



# Rust and what's this thing for?



Abc Xyz  
@dura\_lex

1. Foreword
2. What is Rust?
3. (Un)safe
4. Summary



# Foreword

---

- Since 1.0.0
- Scope (by time)
  - Bindings (FFI — foreign function interface)
  - Analyzers
  - CLI (TUI) tools for PC and IoT
  - GUI for fun
  - Libraries
  - RE
- Nim, Crystal, Zig, Pony





# What is Rust?

---

«**Rust** is a multi-paradigm systems programming language focused on safety, especially safe concurrency».

— Wikipedia

«*Rust is a systems programming language that runs blazingly fast, prevents nearly all segfaults, and guarantees thread safety*».

— [www.rust-lang.org](http://www.rust-lang.org) (2015)



*«Empowering everyone to build reliable and efficient software».*

— [www.rust-lang.org](http://www.rust-lang.org)

The background of the slide features a complex, light gray network pattern. It consists of numerous small dots (nodes) connected by thin, intersecting lines (edges), creating a web-like or molecular structure that covers the entire area.

# What is Rust?

---

## Quick facts about Rust

- Started by Mozilla (sponsorship & support) employee Graydon Hoare
- First announced by Mozilla in 2010
- Community driven development
- 88,281 commits on GitHub
- First stable release: 1.0 in May 2015
- Latest stable release: 1.32

The background of the slide features a complex, light gray network pattern. It consists of numerous small dots (nodes) connected by thin, intersecting lines (edges), creating a web-like or molecular structure that fills the entire frame. The pattern is more dense in some areas and more sparse in others, with lines forming various polygonal shapes.

# What is Rust?

---

Why Rust?




- Performance
  - Fast, memory-efficient
  - No runtime or garbage collector
  - Zero-cost abstractions
- Reliability
  - Rich type system
  - Ownership model
- Productivity
  - Documentation
  - Friendly compiler
  - Top-notch tooling



(Un)safe

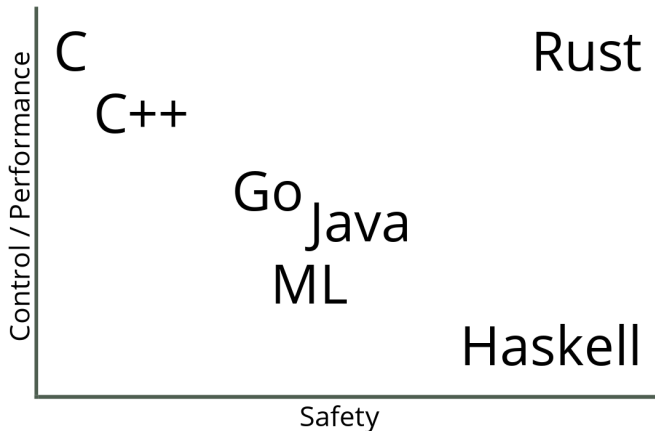
---

The background of the slide features a complex, light gray network pattern. It consists of numerous small dots (nodes) connected by thin, intersecting lines, creating a web-like structure that covers the entire area.

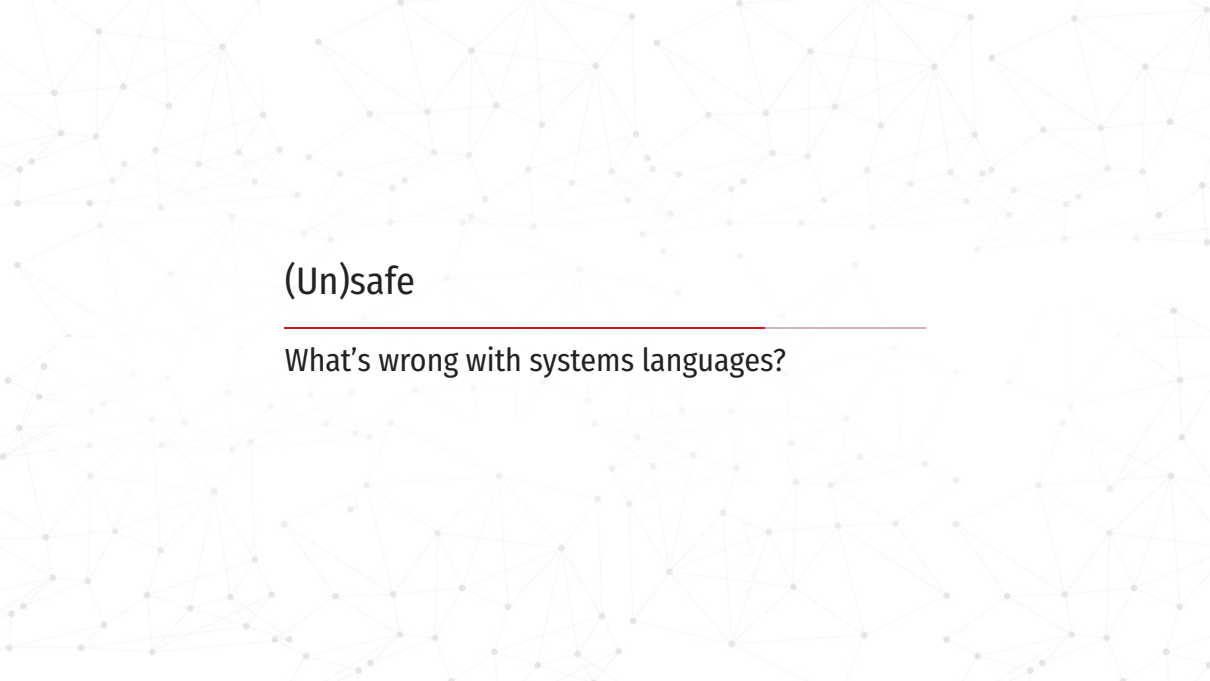
(Un)safe

---

Control vs Safety





The background of the slide features a complex, light gray network pattern. It consists of numerous small dots (nodes) connected by thin, intersecting lines (edges), creating a web-like or mesh structure that covers the entire area.

(Un)safe

---

What's wrong with systems languages?

# What's wrong with systems languages?

- It's difficult to write secure code
- It's very difficult to write multithreaded code

Rust?

The background of the slide features a complex, light gray network pattern. It consists of numerous small, dark gray circular nodes connected by thin, light gray lines, creating a web-like structure that fills the entire frame. The nodes are distributed unevenly, with some clusters and some sparse areas.

(Un)safe

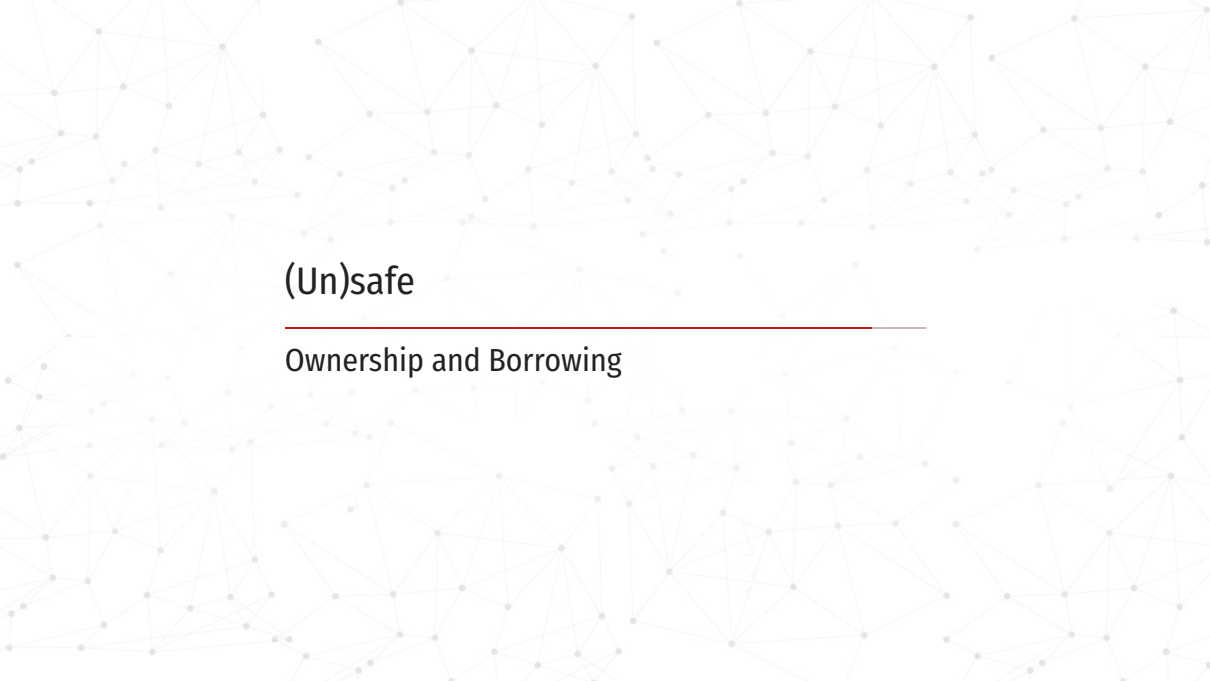
---

Problems

## Memory corruption

- Using uninitialized memory
- Using non-owned memory (null pointer, dangling pointer dereference, out of bounds error)
- Using memory beyond the memory that was allocated (buffer overflow)
- Faulty heap memory management (memory leaks, freeing non-heap or un-allocated memory)



The background of the slide features a complex, light gray network pattern. It consists of numerous small dots (nodes) connected by thin, intersecting lines, creating a web-like or mesh structure that covers the entire area.

(Un)safe

---

Ownership and Borrowing



# *Ownership and Borrowing*


**Nicholas Matsakis**

# Ownership

*n.* The act, state, or right of possessing something.

# Borrow

*v.* To receive something with the promise of returning it.



# Summary

---





Questions?

