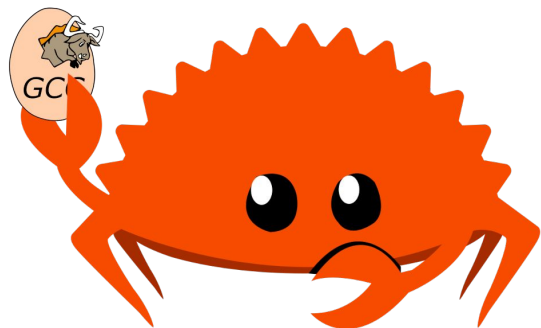




OPEN  
SOURCE  
SECURITY

# Rust-GCC

February - August



Arthur Cohen



# Introduction

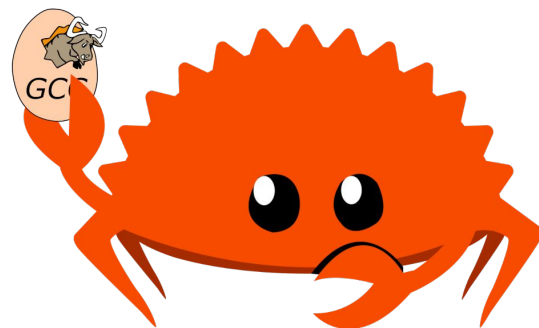
- What is Rust?
- What is GCC?
- What is Rust-GCC?
  - Why make another Rust compiler?

# Subject

- Project history
- Contributing
- Open-source and free software
- Expected work
  - Macros
  - Macro repetition
  - Overflow traps

# gccrs

- Project started in 2014
- Started again full-time in 2019 with Philip
- 1.6k stars on github
- Active community
  - GCC contributors, testers, committee members
  - Rust users, advisors and testers
- Lots of side projects as well!
  - A testing dashboard
  - A website
  - Weekly reports
  - Integration with Rust's build system



# Subject

- Project history
- Contributing
- Open-source and free software
- Expected work
  - Macros
  - Macro repetition
  - Overflow traps

# Macros

- More powerful than C macros
- Safer! (hygiene, “typed” arguments, recursion limits...)
- Allow for repetition
- Another language within the language!
- Used all the time in Rust (*println!()*, *assert\_eq!()*, *dbg!()*, ...)

# Engineering approach

- Contributing and contributing tools
- Calls
- Community
- Testing, side projects
- Constant reviewing
- Working in a massive codebase
  - Lots of legacy code on the GCC side
  - C++11
  - 19 million lines of code
  - 140k are ours!

# Added value

- Macro repetitions
- Privacy pass
- Const generics
- Borrow-checking debates, decision, research and PoC
- Web-app for the dashboard
  - Backend
  - Frontend
  - In Rust!
- It's not over!



# Privacy rules

- Encapsulation
- Prevents users from accessing inner library functions
- Annoys the programmer :)

# Const generics

- Constant values
- Constant-time evaluation
- Constant values initialized with constant functions!
- Complex constructs such as conditional statements inside functions

# Borrow-checking

- One of the core concepts of Rust
- Roughly a static analysis pass to detect memory errors
- Prevents double-frees, memory-leaks, dangling pointers, use-after-frees...
- We can't not have one!

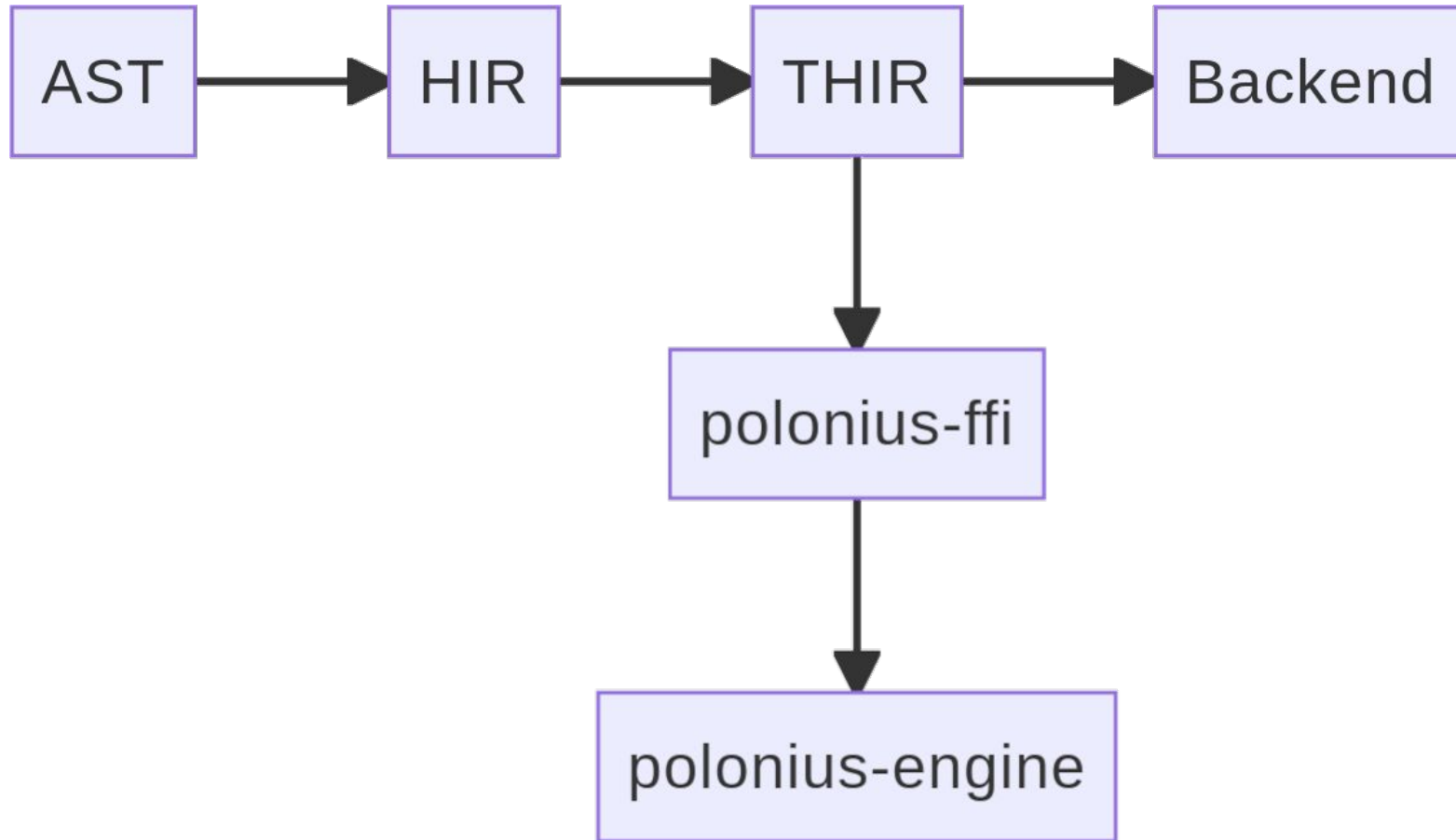
# Testing dashboard

## Rust-GCC testing dashboard

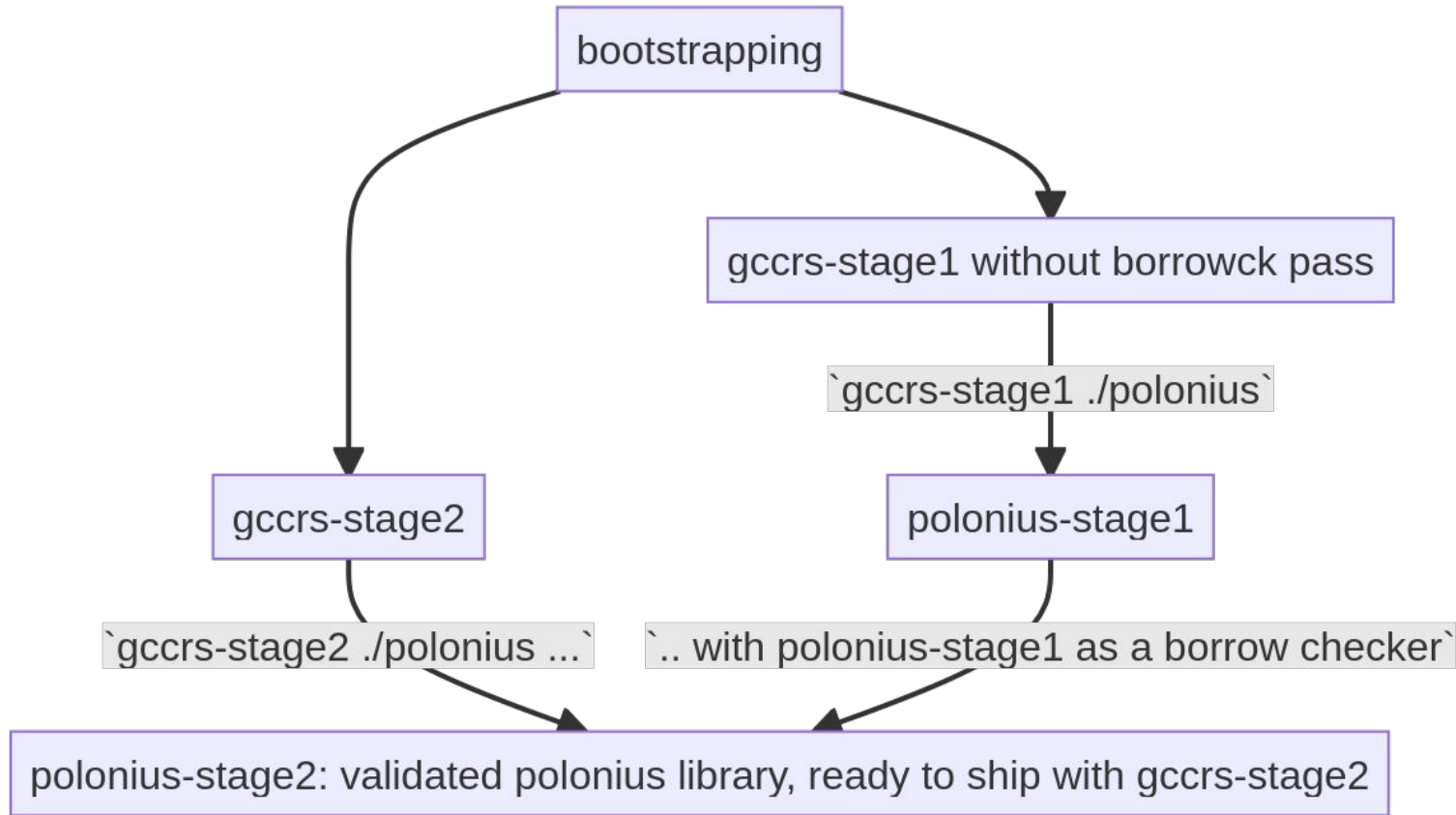
Welcome! Here you'll find a collection of the results we're accumulating while testing our compiler on various test-suites



# Borrow-checking



# Borrow-checking



|                 |               |         |          |
|-----------------|---------------|---------|----------|
| PROJECT TITLE   | gccrs         | COMPANY | Embecosm |
| PROJECT MANAGER | Philip Herron | DATE    | 12/03/18 |



# Conclusion

- I got a job
- Felt like I made a sizeable impact on the project, which is a fantastic feeling
- Best work experience of my life
- Thank you so, so much Philip





OPEN  
SOURCE  
SECURITY

# Questions?

[www.embecosm.com](http://www.embecosm.com)

