Energy-Efficient Mobile Web Computing

Yuhao Zhu

The University of Texas at Austin

http://yuhaozhu.com


NOKIA
17:31
Freigabe


NOKIA

Snake
circa 2000

Snake
circa **2000**

Snake
circa **2017**

Mobile Devices are More Capable

Snake
circa **2000**

Snake
circa **2017**

Mobile Devices are More Capable

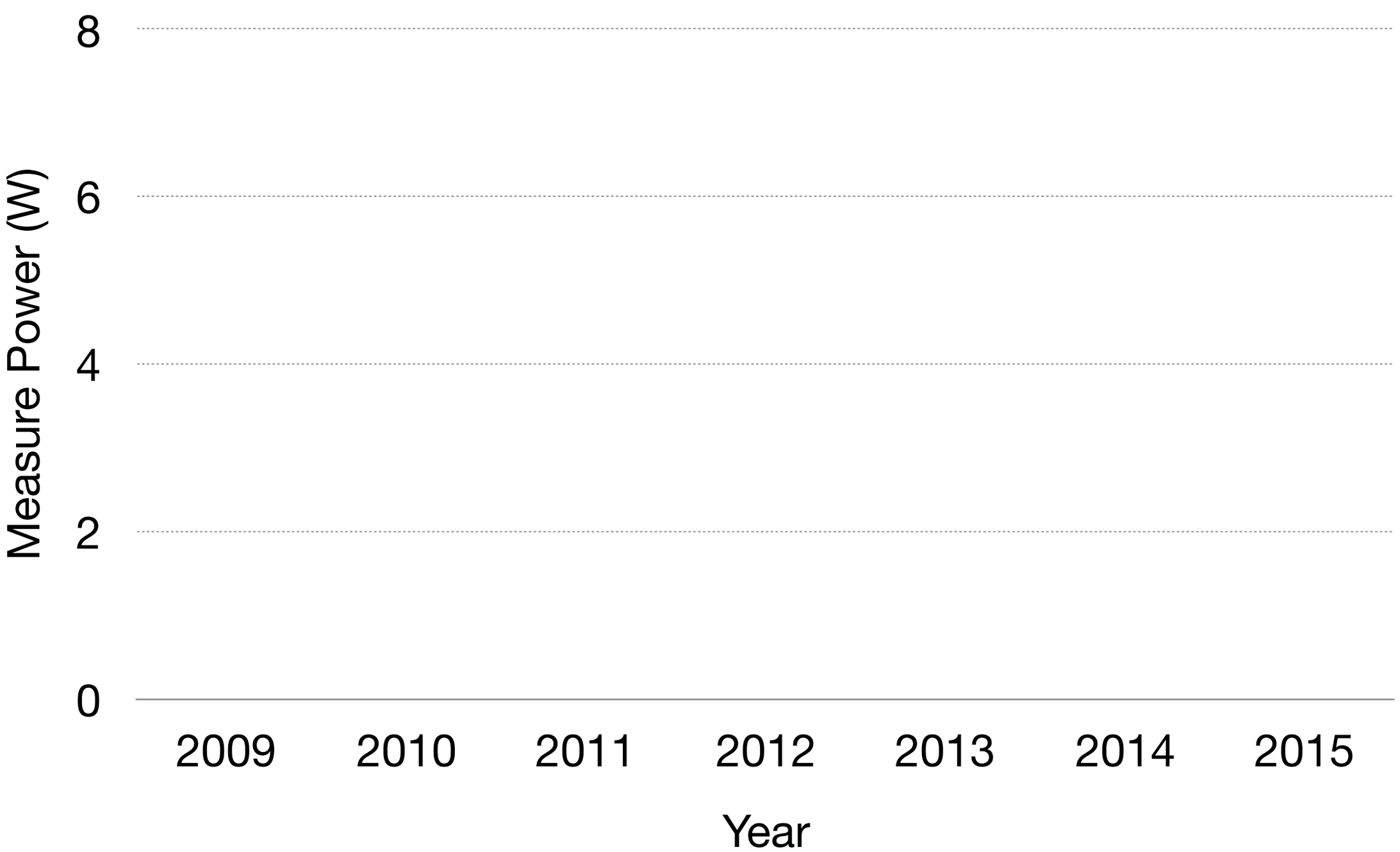
Performance

Mobile Devices are More Capable

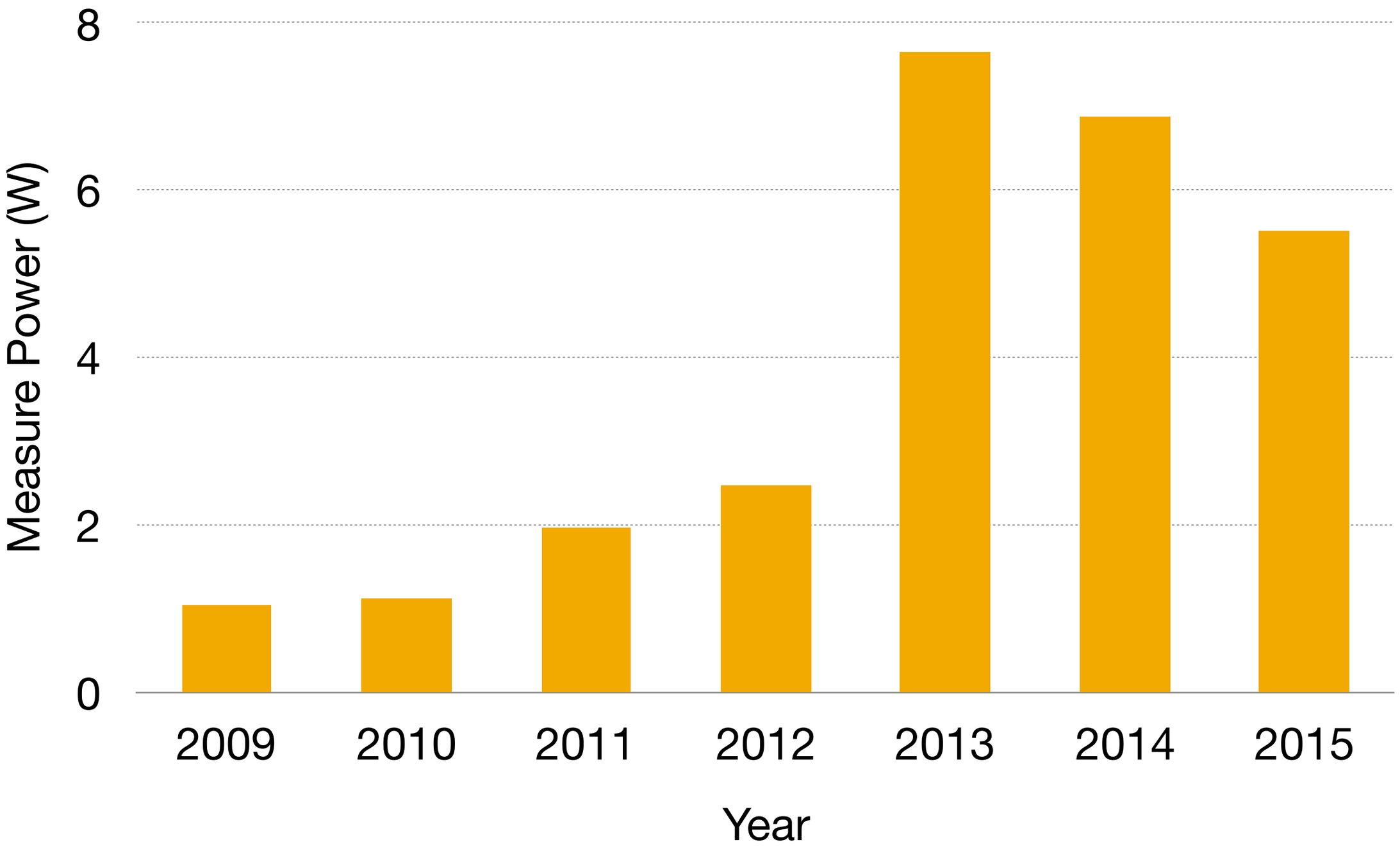
Power

Performance

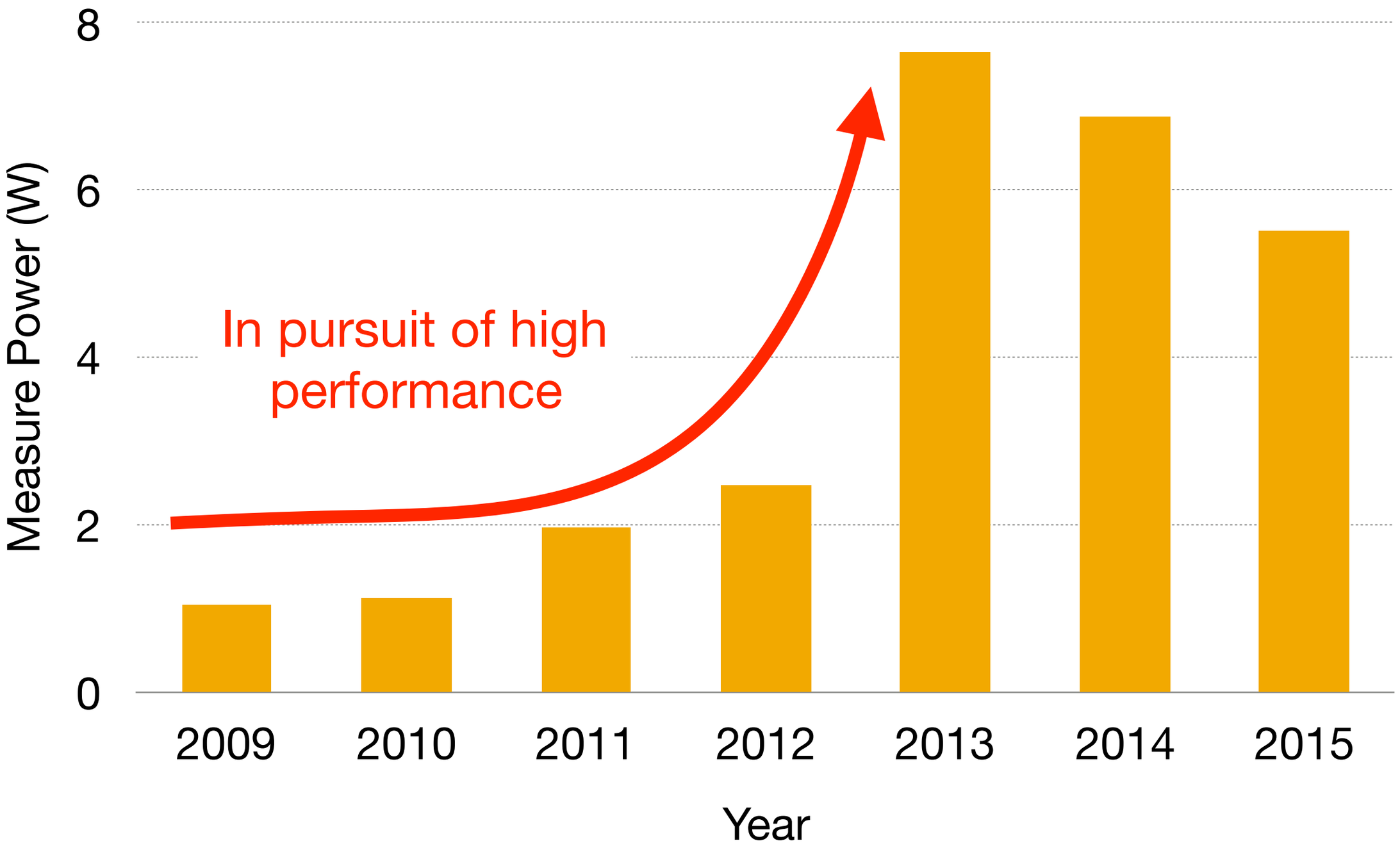
Mobile CPU's Rise to Power [HPCA 2016]



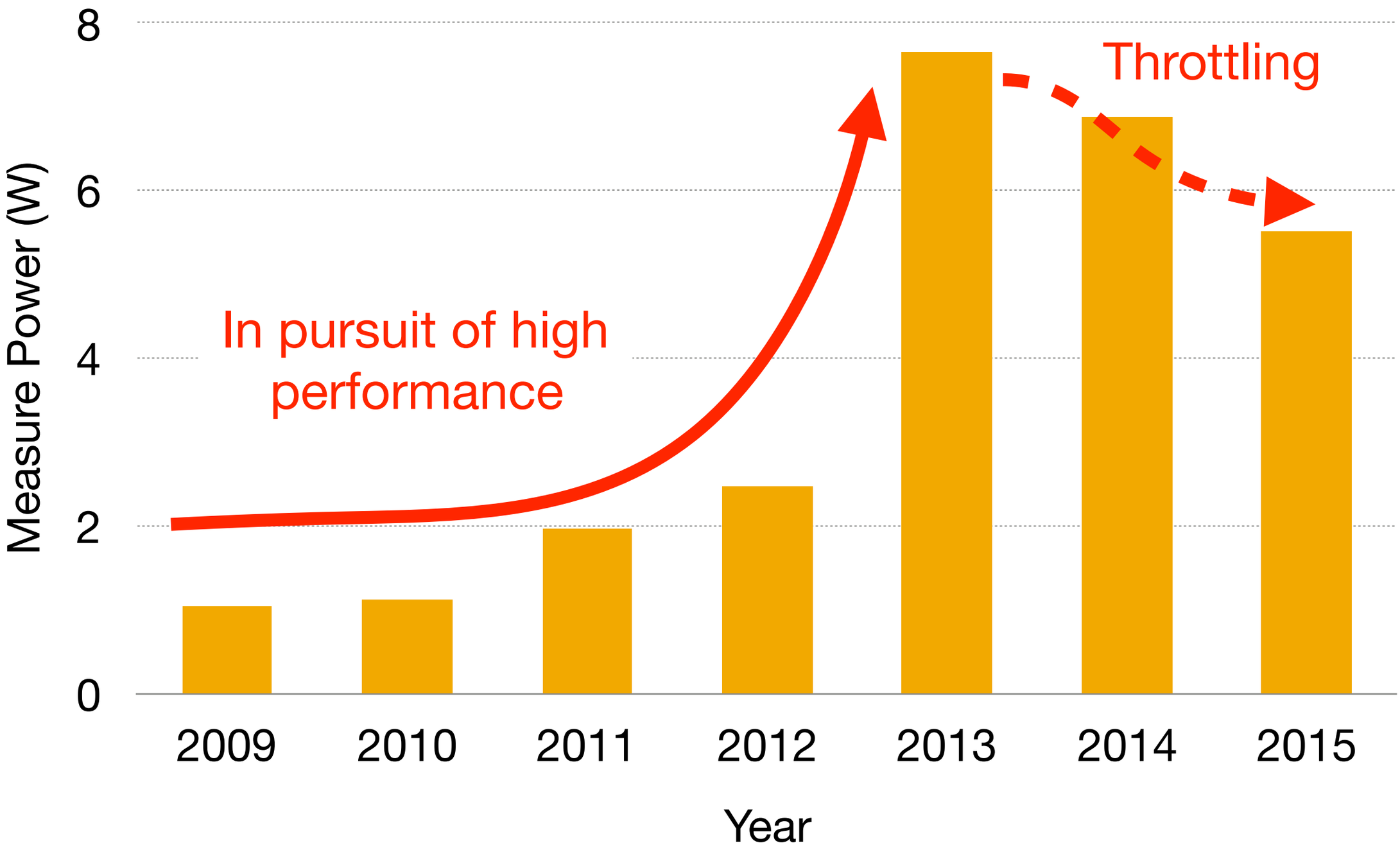
Mobile CPU's Rise to Power [HPCA 2016]



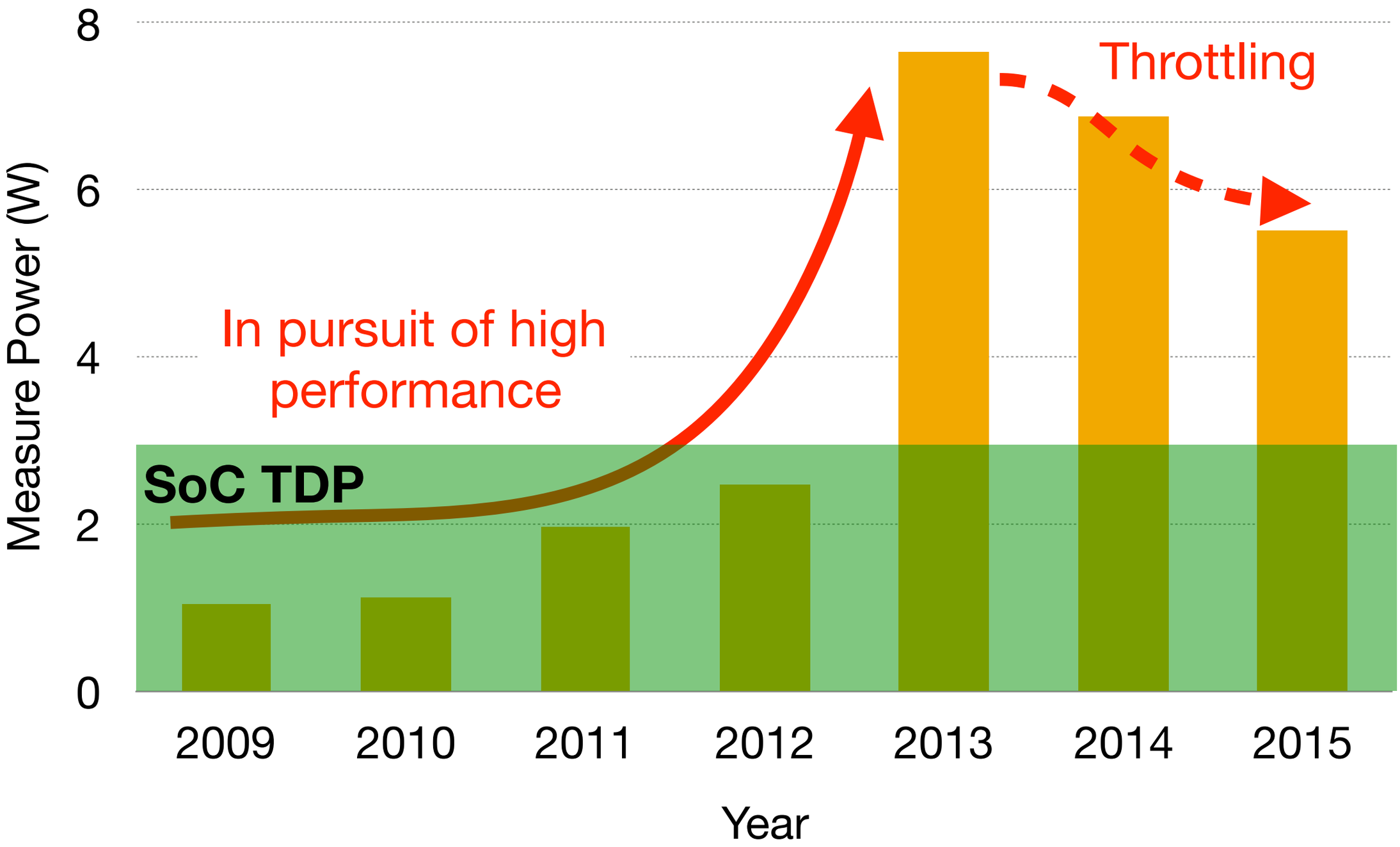
Mobile CPU's Rise to Power [HPCA 2016]



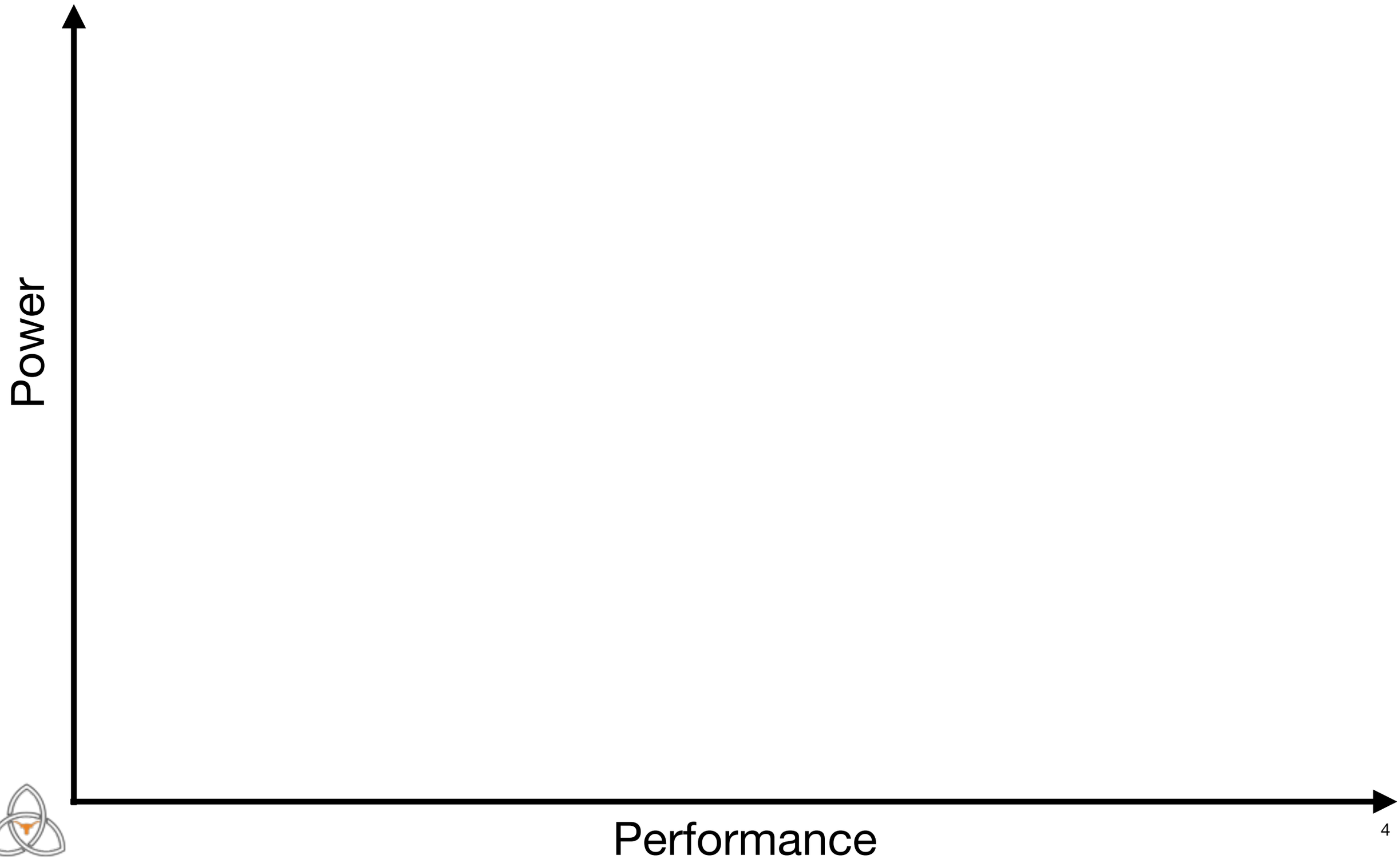
Mobile CPU's Rise to Power [HPCA 2016]



Mobile CPU's Rise to Power [HPCA 2016]



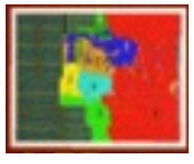
Mobile Processor Design “Strategy”



Mobile Processor Design “Strategy”

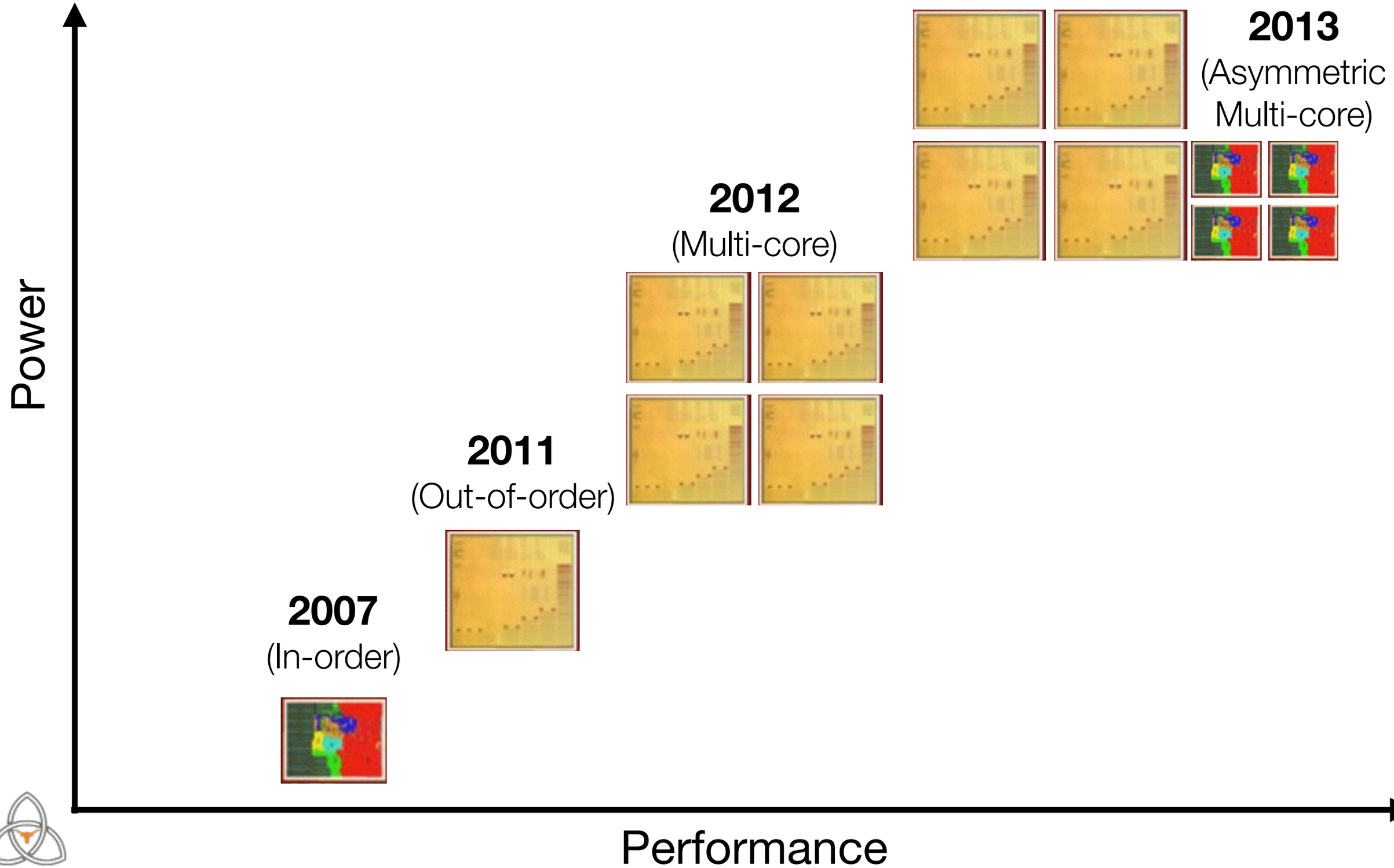
Power

2007
(In-order)



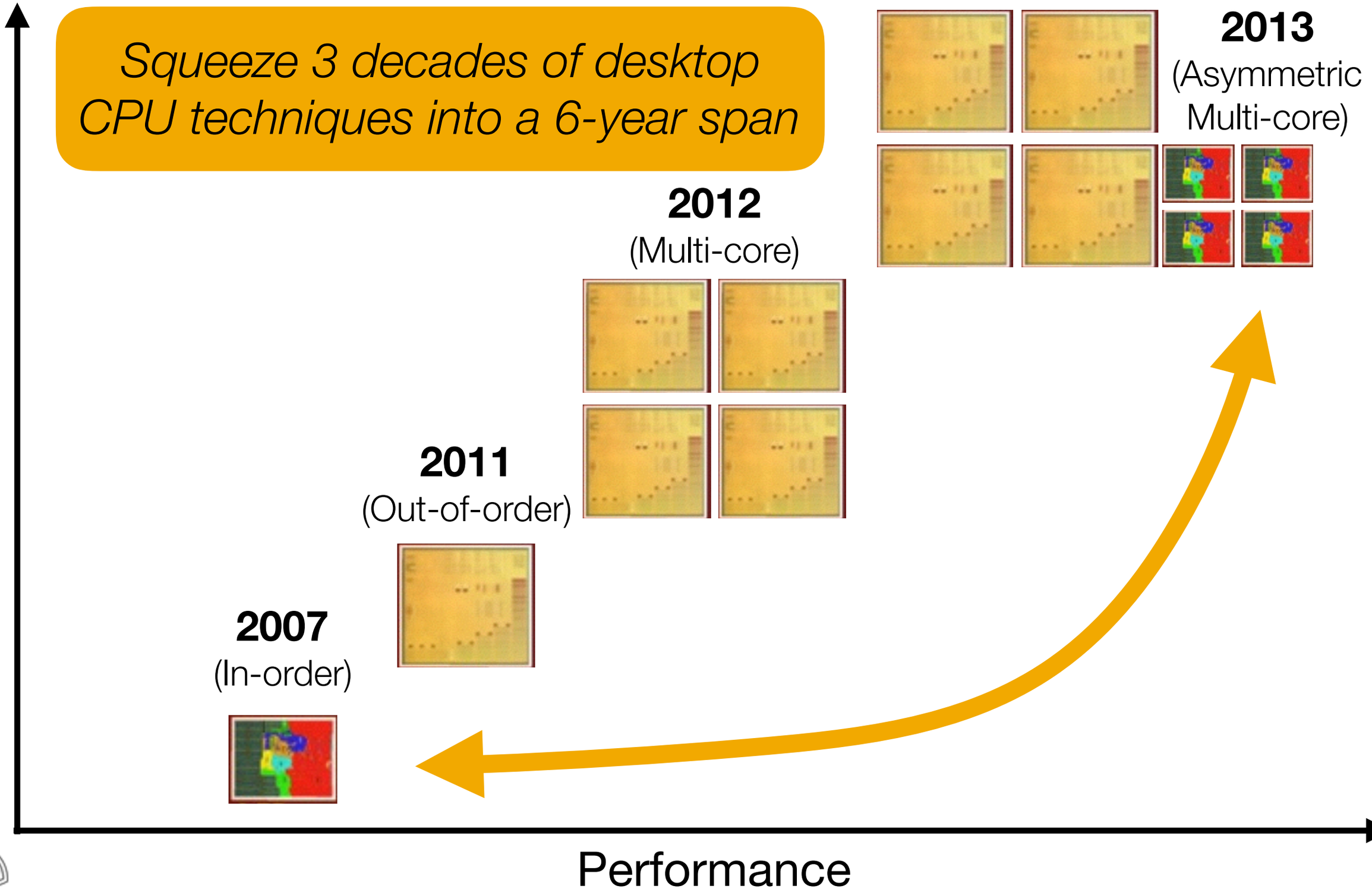
Performance

Mobile Processor Design “Strategy”



Mobile Processor Design "Strategy"

Squeeze 3 decades of desktop CPU techniques into a 6-year span



No Moore's Law for batteries

Fred Schlachter[1]
American Physical Society, Washington, DC 20045

Proceedings of the National Academy of Sciences, 2013

The public has become accustomed to rapid progress in mobile phone technology, computers, and access to information; tablet computers, smart phones, and other powerful new devices are familiar to most people on the planet.

These developments are due in part to the ongoing exponential increase in computer processing power, doubling approximately every 2 years for the past several decades. This pattern is usually called Moore's Law and is named for Gordon Moore, a cofounder of Intel. The law is not a law like that for gravity; it is an empirical observation, which has become a self-fulfilling prophecy. Unfortunately, much of the public has come to expect that all technology does, will, or should follow such a law, which is not consistent with our everyday observations: For example, the maximum speed of cars, planes, or ships does not increase exponentially; maximum speed barely increases at all.

Cars require a portable fuel, preferably one that is widely available, low in cost, and with a high energy density. Gasoline

there is a Moore's Law for computer processors is that electrons are small and they do not take up space on a chip. Chip performance is limited by the lithography technology used to fabricate the chips; as lithography improves ever smaller features can be made on processors. Batteries are not like this. Ions, which transfer charge in batteries, are large, and they take up space, as do anodes, cathodes, and electrolytes. A D-cell battery stores more energy than an AA-cell. Potentials in a battery are dictated by the relevant chemical reactions, thus limiting eventual battery performance. Significant improvement in battery capacity can only be made by changing to a different chemistry.

Scientists and battery experts, who have been optimistic in the recent past about improving lithium-ion batteries and about developing new battery chemistries—lithium/air and lithium/sulfur are the leading candidates—are considerably less optimistic now. Improvement in energy storage density of lithium-ion batteries has been only incremental for the past decade. A large-

OPINION

No Moore's Law for batteries

Fred Schlachter[1]
American Physical Society, Washington, DC 20045

Proceedings of the National Academy of Sciences, 2013

Number of transistors doubles every 2 years.

The public has become accustomed to rapid progress in mobile phone technology, computers, and access to information; tablet computers, smart phones, and other powerful new devices are familiar to most people on the planet.

These developments are due in part to the ongoing exponential increase in computer processing power, doubling approximately every 2 years for the past several decades. This pattern is usually called Moore's Law and is named for Gordon Moore, a cofounder of Intel. The law is not a law like that for gravity; it is an empirical observation, which has become a self-fulfilling prophecy. Unfortunately, much of the public has come to expect that all technology does, will, or should follow such a law, which is not consistent with our everyday observations: For example, the maximum speed of cars, planes, or ships does not increase exponentially; maximum speed barely increases at all.

Cars require a portable fuel, preferably one that is widely available, low in cost, and with a high energy density. Gasoline

there is a Moore's Law for computer processors is that electrons are small and they do not take up space on a chip. Chip performance is limited by the lithography technology used to fabricate the chips; as lithography improves ever smaller features can be made on processors. Batteries are not like this. Ions, which transfer charge in batteries, are large, and they take up space, as do anodes, cathodes, and electrolytes. A D-cell battery stores more energy than an AA-cell. Potentials in a battery are dictated by the relevant chemical reactions, thus limiting eventual battery performance. Significant improvement in battery capacity can only be made by changing to a different chemistry.

Scientists and battery experts, who have been optimistic in the recent past about improving lithium-ion batteries and about developing new battery chemistries—lithium/air and lithium/sulfur are the leading candidates—are considerably less optimistic now. Improvement in energy storage density of lithium-ion batteries has been only incremental for the past decade. A large-

OPINION

No Moore's Law for batteries

Fred Schlachter[1]
American Physical Society, Washington, DC 20045

Proceedings of the National Academy of Sciences, 2013

The public has become accustomed to rapid progress in mobile phone technology, computers, and access to information; tablet computers, smart phones, and other powerful new devices are familiar to most people on the planet.

These developments are due in part to the ongoing exponential increase in computer processing power, doubling approximately every 2 years for the past several decades. This pattern is usually called Moore's Law and is named for Gordon Moore, a cofounder of Intel. The law is not a law like that for gravity; it is an empirical observation, which has become a self-fulfilling prophecy. Unfortunately, much of the public has come to expect that all technology does, will, or should follow such a law, which is not consistent with our everyday observations: For example, the maximum speed of cars, planes, or ships does not increase exponentially; maximum speed barely increases at all.

Cars require a portable fuel, preferably one that is widely available, low in cost, and with a high energy density. Gasoline

there is a Moore's Law for computer processors is that electrons are small and they do not take up space on a chip. Chip performance is limited by the lithography technology used to fabricate the chips; as lithography improves ever smaller features can be made on processors. Batteries are not like this. Ions, which transfer charge in batteries, are large, and they take up space, as do anodes, cathodes, and electrolytes. A D-cell battery stores more energy than an AA-cell. Potentials in a battery are dictated by the relevant chemical reactions, thus limiting eventual battery performance. Significant improvement in battery capacity can only be made by changing to a different chemistry.

Scientists and battery experts, who have been optimistic in the recent past about improving lithium-ion batteries and about developing new battery chemistries—lithium/air and lithium/sulfur are the leading candidates—are considerably less optimistic now. Improvement in energy storage density of lithium-ion batteries has been only incremental for the past decade. A large-

Number of transistors doubles every 2 years.

Li-ion battery density doubles every 10 years.

OPINION

No Moore's Law for batteries

Fred Schlachter[1]
American Physical Society, Washington, DC 20045

Proceedings of the National Academy of Sciences, 2013

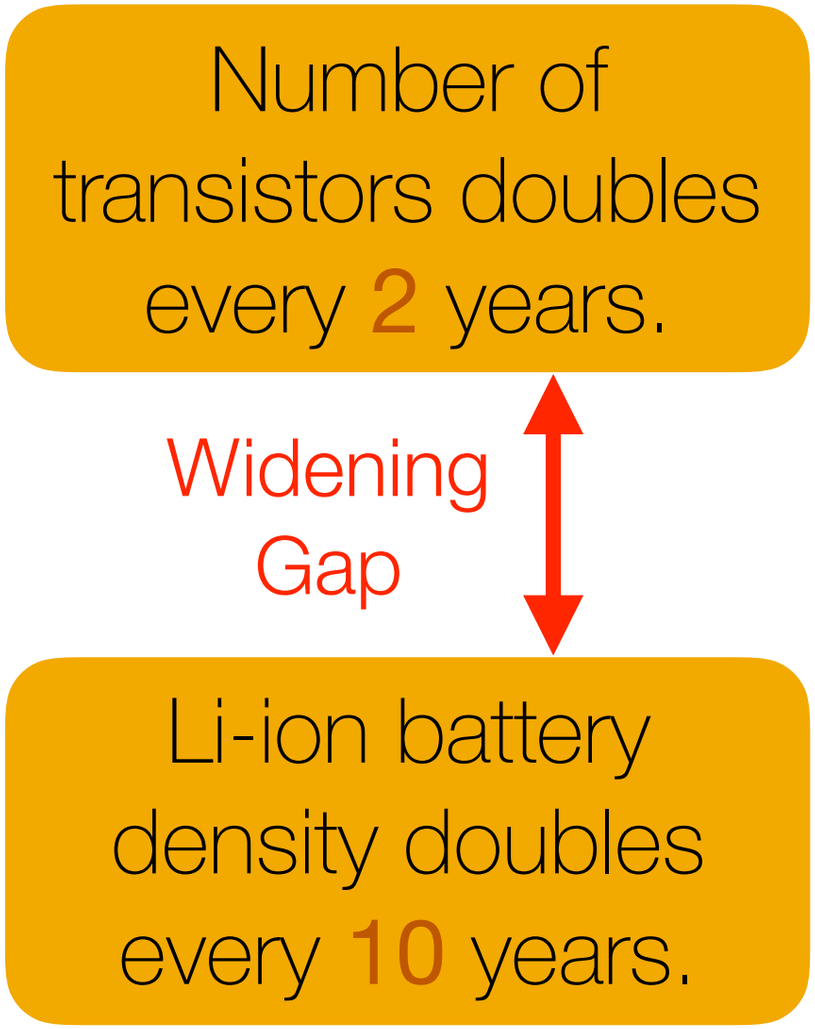
The public has become accustomed to rapid progress in mobile phone technology, computers, and access to information; tablet computers, smart phones, and other powerful new devices are familiar to most people on the planet.

These developments are due in part to the ongoing exponential increase in computer processing power, doubling approximately every 2 years for the past several decades. This pattern is usually called Moore's Law and is named for Gordon Moore, a cofounder of Intel. The law is not a law like that for gravity; it is an empirical observation, which has become a self-fulfilling prophecy. Unfortunately, much of the public has come to expect that all technology does, will, or should follow such a law, which is not consistent with our everyday observations: For example, the maximum speed of cars, planes, or ships does not increase exponentially; maximum speed barely increases at all.

Cars require a portable fuel, preferably one that is widely available, low in cost, and with a high energy density. Gasoline

there is a Moore's Law for computer processors is that electrons are small and they do not take up space on a chip. Chip performance is limited by the lithography technology used to fabricate the chips; as lithography improves ever smaller features can be made on processors. Batteries are not like this. Ions, which transfer charge in batteries, are large, and they take up space, as do anodes, cathodes, and electrolytes. A D-cell battery stores more energy than an AA-cell. Potentials in a battery are dictated by the relevant chemical reactions, thus limiting eventual battery performance. Significant improvement in battery capacity can only be made by changing to a different chemistry.

Scientists and battery experts, who have been optimistic in the recent past about improving lithium-ion batteries and about developing new battery chemistries—lithium/air and lithium/sulfur are the leading candidates—are considerably less optimistic now. Improvement in energy storage density of lithium-ion batteries has been only incremental for the past decade. A large-

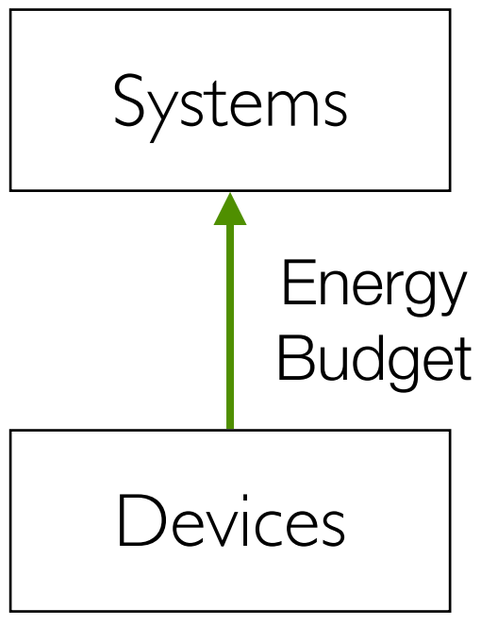


The Mobile Computing Virtuous Cycle

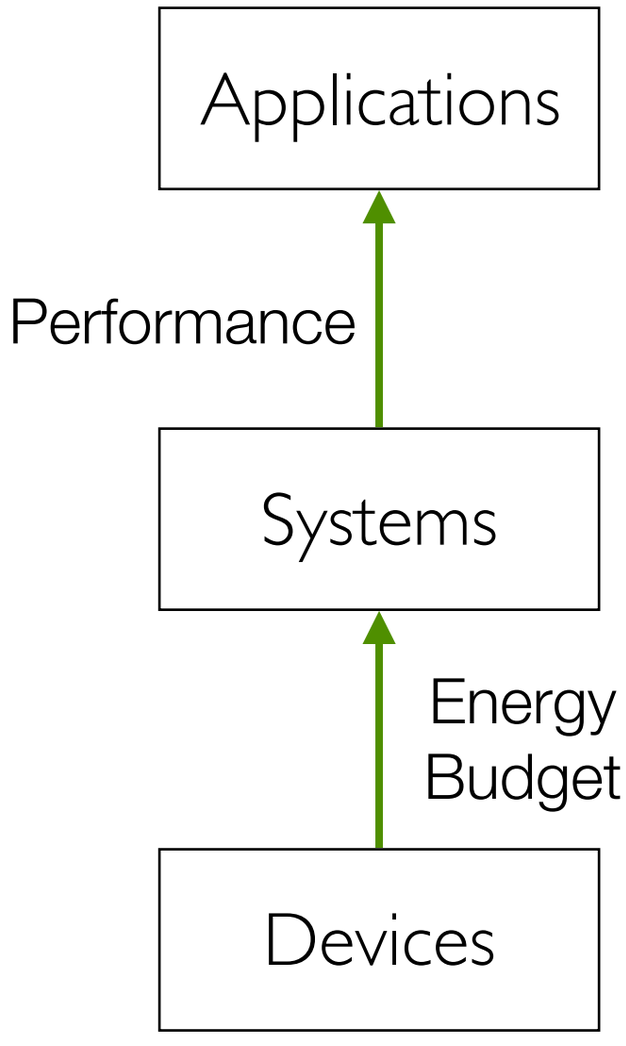
The Mobile Computing Virtuous Cycle

Devices

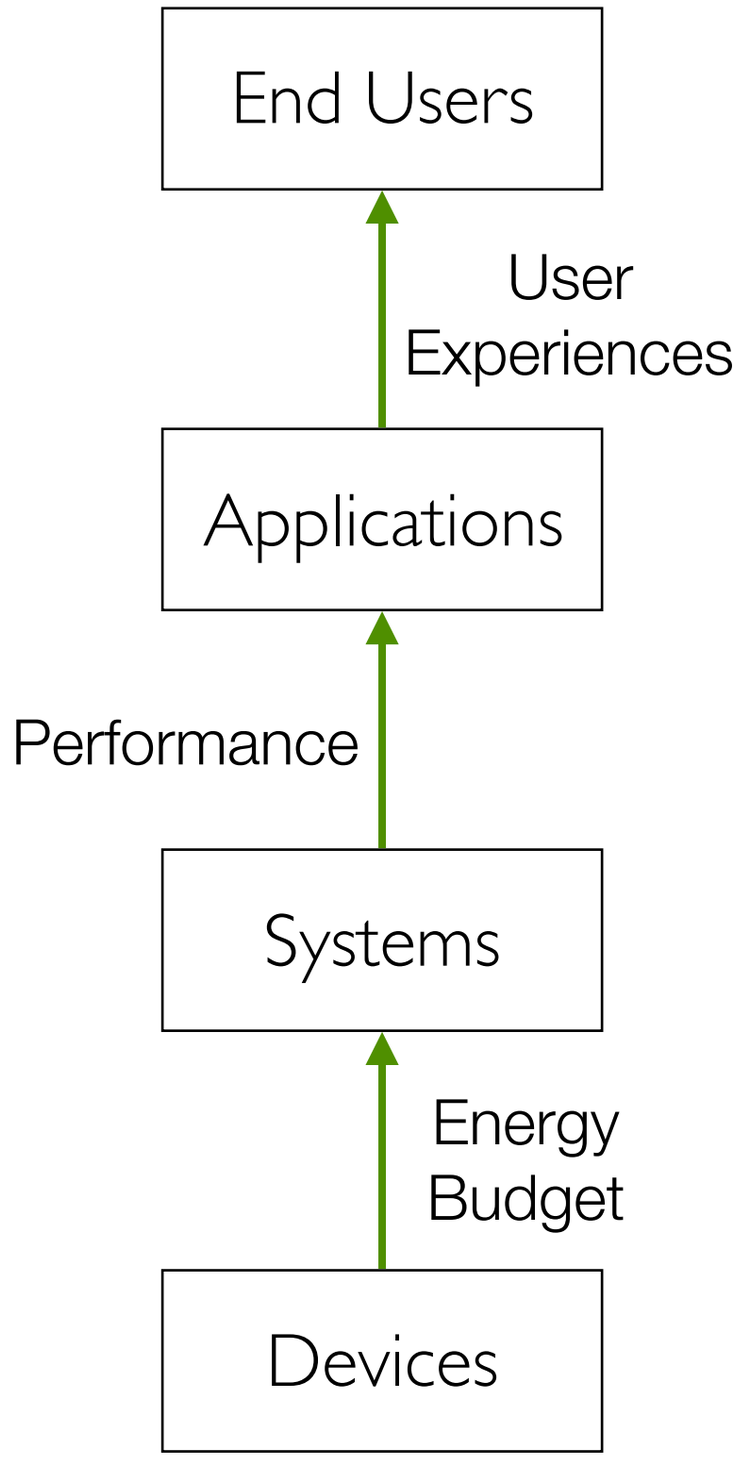
The Mobile Computing Virtuous Cycle



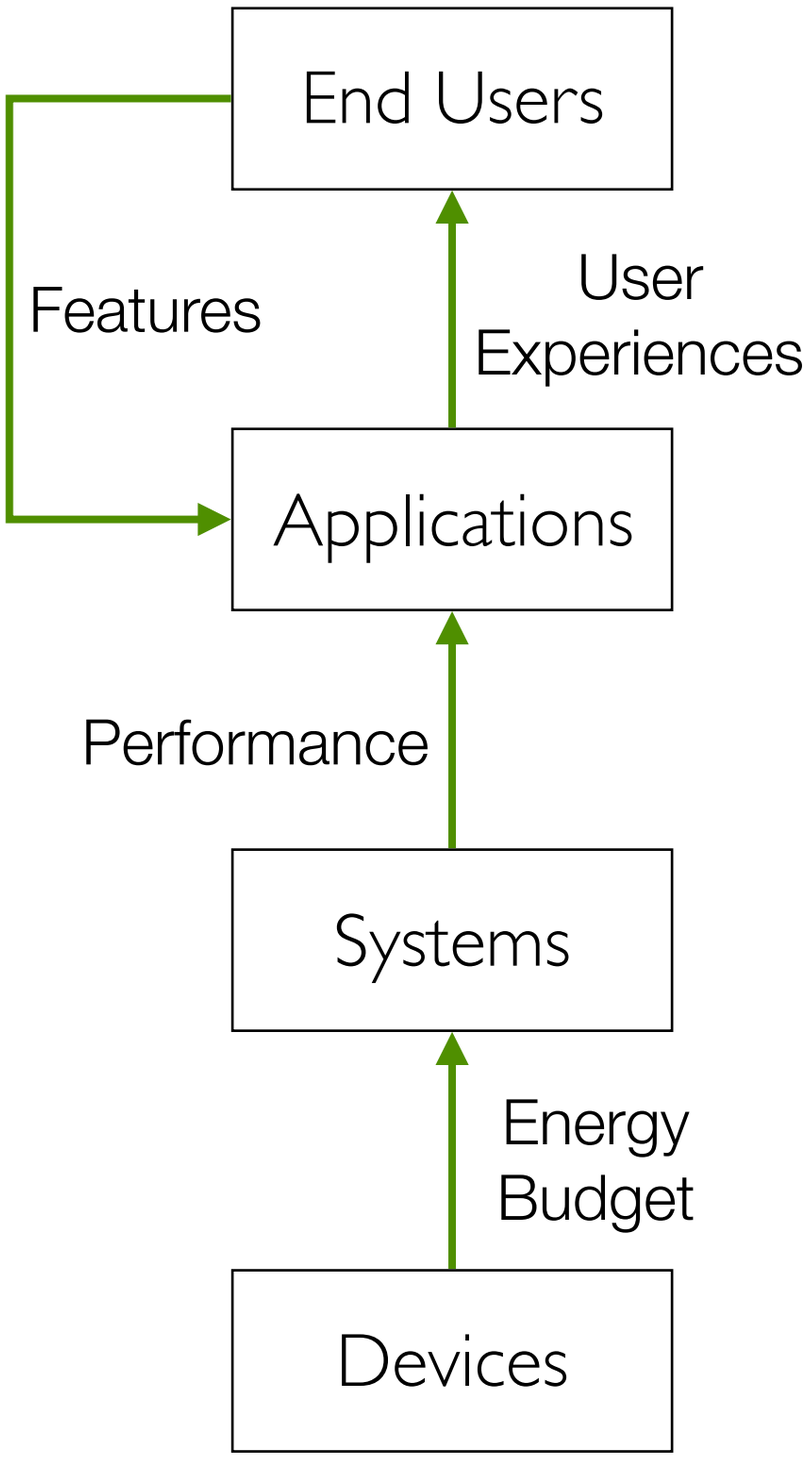
The Mobile Computing Virtuous Cycle



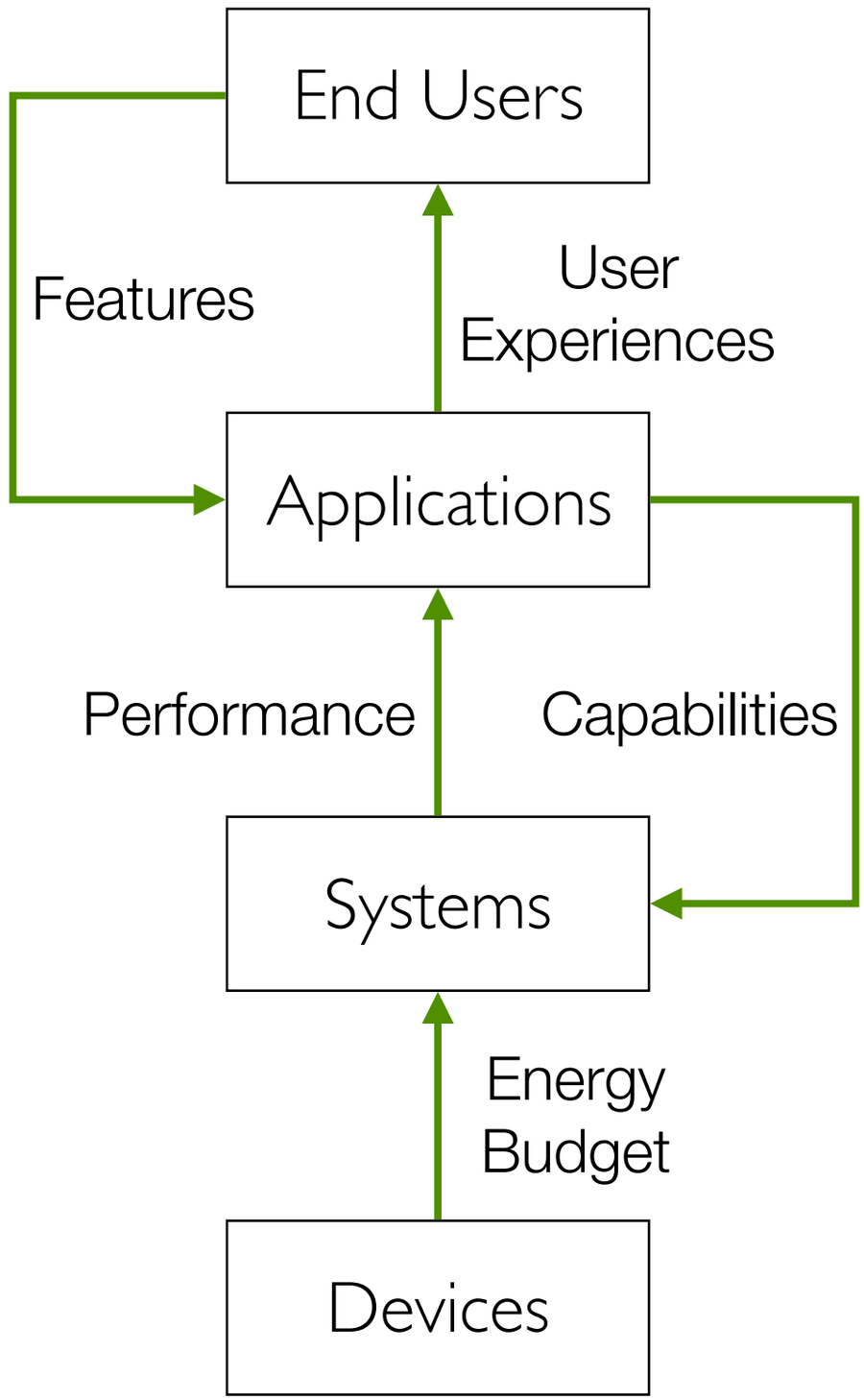
The Mobile Computing Virtuous Cycle



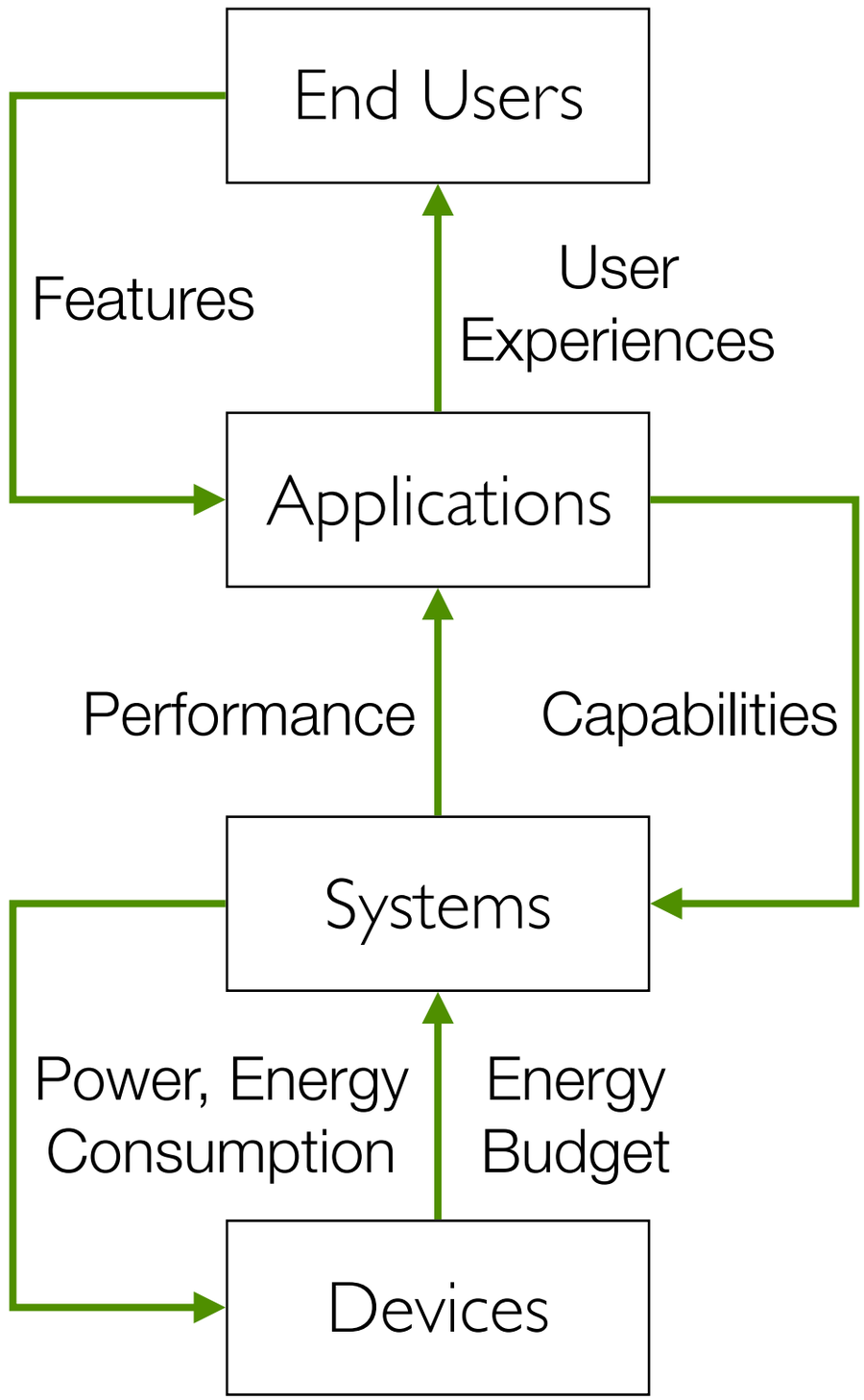
The Mobile Computing Virtuous Cycle



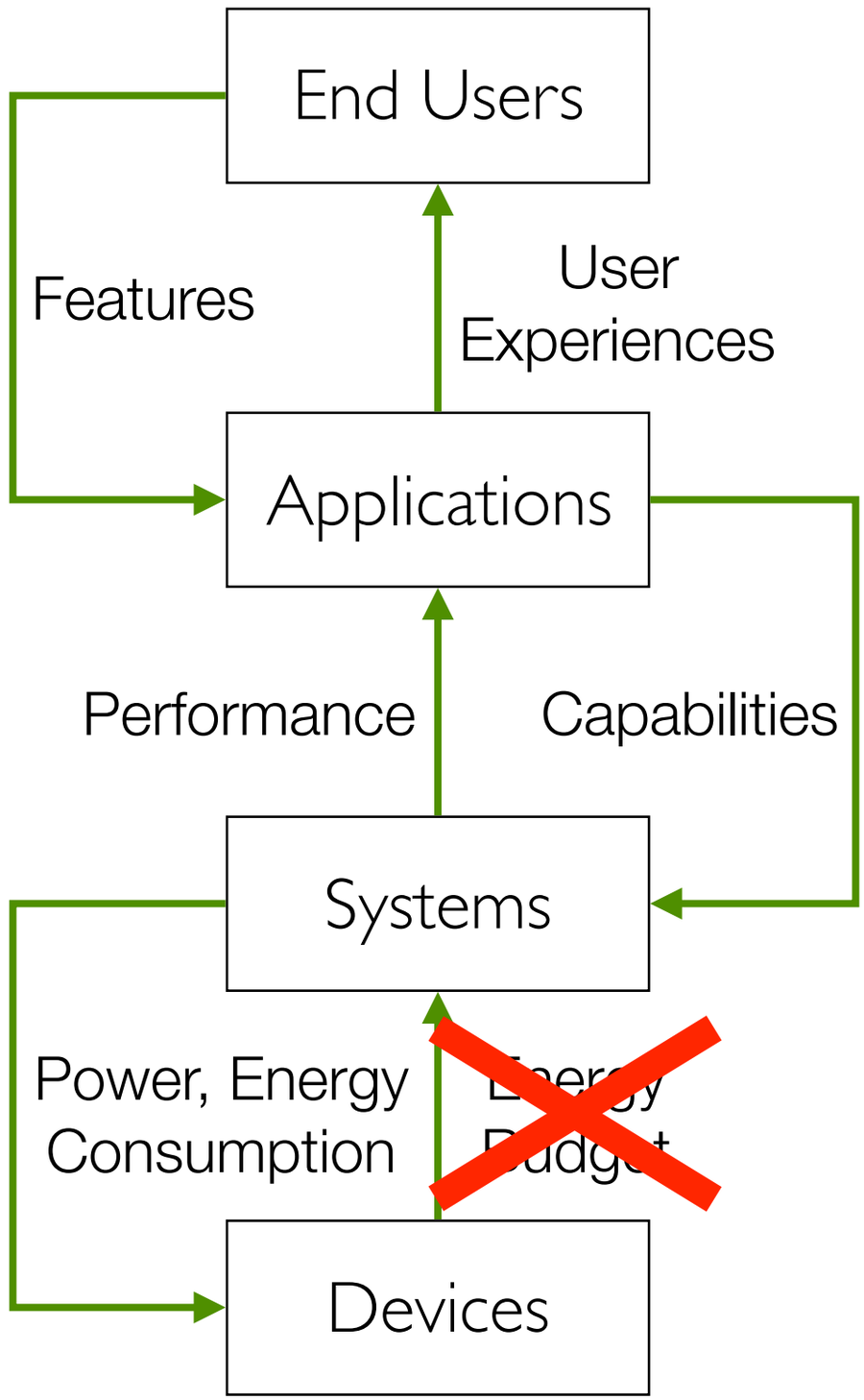
The Mobile Computing Virtuous Cycle



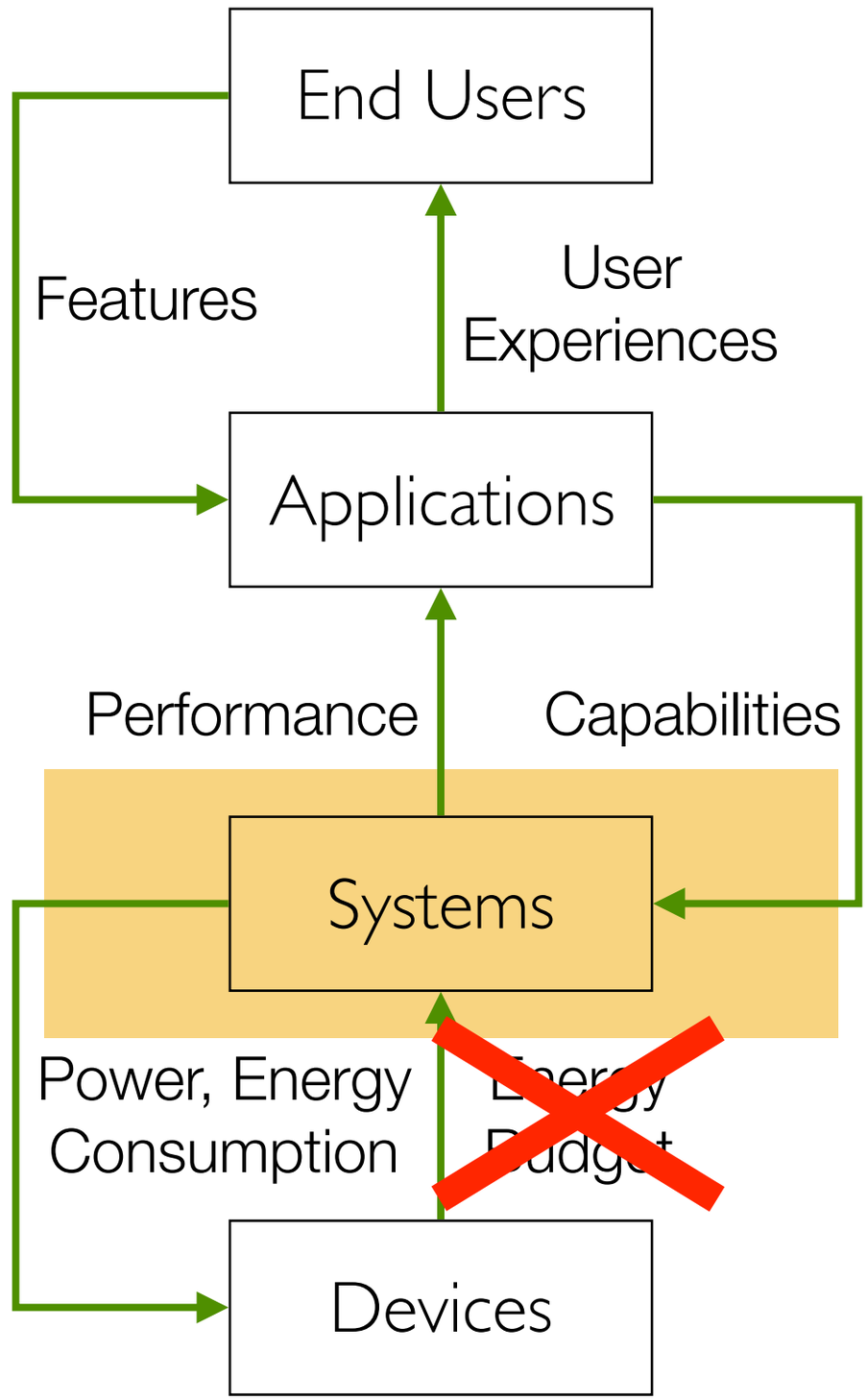
The Mobile Computing Virtuous Cycle



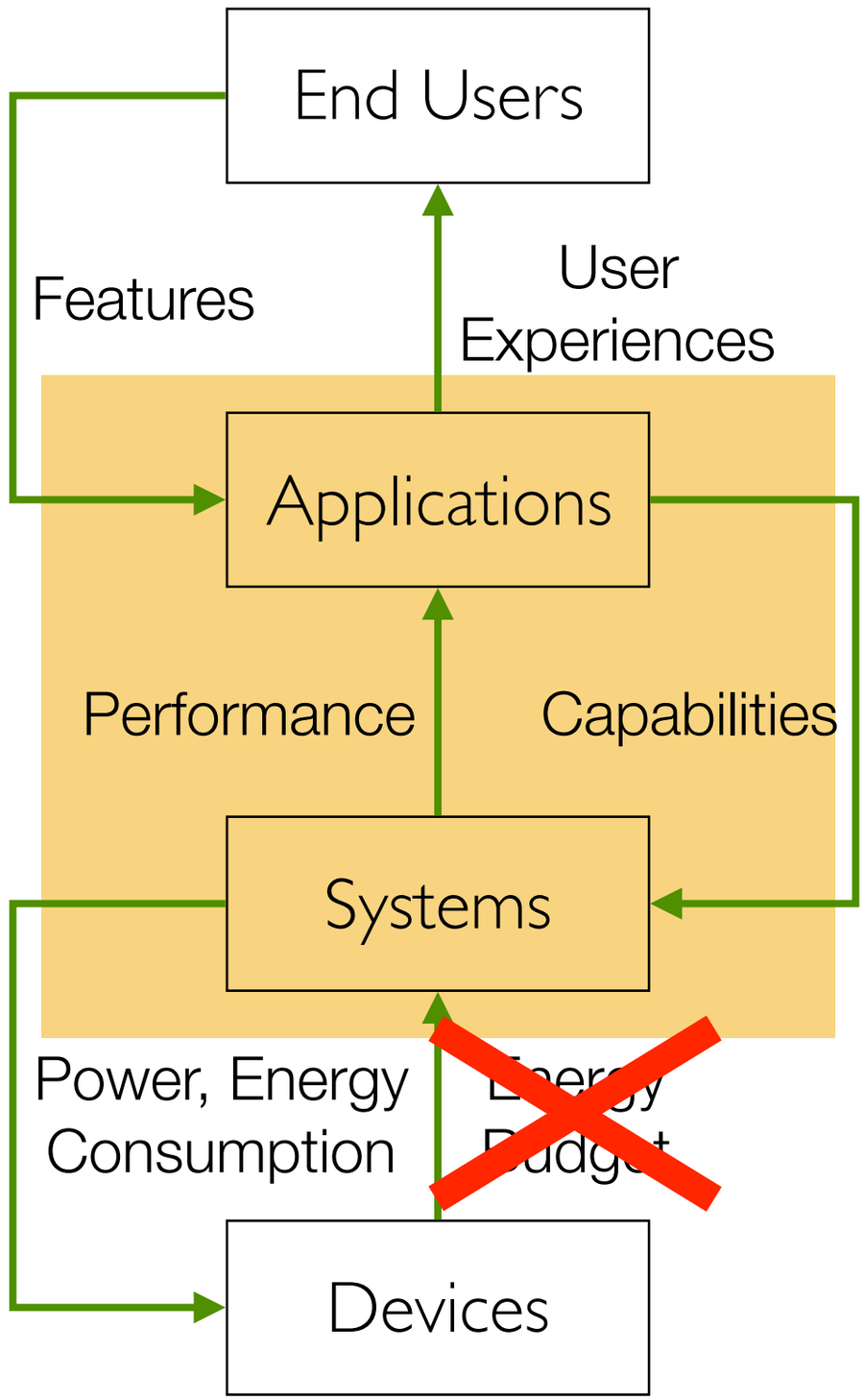
The Mobile Computing Virtuous Cycle



The Mobile Computing Virtuous Cycle



The Mobile Computing Virtuous Cycle



Mobile Applications

Web Mobile Applications

Web Mobile Applications

Web Mobile Applications

Mobile Web Applications

Mobile Web Computation Stack

Application

Runtime

Architecture

Mobile Web Computation Stack

Application

Runtime

Architecture

WebCore

Web-specific Architecture

[ISCA 2014]
[TOCS 2017]

Mobile Web Computation Stack

Application

GreenWeb
Language Support

[PLDI 2016]

Runtime

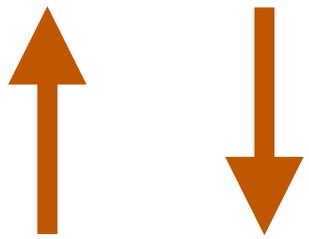
Architecture

WebCore
Web-specific Architecture

[ISCA 2014]
[TOCS 2017]

Mobile Web Computation Stack

Application



GreenWeb
Language Support

[PLDI 2016]

Runtime

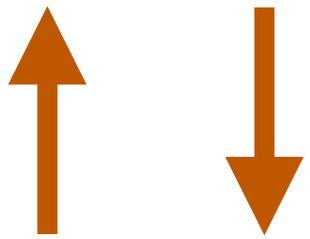
Architecture

WebCore
Web-specific Architecture

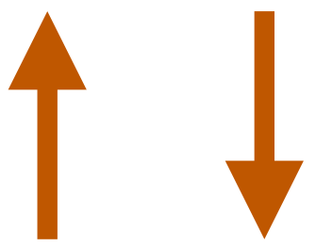
[ISCA 2014]
[TOCS 2017]

Mobile Web Computation Stack

Application



Runtime



Architecture

GreenWeb
Language Support

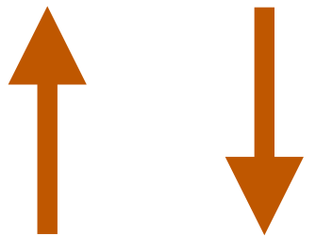
[PLDI 2016]

WebCore
Web-specific Architecture

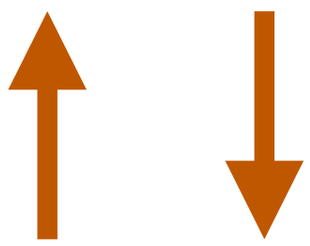
[ISCA 2014]
[TOCS 2017]

Mobile Web Computation Stack

Application



Runtime



Architecture

GreenWeb
Language Support

[PLDI 2016]

WebRT
Energy-aware
Web Runtime

[HPCA 2013]
[HPCA 2015]

WebCore
Web-specific Architecture

[ISCA 2014]
[TOCS 2017]

Mobile Web Computation Stack

Application

GreenWeb

Language Support

[PLDI 2016]

Runtime

WebRT

Energy-aware

Web Runtime

[HPCA 2013]

[HPCA 2015]

Architecture

WebCore

Web-specific Architecture

[ISCA 2014]

[TOCS 2017]

Put Web Applications in Hardware [ISCA 2014]

Put Web Applications in Hardware [ISCA 2014]

Successful Specialization Stories

FFT
GEMM
Codec
Encryption
Stencil Comp.
Molecular Dyn.
Sequence Algn.
Deep Learning
Ultrasound img.
…

Put Web Applications in Hardware [ISCA 2014]

Successful Specialization Stories

FFT
GEMM
Codec
Encryption
Stencil Comp.
Molecular Dyn.
Sequence Algn.
Deep Learning
Ultrasound img.
…

Challenging for the Web stack

Put Web Applications in Hardware [ISCA 2014]

Successful Specialization Stories

FFT
GEMM
Codec
Encryption
Stencil Comp.
Molecular Dyn.
Sequence Algn.
Deep Learning
Ultrasound img.
…

Challenging for the Web stack

Huge code base

Put Web Applications in Hardware [ISCA 2014]

Successful Specialization Stories

FFT
GEMM
Codec
Encryption
Stencil Comp.
Molecular Dyn.
Sequence Algn.
Deep Learning
Ultrasound img.
…

Challenging for the Web stack

Huge code base

- ☑ H264 codec: 0.13M LoC, 6 languages
- ☒ Chrome: 17M LoC, 29 languages

Put Web Applications in Hardware [ISCA 2014]

Successful Specialization Stories

FFT
GEMM
Codec
Encryption
Stencil Comp.
Molecular Dyn.
Sequence Algn.
Deep Learning
Ultrasound img.
…

Challenging for the Web stack

Huge code base

- ☑ H264 codec: 0.13M LoC, 6 languages
- ☒ Chrome: 17M LoC, 29 languages

No regular execution pattern

Put Web Applications in Hardware [ISCA 2014]

Successful Specialization Stories

FFT
GEMM
Codec
Encryption
Stencil Comp.
Molecular Dyn.
Sequence Algn.
Deep Learning
Ultrasound img.
…

Challenging for the Web stack

Huge code base

- ☑ H264 codec: 0.13M LoC, 6 languages
- ☒ Chrome: 17M LoC, 29 languages

No regular execution pattern

- ☑ FFT: Fine-grained parallelism

Put Web Applications in Hardware [ISCA 2014]

Successful Specialization Stories

FFT
GEMM
Codec
Encryption
Stencil Comp.
Molecular Dyn.
Sequence Algn.
Deep Learning
Ultrasound img.
…

Challenging for the Web stack

Huge code base

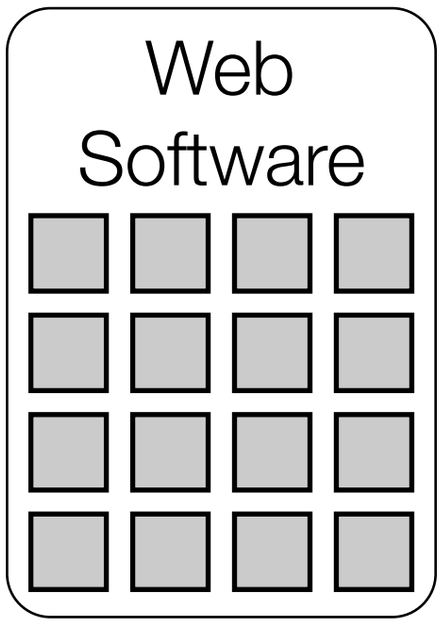
- ☑ H264 codec: 0.13M LoC, 6 languages
- ☒ Chrome: 17M LoC, 29 languages

No regular execution pattern

- ☑ FFT: Fine-grained parallelism
- ☒ Web: Typically task level (if any)

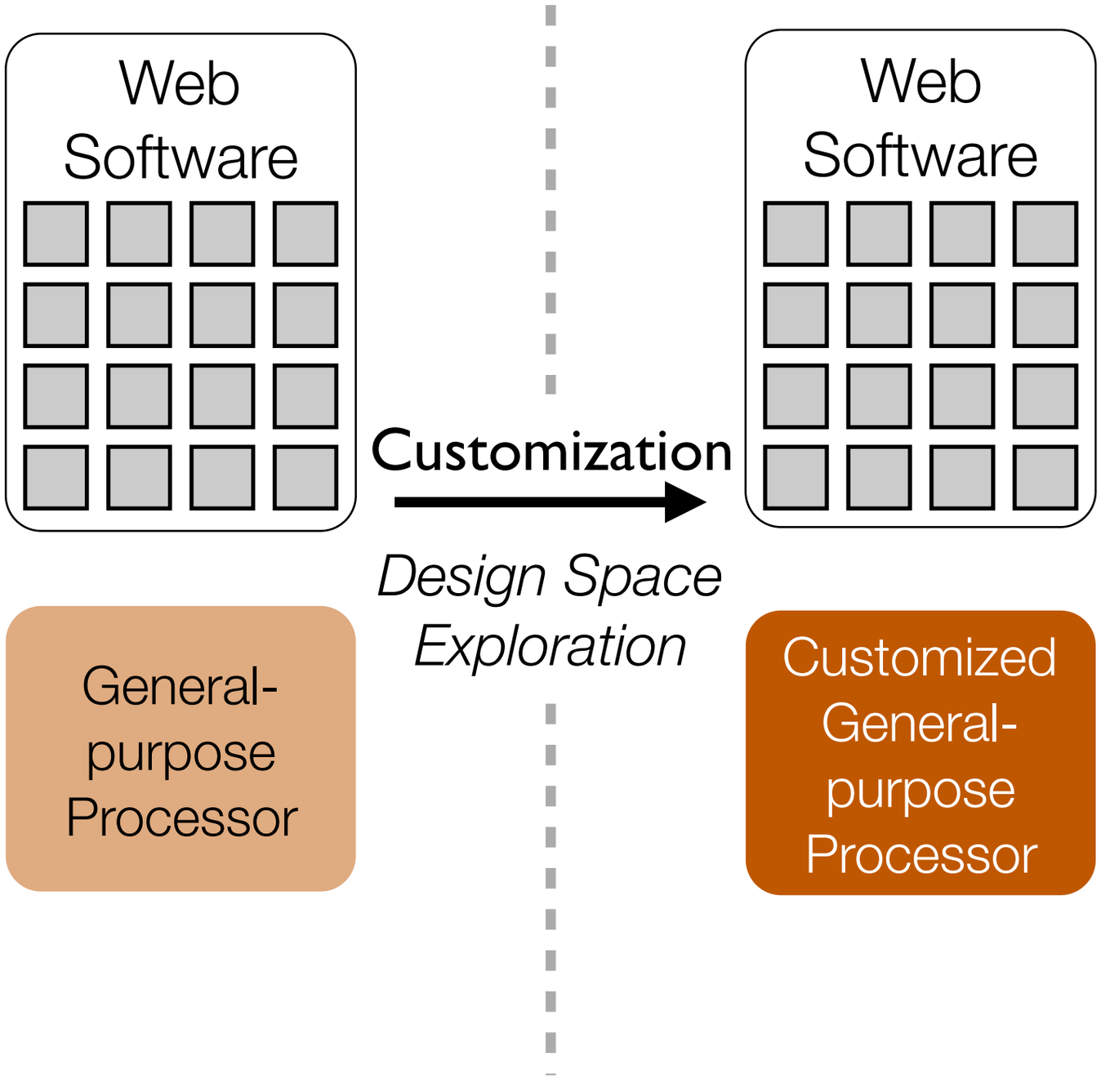
WebCore Philosophy: Step by Step

WebCore Philosophy: Step by Step

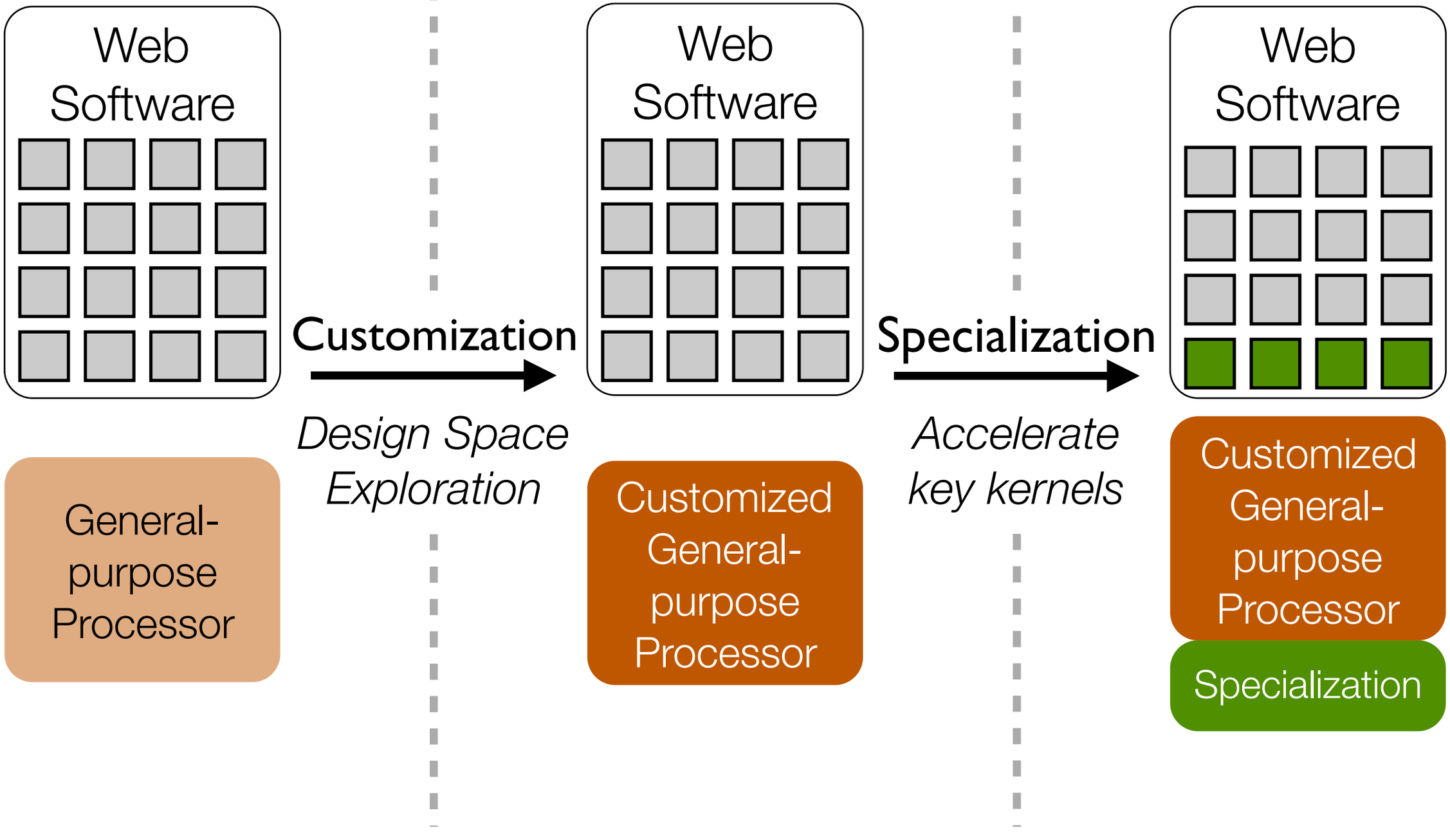


General-purpose Processor

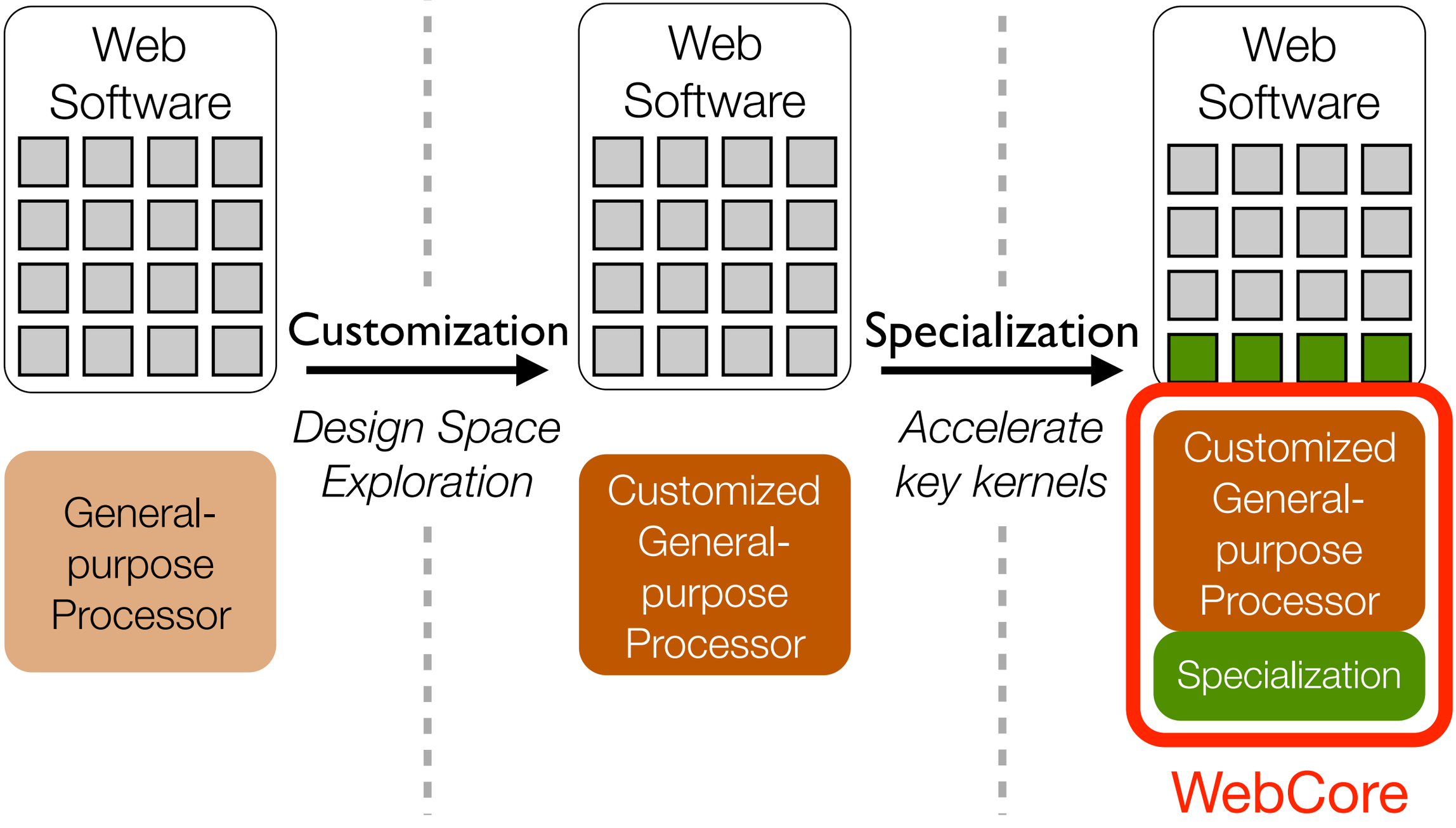
WebCore Philosophy: Step by Step



WebCore Philosophy: Step by Step



WebCore Philosophy: Step by Step



WebCore Customization

WebCore Customization

What is an Ideal General Purpose Architecture for Mobile Web Computing?

WebCore Customization

What is an Ideal General Purpose Architecture for Mobile Web Computing?

Macro-architecture: out-of-order cores or in-order cores?

WebCore Customization

What is an Ideal General Purpose Architecture for Mobile Web Computing?

Macro-architecture: out-of-order cores or in-order cores?

Micro-architecture: are *existing* designs already ideal?

Parameters
Issue width
Functional units
Load queue size
Store queue size
Branch prediction size
ROB size
Physical registers
L1 I-cache size
L1 I-cache delay
L1 D-cache size
L1 D-cache delay
L2 cache size
L2 cache delay

WebCore Customization

What is an Ideal General Purpose Architecture for Mobile Web Computing?

Macro-architecture: out-of-order cores or in-order cores?

Micro-architecture: are *existing* designs already ideal?

Design Space Exploration

Parameters
Issue width
Functional units
Load queue size
Store queue size
Branch prediction size
ROB size
Physical registers
L1 I-cache size
L1 I-cache delay
L1 D-cache size
L1 D-cache delay
L2 cache size
L2 cache delay

WebCore Customization

What is an Ideal General Purpose Architecture for Mobile Web Computing?

Macro-architecture: out-of-order cores or in-order cores?

Micro-architecture: are *existing* designs already ideal?

Design Space Exploration

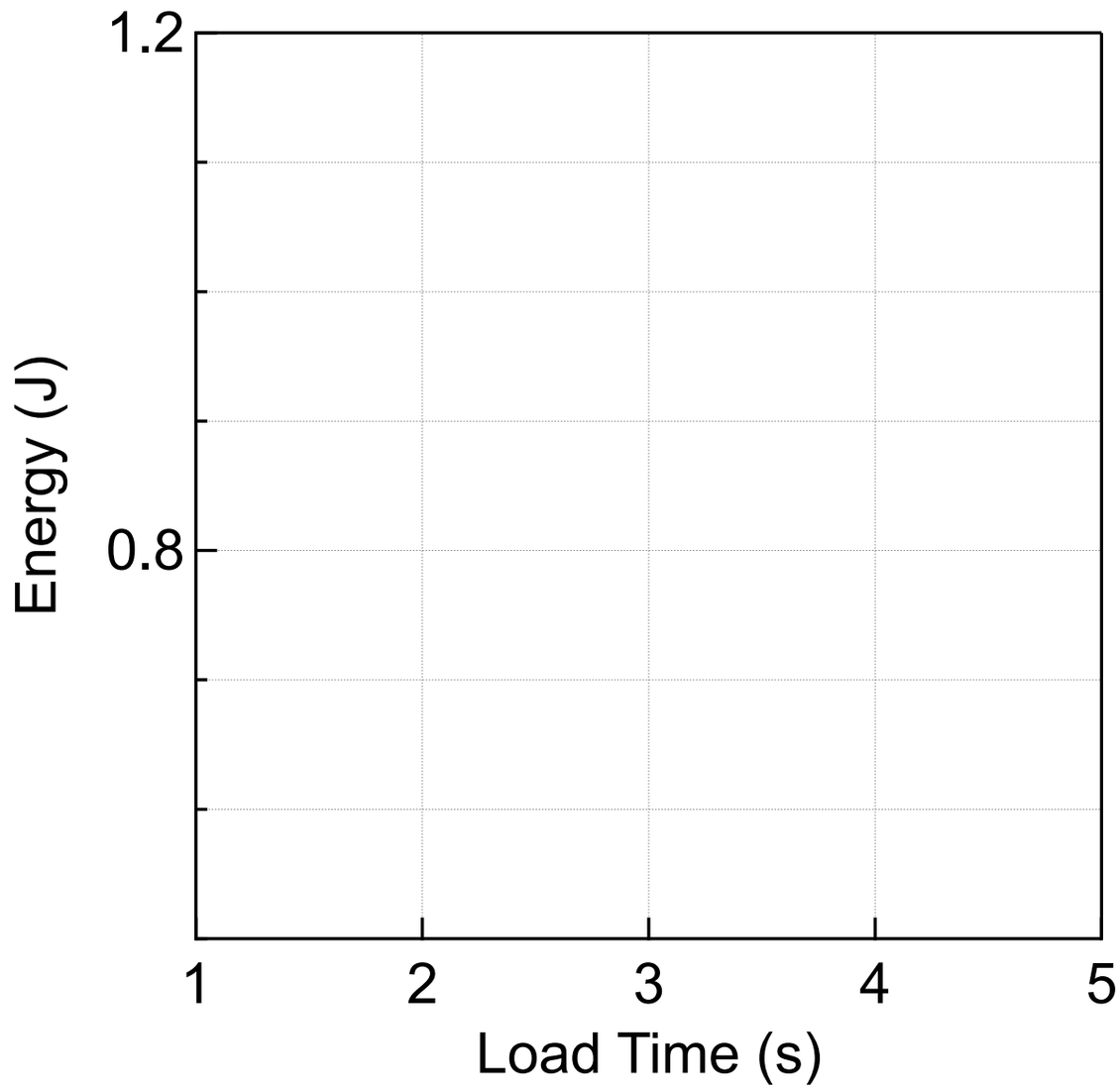
Hyperparameter tuning

Parameters
Issue width
Functional units
Load queue size
Store queue size
Branch prediction size
ROB size
Physical registers
L1 I-cache size
L1 I-cache delay
L1 D-cache size
L1 D-cache delay
L2 cache size
L2 cache delay

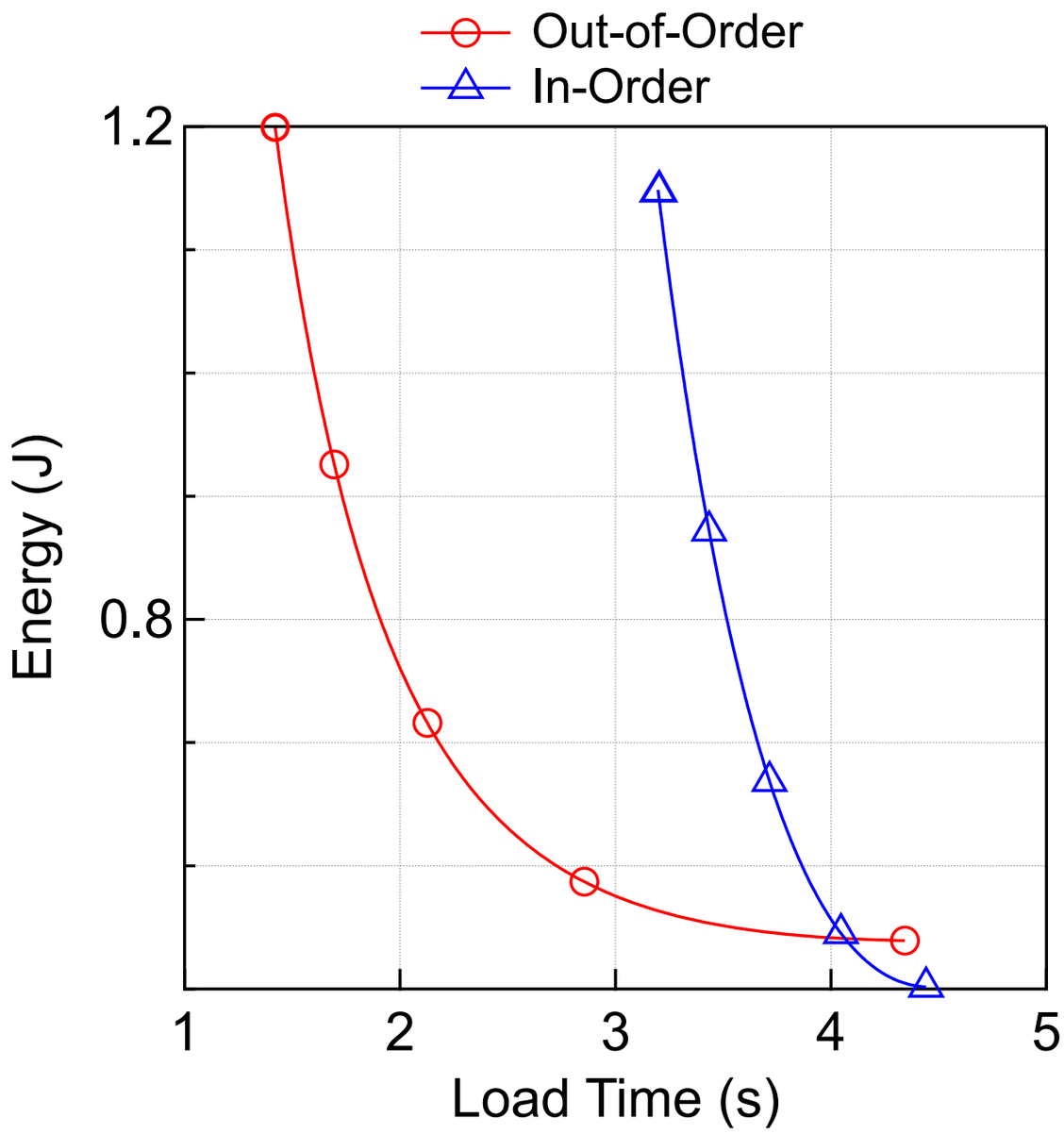
WebCore Customization 3 Billions Design Points

Parameters	Measure	Range
Issue width	count	1,2,4
# Functional units	count	1::1::4
Load queue size	# entries	4::4::16
Store queue size	# entries	4::4::16
Branch prediction size	log_2(#entries)	1::1::10
ROB size	# entries	8::8::128
# Physical registers	# entries	5::5::140
L1 I-cache size	log_2(KB)	3::1::8
L1 I-cache delay	cycles	1::1::3
L1 D-cache size	log_2(KB)	3::1::8
L1 D-cache delay	cycles	1::1::3
L2 cache size	log_2(KB)	7::1::10
L2 cache delay	cycles	16,32,64

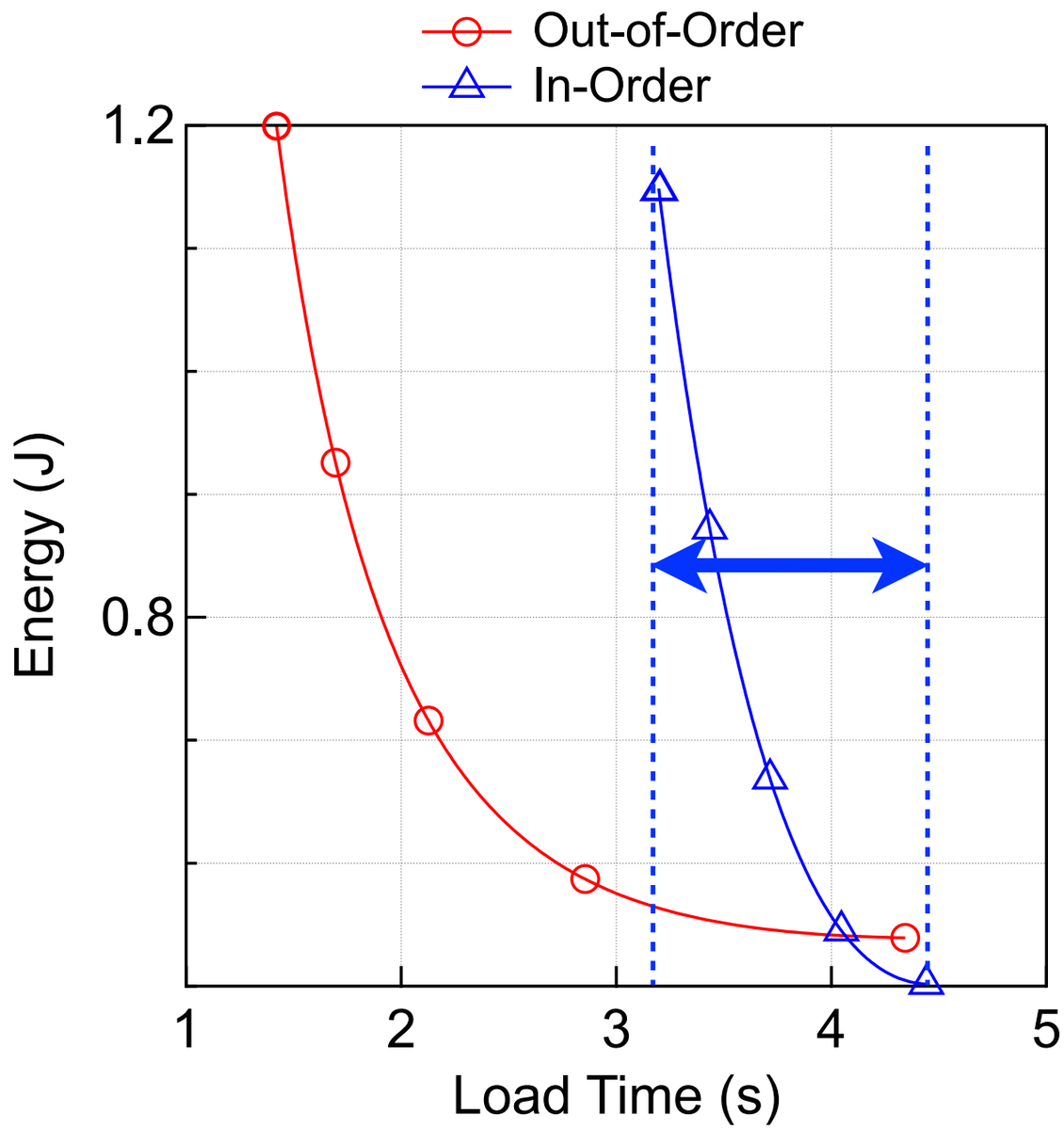
Design Space Exploration (DSE) Insights



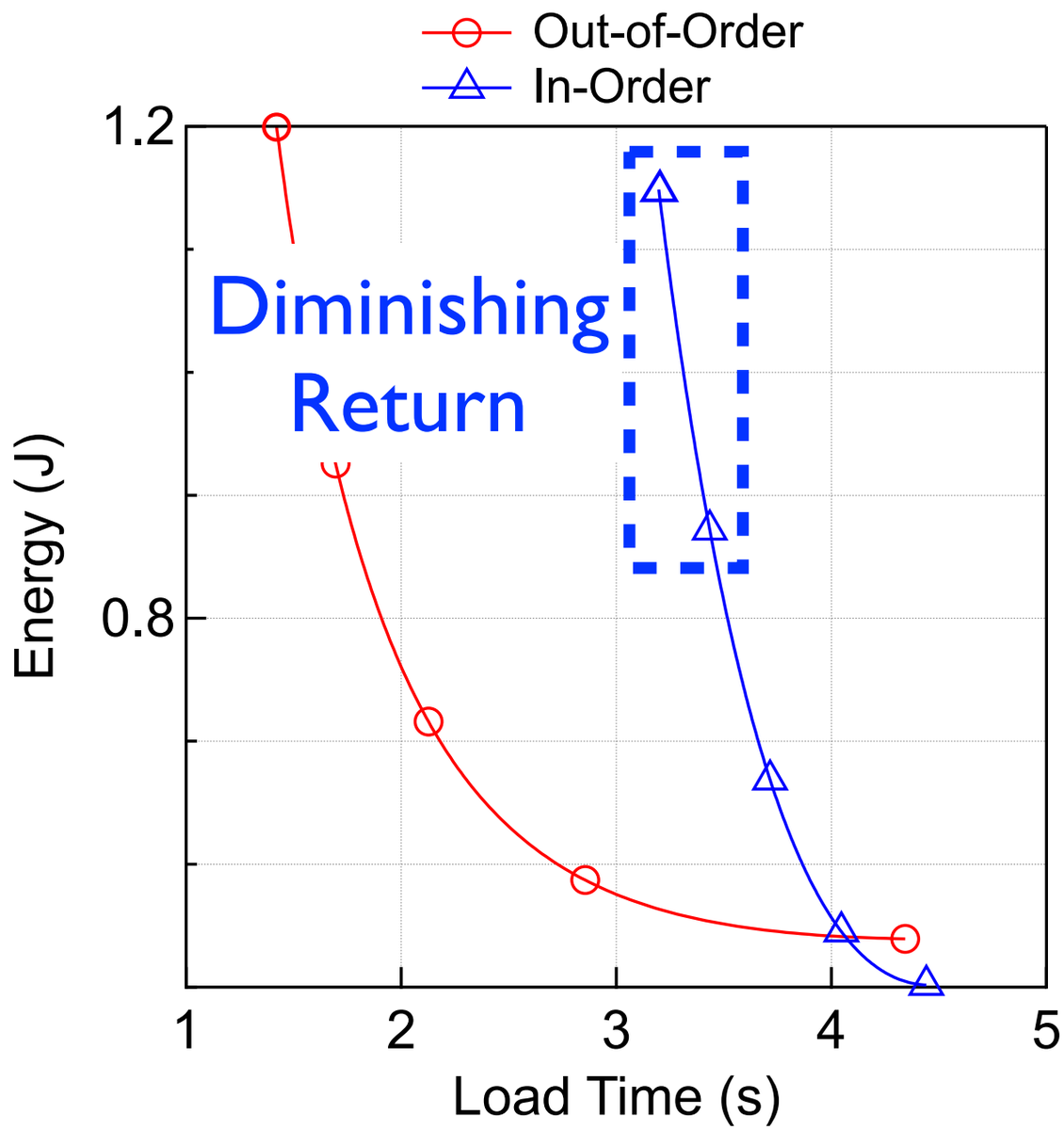
Design Space Exploration (DSE) Insights



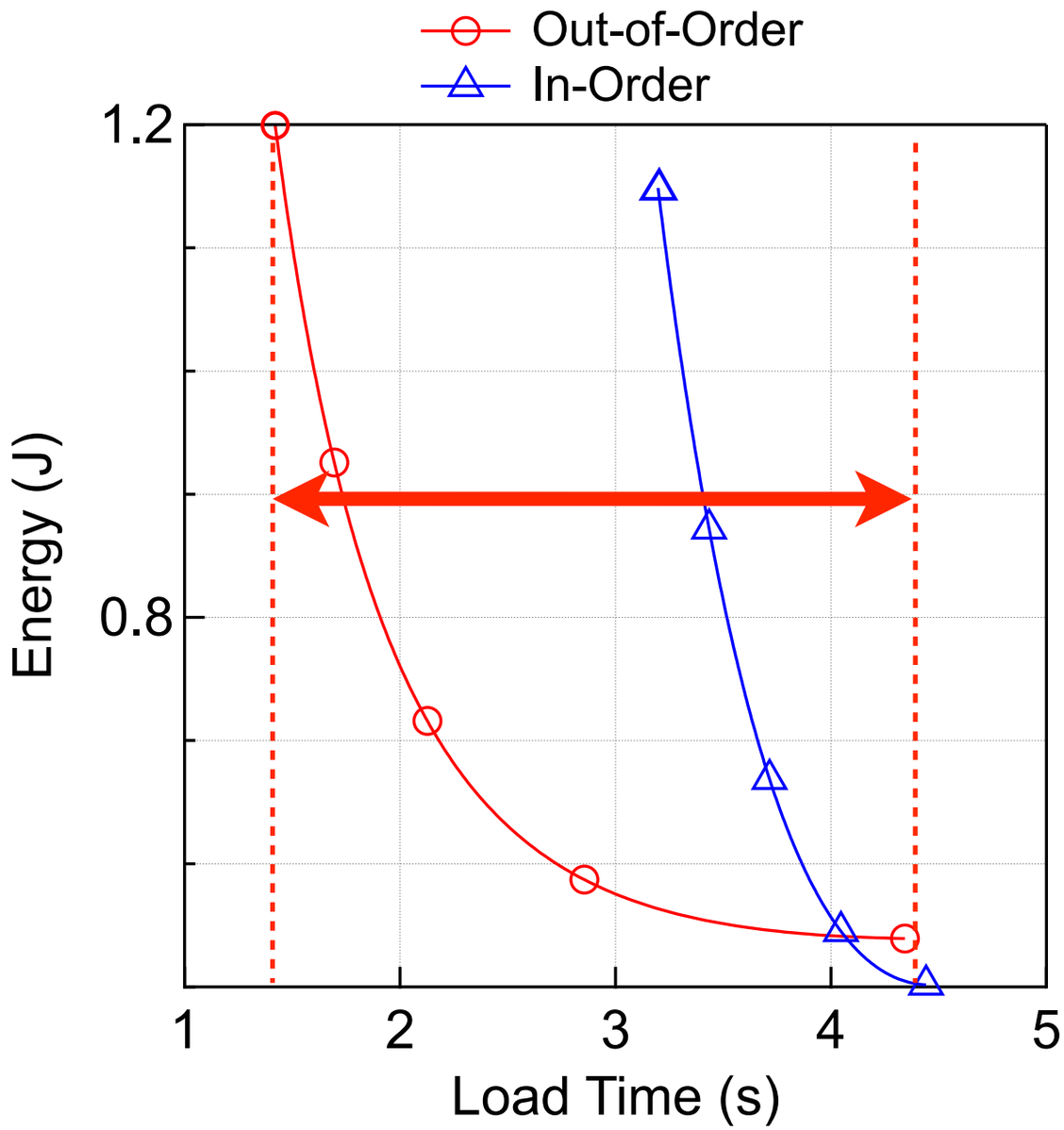
Design Space Exploration (DSE) Insights



Design Space Exploration (DSE) Insights



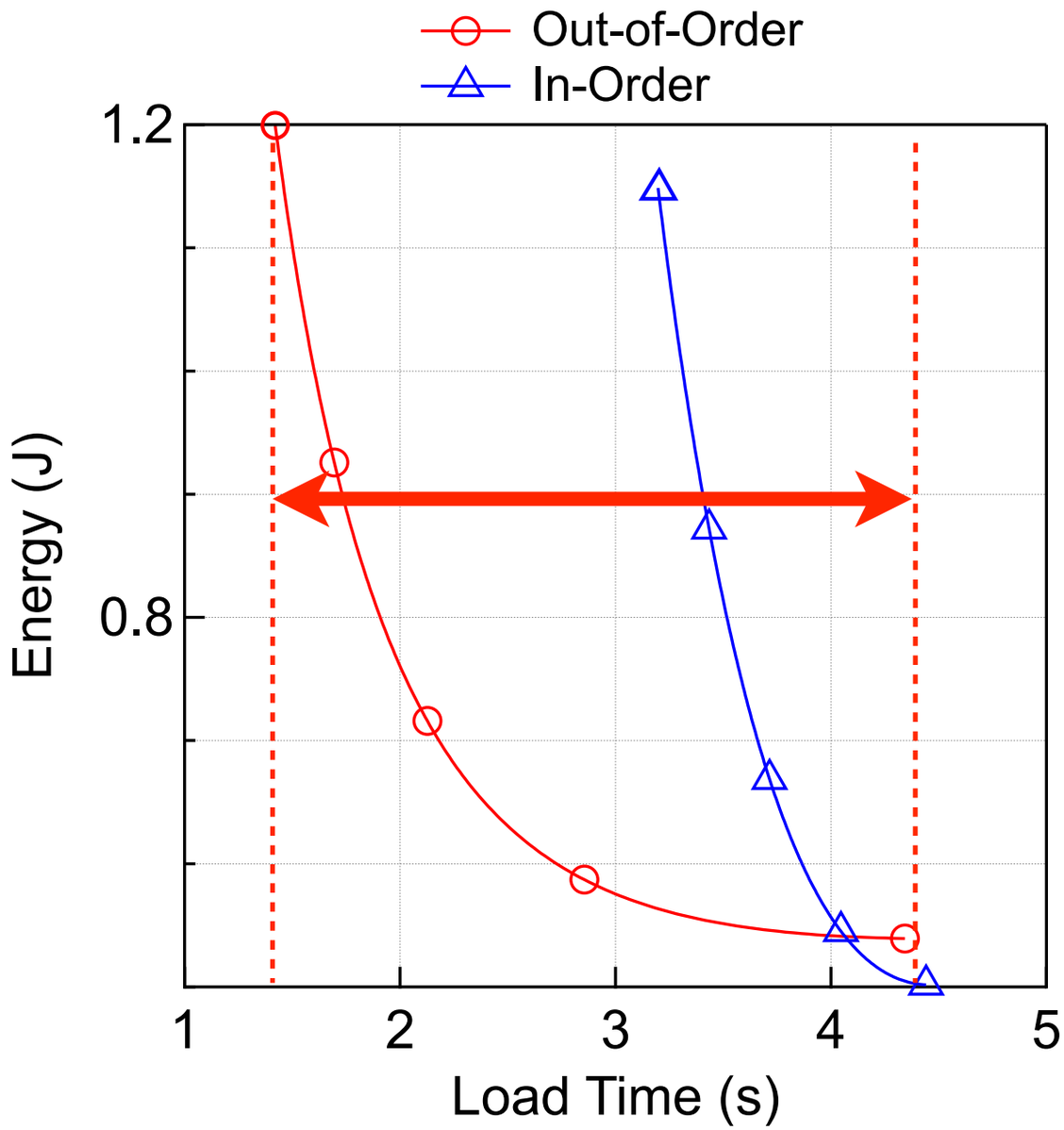
Design Space Exploration (DSE) Insights



Design Space Exploration (DSE) Insights

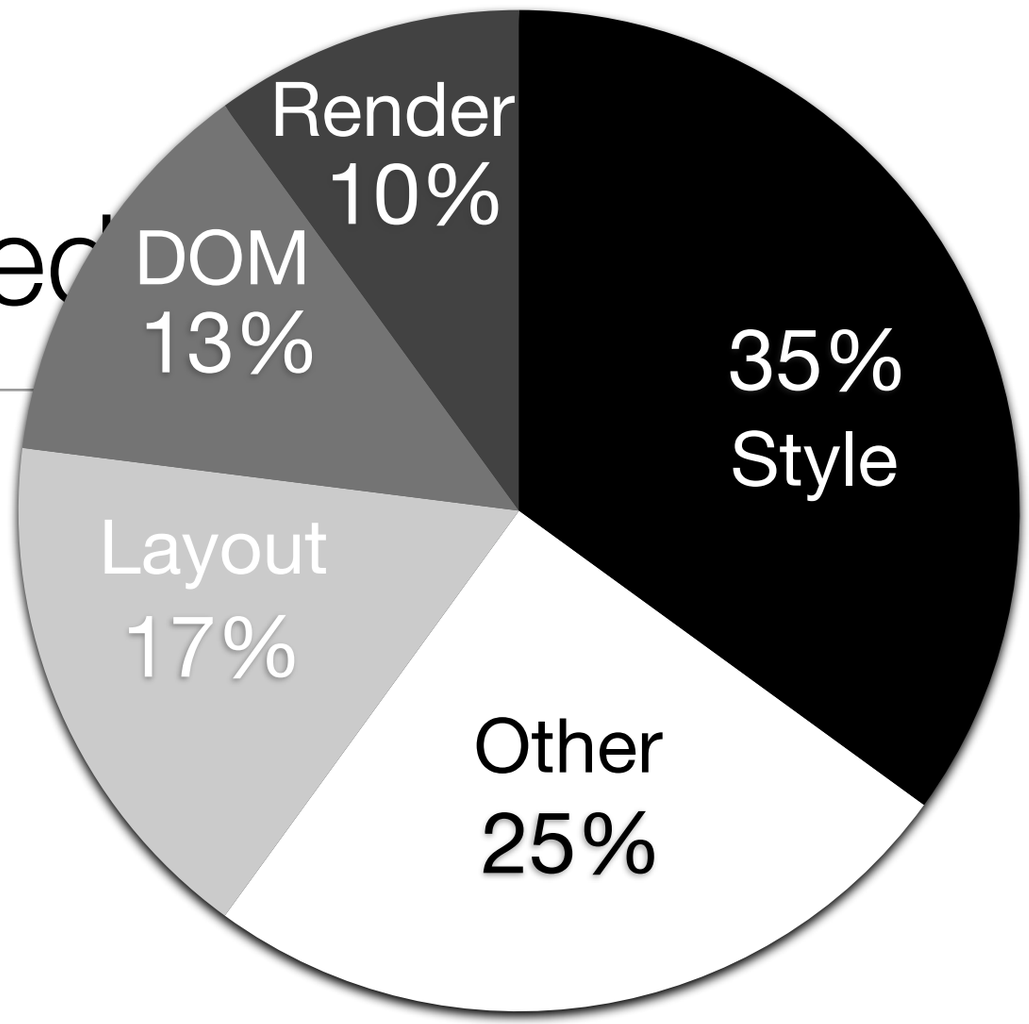
DSE Insights

1. Out-of-order designs provide a better architecture substrate for the mobile Web.



Leveraging Kernel Knowledge

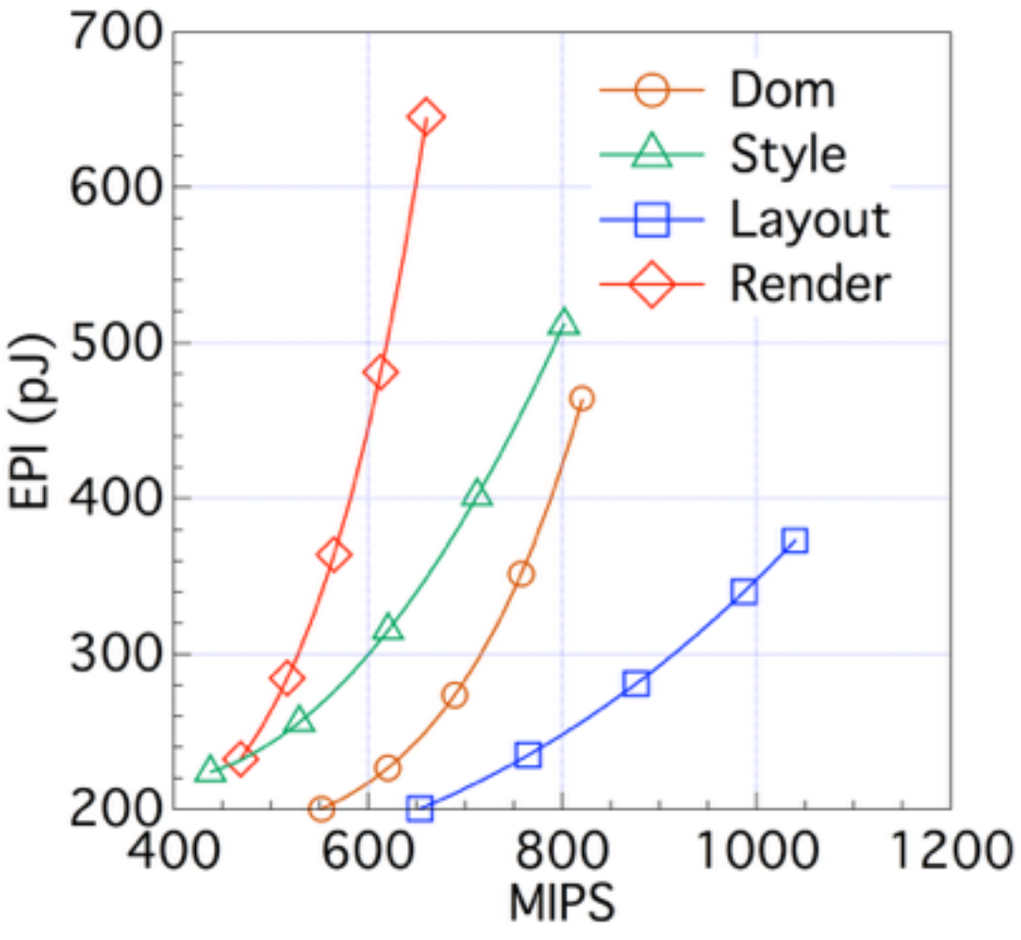
Leveraging Kernel Knowled



Leveraging Kernel Knowledge

In-order design

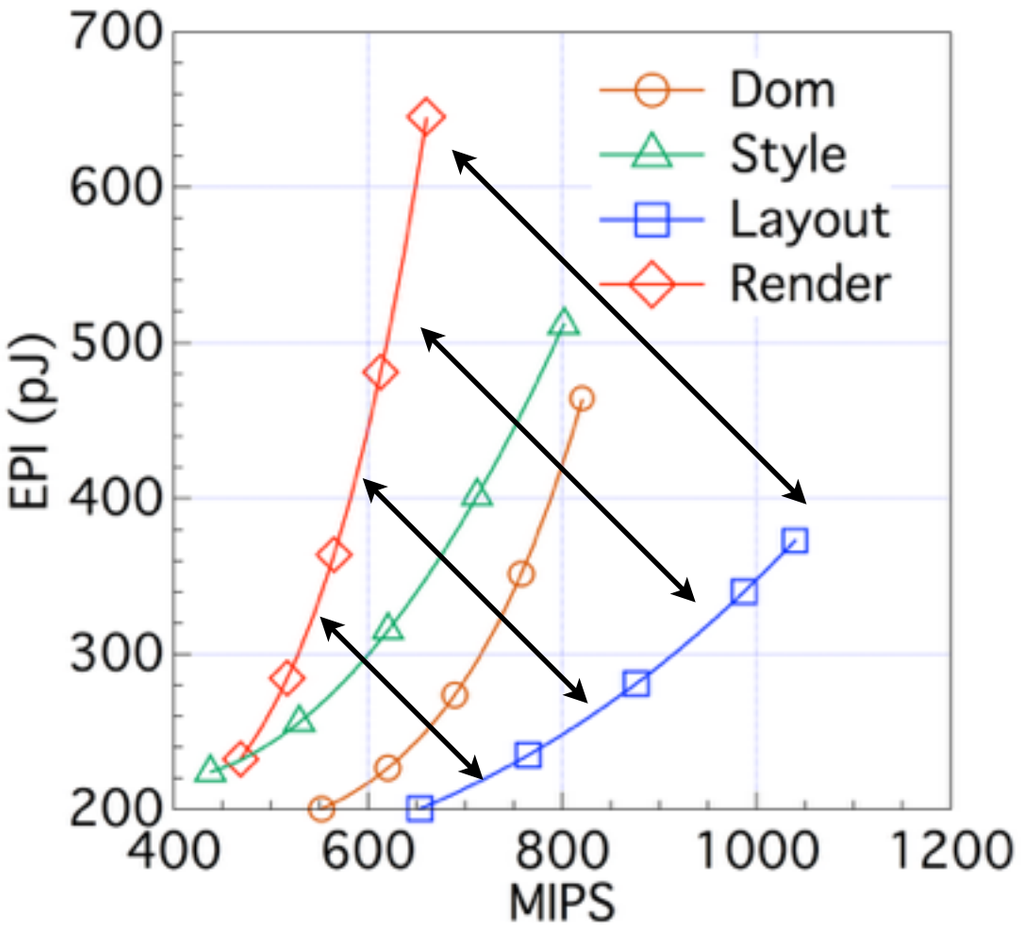
Leveraging Kernel Knowledge



In-order design

Leveraging Kernel Knowledge

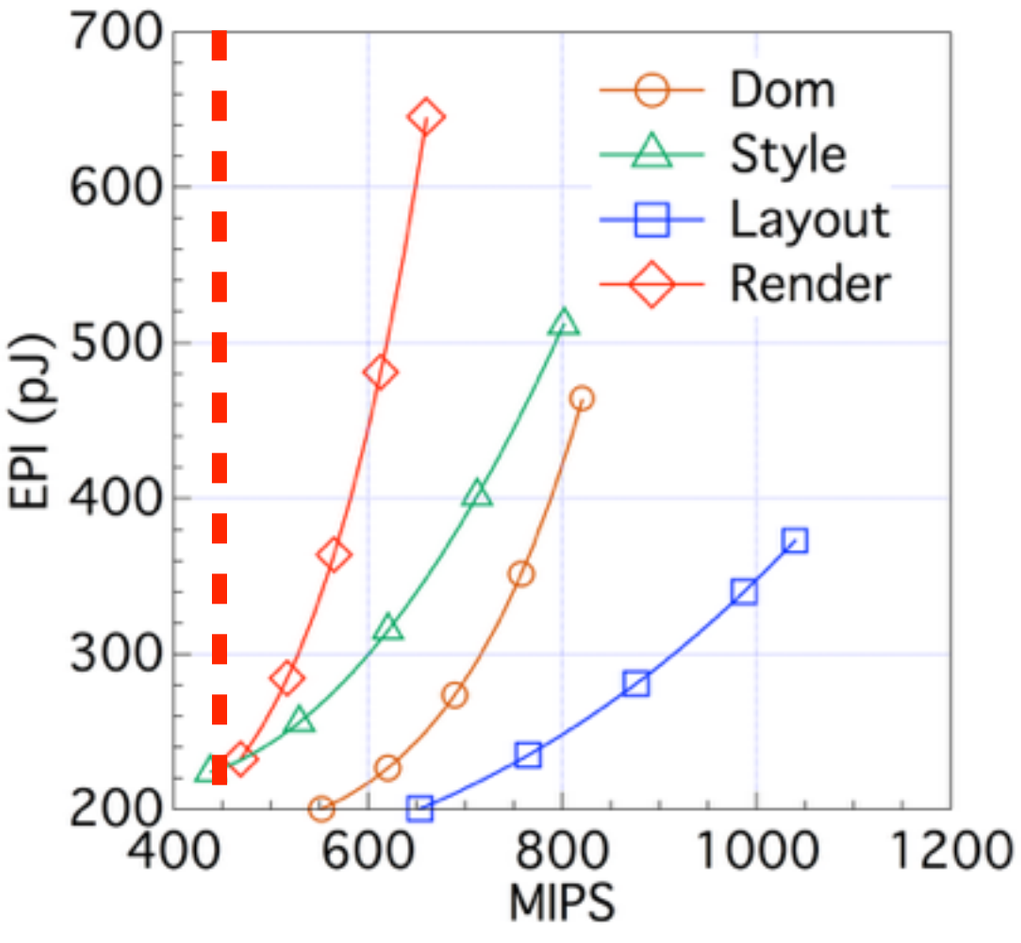
In-order designs show strong kernel variance.



In-order design

Leveraging Kernel Knowledge

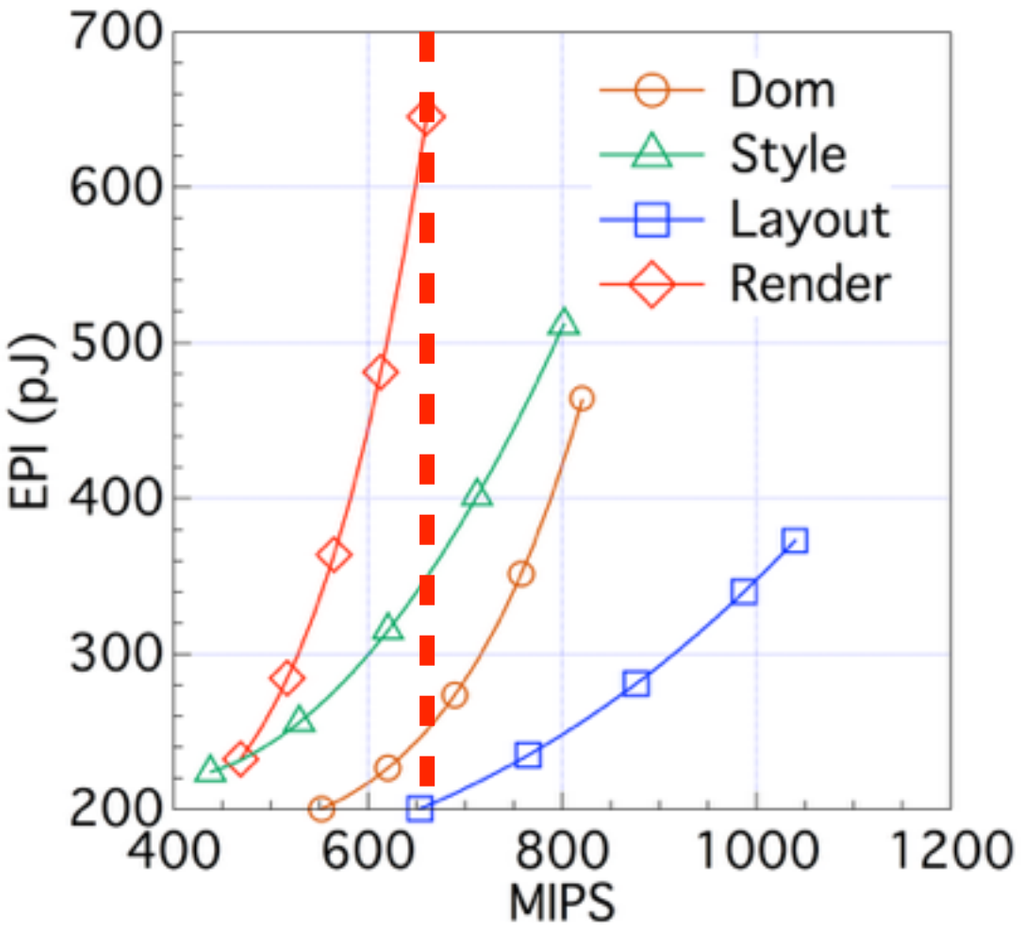
In-order designs show strong kernel variance.



In-order design

Leveraging Kernel Knowledge

In-order designs show strong kernel variance.

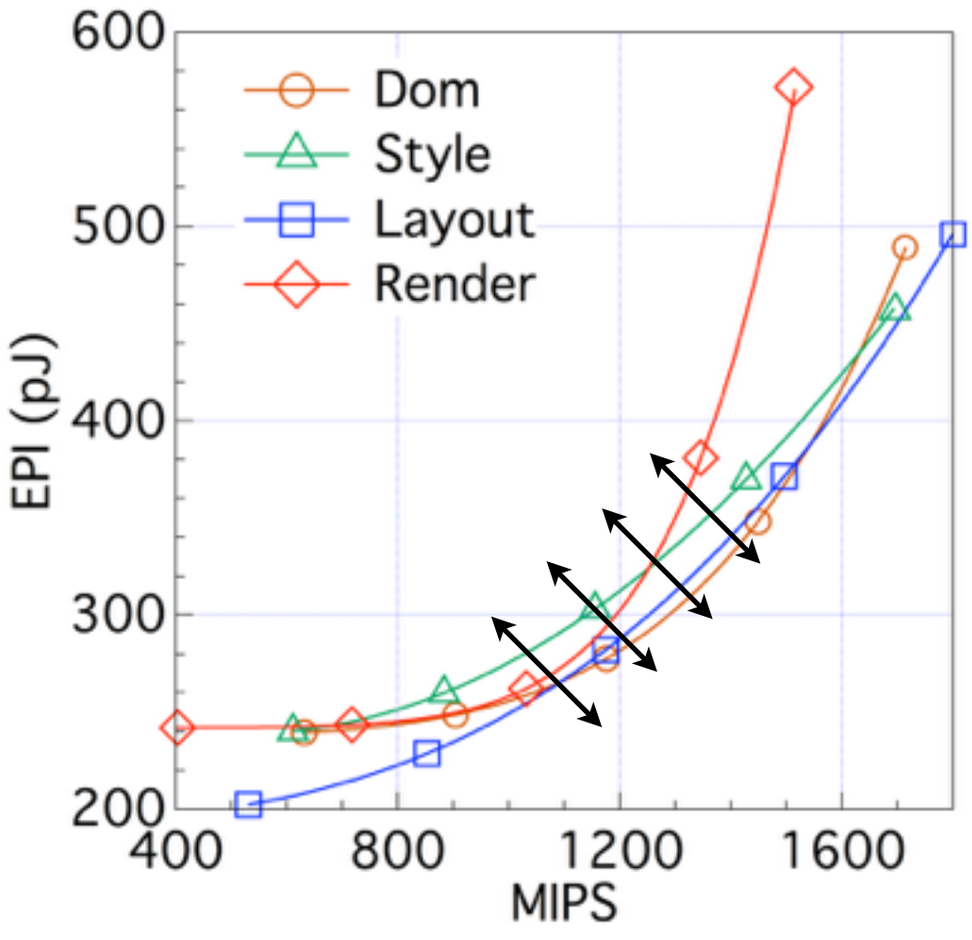


In-order design

Leveraging Kernel Knowledge

In-order designs show strong kernel variance.

In-order design



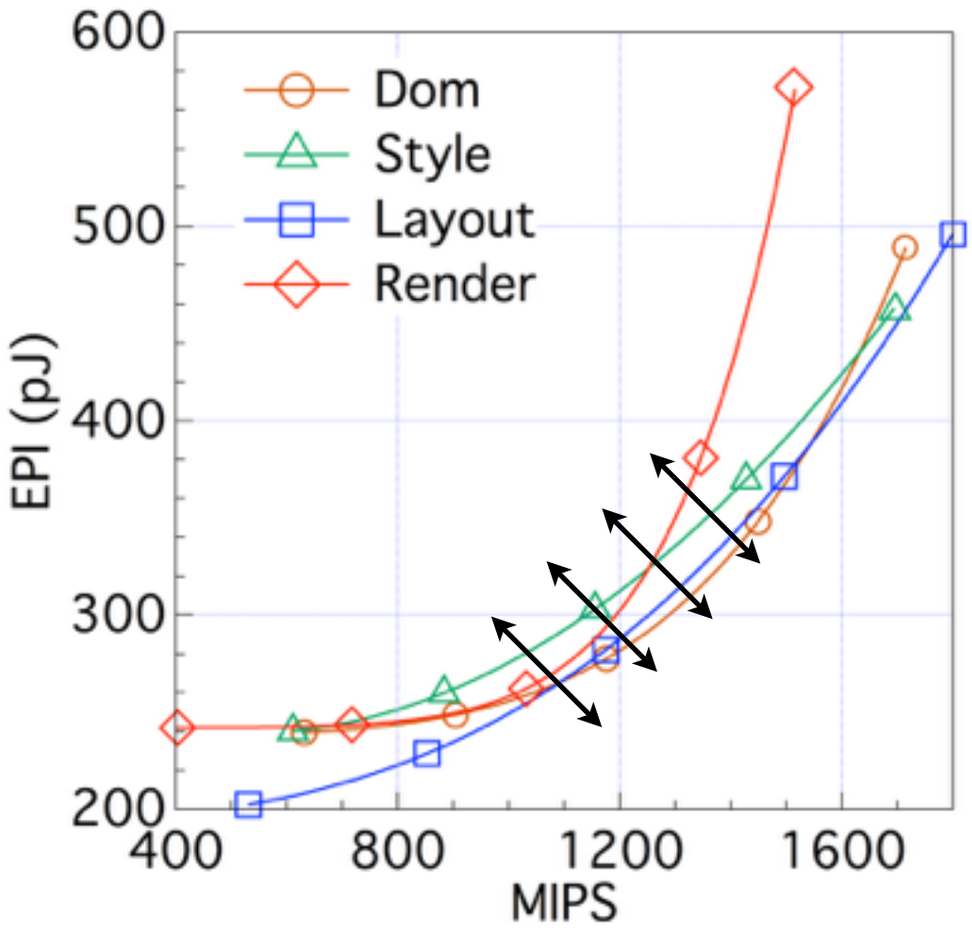
Out-of-order design

Leveraging Kernel Knowledge

In-order designs show strong kernel variance.

Out-of-order designs can accommodate kernel variance.

In-order design

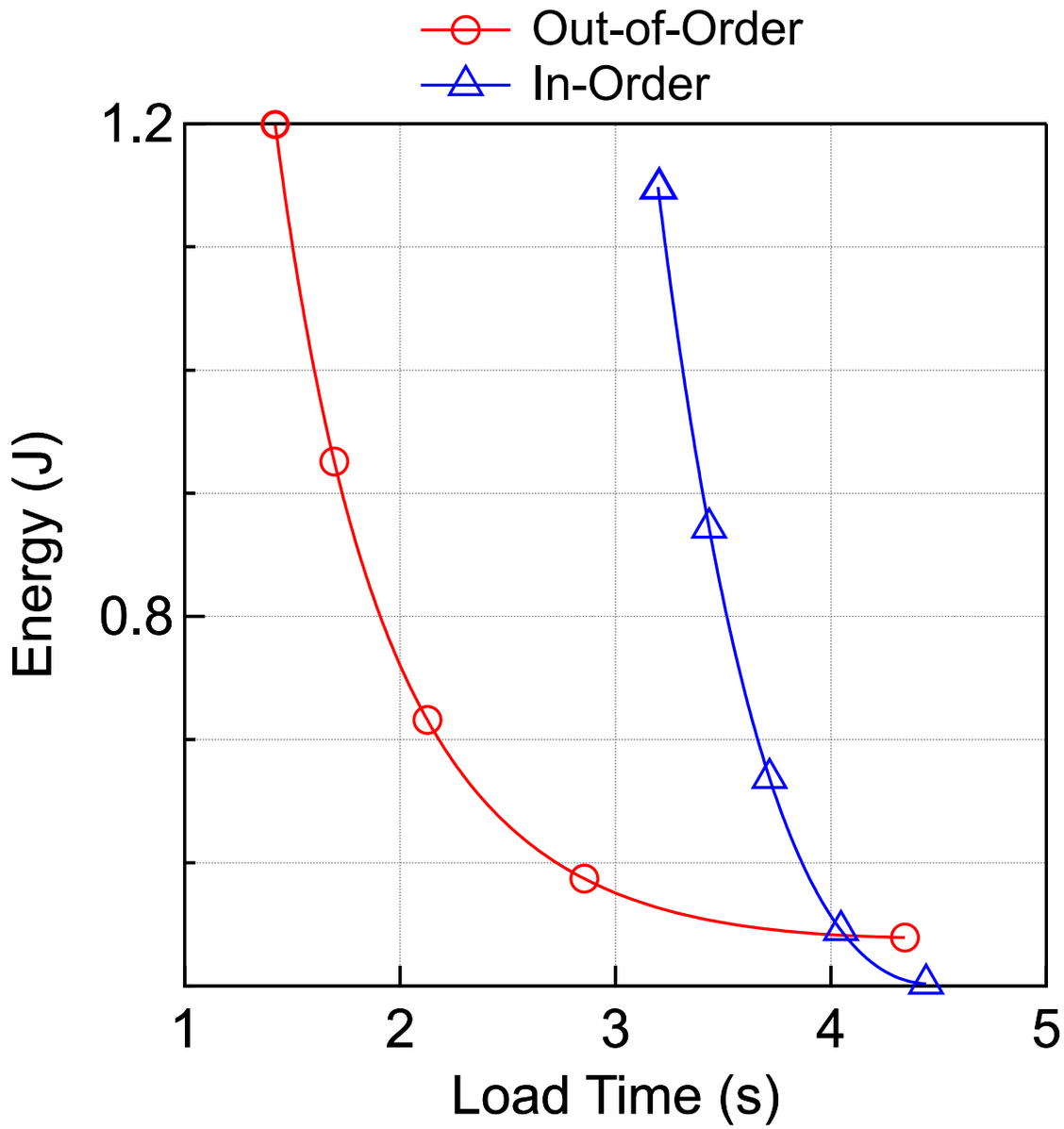


Out-of-order design

Design Space Exploration (DSE) Insights

DSE Insights

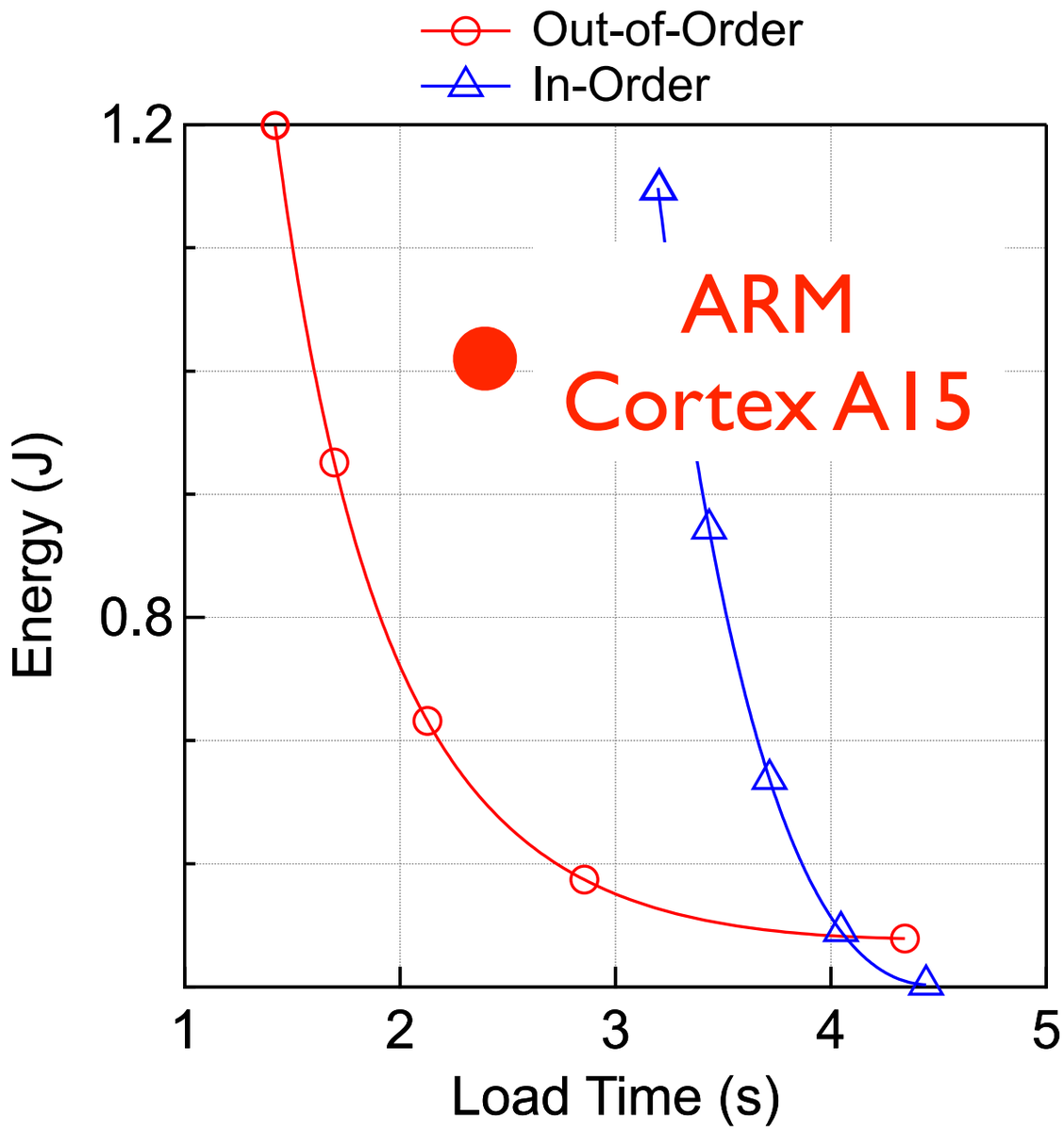
1. Out-of-order designs provide a better architecture substrate for the mobile Web.



Design Space Exploration (DSE) Insights

DSE Insights

1. Out-of-order designs provide a better architecture substrate for the mobile Web.

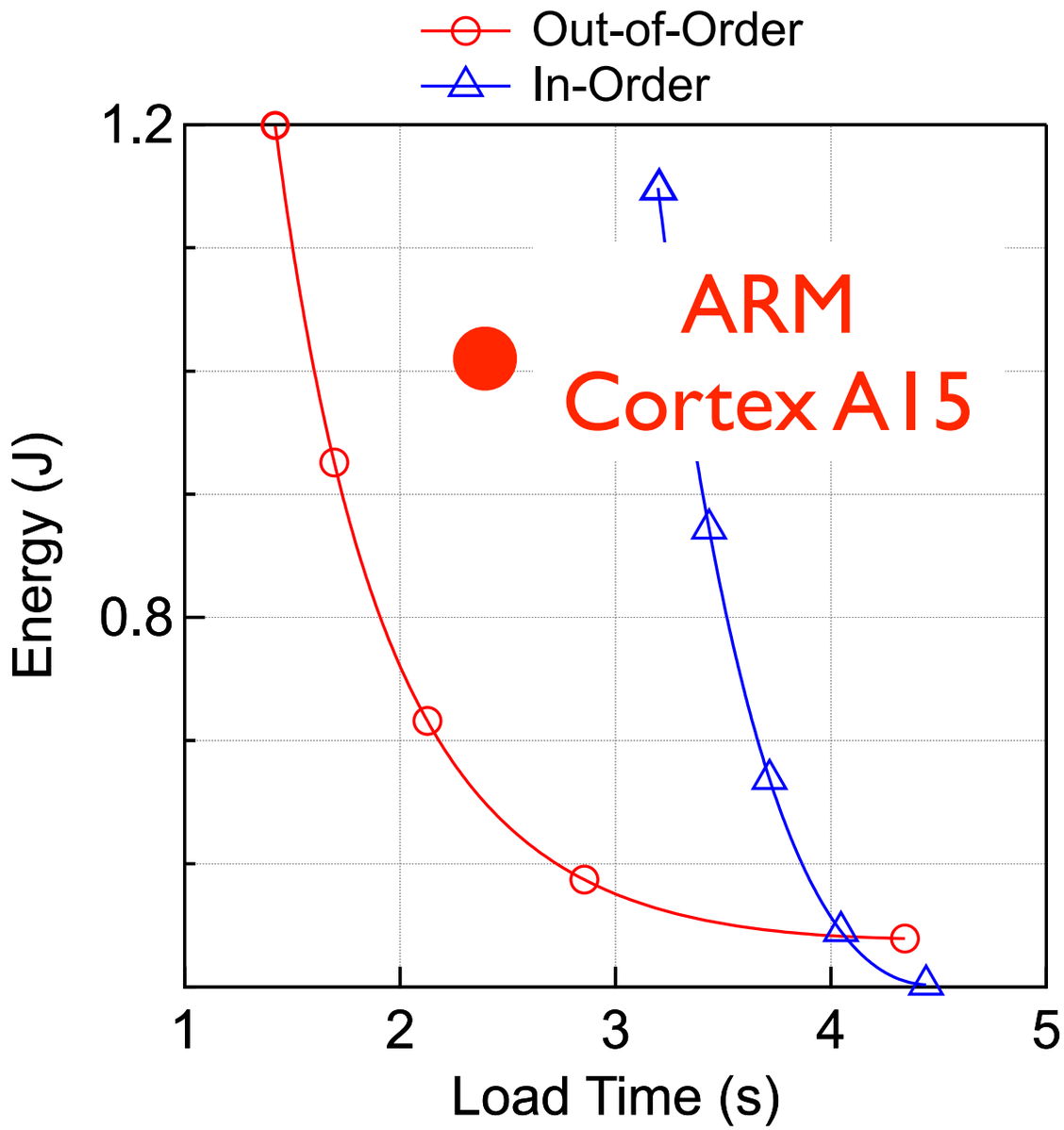


Design Space Exploration (DSE) Insights

DSE Insights

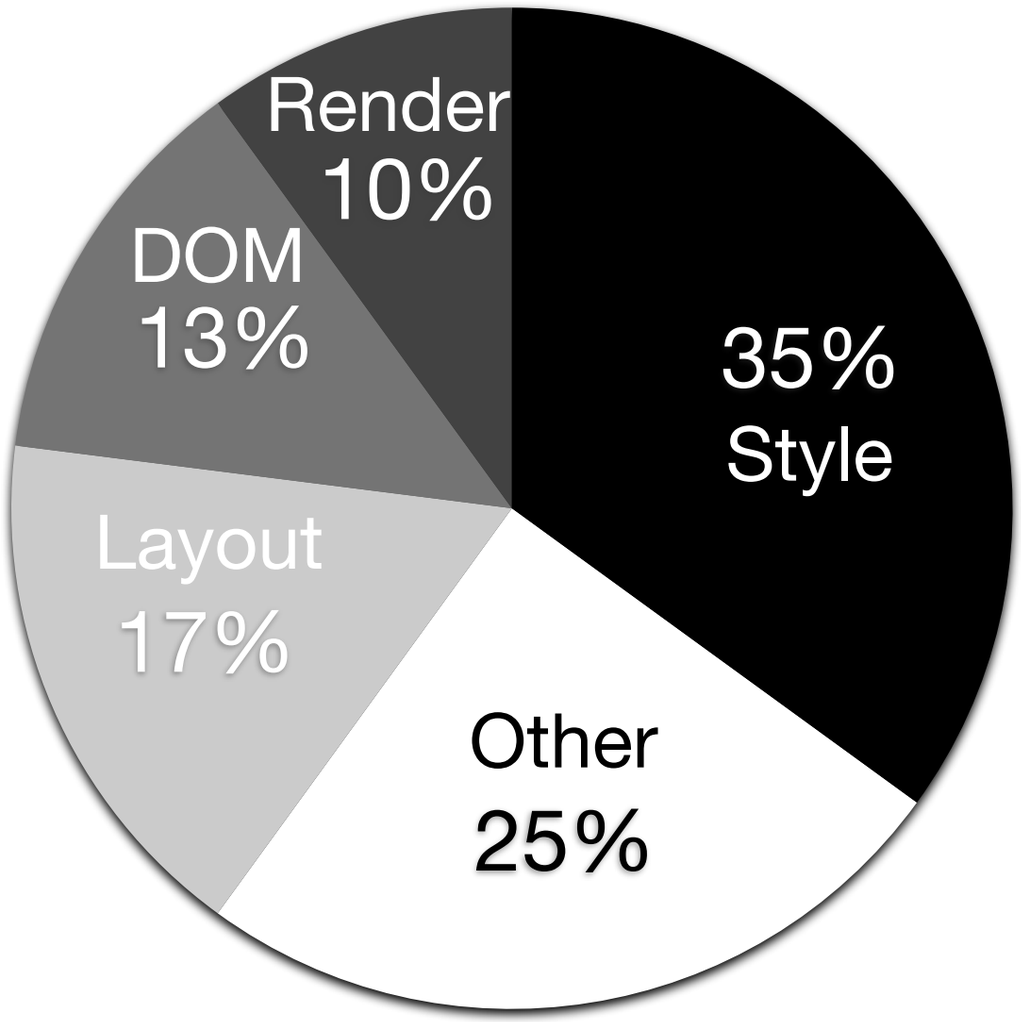
1. Out-of-order designs provide a better architecture substrate for the mobile Web.

2. Existing mobile CPUs are *not* optimized for the mobile Web.

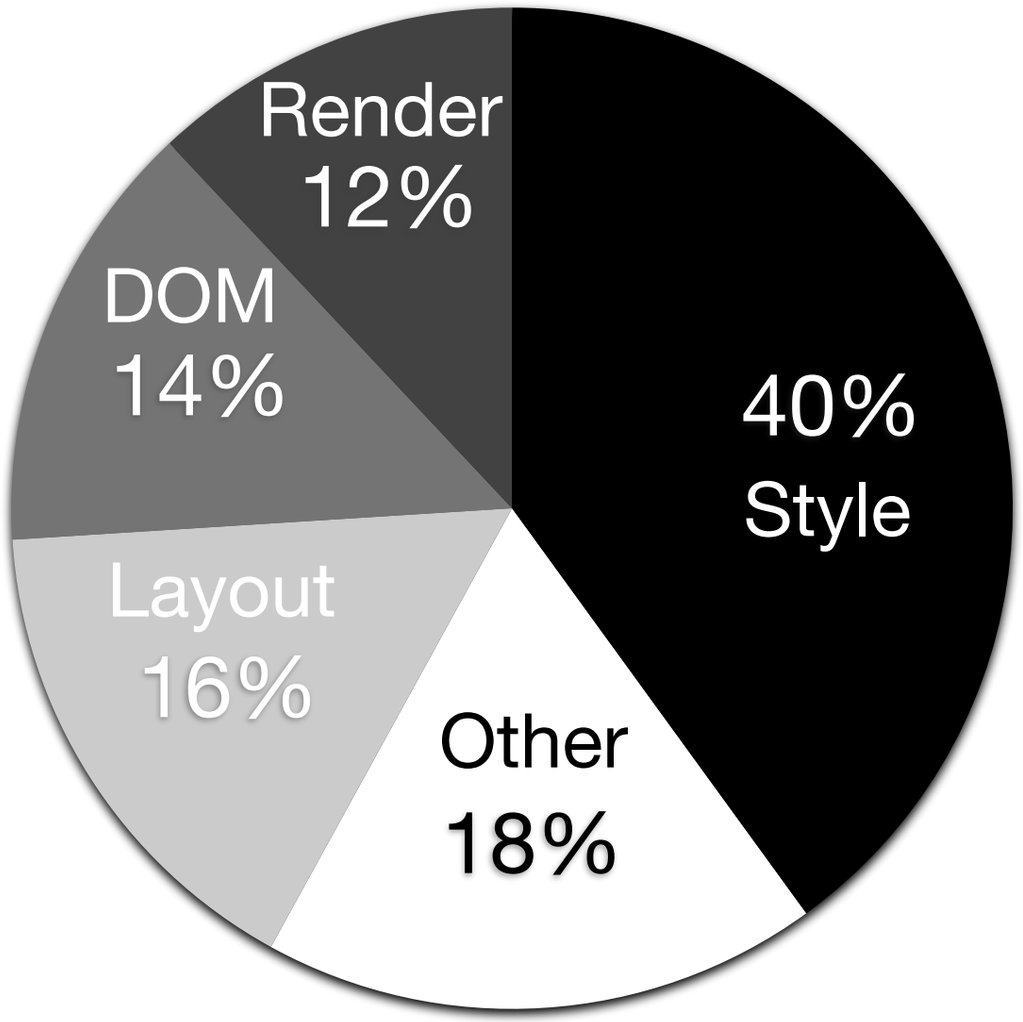


WebCore Specialization

WebCore Specialization

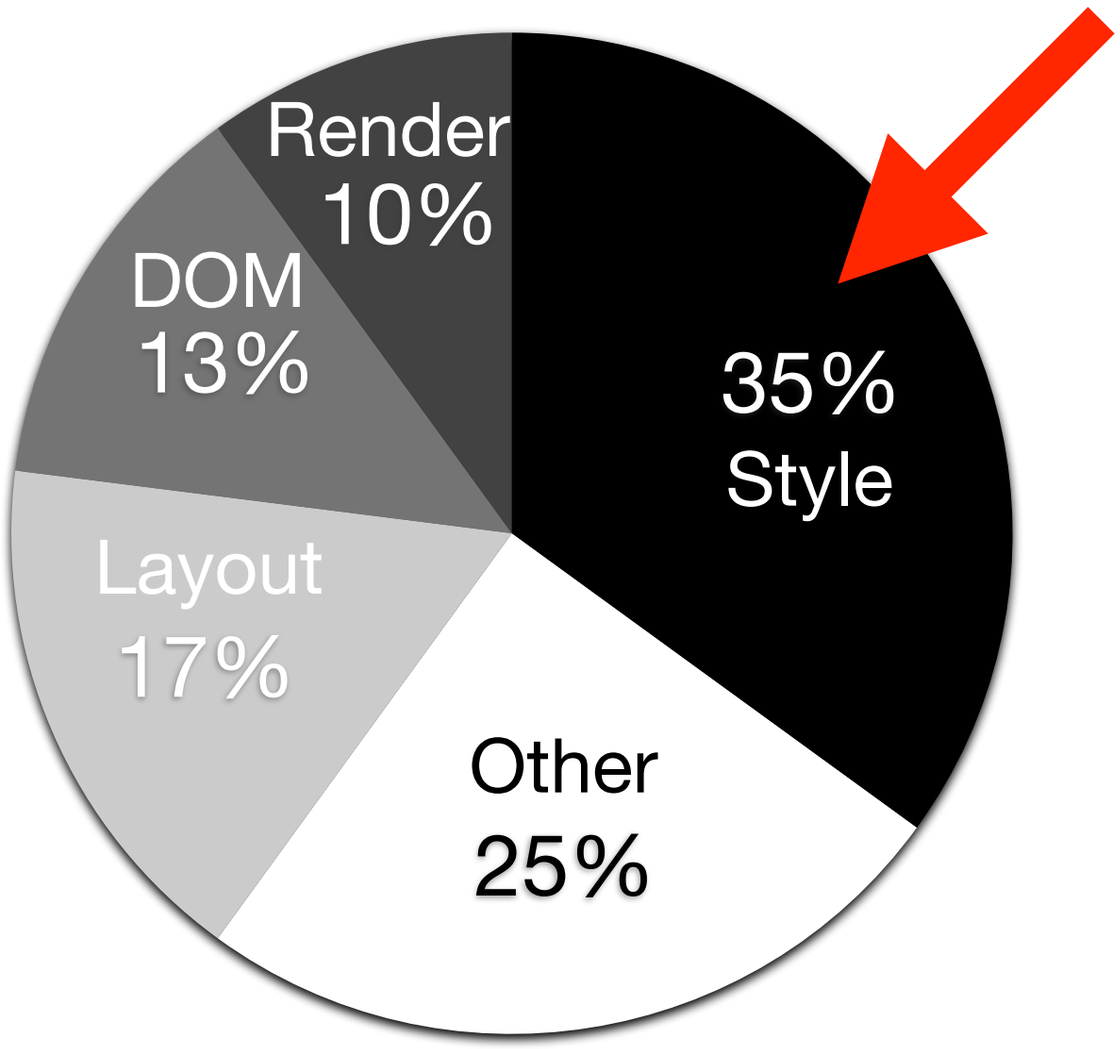


Execution time breakdown

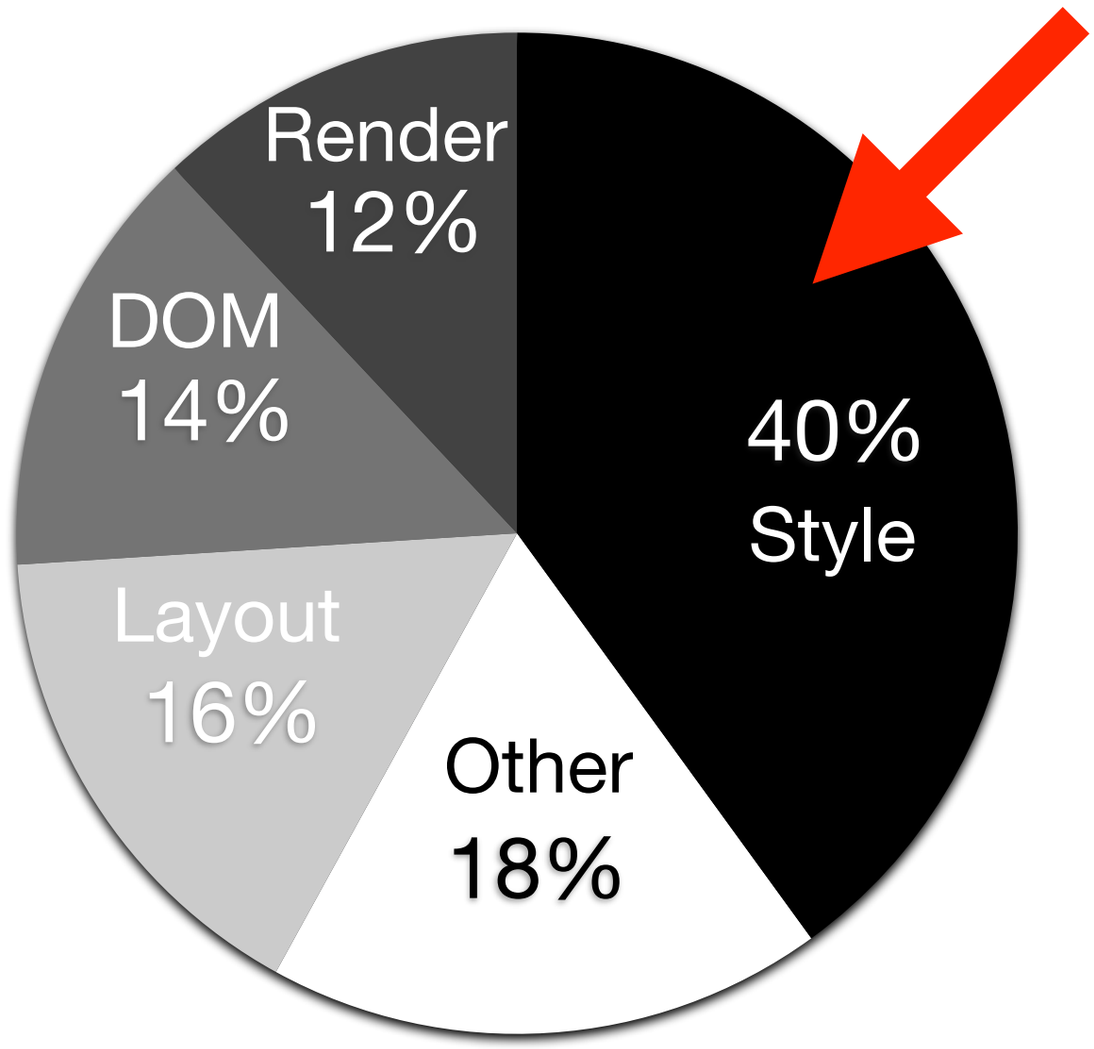


Energy breakdown

WebCore Specialization

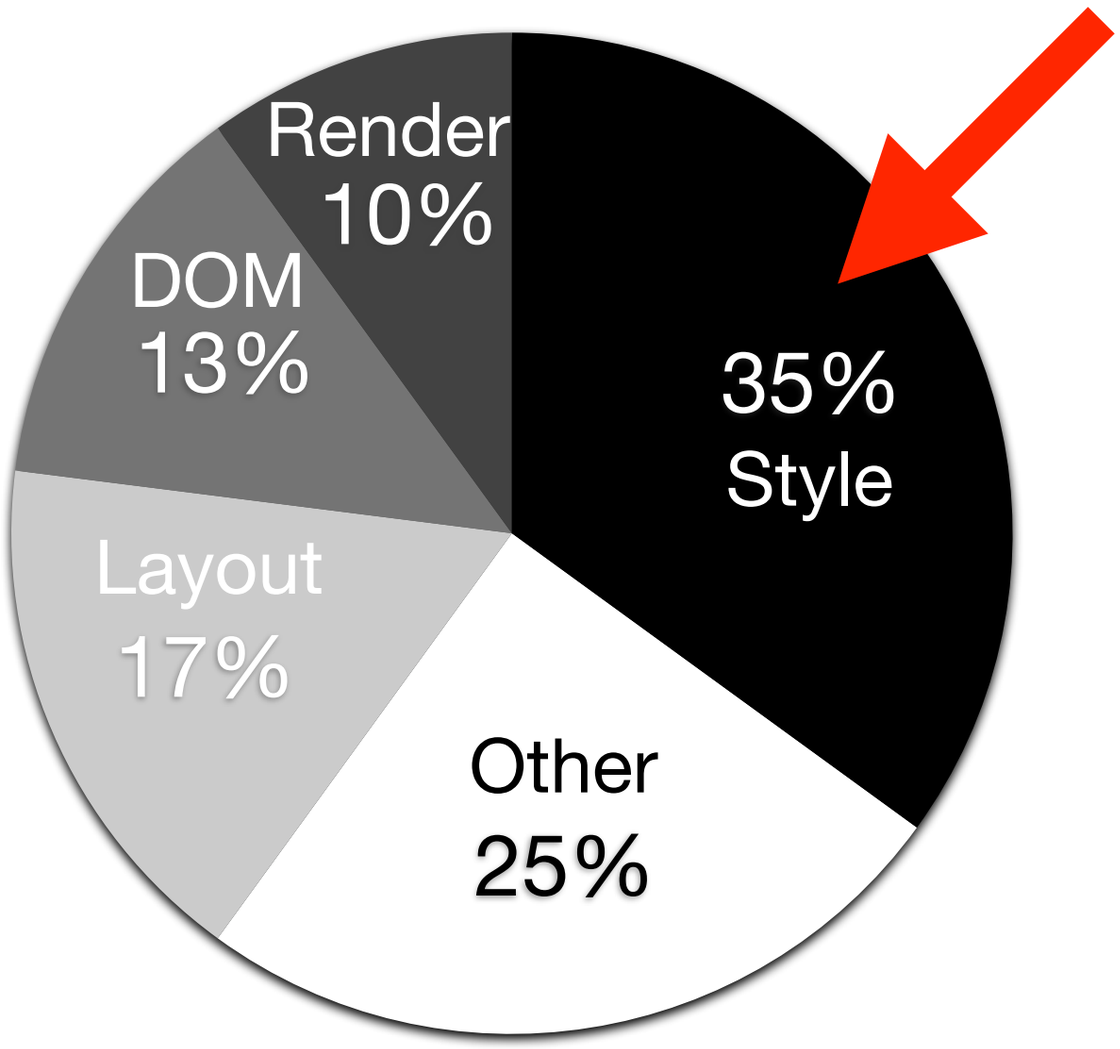


Execution time breakdown

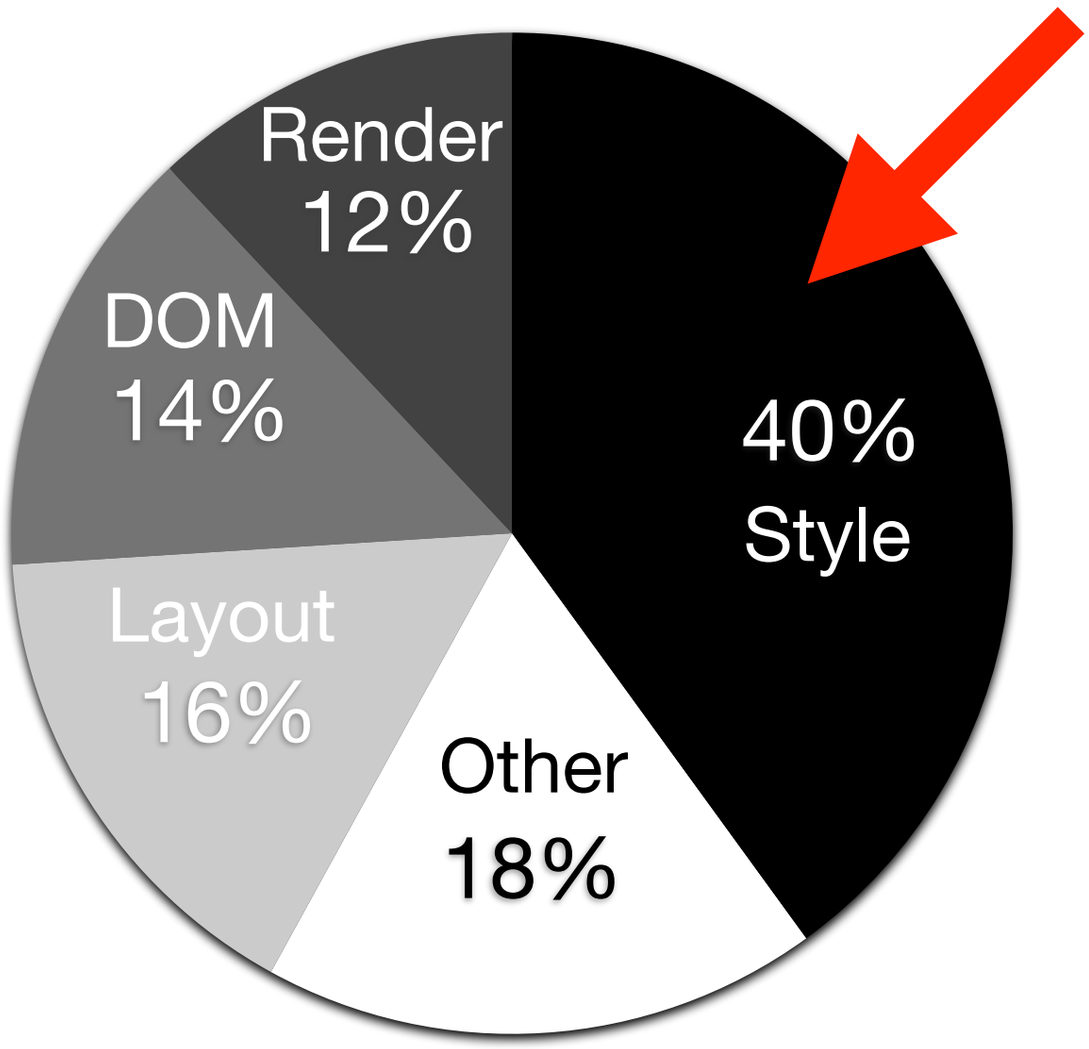


Energy breakdown

WebCore Specialization Style Resolution Kernel



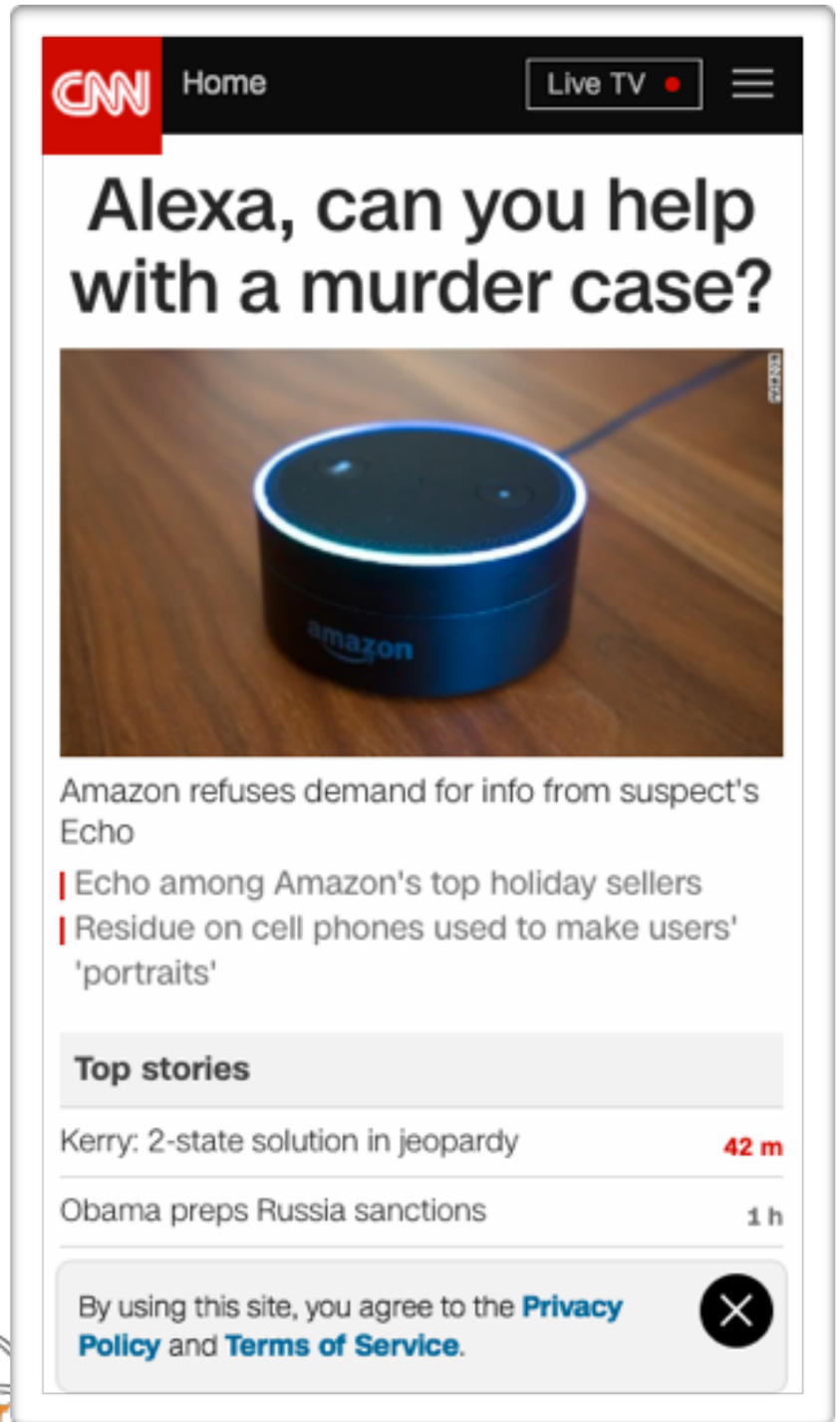
Execution time breakdown



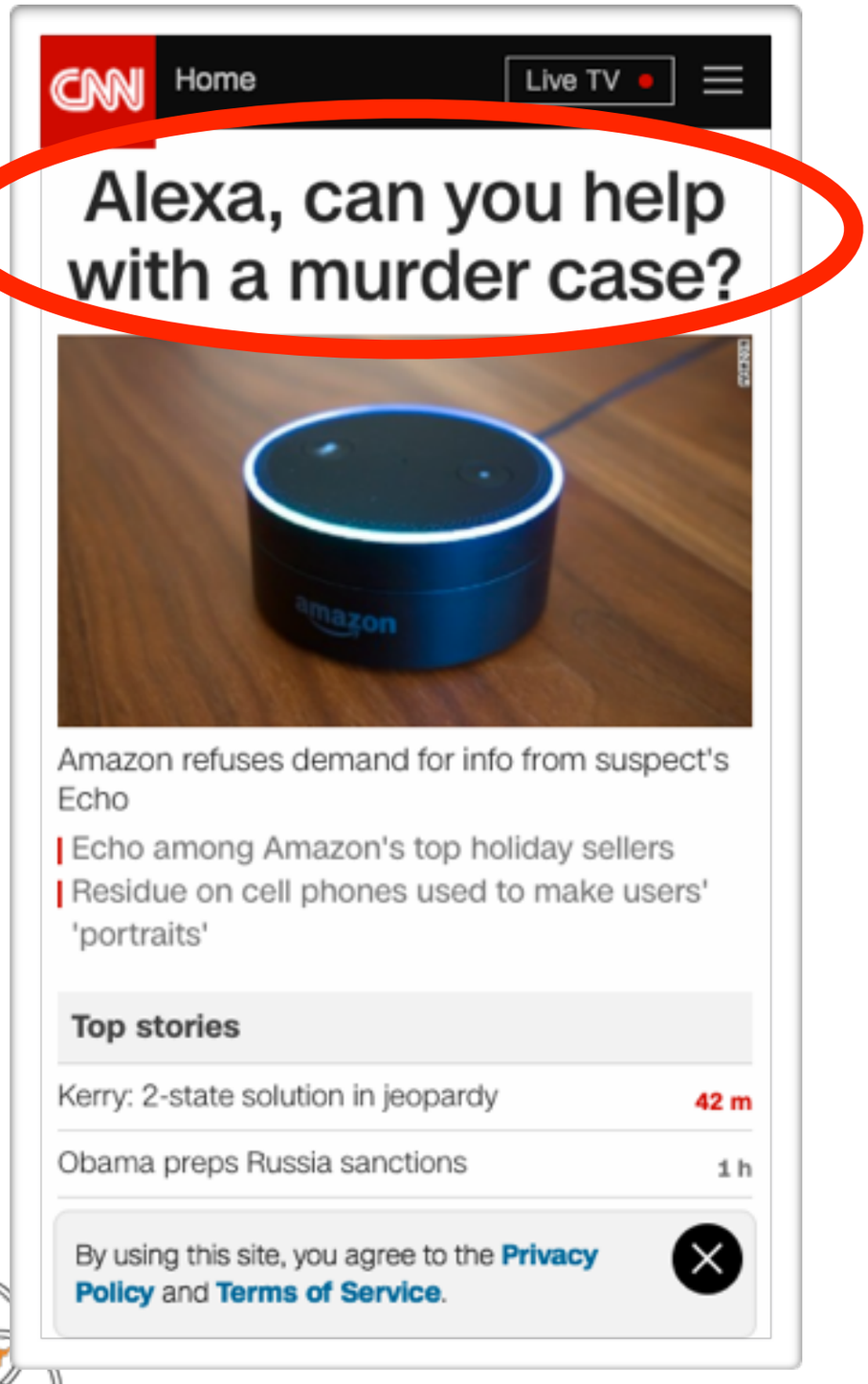
Energy breakdown

Style Resolution: A Running Example

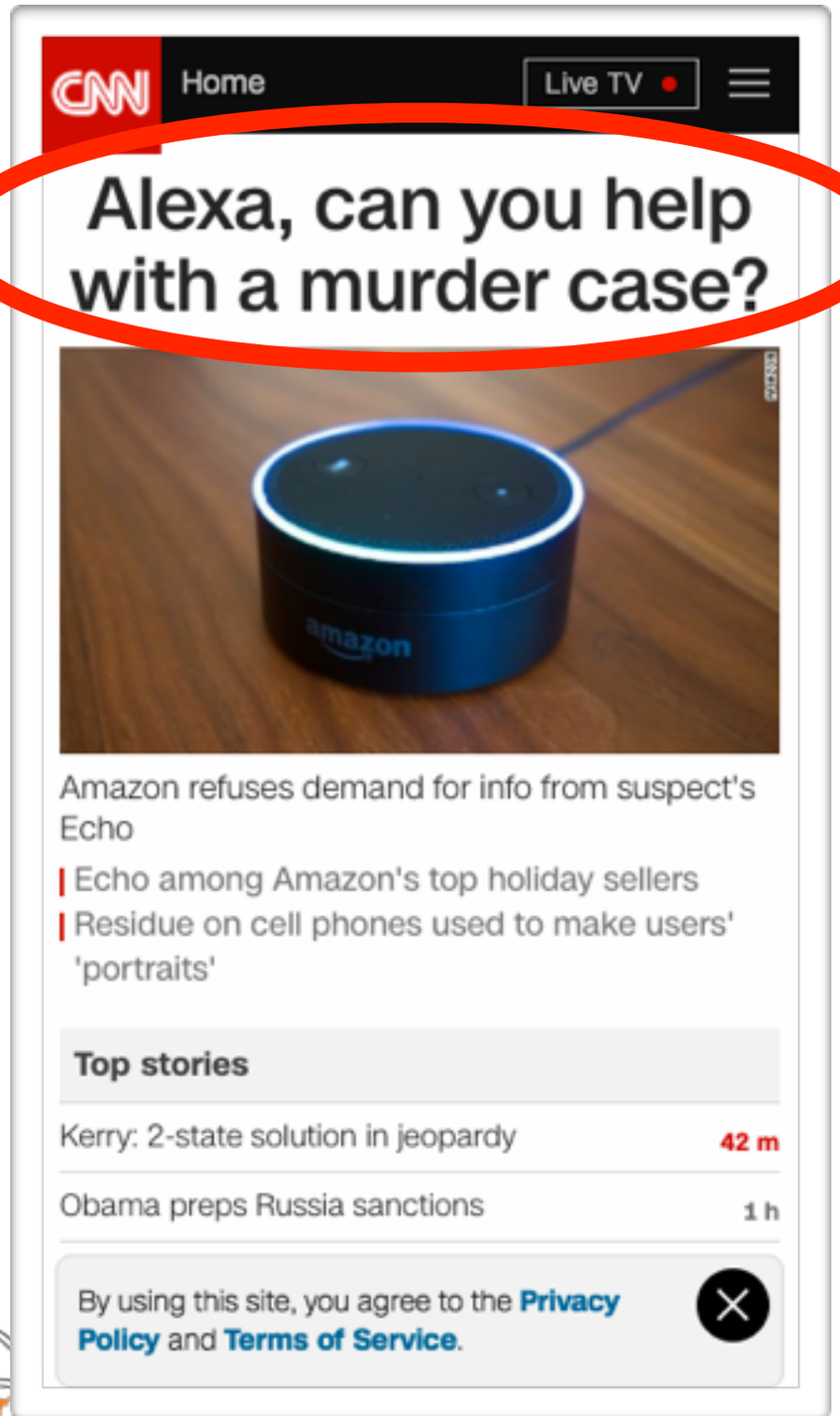
Style Resolution: A Running Example



Style Resolution: A Running Example



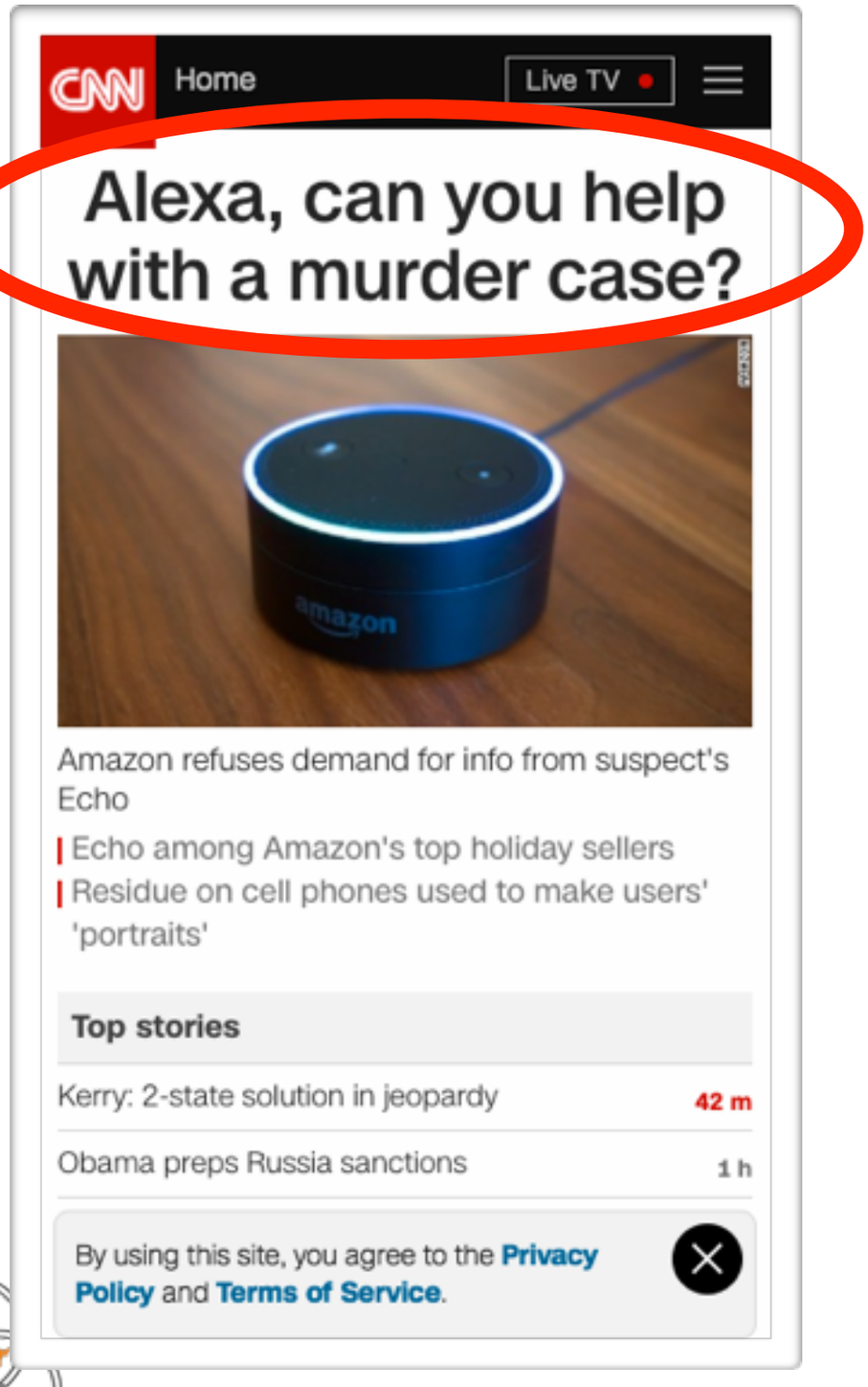
Style Resolution: A Running Example



Style Info

```
color: ???
width: ???
```

Style Resolution: A Running Example



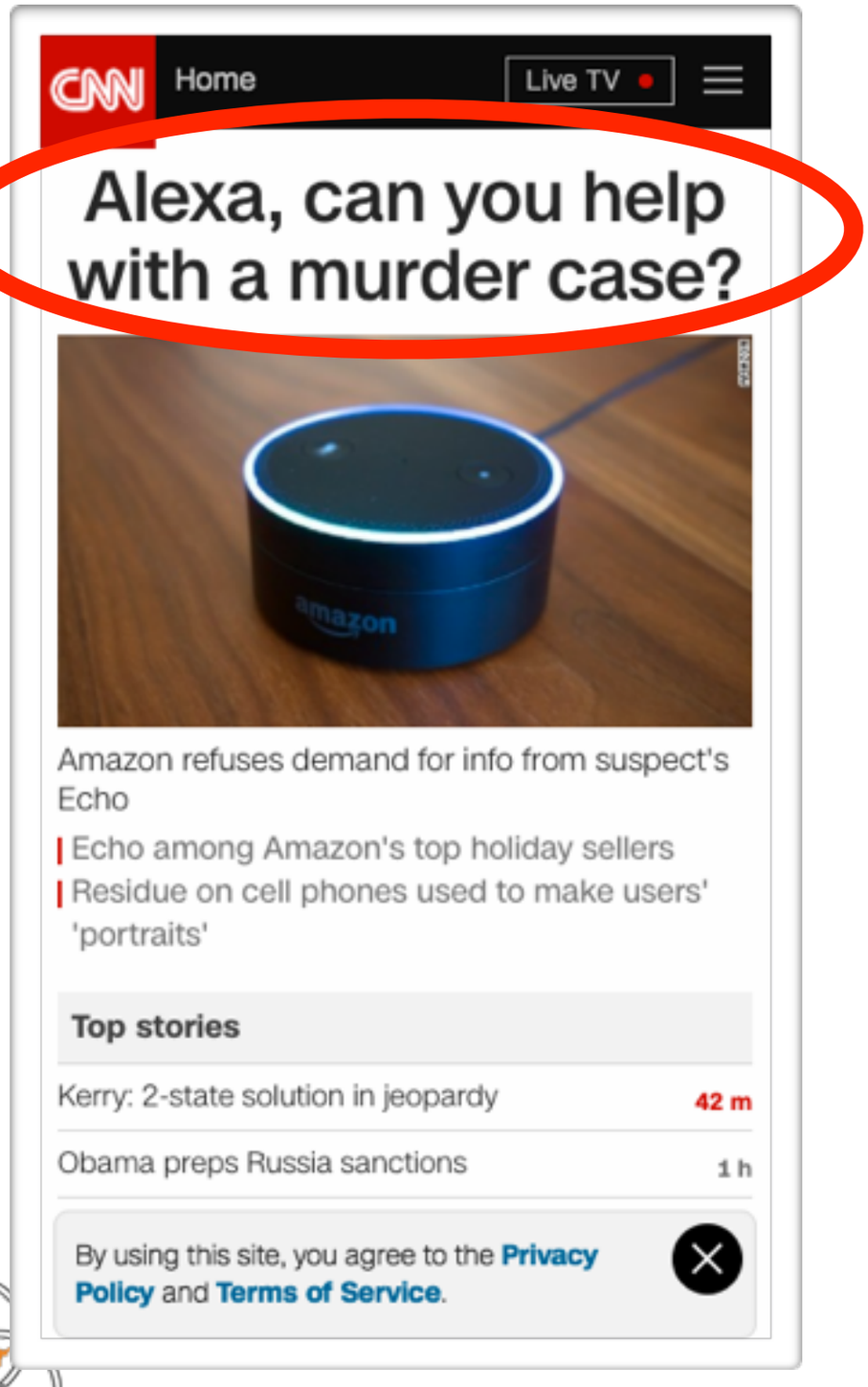
Style Info

```
color: ???
width: ???
```

Style Rules

```
a {
  color: blue;
  width: 100%;
}
```

Style Resolution: A Running Example



Style Info

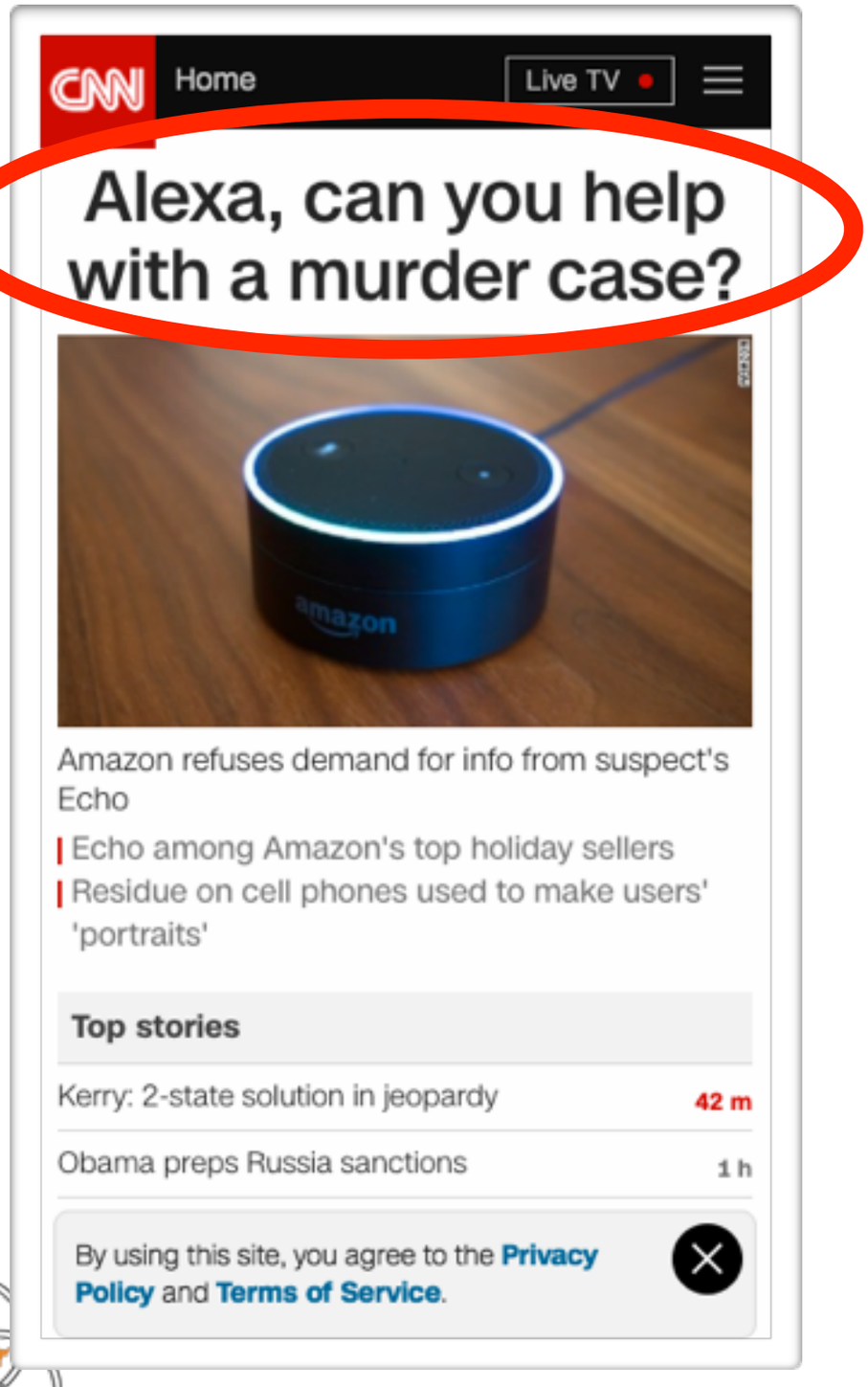
```
color: ???
width: ???
```

Style Rules

```
a {
  color: blue;
  width: 100%;
}
```

```
.header-text {
  color: black;
}
```

Style Resolution: A Running Example



Style Info

```
color: ???
width: ???
```

Style Rules

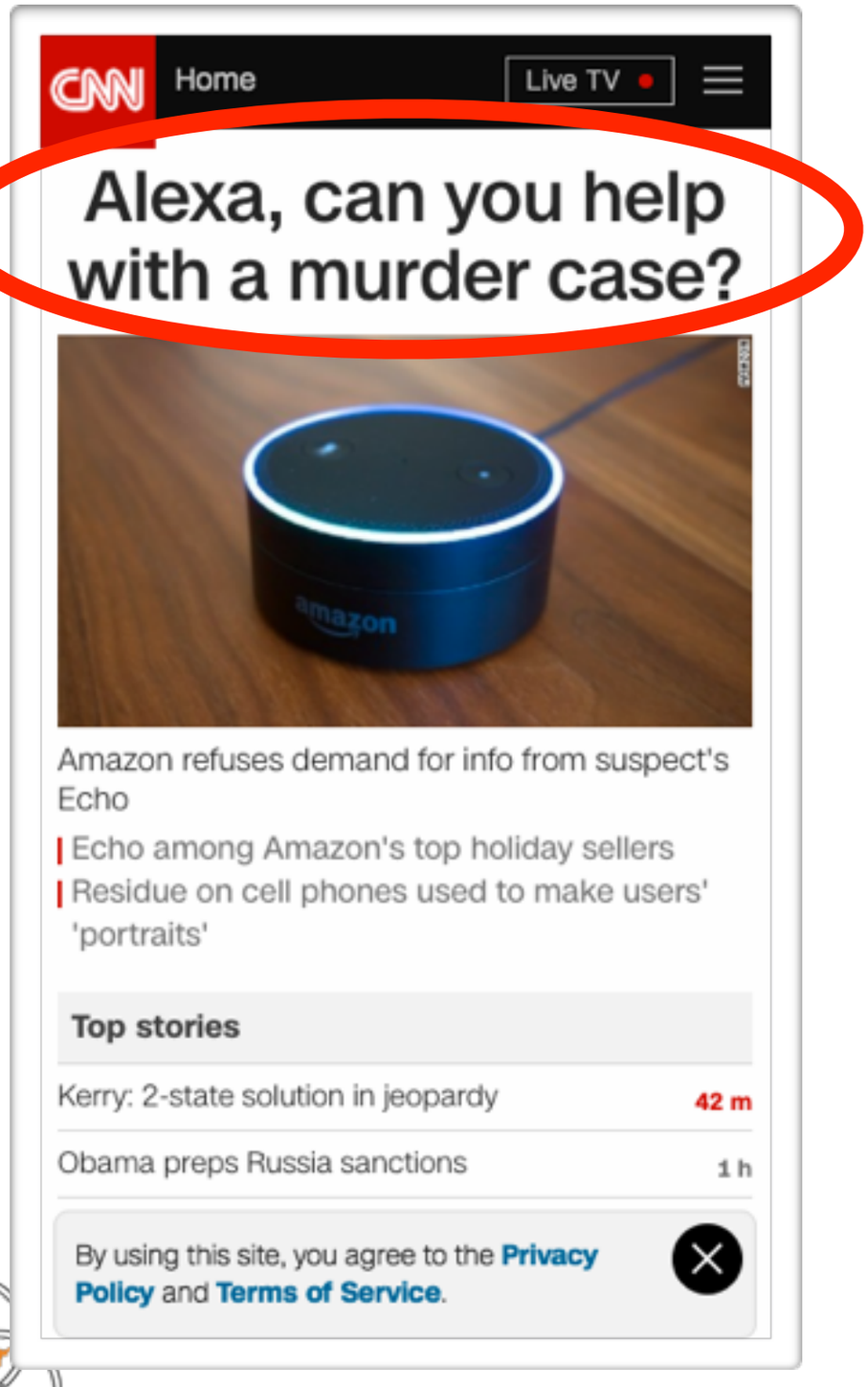
```
a {
  color: blue;
  width: 100%;
}
```

Low Priority

```
.header-text {
  color: black;
}
```

High Priority

Style Resolution: A Running Example



Style Info

```
color: black
width: 100%
```

Style Rules

```
a {
  color: blue;
  width: 100%;
}
```

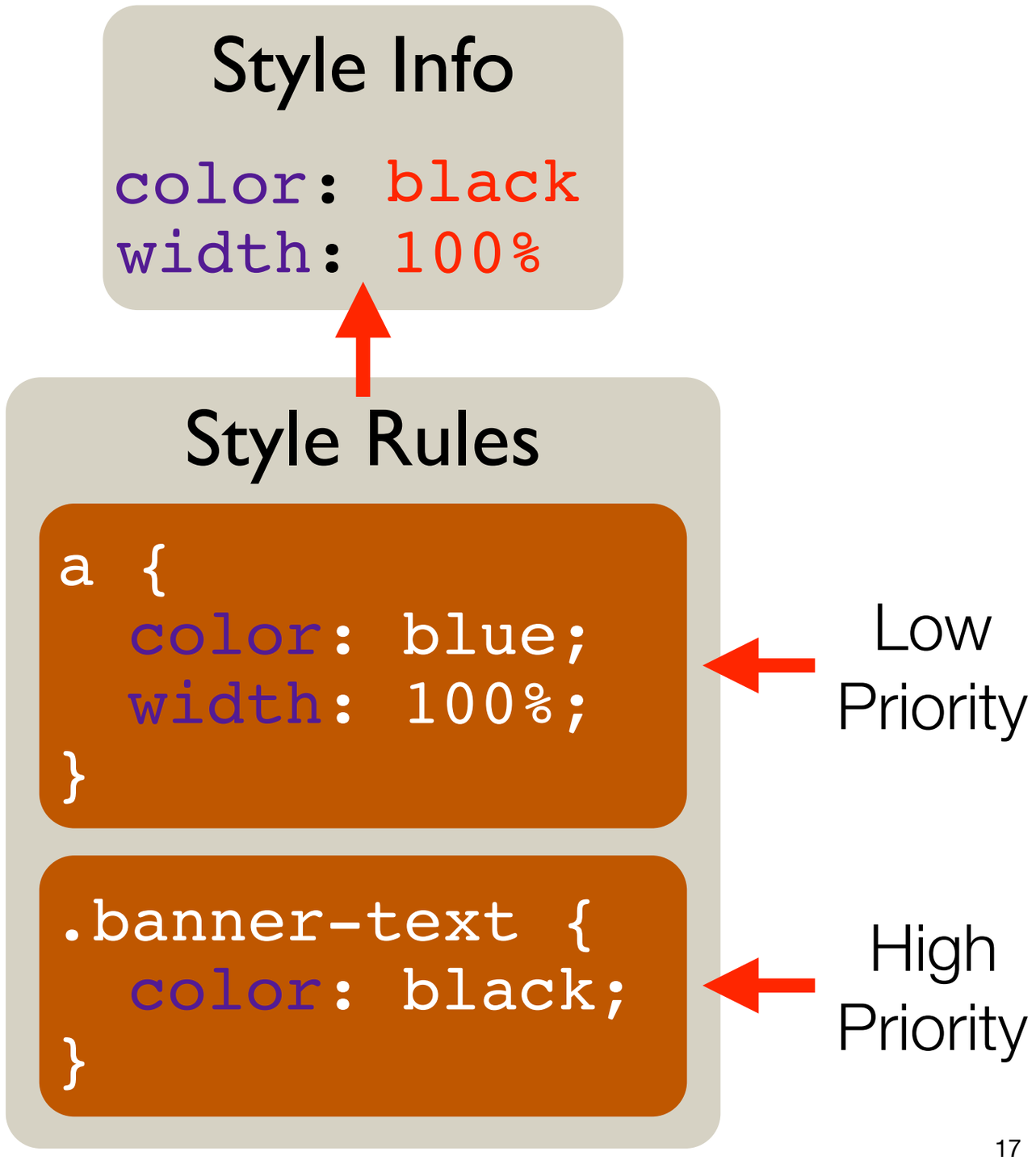
Low Priority

```
.header-text {
  color: black;
}
```

High Priority

Style Resolution: A Running Example

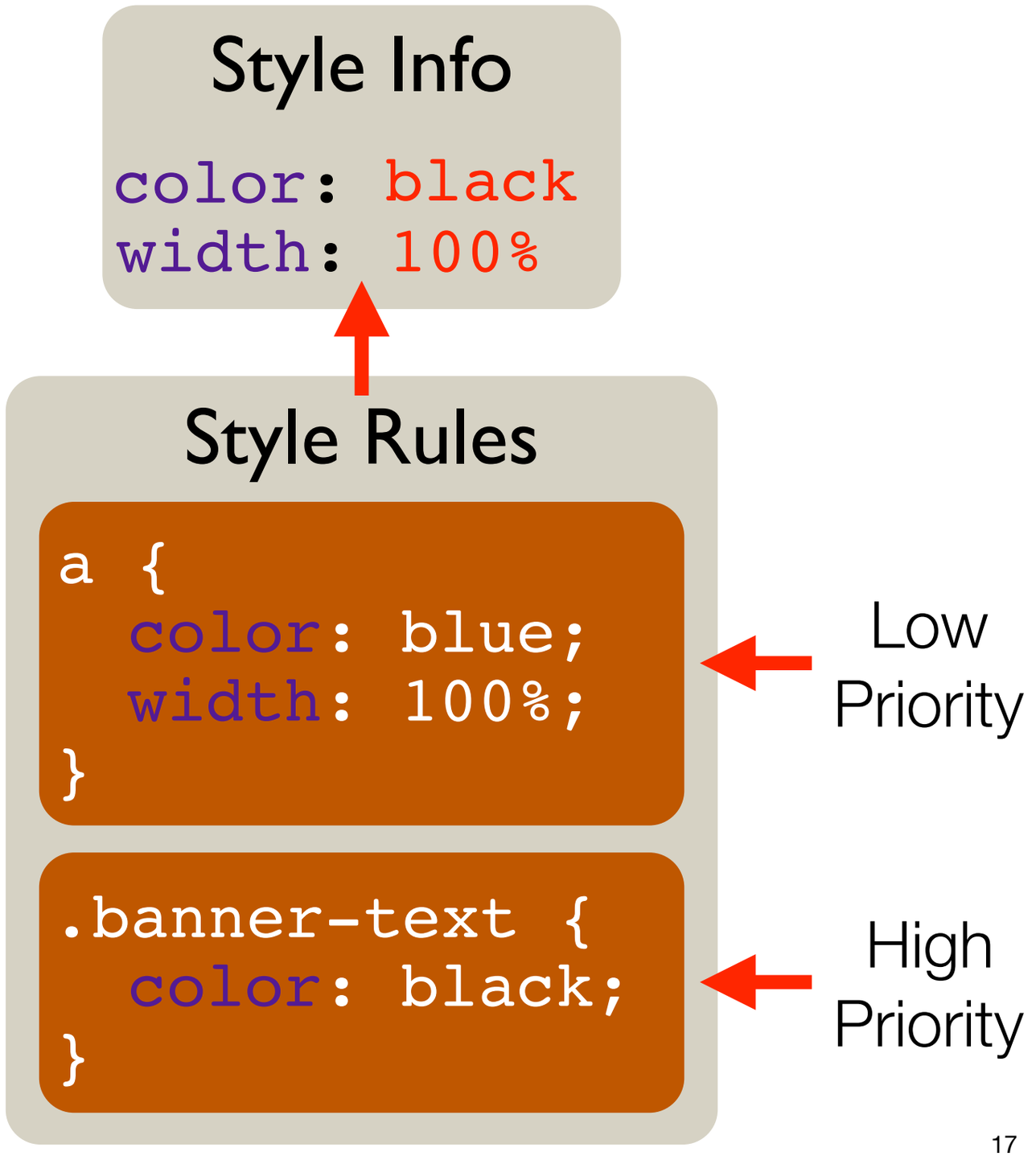
Two Observations



Style Resolution: A Running Example

Two Observations

Rules have "W*rite-after-Write*" dependencies.

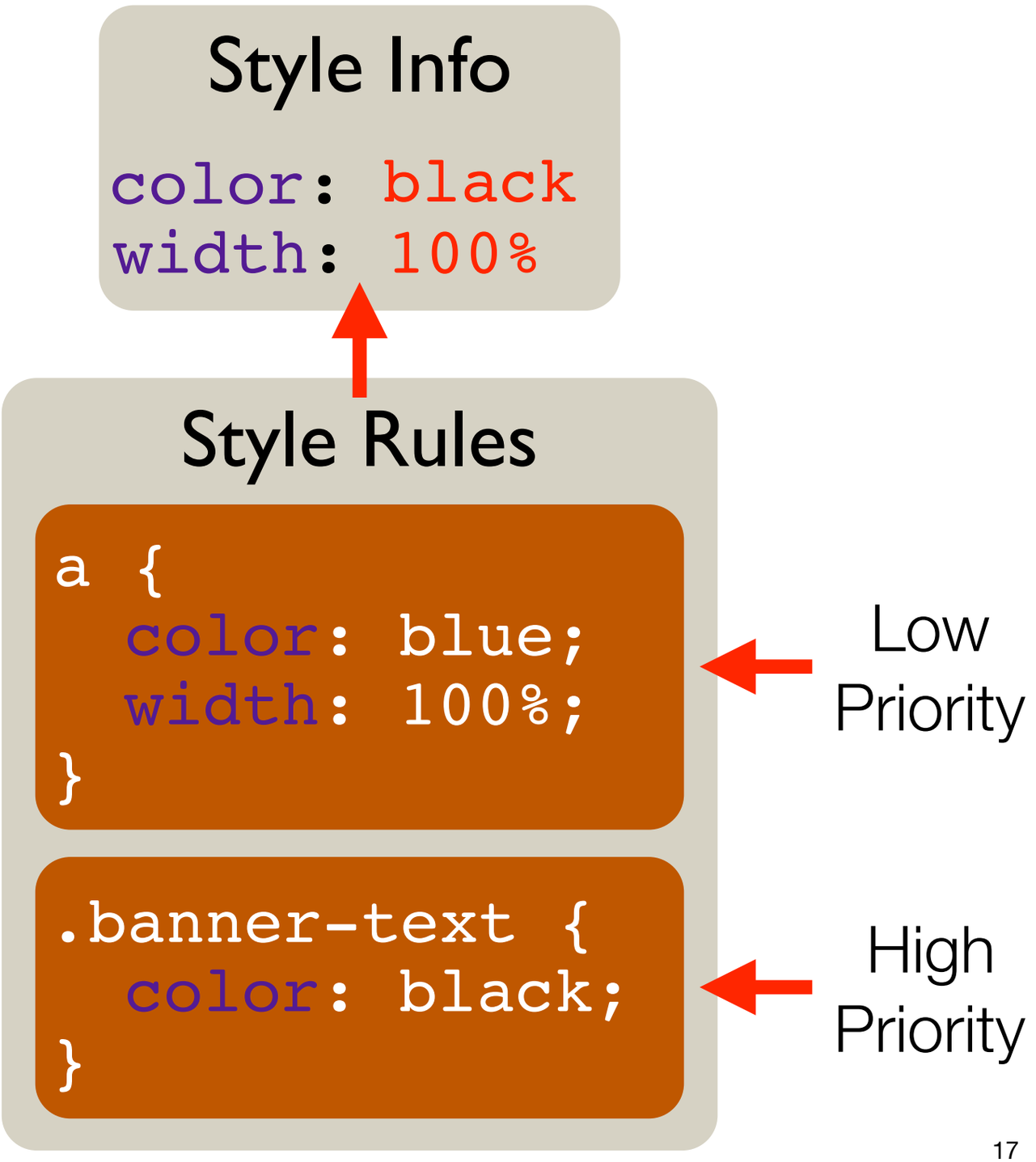


Style Resolution: A Running Example

Two Observations

Rules have "W*rite-after-Write*" dependencies.

Correct *execution order* specified by priorities.



Style Resolution: Software Implementation

```
matchedRules = MatchAndSortRules();

// Iterate from low to high priority
for (rule in matchedRules){ {

  for (property in rule) {

    Handler(property.value, DOMNode);

  }}
```

Style Resolution: Software Implementation

```
matchedRules = MatchAndSortRules();

// Iterate from low to high priority
for (rule in matchedRules){ {
   for (property in rule) {
     Handler(property.value, DOMNode);
    }}
```

Style Resolution: Software Implementation

```
matchedRules = MatchAndSortRules();

// Iterate from low to high priority
for (rule in matchedRules){ {
  for (property in rule) {
    Handler(property.value, DOMNode);
  }}
```

Style Resolution: Software Implementation

```
matchedRules = MatchAndSortRules();

// Iterate from low to high priority
for (rule in matchedRules){ {
  for (property in rule) {
    Handler(property.value, DOMNode);
  }}
```

(**Ordered**) Rule-level Parallelism

Style Resolution: Software Implementation

```
matchedRules = MatchAndSortRules();

// Iterate from low to high priority
for (rule in matchedRules){ {
   for (property in rule) {
     Handler(property.value, DOMNode);
   }}
```

(**Ordered**) Rule-level Parallelism

Style Resolution: Software Implementation

```
matchedRules = MatchAndSortRules();

// Iterate from low to high priority
for (rule in matchedRules){ {

   for (property in rule) {

     Handler(property.value, DOMNode)

   }}
```

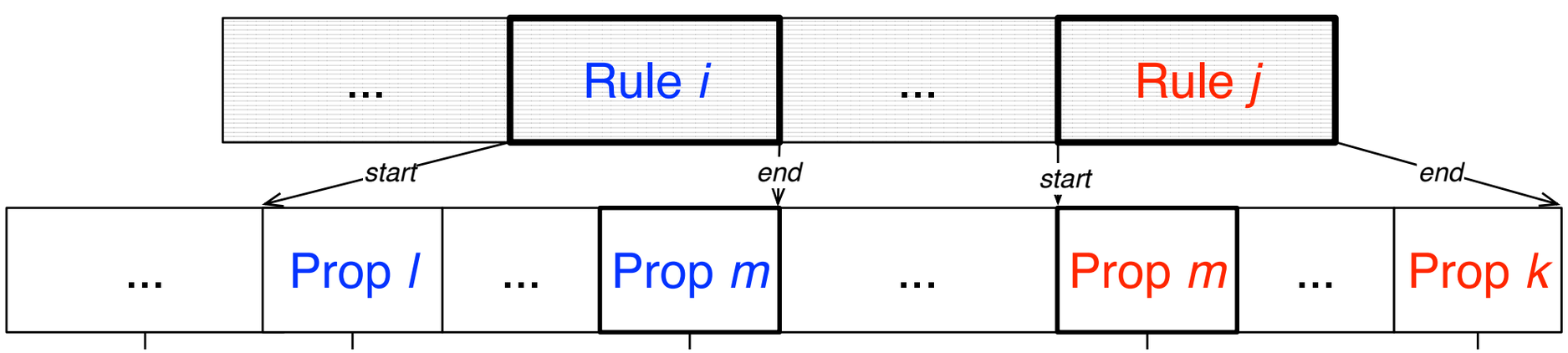
(**Ordered**) Rule-level Parallelism

(**Unordered**) Property-level Parallelism

Style Resolution Unit

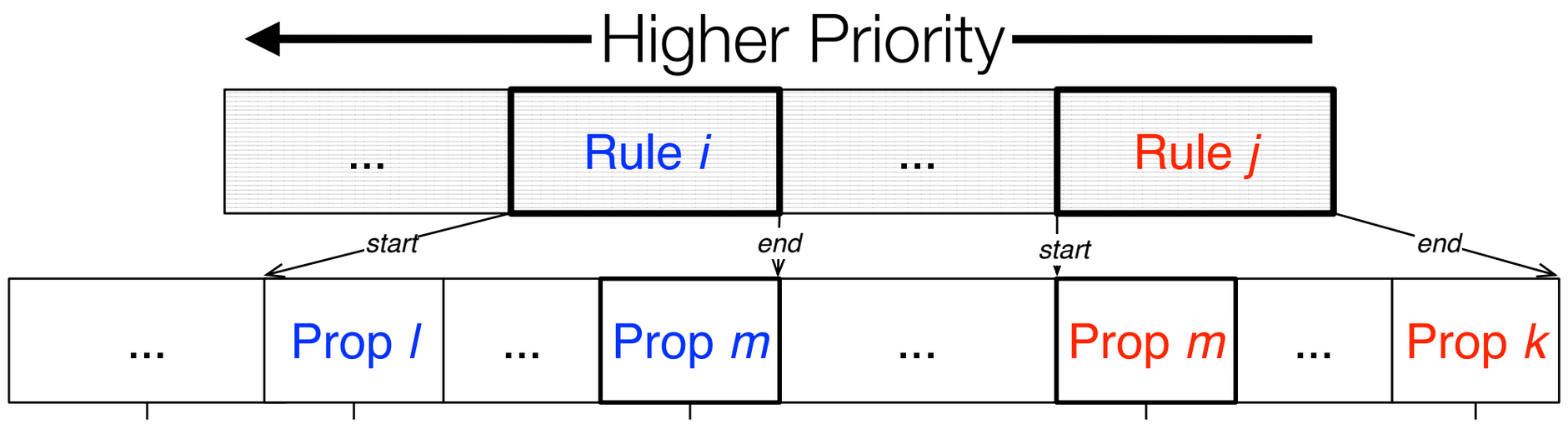
Style Resolution Unit

Input Scratchpad



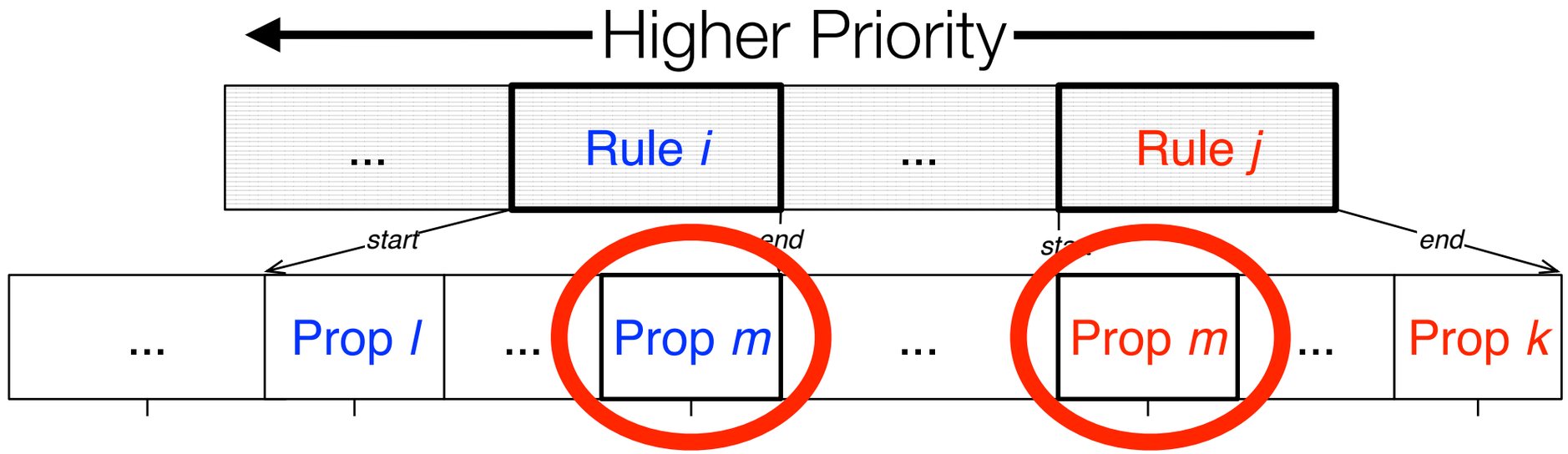
Style Resolution Unit

Input Scratchpad



Style Resolution Unit

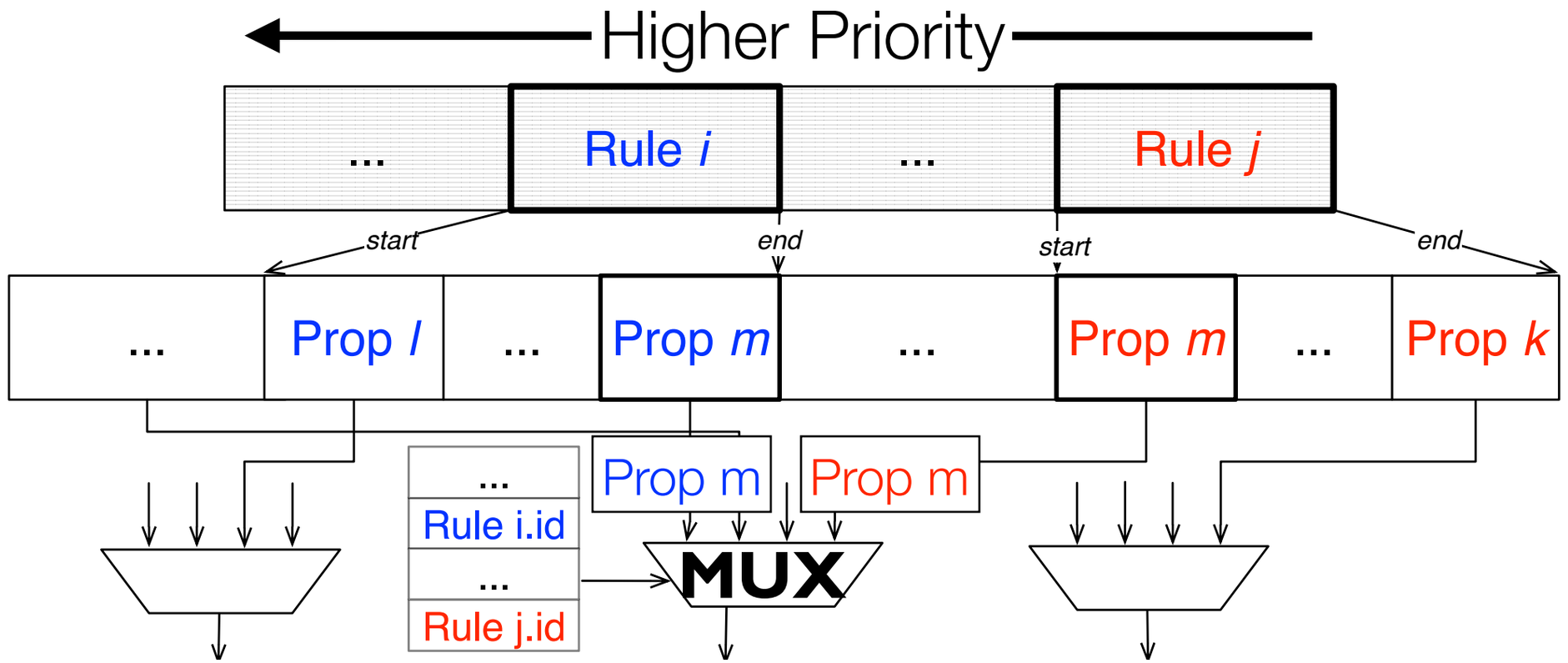
Input Scratchpad



Style Resolution Unit

Input Scratchpad

Conflict Resolution

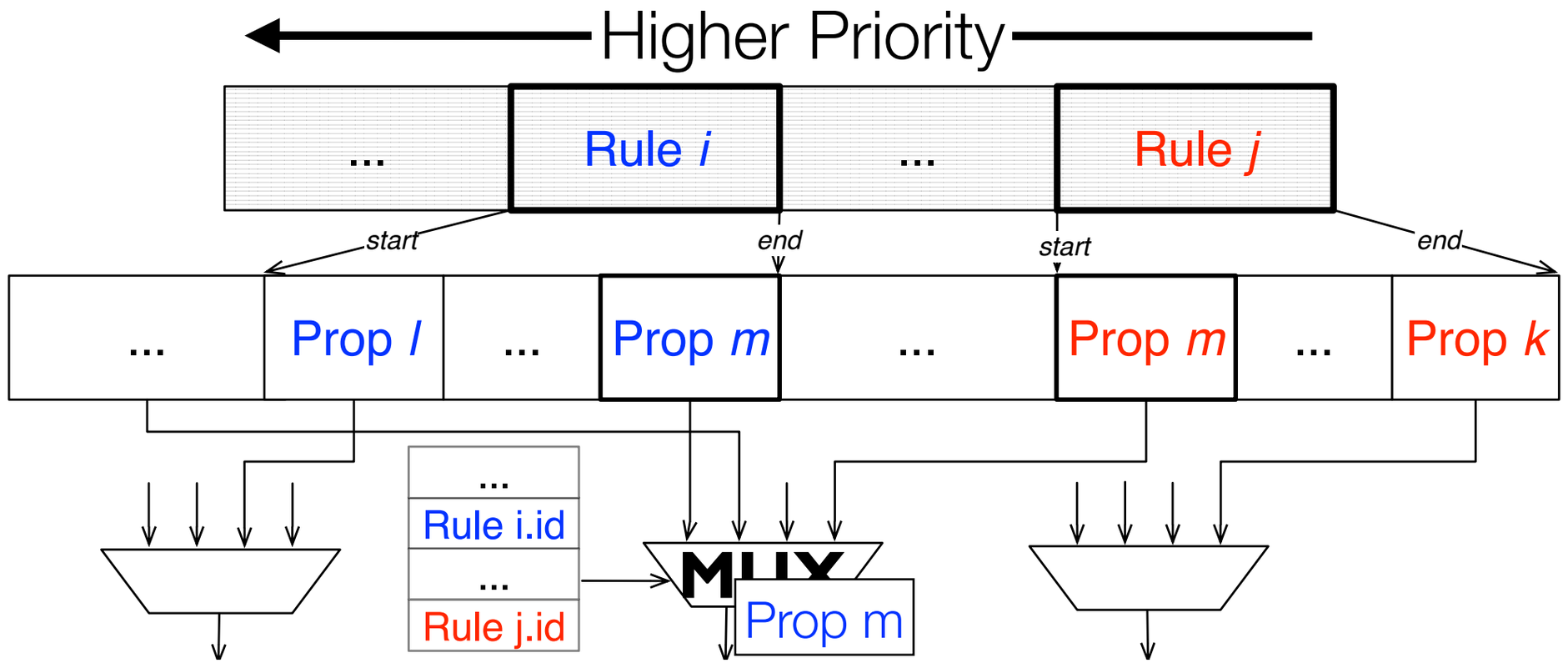


Style Resolution Unit

Ordered Rule-level Parallelism

Input Scratchpad

Conflict Resolution



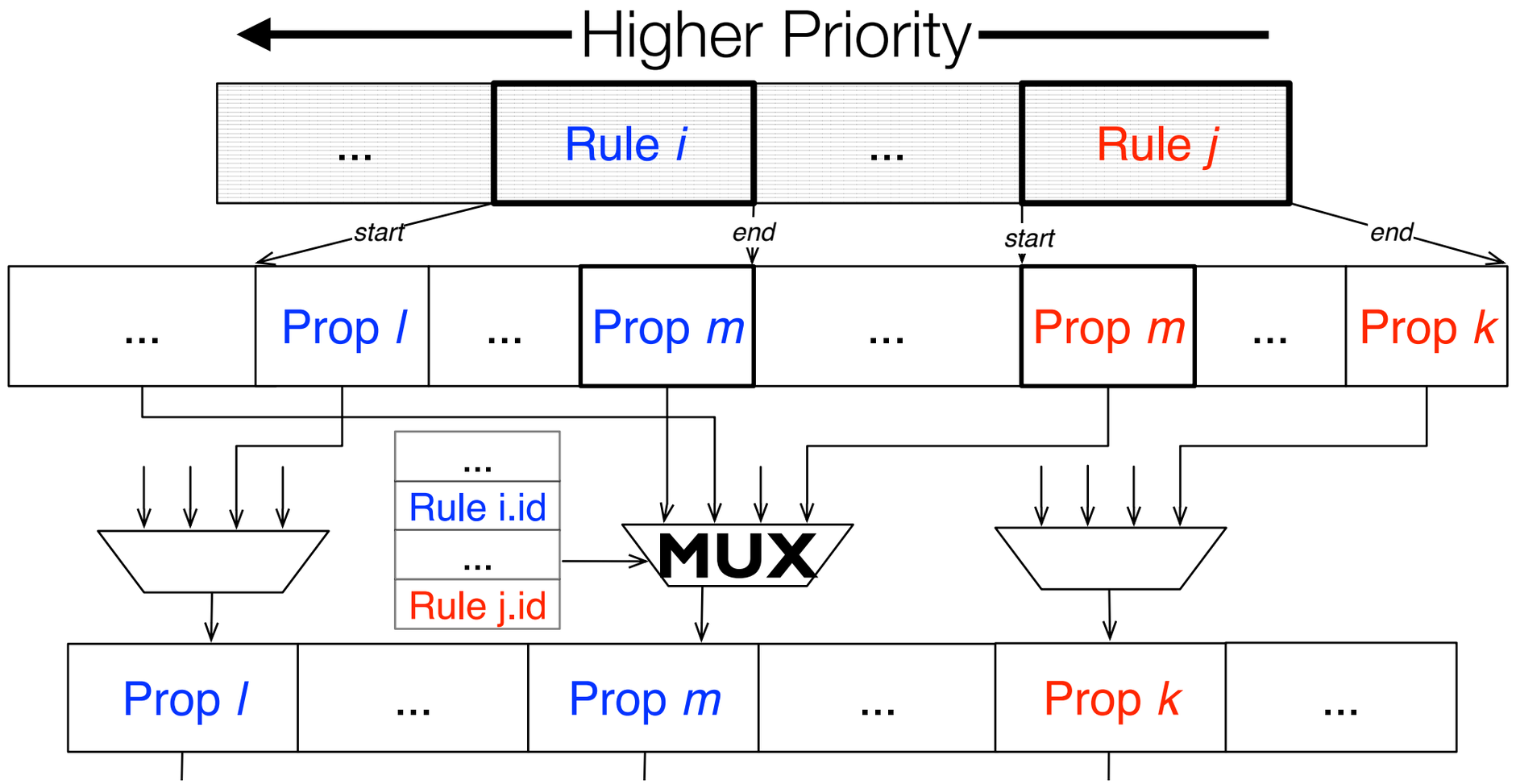
Style Resolution Unit

Ordered Rule-level Parallelism

Input Scratchpad

Conflict Resolution

Compute Lanes



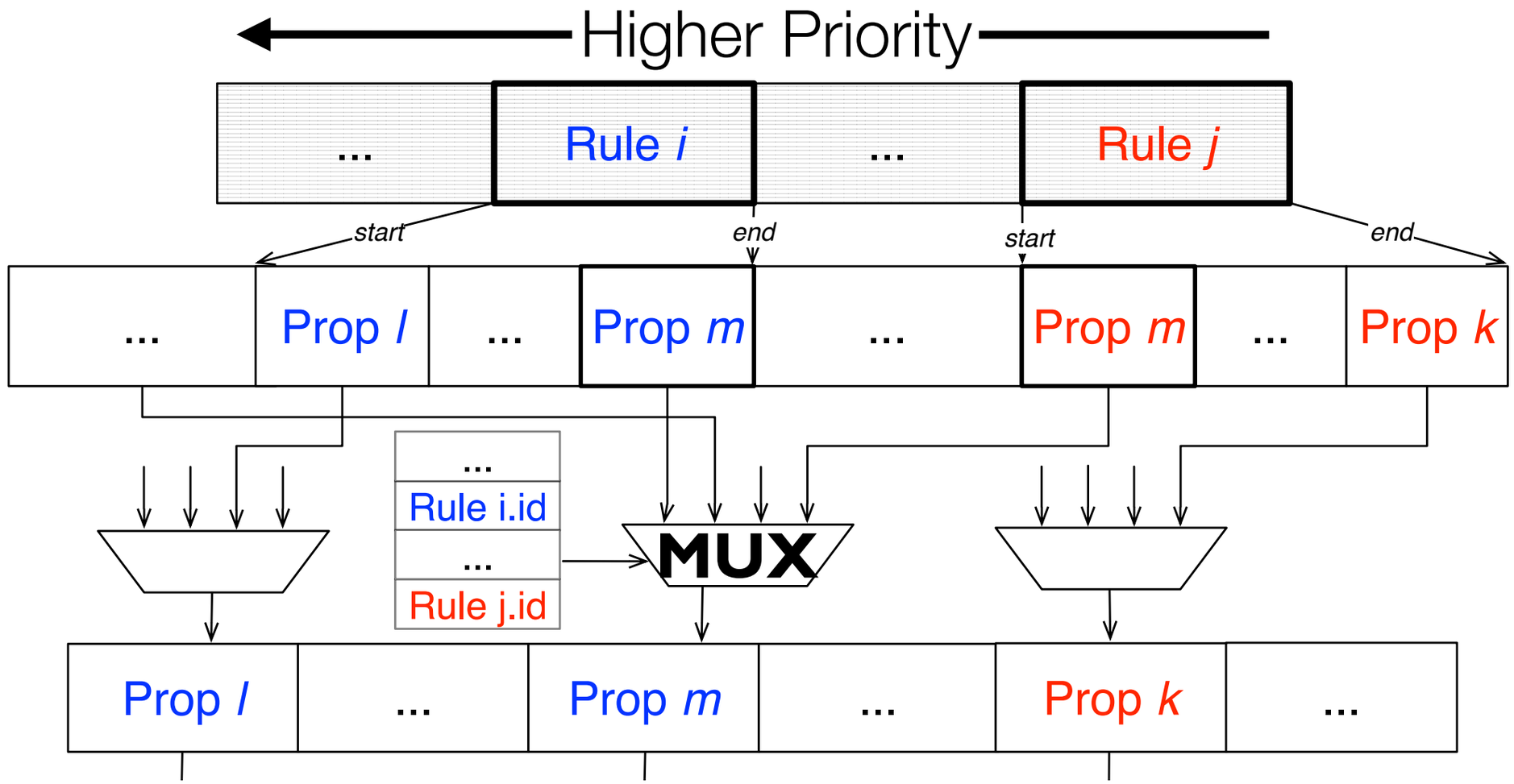
Style Resolution Unit

- ✓ Ordered Rule-level Parallelism
- ✓ Unordered Prop-level Parallelism

Input Scratchpad

Conflict Resolution

Compute Lanes



Style Resolution Unit

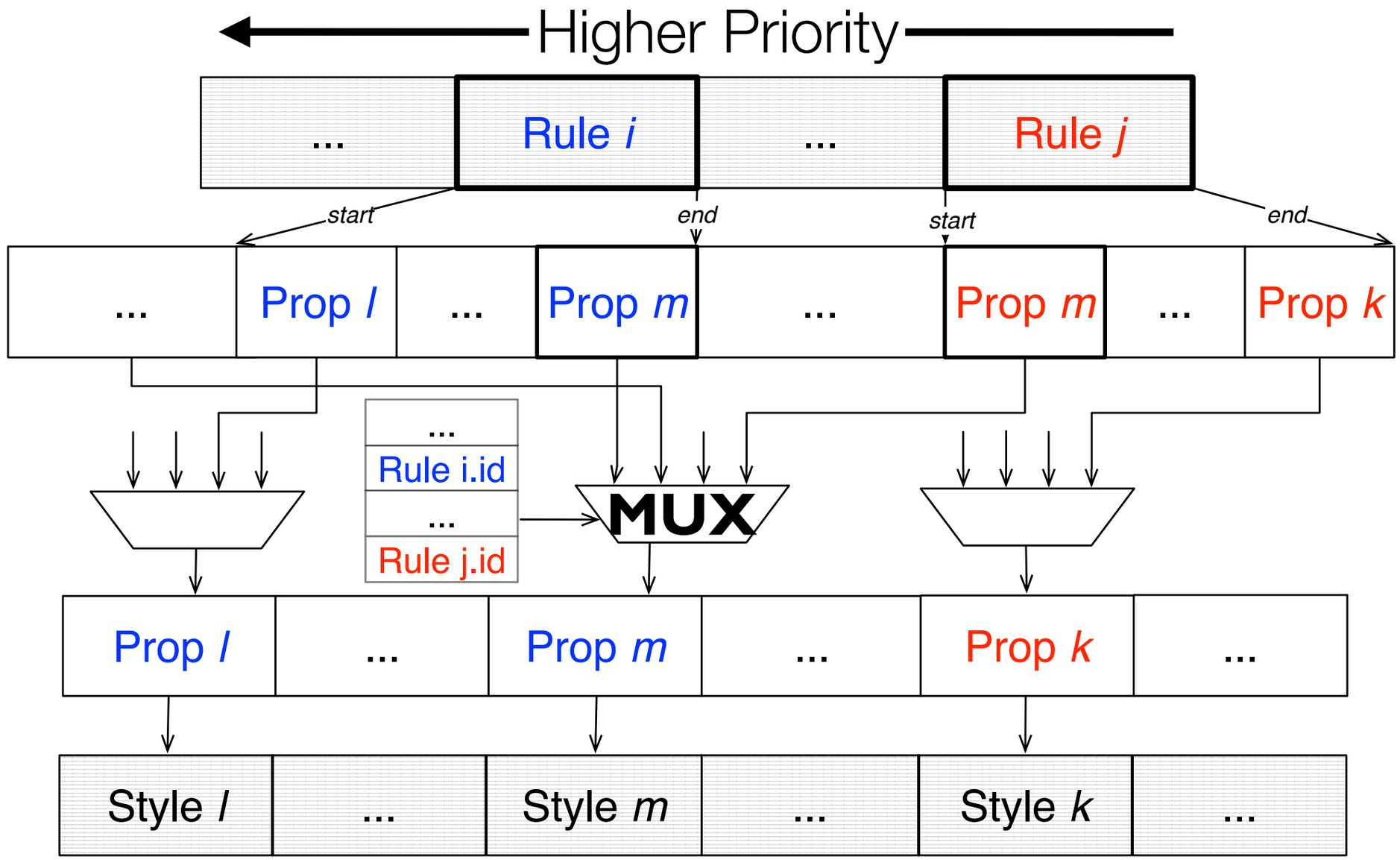
- ✓ Ordered Rule-level Parallelism
- ✓ Unordered Prop-level Parallelism

Input Scratchpad

Conflict Resolution

Compute Lanes

Output Scratchpad



Evaluation Results

Fully synthesized using Synopsys 28 nm toolchain.

Evaluation Results

Fully synthesized using Synopsys 28 nm toolchain.

Compare against ARM A15, a typical mobile CPU.

Evaluation Results

Fully synthesized using Synopsys 28 nm toolchain.

Compare against ARM A15, a typical mobile CPU.

Cost of specialization: **0.59 mm^2** area overhead

- A15's area: **19 mm^2**
- SoC die area is **122 mm^2** in Samsung Galaxy S4

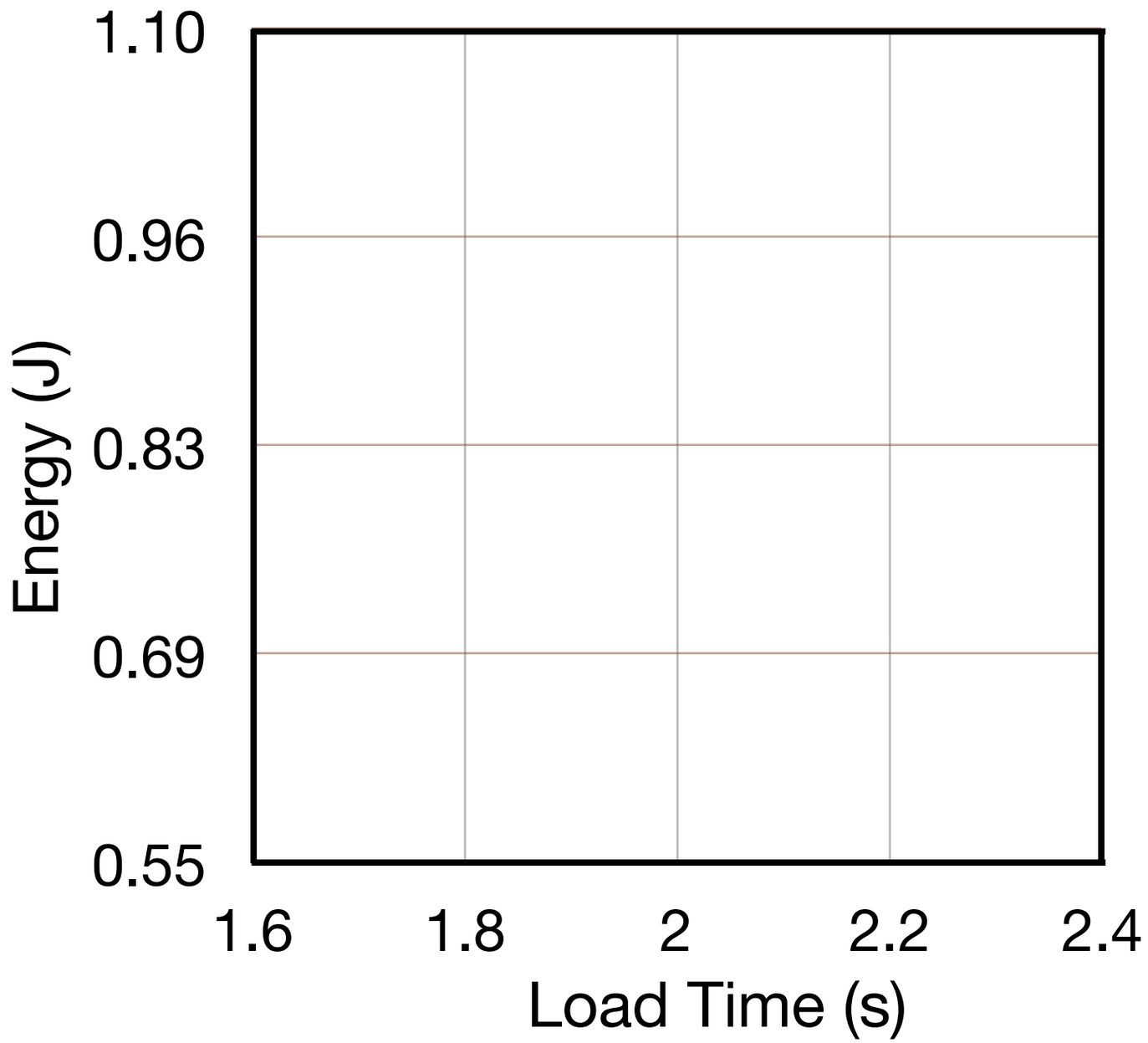
Evaluation Results

Fully synthesized using Synopsys 28 nm toolchain.

Compare against ARM A15, a typical mobile CPU.

Cost of specialization: **0.59 mm²** area overhead

- A15's area: **19 mm²**
- SoC die area is **122 mm²** in Samsung Galaxy S4



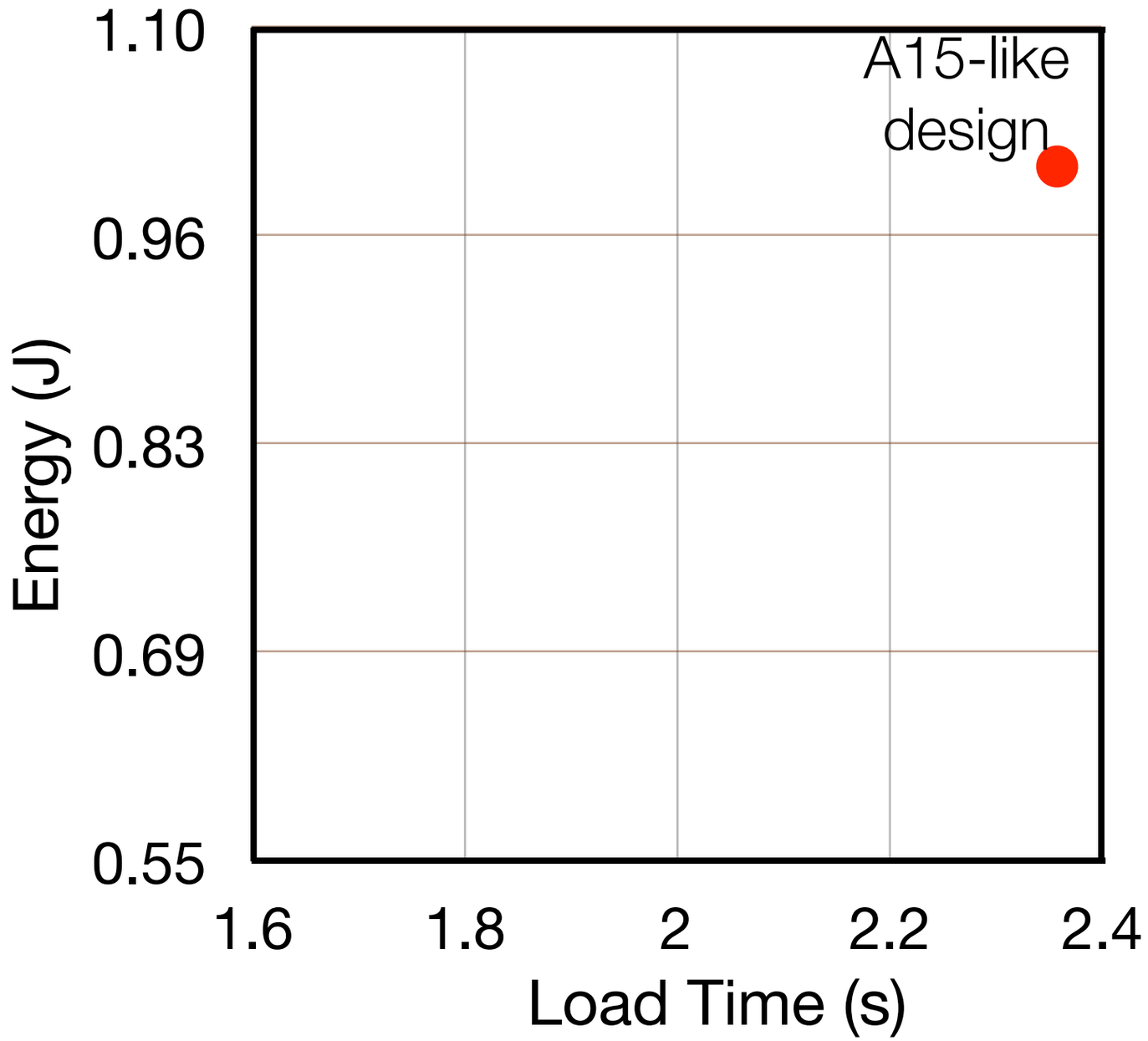
Evaluation Results

Fully synthesized using Synopsys 28 nm toolchain.

Compare against ARM A15, a typical mobile CPU.

Cost of specialization: **0.59 mm²** area overhead

- A15's area: **19 mm²**
- SoC die area is **122 mm²** in Samsung Galaxy S4



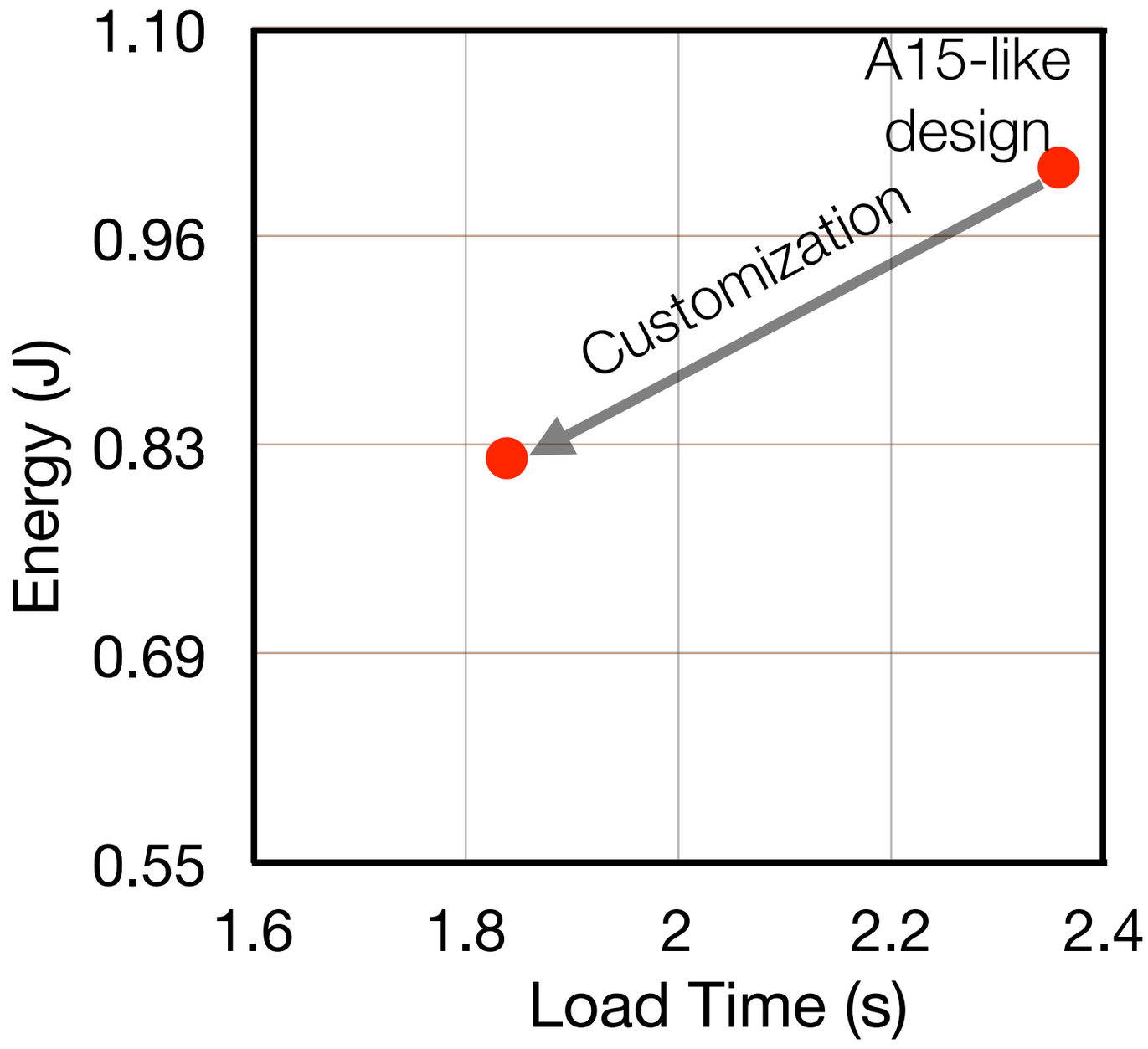
Evaluation Results

Fully synthesized using Synopsys 28 nm toolchain.

Compare against ARM A15, a typical mobile CPU.

Cost of specialization: **0.59 mm²** area overhead

- A15's area: **19 mm²**
- SoC die area is **122 mm²** in Samsung Galaxy S4



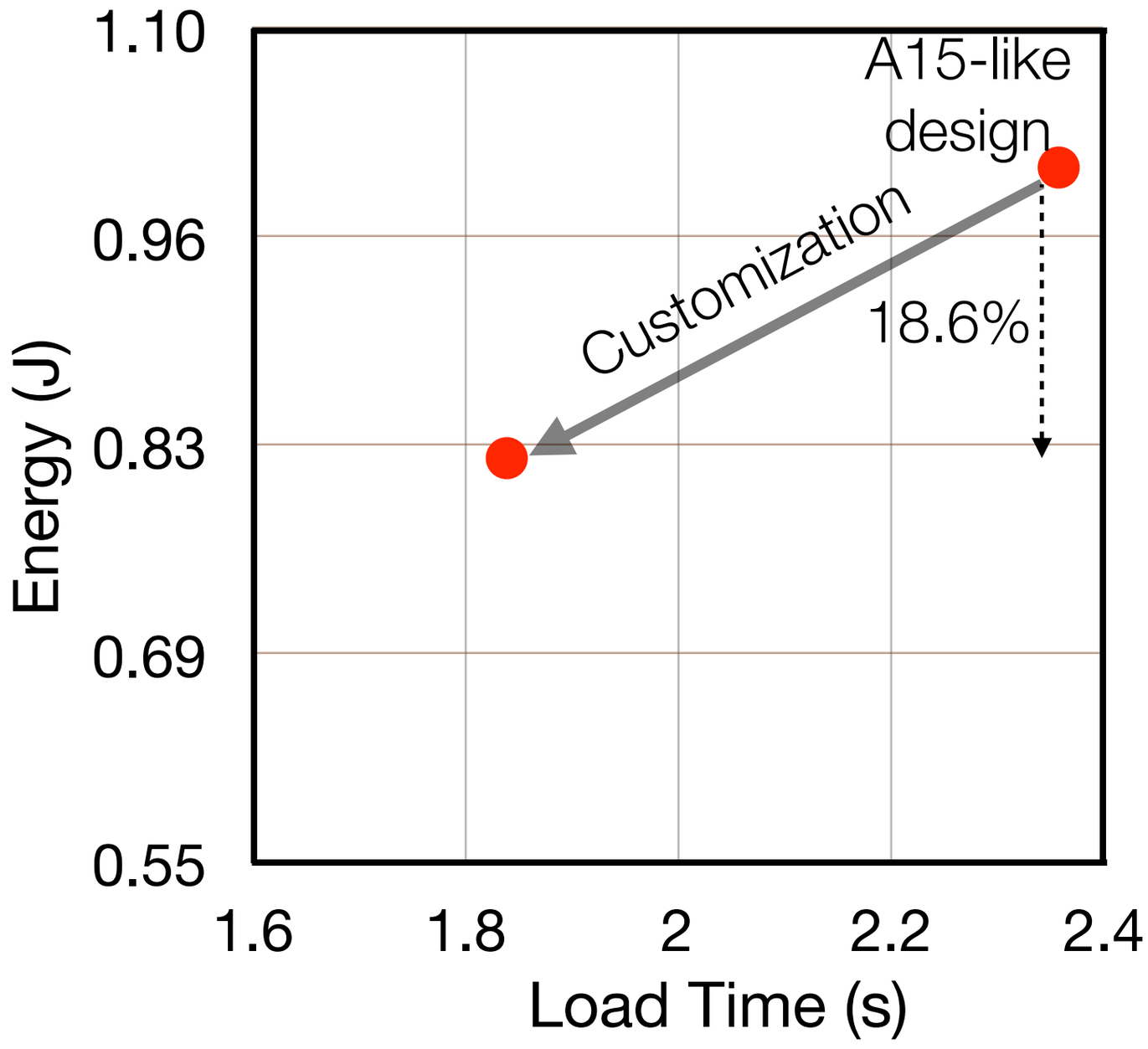
Evaluation Results

Fully synthesized using Synopsys 28 nm toolchain.

Compare against ARM A15, a typical mobile CPU.

Cost of specialization: **0.59 mm²** area overhead

- A15's area: **19 mm²**
- SoC die area is **122 mm²** in Samsung Galaxy S4



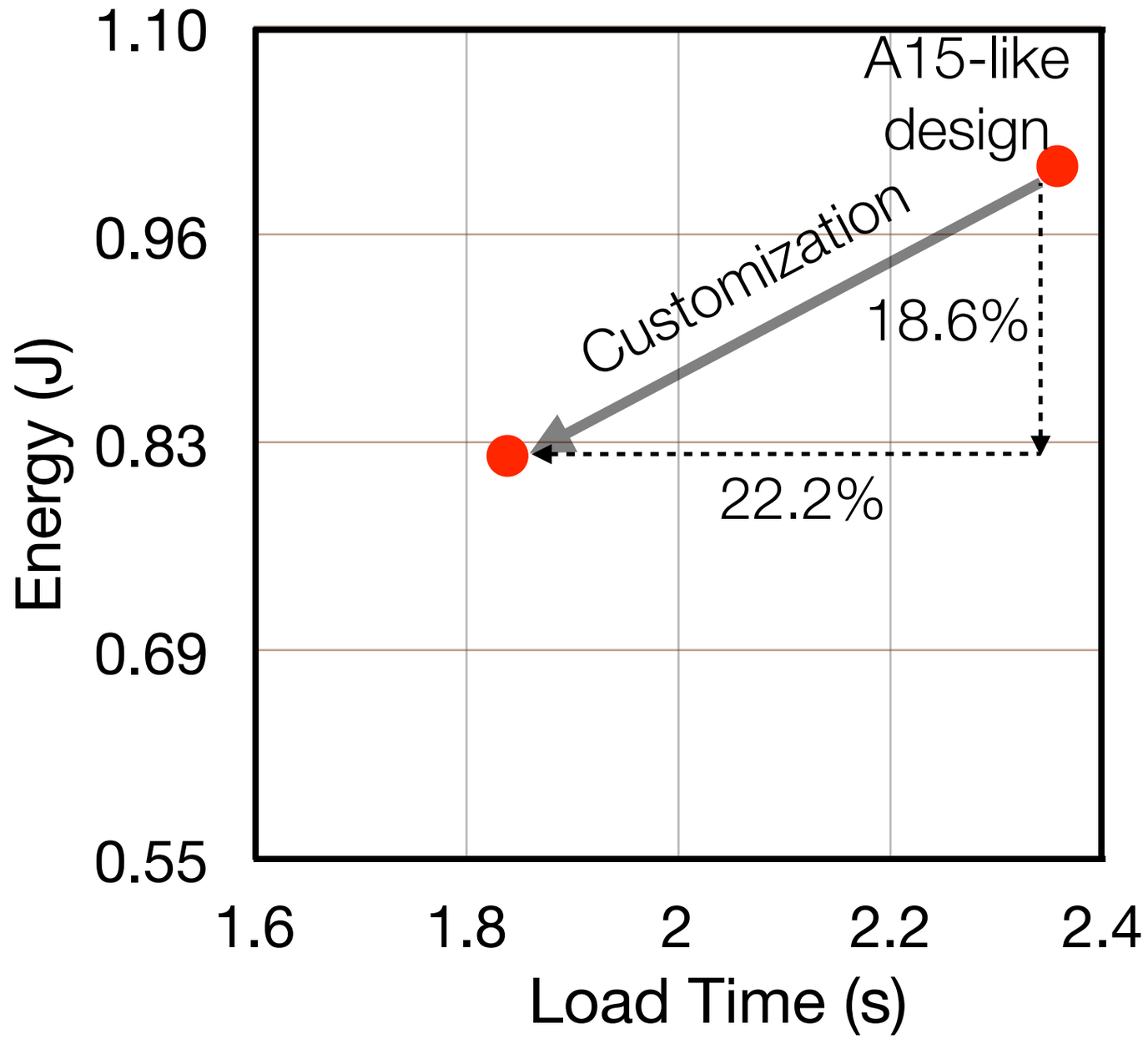
Evaluation Results

Fully synthesized using Synopsys 28 nm toolchain.

Compare against ARM A15, a typical mobile CPU.

Cost of specialization: **0.59 mm²** area overhead

- A15's area: **19 mm²**
- SoC die area is **122 mm²** in Samsung Galaxy S4



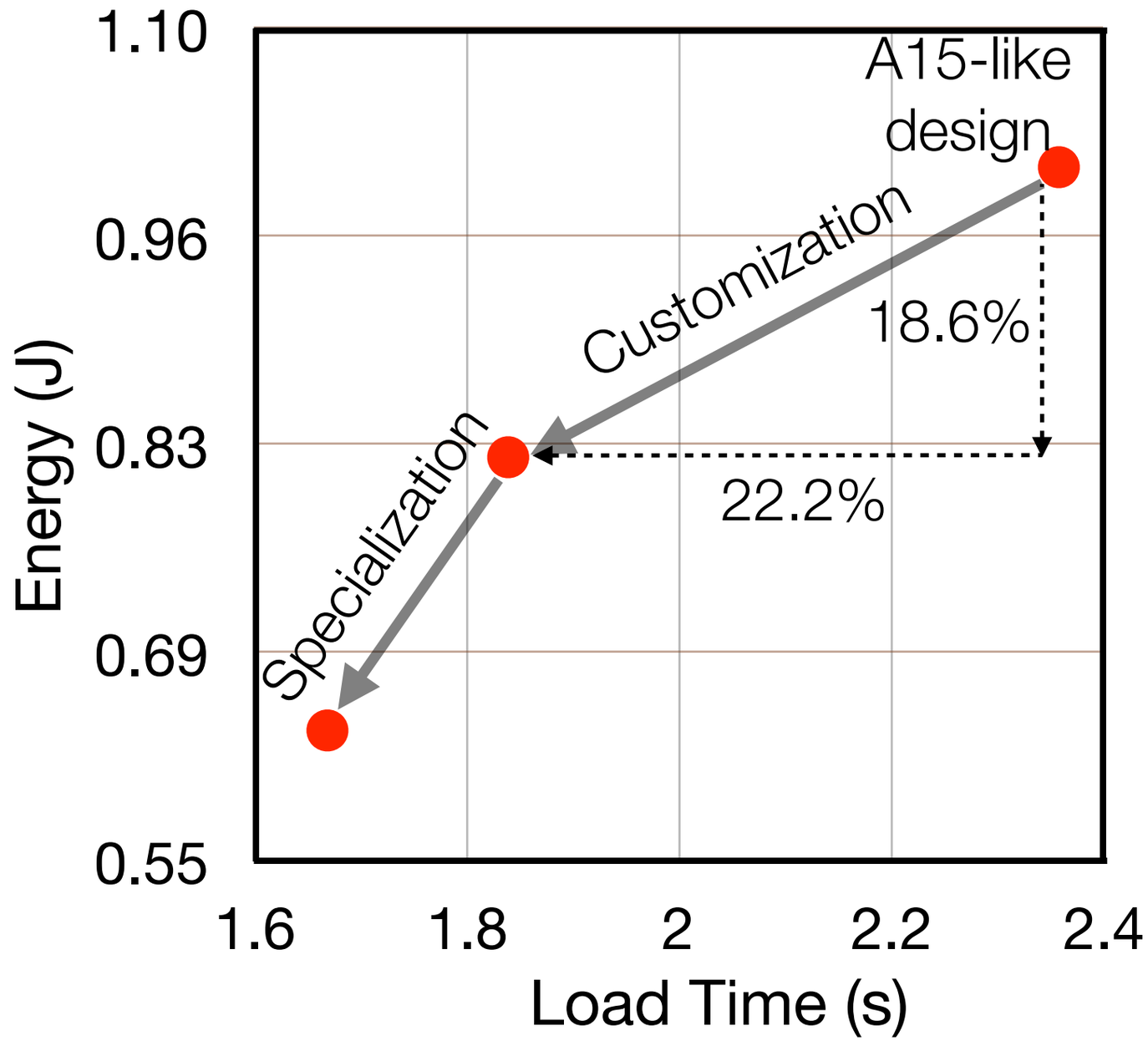
Evaluation Results

Fully synthesized using Synopsys 28 nm toolchain.

Compare against ARM A15, a typical mobile CPU.

Cost of specialization: **0.59 mm²** area overhead

- A15's area: **19 mm²**
- SoC die area is **122 mm²** in Samsung Galaxy S4



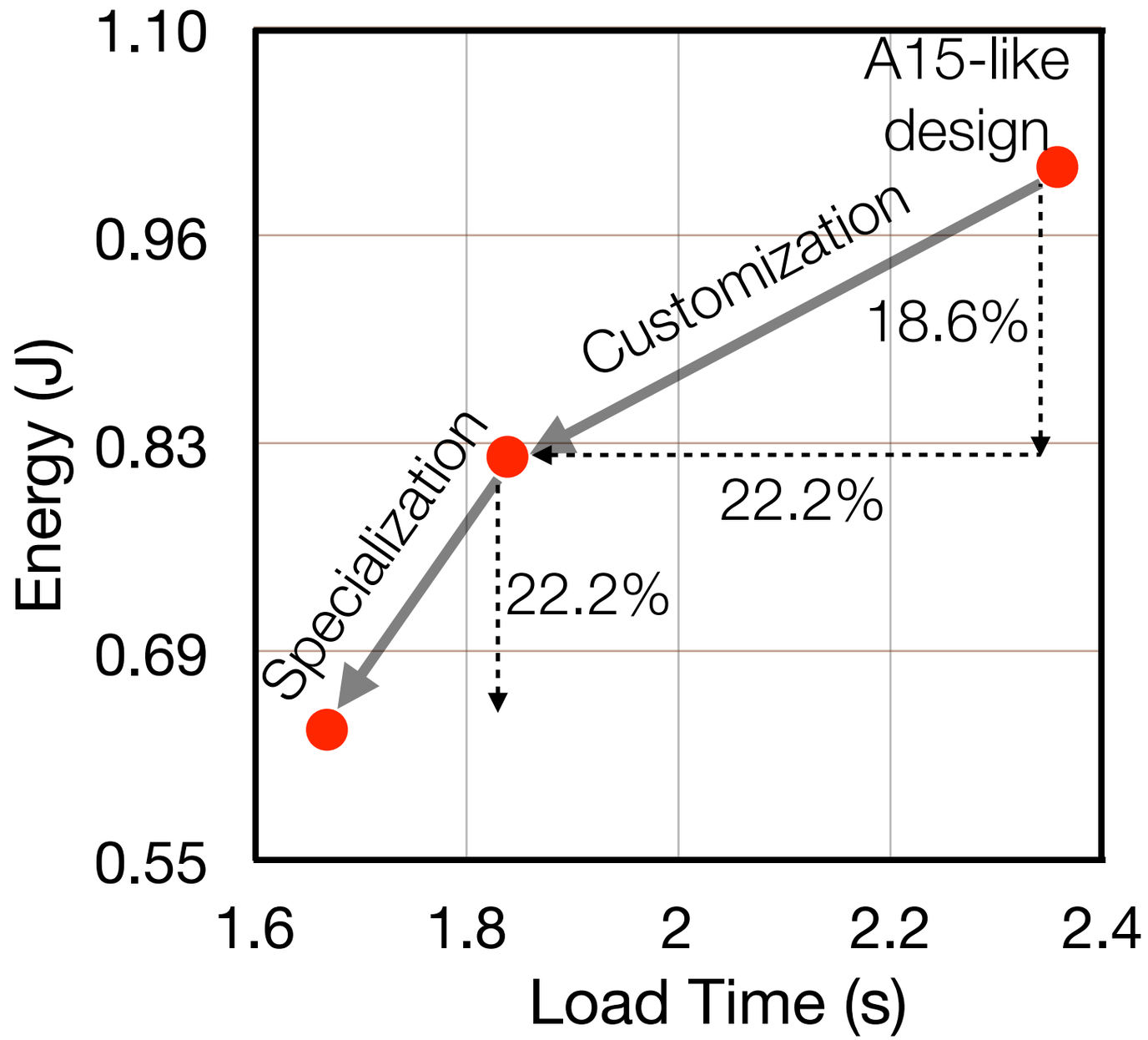
Evaluation Results

Fully synthesized using Synopsys 28 nm toolchain.

Compare against ARM A15, a typical mobile CPU.

Cost of specialization: **0.59 mm²** area overhead

- A15's area: **19 mm²**
- SoC die area is **122 mm²** in Samsung Galaxy S4



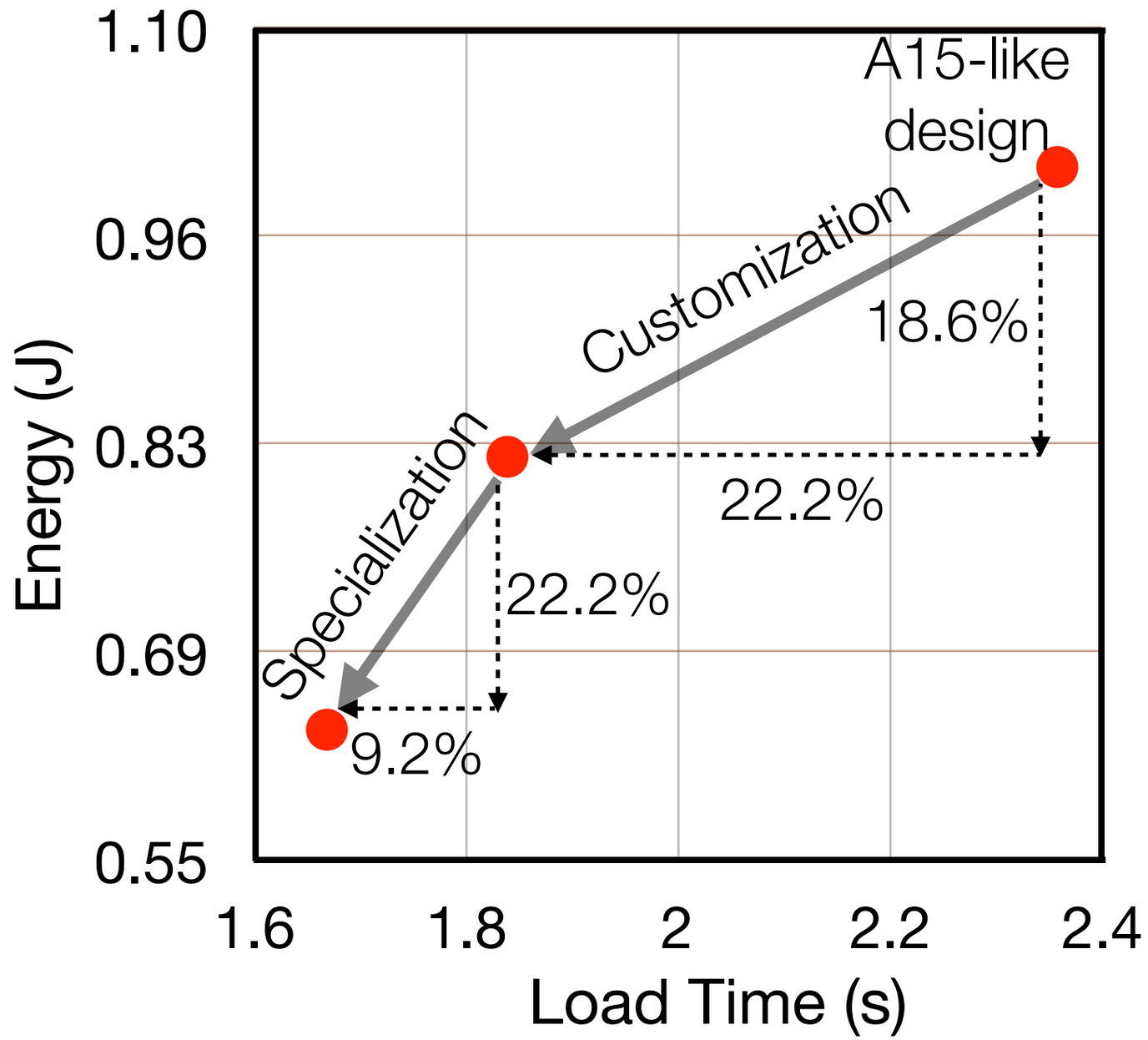
Evaluation Results

Fully synthesized using Synopsys 28 nm toolchain.

Compare against ARM A15, a typical mobile CPU.

Cost of specialization: **0.59 mm²** area overhead

- A15's area: **19 mm²**
- SoC die area is **122 mm²** in Samsung Galaxy S4



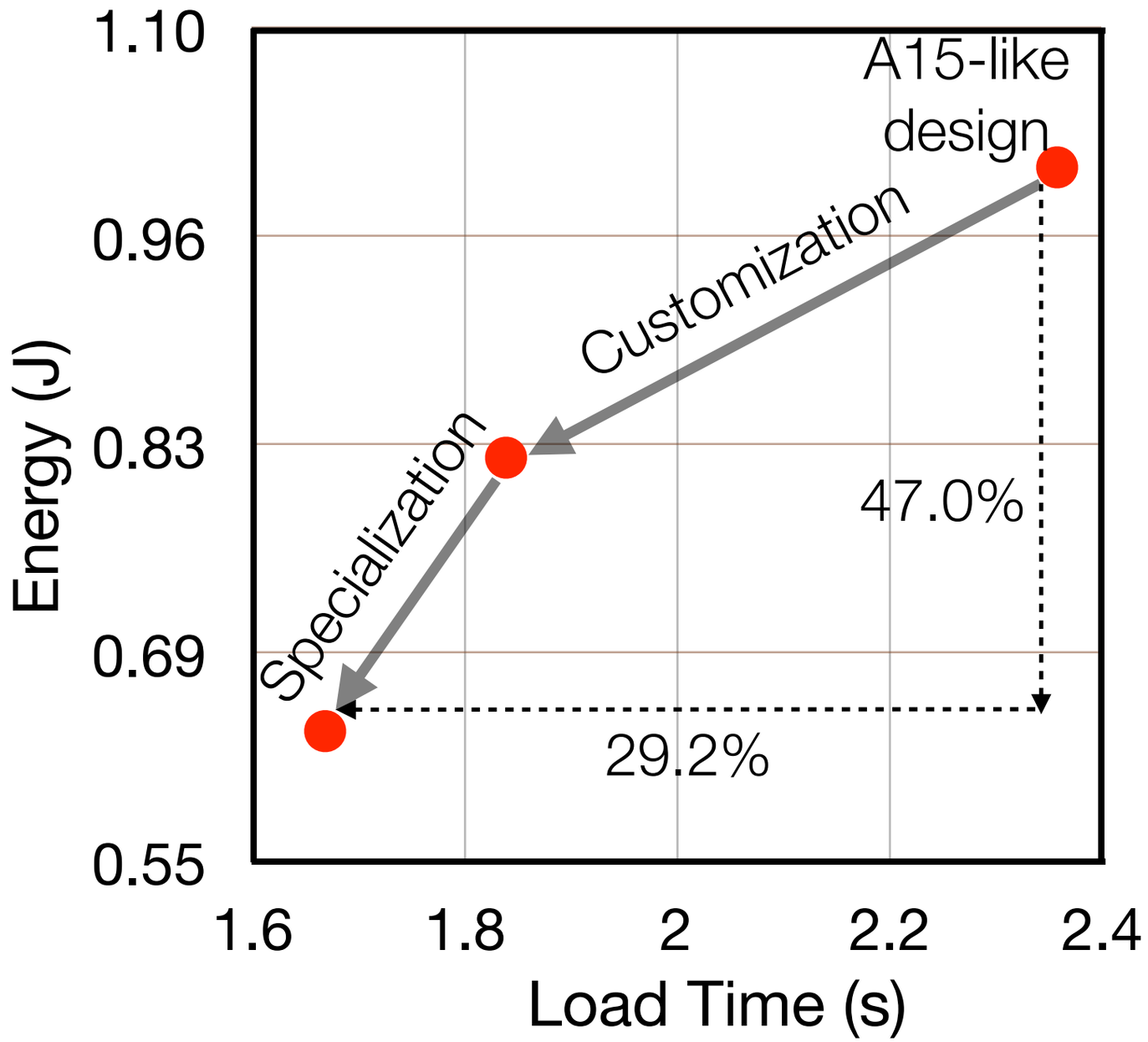
Evaluation Results

Fully synthesized using Synopsys 28 nm toolchain.

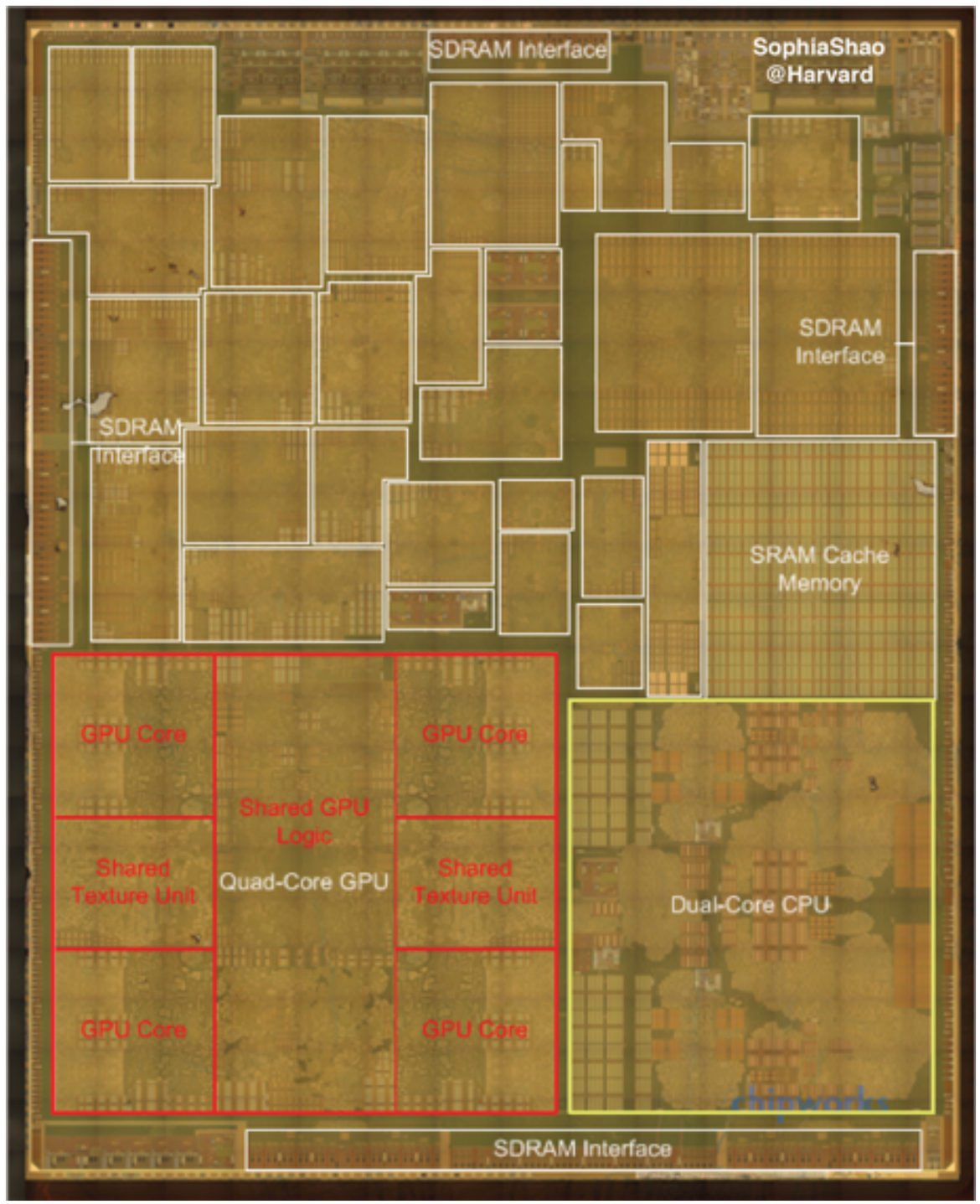
Compare against ARM A15, a typical mobile CPU.

Cost of specialization: **0.59 mm²** area overhead

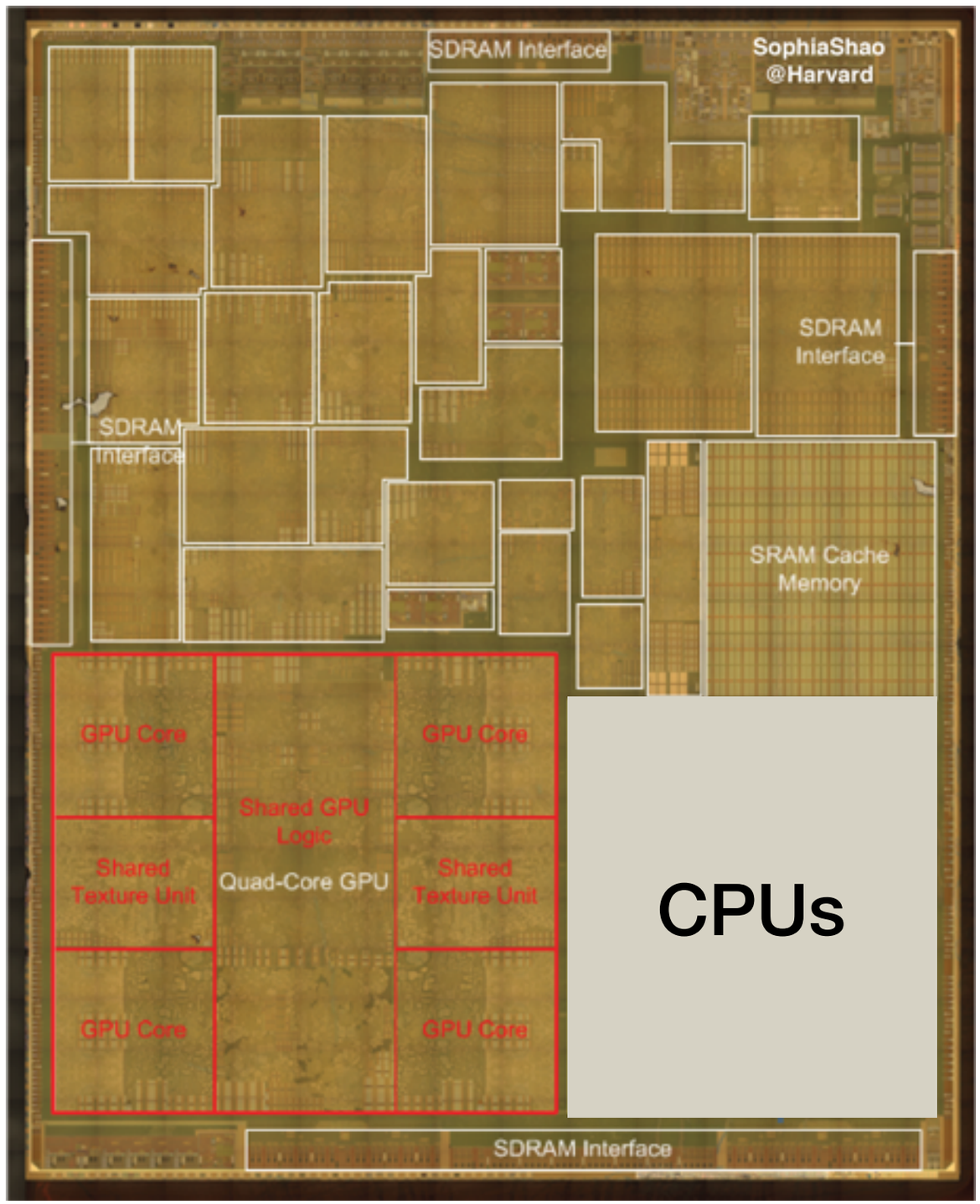
- A15's area: **19 mm²**
- SoC die area is **122 mm²** in Samsung Galaxy S4



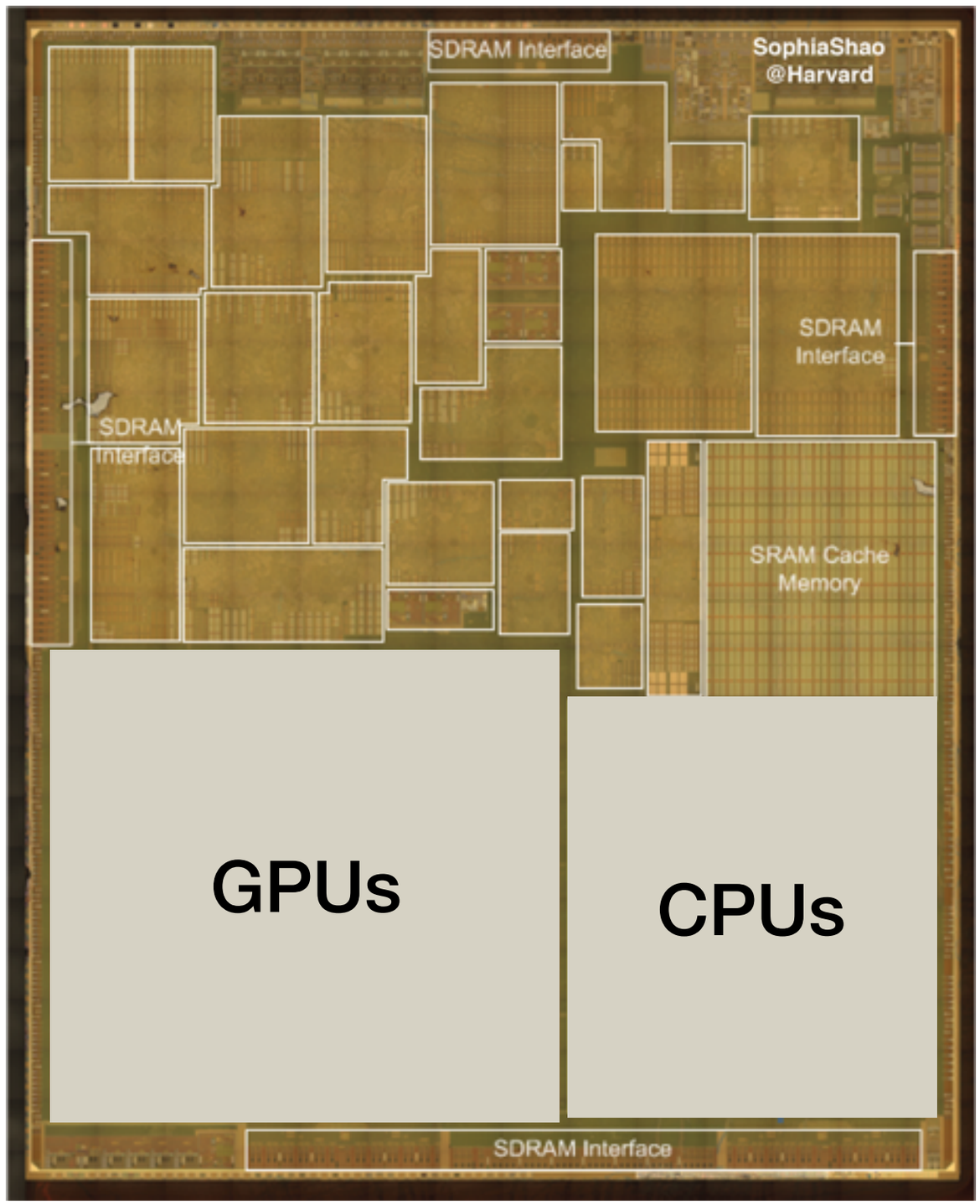
How Does **WebCore** Fit



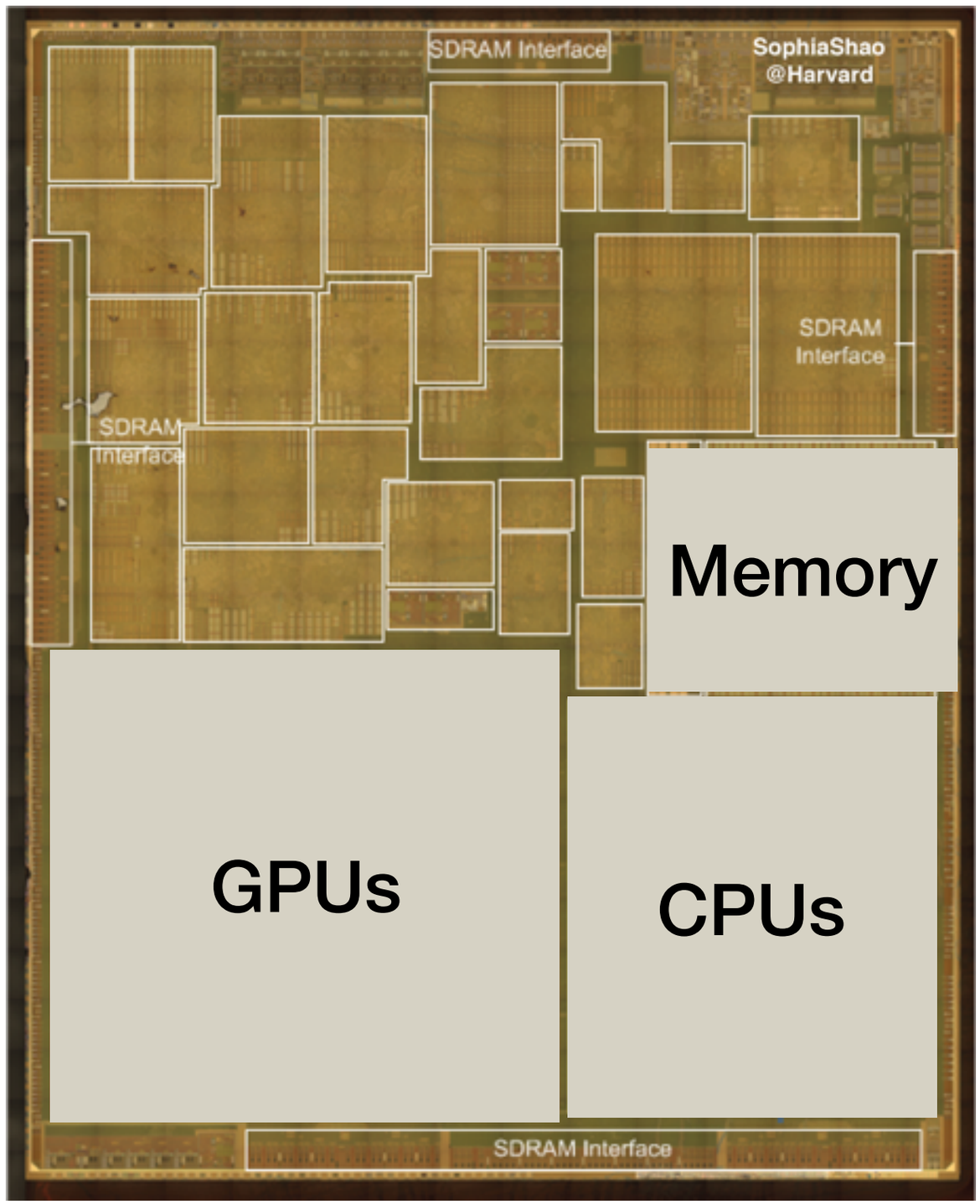
How Does **WebCore** Fit



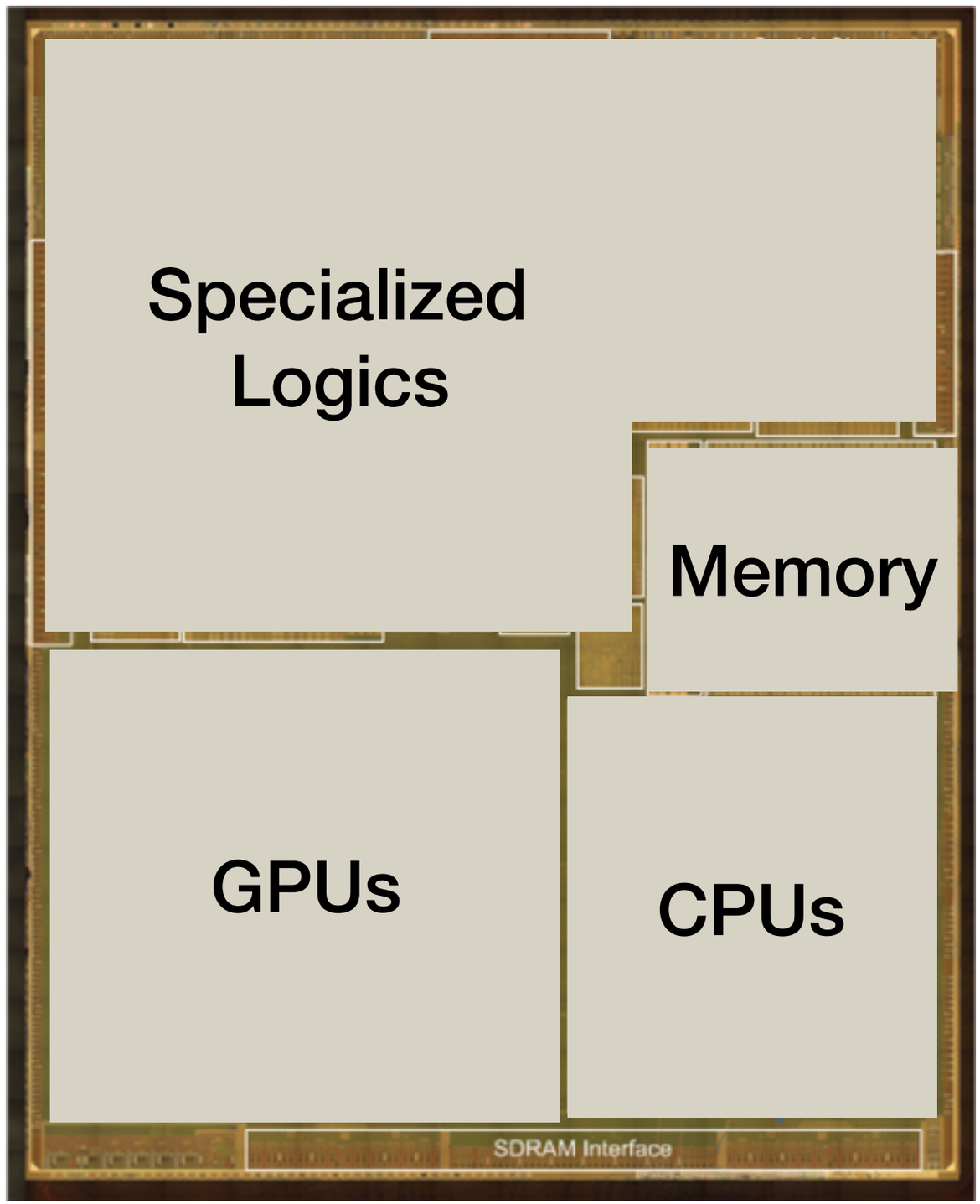
How Does **WebCore** Fit



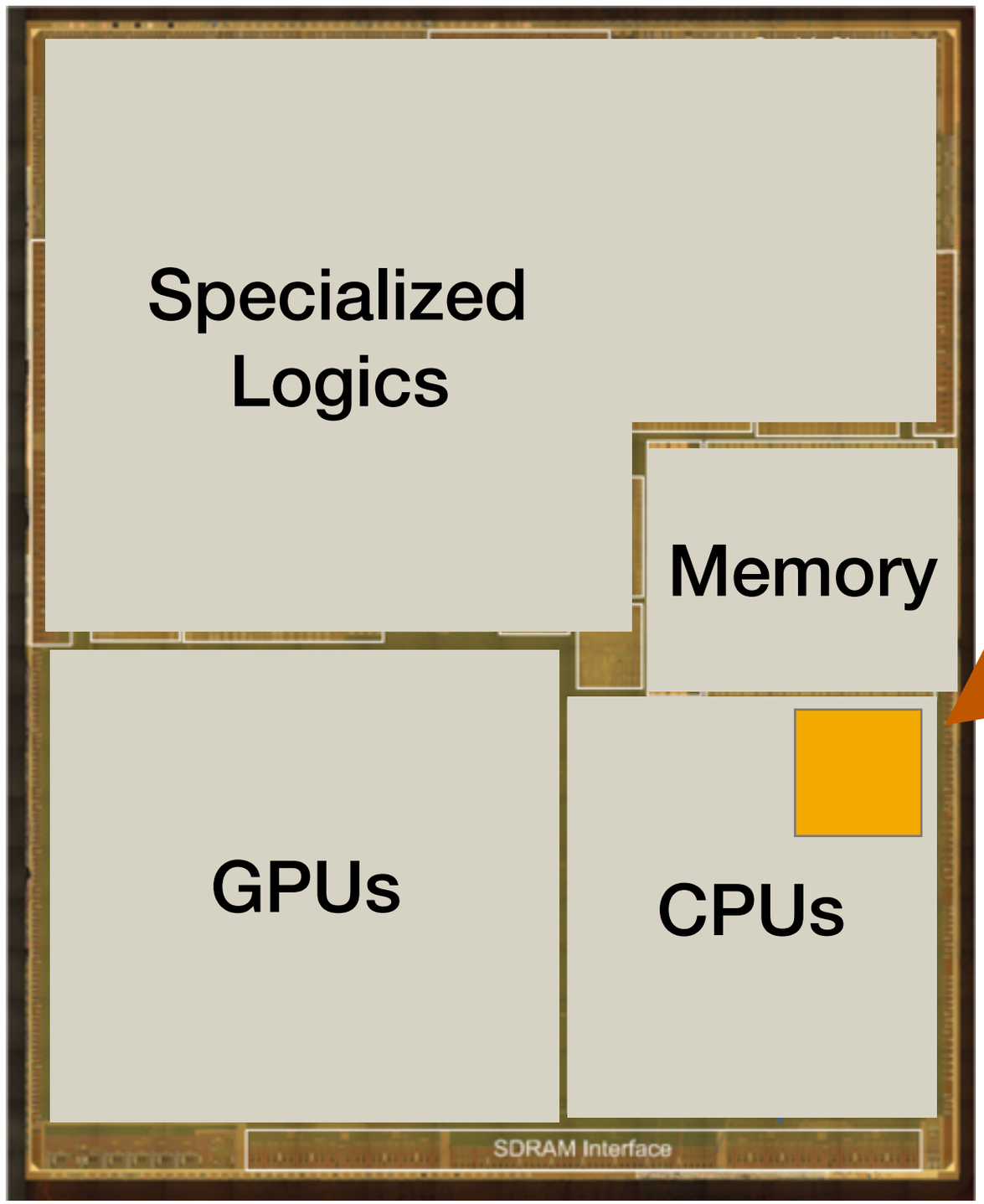
How Does **WebCore** Fit



How Does **WebCore** Fit



How Does **WebCore** Fit



WebCore is one of the cores in the multicore SoC.

Mobile Web Computation Stack

Application

GreenWeb

Language Support

Runtime

WebRT

Energy-aware

Web Runtime

Architecture

WebCore

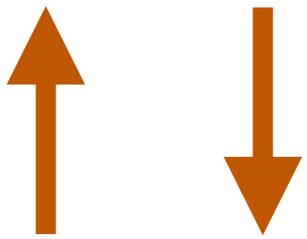
Web-specific Architecture

Mobile Web Computation Stack

Application

GreenWeb

Language Support



Runtime

WebRT

Energy-aware

Web Runtime

Architecture

WebCore

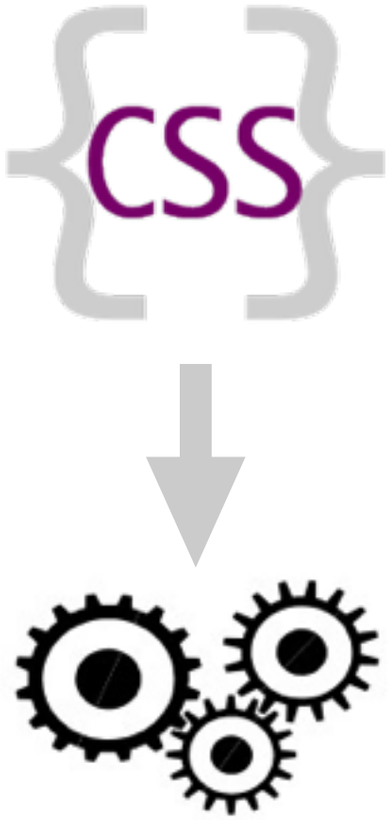
Web-specific Architecture

GreenWeb: Language for Energy-Efficiency

Abstractions

Express QoS constraints

GreenWeb: Language for Energy-Efficiency



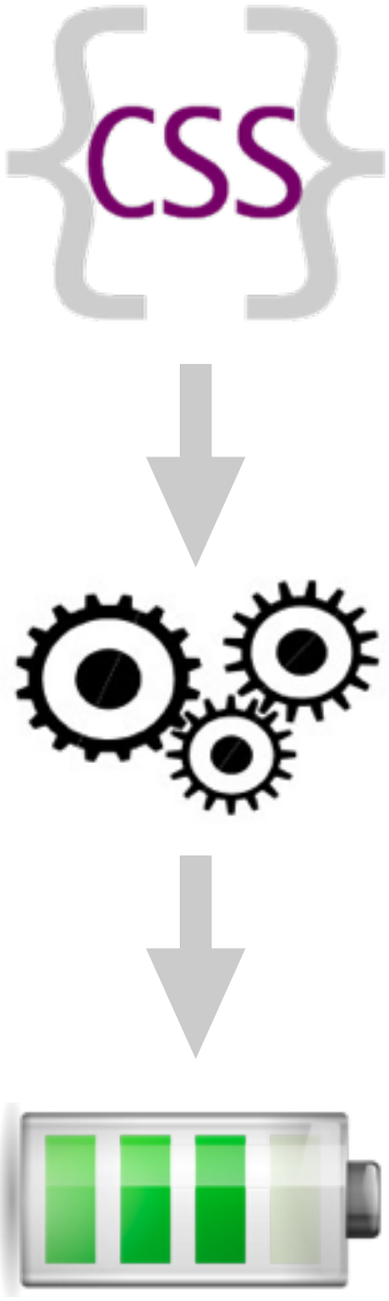
Abstractions

Express QoS constraints

Runtime

Satisfy QoS specifications
while saving energy

GreenWeb: Language for Energy-Efficiency



Abstractions Express QoS constraints

Runtime Satisfy QoS specifications while saving energy

Effect 60% energy savings on real system implementations

GreenWeb: Language for Energy-Efficiency

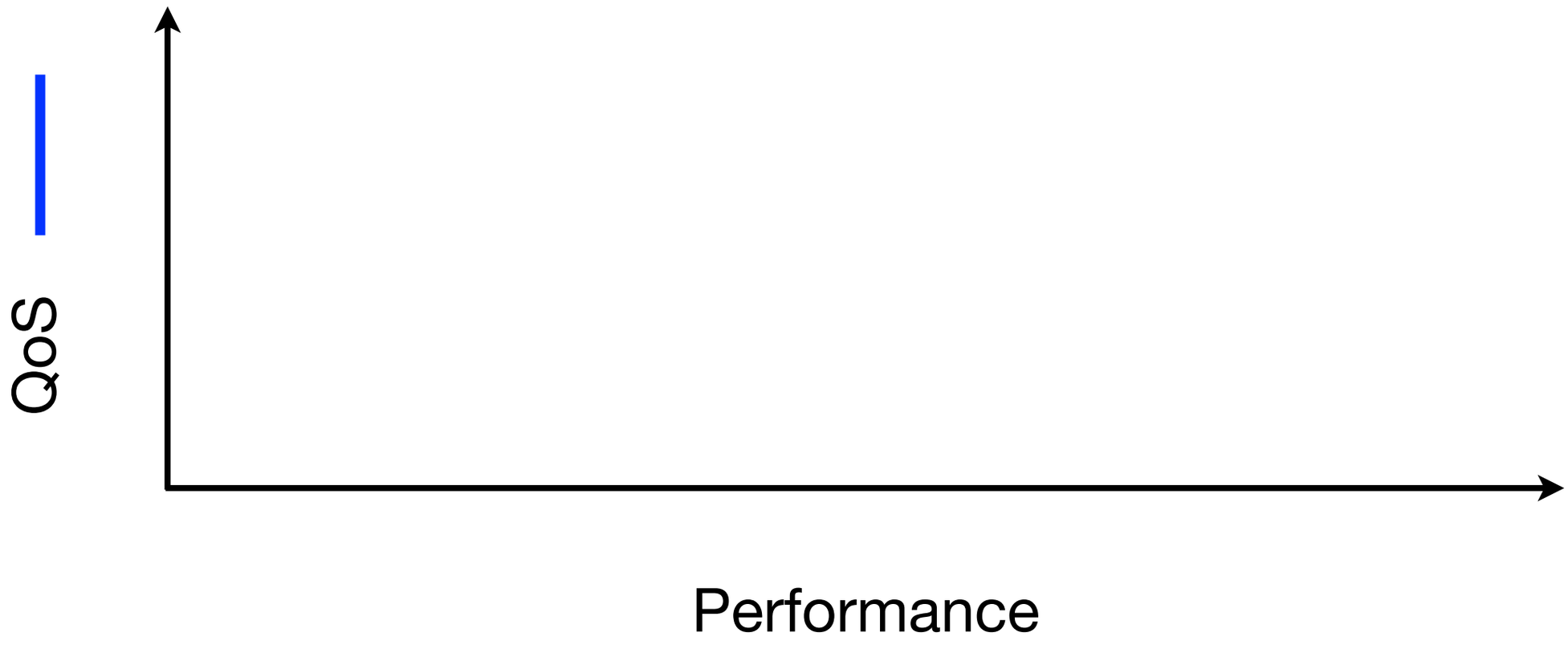
Abstractions — Express QoS constraints

Runtime — Satisfy QoS specifications while saving energy

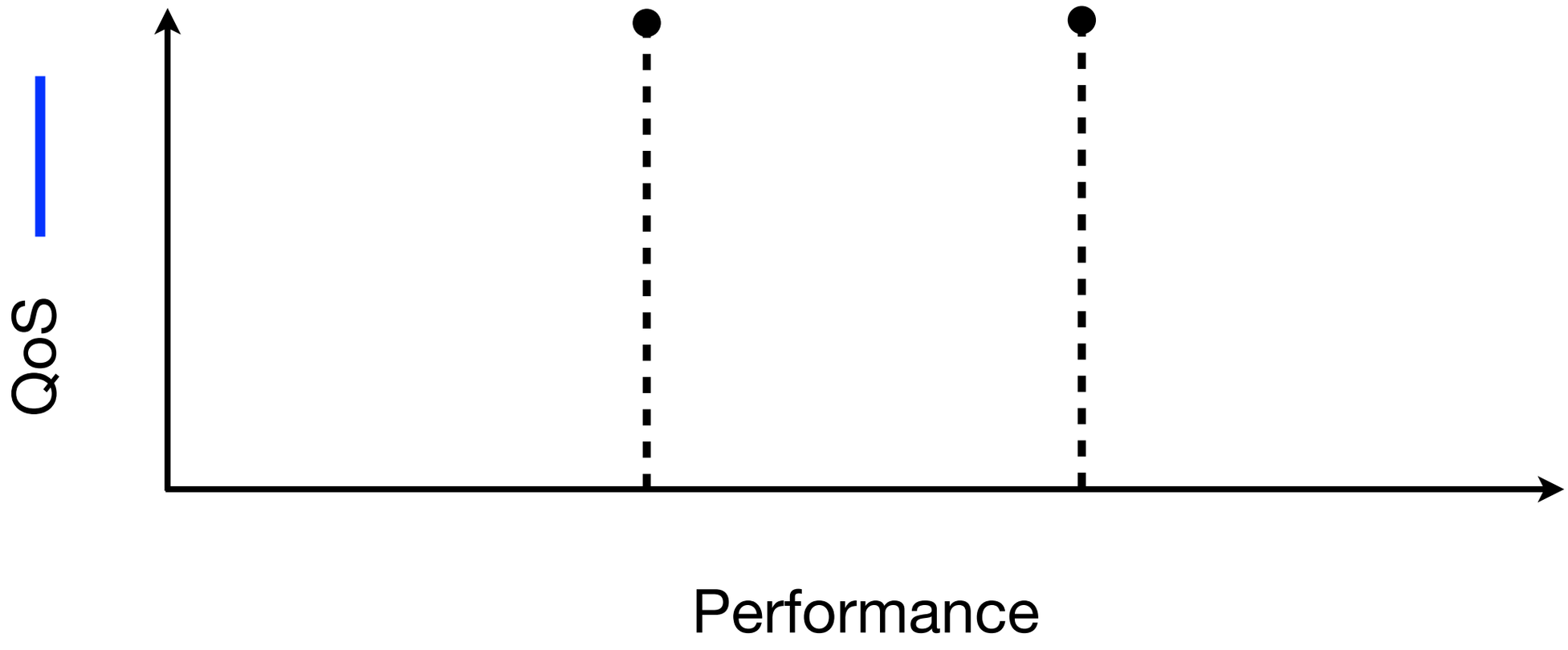
Effect — 60% energy savings on real system implementations

Understanding Mobile Web QoS

Understanding Mobile Web QoS

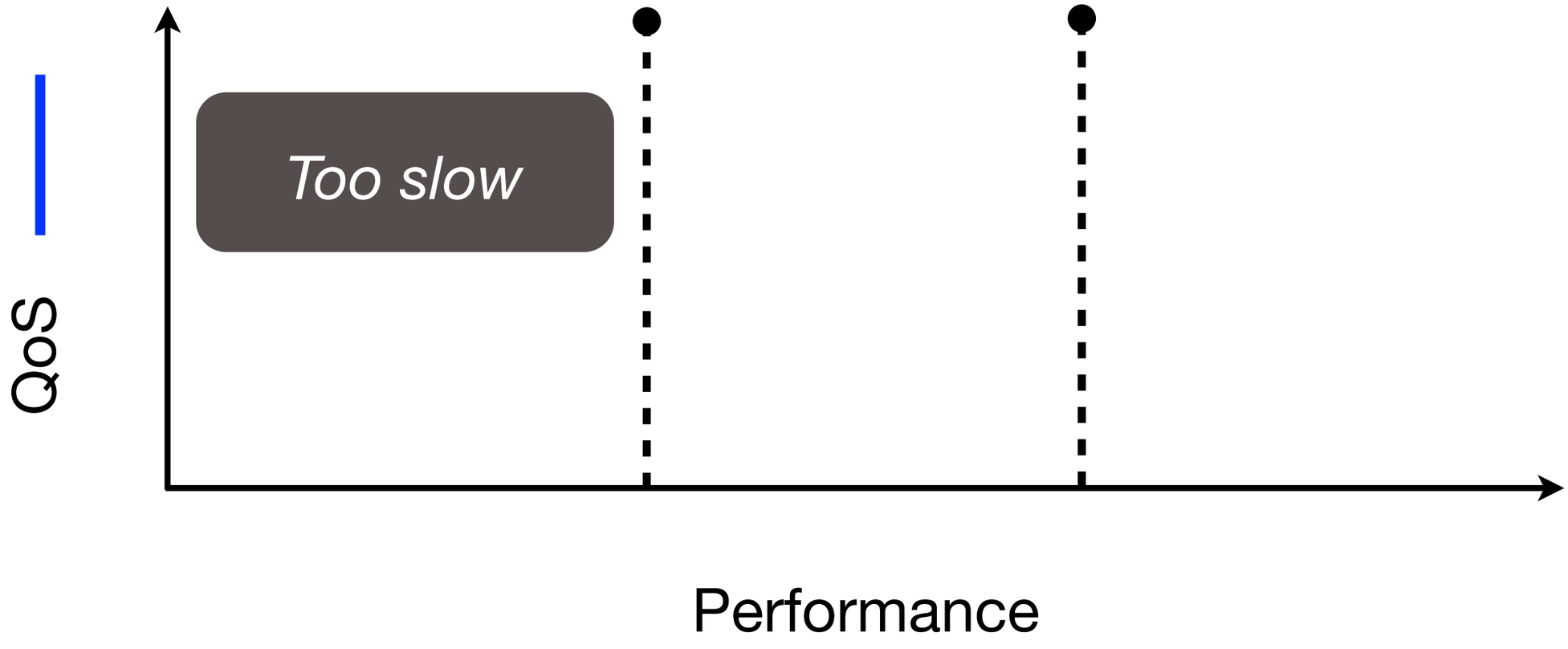


Understanding Mobile Web QoS



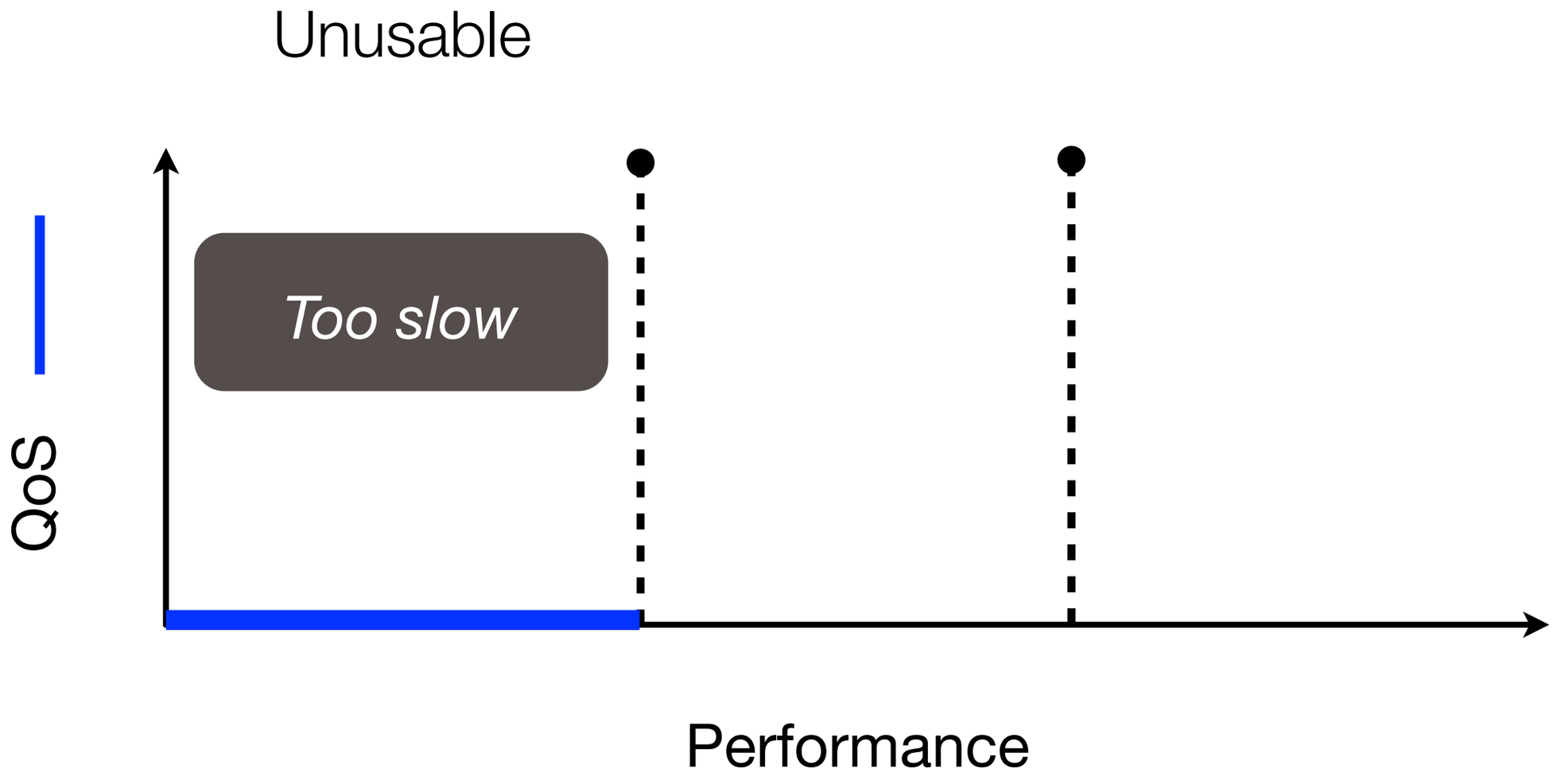
[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

Understanding Mobile Web QoS



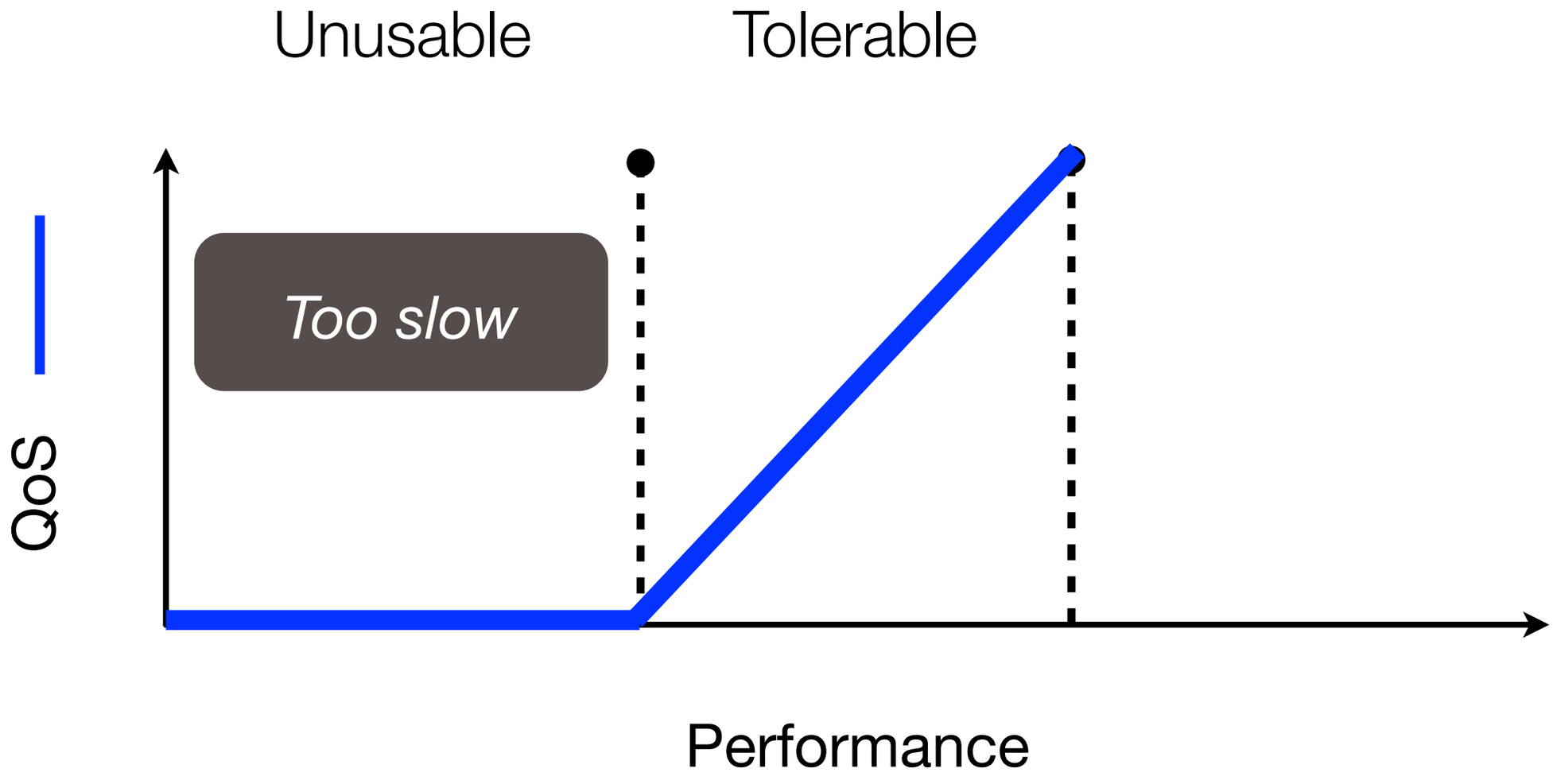
[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

Understanding Mobile Web QoS



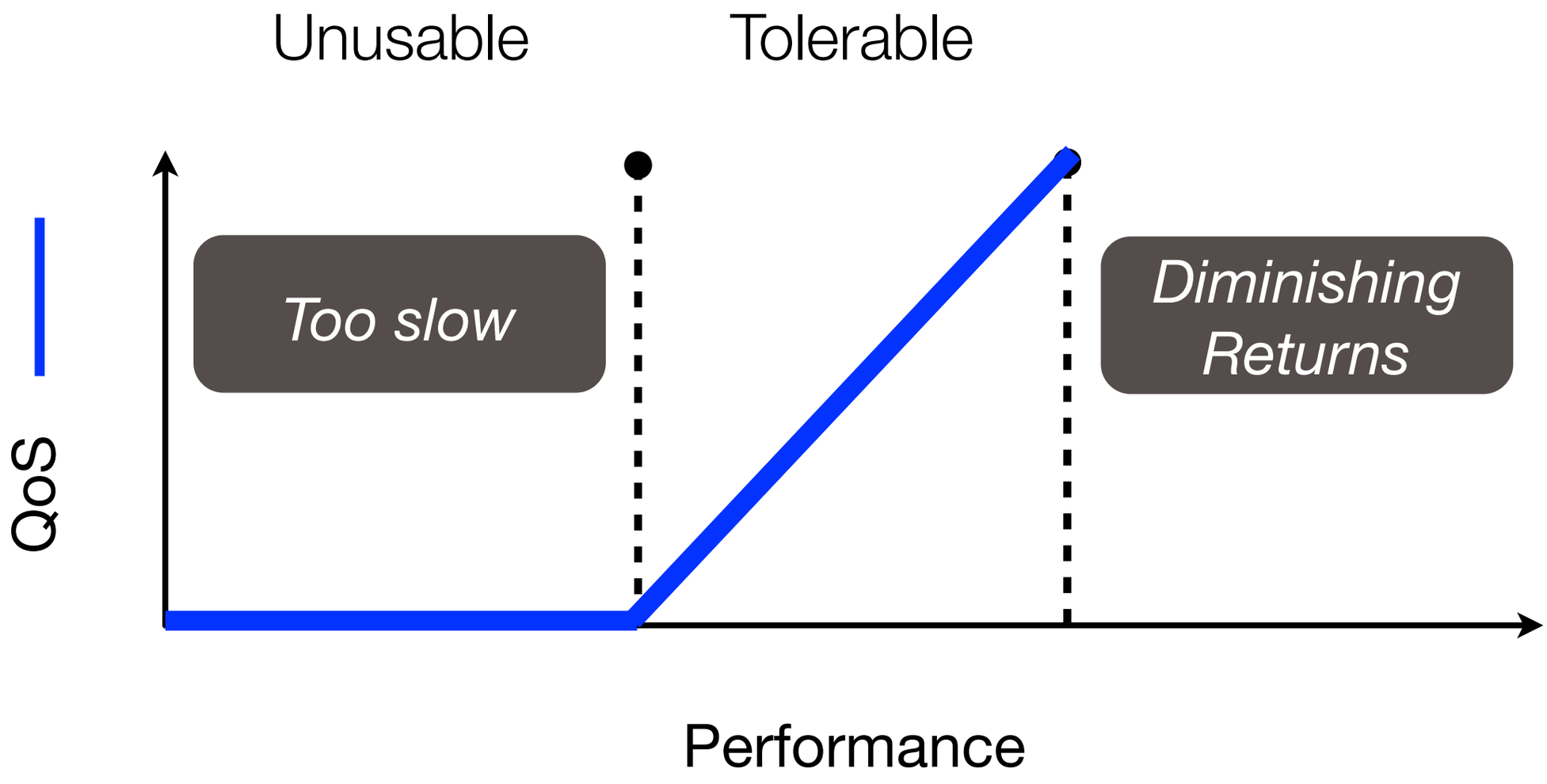
[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

Understanding Mobile Web QoS



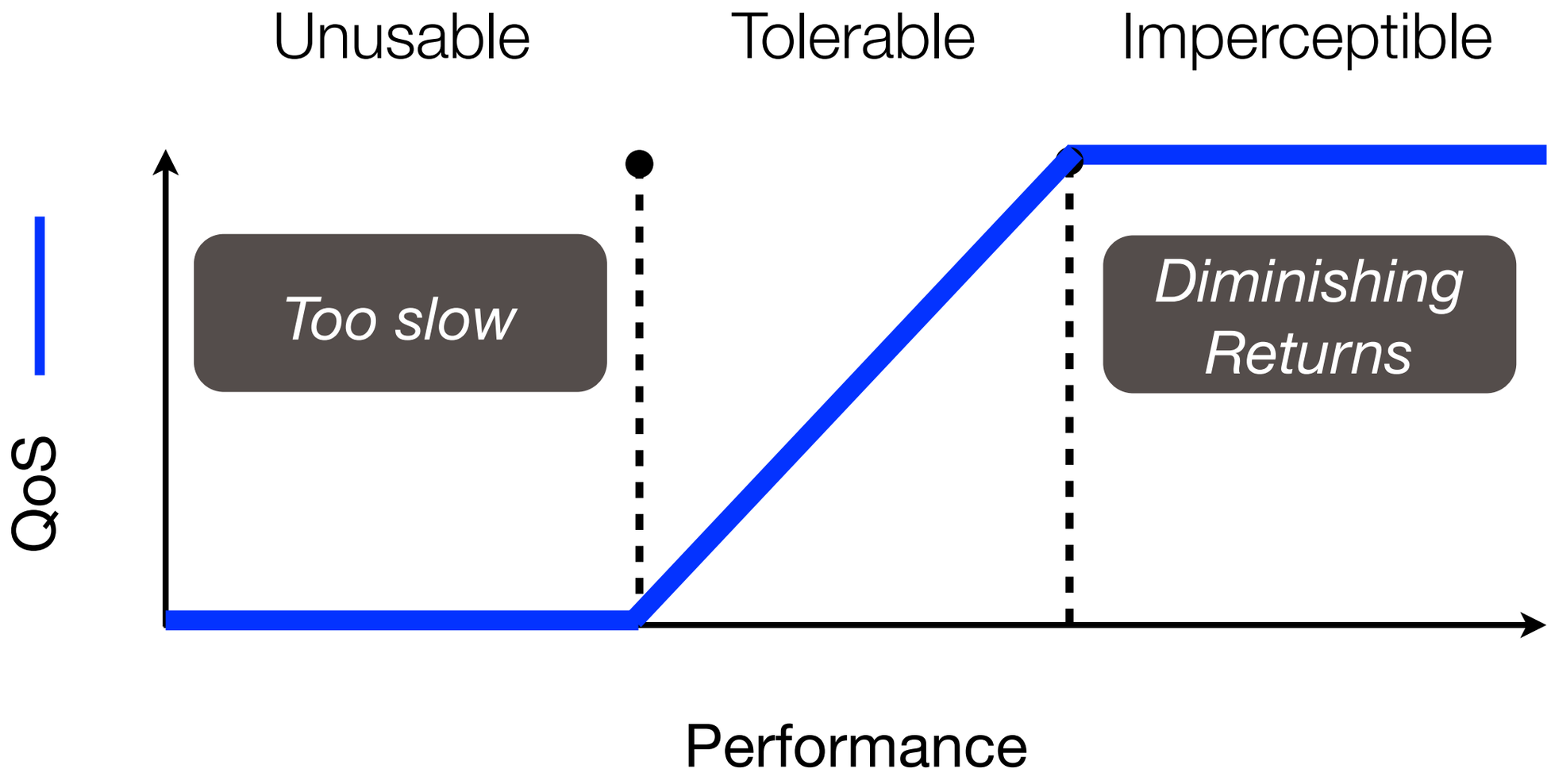
[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

Understanding Mobile Web QoS



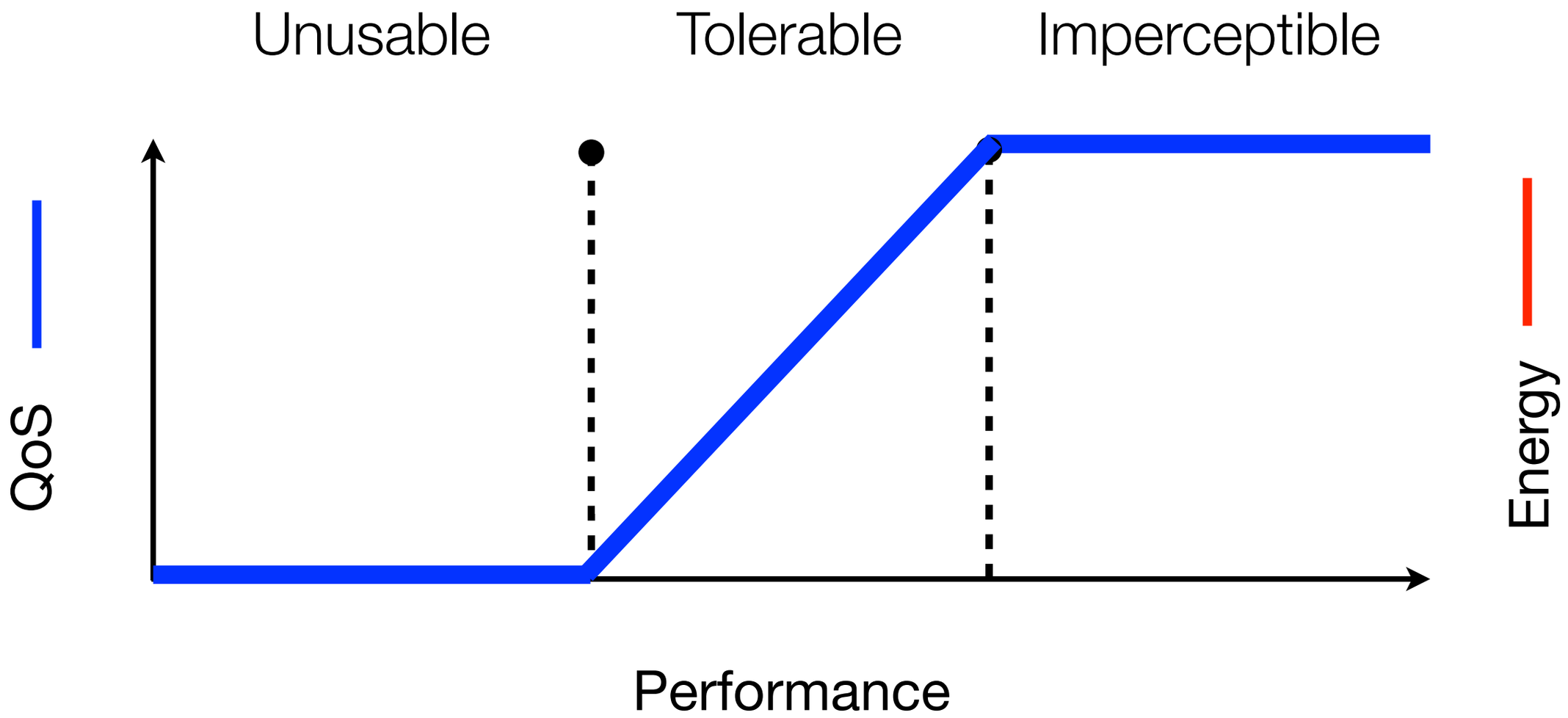
[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

Understanding Mobile Web QoS



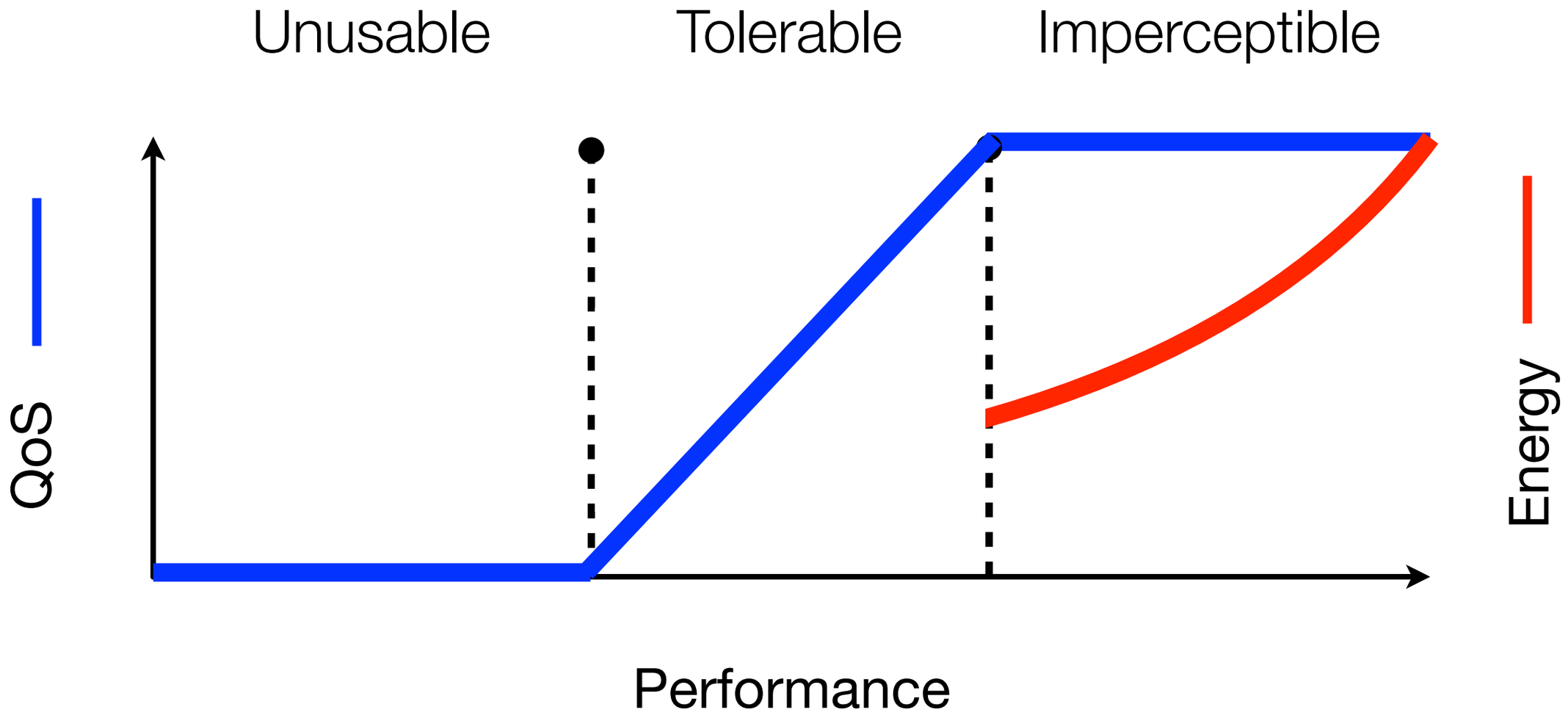
[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

Understanding Mobile Web QoS



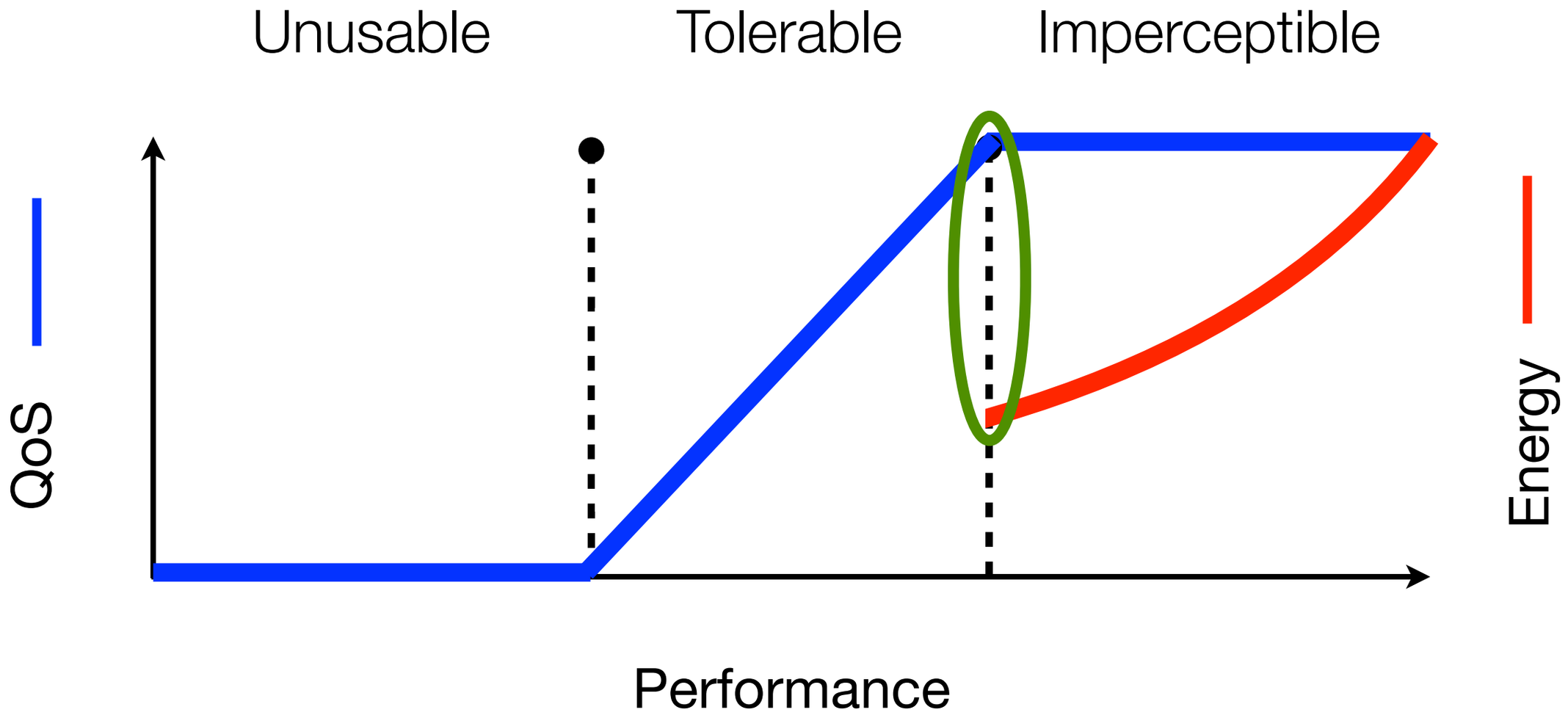
[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

Understanding Mobile Web QoS



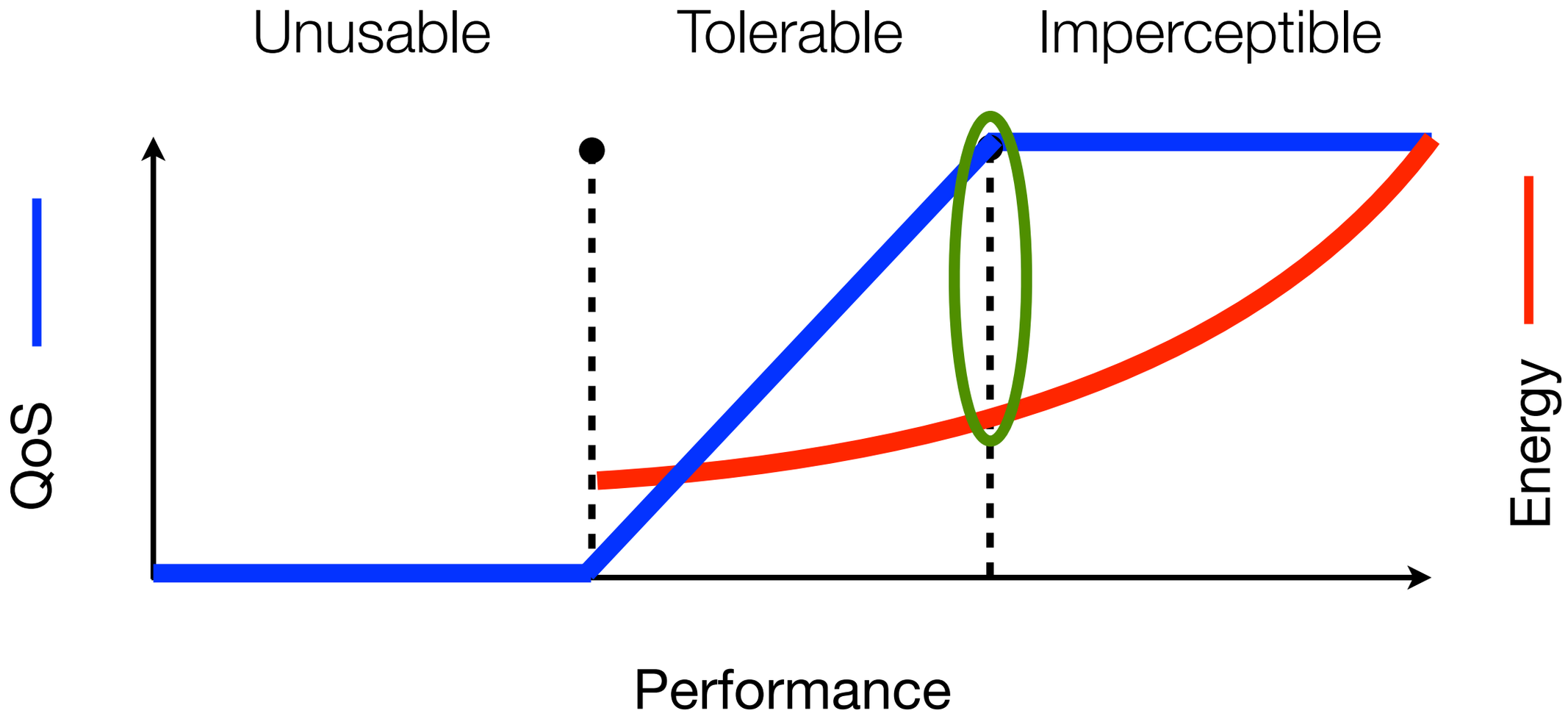
[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

Understanding Mobile Web QoS



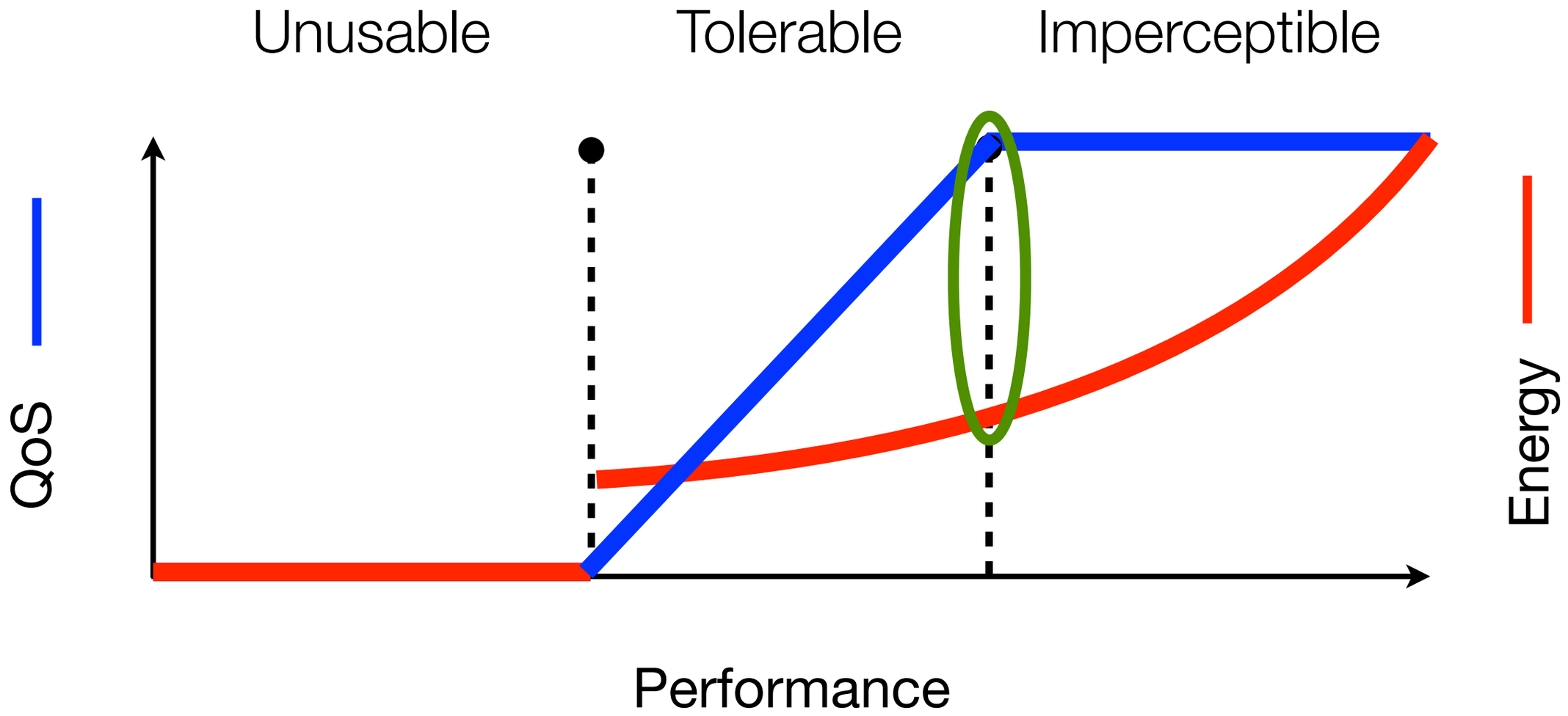
[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

Understanding Mobile Web QoS



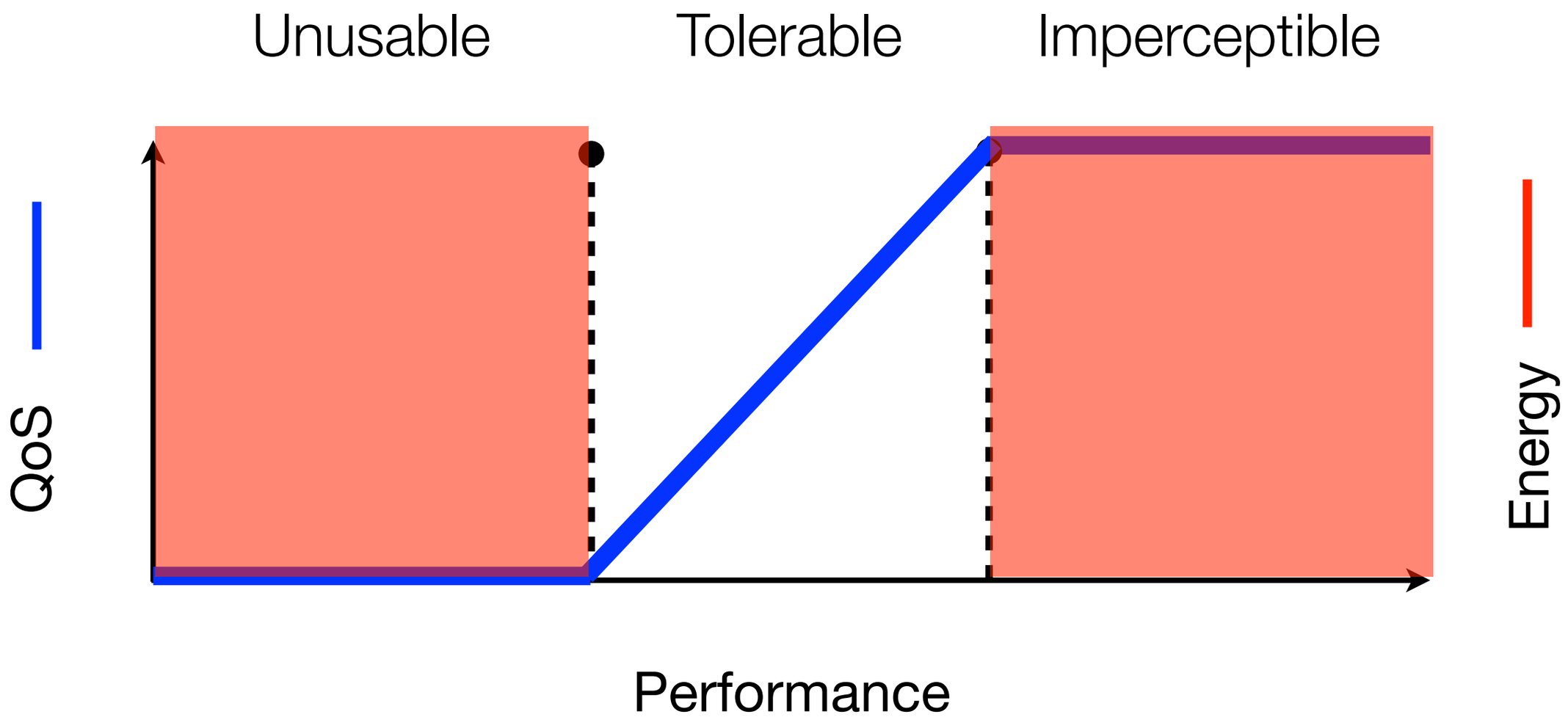
[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

Understanding Mobile Web QoS



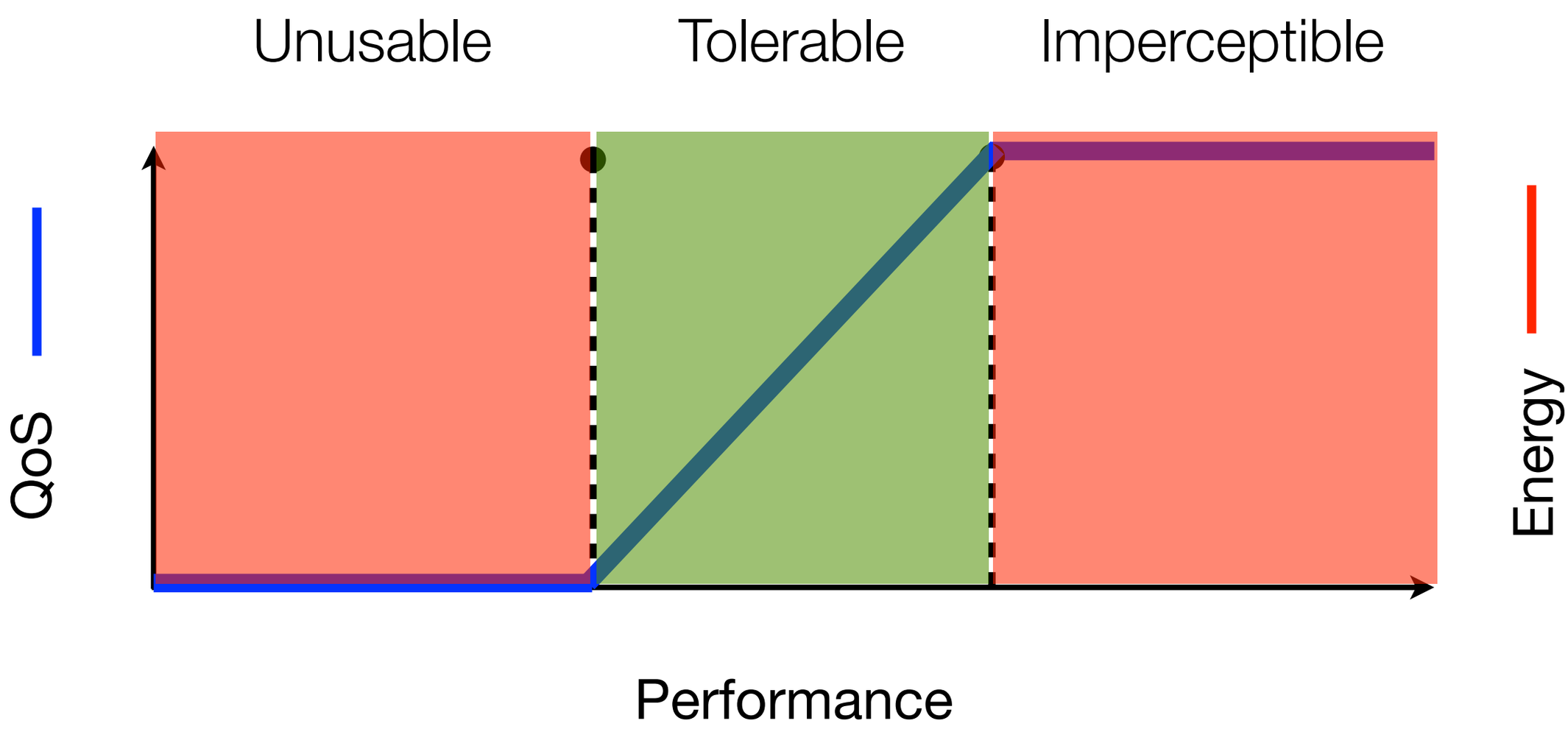
[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

Understanding Mobile Web QoS



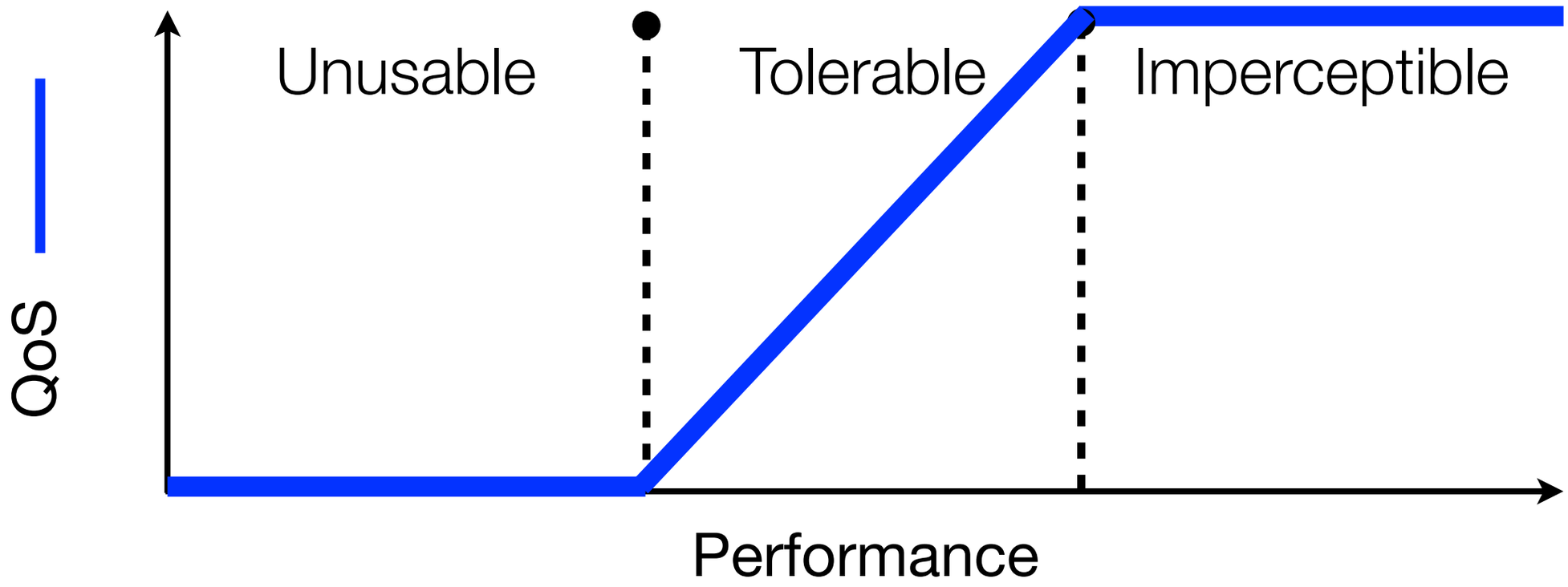
[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

Understanding Mobile Web QoS



[OSDI 1996] Y. Endo et al., “Using Latency to Evaluate Interactive System Performance.”

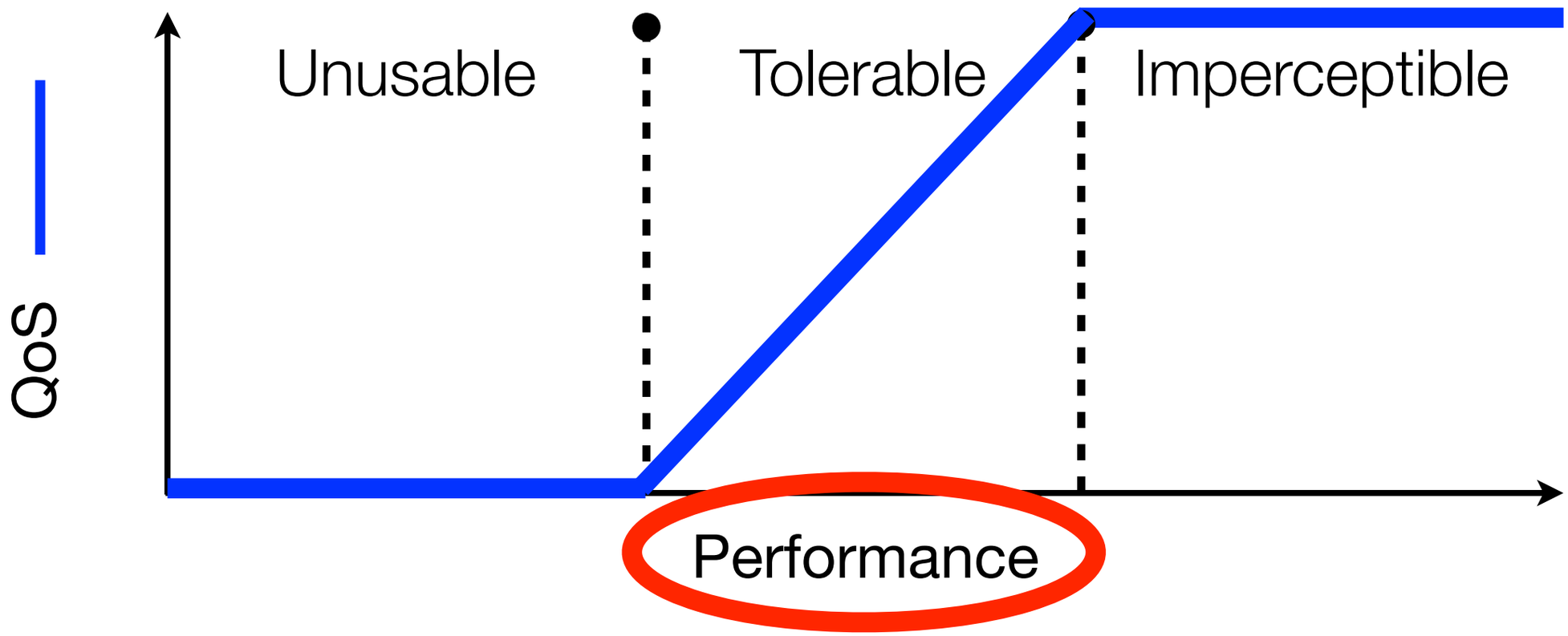
Abstracting Mobile Web QoS



Abstracting Mobile Web QoS

Performance metric

Frame latency or Frame throughput?

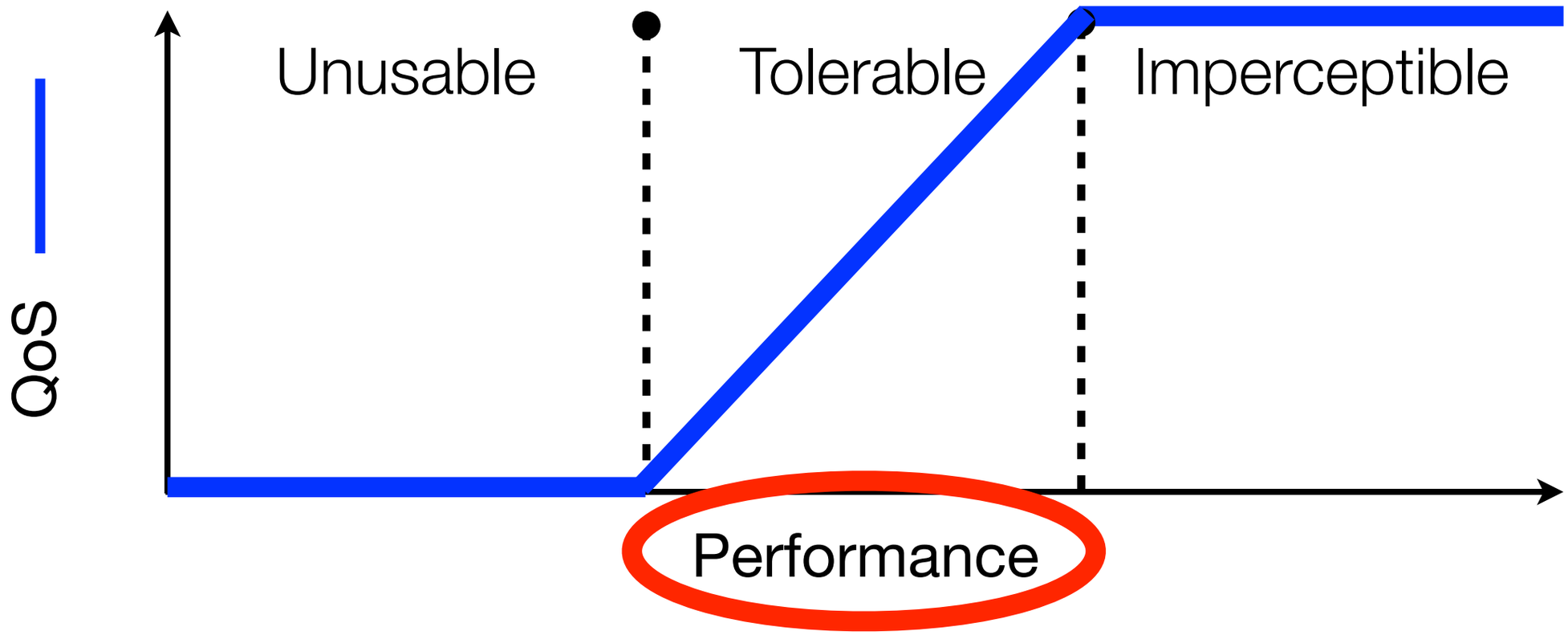


Abstracting Mobile Web QoS

QoS Type

Performance metric

Frame latency or Frame throughput?



Abstracting Mobile Web QoS

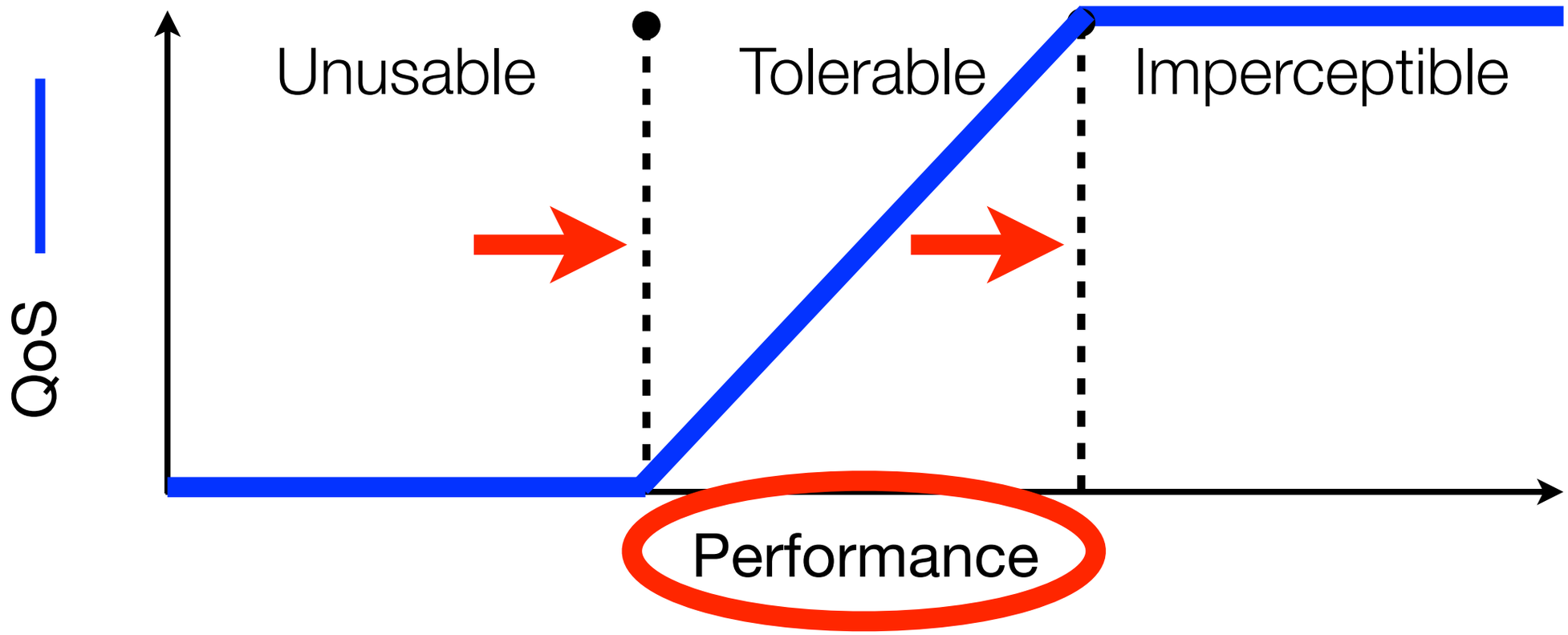
QoS Type

Performance metric

Frame latency or Frame throughput?

Threshold performance

Imperceptible or Usable?



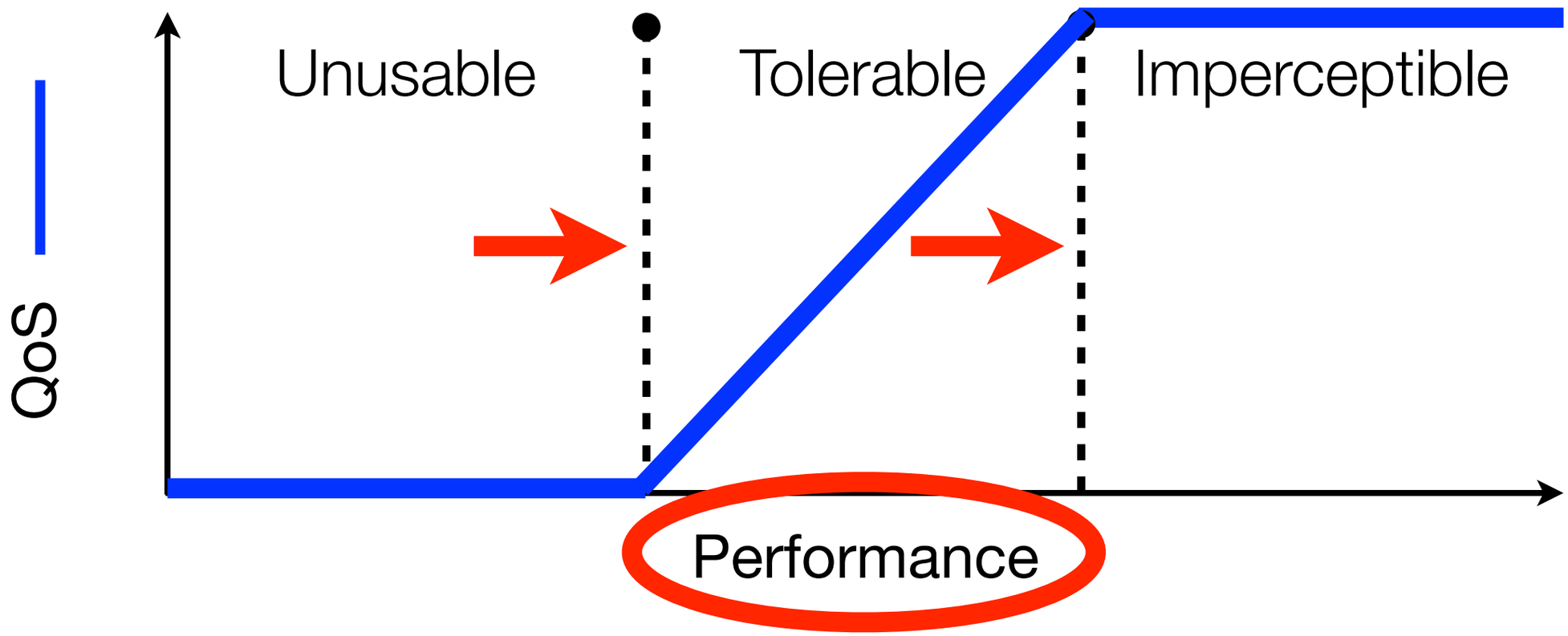
Abstracting Mobile Web QoS

QoS Type — Performance metric

Frame latency or Frame throughput?

QoS Target — Threshold performance

Imperceptible or Usable?



Abstracting Mobile Web QoS

QoS Type

Performance metric

Frame latency or Frame throughput?

QoS Target

Threshold performance

Imperceptible or Usable?

Abstracting Mobile Web QoS

QoS Type

Performance metric

Frame latency or Frame throughput?

QoS Target

Threshold performance

Imperceptible or Usable?

Extend Cascading Style Sheet (CSS) Language

Abstracting Mobile Web QoS

QoS Type

Performance metric

Frame latency or Frame throughput?

QoS Target

Threshold performance

Imperceptible or Usable?

Extend Cascading Style Sheet (CSS) Language


```
element {style prop: value}
```

Abstracting Mobile Web QoS

QoS Type — Performance metric

Frame latency or Frame throughput?

QoS Target — Threshold performance

Imperceptible or Usable?

Extend Cascading Style Sheet (CSS) Language


```
element {style prop: value}
```

```
element {event: type, target}
```

GreenWeb: Language for Energy-Efficiency

Abstractions — Express QoS constraints

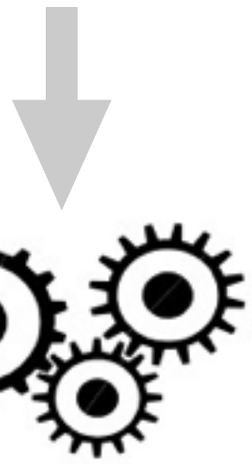
Runtime — Satisfy QoS specifications while saving energy

Effect — 60% energy savings on real system implementations

GreenWeb: Language for Energy-Efficiency

Abstractions

Express QoS



Runtime

Satisfy QoS specifications while saving energy

Effect

60% energy savings on real system implementations

Energy-aware Mobile Web Runtime

GreenWeb can support a range of energy saving techniques

Energy-aware Mobile Web Runtime

GreenWeb can support a range of energy saving techniques

- Dynamic resolution scaling [MobiCom 2015]
- Power-saving display colors [MobiSys 2012]
- Selective resource loading [NSDI 2015]

Energy-aware Mobile Web Runtime

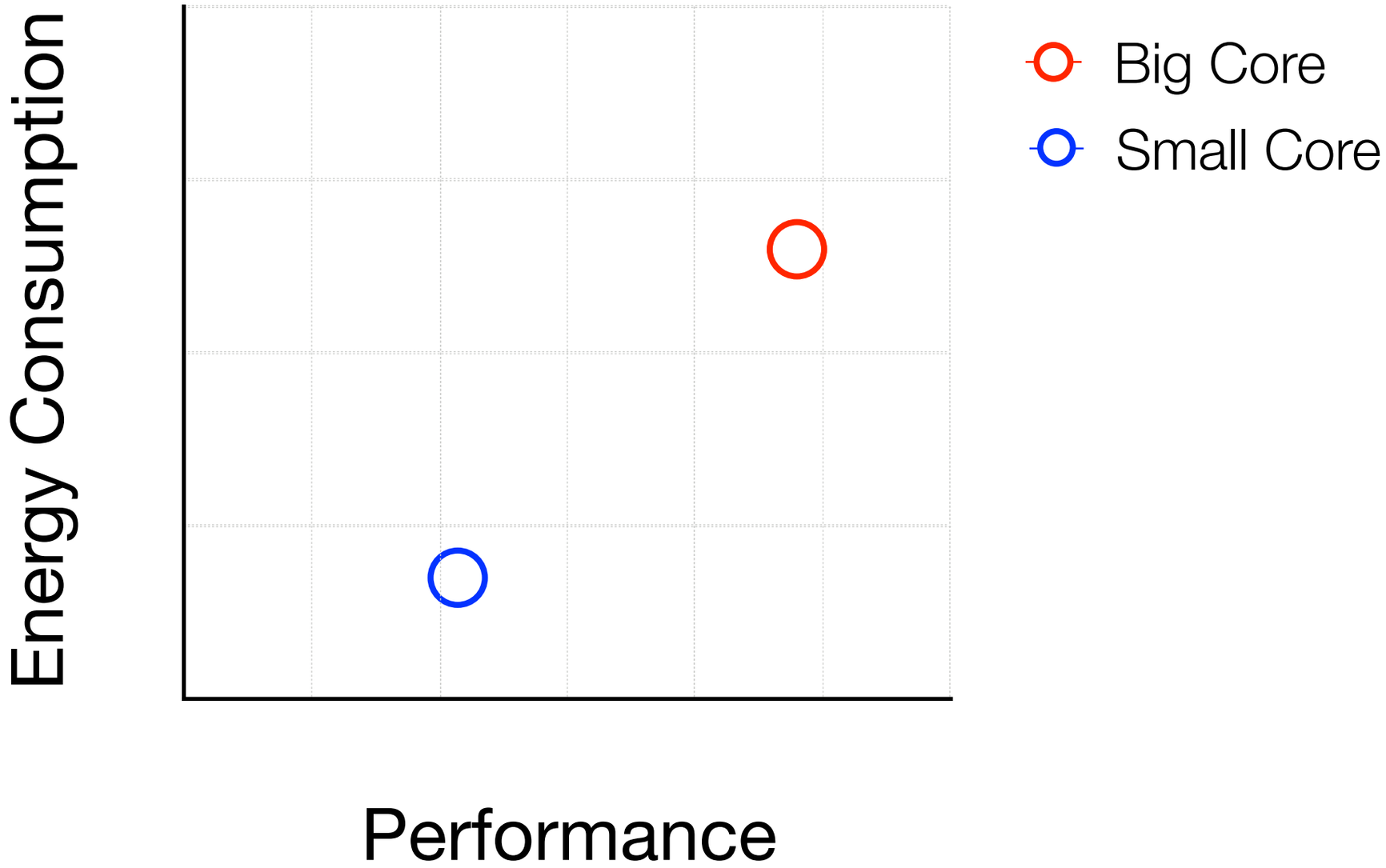
GreenWeb can support a range of energy saving techniques

- Dynamic resolution scaling [MobiCom 2015]
- Power-saving display colors [MobiSys 2012]
- Selective resource loading [NSDI 2015]
- ACMP-based hardware mechanism

Asymmetric Chip-multiprocessor

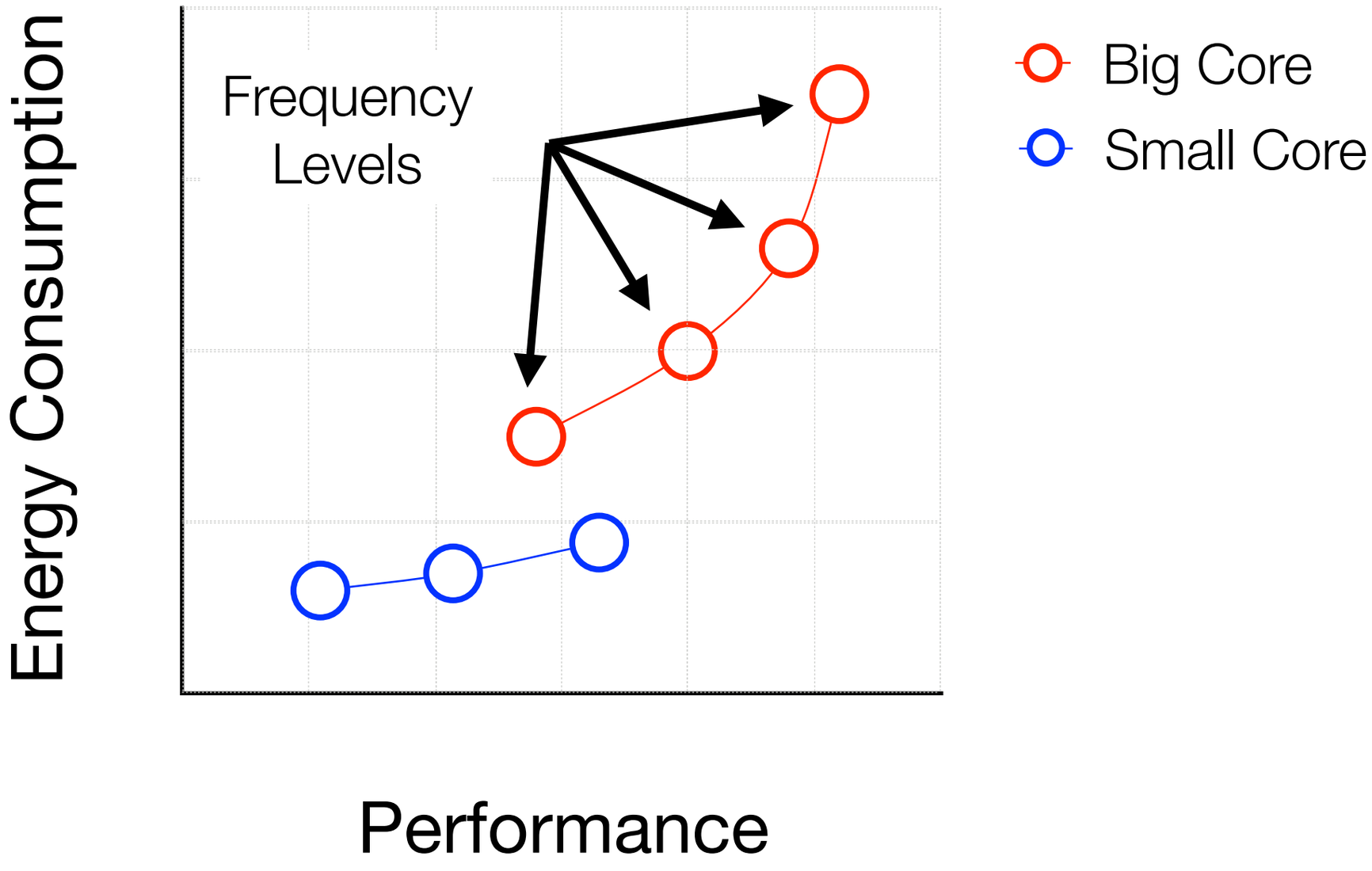
Asymmetric Chip-multiprocessor

- Offer a large performance-energy trade-off space



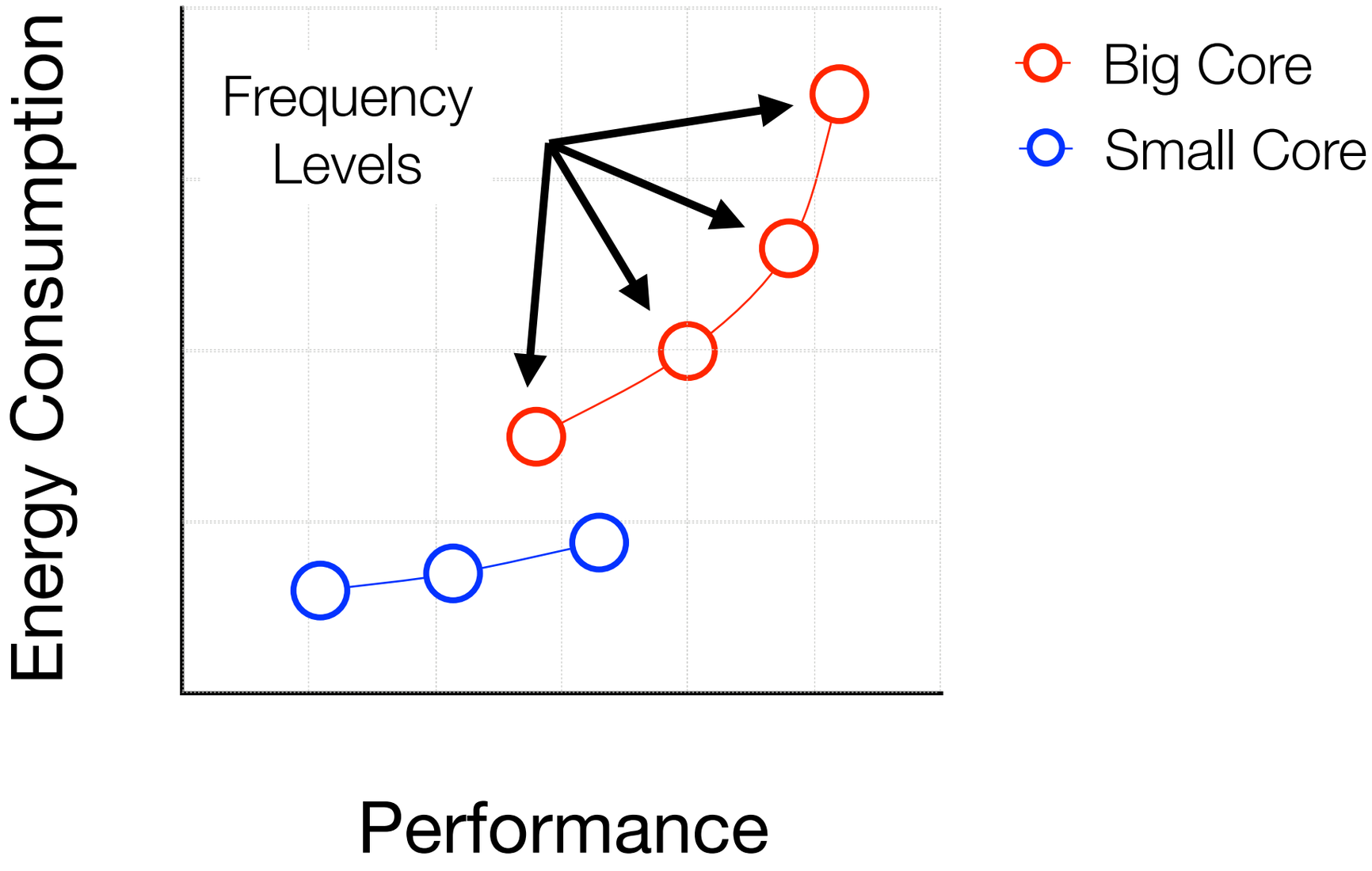
Asymmetric Chip-multiprocessor

- Offer a large performance-energy trade-off space



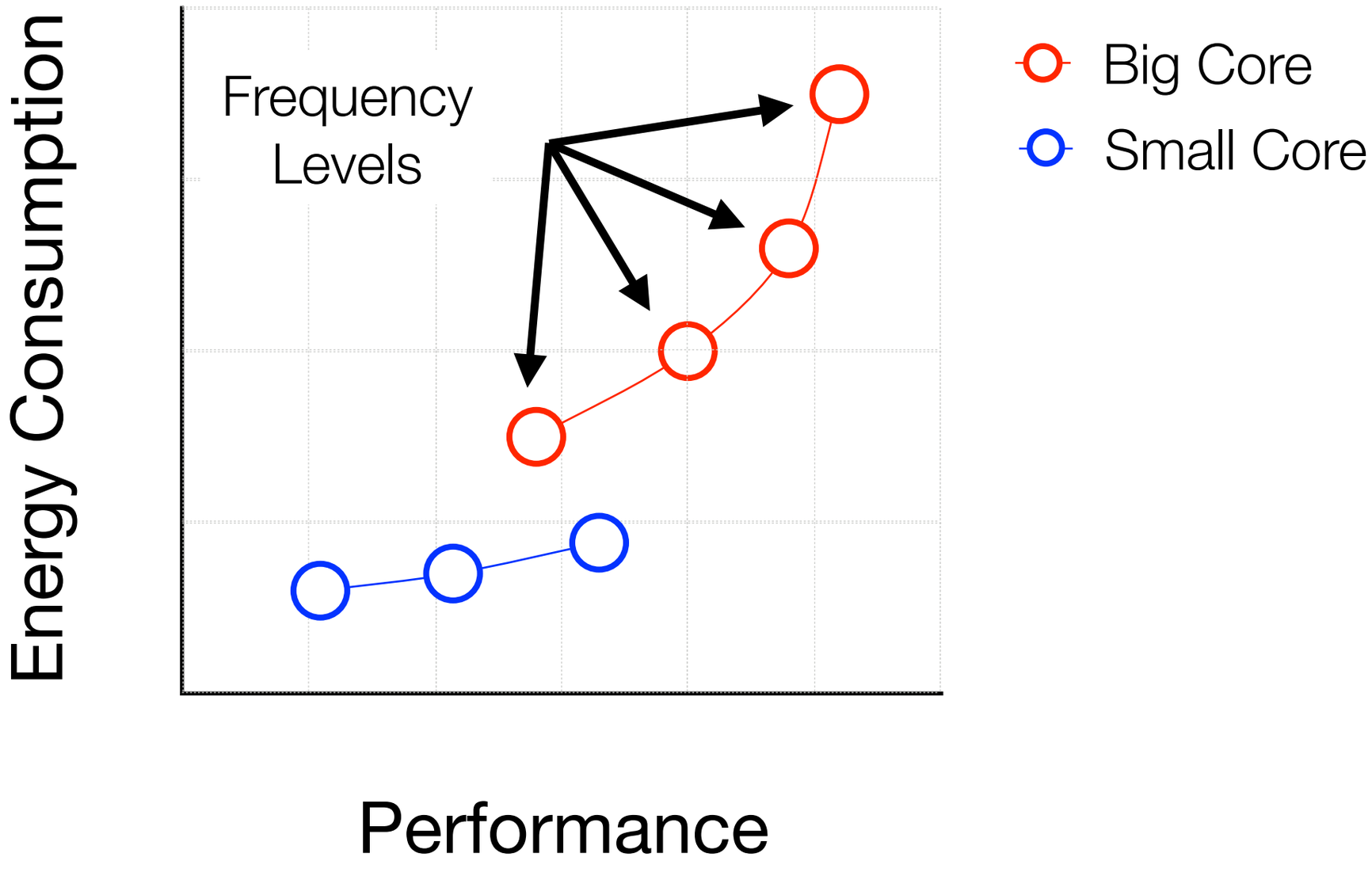
Asymmetric Chip-multiprocessor

- Offer a large performance-energy trade-off space
- Already widely used (e.g., Galaxy S6 & iPhone 7)



ACMP-based **GreenWeb** Runtime

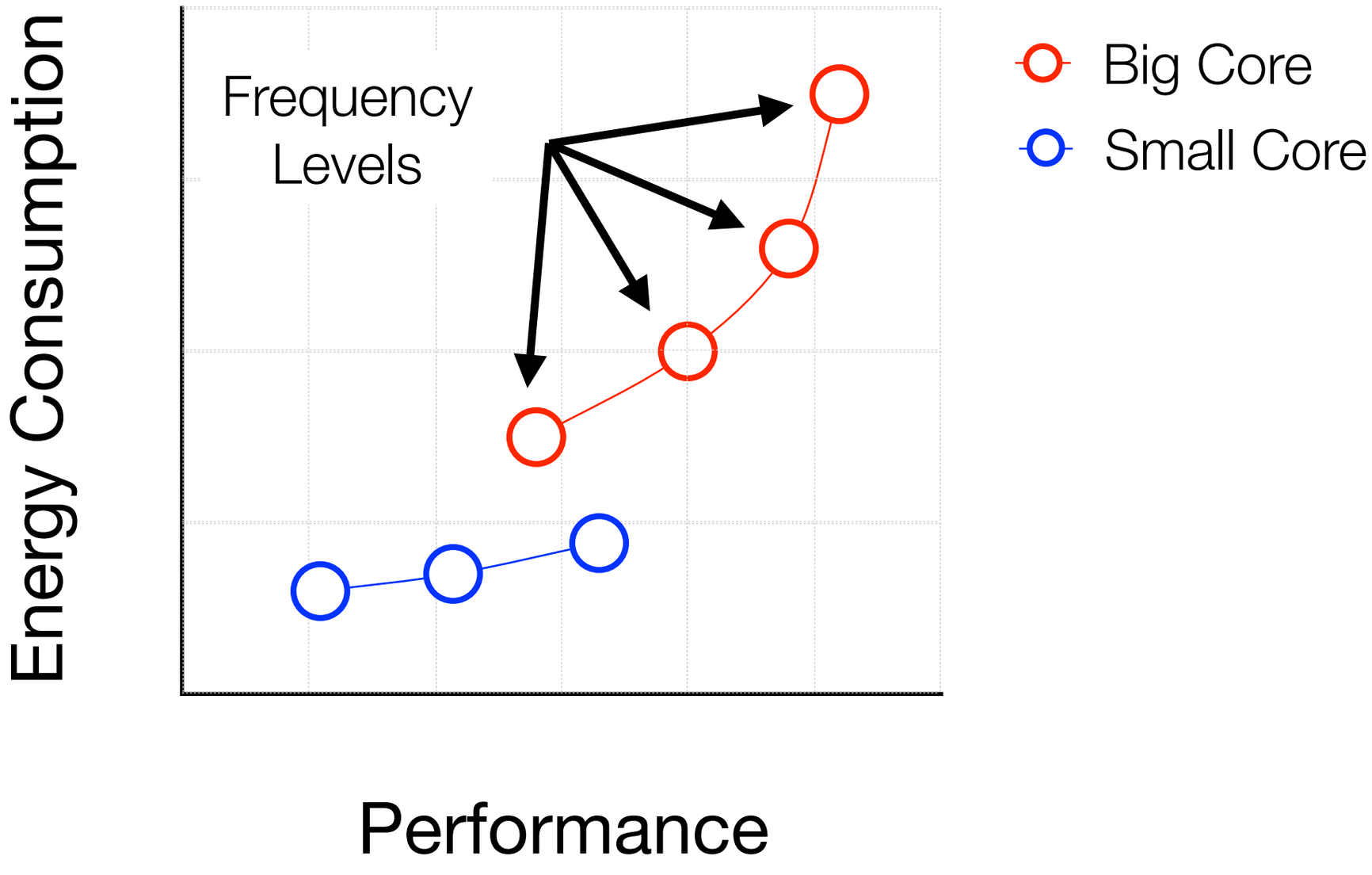
- Provide just enough energy to meet QoS constraints



ACMP-based **GreenWeb** Runtime

- Provide just enough energy to meet QoS constraints

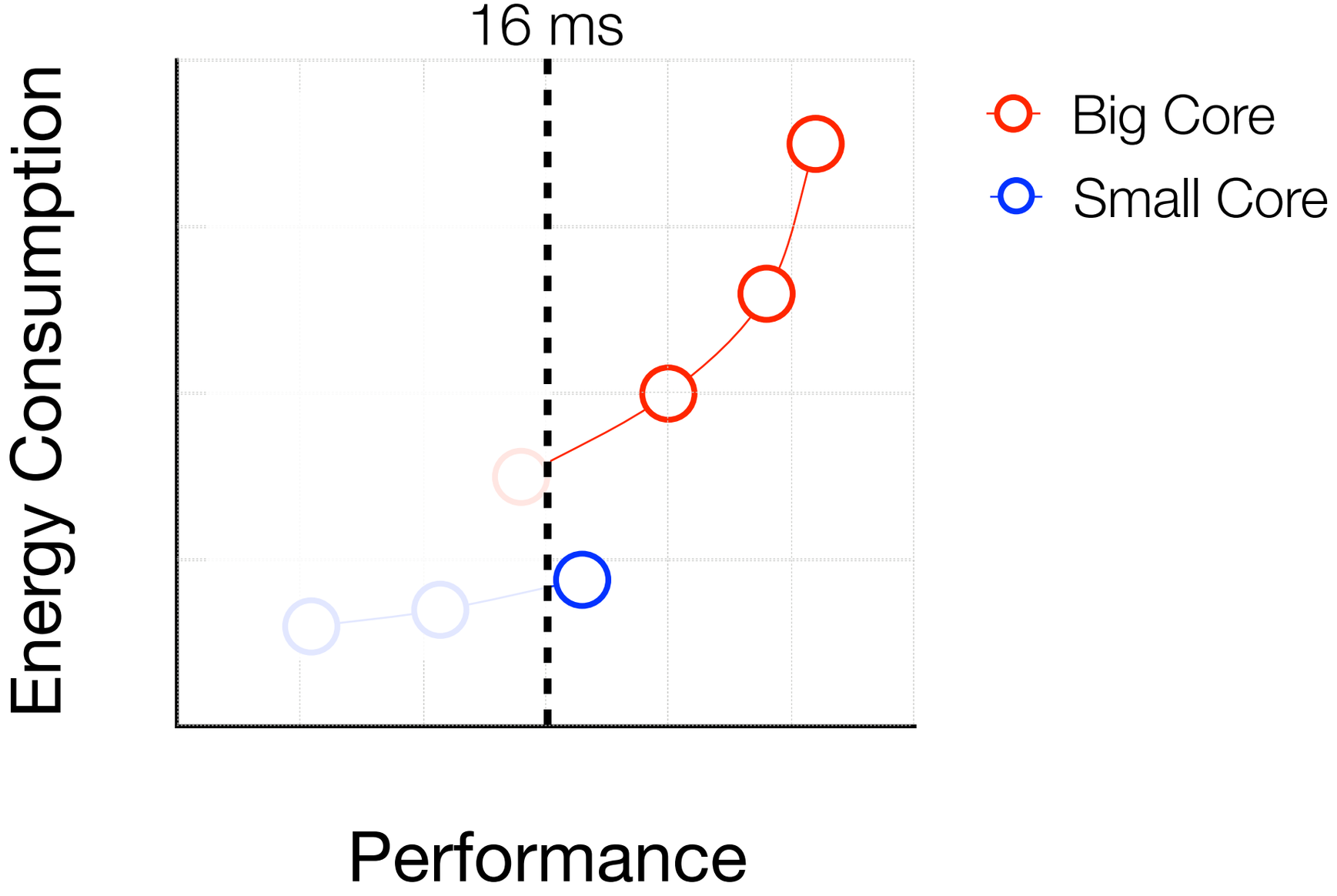
```
div {ontouchend: latency, 16 ms}
```



ACMP-based **GreenWeb** Runtime

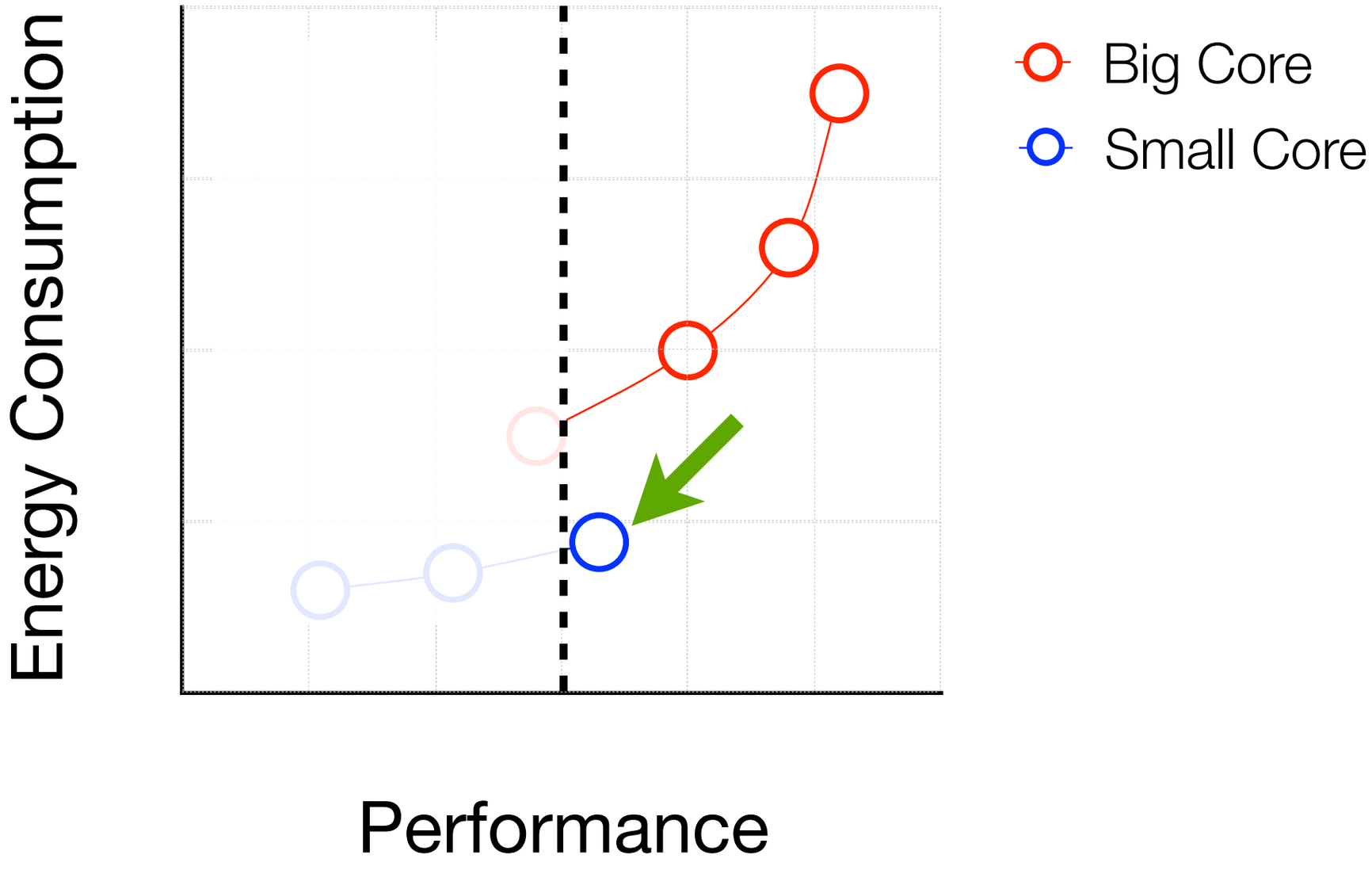
- Provide just enough energy to meet QoS constraints

```
div {ontouchend: latency, 16 ms}
```



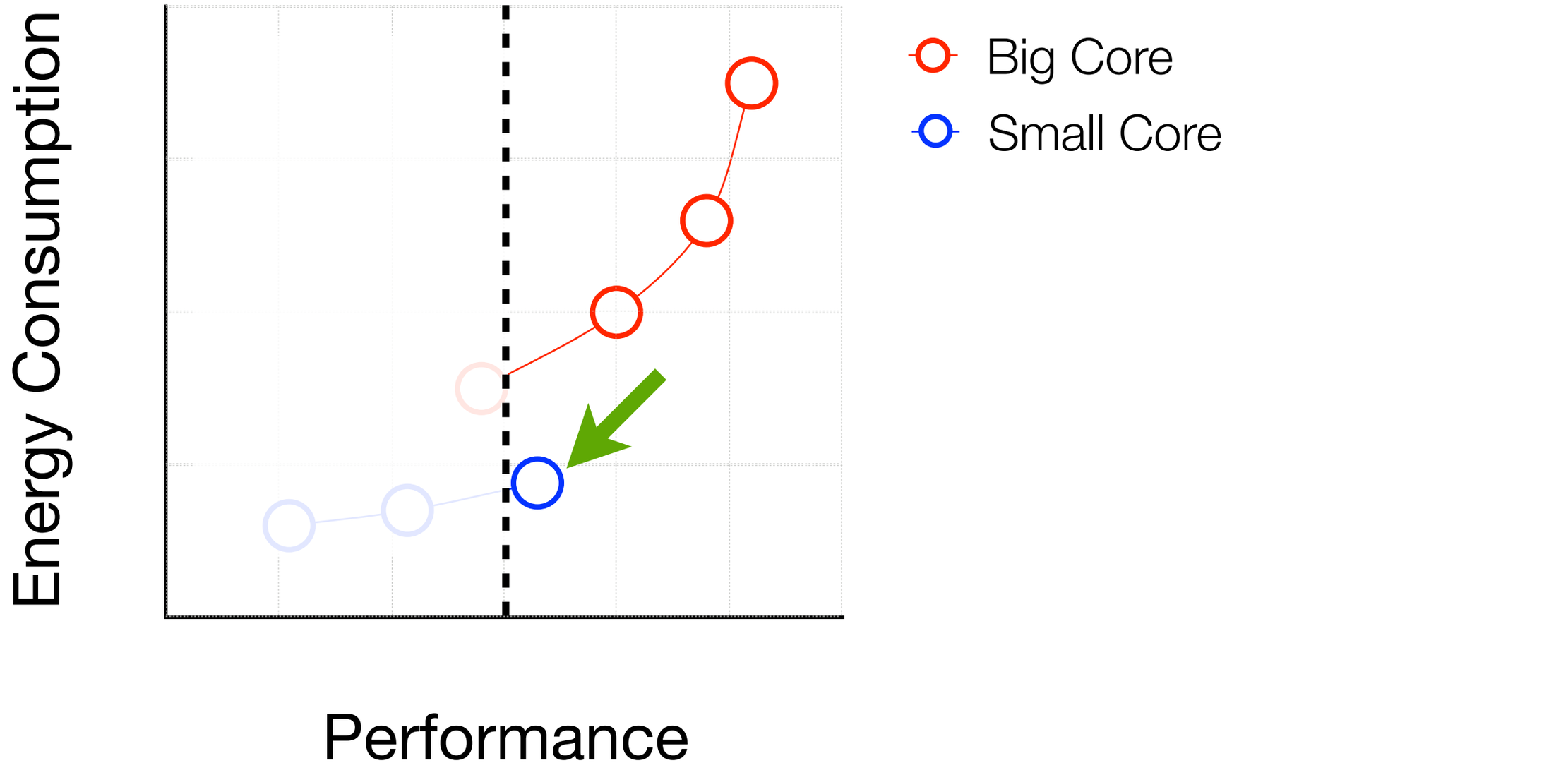
ACMP-based **GreenWeb** Runtime

- Provide just enough energy to meet QoS constraints



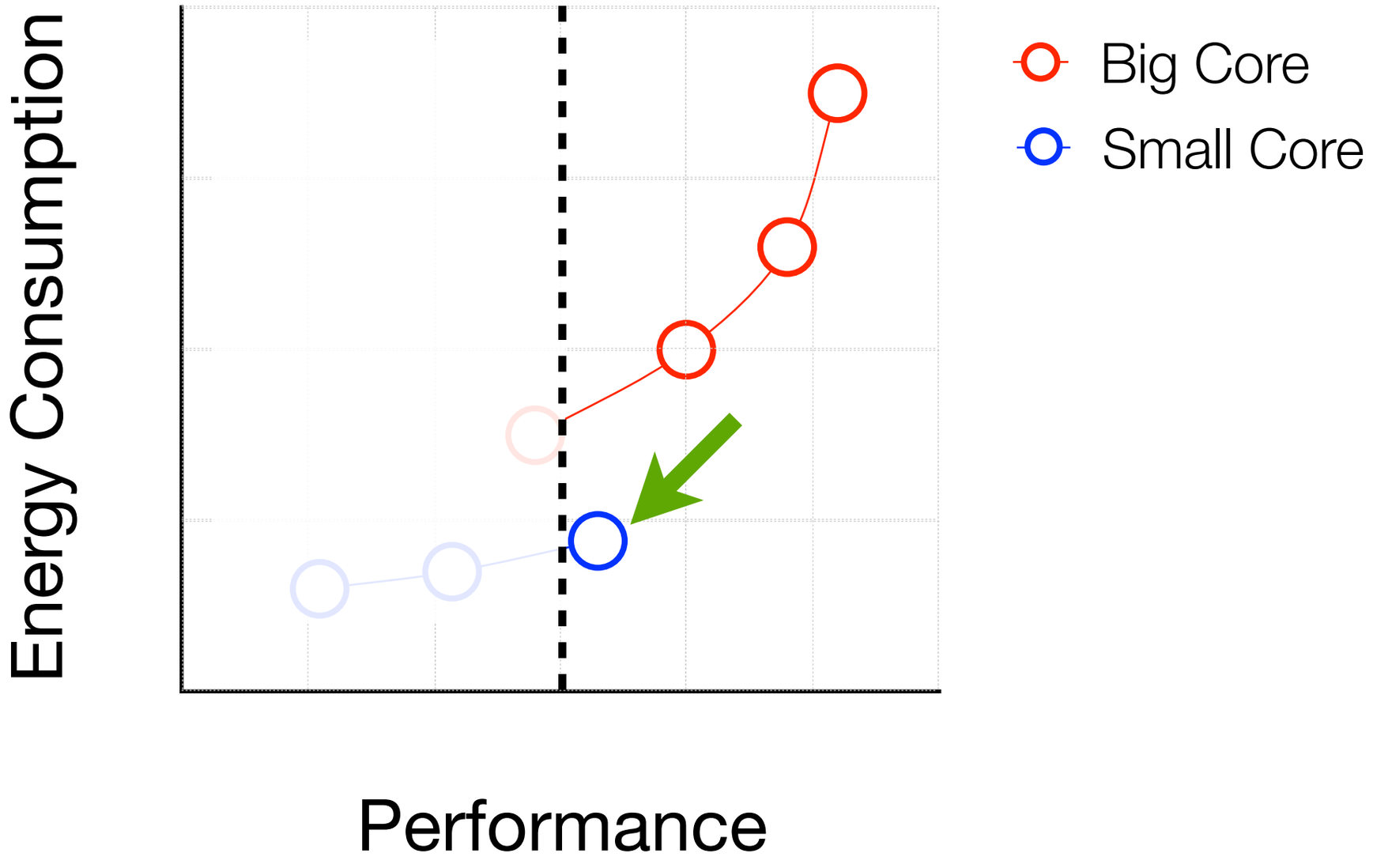
ACMP-based **GreenWeb** Runtime

- Provide just enough energy to meet QoS constraints
- Event-based Scheduling (EBS) [HPCA 2015, 2013]



ACMP-based **GreenWeb** Runtime

Predict the performance and energy of each configuration.



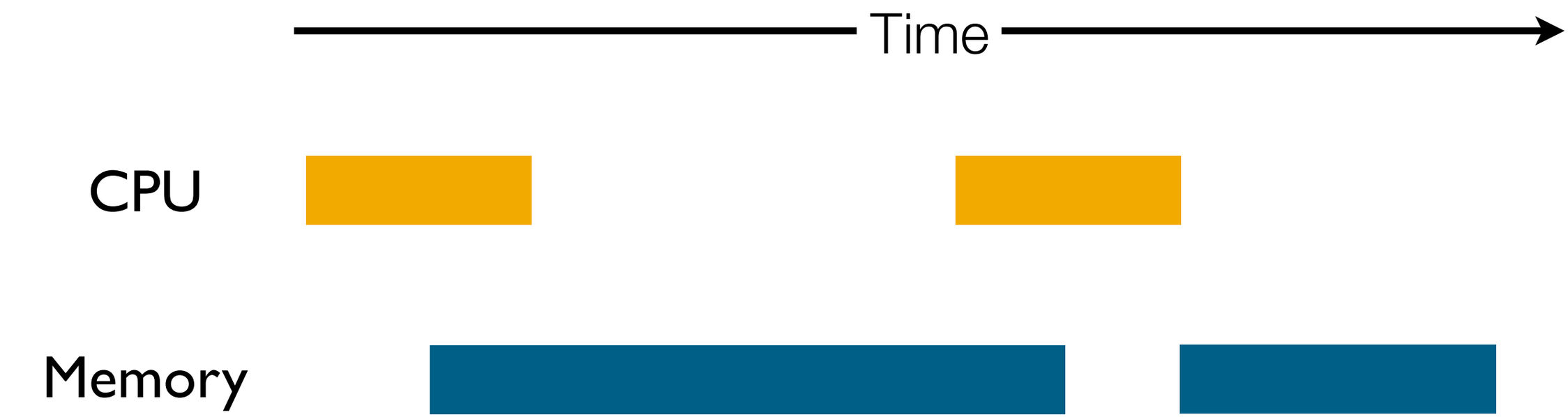
ACMP-based **GreenWeb** Runtime

CPU

Memory

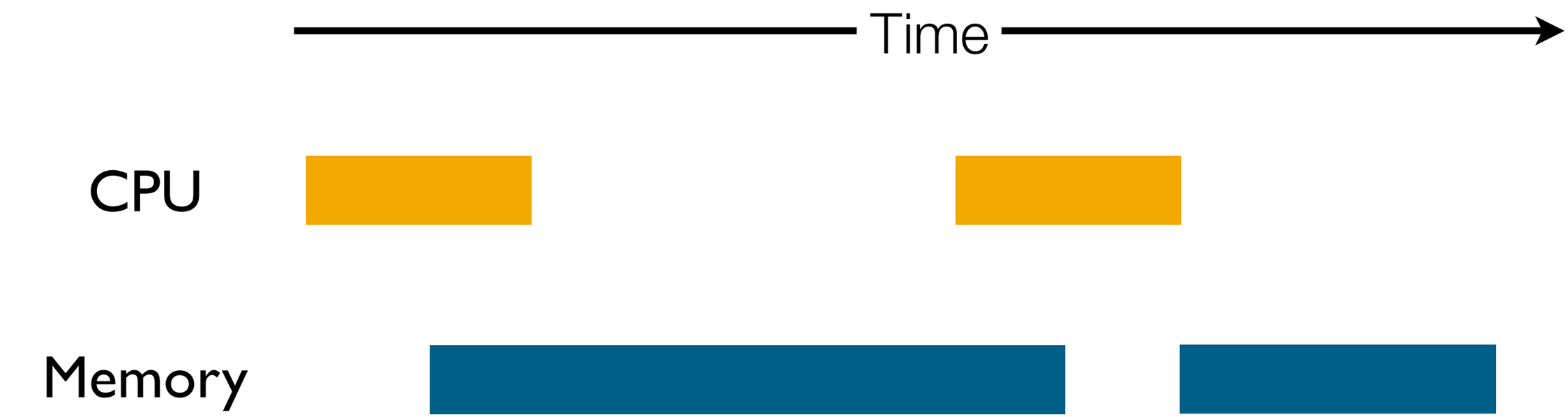
[PLDI 2003] Xie, et al., “Compile-Time Dynamic Voltage Scaling Settings: Opportunities and Limits”

ACMP-based **GreenWeb** Runtime



[PLDI 2003] Xie, et al., “Compile-Time Dynamic Voltage Scaling Settings: Opportunities and Limits”

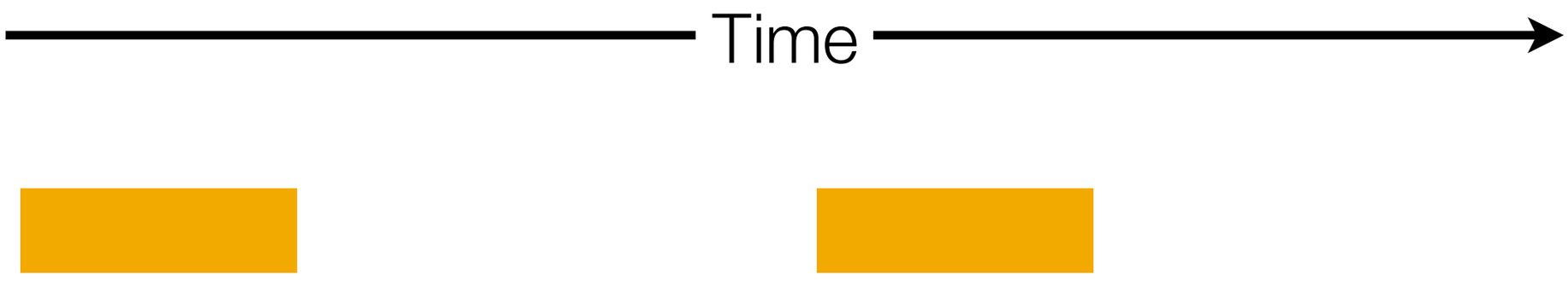
ACMP-based **GreenWeb** Runtime



Execution Time =

[PLDI 2003] Xie, et al., “Compile-Time Dynamic Voltage Scaling Settings: Opportunities and Limits”

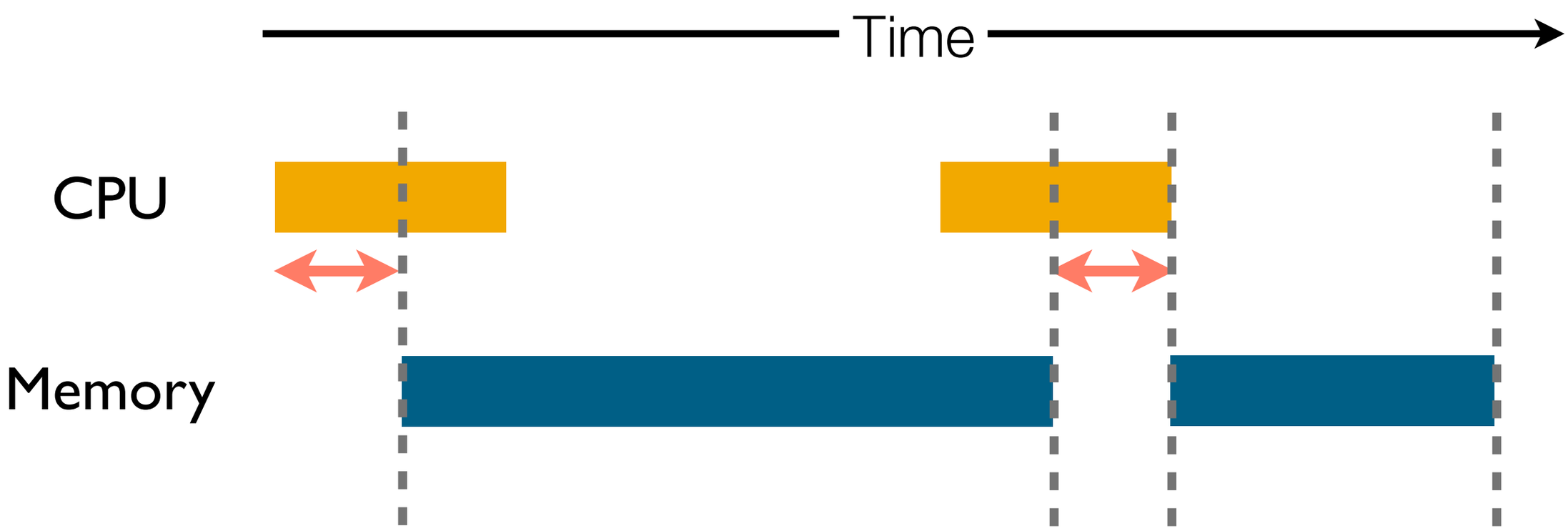
ACMP-based **GreenWeb** Runtime



$$\text{Execution Time} = T_{memory} + \dots$$

[PLDI 2003] Xie, et al., “Compile-Time Dynamic Voltage Scaling Settings: Opportunities and Limits”

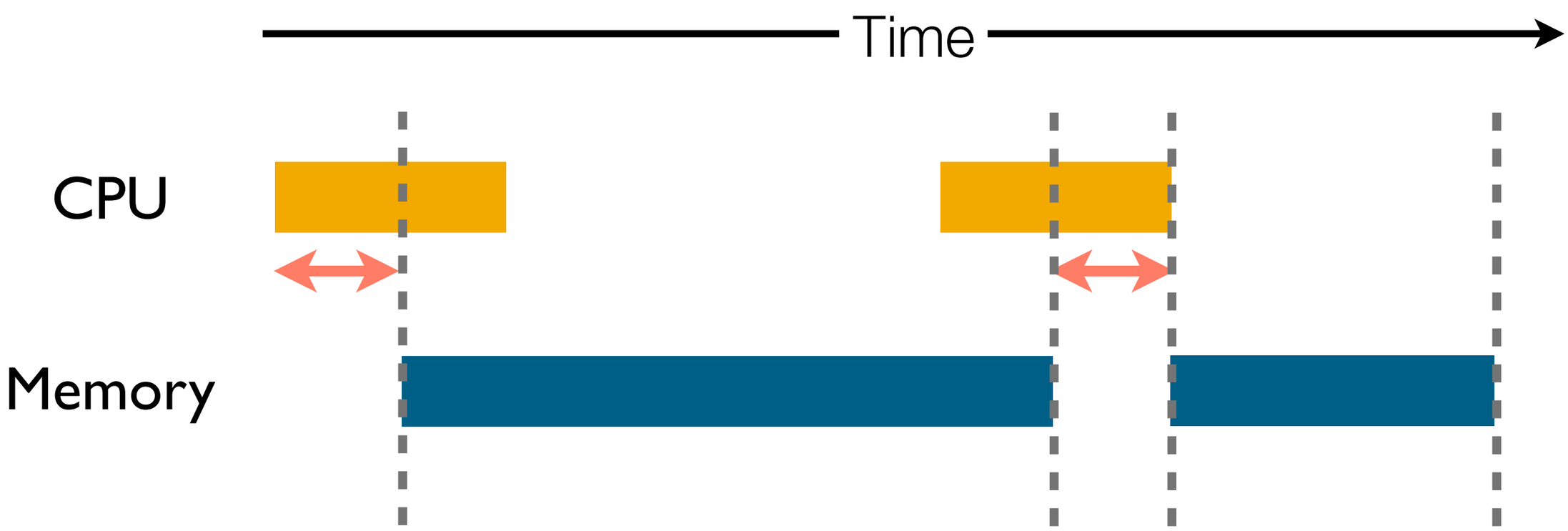
ACMP-based **GreenWeb** Runtime



$$\text{Execution Time} = T_{memory} + T_{non\text{-}overlap}$$

[PLDI 2003] Xie, et al., “Compile-Time Dynamic Voltage Scaling Settings: Opportunities and Limits”

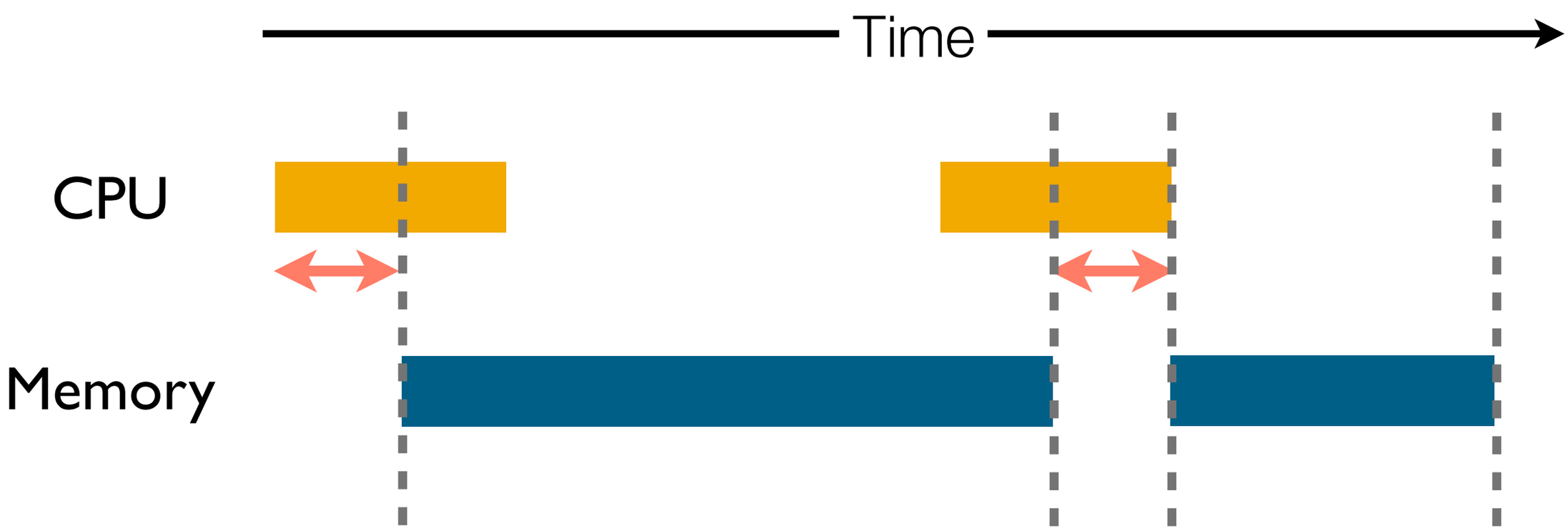
ACMP-based **GreenWeb** Runtime



$$\text{Execution Time} = T_{memory} + N_{cycles} / f$$

[PLDI 2003] Xie, et al., “Compile-Time Dynamic Voltage Scaling Settings: Opportunities and Limits”

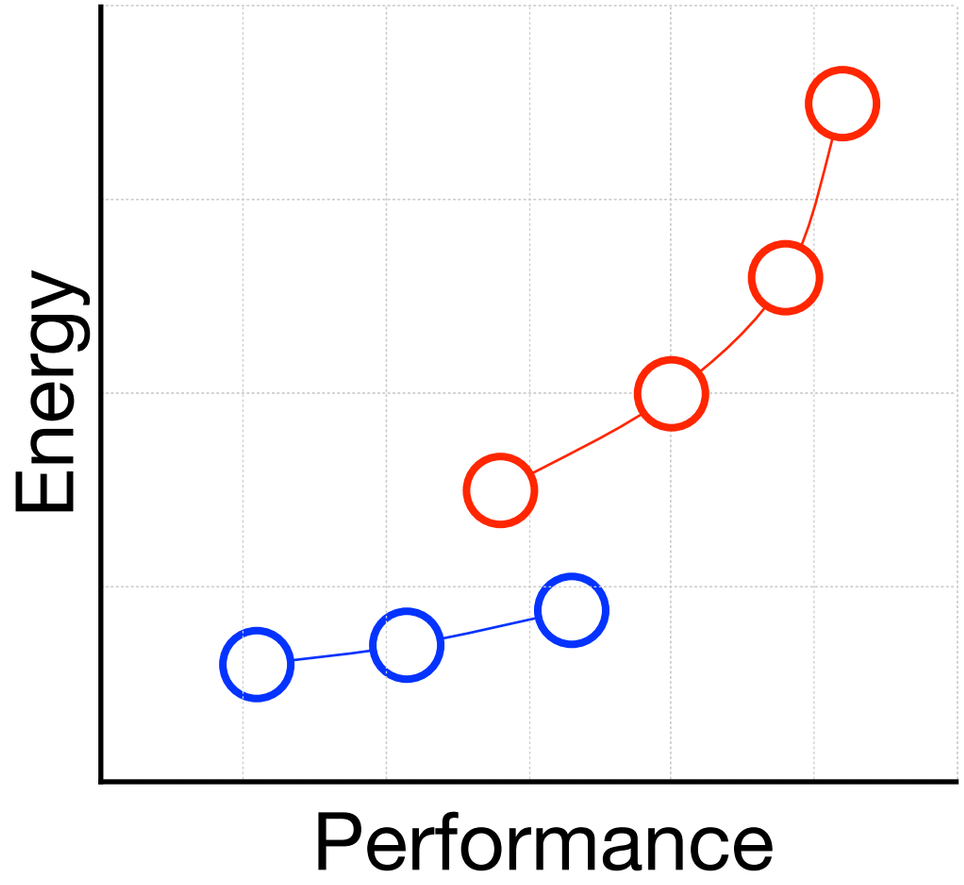
ACMP-based **GreenWeb** Runtime



$$\text{Execution Time} = T_{memory} + N_{cycles} / f$$

[PLDI 2003] Xie, et al., “Compile-Time Dynamic Voltage Scaling Settings: Opportunities and Limits”

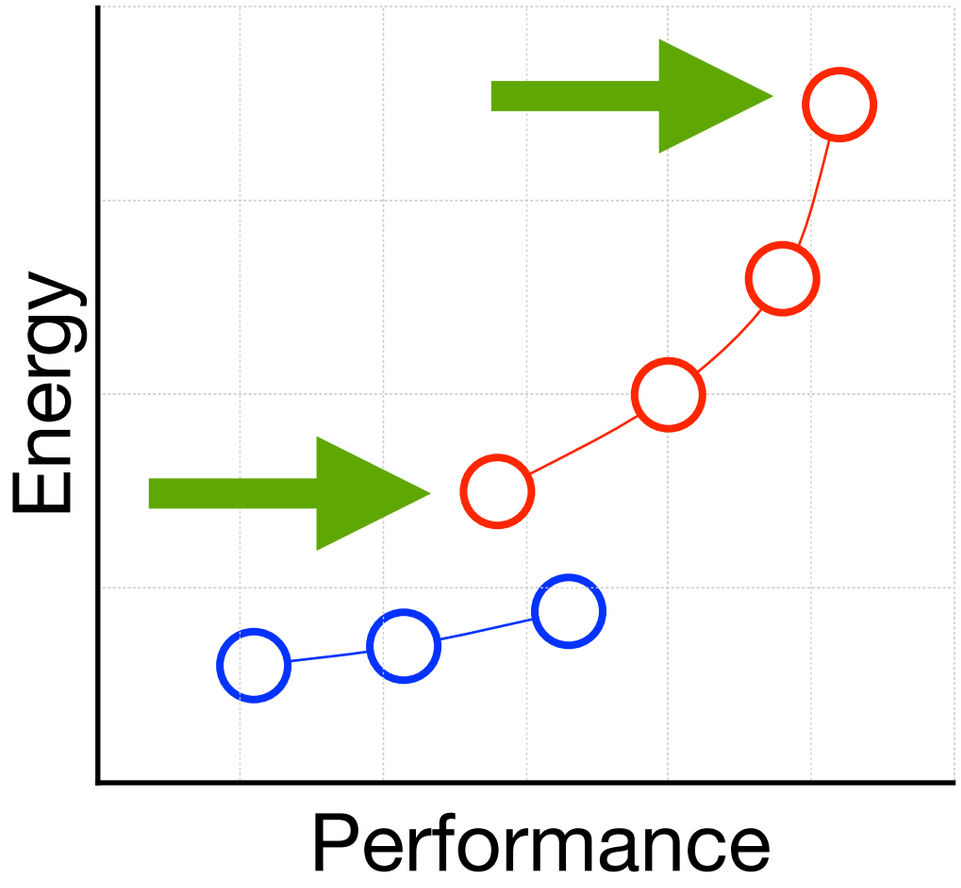
ACMP-based **GreenWeb** Runtime



$$\text{Execution Time} = T_{memory} + N_{cycles} / f$$

[PLDI 2003] Xie, et al., “Compile-Time Dynamic Voltage Scaling Settings: Opportunities and Limits”

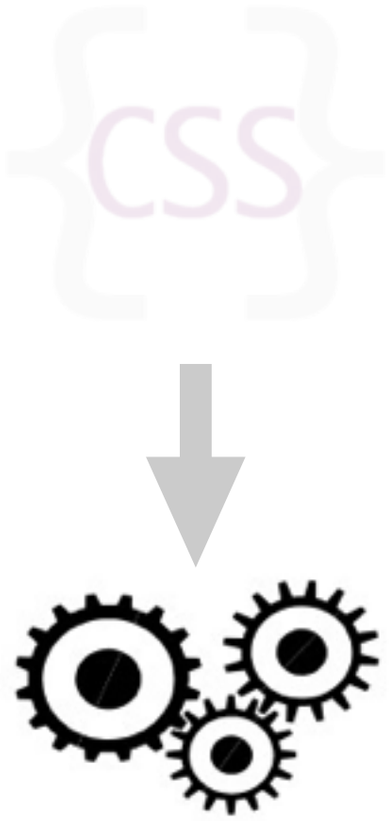
ACMP-based **GreenWeb** Runtime



$$\text{Execution Time} = T_{memory} + N_{cycles} / f$$

[PLDI 2003] Xie, et al., “Compile-Time Dynamic Voltage Scaling Settings: Opportunities and Limits”

GreenWeb: Language for Energy-Efficiency



Abstractions

Express QoS

Runtime

Satisfy QoS specifications while saving energy

Effect

60% energy savings on real system implementations

GreenWeb: Language for Energy-Efficiency

Abstractions

Express QoS

Runtime

Satisfy QoS specifications
while saving energy

Effect

60% energy savings on real system implementations

Real Hardware/Software Setup

ODroid XU+E development board,
which contains an Exynos 5410 SoC
used in Samsung Galaxy S4.

Real Hardware/Software Setup

ODroid XU+E development board, which contains an Exynos 5410 SoC used in Samsung Galaxy S4.

Implementation incorporated into Chromium running on Android.

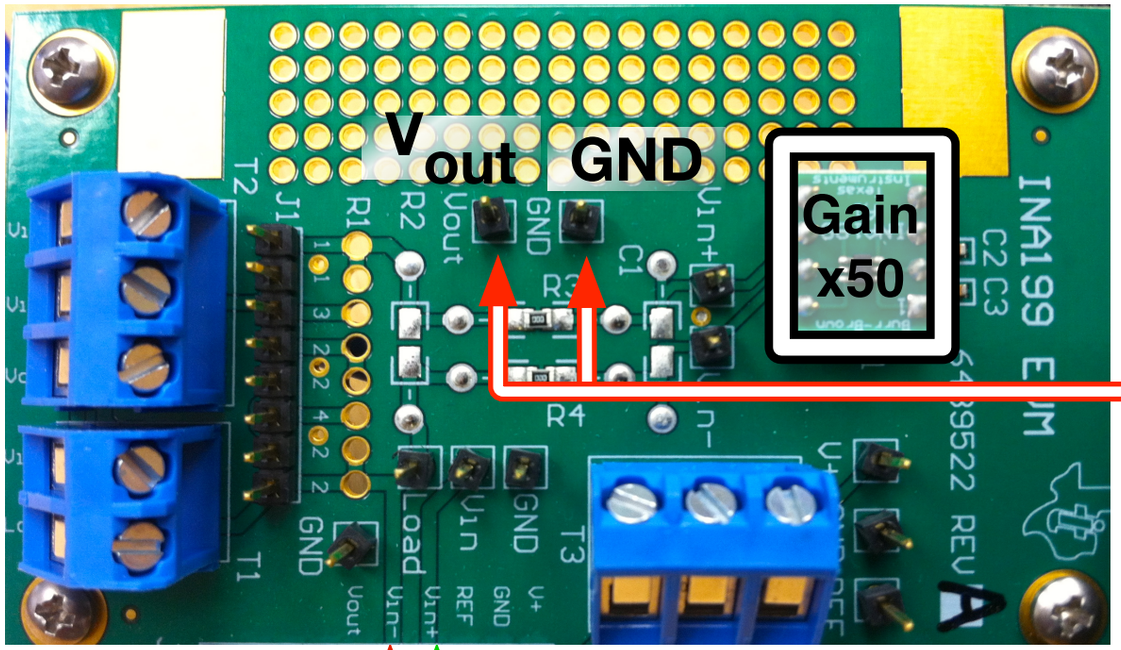
Real Hardware/Software Setup

ODroid XU+E development board, which contains an Exynos 5410 SoC used in Samsung Galaxy S4.

Implementation incorporated into Chromium running on Android.

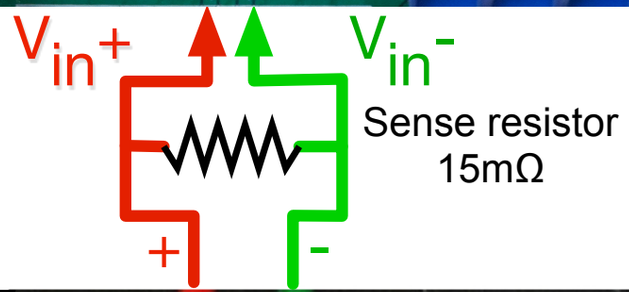
UI-level record and replay for reproducibility. [ISPASS'15]

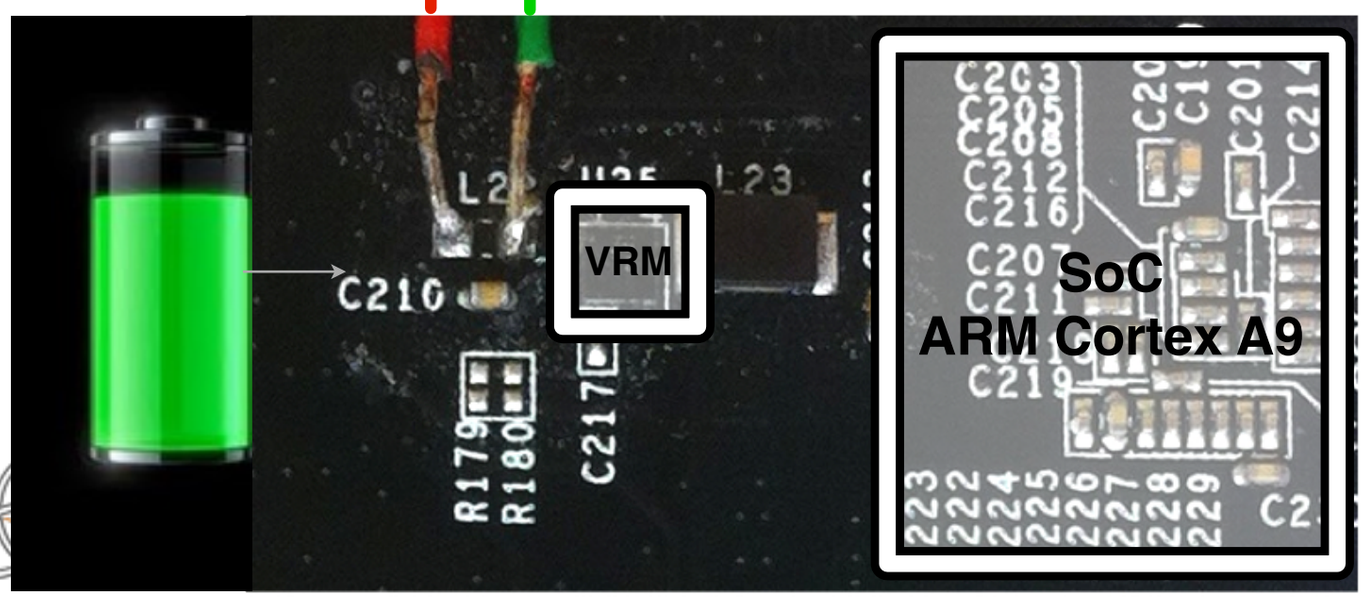
Power and Energy Measurements



Probe

Data Acquisition
(DAQ)





$$\text{Power} = (V_{in}^{+} - V_{in}^{-}) / R_{sense} * V_{in}^{-}$$

Evaluation

Baseline Mechanisms

Evaluation

Baseline Mechanisms

Highest performance: Standard to guarantee responsiveness

Evaluation

Baseline Mechanisms

Highest performance: Standard to guarantee responsiveness

Interactive CPU governor: Android default mechanism

Evaluation

Baseline Mechanisms

Highest performance: Standard to guarantee responsiveness

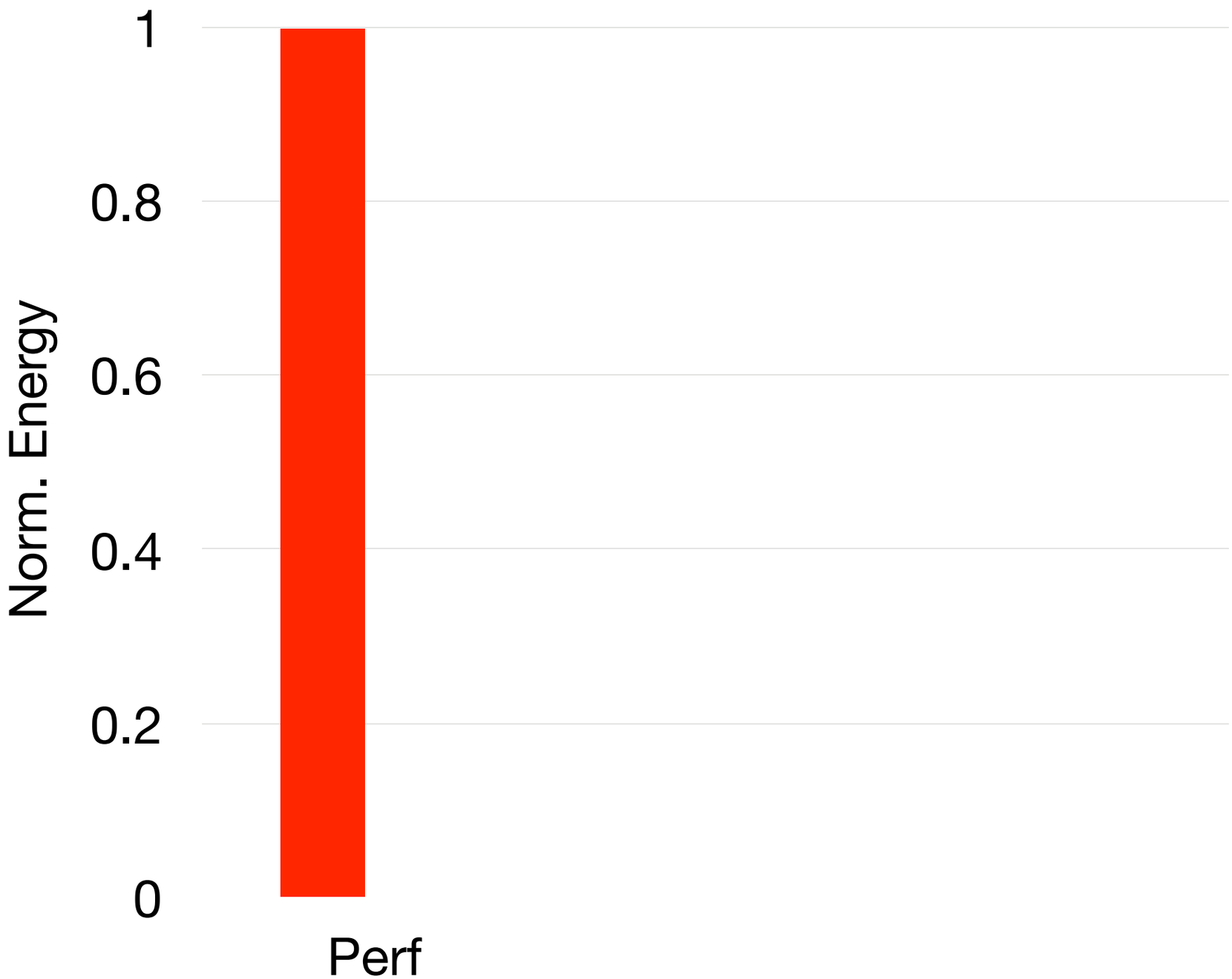
Interactive CPU governor: Android default mechanism

Evaluation Metrics

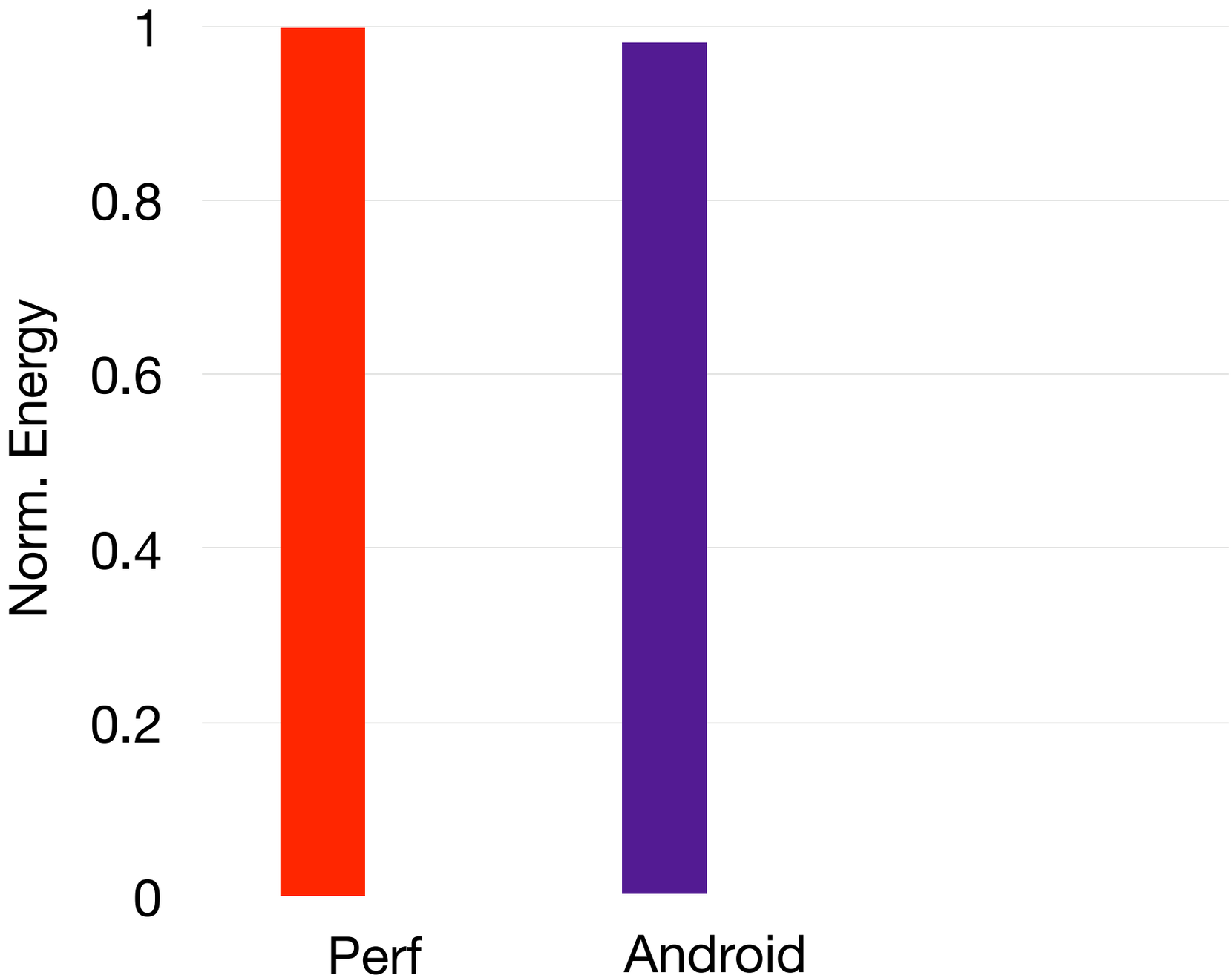
Energy Saving

QoS Violation: Percentage that a QoS target is violated

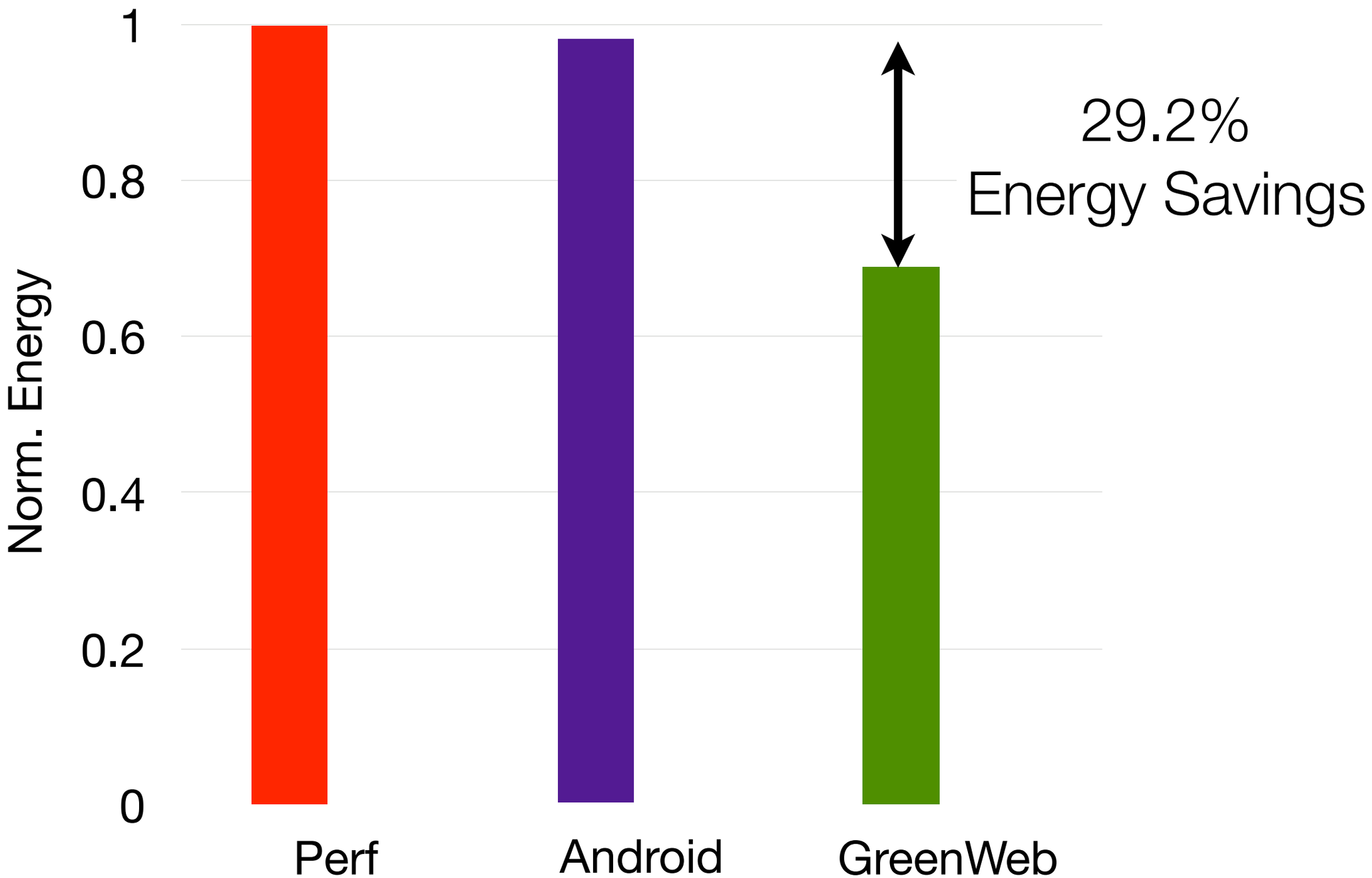
Evaluation Results



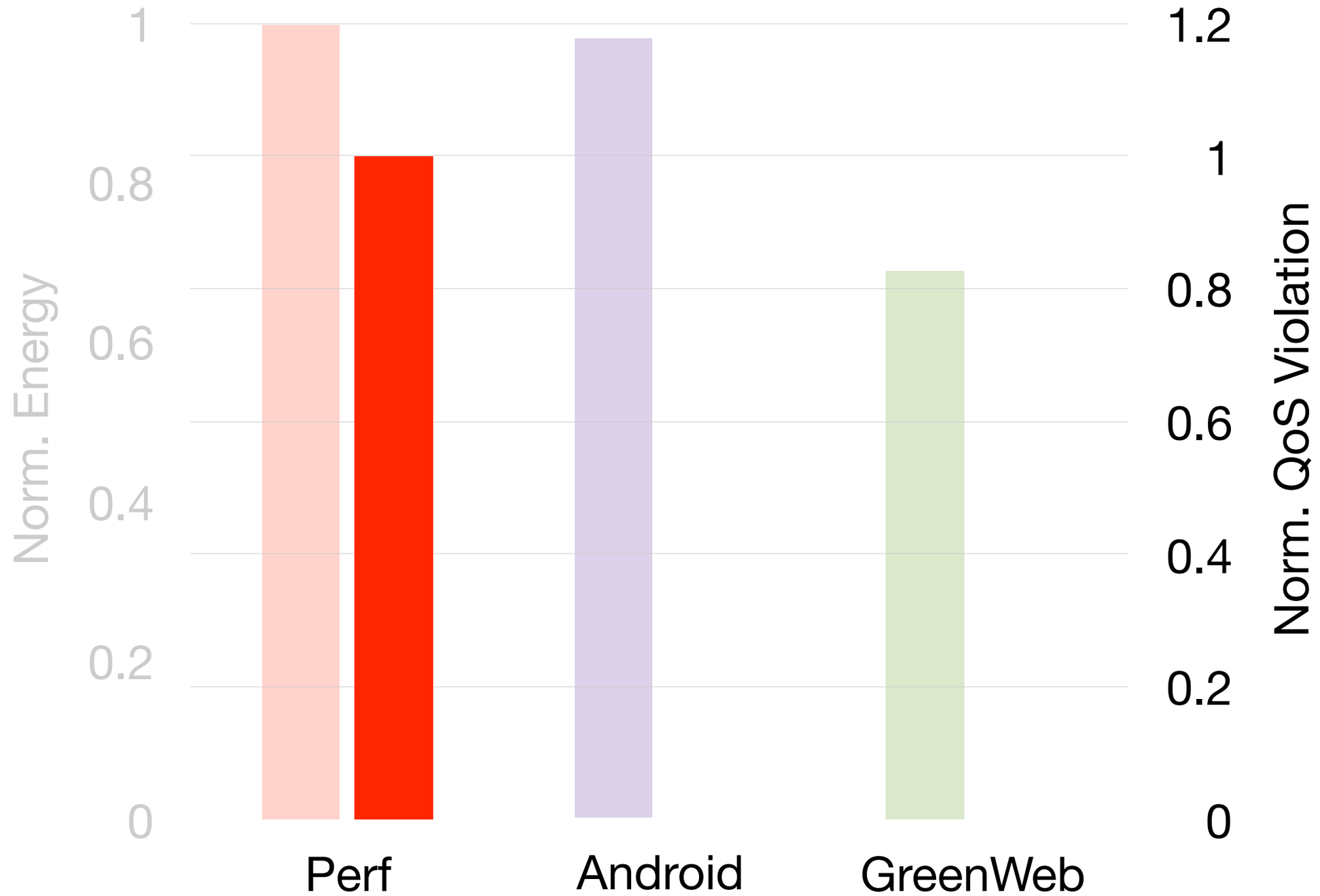
Evaluation Results



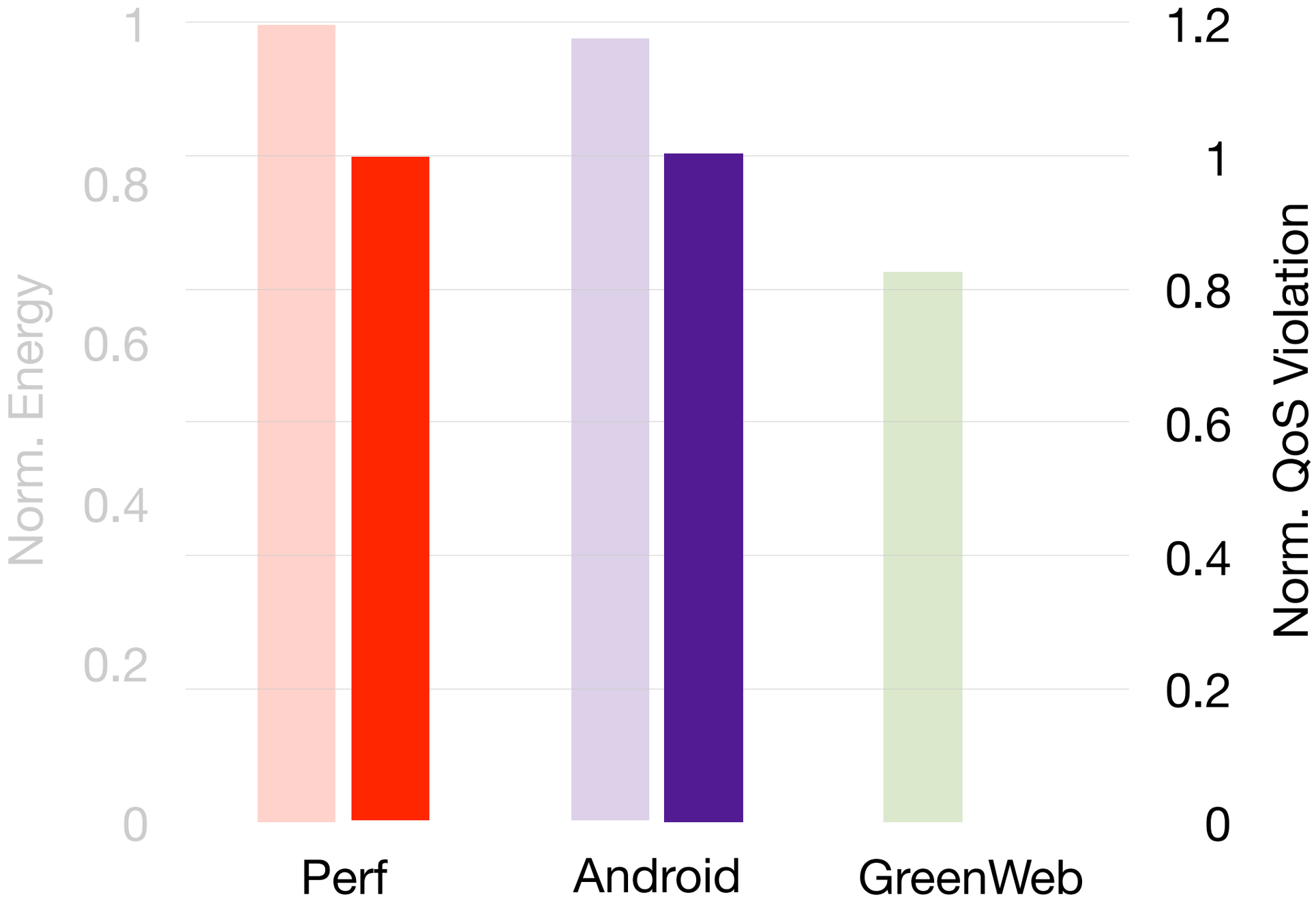
Evaluation Results



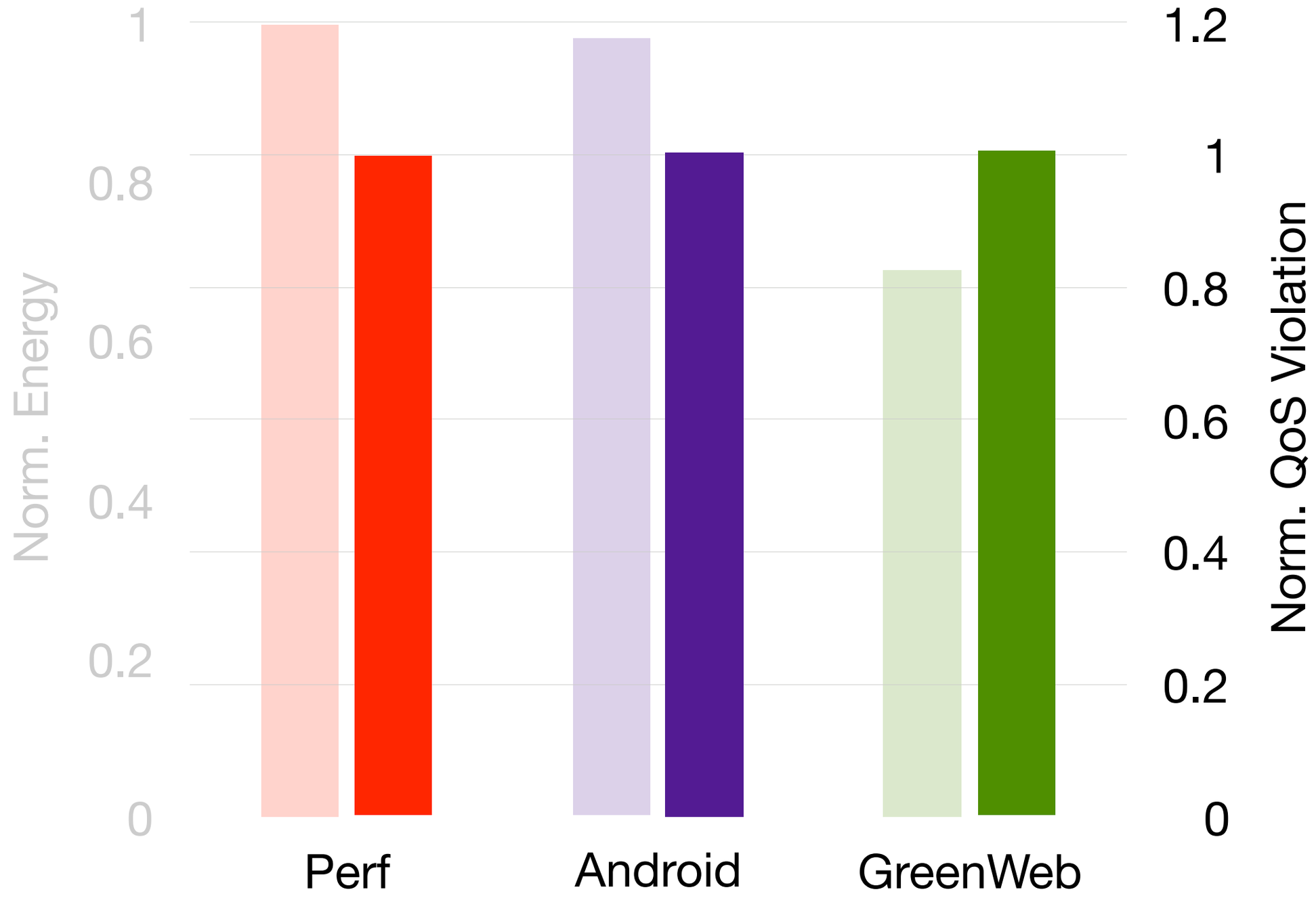
Evaluation Results



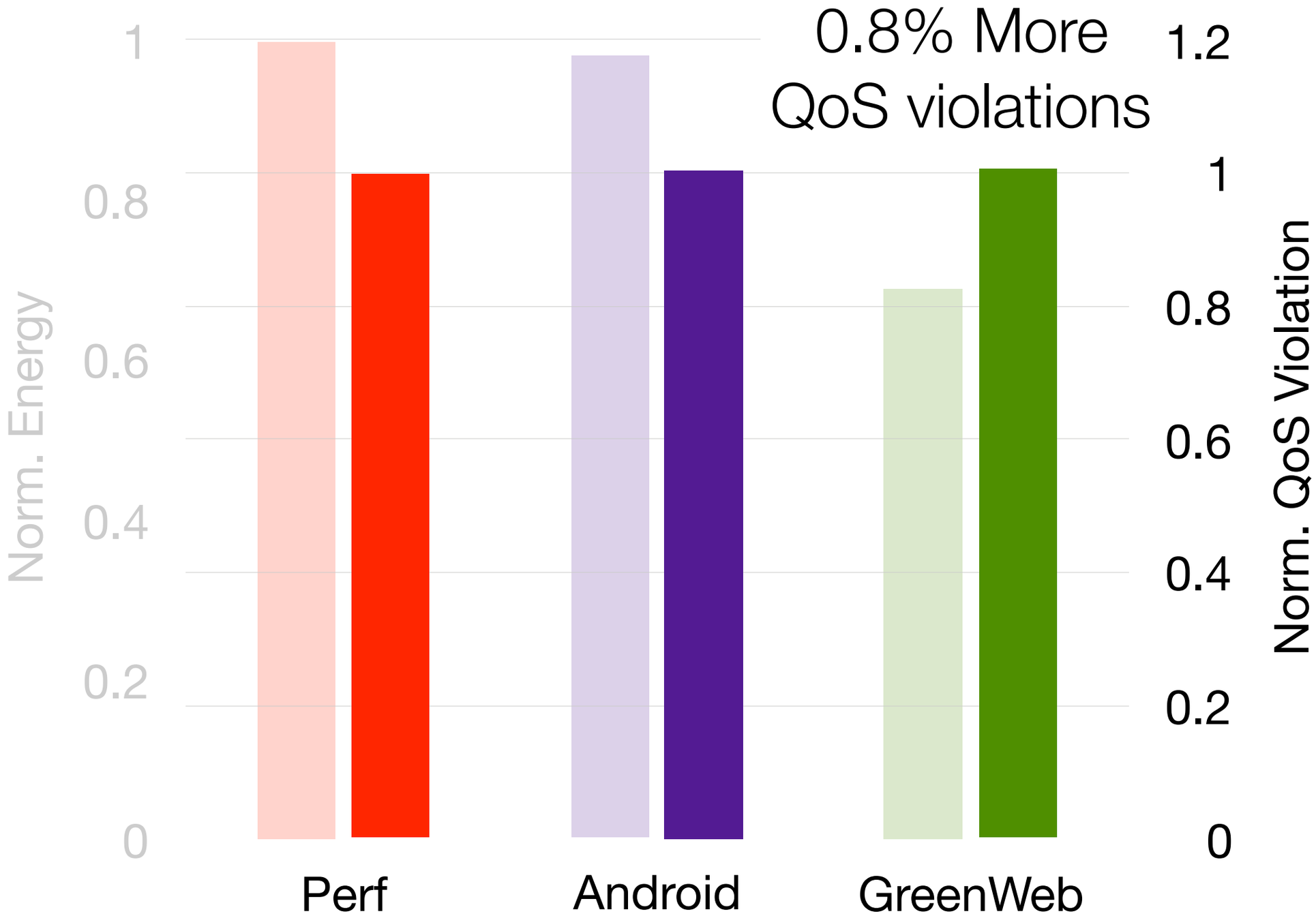
Evaluation Results



Evaluation Results



Evaluation Results



Summary: The Evolutionary Perspective

Summary: The Evolutionary Perspective

1990

HTML

Summary: The Evolutionary Perspective

1990

HTML

1996

JavaScript

Summary: The Evolutionary Perspective

1990
HTML

2008
Mobile Web

1996
JavaScript

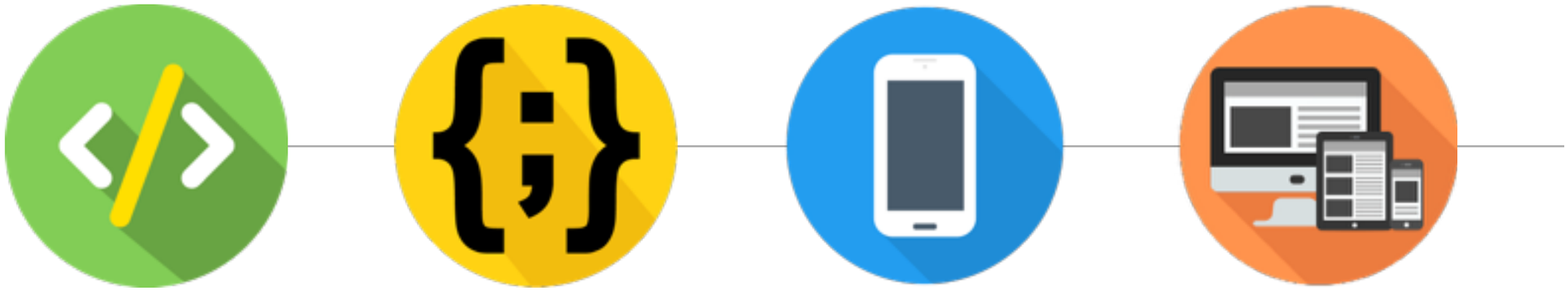
Summary: The Evolutionary Perspective

1990
HTML

1996
JavaScript

2008
Mobile Web

2012
Responsive Web



Summary: The Evolutionary Perspective

1990
HTML

1996
JavaScript

2008
Mobile Web

2012
Responsive Web

2016
Watt Wise Web

My Research Scope

My Research Scope

Mobile Web
Energy
Efficiency

[**PLDI** 2016] [**ISCA** 2014]
[**HPCA** 2016, 2015, 2013]
[**TOCS** 2017] [**CAL** 2014]

My Research Scope

Mobile Web Energy Efficiency

[**PLDI** 2016] [**ISCA** 2014]
[**HPCA** 2016, 2015, 2013]
[**TOCS** 2017] [**CAL** 2014]

Architectural Support for Scripting the Cloud

[**Submitted**]
[**MICRO** 2015]

My Research Scope

Mobile Web Energy Efficiency

[**PLDI** 2016] [**ISCA** 2014]
[**HPCA** 2016, 2015, 2013]
[**TOCS** 2017] [**CAL** 2014]

Network Impact on Mobile Web

[**CACM** 2017]
[**ACM Queue** 2016]
[**IEEE Micro** 2015]

Architectural Support for Scripting the Cloud

[**Submitted**]
[**MICRO** 2015]


SAMSUNG
12:45
Life companion
12:45

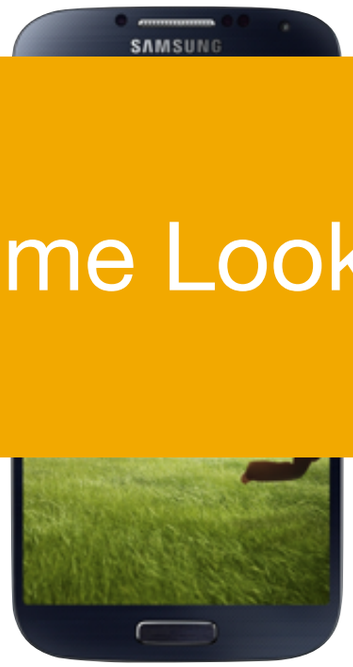
The Internet of Things

Web

The ~~Internet~~ of Things

Web

The ~~Internet~~ of Things



How Should Future Runtime Look Like for Web of Things?

Web

The ~~Internet~~ of Things

Say goodbye to the *monolithic*, “*one-size-fits-all*” Web runtime

Web

The ~~Internet~~ of Things

Embrace the
extensible, lightweight
future Web runtime

Runtime

From Monolithic Runtime
to Extensible Runtime

Runtime Architecture

From Monolithic Runtime
to Extensible Runtime

Runtime

From Monolithic Runtime
to Extensible Runtime

Architecture

Hardware specialization
with *safety* in mind

Expert Opinion

IEEE Micro, Jan/Feb, 2017

Cognitive Computing Safety: The New Horizon for Reliability

YUHAO ZHU
VIJAY JANAPA REDDI
University of Texas at Austin

Architecture

Hardware specialization with *safety* in mind

Application

Runtime

From Monolithic Runtime
to Extensible Runtime

Architecture

Hardware specialization
with *safety* in mind

Application

Language abstractions for
richer WoT semantics

Runtime

From Monolithic Runtime
to Extensible Runtime

Architecture

Hardware specialization
with *safety* in mind

Vijay Janapa Reddi

Scott Mahlke

Derek Chiou

Lizy John

Christine Julien

Mauricio Breternitz

Matt Lease

Thank you

Matt Welsh

Nat Duca

Matthew Halpern

Jingwen Leng

Aditya Srikanth

Wenzhi Cui

Daniel Richins

Energy-Efficient Mobile Web Computing

Yuhao Zhu

Electrical and Computer Engineering Department
The University of Texas at Austin

http://yuhaozhu.com
@yzhu88

THE UNIVERSITY OF TEXAS AT AUSTIN