Санкт-Петербургский государственный университет

Математико-механический факультет
Кафедра системного программирования

Валл Михаил Андреевич

Разработка унифицированного API для различных реализаций технологии блокчейн

Курсовая работа

Научный руководитель:
ст. пр. Кириленко Я. А.

Санкт-Петербург
2017

Оглавление

Введение

В настоящее время, технология распределенного реестра - блокчейн приобретает все большую и большую популярность. Финансовые и технологические компании разрабатывают собственные реализации технологии блокчейн, внедряют блокчейн в бизнес, вступают в консорциумы для совместной работы над технологией. Примерами таких консорциумов являются Enterprise Ethereum Alliance[4], Hyperledger[6], R3[11].

Одной из ключевых проблем технологии блокчейн, как и многих других молодых технологий, является проблема стандартизации. Переход с одной реализации технологии блокчейн на более подходящуу новую может быть очень болезненным.

Целью данной работы является создание унифицированного API для использования различных реализаций технологии блокчейн. В данной работе изложено в каких сценариях может понадобится общее API, приведены сходства и различия данных блокчейнов.

1. Введение в предметную область

1.1. Блокчейн

Блокчейн (blockchain) - это распределенная децентрализованная база данных, которая поддерживает постоянно растущий список неизменяемых записей. Новые записи добавляются следующим образом: формируется новый блок, в котором включаются эти записи в хэш предыдущего блока. Хеш каждого блока зависит от записей блока и служебной информации. После формирования блок рассылается всем участникам сети, которые проверяют его на валидность, и когда он корректен, записывают в локальную копию. Таким образом, в процессе работы получается цепочка блоков, содержащих в себе все записи.

1.2. Умный контракт

"Умный контракт" (smart contract) - это некий электронный алгоритм, описывающий набор условий, при наступлении которых, выполняется заданный алгоритм. Например, при появлении записи о переводе некоторой суммы денег от пользователя A пользователю B, появляется запись о передаче права на владения некоторым активом.

2. Постановка задачи

Детально изучить принцип работы блокчейнов Hyperledger Fabric v0.6[7], Ethereum v1.6.1[5], Bitcoin v0.14.1[3], Multichain v1.0 beta 1 [8] Выделить общие и различные части выше приведенных блокчейнов и реализовать для них унифицированное API, предварительно разработав архитектуру библиотеки.

3. Области применения

Необходимость использовать унифицированное API может возникнуть в любом проекте, не желающем завязываться на каком-то определенном блокчейне.

На данный момент унифицированное API разрабатывается как часть системы, позволяющей собрать набор данных о работе блокчейна под различной нагрузкой, которые в дальнейшем можно использовать для анализа производительности блокчейна и влияющих на нее факторов. Данная система необходима для определения, сможет ли блокчейн обеспечить определенный уровень производительности для решения конкретной задачи. Унифицированное API позволяет не писать код под каждую реализацию блокчейна для генерации нагрузки и сбора данных.

4. Решение

4.1. Сходста и различия блокчейнов

4.1.1. Сходства

1. Во всех рассматриваемых в данной работе блокчейнах есть поддержка получения информации о блоках (хеш предыдущего блока, хеш запрашиваемого блока, список транзакций, время добавления блока в блокчейн), номера последнего блока, информации о подключенных узлах сети (IP адрес и id узла).

4.1.2. Различия

1. Алгоритм достижения консенсуса - у Bitcoin и Ethereum алгоритм достижения консенсуса - proof-of-work[10]. В этом случае право на публикацию нового блока получает тот участник, который решил трудоемкую задачу и предоставил ответ. Минусом такого способа является необходимость больших вычислительных мощностей для решения задачи. Hyperledger Fabric и Multichain используют различные реализации решения задачи о византийских генералах[9]. Проблема такого подхода в том, что узлов-злоумышленников может быть не больше 33%, в отличии от алгоритма proof-of-work, где таких узлов должно быть меньше 51%.

2. Структура блока - различия заключаются в хранении различной служебной информации.

3. Умные контракты - в рассматриваемых реализациях способ описания и публикации умных контрактов кардинально отличается. Например, Bitcoin позволяет писать умные контракты с весьма ограниченной функциональностью, так как язык программирования Bitcoin Script не является Тьюринг-полным, в отличие от других реализаций блокчейна.

4.2. Реализация

API реализовано в виде библиотеки, предоставляющей набор методов для взаимодействия с блокчейном. В унифицированном API написан интерфейс Manager, который реализуют BitcoinManager, EthereumManager, FabricManager, MultichainManager, BlockchainManager. Конечный пользователь пользуется только BlockchainManager, подавая ему такие параметры как URL и название блокчейна. В BlockchainManager происходит подключение к необходимому блокчейну. На данный момент API поддерживает следующую функциональность: авторизация пользователя на узле, отправка транзакции, отправка сообщения, получение последних транзакций, получение информации о блоке, блокчейне, подключенных узлах сети. Из блока можно получить хеш текущего и предыдущего блока, список транзакций, время добавления блока в блокчейн. Из информации о блокчейне можно получить номер последнего блока. На рис. 1 представлена диаграмма классов.

4.3. Используемые технологии

Java использована как основной язык разработки. Golang использовался для написания "умного контракта" для взаимодействия с Hyperledger Fabric. JSON-RPC применялся для работы и получения информации об узлах Bitcoin, Multichain, Ethereum, REST API для получения информации об узлах блокчейна Hyperledger Fabric. Для взаимодействия с Hyperledger Fabric использовалась библиотека fabric-java-sdk. Для взаимодействия с другими блокчейнами было достаточно использовать API, предоставляемое разработчиками блокчейна.

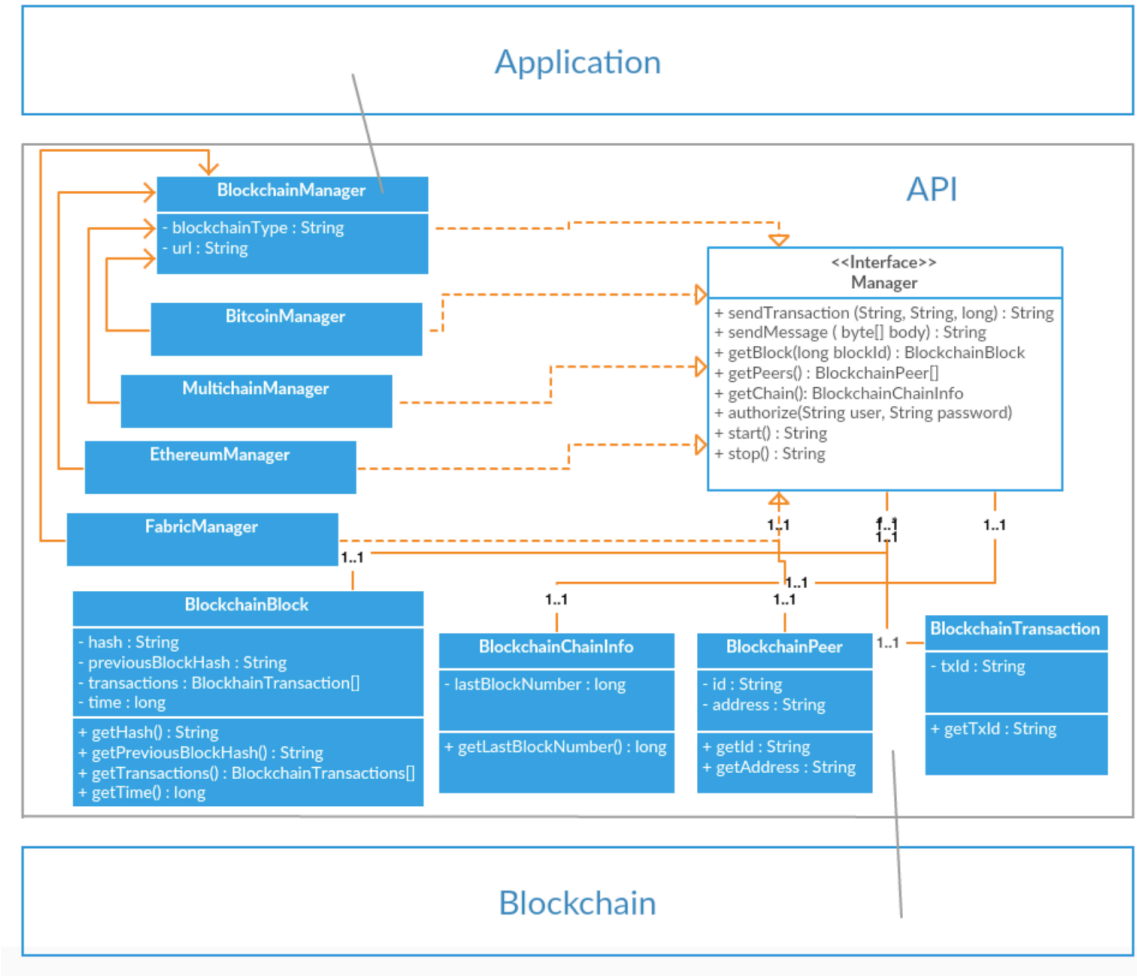

Рис. 1: Диаграмма классов

Заключение

В рамках данной работы получены следующие результаты:

1. Изучен принцип работы технологии блокчейн и ее реализаций (Hyperledger Fabric v0.6, Ethereum v1.6.1, Bitcoin v0.14.1, Multichain v1.0 beta 1).

2. Реализованно унифицированное API для приведенных выше блокчейнов.В дальнейшем планируется добавить поддержку Parity[2] и Corda[1].

Список литературы

[1] URL: https://www.corda.net/.

[2] 2017. — URL: https://parity.io/.

[3] Bitcoin. — 2017. — URL: https://bitcoin.org/en/.

[4] Enterprise Ethereum Alliance. — 2017. — URL: https://entethalliance.org/.

[5] Ethereum. — 2017. — URL: https://www.ethereum.org/.

[6] Hyperledger. — 2017. — URL: https://www.hyperledger.org/.

[7] Hyperledger Fabric. — 2017. — URL: http://hyperledger-fabric.readthedocs.io/en/v0.6/.

[8] Multichain. — 2017. — URL: http://www.multichain.com/.

[9] Practical Byzantine Fault Tolerance. — 1999. — URL: http://pmg.csail.mit.edu/papers/osdi99.pdf.

[10] Proof-of-work. — 1999. — URL: https://www.emc.com/emc-plus/rsa-labs/staff-associates/proofs-of-work-protocols.htm.

[11] R3. — 2017. — URL: http://www.r3cev.com/.