Using Blockchain Platforms to Streamline Naval Shipment Tracking

Abstract

Blockchain not only serves as the foundation of cryptocurrency but has many other applications. The research team at the Naval Postgraduate School in Monterey, CA, investigated how Blockchain could be effectively extended to streamline the Navy's supply chain by tracking serial numbers and logs through smart contracts. The team also explored the usability and efficiency of the IBM Blockchain platform and the Oracle Blockchain platform to benchmark and evaluate their comparative capabilities to maintain an efficient and accurate ledger of all shipment transactions during transportation. This can help reduce the operational cost of the Navy by eliminating the need for clearinghouses within the Navy's supply chain. One of the limitations of Blockchain is that an accurate ledger may only be maintained with the consensus algorithm, if the peer nodes are connected to the Cloud Fabric Environment at all times. However, Navy protocols may require that checkpoints be disconnected unexpectedly. The team investigated creating a serialization function within the smart contracts to allow the ledger to be kept up to date, whenever a peer node created a transaction, i.e. whenever a received shipment was checked in. Though this was not an optimal solution due to the excessive memory usage, it could still be a potential workaround to maintain an accurate ledger.

Overview of Blockchain

Blockchain technology had been initially conceived as an alternative to banks and financial institutions for transaction management. However, its application may be extended to other areas that include serial number tracking and maintaining log integrity, which are essential components of the navy's supply chain.

Blockchain is based on a distributed ledger technology that is able to submit and evaluate transactions using a consensus model on a decentralized peer-to-peer network of shared and cohesive digital data, spanning from multiple sources.

A key element of Blockchain is that users may create their own identity, without third party validation and are allowed to join the network, if they have a certificate authority (CA) or public key. User information is recorded during transactions and distributed on a ledger, without being edit-

ed, unless a smart contract is accessible. A consensus on the recorded information is necessary to generate trust. Blockchain also conceals user information during transactions and eliminates the need for a centralized and secure clearing house that is traditionally provided by centralized authorities. It is able to integrate both individual and communal security within its network using a validation model, called the Democratic Autonomous Organization (DAO). In this model, the "organizational rules are implemented and executed via smart contracts …(and is) composed of participants of validated individuals and organizations who consist of the peer nodes." The decentralized and unhackable security characteristics offered by Blockchain makes it ideal for tracking and streamlining the naval supply chain network, where the peers comprise the Naval Supply Facilities, transportation vehicles, and bases.

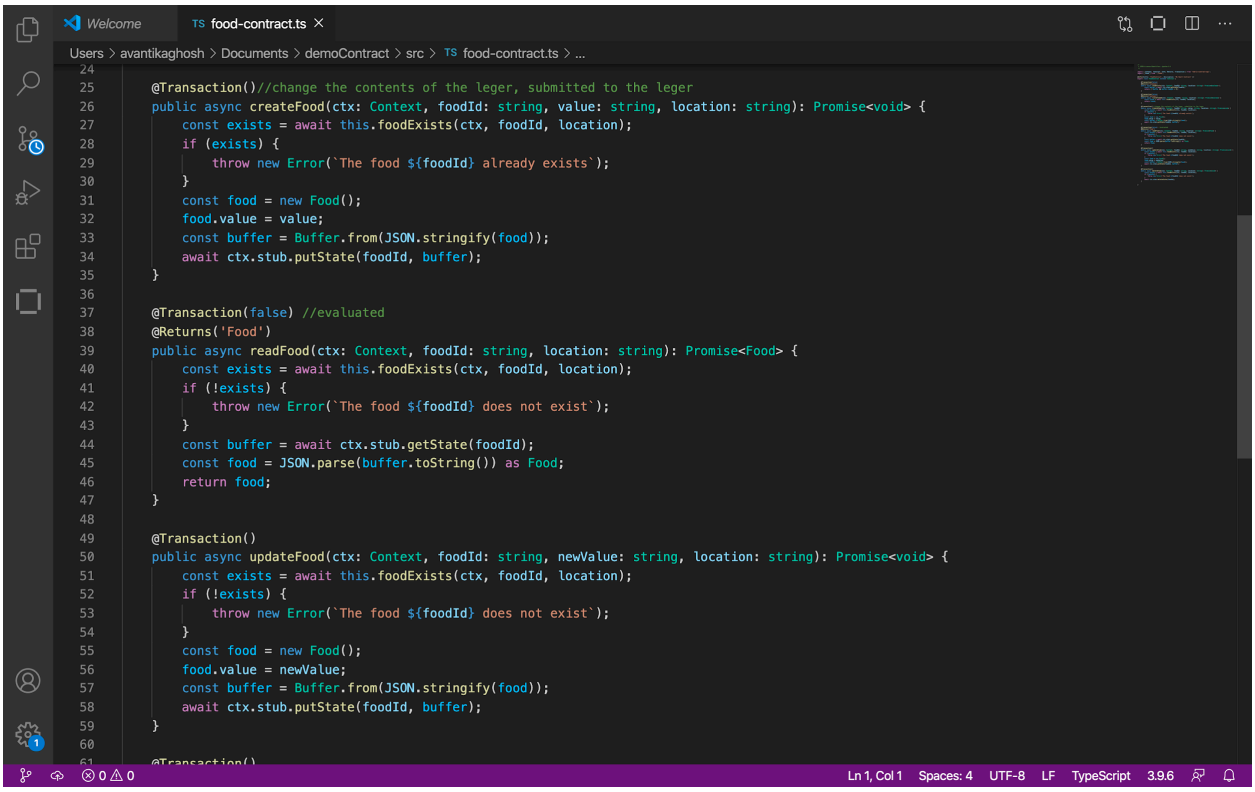
Objective

To investigate how blockchain may be used to track serial numbers and maintain log integrity to ensure the security of the navy's supply chain using smart contracts. To compare the IBM and the Oracle Blockchain Platform and evaluate their ability to maintain an efficient, streamlined, and accurate ledger of all shipment transactions during transportation. Additionally, the team developed a ledger serialization function in the smart contracts for synchronized connection on ships and bases to the Fabric Environment. Using the appropriate Blockchain platform could help to reduce the operational cost of the Navy by eliminating the need for clearinghouses within the Navy's supply chain.

Installation

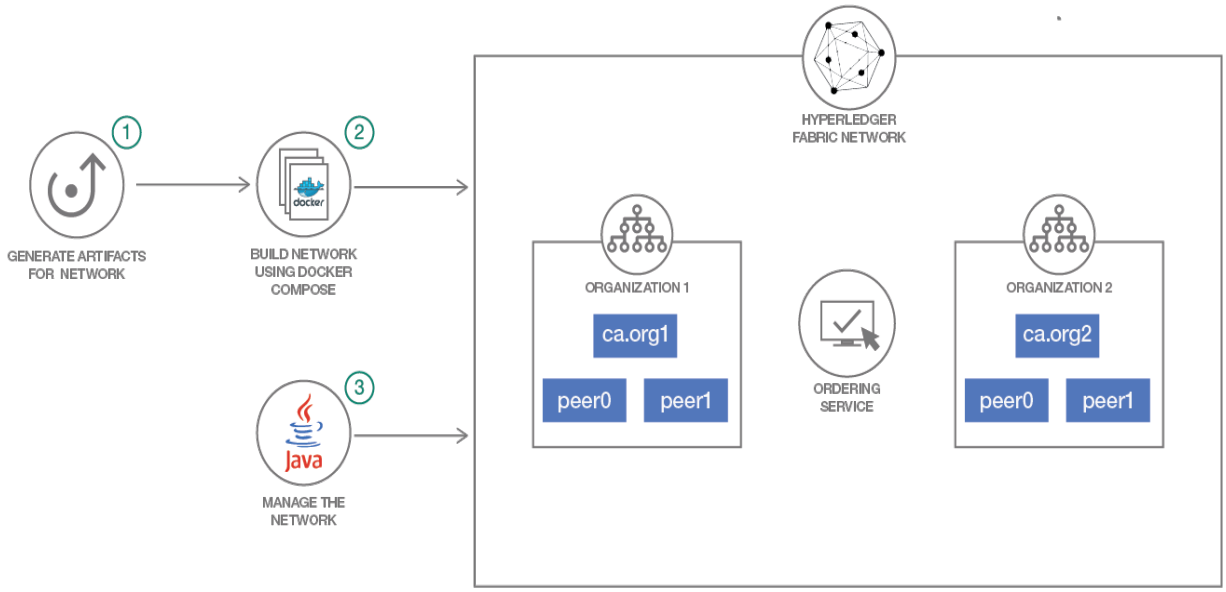
IBM Blockchain Platform: In the IBM Blockchain Platform, users are required to install four vital components: (i) the Virtual Studio Code environment, (ii) Node.js, (iii) Docker, and (iv) Kubernetes. The Virtual Studio Code environment is the offline integrated development environment (IDE), where developers create smart contracts using the open-source programming language, Typescript (developed by Microsoft).

Smart contracts serve as the fundamental basis of the IBM Blockchain Platform because they give certified users the ability to create new transactions and assets, as well as, other functions specific to a project. In this project, the team's main goal was to create a consensus network that has the power to create food shipment assets, update or delete them from the ledger when required, and track their location using their foodId string (which may be replaced by a radio-frequency identification system (RFID)).

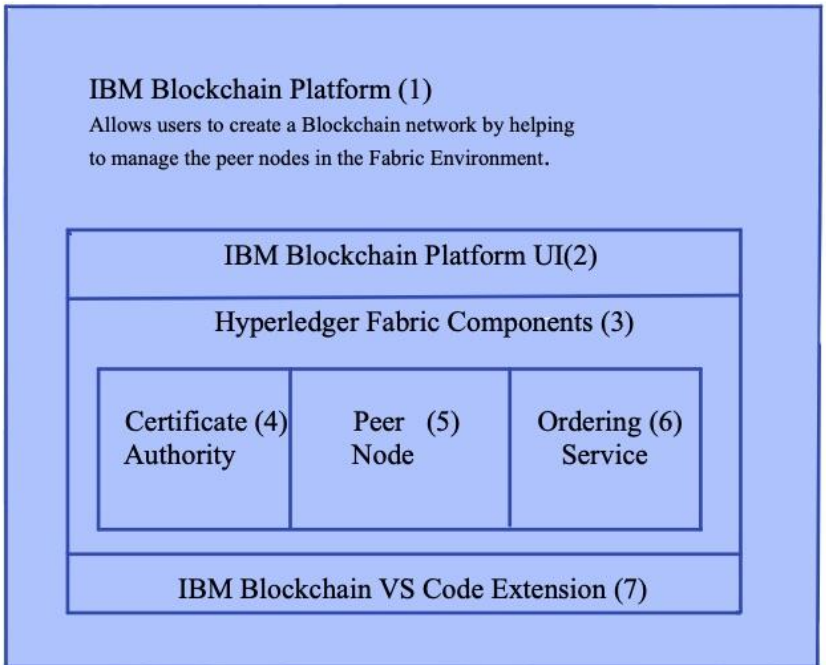


Caption: Sample smart contract for tracking food shipments (Language: Typescript)

Developers must install Node.js and Docker because once logged out of the Fabric Environment, both the ledger and the smart contract are not saved, unless the developer exports both items into a .json file and re-uploads both the files on to the peer nodes as a .CSV file. Docker serves as an OS-level platform to package containers and bundle software, libraries, and configuration files. Using well-defined channels within the software, these containers communicate with each other, to allow the user to connect to the Fabric Environment and add or change the ledger. Finally, the main system that allows the IBM Blockchain Platform to package, install, deploy, and manage the multiple peer nodes in the Blockchain Platform is the Kubernetes system, which was designed by Google and maintained by the Cloud Native Computing Foundation.



Caption: Visual representation of the interaction between the external software and the Hyperledger Fabric Environment

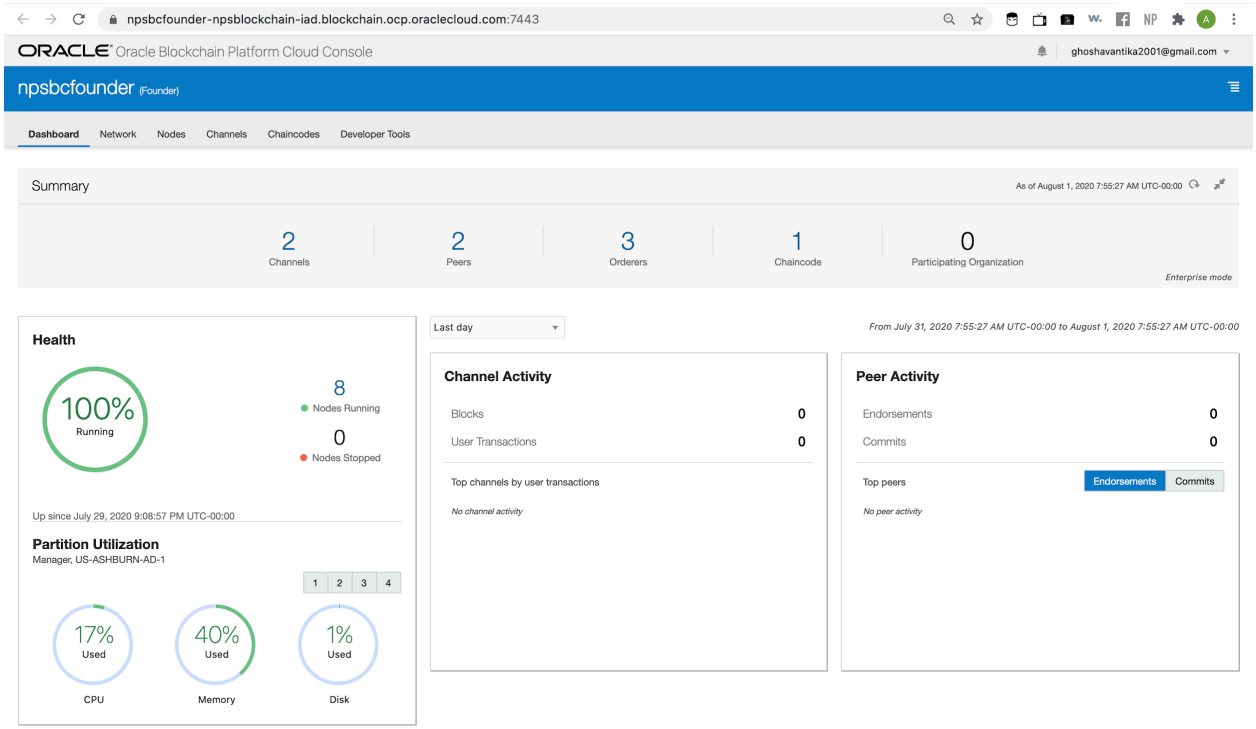


Caption: High-level Representation of IBM Blockchain Architecture

Oracle Blockchain Platform: *[When working with the Oracle Blockchain Platform, the team was given access to the Oracle Cloud Platform thanks to the liaison relationship with the Oracle Blockchain team, provided by NPS.]*

To access the Oracle Blockchain Platform, users must login with authenticated credentials. Once logged into the Platform, users can create a channel and participant nodes, along with Orderers that are responsible for maintaining the order of the ledger. You are not required to download any external software to work with the platform, other than an Integrated Development Environment (i.e. Visual Studio Code) and REST Software Development Kit.

The Oracle Blockchain Platform comes with a Representational State Transfer (REST) API so that developers can invoke a transaction, invoke a query, and view the status of a transaction in the ledger.



Oracle Blockchain Platform Cloud Console Network Summary

Software requirements for IBM and Oracle Blockchain Platforms

Fundamental differences exist in the terminology used to refer to the components within the IBM and Oracle Blockchain Platform software. IBM implements the common Blockchain protocol programs and smart contracts, while the Oracle Blockchain Platform requires developers to write Chaincode in either Java, Go, or Node. When you use the IBM Blockchain Platform, you must install an extension for creating Chaincode.

The following table illustrates the software requirements to use both blockchain platforms:

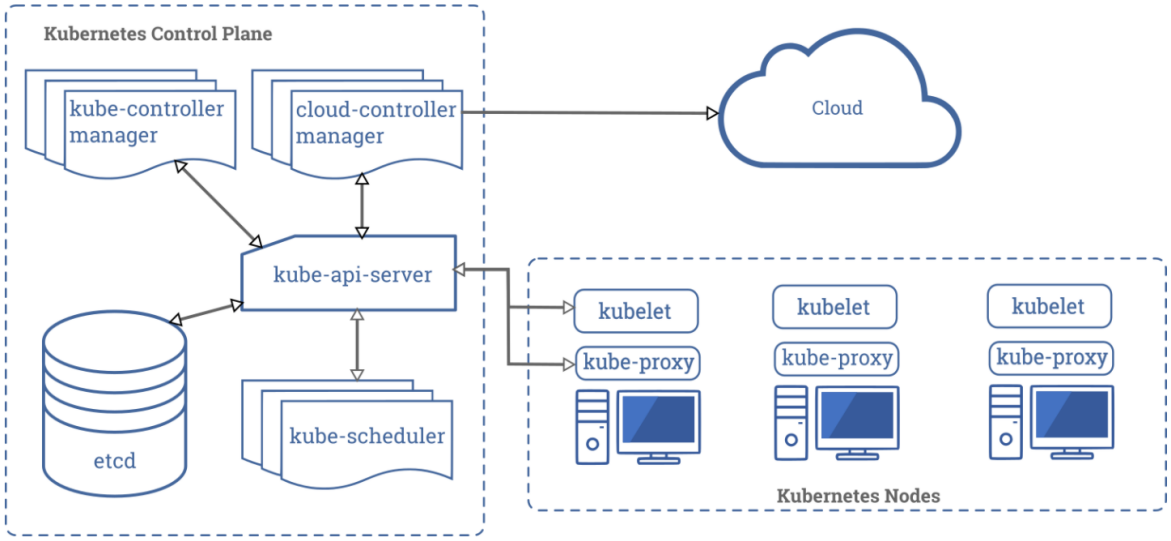
IBM Blockchain Platform	Oracle Blockchain Platform
● VS Code (version 1.32 or higher) ● Node (version 8.x or higher and npm version 5.x or higher) ● Docker (version 17.06.2-ce or higher) ● Docker Compose (version 1.14.0 or higher)	● Integrated Development Environment (Visual Studio Code, Sublime) ● REST API SDK

Caption: Software requirements to install IBM and Oracle Blockchain Platforms

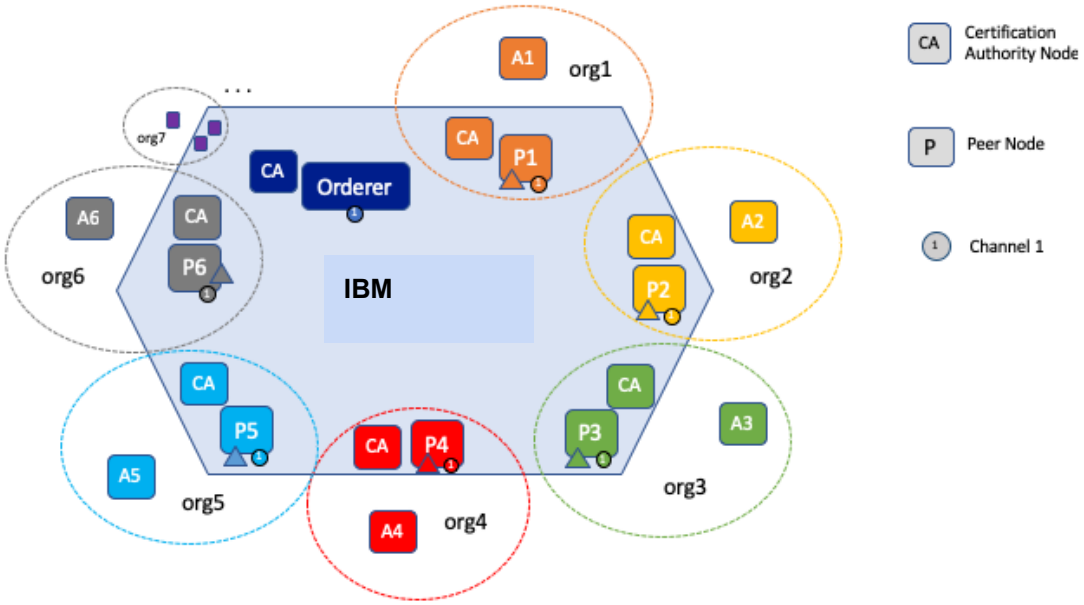
Build a Blockchain Network

Using IBM Blockchain Platform: The Hyperledger Fabric (from the Linux Foundation) is the basis of the IBM Blockchain Platform, in which the individual components are created in a Kubernetes cluster within the IBM Cloud. Hyperledger Fabric is an open-sourced permissioned, distributed ledger that works on the consensus model that is an integral component of the "trust system" in Blockchain. Essentially, the Fabric Environment provides the "common logging" and service management on the platform and the containerized infrastructure allows you to build a blockchain network.

A Kubernetes cluster contains a set of working machines (nodes) that run containerized applications. The modes within the cluster host the components of the application workload. Within the cluster, the control plane manages the nodes and workloads that run across multiple machines, as shown in the following figure:



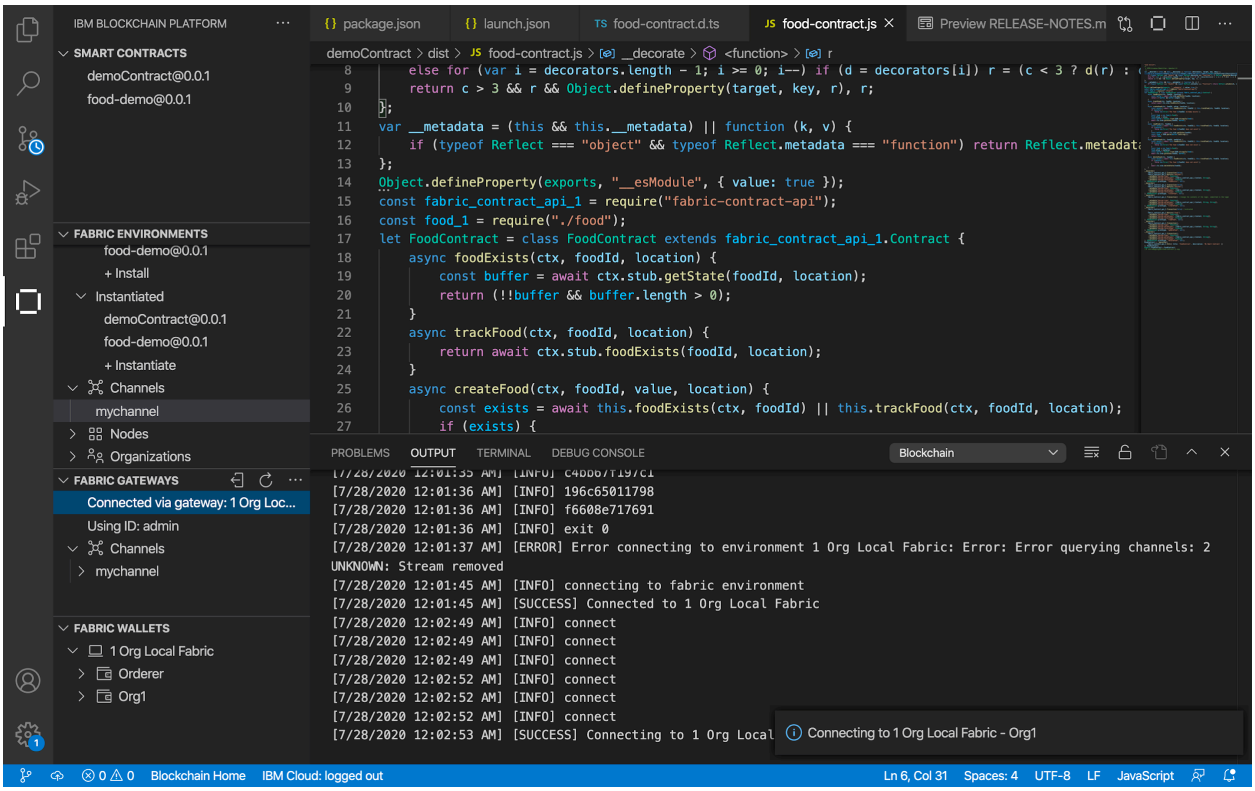
Caption: Visual Representation of the Interaction Between Kubernetes and Cloud



Caption: Visual Representation of the Integration of Security and Nodes in a Blockchain Channel

When the Fabric Environment is running, you can create the Ordering Service. The Ordering Service is a group of "orderers", which accept approved transactions that have been verified by the consensus algorithm and organize the transactions in the appropriate order to be committed to the ledger.

In this research study, the five orderers processed the key-value pairs for each asset based on the smart contract and committed the information to the ledger in order. For example, if a shipment of carrots is approved as the first record in the ledger, the smart contract creates the asset in the form of "001: medical equipment", the orderers process this transaction, and commit the change. The peer nodes host ledgers and smart contracts because they are the backbone of the blockchain network. The smart contract is the transaction protocol that automatically executes, controls, and documents transactions or events occurring on the network. In this study, it was necessary to consistently export the smart contracts as .JSON files and re-upload them as .CSV files to the nodes every time an update occurred in the ledger because disconnecting from the Fabric Environment caused the ledger to be erased completely.



Caption: IBM Blockchain Platform Extension connecting to the Fabric Environment

Like all Blockchain frameworks, the network's legitimacy is upheld by the consensus algorithm because trust is a necessary component when dealing with transactional assets, whether it be through a centralized power or peer network. Each node in the network goes through the entire blockchain and checks that all previous transactions are valid, so that a new transaction may be initiated into the network. However, alternatively, in a permissionless public blockchain, the consensus algorithm is replaced by the "proof of work" which creates a hash system of all of the transactions.

In a "proof of work system", miners constantly attempt to solve the algorithm so that they may mine new blocks and be the first to extend their blockchain. This was not the case in this platform study. Instead, the IBM Blockchain Platform uses a system closer to the "proof of stake" protocol. Essentially, decisions are authorized by users who have a stake in the system as not everyone can join the network. Users may only have "currency" in the network, if the network allows the nodes to participate in the validation process. Unlike "proof of work", no computational power is required, since there are no puzzles needed to obtain "currency". In a "proof of stake" system, "validators" are discouraged from creating faulty empty blocks because they have motivation to incorporate a maximum number of transactions for gains.

In order to ensure security, the hash must be solved by all the peer nodes in the network, so that the new transactions may be approved for the network. While this alternate approach is viable, it is also very energy consuming because ensuring that the ledger is tamper-free requires each ledger copy in the nodes to be changed and hashes to be solved.

Using Oracle Blockchain Platform: In the Oracle Blockchain Platform, the team set up the network using an Oracle Cloud Account. Four peer nodes were set up over a channel. Unlike the IBM Blockchain Platform, obtaining permission to create a Blockchain Platform on Oracle was a more difficult task due to the associated access permissions. Separate email accounts were required to create the platform and access any skeleton chaincode. Additionally, the Cloud platform was used instead of the software package due to the amount of storage required to host the packages on a local computer. However, the fundamental concepts of using a Hyperledger Fabric Environment and consensus algorithm remained the same for both platforms to build a blockchain network.

Current Naval Supply Chain Tracking System

The First Destination Transportation (FDT) refers to the movement and cost of moving shipments from Free on Board (FOB) points of origin to the location at which the shipment is first received for the purpose of use or storage. As naval regulations apply, the first checkpoint of where a shipment is received, whether within the United States (CONUS) or outside (OCONUS), begins with a supplier outside of the Defense Department supply system or an industrial activity that creates the shipment.

The labor and "transportation charges, including freight drayage, cartage, port handling, and other in-transit costs" are processed at the FDT. Freight cartage refers to any inland transit of cargo between locations, which serve as the "checkpoints" in the blockchain network. When a location is assigned responsibility for "cartage of consignments" to land-based activities, ships, or other transportation units, the charges of transportation are given to the location of assigned responsibility, which acts as a peer node checkpoint in the network.

At this point, the initial entry in the ledger may be created and committed by the peer node belonging to the FDT and the Orderers. It is important to note that FDT does not only include shipments of equipment, but also the initial transportation of Navy owned materials that is provided to a contractor for research. This indicates that the charges of a shipment from a contractor's facility to its final destination point is also taken care of by the Government. However, to maintain the legitimacy of a decentralized ledger in this research study, the network for which the ledger is maintained consists of only contractors, supply facilities, and the final base destinations. Essentially, tracking responsibility is passed down from supplier to checkpoint. The checkpoint managers responsible for the charges in a shipment delivery may create and commit the transaction over the blockchain network and the next checkpoint manager may agree or disagree about the condition and extraneous details of the shipment, that the previous manager signs off on.

Currently, the Department of Navy uses the Service-wide Transport (SWT) as a clearinghouse, which is a centralized operations and maintenance manager created to provide transportation funds for naval shipments and mail. Since Naval cargo and the movement of mail to bases is not a responsibility of a destination location, the SWT was created to pay for the movement of material, such as aircraft engines, mission module packages, catapult and arresting gear, propellers, shafts, civil engineering support equipment, safety equipment, drones, overseas mail, and Navy Exchange Service Command (NEXCOM) merchandise shipped from within the United States to international locations.

The Navy struggles with using a decentralized system for these types of "transactions" between checkpoints because there are moments of time where checkpoints are required to disconnect from the Cloud unexpectedly. A downfall of Blockchain is that in order to maintain an accurate ledger with the consensus algorithm, the peer nodes must be connected to the Fabric Environment at all times, unless the peer node decides to save the ledger as a .JSON file and re-upload the ledger as a .CSV file once back online. However, during the time that a node is offline, multiple transactions may occur from other nodes, for which the consensus algorithm will fail. Therefore, configuring a solution where the ledger is automatically updated after predicting potential disconnections from the Fabric Environment will make Blockchain technology a more viable option for all Naval transportation activities.

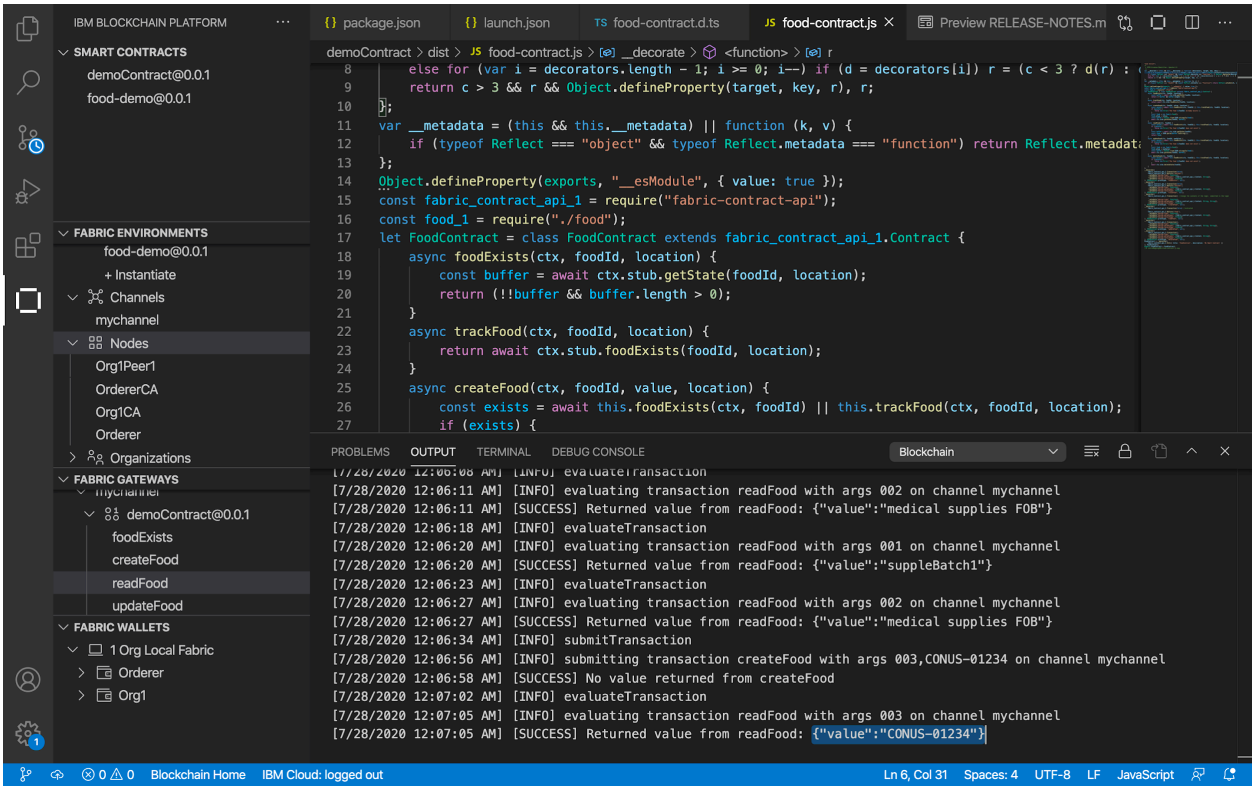
Caption: USS Abraham Lincoln (CVN-72) conducts a replenishment-at-sea (RAS) with the fast combat support ship USNS Arctic (T-AOE-8)

<u>Using Blockchain for Naval Supply Chain Tracking</u>

The Navy desires a multi-functional and secure platform that enables naval personnel working at Naval Supply Facilities to track multiple shipments from production facilities to bases. The Navy is interested in maintaining a secure ledger of inventory that can only be modified with either a unanimous consensus or access to the smart contract. Once a "peer node", or user in the network,

has access to their smart contract, they can modify the transaction protocol that occurs on transactions in the network.

In this Blockchain network, the smart contract contains six methods that carry out the protocol for each transaction on the ledger: foodAssetExists, createFoodAsset, readFoodAsset, updateFoodAsset, trackFoodAsset, and deleteFoodAsset. The method names indicate that each shipment is checked to verify if it already exists at a location denoted by a string. After checking for duplication, the asset is created in the ledger using a key-value pair, such as "001: a shipment of supplies". Once the asset is created, it is always a good practice to read the asset's details into the ledger, so that users further down the network may have a detailed understanding of what a package is supposed to contain. Also, if a shipment is changed, say, a package is redirected to a base that is in need of supplies urgently, the shipment's location is updated within the ledger and deleted once the shipment arrives.



Caption: Sample ledger of shipments that are added and updated (contents/location)

Ledger Serialization

A significant concern when implementing blockchain technology in cargo shipments is its dependence on continuous connection to the Fabric Environment. The Blockchain platform requires you to be connected to the Fabric Environment at all times or consistently reupload the

ledger to the peer nodes. This requirement is non-negotiable for the Navy. Hence, the team investigated creating a serialization function within the smart contracts that would allow the ledger to be kept up to date, whenever a peer node created a transaction, i.e. whenever a received shipment was checked in.

However, creating a serialization function within the smart contract protocols not only requires an excessive amount of memory, but also defies the purpose of a decentralized system because the network would be reliant on different individual nodes to maintain an accurate ledger during points of disconnect from the Cloud.

An alternative to ledger serialization would be to use a paper wallet, which is a method of storing a private key to access funds stored on a single address. Creating a paper wallet does not require you to connect to the Internet, and it is printed out for air-gapped offline storage. However, unlike a Blockchain node with a Certificate Authority, there is no way to recover assets if the private key is lost, since the key must be imported into an online wallet. Furthermore, even though connecting to the Internet is not required to generate a wallet, it is impossible to recreate the same wallet once the machine is turned off. Therefore, if a ship turns off all power at once, the paper wallet is no longer a viable option for ledger maintenance.

Conclusion

Both IBM and Oracle Blockchain Platform may be used to create a secure network of peer nodes or naval hotspots that can generate a consensus for the legitimacy of the shipment ledger, which can only be modified using smart contracts. Since a key component of both platforms is maintaining accuracy and security of the ledger, all users must consistently export and import the smart contracts and ledgers onto their respective peer nodes every time an update is made on the ledger or the transaction protocol on the smart contract is changed. In order to streamline this process, a serialization method can be created on the smart contracts that will automatically save the ledger onto the peer nodes. However, this technique will require a significant amount of memory in the system because the serialization method constantly saves the entire ledger whenever there is an update, instead of simply adding to the ledger in increments.

The team also compared the IBM and Oracle Blockchain Platforms on efficiency and maintainability of a ledger of shipments and discovered that it was easier to use the IBM platform to create and export smart contracts and ledgers. The IBM platform required the user to develop their smart contract on the Visual Studio Code environment, export the contract as a .JSON file, login to the online Blockchain network, and import the contract and ledger as a .CSV file using a con-

verter. The Oracle Blockchain platform, on the other hand, allowed users greater flexibility to join ledgers more cohesively. The Oracle platform allowed users to login to the Oracle Cloud after they were approved by an appropriate administrator, and used simple software like IDE and Software Development Kit. Furthermore, the Oracle Blockchain Platform employed chaincode as a smart contract for transactional protocols in the network. A chaincode is written in either Java, Node.js, or Go and packaged into a ZIP file, which can be installed on the network. This is similar to how smart contracts are exported as .JSON files and uploaded on the IBM network as .CSV files. More specifically, chaincodes outline the structure of the ledger, initialize it, create updates (such as reading or updating entries), and respond to queries (which is not a highlighted attribute of the IBM Platform). We also discovered that chaincodes allow associated applications to be notified of actions that occur after a period of time in the ledger. Therefore, chaincodes are capable of notifying subscribed applications about payments after purchase orders and delivery records have been matched by a chaincode. Once the subscribed application has received a notification based on the chaincode, the internal Enterprise Resource Planning System updates accordingly.

Area of Comparison	**IBM Blockchain Platform**	**Oracle Blockchain Platform**
External software required (computer memory)	Requires installing **four** external software packages	Requires installing **two** external software packages (Cloud account platform)
Ease of development for smart contracts	Easier for developers to create smart contracts on the associated extension	Requires extensive authorization from an Oracle employee to set up an account for developing chaincode
Type and Level of Security	Requires users to create a Certificate Authority (CA) in order to join the network	Uses the Oracle Identity Cloud Service where user tenancy is federated to another Identity Provider and HTTPS/SSH secure protocols to access service instances

Synchronicity	Requires connecting to the Fabric Environment for the entirety of the ledger's application to the use case. The ledger does not update accurately if there are disruptions to the connectivity, unless smart contracts are strategically reuploaded to the peers. See *Ledger Serialization* section	Requires reuploading chaincode to the network, but requires extensive permissions for uploading while the IBM Platform allows users to upload freely
Tools Accessible	Use cases can be achieved by modifying the smart contract code, but tutorials on multiple use cases are not easily available	Developer manual provides multiple use case guidelines for creating chaincode, and provides clear options for three different programming languages (Go, Java, Node.js)

Caption: Comparison of IBM and Oracle Blockchain Platform

Future Areas of Investigation

Disconnecting from the Fabric Environment

As mentioned in multiple occasions above, it is difficult to predict when a ship that is part of the Blockchain network will go offline and disconnect from the Fabric Environment, which is necessary to maintain both the consensus and an accurate ledger. Therefore, understanding how Blockchain can circumvent the issue of constant connectivity, while still maintaining its status as a decentralized system will make the technology more feasible for the Department of Navy.

Maintaining an Accurate Ledger

Beyond Naval convenience, another important area of further investigation is to identify a solution for how to mainstream the ledger in Blockchain so that the ledger maintains a clean, accurate ledger as more peer nodes are added to the network. With a network of ten nodes, it is relatively simple to maintain the ledger in an organized manner. However, when taking into account the complexity of naval shipment transportation, the amount of peer nodes needed to complete the network requires implementing a joins function to the ledger so that multiple validated ledgers are kept, but are joined to create the approved ledger legitimized by the consensus algorithm.

Acknowledgements

The authors would like to thank the Office of Naval Research summer internship program at the Naval Postgraduate School which enabled them to collaborate on this effort. The authors would also like to thank Keyauri Kendrick, Tom Plunkett, Peter Walsh, Mark Rakhmilevich, and Bala Vellanki for their help in navigating both Blockchain platforms.

Arijit Das has been part of the research faculty at the Naval Postgraduate school since 2002, teaching computer science courses such as Java, databases, C, Android programming and HTML5, and working on DoD projects in the areas of databases, big data, mobile development, e-Learning and virtualization. He is a frequent speaker at developer tech events for Oracle technologies applicable to DoD information technology.

Avantika Ghosh is a sophomore at UC Berkeley studying Computer Science and Economics. She has been a research intern at the Naval Postgraduate School (NPS), Monterey, CA in 2018 and 2020. In 2018, she conducted a Data Analytics study on benchmarking various machine learning algorithms' to detect anomalous behavior in military aircraft data. Through her current research, she is exploring possible applications of Blockchain technology in the Navy. Avantika has been

awarded the Bay Area Affiliate Winner award by the National Center for Women in Technology. She has competed in multiple hackathons at her university, where she won First Place in the "Hack for Social Impact Summit" and created a streamlined app design for the Berkeley Food Collective to optimize the user experience when searching for affordable food. Avantika is currently working on developing a predictive trading algorithm that she plans to apply to her own trading portfolio.

Aroshi Ghosh is a junior at Leland High School and the city-wide Youth Commissioner for the City of San Jose. She also works as a research intern at the Naval Postgraduate School under the guidance of Prof. Arijit Das and is passionate about identifying technological solutions to solve real-world issues. Aroshi strongly believes in being an advocate for the community and is the founder of the AI Spectra outreach program that helps young women explore careers and applications in Artificial Intelligence at the cross-section of the arts and humanities. She was awarded the Bay Area Affiliate Winner by the National Center for Women in Technology (NCWIT) and received a $3000 grant for her STEM outreach activities in 2020. She is also an avid blogger and writes for her online magazine “Student Spectator” about tech, politics, arts, race, and their impact on Gen Z. Additionally, Aroshi supports the world ranked Leland Quixilver 8404 Robotics team as Vice-Captain.

Tony Kendall is a faculty member in the information sciences department at the Naval Postgraduate School. His teaching interests include Information Technology for Homeland Security Professionals, Decision Support Systems and Knowledge Management, and IT for Security for developing countries. In his earlier career, he was a Naval lieutenant commander and received his Master of Science at NPS. He is currently working on a project alongside Professor Arijit Das on using “moneyball” and other methods to provide analytics for Naval aviation maintenance. His previous projects include implementing Oracle’s WebCenter at the Naval Postgraduate School.

Resources

Eckstein, Megan. “Study Says Navy Logistics Fleet Would Fall Short in High-End Fight.” *USNI News*, US Naval Institute, 16 May 2019, news.usni.org/2019/05/17/study-says-navy-logistics-fleet-would-fall-short-in-high-end-fight.

Seang, Sothearath, and Dominique Torre. “Proof of Work and Proof of Stake Consensus Protocols: a Blockchain Application for Local Complementary Currencies.” *SciencesConf*, Feb. 2018, gdre-scpo-aix.sciencesconf.org/195470/document.

Chief of Naval Operations. *OPNAV INSTRUCTION 4600.24E*. Department of Defense, 13 Nov. 2018, www.secnav.navy.mil/doni/Directives/04000%20Logistical%20Support%20and%20Services/04-600%20Travel%20Transportation%20Service%20and%20Support/4600.24E.pdf.

"DAO - Decentralized Autonomous Organization." *Horizen Academy*, Horizen Academy, 2019, academy.horizen.global/horizen/advanced/dao-decentralized-autonomous-organization/.

Mackey, Tim K., et al. "A Framework Proposal for Blockchain-Based Scientific Publishing Using Shared Governance." *Frontiers*, Frontiers, 30 Oct. 2019, www.frontiersin.org/articles/10.3389/fbloc.2019.00019/full.

"Empowering App Development for Developers." *Docker*, www.docker.com/.

Porutiu, Horea, and Ed Moffatt. "Develop a Smart Contract with the IBM Blockchain Platform VS Code Extension." *IBM Developer*, IBM, 16 May 2019, developer.ibm.com/tutorials/ibm-blockchain-platform-vscode-smart-contract/.

"Purchasing, Logistics & Supply Careers." *Purchasing, Logistics & Supply Careers | Navy.com*, America's Navy, www.navy.com/careers/purchasing-supply-logistics.

Kenyon, Henry S. "Navy Raises Anchor on Blockchain." *SIGNAL Magazine*, AFCEA International, 1 Mar. 2019, www.afcea.org/content/navy-raises-anchor-blockchain.

"Kubernetes Components." *Kubernetes*, Kubernetes, 22 June 2020, kubernetes.io/docs/concepts/overview/components/.

Ghosh, Avantika. "Ghosh-Avantika/NPS_Blockchain." *GitHub*, GitHub, github.com/ghosh-avantika/NPS_Blockchain. Code for ledger serialization attempt for the IBM Blockchain Platform study.

Dash Core Group, Inc. "Introduction¶." *Introduction - Dash Latest Documentation*, Dash, docs.dash.org/en/stable/wallets/paper.html.

Porat, Amitai, et al. "Blockchain Consensus: An Analysis of Proof-of-Work and Its Applications." *Stanford Secure Computer Systems Group*, Stanford University, www.scs.stanford.edu/17au-cs244b/labs/projects/porat_pratap_shah_adkar.pdf.

"Developing DApps on Oracle Blockchain Platform A Developer's Guide." *Developing DApps on Oracle Blockchain Platform - A Developer's Guide*, Oracle, 2018, www.oracle.com/webfolder/s/assets/ebook/developing-dapps-oracle-blockchain/index.html#/page/27.