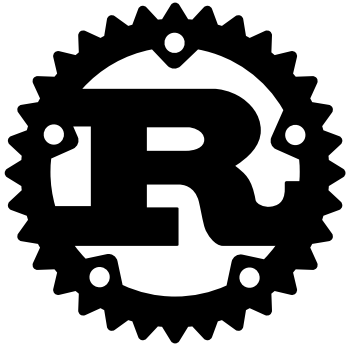
The Oxidation Plan

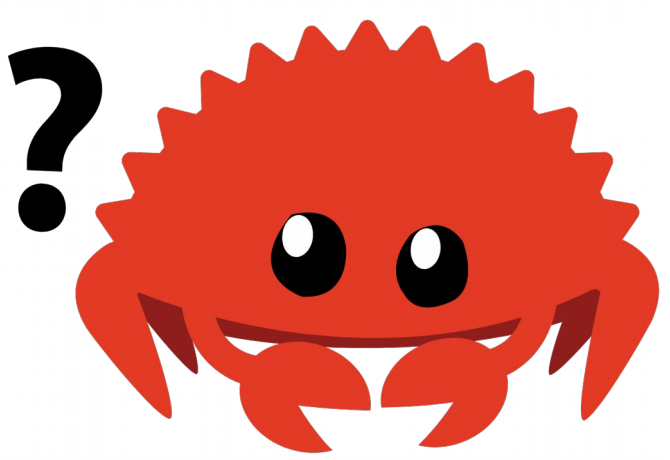
Using Rust in Mercurial



Mercurial Conference — Paris 2019

Raphaël Gomès @ OCTOBUS

Why Rust?



Rust

- Low-level language
- Strong type system
- No garbage collector
- Compile-time memory safety
- Easy(-ier) concurrency

Maintainability

Current optimized code is in C

Rust:

- Higher signal-to-noise ratio
- Stronger compile-time checks
- Modern and standardized tooling
- Safe by default

Performance

- Comparable to C in single-threaded workloads
- Multi-threaded code is simpler to write and safer

Performance

“hg status” experiment

by Valentin Gatien-Baron @ Jane Street

	hg	Rust hg
status	2.4s	50ms
status -u	2.4s	39ms
status -mard	400ms	14ms

Hindsight

- Gradual approach
- Refactor existing code
- High-level developer documentation
- More unit tests

Current progress

Done 

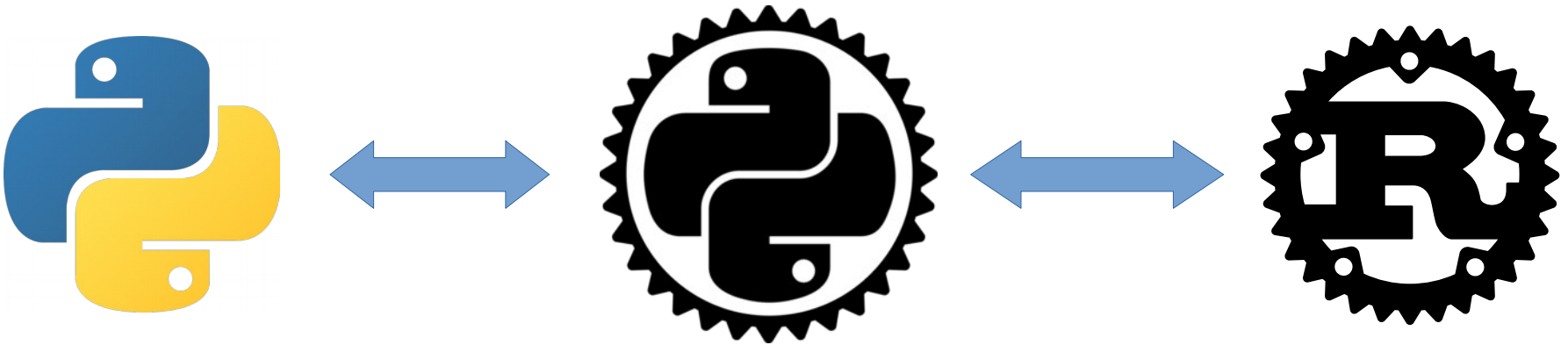
- Ancestors
- Discovery
- .hgignore parsing

In progress

- Persistent nodemap
- Dirstate
- Rust as a first-class citizen

What's next?

Reduce friction with Python



- Waste of CPU and memory
- Complex interface code

Benchmarks

- Test against extreme or pathological cases
- Prevent performance regressions

Extensions support

- Add hooks for Python
- Backwards compatibility
- Eventually define an interface

Fast path

- Rust “hg” executable
- Faster startup
- Fallback to Python

Thank you!