



Memory Safe *

丁羽, Baidu X-Lab

dingyu02@baidu.com

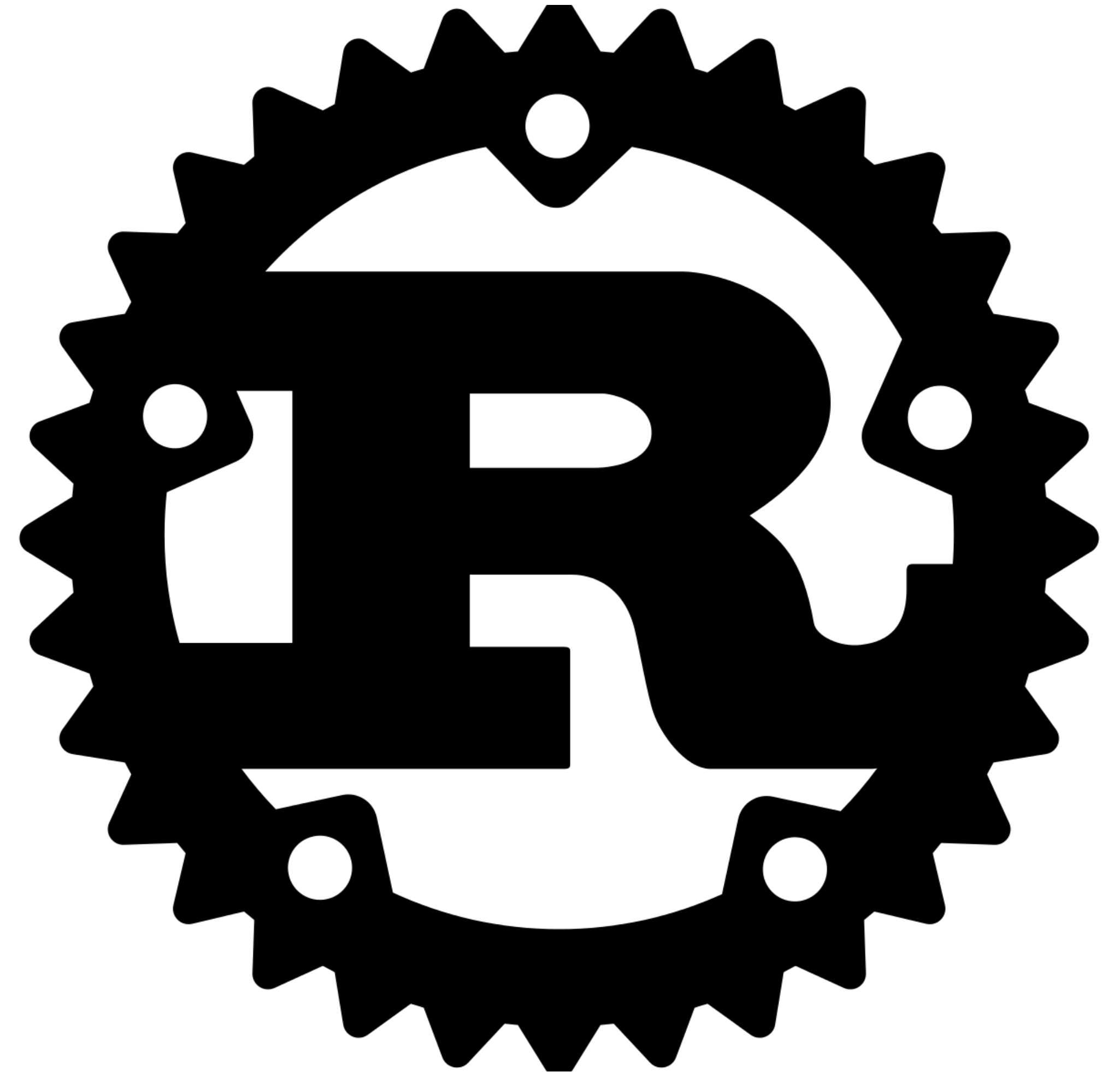


“Rust just happens to be a language that is well known for acing all the things that Go can’t do.”

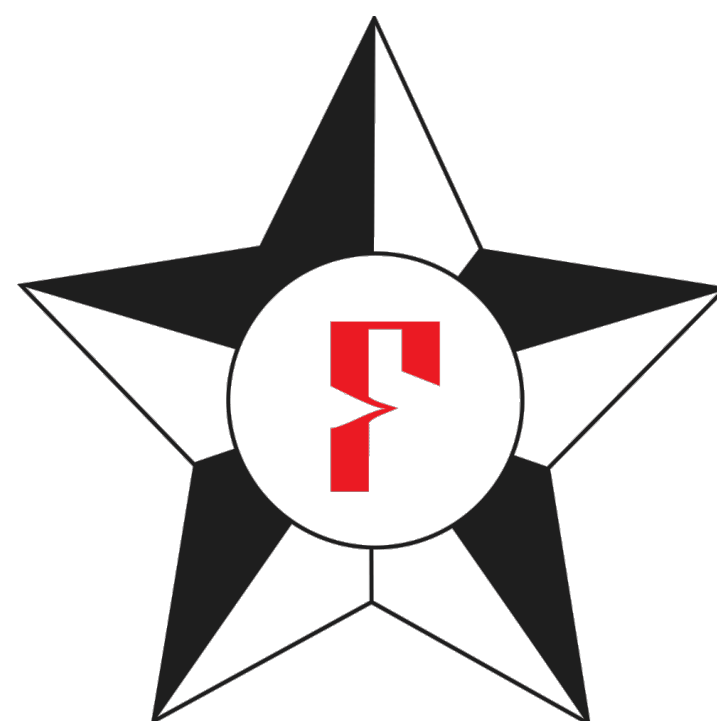
*–George Hosu,
The success of Go heralds that of Rust*

The Good

- High performance
- zero-cost abstractions
- move semantics
- guaranteed memory safety
- trait-based generics
- pattern matching
- type inference
- minimal runtime
- efficient C bindings

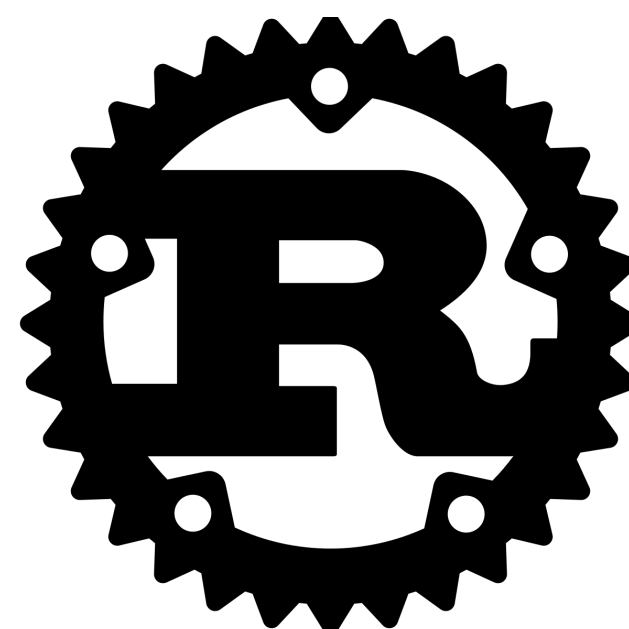


History



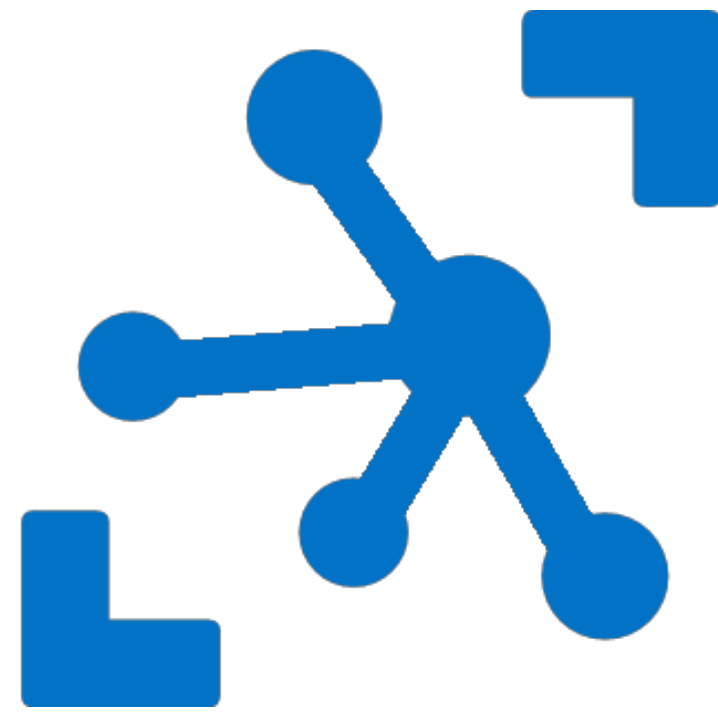
<https://www.fstar-lang.org>

Moz://a



<https://www.rust-lang.org>

Partially in Rust



Firecracker

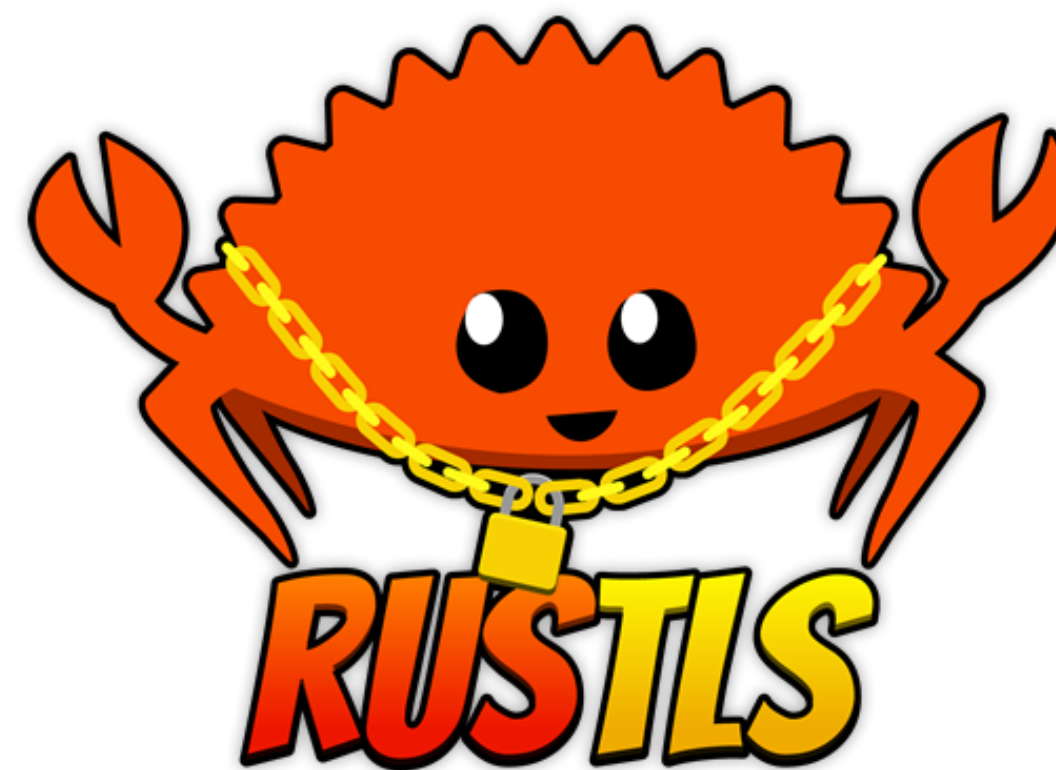
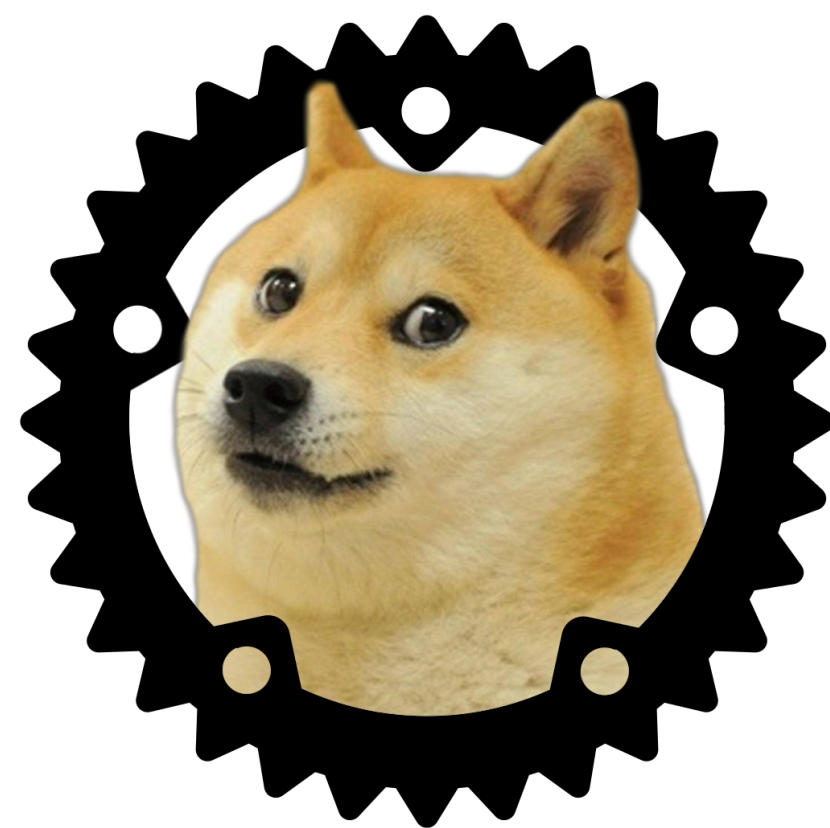


Chrome OS

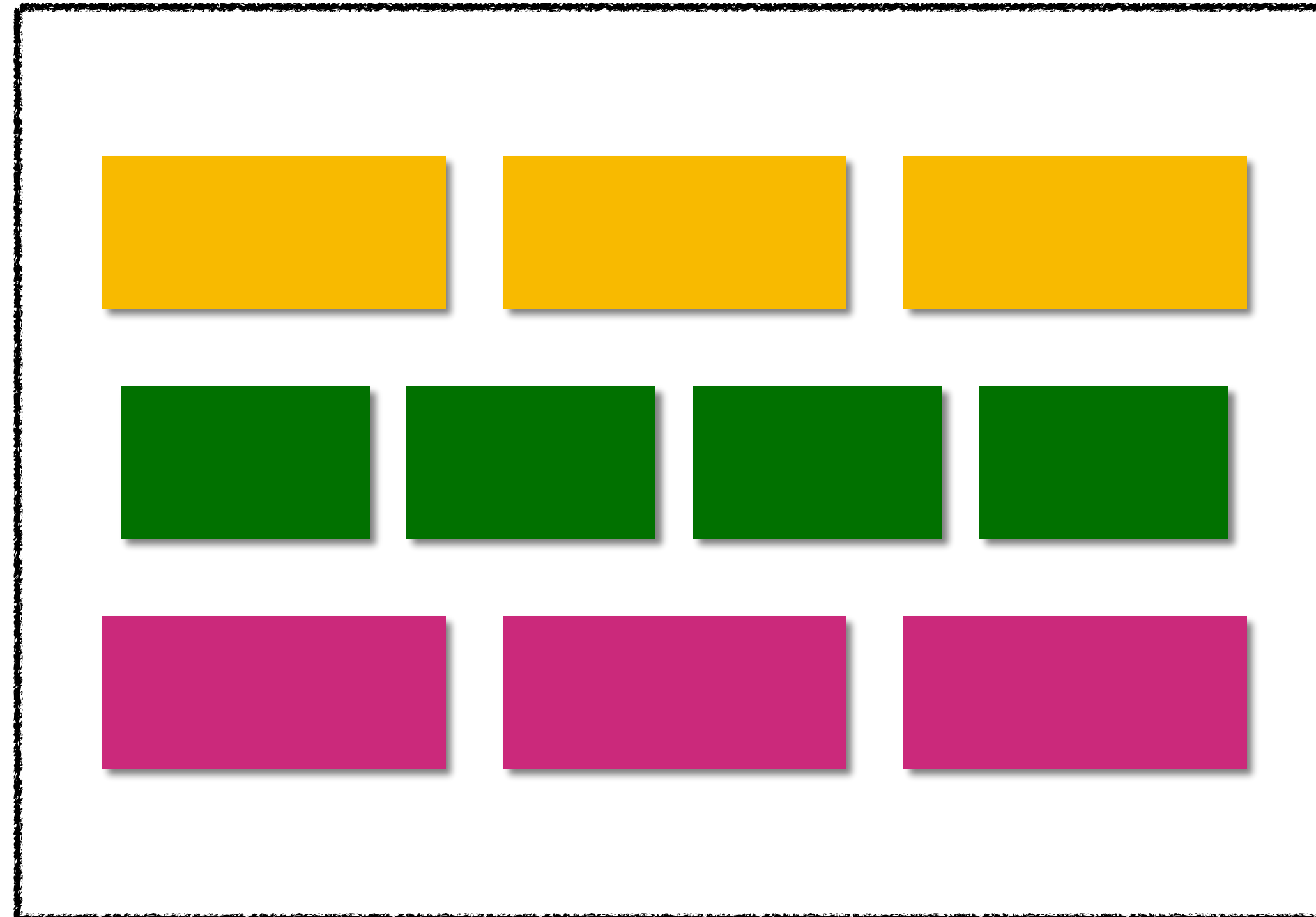


Chromium OS

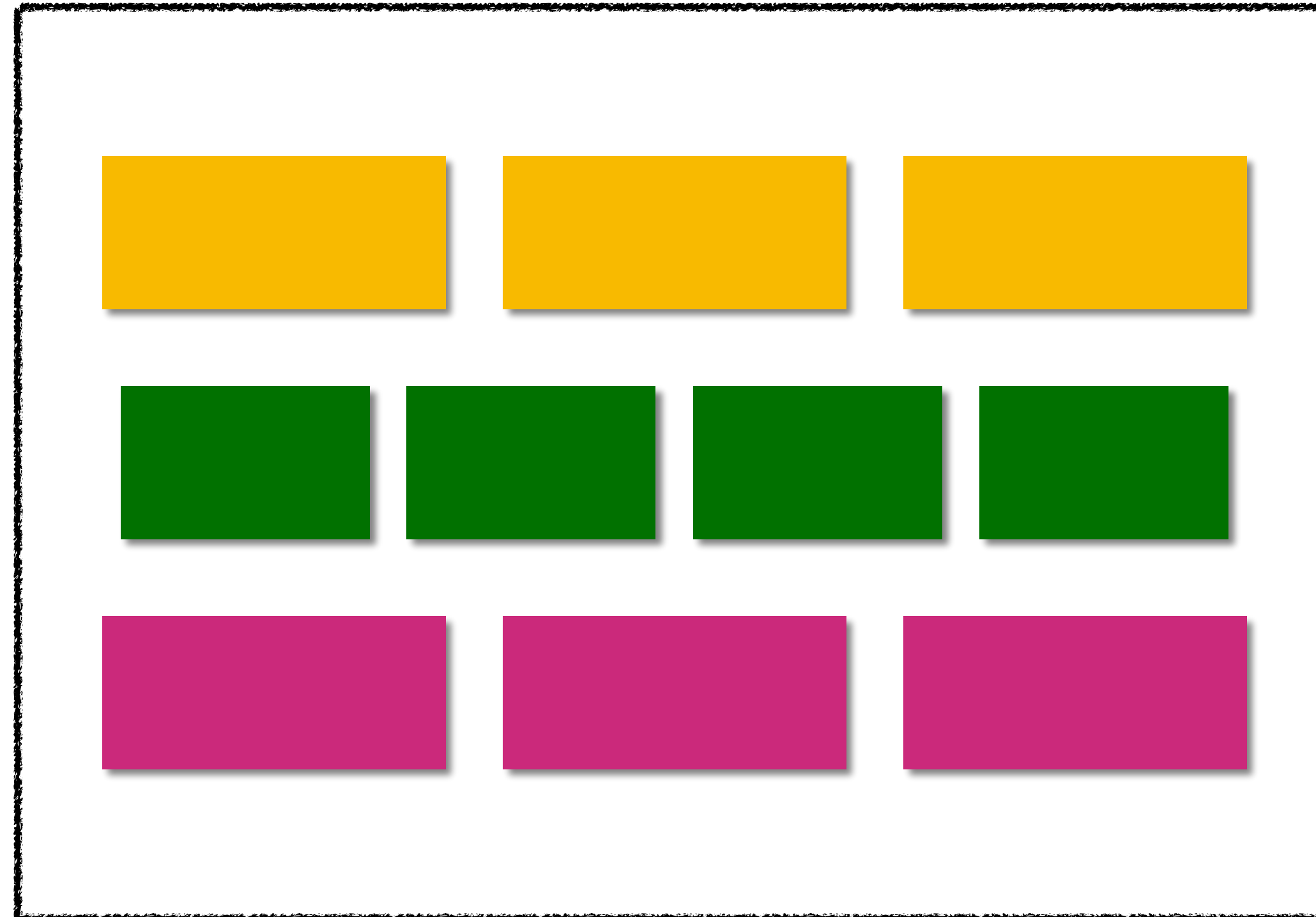
Everything in Rust



Traditional Software. Made up by libs

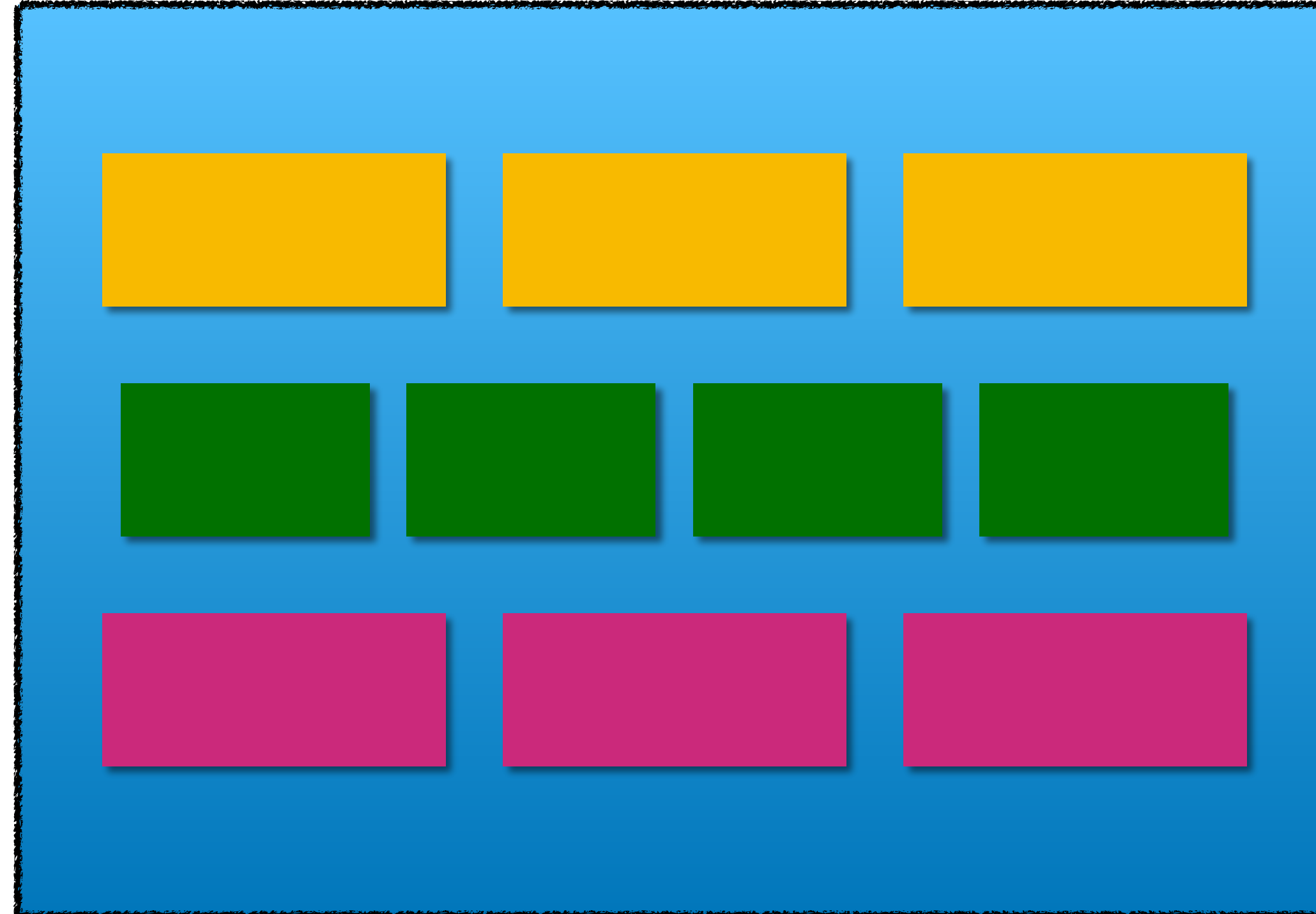


Traditional Software. Made up by libs

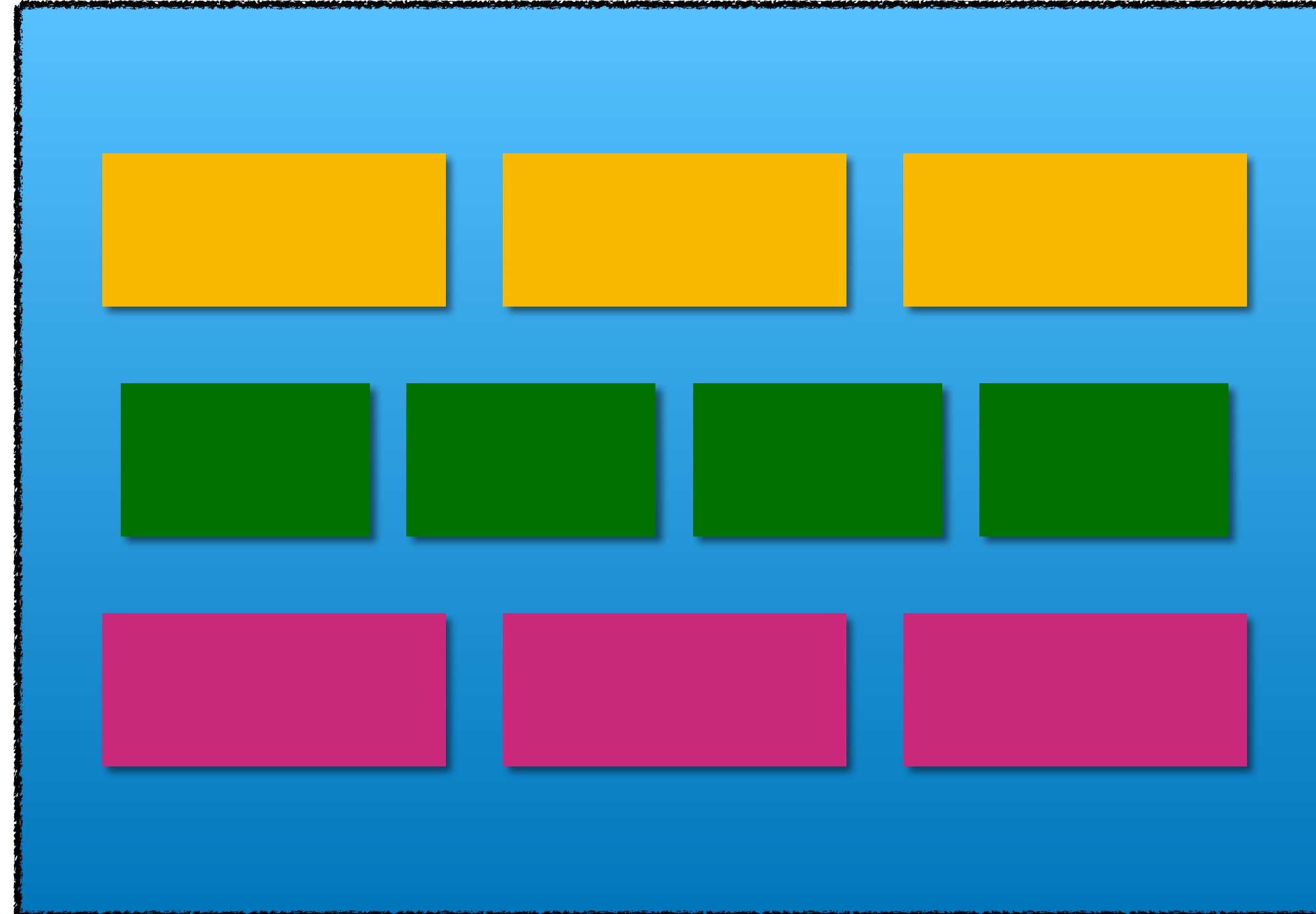


ABI between static/dynamic libs

Software in Rust: mods/crates

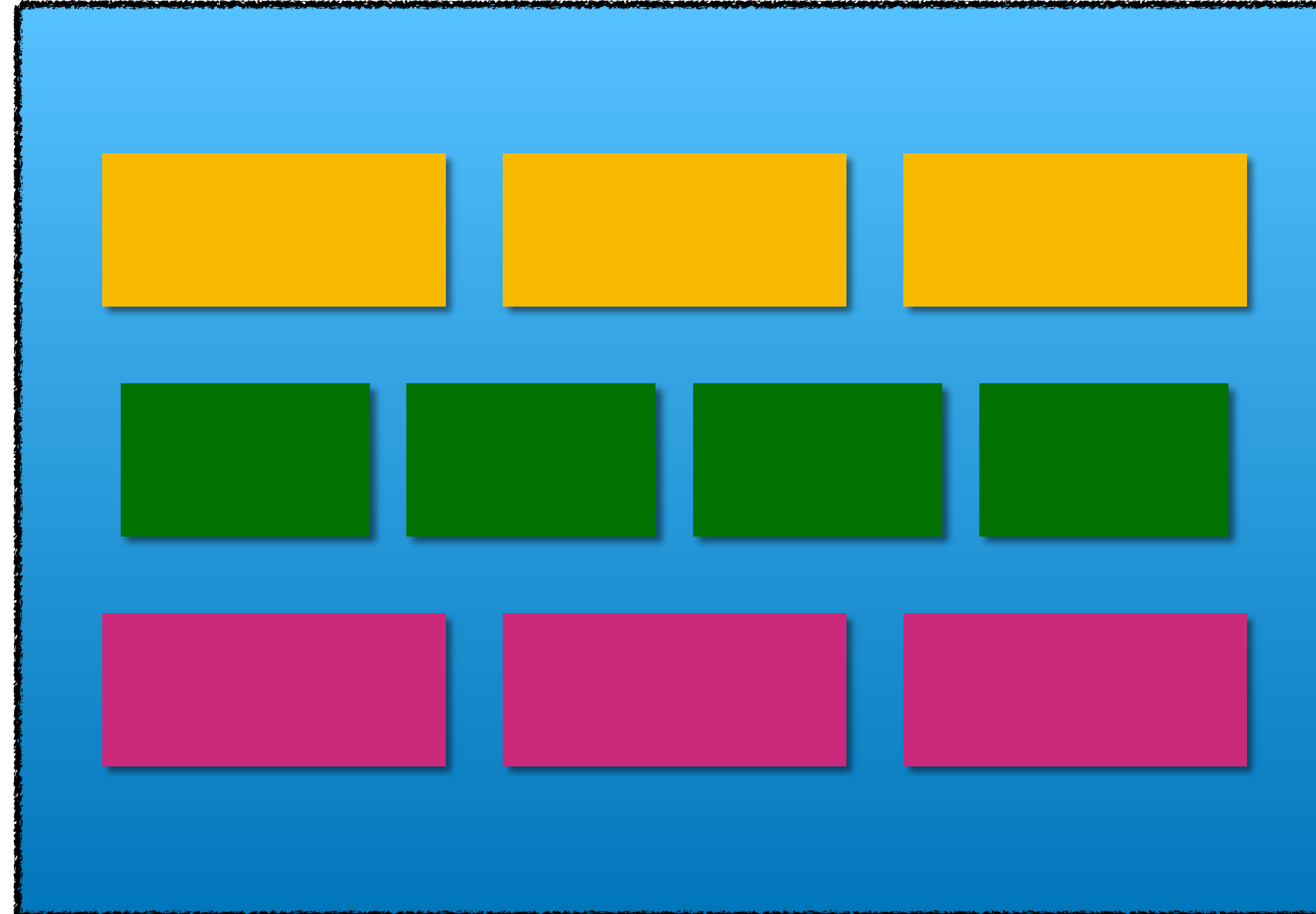


Software in Rust: mods/crates



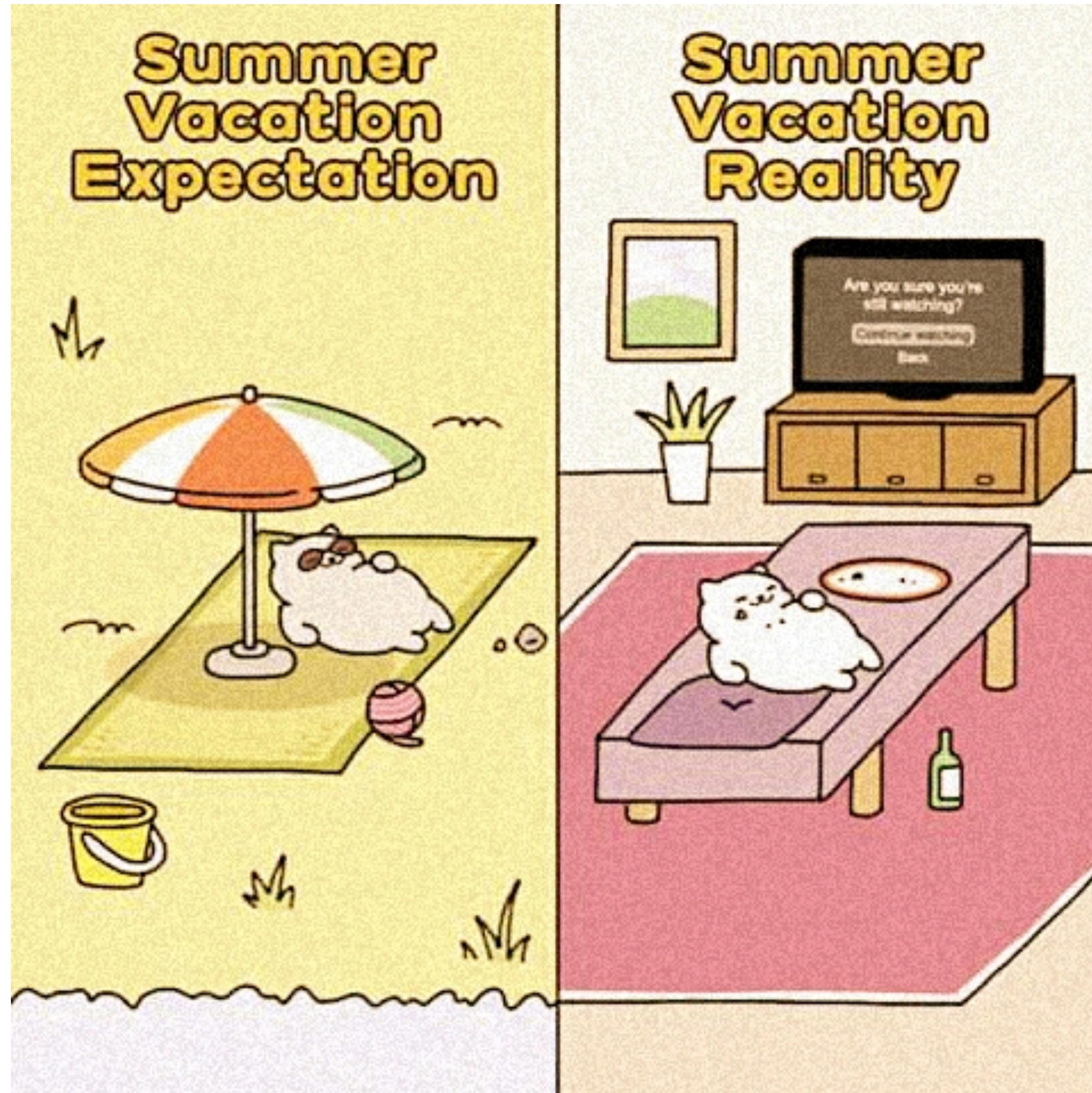
Type/Borrow Check everywhere!

Software in Rust: mods/crates



Type/Borrow Check everywhere!

Expectation vs Reality



People prefer bindings

openssl-sys crates.io v0.9.39
FFI bindings to OpenSSL

Repository

All-Time: 3,038,926
Recent: 509,348

openssl crates.io v0.10.15
OpenSSL bindings

Repository

All-Time: 2,569,961
Recent: 480,363

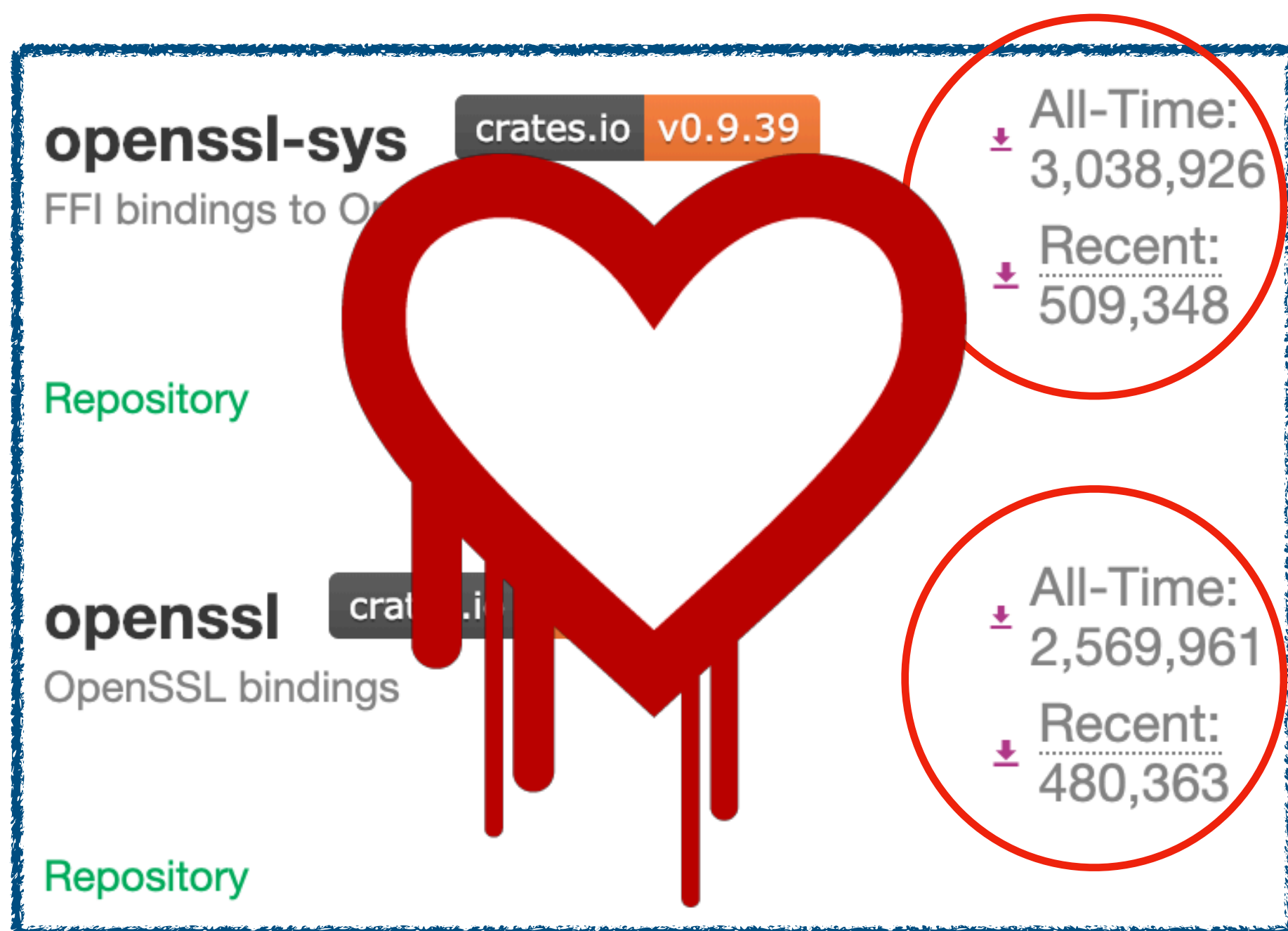
rustls crates.io v0.14.0

Rustls is a modern TLS library written in Rust.

All-Time: 212,971
Recent: 60,311

Homepage Repository

People prefer bindings



The screenshot shows two crates on crates.io. The top crate is 'openssl-sys' (v0.9.39) with 3,038,926 all-time downloads and 509,348 recent downloads. The bottom crate is 'openssl' with 2,569,961 all-time downloads and 480,363 recent downloads. A large red heart is drawn over the crates, with red lines dripping down from its base, symbolizing preference for bindings.

Crane Name	Version	All-Time Downloads	Recent Downloads
openssl-sys	v0.9.39	3,038,926	509,348
openssl		2,569,961	480,363



The screenshot shows the 'rustls' crate (v0.14.0) with 212,971 all-time downloads and 60,311 recent downloads. The description states: 'Rustls is a modern TLS library written in Rust.' Links for 'Homepage' and 'Repository' are visible at the bottom.

Crane Name	Version	All-Time Downloads	Recent Downloads
rustls	v0.14.0	212,971	60,311

New attacks against TLS again!



David Wong

@cryptodavidw

关注



Just released a bunch of attacks on 7 TLS implementations making use of good ol' Bleichenbacher as well as Manger's attack (on pkcs#1 v1.5!) also includes a TLS 1.3 downgrade attack. With [@eyalr0](#), Gillham, Genkin, Shamir and [@yuvalyarom](#) buff.ly/2rcElc5

翻译推文

上午8:02 - 2018年11月30日

39 转推 69 喜欢



1



39



69



So what?



Brian Smith

@BRIAN_____

关注



Nice!

Note that these attacks fundamentally don't work in [@jpixton](#)'s Rustls because it doesn't allow the prerequisite downgrade to RSA encryption, because it doesn't implement RSA encryption at all. Any **ring**-based implementation would be immune to this for the same reason.



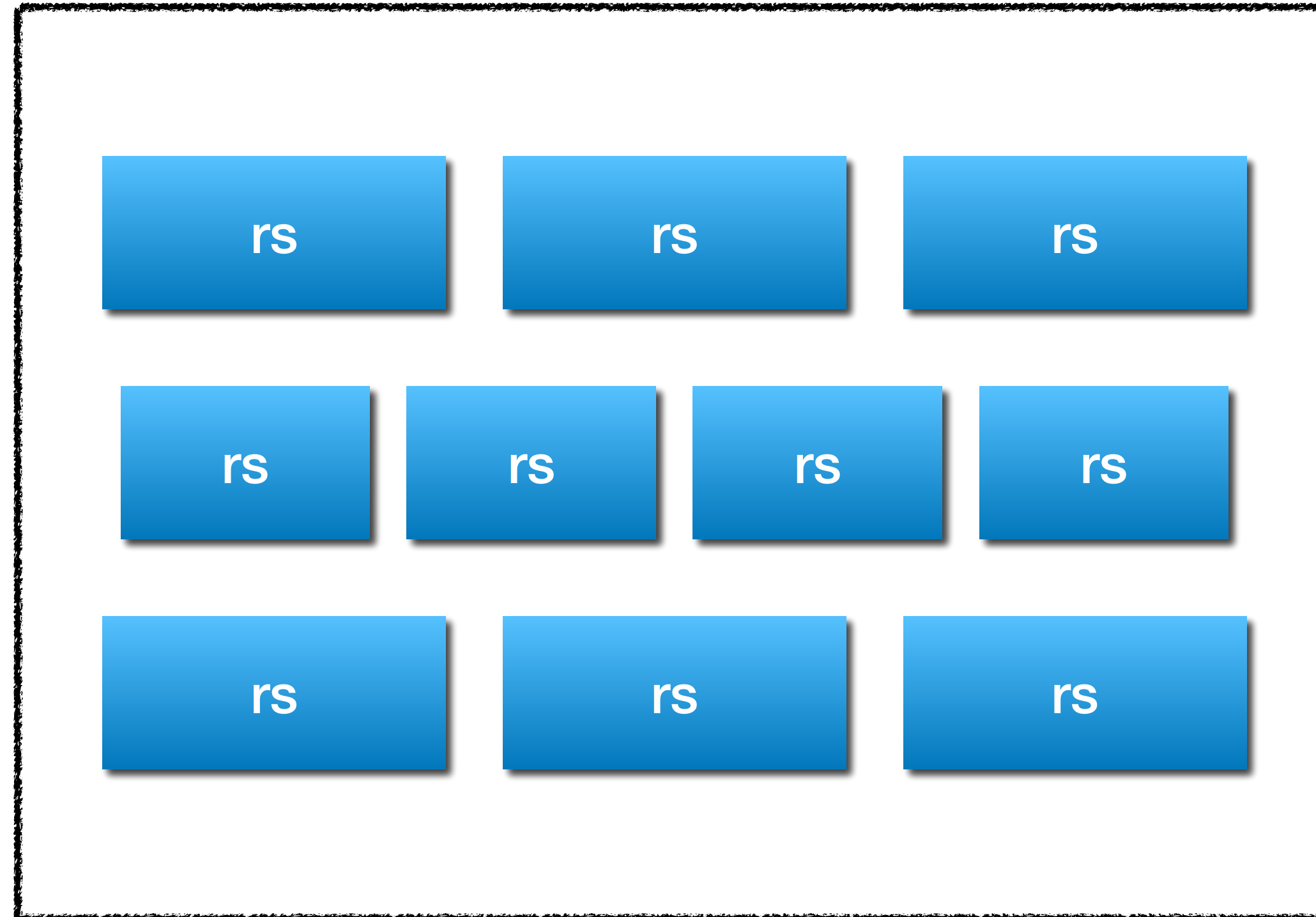
Eyal Ronen @eyalr0

"The 9 Lives of Bleichenbacher's CAT:New Cache Attacks on TLS Implementations ", with Robert Gillham, Daniel Genkin, Adi Shamir, @cryptodavidw and @yuvalyarom is now available at cat.eyalro.net

🌐 翻译推文

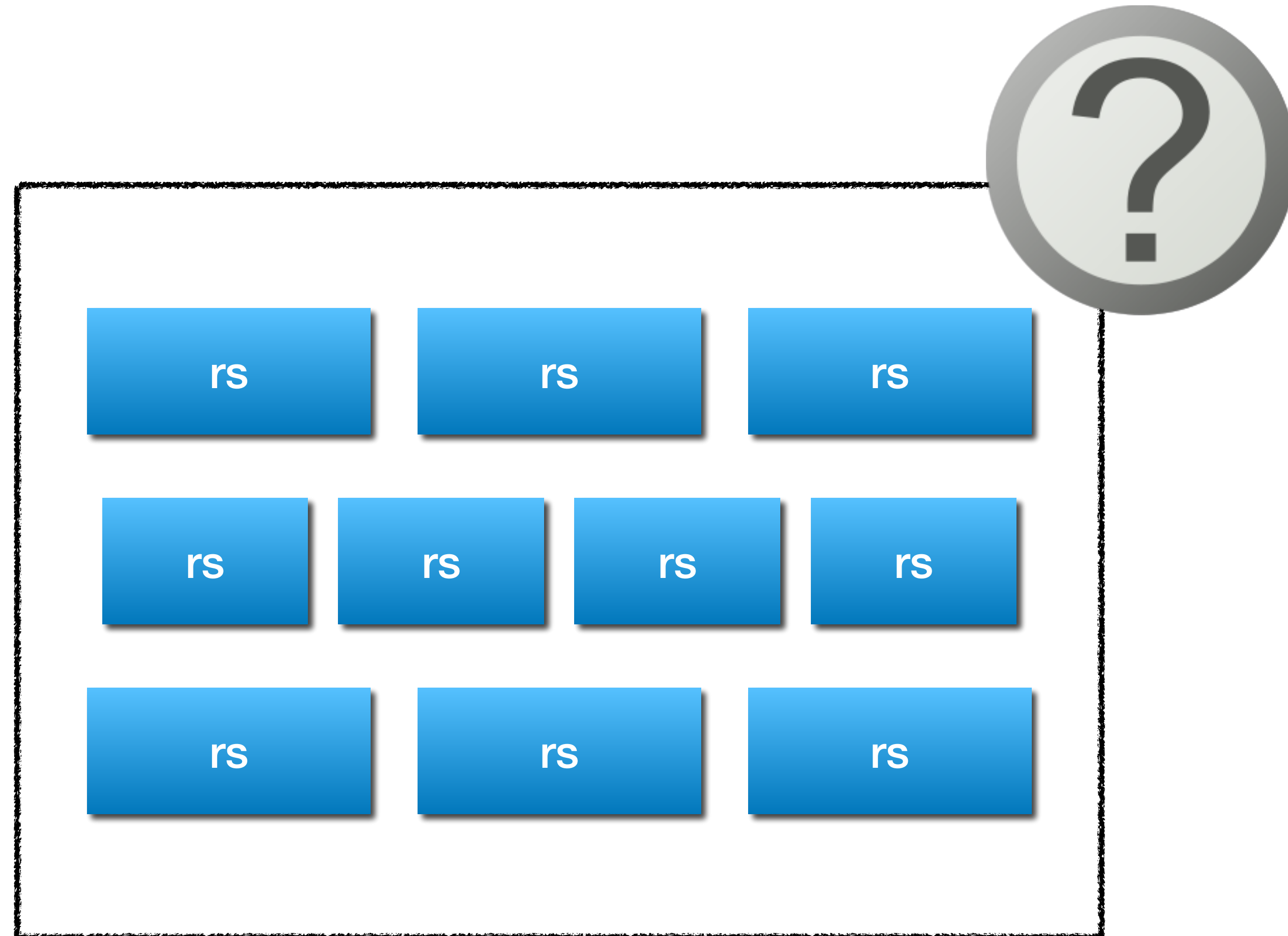
下午12:07 - 2018年11月30日

Our Strategy



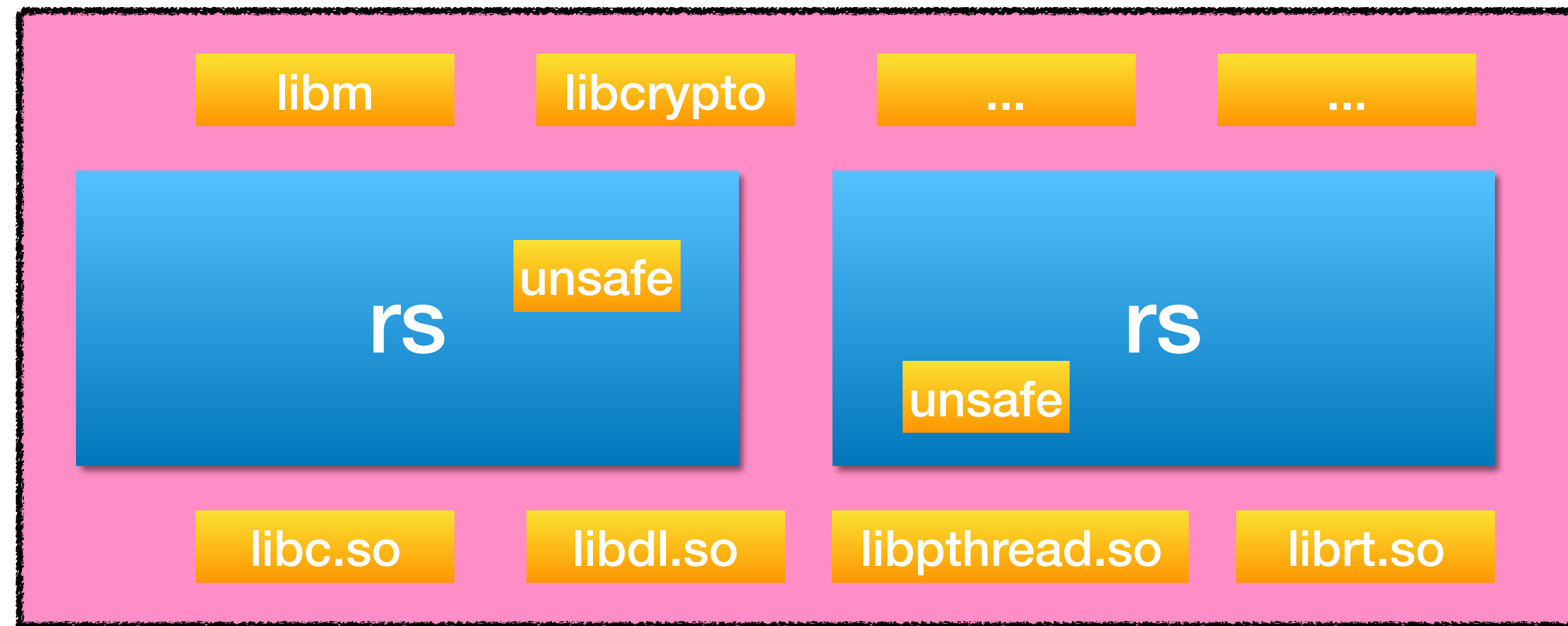
ABI between static/dynamic libs

Our Strategy



ABI between static/dynamic libs

Take a closer look



- **unsafe code**
- **unsafe library**
- **unsafe interface**

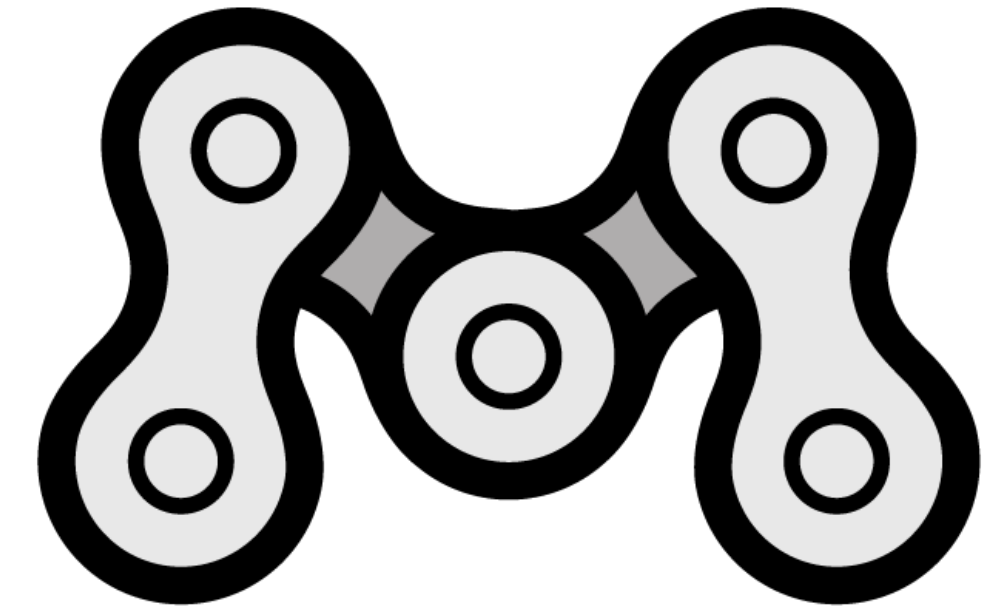
Take a closer look

- unsafe code
 - Categorize unsafe codes
 - Manual code audit / Unit tests / Fuzz
- unsafe library
 - Formal verification
- unsafe interface
 - Dynamic checking

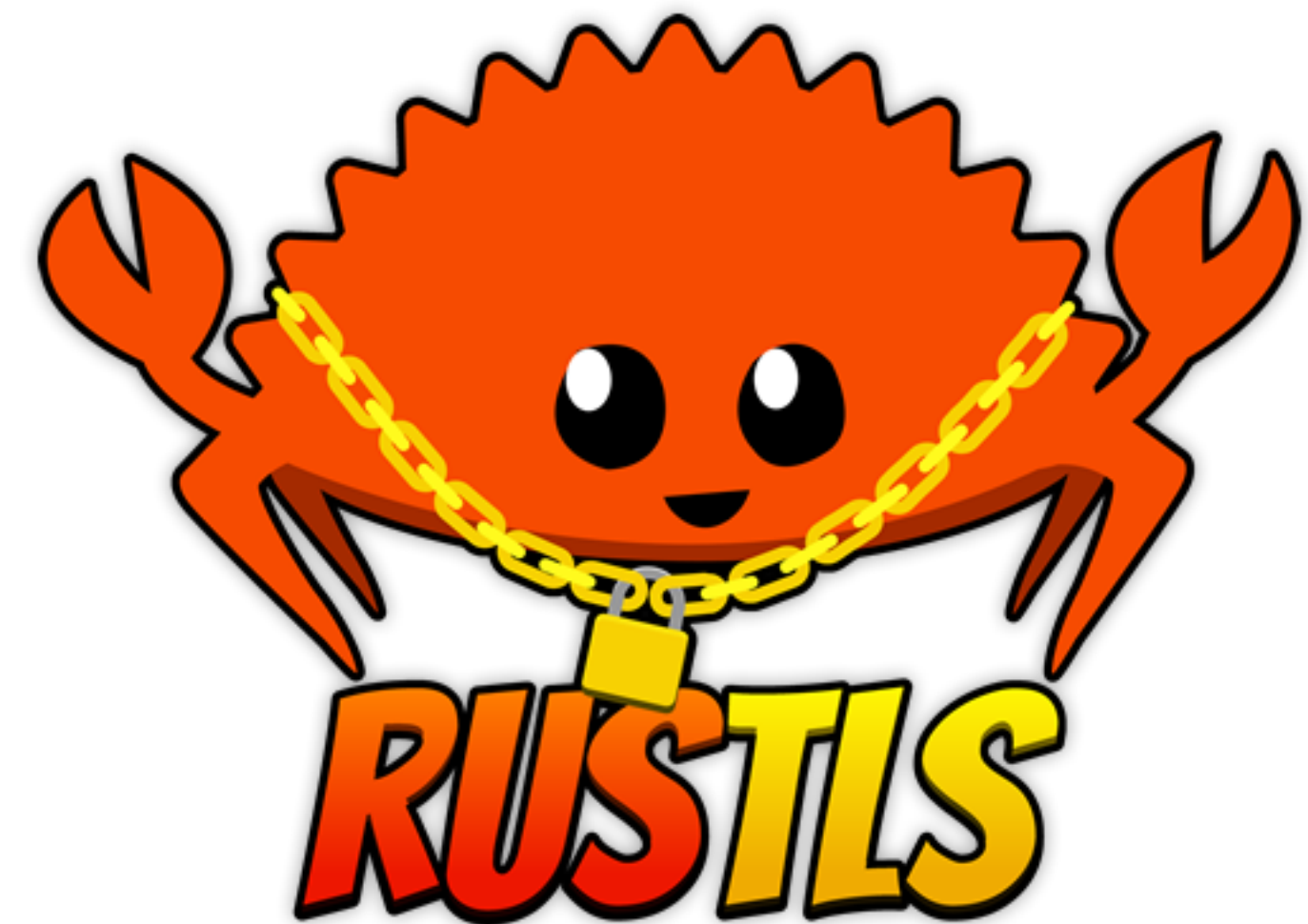
Mesalink as an example

- C bindings of rustls/ring/webpki

```
#include "mesalink.h"  
#include <mesalink/openssl/ssl.h>  
#include <mesalink/openssl/err.h>
```



MesaLink



Compare with Openssl binding

	Mesalink	OpenSSL binding
Unsafe Func	2/2	29/29
Unsafe Expr	94/210	4398/4398
Unsafe Impl	0/0	29/29
Unsafe Trait	0/0	3/3
Unsafe Method	0/0	13/13

*Result generated using `cargo geiger`

Achievement

- **4,500,000** monthly active users and rapidly growing
- Github stars
 - rustls
 - BoringSSL
 - MesaLink

★ Star	816
★ Star	481
★ Star	846



Daniel Stenberg

@bagder

关注

For the first time since 2012, someone is adding a new TLS backend to curl: mesalink. An OpenSSL-compatible library written in rust.

翻译推文



vtls: add a MesaLink vtls backend by kevinis · Pull Request ...

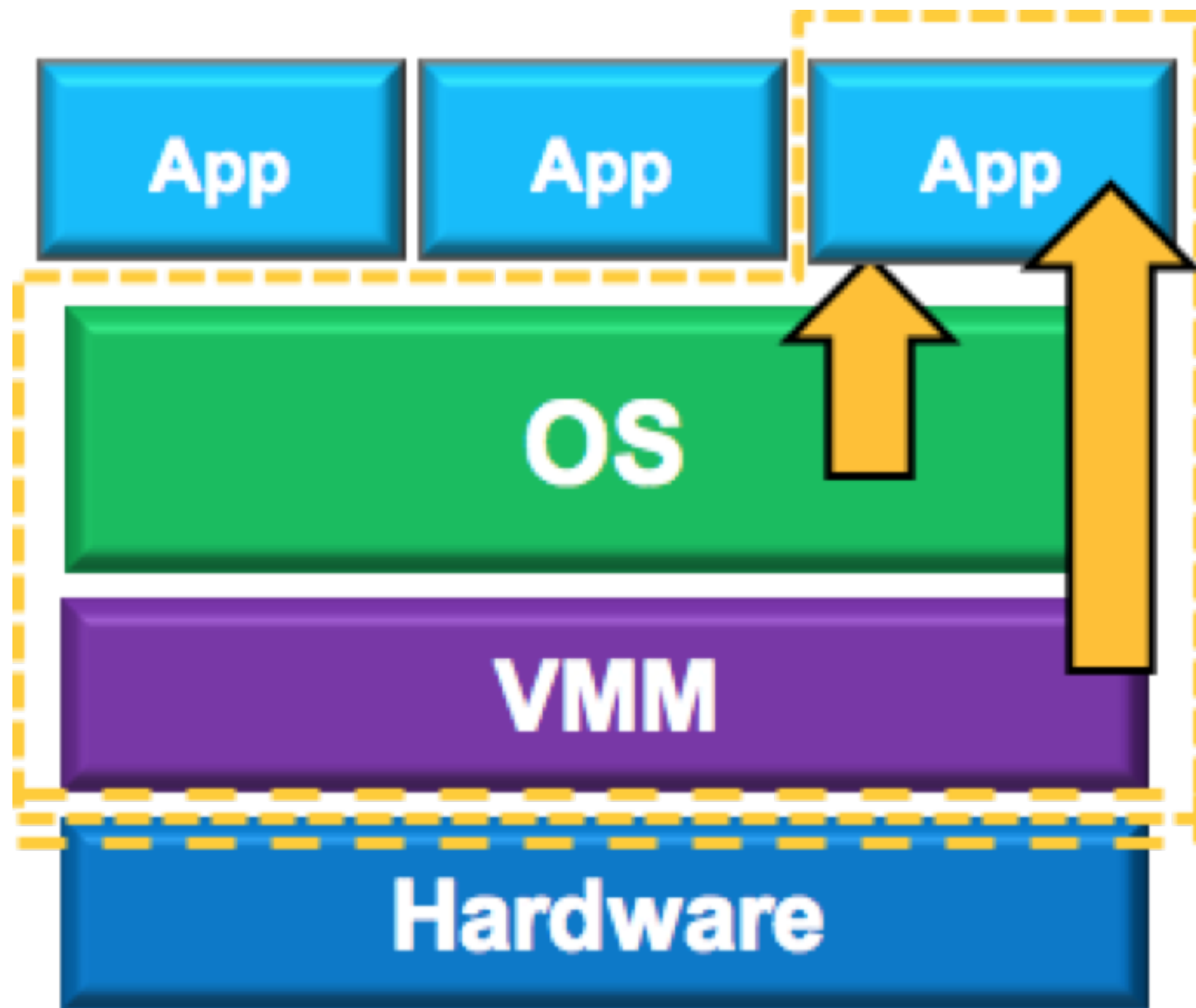
MesaLink is a TLS library written in 100% Rust, a programming language that guarantees memory safety. This PR adds MesaLink as a vtls backend for curl.

github.com

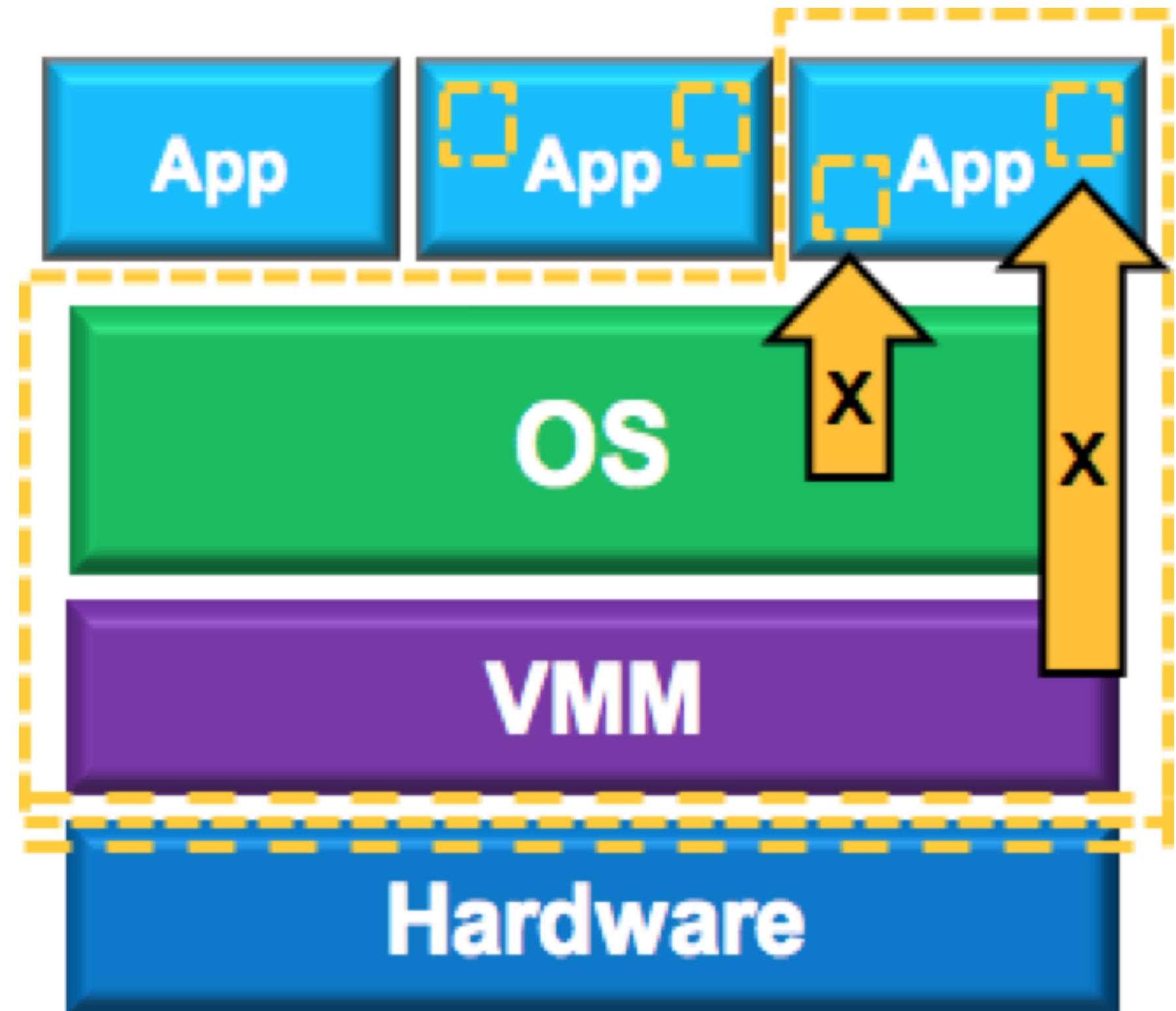
上午10:40 - 2018年8月14日

BUT IT ISN'T ENOUGH!

Rust + SGX



Without SGX



SGX Enforced

Intel SGX Enclave

Stack Overflow

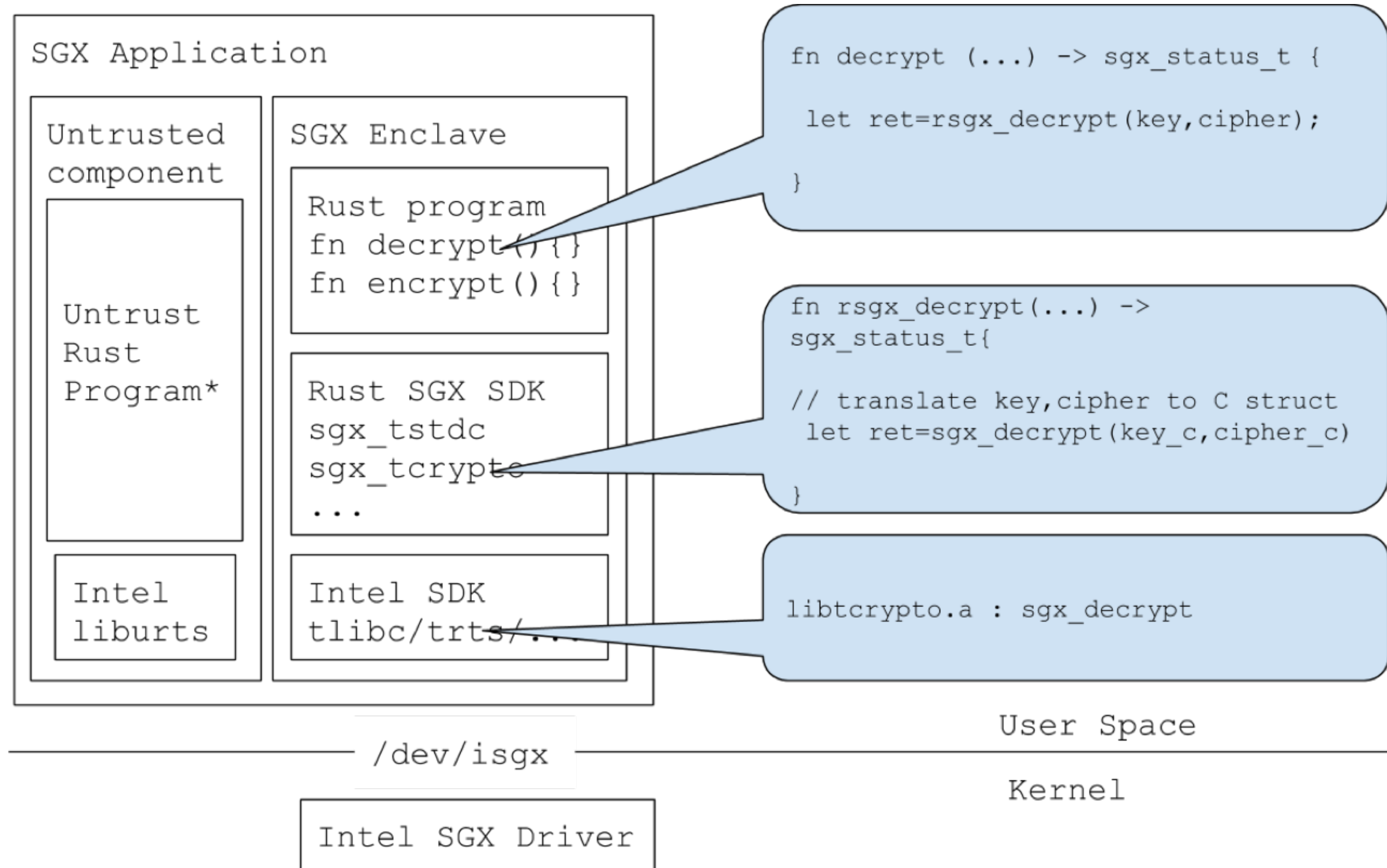
Heap Overflow

Use-After-Free

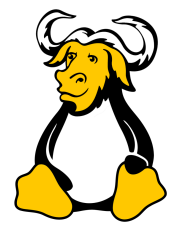
Double Free

SECRET

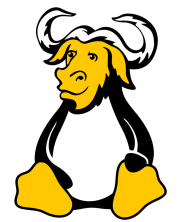
Intel SGX API? Easy!



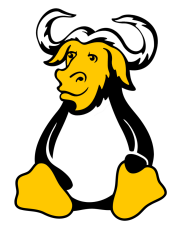
SGX features in Rust SGX



untrusted_fs vs **sgx_file** 



untrusted_time vs **time** 



net, env

- Use on demand

- `std::untrusted::file` 

- `std::fs::SgxFile` 

Exist Structs	Rust Sgx Structs
<code>std::fs::File</code>	<code>sgx_tstd::fs::SgxFile</code>
<code>std::thread::Thread</code>	<code>sgx_tstd::thread::SgxThread</code>
<code>std::thread::ThreadId</code>	<code>sgx_tstd::thread::SgxThreadId</code>
<code>std::sync::Mutex</code>	<code>sgx_tstd::sync::SgxMutex</code>
<code>std::sync::MutexGuard</code>	<code>sgx_tstd::sync::SgxMutexGuard</code>
<code>std::sync::Condvar</code>	<code>sgx_tstd::sync::SgxCondvar</code>
<code>std::sync::RwLock</code>	<code>sgx_tstd::sync::SgxRwLock</code>
<code>std::sync::RwLockReadGuard</code>	<code>sgx_tstd::sync::SgxRwLockReadGuard</code>
<code>std::sync::RwLockwriteGuard</code>	<code>sgx_tstd::sync::SgxRwLockwriteGuard</code>

Rust SGX SDK vs. fortanix-sgx

	Rust SGX SDK	fortanix-sgx
形态	独立的 Rust Crate 无需修改编译器	集成在 Rust libstd 中 需要修改 rustc 编译器
是否依赖于 Intel SGX 套件	是 基本无修改	部分 包含不安全的各类实现 以及大量对 SGX PSW 的修改
是否享受 Intel SGX 的功能	是。可以直接使用protected_fs, PCL, switchless, remote attestation 等支持	否 每个功能需要 Fortanix 再开发
是否可直接使用现有的 Rust crate	是 移植简单	否 缺乏许多基本功能

Achievements

- Recommended by Intel

Supported Languages

Enclave binding interface is supported in C and C++ only.

To develop Intel SGX enclaves in the Rust* programming language, use the Rust SGX SDK in [GitHub*](#).

- RustFest '18 Talk (acc ratio = 11%)



kev @kevinwatters · 5月26日

even more #rustfest - @dingelish hints at a possible (and possibly dystopic imo?) future: secure multiparty computing with intel-provided hardware enclaves providing encrypted memory access. his team built a version of rust's stdlib for Intel **SGX**, impressive!

- Multiple PR merged into Intel's SDK

- Community

↑ mapofcanada rust 8 points · 3 months ago · edited 3 months ago

↓ Code within an SGX enclave can't call out to the operating system, so Baidu has done the heroic effort of re-implementing half of the entire Rust std library. Really appreciate all the hard work you've put into this.

- Used in Blockchain



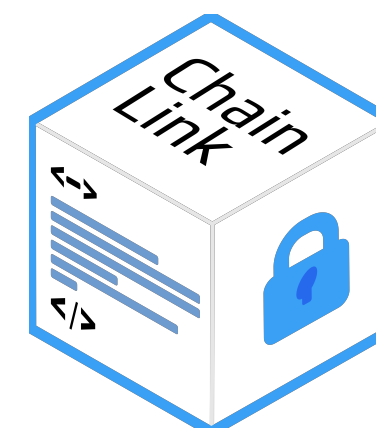
Enigma from MIT, 1st round \$30 Million

```
[target.'cfg(not(target_env = "sgx"))'.dependencies.sgxtstd]  
git = "https://github.com/baidu/rust-sgx-sdk.git"  
rev = "v1.0.0"
```



Ekiden from UC Berkeley, 1st round \$45 Million

```
[dependencies]  
token-api = { path = "./api", features = ["sgx"], default-feat  
sgxtstd = { git = "https://github.com/ekiden/rust-sgx-sdk" }
```

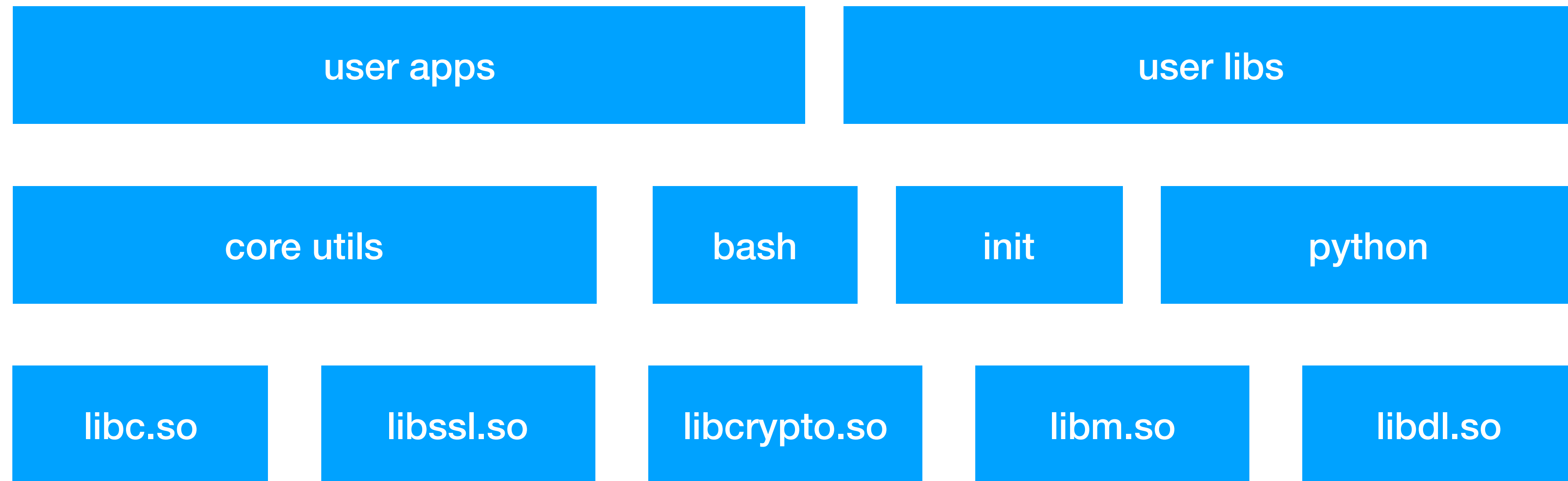


ChainLink, 1st round \$32 Million

```
[target.'cfg(not(target_env = "sgx"))'.dependencies]  
sgxtstd = { path = "/opt/rust-sgx-sdk/sgxtstd" }  
sgxtypes = { path = "/opt/rust-sgx-sdk/sgxtypes" }
```

BUT IT ISN'T ENOUGH!

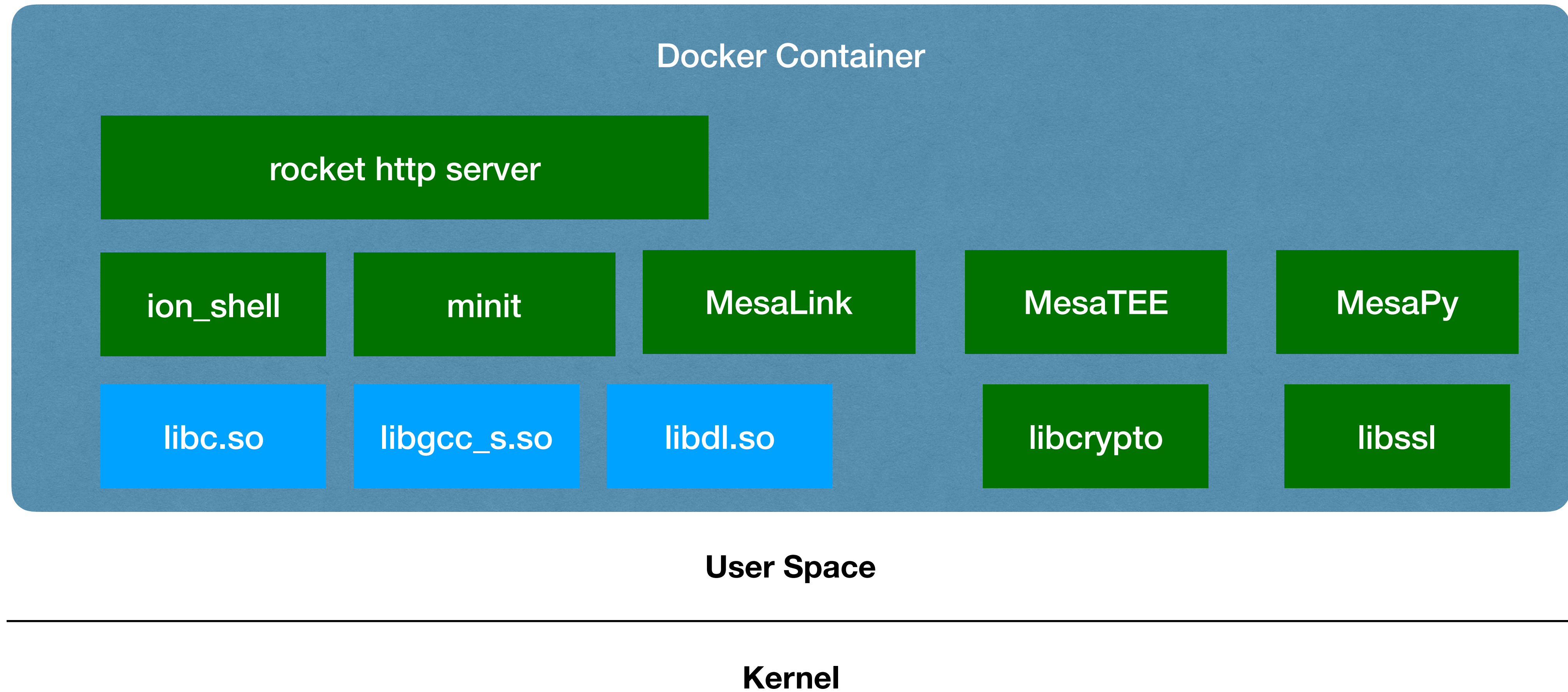
Memory Safe Linux User space



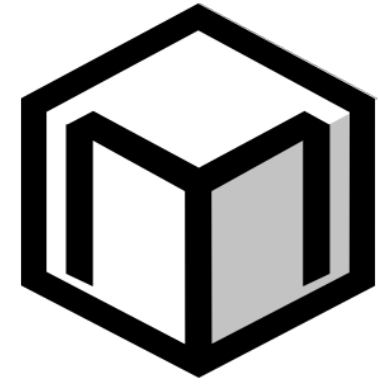
User Space

Kernel

Memory Safe Linux User space



The MeSa Family



MesaTEE



RUST-SGX



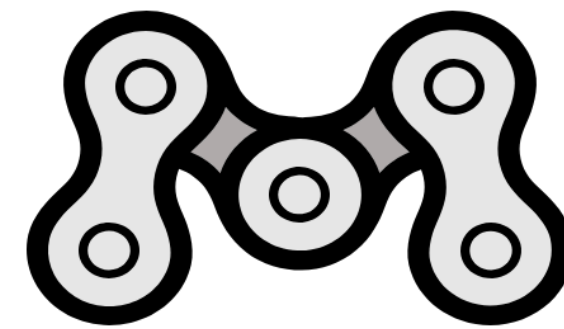
OASP



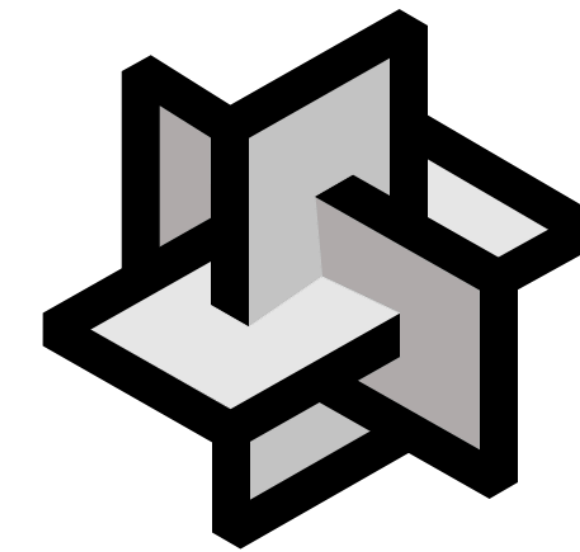
MesaArmor



MesaPy



MesaLink

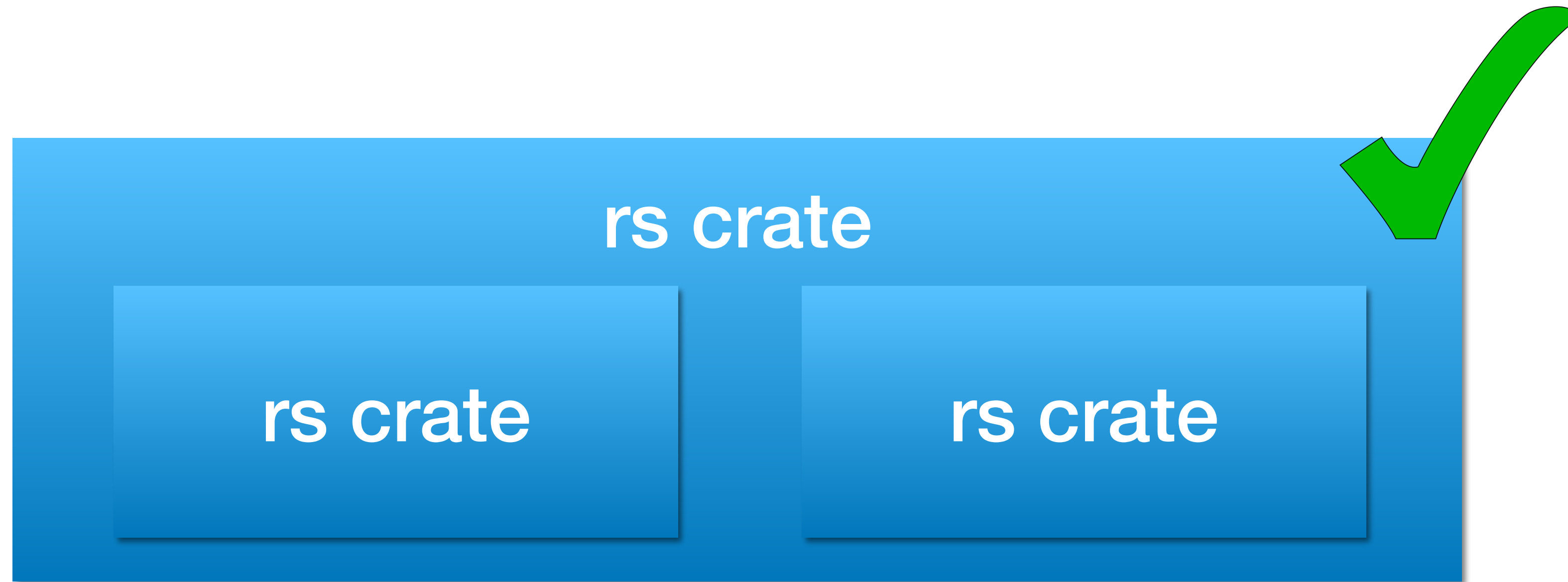


MesaLock

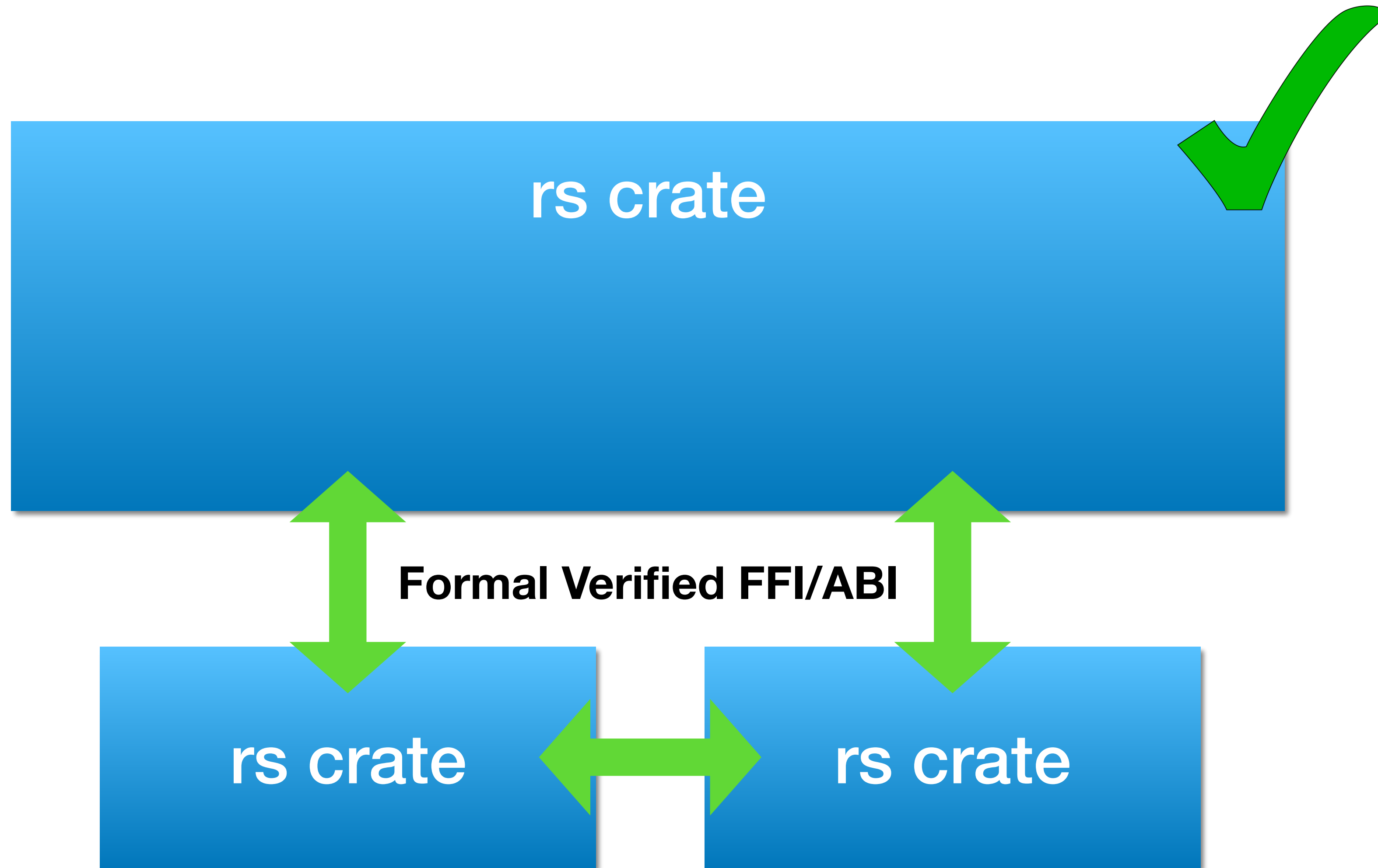
Summary

- To achieve Memory Safety in the **real world**
 - From lib to lib
 - With support of dev community and companies
 - Follow "hybrid memory safety rules-of-thumb"
 - Apply Non-bypassable Security Paradigm (NbSP)

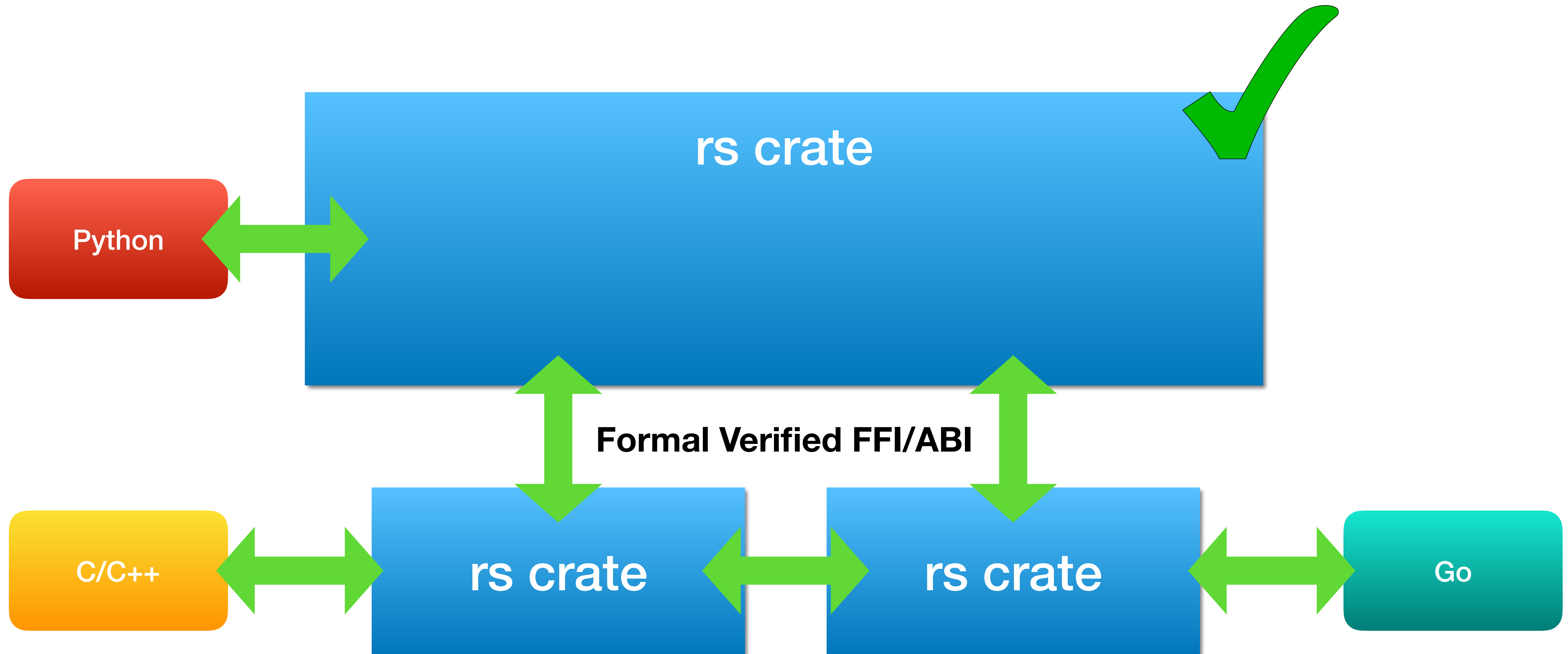
Open Question



Open Question: MesaFFI



Open Question: MesaFFI



A background image showing a close-up of hands writing on papers. The image is faded and serves as a backdrop for the text.

THANKS

Q&A

dingelish@gmail.com

dingyu02@baidu.com