
fabric-peer-ext Documentation

Release 0.0.1

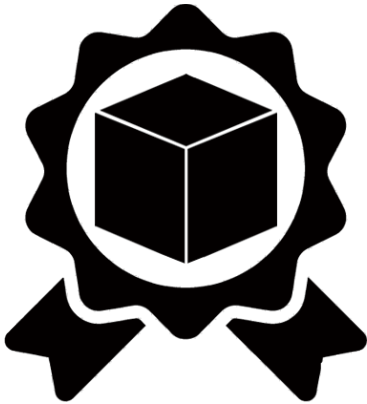
2019, SecureKey Technologies Inc.

Jan 11, 2021

Contents

1	Hyperledger Fabric Peer Ext library	3
2	Have Questions?	7

Libraries that extends the core peer capabilities; and builds an extended fabric-peer image.



Hyperledger Fabric Peer Ext library

Hyperledger Fabric Peer Extension Library implements fabric-mod interface to extend the core peer capabilities. Browse through the implementation of enhancements here :

1.1 Block Storage Enhancements

Block Storage was originally implemented with a FileSystem storage using indexes stored in LevelDB. This filesystem storage is not scalable and requires a separate storage for the indexes. The new storage used in this mod project is CouchDB.

1.1.1 CouchDB

There are several benefits for using CouchDB storage for blocks, among these are:

- The use of a distributed and scalable storage between peers.
- Indexes are managed in the same storage.

<https://www.apache.org/licenses/LICENSE-2.0>

1.2 Private Data Collections

1.2.1 Transient Data Collection

A transient data collection is a **private data collection** that holds temporary data off of the ledger (i.e. it is stored at endorsement time) and therefore does not require a commit.

Transient data should have an *expiration based on time* (as opposed to block height) and the expiration should be assumed to be (typically) short, in which case in-memory store is suitable. In order to prevent too much data from being stored in memory, the data should be off-loaded to a database using an LRU policy. In order to make storing

transient data efficient, transient data should not be stored on every peer but should be distributed to a small subset of peers (calculated using the collection config) according to a deterministic algorithm.

When a client requests the data, then the same algorithm may be used to find the peer in which it was stored. The existing chaincode stub functions, `PutPrivateData` and `GetPrivateData`, may be used by chaincode to put and get transient data. Reads and writes of transient data must not show up in the read/write set if the transaction is committed to the ledger. An API should also be provided in order to read/write transient data from outside of a chaincode invocation.

1.2.2 Off-Ledger and DCAS Collections

An Off-Ledger collection is a **private data collection** that stores data at endorsement time, i.e. the transaction does not need to be committed. Purging of Off-Ledger data should be based on a time-to-live policy as opposed to a block-to-live policy. Reads and writes of Off-Ledger/DCAS data must not show up in the read/write set if the transaction is committed to the ledger. Aside from the above requirements, Off-Ledger private collections behave the same way as standard private data collections.

A **DCAS** (Distributed Content Addressable Store) collection is a subset of the Off-Ledger collection with the restriction that the key **MUST** be the hash of the value. The existing chaincode stub functions, `PutPrivateData` and `GetPrivateData`, may be used by chaincode to put and get Off-Ledger/DCAS data. For **DCAS collections**, the key may be empty (""), since it is always the hash of the value. If the key is provided, then it should be validated against the value. APIs should also be provided for both Off-Ledger and DCAS collections in order to read/write data from outside of a chaincode invocation.

<https://www.apache.org/licenses/LICENSE-2.0>

1.3 Gossip Enhancements

TODO

1.3.1 Enhancement 1

TODO

<https://www.apache.org/licenses/LICENSE-2.0>

1.4 ID Store Enhancements

IDStorage was enhanced with CouchDB

1.4.1 Enhancement 1

There are several benefits for using CouchDB storage for blocks, among these are:

- The use of a distributed and scalable storage between peers.
- Indexes are managed in the same storage.

<https://www.apache.org/licenses/LICENSE-2.0>

1.5 Private Data Storage Enhancements

Private Data Storage was also enhanced with CouchDB.

1.5.1 Enhancement 1

There are several benefits for using CouchDB storage for blocks, among these are:

- The use of a distributed and scalable storage between peers.
- Indexes are managed in the same storage.

<https://www.apache.org/licenses/LICENSE-2.0>

1.6 Peer Roles Enhancements

The peers are split into two roles to enhance endorsements and commit throughput in the Fabric network.

TODO additional comments

These roles are:

1.6.1 Endorser

An endorser peer will only do endorsements of transactions and will never write to the ledger, it will update its data from ledger periodically through gossip service.

There can be multiple endorsers for a given org.

TODO additional comments

1.6.2 Committer

A committer peer will deal with only committing transactions to the ledger. This way all other peers are relieved from writing to the ledger and execute larger number of endorsements.

There should be only 1 committer for a given org.

TODO additional comments

<https://www.apache.org/licenses/LICENSE-2.0>

1.7 Service Enhancements

TODO

1.7.1 Enhancement 1

TODO

<https://www.apache.org/licenses/LICENSE-2.0>

1.8 Transient Private Data Store Enhancements

The Transient Private Data Store enhancements allows for faster (local) transient storage on the peer to enhance transactions endorsements and eventually commits. The original storage used by the peer is leveldb which is limited by the local filesystem I/O operations.

1.8.1 GCache

The LevelDB storage implementation of the Transient store has been replaced by GCache to store private data in-memory as opposed to on disk.

The same store functions are tested against the original Fabric transient store unit tests and additional generic unit tests to ensure private data consistency.

<https://www.apache.org/licenses/LICENSE-2.0>

CHAPTER 2

Have Questions?

We try to maintain a comprehensive set of documentation for various audiences. However, we realize that often there are questions that remain unanswered. For any technical questions relating to TrustBloc not answered here, please use [Gitter](#) (an alternative to Slack) on the [#trustbloc-questions](#) channel.

Note: Please, when asking about problems you are facing tell us about the environment in which you are experiencing those problems including the OS, which version of Docker you are using, etc.

Note: If you have questions not addressed by this documentation, please visit the [Have Questions?](#) page for some tips on where to find additional help.
