Building More Secure Software with Memory-Safe Programming Languages

SHANE MILLER (SHE/HER)

NONRESIDENT SENIOR FELLOW, ATLANTIC COUNCIL

DISTINGUISHED ADVISOR, RUST FOUNDATION

Agenda

- What is “memory safety”?
- Success stories
- Memory safe choices and tradeoffs

Home Expectations

Home Security

Data Security

Regulations

- Encryption
- Least Privilege
- Logging

Secure by Design

Secure by Design Tradeoffs

Flexibility -> Complexity -> Bugs

Memory Bugs

“Memory safety” means making it very difficult for developers to create memory bugs.

The Cost of Safety

Safer “unsafe”?

Definitions

Vulnerability

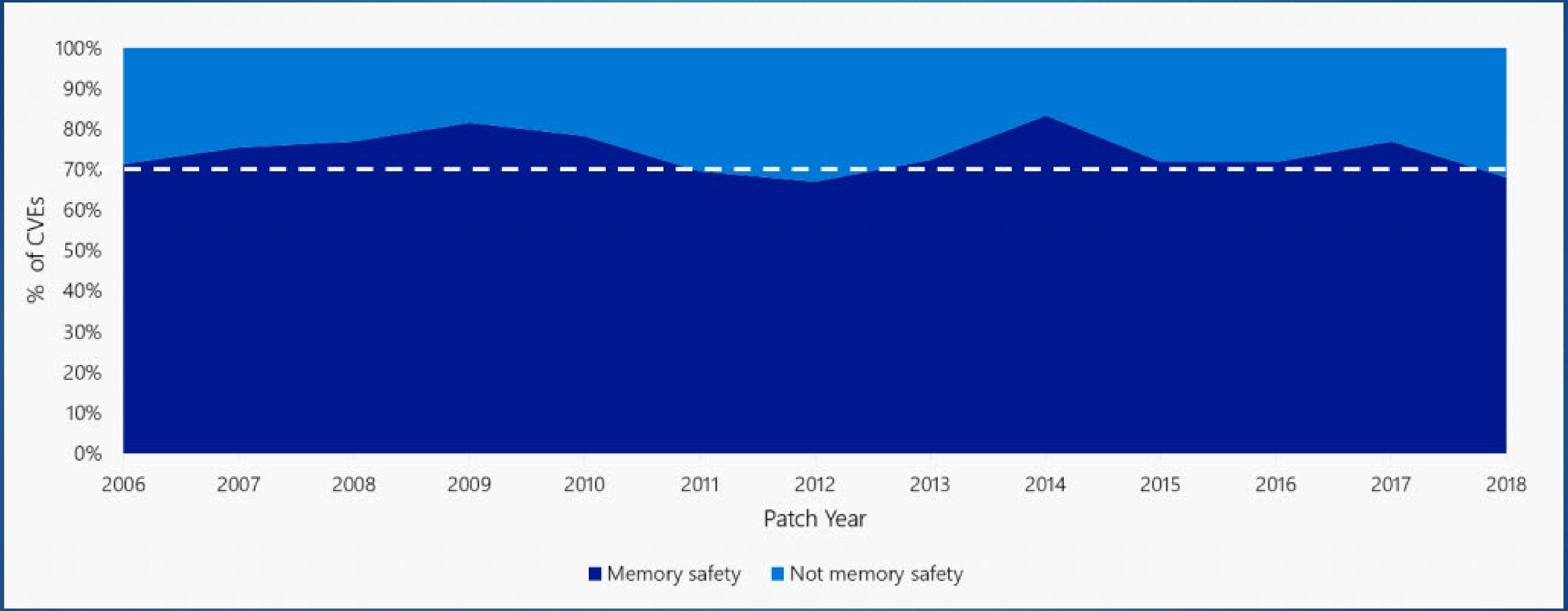
Exploit

Trends

- 35 million active software developers in 2023
- 26,448 security vulnerabilities in 2022
- 59% more critical vulnerabilities in 2022 than 2021
- 4,135 critical vulnerabilities in 2022

Critical Vulnerabilities

Microsoft Security Research Center



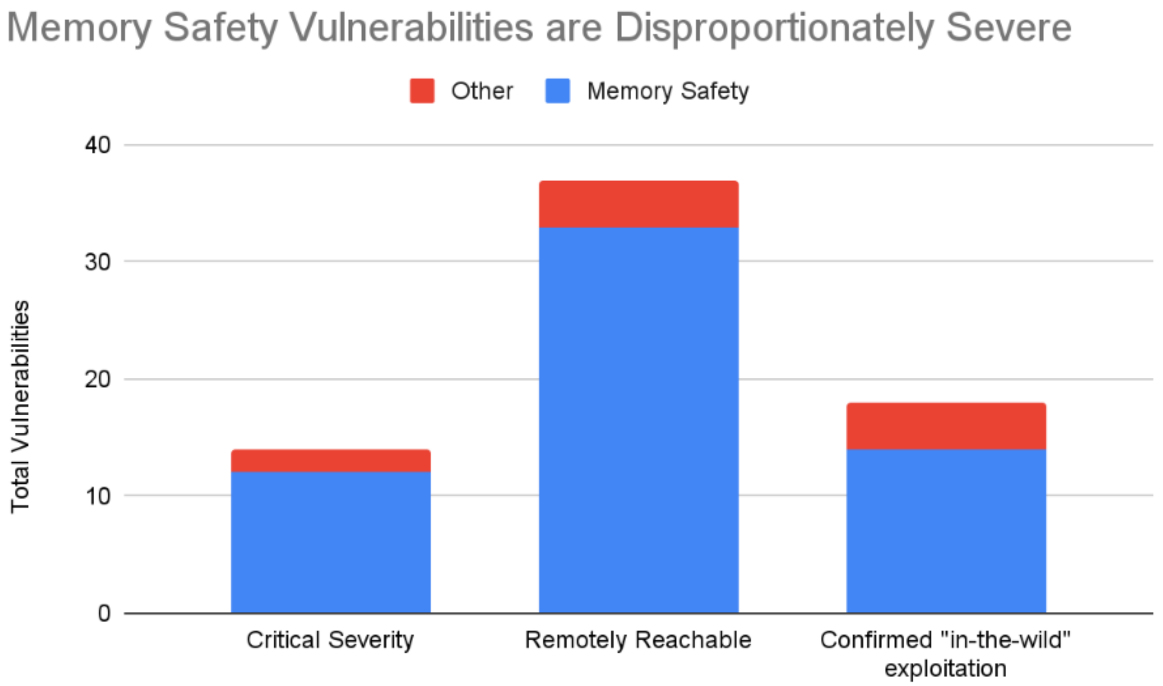
https://msrc.microsoft.com/blog/2019/07/a-proactive-approach-to-more-secure-code/

Google Android

- 86% of critical severity vulnerabilities were memory safety bugs in 2022

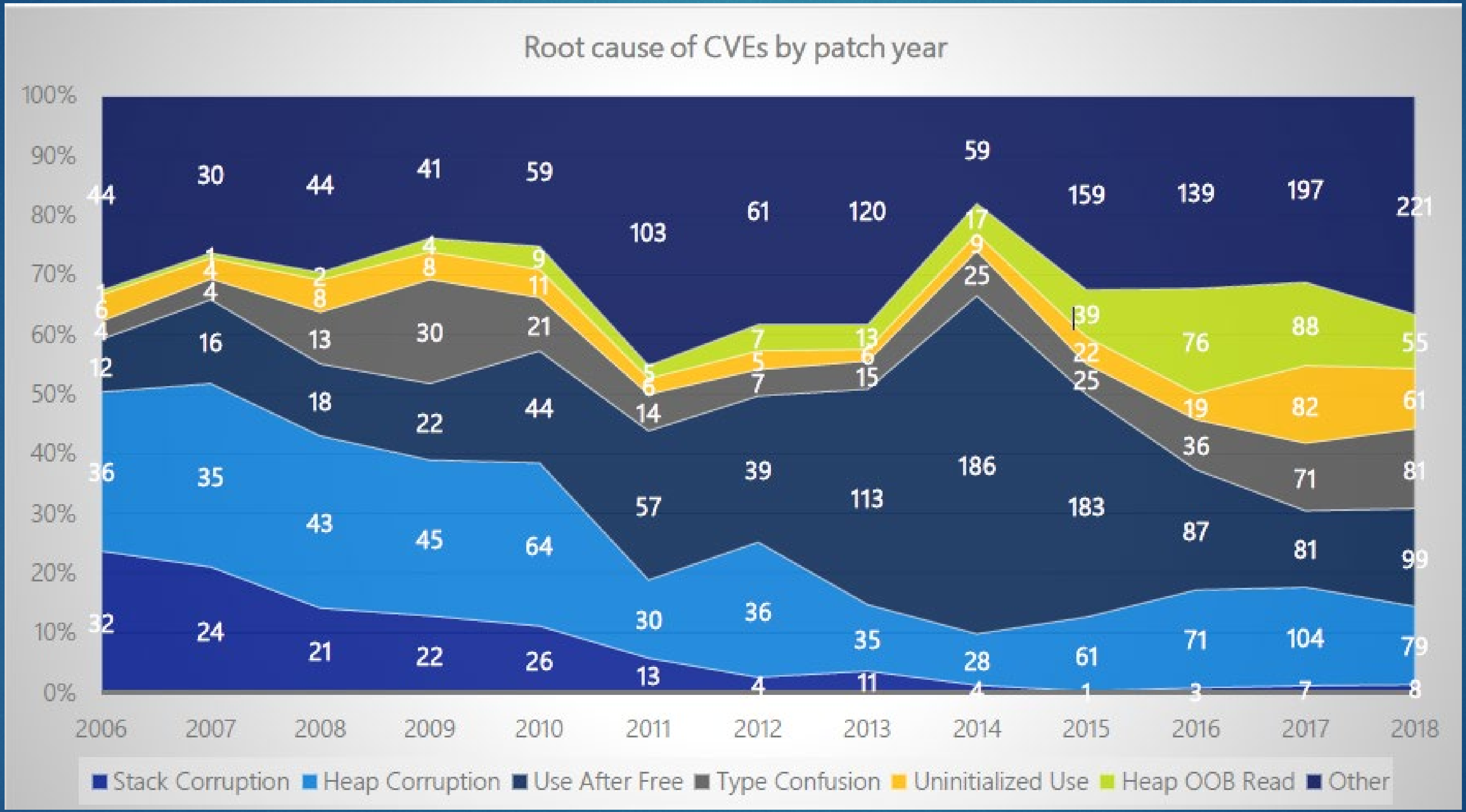
- 89% of remotely exploitable vulnerabilities were memory safety bugs in 2022

- 78% of confirmed exploited vulnerabilities were memory safety bugs over the last several years



https://security.googleblog.com/2022/12/memory-safe-languages-in-android-13.html

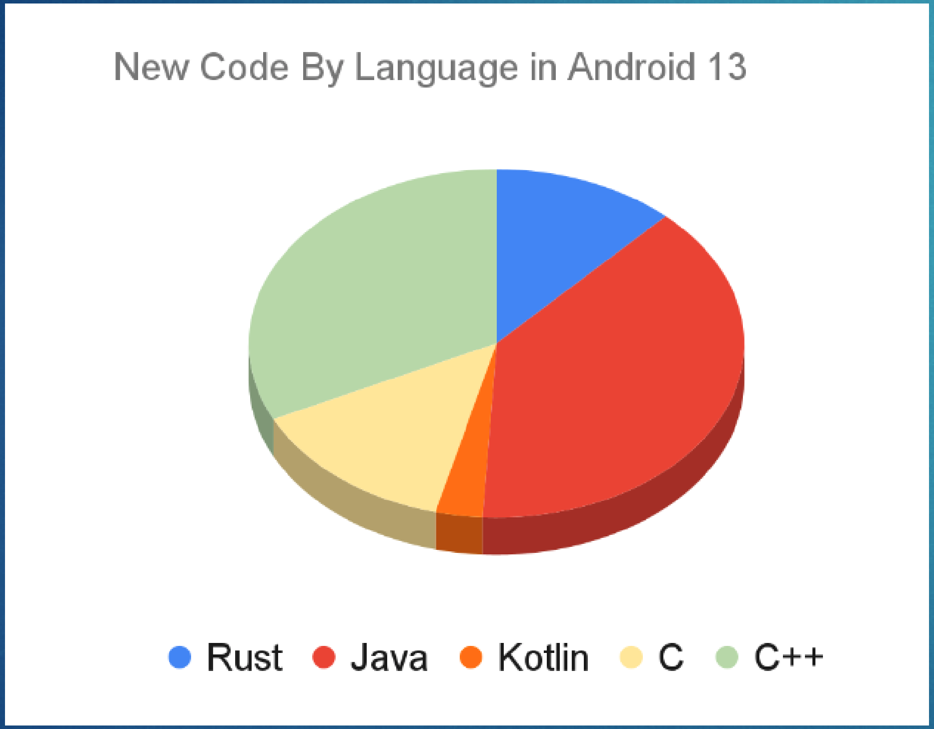
Strategic Security



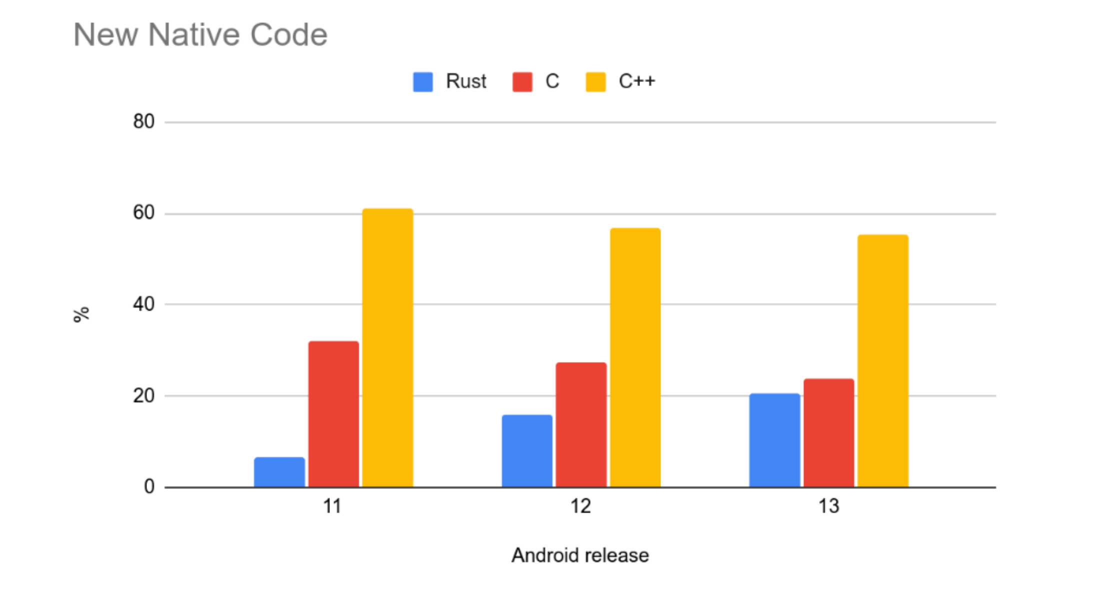
https://github.com/microsoft/MSRC-Security-Research/blob/master/presentations/2019_02_BlueHatIL/2019_01%20-%20BlueHatIL%20-%20Trends%2C%20challenge%2C%20and%20shifts%20in%20software%20vulnerability%20mitigation.pdf

Google Android Languages

Android 13 Languages



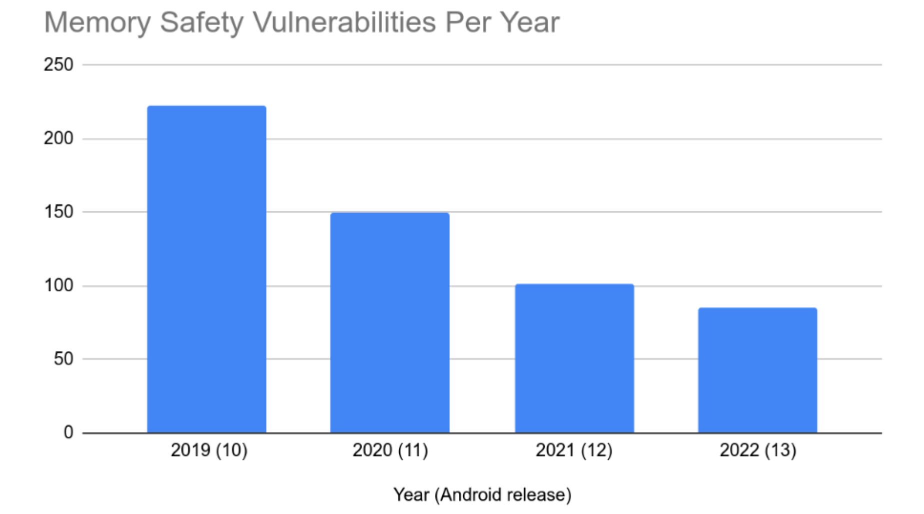
Android Native Code Languages



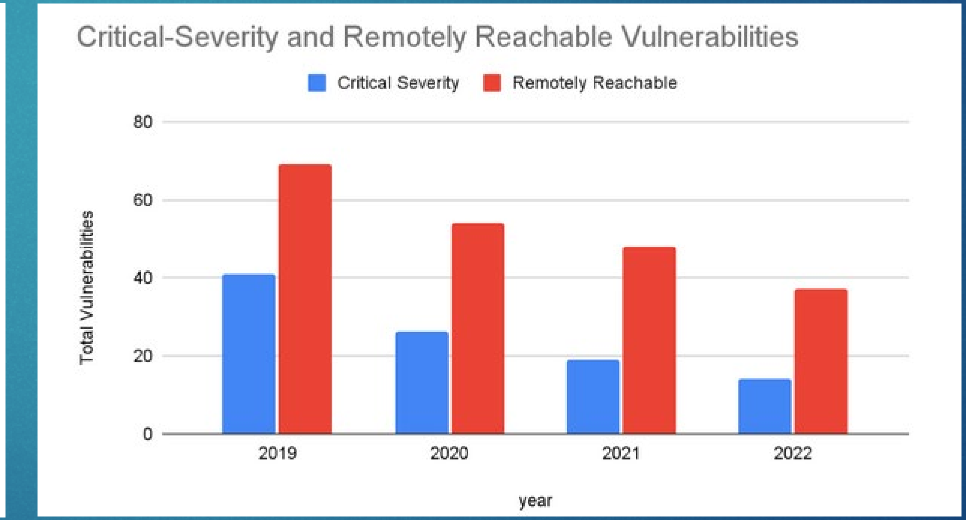
https://security.googleblog.com/2022/12/memory-safe-languages-in-android-13.html

Google Android Vulnerabilities

Memory Safety

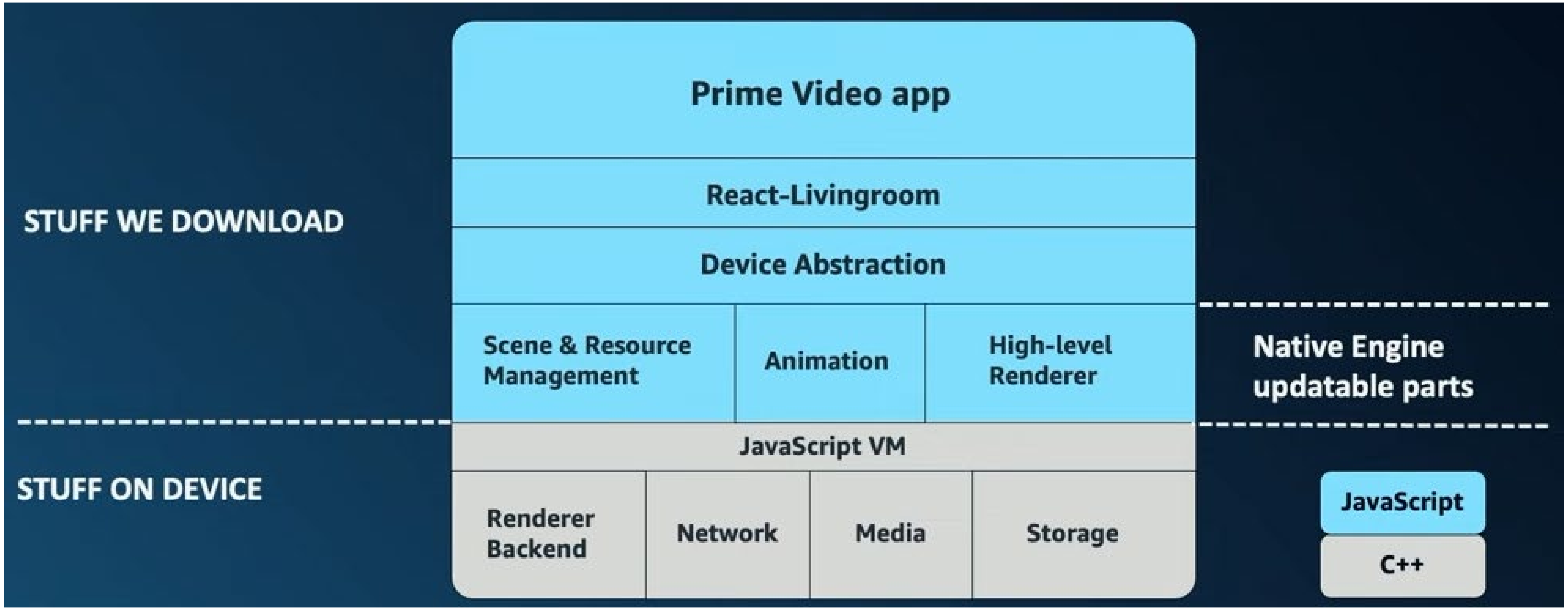


Critical Severity



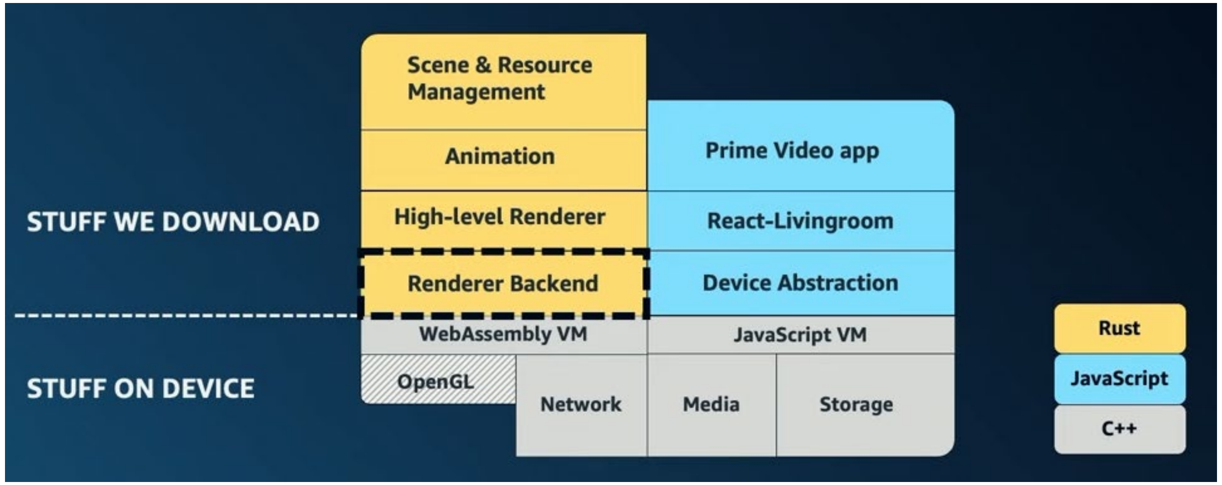
https://security.googleblog.com/2022/12/memory-safe-languages-in-android-13.html

Amazon Prime Video Languages



https://youtu.be/erdHTxghyM0

Amazon Prime Video Results



https://youtu.be/erdHTxghyM0

“Because we use Rust, we have a crash rate that is ten times smaller for the WebAssembly systems compared to the C++ systems. […]

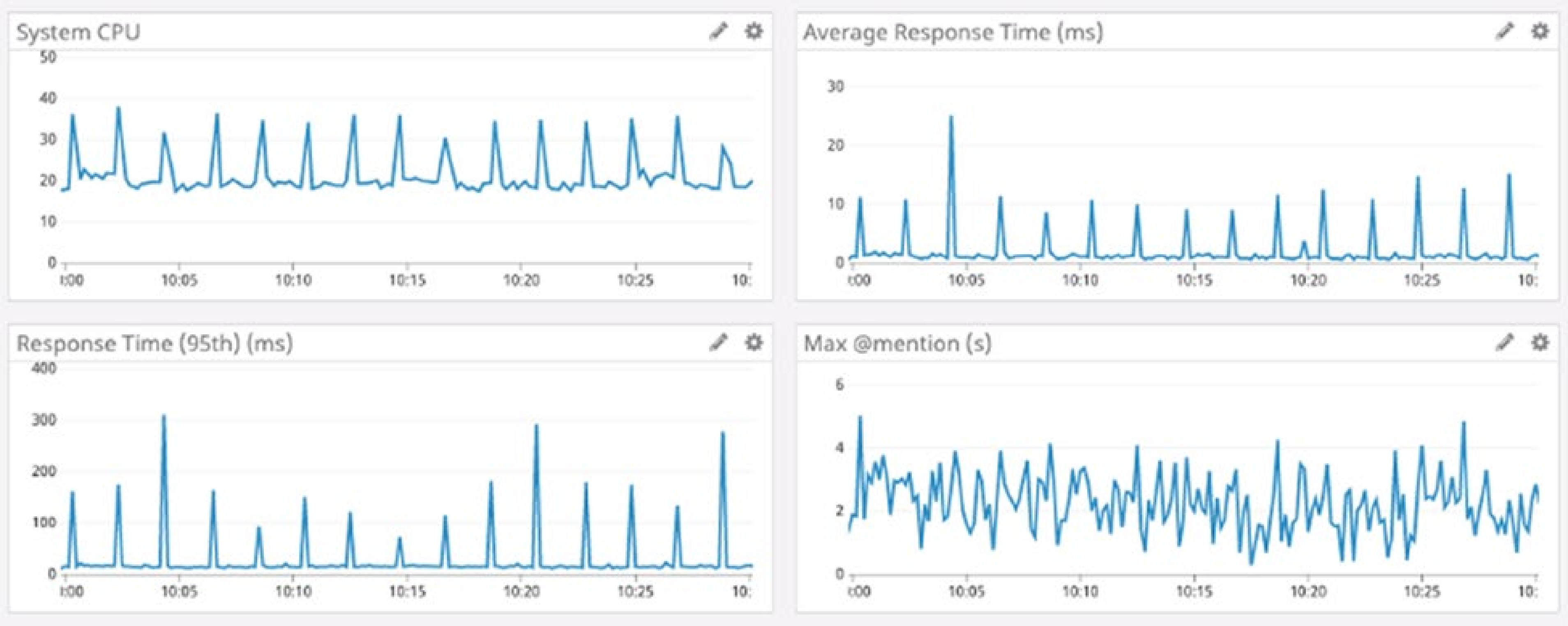
On some devices, actually, the crash rate is most days zero with WebAssembly [Rust].”

Alexandru Ene

Principal Engineer, Amazon Prime Video

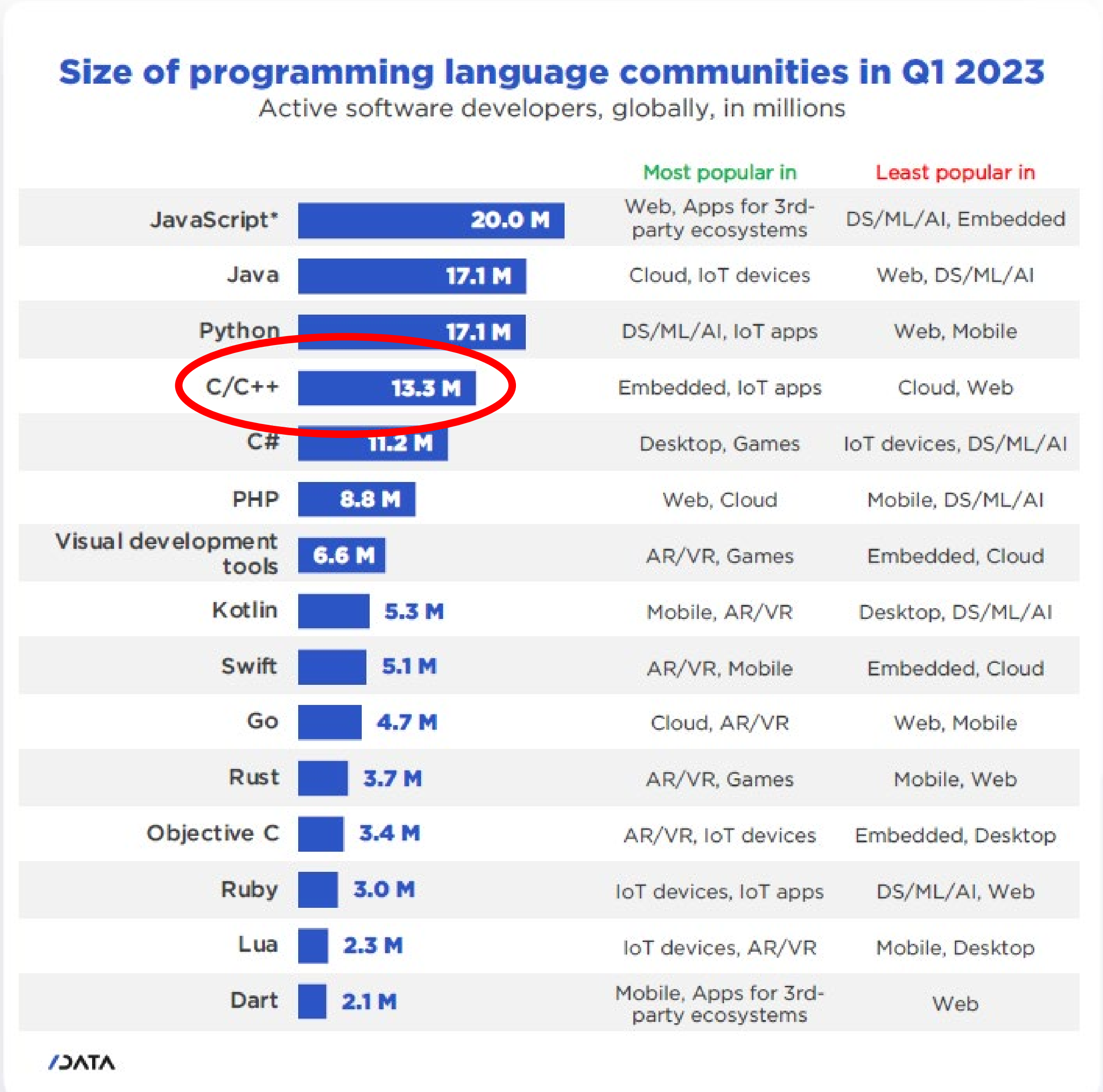
Garbage Collectors

Discord



https://discord.com/blog/why-discord-is-switching-from-go-to-rust

Programming Languages



Size of programming language communities in Q1 2023

Active software developers, globally, in millions

		Most popular in	Least popular in
JavaScript*	20.0 M	Web, Apps for 3rd-party ecosystems	DS/ML/AI, Embedded
Java	17.1 M	Cloud, IoT devices	Web, DS/ML/AI
Python	17.1 M	DS/ML/AI, IoT apps	Web, Mobile
C/C++	13.3 M	Embedded, IoT apps	Cloud, Web
C#	11.2 M	Desktop, Games	IoT devices, DS/ML/AI
PHP	8.8 M	Web, Cloud	Mobile, DS/ML/AI
Visual development tools	6.6 M	AR/VR, Games	Embedded, Cloud
Kotlin	5.3 M	Mobile, AR/VR	Desktop, DS/ML/AI
Swift	5.1 M	AR/VR, Mobile	Embedded, Cloud
Go	4.7 M	Cloud, AR/VR	Web, Mobile
Rust	3.7 M	AR/VR, Games	Mobile, Web
Objective C	3.4 M	AR/VR, IoT devices	Embedded, Desktop
Ruby	3.0 M	IoT devices, IoT apps	DS/ML/AI, Web
Lua	2.3 M	IoT devices, AR/VR	Mobile, Desktop
Dart	2.1 M	Mobile, Apps for 3rd-party ecosystems	Web

/DATA

https://www.developernation.net/resources/reports/state-of-the-developer-nation-24th-edition-q1-2023

Rust

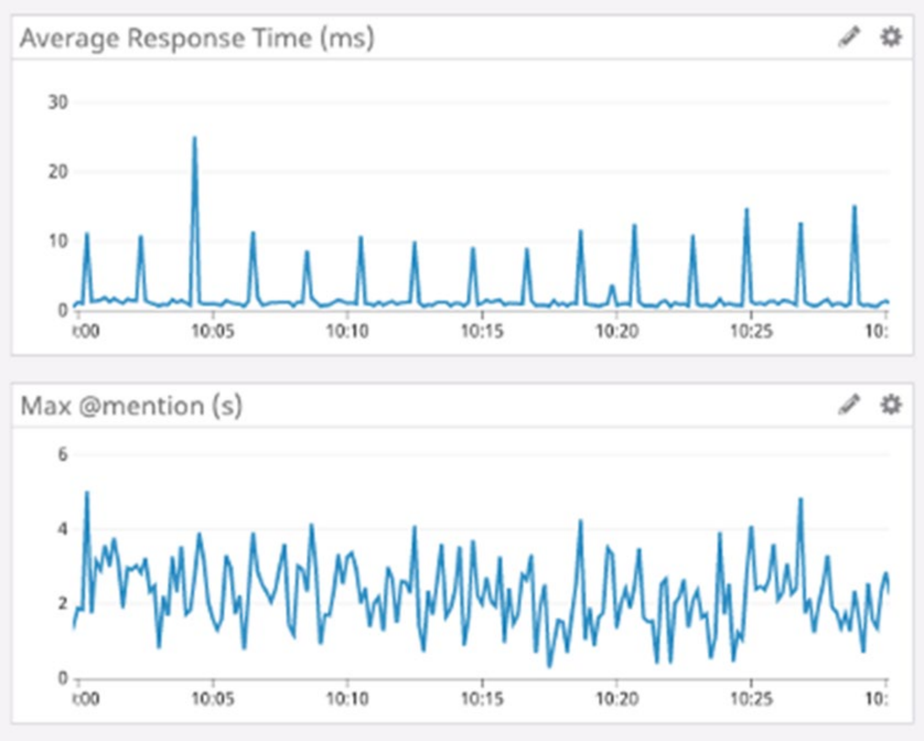
- Ownership
- Borrow Checker

Borrow Checker

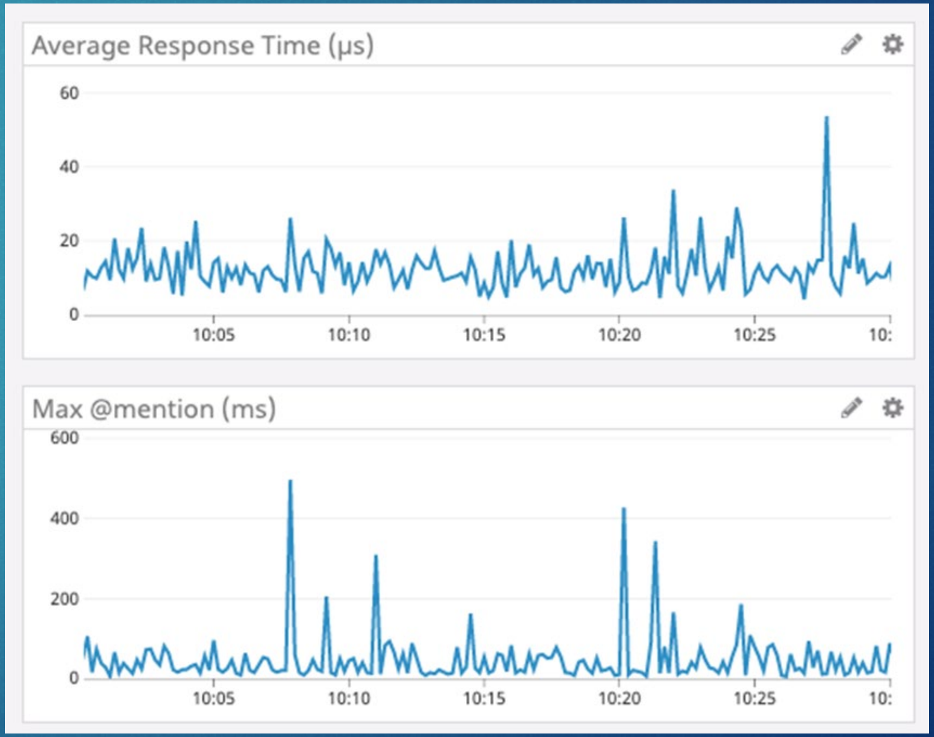
```
fn admire(gift: &Gift) {
    println!(“wow, this {} looks nice!”, gift);
}
let gift = Gift::new();
admire(&gift);
```

Discord

Go

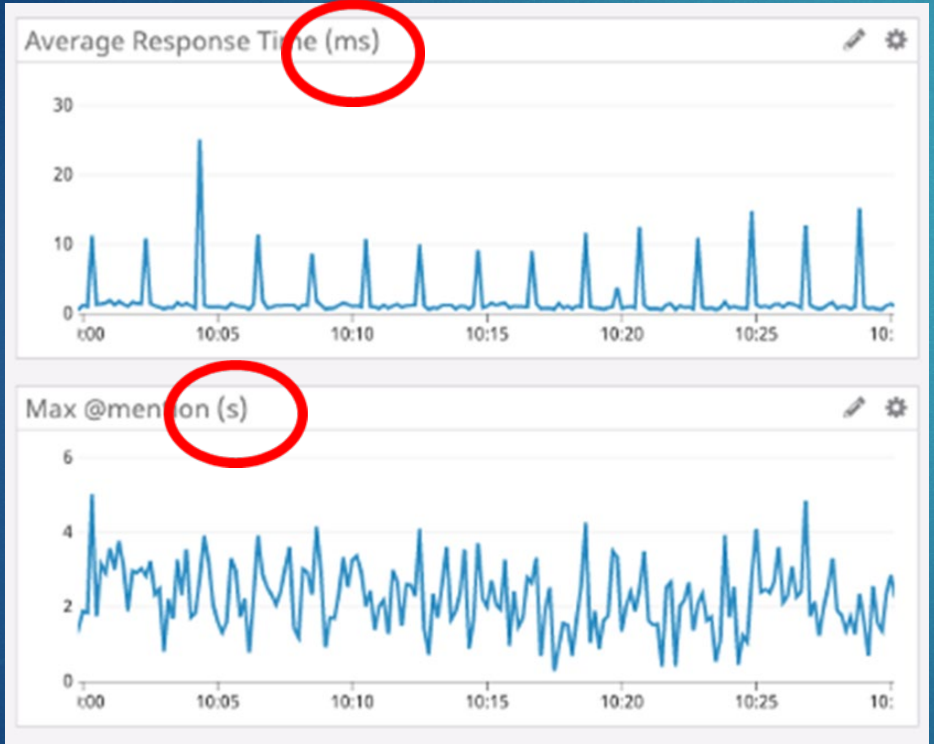


Rust

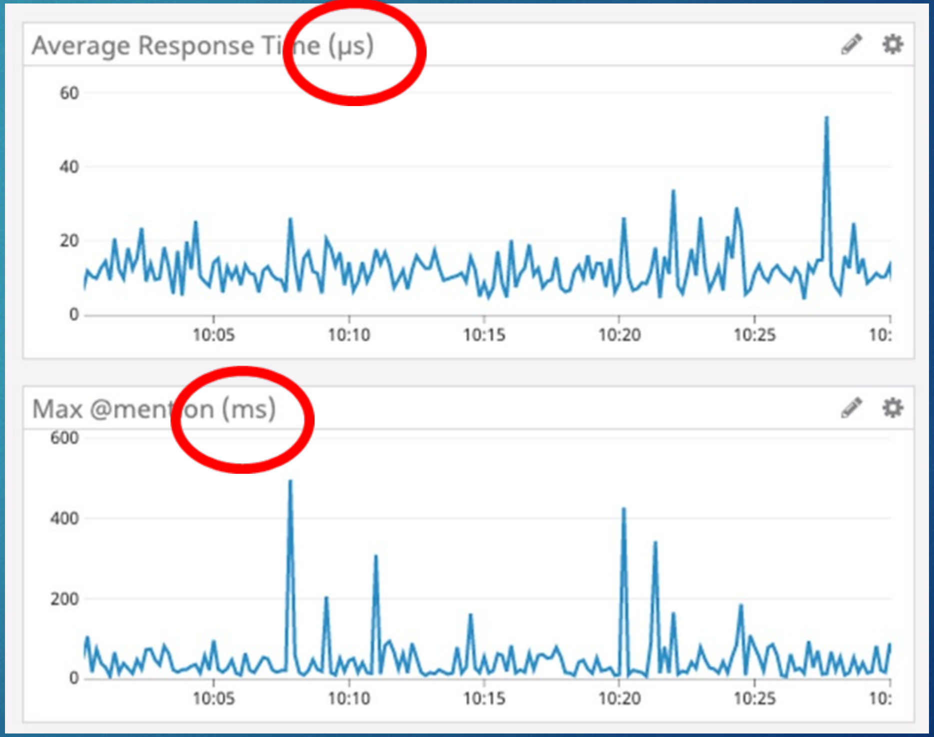


Discord

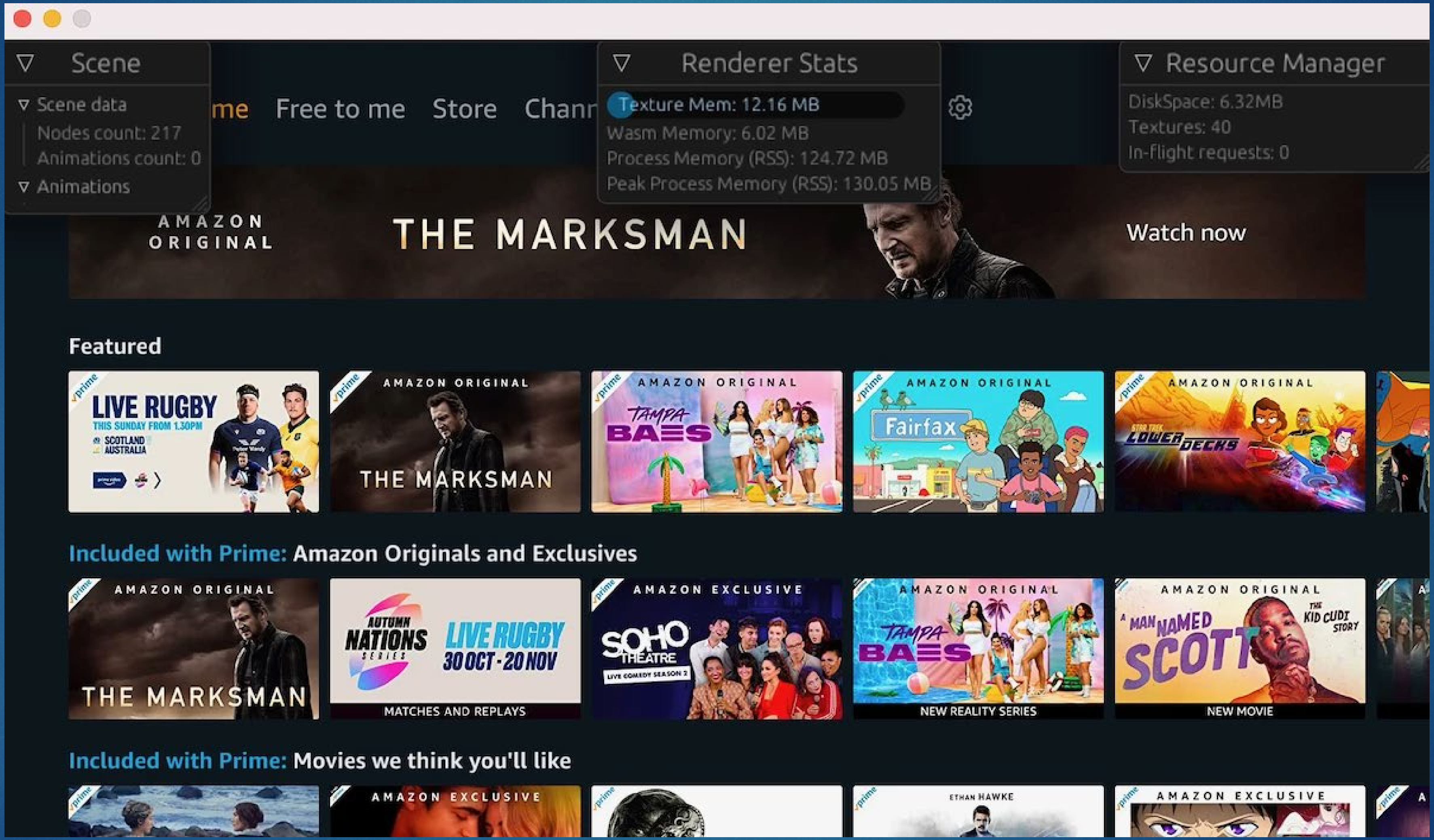
Go



Rust



Amazon Prime Video



```
▽ Scene
▽ Scene data
  Nodes count: 217
  Animations count: 0
▽ Animations
```

```
▽ Renderer Stats
Texture Mem: 12.16 MB
Wasm Memory: 6.02 MB
Process Memory (RSS): 124.72 MB
Peak Process Memory (RSS): 130.05 MB
```

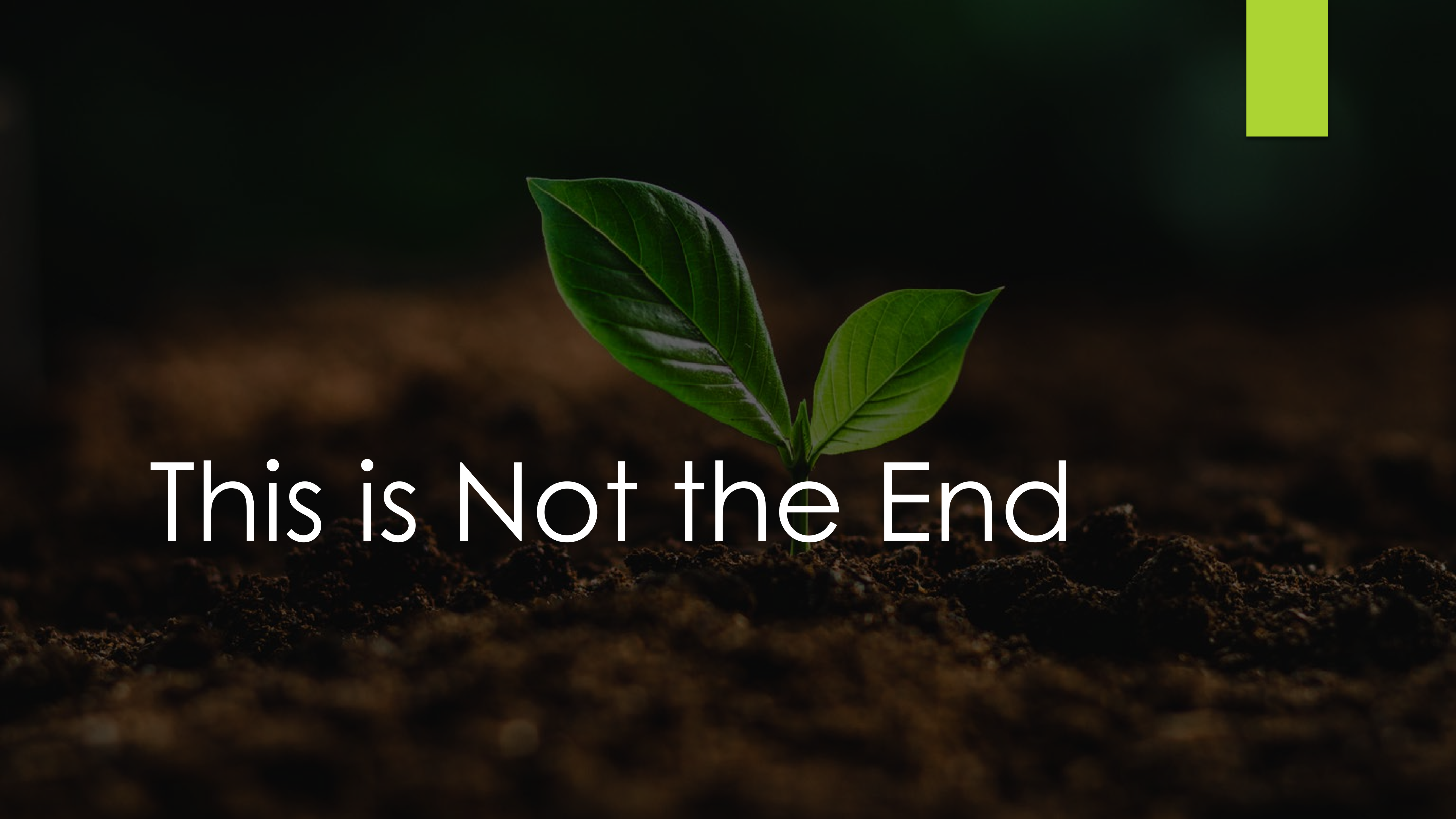
```
▽ Resource Manager
DiskSpace: 6.32MB
Textures: 40
In-flight requests: 0
```

https://www.amazon.science/blog/how-prime-video-updates-its-app-for-more-than-8-000-device-types

Programming Language Tradeoffs

	Language Type		
	Systems (C/C++)	Garbage Collector (Java, Python, etc)	Compile-time Verification (Rust)
Flexibility	🟩	🟥	🟩
Usability	🟥	🟩	🟥
Cost to Run	🟩	🟥	🟩
Performance	🟩	🟥	🟩
Memory Safe	🟥	🟩	🟩

Open Source Entropy



This is Not the End

Acknowledgements

Graydon Hoare

- Creator of Rust
- Safe programming researcher
- Community advocate

Josh Aas

- Leader of the memory safety movement
- Founder and leader of the Internet Security Research Group (ISRG)

Thank you!

SHANE MILLER
HTTPS://SHANE-ONE.COM/
TWITTER: @SKIPPERSWIF
LINKEDIN: SHANEMILLERITMANAGER
MAIL: SHANE@SHANE-ONE.COM