Analyzing Hyperledger Fabric Ledger, Transactions, and Logs using Elasticsearch and Kibana

Title	Analyzing Hyperledger Fabric Ledger, Transactions, and Logs using Elasticsearch and Kibana
Status	**PROJECT COMPLETED**
Difficulty	**MEDIUM**

Description

Each blockchain platform, including Hyperledger Fabric, provide a way to record information on blockchain in an immutable manner. In the case of Hyperledger Fabric, information is recorded as a `key-value` pair. All previous updates to a `key` are recorded in the ledger, but only the latest value of a `key` can be easily queried using CouchDB; the previous updates are only available in ledger files. This mechanism makes it challenging to perform analysis of updates to a `key`, a necessary requirement for information provenance.

The goal of this project is to

- (1) write a Elastic beats module (in Go), that will ship ledger data to Elasticsearch instance
- (2) create generic Kibana dashboards that will allow selection of a particular key and visualization updates to it (channel, id, timestamp etc)

Time permitting, the dashboards can be extended to analyze Fabric logs and in-progress transaction data, as well as creating dashboards similar to Hyperledger Explorer.

Of course, a blockchain solution can track information provenance in multiple ways. In one such mechanism, a solution may always write new key-value pairs to blockchain, and maintain the relationship among key-value pairs within the solution (off-chain), instead of blockchain. This project does not concern itself on how a solution manages relationship among key-value pairs.

Additional Information

- You must be a self-starter and have experience writing Go code. Familiarity with ELK stack is also desired.
- You must be able to solve technical problems and come up with creative solutions.

Learning Objectives

- You will become familiar with Hyperledger Fabric, and ledger analysis using Elastic stack. You will learn to generate dummy transactions in Fabric, and analyze the ledger. You will also learn Go.
- The expectation is at least one (remote) meeting once a week, with several meetings to kick-off the project or as needed.

Expected Outcome

A open source implementation, eventually available as Hyperledger Labs, containing:

- Elastic beats plugin for Hyperledger Fabric
- Kibana dashboards
- Dashboards similar to Hyperledger Explorer
- Create a setup for generating various dummy data in various configurations
  - One peer / CA / order, single user for initial testing
  - A four peers/CA setup with two channels, and two users each associated with two peers. Select (e.g.) 10 keys (through configuration file), to which these users write data, for at least one value per key.

Relation to Hyperledger

- Hyperledger Fabric (ledger analysis)

Education Level

Undergraduate or graduate

Skills

- Elastic beats module
- Familiarity with Go
- Familiarity with using ELK stack and creating Kibana dashboards

Future plans

Create a Hyperledger Lab or incubation project for analyzing ledgers.

Preferred Hours and Length of Internship

Full-time (40 hours a week for 12 weeks during the summer)

Mentor(s) Names and Contact Info

Salman Baset

salman.a.baset@gmail.com

Rocketchat id: salmanbaset

Mentee Name and Contact Info

Balazs Prehoda

prehoda.balazs@gmail.com

Rocketchat id: balazsprehoda

Project Deliverables

Overall goals of the project:

1. write an Elastic beats module (in Go), that will ship ledger data to Elasticsearch instance
2. create generic Kibana dashboards that will show both the operational and data aspects of ledger data. Basically, allow selection of a particular key or a channel configuration, and visualize updates to it (channel, id, timestamp etc)

Week 1	☑ Exploration. Setup Hyperledger Fabric network, connect Hyperledger Explorer to Fabric. Use Filebeat to send data to Kibana.
Week 2	☑ Exploration. Extend Fabric network (Fabric-ca, binary data and json chaincode). Dump ledger data from HL Explorer and visualize it in Kibana.
Week 3	☑ Write Beats agent with configuration that sends data to Elasticsearch.

Week 4	☑ Create operational dashboards similar to HL Explorer. Create data query dashboards.
Week 5	☑ Refine data flow, send every block and transaction data to Elasticsearch. Make keys indexable.
Week 6	☑ Refactor code, prepare the system to receive data from various peers in separate or similar indices. Add peer selection functionality. Add which user issued query in beats agent and in dashboard. Modify the application such that: 1) it only writes the most recently added keys to ledger 2) adds a previous key in the data schema, which can be added to key value in addition to hash.
Week 7	☑ Test agent with multiple peers, multiple channels. Each channel may have its own zero or more chaincodes and data schema. It should be possible to specify per channel chaincode data schema in beats agent.
Week 8	☑ Create example HL Fabric network setups and dashboards for different topics and use-cases (supply chain, medicine provenance, etc.)
Week 9	☑ Refine the examples and prepare for submission as Hyperledger Lab. Evaluate how to read data directly from ledger file instead of using peer APIs.
Week 10	☑ Submit the project as Hyperledger Lab.
Week 11	☑ Create program that dumps data into custom output (default implementation is json, but can be implemented for any databases) for exploring analysis possibilities aside from Elasticsearch.
Week 12	☑ Refine documentation, evaluate how to replace the ledger file with custom database (CouchDB, MongoDB).

Project Milestones

☑ Write a Beats agent that sends ledger data to Elasticsearch Balazs Prehoda23 Jun 2019 (1st Quarter)

☑ Prepare the system to receive data from various peers, refine the data flow Balazs Prehoda14 Jul 2019 (2nd Quarter)

☑ Create examples, prepare for submission as Hyperledger Lab Balazs Prehoda04 Aug 2019 (3rd Quarter)

☑ Prepare the project for excluding the Beats agent by replacing the ledger file or reading the ledger file directly. Balazs Prehoda25 Aug 2019 (4th Quarter)

Project Plan

Project_plan_final.pdf
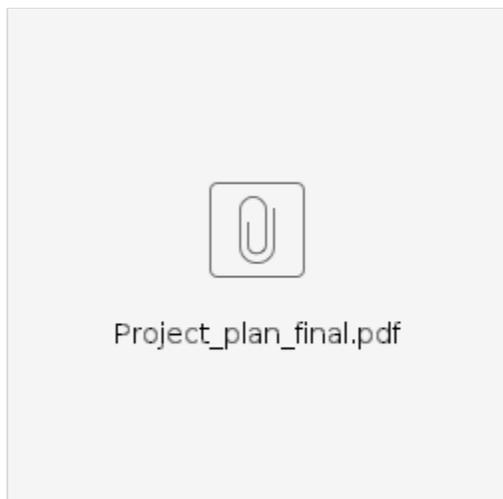
Summary Report

Slides

Hyperled...2019.pdf
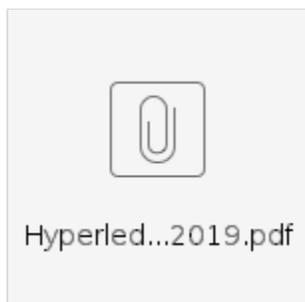
Two short demo videos (without audio):

multichannel_demo.mp4

applechain_demo.mp4