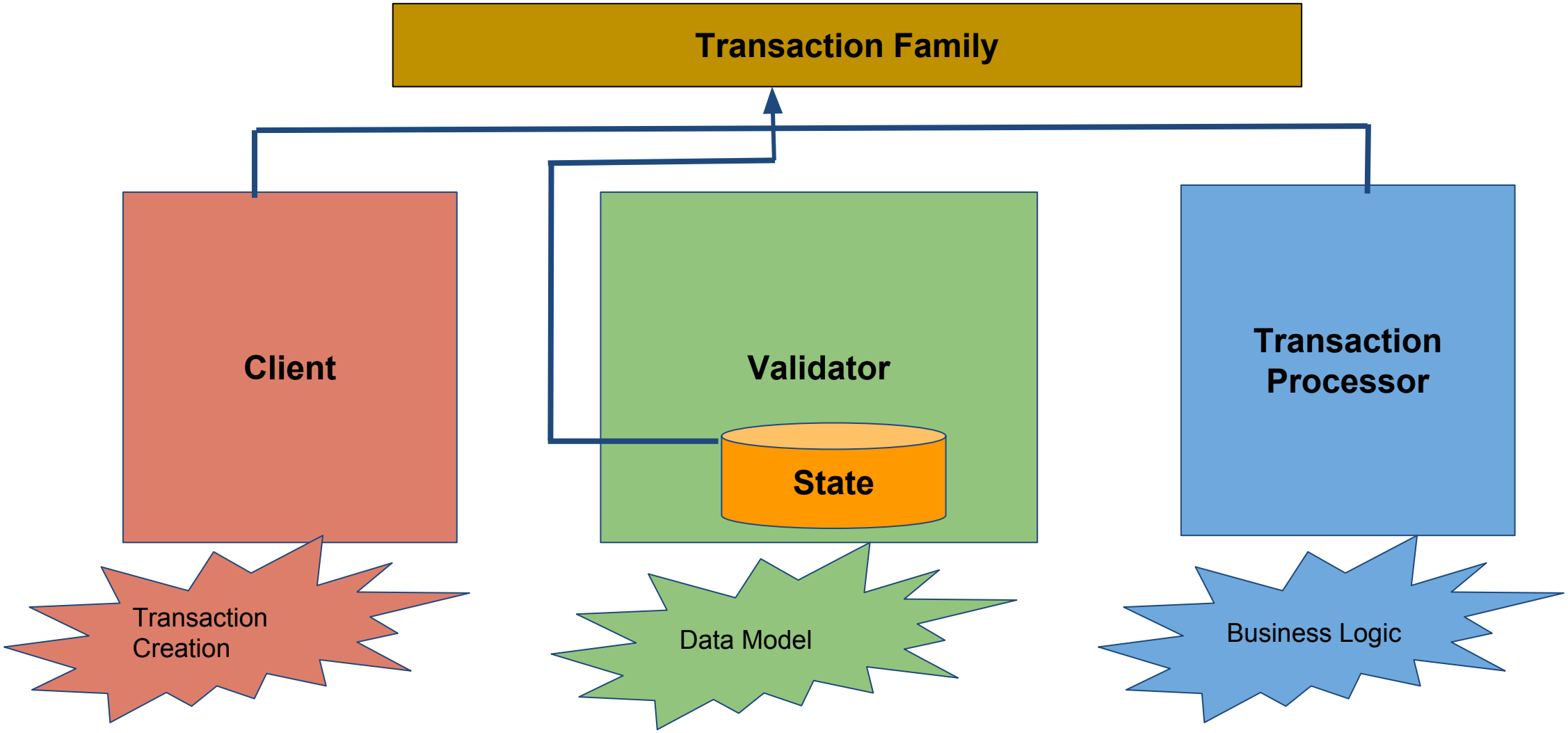
App-Development Fundamentals

Transaction Family



Transaction Family

- The client, TP and state of an application is together called a Transaction Family
- Family is uniquely identified by its name and version
- A Family will have a set of addresses to set state values

Private and Public Key

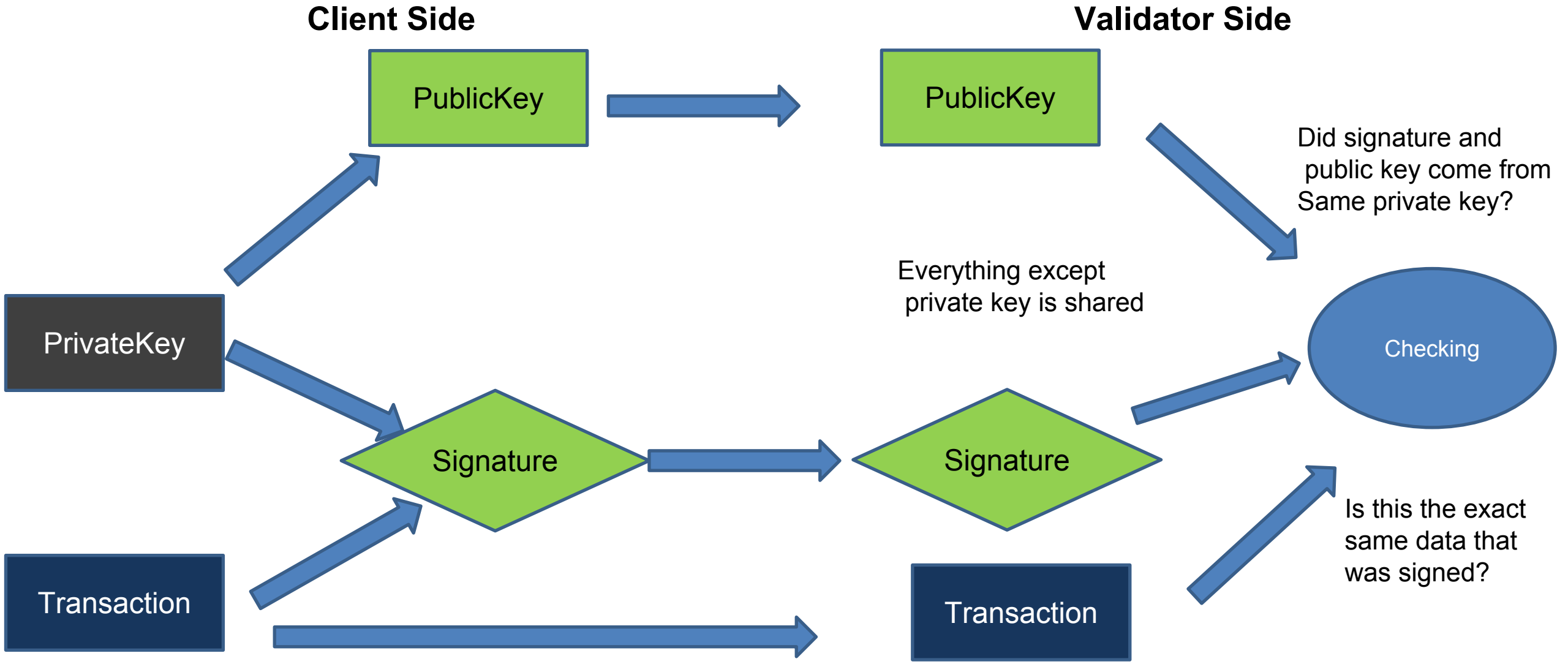
private key

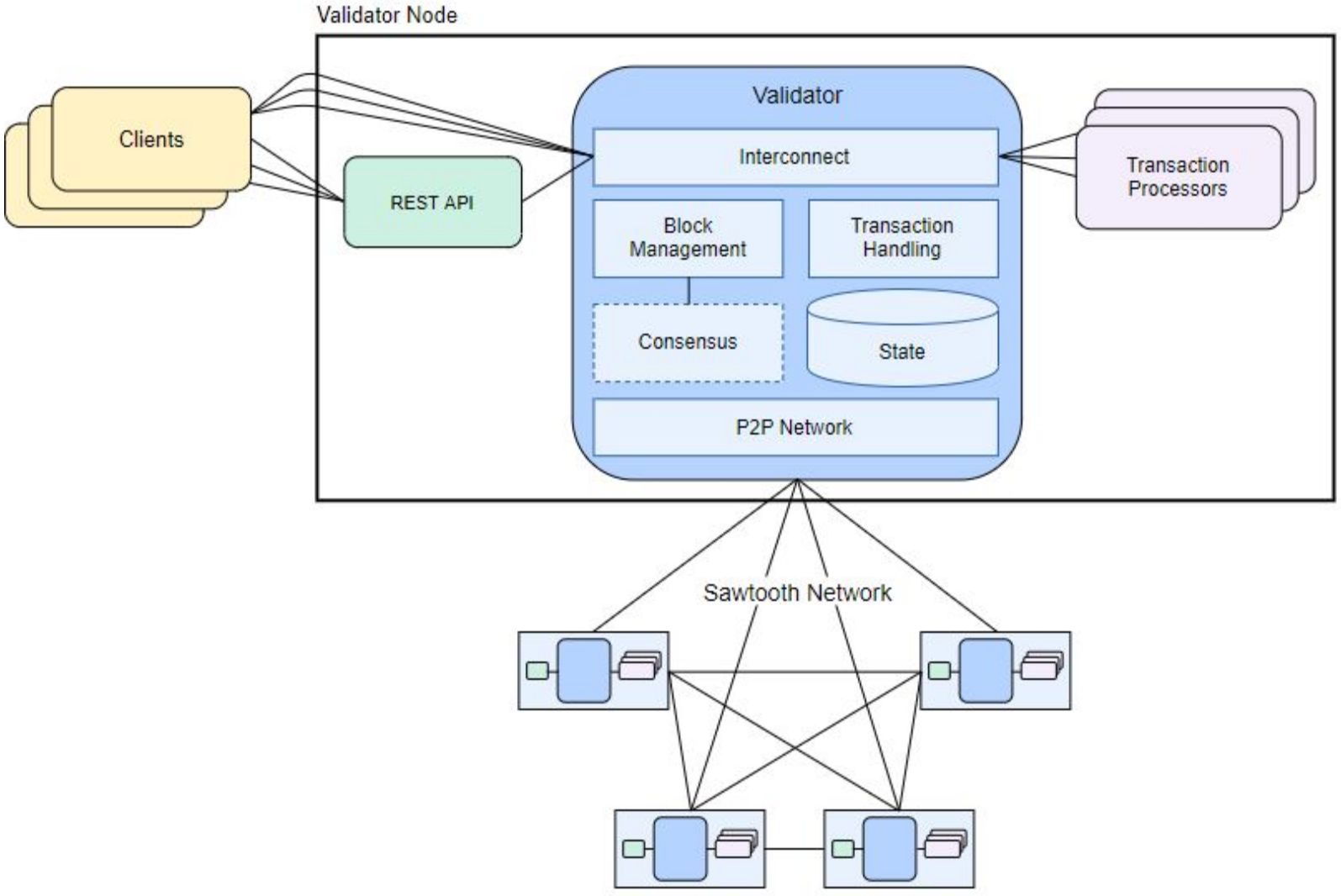
large, randomly generated number

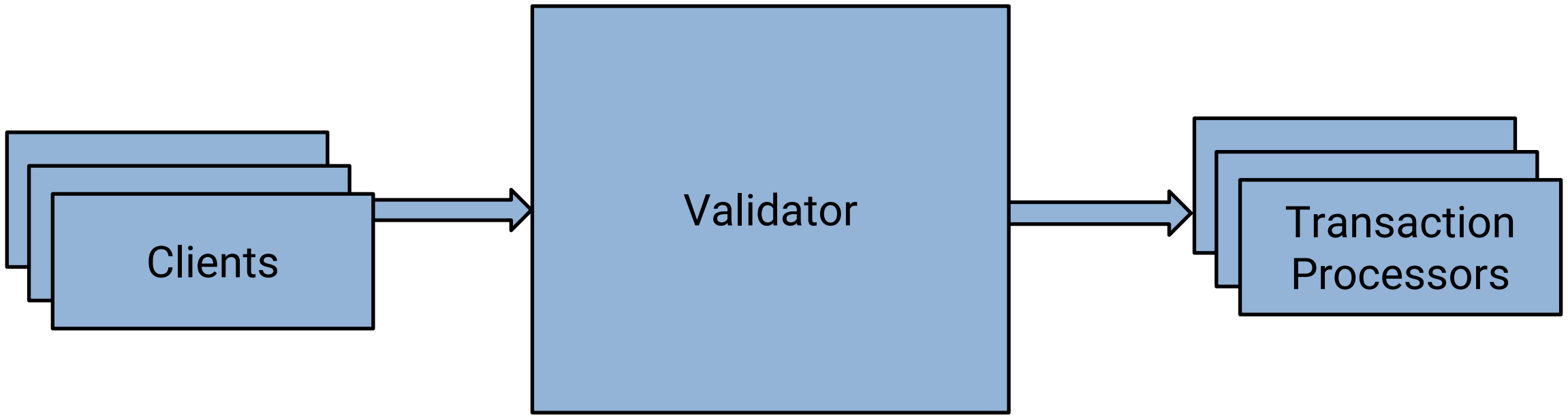
public key

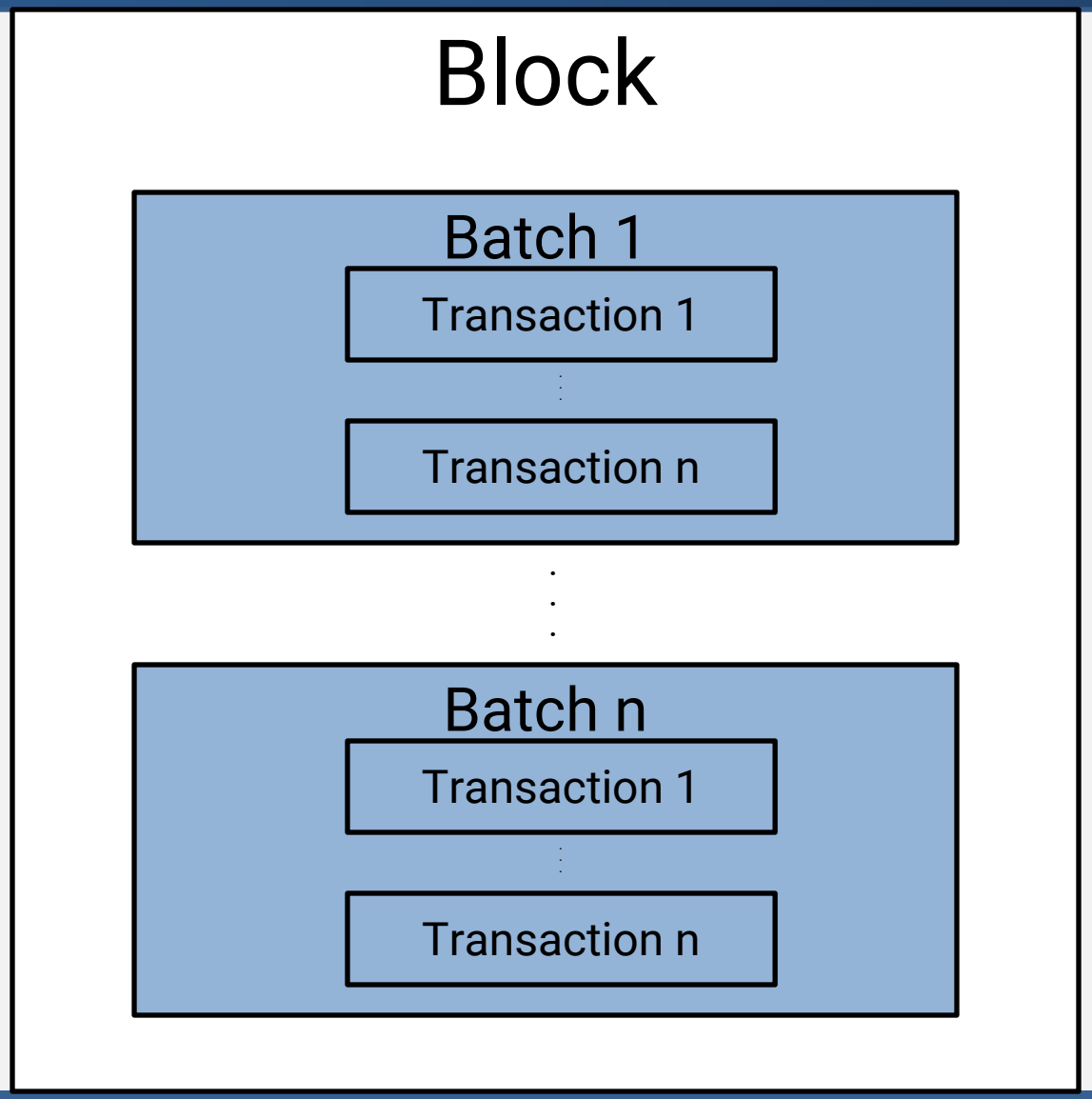
generated from the private key

Signing (Secp256k1 -ECDSA)

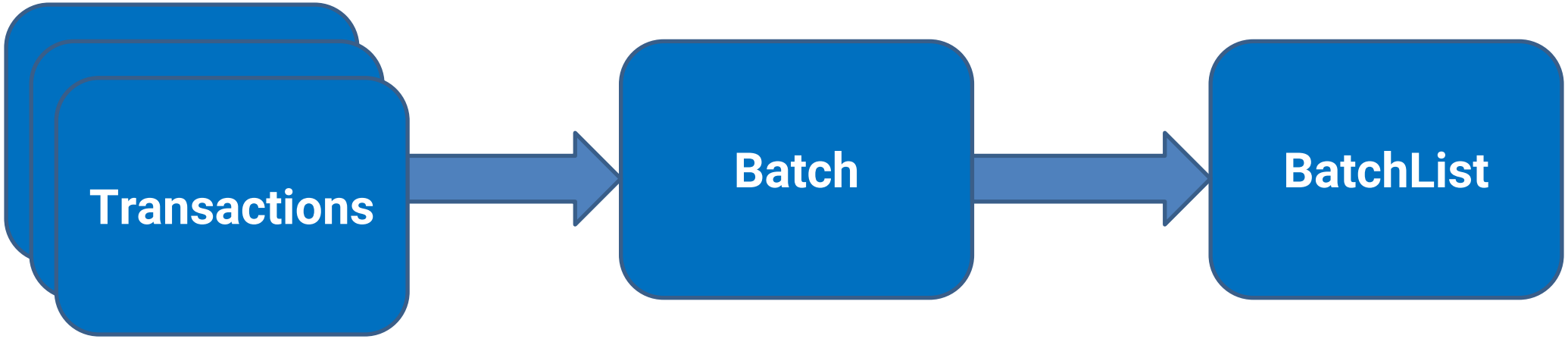








Client



Transaction

TransactionHeader

TransactionHeader (serialized)
batcherPublicKey : string
SignerPublicKey : string
familyName : string
familyVersion : string
Inputs : [string]
Outputs : [string]
dependencies : [string]
nonce : string
Payloadsha512 string

Batcher public key

Signer public key

Family name

Family version

Inputs

Outputs

Dependencies

Nonce

Payload sha512

https://sawtooth.hyperledger.org/docs/core/releases/1.0/javascript_sdk/signing/index.html

Transaction

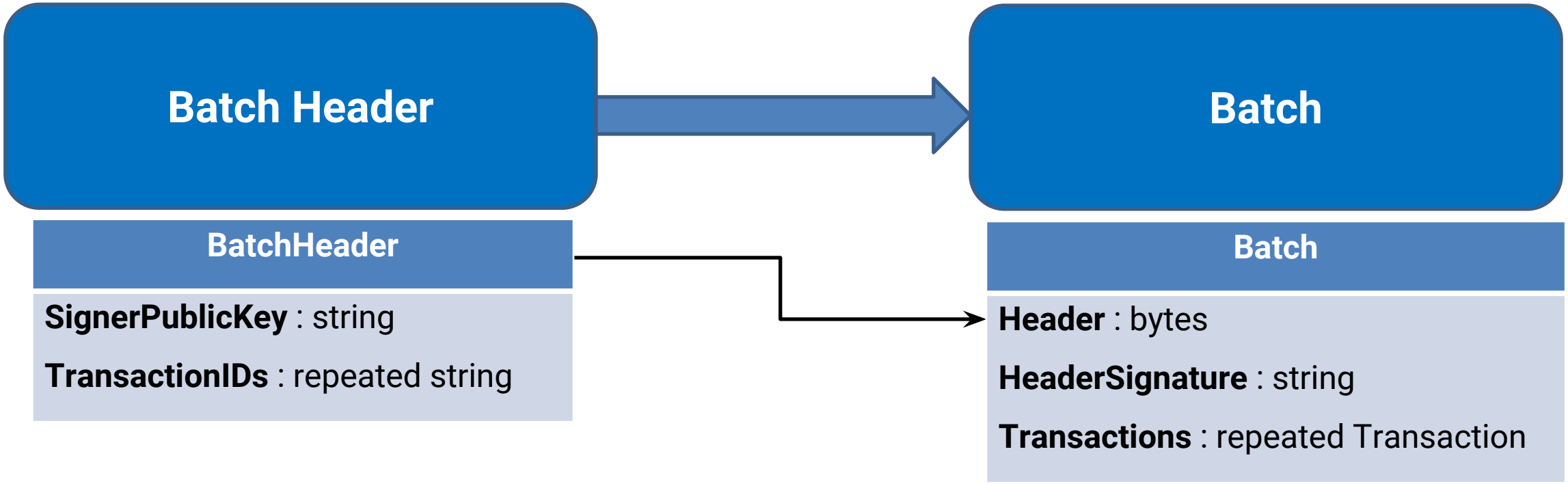
Transaction
Header bytes
headerSignature string
Payload bytes

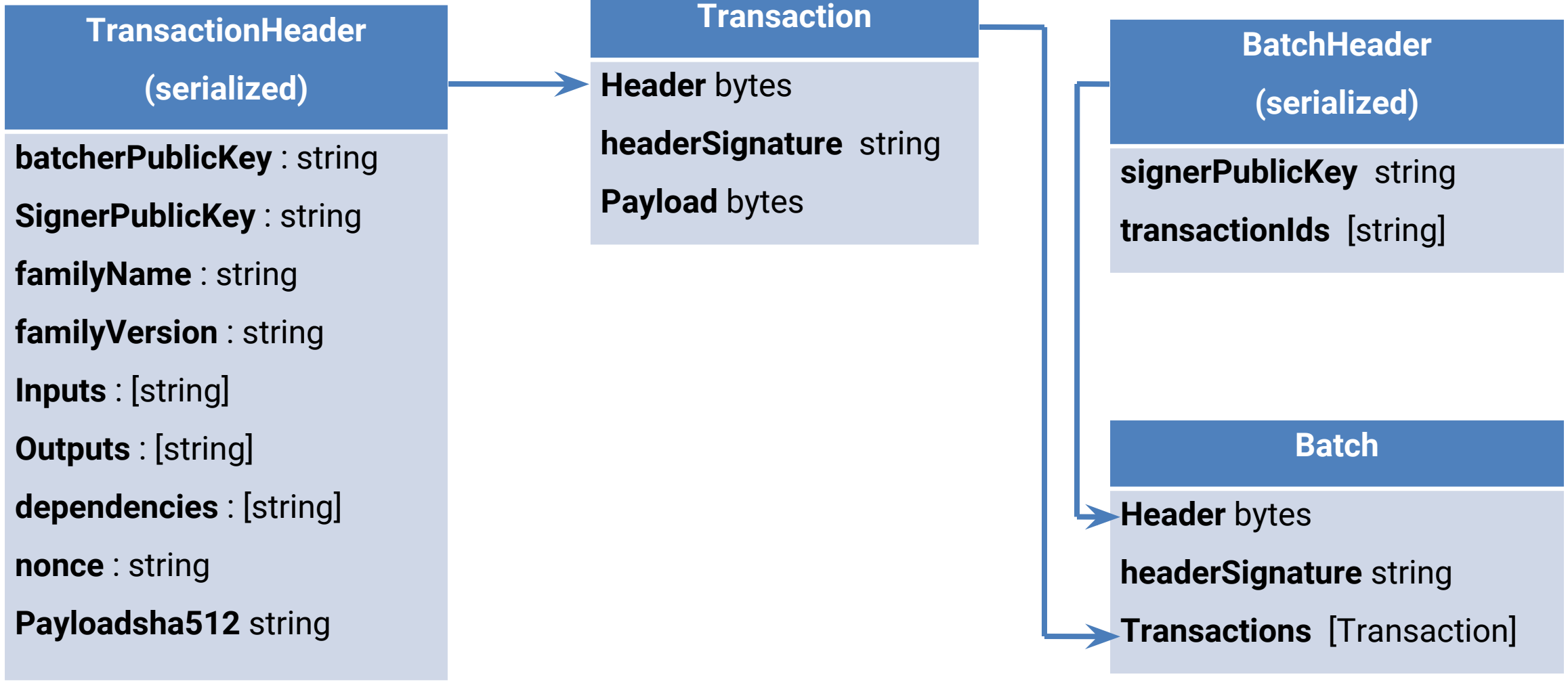
Header

Header signature

Payload

Batch





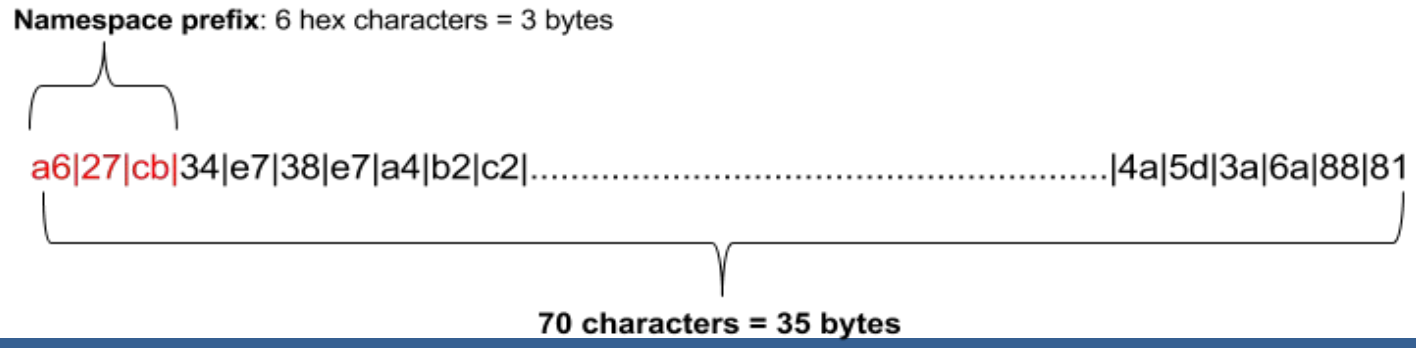
State

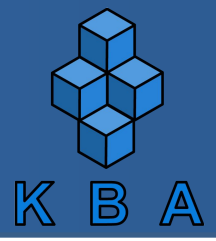
- State is where you can keep application specific data.
- State root hash is added in the block to make sure all validators have same global state
- State store works as a key/value mapping, where key is a 35 byte address and value is your application data

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

Addressing scheme

- By convention, the first three bytes of address is formed by hashing the transaction family name (namespace)
  - That means all 35 byte address whose first three bytes is the hash a family name belongs to that transaction family
  - ie in effect a transaction family will have 2^(32*8) address locations available in the state
- How the addresses are split within a transaction family namespace is upto the application
- Sawtooth addresses are 70 hex characters long (35 bytes)





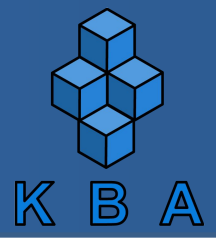
Serialization and Protobuf

Protocol Buffers

Protocol buffers are a language-neutral, platform-neutral extensible mechanism for serializing structured data.

.proto files

Stored inside: sawtooth-core/protos



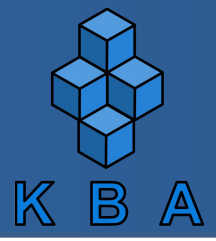
Protobuf

Message.encode()

encodes a **message instance** or valid **plain JavaScript object**.

Message.create()

creates a new **message instance** from a set of properties that satisfy the requirements of a valid message.



Available sdks

	Client Signing			Transaction Processor			State Delta		
	Complete?	Stable API?	Maturity	Complete?	Stable API?	Maturity	Complete?	Stable API?	Maturity
Python	✓	✓	1	✓	✓	1	✓	✓	1
Go	✓	✓	1	✓	✓	1	✓	✓	1
JavaScript	✓	✓	1	✓	✓	2	✓	✓	2
Rust	✓		1	✓		1	✓	✓	1
Java			3			3			3
C++			3			3			3

THANK YOU