
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

BOA201

Rust is interesting, but does it really make sense for me?

Tim McNamara (he/him)

Senior Software Development Engineer
AWS

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The software industry faces problems

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Software development today

- Unreliable
- Insecure
- Wasteful

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“Rust solves pain points present in many other languages, providing a solid step forward with a limited number of downsides.”

Jake Goulding

Integer 32

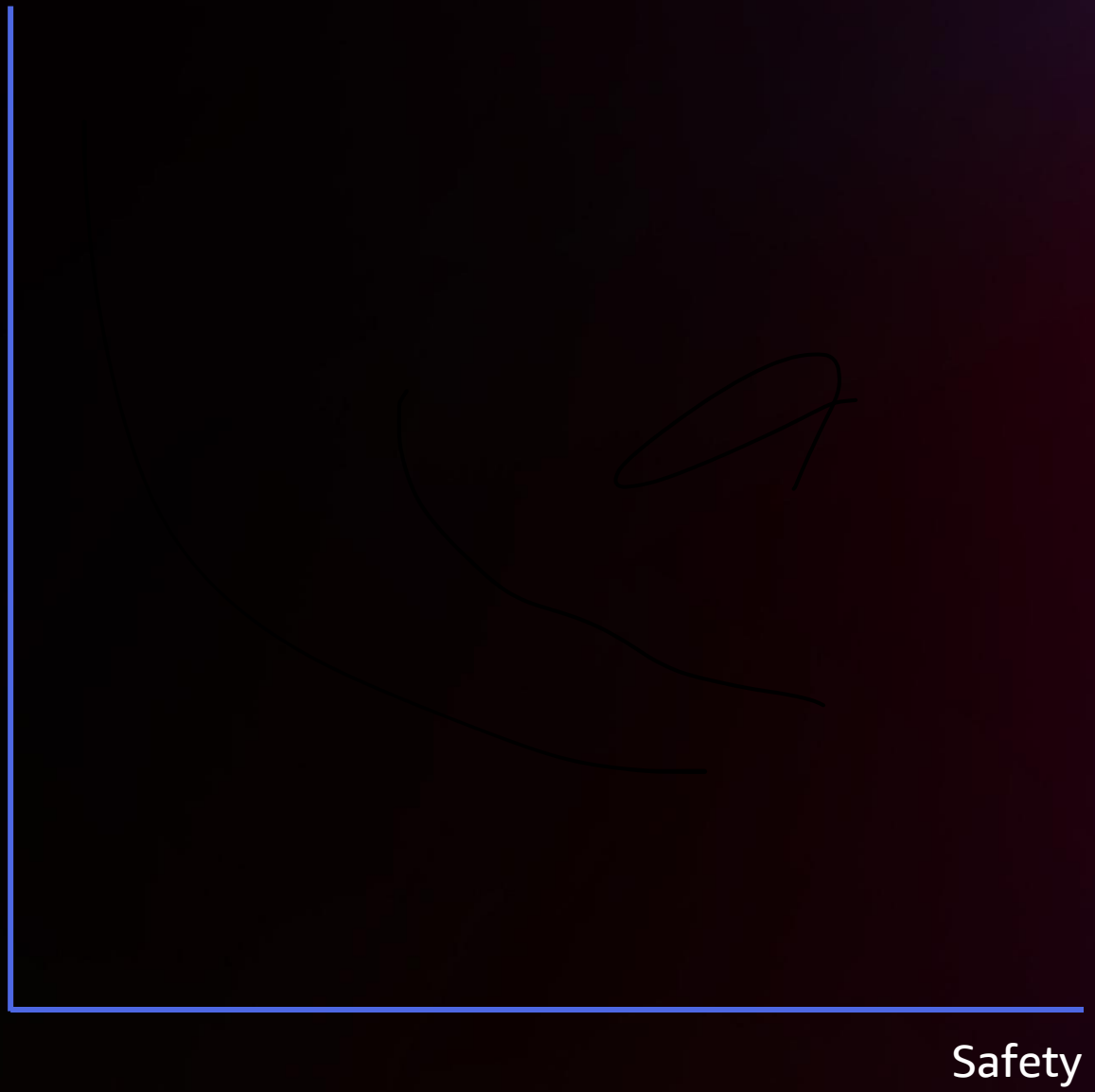
https://stackoverflow.blog/2020/06/05/why-the-developers-who-use-rust-love-it-so-much/

aws

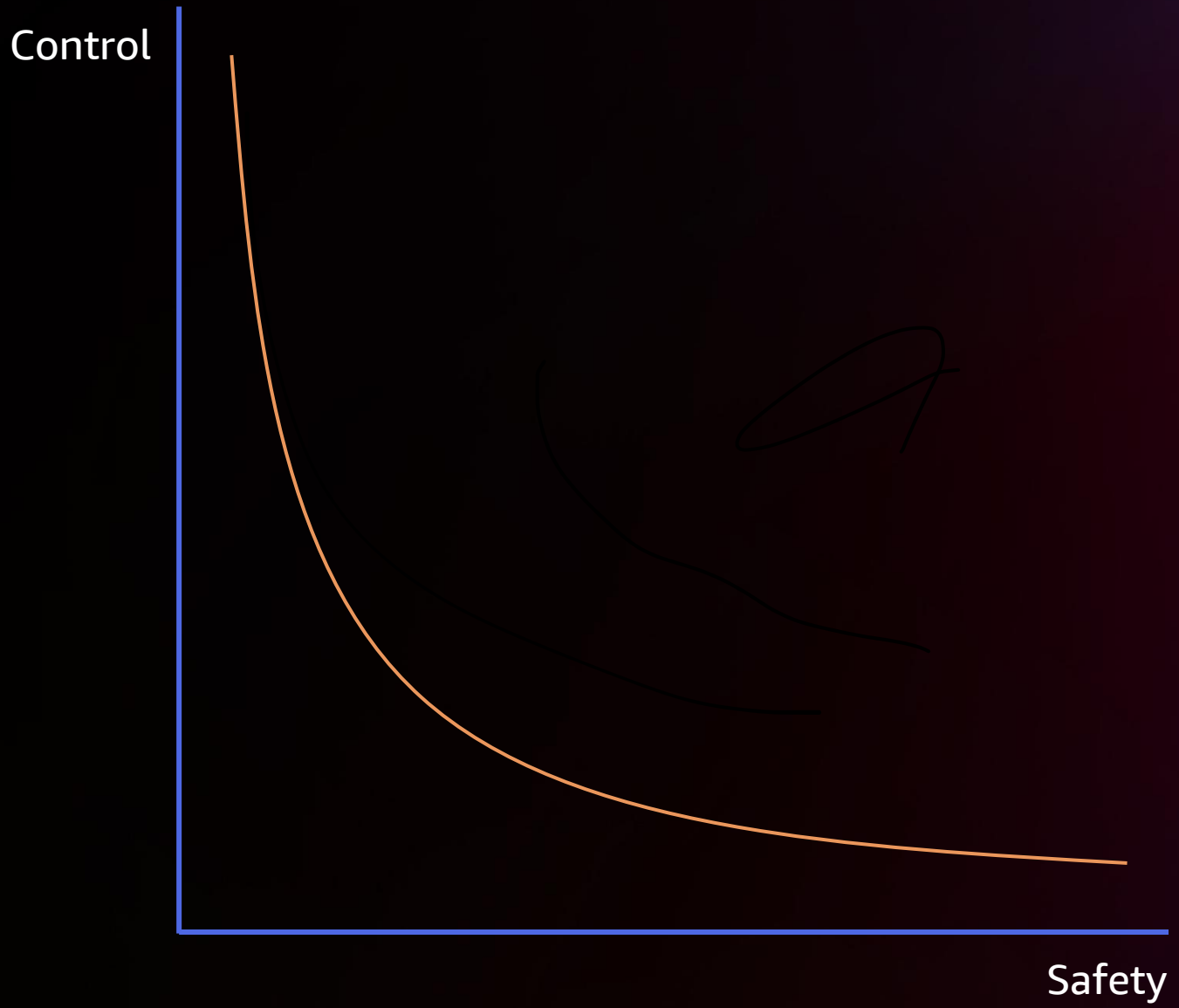
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is Rust?

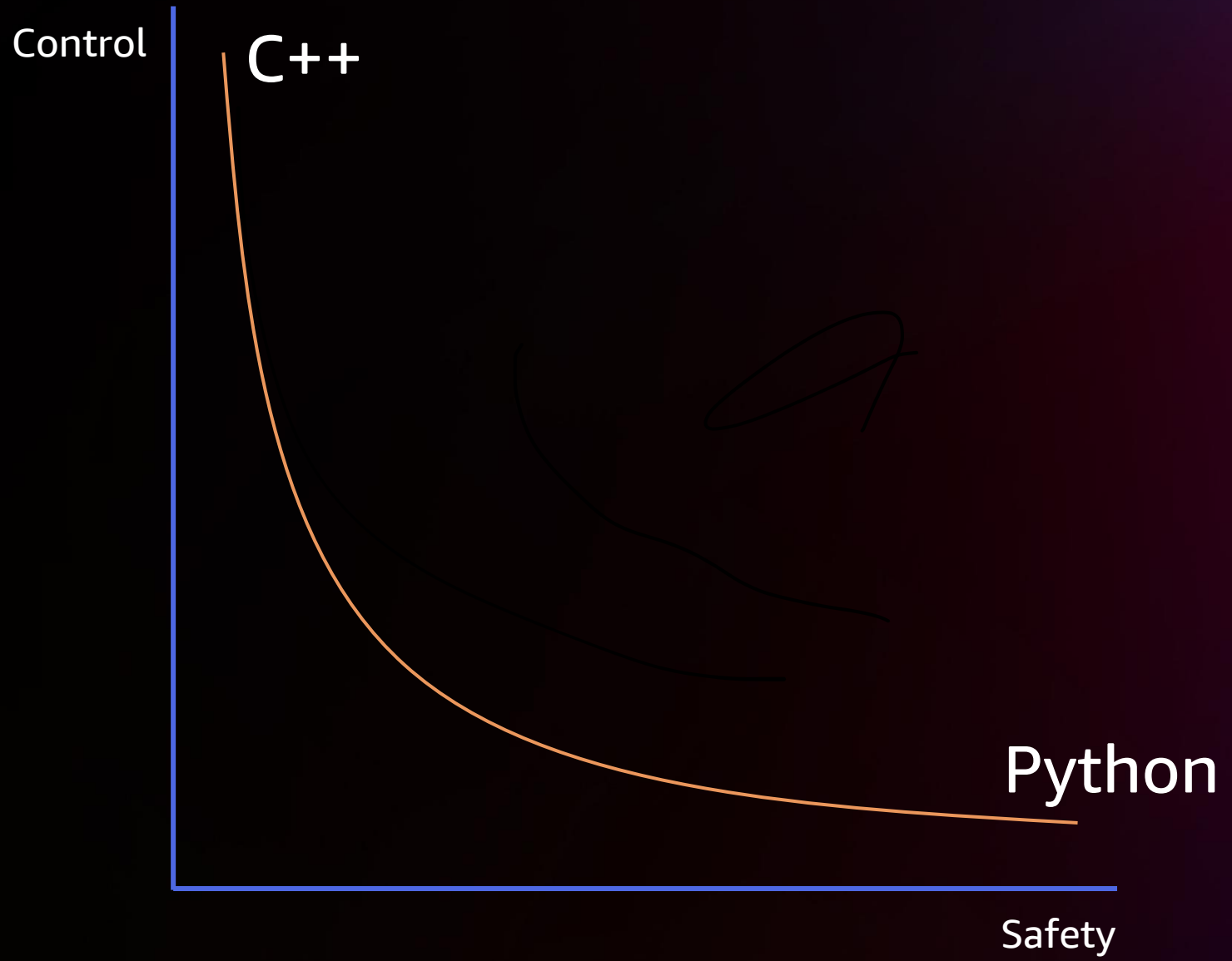
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



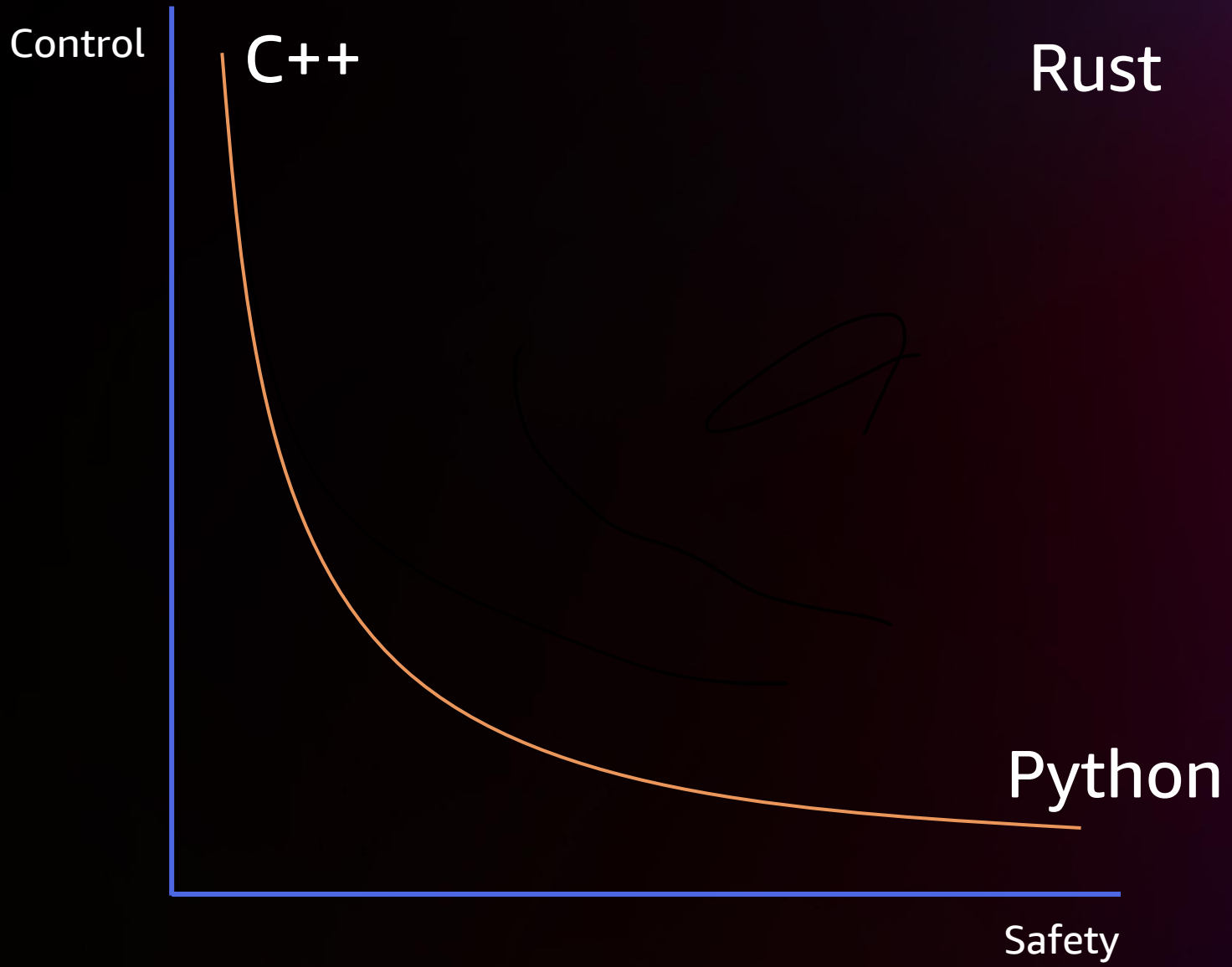
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What does Rust look like?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

```rust
fn sum_of_squares(input: &[i32]) -> i32 {
    input.iter()
        .map(|&i| i * i)
        .sum()
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

```rust
use rayon::prelude::*;

fn sum_of_squares(input: &[i32]) -> i32 {
    input.par_iter()
         .map(|&i| i * i)
         .sum()
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

```rust
use std::collections::HashMap;

fn count_words(text: &str) -> HashMap<&str, u32> {
    let mut counts = HashMap::new();
    for word in text.split_whitespace() {
        let word = word.trim_matches(|c| {
            !char::is_alphabetic(c)
        });
        let count = counts.entry(word).or_default();
        *count += 1;
    }
    counts
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

```rust
fn main() {
    for i in 1..=100 {
        match (i % 3, i % 5) {
            (0, 0) => println!("fizzbuzz"),
            (0, _) => println!("fizz"),
            (_, 0) => println!("buzz"),
            (_, _) => println!("{i}"),
        }
    }
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Sourced from https://rosettacode.org/wiki/FizzBuzz#Rust

Programs that you can’t write in Rust

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

ASSIGNMENT USED WITHIN CONDITIONAL IN C

```c
#include <stdio.h>

int main() {
        int a = 42;

        if (a = 24)
                printf("?");

        return 0;
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

RUNTIME CRASH DUE TO ITERATOR INVALIDATION IN JAVA

```java
import java.util.ArrayList;

class Mathematics {
	public static void main(String[] args) {
		ArrayList<Integer> numbers = new ArrayList<Integer>();
		numbers.add(1);
		numbers.add(2);
		numbers.add(3);

		System.out.println(numbers);
		for (Integer n: numbers) {
			numbers.remove(1);
		}
		System.out.println(numbers);

	}
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

MISLEADING SEMANTICS DUE TO TYPE SYSTEM LIMITATIONS IN JAVASCRIPT

```
let x = new Set();

x.add({ "id": 123 });
x.add({ "id": 123 });

console.log(x); // prints 2
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

RACE CONDITIONS IN GO

```
package main

import "fmt"

func main() {
        done := make(chan bool)
        m := make(map[string]string)
        m["name"] = "world”

        go func() {
                m["name"] = "tim"
                done <- true
        }()
        go func() {
                m["name"] = ”tom"
                done <- true
        }()

        fmt.Println("Hello,", m["name"])
        <-done
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why is Rust interesting?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why is Rust interesting?

Answer 1

It eliminates memory safety errors without any runtime cost

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

"

~70% of the vulnerabilities Microsoft assigns a CVE each year continue to be memory safety issues

A proactive approach to more secure code

Microsoft Security Response Center

https://msrc-blog.microsoft.com/2019/07/16/a-proactive-approach-to-more-secure-code/

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“

Around 70% of our high severity security bugs are memory unsafety problems (that is, mistakes with C/C++ pointers).

Chromium Security: Memory Safety

Chromium

https://www.chromium.org/Home/chromium-security/memory-safety/

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“

Use-after-free (UAF), integer overflows, and out of bounds (OOB) reads/writes comprise **90%** of vulnerabilities with OOB being the most common.



Queue the Hardening Enhancements

Jeff Vander Stoep & Chong Zhang, Android Developers Blog

https://android-developers.googleblog.com/2019/05/queue-hardening-enhancements.html

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

> If we'd had a time machine and could have written this component in Rust from the start, 51 (**73.9%**) of these bugs would not have been possible.

Implications of Rewriting a Browser Component in Rust

Diane Hosfeld, Mozilla Hacks Blog

https://hacks.mozilla.org/2019/02/rewriting-a-browser-component-in-rust/

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

	% Memory Unsafety Bugs
iOS 14	60.4%
iOS 13	59.4%
iOS 12	66.3%
iOS 11	60.5%



2021 in Memory Unsafety – Apple's Operating Systems

Paul Kehrer, Fish in a Barrel

https://langui.sh/2021/12/13/apple-memory-safety/

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

> If we'd had a time machine and could have written this component in Rust from the start, 51 (**73.9%**) of these bugs would not have been possible.



Memory Unsafety in Apple's Operating Systems

Diane Hosfeld, Mozilla Hacks Blog

https://hacks.mozilla.org/2019/02/rewriting-a-browser-component-in-rust/

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“Memory corruption issues are the root-cause of 68% of listed CVEs.



0day “In the Wild”

Ben Hawkes, Project Zero

https://googleprojectzero.blogspot.com/p/0day.html

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why is Rust interesting?

Answer 2

It empowers your teams to build better software with fewer resources

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The pitch

- **Blazingly fast**
  Rust provides zero-cost abstractions and a surprisingly smart compiler.
- **No “billion dollar mistake”**
  It’s impossible to de-reference a null pointer. They don’t exist in Rust.
- **Use preferred programming techniques**
  Developers love programming in Rust. It’s been voted the most loved programming language seven years’ running.
- **Never forget to handle an error**
  Your program won’t compile unless all error cases are accounted for.

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Evaluating Rust

Rust's benefits

- Customer experience – lower latency
- Simpler operations – smaller containers, less memory, fewer crashes
- Memory safety – increase security and stability
- No performance costs – no garbage collector
- Widely applicable – from large applications to small utilities

Rust's costs

- Can feel pedantic and fussy – especially if you are used to "quick and dirty" scripting
- Requires different mental model – short term productivity loss
- Learning curve is steep – 55% of people who have left the language community found Rust too hard

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Some external experiences

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“[With Rust], we saw a 75% reduction in CPU usage and a 95% reduction in memory usage in production.”



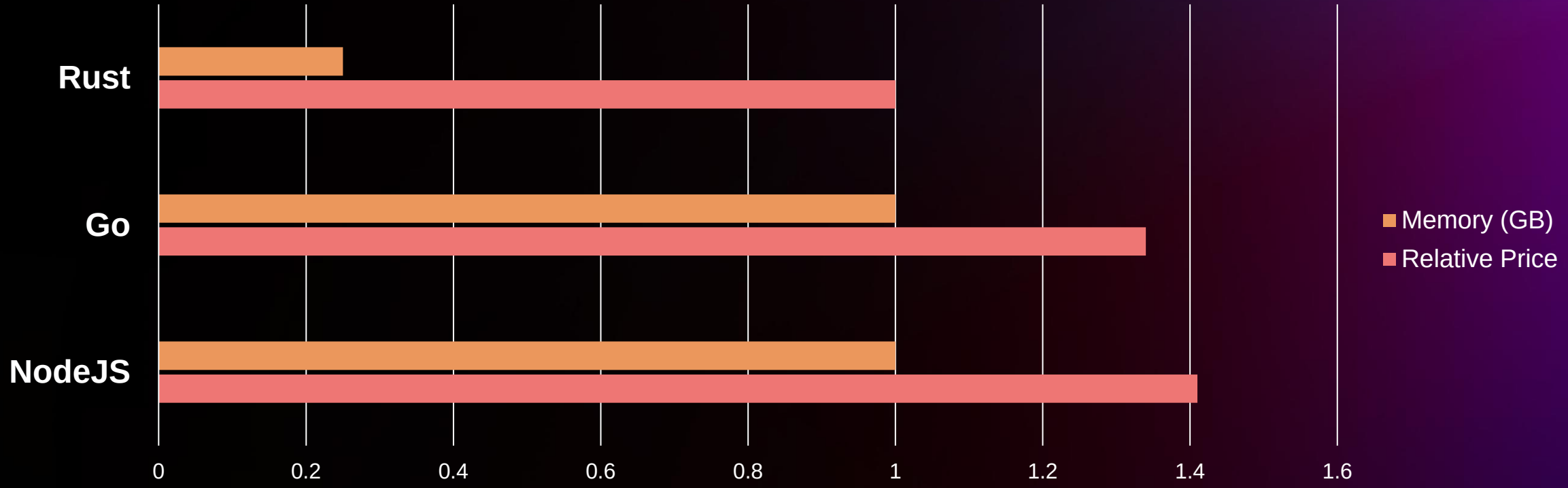
Alan Ning

Tenable

https://medium.com/tenable-techblog/optimizing-700-cpus-away-with-rust-dc7a000dbdb2

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Relative price of equivalent workload on AWS Lambda running 2.5B times

Daniele Frasca
AWS Community Builder
ProSiebenSat.1

https://twitter.com/dfrasca80/status/1578411658975723522

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Memory usage	Agent	Server
Before	>100 MB	5 GB
After	8 MB	50 MB

Memory usage before and after porting two applications to Rust: a daemon living on the client and metrics collection server

Case Study: How Rust is Tilde's Competitive Advantage

Rust Foundation

https://www.rust-lang.org/static/pdfs/Rust-Tilde-Whitepaper.pdf

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The AWS experience

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



https://aws.amazon.com/about-aws/whats-new/2022/08/amazon-cloudfront-supports-http-3-quic/

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Firecracker

https://www.amazon.science/blog/how-awss-firecracker-virtual-machines-work

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

prime video

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

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



https://www.amazon.science/publications/using-lightweight-formal-methods-to-validate-a-key-value-storage-node-in-amazon-s3

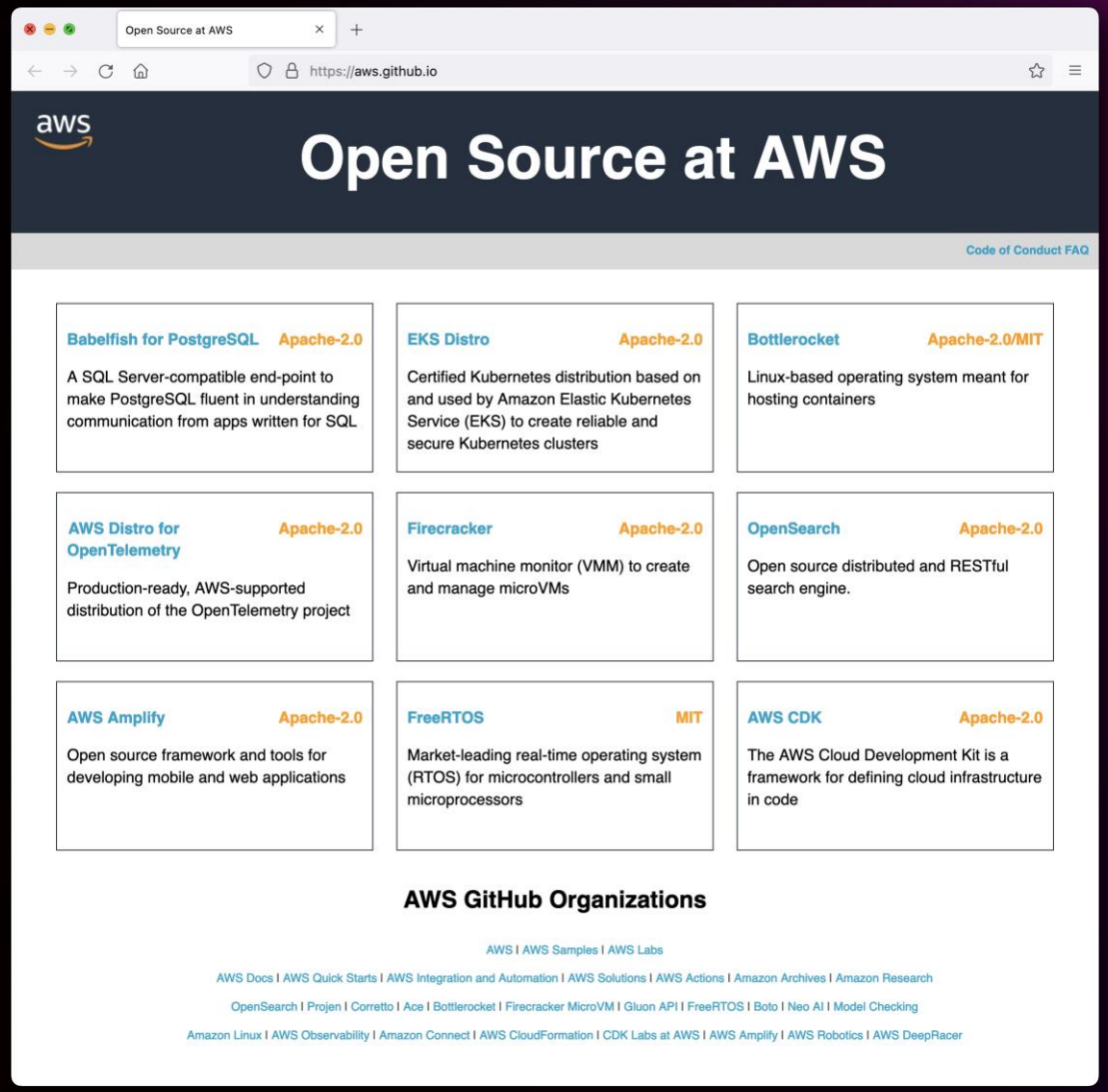
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Where do I learn more?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS GitHub Page

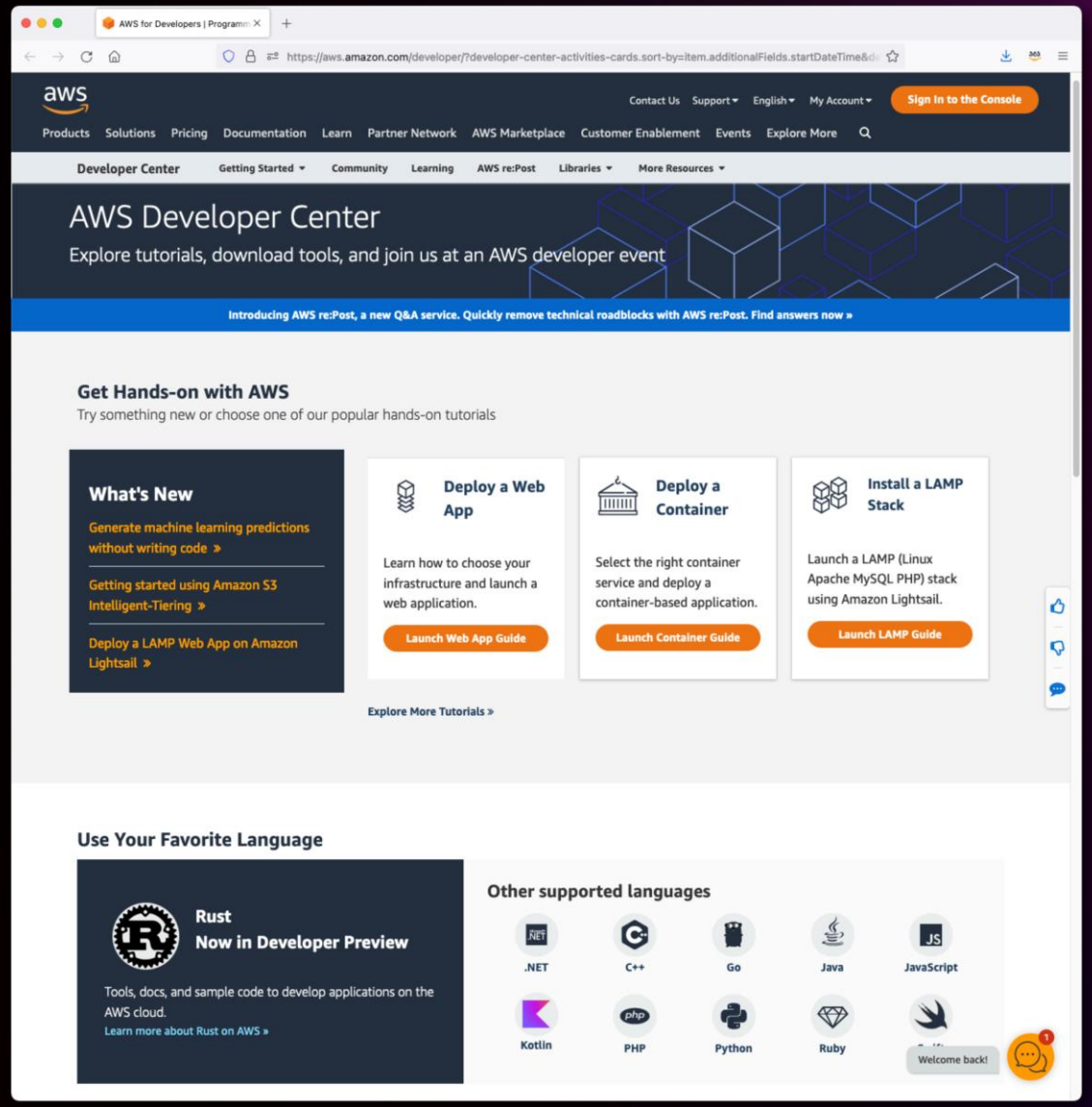
https://aws.github.io/



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Developer Center

https://dev.aws/



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“… on the whole, I don’t hate it.”

Linus Torvalds

Linux Foundation

https://lkml.org/lkml/2021/4/14/1099

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Tim McNamara
@timClicks



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Small selection of Rust usage within Amazon and AWS

	Systems programming	Web services	wasm	Serverless	Libraries	CLIs
Applications and services	AWS Nitro Enclaves	Amazon S3 ShardStore Amazon CloudFront	Prime Video	AWS Lambda AWS Fargate		AWS CloudFormation Guard 2.0 tuftool
Developer tools		AWS SDK for Rust smithy-rs		AWS Lambda Rust Runtime	s2n-quic Prost (protobufs) Tonic (gRPC)	
Major open source contributions	Bottlerocket OS Firecracker	Tokio Rust compiler			Kani Rust Verifier	

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.