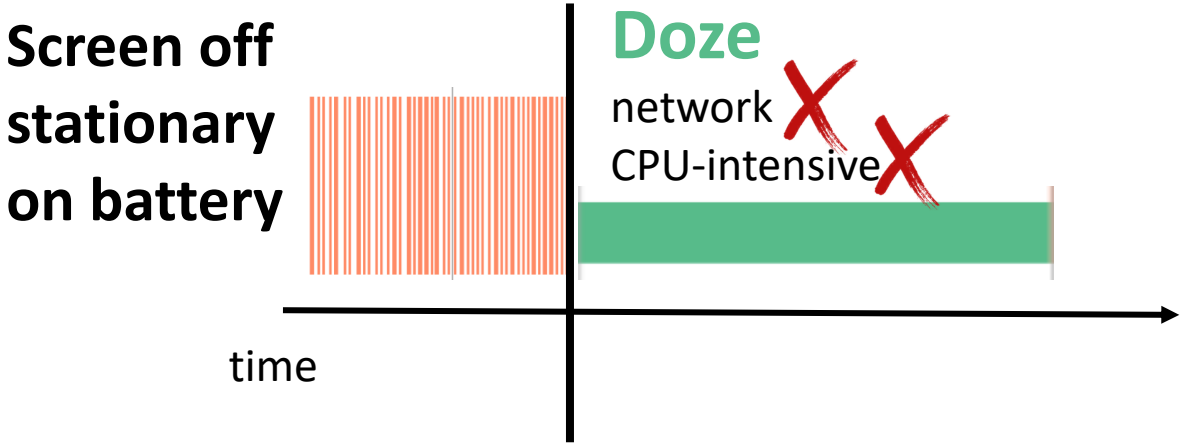
✗ Energy Efficiency

Give app freedom

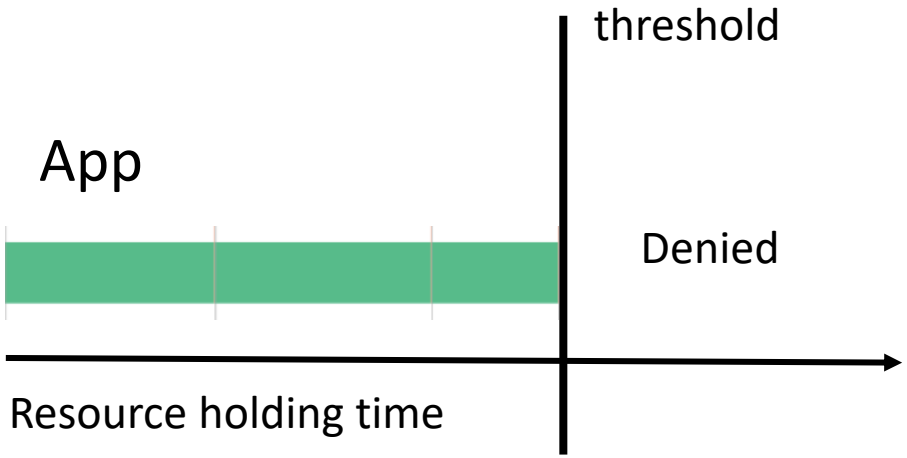
- Allocate resource if permission and sanity checks pass
- Revoke resource only when the app releases it

Recent Authoritarian Approach via Throttling

Android Doze (since 6.0)



DefDroid

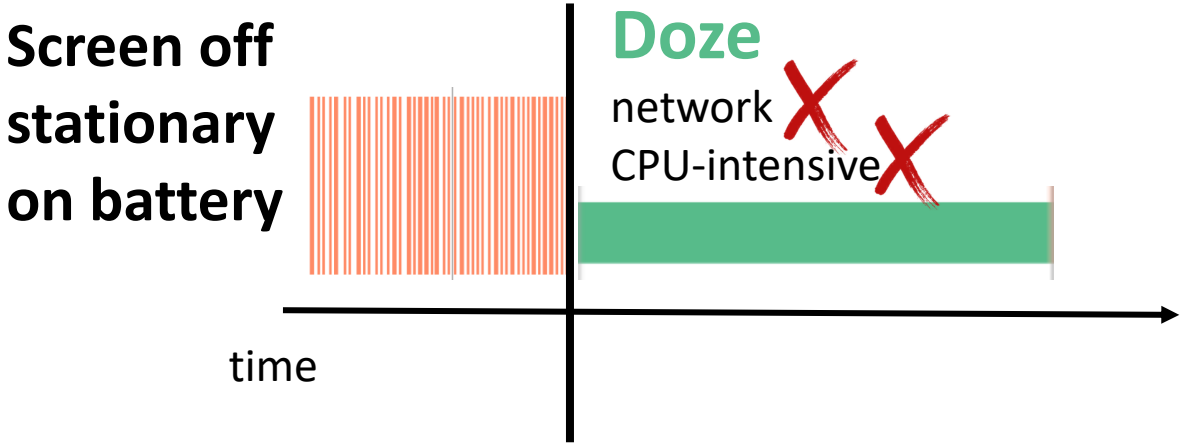


✓ Energy Awareness

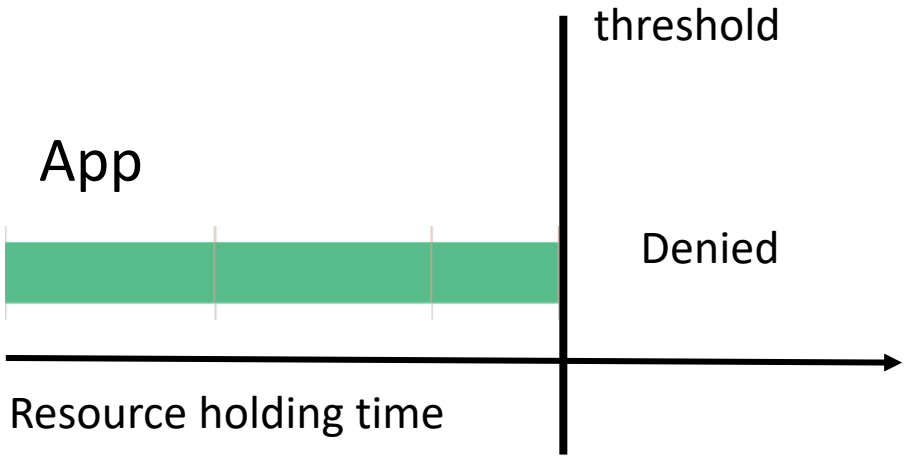
✗ App Usability: legitimate app requests denied

Recent Authoritarian Approach via Throttling

Android Doze (since 6.0)



DefDroid

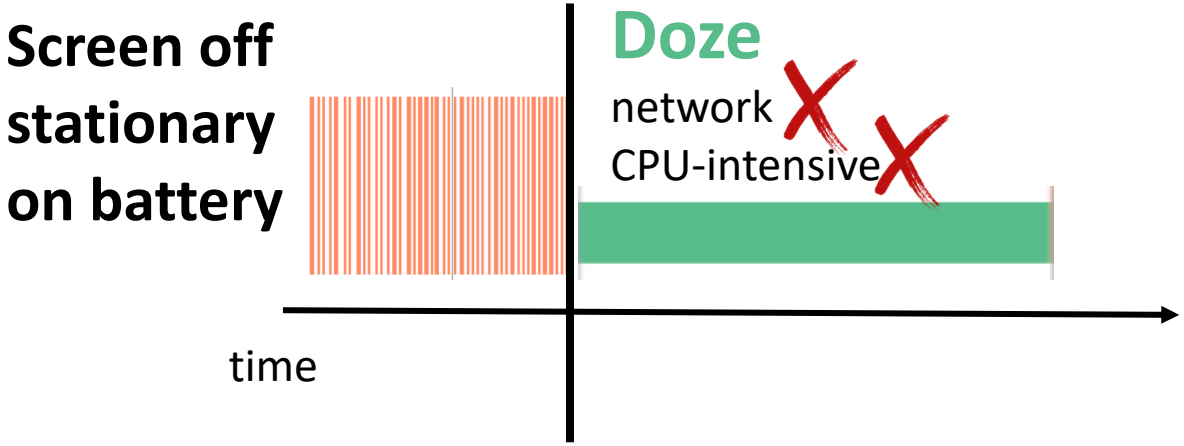


✓ Energy Awareness

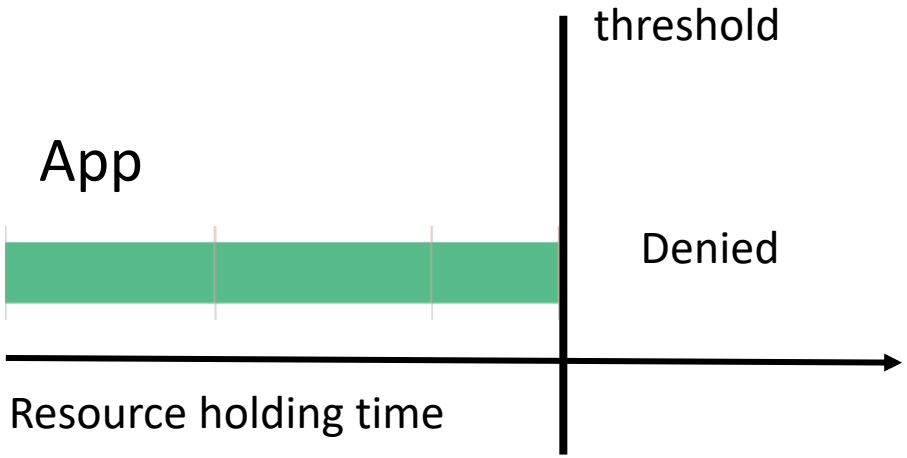
✗ App Usability: legitimate app requests denied

Recent Authoritarian Approach via Throttling

Android Doze (since 6.0)



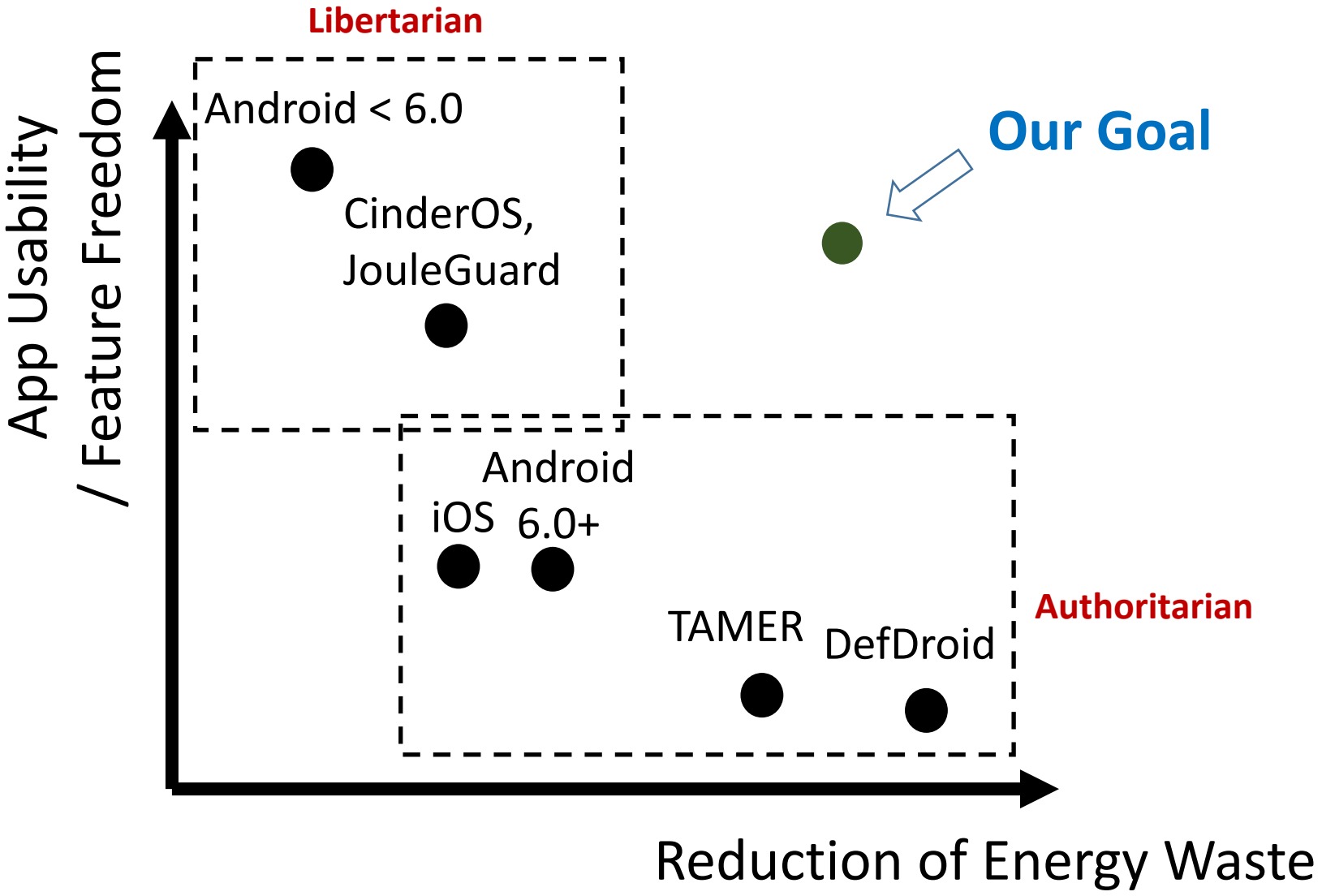
DefDroid



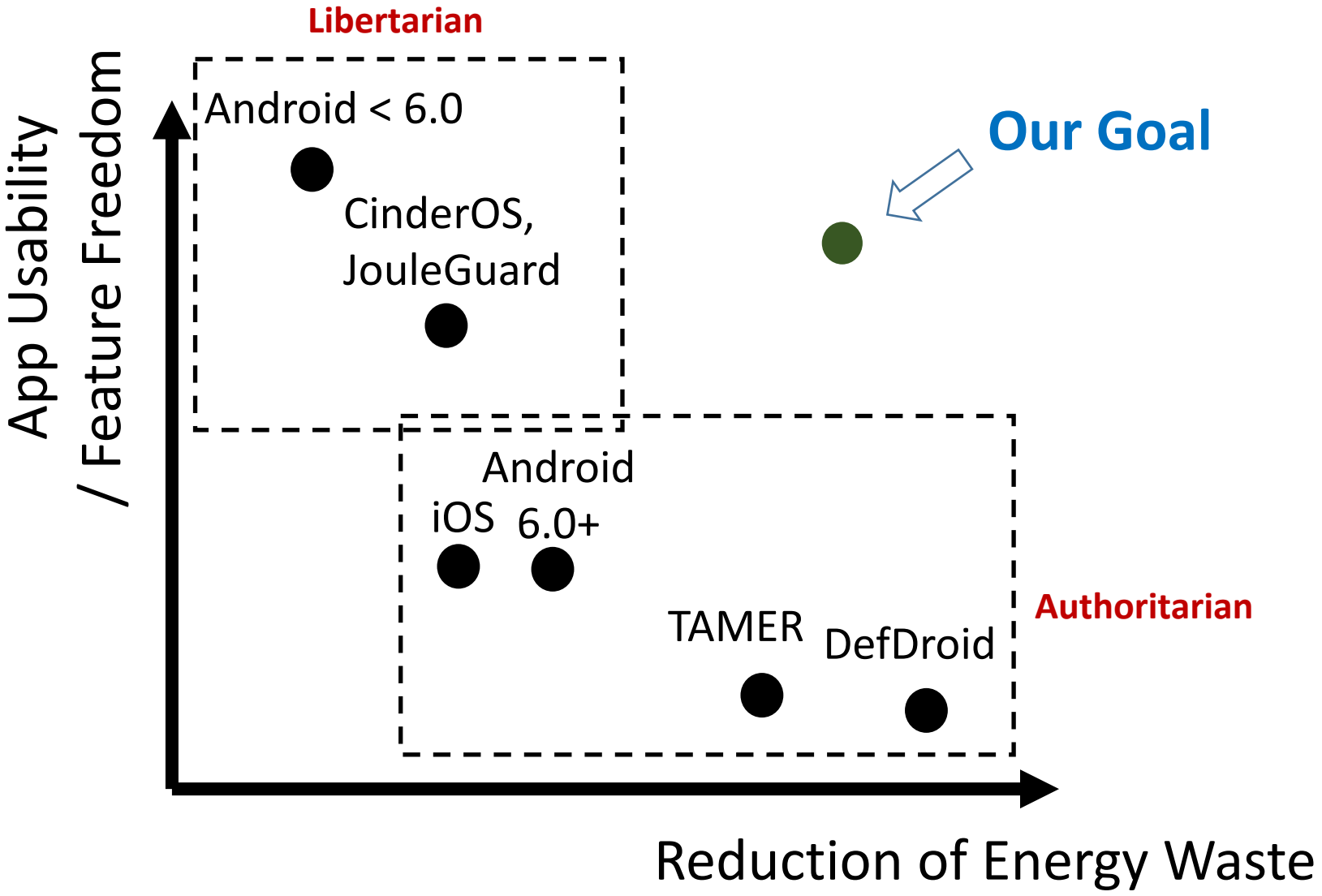
✓ Energy Awareness

✗ App Usability: legitimate app requests denied

Fix the Gap


Libertarian
App Usability
/ Feature Freedom
Android < 6.0
CinderOS,
JouleGuard
Our Goal
Android
iOS
6.0+
TAMER
DefDroid
Authoritarian
Reduction of Energy Waste

Fix the Gap


Libertarian
App Usability
/ Feature Freedom
Android < 6.0
CinderOS,
JouleGuard
Our Goal
Android
iOS
6.0+
TAMER
DefDroid
Authoritarian
Reduction of Energy Waste

Our Solution: LeaseOS

Design of Lease Abstraction for Mobile Systems

+

A Utilitarian Approach for Making Lease Decisions

+

A Lightweight System Runtime With Backward-Compatibility

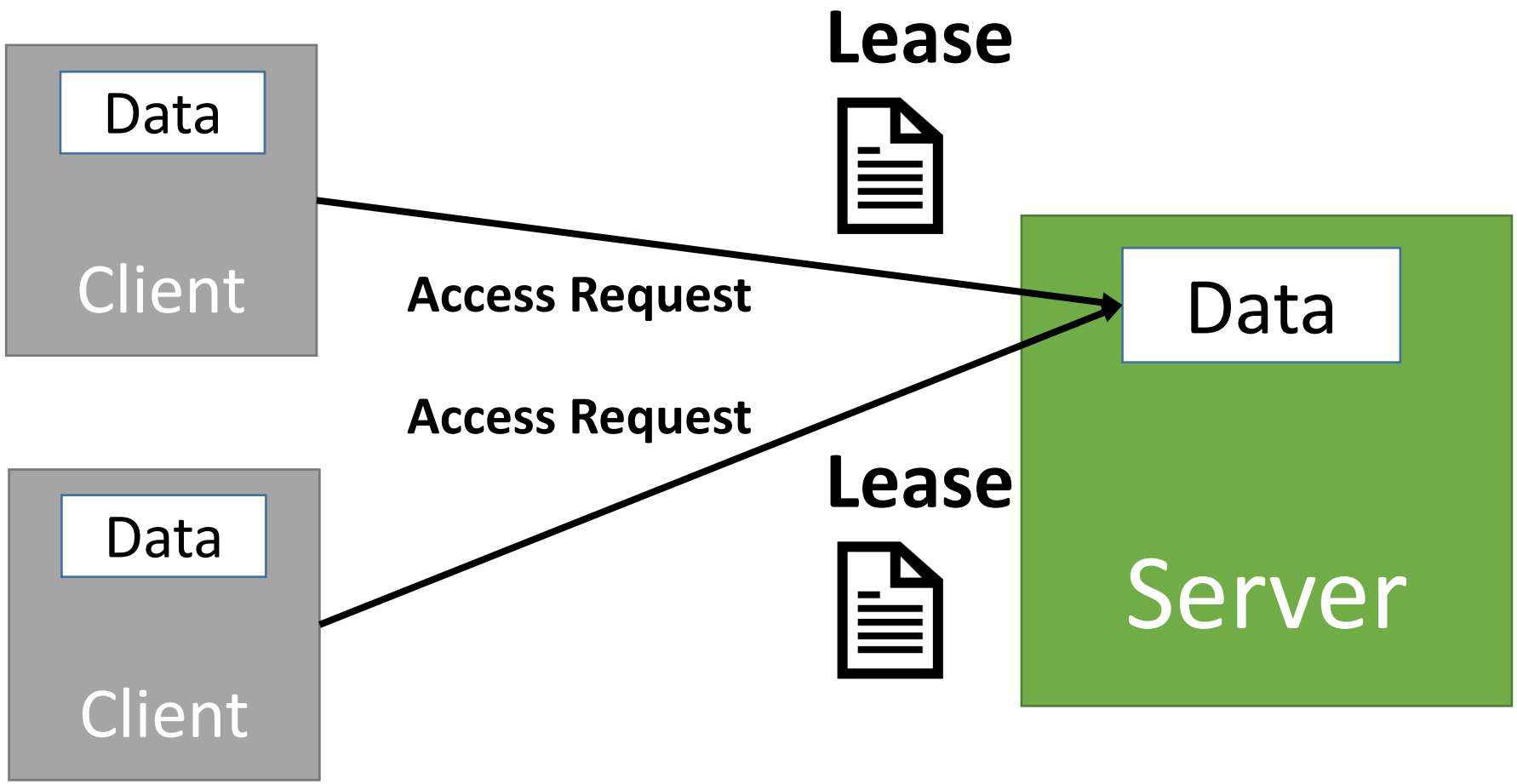
Main Challenges of LeaseOS

- **How to use lease in mobile system to reduce energy waste?**
- **How to reliably distinguish energy misbehavior from legitimate behavior?**

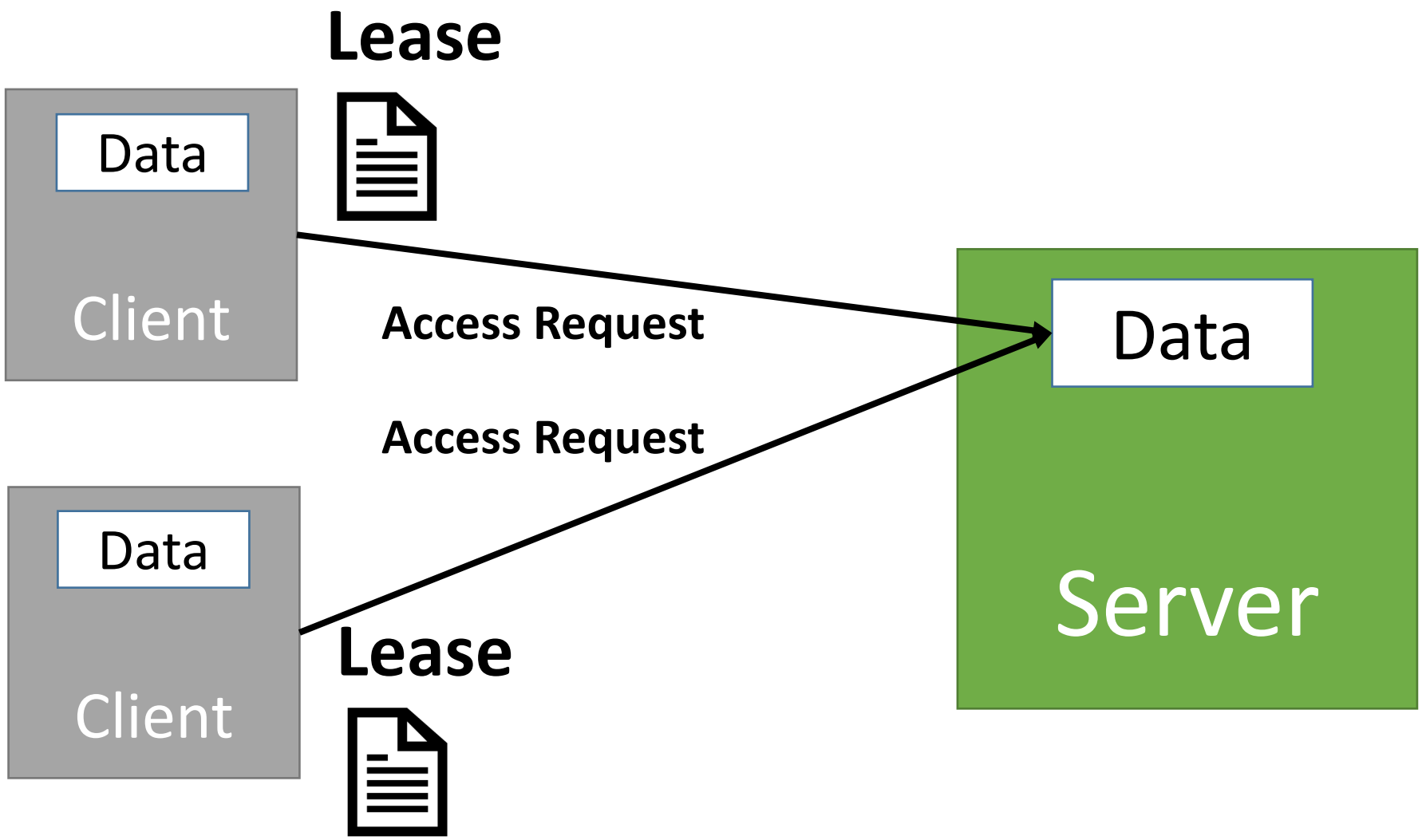
Outline

- Motivation
- **Lease Abstraction**
- **Making Lease Decision**
- **Design of LeaseOS**
- **Evaluation**

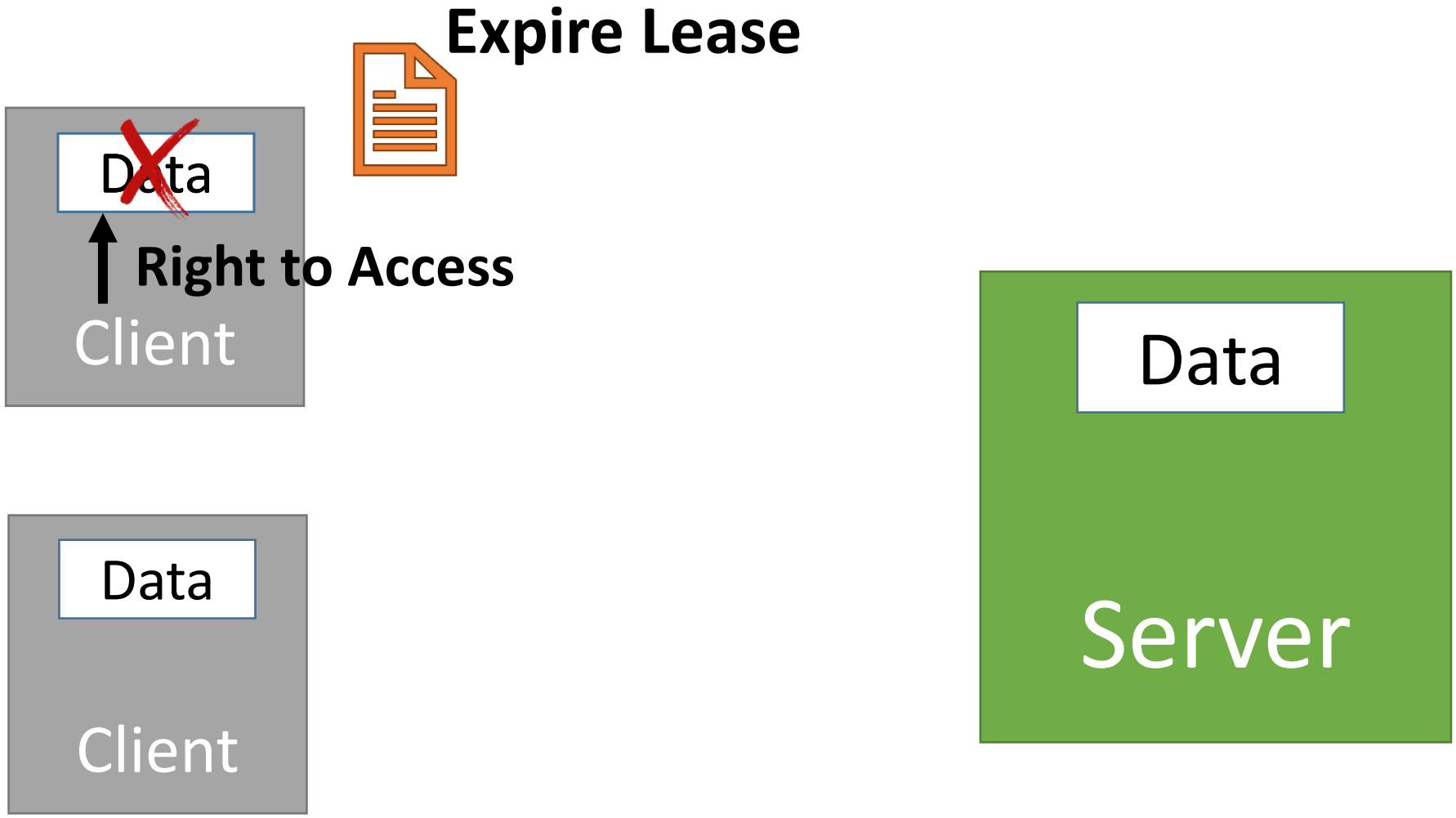
Lease in Distributed System



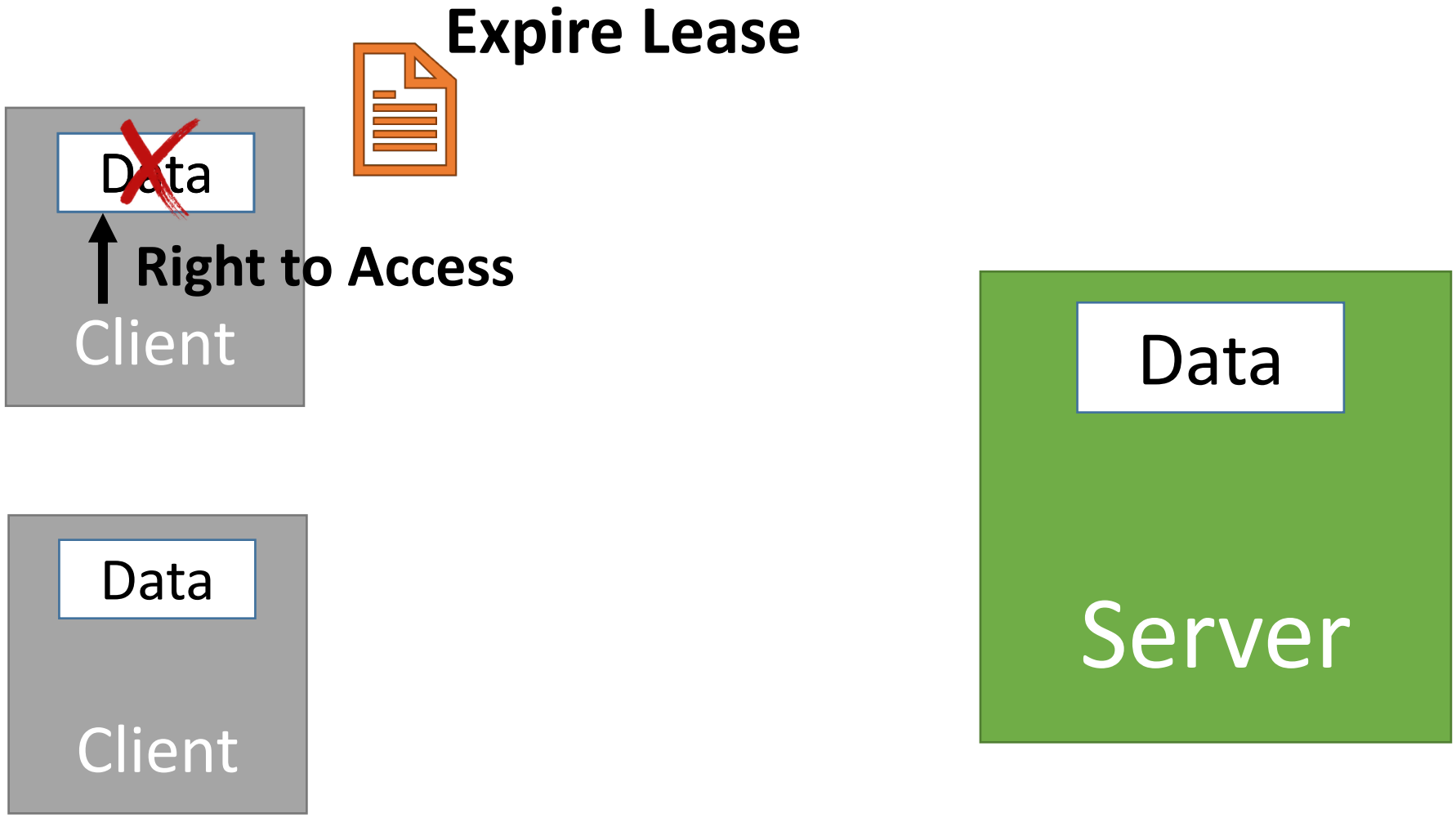
Lease in Distributed System



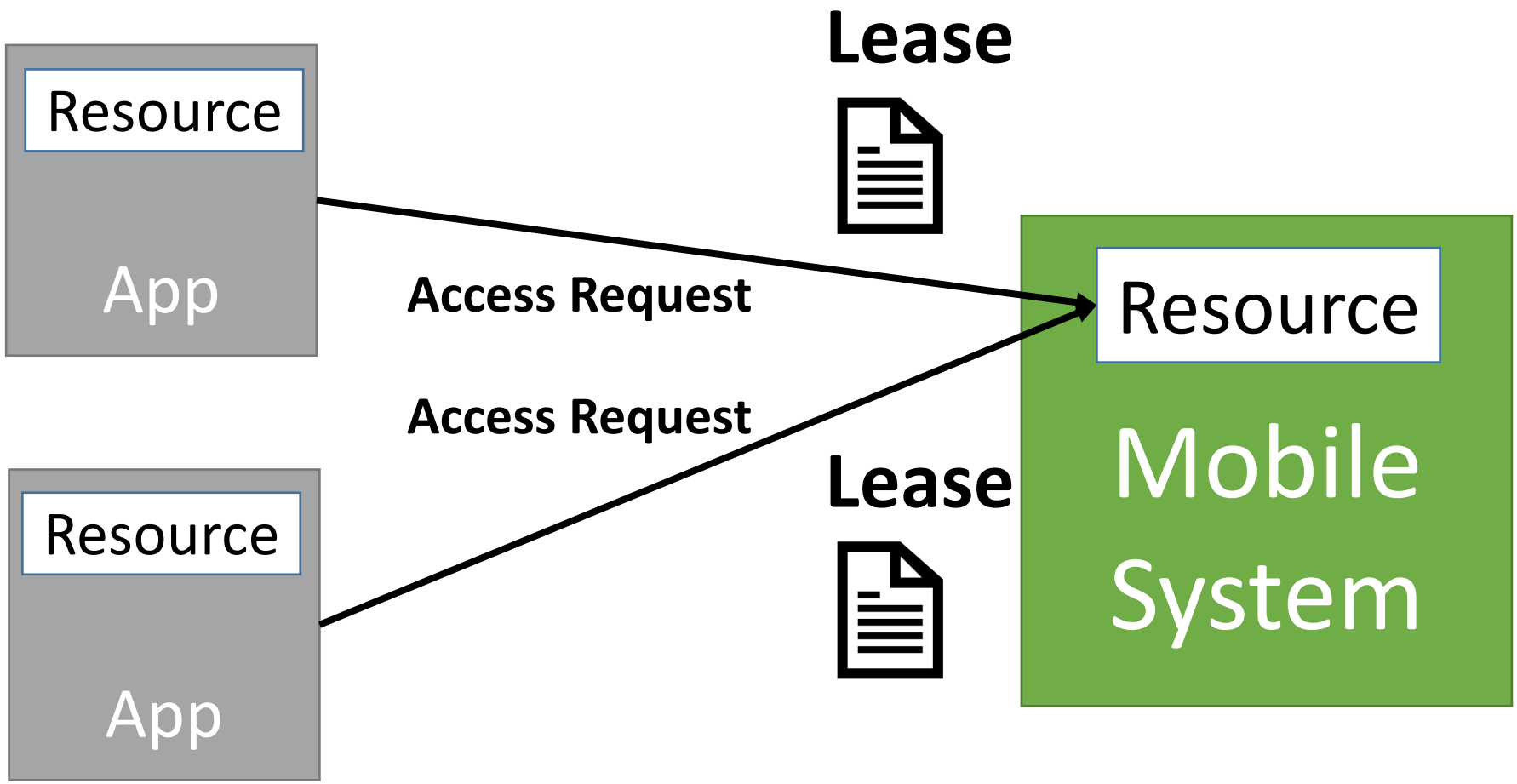
Lease in Distributed System



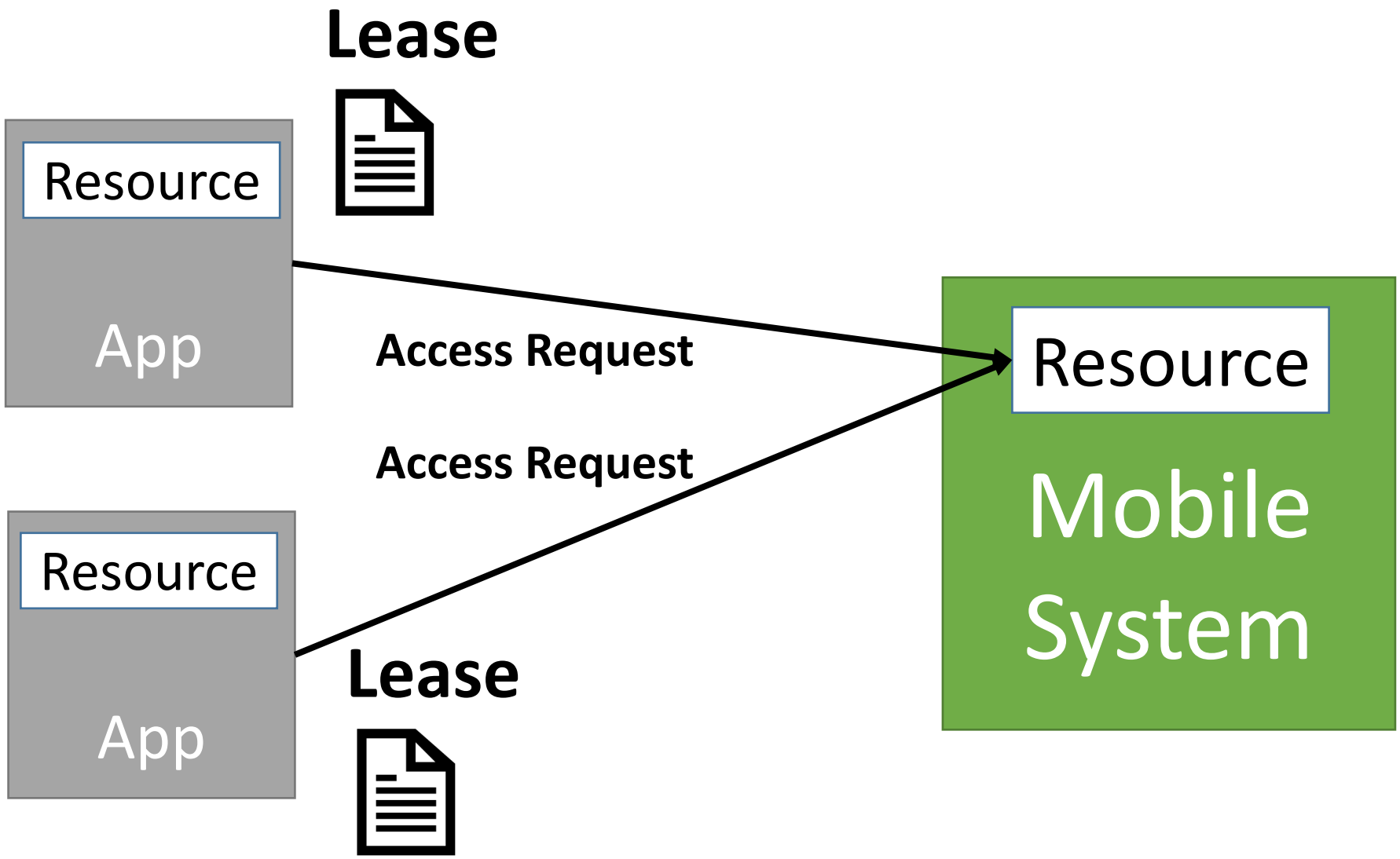
Lease in Distributed System



Lease Abstraction In Mobile System

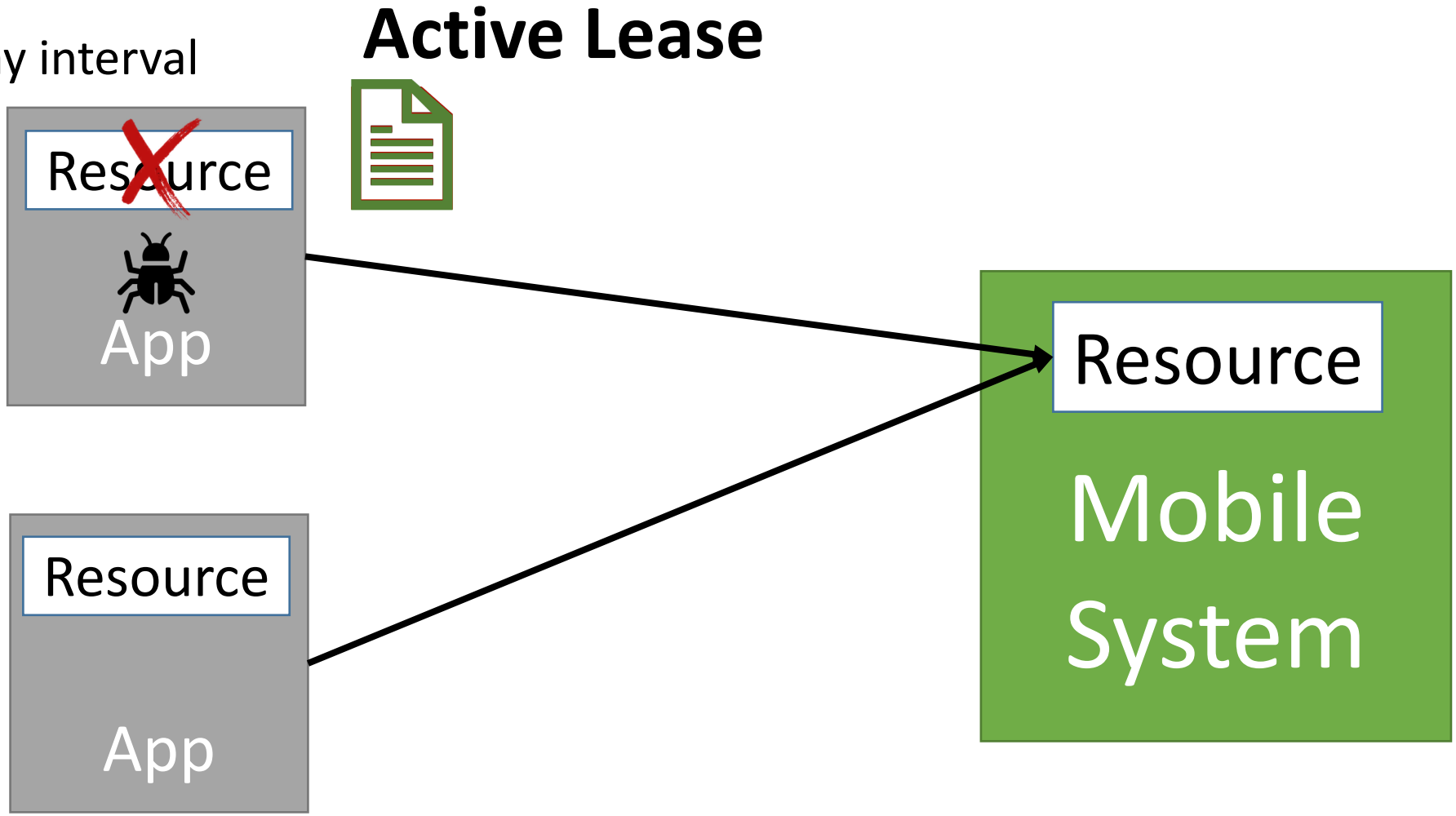


Lease Abstraction In Mobile System



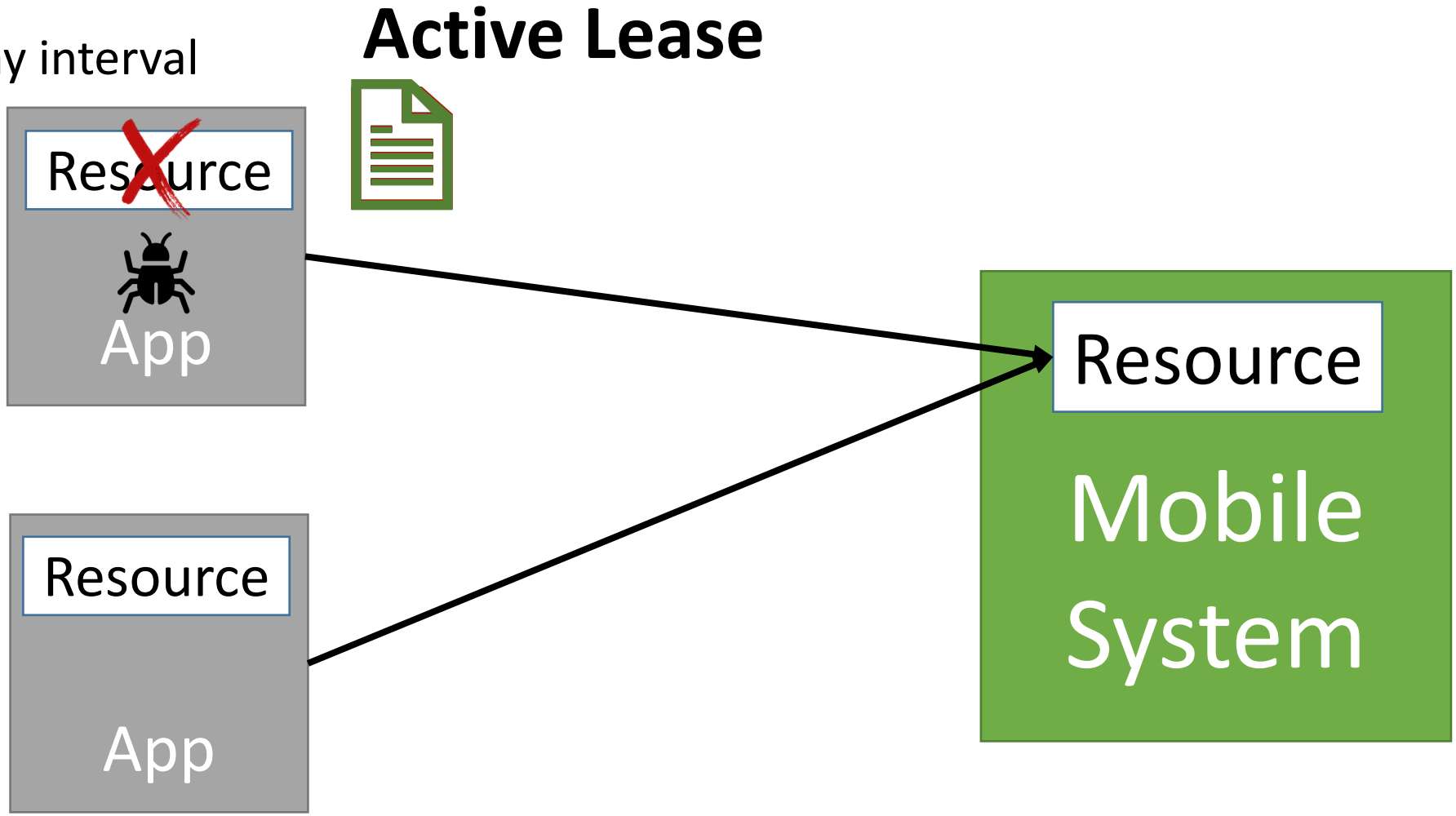
Lease Deferred State

Revoke for Delay interval



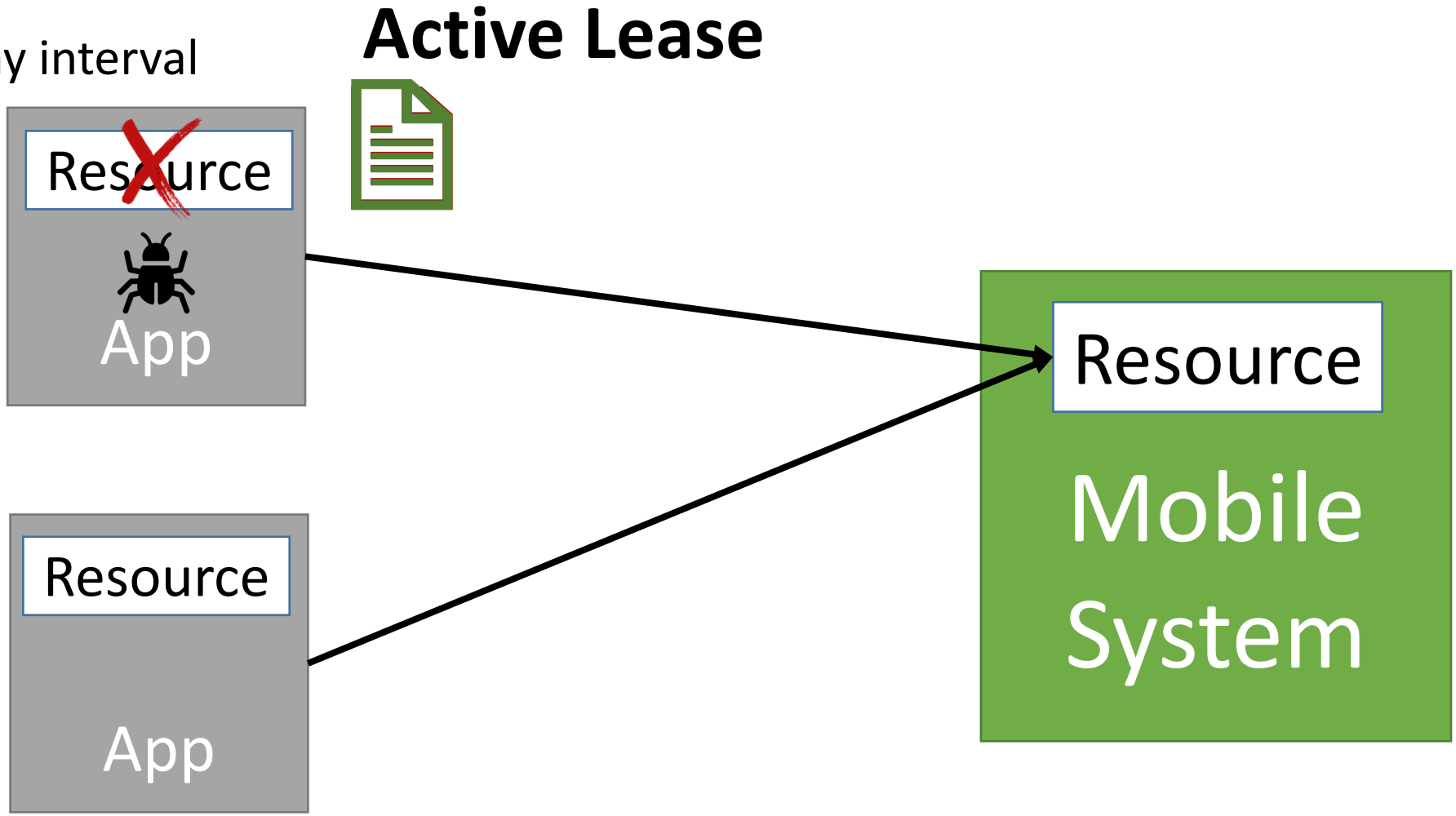
Lease Deferred State

Revoke for Delay interval



Lease Deferred State

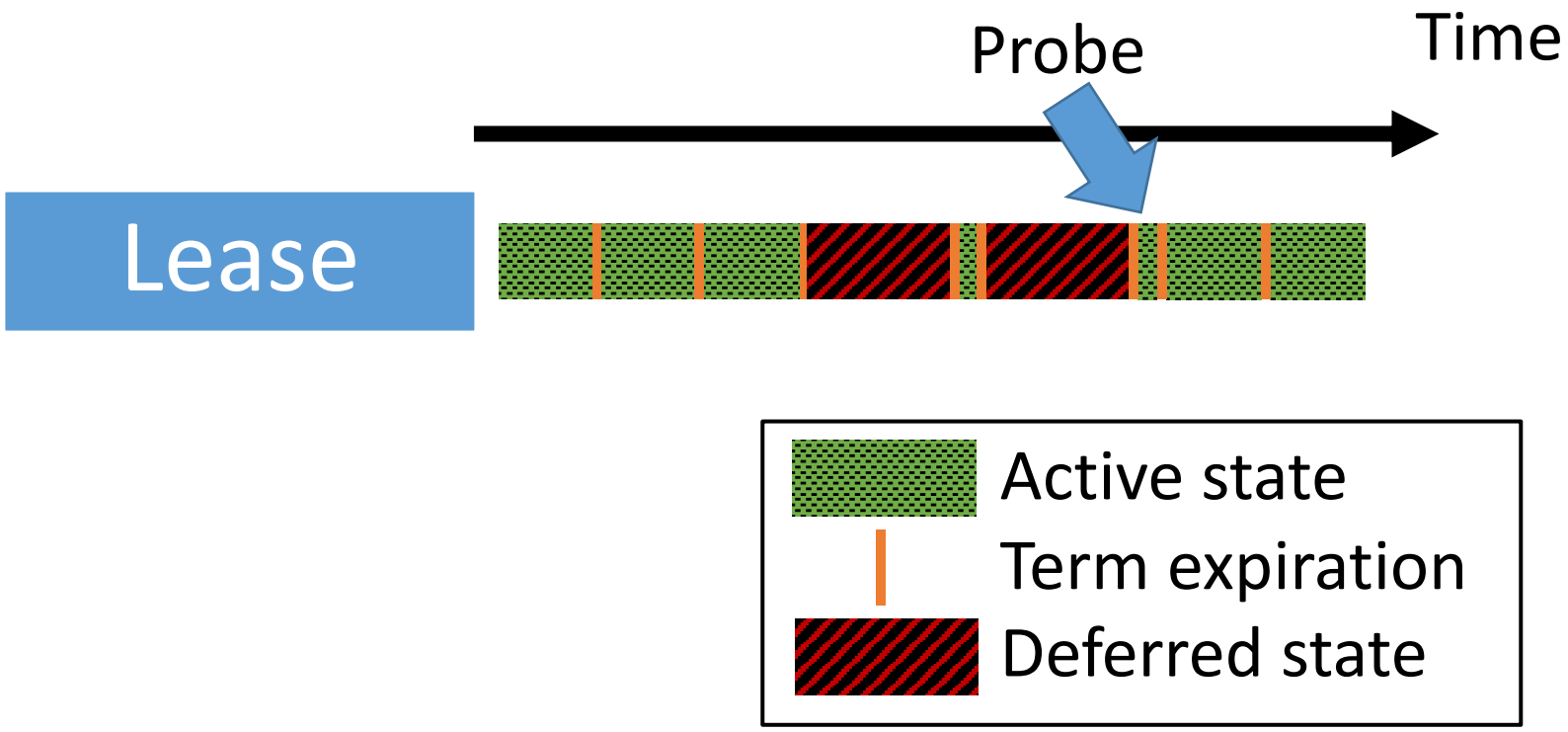
Revoke for Delay interval



An Example – Buggy K-9 Mail

Code Snippet

```
wakeLock.acquire();
while (!stop){
    resp=connect();
        ...
}
wakeLock.release();
```

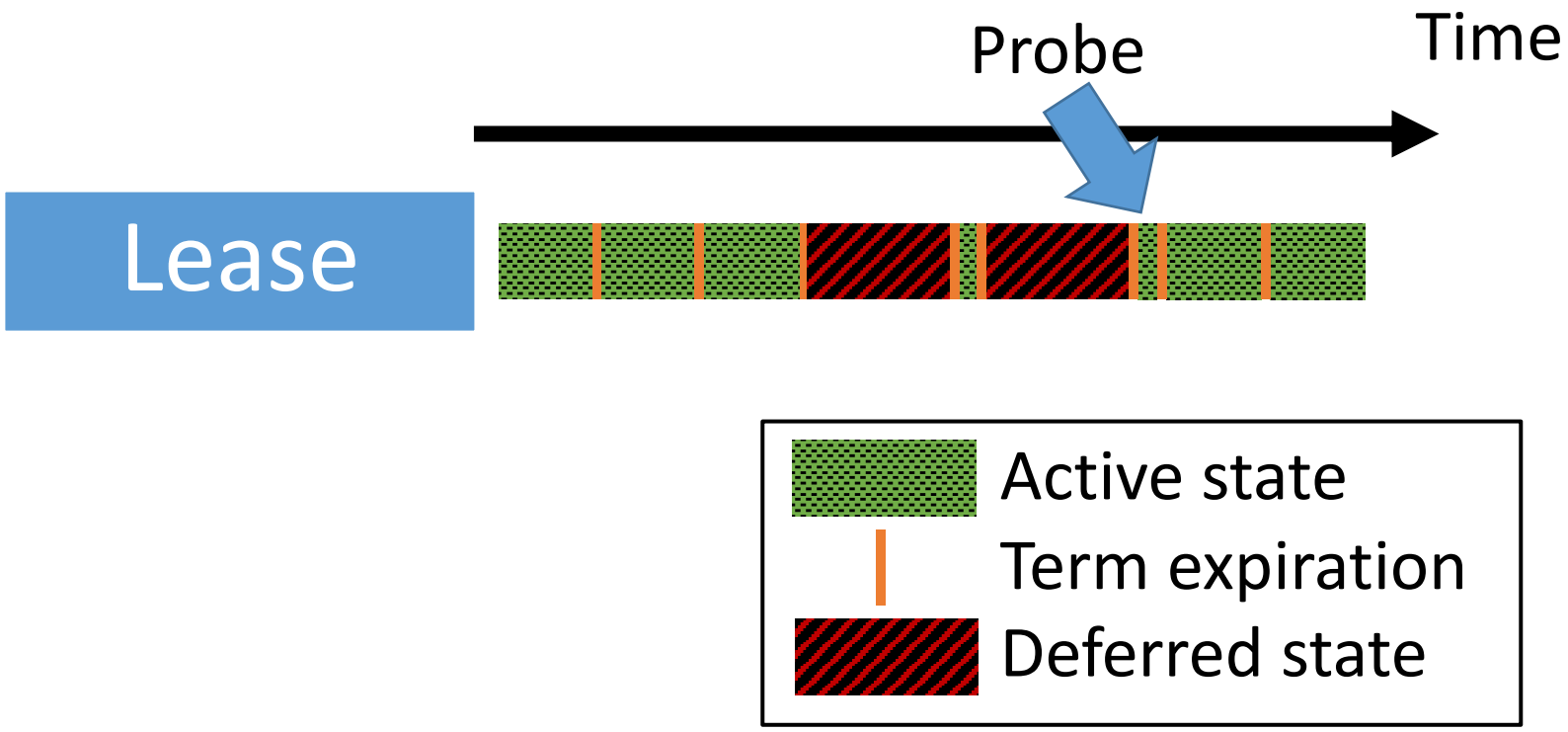


WakeLock: a feature to allow apps to control CPU power state

An Example – Buggy K-9 Mail

Code Snippet

```
wakeLock.acquire();
while (!stop){
    resp=connect();
        ...
}
wakeLock.release();
```

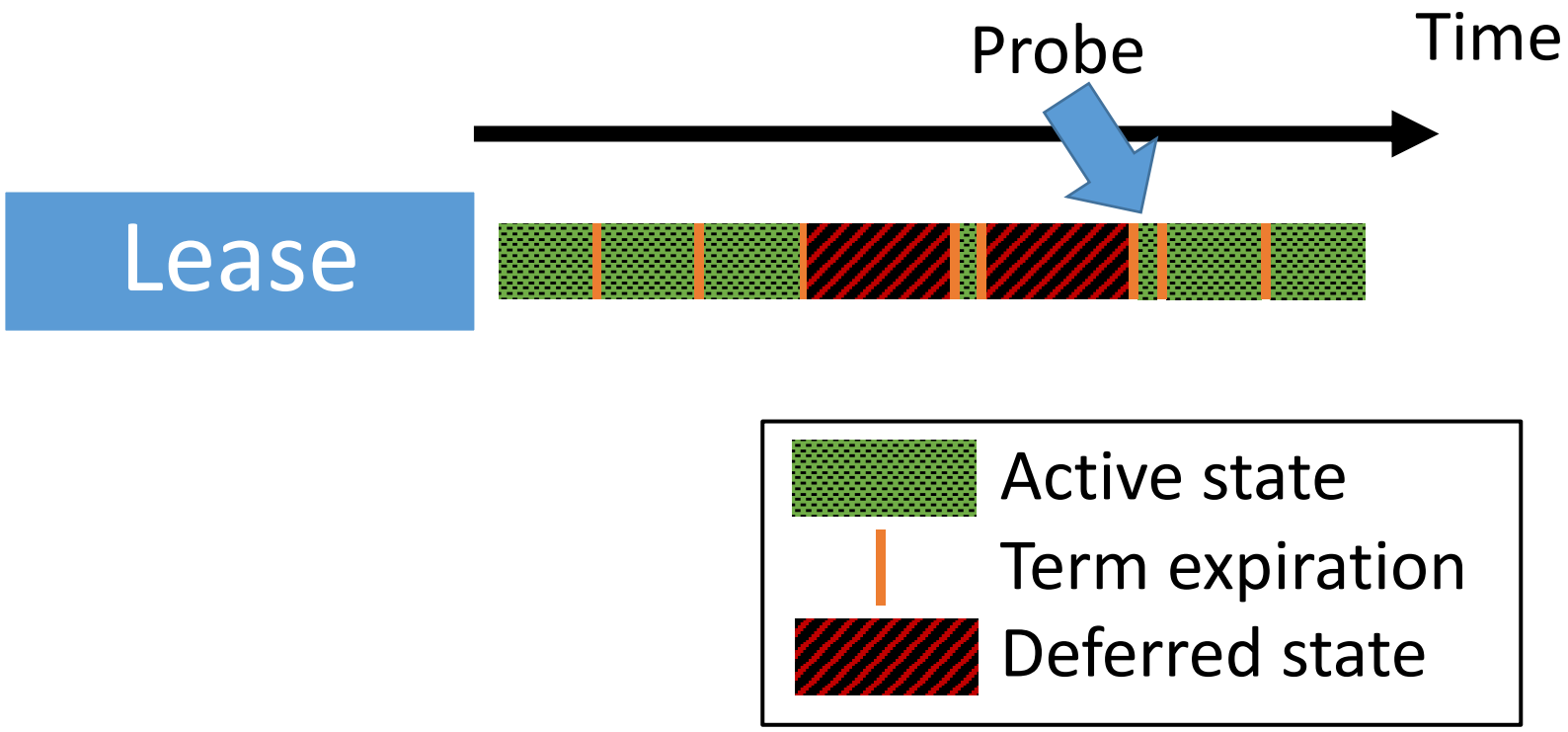


WakeLock: a feature to allow apps to control CPU power state

An Example – Buggy K-9 Mail

Code Snippet

```
wakeLock.acquire();
while (!stop){
    resp=connect();
        ...
}
wakeLock.release();
```

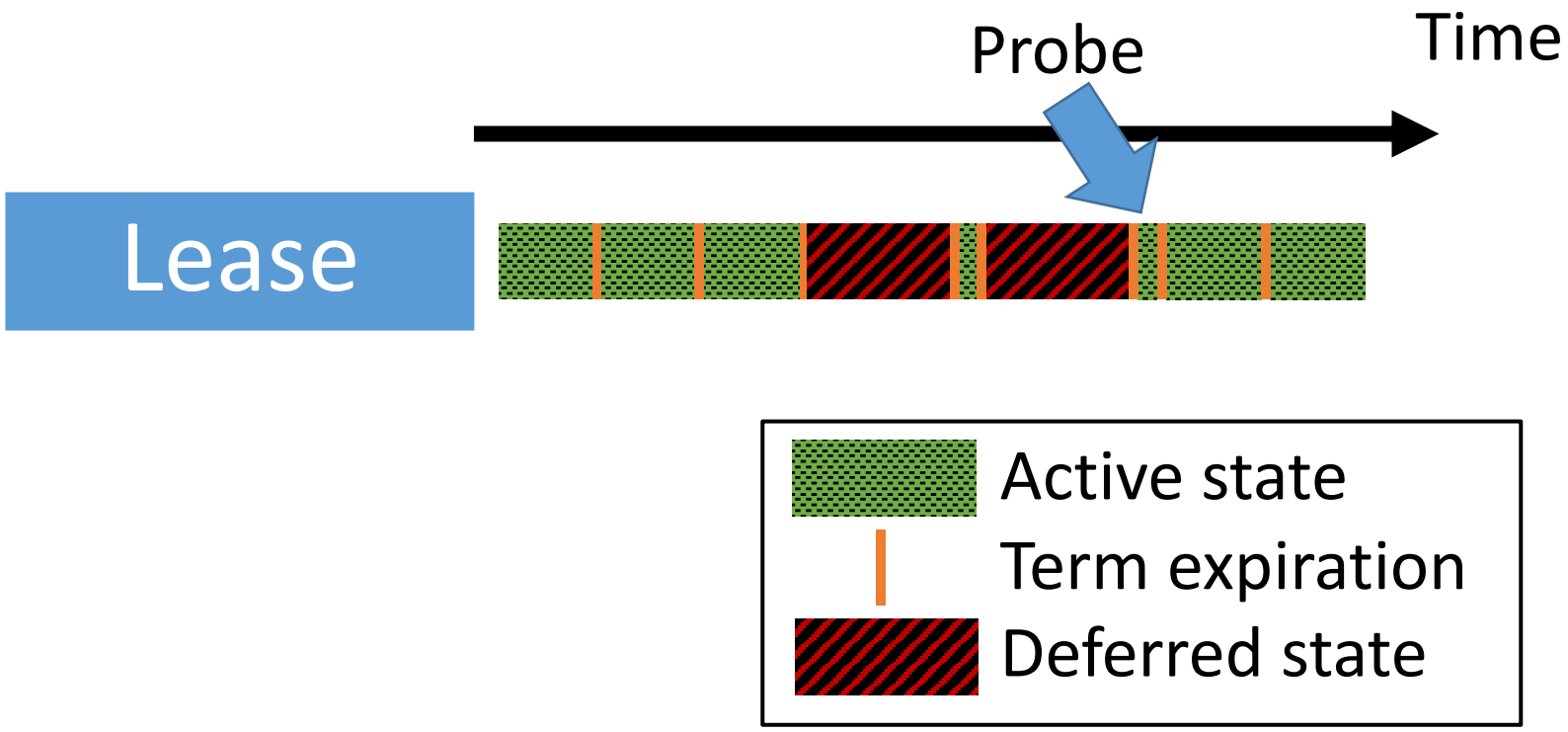


WakeLock: a feature to allow apps to control CPU power state

An Example – Buggy K-9 Mail

Code Snippet

```
wakeLock.acquire();
while (!stop){
    resp=connect();
        ...
}
wakeLock.release();
```

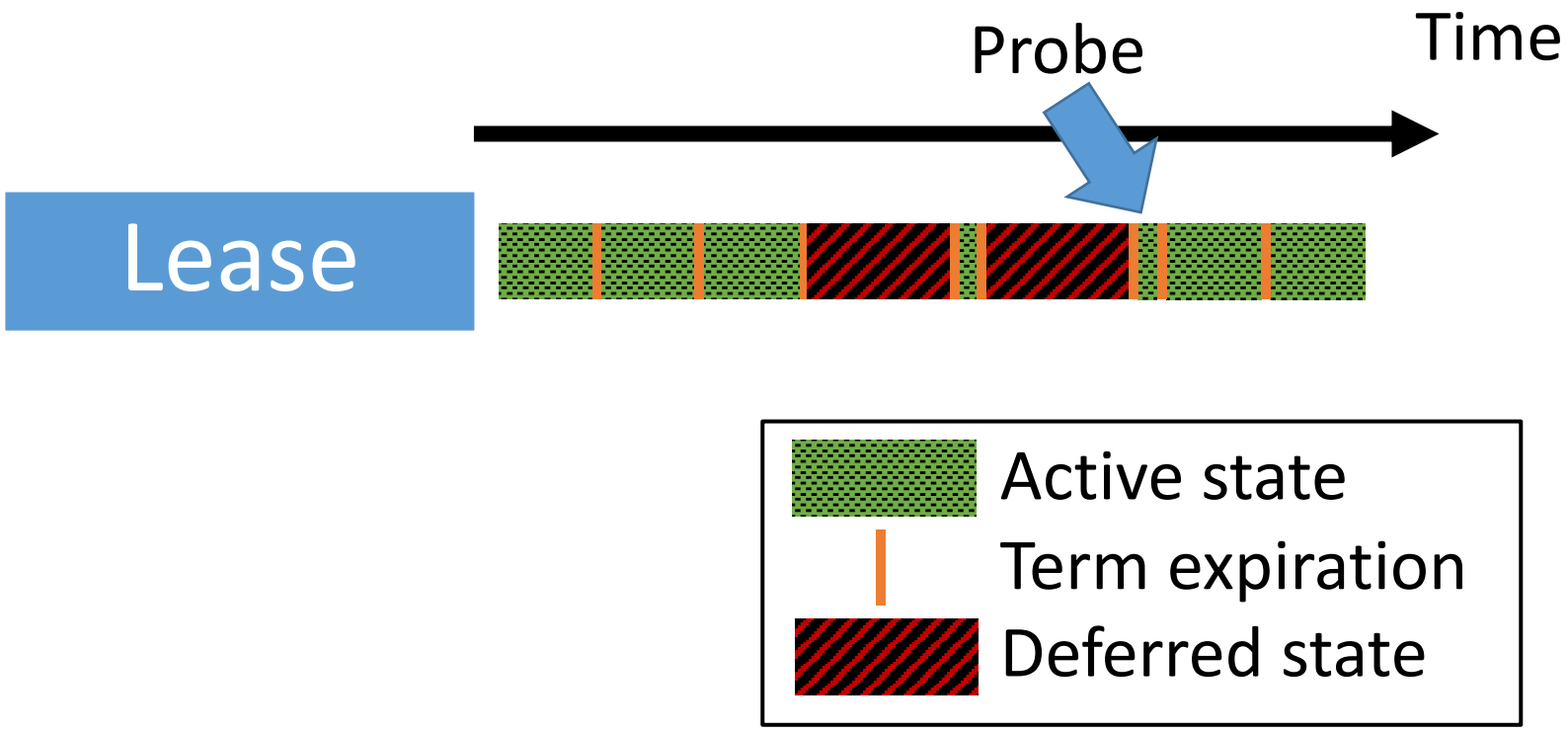


WakeLock: a feature to allow apps to control CPU power state

An Example – Buggy K-9 Mail

Code Snippet

```
wakeLock.acquire();
while (!stop){
    resp=connect();
        ...
}
wakeLock.release();
```

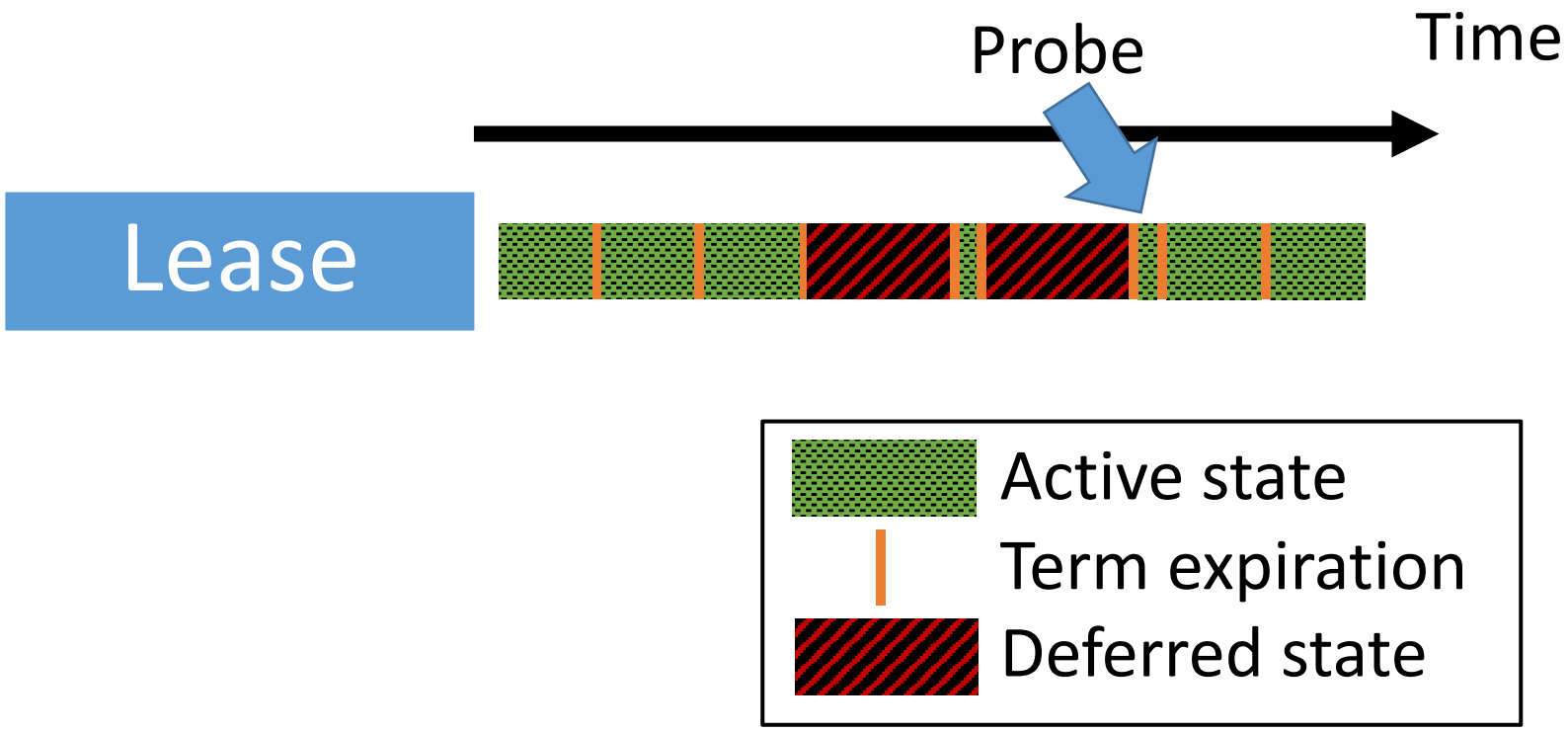


WakeLock: a feature to allow apps to control CPU power state

An Example – Buggy K-9 Mail

Code Snippet

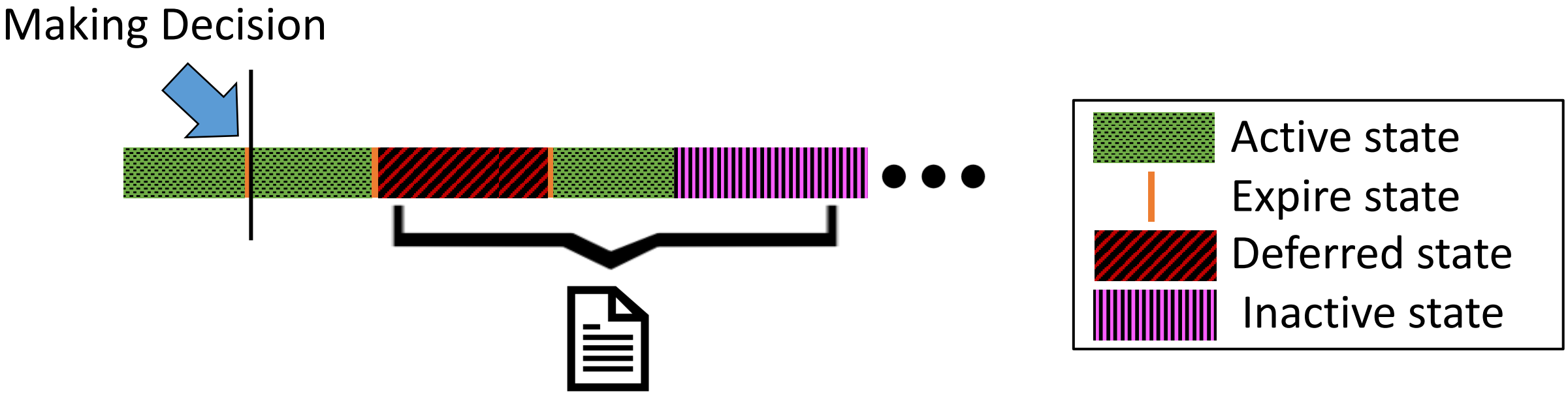
```
wakeLock.acquire();
while (!stop){
    resp=connect();
        ...
}
wakeLock.release();
```



WakeLock: a feature to allow apps to control CPU power state

Lease Decision Rules

Make a lease decision at the end of lease term



Outline

- Motivation
- Lease Abstraction
- **Making Lease Decision**
- **Design of LeaseOS**
- **Evaluation**

Experiment Setup

How buggy apps behave at runtime?

- Measure resource related metrics
- Identify reliable indicators

Test Devices

Moto G | Samsung Galaxy | Nexus 4 | Pixel XL

Low-end

Heavily used

High-end

Lightly used

Experiment Setup

How buggy apps behave at runtime?

- Measure resource related metrics
- Identify reliable indicators

Test Devices

Moto G | Samsung Galaxy | Nexus 4 | Pixel XL

Low-end

Heavily used

High-end

Lightly used

Experiment Setup

How buggy apps behave at runtime?

- Measure resource related metrics
- Identify reliable indicators

Test Devices

Moto G | Samsung Galaxy | Nexus 4 | Pixel XL

Low-end

Heavily used

High-end

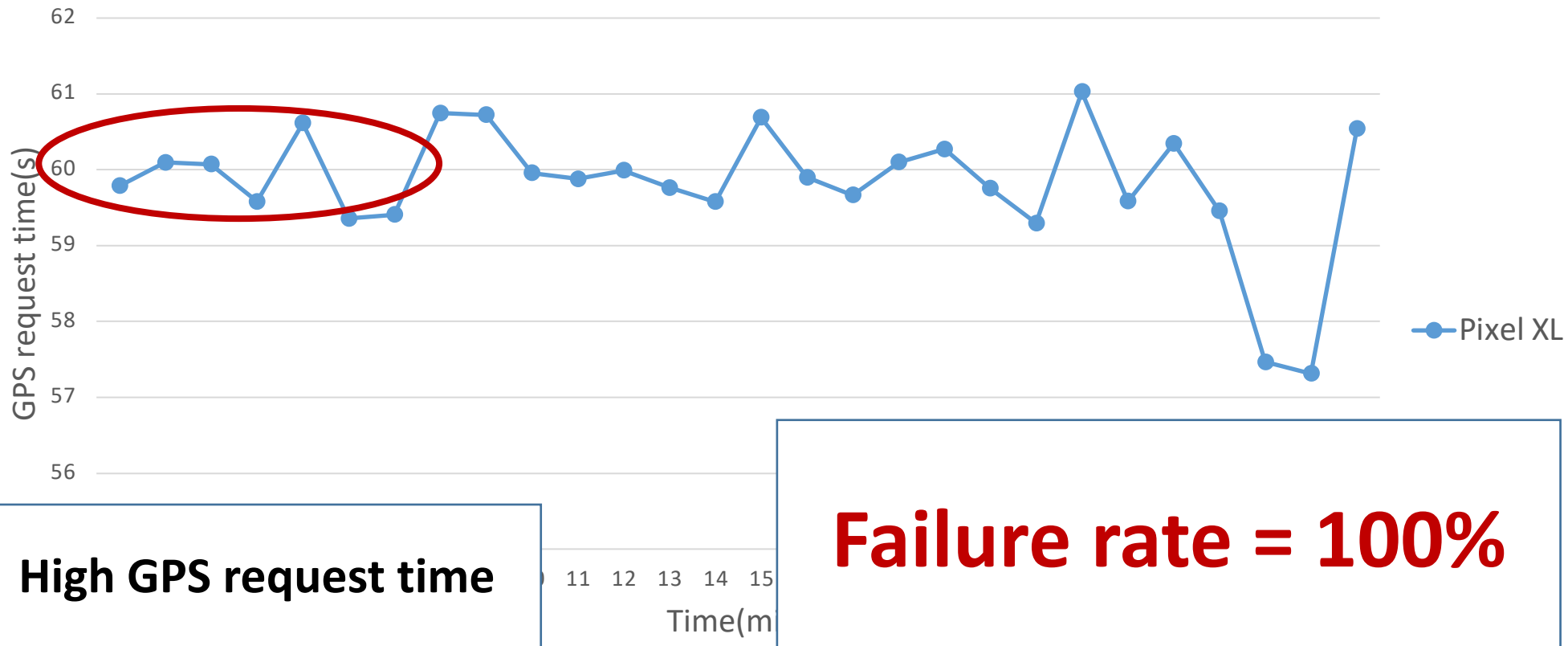
Lightly used

Example 1 - BetterWeather

Weather widget app

Keeps asking for GPS when the app can't get GPS

Test result on Pixel XL



High GPS request time

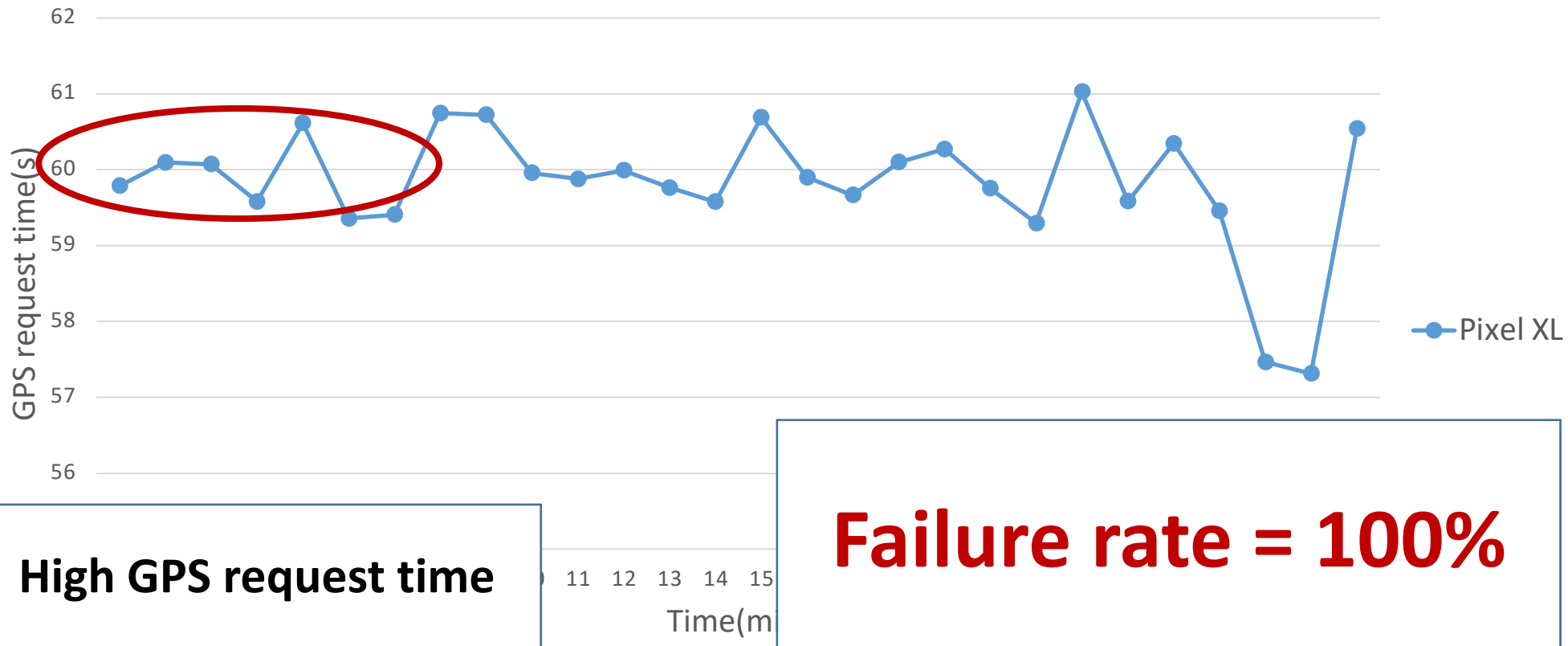
Failure rate = 100%

Example 1 - BetterWeather

Weather widget app

Keeps asking for GPS when the app can't get GPS

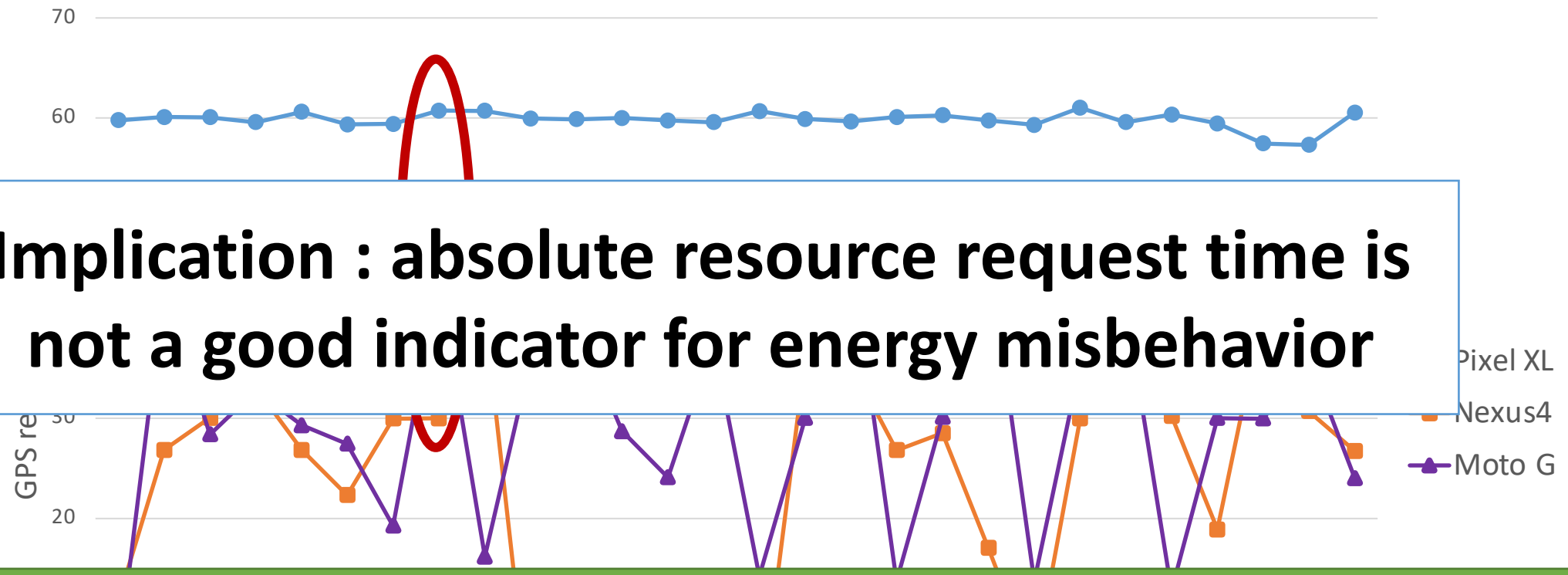
Test result on Pixel XL



High GPS request time

Failure rate = 100%

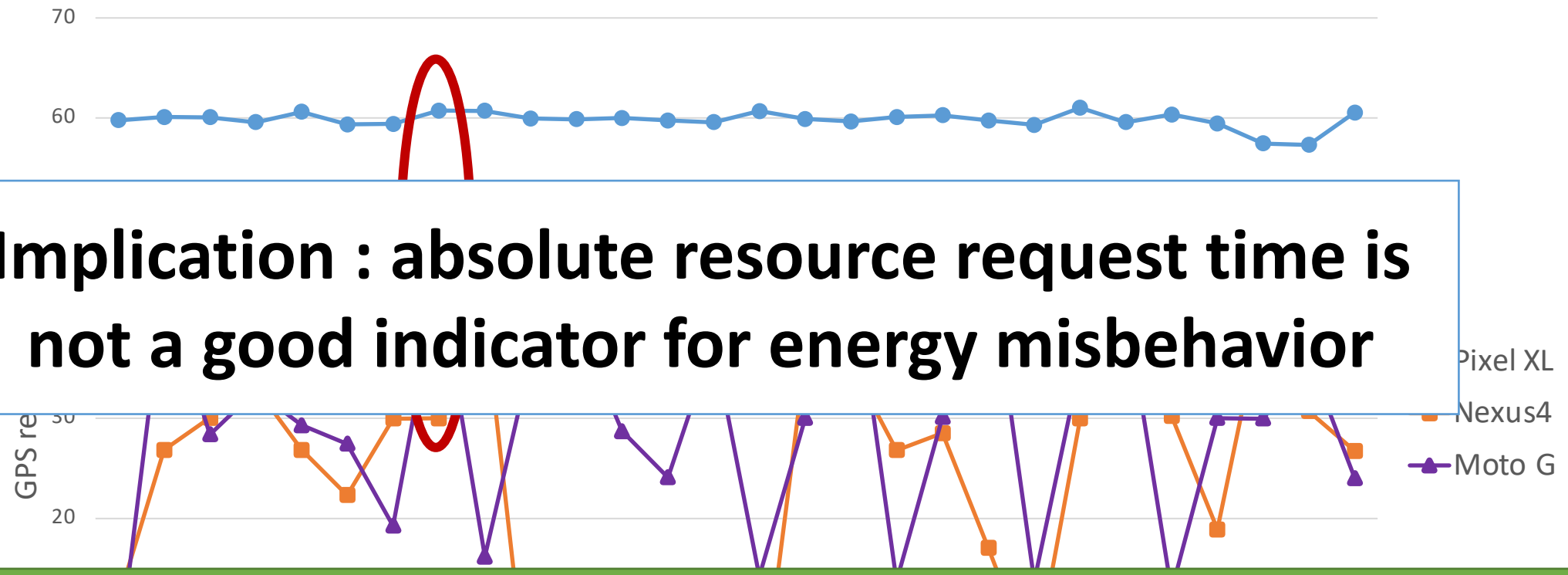
Resource Request Time Varies Among Devices



Implication : absolute resource request time is not a good indicator for energy misbehavior

O1: For the same energy misbehavior, the request time might vary a lot on different devices

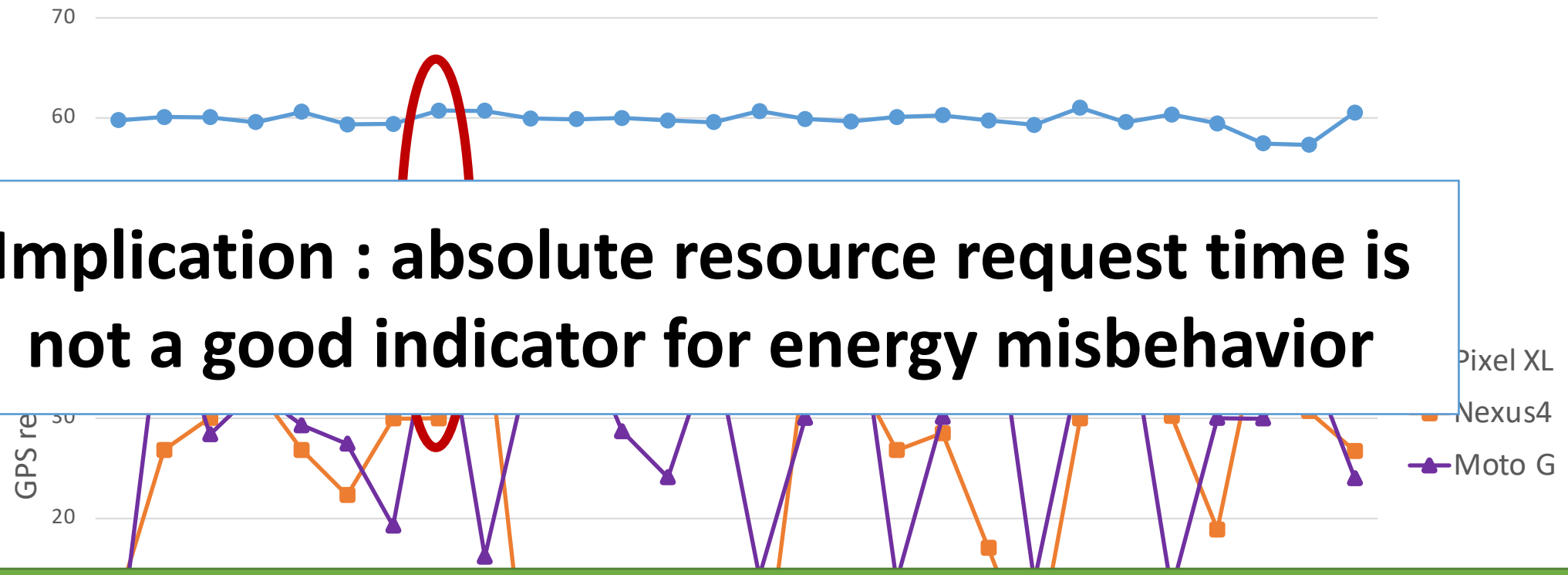
Resource Request Time Varies Among Devices



Implication : absolute resource request time is not a good indicator for energy misbehavior

O1: For the same energy misbehavior, the request time might vary a lot on different devices

Resource Request Time Varies Among Devices



Implication : absolute resource request time is not a good indicator for energy misbehavior

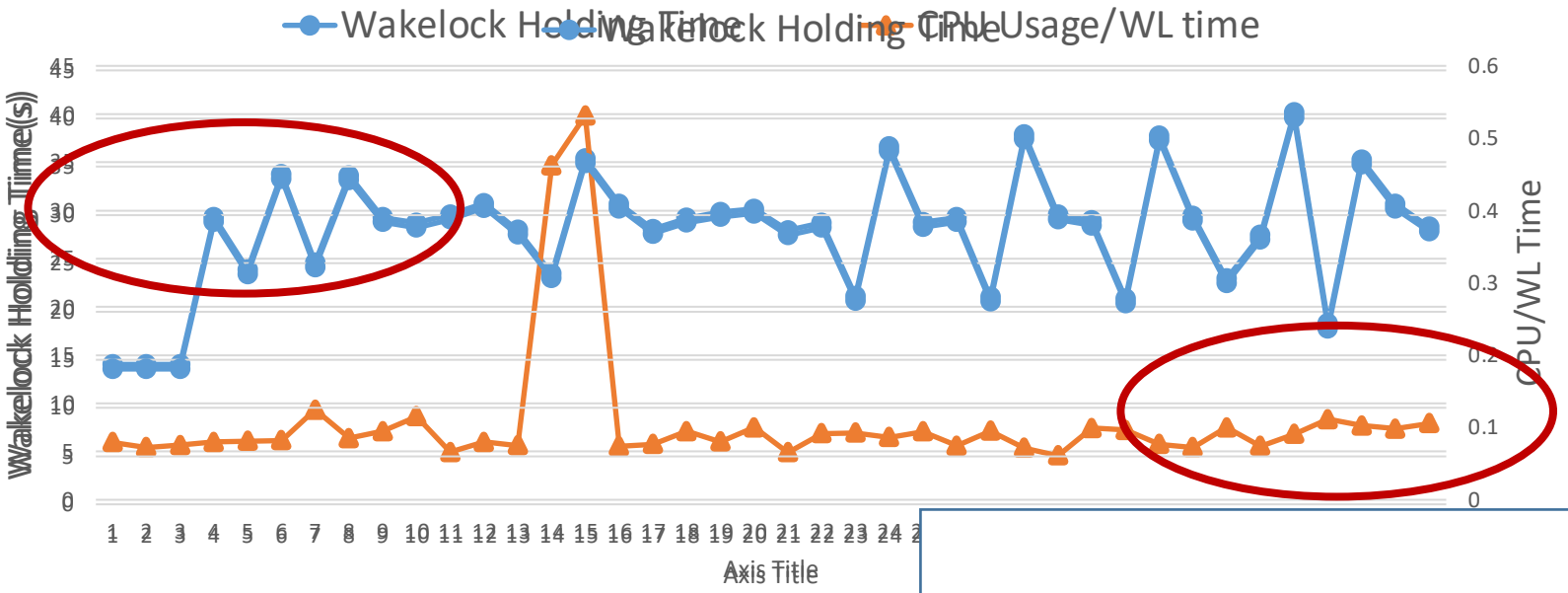
O1: For the same energy misbehavior, the request time might vary a lot on different devices

Example 2 - Kontalk

Messaging app

Releases the wakelock only when the service is destroyed

Test result on Samsung S4



High Wakelock holding time

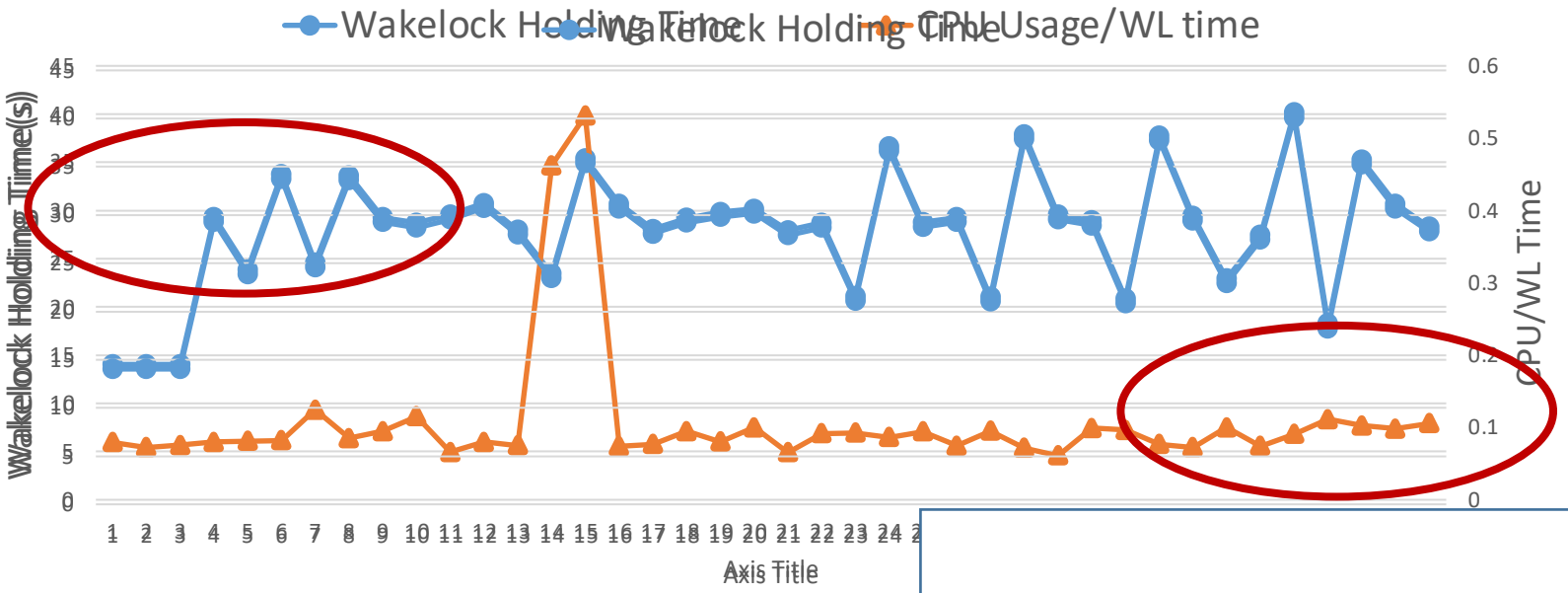
Utilization < 10%

Example 2 - Kontalk

Messaging app

Releases the wakelock only when the service is destroyed

Test result on Samsung S4



High Wakelock holding time

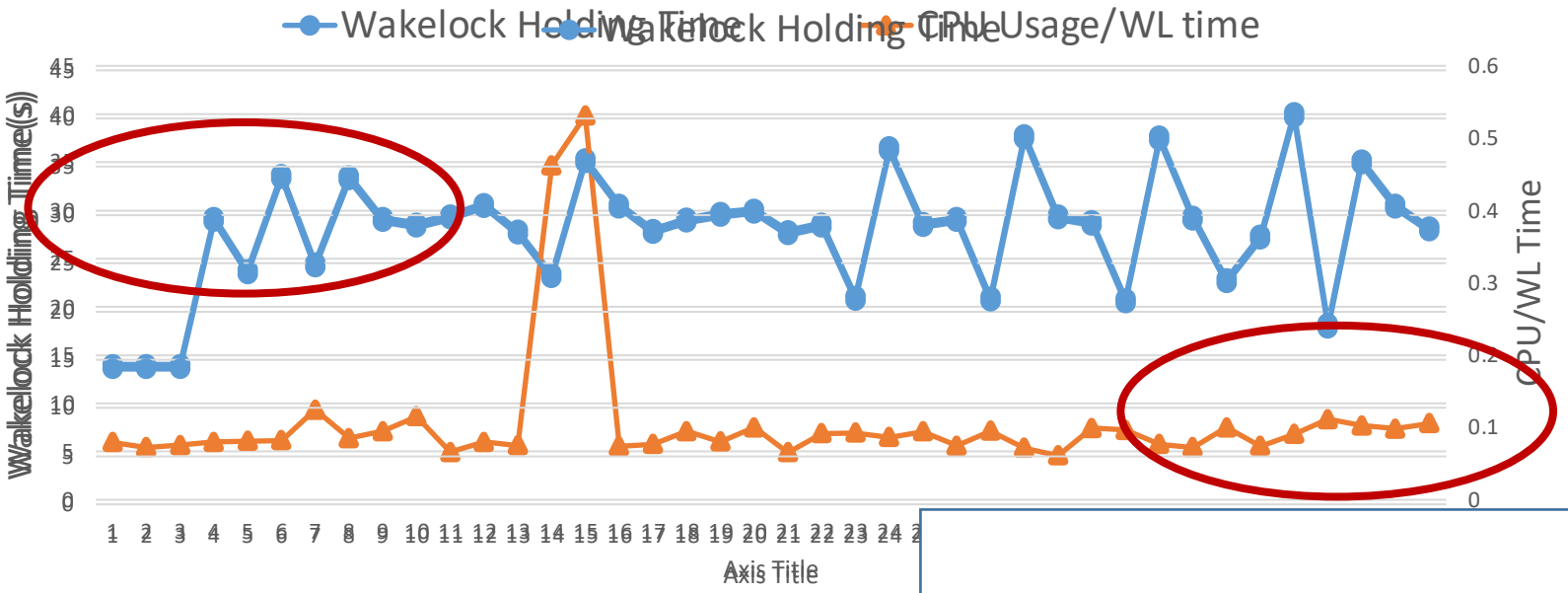
Utilization < 10%

Example 2 - Kontalk

Messaging app

Releases the wakelock only when the service is destroyed

Test result on Samsung S4

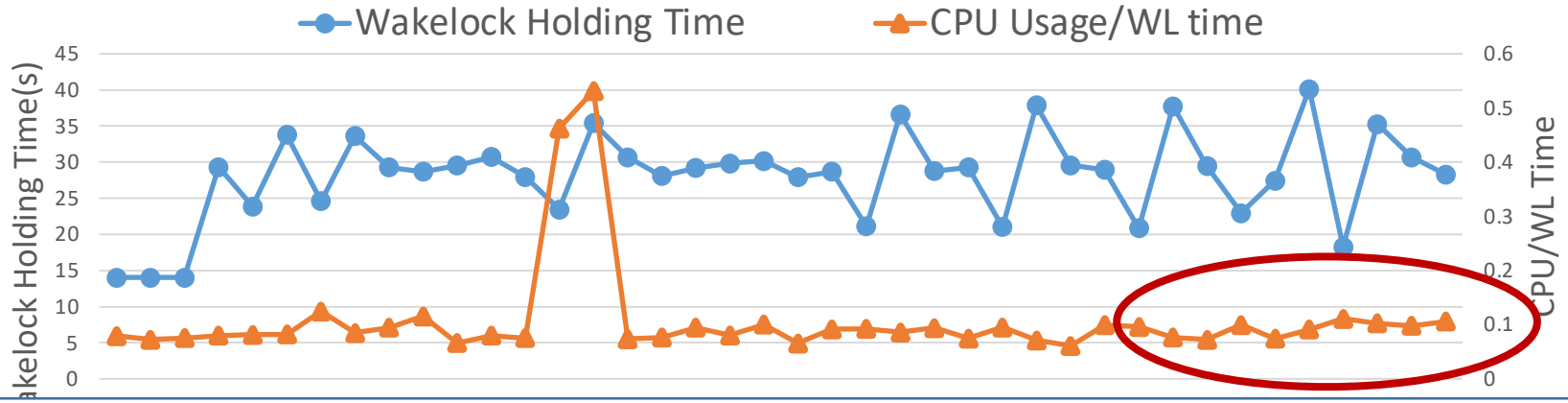


High Wakelock holding time

Utilization < 10%

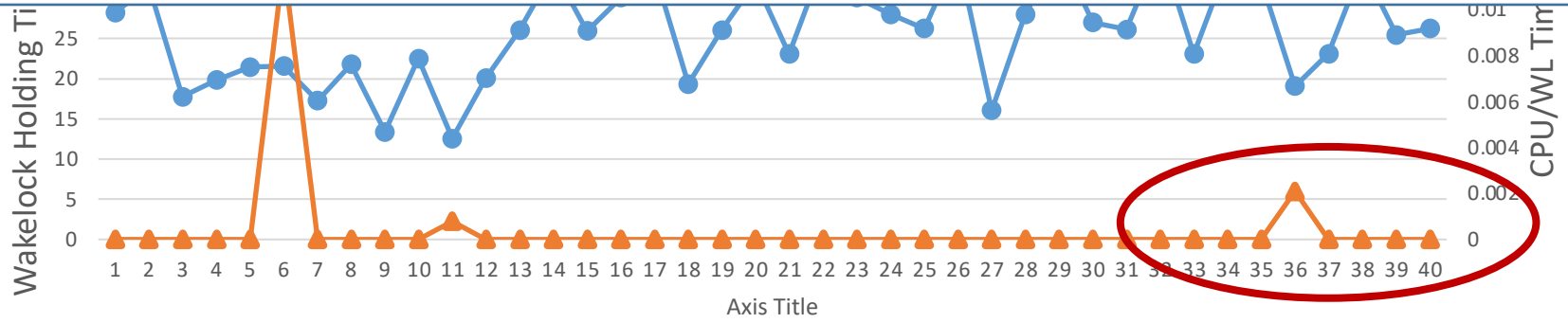
The Low Utilization Ratio Is Consistent

SAMSUNG S4



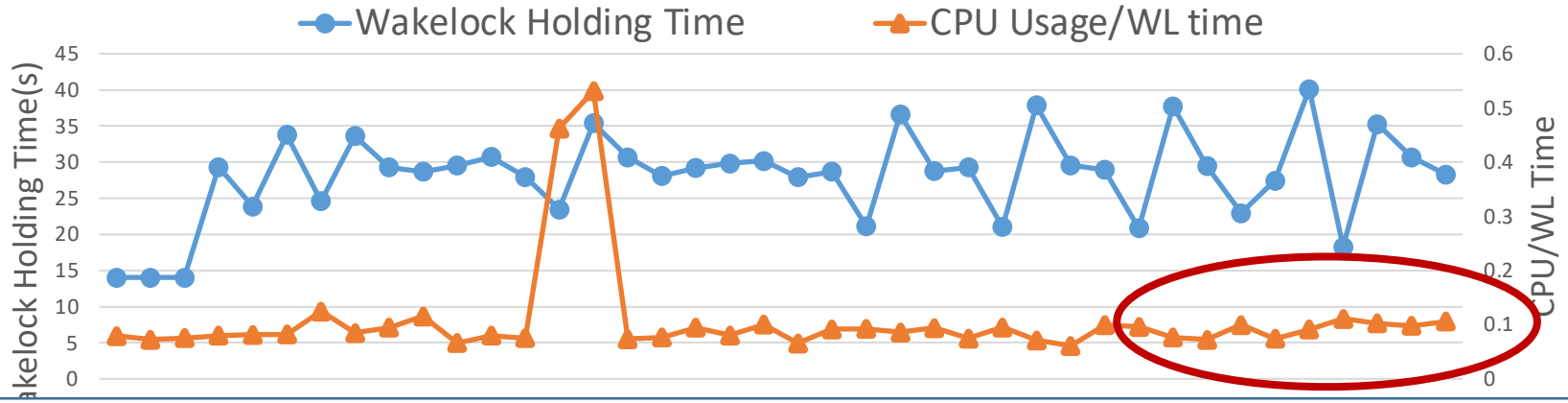
The utilization ratio is always low among different device

Nexus 4



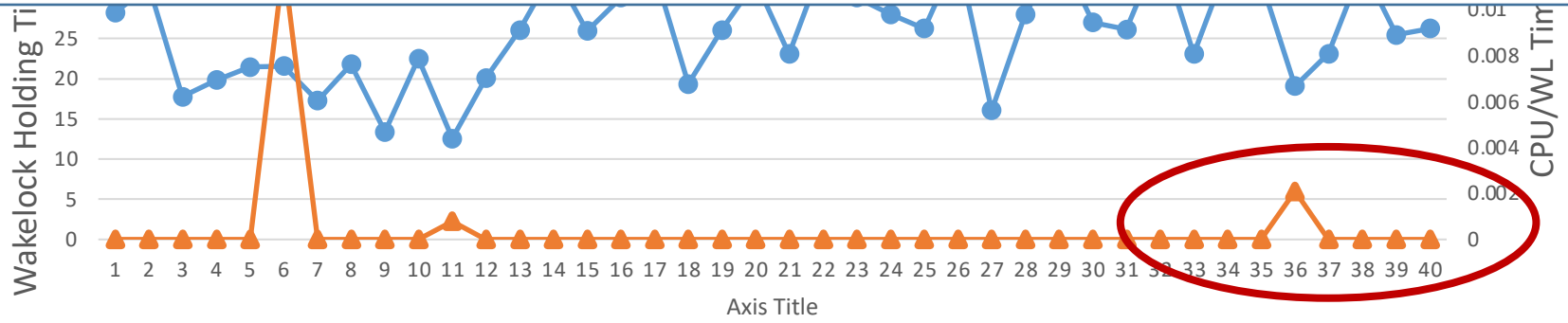
The Low Utilization Ratio Is Consistent

SAMSUNG S4



The utilization ratio is always low among different device

Nexus 4

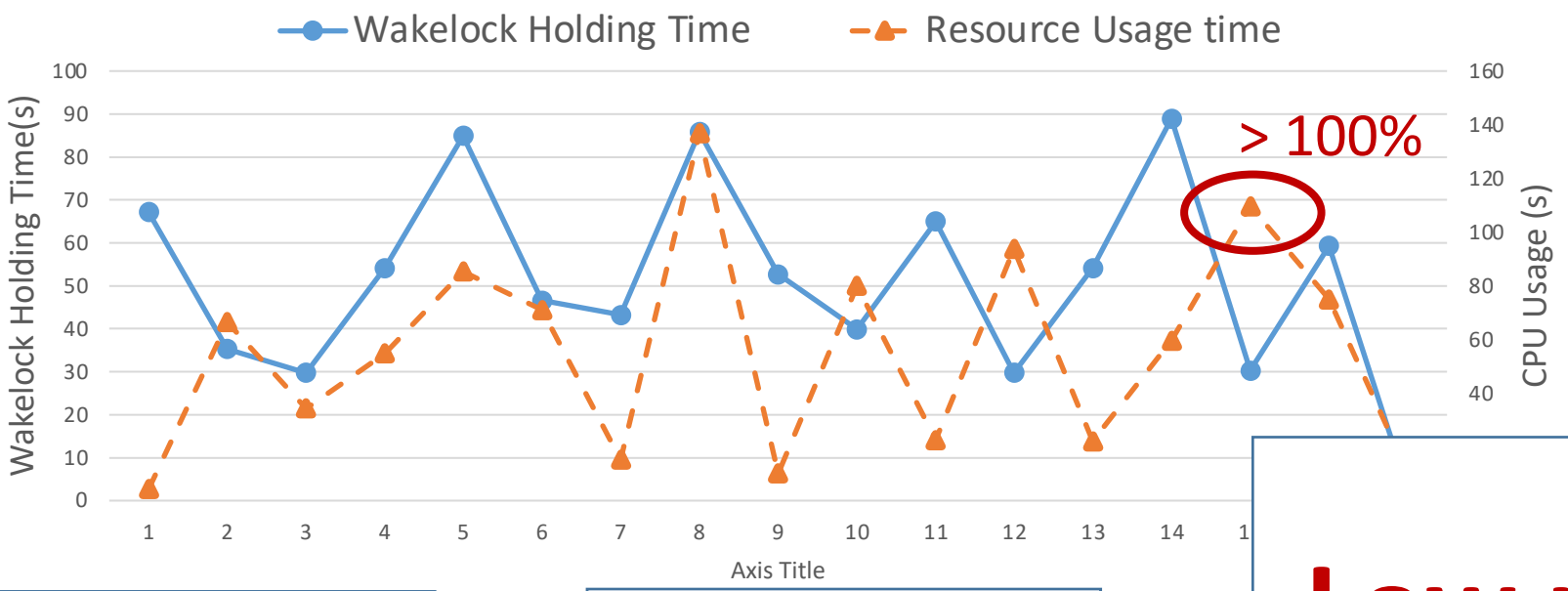


Utility Is Also a Good Indicator

Email app

Gets stuck in a loop when the network is disconnected

Test result on Pixel XL



High Wakelock holding time

High utilization ratio

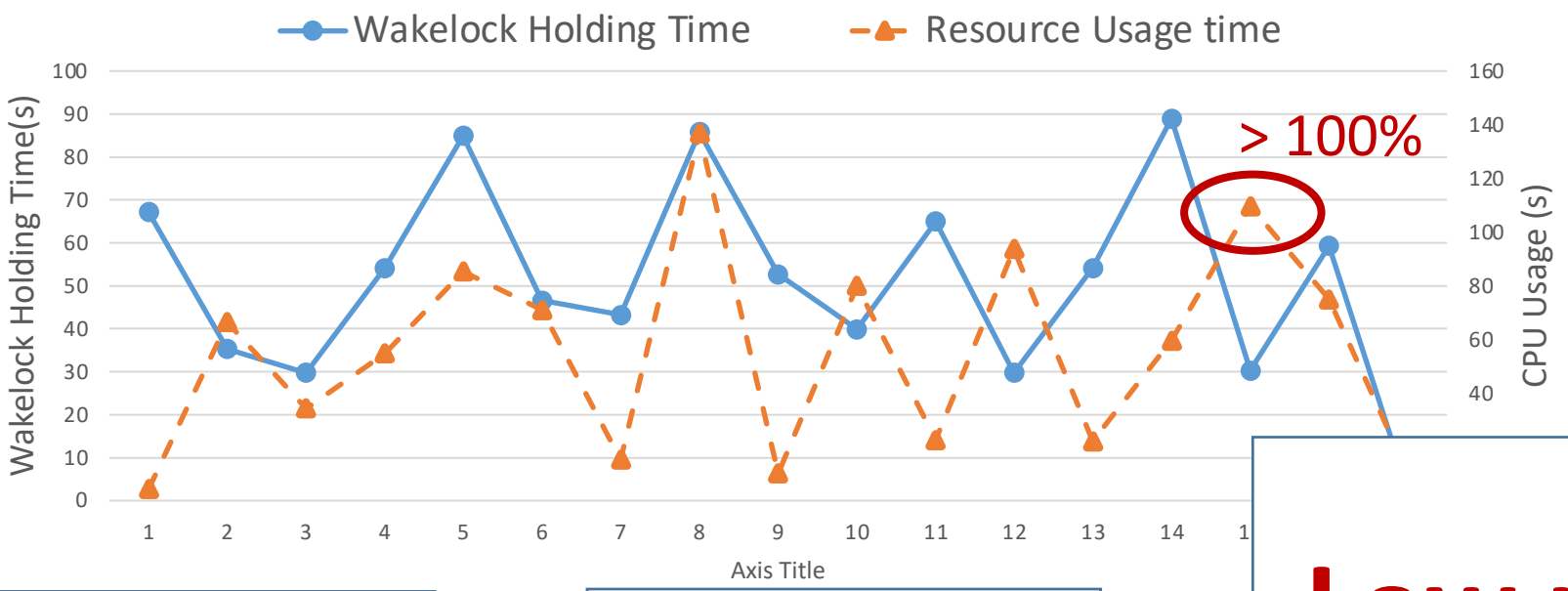
Low utility

Utility Is Also a Good Indicator

Email app

Gets stuck in a loop when the network is disconnected

Test result on Pixel XL

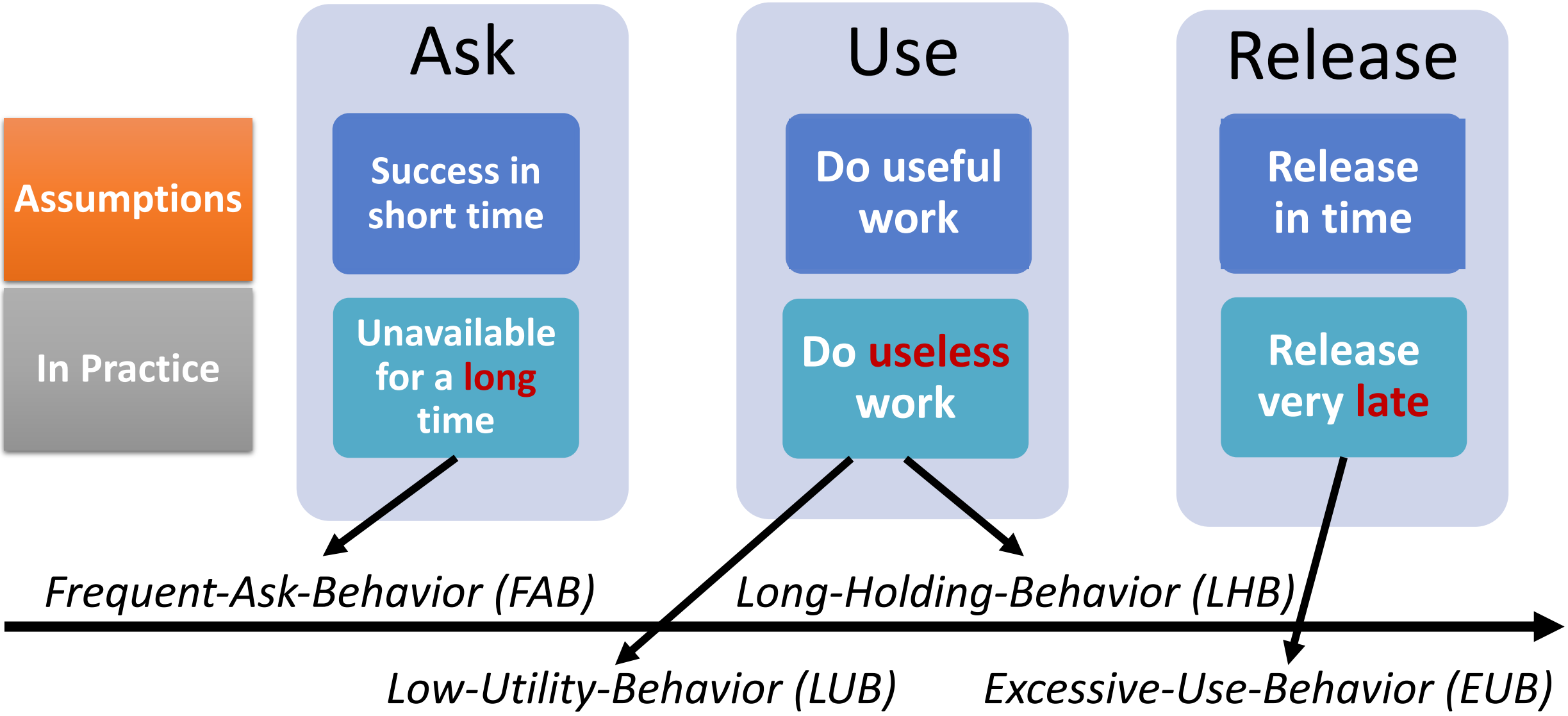


High Wakelock holding time

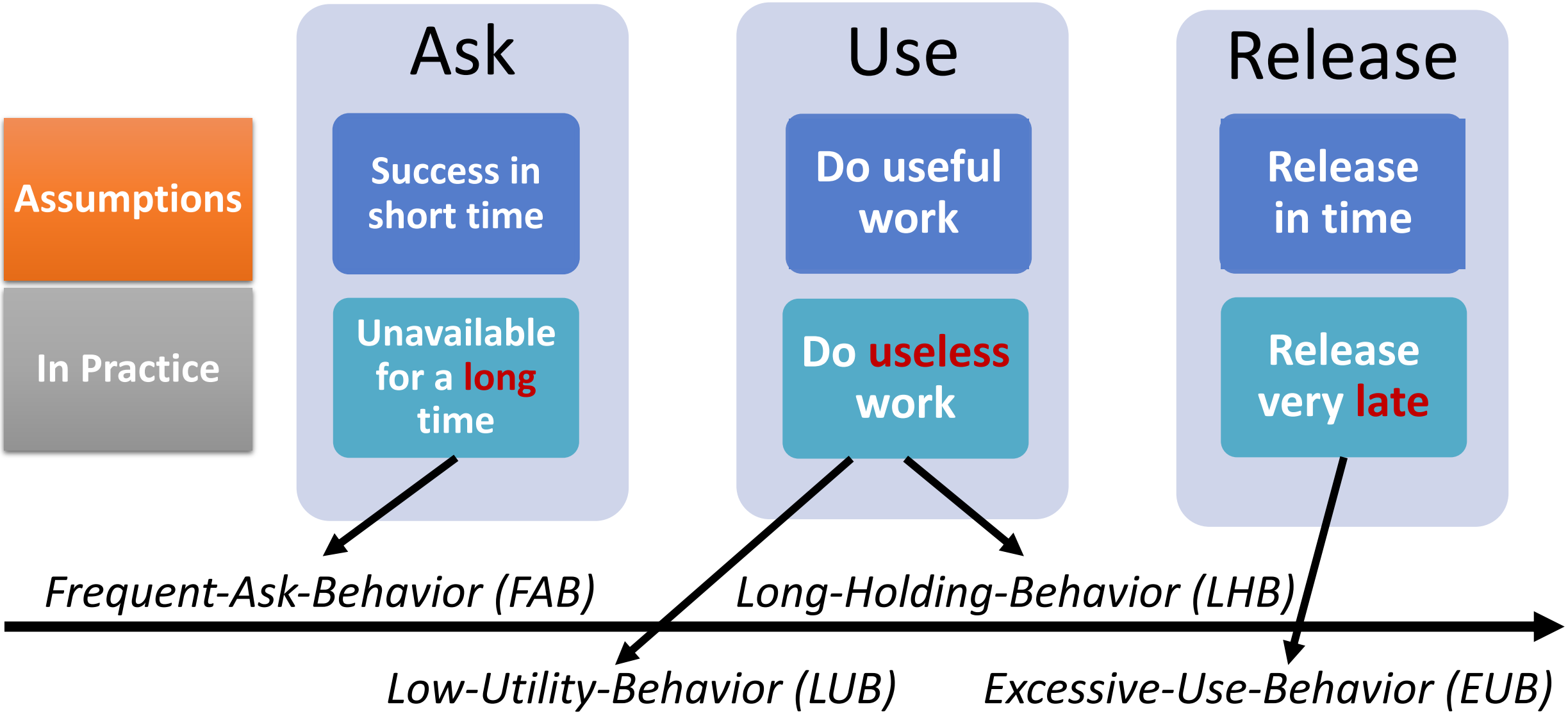
High utilization ratio

Low utility

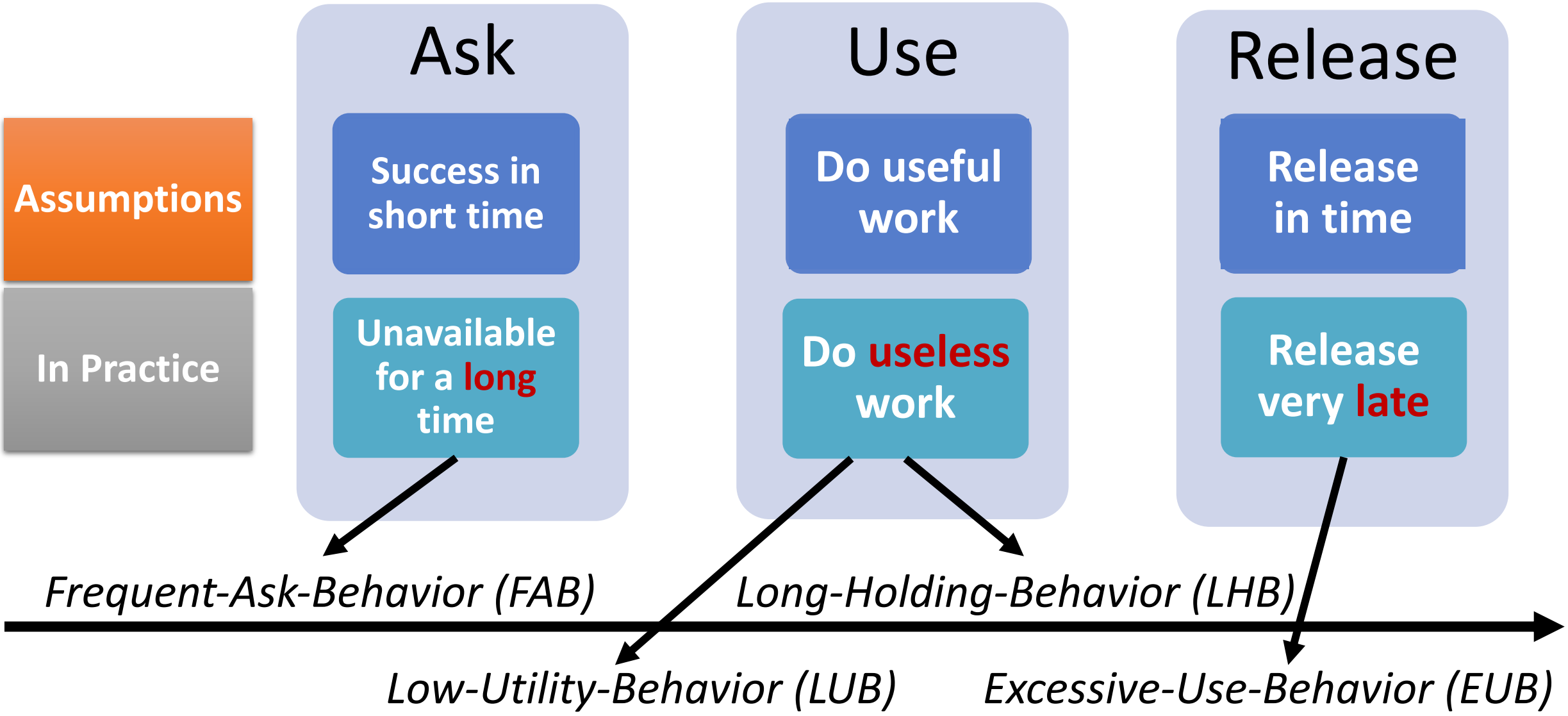
Four Classes of Energy Misbehavior



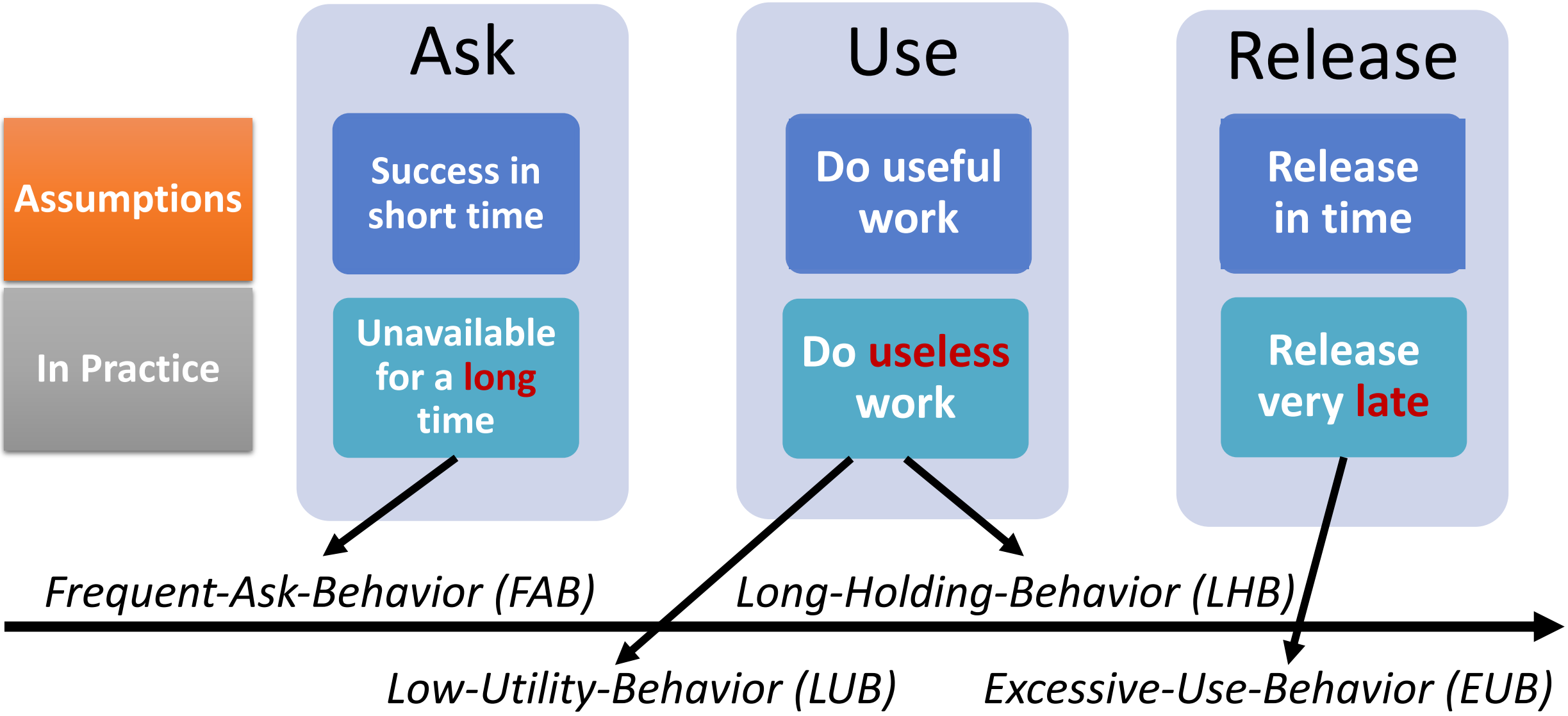
Four Classes of Energy Misbehavior



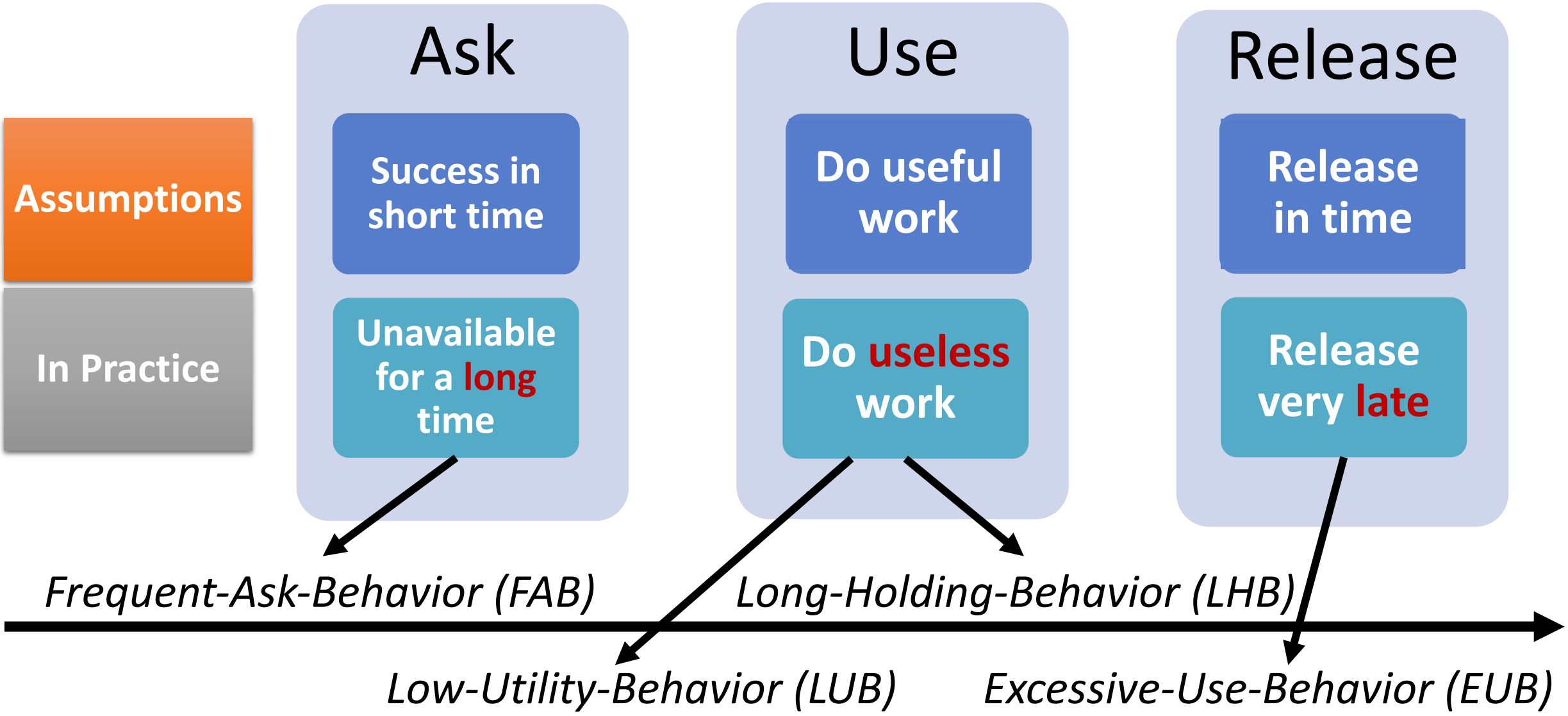
Four Classes of Energy Misbehavior



Four Classes of Energy Misbehavior



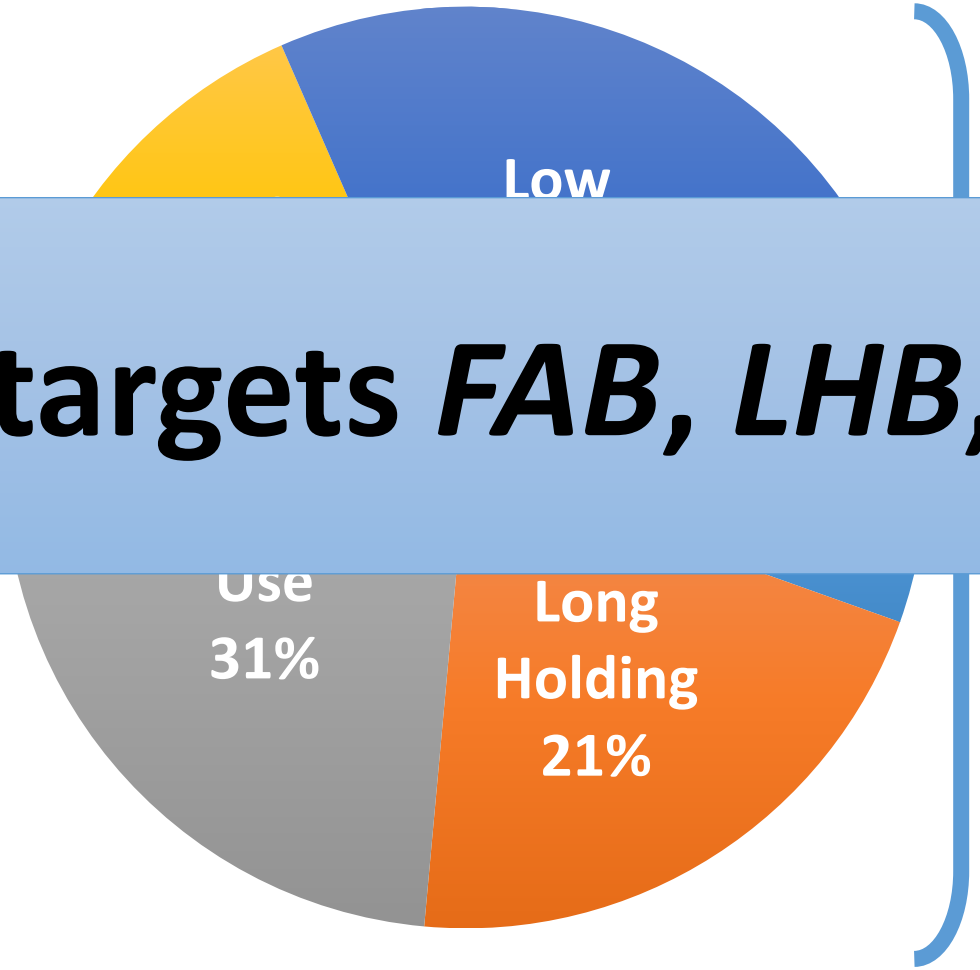
Four Classes of Energy Misbehavior



Prevalence of Each Misbehavior Type

- 109 real-world misbehavior cases

FAB: Frequent-Ask-Behavior
LHB: Long-Holding-Behavior
LUB: Low-Utility-Behavior
EUB: Excessive-Use-Behavior



LeaseOS targets *FAB*, *LHB*, and *LUB*

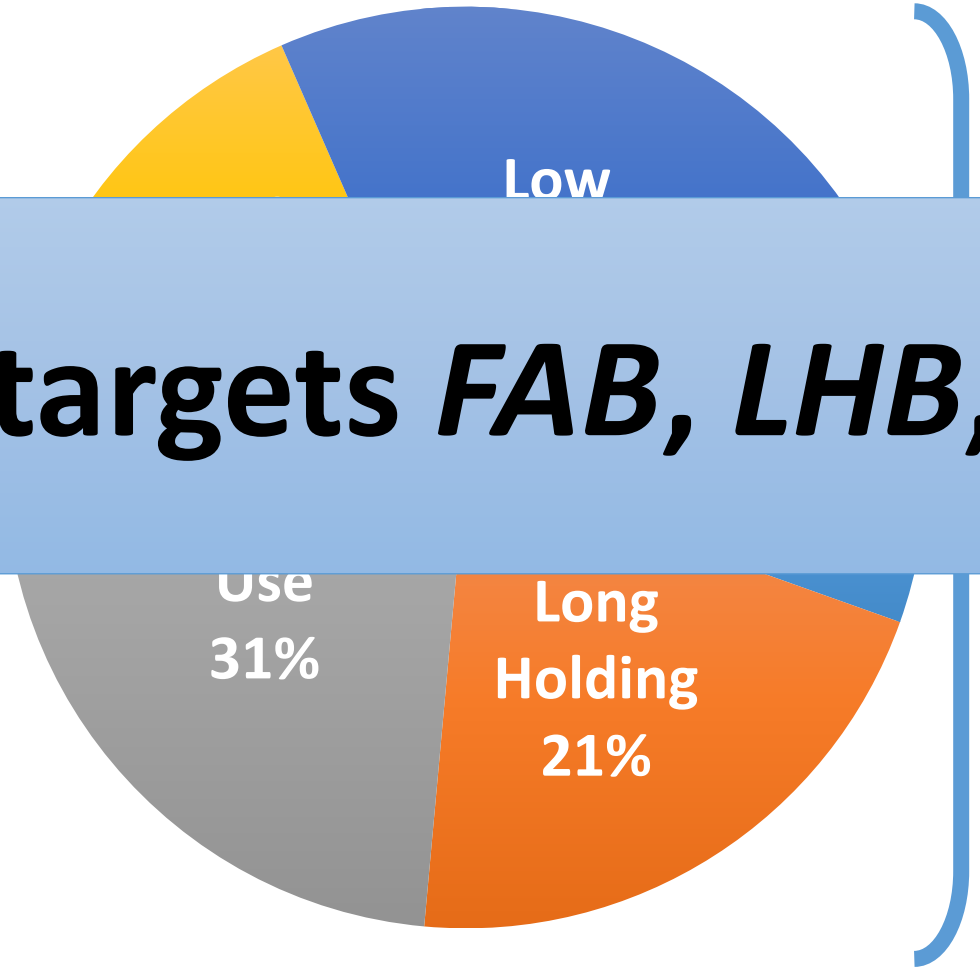
Bug	23%
Configuration	53%
Enhancement	15%
N/A	9%

FAB,LUB,LHB	
Bug	81%
Configuration	16%
Enhancement	3%

Prevalence of Each Misbehavior Type

- 109 real-world misbehavior cases

FAB: Frequent-Ask-Behavior
LHB: Long-Holding-Behavior
LUB: Low-Utility-Behavior
EUB: Excessive-Use-Behavior



LeaseOS targets *FAB*, *LHB*, and *LUB*

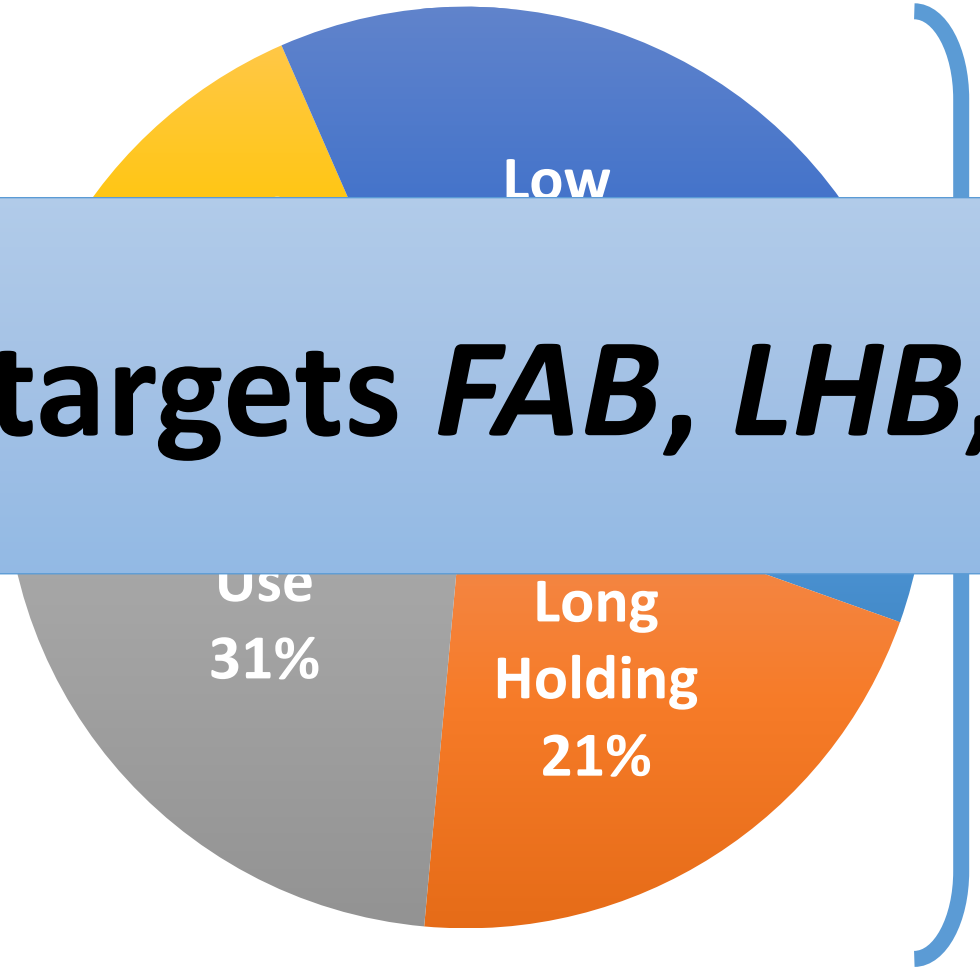
Bug	23%
Configuration	53%
Enhancement	15%
N/A	9%

FAB,LUB,LHB	
Bug	81%
Configuration	16%
Enhancement	3%

Prevalence of Each Misbehavior Type

- 109 real-world misbehavior cases

FAB: Frequent-Ask-Behavior
LHB: Long-Holding-Behavior
LUB: Low-Utility-Behavior
EUB: Excessive-Use-Behavior



LeaseOS targets *FAB*, *LHB*, and *LUB*

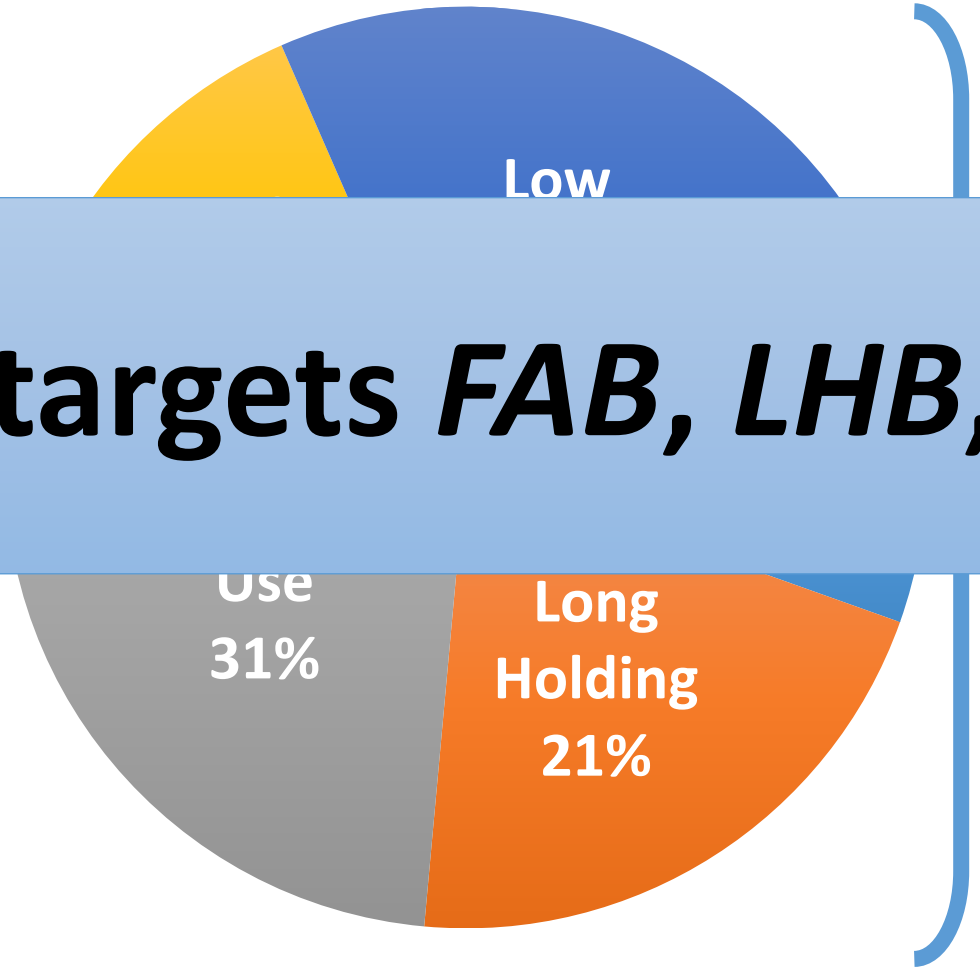
Bug	23%
Configuration	53%
Enhancement	15%
N/A	9%

FAB,LUB,LHB	
Bug	81%
Configuration	16%
Enhancement	3%

Prevalence of Each Misbehavior Type

- 109 real-world misbehavior cases

FAB: Frequent-Ask-Behavior
LHB: Long-Holding-Behavior
LUB: Low-Utility-Behavior
EUB: Excessive-Use-Behavior



LeaseOS targets *FAB*, *LHB*, and *LUB*

Bug	23%
Configuration	53%
Enhancement	15%
N/A	9%

FAB,LUB,LHB	
Bug	81%
Configuration	16%
Enhancement	3%

Metrics for All Type of Misbehavior

Frequent-Ask-Behavior

- success ratio : $\frac{unsuccessful\ request\ time}{total\ request\ time}$

Long-Holding-Behavior

- utilization ratio : $\frac{resource\ usage\ time}{holding\ time}$

Low-Utility-Behavior

- utility rate : $utility\ score\ \in [0,100]$

Generic Utility Metrics

Positive Event

- UI updates
- User interaction
- Moving distance

Negative Event

- Exception
- Stationary time
- Standby time
- Request Frequency

An Example of Customizable Utility Metric

TapAndTurn

Provide an icon to let user control screen rotation

Customizable Metric formula for orientation:

$$\frac{\textit{the times of user icon clicks}}{\textit{the times of icon appearance}}$$

```
UtilityCounter utility =
new UtilityCounter() {
    @Override
    public float getScore() {
        if (eventList == null || eventList.size() == 0)
            return 50;
        int click = 0;
        int rotation = 0;
        for (OrientationButtonOverlay.Event e : eventList) {
        rotation++;
        if (e.click == true)
            click++;
        }
      return 100.0 * click / rotation;
    }
};
```

An Example of Customizable Utility Metric

TapAndTurn

Provide an icon to let user control screen rotation

Customizable Metric formula for orientation:

$$\frac{\textit{the times of user icon clicks}}{\textit{the times of icon appearance}}$$

```
UtilityCounter utility =
new UtilityCounter() {
    @Override
    public float getScore() {
        if (eventList == null || eventList.size() == 0)
            return 50;
        int click = 0;
        int rotation = 0;
        for (OrientationButtonOverlay.Event e : eventList) {
        rotation++;
        if (e.click == true)
            click++;
        }
      return 100.0 * click / rotation;
    }
};
```

An Example of Customizable Utility Metric

TapAndTurn

Provide an icon to let user control screen rotation

Customizable Metric formula for orientation:

$$\frac{\textit{the times of user icon clicks}}{\textit{the times of icon appearance}}$$

```
UtilityCounter utility =
new UtilityCounter() {
    @Override
    public float getScore() {
        if (eventList == null || eventList.size() == 0)
            return 50;
        int click = 0;
        int rotation = 0;
        for (OrientationButtonOverlay.Event e : eventList) {
        rotation++;
        if (e.click == true)
            click++;
        }
      return 100.0 * click / rotation;
    }
};
```

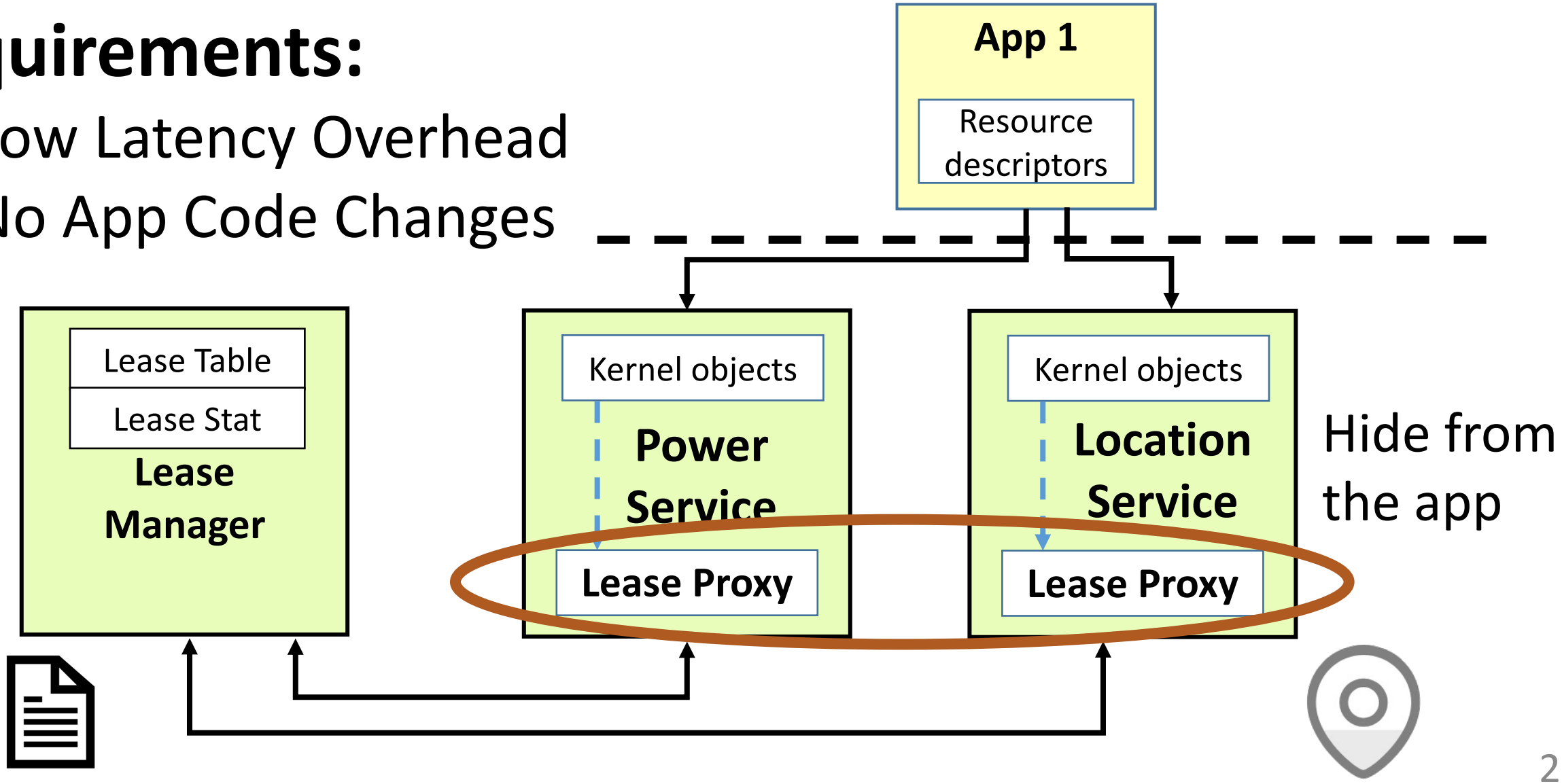
Outline

- Motivation
- Lease Abstraction
- Making Lease Decision
- **Design of LeaseOS**
- **Evaluation**

App-Oblivious Lease Management

Requirements:

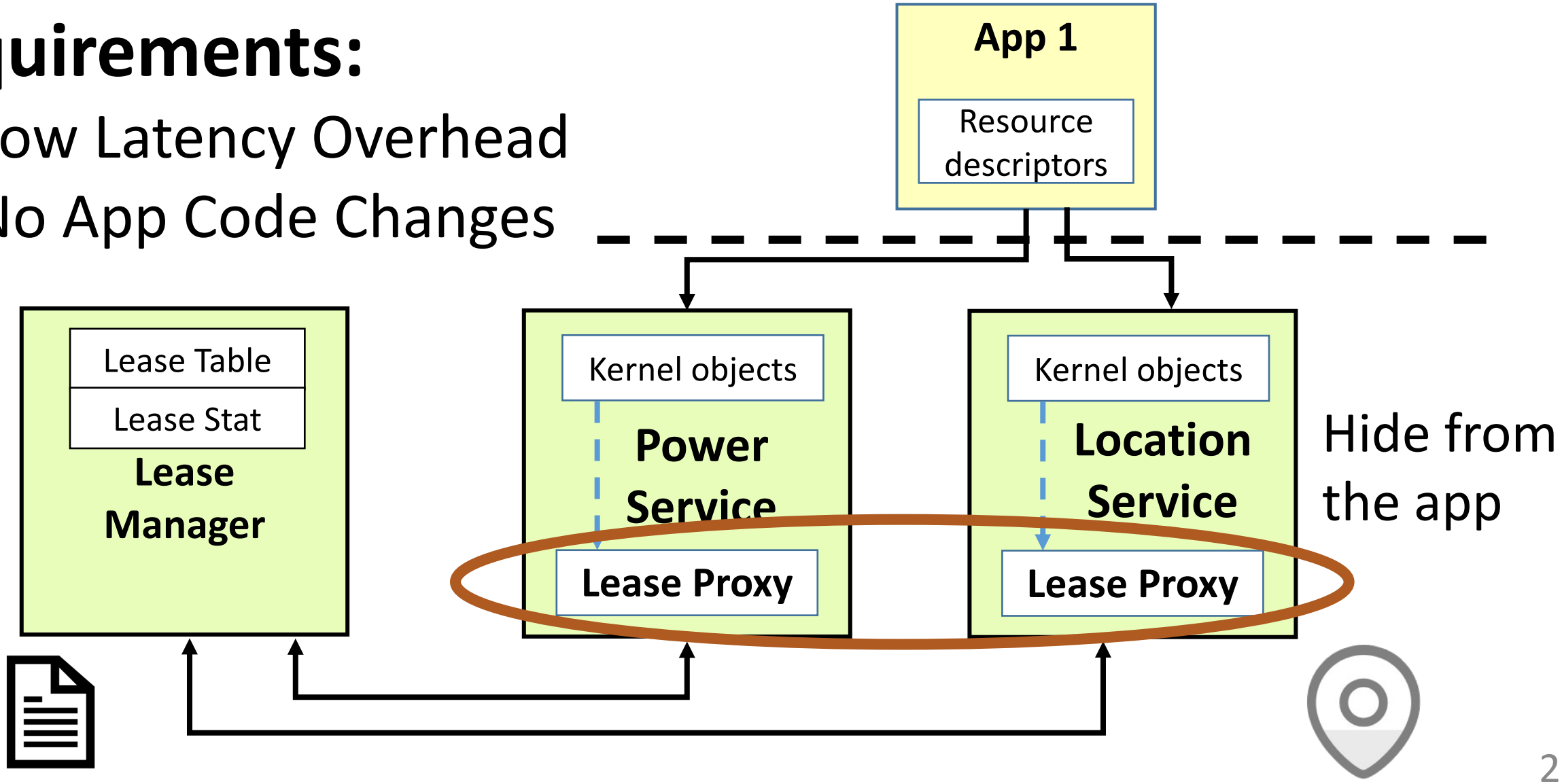
- Low Latency Overhead
- No App Code Changes



App-Oblivious Lease Management

Requirements:

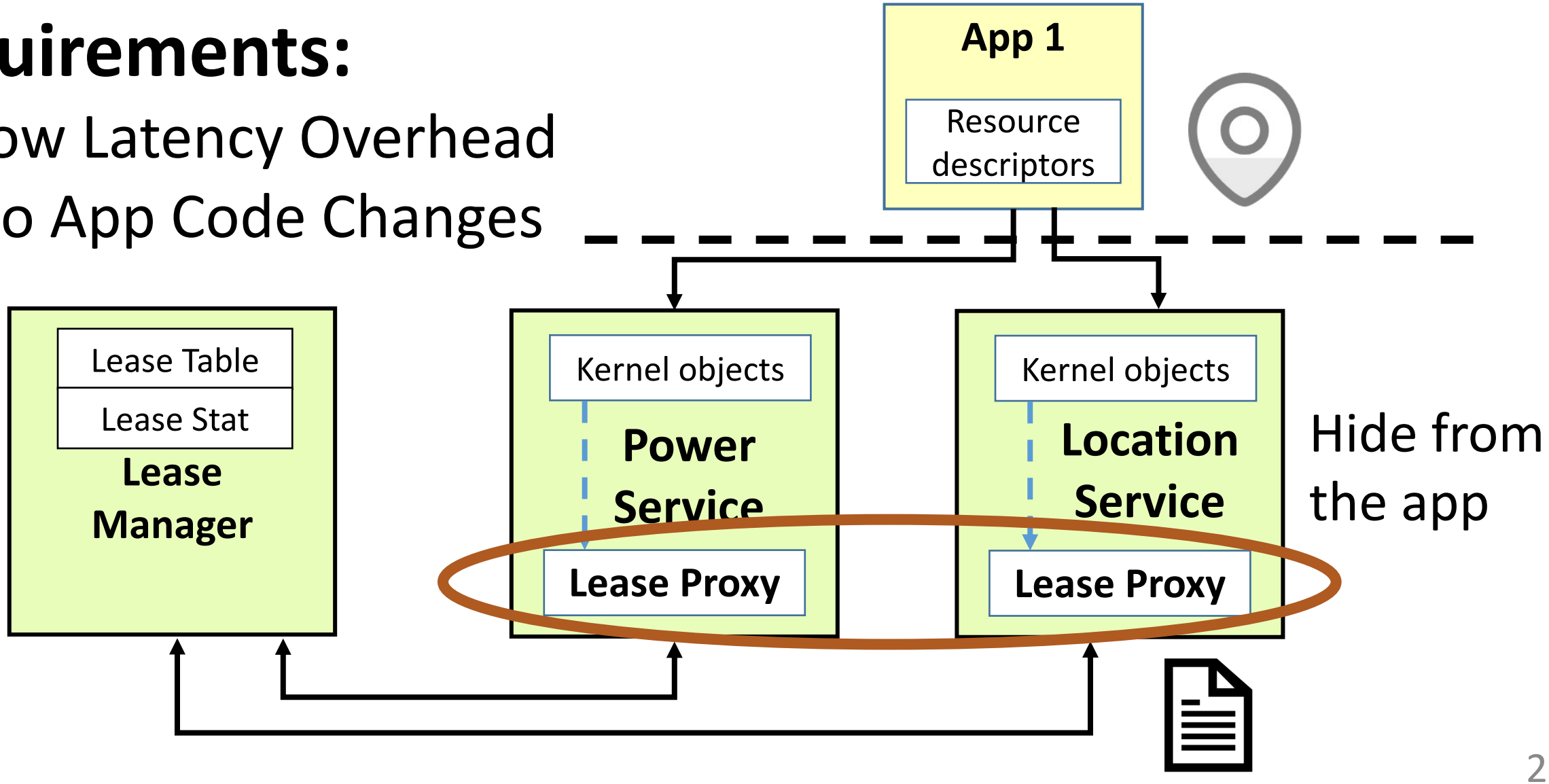
- Low Latency Overhead
- No App Code Changes



App-Oblivious Lease Management

Requirements:

- Low Latency Overhead
- No App Code Changes



LeaseOS Implementation

Build on Android 7.1.2

- with 9,100 lines of code
- Android framework

Implement on 7 types of resource

- Wakelock, GPS, Sensor, WI-FI and etc.

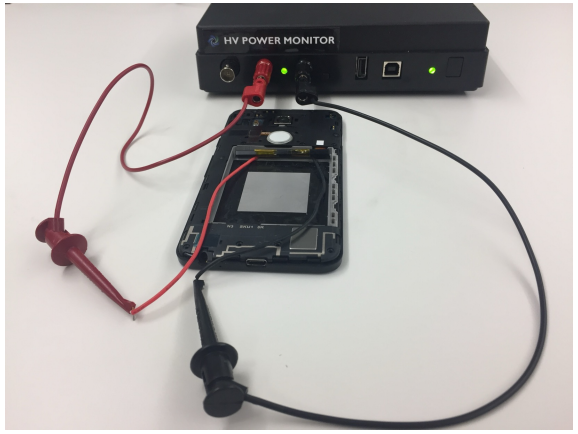
Outline

- Motivation
- Lease Abstraction
- Making Lease Decision
- Design of LeaseOS
- **Evaluation**

Evaluation

- **How effective can LeaseOS mitigate energy misbehavior?**
- **What is the overhead of LeaseOS?**
- **What is the impact to app usability?**

Experiment Setup



Devices

- Google Pixel XL, Nexus 5X

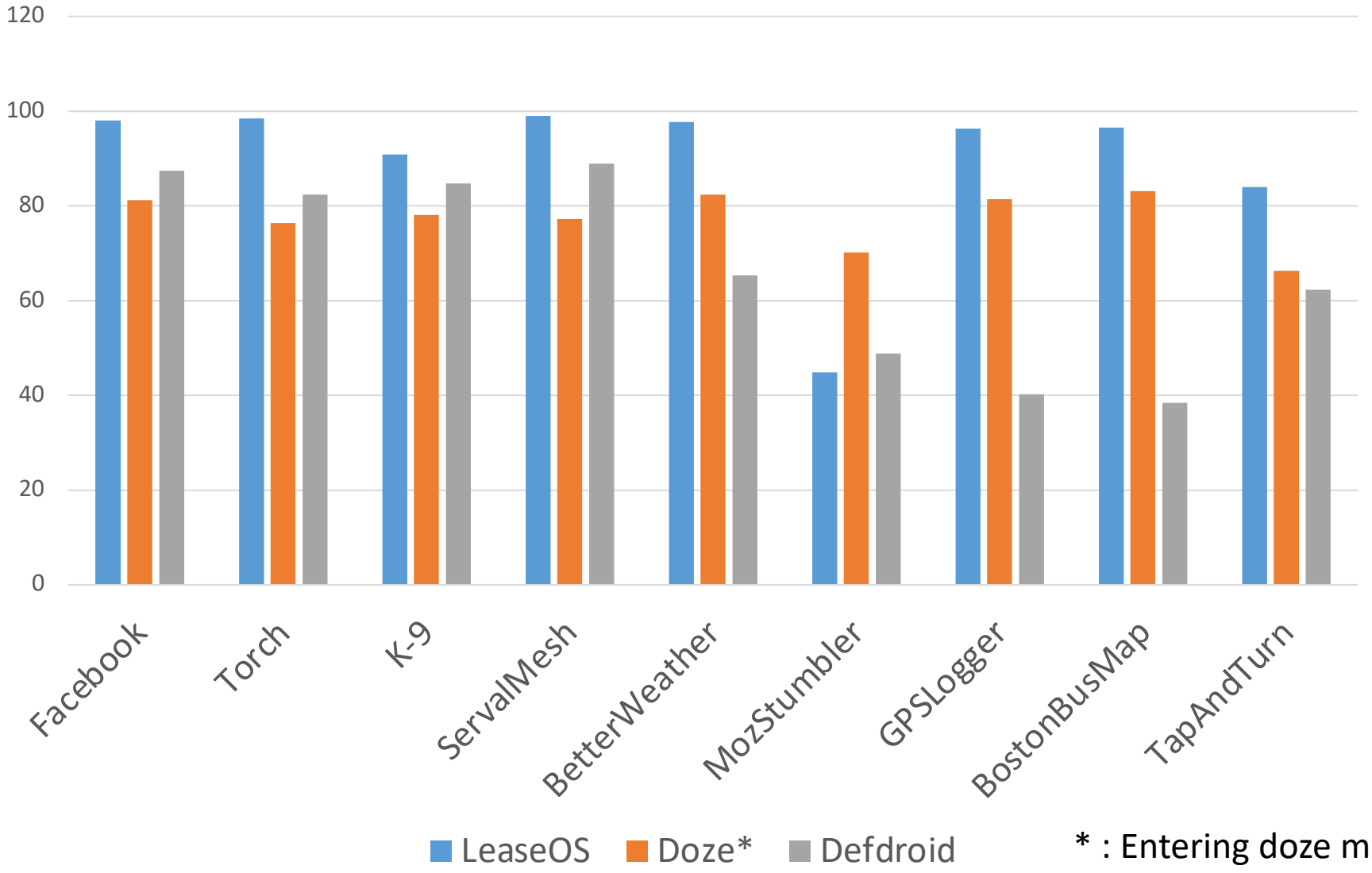
Data collector

- Android build-in tool, Trepn, Monsoon Power Monitor

Dataset

- Reproduce 20 real-world energy misbehavior cases

Energy Waste Reduction



LeaseOS: 92%

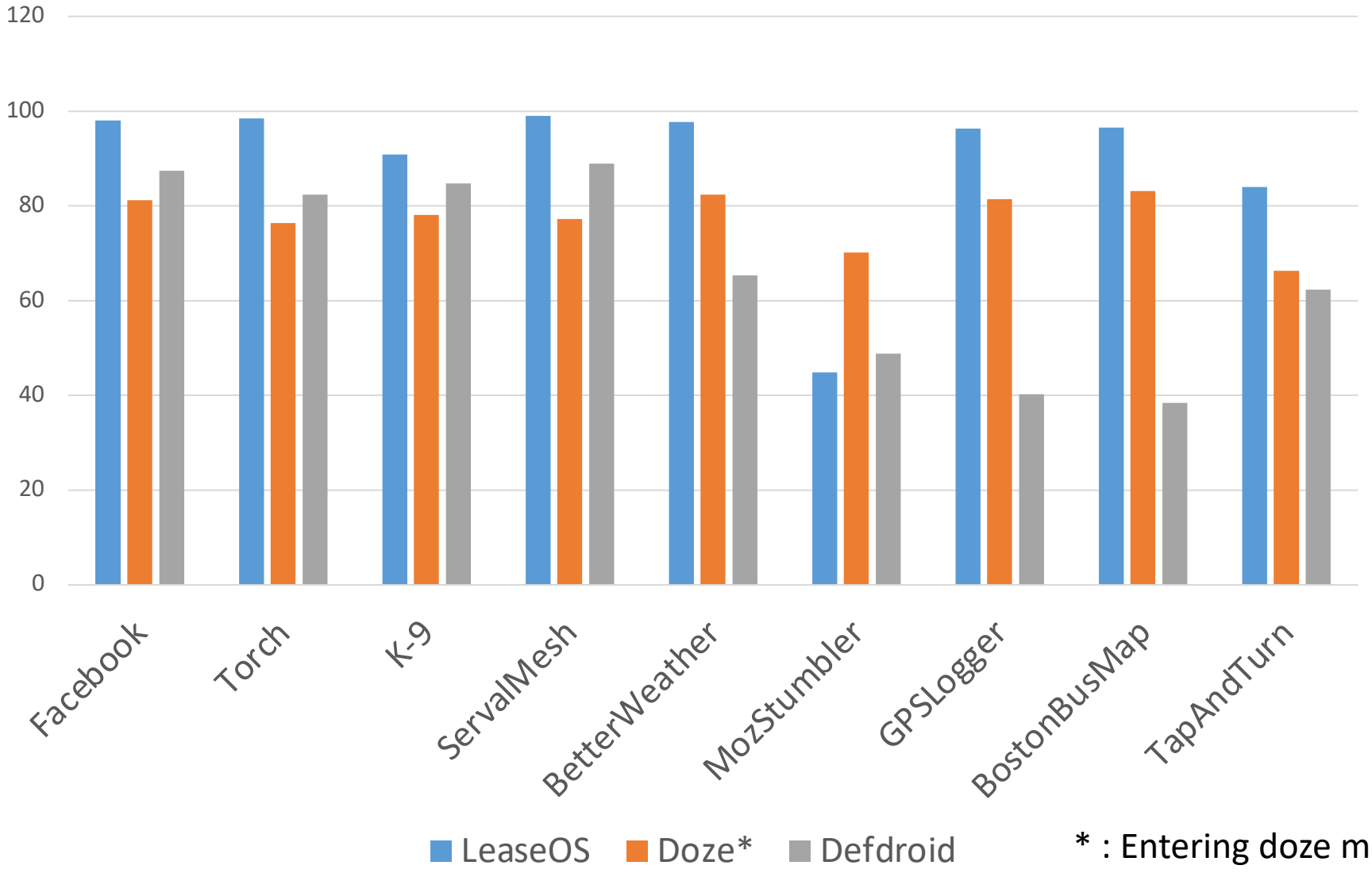
Doze: 70%

Defdroid:62%

...

* : Entering doze mode after 5 mins of each experiment

Energy Waste Reduction



LeaseOS: 92%

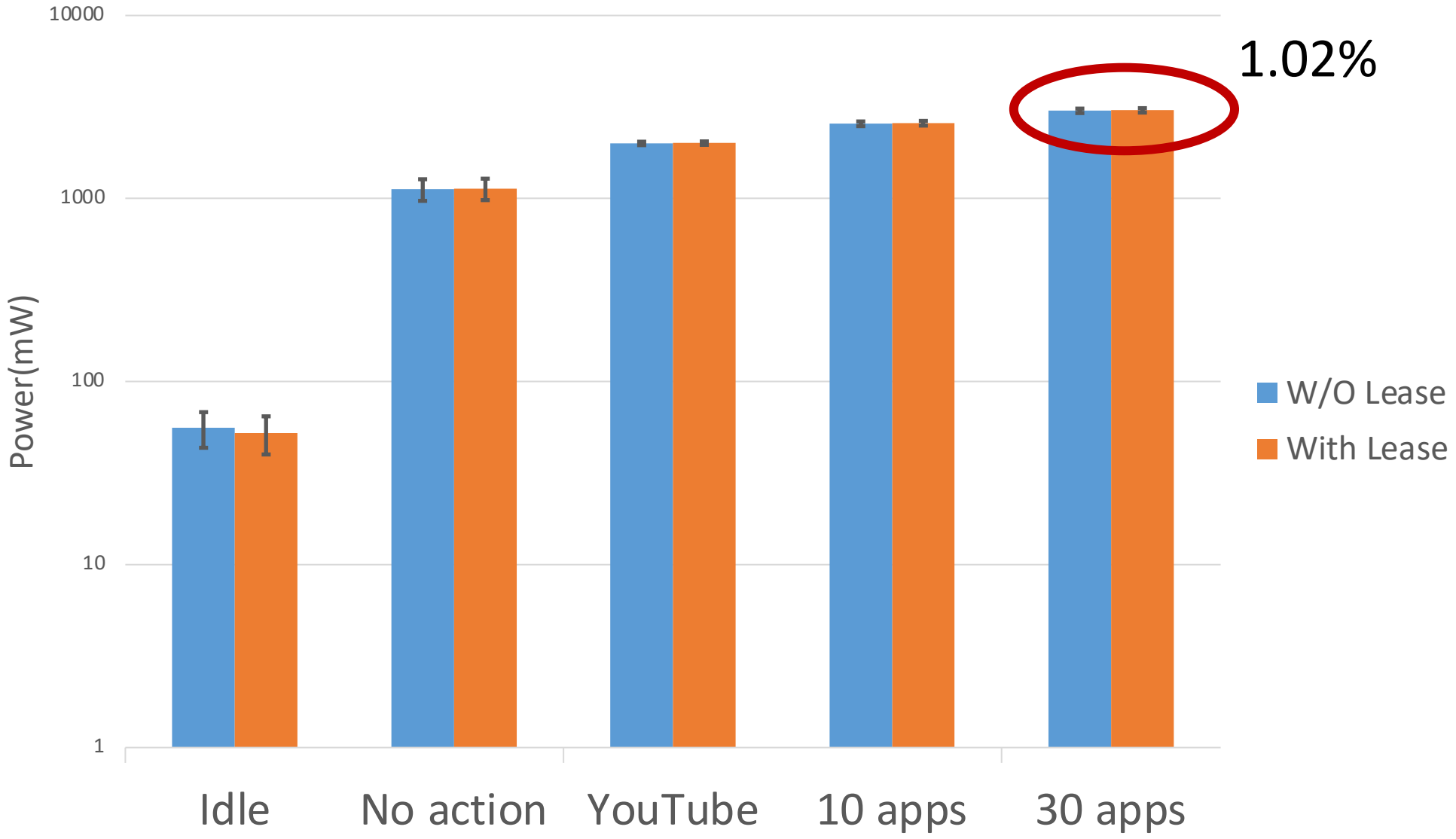
Doze: 70%

Defdroid:62%

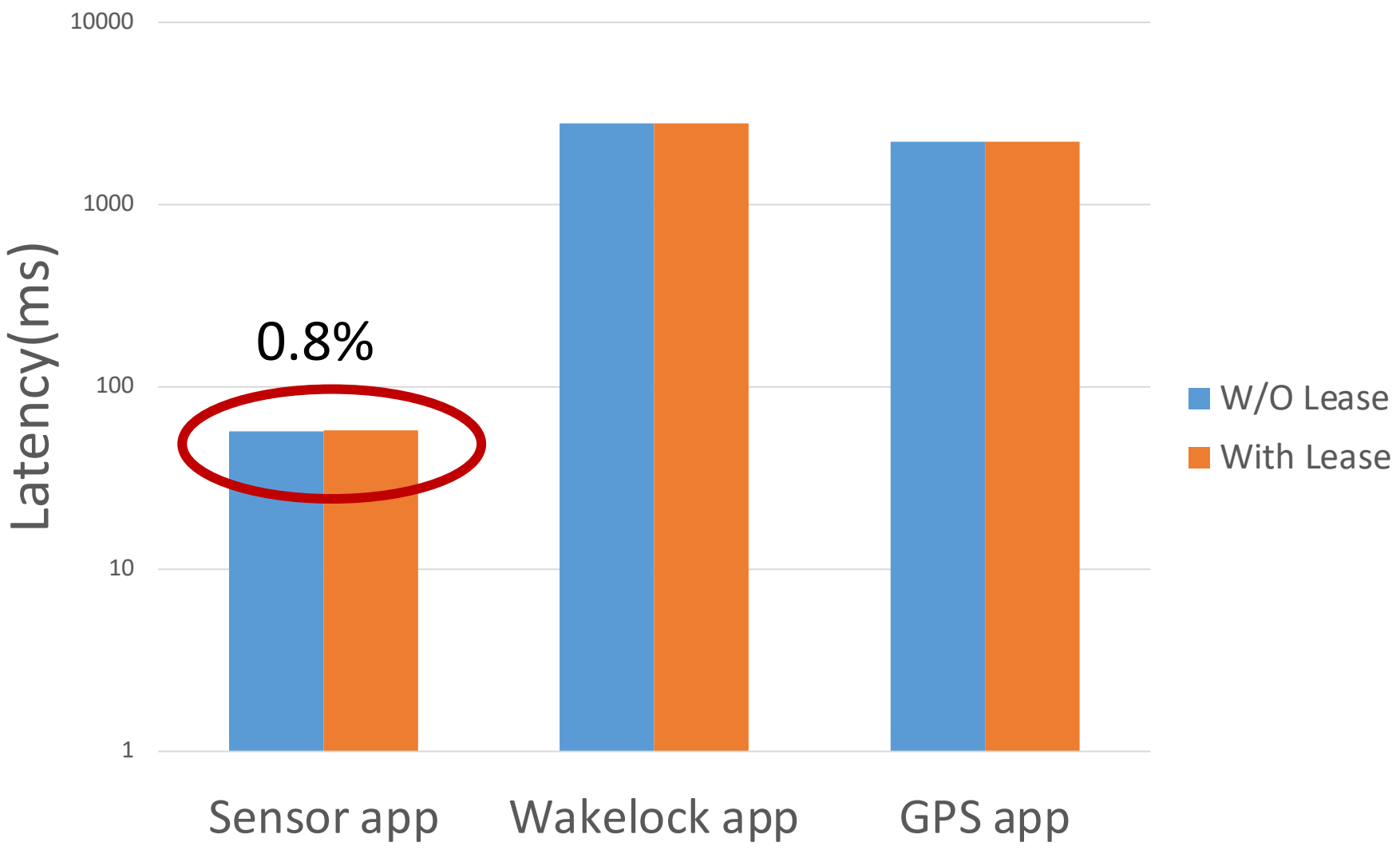
...

* : Entering doze mode after 5 mins of each experiment

System Energy Overhead



Resource Request Latency Overhead



Usability Impact Experiment

App	Usability Impact	
	DefDroid	LeaseOS
Spotify (Music)	Stop playing music	Keep playing in the back ground
RunKeeper (fitness tracker)	Stop tracking location and movement	Keep tracking in the background
Haven (intruder monitoring)	Stop monitoring	Keep monitoring

Related Works

- OS Support
  - ECOSystem [ASPLOS'02], Ghost Hints [Mobisys'04], CinderOS [EuroSys'11], JouleGuard [SOSP'15]
- Energy-Aware Adaption
  - Odyseey [SOSP'99], GraceOS [SOSP'03], SPECTR [ASPLOS'18], CALOREE [ASPLOS'18]
- Runtime Mitigation
  - DefDroid [Mobisys'16]
- Energy Bug Detection
  - No-sleep energy bug [MobiSys'12], eDoctor [NSDI'13], AppDoctor [EuroSys'14], Nchecker [EuroSys'16]

LeaseOS's main difference: a new abstraction to reduce energy waste caused by non-cooperative misbehaving apps

Conclusion

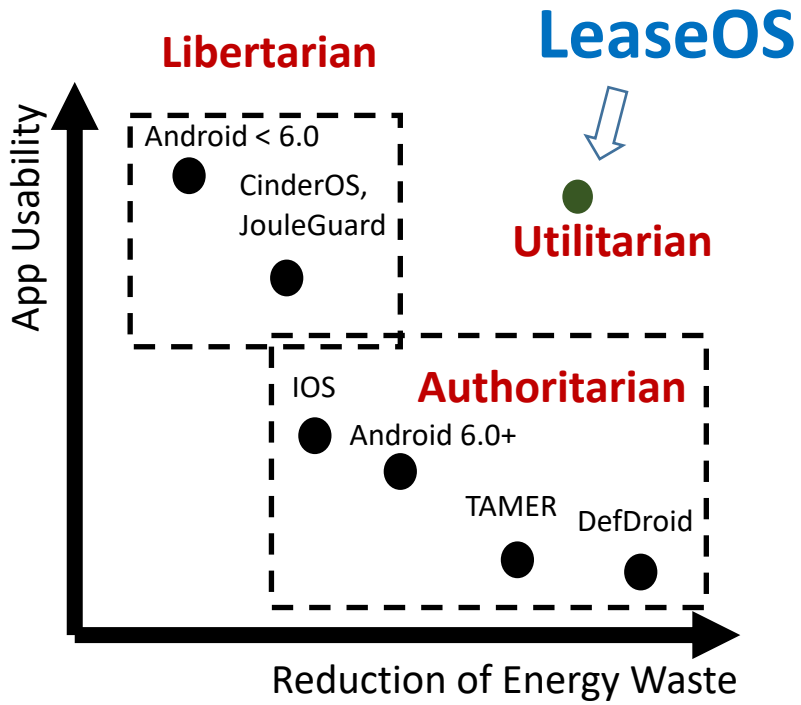
1. Lease is a well-suited abstraction for managing resource in mobile system

2. Explore utilitarian approach for making lease decisions

3. Design of LeaseOS – a lightweight runtime resource management

4. Reduce 92% energy waste on 20 apps

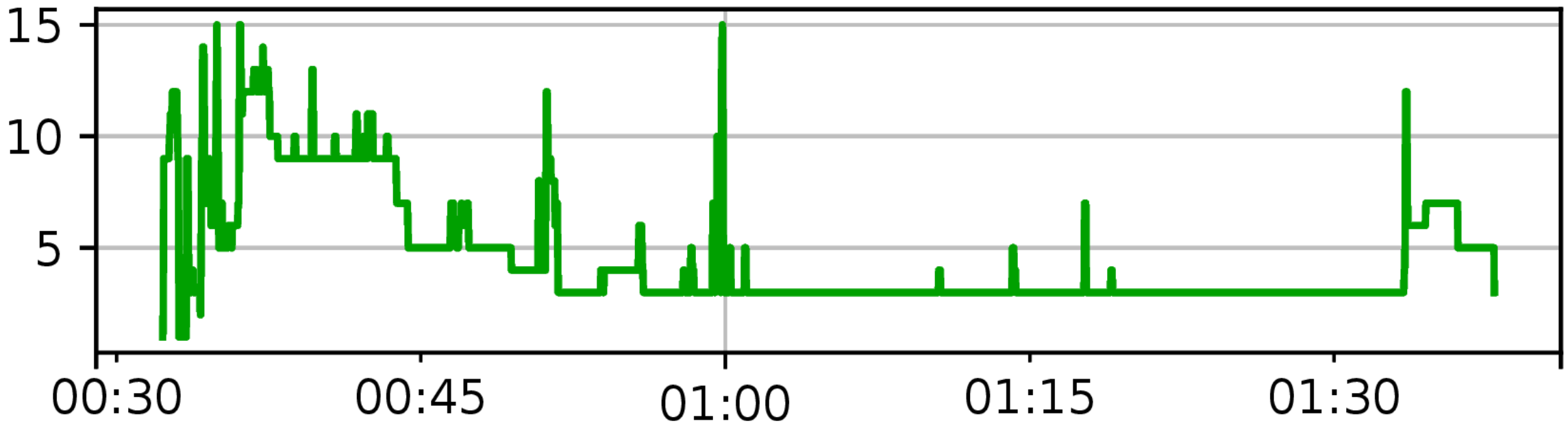
https://orderlab.io/LeaseOS/



Questions

- How hard is it for developers to implement the customizable utility interface?
- What if the developer abuse the customizable interface?
- What is the sensitivity to the threshold settings?
- How do we handle DVFS?

Number of Active Lease



Average Latency of Major Lease Operation

Create	Check (Acc)	Check (Rej)	Update
0.357	0.498	0.388	4.79

Lease Policy

Impact of lease term and delay interval:

$$r = \frac{1}{1+\lambda}, where\ \lambda = \frac{avg(\tau)}{lease\ term}$$