Iroha Devs #13

Meeting of Iroha maintainers
with open-source community

HYPERLEDGER
IROHA

いろはにほへと
ちりぬるを
わかよたれそ
つねならむ
うゐのおくやま
けふこえて
あさきゆめみし
ゑひもせすん

Agenda

1. Progress since Iroha Dev #12
2. Current iteration (Sprint 10)
3. Current process and tools used
4. How to suggest changes or improvements
5. Upcoming α release
6. Q&A

HYPERLEDGER IROHA

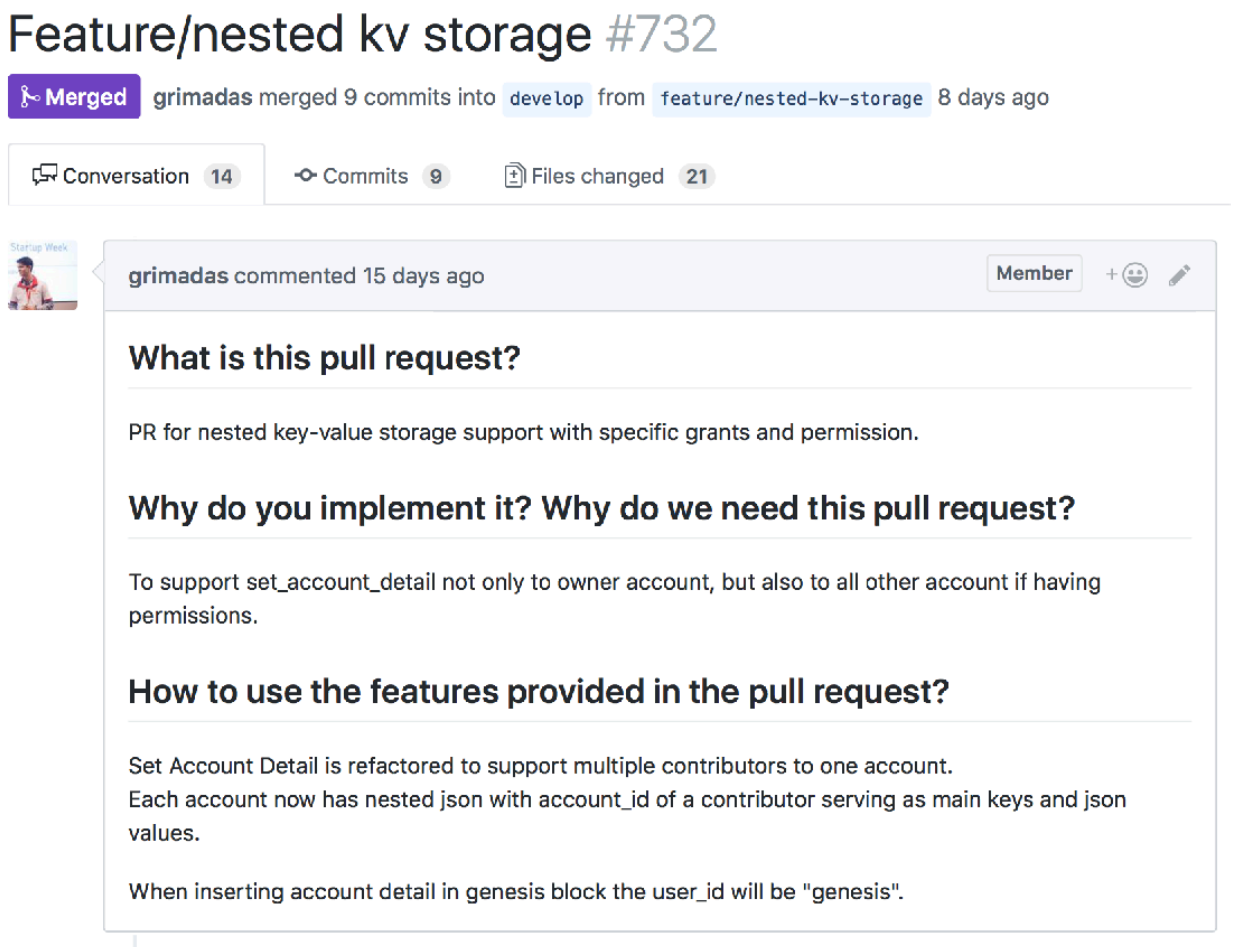
Progress since Iroha Dev #12

Iroha Dev #12 were 1st of December. Since the meeting, we have implemented:

1. Nested key-value storage for account data
2. Changes in SonarQube
3. Homebrew installation
4. Ansible deployment and docs for peer network set up in Ansible
5. Subtract asset quantity command
6. Detach role command
7. ed25519 library with SHA3 hashing
8. SWIG bindings for transaction generation (in Java, Python)
9. GetTransaction API

HYPERLEDGER
IROHA

Nested key-value storage for account data



Feature/nested kv storage #732

Merged grimadas merged 9 commits into `develop` from `feature/nested-kv-storage` 8 days ago

grimadas commented 15 days ago

What is this pull request?

PR for nested key-value storage support with specific grants and permission.

Why do you implement it? Why do we need this pull request?

To support set_account_detail not only to owner account, but also to all other account if having permissions.

How to use the features provided in the pull request?

Set Account Detail is refactored to support multiple contributors to one account.
Each account now has nested json with account_id of a contributor serving as main keys and json values.

When inserting account detail in genesis block the user_id will be "genesis".

HYPERLEDGER IROHA

Nested key-value storage for account data

```
{
"others" : {

  “admin@soramitsu" : {
    “name”:
    “46a8a10ce31963a10ea744b8ff301a0998c4560ddd4903
    1bcae1c3e556e605c81a4c8c1a369249e2bec63869a0f3
    887481c8ef929a5004f28fd3be3985f0af69”
   }

 },
"own" : {
   “name”: “Takemiya”,
   “surname”: “Makoto”
 }
}
```

Hash of data

Data, stored in account

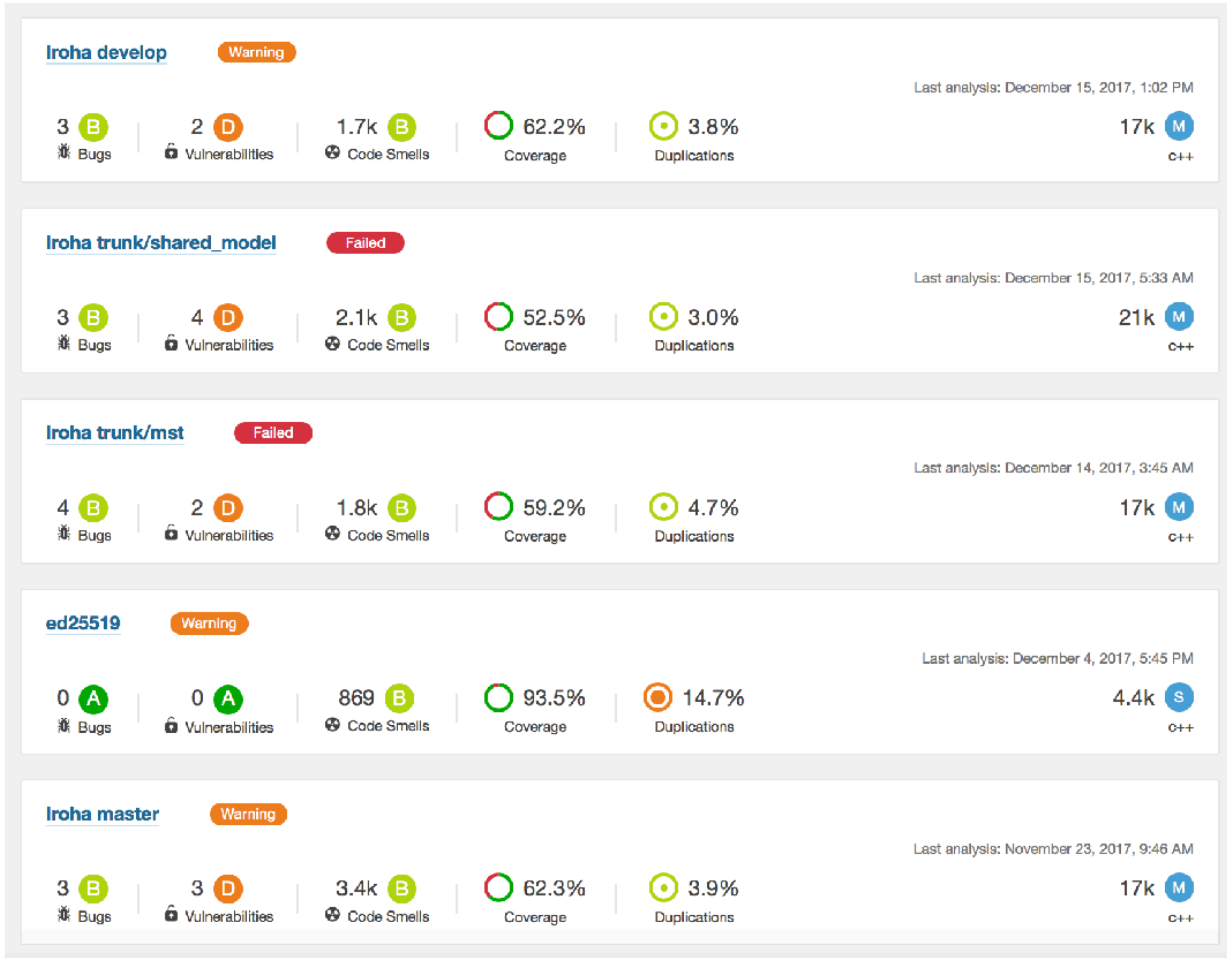
Nested key-value storage for account data

```
message SetAccountDetail{
    string account_id = 1;
    string key = 2;
    string value = 3;
}
```

Can be executed for own account, or for another, if this account has granted permissions to write in the storage

HYPERLEDGER IROHA

SonarQube changes

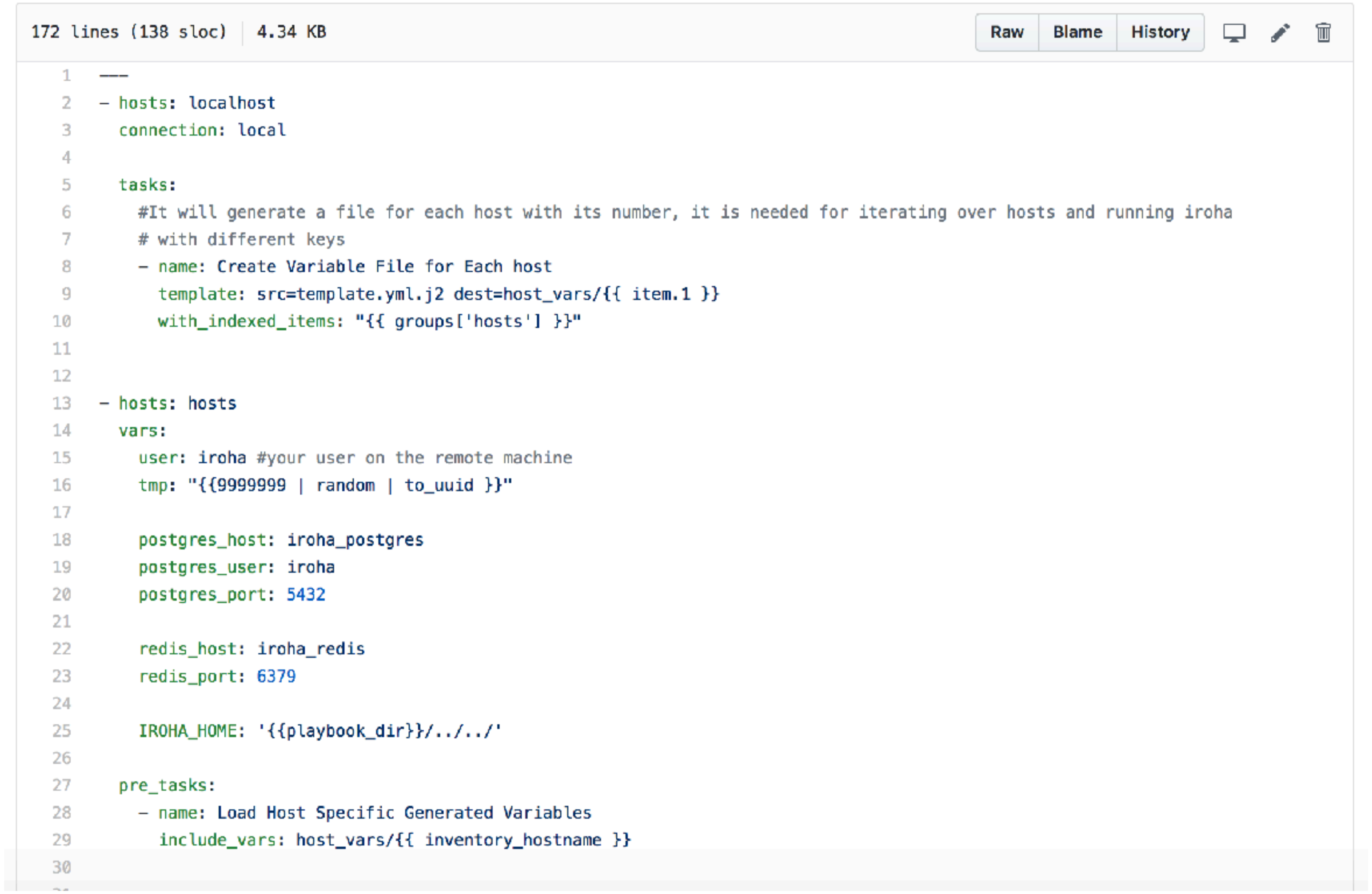


Iroha develop Warning					Last analysis: December 15, 2017, 1:02 PM	
3 B Bugs	2 D Vulnerabilities	1.7k B Code Smells	62.2% Coverage	3.8% Duplications	17k M	C++
Iroha trunk/shared_model Failed					Last analysis: December 15, 2017, 5:33 AM	
3 B Bugs	4 D Vulnerabilities	2.1k B Code Smells	52.5% Coverage	3.0% Duplications	21k M	C++
Iroha trunk/mst Failed					Last analysis: December 14, 2017, 3:45 AM	
4 B Bugs	2 D Vulnerabilities	1.8k B Code Smells	59.2% Coverage	4.7% Duplications	17k M	C++
ed25519 Warning					Last analysis: December 4, 2017, 5:45 PM	
0 A Bugs	0 A Vulnerabilities	869 B Code Smells	93.5% Coverage	14.7% Duplications	4.4k S	C++
Iroha master Warning					Last analysis: November 23, 2017, 9:46 AM	
3 B Bugs	3 D Vulnerabilities	3.4k B Code Smells	62.3% Coverage	3.9% Duplications	17k M	C++

Extended the reports for feature branches to track the coverage.

HYPERLEDGER IROHA

Ansible deployment



172 lines (138 sloc) | 4.34 KB

```
---
- hosts: localhost
  connection: local

  tasks:
    #It will generate a file for each host with its number, it is needed for iterating over hosts and running iroha
    # with different keys
    - name: Create Variable File for Each host
      template: src=template.yml.j2 dest=host_vars/{{ item.1 }}
      with_indexed_items: "{{ groups['hosts'] }}"


- hosts: hosts
  vars:
    user: iroha #your user on the remote machine
    tmp: "{{9999999 | random | to_uuid }}"

    postgres_host: iroha_postgres
    postgres_user: iroha
    postgres_port: 5432

    redis_host: iroha_redis
    redis_port: 6379

    IROHA_HOME: '{{playbook_dir}}/../../'

  pre_tasks:
    - name: Load Host Specific Generated Variables
      include_vars: host_vars/{{ inventory_hostname }}

```

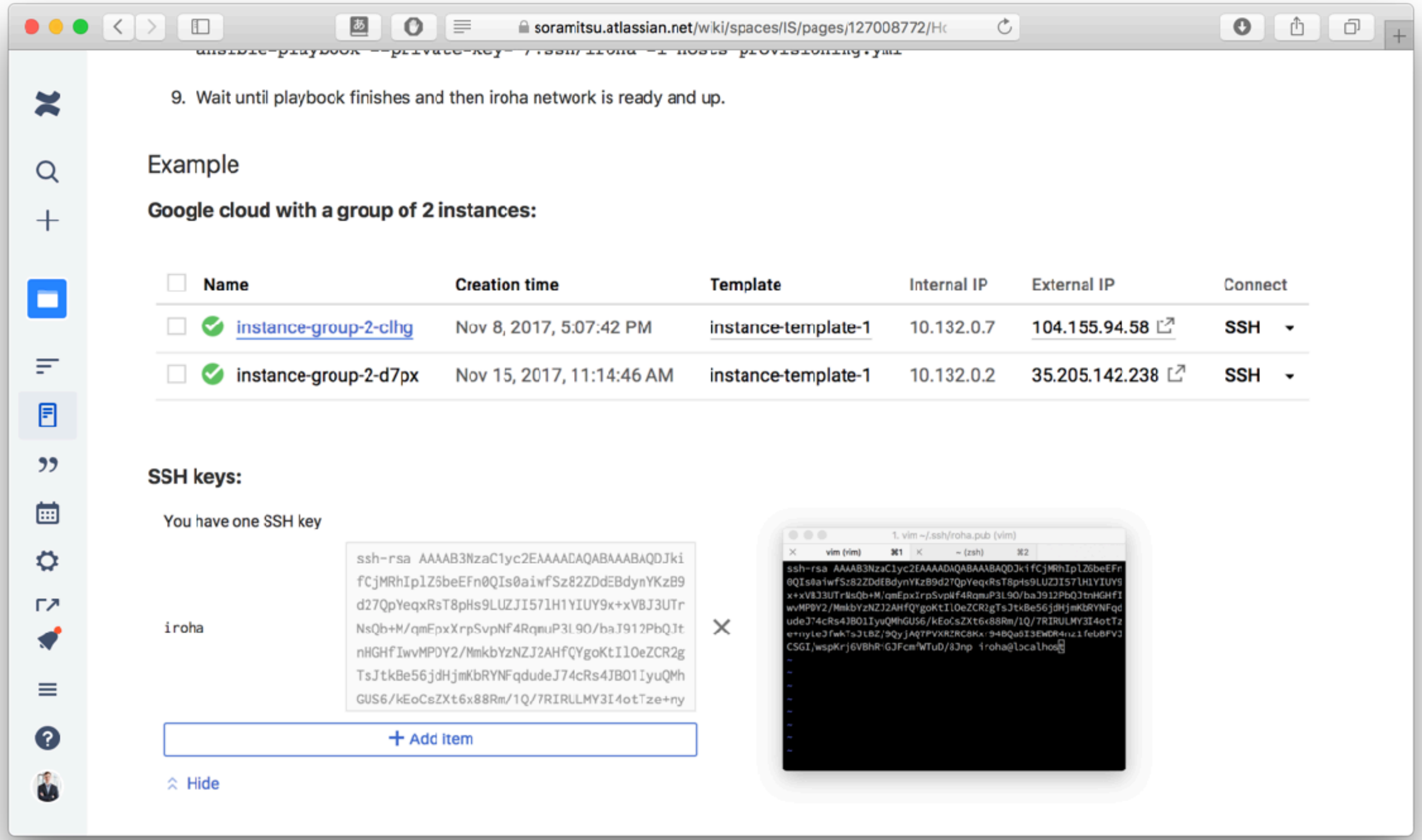
We now use Ansible playbook to set up peer network for arbitrary number of nodes.

Please refer to this guide:

https://soramitsu.atlassian.net/wiki/spaces/IS/pages/127008772/How+to+run+Iroha+network+with+ansible

https://github.com/hyperledger/iroha/blob/master/deploy/ansible/provisioning.yml

Ansible deployment



soramitsu.atlassian.net/wiki/spaces/IS/pages/127008772/H

```
ansible-playbook --private-key=~/.ssh/iroha -i hosts provisioning.yml
```

9. Wait until playbook finishes and then iroha network is ready and up.

Example

Google cloud with a group of 2 instances:

	Name	Creation time	Template	Internal IP	External IP	Connect
☐ ✓	instance-group-2-clhg	Nov 8, 2017, 5:07:42 PM	instance-template-1	10.132.0.7	104.155.94.58	SSH
☐ ✓	instance-group-2-d7px	Nov 15, 2017, 11:14:46 AM	instance-template-1	10.132.0.2	35.205.142.238	SSH

SSH keys:

You have one SSH key

iroha

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDJki
fCjMRhIplZ6beEFn0QIs0aiwfSz82ZDdEBdynYKzB9
d27QpYeqxRsT8pHs9LUZJI57lH1YIUY9x+xVBJ3UTr
NsQb+M/qmEpxXrpSvpNf4RqmuP3L9O/baJ912PbQJt
nHGHfIwvMPDY2/MmkbYzNZJ2AHfQYgoKtIl0eZCR2g
TsJtkBe56jdHjmKbRYNFqdudeJ74cRs4JBO1IyuQMh
GUS6/kEoCsZXt6x88Rm/1Q/7RIRULMY3I4otTze+ny
```

+ Add item

Hide

1. vim ~/.ssh/iroha.pub (vim)

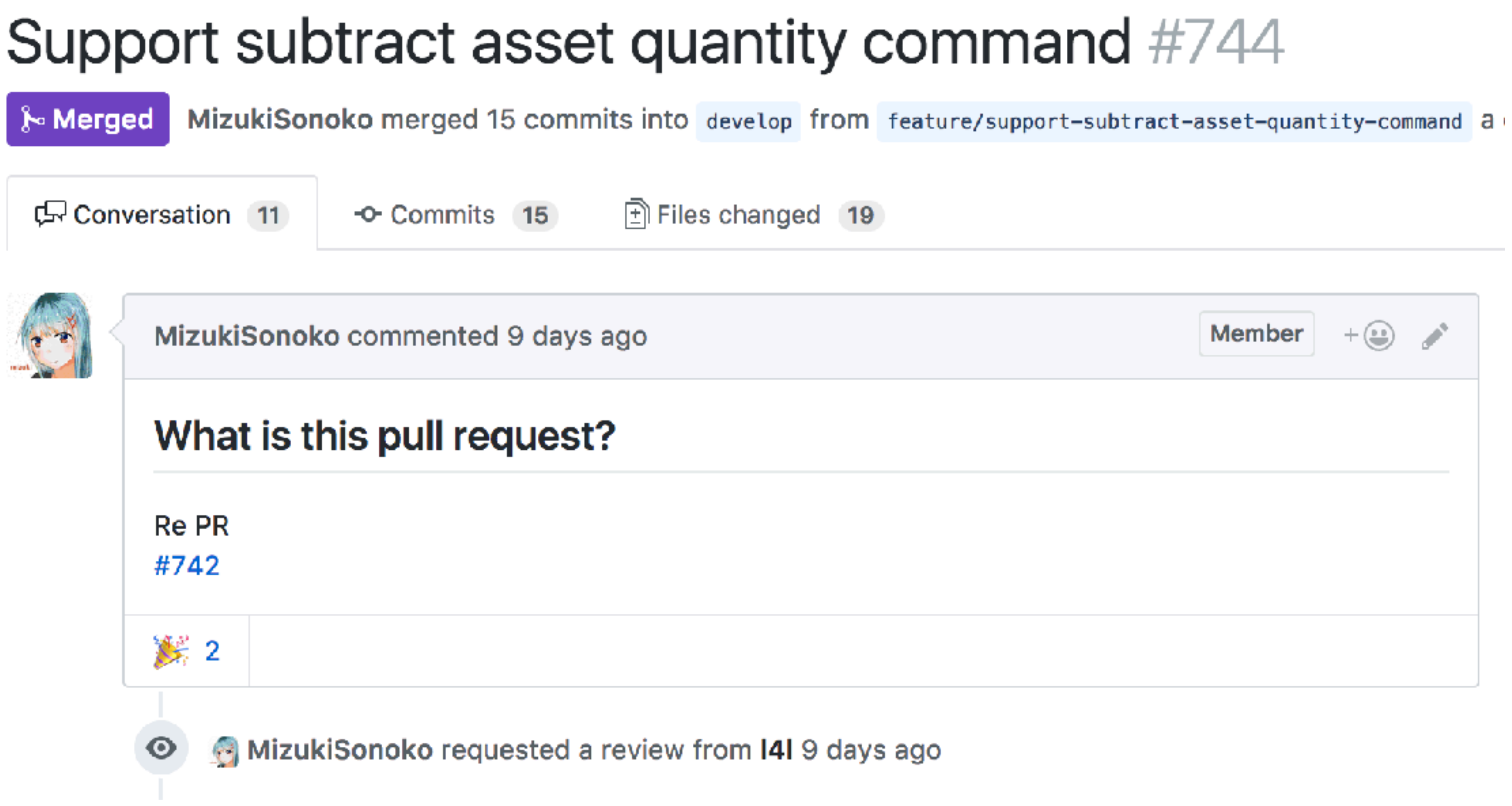
```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDJkifCjMRhIplZ6beEFn
0QIs0aiwfSz82ZDdEBdynYKzB9d27QpYeqxRsT8pHs9LUZJI57lH1YIUY9
x+xVBJ3UTrNsQb+M/qmEpxXrpSvpNf4RqmuP3L9O/baJ912PbQJtnHGHfI
wvMPDY2/MmkbYzNZJ2AHfQYgoKtIl0eZCR2gTsJtkBe56jdHjmKbRYNFqd
udeJ74cRs4JBO1IyuQMhGUS6/kEoCsZXt6x88Rm/1Q/7RIRULMY3I4otTz
e+nyleJfwkTsJtBZ/9QyjAQ7PVXRZRC8Kx[illegible]94BQa5I3EWDR4nz1febBFVJ
CSGI/wspKrj6VBhR[illegible]GJFcm[illegible]WTuD/8Jnp iroha@localhost
~
~
~
~
~
```

We now use Ansible playbook to set up peer network for arbitrary number of nodes.

Please refer to this guide:

https://soramitsu.atlassian.net/wiki/spaces/IS/pages/127008772/How+to+run+Iroha+network+with+ansible

Subtract asset quantity



Support subtract asset quantity command #744

Merged MizukiSonoko merged 15 commits into `develop` from `feature/support-subtract-asset-quantity-command` a

Conversation 11 Commits 15 Files changed 19

MizukiSonoko commented 9 days ago Member

What is this pull request?

Re PR
#742

🎉 2

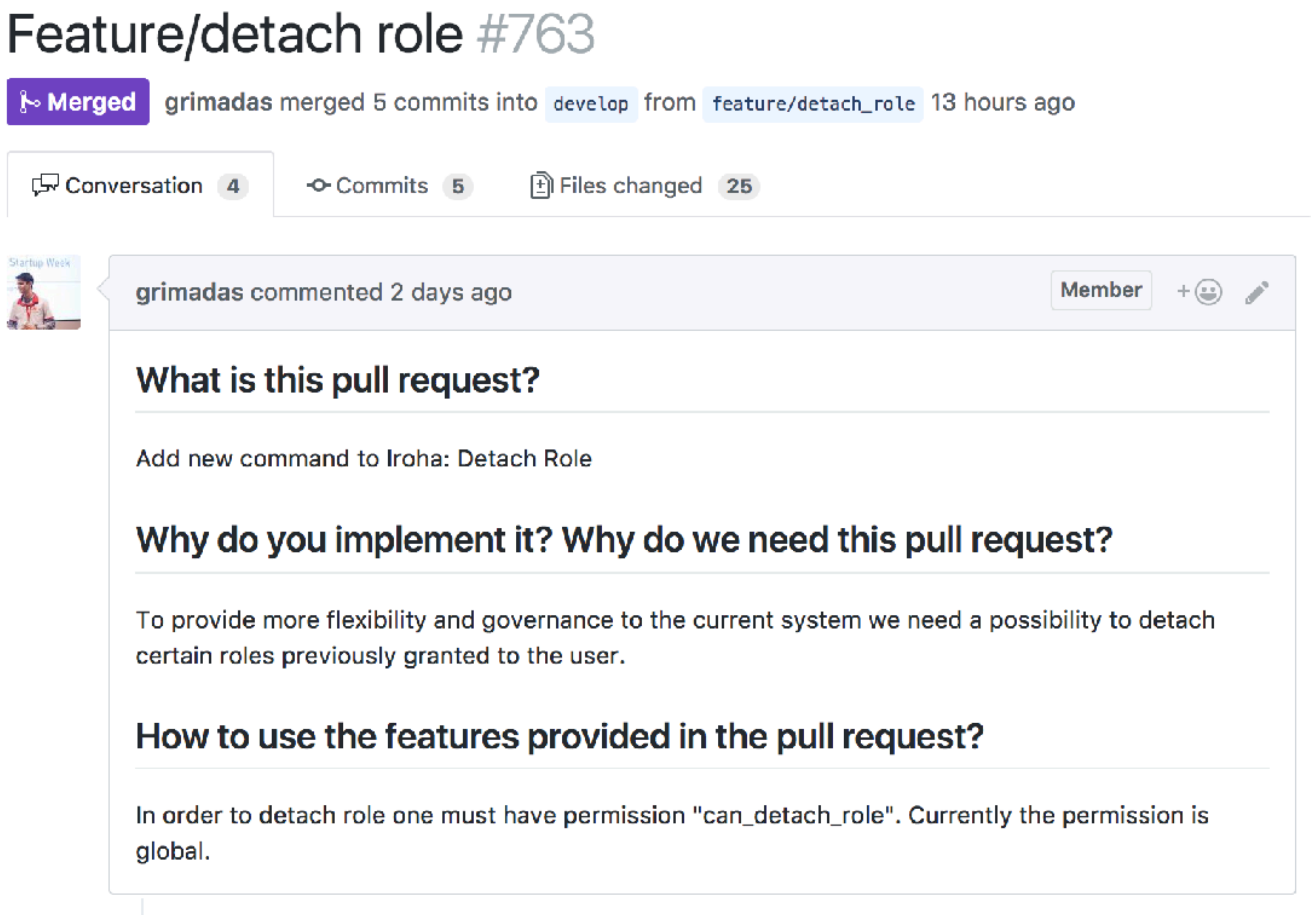
MizukiSonoko requested a review from l4l 9 days ago

Subtract asset quantity

```
message SubtractAssetQuantity {
    string account_id = 1;
    string asset_id = 2;
    Amount amount = 3;
}
```

The idea is to let asset creator user
(who has CanSubtractAssetQuantity permission)
to change number of mutable assets

Detach role command



Feature/detach role #763

Merged grimadas merged 5 commits into `develop` from `feature/detach_role` 13 hours ago

Conversation 4 Commits 5 Files changed 25

grimadas commented 2 days ago Member

What is this pull request?

Add new command to Iroha: Detach Role

Why do you implement it? Why do we need this pull request?

To provide more flexibility and governance to the current system we need a possibility to detach certain roles previously granted to the user.

How to use the features provided in the pull request?

In order to detach role one must have permission "can_detach_role". Currently the permission is global.

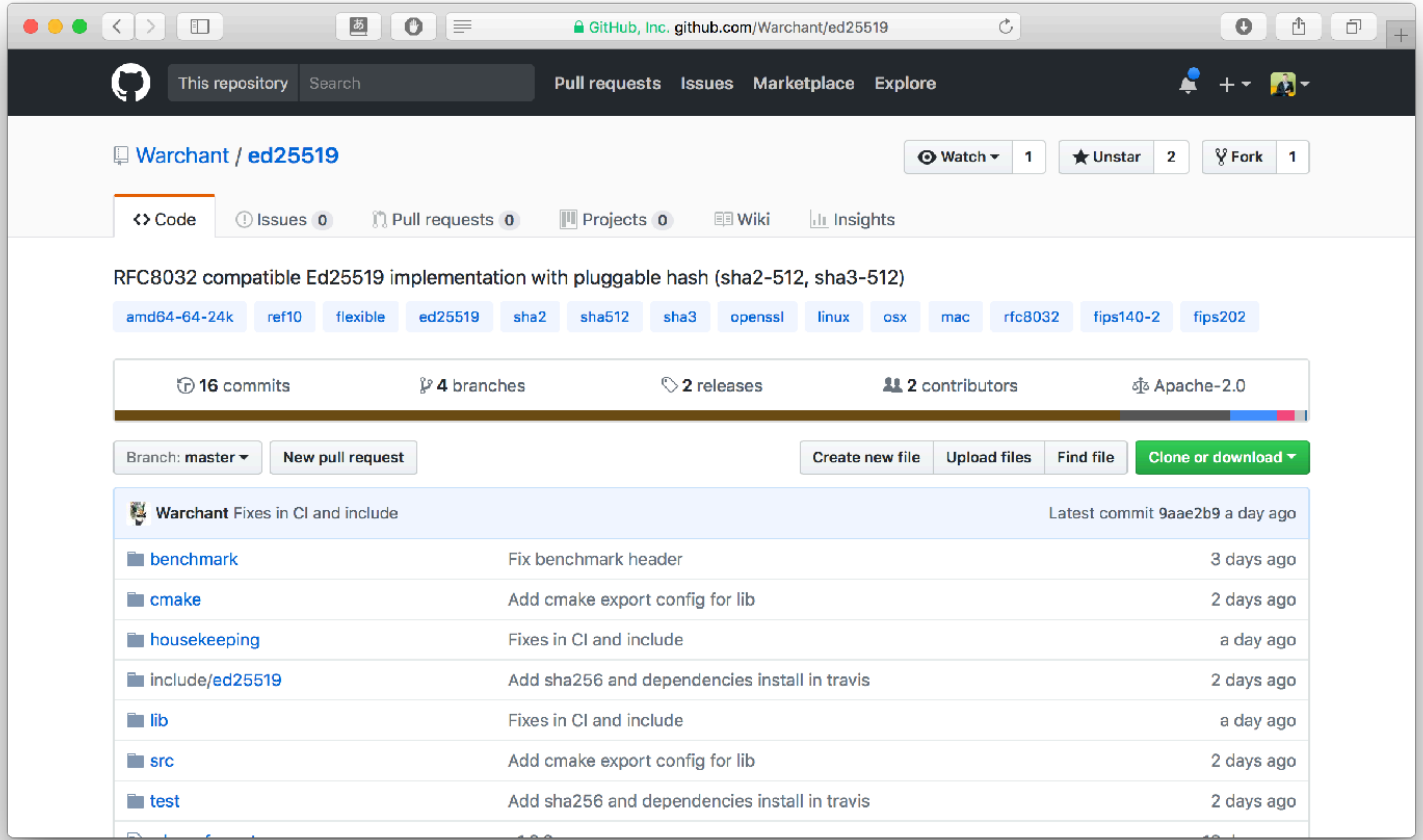
HYPERLEDGER IROHA

Detach role command

```
message DetachRole {
    string account_id = 1;
    string role_name = 2;
}
```

The idea is to manage roles of user — and detach roles of user, if we need to downgrade him/her.

ed25519 library with SHA3 hashing



RFC8032 compatible Ed25519 implementation with pluggable hash (sha2-512, sha3-512)

amd64-64-24k ref10 flexible ed25519 sha2 sha512 sha3 openssl linux osx mac rfc8032 fips140-2 fips202

16 commits · 4 branches · 2 releases · 2 contributors · Apache-2.0

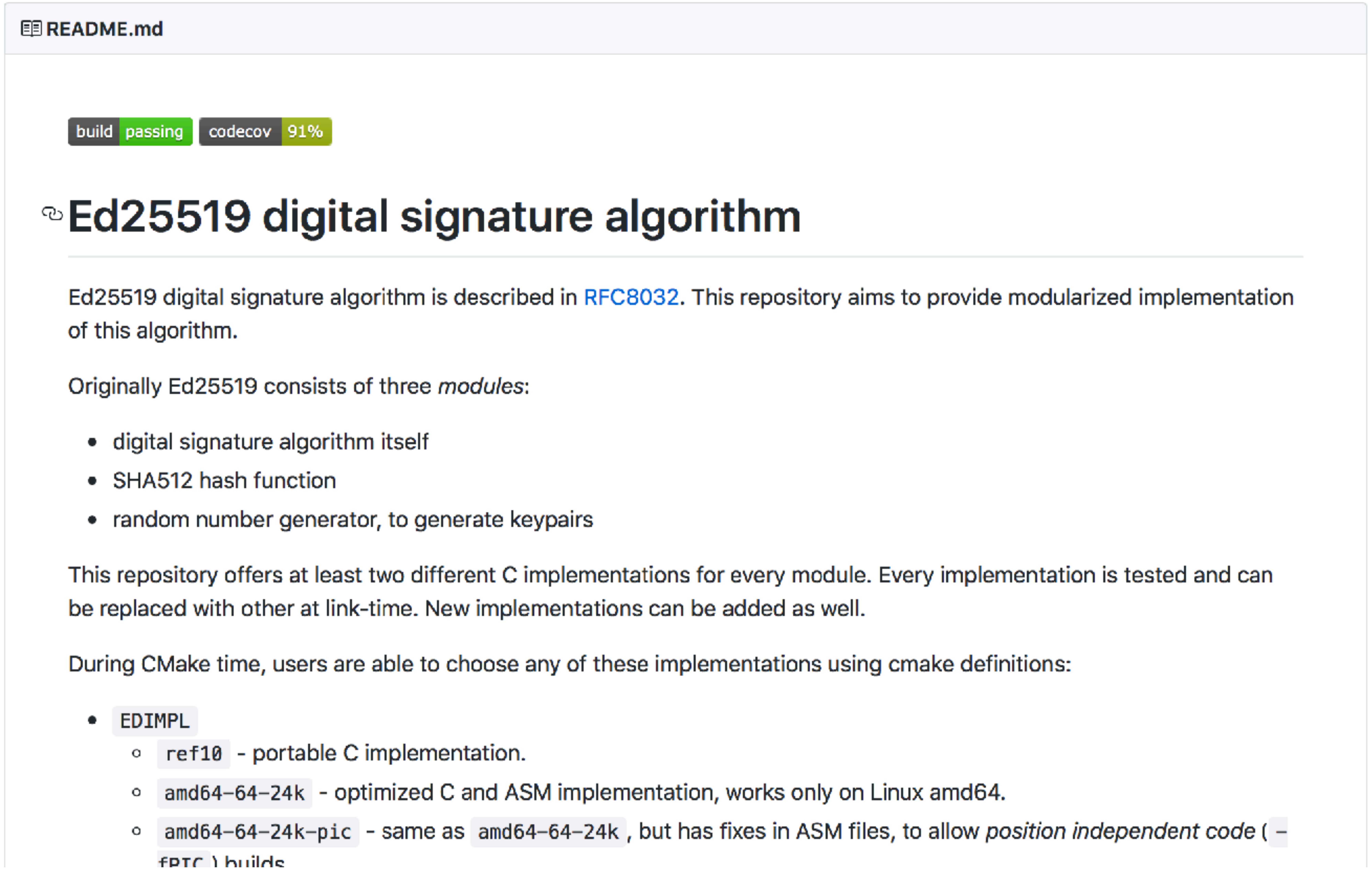
Branch: master · New pull request · Create new file · Upload files · Find file · Clone or download

Warchant Fixes in CI and include — Latest commit 9aae2b9 a day ago

benchmark	Fix benchmark header	3 days ago
cmake	Add cmake export config for lib	2 days ago
housekeeping	Fixes in CI and include	a day ago
include/ed25519	Add sha256 and dependencies install in travis	2 days ago
lib	Fixes in CI and include	a day ago
src	Add cmake export config for lib	2 days ago
test	Add sha256 and dependencies install in travis	2 days ago

HYPERLEDGER IROHA

ed25519 library with SHA3 hashing



README.md

build passing codecov 91%

Ed25519 digital signature algorithm

Ed25519 digital signature algorithm is described in RFC8032. This repository aims to provide modularized implementation of this algorithm.

Originally Ed25519 consists of three *modules*:

- digital signature algorithm itself
- SHA512 hash function
- random number generator, to generate keypairs

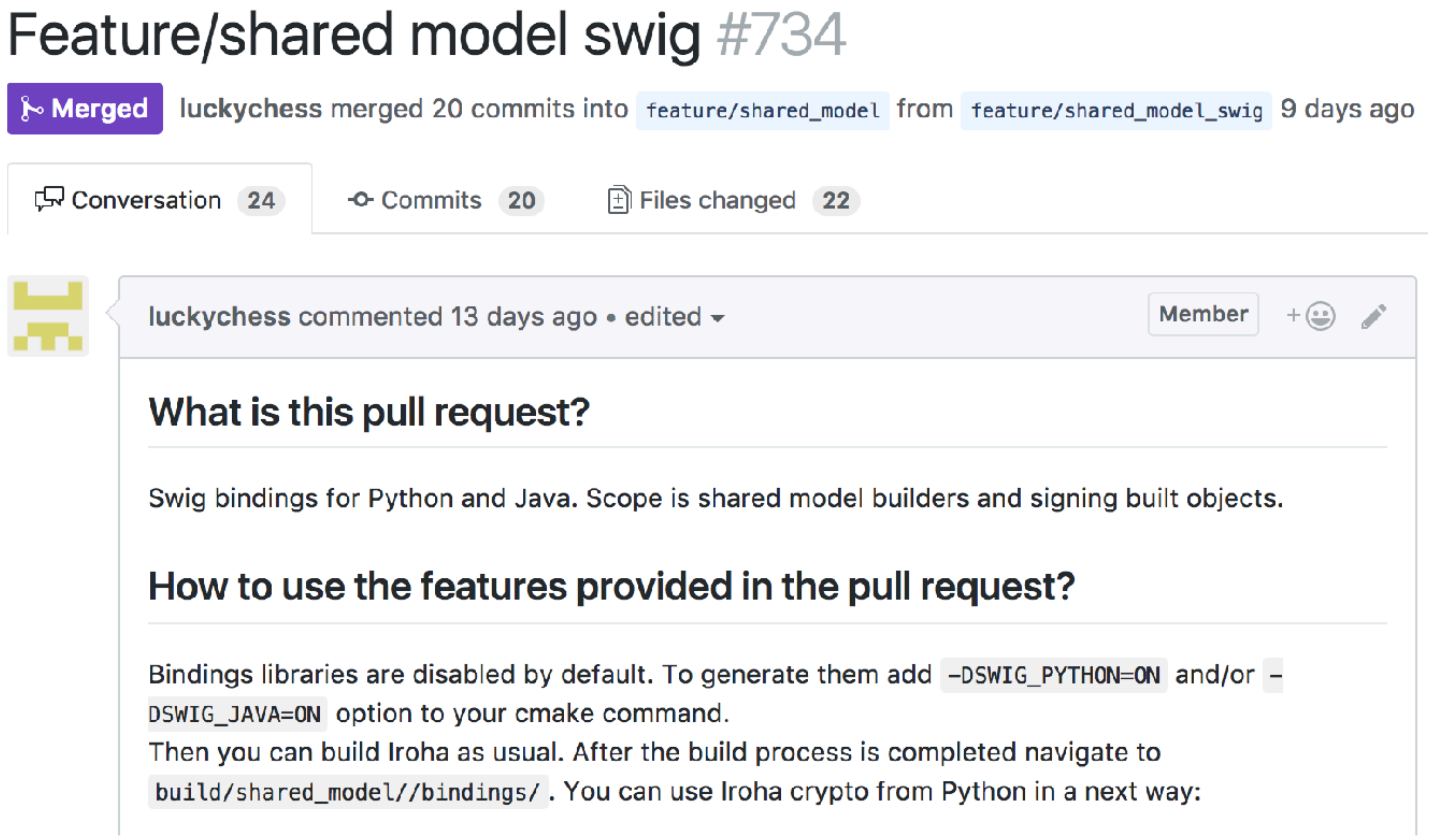
This repository offers at least two different C implementations for every module. Every implementation is tested and can be replaced with other at link-time. New implementations can be added as well.

During CMake time, users are able to choose any of these implementations using cmake definitions:

- `EDIMPL`
  - `ref10` - portable C implementation.
  - `amd64-64-24k` - optimized C and ASM implementation, works only on Linux amd64.
  - `amd64-64-24k-pic` - same as `amd64-64-24k`, but has fixes in ASM files, to allow *position independent code* (`-fPIC`) builds

HYPERLEDGER IROHA

SWIG bindings for transaction generation



Feature/shared model swig #734

Merged luckychess merged 20 commits into `feature/shared_model` from `feature/shared_model_swig` 9 days ago

Conversation 24 Commits 20 Files changed 22

luckychess commented 13 days ago • edited Member

What is this pull request?

Swig bindings for Python and Java. Scope is shared model builders and signing built objects.

How to use the features provided in the pull request?

Bindings libraries are disabled by default. To generate them add `-DSWIG_PYTHON=ON` and/or `-DSWIG_JAVA=ON` option to your cmake command.
Then you can build Iroha as usual. After the build process is completed navigate to `build/shared_model//bindings/`. You can use Iroha crypto from Python in a next way:

HYPERLEDGER IROHA

SWIG bindings for transaction generation

```
import iroha

builder = iroha.ModelBuilder()
crypto = iroha.ModelCrypto()

me_kp = crypto.generateKeypair()
peer_kp = crypto.generateKeypair()
signatory_kp = crypto.generateKeypair()
account_kp = crypto.generateKeypair()
time = 1512549580
startCounter = 1
creator = "me"
signatory = "fyodorkek-san"

commands = []

commands.append(builder.txCounter(startCounter).creatorAccountId(creator)
.createdTime(time).addPeer("127.0.0.1:50051", peer_kp.publicKey()).build())

commands.append(builder.txCounter(startCounter+1).creatorAccountId(creator)
.createdTime(time+1).createDomain("iroha", "admin").build())

commands.append(builder.txCounter(startCounter+2).creatorAccountId(creator)
.createdTime(time+2).createAccount("luckychess", "iroha",
account_kp.publicKey()).build())
```

SWIG bindings for transaction generation

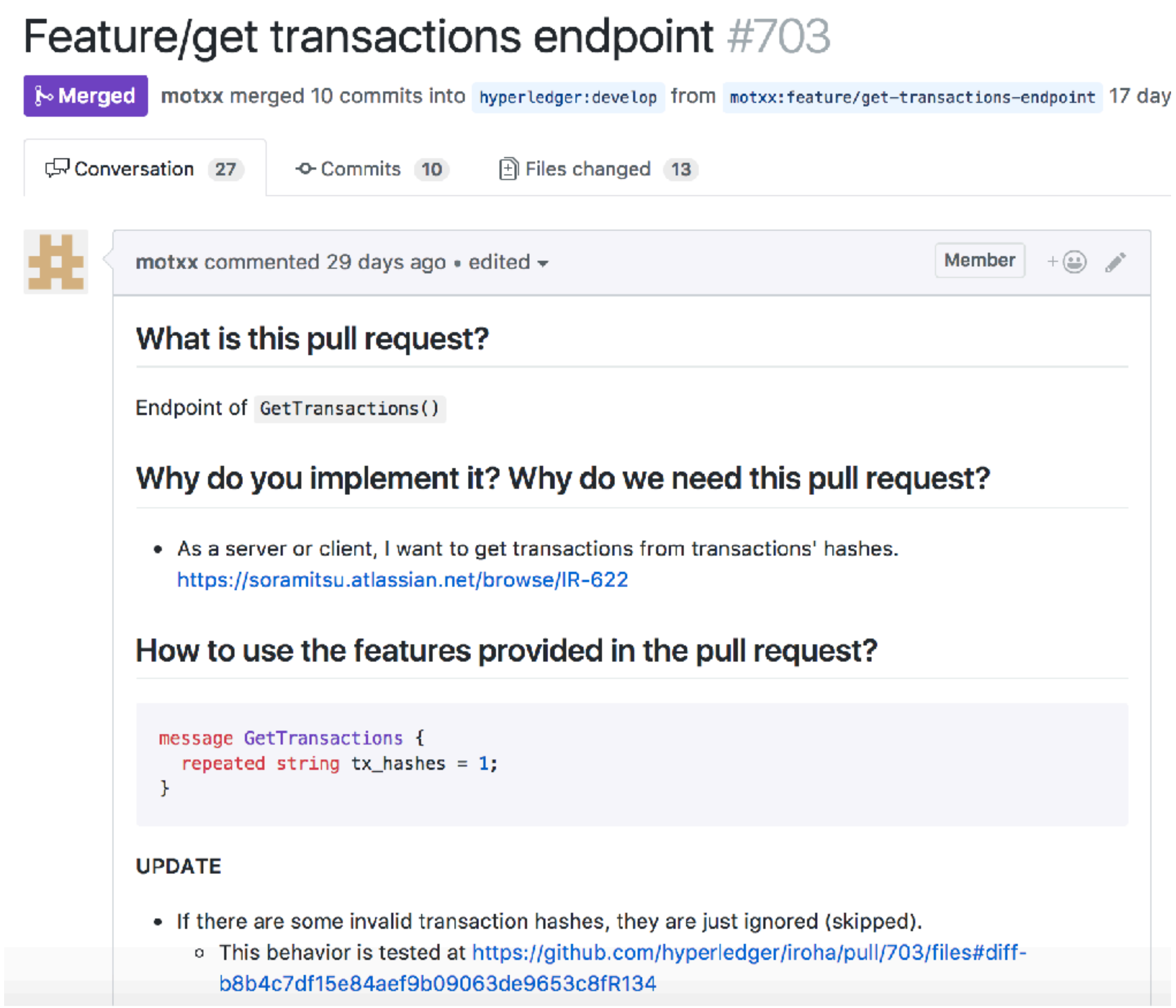
```
import java.math.BigInteger;
import java.util.*;

public class my {
  static {
    try {
        System.loadLibrary("iroha");
    } catch (UnsatisfiedLinkError e) {
      System.err.println("Native code library failed to load. \n" + e);
      System.exit(1);
    }
  }

  public static void main(String argv[]) {
    ModelBuilder builder = new ModelBuilder();
    ModelCrypto crypto = new ModelCrypto();
    Keypair me_kp = crypto.generateKeypair();
    Keypair peer_kp = crypto.generateKeypair();
    Keypair signatory_kp = crypto.generateKeypair();
    Keypair account_kp = crypto.generateKeypair();
    long time = 1512549580;
    long startCounter = 1;
    String creator = "me";
    String signatory = "fyodorkek-san";

    ArrayList<UnsignedTx> commands = new ArrayList<UnsignedTx>();
    commands.add(builder.txCounter(BigInteger.valueOf(startCounter)).creatorAccountId(creator)
.createdTime(BigInteger.valueOf(time)).addPeer("127.0.0.1:50051", peer_kp.publicKey()).build());
```

GetTransaction API



Feature/get transactions endpoint #703

Merged motxx merged 10 commits into `hyperledger:develop` from `motxx:feature/get-transactions-endpoint` 17 day

Conversation 27 Commits 10 Files changed 13

motxx commented 29 days ago • edited Member

What is this pull request?

Endpoint of `GetTransactions()`

Why do you implement it? Why do we need this pull request?

- As a server or client, I want to get transactions from transactions' hashes. https://soramitsu.atlassian.net/browse/IR-622

How to use the features provided in the pull request?

```
message GetTransactions {
  repeated string tx_hashes = 1;
}
```

UPDATE

- If there are some invalid transaction hashes, they are just ignored (skipped).
  - This behavior is tested at https://github.com/hyperledger/iroha/pull/703/files#diff-b8b4c7df15e84aef9b09063de9653c8fR134

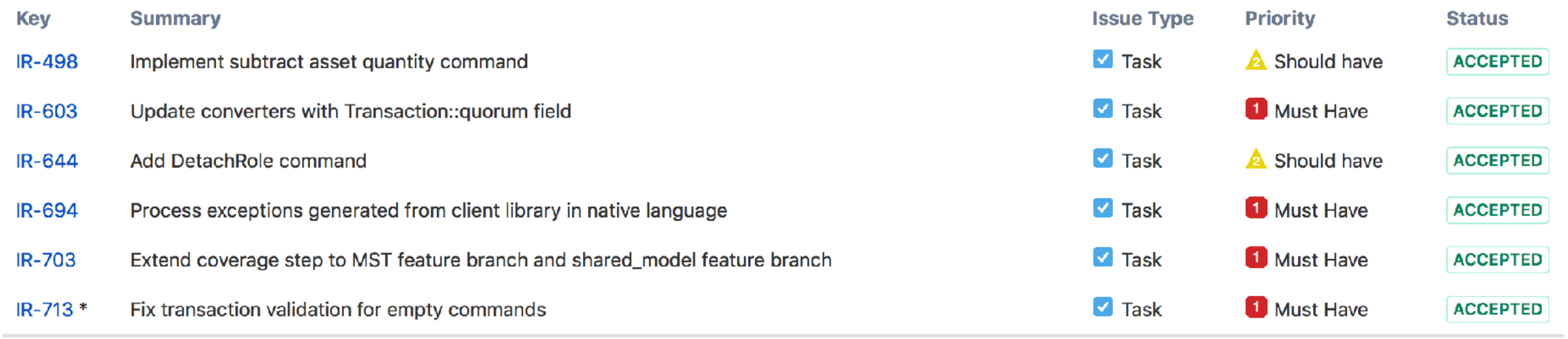
HYPERLEDGER IROHA

GetTransaction API

```
message GetTransactions {
  repeated string tx_hashes = 1;
}
```

Intention for the query is to get transaction contents, based on transaction hash for middleware, which stores only tx hash.

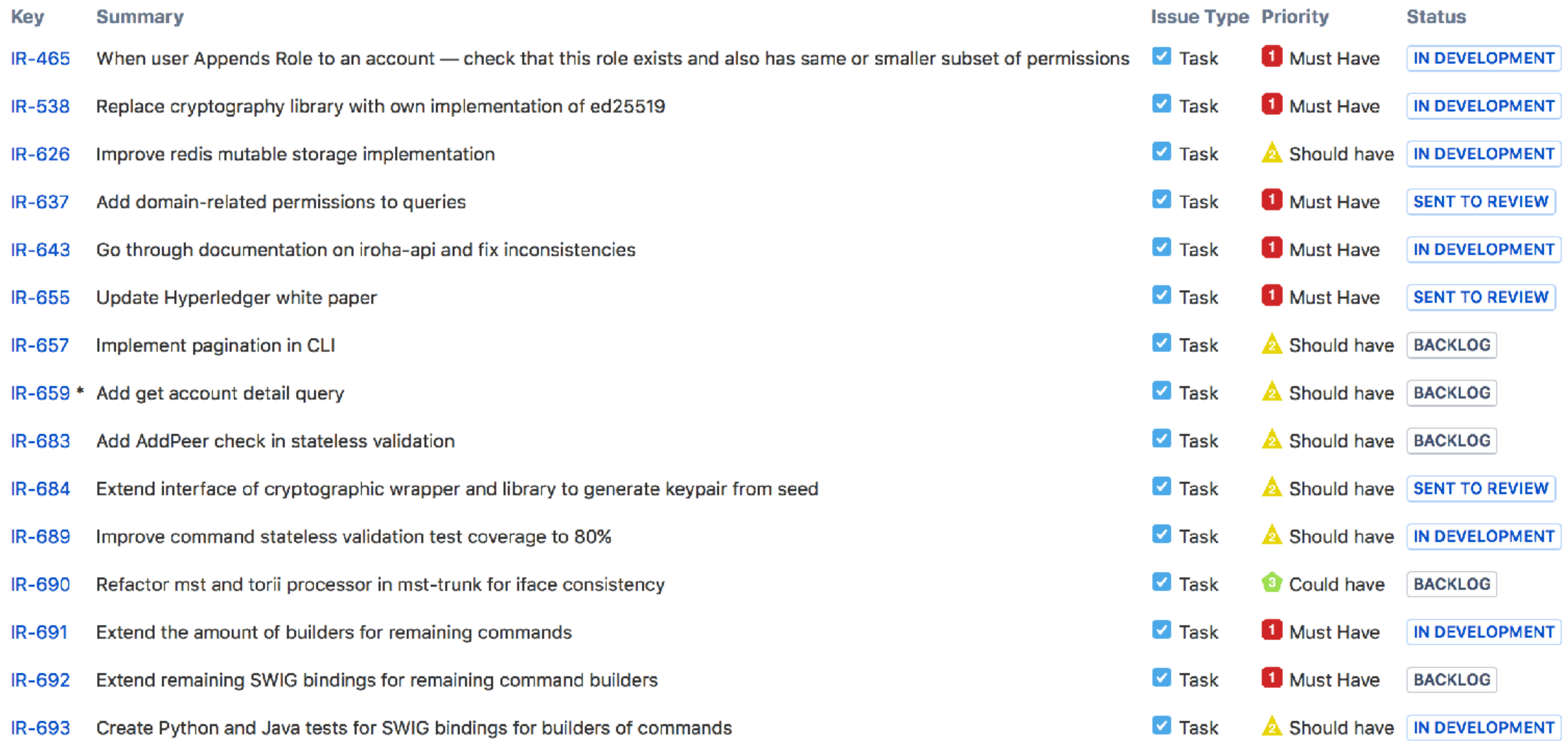
Current iteration



Key	Summary	Issue Type	Priority	Status
IR-498	Implement subtract asset quantity command	Task	Should have	ACCEPTED
IR-603	Update converters with Transaction::quorum field	Task	Must Have	ACCEPTED
IR-644	Add DetachRole command	Task	Should have	ACCEPTED
IR-694	Process exceptions generated from client library in native language	Task	Must Have	ACCEPTED
IR-703	Extend coverage step to MST feature branch and shared_model feature branch	Task	Must Have	ACCEPTED
IR-713 *	Fix transaction validation for empty commands	Task	Must Have	ACCEPTED

HYPERLEDGER IROHA

Current iteration



Key	Summary	Issue Type	Priority	Status
IR-465	When user Appends Role to an account — check that this role exists and also has same or smaller subset of permissions	Task	Must Have	IN DEVELOPMENT
IR-538	Replace cryptography library with own implementation of ed25519	Task	Must Have	IN DEVELOPMENT
IR-626	Improve redis mutable storage implementation	Task	Should have	IN DEVELOPMENT
IR-637	Add domain-related permissions to queries	Task	Must Have	SENT TO REVIEW
IR-643	Go through documentation on iroha-api and fix inconsistencies	Task	Must Have	IN DEVELOPMENT
IR-655	Update Hyperledger white paper	Task	Must Have	SENT TO REVIEW
IR-657	Implement pagination in CLI	Task	Should have	BACKLOG
IR-659 *	Add get account detail query	Task	Should have	BACKLOG
IR-683	Add AddPeer check in stateless validation	Task	Should have	BACKLOG
IR-684	Extend interface of cryptographic wrapper and library to generate keypair from seed	Task	Should have	SENT TO REVIEW
IR-689	Improve command stateless validation test coverage to 80%	Task	Should have	IN DEVELOPMENT
IR-690	Refactor mst and torii processor in mst-trunk for iface consistency	Task	Could have	BACKLOG
IR-691	Extend the amount of builders for remaining commands	Task	Must Have	IN DEVELOPMENT
IR-692	Extend remaining SWIG bindings for remaining command builders	Task	Must Have	BACKLOG
IR-693	Create Python and Java tests for SWIG bindings for builders of commands	Task	Should have	IN DEVELOPMENT

HYPERLEDGER IROHA

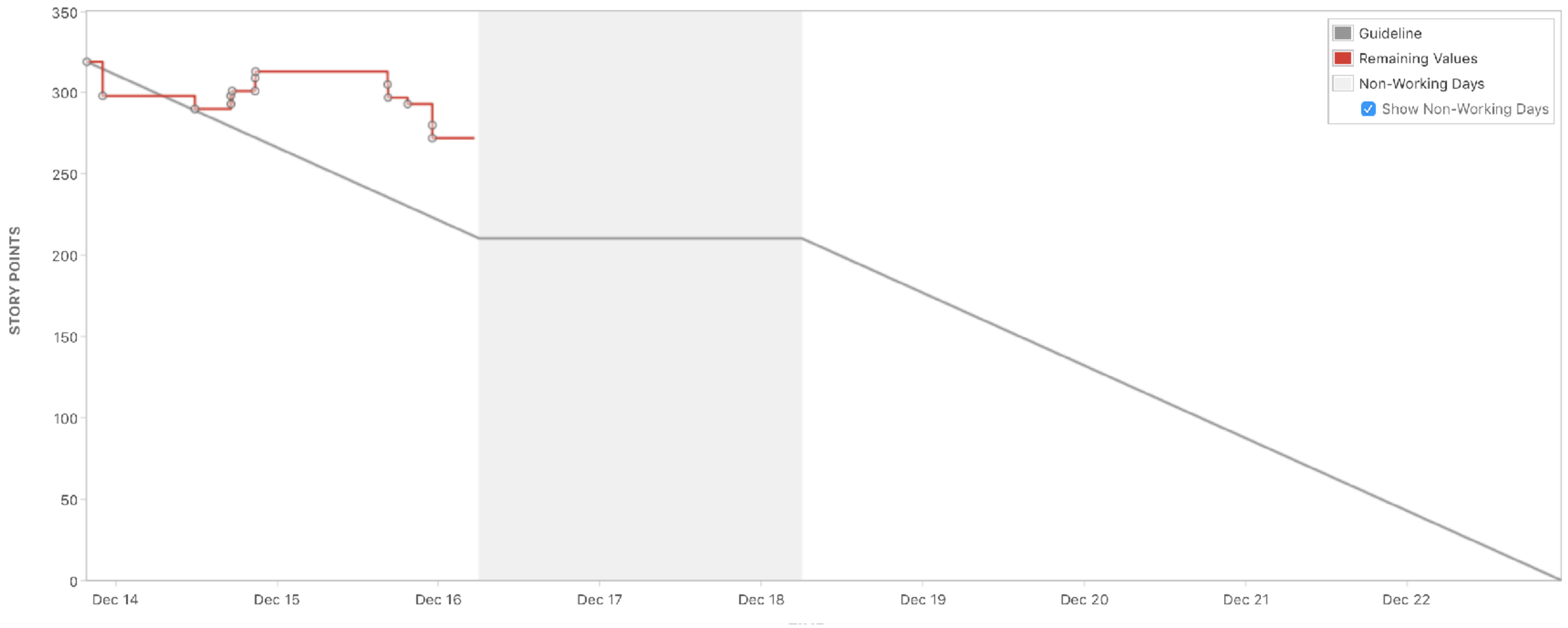
Current iteration

IR-697	Implement stateless validation for query builders	Task	Should have	BACKLOG
IR-698	Create SWIG bindings for Query builders	Task	Should have	BACKLOG
IR-699	Write tests for SWIG-generated code for Query builders	Task	Should have	BACKLOG
IR-700	Use copy of protobuf binary instead of modifying existing binary instance in builders	Task	Could have	SENT TO REVIEW
IR-704	Correct external coverage analysis tool to show coverage of all feature branches	Task	Should have	IN DEVELOPMENT
IR-705	Extend interface of cryptographic wrapper to use existing keypair for signing	Task	Must Have	SENT TO REVIEW
IR-706	Test client library compatibility with Swift code	Task	Should have	BACKLOG
IR-707	Java example code for transaction generation, query generation and transaction status check	Task	Must Have	BACKLOG
IR-708	Python example code for transaction generation, query generation and transaction status check	Task	Must Have	BACKLOG
IR-709	Create artifacts and release notes for alpha release of Iroha	Task	Should have	BACKLOG
IR-712 *	Implement integration test for MST	Task	Must Have	BACKLOG

HYPERLEDGER IROHA

Current iteration

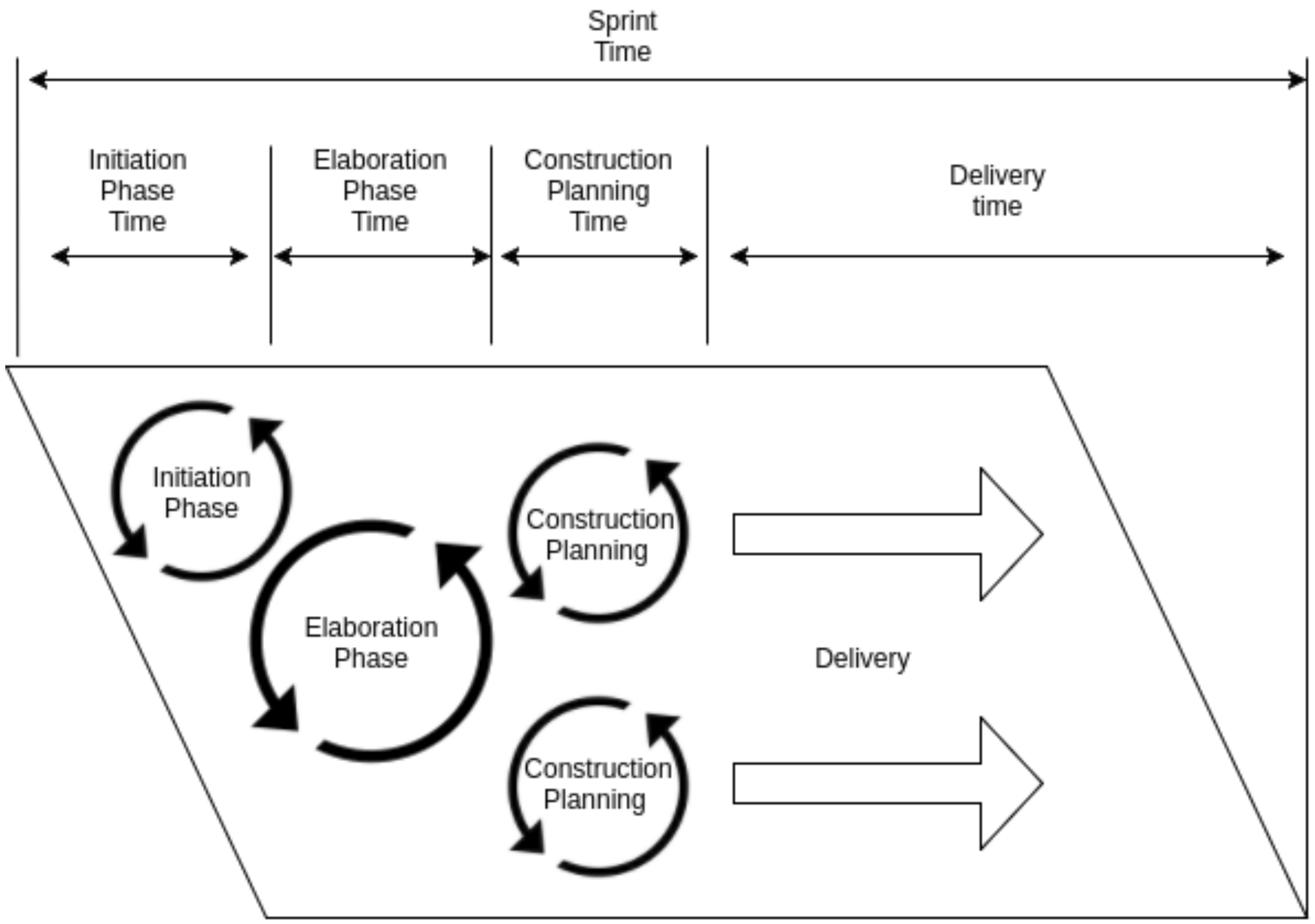
To produce ready-to-utilize client library for Iroha API and release alpha version of Iroha



HYPERLEDGER IROHA

Process and tools used

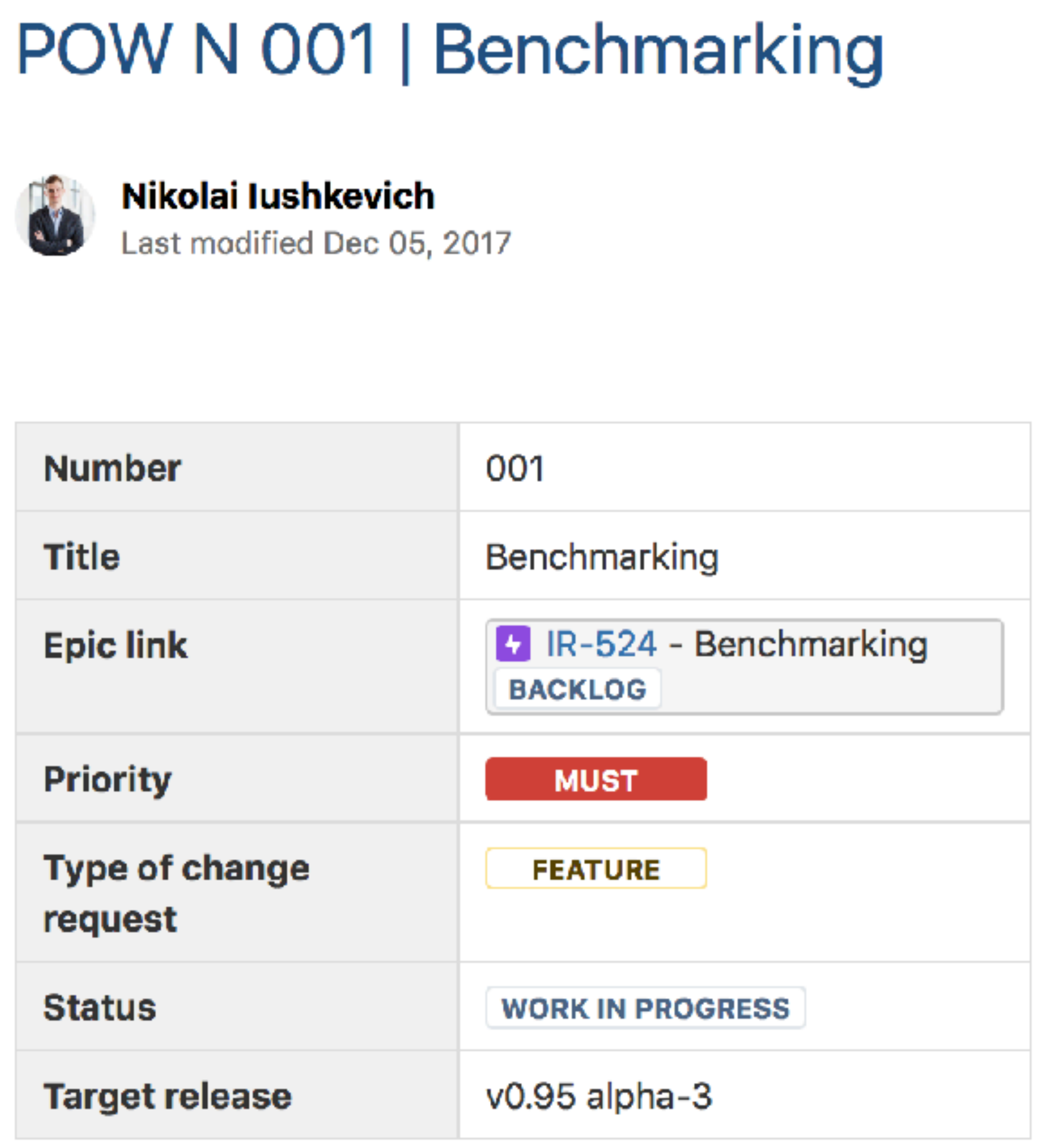
Iroha project Software Development Process | Iteration cycle



HYPERLEDGER IROHA

Process and tools used

Piece of Work



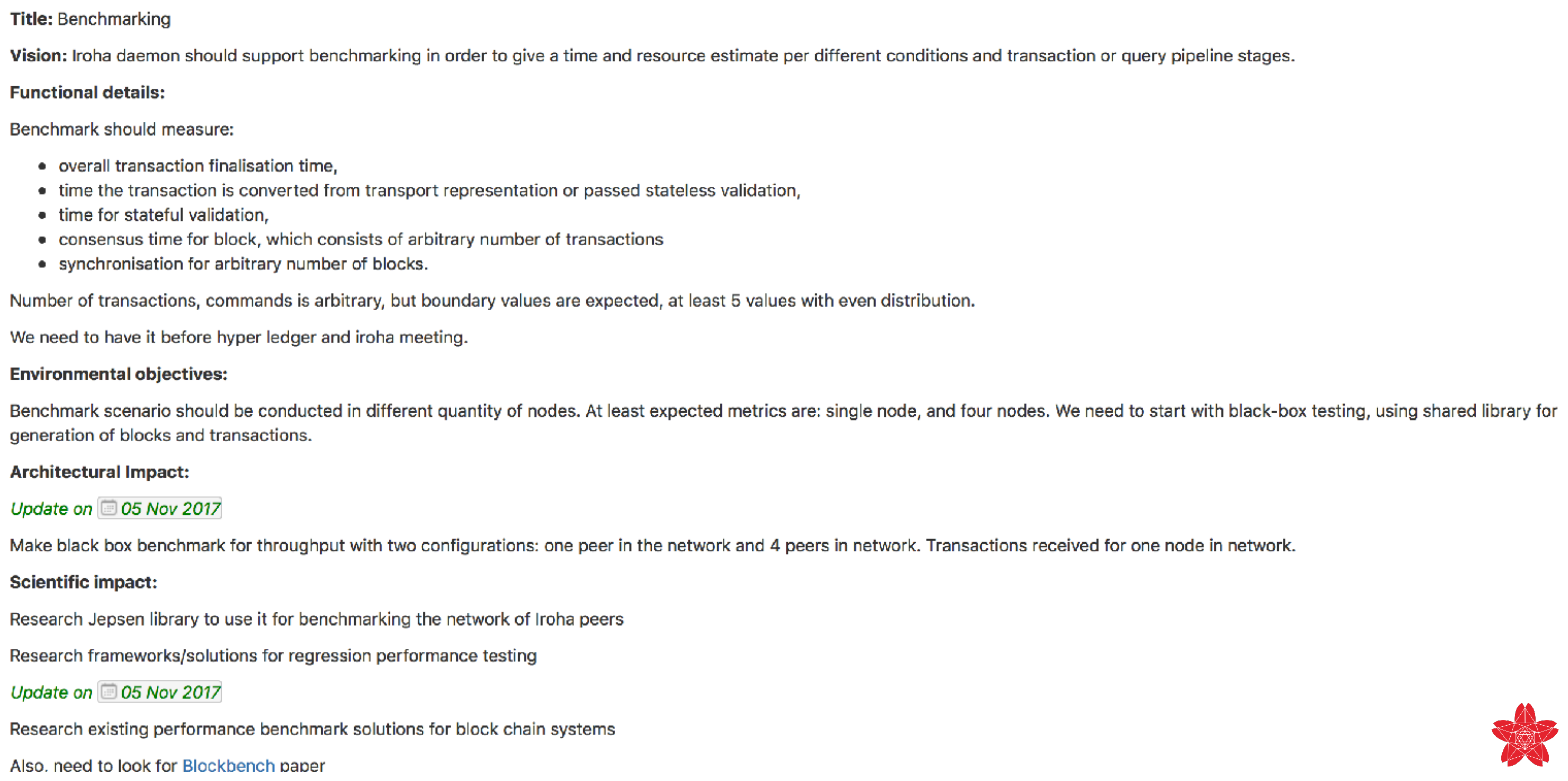
POW N 001 | Benchmarking

Nikolai Iushkevich
Last modified Dec 05, 2017

Number	001
Title	Benchmarking
Epic link	IR-524 - Benchmarking BACKLOG
Priority	MUST
Type of change request	FEATURE
Status	WORK IN PROGRESS
Target release	v0.95 alpha-3

Process and tools used

Piece of Work



Title: Benchmarking

Vision: Iroha daemon should support benchmarking in order to give a time and resource estimate per different conditions and transaction or query pipeline stages.

Functional details:

Benchmark should measure:

- overall transaction finalisation time,
- time the transaction is converted from transport representation or passed stateless validation,
- time for stateful validation,
- consensus time for block, which consists of arbitrary number of transactions
- synchronisation for arbitrary number of blocks.

Number of transactions, commands is arbitrary, but boundary values are expected, at least 5 values with even distribution.

We need to have it before hyper ledger and iroha meeting.

Environmental objectives:

Benchmark scenario should be conducted in different quantity of nodes. At least expected metrics are: single node, and four nodes. We need to start with black-box testing, using shared library for generation of blocks and transactions.

Architectural Impact:

Update on 05 Nov 2017

Make black box benchmark for throughput with two configurations: one peer in the network and 4 peers in network. Transactions received for one node in network.

Scientific impact:

Research Jepsen library to use it for benchmarking the network of Iroha peers

Research frameworks/solutions for regression performance testing

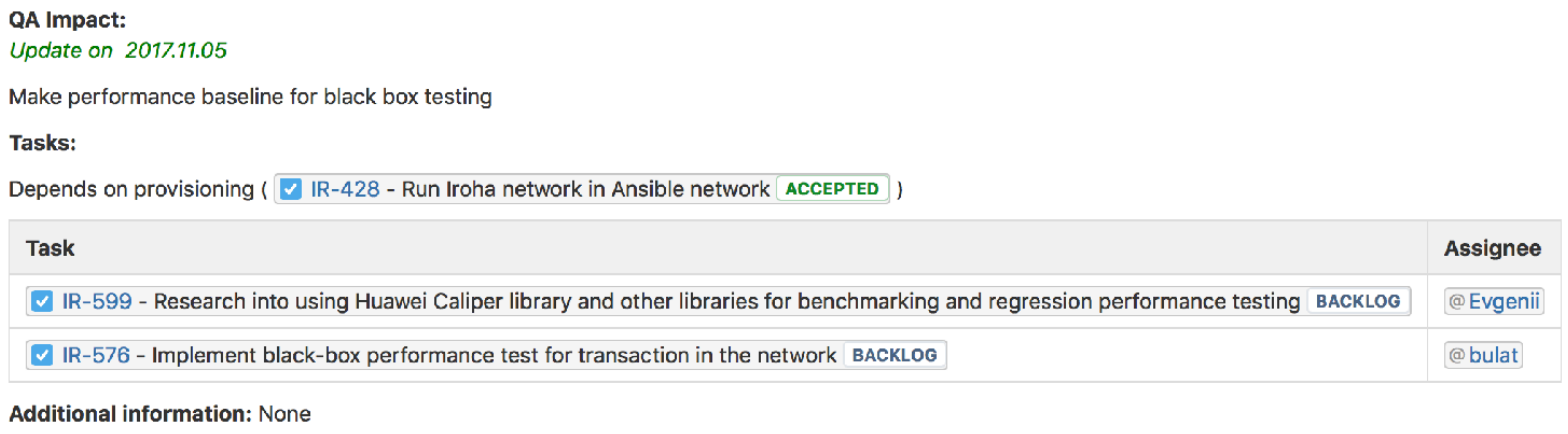
Update on 05 Nov 2017

Research existing performance benchmark solutions for block chain systems

Also, need to look for Blockbench paper

Process and tools used

Piece of Work



QA Impact:

Update on 2017.11.05

Make performance baseline for black box testing

Tasks:

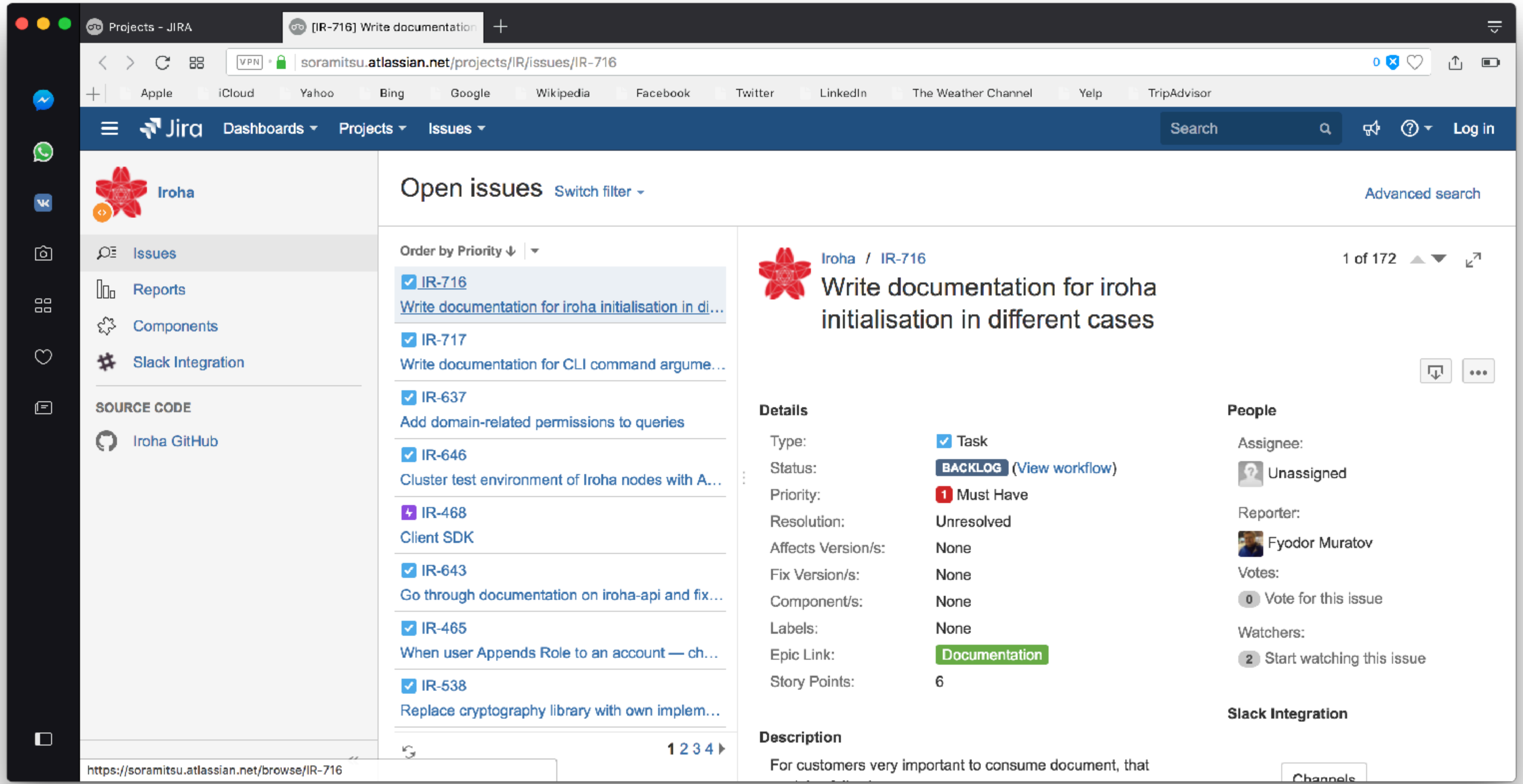
Depends on provisioning (✓ IR-428 - Run Iroha network in Ansible network ACCEPTED)

Task	Assignee
✓ IR-599 - Research into using Huawei Caliper library and other libraries for benchmarking and regression performance testing BACKLOG	@Evgenii
✓ IR-576 - Implement black-box performance test for transaction in the network BACKLOG	@bulat

Additional information: None

HYPERLEDGER IROHA

Process and tools used



Projects - JIRA | [IR-716] Write documentation

soramitsu.atlassian.net/projects/IR/issues/IR-716

Jira | Dashboards | Projects | Issues | Search | Log in

Iroha

- Issues
- Reports
- Components
- Slack Integration

SOURCE CODE

- Iroha GitHub

Open issues Switch filter   Advanced search

Order by Priority

- IR-716 Write documentation for iroha initialisation in di…
- IR-717 Write documentation for CLI command argume…
- IR-637 Add domain-related permissions to queries
- IR-646 Cluster test environment of Iroha nodes with A…
- IR-468 Client SDK
- IR-643 Go through documentation on iroha-api and fix…
- IR-465 When user Appends Role to an account — ch…
- IR-538 Replace cryptography library with own implem…

1 2 3 4

Iroha / IR-716   1 of 172

Write documentation for iroha initialisation in different cases

Details

Type:	Task
Status:	BACKLOG (View workflow)
Priority:	Must Have
Resolution:	Unresolved
Affects Version/s:	None
Fix Version/s:	None
Component/s:	None
Labels:	None
Epic Link:	Documentation
Story Points:	6

People

Assignee: Unassigned

Reporter: Fyodor Muratov

Votes: 0 Vote for this issue

Watchers: 2 Start watching this issue

Slack Integration

Channels

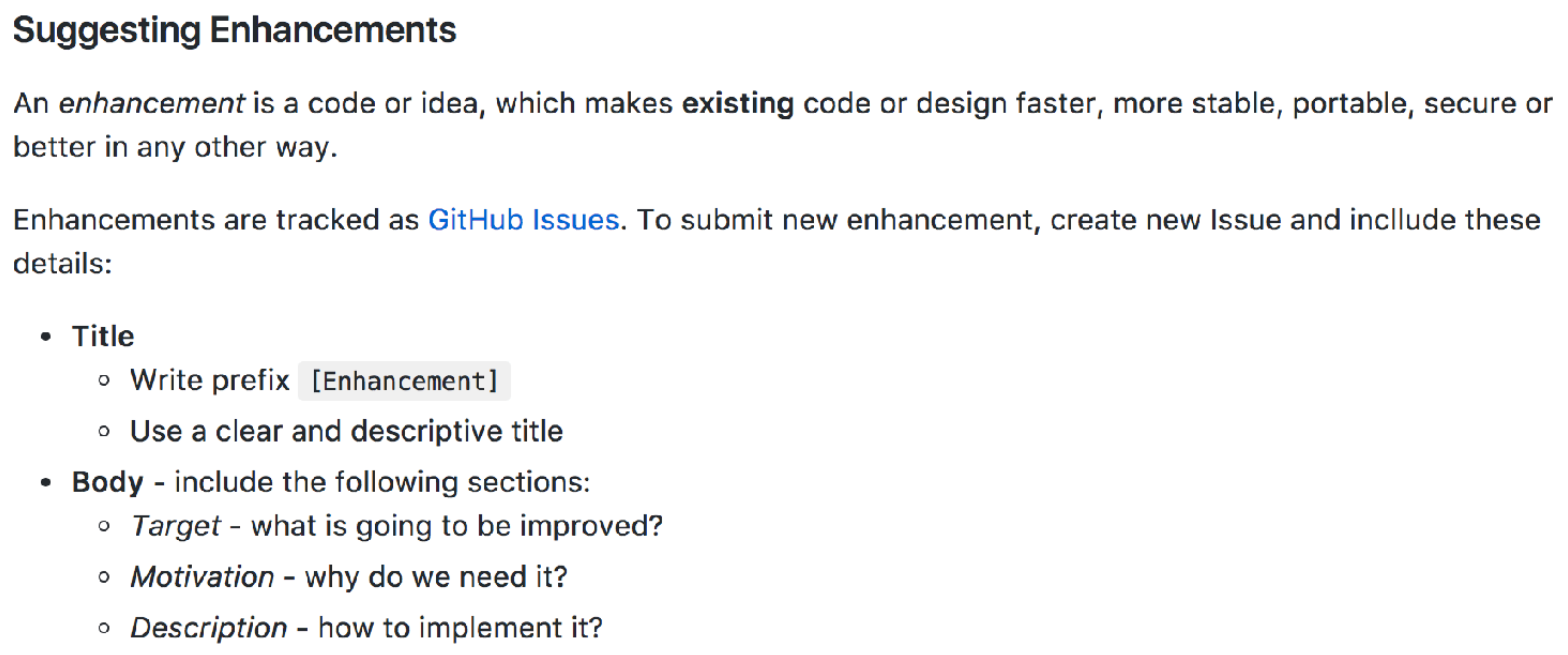
Description

For customers very important to consume document, that

https://soramitsu.atlassian.net/browse/IR-716

HYPERLEDGER IROHA

How to suggest improvements



Suggesting Enhancements

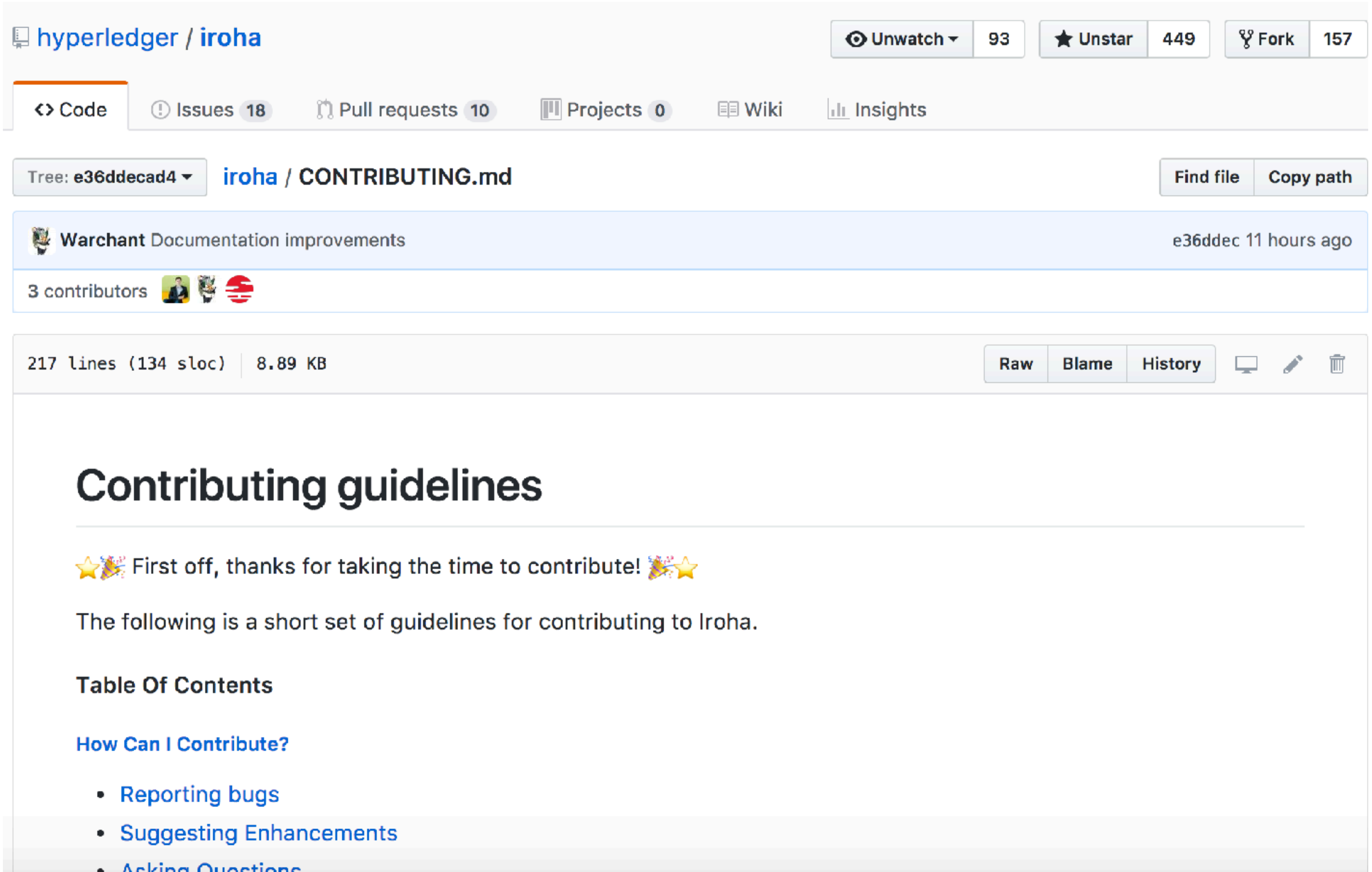
An *enhancement* is a code or idea, which makes **existing** code or design faster, more stable, portable, secure or better in any other way.

Enhancements are tracked as GitHub Issues. To submit new enhancement, create new Issue and incllude these details:

- **Title**
  - Write prefix `[Enhancement]`
  - Use a clear and descriptive title
- **Body** - include the following sections:
  - *Target* - what is going to be improved?
  - *Motivation* - why do we need it?
  - *Description* - how to implement it?

HYPERLEDGER IROHA

How to suggest improvements



Contributing guidelines

⭐🎉 First off, thanks for taking the time to contribute! 🎉⭐

The following is a short set of guidelines for contributing to Iroha.

Table Of Contents

HYPERLEDGER IROHA

Upcoming α release

泉

izumi

HYPERLEDGER IROHA

Upcoming α release

Iroha is ready for **KYC** features (key-value storage of account). ✓

Cryptography library (ed25519) is checked against RFC specification and is using SHA3 hashing. ✓

It is available to be reused by client applications on **Java, Python.** ✓

Client library for Iroha contains following API available in the system:
forming transactions,
queries, ✓
getting transaction status
and is available for **Java, Python.**

Permission model is improved with domain-specific permissions
and detachment of user's role. ✓

Asset naming is checked against regular expressions. ✓
Documentation in API website is consistent with codebase.

Hyperledger α definition

Feature complete, for all features committed to the production release.
Ready for Proof of Concept-level deployments.
Performance can be characterized in a predictable way, so that basic PoC's can be done within the bounds of published expectations.
APIs are documented. First attempts at end-user documentation have been made. Developer documentation is further advanced.
No “highest priority” issues are in an open state.

HYPERLEDGER IROHA

Q&A