

Automated Software Testing to Discover Energy Inefficiencies in Mobile Apps

Sudipta Chattopadhyay
Linköping University, Sweden

Joint work with Abhijeet Banerjee, Lee Kee Chong and Abhik Roychoudhury
National University of Singapore

Context

Developer



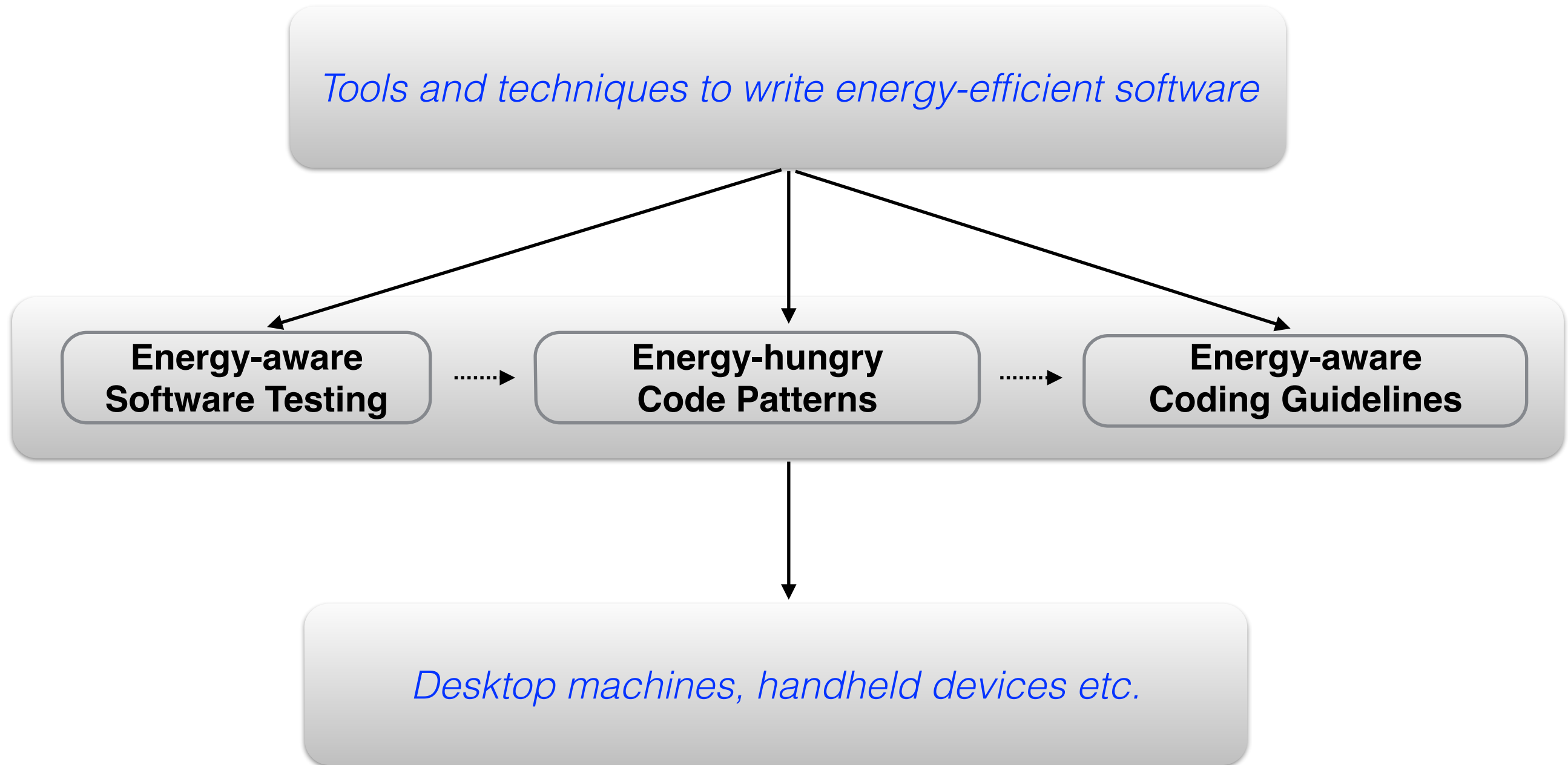
Write energy-efficient software

Programming abstractions

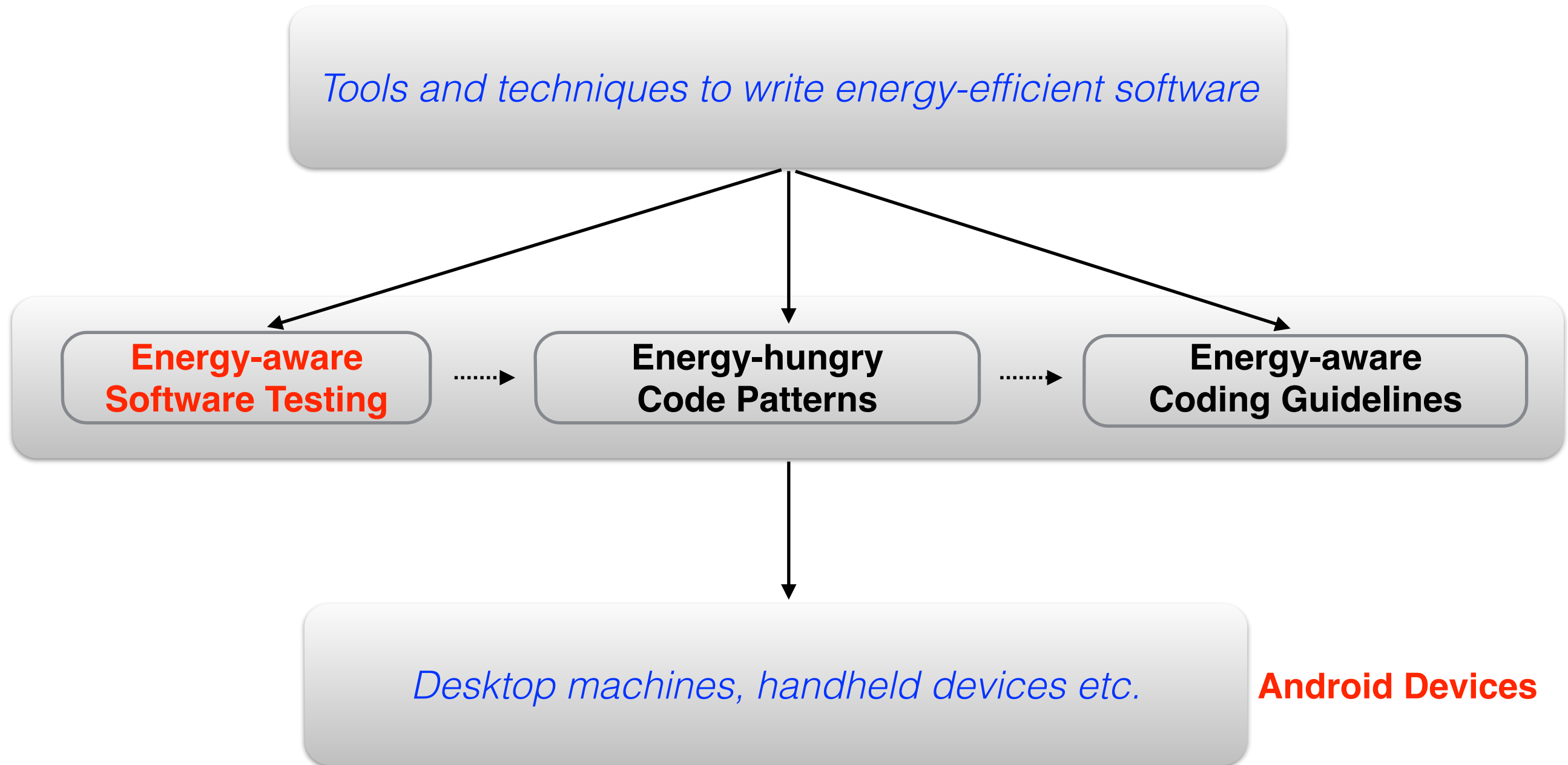
Tools and techniques

Desktops, handheld devices etc.

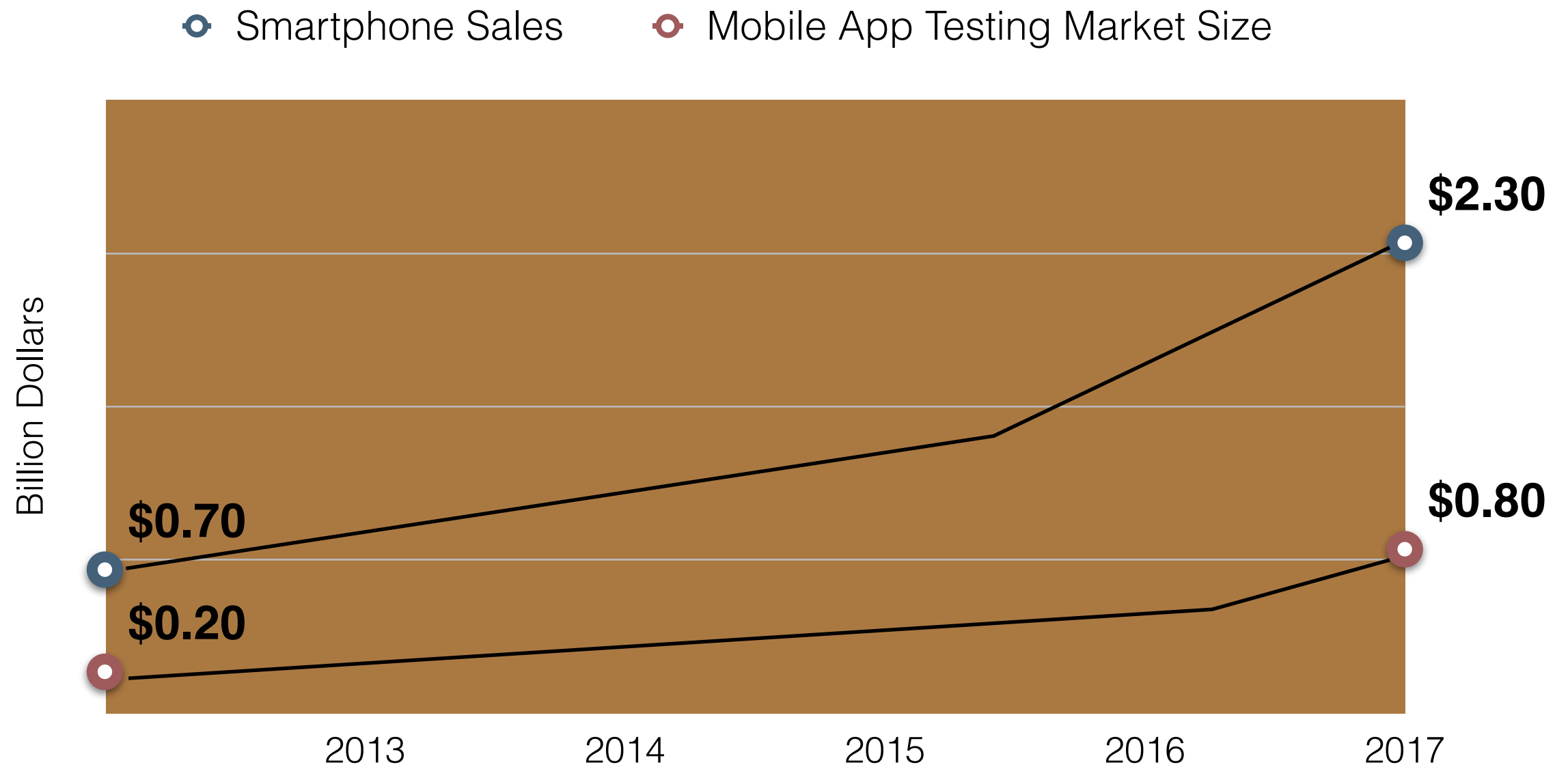
Overview



Overview



Smartphone Market



Data obtained from IDC, Gartner and ABI Research

Energy Inefficiency

- How do we quantify energy inefficiency?
 - High energy consumption, what is *high?*
- High energy consumption
 - High utilization of hardware components
 - *Low utilization of hardware components*
- Ratio Energy/Utilization

Energy Inefficiency

| Cause/Source | | |
|-------------------------|----------------------------|------------------------------|
| Hardware components | Resource leak | Suboptimal resource binding |
| Sleep state transition | Wakelock bug | Tail Energy hotspot |
| Background Service | Vacuous background service | Expensive background service |
| Defective Functionality | Immortality bug | Loop energy hotspot |

Energy Inefficiency



Suboptimal resource binding



Vacuous background service

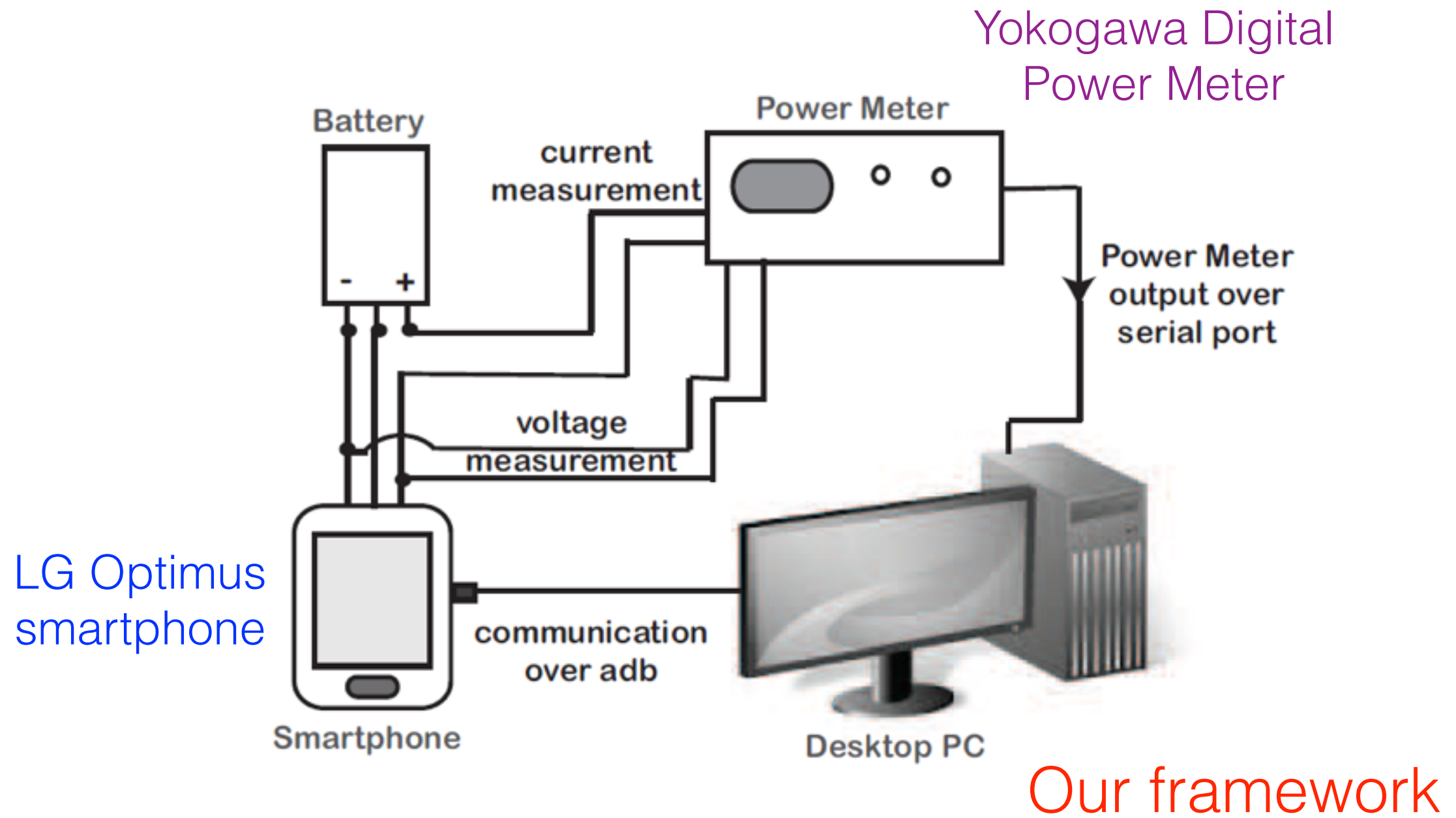
A Broader Categorization

| Cause/Source | Energy Bugs | Energy Hotspots |
|-------------------------|----------------------------|------------------------------|
| Hardware components | Resource leak | Suboptimal resource binding |
| Sleep state transition | Wakelock bug | Tail Energy hotspot |
| Background Service | Vacuous background service | Expensive background service |
| Defective Functionality | Immortality bug | Loop energy hotspot |

*Device does
not return to idle*

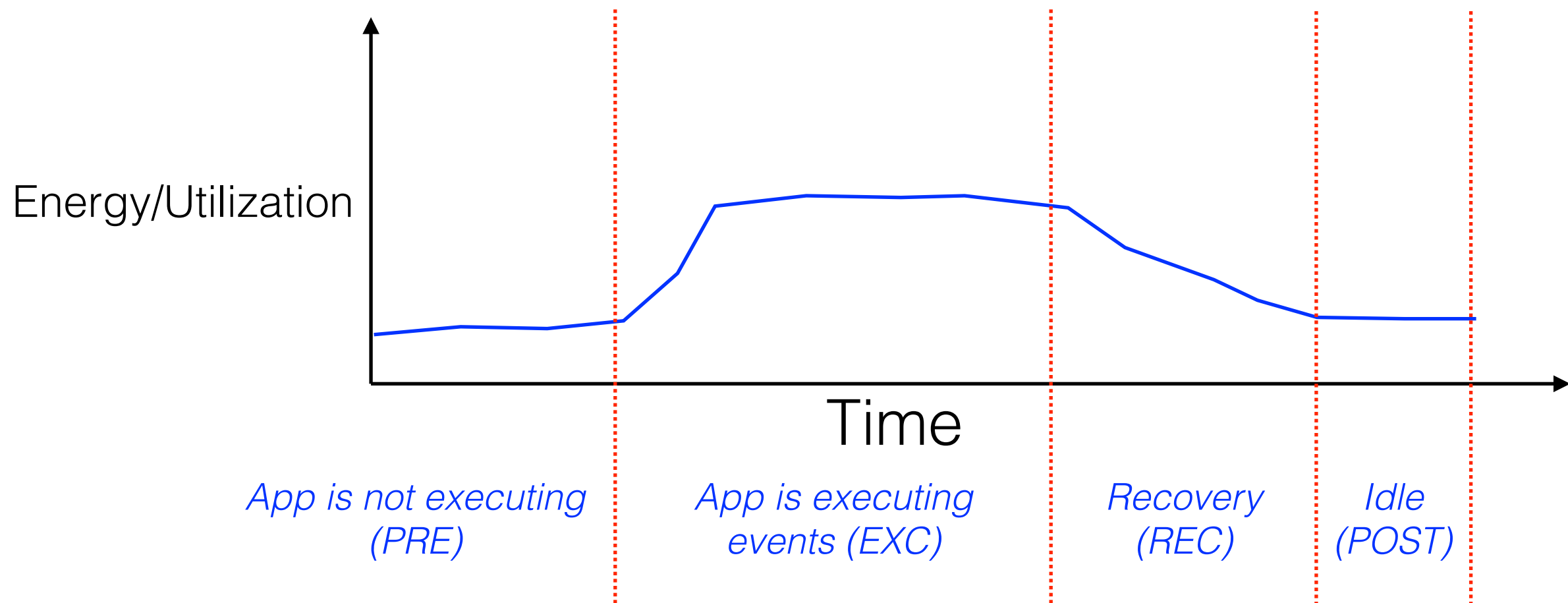
*High energy consumption
+ low utilization*

Measurement

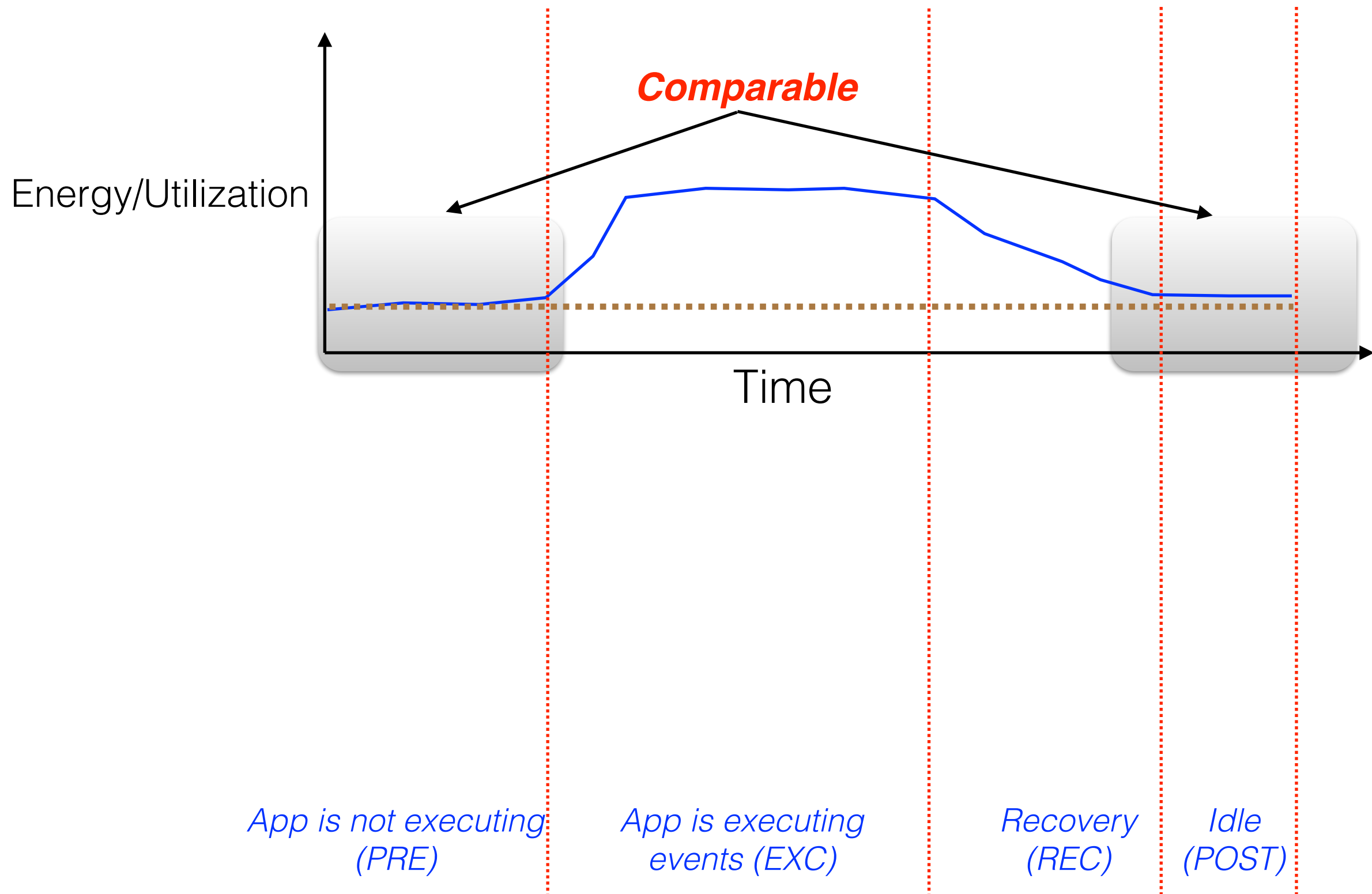


Measurement

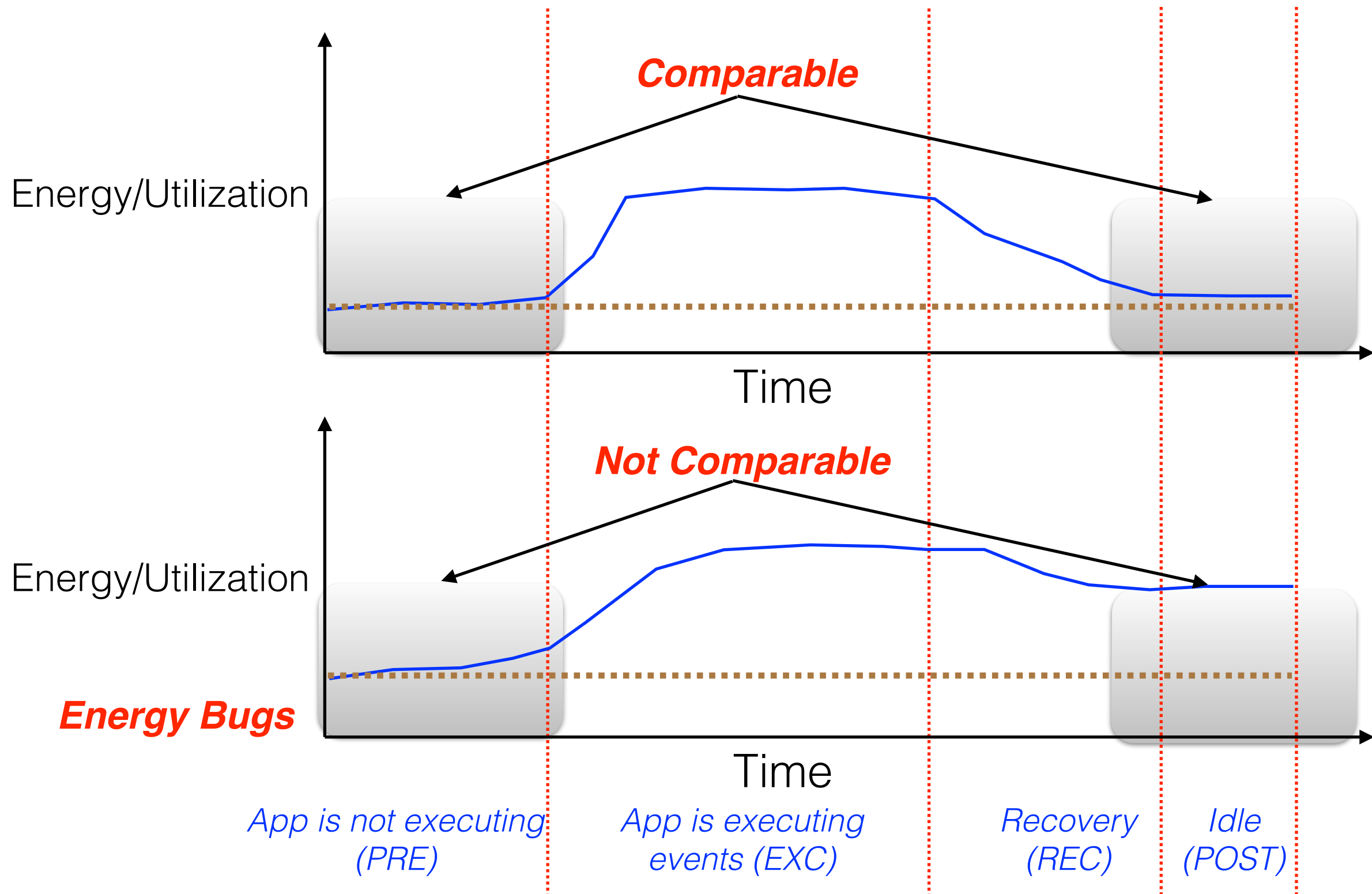
- Measuring Energy/Utilization ratio for an application



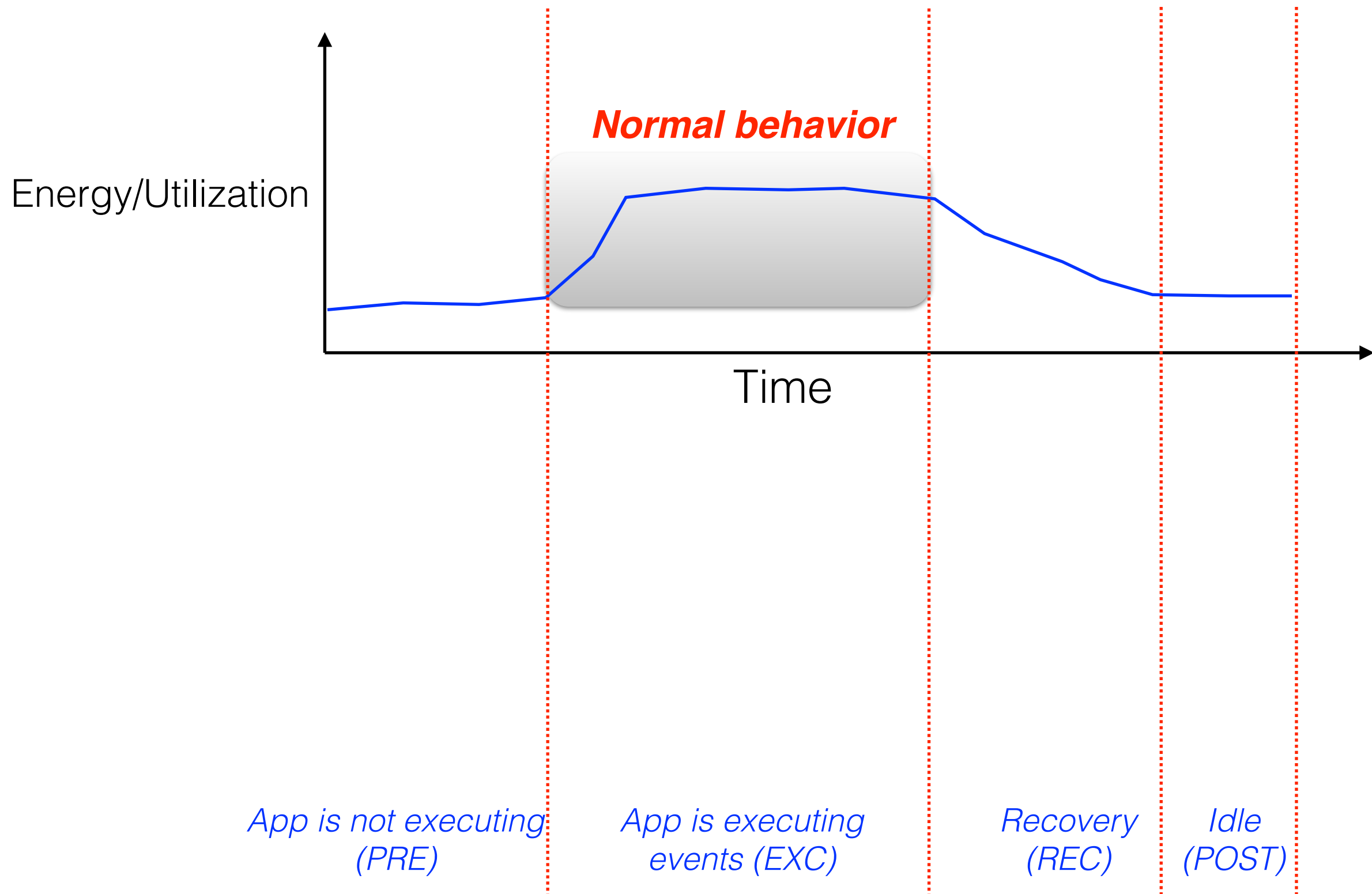
Energy Inefficiency



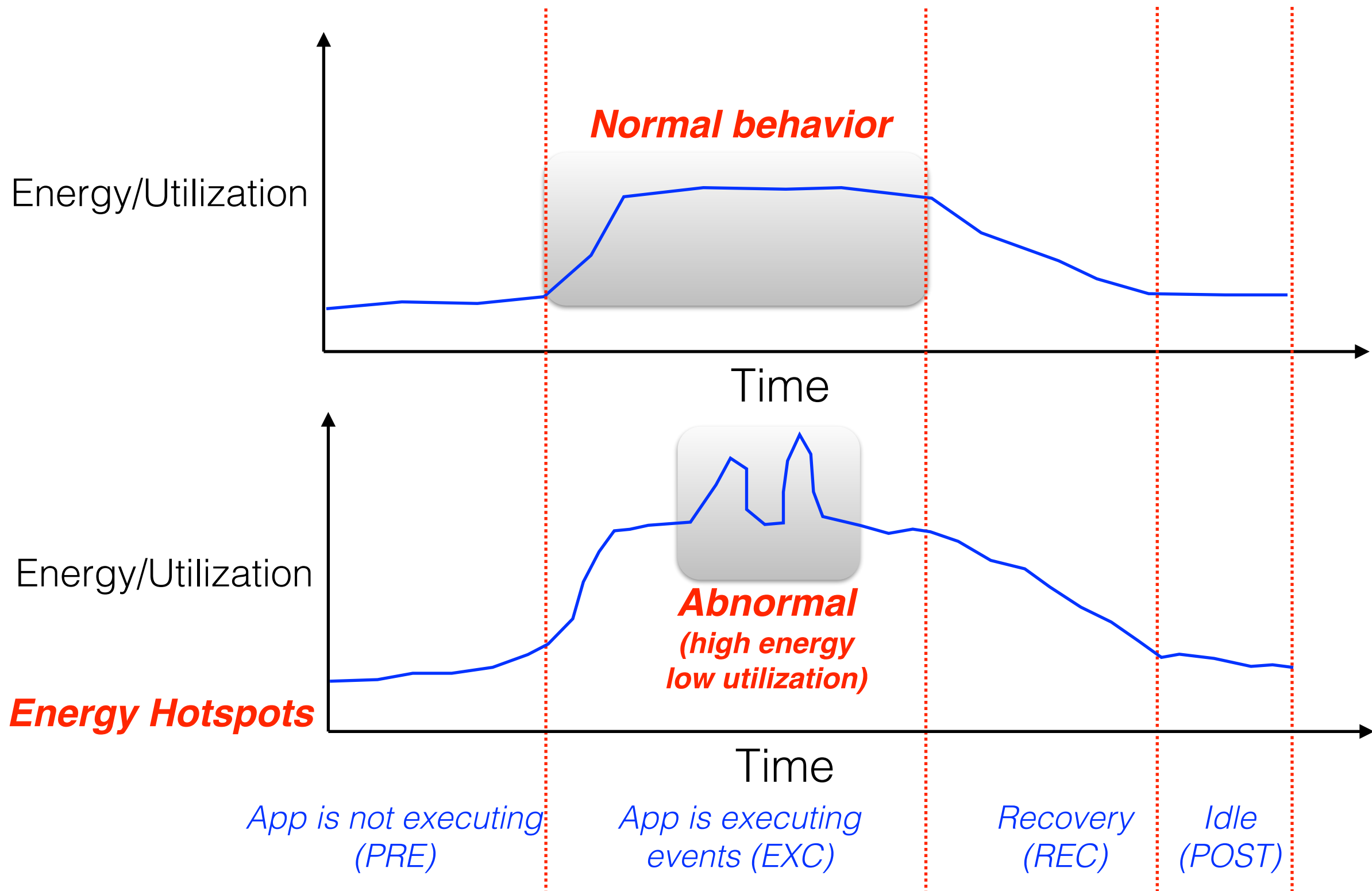
Energy Inefficiency



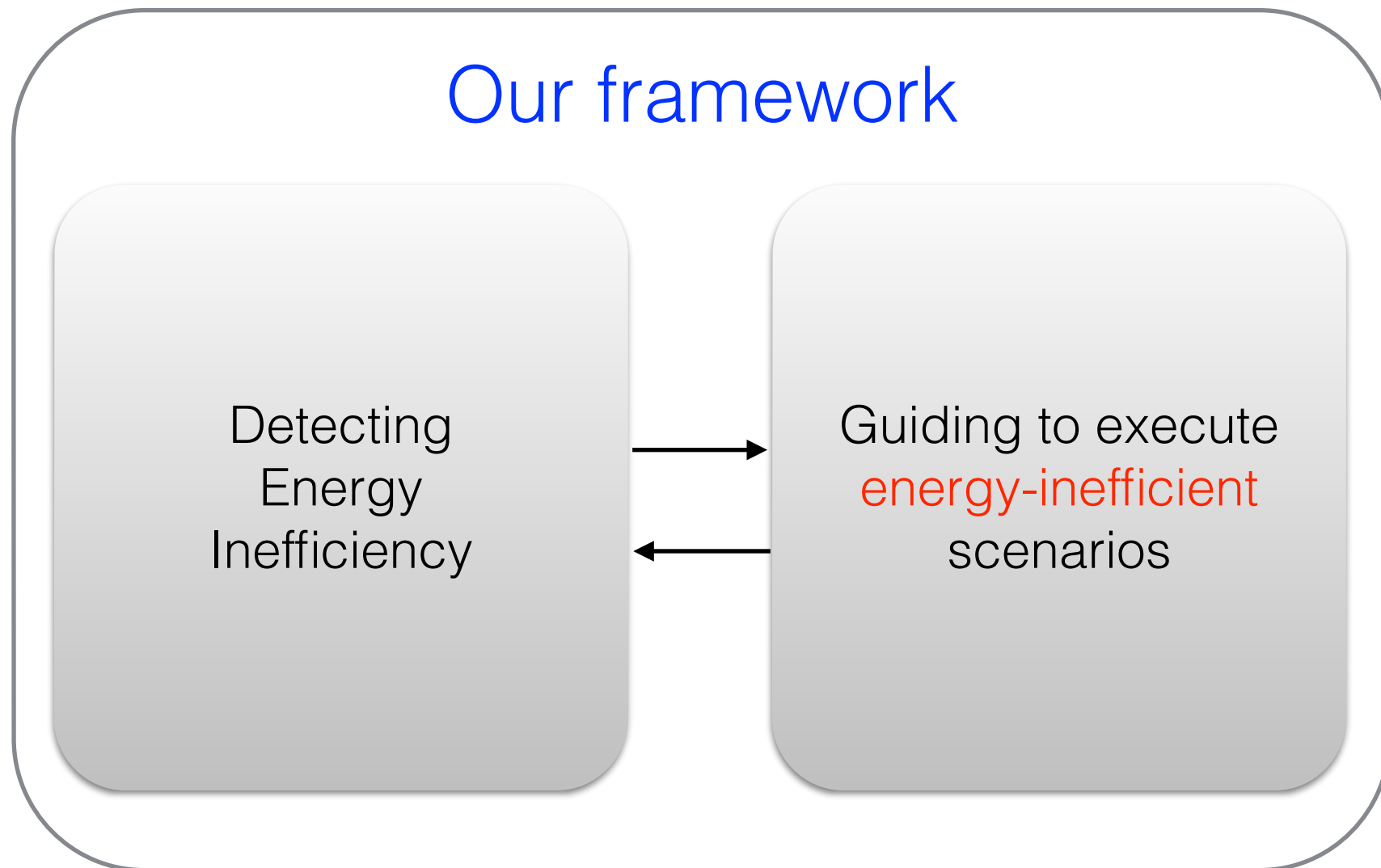
Energy Inefficiency



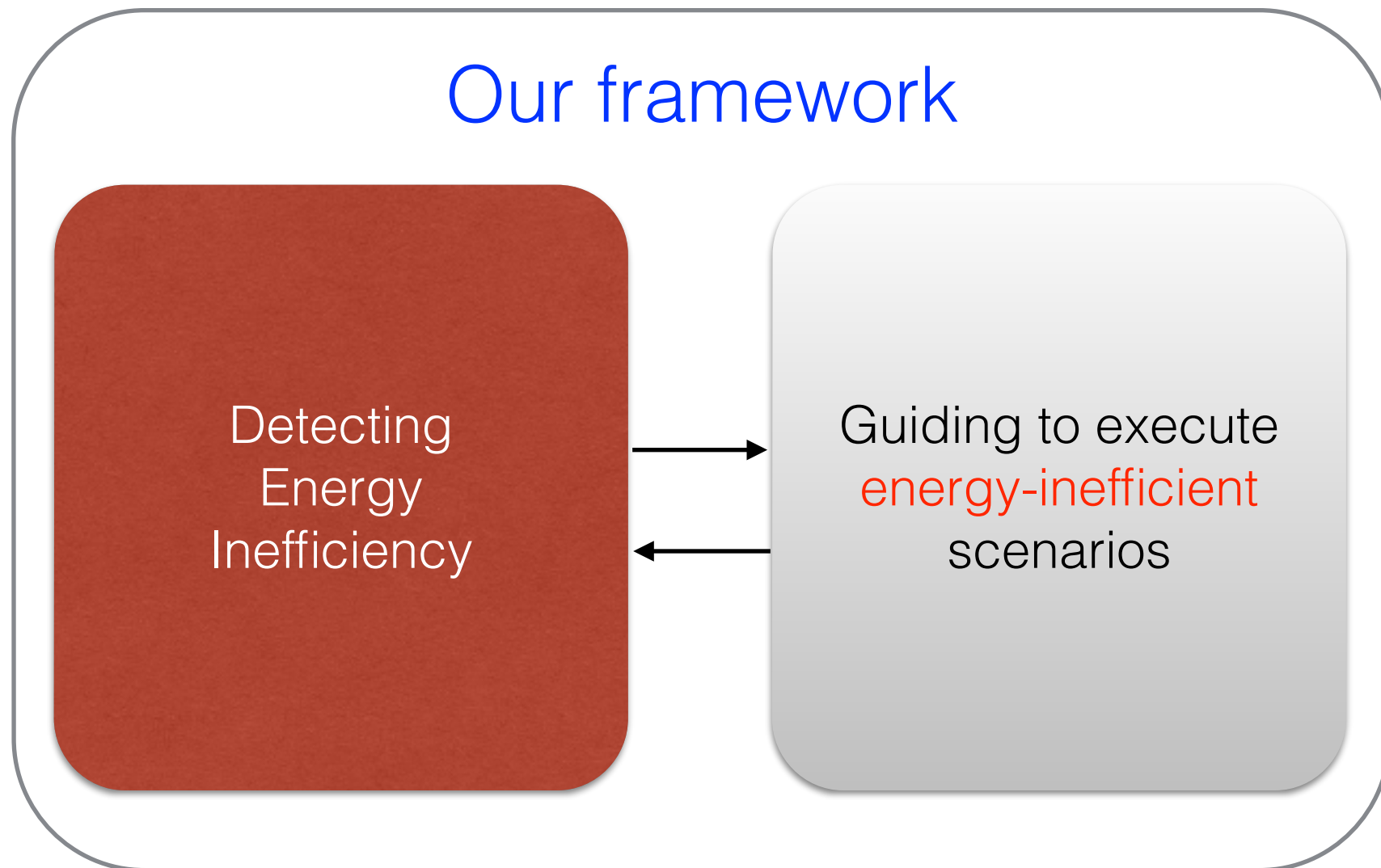
Energy Inefficiency



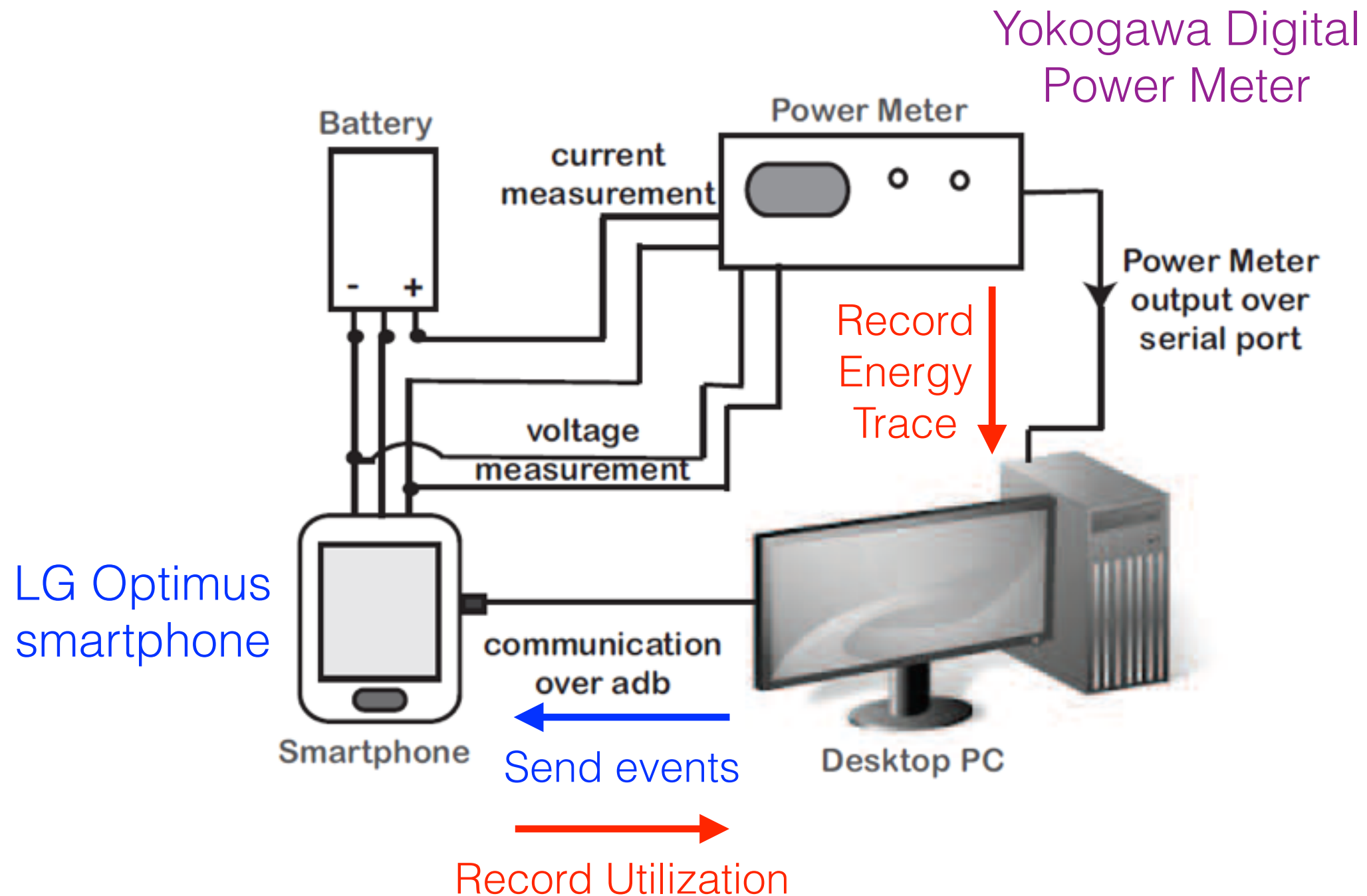
Test Generation



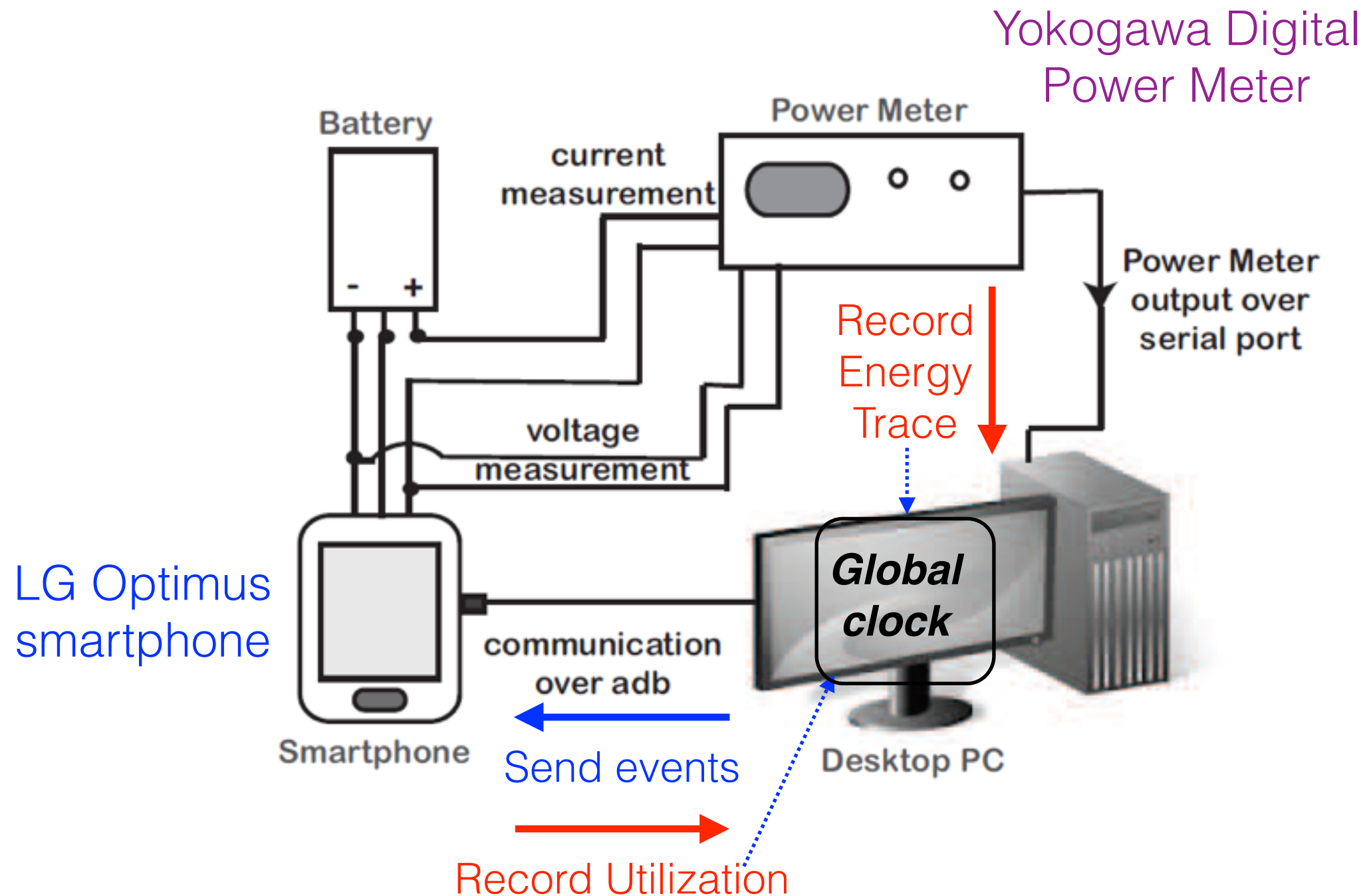
Test Generation



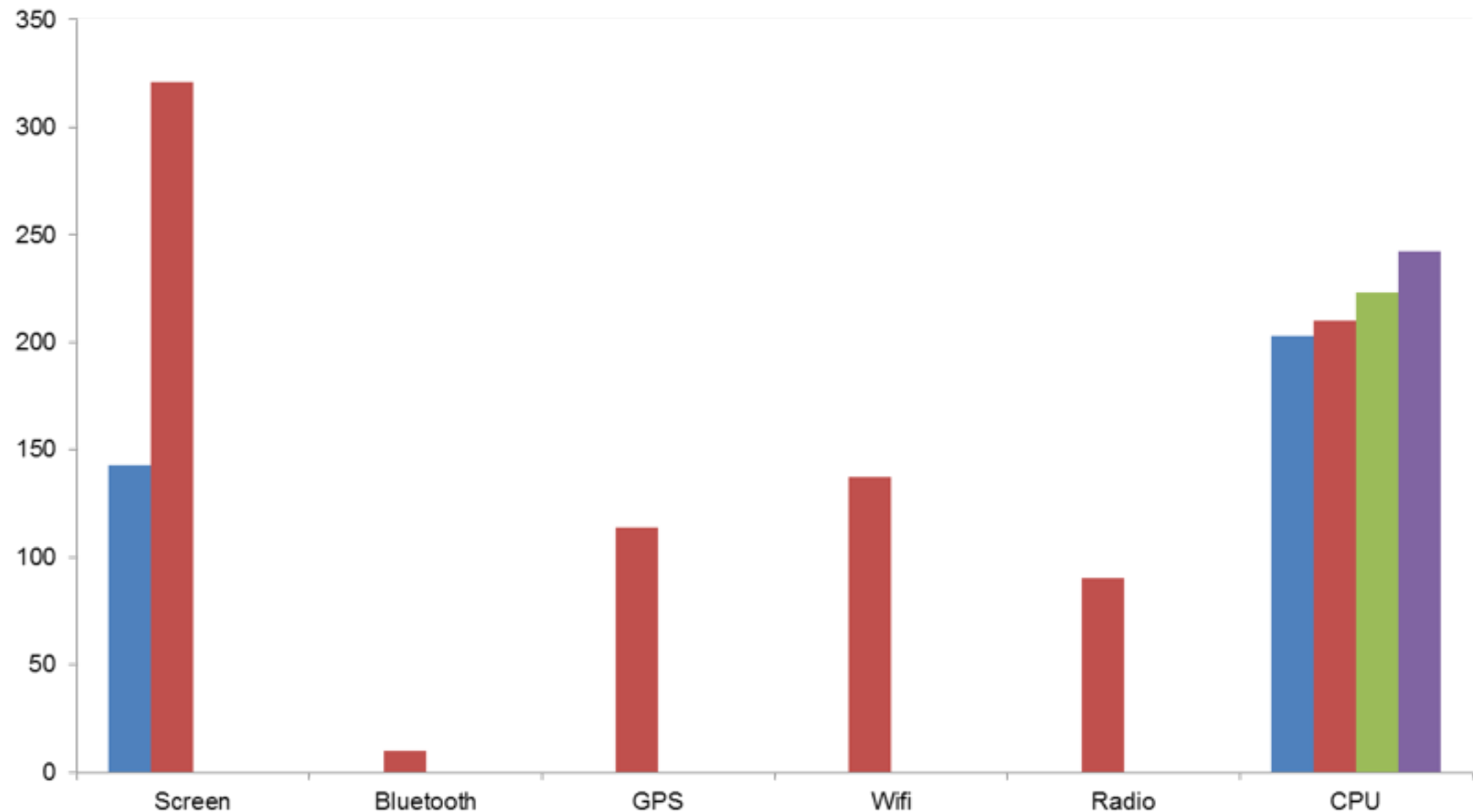
Measurement



Measurement

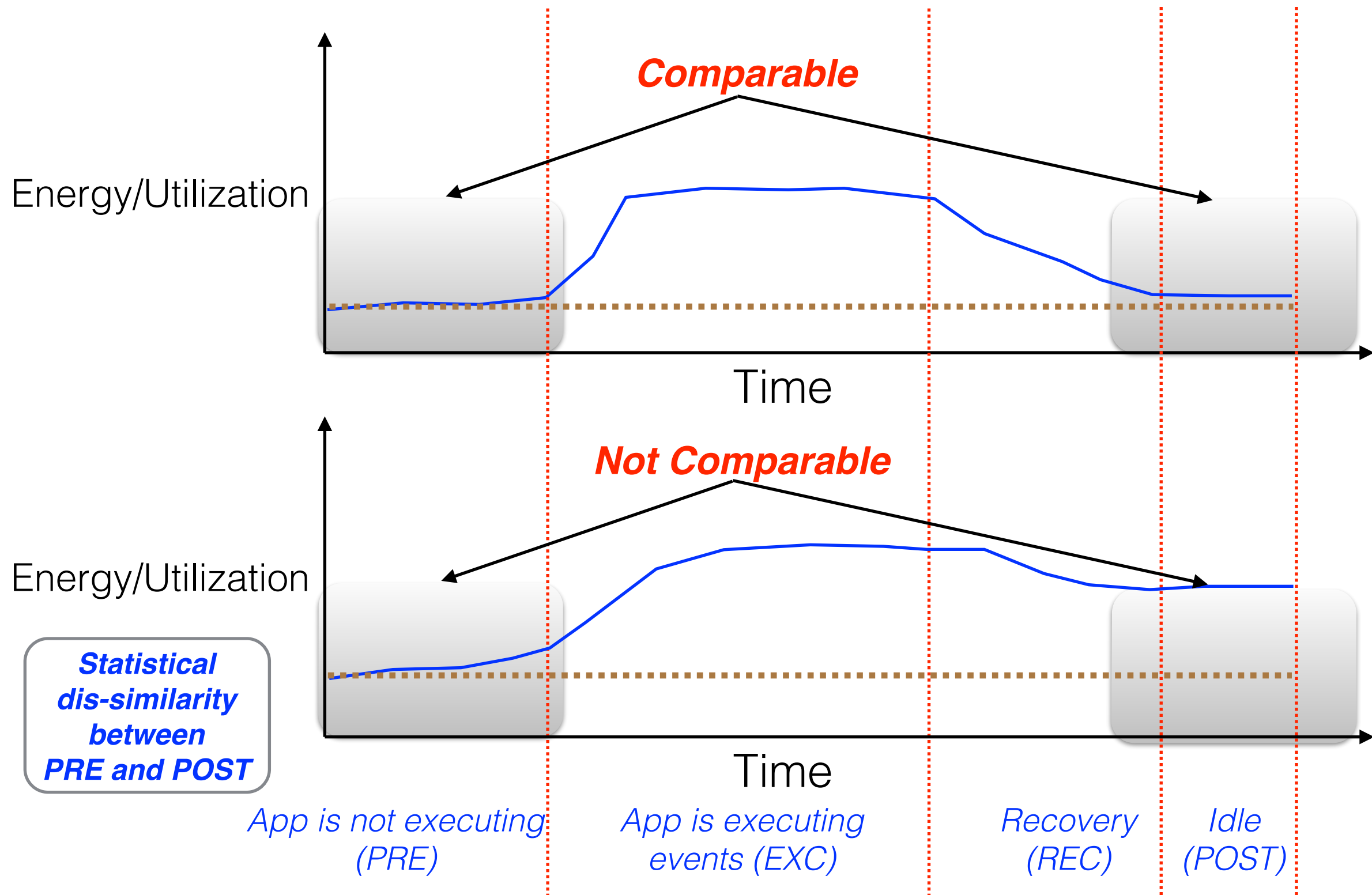


Utilization

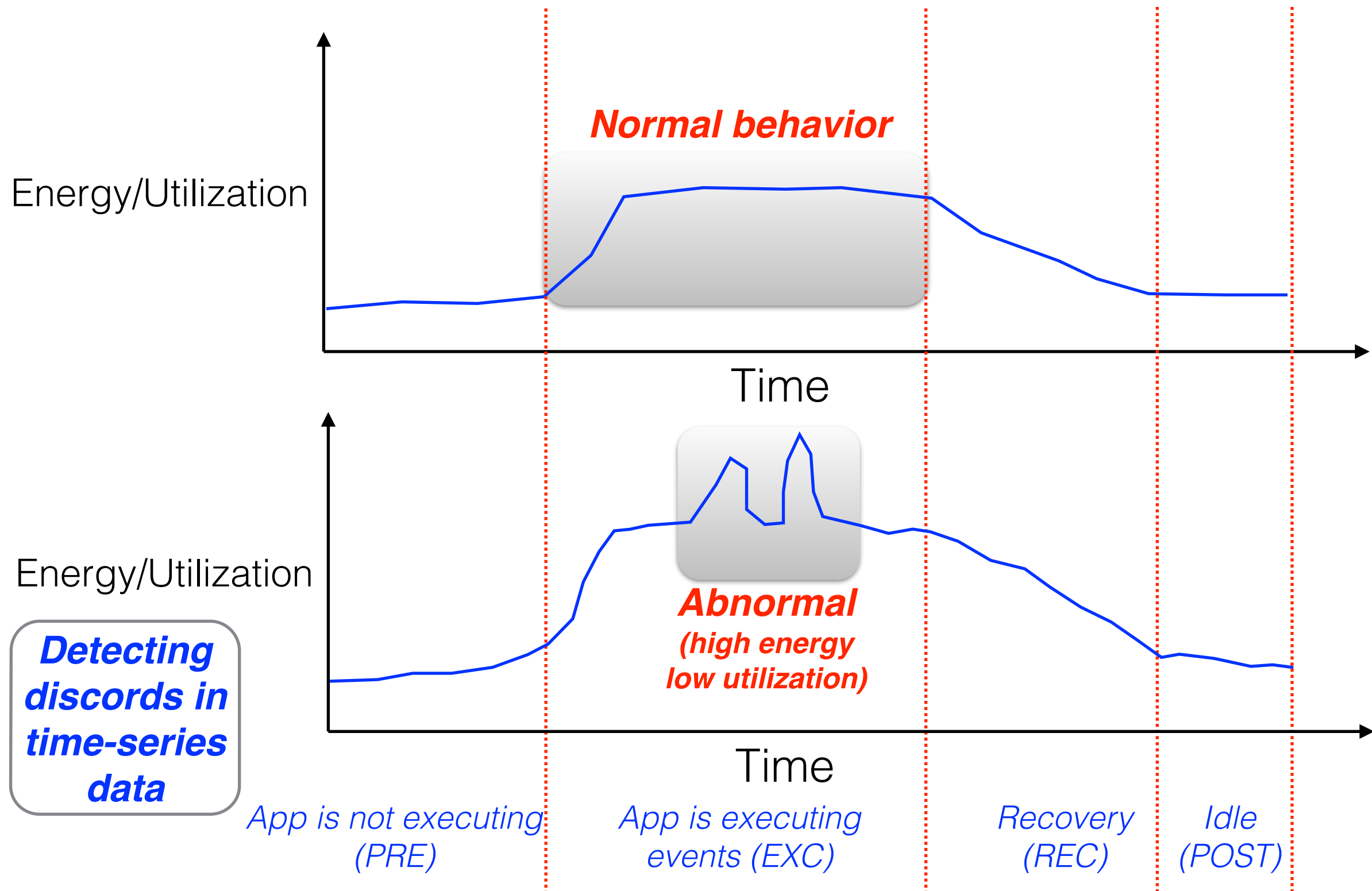


Energy consumption of different components is not even (GPS < CPU)
100% CPU does not consume same energy as GPS being on

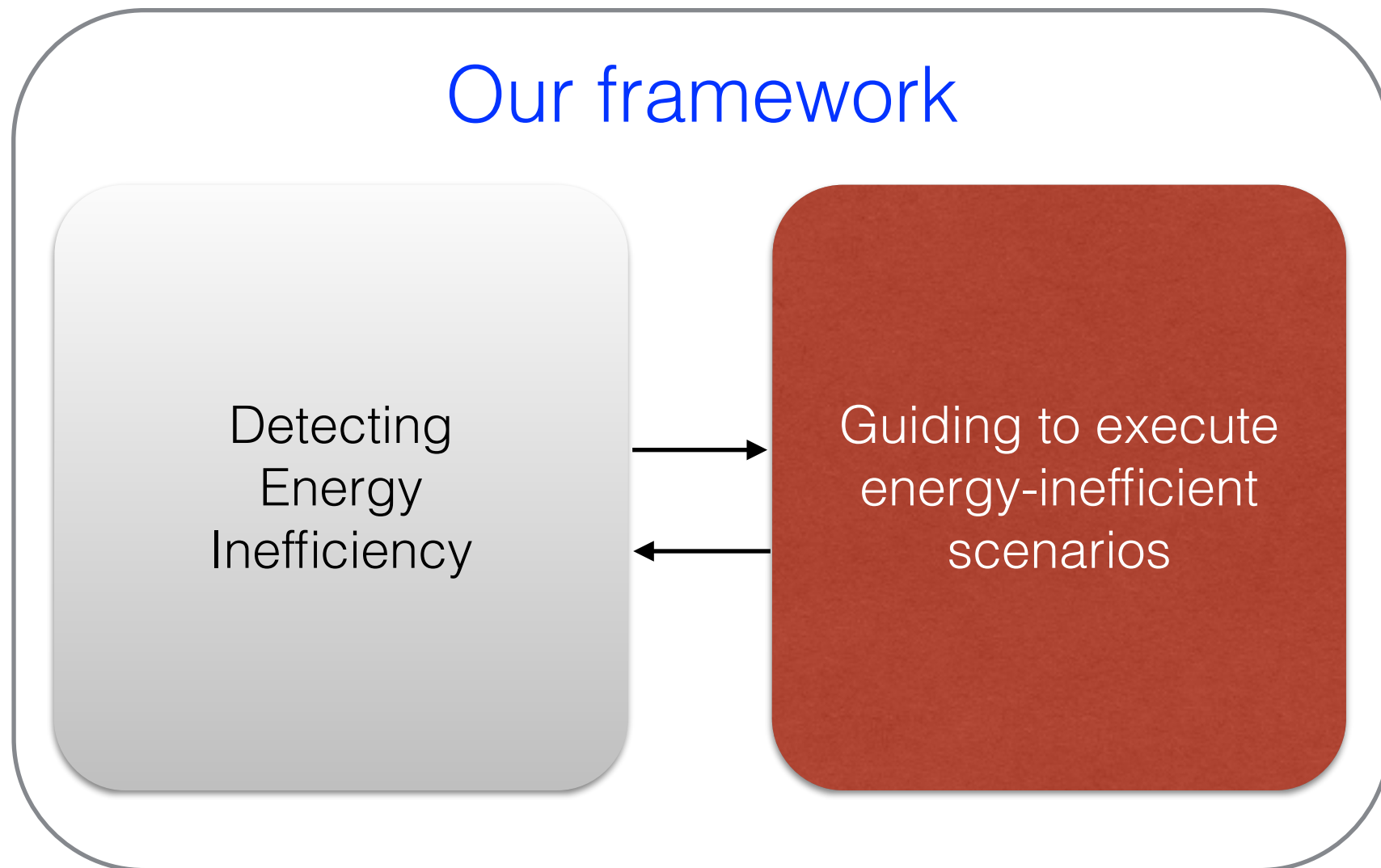
Detecting Energy Bugs



Detecting Energy Hotspots



Test Generation



Guided Exploration

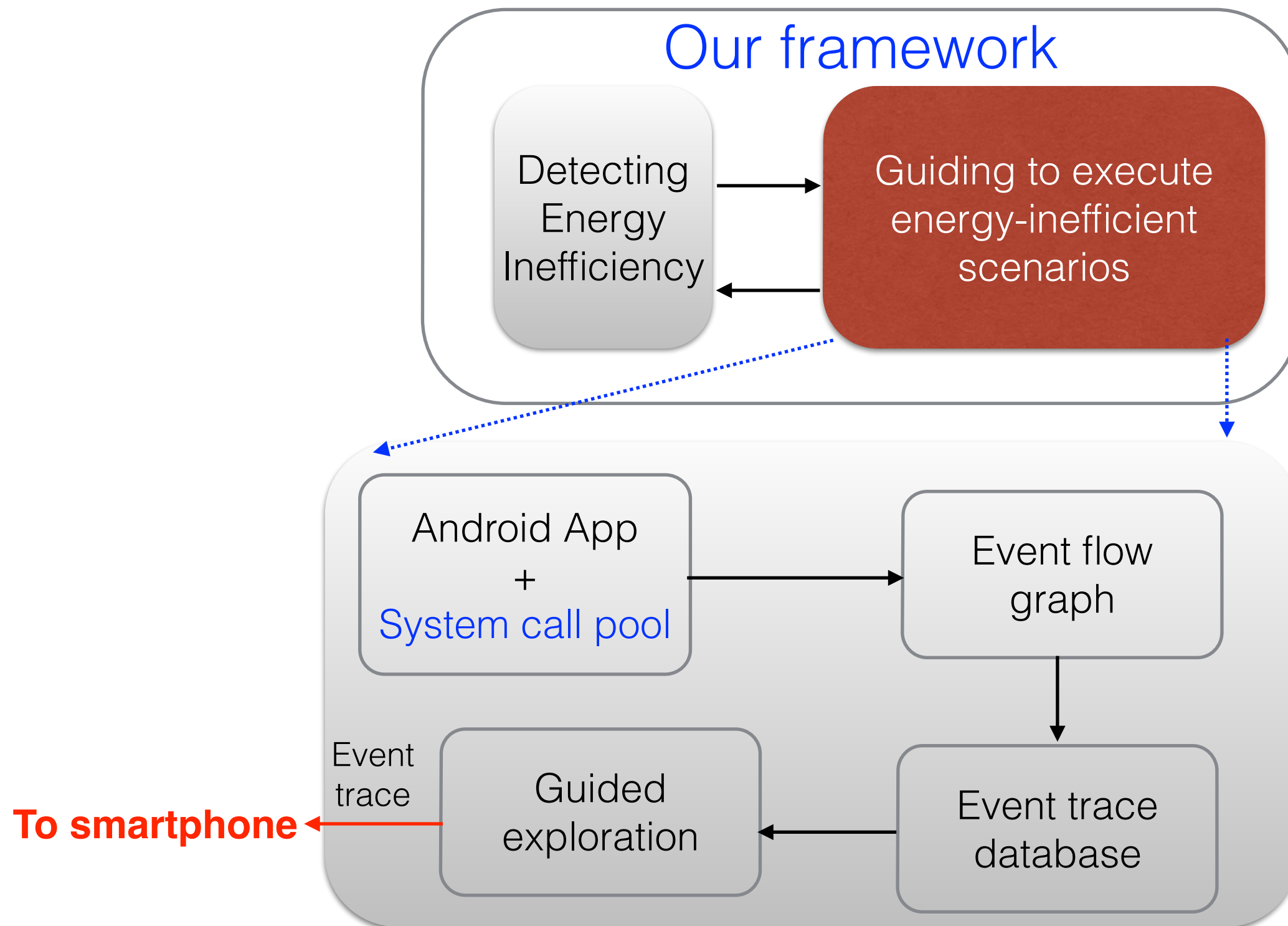
- Energy-inefficient execution
 - Which fragments are energy-inefficient?
 - What is an appropriate coverage metric?

A Broader Categorization

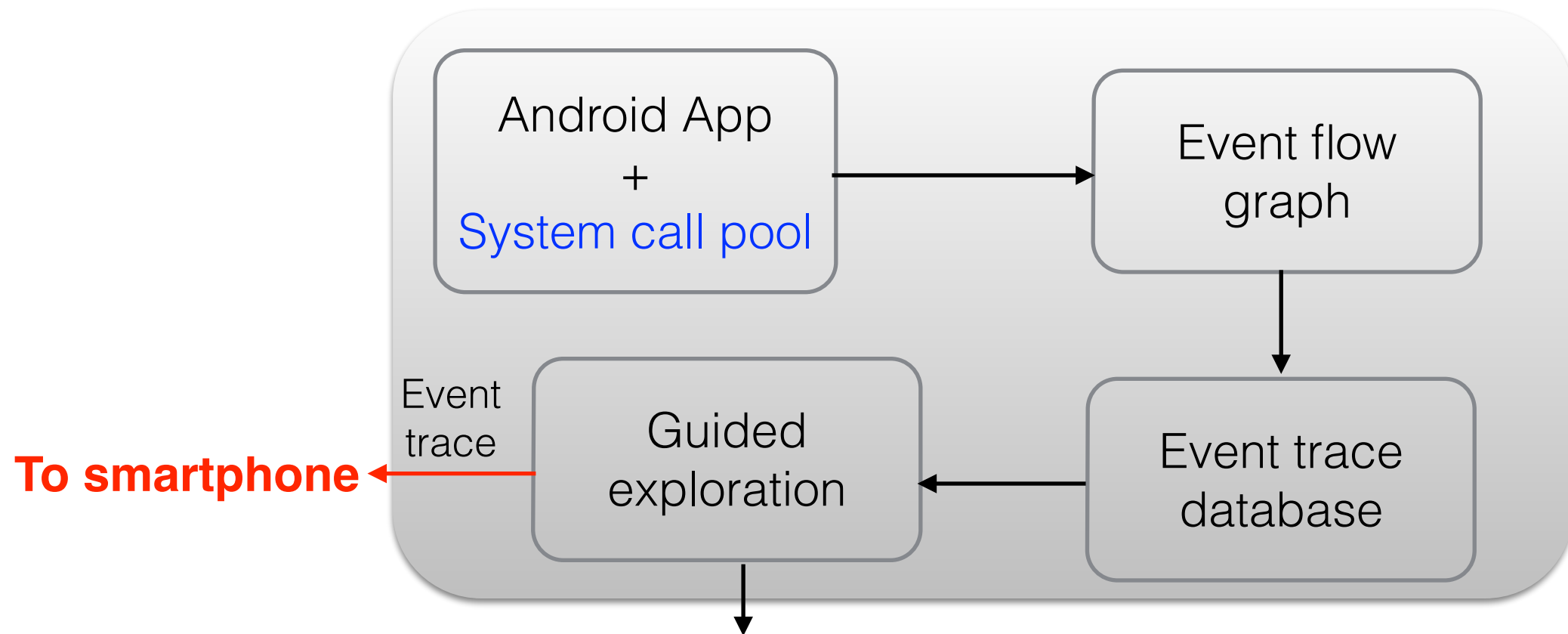
| Cause/Source | Energy Bugs | Energy Hotspots |
|-------------------------|----------------------------|------------------------------|
| Hardware components | Resource leak | Suboptimal resource binding |
| Sleep state transition | Wakelock bug | Tail Energy hotspot |
| Background Service | Vacuous background service | Expensive background service |
| Defective Functionality | Immortality bug | Loop energy hotspot |

Invoked via System Calls

Test Generation



Test Generation



An ordering between trace T1 and T2

$T1 > T2$

s1 s1
↓ ↓
s2 s2
↓
s3

More system calls

Buggy trace

$T1 > T2$

s1 s1 s5
↓ ↓ ↓
s2 s2 s6
↓
s3

Similarity with buggy trace

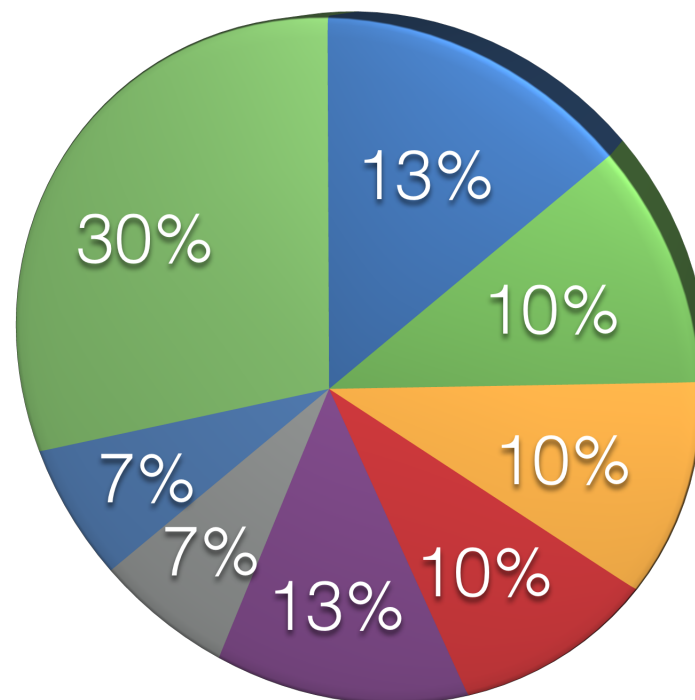
Executed trace

$T1 < T2$

s1 s1 s5
↓ ↓ ↓
s2 s2 s6
↓
s4 s3

Covering more system calls

Evaluation



Category of Android Apps Evaluated

Summary of Evaluation

| App | Feasible traces | Energy Bugs | Energy Hotspots | Type | Reported before |
|------------------|-----------------|-------------|-----------------|---|-----------------|
| Aripuca | 502 | Yes | No | Vacuous background service | No |
| Montreal Transit | 64 | No | Yes | Suboptimal resource binding and more | No |
| Sensor Test | 2800 | Yes | No | Immortality Bug | No |
| 760 KFMB AM | 26 | Yes | Yes | Vacuous background service, suboptimal resource binding | No |

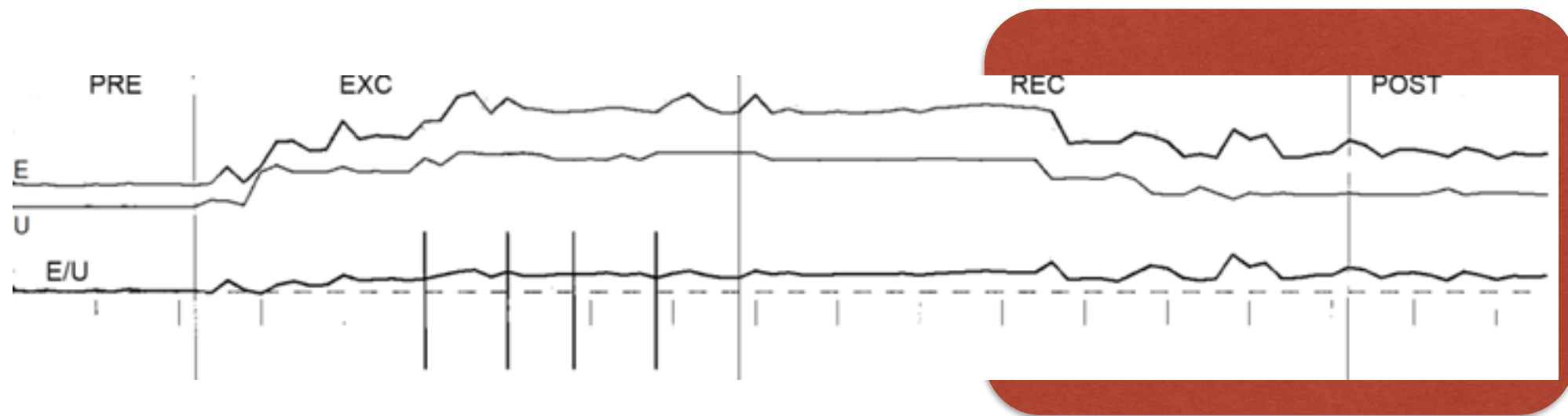
*All Results are in the paper
(10 energy bugs and 3 energy hotspots found
out of 30 tested apps)*

Summary of Evaluation

| App | System call | | Code | | Lines of Code |
|----------------------|-------------|--|------|--|---------------|
| Aagtl | 100 | | 21 | | 11612 |
| Android Battery Dog | 100 | | 17 | | 463 |
| Aripuca | 100 | | 15 | | 4353 |
| Kitchen Timer | 100 | | 30 | | 1101 |
| Montreal Transit | 89 | | 11 | | 10925 |
| NPR News | 100 | | 24 | | 6513 |
| OmniDroid | 83 | | 36 | | 6130 |
| Pedometer | 100 | | 56 | | 849 |
| Vanilla Music Player | 86 | | 20 | | 4081 |

*To cover all system calls, exploring only a small part of the program suffices
A substantial portion of the code is used for provide user feedback, compatibility
over different OS*

Case Studies



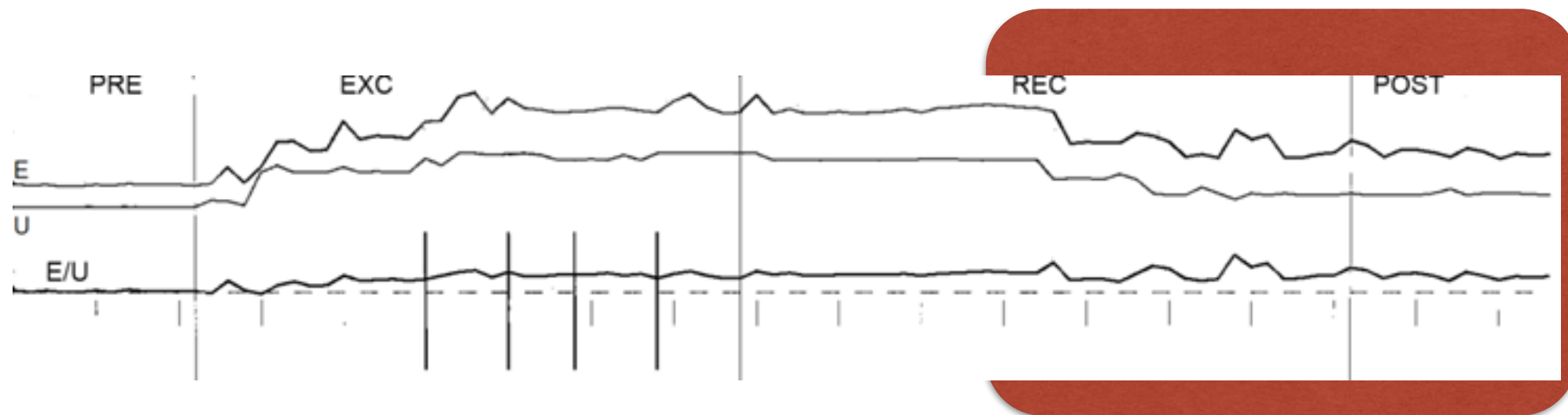
Aripuca (Energy Bug)

Reason: Vacuous Background Service

Fix:

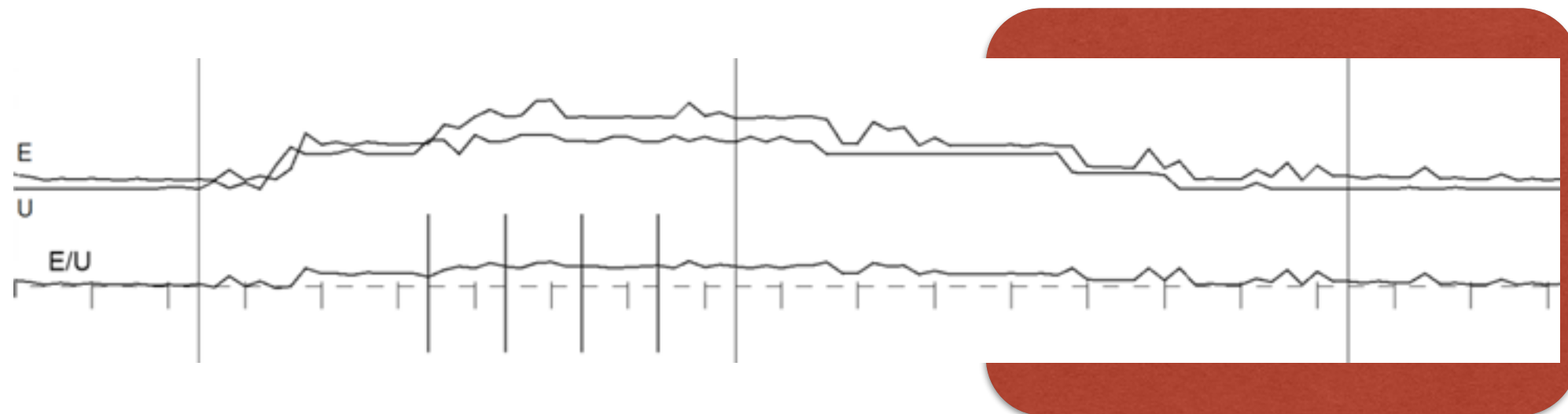
```
serviceConnection.getService().stopLocationUpdates();  
serviceConnection.getService().stopSensorUpdates();
```

Case Studies

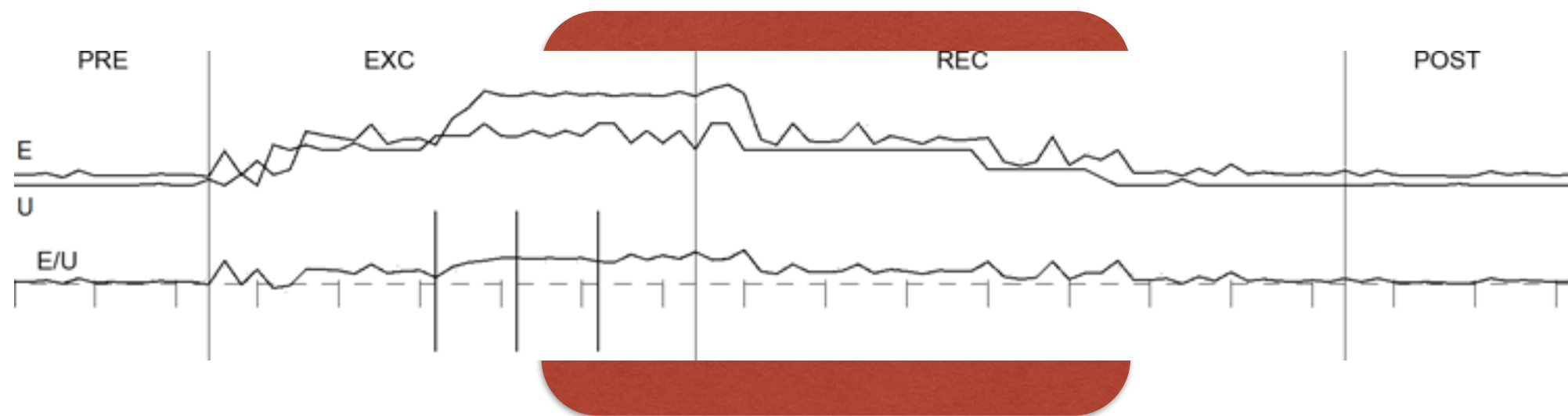


Aripuca (Energy Bug)

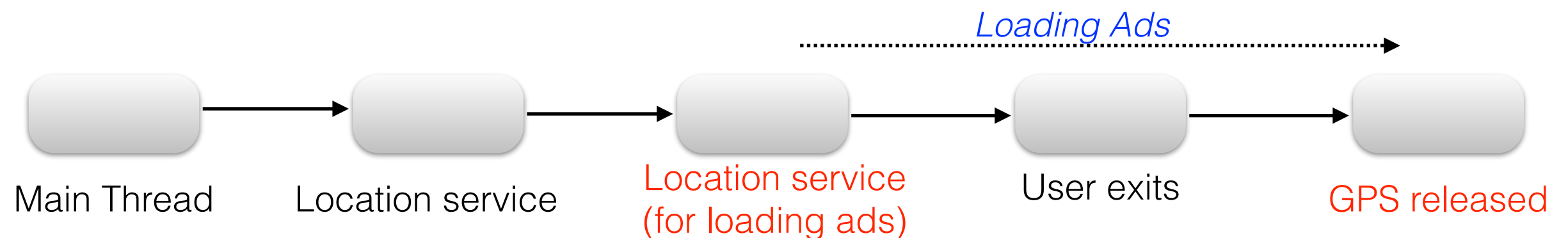
Reason: Vacuous Background Service



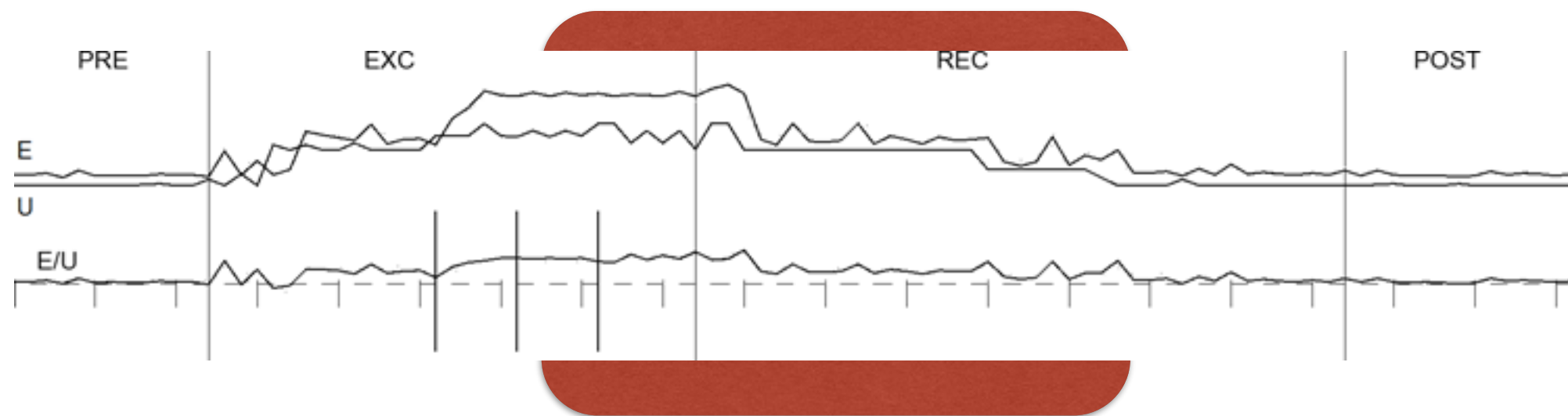
Case Studies



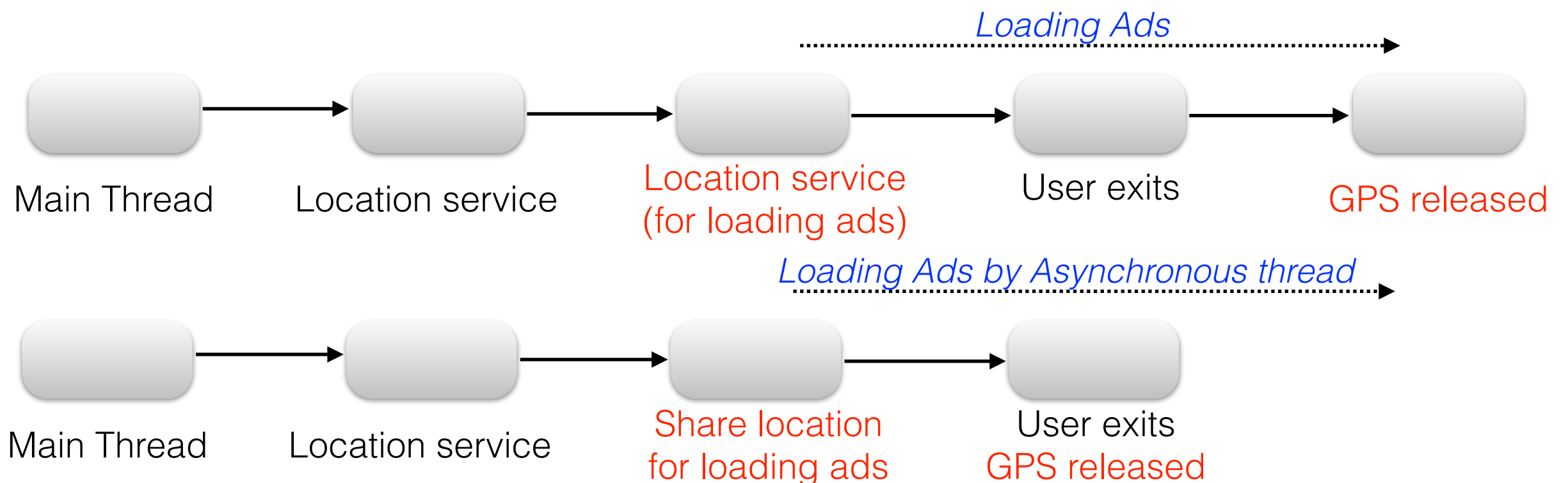
Montreal Transit (Energy hotspot for <5 sec)



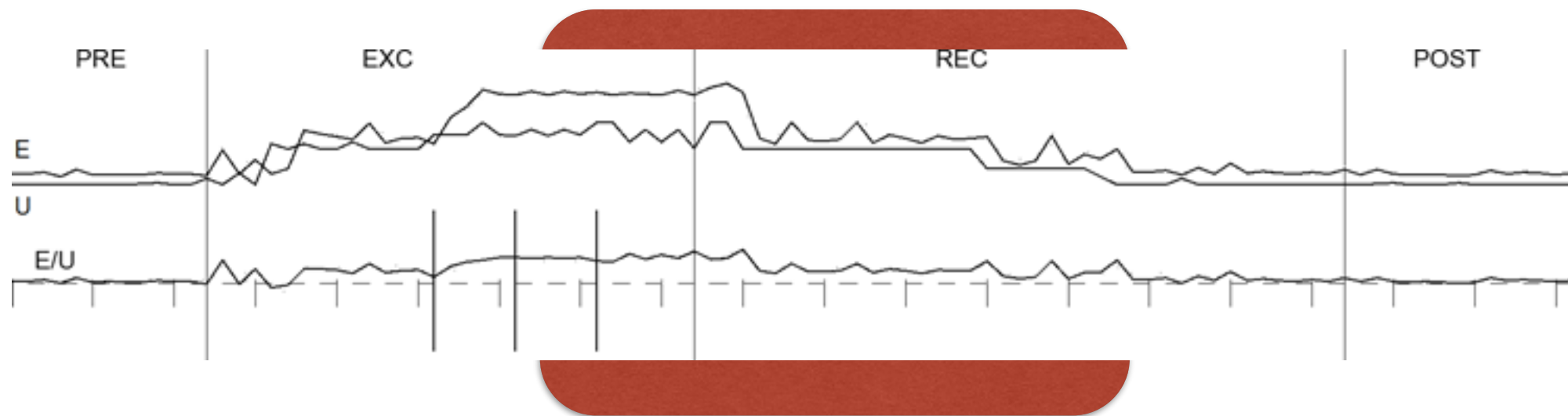
Case Studies



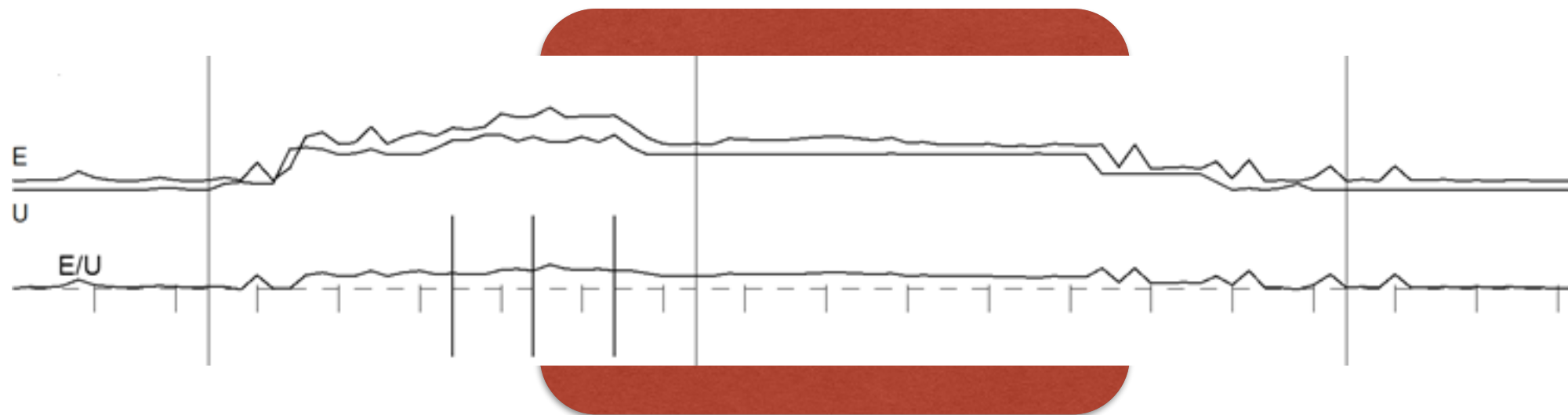
Montreal Transit (Energy hotspot for <5 sec)



Case Studies



Montreal Transit (Energy hotspot for <5 sec)



Montreal Transit (After fixing)

Summary

- Categorization of energy inefficiency
 - Energy bugs
 - Energy hotspots
- A guided exploration of event traces
 - Targeting system call coverage
- Evaluation with Android apps
 - Energy bugs and hotspots exist in several Android apps

Open Problems

- Event Flow Graph may require intelligent inputs
 - Games
- Confirmation of bugs/hotspots require repeated execution
- Debugging and optimization is manual