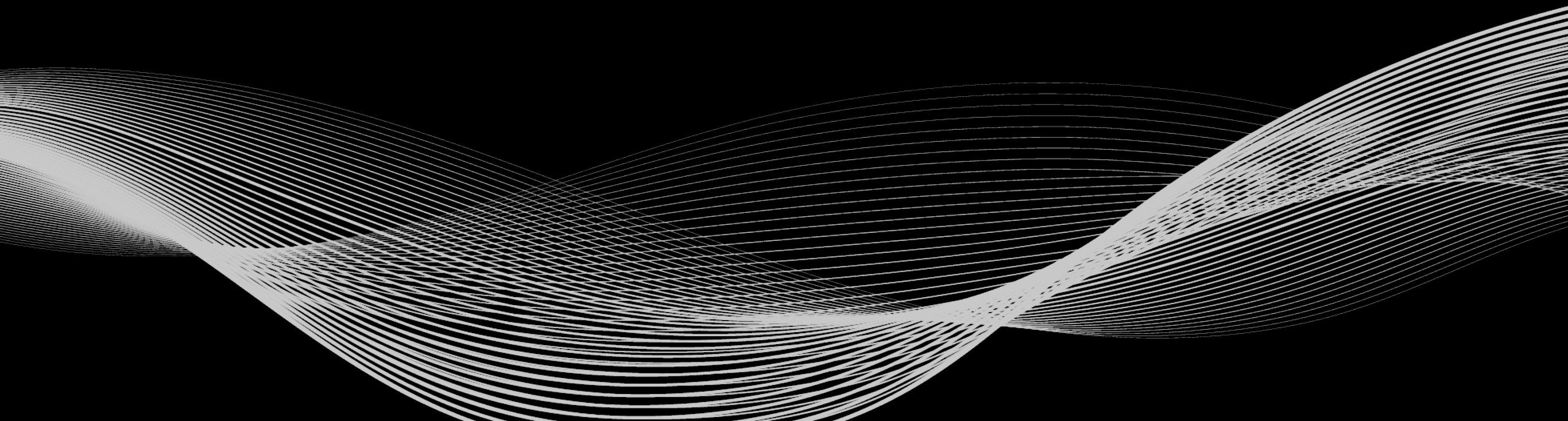


# Substrate 快速入门与实战

---

讲师: Bryan





# 第一节课



# Substrate 介绍



## Substrate 介绍

- Substrate 是 Parity 创建的下一代区块链框架
- 现在已经是 2.0 版本，和 1.0 相比提供了更多的模块和更多的扩展功能
- <https://hackmd.io/@xlc/substrate-2-changelog>
- 大家可以基于 Substrate 轻松的开发出可升级，高效率，创新性的区块链项目



## Substrate 特性

- 可升级
  - 传统区块链项目，每次重大升级都需要硬分叉或者软分叉
  - 很多新的区块链项目都有每半年左右执行一次硬分叉的要求，方便新功能的发布
  - Substrate 提供了一个更好的，不需要分叉的升级方案

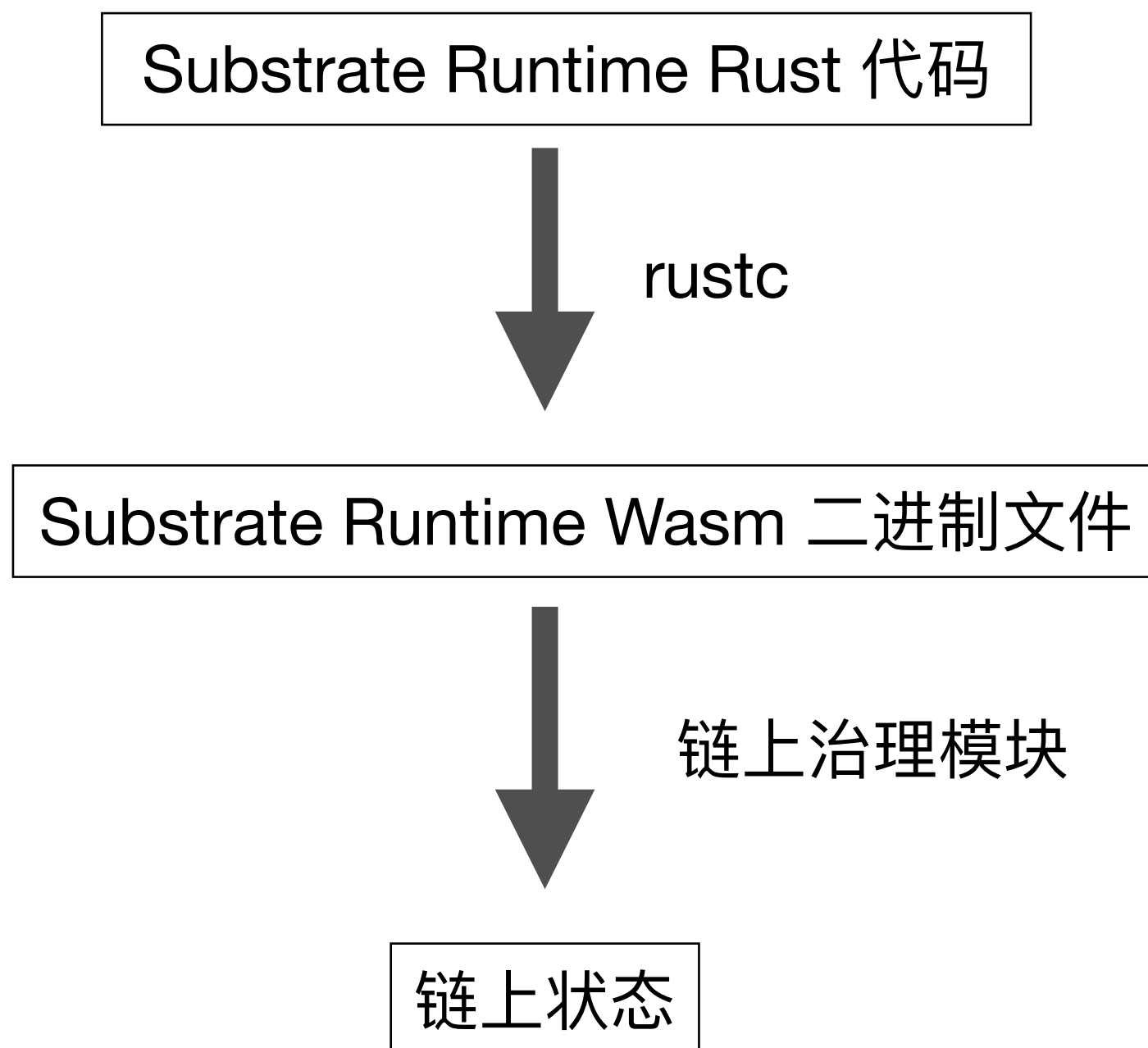


## Substrate 特性：可升级

- Substrate 节点是由两部分组成
  - Runtime
    - 链上逻辑
  - Client
    - p2p, db, cli, 交易池, 出块逻辑, Wasm解释器

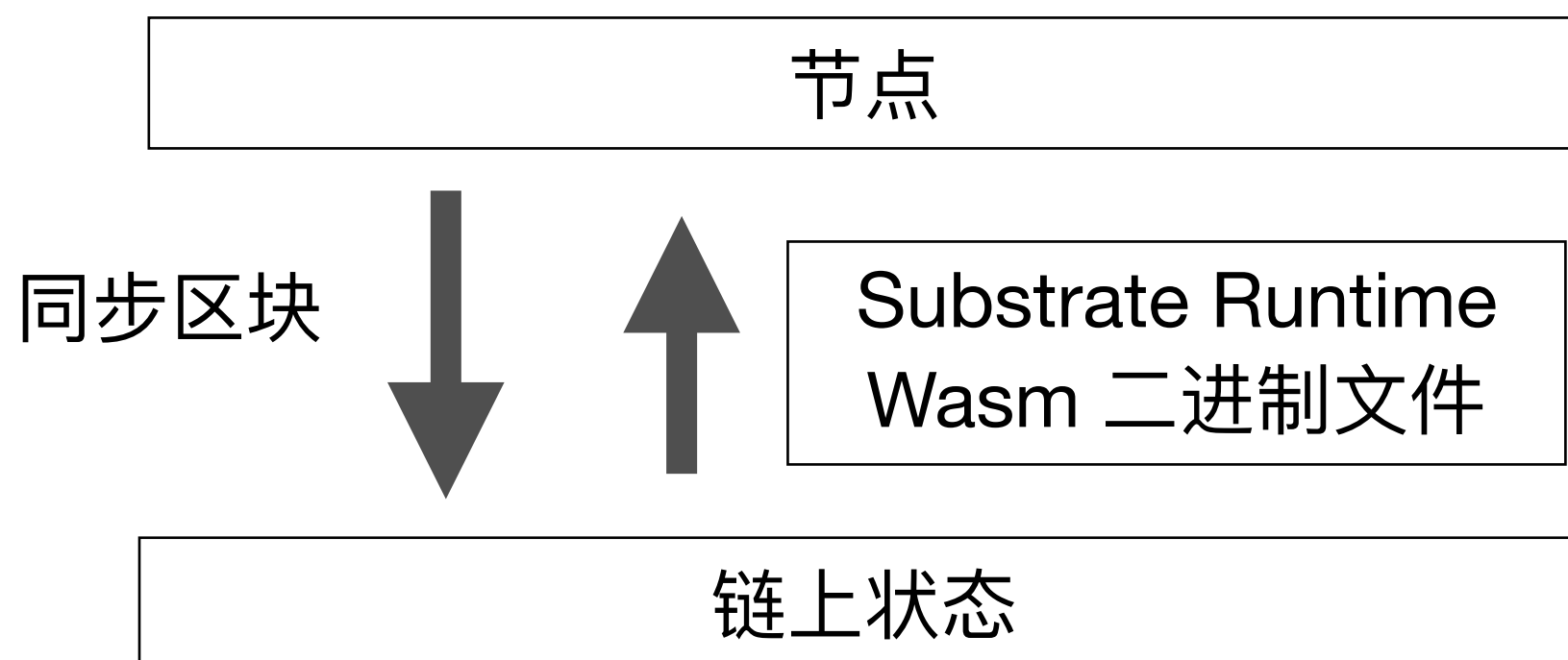


## Substrate 特性：可升级





## Substrate 特性：可升级







## Substrate 特性

- 高效率
  - Substrate 的设计从各个角度都是以最高的执行效率为出发点
  - Rust 语言的选择可以在保证安全性的同时拥有最大自由度优化的可能性
  - 内置的 PoS 共识机制避免了不必要的挖矿和高额度的TPS
- 轻节点
  - Patricia Merkle Tree 实现的 Trie DB
  - 高效率，轻量级，去中心化，去信任化的方式，验证区块头和链上数据
  - 这样使得移动端，物联网端，和网页端的轻节点实现的可能性



## Substrate 特性

- 创新性
  - 抽象的模块化接口
  - WebAssembly解释器
  - 不同的开发语言的选择
    - C++: <https://github.com/soramitsu/kagome>
    - Go: <https://github.com/ChainSafe/gossamer>
- 第三方库
- 定制模块
- Offchain Worker
- 智能合约



## Substrate 特性总结

- Substrate 可以帮助大家开发出最好的区块链
- 抽象的模块化接口和WebAssembly解释器的设计
- 可升级
  - 避免了硬分叉的需求
- 高效率
  - 高额度TPS
  - 轻节点
- 创新性
  - 只有你想不到的没有不能通过 Substrate 实现的



# Polkadot 波卡生态圈



## Polkadot 波卡生态圈

- Polkadot 波卡链
- 波卡是 Parity 是基于 Substrate 开发的下一代区块链项目
- Polkadot PoC-2 版本是 Substrate 的前身



## Polkadot 波卡生态圈

- 波卡链是一个中继链，为了实现跨链沟通
- 波卡链支持私有链，公链，oracles (预言机)，等等组件之间去信任化的交流
- 波卡生态圈包含了 relay chain 中继链，parachain 平行链，bridges 转接桥

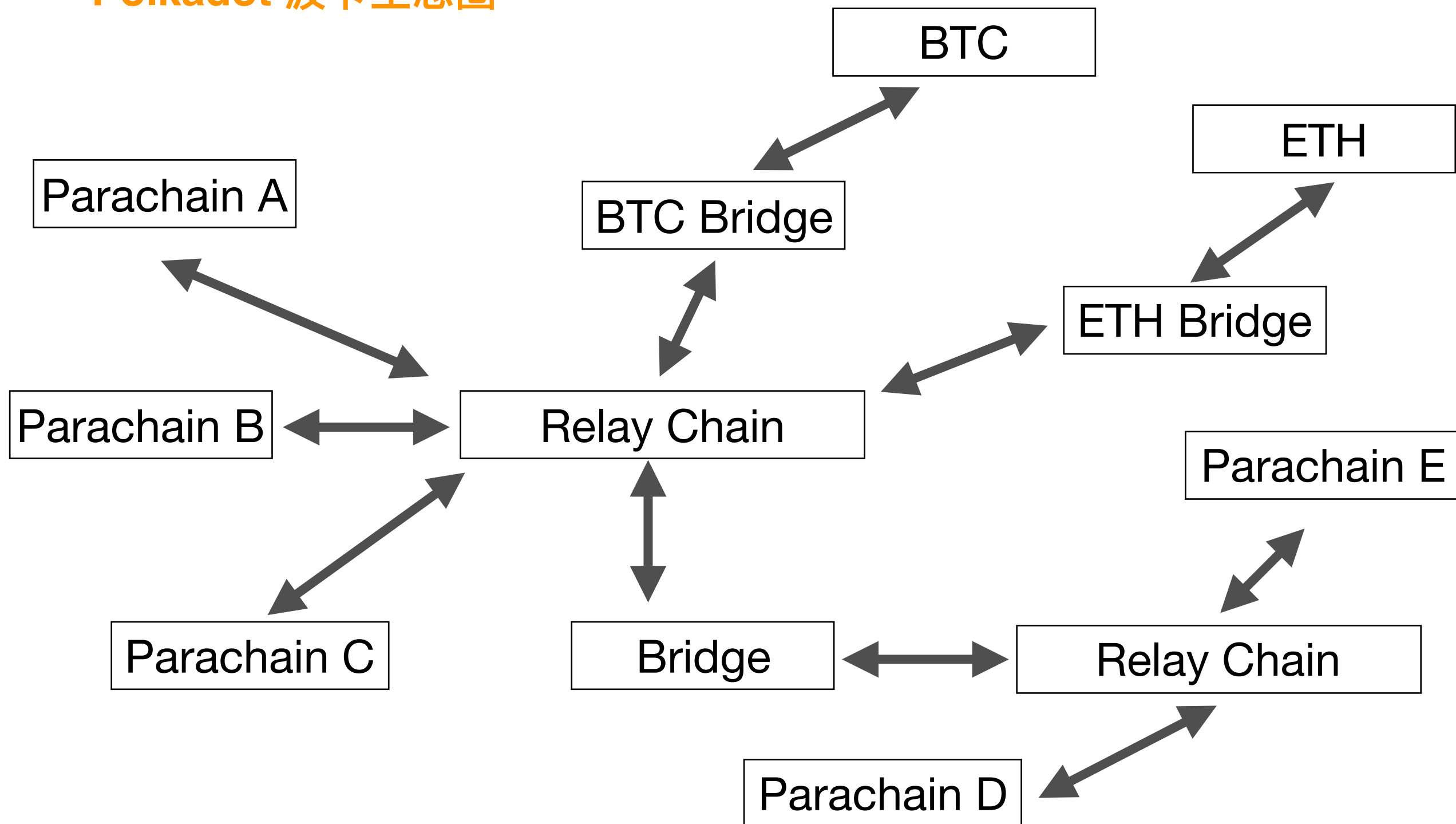


## Polkadot 波卡生态圈

- Relay chain 中继链的作用是达成关于平行链和中继链状态的共识
- Parachain 平行链是独立的区块链，拥有自己独立的链上数据和交易处理，但同时利用了中继链来保证安全性
- Bridges 转接桥让波卡生态圈能够与其他原先不在波卡生态圈中得的独立链进行交流，比如比特币和以太坊



## Polkadot 波卡生态圈







## Polkadot 波卡生态圈 总结

- Polkadot 波卡生态圈可以实现真正的去中心化的互联网生态圈
- Relay chain 中继链
  - 跨链交流
- Parachain 平行链
  - 实现业务逻辑
- Bridges 转接桥
  - 让其他链也能够加入波卡生态圈



# 启动节点



## 环境

- 安装 Rust 环境和相关工具
- `curl https://getsubstrate.io -sSf | bash -s -- --fast`



## 启动节点

- 获取代码
  - `git clone https://github.com/paritytech/substrate.git`
  - `cd substrate`
  - `# commit`  
`409f5aa10a32a7b18f1c5a5eded97d24181865dc`



## 启动节点

- 初始化
- `./scripts/init.sh`



## 启动节点

- 编译
  - cargo build
  - cargo build --release



## 启动节点

- 编译时环境变量
  - SKIP\_WASM\_BUILD
    - 避免编译Wasm目标
  - BUILD\_DUMMY\_WASM\_BINARY
    - 避免编译Wasm，使用空Wasm文件
  - WASM\_BUILD\_TYPE
    - Wasm编译类型：release 或者 debug
  - WASM\_TARGET\_DIRECTORY
    - 将Wasm编译结果拷贝到指定目录。必须是绝对路径



## 启动节点

- 常用命令
  - `SKIP_WASM_BUILD= cargo check`
    - 只进行检查
  - `WASM_BUILD_TYPE=release cargo run -- --dev`
  - Wasm目标使用release, Native目标使用debug





## 启动节点

- 启动
  - `WASM_BUILD_TYPE=release cargo run -- --dev`
  - 启动本地测试网
  - `WASM_BUILD_TYPE=release cargo run`
  - 加入公共测试网
  - `cargo run --release`



## 启动节点

- 测试
- 打开前端
  - <https://polkadot.js.org/apps/#/explorer>
- 点击 Settings 配置页面
- 选择本地节点
- Telemetry 服务 <http://telemetry.polkadot.io>



# 加密猫 dApp



# 作业



## 作业

- 安装 Rust 环境
- 安装IDE
  - VSCode
    - 插件： Rust (rls), CodeLLDB, BetterTOML
  - CLion
    - 插件： Rust, Toml



## 作业

- 创建项目
  - `curl https://raw.githubusercontent.com/paritytech/substrate-up/d4a258a3b7069f92228a245477fed914334c9b82/substrate-node-new -sSf | sh -s <项目名> <作者名>`
- 从新项目启动单节点测试网络
- 使用 `https://polkadot.js.org/apps/` 连接本地测试网络并成功发起一笔转账交易



一块链习

**THANK YOU!**

**Contact us:**  
**[info@yikuailianxi.com](mailto:info@yikuailianxi.com)**

