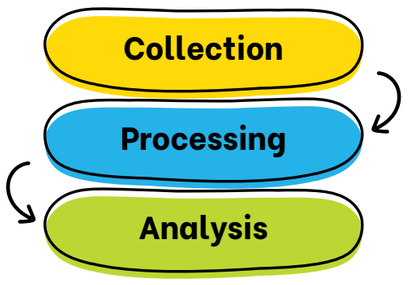
Dynamic Loading of Plugins in Rust in the absence of a stable Application Binary Interface

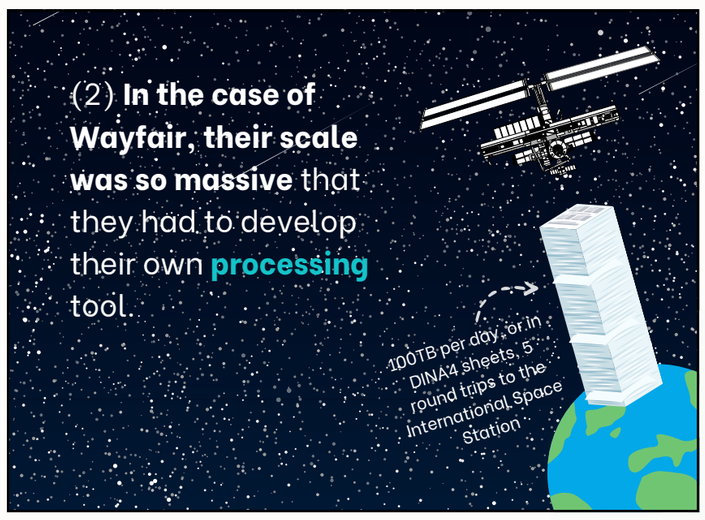
Author: Mario Ortiz Manero
Mentor: Matthias Wahl
Director: Francisco Javier Fabra Cano

This project has been possible thanks to

wayfair

(1) Decision-taking in most modern businesses is **data-driven**.

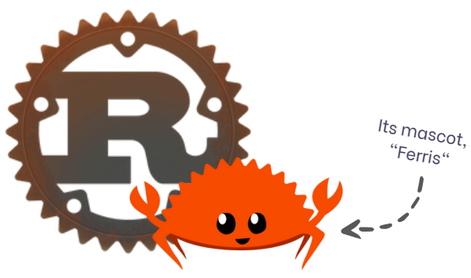




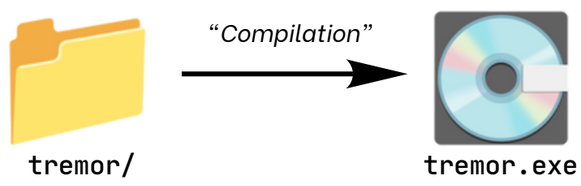
(3) It's called **Tremor**. It's open source, and took **performance** into account since the beginning.

Tremor

(4) Its high efficiency was achieved, in part, through the use of the modern programming language, **Rust**.



(5) Rust offers higher safety and performance due to having a more extensive **compilation** process.



(6) However, the fact that it is more extensive also means that **the waiting times are much longer...**

(7) My project tried to alleviate this **obstacle in the development**, through a "**plugin system**".

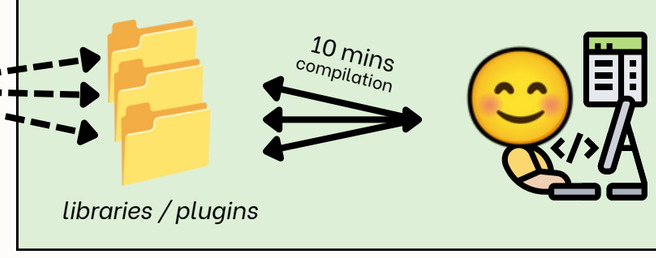
(8) I discovered that the most efficient **technology** for its implementation is named "**dynamic loading**".

(9) BEFORE

A single large library with all the code. Any changes require recompiling the whole binary, so development is slow.

(10) AFTER

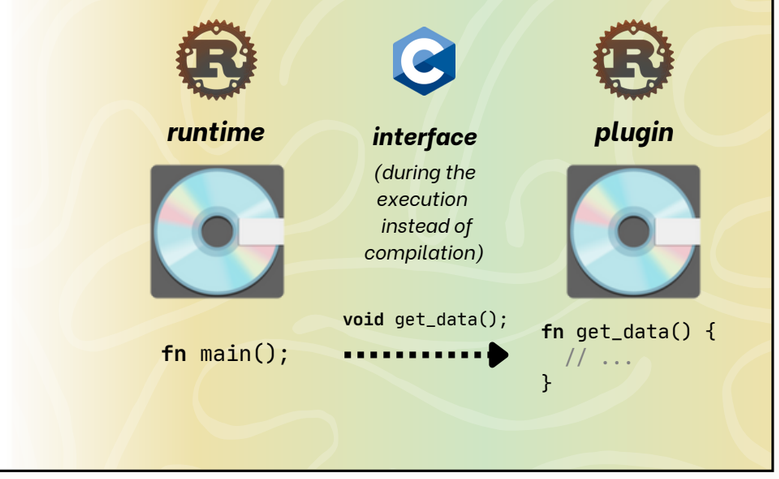
Several small, independent libraries with parts of the code. Any changes only require recompiling the binary ("plugin") to which it belongs, so development is accelerated.



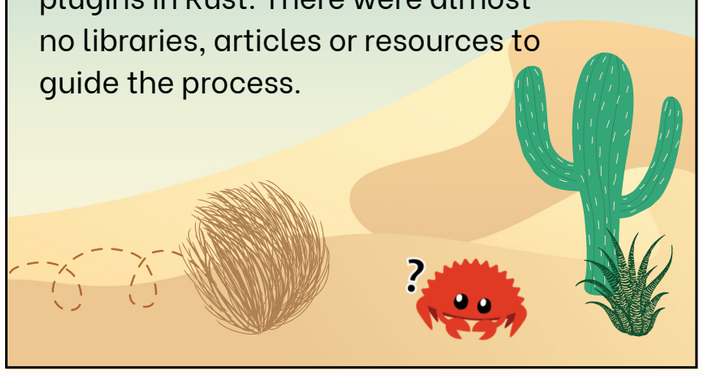
(11) In this new architecture, **plugins must export an interface** through which the main binary ("runtime") can interact with them.

(12) However, this is not directly possible with Rust ("**its Application Binary Interface isn't stable**").

(13) We will have to make **translations from C to Rust** and vice versa.



(14) Another problem is the **immaturity of the ecosystem** for plugins in Rust. There were almost no libraries, articles or resources to guide the process.



(15) Finally, I developed a **working prototype**, but due to these unexpectedly large increases in complexity, it did not achieve the expected performance.

(16) Regardless, it serves as a **very solid foundation** that the Tremor team can continue to build on.

(17) Much of my work consisted of **improving the state of the ecosystem.**

30
contributions to **>10** libraries Tremor uses

10
contributions within Tremor

(18) I also gave **2 talks** and wrote **6 technical articles** about my experience, with 30.000 views.

Talk	Talk	Personal blog
TremorCon	LFX Mentorship	nullderef.com