# Running Smart Contracts on HyperLedger Fabric

For workshop materials, recordings, assessment and **Certificate of Completion** validating your ability build Smart Contracts on Hyperledger Fabric using Solidity and the EVM please visit the website with QR and Access Code.

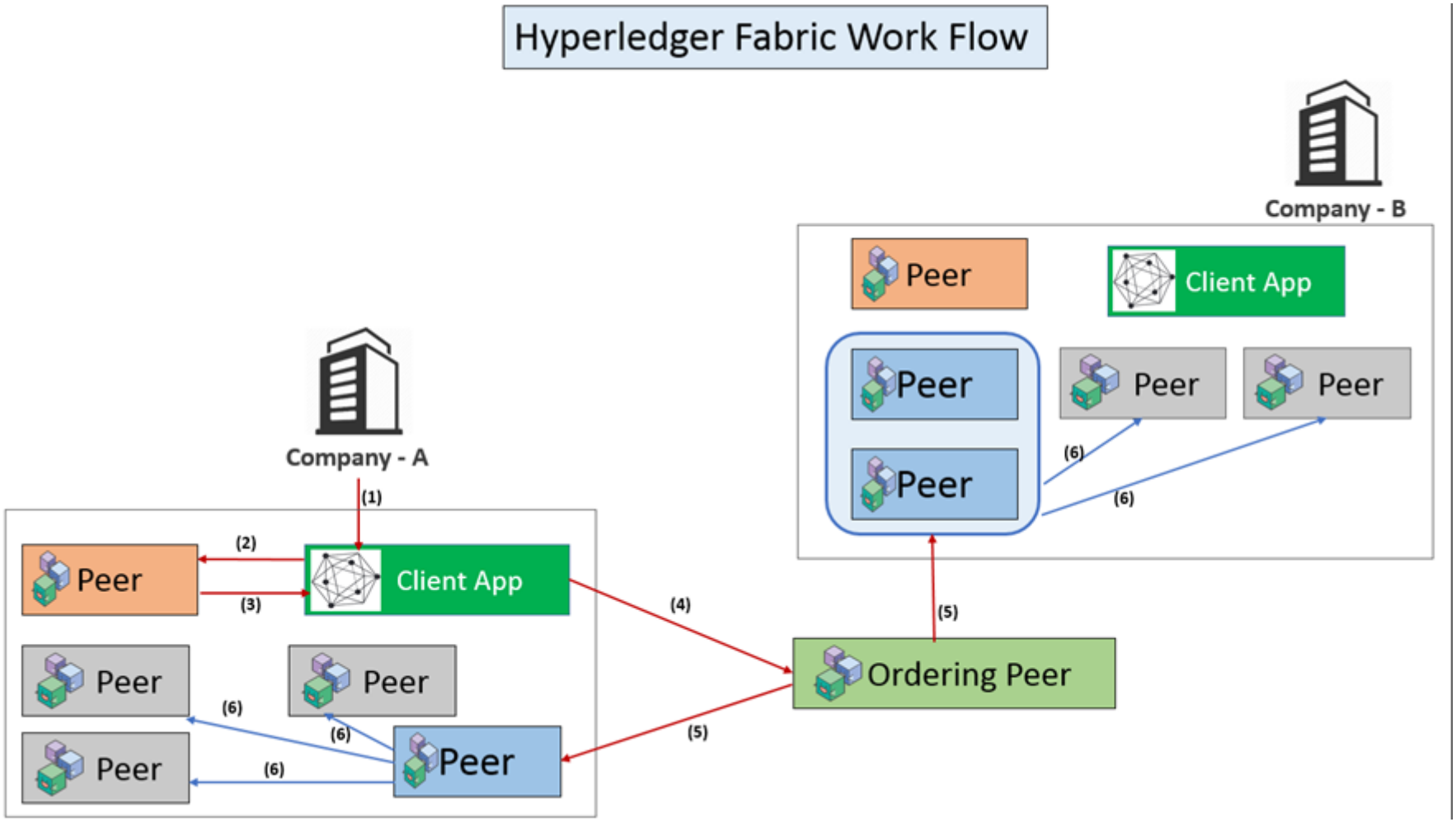
**Access Code: HYP-FAB-SMART**

TECHNOLOGY ENABLED LEARNING SOLUTIONS ®

## Module Objectives:

- Hyperledger Fabric product architecture, including the EVM/Solidity components.
- Smart Contract Development demo
- Solidity Smart Contract Examples
- Why use the EVM for HyperLedger Fabric applications
- Use Remix for building Smart Contracts
- Solidity.
- Data types.
- Functions and Accessibility, i.e., public, private.
- Deployment Demonstration
- Distributed Application (DApp) demonstration
- Next Steps.

# HyperLedger Fabric Architecture



# EVM HyperLedger Fabric Architecture

- Installing the Ethereum Virtual Machine (EVM)
- Add an EVM volume

```
go
├─src
│  ├─github.com
│     │--hyperledger
          │--fabric-samples
          │--fabric-chaincode-evm
```

```
cli:
   volumes:
     - ./../../fabric-chaincode-
evm:/opt/gopath/src/github.com/hyperledger/fabric-chaincode-evm
```

# EVM HyperLedger Fabric Architecture

- Why use the Ethereum Virtual Machine (EVM) on Hyperledger Fabric
- Reuse, Commonality, Community, Resources, more …

SIMBA

## CONFIGURE THE CORE PEER

Next, we need to configure the peer CLI to operate on the core peer for your organization. To do this, select a peer that you want to act as the core peer (this is an arbitrary selection), then define the following environment variables:

(NOTE: This is just an example!! You'll need to set your peer accordingly.)

```bash
export FABRIC_CFG_PATH=<path to directory containing the core.yaml configuration>
export CORE_PEER_TLS_ENABLED=<true | false>
export CORE_PEER_LOCALMSPID=<your org's MSP ID>
export CORE_PEER_TLS_ROOTCERT_FILE=<path to core peer's TLS CA cert>
export CORE_PEER_MSPCONFIGPATH=<path to core peer admin user's MSP>
export CORE_PEER_ADDRESS=<peer address>
export ORDERER_CA=<path to orderer's /msp/tlscacerts/tlsca.pem
```

(HINT: If you are using the sample-network, then you will need to create
organizations will have its own core peer. My suggestion would be to cr
switch between environments. This way you can run the peer cli comma
again to target the other peer.)

120% https://trufflesuite.com/docs/ Search

TRUFFLE SUITE

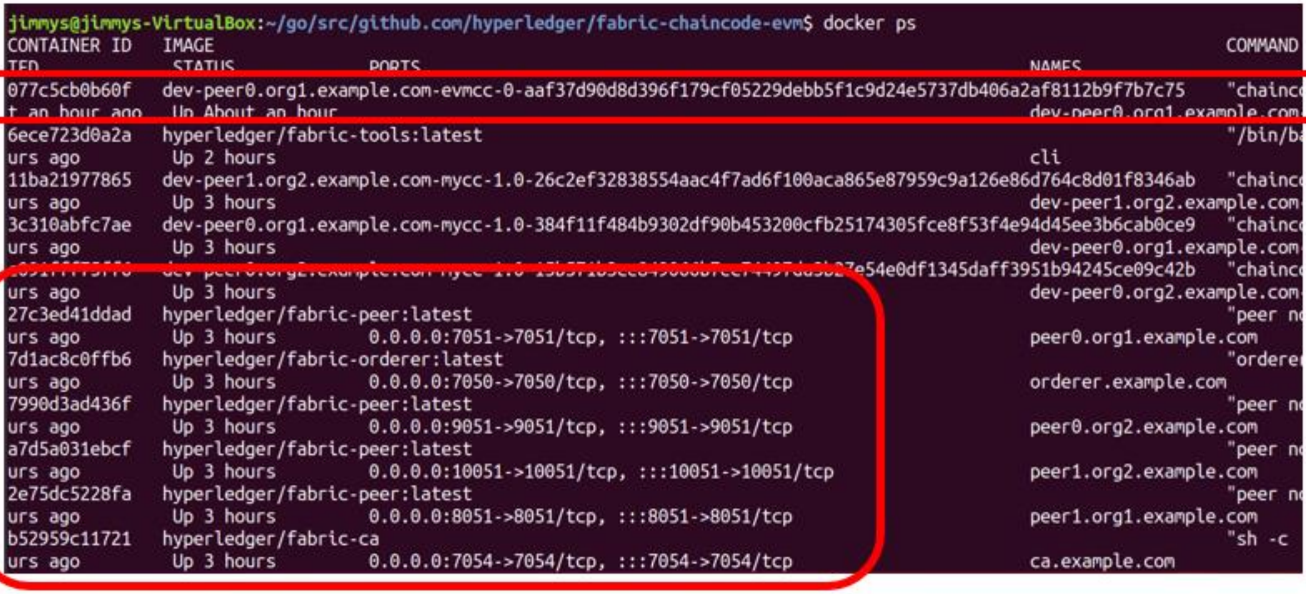
Search

See documentation for **select blockchains**

- Ethereum
- Quorum
- Tezos
- Hyperledger Fabric (EVM)
- Corda
- Filecoin

# EVM HyperLedger Fabric Architecture

- Installing the Ethereum Virtual Machine (EVM)
- Part I: Install the EVM on Hyperledger Fabric
- Part II: Deploy the EVM Smart Contract to Hyperledger Fabric
- Part III: Install NodeJS and Web3 and run the Smart Contract with Node and Web3.



```
jimmys@jimmys-VirtualBox:~/go/src/github.com/hyperledger/fabric-chaincode-evm$ docker ps
CONTAINER ID   IMAGE                                                                                                   COMMAND
TED             STATUS              PORTS                                                   NAMES
077c5cb0b60f   dev-peer0.org1.example.com-evmcc-0-aaf37d90d8d396f179cf05229debb5f1c9d24e5737db406a2af8112b9f7b7c75   "chainc
t an hour ago   Up About an hour                                                            dev-peer0.org1.example.com
6ece723d0a2a   hyperledger/fabric-tools:latest                                                                         "/bin/ba
urs ago         Up 2 hours                                                                  cli
11ba21977865   dev-peer1.org2.example.com-mycc-1.0-26c2ef32838554aac4f7ad6f100aca865e87959c9a126e86d764c8d01f8346ab   "chainc
urs ago         Up 3 hours                                                                  dev-peer1.org2.example.com
3c310abfc7ae   dev-peer0.org1.example.com-mycc-1.0-384f11f484b9302df90b453200cfb25174305fce8f53f4e94d45ee3b6cab0ce9   "chainc
urs ago         Up 3 hours                                                                  dev-peer0.org1.example.com
[illegible]    dev-peer0.org2.example.com-mycc-1.0-[illegible]7e54e0df1345daff3951b94245ce09c42b   "chainc
urs ago         Up 3 hours                                                                  dev-peer0.org2.example.com
27c3ed41ddad   hyperledger/fabric-peer:latest                                                                          "peer nc
urs ago         Up 3 hours          0.0.0.0:7051->7051/tcp, :::7051->7051/tcp               peer0.org1.example.com
7d1ac8c0ffb6   hyperledger/fabric-orderer:latest                                                                       "ordere
urs ago         Up 3 hours          0.0.0.0:7050->7050/tcp, :::7050->7050/tcp               orderer.example.com
7990d3ad436f   hyperledger/fabric-peer:latest                                                                          "peer nc
urs ago         Up 3 hours          0.0.0.0:9051->9051/tcp, :::9051->9051/tcp               peer0.org2.example.com
a7d5a031ebcf   hyperledger/fabric-peer:latest                                                                          "peer nc
urs ago         Up 3 hours          0.0.0.0:10051->10051/tcp, :::10051->10051/tcp           peer1.org2.example.com
2e75dc5228fa   hyperledger/fabric-peer:latest                                                                          "peer nc
urs ago         Up 3 hours          0.0.0.0:8051->8051/tcp, :::8051->8051/tcp               peer1.org1.example.com
b52959c11721   hyperledger/fabric-ca                                                                                   "sh -c
urs ago         Up 3 hours          0.0.0.0:7054->7054/tcp, :::7054->7054/tcp               ca.example.com
```

# EVM HyperLedger Fabric Architecture

- Installing the Ethereum Virtual Machine (EVM)

- Part II: Deploy the EVM Smart Contract to Hyperledger Fabric

```
File  Edit  View  Search  Terminal  Help
root@d626eb9602f5:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode instantiate -n evmcc -v 0 -C mychannel -c '{"Args":[]}' -o o
rderer.example.com:7050 --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/ordere
r.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
2022-10-28 13:03:56.124 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2022-10-28 13:03:56.124 UTC [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
root@d626eb9602f5:/opt/gopath/src/github.com/hyperledger/fabric/peer#
```

# EVM HyperLedger Fabric Architecture

- Installing the Ethereum Virtual Machine (EVM)

- Part III: Install NodeJS and Web3 and run the Smart Contract with Node and Web3.
- A look at EVM other Tools supported for Hyperledger.

```
File  Edit  View  Search  Terminal  Help
root@d626eb9602f5:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -n evmcc -C mychannel -c '{"Args":["5a6a377ed8d4aa527
1f3cba63149497c375f666f","60fe47b1000000000000000000000000000000000000000000000000000000000000000a"]}' -o orderer.example.com:7050 --tls --cafile
 /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.exa
mple.com-cert.pem
2022-10-28 13:31:40.393 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200
root@d626eb9602f5:/opt/gopath/src/github.com/hyperledger/fabric/peer#
```

```
File  Edit  View  Search  Terminal  Help
root@d626eb9602f5:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -n evmcc -C mychannel -c '{"Args":["5a6a377ed8d4aa5271
f3cba63149497c375f666f","6d4ce63c"]}' --hex
000000000000000000000000000000000000000000000000000000000000000a
root@d626eb9602f5:/opt/gopath/src/github.com/hyperledger/fabric/peer#
```

```
File  Edit  View  Search  Terminal  Help
> myContract.set(10)
```

```
File  Edit  View  Search  Terminal  Help
> myContract.get()
BigNumber { s: 1, e: 1, c: [ 10 ] }
> myContract.get().toNumber()
10
>
```

# EVM HyperLedger Fabric Architecture

# EVM HyperLedger Fabric Architecture

- Truffle Configuration for Hyperledger Fabric.

TRUFFLE SUITE



## Working With Hyperledger EVM

As of version `5.0.27`, Truffle supports development with Hyperledger Fabric's EVM chaincode, a **permissioned** version of Ethereum.

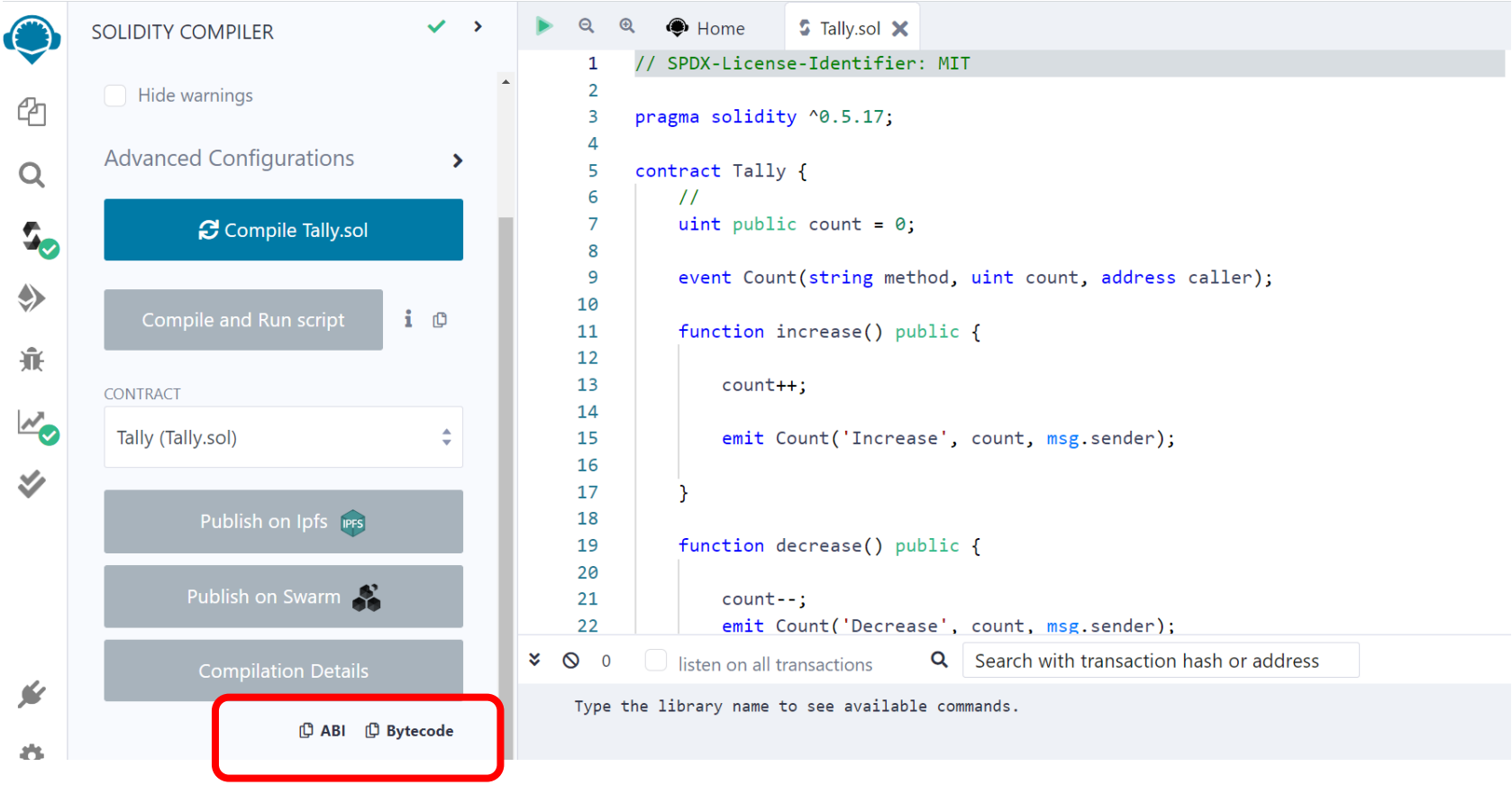
### Configuration

To use Fabric EVM, you must modify your network in `truffle-config.js` to include a parameter `type` set to `"fabric-evm"`. See the example below.

```
module.exports = {
  networks: {
    development: {
      host: "127.0.0.1",
      port: 5000, // default Fab3 port
      network_id: "*",
      type: "fabric-evm"
    }
  }
};
```

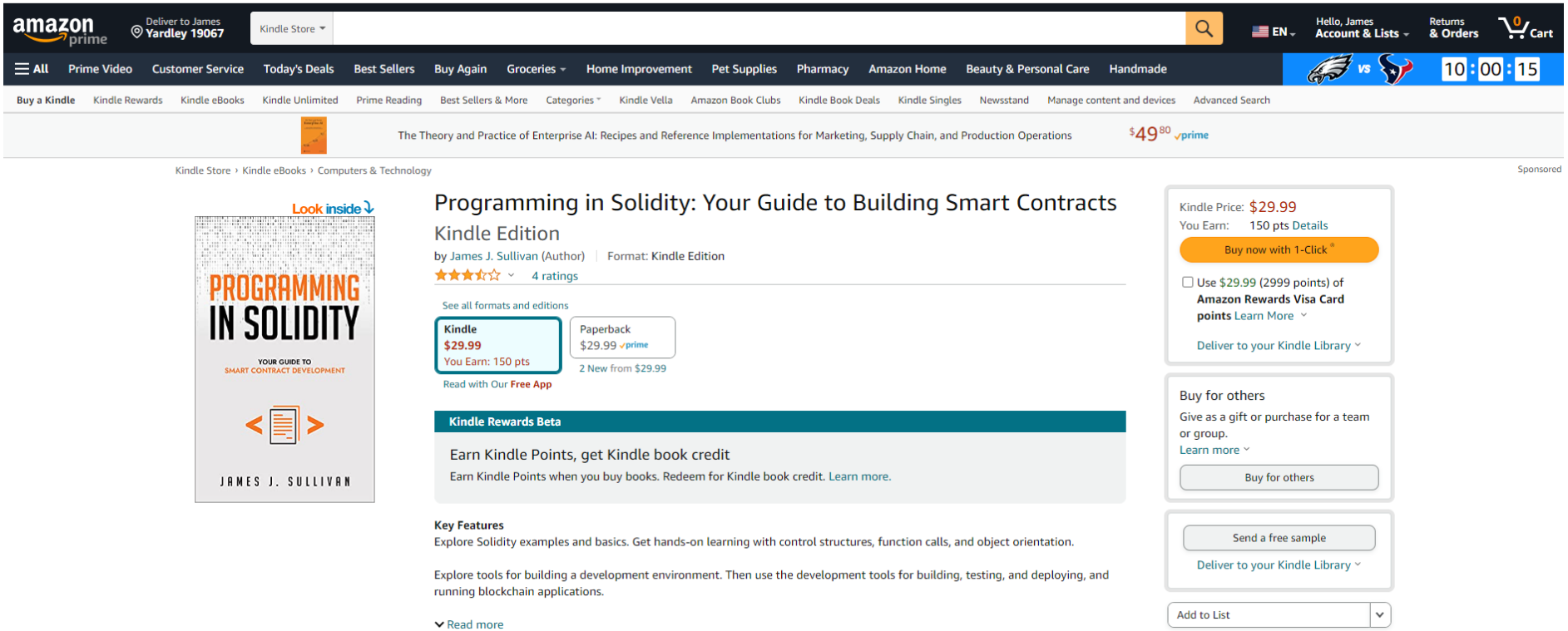
# EVM HyperLedger Fabric Architecture

- Remix for Hyperledger Fabric.
- Remix can connect to Hyperledger Fabric



# EVM HyperLedger Fabric Architecture

- Developer Solidity Smart Contracts Book



# EVM HyperLedger Fabric

- Truffle Project

Compile, Deploy and Run Distributed Applications
src folder stores JavaScript and HTML to call the Web3 Smart Contract API

| Name | Date modified |
|---|---|
| build | 5/21/2021 9:34 PM |
| contracts | 5/21/2021 9:43 PM |
| migrations | 5/21/2021 9:34 PM |
| node_modules | 5/21/2021 9:52 PM |
| src | 5/21/2021 9:34 PM |
| test | 5/21/2021 9:34 PM |
| bs-config.json | 5/21/2021 9:34 PM |
| package.json | 6/16/2021 8:10 PM |
| package-lock.json | 6/16/2021 8:11 PM |
| truffle-config.js | 5/24/2021 2:01 PM |

# EVM HyperLedger Fabric

- Truffle
  - Create a Project
  - Compile, Deploy and Run

Administrator: Windows PowerShell

```
PS C:\EthereumContract\Build-Truffle-Solidity-smart-contract-in-5-minutes-main> truffle migrate --reset

Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.



Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


1_initial_migration.js
======================

   Replacing 'Migrations'
   ----------------------
   > transaction hash:    0x6baf22ba0f3bff9650da1911c802abd122d9c00eefe4912ca841fbc7254ff422
   > Blocks: 0            Seconds: 0
   > contract address:    0xA8Da5dFb4C9ABcBC724980aa0Abc766ee4E6FA68
   > block number:        31
   > block timestamp:     1666378976
   > account:             0x644F120A3DaD32B474a17C4d54e253e8307bBDe3
   > balance:             99.95338024
   > gas used:            153490 (0x25792)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.0030698 ETH
```

# EVM HyperLedger Fabric

- Truffle Console

```
truffle(development)> let newInstance = await Tally.new();
undefined
truffle(development)> newInstance
TruffleContract {
  constructor: [Function: TruffleContract] {
    _constructorMethods: {
      configureNetwork: [Function: configureNetwork],
      setProvider: [Function: setProvider],
      new: [Function: new],
      at: [AsyncFunction: at],
      deployed: [AsyncFunction: deployed],
      defaults: [Function: defaults],
      hasNetwork: [Function: hasNetwork],
      isDeployed: [Function: isDeployed],
      detectNetwork: [AsyncFunction: detectNetwork],
      setNetwork: [Function: setNetwork],
      setNetworkType: [Function: setNetworkType],
      setWallet: [Function: setWallet],
      resetAddress: [Function: resetAddress],
      link: [Function: link],
      clone: [Function: clone],
      addProp: [Function: addProp],
      toJSON: [Function: toJSON],
      decodeLogs: [Function: decodeLogs]
    },
```

# EVM HyperLedger Fabric

- Run the Smart Contract from the Truffle Console

Administrator: Windows PowerShell

```
PS C:\EthereumContract\Build-Truffle-Solidity-smart-contract-in-5-minutes-main> truffle console
```

Administrator: Windows PowerShell

```
truffle(development)> newInstance.address
'0x8594c2Da91B7224ca59079dA7d24B970f3c4666f'
truffle(development)> const res0 = await newInstance.getCount();
undefined
truffle(development)> res0
BN { negative: 0, words: [ 0, <1 empty item> ], length: 1, red: null }
truffle(development)> res0.toNumber();
0
truffle(development)> const res1 = await newInstance.increase();
undefined
truffle(development)> const res1 = await newInstance.getCount();
evalmachine.<anonymous>:1
const res1 = global.__await_outside_result; void delete global.__await_outside_result;
^

Uncaught SyntaxError: Identifier 'res1' has already been declared
truffle(development)> const res2 = await newInstance.getCount();
undefined
truffle(development)> res2
BN { negative: 0, words: [ 1, <1 empty item> ], length: 1, red: null }
truffle(development)> res2.toNumber();
1
truffle(development)> const up2 = await newInstance.increase();
undefined
truffle(development)> const res3 = await newInstance.getCount();
undefined
truffle(development)> res3
BN { negative: 0, words: [ 2, <1 empty item> ], length: 1, red: null }
truffle(development)> res3.toNumber();
2
truffle(development)> const up3 = await newInstance.increase();
undefined
truffle(development)> const res4 = await newInstance.getCount();
```

# EVM HyperLedger Fabric

Administrator: Windows PowerShell

```
PS C:\EthereumContract\Build-Truffle-Solidity-smart-contract-in-5-minutes-main> truffle console
truffle(development)> let newInstance = await Tally.new();
undefined
truffle(development)> newInstance.address
'0xC7E9d4711aDA90dFd456814EcE914748F228F476'
truffle(development)> const res0 = await newInstance.getCount();
undefined
truffle(development)> res0
BN { negative: 0, words: [ 0, <1 empty item> ], length: 1, red: null }
truffle(development)> res0.toNumber()
0
truffle(development)> const res1 = await newInstance.increase();
undefined
truffle(development)> const res2 = await newInstance.getCount();
undefined
truffle(development)> const res3 = await newInstance.getCount();
undefined
truffle(development)> res3
BN { negative: 0, words: [ 1, <1 empty item> ], length: 1, red: null }
truffle(development)> res3.toNumber()
1
truffle(development)>
```

# EVM HyperLedger Fabric

- Truffle and the EVM Smart Contract API

```
App = {
  web3Provider: null,
  contracts: {},
  account: '0x0',
  hasVoted: false,

  init: function() {
    return App.initWeb3();
  },

  initWeb3: function() {
    // TODO: refactor conditional
    if (typeof web3 !== 'undefined') {
      // If a web3 instance is already provided by Meta Mask.
      App.web3Provider = web3.currentProvider;
      web3 = new Web3(web3.currentProvider);
    } else {
      // Specify default instance if no web3 instance provided
      App.web3Provider = new Web3.providers.HttpProvider('http://localhost:8545');
      web3 = new Web3(App.web3Provider);
    }
    return App.initContract();
  },

  initContract: function() {
    $.getJSON("Election.json", function(election) {
      // Instantiate a new truffle contract from the artifact
      App.contracts.Election = TruffleContract(election);
```

```
  initContract: function() {
    $.getJSON("Election.json", function(election) {
      // Instantiate a new truffle contract from the artifact
      App.contracts.Election = TruffleContract(election);
      // Connect provider to interact with contract
      App.contracts.Election.setProvider(App.web3Provider);

      App.listenForEvents();

      return App.render();
    });
  },

  // Listen for events emitted from the contract
  listenForEvents: function() {
    App.contracts.Election.deployed().then(function(instance) {
      // Restart Chrome if you are unable to receive this event
      // This is a known issue with Metamask
      // https://github.com/MetaMask/metamask-extension/issues/2393
      instance.votedEvent({}, {
        fromBlock: 0,
        toBlock: 'latest'
      }).watch(function(error, event) {
        console.log("event triggered", event)
        // Reload when a new vote is recorded
        App.render();
      });
```

# EVM HyperLedger Fabric

- Truffle, EVM Smart Contract API build the DApp along with Node and other tools

localhost:3000 Search

## Coolest Kid Election Results

| # | Name | Votes |
|---|---|---|
| 1 | Ryan Williams | 0 |
| 2 | Bryant Neilson | 0 |
| 3 | Sean Vliet | 0 |
| 4 | Allen C Politician | 0 |
| 5 | Paul I Cash | 0 |
| 6 | Jimmy Jay Sullivan | 0 |

**Select Candidate**

Ryan Williams

Vote

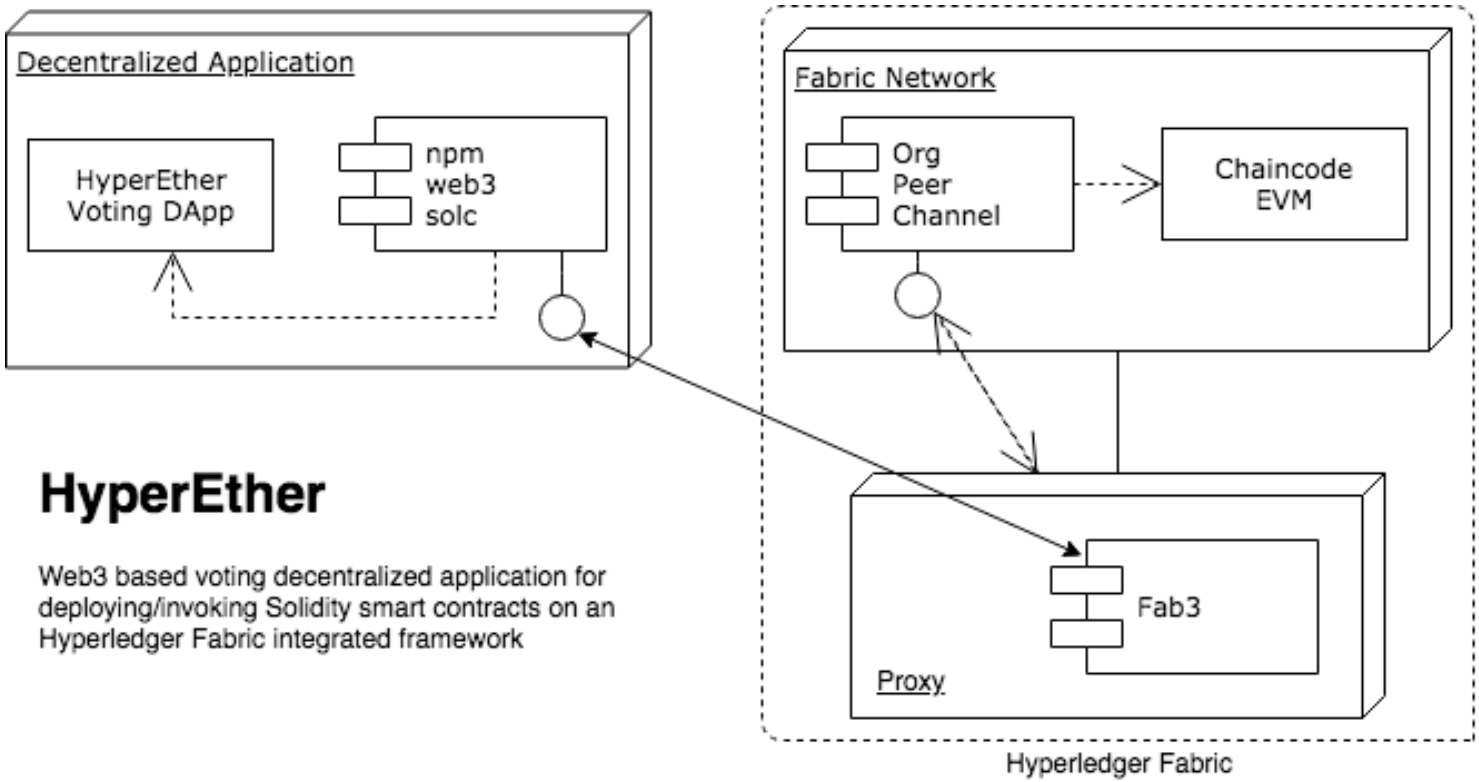
Your Account: 0x118e27e4cd53bd270f8c12a1fdfb38cee4366315

# EVM HyperLedger Fabric Architecture

npm install web3@0.20.2

# EVM HyperLedger Fabric Architecture

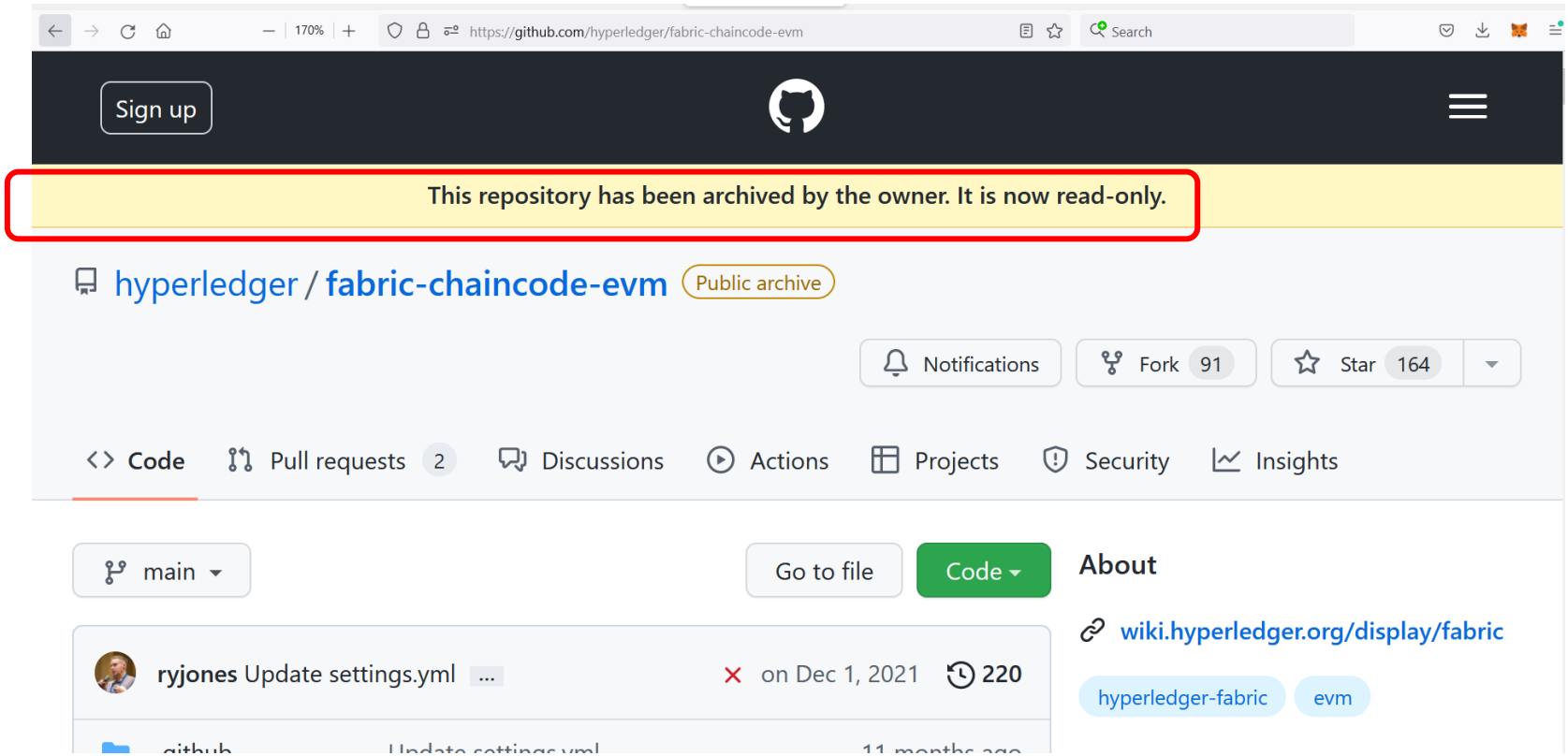
- Voting DApp: https://github.com/IBM/vote-hyperledger-ethereum



# EVM HyperLedger Fabric

- What's Next

Fabric-chaincode-evm is currently archived
It can still be used.



# EVM HyperLedger Fabric

- What's Next
  - Fabric-chaincode-evm is the provider option
  - The Blockchain Academy will continue to be active in Hyperledger EVM Solution

SIMBA

## CONFIGURE THE CORE PEER

Next, we need to configure the peer CLI to operate on the core peer for your organization. To do this, select a peer that you want to act as the core peer (this is an arbitrary selection), then define the following environment variables:

(NOTE: This is just an example!! You'll need to set your peer accordingly.)

```bash
export FABRIC_CFG_PATH=<path to directory containing the core.yaml configuration>
export CORE_PEER_TLS_ENABLED=<true | false>
export CORE_PEER_LOCALMSPID=<your org's MSP ID>
export CORE_PEER_TLS_ROOTCERT_FILE=<path to core peer's TLS CA cert>
export CORE_PEER_MSPCONFIGPATH=<path to core peer admin user's MSP>
export CORE_PEER_ADDRESS=<peer address>
export ORDERER_CA=<path to orderer's /msp/tlscacerts/tlsc
```

(HINT: If you are using the sample-network, then you will need to c
organizations will have its own core peer. My suggestion would be
switch between environments. This way you can run the peer cli co
again to target the other peer.)

120% https://trufflesuite.com/docs/ Search

TRUFFLE SUITE

Search

See documentation for **select blockchains**

- Ethereum
- Quorum
- Tezos
- Hyperledger Fabric (EVM)
- Corda
- Filecoin

## Next Steps

- Front End, React.js, Node

- **Scan QR Code:**
- Register with Access Code: **HYP-FAB-SMART**
  - Included
    - Workshop Materials
    - Step-by-Step Videos
    - Assessment
    - Achieve a **Certificate of Completion**
    - Web3 Foundations Course

## Want more?

- Attend the Hands-on 5 Hour Workshops with snapshotted virtual desktops

- Thank You, for your time.
- Thanks to the Linux Foundation

Access Code: HYP-FAB-SMART

https://on360.io/linux-community-registration-form/