Hyperledger Sawtooth for Application Developers

Module 7: Course Conclusion

[MODULE 7] Course Conclusion

- Summary
- Resources and References
- Glossary

Summary

In this course, you learned these important concepts for Sawtooth application development:

- Blockchain basics
- Features of the Hyperledger Sawtooth enterprise blockchain platform, including the separation of core-level and application-level functionality that allows an application's business logic to be handled by a client front-end and a transaction processor (the equivalent of a smart contract) that runs on the validator node
- How to design and create a distributed application for the Hyperledger Sawtooth enterprise blockchain platform
- A detailed walk-through of a basic application, Sawtooth Simple Supply, that demonstrated how to use the Sawtooth SDKs to create an application with a web front end, a transaction processor for the blockchain business logic, and a custom REST API for communication
- How to run the Sawtooth Simple Supply application and understand its log messages

Resources and References

Learn about Sawtooth application development:

- Sawtooth Application Developer's Guide
- Hyperledger Sawtooth - Introductory Tutorial: rahulr92's article on Wordpress
- Understanding Hyperledger Sawtooth—Proof of Elapsed Time: Keenan Rilee's article on Medium.com

See example Sawtooth applications:

- Sawtooth Supply Chain
- Sawtooth Marketplace:

Join the Sawtooth development community:

- Hyperledger Sawtooth project page: Links to code, documentation, examples, and latest news
- #sawtooth: Hyperledger Sawtooth channel for general discussions and questions
- Hyperledger Sawtooth mailing list

Glossary

Application
Set of Sawtooth components that implement business logic and support functions. A Sawtooth application (also called a transaction family) includes a data model, a state schema, and the transaction operations for the application.

Batch
Group of related transactions. In Sawtooth, a batch is the atomic unit of state change for the blockchain. For a batch with multiple transactions, if one transaction fails, all transactions in that batch fail. (The application is responsible for handling failure appropriately.) See Transactions and Batches in the Sawtooth documentation.

Block
A set of transactions (state changes for the blockchain) that is cryptographically linked to the previous block on the blockchain.

Consensus
Method of reaching agreement between a group of *peers* (Sawtooth nodes) about the state of the blockchain, also called *global state*. Sawtooth supports the following consensus types:

- PoET-SGX: Sawtooth Proof of Elapsed Time (PoET) consensus using secure instruction execution with Intel® SGX
- PBFT: consensus where participants elect a leader to validate transactions, providing Byzantine Fault Tolerance (BFT)
- Raft: elects a leader for a arbitrary time term. Crash Fault Tolerant, but not BFT
- PoET simulator: Provides PoET-style consensus on any type of hardware, including a virtualized cloud environment
- Dev mode: Simplified random-leader algorithm that is useful for development and testing on a single Sawtooth node

Distributed ledger
Another term for blockchain. A Sawtooth distributed ledger records transactions, in chronological order, that are shared by all participants in a Sawtooth network. Each block on the ledger is linked by a cryptographic hash to the previous block.

Glossary (continued)

Event Subscriber
Optional application component that handles Sawtooth events (block commit and state delta information) and optional application-defined events.

Genesis block
First block on the blockchain. The genesis block contains configuration settings and initial values for the Sawtooth network.

Global state
Shared agreement on the state of the blockchain, according to the chosen *consensus* method. Note that each validator has its own instance of the state database, which is represented as a Merkle-Radix tree. For more information, see Global State in the Sawtooth documentation.

Off-chain
Term used for information that is stored by an application rather than on the blockchain.

On-chain
Term used to refer to data that is stored on the blockchain, such as configuration settings.

PBFT
Practical Byzantine Fault Tolerance is a consensus algorithm where participants elect a leader to validate transactions. If malicious participants are less than 50% of the network, they are overruled by the other participants.

Peer
Participant in a Sawtooth network; also called a *Sawtooth node* or *validator node*. Each peer runs a validator and the same set of transaction processors

Glossary (continued)

PoET
Sawtooth consensus algorithm that uses Proof of Elapsed Time (PoET) to agree on which participant should commit a block of transactions to the blockchain. PoET is a Nakamoto-style consensus algorithm that is designed to support large networks. It relies on secure instruction execution to achieve Nakamoto-style scaling benefits without the power consumption drawbacks of Proof of Work (PoW) consensus. Sawtooth includes two versions of PoET consensus: PoET-SGX and PoET simulator. For more information, see the Sawtooth documentation.

Reporting database
Sawtooth application component that stores event data (state changes) that is gathered by an *event subscriber*.

REST API
In Sawtooth, a component that adapts communication between a client and the validator using HTTP/JSON. Sawtooth includes a REST API that is used by clients such as the Sawtooth CLI commands. Developers can use this REST API or develop custom APIs for client-validator communication.

Sawtooth core
Central Sawtooth software that is responsible for block validation and publishing, consensus, and global state management. The Sawtooth architecture separates these core functions from application-specific business logic, which is is handled by Sawtooth applications (also called *transaction families*).

SGX
Intel® Software Guard Extensions. "An Intel® architecture extension designed to increase the security of application code and data." -- https://software.intel.com/en-us/sgx-sdk.

State delta
Event data that contains information about changes to the blockchain state.

Glossary (continued)

Transaction family
Another term for a Sawtooth application. A transaction family includes the design for the application's data model, state schema, and business logic, as well as the client and transaction processor. See Transaction Family Specifications in the Sawtooth documentation.

Transaction processor
Hyperledger Sawtooth component that implements the business logic for a Sawtooth application. The transaction processor validates transactions and updates state based on the rules defined by the application.

Validator
Hyperledger Sawtooth component that is responsible for validating batches of transactions, combining them into blocks, maintaining consensus with the Sawtooth network, and coordinating communication between clients, transaction processors, and other validator nodes.