*Blockchain for Power Grids*

Christian Banks  
USMA  
Class of 2019, Co. B3West Point, US  
Christian.Banks@westpoint.edu

Samuel Kim  
USMA  
Class of 2019, Co. F1West Point, US  
Samuel.Kim@westpoint.edu

Michael Neposchlan  
USMA  
Class of 2019, Co. C1West Point, US  
Michael.Neposchlan@westpoint.edu  
  
Nicholas Velez  
USMAClass of 2019, Co. D4  
West Point, US

Nicholas.Velez@westpoint.edu

***Abstract*— Sharing information is an important part of regulating and maintaining efficient and safe power grids. This project’s goal is to develop a way of using blockchain technology to share transaction information among different power grids in a secure, controlled, monitored, and efficient manner. The biggest concern regarding the data is integrity. With the implementation of proper non-repudiation measures, the data will be reliable and attacks, such as man-in-the-middle and data spoofing attacks, can be prevented. The Hyperledger Fabric that we implemented hosts a permissioned network in which power grids will act as nodes that maintain ledger information. By using a distributed ledger as a way to validate transactions though the process of consensus, the system is able to share information in a manner that is more secure and transparent than traditional information sharing systems in which data is less secure and takes longer to validate. The additional layers of security and speed that Hyperledger technology provide help to prevent issues, such as power grid failures, that could stem from the latency involved with traditional methods of validating, processing, and reacting to shared data.**

***Keywords—* Blockchain, Hyperledger fabric**

# Introduction

Sharing information across large networks poses many issues and potential risks involving security and usability. For the purpose of sharing information between microgrids, security and confidentiality are paramount. Blockchain architectures provide a variety of solutions to sharing information. Of the many different blockchain solutions currently available, Hyperledger provides superior scalability, modularity, and security [1]. For this project, in order to facilitate the sharing of information among various microgrids in a way that will potentially be scalable to much larger industry grids, a hyperledger fabric network is being implemented.

# Background

* 1. *Blockchain*

Blockchain is a group of transactions that are linked to their previous modification on a specific channel.[2] The chain is a log that contains the transactions of all previous ‘blocks’ for that particular chain. When a new block is appended unto the chain, the transaction from that previous block is also appended.[2]

A large feature of a blockchain is the use of a distributed ledger.[2] Blockchain ledgers are often decentralized because each person on the network is working with their own replication of the block.[2] Utilizing a decentralized ledger helps add security to the network as all information is not funneling into one node. The ledger contains two characteristics, a world state and a blockchain.[3]

A blockchain was explained in the previous paragraph, however, a world state is a database that keeps the values of a ledger state at the current time.[3] This allows for the pulling of the current state at any amount of time without having to sweep a log.[3]

* 1. *Hyperledger fabric*

Hyperledger fabric is a type of blockchain that involves a collaborative approach to community sharing, property rights and the development of standards over time.[5] Like other Blockchain implementations, Hyperledger has a ledger and utilizes smart contracts.[5]

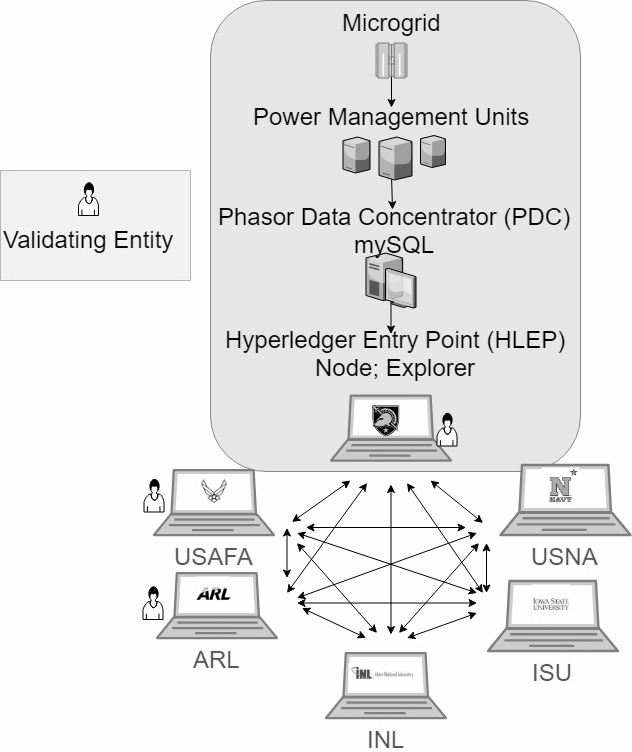
# Design (PLAN)

## Powergrid Network

Our planning and design for our project relied on a previously completed Wide Area Monitoring and Control system (WAMC).[6] In this project, an existing smart grid testbed was developed.[6] In St. leger, Spruce, Banwell and Collins IEEE paper, “*Smart Grid Testbed for Wide-Area Monitoring and Control Systems*,” the specific foundations of the testbed are explained. 10 relays provide phasor measurement information for each phase.[6] All the PMU (Phasor Management Unit) data are consolidated in a Microsoft SQL database that is monitored by OpenPDC.[6] How the data is being pulled from the SQL database will be explained later in the paper.

The purpose of this project is to provide a foundation to solve the problem of the data distribution that would be a major driving force in implementing a safe and efficient WAMC in the real world. In the future, WAMC can help to prevent issues and errors that stem from anomalous activity in a power grid. Traditional Supervisory Control and Data Acquisition (SCADA) systems are limited in their capability to detect anomalies in power grids. SCADA provides asynchronous data. Additionally, there is a significant delay, ranging in length from seconds to minutes. [6] Using blockchain technology for power grids could potentially be a solution to these issues seen in SCADA systems.

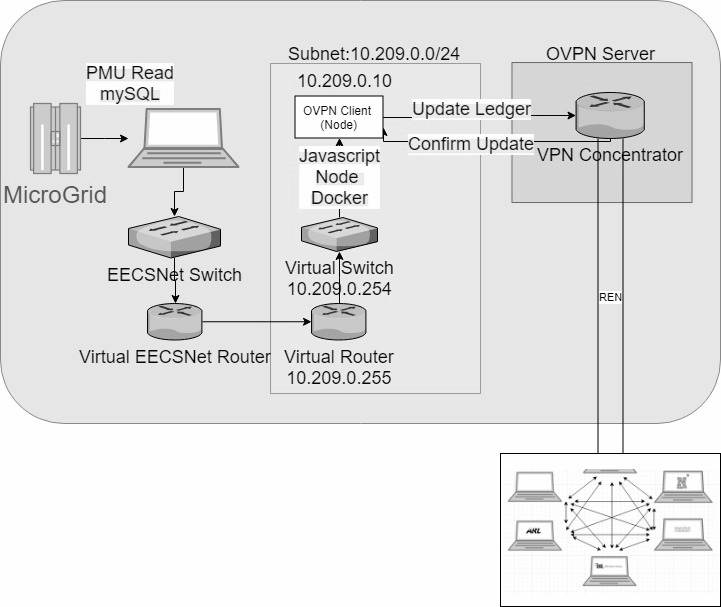
Currently, our team has completed a two node architecture and were able to get the two nodes to communicate. Referencing the diagram, the two nodes that we have successfully completed reside in the USMA domain. Figure 1 is an example of the final network design in which all six peer nodes will reside.



1. Final Network Design

## Internal Network

The process of data retrieval from the node network is explained in our internal network diagram found on Figure 2 below.

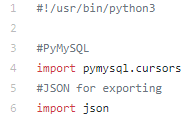


1. Internal Network Design

MySQL data from the USMA MicroGrid is read by one of the hosts located at USMA. The data from that PC is read by the database OPVN client (Node). Data is obtained by running a database query algorithm. This algorithm which is explained in the next paragraph. Then from the OPVN client, data will be sent out the VPN concentrator with a update request to the rest of the nodes on the network, and the nodes will confirm the request to update the ledger. The other nodes are then able to connect to the VPN concentrator in order to send confirmation requests back to our client. Once the request is sent back to our client node validating the update, the ledger state is updated with the appropriate information. Each node on the network will run a similar instance of OPVN client node as seen on Figure 2.

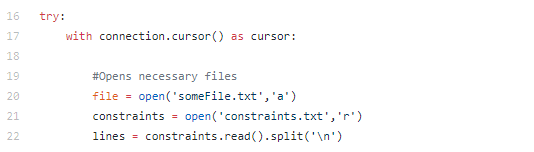
## Database Query Algorithm

The modules that we used were PyMySQL and JSON. The PyMySQL module enables us to interface with the SQL database. [7] The JSON module acts like and encoder and decoder to help us convert our data into a useable JSON format. [8]



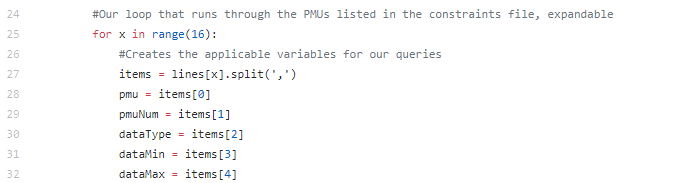
1. Python Code: Modules

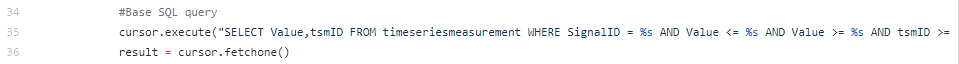
The process of data retrieval from the node network is explained in our internal network diagram found on Figure 2. The python script first establishes a connection to the MicroGrid SQL database using proper authentication. After successfully connecting, the script is read from a constraints file provided to by the Real-Time Smart Grid Capstone team.



1. Python Code: Establishing a connection and Script Reading

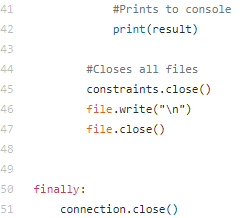
This file contains the IDs of all the PMUs and applicable variables necessary for queries. The program then loops through each PMU and queries it for the latest readings that fall within the established minimums and maximums.





1. Python Code: Looping and SQL Query

The queries are written to a file in readable format. Additionally, each query is printed to the console (for verification the script is working).



1. Python Code: Printing the Result, Closing Files, Ending Connection

Finally closes all files and connections. In order to more easily create historic data for our network, the script is being updated to accommodate JSON formatting.

## Scalability

Once a successful proof of concept can be run, our framework can be adapted on a larger scale to allow peer nodes to communicate over large distances. Latency would be affected by the changes in the environment. However, as the algorithm improves and is optimized, these effects can be mitigated. Figure 7 shows the potential probably with latency because of distance. Figure 3 shows the 6 microgrids that will share information from the blockchain.



1. West Point, NY to Idaho National Laboratory

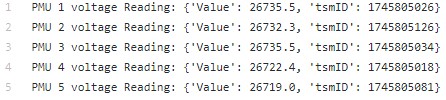


As observed on the figure, latency issues arise due to the long distance between the microgrids in Idaho, Colorado, and Iowa and the microgrids on the east coast. With larger amounts of latency, the potential for the control system to become unstable rises. This is something that we are accounting for in our plan for expanded scalability. Currently, our team has successfully created a database query algorithm with python with a successful update to accommodate JSON formatting. Furthermore, we were successful in pulling data from the USMA micro grid.

# Implementation

## Database interface

The data that is being shared in this project is collected from Phasor Measurement Units (PMUs) connected to the local power grid. These units collect voltage and current measurements from different parts of the microgrid. Through python coding, we are able to successfully pull data from the microgrid at West Point. The data includes: wattages, currents, voltages, and time stamps exiting the microgrid. Figure 7 is an example of a successful pull from the West Point microgrid.



1. Example of Successful Data Pull from WestPoint MicroGrid

Our code is modified to display what PMU we are accessing, the value of the voltage and the time stamp ID (seen as tsmID in Fig. 6). The voltage reading that displays none, means that the PMU has not been recording or has no data for that timestamp. Our next goal is to use javascript code to inject our script into the hyperledger network that we created.

## Peer Node

For the peer node network, we are currently implementing Hyperledger architecture in a virtual environment. Our team adapted a tutorial in order to create a two node hyperledger architecture utilizing virtual machines to successful conduct a hello world test. In order to verify the communication amongst the nodes, a CouchDB server was utilized to record all relevant information - which was encrypted through hashing. The Couchdb server also allows for the database to be viewed through a web interface. On the client node, there is an orderer terminal, a command line interface, and a log file being updated in the background. On the manager node, there is a certificate authority server, an orderer, and a Couchdb server.

# Future work

## End of the Project

The endstate for this project is a network with connectivity across six peer nodes including: the United States Military Academy (USMA), the United States Naval Academy (USNA), the United States Air Force Academy (USAFA), Army Research Labs (ARL), Iowa State University (ISU), and Idaho national labs (INL). In essence, all of these nodes will represent microgrids that can communicate with each other and share data in a secure way on the network using Hyperledger. The validating entities inside of the network are USMA, USAFA and ISU. The validating entities validate transactions and maintain the ledger while ensuring consensus has been achieved on all transactions. Additionally, the latency will be optimized to ensure that data is being transferred and received at the fastest speed possible without bearing a large load on the system.

## Future Work

The implications of this technology apply to a global level of communication amongst power grids that rely on one another to supplement power when needed. A hyperledger architecture supports a secure method of communication in an infinitely large network. This process expedites the sharing of information which results in the prevention of potential blackouts relevant to power grids.

# Conclusion

In this paper, the basics of blockchain and hyperledger fabric are explained to help give understanding to the project: Blockchain for powergrid. First, the design of the powergrid network, and internal network, database query algorithm and scalability is discussed. The implementation of our system and the current results were also discussed. Lastly, recommendations for future work work were given, such as upscaling our project past communication between schools and movement to communication between larger scale power grids. By implementing our design we can create a system that allows microgrids to communicate relevant data across many peer nodes. This can help solve issues that stem from the current information sharing systems.

##### References

1. <https://www.hyperledger.org/projects/fabric>
2. https://hyperledger-fabric.readthedocs.io/en/release-1.2/glossary.html
3. https://hyperledger-fabric.readthedocs.io/en/release-1.2/ledger/ledger.html
4. <https://hyperledger-fabric.readthedocs.io/en/release-1.2/blockchain.html>
5. A. S. Leger, J. Spruce, T. Banwell, and M. Collins, “Smart grid testbed for Wide-Area Monitoring and Control systems,” *2016 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, pp. 1-5, 2016.
6. S. J. Matthews and A. S. Leger, “Leveraging single board computers for anomaly detection in the smart grid,” *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp 1-7, 2017.
7. “User Guide,” *Examples - PyMySQL 0.7.2 documentation*. [Online]. Available: https://pymysql.readthedocs.io/en/latest/user/index.html. [Accessed: 05-Feb-2019].
8. “18.2. json - JSON encoder and decoder,” *16.2. threading - Higher-level threading interface - Python 2.7.15 documentation*. [Online]. Available: https://docs.python.org/2/library/json.html. [Accessed: 05-Feb-2019].