### 

UND NORTH DAKOTA

Project: Incorporating Blockchain/P2P Technology into an SDN-Enabled Cybersecurity System to Safeguard Fossil Fuel Power Generation Systems Project Number: DE-FE0031742

University of North Dakota College of Engineering and Mines

> Annual Technical Review (Spring 2022) May 20<sup>th</sup>, 2022

#### Overview

#### This project aims to strengthen the security protection of software defined networking (SDN) for facilitating its deployment in fossil fuel power generating systems.

- The security protection solution makes use of the blockchain and the peer-to-peer (P2P) technologies.
- □ This project is in response to Area of Interest 2 of DE-FOA-0001991.
- AOI 2: "investigate how cutting-edge network technologies such as blockchain may be leveraged and integrated into industrial monitoring and process control systems for optimized, cybersecure operation of electricity generating units."

#### This project aims to produce two deliverables:

- A cloud-based networking platform for prototyping and experimenting various designs of safeguarding the software-defined networks deployed in electric power systems.
- A blockchain/P2P-based technology for detecting the compromised controllers in a software defined network.
  - The application will operate in the cloud-based networking platform.

#### The outcomes of this project will serve in

- Meeting the general security requirements of the electric power generating systems.
- Mitigating the security risks targeting the vulnerabilities of SDN-enabled operational networks.



### Strategic Alignment of Project to Fossil Energy Objectives (1)

#### Serving for Meeting the General Security Requirements of Electric Power Systems

□ Safe operations of power systems rely on the fundamental security mechanisms

o Authentication, Authorization, and Anti-spoofing.

#### > Serving for Facilitating the Deployment of SDN-Enabled Operational Networks

□ Software Defined Network (SDN) technologies will be increasingly adopted to support data communications in electric power systems.

□ The Department of Energy had sponsored research projects on

- Applying SDN technology to support the device-to-device communications;
- Prototyping a dashboard application for providing the operators with a global view of the SDN-enabled operational networks.

Serving for Addressing the Threats Targeting the SDN-Enabled Operational Networks

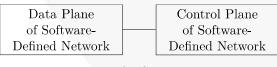
- □ SDN paradigm faces new security threats and attacks.
- Our project addresses the security risks targeting SDN technology and the protection solutions.





#### Strategic Alignment of Project to Fossil Energy Objectives (2)

### System Diagrams of traditional Software-Defined Network

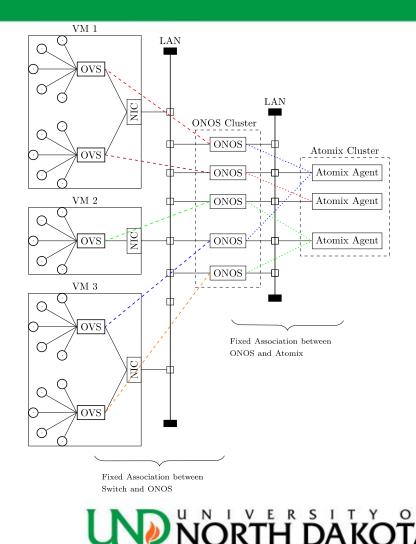


Inquiry for SDN Rules Responce with SDN Rules

Inquiry for Traffic Statistics

#### Key vulnerabilities

- Lack of detection on security breaching.
- Lack of effective mechanism for excluding compromised SDN controllers from a SDN.



### Technology Benchmarking (1)

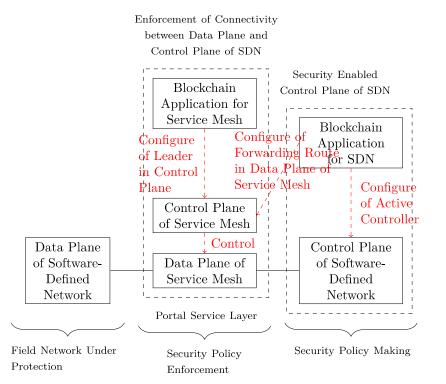
- This project aims to construct
  - A cloud-based networking platform which can be used for
    - Studying the threats targeting SDN-enabled operational networks deployed in electric power systems, and
    - Prototyping security protection solutions thwarting the attacks targeting the control plane, the forwarding plane, and the communications between the control plane and the forwarding plane.
  - A blockchain/P2P-based technology for detecting the compromised controllers in software defined networks.
- Industry/input or validation
  - This project is in collaboration with Minnkota Power Cooperative.
  - *Minnkota Power Cooperative* helps to facilitate decision-makings on the scientific and technical direction of the project and will be a user of the cloud-based networking platform.
  - This project has also attracted attentions by a cybersecurity marketing firm and a network equipment supplier serving for data communications used in power generation and transmission.





#### Technology Benchmarking (2)

- This project aims to enable security protection for SDN.
  - Enabling detection of compromised SDN controllers by constructing blockchain applications.
  - Enabling exclusion of compromised SDN controllers by decoupling the direct and fixed connectivity between the data plane and control plane in SDN.
    - A portal service layer is used to decouple the two planes.





#### Overview of Major Efforts (1)

#### > We have conducted 8 efforts in this project.

Efforts made before 2021 Review Meeting

- Effort #1: Determined the overall structure of the security-enabled SDN system.
- Effort #2: Constructed a private cloud platform over 3 Dell servers.
  - The testbed supports simultaneous run of multiple SDN simulations.
- Effort #3: Constructed the SDN with a controller cluster.
  - Mininet is used for simulating the data plane of the SDN.
  - A cluster of ONOS controllers is used for the control plane of the SDN.
- Effort #4: Constructed the portal service layer to bridge the data plane and the control plane of an SDN.
  - The portal service layer is materialized in the form of a service mesh which consists of
    - ✤ A data plane: a set of *Envoy* proxies.
    - ✤ A control plane: a set of Consul agents.



**Overview of Major Efforts (2)** 

>We have conducted 8 efforts in this project.

□Efforts made since 2021 Review Meeting

- Effort #5: Added the Discovery Service (xDS) in the portal service layer to dynamically forward SDN traffic.
- Effort #6: Constructed the base blockchain system running on top of a peer-to-peer (P2P) data storage
  - Adopted the InterPlanetary File System (IPFS) as the P2P data storage.
  - Adopted Hyperledger Fabric (HLF) software (version 2.4) as the base blockchain system.
  - Added Orbit-DB as the key-value store of (key=readable event name, value=block ID).



**Overview of Major Efforts (3)** 

>We have conducted 8 efforts in this project.

□Efforts made since 2021 Review Meeting

- Effort #7: Programmed an Application Programing Interface (API) server to facilitate the application of malicious attack detection to access the HLF blockchain sub-system.
  - The API server serves the requests sent from the application through performing a sequence of operations on the HLF and IPFS sub-systems.
  - The API server provides the responses with respect to the application's requests.
- Effort #8: Performed the literature study on BFT consensus and the theoretical preparation on constructing an information-theoretic framework of an efficient BFT consensus.



**Current Status of Project** 

- ≻This project started on September 1<sup>st</sup>, 2019 for a 3 years duration.
- This project , and there is no available comparison with known benchmark.
- >There is no major change in the project goals/objectives.
- We have made some changes in the actual implementation of the tasks.
  - We have decided to construct the originally proposed testbed in the form of a cloud-based networking platform.
  - We have decided to adopt the proof-of-reputation consensus model for detecting the compromised network controllers in SDN networks.
  - We have simplified the structure of the system of detecting the compromised network controllers.





#### **Current Status of Project**

### > This project started on September $1^{st}$ , 2019 for a 3 years duration.

Timeline of major tasks and milestones

Task Description	Planned		Achieved	
	Start Date	End Date	Start Date	End Date
Task 1.0 Update project management plan	9/1/19	9/30/19	9/1/19	9/30/19
Task 2.0 Demonstration of Sample Runs of an SDN System		4/30/20	10/1/19	5/30/20
Subtask 2.1 Demonstration of Installation of Software on Controllers and Switches	10/1/19	11/30/19	10/1/19	12/31/19
Subtask 2.2 Demonstration of Traffic Flows Between SDN Switches		1/30/20	12/1/19	2/28/20
Subtask 2.3 Demonstration of Query for Rules		2/28/20	2/1/20	3/31/20
Subtask 2.4 Demonstration of Traffic Flow Handling Based on Rule Specifications	3/1/20	4/30/20	3/1/20	5/30/20
Task 3.0 Demonstration of a P2P Inquiry Platform in the SDN System	5/1/20	4/30/21	5/1/20	4/20/21
Subtask 3.1 A Justification Report of the Choice of a P2P Open-Source Package	5/1/20	5/30/20	5/1/20	6/15/20
Subtask 3.2 Demonstration of Querying Rules from the P2P System	6/1/20	11/30/20	6/1/20	3/15/21
Subtask 3.3 Making SDN Forwarding Switches to Query Rules from the Inquiry Platforn	12/1/20	4/30/21	12/1/20	4/20/21
Task 4.0 Demonstration of Use Case of Identifying a Compromised Controller	5/1/21	8/31/22	2/1/21	
Subtask 4.1 Demonstration of a Blockchain System Running on Top of a P2P System	5/1/21	10/1/21	2/1/21	3/15/22
Subtask 4.2 Demonstration of Replicated Rules in Blockchain System	11/1/21	1/30/22	12/1/21	3/31/22
Subtask 4.3 Demonstration of Storing Replicated Data Chunks in Blockchain System	2/1/22	6/1/22	3/15/22	
Subtask 4.4 Demonstration of Identifying a Compromised Controller	7/1/22	8/30/22		

#### Accomplishments Before 2021 Review Meeting (1)

### All software development is performed in a cloud infrastructure running across 3 servers

- □ Hardware: 3 high-end Dell servers (Model PowerEdge R540).
- □ Software: Proxmox Virtual Environment (PVE) and OpenStack.

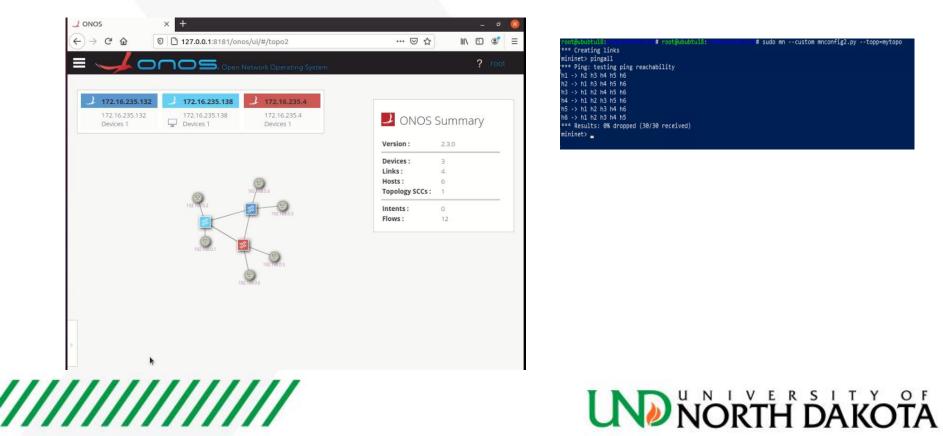


Ð 00 € 172 16 234 18 XPROXMOX Virtual Environment 6.4-13 Documental Server View Virtual Machine 115 (Ubuntu14.04-6-amd64) on node 'prox6node0' ▶ St... () Shutdo. ≪ Migr... >... Cons Q H. Datacenter (ProxmoxCluster) Summary prox6node01 106 (ACEO-base-Node-2) > Console Ubuntu14.04-6-amd64 107 (ACEO-base-Node-1) Hardware 110 (ACEO-base-Node-3) Cloud-Init i Status 119 (Ubuntu20.10-1-amd64) stopped 131 (ubuntu20.10-1-amd64) Status: stopped Options 😻 HA State none 200 (ACEO-base) Task Histor Node prox6node01 100 (CentOS8-MasterNode) Monitor CPU usage 0.00% of 8 CPU(s) 115 (Ubuntu14.04-6-amd64) Backup 116 (Ubuntu18.04-5-64-bits) 🚥 Memory usag 0.00% (0 B of 8.00 GiB) 117 (Ubuntu18.04-5-64-bits) 13 Replication A Bootdisk size 128.00 GiB 118 (Ubuntu18.04-5-64-bits) Snapshots 121 (CentOS8-Node2) ≓ IPs No Guest Agent configured Firewall 122 (CentOS8-Node3) 123 (CentOS8-Node4) Permissions S. local (prox6node01) prox6node02 CPU usag CPU usage 101 (ACO-base) 103 (ACEO-base-Node-1) 108 (ACEO-base-Node-2) 1.8 113 (ACEO-base-Node-3) 1.6 120 (Ubuntu-18.04-amd64) 124 (MasterNode1) 961.2 125 (MasterNode2) ▲ 126 (MasterNode3) ▲ 127 (WorkerNode1) 🛃 129 (Ubuntu20.04-2.0-amd64) 0.4 132 (Ubuntu20.04-2.0-amd64) 133 (Copy-of-VM-Ubuntu20.04-2.0-amd64) 2022-04-12 2022-04-12 2022-04-12 2022-04-12 2022-04-12 2022-04-12 2022-04-12 2022-04-12 2022-04-12 2022-04-12 2022-04-12 2022-04-12 17.29.00 17.47.00 17.47.00 17.47.00 17.53.00 17.53.00 17.53.00 18.01. 41 (Ubuntu20.04-2.0-AMD64 17:23:00 Iocal (prox6node02) prox6node03 Memory usage Total RAM usage Pool-Cluster log End Time Status Apr 12 18:31:54 VM 115 - Stop OK Apr 12 18:31:53 Apr 12 18:31:36 Apr 12 18:31:33 VM 132 - Stor OK Apr 12 18:30:26 Apr 12 18:31:26 VM 115 - Shutdowr Error: VM guit/powerdown fa Apr 12 18:30:13 Apr 12 18:31:13 root@pan VM 132 - Shutdown Error: VM guit/powerdow Apr 12 08:46:09 Apr 12 08:55:19 VM/CT 141 - Console prox6node01 root@nam 

#### Accomplishments Before 2021 Review Meeting (2)

#### Constructed the SDN with a controller cluster.

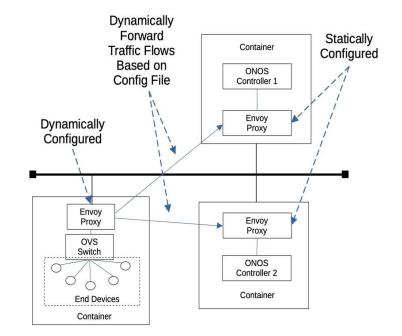
- □ Mininet is used for simulating the data plane of the SDN (DP-SDN).
- □ A cluster of ONOS controllers is used as the control plane of the SDN (CP-SDN).
- □ A cluster of Atomix agents is needed for forming a cluster of ONOS controllers.



#### Accomplishments Before 2021 Review Meeting (3)

- Constructed the portal service layer to bridge the data plane and the control plane of an SDN.
  - The portal service layer is materialized in the form of a service mesh which consists of
    - ✤ A data plane: a set of *Envoy* proxies.
    - ✤ A control plane: a set of Consul agents.
  - OVS switches interact with ONOS cluster through the portal service layer.

- Only the data plane (Envoy proxy) has been successfully functional.
- Envoy proxy has been manually configured to dynamically forward SDN traffic to a target ONOS controller.

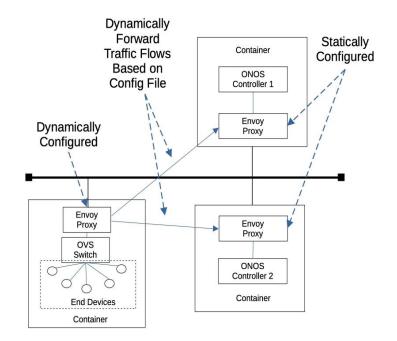




#### Accomplishments Since 2021 Review Meeting (1)

- Added a cluster of Consul agents as the control plane of the portal service layer.
- Consul agents maintain the configurations of the routing paths between an OVS switch and an ONOS controller.
  - Configurations can be dynamically changed to exclude the compromised SDN controllers after being detected by the detection program.
- Envoy proxy is configured to obtain rules of forwarding SDN traffic from the (currently active) Consul agent in a Consul cluster.

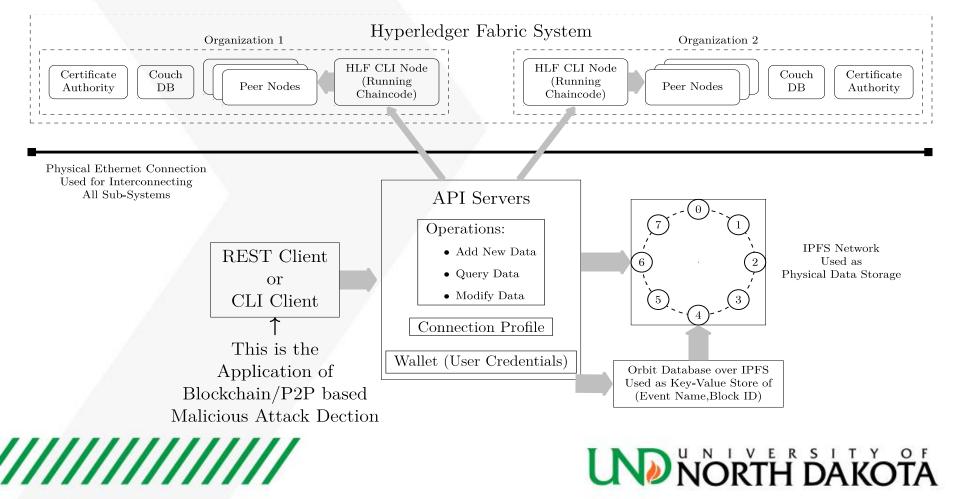






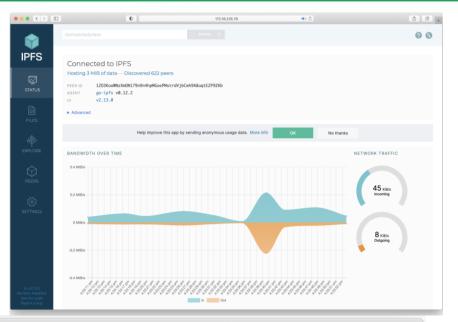
Accomplishments Since 2021 Review Meeting (2)

#### Structure of the Hyperledger Fabric/IPFS sub-System



#### Accomplishments Since 2021 Review Meeting (3)

- Deployed the InterPlanetary File System (IPFS) docker container.
  - Interaction with the IPFS data storage can be made through command-line interface through using the curl command.
  - Web user-interface of the IPFS data storage is accessible from remote hosts.



Figures — -bash — 80×7

localadminsimac:Figures junliu\$ curl -F file=111 http://172.16.235.78:5001/api/v 0/add

{"Name":"QmewtY21Ufyqa166AWXrSC3Y3FMqKMhpRRPH2tqTNMUALE","Hash":"QmewtY21Ufyqa16 6AWXrSC3Y3FMqKMhpRRPH2tqTNMUALE","Size":"11"}

localadminsimac:Figures junliu\$ curl -X POST "http://172.16.235.78:5001/api/v0/c]
at?arg=QmewtY21Ufyqa166AWXrSC3Y3FMqKMhpRRPH2tqTNMUALE"

111
localadminsimac:Figures junliu\$

#### Accomplishments Since 2021 Review Meeting (4)

- The entire docker cluster of a HLF/IPFS sub-system can be launched through running a single bootstrapping script.
  - > More peers or organizations can be added by modifying the bootstrapping script.

		PIIX-1996: ~/tmp/private-network-lpfs — ssh 172.16.235.78 — 206×32		
junliu@junliu-Standard-PC-i440FX-PIIX-1996: Mon 11 Apr 2022 04:37:32 PM CDT	<pre>~/tmp/private-network-ipfs\$ date</pre>			
junliu@junliu-Standard-PC-i440FX-PIIX-1996:	<pre>~/tmp/private-network-ipfs\$ docker ps</pre>			
CONTAINER ID IMAGE				COMMAND
CREATED STATUS bf5595526c38 ipfs/go-ipfs:latest	PORTS		NAMES	N/-bin/bini /
	0 0 0 0·4001->4001/tcp 0 0 0 0·5001->5001/tcp 400	1/udp, 0.0.0.0:8080->8080/tcp, :::4001->4001/tcp, 8081/tcp		"/sbin/tini /usr
		47d90577cf90c945d1a2-3896b0e9005bc21dba92b655fd44e7206e0172a7		"docker-entrypoint.
2 weeks ago Up 2 weeks			dev-peer0.org2.example.com-	and a second second with a second
a3bd5d3837a30cd1117b01503526c20247d90577cf9	0c945d1a2			
698926438b2d dev-peer0.org1.example.com-f	abipfs_1-f21ebc8c5a32a3bd5d3837a30cd1117 <u>b01503526c202</u> 4	47d90577cf90c945d1a2-67f617e5add4d93fc7c23d06e42ef434f85b7fd4	872a860d7c8121ad67895cd8	"docker-entrypoint
2 weeks ago Up 2 weeks			dev-peer0.org1.example.com-	fabIPFS_1-f21ebc8c
a3bd5d3837a30cd1117b01503526c20247d90577cf9	0c945d1a2			
e351fa854a39 hyperledger/fabric-tools:2.4				"/bin/bash"
2 weeks ago Up 2 weeks			cli	Uncon node stant!
58f9db0e18fe hyperledger/fabric-peer:2.4 2 weeks ago Up 2 weeks	0.0.0.0:7051->7051/tcp, :::7051->7051/tcp, 0.0.0.0::	17051->17051/tcn ····17051->17051/tcn	peer0.org1.example.com	"peer node start"
4e5737536aef hyperledger/fabric-peer:2.4	6.6.6.6.7651-776517tcp,7651-776517tcp, 6.6.6.6.	1/051->1/051/tcp,1/051->1/051/tcp		"peer node start"
2 weeks ago Up 2 weeks	0.0.0.0:9051->9051/tcp, :::9051->9051/tcp, 7051/tcp	. 0.0.0.0:19051->19051/tcp. :::19051->19051/tcp	peer0.org2.example.com	peer node seare
44570807d812 couchdb:3.1.1				"tini /docker-e
2 weeks ago Up 2 weeks	4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp, :::5984-	->5984/tcp	couchdb0	
e947b5a0daa9 couchdb:3.1.1				"tini /docker-e
2 weeks ago Up 2 weeks	4369/tcp, 9100/tcp, 0.0.0.0:7984->5984/tcp, :::7984-	->5984/tcp	couchdb1	
0948becd7674 hyperledger/fabric-orderer:2				"orderer"
2 weeks ago Up 2 weeks	0.0.0.0:7050->7050/tcp, :::7050->7050/tcp, 0.0.0.0::	17050->17050/tcp, :::17050->17050/tcp	orderer.example.com	Nob - Laborator -
3085e03d395c hyperledger/fabric-ca:latest 2 weeks ago Up 2 weeks	0.0.0.0:9054->9054/tcp, :::9054->9054/tcp, 7054/tcp	0 0 0 0:100E4->100E4/tcp :::100E4->100E4/tcp		"sh -c 'fabric-ca-
z weeks ago op z weeks c00b76d2d14d hyperledger/fabric-ca:latest	0.0.0.0.9034-29034/tcp,9034-29034/tcp, /034/tcp	, 0.0.0.0.19034-219034/tcp,19054-219054/tcp	ca_orderer	"sh -c 'fabric-ca-
2 weeks ago Up 2 weeks	0.0.0.0:8054->8054/tcp, :::8054->8054/tcp, 7054/tcp	0 0 0 0.18054->18054/tcp :::18054->18054/tcp	ca_org2	Sil - C Pablic-Ca-
cd051332be63 hyperledger/fabric-ca:latest		, otototot.2003 / 2003 // 2003 // 2003 // 2003 // 2003 // 2003		"sh -c 'fabric-ca-
2 weeks ago Up 2 weeks	0.0.0.0:7054->7054/tcp, :::7054->7054/tcp, 0.0.0.0::	17054->17054/tcp, :::17054->17054/tcp	ca org1	
junliu@junliu-Standard-PC-i440FX-PIIX-1996:				

#### Accomplishments Since 2021 Review Meeting (5)

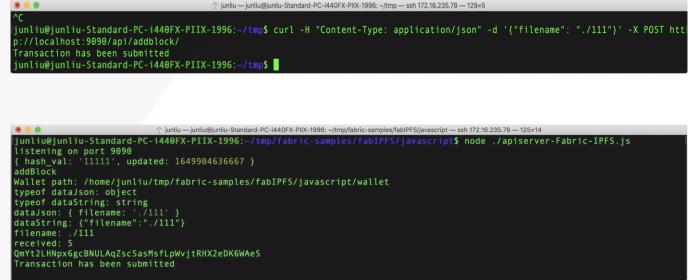
Command-line interaction can be supported between the CLI client and the API server.



Sample Run: the client sends a request to the API server for adding a file to the HLF blockchain.

At client: curl command sent to API server and the response received from API server.

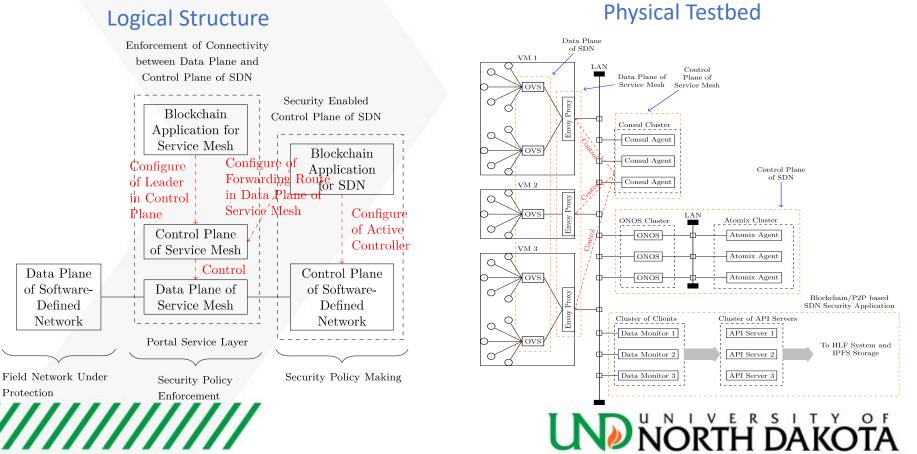
API server: serving client's request and displaying the response received from IPFS and HLF.





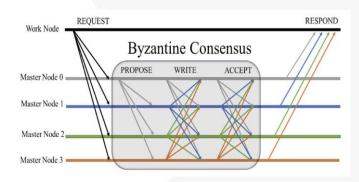
#### Accomplishments Since 2021 Review Meeting (6)

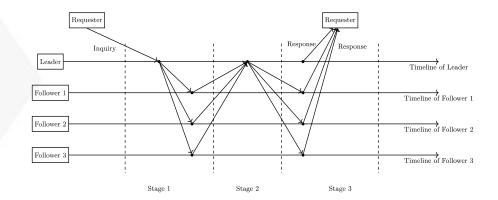
Following the deployment of HLF/IPFS sub-system, simplification has been made to the structure of the testbed of detecting compromised SDN controllers.



#### Accomplishments Since 2021 Review Meeting (7)

Performed the literature study and the planning step toward an efficient BFT consensus mechanism..





Predominant BFT consensus mechanism. (Figure is from a public website).



Planned BFT consensus mechanism



#### Accomplishments Since 2021 Review Meeting (8)

- Developed a basic analytical framework as the basis for constructing an efficient cryptographic BFT consensus mechanism.
- The basic analytical framework is an information-theoretic scheme which can provide unconditional security, rather than computational security.
- In this analytical framework, two parties individually derive their own local outcomes using the private values and the publicly disseminated values.
- This analytical framework can be converted to establish different cryptographic solutions, as well as being extended to more than 2 parties.
- This analytical work is still under development. Proof of correctness and security analysis are very challenging.





### Next Steps (1)

Develop a BFT consensus algorithm with linear communication overhead.

- □ We are close to finish developing a 2-rounds group key agreement protocol which allows a group of participants to agree upon a common secret value.
- The communication overhead of this group key agreement protocol is linear in the number of participants.
- □ We plan to convert this group key agreement protocol into a BFT consensus algorithm.
- > Prototyping the BFT consensus algorithm by modifying the Raft code.

➢ Risk:

- □ Prototyping the BFT consensus algorithm may be risky and time consumptive.
- □ Many open-source implementations of practical BFT (pBFT) algorithm are incomplete.

Risk mitigation:

Bottom line: we will ensure pBFT with quadratic communication overhead to be functional to provide BFT consensus in the application of detection.





#### Next Steps (2)

- ➤To develop a blockchain/P2P based application for detecting and excluding a compromised SDN controller.
  - The detection application only relies on the passive snooping on network traffic.
    - In order to avoid technical complications, the application does not rely on obtaining operational data from ONOS/Atomix and Envoy/Consul service mesh.
  - Detection: A compromised SDN controller is the one which had issued inconsistent rule to SDN switches.
  - Exclusion: The blockchain application configures the Discovery Service in the Consul cluster to exclude a compromised SDN controller after the detection.





# Acknowledgment

This material is based upon work supported by the Department of Energy Award Number DE-FE0031742.

Disclaimer:

"This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof."





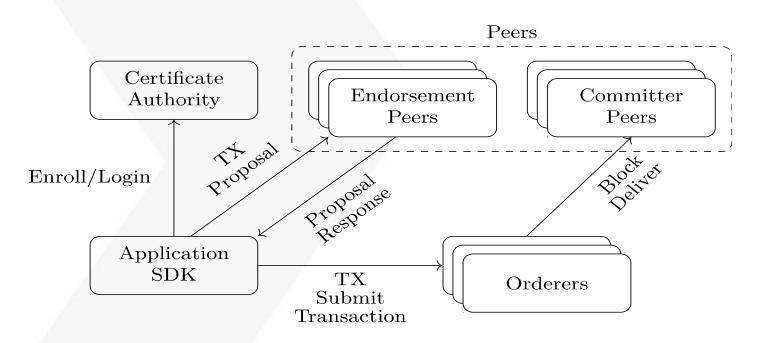
### **Backup Slides**





#### Accomplishments Since 2021 Review Meeting

Logical flowchart of making a transaction in the Hyperledger Fabric framework

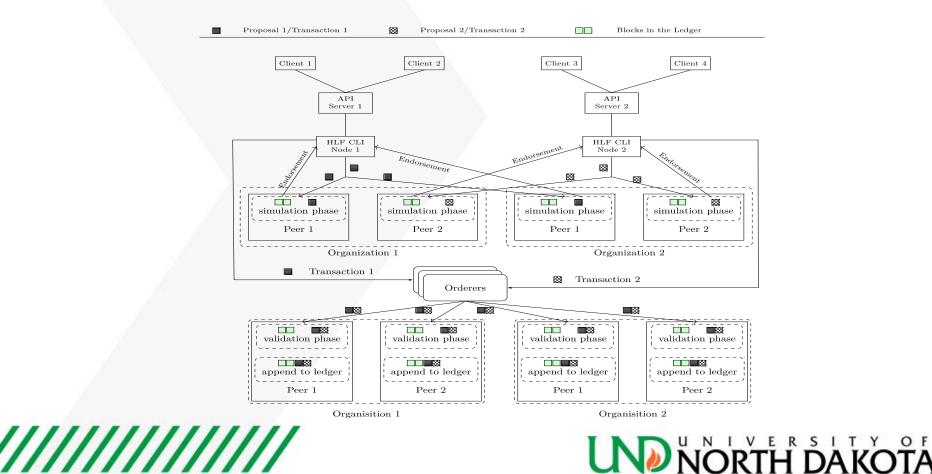






#### Accomplishments Since 2021 Review Meeting

#### Actions taken for adding blocks in the Hyperledger Fabric framework



#### Accomplishments Since 2021 Review Meeting

