Adding Network Fuzzing Capabilities to Hyperledger Umbra

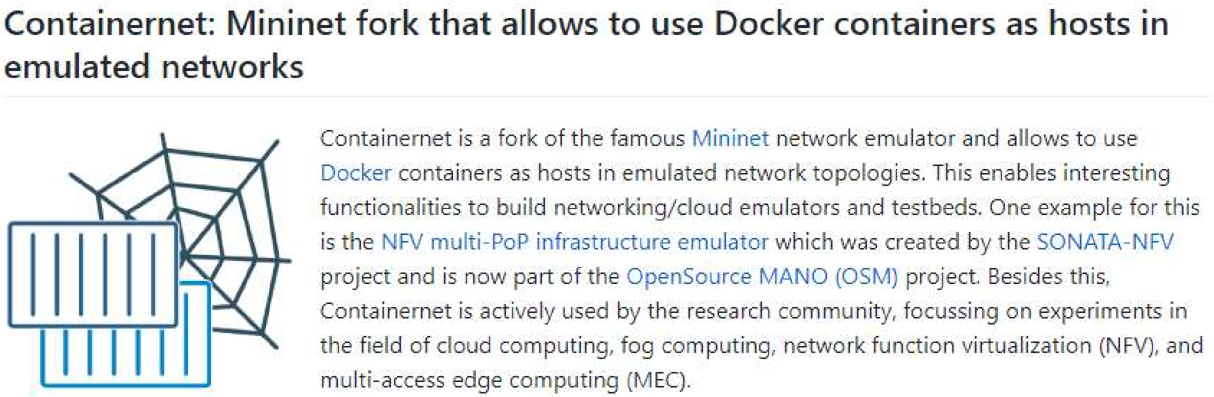
August 2020

Hyperledger Umbra

- **Introduction**
  - **Name**: Badiuzzaman Iskhandar
  - **Location:** Göteborgs, Sweden. Originally from Malaysia
  - **University:** Chalmers University of Technology (CTH), Sweden
  - **Mentor(s):** David Huseby, Raphael Rosa
  - **Hyperledger Project**: Adding Network Fuzzing Capabilities to Hyperledger Umbra

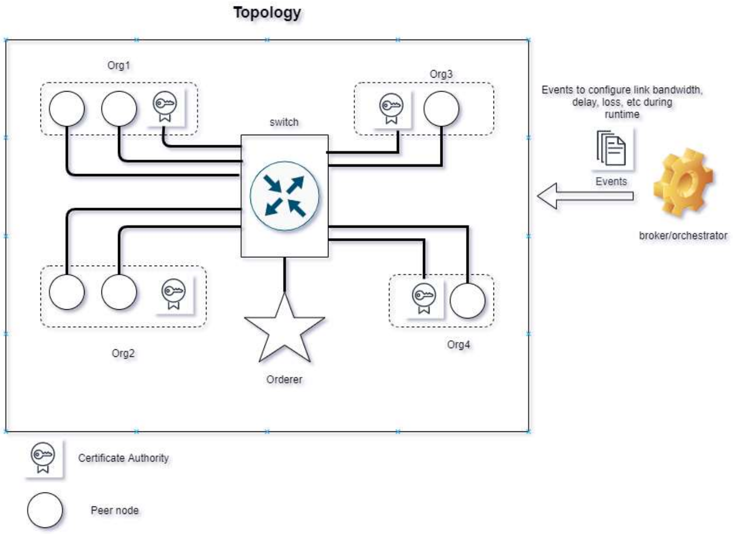
Hyperledger Umbra

- **Project Description**: Umbra is an emulation platform to run a blockchain network. *mininet/containernet* as the main ingredient. The aim for this year is for Umbra to support dynamic environment modification in runtime and network stimulus



Containernet: Mininet fork that allows to use Docker containers as hosts in emulated networks

Containernet is a fork of the famous Mininet network emulator and allows to use Docker containers as hosts in emulated network topologies. This enables interesting functionalities to build networking/cloud emulators and testbeds. One example for this is the NFV multi-PoP infrastructure emulator which was created by the SONATA-NFV project and is now part of the OpenSource MANO (OSM) project. Besides this, Containernet is actively used by the research community, focussing on experiments in the field of cloud computing, fog computing, network function virtualization (NFV), and multi-access edge computing (MEC).



Hyperledger Umbra

- **Project Objectives**:
  - Obj 1: Add capabilities to change environment behavior, e.g., change a node cpu/mem resources, disconnect/reconnect a node in the network
  - Obj 2: Integrate umbra-agent and umbra-monitor component
  - Obj 3: Run a test scenario that demonstrate environment changes. In particular, in a Fabric network, simulate a network partition for the orderer node by turning of its link interface. Generate report after the test completes

Hyperledger Umbra

- **Project Deliverables:**
  - Deliverable 1: Modify environment behavior during runtime – PR at hyperledger-labs/umbra under review
  - Deliverable 2: Integrate umbra-agent, umbra-monitor, test report, dashboard, update docs – PR under review as well

Hyperledger Umbra

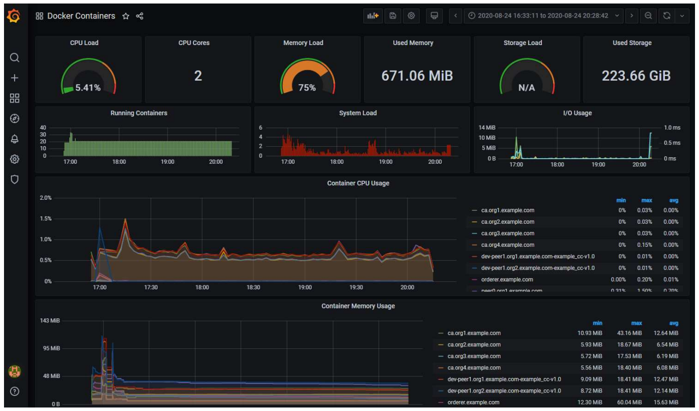
- **Project Execution & Accomplishments**:
  - A peak into mininet internals and Linux networking stack – used mininet in a course before but did not know how all these “magics” work. It uses some Linux network configuration commands like `*ifconfig*` or `*ip*` for creating virtual network interfaces, `*tc*` to change interface behavior (like bandwidth and delay), plus other SDN tools like OpenFlow
  - Learn more about Python, especially the concurrent execution model
  - Missing events like start/stop a container (node)
  - Better simulation report instead of just parsing output log. This might need more changes in umbra-broker to concurrently receive and parse responses from the scheduled events

Hyperledger Umbra

- **Recommendations for future work:**
  - Add a UI panel in Grafana dashboard that monitor topology changes (node leaves/join the network) across time
  - Unify the API for event creations (fabric, agent, monitor, environment, etc). Likely needs major refactor

Hyperledger Umbra

- **Project Output or Results:**
  - Monitoring dashboard
  - Simulation report



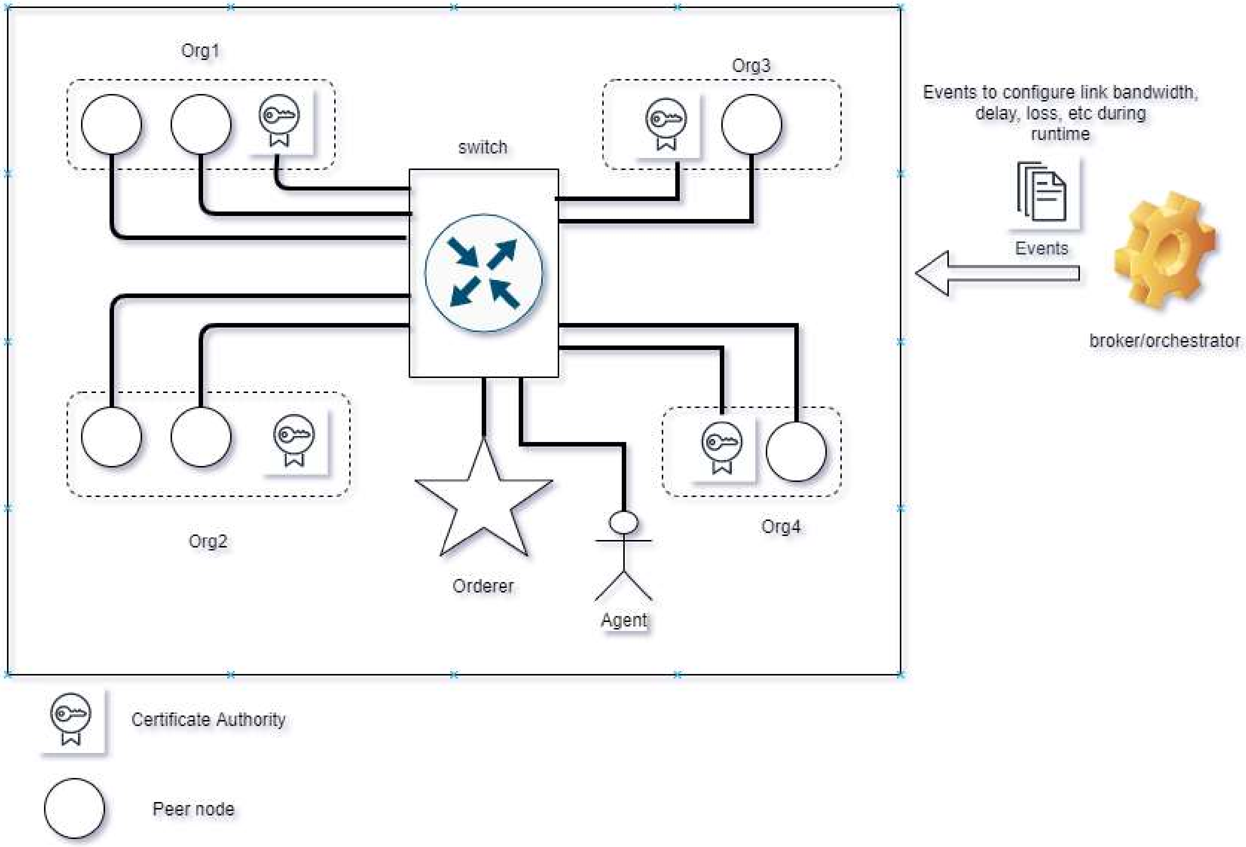
Container CPU Usage

	min	max	avg
ca.org1.example.com	0%	0.03%	0.00%
ca.org2.example.com	0%	0.03%	0.00%
ca.org3.example.com	0%	0.03%	0.00%
ca.org4.example.com	0%	0.15%	0.00%
dev-peer1.org1.example.com-example_cc-v1.0	0%	0.01%	0.00%
dev-peer1.org2.example.com-example_cc-v1.0	0%	0.01%	0.00%
orderer.example.com	0.00%	0.20%	0.01%
peer0.org1.example.com	0.31%	1.50%	0.70%

Container Memory Usage

	min	max	avg
ca.org1.example.com	10.93 MiB	43.16 MiB	12.64 MiB
ca.org2.example.com	5.93 MiB	18.67 MiB	6.54 MiB
ca.org3.example.com	5.72 MiB	17.53 MiB	6.19 MiB
ca.org4.example.com	5.56 MiB	18.40 MiB	6.08 MiB
dev-peer1.org1.example.com-example_cc-v1.0	9.09 MiB	18.41 MiB	12.47 MiB
dev-peer1.org2.example.com-example_cc-v1.0	8.72 MiB	18.41 MiB	12.14 MiB
orderer.example.com	12.30 MiB	60.04 MiB	15.63 MiB

Topology



Events to configure link bandwidth, delay, loss, etc during runtime

Hyperledger Umbra

- **Insights Gained:**
  - Learn how to navigate and read the language/framework official reference manual and documentation, e.g. https://docs.python.org/3/. Paradox of information overload: old tutorials, obsolete answers, outdated blogposts, etc.
  - Typical challenges of remote work. Try to think and write the problem statement clearly. Sometimes you will get to the answer in the process of writing a coherent and sound statement

THANK YOU!