WebCore:

Architectural Support for Mobile Web Browsing

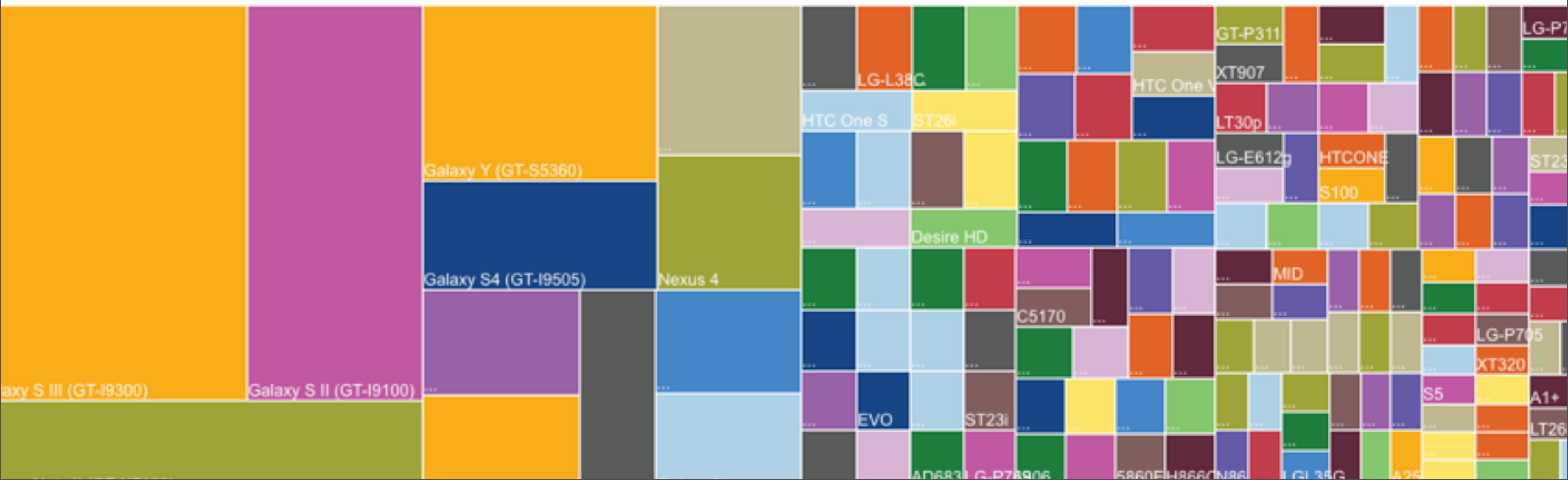
Yuhao Zhu, Vijay Janapa Reddi
Department of Electrical and Computer Engineering
The University of Texas at Austin

ISCA MainTalk — June 18th, 2014

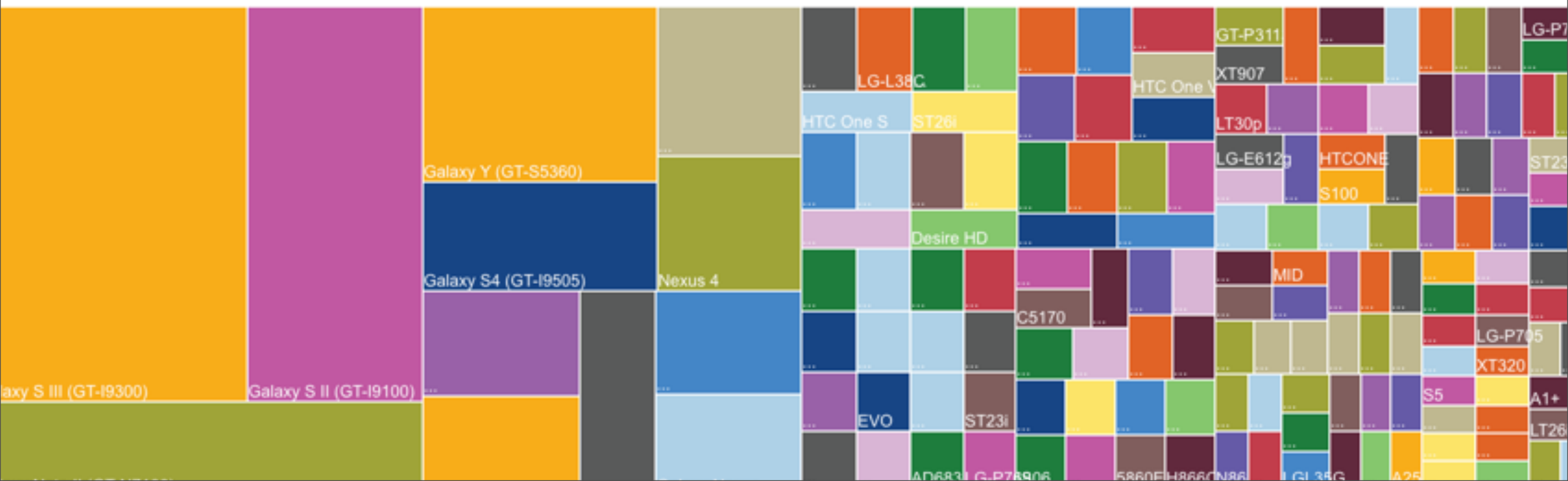
http2

HTML

CSS


Galaxy Y (GT-S5360)
Galaxy S4 (GT-I9505)
Nexus 4
LG-L38C
HTC One S
ST26i
Desire HD
GT-P311
XT907
LT30p
LG-E612g
HTCONE
S100
MID
C5170
EVO
ST23i
S5
XT320
A1+


iOS

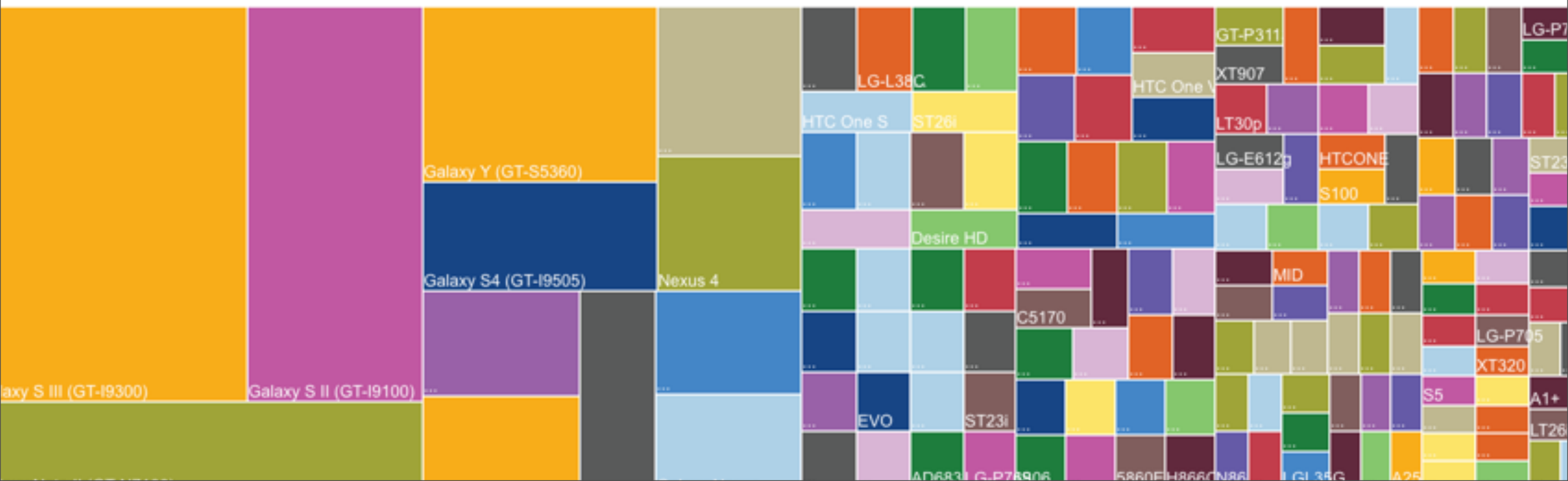

Galaxy Y (GT-S5360)
Galaxy S4 (GT-I9505)
Nexus 4
Galaxy S II (GT-I9100)
LG-L38C
HTC One S
ST26i
GT-P311
XT907
LT30p
LG-E612g
HTCONE
S100
Desire HD
MID
C5170
EVO
ST23i
XT320
S5
A1+

Swift

Java

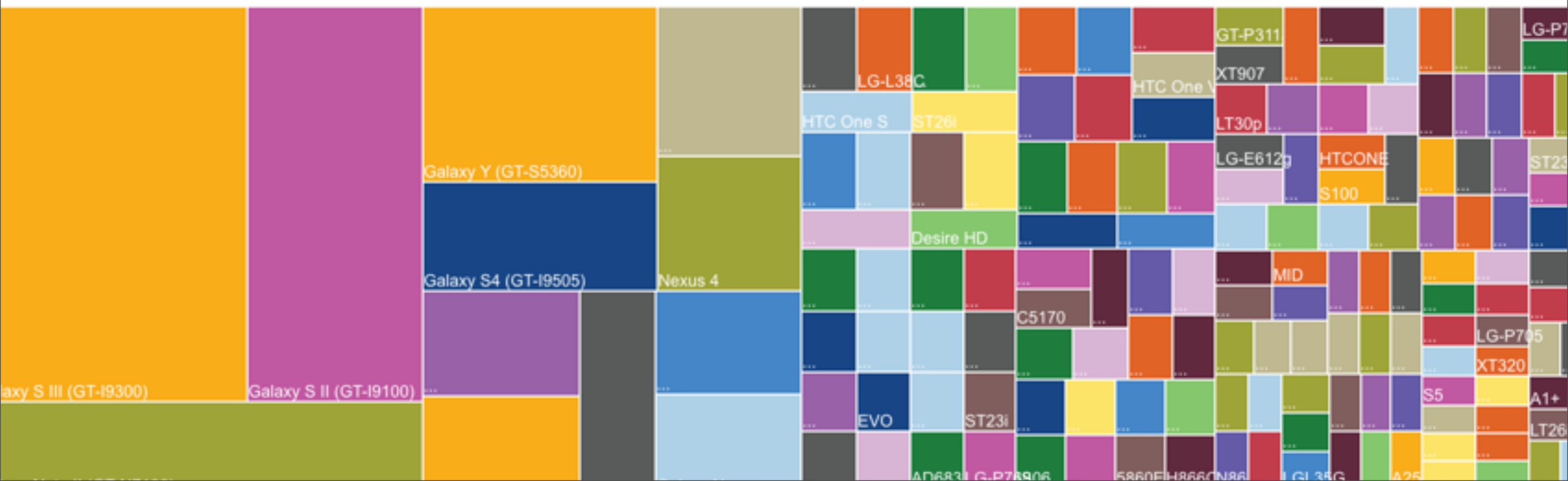
TM

iOS

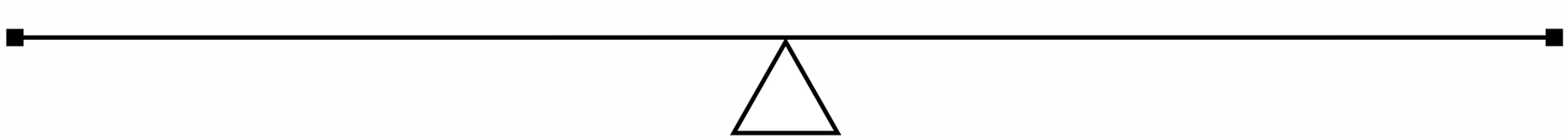



HTML
CSS
JS


iOS

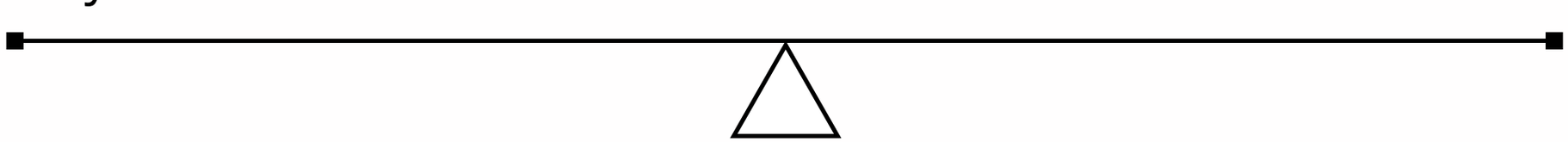

Galaxy Y (GT-S5360)
Galaxy S4 (GT-I9505)
Nexus 4
Galaxy S II (GT-I9100)
LG-L38C
HTC One S
ST26i
HTC One V
GT-P311
XT907
LT30p
LG-E612g
HTCONE
S100
Desire HD
MID
C5170
LG-P705
XT320
S5
A1+
LT26
EVO
ST23i

The Fundamental Challenges



The Fundamental Challenges

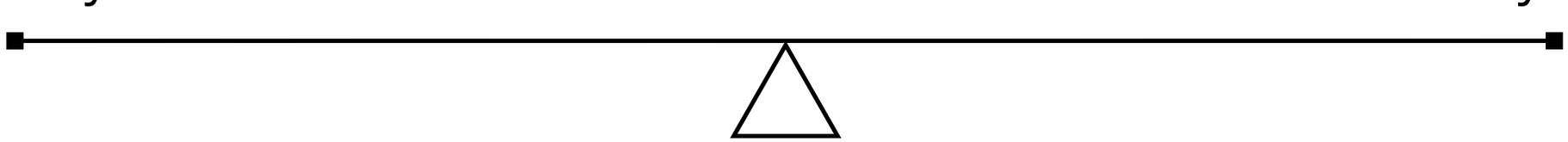
Achieving High Performance
Demanded by End-User



The Fundamental Challenges

Achieving High Performance Demanded by End-User

Conserving Energy Due to Limited Battery Capacity

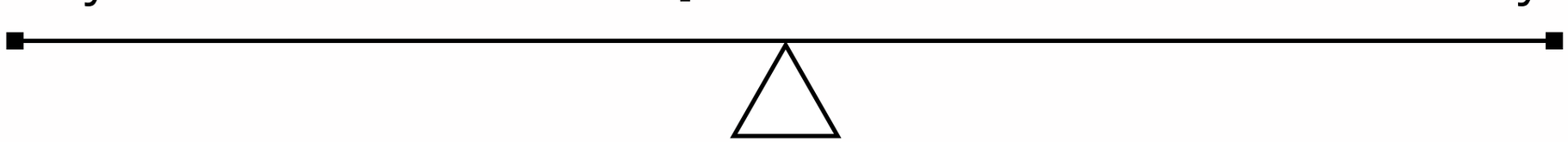


The Fundamental Challenges

Achieving High Performance Demanded by End-User

Conflicting requirements

Conserving Energy Due to Limited Battery Capacity

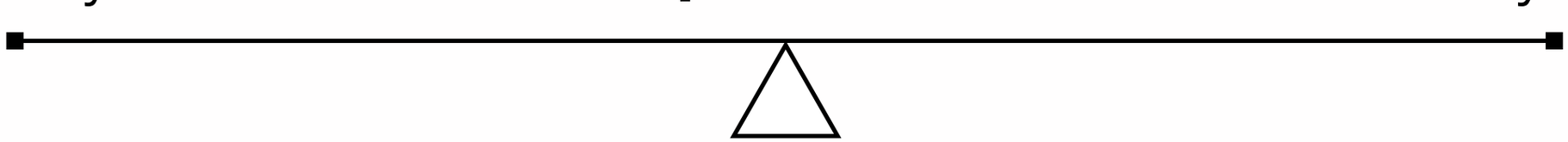


The Fundamental Challenges

Achieving High Performance Demanded by End-User

Conflicting requirements

Conserving Energy Due to Limited Battery Capacity



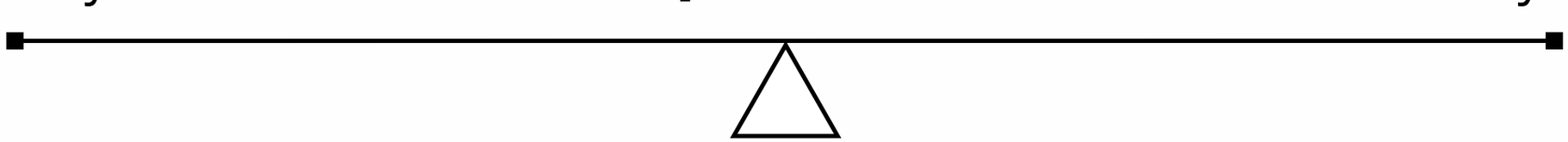
How to achieve high performance with low energy?

The Fundamental Challenges

Achieving High Performance Demanded by End-User

Conflicting requirements

Conserving Energy Due to Limited Battery Capacity



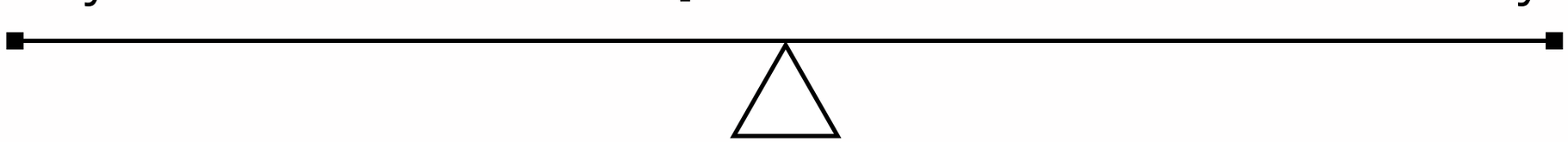
A mobile architecture to achieve high performance with low energy?

The Fundamental Challenges

Achieving High Performance Demanded by End-User

Conflicting requirements

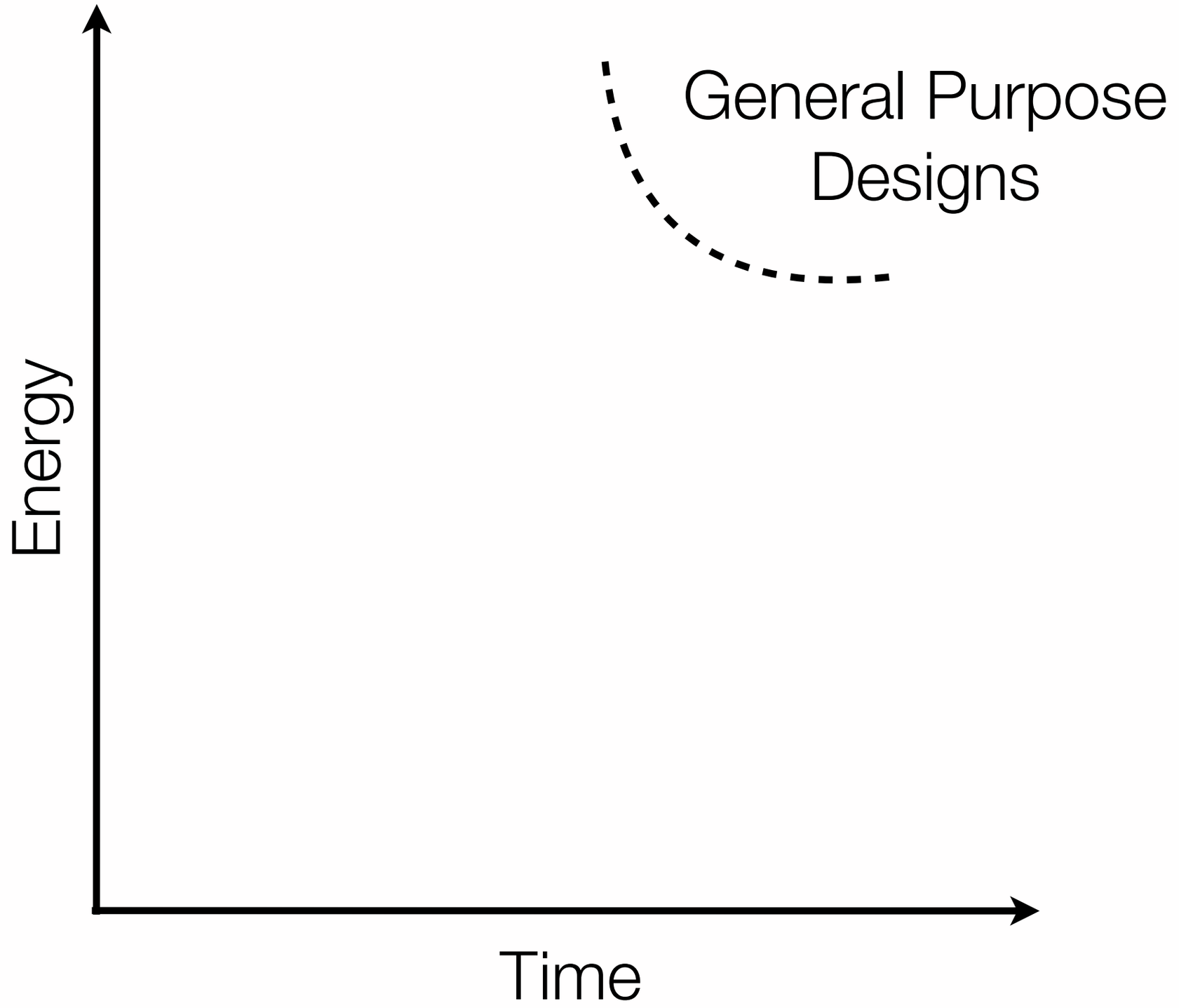
Conserving Energy Due to Limited Battery Capacity



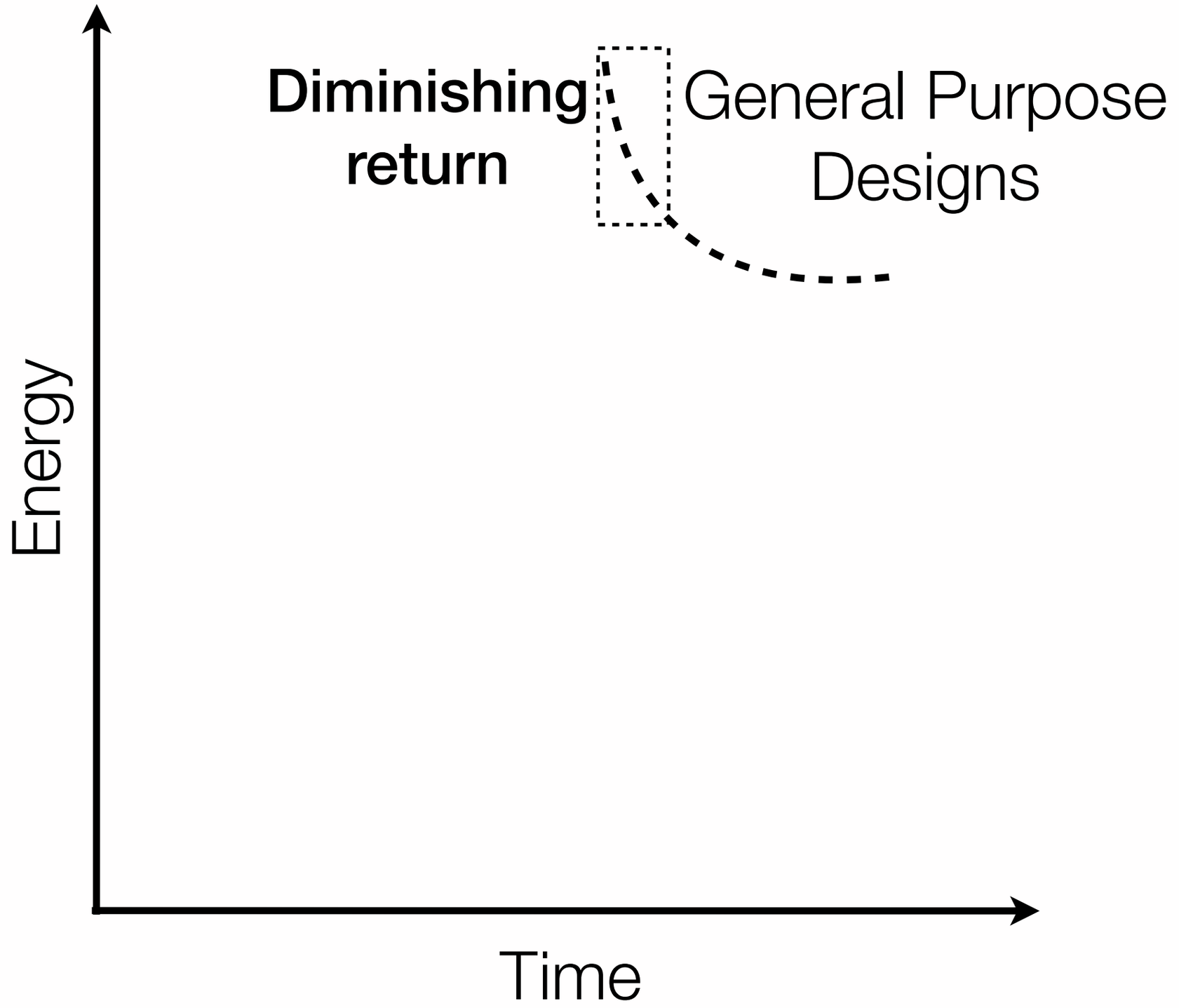
WebCore:

A mobile architecture to achieve high performance with low energy

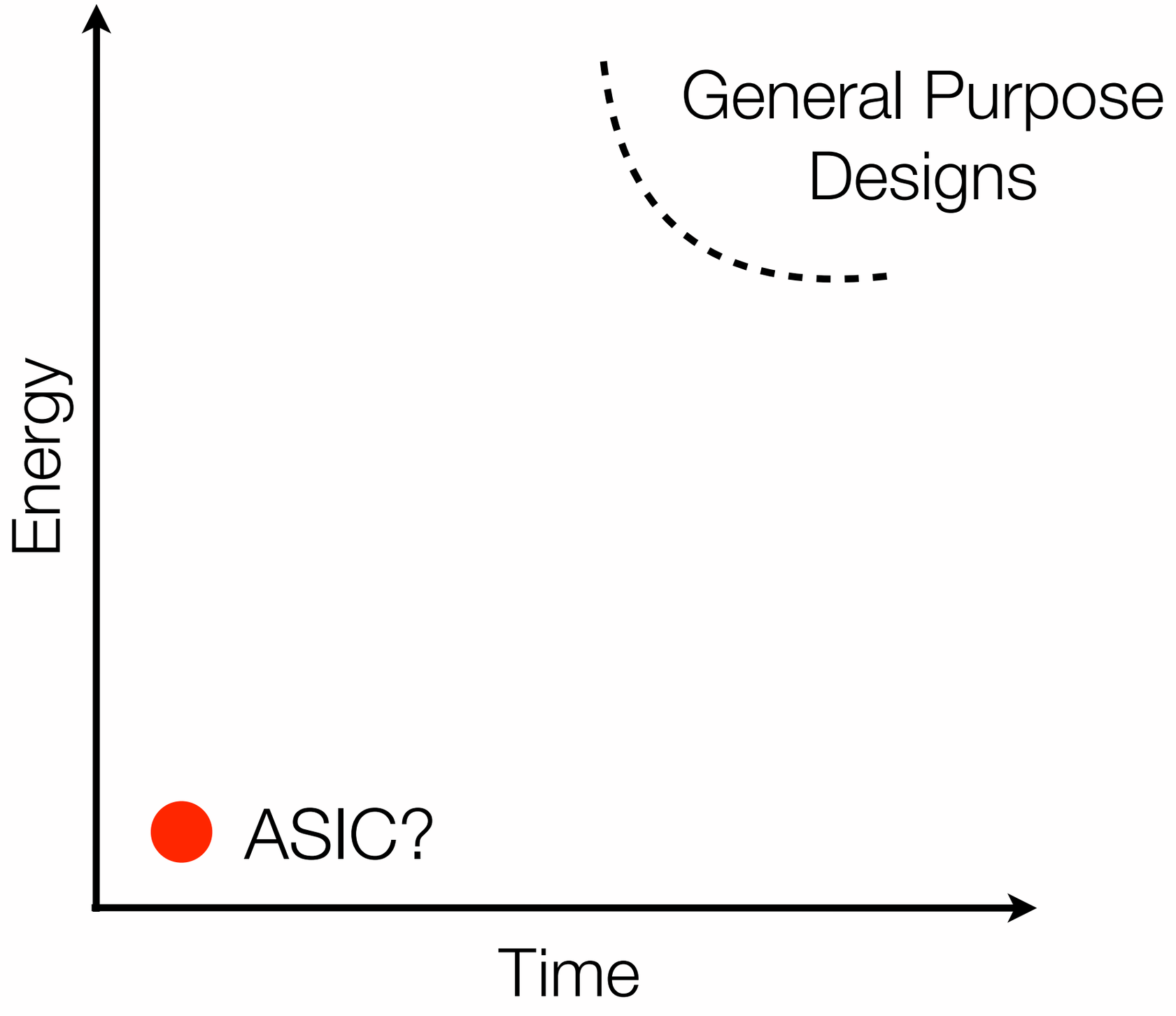
Executive Summary



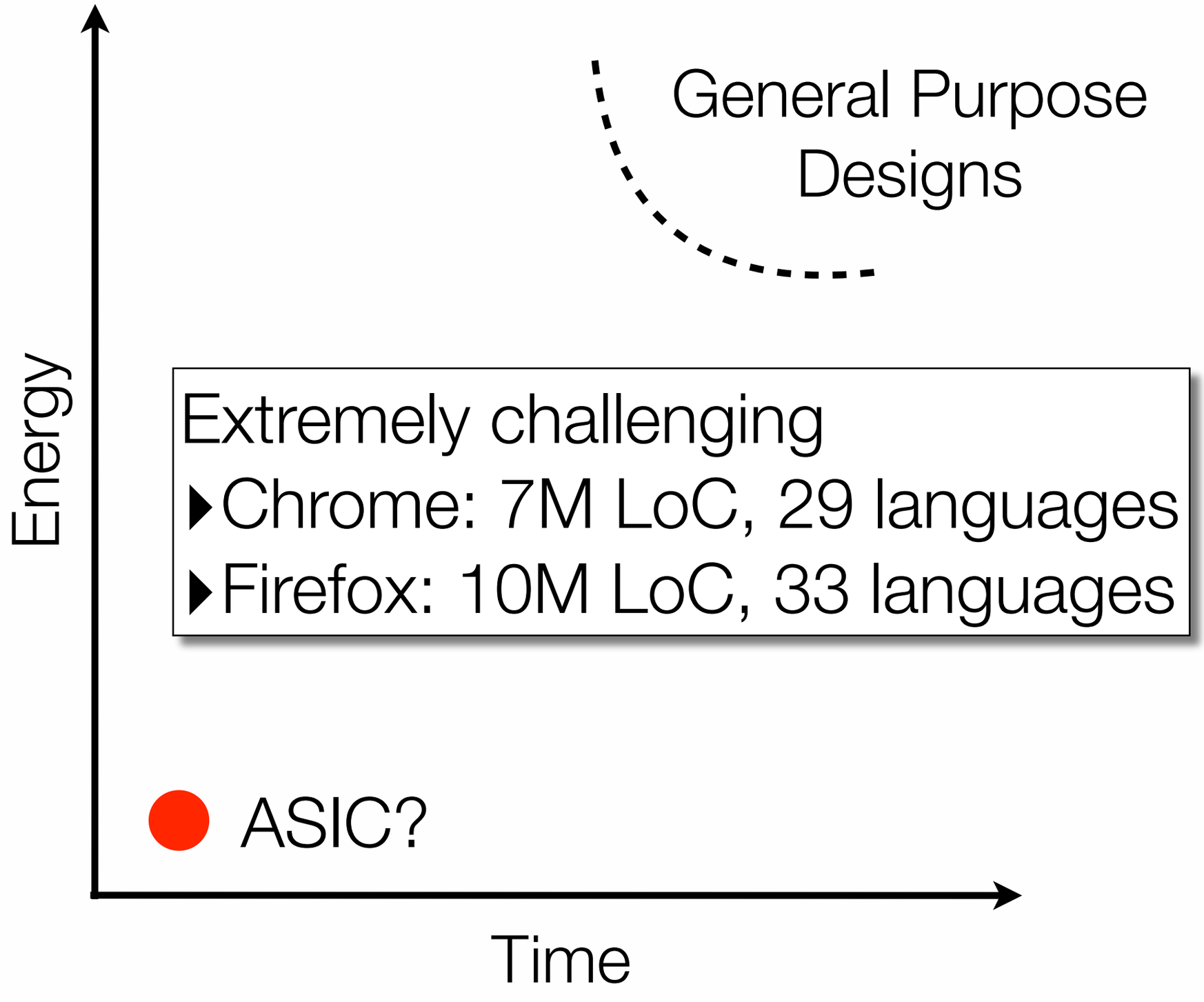
Executive Summary



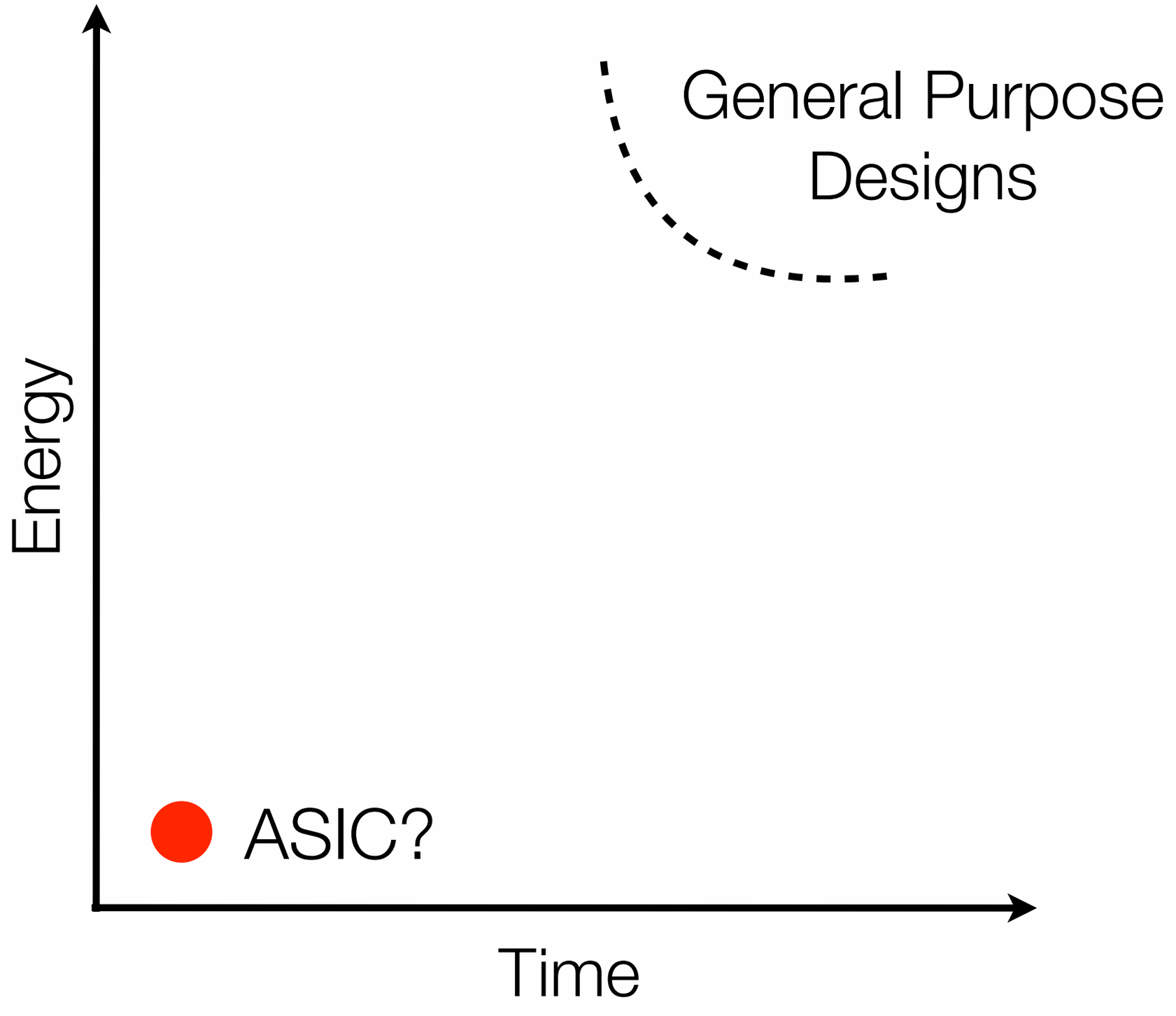
Executive Summary



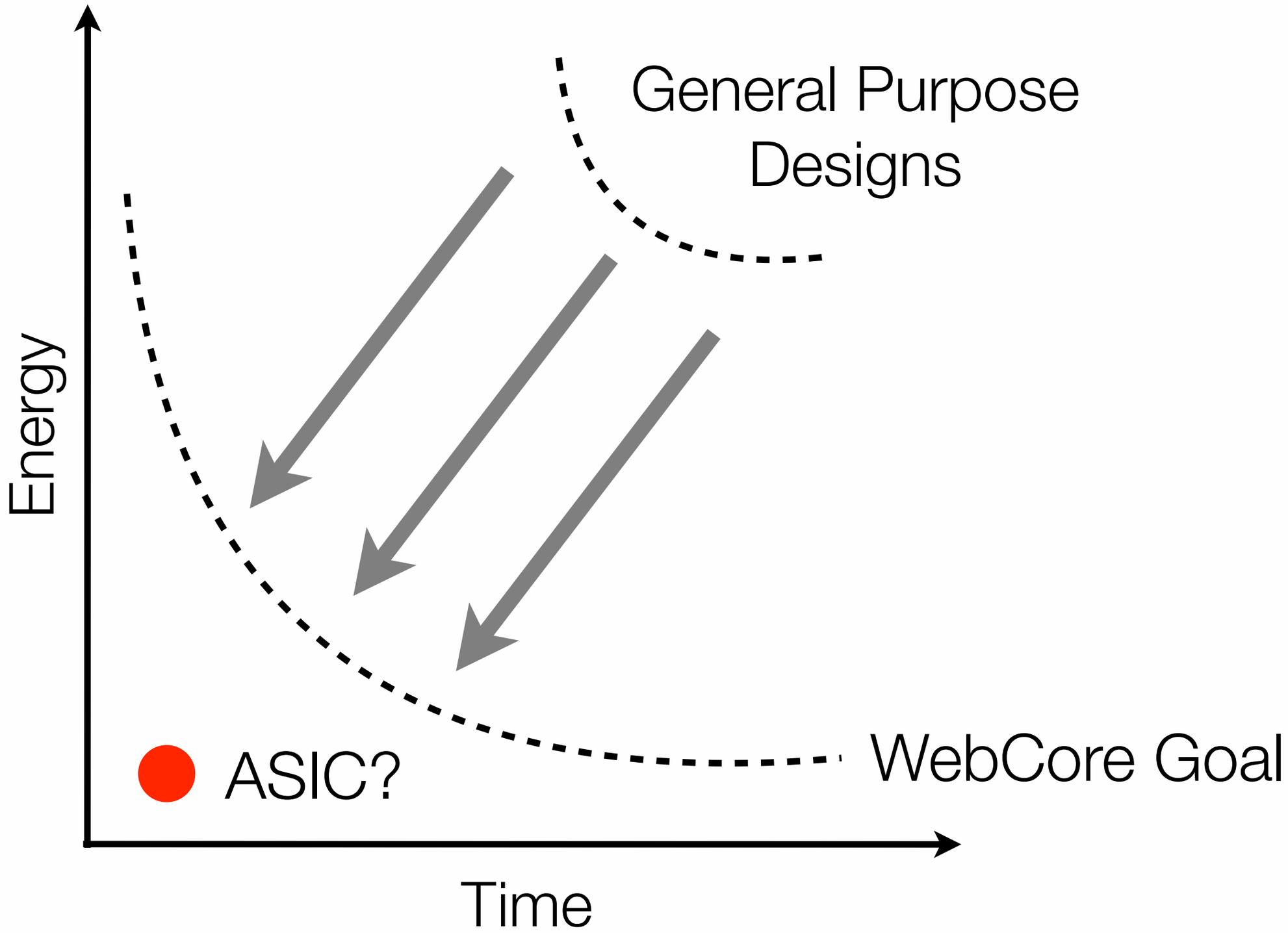
Executive Summary



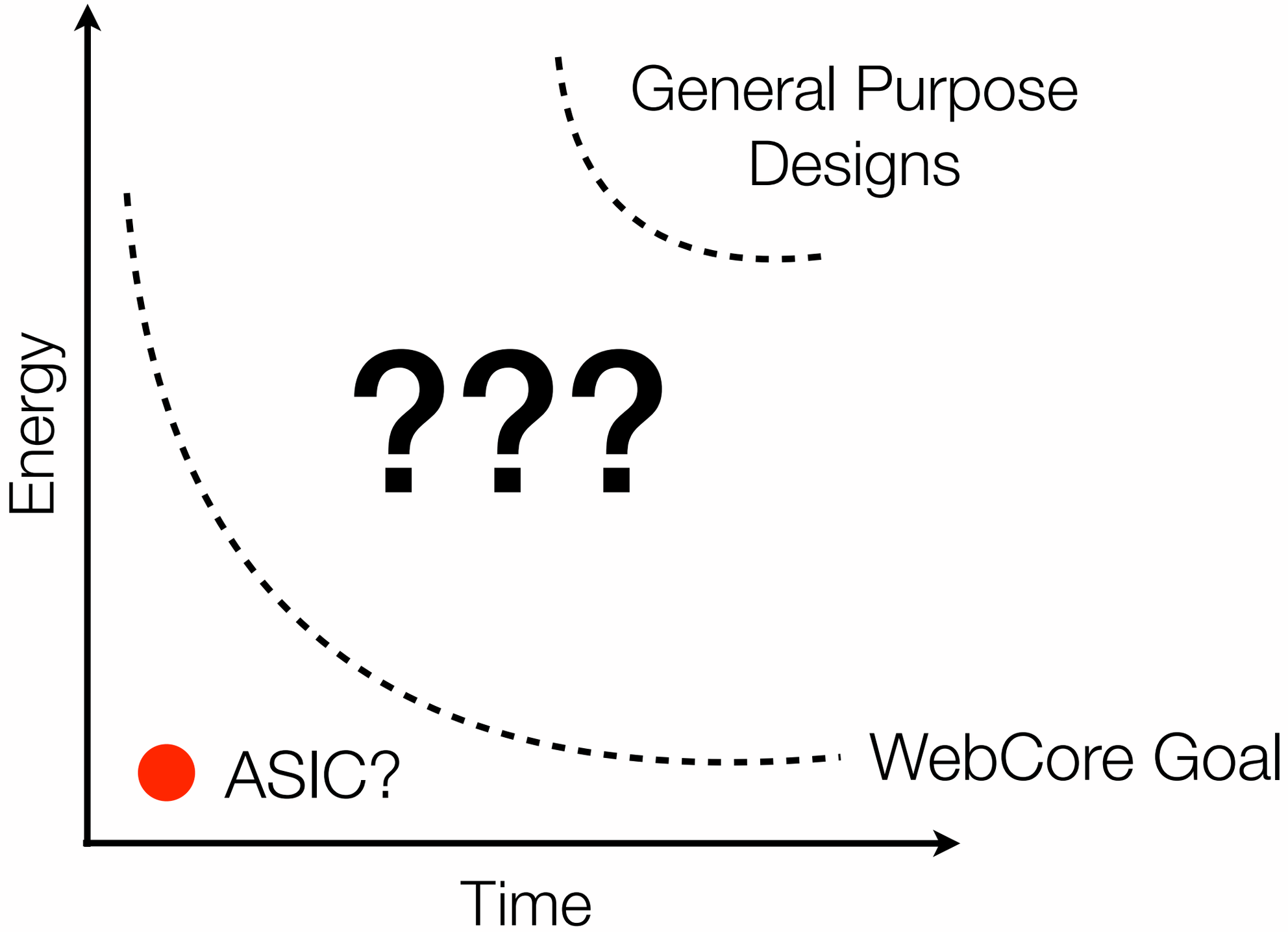
Executive Summary



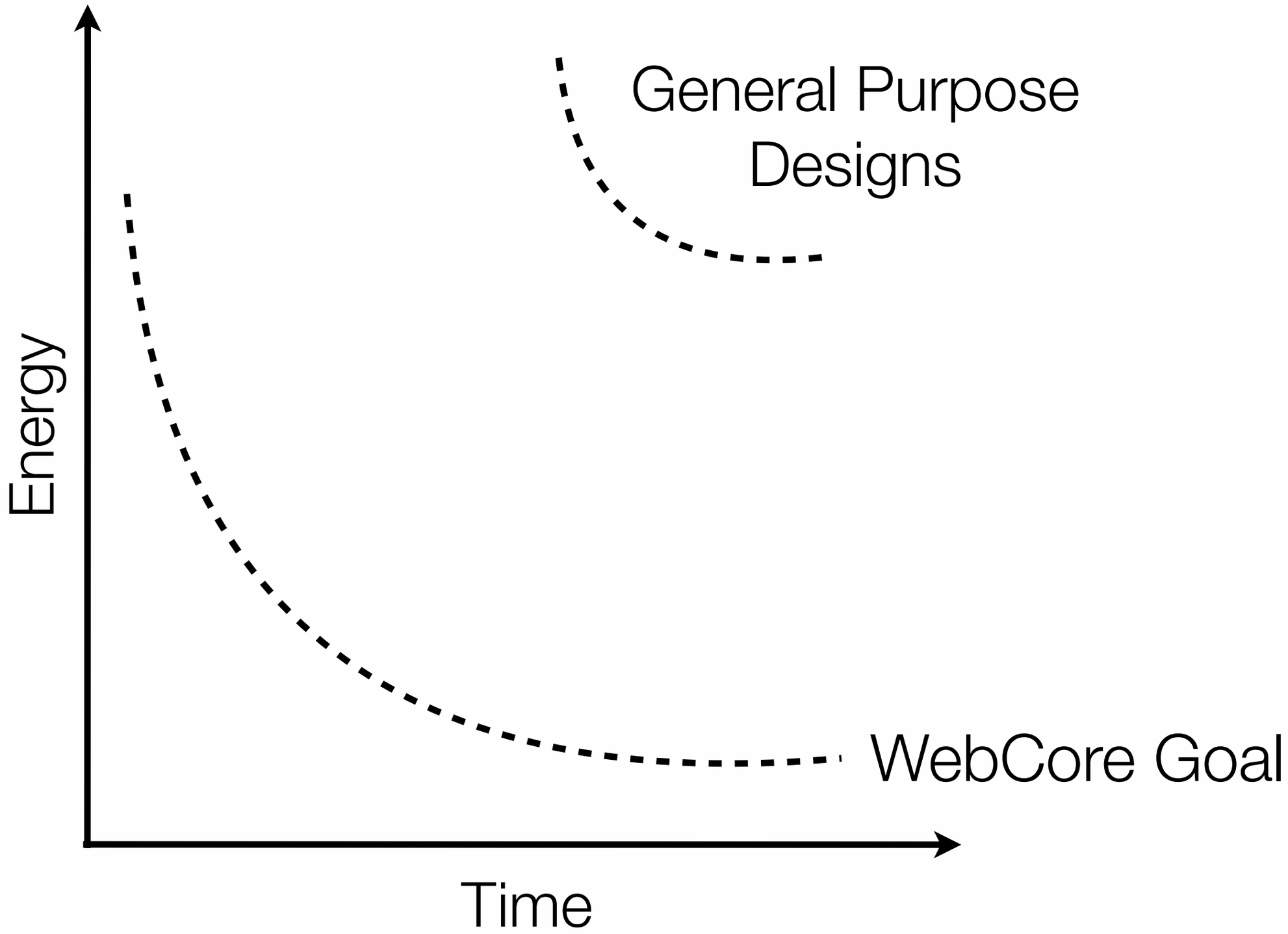
Executive Summary



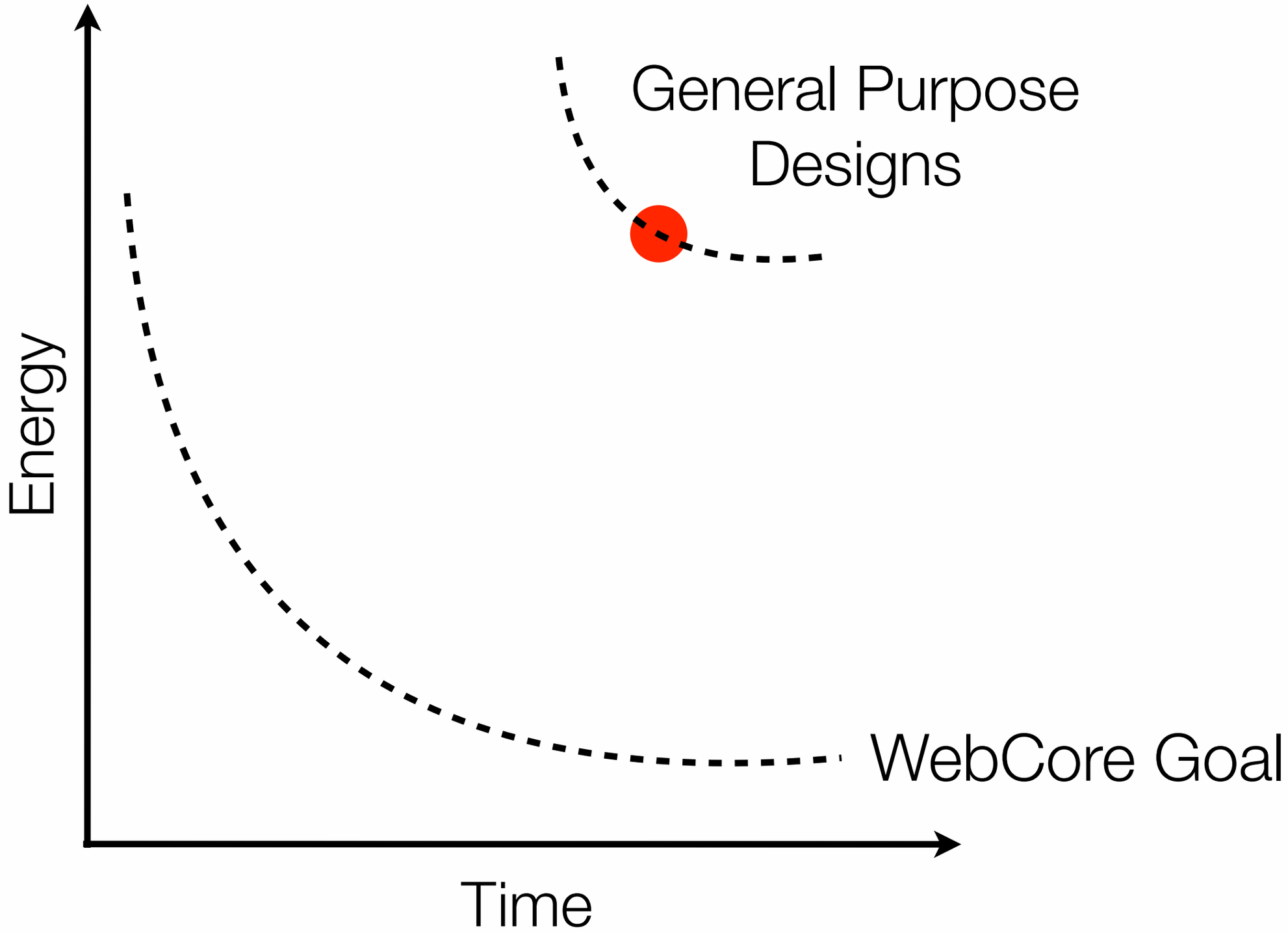
Executive Summary



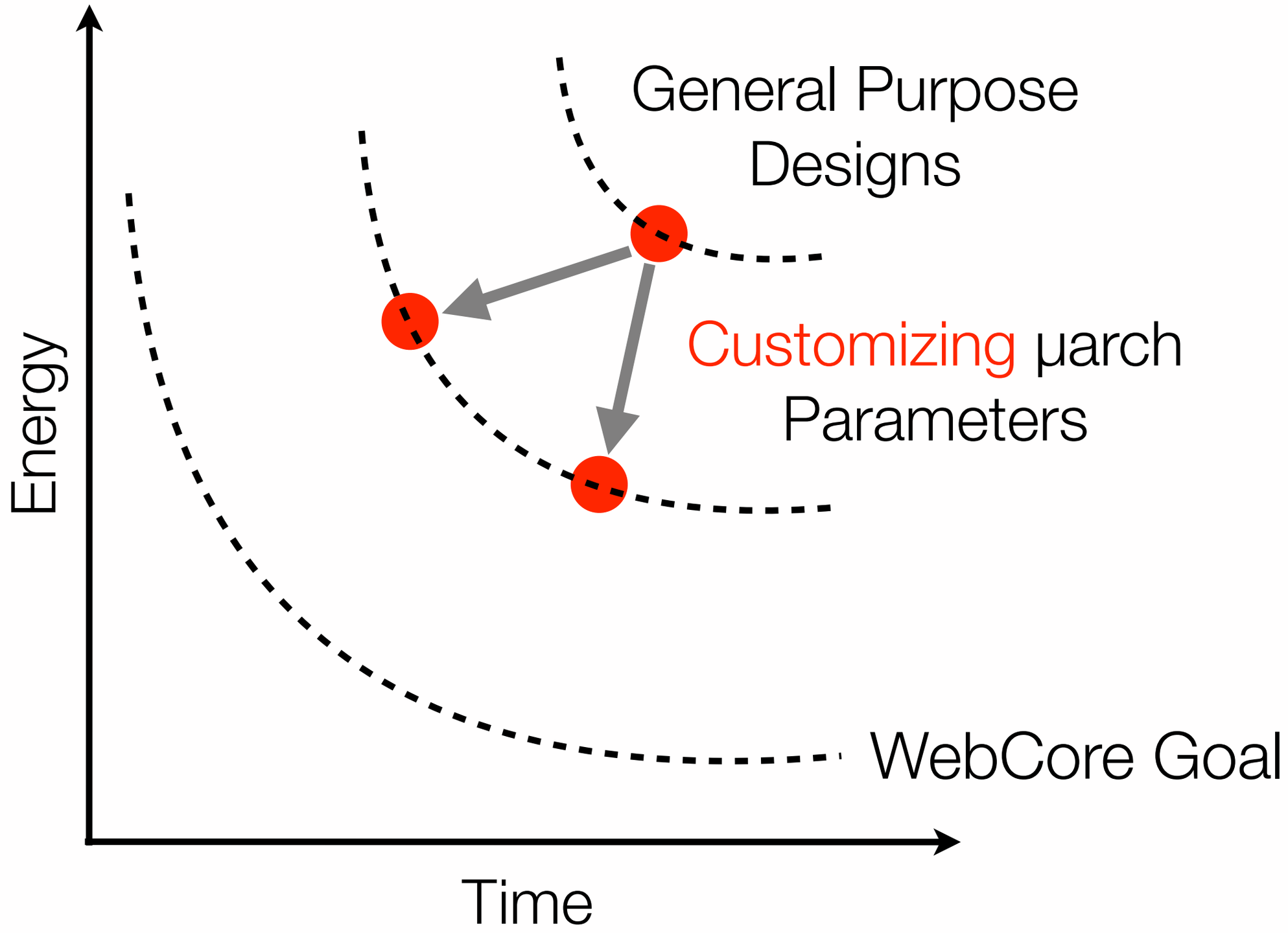
Executive Summary



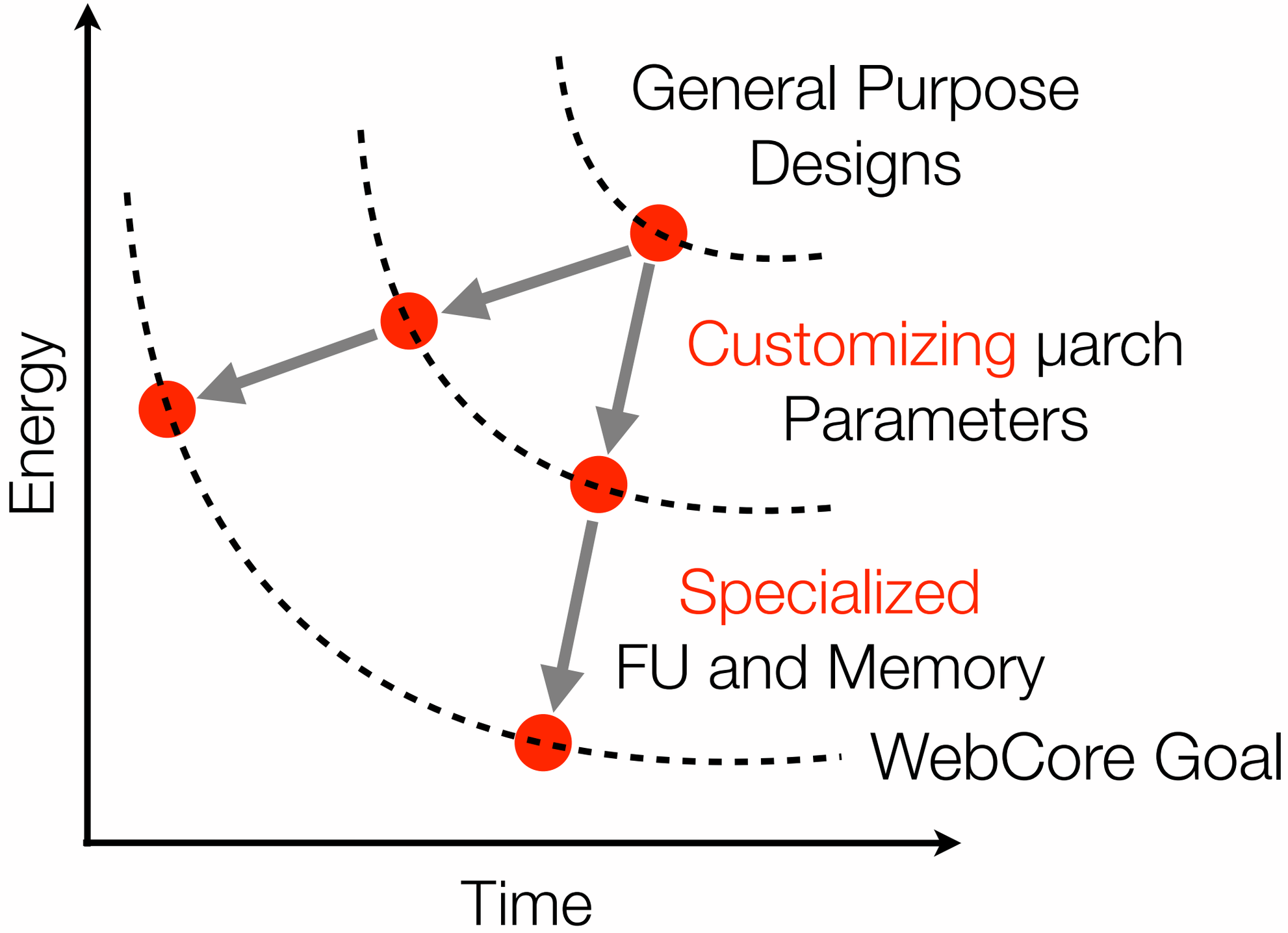
Executive Summary



Executive Summary



Executive Summary



Agenda of Today's Talk

- Motivation of our work: energy-efficiency of the mobile Web

Agenda of Today's Talk

- Motivation of our work: energy-efficiency of the mobile Web
- How does WebCore improve the energy-efficiency?
  - Customization
  - Specialization

Agenda of Today's Talk

- Motivation of our work: energy-efficiency of the mobile Web
- How does WebCore improve the energy-efficiency?
  - Customization
  - Specialization
- Evaluation Results

Agenda of Today's Talk

- Motivation of our work: energy-efficiency of the mobile Web
- How does WebCore improve the energy-efficiency?
  - Customization
  - Specialization
- Evaluation Results
- Related Work

Agenda of Today's Talk

- Motivation of our work: energy-efficiency of the mobile Web
- How does WebCore improve the energy-efficiency?
  - **Customization**
  - Specialization
- Evaluation Results
- Related Work

Customization: Find the Ideal General Purpose Baseline Architecture

Customization: Find the Ideal General Purpose Baseline Architecture

- Why customization?!?

Customization: Find the Ideal General Purpose Baseline Architecture

- Why customization?!?
- What is a proper general purpose baseline architecture?

Customization: Find the Ideal General Purpose Baseline Architecture

- Why customization?!?
- What is a proper general purpose baseline architecture?
  - Out-of-order (Silvermont, A15) or in-order (Saltwell, A7)?

Customization: Find the Ideal General Purpose Baseline Architecture

- Why customization?!?
- What is a proper general purpose baseline architecture?
  - Out-of-order (Silvermont, A15) or in-order (Saltwell, A7)?
  - Are existing general purpose mobile designs ideal?

Customization: Find the Ideal General Purpose Baseline Architecture

- Why customization?!?
- What is a proper general purpose baseline architecture?
  - Out-of-order (Silvermont, A15) or in-order (Saltwell, A7)?
  - Are existing general purpose mobile designs ideal?
- Exhaustive design space exploration

Customization: Find the Ideal General Purpose Baseline Architecture

- Why customi
- What is a pro
  - Out-of-orde
  - Are existing
- Exhaustive d

Parameters	Measure
Issue width	count
# Functional units	count
Load queue size	# entries
Store queue size	# entries
Branch prediction size	log_2(#entries)
ROB size	# entries
# Physical registers	# entries
L1 I-cache size	log_2(KB)
L1 I-cache delay	cycles
L1 D-cache size	log_2(KB)
L1 D-cache delay	cycles
L2 cache size	log_2(KB)
L2 cache delay	cycles

Customization: Find the Ideal General Purpose Baseline Architecture

- Why customi
- What is a pro
  - Out-of-orde
  - Are existing
- Exhaustive d

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) Setup

- Integrated power (McPAT) and performance x86 full-system simulator (Marss86)
- WebKit engine in the Chromium Web browser

Design Space Exploration (DSE) Setup

- Integrated power (McPAT) and performance x86 full-system simulator (Marss86)
- WebKit engine in the Chromium Web browser

Design Space Exploration (DSE) Setup

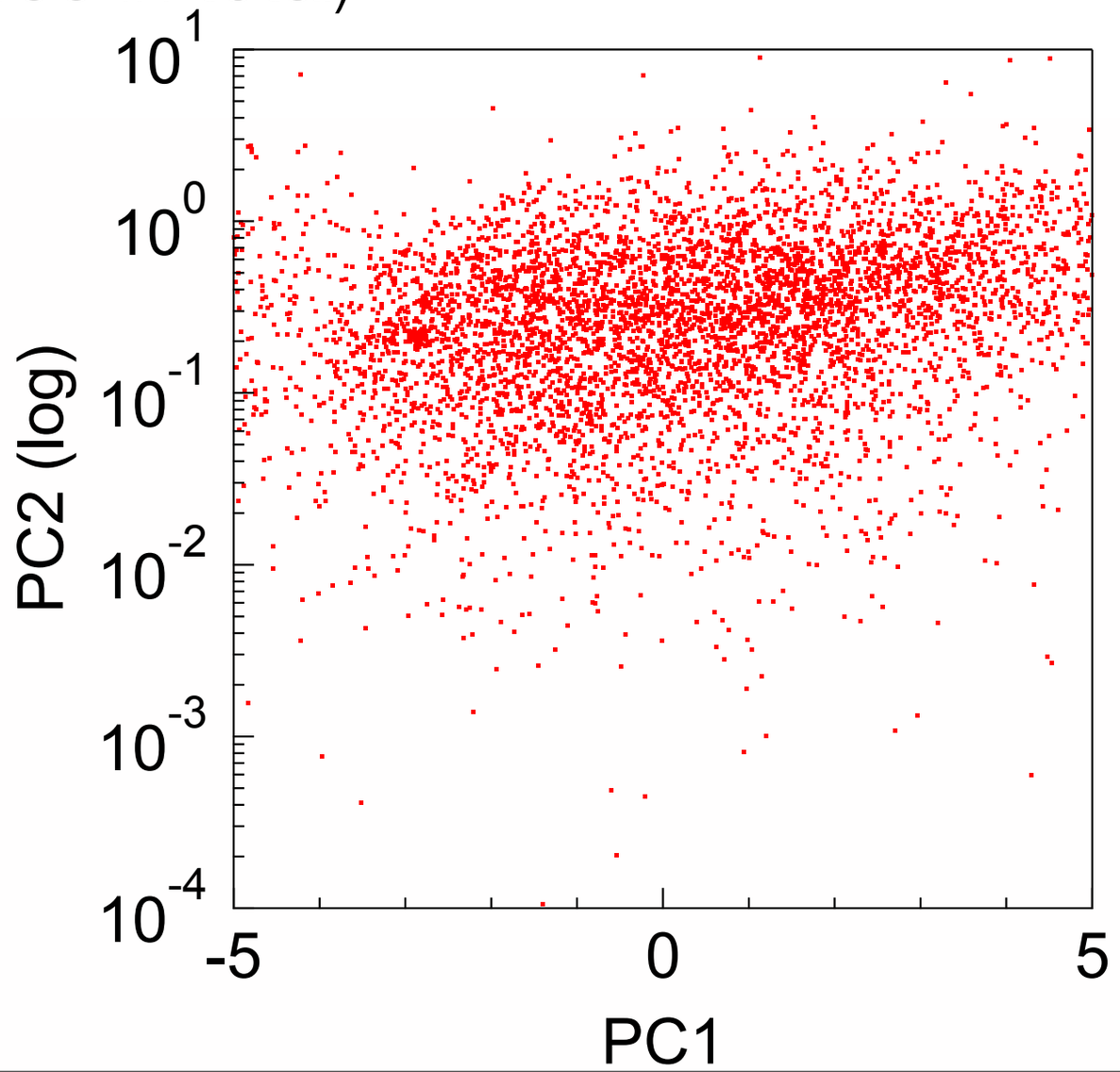
- Webpages selection using PCA

Design Space Exploration (DSE) Setup

- Webpages selection using PCA
  - PCs calculated from webpage-inherent and µarch-dependent features (~400 in total)

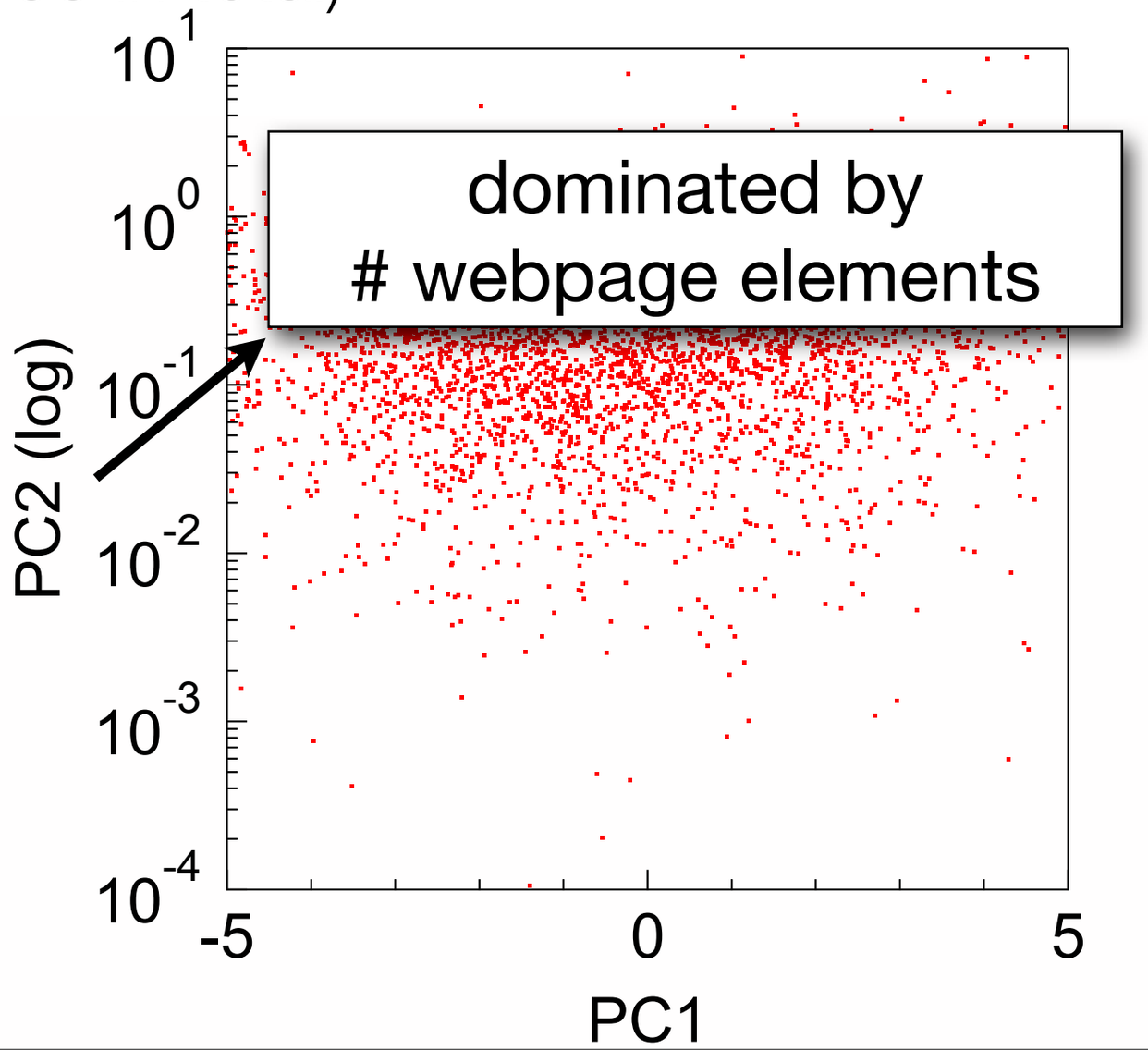
Design Space Exploration (DSE) Setup

- Webpages selection using PCA
  - PCs calculated from webpage-inherent and µarch-dependent features (~400 in total)



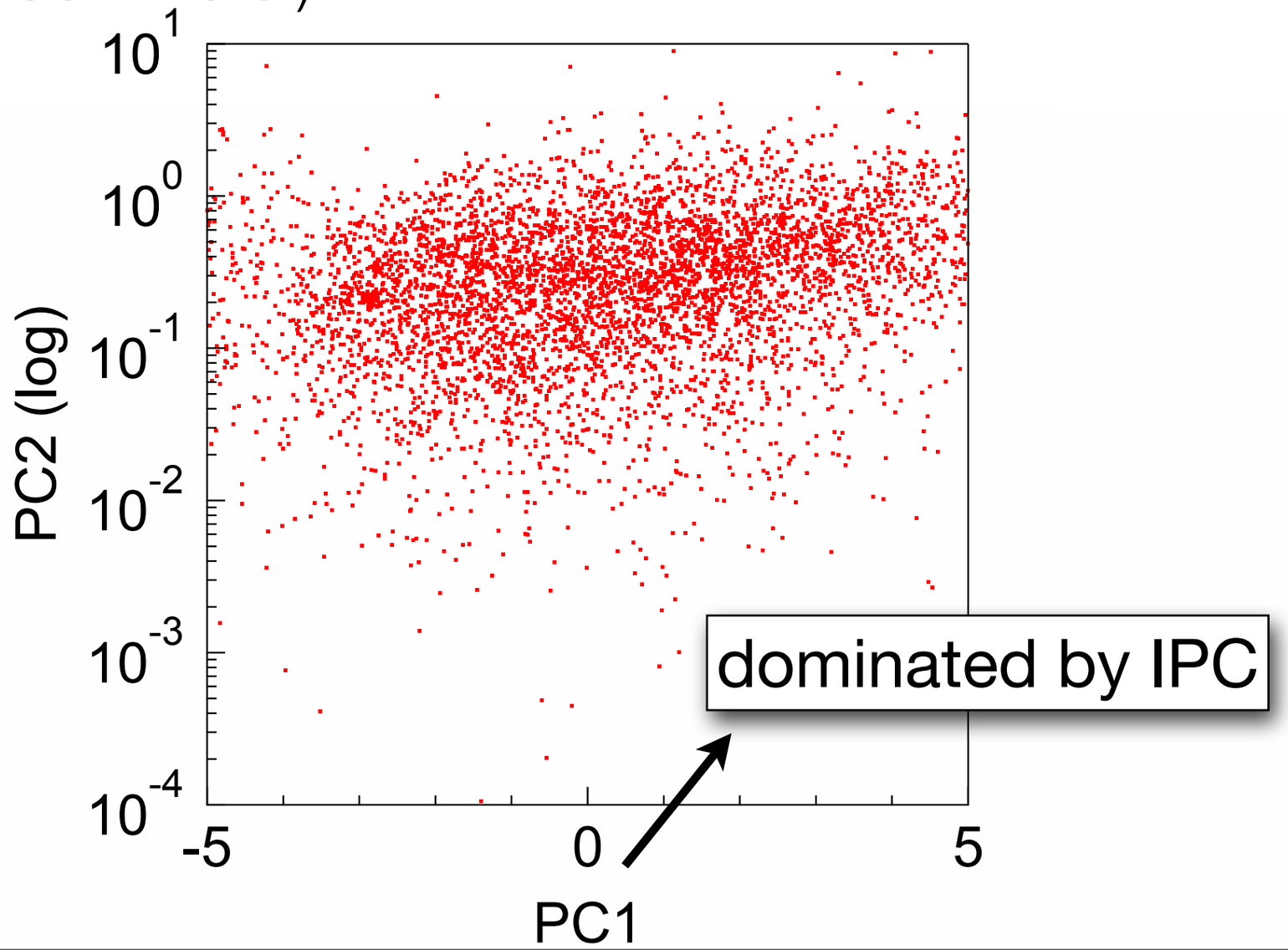
Design Space Exploration (DSE) Setup

- Webpages selection using PCA
  - PCs calculated from **webpage-inherent** and µarch-dependent features (~400 in total)



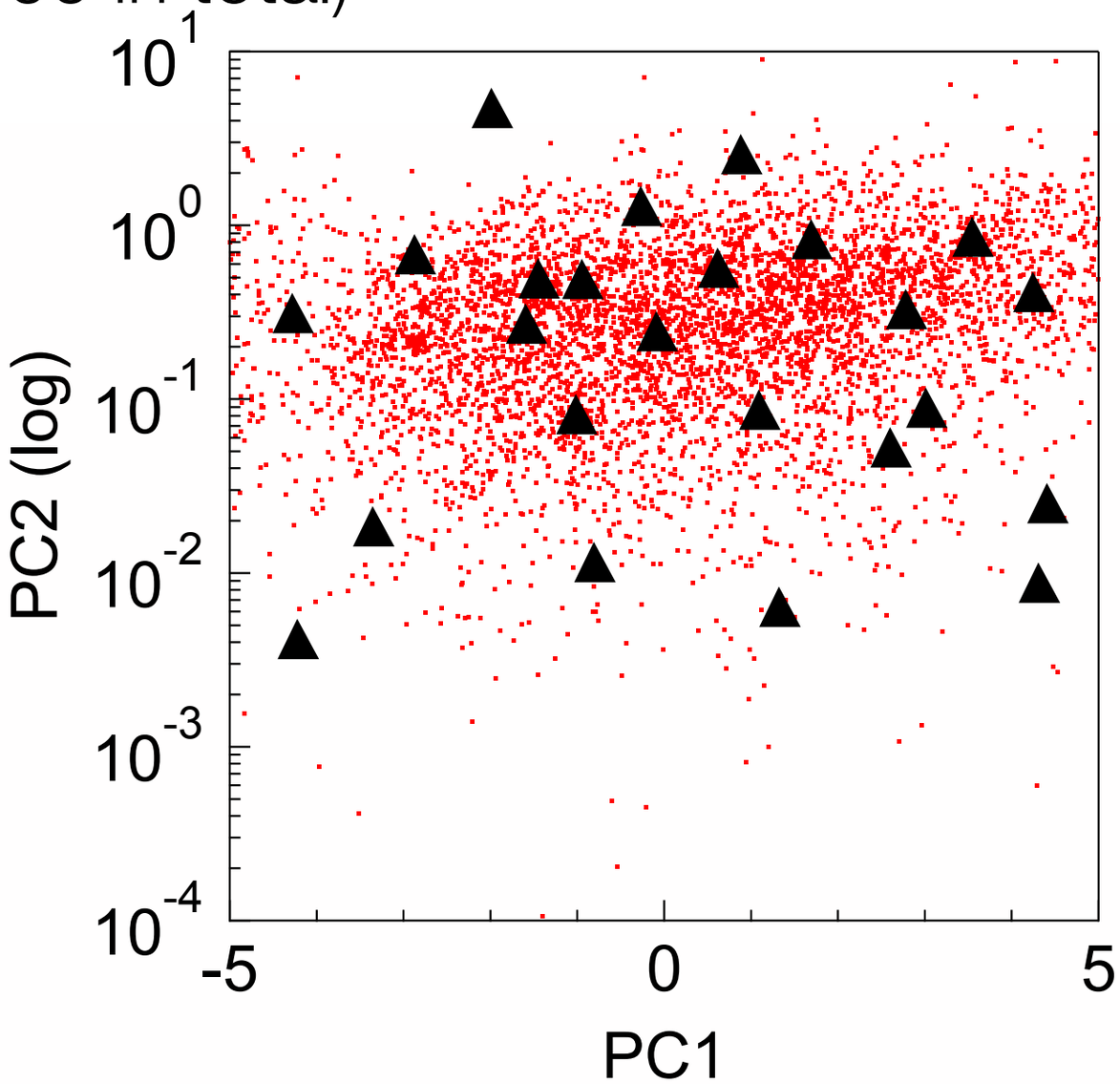
Design Space Exploration (DSE) Setup

- Webpages selection using PCA
  - PCs calculated from webpage-inherent and **µarch-dependent** features (~400 in total)

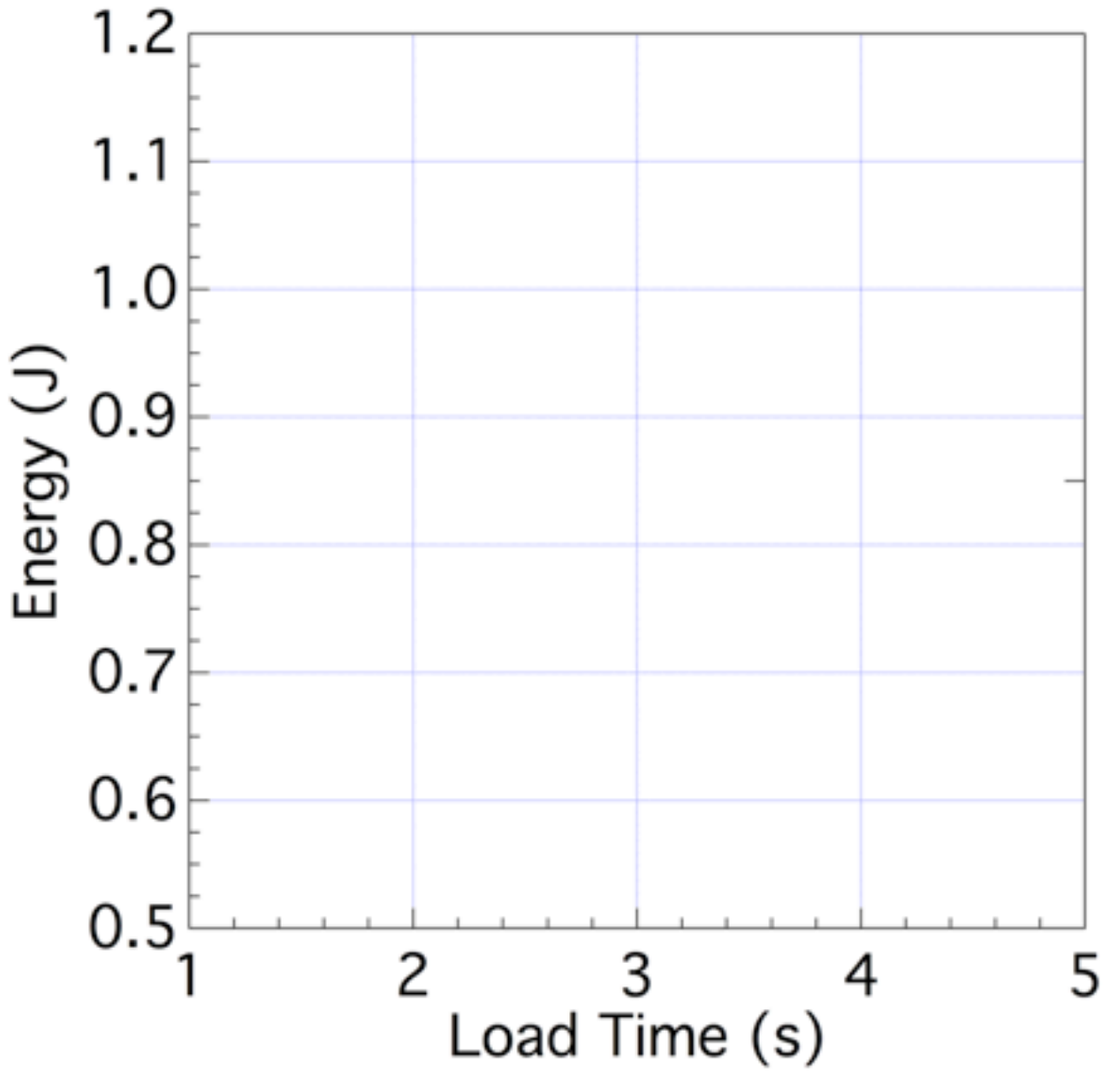


Design Space Exploration (DSE) Setup

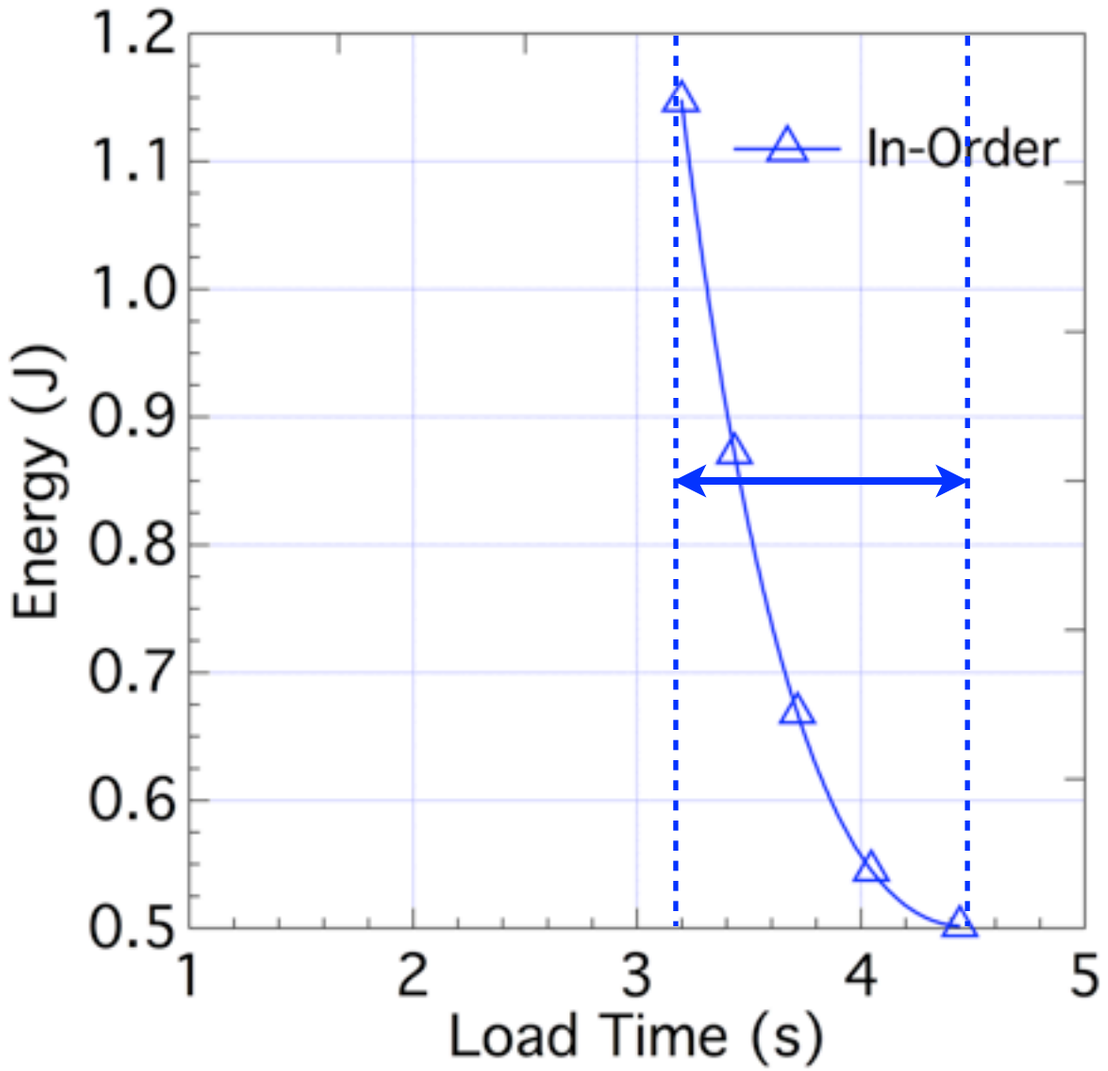
- Webpages selection using PCA
  - PCs calculated from webpage-inherent and µarch-dependent features (~400 in total)



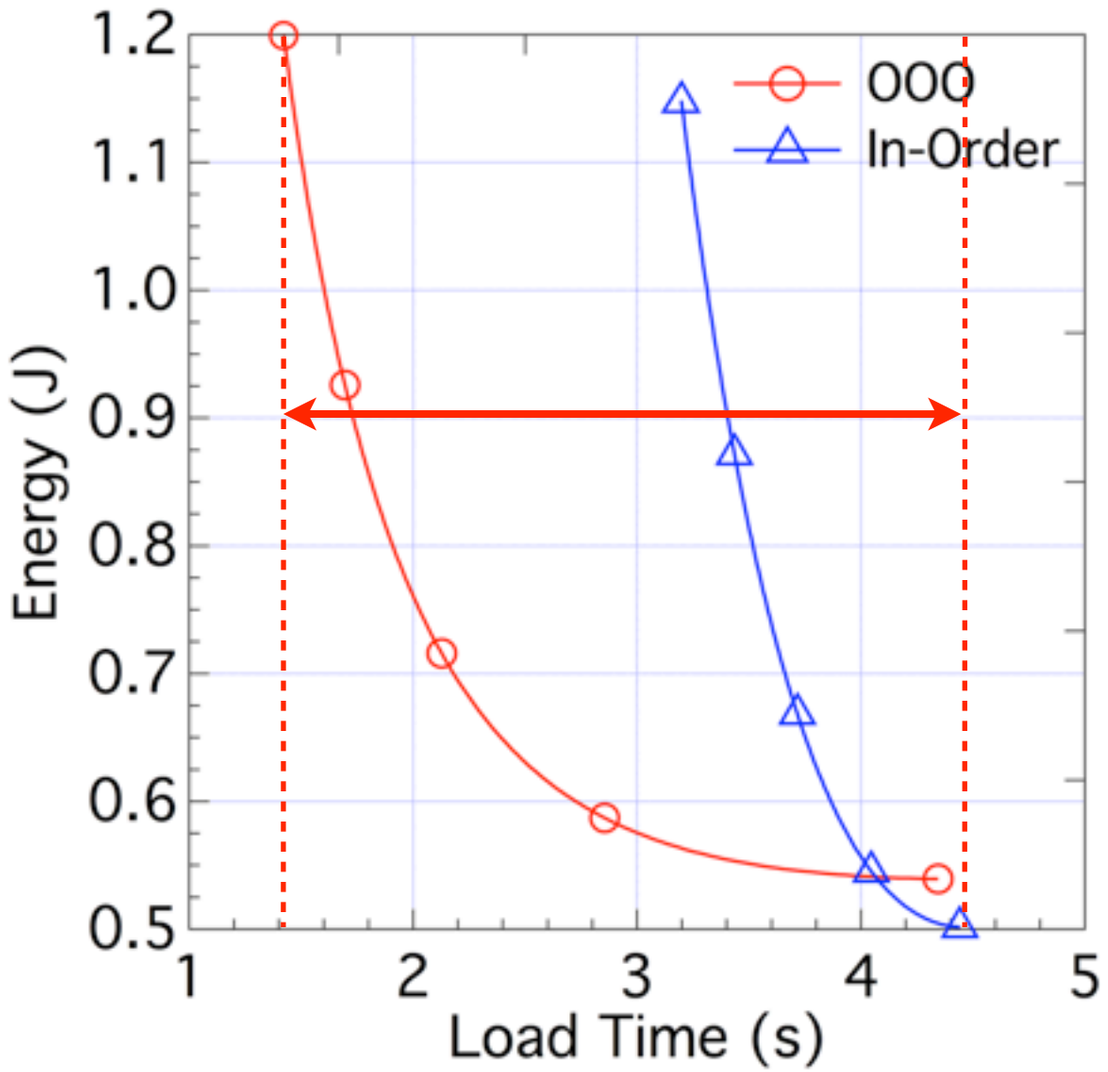
Design Space Exploration (DSE) Findings



Design Space Exploration (DSE) Findings

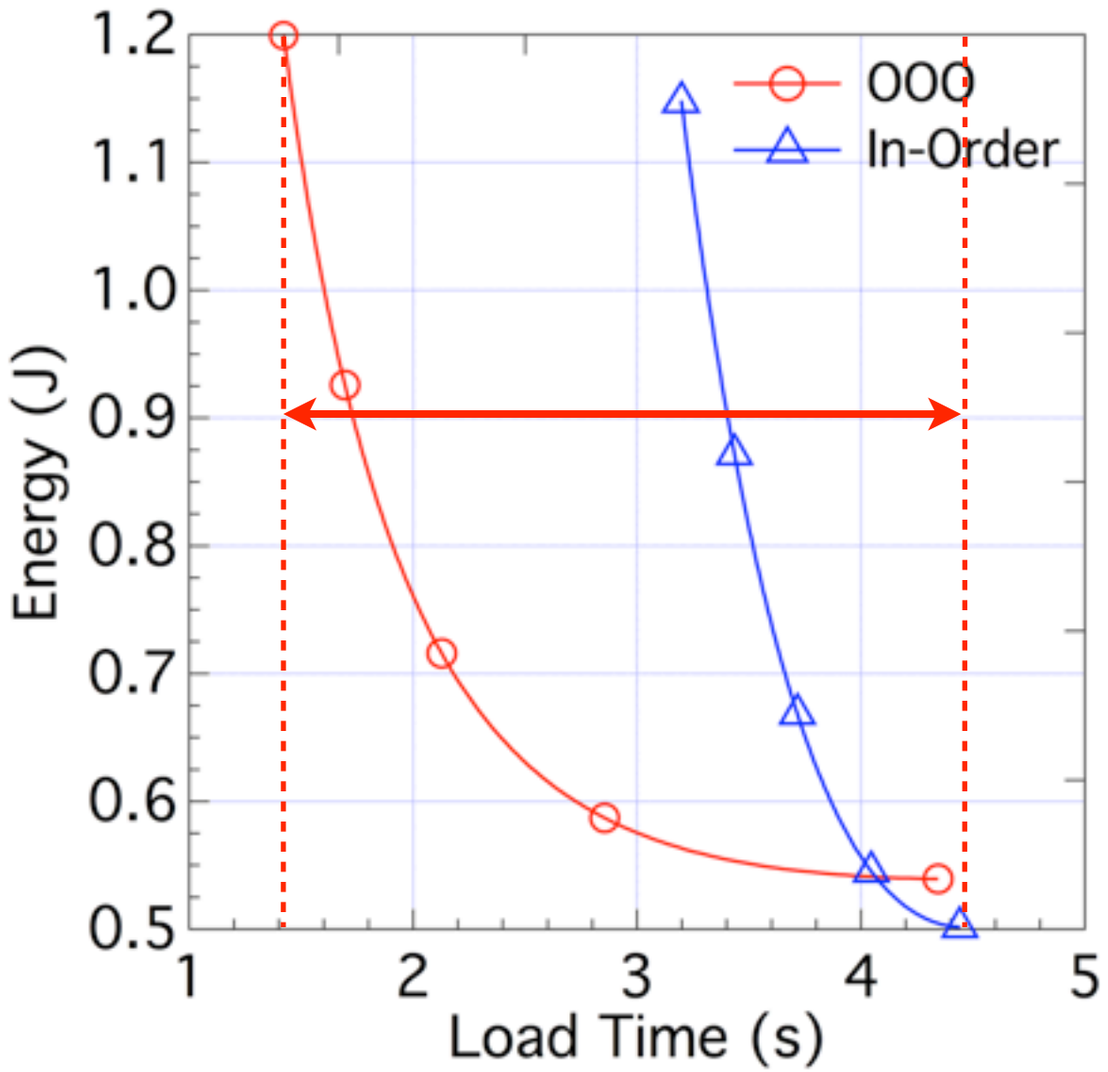


Design Space Exploration (DSE) Findings



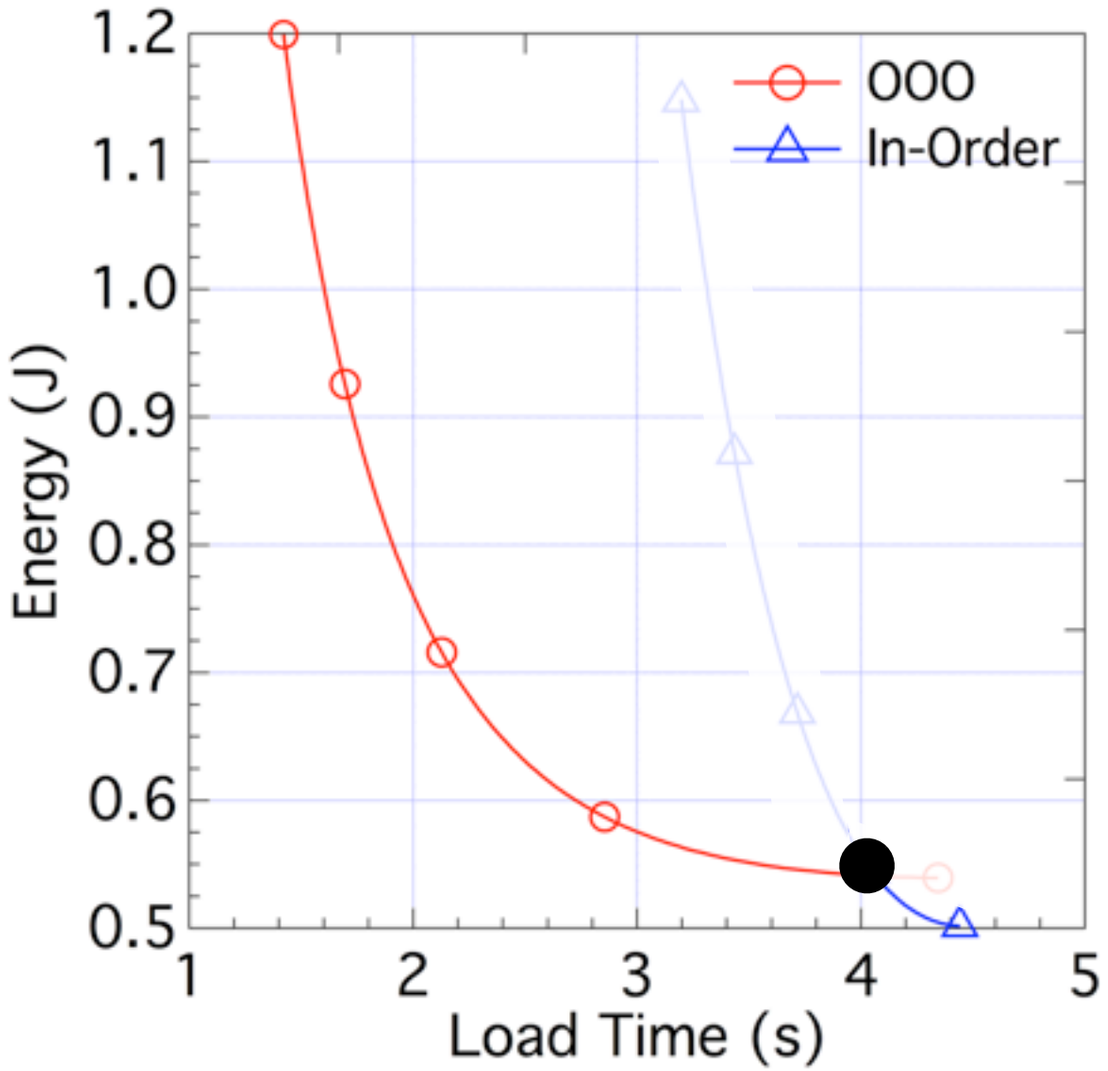
Design Space Exploration (DSE) Findings

- Out-of-order μarchitecture is much more flexible



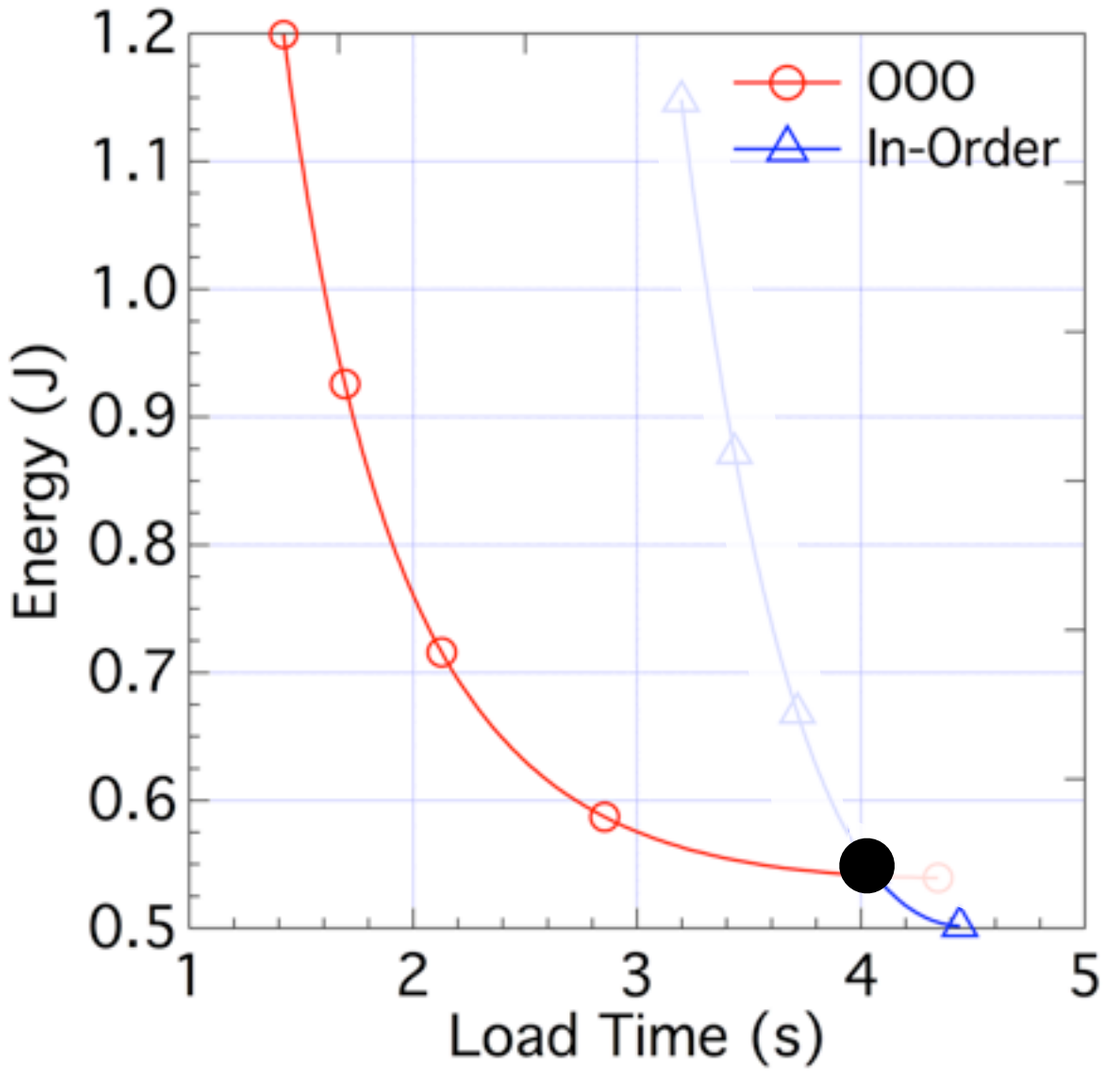
Design Space Exploration (DSE) Findings

- Out-of-order μarchitecture is much more flexible



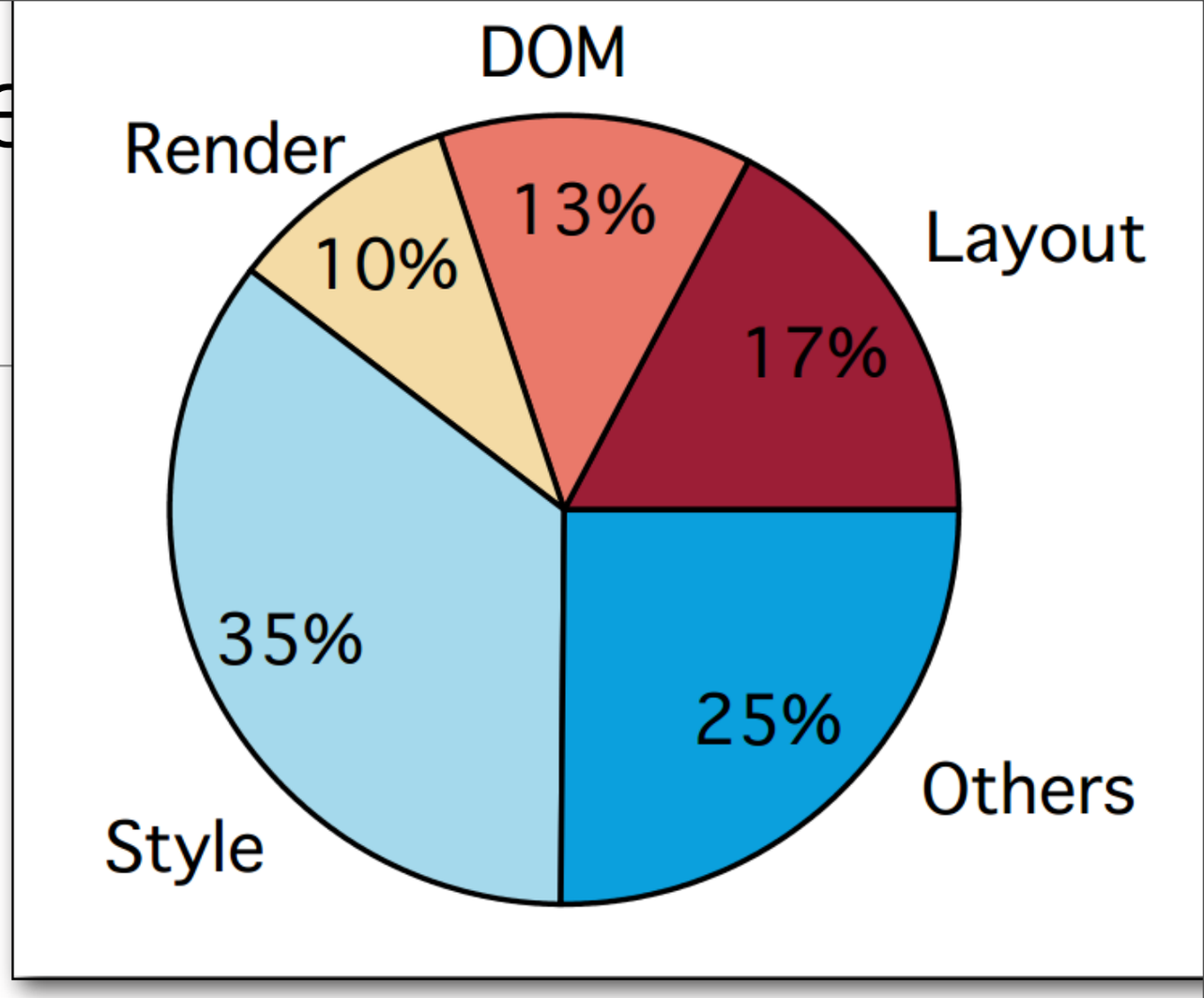
Design Space Exploration (DSE) Findings

- Out-of-order μarchitecture is much more flexible
- In-order cores are acceptable if end-users can tolerate latency



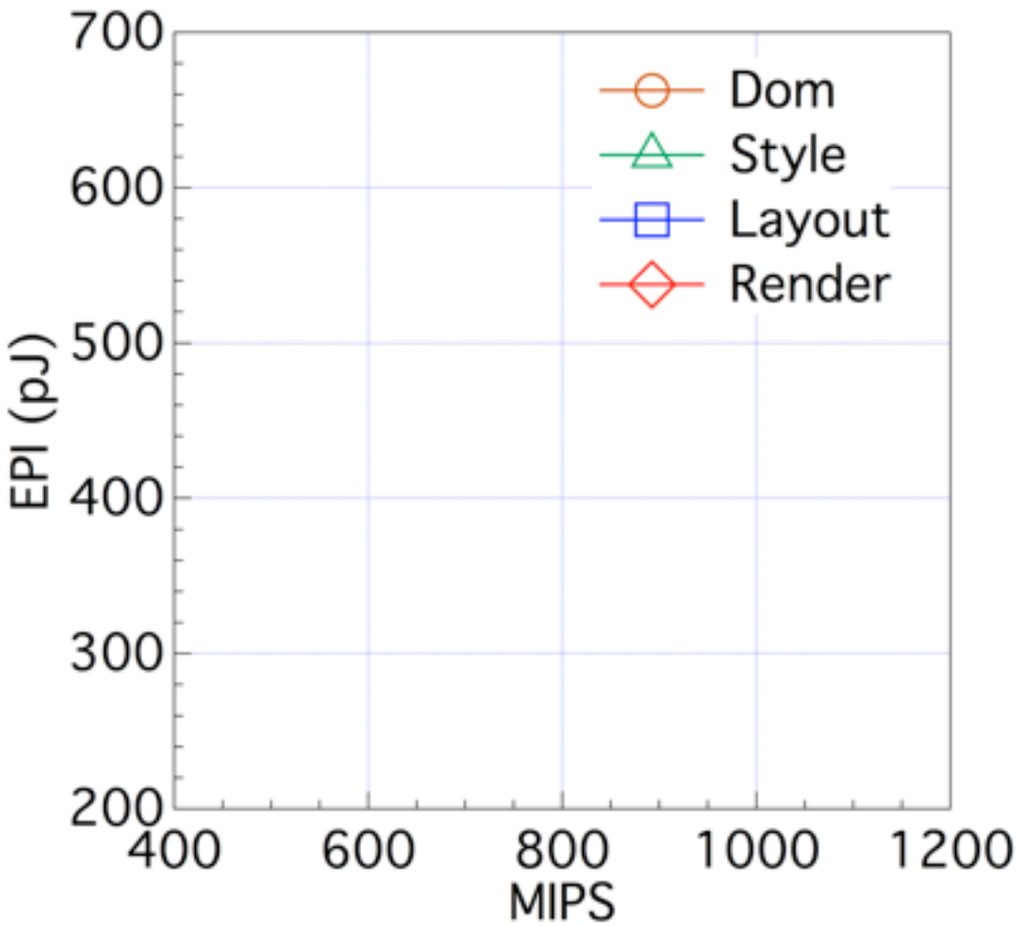
Understand the Difference Using Kernel Knowledge

Understand the Differe
Knowledge



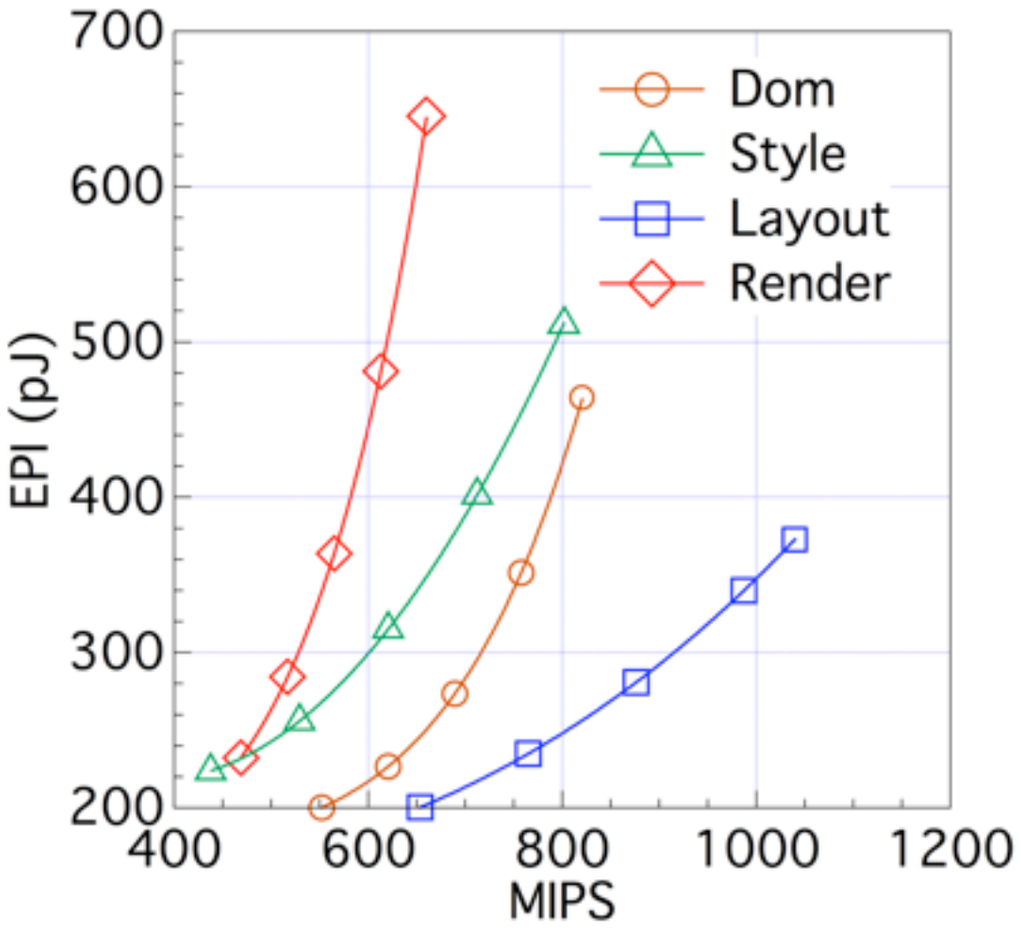
Execution time breakdown

Understand the Difference Using Kernel Knowledge



In-order design

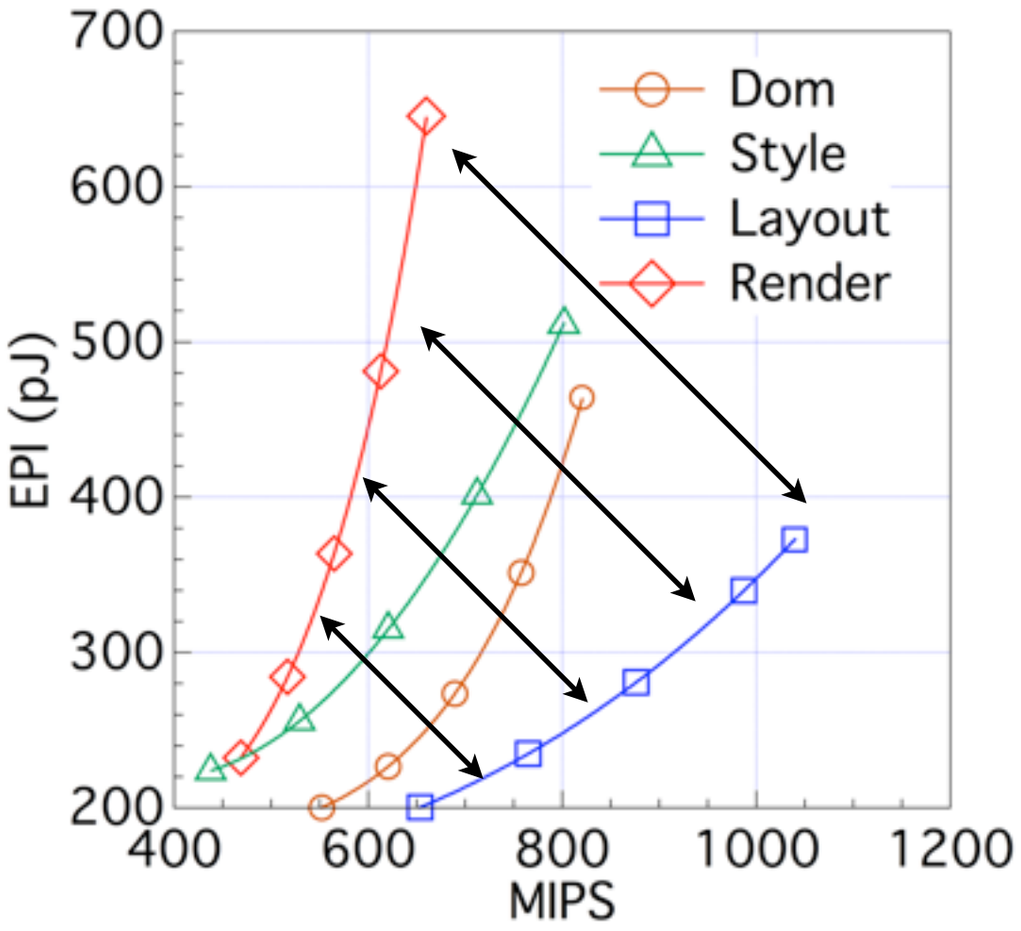
Understand the Difference Using Kernel Knowledge



In-order design

Understand the Difference Using Kernel Knowledge

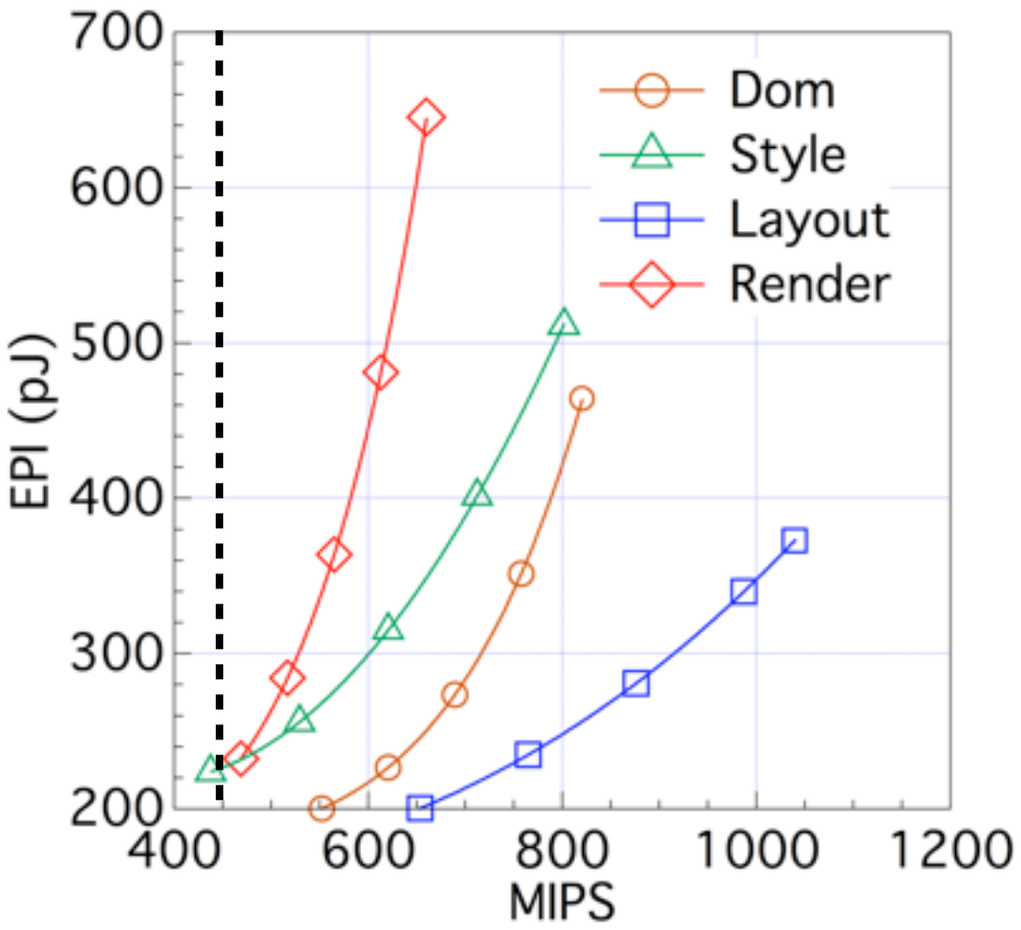
- In-order designs show strong kernel variance



In-order design

Understand the Difference Using Kernel Knowledge

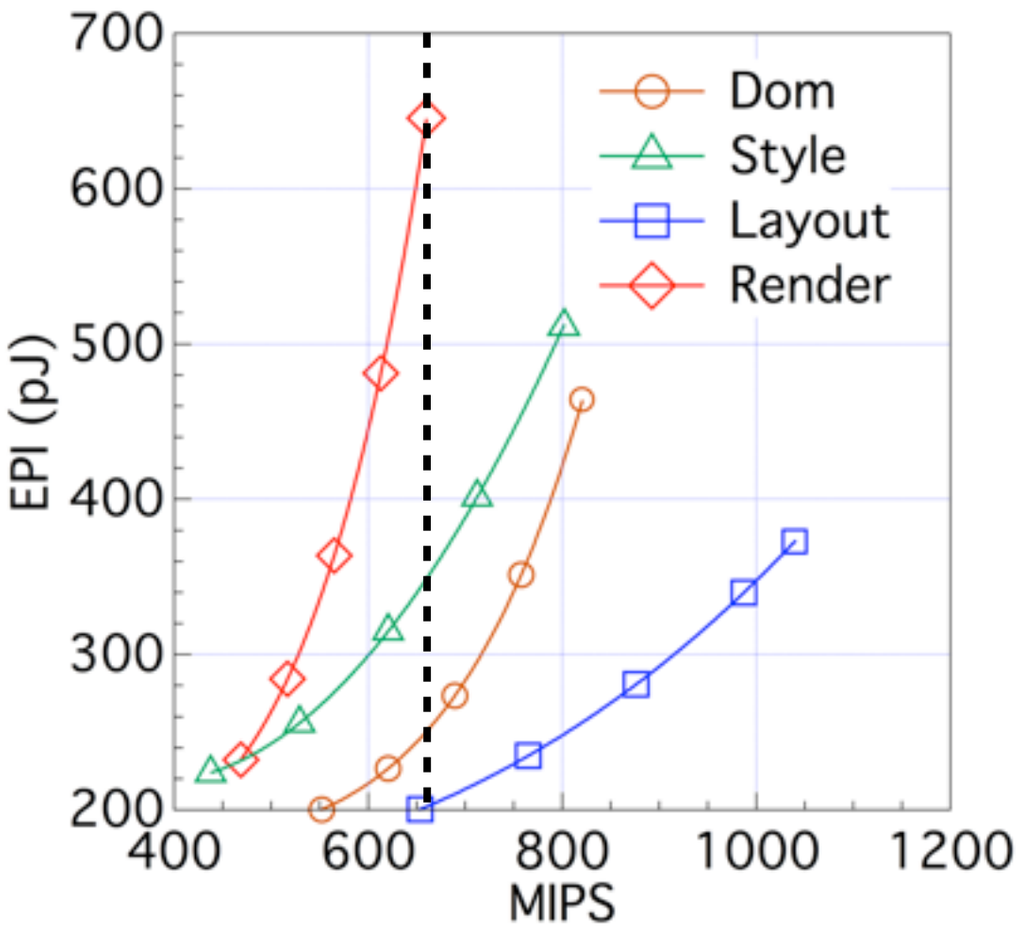
- In-order designs show strong kernel variance



In-order design

Understand the Difference Using Kernel Knowledge

- In-order designs show strong kernel variance

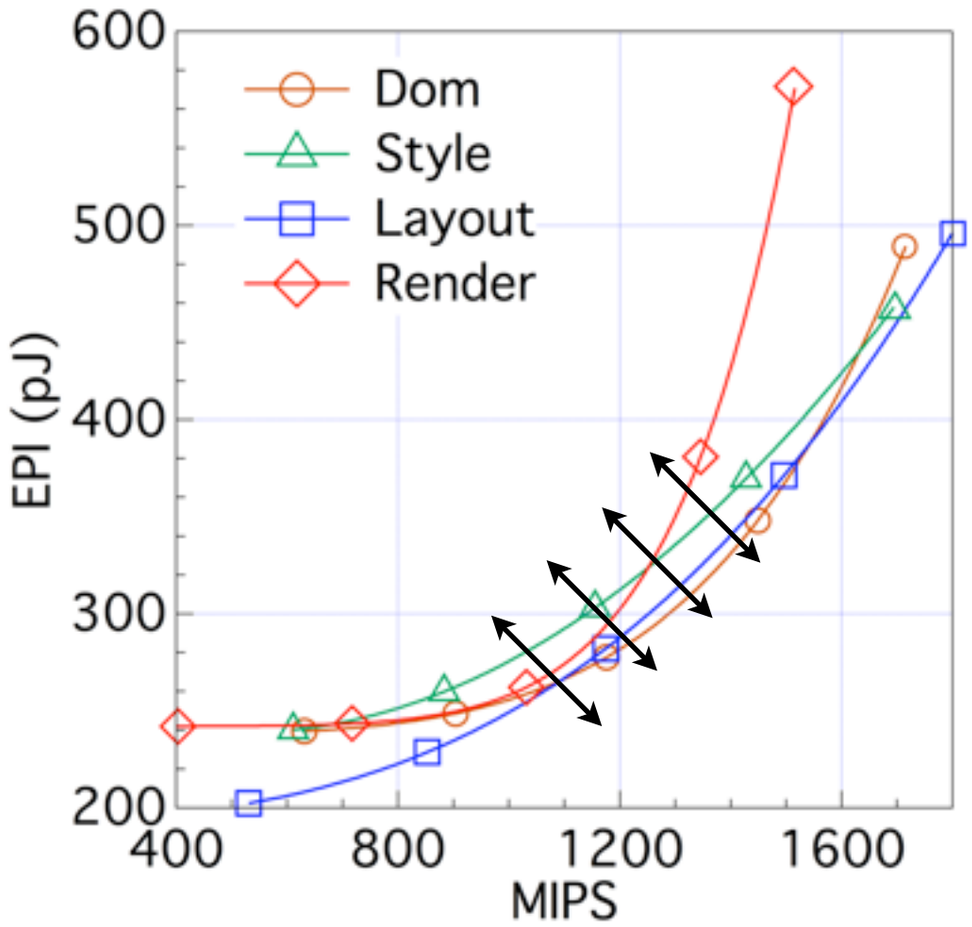


In-order design

Understand the Difference Using Kernel Knowledge

- In-order designs show strong kernel variance

In-order design

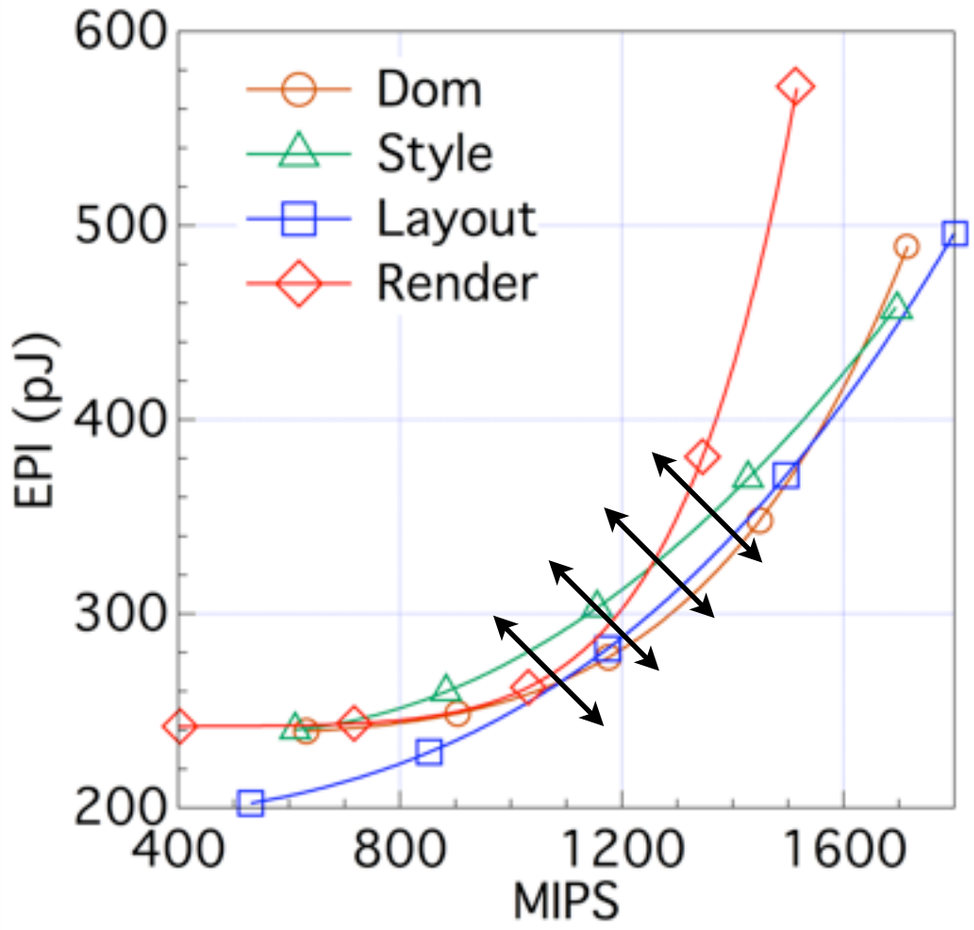


Out-of-order design

Understand the Difference Using Kernel Knowledge

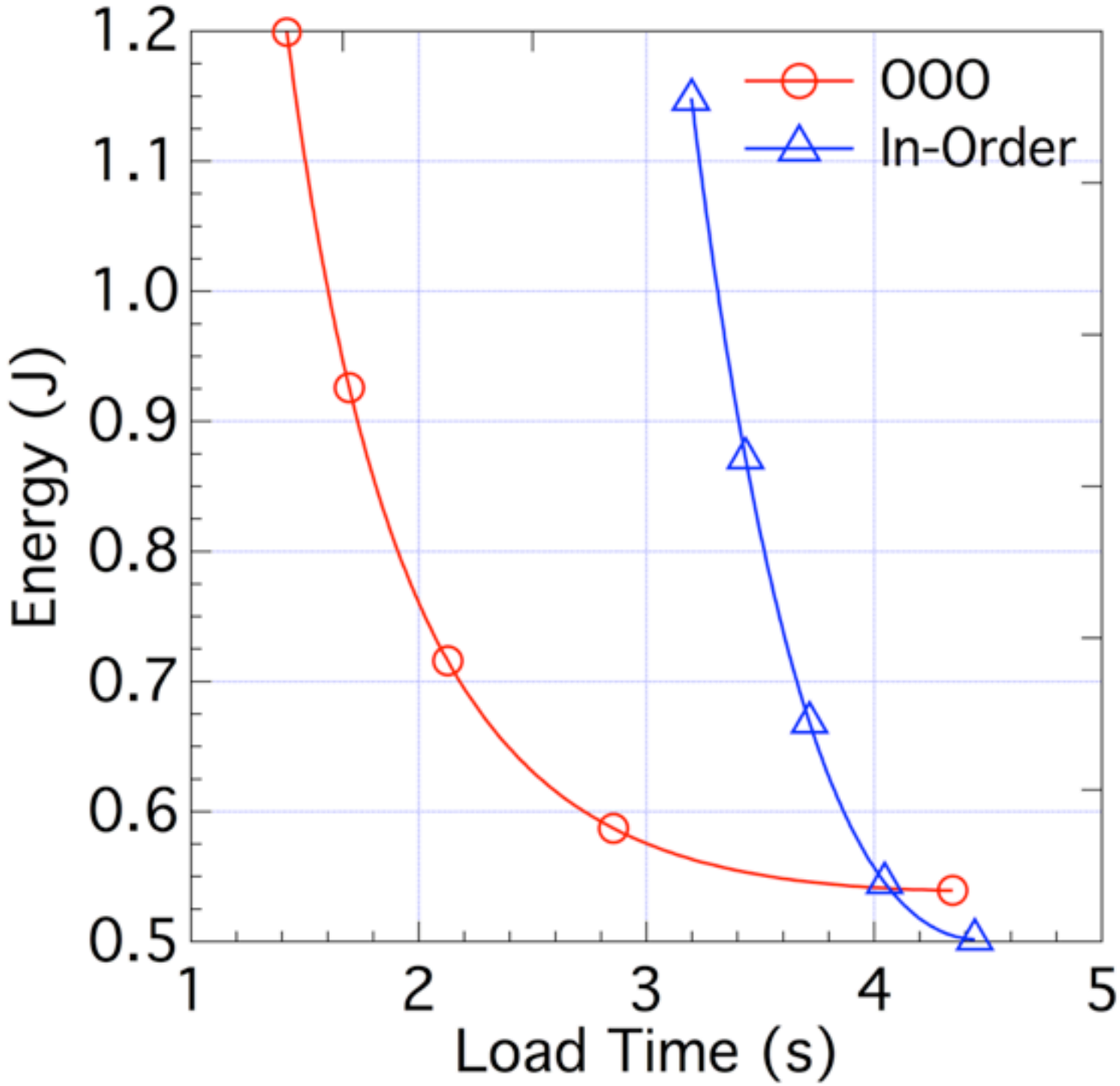
- In-order designs show strong kernel variance
- An Out-of-order design can accommodate kernel variance

In-order design

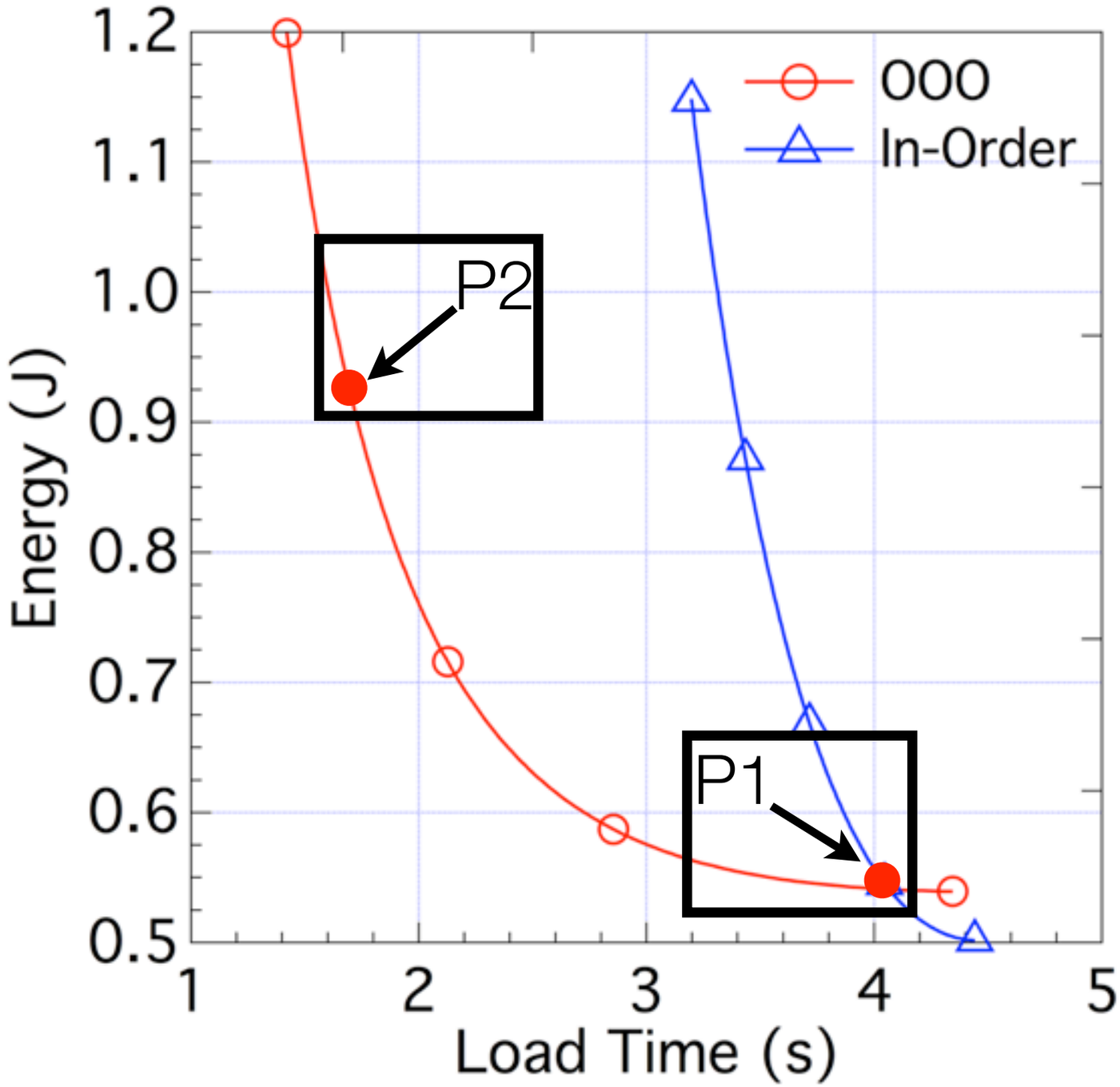


Out-of-order design

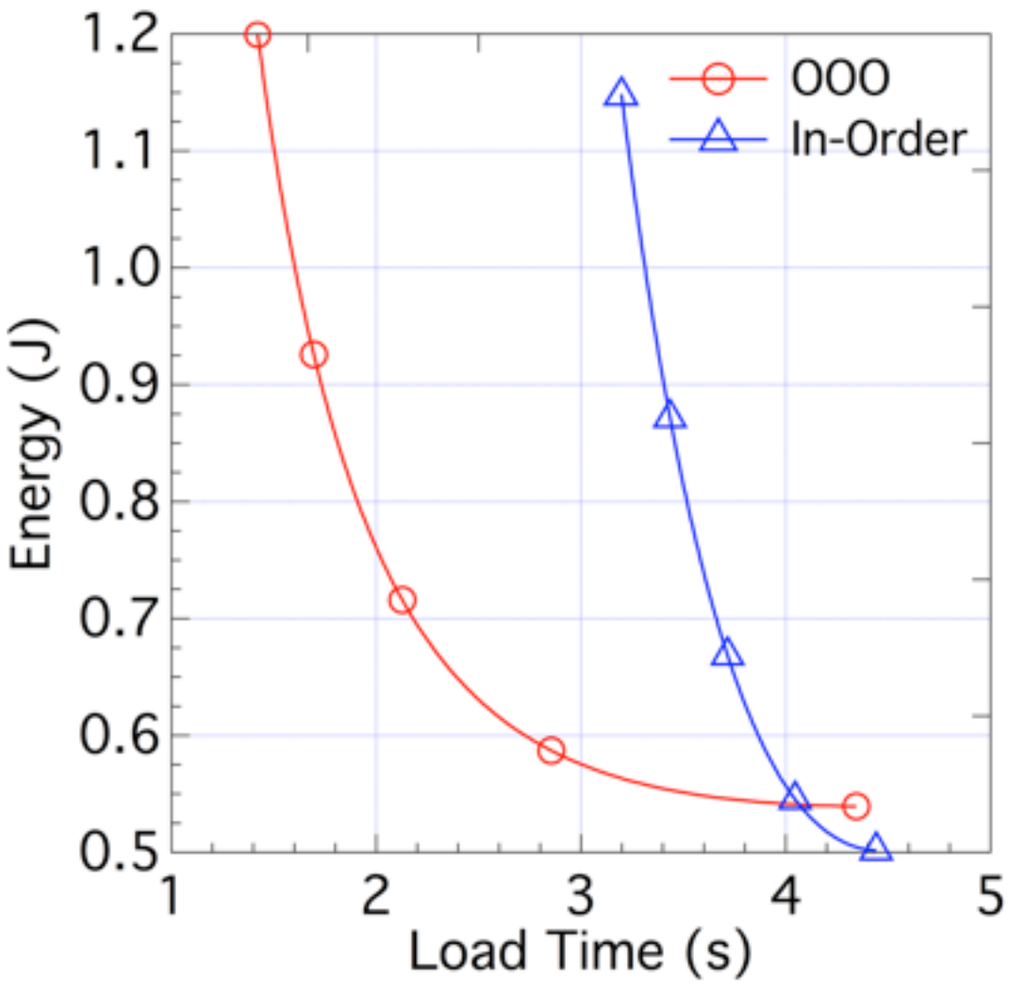
Customization: Identifying Major Sources of Energy Inefficiency



Customization: Identifying Major Sources of Energy Inefficiency

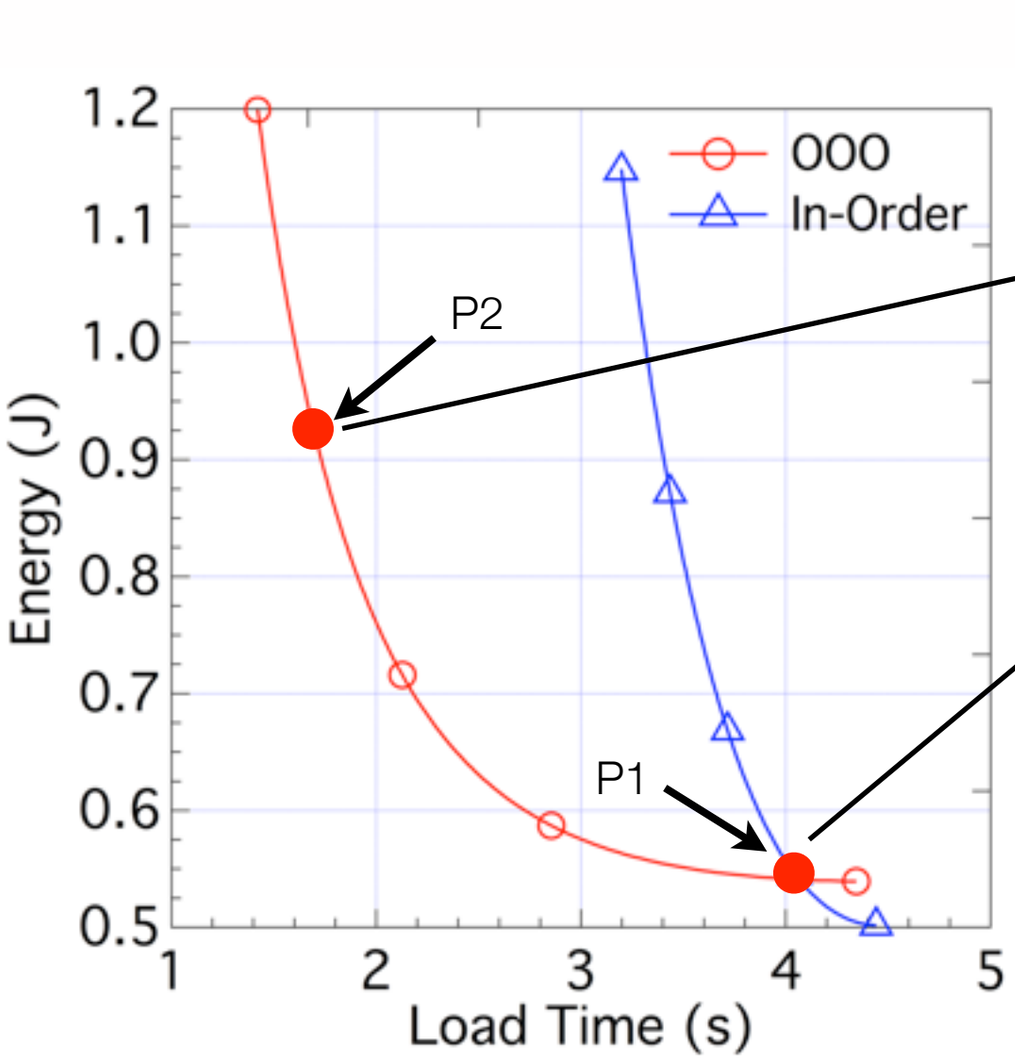


Customization: Identifying Major Sources of Energy Inefficiency



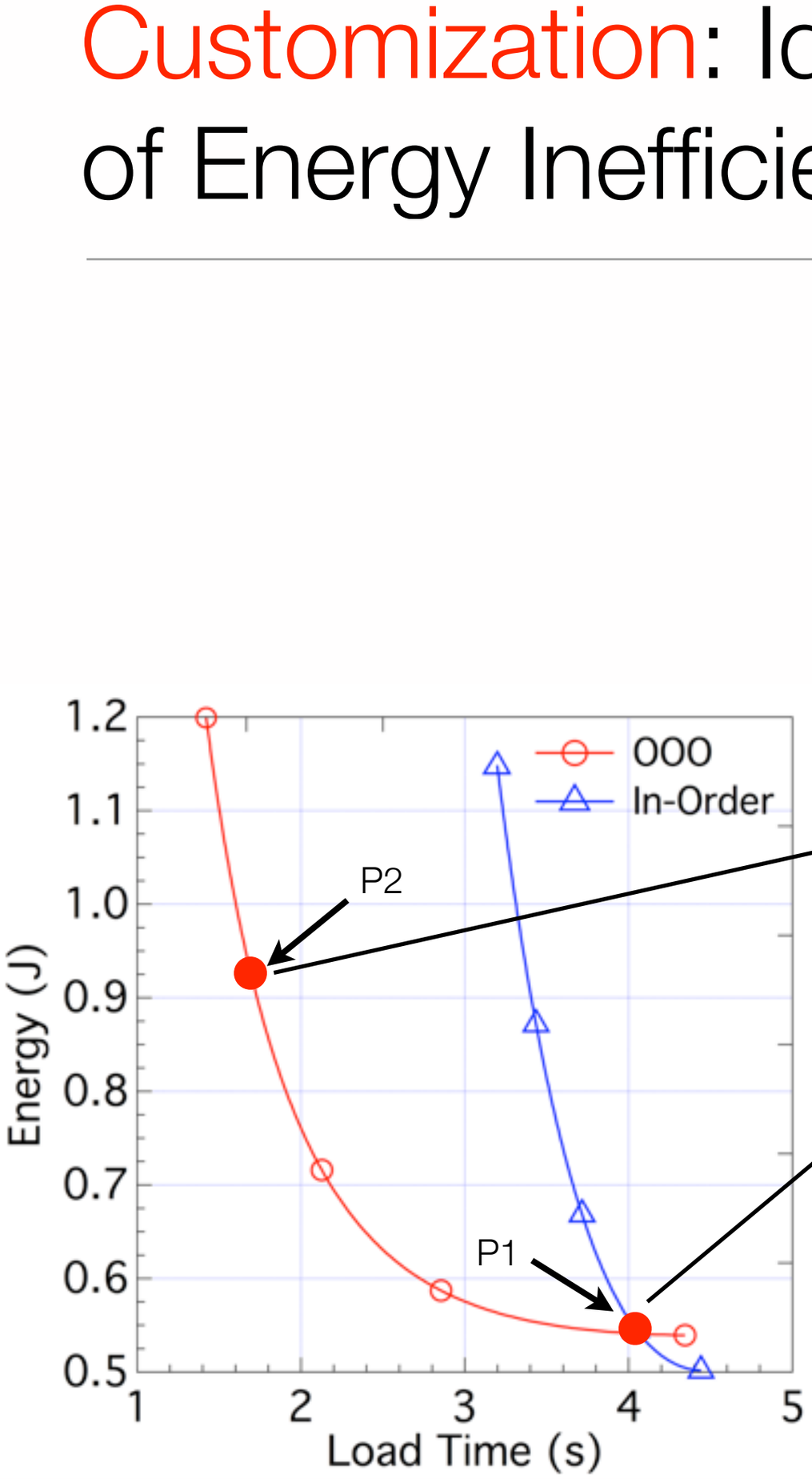
	P1	P2
Issue width	1	3
# Function units	2	3
Load queue size	4	16
Store queue size	4	16
BTB size	1024	128
ROB size	128	128
L1 I-$ size (KB)	64	128
# Physical	128	140
L1 D-$ size (KB)	8	64
L2-$ size (KB)	256	1024

Customization: Identifying Major Sources of Energy Inefficiency



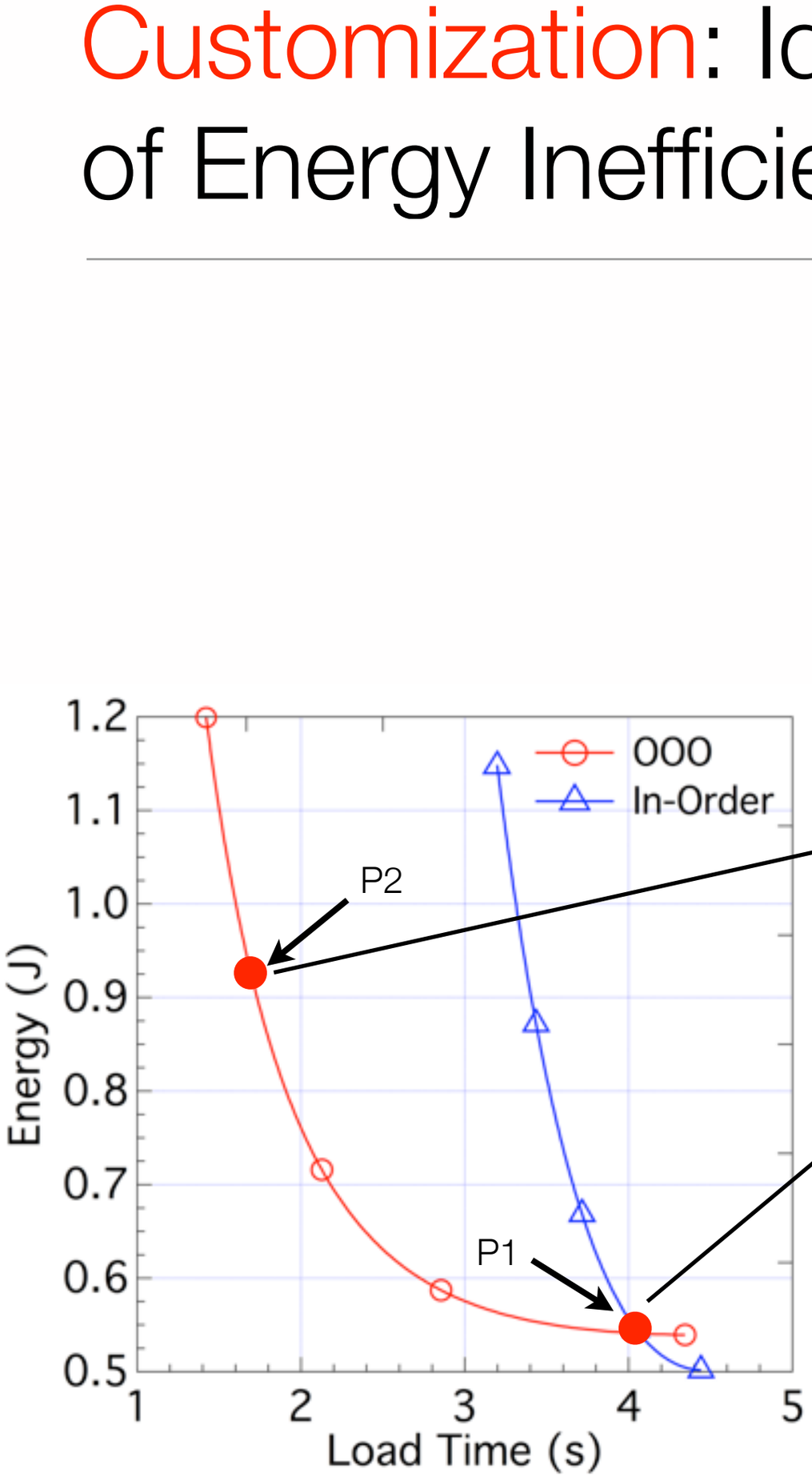
	P1	P2
Issue width	1	3
# Function units	2	3
Load queue size	4	16
Store queue size	4	16
BTB size	1024	128
ROB size	128	128
L1 I-$ size (KB)	64	128
# Physical	128	140
L1 D-$ size (KB)	8	64
L2-$ size (KB)	256	1024

Customization: Identifying Major Sources of Energy Inefficiency



	P1	P2	ARM A15
Issue width	1	3	3
# Function units	2	3	8
Load queue size	4	16	16
Store queue size	4	16	
BTB size	1024	128	256
ROB size	128	128	40+
L1 I-$ size (KB)	64	128	32
# Physical	128	140	?
L1 D-$ size (KB)	8	64	32
L2-$ size (KB)	256	1024	<4096

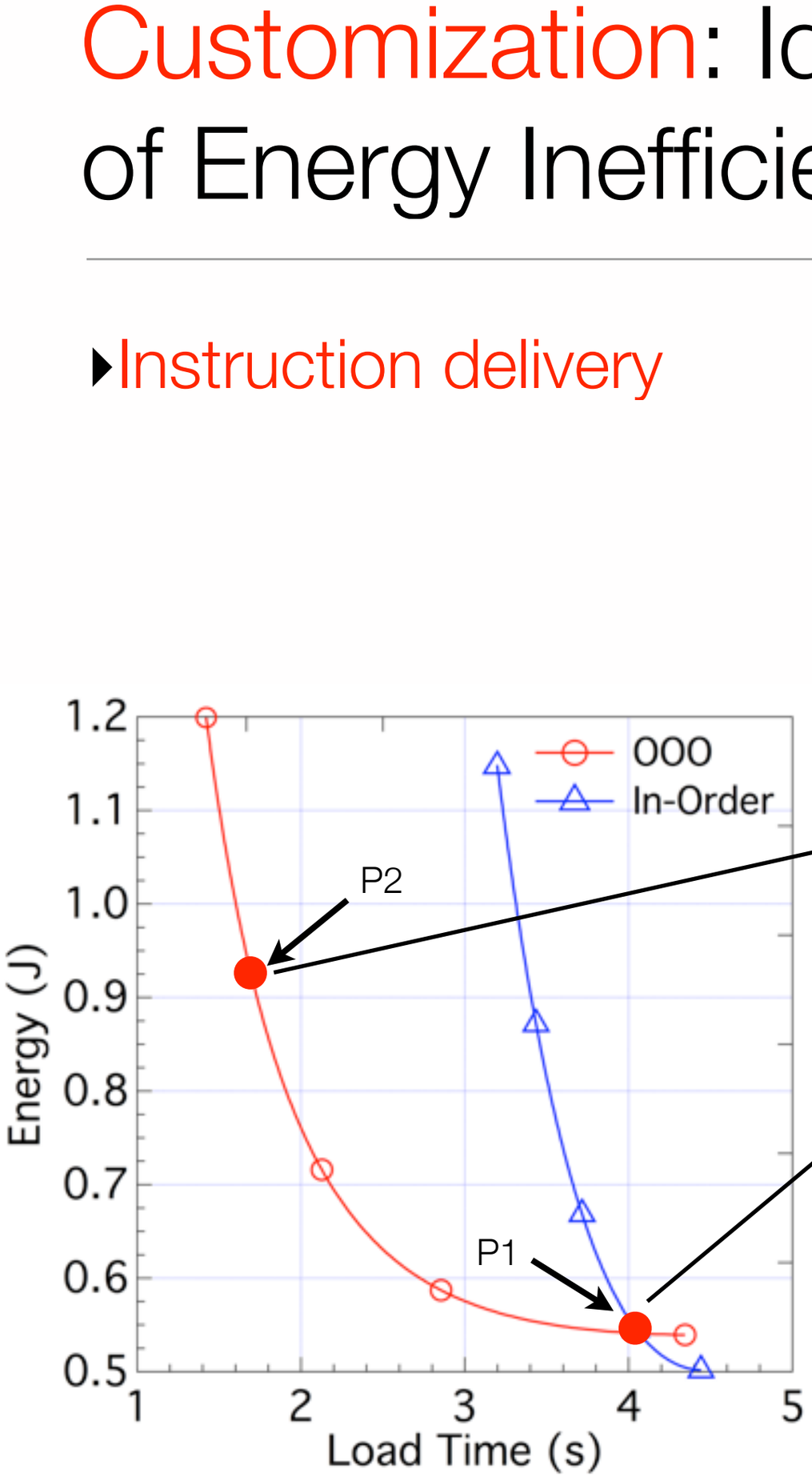
Customization: Identifying Major Sources of Energy Inefficiency



	P1	P2	ARM A15
Issue width	1	3	3
# Function units	2	3	8
Load queue size	4	16	16
Store queue size	4	16	
BTB size	1024	128	256
ROB size	128	128	40+
L1 I-$ size (KB)	64	128	32
# Physical	128	140	?
L1 D-$ size (KB)	8	64	32
L2-$ size (KB)	256	1024	<4096

Customization: Identifying Major Sources of Energy Inefficiency

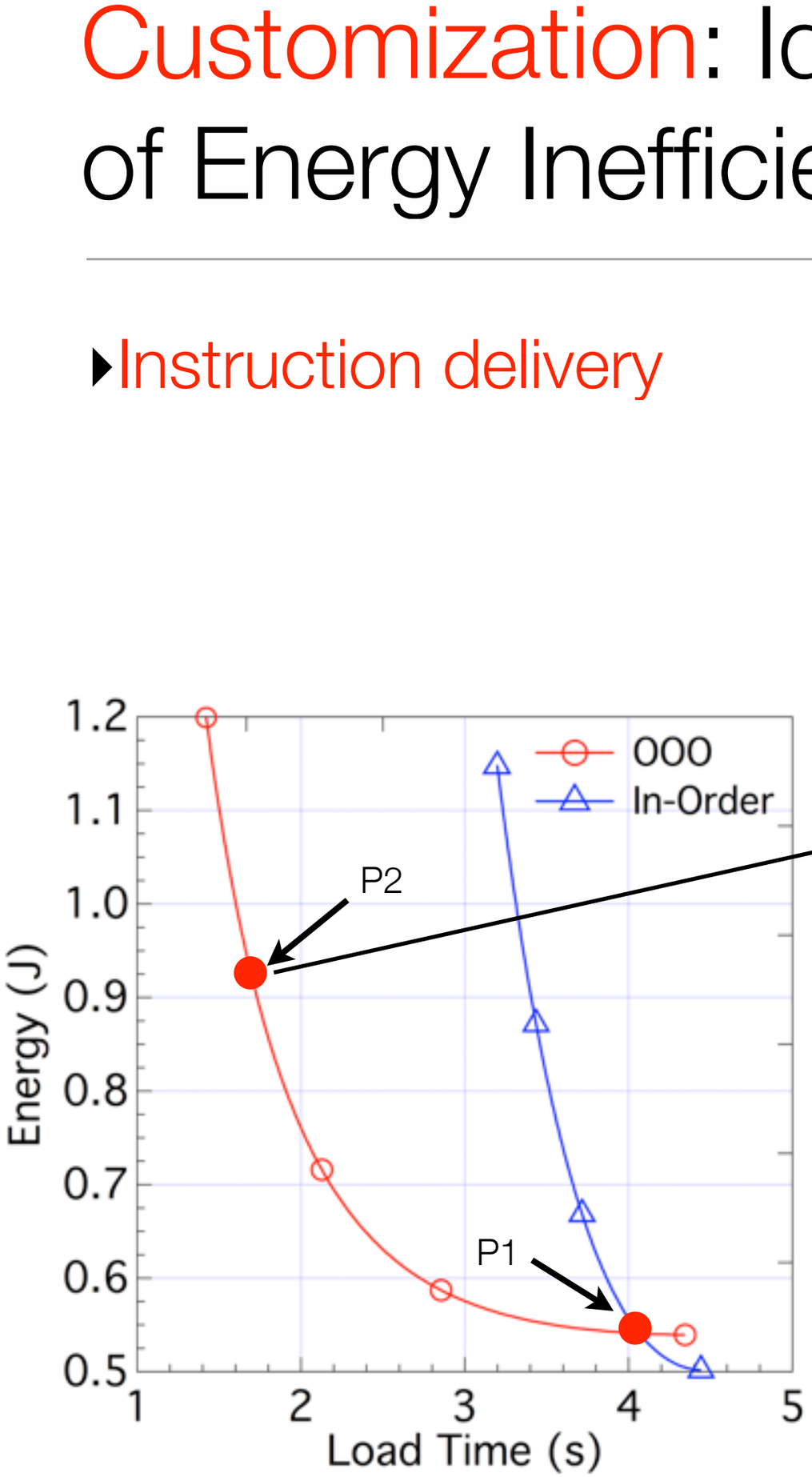
- Instruction delivery



	P1	P2	ARM A15
Issue width	1	3	3
# Function units	2	3	8
Load queue size	4	16	16
Store queue size	4	16	
BTB size	1024	128	256
ROB size	128	128	40+
L1 I-$ size (KB)	64	128	32
# Physical	128	140	?
L1 D-$ size (KB)	8	64	32
L2-$ size (KB)	256	1024	<4096

Customization: Identifying Major Sources of Energy Inefficiency

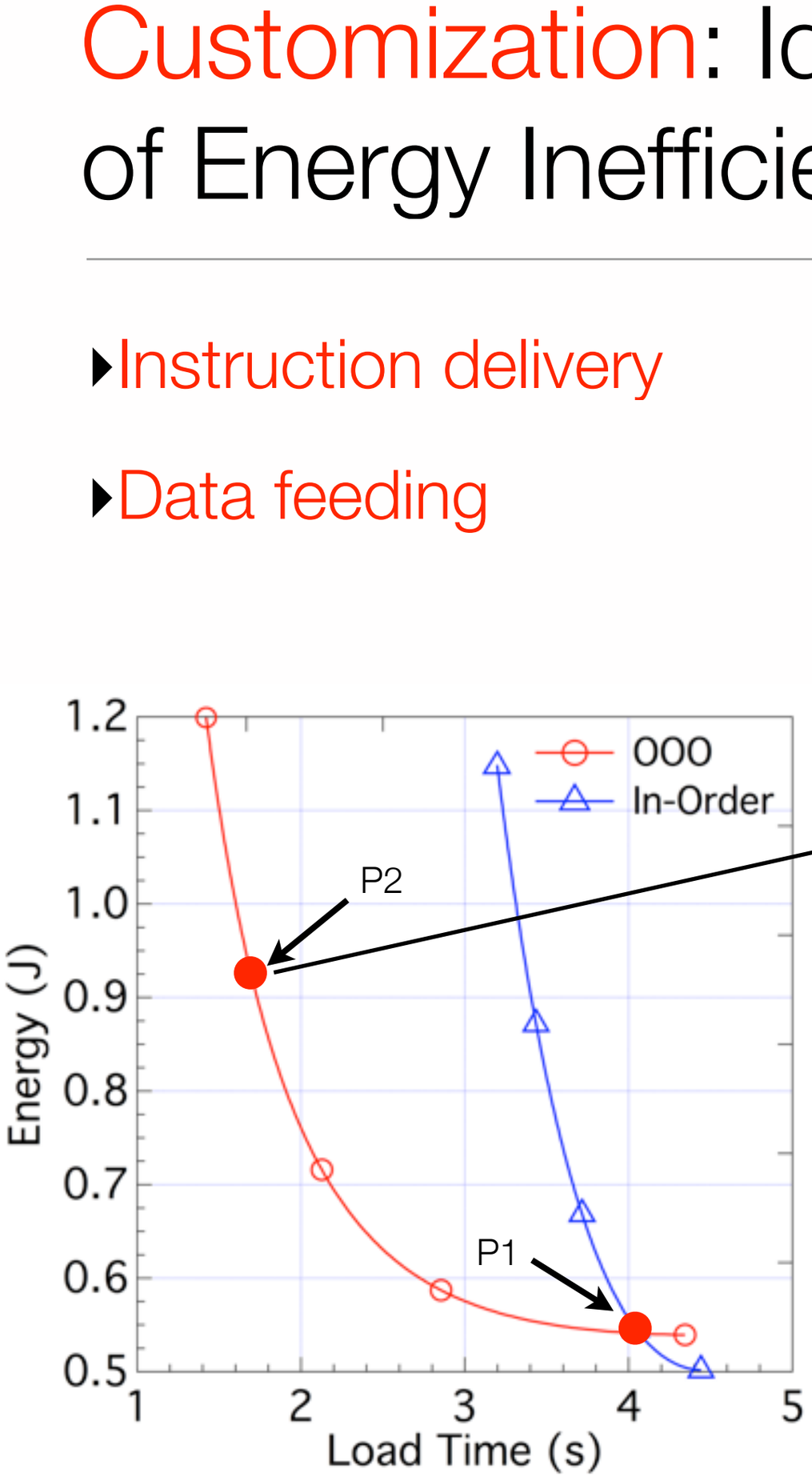
- Instruction delivery



	P1	P2	ARM A15
Issue width	1	3	3
# Function units	2	3	8
Load queue size	4	16	16
Store queue size	4	16	
BTB size	1024	128	256
ROB size	128	128	40+
L1 I-$ size (KB)	64	128	32
# Physical	128	140	?
L1 D-$ size (KB)	8	64	32
L2-$ size (KB)	256	1024	<4096

Customization: Identifying Major Sources of Energy Inefficiency

- Instruction delivery
- Data feeding



	P1	P2	ARM A15
Issue width	1	3	3
# Function units	2	3	8
Load queue size	4	16	16
Store queue size	4	16	
BTB size	1024	128	256
ROB size	128	128	40+
L1 I-$ size (KB)	64	128	32
# Physical	128	140	?
L1 D-$ size (KB)	8	64	32
L2-$ size (KB)	256	1024	<4096

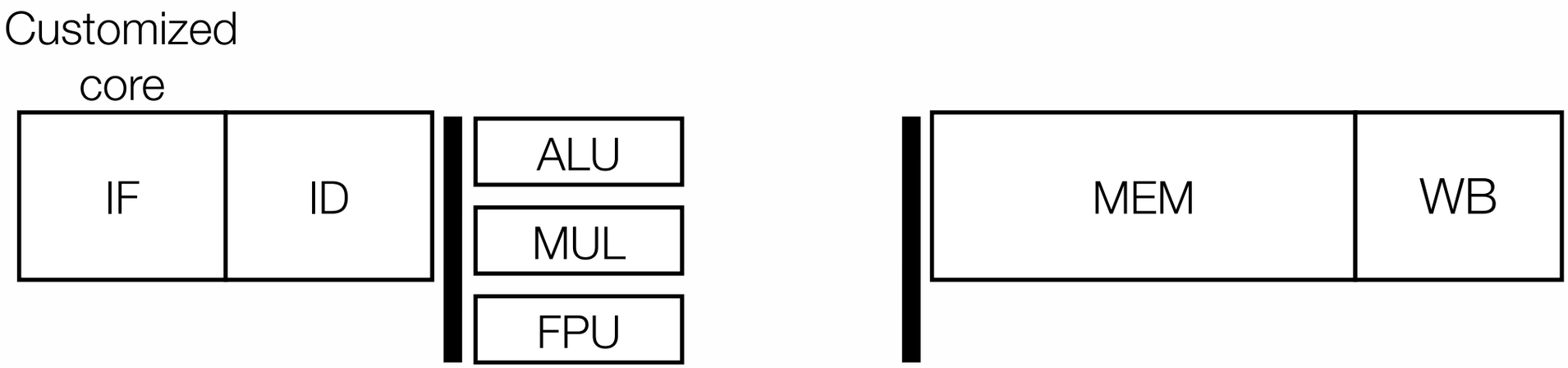
Agenda of Today's Talk

- Motivation of our work: energy-efficiency of the mobile Web
- How does WebCore improve the energy-efficiency?
  - Customization
  - **Specialization**
    - Mitigate instruction delivery: Style resolution unit (SRU)
    - Improving data feeding: Browser engine cache
- Evaluation Results
- Related Work

WebCore Specialization Overview

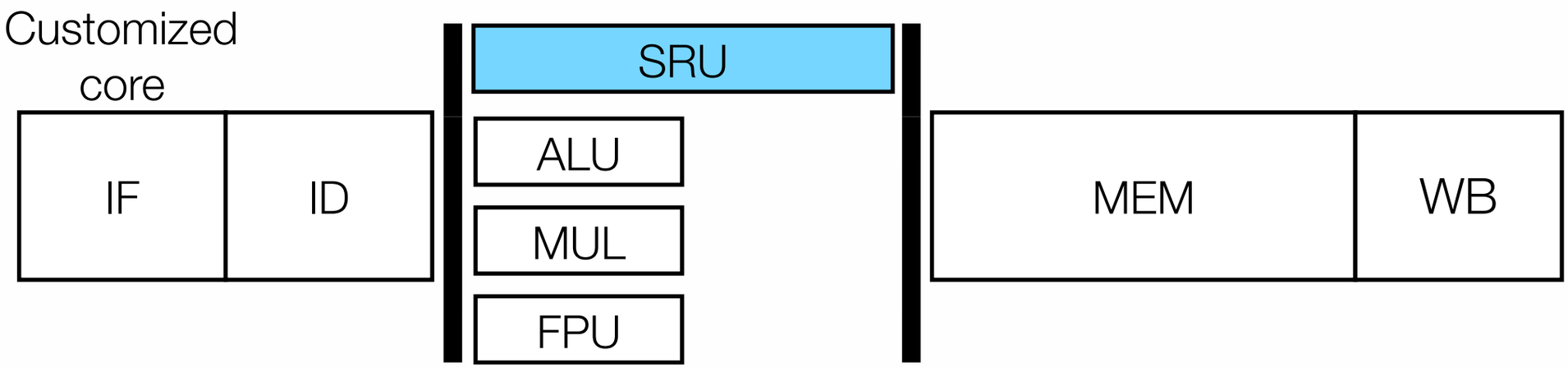
WebCore Specialization Overview

Hardware Layer



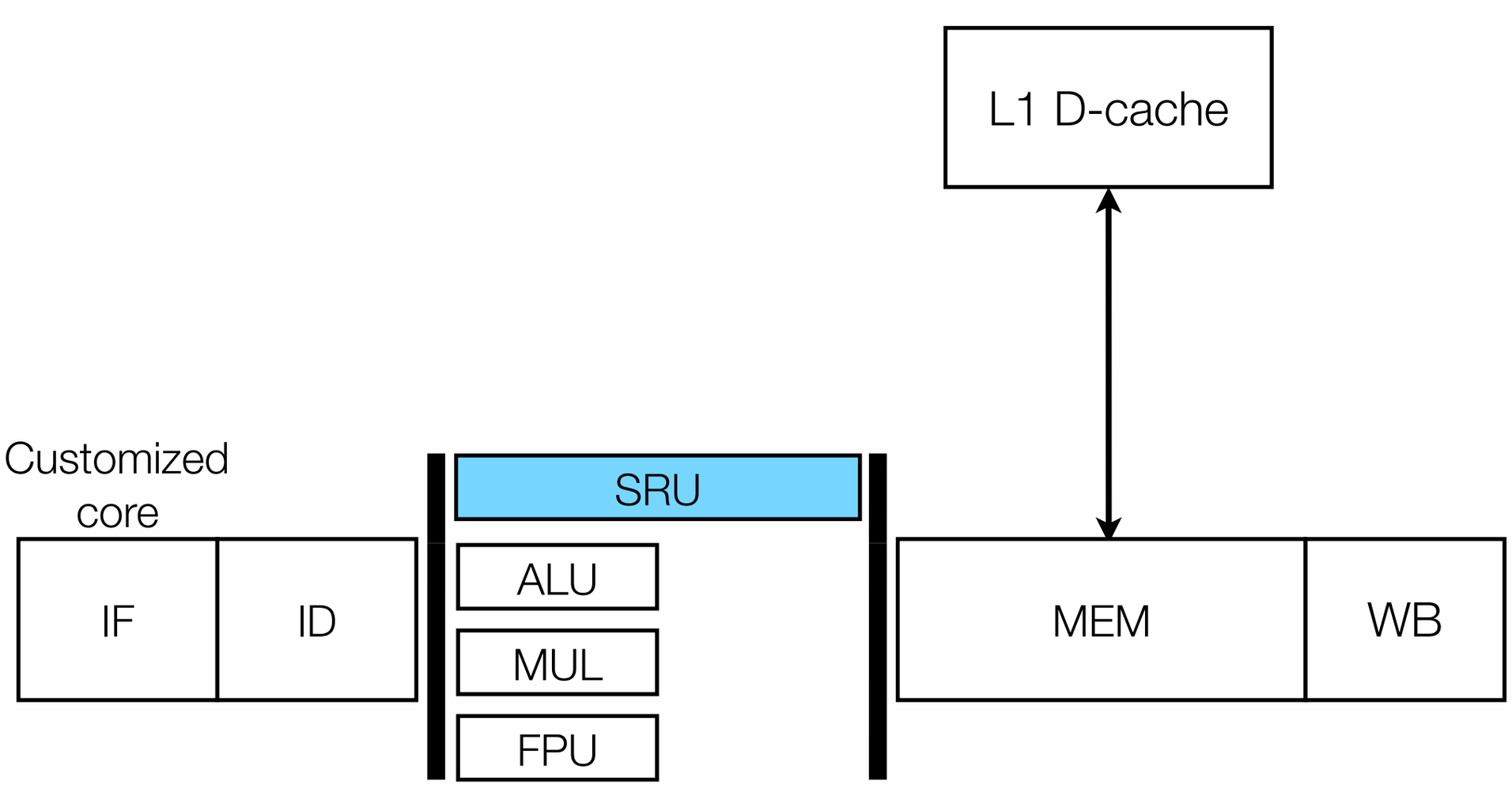
WebCore Specialization Overview

Hardware Layer



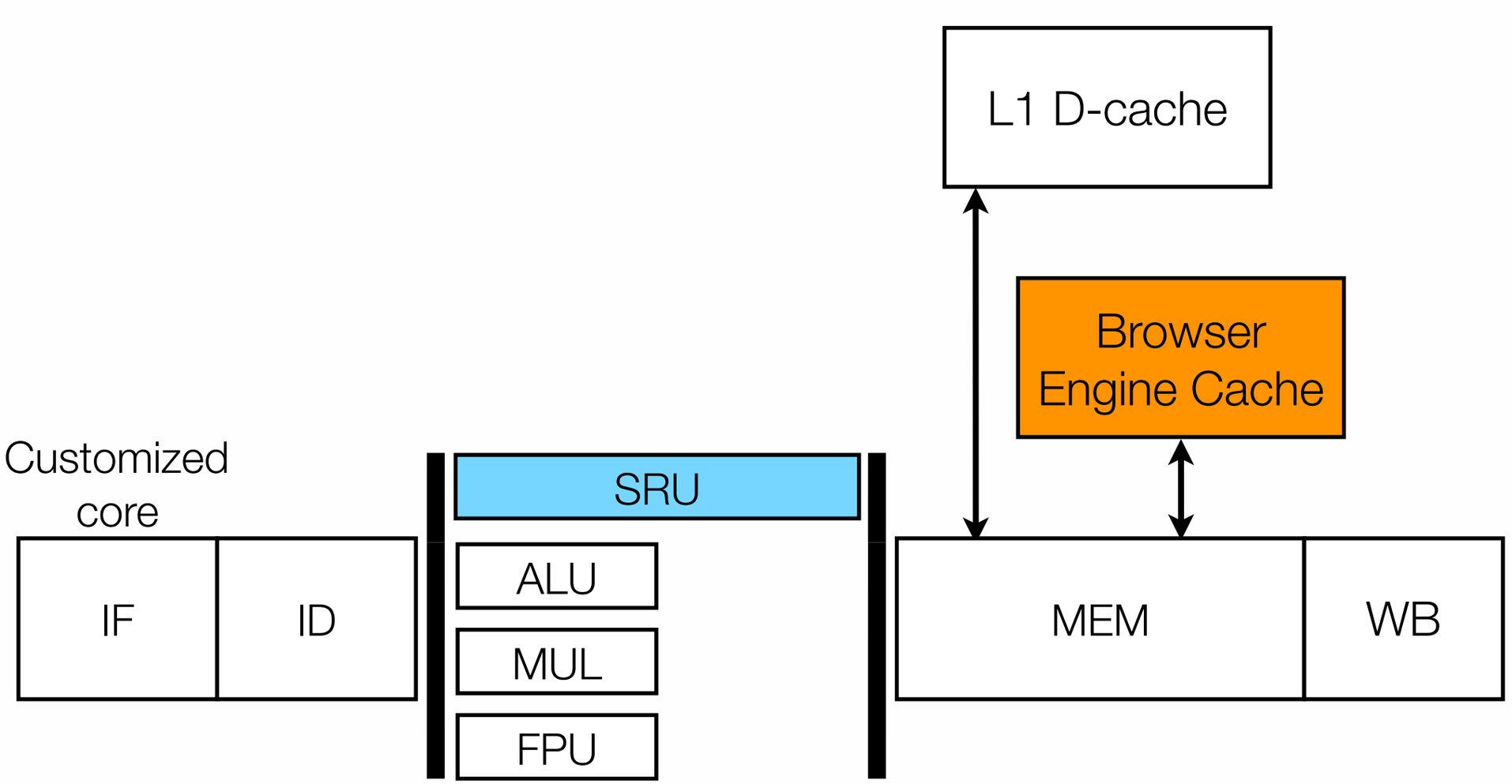
WebCore Specialization Overview

Hardware Layer



WebCore Specialization Overview

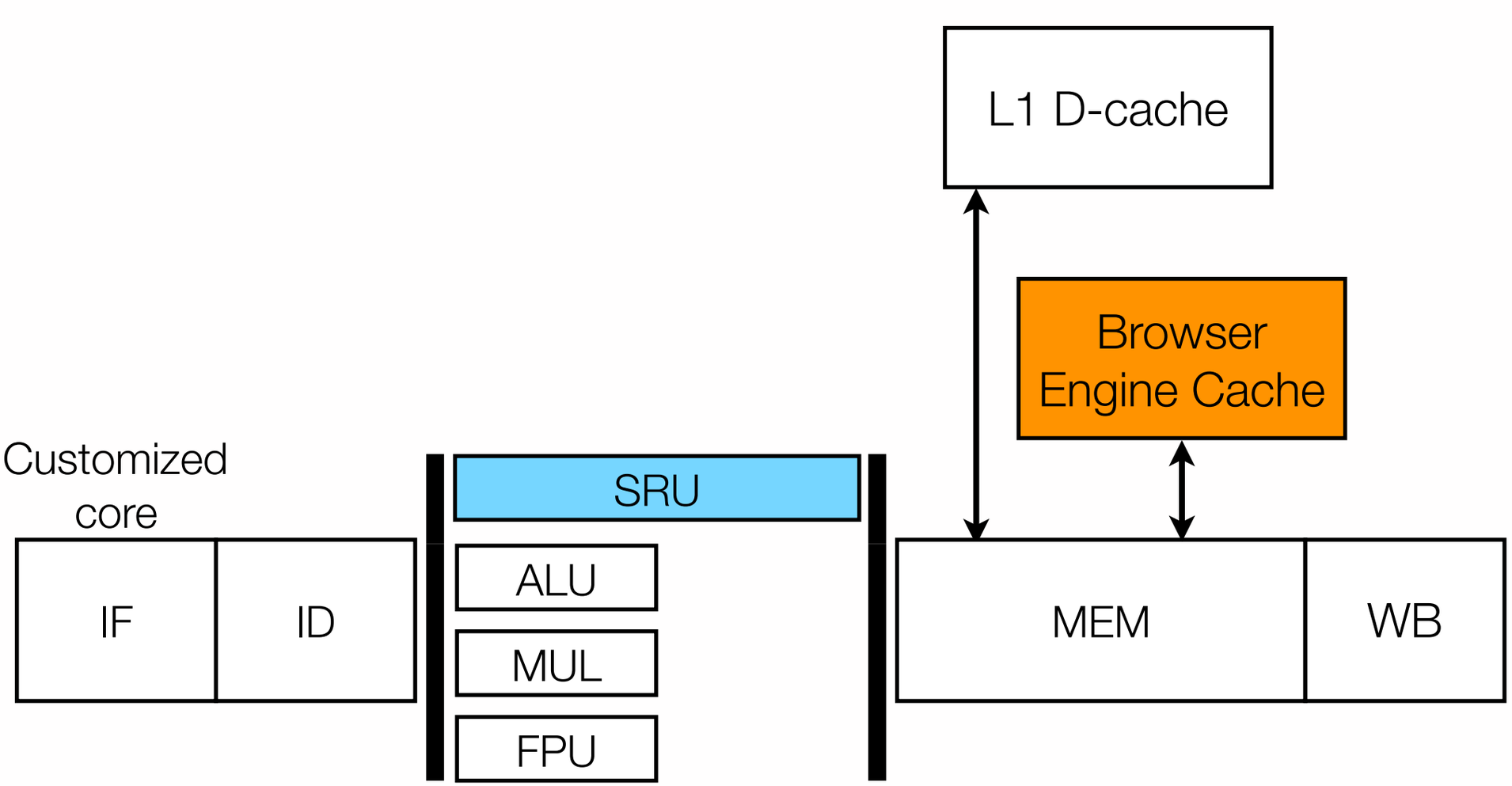
Hardware Layer



WebCore Specialization Overview

API Layer

Hardware Layer

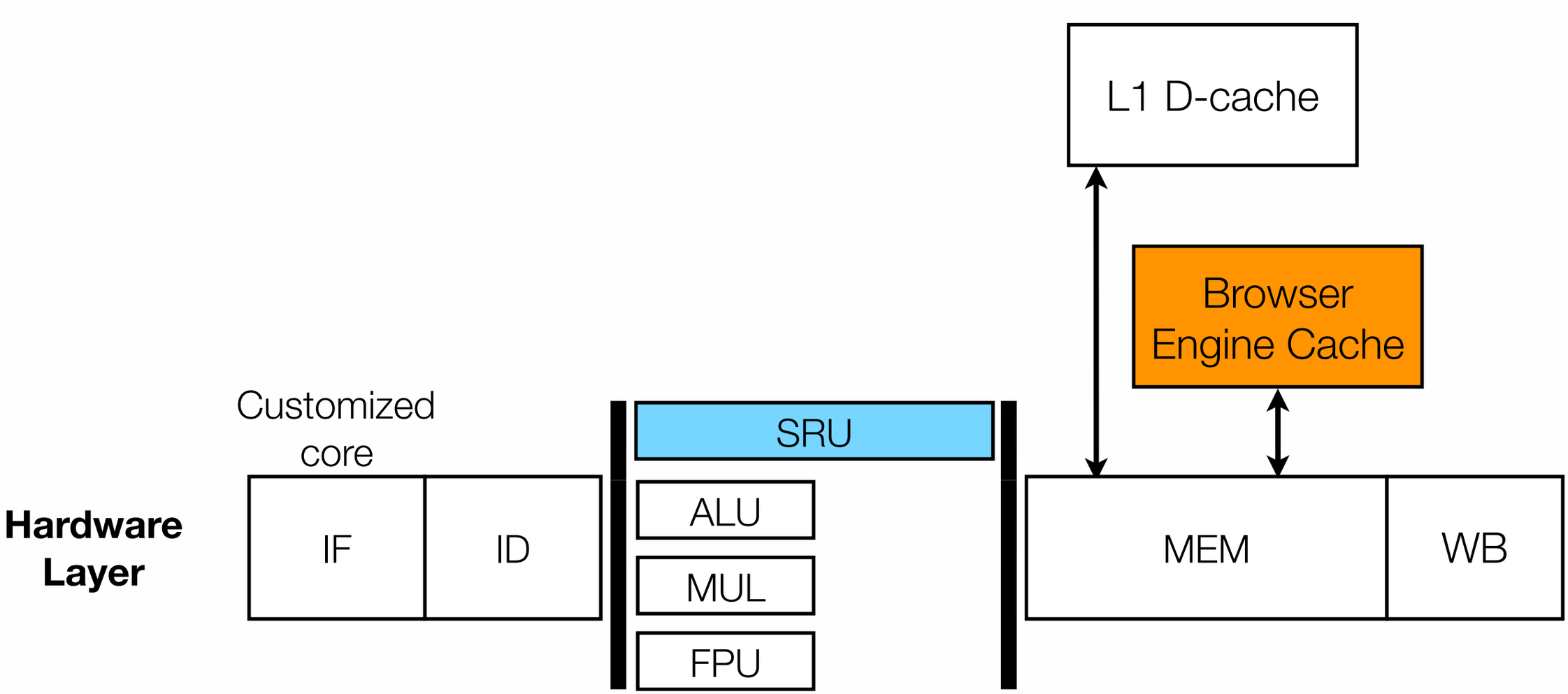


WebCore Specialization Overview

API Layer

```
Style_apply(Id);
```

Hardware Layer

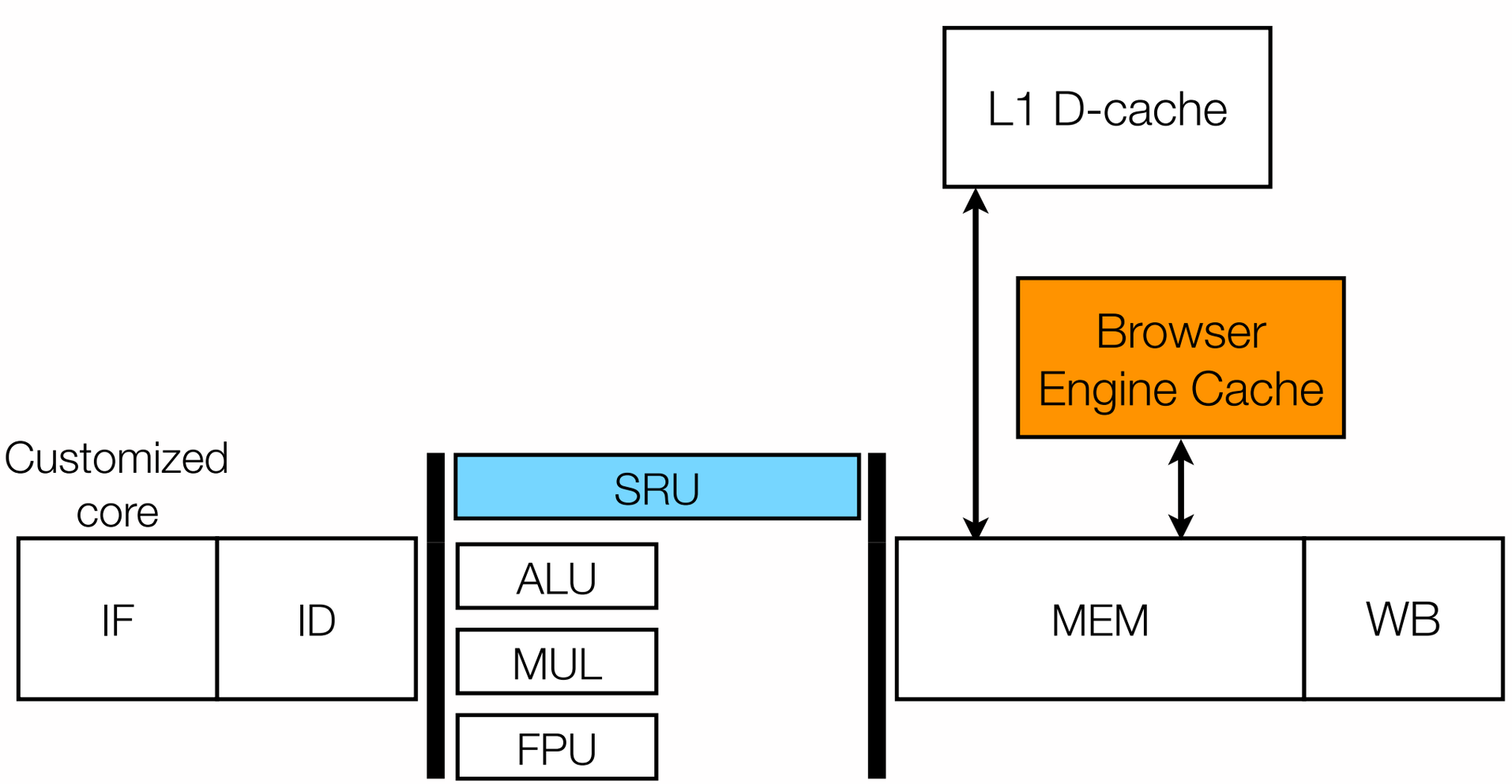


WebCore Specialization Overview

API Layer

```
                                        DOM_LD(Id, &attr);
                                        DOM_ST(Id, &attr);
                    Style_apply(Id);
```

Hardware Layer

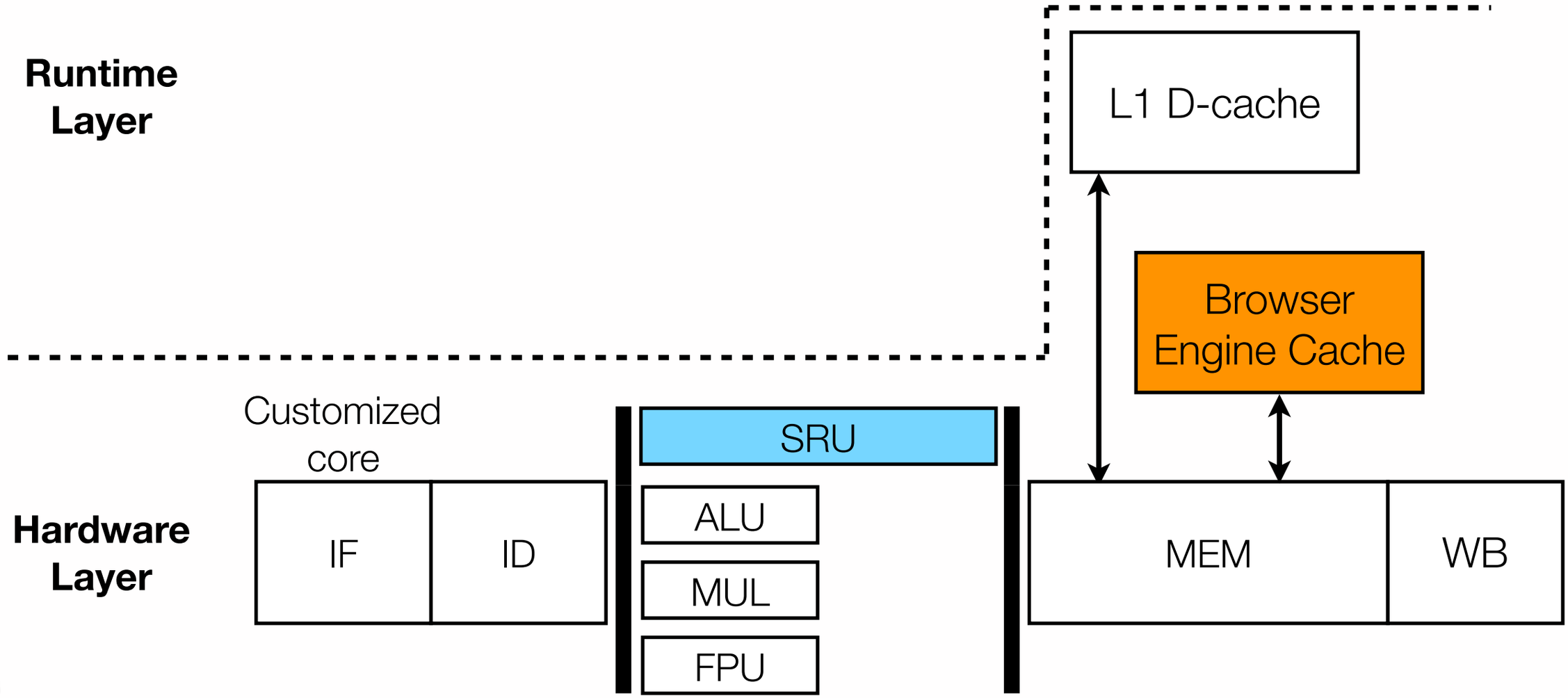


WebCore Specialization Overview

API Layer

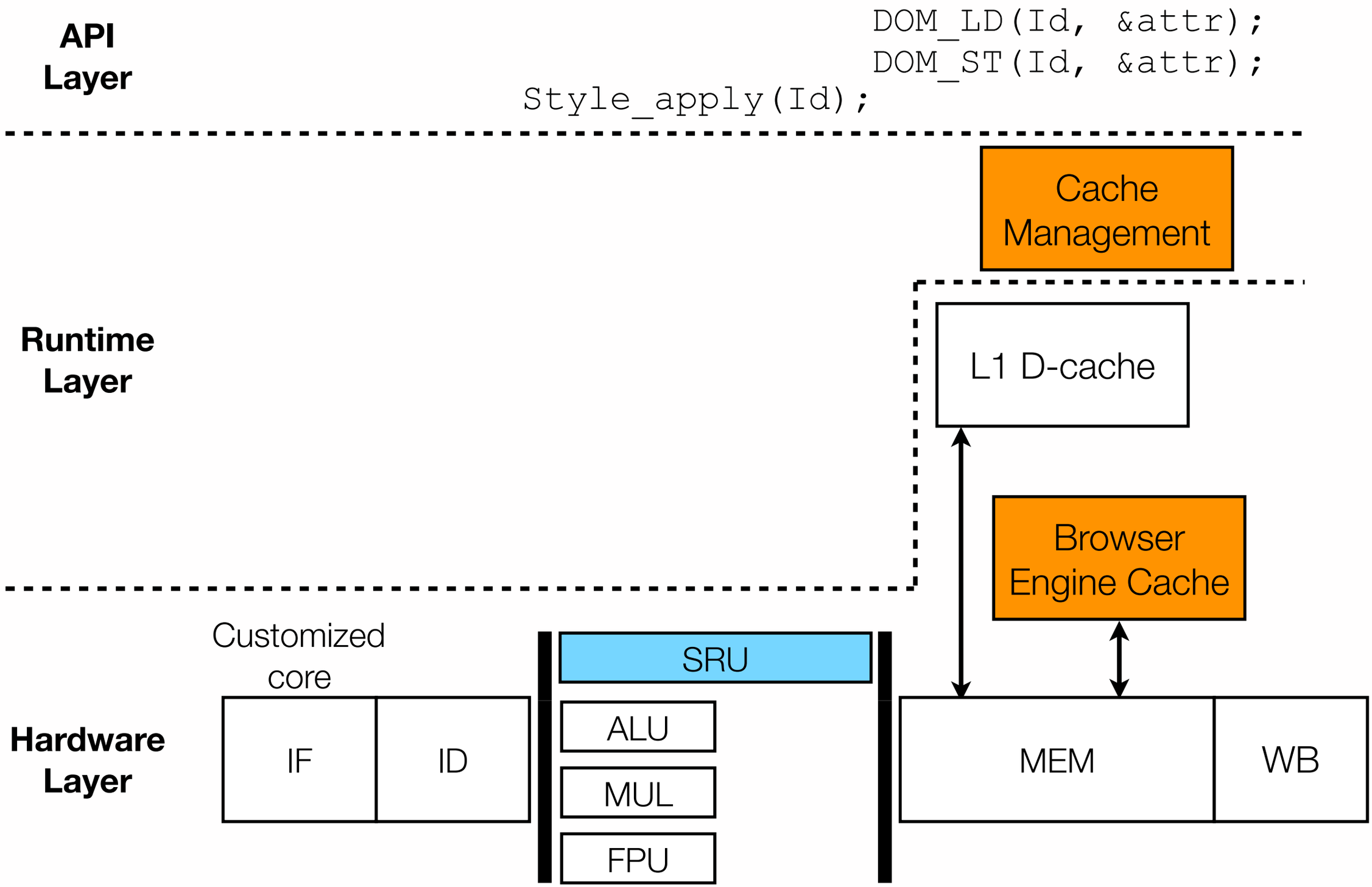
```
DOM_LD(Id, &attr);
DOM_ST(Id, &attr);
Style_apply(Id);
```

Runtime Layer

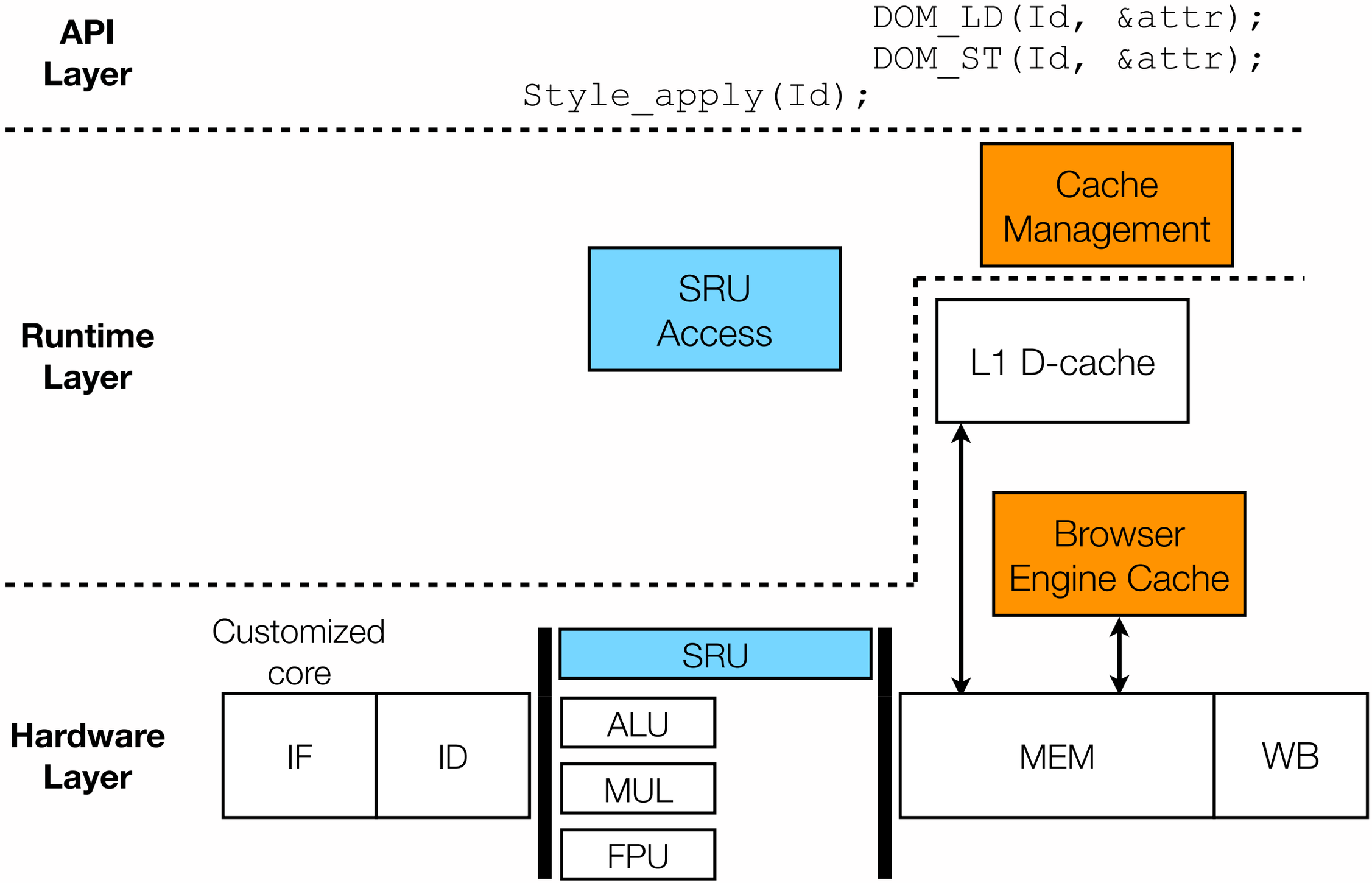


Hardware Layer

WebCore Specialization Overview



WebCore Specialization Overview



WebCore Specialization Overview

API Layer

```
DOM_LD(Id, &attr);
DOM_ST(Id, &attr);
Style_apply(Id);
```

Runtime Layer

Cache Management

SRU Access

Software Failsafe

L1 D-cache

Browser Engine Cache

Hardware Layer

Customized core

IF | ID

SRU

ALU

MUL

FPU

MEM | WB

Agenda of Today's Talk

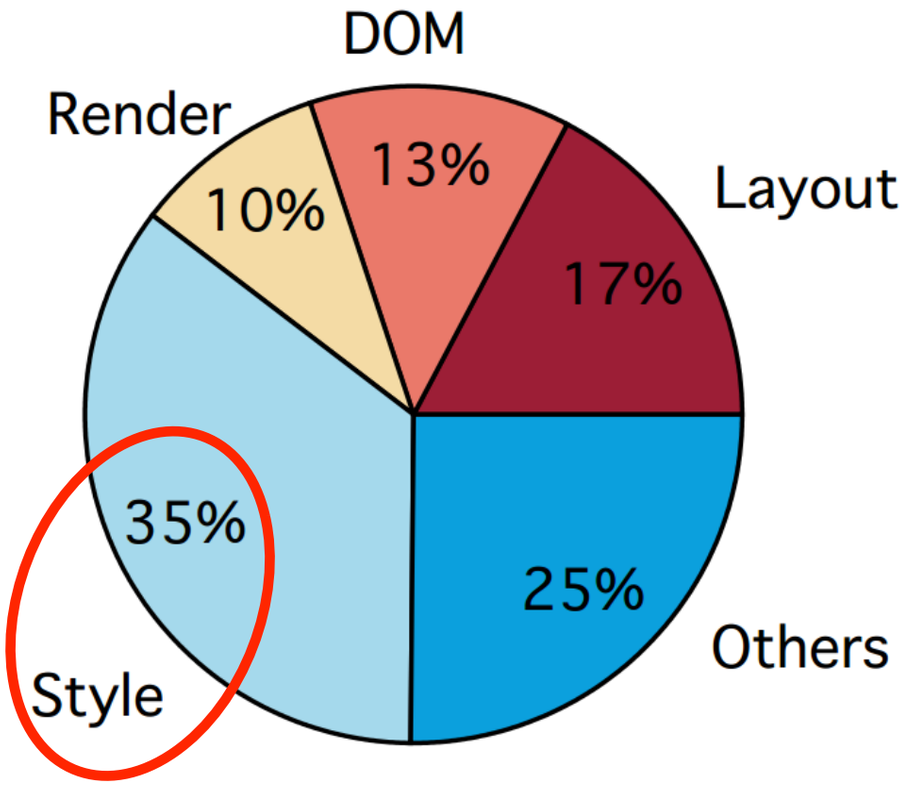
- Motivation of our work: energy-efficiency of the mobile Web
- How does WebCore improve the energy-efficiency?
  - Customization
  - **Specialization**
    - **Mitigate instruction delivery: Style resolution unit (SRU)**
    - Improving data feeding: Browser engine cache
- Evaluation Results
- Related Work

Style Resolution Unit

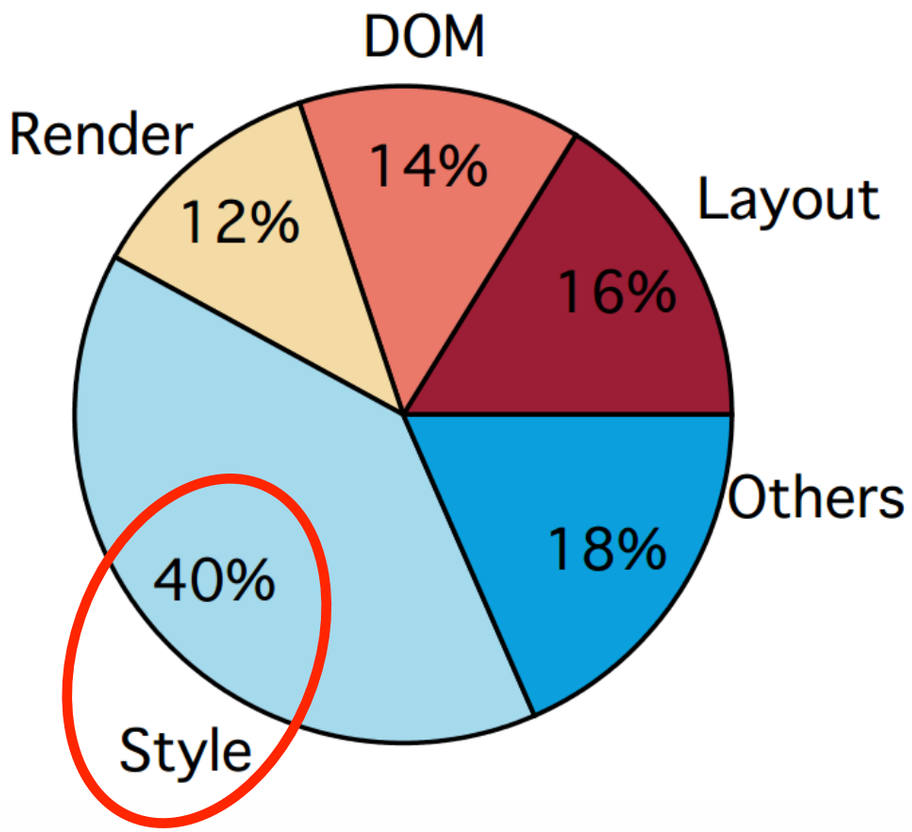
- *Style* kernel is the most critical kernel

Style Resolution Unit

- *Style* kernel is the most critical kernel



Execution time breakdown



Energy consumption breakdown

Style Resolution Unit

- *Style* kernel is the most critical kernel

```
for (each rule in matchedRules) {
  for (each property in rule) {
    switch (property.id) {
      case Font:
        Style[Font] = Handler(property.value, DOMNode);
        break;
      case N: ...}}}
```

Style Resolution Unit

- *Style* kernel is the most critical kernel

```
for (each rule in matchedRules) {
   for (each property in rule) {
      switch (property.id) {
       case Font:
         Style[Font] = Handler(property.value, DOMNode);
         break;
       case N: ...}}}
```

Style Resolution Unit

- *Style* kernel is the most critical kernel

```
for (each rule in matchedRules) {
  for (each property in rule) {
    switch (property.id) {
      case Font:
        Style[Font] = Handler(property.value, DOMNode);
        break;
      case N: ...}}}
```

Rule-level Parallelism (RLP)

Style Resolution Unit

- *Style* kernel is the most critical kernel

```
for (each rule in matchedRules) {
    for (each property in rule) {
      switch (property.id) {
        case Font:
          Style[Font] = Handler(property.value, DOMNode);
          break;
        case N: ...}}}
```

Rule-level Parallelism (RLP)

Style Resolution Unit

- *Style* kernel is the most critical kernel

```
for (each rule in matchedRules) {
    for (each property in rule) {
      switch (property.id) {
       case Font:
         Style[Font] = Handler(property.value, DOMNode);
         break;
       case N: ...}}}
```

Rule-level Parallelism (RLP)

Property-level Parallelism (PLP)

Style Resolution Unit

- *Style* kernel is the most critical kernel

```
for (each rule in matchedRules) {
  for (each property in rule) {
    switch (property.id) {
     case Font:
       Style[Font] = Handler(property.value, DOMNode);
       break;
     case N: ...}}}
```

Rule-level Parallelism (RLP)

Property-level Parallelism (PLP)

- Exploiting the parallelism to increase the arithmetic intensity and reduce instruction footprint

Style Resolution Unit (2)

- A running example from www.cnn.com

Style Rules

Rule	**Property** 1		**Property** 2	
	id	value	id	value
1	padding	0	margin	0
2	padding	6 px	width	36 px

Style Resolution Unit (2)

- A running example from www.cnn.com

Style Rules

High priority

Rule	**Property** 1		**Property** 2	
	id	value	id	value
1	padding	0	margin	0
2	padding	6 px	width	36 px

Style Resolution Unit (2)

- A running example from www.cnn.com

Style Rules

Final Style Info

High priority

Rule	Property 1		Property 2	
	id	value	id	value
1	padding	0	margin	0
2	padding	6 px	width	36 px

Property 1		Property 2		Property 3	
id	value	id	value	id	value

Style Resolution Unit (2)

- A running example from www.cnn.com

Style Rules

Final Style Info

High priority

Rule	Property 1		Property 2	
	id	value	id	value
1	padding	0	margin	0
2	padding	6 px	width	36 px

Property 1		Property 2		Property 3	
id	value	id	value	id	value

Style Resolution Unit (2)

- A running example from www.cnn.com

Style Rules

Final Style Info

High priority

Rule	Property 1		Property 2	
	id	value	id	value
1	padding	0	margin	0
2	padding	6 px	width	36 px

Property 1		Property 2		Property 3	
id	value	id	value	id	value
padding	0				

Style Resolution Unit (2)

- A running example from www.cnn.com

Style Rules

Final Style Info

High priority

Rule	Property 1		Property 2	
	id	value	id	value
1	padding	0	margin	0
2	padding	6 px	width	36 px

Property 1		Property 2		Property 3	
id	value	id	value	id	value
padding	0	margin	0		

Style Resolution Unit (2)

- A running example from www.cnn.com

Style Rules

Final Style Info

High priority

Rule	Property 1		Property 2	
	id	value	id	value
1	padding	0	margin	0
2	padding	6 px	width	36 px

Property 1		Property 2		Property 3	
id	value	id	value	id	value
padding	0	margin	0		

Style Resolution Unit (2)

- A running example from www.cnn.com

Style Rules

Final Style Info

High priority

Rule	Property 1		Property 2	
	id	value	id	value
1	padding	0	margin	0
2	padding	6 px	width	36 px

Property 1		Property 2		Property 3	
id	value	id	value	id	value
padding	6 px	margin	0		

Style Resolution Unit (2)

- Order Matters in RLP

- A running example from www.cnn.com

Style Rules

Final Style Info

High priority

Rule	Property 1		Property 2	
	id	value	id	value
1	padding	0	margin	0
2	padding	6 px	width	36 px

Property 1		Property 2		Property 3	
id	value	id	value	id	value
padding	6 px	margin	0		

Style Resolution Unit (2)

- Order Matters in RLP

- A running example from www.cnn.com

Style Rules

Final Style Info

High priority

Rule	Property 1		Property 2	
	id	value	id	value
1	padding	0	margin	0
2	padding	6 px	width	36 px

Property 1		Property 2		Property 3	
id	value	id	value	id	value
padding	6 px	margin	0	width	36 px

Style Resolution Unit (2)

- Order Matters in RLP
- Order Does **Not** Matter in PLP

- A running example from www.cnn.com

Style Rules

Final Style Info

High priority

Rule	**Property** 1		**Property** 2	
	id	value	id	value
1	padding	0	margin	0
2	padding	6 px	width	36 px

Property 1		**Property** 2		**Property** 3	
id	value	id	value	id	value
padding	6 px	margin	0	width	36 px

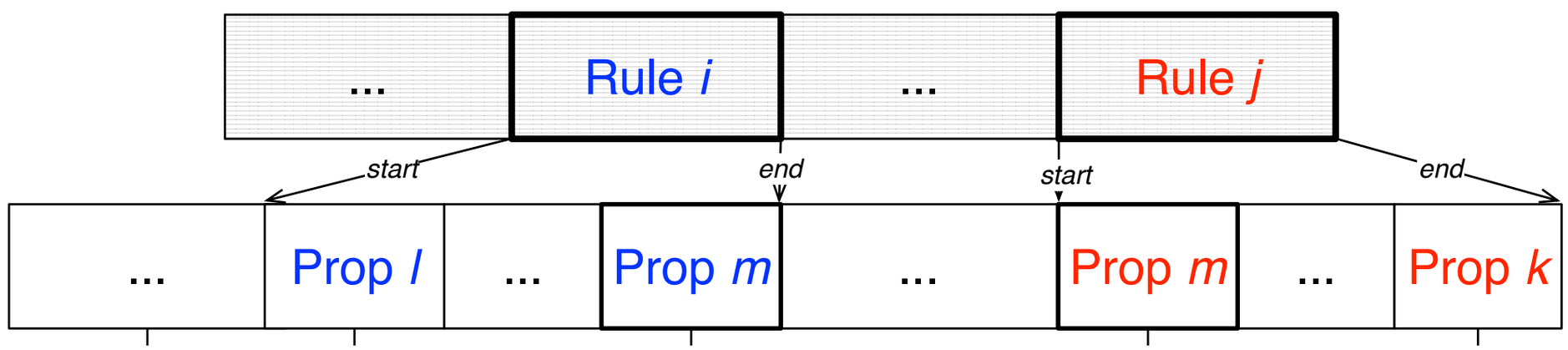
Style Resolution Unit (3)

- Order Matters in RLP
- Order Does **Not** Matter in PLP

Style Resolution Unit (3)

- Order Matters in RLP
- Order Does **Not** Matter in PLP

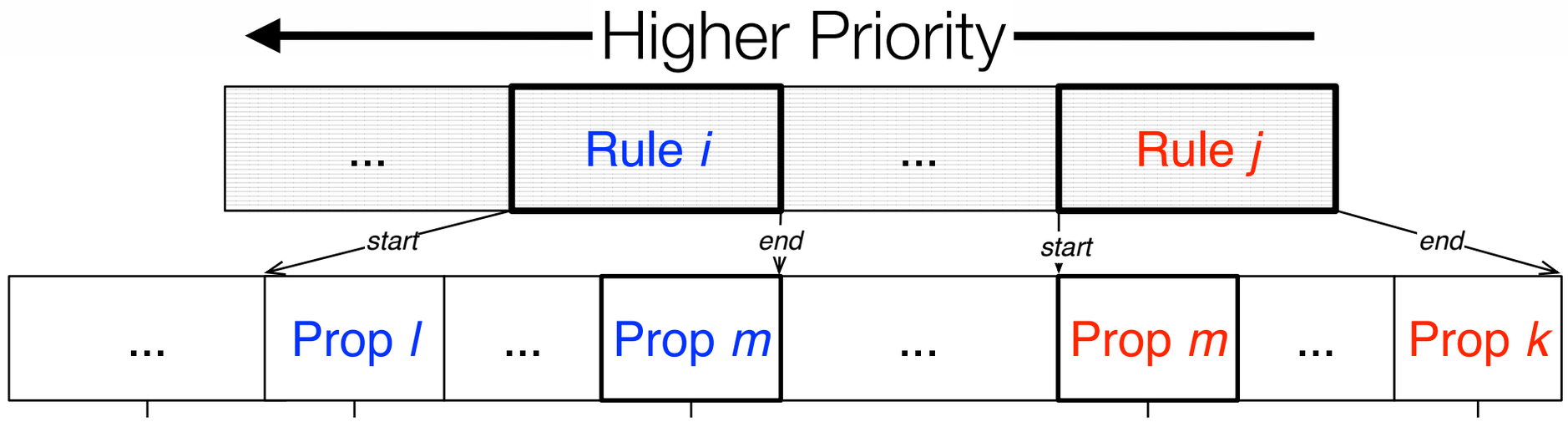
Input Scratchpad Memory



Style Resolution Unit (3)

- Order Matters in RLP
- Order Does **Not** Matter in PLP

Input Scratchpad Memory

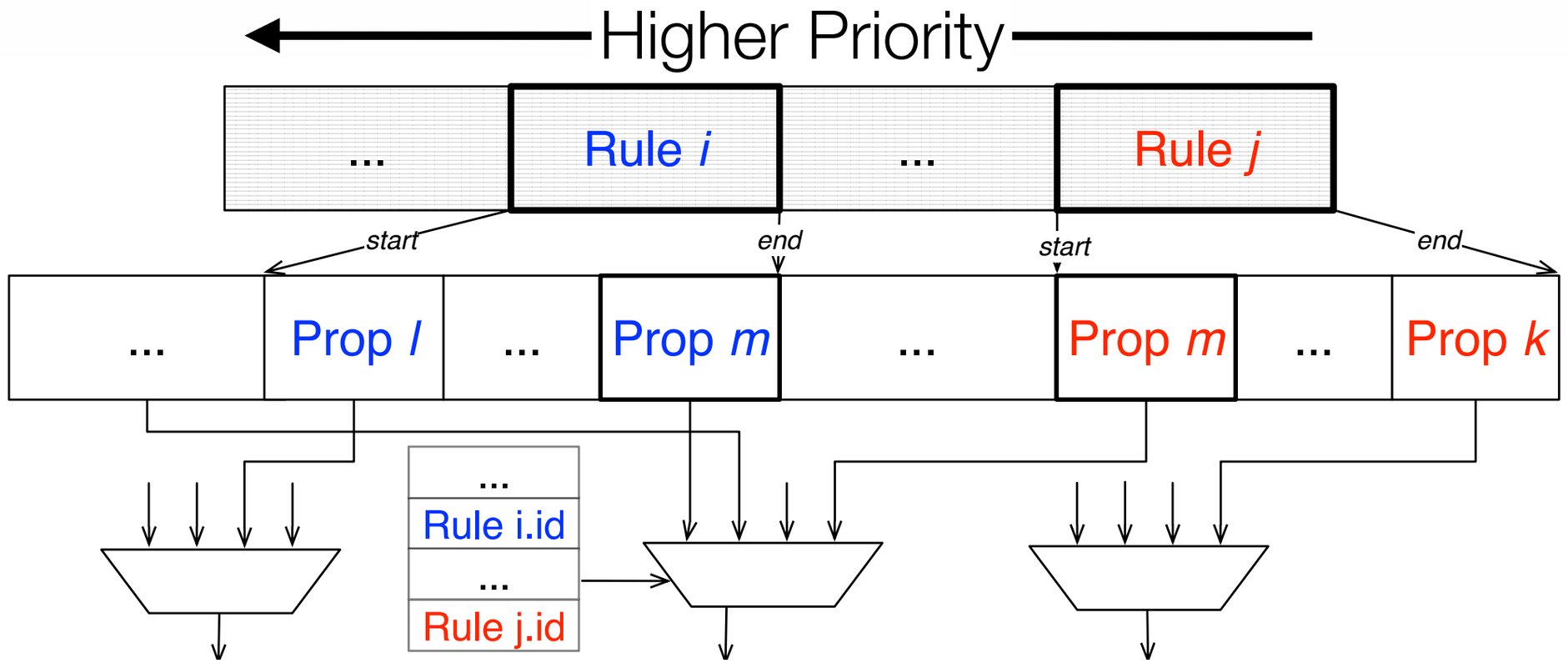


Style Resolution Unit (3)

- Order Matters in RLP
- Order Does **Not** Matter in PLP

Input Scratchpad Memory

Conflict Resolution

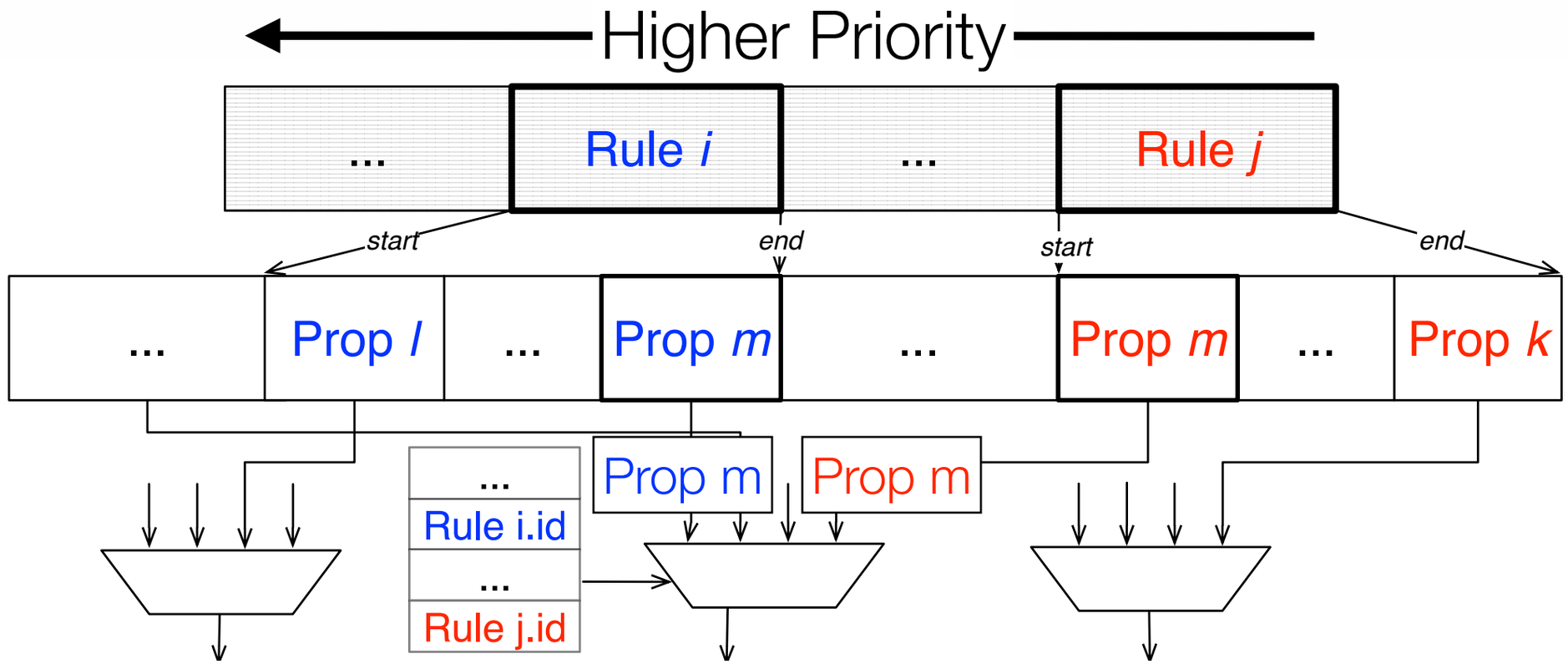


Style Resolution Unit (3)

- Order Matters in RLP
- Order Does **Not** Matter in PLP

Input Scratchpad Memory

Conflict Resolution

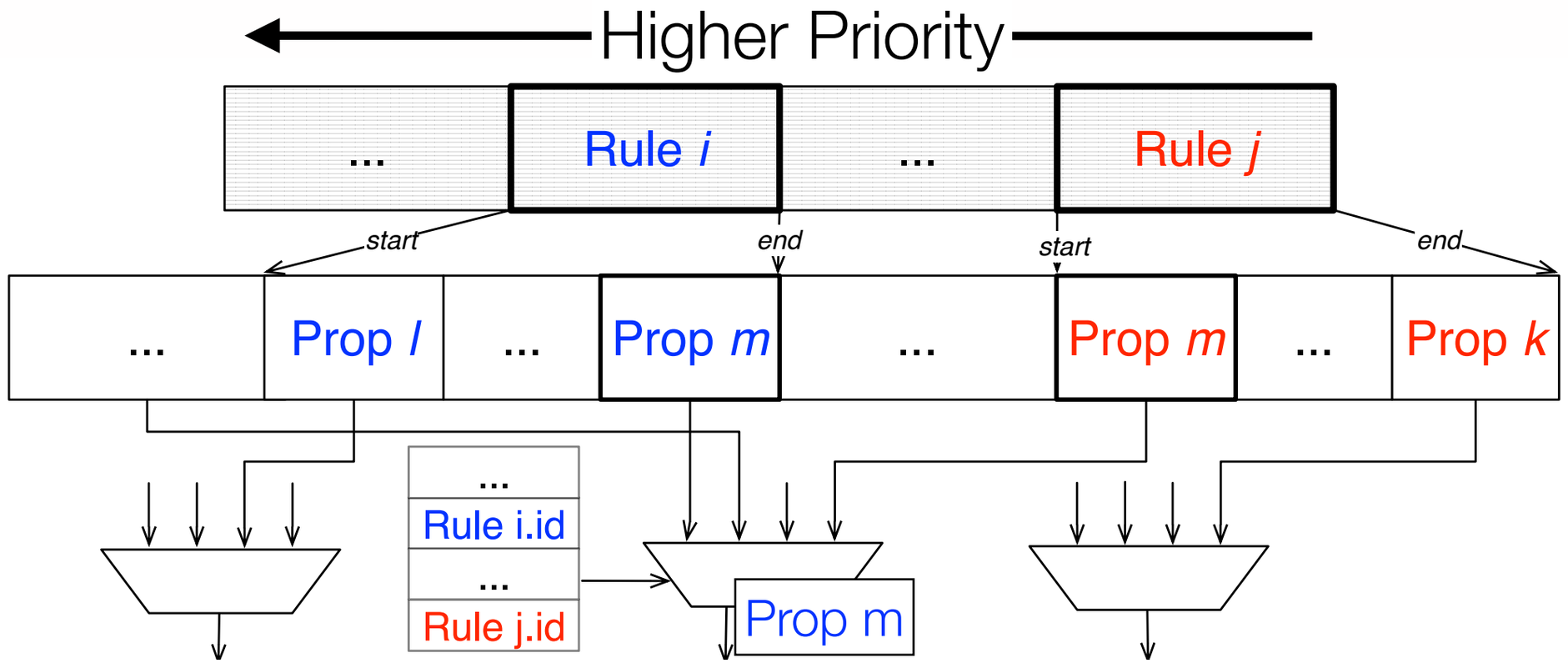


Style Resolution Unit (3)

- Order Matters in RLP
- Order Does **Not** Matter in PLP

Input Scratchpad Memory

Conflict Resolution



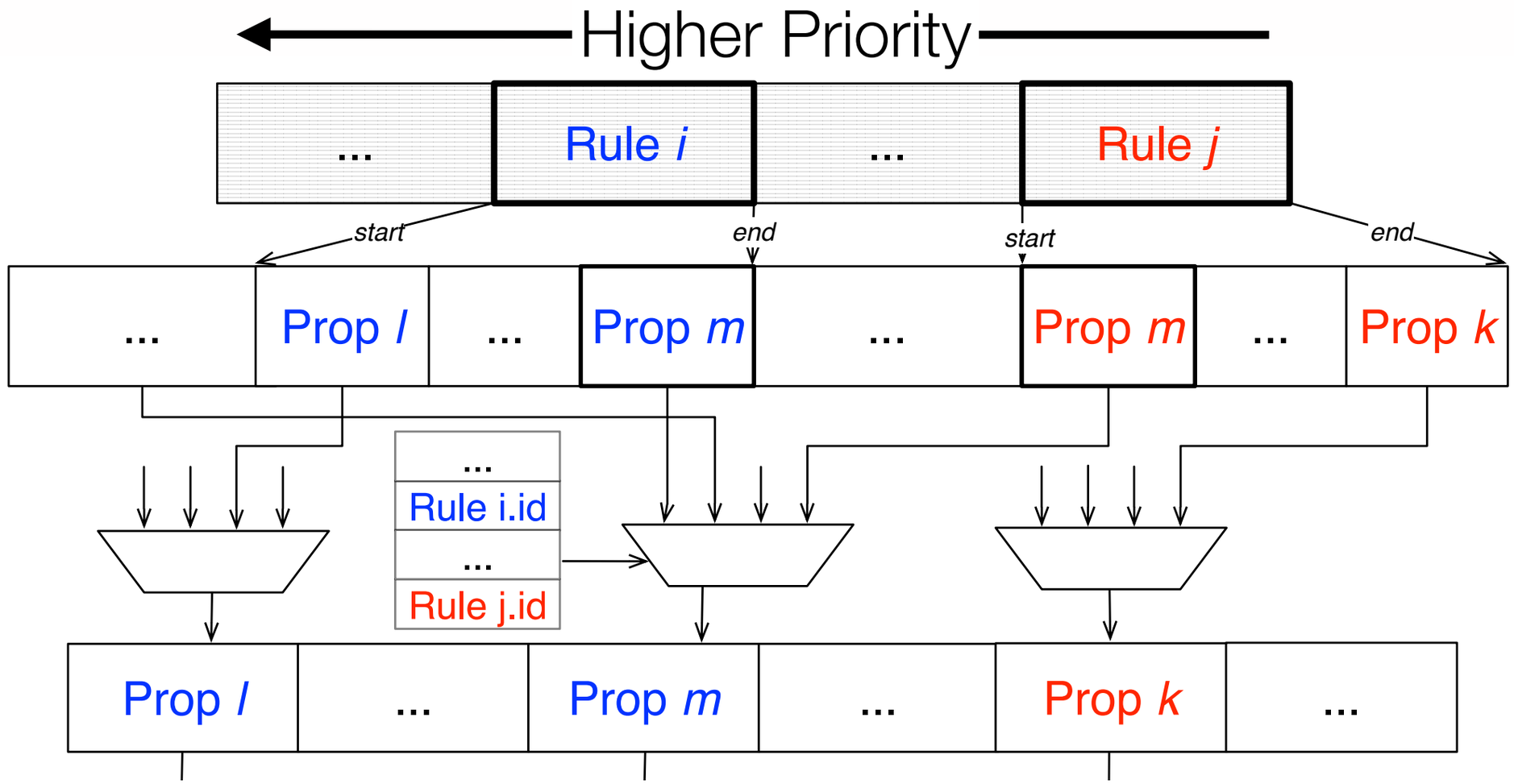
Style Resolution Unit (3)

- Order Matters in RLP
- Order Does **Not** Matter in PLP

Input Scratchpad Memory

Conflict Resolution

Compute Lanes



Style Resolution Unit (3)

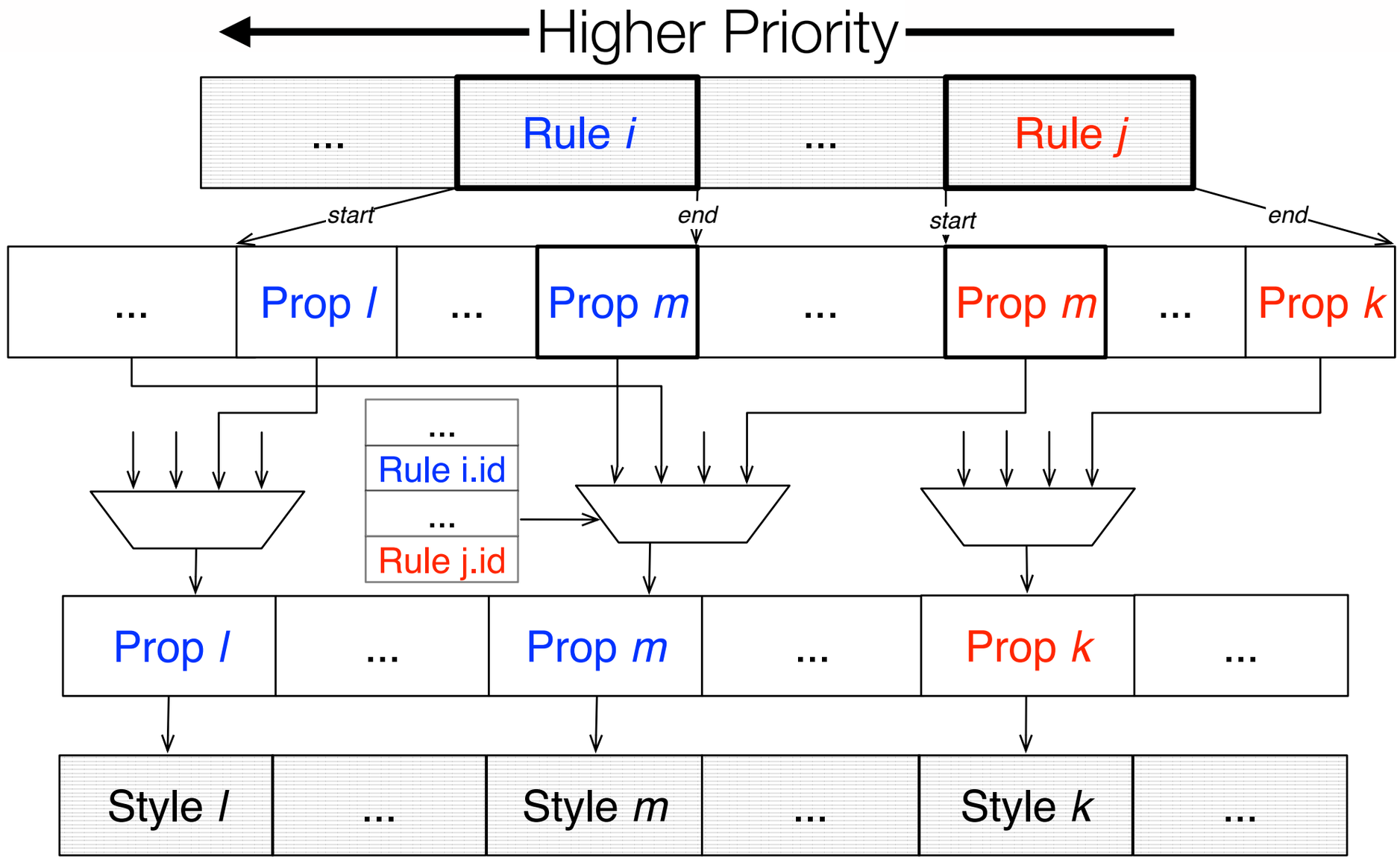
- Order Matters in RLP
- Order Does **Not** Matter in PLP

Input Scratchpad Memory

Conflict Resolution

Compute Lanes

Output Scratchpad Memory



Agenda of Today's Talk

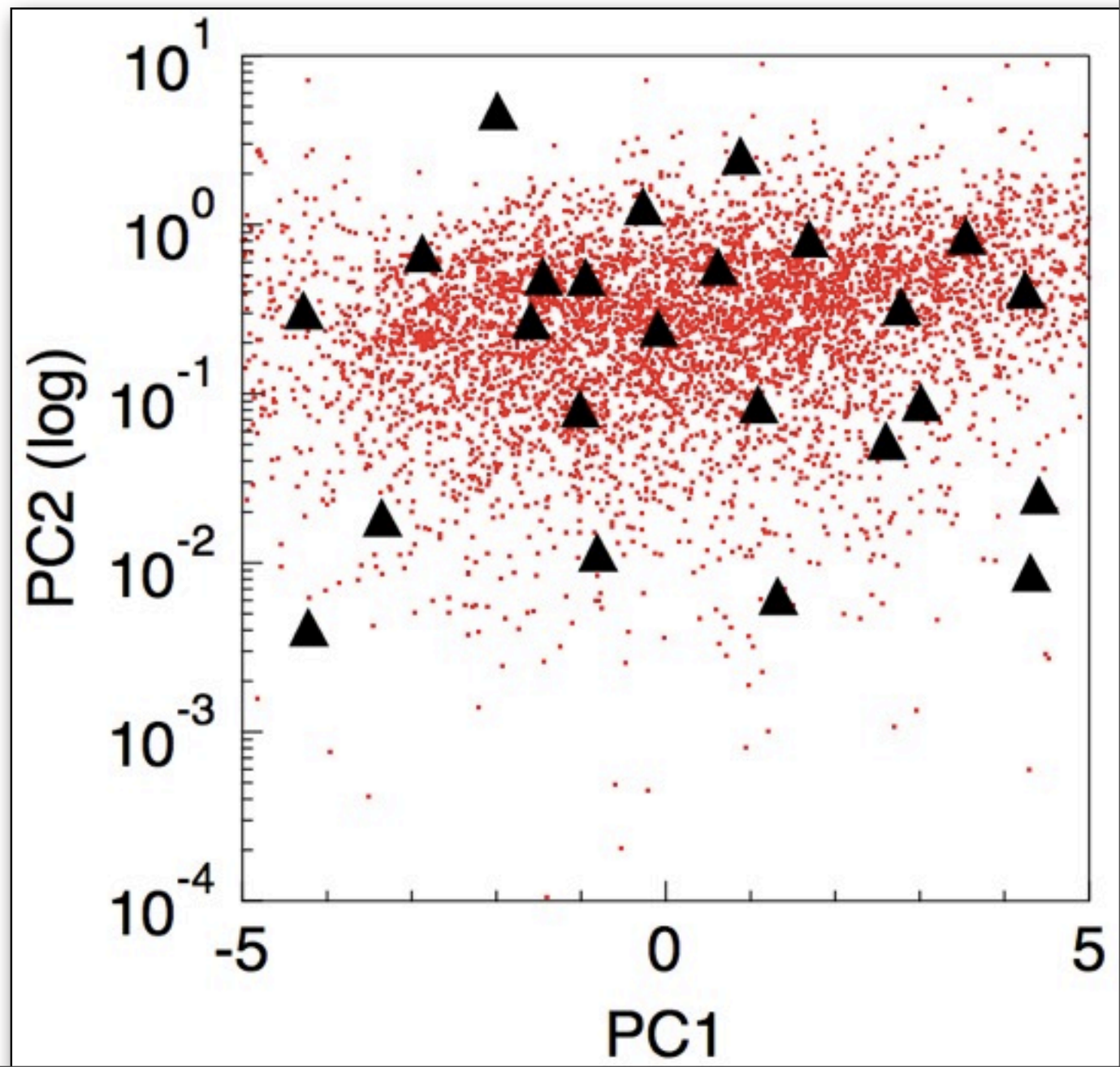
- Motivation of our work: energy-efficiency of the mobile Web
- How does WebCore improve the energy-efficiency?
  - Customization
  - Specialization
- **Evaluation Results**
- Related Work

Evaluations

- Fully synthesized using Synopsys 28 nm toolchain

Evaluations

- Fully synthesized using Synopsys 28 nm toolchain
- 24 representative webpages

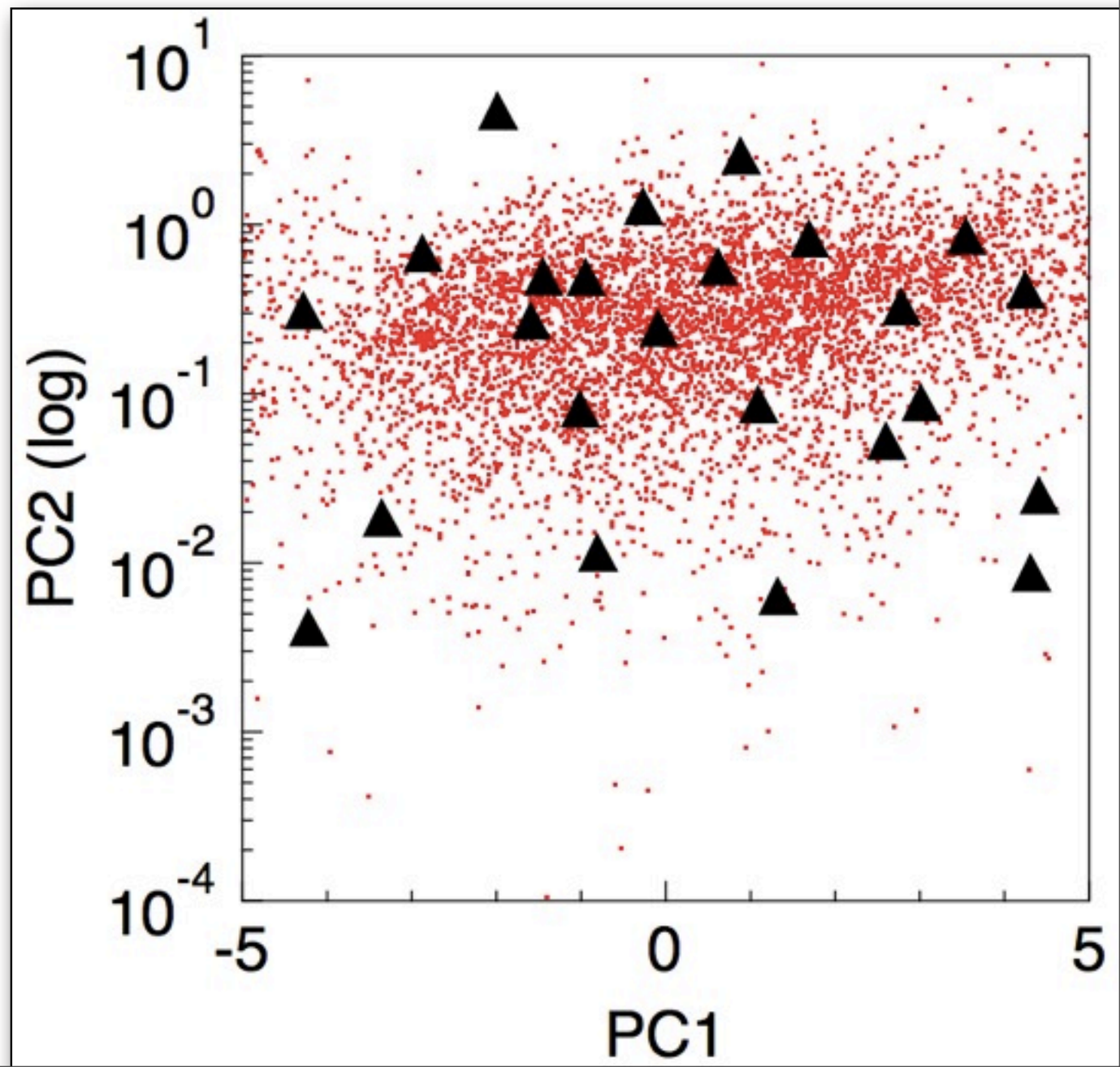


Evaluations

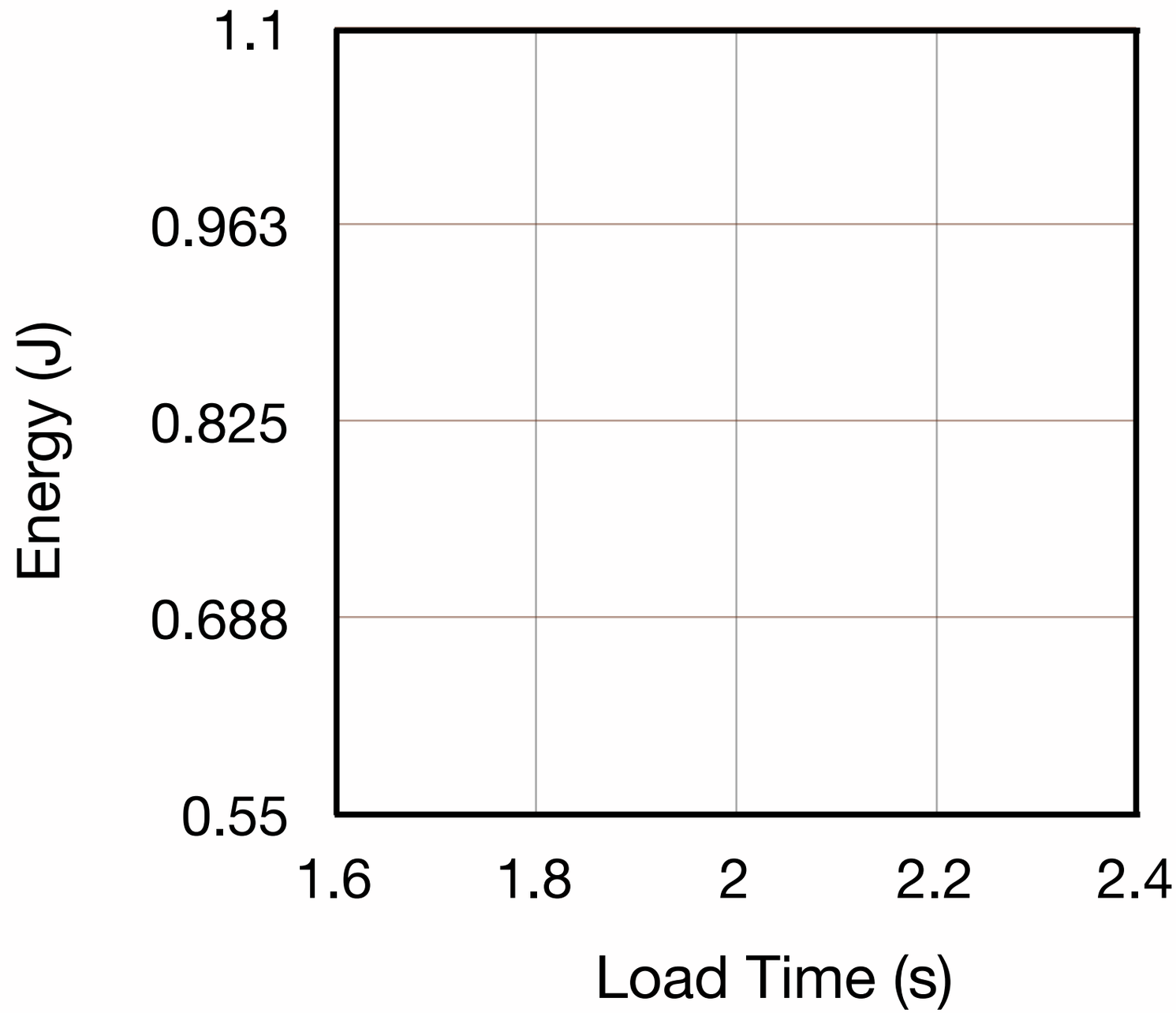
- Fully synthesized using Synopsys 28 nm toolchain
- 24 representative webpages

Desktop and mobile versions

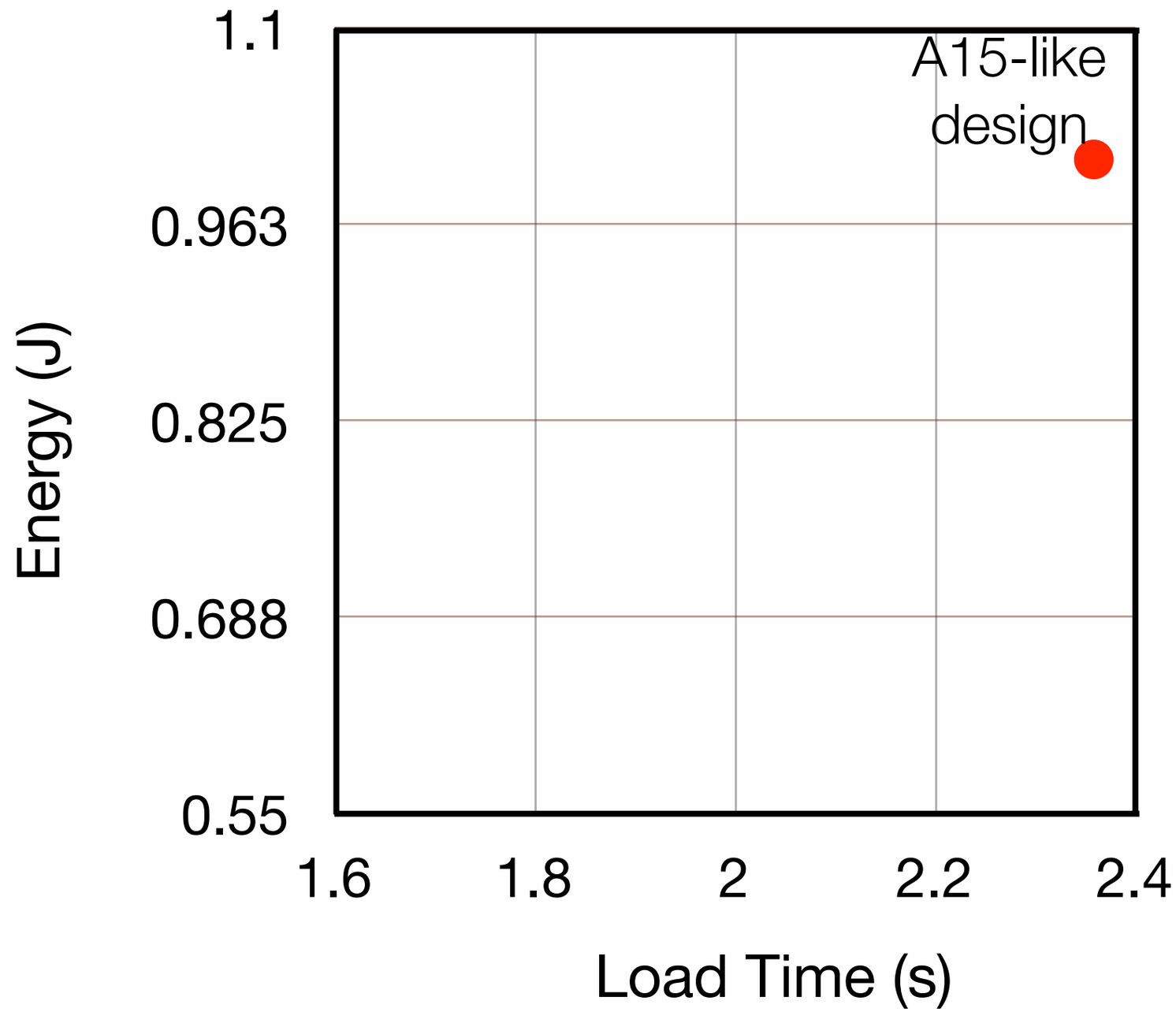
www.amazon.com
www.cnn.com
www.msn.com
www.google.com.hk
www.twitter.com
www.espn.go.com
www.bbc.co.uk
www.slashdot.org
www.youtube.com
www.ebay.com
www.sina.com.cn
www.163.com



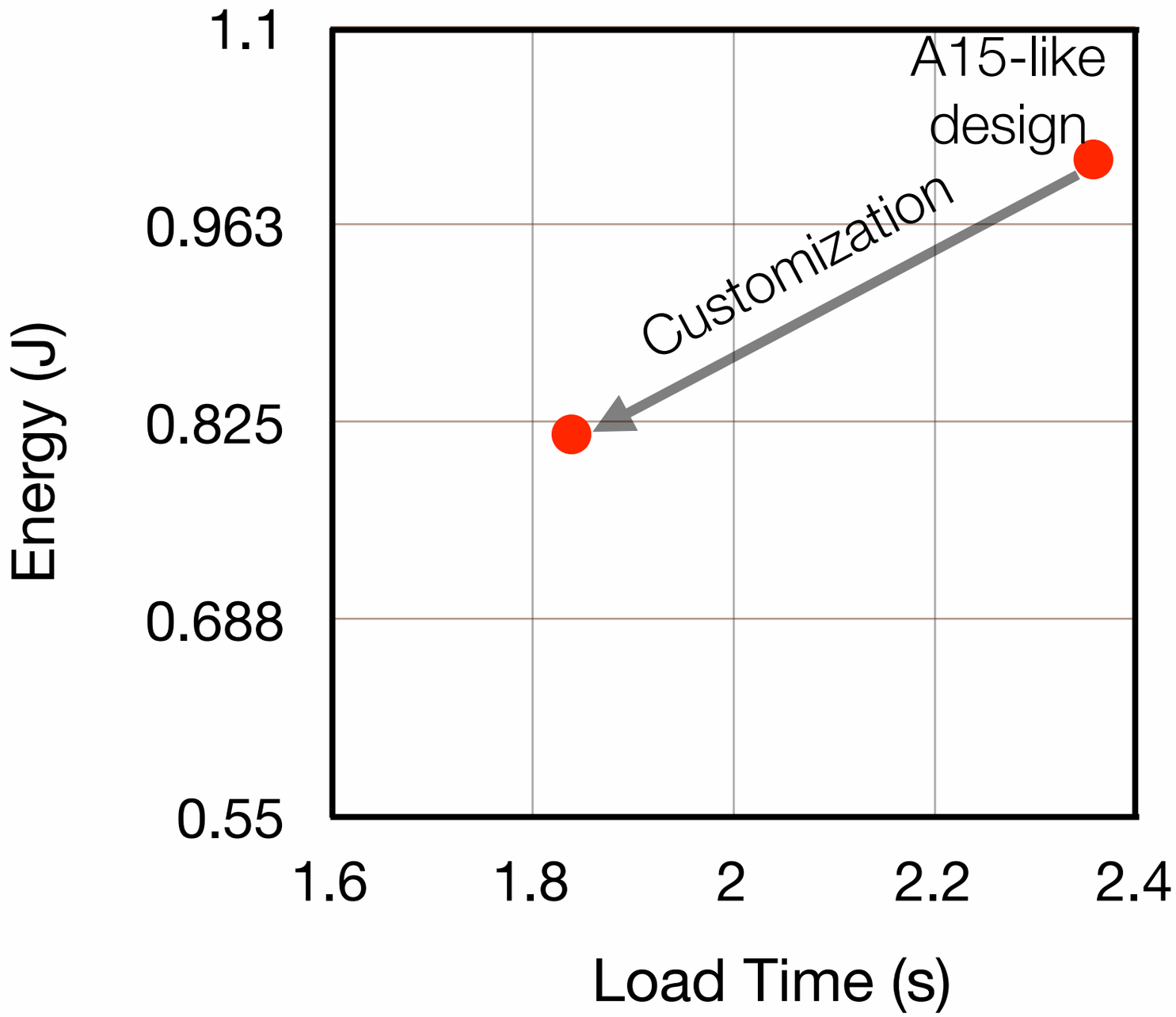
Evaluations



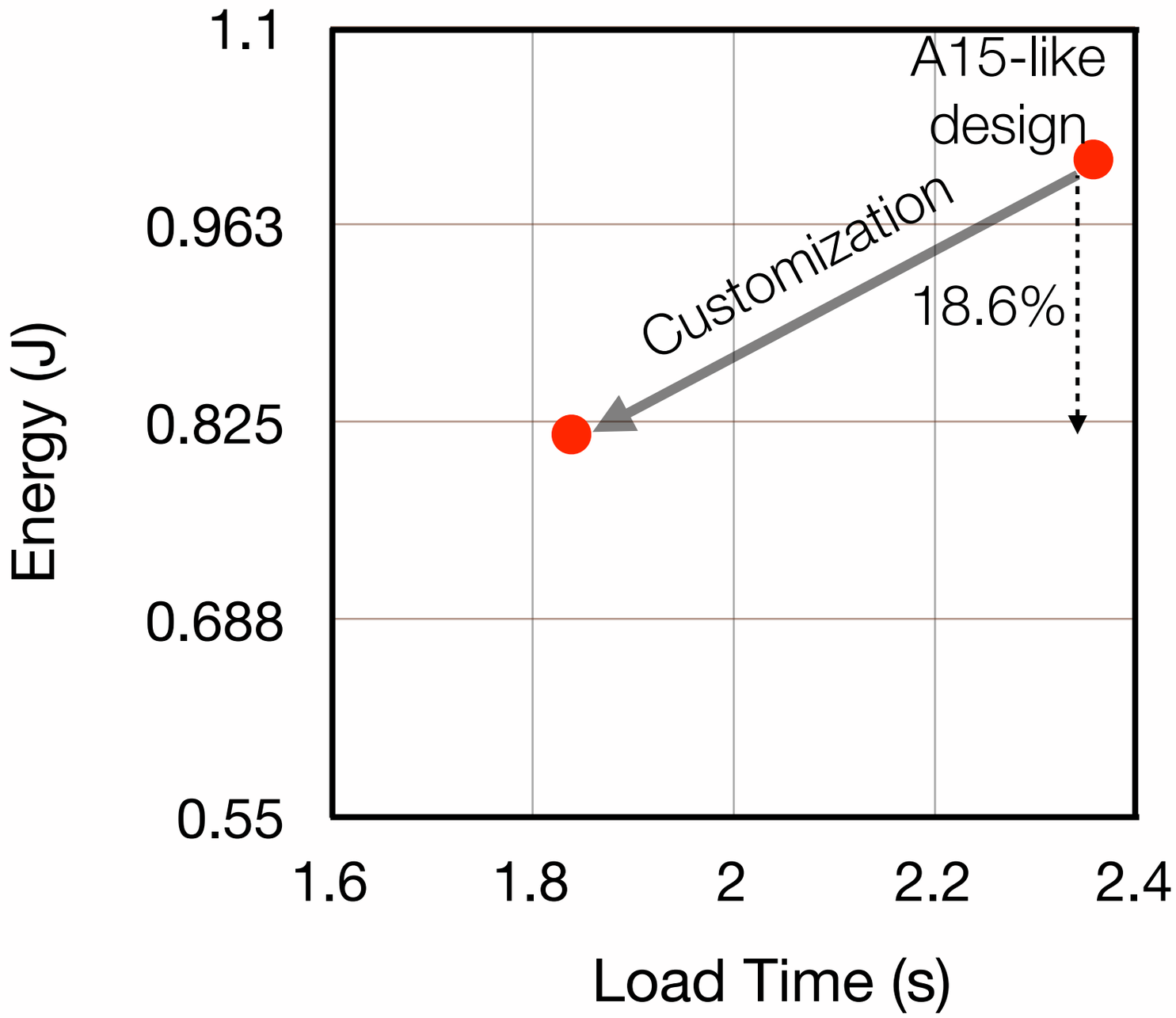
Evaluations



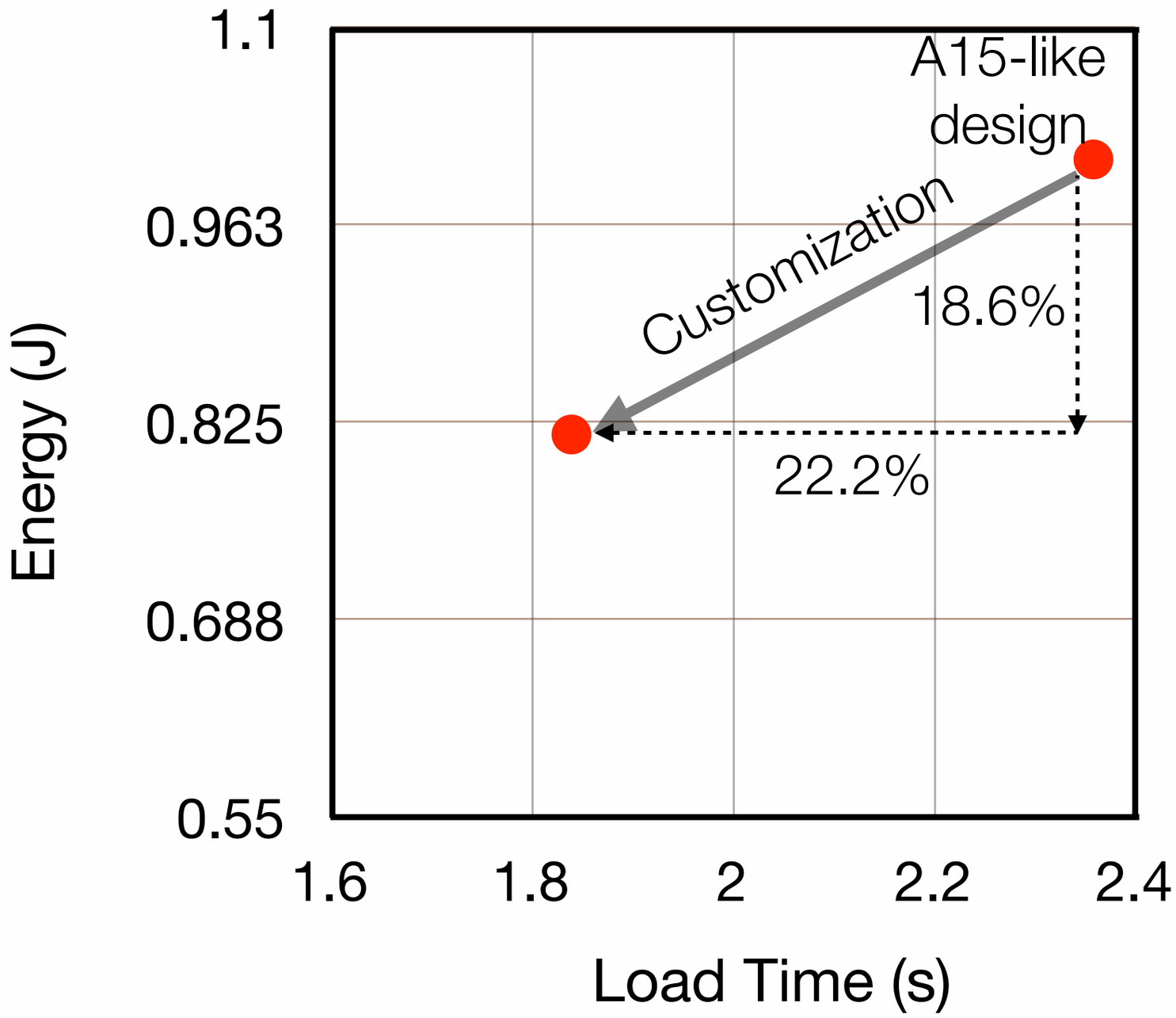
Evaluations



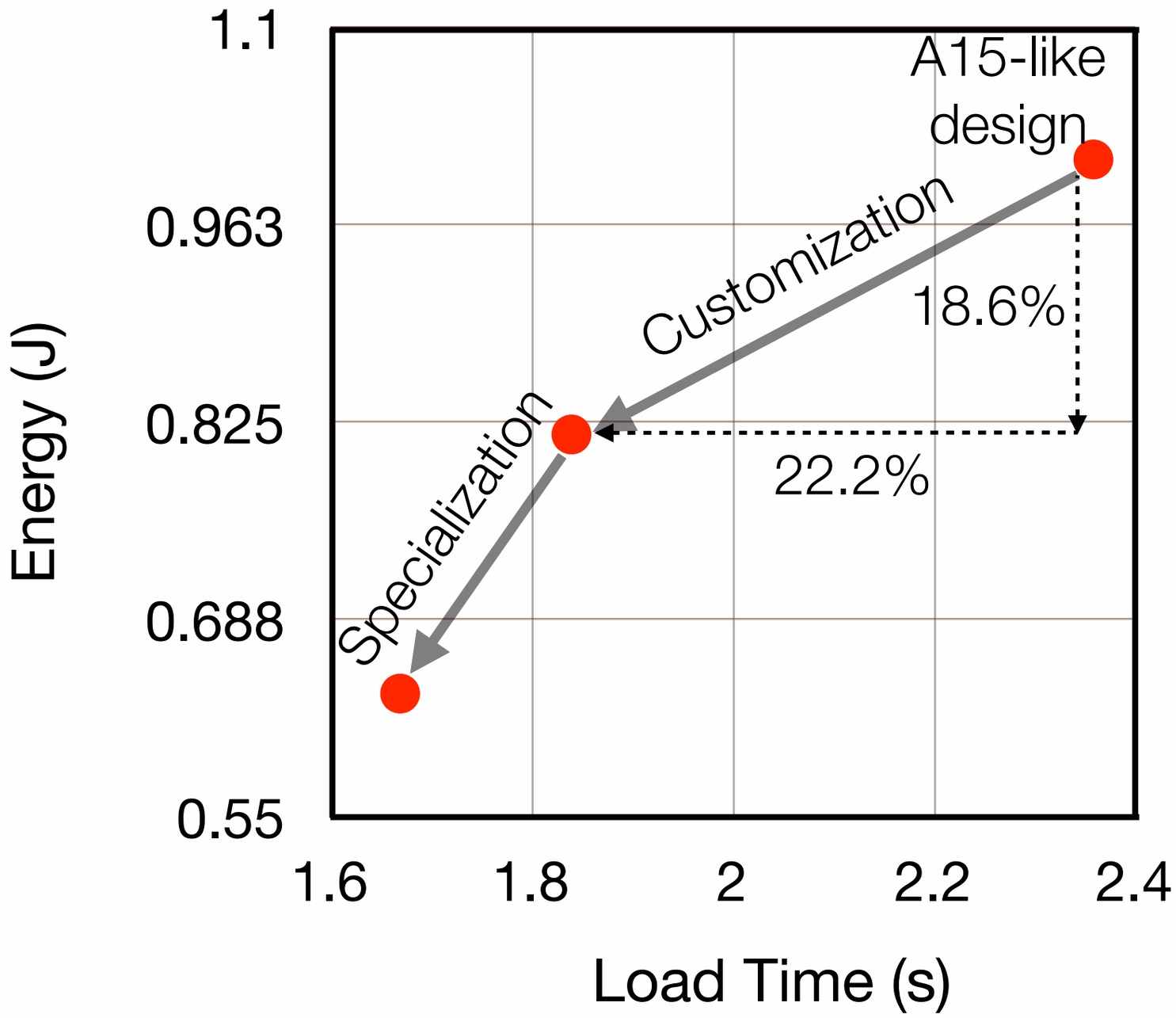
Evaluations



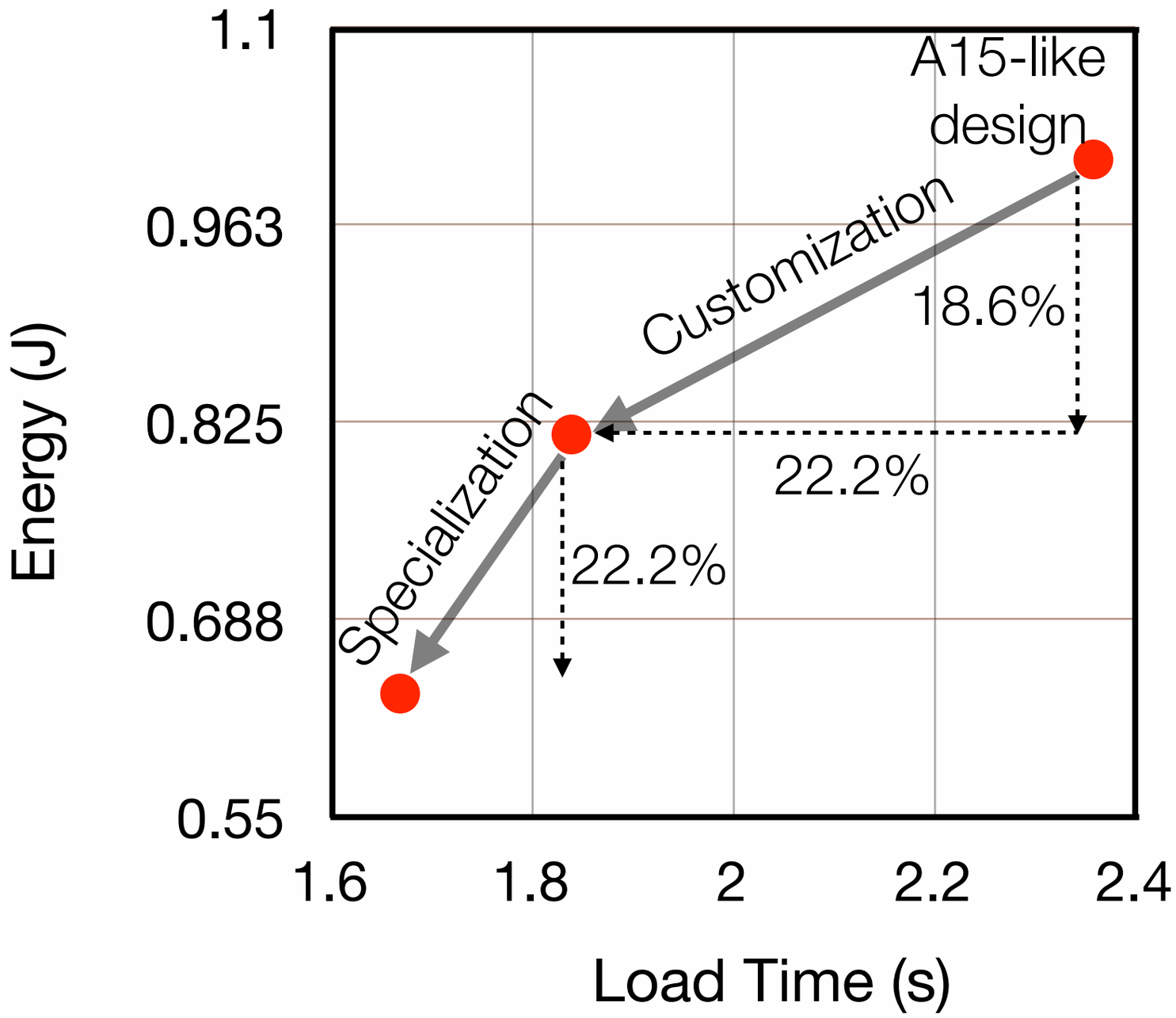
Evaluations



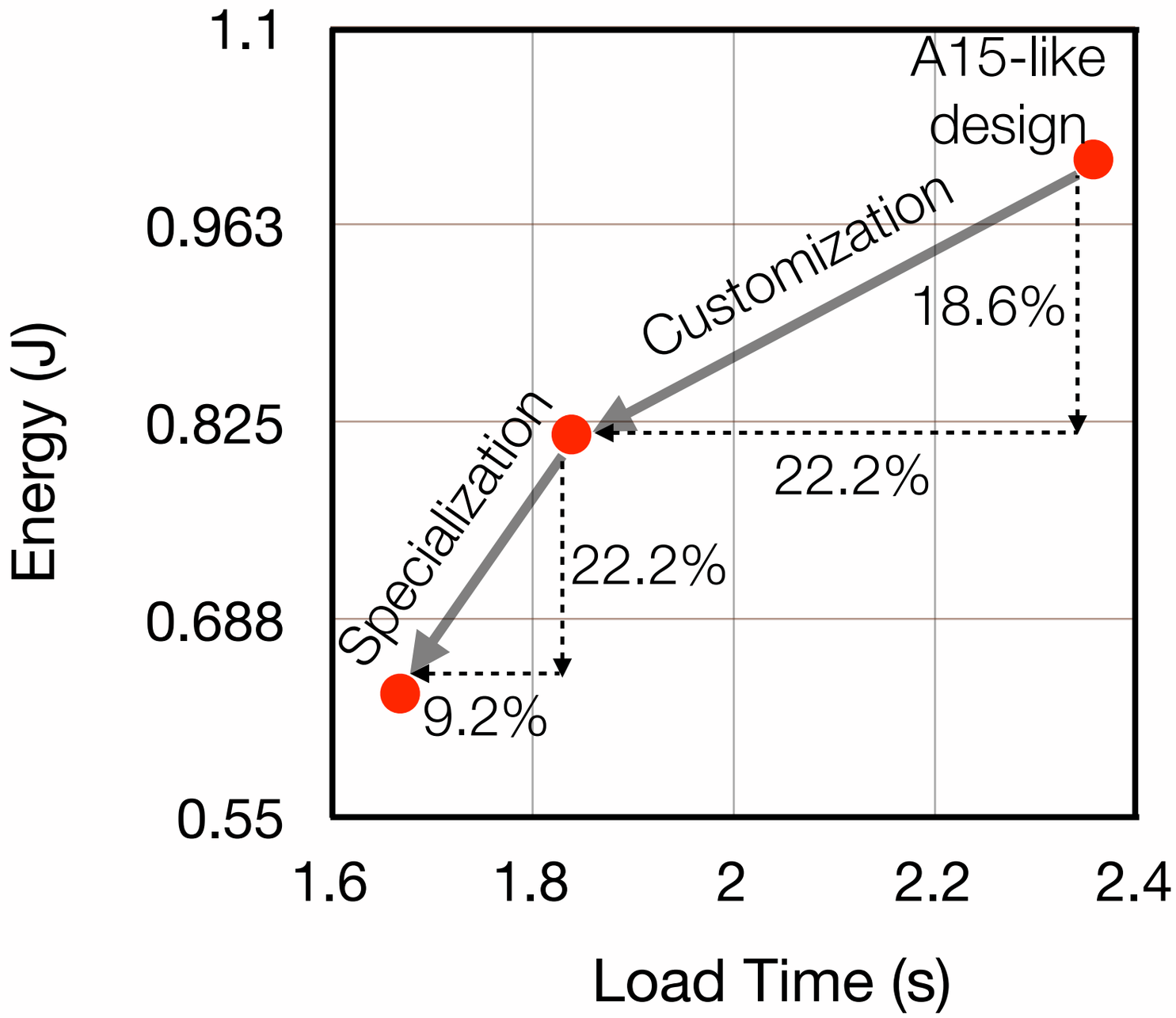
Evaluations



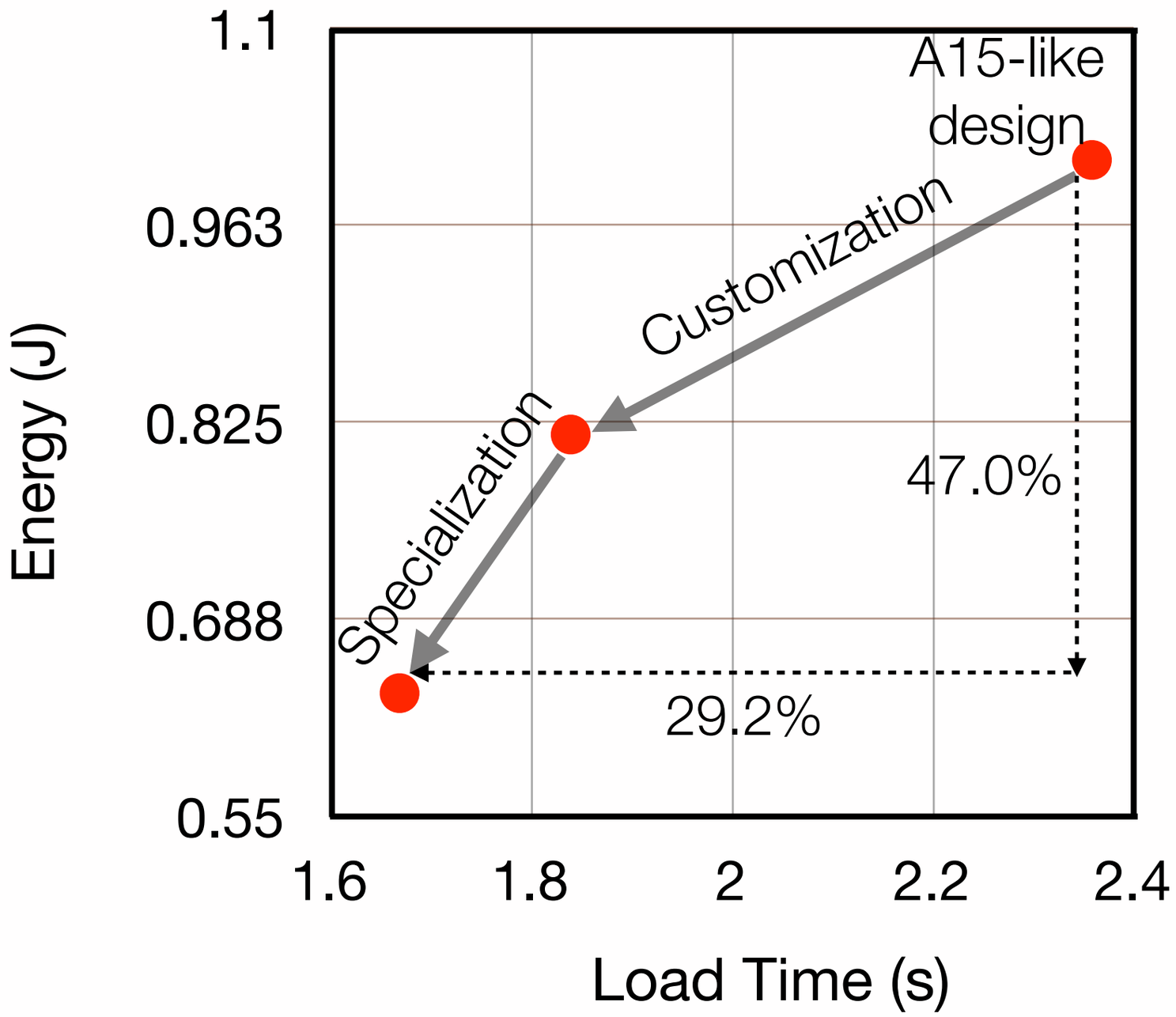
Evaluations



Evaluations

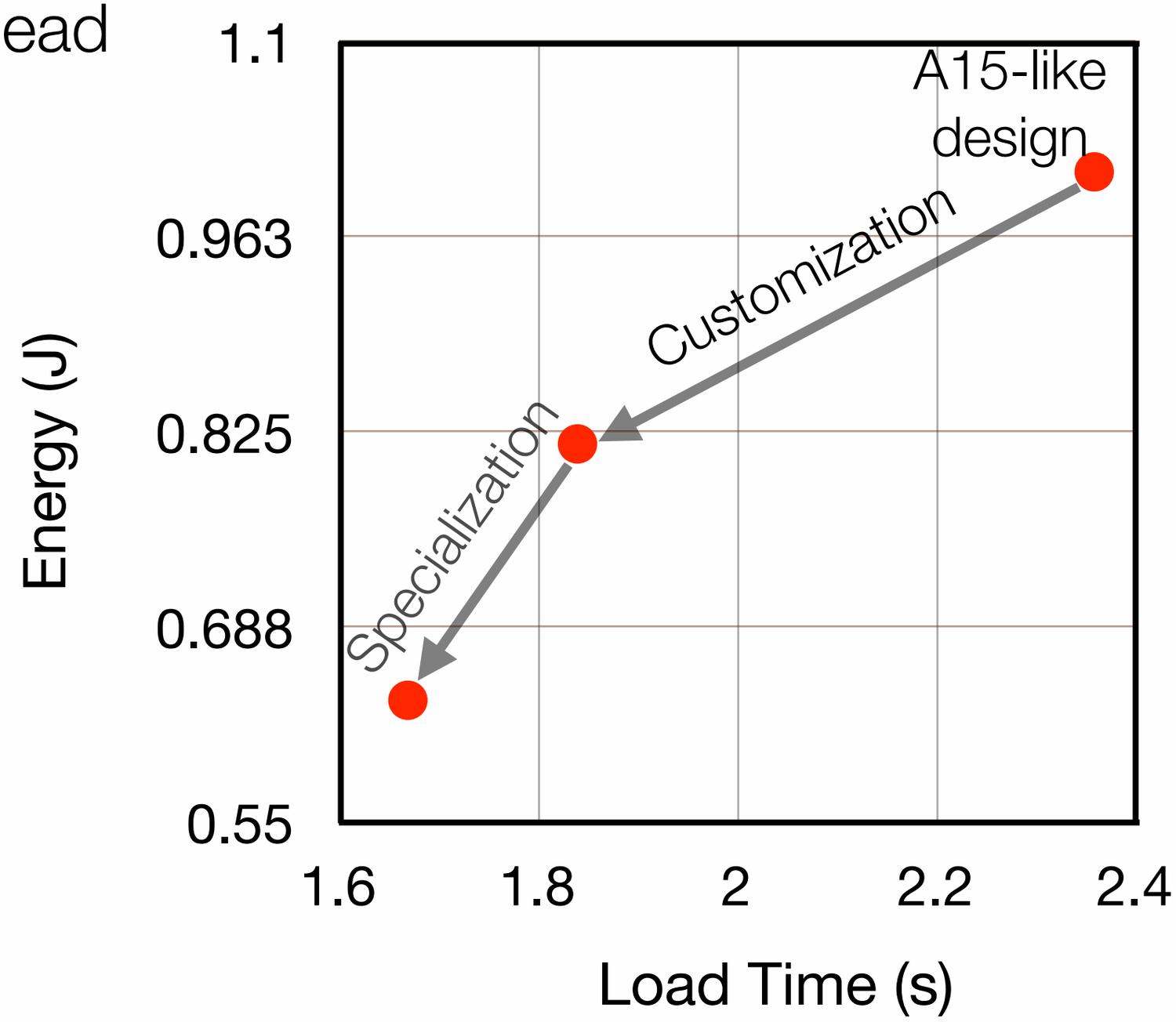


Evaluations



Evaluations

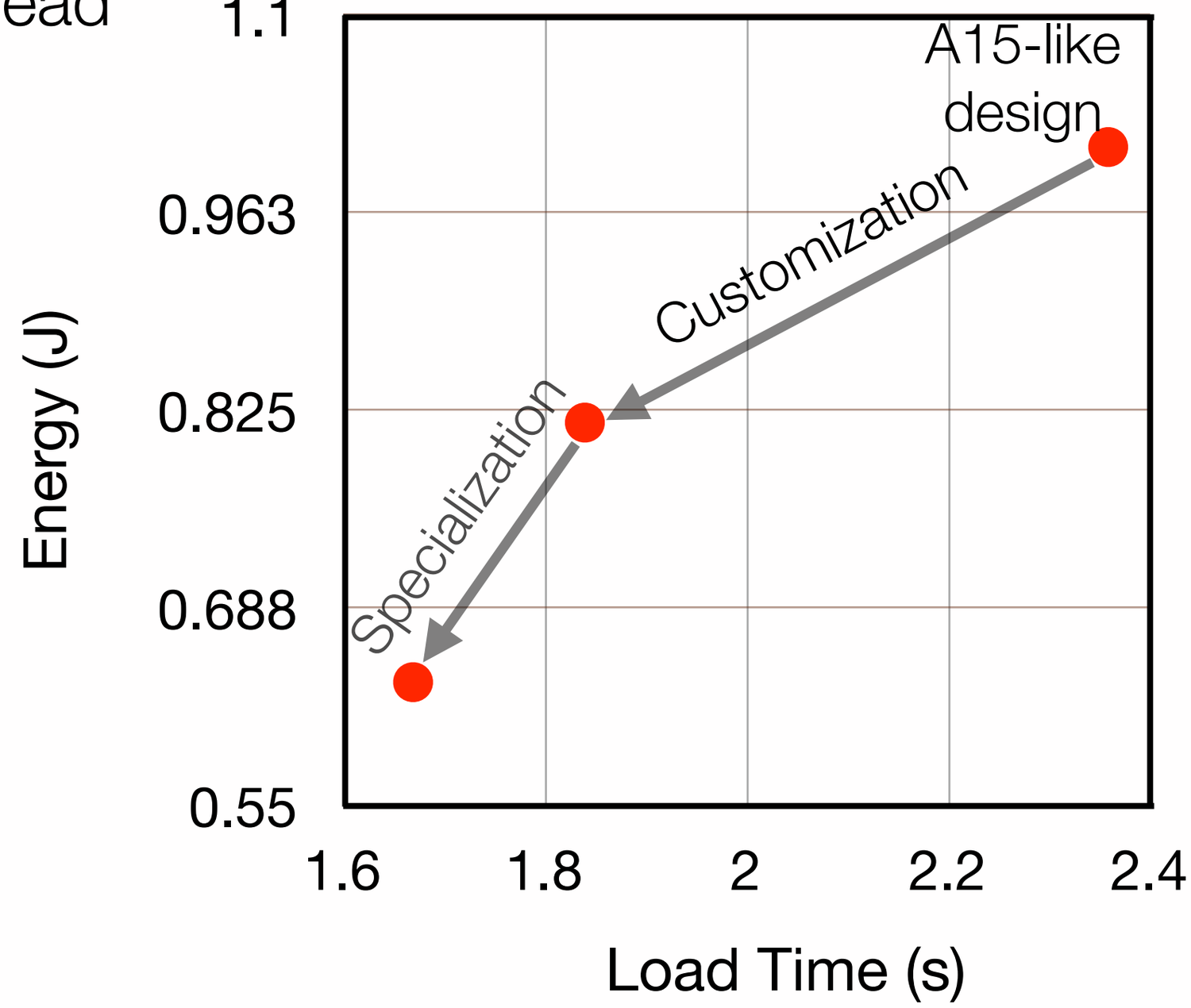
Cost of specialization:
0.59 mm^2 area overhead



Evaluations

Cost of specialization:
0.59 mm^2 area overhead

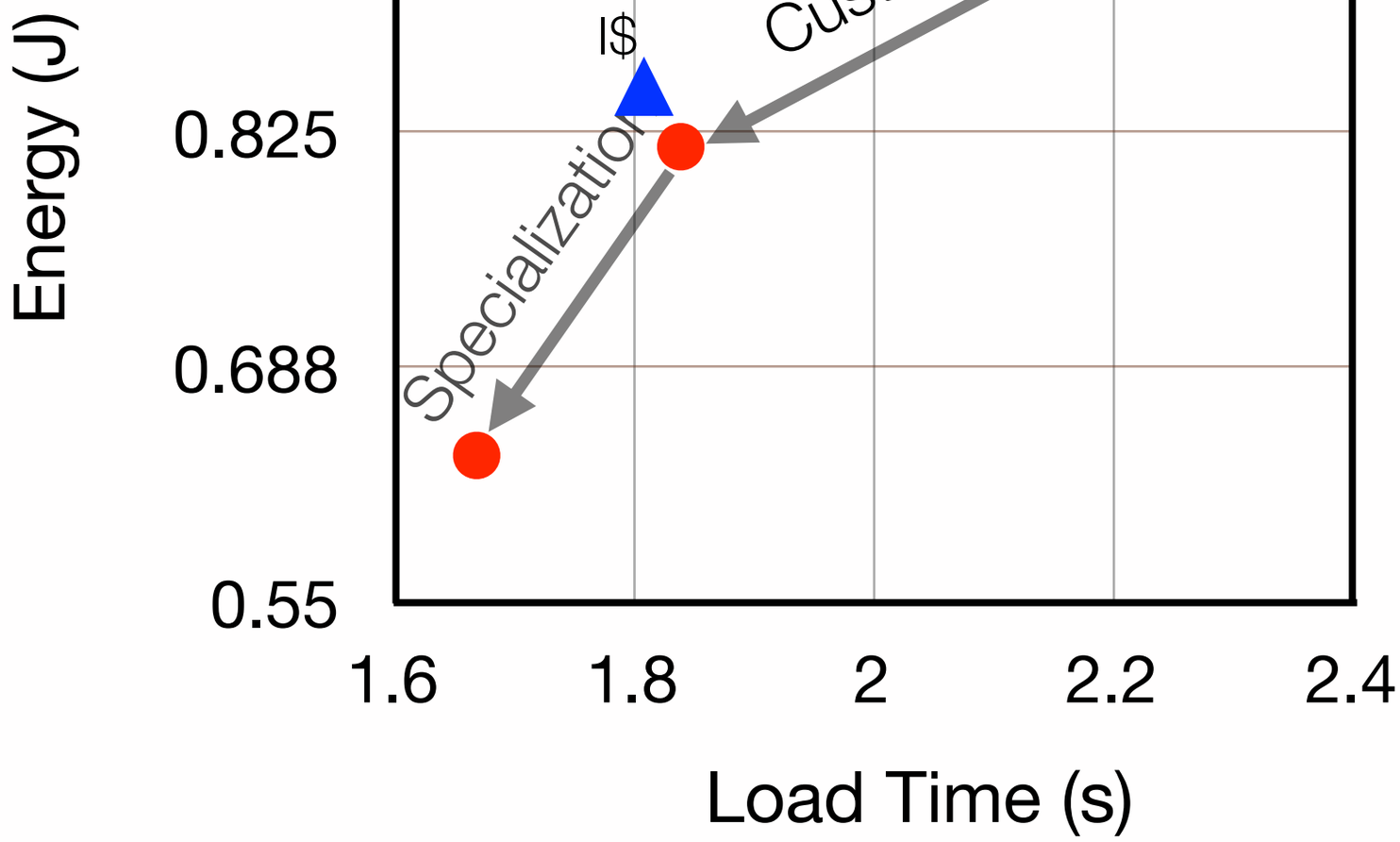
Better than scaling-up approaches



Evaluations

Cost of specialization:
0.59 mm^2 area overhead

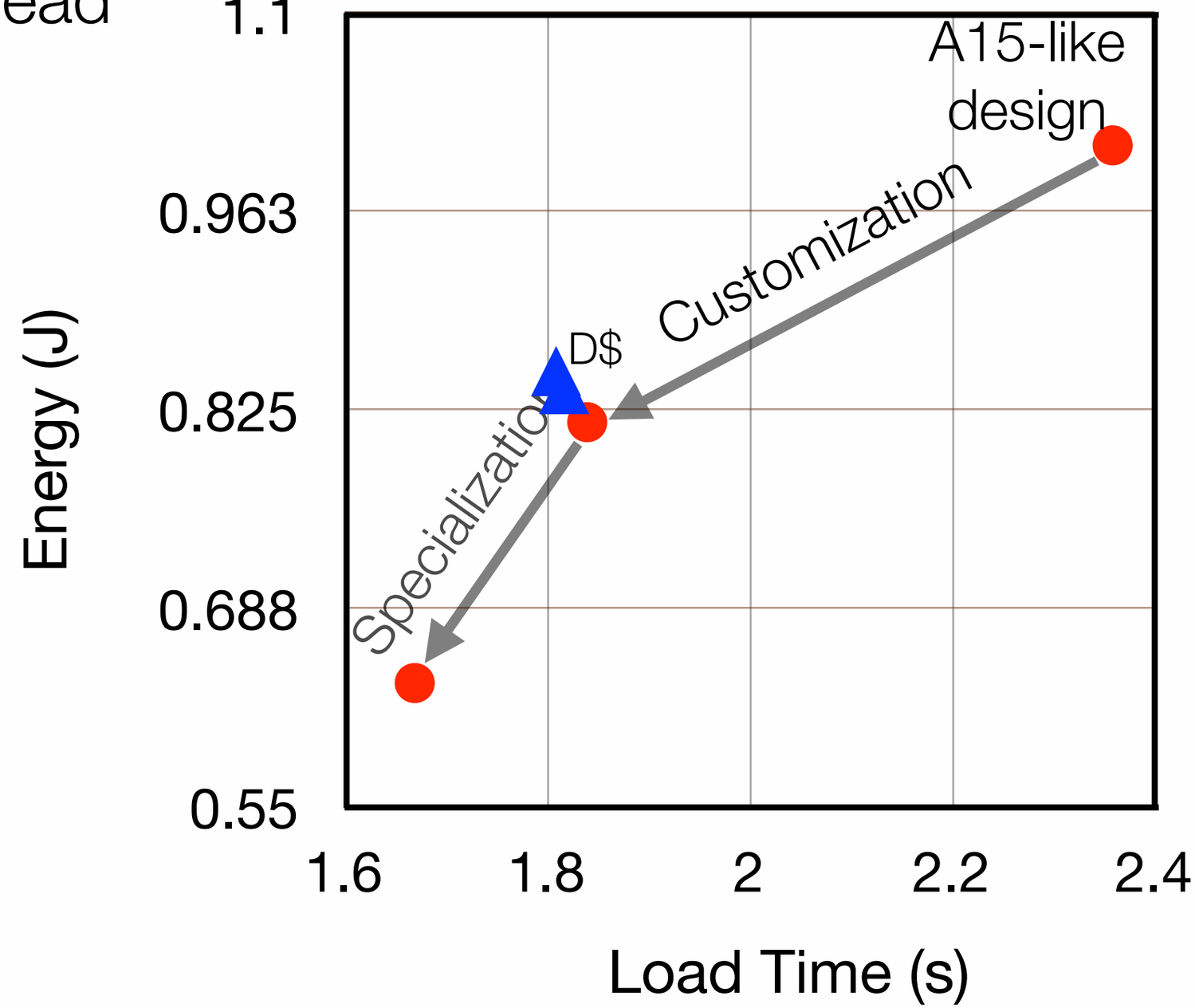
Better than scaling-up approaches



Evaluations

Cost of specialization:
0.59 mm^2 area overhead

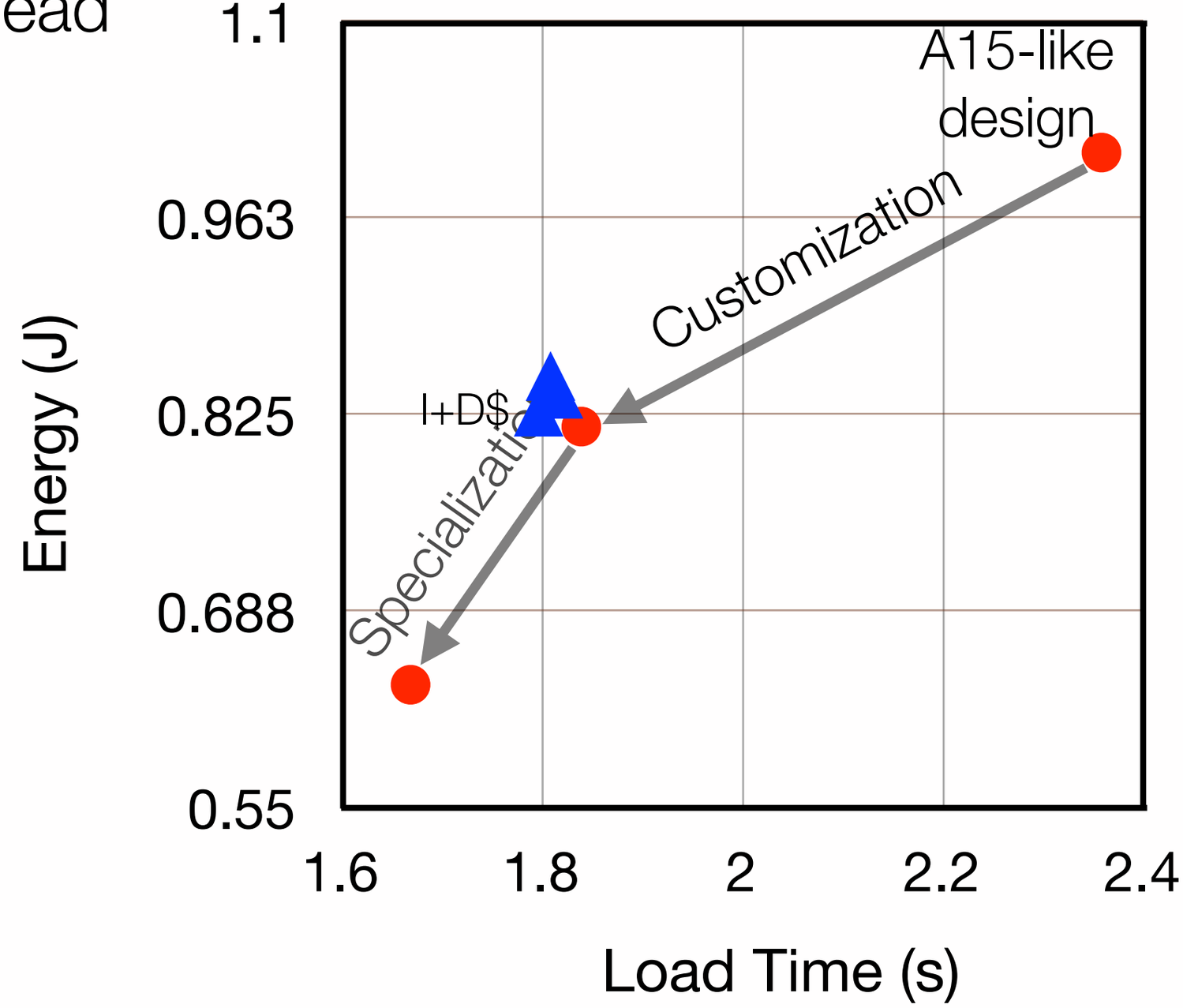
Better than scaling-up approaches



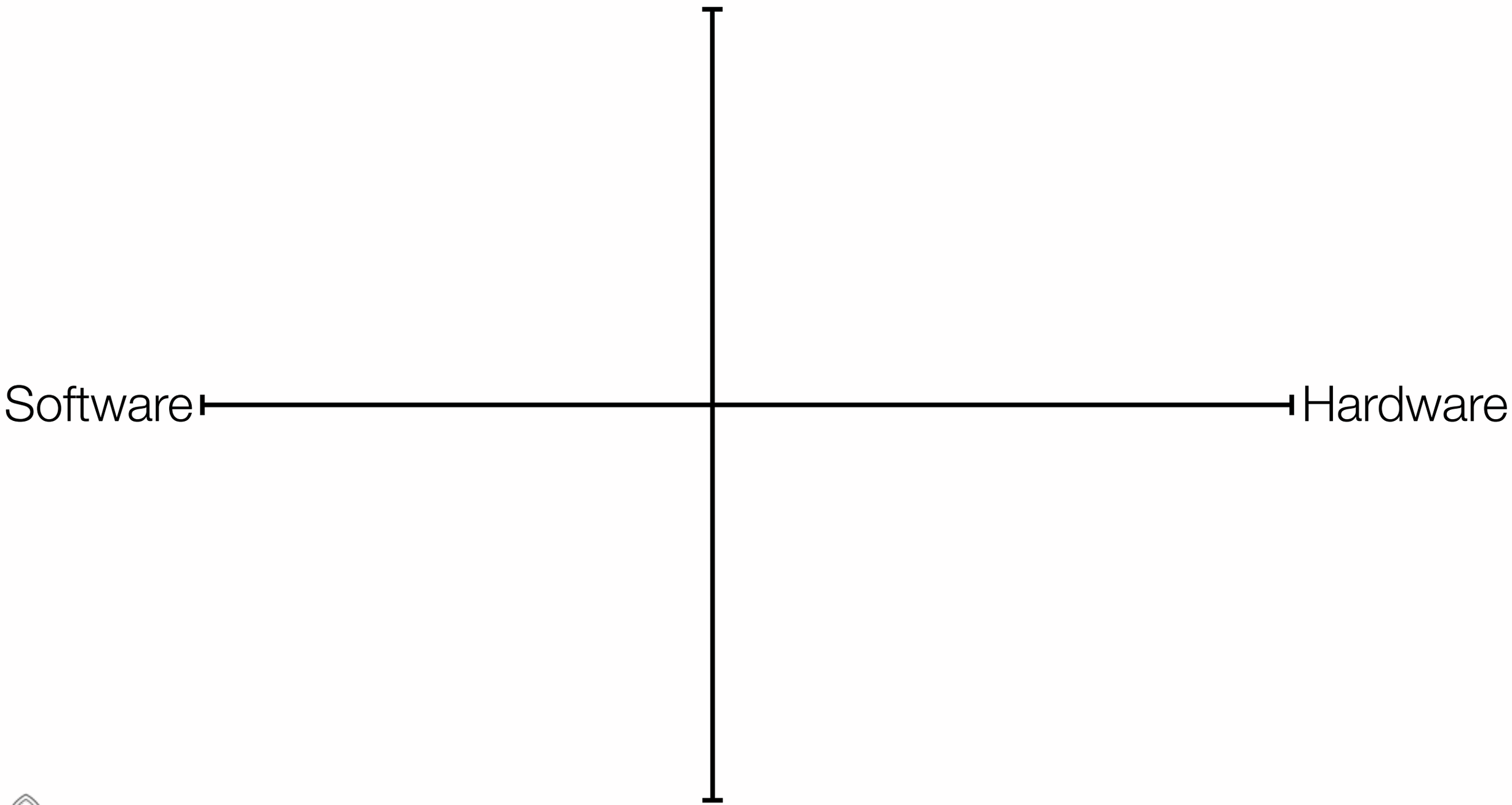
Evaluations

Cost of specialization:
0.59 mm^2 area overhead

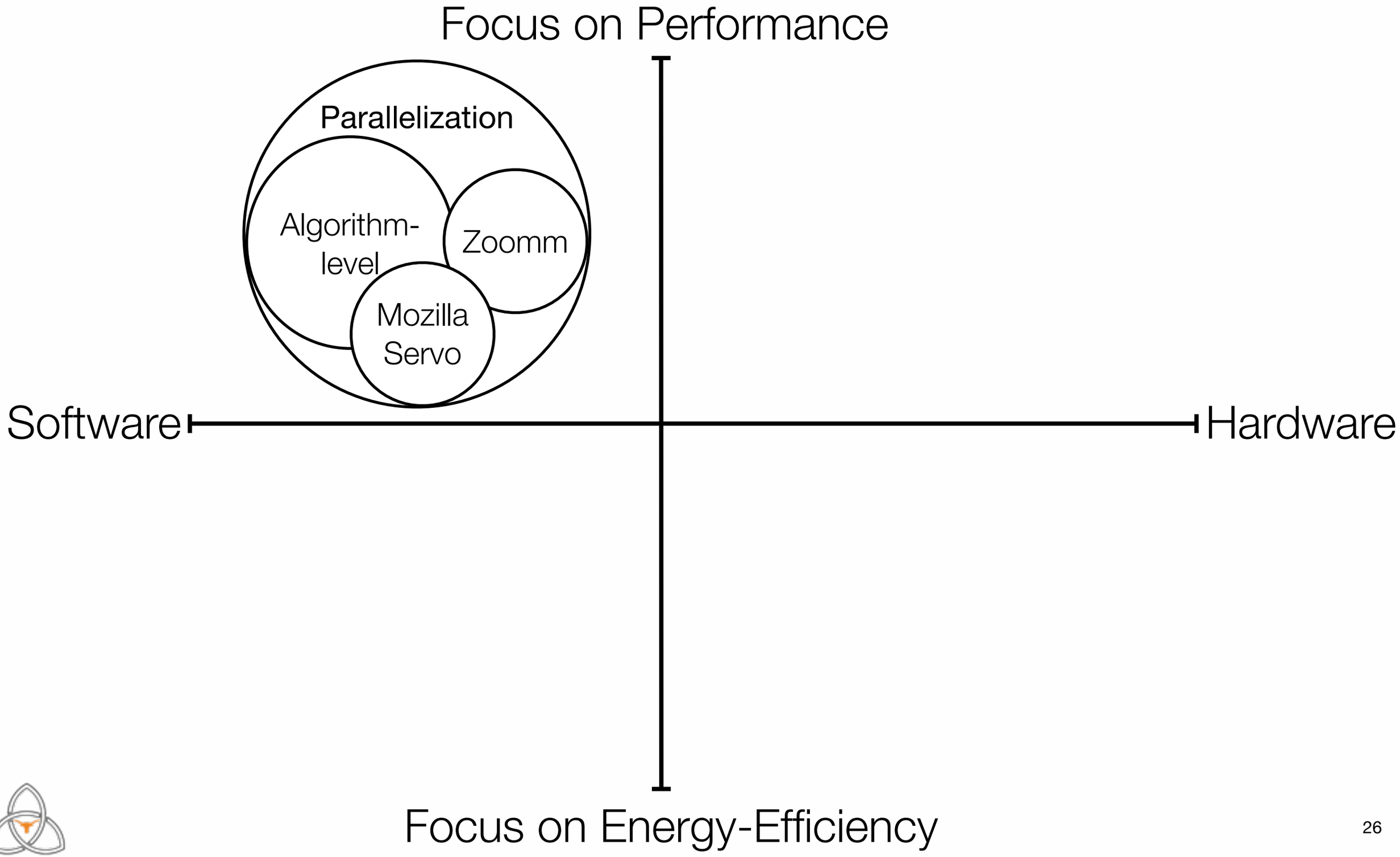
Better than scaling-up approaches



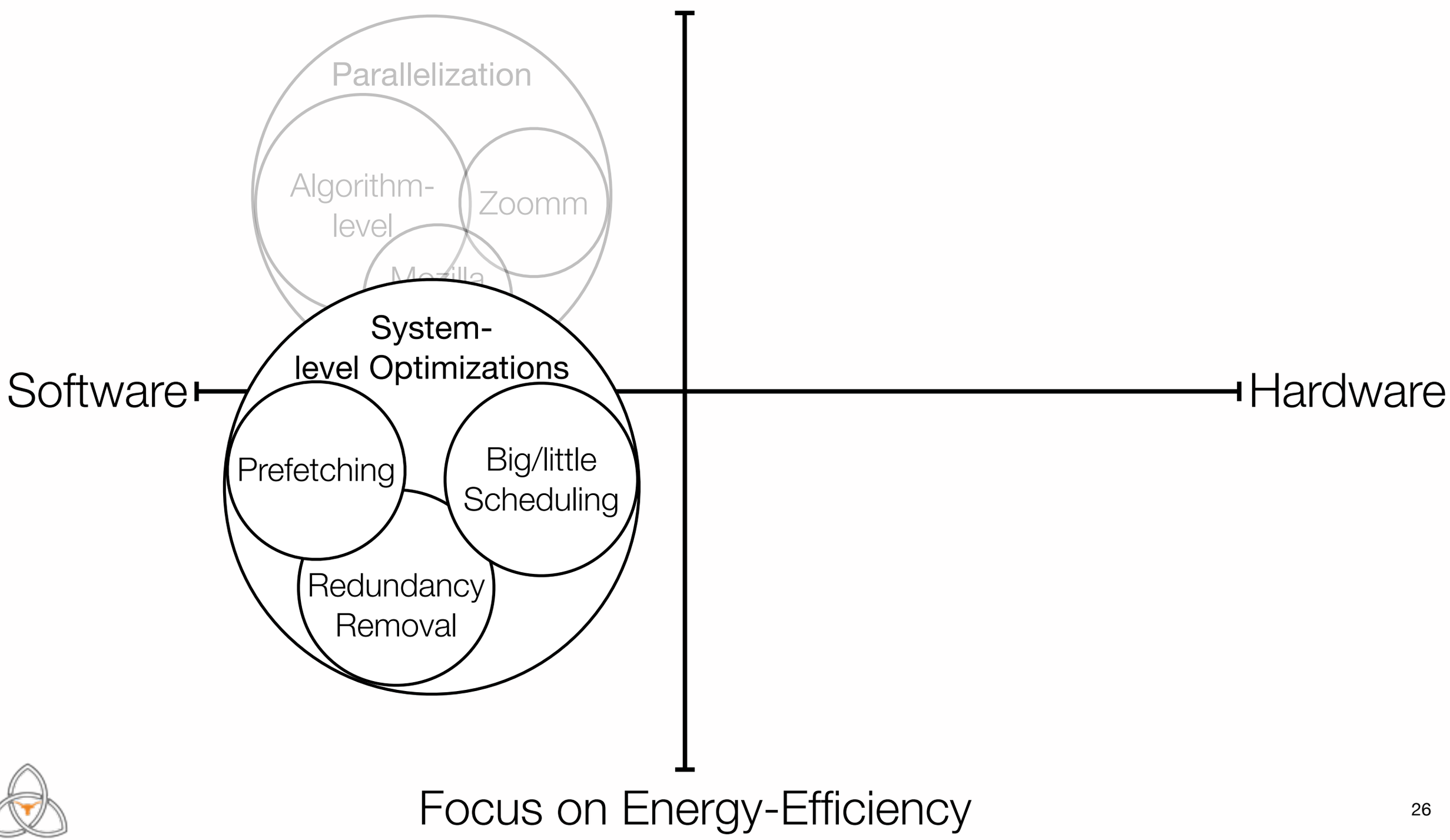
Related Work



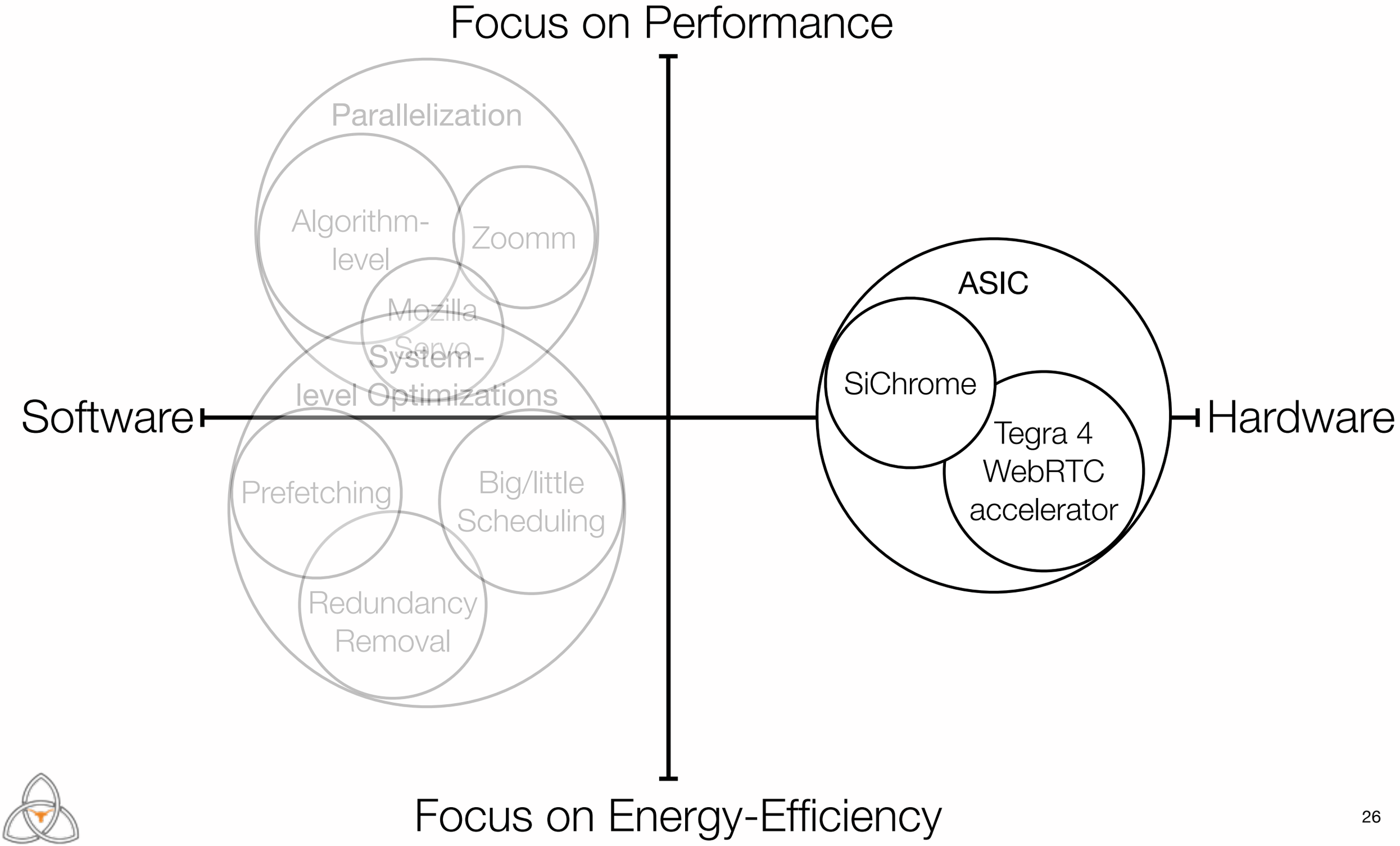
Related Work



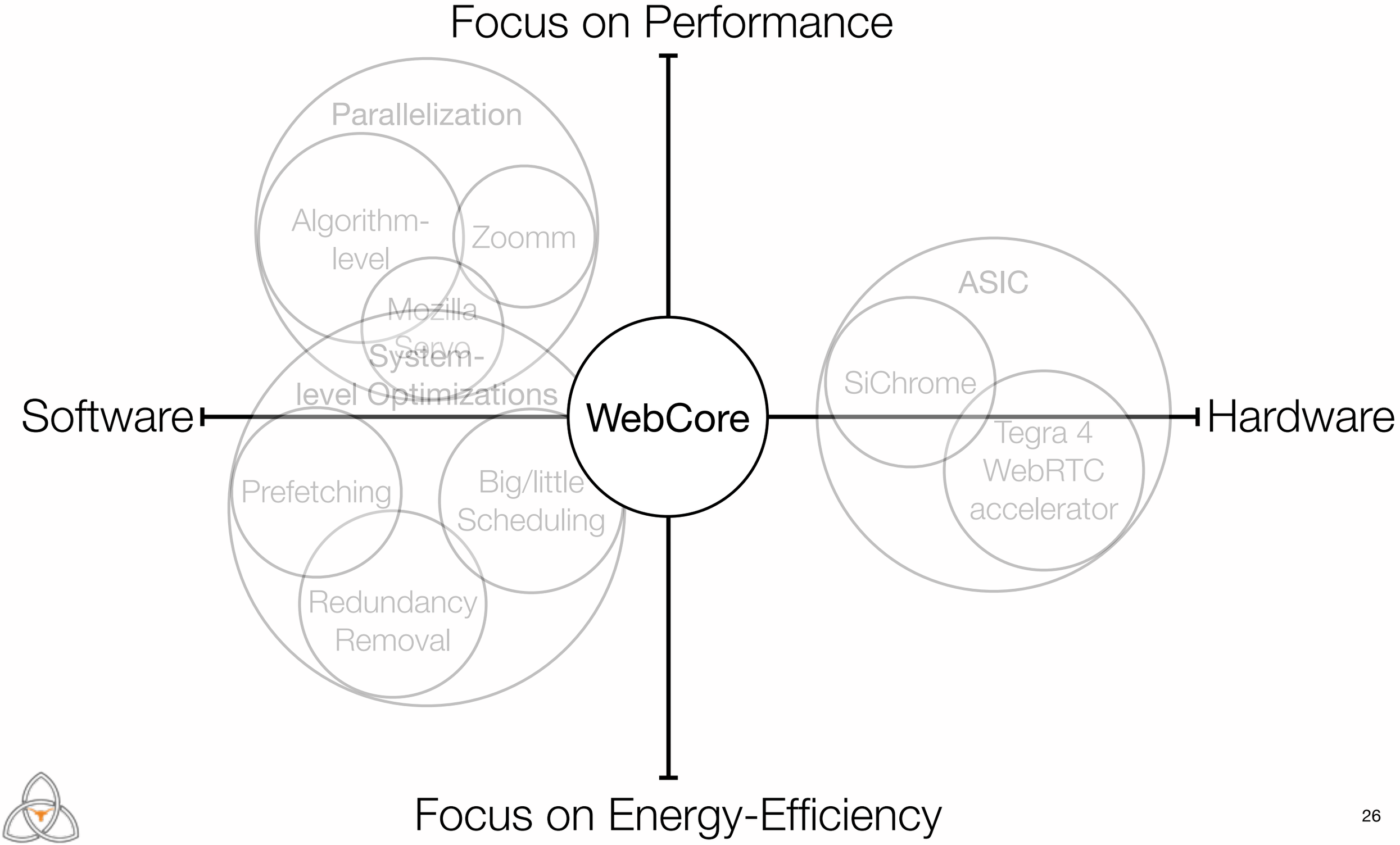
Related Work



Related Work

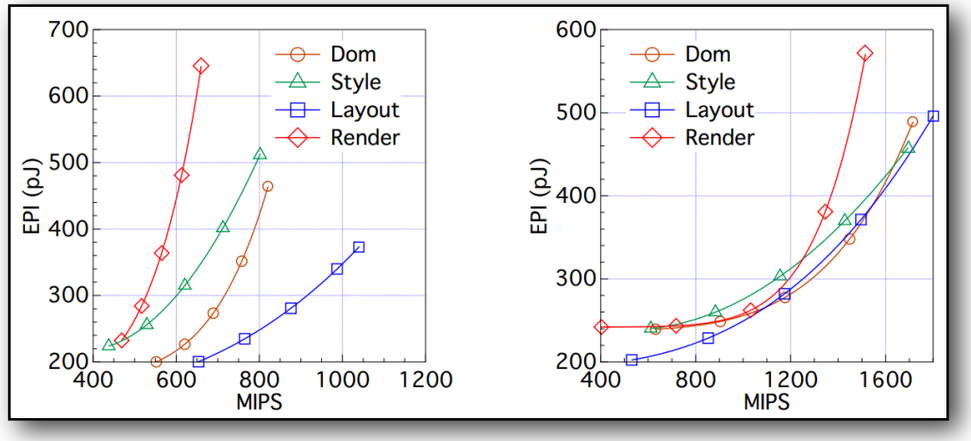


Related Work

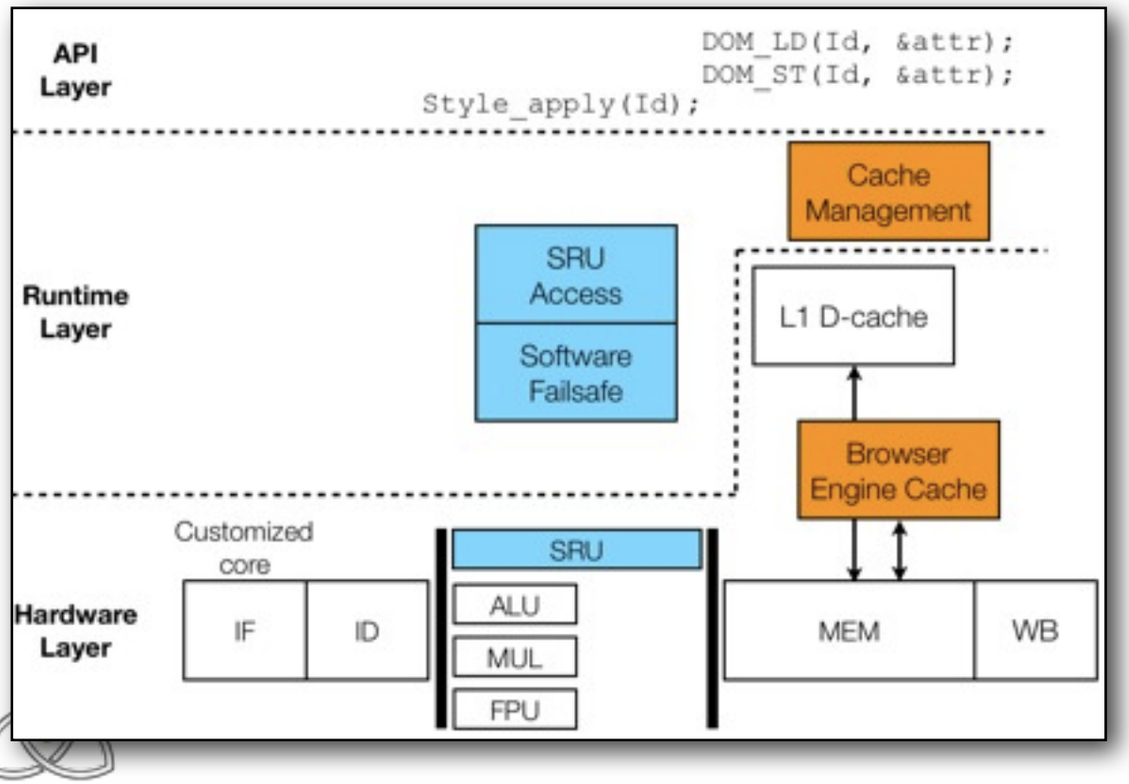


Conclusions

The Web browser has become a general purpose platform that supports a wide range of mobile Web applications



Customization allows us to find the ideal general-purpose baseline architecture



Hardware/software collaborative specialization leverages application knowledge to mitigate inefficiencies in general-purpose architectures

Thank you