区块链技术与应用

(2024年春季)

计算机科学与技术学院　李京

8章　区块链跨链技术

目 录

8.1 概念与定义

跨链技术-互操作性

互联网三大基本目标为：

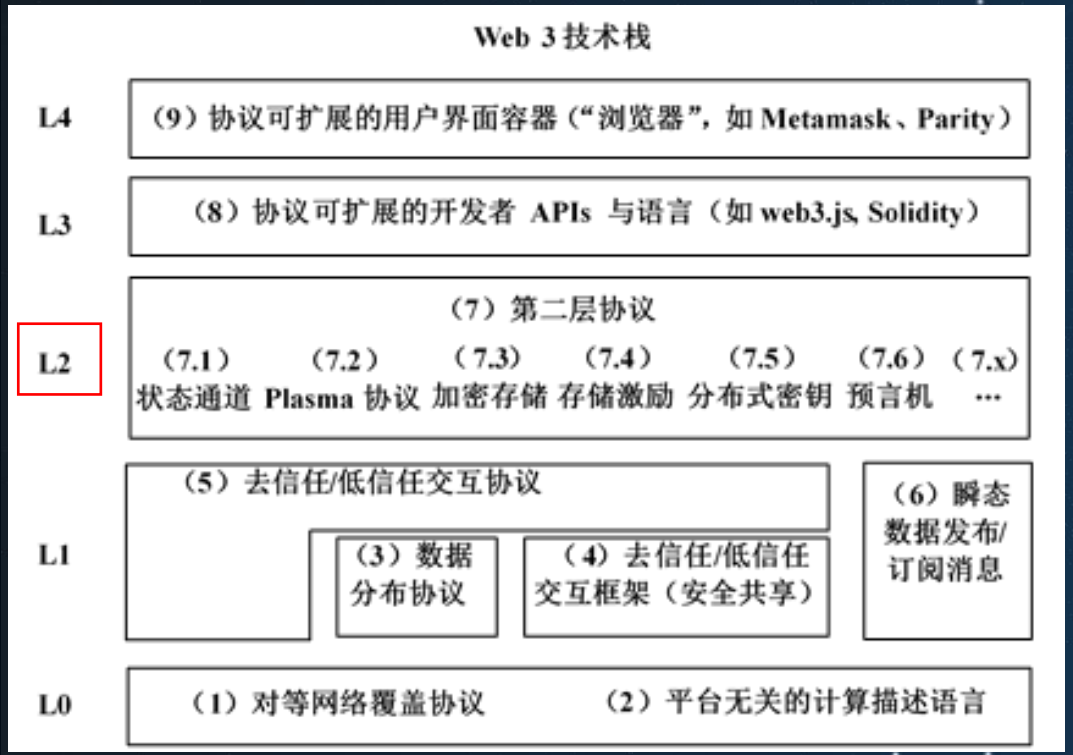
1) 可生存性（Survivability）：尽管网络或网关受损，互联网通信仍必须能够继续进行；

2) 服务类型的多样性（Varieties of service types）：互联网必须支持多种类型的通信服务；

3) 网络的多样性（Varieties of networks）：互联网必须可以承载各种各样的网络。

- 如果区块链系统要成为未来全球经济的重要基础设施，成为全球商业与价值分布式网络，那么其体系结构也必须满足互联网的基本目标，区块链系统间互操作性将是一个核心需求。
- 一个可互操作的区块链架构应具备的特征：可互操作的区块链体系结构是可区分的区块链系统的组合，每个区块链系统代表一个分布式数据账本，其中交易执行可能跨越多个区块链系统，并且其中记录在一个区块链中的数据可以通过语义兼容的方式被另一种可能来自外部的交易访问和验证。

跨链技术所在层级



- Layer0层是数据传输层，主要指P2P网络和传播机制，侧重区块链与传统网络的结合。
- Layer1层是On-Chain层，侧重底层账本公链自身，包括数据存储协议（如IPFS）、去信任协议（如比特币）等。
- Layer2层是Off-Chain层，链下的意思是脱离公链，侧重扩展性延伸和链上链下打通。
- Layer3层主要包括开发API和语言，如Web3.js、Solidity等。
- Layer4层主要包括协议可扩展的用户接口，如Metamask、Parity等。

跨链技术-互操作性可能满足的场景

便携式资产（Portable assets）：即资产转移场景，数字加密货币或资产可以在不同链之间来回转移。

银货两讫（Payment-versus-delivery）：即一手交钱一手交货，强调链间资产的同时交换，即原子互换（atomic swap）概念。同步交收（payment-versus-payment）也是类似意思。

跨链预言机（Cross-chain oracles）：链A上的智能合约的触发和执行依赖另一条链B上的预言机的证据，即具备他链信息或事件的读取与验证能力。

资产留置（Asset encumbrance）：链A上的资产被锁定，解锁条件取决于链B的行为。如金融衍生品的抵押品，破产追回，法院命令和涉及保证金的各种场景。

通用跨链合约（General cross-chain contracts）：根据链A上的资产证明在链B上分发股息。

跨链技术发展历史


2012年Ripple致力建立适用于所有记账系统的协议
2014年，BlockStream团队首次提出锚定式侧链
2015年10月，Ripple公司进一步引入了一种跨链价值传输的Inter-Ledger Protocol
2016年9月，Vitalik Buterin为R3区块链联盟写了一份关于跨链互操作的报告
闪电网络提出基于微支付通道构建跨链方案
BTC-Relay基于中继跨链方式实现从比特币到以太坊单向流通
万维链则利用多方计算和门限密钥共享方案，实现公有链间的跨链交易
以Polkadot和Cosmos为代表的跨链技术，更多关注的是跨链基础设施建设

跨链的定义

- 跨链是指实现区块链账本之间资产的互操作，即在可以引入第三方但不改变原生链的前提下实现区块链之间资产的互换、转移。
  - 资产互换：通常指发生在两条链之间不同用户间的资产互换。
  - 资产转移：通常指发生在两条链之间单用户的资产迁移，可以分为单向或者双向。

8.2 难点与解决方案

- 跨链交易提供了一种链间清算机制，清算的本质就是精确记账，因此跨链传递的不仅是信息流，更在于其背后对应的需要精确记录的价值。
- 实现跨链交易首要需要解决两个难题：
  - 一是如何实现对交易的确认。跨链交易状态信息的获取、交易的等待和取消
  - 二是如何保证交易的原子性。

难点与解决方案-难点一：如何实现对交易的确认

根据通过“中间人”传输和验证交易信息的方式可以分为三种方案：

- 公证人机制：一般是“可信第三方”，较通用与成熟的模式
  - 单签名公证人机制
  - 多签名公证人机制
  - 分布式签名公证人机制
- 中继机制：负责交易相关数据的收集与转发
  - 在跨链中，中继机制不依赖可信的第三方帮助其进行交易验证
  - “中间人”仅仅负责交易相关数据的收集与转发，目标链可以在拿到发送链的数据后自行验证
  - 整个过程中，“中间人”更多体现的是桥接的功能
  - 中继方案松耦合、更加灵活且易于扩展，具有多种实现形式，如Cosmos中的Hub、Polkadot中的Relay Chain等
- 侧链机制：一种强耦合结构的跨链模式
  - 侧链被定义为可以验证来自其他区块链数据的区块链；
  - 主链可以不知道侧链的存在，但侧链必须知道主链；
  - 侧链一般基于SPV证明验证数据，需要同步所有的区块头；
  - 侧链受到主链的技术限制较多，可认为是一种强耦合结构的跨链模式

难点一：如何实现对交易的确认

- 应对交易可能撤销的场景的常见方案如下：
  - 等待足够多的确认，这种方案不足在于拉长处理周期；
  - 区块纠缠，令两个链之间建立依赖关系，当一个链上区块被撤销时，级联撤销关联链上相关区块；
  - 使用强一致性的共识算法，如DPoS/PBFT等。

难点与解决方案-难点二：如何保证交易的原子性

- 原子互换（Atomic Swaps）
- 哈希时间锁定合约
- 哈希时间锁定协议

原子互换

- 原子互换是保证交易原子性的最重要的基本概念
  - 原子跨链交换是一种实现多方跨多个区块链交换资产的分布式协调任务。
  - 原子性是计算机领域非常重要的概念，原子操作是不可分割的，在执行完毕之前不会被任何其它任务或事件中断，整个操作要么成功、要么失败，不存在中间状态。
  - 原子互换以去中心化的方式实现资产交易，在点对点的基础上实现两种加密货币的交换，无需第三方介入，也不存在交易一方在交易中违约的风险。

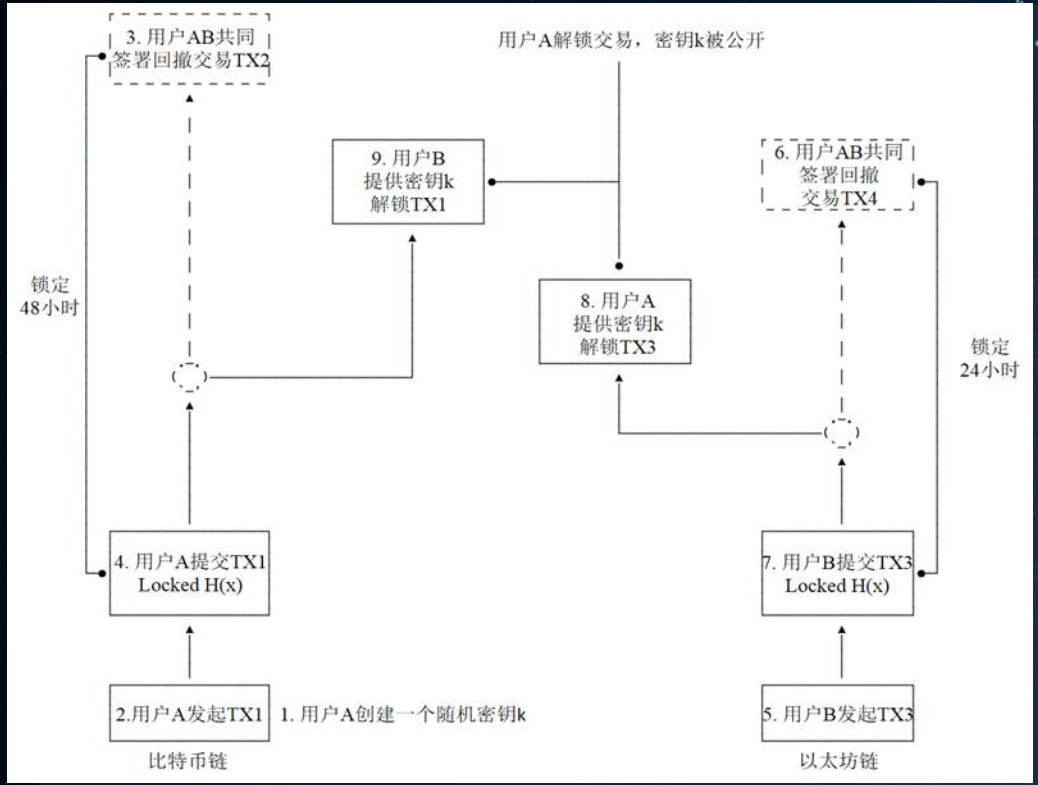
具体实现

场景：两人想要在不通过中心化交易所的方式进行比特币与以太币的交换，在兑换比率达成一致后，用户A与B即可互换。然而，由于区块链上交易不可逆转，如果A先发送B比特币，于A不利，因为并不确定B会发送他以太币。为了能使这种场景的交易履约进行，需要设计一种机制能够确保A和B都不会违背交易。

整个过程的关键在于用户A和用户B商定一个“定时智能合约(Timed Smart Contracts)”并先后锁定待转账的资产，定时智能合约的约定如下：

(1) 条件a：如果有人能在T小时内向智能合约输入随机密钥k’，并且能够验证Hash(k’)==m，那么用户B锁定的以太币将发送给用户A，超时则将以太币返还用户B；

(2) 条件b：如果有人能在2T小时内将原始密码k发送给智能合约，则用户A的比特币将自动转给用户B，否则返还给用户A。



(1) 用户A创建一个随机密钥k，该密钥只有用户A知道。

(2) 用户A在比特币链上创建交易TX1：“Pay w BTC to <B，s public key> if (k for Hash(k) known and signed by B)”。

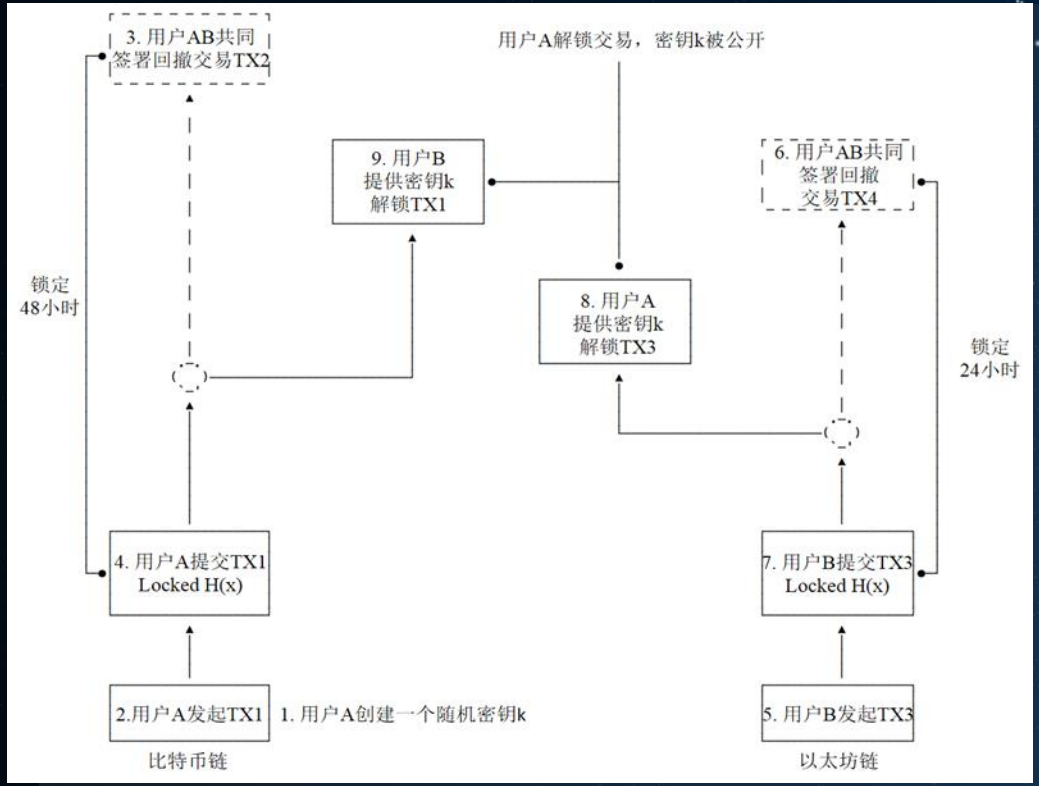
(3) 在TX1广播之前，用户A先在比特币链上广播一个回撤交易TX2：“Pay w BTC from TX1 to VA‘s public key>,locked 48 hours in the future,signed by A”。TX2需要双方共同签名，才能生效。用户B同意交易，便签署TX2，并返回给用户A。

(4) 用户A在比特币链上提交TX1，向全网广播。

(5) 用户B在以太坊币链上创建交易TX3：“Pay v ETH to <A-public-key> if (k for Hash(k) known and signed by A)”。

用户A发起了向用户B转w个比特币的交易

用户B发起了向用户A转v个以太币的交易



(6) 同样，TX3广播之前，用户B先在以太坊链上广播一个需要双方共同签名的回撤交易 TX4: “Pay v ETH from TX3 to <B’s public key>, locked 24 hours in the future, signed by B”，即如果24小时内未有人解锁TX3，那么将w以太币返还给用户B。用户A看到用户B发起的TX4,附上自己的签名，返回给用户B。

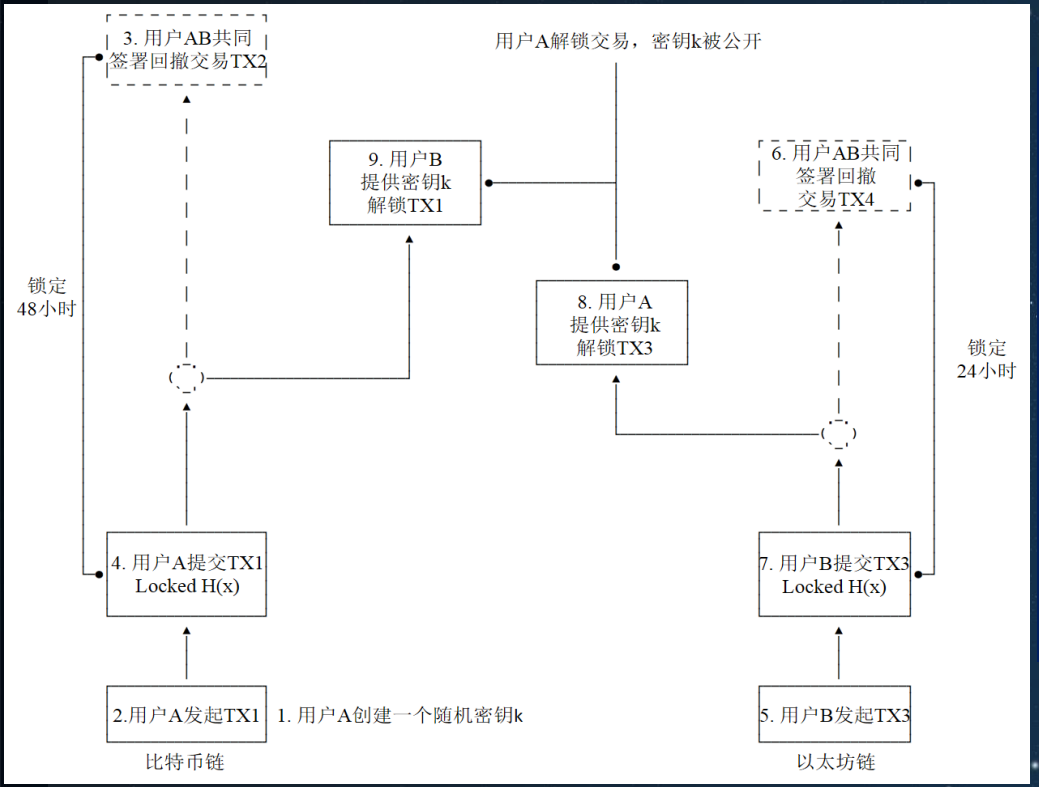
(7) 用户B在以太币链上提交TX3，向全网广播。

(8) 用户A为了获得v个以太币，便在以太币链上提供密钥k，并附上自己的签名以解锁TX3，交易成功后，用户A获得v个以太币，用户B也知晓密钥k。

(9) 用户B利用密钥k与自己的签名在比特币链上解锁TX1，最终获得用户A的w个比特币。

用户A发起了向用户B转w个比特币的交易

用户B发起了向用户A转v个以太币的交易



用户A发起了向用户B转w个比特币的交易

用户B发起了向用户A转v个以太币的交易

依照流程，交易可以成功完成，再分析一下其他情况：

(1) 如果用户A始终不提供密钥k，那么超时后锁定资产返回原所有者。

(2) 如果用户A在T至2T时段提供密钥k，那么用户B不仅会获得用户A的资产，原本锁定的资产也会返回。

(3) 如果用户B未在T至2T时段提供密钥k，那么用户B会丢失自己的以太币，而且也拿不到用户A的比特币。

情况2、3都包含着强约束，即“时间限制”和“强制执行交易”的机制迫使用户A、B理性选择，交易原子性可以获得保证。

哈希时间锁定合约

- 哈希时间锁定合约约(Hashed Timelock Contract,简称HTLC)可以看做是原子互换的一种具体实现。
- 哈希锁定Hashlock
  - 原子互换实例中提及的交易条件中出现了Hash函数，这种函数是单向
  - 在比特币系统中，哈希计算操作通常用OP_SHA256或OP_HASH160来实现
- 时间锁定Timelock
  - OP_CHECKLOCKTIMEVERIFY，该操作码通常简称为CLTV，是在BIP-0065提出的，将特定事务冻结到将来的某个特定点，可以是时间戳或者块高度。
  - OP_CHECKSEQUENCEVERIFY，该操作码通常简称为CSV，是在BIP-0112提出的，锁定的是相对时间，例如：一年之后币可用。

哈希时间锁定协议

- 哈希时间锁定协议议(Hashed Timelock Agreement，HTLA)可以看作是HTLC概念的泛化
  - 哈希时间锁定协议，可以用来在不支持HTLC的账本间执行HTLC，跨的不仅是链，中心化或者去中心化账本都支持；
  - Interledger中应用了该理念，支持的账本不仅包括区块链，还有银行、金融相关的各类传统中心化账本；
  - 在付款方和收款方中间，起到核心作用的是一系列的连接器（Connector）；
  - 付款方和连接器之间，连接器之间，连接器和收款方之间都是通过哈希时间锁定的概念来完成有条件的转帐，并且可以扩展到支持多跳支付。

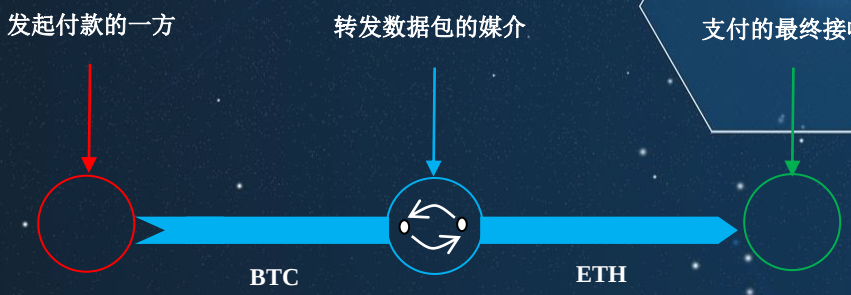
方案比较

	公证人机制（Notaries）	中继（Relays）	哈希锁定机制（Hash-locking）
互操作支持类型	全部	全部（需要两条链上都有中继，否则仅支持单向的）	只支持交叉依赖
信任模型	大多数公证人诚实	链不宕机或者遭受“51%攻击”	链不宕机或者遭受“51%攻击”
支持跨链资产互换	是	是	是
支持跨链资产转移	是（但是要求共同的、长期公证人可信）	是	否
支持跨链预言机	是	是	不直接支持
支持跨链资产质押	是（但是要求长期公证人可信）	是	多数情况下支持，但是有难度
实现难度	中	高	低

维塔利克-布特林(Vitalik Buterin)提出了这三种跨链技术

8.3 典型跨链案例-Interledger Protocol

- Interledger Protocol：公证人机制的典型代表，该协议规定了Sender（发起付款的一方）、Connector（转发数据包，介于发送者和接收者之间的媒介）以及Receiver（支付的最终接收者）三种角色，以及数据交换的顺序和内容。



ILP协议中3种角色示意图

Interledger Protocol核心内容

协议簇

- Interledger Payment Request Protocol
- Pre-Shared Key Transport Protocol
- Simple Payment Setup Protocol
- Interledger Quoting Protocol
- Connector-To-Connector Protocol

地址和路由

- 地址是由“.”字符分隔的段组成的分层结构字符串，这是在ILP上识别账号的机制
- 还有一种地址称为“地址前缀”一种模糊匹配的模式
- Connector维护整个网络的路由表，当连接器接受查询时，会按照最长前缀匹配项原则基于若干连接器的路由表递归查找

报价

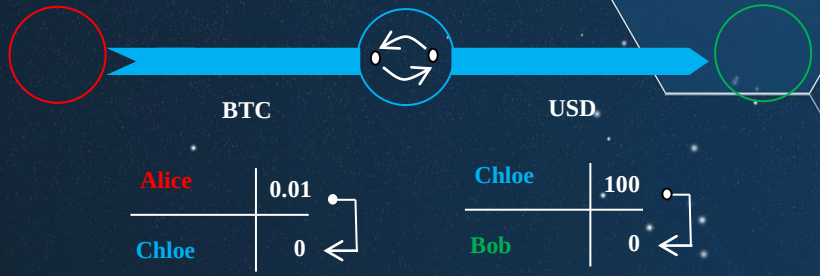
- 主要是估算向连接器支付的费用，即转账手续费，Interledger Quoting协议详细规定了相关估算方法

哈希时间锁定协议

- 哈希时间锁定协议上文已经提及，ILP中依据资金托管设置的条件和时间限制来保障交易原子性的。该协议采用密码算法用Connector为这两个记账系统创建资金托管，当所有参与方对交易达成共识时，便可相互交易

22

- 下面以经典的Alice与Bob之间的支付场景为例，Alice是Sender，在区块链上有一个实现HTLCs的账户，Bob是Receiver在银行有一个不实现HTLCs的账户，Chloe是Connector。Alice有比特币账户，Bob有美元账户，Chloe拥有比特币账户和美元账户。Alice只有比特币，想给Bob转账，但Bob只接受美元。



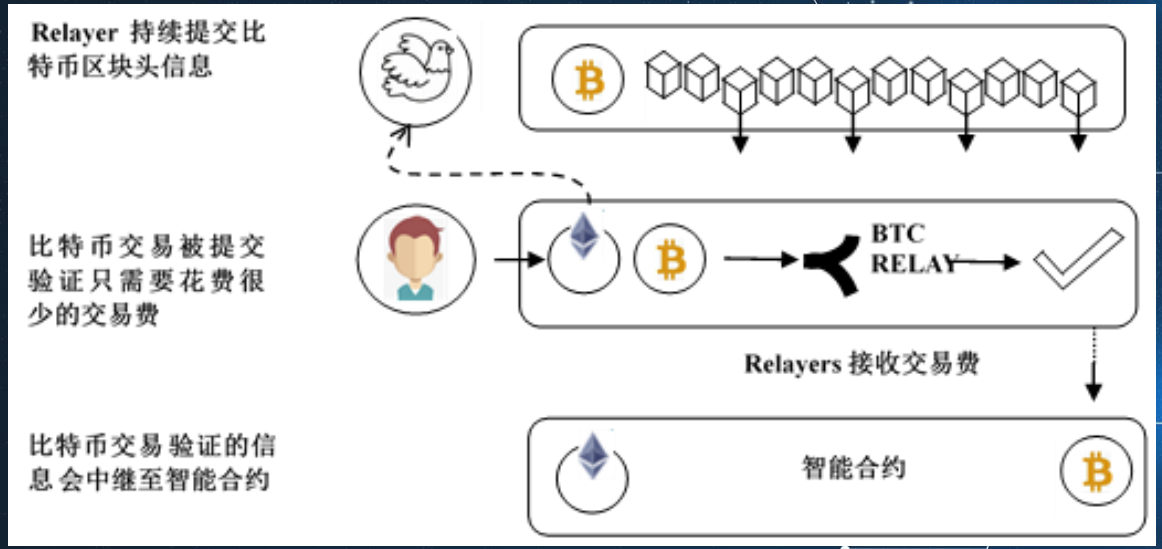
- 通过Interledger转账流程如下所示：
- (1) Alice和Bob商议一个共享密钥（也许基于Pre-Shared Key Transport Protocol）。
- (2)Chloe也许是一个流动性提供商，Alice向Chloe咨询比特币与美元之间的汇率，假设是0.01：100，同时Chloe收取一定手续费，最终Alice获知需要向Chloe支付0.01000001个比特币。
- (3)Alice构建ILP数据包，目标地址为Chloe，数量是0.01000001个比特币，并附上基于共享密钥生成托管条件以及超时时间，并在比特币账本系统发起“托管”操作。
- (4)Chloe监测到涉及自己的“托管”操作，解析获知自己需向Bob转100美元，因此，将ILP数据包中目标地址改为Bob。
- (5)Chloe基于与Bob共享的trustline发起一个“托管”操作，设置了步骤3中的“托管”条件以及一个超时时间（要求小于步骤3中的超时时间）。
- (6)Bob监测到涉及自己的“托管”操作，在设定超时时间之前提供出“共享密钥”，以通过“托管”操作携带的“条件”。
- (7)Bob确定后在trustline上发起一个“托管”确认操作，附上共享密钥，trustline上的“托管”交易完成，Bob获得100美元。
- (8)Chloe监测到涉及自己的“托管”确认操作，解析后获得共享密钥。
- (9)Chloe在比特币账本系统发起一个“托管”确认操作，附上共享密钥，比特币账本系统上的“托管”交易完成，Chloe获得0.01000001个比特币。

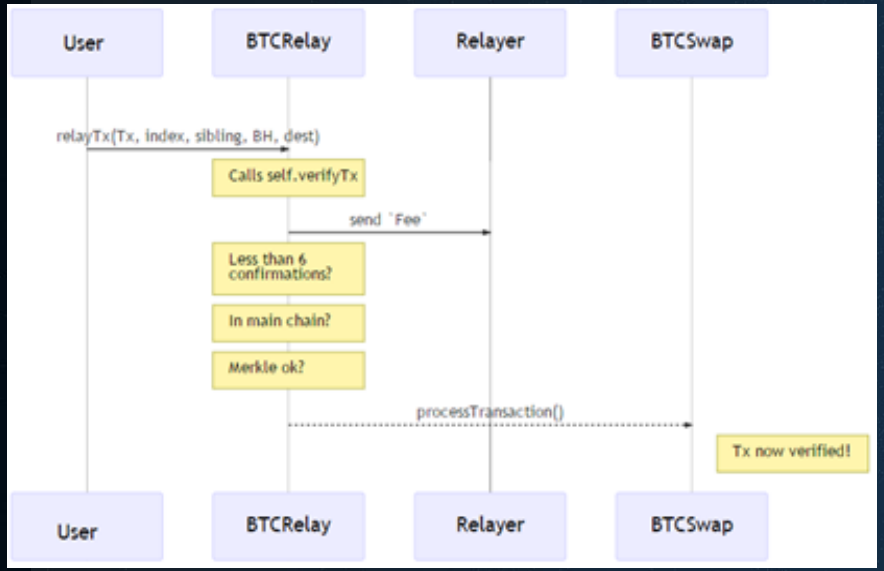
多跳间接跨账本交易

- 对于没有直接支付通道的两个账本系统，还可以通过多跳间接跨账本交易

BTC-Relay：侧链机制的典型代表

- BTC Relay的本质其实是以太坊的一个智能合约，扮演了预言机这样的角色



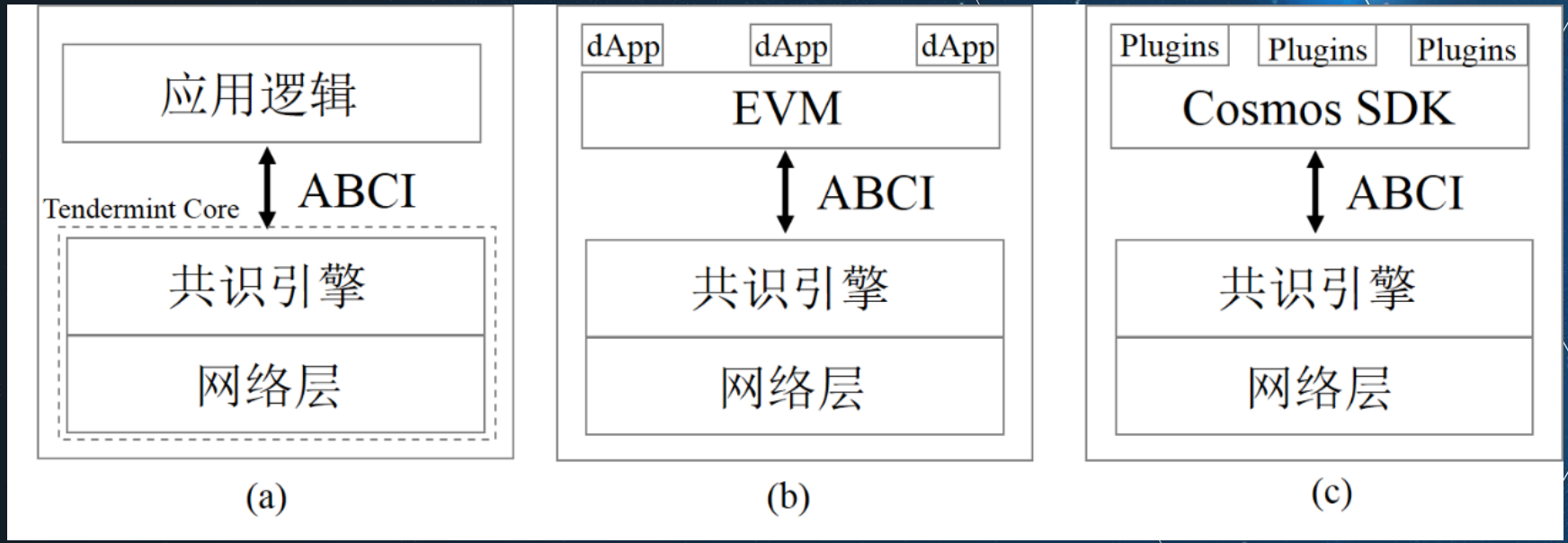


场景是Alice有BTC，想从Bob那儿换点ETH，因此两人在以太坊区块链上构建一个“BTCSwap”的智能合约，Bob将他的以太币发送给“BTCSwap”合约，并将之锁定。

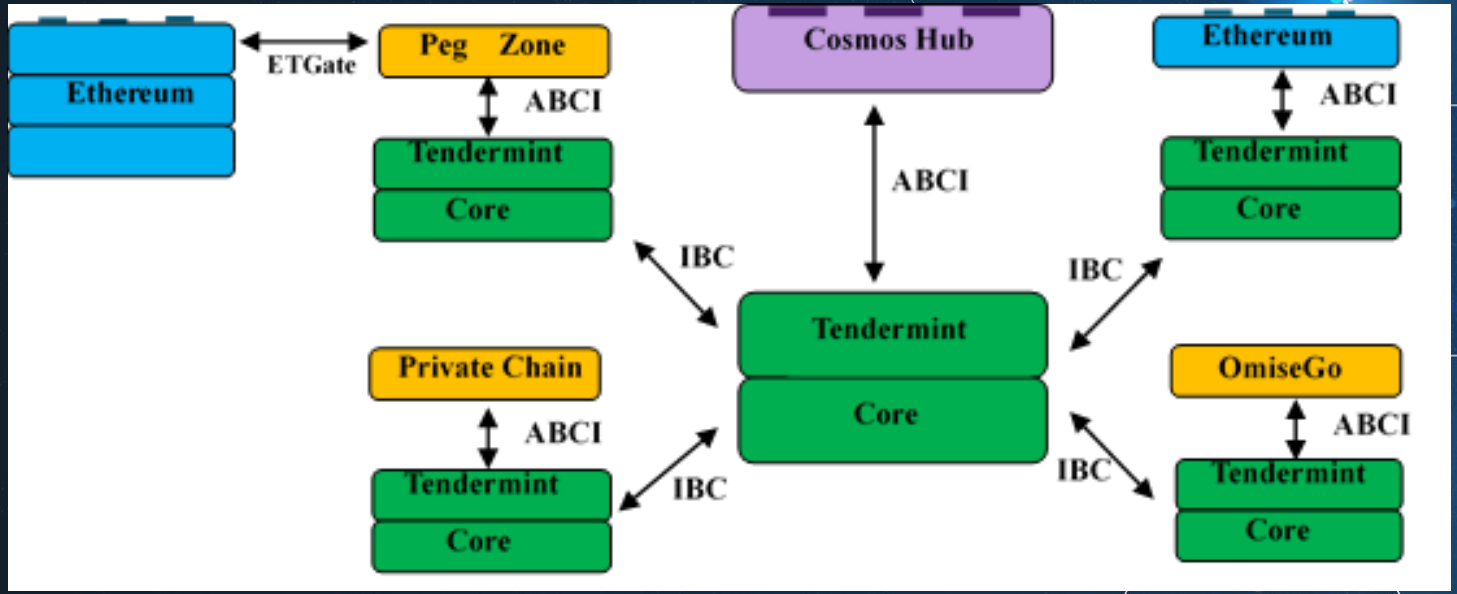
1) Alice在比特币区块链上将BTC发送给Bob，同时，她希望“BTCSwap”这个合约能够知晓这笔交易，并将Bob冻结的ETH转给她；
2) Alice基于比特币上的交易信息与“BTCSwap”合约地址来调用BTCRelay.relayTx()；
3) BTCRelay首先调用verifyTx方法验证比特币交易有效性；
4) 调用BTCRelay验证比特币，需要向提供相关比特币区块头的Relayer支付一些手续费；
5) 一旦通过“区块是否已确认，区块是否在主链”等规则验证，将触发BTCSwap合约里面的processTransaction方法；
6) BTCSwap合约被触发后首先确认这个BTCRelay的合法性，通过后将解冻之前Bob的ETH，整个交易完成。

典型跨链案例- Cosmos

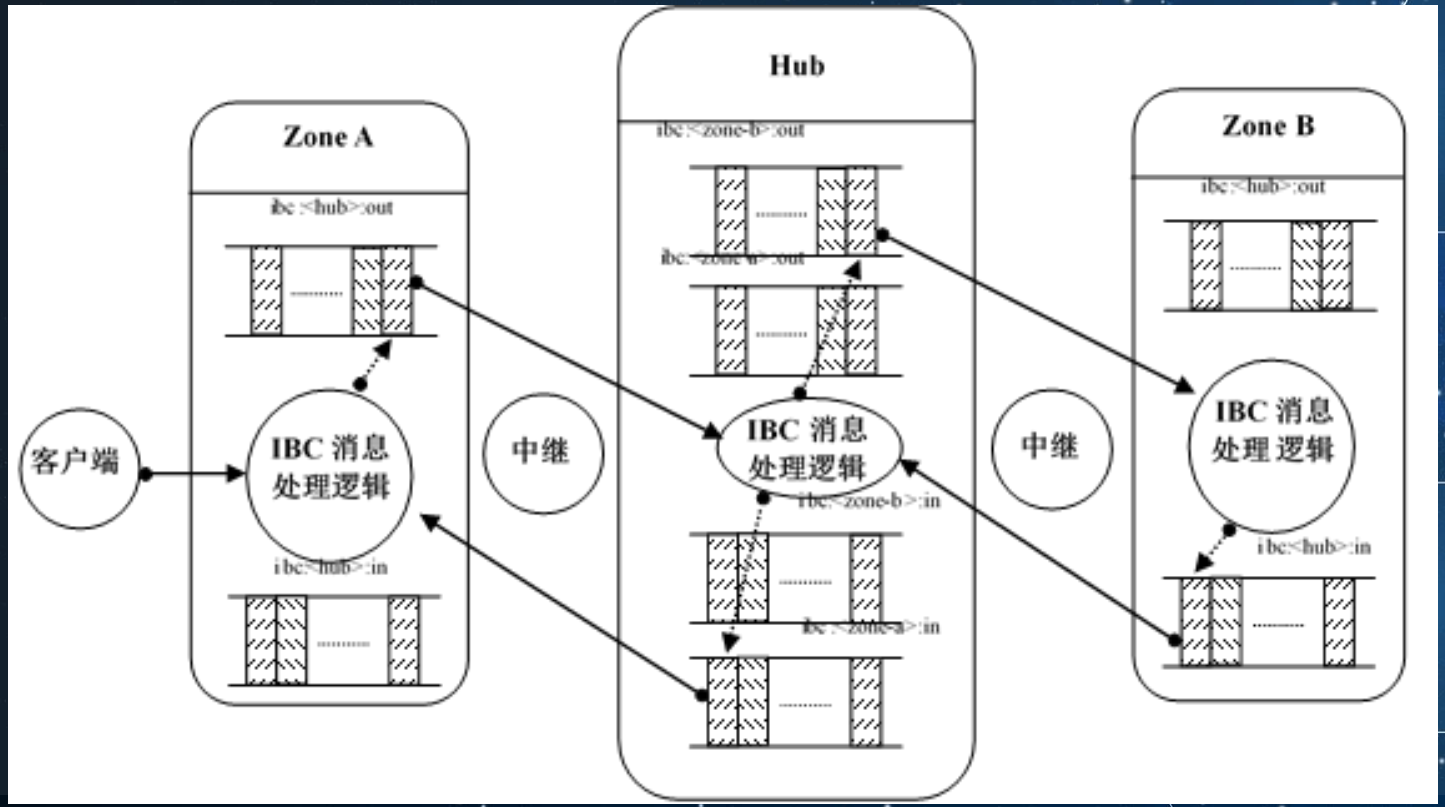
- Cosmos：Tendermint团队发起的一个公有链项目，旨在构建“区块链的互联网”。



Cosmos生态系统

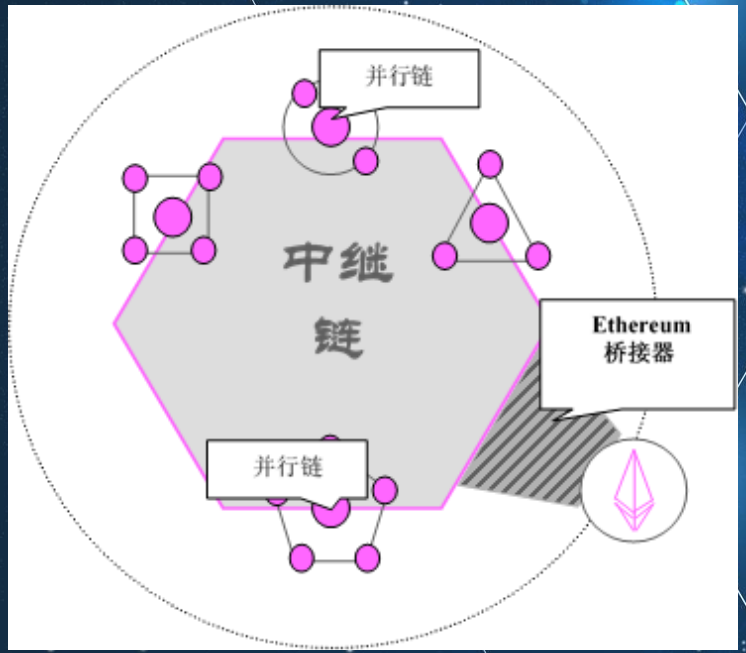

Ethereum
ETGate
Peg Zone
ABCI
Tendermint
Core
Cosmos Hub
ABCI
Tendermint
Core
Ethereum
ABCI
Tendermint
Core
IBC
IBC
Private Chain
ABCI
Tendermint
Core
IBC
IBC
OmiseGo
ABCI
Tendermint
Core

Cosmos上跨链交易示意图

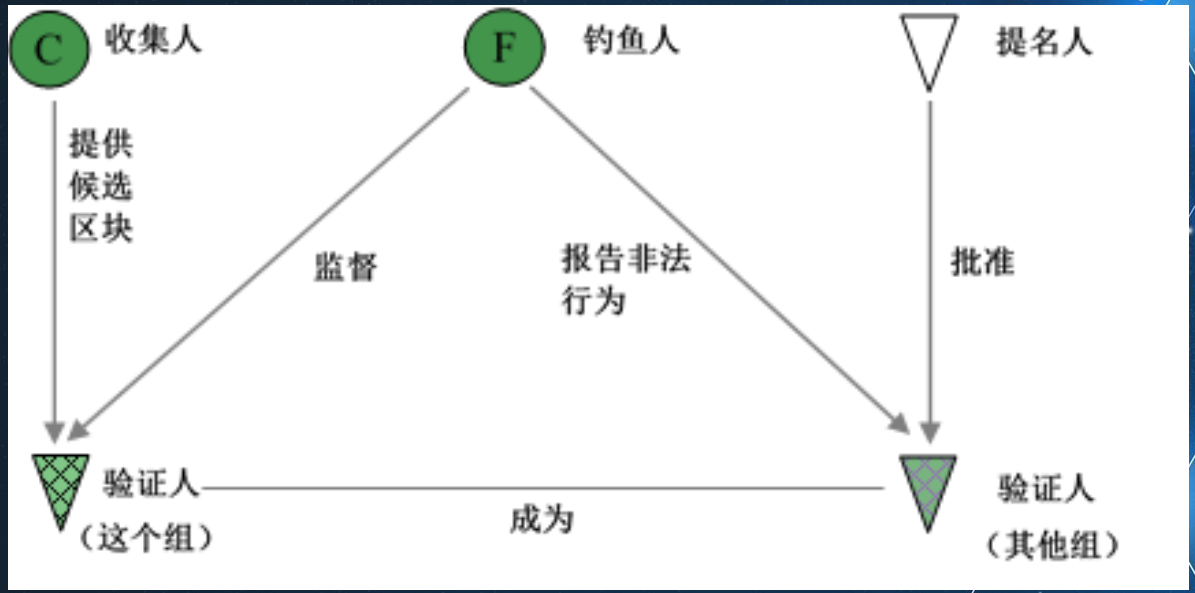


典型跨链案例-Polkadot

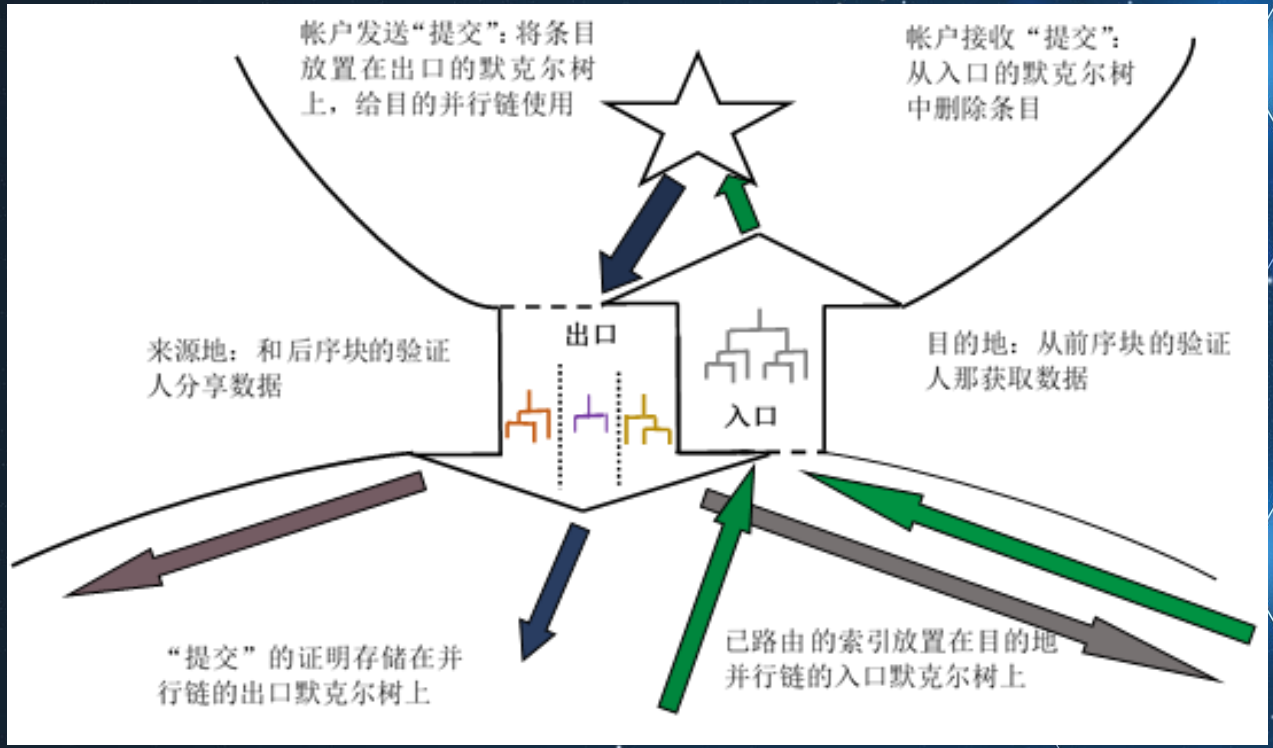
- Polkadot：是Web3基金会支持，由以太坊前任CTO Gavin Wood主导的团队（开发以太坊钱包Parity的团队）研发的跨链项目。



Polkadot中4种角色交互示意图

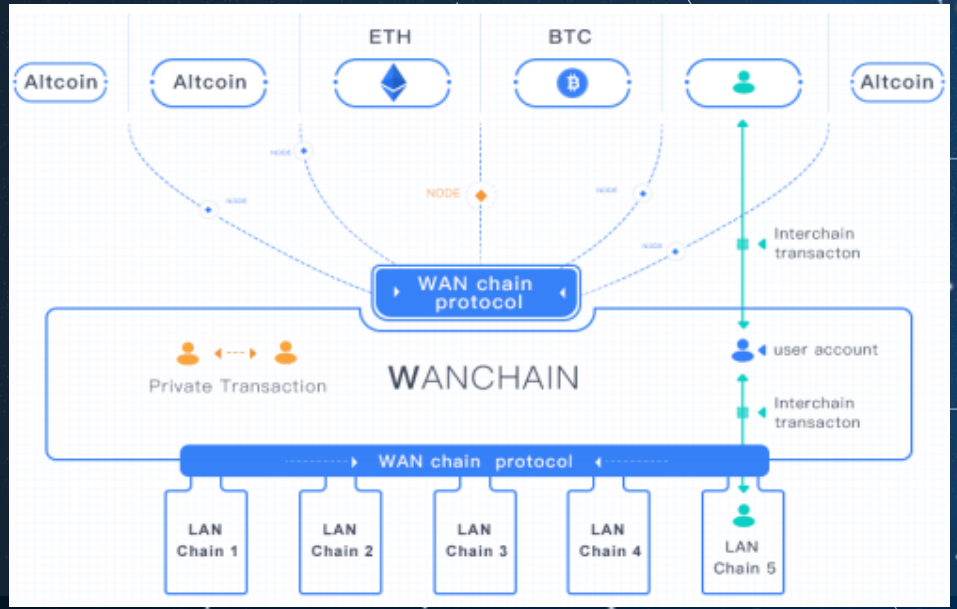


Polkadot上跨链交易示意图



典型跨链案例-Wanchain

- Wanchain ： 是一条条去适配现有的异构链，将其纳入Wanchain平台，Wanchain 3.0已打通了比特币、以太坊之间的跨链交易。



- Wanchain ：万维链采用的是PoS共识，万币（WANCoin）是万维链的原生币，总量为2.1亿个。
- 万维链设想的跨链场景是这样的：假设Alice要将1个BTC转移至万维链，那么首先她需要在比特币链上发送一个BTC到锁定账户；万维链验证完毕后，万维链上Alice的账户会生成等值的锚定币WBTC（即万维链上特有的BTC），Alice可以使用锚定币在万维链上流通，比如在去中心化交易所买Bob的10个WETH（ETH的锚定币），WETH也可以转至Alice以太坊上的账户，变成ETH。一旦资产返回原链，WBTC或WETH会被销毁。不同公链的锚定币可以与万币兑换，支持者越多，意味着万维链接入应用越广。

本章参考书

3.2节 跨链技术

谢谢！