Using Factom

The Factom API
The Factom Application Programming Interface

In the first pre-Alpha for Factom, the following web service calls will be supported by a centralized Factom server. When we deploy the decentralized server, these calls will be supported either by web service calls to a local full Factom node, or through messages submitted to node.

The API version is the first parameter to all calls.

SubmitChain

Pay for a new Chain, and specify the first Entry.

Parameters:

- version int (2 Bytes)
- hash of the ChainID
- hash of the ChainID concatenated with the First Entry hash
- First Entry hash

Discussion:

The Submission of a new Chain ensures that the user creating a Chain will have the right to specify the first Entry in the Chain. This allows a Chain to be self verifying, since the creator (who logically has the right to define the rules for the Chain) can document the auditing of the Chain.

The Submission of a Chain creation request does not reveal the Chain Name, the ChainID, nor the contents of the Entry. However, the ChainID is a revealable secret (in that once presented with the ChainID, the protocol can hash the ChainID and match it to the request to create the Chain.

Furthermore, knowledge of the ChainID is also provable. Once the Chain ID is revealed, it can be concatenated with the EntryID and hashed, which will match the second parameter. This proves the submitter *knew the ChainID* at the time of the submission.

Now a call to add the Entry can be validated, the Chain created, and the Entry installed as the first Entry in the Chain. Any attacker would have to have known the ChainID prior to the proper owner from revealing the ChainID.

CreateChain

Complete the creation of a Chain. This call reveals the name of the Chain from which the ChainID can be computed, and the Entry. The Chain cannot yet exist, or this call will be rejected. Further, the hash of the ChainID and the Entry hash must match the first unexpired submit to create the Chain.

Paramters:

- version int (2 Bytes)
- Name [ ] [ ] byte (2 Bytes to indicate the number of path levels)
- Entry (see Commitentry for parameters)
- Signature (of EntryHash)
    - publicKey [ ] byte
    - r [ ] byte
    - s [ ] byte

Discussion:

The name must be submitted so that Names can be converted to ChainIDs (which the rest of Factom uses to manage Chains. The name is created as a “Meta Entry” followed by the First Entry in the Chain.

There is no need for a signature, since the hash of the Entry must match the hash in the SubmitCreateChain call. It doesn’t matter who provides the proper information, as the hash of the Entry proves the Entry to be the proper Entry.

SubmitEntry

Pay for a new Entry to some Chain in Factom. The actual Chain is specified by the Entry,

Paramters:

- version int
- Hash of the Entry
- Number of Entries (1 byte)
- Signature (of EntryHash)
    - publicKey [ ] byte
    - r [ ] byte
    - s [ ] byte

Discussion:

The hash of the Entry to be added to Factom is provided, along with the signature of a public key that has the appropriate credit to cover the number of Entries to be used (1 Entry = 1024 bytes)

The actual ChainID must be specified by the Entry when it is submitted. This approach commits Factom to record the Entry prior to knowing what Chain the Entry will be added to, and prior to knowing the contents of the Entry. This limits the ability of the network to censor a particular user or Chain.

Commitentry

Supply the actual Entry. The Entry will be constructed from the parameters supplied, and hash of the Entry must match a hash supplied by a previous SubmitEntry

Paramters:

- ChainID [32] byte
- ExternalHash [ ] [ ] byte (2 Bytes to indicate the number of external hashes)
- Data [ ] byte

Discussion:

Since a previous submit has already committed the Factom Servers to recording the Entry, censorship of a Chain or by Entry contents is not possible without detection by the network of users. Since users of Factom can use aggregators to submit Entries and/or multiple public keys to submit Entries, the public key used to do the SubmitEntry does not necessarily identify the user, nor the contents.

FactomTransaction

A transaction on the Factom Coin Chain(s)

Parameters:

- version int
- input [ ] Input
  - EntryHash [ ] byte
  - Output index
- output [ ] Output
  - amount int64
  - FactomAddress [ ] byte
- Signature (of input and output arrays)
  - publicKey [ ] byte
  - r [ ] byte

- s [ ] byte

Discussion:

The full design of the Factom Coin is still in progress. The Factom Token needs to have the common functionality of a cryptocurrency. Features such as multisig must be supported. It is less clear if we need a script engine to process transaction types.

This transaction type executes a single transfer from a set of inputs to a set of outputs.

PurchaseEntries

Return an amount of Factom to the protocol in exchange for allocating to a specified public address the right to submit the specified number of Entries.

Parameters:

- version int
- input [ ] Input
  - EntryHash [ ] byte
  - Output index
- output [ ] Output
  - amount int64
  - FactomAddress [ ] byte
- Signature (of input and output arrays)
  - publicKey [ ] byte
  - r [ ] byte
  - s [ ] byte

Discussion:

The separation of paying for transactions from the actual submission of transactions makes the protocol more secure against censorship, reduces the overhead of validation of Entries, and enables a business model where Factoids are issued to Factom servers and auditors as a reward for securing the protocol, and those entities can sell access to the Factom application to users, completing the cycle.