AWS re:Invent

FSI407

Implement a smart contract with Amazon Managed Blockchain

Evren Sen

Solutions Architect
Amazon Web Services

aws

Agenda

Trade application on Hyperledger Fabric with Off-Chain Interaction

- Rehearse fabric application design patterns

- Connect fabric applications to external business systems

Related breakouts

BLC201 - Building your first blockchain application with Managed Blockchain

BLC204 - Building enterprise solutions with blockchain technology

BLC206 - Smart supply chain management systems based on blockchain, IoT & AI

BLC207 - Trade finance goes high tech using Amazon Managed Blockchain

Why and what are we building?

AWS re:Invent

aws

Why decentralized Ledgers?

1 LEDGERS WITH CENTRALIZED TRUST

2 TRANSACTIONS WITH DECENTRALIZED TRUST

Trade Finance Lifecycle

Trading Entities
Agree

↓

Letter of Credits
Regulatory Licenses
Trade Asset Transfer
Reconciliations

↓

Trade Confirmation
Payment Settlement

Financial institutions
Peer-to-peer payments,
Settlements, Trade Operations

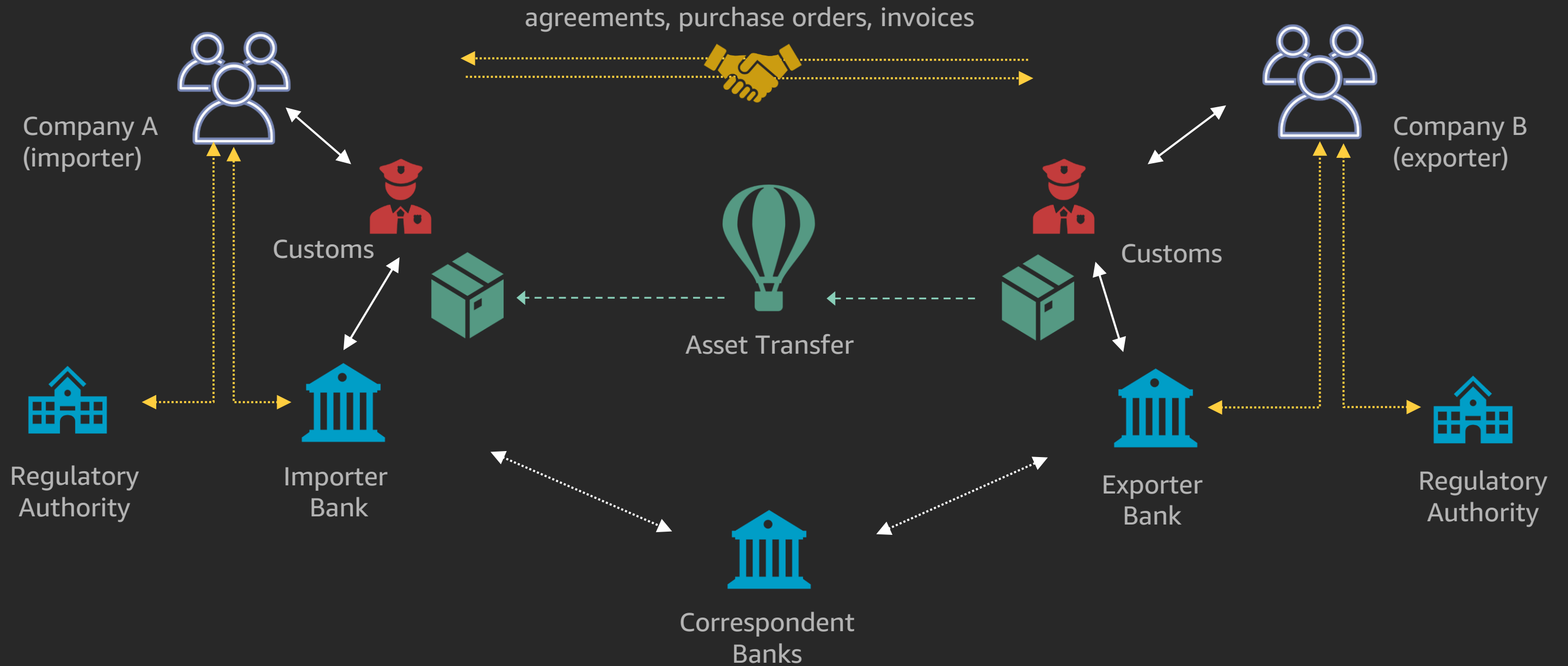**Mortgage lenders**
Process syndicated loans

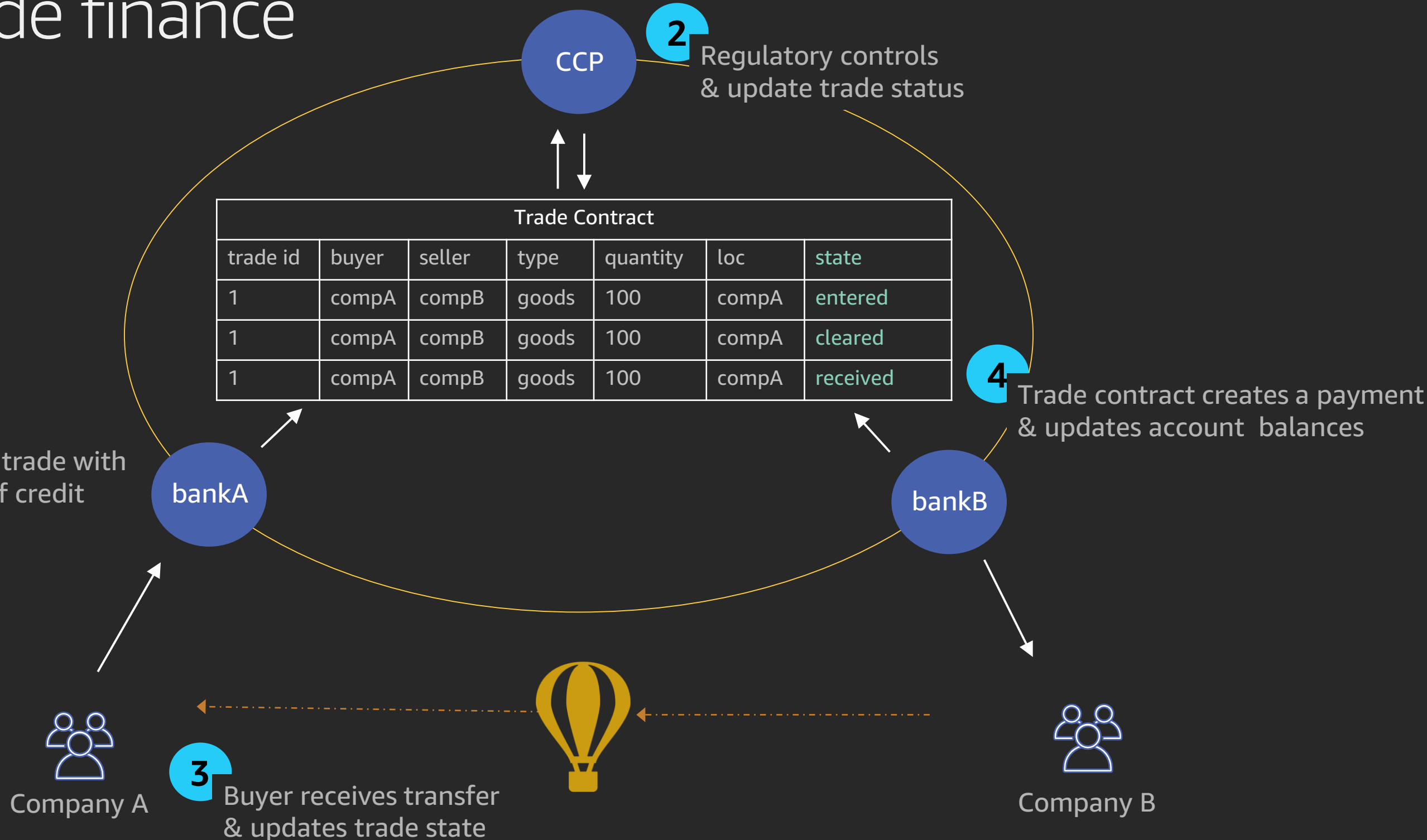**Supply Chain**
Transact with suppliers
and distributers

Retail
Streamline customer
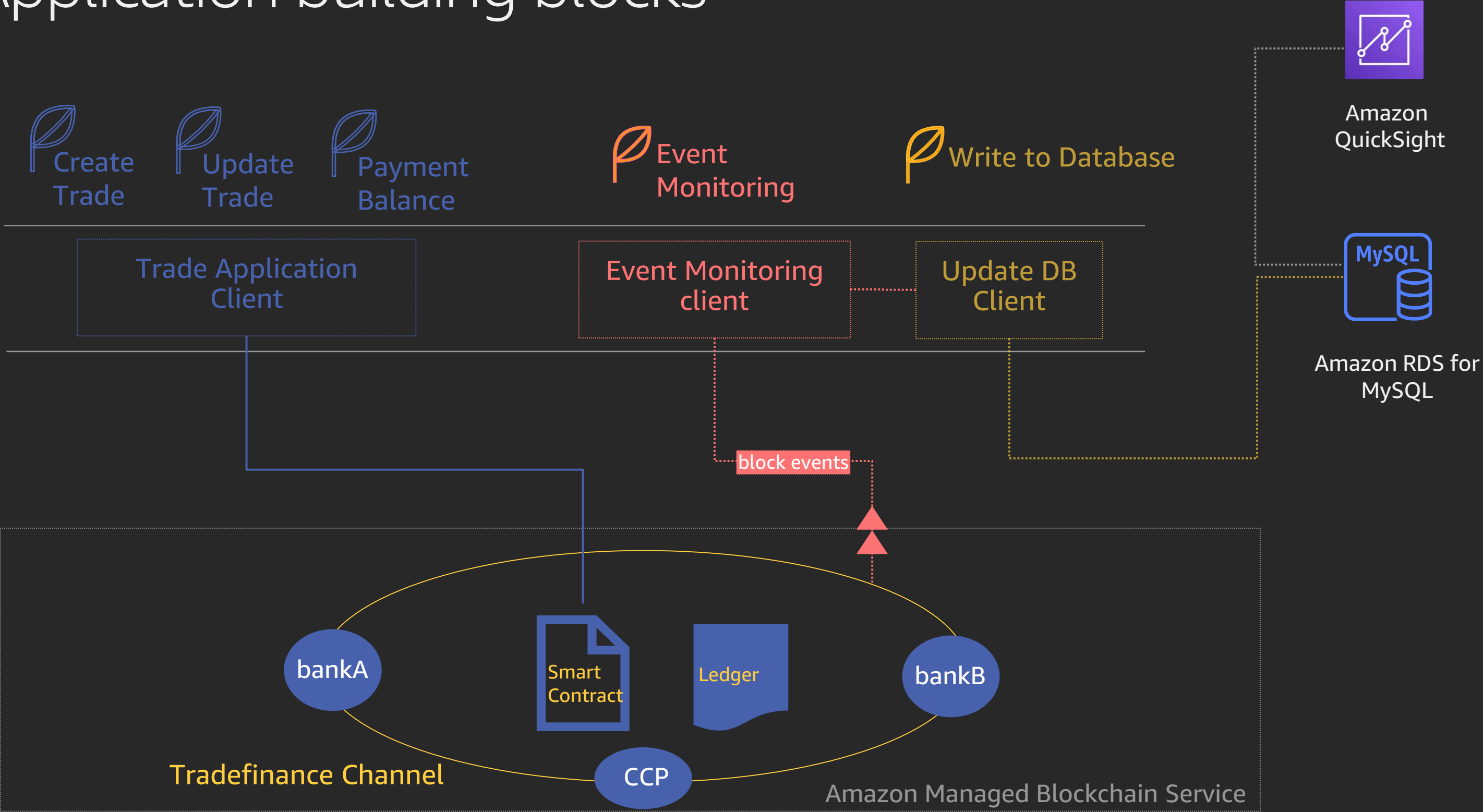rewards

Trade finance flow

agreements, purchase orders, invoices

Company A (importer)

Company B (exporter)

Customs

Customs

Asset Transfer

Regulatory Authority

Importer Bank

Exporter Bank

Regulatory Authority

Correspondent Banks

Trade finance

CCP

2 Regulatory controls & update trade status

Trade Contract

trade id	buyer	seller	type	quantity	loc	state
1	compA	compB	goods	100	compA	entered
1	compA	compB	goods	100	compA	cleared
1	compA	compB	goods	100	compA	received

1 Initiate trade with letter of credit

bankA

bankB

4 Trade contract creates a payment & updates account balances

Company A

3 Buyer receives transfer & updates trade state

Company B

Application building blocks

Create Trade   Update Trade   Payment Balance

Event Monitoring

Write to Database

Amazon QuickSight

Trade Application Client

Event Monitoring client

Update DB Client

MySQL

Amazon RDS for MySQL

block events

bankA

Smart Contract

Ledger

bankB

CCP
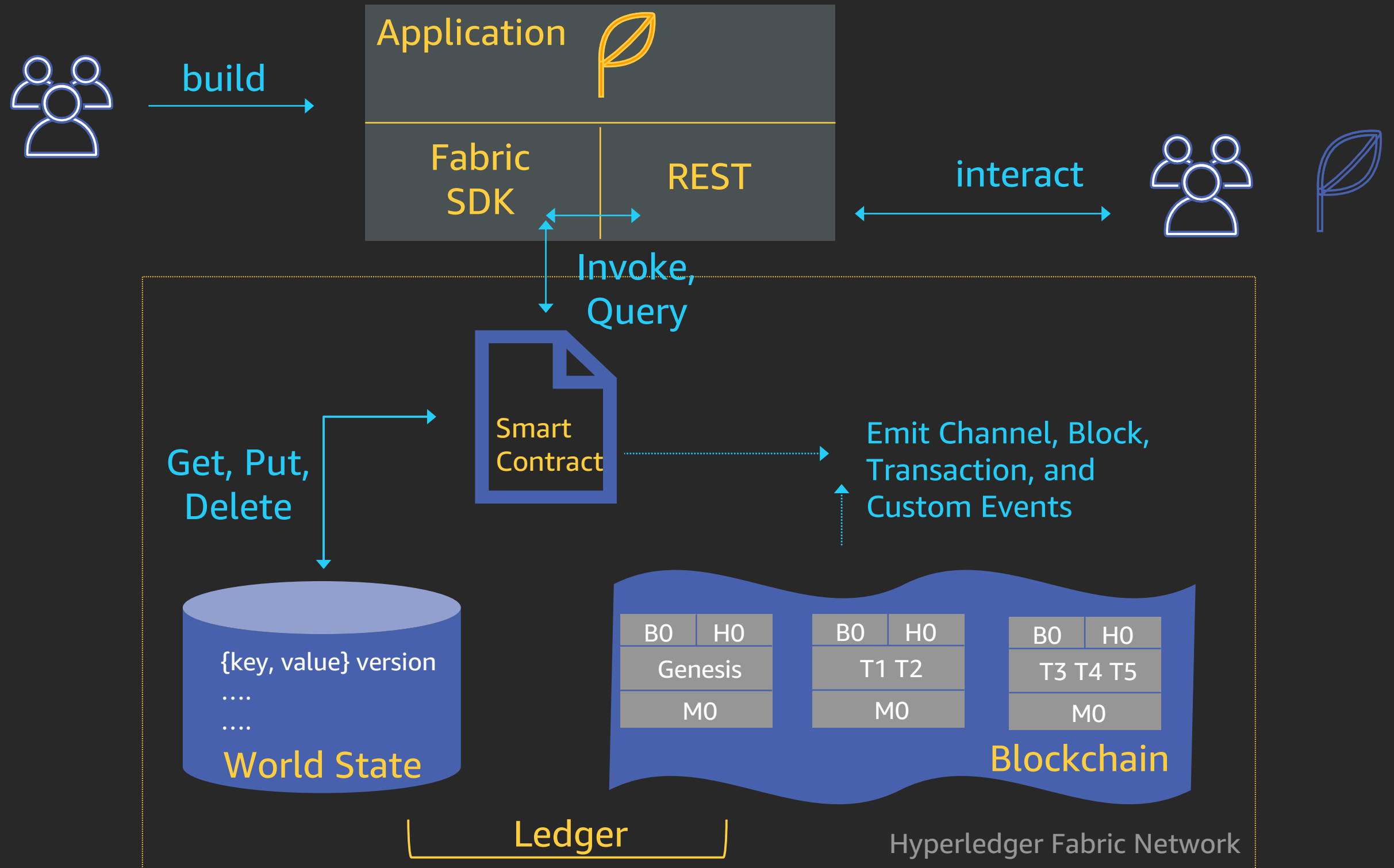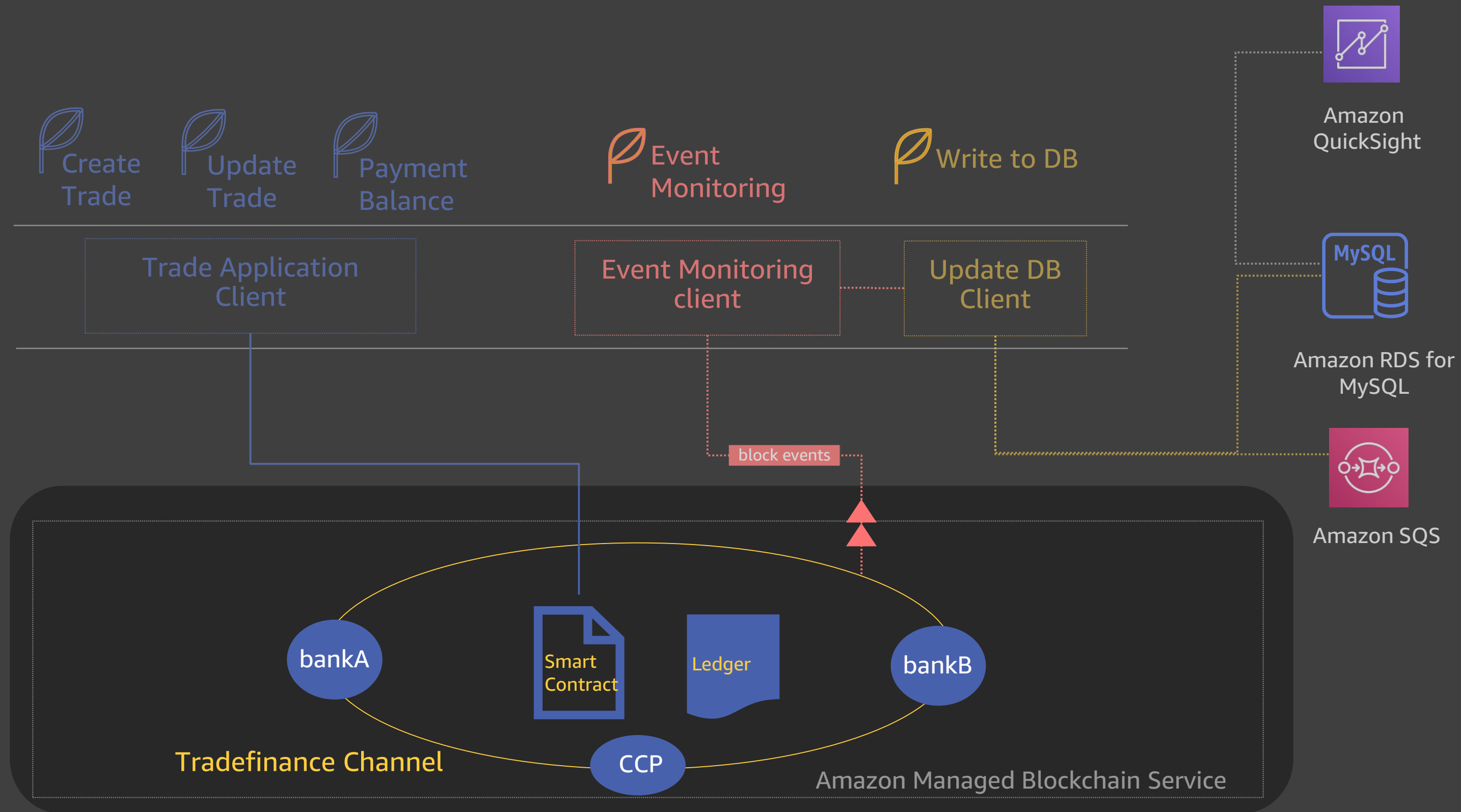
Tradefinance Channel

Amazon Managed Blockchain Service

Demo

aws

Application design

Application building blocks

Application 🍃

build →

Fabric SDK | REST

interact ↔

Invoke, Query

Smart Contract

Get, Put, Delete

Emit Channel, Block, Transaction, and Custom Events

{key, value} version
....
....

World State

B0	H0
Genesis	
M0	

B0	H0
T1 T2	
M0	

B0	H0
T3 T4 T5	
M0	

Blockchain

Ledger

Hyperledger Fabric Network

Create
Trade

Update
Trade

Payment
Balance

Event
Monitoring

Write to DB

Amazon
QuickSight

Trade Application
Client

Event Monitoring
client

Update DB
Client

MySQL

Amazon RDS for
MySQL

block events

Amazon SQS

bankA

Smart
Contract
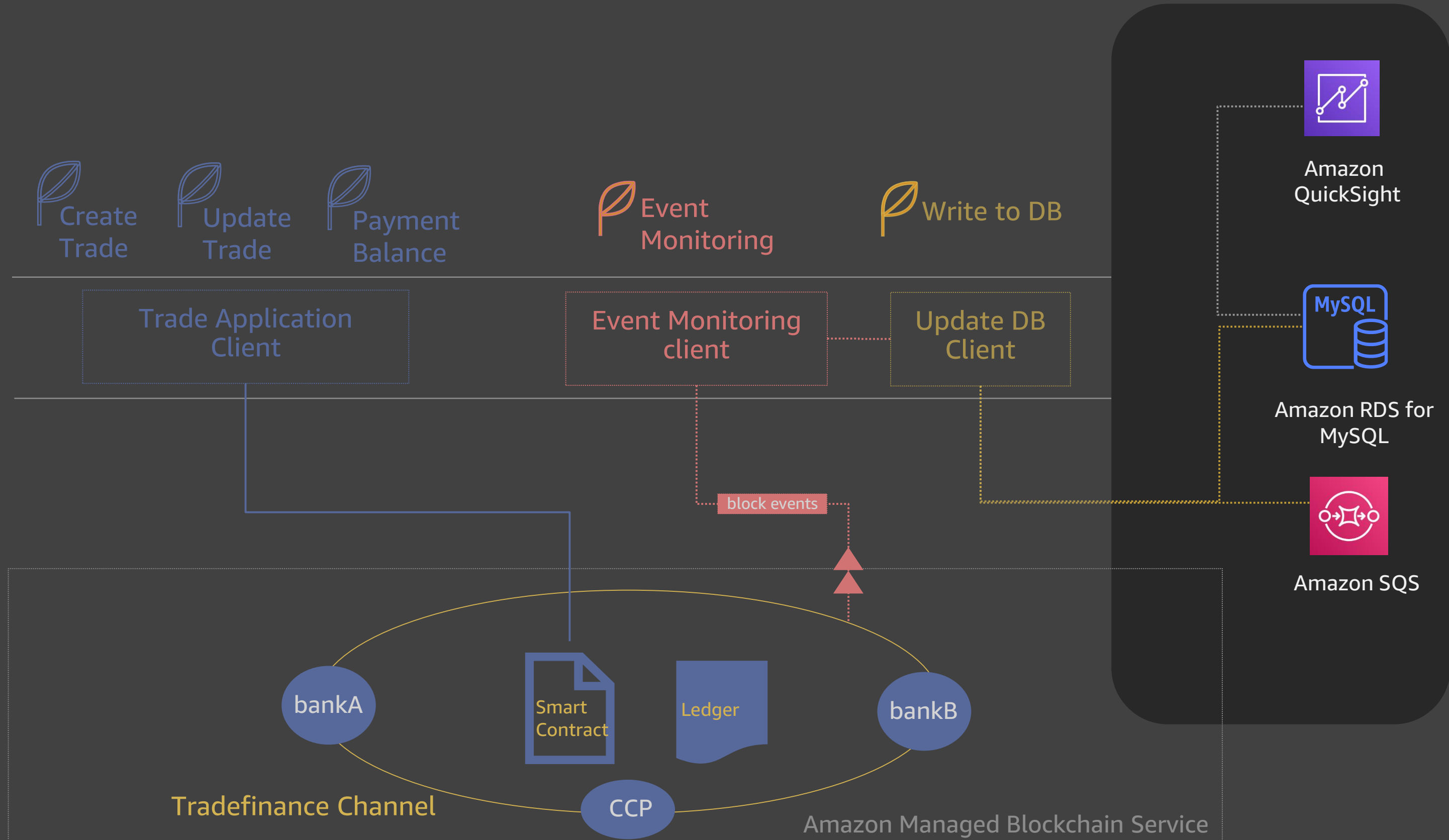
Ledger

bankB

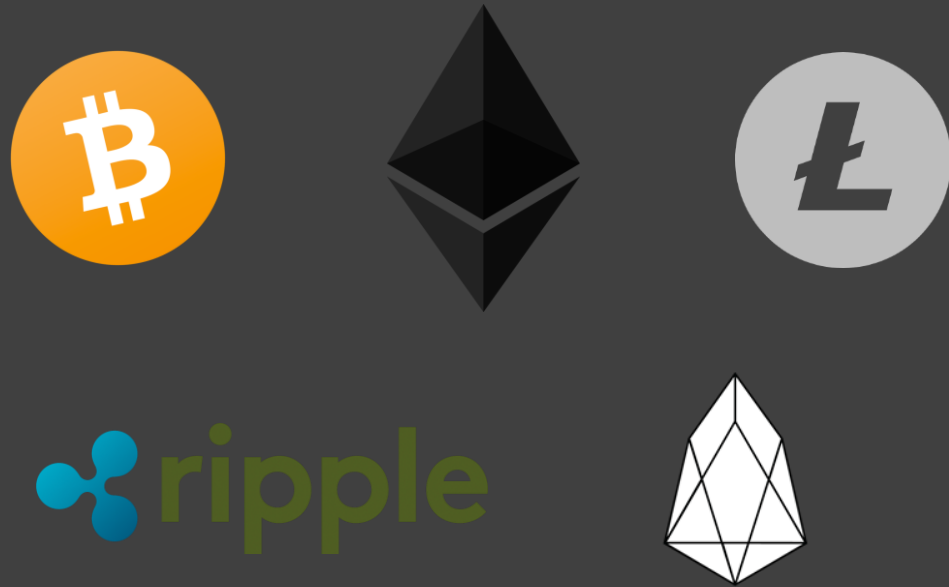Tradefinance Channel

CCP

Amazon Managed Blockchain Service

Public and permissioned networks

Public networks allow anyone to participate in the network. Members are incentivized to maintain the network.

Permissioned networks limit the members to known entities.

Hyperledger Fabric

- Create **permissioned networks** with channels to limit the transactions on the ledger each member can see

- **Chaincode (smart contracts)** written in Go, Java, and Node.js and are executed in Docker containers

- **Validation policy** for executing chaincode is configurable

- Does not require a native cryptocurrency for chaincode execution

Hyperledger mechanics
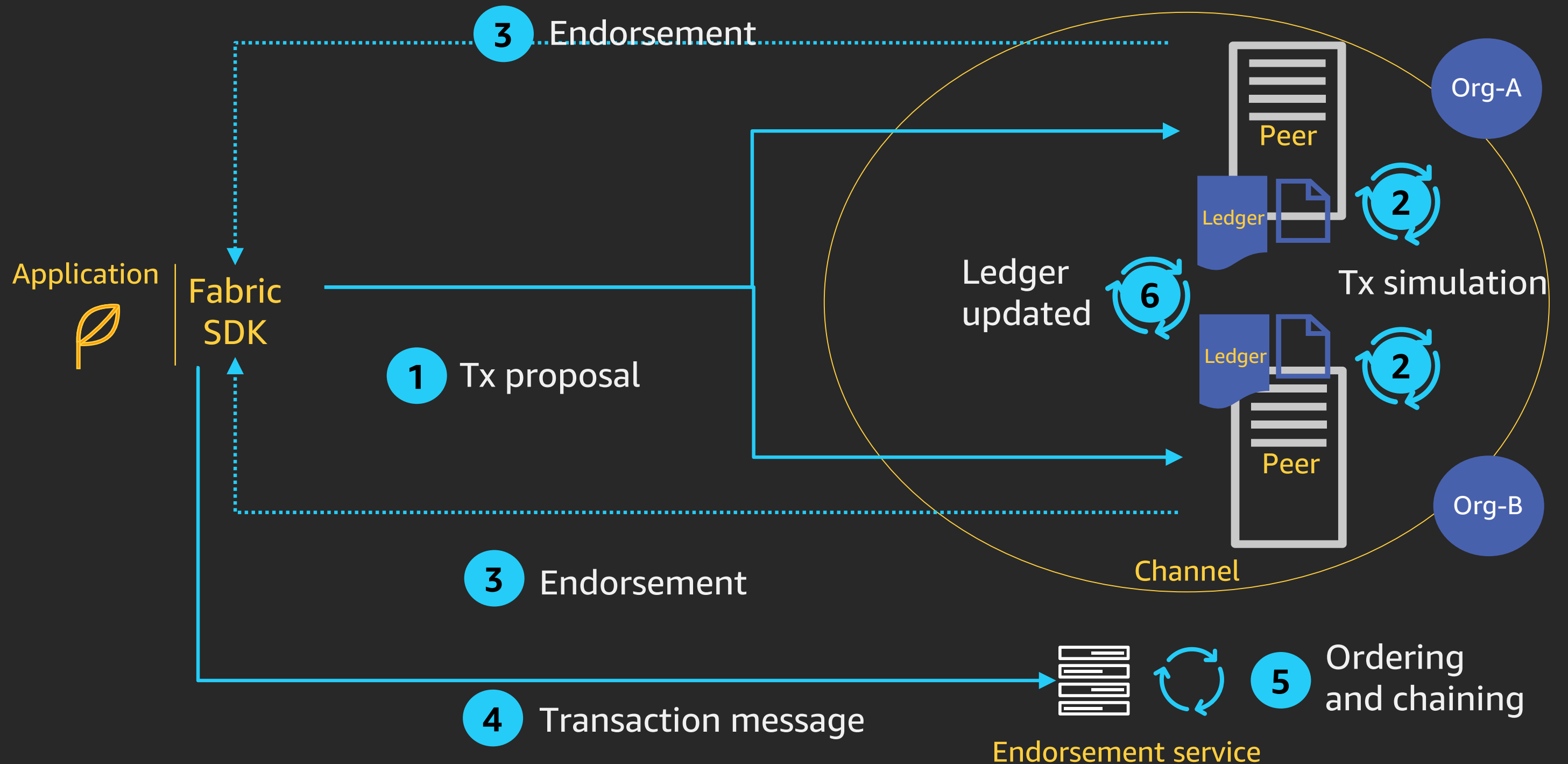
Hyperledger mechanics

Application

REST | Fabric SDK

Select identity

Connect to network channel

Form transaction

Submit {invoke} transaction

Query world-state

Process any response

Smart contract

```
contract {
    create (….)
    update (….)
    query (….)
    …
}
```

Thank you!

Please complete the session survey in the mobile app.