

[Open in app](#)[Sign up](#)[Sign in](#)

Medium

 Search

An Introduction to Hyperledger Sawtooth and a Guide to Setting It Up

LeewayHertz · [Follow](#)

Published in HackerNoon.com

7 min read · Jul 6, 2019



Listen



Share



Image Source: Hyperledger

What's the Hype About Hyperledger Sawtooth

Hyperledger Sawtooth is an open source Blockchain platform founded by the Linux Foundation's open-source blockchain project, Hyperledger. Proposed by Intel, it is an enterprise-grade distributed ledger and was one of the first projects under the Hyperledger Umbrella.

With a pluggable consensus model, Hyperledger Sawtooth is :

1. Modular

2. Scalable

3. Supports Permissionless and Permissioned Infrastructure.

Its underlying design is aimed at keeping ledgers distributed and adding business logic to smart contracts.

In many standard blockchain based platforms, applications and core are executed and hosted on their platform, that may cause performance issues and security concerns.

However, Hyperledger Sawtooth is different from such platforms in a way that it isolates the core ledger system from the application specific environment. As a result, it simplifies the application development yet keeps the system safe and secure.

With Sawtooth's modular architecture, developers can develop applications in the programming language of their choice and can host, run and operate it on system periphery without hindering the core blockchain system.

An application on Hyperledger Sawtooth can be based on a core business logic to run business operations, or it can be built and run as a smart contract virtual machine with a self-governing mechanism to create, notify and execute the contracts between various members on the blockchain network.

Sawtooth's core system allows both of these applications to co-exist, allowing different types of applications to stay in the same instance of the blockchain network.

Applications on Sawtooth can select transaction rules, define the consensus mechanisms and select the required permissioning to decide the working of the digital ledger in a way that it meets the requirements of an enterprise.

Being extremely scalable. Hyperledger Sawtooth can offer high transaction throughput of data and is, therefore, a great option for handling production supply chains.

Features of Hyperledger Sawtooth

- **Private networks with the permissioning features**

Sawtooth overcomes the challenges of the permissioned network as the cluster of its nodes can be deployed with isolated permissioning. No centralized service can leak confidential information or transaction patterns.

The blockchain manages the settings that specify identities and roles so that participants within the network can access the information on the network.

- **Ethereum Contract Compatibility with Seth**

The Seth, Sawtooth-Ethereum integration project allows the interoperability of the Sawtooth platform to Ethereum. The project makes it possible for Ethereum smart contracts to get deployed to the Sawtooth platform with Seth transaction family.

- **Parallel Transaction Execution**

Many blockchains need serial transaction execution to ensure consistent ordering at every node on the network. However, Sawtooth contains an advanced parallel scheduler that classifies transactions into parallel flows. Based on locations in state accessed by a transaction, Sawtooth separates the transactions' execution from one another while handling contextual changes.

Transactions in Sawtooth are executed parallelly to prevent double spending even with several modifications to the same state. This type of scheduling leads to the potential increase in performance as compared to serial execution.

- **Isolation between the core system and the application level**

Hyperledger Sawtooth eases the development and deployment of an application by separating the application level from the core system level. It offers smart contract abstraction to allow developers to create contract logic in the programming language of their choice.

- Each application can define the custom transaction processors to meet the unique requirements. It provides transaction families to serve as an approach for low-level functions, like storing on-chain permissions, managing chain-wide settings and for particular applications such as saving block information and performance analysis.

- **Pluggable Consensus Mechanisms**

The consensus mechanism is the process of coming to an agreement among a group of participants on the blockchain network. Algorithms for achieving consensus with arbitrary faults need some kind of voting among a specific set of participants.

General consensus approaches comprise Nakamoto-style consensus that selects a leader through the lottery and variants of Byzantine Fault Tolerance mechanisms that run multiple rounds of explicit voting to form a consensus. Sawtooth supports multiple consensus types on the same blockchain which is chosen initially during the network setup and can be changed on a running blockchain with a transaction.

It supports the following consensus implementations:

Proof of Elapsed Time (PoET):

It is a Nakamoto-style consensus, designed as a production-grade protocol that can support large network populations. Relying on a secure instruction execution, it can achieve the scalability of Nakamoto-style consensus mechanism without the drawbacks of power consumption of the PoW algorithm.

PoET Simulator:

It enables PoET-style consensus on any hardware and a virtualized cloud environment.

Dev mode:

It is a simplified random-leader mechanism used for development and testing.

- **Sample Transaction Families**

Transaction language and the data model are implemented in a transaction family. Though users can build custom transaction families to represent their unique requirements, Sawtooth also provides various transaction families as models:

1. **IntegerKey** for testing deployed ledgers.

2. **Identity** for managing on-chain permissioning for validator and transactor keys to simplify identity management for public keys' lists.
3. **Settings** to provide an implementation for saving on-chain configuration settings.
4. **BlockInfo** for providing an approach to store information about a number of historic blocks.
5. **Smallbank** for managing performance analysis for performance testing and benchmarking when doing the comparison of blockchain systems' performance.

Getting started with Hyperledger Sawtooth Application Development?

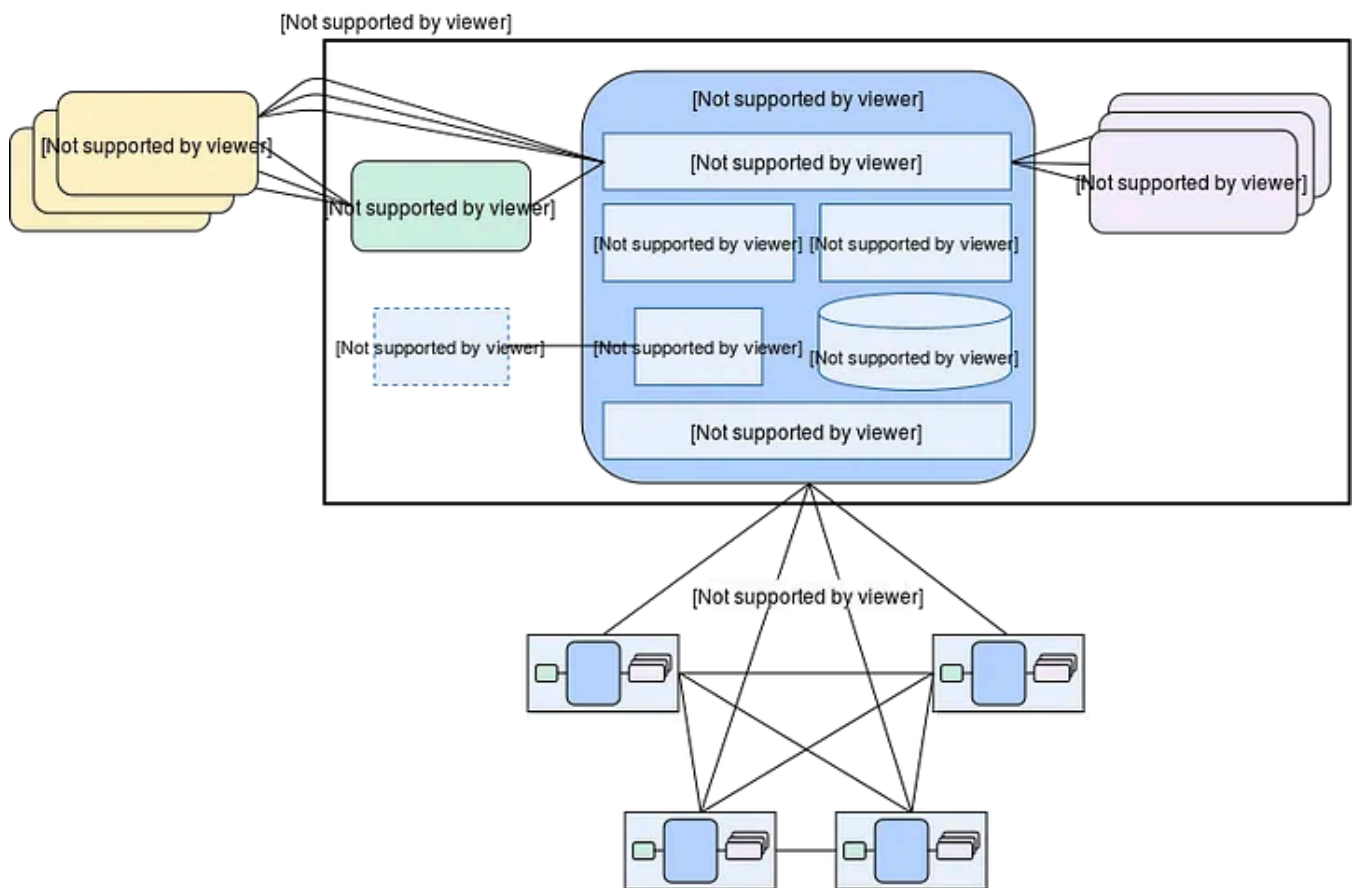


Image Source: Hyperledger Sawtooth Documentation

You will have to set up and execute a local validator to test your application before you start developing an application on Hyperledger Sawtooth platform. Once the network will be running, you can conduct new transactions and fetch the block data and resulting state from the blockchain via Sawtooth REST API and HTTP.

Sawtooth validators can run from prebuilt Docker containers which are installed using Ubuntu 16.904 or deployed in AWS from the AWS marketplace.

We will walk you through the process of Hyperledger Sawtooth Application Development using Docker Compose.

Installing Docker Engine and Docker Compose

Windows:

Docker Compose can be installed automatically when you install Docker Engine on Windows.

Linux:

On the Linux operating system, you will need to install both Docker Engine and Docker Compose.

macOS:

On macOS, you will have to install the latest version of the Docker Engine. Similar to Windows, Docker Compose is also installed automatically with Docker Engine on macOS.

Setting up Environment

A Docker Compose file is provided to state the process for building a Sawtooth environment. The Sawtooth environment comprises of the following containers:

- REST API connected to the validator
- A client container to run Sawtooth commands
- A single validator that uses dev-mode consensus
- The Settings, XO transaction processors and IntegerKey

Docker Compose file also represents the container images to obtain from Docker Hub and the network settings required for containers to interact correctly.

The Docker Compose File serves as the basis for the multi-container Sawtooth application development environment. Download the Docker Compose File and set up the environment for the development of Sawtooth application.

Starting up the Sawtooth environment involves the following tasks:

- Open a terminal window.
- Switch the working directory to the same directory where you stored the Docker Compose file.
- Run the command: `% docker-compose -f sawtooth-default.yaml up`

Downloading of the docker images may take a few minutes. After you find the containers registering and building initial blocks, move to the next step.

In case the environment needs to be stopped or reset, you need to log out of the client container to return to the default state and then press CTRL- C from the Windows system where you initially executed docker-compose.

How to Create and Submit transactions to the Sawtooth network

Sawtooth provides the `intkey` command to make sample transactions of the IntegerKey (`intkey`) transaction type for testing. This method uses `intkey` for preparing batches of `intkey` transactions that assign a few keys to random values, then increase and decrease those values randomly.

These batches are stored locally and distributed to the validator within the network.

Run the mentioned-below commands from the client container:

```
$ intkey create_batch -- count 10 -- key-count 5 $ intkey load -f batches.intkey -U http://rest-api:8008
```

The terminal window where you ran the `docker-compose` command will start logging output as a validator and `intkey` transaction processor manages the transactions submitted recently.

Rather than using `intkey load`, you can also use “`sawtooth batch submit`” command to submit transactions. For instance, submit the transactions in `batches.intkey` file with the command:

```
$ sawtooth batch submit -f batches.intkey -- url http://rest-api:8008
```

Viewing the Blockchain

Following are the different commands for viewing the block, list of blocks, a specific block and global state:

- **sawtoothblock**

This subcommand is used to see the blocks saved in the blockchain.

- **sawtoothblocklist**

This command is used to view the blocks stored by the state.

- **sawtooth block show — url <http://rest-api:8008> {BLOCK_ID}**

This command is used to view a specific blockchain using a block ID.

- **sawtoothstatelist**

This command is used for listing the nodes in the Merkle tree.

- **sawtooth state show — url <http://rest-api:8008> {STATE_ADDRESS}**

This command is used to view the data at a specific address.

Connecting to the REST API

Use the command “\$ curl <http://rest-api:8008/blocks>” to ensure that you can connect to the REST API via the host.

To connect the REST API from the host, use the command: “\$ curl <http://localhost:8008/blocks>”.

Due to the potential of Hyperledger Sawtooth to build modular and scalable blockchain apps, it is being used widely by blockchain companies to provide blockchain business solutions to enterprises and startups.

Shameless Plug a.k.a Our Work on Hyperledger Sawtooth

We have built a distributed ledger platform called TraceRx on the Hyperledger Sawtooth that enables end-to-end traceability across the pharma supply chain on the blockchain. The platform allows UNO to track the distribution of free aid and perform recalls quickly.

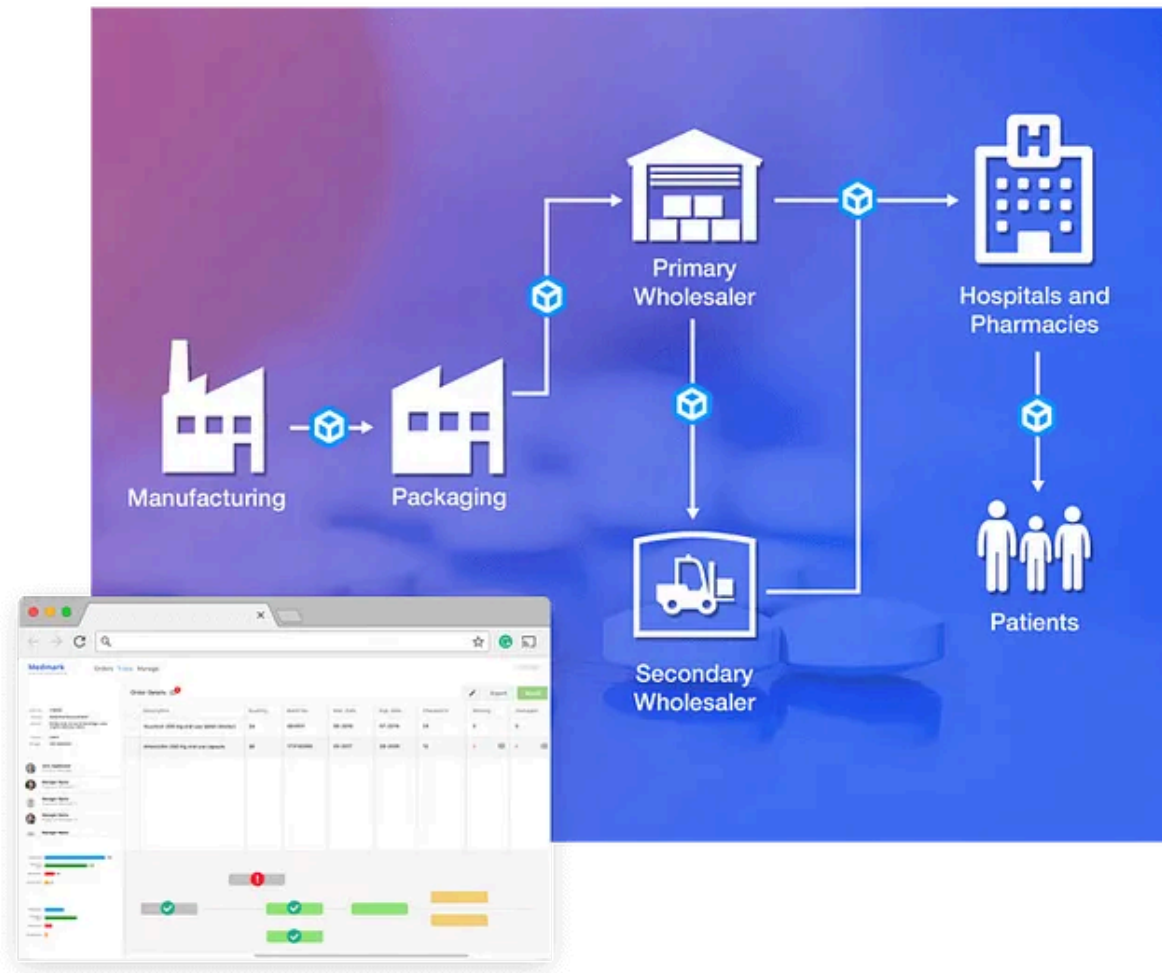


Image Source: LeewayHertz

At LeewayHertz, we have a team of Blockchain Consultants who can assist companies and individuals interested in advancing their business operations with distributed ledger technology.



Hackernoon Newsletter curates great stories by real tech professionals

Get solid gold sent to your inbox. Every week!

Sign Up



If you are ok with us sending you updates via email, please tick the box.
Unsubscribe whenever you want.

[Terms of Service](#)



I agree to leave medium.com and submit this information, which will be collected and used according to [Upscribe's privacy policy](#).

Powered by [Upscribe](#)

Hyperledger

Hyperledger Sawtooth

Blockchain

Blockchain Development

Blockchain Platforms



Follow

Published in **HackerNoon.com**

466K Followers · Last published May 27, 2020

Elijah McClain, George Floyd, Eric Garner, Breonna Taylor, Ahmaud Arbery, Michael Brown, Oscar Grant, Atatiana Jefferson, Tamir Rice, Bettie Jones, Botham Jean



Follow

Written by LeewayHertz

574 Followers · 68 Following

AI development company enabling innovation and rapid development We build cutting edge software solutions for startup. <https://www.leewayhertz.com>

Responses (1)



What are your thoughts?

Respond



Anna

over 2 years ago



I can't focus on the content because the green backgroud hurt my eyes. You might consider changing it :)



Reply

More from LeewayHertz and HackerNoon.com