HYPERLEDGER

BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

Sawtooth v1.0

Dan Middleton
Hyperledger Sawtooth Maintainer

February 2018

Agenda

Sawtooth Design Motivations

Sawtooth 1.0 Features

Distributed App Development

- Code: https://github.com/hyperledger/sawtooth-core
- Demos: https://sawtooth.hyperledger.org/examples/
- Docs: https://sawtooth.hyperledger.org/docs/

Hy

Hyperledger Sawtooth is an open source distributed ledger framework and one of the nine business blockchain and distributed ledger technologies hosted by The Linux Foundation. Hyperledger Sawtooth delivers unique capabilities. A few examples are included below:

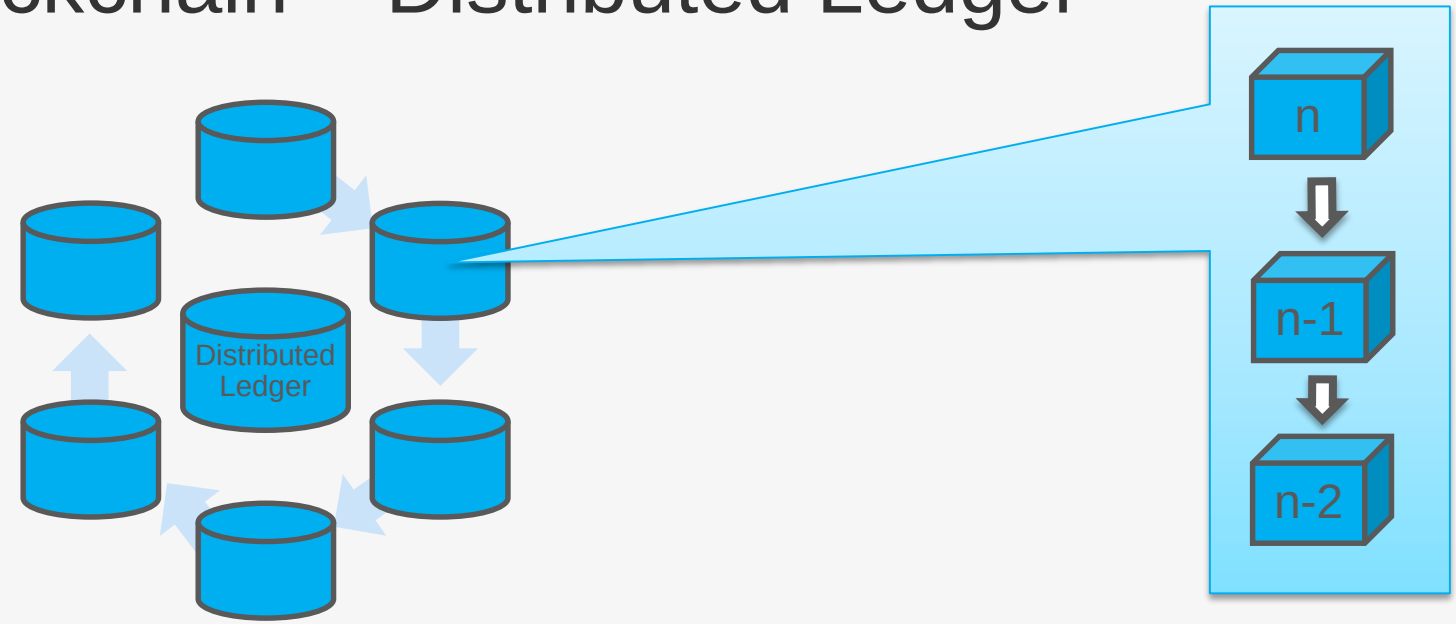
- ***On-chain governance** – Utilize smart contracts to vote on blockchain configuration settings such as the allowed participants and smart contracts.*
- ***Advanced transaction execution engine** – Process transactions in parallel to accelerate block creation and validation.*
- ***Support for Ethereum** – Run solidity smart contracts and integrate with Ethereum tooling.*
- ***Dynamic consensus** – Upgrade or swap the blockchain consensus protocol on the fly as your network grows, enabling the integration of more scalable algorithms as they are available.*
- ***Broad language support** – Program smart contracts in your preferred language, with support including Go, JavaScript, Python and more.*

*The efforts around Hyperledger Sawtooth have grown significantly; from the initial code contribution in April 2016, to Active status graduation in May 2017, to today's version 1.0 availability. Hyperledger Sawtooth is supported by an active community: organizations including **Amazon Web Services, Active Ticketing, Bitwise.io, Cloudsoft, Context Labs, Dot BC Media, Ericsson, Hacera, Huawei, IBM, Intel, Microsoft Azure, Monax, Open Music Initiative, PokitDok, R3, T-Mobile, Wind River and more than 50 engineers have contributed to the project.** Additionally, Proof of Concepts (PoC's) have been deployed to support multiple business cases including music and media content rights attribution, recording healthcare transactions, Know Your Customer (KYC) in financial services and others.*

Hyperledger Sawtooth **Open Source Community**

HYPERLEDGER SAWTOOTH

Blockchain = Distributed Ledger

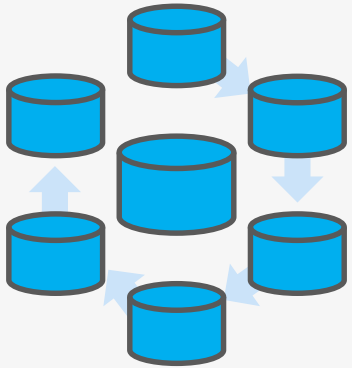


Each node is an instance of a database (ledger) managed by all participants.

Within each database, blocks of transactions are cryptographically chained in order.

Why Blockchain

Mutually distrusting organizations that update the same database.



Immutable transaction history

High availability

- Crash fault tolerant
- Byzantine fault tolerant
- Liveness

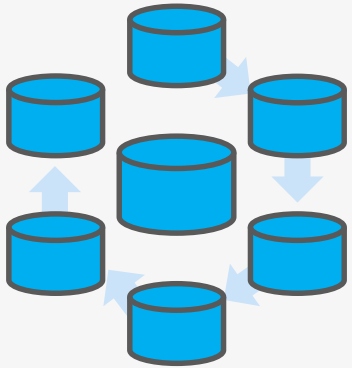
Why Not Blockchain

Active Research Areas:

- Throughput
- “Private” Transactions

Wrong Usage Model:

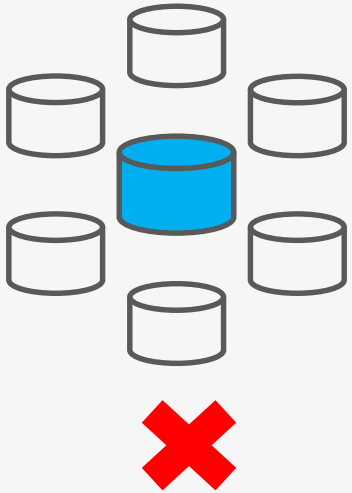
- Internal-only Business Process

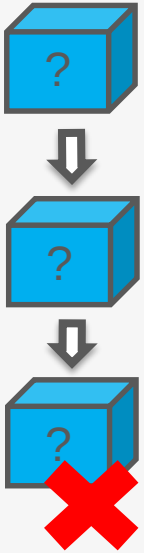


When people talk about blockchain security, they mostly mean availability and integrity guarantees. Confidentiality is open research.

Bad Enterprise Blockchain Shortcuts

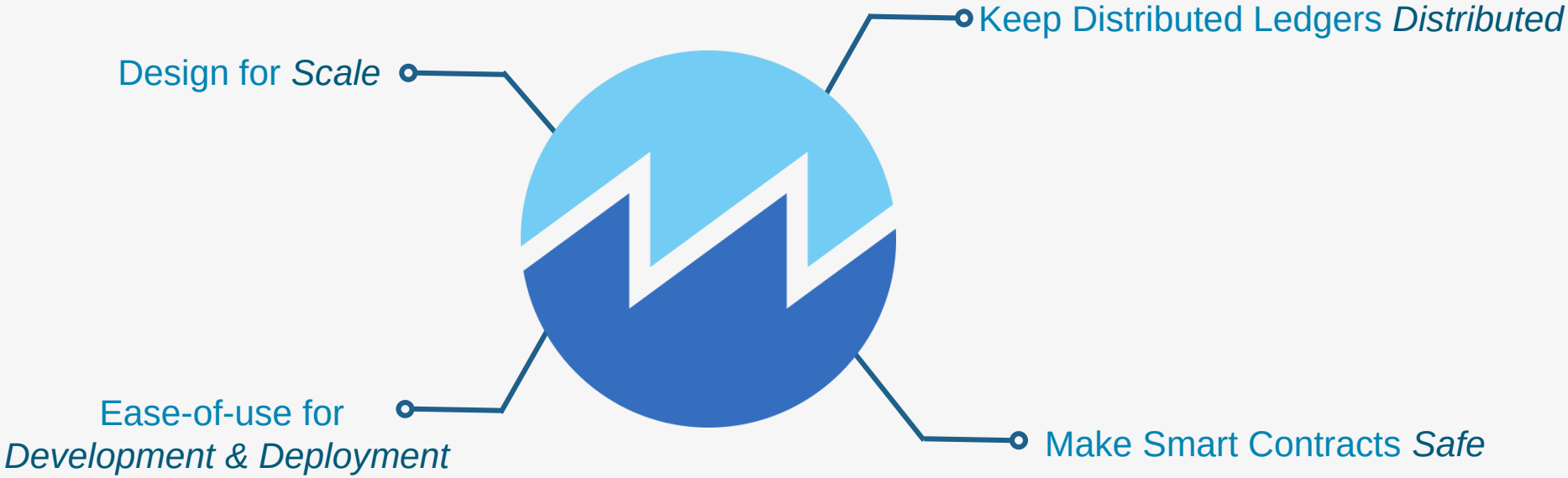
- Centralized Architectures
- No Ledger State (database fields)





A centralized architecture removes the main value of a distributed ledger.
Committing only transaction receipts turns the database into a log of opaque events.

Sawtooth Design Philosophy



HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

Hyperledger Sawtooth 1.0 Architecture & Features

1.0 Released January 2018

v1.0 Highlighted New Features

Advanced Transaction Execution

- Parallel Execution
- Multi-Language Support
  - Build apps in your language of choice

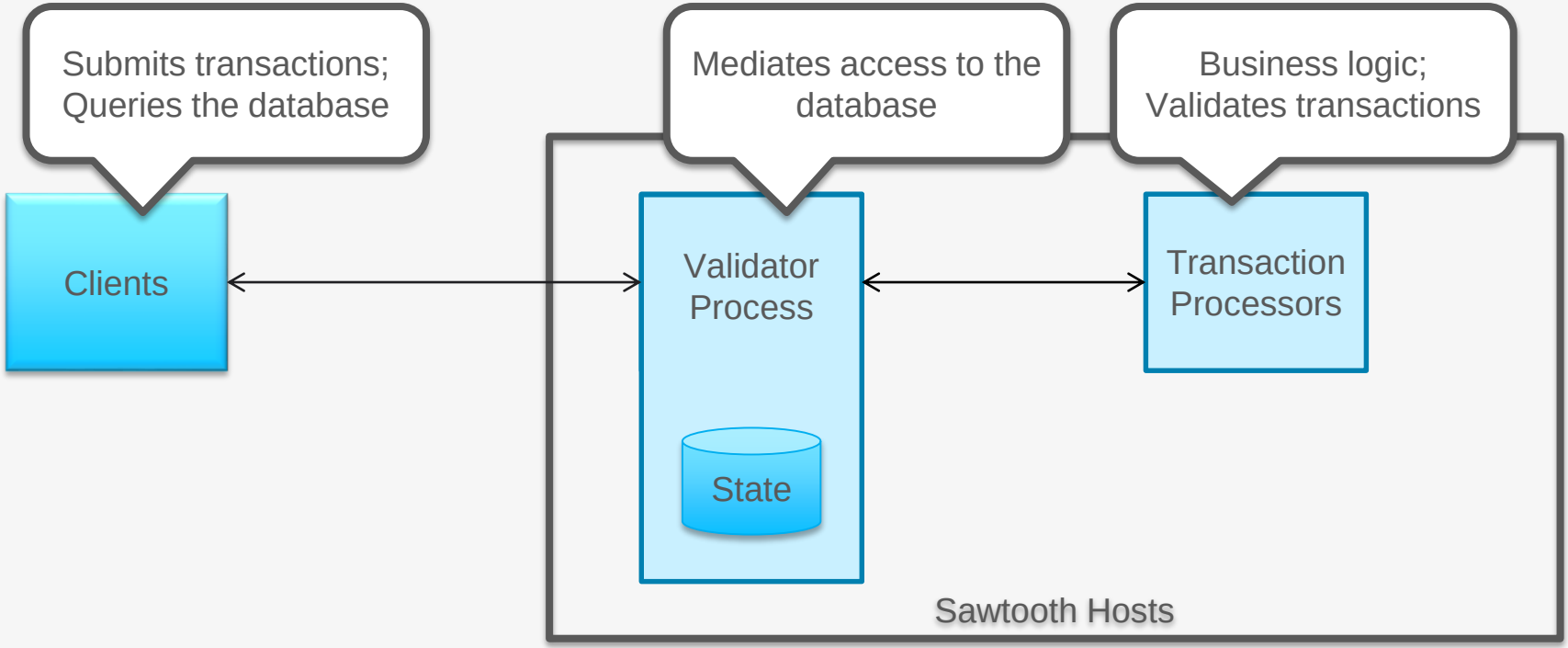
On-chain Governance

- Dynamic Consensus
  - Proof of Elapsed Time (PoET)
- New Permissioning Features

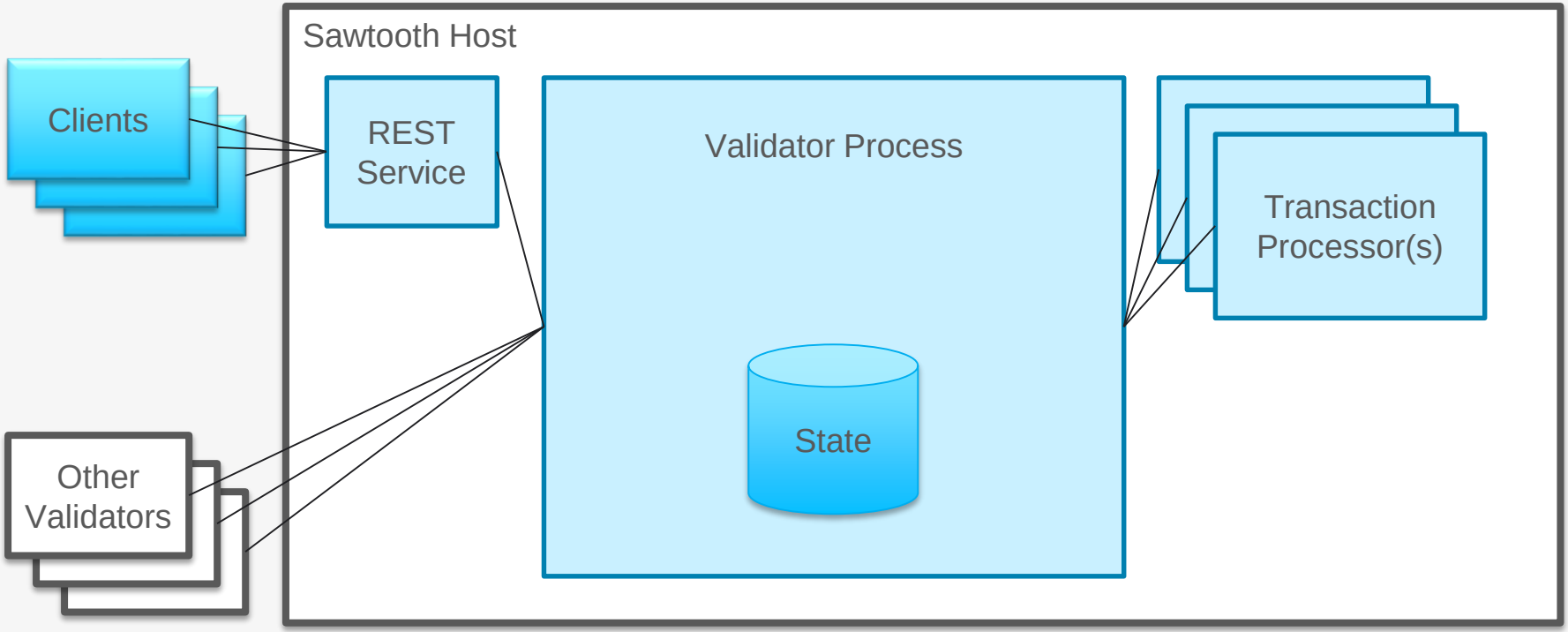
Distributed Applications

- Seth
  - Sawtooth + Ethereum
  - Run solidity on Sawtooth
- Supply Chain
  - Provenance of goods
  - Telemetry / tracking

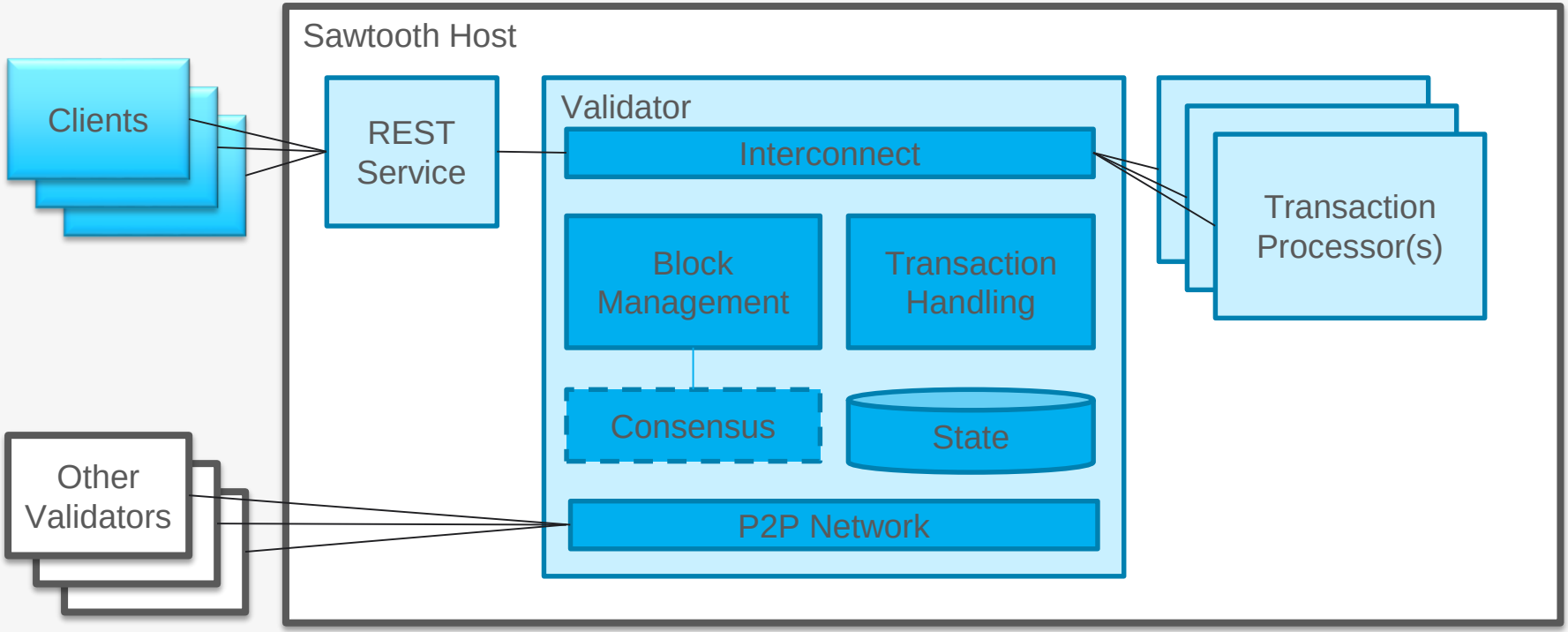
Basic Concept



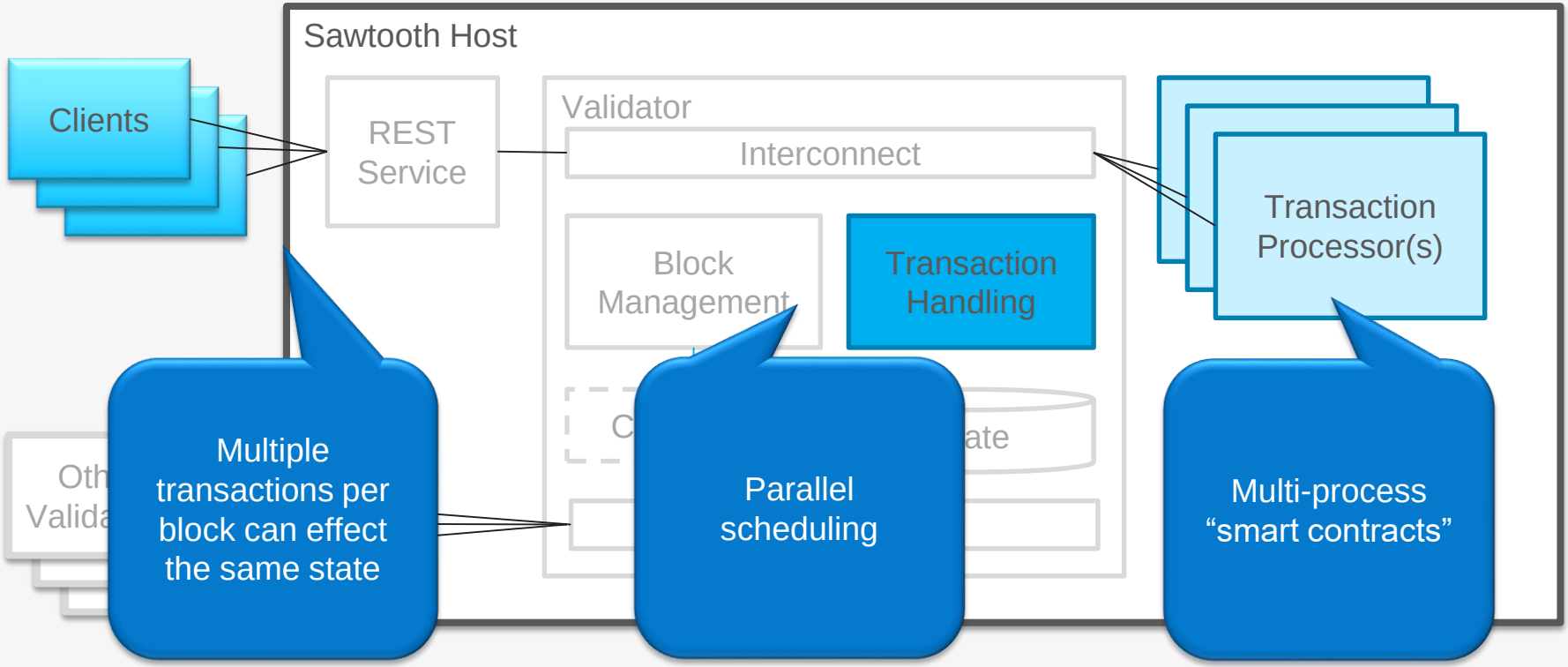
A Couple More Pieces



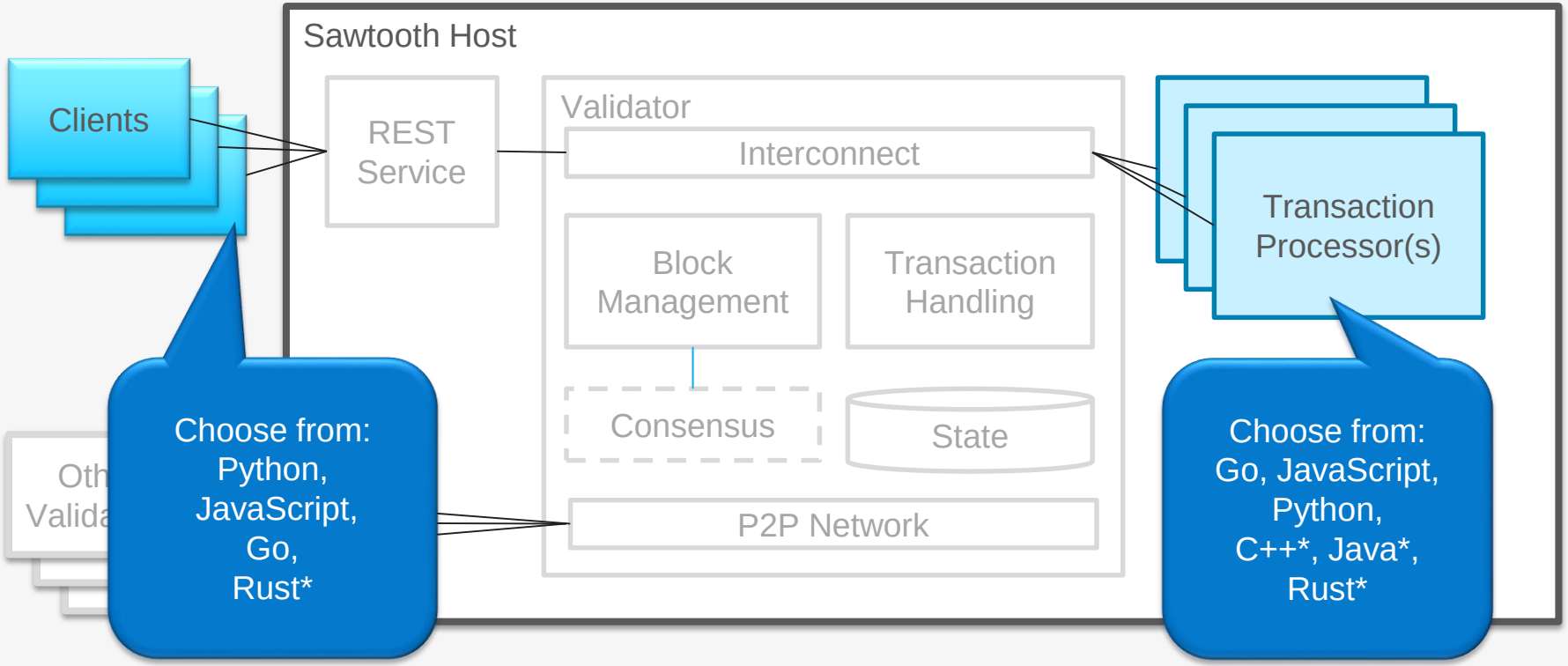
High-level Sawtooth Architecture



v1.0 Features: Parallel Execution

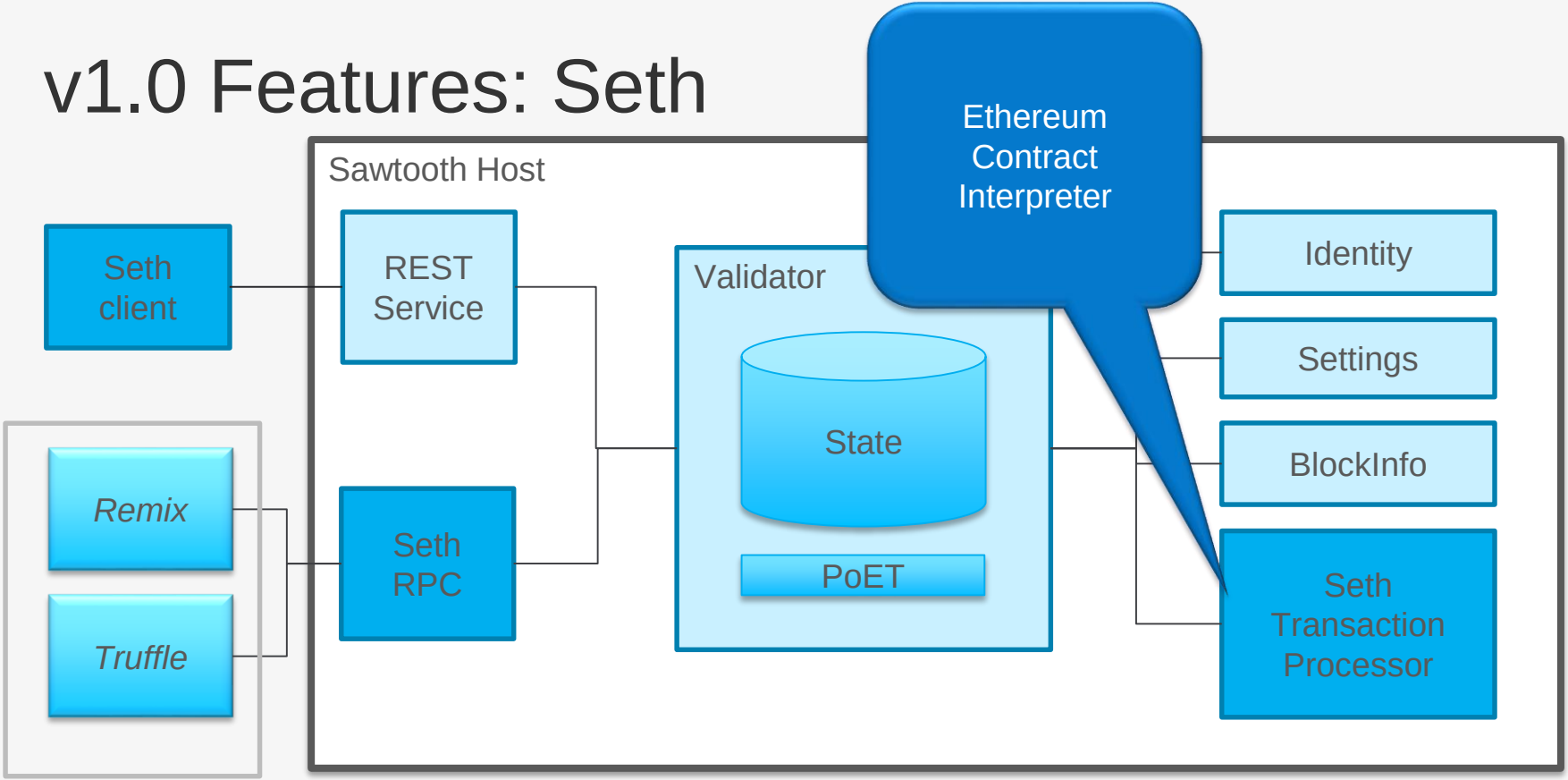


v1.0 Features: Multi-Language Support



*No all features are available in these languages in v1.0.1

v1.0 Features: Seth



v1.0 Features: On-chain Governance

Control the blockchain on the blockchain

Settings Transaction Family enables participants to agree on network policies

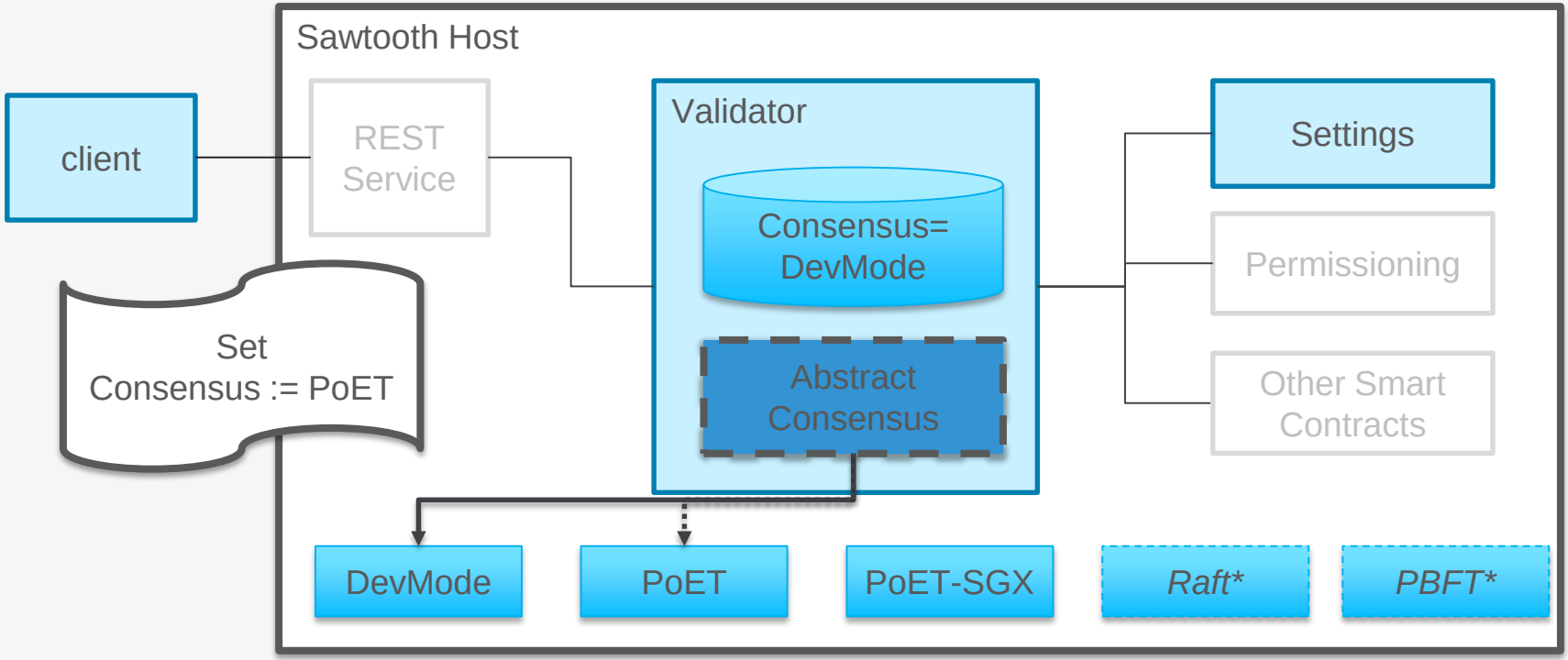
For example, vote on changing consensus parameters using registered public keys of consortia members.

Settings are extensible – they can be added after genesis.

Setting (Examples)	Value
sawtooth.poet.target_wait_time	5
sawtooth.validator.max_transactions_per_block	100000

sawtooth.validator.transaction_families	[{
	"family": "intkey",
	"version": "1.0"
	},
	{
	"family": "xo",
	"version": "1.0"
	}]

v1.0 Features: Dynamic Consensus



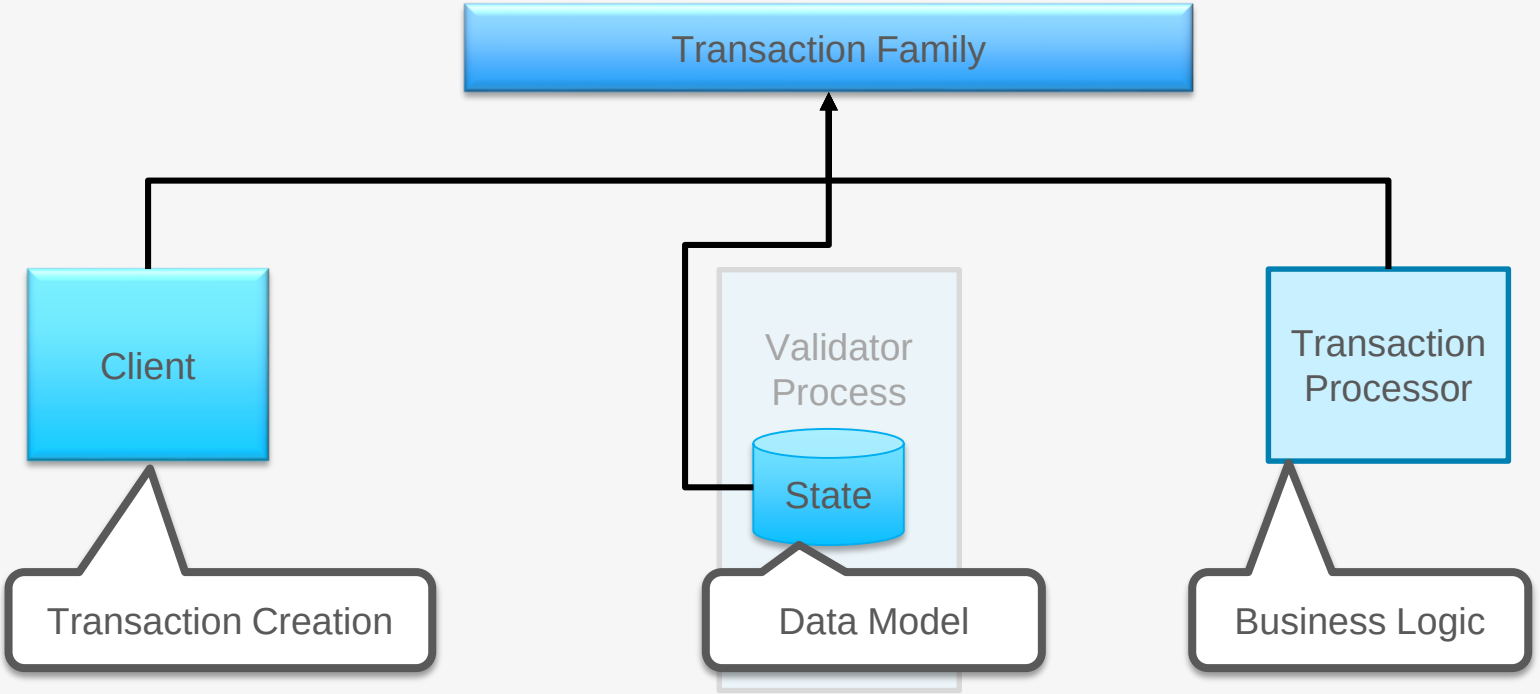
*Future consensus options

HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

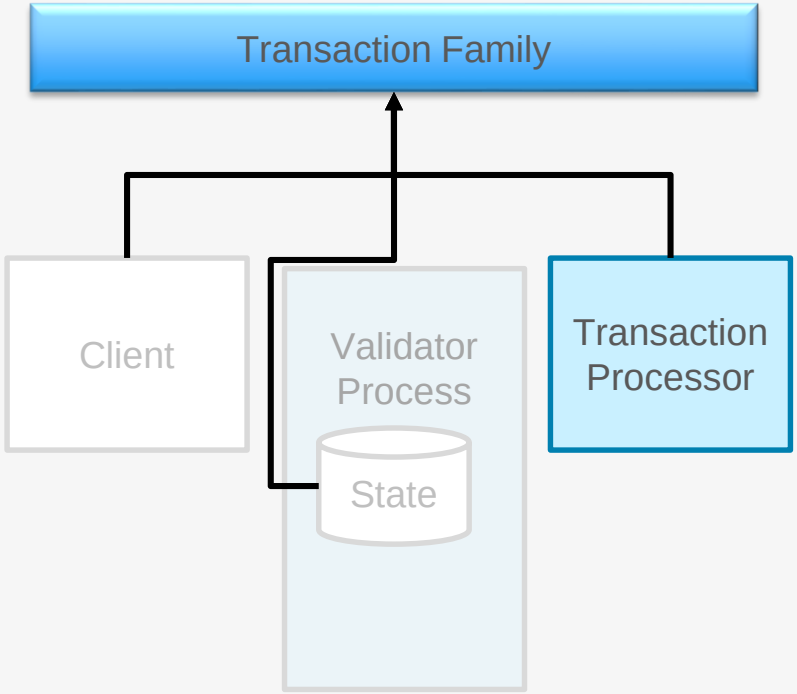
Hyperledger Sawtooth 1.0 App Development

1.0 Released January 2018

Application Development



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.

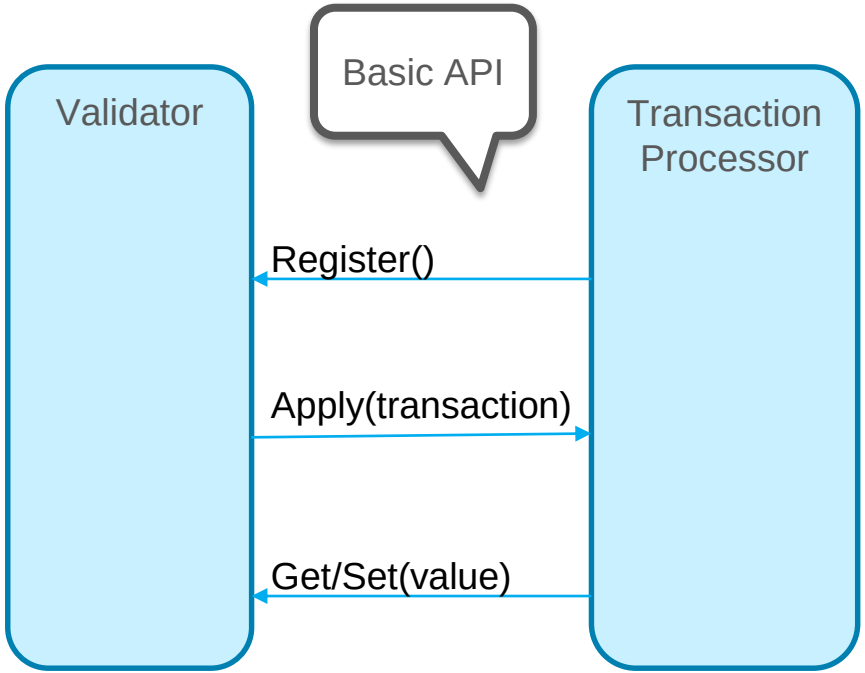
Sawtooth allows these concepts to **coexist** in the same instance of the blockchain -- same blocks, same global state.

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.



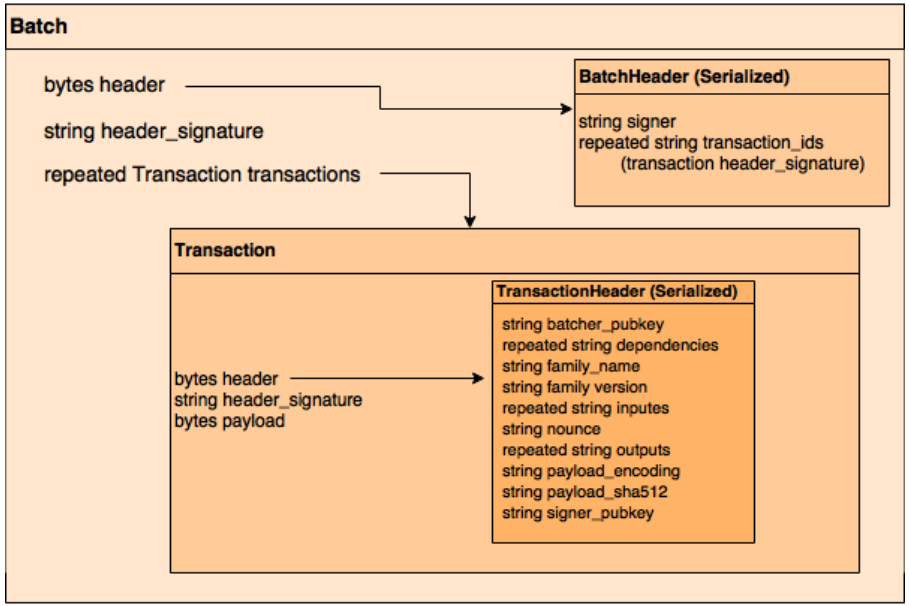
Transaction Families: The Client

Clients can be browser apps, CLIs, etc.

Main job is to package and sign transactions & batches

Clients can post batches through the Rest API or connect to the validator directly

Transactions, Batches, and Blocks



Transactions are wrapped in batches which provide an atomic unit of commit for multiple transactions (which can span transaction families).

Transactions declare input and output addresses (including wildcards) to allow for state access isolation calculations (topological sort on DAG) in the scheduler.

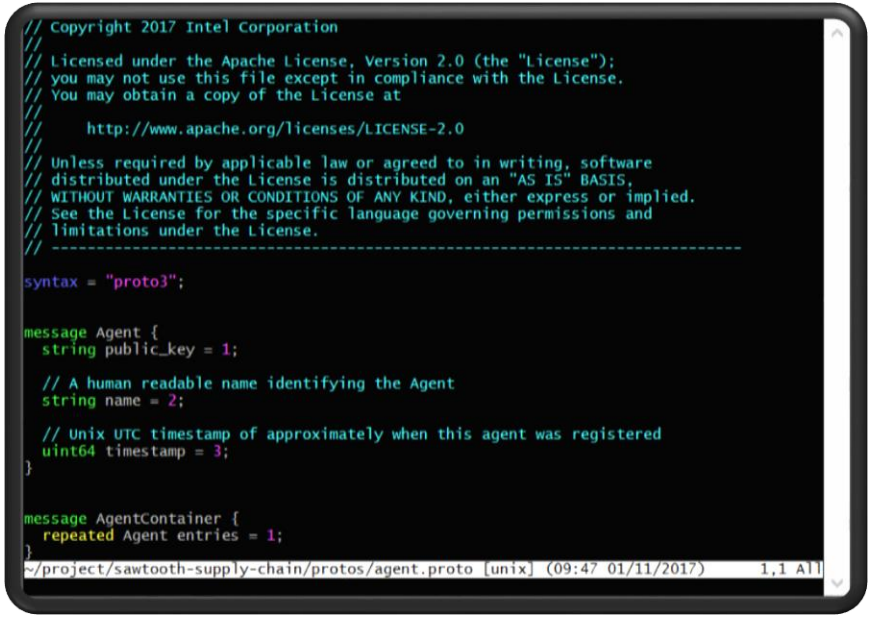
These inputs and outputs are enforced by the Context Manager on the context established for the transaction.

This allows parallel validation and state delta aggregation across a potentially large number of transactions (and across blocks).

Transaction Families: The Data Model

Both Client and Transaction Processor must use the same…

- Data model
- Serialization / encoding
- Addressing scheme



```
// Copyright 2017 Intel Corporation
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
//     http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// ------------------------------------------------------------------------------

syntax = "proto3";


message Agent {
  string public_key = 1;

  // A human readable name identifying the Agent
  string name = 2;

  // Unix UTC timestamp of approximately when this agent was registered
  uint64 timestamp = 3;
}


message AgentContainer {
  repeated Agent entries = 1;
}
~/project/sawtooth-supply-chain/protos/agent.proto [unix] (09:47 01/11/2017)      1,1 All
```

Example Applications & Code

Supply Chain: https://github.com/hyperledger/sawtooth-supply-chain

Marketplace: https://github.com/hyperledger/sawtooth-marketplace

RBAC: https://github.com/hyperledger/sawtooth-hyper-directory

Dev Guide: https://sawtooth.hyperledger.org/docs/core/releases/1.0.1/app_developers_guide.html

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

Thanks!