

GOOGLE'S TEN COMMANDMENTS FOR MOBILE COMPUTING: AS OUR WORLD GOES MOBILE, TECH GIANT CALLS FOR BETTER RESEARCH. AT STAKE IS YOUR FUTURE SMARTPHONE.

*Work done during sabbatical @ Google in the Mobile SoC Team**

Vijay Janapa Reddi, Hongil Yoon, Allan Knies
Associate Professor, Harvard University

Infrastructure and Methodology for SoC Performance & Power Modeling
ASPLOS 2019 Workshop, April 13th 2019.

*The thoughts expressed here are the views, thoughts, and opinions expressed in the text belong solely to the author, and not necessarily to the author's employer, organization, committee or other group or individual.



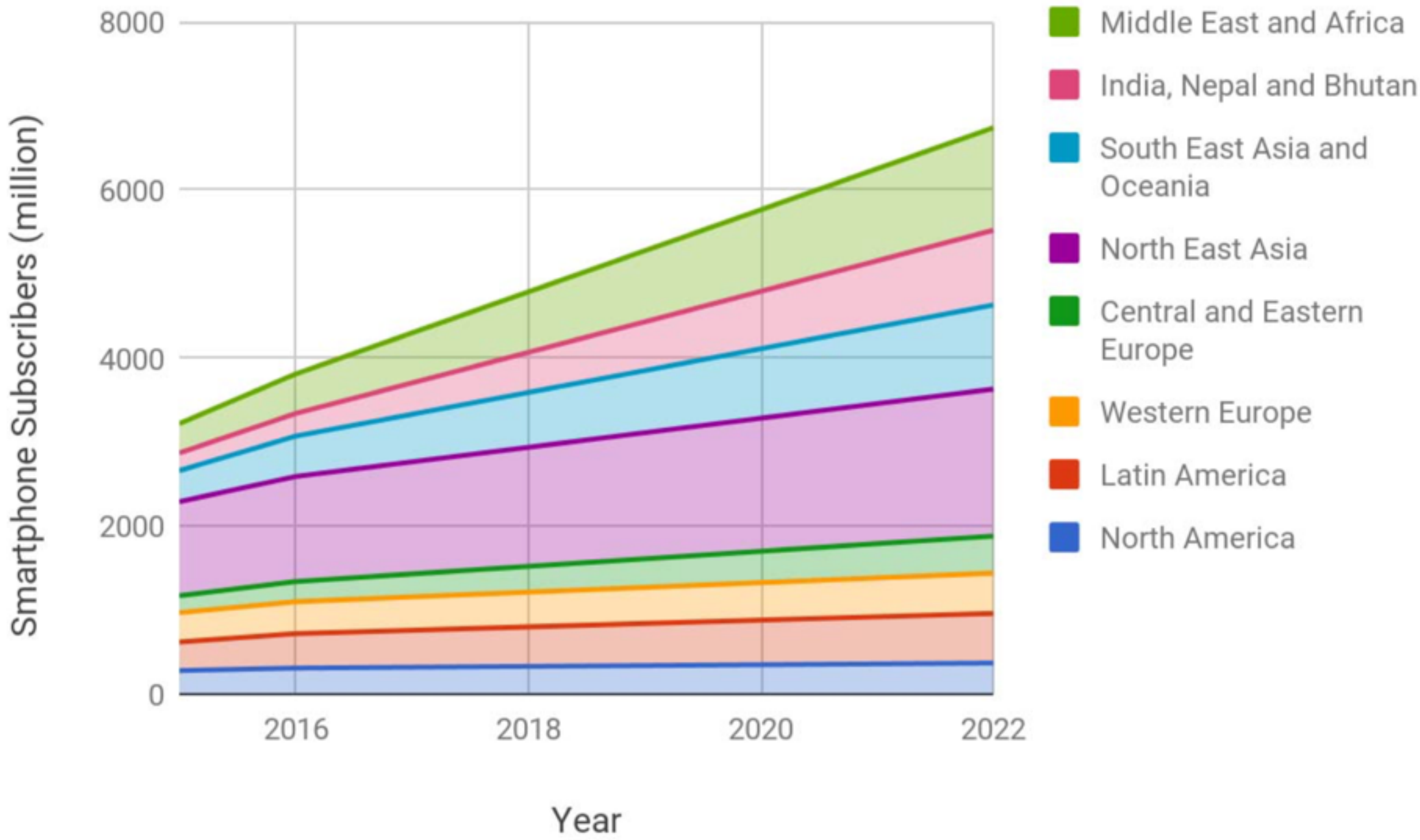
“Why does Google care?”

“Why does Google care?”

How much time do people spend on their mobile phones in 2017?

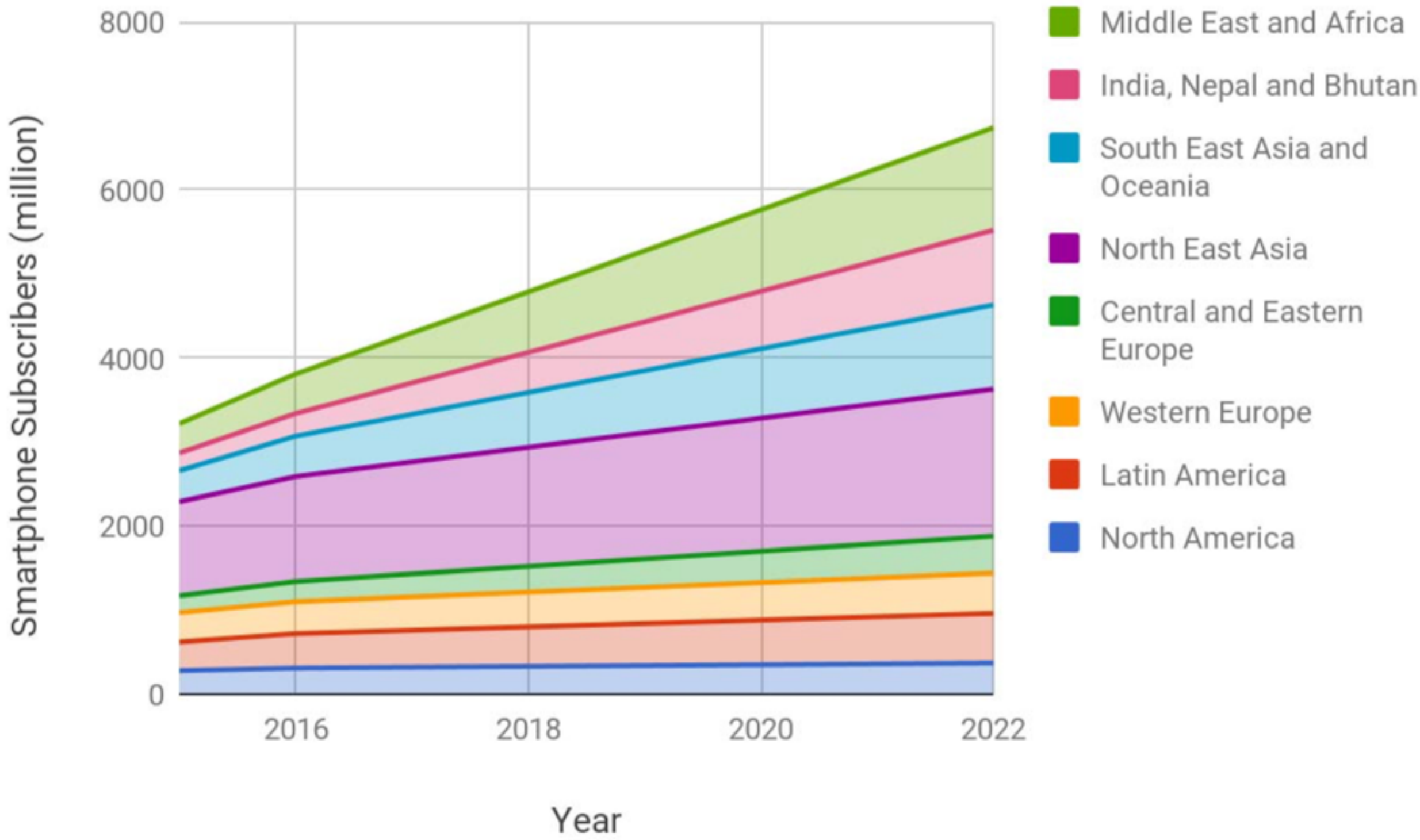
The simple answer is “over 4 hours a day.”

Smartphone Growth Prevails



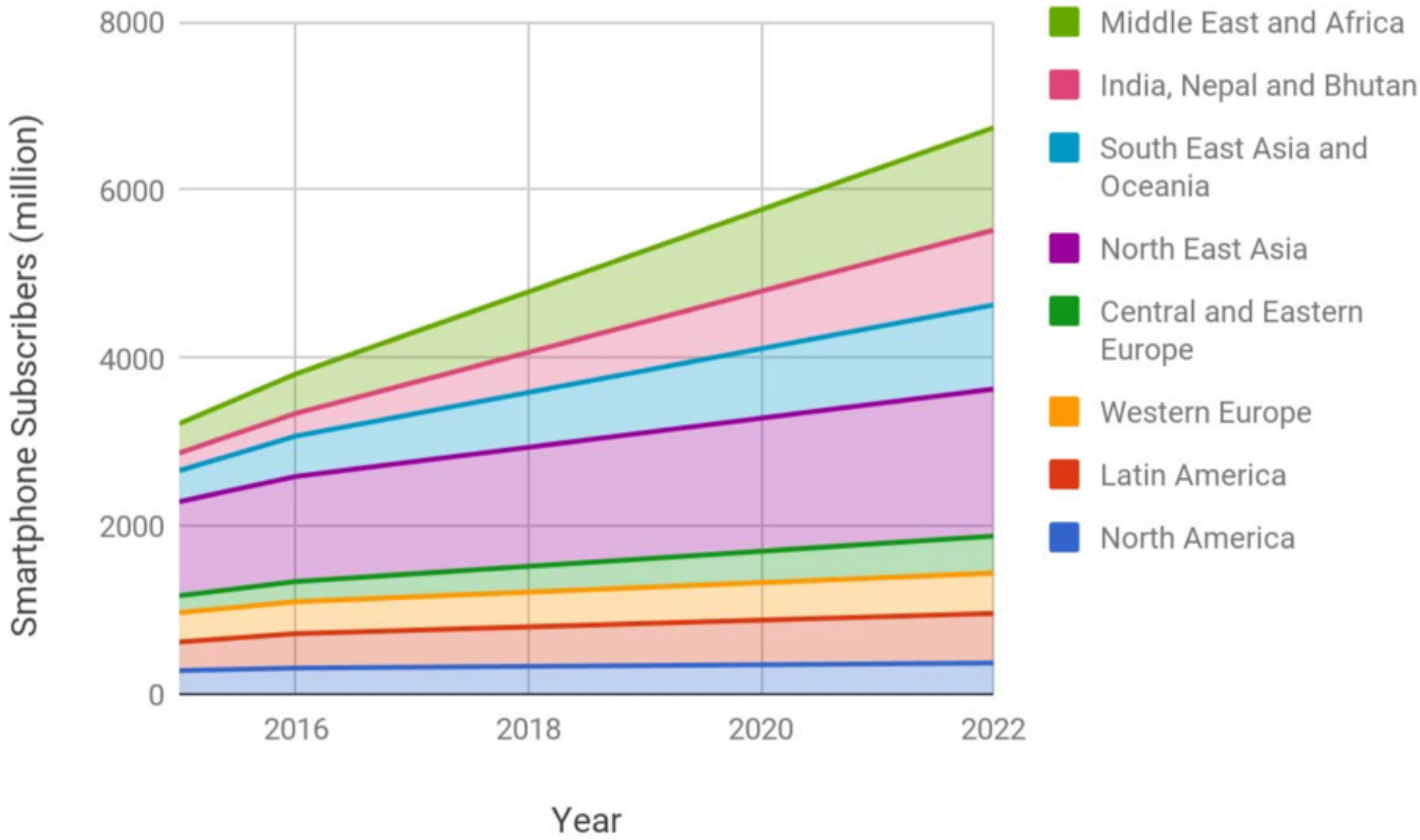
Smartphone Growth Prevails

- The number of smartphone subscribers continues to grow.



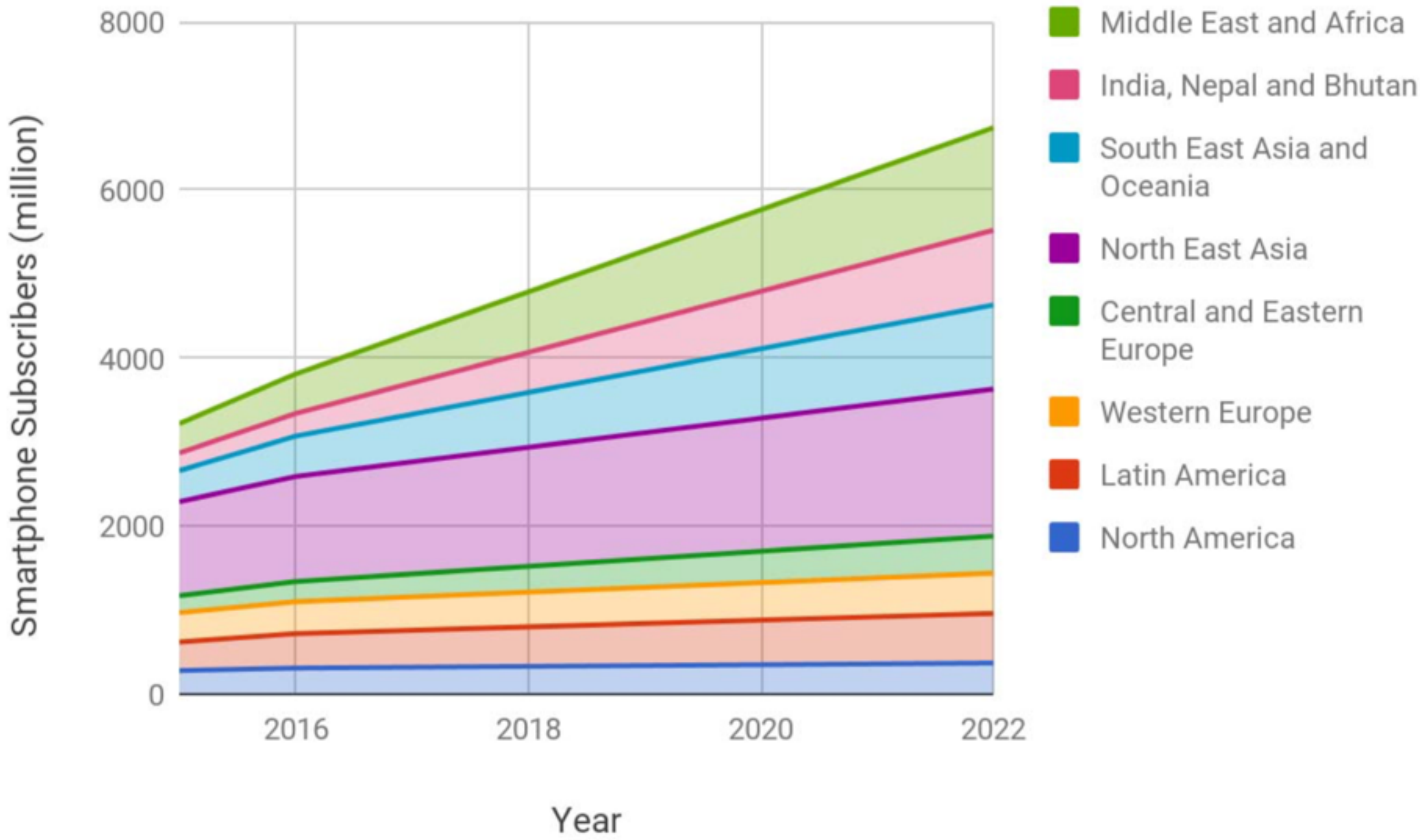
Smartphone Growth Prevails

- The number of smartphone subscribers continues to grow.
- The smartphone is a gateway into user's lives and their data.



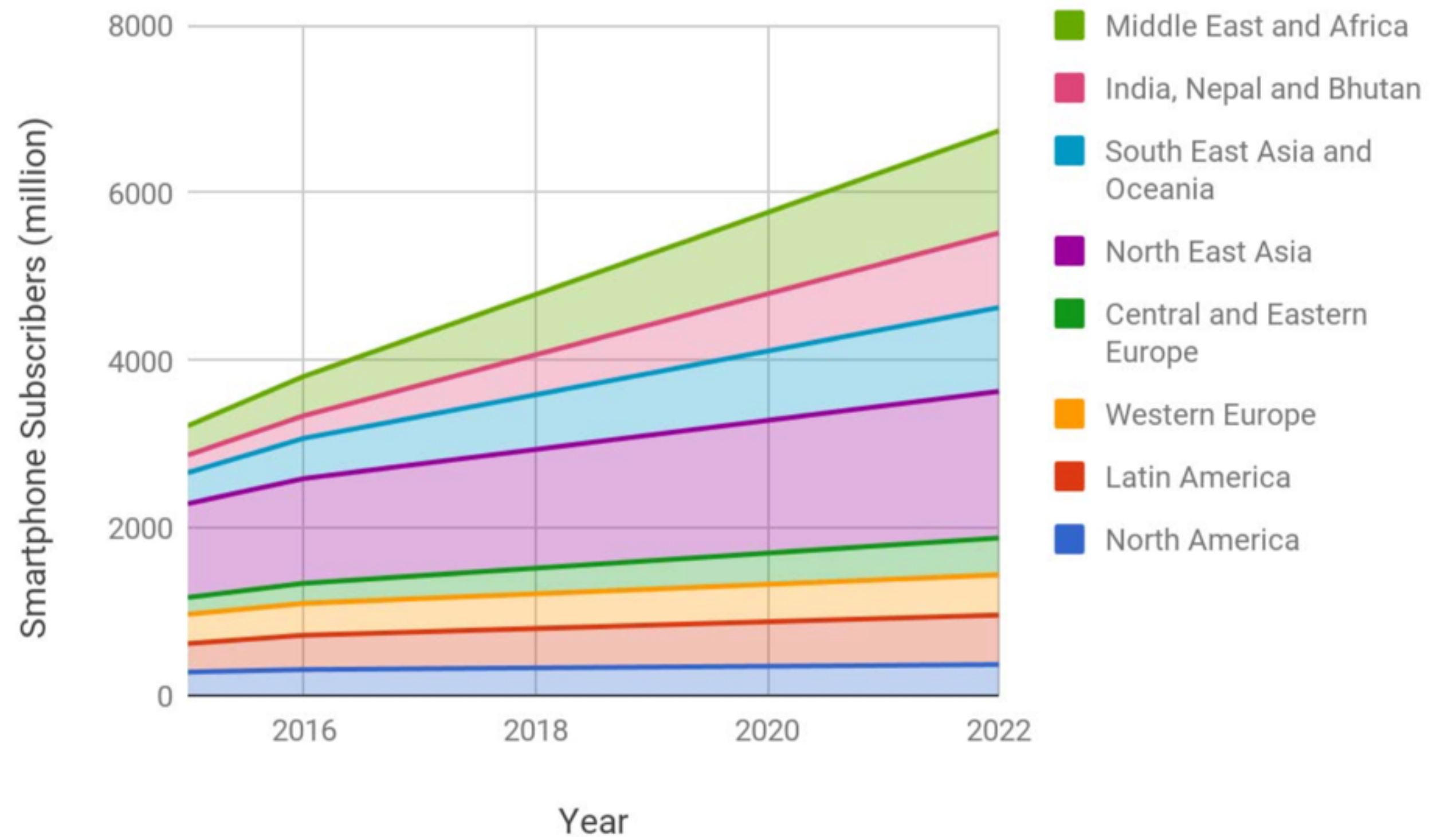
Smartphone Growth Prevails

- The number of smartphone subscribers continues to grow.
- The smartphone is a gateway into user's lives and their data.
- To gain access to this valuable data you need to have differentiating features.



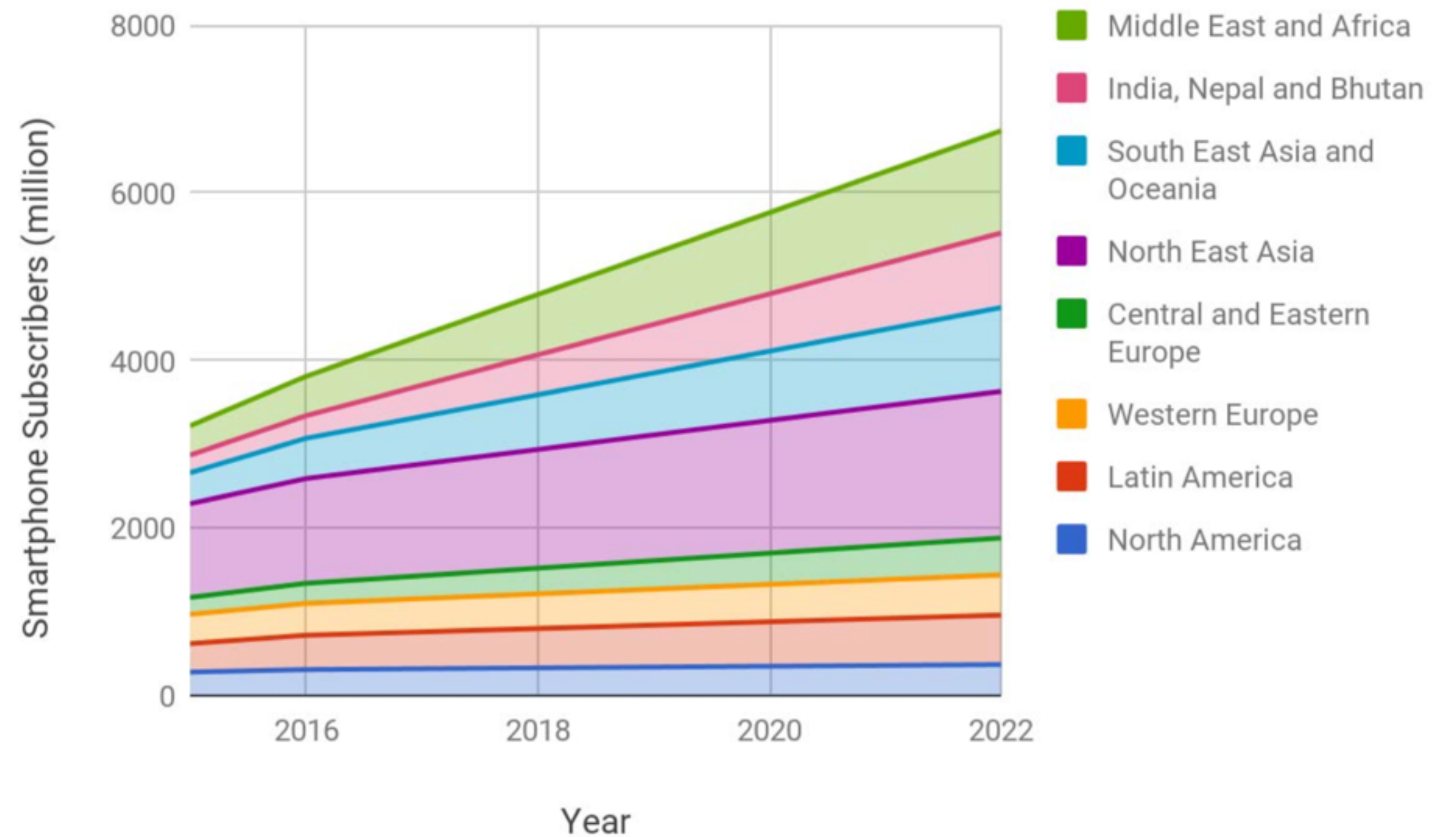
Smartphone Growth Prevails

- The number of smartphone subscribers continues to grow.
- The smartphone is a gateway into user's lives and their data.
- To gain access to this valuable data you need to have differentiating features.
- Providing those differentiating features requires you to innovate at the hardware

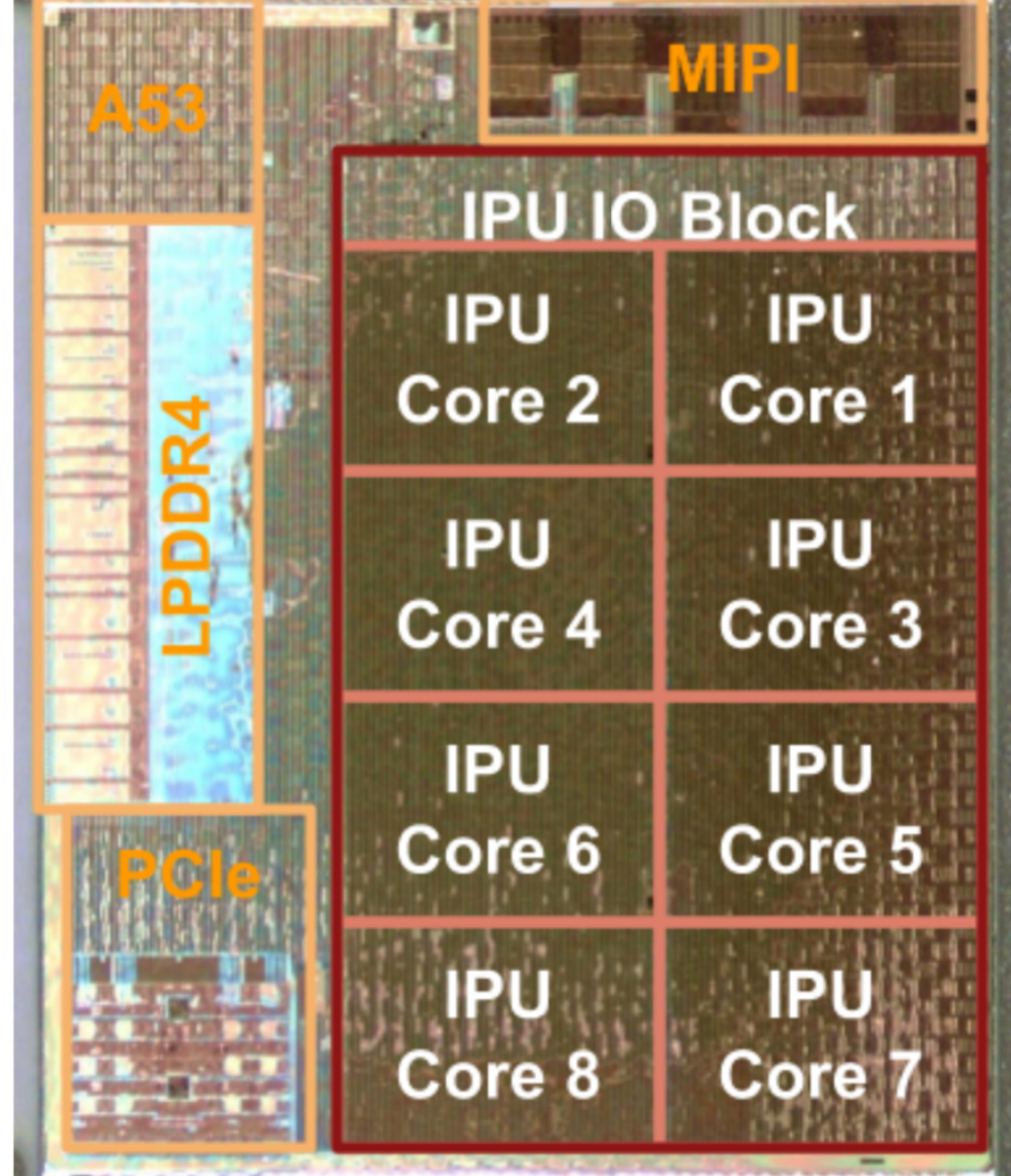


Smartphone Growth Prevails

- The number of smartphone subscribers continues to grow.
- The smartphone is a gateway into user's lives and their data.
- To gain access to this valuable data you need to have differentiating features.
- Providing those differentiating features requires you to innovate at the hardware
- Provide an integrated solution that can provide a unique quality of experience

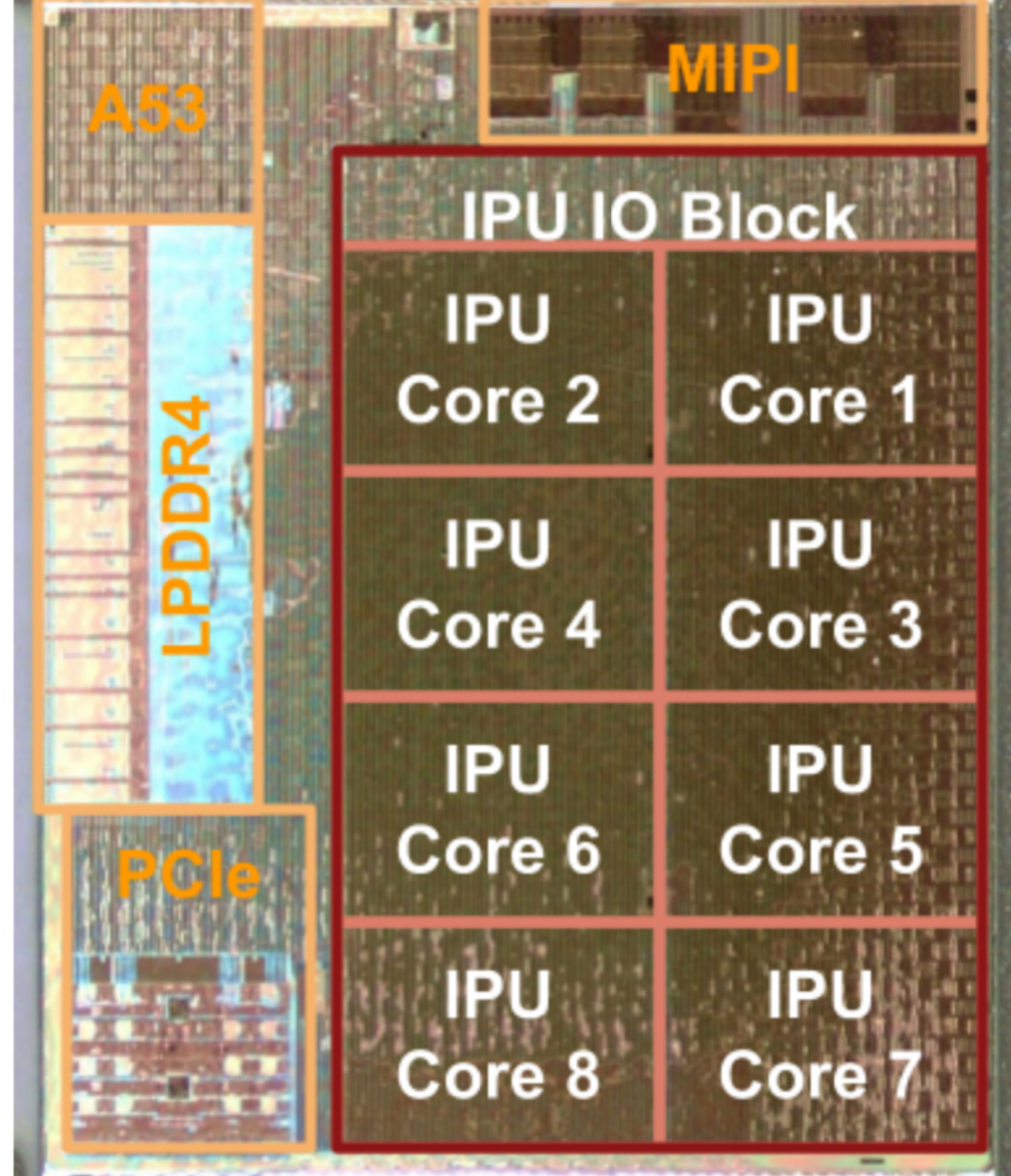


A New Beginning - The Pixel Visual Core from Google



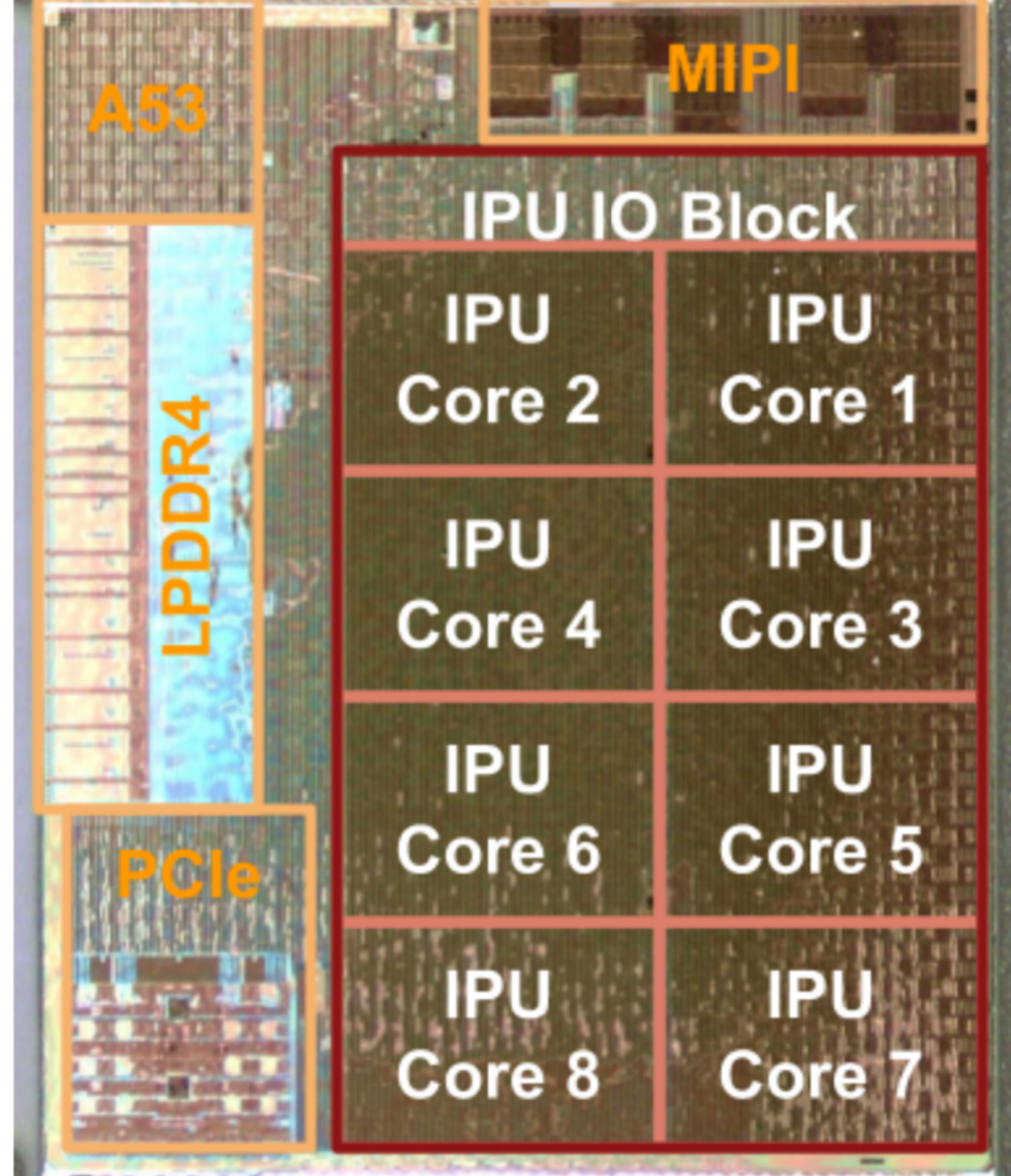
A New Beginning - The Pixel Visual Core from Google

- The camera on the new Pixel 2 is packed full of great hardware, software and machine learning (ML), so all you need to do is point and shoot to take amazing photos and videos.



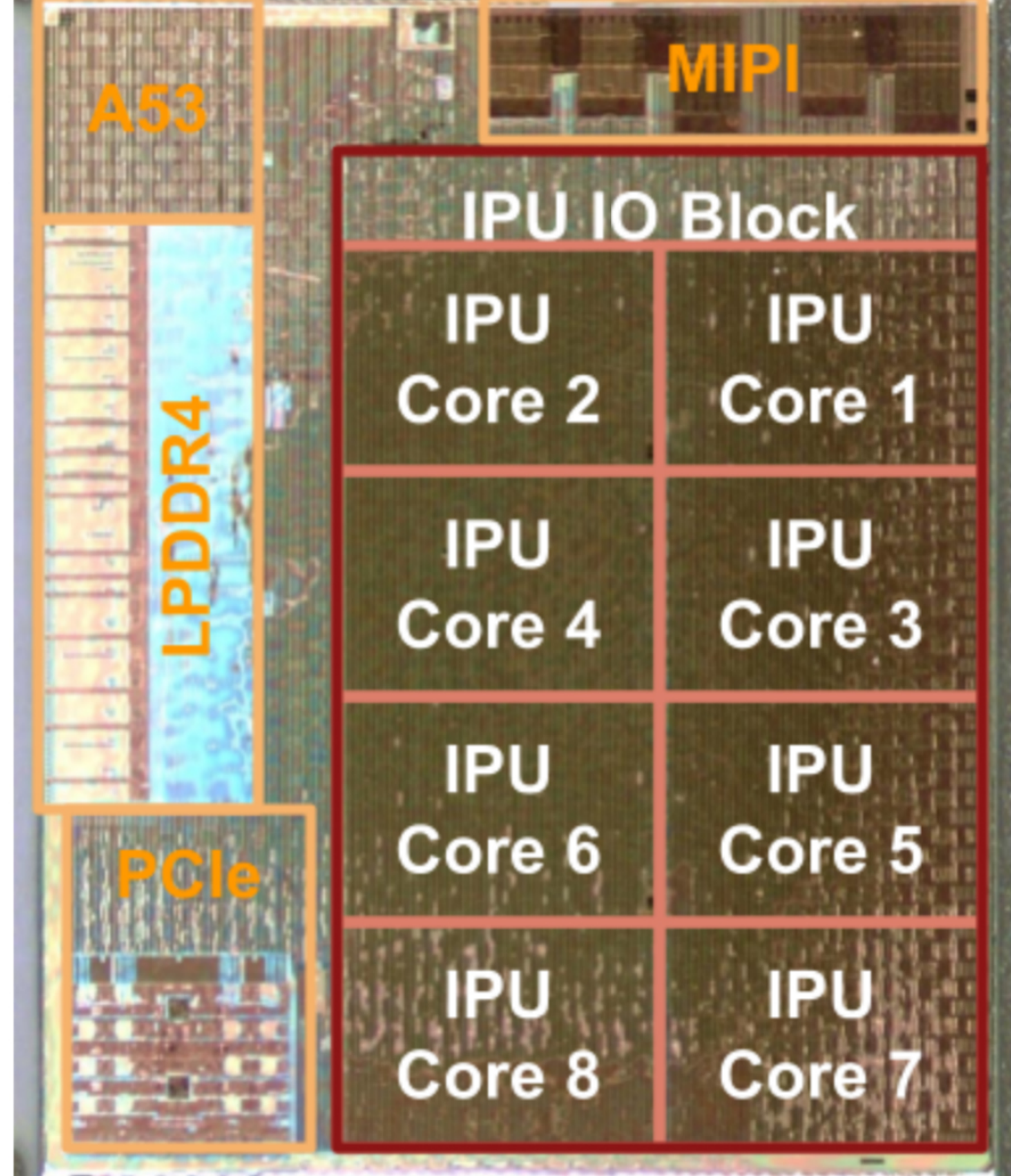
A New Beginning - The Pixel Visual Core from Google

- The camera on the new Pixel 2 is packed full of great hardware, software and machine learning (ML), so all you need to do is point and shoot to take amazing photos and videos.
- HDR+ produces beautiful images, and they evolved the algorithm that powers it to use the Pixel 2's processor efficiently, and enable you to take multiple pictures in sequence by intelligently processing HDR+ in the background.



A New Beginning - The Pixel Visual Core from Google

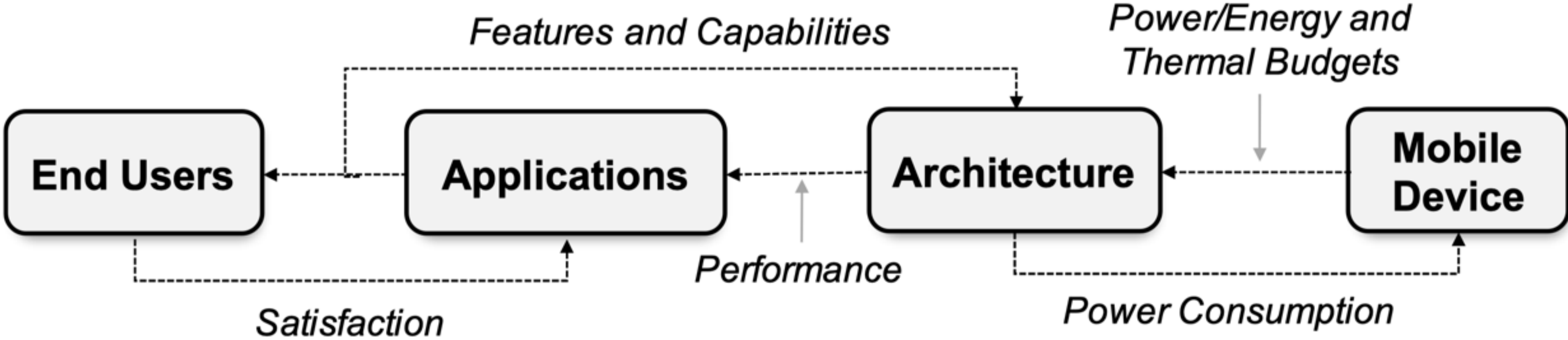
- The camera on the new Pixel 2 is packed full of great hardware, software and machine learning (ML), so all you need to do is point and shoot to take amazing photos and videos.
- HDR+ produces beautiful images, and they evolved the algorithm that powers it to use the Pixel 2's processor efficiently, and enable you to take multiple pictures in sequence by intelligently processing HDR+ in the background.
- HDR+ will be the first application to run on Pixel Visual Core. Notably, because Pixel Visual Core is programmable, and they are preparing the next set of applications. The great thing is that as they port more machine learning and imaging applications to use Pixel Visual Core, Pixel 2 will continuously improve. So keep an eye out!







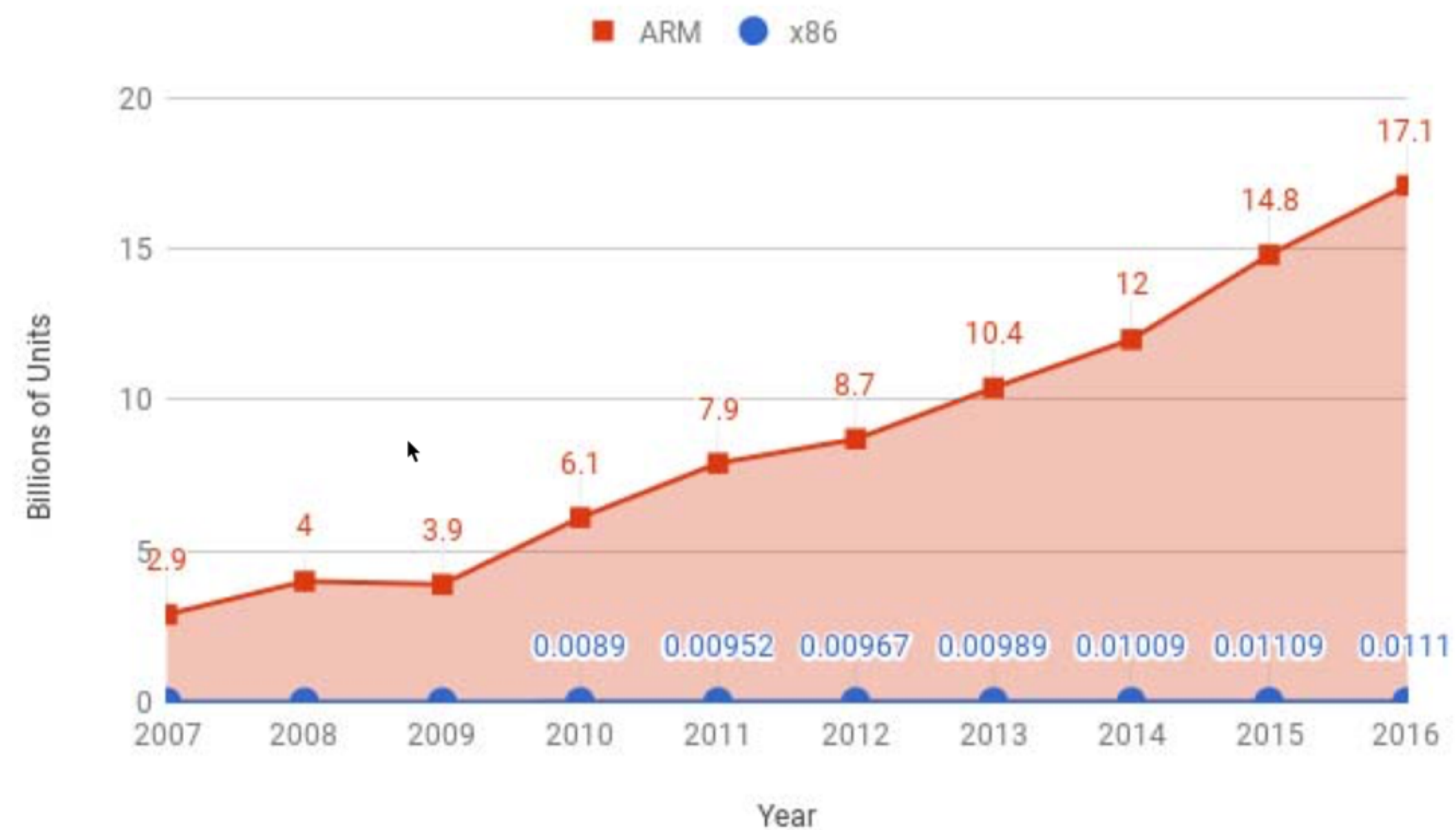
The Mobile Device Virtuous (Vicious) Cycle



Thinking of the Mobile Hardware Ecosystem at Large

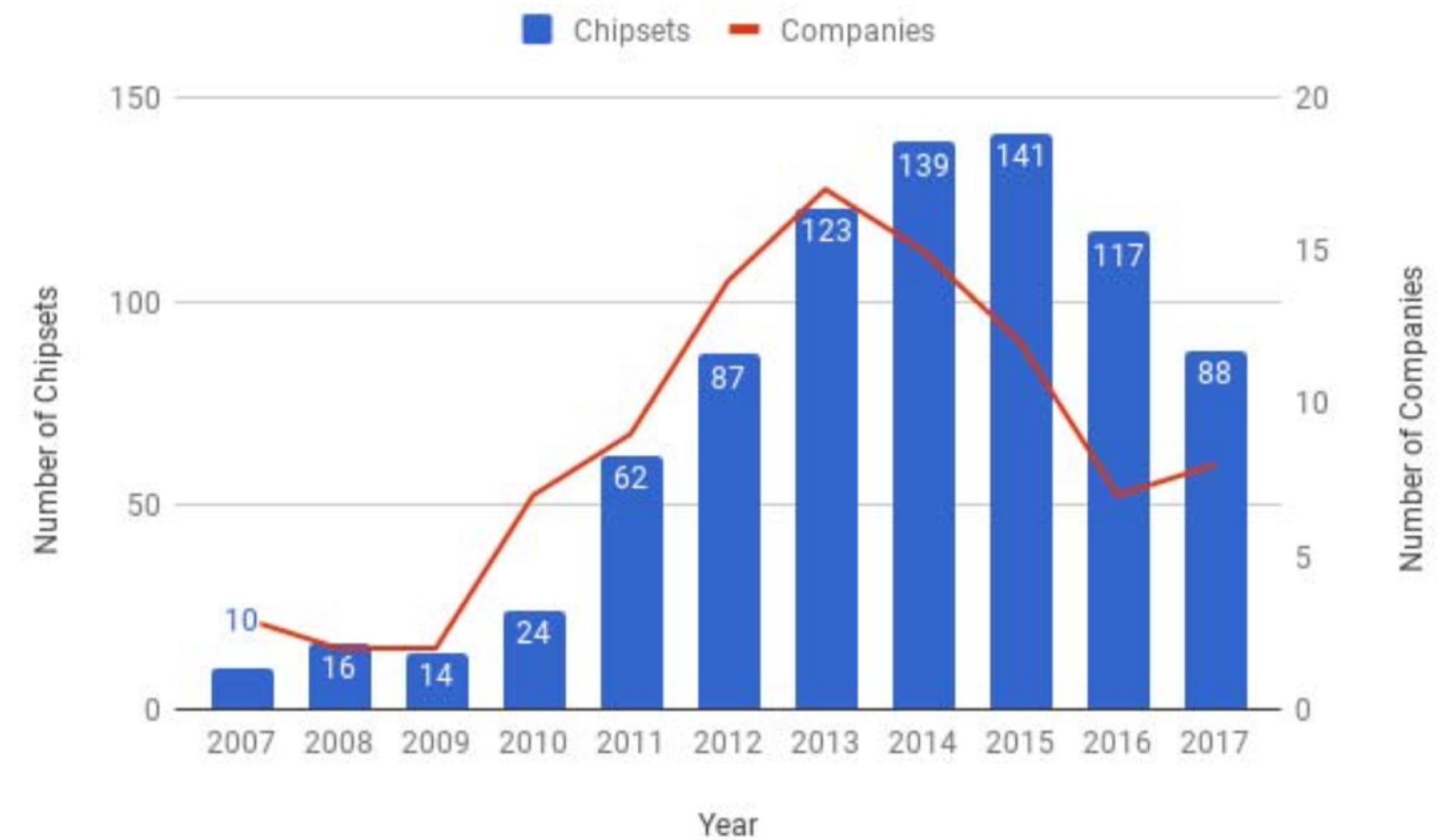
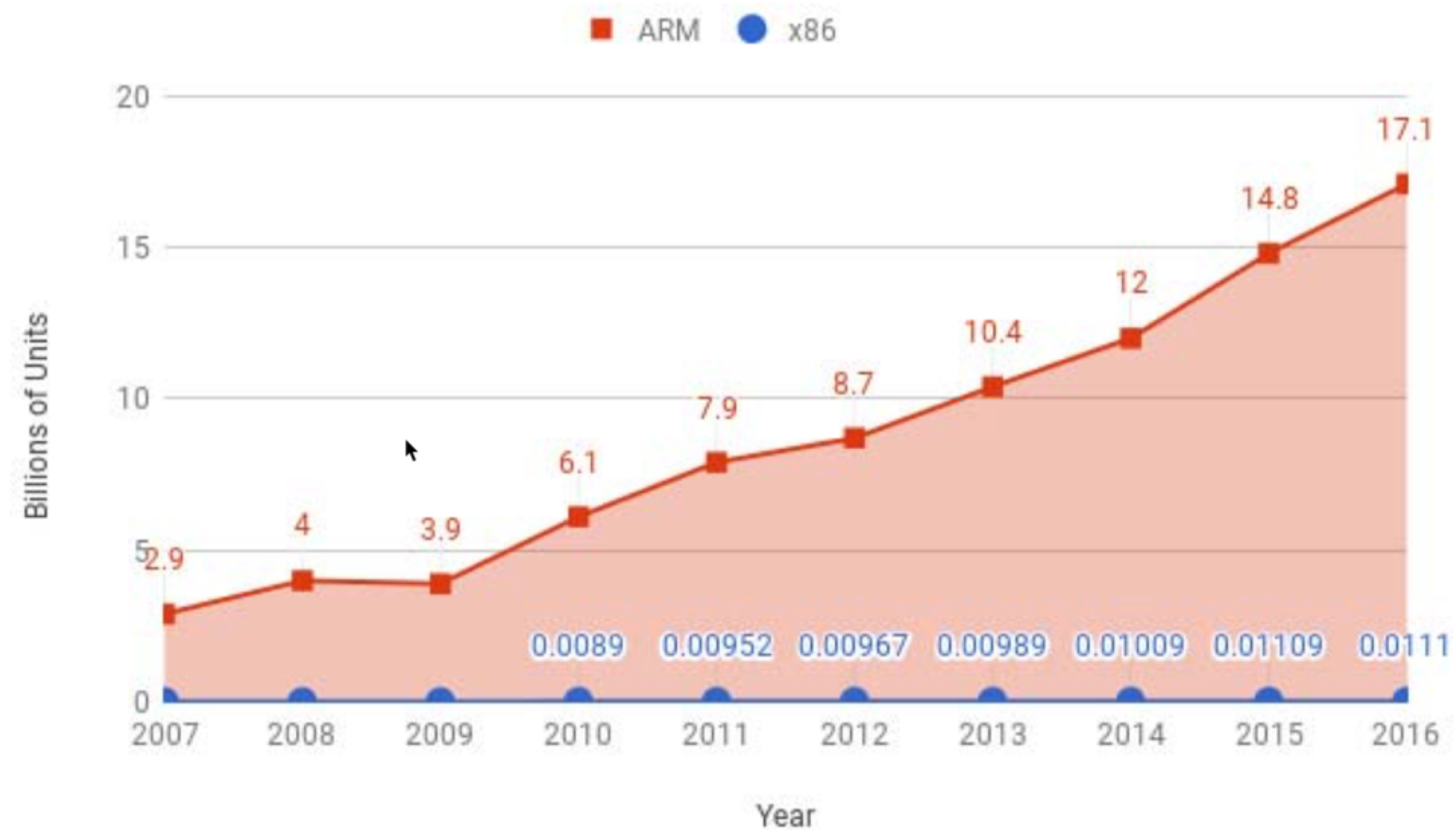


Thinking of the Mobile Hardware Ecosystem at Large



- Driven the ability to adopt ARM ISA easily and integrate to create custom chipsets.

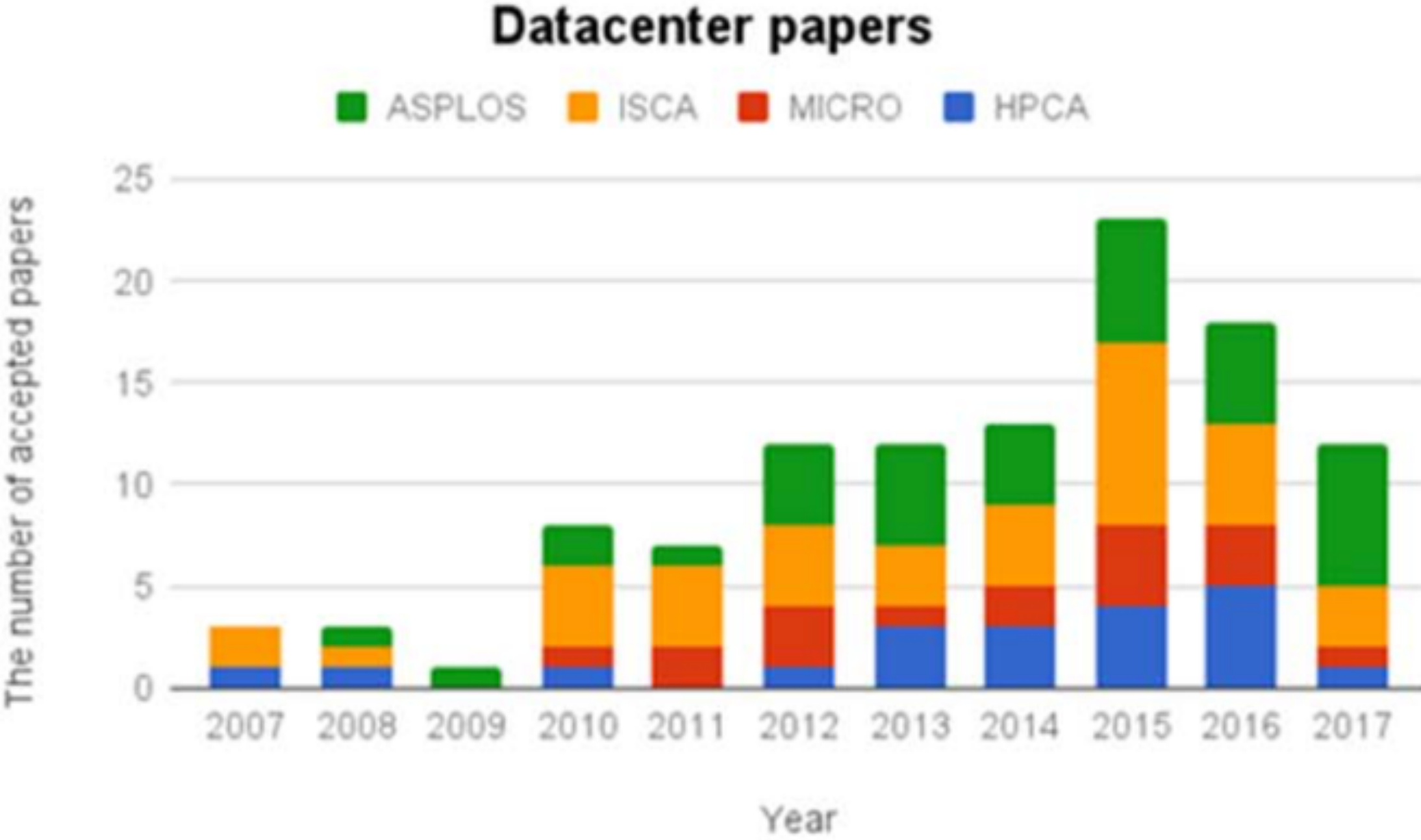
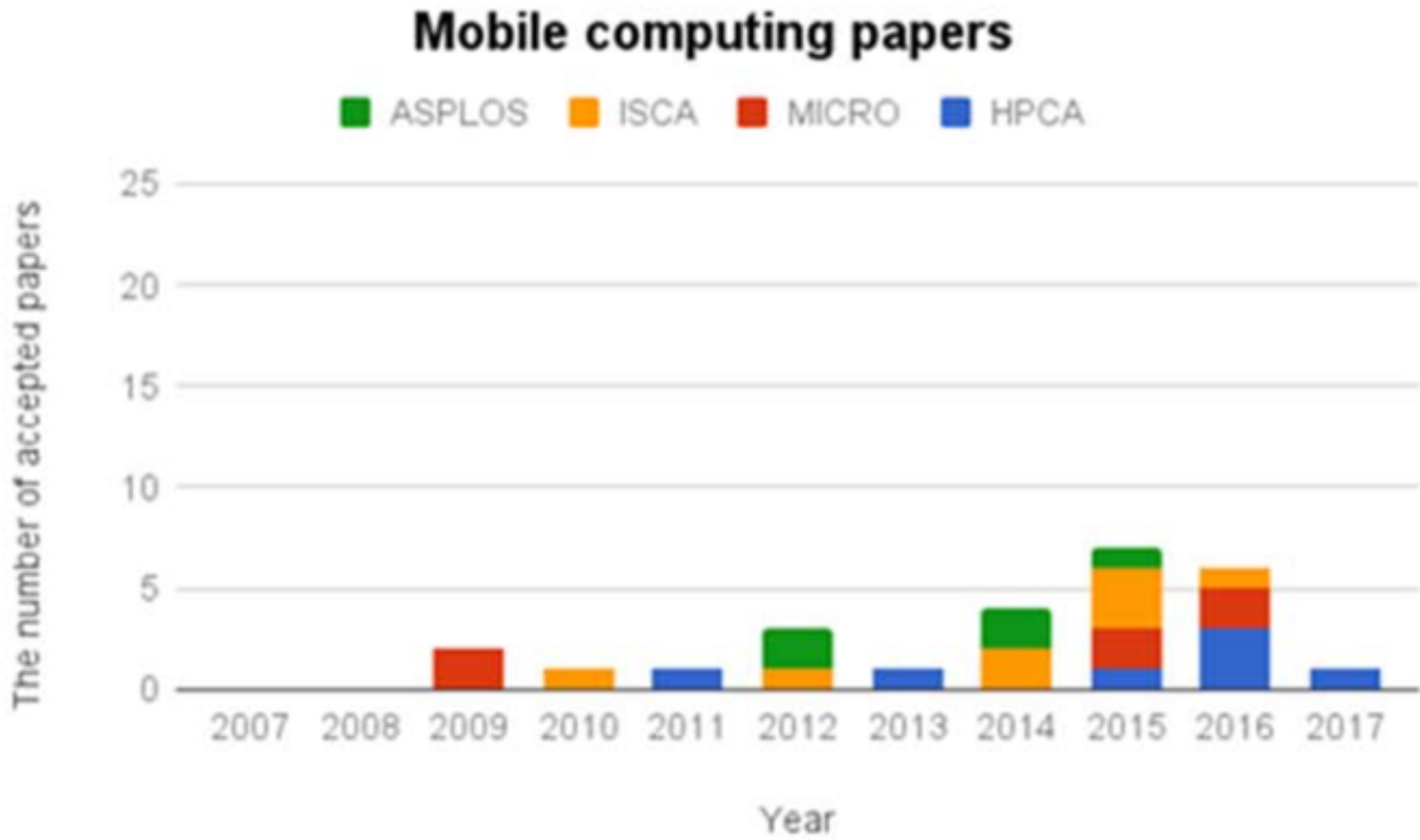
Thinking of the Mobile Hardware Ecosystem at Large



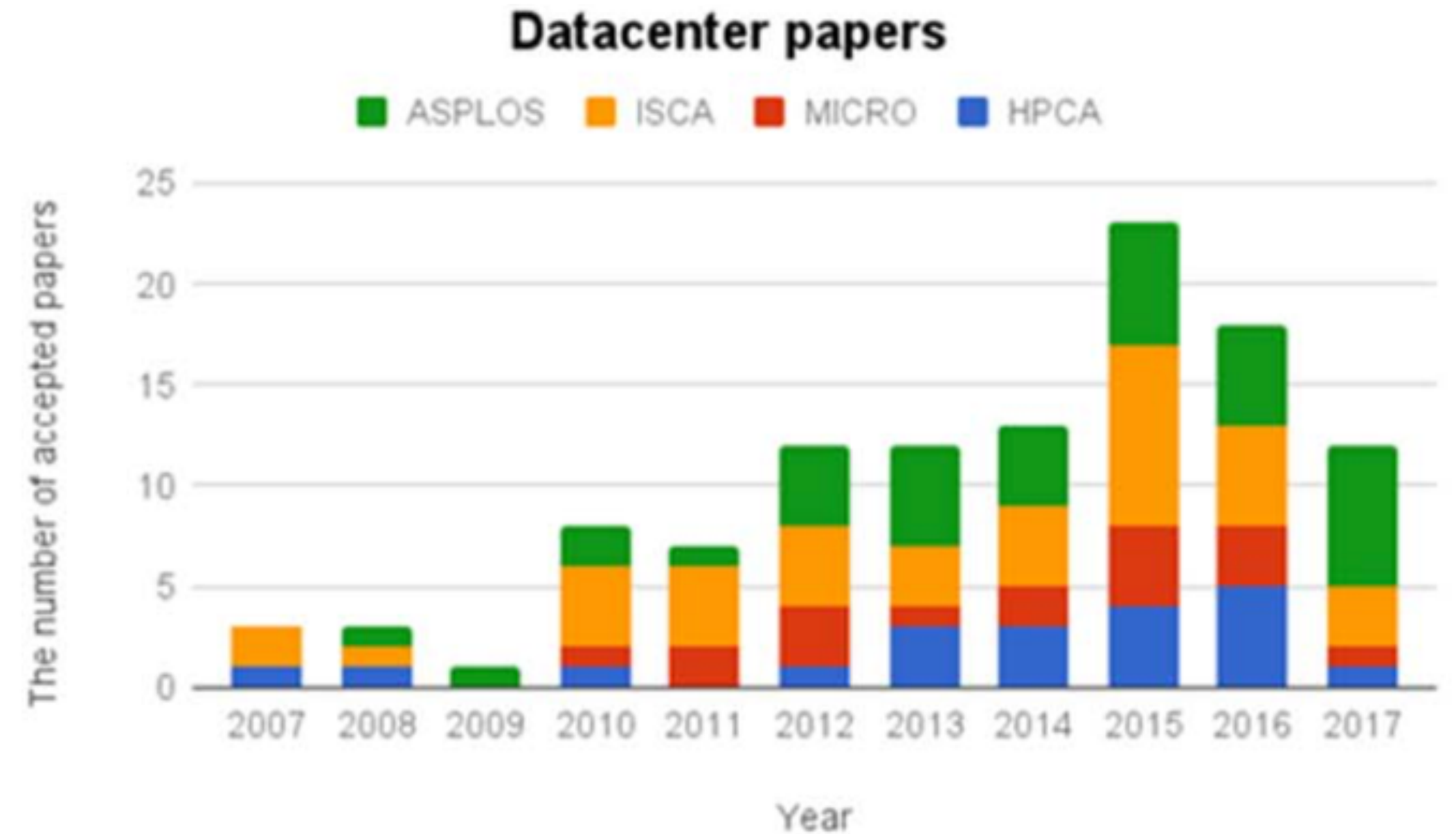
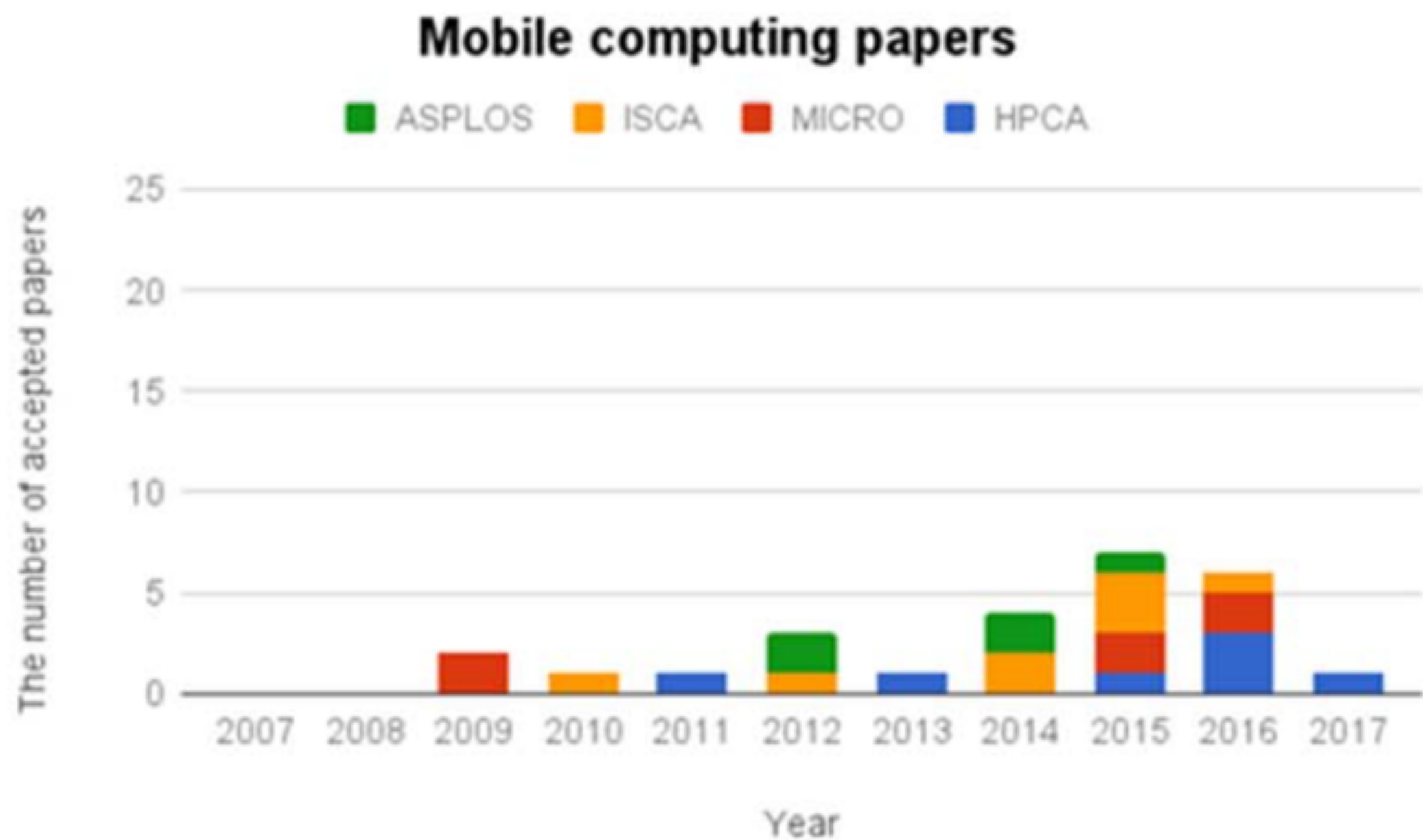
- Driven the ability to adopt ARM ISA easily and integrate to create custom chipsets.
- Lead to the widespread diversification of the chipset market for the mobile processor community.



The Dearth of Mobile Computer Architecture Research



The Dearth of Mobile Computer Architecture Research



- Mobile computer architecture is severely lagging behind in the community
- Mobile systems are highly advanced and consist of many processing engines



The Ten Commandments of Mobile Computer Architecture



The Ten Commandments of Mobile Computer Architecture

1. Thou shalt not rely solely on benchmarks. (§1)

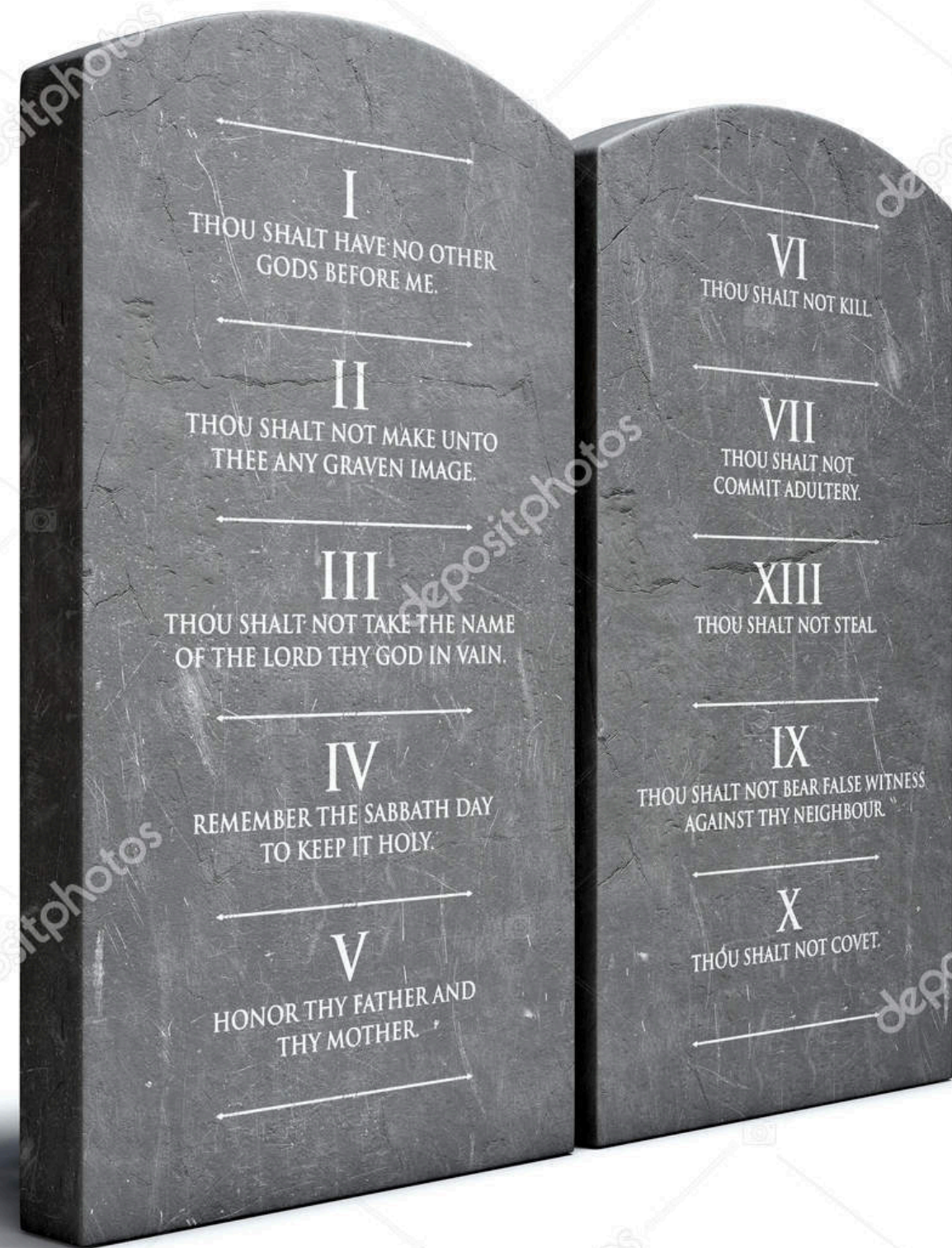


The Ten Commandments of Mobile Computer Architecture

1. Thou shalt not rely solely on benchmarks. (§1)
2. Thou shalt not cherry-pick applications. (§1)



The Ten Commandments of Mobile Computer Architecture



1. Thou shalt not rely solely on benchmarks. (§1)
2. Thou shalt not cherry-pick applications. (§1)
3. Thou shalt not ignore the web browser. (§1)

The Ten Commandments of Mobile Computer Architecture



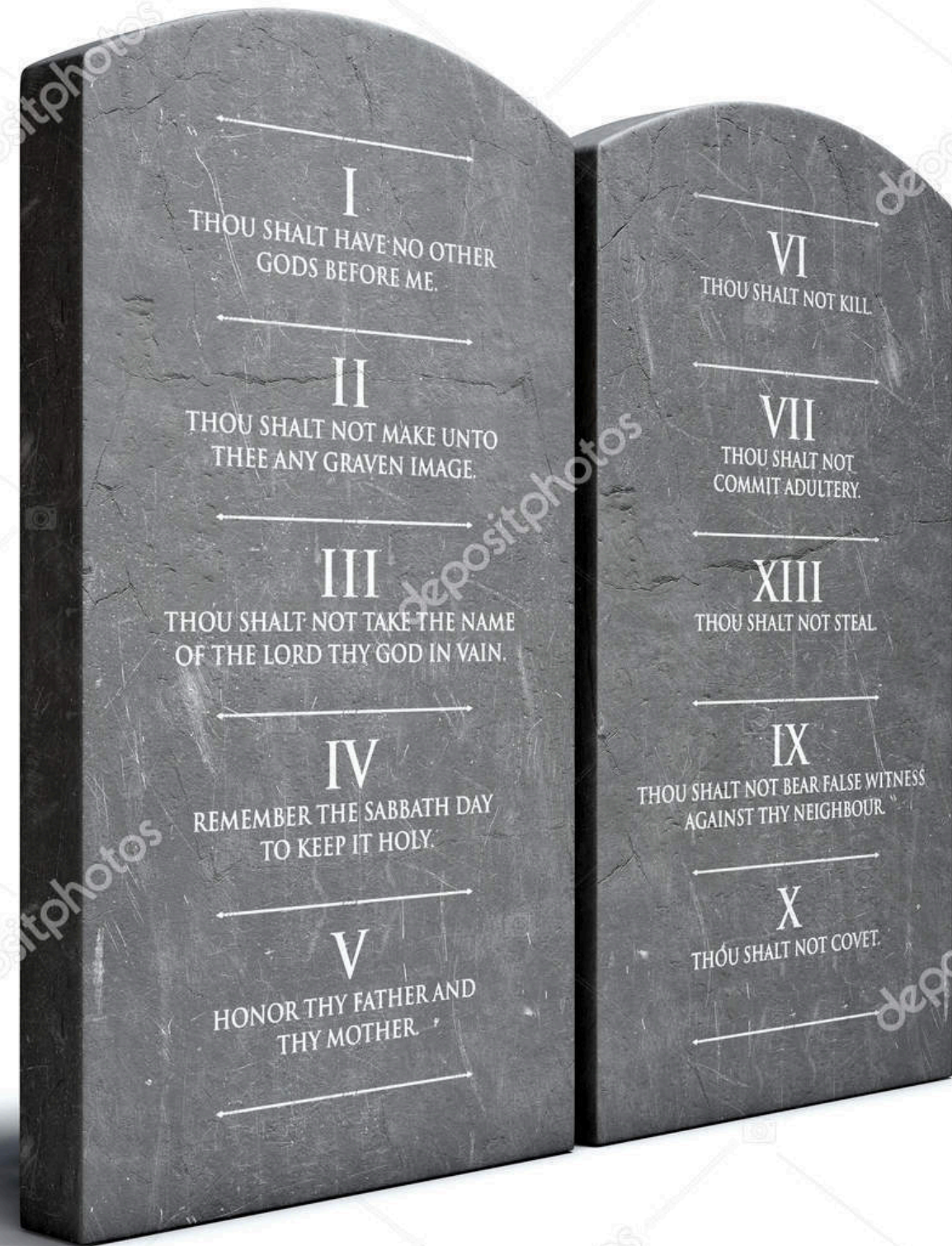
1. Thou shalt not rely solely on benchmarks. (§1)
2. Thou shalt not cherry-pick applications. (§1)
3. Thou shalt not ignore the web browser. (§1)
4. Thou shalt not drop a frame. (§2)

The Ten Commandments of Mobile Computer Architecture



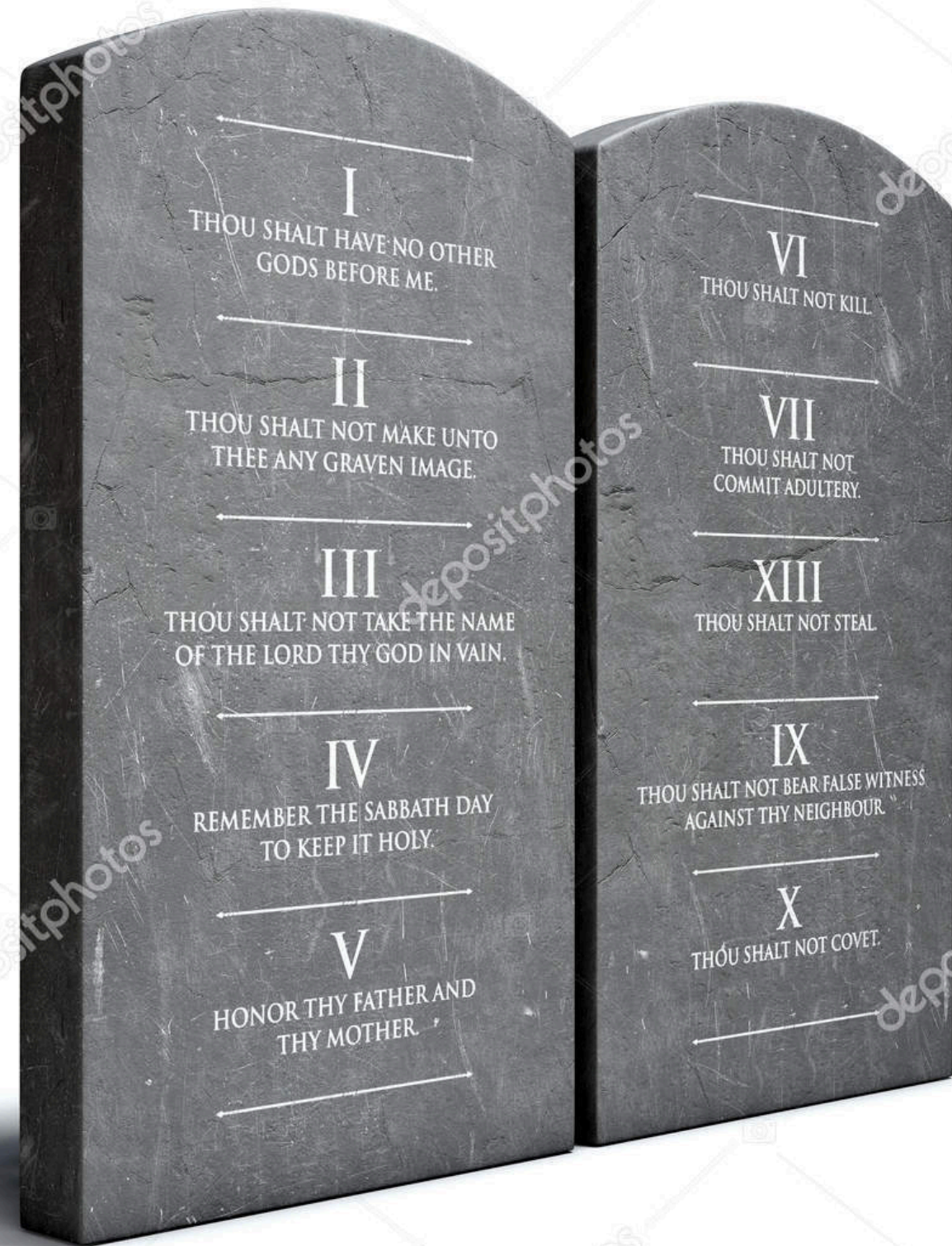
1. Thou shalt not rely solely on benchmarks. (§1)
2. Thou shalt not cherry-pick applications. (§1)
3. Thou shalt not ignore the web browser. (§1)
4. Thou shalt not drop a frame. (§2)
5. Thou shalt not idolize micro-architectural efficiency. (§2)

The Ten Commandments of Mobile Computer Architecture



1. Thou shalt not rely solely on benchmarks. (§1)
2. Thou shalt not cherry-pick applications. (§1)
3. Thou shalt not ignore the web browser. (§1)
4. Thou shalt not drop a frame. (§2)
5. Thou shalt not idolize micro-architectural efficiency. (§2)
6. Thou shalt not ignore tails in user experience. (§2)

The Ten Commandments of Mobile Computer Architecture



1. Thou shalt not rely solely on benchmarks. (§1)
2. Thou shalt not cherry-pick applications. (§1)
3. Thou shalt not ignore the web browser. (§1)
4. Thou shalt not drop a frame. (§2)
5. Thou shalt not idolize micro-architectural efficiency. (§2)
6. Thou shalt not ignore tails in user experience. (§2)
7. Thou shalt not assume software is hardware-agnostic. (§3)

The Ten Commandments of Mobile Computer Architecture



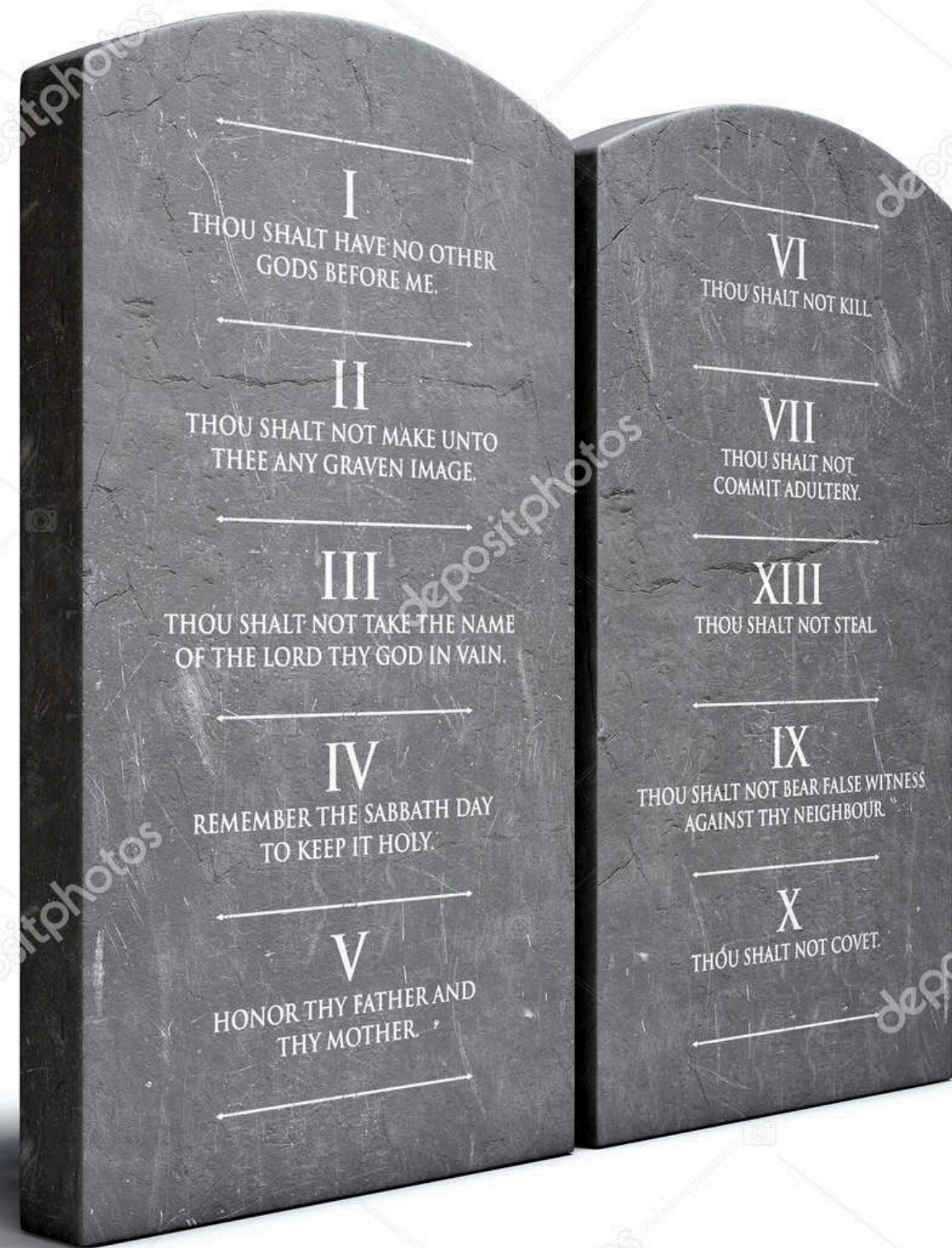
1. Thou shalt not rely solely on benchmarks. (§1)
2. Thou shalt not cherry-pick applications. (§1)
3. Thou shalt not ignore the web browser. (§1)
4. Thou shalt not drop a frame. (§2)
5. Thou shalt not idolize micro-architectural efficiency. (§2)
6. Thou shalt not ignore tails in user experience. (§2)
7. Thou shalt not assume software is hardware-agnostic. (§3)
8. Thou shalt not disregard the IP blocks. (§3)

The Ten Commandments of Mobile Computer Architecture



1. Thou shalt not rely solely on benchmarks. (§1)
2. Thou shalt not cherry-pick applications. (§1)
3. Thou shalt not ignore the web browser. (§1)
4. Thou shalt not drop a frame. (§2)
5. Thou shalt not idolize micro-architectural efficiency. (§2)
6. Thou shalt not ignore tails in user experience. (§2)
7. Thou shalt not assume software is hardware-agnostic. (§3)
8. Thou shalt not disregard the IP blocks. (§3)
9. Thou shalt not turn a blind eye to energy and thermal. (§3)

The Ten Commandments of Mobile Computer Architecture



1. Thou shalt not rely solely on benchmarks. (§1)
2. Thou shalt not cherry-pick applications. (§1)
3. Thou shalt not ignore the web browser. (§1)
4. Thou shalt not drop a frame. (§2)
5. Thou shalt not idolize micro-architectural efficiency. (§2)
6. Thou shalt not ignore tails in user experience. (§2)
7. Thou shalt not assume software is hardware-agnostic. (§3)
8. Thou shalt not disregard the IP blocks. (§3)
9. Thou shalt not turn a blind eye to energy and thermal. (§3)
10. Thou shalt not presume this list is complete. (§1,§2,§3)

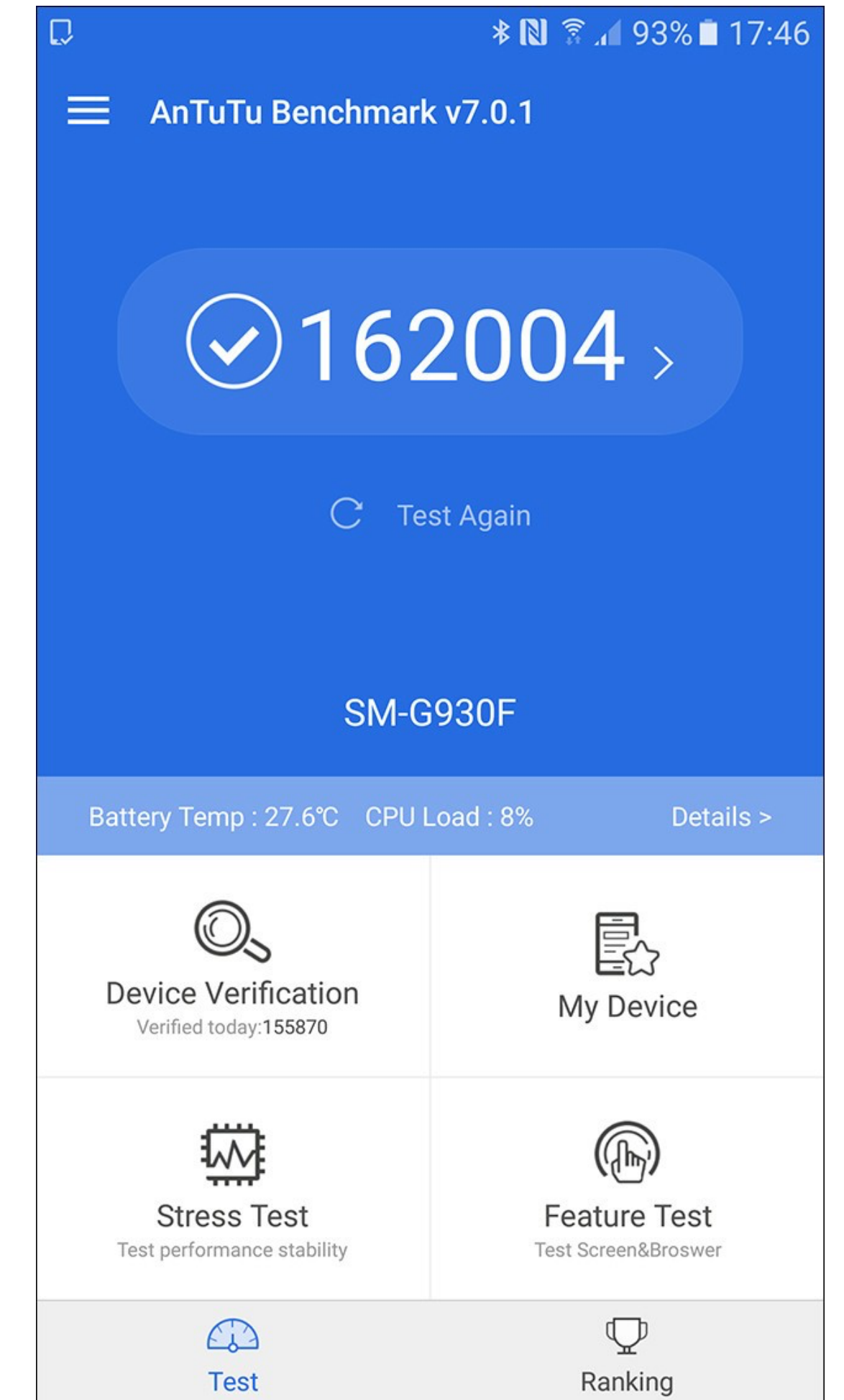
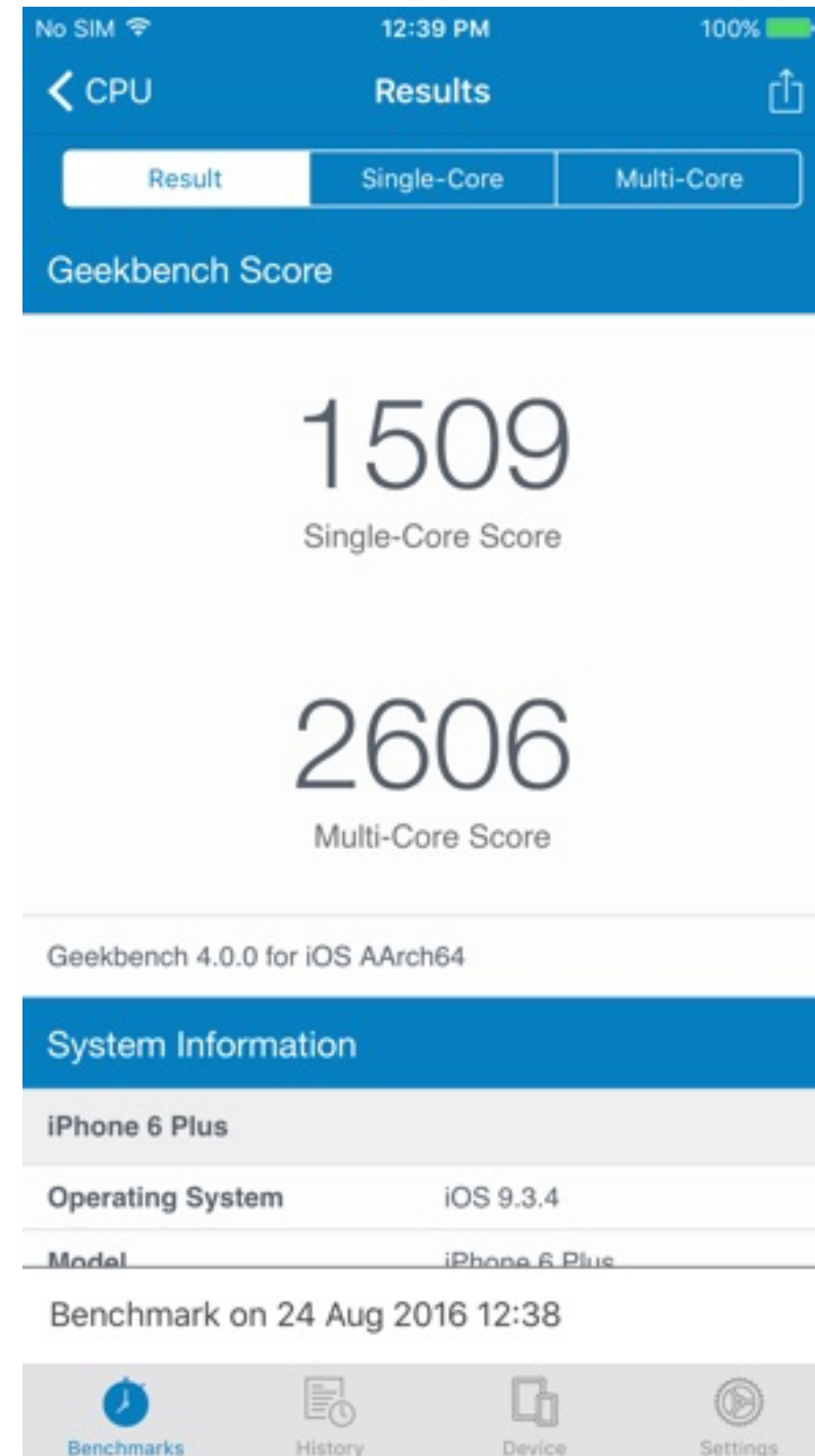
1. Thou shalt not rely solely on benchmarks.

1. Thou shalt not rely solely on benchmarks.

Benchmarks like Geekbench and AnTuTu focus on core computational kernels (such as Dijkstra, JPEG compression, GEMM, and FFT) that exercise the CPU and GPU.

1. Thou shalt not rely solely on benchmarks.

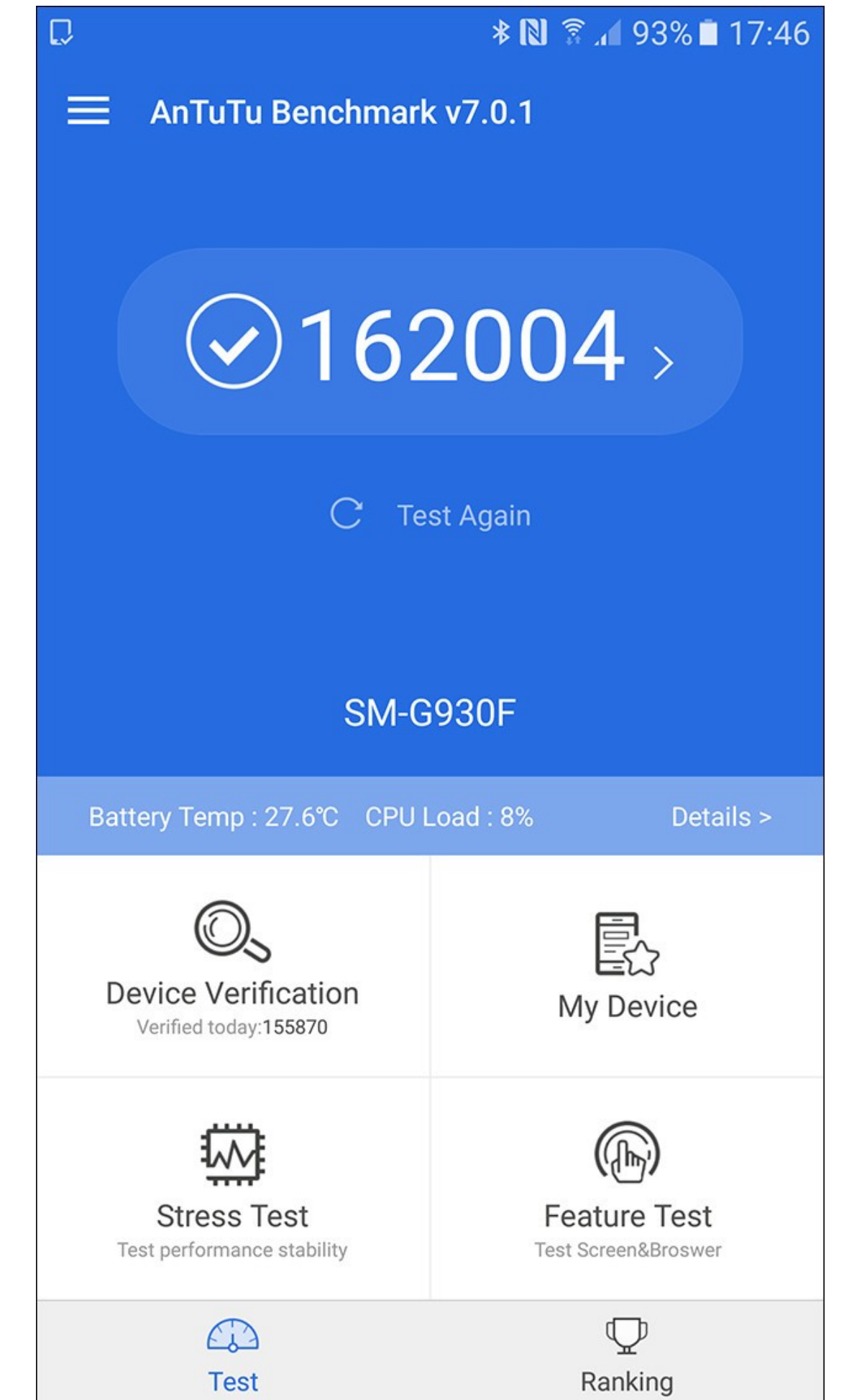
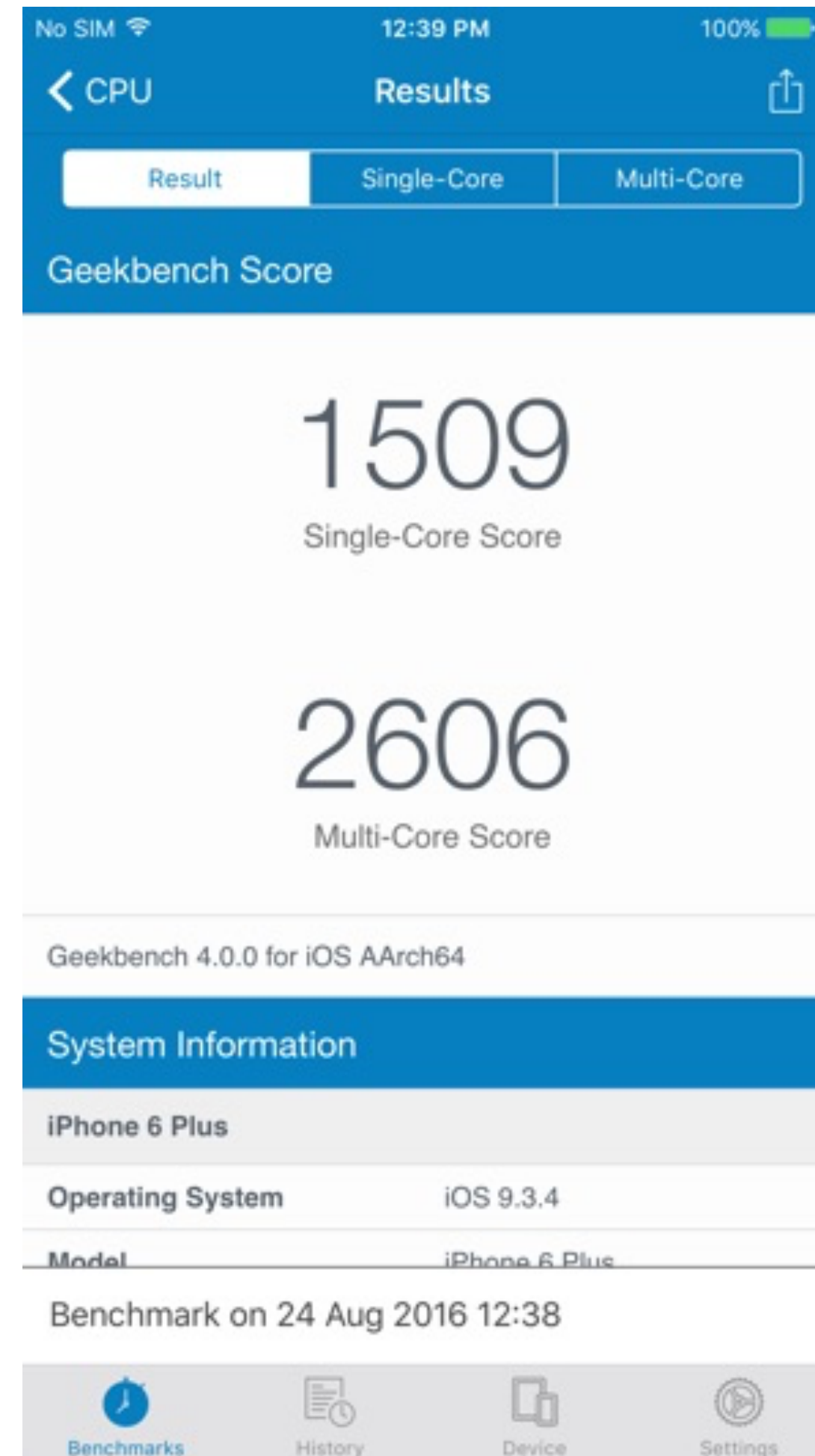
Benchmarks like Geekbench and AnTuTu focus on core computational kernels (such as Dijkstra, JPEG compression, GEMM, and FFT) that exercise the CPU and GPU.



1. Thou shalt not rely solely on benchmarks.

Benchmarks like Geekbench and AnTuTu focus on core computational kernels (such as Dijkstra, JPEG compression, GEMM, and FFT) that exercise the CPU and GPU.

These benchmarks continuously stress the component, rather than mimic the intermitted nature of mobile usage.

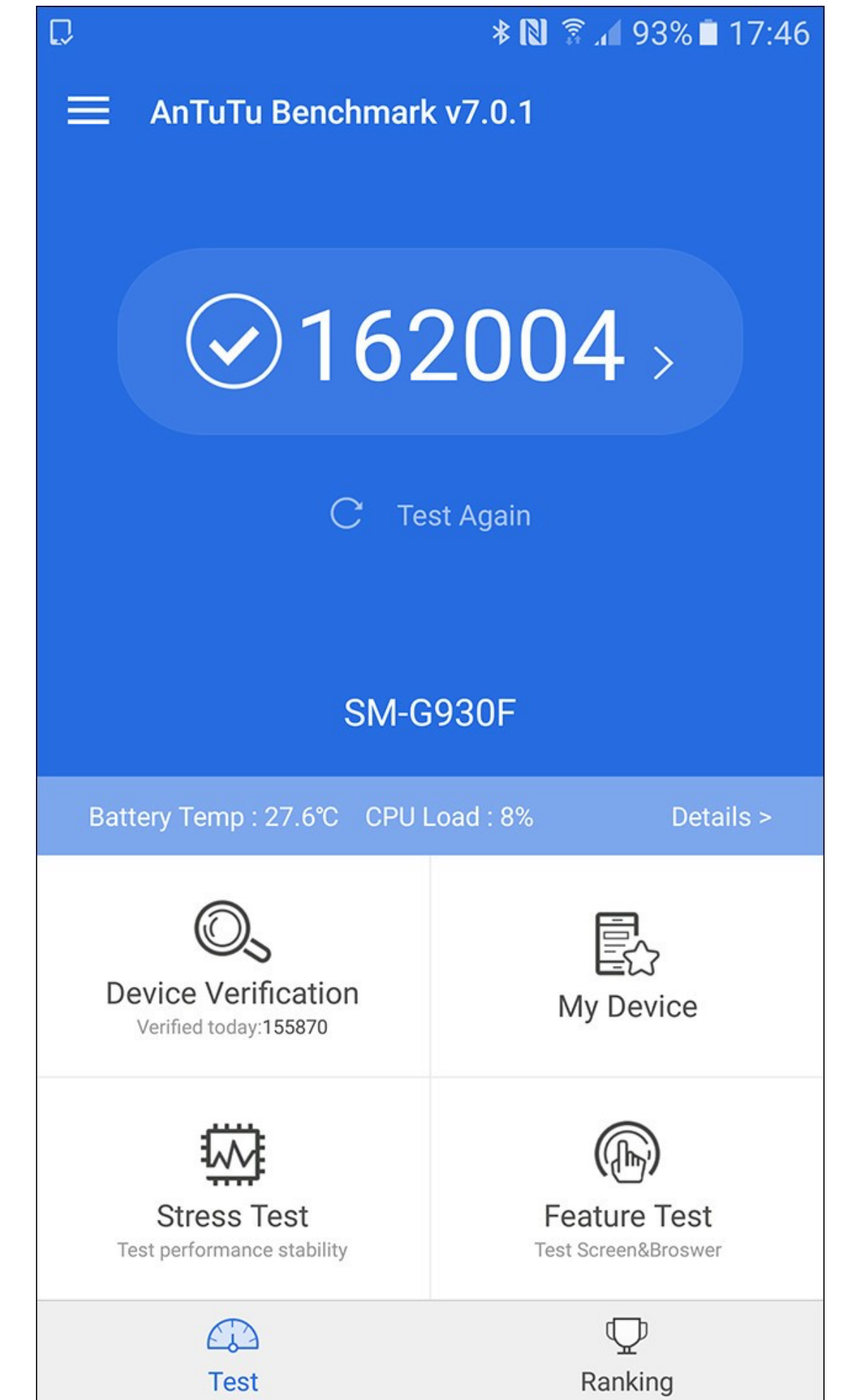
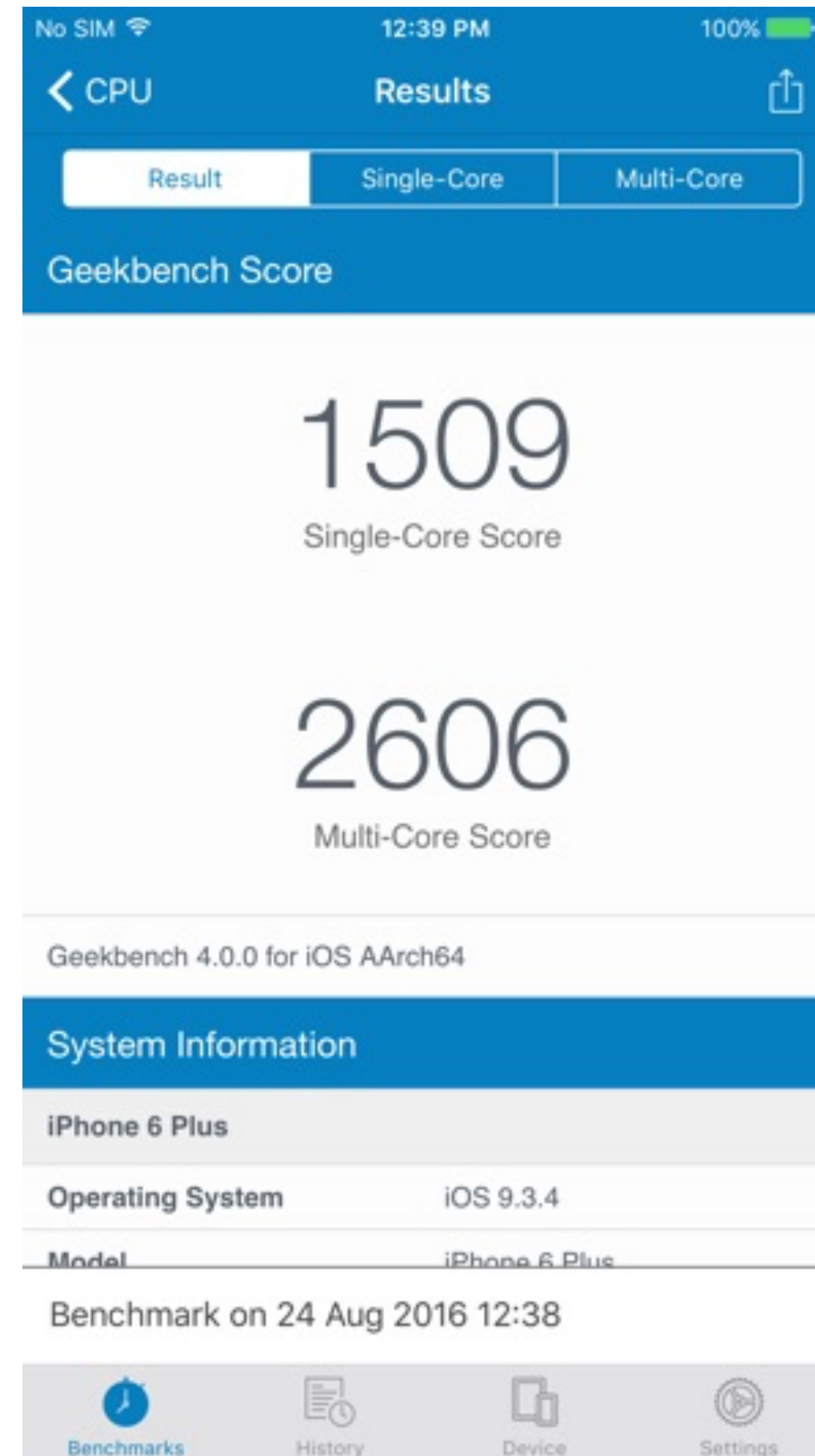


1. Thou shalt not rely solely on benchmarks.

Benchmarks like Geekbench and AnTuTu focus on core computational kernels (such as Dijkstra, JPEG compression, GEMM, and FFT) that exercise the CPU and GPU.

These benchmarks continuously stress the component, rather than mimic the intermitted nature of mobile usage.

Over 60,000 new applications are released in the Google Play Store each month.



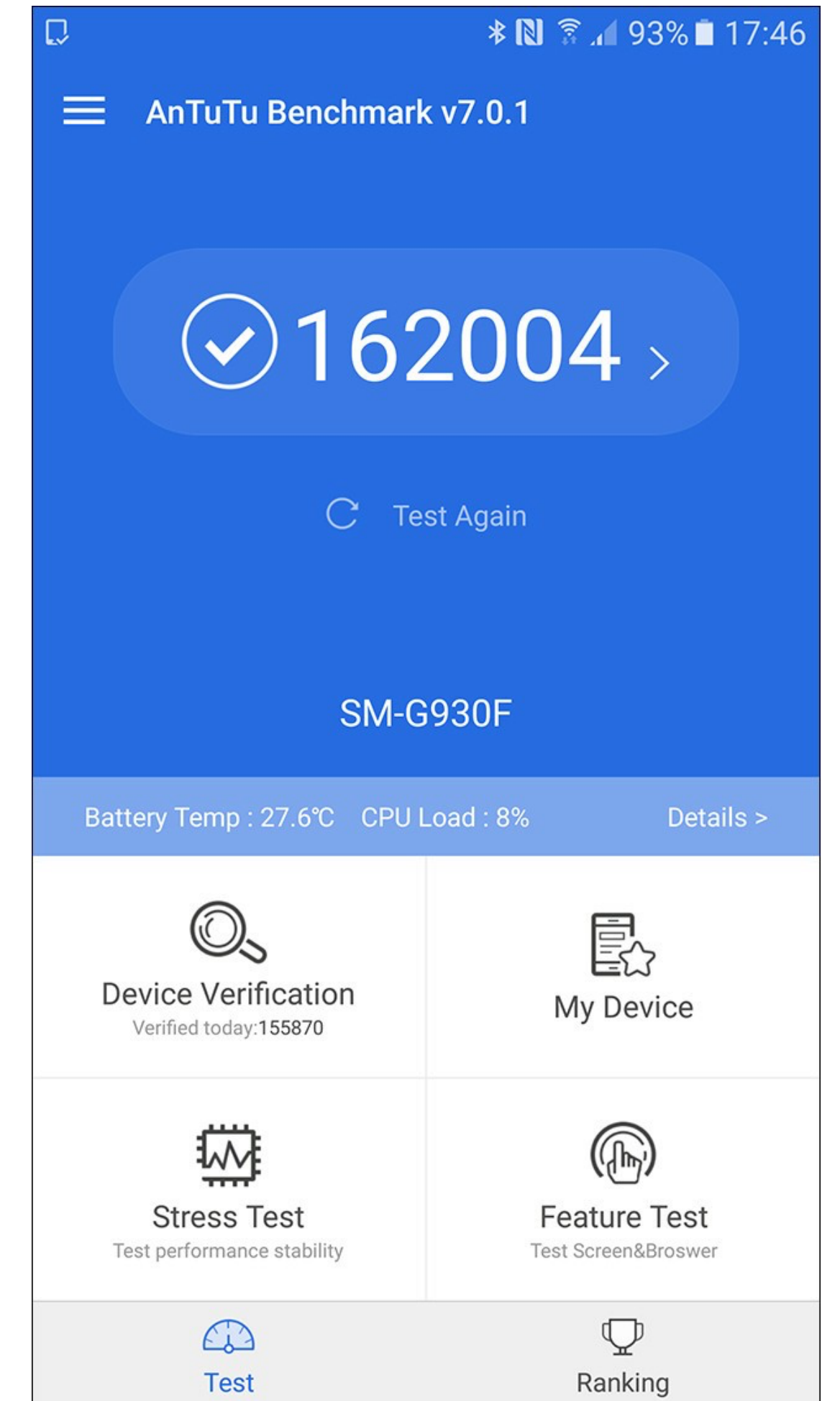
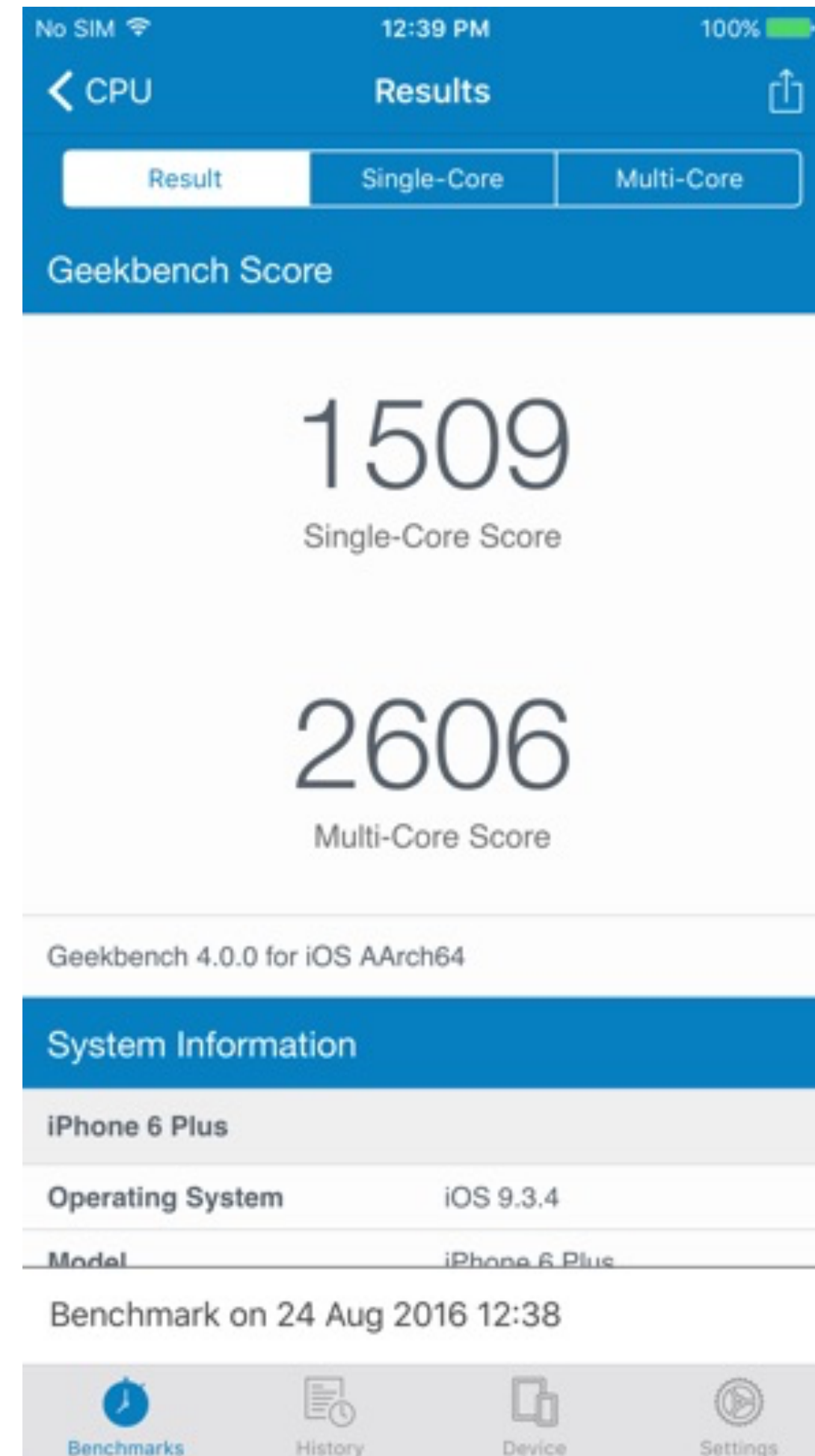
1. Thou shalt not rely solely on benchmarks.

Benchmarks like Geekbench and AnTuTu focus on core computational kernels (such as Dijkstra, JPEG compression, GEMM, and FFT) that exercise the CPU and GPU.

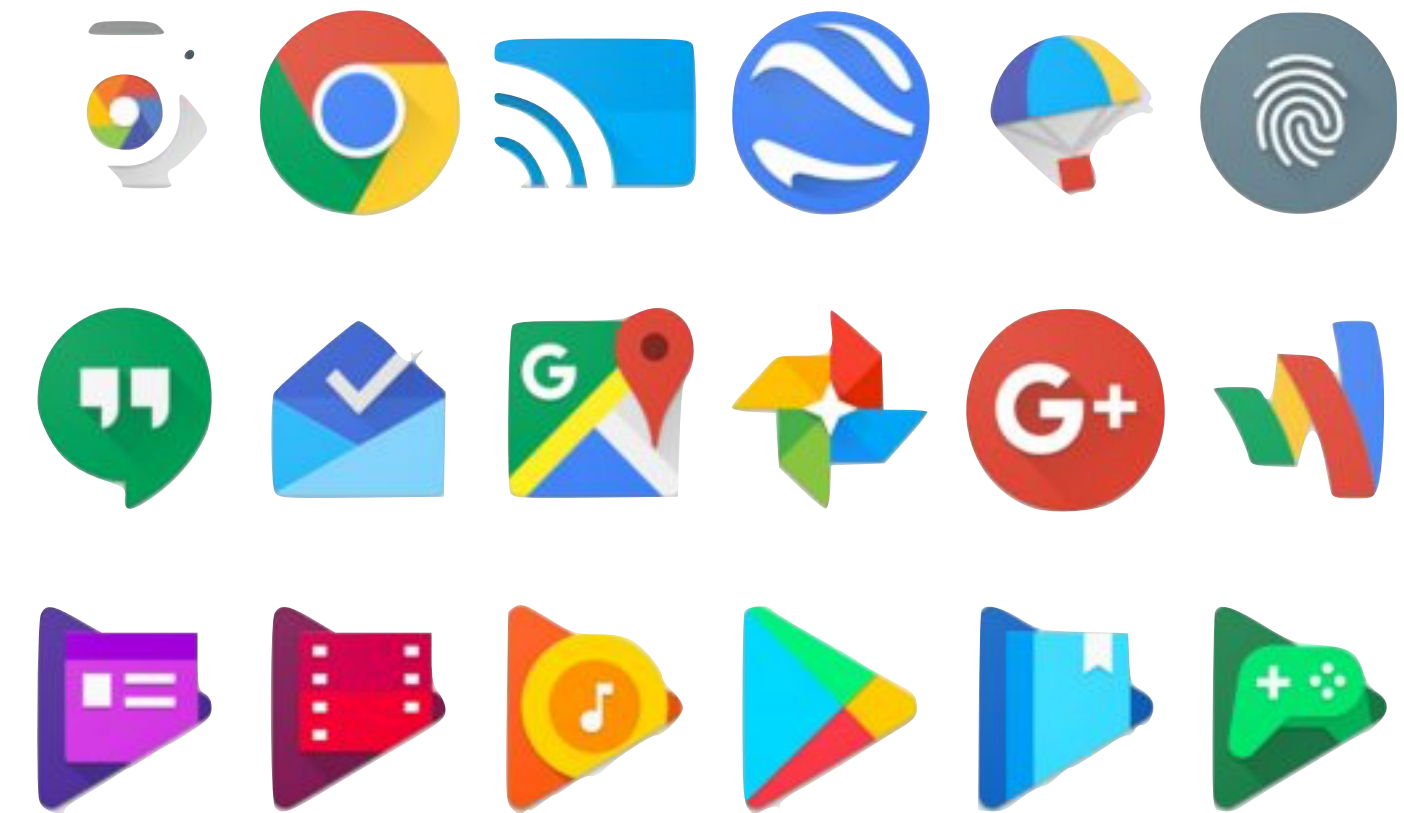
These benchmarks continuously stress the component, rather than mimic the intermitted nature of mobile usage.

Over 60,000 new applications are released in the Google Play Store each month.

➡ Need a more representative solution to benchmarking mobile systems.

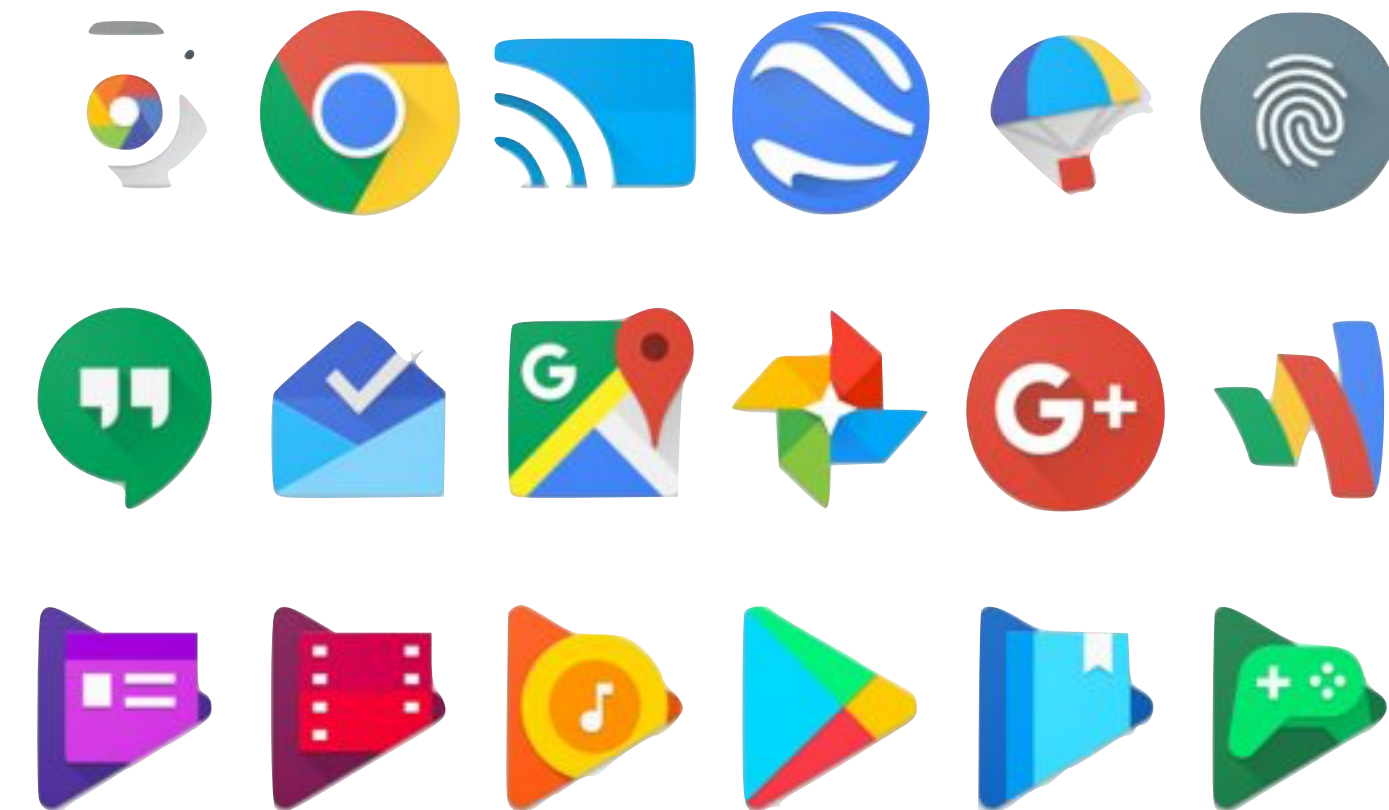


2. Thou shalt not cherry-pick applications.



2. Thou shalt not cherry-pick applications.

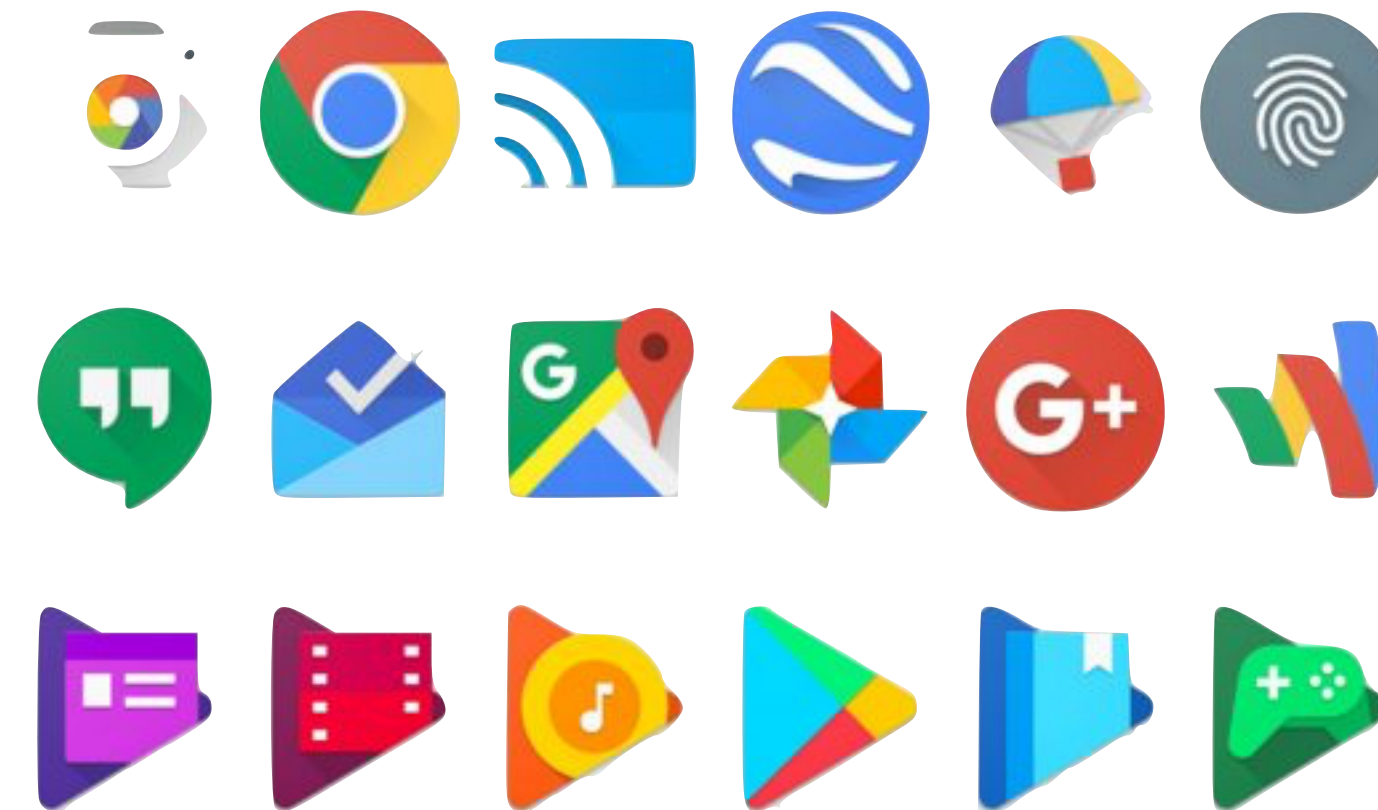
Ensure that we pick the right applications to investigate deeply for architectural design decisions.



2. Thou shalt not cherry-pick applications.

Ensure that we pick the right applications to investigate deeply for architectural design decisions.

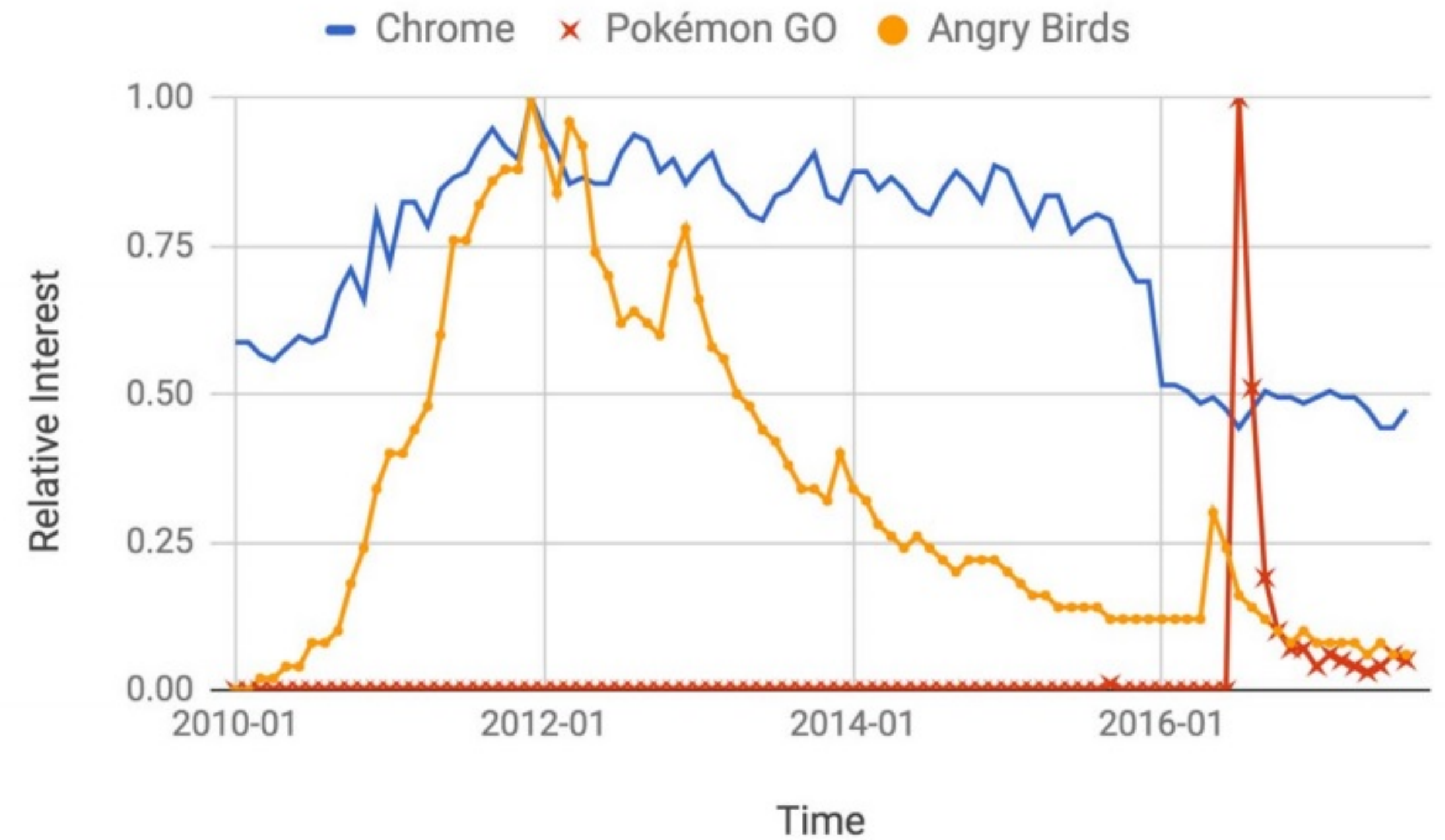
Figure shows the popularity for three well-known applications: Google Chrome, Angry Birds, and Pokémon GO. See rapid change.



2. Thou shalt not cherry-pick applications.

Ensure that we pick the right applications to investigate deeply for architectural design decisions.

Figure shows the popularity for three well-known applications: Google Chrome, Angry Birds, and Pokémon GO. See rapid change.

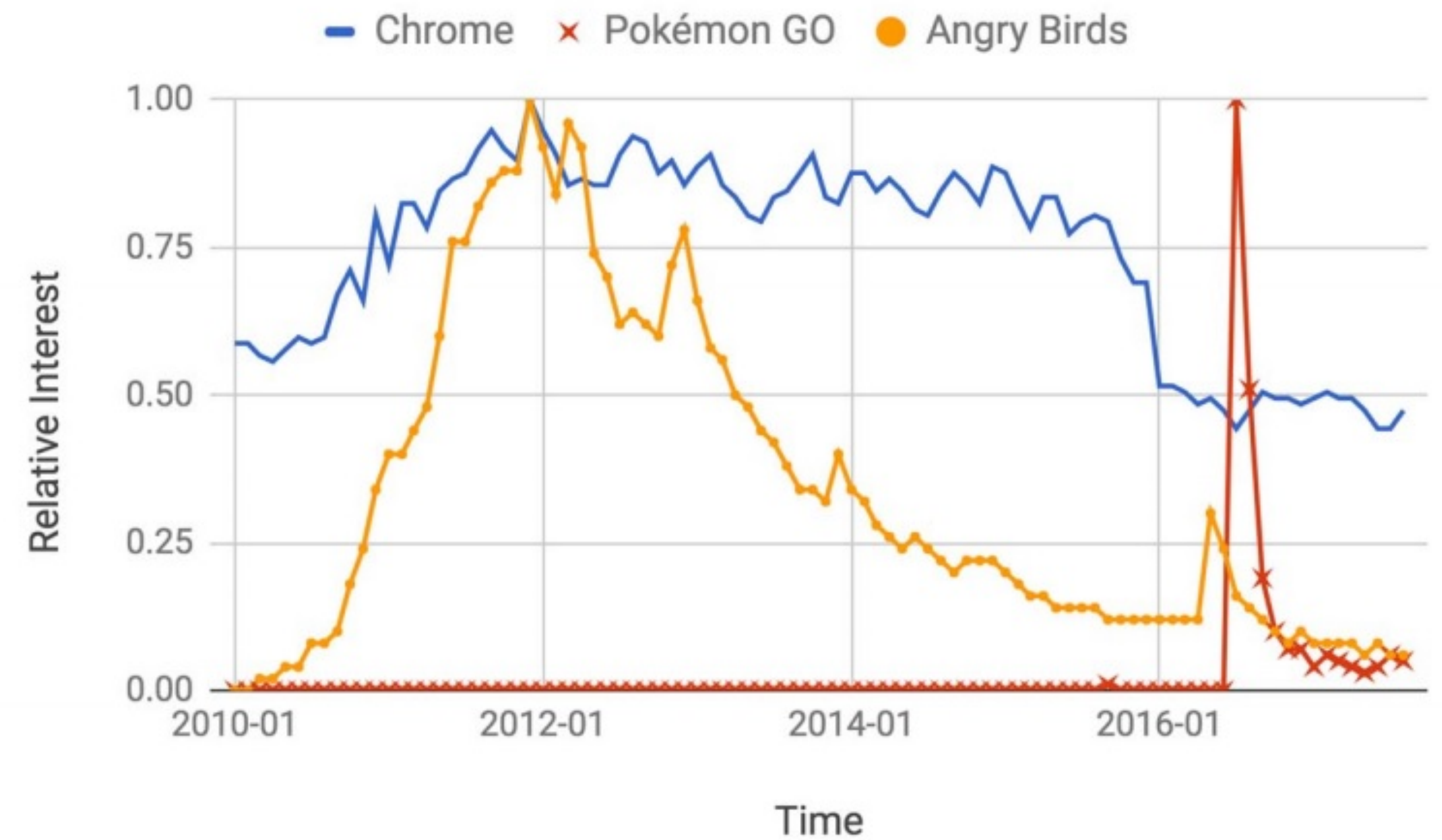


2. Thou shalt not cherry-pick applications.

Ensure that we pick the right applications to investigate deeply for architectural design decisions.

Figure shows the popularity for three well-known applications: Google Chrome, Angry Birds, and Pokémon GO. See rapid change.

Angry Birds is still a popular application for benchmarking mobile systems in the literature for academic research.



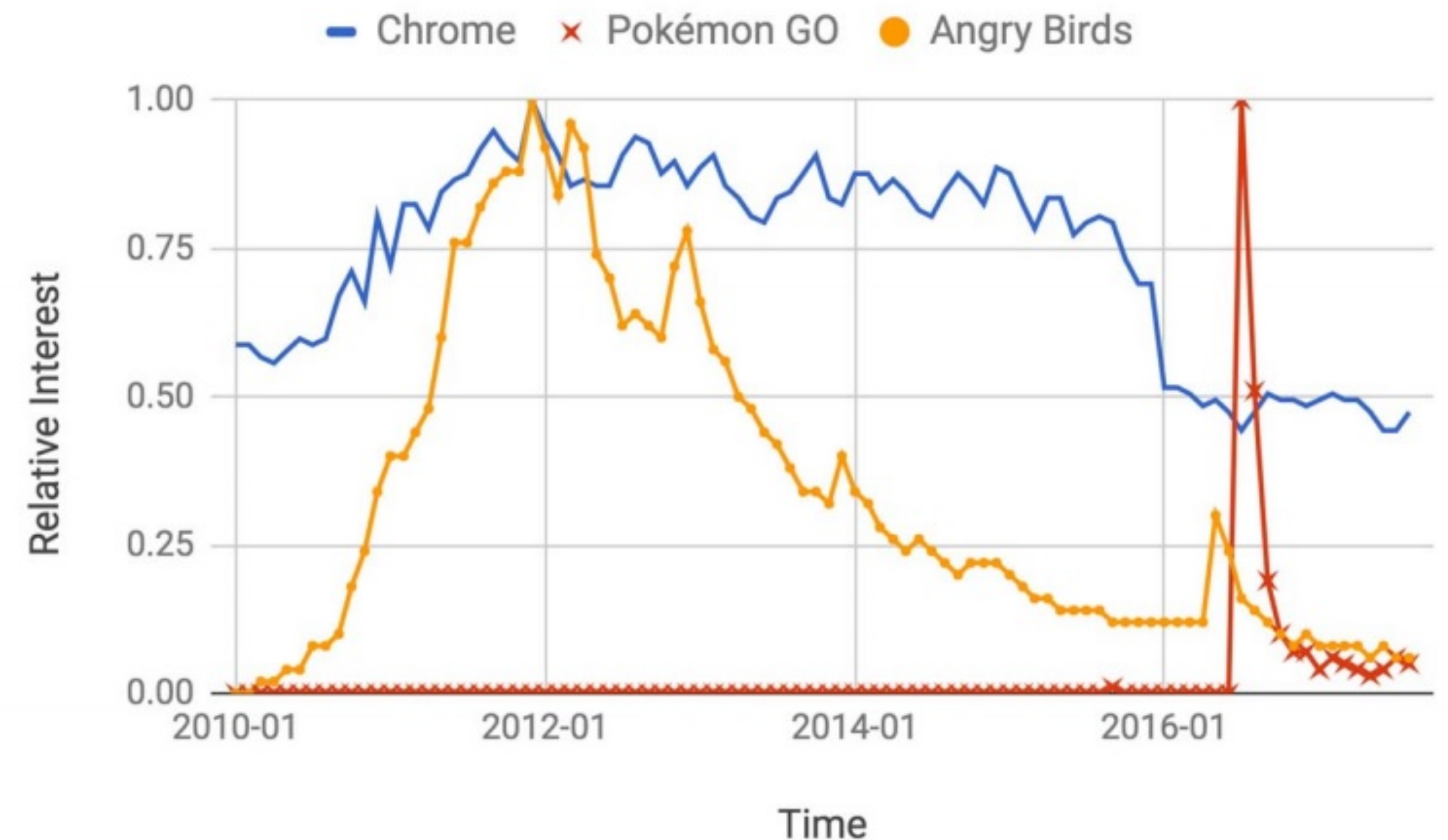
2. Thou shalt not cherry-pick applications.

Ensure that we pick the right applications to investigate deeply for architectural design decisions.

Figure shows the popularity for three well-known applications: Google Chrome, Angry Birds, and Pokémon GO. See rapid change.

Angry Birds is still a popular application for benchmarking mobile systems in the literature for academic research.

➡ It is important for hardware vendors and architects to keep pace with which applications are “hot” and keep the mobile benchmarking application suite updated.



3. Thou shalt not ignore the web browser.



3. Thou shalt not ignore the web browser.

In developing countries, users tend to prefer the browser over installing single-purpose or dedicated native applications.



3. Thou shalt not ignore the web browser.

In developing countries, users tend to prefer the browser over installing single-purpose or dedicated native applications.

Many mobile applications, such as social and messenger applications, allow users to load webpages inside using “WebView.”



3. Thou shalt not ignore the web browser.

In developing countries, users tend to prefer the browser over installing single-purpose or dedicated native applications.

Many mobile applications, such as social and messenger applications, allow users to load webpages inside using “WebView.”

The browser’s broad capabilities and flexibility are causing it to outpace many mobile applications in density and complexity.

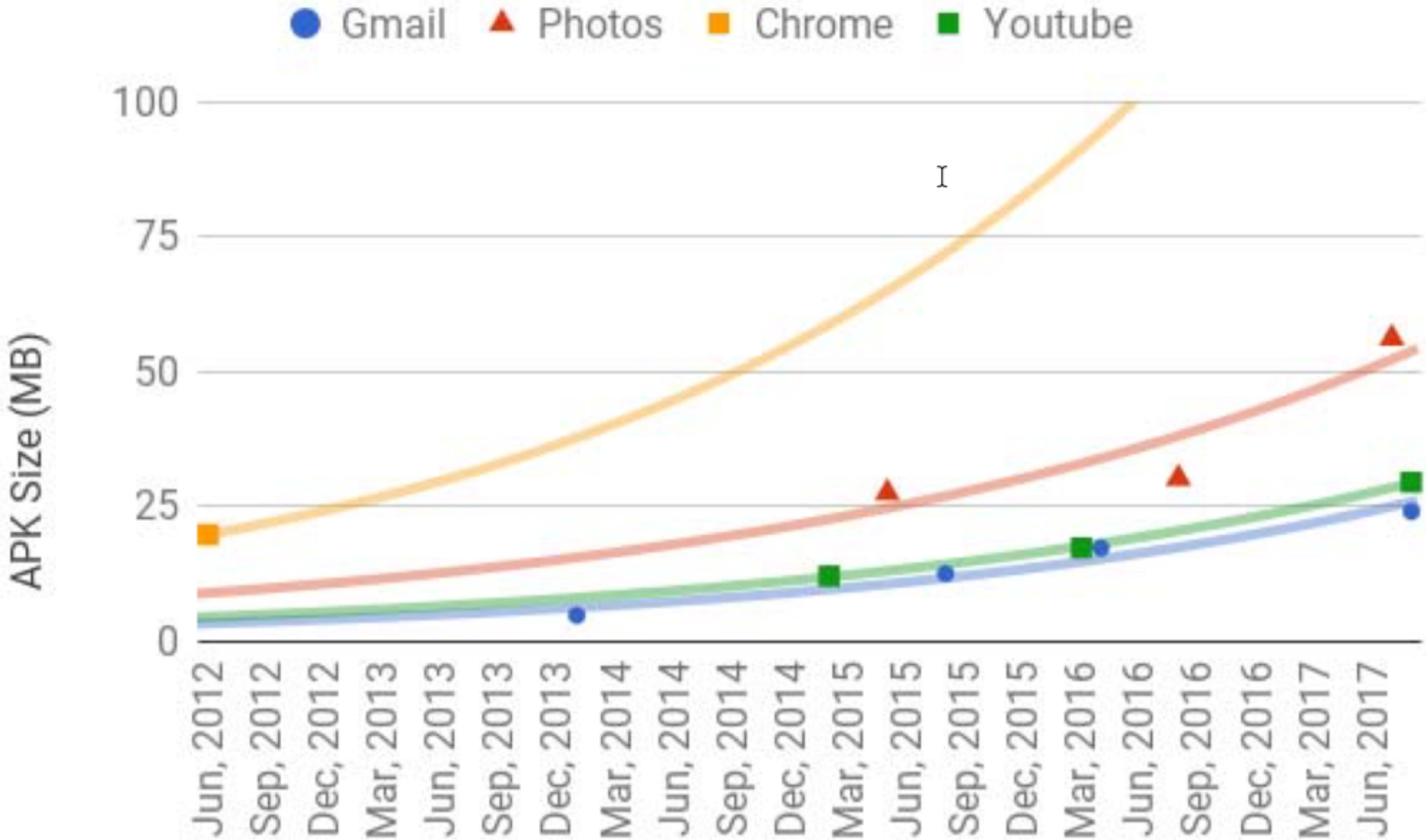


3. Thou shalt not ignore the web browser.

In developing countries, users tend to prefer the browser over installing single-purpose or dedicated native applications.

Many mobile applications, such as social and messenger applications, allow users to load webpages inside using “WebView.”

The browser’s broad capabilities and flexibility are causing it to outpace many mobile applications in density and complexity.



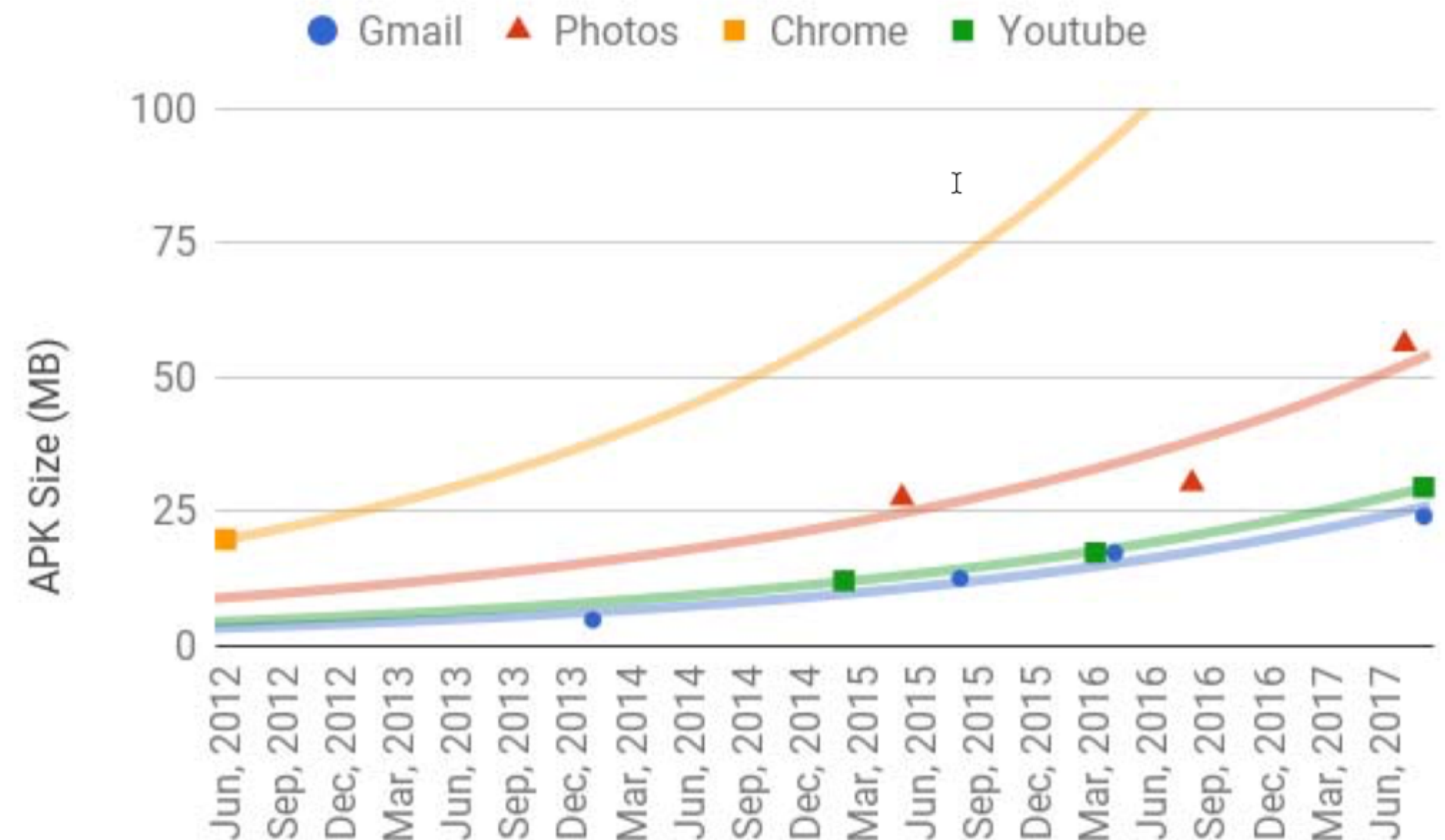
3. Thou shalt not ignore the web browser.

In developing countries, users tend to prefer the browser over installing single-purpose or dedicated native applications.

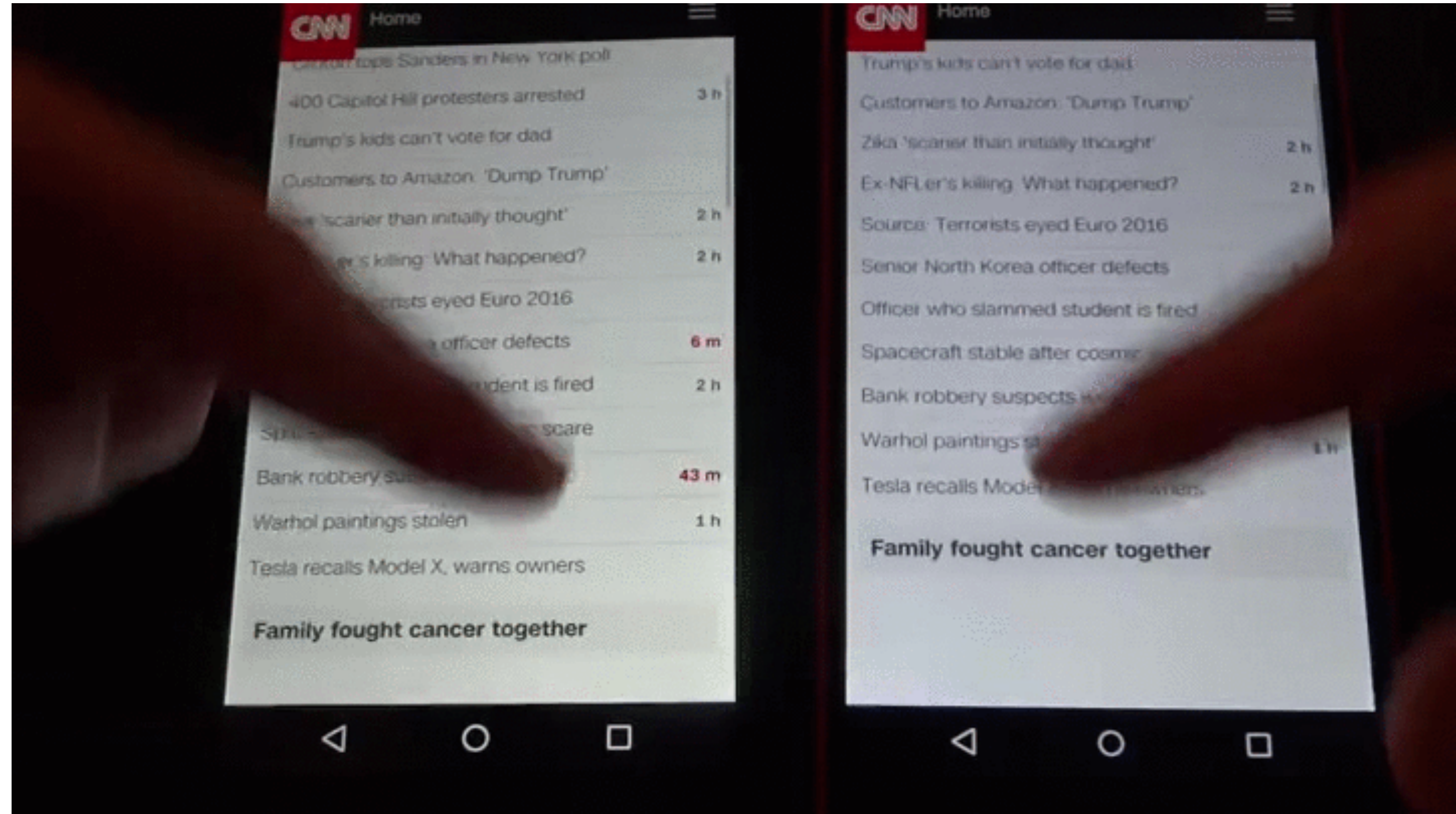
Many mobile applications, such as social and messenger applications, allow users to load webpages inside using “WebView.”

The browser’s broad capabilities and flexibility are causing it to outpace many mobile applications in density and complexity.

➡ Mobile processor architects must include the browser in the creation of a benchmark suite and any sort of hardware or software optimization analysis. It is akin to GNU gcc.

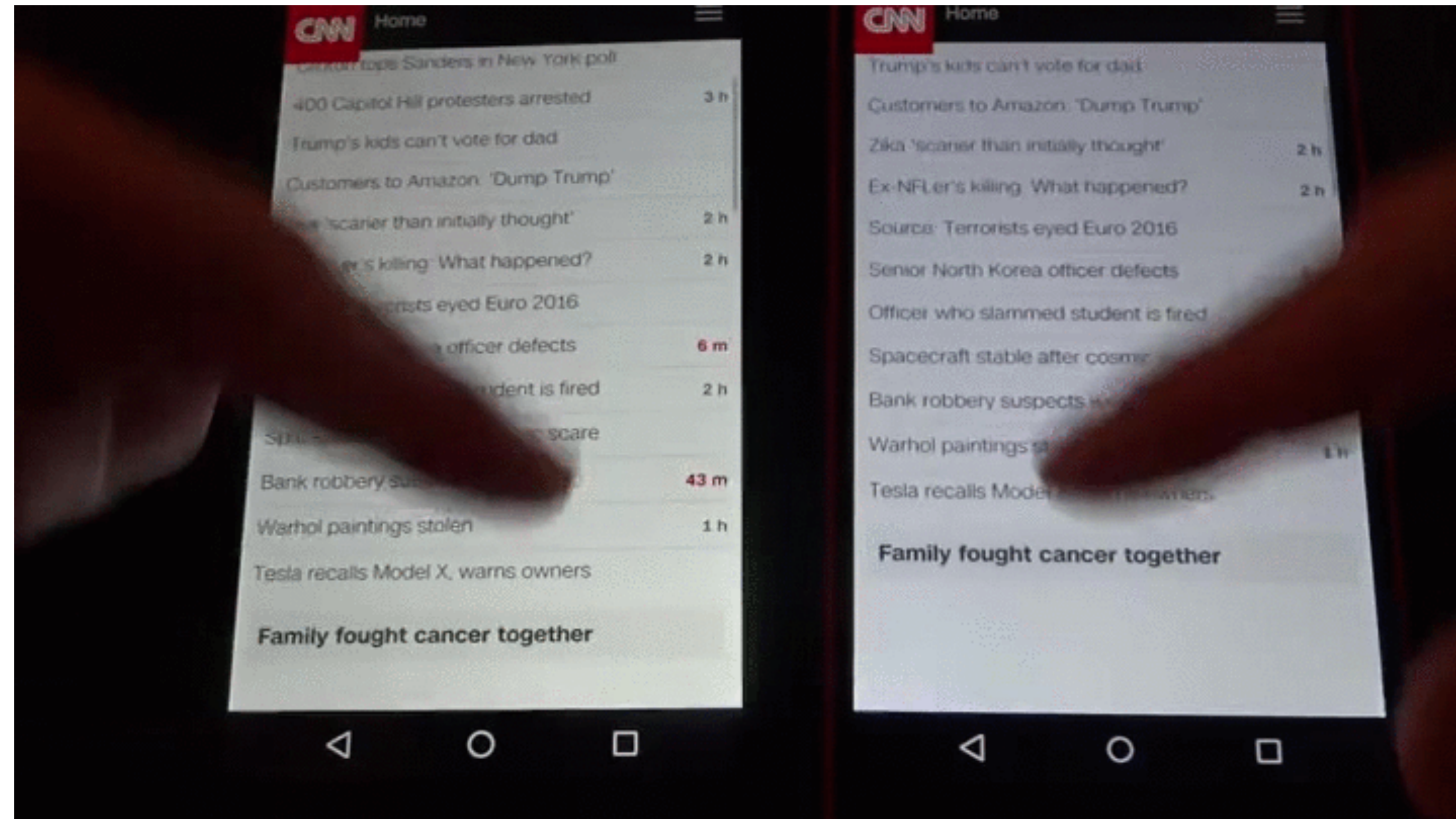


4. Thou shalt not drop a frame.



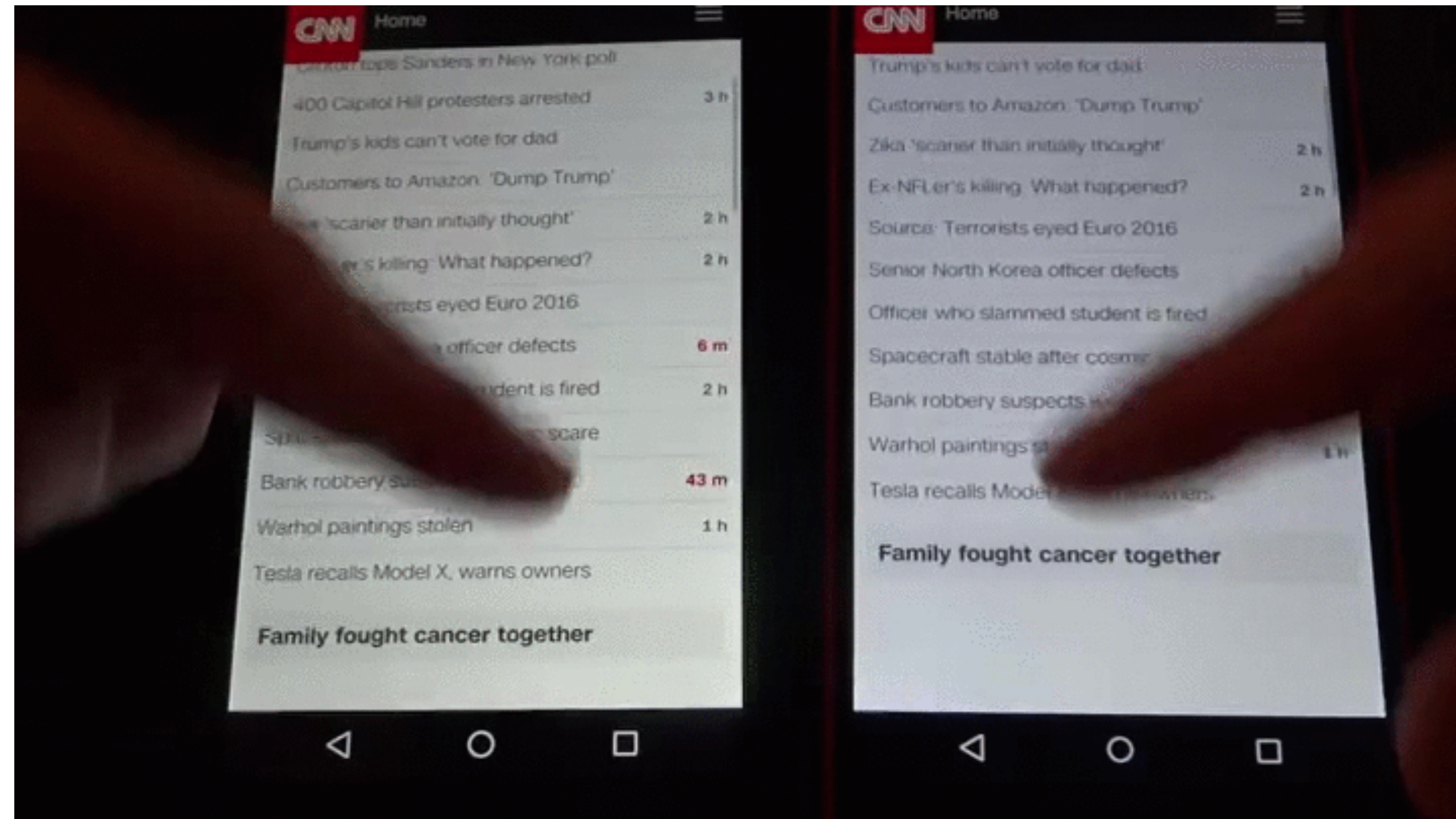
4. Thou shalt not drop a frame.

An average user taps, types, swipes, or clicks the device 2K times a day, and about 10% of us perform those actions 5,427 times a day.



4. Thou shalt not drop a frame.

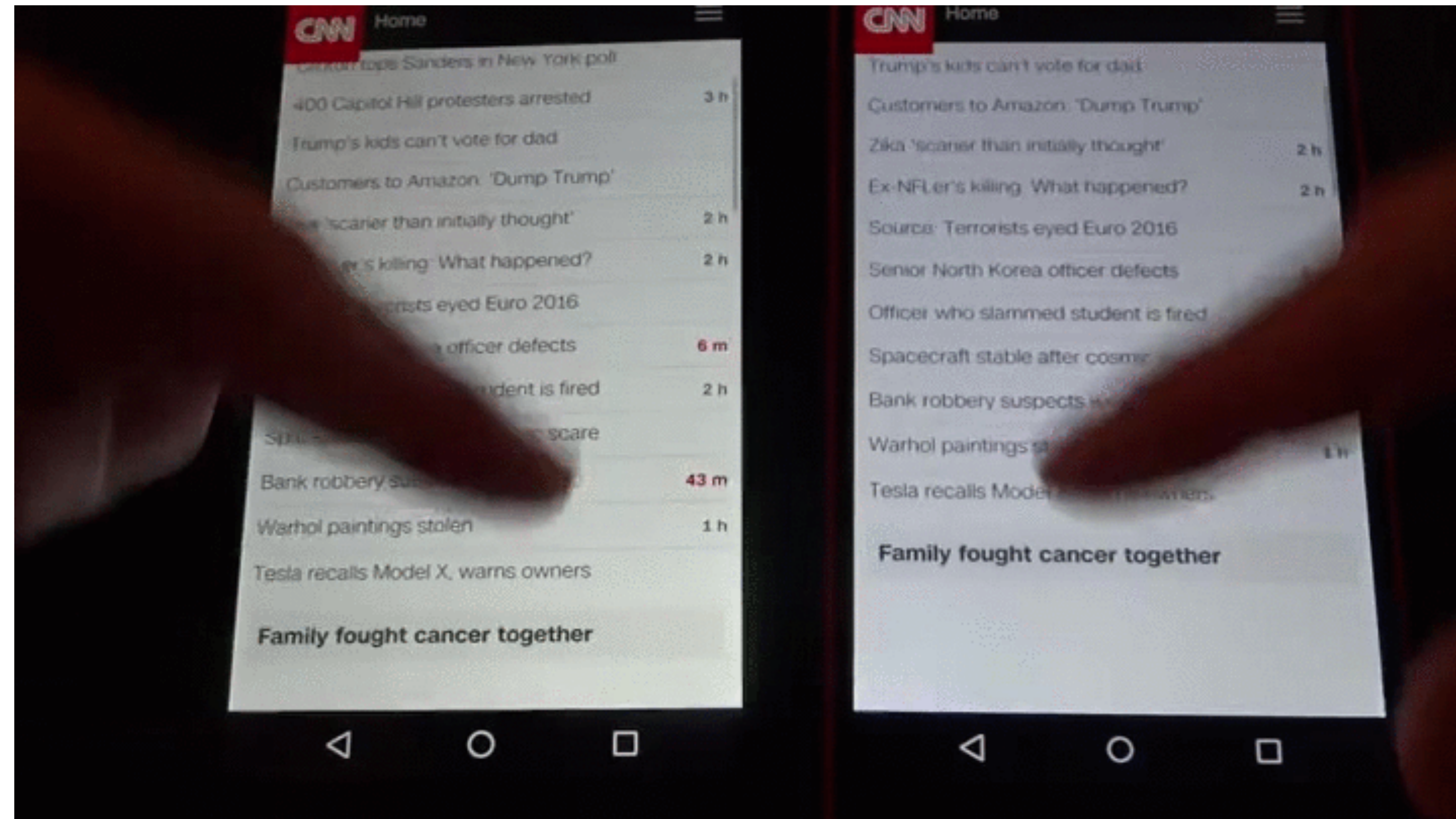
An average user taps, types, swipes, or clicks the device 2K times a day, and about 10% of us perform those actions 5,427 times a day.



4. Thou shalt not drop a frame.

An average user taps, types, swipes, or clicks the device 2K times a day, and about 10% of us perform those actions 5,427 times a day.

To ensure responsiveness, the system must maintain 60 FPS consistently without any dropped (or delayed) frames, more commonly or typically referred to as “jank”.

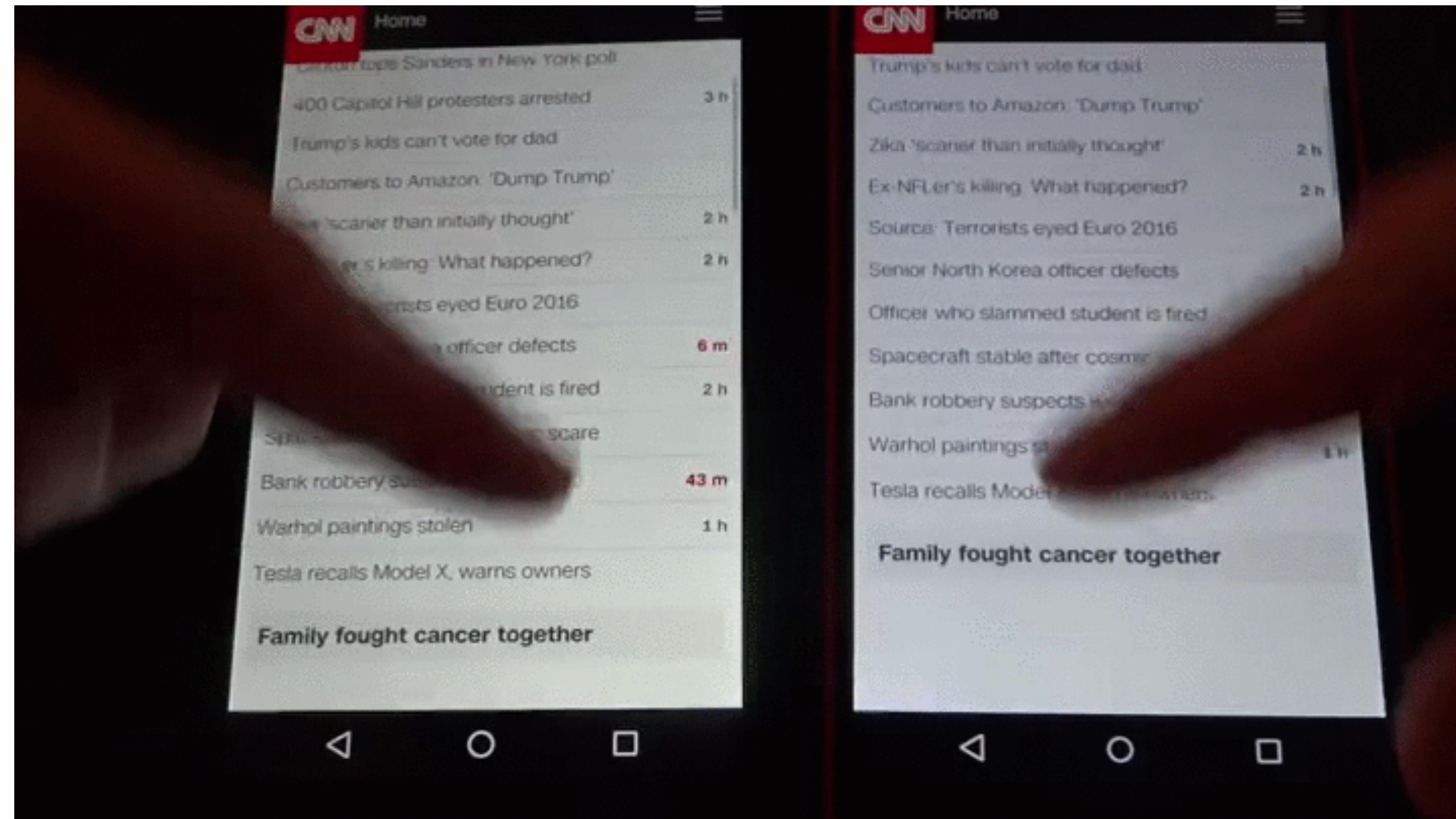


4. Thou shalt not drop a frame.

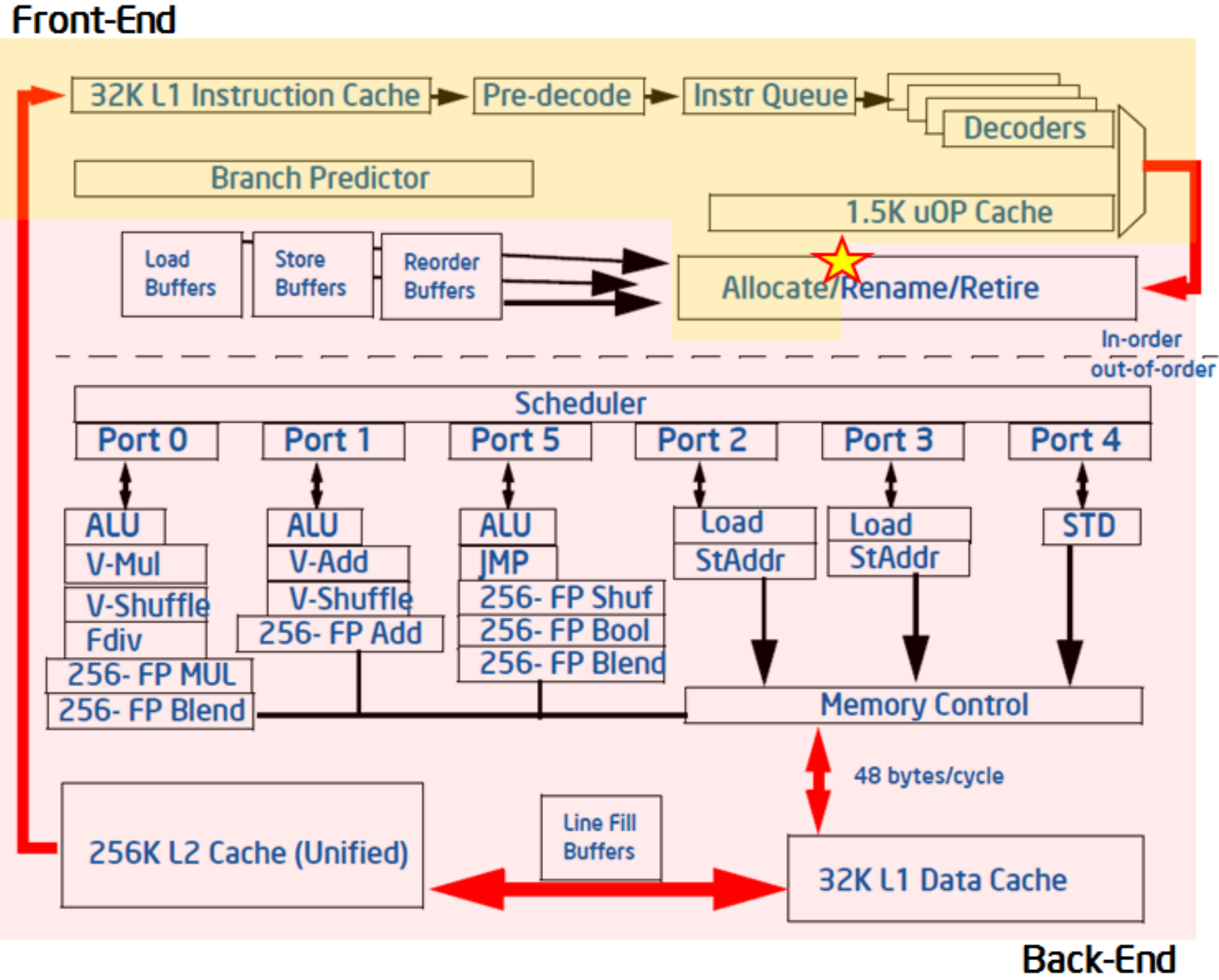
An average user taps, types, swipes, or clicks the device 2K times a day, and about 10% of us perform those actions 5,427 times a day.

To ensure responsiveness, the system must maintain 60 FPS consistently without any dropped (or delayed) frames, more commonly or typically referred to as “jank”.

👉 We need to design a system that can provide a sustainable throughput with no dropped frames, rather than build a system that has a high frame rate (such as 120 FPS) but drops frames even occasionally.

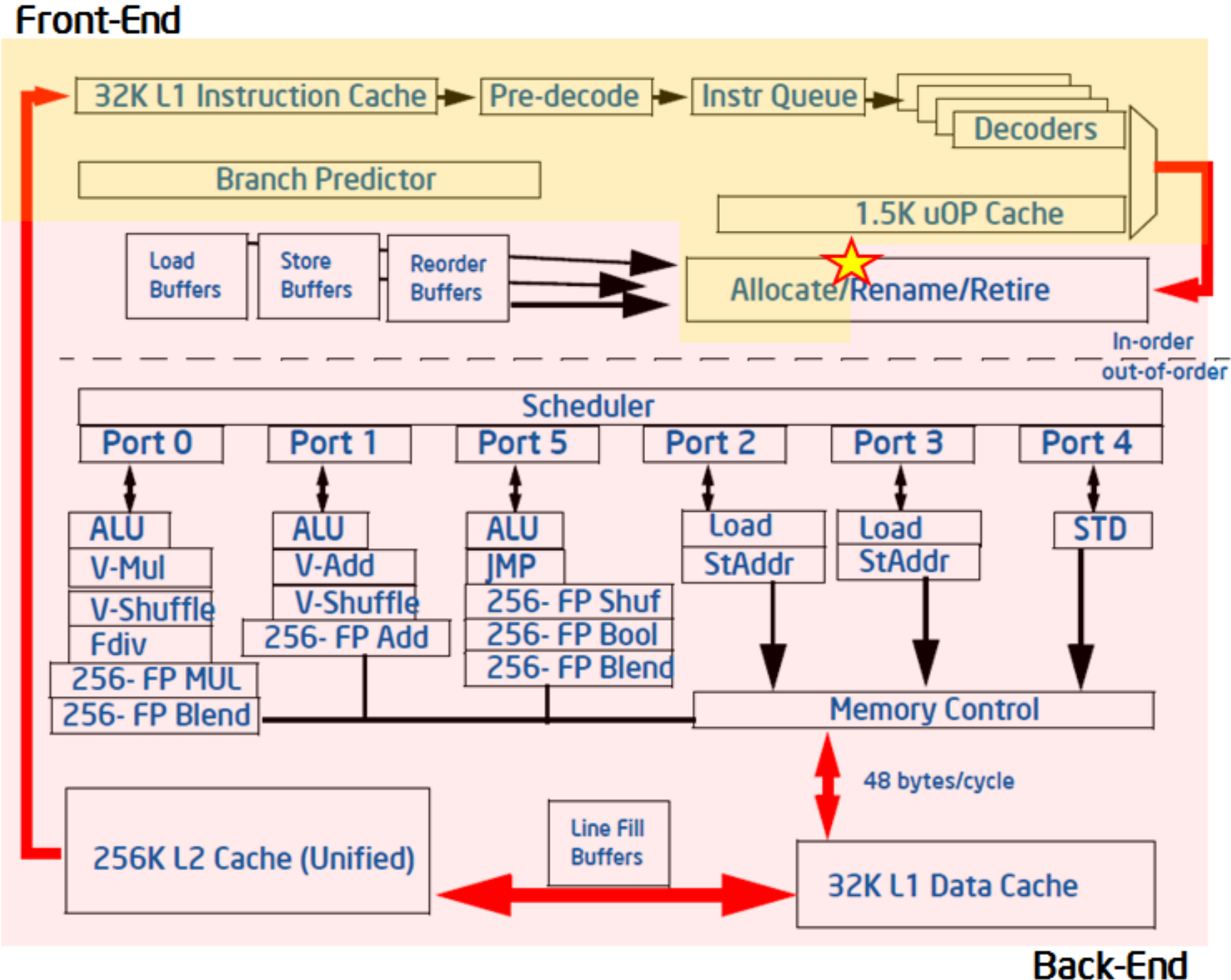


5. Thou shalt not idolize microarchitectural efficiency.



5. Thou shalt not idolize microarchitectural efficiency.

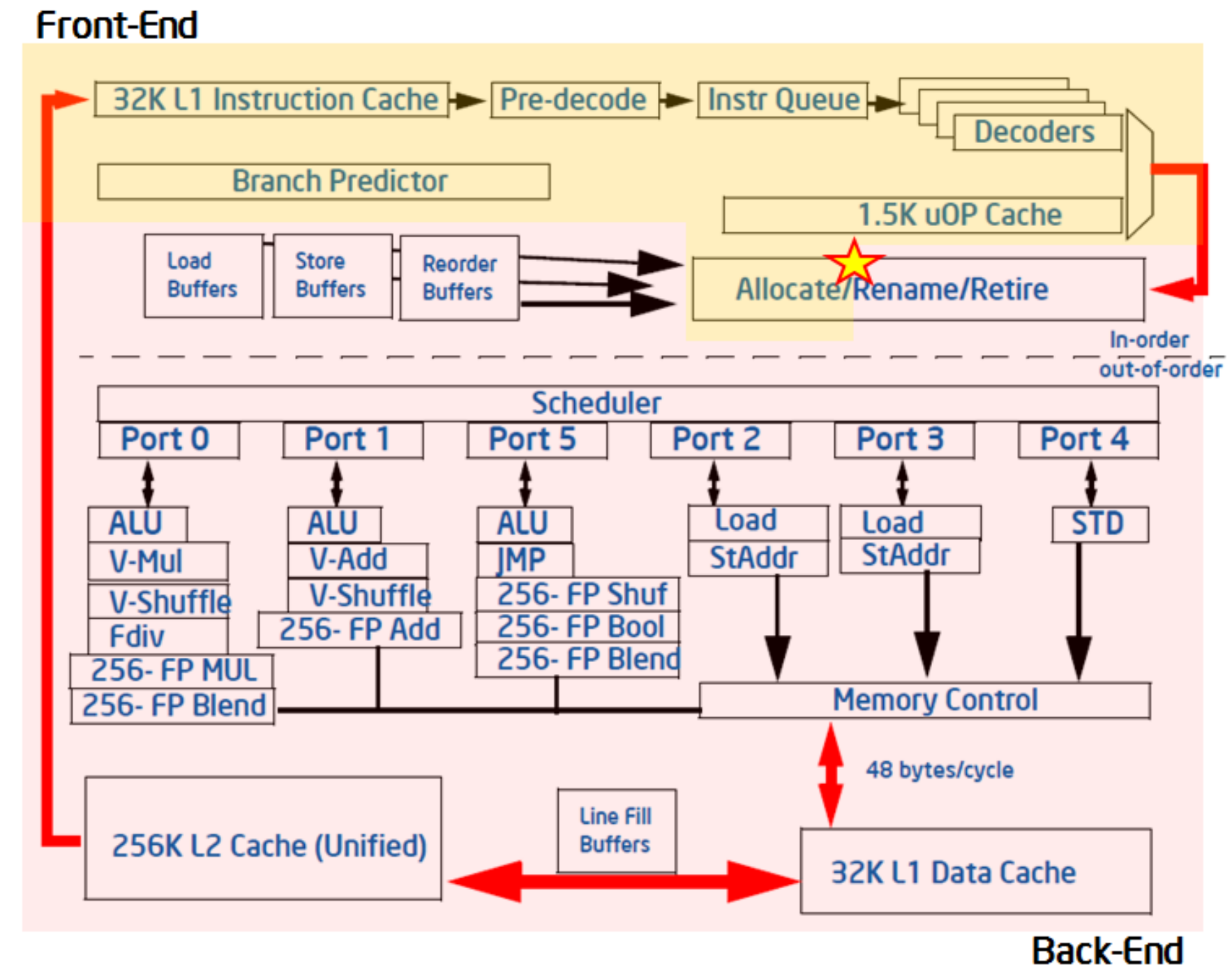
The textbook “Computer Architecture: A Quantitative Approach” by David A. Patterson and John L. Hennessy has taught architects to have a strong and quantitatively rigorous approach to measuring simulated or real hardware performance.



5. Thou shalt not idolize microarchitectural efficiency.

The textbook “Computer Architecture: A Quantitative Approach” by David A. Patterson and John L. Hennessy has taught architects to have a strong and quantitatively rigorous approach to measuring simulated or real hardware performance.

The measure of performance in a mobile device is not how fast a processor can compute; rather, its true capability lies in its ability to deliver user-perceivable satisfaction improvements.

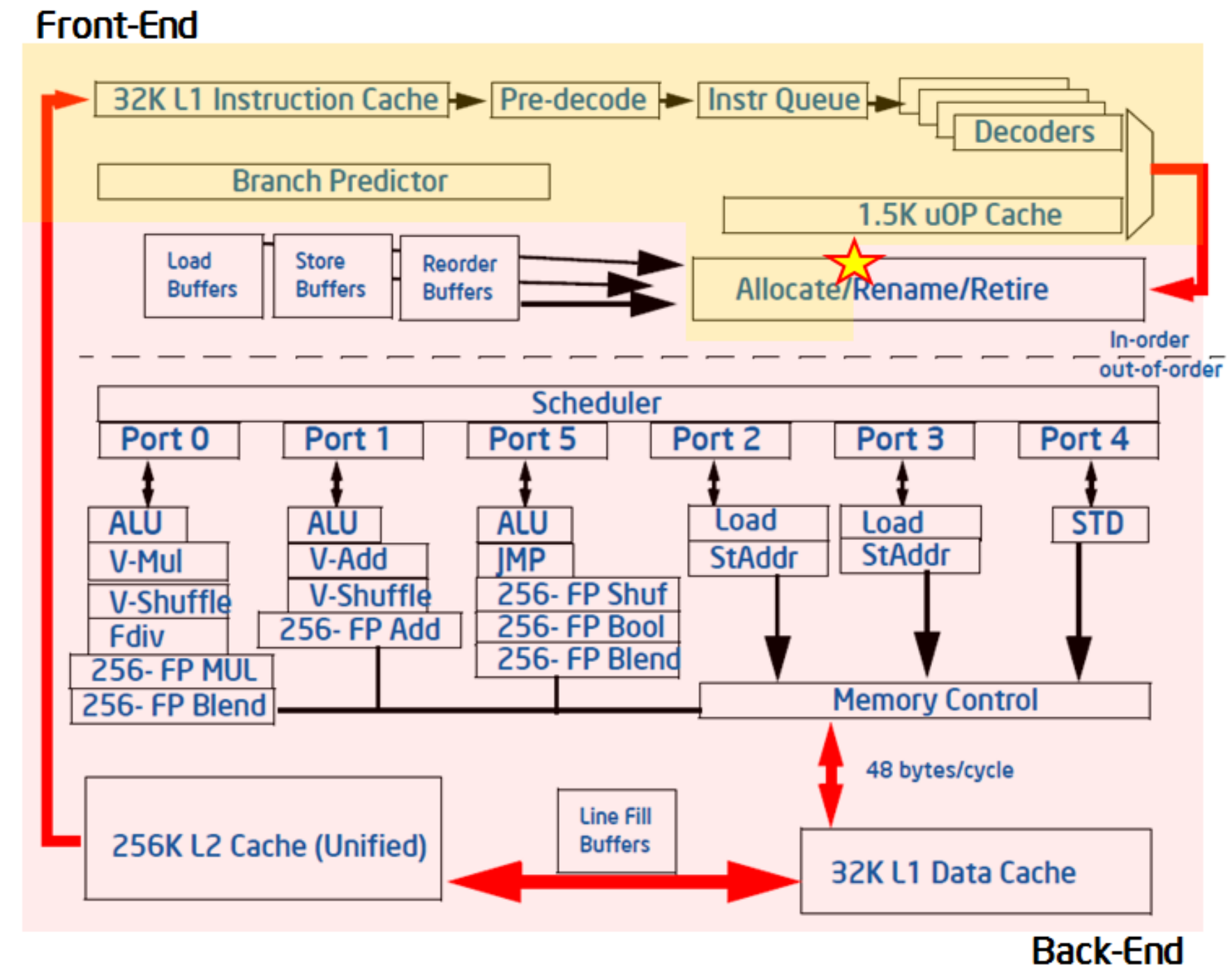


5. Thou shalt not idolize microarchitectural efficiency.

The textbook “Computer Architecture: A Quantitative Approach” by David A. Patterson and John L. Hennessy has taught architects to have a strong and quantitatively rigorous approach to measuring simulated or real hardware performance.

The measure of performance in a mobile device is not how fast a processor can compute; rather, its true capability lies in its ability to deliver user-perceivable satisfaction improvements.

If the microarchitectural enhancements cannot translate to measurable user experience, doing so won't be useful.

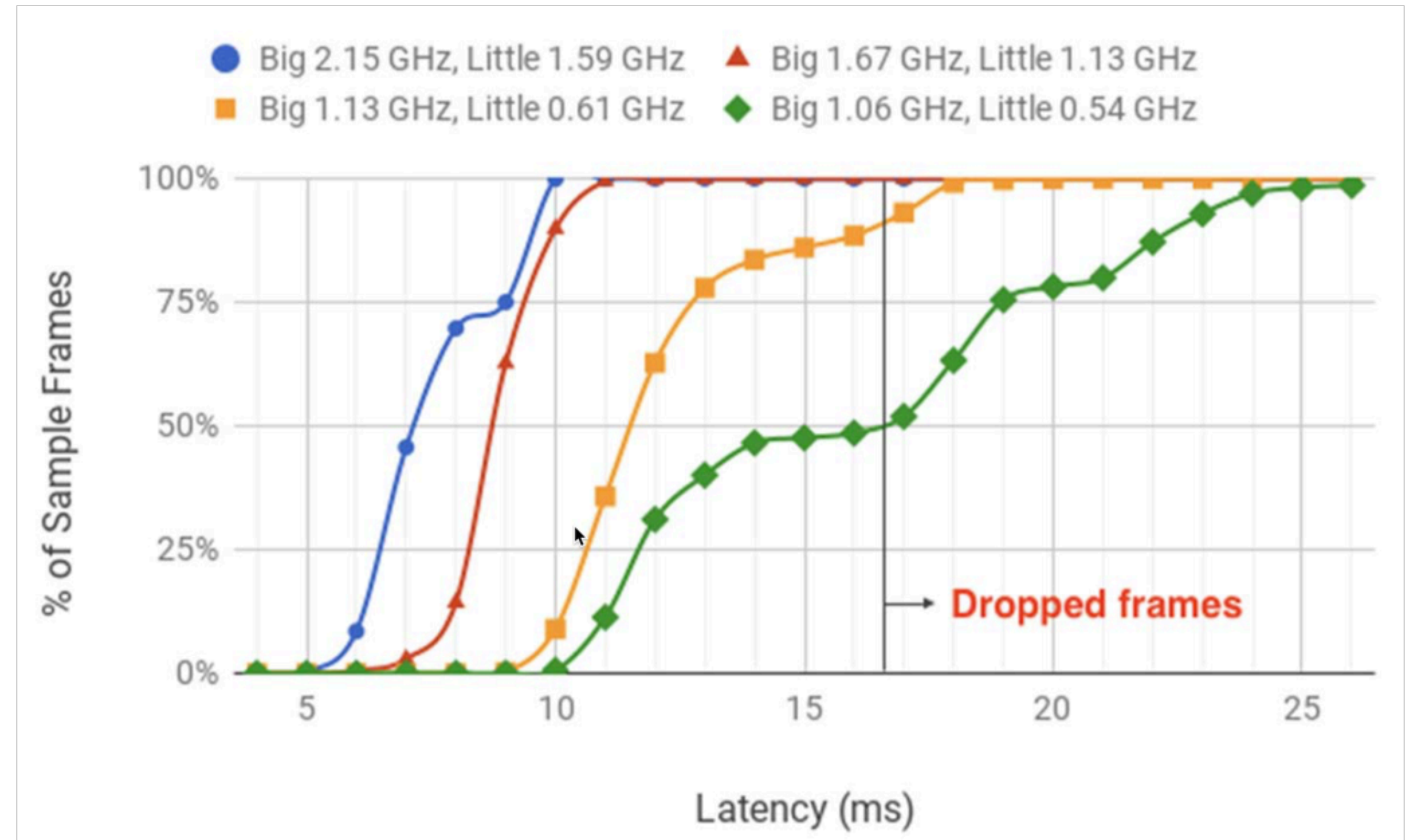


5. Thou shalt not idolize microarchitectural efficiency.

The textbook “Computer Architecture: A Quantitative Approach” by David A. Patterson and John L. Hennessy has taught architects to have a strong and quantitatively rigorous approach to measuring simulated or real hardware performance.

The measure of performance in a mobile device is not how fast a processor can compute; rather, its true capability lies in its ability to deliver user-perceivable satisfaction improvements.

If the microarchitectural enhancements cannot translate to measurable user experience, doing so won't be useful.



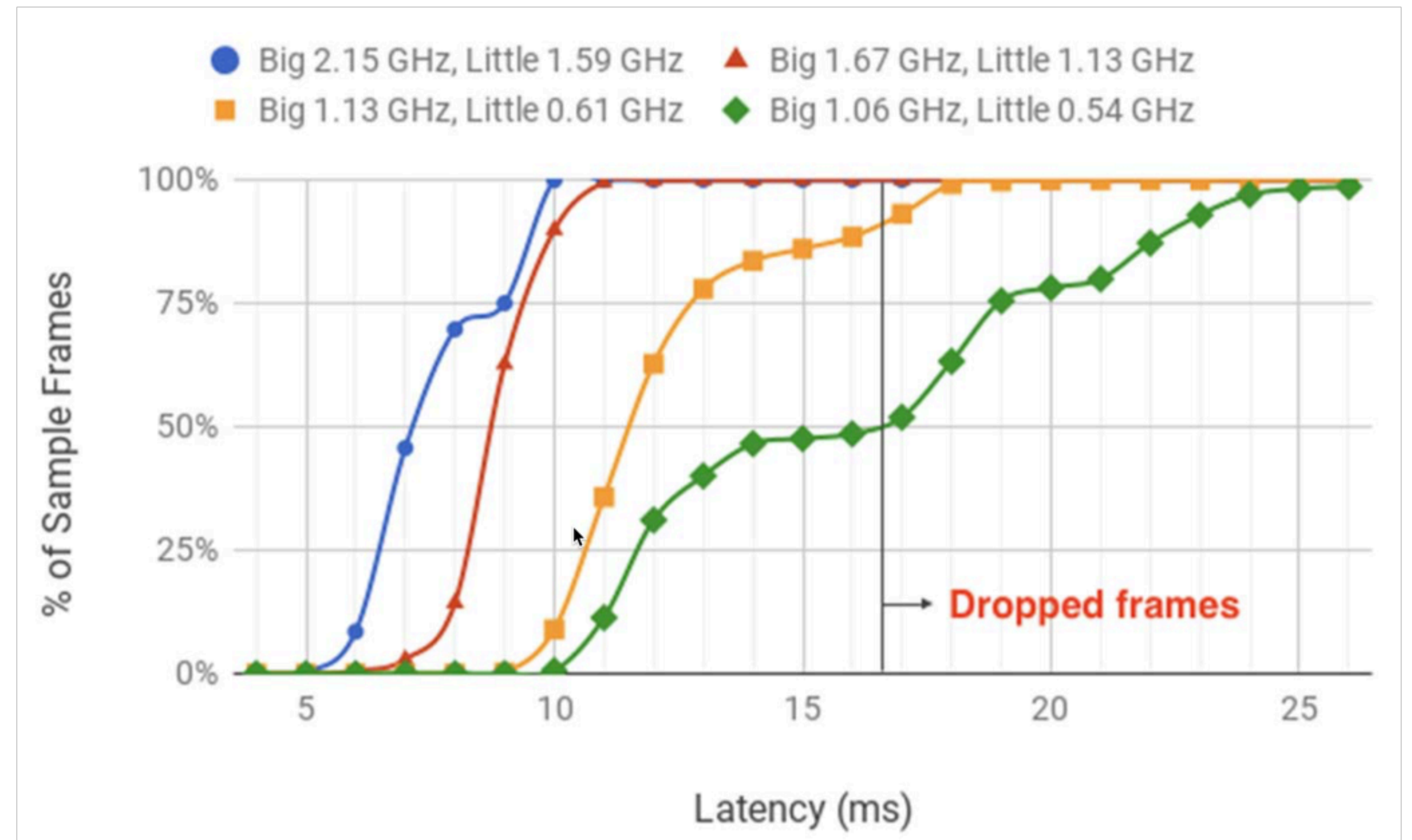
5. Thou shalt not idolize microarchitectural efficiency.

The textbook “Computer Architecture: A Quantitative Approach” by David A. Patterson and John L. Hennessy has taught architects to have a strong and quantitatively rigorous approach to measuring simulated or real hardware performance.

The measure of performance in a mobile device is not how fast a processor can compute; rather, its true capability lies in its ability to deliver user-perceivable satisfaction improvements.

If the microarchitectural enhancements cannot translate to measurable user experience, doing so won't be useful.

➡ It is important to understand the relationship between microarchitectural enhancements and the actual user impact.



6. Thou shalt not ignore tails in user experience.

| | 50%ile latency | 95%ile latency | 99%ile latency |
|----------------------------------|-----------------------|-----------------------|-----------------------|
| One random leaf finishes | 1ms | 5ms | 10ms |
| 95% of all leaf requests finish | 12ms | 32ms | 70ms |
| 100% of all leaf requests finish | 40ms | 87ms | 140ms |

Single Request Completion time (Source: The Tail at Scale)

6. Thou shalt not ignore tails in user experience.

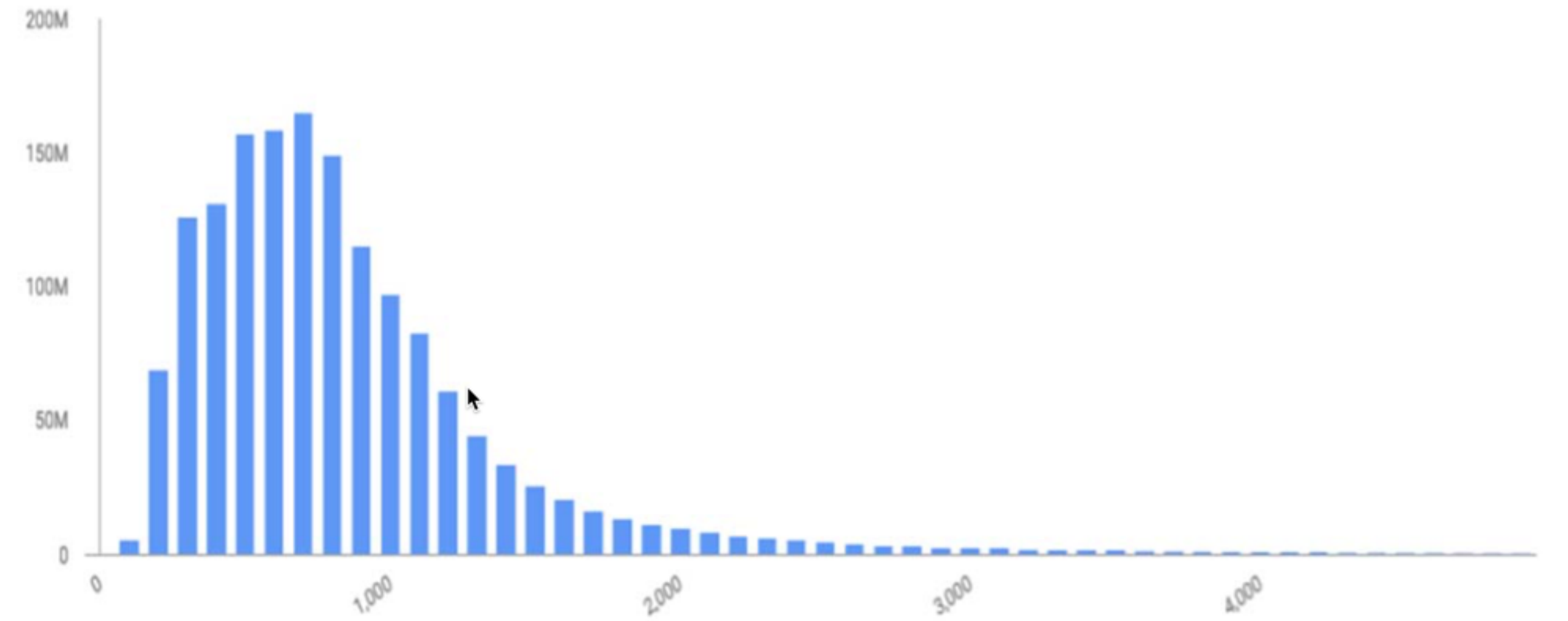
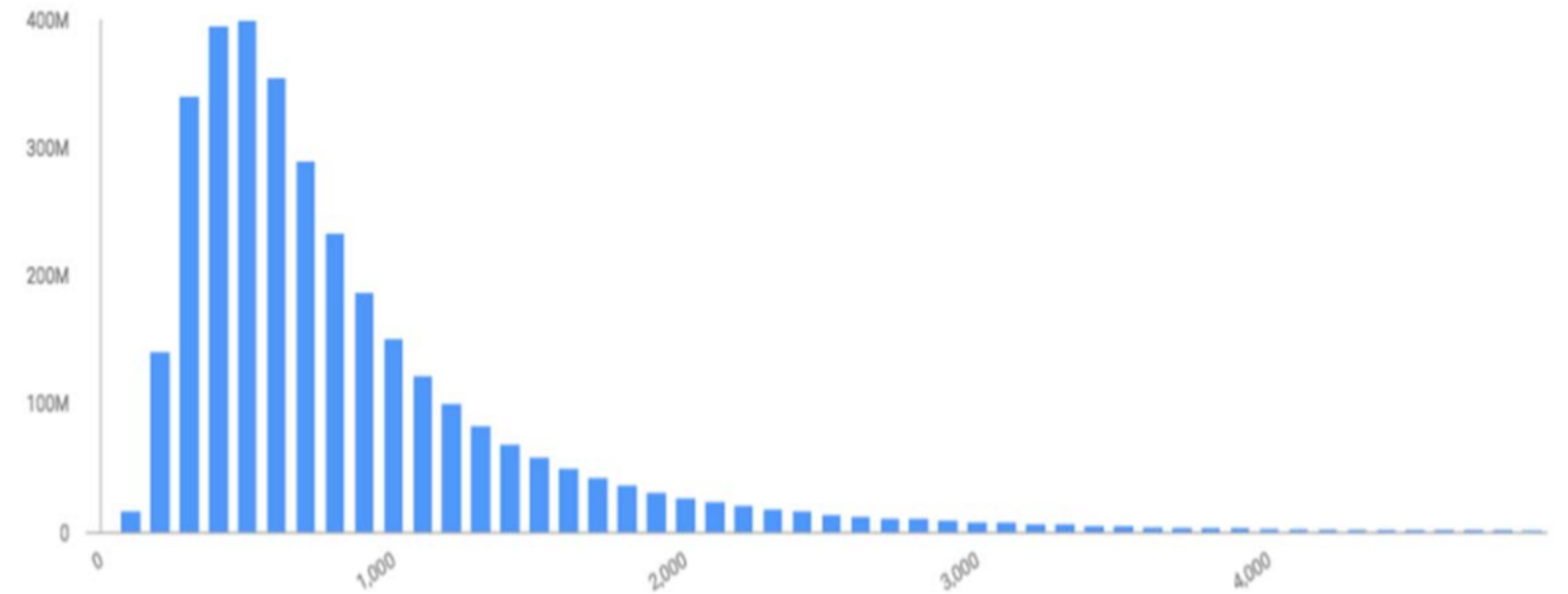
Tail latency is a problem that is well understood that it affects user experience

| | 50%ile latency | 95%ile latency | 99%ile latency |
|----------------------------------|-----------------------|-----------------------|-----------------------|
| One random leaf finishes | 1ms | 5ms | 10ms |
| 95% of all leaf requests finish | 12ms | 32ms | 70ms |
| 100% of all leaf requests finish | 40ms | 87ms | 140ms |

Single Request Completion time (Source: The Tail at Scale)

6. Thou shalt not ignore tails in user experience.

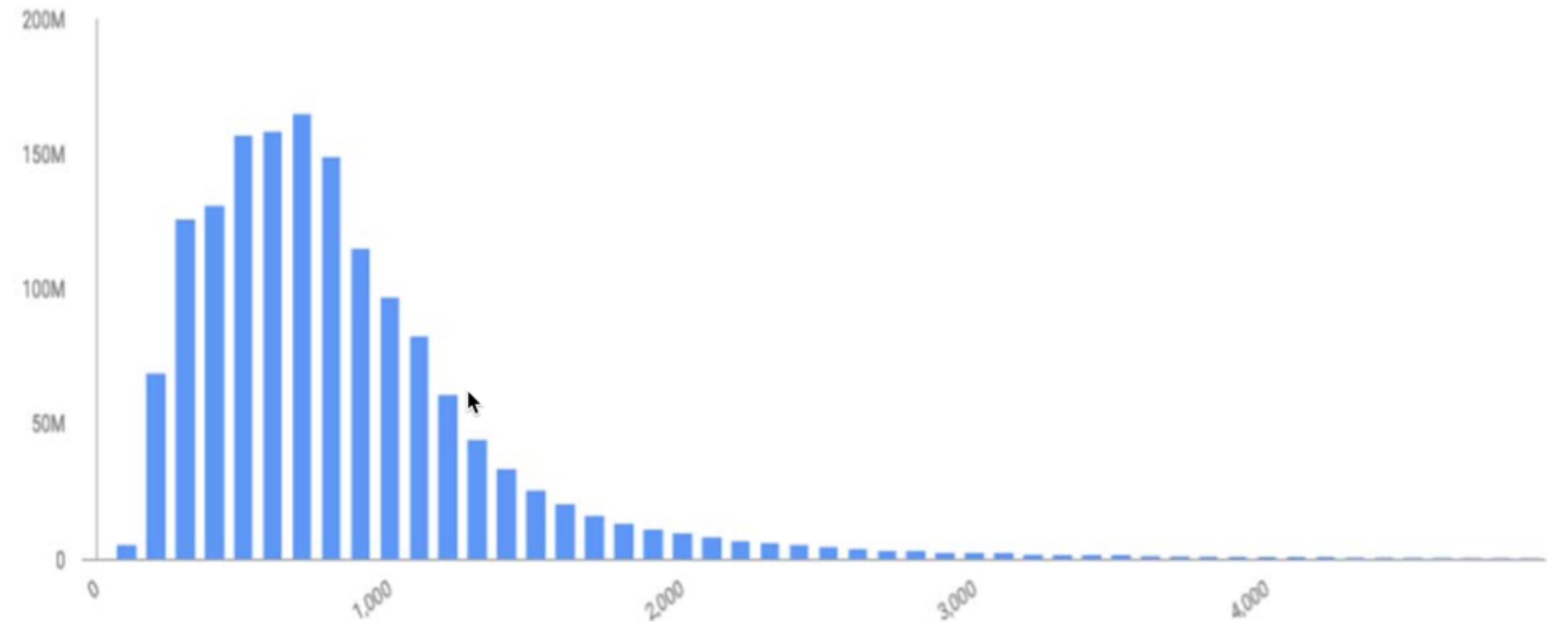
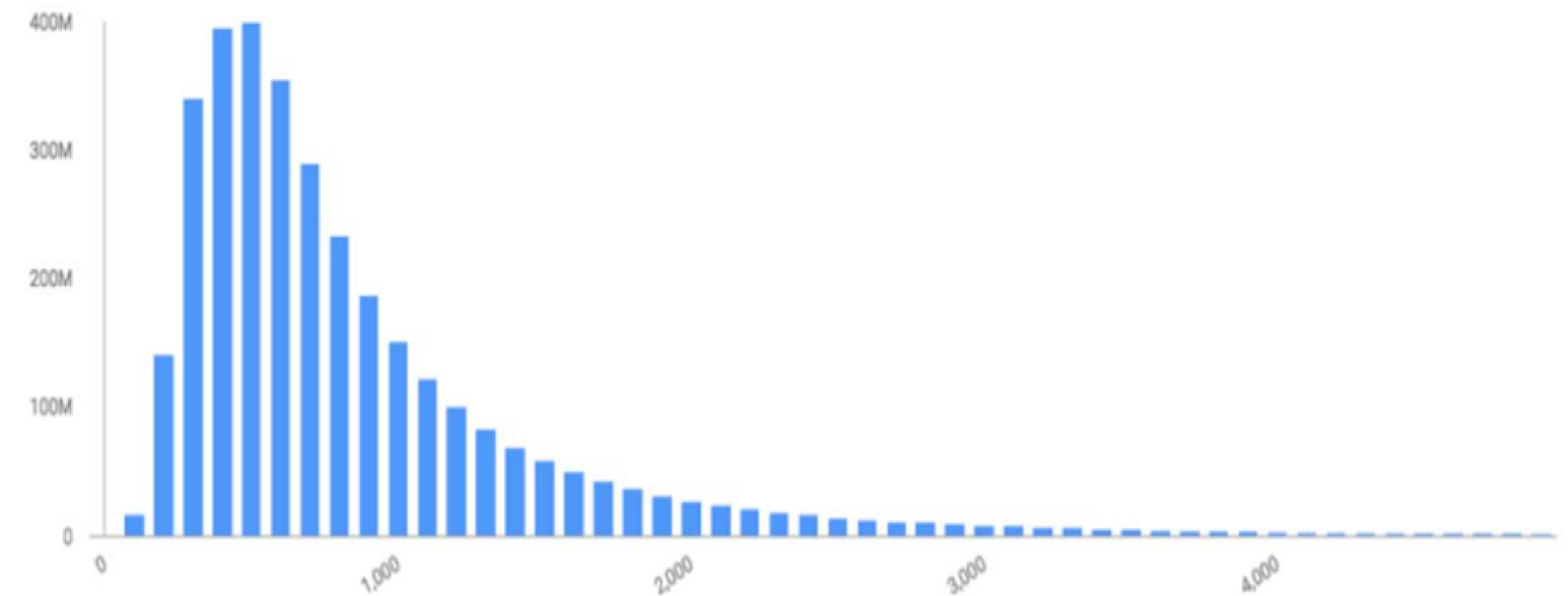
Tail latency is a problem that is well understood that it affects user experience



6. Thou shalt not ignore tails in user experience.

Tail latency is a problem that is well understood that it affects user experience

Little do researchers realize that mobile applications also suffer from long tail latency issues, similar to datacenter applications.

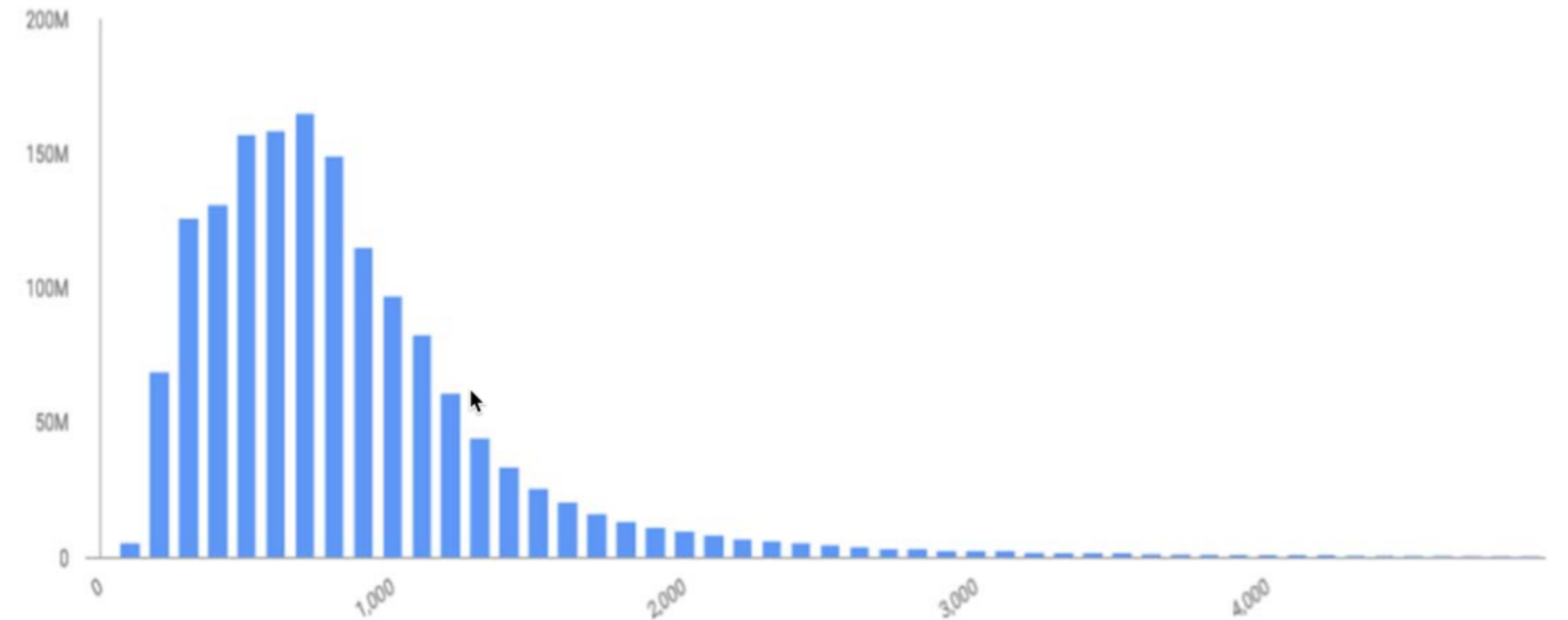
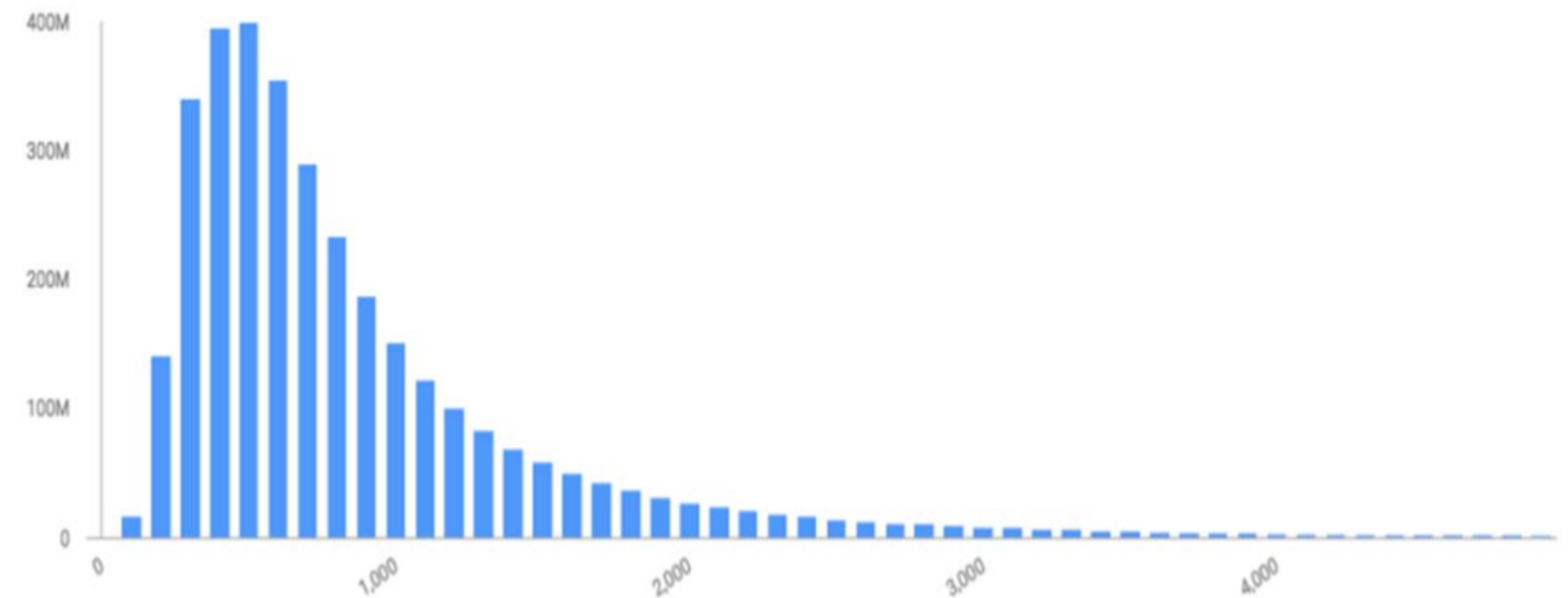


6. Thou shalt not ignore tails in user experience.

Tail latency is a problem that is well understood that it affects user experience

Little do researchers realize that mobile applications also suffer from long tail latency issues, similar to datacenter applications.

There are multiple sources of variation in a mobile device: thermal throttling, dynamic recompilation, garbage collection, background killing of applications, Android and kernel scheduling differences, and so on.



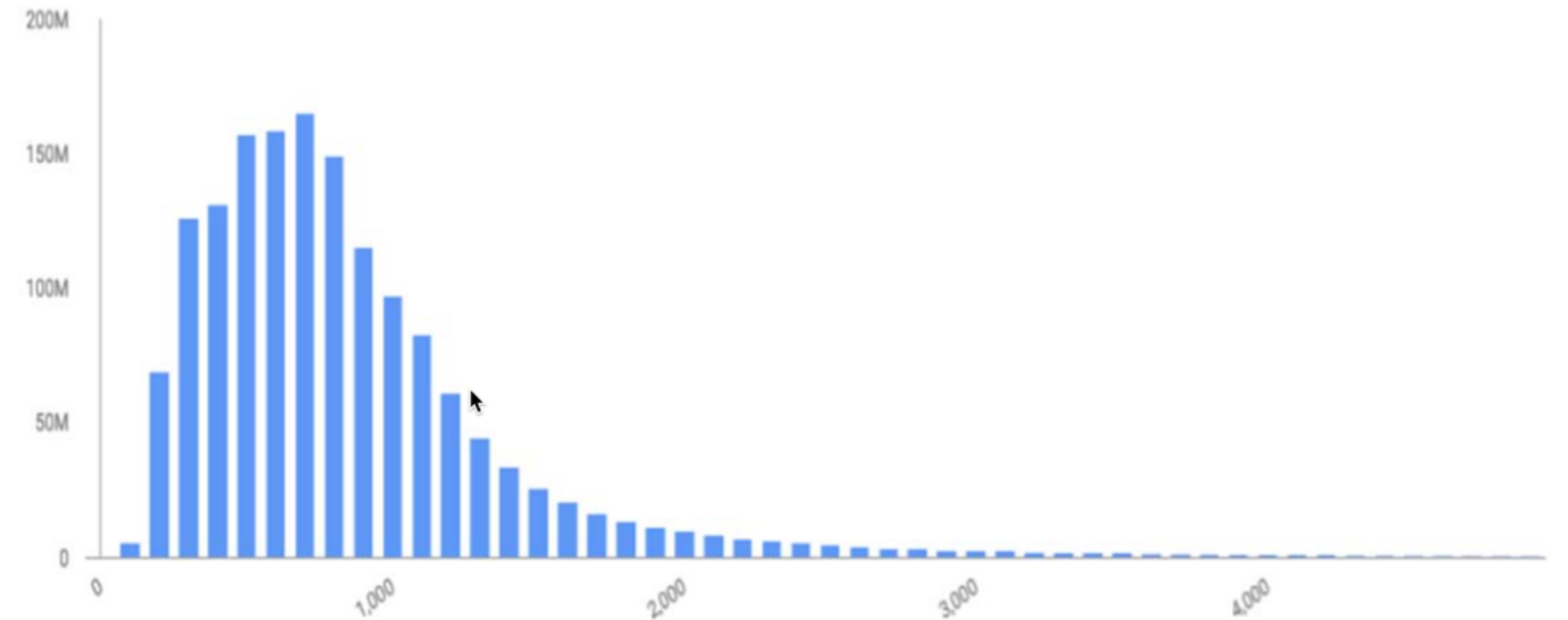
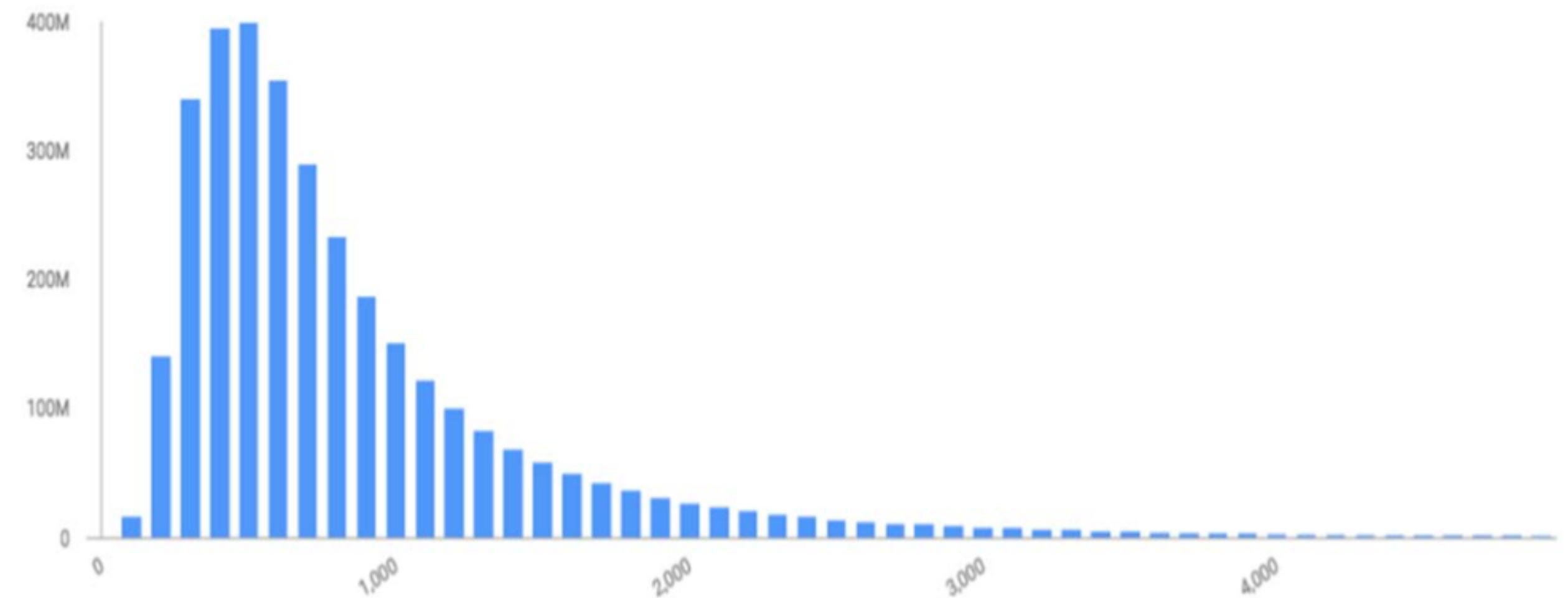
6. Thou shalt not ignore tails in user experience.

Tail latency is a problem that is well understood that it affects user experience

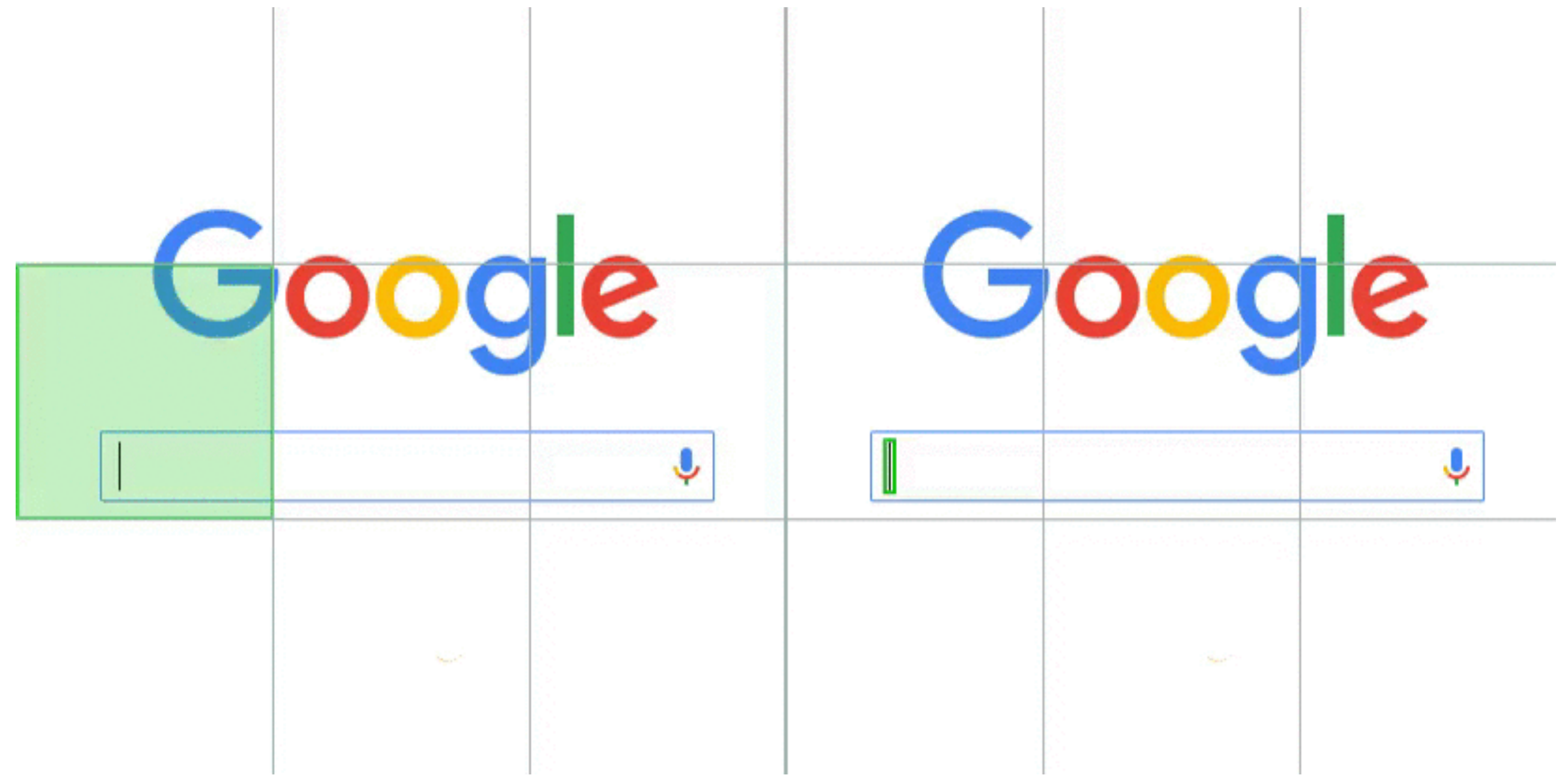
Little do researchers realize that mobile applications also suffer from long tail latency issues, similar to datacenter applications.

There are multiple sources of variation in a mobile device: thermal throttling, dynamic recompilation, garbage collection, background killing of applications, Android and kernel scheduling differences, and so on.

👉 It is important to report the distribution or assume the worst case and report results and conduct analysis accordingly.

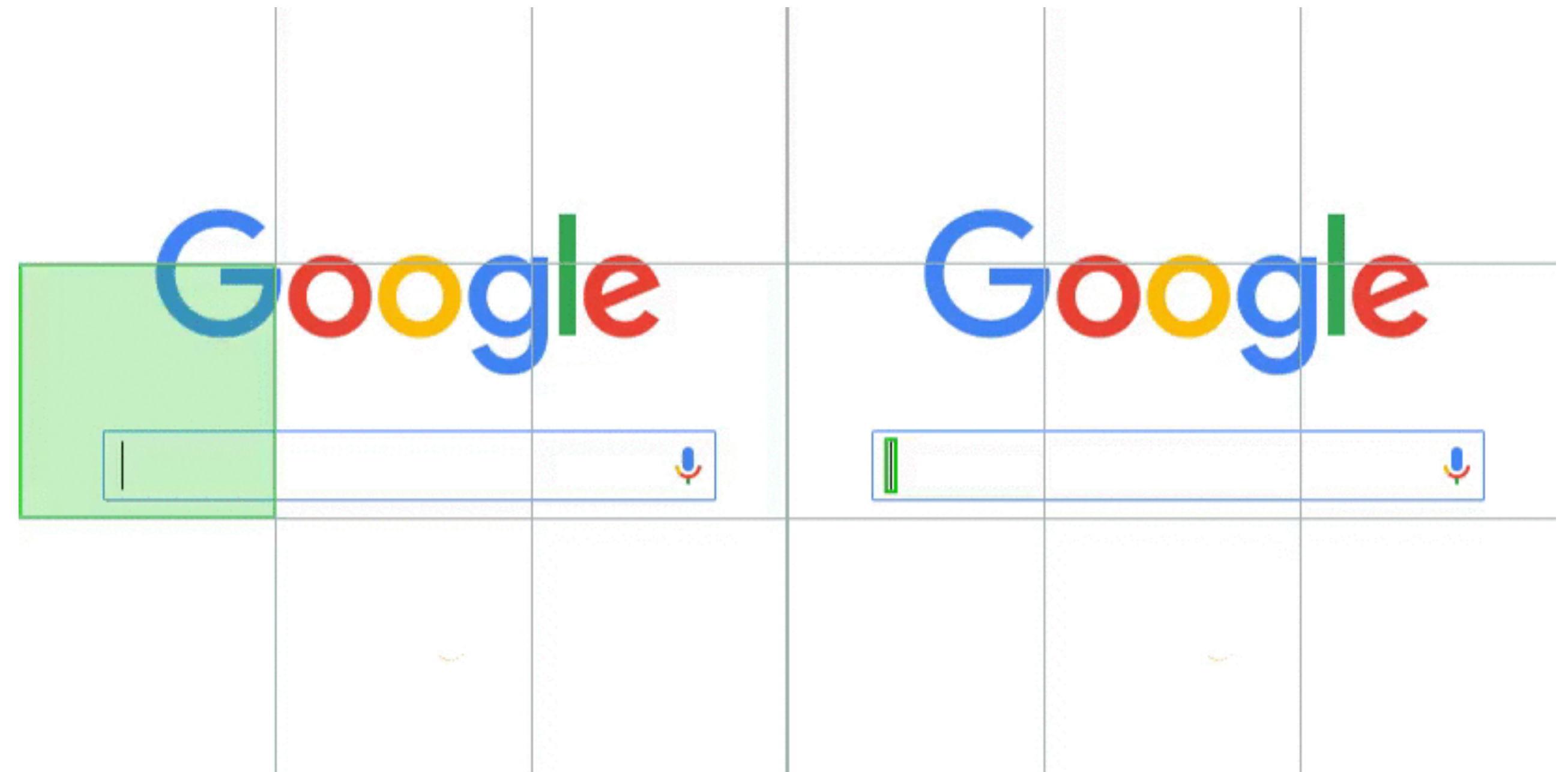


7. Thou shalt not assume software is hardware-agnostic.



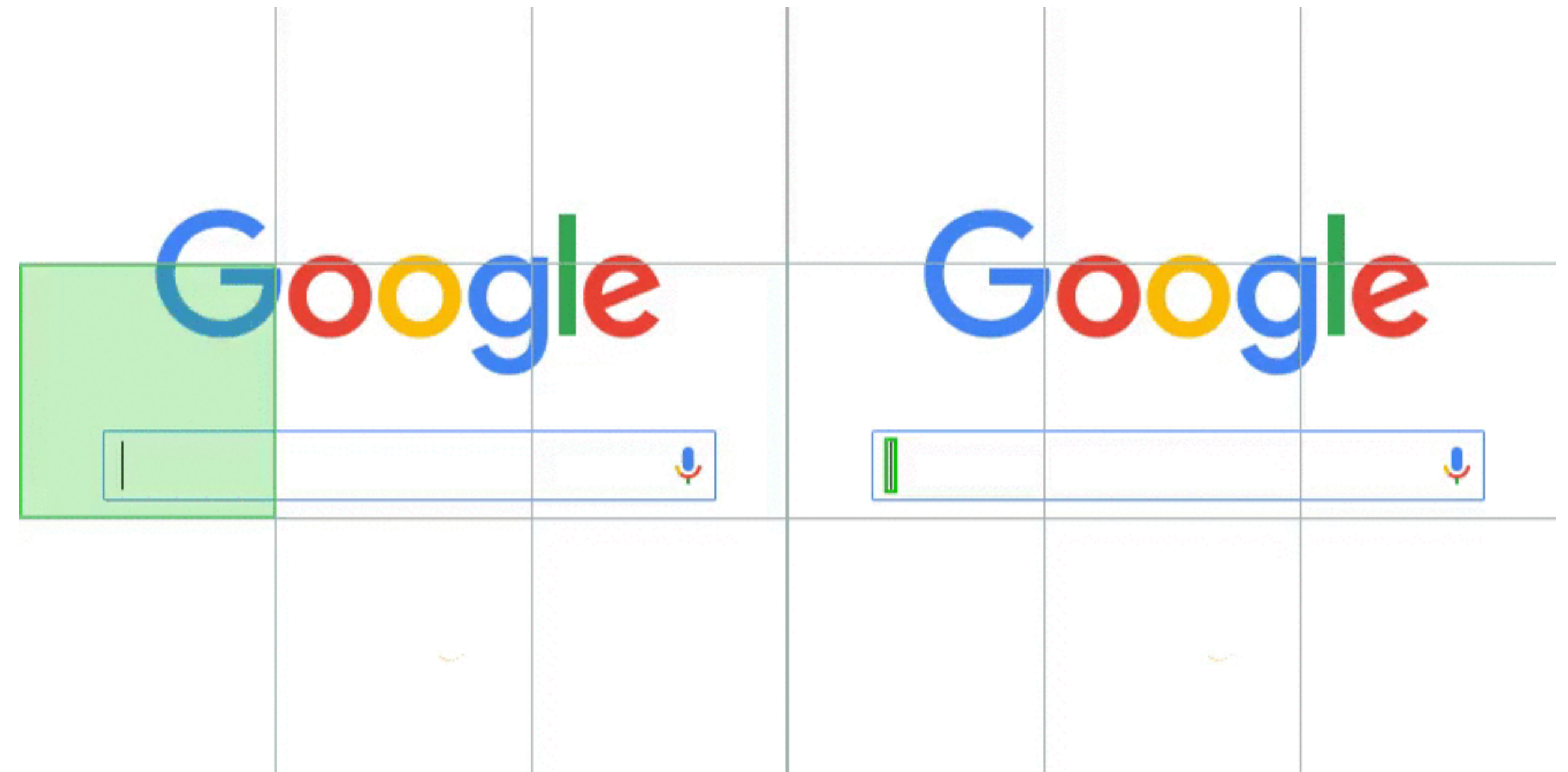
7. Thou shalt not assume software is hardware-agnostic.

Mobile applications are written with user experience and hardware capabilities in mind. There are more than 24,093 distinct Android devices in the world, and the performance capabilities of these devices vary drastically.



7. Thou shalt not assume software is hardware-agnostic.

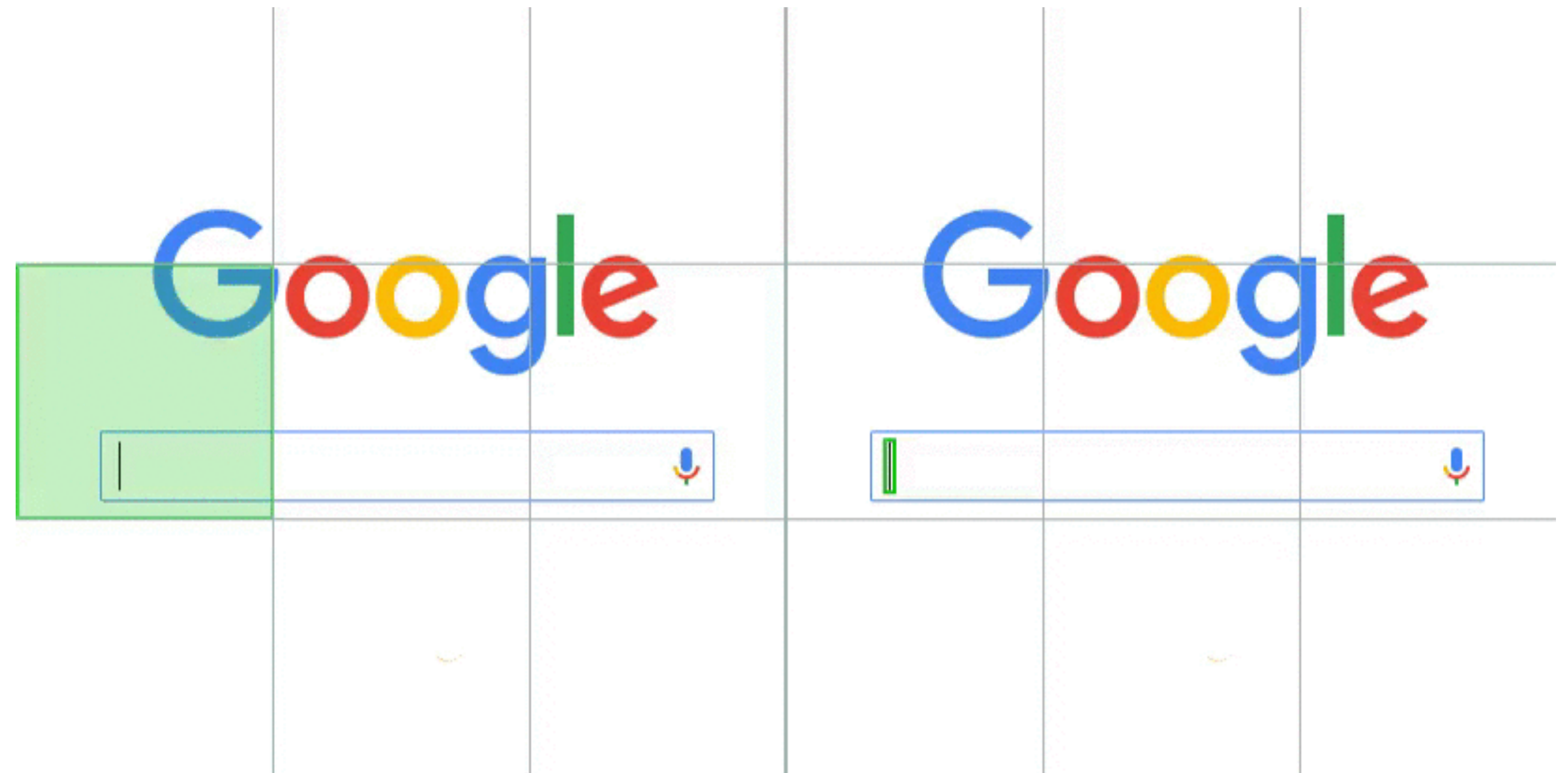
Mobile applications are written with user experience and hardware capabilities in mind. There are more than 24,093 distinct Android devices in the world, and the performance capabilities of these devices vary drastically.



7. Thou shalt not assume software is hardware-agnostic.

Mobile applications are written with user experience and hardware capabilities in mind. There are more than 24,093 distinct Android devices in the world, and the performance capabilities of these devices vary drastically.

Different mobile devices support different codecs (or codec variations) in their hardware decode engine, and so the browser must adapt to them, with a fallback to software decoding in the event that the codec used by a video isn't supported in hardware.

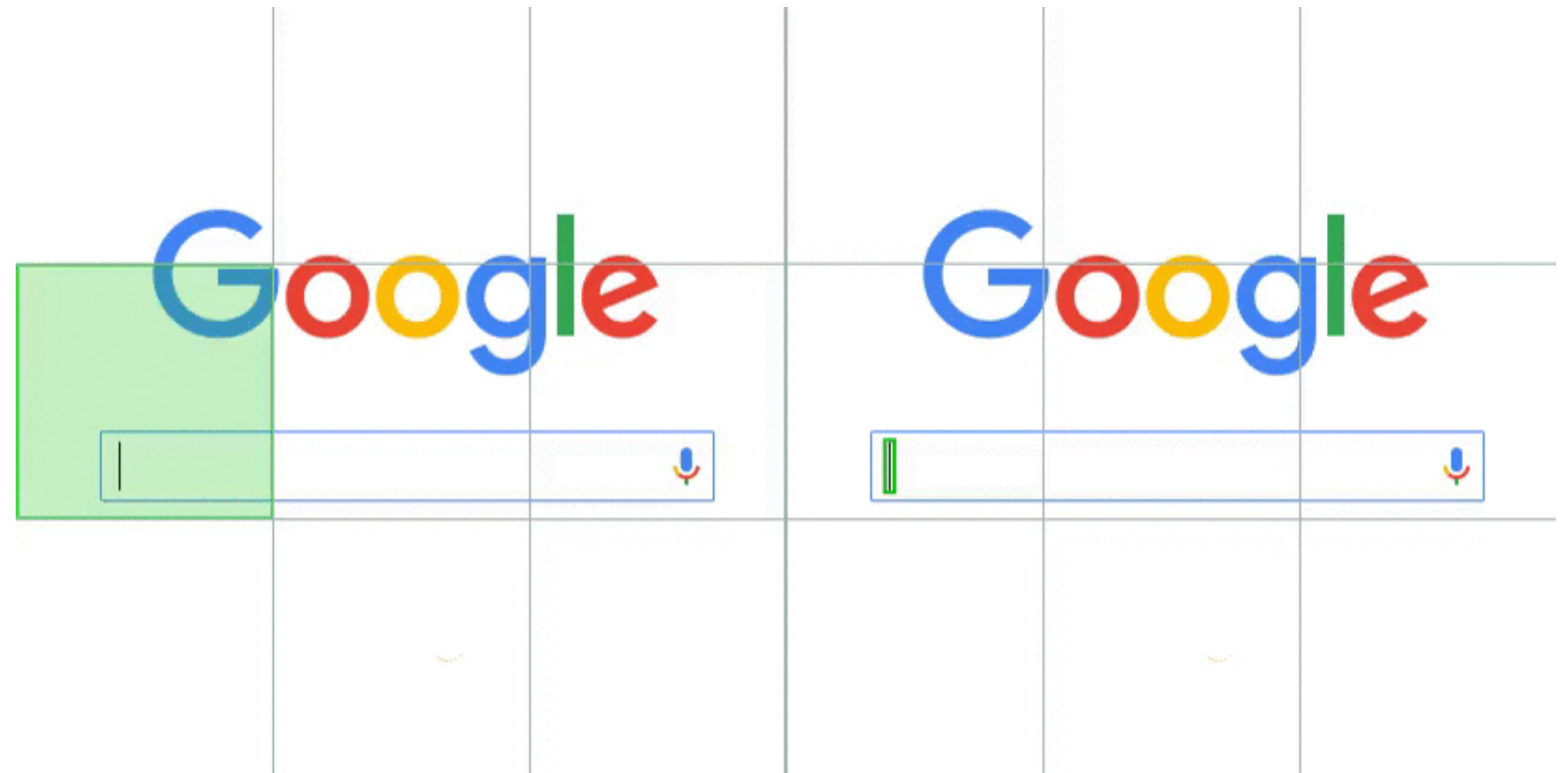


7. Thou shalt not assume software is hardware-agnostic.

Mobile applications are written with user experience and hardware capabilities in mind. There are more than 24,093 distinct Android devices in the world, and the performance capabilities of these devices vary drastically.

Different mobile devices support different codecs (or codec variations) in their hardware decode engine, and so the browser must adapt to them, with a fallback to software decoding in the event that the codec used by a video isn't supported in hardware.

➡ Given the heterogeneity in mobile hardware, one needs to understand that the same application can be hard-coded differently on different devices to perform differently.



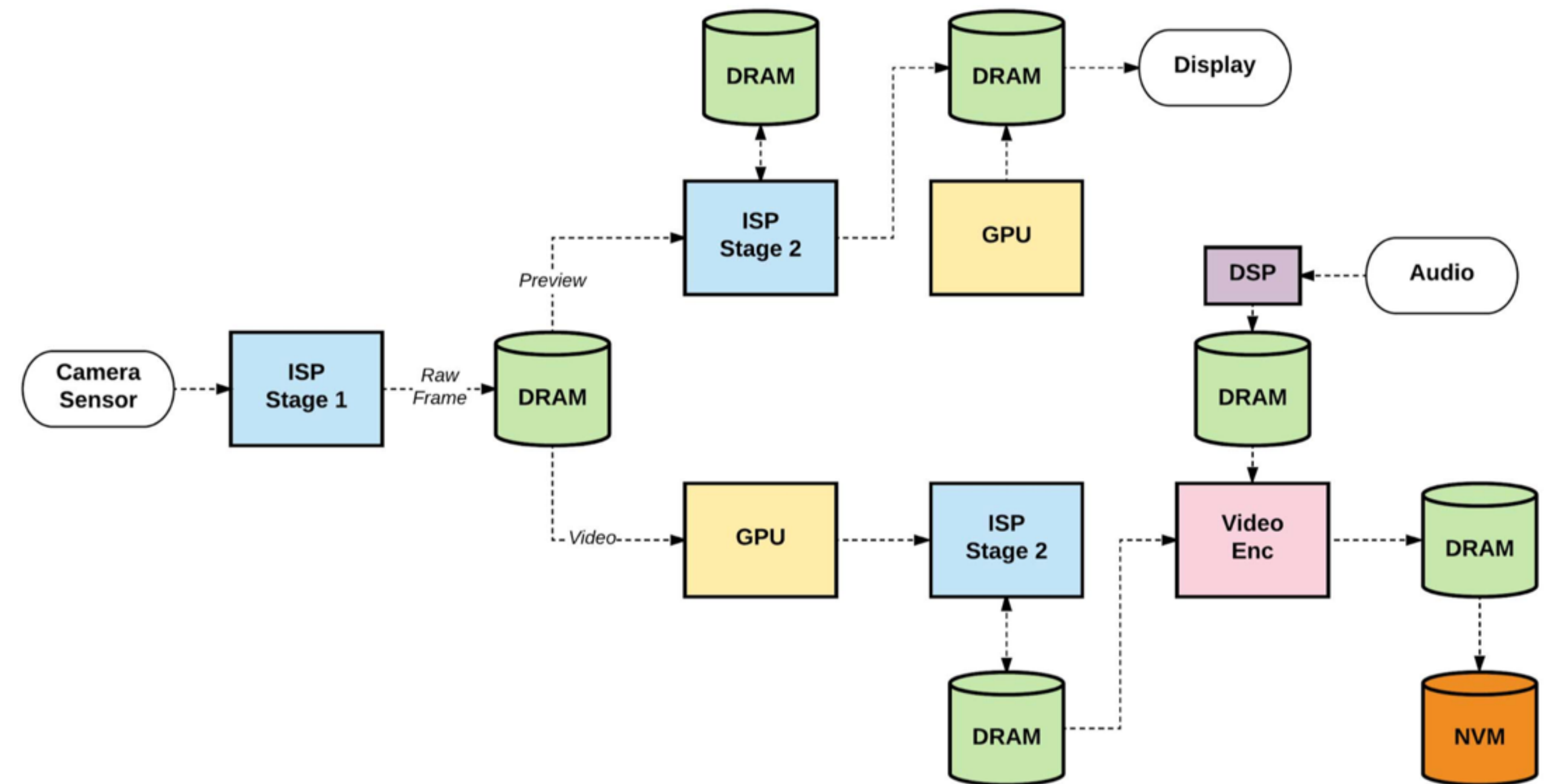
8. Thou shalt not disregard
the accelerator “IP” blocks.

8. Thou shalt not disregard the accelerator “IP” blocks.

The architecture community has long focused on general-purpose CPUs and GPUs. However, unlike CPUs and GPUs that have been optimized for decades, SoCs are, by definition, a modular collection of special-purpose compute, communication, and storage IP units.

8. Thou shalt not disregard the accelerator “IP” blocks.

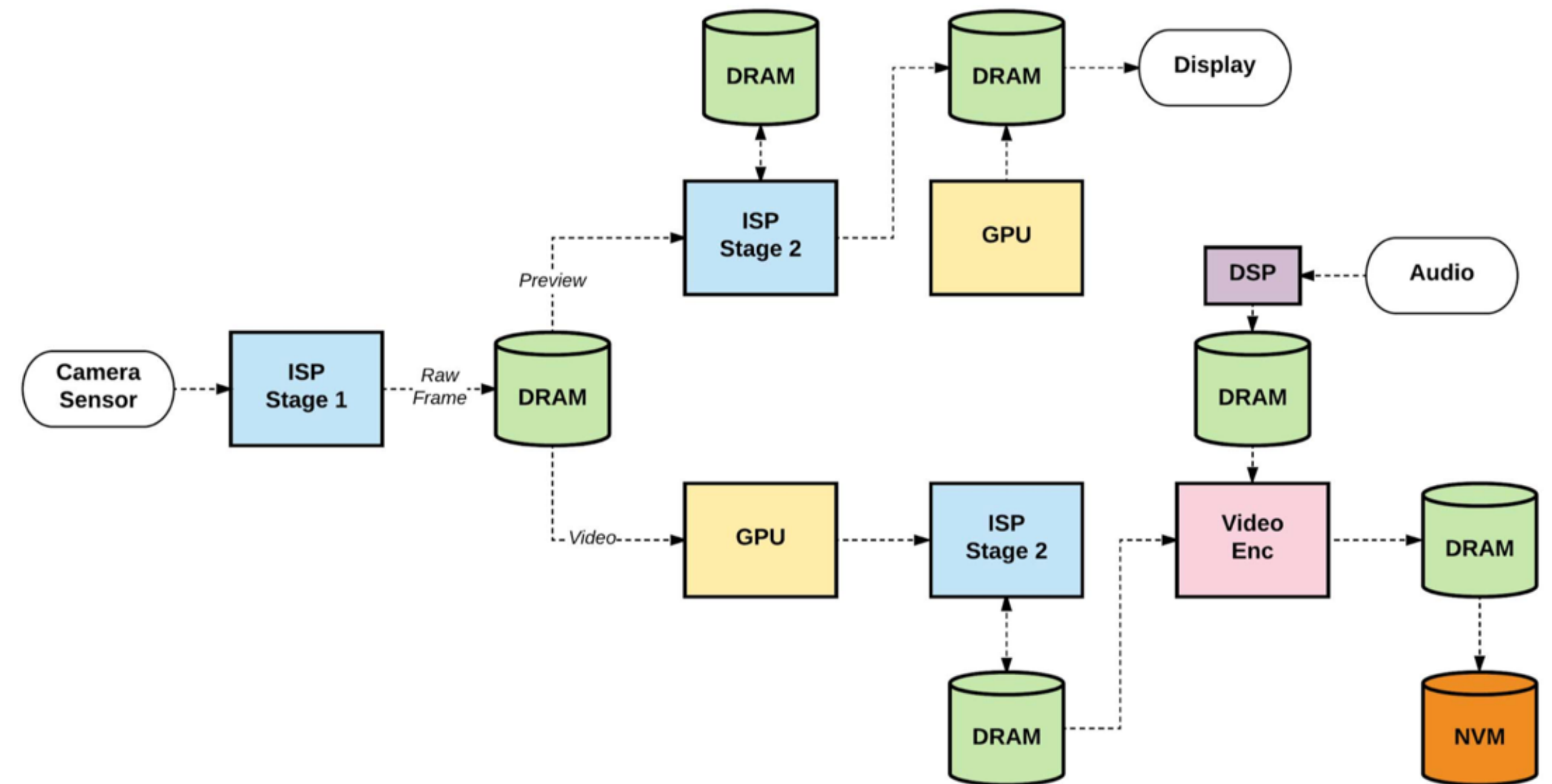
The architecture community has long focused on general-purpose CPUs and GPUs. However, unlike CPUs and GPUs that have been optimized for decades, SoCs are, by definition, a modular collection of special-purpose compute, communication, and storage IP units.



8. Thou shalt not disregard the accelerator “IP” blocks.

The architecture community has long focused on general-purpose CPUs and GPUs. However, unlike CPUs and GPUs that have been optimized for decades, SoCs are, by definition, a modular collection of special-purpose compute, communication, and storage IP units.

In an SoC, multiple IP blocks are active at the same time and communicate frequently with each other over the Network-on-Chip (NOC) fabric. Often, the IPs are studied in isolation.

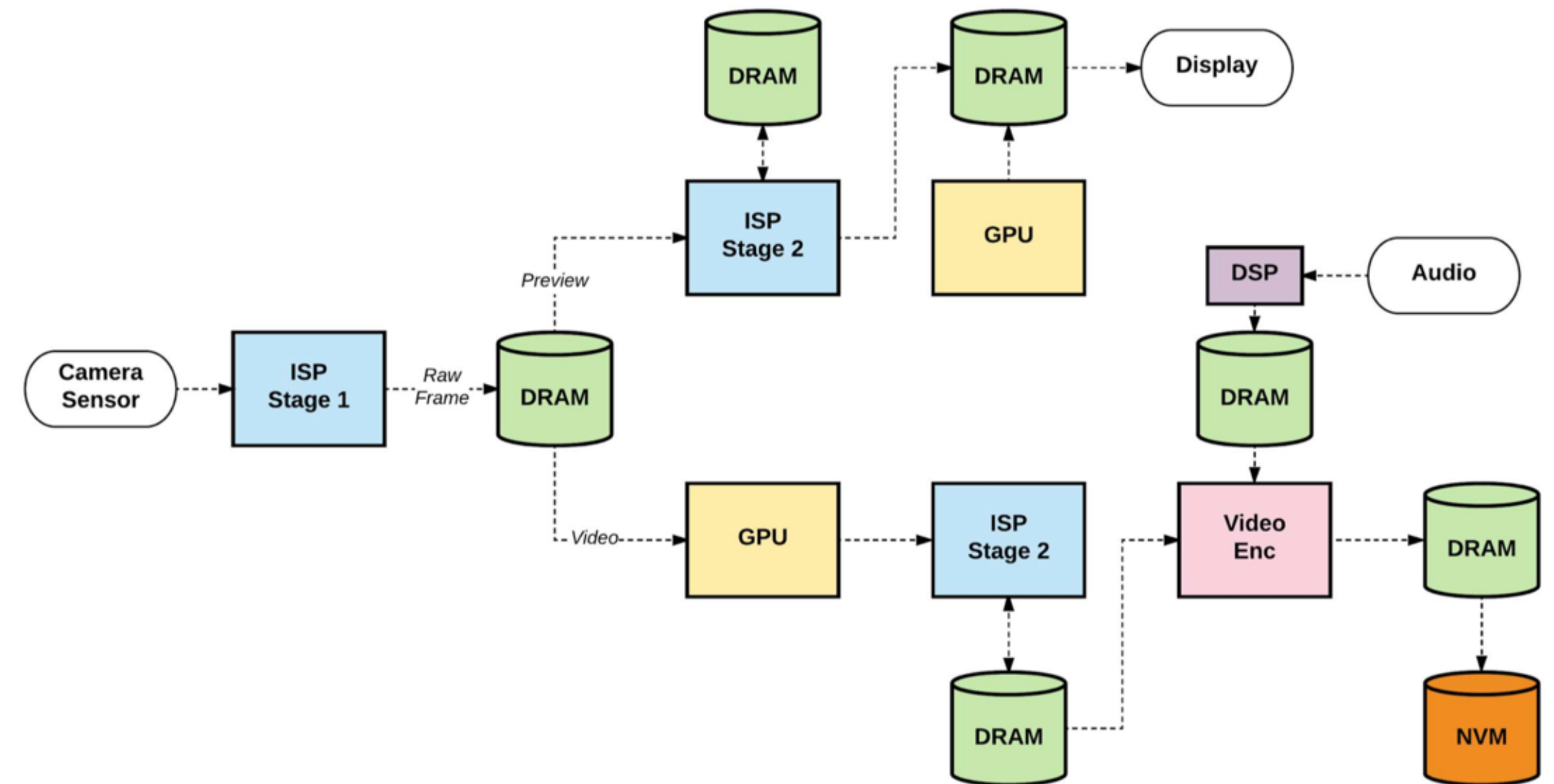


8. Thou shalt not disregard the accelerator “IP” blocks.

The architecture community has long focused on general-purpose CPUs and GPUs. However, unlike CPUs and GPUs that have been optimized for decades, SoCs are, by definition, a modular collection of special-purpose compute, communication, and storage IP units.

In an SoC, multiple IP blocks are active at the same time and communicate frequently with each other over the Network-on-Chip (NOC) fabric. Often, the IPs are studied in isolation.

➡ Therefore, architects are strongly encouraged to consider the role of the various IP blocks in the process of evaluating a mobile system, as there are numerous opportunities for research and development of novel solutions to enhance mobile processor performance



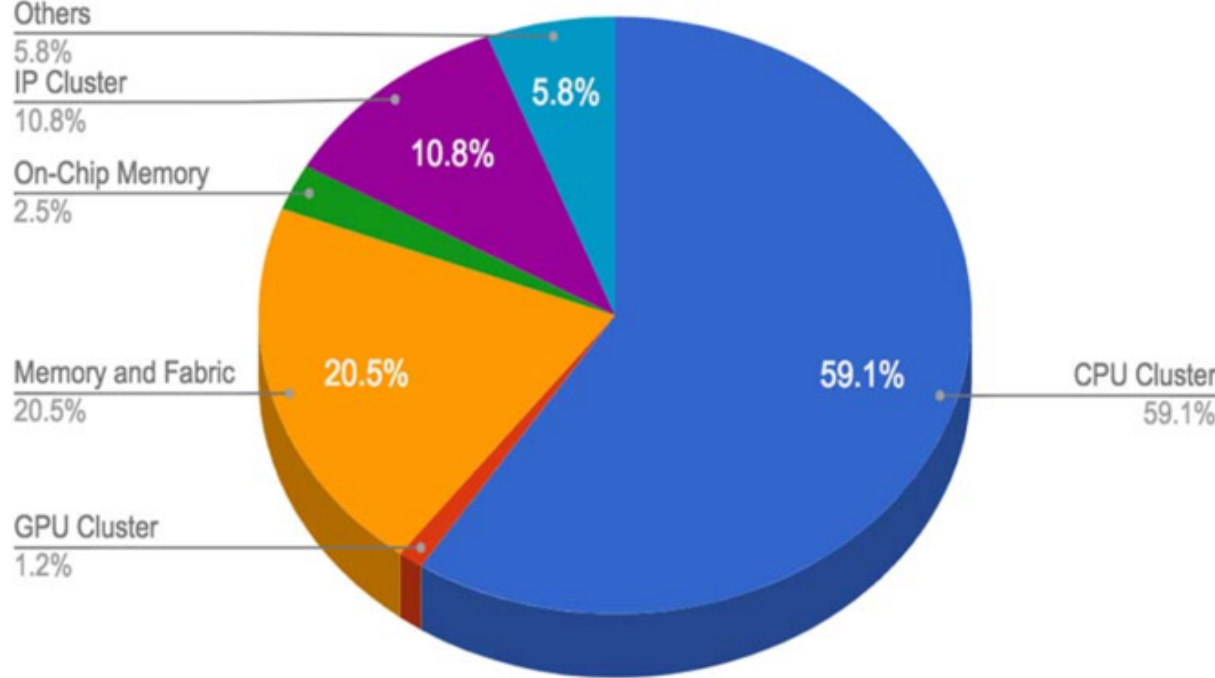
9. Thou shalt not turn a blind eye to energy and thermal.

9. Thou shalt not turn a blind eye to energy and thermal.

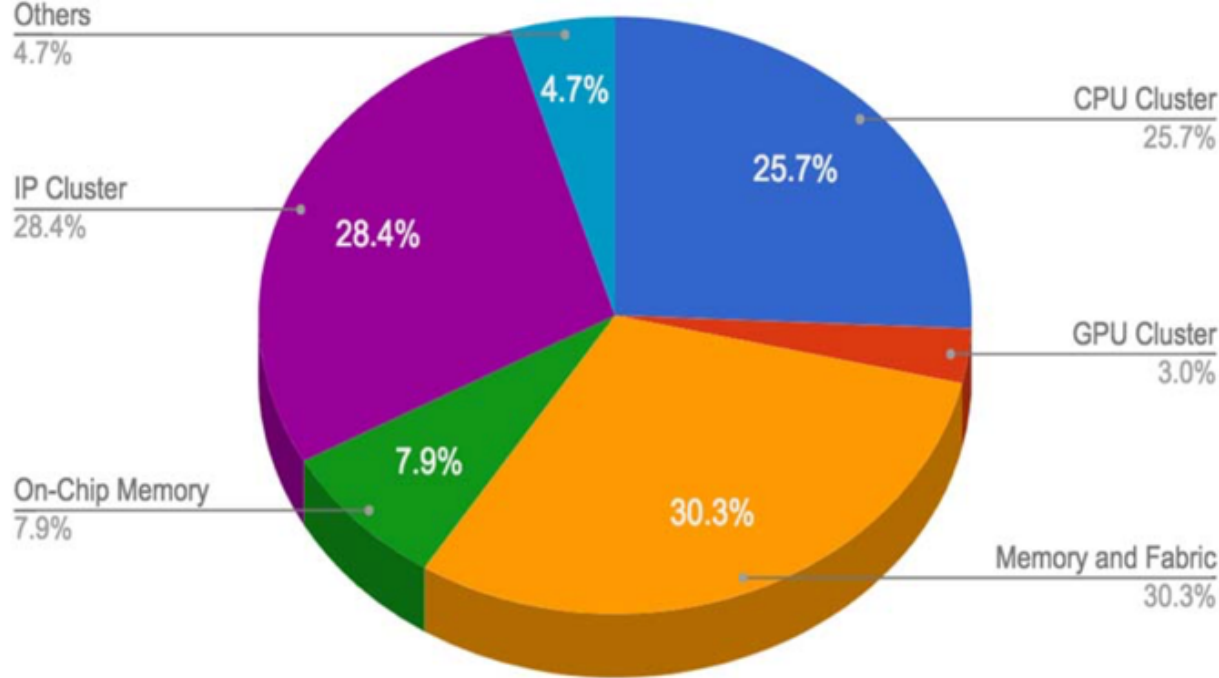
No discussion on mobile is ever complete without power discussion. Little known fact is that there is no Moore's law for batteries.

9. Thou shalt not turn a blind eye to energy and thermal.

No discussion on mobile is ever complete without power discussion. Little known fact is that there is no Moore's law for batteries.



Web browsing

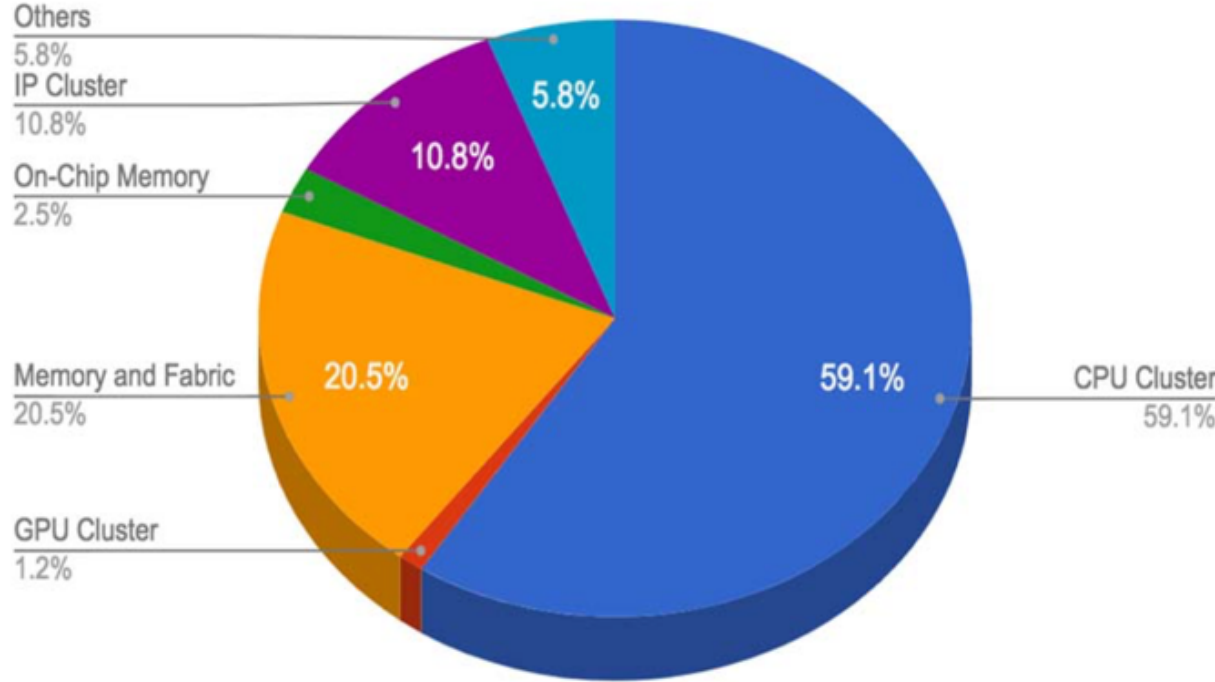


4K video recording

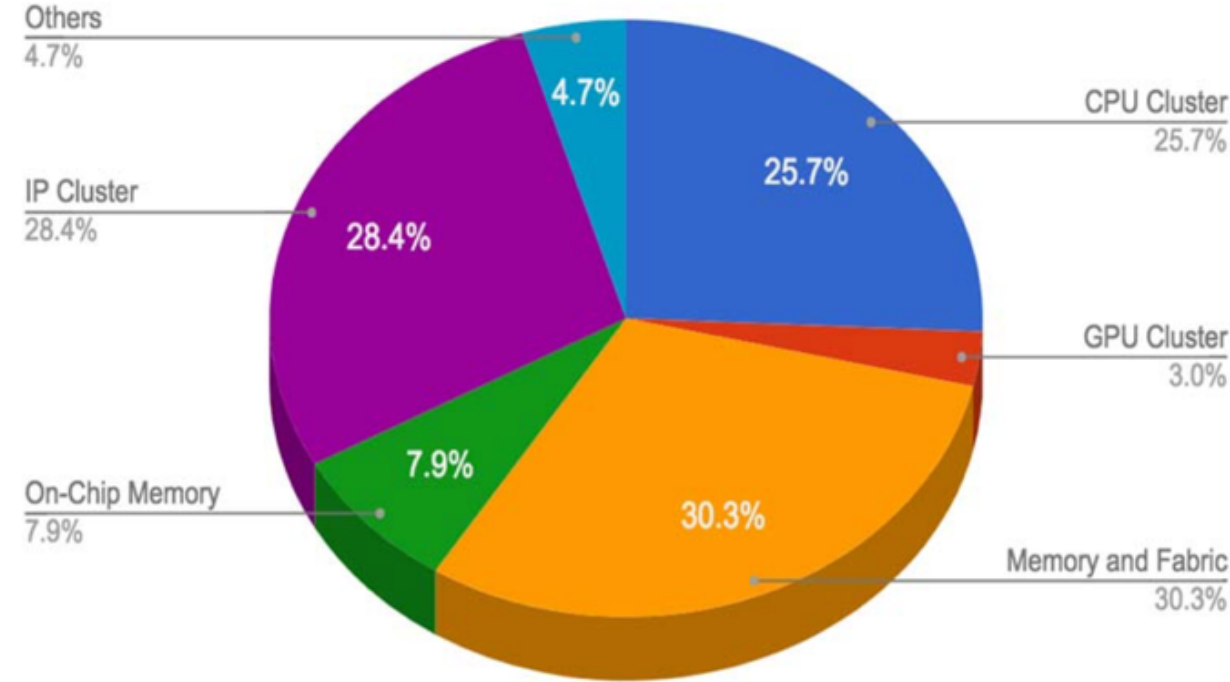
9. Thou shalt not turn a blind eye to energy and thermal.

No discussion on mobile is ever complete without power discussion. Little known fact is that there is no Moore's law for batteries.

Temperature cooling is a major problem because all humans have a low tolerance for high surface temperatures. We can handle at most 45 degrees Celsius before our skin starts experiencing a painful sensation.



Web browsing



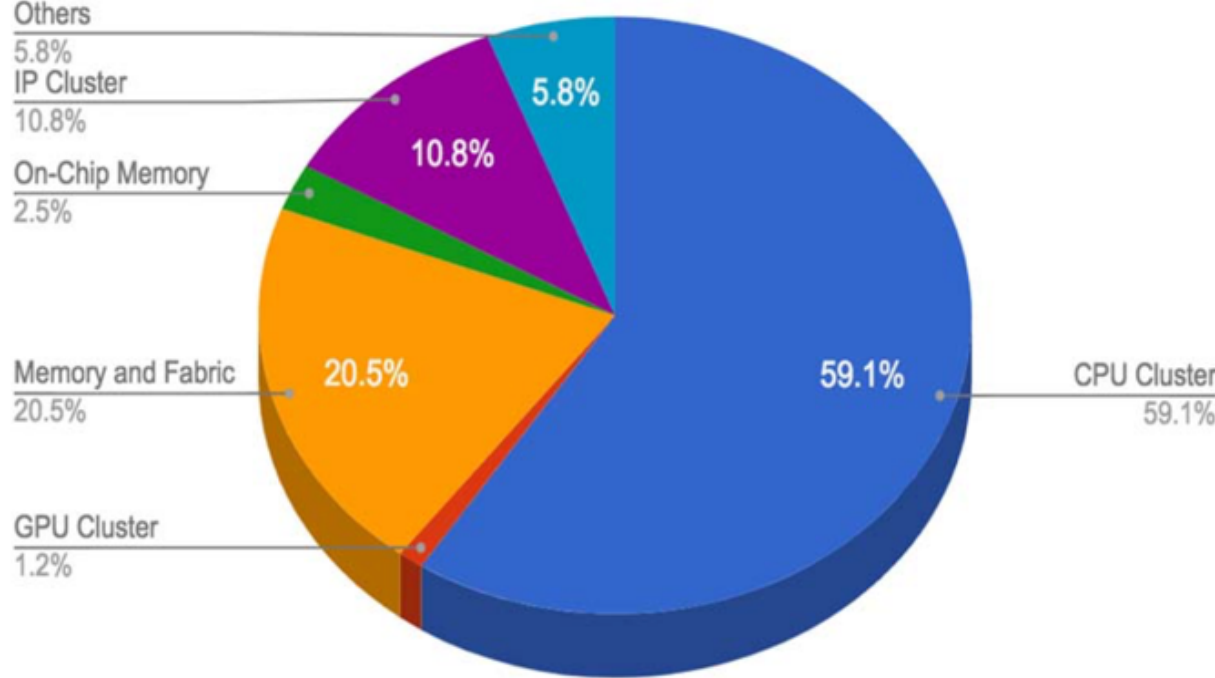
4K video recording

9. Thou shalt not turn a blind eye to energy and thermal.

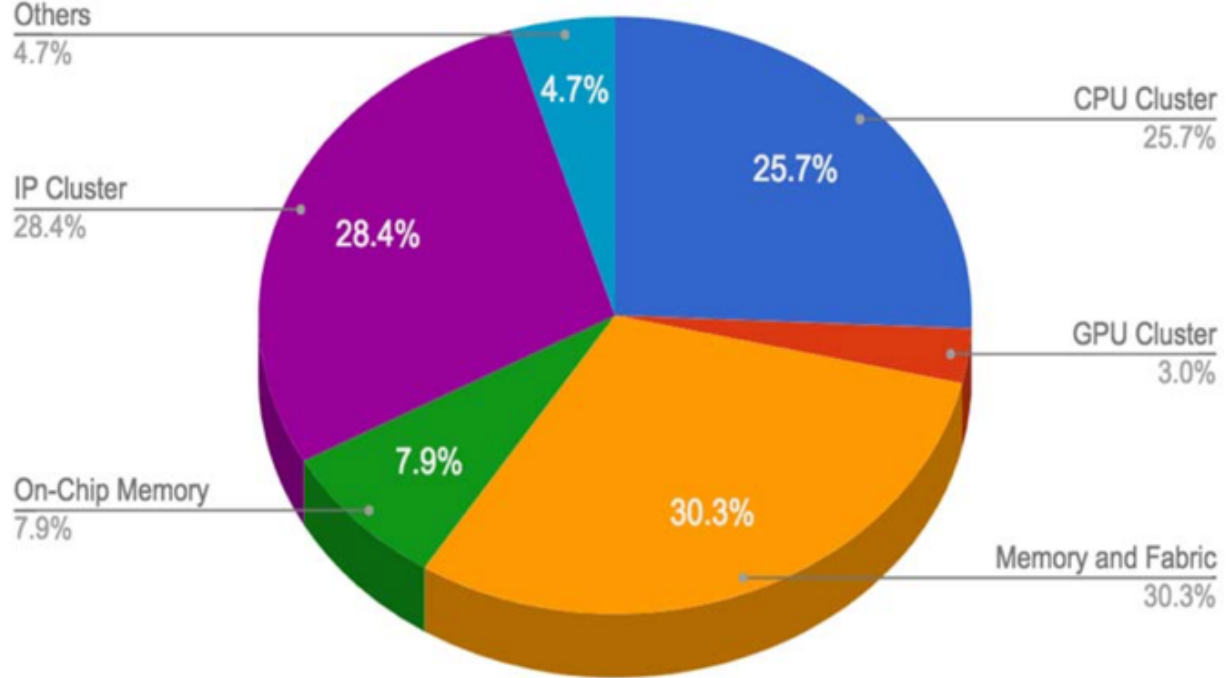
No discussion on mobile is ever complete without power discussion. Little known fact is that there is no Moore's law for batteries.

Temperature cooling is a major problem because all humans have a low tolerance for high surface temperatures. We can handle at most 45 degrees Celsius before our skin starts experiencing a painful sensation.

➡ Measuring the power peaks and energy consumption of new ideas is necessary given that battery and heat dissipation are first-order constraints for mobile systems.



Web browsing



4K video recording

10. Thou shalt not presume
this list is complete.

10. Thou shalt not presume this list is complete.

Mobile computing is still a rather rapidly and continuously evolving ecosystem, and the processor hardware requirements will continue to evolve as the time progresses.

10. Thou shalt not presume this list is complete.

Mobile computing is still a rather rapidly and continuously evolving ecosystem, and the processor hardware requirements will continue to evolve as the time progresses.

It would be only all too presumptuous for anyone to assume that any one set of commandments can ever be complete.



10. Thou shalt not presume this list is complete.

Mobile computing is still a rather rapidly and continuously evolving ecosystem, and the processor hardware requirements will continue to evolve as the time progresses.

It would be only all too presumptuous for anyone to assume that any one set of commandments can ever be complete.

So, we must acknowledge that the current list is incomplete and that the aforementioned commandments should be improved upon over time as mobile computing evolves.



10. Thou shalt not presume this list is complete.

Mobile computing is still a rather rapidly and continuously evolving ecosystem, and the processor hardware requirements will continue to evolve as the time progresses.

It would be only all too presumptuous for anyone to assume that any one set of commandments can ever be complete.

So, we must acknowledge that the current list is incomplete and that the aforementioned commandments should be improved upon over time as mobile computing evolves.

➡ Look beyond today's mobile computing systems (i.e., smartphones) and embrace the future (e.g., AR headsets that add to phones).



Thank You to 

To find out more... *“Vijay Janapa Reddi, Hongil Yoon, and Allan Knies. "Two billion devices and counting." IEEE Micro 38, no. 1 (2018): 6-21.”*

Two Billion Devices and Counting

An Industry Perspective on the State of Mobile Computer Architecture

Vijay Janapa Reddi, Hongil Yoon, and Allan Knies
Google

Mobile computing has grown drastically over the past decade. Despite the rapid pace of advancements, mobile device understanding, benchmarking, and evaluation are still in their infancies, both in industry and academia. This article presents an industry perspective on the challenges facing mobile computer architecture, specifically involving mobile workloads, benchmarking, and experimental methodology, with the hope of fostering new research within the community to address pending problems. These challenges pose a threat to the systematic development of future mobile systems, which, if addressed, can elevate the entire mobile ecosystem to the next level.

Mobile devices have come a long way from the first portable cellular phone developed by Motorola in 1973. Most modern smartphones are good enough to replace desktop computers. A smartphone today has enough computing power to be on par with the fastest supercomputers from the 1990s. For instance, the Qualcomm Adreno 540 GPU found in the latest smartphones has a peak compute capability of more than 500 Gflops, putting it in competition with supercomputers that were on the TOP500 list in the early to mid-1990s.

Mobile computing has experienced an unparalleled level of growth over the past decade. At the time of this writing, there are more than 2 billion mobile devices in the world.¹ But perhaps even more importantly, mobile phones are showing no signs of slowing in uptake. In fact, smartphone adoption rates are on the rise. The number of devices is rising as mobile device penetration increases in markets like India and China. It is anticipated that the number of mobile subscribers will grow past 6 billion in the coming years.² As Figure 1 shows, while the Western European and North American markets are reaching saturation, the vast majority of growth is coming from countries in Asia. Given that only 35 percent of the world's population has thus far adopted mobile technology, there is still significant room for growth and innovation.