Deploying and managing Hyperledger Sawtooth

Duncan Johnston-Watt & Kai Davenport
Blockchain Technology Partners
Agenda

- BTP Introduction
- Hyperledger Sawtooth
- Demo
“BTP has a clear value proposition — bringing the benefits of blockchain to business — and its leadership team has the necessary expertise to bring together the components to deliver”

Csilla Zsigri & William Fellows, 451 Research Impact Report
Focus

- We provide our customers with a blockchain management platform that leverages world class open source components *Hyperledger Sawtooth* and *Kubernetes* (CNCF) backed by the Linux Foundation.

- We work with our customers to co-create blockchain-based solutions that differentiate their business from their competitors.

- We support our customers and in collaboration with them create an operational model that meets their needs now and in the future.
Platform

- BTP Sextant
  - Hyperledger Sawtooth distribution
  - Leveraging cloud native Kubernetes
  - Fully curated and hardened platform
  - Providing unified user experience

- AWS Marketplace (Q4 18)
  - Standard & Professional Editions
  - Easy to fire up and get started
  - Industry standard PAYG pricing
  - With upgrade option to subscription

- BTP Sextant Editions
  - Standard
  - Professional
  - Enterprise

- Future Targets
  - Google Cloud Platform Marketplace
  - Azure Marketplace
  - Alibaba Cloud Marketplace
Sawtooth Design Philosophy

- Design for Scale
- Keep Distributed Ledgers Distributed
- Ease-of-use for Development & Deployment
- Make Smart Contracts Safe

@blockchainptp
Hyperledger Sawtooth 1.0
Architecture & Features

1.0 Released January 2018
Basic Concept

Clients
- Submits transactions; Queries the database

Validator Process
- Mediates access to the database

Transaction Processors
- Business logic; Validates transactions

Sawtooth Hosts
- State
A Couple More Pieces

Sawtooth Host

Clients

REST Service

Validator Process

State

Transaction Processor(s)

Other Validators

@blockchaintp
High-level Sawtooth Architecture

Sawtooth Host

- Clients
- REST Service
- Validator
  - Interconnect
  - Block Management
  - Transaction Handling
  - Consensus
  - State
  - P2P Network
- Transaction Processor(s)
- Other Validators
v1.0 Features: Parallel Execution

Sawtooth Host

Clients
- REST Service
- Validator
- Block Management
- Interconnect
- Transaction Handling
- Transaction Processor(s)

Clients
- Other Validators
- Multiple transactions per block can effect the same state
- Parallel scheduling
- Multi-process “smart contracts”
v1.0 Features: Multi-Language Support

Choose from:
- Python,
- JavaScript,
- Go,
- Rust*

Choose from:
- Go,
- JavaScript,
- Python,
- C++*,
- Java*,
- Rust*
v1.0 Features: On-chain Governance

Control the blockchain on the blockchain

Settings Transaction Family enables participants to agree on network policies

For example, vote on changing consensus parameters using registered public keys of consortia members.

Settings are extensible – they can be added after genesis.

<table>
<thead>
<tr>
<th>Setting (Examples)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>sawtooth.poet.target_wait_time</td>
<td>5</td>
</tr>
<tr>
<td>sawtooth.validator.max_transactions_per_block</td>
<td>100000</td>
</tr>
</tbody>
</table>
| sawtooth.validator.transaction_families    | [{
|                                           |   "family": "intkey",
|                                           |   "version": "1.0"
|                                           | },
|                                           |   { "family": "xo",
|                                           |     "version": "1.0"
|                                           | }])
Sawtooth Host

client

REST Service

Validator

Consensus = DevMode

Abstract Consensus

Set Consensus := PoET

DevMode

PoET

PoET-SGX

Settings

Permissioning

Other Smart Contracts

Raft*

PBFT*

* Available in Sawtooth 1.1

v1.0 Features: Dynamic Consensus

@blockchain tp
Hyperledger Sawtooth 1.0
Application Development

1.0 Released January 2018
Application Development

Transaction Family

Client

Transaction Creation

Validator Process

State

Data Model

Transaction Processor

Business Logic

@blockchaintp
Transaction Processor ≈ Smart Contracts

Transaction Families **encapsulate business logic** on Sawtooth

A Transaction Family can be as simple as a single transaction format, with associated validity and state update logic…

…or as complex as a VM with opcode accounting and bytecode stored in state -- ‘smart contracts’

The *choice* is up to the developer

Sawtooth allows these concepts to **coexist** in the same instance of the blockchain -- same blocks, same global state
Transaction Families: The Transaction Processor

All validators in the network run every authorized transaction processor.

On receipt of a transaction the validator will call the TP’s Apply() method.

Business logic simply goes in Apply() and gets and sets state as needed.
Hyperledger Sawtooth

Source: https://sawtooth.hyperledger.org/docs/core/releases/1.0/app_developers_guide/aws.html#overview-of-sawtooth-components

October 2018

@blockchaintp
Check it out

Give Sawtooth a try
- Work through the tutorials
- Build your own transaction family to explore use cases

Become a contributor
- Join the community
- Help with docs, code, examples
- Become an expert and help others on chat

Links
- Code: https://github.com/hyperledger/sawtooth-core
- Docs: https://sawtooth.hyperledger.org/docs/
- Chat: https://chat.hyperledger.org/channel/sawtooth
Sextant

- Infrastructure
  - Create a new Kubernetes cluster
  - Join an existing Kubernetes cluster
  - Failover for high availability

- Sawtooth
  - Deploy managed version of sawtooth
  - Auto configure network and storage
  - Deploy and manage standard or custom transaction processors
  - Monitor deployment
Sextant: Infrastructure

- Provision a new Kubernetes cluster
  - From nothing to running cluster with a button click
  - Automate the setup across popular cloud providers
  - Handles multi-zone for high availability
  - Download “kubeconfig” for CLI “kubectl” access

- Join an existing Kubernetes cluster
  - For bare metal or custom clusters - provide a “kubeconfig” file and sextant can deploy to it
Sextant: Sawtooth

- Deploy managed Sawtooth
  - Configuration of validators
  - Networking between validators
  - Connect to external seeds
  - Persistent storage for validators

- High availability
  - k8s persistent volumes provide HA on node failure
  - k8s ingress provides static external access endpoint
Sextant: Sawtooth

**Standard transaction processors**
- Deploy standard TP’s like RBAC with sensible configuration options
- We can begin to curate a library of commonly used TP’s

**Custom transaction processors**
- Deploy custom TP’s to the cluster
- Any Docker image can be consumed
- Easy network access to the validator (localhost:4004)
Sextant: Sawtooth

- Monitoring
  - InfluxDB powering Grafana dashboards
  - Monitor things like “transactions per second”
  - Fully integrated into Sawtooth validators

- Updates
  - We curate Kubernetes manifests and Docker images
  - The cluster can easily be upgraded in situ

October 2018
Sextant: Components

- **UI**
  - Browser application for easy and intuitive management of clusters
  - Real time view on cluster status

- **Management server**
  - Connects to cloud providers to provision of new clusters
  - Communicates to k8s api server for running cluster
  - Deploys and manages Sawtooth resources on k8s cluster
Let’s deploy sawtooth to AWS!