fabric-sdk-go

2017-02-28

# HIP identifier

Hyperledger Fabric Go SDK

# Sponsors

* Troy Ronda, SecureKey Technologies (troy.ronda@securekey.com)
* Aleksandar Likic, SecureKey Technologies (aleksandar.likic@securekey.com)
* Firas Qutishat, SecureKey Technologies (firas.qutishat@securekey.com)
* Jim Zhang, IBM (jzhang@us.ibm.com)
* Gari Singh, IBM (garis@us.ibm.com)

# Abstract

Implementation of the Hyperledger Fabric SDK Design Specification v1.0 with extensions in the Go language.

# Context

At the moment, there is no publicly available implementation of the Fabric SDK written in Go. We propose to enable both Fabric Clients and Peers to use this SDK.

# Motivation

The primary motivation is to allow programs written in Go to interact with the Hyperledger Fabric as clients, as per [1].

In addition, as the Hyperledger Fabric itself is written in Go, the Go version of the Client SDK is required to implement some extensions to the Fabric, such as [5].

Some of the new Fabric features ([5], [6], [7], [8], [9]) require extensions to the Client SDK Specification. We are planning to implement client support for these features and extensions in the Fabric Go SDK, and also contribute to the Fabric client specification .

# Status

The implementation effort started in December 2016 and is being used internally by SecureKey. The code is hosted at GitHub[2] under the Apache v2.0 license.

The implementation generally follows the dynamics of the Node.JS Client SDK implementation. At the moment, the current Go SDK implementation is missing two features (compared to the Node.JS Client SDK): ChainCode deployment and Chain initialization. These features should be implemented in due time.

# Solution

Our Go Client SDK implementation supports only the gRPC API. REST is not supported. The only exception is the interaction with the Fabric-CA, which requires REST.

Generally, we attempt to reuse as much code from the Fabric as possible. For example, we use the bccsp library[3] for cryptographic key management and cryptographic operations, and the Fabric-CA client[4] for interaction with the Fabric-CA.

In the process, whenever it makes sense, we propose improvements and patches to expose parts of the Fabric libraries so they can be used from other packages.

# Effort and resources

SecureKey has built the initial implementation for the Go Client SDK using a member of one of our Scrum teams over a period of 5 weeks. That Scrum team will continue to update the SDK on a part-time basis.

# How to

The project is currently hosted at [2]. We, so far, generally implement the same set of tests that are present in the Node.JS Client SDK. All tests are currently passing against the commit levels of Fabric and Fabric-CA specified in the README file. We have tested using Go 1.7.3. Developers can clone the project from [2] and follow the README file to try it out.

We propose to host the Go SDK code at <http://gerrit.hyperledger.org/r/fabric-sdk-go>.

From the TSC discussion, the initial maintainer list is proposed to be the same as the sponsor list above.

As a result of proposal acceptance, the GitHub repo URL has moved from [2] to [11].

# References

1. “Hyperledger Fabric SDK Design Specification v1.0” - <https://docs.google.com/document/d/1R5RtIBMW9fZpli37E5Li5_Q9ve3BnQ4q3gWmGZj6Sv4/edit#heading=h.kspvx6g87vie>
2. “Hyperledger Fabric Client SDK for Go” - <https://github.com/securekey/fabric-sdk-go>
3. “BCCSP” - <https://github.com/hyperledger/fabric/tree/master/bccsp>
4. “Fabric-CA Client” - <https://github.com/hyperledger/fabric-ca/blob/master/lib/client.go>
5. <https://jira.hyperledger.org/browse/FAB-2438>
6. <https://jira.hyperledger.org/browse/FAB-2440>
7. <https://jira.hyperledger.org/browse/FAB-2441>
8. <https://jira.hyperledger.org/browse/FAB-2450>
9. <https://jira.hyperledger.org/browse/FAB-1154>
10. <https://jira.hyperledger.org/browse/FAB-2567>
11. <https://github.com/hyperledger/fabric-sdk-go>

# Closure

We are using the Go Client SDK in our development activities. We are planning to eventually implement the full Hyperledger Fabric Client SDK Specification, and also plan to keep it up to date with future SDK changes. As it already implements most SDK features, we expect that it will be used for new development by others whose language of choice is Go.