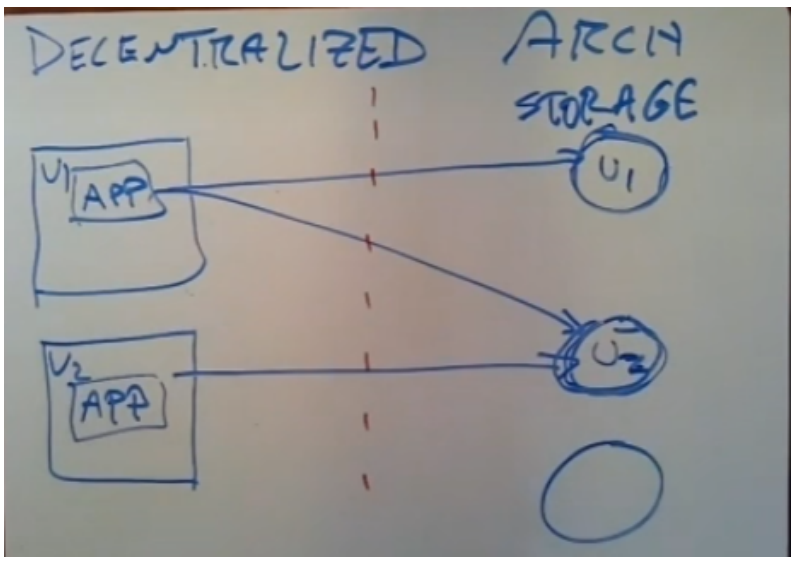
LECTURE 20 BLOCKSTACK

1. 背景知识

- Blockstack作用：
  - 构建一个以名称映射的命名系统/公钥基础设施（PKI: Public Key Infrastructure）
  - 构建去中心化应用。去中心化：用户把自己的数据从集中控制的网站中移除出来
- 传统中心化应用的缺点：
  - 用户对于自己数据的控制权很弱。不能使用其他软件/用户界面等
  - 用户数据可能被服务器嗅探，导致个人隐私信息泄露

2. 去中心化应用基本架构



- 用户购买一些存储服务，如右侧所示，左侧是用户的应用程序。
  - 应用程序和数据是分离的，不同的应用程序都可以访问数据。
  - 数据允许共享，例如U2允许U1访问其数据，则U1的各类app可以通过某种方式访问到U2的数据。
  - 存储节点需要拥有的特性：
    - 统一通用的接口
    - 具有良好的**用户间**共享/访问控制

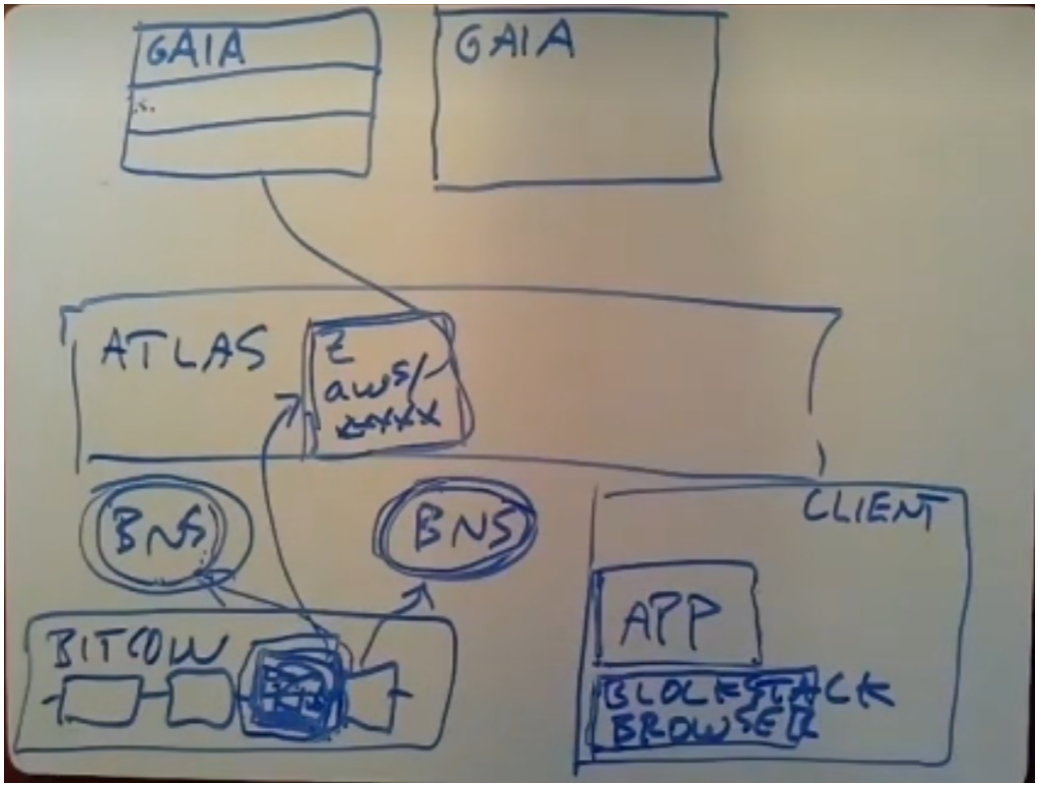
        - 对不同应用程序有隔离

- 缺点：
    - 对于特定应用程序的数据，需要约定存储节点上数据存储的格式
    - 会降低应用程序的性能
    - 需要更改应用程序的编写方式
    - 由于存储结点不具备像SQL DB那样的查询能力，因此在客户端应用程序和存储结点之间可能会有大量的数据载荷，需要客户端从存储节点处获取大片数据，而后自己进行过滤筛选等工作
    - 密码难以保护隐私。在多用户共享数据的情况下，很难采用一套加密手段既保证数据的机密性和完整性，又能保证所有用户能看到数据并且满足用户权限管理的要求
    - 对于特定用途的程序，不适合本架构（例如拍卖系统，所有的用户的出价都存在自己的存储结点中，如果拍卖程序需要获得出价排名，则必须要到别的用户的存储节点中访问数据，这样的拍卖系统不够可信）

3. Block Stack概述

- 实现映射1：$name \longrightarrow data\ location$。用于多用户共享数据
- 实现映射2：$name \longrightarrow public\ key$ 。
    - 用于验证data的所有者是否为该用户，要求用户对数据进行签名
    - 共享数据 & 加密数据。使用 被共享者 的公钥加密即可。
    - 实现ACL（Access Control List）/PKI（public key infrastructure）
- 名字必须拥有的三个特征，通常这三个特征很难同时满足（但Blockstack解决了这一问题），会相互影响：
    - Unique：在整个Blockstack系统上唯一。面对相同的名字，需要在名字后面增加序号来作为Blockstack所使用的名字。因此，随着Blockstack规模的增大，human readable属性将会逐渐下降。
    - human readable：可读。名字是用户自己取的，服从FCFS原则，名字不带有任何意义。human readble的名字有的时候会具有误导性从而降低可读性 （因为本质上名字没有现实意义，可以自己取）
    - decentralized：名字应该由网络边缘的用户指定，而不是由中心化的结点指定

4. 实现



- 结构：
  - BNS服务器（BlockChain Name Server）
  - ATLAS：为每一个用户建立一个块，存储到各个存储结点的地址。比特币区块链中存储了hash of（用户的公钥信息 && 用户存储结点位置）。ATLAS的作用为：$information_{User} \xrightarrow{hash} Zone_{user}$，其中$information_{User}$即为上面所描述的交易中所存储的相关信息的哈希值。
  - GAIA：存储系统。存储系统需要经过特殊设计。内部存储的数据需要有用户私钥的签名。
  - 客户端：需要一个**blockstack browser** 来管理私钥，因为应用程序或许不可信
- 数据访问流程：
  1. 用户访问BNS服务器，找到名字为键的映射
  2. BNS服务器会找到ATLS中用户对应的zone
  3. 通过zone中存储的各个存储节点的地址去访问数据
  4. 通过数据的签名验证数据所有者

- blockstack依赖比特币来同时达到命名的三个特征
  - 用户在一次交易中给出了一个 **从自己的名字到公钥的映射**
  - Blockstack会跟踪比特币区块链，当发现新的$name \longrightarrow public\ key$映射时，就会将其加入自己的**命名数据库** 中。用于满足命名是惟一的。命名的分配采用FCFS的方式。若后续有别的恶意用户使用同样的名字映射到另一个公钥，则Blockstack会简单的忽略这一个映射，不做操作。
  - Blockstack会在缓存比特币区块链上某个区块之前的所有命名映射状态，然后从那个区块开始，不断地跟踪区块，将新的映射加入到命名服务器中
- 缺陷：
  - 用户想要寻找互联网上的某个用户进行加密通信（需要使用用户的公钥），需要知道他们的Blockstack名字。
  - 比特币区块数据很长，因此需要大量的时间构建命名服务器（通常需要几天，但是没有别的更好的办法可以同时满足命名的三个必要特性）
