VoidPhone Project
P2PSEC (IN2194)
Prof. Dr.-Ing Georg Carle
Sree Harsha Totakura, Dr. Heiko Niedermayer

Initial Approach Report
Team 45 - Rhodium
Benedikt Seidl, Stefan Su
Due Date: 05/25/18

Team Composition

We are Benedikt Seidl and Stefan Su, both studying Informatics in the sixth semester of our Bachelor's degree. As a team name we chose "Rhodium" [1] and will be working on the Distributed Hash Table (DHT) project.

Programming Language

We decided to use Rust [1] to implement the DHT. Rust is a modern system programming language with features known from high level programming languages. It includes a strong type system and supports object orientation as well as functional programming. This allows to write safe and fast code on a high abstraction level. One does not have to pay for this with runtime costs like garbage collection or interpreters as Rust compiles to native LLVM code using zero-cost abstractions. [2]

Operating System

As operating systems, we use macOS and Linux which have the advantage of being Unix-based and thus supporting most development tools and libraries that may be needed during development. However, since Rust supports all major operating systems, it should also be possible to run the software under Windows, at least as long as we do not require any special libraries.

Build System

Rust comes with its own build system called "Cargo". [3] Cargo serves as a build tool but can do much more. It only compiles files that have changed since the last compilation. Furthermore, it serves as a test runner, can create HTML documentation based on inline comments and manages the depencies of our project.

Since Cargo is so easy to setup, we currently do not plan to provide further build files for Docker or Vagrant. If it turns out that our build process is more complicated than expected, we can still integrate a more elaborate build system.

Quality Measures

Rust supports writing automated unit and integration tests natively and with Cargo it is a breeze to run them. By applying the concept of test driven development (TDD) we make sure that our code matches the intended design. We also aspire to maintain a high test coverage.

Furthermore, Rust ensures by its language design that a lot of common bugs cannot occur. It does not allow invalid memory access such as use after free or null pointers and its strong type system guarantees that all type errors are caught at compile time.

[1] Rhodium is the element with atomic number 45.

Available Libraries

We can obtain packages (= dependencies) from the central package registry crates.io [5] by referencing external crates in our project dependencies.

explain crates.io

Software License

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

use AGPL to get improvements back to the open source community

Previous Programming Experience within the Team

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Workload Sharing

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

References

[1] *The Rust Programming Language.* Available: `https://www.rust-lang.org/en-US/index.html` [Accessed: May 2018].

[2] *Frequently Asked Questions.* Available: `https://www.rust-lang.org/en-US/faq.html` [Accessed: May 2018].

[3] *The Cargo Book.* Available: `https://doc.rust-lang.org/cargo/index.html`[Accessed: May 2018].

[4] N. Matsakis, A. Turon. *The Rust Programming Language*, 2nd edition. Available: `https://doc.rust-lang.org/stable/book/second-edition/` [Accessed: May 2018].

[5] *crates.io Rust Package Registry.* Available: `https://crates.io/` [Accessed: May 2018].