

第十八届全国科学计算与信息化会议

2017-07-03--2017-07-07 山东威海

区块链技术与科学计算应用探究



陈建海 博士

浙江大学计算机学院
智能计算与系统实验室Incas-lab
云象区块链CTO
2017年7月6日

报告提纲

- 01 区块链简介
- 02 联盟链技术架构与智能合约
- 03 应用案例-供应链金融
- 04 科学计算应用思考



01

区块链简介

IT的昨天、今天、明天

- 昨天到今天——IT的美好春天

- 云计算和超算；
- Web20互联网+；
- 大数据、深度学习、人工智能2.0和工业4.0；

- 今天——

- 互联网很成功；大数据开始有成效；人工智能看到希望；开始了工业4.0的探索.....貌似信息系统很成功

- 然而，昨天、今天乃至未来相当长的时间

- 传统应用信息系统存在的重大问题——不能构建和传递信任



信任

by IXX

信任

- 贷款、融资、办证、买卖、交易、下订单、项目签署、住酒店……无一离不开**信任**
- 快速传递**信任**的好处
 - 互联网电商B2B,B2C,C2C……实现了完全不相识的两人达成交易，——如阿里平台，支付宝
- **信任**的代价
 - 业务流各环节为**信任**不断重复着本不该做的许多事。



身份证、户籍、学位证、合同、公证、公章、签名、版权……**信任**的载体

- 我是谁? ——身份证
- 夫妻? ——结婚证
-



- ——依赖第三方的**信任**——
- 一个中心机构：政府、银行、法律、中介...

尴尬

杭州市长暗访“最多跑一次” 为啥四件事都没办成？

2017-04-26 21:39 | 浙江新闻客户端 记者 袁华明



信息孤岛-信任不流通

一对夫妇去**房产局**办证



户籍身份
核实

跑1次



婚姻状况
核实

跑2次



税费状况
核实

跑3次



迫切需要

可信的
网络
可信的
平台
安全的
环境

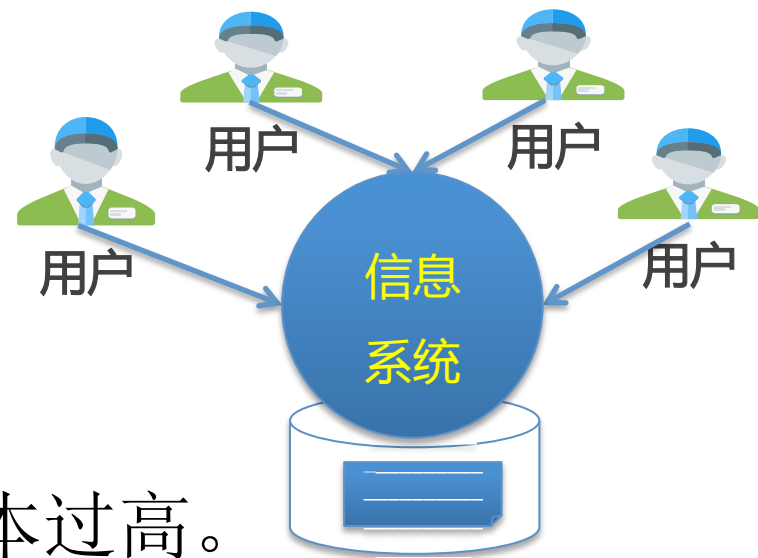
一个事实

——互联网发达的今天处处“信息孤岛”，信息不共享且不可信

痛点

- 中心化信息系统

- 不透明、不共享、可篡改、不可信



- 信任——多部门的协作和办事效率低下，交易成本过高。

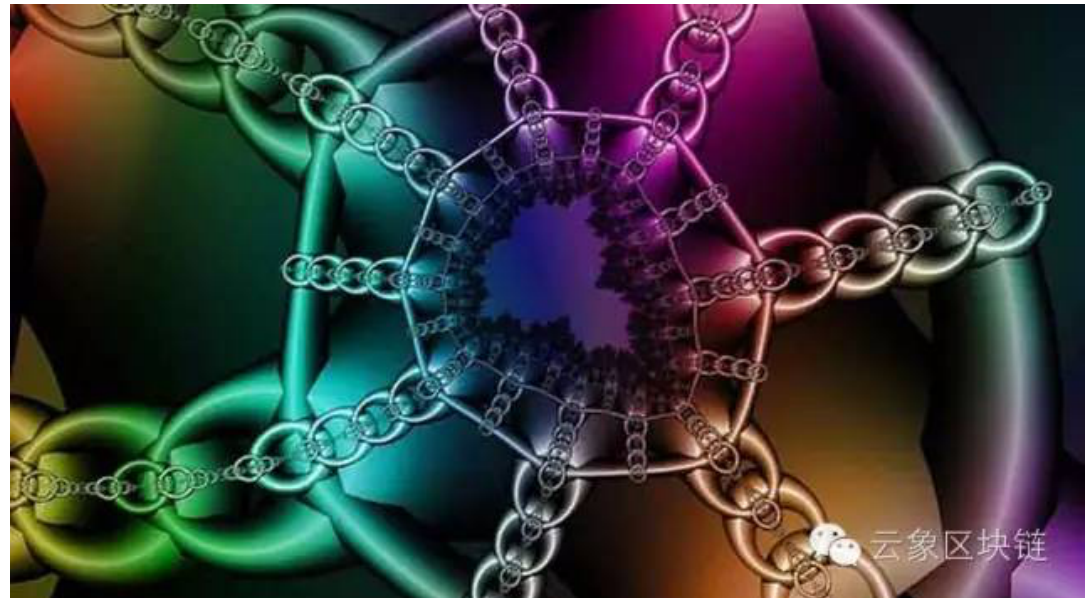
- 举例：用户去银行办理抵押贷款时对房产证信息的核实，对用户身份的核实，对权利人婚姻状况的核实……

- 去中心——靠什么？让数据透明、共享、不可篡改、真实可信

“区块链”来了——成了互联网新热词

区块链概念的产生

- **区块链**概念最早源于2009年化名为“中本聪”的一篇关于比特币的论文——区块链技术是为**比特币（一种数字货币）**而设计的一种数据库技术，基于**密码学的椭圆曲线数字签名算法（ECDSA）**来实现去中心化的P2P系统设计。
- 随着比特币应用的深入及广泛接受，区块链技术引起广泛关注。





比特币是区块链第一个成功应用



弗里德里希·冯·哈耶克
(Friedrich von Hayek)

革命性建议：
货币的非国家化

1931



米尔顿·弗里德曼
(Milton Friedman)

伟大灵感：
可以用计算机技术（数学）
来建立比国家信用更可靠
的货币体系。自动化装
置——程序发行货币

1976



中本聪
(Satoshi Nakamoto)

区块链论文：
(1)一种数据库技术——记录
比特币交易历史信息
(2)密码学技术实现的点对点
网络（P2P）分布式系统

2009

革命性建议和伟大灵感滋生了比特币。

但更为可贵的是支持比特币背后的一套无需中心机构支撑，**平稳运行8年**的自治系统——**区块链**

区块链的构成要素



数据结构

交易 (Transaction)

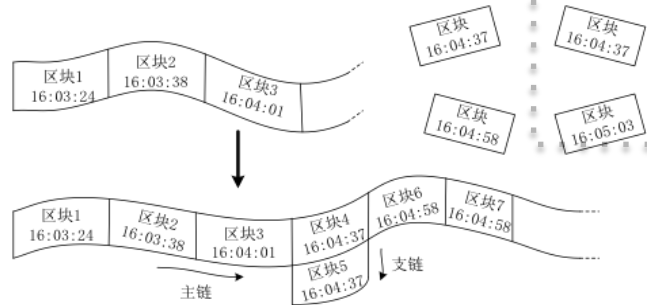
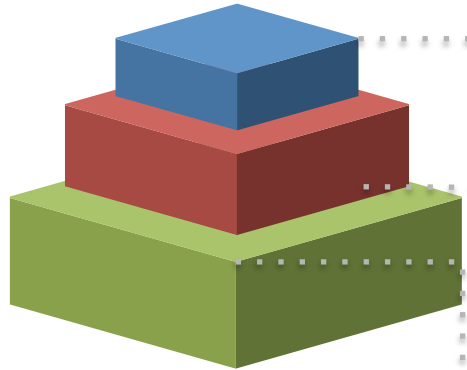
一次操作，导致账本状态的一次改变，如添加一条记录

区块 (Block)

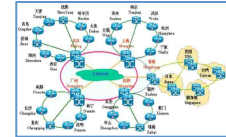
记录一段时间内发生的交易和状态结果，是对当前账本状态的一次共识；

链 (Chain)

由一个个区块按照发生顺序串联而成，是整个状态变化的日志记录



技术融合体



P2P网络通信
分布式系统



分布共识协议

交易的多方共识



智能合约

实现可编程的交易逻辑，
执行合约校验正确性



隐私保护

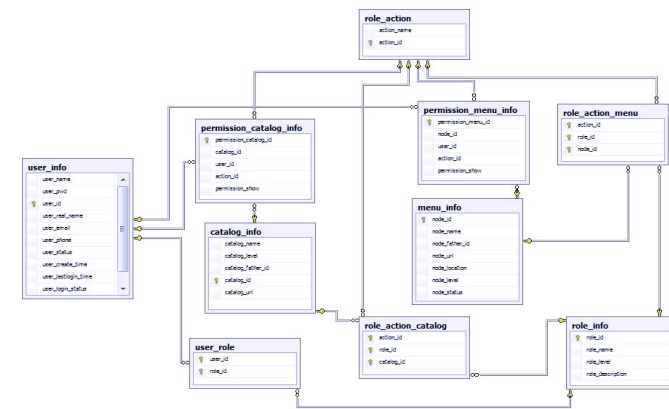
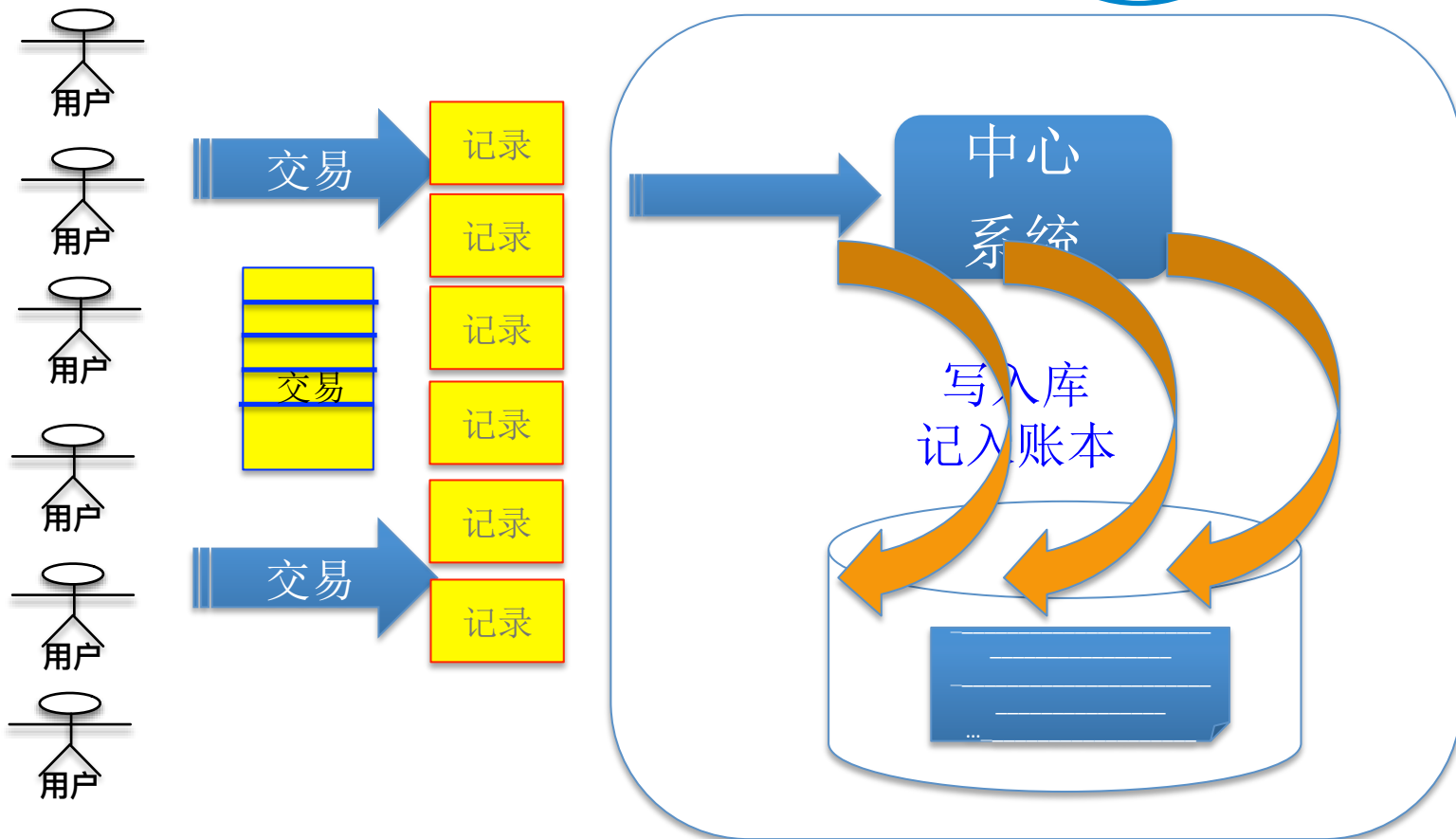
数字签名、加密解密实现
隐私安全



“区块+链”式数据库

区块+链式的结构，非关系的
日志型数据库

集中化系统



Student表

Sno (学号)	Sname (姓名)	Ssex (性别)	Sage (年龄)	Sdept (系名)
12001	王勇	男	21	CS
12002	刘阳	女	19	IS
12003	李敏	女	18	MA
12004	欧阳峰	男	23	IS

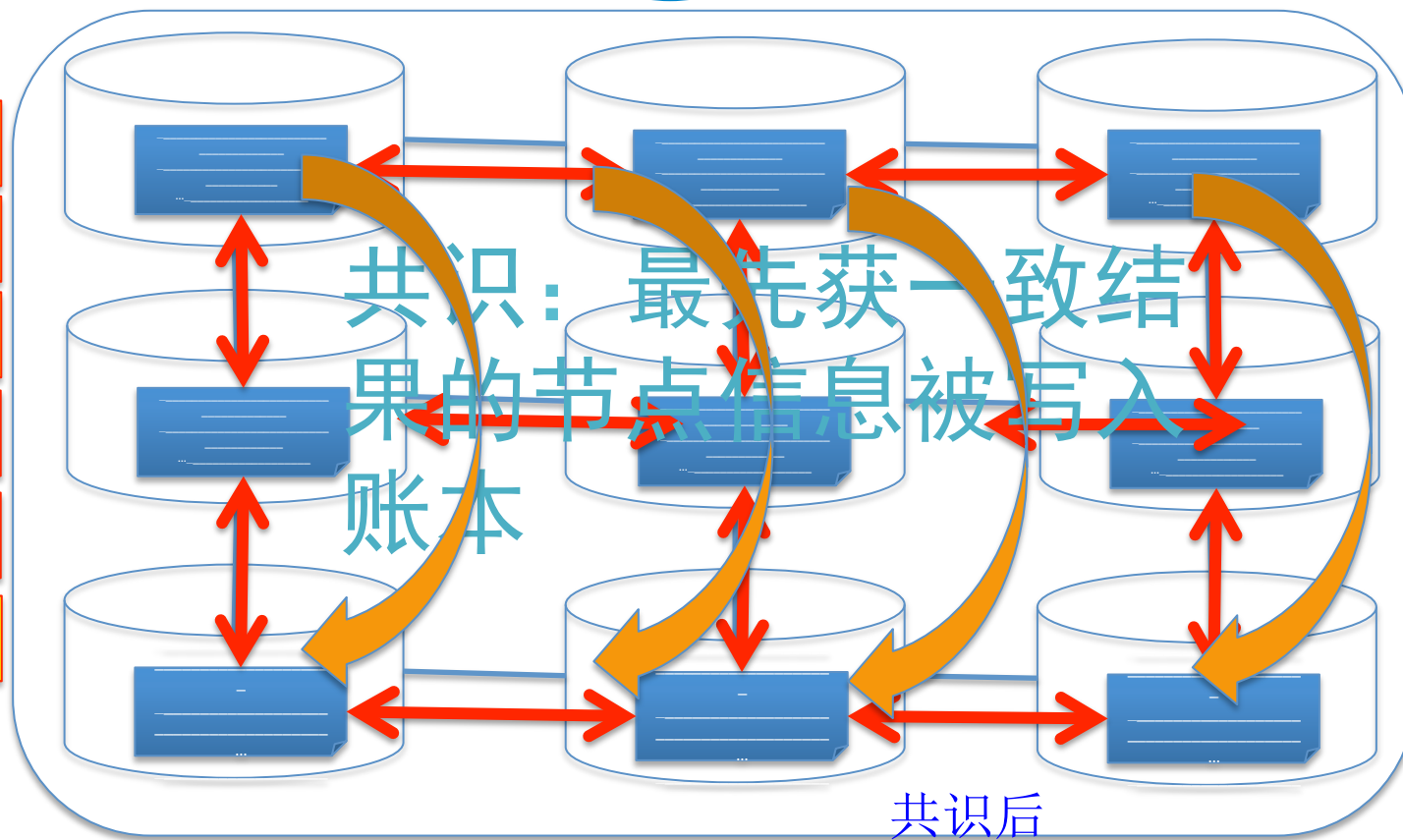
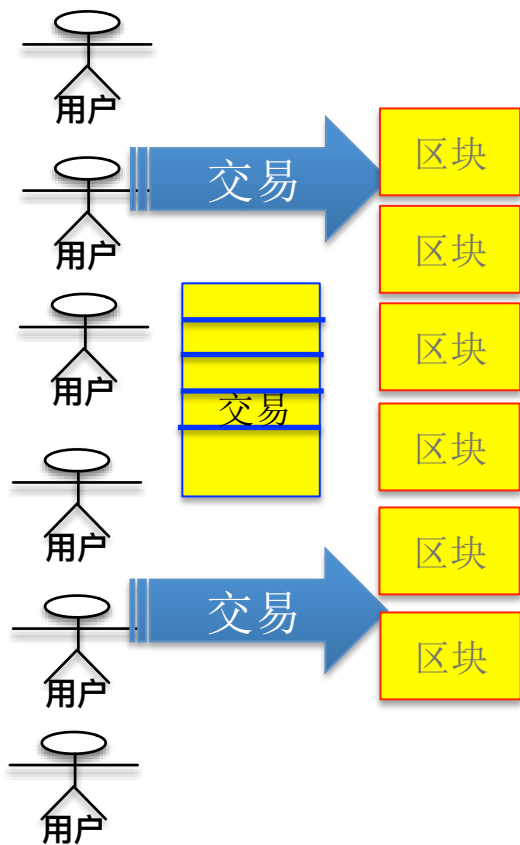
SC表

Sno (学号)	Cno (课程号)	Grade (成绩)
12001	1	82
12001	2	95
12001	3	88
12002	1	96
12002	3	93

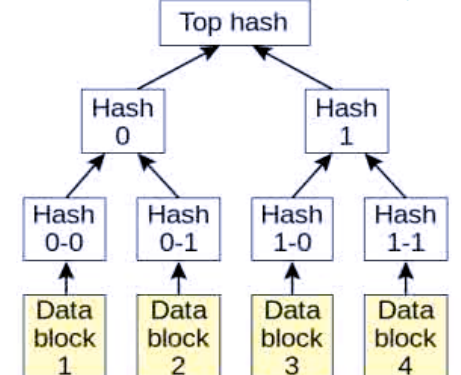


- 关系型数据可篡改
- 单点记录弱安全性

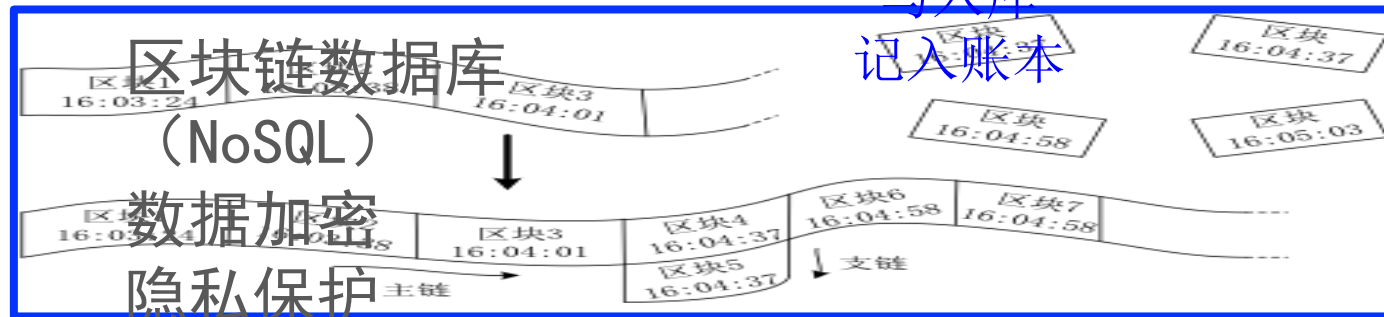
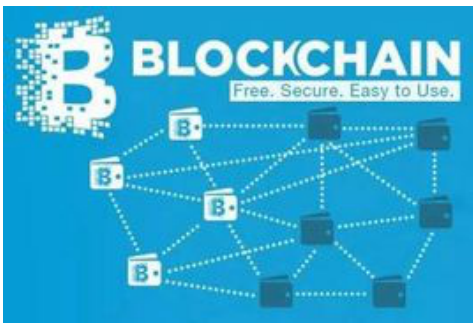
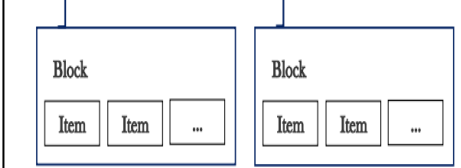
去中心——变集中到多中心



merkle哈希树
保证快速同步



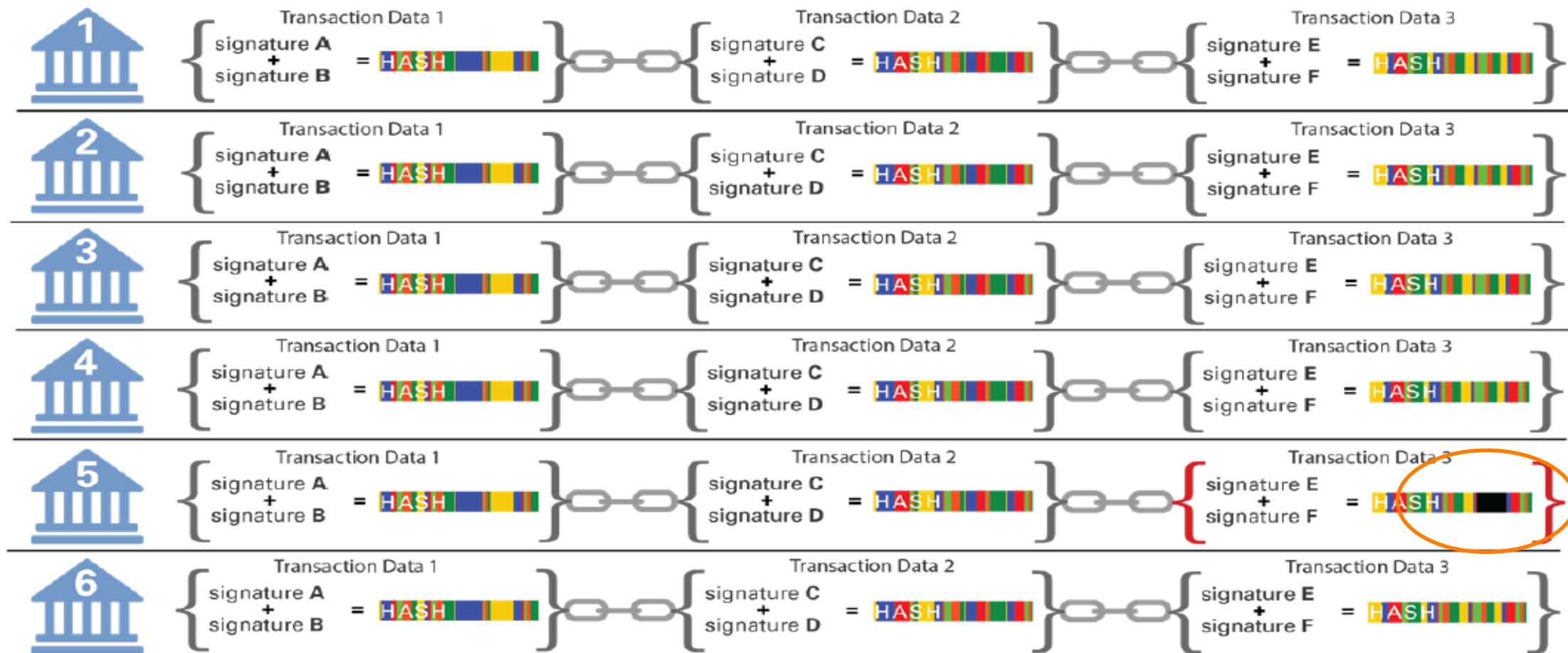
时间戳服务器
保证不可篡改



共识后
写入库
记入账本

分布式共识保证强一致性

Exhibit 2: The blockchain ledger is replicated across multiple locations (we show just six here for simplicity), and each maintains its own copy, which is separately updated based on new transaction data. We show a sequence of three transactions. In the first two transactions, data and signature information are properly validated by all six nodes with matching "hash" values. However for Transaction #3 at Location #5, the hash does not match the others, and will be corrected by the others via "consensus."



遭到攻击

技术特征与优势

技术特点

◆ 分布式容错性

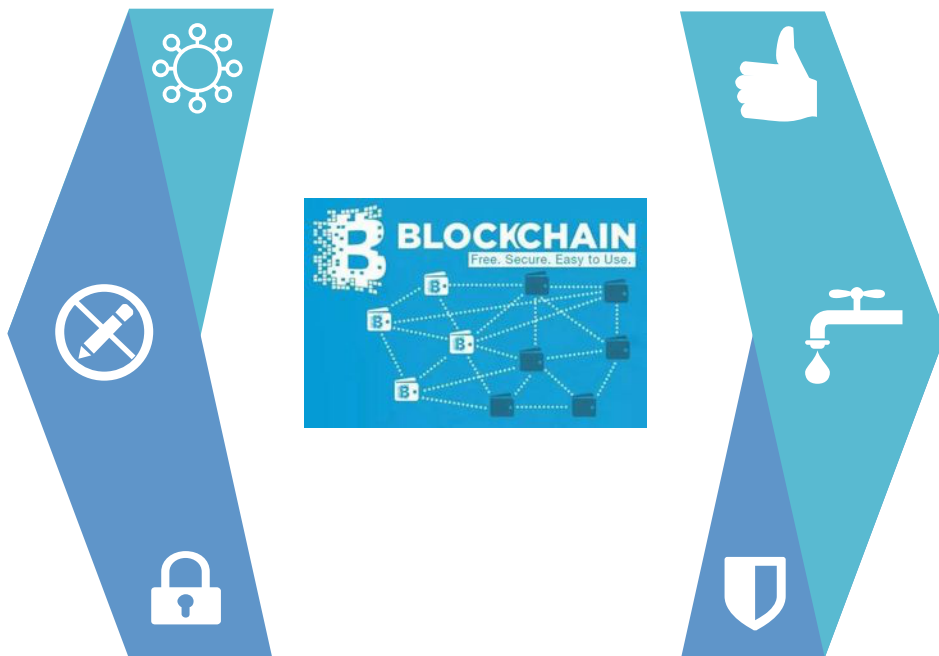
网络极其鲁棒，容错 1/3 左右节点的异常状态

◆ 不可篡改性

一致提交后的数据会一直存在，不可被销毁或修改

◆ 隐私保护性

密码学保证了未经授权者能访问到数据，但无法解析



业务特性

◆ 可信任性

提供可信的分布式账本平台，交易通过智能合约自动执行，不需要额外第三方中介机构

◆ 降低成本

跟传统技术相比，带来更短的时间、更少的人力和维护成本

◆ 增强安全

有利于安全可靠的审计管理和清算，减少犯罪可能性，和各种风险

适用场景

所有跟信息、价值（包括货币、证券、专利、版权、数字商品、实际物品等）、信用等相关的交换过程，都将可能从区块链技术中得到启发或直接受益

区块链的发展阶段

- **区块链1.0：可编程货币——比特币**

区块链构建了一种全新的去中心化的数字支付系统，随时随地的货币交易、毫无障碍的跨国支付以及低成本运营的去中心化体系都让这个系统变得魅力无穷。



- **区块链2.0：可编程金融——以太坊、超级账本**

基于区块链技术可编程的特点，人们尝试将“**智能合约**”的理念加入区块链中，形成了可编程金融。有了合约系统的支撑，区块链的应用范围开始从单一的货币领域扩大到涉及合约功能的其他金融领域。



- **区块链3.0：可编程社会——走向领域应用**



区块链技术又陆续被应用到了公证、仲裁、审计、域名、物流等其他领域中来，应用范围扩大到了整个社会。人们试图用区块链来颠覆互联网的最底层协议，让整个社会进入智能互联网时代，形成一个可编程的社会。



区块链的类型

	去中心化程度	权限和范围	经济奖励
公有链	完全去中心化	全球范围可以访问、交易	个人从中可获得的经济奖励与对共识过程做出的贡献成正比
联盟链	部分去中心化	读取、交易权限可设定	未知
私有链	中心化	写入权限仅在一个组织手里、读取权限可能被限制	不需要奖励，可能没有虚拟货币

区块链技术和应用标准

《中国区块链技术和应用发展白皮书》

2016年10月18日，由**工信部**指导的中国区块链技术和产业发展论坛成立并发布了《中国区块链技术和应用发展白皮书》，建议及时出台区块链技术和产业发展扶持政策，同时围绕产业发展的重点环节，逐步完善区块链技术应用和标准体系。



央行推进数字货币

数字货币是区块链技术在金融领域最广泛、最成功的应用

- 2014年，央行成立发行法定数字货币的专门研究小组，论证央行发行法定数字货币的可行性
- 2015年，对数字货币关键技术进一步深入研究，形成了系列研究报告
- 2016年1月20日，央行召开的数字货币研讨会，进一步明确了发行数字货币的战略目标
- 央行发行法定数字货币的原型方案也已完成两轮修订，原型系统Demo有望在2017年发布
- 2016年11月，中国人民银行印制科学研究所公开招聘相关专业人员，从事数字货币研究与开发工作



02

联盟链架构和智能合约

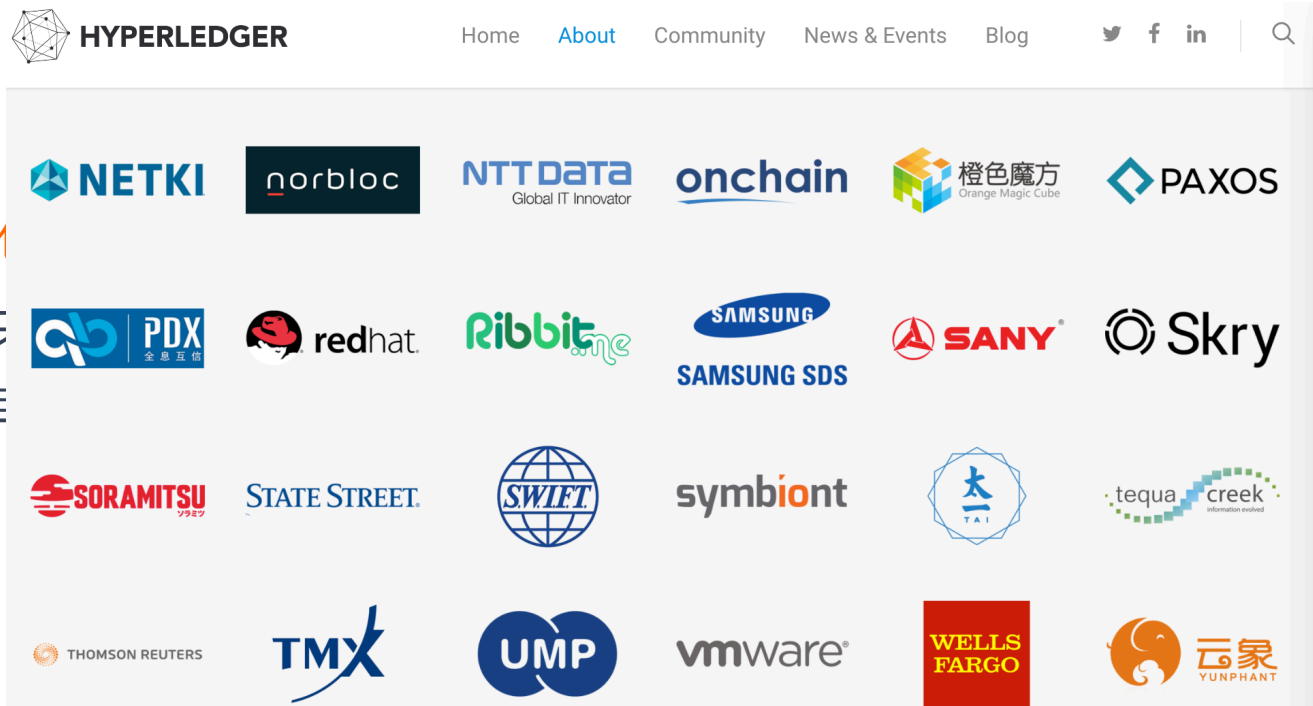
分布式账本-Hyperledger-fabric项目

- 分布式账本技术已经成为联盟链的主流技术——Hyperledger（超级账本）
 - 2016年由Linux基金会发起的开源HyperLedger（超级账本）平台——成为最早公开的联盟链代码。
 - HyperLedger旨在成为跨行业的区块链技术的标准，多个不同行业公司都在平台上贡献提交自己的代码。
 - 其中最活跃最被认可的，作为基础设施的项目是由IBM推出的Fabric项目。
 - 该项目试图打造一个透明、公开、去中心化的超级账本项目，作为区块链技术的开源规范和标准，让更多的应用能更容易的建立在区块链技术之上。

Hyperledger项目及成员

- Linux基金会是牵头单位，联合 30 家初始成员（DTCC、FUJITSU、HITACHI、SWIFT、Cisco 等），
- 目前已经有100多成员单位加入，如云象是国内早

<https://www.hyperledger.org/>



Hyperledger fabric的逻辑架构

□ 成员管理服务（Membership service）

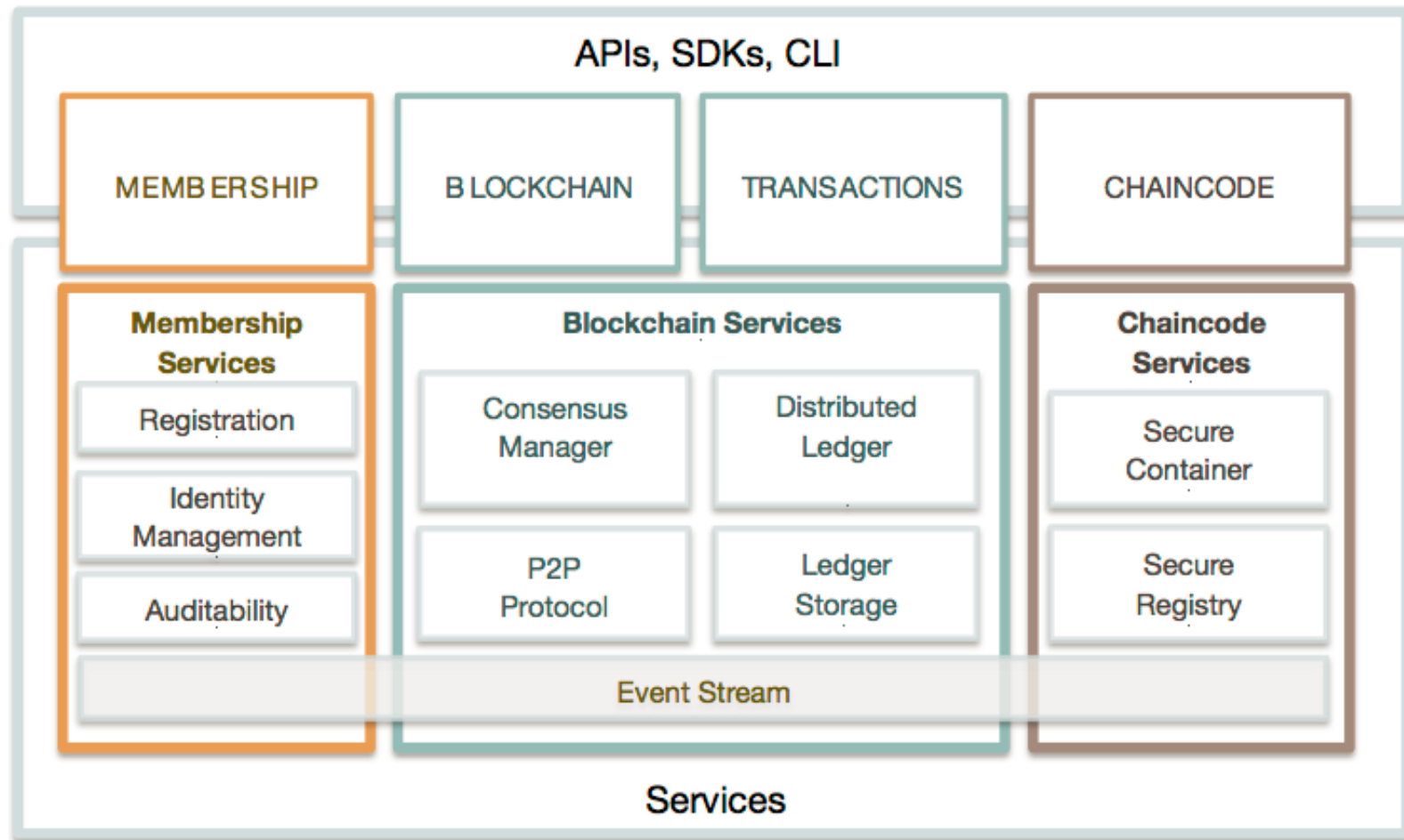
提供区块链系统的成员管理，包括身份管理、隐私、保密和审计性的服务

□ 区块链服务（Blockchain service）

提供基于区块链和交易形成的分布式总账的管理服务

□ 智能合约服务（链码服务，Chaincode service）

提供智能合约运行和管理服务



区块链系统构建的核心功能

会员管理功能

实现区块链用户成员的增加、修改和删除等功能，成员权限功能



代币服务

提供了代币功能支持的相关模块，为实现区块链上的基于代币的交易系统提供接口服务

智能合约服务

实现交易业务规则，包括合约的编辑、执行、调度功能。资产登记时创建资产合约



隐私保护服务

用密钥和数据加密算法将信息进行加密，在网络中传输和存储，保证数据的安全性

共识验证服务

实现交易区块信息在全网的共识，支持共识需要共识协议的支持，主要的协议有noops，BFT, PBFT等



数据库服务

实现跟区块链数据库的访问，类型是NoSQL数据库

节点监控服务

实现了监控节点的运行状态，还实现了节点故障预警和自动切换功能



链浏览器与开发平台

查看和统计区块链中的信息，基于此平台，用户可以开发智能合约并部署到区块链上

什么是智能合约



用户角度：智能合约通常被认为是一个自动担保账户，例如，当特定的条件满足时，程序就会释放和转移资金。**合约：构建信任的工具，用合约替代支付宝，合约充当第三方中介。**

案例：甲乙交易
基本流程：
甲方付款X给乙方
乙方交货物给甲方
双方基于智能合约
实现互信交易

甲方



-X

1.甲方发起交易申请
创建交易智能合约

2.冻结本地账户X元
合约账户+X元

4.甲方收到乙方货物
后，再次查看合约，
并在合约上确认

智能合约

合约账户



合约条款

写入DB

3.乙方查看交易智能合约
在合约上确认交易申请，
并将交易货物发给甲方

5. 系统得知双方确认后，
触发区块链合约条款
自动执行，
合约账户X元生效，
更新乙方账户+X
更新甲方账户-X

乙方



+X



技术角度：用一段程序代码的形式实现了可校验的业务流程和业务逻辑，被部署在区块链上。所有交易信息的记录，可以通过合约的再次执行进行校验并重新生成信息。

智能合约通过共识永远记入区块链

合约部署

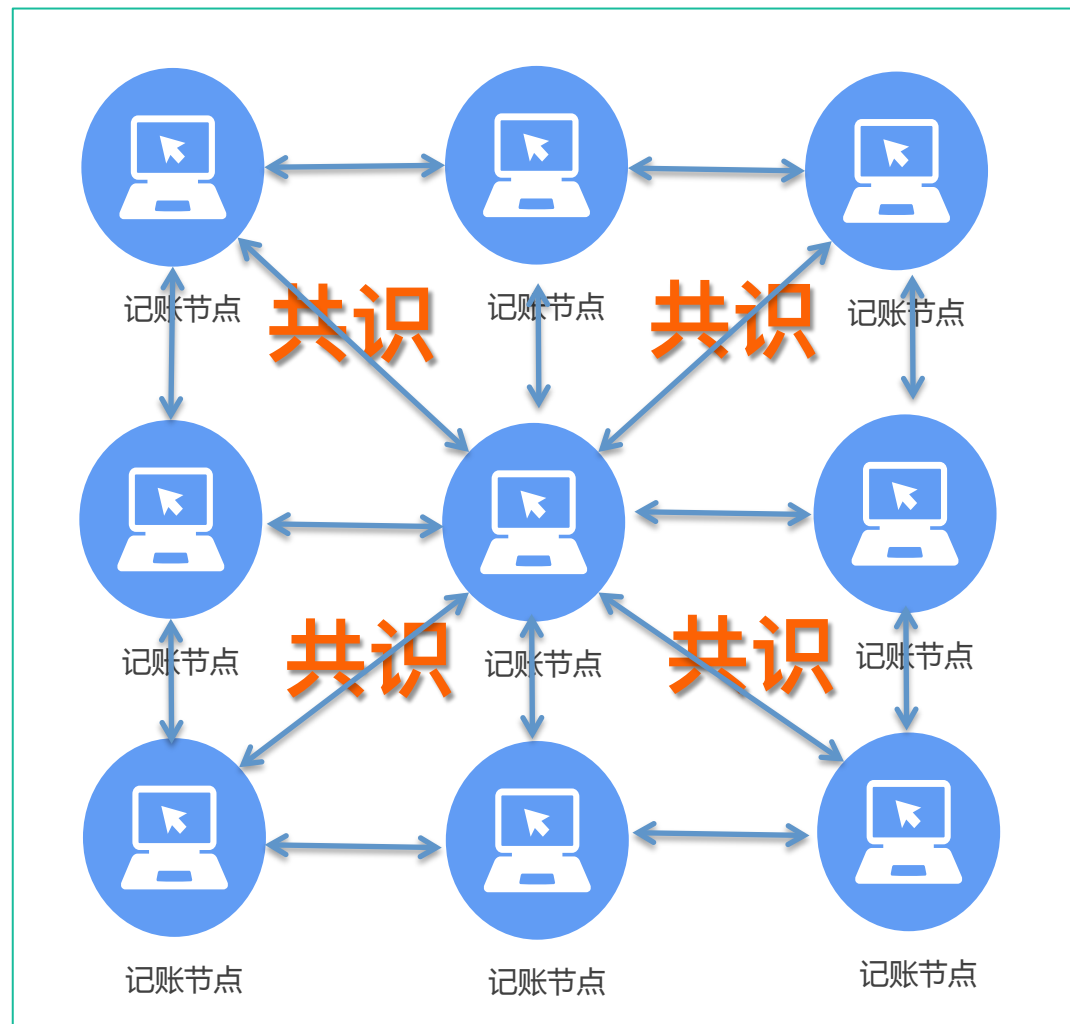
由一个授权用户（管理员）向一个节点发出部署请求，广播到所有记账节点进行共识，共识验证无误被写入账本。

合约调用

在一个特定的业务流程中，在一次具体的操作中合约被业务系统调用执行。

查看合约

合约部署后，被记入账本，要查看合约，可以通过访问账本的查询函数，传入合约的哈希地址，就可以查看具体的合约内容。



合约代码结构

hyperledger的智能合约demo解析
样例: chaincode_example2

.go文件

数据存储 基本函数

数据存储：
存储变量、结构、数组、映射
基本函数：
初始化Init
调用Invoke
查询Query

```
type SimpleChaincode struct {
}

func (t *SimpleChaincode) Init(stub shim.ChaincodeStubInterface,
function string, args []string) ([]byte, error) {...}
init函数在起始过程中初始化两个账户A, B, 并且在A, B账户上发行一定数量的资产。

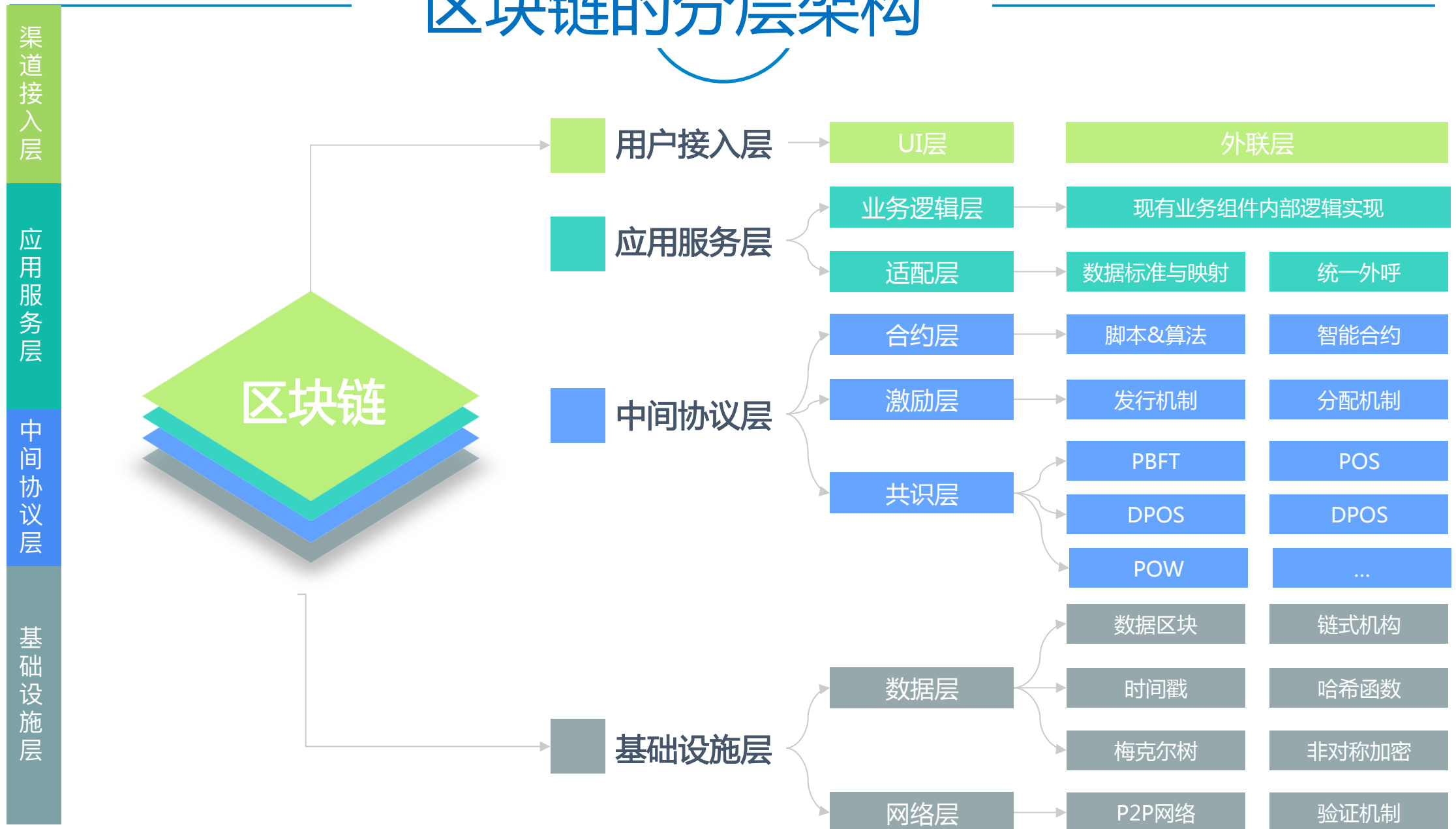
func (t *SimpleChaincode) Invoke(stub shim.ChaincodeStubInterface,
function string, args []string) ([]byte, error) {...}
invoke函数在A、B上进行转账。

func (t *SimpleChaincode) Query(stub shim.ChaincodeStubInterface,
function string, args []string) ([]byte, error) {...}
query函数查询A、B函数上的账户余额。

func main() {
    err := shim.Start(new(SimpleChaincode))
    if err != nil {
        fmt.Printf("Error starting Simple chaincode: %s", err)
    }
}
```

<http://blog.csdn.net/mcdullsin/article/details/51670780>

区块链的分层架构



03

案例-供应链金融

供应链金融

供应商、制造商、分销商、零售商、金融机构

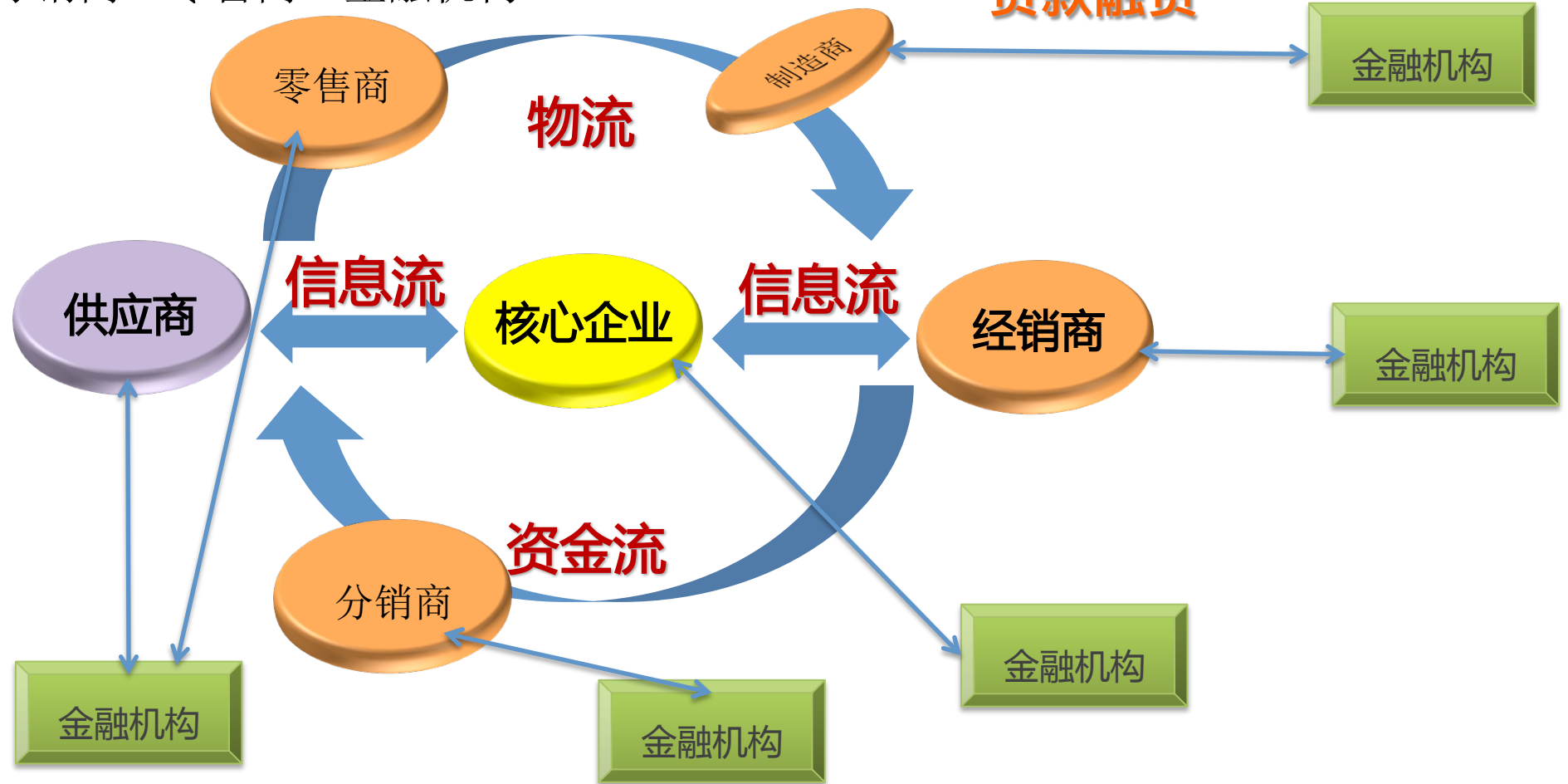
采购

产品

销售

贷款

融资



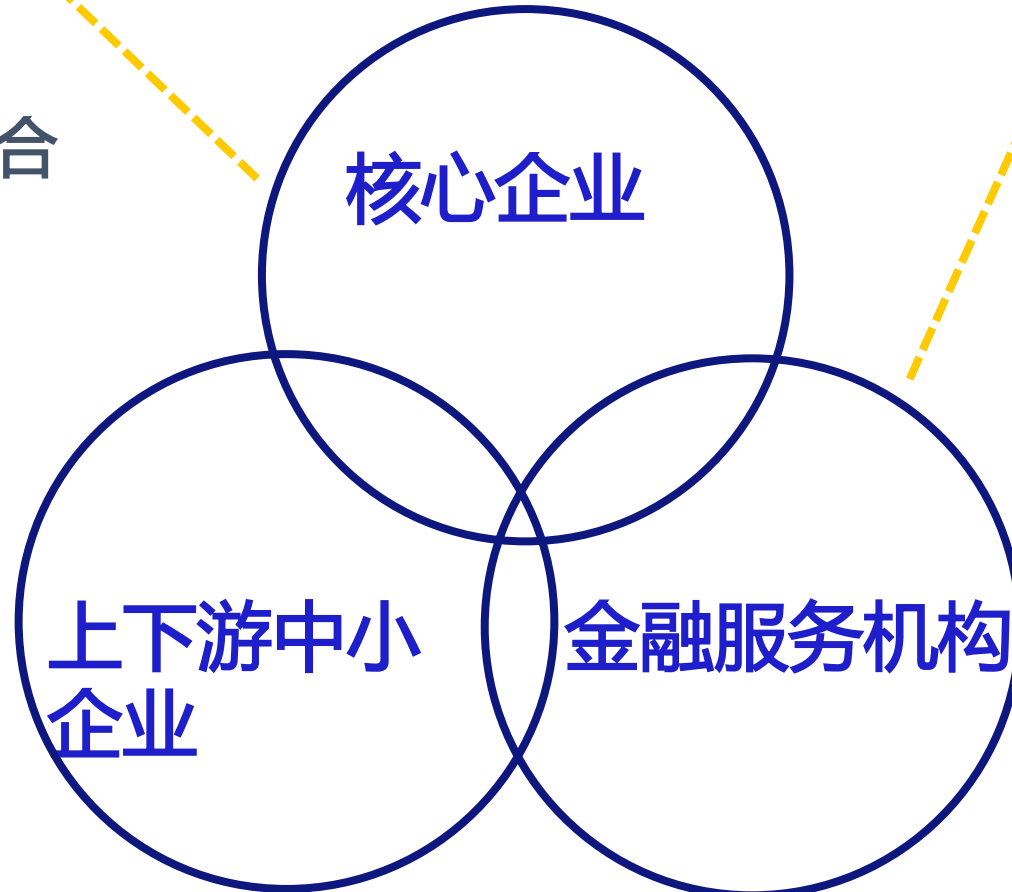
依托真实贸易、整体授信、新型融资服务

供应链金融解决的问题

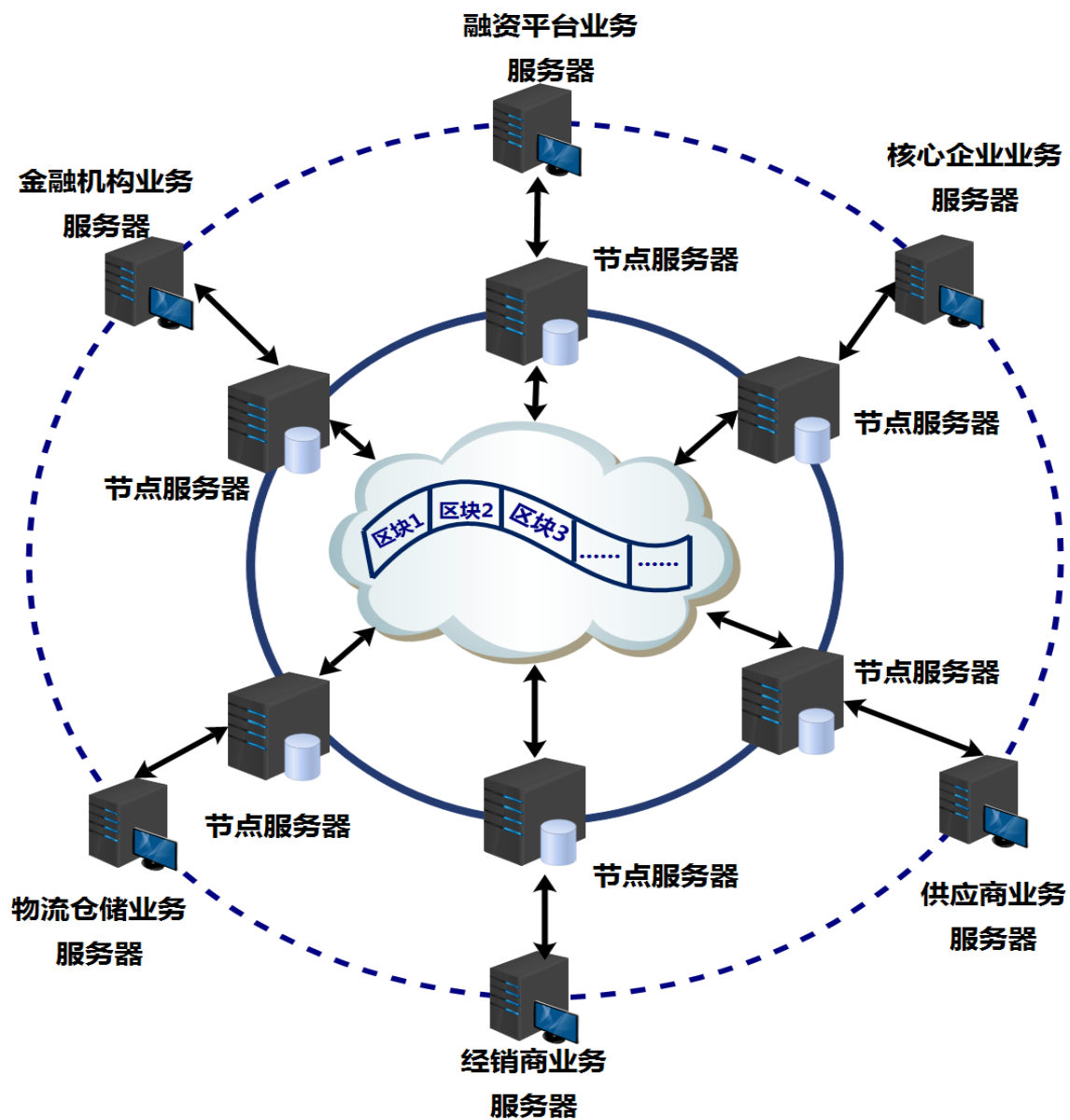
- 提升产业链的效率
- 加速资金流动
- 升级转型，产金结合

- 差异化竞争
- 扩大中小企业市场
- 增加客户粘性

- 融资困难问题



构建供应链金融联盟



◆构建区块链联盟

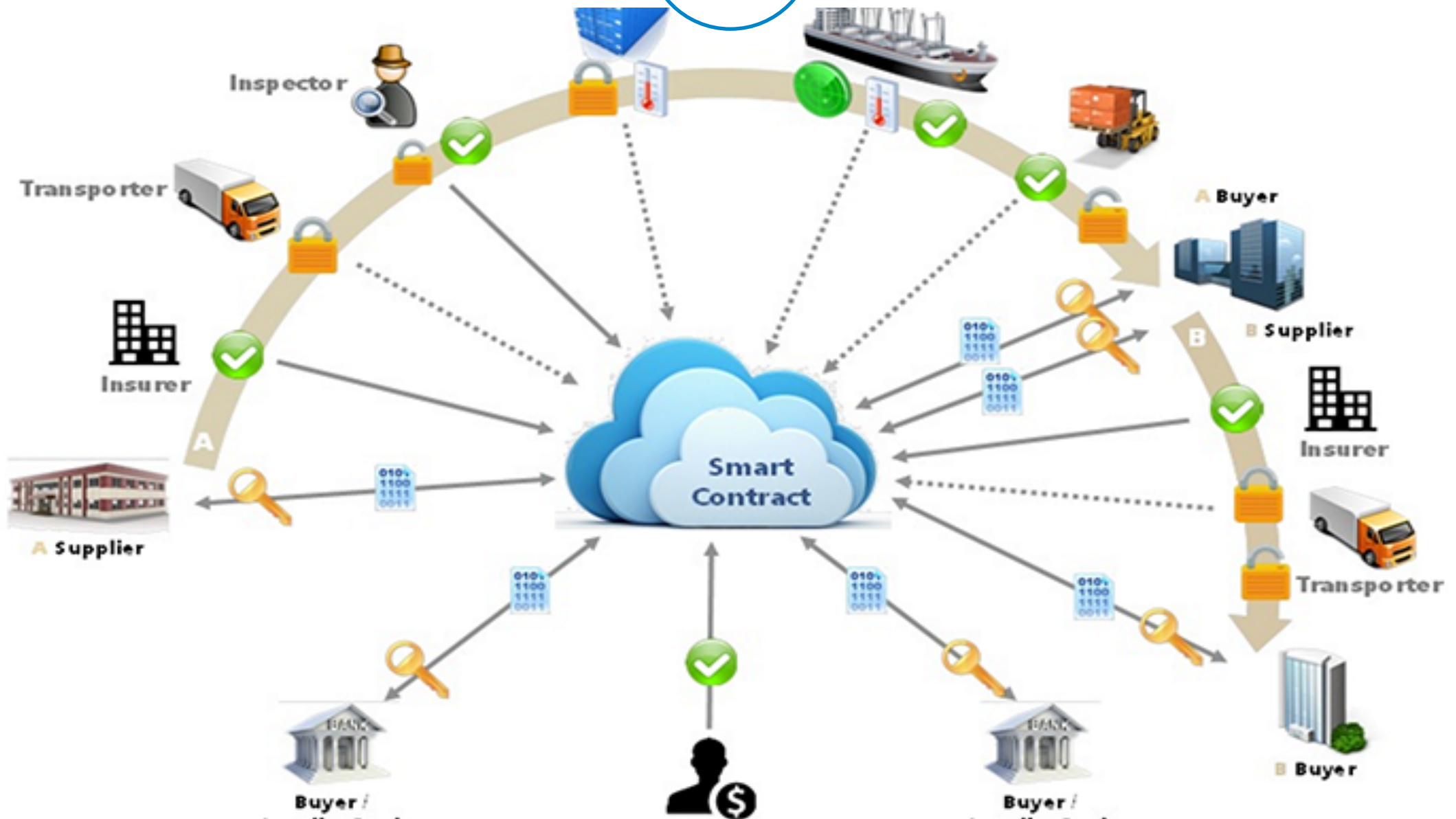
◆选择产业链

◆确定联盟成员，共同维护交易账簿

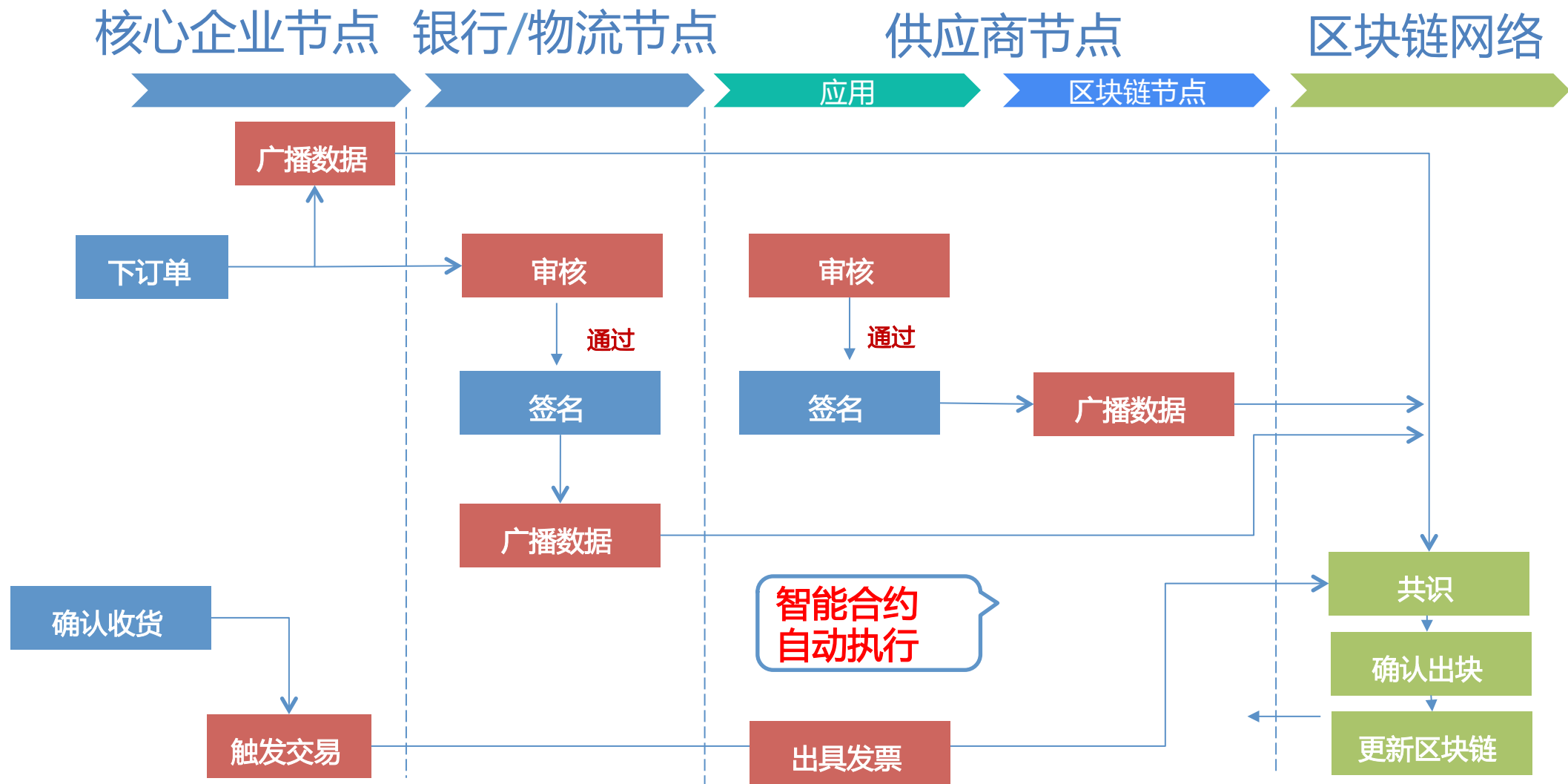
◆确定共识机制与证书发放机构

◆根据业务，制定联盟合约规则

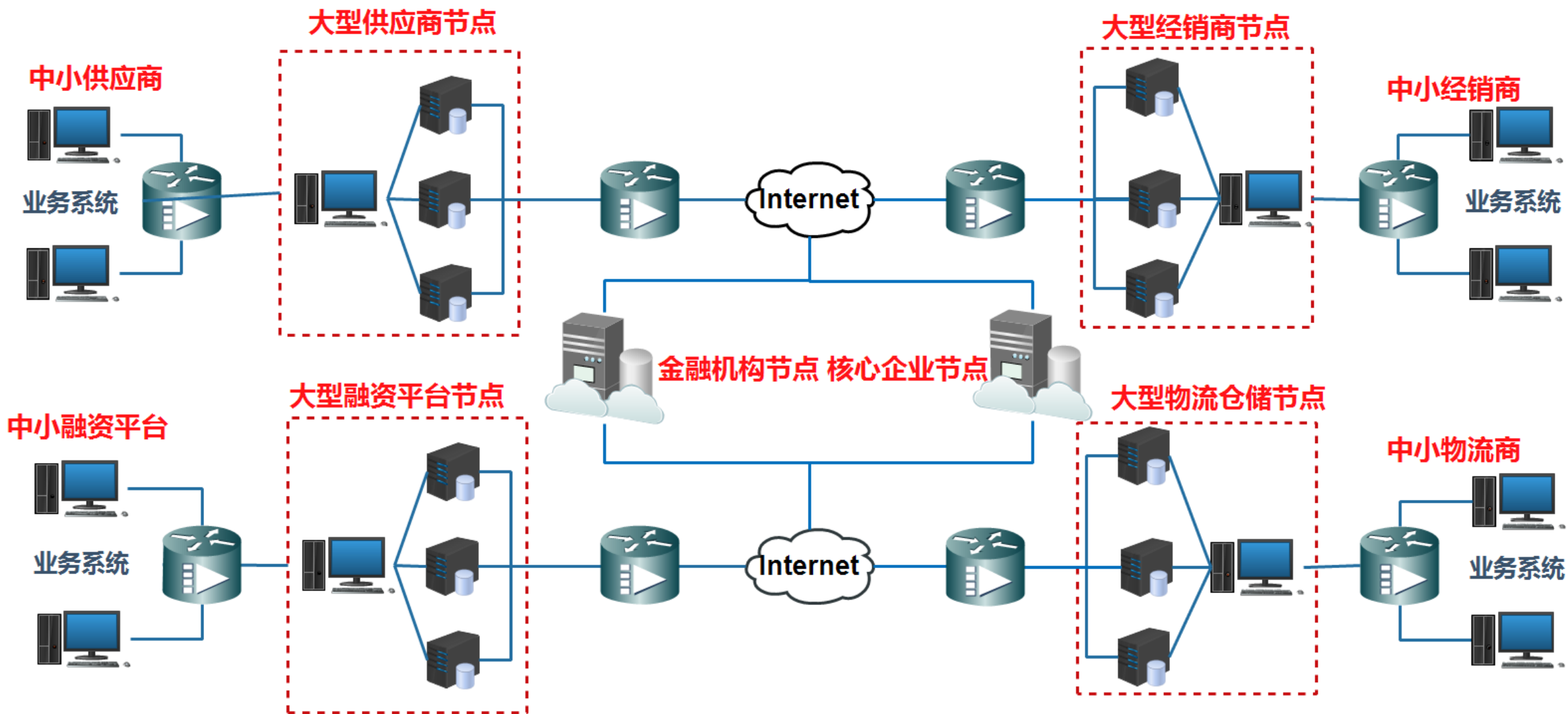
开发智能合约与各应用系统对接



智能合约：订单签订流程



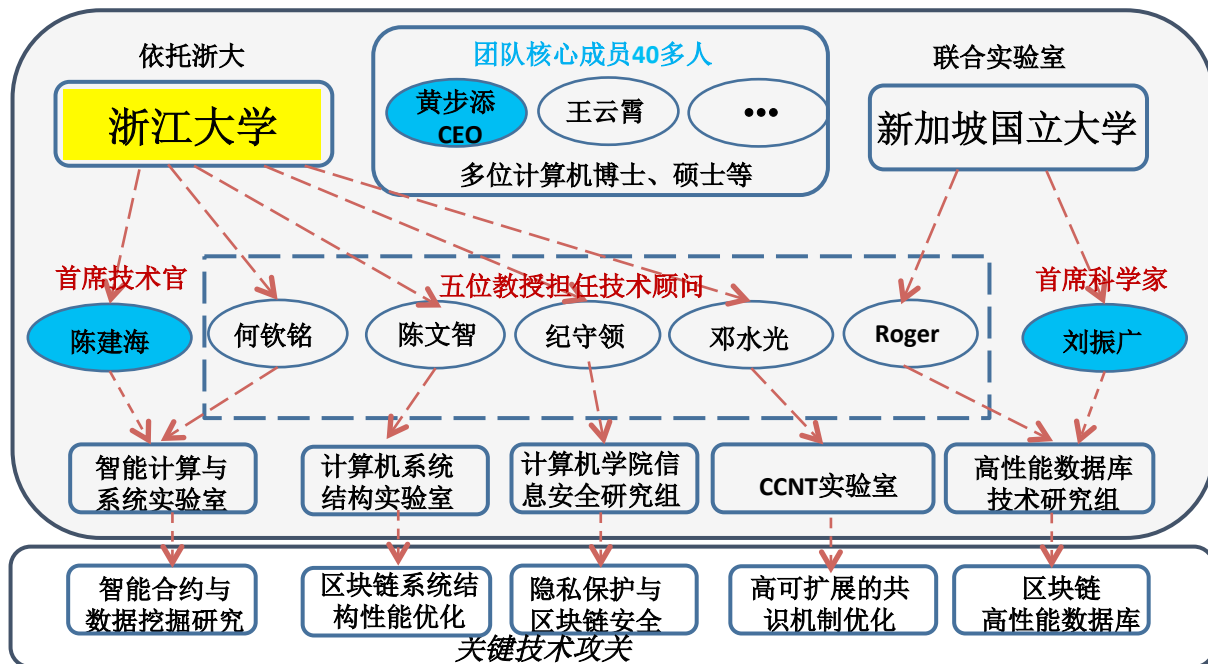
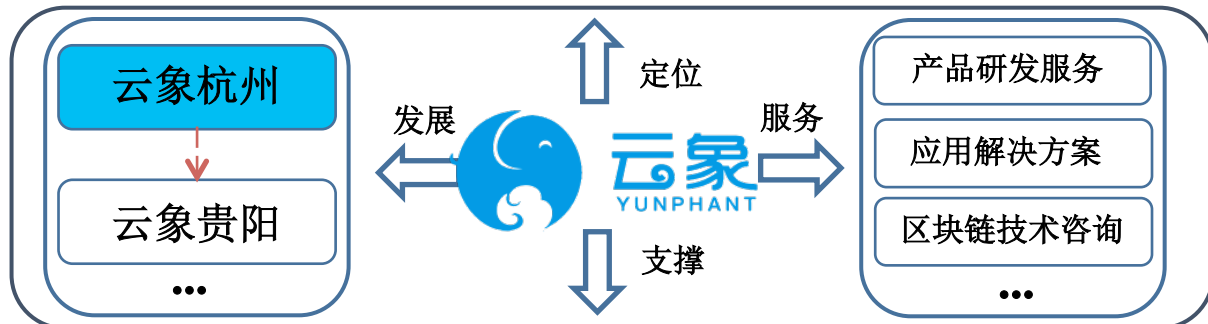
系统部署的架构



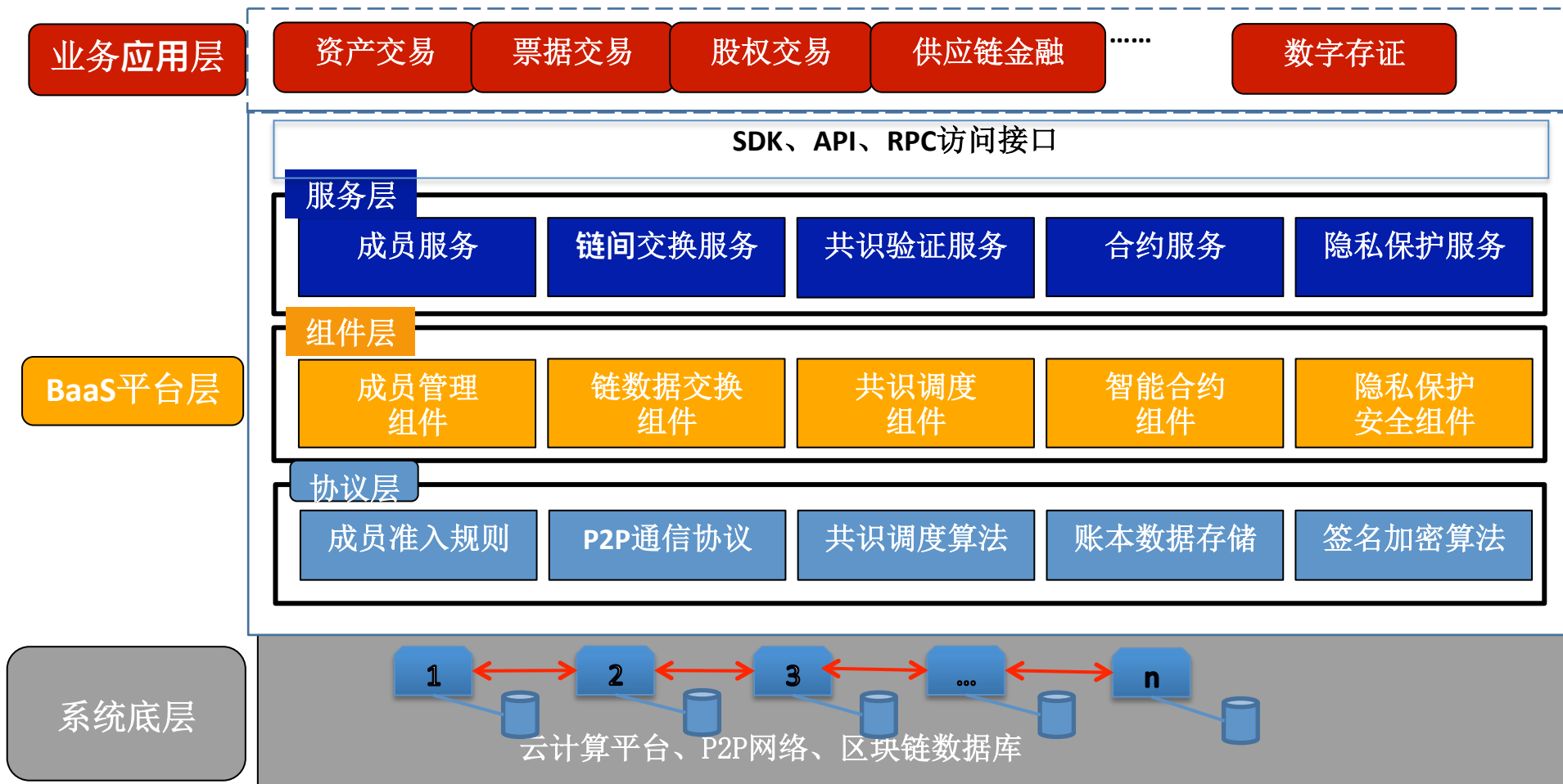
为应用提供支撑的云象区块链平台

打造一流的企业级联盟链技术平台

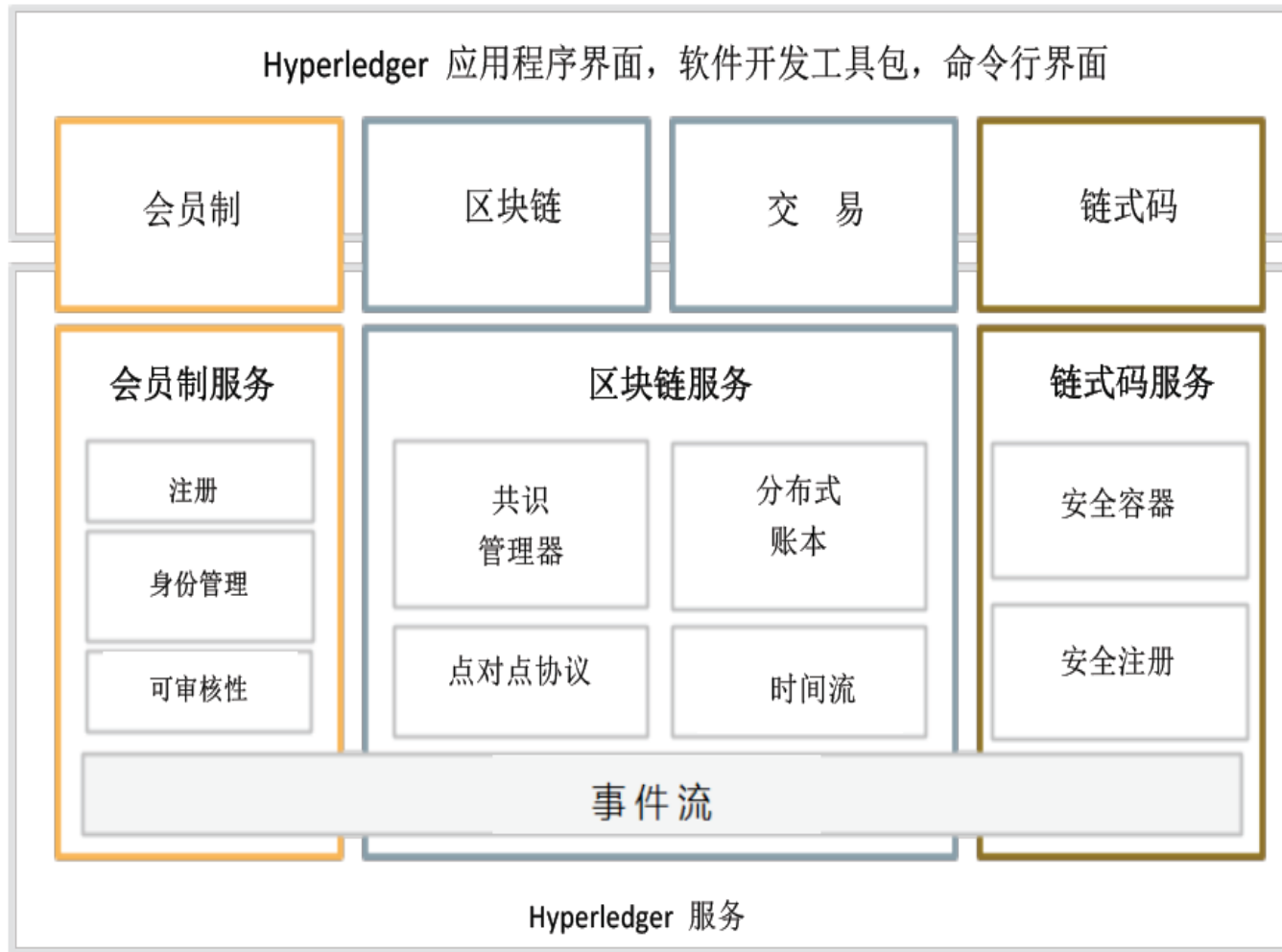
(云象区块链BaaS平台、隐私保护中间件、智能合约套件、区块链数据库产品等)



联盟区块链技术平台架构



区块链平台架构的优化



➤ 会员服务优化

- 节点动态增加，不影响业务处理
- 成员准入控制优化，策略可配置

➤ 区块链服务优化

- 区块定时同步，间隔可配置
- 节点交易去重，避免重复写入
- 共识消息处理优先，提升交易数量较多时的共识响应速度
- 交易类型参数配置，提升处理效率

➤ 合约服务优化

- 合约部署去除源码，提升安全性
- 虚拟机优化，支持多线程，提升吞吐率
- 虚拟机自动重启

➤ 区块链监控平台

- 交易、节点状态可视化监控
- 运行日志监控
- 设计轻客户端，提升证书安全性

基于Hyperledger架构，提供区块链应用一体化解决方案

云象区块链开发者平台

节点日志

```

18:16:25.386459 [consensus/pbft] ProcessEvent -> INFO 56d: Replica 0[V:397] received view change quorum, processing new view
18:16:25.386506 [consensus/pbft] receiveViewChange -> INFO 56d: Replica 0[V:397] received view change from replica 1, v:397, h:3965, [C]:2, [P]:5, [Q]:5
18:16:25.386807 [consensus/pbft] receiveViewChange -> INFO 56d: Replica 0[V:397] now has 4 view change requests for view 397
18:16:25.388841 [consensus/pbft] ProcessEvent -> INFO 56f: Replica 0[V:397] received view change quorum, processing new view
18:16:25.389868 [consensus/pbft] receiveViewChange -> INFO 56e: Replica 0[V:397] received view change request from replica 1
18:16:25.311875 [consensus/pbft] processNewView -> INFO 56e: Replica 0[V:397] accepting new-view to view 397
18:16:25.311815 [consensus/pbft] processNewView -> INFO 56e: Replica 0[V:397] updateViewChangeSeqNo
18:16:25.311811 [consensus/pbft] updateViewChangeSeqNo -> INFO 56e: Replica 0[V:397] updating viewChangeSeqNo to 3980
18:16:57.311993 [consensus/pbft] executeSync -> INFO 56e: Replica 0 executing/committing null request for view397/seqNo=3971
18:16:57.311979 [consensus/pbft] executeSync -> INFO 56e: Replica 0[V:397] finished execution 3971, trying next
18:16:45.806887 [rest] processChaincodeInvokeQuery -> INFO 56f: REST processing chaincode request...
18:16:45.806838 [rest] processChaincodeInvokeQuery -> INFO 56f: REST invoke chaincode...
18:16:45.806806 [rest] processChaincodeInvokeQuery -> INFO 56e: Local user 'test_msp@' is already logged in. Retrieving login token.
18:16:45.806825 [devops] invokeQuery -> INFO 56f: Transaction ID: 516c1386-9f98-4b13-9774-43c6a24e48c
18:16:45.806833 [crypto] InitClient -> INFO 56e: Initializing client [test_msp]...
18:16:45.806833 [crypto] closeClientInternal -> INFO 56e: Closing client [test_msp]...
18:16:45.806866 [rest] processChaincodeInvokeQuery -> INFO 56e: Successfully submitted invoke transaction with txid (516c1386-9f98-4b13-9774-43c6a24e48c)
18:16:45.822851 [rest] processChaincodeInvokeQuery -> INFO 56e: REST successfully submitted invoke transaction: {"jsonrpc": "2.0", "result": {"status": "OK", "message": "516c1386-9f98-4b13-9774-43c6a24e48c"}, "id": 1}
18:16:46.784848 [devops] invokeQuery -> INFO 56f: Local user 'test_msp@' is already logged in. Retrieving login token.
18:16:46.784872 [rest] processChaincodeInvokeQuery -> INFO 56f: REST invoke chaincode...
18:16:46.784867 [crypto] InitClient -> INFO 56f: Local user 'test_msp@' is already logged in. Retrieving login token.
18:16:46.784848 [devops] invokeQuery -> INFO 56f: Transaction ID: 5d2212d5-b8d8-46f1-b244-22d8b1f8b1a
18:16:46.834876 [crypto] InitClient -> INFO 56f: Initializing client [test_msp]...
18:16:46.834876 [crypto] closeClientInternal -> INFO 56f: Closing client [test_msp]...
18:16:46.835116 [rest] processChaincodeInvokeQuery -> INFO 56f: REST successfully submitted invoke transaction with txid (5d2212d5-b8d8-46f1-b244-22d8b1f8b1a)
18:16:46.835077 [rest] processChaincodeInvokeQuery -> INFO 56f: REST successfully submitted invoke transaction: {"jsonrpc": "2.0", "result": {"status": "OK", "message": "5d2212d5-b8d8-46f1-b244-22d8b1f8b1a"}, "id": 1}
18:16:47.895875 [devops] invokeQuery -> INFO 56f: Local user 'test_msp@' is already logged in. Retrieving login token.
18:16:47.895874 [rest] processChaincodeInvokeQuery -> INFO 56f: REST invoke chaincode...
18:16:47.895874 [crypto] InitClient -> INFO 56f: Local user 'test_msp@' is already logged in. Retrieving login token.
18:16:47.895813 [devops] invokeQuery -> INFO 56f: Transaction ID: 22465795-d554-476c-b8ef-87f78b0e5d87
18:16:47.895807 [crypto] InitClient -> INFO 56f: Initializing client [test_msp]...
18:16:47.895797 [crypto] closeClientInternal -> INFO 56f: Closing client [test_msp]...
18:16:47.895801 [rest] processChaincodeInvokeQuery -> INFO 56f: Successfully submitted invoke transaction with txid (22465795-d554-476c-b8ef-87f78b0e5d87)
18:16:48.844983 [rest] processChaincodeInvokeQuery -> INFO 570: REST processing chaincode request...
18:16:48.845116 [rest] processChaincodeInvokeQuery -> INFO 570: REST invoke chaincode...
18:16:48.845121 [rest] processChaincodeInvokeQuery -> INFO 570: Local user 'test_msp@' is already logged in. Retrieving login token.
18:16:48.845454 [devops] invokeQuery -> INFO 570: Transaction ID: a93a2ab-38d3-44d8-b5fb-5258a1c217d
18:16:48.845479 [crypto] InitClient -> INFO 570: Initializing client [test_msp]...
18:16:48.845454 [crypto] closeClientInternal -> INFO 570: Closing client [test_msp]...
18:16:48.845933 [rest] processChaincodeInvokeQuery -> INFO 570: Successfully submitted invoke transaction with txid (a93a2ab-38d3-44d8-b5fb-5258a1c217d)
18:16:48.845910 [rest] processChaincodeInvokeQuery -> INFO 570: REST successfully submitted invoke transaction: {"jsonrpc": "2.0", "result": {"status": "OK", "message": "a93a2ab-38d3-44d8-b5fb-5258a1c217d"}, "id": 1}
18:16:49.312538 [consensus/pbft] executeSync -> INFO 570: Replica 0[V:397] finished execution 3972, trying next
18:16:49.312529 [consensus/pbft] executeSync -> INFO 570: Replica 0[V:397] finished execution 3972, trying next
18:16:49.794863 [rest] processChaincode -> INFO 570: REST processing chaincode request...
    
```

节点日志

区块链浏览器

区块信息

区块号	区块哈希	状态哈希	详情
1	GQWt11m/mAQkxp0vXwZicYnshMwpU9G4/dqZCxd1R	e7f1NTD+5W1b0h6tEg1bEIMHdwyoMCXadq7YOZMqGC	查看
6	cGnls08MHeeWWQ37nG/F87j4hKBds9QCJSzGTgymlQ==	92/KduCd1w/Zk4Hk3FopL8XCpab79BGCxGoR-HIQ==	查看
1	D3ATvPRIN8TWzNocmaoOe9k3DDJaM562pZRP5N4+xn/	e+DwxdGibKMnY3GRsMtlr6bCRX1jNQC3zsd/n3av31YEfO	查看
5	7a/W5Fmf3mVvoWd235dm45+5lRqpY7AppZuL1jg==	oUaicuCte0V/xStDwnL+7gKr4LDZCXfdtPw==	查看
1	mi/QhMclJgxcU4zQwA5NTEaY3nMI3VGgVLxk7ID2BIMbb	WA8KxMoxItq+qoONW6Fd6kVXdEY82QV3YSk8dVskVQQ	查看
4	Wb5aP5fcaWBGQY3Hut5VE8RKEiQPKh7QIK9P2YA==	+VWR5B1HPx8BEWkZokuk7XT8H6l8W55i5yj/OeauQ==	查看
1	9f9wL+NhZppXjJdV5i14FiSNbonum/otV/p6u+rcPEM+rlm	/WljiE8yD/KYBeyirG/3w0hY8XRv6cpeRgkUgtb6Mud9UH	查看
3	aj1WU5VMdVvEwBybnXsFu1+IAEhX2sv6C5Qg==	CO/Kw7V8T8JqruR0wgXG6jLTSzdnK/VQCpPg==	查看

智能合约编辑器

区块链监控平台

节点操作

类型	名称	地址	状态	操作
MemberSvc	membersvc	116.62.216.208:17054	复制 ● Running	▶ ⏸ 🗑
Peer	vp0	116.62.216.208:17050	复制 ● Running	▶ ⏸ 🗑
Peer	vp1	116.62.216.208:17150	复制 ● Running	▶ ⏸ 🗑
Peer	vp2	116.62.216.208:17250	复制 ● Running	▶ ⏸ 🗑
Peer	vp3	116.62.216.208:17350	复制 ● Stopped	▶ ⏸ 🗑

core.yaml config.yaml

最新区块 #16

交易吞吐量 0tps

区块时间 2017-05-05 17:24:01

活跃节点 4 / 4

交易数量

区块产生延时

节点拓扑

平台特色及优势

- 平台性能：提升原生fabric的**500 TPS**到 **3000**笔
- **28**项区块链专利发明
- 从底层协议到成员准入组件、隐私保护组件、链间信息交换、智能合约等组件再到应用层均自主研发
- 从物理网络、应用到数据各个维度的全方位**安全**解决方案
- 基于分层多链和多重审计共识的**隐私**保护设计
- 优化**RPC**通讯效率以及网络协议栈协议，以达到领先的交易性能

04

科学计算区块链应用思考

区块链应用思考

- 实验科学结果的数字存证保护
 - 数字存证保护科学论文版权
- 科学计算系统的安全解决方案
 - 分布式容错、共享：科学实验发现和结果的多科研实验团队共享
- 科学实验计算资源服务的权益保障和快捷支付结算
 - 基于不可篡改的交易历史记录，保证资源服务交易的真实性；
 - 基于区块链的代币服务实现快捷支付结算

Q&A Thank You !

Homepage : <http://www.yunphant.com>

Email : service@yunphant.com

Telephone : 18267147460

陈建海 : 13958011808

微信公众号 : 云象区块链

