开放原子开源基金会
OPENATOM FOUNDATION

RustyVault
一个用Rust编写的Vault替代

Tongsuo Project Team

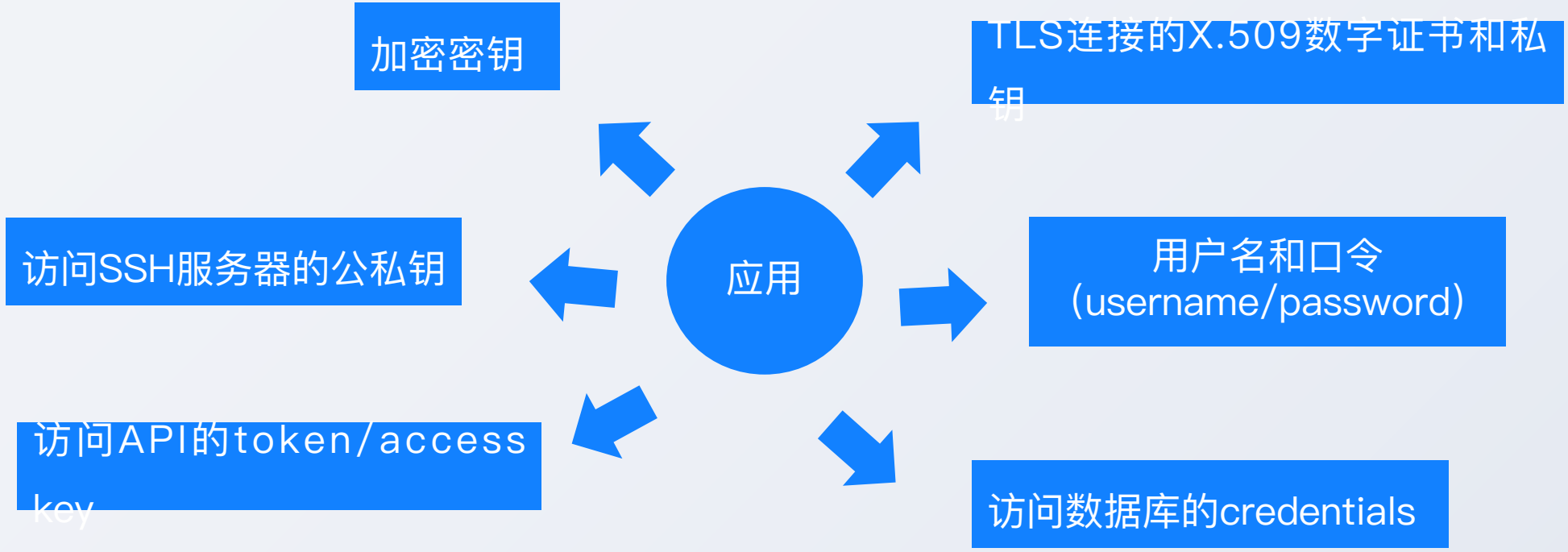
铜锁密码学开源社区简介

铜锁密码学开源社区，全称开放原子铜锁，是一个关于密码学的开源社区/项目群（20+仓库），核心产品是：

- 铜锁密码学算法库（软件密码模块）
- RustyVault

什么是Secrets：从Key到Secrets

我们将云原生应用在其生命周期中所需的各种敏感信息统称为Secret，其包括但不限于以下内容：



Secrets Sprawl问题

简单来讲，就是对于各类的secret，采用了很随意的方式来使用，比如直接在代码中硬编码token的值，或者是在某个配置文件中记录了用户名和口令，或者是私钥明文存储在服务器的磁盘上等等，造成了一些问题：

- 容易泄露
- 无法管理（不了解机密信息使用情况和泄露情况）

Secrets Sprawl问题的根源是没有使用统一的管理系统，这个问题并不是只有在云计算时代才出现的，而是一直伴随着IT行业的发展，只不过云计算时代的到来加速了上述问题的发生，因为：

- 每个应用程序对外部API的调用急剧增大（云计算导致协作变快）
- 每个应用程序的公网暴露程度加大（云计算对应用程序形态的改变）
- 组织里应用程序的数量急剧增大（信息化程度上升、开发门槛降低）

TongSUO 铜锁

云原生Secrets管理

综上所述，结合云原生资源的动态使用和Secret类型的多样化等特点，我们期望存在一种新的解法来满足这些要求。对于上述Secrets Sprawl问题，可行的解决方式就是使用“集中化”/“中心化”的机密信息管理系统，即Secret Management System。这里提到的“集中化”/“中心化”，是指逻辑上的概念，而不是物理部署的实际状态。

具体来说，为了解决Secrets Sprawl问题，我们对所需的Secret Management System提出了如下的必要功能：

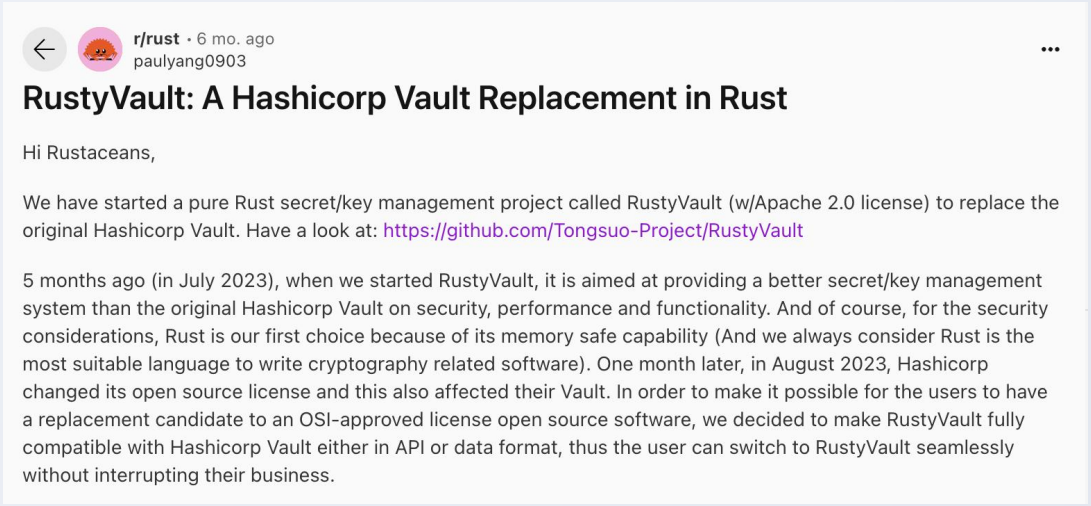
1. 该系统是以安全存储为核心，所有的secrets在落盘、传输和使用的过程中应该是受保护的
2. 该系统的重要功能之一是身份的认证和鉴权，即可以将某个用户（人或者服务）的身份和其所拥有的secrets进行关联
3. 该系统的重要功能之一是日志和审计能力，可以使其保存的Secrets数据有据可查、有迹可循
4. 该系统的重要功能之一是和云计算厂商的关联，可以帮助用户实现对使用云计算服务的自动化权限管理
5. 该系统的部署尽可能灵活，可以基于硬件设备实现更高的安全等级（如使用加密卡存储主密钥、在TEE中运行加解密算法等）；也可以在纯软的环境中提供受限的安全能力（比如Web3的客户端节点上）

Hashicorp Vault

- Golang开发
- 开源版本和企业版本
- HCP服务
- 开源许可证：BSL
- Hashicorp被IBM收购
- 分叉版本和竞品
  - Linux Foundation OpenBao

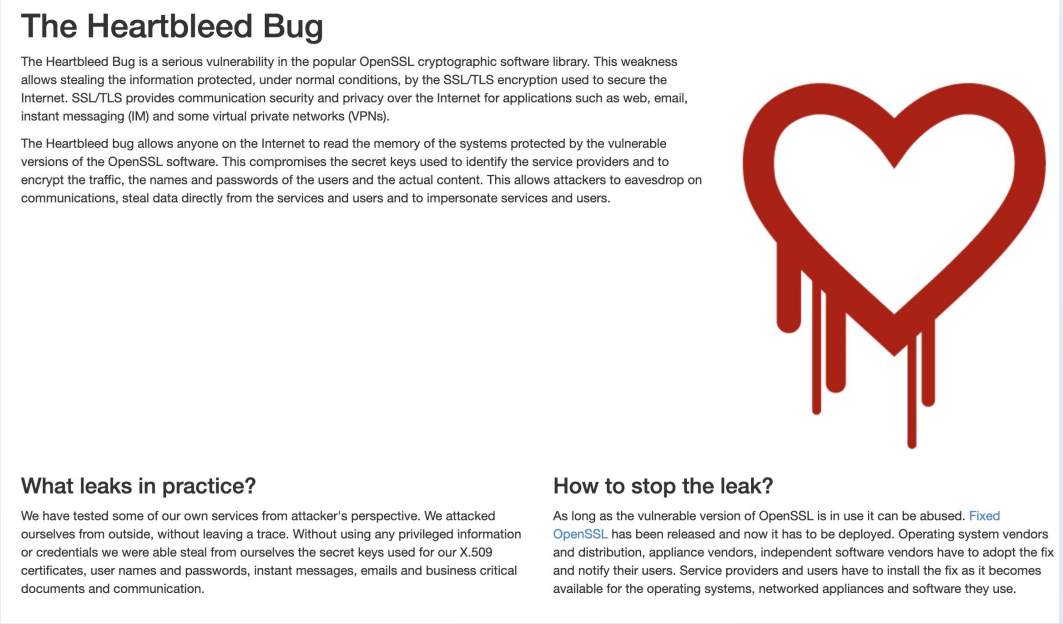
创造RustyVault的动机

- Hashicorp Vault的缺点
  - 部分功能仅企业版支持：如集群
  - 性能一般
  - 开源License风险
  - Hashicorp 曾禁止 Vault 企业版在中国售卖（2020）
  - 做为密码产品无法做到国内合规
- 铜锁社区于2023.7启动RustyVault项目
  - API全面兼容\存储结构和配置文件全面兼容
  - Apache 2.0 License
  - 性能更高
  - 国产和密码合规
  - Vault企业版特性支持



为什么是Rust

- RustyVault是一个安全产品
  - 自身首先要安全
- 内存安全
- 密码学算法性能
  - 计算密集型
  - 监管合规
  - 集成铜锁（C库）
  - Rust FFI vs cgo
- 降本增效
  - Google: Rust 2x C++ productive



The Heartbleed Bug

The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet. SSL/TLS provides communication security and privacy over the Internet for applications such as web, email, instant messaging (IM) and some virtual private networks (VPNs).

The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users and to impersonate services and users.

What leaks in practice?

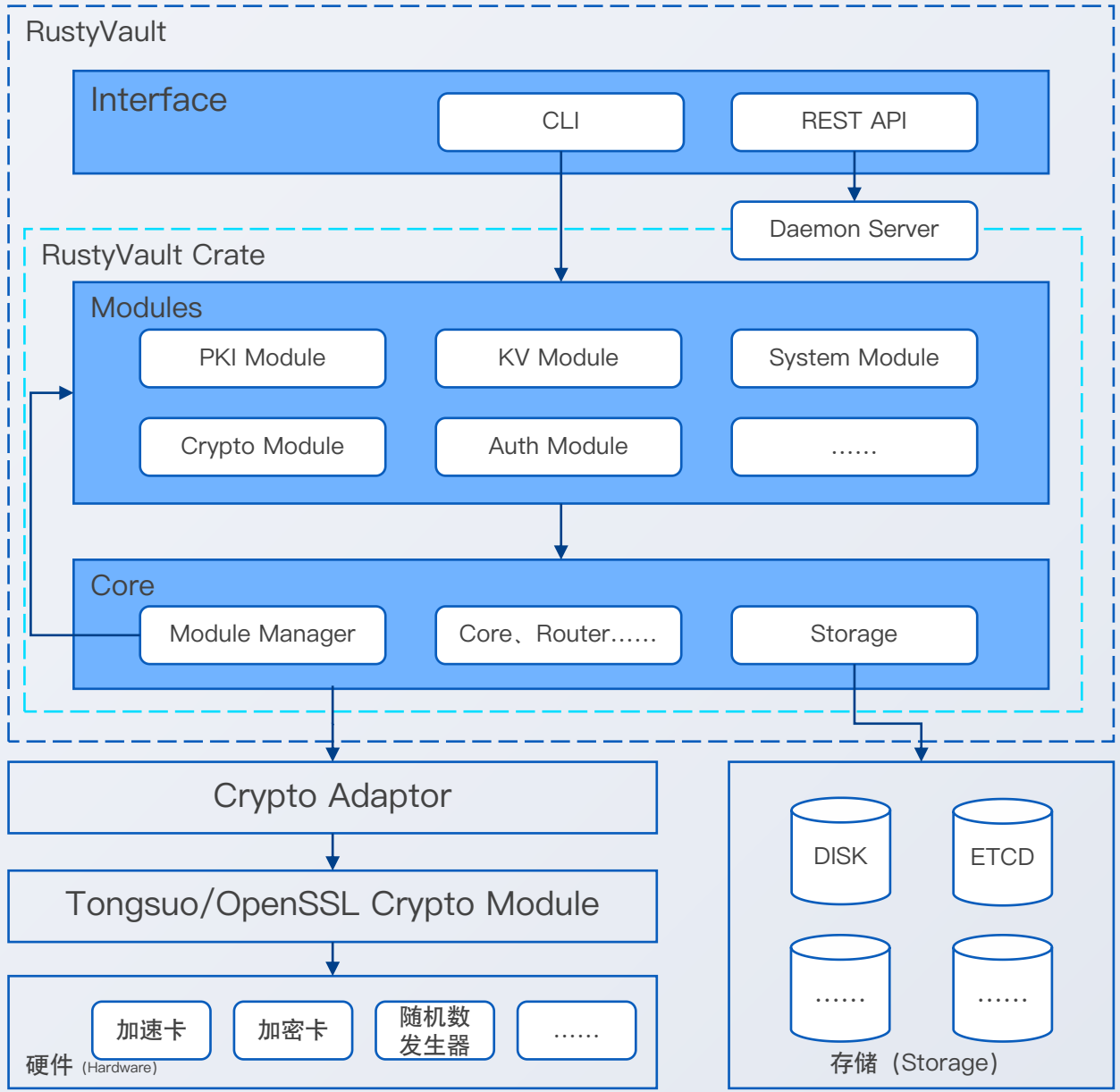
We have tested some of our own services from attacker's perspective. We attacked ourselves from outside, without leaving a trace. Without using any privileged information or credentials we were able steal from ourselves the secret keys used for our X.509 certificates, user names and passwords, instant messages, emails and business critical documents and communication.

How to stop the leak?

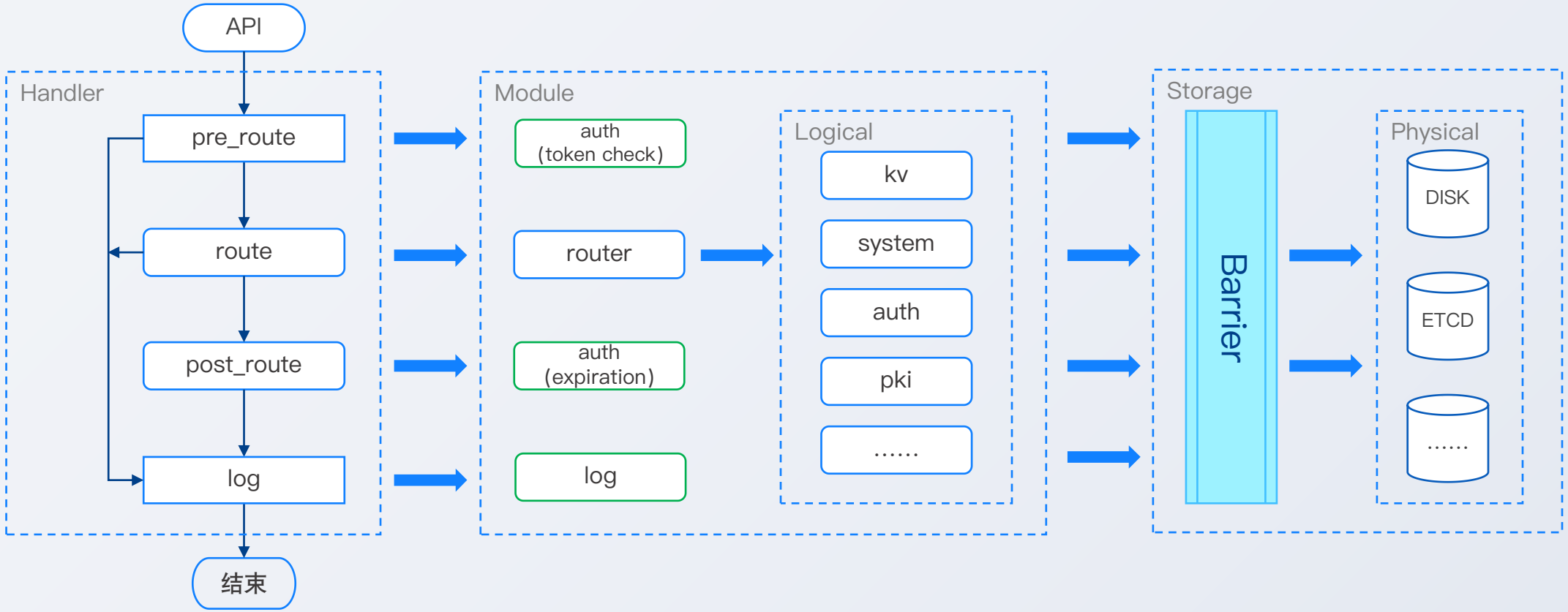
As long as the vulnerable version of OpenSSL is in use it can be abused. Fixed OpenSSL has been released and now it has to be deployed. Operating system vendors and distribution, appliance vendors, independent software vendors have to adopt the fix and notify their users. Service providers and users have to install the fix as it becomes available for the operating systems, networked appliances and software they use.

RustyVault–架构设计

- 两种形态
  - 独立进程，RESTful API
  - Rust crate的库形态
- 可切换的底层密码模块



RustyVault–架构设计



RustyVault–功能特性

密码学功能：

- 密钥管理
  - 生成/存储/轮转
- 公钥基础设施（PKI）
  - X.509 证书签发：RSA/ECC/SM2
  - X.509 证书吊销：OCSP，CRL
- 传统密码学算法
  - 对称加密算法：AES/SM4
  - 非对称加密算法：
    - 签名算法：RSA/ECDSA/SM2/环签名
    - 加密算法：RSA/SM2
  - 哈希算法：SHA1/SHA2/SM3
  - 随机数生成：国密合规的随机数生成算法
- 密态计算和隐私计算
  - 零知识证明：Bulletproofs
  - 半同态加密：Paillier/EC-ElGamal

软件基础功能：

- AuthN & AuthZ
  - 用户身份认证：证书/用户名口令
  - 访问规则控制
- 配置方式
  - 配置文件
  - 支持热加载
- API
  - Rest API/CLI/crate
- 存储能力
  - 本地加密存储
  - ETCD 等中心化存储
- 集群和高可用
  - 全主模式（多读多写）
- 日志和监控：文件日志/Prometheus
- 硬件适配
  - 通过指令集、加速卡等进行密码算法性能优化
  - 通过加密卡、加密机等进行密钥保护和密钥管理

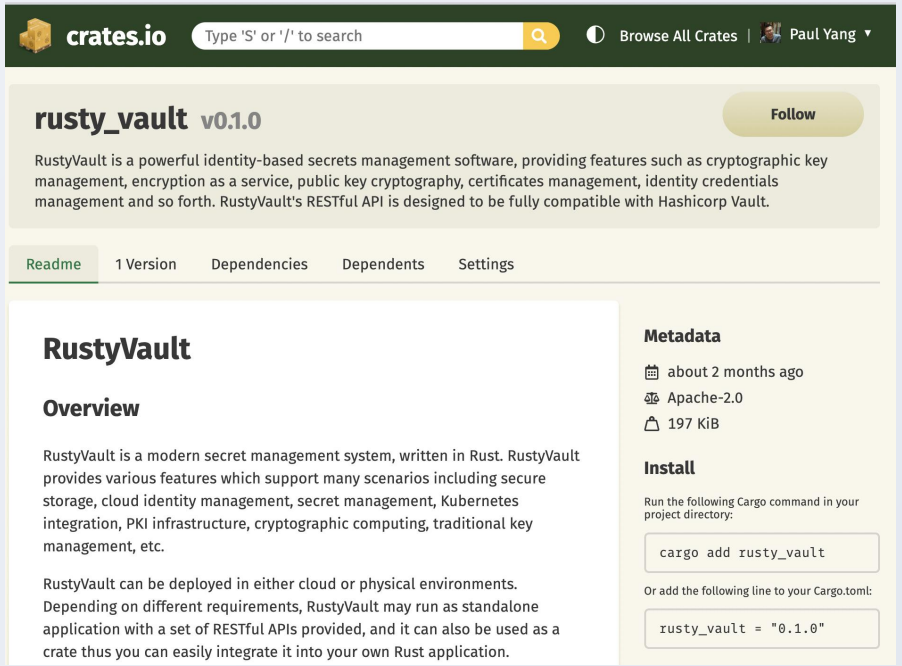
功能亮点：可切换密码模块

- 统一密码学操作抽象层
- 高性能密码学算法实现
- 实现不同国家和地区的监管要求
  - 信创
  - 密评
  - 等保
- 支持底层密码硬件
  - 加密卡
  - 加密机
  - 加密芯片
  - CPU指令集

Tongsuo 铜锁 功能亮点：库形态使用

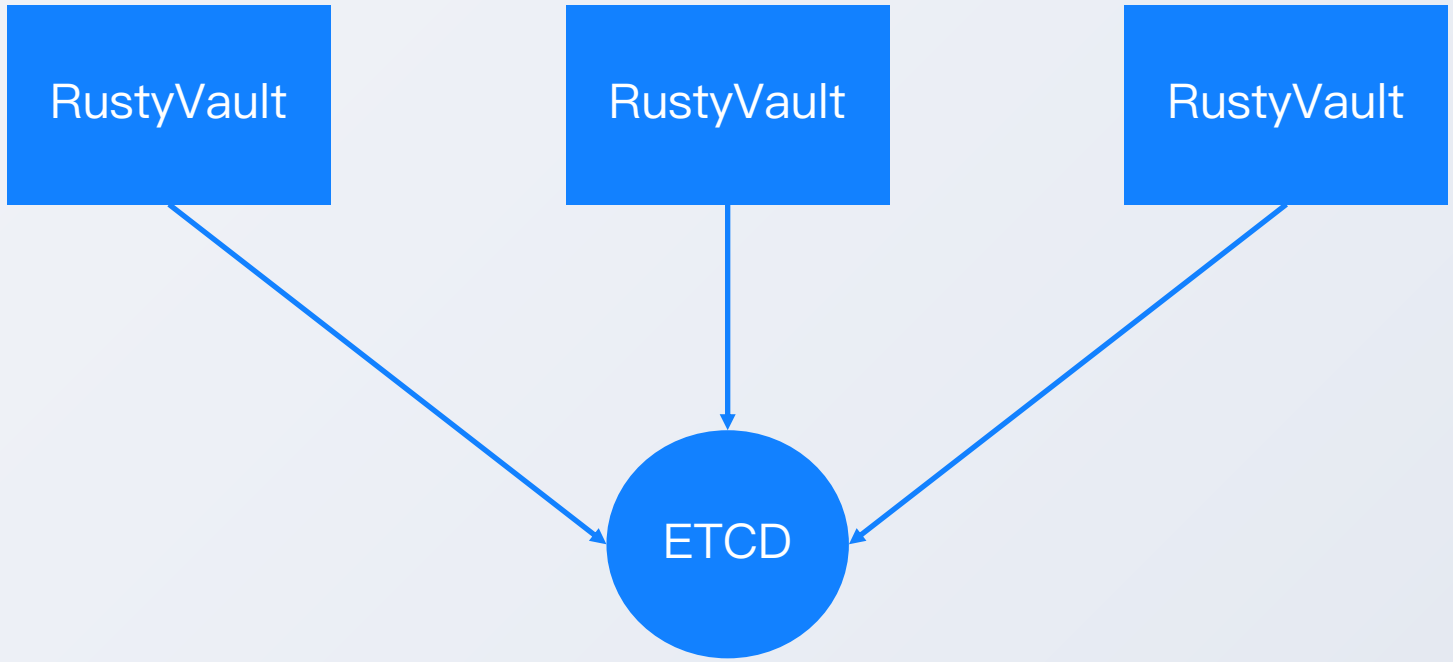
- 做为Rust的crate
- 已经发布到crates.io
- 可以编译到应用程序中提供机密信息管理能力
- W3IF之Mega项目
  - 管理加密密钥，实现数据加密

W3IF



功能亮点：集群和高可用

- 主主模式（Active – Active）
- 提升性能
- 容灾
- 无Load Balancer模式



TongSUO 铜锁

RustyVault–使用示例

配置文件

- 兼容 vault 的 hcl 格式
- json 格式

```
# jinjiu @ jinjiu in ~/github/RustyVault on git:pki x [10:14:11]
$ cat config.hcl
storage "file" {
    path    = "./data"
}

listener "tcp" {
    address     = "127.0.0.1:8200"
    tls_disable = "false"
}

daemon = true
daemon_user = "jinjiu"
daemon_group = "staff"

work_dir = "/Users/jinjiu/work/rusty_vault/"

api_addr = "http://127.0.0.1:8200"
log_level = "debug"
pid_file = "rusty_vault.pid"
```

TONG SUO 铜锁

RustyVault–使用示例

启动进程和初始化

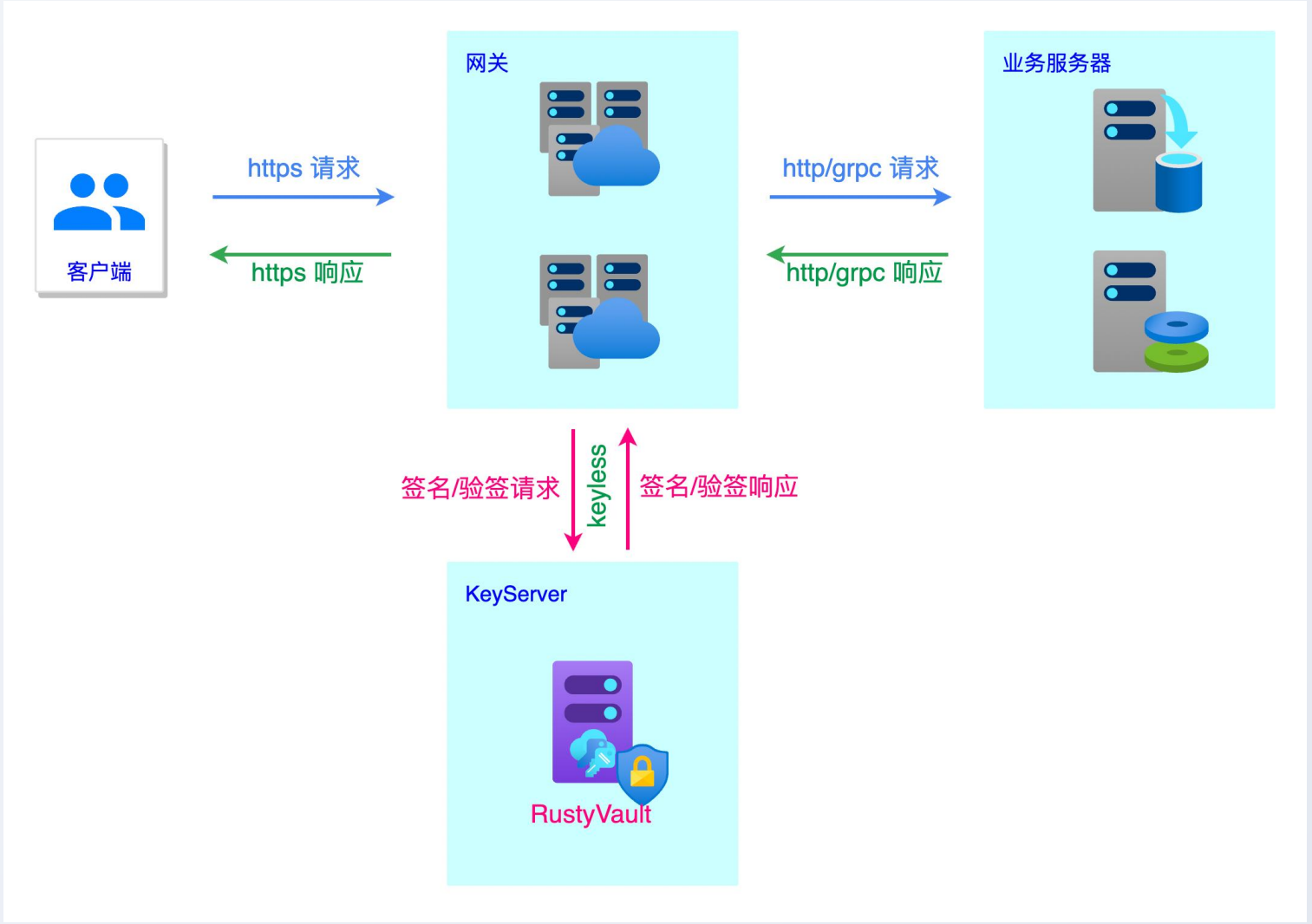
- sys/init：初始化
- sys/seal：密封
- sys/unseal：解封

```
# jinjiu @ jinjiu in ~/github/RustyVault on git:pki x [10:15:25]
$ rm -rf /Users/jinjiu/work/rusty_vault/

# jinjiu @ jinjiu in ~/github/RustyVault on git:pki x [10:15:48]
$ ./target/debug/rvault server --config ./config.hcl
 2023-11-25T02:16:11Z INFO  rusty_vault::cli::command::server  create work_dir: /Users/jinjiu/work/rusty_vault/

# jinjiu @ jinjiu in ~/github/RustyVault on git:pki x [10:16:12]
$ tree /Users/jinjiu/work/rusty_vault/
/Users/jinjiu/work/rusty_vault/
├── rusty_vault.log
└── rusty_vault.pid

0 directories, 2 files

# jinjiu @ jinjiu in ~/github/RustyVault on git:pki x [10:16:44]
$ curl 'http://localhost:8200/v1/sys/init' -X PUT --data '{"secret_shares":3,"secret_threshold":2}'

{"keys":["05ce1abc1f913de5407c86869bb298e5645748e01bdfd14c7ac43c05c4bc204b01","76ac6aa13dff2dc1f95fb2bfb931b2fb559d24140fe7183ac3cef06bd4
6b61c502","ae7bb3aa232cd4dd67b757a8a7b95df1b3db00b103065fe15dc8b4b82d265ebf03"],"root_token":"15f13741-b137-fcd2-2978-c212984e186a"}%

# jinjiu @ jinjiu in ~/github/RustyVault on git:pki x [10:17:19]
$ curl 'http://localhost:8200/v1/sys/unseal' -X PUT --data '{"key":"05ce1abc1f913de5407c86869bb298e5645748e01bdfd14c7ac43c05c4bc204b01"}'
{"sealed":true,"t":3,"n":2,"progress":1}%

# jinjiu @ jinjiu in ~/github/RustyVault on git:pki x [10:18:36]
$ curl 'http://localhost:8200/v1/sys/unseal' -X PUT --data '{"key":"ae7bb3aa232cd4dd67b757a8a7b95df1b3db00b103065fe15dc8b4b82d265ebf03"}'
{"sealed":false,"t":3,"n":2,"progress":0}%
```

RustyVault-典型场景（密码即服务）

- 传统网关问题
  - 扩容堆机器
  - 私钥分散、明文落盘

- RustyVault-keyless 方案
  - ✓ 低成本
  - ✓ 高性能
  - ✓ 高安全

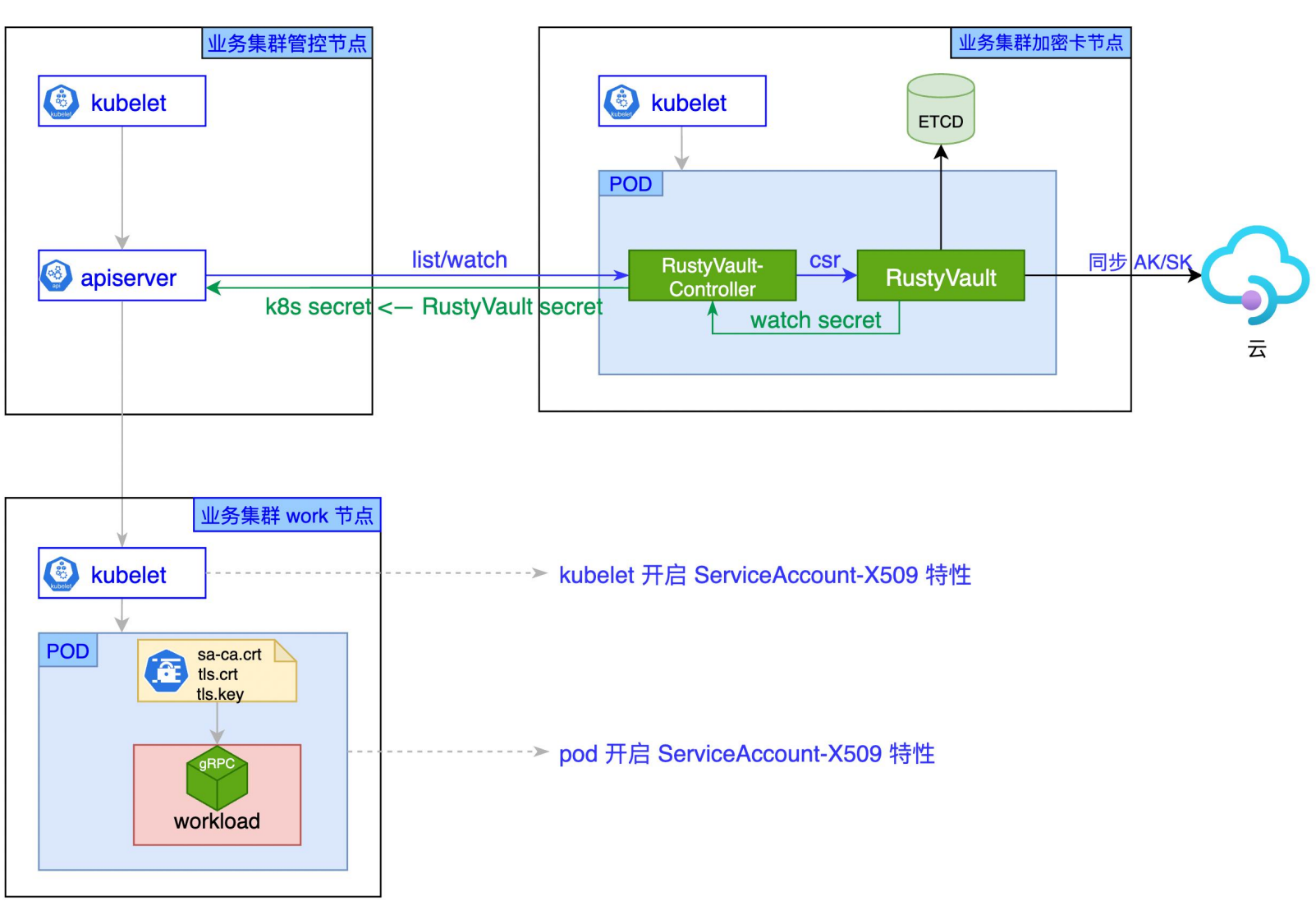


RustyVault–典型场景（K8s Pod身份）

- K8S 证书体系问题
  - 私钥明文落盘
  - 无吊销机制
  - 无 Pod 身份证书
- K8S secret 使用问题
  - secret 线下泄漏风险
  - 开发效率低下

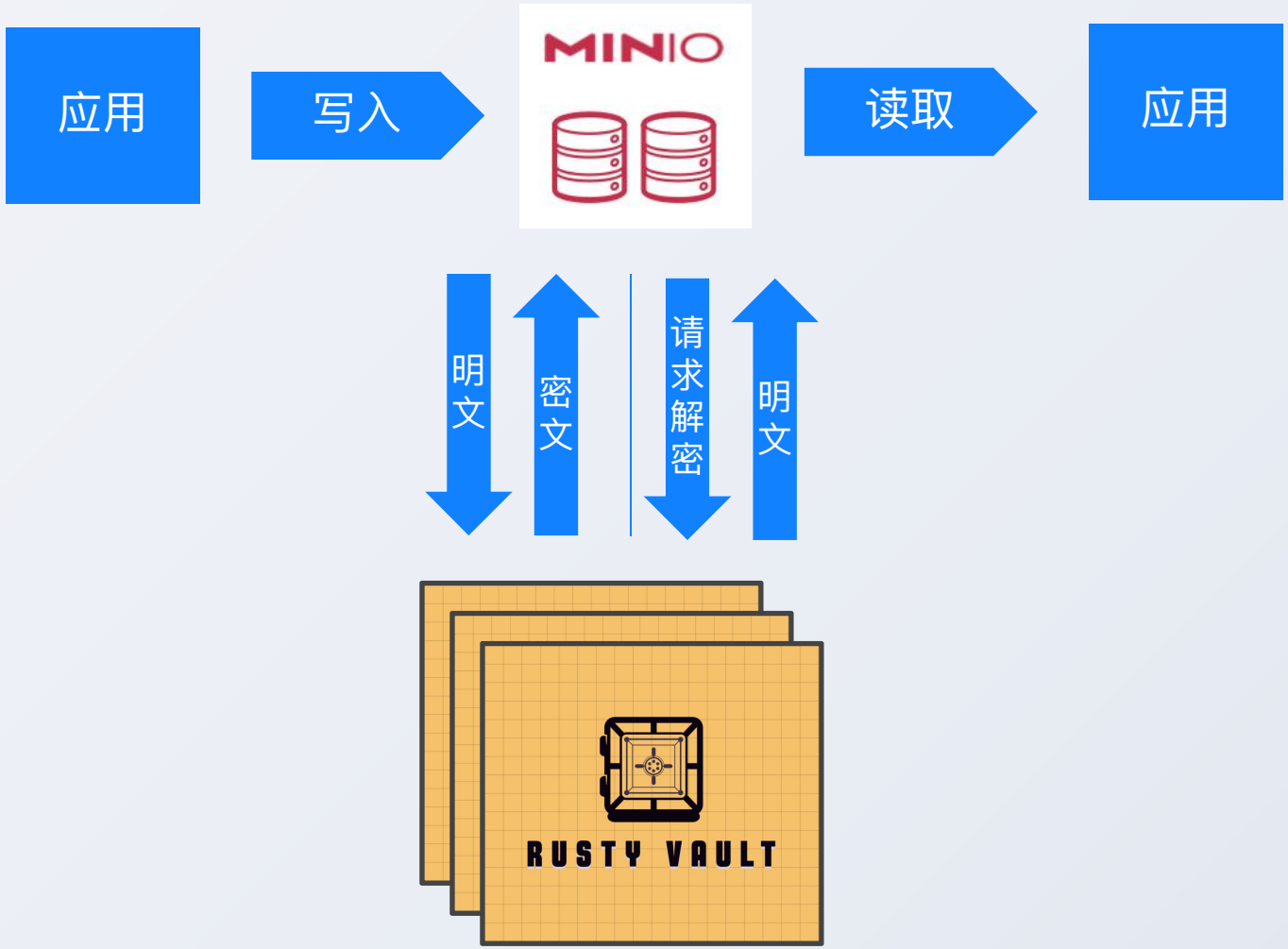
➢ RustyVault–密钥管理方案

- ✓ 权限收敛
- ✓ 私钥集中加密存储
- ✓ 自动化部署，提高研发效能



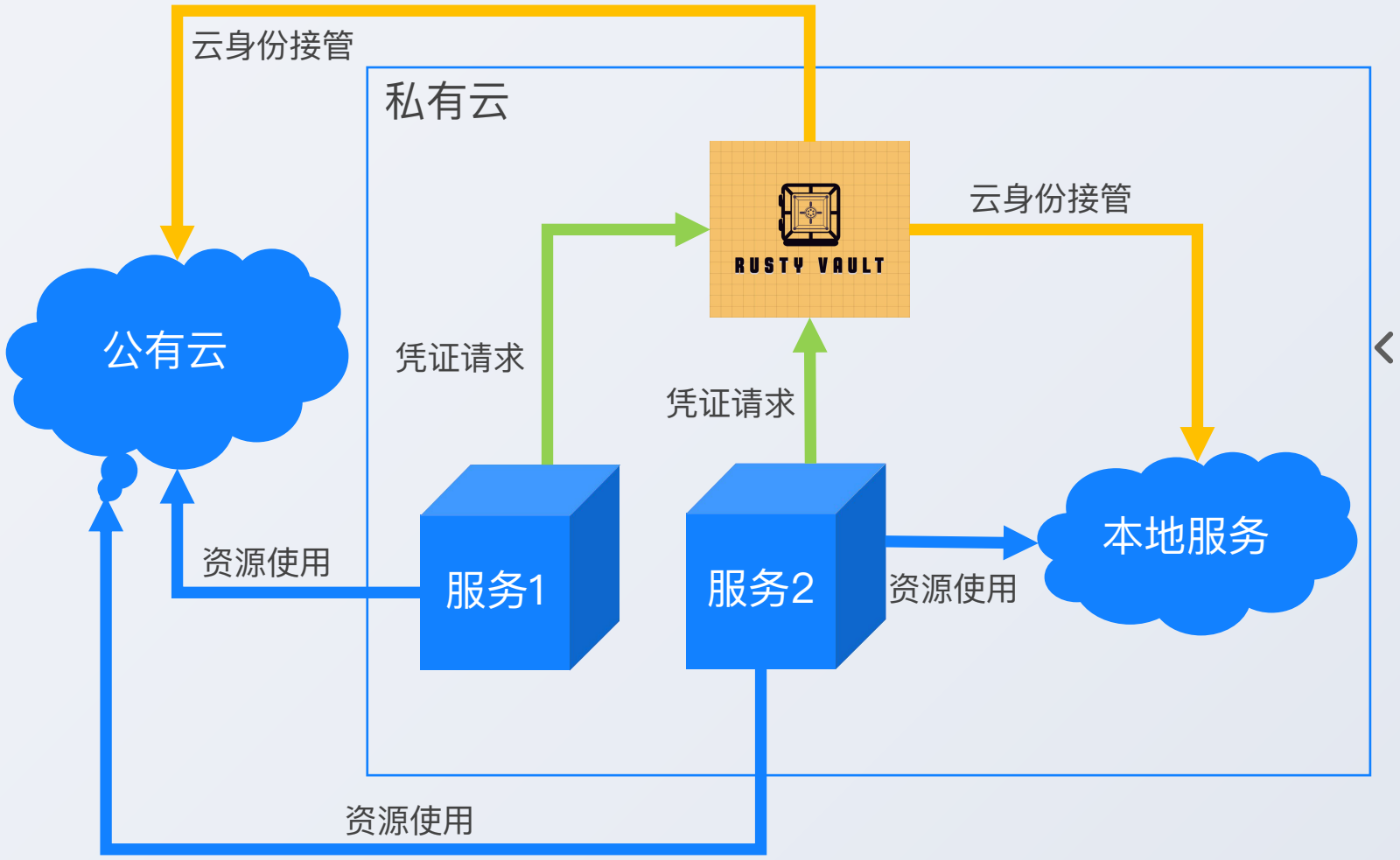
TongSUO 铜锁 RustyVault–典型场景（存储）

- 传统存储加密问题
  - 应用程序有感，需改动
  - 密钥安全性低
- RustyVault 方案
  - ✓ 普适性：应用程序免改造
  - ✓ 高安全性：密钥对外不可见、不感知
  - ✓ 高性能：不占用业务资源



RustyVault–典型场景（混合云/多云）

- 机密信息散落问题
  - 应用所需账密、密钥、凭证等机密信息明文存储
  - 机密信息没有使用审计
  - 机密信息没有生命周期管理
- RustyVault–机密信息管理方案
  - ✓ 统一身份认证
  - ✓ 建立身份和机密信息的映射
  - ✓ 全生命周期管理



项目地址

https://github.com/Tongsuo-Project/RustyVault