Lessons Learned from running Hyperledger Demos on z/VM Linux

Yongkook(Alex) Kim
Vicom Infinity
April 11th 2017 MVMUA Meeting

What is blockchain and why so much noise?

- You can deploy blockchain demo app yourself on IBM Bluemix in less than an hour without knowing anything about it!
- From zero knowledge on blockchain to running (pre-built) demo apps on local z Linux in 4 months – but almost impossible to learn about everything and updates that come out every day (Hyperledger is the fastest growing open source project on earth)
- There are three major blockchain/Dledger technologies competing but still none of them are on production for enterprises
- There is a reason for public cloud providers servicing blockchain on their infrastructure

What is Hyperledger?

- **Hyperledger** is an open source collaborative effort created to advance **cross-industry blockchain technologies**. It is a global collaboration, hosted by The Linux Foundation, including leaders in finance, banking, IoT, supply chain, manufacturing and technology.

Birth of blockchain and revision by Hyperledger

- Very good YouTube video showing Bitcoin's blockchain concept/demo

→https://www.youtube.com/watch?v=_160oMzblY8

→https://anders.com/blockchain/blockchain.html

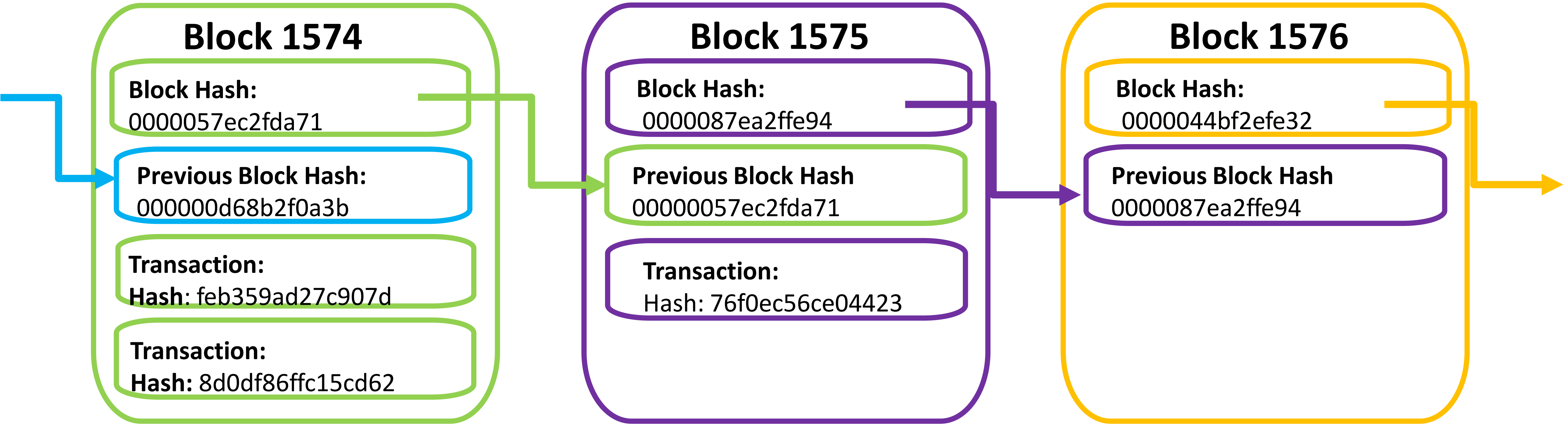
- It was powerful technology and all good but…..

Enterprises also wanted to have:

- Permissioned/private network
- Highly available and maximum security
- High throughput for business applications

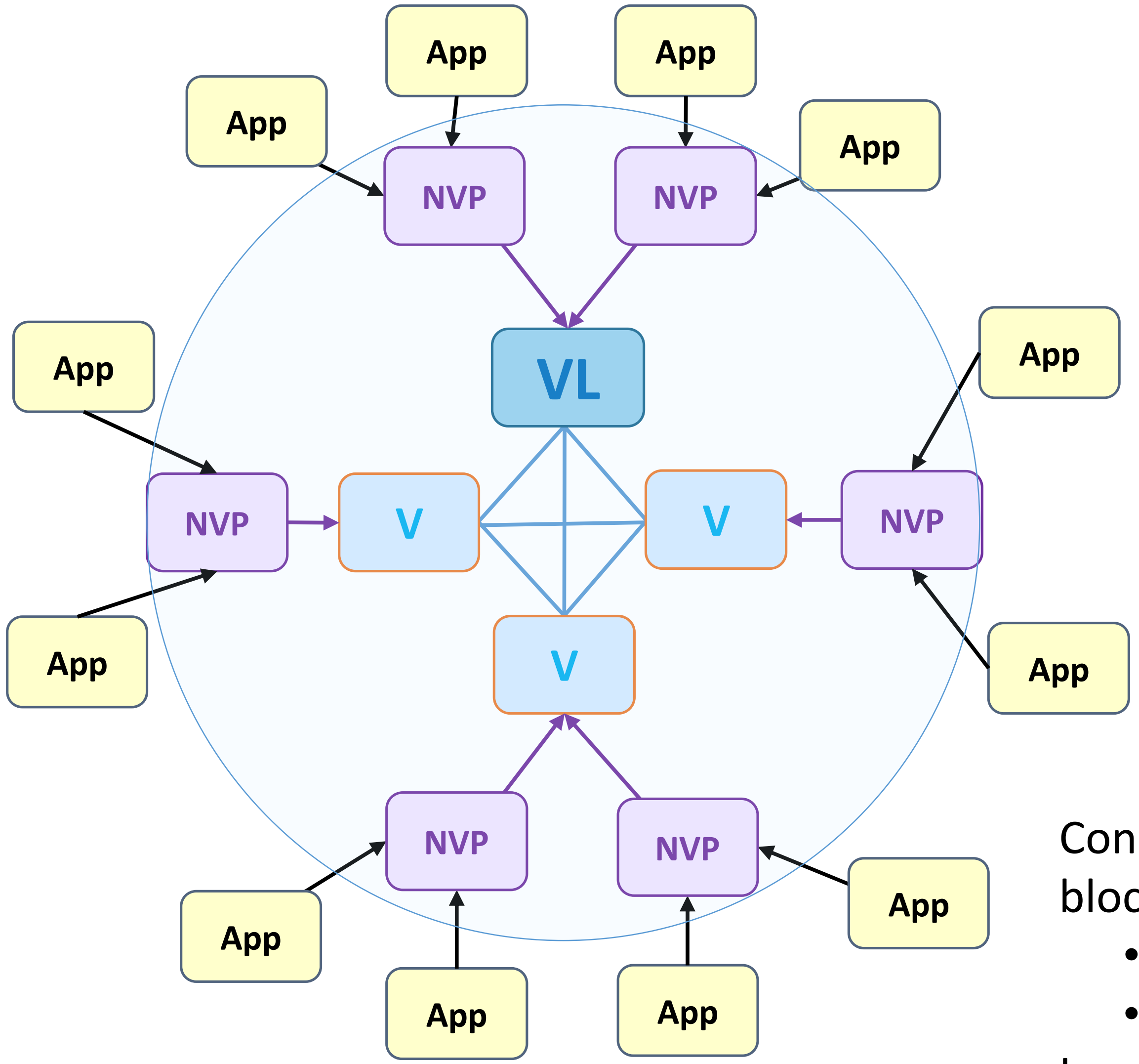
- Therefore new and improved blockchain was introduced with accompanying validation methodologies and application framework – the birth of Hyperledger

This is quick representation of Blockchain


Block 1574
Block Hash: 0000057ec2fda71
Previous Block Hash: 000000d68b2f0a3b
Transaction: **Hash**: feb359ad27c907d
Transaction: **Hash:** 8d0df86ffc15cd62
Block 1575
Block Hash: 0000087ea2ffe94
Previous Block Hash 00000057ec2fda71
Transaction: Hash: 76f0ec56ce04423
Block 1576
Block Hash: 0000044bf2efe32
Previous Block Hash 0000087ea2ffe94

Made up of a series of blocks added in chronological order

Hyperledger Network



VL	V	NVP

	Validating Leader	Validating Peer	Non Validating Peer
Application Connection	✓	✓	✓
Executes Smart Contract	✓	✓	✗
Packages Transactions Into Blocks	✓	✗	✗
Executes Consensus Algorithms	✓	✓	✗

Consensus: Method by which the validating nodes agree to append a block to the chain

- Based on Byzantine Agreement (PBST)
- Open Framework to invite innovation

Immutability: However consensus was achieved, once entered information cannot be easily erased

Enough with high level overview – what’s under the hood?

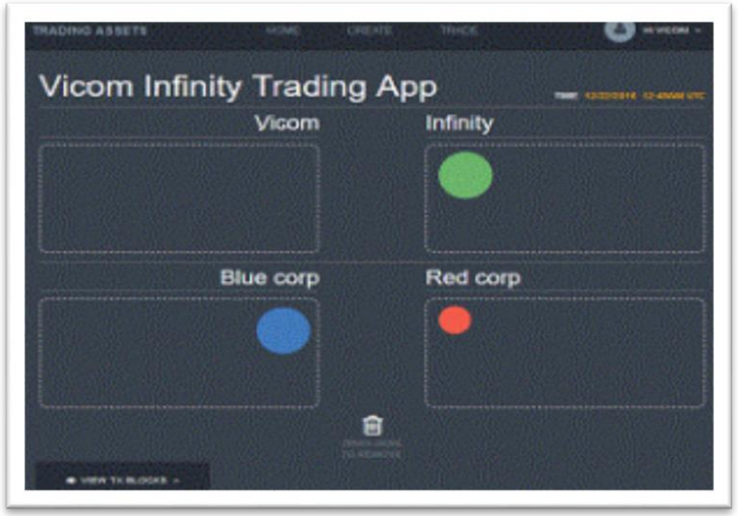
Hyperledger deploys following open source technologies

- GoLang, Java and Node.js - to provide runtime for Chaincode(business logic)
- Docker/Vagrant – to encapsulate chaincode in a secure manner
- gRPC – enables peer-to-peer network
- RocksDB - Persistent state database using a key-value store interface
- PKI/TLS – for certificate authority and secure transmission/authentication

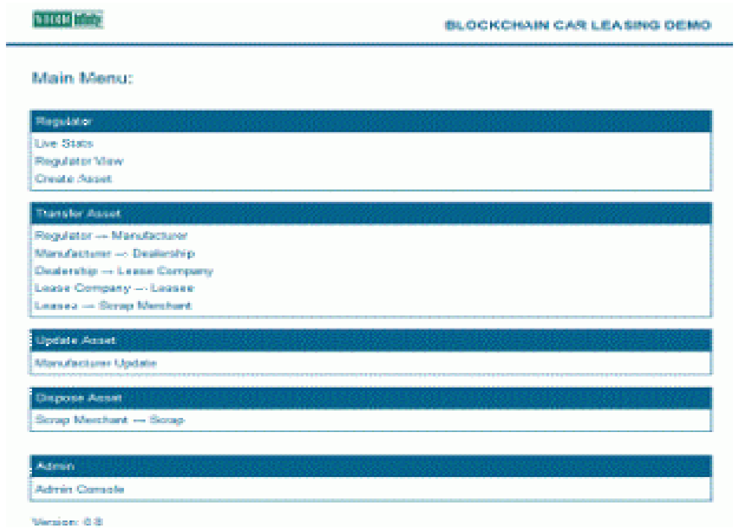
-- very clear architecture description on Hyperledger with references: https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf

Three demo apps I deployed on z Systems

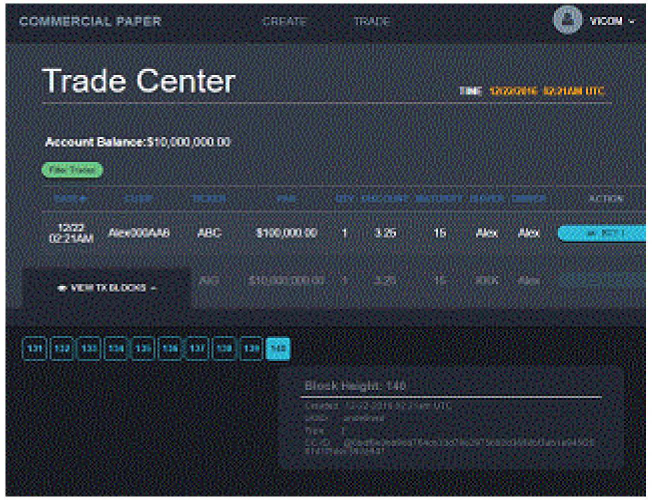
- Marbles App – create& trade marbles and watch how blockchains are created and added on each transaction.



- Car Lease App – a showcase for a car lease industry that how blockchain can provide security and efficiency using it's multi-party distributed ledger.



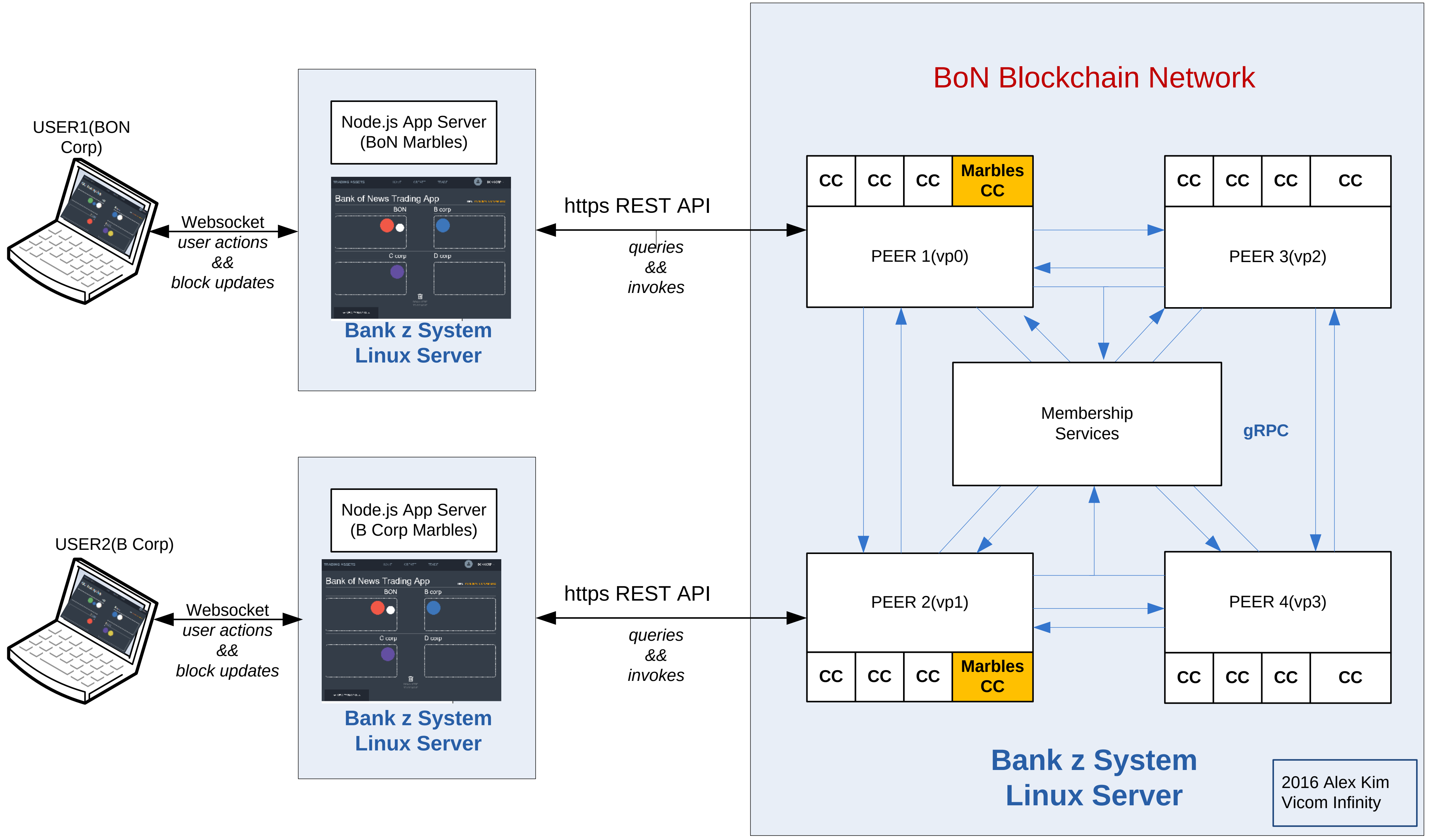
- Commercial Paper App – a simple demonstration of how a commercial paper trading network might be working with blockchain technology.



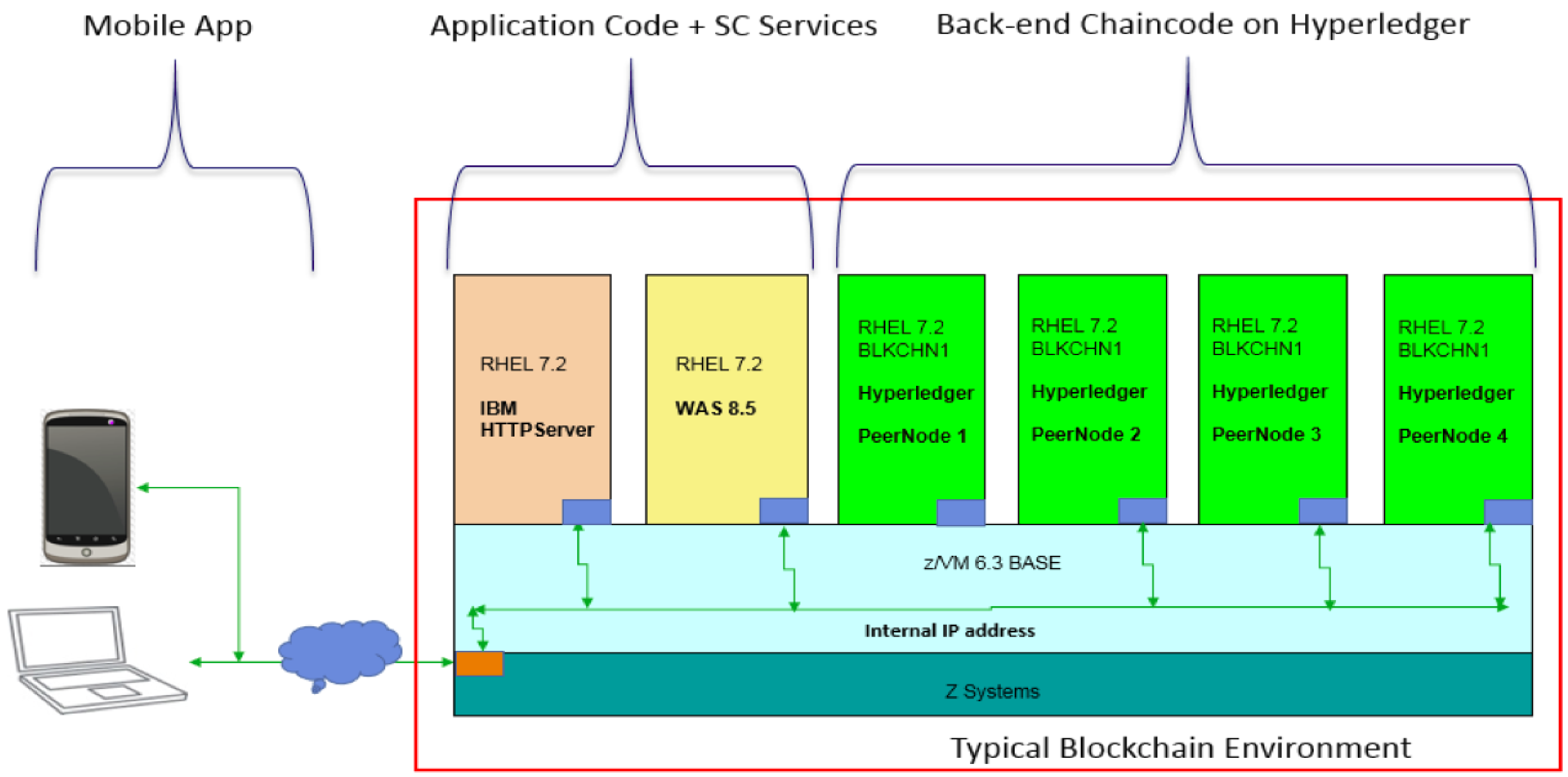
12/22 02:21AM Alex000AA6 ABC $100,000.00 1 3.25 15 Alex Alex

Visit http://blockchain.infinite-blue.com try all of these demo apps and get more information

Marbles demo app sample diagram



High level Infrastructure View



Lesson #1 – use Bluemix fabric first and use docker-compose to launch local fabric

- All of these apps made to run on IBM Bluemix, which contains many predefined Hyperledger fabric definitions
- It's a lot easier to try first with Bluemix as it gives you higher chance to build it successfully
- Once Bluemix model is built, you now have a golden model to compare with when you having issues while building it locally
- If you want to deploy both demo app and Hyperledger fabric on local, try port app first using Bluemix fabric
- Follow instruction on dockerhub to create/run fabric using docker-composer : https://hub.docker.com/r/ibmblockchain/

Lesson #2 – be a good friend with Docker commands

- Launching Hyperledger fabric with docker-compose is quick and easy, but understanding detailed requirements of the fabric and connections can take time
- Start with single peer model and extend to four-peer if required
- Make sure docker images are fresh(not corrupted) and remove any cached images using ‘docker rm –f $(docker ps –a –q)’ when re-launching fabric
- Examine fabric console logs with ‘docker logs -f <containerID>’
- More details on : https://hub.docker.com/r/ibmblockchain/fabric-peer/

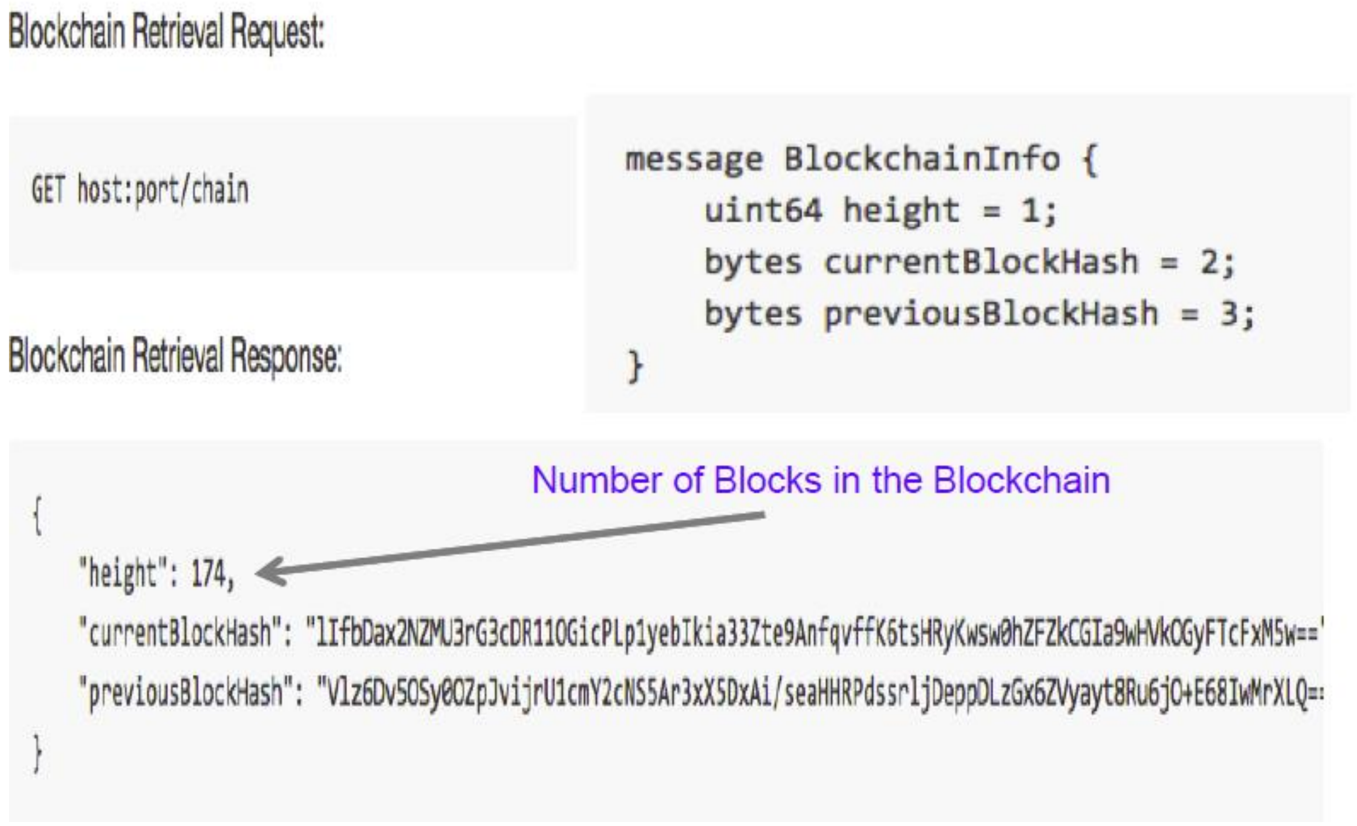
Lesson #3 – understand how local networks are set up

- When deploying Hyperledger fabric and demo apps on local systems, it is key to understand what ports are limited on your network
- If there is any firewalls/proxy between servers, your workstations – make sure what ports/protocols can be passed. Some of the demo apps utilizes WebSocket and some firewalls/proxy may not work well with it. When it happens, try to make SSH tunneling or make HTTPS for the app server
- Test REST API calls for Hyperledger using utilities like PostMan app

Hyperledger REST APIs

- Block
    - GET /chain/blocks/{block-id}
- Blockchain
    - GET /chain
- Chaincode
    - POST /chaincode
- Network
    - GET /network/peers
- Registrar
    - POST /registrar
    - GET /registrar/{enrollmentID}
    - DELETE /registrar/{enrollmentID}
    - GET /registrar/{enrollmentID}/ecert
- Transactions
    - GET /transactions/{UUID}

Example:



Blockchain Retrieval Request:

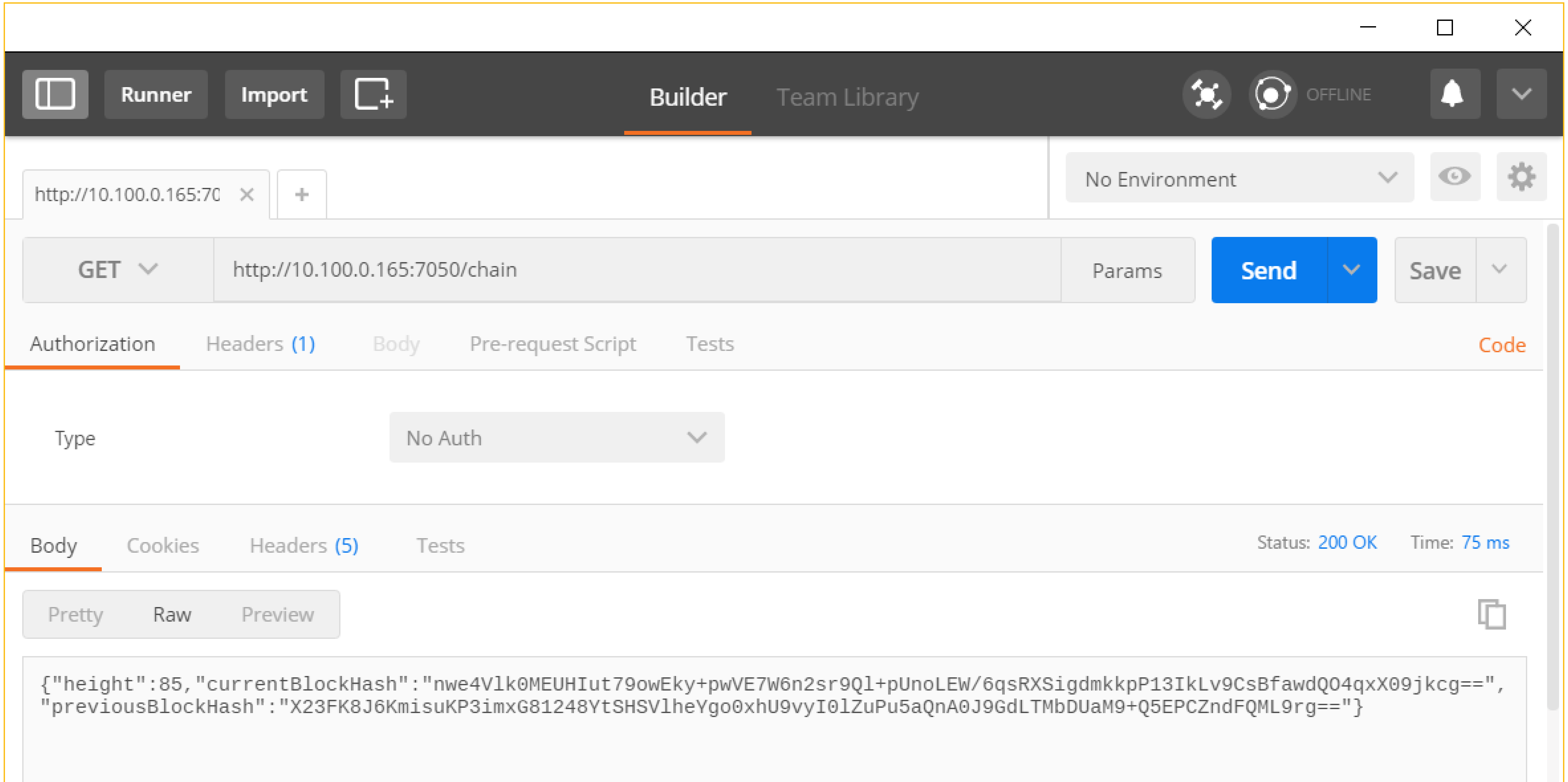
```
GET host:port/chain
```

```
message BlockchainInfo {
    uint64 height = 1;
    bytes currentBlockHash = 2;
    bytes previousBlockHash = 3;
}
```

Blockchain Retrieval Response:

```
{
    "height": 174,
    "currentBlockHash": "lIfbDax2NZMU3rG3cDR11OGicPLp1yebIkia33Zte9AnfqvffK6tsHRyKwsw0hZFZkCGIa9wHVkOGyFTcFxM5w=='
    "previousBlockHash": "Vlz6Dv5OSy0OZpJvijrU1cmY2cNS5Ar3xX5DxAi/seaHHRPdssrljDeppDLzGx6ZVyayt8Ru6jO+E68IwMrXLQ=:
}
```

PostMan example(Chrome App)



GET http://10.100.0.165:7050/chain

Type: No Auth

Status: 200 OK Time: 75 ms

```
{"height":85,"currentBlockHash":"nwe4Vlk0MEUHIut79owEky+pwVE7W6n2sr9Ql+pUnoLEW/6qsRXSigdmkkpP13IkLv9CsBfawdQO4qxX09jkcg==",
"previousBlockHash":"X23FK8J6KmisuKP3imxG81248YtSHSVlheYgo0xhU9vyI0lZuPu5aQnA0J9GdLTMbDUaM9+Q5EPCZndFQML9rg=="}
```

Lesson #4 – connect with who already worked on it

- I got so much help from colleagues at IBM and customers
- Using Slack/github/RocketChat I could get quick responses from many people - https://chat.hyperledger.org
- Special thanks to (Volodymyr Paprotski, John Harrison, Barry Silliman, Dave Huffman)@IBM, (Sam D’Angelo, Vincent Terrone)@AIG

What’s Next?

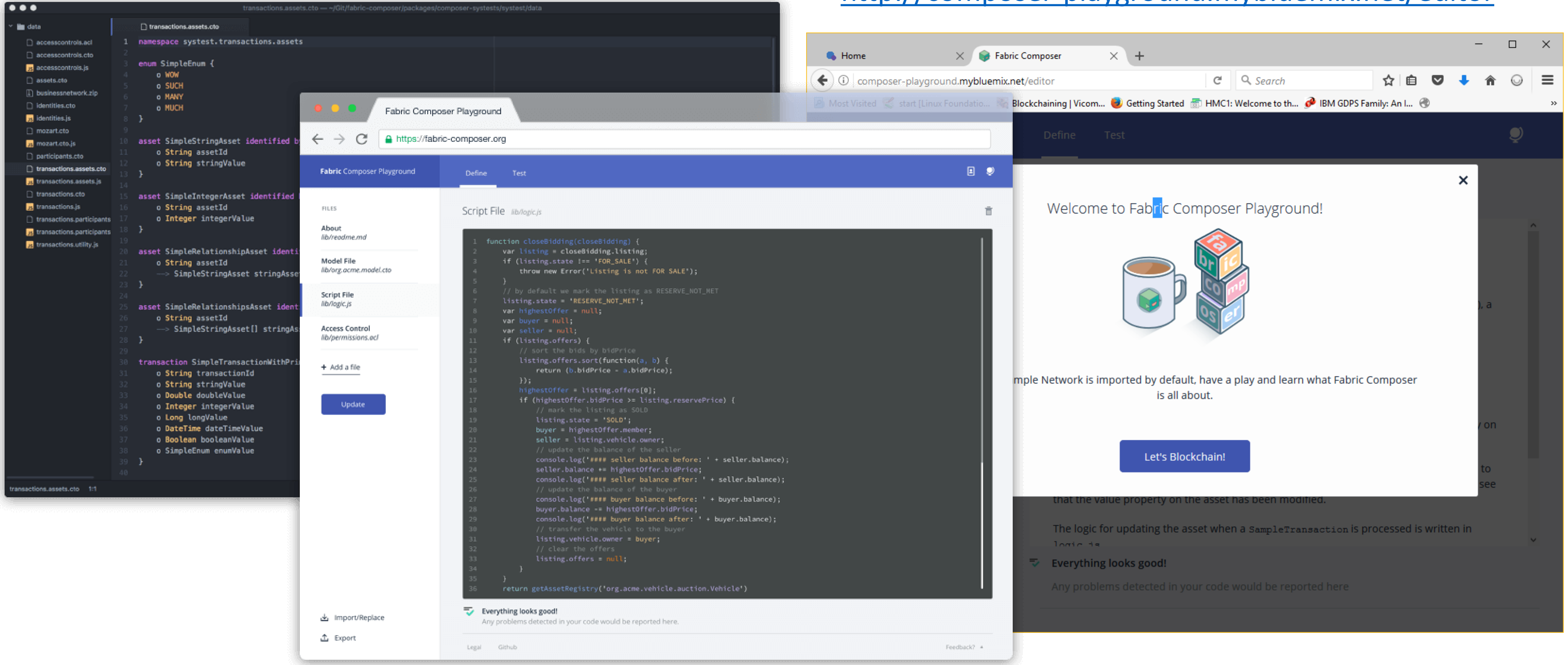
- Hyperledger Fabric v1.0 is announced and available on Bluemix

(limited open beta for HSBN vNext is available now)

- Fabric-composer can be helpful to create app to deploy business logics
- More from z Sysetms to be highlighted supporting Hyperledger
- Hyperledger Hackfest on April 24-25 @McLean, VA

Fabric Composer - https://fabric-composer.github.io/

http://composer-playground.mybluemix.net/editor



```
namespace systest.transactions.assets

enum SimpleEnum {
    o WOW
    o SUCH
    o MANY
    o MUCH
}

asset SimpleStringAsset identified b
    o String assetId
    o String stringValue
}

asset SimpleIntegerAsset identified
    o String assetId
    o Integer integerValue
}

asset SimpleRelationshipAsset identi
    o String assetId
    --> SimpleStringAsset stringAsse
}

asset SimpleRelationshipsAsset ident
    o String assetId
    --> SimpleStringAsset[] stringAs
}

transaction SimpleTransactionWithPri
    o String transactionId
    o String stringValue
    o Double doubleValue
    o Integer integerValue
    o Long longValue
    o DateTime dateTimeValue
    o Boolean booleanValue
    o SimpleEnum enumValue
}
```

Script File lib/logic.js

```
function closeBidding(closeBidding) {
    var listing = closeBidding.listing;
    if (listing.state !== 'FOR_SALE') {
        throw new Error('Listing is not FOR SALE');
    }
    // by default we mark the listing as RESERVE_NOT_MET
    listing.state = 'RESERVE_NOT_MET';
    var highestOffer = null;
    var buyer = null;
    var seller = null;
    if (listing.offers) {
        // sort the bids by bidPrice
        listing.offers.sort(function(a, b) {
            return (b.bidPrice - a.bidPrice);
        });
        highestOffer = listing.offers[0];
        if (highestOffer.bidPrice >= listing.reservePrice) {
            // mark the listing as SOLD
            listing.state = 'SOLD';
            buyer = highestOffer.member;
            seller = listing.vehicle.owner;
            // update the balance of the seller
            console.log('#### seller balance before: ' + seller.balance);
            seller.balance += highestOffer.bidPrice;
            console.log('#### seller balance after: ' + seller.balance);
            // update the balance of the buyer
            console.log('#### buyer balance before: ' + buyer.balance);
            buyer.balance -= highestOffer.bidPrice;
            console.log('#### buyer balance after: ' + buyer.balance);
            // transfer the vehicle to the buyer
            listing.vehicle.owner = buyer;
            // clear the offers
            listing.offers = null;
        }
    }
    return getAssetRegistry('org.acme.vehicle.auction.Vehicle')
```

Everything looks good!
Any problems detected in your code would be reported here.

Welcome to Fabric Composer Playground!

mple Network is imported by default, have a play and learn what Fabric Composer is all about.

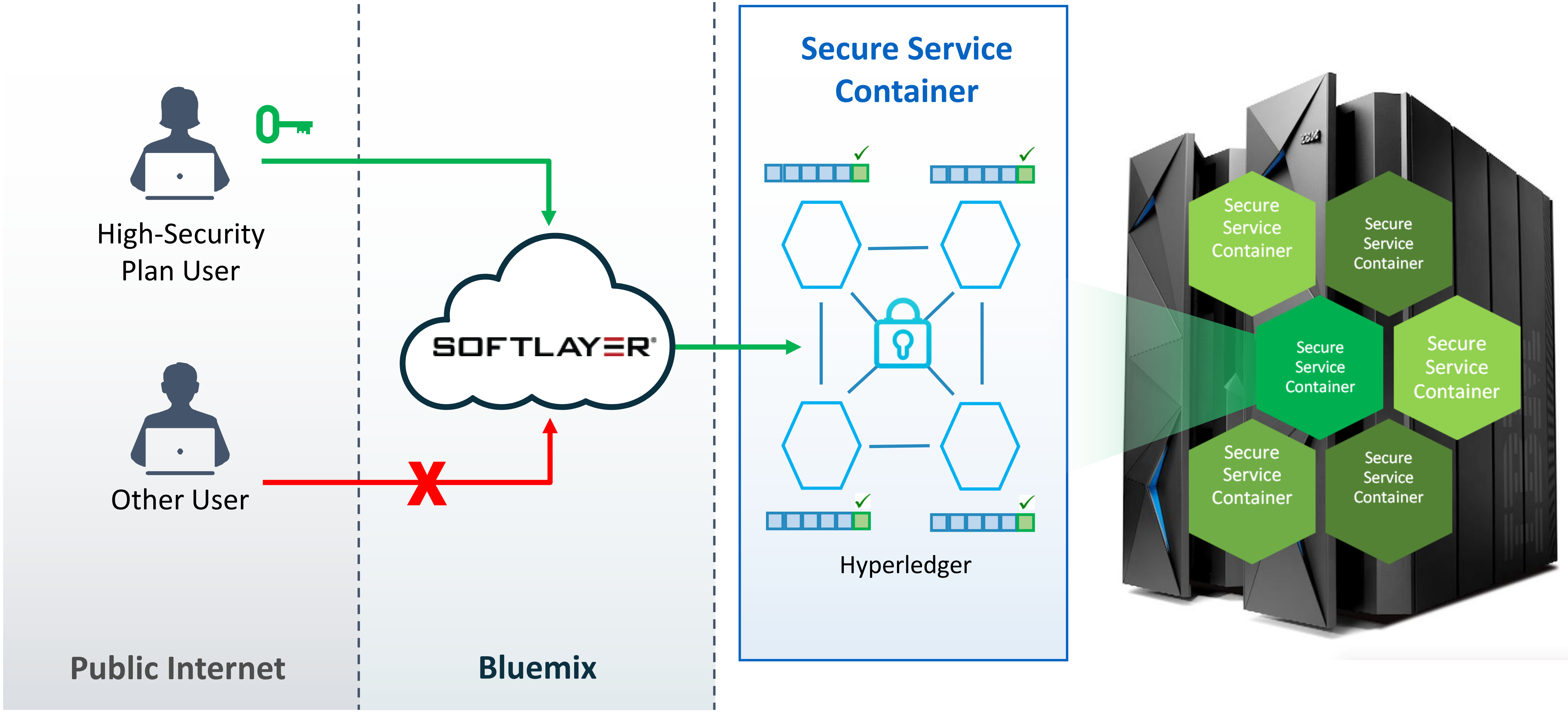
The logic for updating the asset when a `SampleTransaction` is processed is written in

Demo Time?

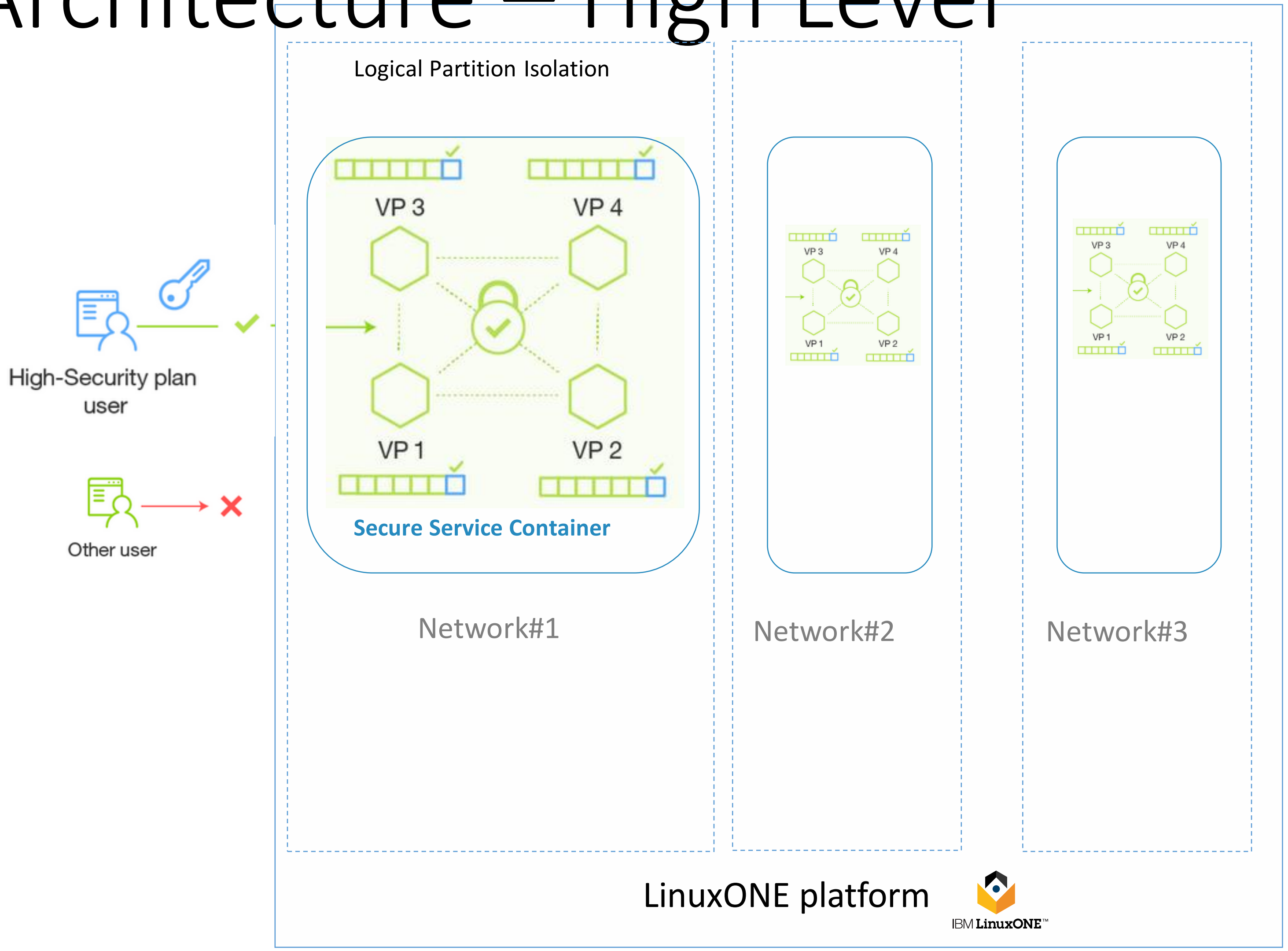
- Please go to http://blockchain.infinite-blue.com
- Click on Blockchain Demo Apps

BACK UP Charts

Architecture – Overview



Architecture – High Level

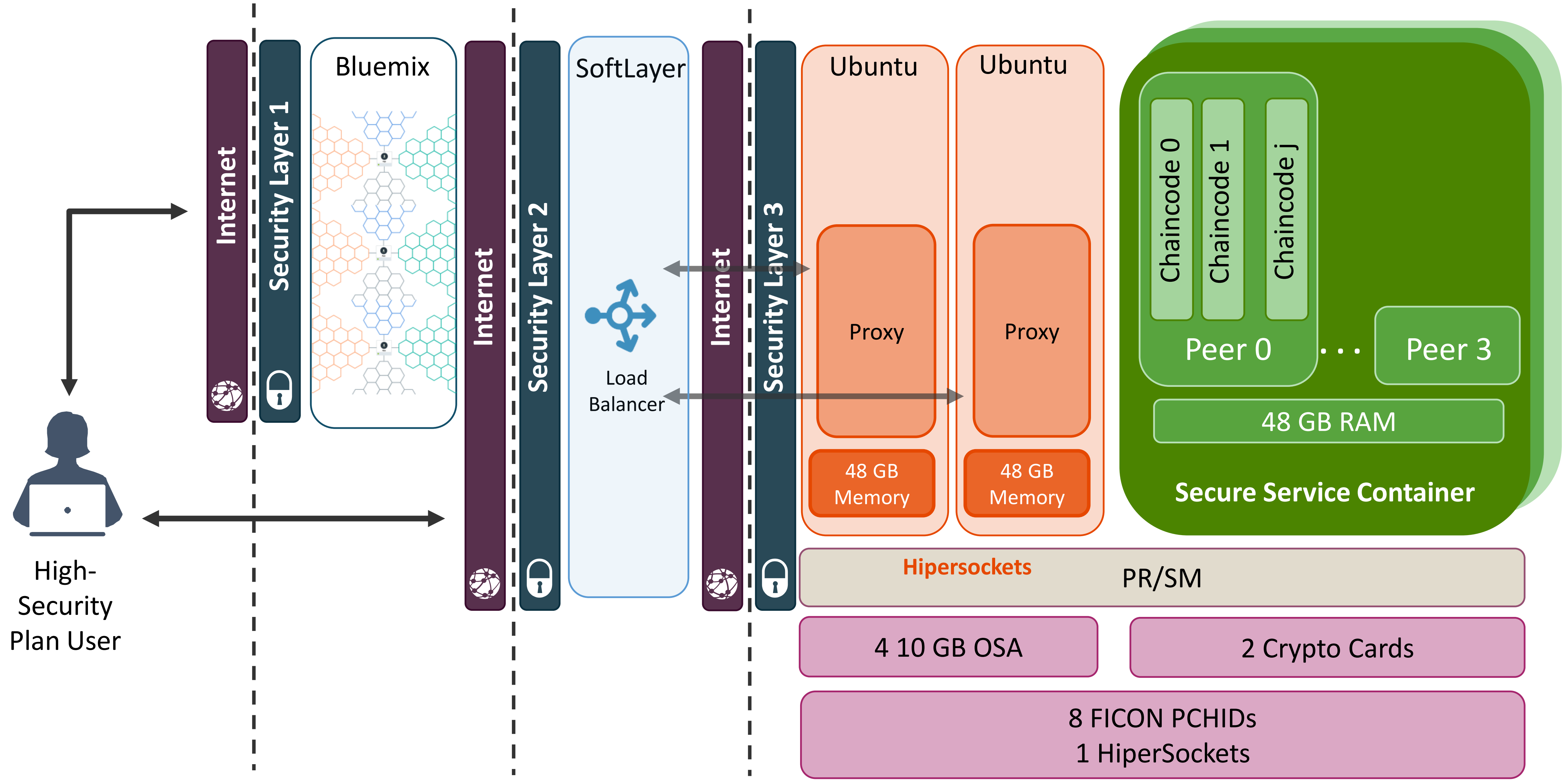


The high security business network is deployed as an appliance into a Secure Service Container, which provides the base infrastructure for hosting blockchain services. The appliance combines operating systems, Docker, middleware, and software components that work autonomously to provide core services and infrastructure with optimized security.

Overview: https://console.ng.bluemix.net/docs/services/blockchain/etn_ssc.html

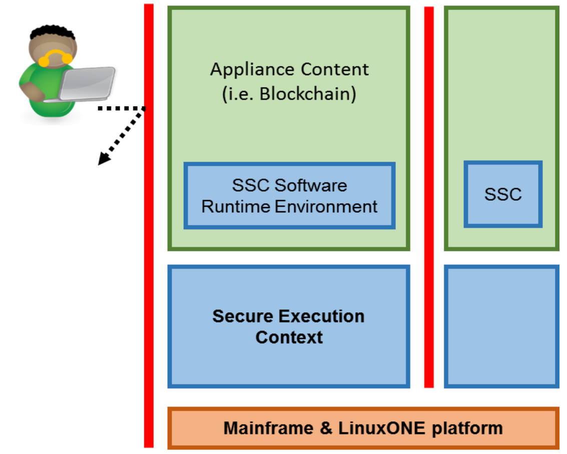
Reference Architecture

High Security Business Network



Secure Service Container

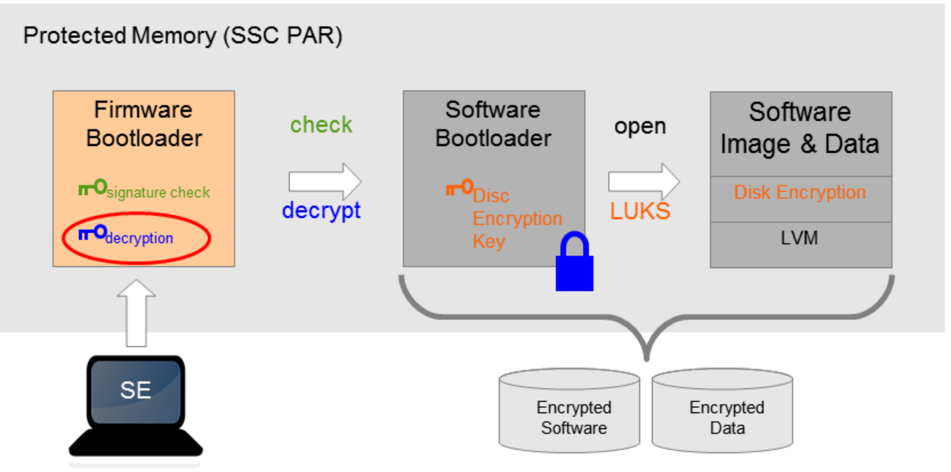
Secure Service Container ensures…


Appliance Content (i.e. Blockchain)
SSC Software Runtime Environment
SSC
Secure Execution Context
Mainframe & LinuxONE platform

No system admin access, ever

- Once the appliance image is built, OS access (ssh) is not possible
- Only Remote APIs available
- Memory access disabled
- Encrypted disk
- Debug data (dumps) encrypted

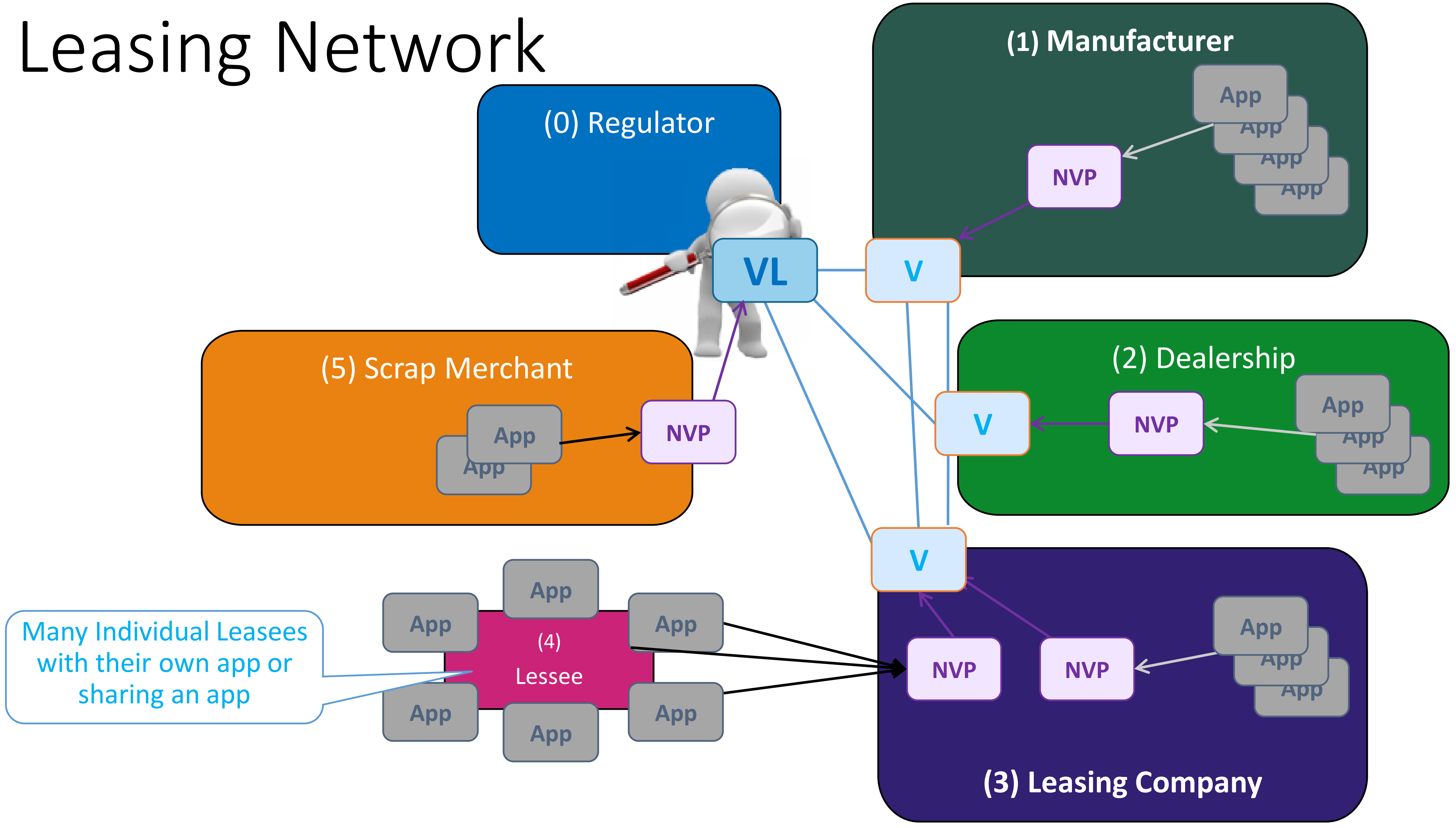
How the Secure Service Container boot sequence works…



Boot sequence

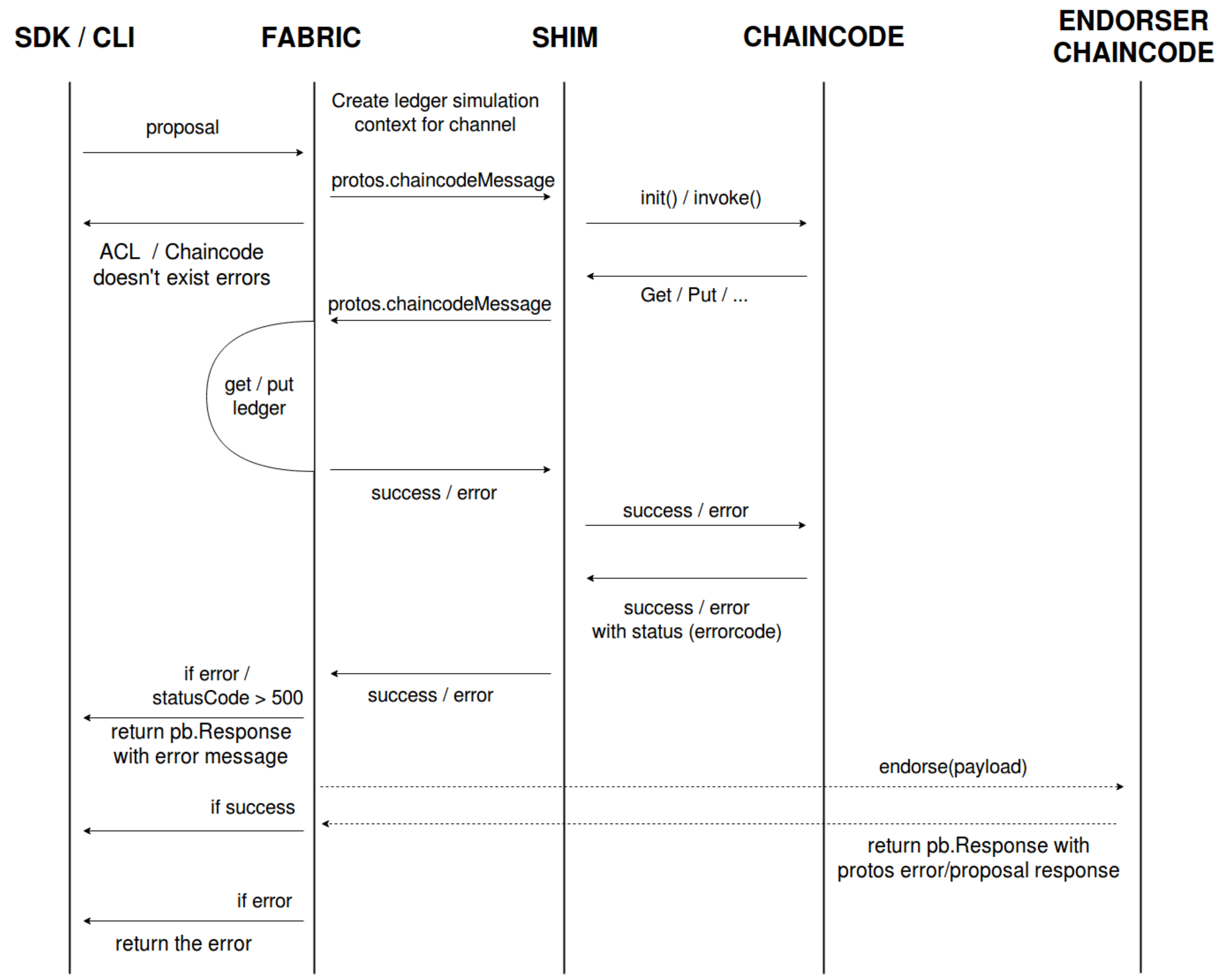
1. Firmware bootloader is loaded in memory
2. Firmware loads the software bootloader from disk
   1. Check integrity of software bootloader
   2. Decrypt software bootloader
3. Software bootloader activate encrypted disks
   1. Key stored in software bootloader (encrypted)
   2. Encryption/decryption done on the flight when accessing appliance code and data

Car Leasing Network



How is Hyperledger Fabric different from other blockchain implementations?

	Bitcoin	Ethereum	Hyperledger
Cryptocurrency required	bitcoin	ether, user-created cryptocurrencies	none
Network	public	public or permissioned	permissioned
Transactions	anonymous	anonymous or private	public or confidential
Consensus	proof of work	proof of work	PBFT
Smart contracts (business logic)	none	yes (Solidity, Serpent, LLL)	yes (chaincode)
Language	C++	Golang, C++, Python	Golang, Java


SDK / CLI
FABRIC
SHIM
CHAINCODE
ENDORSER CHAINCODE
proposal
Create ledger simulation context for channel
protos.chaincodeMessage
init() / invoke()
ACL / Chaincode doesn't exist errors
Get / Put / ...
protos.chaincodeMessage
get / put ledger
success / error
success / error
success / error with status (errorcode)
if error / statusCode > 500
success / error
return pb.Response with error message
endorse(payload)
if success
return pb.Response with protos error/proposal response
if error
return the error