Building Blockchains With Hyperledger Fabric

Overview

A blockchain is an immutable, verifiable and distributed ledger of transactions

Hyperledger is an umbrella of open-source blockchain projects

Hyperledger Fabric is the most mature and stable of the Hyperledger frameworks

AWS Blockchain Template allows us to quickly spin up a Fabric network on AWS

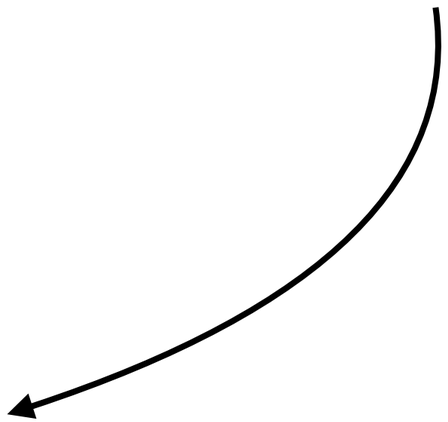
Chaincode (Smart Contract) enables implementation of rules for Fabric transactions

Quick Overview of Blockchain

Blockchain

An encrypted database of agreements

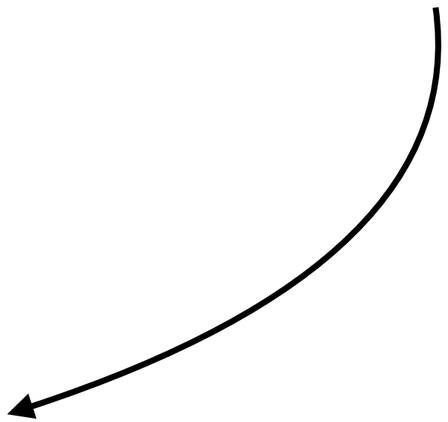
Once a deal has been recorded neither party can go back and rewrite terms



Blockchain

An encrypted database of agreements

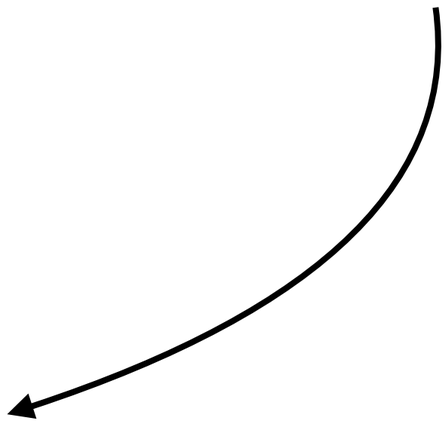
Recorded in the form of verified transactions



Blockchain

An encrypted database of agreements

Multiple transactions are stored in the form of blocks

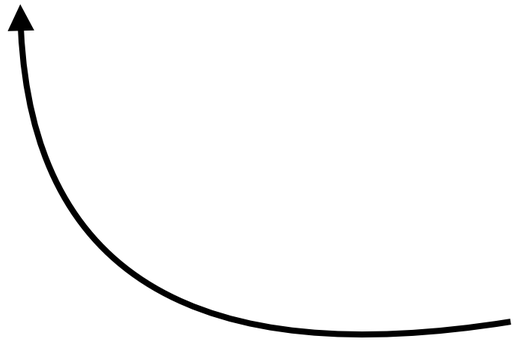


Blockchain

An encrypted database of agreements

Blockchain

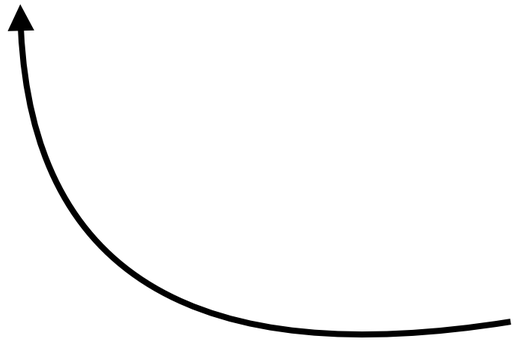
An encrypted database of agreements



Serves as a bookkeeping platform or a ledger

Blockchain

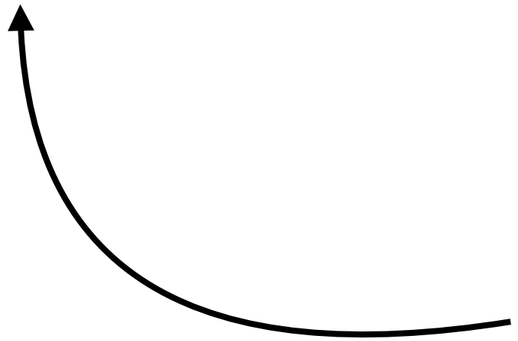
An encrypted database of agreements



Incorruptible, enforces transparency and bypasses censorship

Blockchain

An encrypted database of agreements



Open for the world to view or can be restricted to specific users

Blockchain

An encrypted database of agreements

What Agreements?

Financial transactions

Real-estate sales

Supply chain management

Voter records

Any contractual agreement

Blockchain

A blockchain is a growing list of records (called blocks) which are linked cryptographically

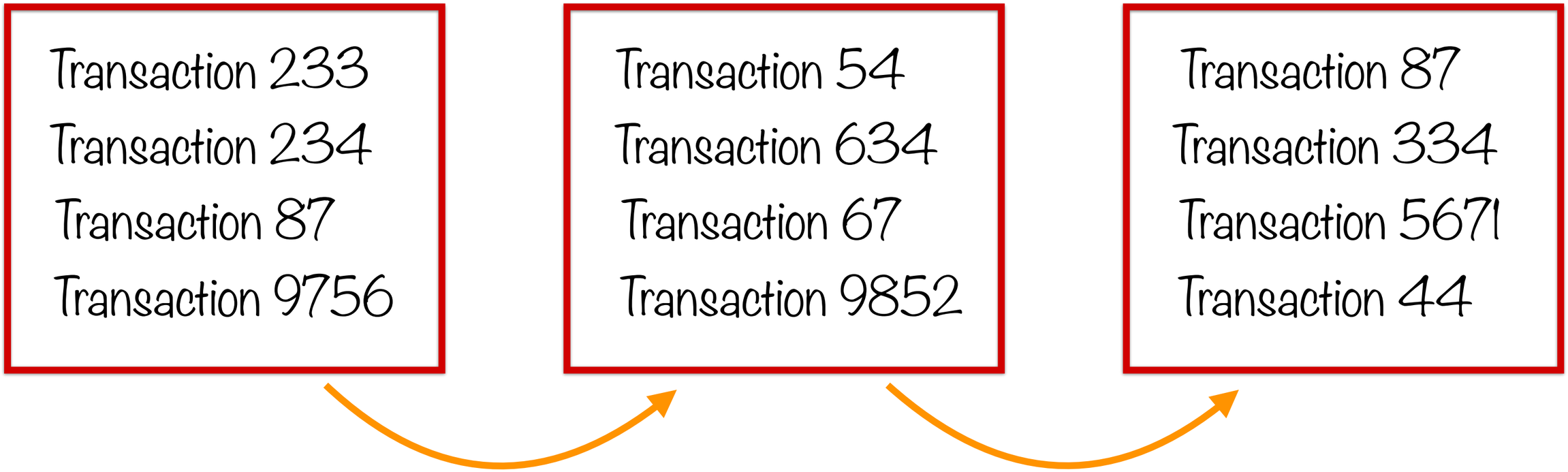
Block

Transaction 233
Transaction 234
Transaction 87
Transaction 9756

Transaction 54
Transaction 634
Transaction 67
Transaction 9852

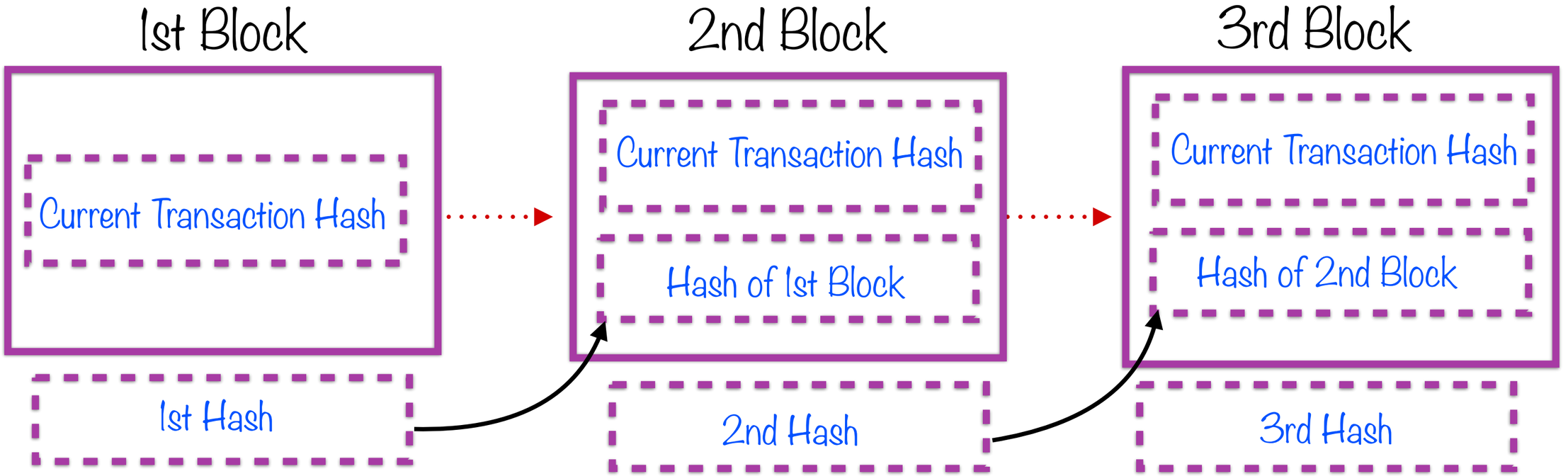
Transaction 87
Transaction 334
Transaction 5671
Transaction 44

Blocks Linked Together Cryptographically



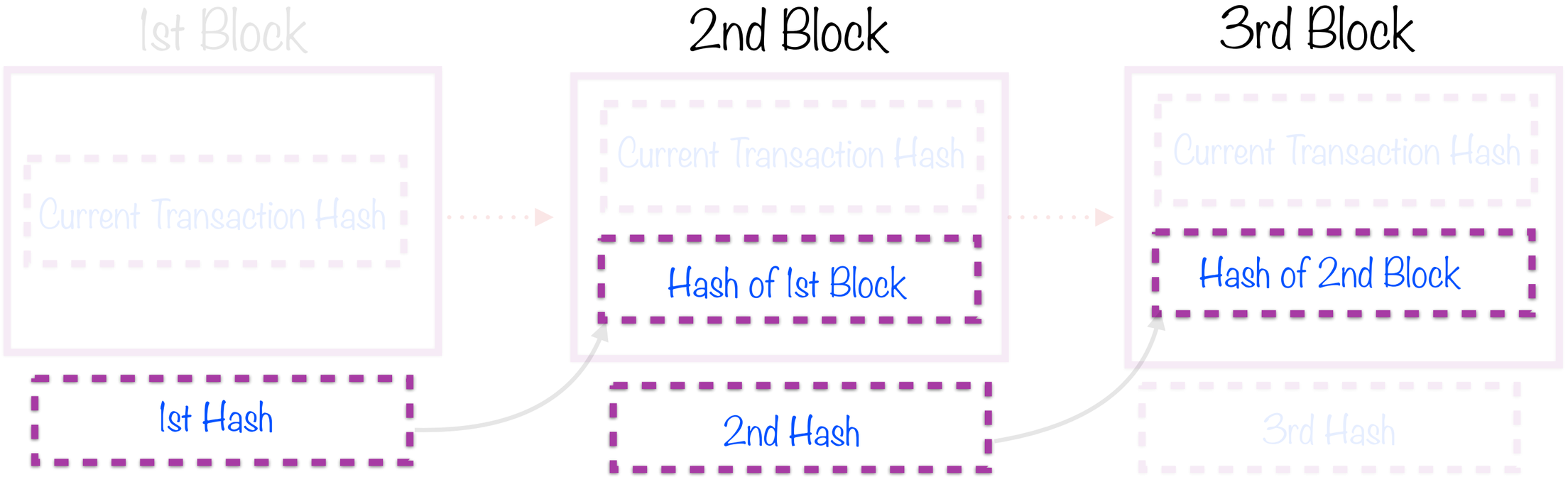
Every block is linked to the next block containing the next set of transactions

Chain of Blocks



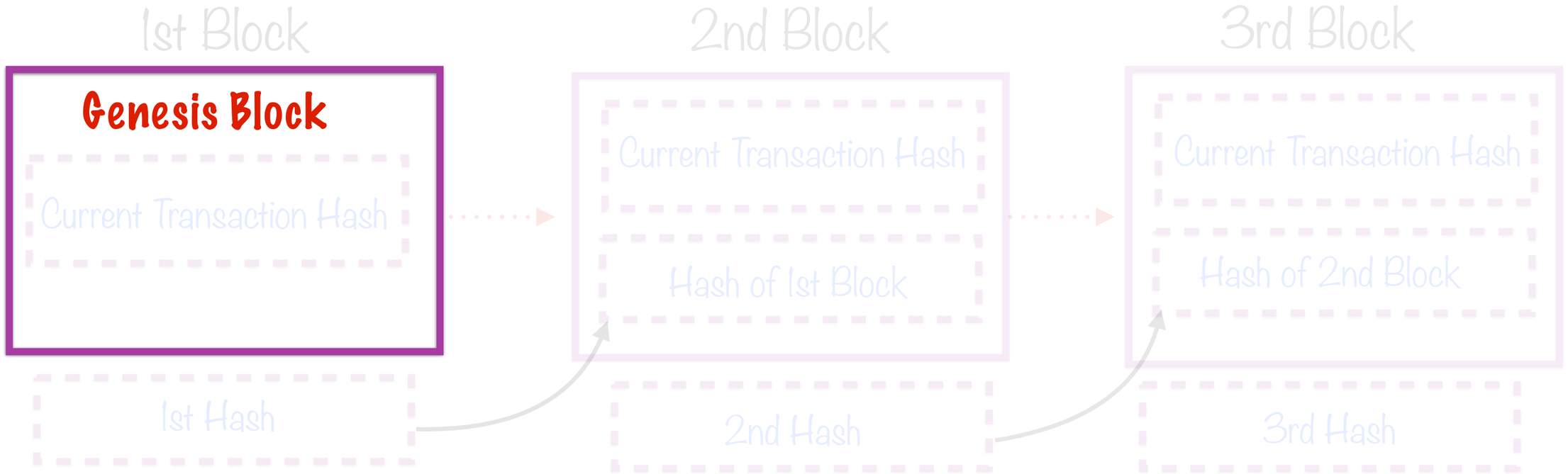
Each block contains hash of the preceding one in chain

Chain of Blocks



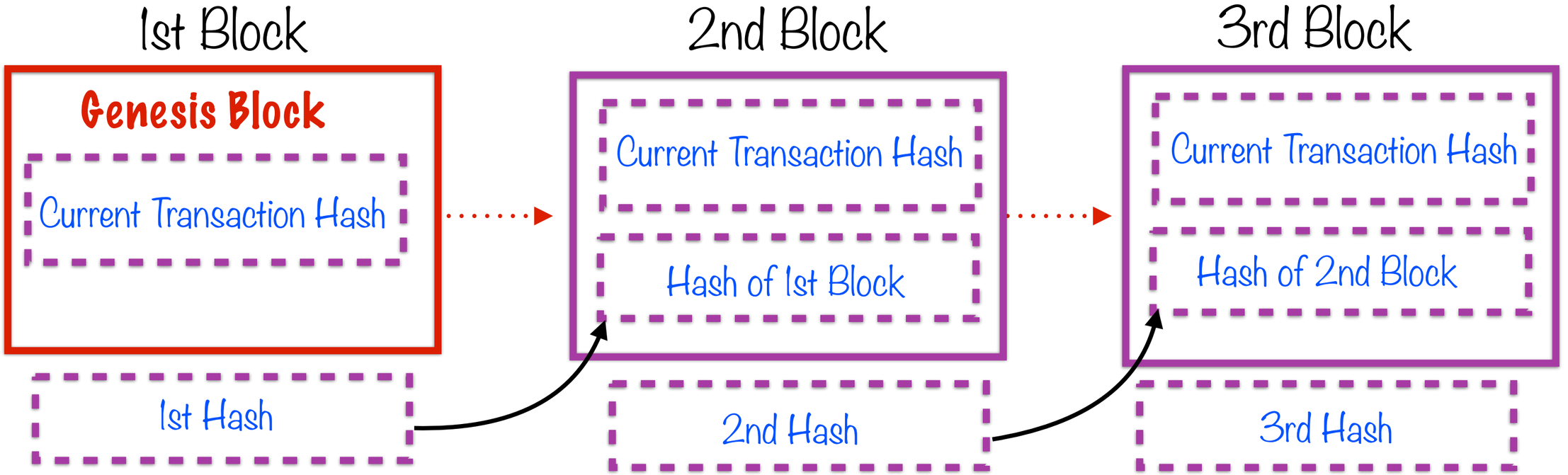
Each block contains hash of the preceding one in chain

Chain of Blocks



This is not true for the first block - called the Genesis Block

Chain of Blocks



Genesis block does not contain previous hash value

Cryptographic Link

Transaction data in that block can not be altered after-the-fact

- except by altering all subsequent blocks
- which is complicated and needs consensus

Ledger is distributed (not centralized)

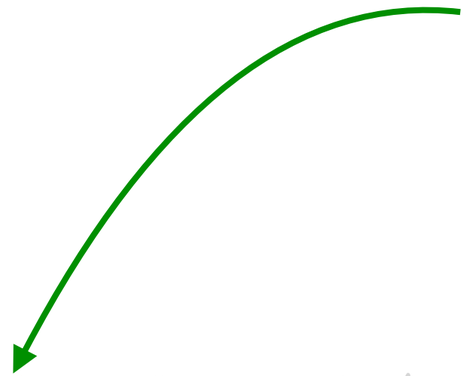
Blockchain

A blockchain is a growing list of records (called blocks) which are linked cryptographically

Blockchain

A blockchain is a growing list of records (called blocks) which are linked cryptographically

The list is not stored in its entirety on any one node of that peer-to-peer network



"Open, distributed ledger"

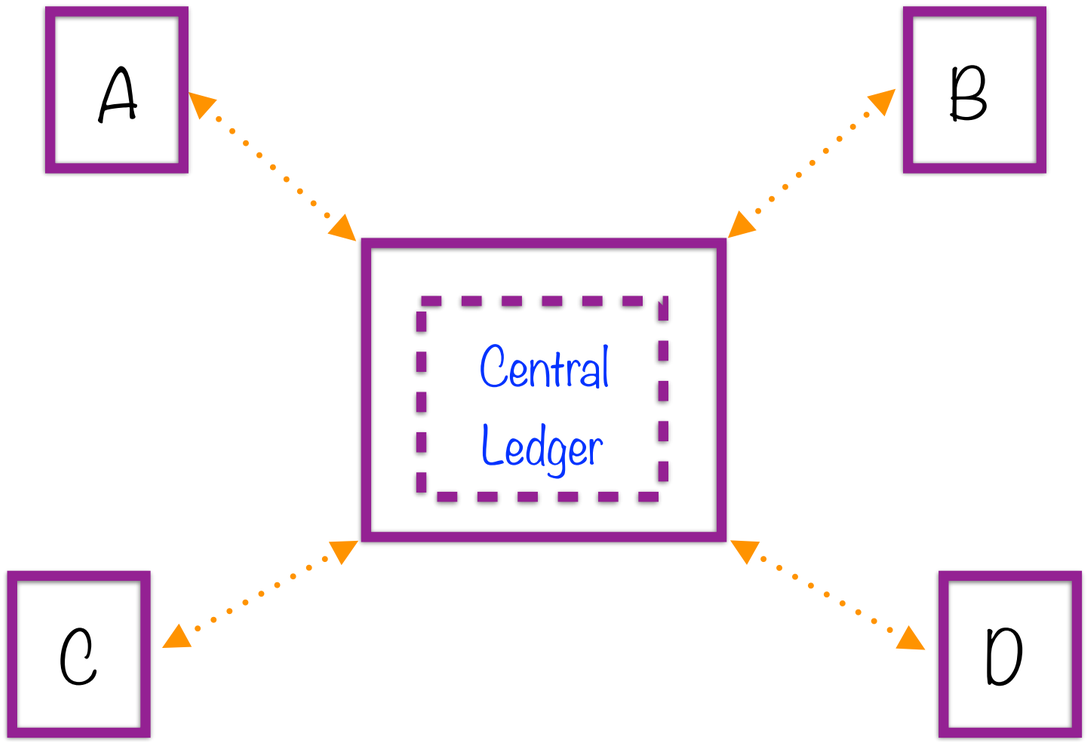
Blockchain

A blockchain is a growing list of records (called blocks) which are linked cryptographically

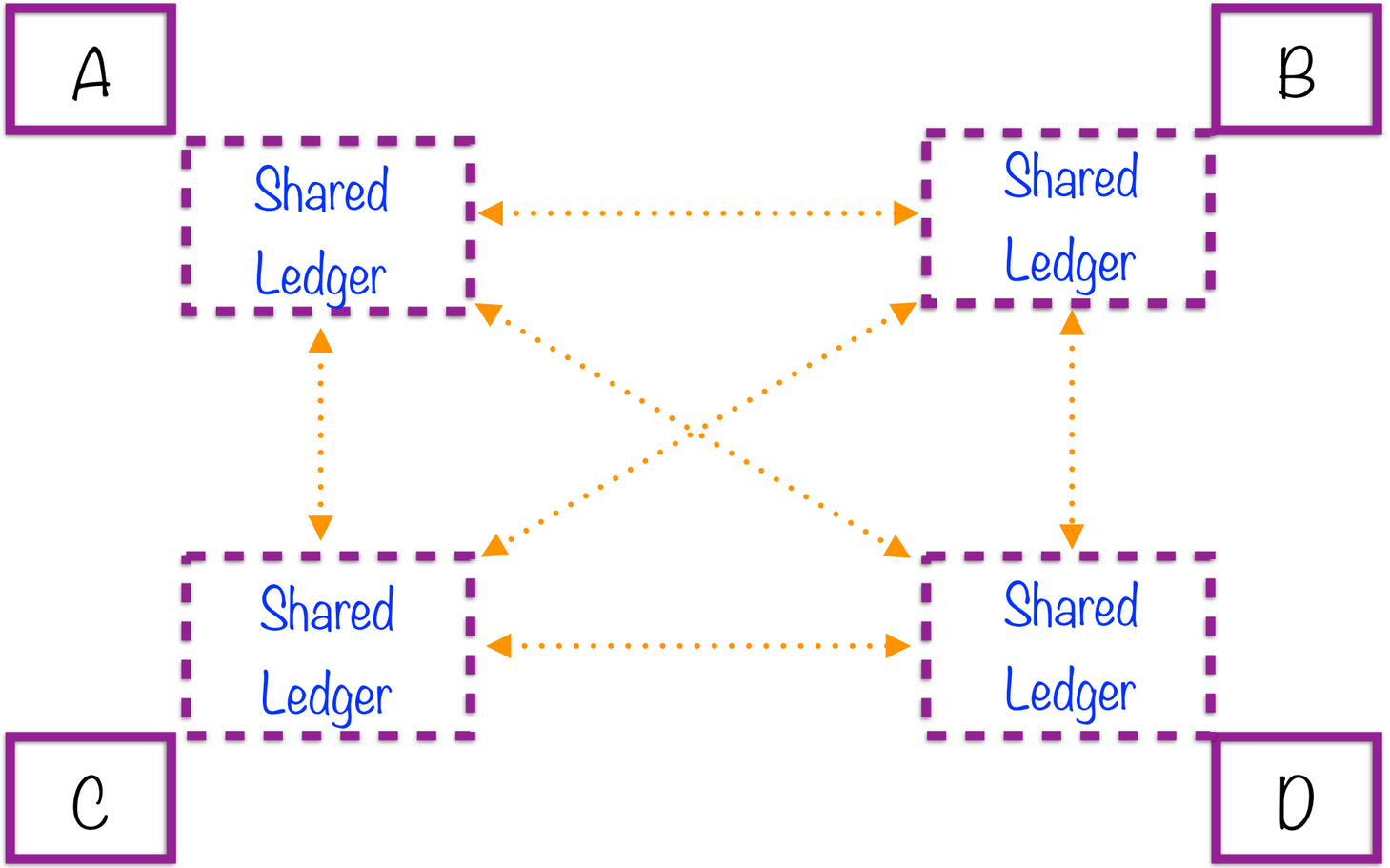
Blockchain

An open, distributed ledger that can record transactions in a verifiable and permanent manner

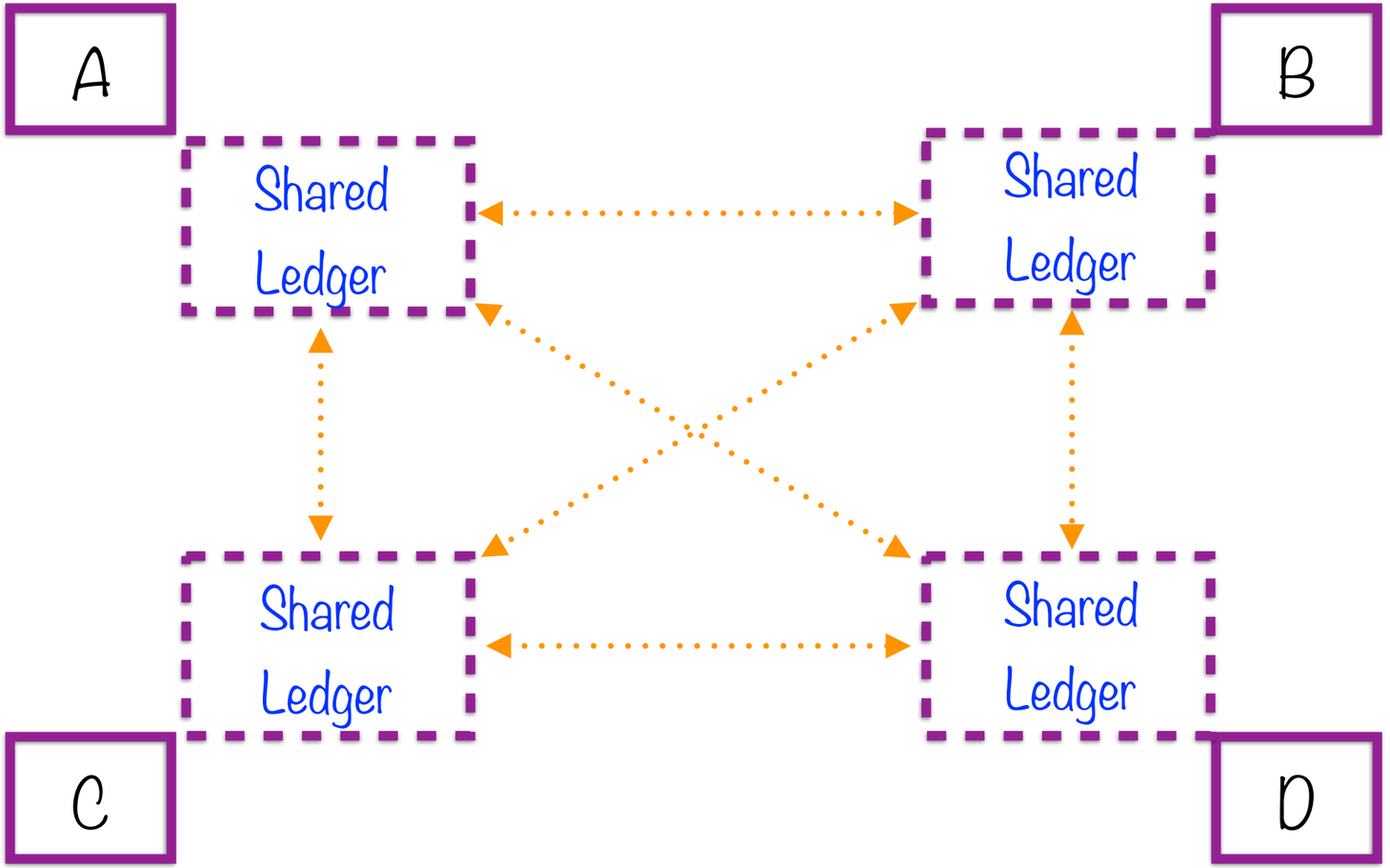
Centralized Ledger



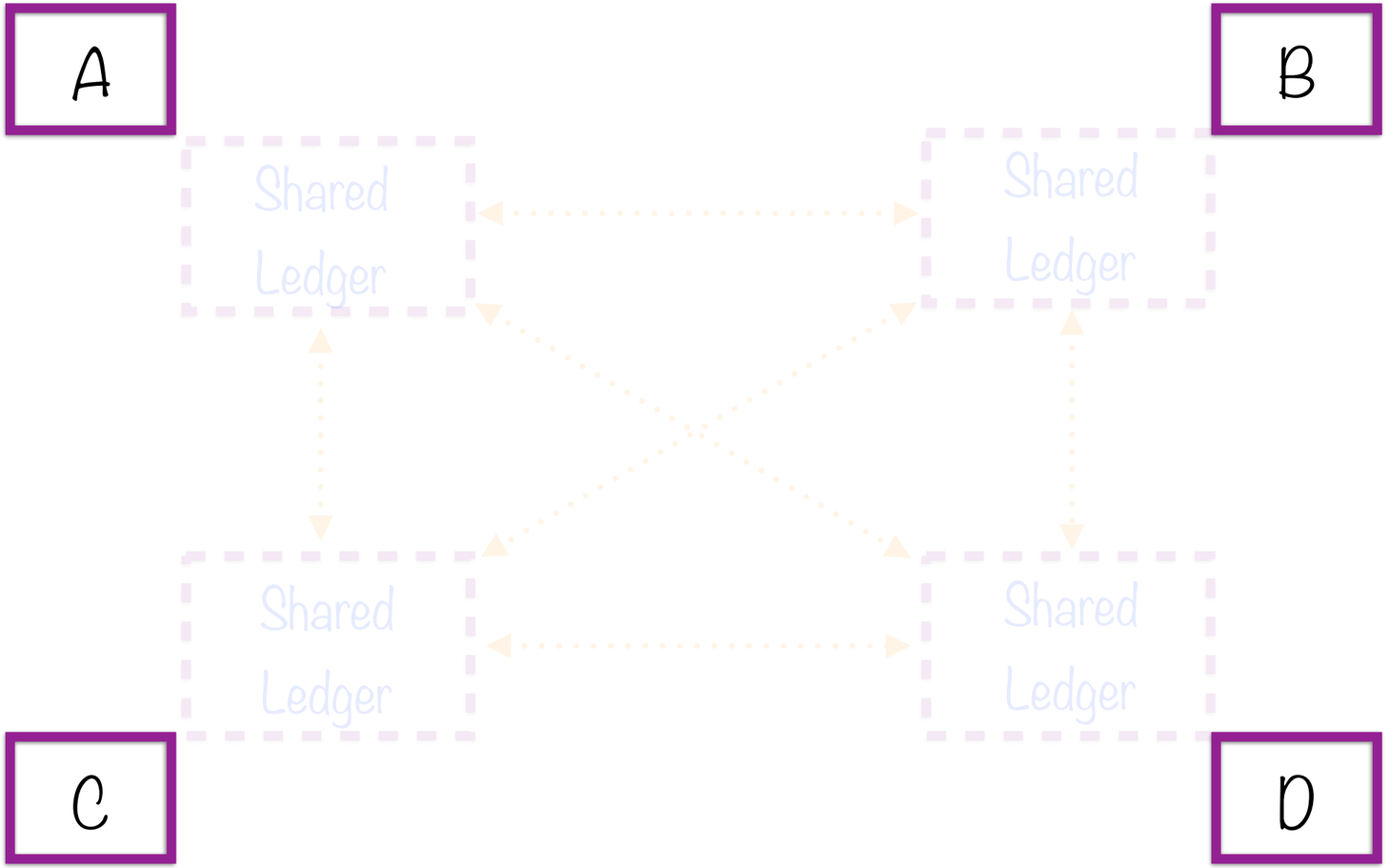
Distributed Ledger



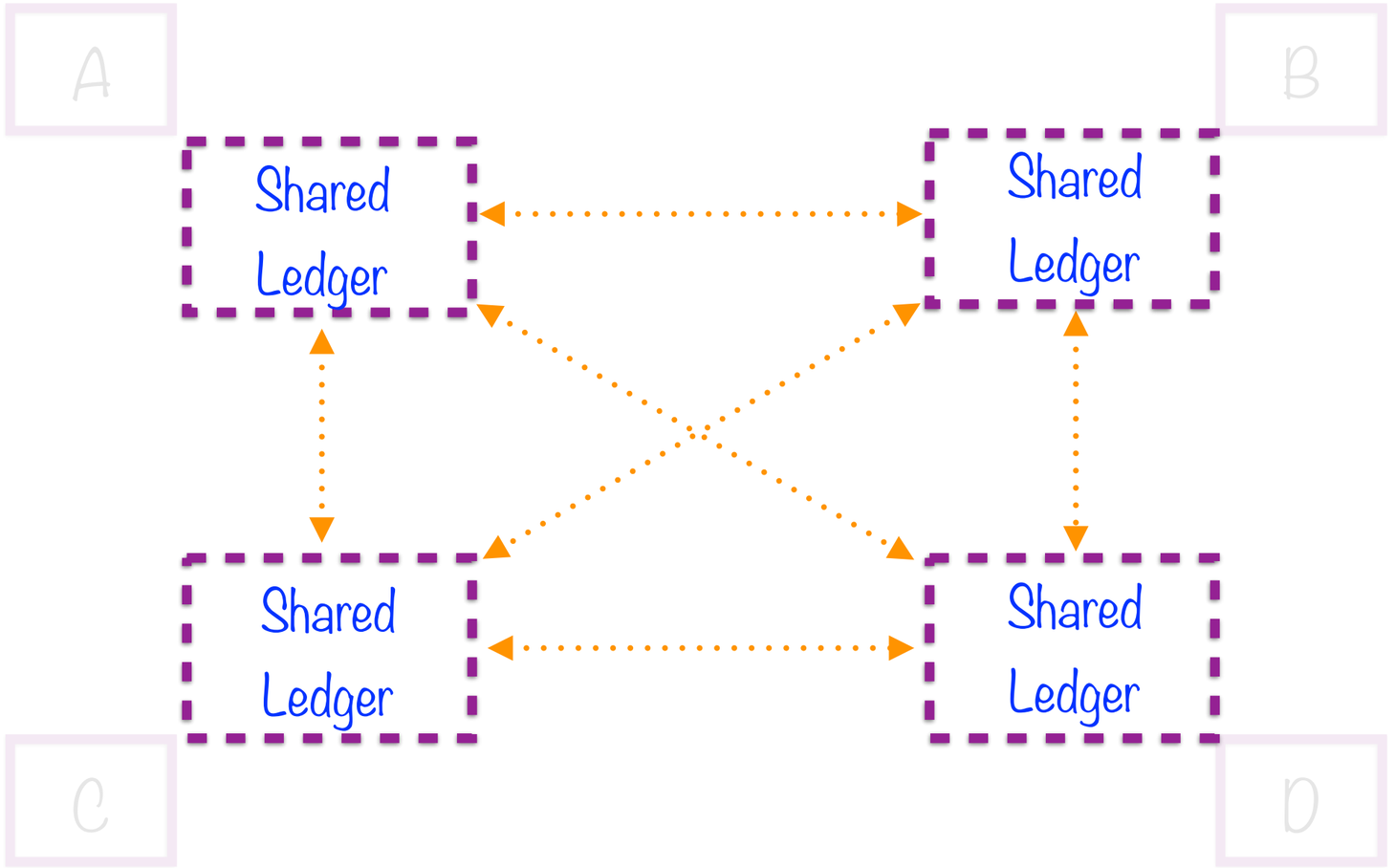
Blockchain Network



Nodes in a Blockchain Network



Blockchain


A
B
Shared
Ledger
Shared
Ledger
Shared
Ledger
Shared
Ledger
C
D

Public and Private Blockchain Networks

Public Network	Private Network
Truly open membership - anyone can join	Restricted entry - selection criteria
Lots of nodes	Fewer nodes
Slower to converge on consensus*	Faster convergence on consensus*
Higher probability of fraud	Lower probability of fraud

*Will be explained in just a bit

Hyperledger

Ethereum

Open-source, public blockchain-based platform that supports smart contracts

Smart Contract

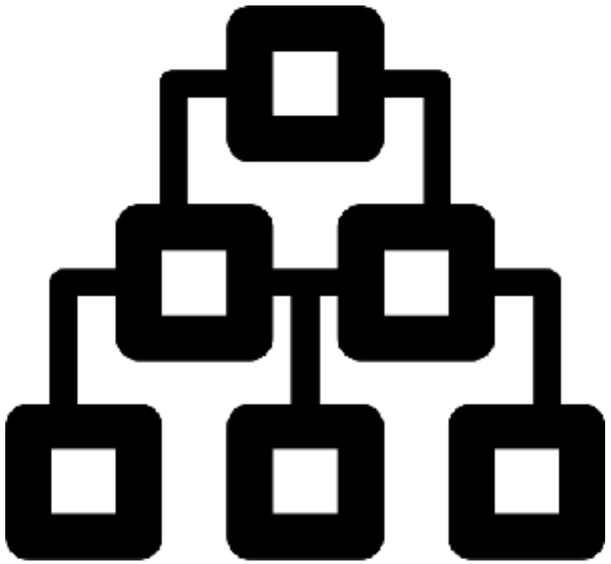
Mechanism that allows common contractual clauses to be specified, verified or enforced even in the absence of trust between contracting parties and in the absence of a third party

Ether

Cryptocurrency whose blockchain is generated by Ethereum and is used to compensate nodes on Ethereum for their participation

Ethereum Virtual Machine (EVM)

Execution environment that all Ethereum nodes run



Nodes

Each node runs the Ethereum Virtual Machine (EVM)

EVM is runtime environment responsible for

- Executing contract code
- Calculating transaction complexity (gas consumption)
- Verifying transactions

Concerns Around Ethereum

Scaling

Each peer node executes all transactions

Resource Usage

Proof-of-work algorithm incurs significant wasted effort

Confidentiality

Each transaction is broadcast to all peers in network

Hyperledger is designed to mitigate these concerns

Hyperledger

Umbrella project of open source blockchains; started in December 2015 by the Linux Foundation

Umbrella of Frameworks

Hyperledger Fabric

Hyperledger Sawtooth

Hyperledger Iroha

Hyperledger Burro

Hyperledger Indy

Hyperledger Burrow [edit]

Burrow[16] is a blockchain client including a built-to-specification Ethereum Virtual Machine. Contributed by Monax[17] and sponsored by Monax and Intel.[18]

Hyperledger Fabric [edit]

Hyperledger Fabric is a permissioned blockchain infrastructure, originally contributed by IBM[19] and Digital Asset, providing a modular architecture with a delineation of roles between the nodes in the infrastructure, execution of Smart Contracts (called "chaincode" in Fabric) and configurable consensus and membership services. A Fabric Network comprises "Peer nodes", which execute chaincode, access ledger data, endorse transactions and interface with applications. "Orderer nodes" which ensure the consistency of the blockchain and deliver the endorsed transactions to the peers of the network, and MSP services, generally implemented as a Certificate Authority, managing X.509 certificates which are used to authenticate member identity and roles.[20]

Fabric is primarily aimed at integration projects, in which a Distributed Ledger Technology (DLT) is required, offering no user facing services other than an SDK for Node.js, Java and Go.

Fabric supports chaincode in Go and JavaScript (via Hyperledger Composer, or natively since v1.1) out-of-the-box, and other languages such as Java by installing appropriate modules. It is therefore potentially more flexible than competitors that only support a closed Smart Contract language.

Hyperledger Iroha [edit]

Based on Hyperledger Fabric, with a focus on mobile applications. Contributed by Soramitsu.[21]

Hyperledger Sawtooth [edit]

Originally contributed by Intel, Sawtooth includes a dynamic consensus feature enabling hot swapping consensus algorithms in a running network. Among the consensus options is a novel consensus protocol known as "Proof of Elapsed Time," a lottery-design consensus protocol that optionally builds on trusted execution environments provided by Intel's Software Guard Extensions (SGX).[22] Sawtooth supports Ethereum smart contracts via "seth" (a Sawtooth transaction processor integrating the Hyperledger Burrow EVM).[23] In addition to Solidity support, Sawtooth includes SDKs for Python, Go, Javascript, Rust, Java, and C++ [24]

Hyperledger Indy [edit]

Indy[25] is a Hyperledger project for supporting independent identity on distributed ledgers. It provides tools, libraries, and reusable components for providing digital identities rooted on blockchains or other distributed ledgers. Contributed by the Sovrin Foundation.[26]

Umbrella of Tools

Hyperledger Cello

Hyperledger Composer

Hyperledger Caliper

Hyperledger Explorer

Hyperledger Quilt

Hyperledger Tools [edit]

Hyperledger Caliper [edit]

Hyperledger Caliper is a blockchain benchmark tool and one of the Hyperledger projects hosted by The Linux Foundation. Hyperledger Caliper allows users to measure the performance of a specific blockchain implementation with a set of predefined use cases. Hyperledger Caliper will produce reports containing a number of performance indicators, such as TPS (Transactions Per Second), transaction latency, resource utilisation etc. The intent is for Caliper results to be used by other Hyperledger projects as they build out their frameworks, and as a reference in supporting the choice of a blockchain implementation suitable for a user's specific needs. Hyperledger Caliper was initially contributed by developers from Huawei, Hyperchain, Oracle, Bitwise, Soramitsu, IBM and the Budapest University of Technology and Economics.[27]

Hyperledger Cello [edit]

Hyperledger Cello is a blockchain module toolkit and one of the Hyperledger projects hosted by The Linux Foundation. Hyperledger Cello aims to bring the on-demand "as-a-service" deployment model to the blockchain ecosystem to reduce the effort required for creating, managing and terminating blockchains. It provides a multi-tenant chain service efficiently and automatically on top of various infrastructures, e.g., baremetal, virtual machine, and more container platforms. Hyperledger Cello was initially contributed by IBM, with sponsors from Soramitsu, Huawei and Intel.[28]

Baohua Yang and Haitao Yue from IBM Research are committed part-time to developing and maintaining the project.

Hyperledger Composer [edit]

Hyperledger Composer is a set of collaboration tools for building blockchain business networks that make it simple and fast for business owners and developers to create smart contracts and blockchain applications to solve business problems. Built with JavaScript, leveraging modern tools including node.js, npm, CLI and popular editors, Composer offers business-centric abstractions as well as sample apps with easy to test devops processes to create robust blockchain solutions that drive alignment across business requirements with technical development.[29]

Blockchain package management tooling contributed by IBM. Composer is a user-facing rapid prototyping tooling, running on top of Hyperledger Fabric, which allows the easy management of Assets (data stored on the blockchain), Participants (identity management, or member services) and Transactions (Chaincode, a.k.a Smart Contracts, which operate on Assets on the behalf of a Participant). The resulting application can be exported as a package (a BNA file) which may be executed on a Hyperledger Fabric instance, with the support of a Node.js application (based on the Loopback application framework) and provide a REST interface to external applications.

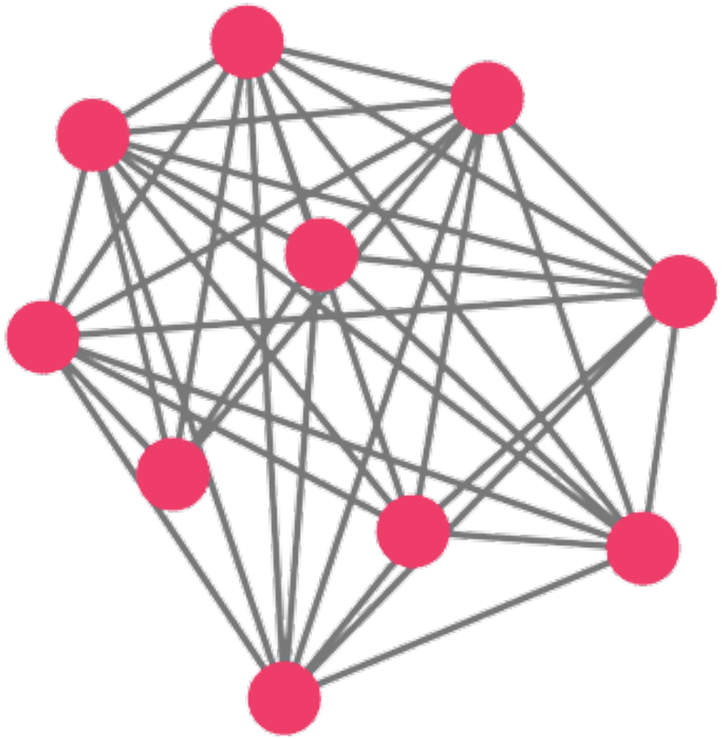
Composer provides a GUI user interface "Playground" for the creation of applications, and therefore represents an excellent starting point for Proof of Concept work.

Hyperledger Explorer [edit]

Hyperledger Explorer is a blockchain module and one of the Hyperledger projects hosted by The Linux Foundation. Designed to create a user-friendly Web application, Hyperledger Explorer can view, invoke, deploy or query blocks, transactions and associated data, network information (name, status, list of nodes), chain codes and transaction families, as well as any other relevant information stored in the ledger. Hyperledger Explorer was initially contributed by IBM, Intel and DTCC.[30]

Hyperledger Quilt [edit]

Hyperledger Quilt is a business blockchain tool and one of the Hyperledger projects hosted by The Linux Foundation. Hyperledger Quilt offers interoperability between ledger systems by implementing the Interledger protocol (also known as ILP), which is primarily a payments protocol and is designed to transfer value across distributed ledgers and non-distributed ledgers. The Interledger protocol provides atomic swaps between ledgers (even non-blockchain or distributed ledgers) and a single account namespace for accounts within each ledger. With the addition of Quilt to Hyperledger, The Linux Foundation now hosts both the Java (Quilt) and JavaScript (Interledger.js) Interledger implementations. Hyperledger Quilt was initially contributed by NTT Data and Ripple.[31]



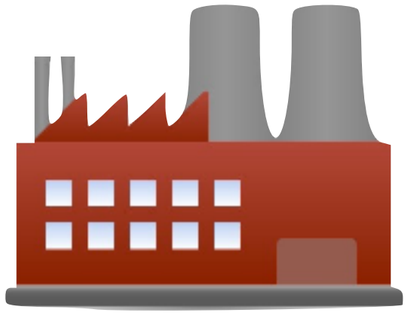
Hyperledger is not a

- Cryptocurrency
- Blockchain

Collaborative Umbrella Project

Framework

Developers



Industries

Communities of software developer and companies meet and coordinate to build blockchain frameworks

Hyperledger Fabric

Permissioned blockchain

Restricted access

Great for private channels

Supports multiple

- consensus algorithms
- membership services

Hyperledger Fabric

No mining

No associated cryptocurrency

High throughput

Hyperledger Fabric

Written in Go

IBM origins

Most active, stable

Wide commercial usage

Hyperledger Sawtooth

Designed for broad, flat networks

Proof of Elapsed Time

Originated at Intel

Hyperledger Composer

Browser-based UI to build apps

Allows easy specification of

- Assets
- Participants
- Transactions
- Blockchain state storage

Hyperledger Fabric vs. Ethereum

Hyperledger Fabric	Ethereum
Permissioned blockchain platform - not all users can validate transactions	Permissionless blockchain platform - all users can participate in validation
Private - entry by invitation only	Public - anyone can join

Hyperledger Fabric vs. Ethereum

Hyperledger Fabric	Ethereum
No associated cryptocurrency	Ether
Tokens exchanged via chaincode	Tokens exchanged via smart contracts

Hyperledger Fabric vs. Ethereum

Hyperledger Fabric

Transaction lifecycle: Execute-order-validate

Transactions can be kept private from some peers

Transactions can be executed in any order, even in parallel*

*Execution is followed by an explicit validation step

Ethereum

Transaction lifecycle: Order-execute (no separate validation phase)

Transactions are broadcast to all members of network

Transaction order must be agreed upon by all participants

No explicit validation step

Hyperledger Fabric vs. Ethereum

Hyperledger Fabric	Ethereum
Smart contracts without trusted third-party	Smart contracts without trusted third-party
Chaincode in general purpose languages (Go, Node.js, Java)	Smart contracts in domain-specific-language (DSL) e.g. Solidity

Hyperledger Fabric vs. Ethereum

Hyperledger Fabric	Ethereum
Broad, generic consensus mechanism	Proof-of-work (soon proof-of-scale)
Little wasted effort	Much wasted effort
Ease of scaling	Difficulty scaling

Concerns Around Ethereum

Scaling

Each peer node executes all transactions

Resource Usage

Proof-of-work algorithm incurs significant wasted effort

Confidentiality

Each transaction is broadcast to all peers in network

Hyperledger is designed to mitigate these concerns

AWS Blockchain Templates

AWS Blockchain Templates

AWS Blockchain Templates helps you quickly create and deploy blockchain networks on AWS using different blockchain frameworks.

Frameworks Supported

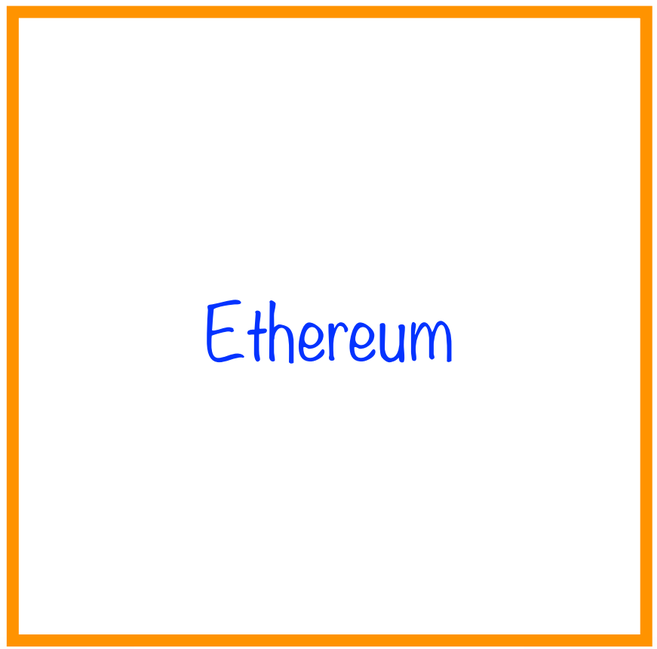
Ethereum

Open source blockchain framework from the Ethereum foundation

Hyperledger Fabric

Open source blockchain framework from the Linux foundation

Ethereum



Build a network of nodes to connect to the public network

Build your own private network for your organization or consortium

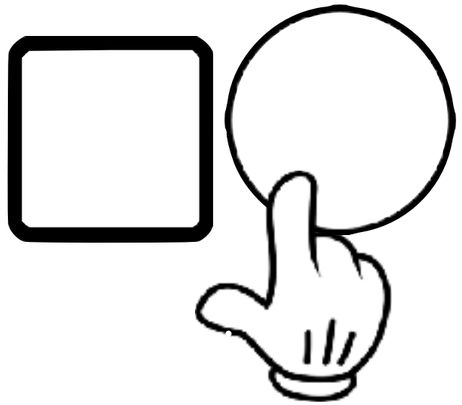
Hyperledger Fabric

Build your own private network for your organization or consortium

Limit transactions viewable by parties - allows access control for data on the blockchain

How AWS Blockchain Frameworks Work

Templates for network components



Ethereum or Hyperledger Fabric

Configure the blockchain network

Run decentralized applications

AWS Cloud Services Used

EC2: Compute capacity, miner nodes and client nodes which run EVM

ECS: Orchestrates container deployment among EC2 devices set up in a cluster

VPC: Provides network access to all cloud resources

Application Load Balancing: Single point of access to resources

Blockchain Templates configure and launch CloudFormation stacks to create blockchain networks

AWS CloudFormation

Model infrastructure in a text file

Standardize infrastructure components

Automate deployments

Provisions resources in a safe, repeatable manner

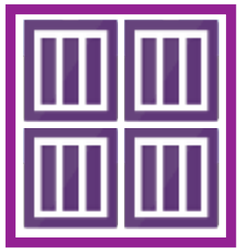
Infrastructure as code

Demo

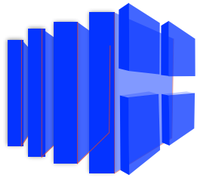
Configuring the AWS account

Blockchain Network Components

Blockchain framework containers



ECR registry



Elastic Load Balancing

OR

Amazon Route 53

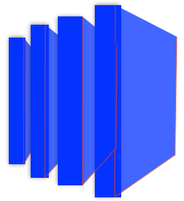
ECS container platform only

Multi-node blockchain network

Blockchain explorer and monitoring tools

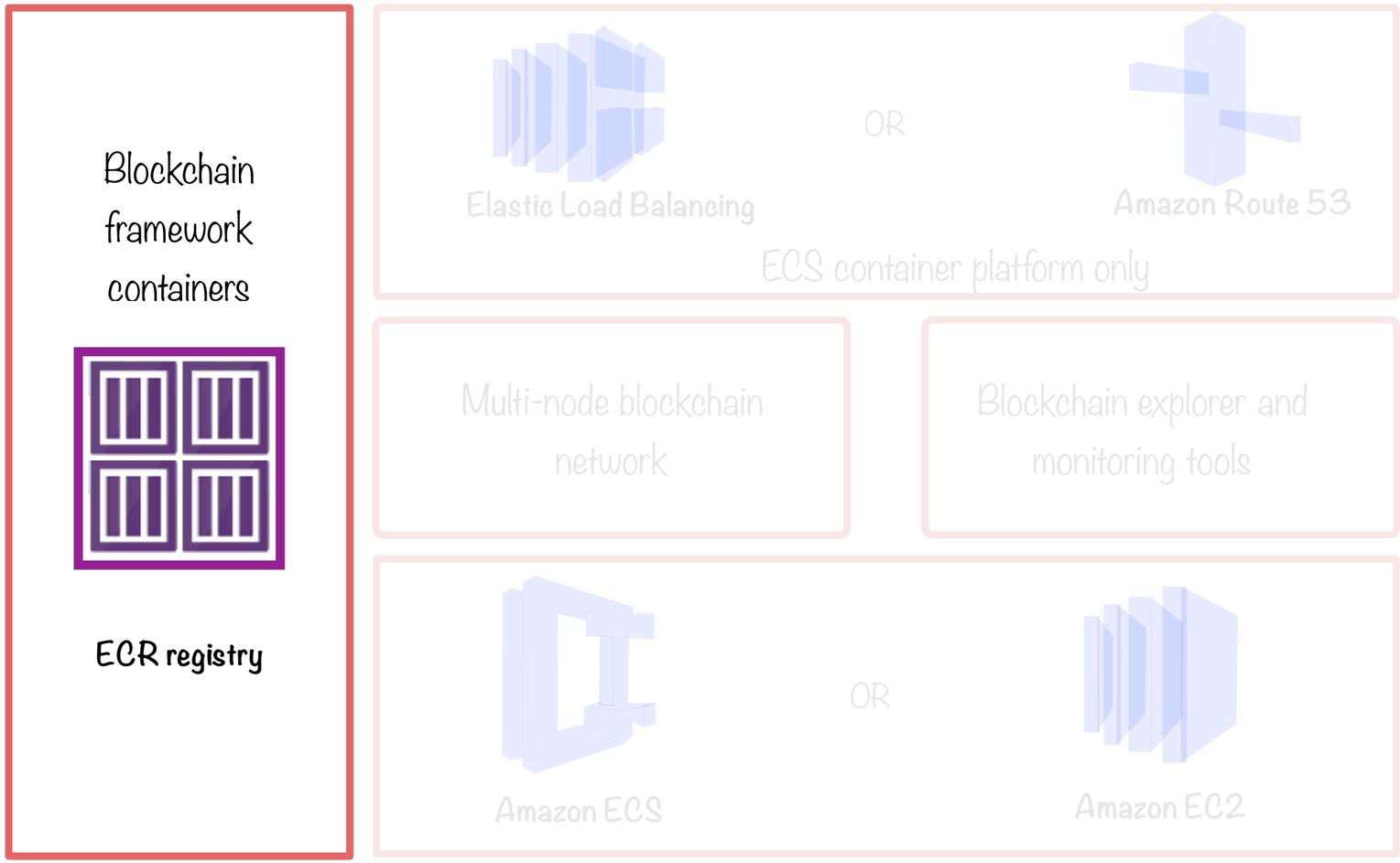
Amazon ECS

OR



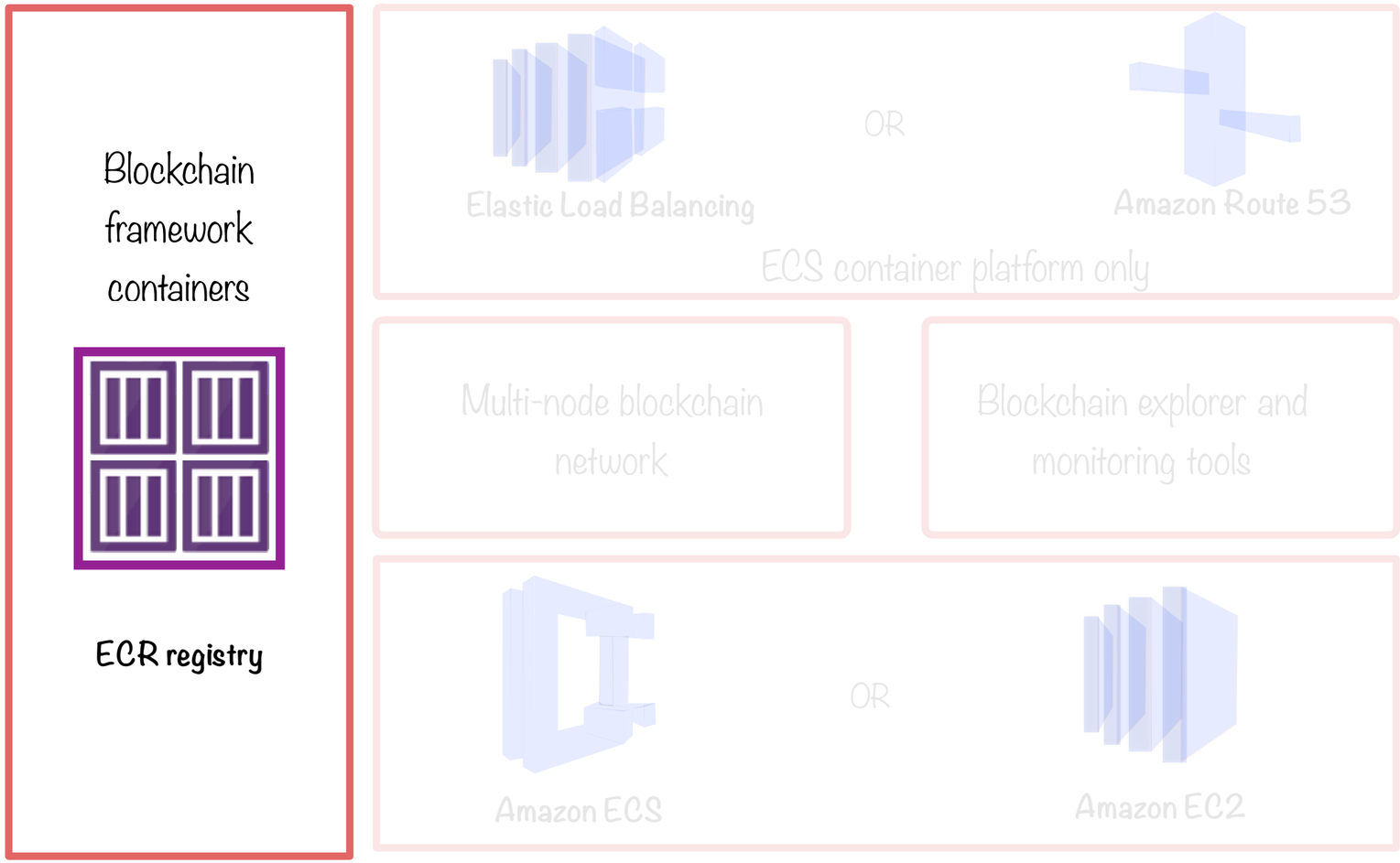
Amazon EC2

Docker Containers run Blockchain Software



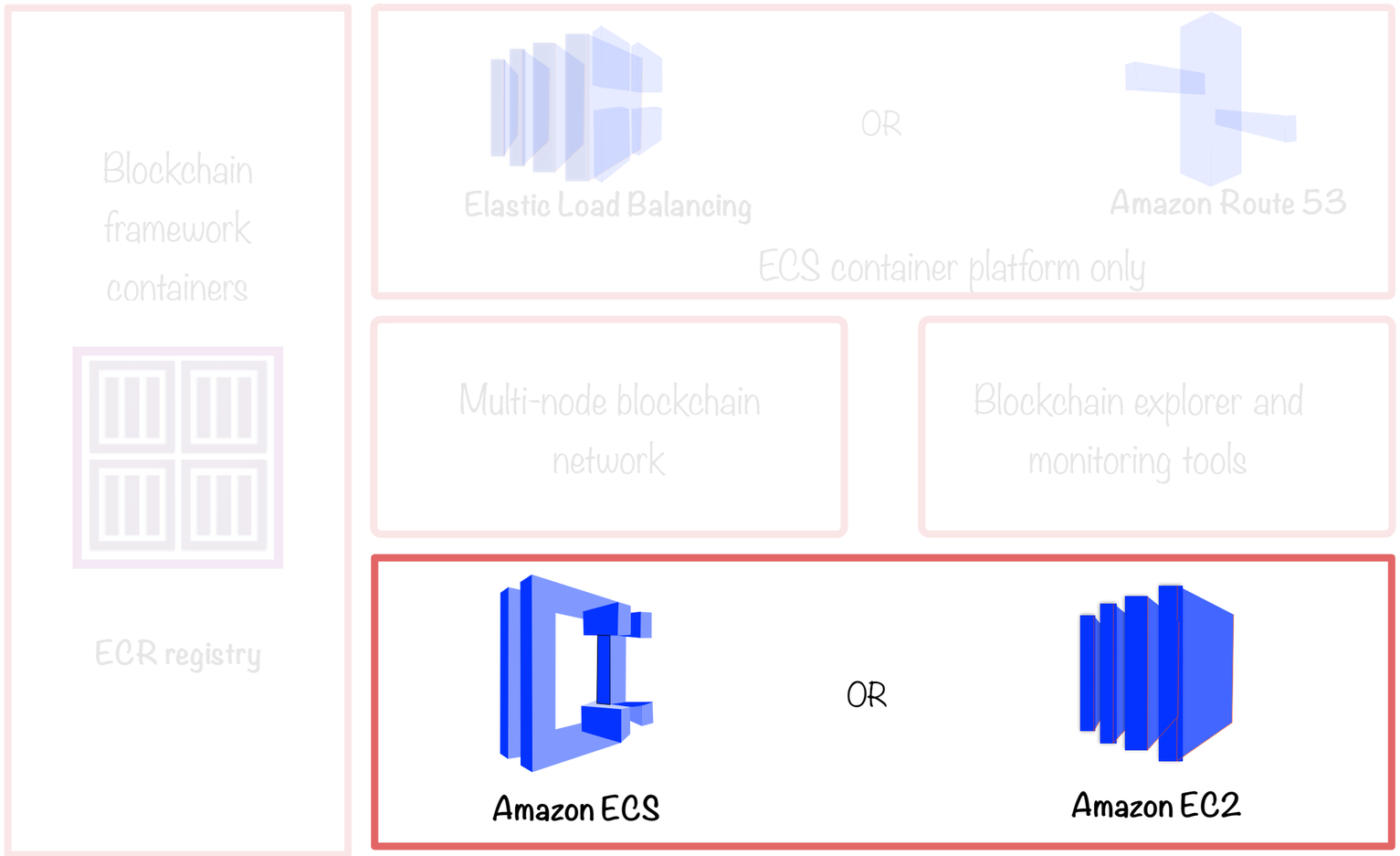
Blockchain software is present on Docker containers

Docker Containers run Blockchain Software



For Ethereum these container nodes run the EVM

ECS Setup or Docker Local Setup

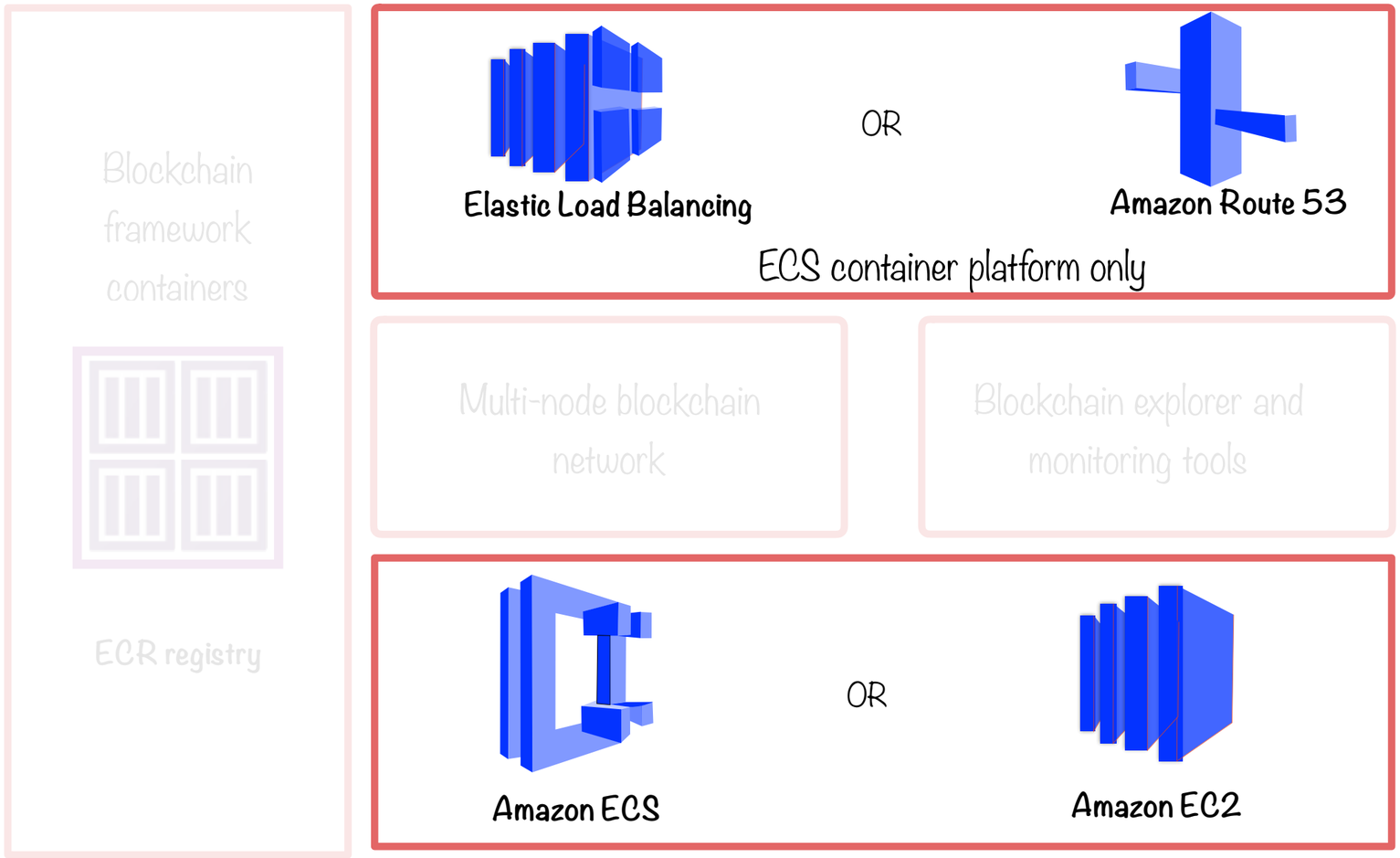


Ethereum containers can be on multiple ECS nodes

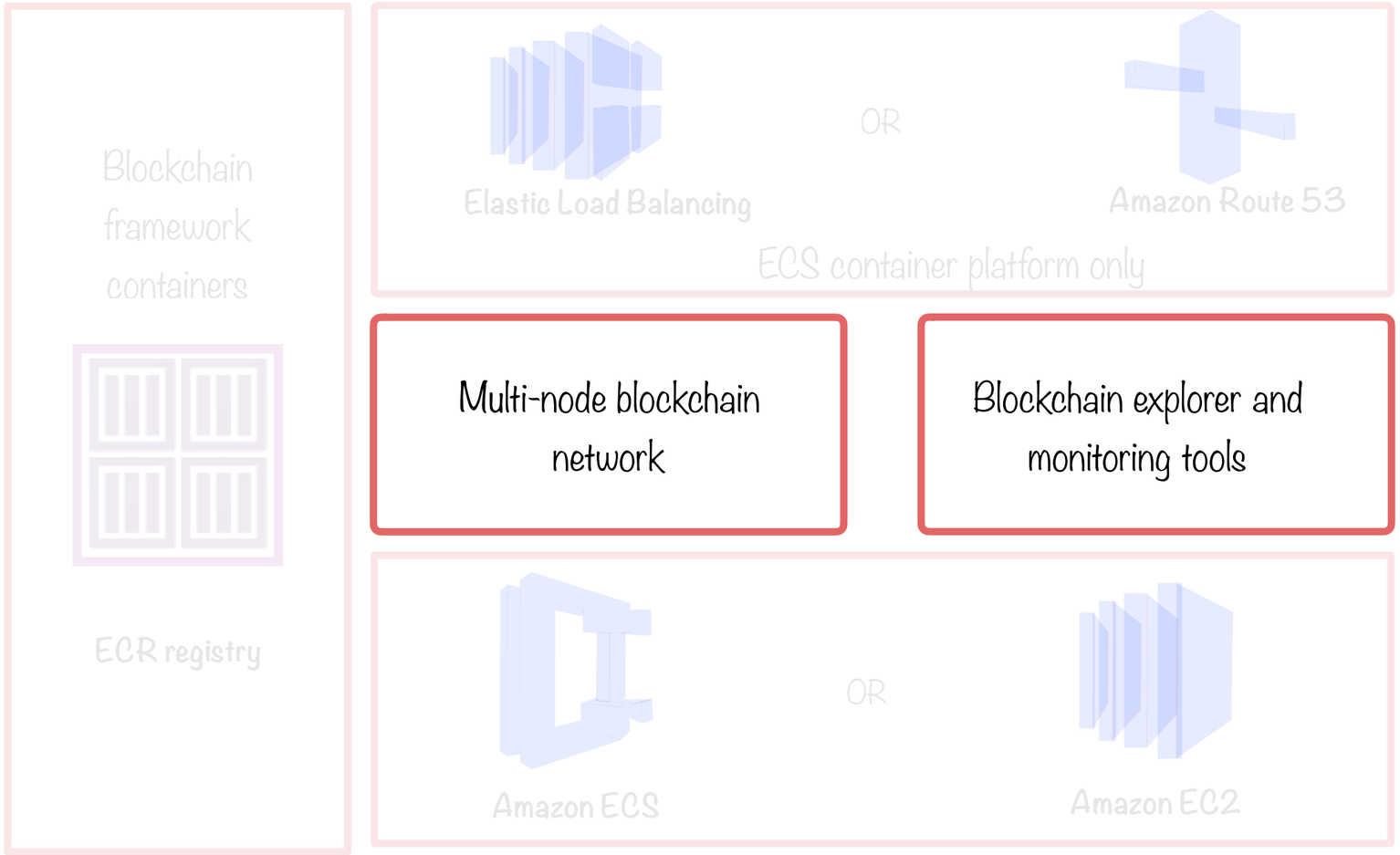
ECS Setup or Docker Local Setup

Simplified set up - have all Ethereum containers on the same EC2 instance

Load Balancers Route Traffic to Nodes

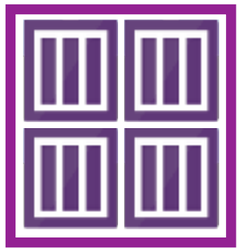


Monitoring Tools to Manage Network

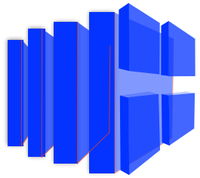


Blockchain Network Components

Blockchain framework containers



ECR registry



Elastic Load Balancing

OR

Amazon Route 53

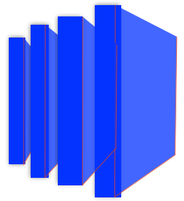
ECS container platform only

Multi-node blockchain network

Blockchain explorer and monitoring tools

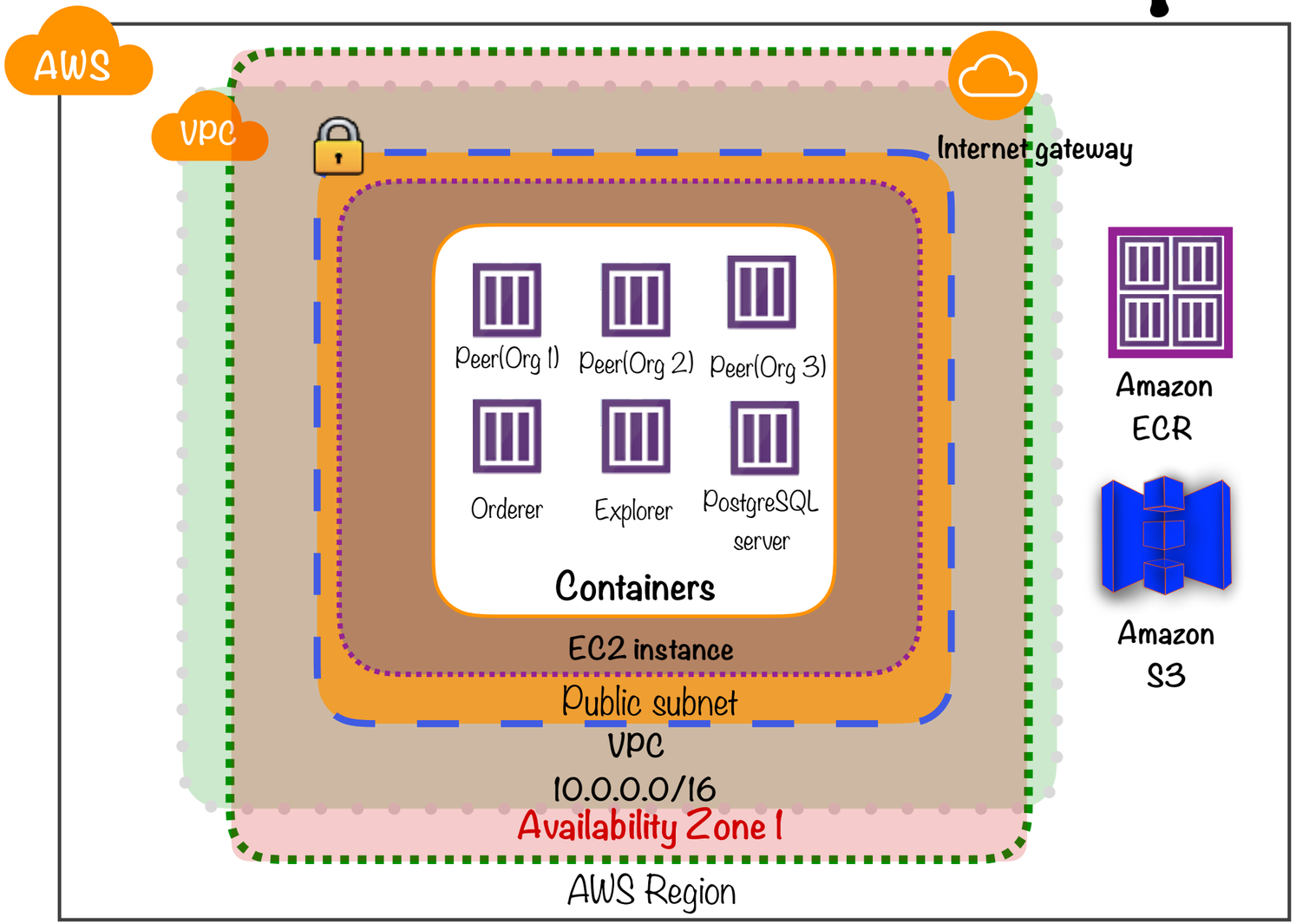
Amazon ECS

OR



Amazon EC2

Blockchain Network Components

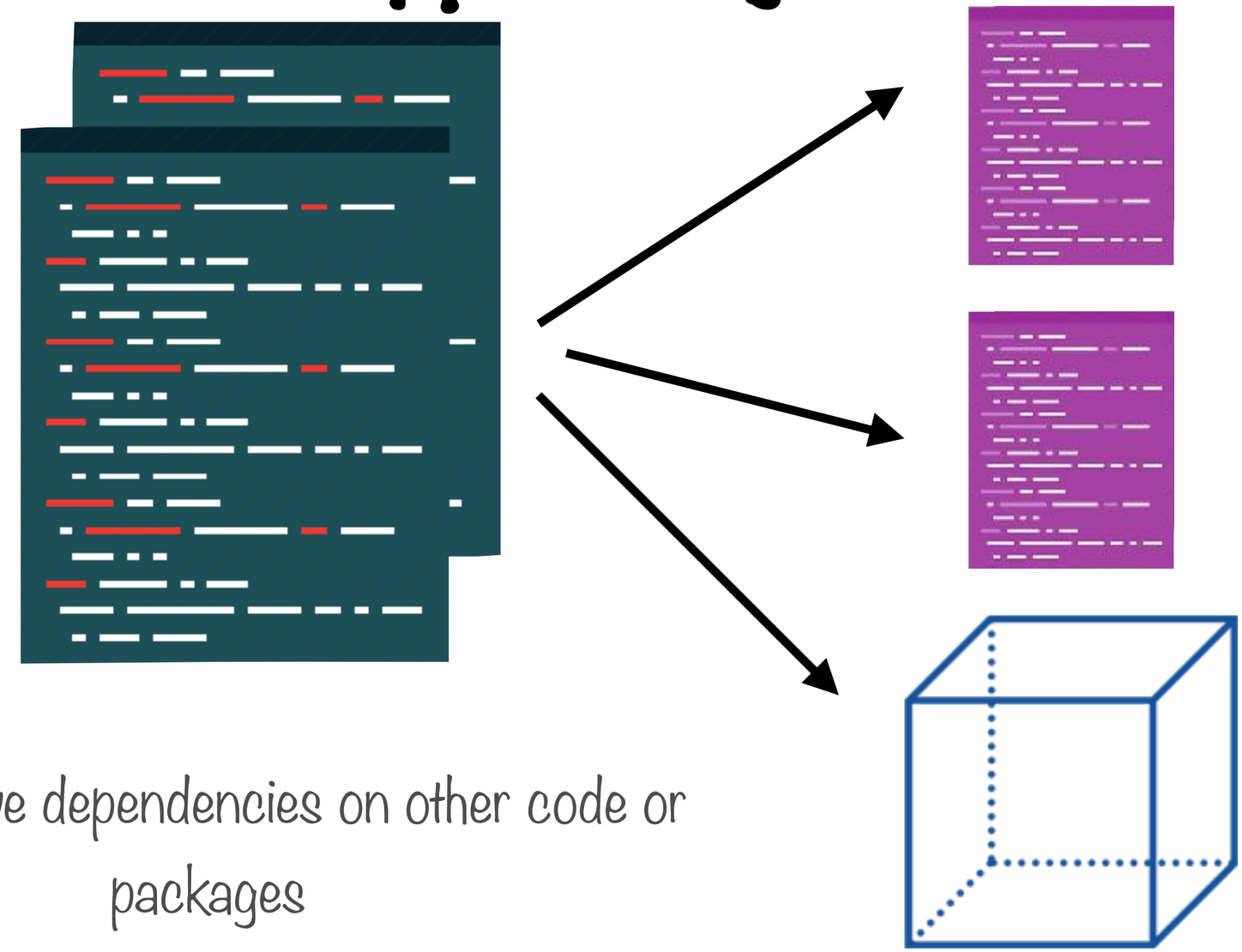


Docker for Hyperledger Fabric on AWS

Docker for Hyperledger Fabric on AWS

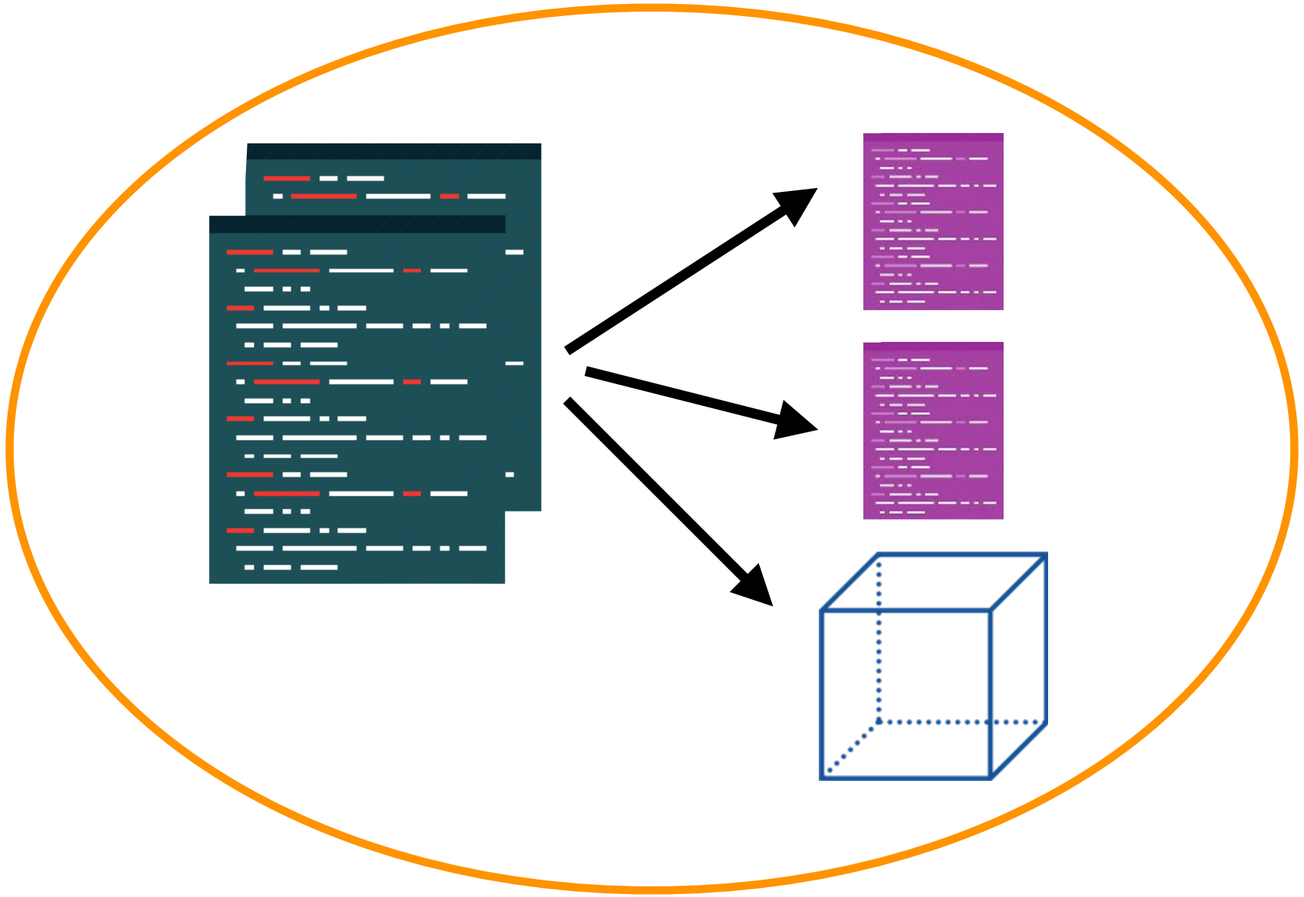
Application code

Docker for Hyperledger Fabric on AWS



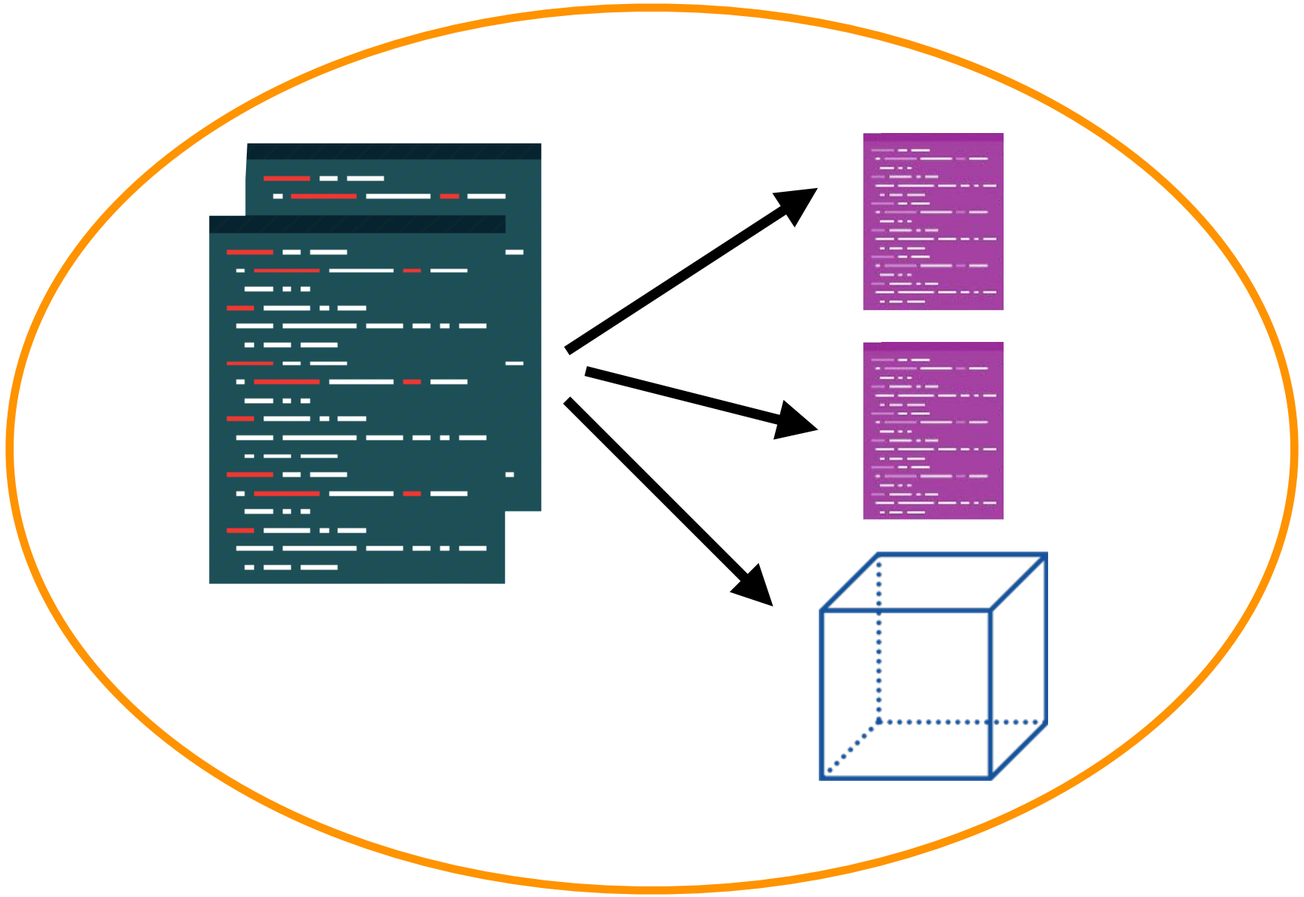
Code will have dependencies on other code or packages

Docker Containers



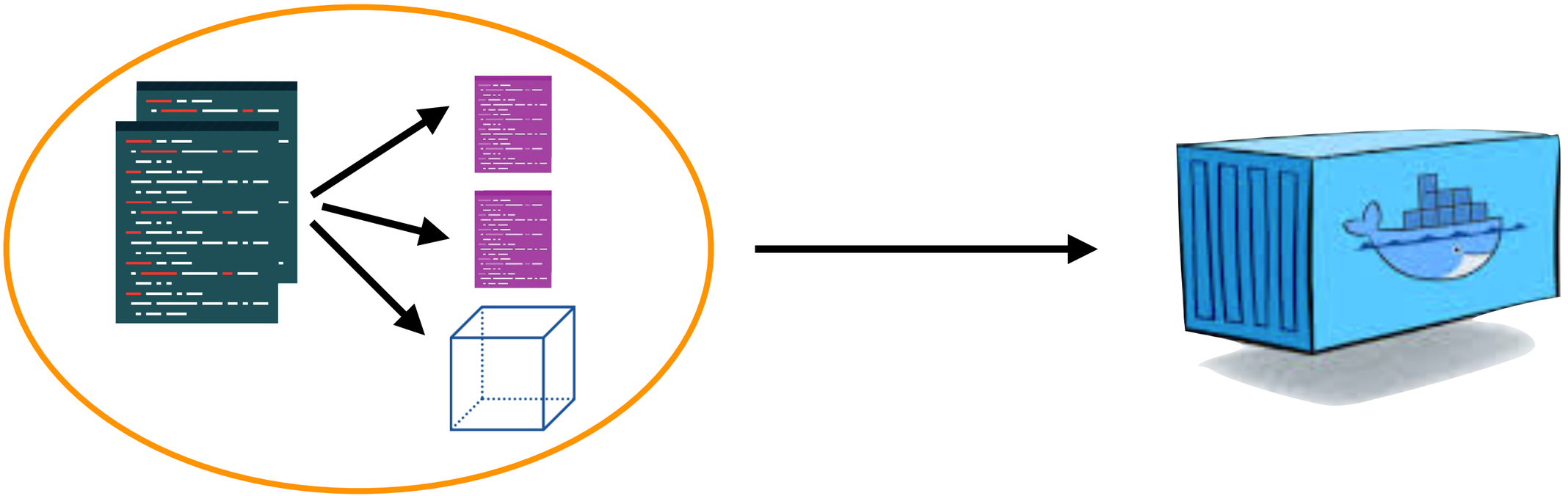
Docker packages all of this arbitrary code into an image

Docker Containers



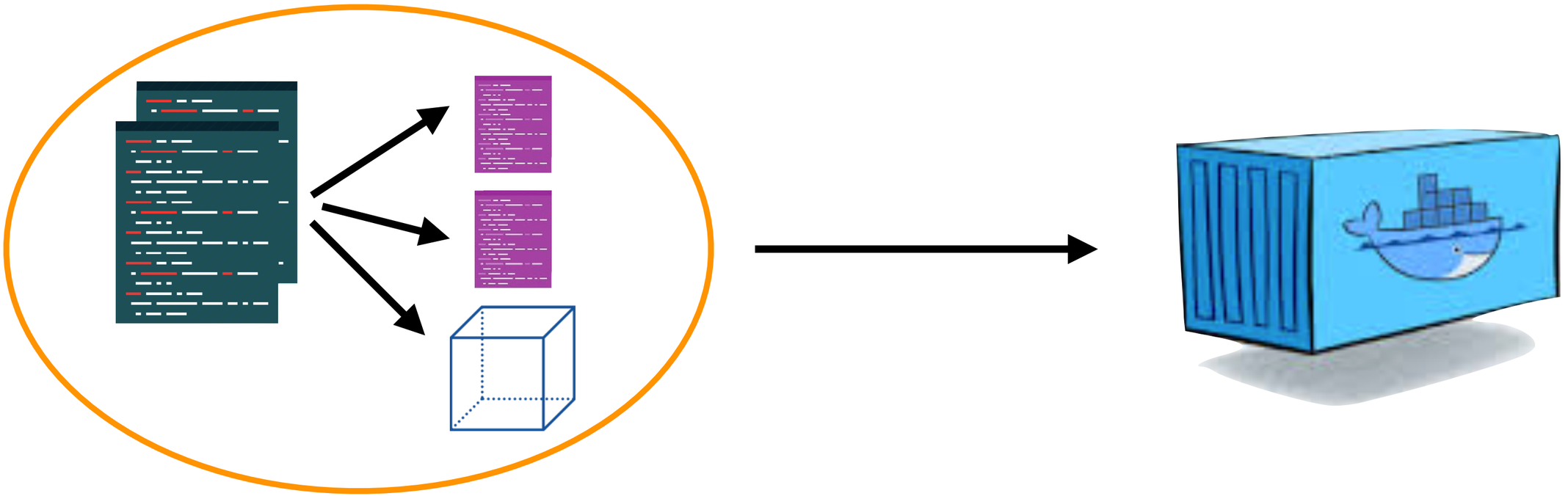
An image is completely self-sufficient

Docker Containers



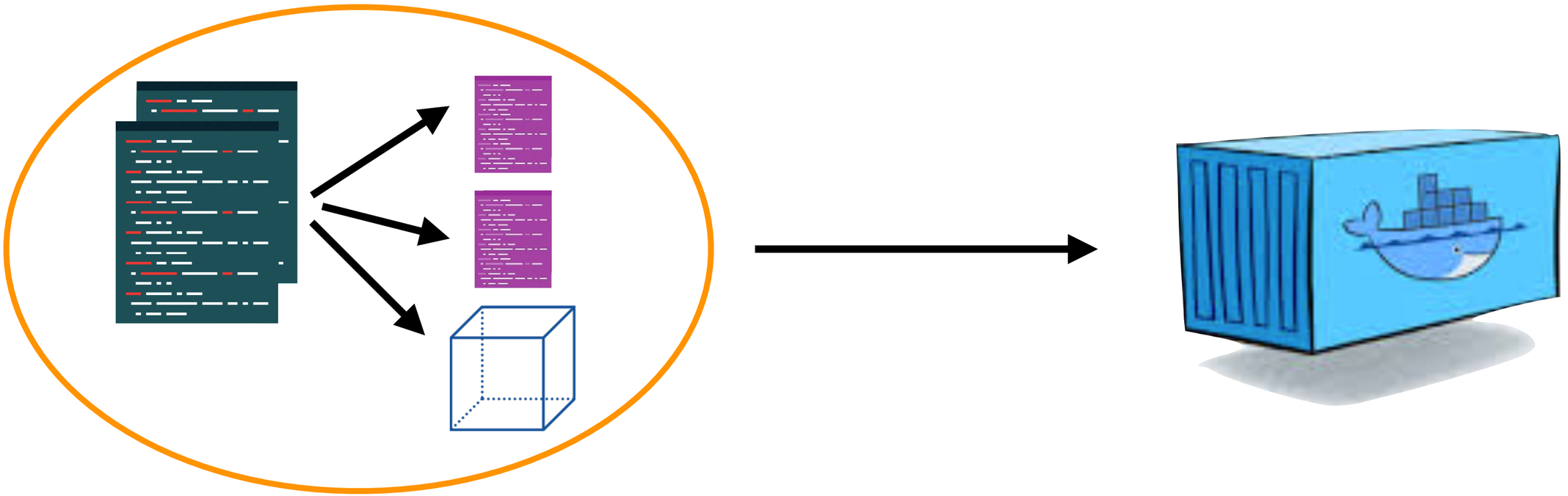
An image can be used to run a Docker container

Docker Containers



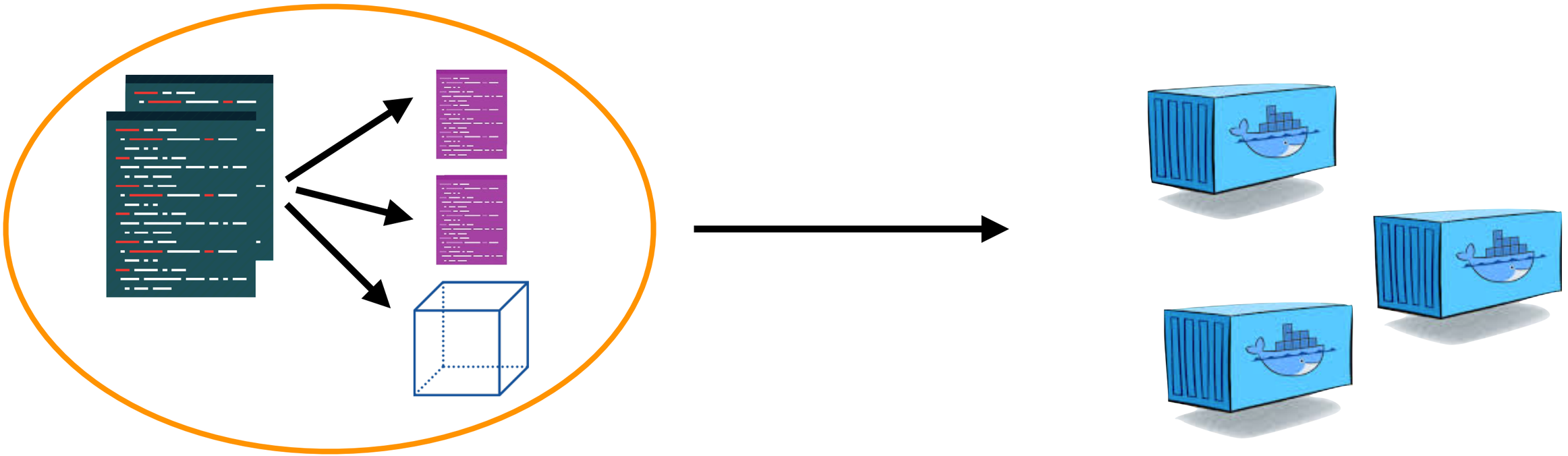
Containers are a fully self contained environment which executes code

Docker Containers



Containers do not contain the OS and your code is abstracted away from the machine

Docker Containers



You can create as many containers as you want from the same image

Secure | https://www.docker.com

docker

What is Docker? Product Community Support Create Docker ID Sign In

DOCKER PLATFORM ADDS KUBERNETES

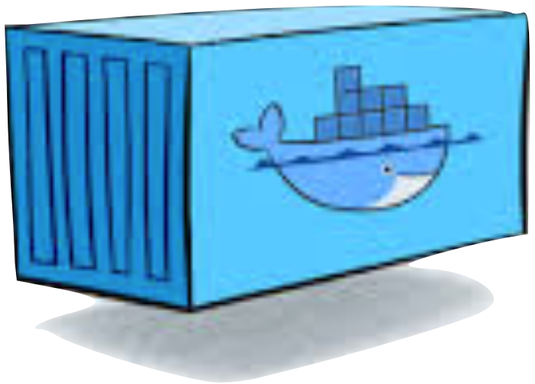
Simplify and advance the management of Kubernetes for enterprise IT

LEARN MORE

SIGN UP FOR THE BETA

Use Docker containers to use Hyperledger Fabric on AWS

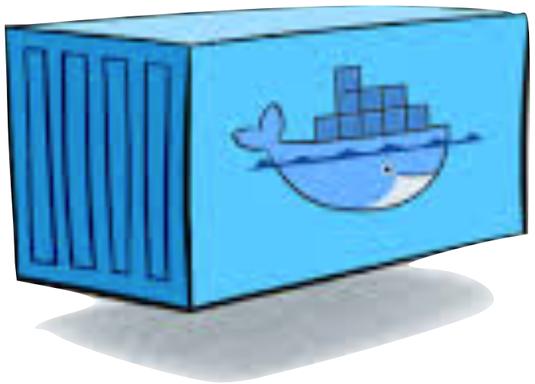
Docker



Create package with

- application code
- dependencies
- user space of Linux OS where app was developed

Docker Compose

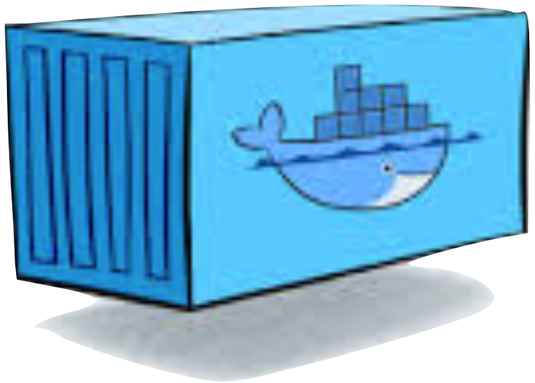


Create multi-container app

Specification in YAML file

- Yet Another Markup Language

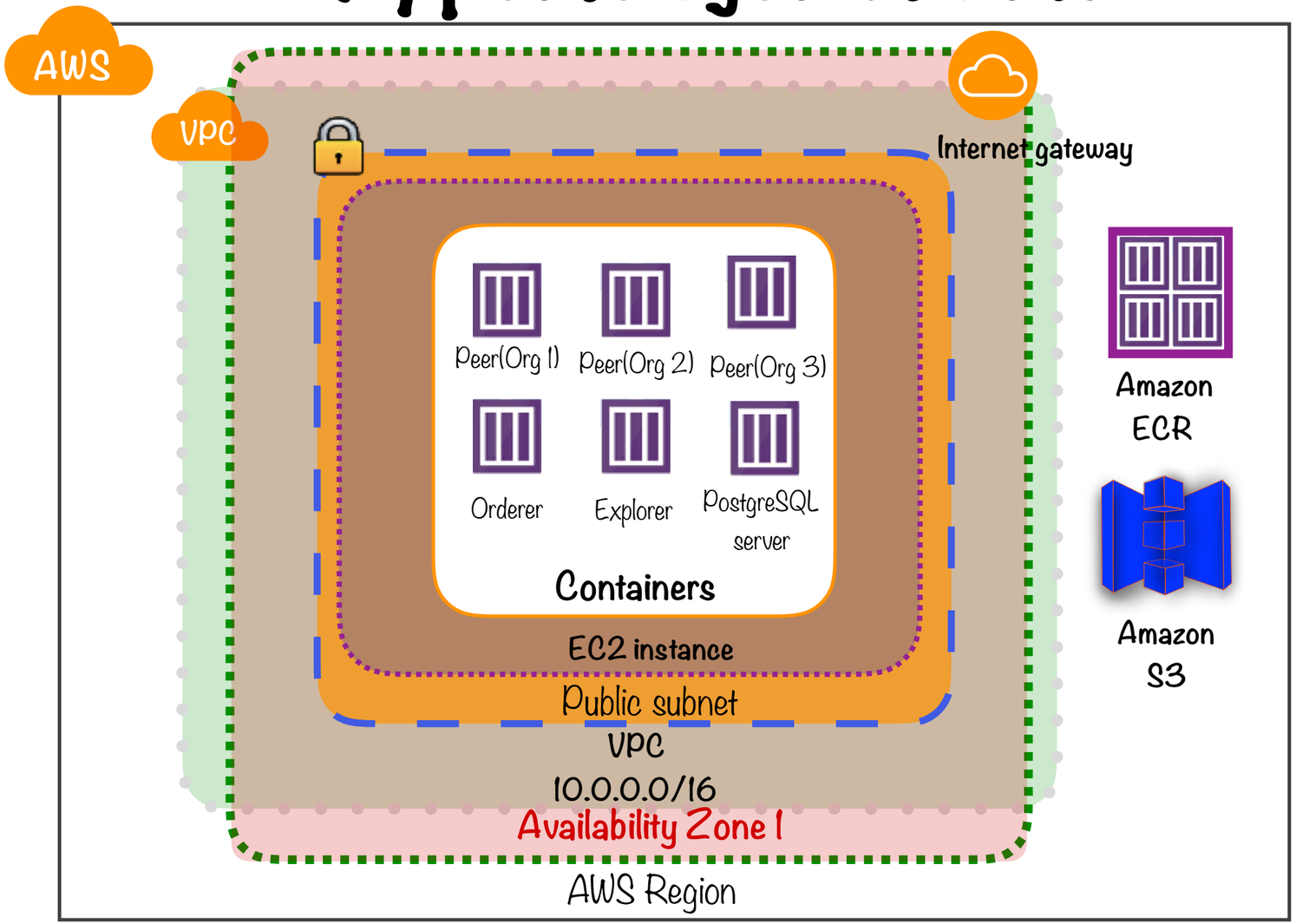
Docker Compose for Hyperledger Fabric on AWS



AWS provides YAML file

Creates required containers

Using the AWS Blockchain Template for Hyperledger Fabric



Demo

Setting up the Hyperledger Fabric network

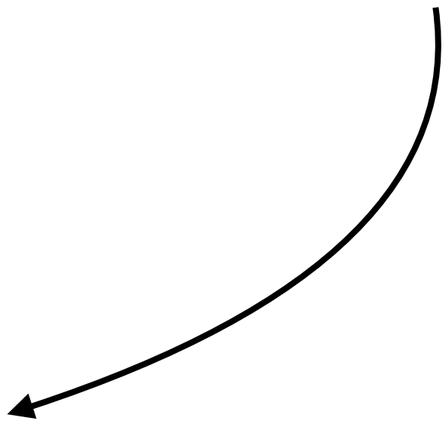
Demo

Creating nodes in the Hyperledger Fabric network

Blockchain

An encrypted database of agreements

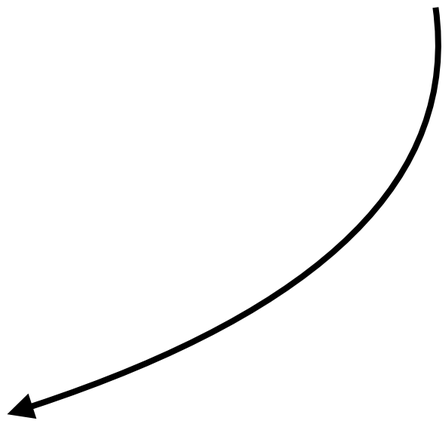
Once a deal has been recorded neither party can go back and rewrite terms



Blockchain

An encrypted database of agreements

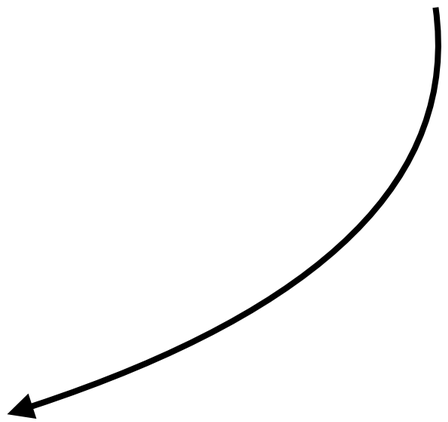
Recorded in the form of verified transactions



Blockchain

An encrypted database of agreements

Multiple transactions are stored in the form of blocks



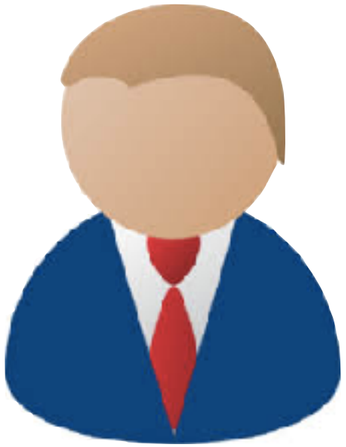
Blockchain

An encrypted database of agreements

Transactions

A

Sender



B

Recipient

Messages sent by participants to each other to transfer value

Transactions

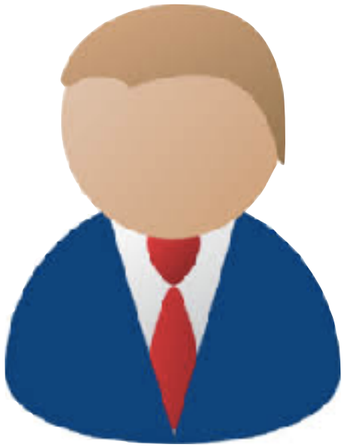
Contents of transaction

- recipient
- signature of sender
- value of assets to send to recipient

Transactions

A

Sender



B

Recipient

These transactions need to be verified and reflected in the state of all nodes in the network

Transaction Verification

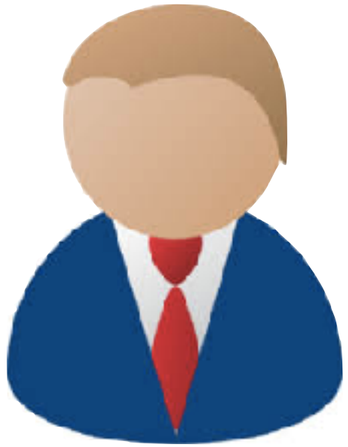
A

Sender

Sender : A

Recipient : B

Value : 5 units



B

Recipient

Initiated transaction requests are broadcast to the entire blockchain network

Transaction Verification

Verifying sender and transaction integrity

Confirm that sender owns assets to send

Smart Contract

Mechanism that allows common contractual clauses to be specified, verified or enforced even in the absence of trust between contracting parties and in the absence of a third party

Chaincode

A self-contained program that runs on Hyperledger Fabric and satisfies a standard interface; typically used to implement smart contracts

Implemented in Go,
Node.js or Java

Chaincode

A self-contained program that runs on Hyperledger Fabric and satisfies a standard interface; typically used to implement smart contracts

Maintains its own private state

Chaincode

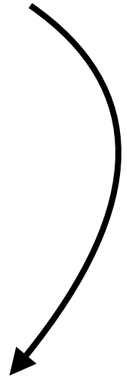
A self-contained program that runs on Hyperledger Fabric and satisfies a standard interface; typically used to implement smart contracts

Chaincode

A self-contained program that runs on Hyperledger Fabric and satisfies a standard interface; typically used to implement smart contracts

ChainCodeStubInterface which specifies Init and Invoke methods

Any peer on the Hyperledger Fabric network can invoke them

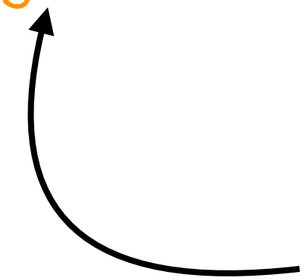


Chaincode

A self-contained program that runs on Hyperledger Fabric and satisfies a standard interface; typically used to implement smart contracts

Chaincode

A self-contained program that runs on Hyperledger Fabric and satisfies a standard interface; typically used to implement smart contracts



Chaincode contains coded-up logic of the agreements between participants

Consensus

Blockchain avoids use of trusted third parties

Verification needs to adhere to a carefully designed algorithm

Ensure agreement among peers on network

“Consensus algorithms”

Consensus in Hyperledger Fabric: Execute-order-validate

Common Transaction Lifecycle

Most blockchain platforms operate in two steps

Order

Transactions added to ledger

All peers need to execute transaction in same order

Execute

Smart contract code runs

Typically in domain-specific language e.g. Solidity

Common Transaction Lifecycle

Hyperledger follows a different, more elaborate process

Execute

Chaincode for smart contracts

No need for DSL; use Go, Java, Node.js

Validate

Each peer validates transaction

Prevents double-spending

Order

Add to ledger if enough peers endorse transaction

Only endorsed transactions need to be ordered

Hyperledger Transaction Lifecycle

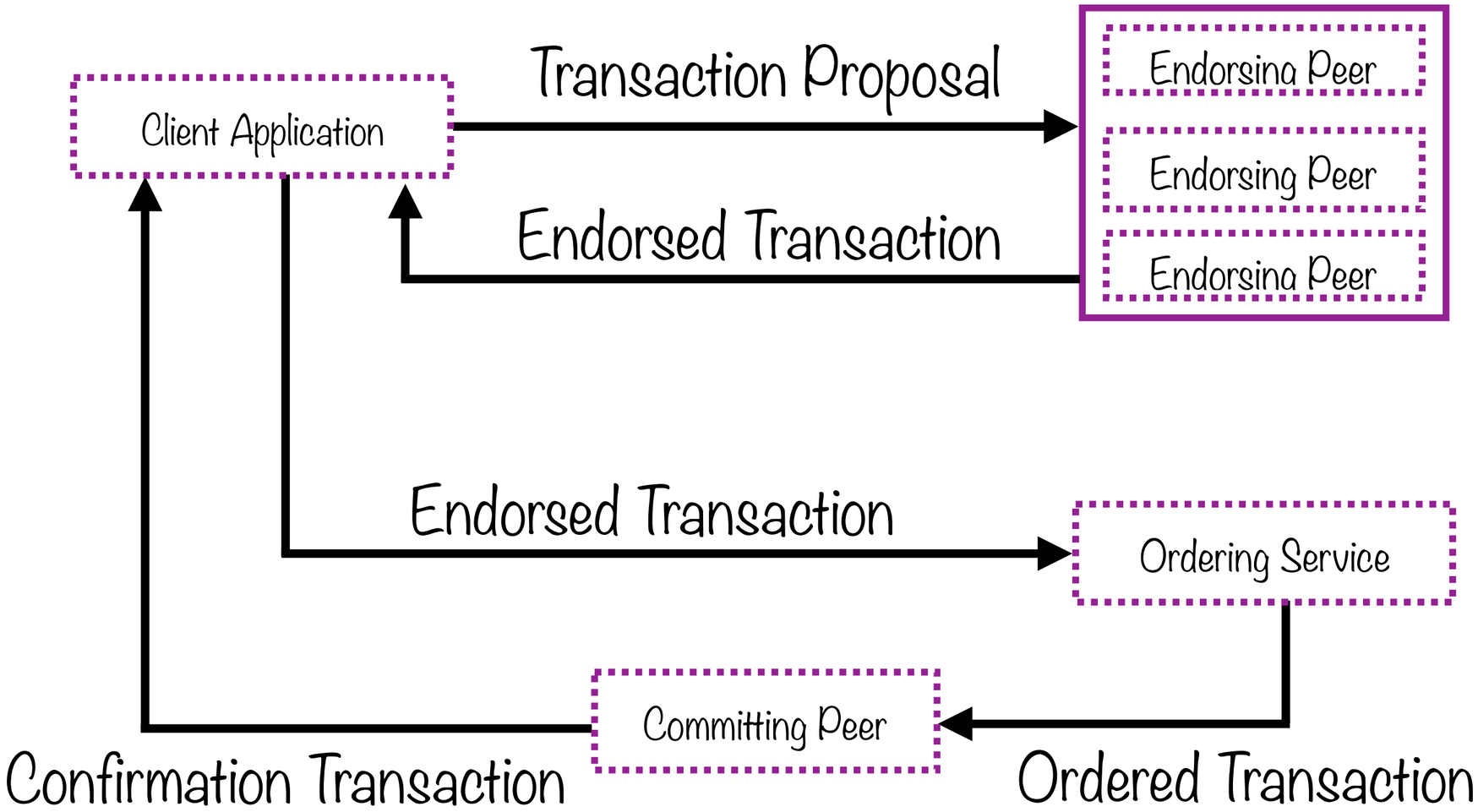
Hyperledger follows a different, more elaborate process

Execute

Chaincode for smart contracts

No need for DSL; use Go, Java, Node.js

Transaction Flow in Fabric



Execution

Execute

Chaincode for smart contracts

No need for DSL; use Go, Java, Node.js

Can happen in any order

Even in parallel

Relies on Endorsement Policy

Transaction Proposal

Information needed to trigger chaincode

Does the proposal have support?

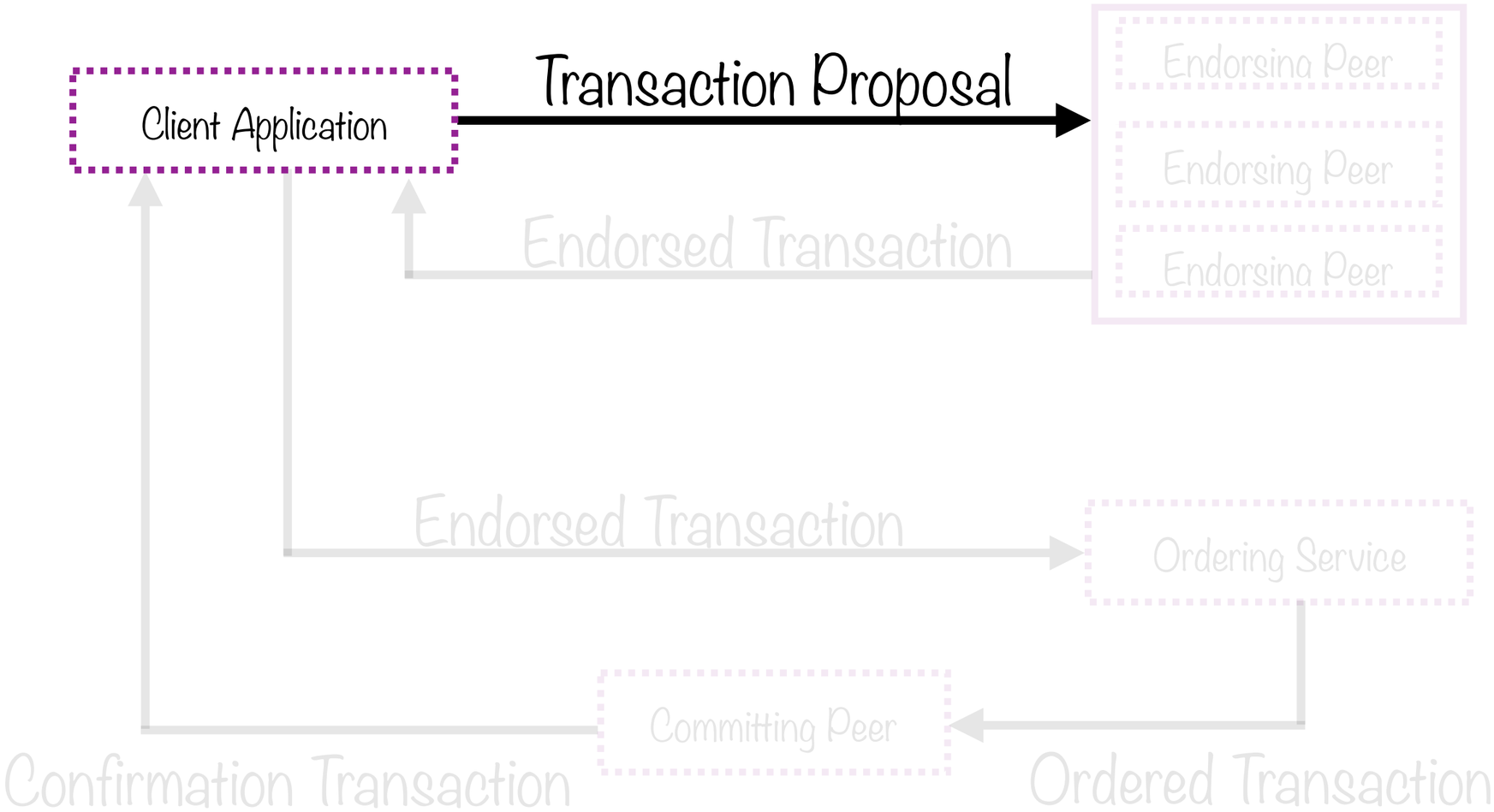
Check the endorsement policy

Execute

Chaincode for smart contracts

No need for DSL; use Go, Java, Node.js

Transaction Flow in Fabric



Endorsement Policy

Execute

Chaincode for smart contracts

No need for DSL; use Go, Java, Node.js

Specific policy to ensure some nodes agree on transaction results

Not all nodes need to be involved

Endorsement Policy

Execute

Chaincode for smart contracts

No need for DSL; use Go, Java, Node.js

Only move to next step if transaction proposal is endorsed

Not every peer needs to be aware

Allows private transactions

Endorsement Policy

Execute

Chaincode for smart contracts

No need for DSL; use Go, Java, Node.js

Examples

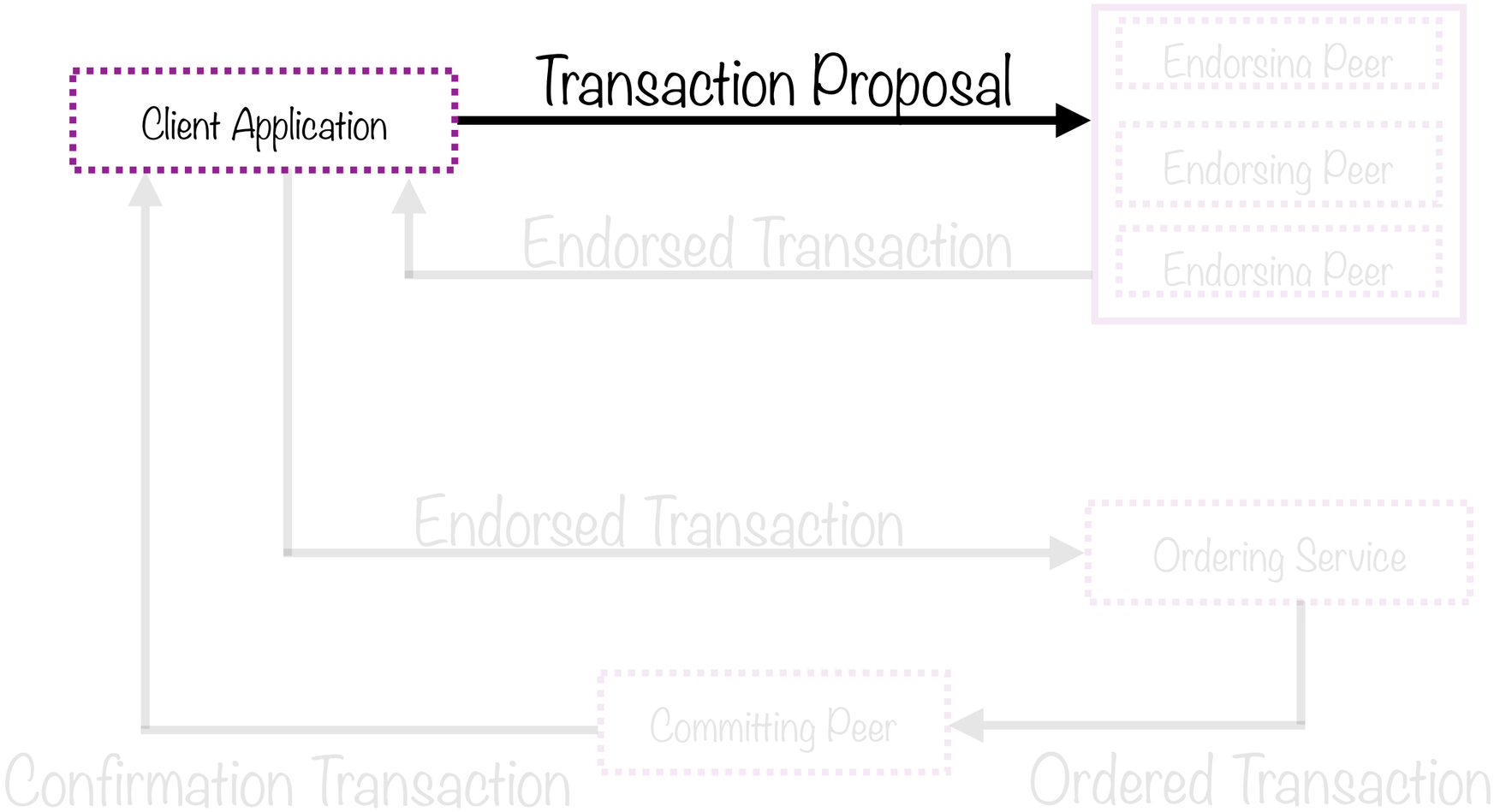
- "Specific list of peers must all endorse"
- "Majority of peers must endorse"
- "At least N peers in list must endorse"

Endorsement Policy

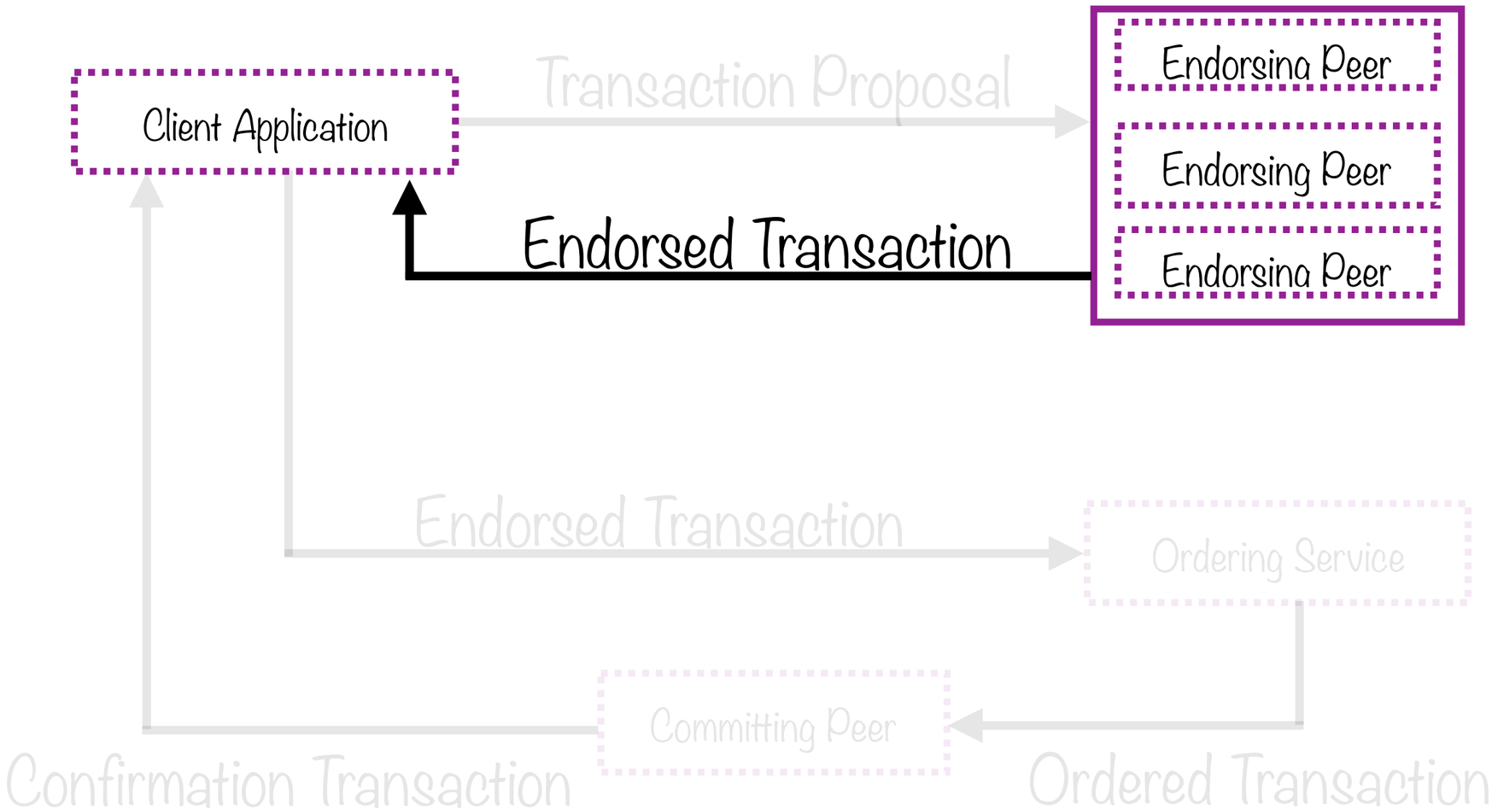
Need to actually execute smart contract

So, must have copy of chaincode

Transaction Flow in Fabric



Transaction Flow in Fabric



Hyperledger Transaction Lifecycle

Hyperledger follows a different, more elaborate process

Execute

Chaincode for smart contracts

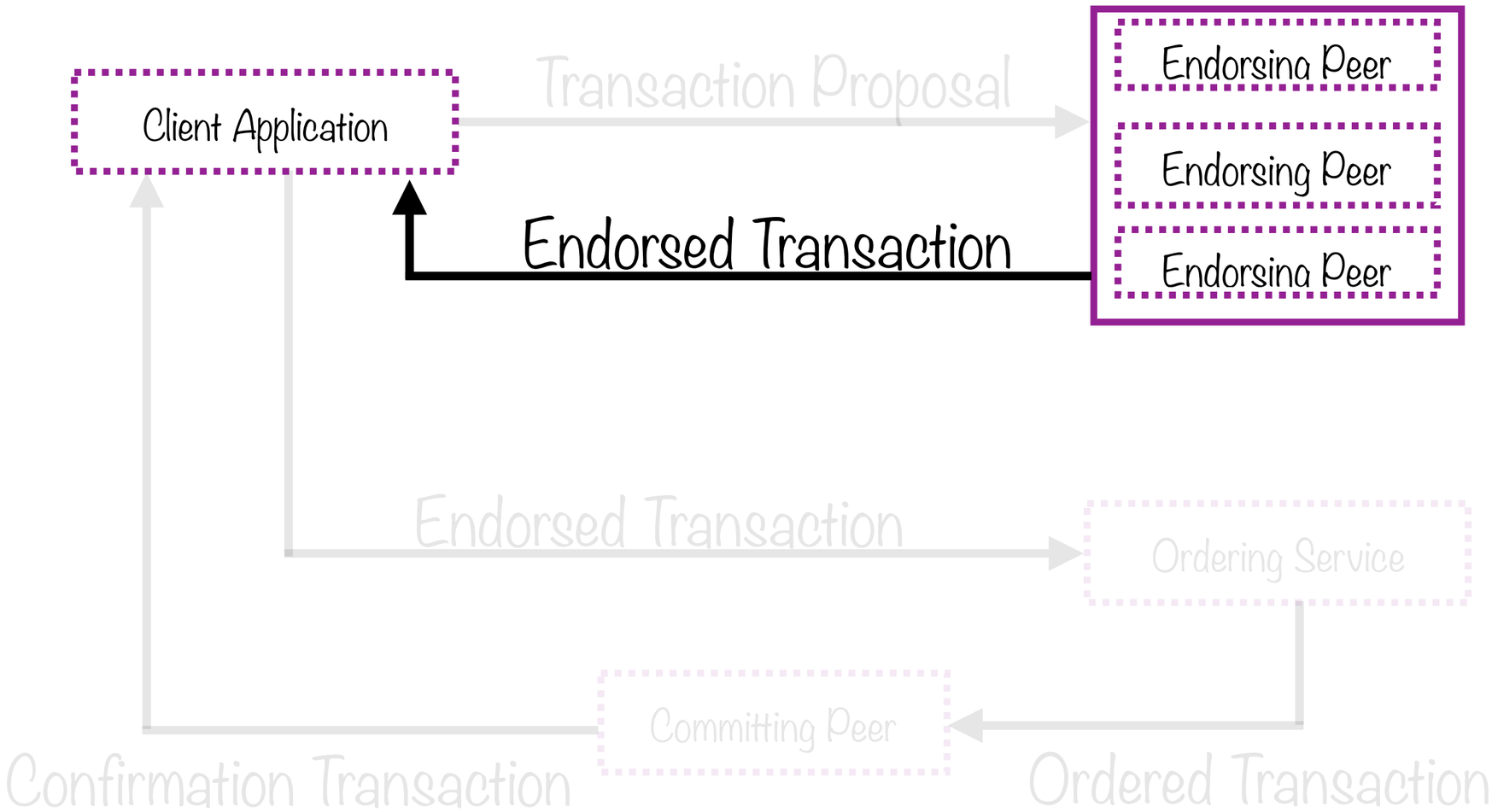
No need for DSL; use Go, Java, Node.js

Order

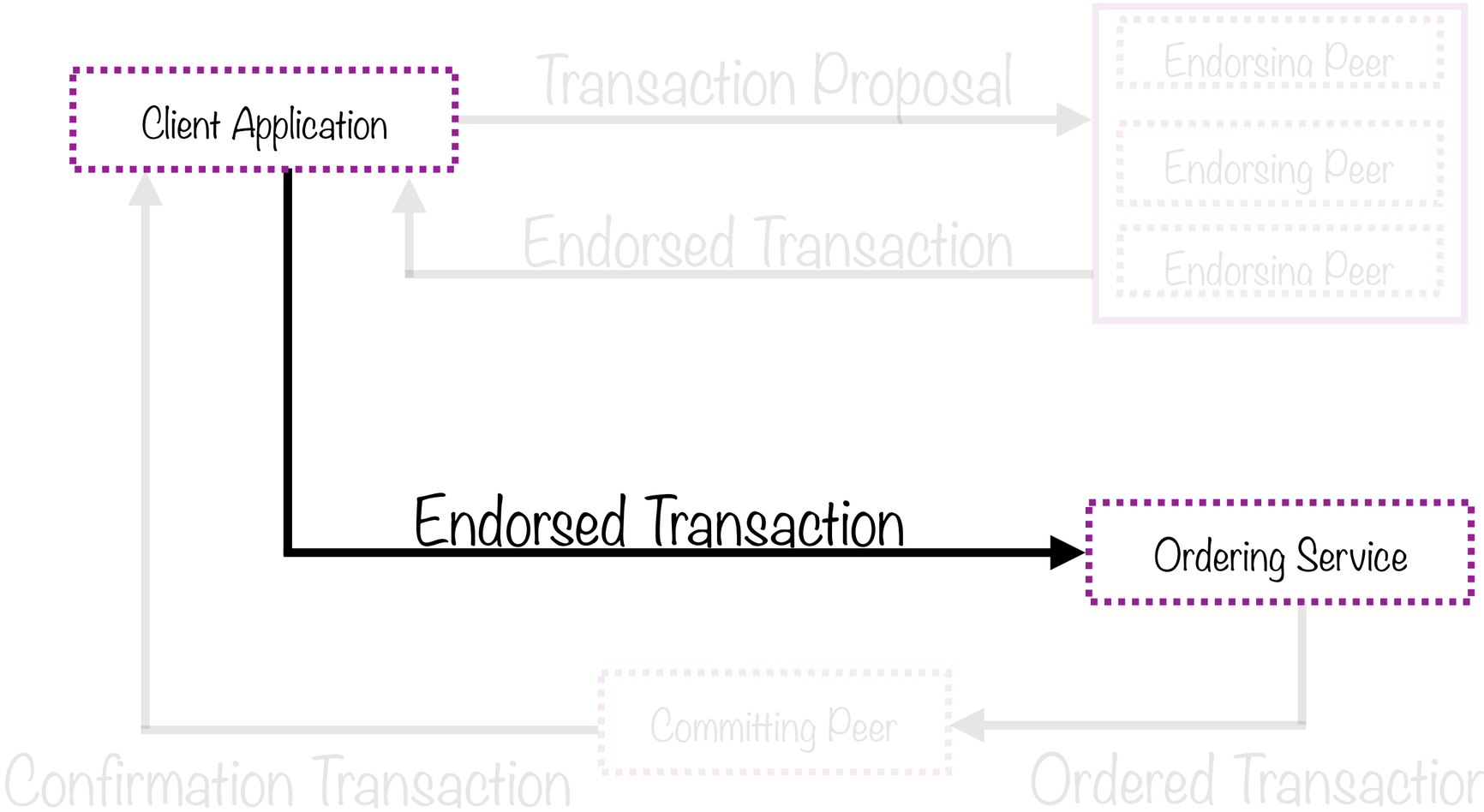
Add to ledger if enough peers endorse transaction

Only endorsed transactions need to be ordered

Transaction Flow in Fabric



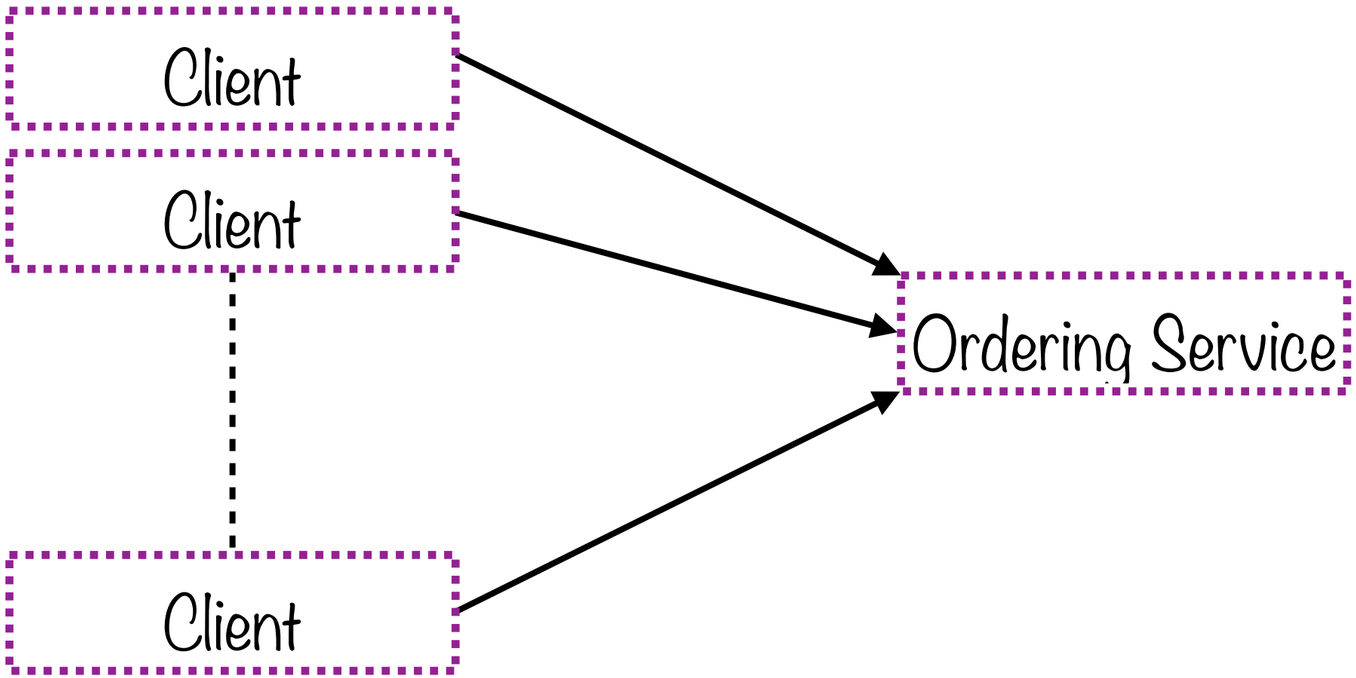
Transaction Flow in Fabric



Transaction Ordering

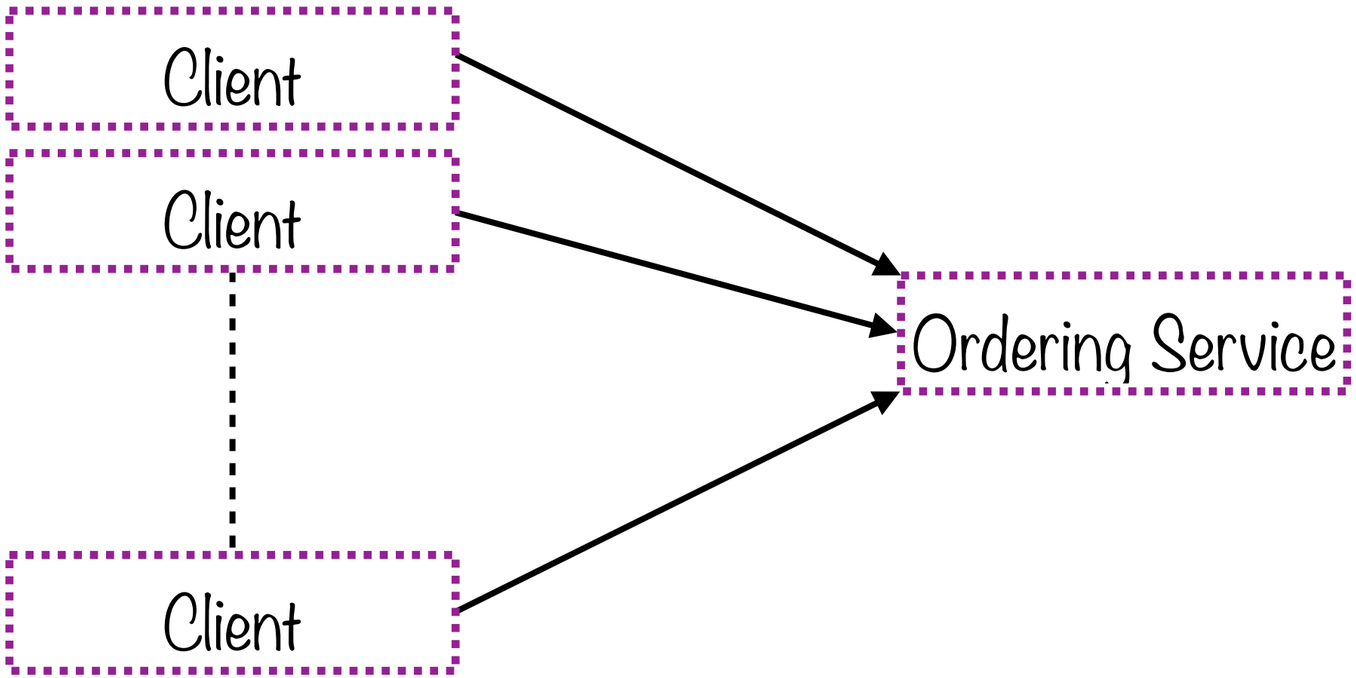
Client application submits the endorsed transaction to the Ordering service

Transaction Ordering



In reality multiple clients will be submitting endorsed transactions to the Ordering service

Transaction Ordering



Ordering service accepts the endorsed transactions and specifies the order in which those transactions will be committed to the ledger

Ordering Mechanisms in Hyperledger Fabric

Agreed-upon ordering mechanisms

Ordering Mechanisms in Hyperledger Fabric

SOLO

Kafka

SBFT

Single ordering node - useful for experimental uses

Ordering Mechanisms in Hyperledger Fabric

SOLO

Kafka

SBFT

Widely used and trusted technology

Kafka

Default ordering service

Production use-cases

Distributed

Fault-tolerant

Unified, high-throughput, low-latency

Ordering Mechanisms in Hyperledger Fabric

SOLO

Kafka

SBFT

SBFT

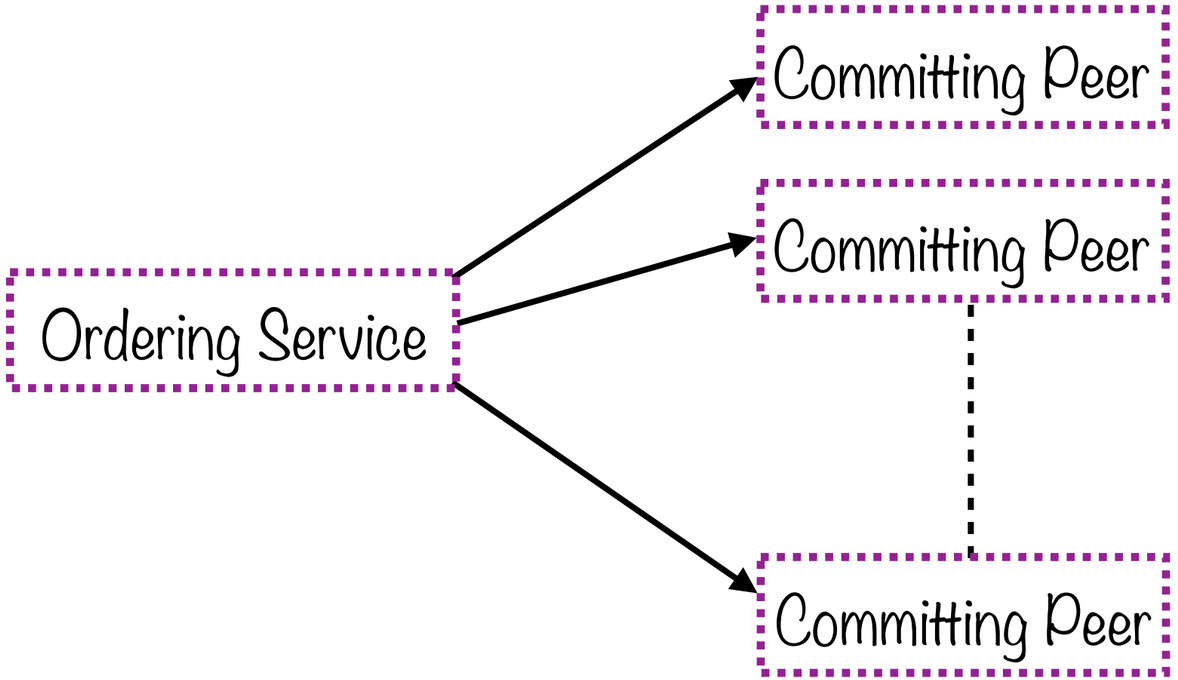
Simplified Byzantine Fault Tolerance

Works even in presence of malicious nodes

Two types of fault tolerance

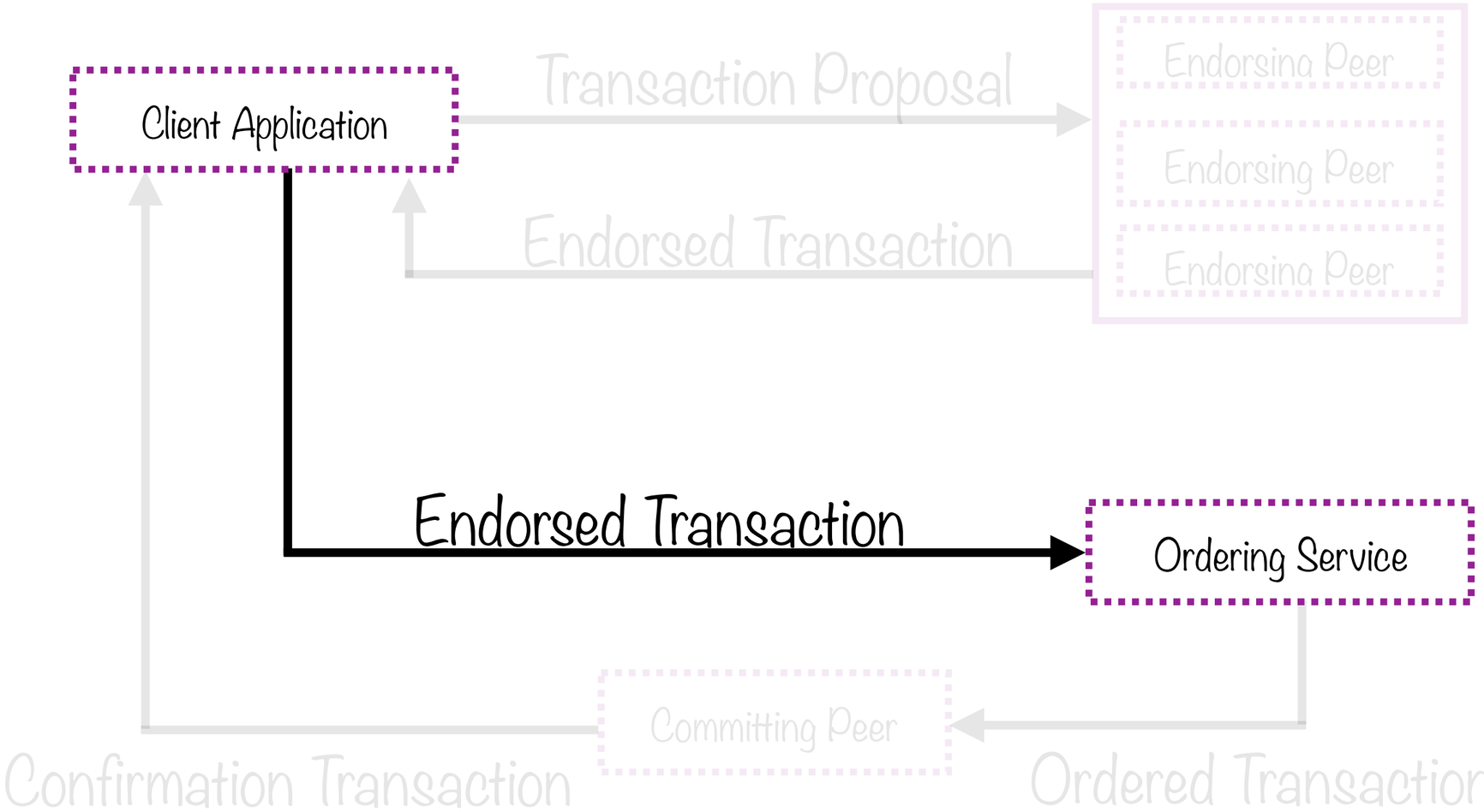
- crash fault tolerance
- byzantine fault tolerance

Ordering Mechanisms in Hyperledger Fabric

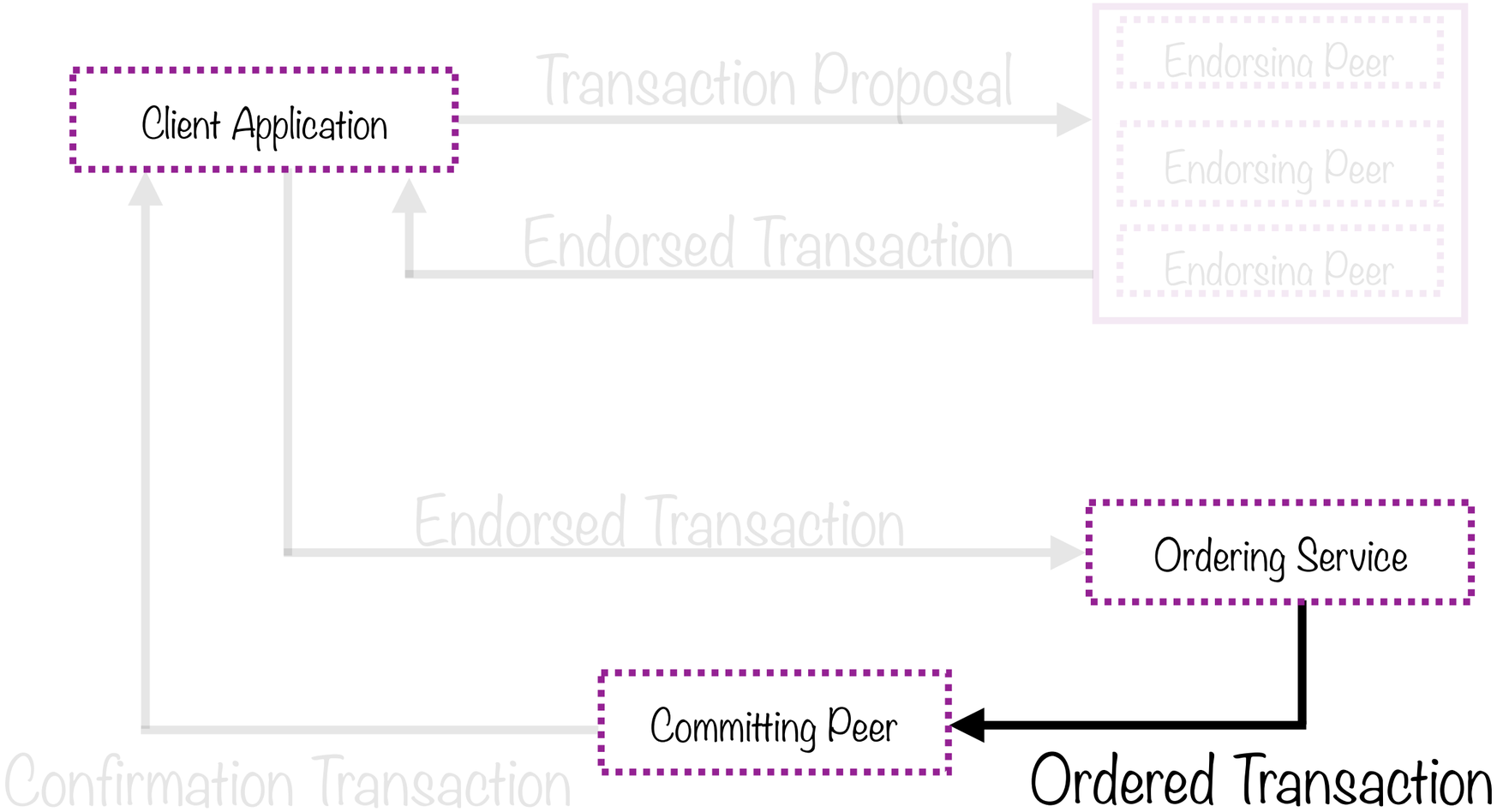


Ordering service orders endorsed transactions; these are then set into a block which is sent to each committing peer

Transaction Flow in Fabric



Transaction Flow in Fabric



Hyperledger Transaction Lifecycle

Hyperledger follows a different, more elaborate process

Execute

Chaincode for smart contracts

No need for DSL; use Go, Java, Node.js

Validate

Each peer validates transaction

Prevents double-spending

Order

Add to ledger if enough peers endorse transaction

Only endorsed transactions need to be ordered

Committing Peers

Need to validate (not execute) transaction

Need not have copy of chain code

Committing Peers

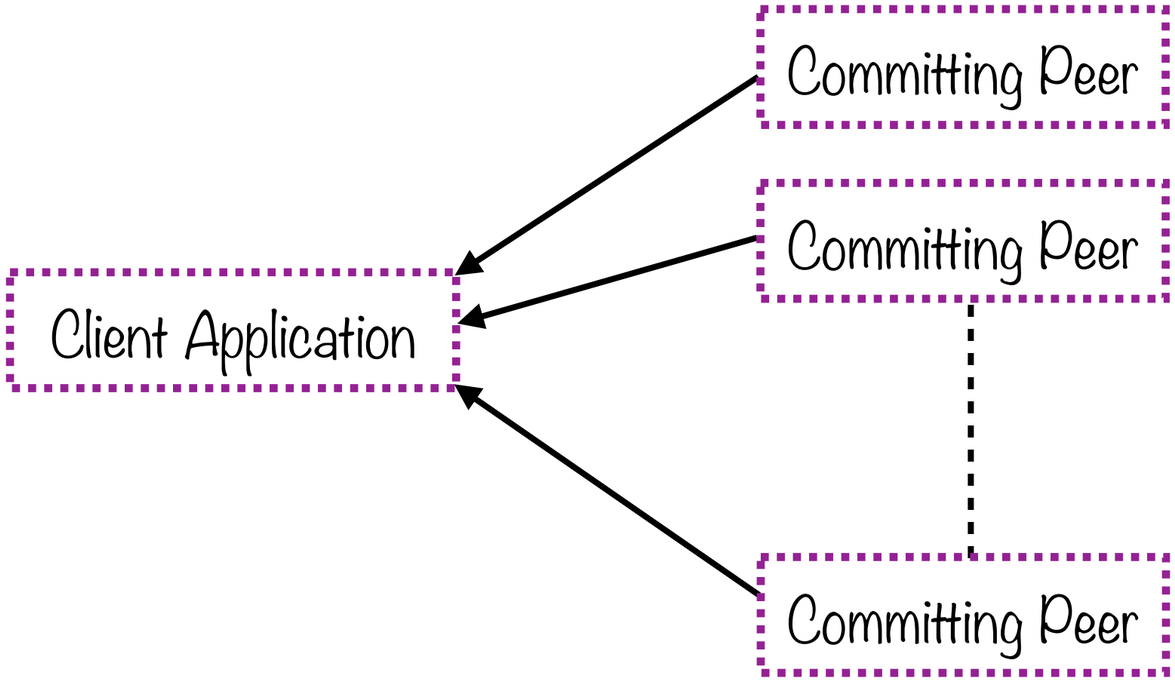
Must process transactions in order

Ensure that endorsement policy satisfied

Check if any previous transaction invalidated this transaction

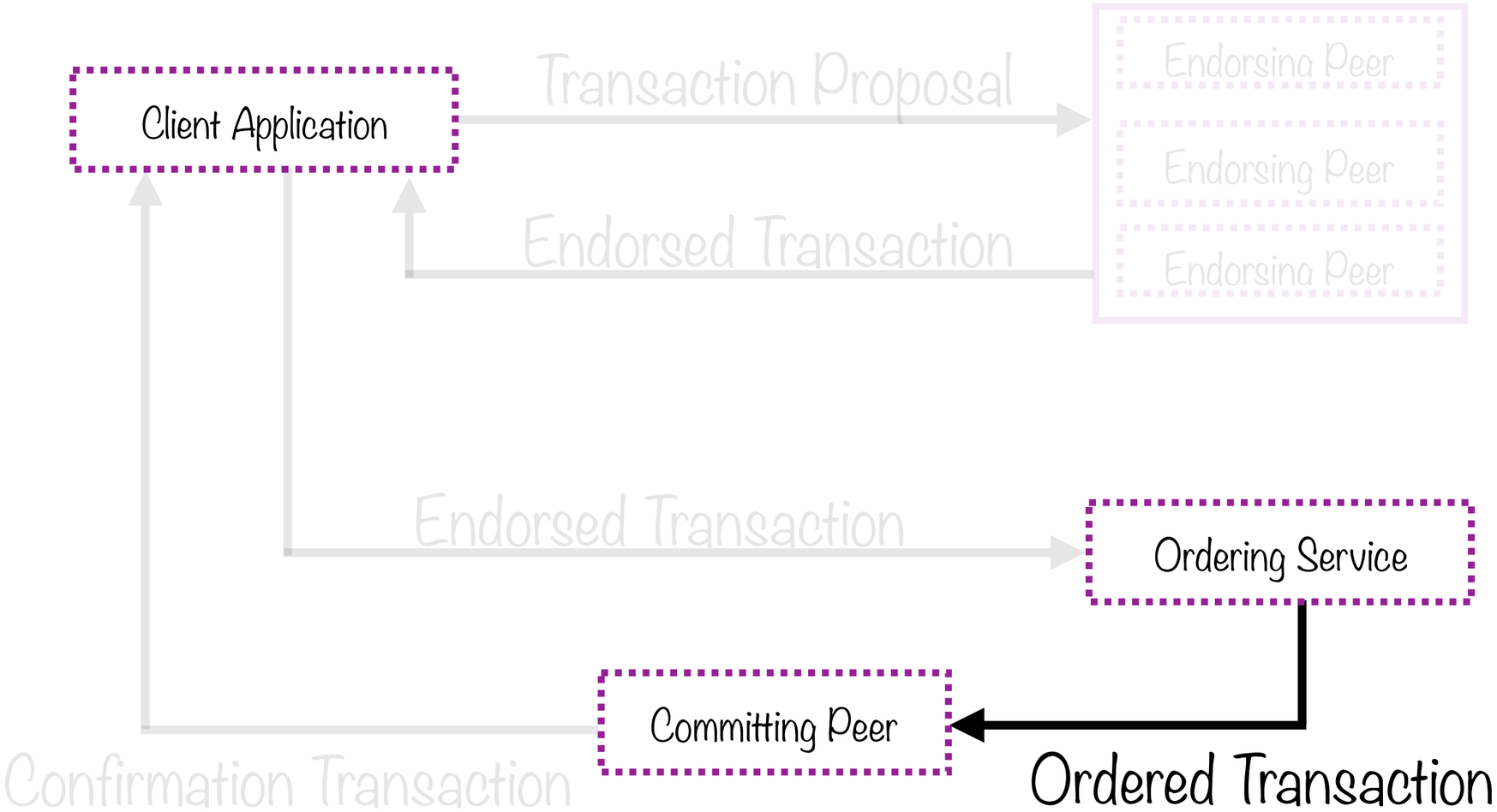
If not, write transaction to ledger

Validation and Commitment

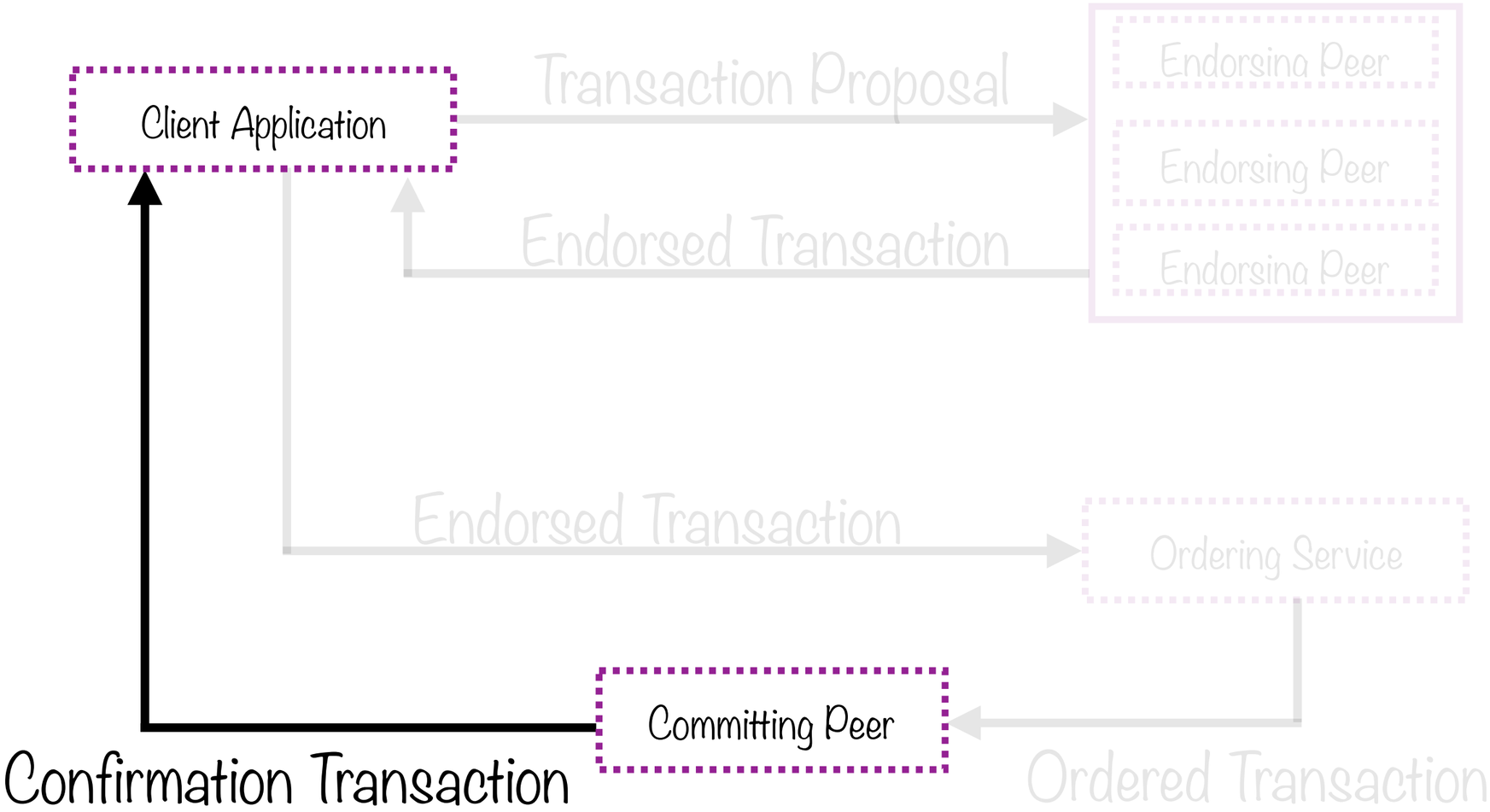


Each committing peer notifies client application of result

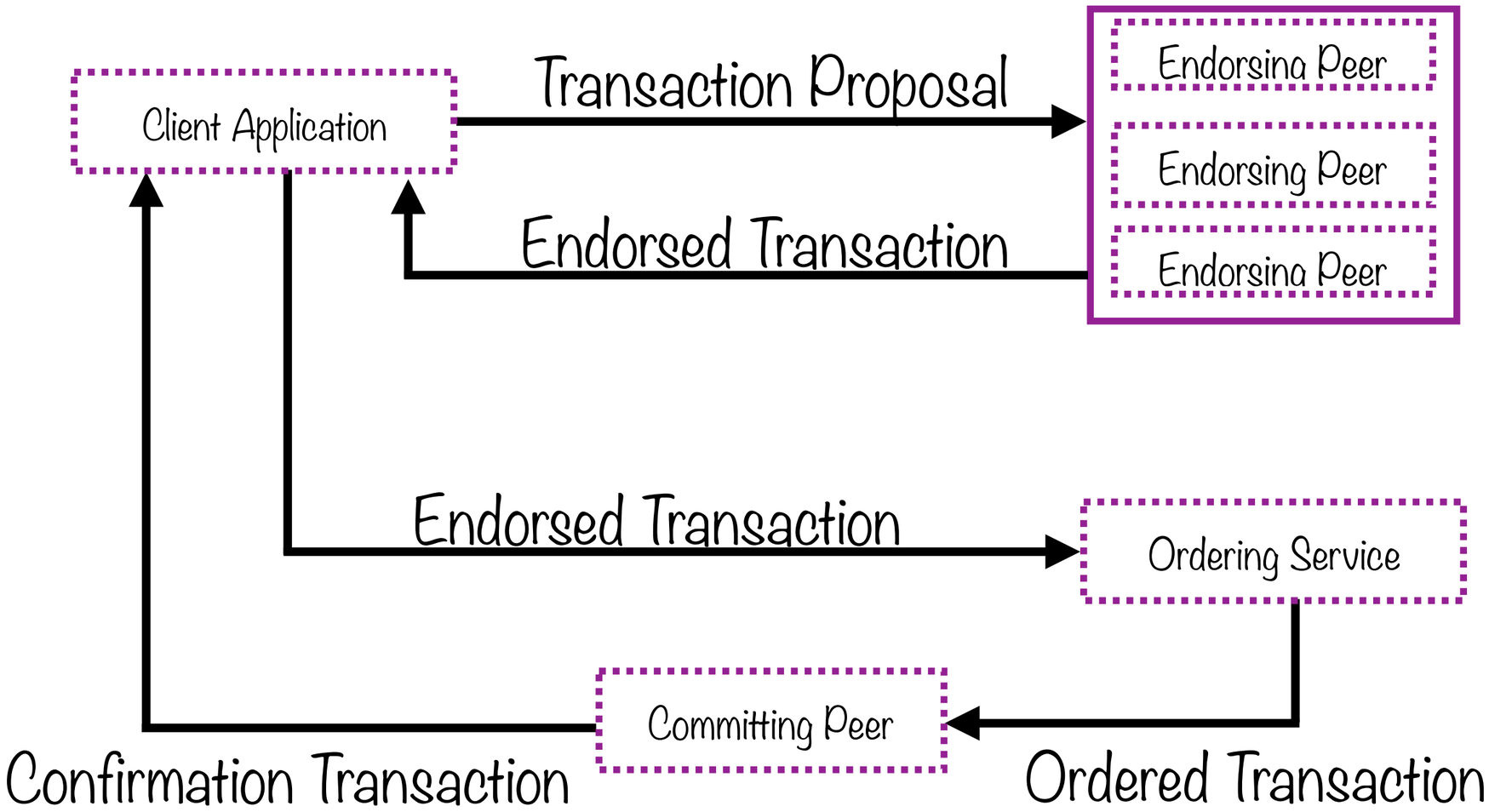
Transaction Flow in Fabric



Transaction Flow in Fabric



Transaction Flow in Fabric



Demo

Writing chaincode

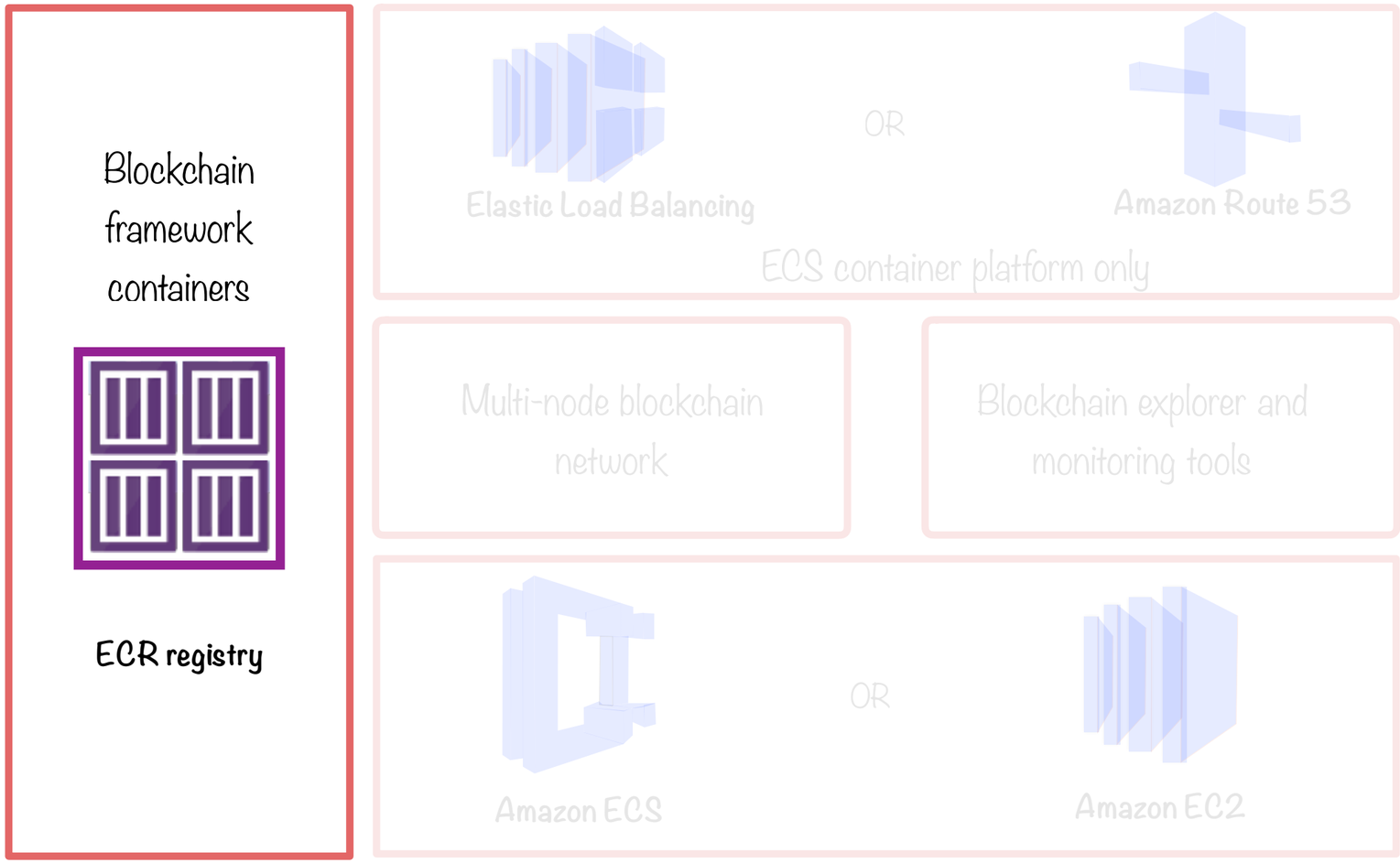
Demo

Deploying chaincode

Demo

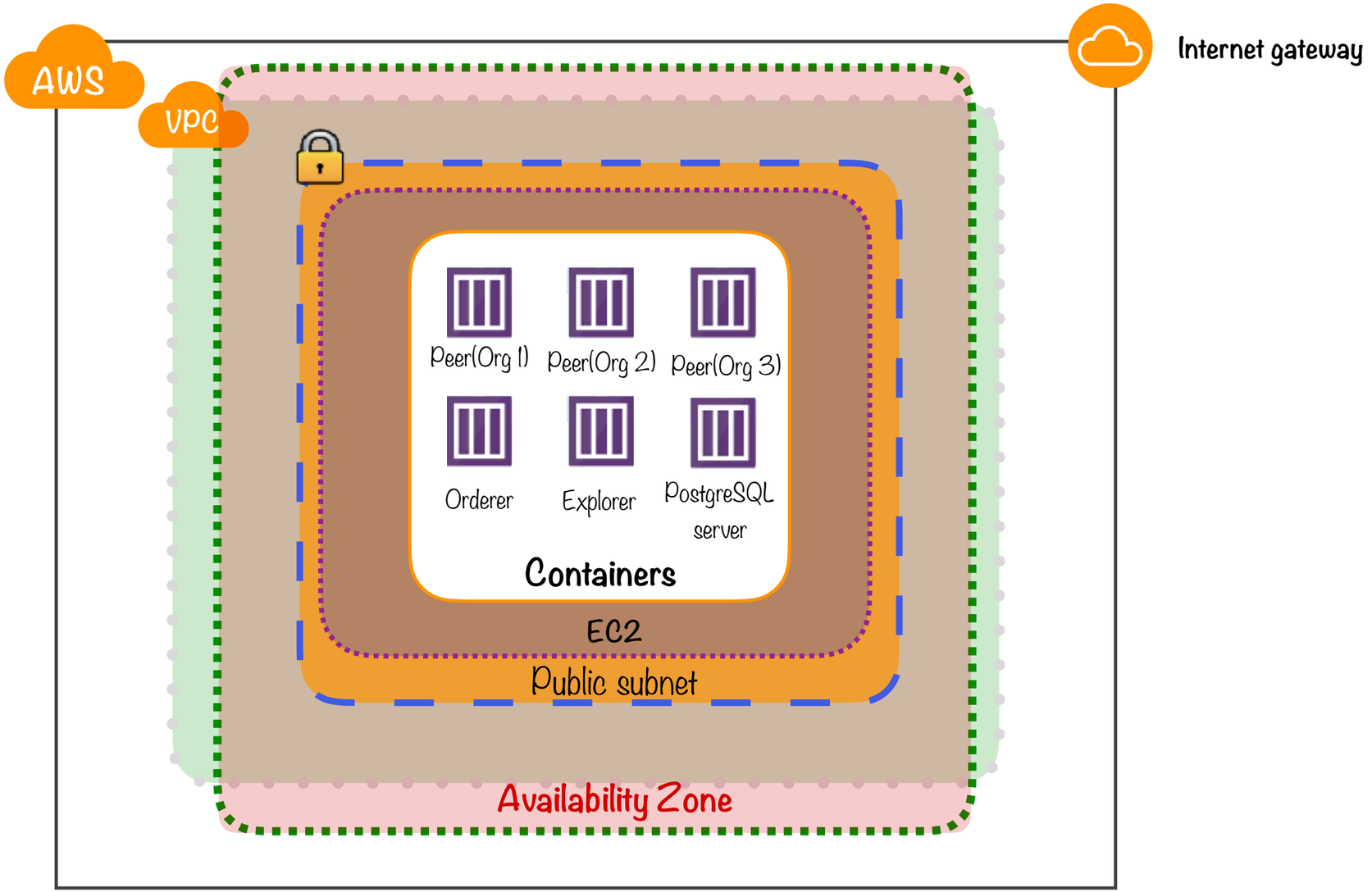
Cleaning up AWS resources

Docker Containers run Blockchain Software

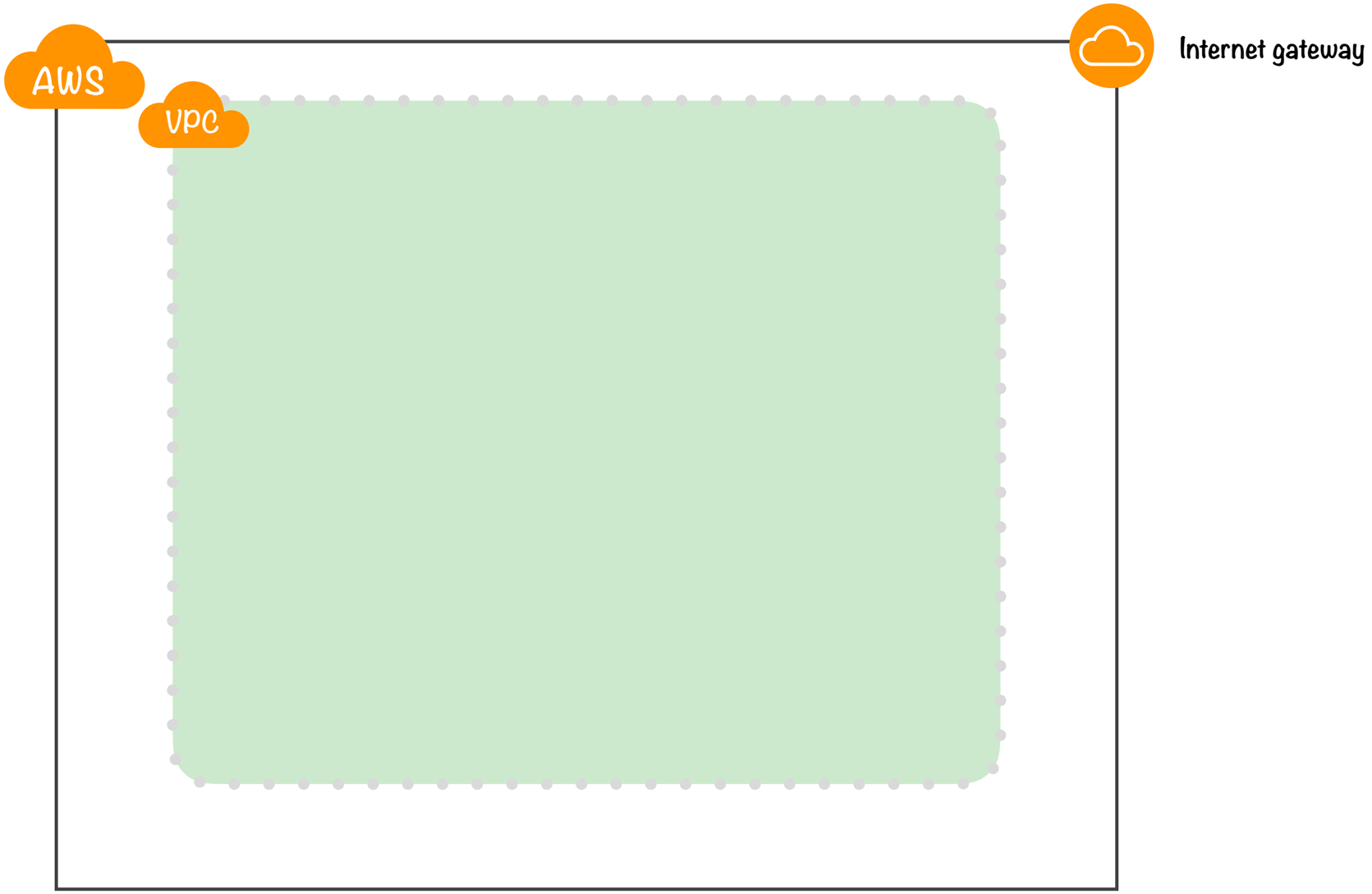


Blockchain software is present on Docker containers

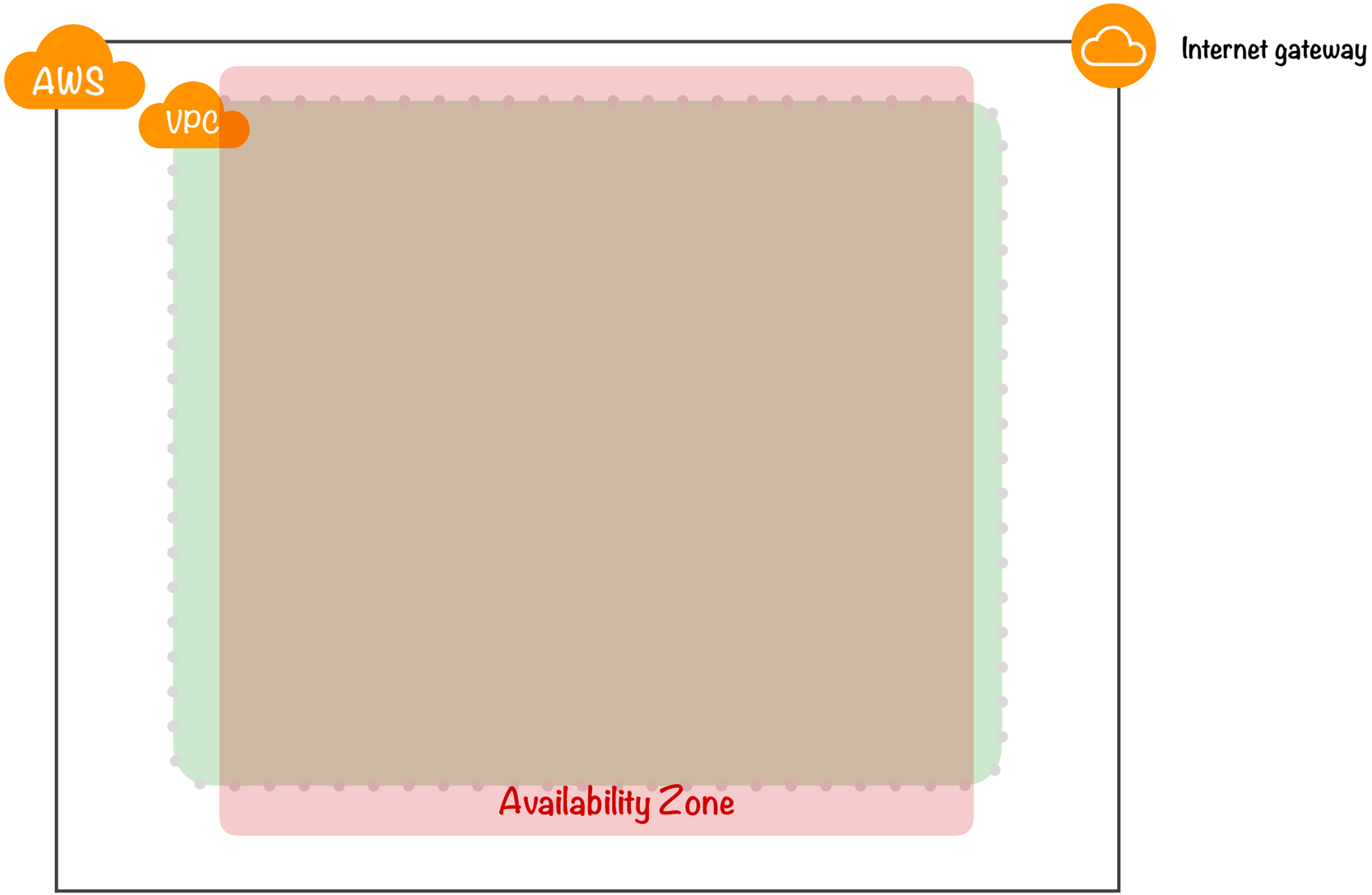
Network Architecture for Hyperledger Fabric



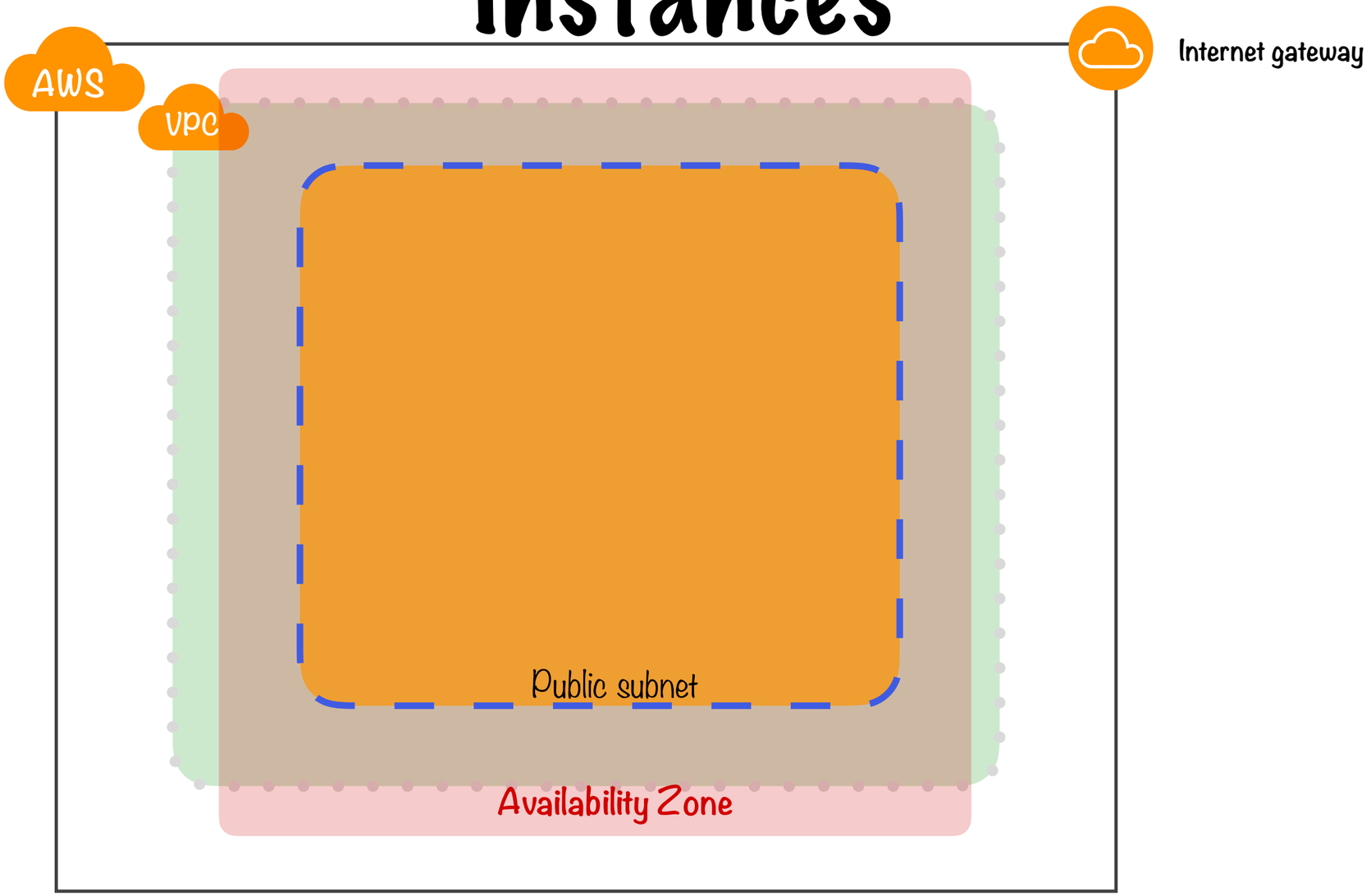
Virtual Private Cloud to Hold Network Resources



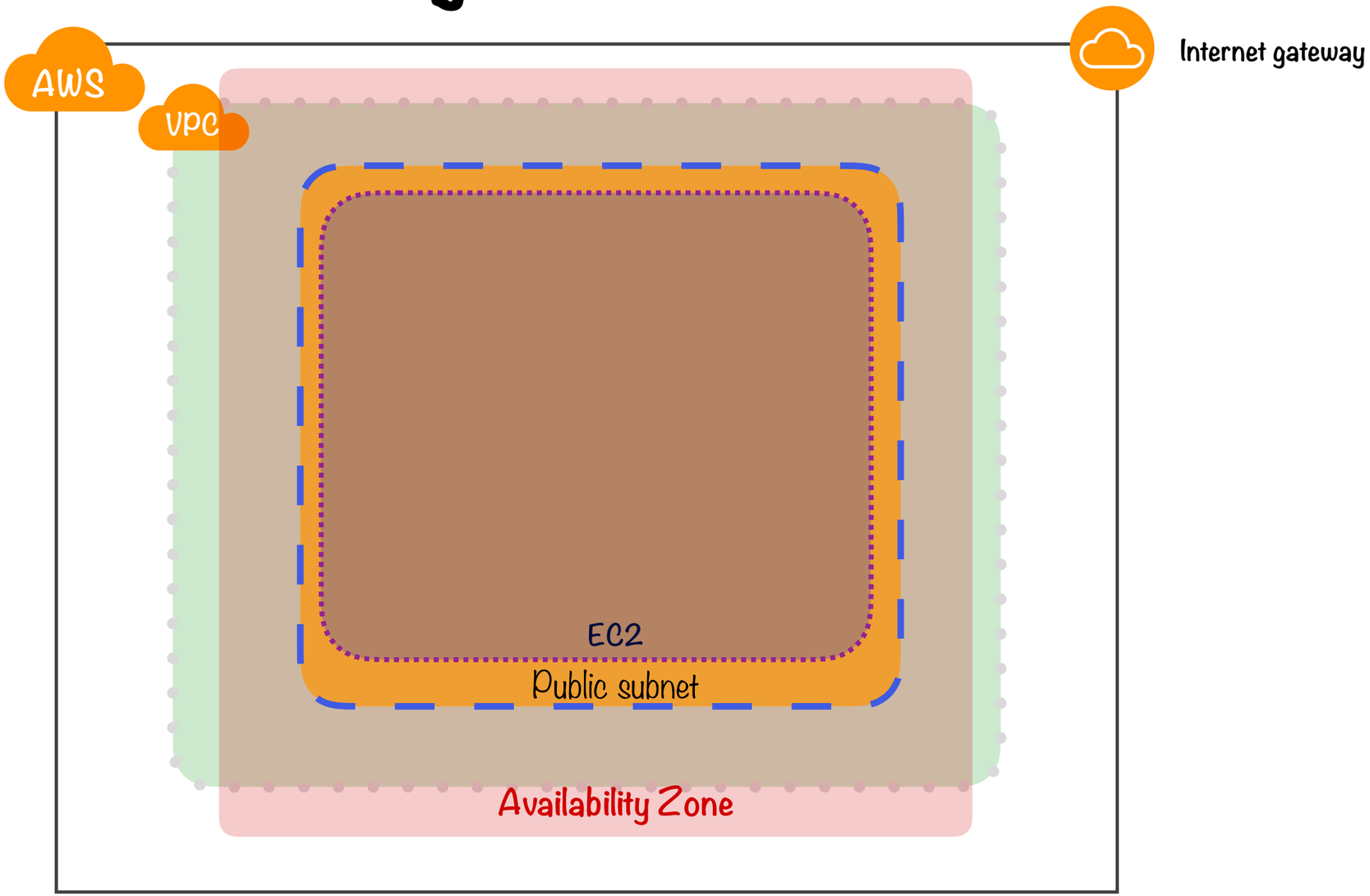
Public Subnet in an Availability Zone



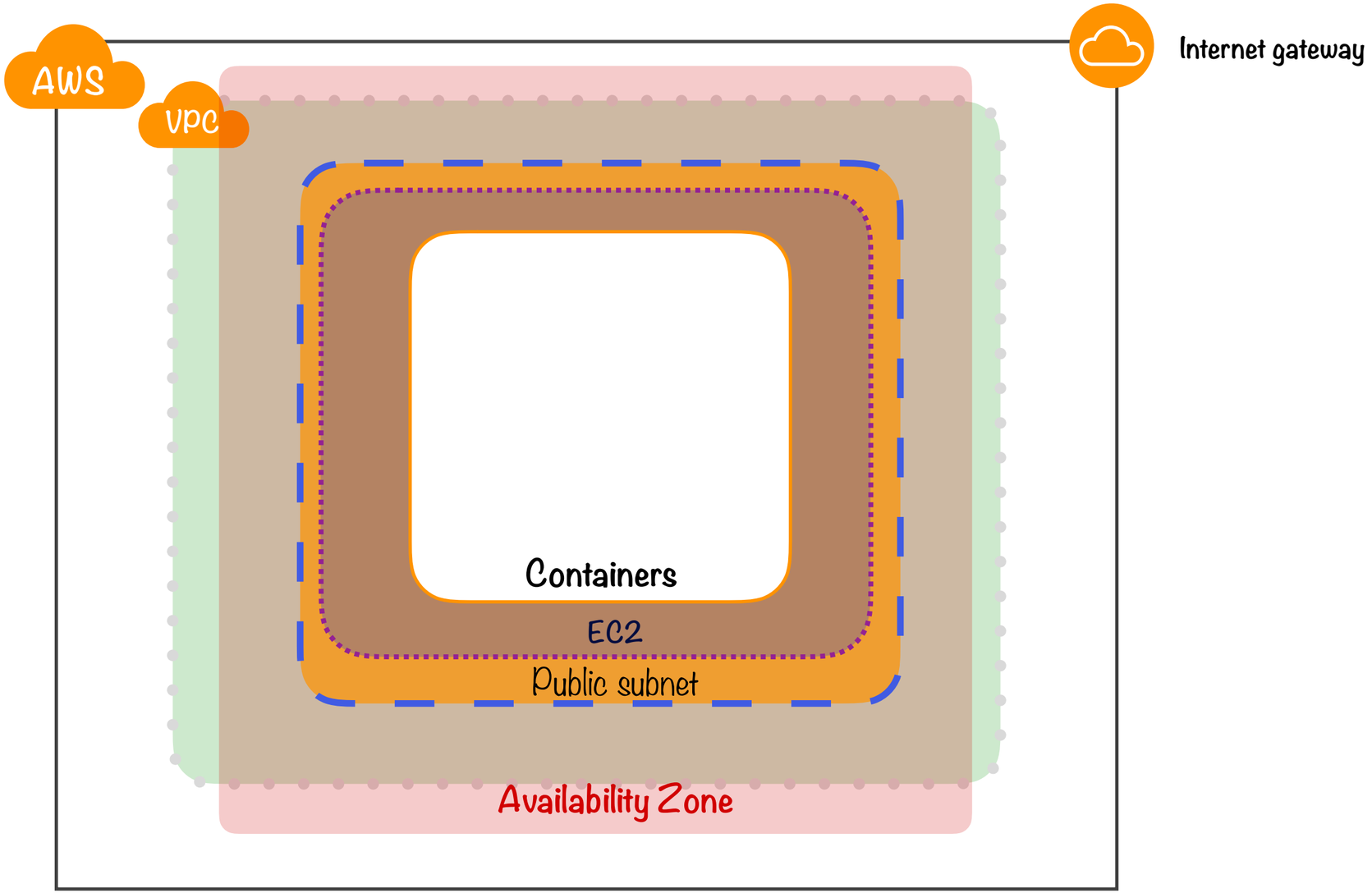
Public Subnet Assigns Public IP Addresses to Instances



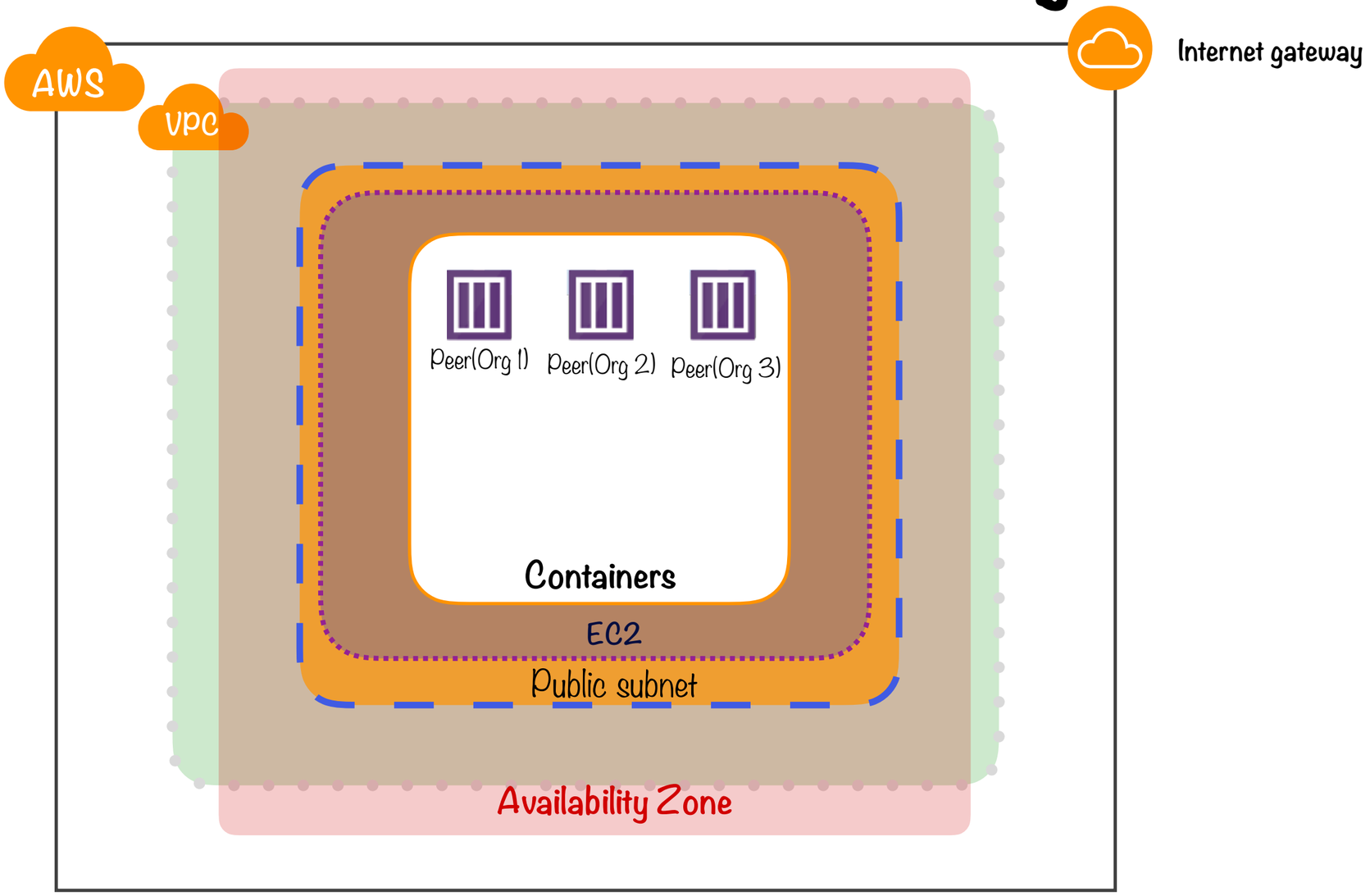
A Single EC2 Instance



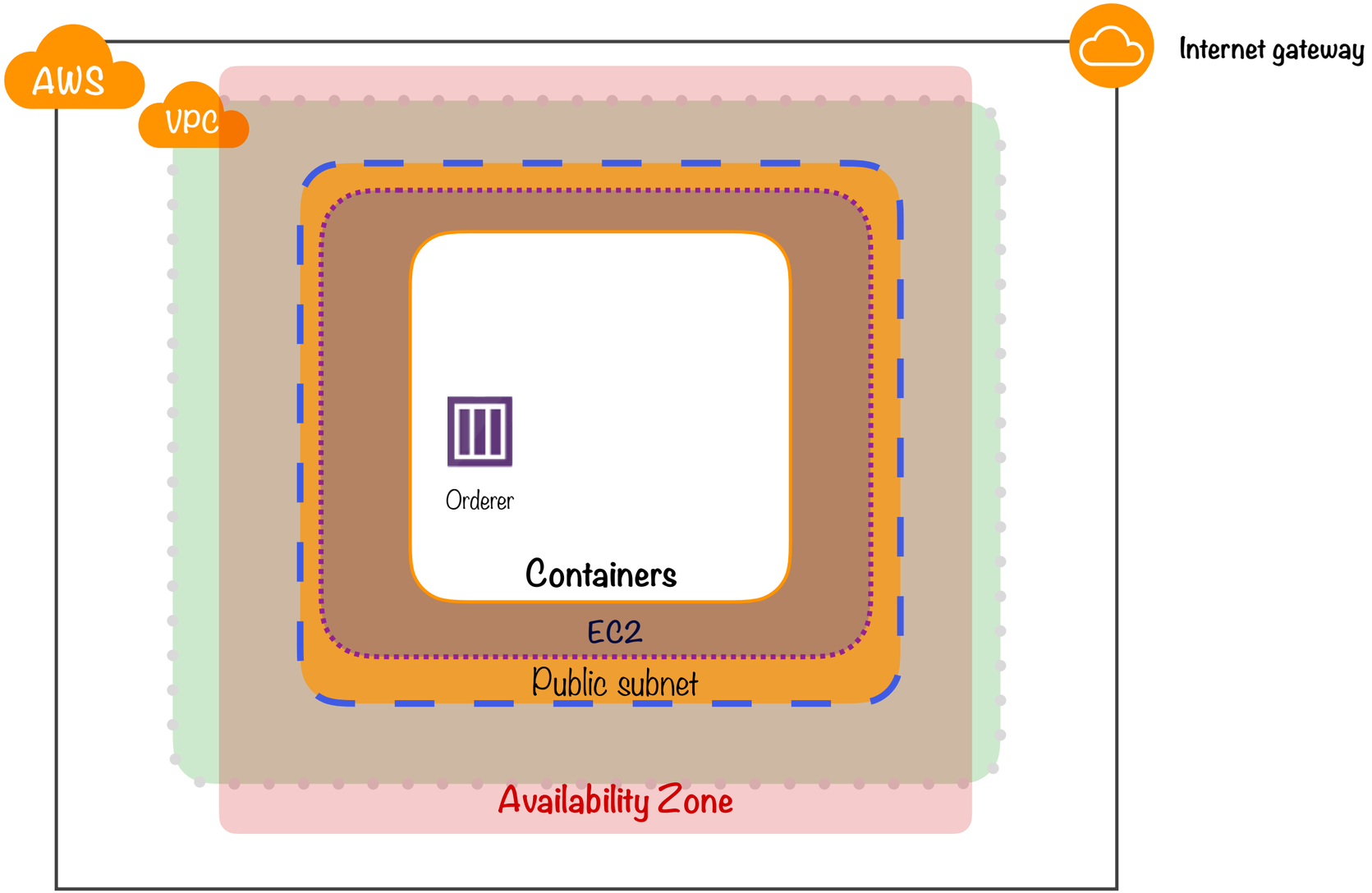
Containers for Nodes in This Instance



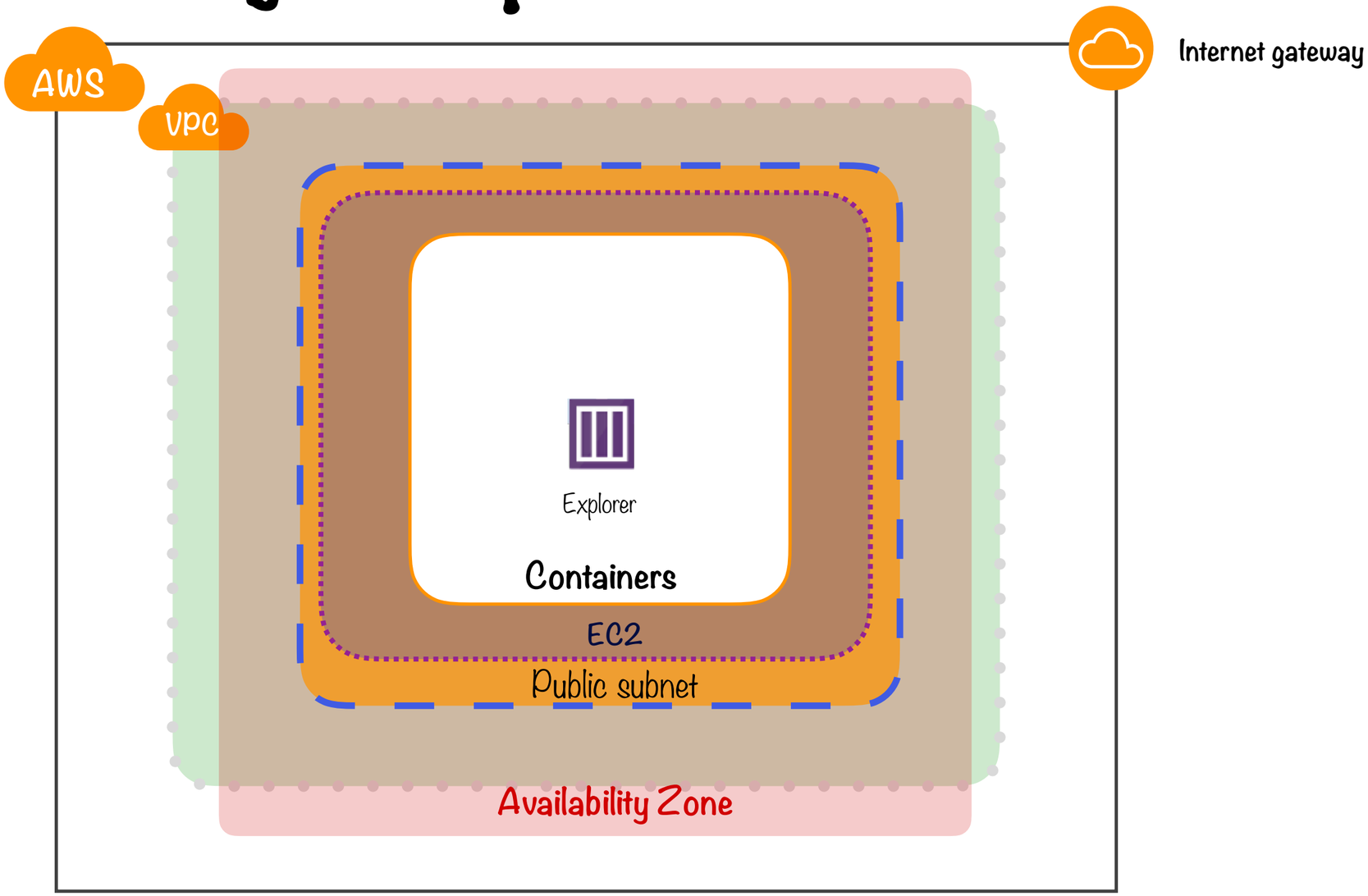
Three Peer Nodes for Three Organizations



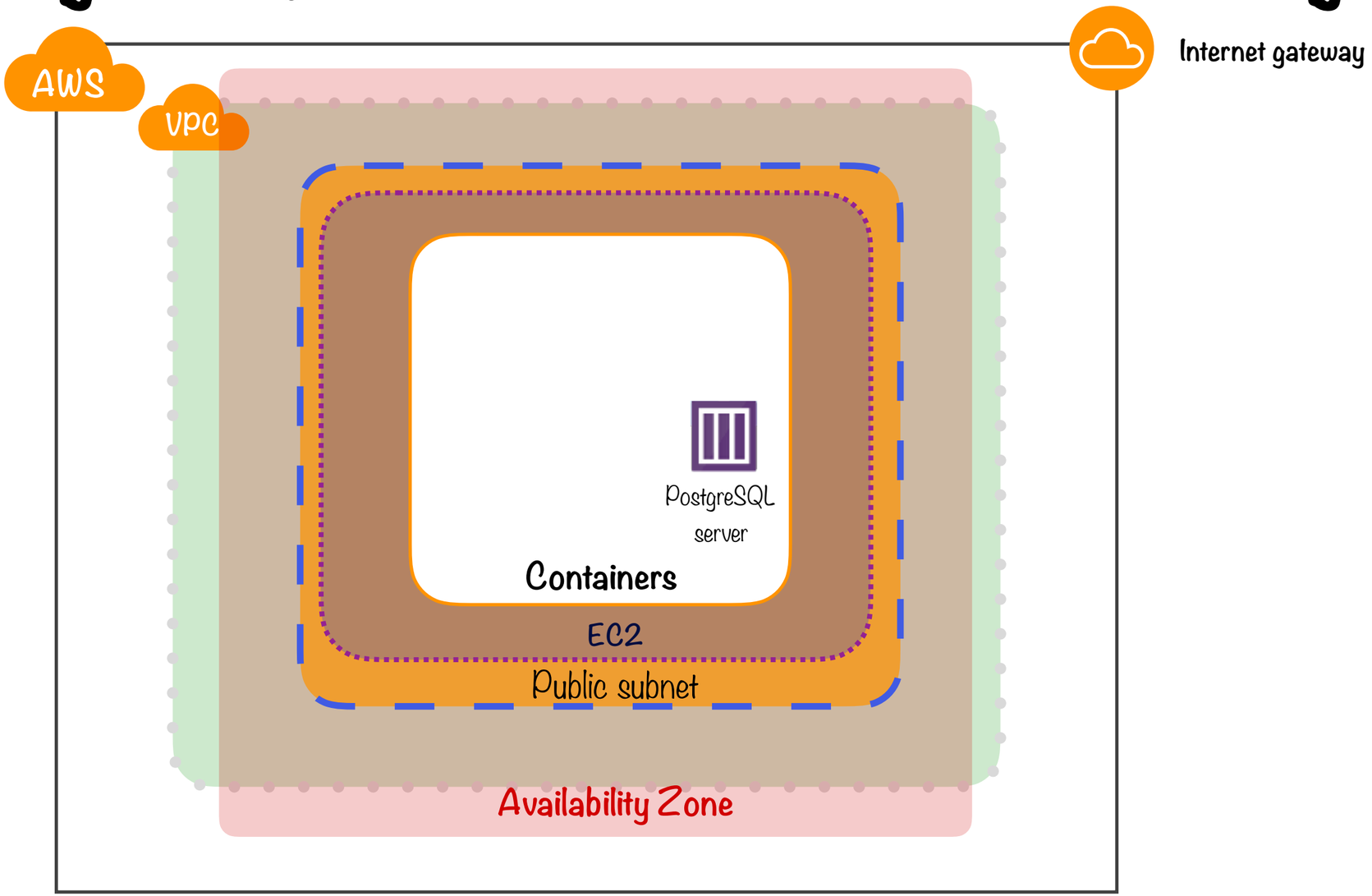
One Orderer Node



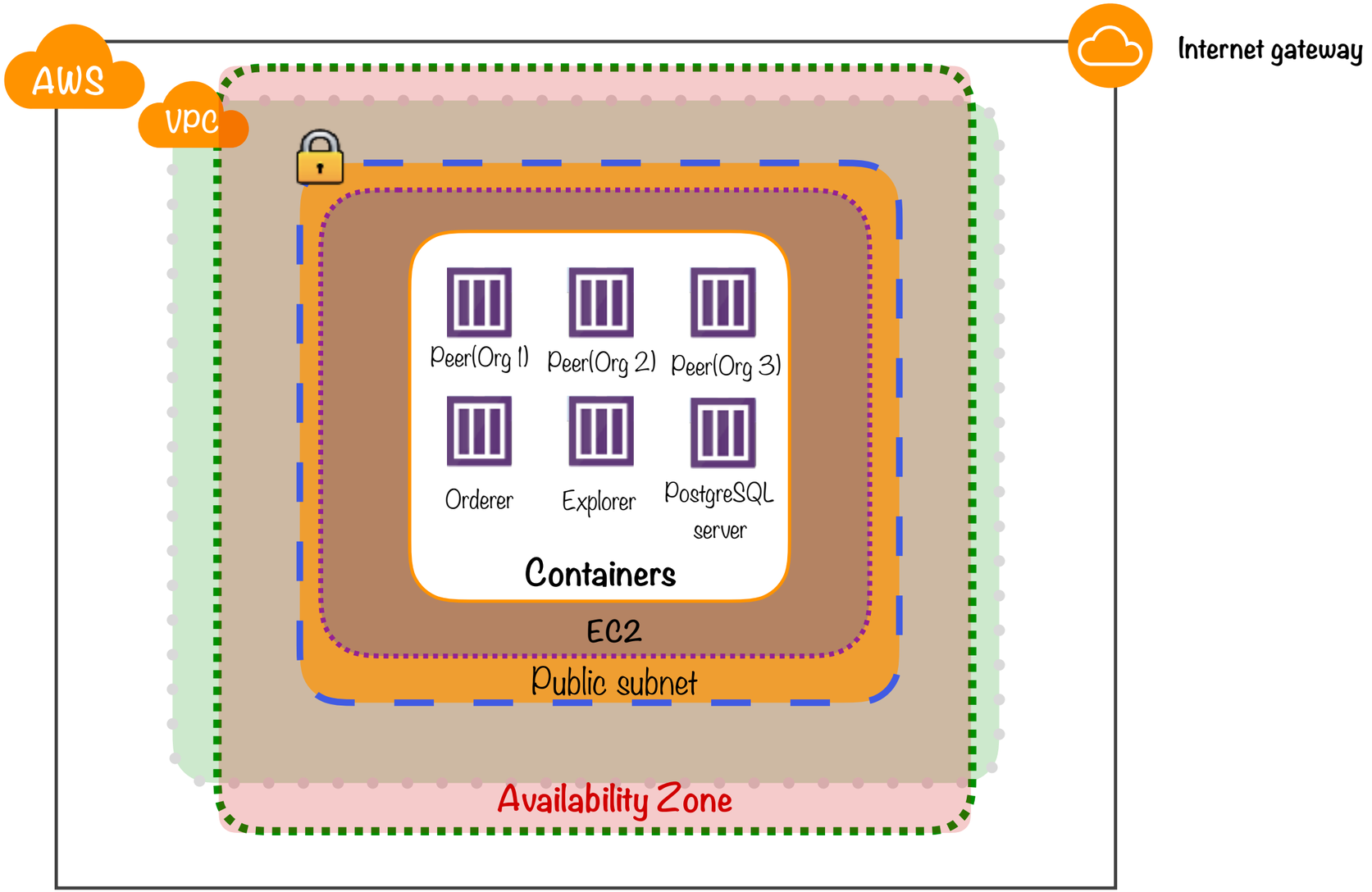
Hyperledger Explorer for Monitoring



PostgreSQL Database to Store Ledger



Network Architecture for Hyperledger Fabric



Summary

A blockchain is a ledger of transactions which is distributed, immutable and verifiable

Hyperledger is an umbrella of open-source blockchain initiatives

Hyperledger Fabric is a mature and stable blockchain framework

Hyperledger Fabric addresses several flaws present in other blockchain implementations

AWS provides templates to quickly provision a Fabric network

Chaincode enables implementation of digital contracts on Fabric