CasualChain
Technology overview

2024/05/21 Ryoji Kamei

What is CasualChain?

A blockchain core system that doesn't have any incentive features inside
- Man shall not live by crypto currency alone

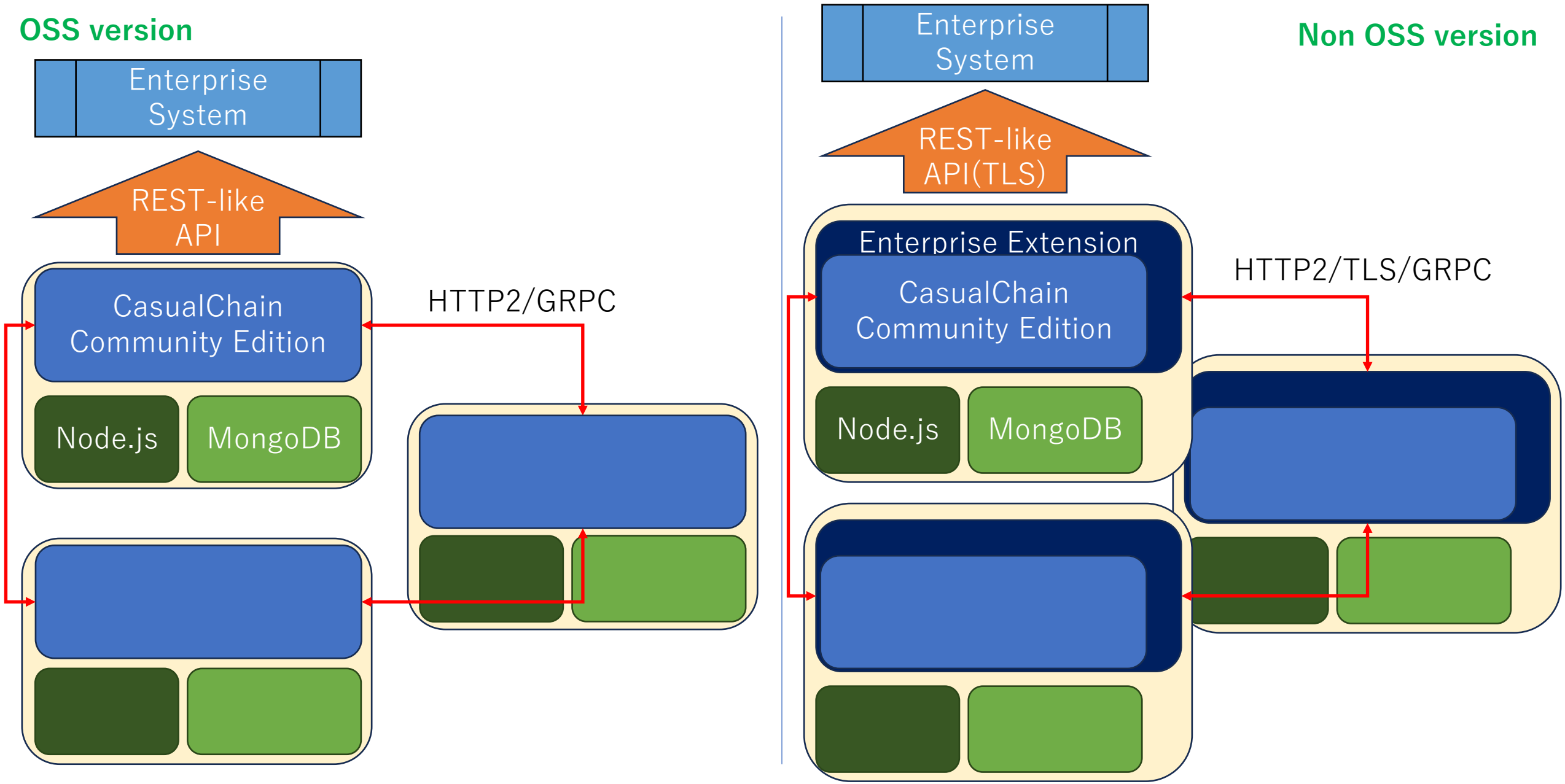
[from README]

A private blockchain engine intended for CASUAL integration into general enterprise systems.
1. Benefit from blockchain technology without holding crypto assets
2. Easy access using REST-like API. No dependence on specific programming languages
3. It does not use blockchain jargon. It can be understood using common DB terminology
4. License: MIT for the main part, non-profit free license (PUEL) for the extended part
5. No incentive function, so cannot be used by an unspecified number of people (only digital signature function at each node)

It runs on Linux systems with the following components installed
- Node.js 18 (20 will be supported soon)
- OpenSSL
- Git
- MongoDB (4.4 or later)

OSS (Community Edition, CE) and non-OSS (Enterprise Edition, EE) versions exist

System Architecture



Features and comparison between CE and EE

Features	CE	EE
Transactional additive recording	✓	✓
Support for large transactions (up to 15MiB per transaction)	✓	✓
Mechanism for temporarily pooling transactions waiting to be blocked and transparent handling of transactions in the pool	✓	✓
Original blocking algorithm	✓	✓
Configure access rights per node	✓	✓
Blocking communication with rogue nodes	✓	✓
Bulk blocking of multiple transactions	✓	✓
Uses elliptic curve cryptography to reduce data volume and increase speed	✓	✓
REST-like API	✓	✓
HTTP2 inter-node communication	✓	✓
Easily retrieve the history chain between transactions	✓	✓
Detect and repair tampering, branching, and inter-node data inconsistencies	✓	✓
Scheduling settings for internal tasks, such as automatic blocking	✓	✓
Transactions waiting to be blocked can be preserved by writing them to disk		✓
On-memory transaction I/O available		✓
More advanced APIs are available, such as blocking at any given time		✓
TLS available for RESTful API and inter-node communication		✓
Multi-tenant (*A single chain can be safely shared by multiple tenants to increase chain strength)		✓

REST-like API

For users	CE	Descriptions
/get/byjson	✓	Get data hit by KV pairs in JSON
/get/byoid/:oid(¥¥w{24})	✓	Get specific data in JSON by specifying oid
/get/alltxs	✓	Get all transactions (including transactions waiting to be blocked)
/get/blocked	✓	Get all blocked transactions
/get/pooling	N	Get all transactions in the pool
/get/poolingdelivered	N	Get only those transactions in the pool that have already been transferred to other nodes
/get/lastblock	N	Get the last block in the chain
/get/totalnumber	N	Get total number of blocks
/get/history/:oid(¥¥w{24})	✓	Get a series of connected transactions prior to the specified transaction
/post/byjson	✓	Register one data in JSON format

In the non-OSS version (EE), GET-based APIs obtain only the data of its own tenant; the POST-based API sets its own tenant ID.

For admins	CE	Descriptions
/sys/initbc	✓	Initialization of the blockchain (re-initialization is not possible in principle)
/sys/opentenant	N	Securing a new tenant parcel
/sys/closetenant	N	Close an existing tenant parcel
/sys/deliverpooling	A	Replicating transactions in the pool to other nodes
/sys/blocking	A	Block synchronized transactions
/sys/syncblocked	✓	Repair tampering, branching, and inter-node data inconsistencies in blocked transactions
/sys/syncpooling	✓	Repair inter-node data inconsistencies for transactions in the pool
/sys/synccache	N	Forced synchronization between cache and DB in on-memory transaction mode

Limitations in OSS version (CE)
✓: Fully available
A: Automatically run only
N: Not available

Note: MongoDB's License Binding

❗ If you use MongoDB as a non-open source product and provide it as a service, you must purchase a commercial license (paid version or cloud version).

Cases	Use OSS version (CE) as is or customize it and release it as OSS	Use non-OSS version (EE) or use non-OSS customized version
You or the company to which you delivered the system uses the system internally	Free version of MongoDB available	Free version of MongoDB available
You or the company to whom you delivered the system provides services to third parties	Free version of MongoDB available	Requires purchase of MongoDB commercial license

Therefore, from the user's point of view, CasualChain CE itself is under a loose MIT license, but to avoid paying MongoDB, it is necessary to release the OSS of the customized part of CasualChain, and the overall license is close to the GPL.

Future issues/ideas for consideration

Current version

- Support node version 20
- Distribute in SEA(Single Executable Application) format
- Support dynamic configuration change
- Support for merging as well as branching of transaction history

Next version or later

- Fast API via HTTP2
- Support for data stores other than MongoDB (cost savings, EE only)
- Expanded cloud-native functionality, including scale-out support
- Per-tenant authentication configuration
- Develop a variant of PoA and started supporting public chains *Not incentive-based

Cooperation with IPFS (separate service)

- Support for handling larger files
- Supports executable file distribution (equivalent to so-called smart contracts)