

BUILDING SMART CONTRACTS ON PUBLIC CLOUD THAT PAY

Srini Karlekar – Director Specialist, Software Architecture, KPMG.

Twitter: @skarlekar

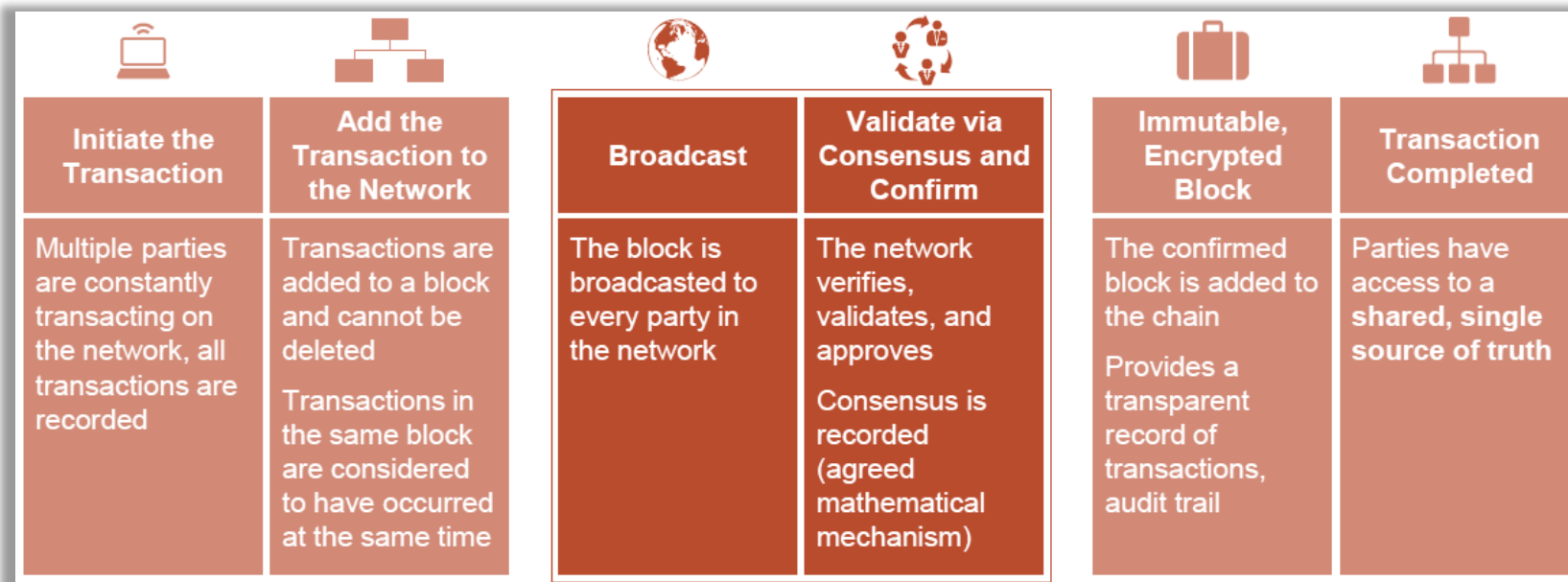
LinkedIn: skarlekar



SMART CONTRACTS

DEMYSTIFYING BLOCKCHAIN

- Blockchain, or distributed ledger technology, is a digital record of transactions and ownership that is continually reconciled and replicated among many different nodes in a peer-to-peer network. Each transaction is uniquely signed with a private key.
- This "chain" maintains a continuously growing list of records, called blocks, which are inherently resistant to modification – once recorded, the data cannot be altered retroactively.






HOW BLOCKCHAIN SOLVES BUSINESS PAIN POINTS



- Blockchain technology can simplify the management of trusted information such as information about individuals, organizations, their assets and activities making it easier for organizations to access and use critical information while enabling individuals and organizations to authorize who can access them while maintaining the security of the information.

Transparency






Increasing customer & shareholder demands for transparency into how funds are being spent and customer data is being used and shared

Data Sharing

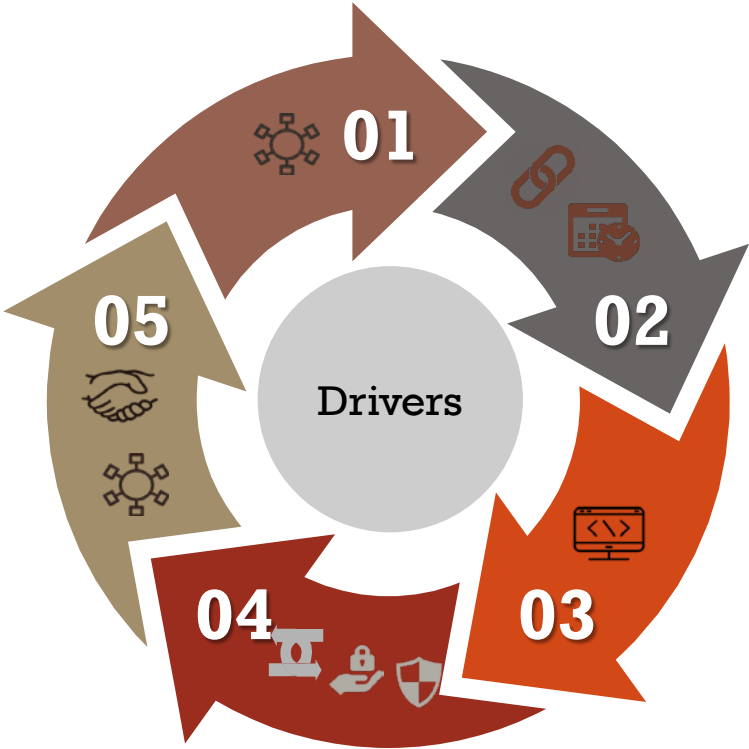



Increasing demand for data consistency and sharing of data among business partners

Confidentiality




Growing concerns around trust, security and keeping personally identifiable information private.



Provenance & Lineage

Increasing interest for recording asset ownership, lineage of data used for supporting its ownership, valuation, depreciation, disposal and changes to its location over time while improving system interfaces to access this information

Regulatory Compliance

Increasing need to efficiently create, update, and enforce regulations or resolve disputes with use of tamper-proof mechanisms and improve the resiliency of public sector transactions

The distributed nature of blockchain technology allows for a high degree of transparency and accountability, which is essential for businesses to maintain trust and integrity. The distributed nature of blockchain technology allows for a high degree of transparency and accountability, which is essential for businesses to maintain trust and integrity. The distributed nature of blockchain technology allows for a high degree of transparency and accountability, which is essential for businesses to maintain trust and integrity.



SMART CONTRACTS - PUTTING BLOCKCHAIN TO USE

- Smart Contracts are self-executing contracts with terms of agreements between counterparties that is written into code on the Blockchain and is triggered by an event. As a result, Smart Contracts secure, enforce and execute settlement of recorded agreements in a predefined manner within a secure boundary without the need for a third-party involvement and verification.



**AN ATTEMPT TO DEMYSTIFY SMART
CONTRACTS USING A MOVIE THEATER
TRANSACTION**



MOVIE THEATER EXAMPLE - 1

Contract: Usher exchanging money for allowing entry into the theater.

Implication:

1. Allows entry.
2. Not enough seats.
3. Cannot reserve seats.
4. Is the customer using authentic currency to pay?



MOVIE THEATER EXAMPLE - 2

Contract: Purchasing tickets over-the-counter.

Implication:

1. Solves issues from case 1, but the trust issue still exists.
2. Is the ticket a duplicate of one that is already issued?
3. Will the customer exit the premise after watching one movie?

Point #2 is not usually an issue because customers trust the theater management, otherwise the theater is going to loose business.

Here the ticket acts as a token for proving that the customer has paid for the screening. Trust is one way in this scenario.



MOVIE THEATER EXAMPLE - 3

Contract: Purchasing tickets online.

Implication:

1. Customers have to trust the website selling tickets is associated with the theater.
2. Is the website using certificates issued by a trusted authority?
3. If I have to sell the ticket to a third-party, will the buyer trust that I am selling a valid ticket?
4. Even if it is a valid ticket, what is the assurance that I am not double-selling the same tickets to multiple parties?



HOW SMART CONTRACTS FIT IN

1. Transactions are signed by both buyer and seller using private keys.
2. Within the transaction buyer sends crypto to seller.
3. Seller provides a non-fungible token that contains the identity of the buyer.
4. A transaction can only use crypto from a previous transaction – which has already been verified.
5. Transactions are verified by multiple nodes in the Blockchain.
6. Movie theater allows entry for the customer after verifying the identity on the token with the customer's digital identity.
7. Once the token is presented, it is marked as used and the transaction logged.



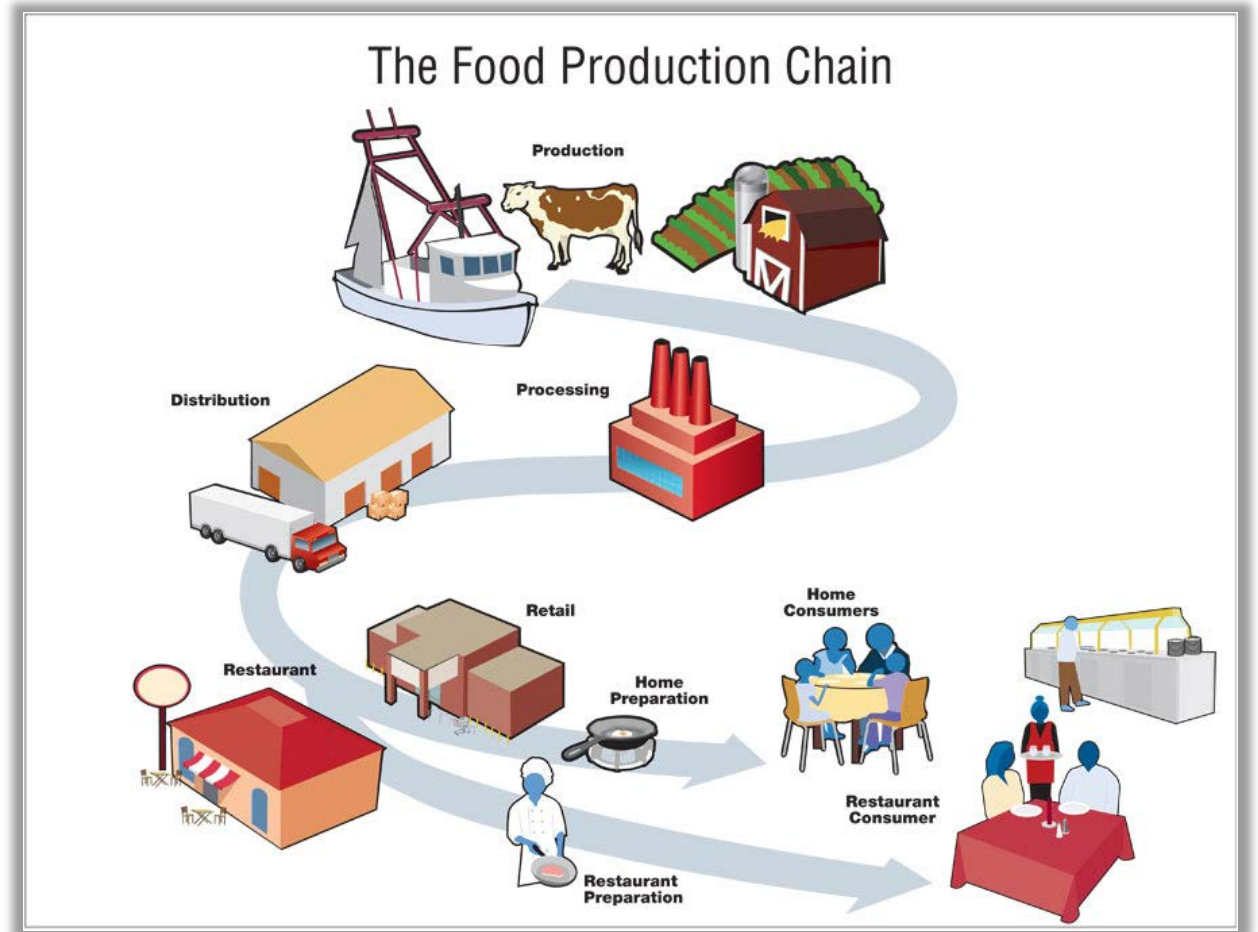
AN ATTEMPT TO DEMYSTIFY SMART CONTRACTS USING A SUPPLY CHAIN EXAMPLE



FOOD SUPPLY CHAIN EXAMPLE

Challenges:

1. Tracking the product and its state.
2. Customs check.
3. Bureaucracy & lack of trust among parties.
4. Multiple data bases
5. Expensive data migrations.
6. No clear authority.
7.



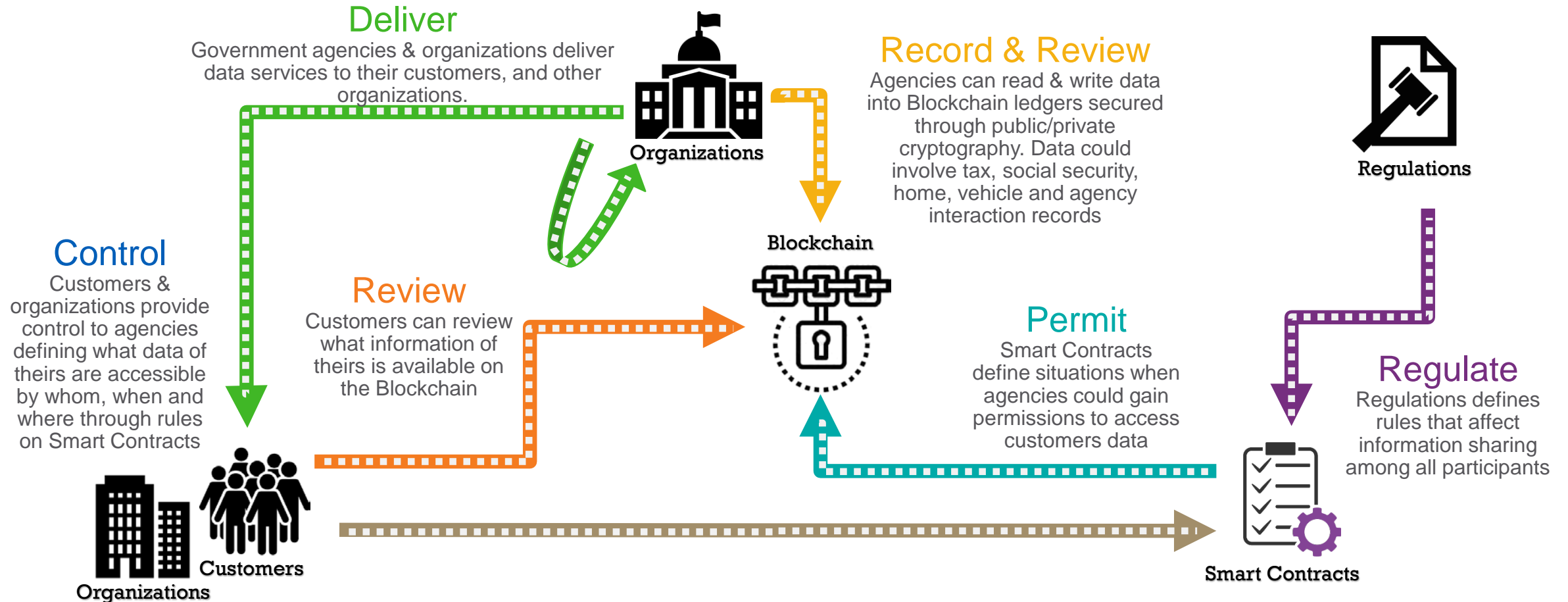
HOW SMART CONTRACTS FIT IN

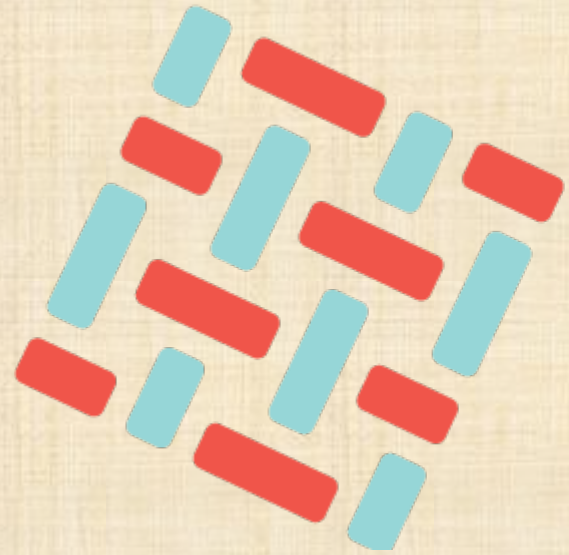
1. Buyer and seller enters an escrow-style agreement. Buyer puts crypto in an escrow account. Escrow is released to the seller after buyer receives the product based on the condition of the product.
2. Raw material serial numbers are tokenized and new non-fungible tokens are generated using these tokens when products are assembled (see <https://cryptokitties.io> & <https://cryptozombies.io/>)
3. Data is maintained in Blockchain and can be updated only using Smart Contracts.
4. Audit-trail from farm-to-table.



BETTER BUSINESS THROUGH NETWORKED SERVICES

Blockchain technologies such as Distributed Ledgers and Digital Identity can be used by organizations and public sector to provide networked services to customers. Customers could define which agencies can have access to their data and when.





HYPERLEDGER **FABRIC**

HYPERLEDGER FABRIC

Hyperledger Fabric is a blockchain framework implementation and one of the Hyperledger projects hosted by The Linux Foundation.

Intended as a foundation for developing applications or solutions with a modular architecture, Hyperledger Fabric allows components, such as consensus and membership services, to be plug-and-play.

Hyperledger Fabric leverages container technology to host smart contracts called “chaincode” that comprise the application logic of the system.

Hyperledger Fabric was initially contributed by Digital Asset and IBM.

CREATED FOR DEVELOPING ENTERPRISE APPLICATIONS

ANATOMY OF HYPERLEDGER FABRIC BNA

In the world of Hyperledger Fabric, DApps are called BNA or Business Network Applications.

Participants - Participants represents users who interact with assets.

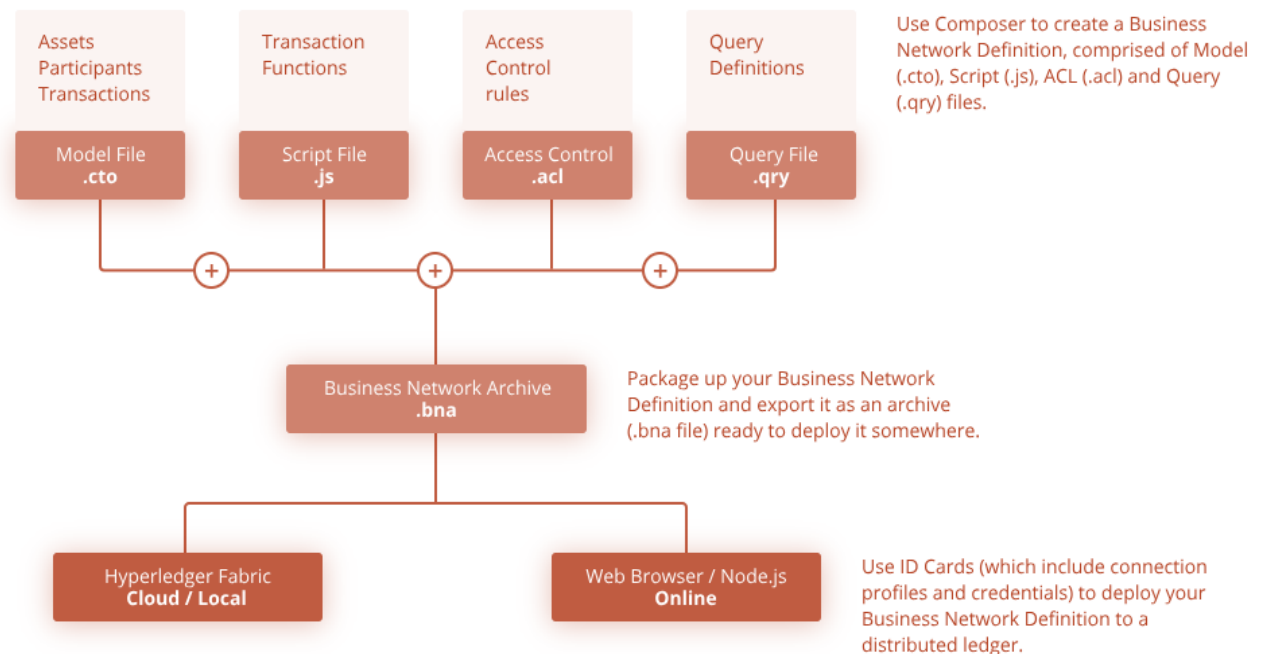
Assets - Assets represents entities which could represent place or things.

Transactions - Transactions are actions that participants can carry out on assets.

Events - Events are emitted by Hyperledger Composer as a result of transactions.

Queries - SQL-like queries that can be used to search for assets based on its attributes.

Access Control - Access Control provides declarative access control over the elements of the domain model.

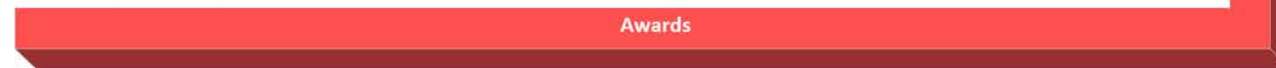
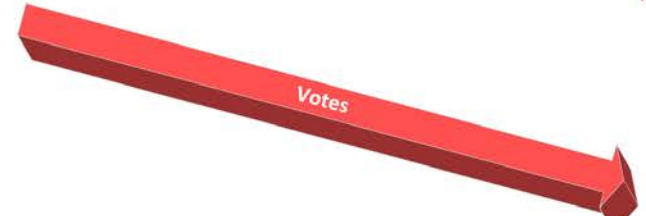


SMARTQUORA DOMAIN MODEL

Participants

Transactions

Assets



SMARTQUORA PARTICIPANTS

There are three types of Participants in SmartQuora BNA.

- Inquirers
- Responders
- Administrators

Inquirers and Responders are represented as QuoraUsers in the application because their function can interchange - an inquirer can respond to questions from other inquirers or a respondent for a question can pose his/her own questions.

Participants maintain tokens that is placed as a stake when asking questions.

SMARTQUORA RULES

The following rules are enforced by the SmartQuora smart contracts:

- Inquirers cannot answer their own questions.
- Responders cannot vote for their own answers.
- Responders cannot vote more than once for the same answer.
- Answers will not be accepted after the due date.
- Only administrators can award questions and distribute the reward among the voted answers after the due date.

SMARTQUORA ASSETS

There are two types of Assets modeled in the SmartQuora application. They are:

- **Question** - A question consists of an unique id, question description, owner, status (CREATED, ANSWERED, AWARDED, or DEFAULTED), reward amount and a list of answers. If a question is answered and is voted for, the stake is equally distributed among the owners of the voted answers.
- **Answer** - An answer consists of an unique id, answer description, owner, status (CREATED, VOTED, AWARDED), earnings, and a list of voters. When a question is awarded, the earnings attribute reflect the earnings that was generated by that particular answer for the respondent.

SMARTQUORA TRANSACTIONS

- **CreateQuestion** - A *CreateQuestion* transaction is invoked by the Inquirer to pose a question. The reward amount and time by which answers are due has to accompany the request.
- **CreateAnswer** - A *CreateAnswer* transaction is invoked when a respondent provides an answer to an existing question. This transaction ensures that the question owners cannot answer their own questions.
- **VoteAnswer** - A *VoteAnswer* transaction is invoked to vote up or vote down an existing answer. This transaction ensures that respondents cannot vote for their own answers or vote multiple times for an answer.
- **AwardQuestion** - A *AwardQuestion* transaction is invoked to find out the highest voted answers and distribute the reward proportionately amongst the voted answers.

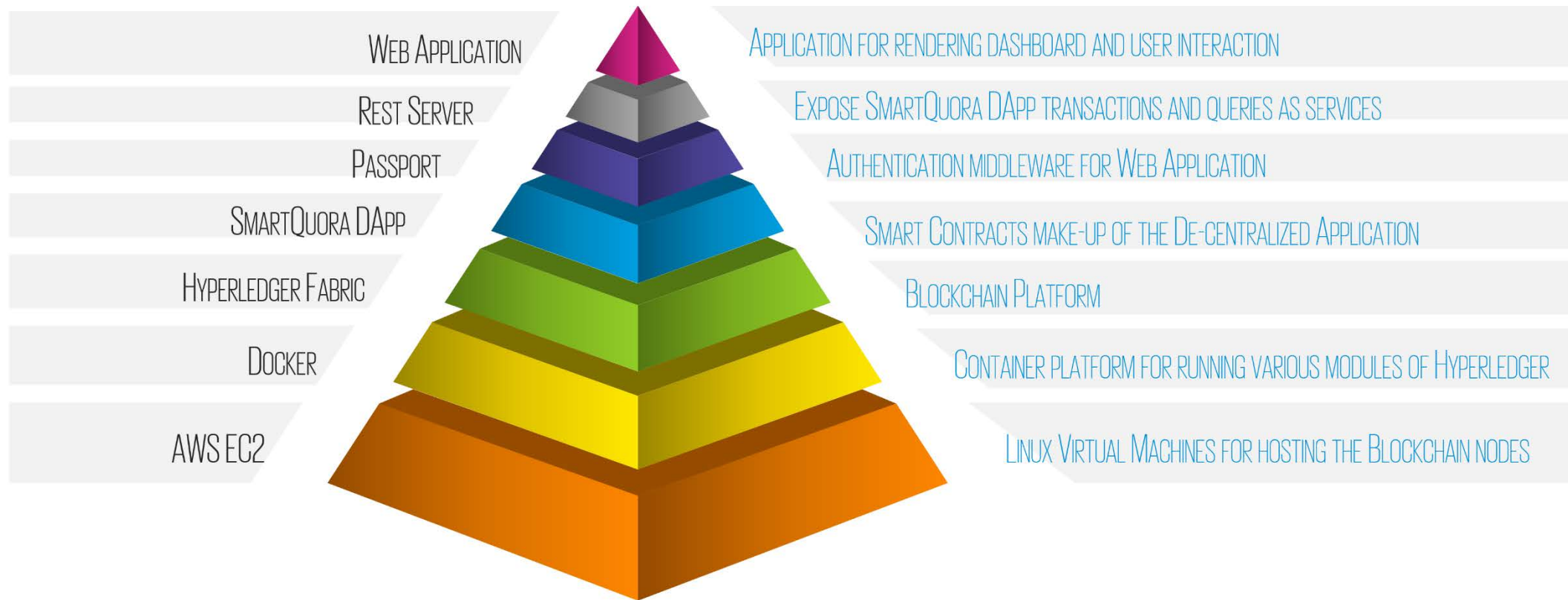
SMARTQUORA EVENTS

Events in a Hyperledger Fabric BNA is pushed on a Web Socket. Any system listening to the Web Socket will get notified of all the events. As a result, the listeners have to filter out events that are not relevant to it.

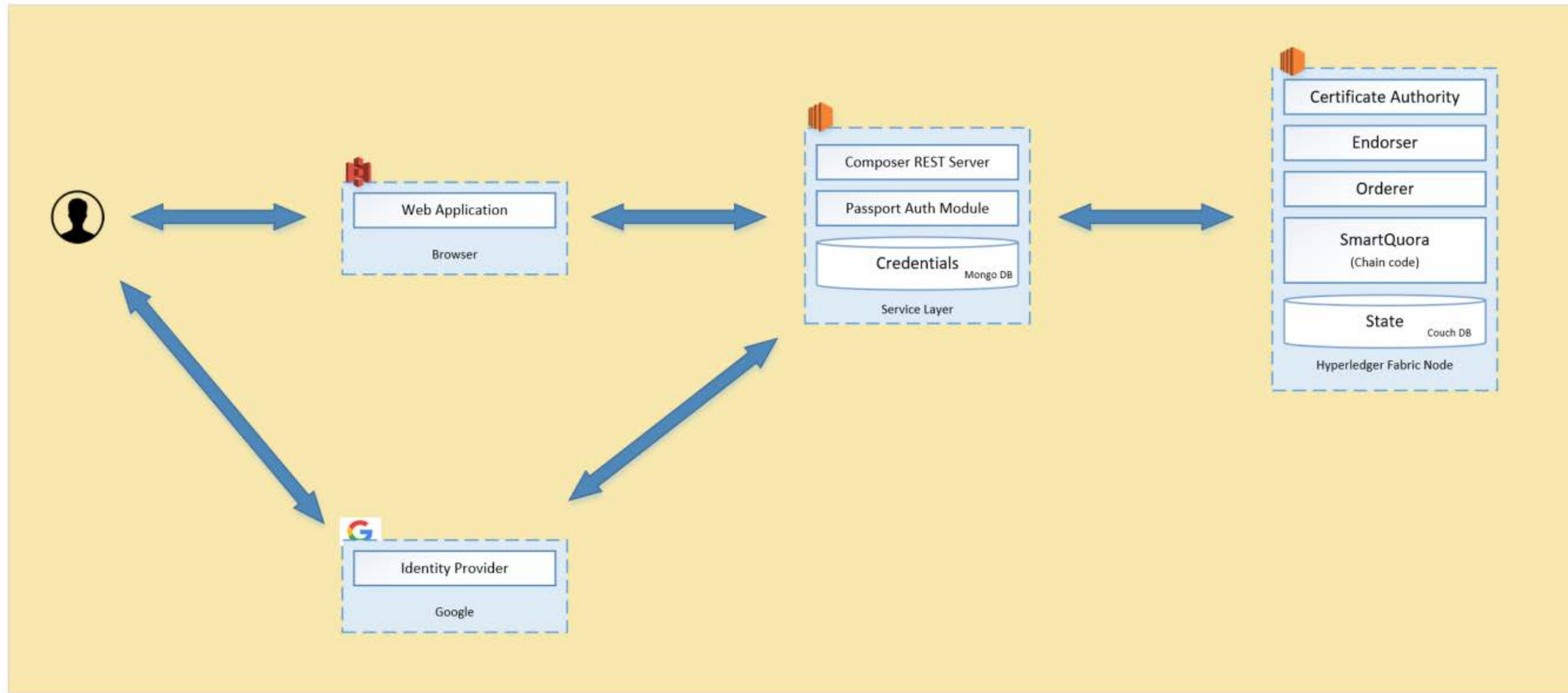
SmartQuora events are as follows:

- **QuestionCreated** - Generated by the *CreateQuestion* transaction.
- **AnswerCreated** - Generated by the *CreateAnswer* transaction.
- **AnswerVoted** - Generated by the *VoteAnswer* transaction.
- **QuestionAwarded** - Generated by the *AwardQuestion* transaction.

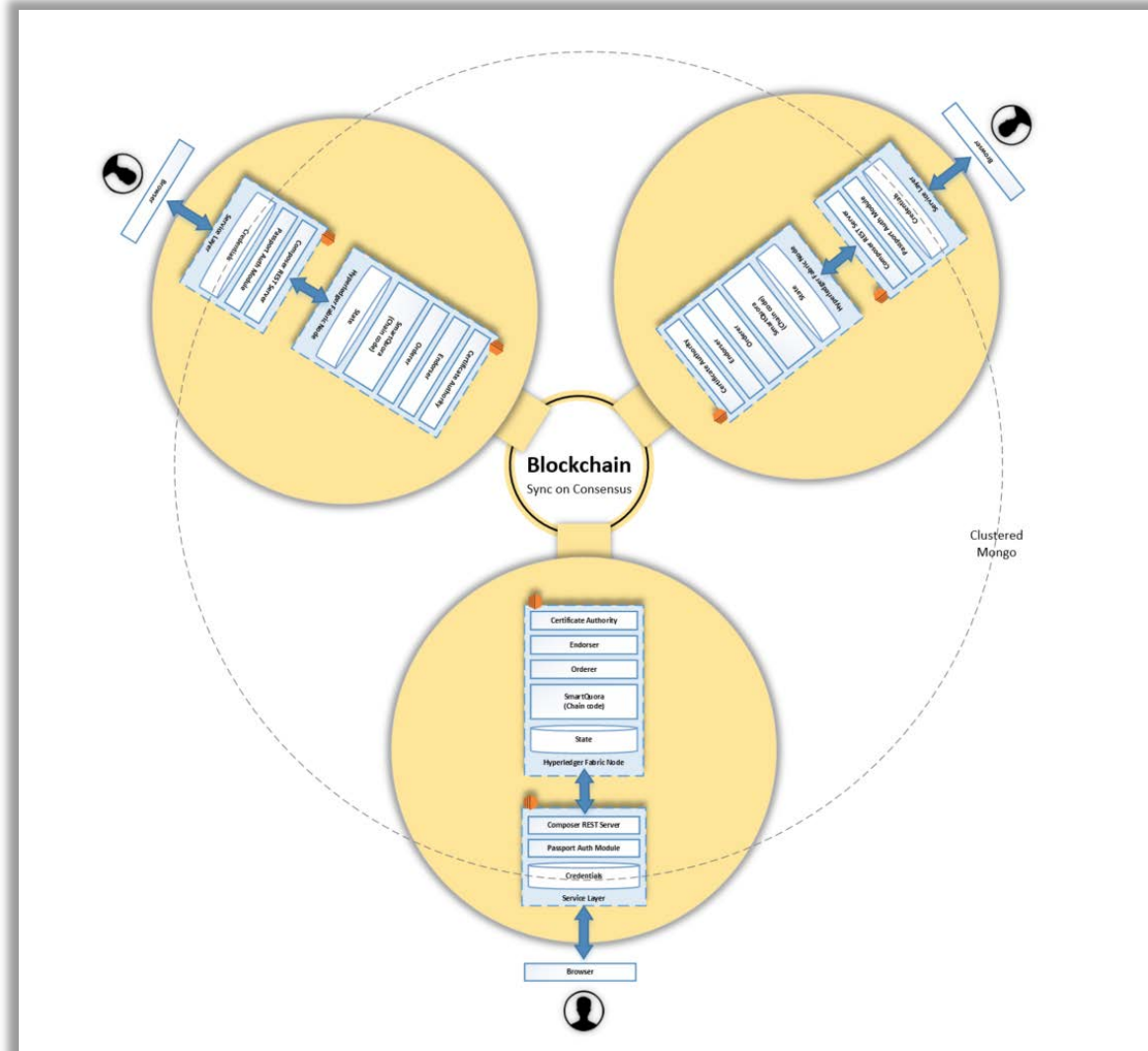
SMARTQUORA SYSTEM COMPONENTS



SMARTQUORA ARCHITECTURE



SMARTQUORA ARCHITECTURE



SMARTQUORA - APPLICATION DEMO

SMARTQUORA – CODE WALK-THROUGH

The SmartQuora application can be cloned from:

<https://github.com/skarlekar/smart-quora>

For Installation, Deployment and Usage instructions go to:

<http://bit.ly/sqreadme>

FURTHER READING

To understand the concept of Blockchain and Smart Contracts and explore the differences between various Blockchain frameworks go to:

<http://bit.ly/blkchn2018>