Rust China Conf 2020

Shenzhen, China

2020conf.rustcc.cn

Rust语言与嵌入式开发

洛佳

华中科技大学 网络空间安全学院

2020年12月

关于我自己

01 姓名蒋周奇，笔名洛佳。华中科技大学网络空间安全学院。1999年，热爱民族乐器。

02 Rust社区工作者。《Rust日报》编辑，翻译《编写Rust语言的操作系统》。

03 热爱Rust语言和嵌入式、操作系统。参与开发“GD32V”和“BL602”芯片Rust支持。

04 3年Rust开发经验。曾开发“核能”和“科洛桑”游戏服务端引擎，与网易公司商业合作。

裸机上的Rust语言

没有运行时环境，我们仍要选择最优的技术。Rust是我们非常好的选择。

自己的运行时环境

例如RTOS操作系统，程序都运行在环境中。但我们怎么开发自己的环境？

如何搭建嵌入式生态

我们如何提高对新技术的信心？对于嵌入式Rust，搭建生态至关重要。

RustSBI

竞品OpenSBI哪里都很好，只有一个缺点——它是用C语言写的。

裸机上的**Rust**语言

没有运行时环境时，我们挑选编程技术的方法

Rust语言：二十一世纪的语言新星

01

有竞争力的性能

编译型编程语言，开销较小的外部语言接口
极小运行时，无垃圾回收设计
可适用于高性能或性价比嵌入式设备

02

较强的可靠性

严格的代数类型系统、所有权模型
特有的移动语义
社区、学术和产业界项目实际验证

03

生产效率高

文档齐全，各类编译器提示友好、有帮助
统一的包管理器工具链
编译期找到内存、线程安全问题

04

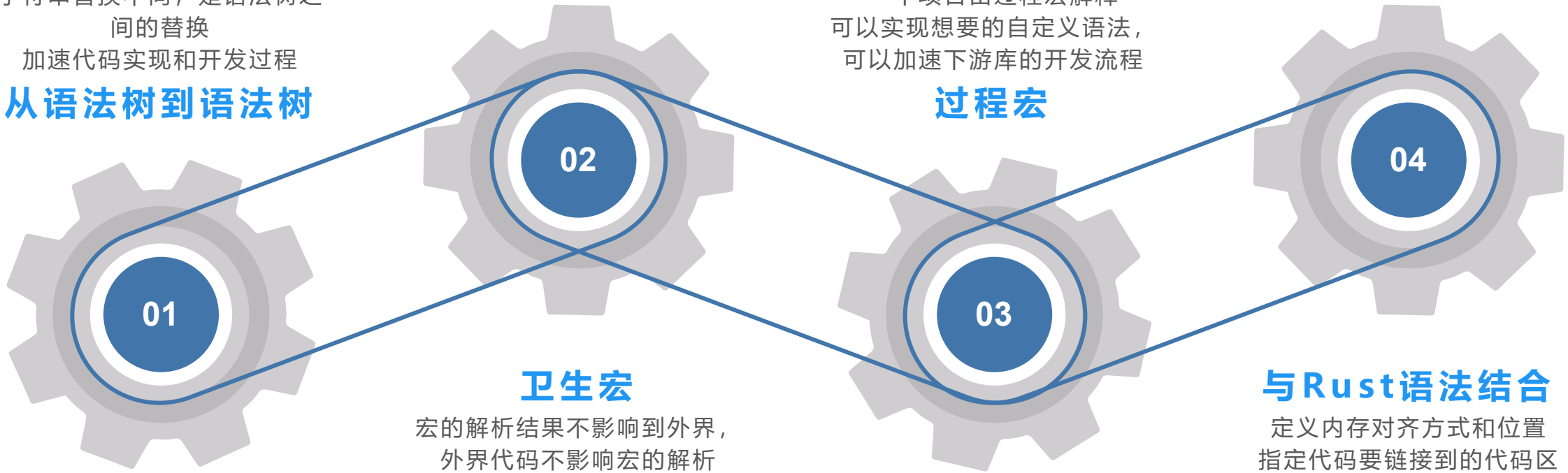
应用范围广

多种编译目标、模块化指令集
厂家可提供自己的编译目标
适用于各类裸机平台

裸机上的Rust语言：宏语法

Rust语言的宏语法和传统的
字符串替换不同，是语法树之
间的替换
加速代码实现和开发过程

从语法树到语法树



卫生宏

宏的解析结果不影响到外界，
外界代码不影响宏的解析
可认为是代码的“内部展开”
不影响代码的上下文

附着在语法项目之前，表示下
一个项目由过程宏解释
可以实现想要的自定义语法，
可以加速下游库的开发流程

过程宏

与Rust语法结合

定义内存对齐方式和位置
指定代码要链接到的代码区
导出宏给其它库使用

裸机上的**Rust**语言：模块化编程

模块

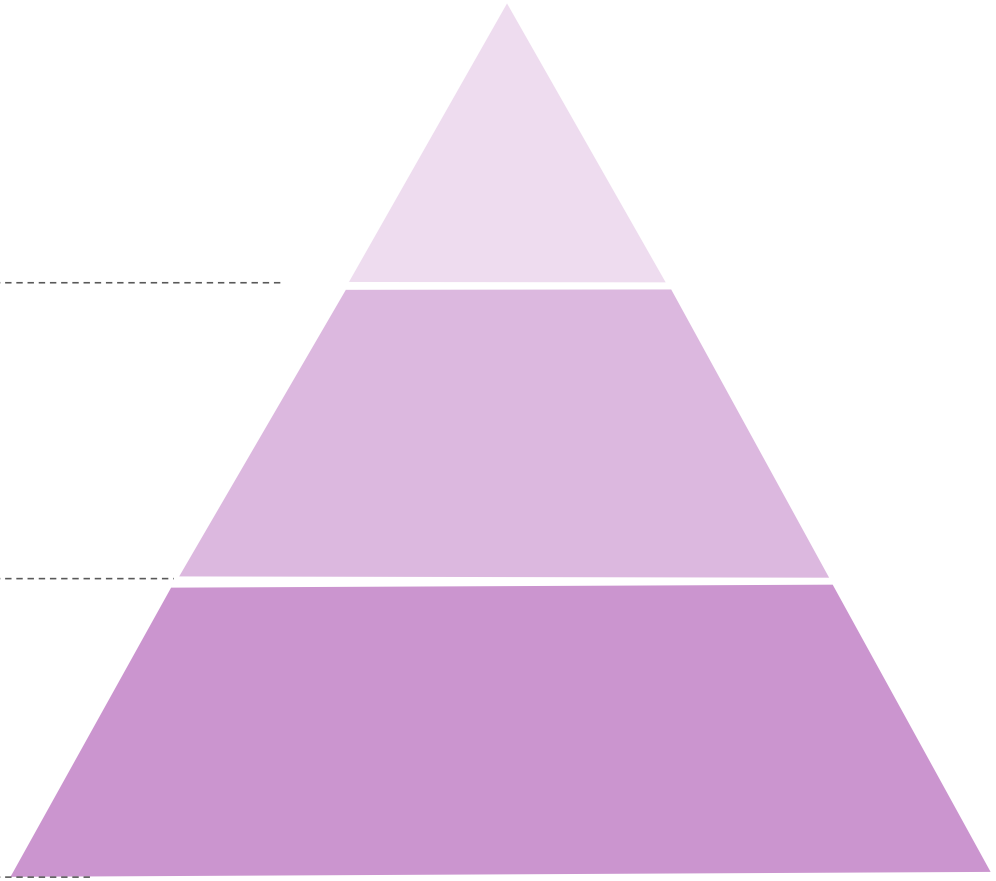
Rust语言可见性分划的最小单位
由专门的模块定义关键字区分不同模块的代码
代码之间的可见性，由专门的关键字规定

包

Rust项目的每个包对应确定的一个二进制目标
由Rust工具链规定的模块化等级
每个包有版本号、作者和许可协议等元数据

项目

核心和外围包组成，或者功能相近的一组包
通常由同一个团队组织和维护，允许添加扩展
习惯上由核心包到功能包，以依赖形式构成



Rust语言丰富的工具链和生态

可自定义的嵌入式目标

支持RISC-V、ARM、MIPS等各类嵌入式架构。若适配了LLVM、GNU等，可添加自主的架构和编译目标。

下载和调试器probe-rs

调试软件解决方案，支持主流硬件和逻辑接口。虽然不是插件式设计，可添加自己的调试设备。

包管理器Cargo

Rust项目统一的包管理器。功能包括编译、功能测试和性能测试，可编写示例程序或使用编译脚本。

社区包网站crates.io

类似于NPM。开源项目可在这里发布，以供用户下载。可以获得大量久经考验的小项目。

自己的运行时环境

程序都运行于环境中。怎么编写自己的环境?

自己的运行时环境：描述硬件

处理器核心

社区统一配发架构软件标准，
也允许厂家添加扩展包

中断和异常

由运行时提供实现，可委托，
允许厂家自己的中断控制器

系统调用

编译型语言，内联汇编和混
合编译，最大化架构优势

外设及寄存器

抽象为拥有所有权的资源，
执行移动语义和单占有模型

用户和系统特权级

根据架构具体设计，通常由
运行时完成切换和返回过程

虚拟内存空间

统一的架构软件标准，可编
译重定向，自己的编译目标

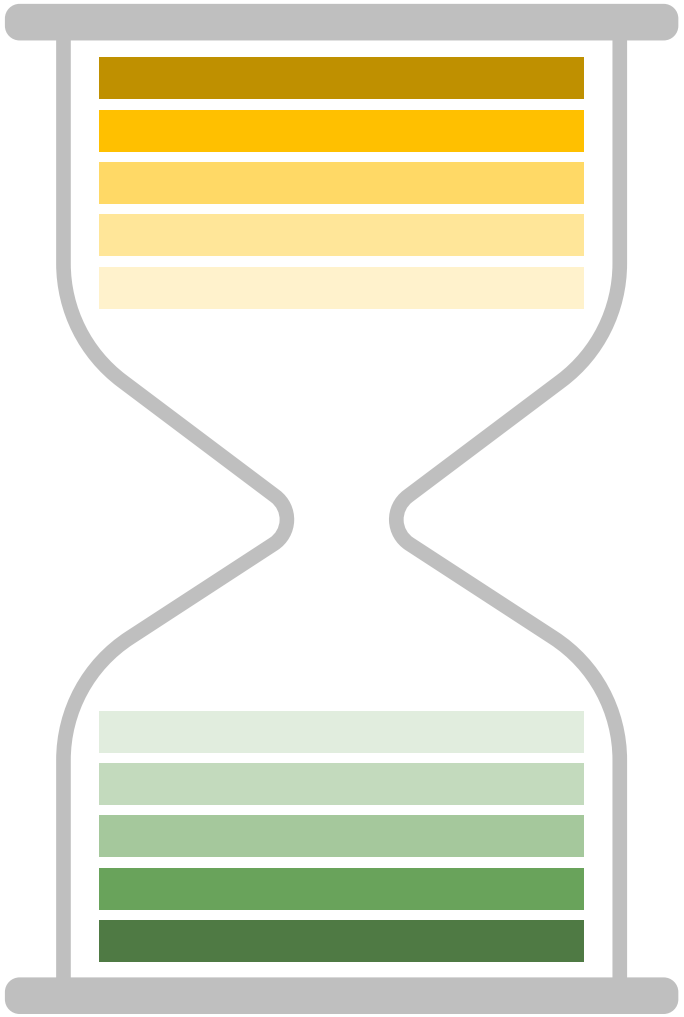
自己的运行时环境：并发性

非阻塞流程

- 使用最小的异步抽象“nb”库，返回“等待中”或“完成”；
- 配合中断，嵌入式较为常用。

线程和进程

- 也就是资源的时空配置；
- “alloc”包实现动态内存；
- 复用已有的小工具包，开发便捷。



中断与上下文

- 仅存在单核时，关闭中断，大胆认为此时仅有单上下文；
- 语言内置async/await语法。

同步机构

- 条件变量、互斥锁等等；
- 和架构提供的原子指令配合，避免数据竞争。

自己的运行时环境：支持你的应用程序

```
aarch64-wrs-vxworks
armv7-wrs-vxworks-eabihf
i686-wrs-vxworks
powerpc-wrs-vxworks
powerpc-wrs-vxworks-spe
powerpc64-wrs-vxworks
x86_64-wrs-vxworks
```

“riscv64gc-vendorname-youros”

片内存储的烧录和调试软件
选用海量成品文件系统和存储卡支持，也可以提供自己的支持软件

系统模块、插件和动态链接库等等
内存安全的语言特性，适合现代对安全敏感的开发需求

编译到特定目标

嵌入式和通用存储

使用环境内的设备

编写系统功能

厂家可添加自己的编译目标
实时操作系统不一定用Rust编写，但都可以运行Rust程序

Rust是适合编写硬件驱动的语言
有产权的代码，可以以混合链接的方式，与Rust联合编译为二进制

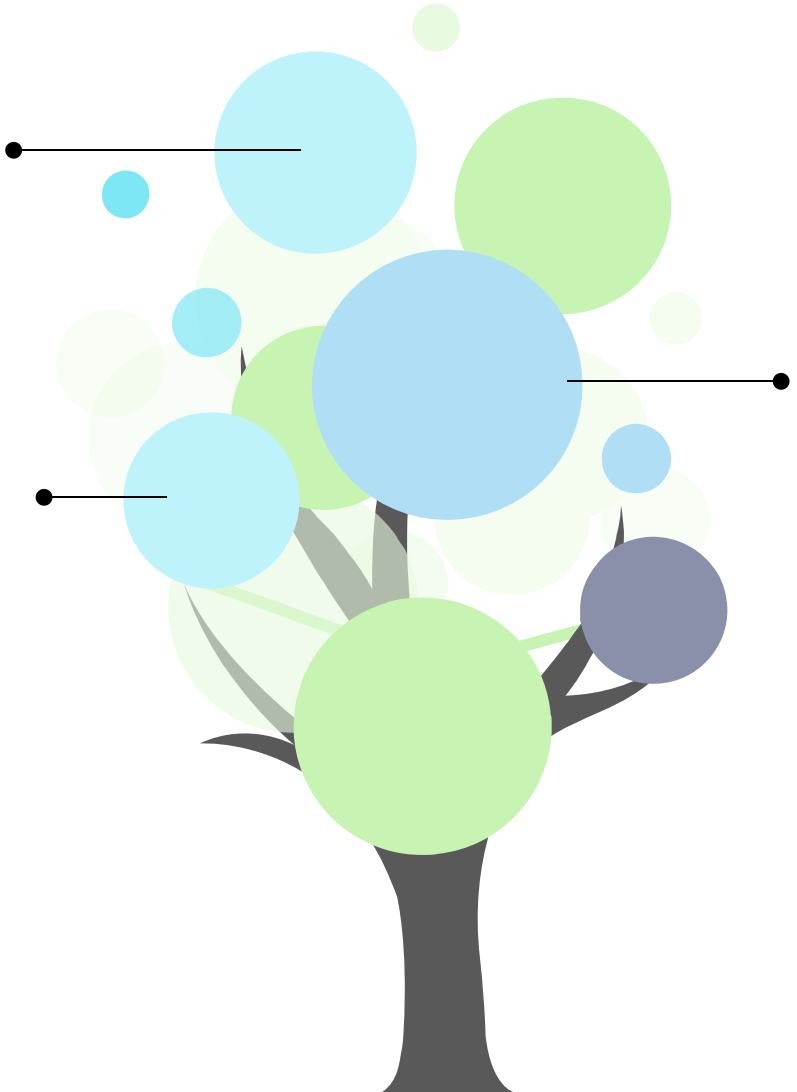
自己的运行时环境：计算机网络

有线和射频连接

Rust嵌入式社区正在探索技术标准，包括蓝牙、WiFi等硬件

smoltcp库

非常好的TCP系列协议实现抽象和性能都好，不妨试试看？



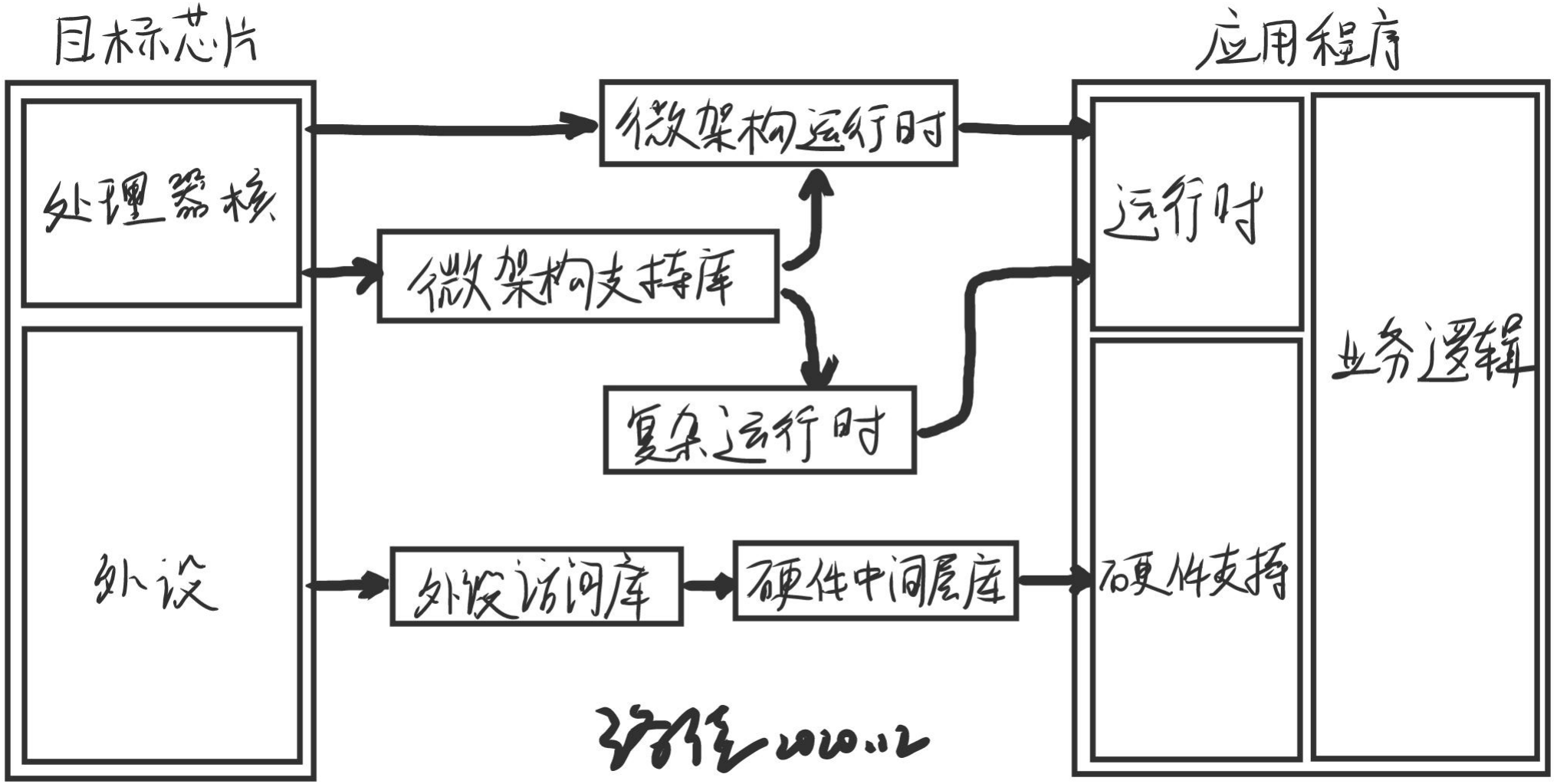
使用因特网

抽象得到的网络连接，可搭配多种缓冲区库和协议库共同使用

如何搭建嵌入式生态

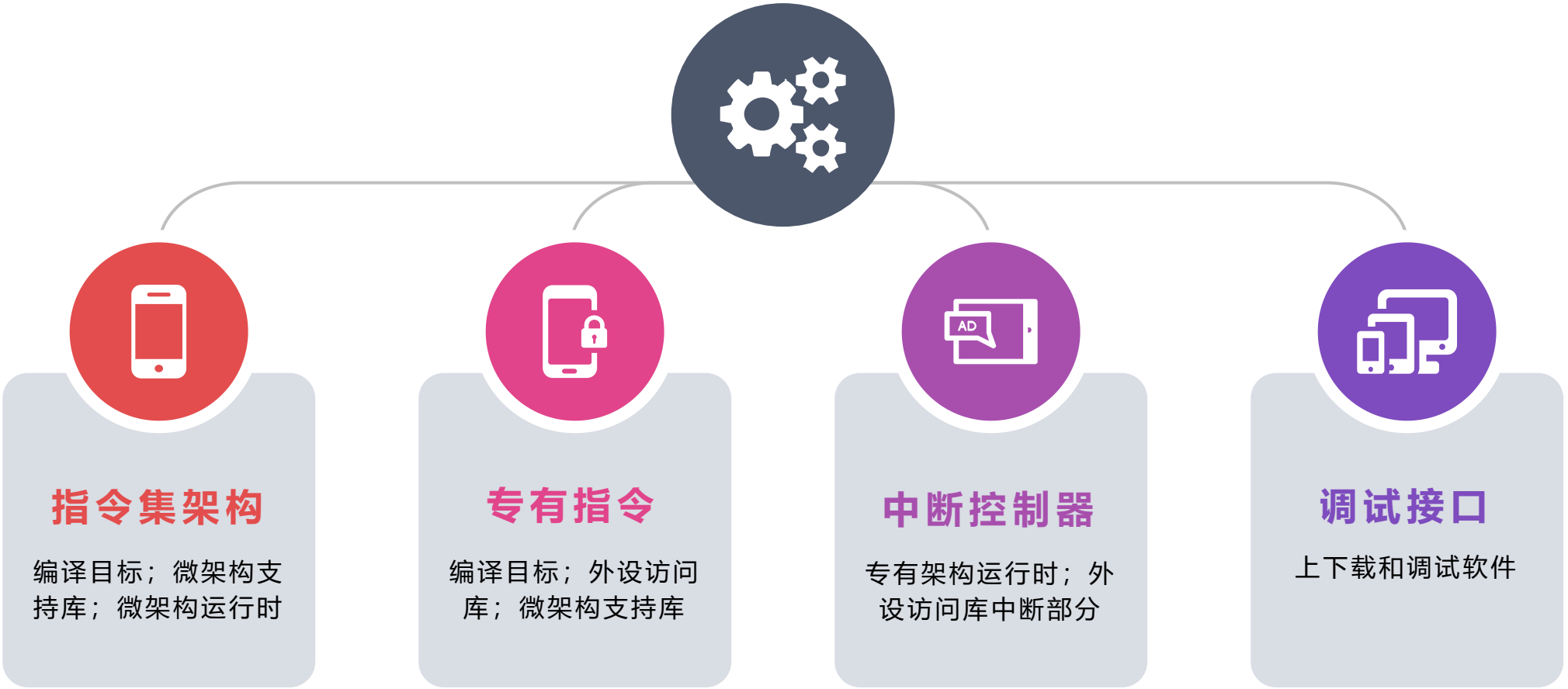
要提高对新技术的信心，搭建生态至关重要。

Rust语言的嵌入式生态（2020年12月）



嵌入式生态：支持你的架构和指令集

通用或专有架构



指令集架构

编译目标；微架构支持库；微架构运行时

专有指令

编译目标；外设访问库；微架构支持库

中断控制器

专有架构运行时；外设访问库中断部分

调试接口

上下载和调试软件

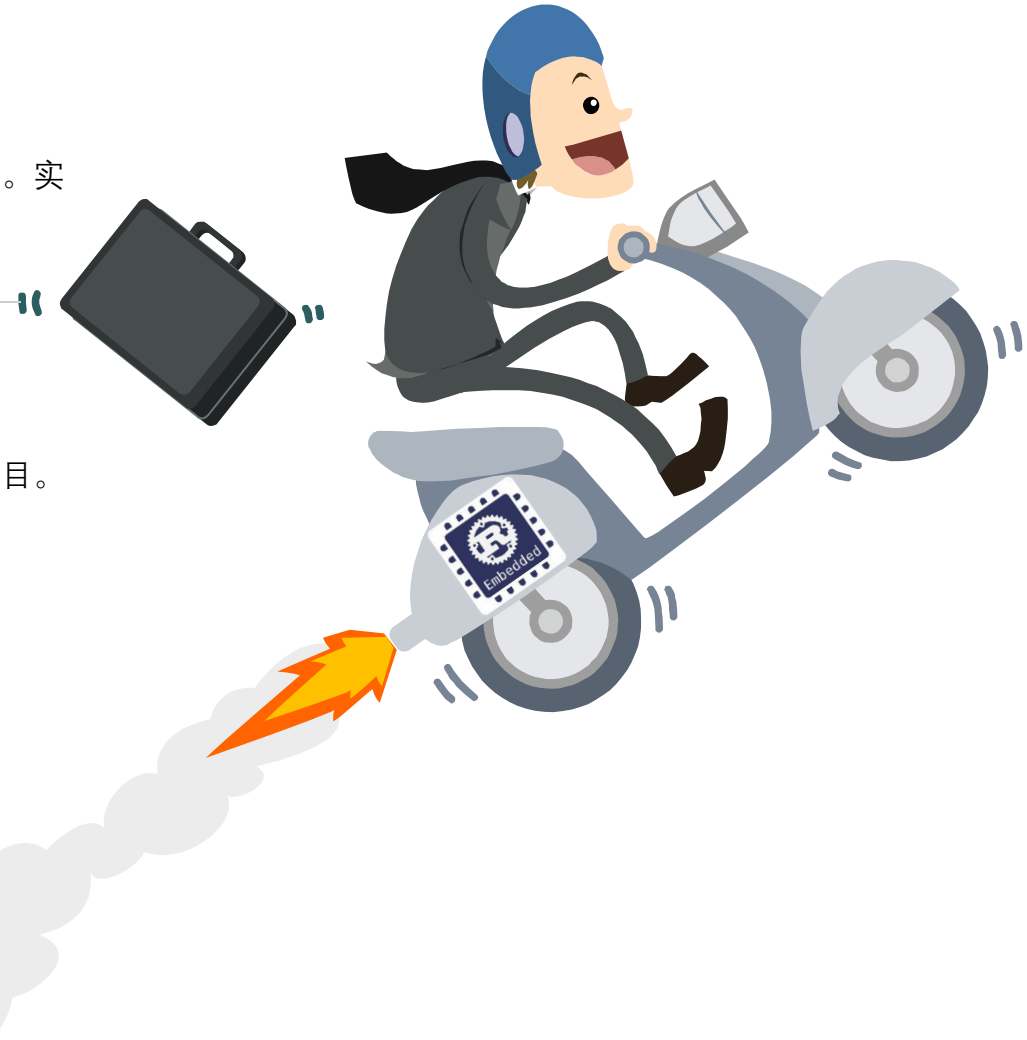
嵌入式生态：**embedded-hal**标准

统一的标准

使用Rust语言，针对外设本身的抽象。实现由实现库完成

扩展性较好

模块间衔接嵌套便捷，能整合相关项目。非常容易为新的芯片编写支持库



生态圈庞大

支持海量市售芯片，包括K210、GD32V和BL602等。支持片内外设和外挂外设

编码更容易

厂家只需要机器生成外设库，然后编写中间层库，即可完成对此标准的支持

嵌入式生态：成熟的运行时

RTIC框架

中断驱动的异步实时系统。社区开源项目，生态成熟，论文成果支持*。

2017.5

Tock操作系统

针对微处理器的安全实时操作系统。已用于手表、智能路标和加密狗等产品。

2017.10

微架构运行时

如riscv-rt等，我们有统一的运行时。也能针对硬件，设计自己的运行时。

2018.2

Drone操作系统

高效并发的操作系统，完整的实时操作系统功能。拥有优秀的配套调试软件。

2019.7

* Eriksson, J., et.al. (2013, June). Real-time for the masses, step 1: Programming API and static priority SRP kernel primitives. In Industrial Embedded Systems (SIES), 2013 8th IEEE International Symposium on (pp. 110-113). IEEE.

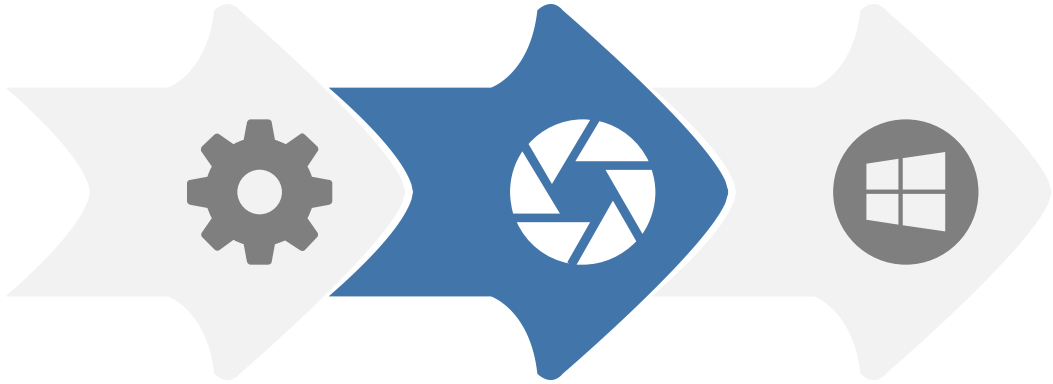
为什么嵌入式**Rust**中没有**BSP**的概念

良好的抽象方法

由外设库和中间层库组成

换用新架构

板级支持库已经被拆分



模块化开发

很少需要庞大的单个软件包

RustSBI：新型操作系统引导软件

OpenSBI哪都好，只有一个缺点：它是用C语言写的

什么是SBI?

引导程序

- 启动系统内核
- 收集设备信息，提供给操作系统
- 类似于UEFI

统一的硬件环境

- 通过系统调用，提供实用功能
- 监控所有的处理器核
- 发送跨核软中断
- 提供兼容性支持

是一个标准

- 适用于RISC-V
- 期望消除部分硬件差异
- 它有多个实现，如OpenSBI

SBI实现的组成部分


SBI
实现
硬件环境接口
Supervisor Binary Interface
硬件运行时
引导启动模块
兼容性模块

欢迎使用**RustSBI——发布0.1版本**

功能齐全

已实现OpenSBI的大量功能，
支持针对实现自定义扩展功能

开发快捷

从引导程序到用户内核，统一编译工具链

安全高效

完全使用Rust语言编写

技术标准

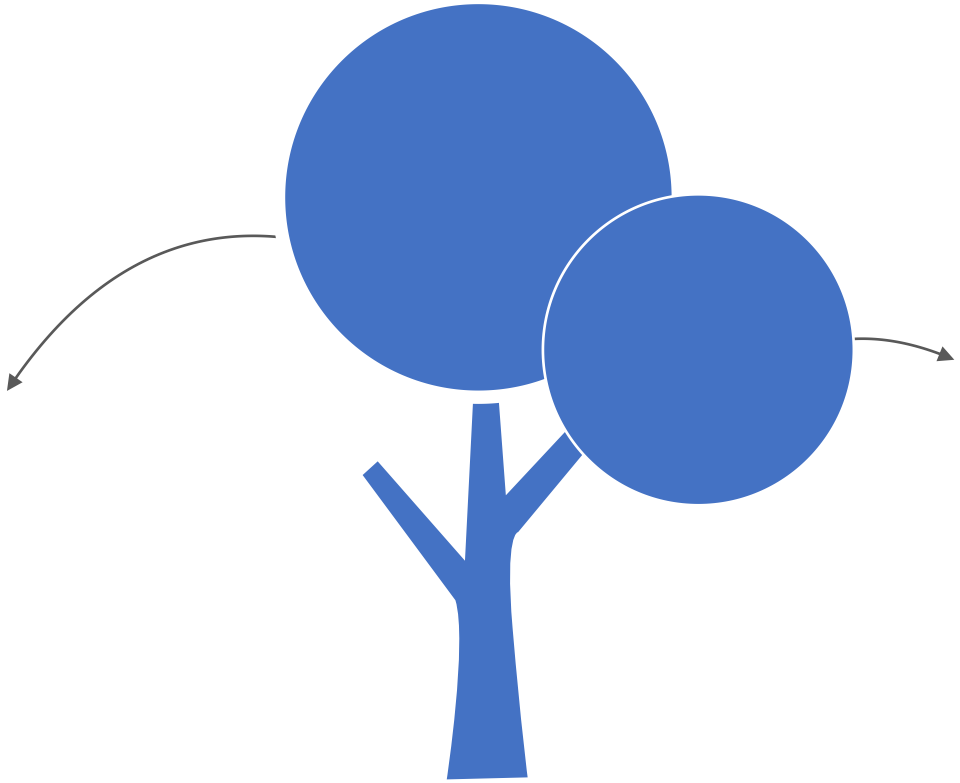
已被收录入RISC-V SBI标准，
实现编号为4

为什么用**Rust**语言开发**SBI**实现

生态兼容

嵌入式Rust生态圈

相比在C语言项目外包一层



开发便捷

容易支持新硬件

编译期语义约束强

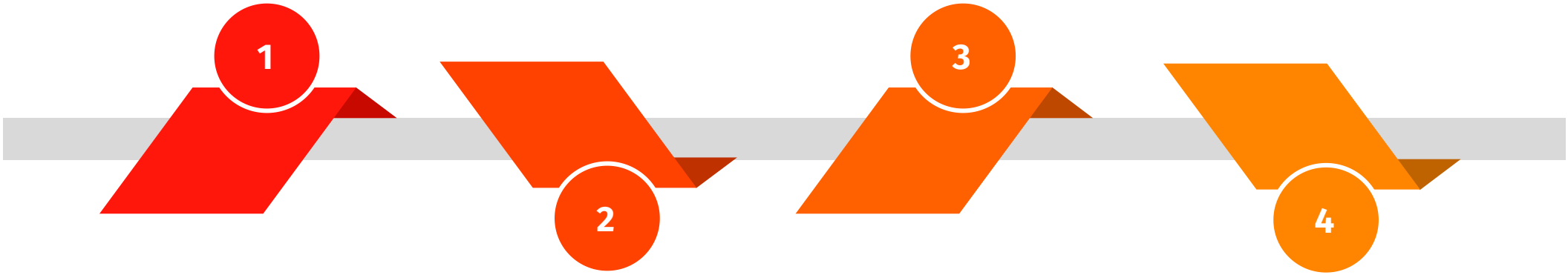
RustSBI：兼容性设计的实现与展望

适用范围

目前可以模拟不存在的指令、寄存器等

延长生命周期

硬件上设计专门的外设，用它修改指令的功能和寄存器的功能



旧的RISC-V标准

标准会不断演化；预留未来的标准

支持更多指令集模块

如使用硬件外设模拟密码学指令，允许旧的硬件上使用新的指令集

致谢

- 感谢Rust中文社区提供这次演讲机会。
- RustSBI项目得到了许多同学的支持，他们包括：王润基学长、吴一凡学长。学长们提供的灵感让我受益匪浅。感谢在鹏城实验室指导我的向勇教授和陈渝教授。
- 应用Rust嵌入式开发得到了许多同学的实际项目验证。感谢车春池和更多的同学在操作系统内核开发上的研究。
- 感谢支持Rust嵌入式开发的公司和研究团队，感谢世界各地使用Rust的嵌入式爱好者。

谢谢各位

Rust语言与嵌入式开发

洛佳

华中科技大学 网络空间安全学院