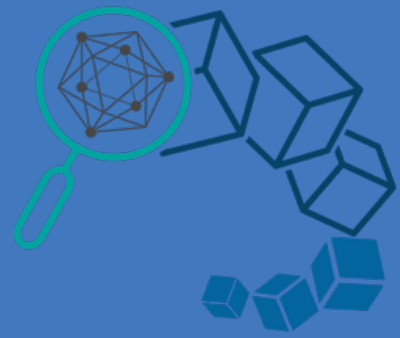
HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS



Hyperledger Workshop

Projects, Community and Fabric Architecture

Baohua Yang

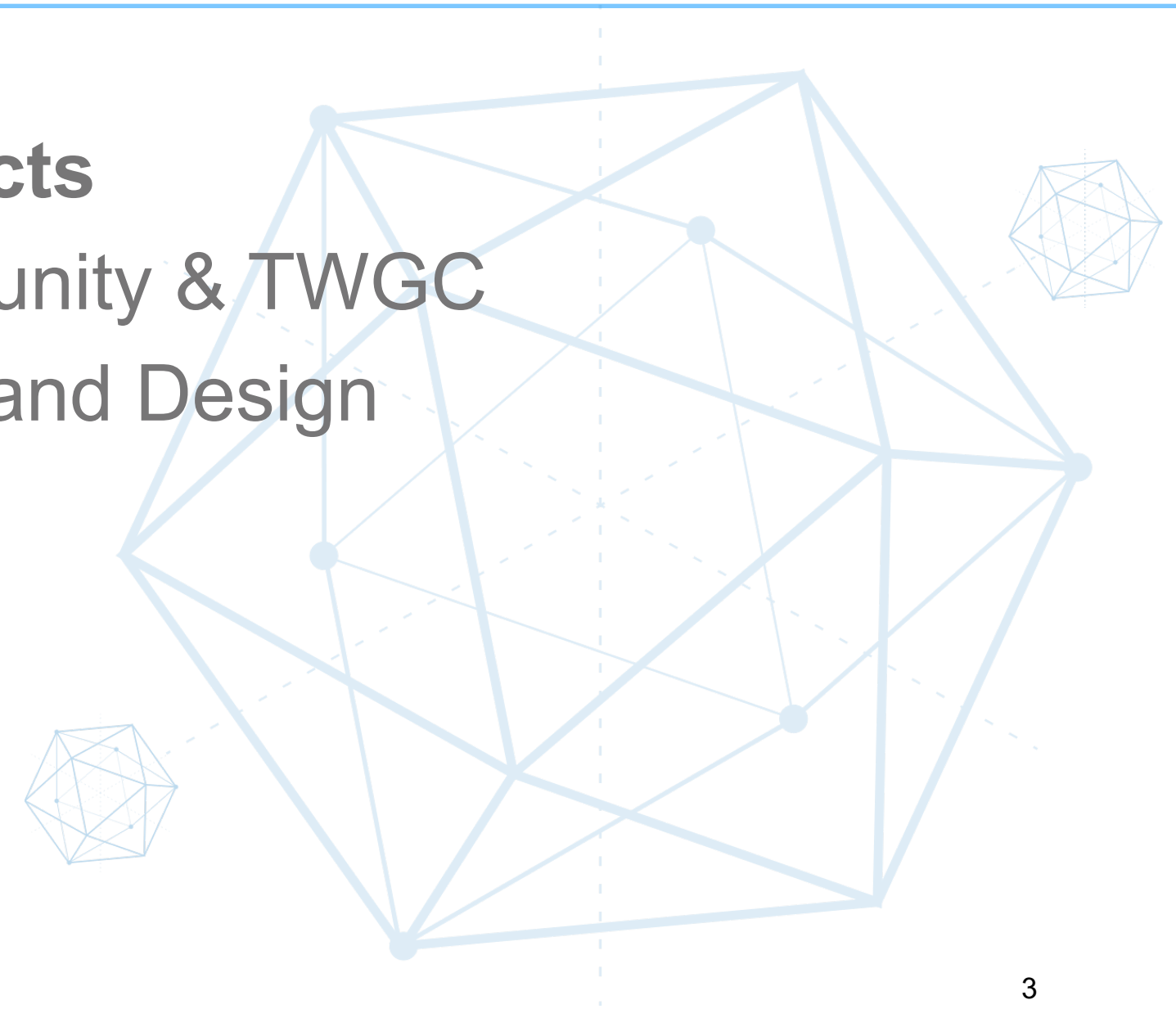
June 18, 2017

About Me

- **Senior Researcher in IBM**
  - Fintech, Cloud and Analytics
- **Open-Source Contributor**
  - Hyperledger, OpenStack, OpenDaylight, etc.
- **Hyperledger Developer**
  - Core designer & committer of Fabric, Cello, sdk etc.
  - PTL of Cello project and fabric-sdk-py project
  - Chair of Hyperledger Technical Working Group China

Outline

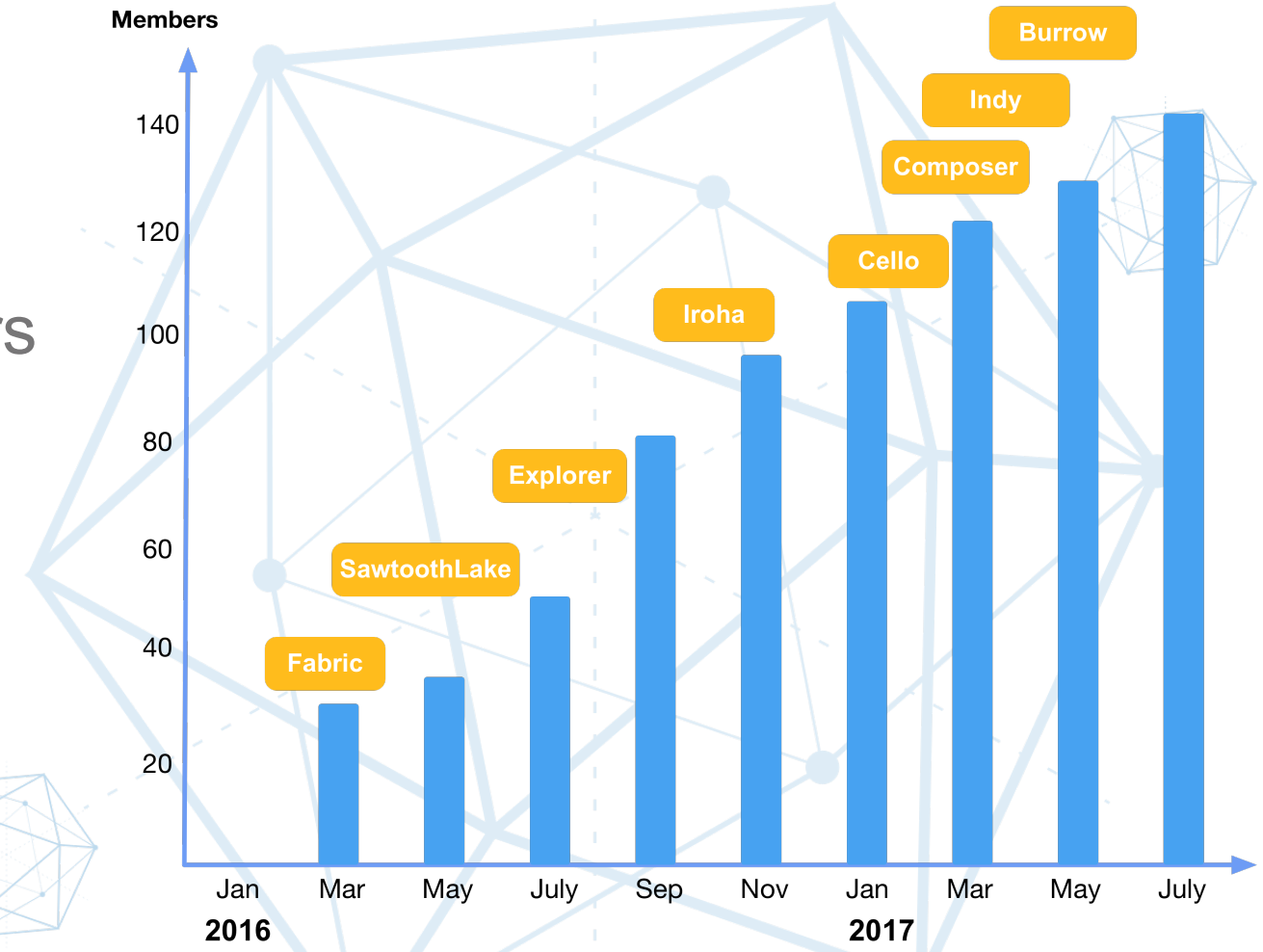
- **Hyperledger Projects**
- Hyperledger Community & TWGC
- Fabric Architecture and Design
- Q&A



Hyperledger Projects

- Since Dec 17, 2015
- Apache v2 License
- 30 Founding Members
- 18/36/142 Premier/China/Members
- 8 Top Projects
- 200+ Contributors
- 10000+ Commits

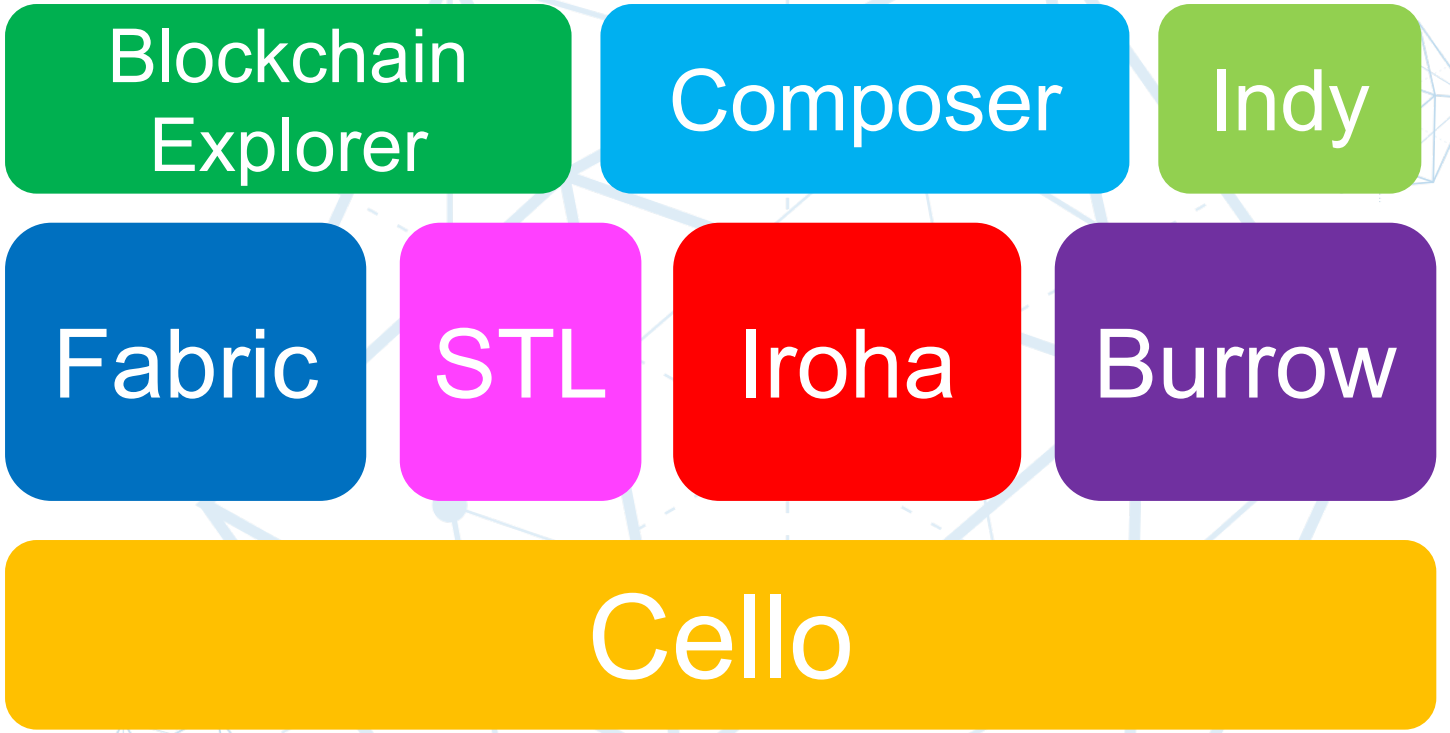
Enterprise grade, open source distributed ledger framework!



accenture AIRBUS AMERICAN EXPRESS CHANGE HEALTHCARE CME Group DEUTSCHE BÖRSE GROUP DAIMLER Digital Asset DTCC FUJITSU HITACHI Inspire the Next IBM intel J.P.Morgan NEC r3 SAP 万达·飞凡科技

Hyperledger Projects

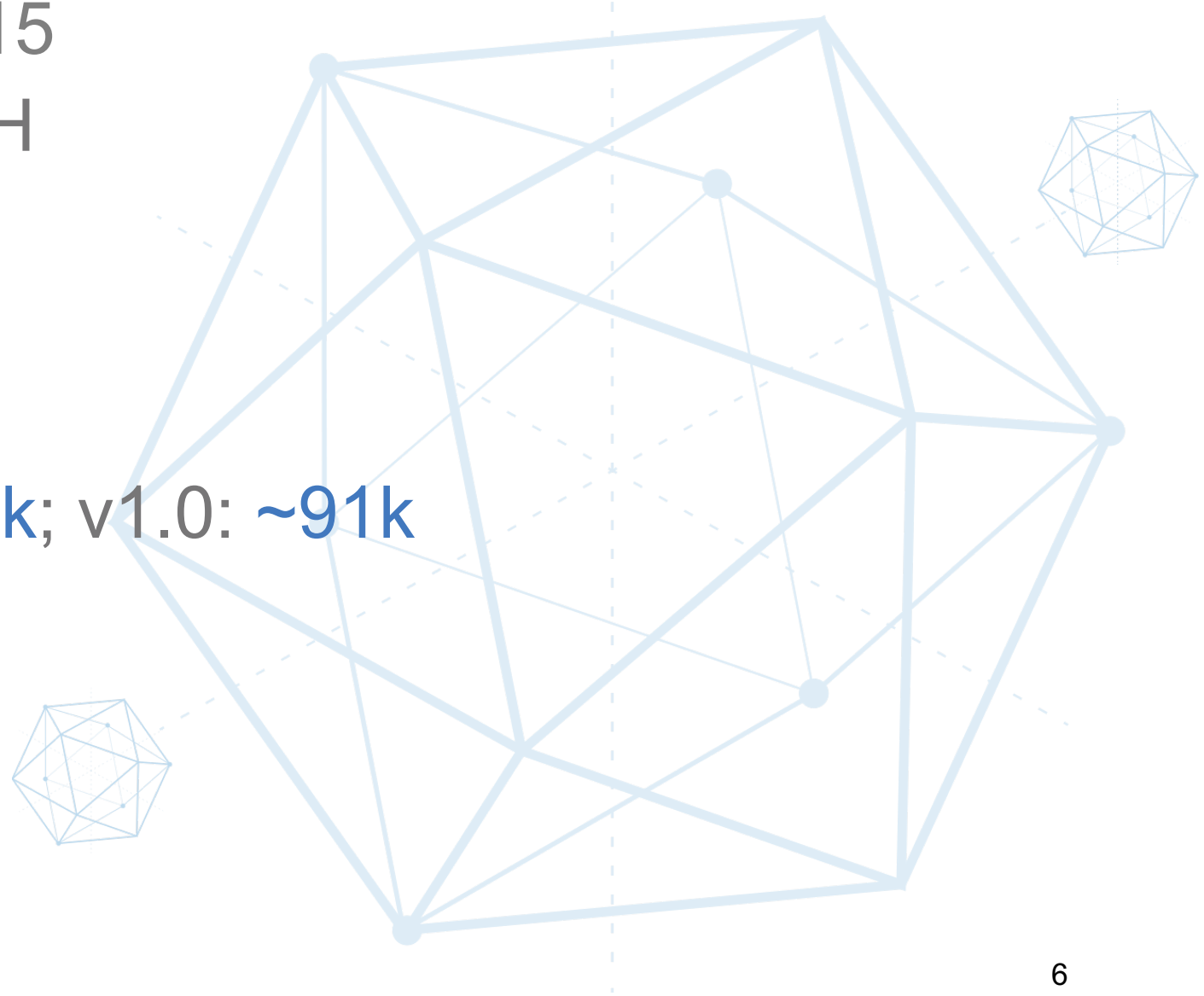
- 8 Top Projects
  - Fabric
  - SawtoothLake
  - Iroha
  - Cello
  - Blockchain Explorer
  - Composer
  - Burrow
  - Indy



Hyperledger Fabric

- Open-sourced at Dec, 2015
- Proposed by IBM and DAH
- Written in Golang
- 90+ contributors
- 5000+ commits
- Core code (loc): v0.6: ~49k; v1.0: ~91k

- Active now, in 1.0 rc1

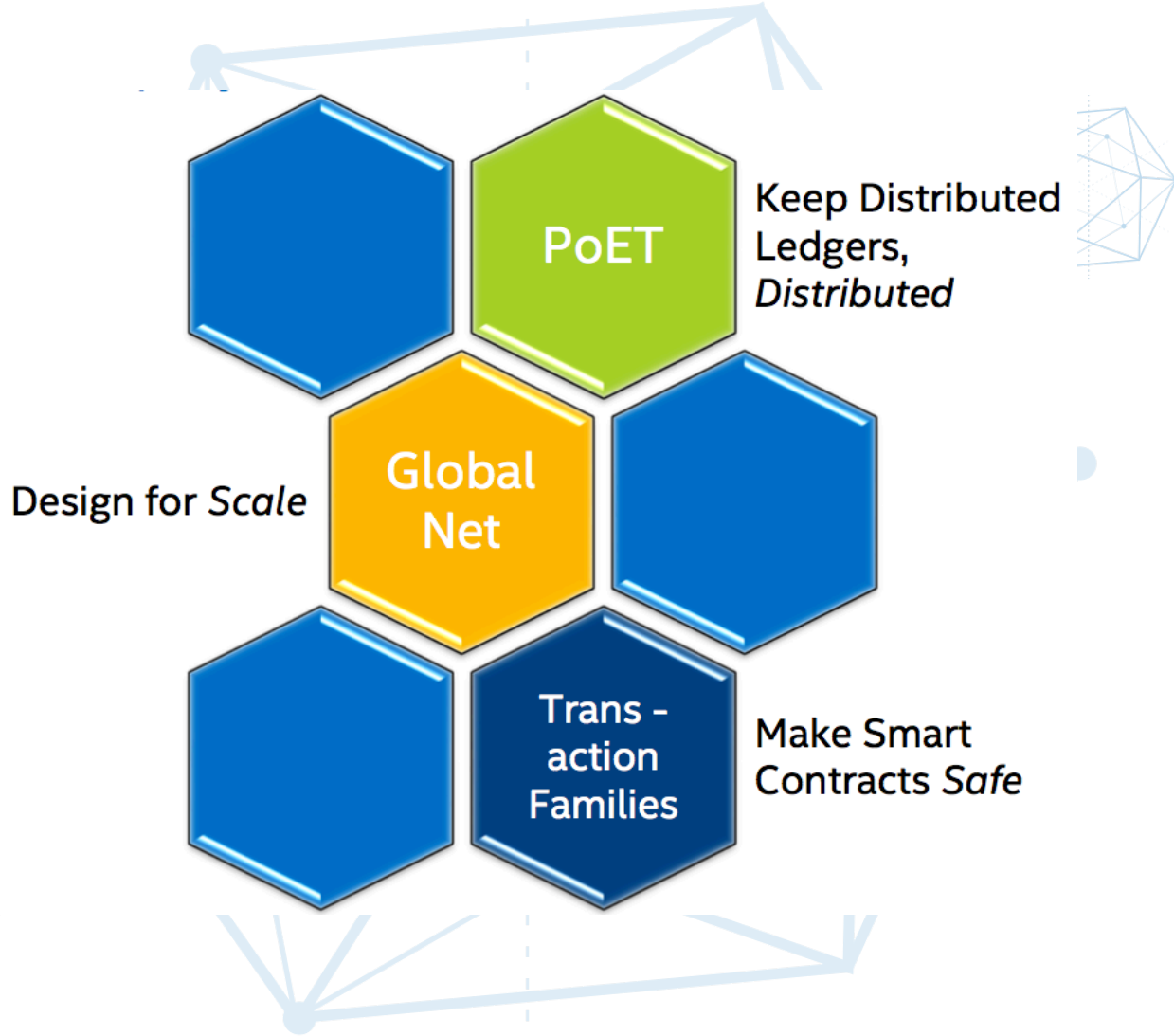


Hyperledger Fabric SDK

- Node.Js
  - https://github.com/hyperledger/fabric-sdk-node
- Python
  - https://github.com/hyperledger/fabric-sdk-py
- Java
  - https://github.com/hyperledger/fabric-sdk-java

Hyperledger SawtoothLake

- Open-sourced at April, 2016
- Proposed by Intel
- Python
- 40+ contributors
- 4000+ commits
- Key features
  - PoET consensus
  - Transaction Families
  - Scalability

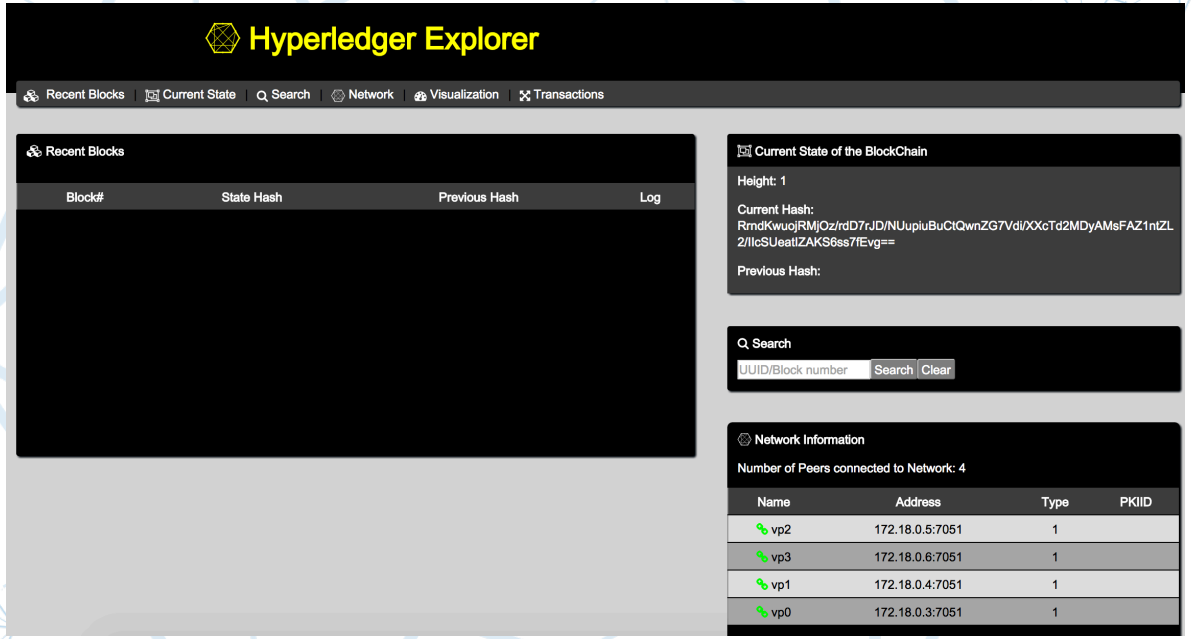


Hyperledger Iroha

- Open-sourced at Oct, 2016
- Proposed by Soramitsu
- C++
- 20+ contributors
- 2000+ commits
- Key features
  - C++ environment
  - Mobile and Web application Support
  - Sumeragi consensus

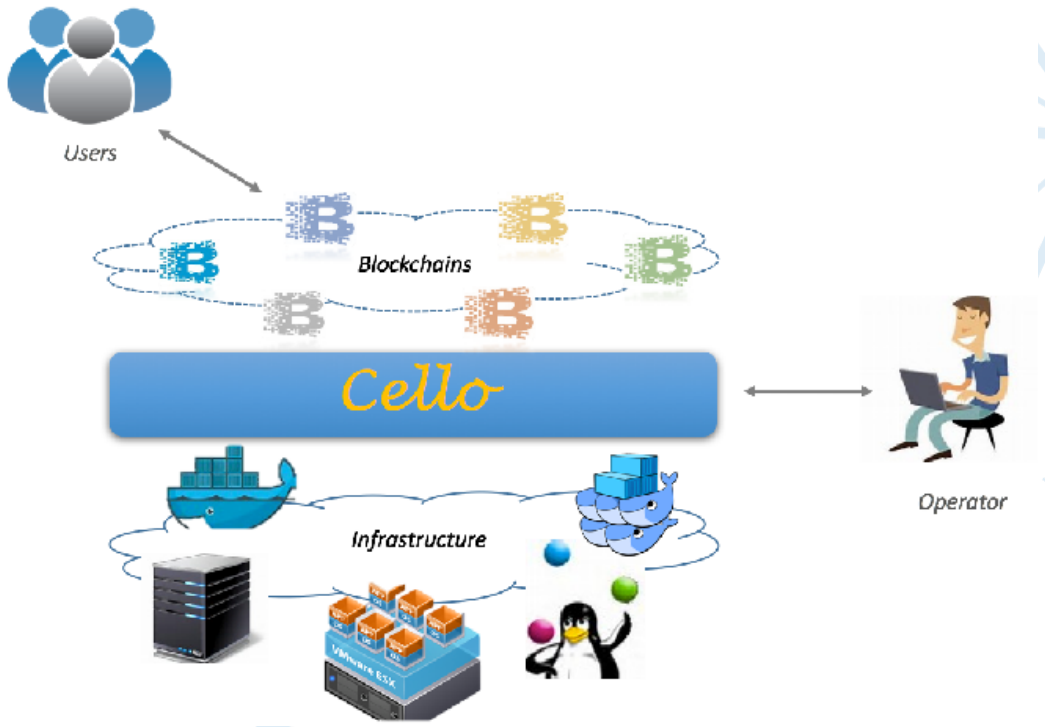
Hyperledger Blockchain Explorer

- Open-sourced at Aug, 2016
- Proposed by Intel, DTCC, IBM
- UI to interact with ledger
- Node.js
- Under-development
- Key features
  - Web UI to explorer a blockchain
  - Single-Page Application



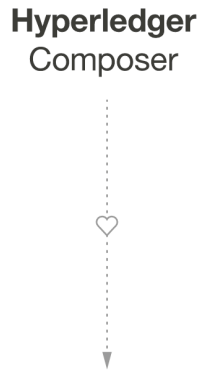
Hyperledger Cello

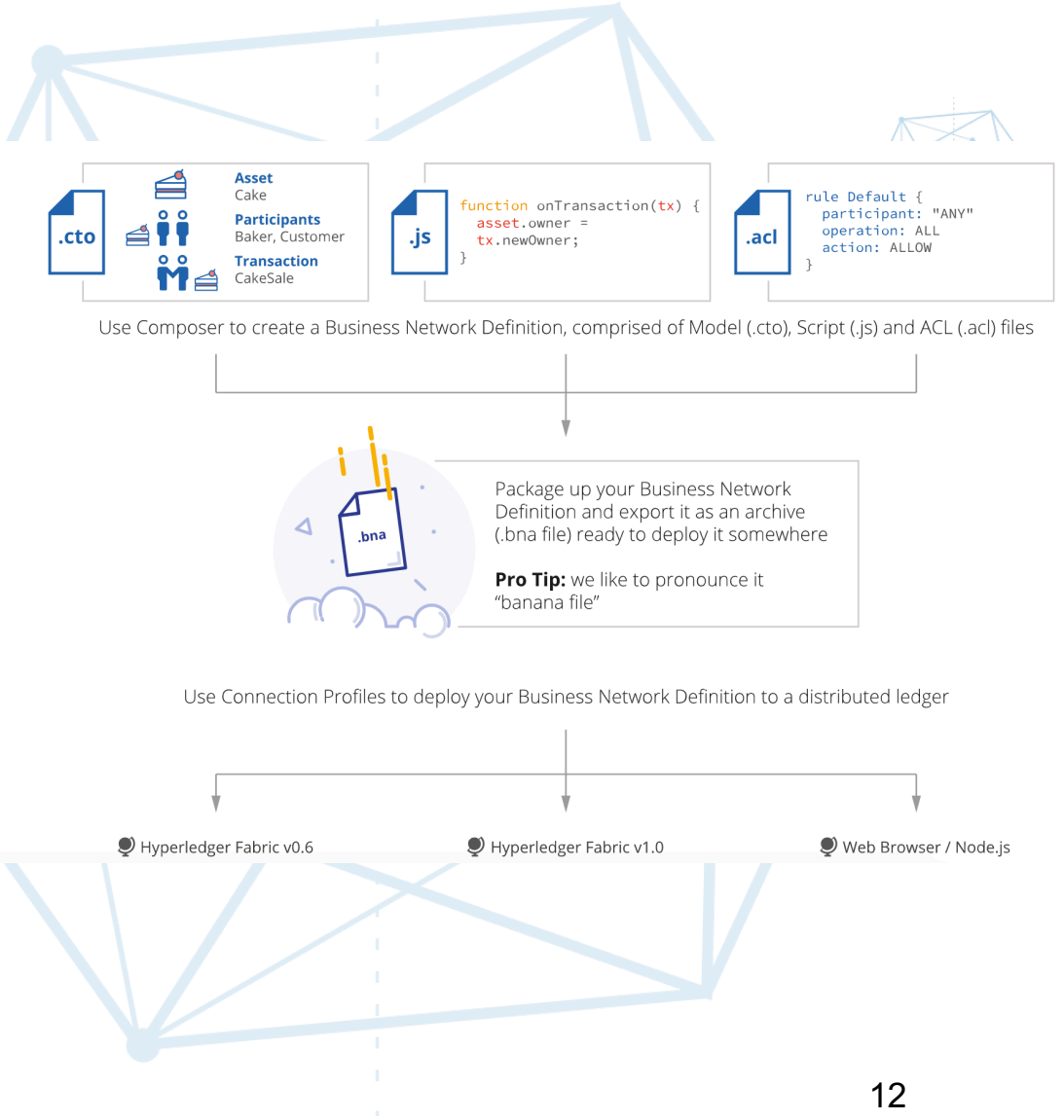
- Open-sourced at Jan, 2017
- Proposed by IBM
- Python, JavaScript
- 400+ commits
- Key features
  - Blockchain as a Service
  - Support various infrastructures
  - High-performance
  - Scalability
  - Pluggability



Hyperledger Composer

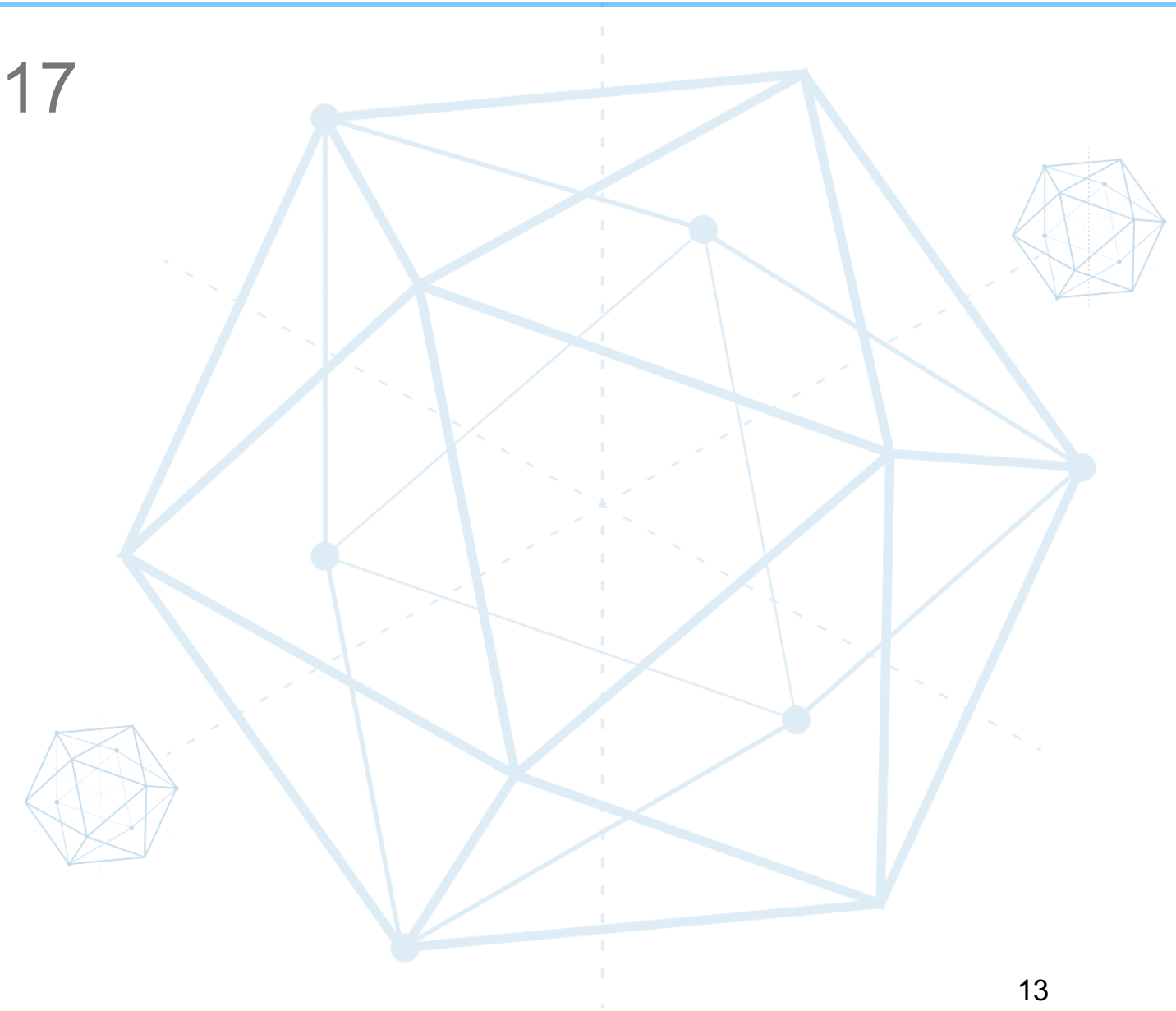
- Open-sourced at Mar, 2017
- Proposed by IBM
- JavaScript
- 4000+ commits
- Key features
  - For fabric
  - Simplify application creation





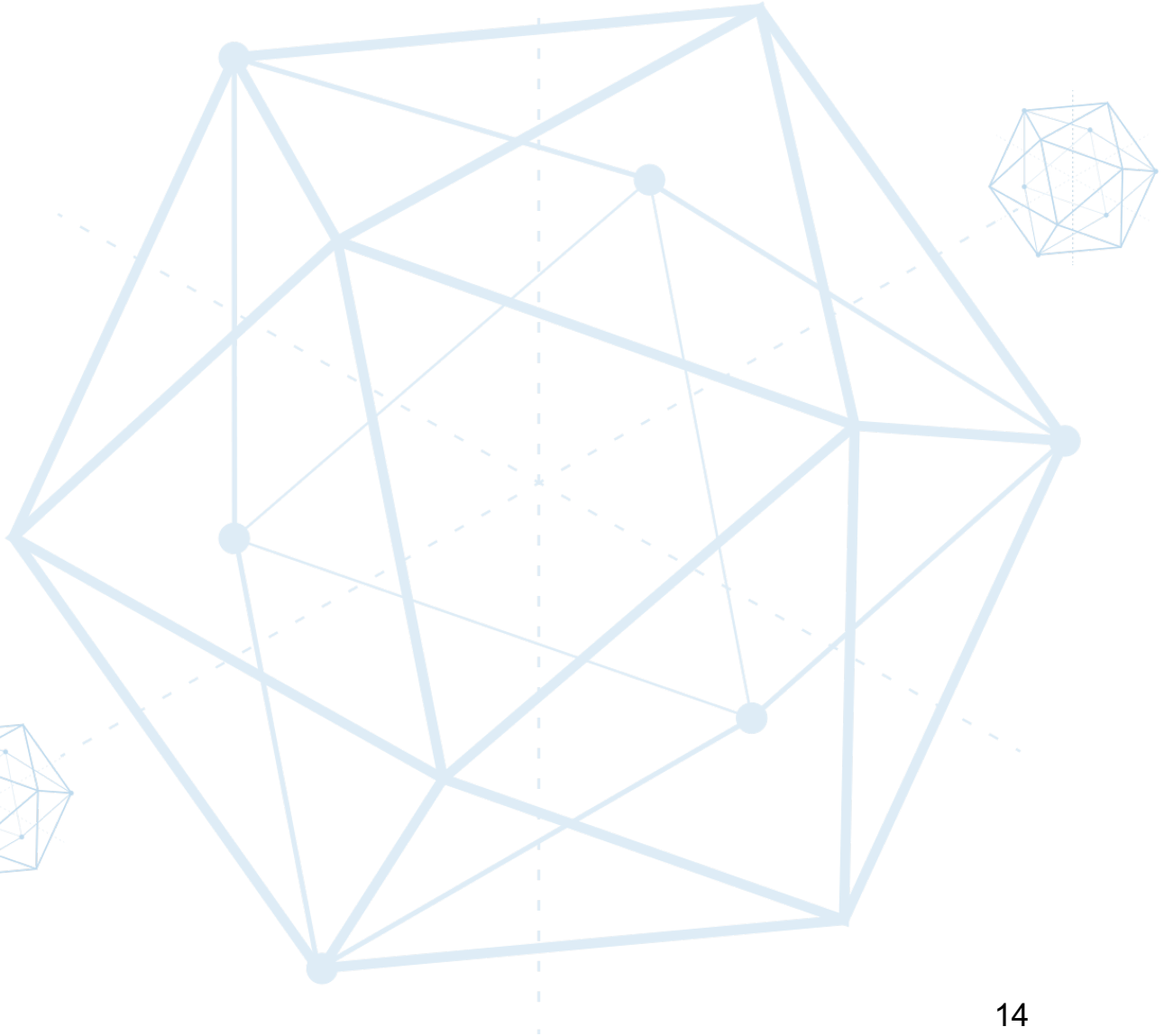
Hyperledger Burrow

- Open-sourced at April, 2017
- Proposed by Monax, Intel
- Written in Golang
- 1000+ commits
- Key features
  - From eris-db
  - Ethereum vm support
  - Proof-of-Stake

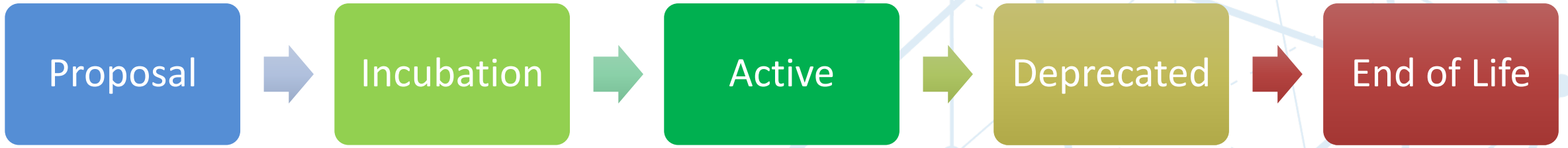


Hyperledger Indy

- Open-sourced at April, 2017
- Proposed by Sovrin
- Written in Python
- 1000+ commits
- Key features
  - Digital ID management
  - Cross-chain, cross-application

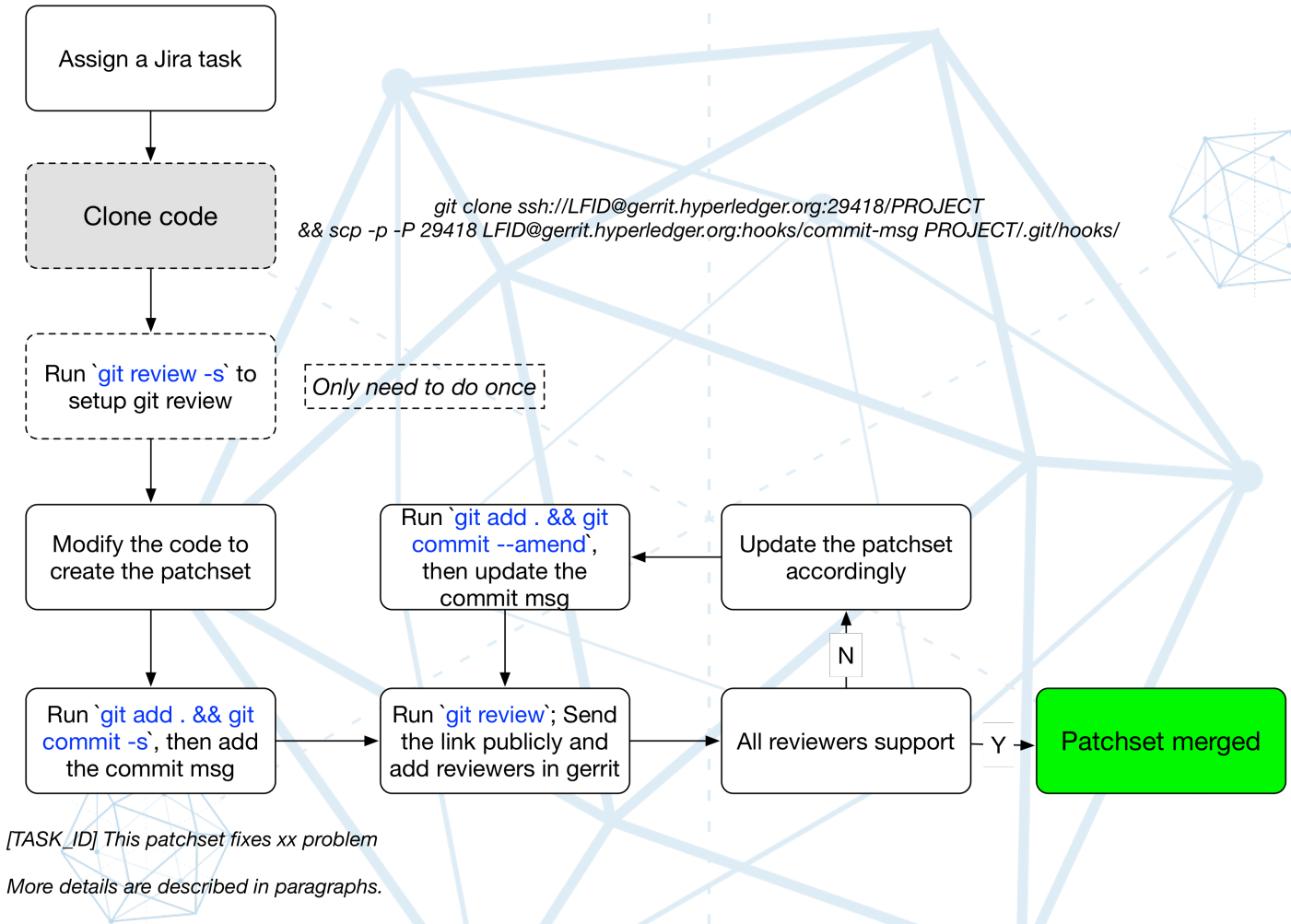


Hyperledger Projects Lifecycle



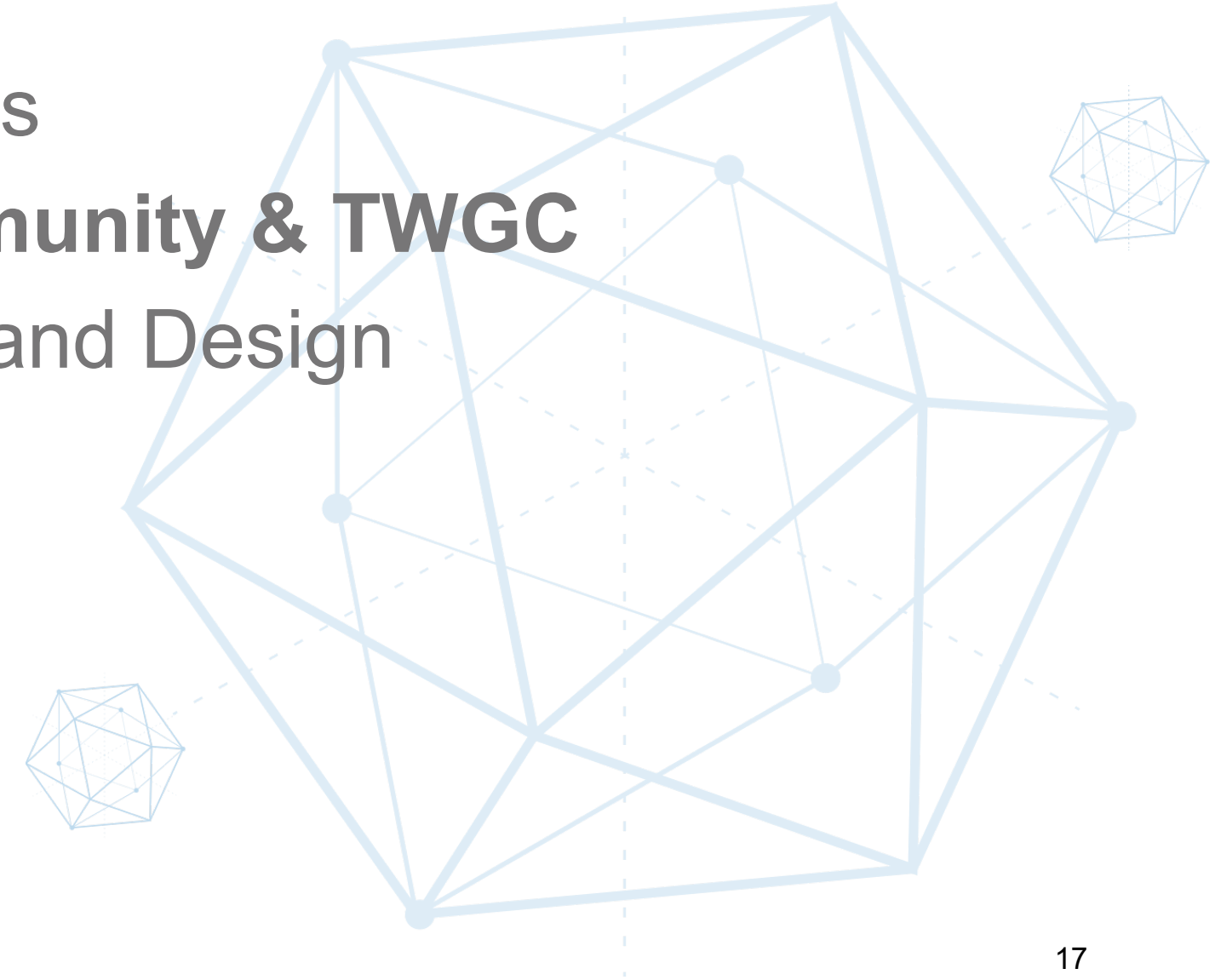
How to Contribute

- Linux Foundation ID
- Jira to manage tasks
- Gerrit to host code
- RocketChat



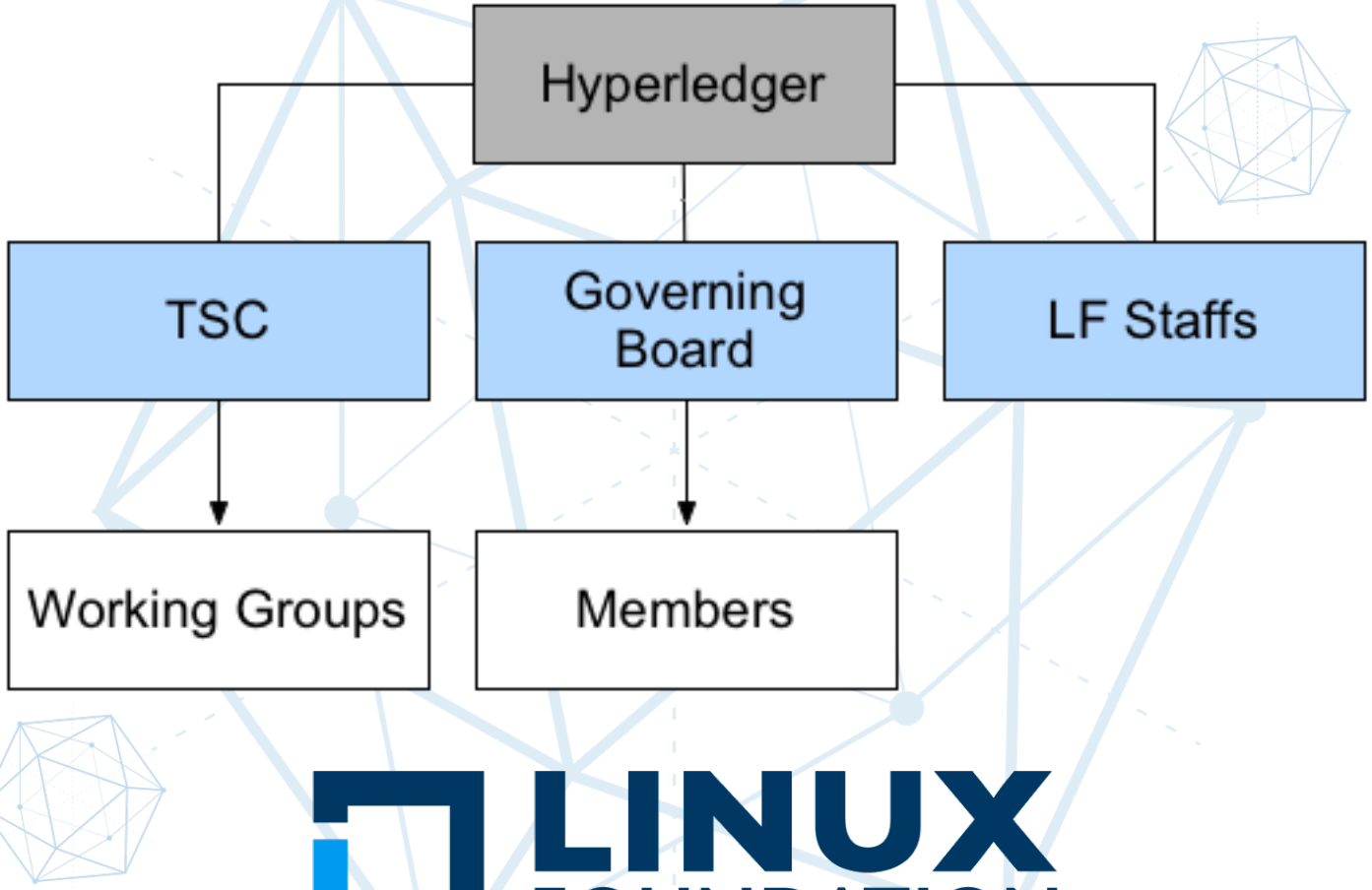
Outline

- Hyperledger Projects
- **Hyperledger Community & TWGC**
- Fabric Architecture and Design
- Q&A



Hyperledger Community

- Linux Foundation Support
- Organizations
  - Technical Steering Committee
  - Governing Board
  - Linux Foundation Staffs
- TWG-China



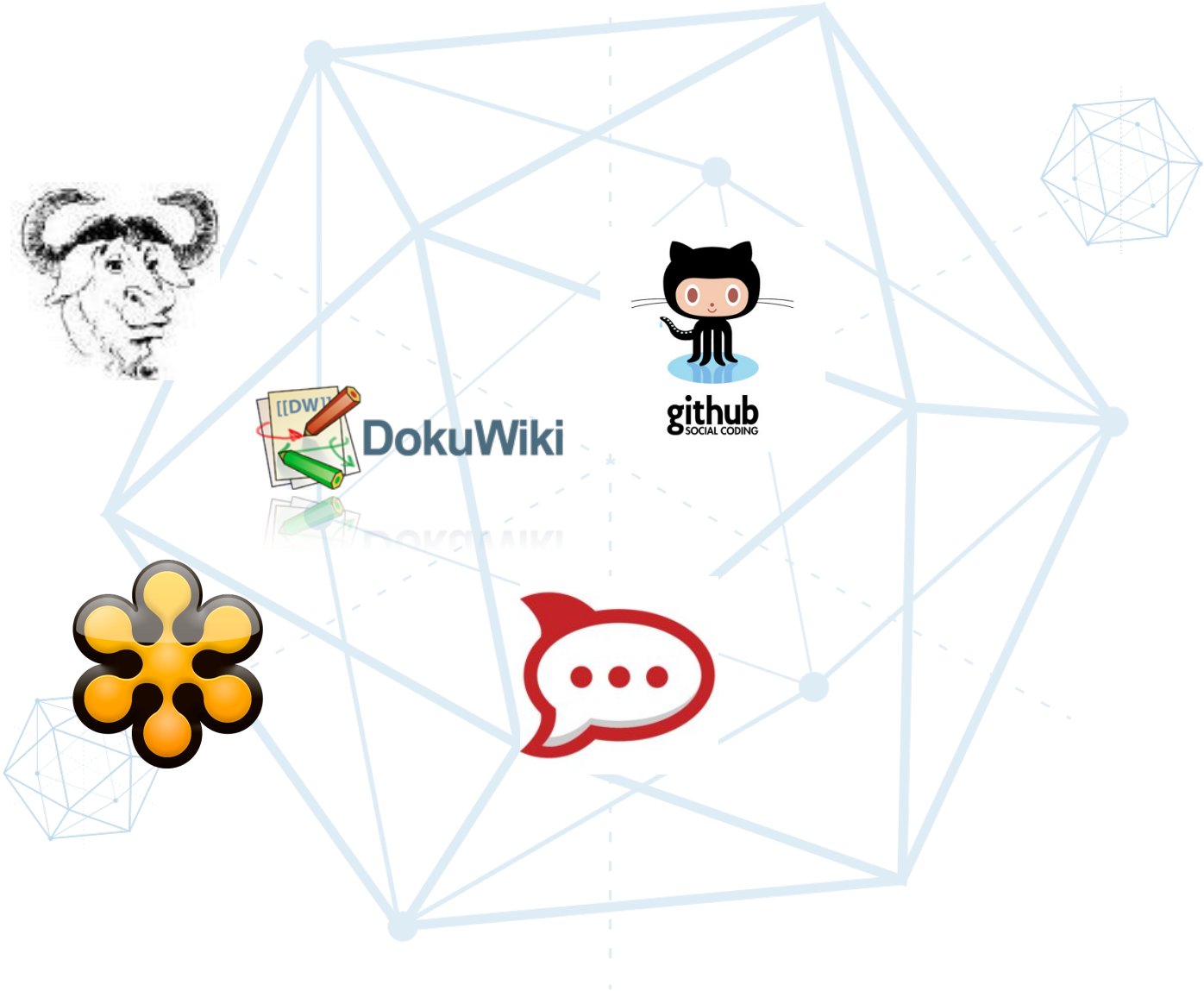
LINUX FOUNDATION

Hyperledger Community

- Events
  - Meetups
    - 13000+ attendees across 52 meetup groups
  - Hackathon
    - 1st Asia Hackathon at Shanghai on Mar 11,12
  - Hackfest
    - Beijing in June 19, 20
    - https://tinyurl.com/hackfest-beijing
- Wechat Public Channel

Open Communications

- Open Communications
  - Mail-list
  - Rocket chat
  - Meetings
  - Wiki
  - Github



Technical Working Group China

- About TWG-China
  - Since Dec 1, 2016
  - wiki.hyperledger.org/groups/tsc/technical-working-group-china
- Email
  - lists.hyperledger.org/mailman/listinfo/hyperledger-twg-china
  - hyperledger-twg-china@lists.hyperledger.org
- Online Chat
  - twg-china
- Call Meetings Bi-Wednesday Morning
  - 11AM at https://www.uberconference.com/hyperledger-community

Technical Working Group China

Innovation

Collaboration

Education

Events

Technical Working Group China

- 1st Asia Hackathon
  - Mar 11/12, 2017 at Shanghai
- 1st Asia Hackfest
  - June 19/20, 2017 at Beijing
- Meetups
  - Beijing, Shenzhen, Shanghai, Hangzhou
- Projects Development
- Educations/Trainings
- i18n and Documentation

- Welcome for more volunteers!

Hackfest

- June 19,29 at 北京北辰洲际酒店
- Free Registration
- Representative speakers from governing board, TSC and developers
- Agenda (updating)
  - Day 1: Hyperledger Project Overview, Working Groups, Fabric, Cello, Composer.
  - Day 2: How to Make Open-source Contribution, China Community, Iroha, ChainNova usage case, K8s based Baas solution, OnChain usage case.

TWGC Volunteers

Welcome for More Volunteers

ENERGY BLOCKCHAIN LABS

ChainNova

onchain

newell BRANDS

NTT

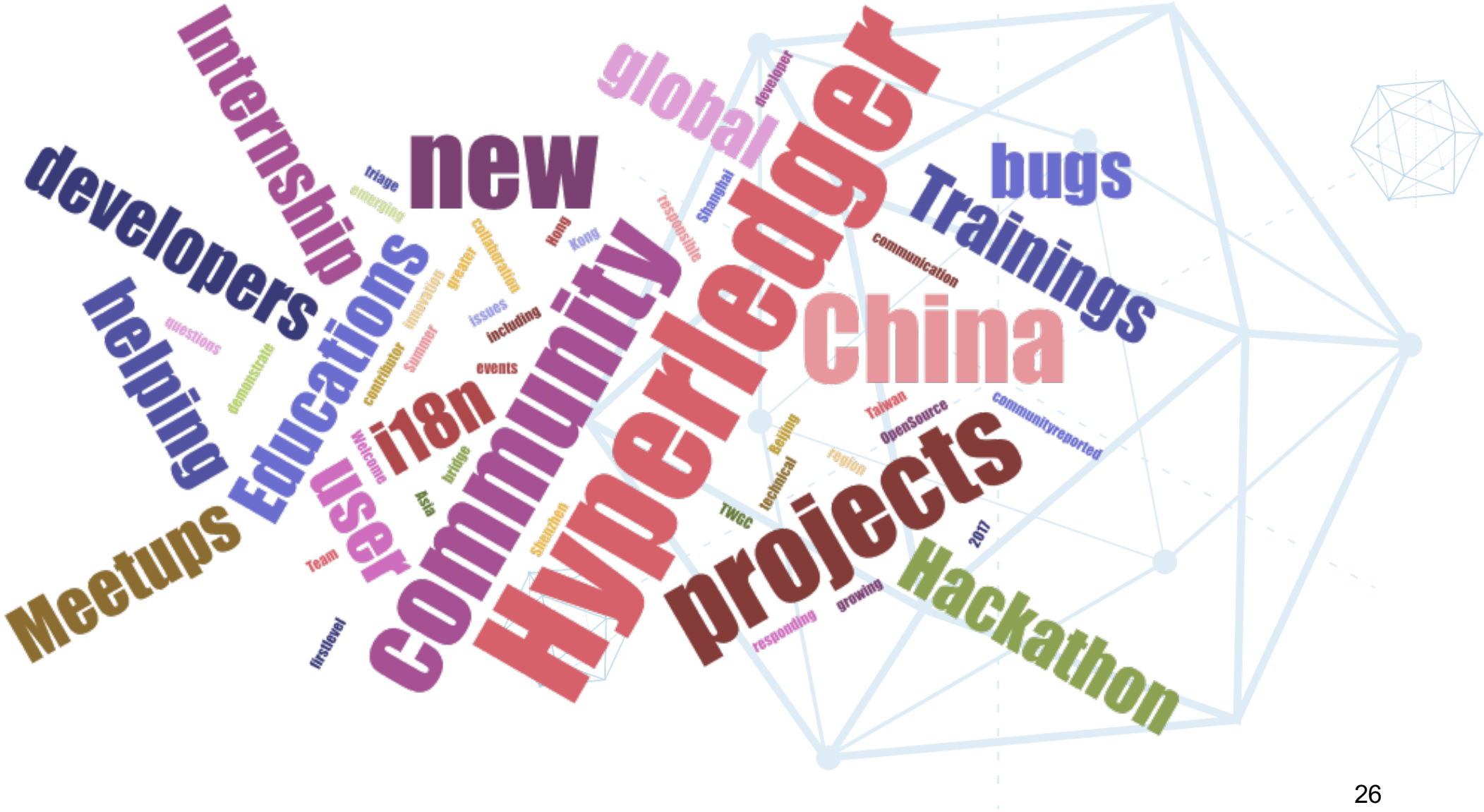
趣链科技 Hyperchain

万达集团 WANDA GROUP

IBM

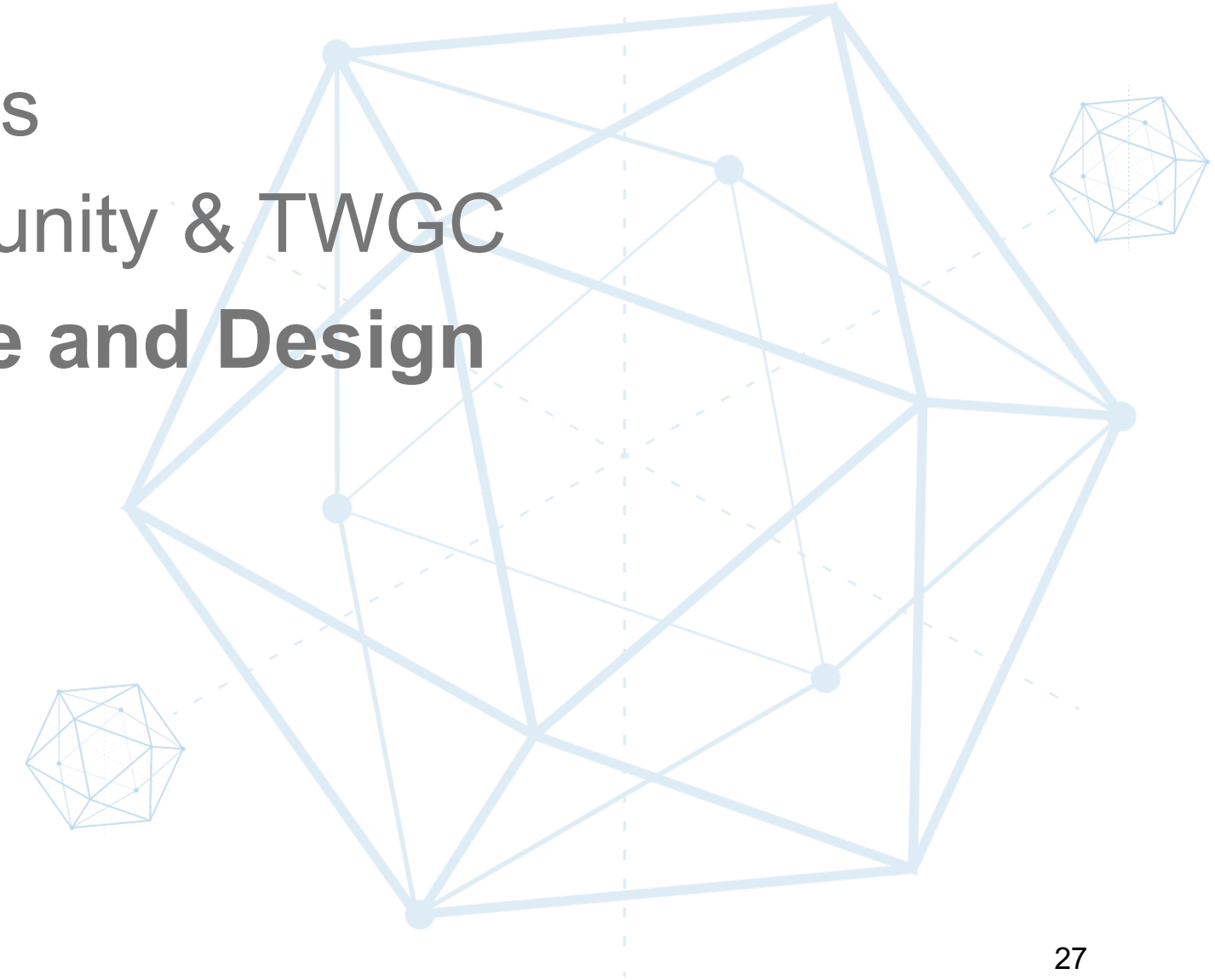
HITACHI

Technical Working Group China



Outline

- Hyperledger Projects
- Hyperledger Community & TWGC
- **Fabric Architecture and Design**
- Q&A



Existing Blockchain Technologies

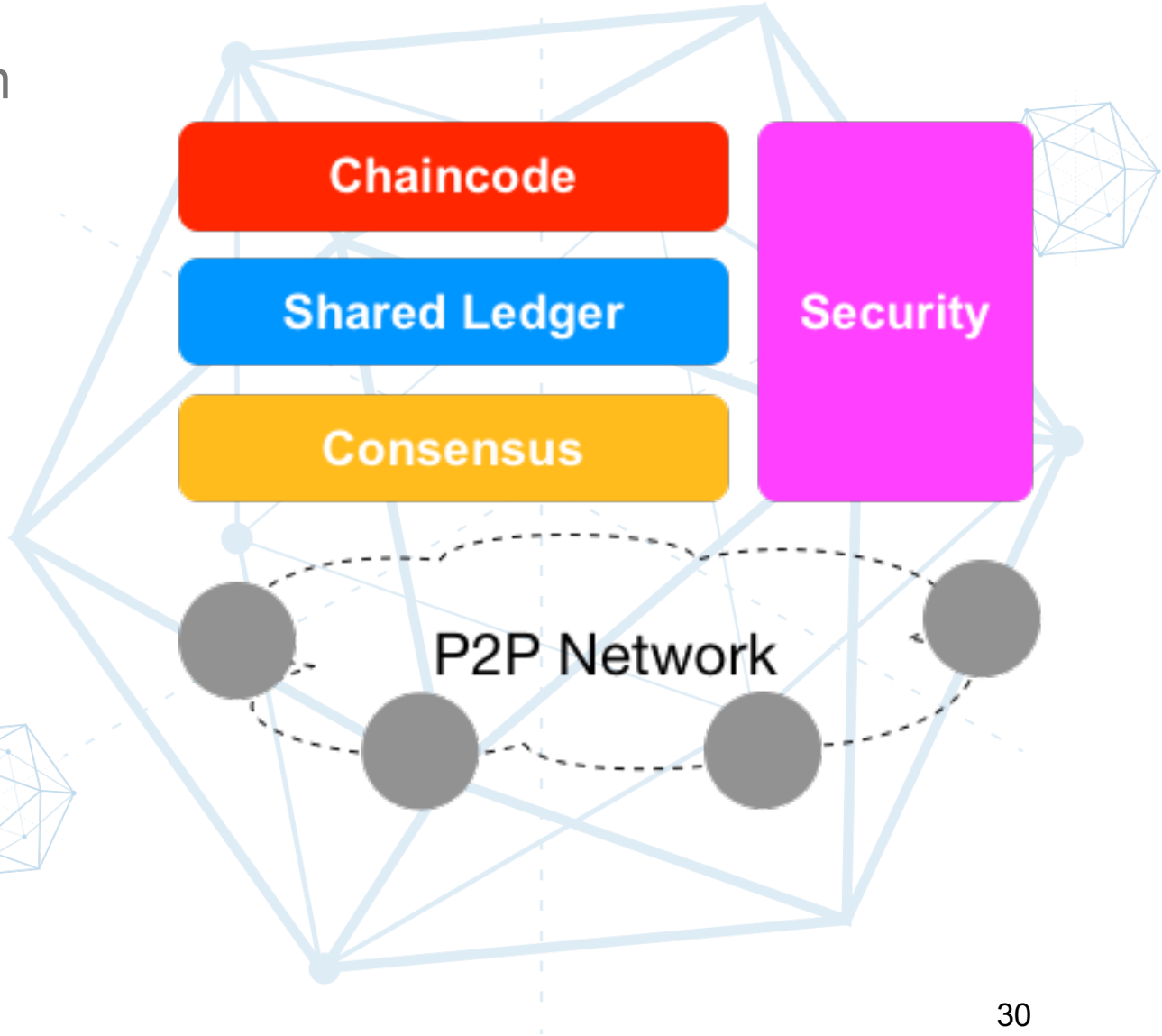
- Limited Throughput
- Slow Transaction Confirmation
- Designed for Cryptocurrency
- Poor Governance
- No Privacy
- No Settlement Finality
- Anonymous Processors
- …

Hyperledger Fabric: Ledger for Enterprise

- Privacy, Confidentiality, Auditability, Performance and Scalability
- Permissioned with better trust among members, while enable optimized consensus
- Open protocol/standard with open-source code

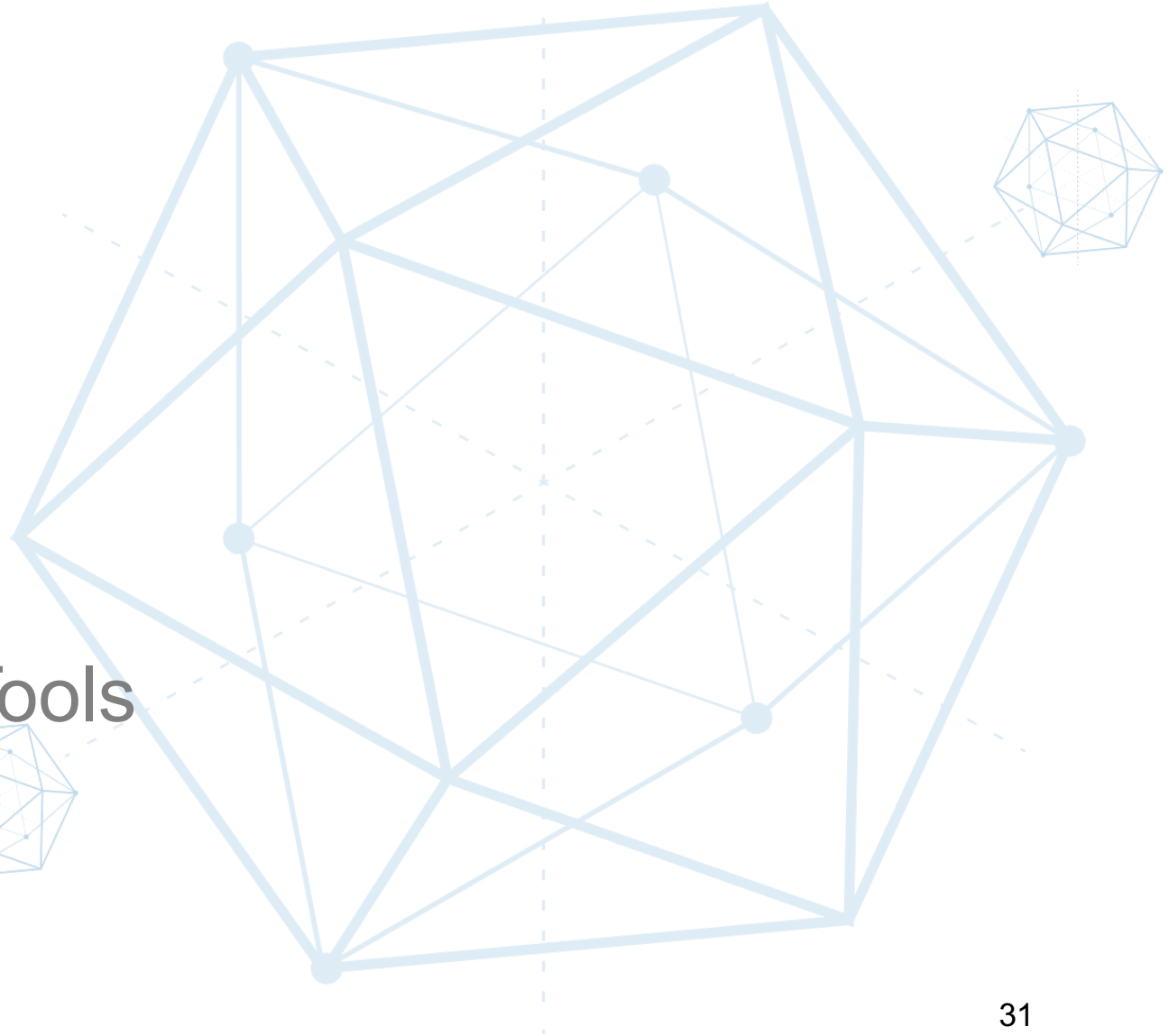
Fabric Main Components

- Shared Ledger
  - Append-only distributed record system
  - Blocks + States
- Smart Contract (Chaincode)
  - Business logics with transactions
  - Stateless and deterministic
- Consensus
  - Verified and ordered transactions
- Security
  - Access control
  - Privacy protection
  - Verification
  - CA



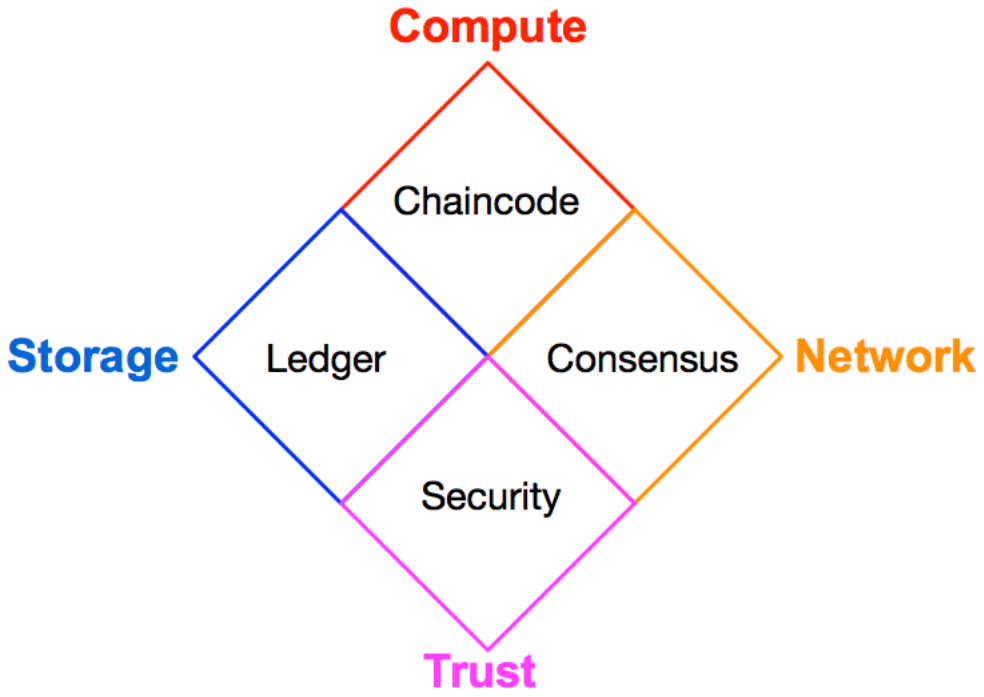
Fabric 1.0 New Design

- Node Functionality Decouple
- Multi-Channel/Chain
- Consensus
- Permission and Privacy
- System Chaincode
- Pluggable Components
- Configuration Management Tools



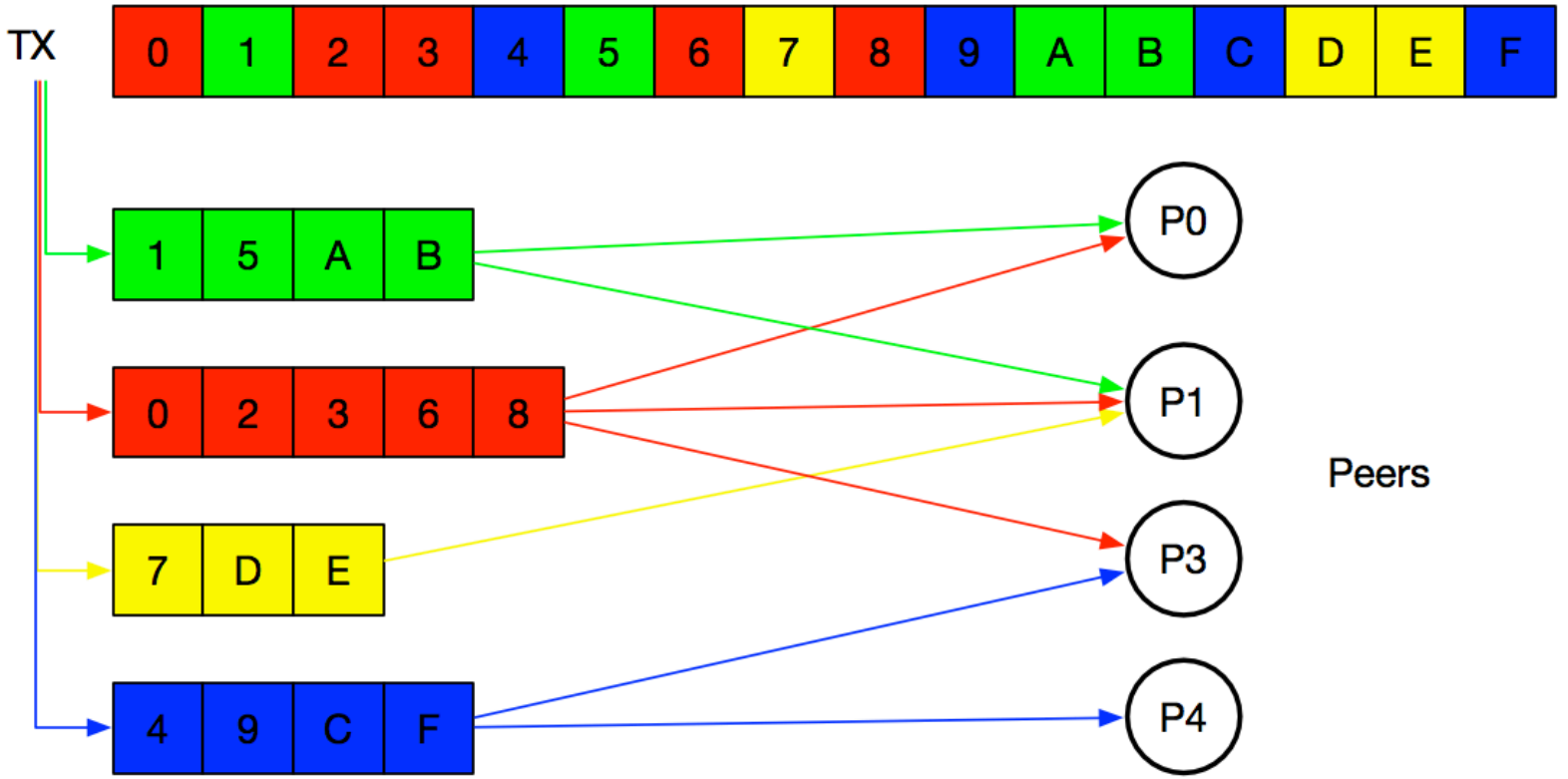
Node Functionality Decouple

- Various Intensive Requirements/Workloads
  - Chaincode: Compute intensive
  - Shared Ledger: Storage intensive
  - Consensus: Network intensive
  - Security: Trust intensive
- Decouple Full-functional Nodes
  - **Endorser**: Endorse TX proposal
  - **Committer**: Write down block
  - **Orderer**: Only order, no TX aware
  - **CA**: Certificate management



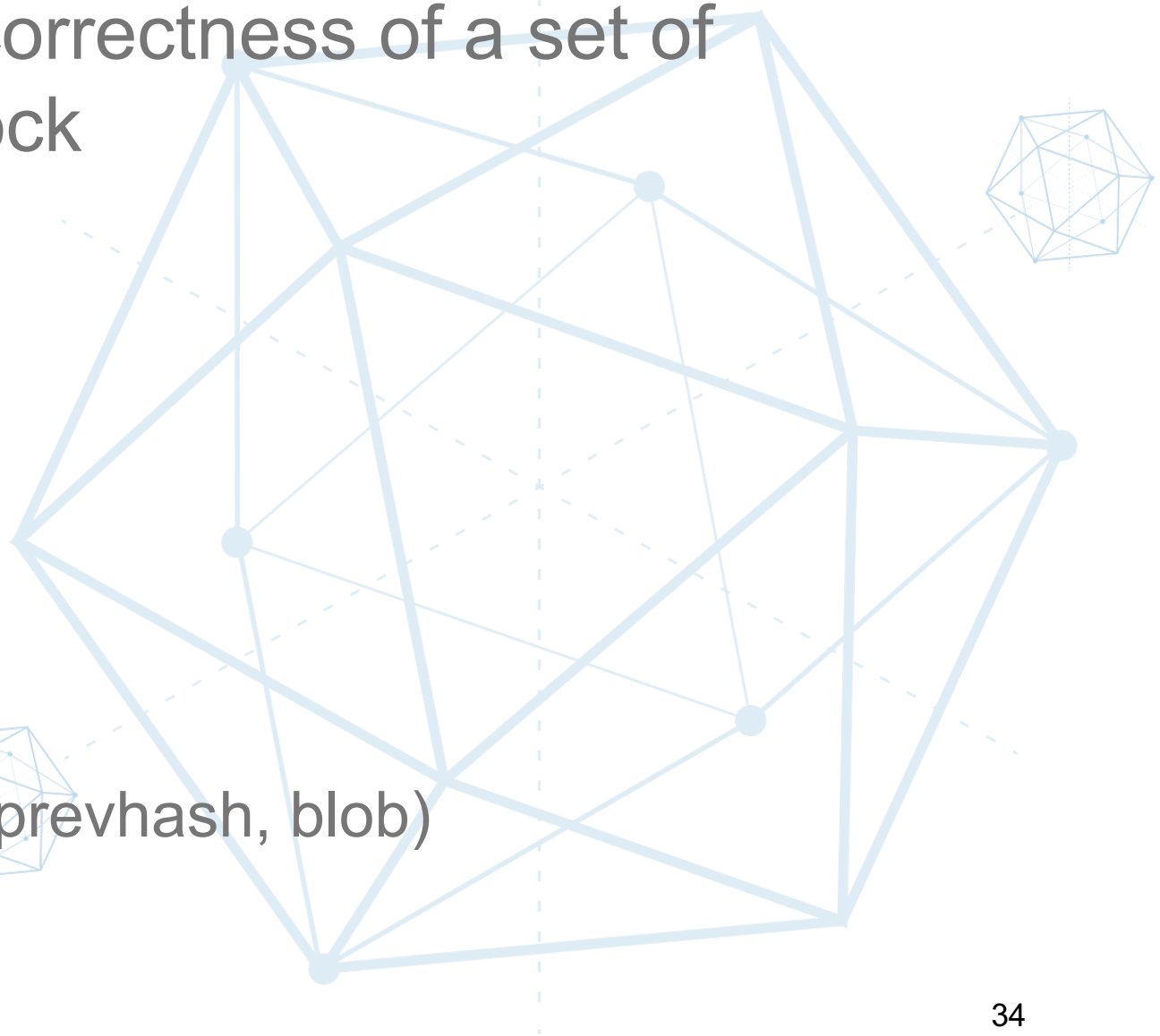
Multi-Channel/Chain

- Isolate the transactions, ledgers between organizations – overlay networks
- Peer can join channels accordingly



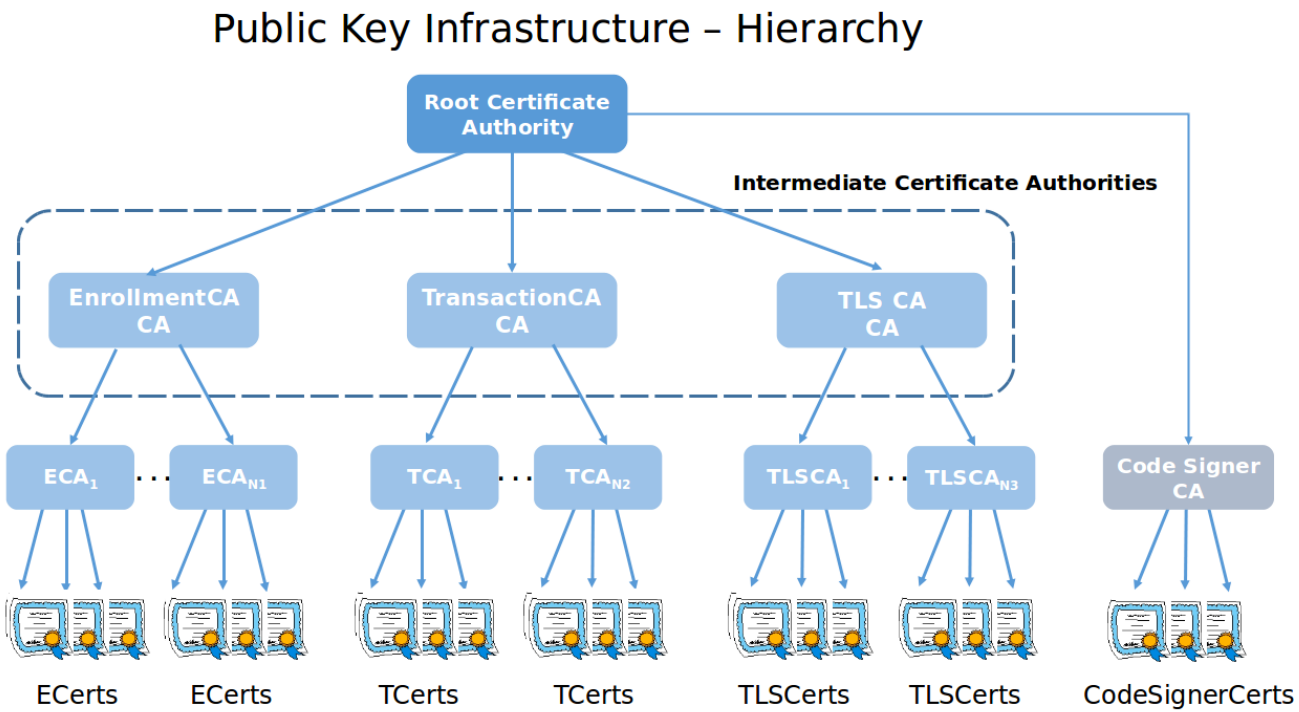
Consensus

- Full-circle verification of the correctness of a set of transactions comprising a block
  - Endorsement policy
  - Ordering
  - MVCC validation on RW sets
  - ACL
- Orderer
  - Solo, Kafka, BFT, and more…
  - Broadcast(blob), Deliver(seqno, prevhash, blob)



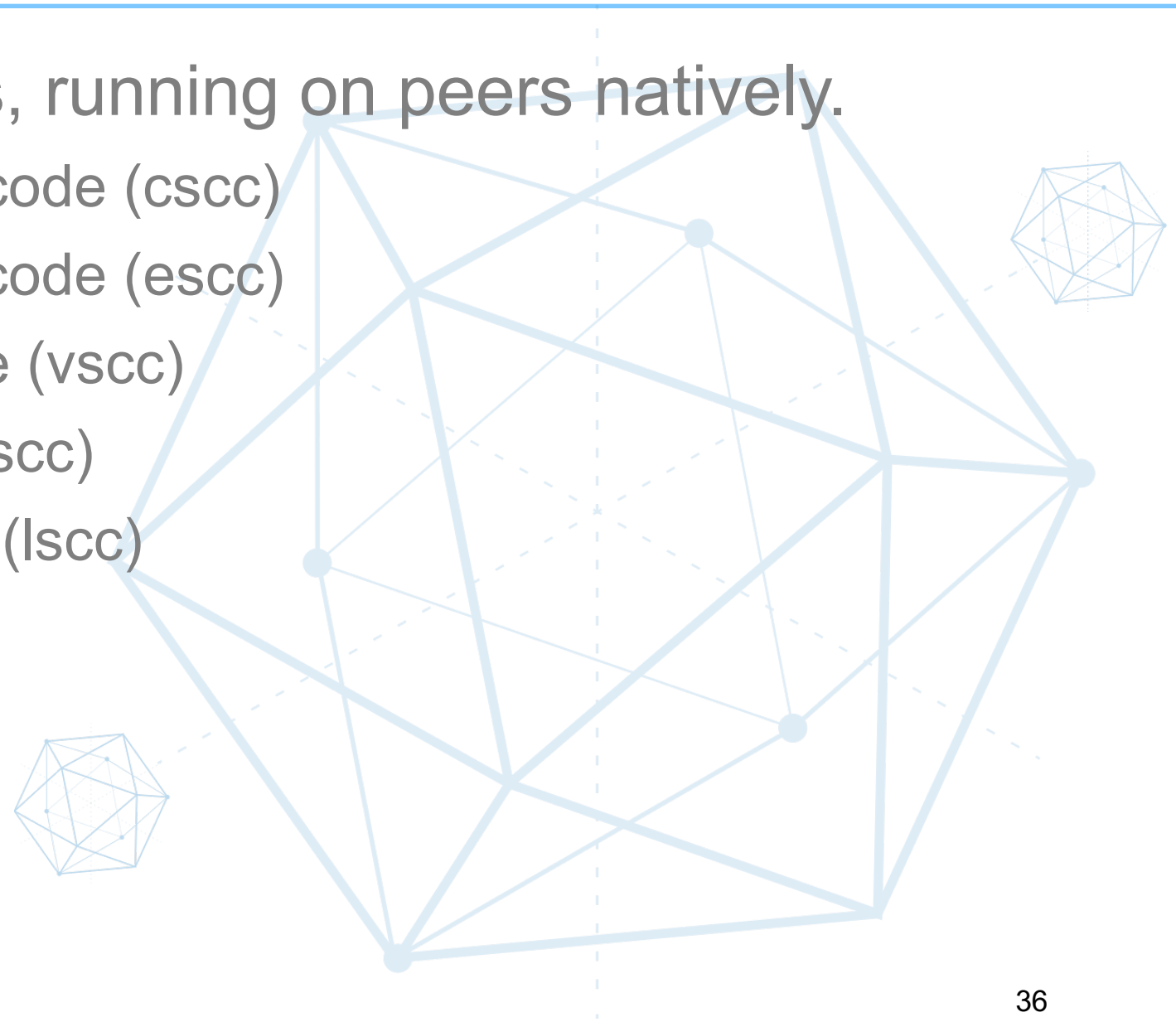
Permission and Privacy

- Permission at Various Levels
  - Network, channel, transaction
- Privacy for Business
  - Anonymity
  - Un-linkability
  - Auditability and Accountability
- Fabric CA (PKI)
  - Identity Registration Management
  - Enrollment Cert (Ecert)
  - Transaction Cert (Tcert, not supported in 1.0)



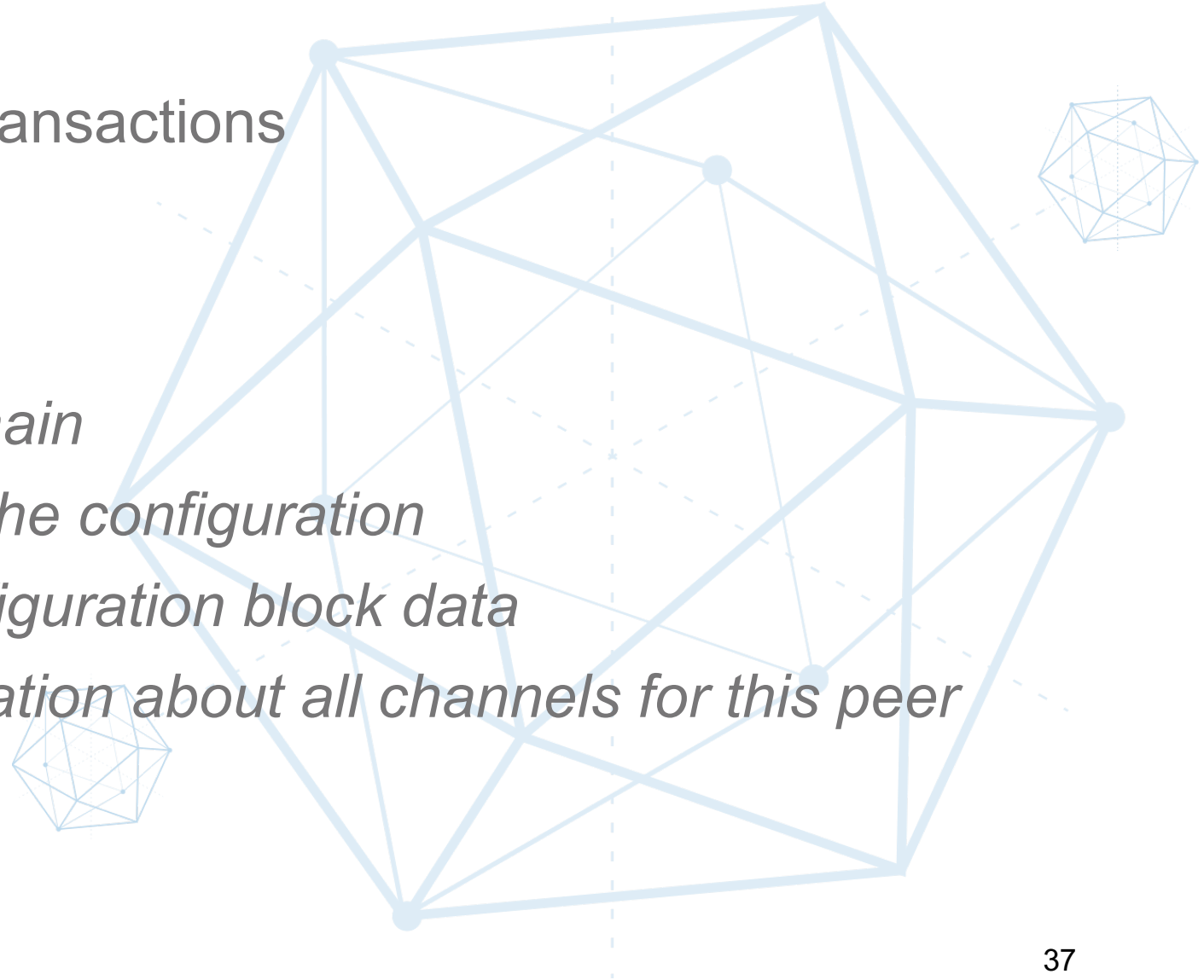
System Chaincode

- Handle system operations, running on peers natively.
  - Configuration System Chaincode (cscc)
  - Endorsement System Chaincode (escc)
  - Validation System Chaincode (vscc)
  - Query System Chaincode (qscc)
  - Lifecycle System Chaincode (lscc)



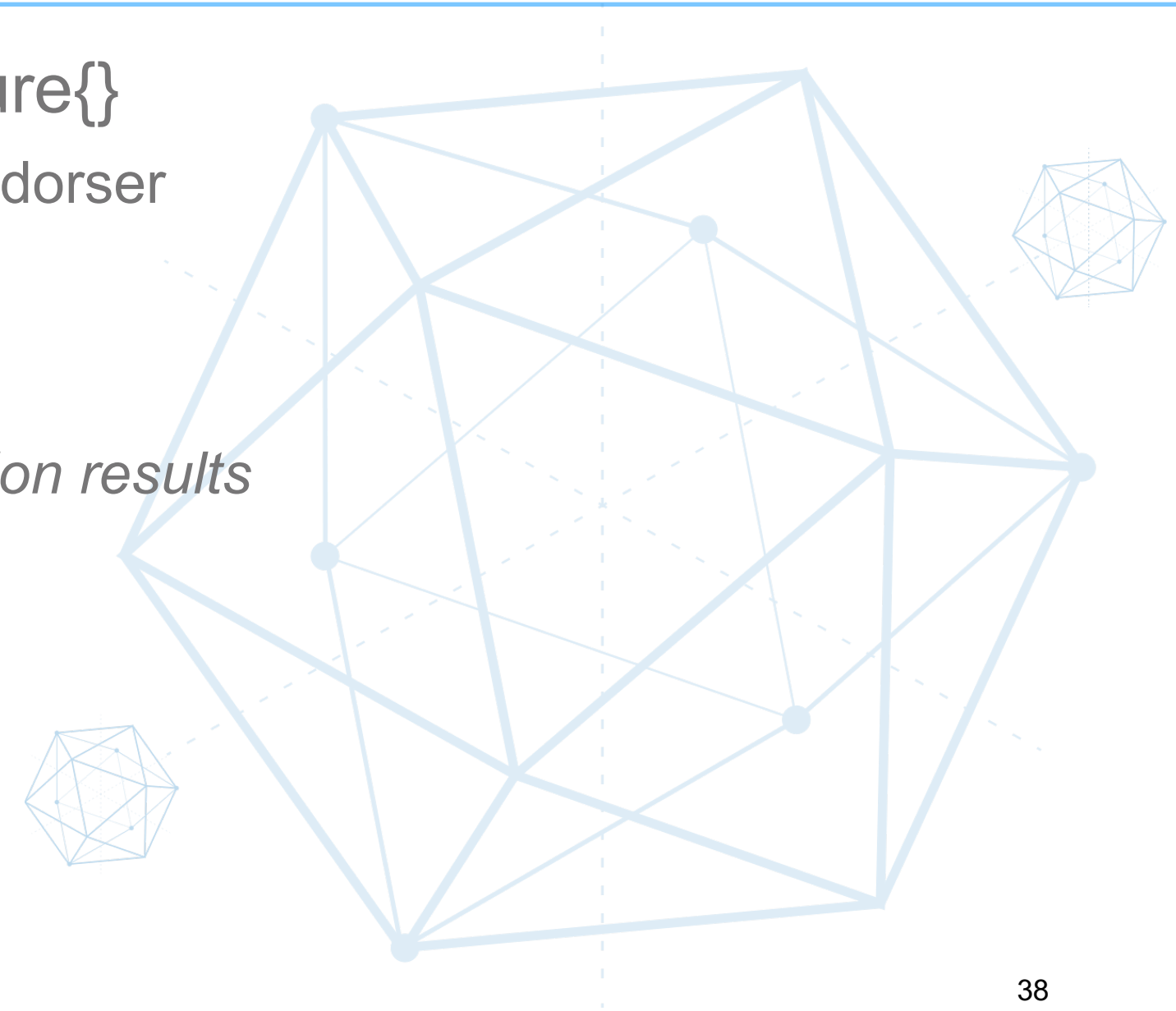
Configuration System ChainCode

- PeerConfiger{}
  - Handle those configuration transactions
- Init()
- Invoke()
  - JoinChain: *peer join into a chain*
  - UpdateConfigBlock: *update the configuration*
  - GetConfigBlock: *get the configuration block data*
  - GetChannels: *returns information about all channels for this peer*



Endorsement System ChainCode

- EndorserOneValidSignature{}
  - Endorsement process on Endorser
- Init()
- Invoke()
  - *Sign on chaincode's simulation results*
  - *More explicit rules (TBD)*



Validation System ChainCode

- ValidatorOneValidSignature{}
  - Validation process on Committer
- Init()
- Invoke()
  - *Validate the specified block of transactions, e.g., rwsets, signatures*

Query System ChainCode

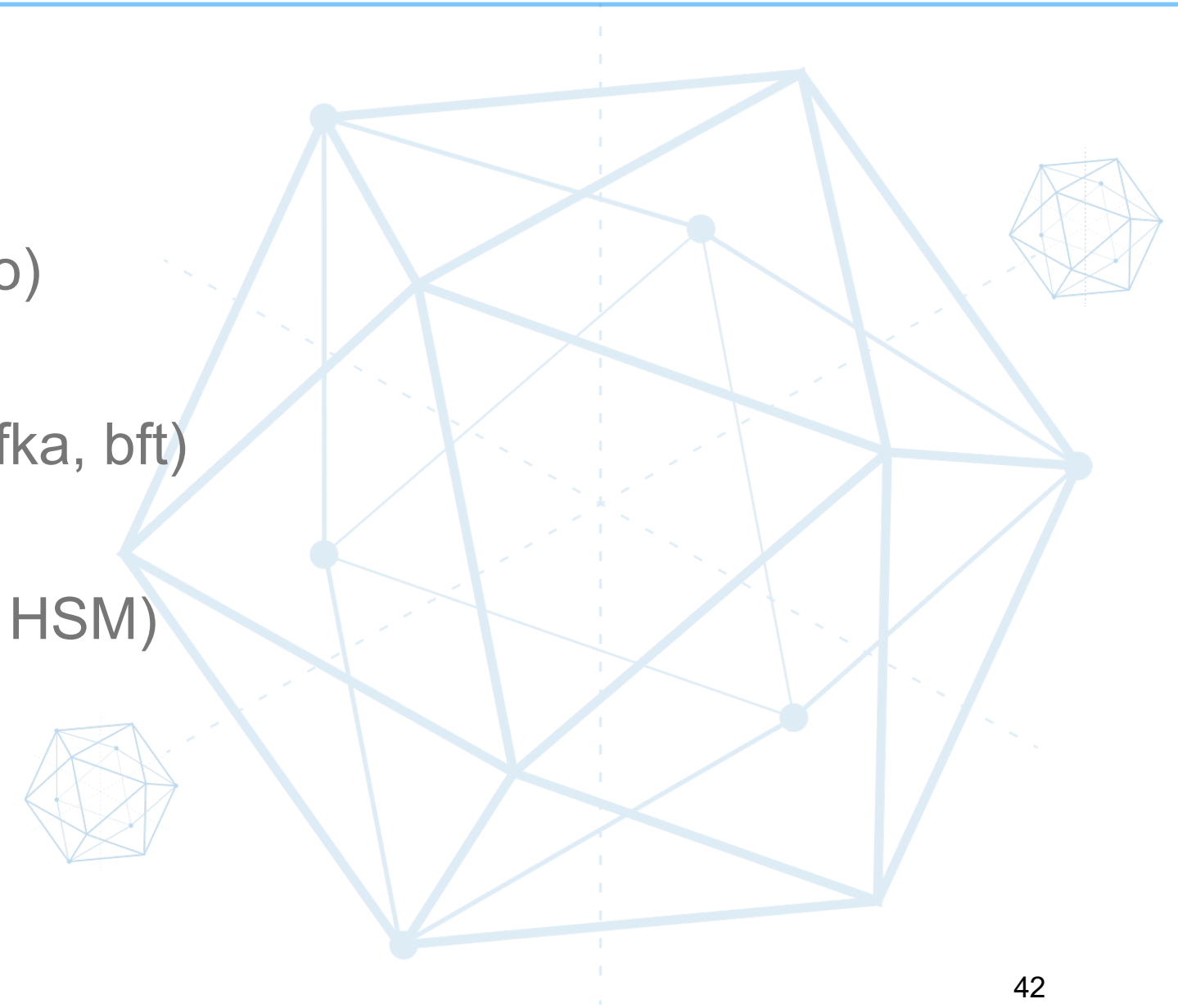
- LedgerQuerier{}
  - Ledger query functions
- Init()
- Invoke()
  - *GetChainInfo: Get information of a chain*
  - GetBlockByNumber: Get the block data by its number
  - GetBlockByHash: Get the block data by its hash value
  - GetTransactionByID: Get the transaction data by its id
  - GetBlockByTxID: Get the block data by contained transaction id

Life-cycle System ChainCode

- LifecycleSysCC{}
  - Application chaincode lifecycle management process
- Init()
- Invoke()
  - install: *install a chaincode on a peer*
  - deploy: *deploy a chaincode on a peer*
  - upgrade: *upgrade a chaincode*
  - getid: *get chaincode info*
  - getdepspec: *get ChaincodeDeploymentSpec*
  - getccdata: *get ChaincodeData*
  - getchaincodes: *get the instantiated chaincodes on a channel*
  - getinstalledchaincodes: *get the installed chaincodes on a peer*

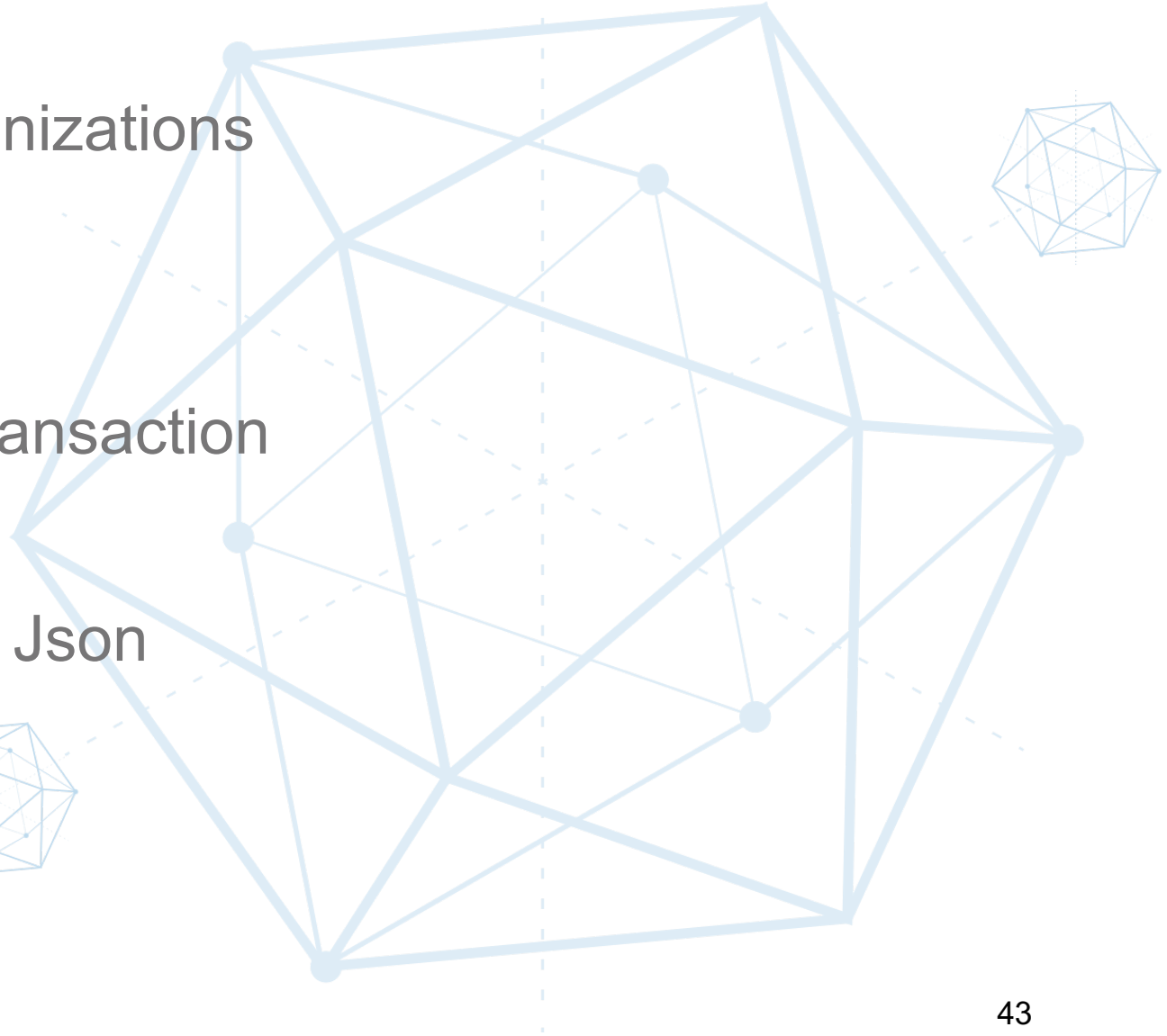
Pluggable Components

- Modular and Pluggable
  - Membership Services (CA)
  - SDKs (node, python, java, go)
  - Endorsement/Verification
  - Consensus service (solo, kafka, bft)
  - Ledger
  - Crypto algorithms (software, HSM)

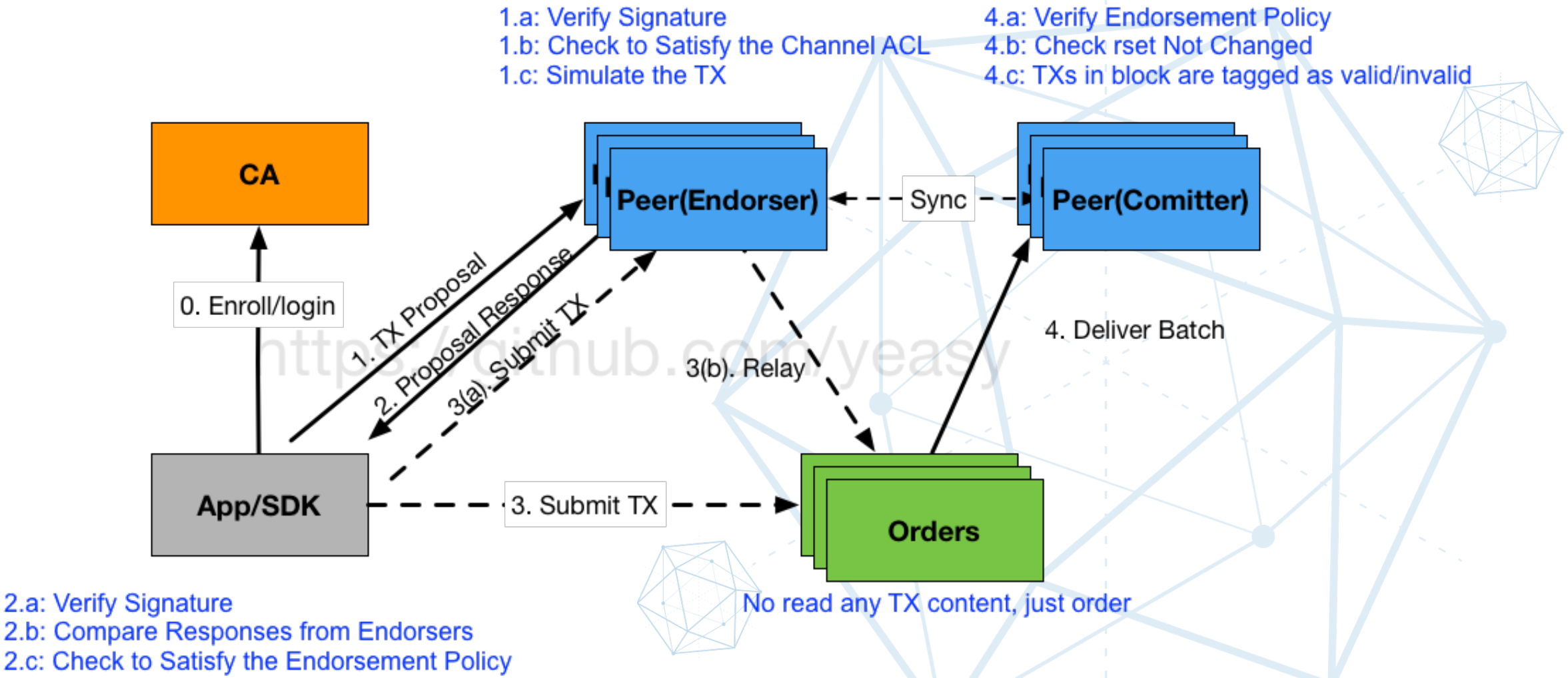


Configuration Management Tools

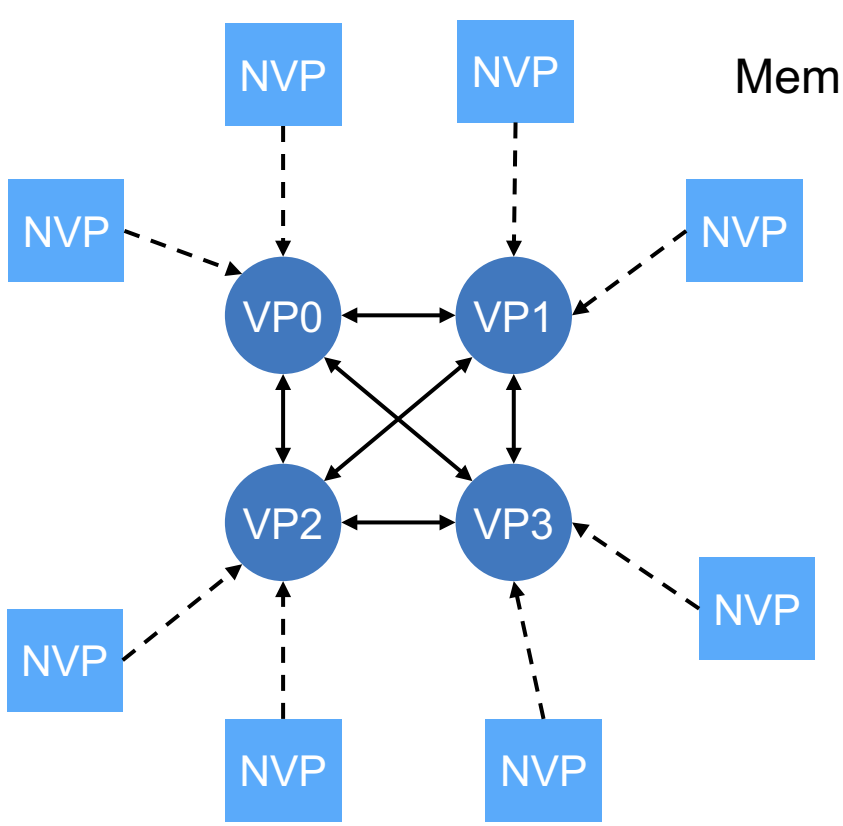
- Cryptogen
  - Generate certificate files for organizations
- Configtxgen
  - Generate genesis block
  - Generate configuration update transaction
- Configtxlator
  - Convert configuration artifacts to Json



Fabric 1.0 Workflow

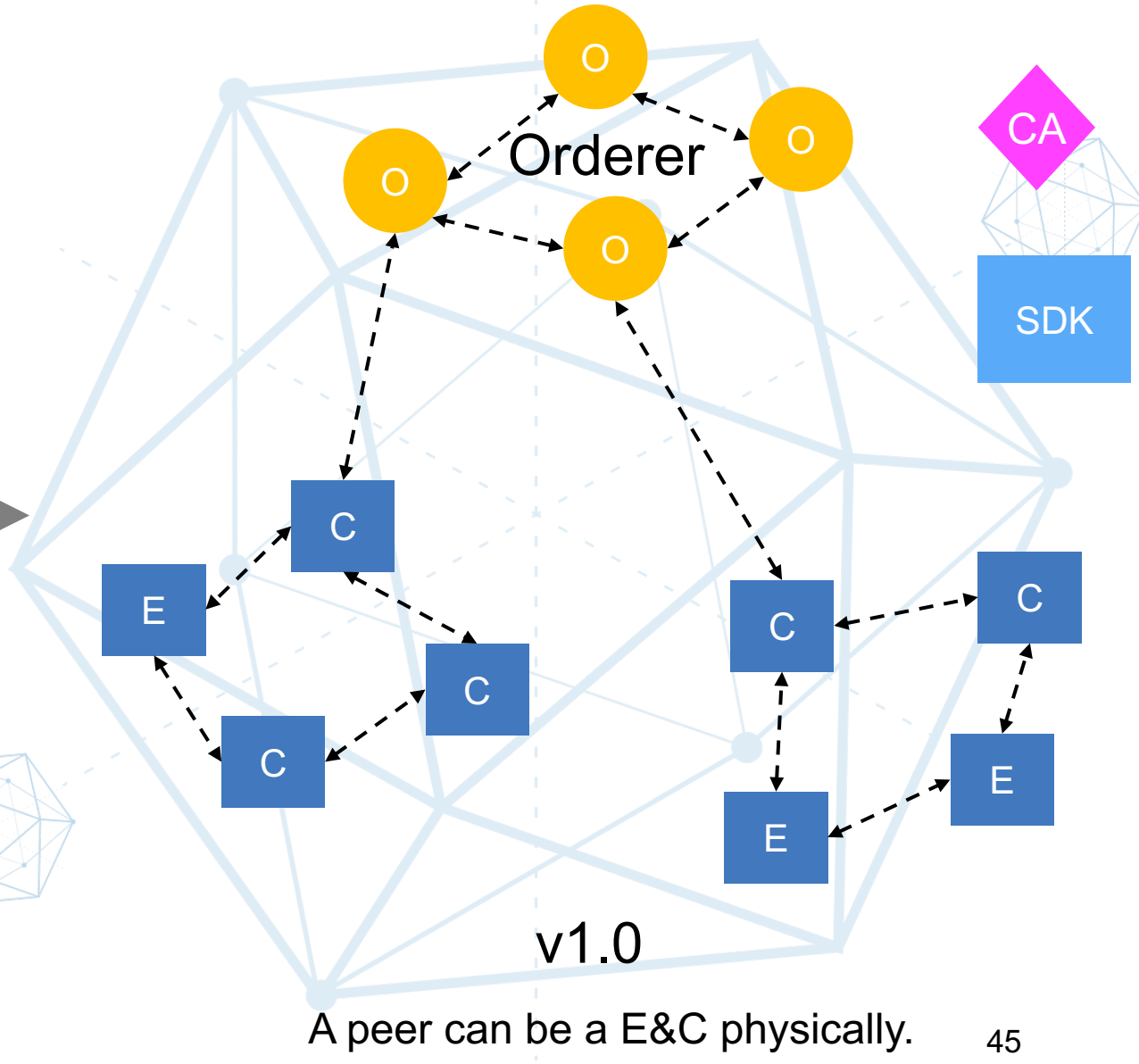


Fabric 1.0 Deployment Scenarios



Member Service

v0.6



v1.0

A peer can be a E&C physically.

Hyperledger Fabric Roadmap

Hack Fest docker images

- **60 participates tested**
- Basic v1 architecture in place
- Add / Remove Peers
- Channels
- Node SDK
- Go Chaincode
- Ordering Solo
- Fabric CA

V1 Alpha *

- Docker images
- Tooling to bootstrap network
- Fabric CA or bring your own
- Java and Node SDKs
- Ordering Services - Solo and Kafka
- Endorsement policy
- Level DB and Couch DB
- Block dissemination across peers via Gossip

V1 GA *

- Hardening, usability, serviceability, load, operability and stress test
- Java Chaincode
- Chaincode ACL
- Chaincode packaging & LCl
- Pluggable crypto
- HSM support
- Consumability of configuration
- Next gen bootstrap tool (config update)
- Config transaction lifecycle
- Eventing security
- Cross Channel Query
- Peer management APIs
- Documentation

V Next *

- SBFT
- Archive and pruning
- System Chaincode extensions
- Side DB for private data
- Application crypto library
- Dynamic service discovery
- REST wrapper
- Python SDK
- Identity Mixer (Stretch)
- Tcerts

2016/17 December | March | June | Future

Connect-a-thon

- 11 companies in Australia, Hungary, UK, US East Coast, US West Coast, Canada dynamically added peers and traded assets

Connect-a-cloud

- Dynamically connecting OEM hosted cloud environments to trade assets

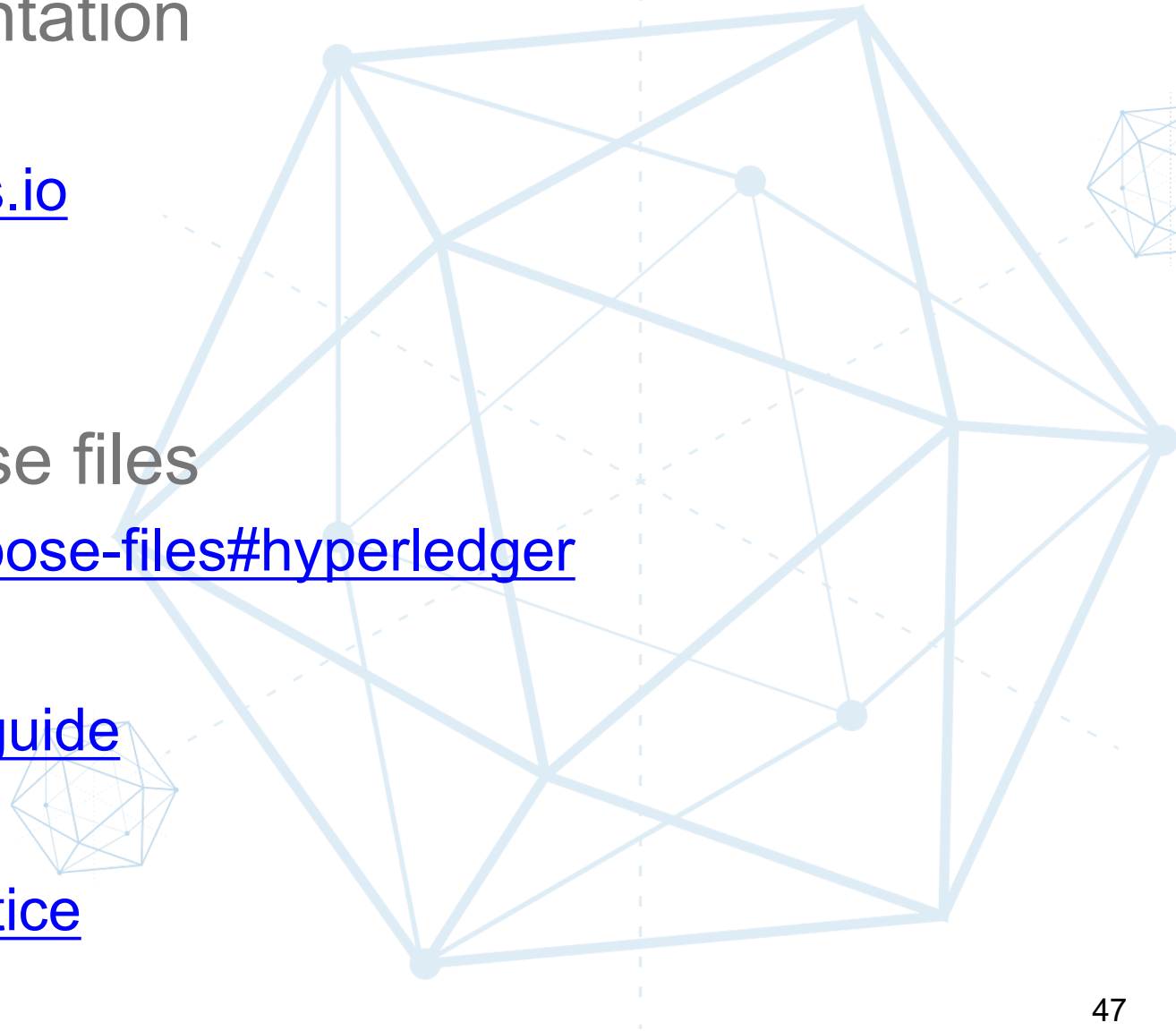
HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

*** Dates for Alpha, Beta, and GA are determined by Hyperledger community and are currently proposals.**

Proposed Alpha detailed content:
https://wiki.hyperledger.org/projects/proposedv1alphacontent

Reference

- Hyperledger Wiki&Documentation
  - wiki.hyperledger.org
  - hyperledger-fabric.readthedocs.io
- IBM 区块链
  - ibm.com/ibm/cn/blockchain/
- Hyperledger Fabric Compose files
  - github.com/yeasy/docker-compose-files#hyperledger
- 《区块链技术指南》
  - github.com/yeasy/blockchain_guide
- 《Docker 从入门到实践》
  - github.com/yeasy/docker_practice



HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

Questions?

Thank You!
@baohua

Slides available at github.com/yeasy/seminar-talk