IT大咖说
知识分享平台

智链
ChainNova

HyperLedger Fabric 项目核心模块剖析

于林生

IT大咖说
知识分享平台

公司简介

智链 是由美国硅谷PeerNova和中南建设合资成立的一家高科技公司，同时并购了Hyperledger成员、创新科技公司云图智链（梧桐树）。我们的定位是服务于金融科技产业实践，融合云计算、大数据以及人工智能等科技板块，共同为国内乃至全球用户实现行业应用场景落地，构建互联网新金融、区块链+产业、商业认知等解决方案。

http://www.chainnova.com

中南集团·中南建设
ZHONGNAN GROUP · ZHONGNAN CONSTRUCTION
建家 · 筑城 · 启未来

*中南建设成立于**1988**年，于**2009**年在深圳交易所上市，现有员工**50,000**余人，总资产逾千亿，**2016**年综合营业收入**850**亿，下设中南置地、中南建筑、中南土木、中南资本、中南金融、中南工业、中南建投等**7**大产业板块。*

PEERNOVA

PeerNova作为一家硅谷的技术公司，在分布式系统、区块链技术、网络解决方案、大数据和金融服务方面拥有专业经验，专注于将区块链和大数据结合，应用于银行等金融机构数据资产管理领域。

IT大咖说
知识分享平台

大纲

- Blockchain 区块链
- Hyperledger 超级账本
- Fabric
  - Fabric关键术语
  - Fabric结构
  - Fabric交易流程
  - Fabric开发
  - Fabric智能合约

IT大咖说
知识分享平台

Blockchain

“区块链”技术最初是由中本聪为比特币（一种数字货币）设计出的一种特殊的数据库技术，它基于密码学中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。

区块链技术是利用**块链式数据结构**来验证与存储数据、利用分布式节点**共识算法**来生成和更新数据、利用**密码学**的方式来保证数据传输和访问的安全、利用自动化脚本代码组成的**智能合约**来编程和操作数据的一种全新的分布式基础架构和计算范式。**本质上就是个分布式数据库。**

Blockchain

传统商业网络面临的挑战

1. 每个参与方都有自己的账本，交易发生时各自更改
2. 因此产生为了协同各参与方而带来额外的工作和成本
3. 业务条件-“合同”-重复分散在各参与方，造成整体业务流有效性低
4. 整个网络依赖于一个或几个中心系统，缺乏透明、存在风险

Blockchain

区块链架构带来以下改变

1. 每一个商业网络的参与方都具有一个共享的帐本，当交易发生时，通过点对点的复制更改所有账本
2. 使用密码算法确保网路上的参与者仅仅可以看到和他们相关的账本内容，交易是安全的、授权的和验证的
3. 将资产转移交易相关的合同条款嵌入交易数据库以做到满足商务条件下交易才发生
4. 网络参与者基于共识机制来保证交易是共同验证，商业网络满足政府监管、合规及审计

IT大咖说
知识分享平台

Blockchain

Centralized

Distributed

Decentralized

从中心化到去中心化

IT大咖说
知识分享平台

Blockchain

区块链特点

- 去中心化：不存在中心化的硬件或机构，所有节点共同维护
- 不可篡改：账本数据永久存储，单点修改无效，除非51%
- 匿名：节点间数据交换无需信任，无需公开身份
- 账本一致：所有节点账本数据相同
- 公开透明：除交易私有信息外，账本数据可公开查询
- 不可伪造：交易使用数字签名技术，不可伪造

IT大咖说
知识分享平台

Blockchain

- **公有链**　世界上所有人都可以阅读和发送交易，都能参与到共识形成过程——决定在链条上添加什么区块以及现状是怎样的。通常被认为是"完全去中心化"。

- **私有链**　书写许可对一个组织保持中心化。阅读许可可能是公开的或者限制在任意程度。应用很可能包含对单个公司内部的数据库管理，审查等，因此公共的可读性在很多情况下根本不必要，但在另一些情况下人们又想要公共可读性。

- **联盟链**　共识形成过程由预先选择的一系列的节点所掌控，例如，一个有15个金融机构的团体，每个机构都操作一个节点，为了使区块生效，其中的10个必须签署那个区块。阅读区块链的权利可能是公开的，或仅限于参与者。通常被认为是"部分去中心化"。

IT大咖说
知识分享平台

HYPERLEDGER

超级账本 是Linux基金会于2015年发起的推进区块链数字技术和交易验证的开源项目，目标是让成员共同合作，共建开放平台，满足来自多个不同行业各种用户案例，并简化业务流程。通过创建分布式账本的公开标准，实现虚拟和数字形式的价值交换，例如资产合约、能源交易、结婚证书、能够安全和高效低成本的进行追踪和交易。

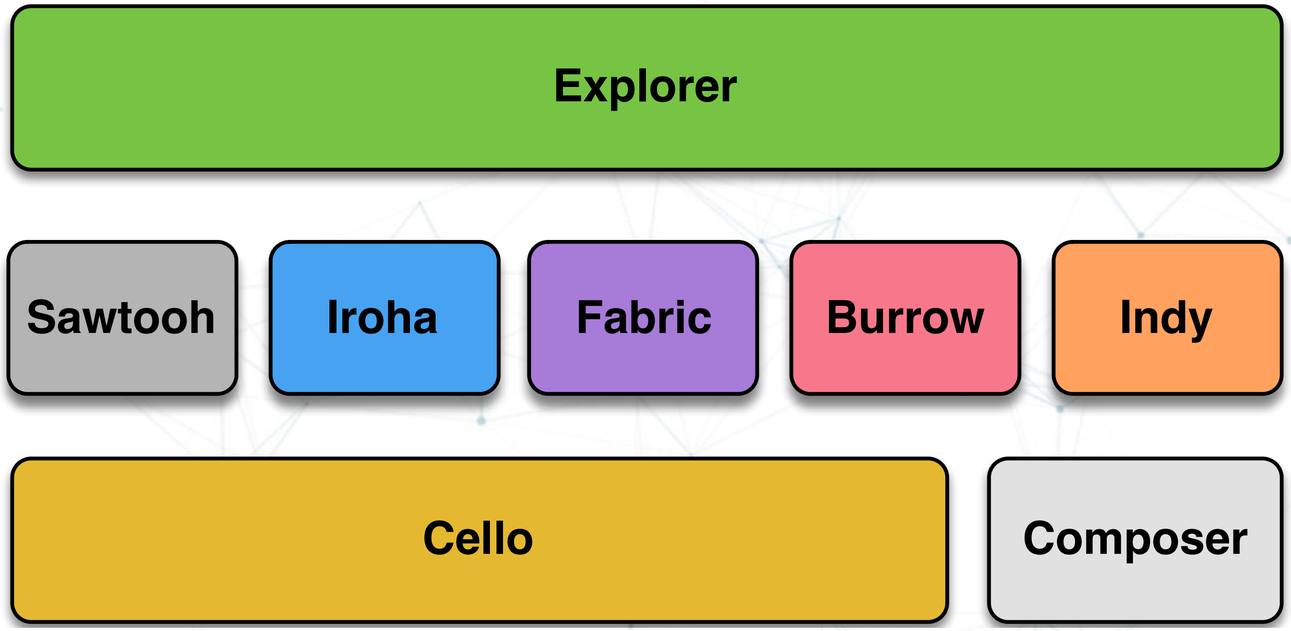
官网：https://www.hyperledger.org

GitHub：https://github.com/hyperledger

IT大咖说
知识分享平台

HYPERLEDGER

Hyperledger孵化和推广了一系列商业区块链技术，包括分布式分类账本框架、智能合约引擎、客户端库、图形化界面、工具库和示例应用程序。



IT大咖说 知识分享平台

HYPERLEDGER

- **Sawtooth** 高度模块化的分布式账本平台，提供了新的共识算法—PoET
- **Iroha** 轻量级分布式账本框架，侧重于移动
- **Fabric** 提供一个模块化架构开发区块链应用
- **Burrow** 基于以太坊EVM的智能合约执行引擎
- **Indy** 实现工具、库及可复用组件，提供基于区块链或其他分布式账本的数字身份，以便可以跨管理域、应用程序或其他仓库进行互操作
- **Cello** 部署BaaS区块链生态系统，协助创建、管理、终止区块链
- **Composer** 构建区块链商业网络的协作工具，加速智能合约的开发及其在分布式账本中的部署
- **Explorer** web工具，可以预览、调用、部署和查询区块、交易及相关数据、网络信息、智能合约（chaincode）以及存储在账本中的任何相关信息

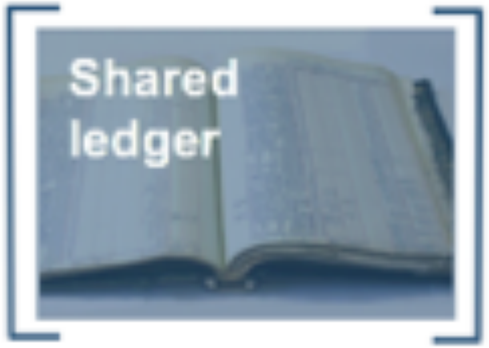
IT大咖说 知识分享平台

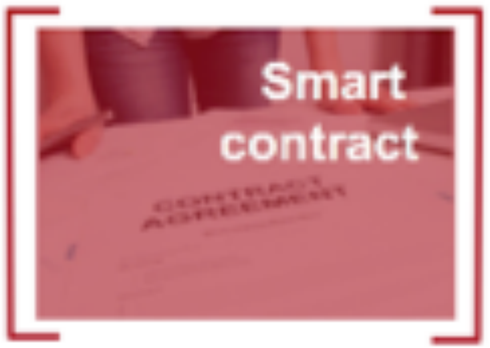
Fabric

What

Requirements of blockchain for business

Append-only distributed system of record shared across business network





Business terms embedded in transaction database & executed with transactions

Ensuring appropriate visibility; transactions are secure, authenticated & verifiable

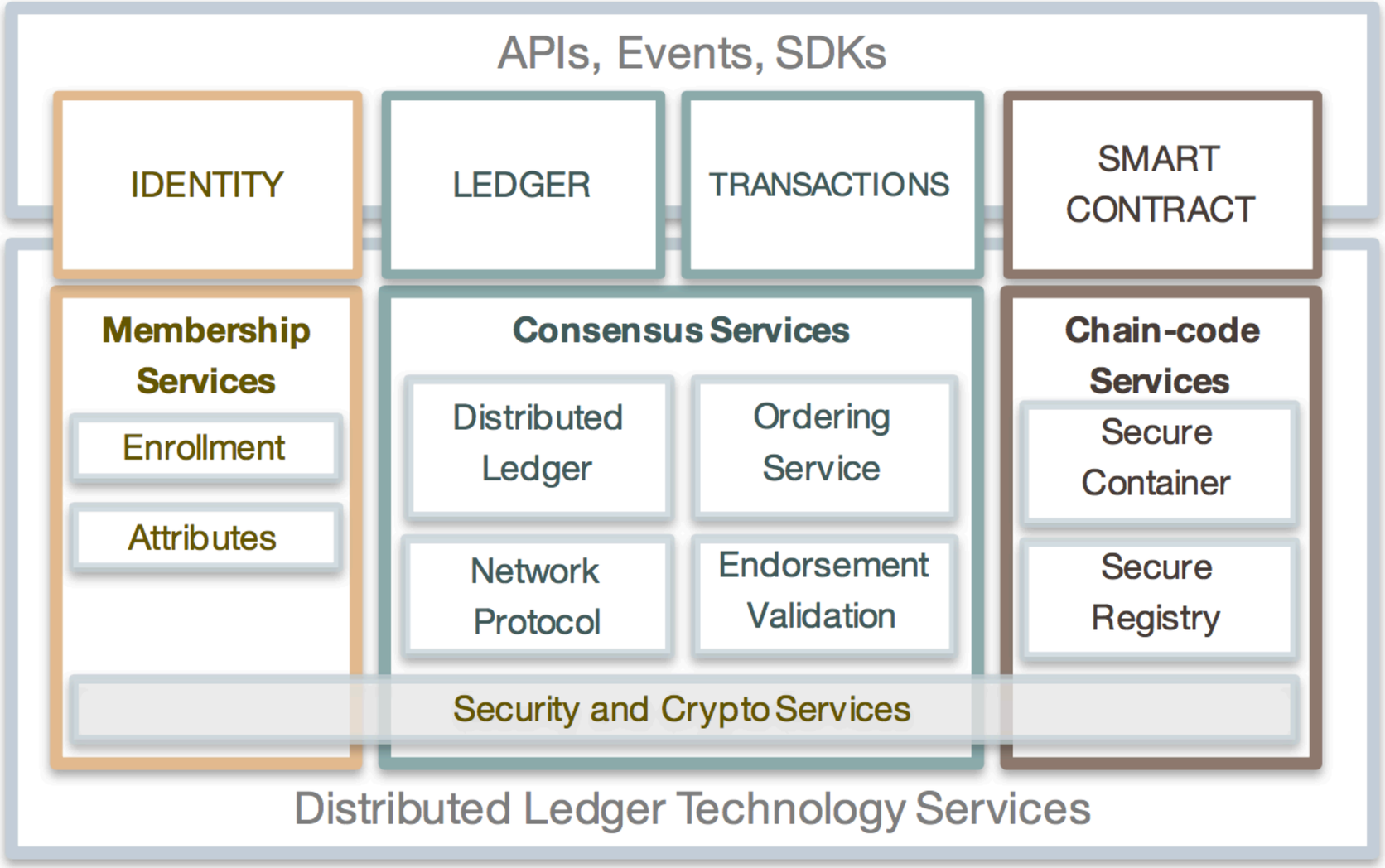
Transactions are endorsed by relevant participants

IT大咖说 知识分享平台

Fabric术语

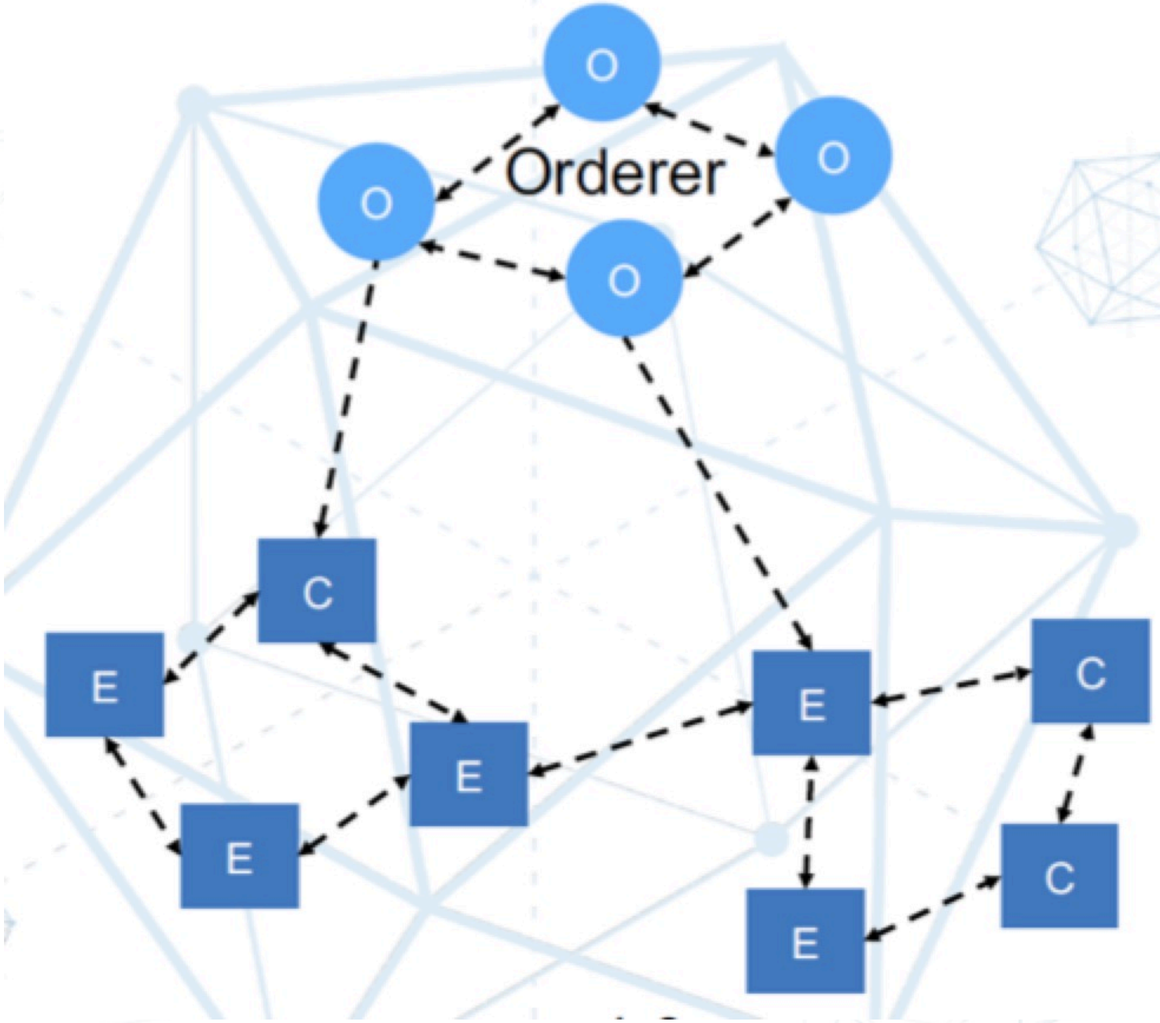
- Chaincode　链码，即智能合约
- Transaction　交易，对智能合约中函数的调用，修改ledger与state
- WorldState　数据库中存储的k-v，即交易的实际操作数据
- Peer　维护账本的网络实体，分为背书节点和记账节点
- Endorsement　背书，节点对交易模拟执行结果签名
- Commitment　提交，节点对区块校验并写入账本

Fabric逻辑结构

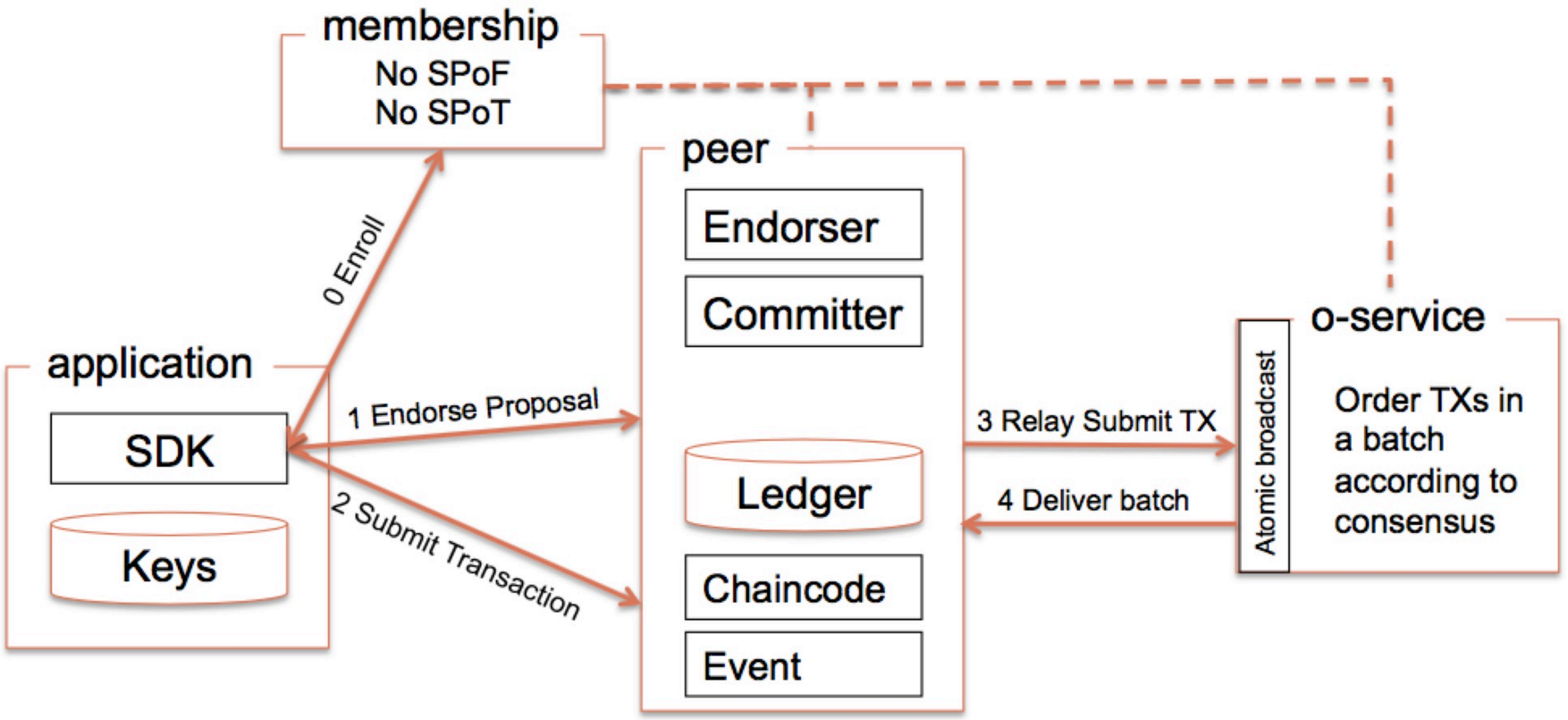


IT大咖说
知识分享平台

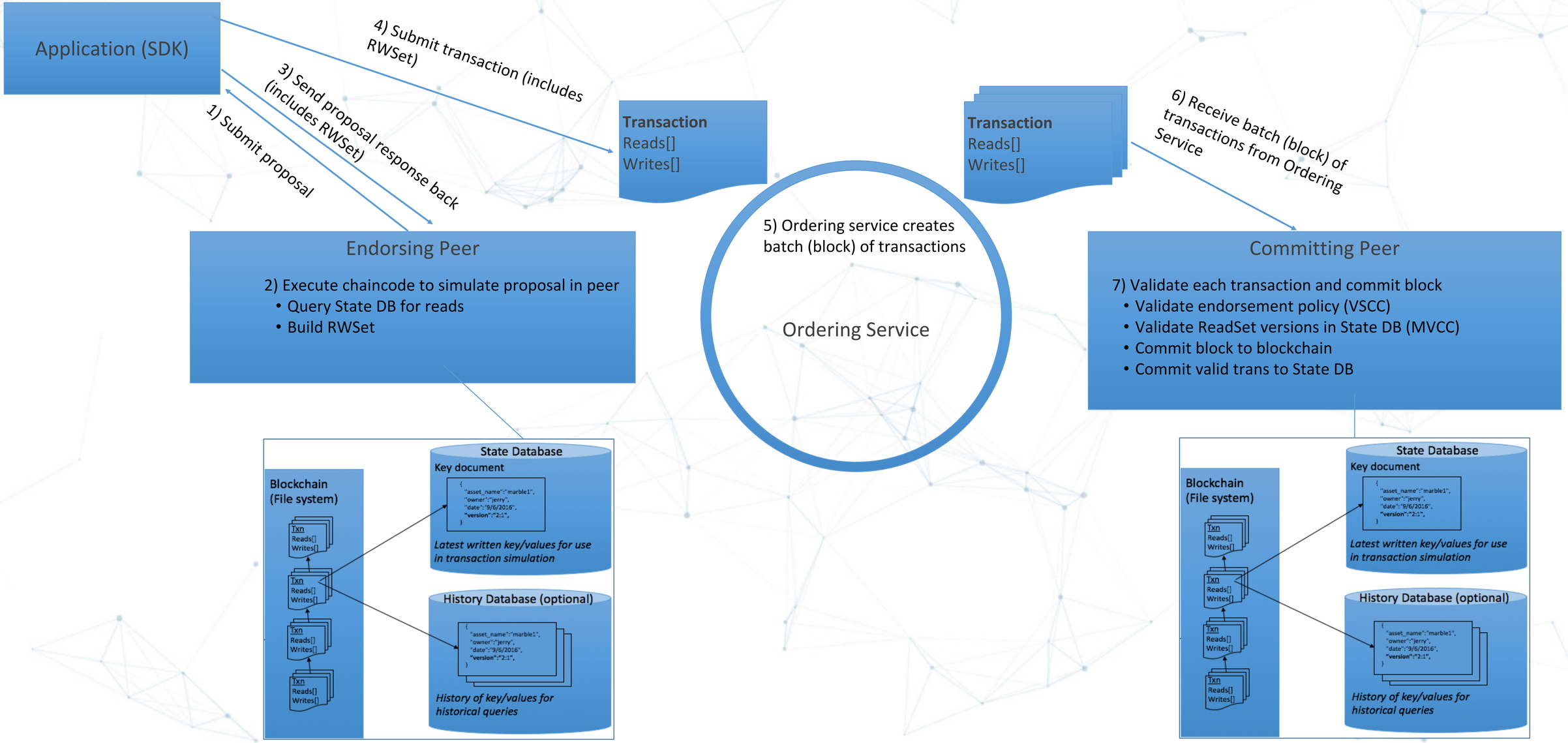
Fabric网络结构



Fabric运行结构


membership
No SPoF
No SPoT
peer
Endorser
Committer
Ledger
Chaincode
Event
application
SDK
Keys
0 Enroll
1 Endorse Proposal
2 Submit Transaction
3 Relay Submit TX
4 Deliver batch
o-service
Atomic broadcast
Order TXs in a batch according to consensus

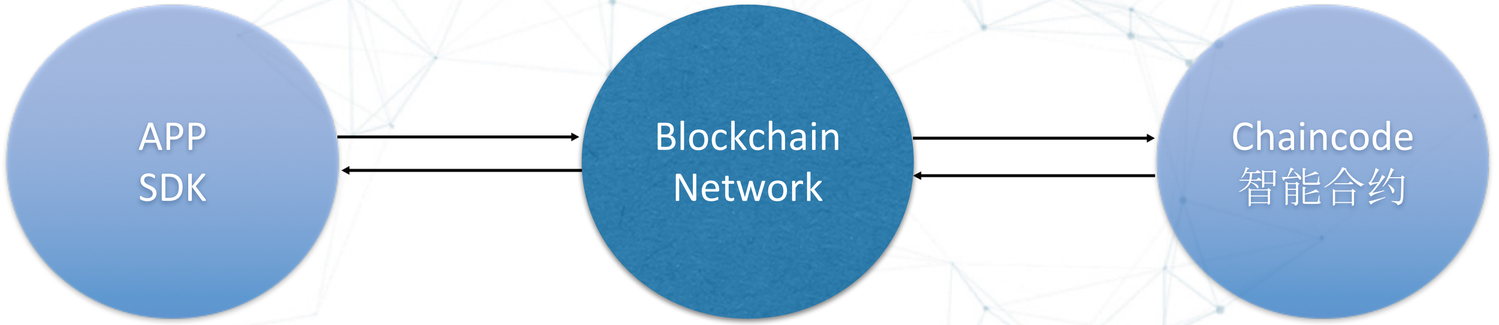
Fabric交易流程


Application (SDK)
4) Submit transaction (includes RWSet)
3) Send proposal response back (includes RWSet)
1) Submit proposal
Transaction
Reads[]
Writes[]
Transaction
Reads[]
Writes[]
6) Receive batch (block) of transactions from Ordering Service
Endorsing Peer
2) Execute chaincode to simulate proposal in peer
• Query State DB for reads
• Build RWSet
5) Ordering service creates batch (block) of transactions
Ordering Service
Committing Peer
7) Validate each transaction and commit block
• Validate endorsement policy (VSCC)
• Validate ReadSet versions in State DB (MVCC)
• Commit block to blockchain
• Commit valid trans to State DB
Blockchain (File system)
Txn
Reads[]
Writes[]
State Database
Key document
Latest written key/values for use in transaction simulation
History Database (optional)
History of key/values for historical queries

IT大咖说 知识分享平台

Fabric开发

- APP　　根据业务需求编写与区块链网络交互的应用程序，发起交易查询账本。目前提供了Java、Go、Node、Python的SDK
- Chaincode　根据不同业务逻辑编写相应智能合约
- 插件　　实现可插拔插件（interface），如共识算法、数据存储、加密算法、CA等



IT大咖说 知识分享平台

Fabric智能合约

Chaincode 是一个实现了特定接口的可执行程序，通常运行在docker container中。Chaincode通过处理app提交的交易来管理ledger state，是执行交易的唯一途径。

Fabric v1.0-beta后仅支持Go，之前的版本可支持Java

```
type Chaincode interface {
    // 初始化，一般只执行一次
    Init(stub ChaincodeStubInterface) pb.Response

    // 查询或更新world state
    Invoke(stub ChaincodeStubInterface) pb.Response
}
```

IT大咖说 知识分享平台

Fabric智能合约

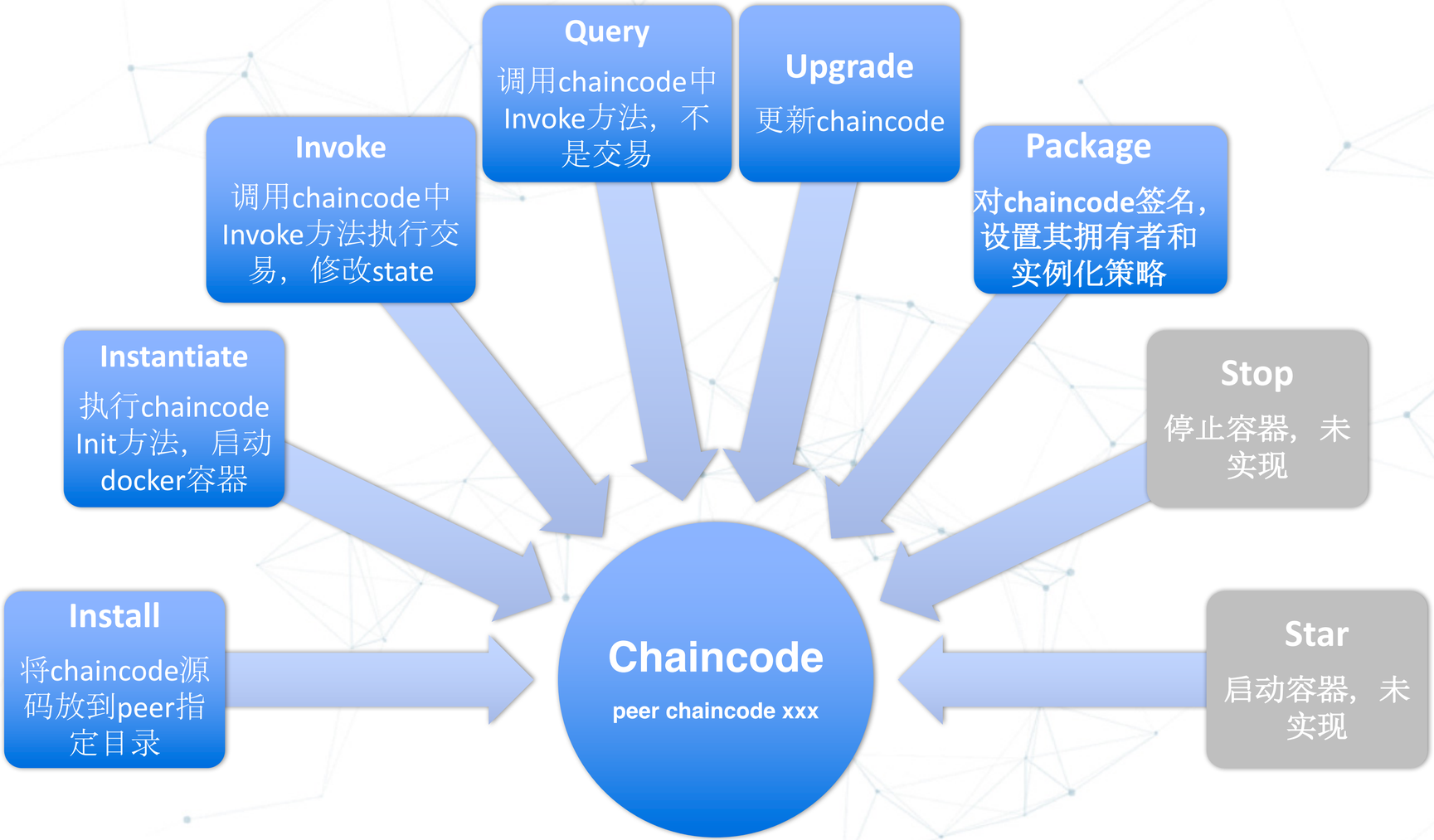
```
package main

import (
    "fmt"

    "github.com/hyperledger/fabric/core/chaincode/shim"
    pb "github.com/hyperledger/fabric/protos/peer"
)

type SimpleChaincode struct {
}

func (s *SimpleChaincode) Init(stub shim.ChaincodeStubInterface) pb.Response {
    return shim.Success(nil)
}

func (s *SimpleChaincode) Invoke(stub shim.ChaincodeStubInterface) pb.Response {
    return shim.Success(nil)
}

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

IT大咖说 知识分享平台

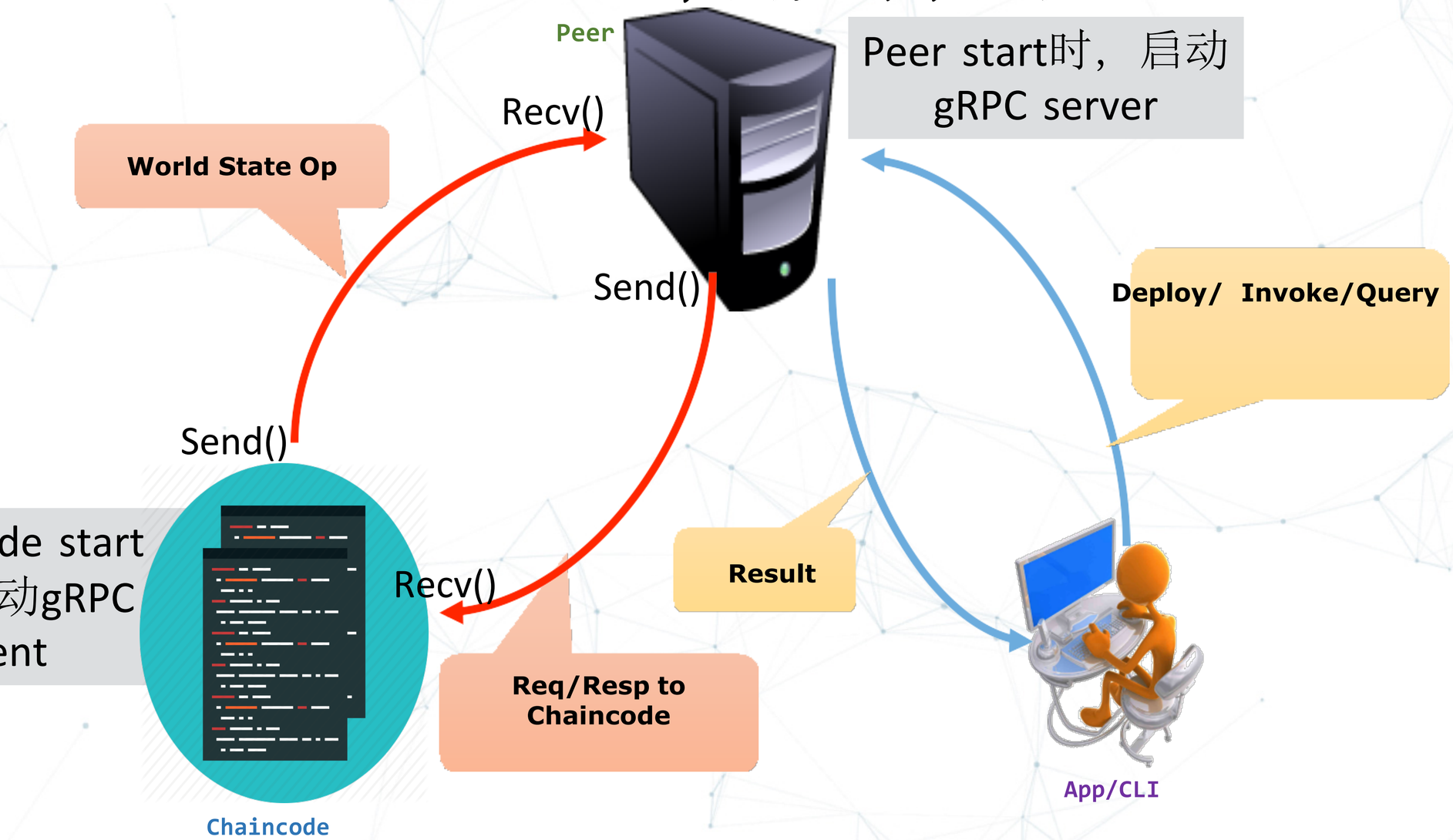
Fabric智能合约

```
type ChaincodeStubInterface interface {
    GetArgs() [][]byte
    GetStringArgs() []string
    GetFunctionAndParameters() (string, []string)
    GetArgsSlice() ([]byte, error)
    GetTxID() string
    InvokeChaincode(chaincodeName string, args [][]byte, channel string) pb.Response
    GetState(key string) ([]byte, error)
    PutState(key string, value []byte) error
    DelState(key string) error
    GetStateByRange(startKey, endKey string) (StateQueryIteratorInterface, error)
    GetStateByPartialCompositeKey(objectType string, keys []string) (StateQueryIteratorInterface, error)
    CreateCompositeKey(objectType string, attributes []string) (string, error)
    SplitCompositeKey(compositeKey string) (string, []string, error)
    GetQueryResult(query string) (StateQueryIteratorInterface, error)
    GetHistoryForKey(key string) (HistoryQueryIteratorInterface, error)
    GetCreator() ([]byte, error)
    GetTransient() (map[string][]byte, error)
    GetBinding() ([]byte, error)
    GetSignedProposal() (*pb.SignedProposal, error)
    GetTxTimestamp() (*timestamp.Timestamp, error)
    SetEvent(name string, payload []byte) error
}
```

Fabric智能合约


Query
调用chaincode中Invoke方法，不是交易
Upgrade
更新chaincode
Invoke
调用chaincode中Invoke方法执行交易，修改state
Package
对chaincode签名，设置其拥有者和实例化策略
Instantiate
执行chaincode Init方法，启动docker容器
Stop
停止容器，未实现
Install
将chaincode源码放到peer指定目录
Chaincode
peer chaincode xxx
Star
启动容器，未实现

Fabric智能合约




```
service ChaincodeSupport {
      rpc Register(stream ChaincodeMessage) returns (stream ChaincodeMessage) {}
}
```

IT大咖说 知识分享平台

Fabric智能合约

Getstate

Send:

gRPC Send

Recv:

```
for {
    select {
    case err := <-errc:
        if err == nil {
            continue
        }
        return …
    case outmsg, val := <-c:
        if !val {
            return …
        }
        return …
    }
}
```

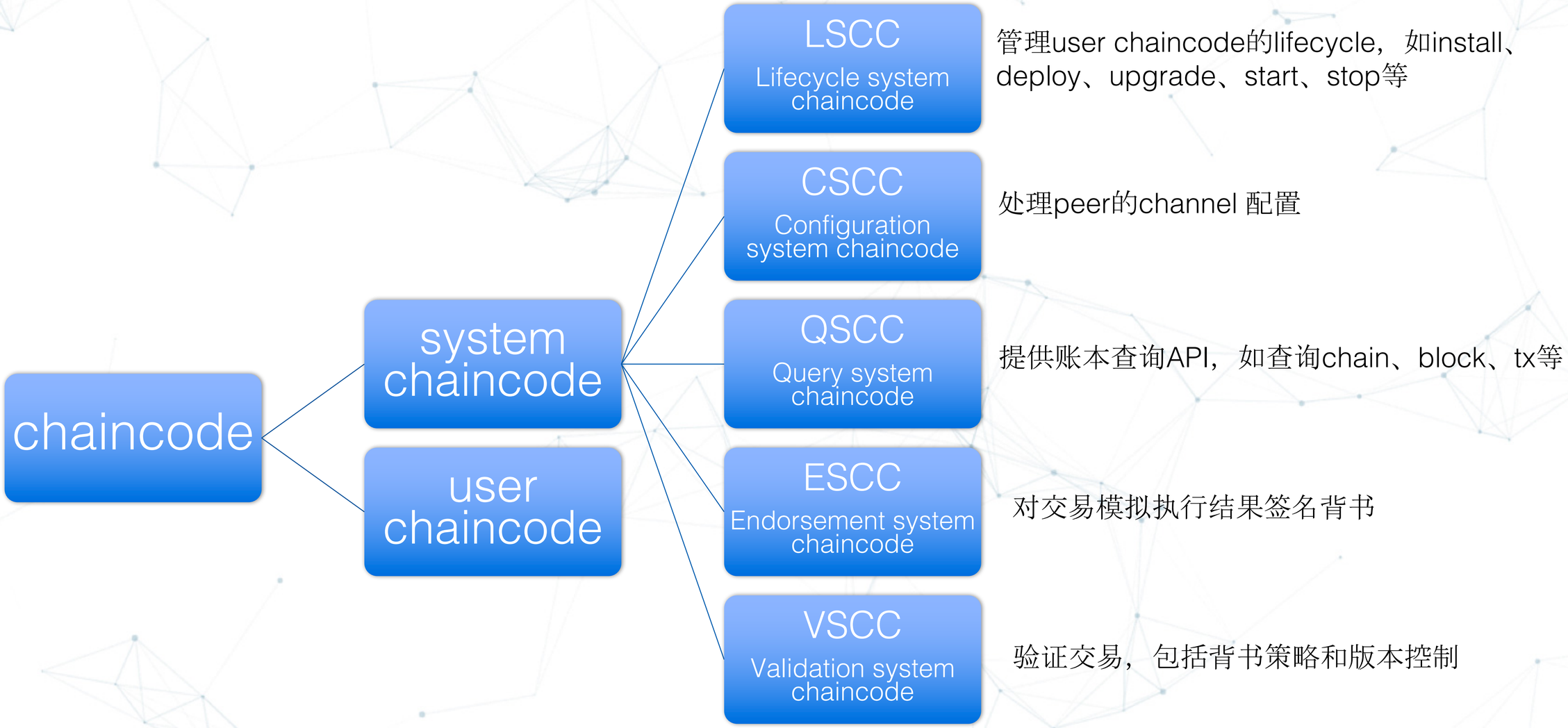
```
gRPC Send:
    go func() {
        err := handler.ChatStream.Send(msg)
        if errc != nil {
            errc <- err
        }
    }()
```

```
gRPC Recv:
    go func() {
        for {
            go func() {
                in2, err = stream.Recv()
                msgAvail <- in2
            }()
            select {
            case in = <-msgAvail:
                c <- in
            }
        }
    }()
```

peer

IT大咖说 知识分享平台

Fabric智能合约



IT大咖说
知识分享平台

谢谢！！