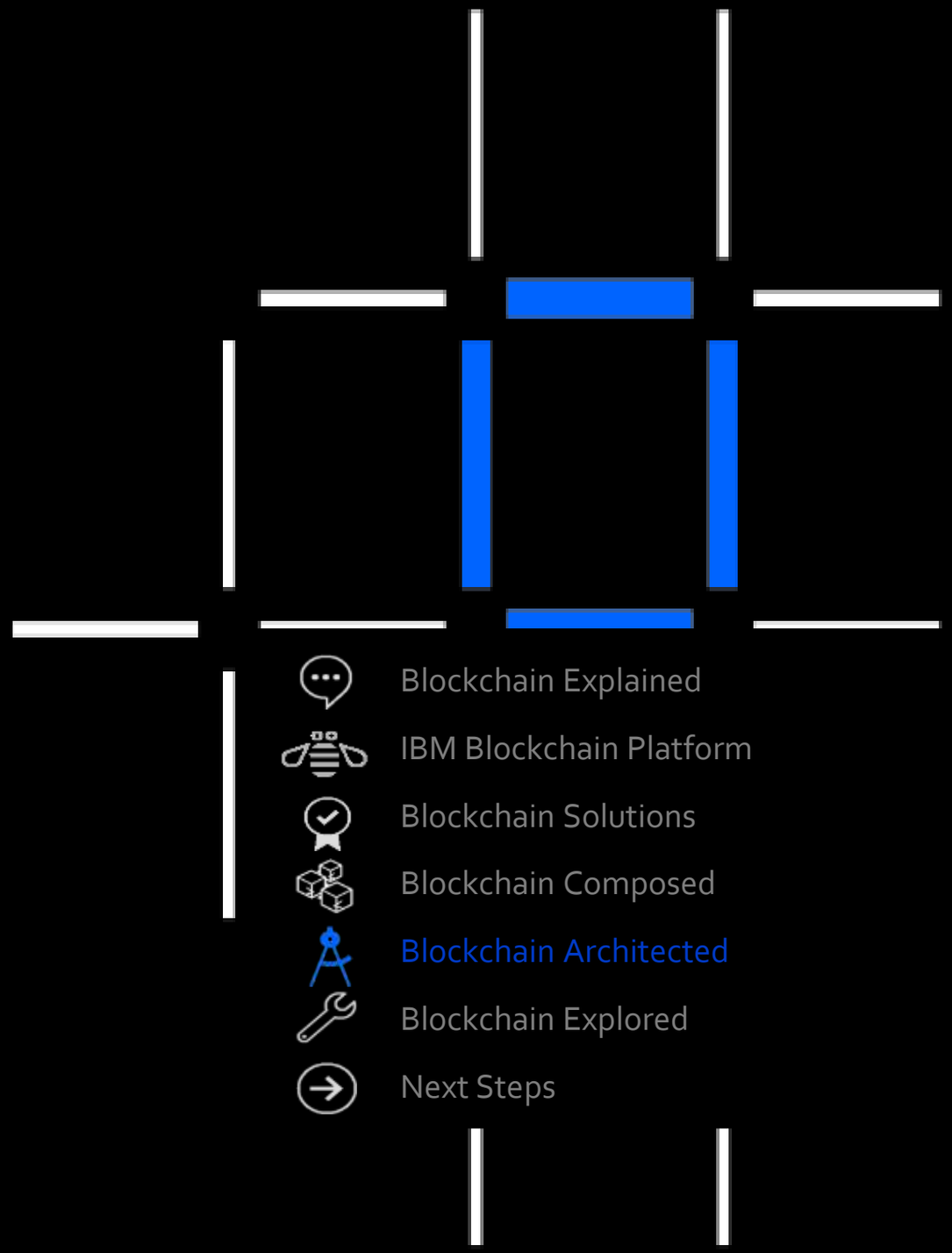
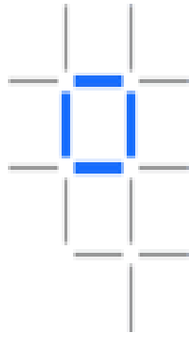
Deploy a blockchain web-app with Hyperledger Fabric

Horea Porutiu
Advisory Software Engineer, IBM
July 2019



IBM **Blockchain**

By a show of hands…

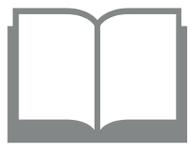


1. Understands how blockchains work?
2. Know what Hyperledger Fabric is?
3. Developed a smart contract?
4. Developed a dapp?
5. Deployed a network to production?

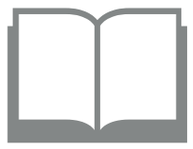
What you will learn

1. Blockchain vocabulary

What you will learn

1. Blockchain vocabulary 
2. A use-case of blockchain

What you will learn

1. Blockchain vocabulary 
2. A use-case of blockchain
3. Architecture behind a Hyperledger Fabric solution

What you will learn

1. Blockchain vocabulary

2. A use-case of blockchain

3. Architecture behind a Hyperledger Fabric solution

4. Difference between the world state and the ledger

What you will learn

1. Blockchain vocabulary

2. A use-case of blockchain

3. Architecture behind a Hyperledger Fabric solution

4. Difference between the world state and the ledger

5. Difference between public and private blockchains

What you will learn

1. Blockchain vocabulary
2. A use-case of blockchain
3. Architecture behind a Hyperledger Fabric solution
4. Difference between the world state and the ledger
5. Difference between public and private blockchains
6. How a client application invokes a smart contract

Security: Public vs. private blockchains

Public blockchains

- Bitcoin
- Users treated equally
- Identity is anonymous

Security: Public vs. private blockchains

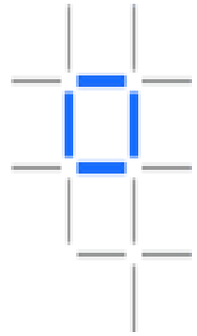
Public blockchains

- Bitcoin
- Users treated equally
- Identity is anonymous

Private blockchains

- Hyperledger Fabric, Quorum
- Network members known, transactions can be secret

Security: Public vs. private blockchains



Public blockchains

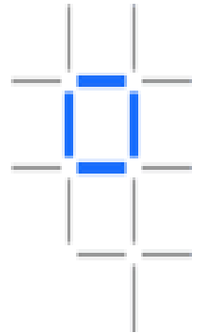
- Bitcoin
- Users treated equally
- Identity is anonymous

Private blockchains

- Hyperledger Fabric, Quorum
- Network members known, transactions can be secret

- Identity management (anonymous vs. known)

Security: Public vs. private blockchains



Public blockchains

- Bitcoin
- Users treated equally
- Identity is anonymous

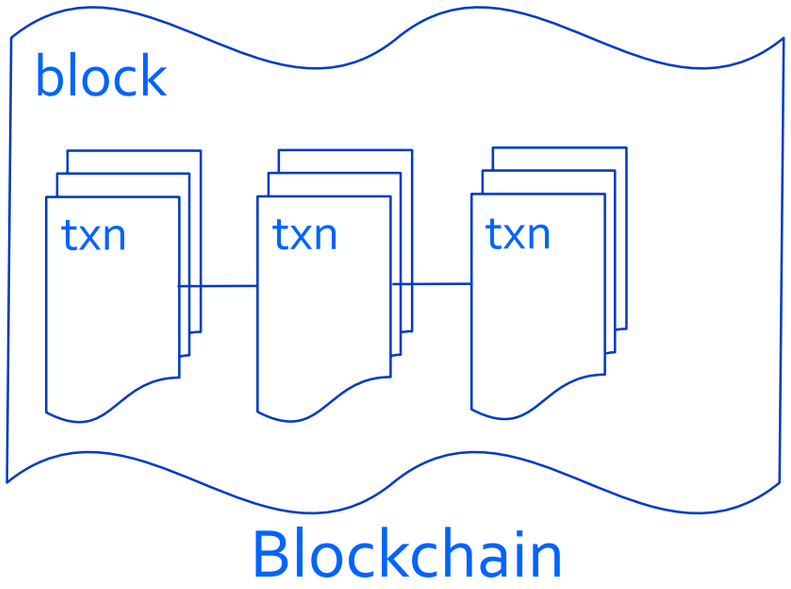
Private blockchains

- Hyperledger Fabric, Quorum
- Network members known, transactions can be secret

- Identity management (anonymous vs. known)

- Most business use-cases require private, permissioned blockchains
  - Network members know who they're dealing with (KYC)
  - Membership is controlled

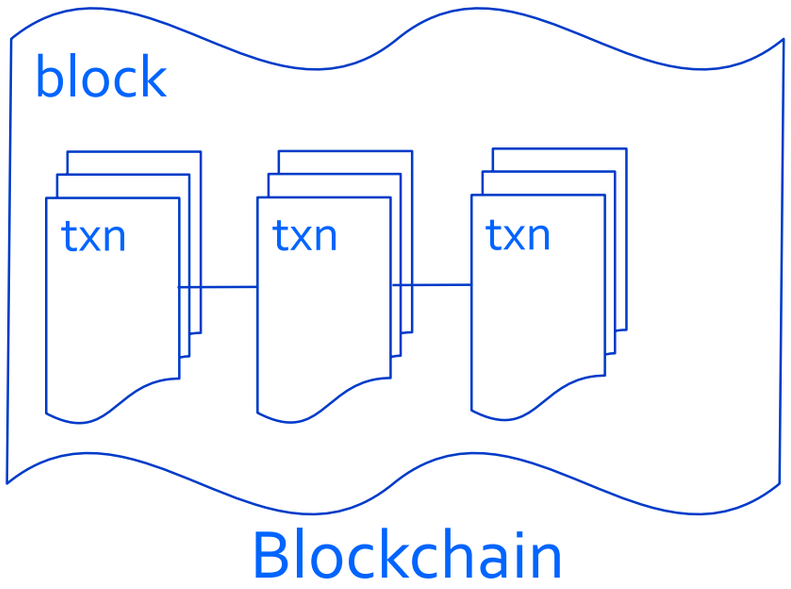
A ledger often consists of two data structures



- Blockchain
  - A linked list of blocks
  - Each block describes a set of transactions

    (e.g. the inputs to a smart contract invocation)
  - Immutable – blocks cannot be tampered

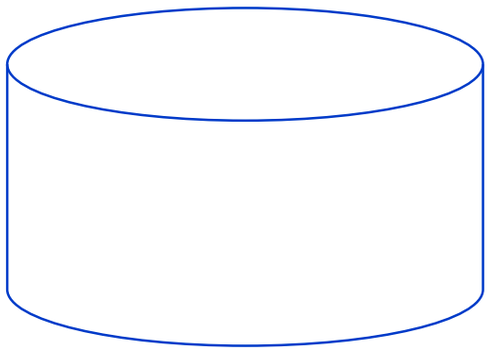
A ledger often consists of two data structures



Blockchain

- Blockchain
  - A linked list of blocks
  - Each block describes a set of transactions

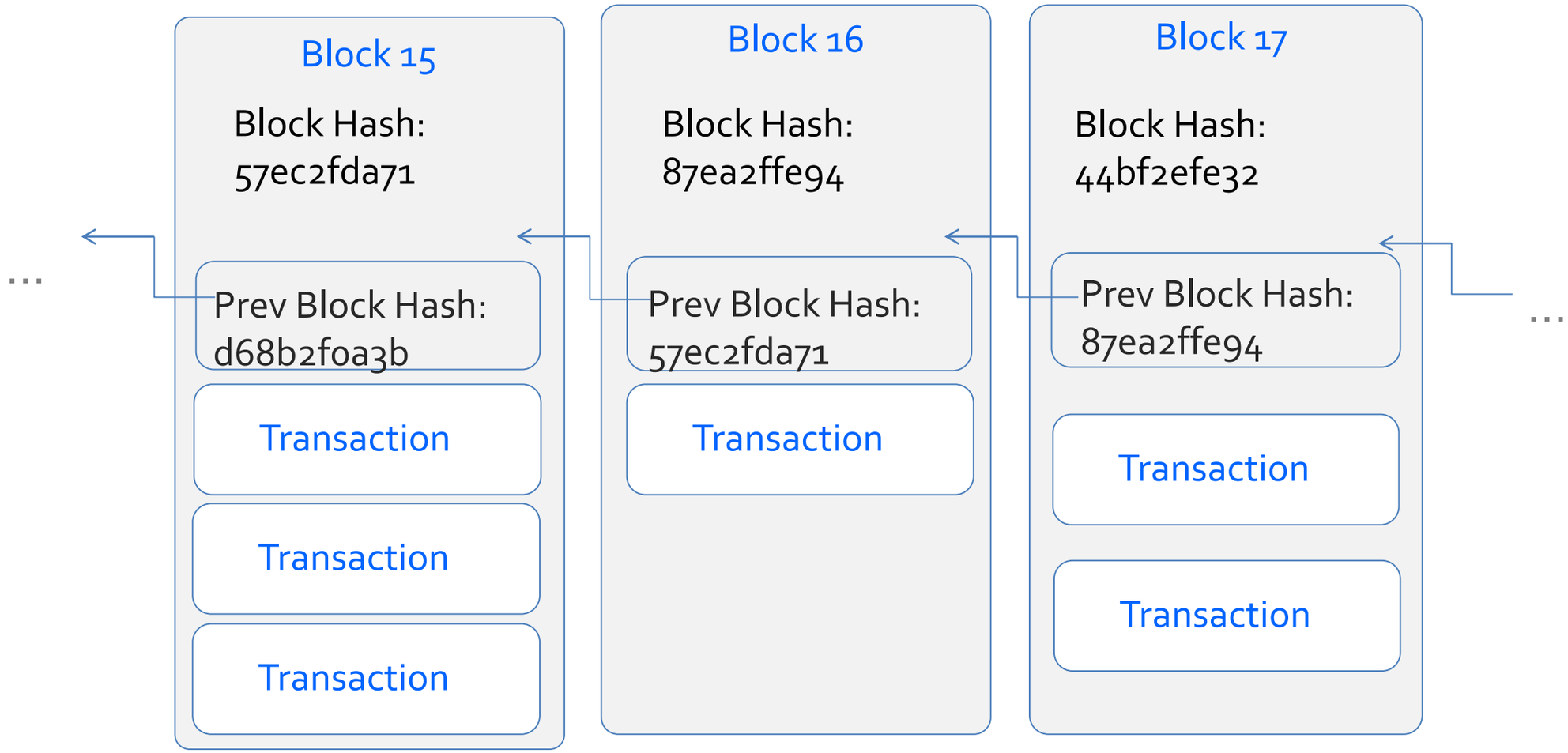
    (e.g. the inputs to a smart contract invocation)
  - Immutable – blocks cannot be tampered



World state

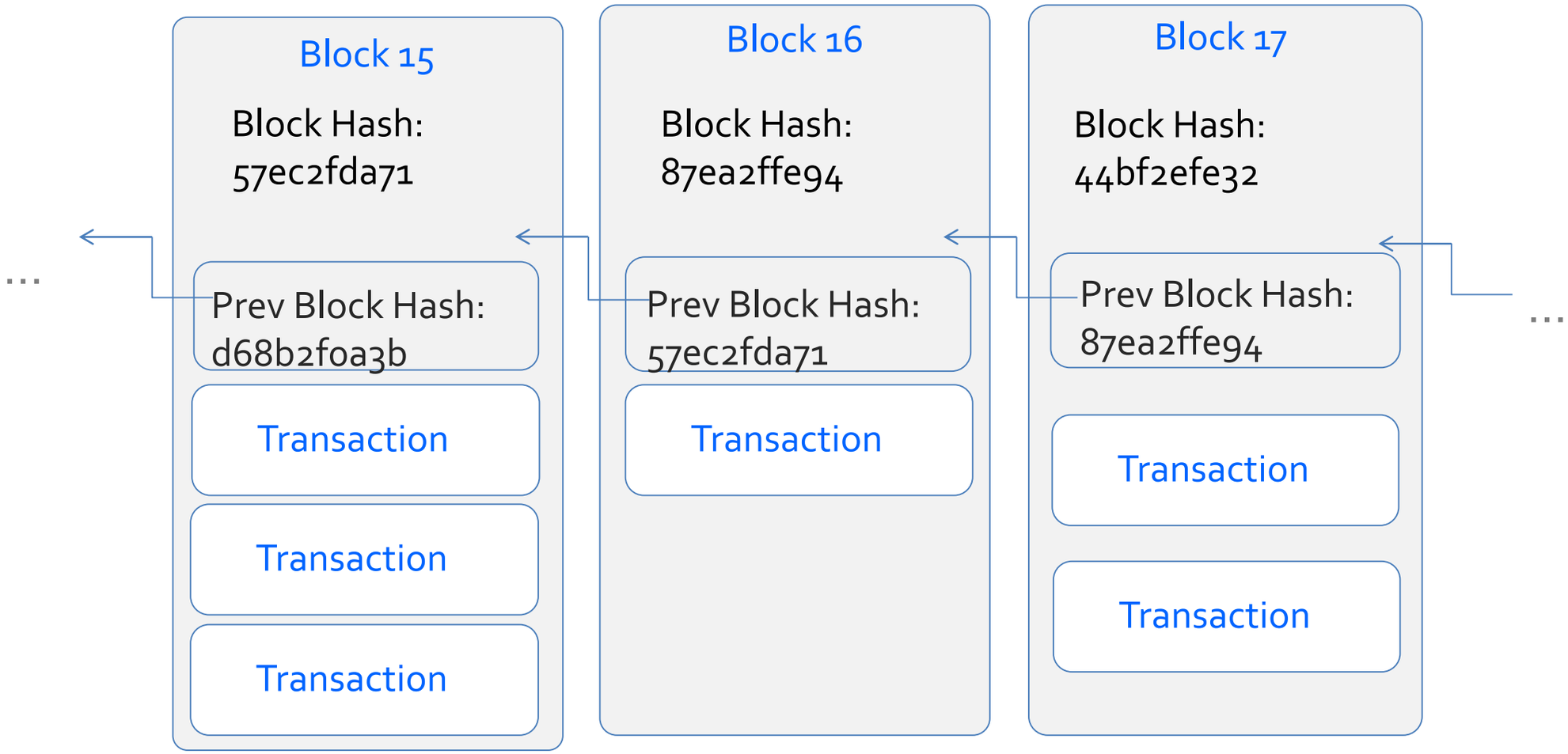
- World State
  - An ordinary database (e.g. key/value store)
  - Stores the combined outputs of all transactions
  - CAN delete

Block detail (simplified)



- New blocks always added to the end

Block detail (simplified)

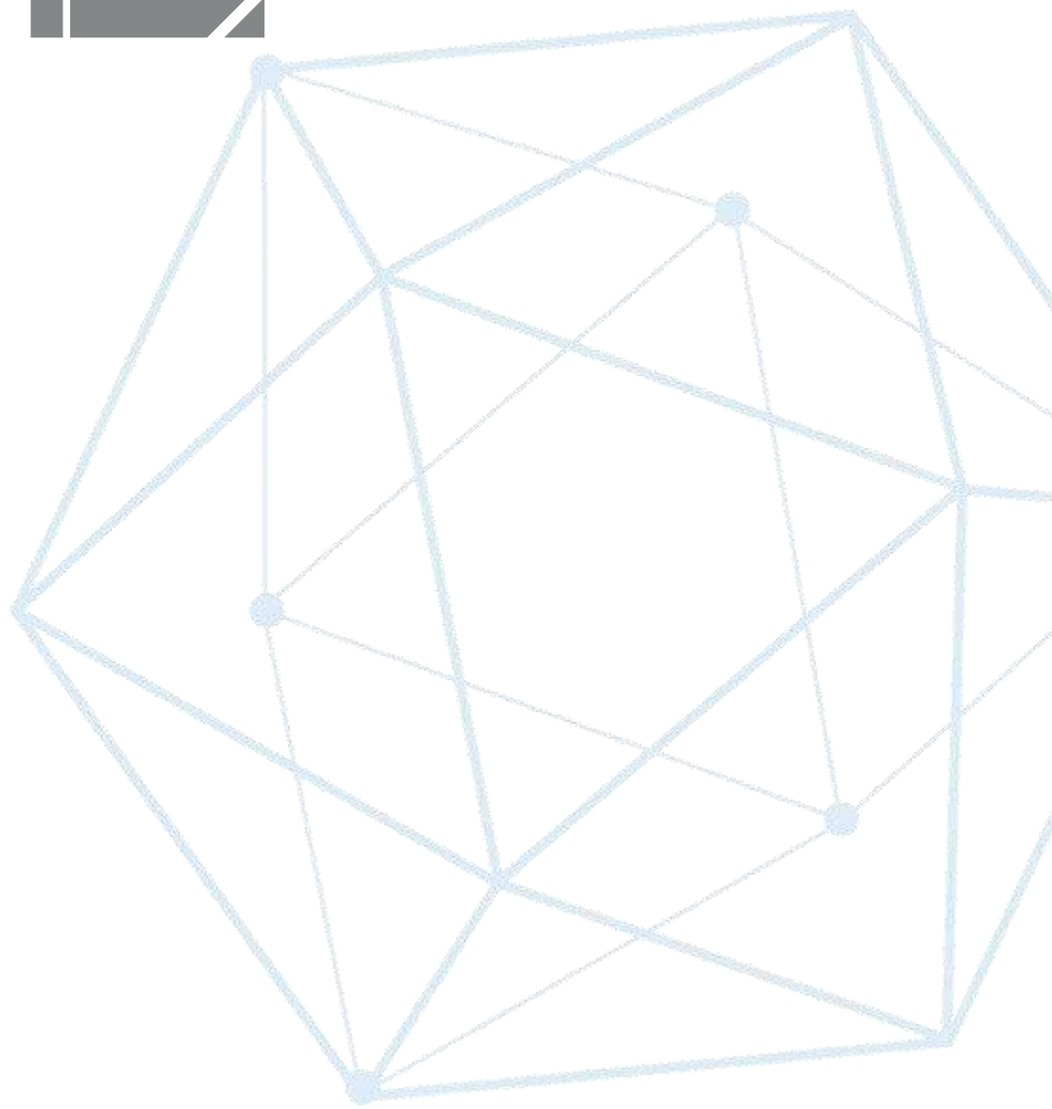


- New blocks always added to the end

- Each block header includes a hash of the current block transactions and the previous block's transactions

What is Hyperledger Fabric

- Platform for distributed ledger solutions



What is Hyperledger Fabric

- Platform for distributed ledger solutions

- Open Source
  - Contributions by hundreds of engineers from tens of organizations

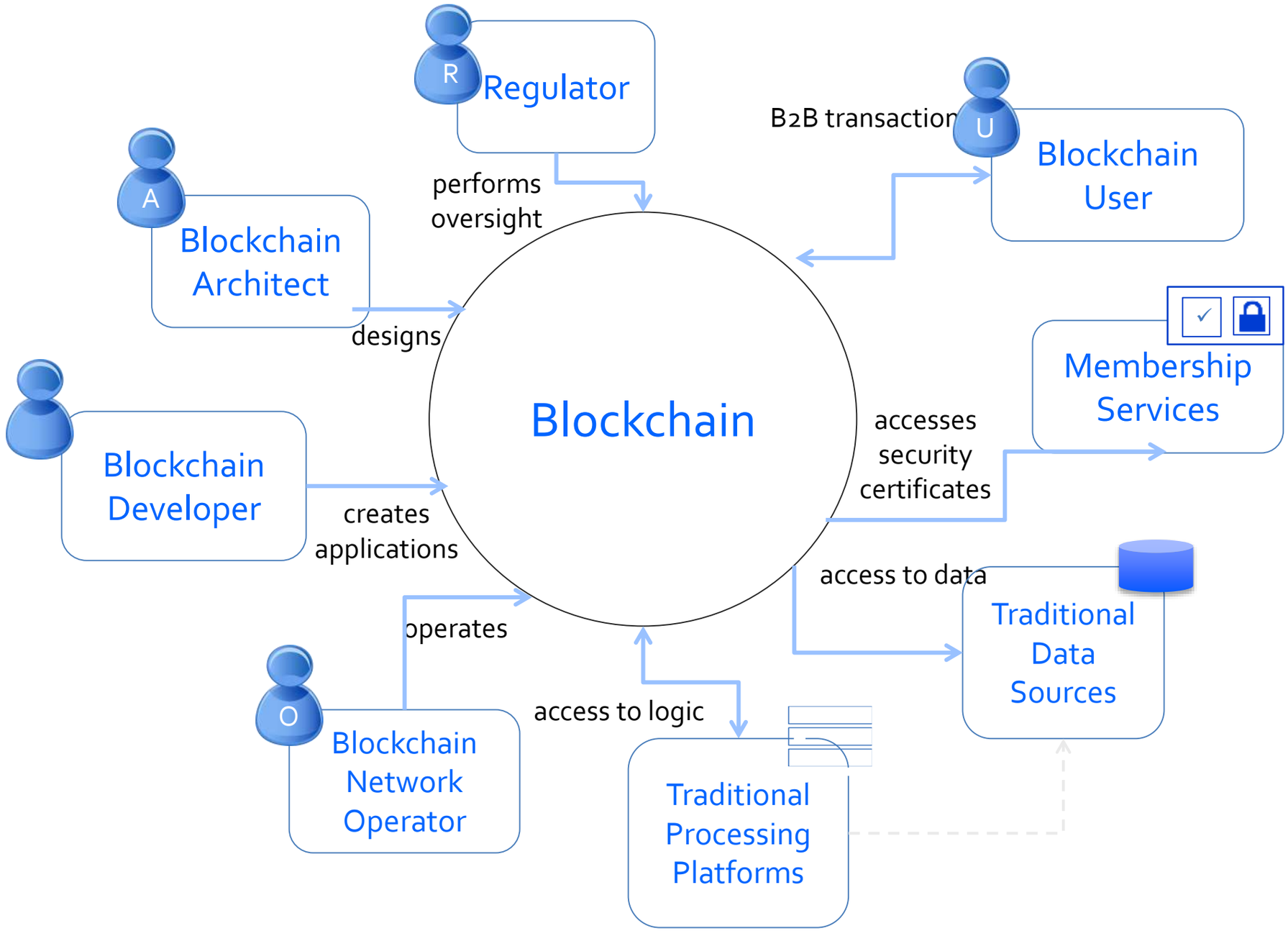
What is Hyperledger Fabric

- Platform for distributed ledger solutions

- Open Source
  - Contributions by hundreds of engineers from tens of organizations

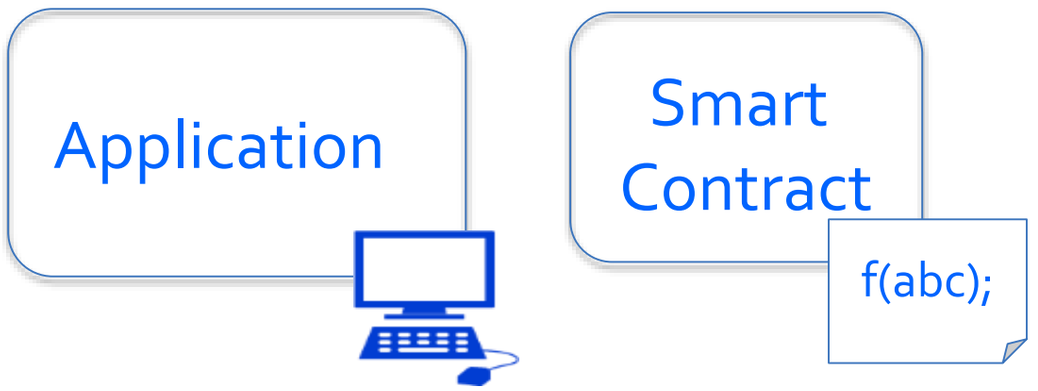
- Features
  - Smart Contracts (updates the ledger)
  - Consensus (synch ledger across network)
  - Privacy (channels)

Actors in a blockchain solution



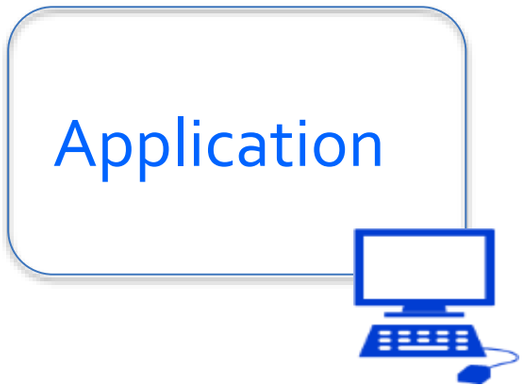
The blockchain developer

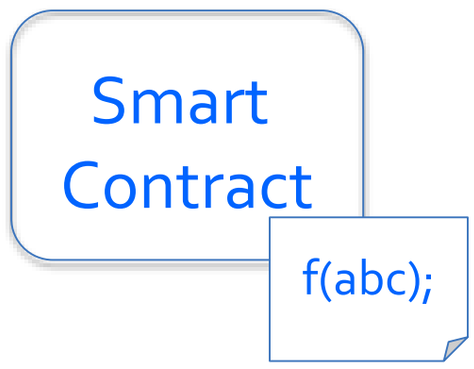
Blockchain developers' primary interests are…

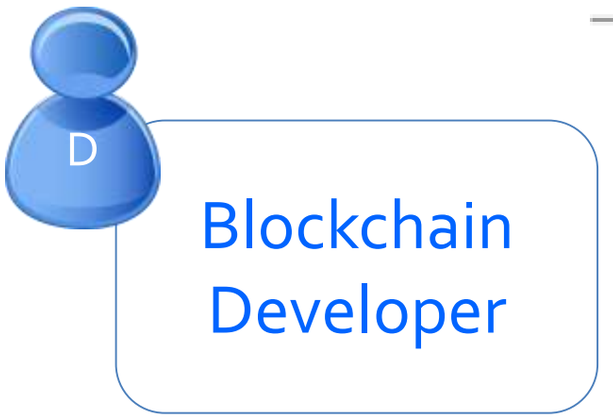


The blockchain developer

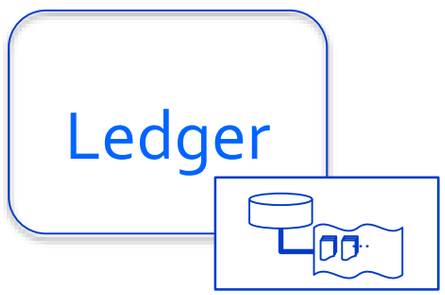
Blockchain developers' primary interests are…

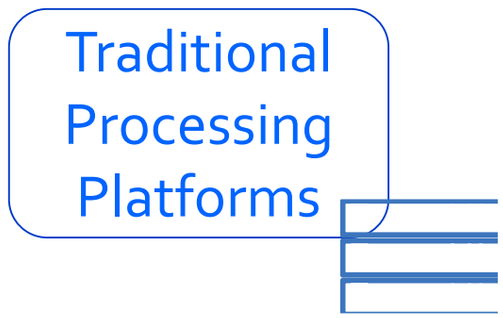


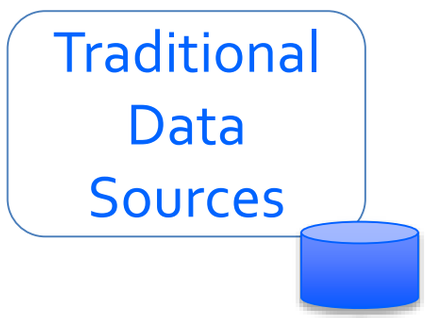


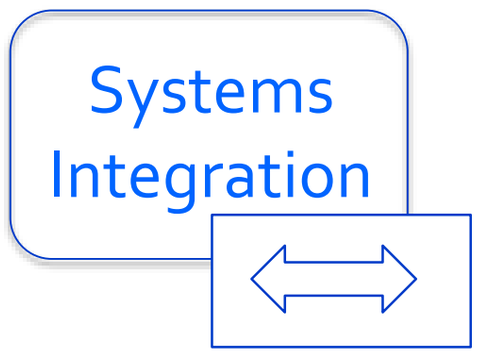


…and how they interact with the ledger and other systems of record:







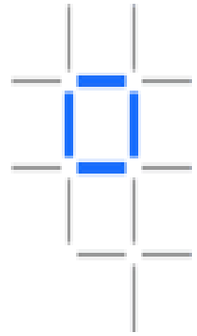


Components in a blockchain solution

Ledger

List of transactions maintained by peers

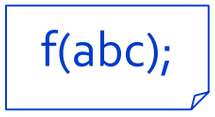
Components in a blockchain solution



Ledger

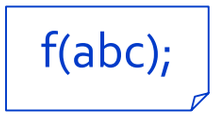
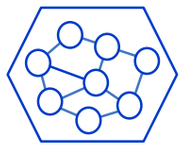
List of transactions maintained by peers

Smart Contract

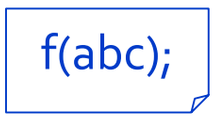
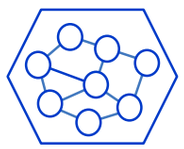
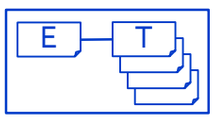


Software running on peer, updates the world state

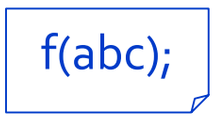
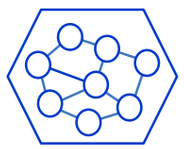
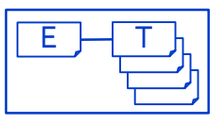
Components in a blockchain solution

Ledger		List of transactions maintained by peers
Smart Contract		Software running on peer, updates the world state
Peer Network		Network which reaches consensus to add blocks

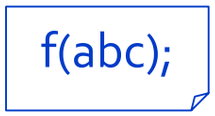
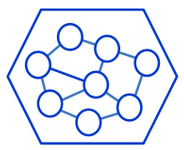
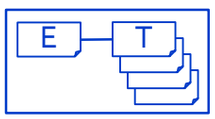
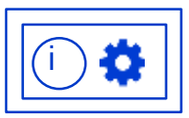
Components in a blockchain solution

Ledger		List of transactions maintained by peers
Smart Contract		Software running on peer, updates the world state
Peer Network		Network which reaches consensus to add blocks
Membership		Authenticates and manages identities on network

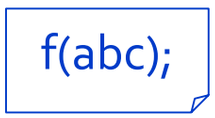
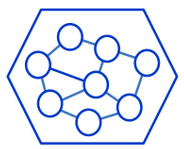
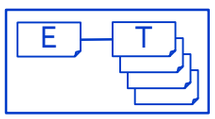
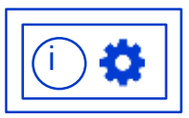
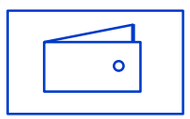
Components in a blockchain solution

Ledger		List of transactions maintained by peers
Smart Contract		Software running on peer, updates the world state
Peer Network		Network which reaches consensus to add blocks
Membership		Authenticates and manages identities on network
Events		Emits notifications of operations on network

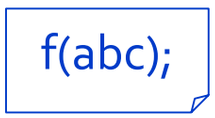
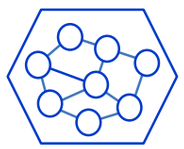
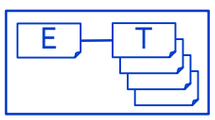
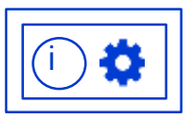
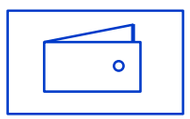
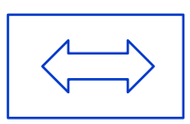
Components in a blockchain solution

Ledger		List of transactions maintained by peers
Smart Contract		Software running on peer, updates the world state
Peer Network		Network which reaches consensus to add blocks
Membership		Authenticates and manages identities on network
Events		Emits notifications of operations on network
Systems Management		Enables us to create/monitor blockchain components

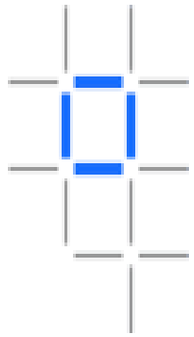
Components in a blockchain solution

Ledger		List of transactions maintained by peers
Smart Contract		Software running on peer, updates the world state
Peer Network		Network which reaches consensus to add blocks
Membership		Authenticates and manages identities on network
Events		Emits notifications of operations on network
Systems Management		Enables us to create/monitor blockchain components
Wallet		Securely manages a user's credentials

Components in a blockchain solution

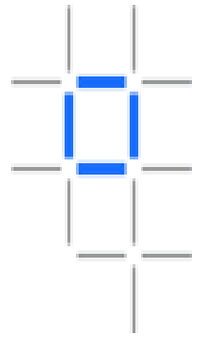
Ledger		List of transactions maintained by peers
Smart Contract		Software running on peer, updates the world state
Peer Network		Network which reaches consensus to add blocks
Membership		Authenticates and manages identities on network
Events		Emits notifications of operations on network
Systems Management		Enables us to create/monitor blockchain components
Wallet		Securely manages a user's credentials
Systems Integration		Integrate blockchain with eternal systems

Smart Contracts



Smart Contracts contain the business logic deployed to peers

Smart Contracts



Smart Contracts contain the business logic deployed to peers

- Interact with the world state through the Fabric shim interface

Smart Contracts

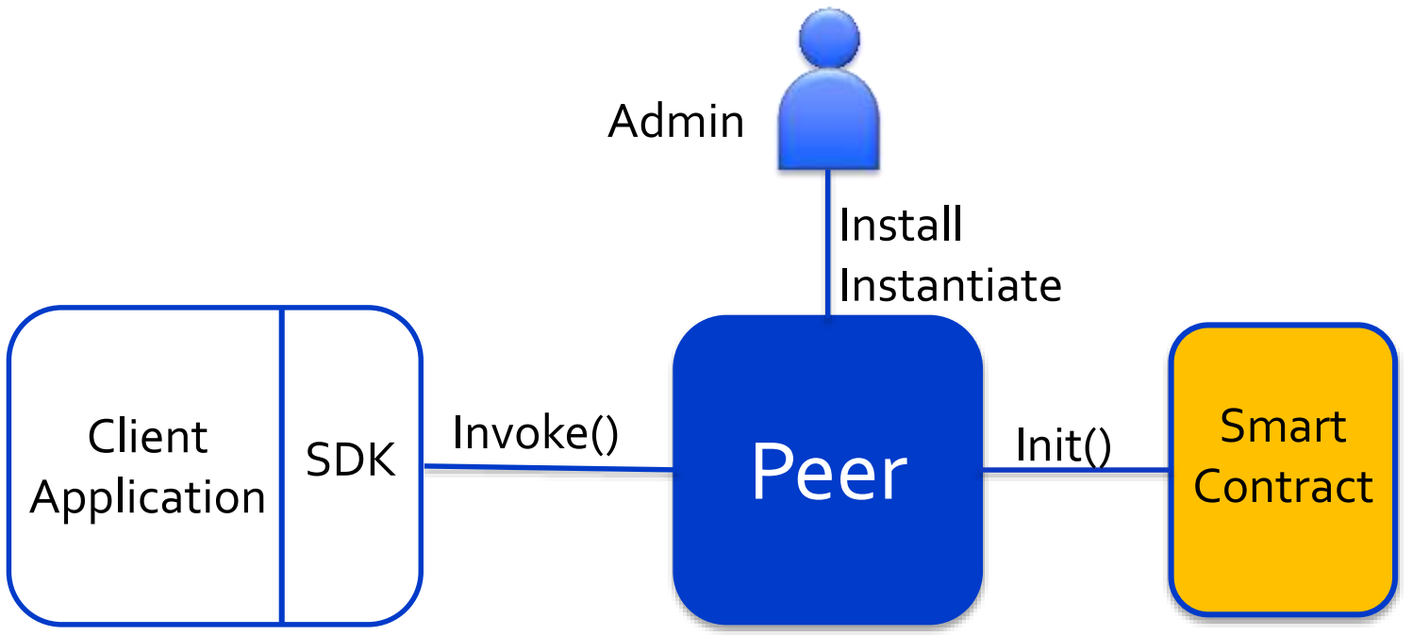
Smart Contracts contain the business logic deployed to peers

- Interact with the world state through the Fabric shim interface
- Language support for:
  - Golang
  - Node.js
  - Java

Smart Contracts

Smart Contracts contain the business logic deployed to peers

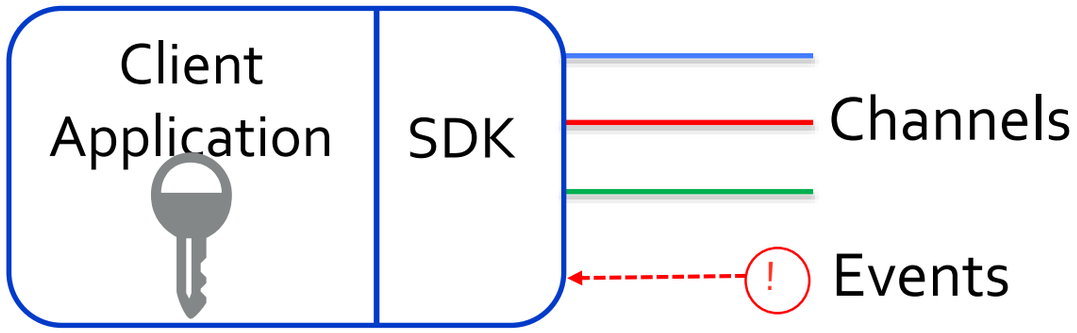
- Interact with the world state through the Fabric shim interface
- Language support for:
  - Golang
  - Node.js
  - Java



Client Application

Client applications use Fabric SDK to:

- Connects over channels to peer and orderer nodes
- Provide public / private keys



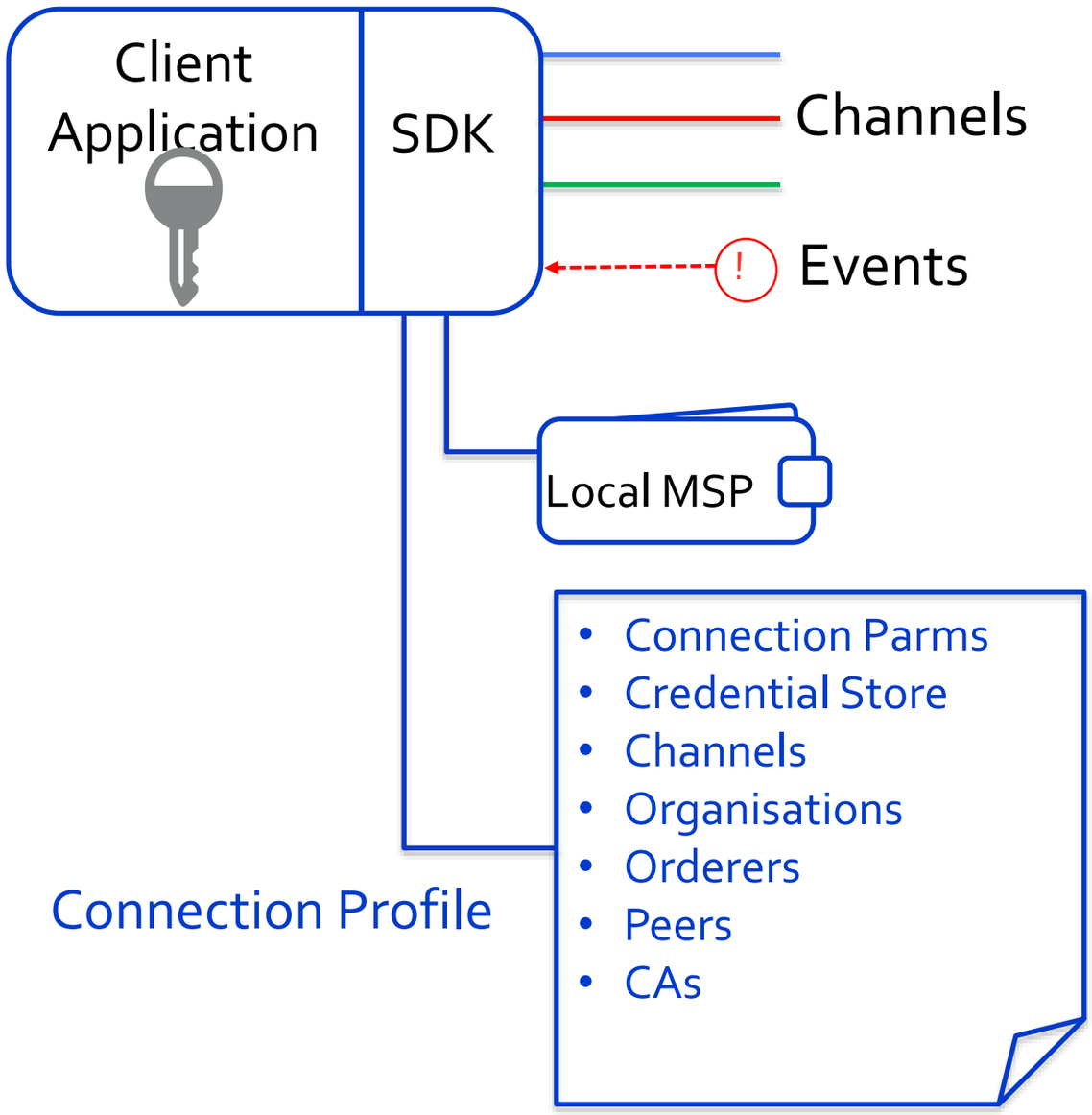
Client Application

Client applications use Fabric SDK to:

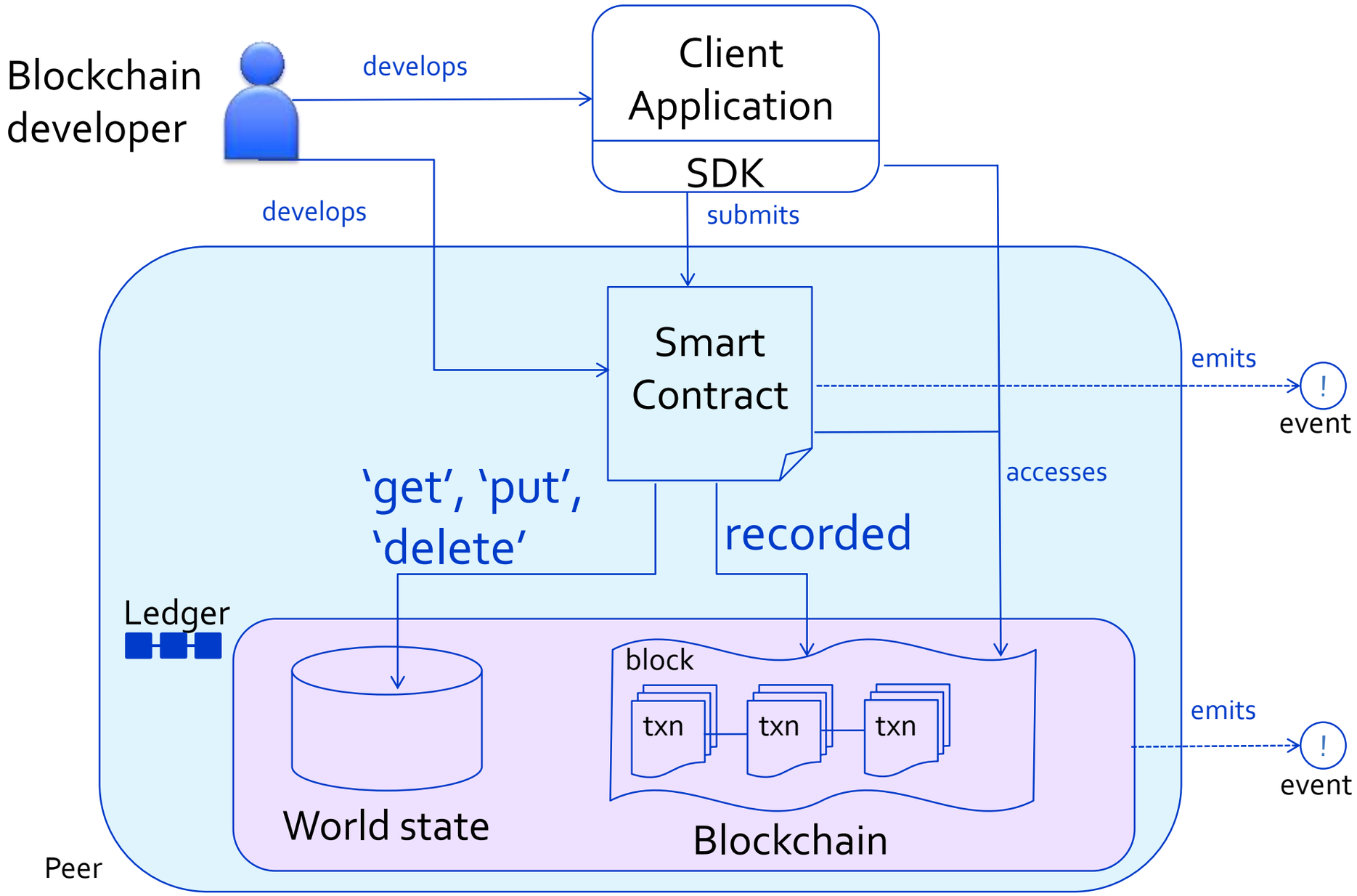
- Connects over channels to peer and orderer nodes
- Provide public / private keys

Connection Profile

- Network end-points and connection parameters
- The gateway to submit transactions to a Hyperledger Fabric network



How applications interact with the ledger



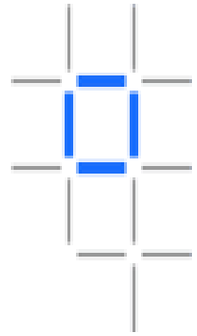
Sample blockchain use-case: e-voting

Components in an e-voting blockchain solution

Ledger

The ledger containing history of submitted votes

Components in an e-voting blockchain solution



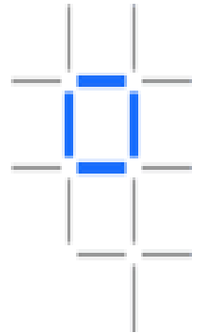
Ledger

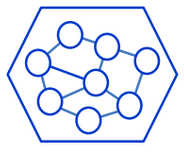
The ledger containing history of submitted votes

Smart Contract

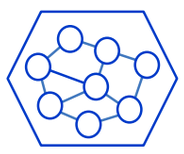
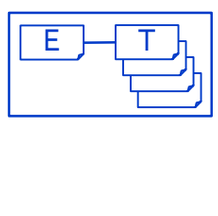
voterContract. Registers voters & submits votes

Components in an e-voting blockchain solution

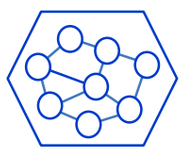
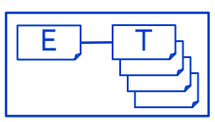


Ledger		The ledger containing history of submitted votes
Smart Contract		voterContract. Registers voters & submits votes
Peer Network		The peers which run the voterContract

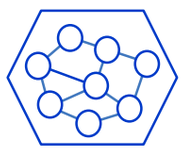
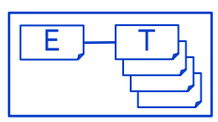
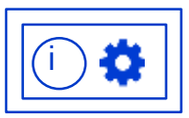
Components in an e-voting blockchain solution

Ledger		The ledger containing history of submitted votes
Smart Contract		voterContract. Registers voters & submits votes
Peer Network		The peers which run the voterContract
Membership		Public and private key for each registered voter

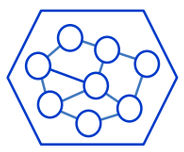
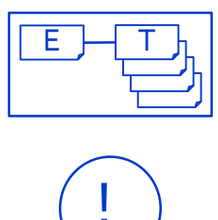
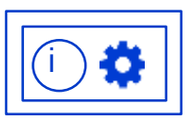
Components in an e-voting blockchain solution

Ledger		The ledger containing history of submitted votes
Smart Contract		voterContract. Registers voters & submits votes
Peer Network		The peers which run the voterContract
Membership		Public and private key for each registered voter
Events		Emit events to when a transaction is complete

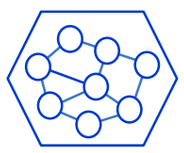
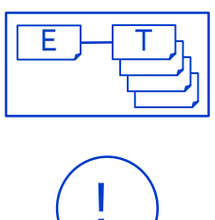
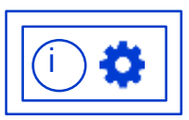
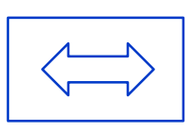
Components in an e-voting blockchain solution

Ledger		The ledger containing history of submitted votes
Smart Contract		voterContract. Registers voters & submits votes
Peer Network		The peers which run the voterContract
Membership		Public and private key for each registered voter
Events		Emit events to when a transaction is complete
Systems Management		VSCode extension to manage nodes & network

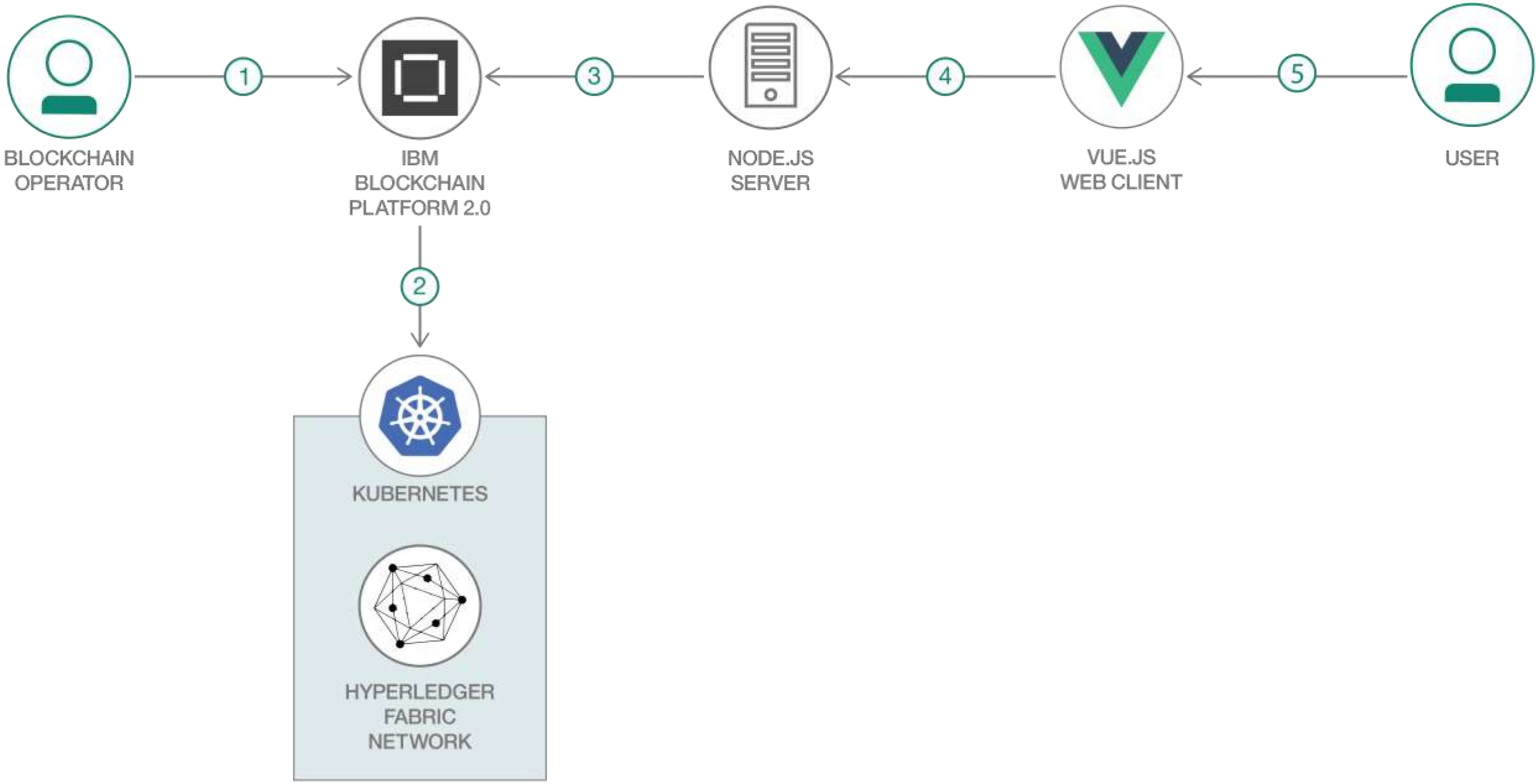
Components in an e-voting blockchain solution

Ledger		The ledger containing history of submitted votes
Smart Contract		voterContract. Registers voters & submits votes
Peer Network		The peers which run the voterContract
Membership		Public and private key for each registered voter
Events		Emit events to when a transaction is complete
Systems Management		VSCode extension to manage nodes & network
Wallet		Stores our voter's public/private keys and certs

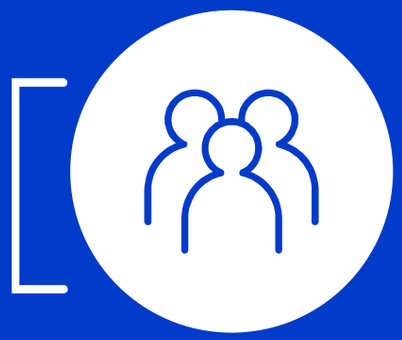
Components in an e-voting blockchain solution

Ledger		The ledger containing history of submitted votes
Smart Contract		voterContract. Registers voters & submits votes
Peer Network		The peers which run the voterContract
Membership		Public and private key for each registered voter
Events		Emit events to when a transaction is complete
Systems Management		VSCode extension to manage nodes & network
Wallet		Stores our voter's public/private keys and certs
Systems Integration		An API to validate voter registration (DMV API)

App Architecture

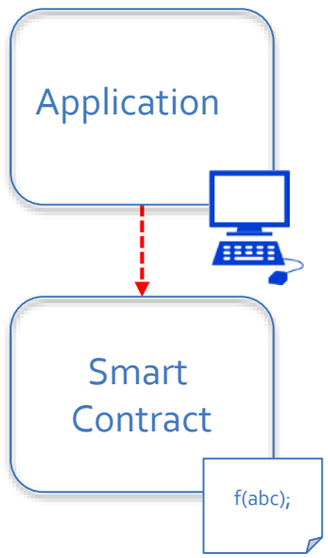


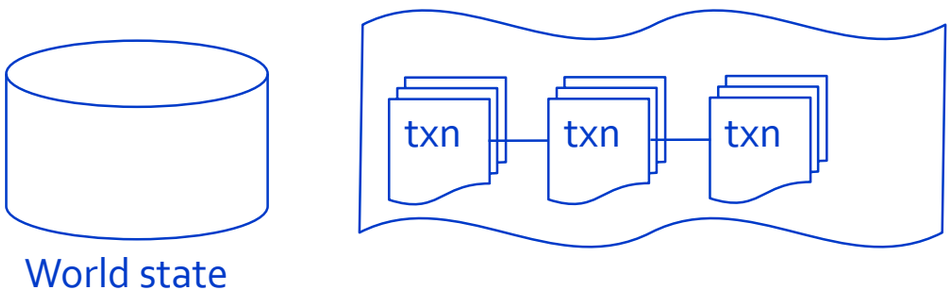
Concepts and Components



Considerations for the Developer,
Operator and Architect

Working with the ledger example: a change of ownership transaction

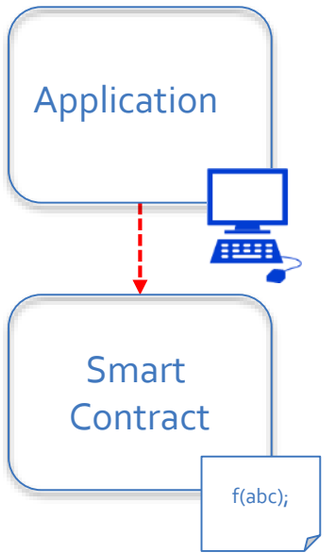


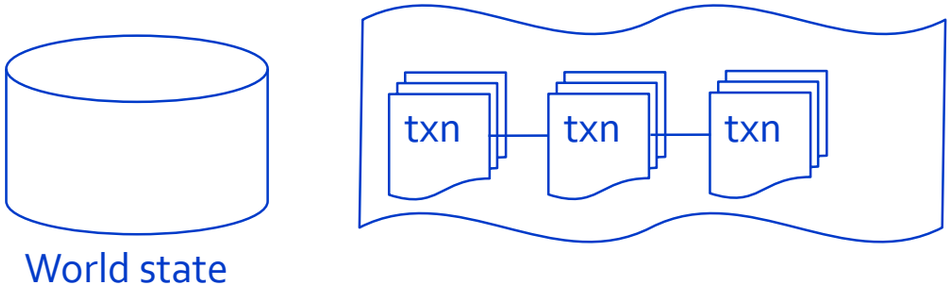


Transaction input - sent from application

invoke(voterContract, castVote, *2020election*, *123123123*, *democrat*)

Working with the ledger example: a change of ownership transaction





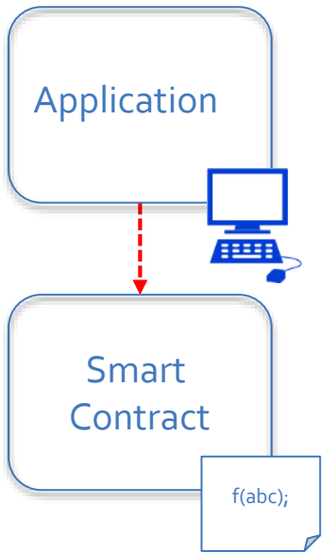
Transaction input - sent from application

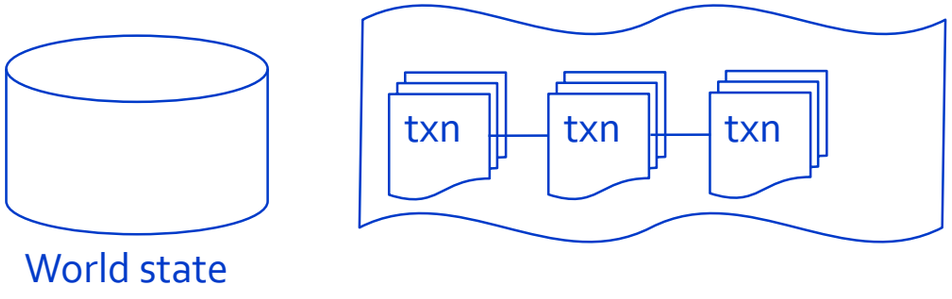
invoke(voterContract, castVote, *2020election*, *123123123*, *democrat*)

Smart contract implementation

```
castVote(ctx, args) {
  args.democrat.count++
}
```

Working with the ledger example: a change of ownership transaction





Transaction input - sent from application

invoke(voterContract, castVote, *2020election*, *123123123*, *democrat*)

Smart contract implementation

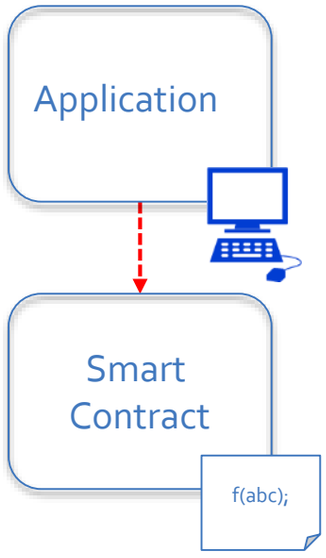
```
castVote(ctx, args) {
  args.democrat.count++
}
```

World state: new contents

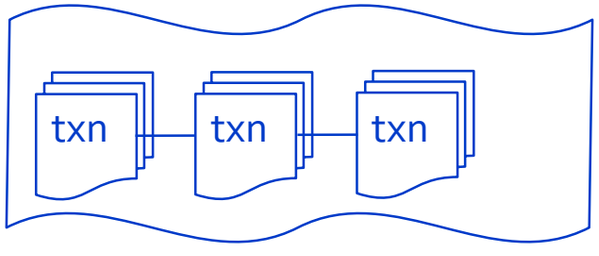
democrat.count = 1

voterId.castBallot = true

Working with the ledger example: a change of ownership transaction



World state



Transaction input - sent from application

invoke(voterContract, castVote, *2020election*, *123123123*, *democrat*)

Smart contract implementation

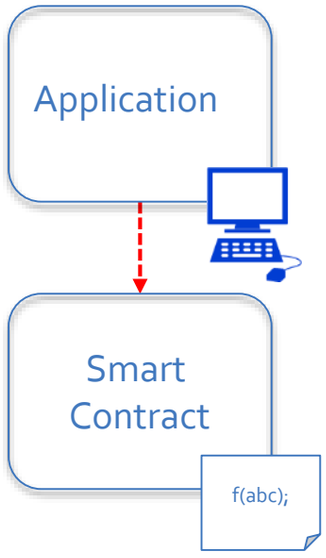
```
castVote(ctx, args) {
  args.democrat.count++
}
```

World state: new contents

democrat.count = 1

voterId.castBallot = true

Working with the ledger example: a change of ownership transaction

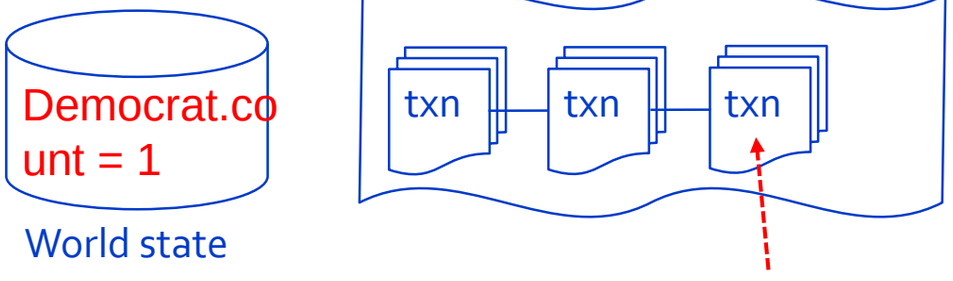


Transaction input - sent from application

invoke(voterContract, castVote, *2020election*, *123123123*, *democrat*)

Smart contract implementation

```
castVote(ctx, args) {
  args.democrat.count++
}
```



World state: new contents

democrat.count = 1

voterId.castBallot = true

"Invoke, voterContract, castVote, 2020election, 123123123, democrat"

E-voting Demo

What we learned

1. Blockchain vocabulary

2. A use-case of blockchain

3. Architecture behind a Hyperledger Fabric solution

4. Difference between the world state and the ledger

5. Difference between public and private blockchains

6. How a client application invokes a smart contract

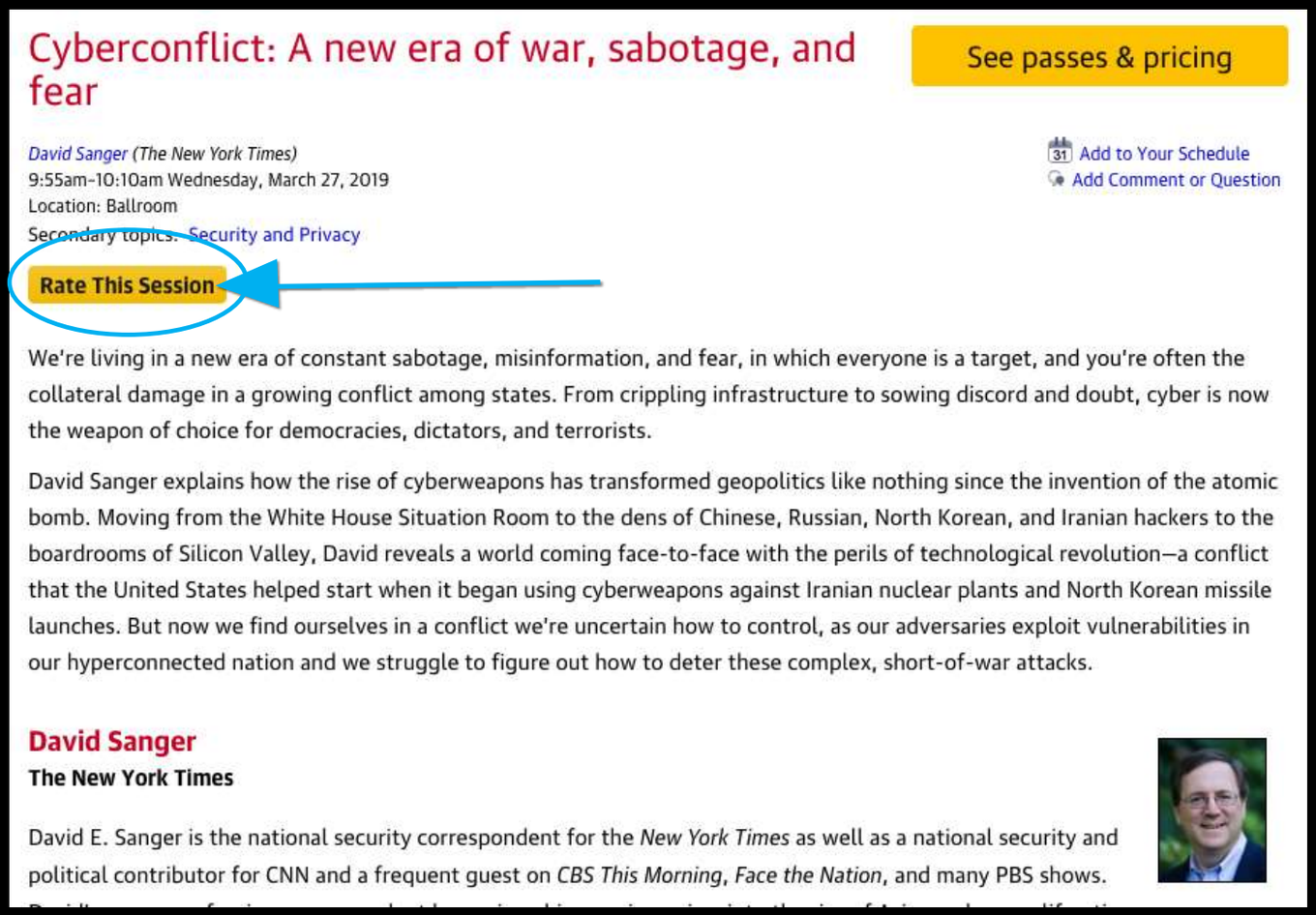
Contact

Questions!?: horea.porutiu@ibm.com

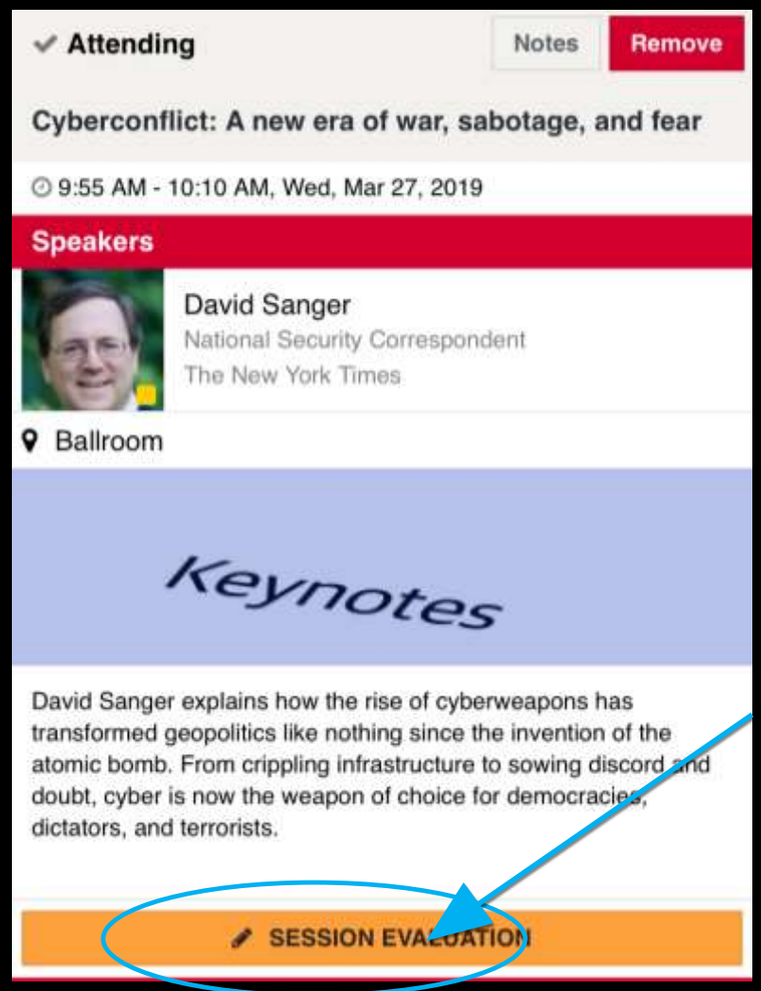
Twitter: @horeaporutiu

YouTube📹: horeaporutiu

Rate today’s session



Session page on conference website



O’Reilly Events App