原子性问题解决方案

假如你要操作一笔交易，但是你不能百分百确认交易是否完全执行并发送到目的地，而且可能只有部分操作被记录，也可能某一方会丢钱。如果出现诸如此类的情况，相信应该没有人会再用这个网络了。

电源故障或事件故障可能会给数据库造成严重影响。为了保证有效性，每个数据库事务要满足四大标准，也就是所谓的ACID模型，即原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）以及持久性（Durability）。本文主要围绕原子性展开。

首先，什么是原子交易？原子交易是指要么完全执行要么完全不执行的数据库事务。那么，为什么要使用这些事务呢？

先来看下我们所处理的问题，以分布式数据库为例。假设我们想要更改一些数据，而数据库不支持原子更改，那么可能会导致一部分数据与另一部分不一致。如果某位用户不知道有过更改，他会看到替代数据，然后无法确定哪些数据是正确的。但是，如果数据库支持原子交易，那么任何更改都只能通过两种方式执行：要么全改要么全不改。下面我们会深入研究最新的分布式数据库所遇到的问题，然后分析链下实现方案。

传统解决方案

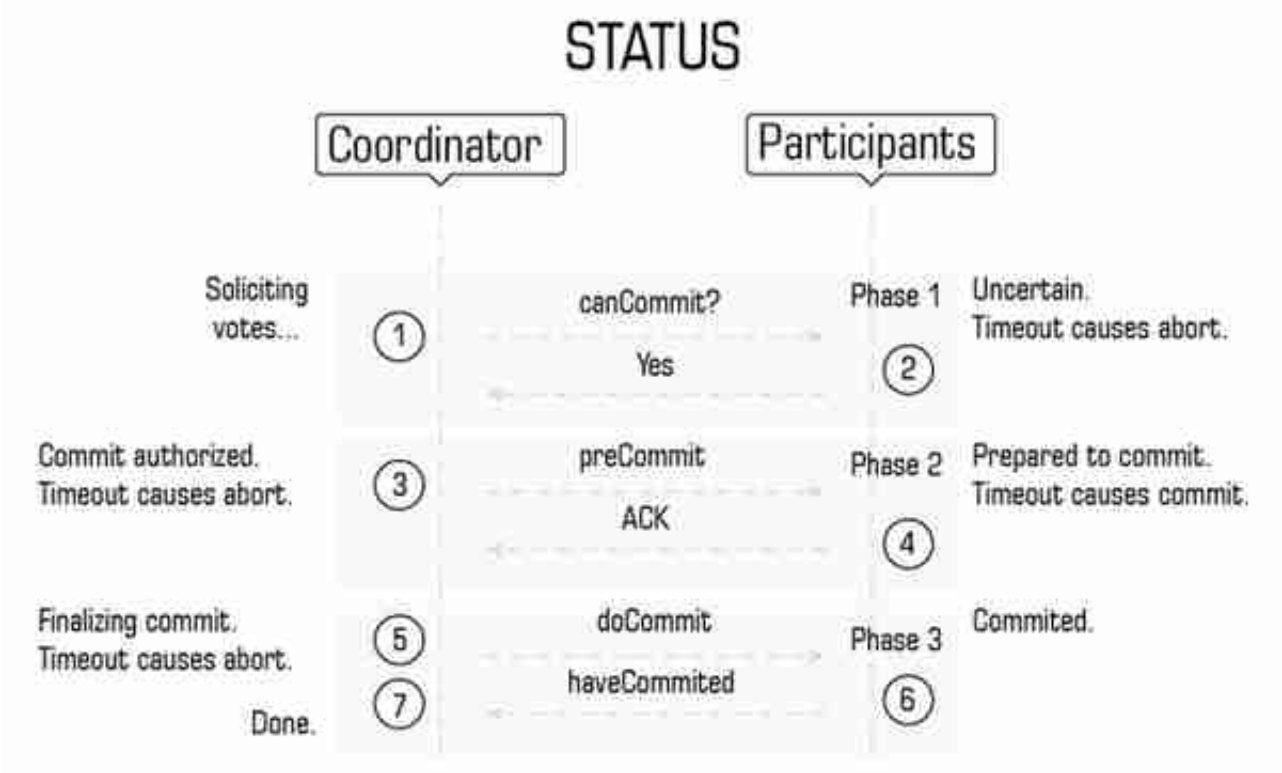
考虑原子性的话，多阶段提交是目前使用较多的。两阶段提交协议是最常见的，而一阶段和三阶段提交协议也广为接受。

一阶段提交是实现原子性最直接的方法，但也是较粗糙和低效的方法。更改由事务管理器发出、参与者执行指令来完成。显然，这个模型有太多固有威胁和陷阱，比如参与者可能会掉线，然后回来的时候就错过了提交。

两阶段提交更全面些，它将每个事务分成两个阶段。首先，事务管理器会查询每个参与者以确定是否提交事务。他们会创建必要的临时项（在多跳支付系统的情况下会进行分配）并投票提交。当管理器收到所有参与者回复“是，我已准备好付款”，它会向他们发出提交请求，但只要有一个人回答“否”或没回复，管理器都会撤销付款。这个方案比一阶段提交更稳健、更安全，但也不是无懈可击。实际上，如果有参与者拒绝该事务，那么这一项（分配）就会停留在那里，在管理器发出回滚指令之前都无法删除，从而导致网络锁定。

为了解决这个问题，三阶段提交方案面世，它将第一阶段分为两部分。事务管理器

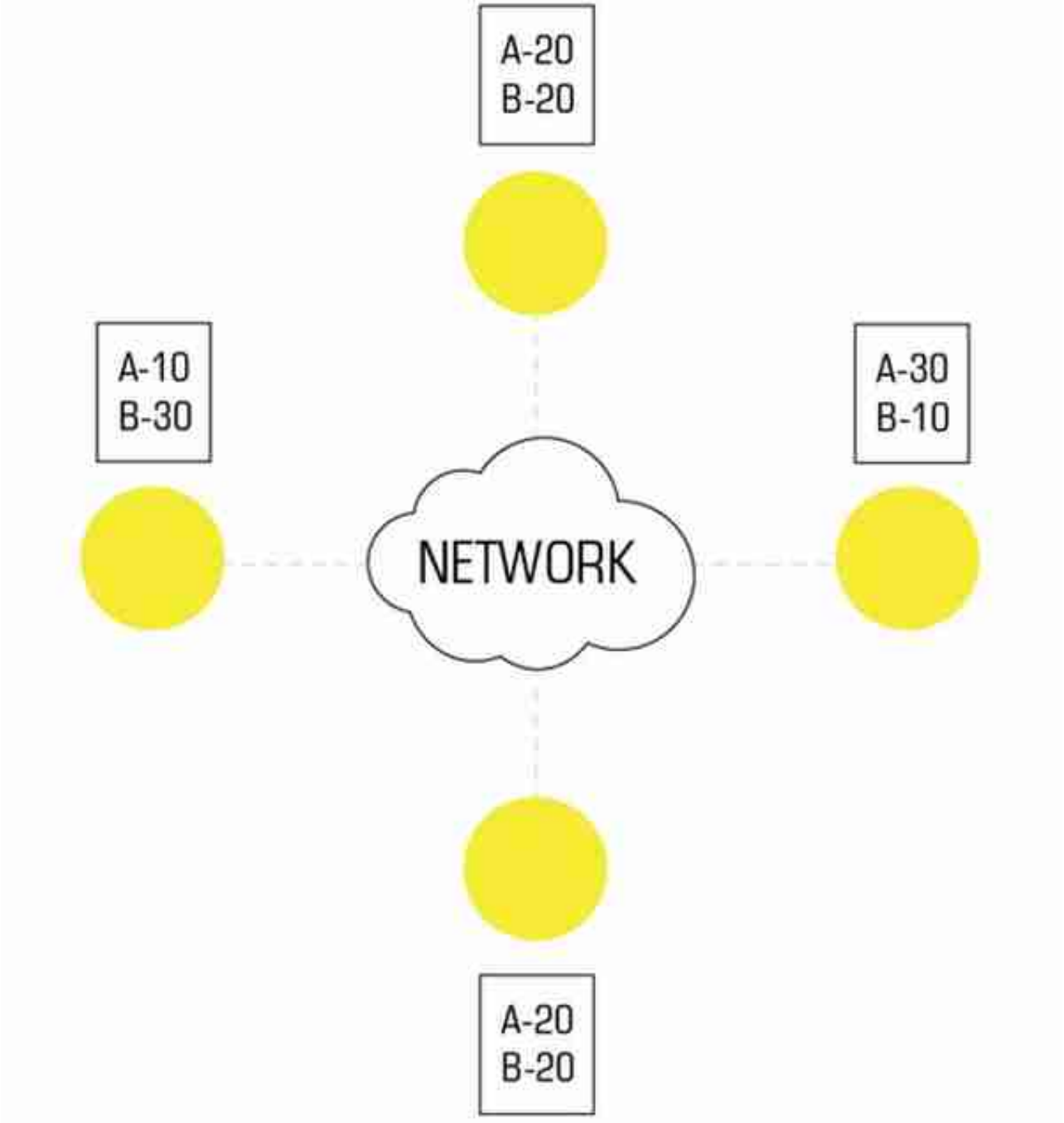
也是从查询参与者的投票开始，但会推迟预备指令，直到收到所有参与者的肯定回复。然后参与者创建项（即进行分配）并确认他们已为下一阶段做好准备。
与两阶段提交一样，最后阶段仅在收到所有确认后执行。



三阶段提交具体步骤

虽然三阶段提交安全级别更高，但由于阶段多，要交换的信息多，它的表现也没有那么好。所以就这些传统解决方案而言，很难确定哪个方案更好。

这里有必要说一下单个数据库节点的原子性问题。它是在前馈分类账的帮助下实现的。一般来说，无论用户什么时候请求将事务反映到数据库，第一步就是让这一项持久存在（防止停电这类低级问题出现，确保恢复操作时更改仍旧存在），然后将其写入磁盘分类账。如果过程中出现系统故障，会出现两种可能的结果。一，如果磁盘上的账本项不见了，事务就会回滚；二，如果还在，那么重启时磁盘上的事务就可以恢复。



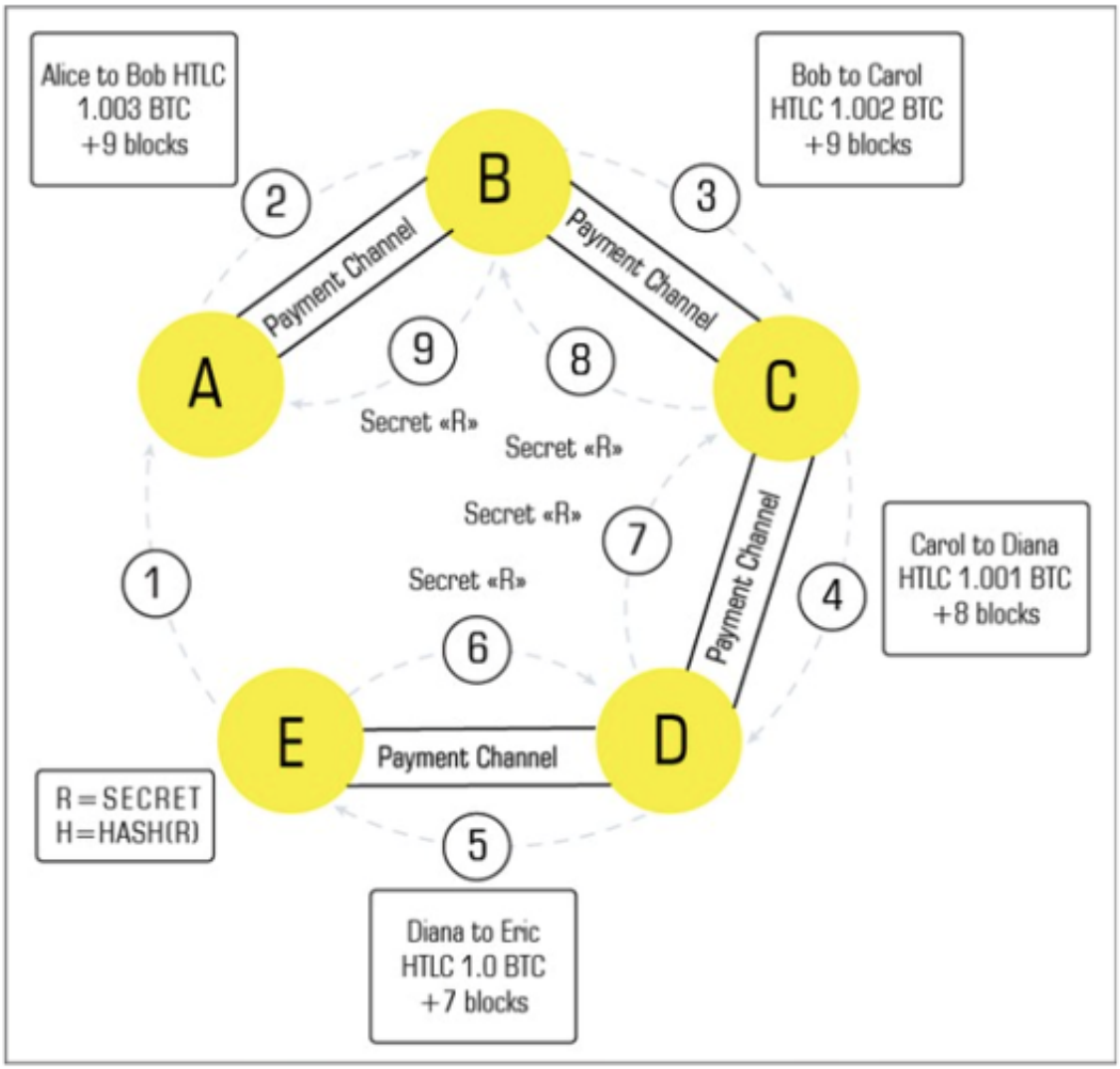
有四个本地数据库的分布式数据库，每个数据库中关于Bob和Alice的余额信息都不同

链下网络中的原子交易

自2009年加密货币出现以来，比特币用户的数量呈指数级增长。相应的，交易佣金和交易延迟情况也在增加。因此，社区积极地寻找着可扩展性解决方案。在他们的努力下，闪电网络成为第二层解决方案，实现了支付通道及多跳交易。为了防止因某人的过错而损失资金，原子性也是需要的。下面是现有的各种解决方案。

HTLC

为了实现原子性，目前用的最多的就是HTLC（哈希锁合同），即在预设锁定时间前呈现初始密钥可以花费资金。为了深入了解这一方案，我们先来看下基于闪电网络的事务流程。首先，接收方节点生成密钥并计算哈希值。之后，将该哈希值发送到发送方节点作为HTLC生成的基础。发送方生成合约并将其发送到节点1，即路径上下一个节点，该节点用递减的时间锁创建新合同（使用相同的哈希值）。这个新生成的合同由节点1沿着路径发送到节点2，然后节点2重复相同操作并继续缩短时间锁。合约一路前进到接收方，由接收方通过自己一开始就生成的密钥签名释放资金（解锁支付），然后从发出合约的节点处获得资金。反过来，这一动作向最靠近发送发的节点揭示密钥，并授权解锁支付并接收资金，然后再向前一节点出示密钥。整个路径重复这一过程后，每个节点都收到了付款，至此支付完成。所以，造成节点丢钱的唯一因素就是它无法赶在时间锁的时限内签名释放资金，比如在接收到密钥后就离线。



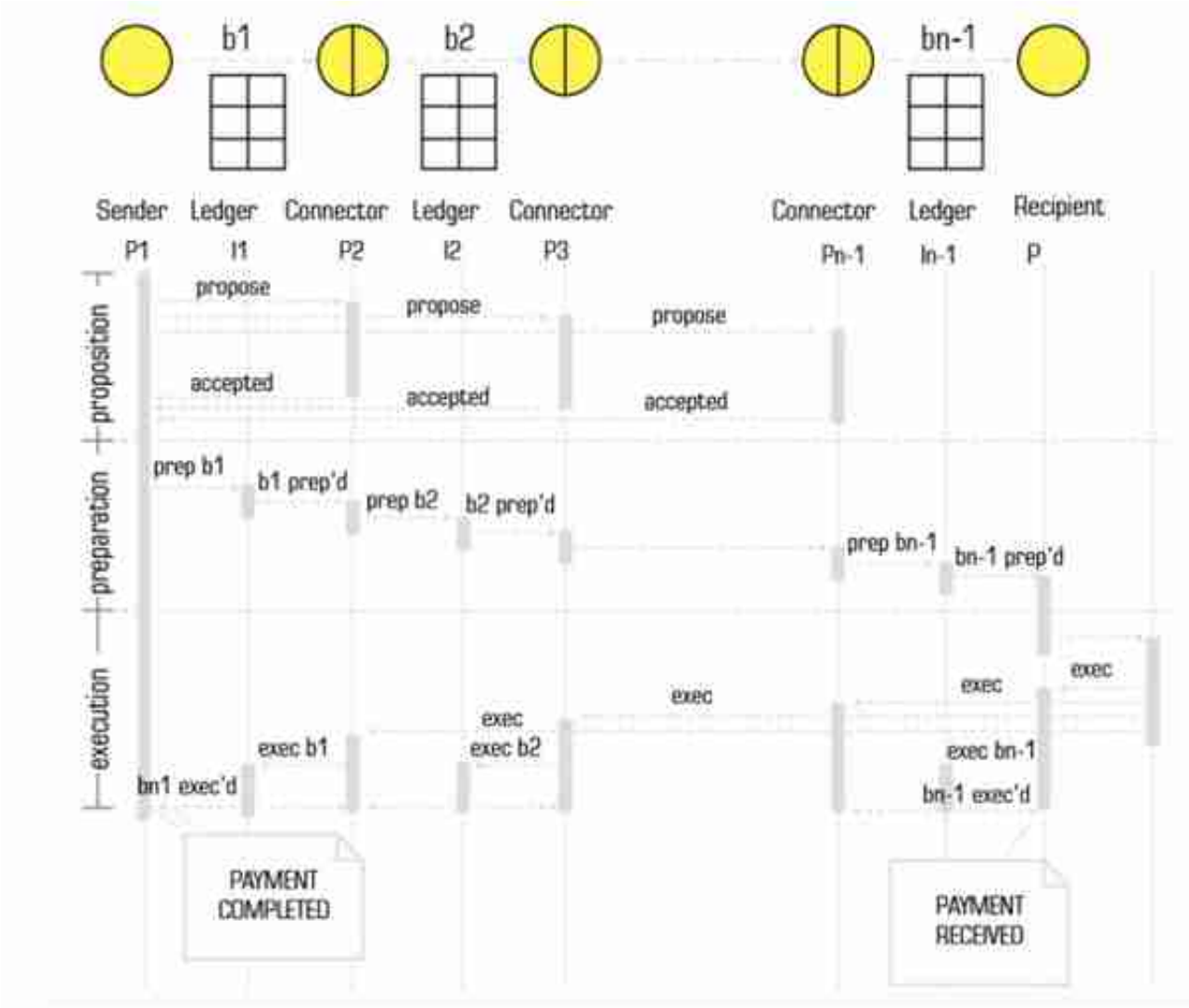
使用HTLC进行的Alice到Eric的支付（中间经过三个中间人Bob, Carol以及Diana）

该方案的缺点是，在不利情况发生的同时，偶尔会出现关于合同到期时间及客户资金损失方面的分歧。

HTLA

Ripple的Interledger协议是一个开放协议套件，用于各类账本间转账（跨链交易）。该项目白皮书中提出了“公证人”这一概念。为了实现原子性，最初建议使用通用模式（Universal Mode）和原子模式。

在通用模式下，Interledger的原子性通过HTLA（哈希时间锁定协议）实现——HTLA本质上是HTLC的改良版，两者的不同在于，HTLA能够在区块链不支持HTLC的情况下支持各类连接，包括有条件支付通道（通过HTLC更新实现）、On-Ledger持有 / 托管（使用HTLC）、简单支付通道、Trustline等等。



Interledger跨账本支付框架

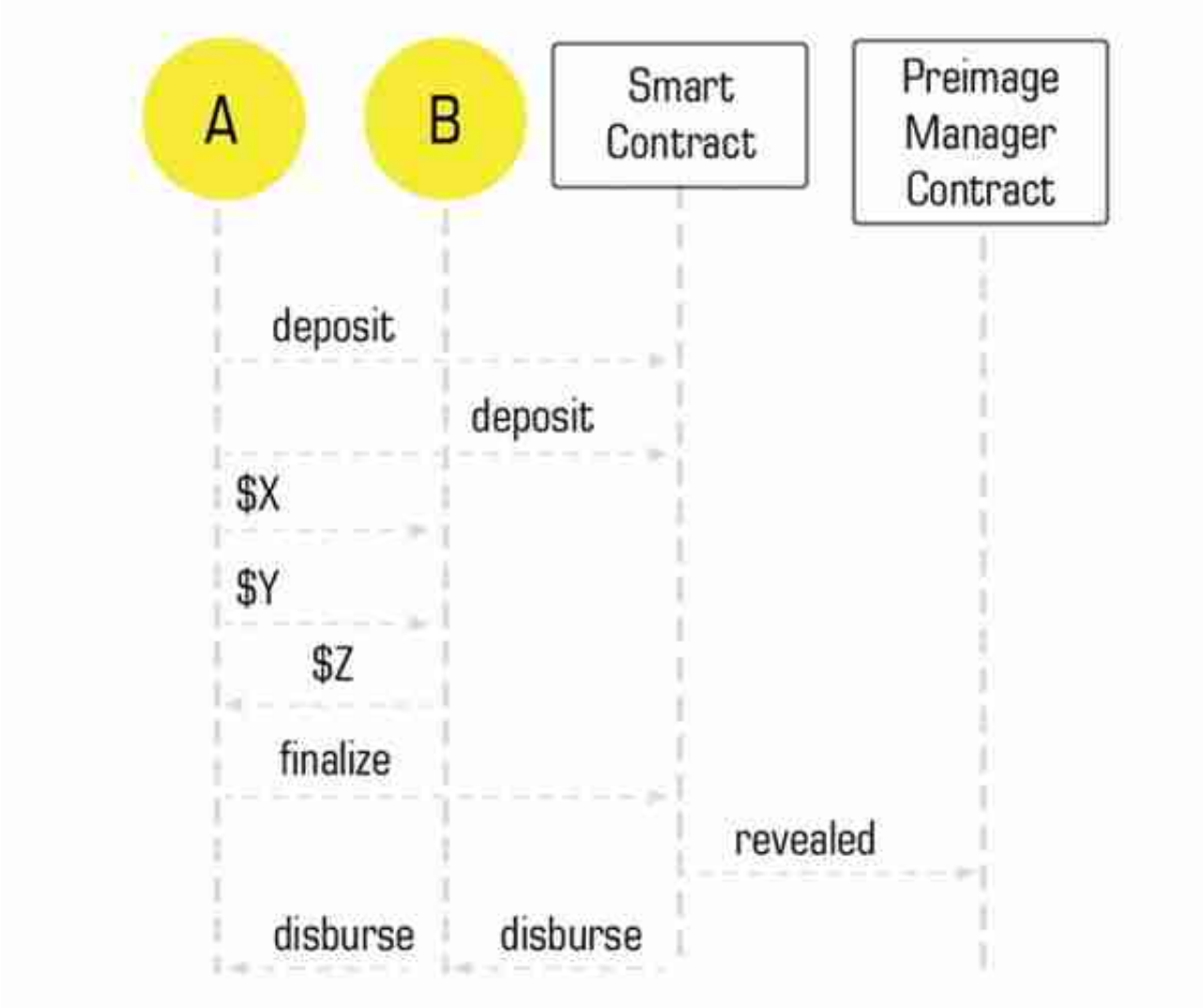
换句话说，如果跨链支付要通过不支持HTLC的区块链，那么连接器（负责传送路线的特殊Interledger节点）可以使用其他方法复制，以便满足所有合约条款，例如支付时间、金额、支付解锁条件等。

PM

Sprite通道的目标是开发新的支付通道，解决闪电网络原子性、部分存款与通道资金注销等问题。

通过添加原始合约管理器（PM），HTLC得到了显著升级。开发者是想让PM成为HTLC的仲裁人，然后将任何单个节点的合约到期决策权委托给相应的软件，防治有参与者离线并丢失钱财。仲裁人应该是常规的以太坊智能合约（或任意其他区块链），登记类似'H哈希的X 原值已于到期前在区块链上发布'这样的声明。

Sprite通道也应该有统一的合约到期时间。如果原值在合约到期前及时发布了，那么就受理所有争议。逻辑会排除有人收到了钱而另一人指出所有人的到期时间都一样的情况。但是，如果原值发布时间无效，那么这笔付款就无争议。



SpriteChannel争议解决过程

HTLR

Celer Network是针对公链可扩展性并通过离线技术实现性能最大化的解决方案。在这个体系中，PM（原值管理器）变成了一个哈希时间锁注册表（HTLR），但大部分功能仍旧保留。HTLR有两个依赖端点，即IsFinalized和QueryResult。前者返回的结果是原值是否先于区块数字完成注册；而后者返回的是原值是否已注册。这两个功能最终可以实现合并。需要注意的是，HTLR始终是链上的。

Notaries

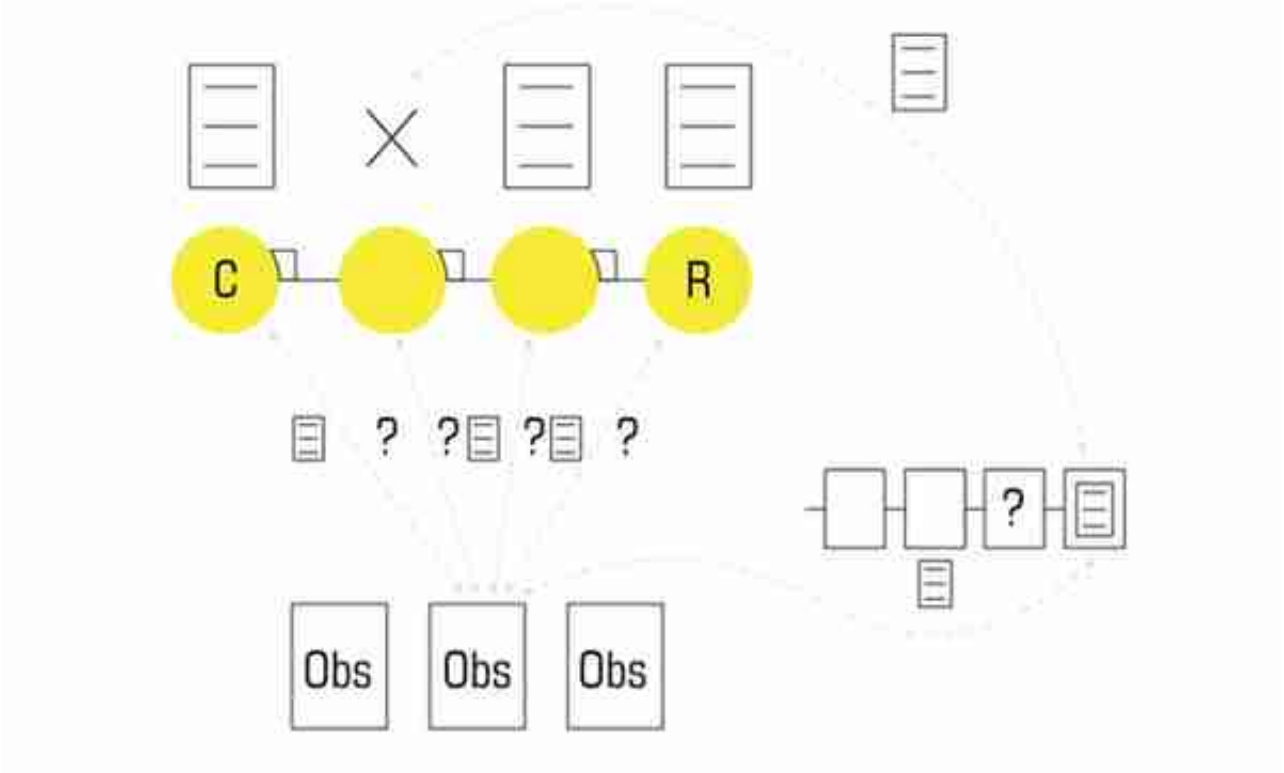
Interledger的原子模式应该就用到了“公证人”这个角色。通过公证人完成的支付和通过闪电网络HTLC进行的支付很相似，两者唯一的区别是，在出示密钥之前，接收方节点会将合约转交给公证人，即从其通用地址列表中随机选择的特殊实体。

公证人的设定是允许发送方设置验证支付的实体数量及可接受的恶意公证人数量（30%以内）。公证人必须在拜占庭容错（BFT）共识的基础上，对批准付款进行投票。如果一切正常，他们将“标记”交易，使接收方节点能够解锁资金。这个概念在白皮书中看起来不错，但很难实现跨链交易。此外，它要求用户信任公证人。

Observers

这一角色在Geo协议中出现，该概念为原子性问题提供了独特的解决方案。目前项目团队在创建一个去中心化的点对点链下网络来进行资产交换。

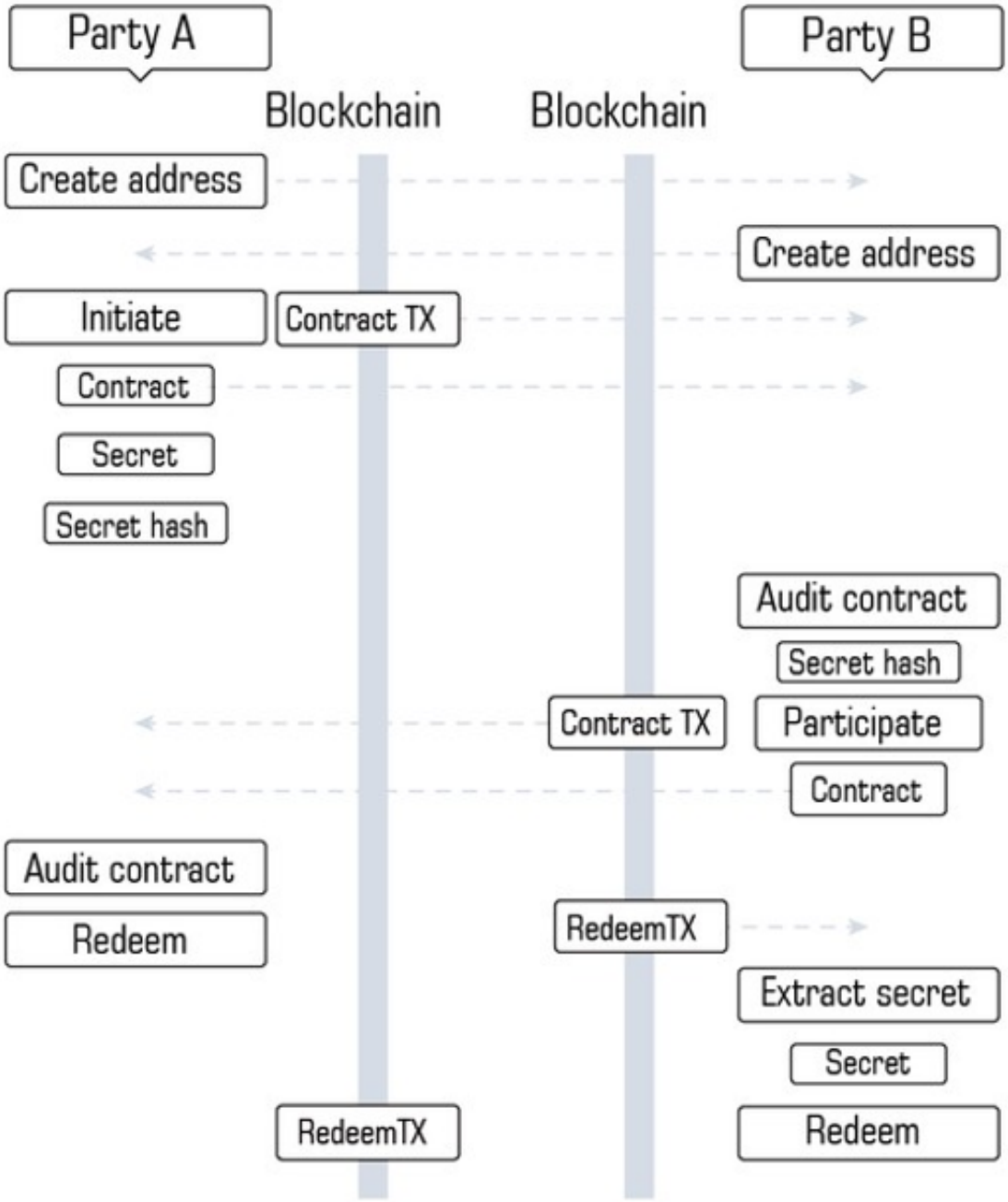
只要事务参与者在执行期间遇到问题，观察者就会介入。观察者既不会影响事务方向也不会改变任何内容。他们不是用来验证每一个事务的，只是用来干涉用户请求的。Geo采用类似二阶段提交的框架来处理常规事务。首先，所有参与者签名表示已准备好付款。其次，如果每个人都有签名列表（即“文件”），则执行付款。如果有参与者说“文件”不存在，那么观察者会在两个阶段间采取行动。这种情况下，他们会从任意节点处拿到签名列表并发送给所有参与者，如果无法完成就不执行任何操作，然后交易会按时到期失效。



GEO协议观察者工作流程

原子跨链支付

原子交换是目前实现跨链支付最常用的方式。通过原子交换进行的跨链交易无需托管服务或第三方介入。它使用时间锁合同且必须是由执行事务的区块链支持的。在现实生活中，用户在区块链上按预先设置的时间锁定想要交换的金额（如1个 BTC），然后生成一个原值、计算哈希，再声明另一个用户只有呈现出这个原值才能获得这笔钱。时间锁和哈希可以在区块链上看到（但不是原值）。这时，另一位用户想用3个莱特币换这个比特币。为此，他使用和第一个用户一样的哈希值将3个莱特币锁在合约上——他可以在另一条区块链上看到这个哈希值。而第一个用户要想从第二个用户的合约里拿到币，就必须透露他的原值，然后第二个用户再用这个公布了的原值从第一个用户的智能合约里拿到他的币。要想完成交易，第一个用户就必须透露他的原值。如果他不这样做，交易就无法结束。在这种情况下，为避免出现两位用户的资金同时被阻拦的情况，合同有效性有一定时限。



原子交易各阶段展示

多路径

闪电网络的设计者提出了多路径原子支付技术，但尚未实现。这个技术旨在将大额支付分解为多笔小额支付以缓解网络流动性问题。其原子性通过改进后的HTLC实现。接收方创建一个基础原值（BP）以用于之后创建每笔小额支付的部分原值。一旦接收方收到部分付款，就可以通过这个基础原值解锁资金，无论款项到达顺序如何。

在Geo协议里，多路径原子性的提供方式与单路径原子性完全相同（包括分配、签名集合、签名传播列表以及在遇到问题时提供解决方案的观察者）

结论

在去中心化网络中，原子性的开发受到了很多新概念的影响。

第一个出现的是哈希时间锁合同（HTLC），其优势在于节点掉线时减少损失并且保障发送方和接收方的安全。而其问题在于资金必须冻结在通道中，参与者必须保持在线以避免损失。

然后是在HTLC基础上改良后的HTLA和HTLR。前者可以实现在各种注册表中使用HTLC，甚至是那些不支持相应合同的注册表。后者则解决了节点脱机的问题。

之后的新方案中出现了观察者和公证人。虽然在使用这些方案时我们要非常小心，因为观察者/公证人的中心化可能会损害网络，但是一个设计得当的系统可以帮助这两个角色维持去中心化状态。

我们会继续进行研究，也欢迎大家一起加入分布式系统的开发与问题的解决。