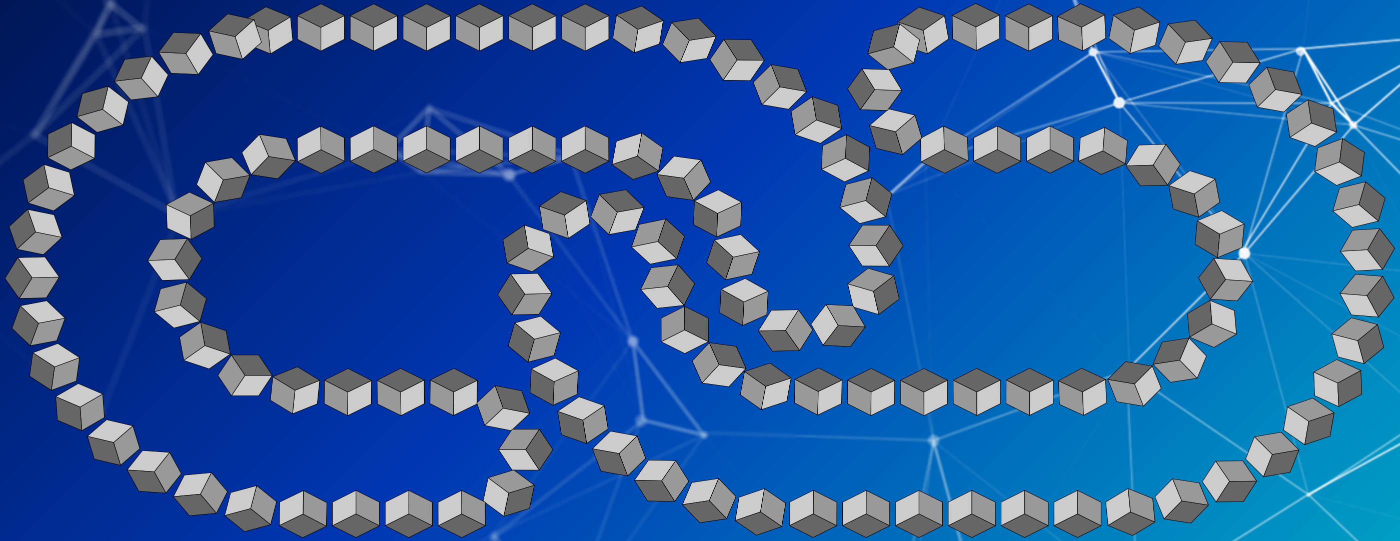
Blockchain Project

Storing and retrieval of Rocket Chat message on Quorum Blockchain



By Arpan Sur (391842) & Samuel Roy (391822)

Image by Gordon Johnson from Pixabay

Some Basics about Blockchain:

- Blockchain is a **decentralized peer to peer** network.
- Blockchain stores the transactions in the form of blocks.
- Once a transaction is sent to blockchain, the nodes in the blockchain check for the validity of the transaction.
- Once validity is established every node in blockchain is distributed a copy of the ledger of all transactions.

Some Basics about Blockchain:

- The transactions are stored in the blockchain using a hash function

- These blocks contain multiple transactions and are stored as chain in the form of hash pointer linked list of blocks which can be appended sequentially at the end. (Hash tree, Merkel tree).

Types of Blockchain:

- Private Blockchain
- Public Blockchain

Private Blockchain:

- It refers to permissioned blockchain platforms
- This type of blockchain places restriction on who can participate in the network.
- Private blockchain are limited to an organization
- It is important in Private blockchain to define access rights and check identity of a user before giving access.
- Examples: Hyperledger, Hashgraph

Public Blockchain:

- It refers to permission-less blockchain platforms
- It is a public system that anyone with a copy can access it.
- Public blockchain are decentralized and no one has a control over the network.
- This is secure because the data cannot be changed once validated by full node.
- Examples: Ethereum, Bitcoin

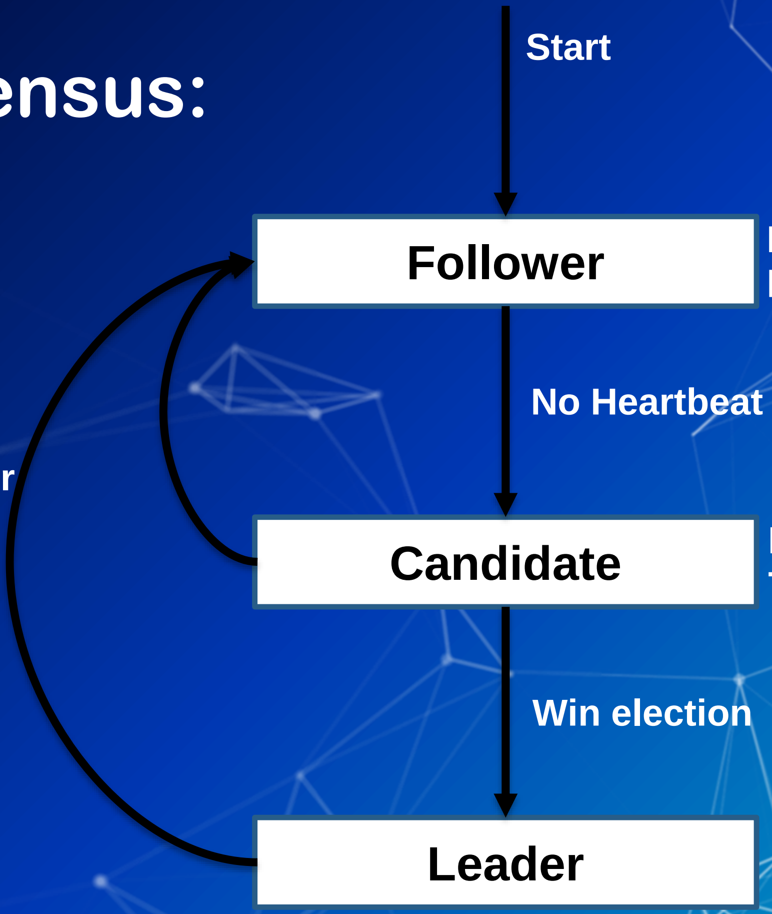
How to reach the Consensus:

- Consensus must be reached between the nodes on which transactions should be included in the blockchain.

- Proof of Work is a Consensus Mechanism in which the nodes (miners) solve a cryptographic puzzle in order to add a transaction to the chain.

- The validation is carried out by nodes called as miners, who also append the transaction to the chain.

Raft Consensus:



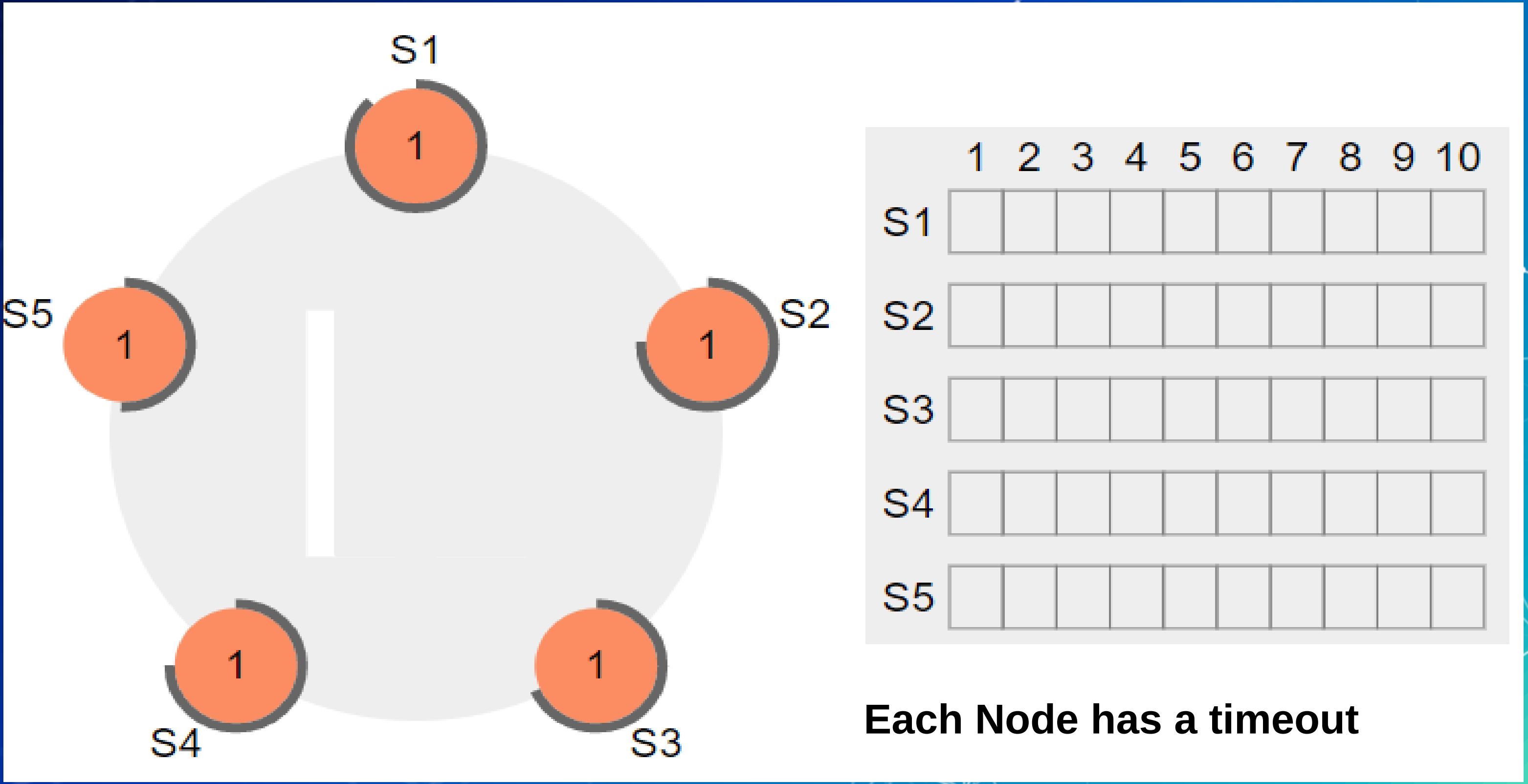
Passive (but expects regular heartbeats

Issues RequestVote RPCs
To get elected as leader

Issues AppendEntries RPCs:
Replicate its log and Heartbeats
to maintain leadership

https://www.youtube.com/watch?v=vYp4LYbnnW8&feature=youtu.be

Raft Consensus:

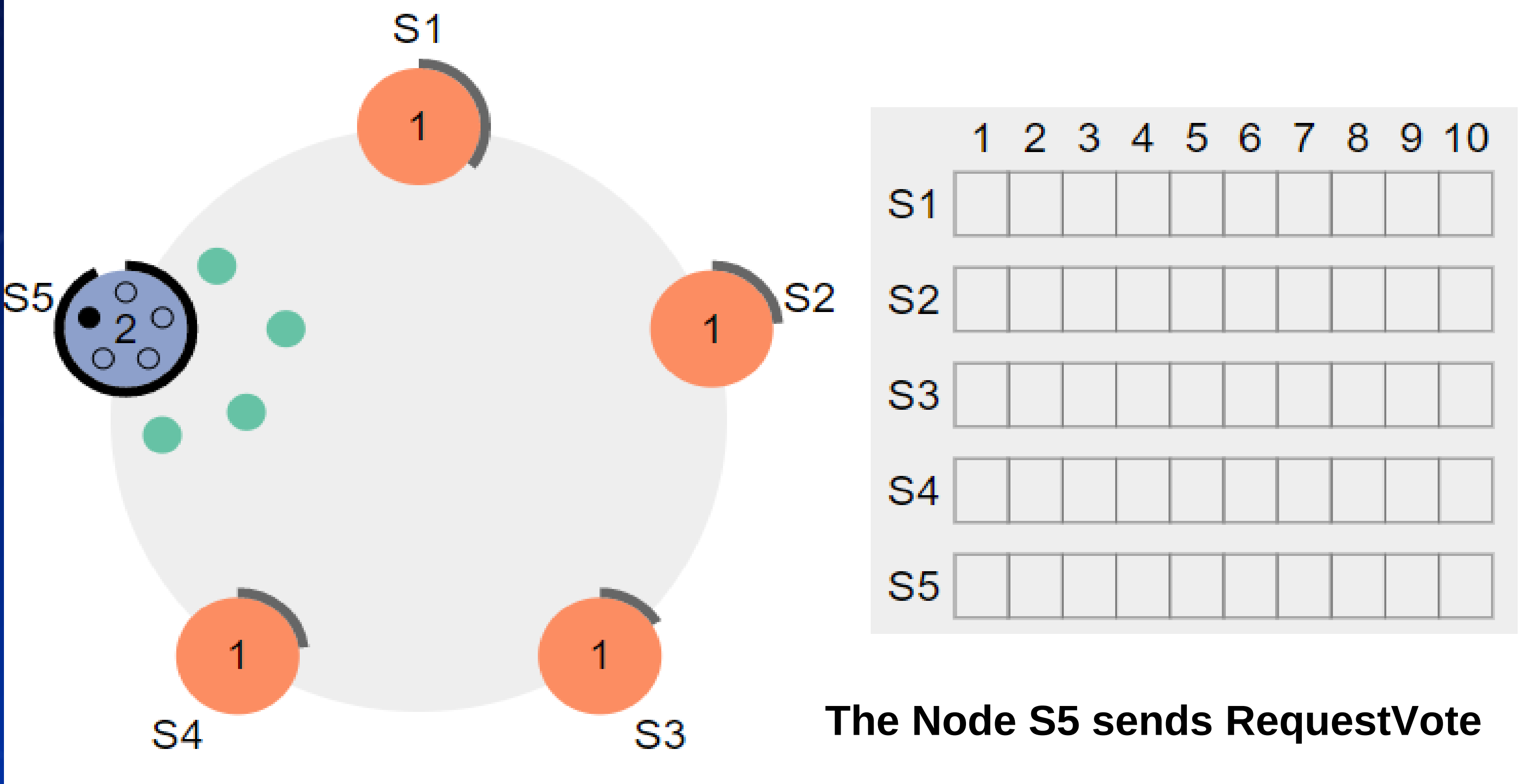


	1	2	3	4	5	6	7	8	9	10
S1										
S2										
S3										
S4										
S5										

Each Node has a timeout

https://raft.github.io/

Raft Consensus:

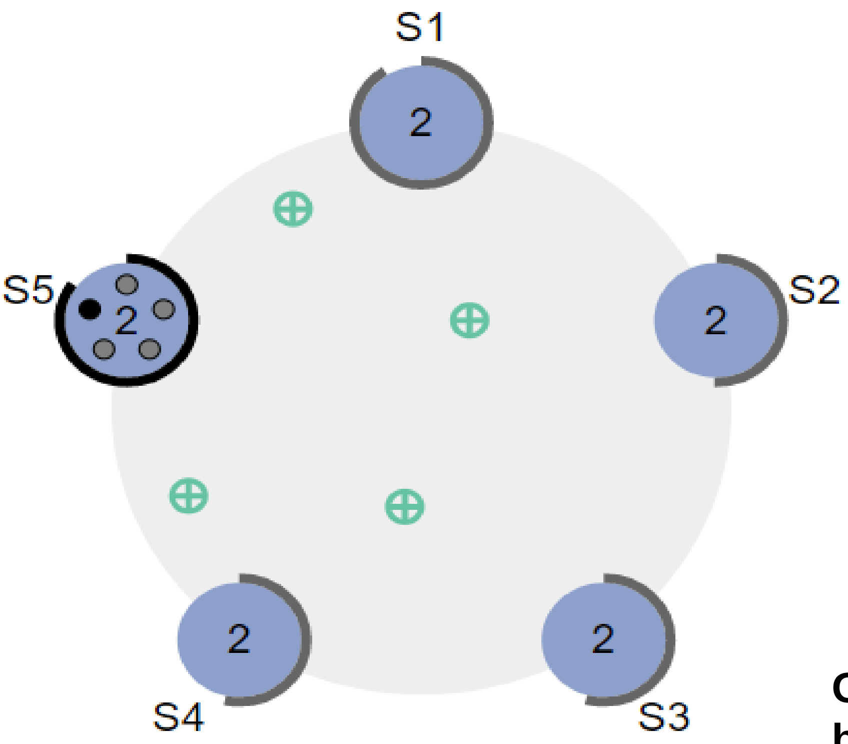


	1	2	3	4	5	6	7	8	9	10
S1										
S2										
S3										
S4										
S5										

The Node S5 sends RequestVote

https://raft.github.io/

Raft Consensus:

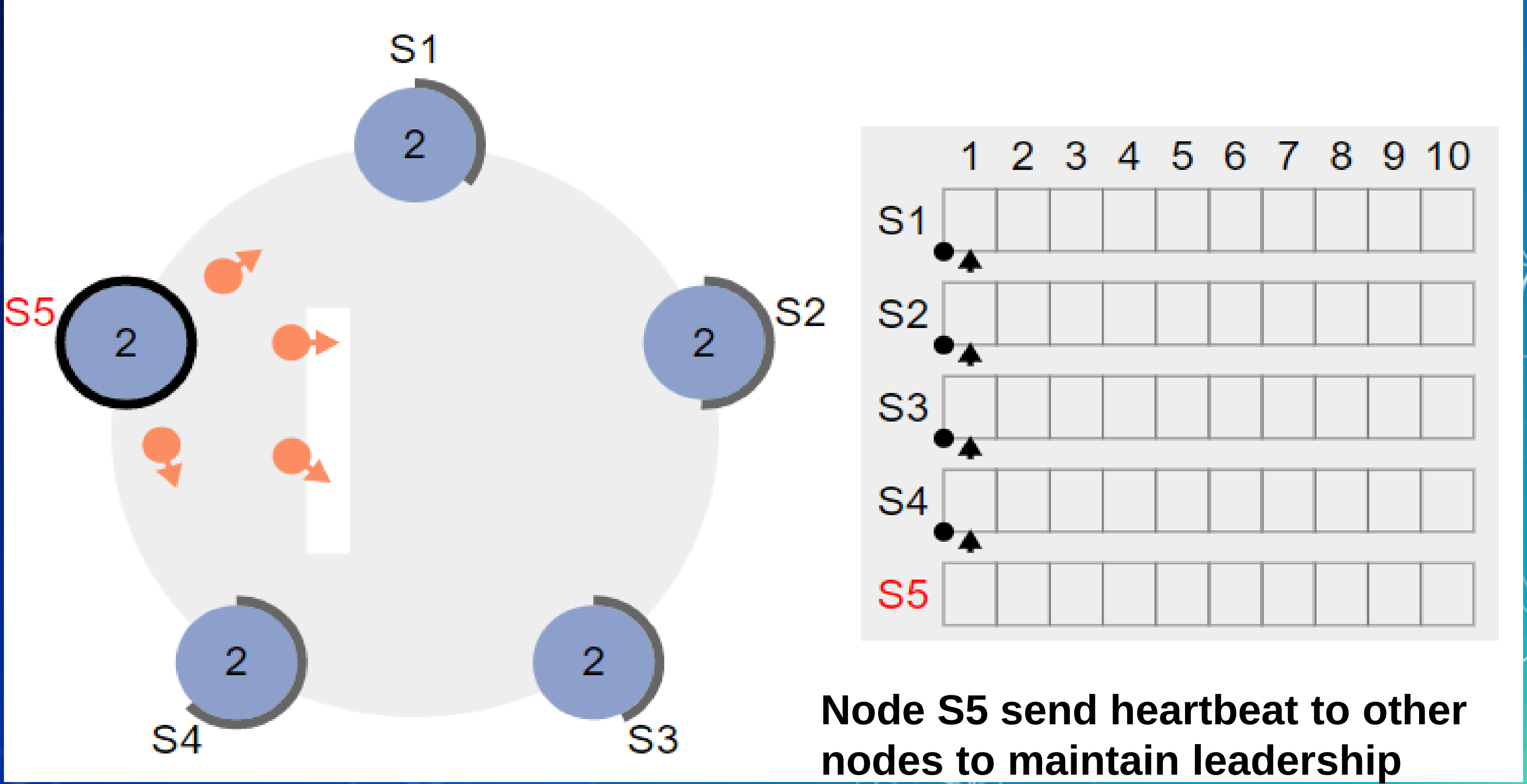


	1	2	3	4	5	6	7	8	9	10
S1										
S2										
S3										
S4										
S5										

Other Nodes sends votes and S5 becomes the leader

https://raft.github.io/

Raft Consensus:

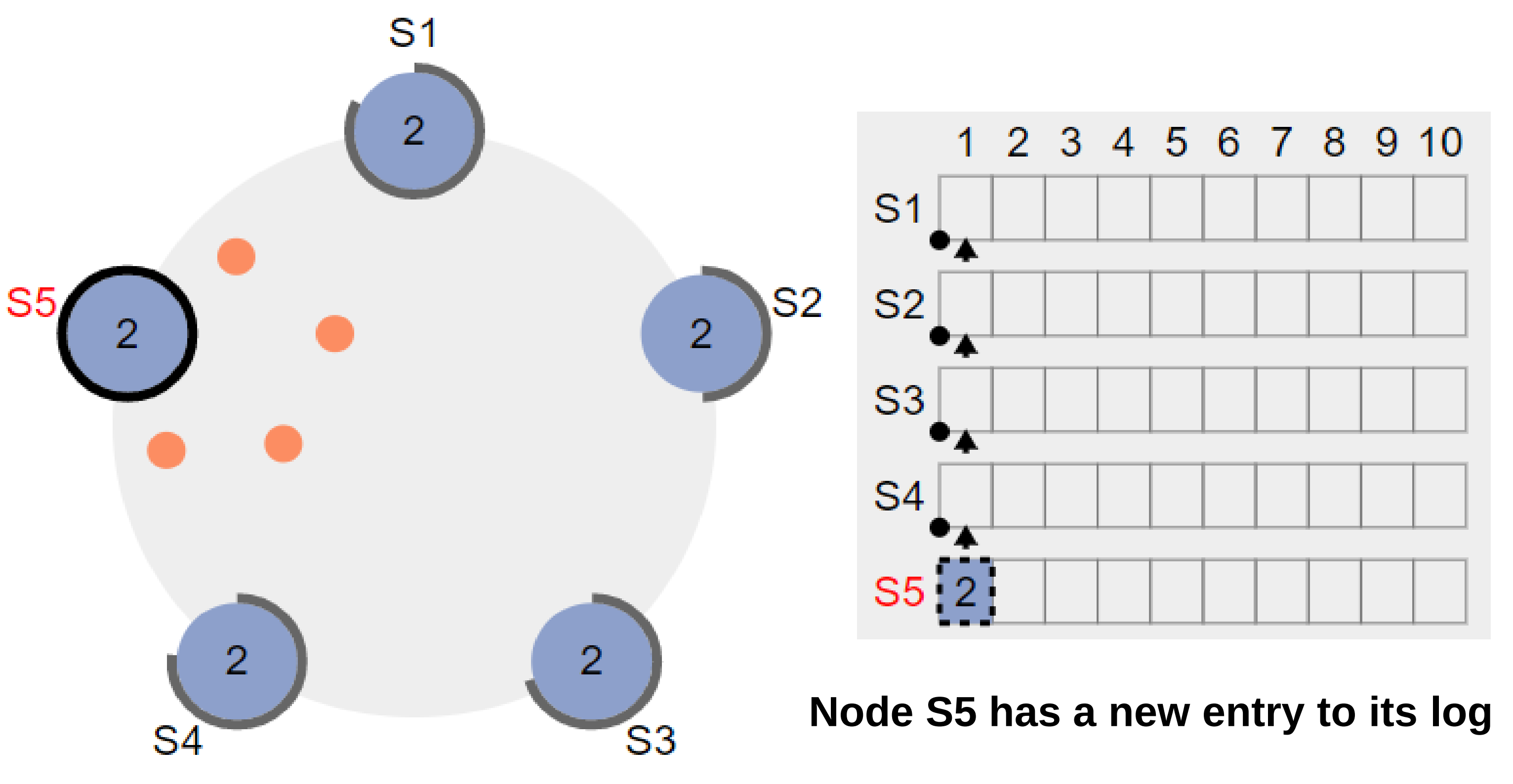


	1	2	3	4	5	6	7	8	9	10
S1										
S2										
S3										
S4										
S5										

Node S5 send heartbeat to other nodes to maintain leadership

https://raft.github.io/

Raft Consensus:

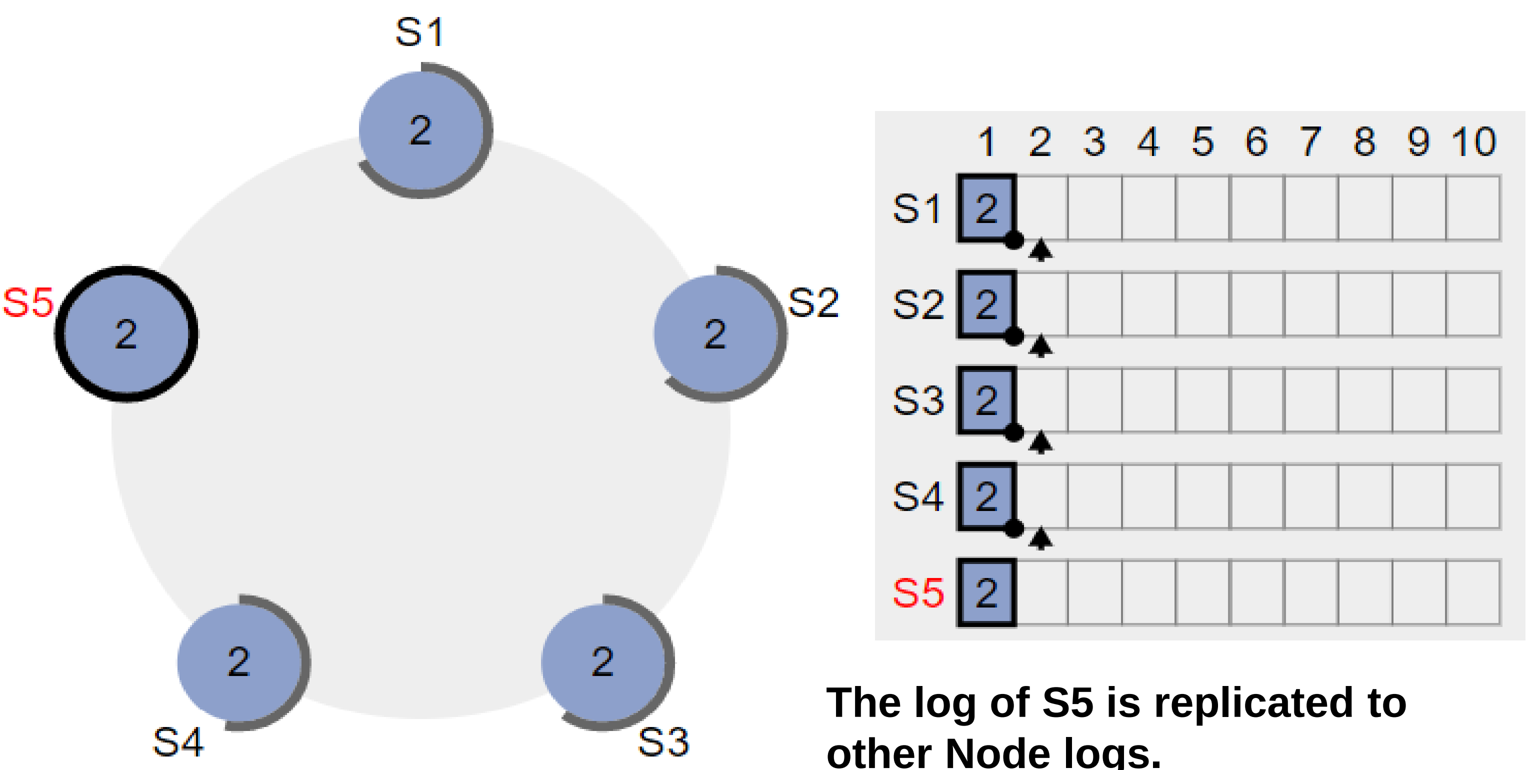


	1	2	3	4	5	6	7	8	9	10
S1										
S2										
S3										
S4										
S5	2									

Node S5 has a new entry to its log

https://raft.github.io/

Raft Consensus:



	1	2	3	4	5	6	7	8	9	10
S1	2									
S2	2									
S3	2									
S4	2									
S5	2									

The log of S5 is replicated to other Node logs.

https://raft.github.io/

Smart Contract:

- Smart Contract is a protocol that is used to verify that the transactions can take place credibly without third party involvement.

- Smart contract defines the processes, regulations and organizational principles and enforces automation of them.

Why Blockchain:

- Data immutability & Irreversibility
- Transparent
- Decentralization
- Consensus Mechanism and Distributed Ledger improves trust

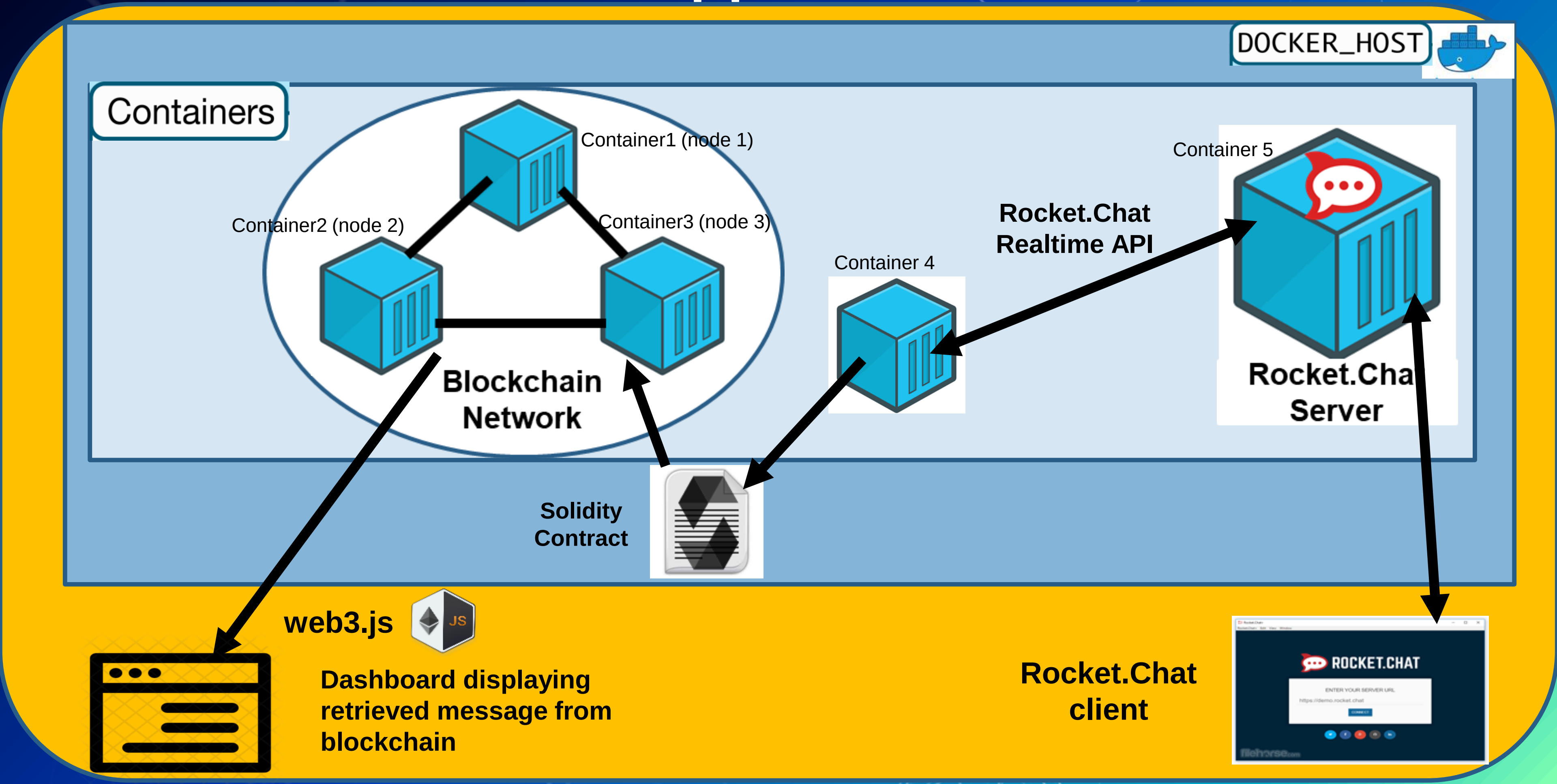
About the Application:

- The Application stores text messages sent on Rocket.Chat in a specific channel on to blockchain.

- The storage mechanism is activated by writing &RecBlock& before the message for a single message to be stored and &On& can be written for storing multiple messages until &Off& is written to a block.

- The stored messages is hashed before storing on blockchain along with sender, receiving room id and timestamp of the stored message. This message is finally retrieved on a dashboard.

Architecture of the Application:



Installation Guide:

- Installation guide is available as a readme file in the application folder RCtoBC.

- The application folder is uploaded as a zip file (RCtoBC.zip) to BSCW in the workspace – “2019 CSCW Blockchain Lecture”

Future and Outlook:

- There are many APIs (Realtime and REST) which could be used to create a Rocket.Chat client of our own.

- Currently the messages are stored on the Rocket.Chat server and then retrieved to be stored on Blockchain. This could be improved to store the messages directly on to blockchain as implemented using Ethereum in project EthChat (https://github.com/austenke/EthChat)

Future and Outlook:

- Instead of the hash message being stored on to Blockchain (due to limitations in storing String using solidity), We can further improve it to store String message.

- Current implementation needs the administrator to be part of the channel to retrieve the message from Rocket.Chat. Further exploration of APIs can be done to retrieve messages also keeping in mind the access rights of each user.

What have we learnt:

- Learnt basics of Blockchain and its implementation in areas of application such as Internet of Things, Smart Grid from the course “Blockchain 360” on coursera [3].

- Understood more about underlying technology in Blockchain, Smart Contracts and opportunities of application of these technology in automation of several areas from “BLOCKCHAIN AND SMART CONTRACTS - Technologies, research issues and applications” [1]

What have we learnt:

- Understood concepts of cryptocurrencies, mechanics of Bitcoin and how Bitcoin achieves decentralization from the book : “Bitcoin and Cryptocurrency Technologies” [4]

- Searched about different Consensus Mechanisms such as Proof of Work and Raft algorithm [2]

- Familiarized the working and necessity of smart contract and implemented simple smart contract with Solidity

What have we learnt:

- Gathered information to start working with Quorum Blockchain and Quorum maker.

- Followed the tutorials in “Simple demo” provided by Professor to create a blockchain using Quorum maker.

- Learnt to manage the docker infrastructure and running instances.

- Learnt to install Rocket.Chat on docker

What have we learnt:

- Worked with Rocket.Chat realtime API to retrieve information from Rocket.Chat
- Familiarized with web3.js and web3.py to save and retrieve information from blockchain
- Implemented an application that can store and retrieve Rocket Chat messages on Quorum Blockchain by extending the simple demo application provided by Professor

Task Performed:

1. Infrastructure setup:
   - Installing docker and docker-compose : Samuel
   - Installing Rocket.chat : Samuel

2. Application setup:
   - Modifications to simple-demo to get it running : Arpan

3. Application Development:
   - Creating new Smart contract using remix IDE : Arpan and Samuel
   - Deployment of smart contract : Arpan
   - Interaction with Rocket chat real time API using web socket : Arpan
   - Switching blockchain recording ON and OFF : Samuel
   - Changes to BC manager for transact and deploy methods: Arpan
   - Storing and retrieving data into blockchain : Arpan and Samuel

Task Performed:

4. Dashboard

  - Dashboard design(html and css): Samuel
  - Javascript to interact with block chain using web3 and to populate the dashboard : Arpan

5. Presentation : Arpan and Samuel

Biblography:

- [1]Weimert, Birgit & Prinz, Wolfgang & Urbach, Nils & Holly, Steffen & Schulte, Axel & Fridgen, Gilbert & Rose, Thomas & Schütte, Julian & Dalheimer, Mathias & Wenzel, Markus & Otto, Boris & Schwede, Christian & Leiner, Ulrich & Fritz, Michael & Kreutzer, Michael & Nouak, Alexander & Hoeren, Thomas & Welzel, Christian & Sprenger, Philipp & Guggenberger, Nikolas. (2018). BLOCKCHAIN AND SMART CONTRACTS - Technologies, research issues and applications.

- [2] https://raft.github.io/

- [3] https://www.coursera.org/learn/blockchain-professionals

- [4]https://www.lopp.net/pdf/princeton_bitcoin_book.pdf

- [5]https://www.slideshare.net/Ferdinando1970/20161110-rome-icc-intro-to-blockchain

- [6]https://www.slideshare.net/JrmeKehrli/the-blockchain-the-technology-behind-bitcoin

Source of Images used in the Presentation:

- https://www.google.com/search?biw=1366&bih=576&tbm=isch&sa=1&ei=COVcXbG0CuSC1fAPq7im4AU&q=Rocket+Chat+client&oq=Rocket+Chat+client&gs_l=img.3..0i24l8.19625.25398..25625...0.0..0.129.1718.16j3......0....1..gws-wiz-img.....0..0j0i67j0i10j0i8i30.lKyACvBfcwg&ved=0ahUKEwix2P-tq5PkAhVkQRUIHSucCVwQ4dUDCAY&uact=5#imgrc=7lyvv8i1_Vt1DM:

- https://www.google.com/search?biw=1366&bih=576&tbm=isch&sa=1&ei=4-FcXdevHYKi1fAPz_iswAE&q=Rocket+chat&oq=Rocket+chat&gs_l=img.3..0l6j0i10j0l3.179464.182586..182760...1.0..0.255.1749.3j5j4......0....1..gws-wiz-img.....0..0i67.g3lhRPsdaKk&ved=0ahUKEwiXraWuqJPkAhUCURUIHU88CxgQ4dUDCAY&uact=5#imgrc=5Y38hKzVzo56UM:

- https://www.google.com/search?biw=1366&bih=576&tbm=isch&sa=1&ei=2uFcXdXOG6aH1fAP6bSeuAM&q=Docker+container&oq=Docker+container&gs_l=img.3..0l10.5509.7590..7806...0.0..0.211.1358.1j7j1......0....1..gws-wiz-img.......0i67.1PNsdTSZqUc&ved=0ahUKEwjVo_6pqJPkAhWmQxUIHWmaBzcQ4dUDCAY&uact=5#imgrc=7ohK8XSn6r07YM:

- https://www.google.com/search?q=Docker&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjb_8nNp5PkAhVWwMQBHeQAAUkQ_AUIESgB&biw=1366&bih=625#imgrc=Garl5qNG7SGeKM:

- https://www.google.com/search?biw=1366&bih=576&tbm=isch&sa=1&ei=helcXeGgCMeaa4u0uJgK&q=web3.js&oq=web3.js&gs_l=img.3..0j0i24l9.160024.164449..164631...0.0..0.141.513.7j1......0....1..gws-wiz-img.....0..0i67j0i10j0i30.QUrhNbaFacA&ved=0ahUKEwih9-7Rr5PkAhVHzRoKHQsaDqMQ4dUDCAY&uact=5#imgrc=0ouzMNz5nAQgRM:

- https://www.google.com/search?biw=1366&bih=576&tbm=isch&sa=1&ei=belcXejqGqS_lwS2oKKoBA&q=Webpage+icon&oq=Webpage+icon&gs_l=img.3..0l10.20391.22407..22711...0.0..0.67.497.8......0....1..gws-wiz-img.......0i67j0i7i30j0i7i10i30j0i7i5i30.UP1F755shAw&ved=0ahUKEwio1cjGr5PkAhWk34UKHTaQCEUQ4dUDCAY&uact=5#imgrc=rmOz5K7GBVO3EM: