

# Analyzing Hyperledger Fabric Ledger Transactions using Elasticsearch and Kibana

## August, 2019

# Analyzing Hyperledger Fabric Ledger Transactions using Elasticsearch and Kibana

## › Introduction

- › **Name:** Balázs Prehoda
- › **Location:** Veresegyház, Hungary
- › **University:** Budapest University of Technology and Economics, Budapest, Hungary
- › **Mentor(s):** Salman Baset
- › **Hyperledger project:** Analyzing Hyperledger Fabric Ledger Transactions using Elasticsearch and Kibana



# Analyzing Hyperledger Fabric Ledger Transactions using Elasticsearch and Kibana

## › Project Description:

- Main goal: create a mechanism to analyze and display ledger data from both operational and data-oriented aspects, and submit it as Hyperledger Lab project.
- The main components:
  - 3 different Fabric networks with chaincodes written in Go and nodejs (for testing).
  - Applications for user enrollment and transaction submission using the node SDK (for testing) for generating test data.
  - An Elastic Beats agent (fabricbeat) written in Go, that connects to peer(s), queries it, and ships its data to Elasticsearch and Kibana. (core contribution of this project)
  - Elasticsearch cluster and Kibana server.
  - Generic Kibana dashboards similar to Hyperledger Explorer, plus data-centric dashboards. (contribution of this project)

# Analyzing Hyperledger Fabric Ledger Transactions using Elasticsearch and Kibana

## › Project Objectives:

- › Obj 1: Elastic Beats plugin for Fabric.
- › Obj 2: Generic Kibana dashboards.
- › Obj 3: Example network setups and use-cases with dummy data.
- › Obj 4: Evaluate how to read data directly from ledger file.

# Analyzing Hyperledger Fabric Ledger Transactions using Elasticsearch and Kibana

## › Project Deliverables:

- › Deliverable 1: The fabricbeat agent written in Go.
- › Deliverable 2: Explorer-like generic Kibana dashboards.
- › Deliverable 3: Data-oriented generic Kibana dashboards.
- › Deliverable 4: Example network setups with dummy data.
- › Deliverable 5: Prepare the system to receive data from multiple peers.
- › Deliverable 6: Documentation
- › Deliverable 7: Guides for demos

# Analyzing Hyperledger Fabric Ledger Transactions using Elasticsearch and Kibana

## › **Project Execution & Accomplishments:**

- › Majority of the project deliverables are accomplished.
- › I am probably most proud of the configurability of the components, the design of the fabricbeat agent, and the prepared demo setups that can be installed and run with only one make command.
- › The most challenging part was using the Go SDK in the Beats agent to query and process ledger data.
- › I documented a few bugs as issues on Github, they got resolved by now (I am looking for new ones..).

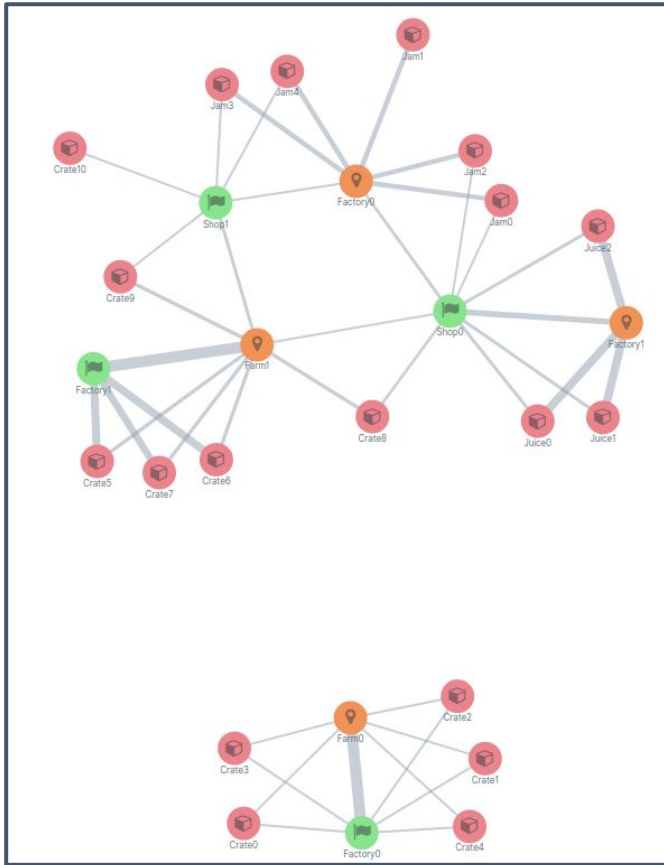
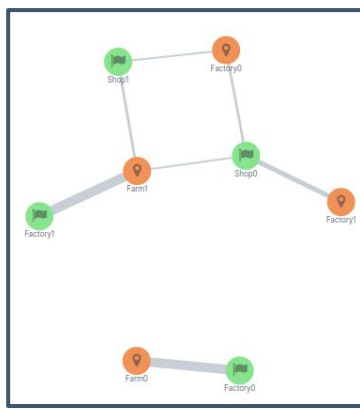
# Analyzing Hyperledger Fabric Ledger Transactions using Elasticsearch and Kibana

## › Recommendations for future work:

- › By reading the ledger file directly (or replacing it with Elasticsearch, MongoDB or CouchDB), the fabricbeat agent could be put aside.
- › A troubleshooting section in the docs would be very helpful.
- › Peer and CA logs could also be analyzed similarly to the ledger.
- › A more distant goal could be to generalize this analysis tool for any blockchain platforms.

# Backup

The facilities in applechain example



The transports in applechain example

### Block\_count Visualization (org1)

16

Block Count

### Peer\_selection Visualization (org1)

peer: peer0.org1.lel-network.com

Apply changes Cancel changes

Clear form

### Channel\_selection Visualization (org1)

channel\_id: fourchannel

Apply changes Cancel changes

Clear form

### Transaction\_count Visualization (org1)

16

Transaction Count

### Transaction\_per\_organization Visualization (org1)

### Transaction\_count\_timeline Visualization (org1)

### Block Search (org1)

Time	block_number	created_at	block_hash	previous_hash	transactions
Aug 5, 2019 @ 14:03:37858	15	Aug 5, 2019 @ 14:03:37858	b0f9e620a921934a	208703348687603	4232046408144
Aug 5, 2019 @ 14:03:35642	14	Aug 5, 2019 @ 14:03:35642	baaf119e44ad572d	4928819126835756c2	83170e4f776076
Aug 5, 2019 @ 14:03:33452	13	Aug 5, 2019 @ 14:03:33452	16718969e804da	7aa246c55d06ba804	8ba39746c7b5f1f1
Aug 5, 2019 @ 14:03:31255	12	Aug 5, 2019 @ 14:03:31255	7aa246c55d06ba804	4c3b1899e633905c	218519658a5d7389

### Transaction Search (org1)

Time	block_number	chaincode_name	chaincode_version	channel_id	created_at
Aug 5, 2019 @ 14:03:37858	15	dummycc	5.5	fourchannel	Aug 5, 2019 @ 14:03:37858
Aug 5, 2019 @ 14:03:35642	14	dummycc	5.5	fourchannel	Aug 5, 2019 @ 14:03:35642
Aug 5, 2019 @ 14:03:33452	13	dummycc	5.5	fourchannel	Aug 5, 2019 @ 14:03:33452

### Key Search (org1)

Time	created_at	key	value.from	value.to	value.asset	chaincode_name	tx_id
Aug 5, 2019 @ 15:17:25961	Aug 5, 2019 @ 15:17:25961	Transport 5	Factory0	Shop1	Jam1	applechain	a6538f8e4d1726f6c3a4889e4c0c7455564f0a72f8e4e1742177a
Aug 5, 2019 @ 15:17:23800	Aug 5, 2019 @ 15:17:23800	Transport 7	Factory0	Shop1	Jam3	applechain	6a27e80817e60172016e013e7850e4148a0c373201463768808f7
Aug 5, 2019 @ 15:17:08756	Aug 5, 2019 @ 15:17:08756	Transport 3	Factory0	Shop0	Jam2	applechain	a39682c7e6680a0e7e45821501218a455a1313201463768808f7
Aug 5, 2019 @ 15:17:06809	Aug 5, 2019 @ 15:17:06809	Transport 8	Factory0	Shop0	Jam1	applechain	3a87788217a64a1c24757850e4148a0c373201463768808f7

Search for transports from Factory0...