

## RustIEC: first user group meeting

### *Attendance:*

Roel Van Den Broeck, Devon Kerckhove, Ludo Cuypers, Andras Tilhof, Lieven Hollevoet, Michael Allwright, Luca Gattobigio, Robrecht Dewaele, Andrew Gordon, Mathieu David, Jehan Bruggeman, Tom Heremans, Lukas Wittevrongel, Jeroen Meulemeester, Hans Mortier, Stijn Volckaert, Alicia Andries, Jorn Lapon, Robrecht Blancquaert, Diana Deac, Thibaut Vandervelden, Ruben De Smet, Kris Steenhaut, An Braeken

## Introduction of project and participants

Roel Van Den Broek works at OTN Systems. The company is interested in the security aspects of the Rust programming language.

Devon Kerckhove from Ausy leads a team of developers who are interested in Rust. They see that the industry is asking about Rust, since they are interested in the security aspects of the language.

Ludo Cuypers from COMmeto wants to know what the real time capabilities are of Rust. They want to learn more about how Rust can be used on embedded devices. Furthermore, they want to know how the maintainability of Rust projects is.

Quicksand, represented by Lieven Hollevoet, uses C as their programming language for their embedded devices. They want to know more about the tools that can be used for embedded development, as well as how Rust can interact with existing code bases already written in C.

Michael Allwright, a freelancer, is interested in WebAssembly, both in back- and front-end development.

Luca Gattobogio is representing LSEC. They are interested in the security aspects of the language.

Verhaert Digital, represented by Robrecht Dewaele, wants to know more about how Rust can be used in embedded programming, but also how Rust can be used for the Internet.

Sky-Hero, with Andrew Gordon and Mathieu David, is already using Rust in production. They are interested to see if other developers are having the same problems while using Rust and see how others tackle these issues.

Digazu, represented by Jehan Bruggeman, wants to use Rust to reduce their memory footprint of an existing application that they have.

MatchID, with Lukas Wittevrongel, just wants to learn more about the Rust programming language and see how they can use for their products.

Jeroen Meulemeester represented GemOne. They want to learn Rust on their embedded devices and are interested in the security aspects of the language, but also how Rust is used for bare metal programming.

Hans Mortier represented Barco. They are interested in writing Linux kernel modules in Rust.

Tom Herremans represented Vlaio.

## First presentation: Safe Systems Programming in Rust by Stijn Volckaert

***There are tools, such as C2Rust, that can transform C and C++ to Rust. Are their tools that can transform other programming languages to Rust?***

This is technically possible. The reason why there are tools to transform C and C++ to Rust is because Rust is closer to C/C++ than any other languages. They are all systems programming languages. However, Rust is a very productive language, such as Python. Automatic translations of those languages to Rust is technically possible, but currently very difficult.

***Zig is also a new programming language that could replace C and C++. How does this programming language compares to Rust?***

There are more and more replacement programming languages for C and C++, however, they are very new and young. Zig can compile C code. Rust is a safe programming language.

Transforming a big code base is very challenging because these big code bases usually bend the rules of C/C++ and therefore cannot be translated to other safer languages. The question was asked because AWS already uses Zig. We should take a deeper look at how Zig compares to Rust as a replacement for C/C++.

***The presentation talks a lot about safety and how Rust is a safe programming language. However, when using crates (libraries), how can I be sure that those crates are safe to use without taking a look at the code? Are there tools that can help me with that?***

People are working on those tools, however, this is still very academic. Those tools try to verify if the unsafe code in crates is actually safe or not. In some cases, unsafe code can just be safe.

However, it is still unclear on how to tackle this issue. Currently, the following tools exist:

***cargo audit***: this tool checks in a public database if a crate contains vulnerabilities. This database is updated by the community and is thus not automated.

***cargo miri***: miri is a tool that can execute intermediate Rust representation. This can be compared with an interpreter for Rust code. Miri can execute tests that contain unsafe code. While executing those tests, Miri checks if the unsafe blocks don't do undefined behaviour.

***cargo geiger***: this tool shows statistics of the usage of the unsafe keyword in a codebase and its dependencies.

There are definitely more tools available and these should be checked.

## Second presentation: The Rust Ecosystem by Ruben De Smet and Thibaut Vandervelden

***Embassy is a framework for using async functions on embedded devices. The embedded-hal library has traits for different embedded peripherals. Are there async traits for embedded?***

Embassy currently has its own async traits for the peripherals. However, the community is currently discussing how the API of the embedded-hal-async crate should look like. The maintainer of the Embassy framework is involved in these discussions. The async traits for embedded will be available in the near future.

***How should one learn Rust?***

The best way to learn Rust (our opinion) is by following the official Rust book, which is available online. <https://doc.rust-lang.org/book/>

## Wrapping up

Ruben De Smet explains the milestones and the outputs of the RustIEC project. These are the questions that were asked:

***What is the status of kernel space drivers written in Rust?***

This is very much work in progress. A lot of things are not yet supported. Therefore, a lot of unsafe code is required to communicate with the kernel.

***How is compilation done for Rust kernel drivers?***

The best way is to build the out of tree (fewer panics by the compiler) and then loading them as a module in Linux.

Jorn Lapon mentions that Gent organizes an event on the 25<sup>th</sup> of January. This event will have presentations about Rust and testimonials of companies.

***[Tom Heremans] How are you going to interact with the user groups?***

Mails from us, regular user meetings.

***[Tom Heremans] How can they (the industry) steer the project?***

They can send emails to us (Ruben and An). Also, the website has links to different profiles with emails and phone numbers. During every session we ask how the industry wants to steer the project.

Closing with networking reception