HLF Service Discovery

Session Topic or Project

Hyperledger Fabric (HLF) is a modular and extensible permissioned blockchain platform released to open-source and hosted by the Linux Foundation. The platform’s design exhibits principles required by enterprise grade business applications like supply-chains, financial transactions, asset management, food safety, and many more. For that end, HLF introduces several innovations, two of which are smart contracts in general-purpose languages (chaincode in HLF), and flexible endorsement policies, which govern whether a transaction is considered valid. Typical blockchain applications are comprised of two tiers: the first tier focuses on the modeling of the data schema and embedding of business rules into the blockchain by means of smart contracts (chaincode) and endorsement policies; and the second tier uses the SDK (Software Development Kit) provided by HLF to implement client-side application logic. However, there is a gap between the two tiers that hinders rapid adoption of changes in the chaincode and endorsement policies within the client SDK. Currently, the chaincode location and endorsement policies are statically configured into the client SDK. This limits the reliability and availability of the client in the event of changes in the platform, and makes the platform more difficult to use. In this work we address and bridge the gap by describing the design and implementation of Service Discovery. Service Discovery provides APIs which allow dynamic discovery of the configuration required for the client SDK to interact with the platform, alleviating the client from the burden of maintaining it. This enables the client to rapidly adapt to changes in the platform, thus significantly improving the reliability of the application layer. It also makes the HLF platform more consumable, simplifying the job of creating blockchain applications.

Session Leader

Artem Barger

Time Slots and Spaces

Experience Level of Participants

Hyperledger Fabric maintainer

Session Language

Russian

Programming Language(s)

- C / C++
- Go
- Java
- Javascript
- Python
- Rust
- Other

Other Prerequisites

The Plan

The Goals

Software to Install Before

Special Equipment Required