GOPHER CHINA 2020

中国 上海 / 2020-11.21-22

如何用go module构建模块化跨链平台

汪小益 趣链科技

趣链科技

1 什么是区块链

2 跨链的重难点分析

3 跨链平台架构设计

4 Go module和plugin 的应用实践

趣链科技

1 什么是区块链

1 什么是区块链

趣链科技

交易

交易指的是一次对账本的操作，如一笔转账交易。

区块

将一段时间内发生的所有交易和状态打包成为一个区块。

块链式数据结构（狭义区块链）

区块以时间顺序前后相连，组成一种块链式数据结构，即“区块链”一词的由来。

分布式账本（广义区块链）

多参与方各自部署，互联互通，构成分布式网络。

1 异构区块链生态

趣链科技

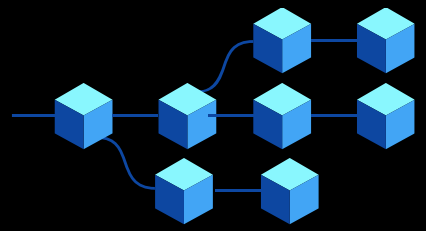
币/模式

比特币

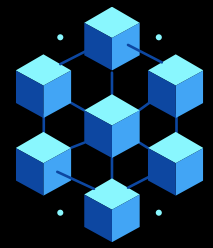
天秤币
（由Facebook发行）

摩根币
（由摩根大通发行）

链/技术



公有链
（Bitcoin/Ethereum/EOS）



联盟链
（Fabric/趣链区块链...）

趣链科技

2 跨链的重难点分析

什么是跨链

区块链之间实现可信互操作

资产交换

区块链账本间的资产互操作

数据交换

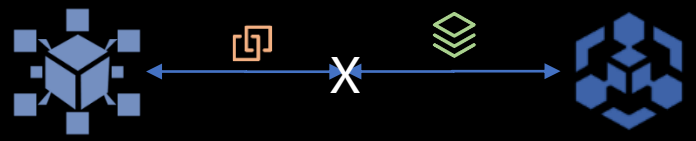
区块链上数据的共享与同步

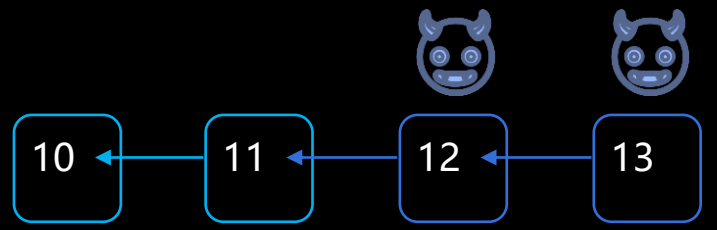
业务互补

调用他链服务完善己链业务

2 跨链难点分析

趣链科技

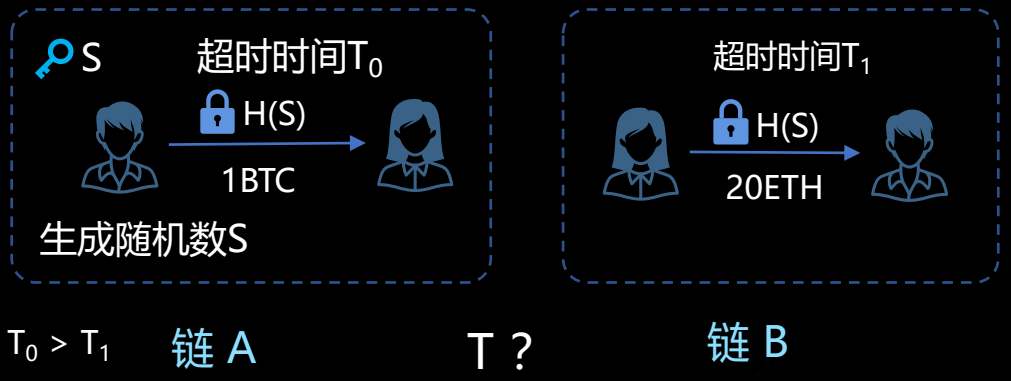




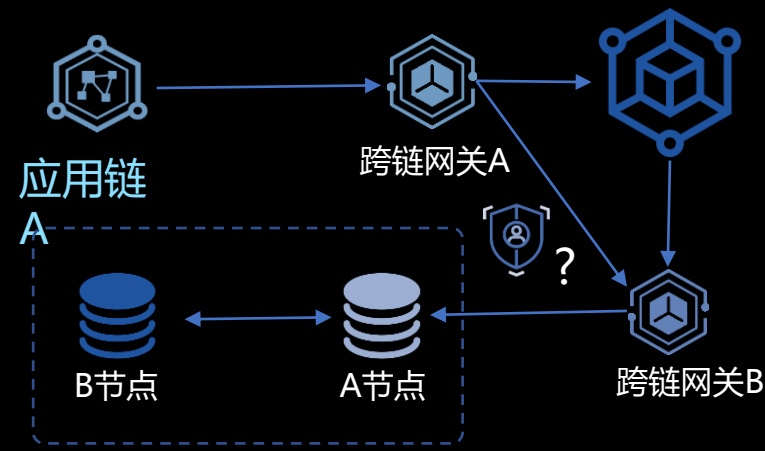
1 支持异构区块链

2 跨链交易存在性和有效性

3 跨链事务难



4 隐私保护和权限控制



2 跨链难点分析

趣链科技

解决方案

BitXHub
跨链技术平台

- 实现多中心化可信中继的中继链
- 适用于异构区块链的跨链消息传输协议
- 实现无侵入适配不同区块链的跨链网关
- 支持异构区块链交易验证的验证引擎
- 适用于不同场景的多种跨链事务方案
- 多层级的隐私保护与权限控制机制

难点

- 支持异构区块链
- 跨链交易存在性和有效性
- 跨链事务难
- 隐私保护和权限控制

趣链科技

3 跨链平台架构设计

3 跨链传输协议

趣链科技

IBTP 结构的 Fabric 交易示例

From	来源链 ID
To	目的链 ID
Index	跨链交易索引
Timestamp	跨链事件发生的时间戳
Payload	跨链调用内容编码
Proof	跨链交易证明
Version	协议版本号

Encrypted	false
Content	
Chaincode_id	0x12345
Func	put
Args	bitxhub, 1000

加密

序列化

Proposal	Hash, Content
Endorser	$Cert_1$, $Cert_2$, $Cert_3$
Signature	S_1, S_2, S_3

IBTP：Inter Blockchain Transfer Protocol，是平台提出的一种通用的跨链交互的消息传输协议。

设计考量

1 **通用的交易格式**

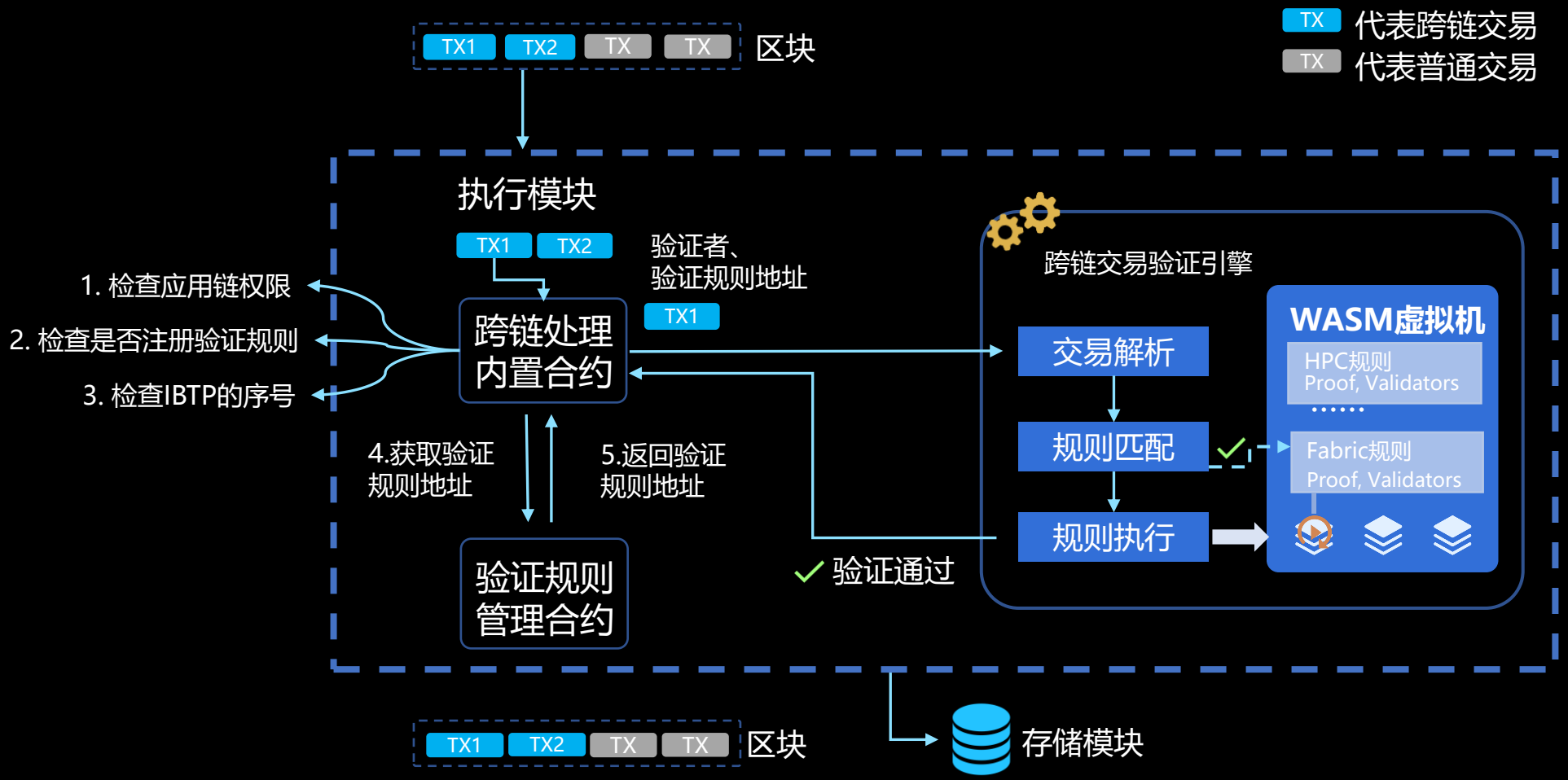
消除不同共识算法、加密机制等技术实现导致的交易合法性证明差异

2 **可扩展性**

证明信息和调用信息可根据链的特性进行适配

3 跨链验证引擎

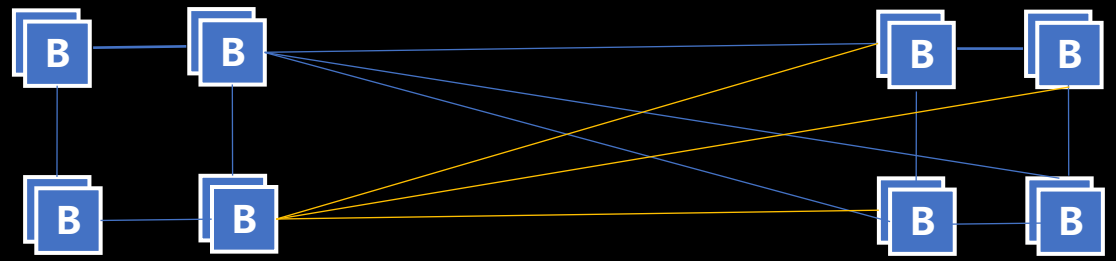
趣链科技



3 如何跨链互联

趣链科技

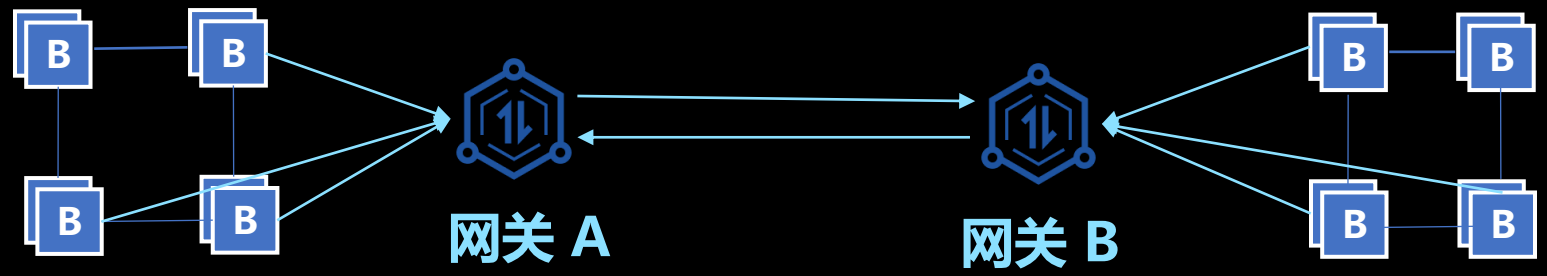
首先解决网络互通的问题多对多的网络拓扑结构



3 如何跨链互联

趣链科技

多对多转换成一对多的结构，降低网络拓扑复杂度
如何保障安全？



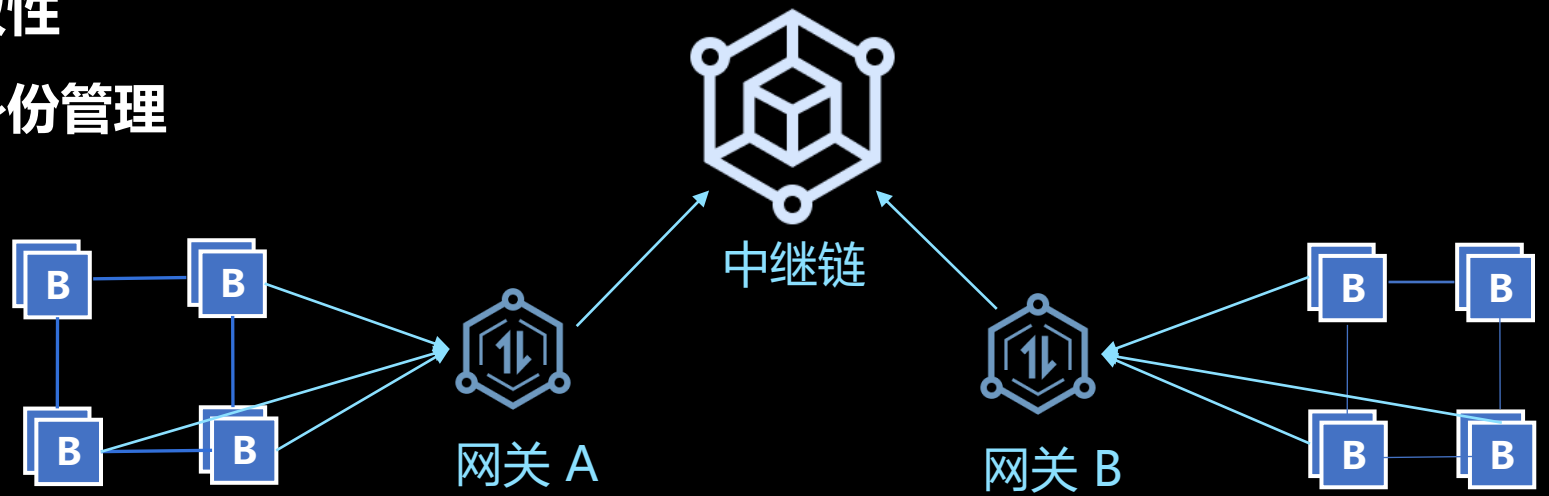
3 如何可信跨链互联

趣链科技

一对多的情况下如何保障安全性?

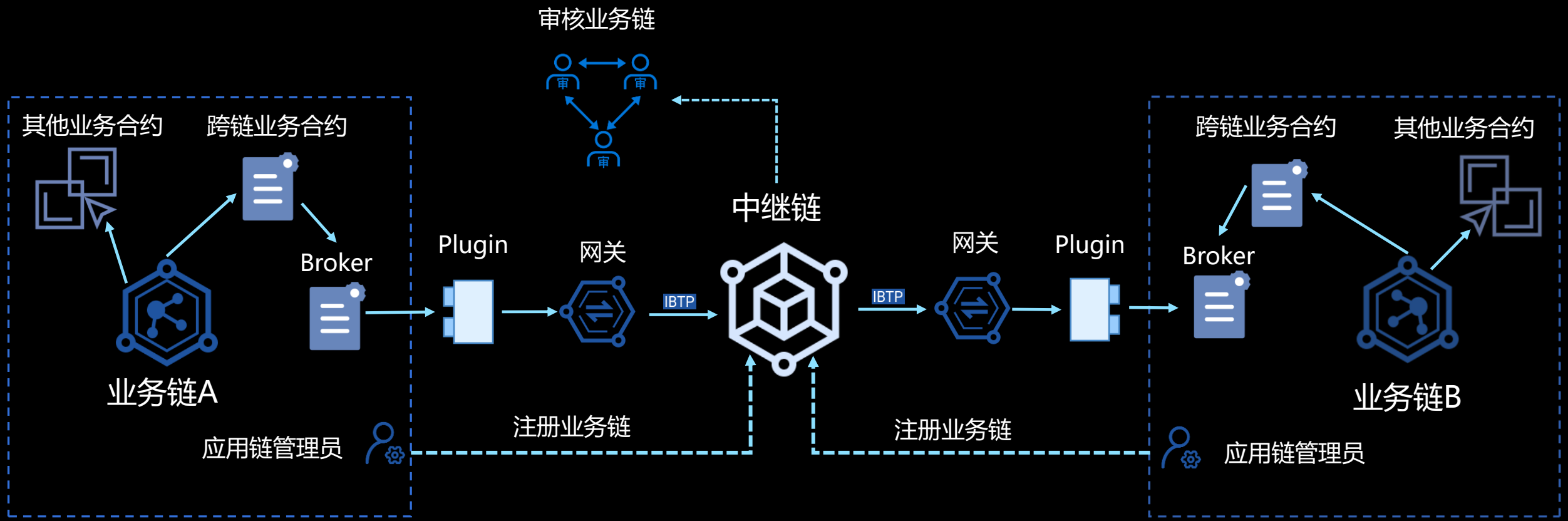
- 引入中继链协同见证

1. 校验跨链消息有效性和存在性
2. 保障跨链事务一致性
3. 提供跨链路由及身份管理



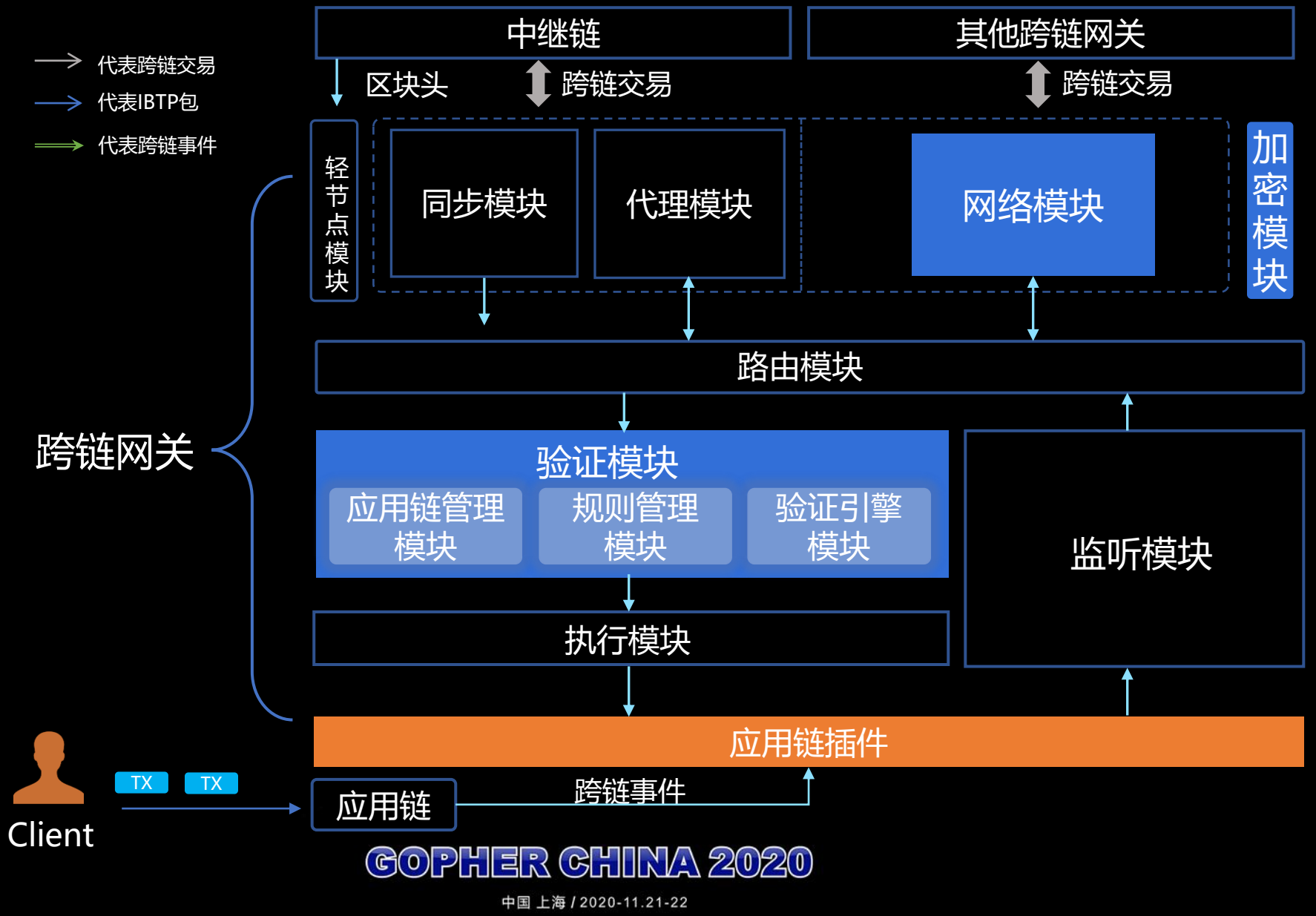
3 跨链网关与跨链中继

趣链科技



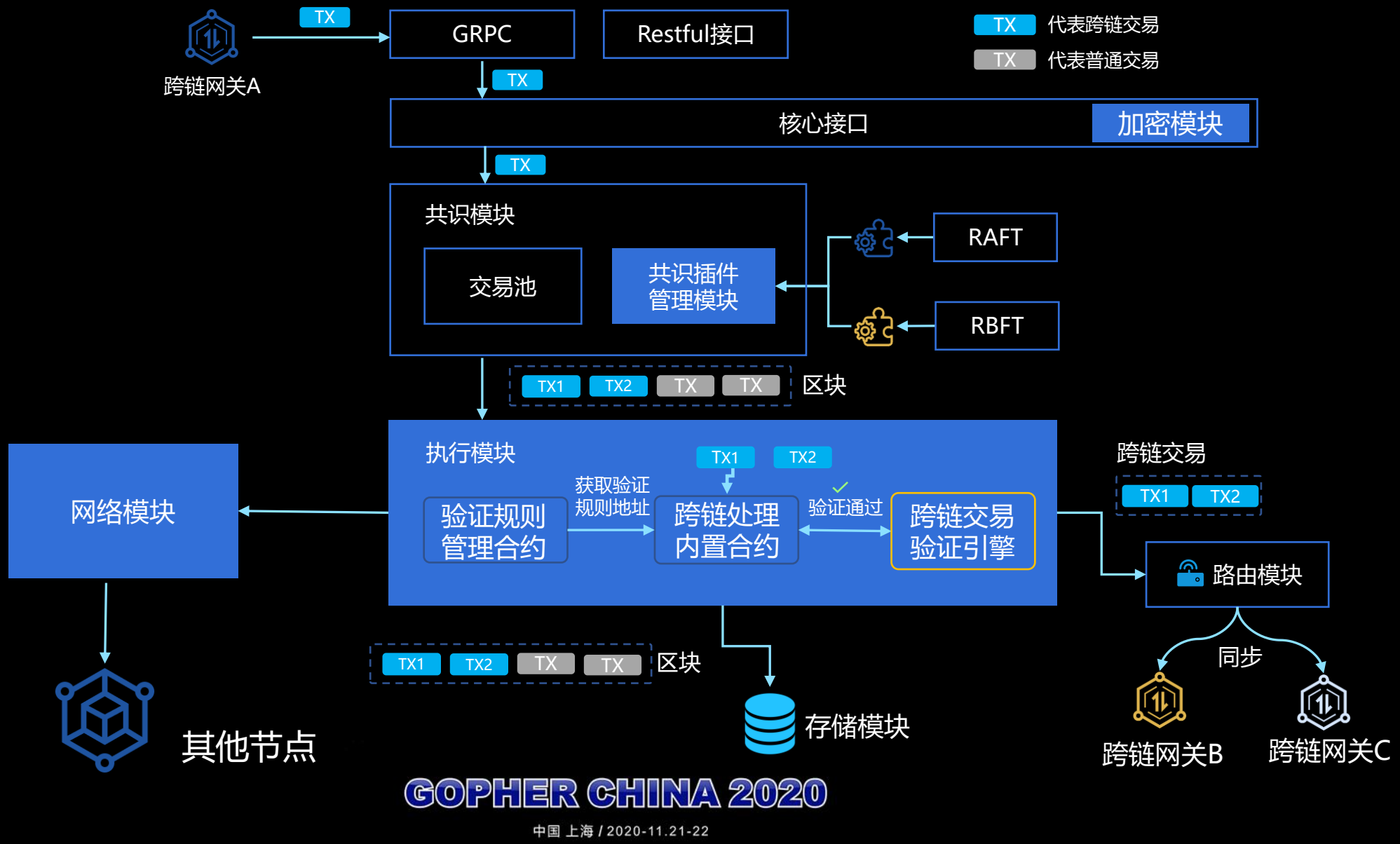
3 跨链网关与流程

趣链科技



3 中继链模块与流程

趣链科技



3 为什么需要模块化

趣链科技

1 中继链和网关有大量公共组件

- 网络模块
- 执行模块
- 验证模块
- 协议处理模块
- 加密模块

2 灵活适配不同需求和版本

- 应用链访问模块

以太坊　Fabric　趣链区块链　天平链

- 共识模块

PBFT　RAFT　POS　POA　HOTSTUFF

Golang提供了两种模块化工具，一个是go module，一个是go plugin

趣链科技

4 GO module和plugin实践

4 GO modules实践：无处不在的模块

趣链科技

工具库(存储、加密、网络、日志等)
https://github.com/meshplus/bitxhub-kit

验证引擎&IBTP协议库
https://github.com/meshplus/bitxhub-core

数据模型库
https://github.com/meshplus/bitxhub-model

中继链主仓
https://github.com/meshplus/bitxhub

跨链网关主仓
https://github.com/meshplus/pier

```
type Ledger interface {
    BlockchainLedger
    StateAccessor

    AccountCache() *AccountCache

    // PersistBlockData
    PersistBlockData(blockData *BlockData)

    // AddEvent
    AddEvent(*pb.Event)

    // Events
    Events(txHash string) []*pb.Event

    // Rollback
    Rollback(height uint64) error

    // RemoveJournalsBeforeBlock
    RemoveJournalsBeforeBlock(height uint64) error

    // Close release resource
    Close()
}
```

4 GO modules实践：经验建议

趣链科技

GO module几个好用的命令

- go mod tidy　　移除无效依赖，新增缺少的依赖
- go mod graph　　列出所有的依赖
- go mod vendor　　依赖复制打包到vendor目录下（网络受限）
- go mod why　　分析依赖原因
- replace　　依赖包信息替换

eg: replace github.com/ultramesh/crypto-gm =>
git.hyperchain.cn/ultramesh/crypto-gm.git v0.1.0

4 GO modules实践：经验建议

趣链科技

- Go的项目模块架构划分尽量以功能为单位垂直拆分
- 模块不要拆分的太细，过多交叉依赖管理会很头疼
- 配置好goproxy，方便私有化仓库管理以及依赖加速

```
go env -w GO111MODULE=on
go env -w GOPROXY=https://goproxy.cn,direct
```

4 GO plugin实践：共识算法为例

趣链科技

go build --buildmode=plugin -o build/solo.so order/solo/*.go

```go
//Load order plugin
func New(opts ...order.Option) (order.Order, error) {
    config, err := order.GenerateConfig(opts...)
    if err != nil {
        return nil, err
    }
    pluginPath := config.PluginPath

    if !filepath.IsAbs(pluginPath) {
        pluginPath = filepath.Join(config.RepoRoot, pluginPath)
    }

    p, err := plugin.Open(pluginPath)
    if err != nil {
        return nil, fmt.Errorf("plugin open: %s", err)
    }

    m, err := p.Lookup("NewNode")
    if err != nil {
        return nil, fmt.Errorf("plugin lookup: %s", err)
    }

    NewNode, ok := m.(func(...order.Option) (order.Order, error))
    if !ok {
        return nil, fmt.Errorf("assert NewOrder error")
    }
    return NewNode(opts...)
}
```

```go
type Order interface {
    // Start the order service.
    Start() error

    // Stop means frees the resources which were allocated for this service.
    Stop()

    // Prepare means send transaction to the consensus engine
    Prepare(tx *pb.Transaction) error

    // Commit recv blocks form Order and commit it by order
    Commit() chan *pb.Block

    // Step send msg to the consensus engine
    Step(ctx context.Context, msg []byte) error

    // Ready means whether order has finished electing leader
    Ready() bool

    // ReportState means block was persisted and report it to the consensus engine
    ReportState(height uint64, hash types.Hash)

    // Quorum means minimum number of nodes in the cluster that can work
    Quorum() uint64

    // GetPendingNonce will return the latest pending nonce of a given account
    GetPendingNonceByAccount(account string) uint64
}
```

4 GO plugin实践：经验建议

趣链科技

- 使用接口类型作为插件的边界
- 一个插件提供的服务尽量单一
- 使用语义版本控制插件版本，如pbft.so.1.2.0
- 通过代码签名等手段进行安全校验

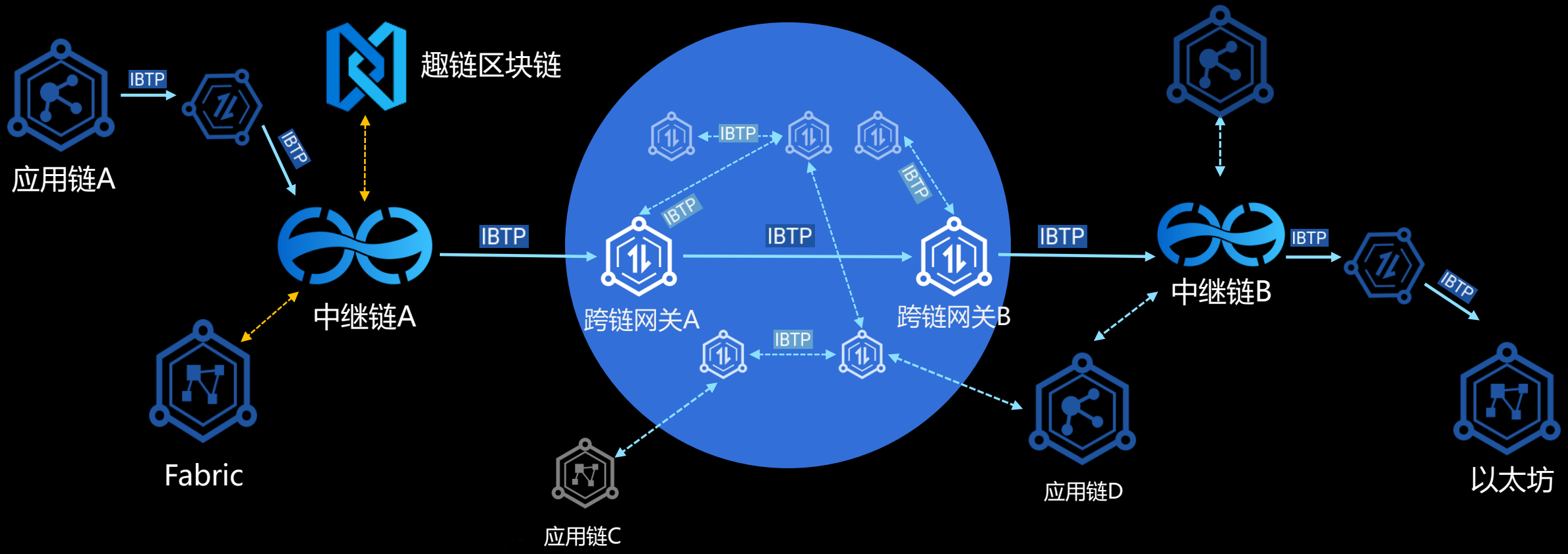
- 插件依赖的组件版本需要和主仓库所依赖的组件版本需要一致
- so 包编译出来的大小过大
- 插件加载之后内存占用较高
- 目前仅支持 Linux, FreeBSD, and macOS

4 模块化垮链实践结果

趋链科技

基于多模块可组合性形成的积木型跨链体系

- 链对链直接
- 中继见证
- 主侧扩展



GOPHER CHINA 2020

中国 上海 / 2020-11.21-22

THANKS

GitHub-BitXHub

添加趣链小助手桔子
进“技术群”