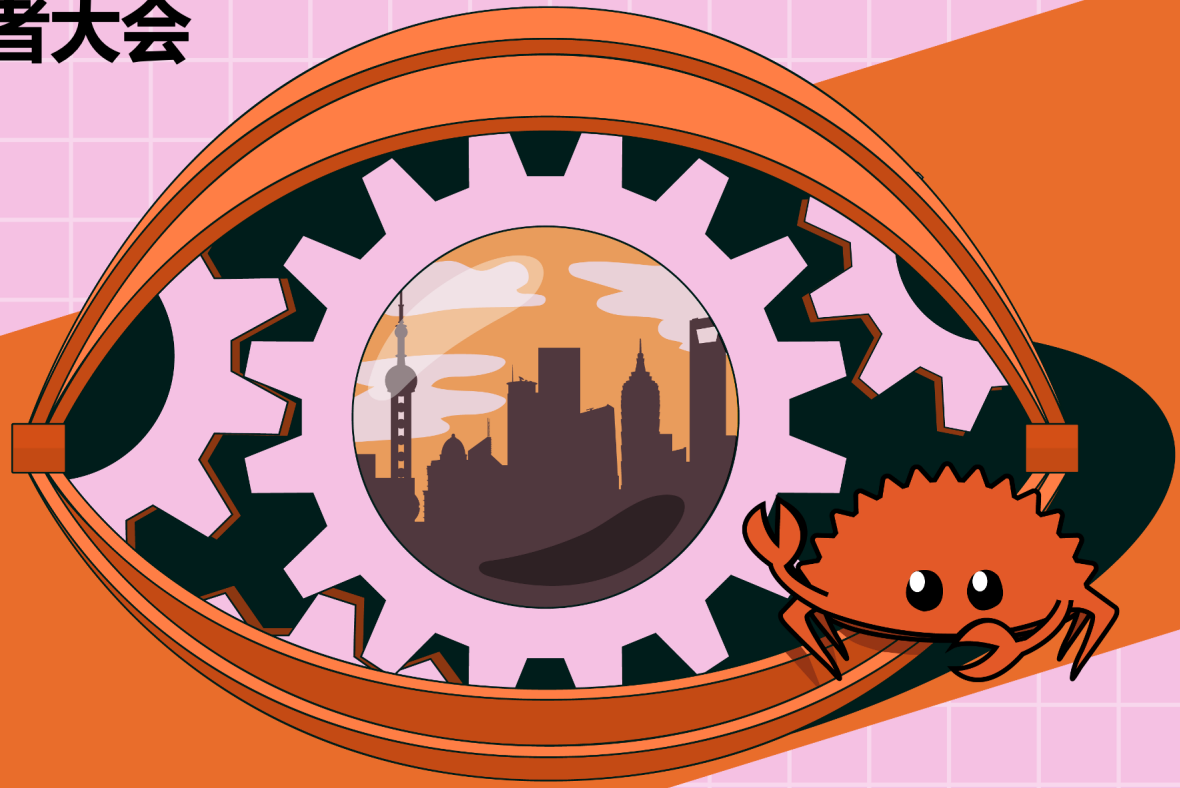
RUST CHINA CONF 2023

第三届中国Rust开发者大会



6.17-6.18 @Shanghai

运行在浏览器中的 P2P 网络

李敏成
from RingsNetwork

Montivation

连接所有钱包持有者

去中心化的 Pure P2P 网络

The Idea

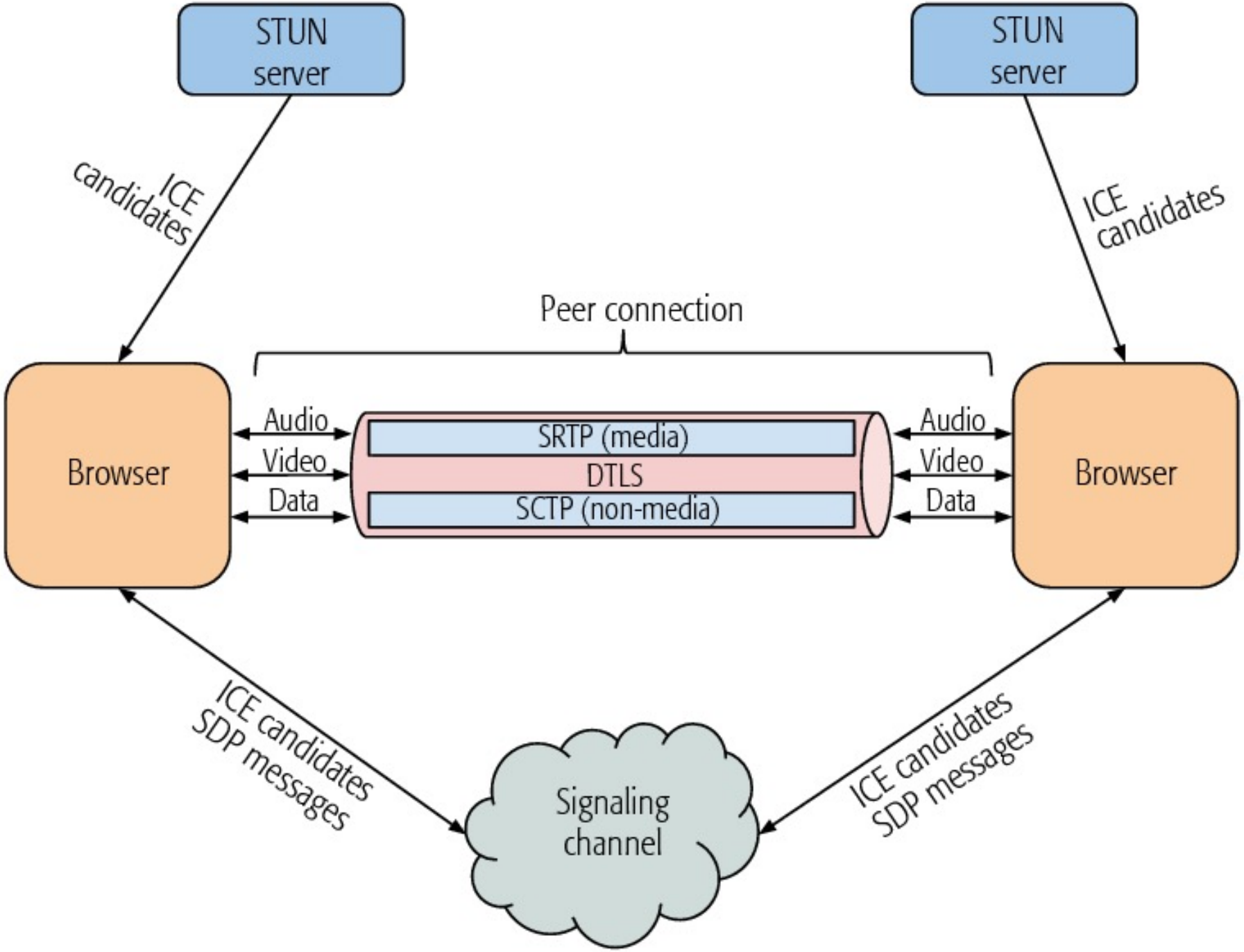
最好的节点载体是浏览器

- 用户群体
- 钱包插件
- 运行环境

How to P2P

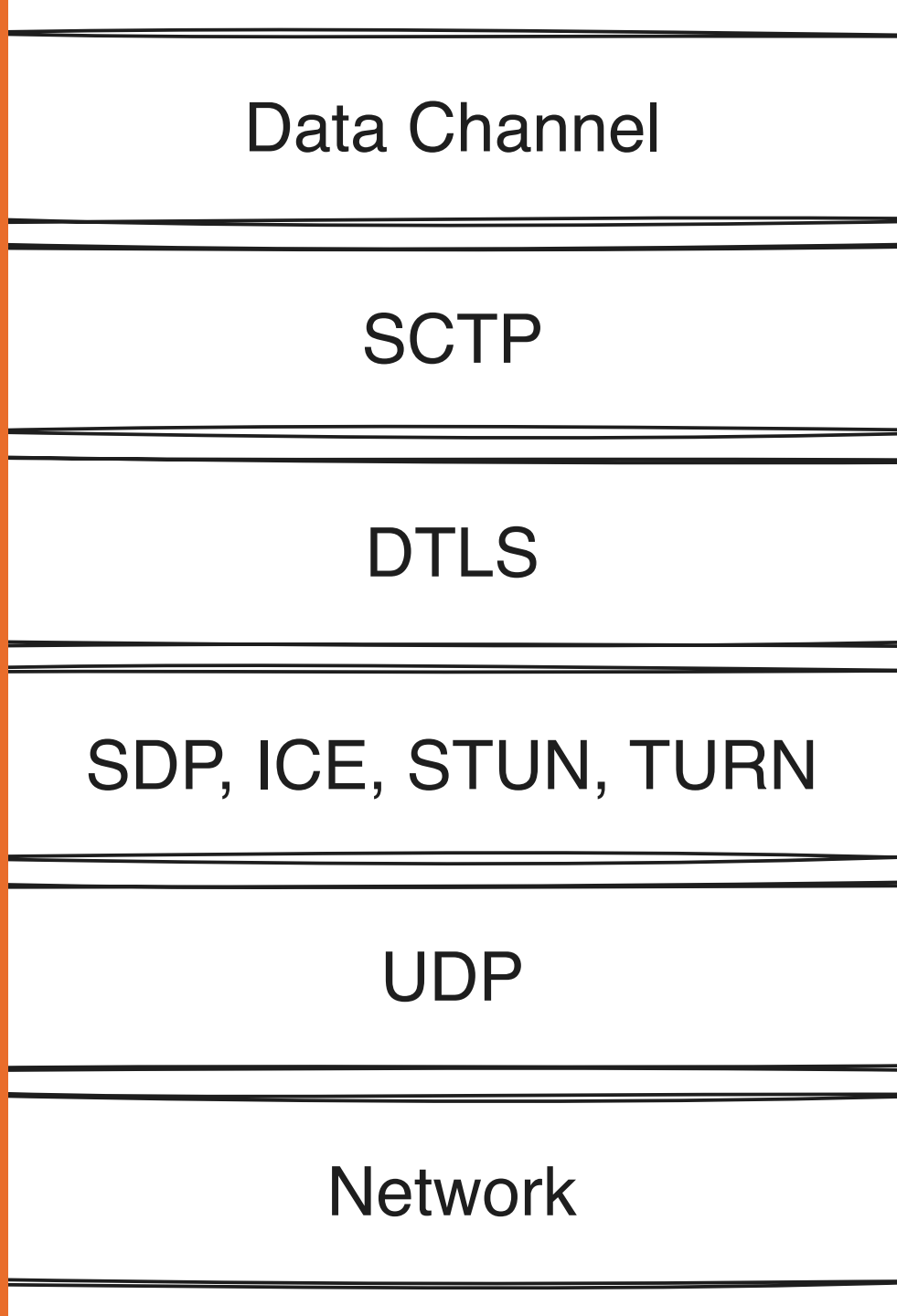
- Did: Wallet Address
- E2E secure(sign/encryption): Wallet
- Discovery and Routing (DHT): Chord / Correct Chord / Kademlia
- NAT and Firewall Traversal: STUN / TURN / Relaying
- Transport: ?

Introduce WebRTC



WebRTC Data Channel

1. Stream Control Transmission Protocol (SCTP)
2. Datagram Transport Layer Security (DTLS)
3. Session Description Protocol (SDP)
4. Interactive Connectivity Establishment (ICE)
5. Session Traversal Utilities for NAT (STUN)
6. Traversal Using Relay NAT (TURN)



WebRTC Implementations in Rust

Lib	Runtime
webrtc-rs	native with tokio
web-sys	browser

Browser Node (WASM) + Server Node (Native)

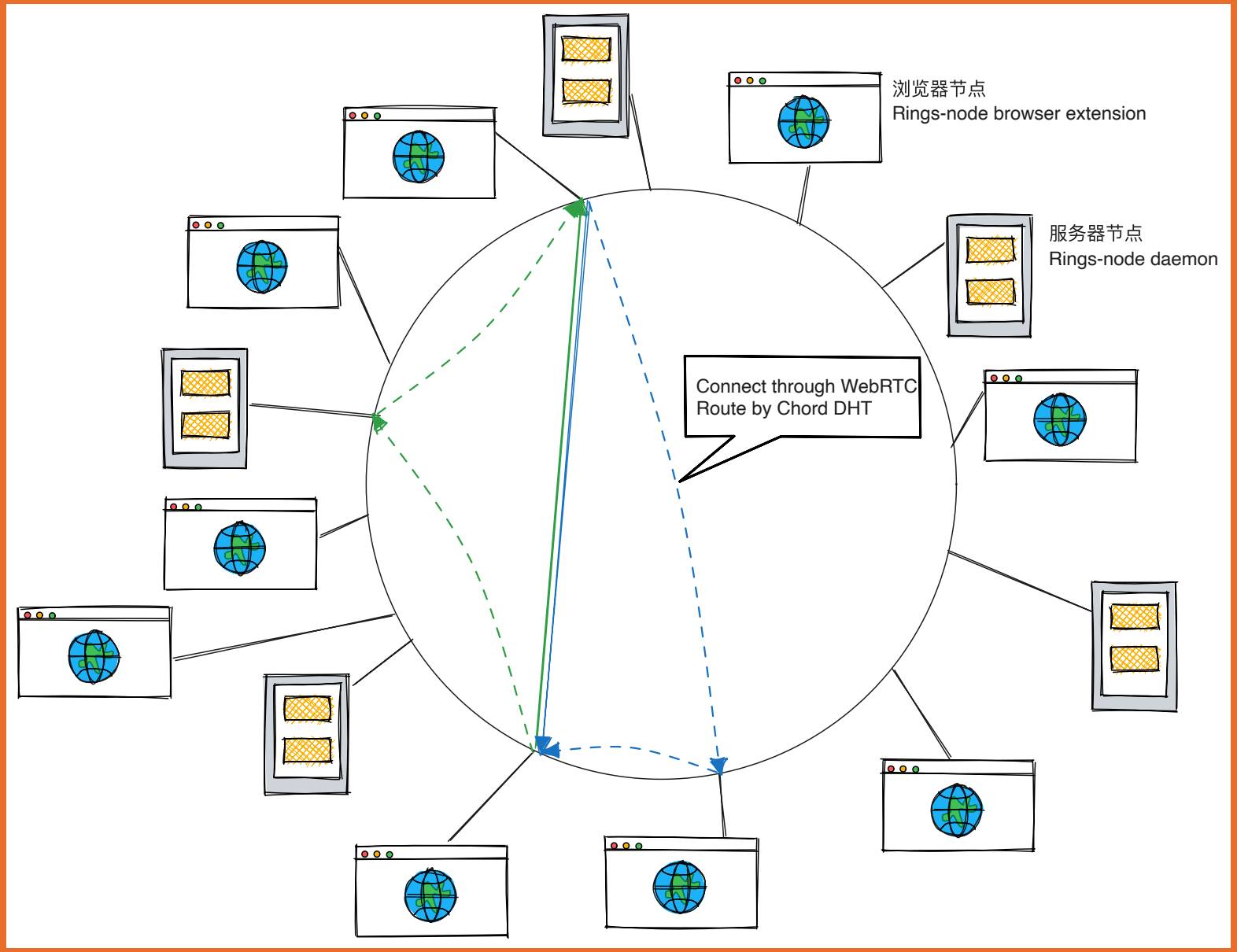
Implement once (part of), run anywhere!

Introduce rings-node

- A structured P2P network implementation.
- Using WebRTC and Chord algorithm.
- With full WebAssembly (WASM) support.

Repo: https://github.com/RingsNetwork/rings-node

Node communication



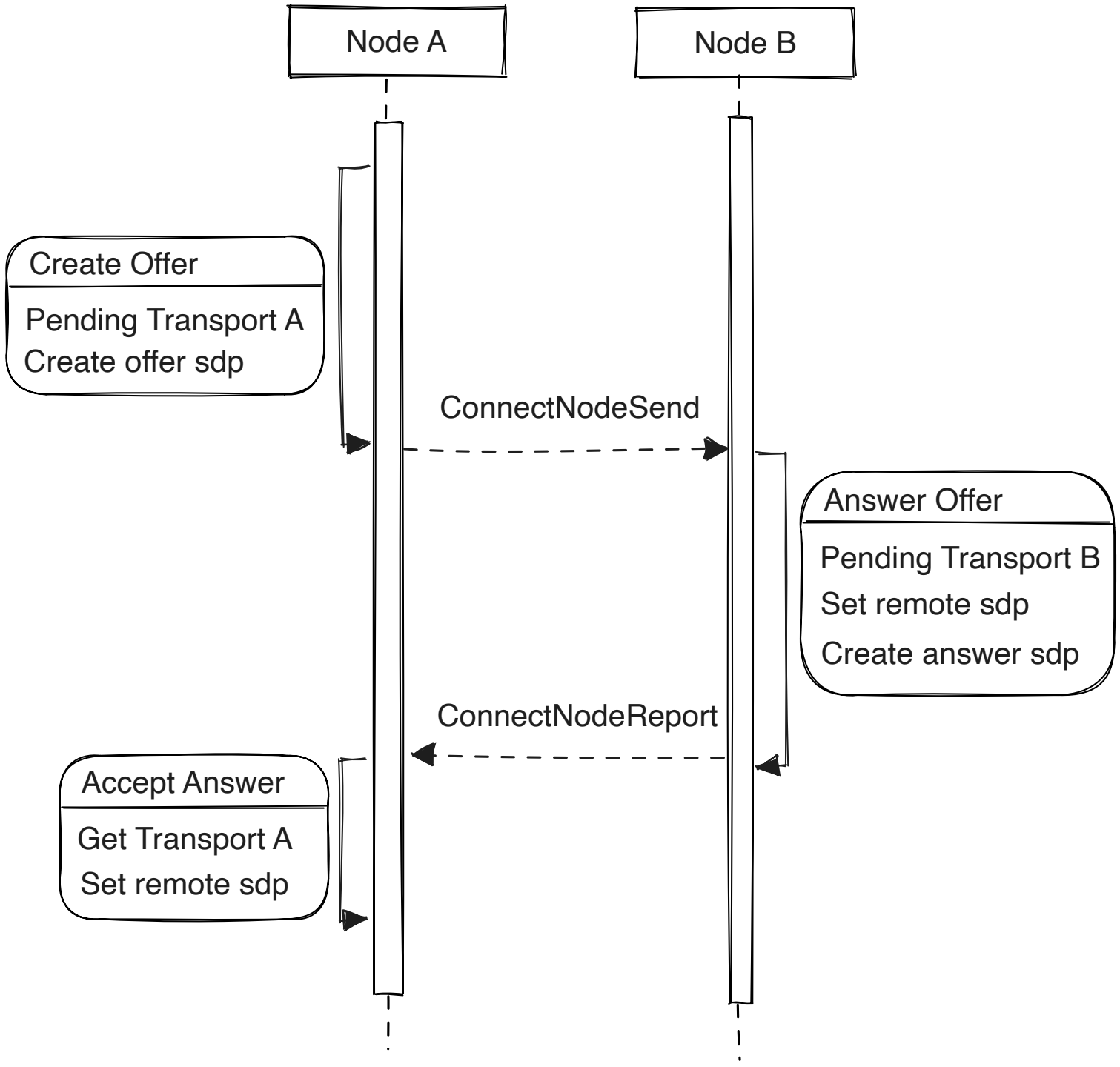
Handshake

- manually
- via DHT
- via HTTP

```rust
pub struct HandshakeInfo {
    pub sdp: String,
    pub candidates: Vec<IceCandidate>,
}

pub struct ConnectNodeSend {
    pub transport_uuid: String,
    pub offer: HandshakeInfo,
}

pub struct ConnectNodeReport {
    pub transport_uuid: String,
    pub answer: HandshakeInfo,
}
```



```rust
/// All messages transmitted in RingsNetwork should be wrapped by MessagePayload.
/// It additionally offer transaction ID, origin did, relay, previous hop verification,
/// and origin verification.
pub struct MessagePayload<T> {
    /// Payload data
    pub data: T,
    /// The transaction ID of payload.
    /// Remote peer should use same tx_id when response.
    pub tx_id: uuid::Uuid,
    /// The did of original sender.
    pub addr: Did,
    /// Guide message passing on rings network.
    pub relay: MessageRelay,
    /// This field hold a signature from a node,
    /// which is used to prove that the message was sent from that node.
    pub verification: MessageVerification,
    /// Same as verification, but the signature was from the original sender.
    pub origin_verification: MessageVerification,
}
```

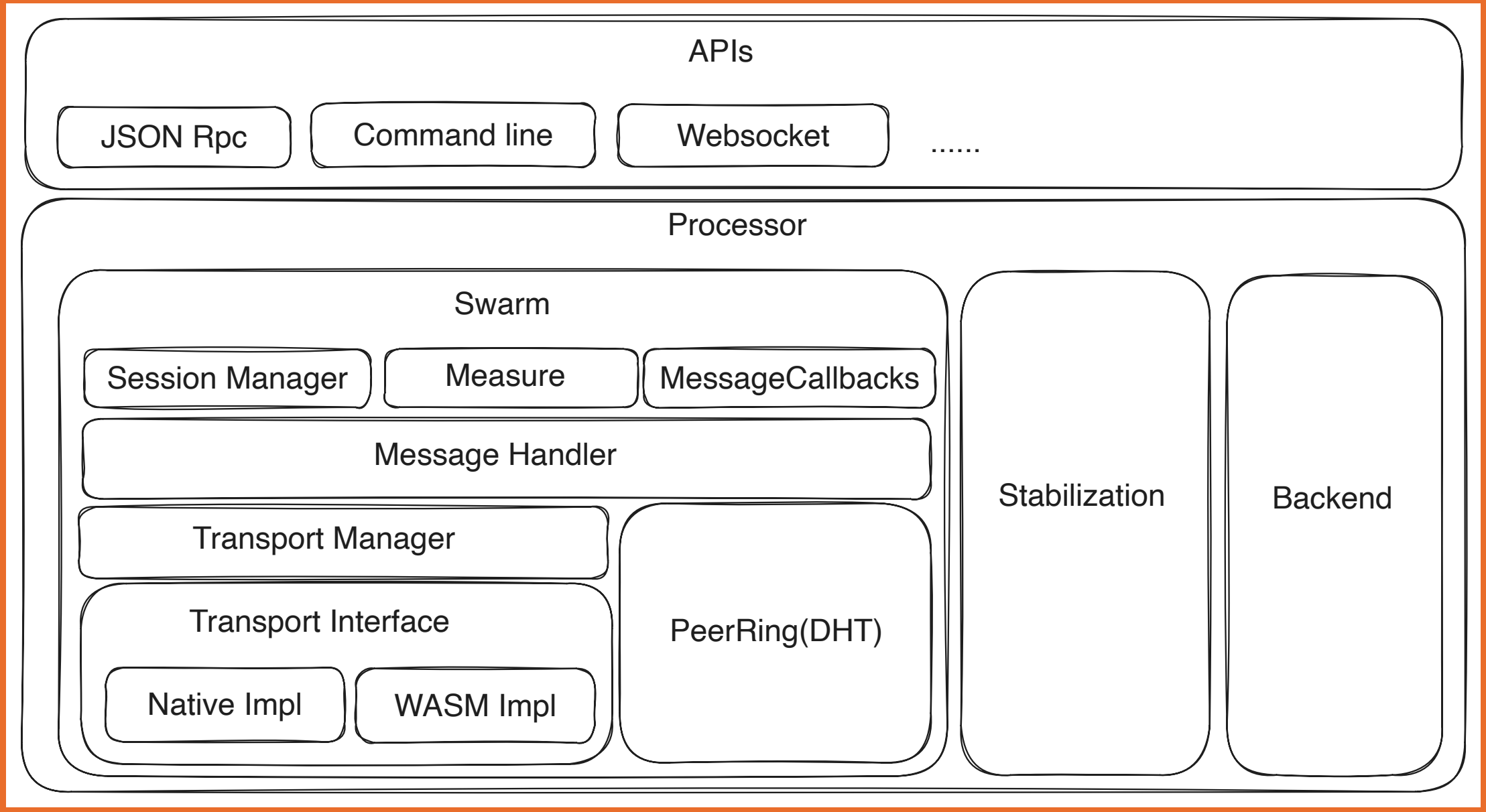
Inbound Message

1. Transport
2. SessionManager.verify
3. PeerRing(DHT)
4. MessageHandler
5. TransportManager/Callback(Backend)

Outbound Message

1. Multiple ways
    - API SendCustomMessage
    - Stabilization Notify
    - Backend Report
    - MessageHandler Forward/Send/Report
2. SessionManager.sign
3. PeerRing(DHT)
4. TransportManager
5. Transport

Arch of rings-node



Stabilization Philosophy

- The rule is formulated by DHT.
- The availability is ensured by redundancy.
- The reality is observed by payloads.

Stabilization Artifacts

- Correct Chord (https://arxiv.org/abs/1502.06461)
- Multiple Successor
- Continuous notification
- Transport events
- Detect peers in payloads (Relay path)

Account abstraction

- No private key access
- All signature algorithms are available
- Union different wallet with a session

End to End Encryption

- Recover public key from session
- ECIES vs ElGamal

The Plugable Hooks for Geeks

- Validator
- MessageCallback
- Native Backend
- WASM Backend (WIP)

Future

- Inside
  - 运行时抽象
  - 节点共识机制
  - 大规模节点集成测试
- Outside
  - 跨平台的 WASM Backend
  - Backend 开发文档

Contact Us

- Twitter: @ringsnetworkio
- Email: contact@ringsnetwork.io
- Website: https://ringsnetwork.io

Thank you !