sett: data encryption and transfer made easy(ier)

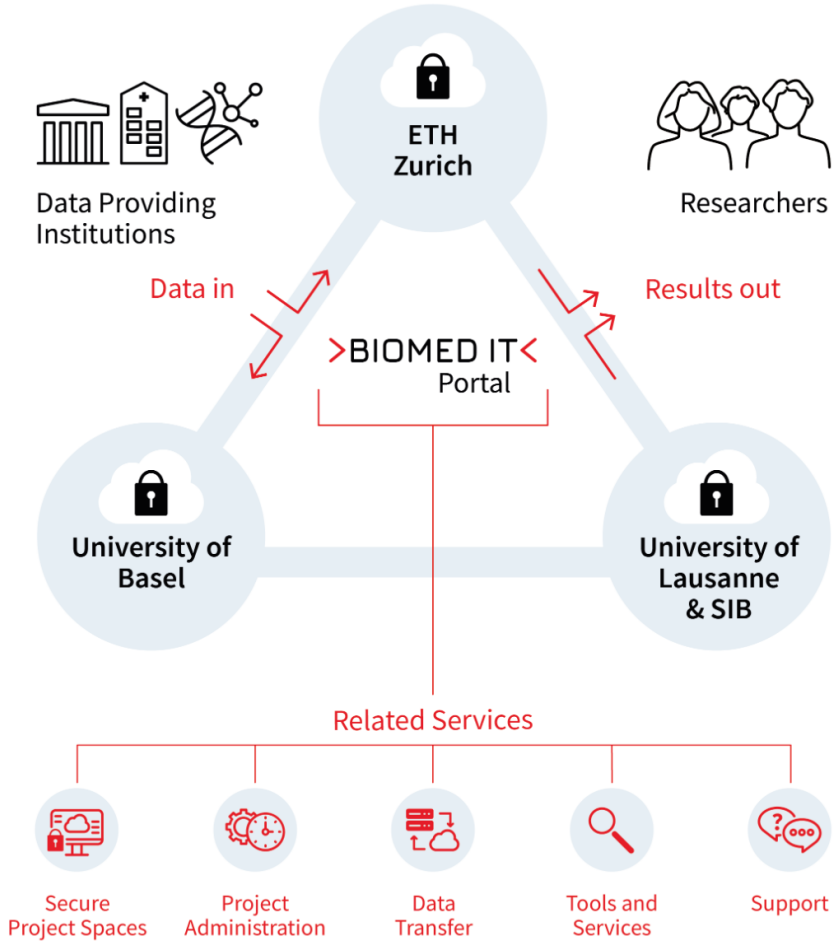
by Christian Ribeaud and Jarosław Surkont

biomedit.gitlab.io/presentations/sett-zurich-2023

Outline

- **BioMedIT** mission 👩‍🚀
- 1st attempt (production): Python 🐍
- 2nd attempt (in progress): Rust 🦀
- Maintenance 🛠️
- Future plans 🔮
- Summing up ⭐

Big picture


ETH
Zurich
Data Providing
Institutions
Researchers
Data in
Results out
>BIOMED IT<
Portal
University of
Basel
University of
Lausanne
& SIB
Related Services
Secure
Project Spaces
Project
Administration
Data
Transfer
Tools and
Services
Support

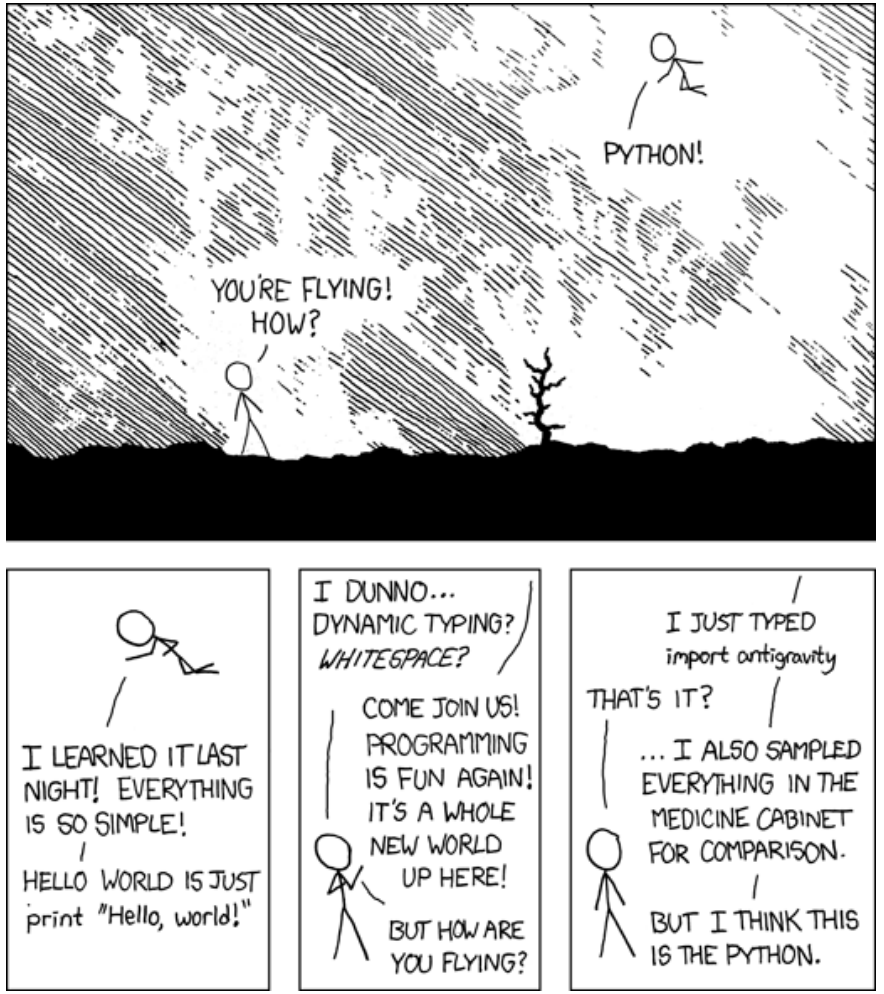
sett - Characteristics

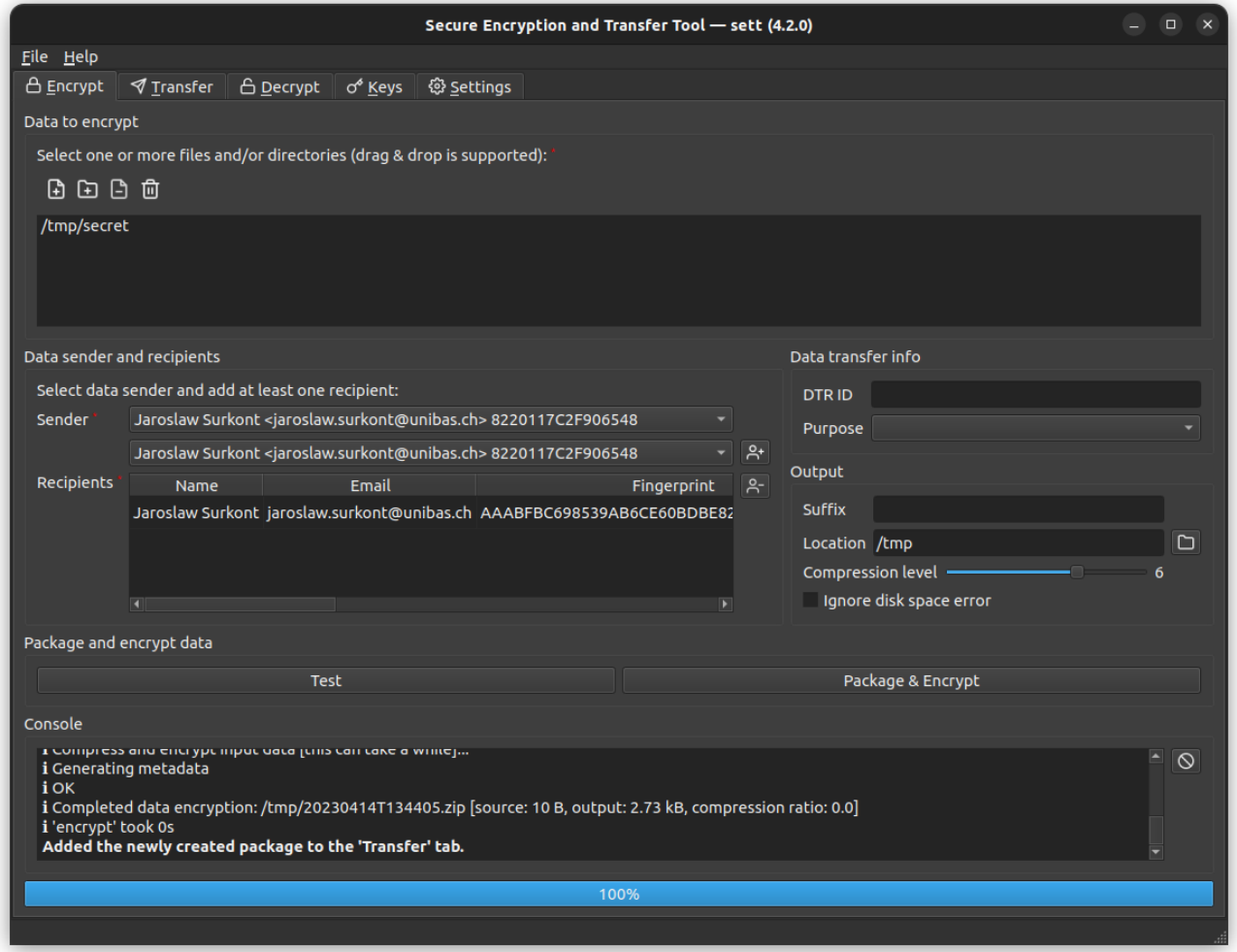
- End-to-end encryption (**OpenPGP**)
- Trust: (meta)data signing (**OpenPGP**)
- Data compression (**gzip**)
- Data integrity validation (**sha256**)
- Packaging: a single, self-contained file (**zip**)
- Transfer (**sftp**, **s3**)
- Cross-platform support
- **CLI** and **GUI**
- Interoperability with other tools

UX: Keep it easy!

Package structure

Python 🐍


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.


Secure Encryption and Transfer Tool — sett (4.2.0)
File Help
Encrypt
Transfer
Decrypt
Keys
Settings
Data to encrypt
Select one or more files and/or directories (drag & drop is supported):
/tmp/secret
Data sender and recipients
Select data sender and add at least one recipient:
Sender
Jaroslaw Surkont <jaroslaw.surkont@unibas.ch> 8220117C2F906548
Jaroslaw Surkont <jaroslaw.surkont@unibas.ch> 8220117C2F906548
Recipients
Name
Email
Fingerprint
Jaroslaw Surkont
jaroslaw.surkont@unibas.ch
AAABFBC698539AB6CE60BDBE82
Data transfer info
DTR ID
Purpose
Output
Suffix
Location
/tmp
Compression level
6
Ignore disk space error
Package and encrypt data
Test
Package & Encrypt
Console
i Generating metadata
i OK
i Completed data encryption: /tmp/20230414T134405.zip [source: 10 B, output: 2.73 kB, compression ratio: 0.0]
i 'encrypt' took 0s
Added the newly created package to the 'Transfer' tab.
100%

Support

- **Linux** (including **CentOS 7** 🤯 - EOL 2024-06-30), **MacOS**, **Windows**
- **Python** 3.7-3.11
- **GnuPG** 2.0.22 (EOL 2017-12-31 💣) - latest (with gpg-lite)

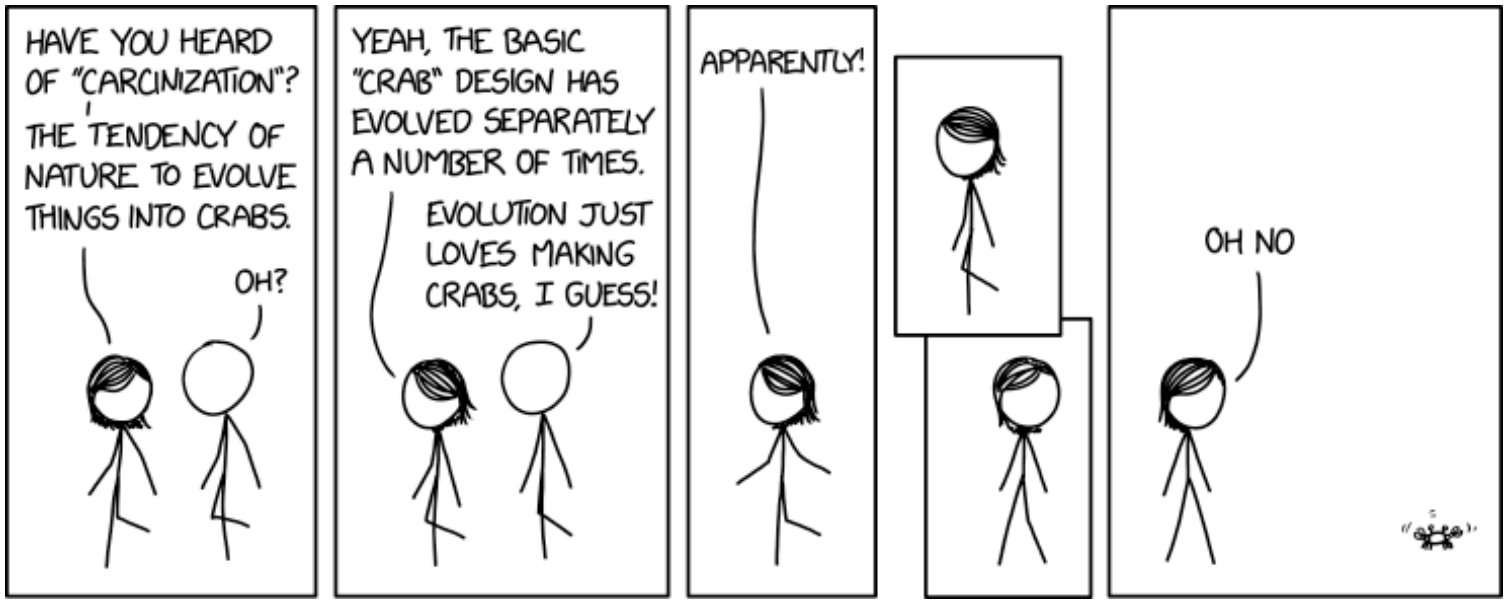
Lessons learned

- **Cross-platform** support is difficult, especially if none of the developers uses a given platform.
- Relying on packages installed separately (**Python**, **GnuPG**, **Qt**, **glibc**) adds a significant development cost and you'd better master them!
- App installation/upgrade is challenging to end users.

What can we do? 🤷‍♀️

1. Reduce the number of dependencies: **GnuPG** is the most problematic one!
2. Rewrite everything from scratch in **JavaScript**! Everything is a browser and the browser is the new **OS**.
3. Surgical replacement of specific workflows (encryption, transfer, decryption, **OpenPGP** key management) and/or user interfaces (**CLI**, **GUI**) in **Rust**.

Let's rewrite it in 🦀!



Reasons

- Portability
- Security
- Correctness: capture bugs earlier
- Ecosystem
  - SequoiaPGP
  - PyO3

Challenges

- Steep learning curve: you need an expert and a lot of motivation!
- In the transition period, both 🦀 and 🐍 codebases need maintenance.

SequoiaPGP

- A new OpenPGP implementation (**GnuPG** replacement)
- **v1.0.0** released on *16th December 2020*
- Library-first approach
- Active community, ready to help and attentive
- An ecosystem, e.g. sq, keys.openpgp.org

Python bindings

PyO3

Rust bindings for Python, including tools for creating native Python extension modules. Running and interacting with Python code from a Rust binary is also supported.

Maturin

Build and publish crates with pyo3, rust-cpython, cffi and uniffi bindings as well as rust binaries as python packages.

Repository Structure

- sett - the main lib+bin crate
- sett-rs - Python bindings to the main library

```
❯ sett
Rust port of sett (data compression, encryption and transfer tool).

Usage: sett <COMMAND>

Commands:
  encrypt    Encrypt data package
  decrypt    Decrypt data package
  transfer   Transfer data package
  certstore  PGP key management
  help       Print this message or the help of the given subcommand(s

Options:
  -h, --help     Print help
  -V, --version  Print version
```

High-level API

```rust
pub fn encrypt<P: AsRef<Path>, S: AsRef<str>, A: AsRef<[S]>, T: Encrypt>(
    opts: &EncryptOpts<P, S, A, impl Progress>,
    dest: &T,
) -> Result<Option<String>> {}

pub trait Encrypt {
    fn encrypt<P: AsRef<Path>, S: AsRef<str>, A: AsRef<[S]>>(
        &self,
        opts: &EncryptOpts<P, S, A, impl Progress>,
    ) -> Result<Option<String>>;
}

pub enum Destination<'a> {
    Local(local::LocalOpts<'a>),
    Sftp(sftp::SftpOpts<'a>),
    S3(s3::S3Opts<'a>),
}
impl Encrypt for Destination<'_> {}
```

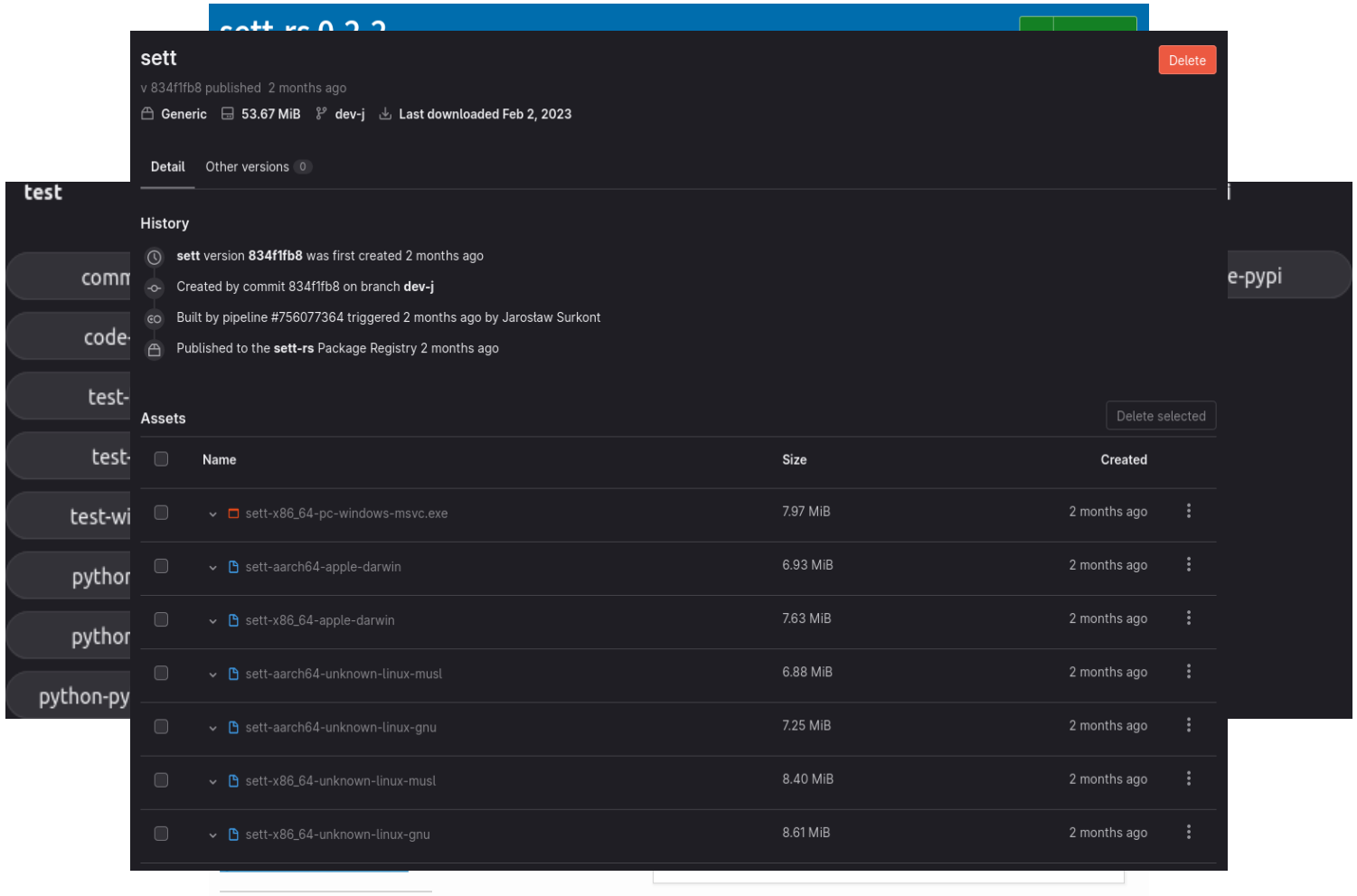
Python bindings (sett-rs)

```rust
#[pyfunction]
fn encrypt(
    _py: Python,
    opts: EncryptOpts,
    dest: Destination,
    progress: Option<Callable<'_>>,
    two_factor_callback: Option<Callable<'_>>,
) -> PyResult<Option<String>> {}

#[pymodule]
fn sett_rs(_py: Python, m: &PyModule) -> PyResult<()> {
    pyo3_log::init();
    m.add_function(wrap_pyfunction!(encrypt, m)?)?;
    Ok(())
}

// from sett_rs import encrypt, EncryptOpts, Destination
// encrypt(EncryptOpts(...), Destination(...), None, None)
```

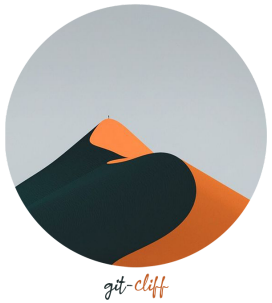
Maintenance

Living with Rust Long-Term - Jon Gjengset

CI/CD 💪


sett
v 834f1fb8 published 2 months ago
Generic
53.67 MiB
dev-j
Last downloaded Feb 2, 2023
Delete
Detail
Other versions 0
History
sett version 834f1fb8 was first created 2 months ago
Created by commit 834f1fb8 on branch dev-j
Built by pipeline #756077364 triggered 2 months ago by Jarosław Surkont
Published to the sett-rs Package Registry 2 months ago
Assets
Delete selected
Name
Size
Created
sett-x86_64-pc-windows-msvc.exe 7.97 MiB 2 months ago
sett-aarch64-apple-darwin 6.93 MiB 2 months ago
sett-x86_64-apple-darwin 7.63 MiB 2 months ago
sett-aarch64-unknown-linux-musl 6.88 MiB 2 months ago
sett-aarch64-unknown-linux-gnu 7.25 MiB 2 months ago
sett-x86_64-unknown-linux-musl 8.40 MiB 2 months ago
sett-x86_64-unknown-linux-gnu 8.61 MiB 2 months ago

Releases 🚀

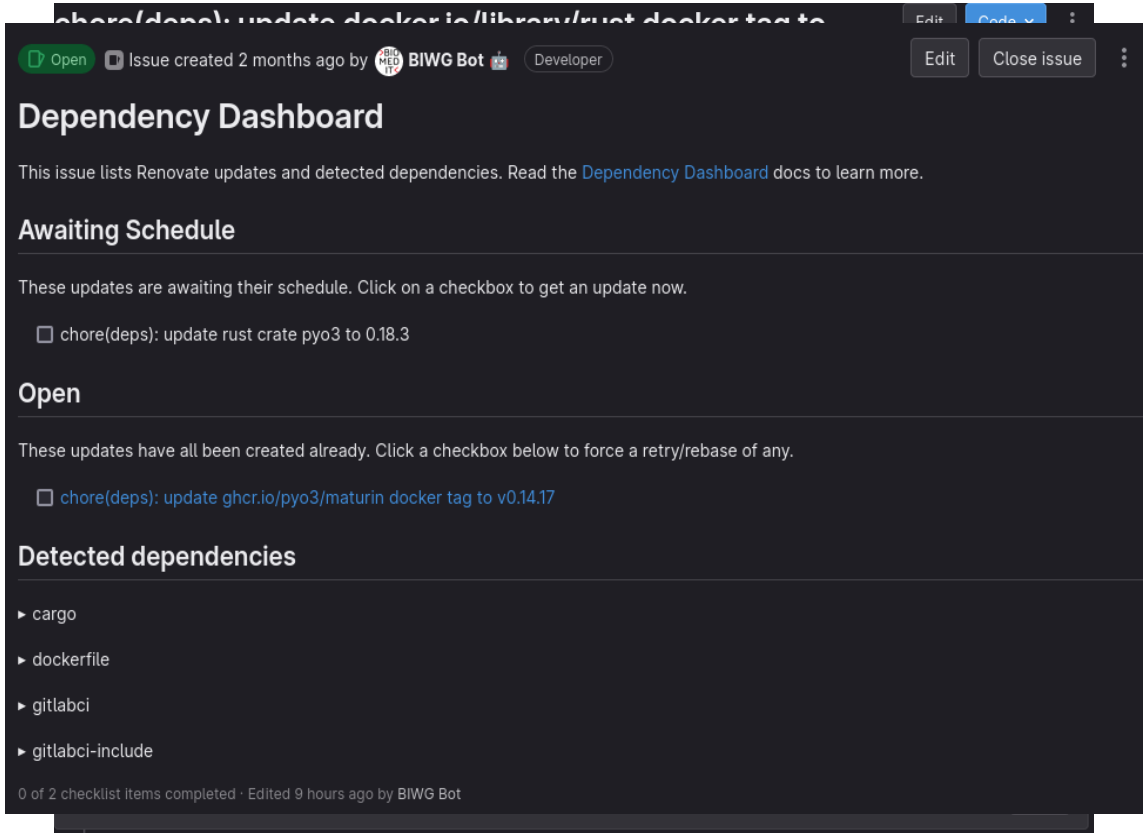


git-cliff can generate changelog files from the Git history by utilizing conventional commits as well as regex-powered custom parsers. The changelog template can be customized with a configuration file to match the desired format.

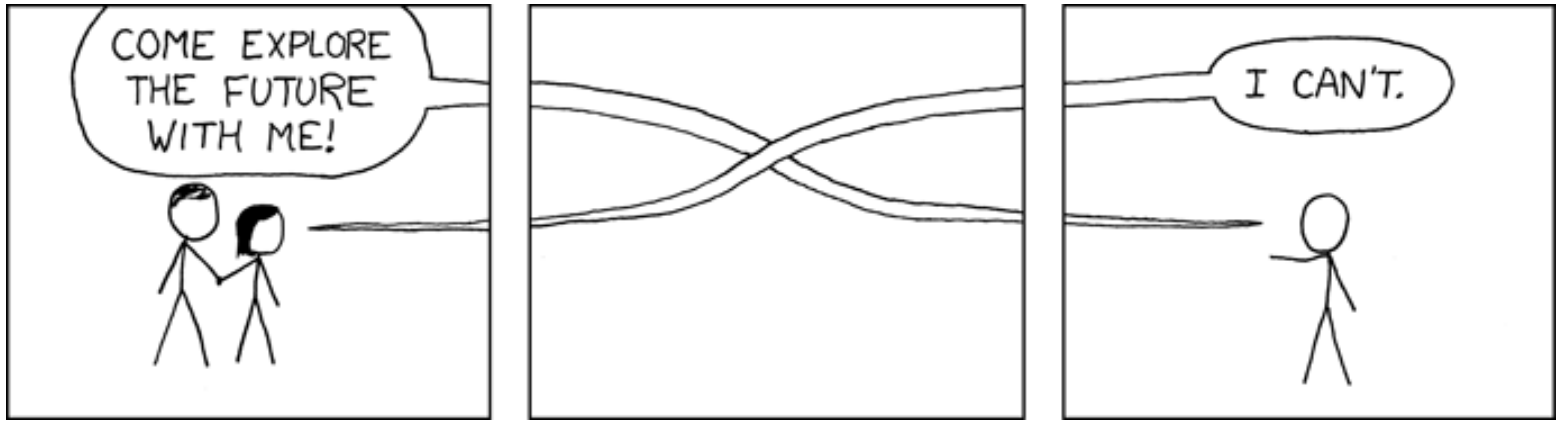
Separate releases for **sett** and **sett-rs** crates automated with a simple script.

Stay up-to-date

Renovate bot

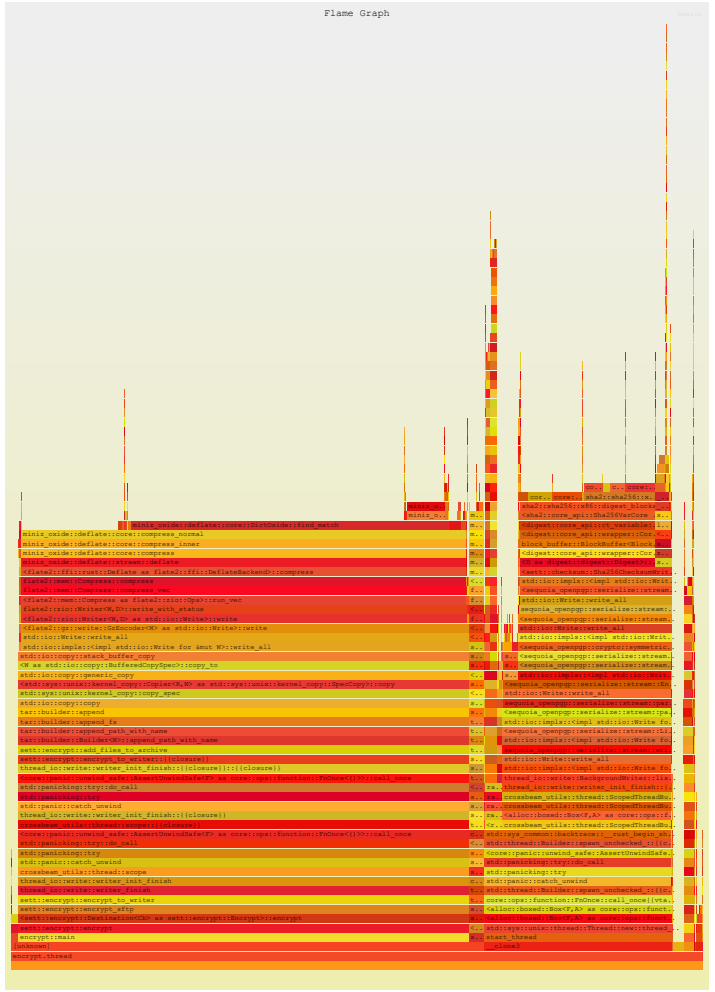


What's next?



Performance improvements

- Multithreaded compression (gzp)
- Streaming support for **S3**

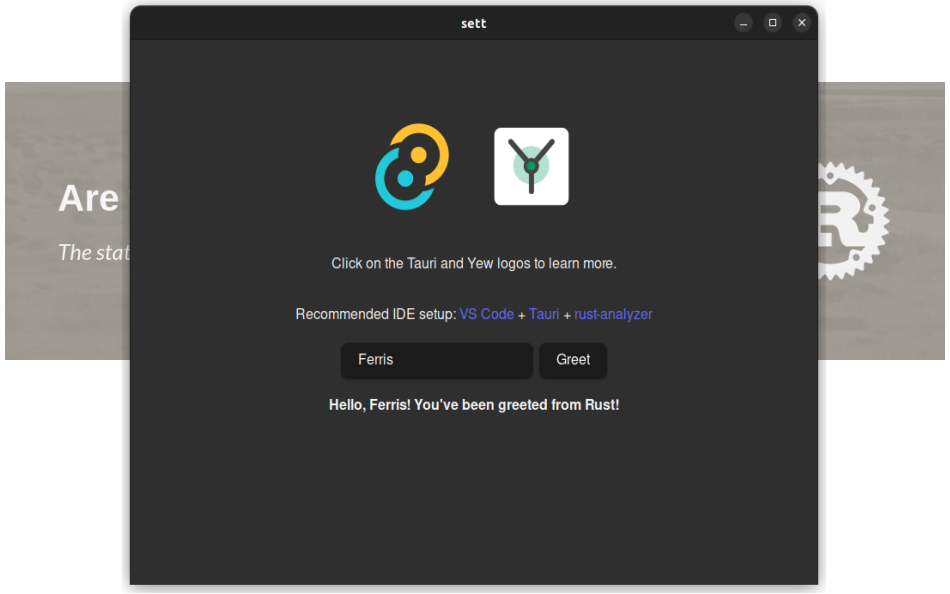


Packaging and distribution 📦

- Easy (native) installer
- Self updater
- **CLI** and **GUI** in one executable

GUI

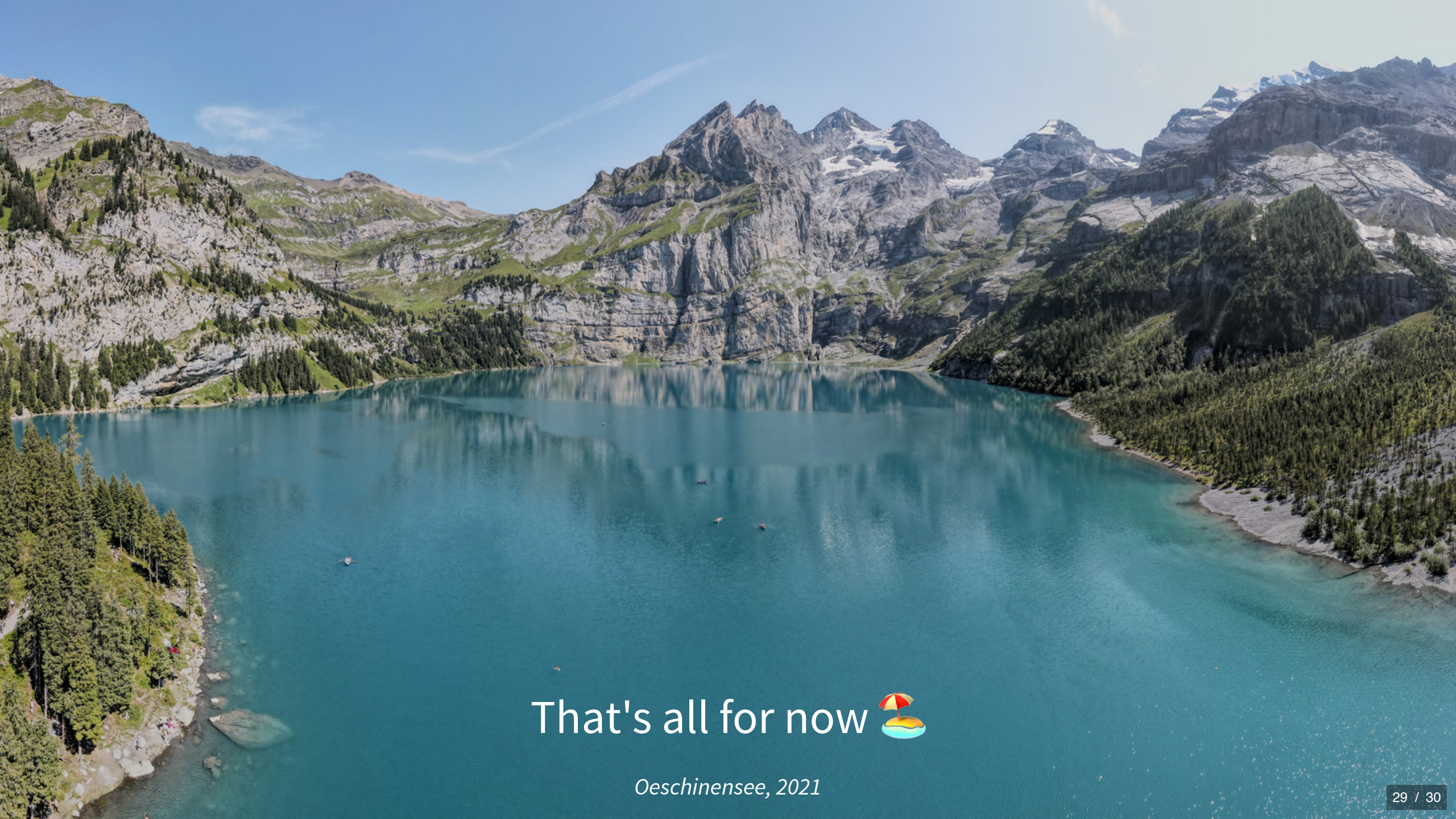
Tauri + Yew?



sett in a browser with WebAssembly?

Summary

- Python
  - Easy initial implementation
  - Difficult to maintain
  - Complicated distribution
- Rust
  - Steep learning curve
  - Early bug detection
  - Easy to maintain and distribute
- Good CI/CD and dependency update automation saves time and effort in the long-term.



That's all for now 🏝️

Oeschinensee, 2021

References

- sett-rs 🦀
- sett 🐍
- biomedit.ch