# 远程USB设备自动控制—需求分析

## 背景

少人值班，无人值守

开发目标:

1. 增加企业、代账公司、云帐房服务间的连接
2. 提高云帐房系统对信息的管理能力，降低用户的财税风险
3. 通过WEB页面上的简单操作，高度自动化、集成化财税服务。代替传统插拔USB设备、打开目标系统（浏览器、客户端）、执行功能等，简单、重复、低效的操作减少现有系统使用、维护复杂度，提高用户使用体验
4. 所有USB设备（CA、税控盘等）相关功能，在代账公司或云帐房机房执行
5. 云帐房系统与云帐房用户共享USB设备(和蒙柏确认)，云帐房使用USB\_SHARE远程映射USB设备时，企业是否可以在本机正常使用接入到USB\_SHARE的USB设备及使用相应软件系统进行日常工作？
6. 通过WEB页面进行USB\_SHARE管理、USB\_SHARE端口接入USB设备管理
7. 利用云计算集中数据管理，集中数据计算
8. 利用接口系统并行处理与外部系统对接
9. 针对独立用户（企业持有CA、税控盘）

分布式实时并行计算系统

财税垂直细分领域：

互联网

+人工智能(专家系统、模式识别)

+机器学习(采集/申报)

+物联网

+虚拟化(USB设备)

+GIS+LBS

+分享经济

+实时监控

+区块链（云帐房地区中心机房）

的最佳实践

任务分发考虑在云端计算(用C++实现？)，使用消息队列框架

根据客户端运行机器（虚拟机）能力(CPU、MEMORY、TaxSystem等)，动态请求并执行任务

充分利用用户运行客户端机器（虚拟机），提高并行计算程度

--:不同税局系统、开票系统在同一主机同时运行是否存在冲突，形成可同时运行任务组合列表，分发任务时参与计算

共享用户运行客户端机器

自主开发可视化远程控制(如：VNC, Virtual Network Console)，实现远程操作运行客户端的主机，不依赖TeamViewer运行(TeamViewer商用)。

* 点对点直连方式需要穿越内网（很困难）
* 中继方式效率低(开发简单)，在业务量不大的情况下可以考虑使用

人工智能(发票扫描采集识别接入)

提供客户端运行状态统一实时监控(ZooKeeper)系统，地理信息系统+基于位置的服务，查看客户端运行设备、运行任务、接入USB设备等信息，控制运行客户端主机

云帐房部署集群，实现非代账公司用户的财税业务？

全网财税系统实时运行监控系统

地理信息系统（Geographic Information System或 Geo－Information system，GIS）

LBS(基于位置服务)

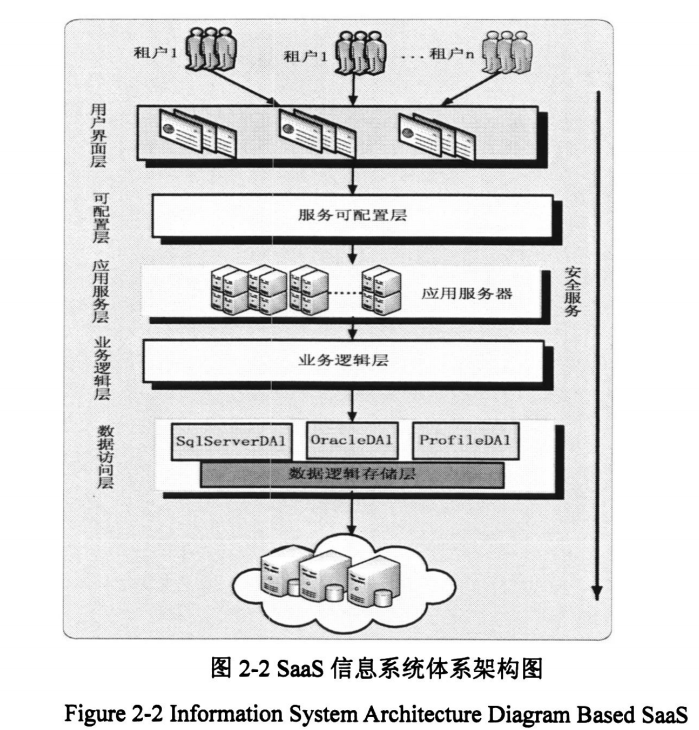
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MAC/IP | OS | Power | CPU | Memory | RunTask | Operator |
| 192.168.0.176 | Windows7 | On | 7% | 8% | NULL | On/Off Power, Start/Stop Client |
|  |  |  |  |  |  |  |

* 远程启动/停止主机（虚拟机）
* 远程启动/停止/更新升级InterfaceClient
* 运行环境展示：如：硬件信息(CPU、Memory)、操作系统、主机名称、网络信息（MAC、IP、Gateway、Subnet Mask）
* 运行（历史）任务展示（柱状、曲线等）

## 系统物理结构图



## 系统逻辑结构图



InterfaceService：管理USB\_SHARE(Device)、管理客户端、降低云端并发、用户公网访问限制

接口系统简化：多机分布式→单机多进程

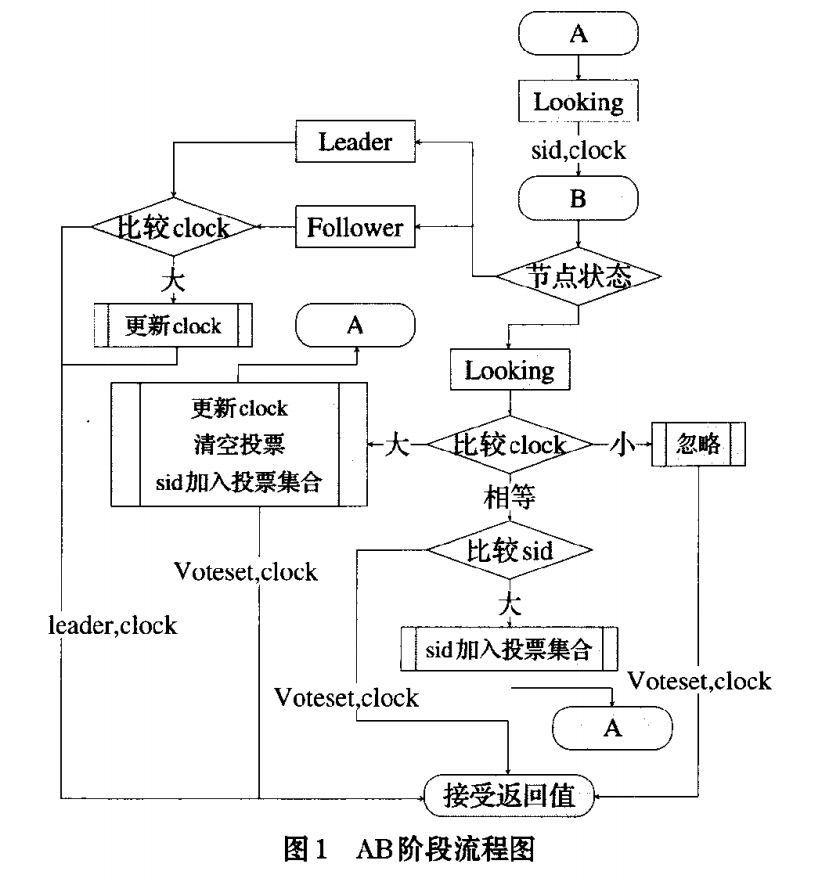
云端与接口系统建立TCP长连接？

内存数据库

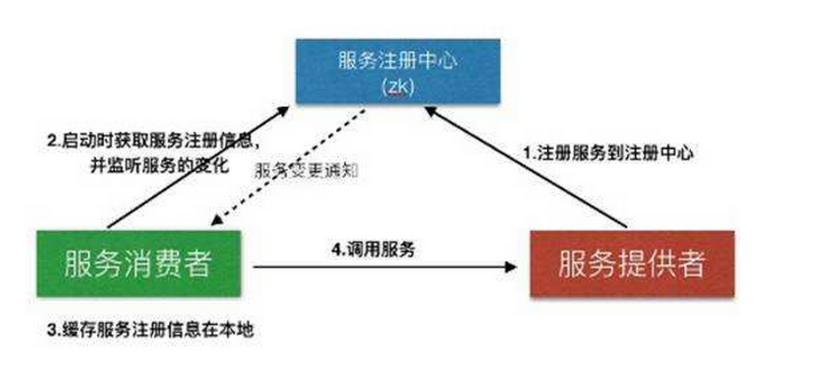
读写分离

负载均衡

分布式 实时 并行 并发量 计算量 通信量



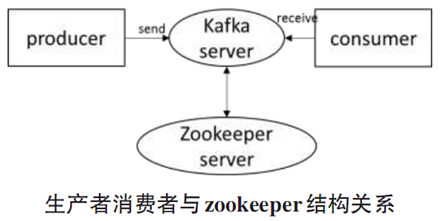
1. 最终一致性：Client不论连接到哪个Service，得到的都是同一个视图，这是Zookeeper最重要的功能
2. 可靠性：具有简单、健壮、良好的性能，如果消息message被一个Service接收，那么message将被所有的服务器接收
3. 实时性：Zookeeper保证Client将在一个时间间隔范围内获得Service的更新信息，或者Service失效的信息。但由于网络延时等原因，Zookeeper不能保证两个客户端能同时得到刚更新的数据，如果需要最新数据，应该在读数据之前调用sync()接口
4. 等待无关（wait-free）：慢的或者失效的client不得干预快速的client的请求，使得每个client都能有效的等待
5. 原子性：更新只能成功或者失败，没有中间状态
6. 顺序性：包括全局有序和偏序两种：全局有序是指如果在一台服务器上消息a在消息b前发布，则在所有Server上消息a都将在消息b前被发布；偏序是指如果一个消息b在消息a后被同一个发送者发布，a必将排在b前面

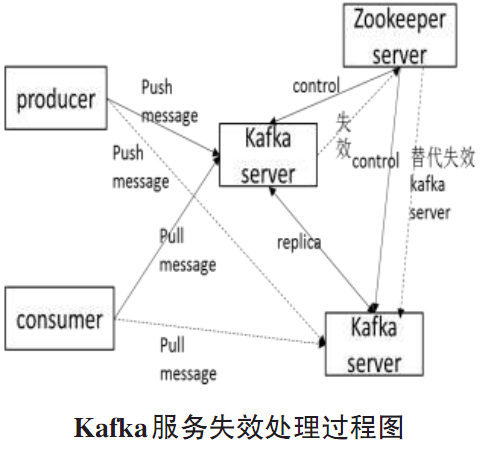


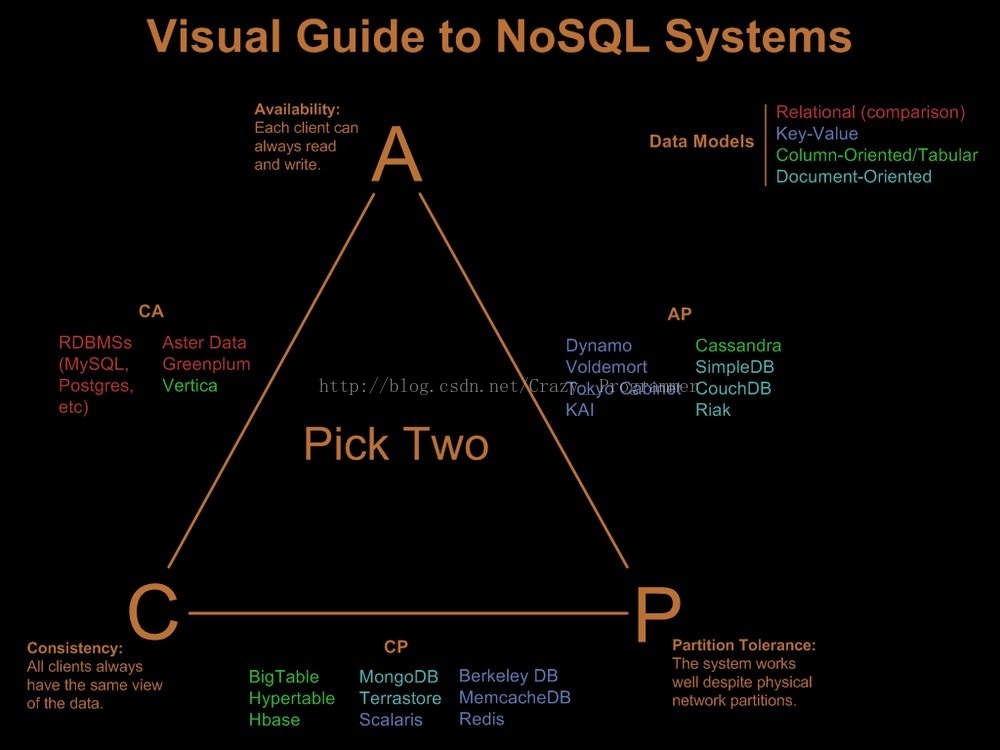
Zookeeper

单一故障点(双通道冗余热备)

投票选举Dispatcher







* Consistency(一致性), 数据一致更新，所有数据变动都是同步的
* Availability(可用性), 好的响应性能
* Partition tolerance(分区容错性), 可靠性
* CA without P：如果不要求P（不允许分区），则C（强一致性）和A（可用性）是可以保证的。但其实分区不是你想不想的问题，而是始终会存在，因此CA的系统更多的是允许分区后各子系统依然保持CA
* CP without A：如果不要求A（可用），相当于每个请求都需要在Server之间强一致，而P（分区）会导致同步时间无限延长，如此CP也是可以保证的。很多传统的数据库分布式事务都属于这种模式
* AP wihtout C：要高可用并允许分区，则需放弃一致性。一旦分区发生，节点之间可能会失去联系，为了高可用，每个节点只能用本地数据提供服务，而这样会导致全局数据的不一致性。现在众多的NoSQL都属于此类



云计算和跨地理区域的多数据中心（multi‐data‐centers），复制数据副本已经成为解决伸缩性（scalability）问题、负载均衡以及高并发等问题的首要方法。

任务（采集、申报等）数据在分发执行返回等过程中存在副本，这是分布式机制不可避免的代价。

维护数据副本（replica）之间的一致性（consistency）会增加系统的负担，对系统性能造成影响。

正如CAP理论和PACELC模型指出的：人们在设计分布式存储系统时，需要根据不同的业务需求，在一致性和可用性以及一致性和延迟之间做出权衡

**数据库系统一致性 与 分布式系统一致性 理解、定义不同**

在讨论数据一致性时，数据的一致性状态指的是数据及其全部副本都服从某种规则的约束

这种对数据和副本进行约束的规则，在数据库领域（主要是关系型数据库）和分布式系统领域都有存在，但是两个领域这种规则约束的理解和强弱要求有很大差别。

* **数据库领域对****一致性定义**

在（R）DBMS中，可使用“事务”（transaction）作为数据库操作逻辑工作单位，事务具有四项特性：原子性、持久性、隔离性和一致性。一致性在数据库领域指的是事务的一种属性。

具体而言，事务要么全部执行，要么完全不执行（原子性）；事务执行完毕后，它对数据库进行的更改被永久性地存储在数据库中（持久性）；当多个事务并发地操作数据库中同一数据时，采用合适的机制使它们以串行的方式执行下去（隔离性）。

所有对数据库的操作都要遵循一定约束，保证在事务结束后，数据库的完整性不被破坏。

* **分布式领域对一致性定义**

分布式系统主要研究数据各个副本所呈现的状态，即同一份数据的多个副本之间是否存在不一致的情况。分布式系统中对一致性的要求是：数据和其副本达到完全相同的状态。在这些副本上所执行的操作可能是读取，或者改变一个或者多个副本的状态。

一致性模型（或者叫一致性协议）定义了在具体分布式系统中哪些操作被认为是“正确的”，或者是定义了以何种顺序执行操作可以保持分布式系统中数据的正确性。

分布式系统处于一致性状态，当且仅当所有的数据及副本都保持一致，并且某种特定一致性模型的顺序保证(ordering guarantee)不被破坏

**一致性的理解**

* **“以客户为中心”**（client-centric）一致性：服务水平协议(service level agreement)，指客户端和服务端在某些指标（比如客户端情况 API 的速率、服务器端的延迟等）上形成一致的协议。

侧重描述一致性协议的最终效果，对开发人员友好，却忽略了协议实现的内部细节。

* **“以数据为中心”**（data-centric）一致性：关注分布式系统的内部状态，具体而言，主要关注副本之间的进程同步问题和操作的执行顺序问题（共享数据上的并发操作和顺序一致性）。最终目的可以归结为：从数据存储的角度提供系统级别的一致性。

侧重描述一致性协议的概念和定义，对于相关属性的描述严谨、清晰，但不涉及协议的具体实现及效果，对应用级开发人员没有直接帮助

**一致性评价**

* **滞后性**（staleness）:指数据发生改变的时刻和其副本接收到更新的时刻之间的时间段，描述了更新传播的快慢程度。主要用于考量“以客户为中心”的一致性协议
* **顺序性**（ordering）：描述系统内部的运行机制，如：各种操作在数据及其各个副本上执行的顺序。在关系型数据库中，为了获得强一致性保证，操作被以发生的时序顺序串行的执行；而在分布式环境中，这种强一致性很难实现，因此，分布式一致性协议往往会根据需求适当放松对顺序性的要求。主要用于考量“以数据为中心”的一致性协议。

**以客户为中心的一致性模型**

* **单调读一致性**(monotonic-read consistency)：保证已经读取过数据m的用户将来再次读取数据m时，得到的是数据m的更新版本。即：对于用户n，在t1时刻读取到版本号为v1的数据d，那么用户n再次读取数据d时，得到的数据版本v2不小于v1(v2>=v1)

读取不同副本的相同数据，可以得到最新数据

* **读写一致性**(read-your-writes consistency)：当一个进程p对数据d执行一次写操作生成数据版本v后，该进程对数据d执行的后续读操作一定可以得到版本v。换句话说，进程p对数据d进行修改后，该进程无论访问数据d的哪个副本，都能得到这次修改的结果。**在对某副本发生变更后，读取不同副本的变更后数据，可以得到最新数据。**

例如，某用户登录网上银行进行转账操作，在这次操作结束后，该用户无论何时何地以何种方法查询账户余额，得到的都是转账后的结果

* **单调写一致性**(monotonic-write consistency)：要求同一进程所发起的写操作必须以串行且顺序相同的方式在各个副本上执行。定义：对于进程p，它对数据d执行的写操作必须在其所开展的所有后续操作开始之前完成。也就是说，同一进程先后所发起的两次读操作，将以串行的方式在各个副本上执行。单调写一致性避免了更新丢失问题，即之前一次的写操作在还没执行完时被更新的一次写操作覆盖。

例如，在没有单调写一致性的保障下，如果某用户连续进行两次转账操作，后一次操作就有可能将前一次操作覆盖掉，导致十分严重的后果

* **写读一致性**（write-follow-reads consistency）：可以看做是单调写一致性的扩充。定义：进程p读取到版本为v的数据d后，进程p对数据d所执行的所有操作都要在发生在版本号不小于v的d的副本上执行。也就是说，对数据项d所执行的任何后续写操作都会在d的副本上执行，而该副本是用最新读取的值更新的。

以微博系统为例，假设用户A读取了某条微博B并对该微博进行了评论，那么写读一致性保证用户A对微博B的评论会被写入在所有包含微博B的副本上

**以数据为中心的一致性模型**

这类模型以顺序性为考量指标而忽略滞后性，以模型对顺序性的保障，从弱到强地顺序介绍。

* 最终一致性(eventually consistency)：是以数据为中心的一致性协议中对顺序性保障最弱的协议，对操作的执行顺序几乎没有要求。降低一致性要求，提高系统可用性。对于任何更新操作，系统都允许其对数据副本进行修改。然后在未来的某一时刻，通过复制(replication)机制实现全部副本的一致性。也就是说当没有更新操作时，所有副本将逐渐统一，最终达成一致状态。

适用于对一致性要求较低的业务场景。通常数据的读取要比更新频繁得多，对于读取速度的需求要大于对一致性的需求，并且写操作冲突(write conflict)发生的概率较小，这种场景中很少甚至没有对同一数据的并发更新操作。

* **因果一致性(causal consistency)**：在顺序性方面对具有潜在因果关系的操作和没有因果关系的操作进行了区分。因果一致性定义：如果操作r２是由操作r１激发的，或者操作r２依赖于操作r１的结果，则操作r１和r２具有因果关系，r１和r２应当以这样的因果顺序被所有进程看到；而对于不具备这样因果关系的操作，则允许以任意顺序被所有进程看到，这类操作称为“并发的”。因果关系包括以下三种：

１）同一进程先后发起的操作r１和r２；

２）操作r２依赖于操作r１的结果；

３）操作之间存在因果律（transitiverelation），即：操作r３依赖于r２，r２依赖于r１，则r３依赖于r１

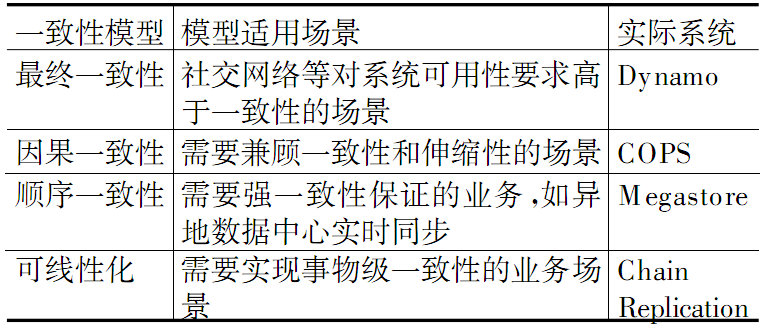
可通过定义dependencies来捕捉操作之间的因果关系，以实现因果一致性

* **顺序一致性(sequential consistency)**：要求全部进程以相同的某种顺序在各个副本上执行。 即：顺序一致性要求全部进程看到完全相同的操作执行顺序。

Paxos算法：如果各个节点的初始状态一致，每个节点执行相同的操作序列，那么它们最终能达到一致状态

BigTable(Google)

* **可线性化(linearizability)**：首先，在全部操作序列化的基础上，可线性化增加了对时间的要求，这种属性的形式化描述是：如果r１→r２，那么t（r１）＜t（r２）。可以认为，可线性化一致性模型要求操作按照发生时间的先后顺序来序列化。



**采用二阶段提交**(two-phase commit)来保证一致性的分布式系统，每次操作都需延迟数秒。

最终一致性的兴起：通过减低对一致性的要求，即采用或设计强度较弱的一致性模型，可以实现降低时间代价的目标。

## 系统时序图

## 流程图

## 系统功能

1. USB\_SHARE设备信息注册云服务

UI需区分代账公司侧、小微企业侧、直客

1. USB设备(税控盘、CA)信息注册云服务

UI需区分代账公司侧、小微企业侧、直客

1. 云服务持久化映射关系(企业↔USB\_SHARE、USB\_SHARE↔USB设备、企业↔USB设备)
2. WEB前端维护映射关系(增删改查)
3. WEB前端提供税控盘对应功能(同步、采集、抄税、清卡、开票等)（“一键”）
4. 接口系统获取USB使用权（已占用）、打开端口（映射本机、接入USB设备）、关闭端口（取消映射、移除USB设备）
5. 接口系统(自动/手动)（定期）枚举已知USB\_SHARE设备上接入的USB设备
6. 接口系统(自动/手动)（定期）与云端同步已知USB\_SHARE设备上接入的USB设备信息
7. 接口系统定时向云端请求任务（税控盘、CA）

10、接口系统向云端返回任务执行结果

11、接口系统自动控制开票软件（百旺、航信、中软四方等）启动/关闭

12、接口系统自动控制开票软件（百旺、航信、中软四方等）相关功能的执行

13、接口系统自动发现开票软件安装路径

14、云端服务拆分

15、USB\_SHARE管理：目前USB共享单元、USB设备管理在接口系统侧，新场景需要上云管理。原因如下：

1. USB设备跨公网映射
2. 与蒙柏云服务交互(信息同步、设备映射)
3. 降低数据一致性维护复杂度

一张表表示全部：开发简单、维护复杂、扩展复杂

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 备注 |
| Unique\_Id | 唯一标志(KEY、Index) |  |  |
| EnterpriseId | 代账公司唯一标识 |  |  |
| EnterpriseName | 企业名称 |  |  |
| TaxCode | 纳税人识别号 |  |  |
| ManageCode | 税务管理码 |  |  |
| USB\_SHARE\_ID | 企业所属USB\_SHARE标志 |  |  |
| USB\_SHARE\_PORT\_CA | 企业CA位于USB\_SHARE端口 |  |  |
| USB\_SHARE\_PORT\_TAX\_DISK | 企业税控盘位于USB\_SHARE端口 |  |  |
| 蒙柏云端交互用字段 |  |  |  |

拆分上表，外键关联：开发复杂、维护方便、扩展简单

## 安全性

SaaS模式下通过三种基本方式保证其安全性，

一、隔离：在用户和数据源之间建立有效的中间层，通过设计数据存储层模型来实现

二、权限，采用访问控制列表来决定访问级别；

三、加密，对于每个租户拥有的数据均进行加密处理。

过去CA、税控盘等设备在代账公司（企业）侧，设备使用安全是云帐房用户独立维护的。

现在USB设备上公网，存在非法使用风险，需增加系统安全性。

HTTPS(openssl)

通信(非)对称加密

鉴权、授权等

非云端数据本地持久化加密，提供可视化程序维护

用户使用客户端登录

客户端运行方式、运行情况监控

## 开发方式

1. 直接在现有系统上进行迭代

不推荐

风险较大

线上正在运行，用户量大

合并功能较多，新功能的使用，对原有功能影响

1. 先开发独立系统，再融合到现有系统

推荐

风险可控

后台客户端运行，服务化

## 异常处理

## 系统运行监控

划分系统运行状态

记录系统运行关键信息(环境、时间、结果、历史)

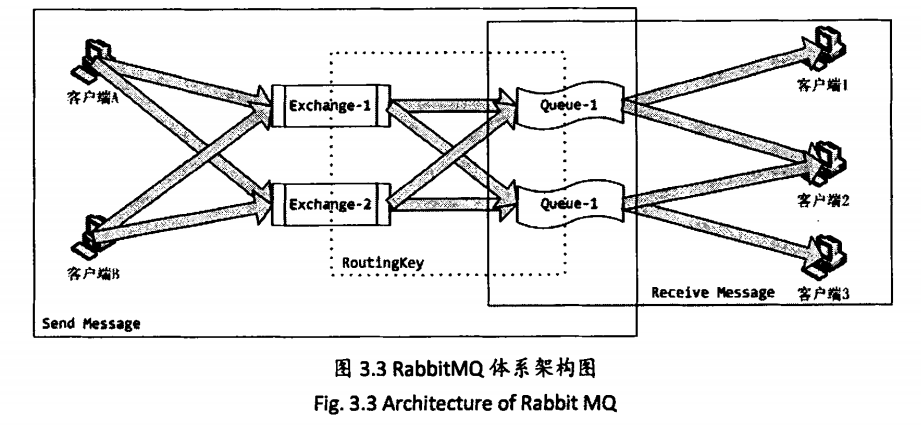
集中记录状态、信息

实时展示状态、信息

分析、定位、恢复故障

## 名词解释

RabbitMQ



ActiveMQ

ZeroMQ

nanomsg

1. 每月抄税起始日起不抄税不能开票;必需在十日内抄税。(写IC卡)报税，若过期未抄报税，机器自动锁死

2. 本月没开票，也必需抄报税

3. 抄税时不能拔出IC卡

4. 若下月报税期恰好是节假日，本月税务机关在您报税时会作授权，延期报税

5. 写IC卡后要到“报税处理/状态查询”中查看是否抄税成功

6. 要先报税再购票，报税后一定要进入系统一次

7. 每月7号前必需报税

**1、增值税专用发票的开具**

增值税专用发票的开具是通过增值税开票系统进行的。纳税单位经过税务机关的认证，取得了增值税一般纳税人资格后，就可以安装增值税防伪开票系统。防伪开票系统由金税卡、系统软件、IC卡及读卡器、实物发票、打印机组成。

系统安装完必后，第一步是到税务机关购买增值税发票，此时要带上IC卡，税务机关出售发票后会将发票的电子信息读到IC卡上，如发票号、版本、开票限额等，这些信息与同时购买的纸质发票一致。

购买发票后，回到企业，将IC卡插入本单位开票系统读卡器，此时，发票的电子信息将被读入开票系统。

有了发票之后，就可以开具发票了，此时要注意，系统开具的发票是顺号的，在打印实物发票之前必须将系统开具发票的号码与从税务机关购得的实物发票号码核对一致。

　　发票打印出以后，加具开票单位的发票专用章，就可以将发票送给购货单位了。如果发票有错，则需要作废，此时在作废实物发票的同时，也要作废系统发票。

　　有时可能由于销货退回、折扣与折让等原因，需要开具经字发票，开票系统提供了这一功能，这时只要输入原蓝字发票，就可以开具对应的红字发票。但需要注意的是，开具之前一定要取得购货单位到税务局办理的销售退回或折扣与折让证明单。

2、增值税进项发票的认证。

　　增值税进项发票的认证在税务机关的认证系统进行。纳税人在取得增值税进项发票以后，就可以到税务机关大厅进行增值税发票的认证，认证的目的是确认增值税发票的真假，只有通过认证的发票才能抵扣。认证的过程很简单，只需要将发票准备好，拿到国税申报大厅，由税务工作人员将发票信息扫入系统，由系统自动进行比对，就可以确认发票的真假。对于开展了网上认证的地区，纳税人可以在自己单位将发票通过扫描仪扫入，将数据文件传给税务机关就可以完成认证。增值税发票认证的期限为从开票之日起三个月，目前当月认证的发票必须在当月抵扣。

**3、抄报税**

　　抄报税指的是将防伪开票系统开具发票的信息报送税务机关。这个过程分为两步，第一步是到月底，在开票系统进行抄税处理，将本月开具增值税专用发票的信息读入IC卡(抄税完成后本月不允许再开具发票)，第二步就是将IC卡拿到税务机关，由税务人员将IC卡的信息读入税务机关的金税系统，整个过程就完成了。

　　经过抄税，税务机关确保了所有开具的销项发票到进入到了金税系统，经过报税，税务机关确保了所有抵扣的进项发票都进入了金税系统，就可以在系统内由系统进行自动进行比对，确保仍何一张抵扣的进项发票都有销项发票与其对应。

**4、纳税申报**

　　完成了本月核算工作，抄报税完成后，企业就可以进行本月增值税的纳税申报。一般情况下销项税额与进项税额的差额就是本月应纳税额.其中销项税额根据本月销售收入与适应税率计算出，这个数字应当大于等于本月抄税中的销项税额，这是因为还可能存在未开票的销售收入，如果小于，纳税申报就会有问题。进项税额不得大于本月认证的进项税额(当然如果有农产品收购、运输发票抵扣的除外)。

5、税款缴纳

　　增值税申报之后，税务机关会开具税款缴纳的单据，纳税单位就可以直接将这些单据送到自己开户银行，由银行进行转账处理。当然也可以到税务机关指定银行进行现金缴纳。

抄报税流程

　　进入系统----抄税处理-----抄税处理-----点击确定----- 抄税成功，

　　抄税处理------发票资料----打印本月汇总表明细表，到国税所进行抄报税处理

一般纳税人报税流程

一般纳税人每月1日必须进行防伪税控开票机的抄税工作，在税务机关规定的报税时间前到办税服务厅1-10号综合办税窗口、11-12号企业发行窗口或13-14号代开专票窗口中任意一个服务窗口都可以办理报税工作。报税时纳税人需要携带税控IC卡、增值税防伪税控开票系统打印的增值税销项发票统计表和增值税销项发票明细表(2-9表)，抄税不成功的纳税人还需要携带上月已开具发票的记帐联进行存根联补录。

一般纳税人抄税、报税时应当注意：

1、上月纳税人开具的所有的增值税专用发票是否全部抄税、报税;否则就有可能造成存根联漏采集;

2、一般纳税人月度中间重新更换金税卡的要特别注意在更换金税卡前已经开具的增值税专用发票是否抄报税成功，否则要携带未抄税成功发票的存根联或记帐联到办税服务厅服务窗口进行非常规报税并进行存根联补录;

3、纳税人报税后申报时发现比对不符的要及时查明原因，不能随便调整申报表数据进行申报。

一般纳税人

　　1.首先作帐，出报表，这个是必须的。

　　2.如果有增值税，进项票认证好，然后到开票系统，先把增值税开票的IC卡插到读卡器，点发票管理抄税处理。填写表格，5张表。写卡，打印，增值税明细表正数负数表等.

　　3.准备中国税务字样的IC卡，插入读卡器，和小规模的程序一样的，先读卡，填写表格，应该是5张增值税的，一张主表，4张附表。资产负债表，损益表。另外如果是机动车的还有3张表。填写好，调整打印纸张，进行打印，一式两份。写卡。因为我单位是机动车销售的应该有一个步骤是从机动车开票系统导出一个报表导入报税系统.直接生成机动车销售清单.连续打印报表.

4.完毕后盖公章，拿着报表到报税大厅报税

一、每月月底将本月需认证的进项发票在网上认证，期限为90天。

打开“企业电子报税管理系统”，点击“进项发票管理”下的“防伪税控进项专用发票管理”，点击左上角的“新增”，再录入相应的票面信息，全部核对正确后点“保存”后，再点“审核”，有多少张新增多少次。

　　所有待认证发票输入完后，点击“进项发票管理”下的“待认证进项发票导出”，点击左上角“录入日期”，然后点确定，等需认证发票全部显示出后，再点击右上的“写网上认证文件”(如果需要去税局认证，需点击“写XML文件”)，选好文件存放路径(一般存放在 A 盘或移动 U 盘中)，点击“确定”。弹出写文件成功提示框，继续点“确定”，然后进入广东税局网开始发票认证。

　　网上认证需要1-2工作日以后得知认证结果，待结果出来后，在“认证查询”栏中查询认证结果，如果全部发票认证成功，①点击“认证结果下载”的“进入下载”，“生成文件类型”选择“认证101”，“所属年月”选择本月，点击下载。②点击“认证结果通知书下载”， 下载后把通知书打印出来。 如有发票认证未通过需再上传一次

　　未通过认证的发票。还需将所有认证的进项发票各复印一份自用。

二、每月1日必须抄税

　　打开“防伪开票系统”，点击“报税处理”下的抄税处理，否则在1日24:00以后系统自动锁死，无法开票，需到税务机关解锁。

三、每月1-10日报税(如遇节假日，可延后)

　　1、将销项发票读入到本月开票数据中

　　需要先打开防伪开票系统下的发票查询，选择当前需导入发票月份，窗口必须最大化。然后切换到企业电子报税管理软件，点击“销项发票管理”下的“从界面读开票数据”点击“读发票查询界面”，数据自动一个个读入。在读取过程中最好不要操作键盘和鼠标，不然有可能影响到发票正常读入，读完后点“保存”。

　　2、纳税报表生成

　　点击“纳税申报”下的报表，按顺序一个个生成完，一定要审核。 生成完毕后再点击“申报主界面”里的“生成申报盘” (全部是审核状态才可生成申报盘)，按提示存入存放路径(一般为A盘或移动U盘中)，然后进入网站按提示进入申报。网上申报成功后就打印全部报表(除增值税纳税主报表需打印二份，其余都打印一份)。

　　3、前往税局报税所需报表及资料：

　　◆ 增值税纳税申报表二份

　　◆ 增值税纳税申报表附例资料表各一份 (表五、表一、表二、表三、表四、表六)

　　◆ 资产负债表、损益表各一份

　　◆ 增值税发票汇总表各一份 (专用发票汇总表和普通发票汇总表)

　　◆ 本月所认证的进项发票及认证通知书

　　还需填制“申报扣税凭证封面”2份,将所有认证成功的进项发票装入一个文件袋中，文件袋封面上需写上：厂名、纳税识别号、卡号、联系电话、联系人及申报月份。还要带上IC卡、资料卡及公章。

　　发票核销及购买

　　如本次购买发票己开完，需在防伪开票系统中打印“正数发票清单”，如果有本次有作废发票，还需打印“正数废票表单”，然后填制一份“发票存根联封面”和“购买发票申请表”(表中金额全部为含税金额)，并全部盖上公章。然后带上IC卡、资料卡、公章、购票人身份证原件前往税局先核销发票，再购买。如果本次购买发票存在跨月情况，那么核销发票时要填二张“发票存根联封面”。

　　发票购买回来后及时将其读入到电脑系统中。