

A deep dive into tooling

---

# NTP in Rust



**tweede golf**

web / security / embedded

# Who am I

---

David Venhoek

Software developer at Tweede Golf

Background in Physics and Mathematics

Working on NTP for ISRG's Prossimo project, building ntpd-rs



# Overview

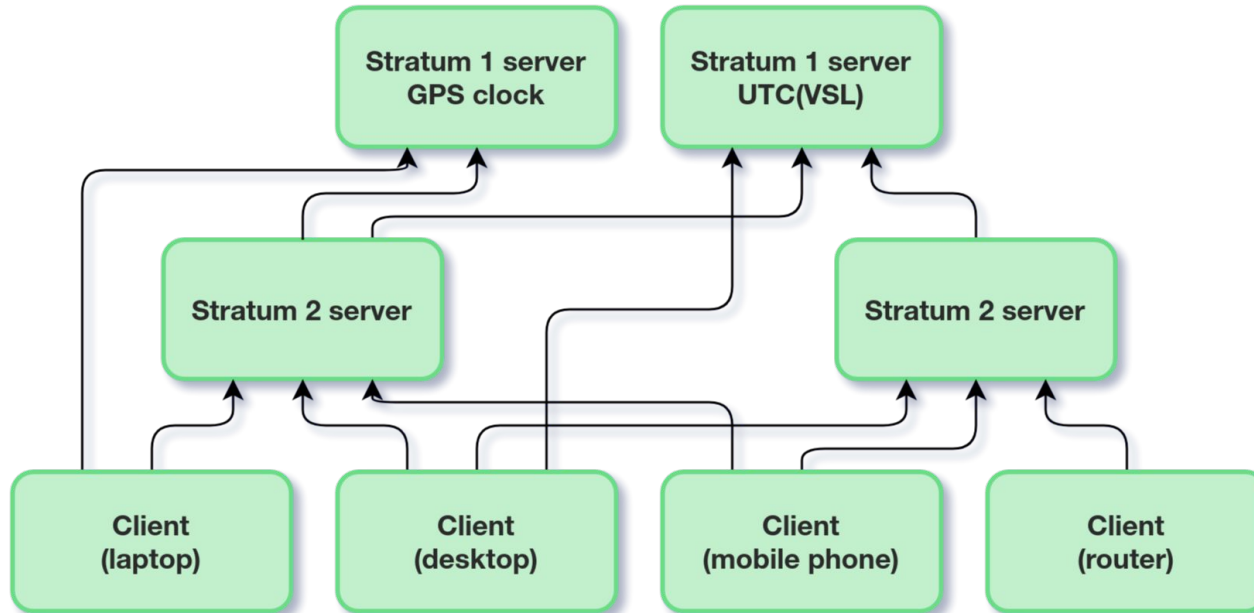
---

- What is NTP?
- Why use Rust?
- Tooling deep dive: Fuzzing

# What is NTP?

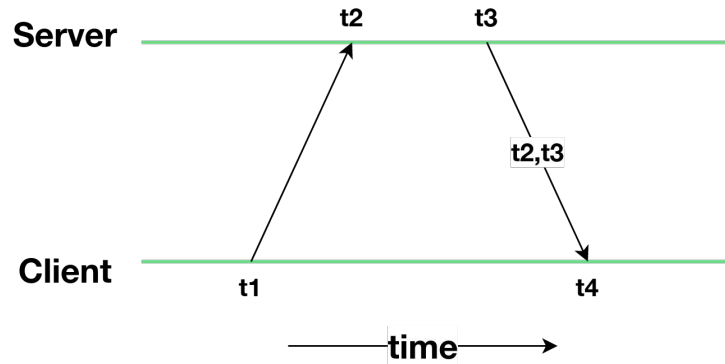
---

Synchronization of clocks over the internet.



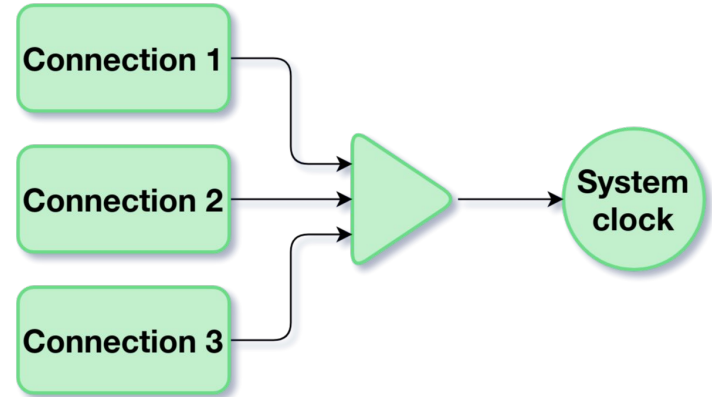
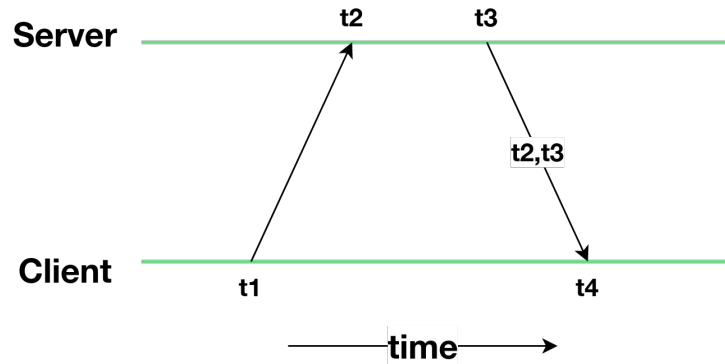
# What is NTP?

---



# What is NTP?

---



# Why Rust?

---

## Current Implementations:

- NTP reference implementation (NTPD)
  - C
  - Lots of security issues
- NTPSec
  - C, Python (small parts)
  - Stripped down NTPD, some extra features
- Chrony
  - C
  - From scratch, significantly better than NTPD

# Why Rust?

---

Memory safety



# Why Rust?

---

Memory safety

But also, better tooling:

- Type safety
- Dependency management
- Easy, integrated testing support
- **Fuzzing tools**

# Fuzzing

---

What is it?

- Testing with random inputs
- Find crashes/panics
- Also useful for testing invariants

Useful for:

- Checking parsing code
- Property testing algorithms/data structures

# Fuzzing

---

Cargo fuzz provides excellent tooling

Install:

```
cargo install cargo-fuzz
```

Project Setup:

```
cargo fuzz init
```

Example, run with

```
cargo +nightly fuzz run <example>
```

```
#![no_main]
use libfuzzer_sys::fuzz_target;

fuzz_target!(|t: (isize, isize)| {
    let _ = t.0 + t.1;
});
```

# Fuzzing

---

ntpd-rs examples

# Useful URLs

---

- Rust fuzz book: <https://rust-fuzz.github.io/book/introduction.html>
- ntpd-rs: <https://github.com/memorysafety/ntpd-rs>
- Prossimo NTP project: <https://www.memorysafety.org/initiative/ntp/>



# tweede golf

**web / security / embedded**

Castellastraat 26, 6512 EX Nijmegen

[info@tweedegolf.com](mailto:info@tweedegolf.com)

024 3010 484