Slán-Chain™:

Blockchain, Cryptocurrency, Storage and Contract Platform

Design Notes and Background

(the tl;dr version)

Brian McMillin

Table of Contents

Preface

This document is a set of notes and analyses of a possible Blockchain Architecture that addresses many of the shortcomings of the present cryptocurrency and blockchain implementations. In addition, proposals are made to offer valuable services at negligible cost in a form that can scale into the foreseeable future.

For discussion purposes, the terms Slán-Chain™ and Slán-Coin™ are used in lieu of a professionally researched and registered international trademark. I think that Gaelic is both sufficiently well known and obscure to afford a suitably unique word. **Slán means Strong.**

Comments and well-reasoned criticisms are welcome.

> **And it's this type of integrity, this kind of care not to fool yourself, that is missing to a large extent in much of the research in cargo cult science.**
>
> — Richard Feynman

It is extremely difficult to work through an endeavor of this magnitude without risking fooling yourself. And, in all cases, nature will prevail over delusions.

> **Then the Gods of the Market tumbled, and their smooth-tongued wizards withdrew**
> **And the hearts of the meanest were humbled and began to believe it was true**
> **That All is not Gold that Glitters, and Two and Two make Four**
> **And the Gods of the Copybook Headings limped up to explain it once more.**
>
> — Rudyard Kipling

Difficult Problems for Cryptographic Smart Contracts

Certain problems are difficult to solve economically using the current implementations of Blockchain technology and cryptographic smart contracts. These difficult problems can be divided into several classes based on the type of difficulty involved.

1. Contracts that would involve too many small transactions to be economically viable.
2. Contracts that would involve computations that are infeasible to verify.
3. Contracts that must reveal a result only after a certain time or event.
4. Contracts that must generate an anonymous token after verifying a condition.
5. Contracts that must allow proxy signatures on transaction inputs.
6. Multiple token classes - Token quantity splits.
7. Proof-of-Stake mining and secure allocation of earnings.
8. Contracts that require different behavior based on persistent global states.
9. Generation and use of secure Pseudo-Random token descriptors
10. Any type of database operation, especially non-indexed (sequential search) database lookups
11. Database of all Contract Participants (for triggered revert/payout, etc.)

On Distributed Consensus for Very High Transaction Volumes

Expect peer-to-peer flood of all candidate transactions across the network.

Minimize unnecessary network traffic devoted to replication of transactions.

Establish a distributed consensus to pre-select block-approving candidates from a pool of bidding miners.

Pay successful miners as well as *ommers*.

Ensure that all participating peers execute and validate all transactions before forwarding on network.

Allow essentially unlimited block size as scaling mechanism for high transaction speed. Typical of blockchain architectures we use a near constant block creation rate, but variable block sizes.

Use fixed payment rate for bytes on the chain to balance transaction volume with cost to use.

Choice of persistent “Account” concept vs. one-time HD Wallet addresses.

Blockchain Systems typically try to establish costs for

1. Telecom / network costs for distributing the transactions
2. Disk cost for maintaining the blockchain copies
3. Memory cost for maintaining non-volatile account data (if any)
4. Memory cost for maintaining Merkle Tree lookup for Account references
5. CPU costs for accessing and running smart contracts

There is always a trade-off between sender-pays vs. receiver-pays. This needs to be clearly established. Also appropriate costs for extravagant CPU utilization for view-only functions.

Establish **minimum** allowed expectations for

1. transaction volume,
2. number of peer servers,
3. number of concurrent *ommer* selectors

List chain control parameters that are subject to automatic dynamic control and consensus modification.

All of the extreme bitwise optimizations for transaction size become moot if the blockchain is intended to be used for searchable, ad-hoc data and there is sufficient financial incentive to pay for it.

On Voting System Requirements

Major issues in all voting systems are transparency, security, accessibility and audit-ability.

How can we create a contract for straight Up/Down Voting?

1. Who can vote? Ballot generation for each Voter.
2. Vote Options: Veto, Against, Abstain, For, Ratify
3. Max One vote per Ballot. Change your mind, last submitted Ballot counts
4. Voter can Review Ballot Selection and verify correct vote was cast
5. Predefine What Is the Quorum
6. Predefine Approval Threshold, either absolute or by a margin
7. Maintain secrecy of ballots cast
8. Reveal final Tally only after deadline

How can we verify Voter Eligibility before generating a Ballot for her.

Do we need to be able to revoke an issued ballot during the election period?

What about Proxies?

What about selection from multiple candidates?

How to handle Top M of N voting. Each Voter gets to vote X times.

How to handle weighted voting where vote is proportional to the value of a stake. Must be stake value at a specified time, i.e. the end of the last quarter or start of voting.

Fees for Perpetual Storage Viewed as an Annuity

We need to incentivize long-term storage of large data sets.

Storage must be: **secure**, **redundant** and **available**

When the data is added to the Store:

> collect a **fee** proportional to the amount of data
>
> use the fee to purchase an **annuity**
>
> the annuity periodically **pays** Storage/Network providers

We must allow proportional claims on the annuity payout

- By any or all Storage/Network Providers
- After providing the service

We must verify the level of service, probably using continuous dynamic audit.

Network architecture should include:

- Continuous background sweep of the Distributed, Peer-to-Peer Block Store
- Confirm the identity (payment credentials) of the active Providers
- Prevent block forgeries or audit spoofing by verifying all Blocks
- Establish maximum background level of network traffic for replication and audit
- Rely on statistical validation of Block and Provider existence

The performance statistics should be placed on the blockchain.

Make payments – upon request – to Provider via a Contract.

Actuarial analysis is required to ensure the real-time fees and payouts are properly adjusted based on:

- Transaction fee per GB Storage
- Growth Rate of the annuity cryptocurrency account
- Exchange rate of cryptocurrency for actual services rendered
- Value of Storage and Network services in the world at large
- Potential Interest or Earnings from the Annuity Cryptocurrency Account

Concerning Access to the Distributed Blockchain Storage

Previous Blockchain implementations have assumed that the Blockchain would be small enough to be replicated to all nodes that needed to perform audits or analyses.

The true Peer-to-Peer Distributed Blockchain has no size limitation.

It is therefore necessary to collect Access Fees from third parties and distribute those fees equitably among the providers of Storage and Networking Services.

Storage and Network Service Providers (SNSP) are expected to handle four general types of requests:

1. Handling the Distribution of Candidate Transactions to Active Block Producers (Miners)
2. Adding and Distributing New Blocks among Active Block Producers (Miners)
3. Performing Continuous Background Audit and Replication (CBAR)
4. Handling the fulfillment of Directed Data Requests from dApps and Wallets

Candidate Transactions pay fees – proportional to their size - to the network and Producers when they are added to a Produced Block.

Continuous Background Audit and Replication (CBAR) operations ensure the equitable distribution of fees and the integrity of the Storage and Networking.

Directed Data Requests from outside the Blockchain Core Functionality are made in the form of Transactions with a specialized fee structure.

Instead of the normal “fee per GB Stored” we use a “fee per GB Retrieved”.

This is made practical by the structure of the underlying distributed Block Store which uses fixed-size Segments and allows direct random access to specific Segments from anywhere in the Distributed Store.

These Directed Data Requests may be viewed as micro-transactions with (normally) extremely small fees.

The scope of the requested data will be known at the time the request is made so the fee is deterministic and can be paid up-front by the requester.

The server that presents the request to the network will appear to the network as an ordinary Storage node and the actions required to retrieve the selected Segments will fall in among the ordinary network traffic between Storage nodes.

It is normally expected that nodes that handle incoming Directed Data Requests will also act as Storage nodes since the hardware and networking requirements are the same and Storage nodes can participate in the Continuous Background Audit and Replication (CBAR) compensation structure.

Fees paid for Directed Data Requests join the pool of funds available for distribution as CBAR payments.

Requests for Directed Data are verified against the Requester's balance as a normal part of the micro-transaction operation and thus inherently limit the possibilities for abuse.

Concerning the Physical Storage of the Blockchain

The Storage of the Blockchain must be:

- Distributed – among many Service Providers
- Redundant – every portion of the Blockchain must exist in multiple locations
- Fault Tolerant – Error Correction technology ensures against loss of blocks or connectivity
- Indexable – Specific Segments of the Blockchain must be randomly accessible
- Searchable – Queries concerning the Content of the Blockchain must be fast and feasible
- Permanent – Blocks and Segments of the Blockchain must be immutable
- Unlimited – There should be no arbitrary limit to the size of the Blockchain.

In addition, there must be a mechanism for auditing the performance of **Storage and Network Service Providers (SNSPs)** and ensuring fair and appropriate compensation for those Services.

This can be accomplished by breaking the Blockchain into fixed-size Segments for access, storage and transport. Segments become the redundant, distributed element among multiple SNSPs.

New Blocks are added to the end of the Blockchain, broken into Segments as required, and distributed among SNSPs. The addition of new Segments also entails the addition of Index and Error Correction Segments used in the access and fault recovery processes.

Once created, Segments are a permanent, unchanging feature of the Blockchain and are identified by their unique Hash. It is this Hash value that is used by any server to request a specific Segment from the Blockchain Network.

Requests for particular Segments across the Network are handled as micro-transactions placed onto the Blockchain.

By making the Networking Access requests look like ordinary Transactions, the payment for services, audit of performance and quality, and background replication become an inherent feature of the operation of the Blockchain.

The specific Transactions that request Segments from the Blockchain may have specific features that facilitate access to sets of Segments or data related to different aspects of the Blockchain – the important thing being that there will be a fixed upper limit to the amount of data that will be retrieved by any request, and that it will be paid for appropriately by the requester.

The Segment-based underlying storage mechanism should be thought of as the Storage and Network Protocol layer, underlying the actual Blockchain and Transaction layer.

The peer-to-peer networking protocol supports the secure propagation of Segments, generation of performance Audit records and – importantly – the timely sharing of Candidate Transactions among Active Block Producers to allow the Candidate Transactions to be properly incorporated into new Blocks.

Active Block Producers announce their requests for and availability of Candidate Transactions by simply publishing signed, timestamped Bloom Filters to their peers and sending and receiving Candidate Transaction data.

Concerning the Scalability of the Blockchain

The Bitcoin Blockchain is currently about 165 GB and can grow at a maximum of 52 GB per year.

Much has been written concerning "Blockchain Bloat" and how to prevent it.

"It is a Capital Mistake to Optimize Too Soon."

It should be considered axiomatic that the uses, data scope and transaction rate of the Blockchain should be expected to grow.

Placing *a priori* limitations on the design – in an attempt to "save a few bytes" – is fatally shortsighted.

Some solutions attempt to place large data sets in "off blockchain" storage.

Claims that these solutions still have the features of Blockchain are fundamentally flawed.

1. It is not possible for the actual contents of a Blockchain Transaction to be searched by third-parties if it requires off-chain access.
2. There is no assurance of the long-term availability of off-chain data.
3. The integrity of off-chain data is dependent on the arbitrary backup and security policies of unknown providers.
4. The accessibility and bandwidth available from off-chain providers is unknowable.
5. The enforceability of Contracts involving off-chain data can be arbitrarily compromised.

Therefore, **ALL** data relating to Blockchain Transactions and Contracts **MUST** be stored on the Blockchain and make use of the Blockchain's own Distributed Search and Distributed Consensus mechanisms.

Further, all efforts to reduce the number or scope of Transactions on the Blockchain are inappropriate and doomed to failure.

The fee structure associated with Blockchain Transactions should be fundamental to its growth:

1. Charge adjustable fees sufficient to support the long-term operation of the Blockchain Infrastructure,
2. Allow Market Forces to adjust the scope of Transaction size and frequency based on their value to the dApps, Wallets or other users of the Blockchain that generate the Transactions.

Concerning Turing Completeness in Blockchain VMs

The Ethereum White-paper claims that there is no penalty for providing Turing Completeness in the Virtual Machine (VM).

This claim assumes:

- Use a Gas Limit to ensure timely termination
- Requirement for doing CALL instructions to other contracts
- No downside to having program loops

HOWEVER, it turns out that:

1. Miners do not actually need any form of non-monotonic execution sequences

    Conditionals are required, but using forward branches only.

2. Miners do not actually need to execute contract code themselves

We can shift the burden of Contract evaluation

FROM a Consensus of **Trusted Miners**

TO a Consensus of **Trusted Judiciary**.

Contract Code can be written in any convenient programming language.

Payments to Judges may be made under arbitrary terms – even “off chain”, or private currency.

Arbitrarily huge data sets may be involved in Transaction I/O.

It is incumbent on the User to ensure that the large data sets are available on the blockchain in a timely manner when presented to the Judiciary and Block Producers in time to be validated.

We need to be able to revise or upgrade Contracts.

This implies some type of Table Lookup or Redirection step within the VM.

We have to be careful to prevent Recursion in the VM – by design.

Maybe limit code to ONE CALL (Direct or Indirect). Period.

No Loops or Functions or Libraries. All VM code is specified up-front and in-line.

Let the Judiciary handle all complex Contracts, including:

- Long-winded computations.
- Database Lookups.
- Anything that requires Turing Completeness.

Hard Rule: Judiciary Evaluation may NOT access ANY off-chain data.

What is a Blockchain Miner?

1. Verifies all blocks in the current chain
2. Verifies all Proposed Transactions
3. Selects Proposed Transactions to build a Proposed Block
4. Confers with other Miners to reach a consensus on next Block
    1. Bitcoin uses Proof-of-Work
    2. Dash uses Buy-in for voting privileges
    3. IOTA mines everything and adds to a web of adjacent points
5. Receives payment for Block plus Commissions and Tips
6. Commissions are based on transaction length, memory usage

I propose the deterministic selection of the next Miner from a pool of Miner Candidates that have paid a per-round ante. Think of the ante as a bonded contract to perform a specific service at a given time.

For each Block Interval, the next winning Miner must produce a valid Block on time or lose his ante.

Bids for future slots will be taken at periodic intervals.

Multiple winners will be selected from the candidate pool and deterministically assigned future Block numbers to Produce.

Overlapping lists of bid winners cause multiple Miners to compete to accurately Produce every block.

Multiple Miners that compete for a particular block are *ommers* and each gets a reward for Producing valid complementary blocks. The blocks will not be identical since they are Produced and Signed by different Producers, but they will contain identical transaction content.

The "Winning" block is deterministically selected from the set of valid *ommers* and receives an additional reward.

An *ommer* that signs a mismatched Block, or who fails to Produce a block at all, forfeits his ante (Bond).

Bids are all made ahead of time. A bid is a transaction added to a specifically numbered block and the bids close before any selection is made. The selection of winning bids involves XOR-ing bid hashes with a block hash that was created after the closing of the bids. The N lowest numerical values of the XOR results are used to select and order winning bidders. Bidders that are not Winners (greater than N in sequence) are not used and have their ante immediately refunded.

All blocks are built by Producers deterministically and are verified by all other Producers.

Producers must have no free will in the selection or organization of Transactions into Blocks.

The lack of free-will allows agreement among Producers and ensures that Block Hashes will be truly random and cannot be manipulated by Users, Producers or Miners.

Blockchain Incentives

Many operations of a blockchain ecosystem can be dictated explicitly by the rules and operating parameters of the programs that implement the technology.

Other aspects require incentives to achieve desired behavior.

Typically the use of a financial profit motive can provide the necessary incentive.

Every participant in the blockchain ecosystem needs to be able to expect the possibility of a positive return for his contribution.

Careful study of the incentive structure must be made to ensure that no rewards are present for undesired behavior.

This is especially true in the case of off-chain rewards that can accrue to bad actors who cause outages or corruption of services.

We need to look at:

1. Incentive to Ante Up and become a Bonded Producer
2. Incentive for Honesty
3. Penalty for trying to Buy the Producer Bidding Pool
4. Penalty for Collusion (or prevent it altogether)

Producers profit from

1. New Block Bounty (maybe as an Asymptote like Bitcoin)
2. Transaction Commission - What fees should be required
3. Forfeited Bonds from misbehaved Producers

Blockchain Requirements

How can we combine the principle features of Blockchain:

1. Distributed Consensus, and
2. Immutable Historical Ledger

with:

1. Unlimited Data Store
2. High Transaction Volume
3. Small Data Descriptors (i.e. Data Handles)
4. Fast, Deterministic Programmability

We must ensure that

Data is only stored once in a chain

Data Store is in Immutable Creation Order

Data is accessible through indexes to variable-sized blocks

Data descriptors within individual Transactions are linked to (single) instance of actual data

Common operations must be able to be accomplished as rapidly as possible in a distributed environment:

Transactions are grouped into Blocks which are verified by the distributed consensus mechanism

B-Tree and linked lists support rapid search for all references to specific data elements

B-Tree is periodically added to the blockchain store to allow rapid restart

The foundation of the storage and network transport is based on:

Blocks can span fixed-sized Segments

Segments are identified by hashes

Segment Order is described by Directory Segments

There exist Error Correction (ECC) Segments to provide fault tolerance

On Replication of Extremely Large Datasets

The “Blockchain” needs to contain ALL data required to establish the initial conditions, progress, and final dispensation of contractual obligations.

Unlike the Bitcoin situation which embodies a simple crypto-currency payment model, smart contracts covering business and personal information may require large elements of rarely-used Bulk Data.

Bulk Data must remain permanently available for audit or settlement purposes.

Off-chain storage is not acceptable due to lack of guarantees relating to custodianship and availability, which would fundamentally undermine the value of smart contracts in the first place.

Large elements of rarely-used Bulk Data must be interspersed with very small transaction elements that may be needed with extraordinary frequency.

An example of a rather large element would be a copy of the current Transaction-List B-tree.

During normal operation it is expected that periodic copies of the complete B-Tree are appended to the data store in order to enable rapid access by new Nodes that may join the blockchain.

In order to prevent unacceptably asymmetric growth of the B-Tree used to look up Transactions in the blockchain it is occasionally necessary to perform a re-leveling operation.

Re-leveling operations on the B-Tree occur as needed and will always occur at the same time throughout the network since they are caused by the addition of identical transactions.

These re-leveling operations require minimal processor time but do result in the rewriting of many links within the B-Tree.

When a new Producer or Server joins the network it will be able to request the current blockchain backwards to the most recently added B-Tree image.

The new Server then updates the B-Tree using the sequence of recent Transactions to create the correct, current B-Tree and therefore an internal state that matches every other Server in the network.

There may be a rule that ensures that current copies of the B-Tree are appended frequently enough to make the startup time for these new Servers reasonable.

Like many large data elements on the blockchain, the Transaction-List B-Tree in the example might be accessed a few hundred times before a more recent instance supersedes it.

Older B-Tree Elements are purely archival and might never be needed again – HOWEVER – they ensure that time-travel against a huge blockchain can work in reasonable time.

Storing and Sharing a Blockchain

There is a difference between the Segment Store and the Blockchain Logical Address Space.

The Segment Store is a collection of fixed-size segments, each identified by the hash of their contents.

Every segment maps to the logical addresses of a portion of the Blockchain.

The segments are ordered to provide an indexable list that allows locating particular data within segments.

Segments are added as new data is created and added to the blockchain.

Segment directories are segments that occur periodically within the sequence of segments and list the Segment IDs (hashes) of each of the previous Segments.

Segment directories also list the IDs of Error Correcting (ECC) Segments that are periodically added to the Segment Store to provide overlapping fault tolerance.

Tree Directory Segments may also be added to allow rapid access to particular areas of the Segment Store (a therefore the Blockchain) without the need to trace the entire chain of Segment Directories.

We have the following different types of segments in the Segment Store:

1. Blockchain Address Space - containing Blockchain Data.
2. Segment Directory – Linked List of Segment IDs
3. ECC Segments – Periodic Error Correction data for fault tolerance within the sequence of Segments
4. Segment Directory Tree – Tree structure for rapid access to Segment Directories.

Segments form a continuous list of storage regions that can be indexed by number.

These segment numbers can be used to determine algorithmically which of the segment types is referenced.

Looking up the particular Segment ID (i.e., the hash of the contents of the particular Segment) allows the actual data Segment to be requested from the network.

Blockchain Logical Addresses form a unified, contiguous sequential are of memory spread across Segments.

Segment Directories can be used to find the ID (hash) of a Segment containing a particular Logical Address.

The distributed network protocol allows a Node or Peer to request sets of Segments using only their IDs.

Requests for IDs propagate, as do Responses, sending matching Segments back to Requestors.

This tends to make more recently used or popular Segments more widely and rapidly available.

How can we Improve the Bitcoin Blockchain?

The Bitcoin Mistake is allowing Miners any **free will** at all.

Bitcoin allows each individual Miner to choose Candidate transactions for each block independently, and to order those transactions within the block at will.

The restricted size of a Block causes the miners to preferentially select high-fee transactions and delay low-fee ones for possibly unlimited periods.

Low-fee transactions might not be mined at all, even when blocks are not full.

This means that it is not possible to positively assure that a candidate transaction, once submitted, will EVER be added to the blockchain, or that, after a number of blocks, that a particular transaction will NEVER be added.

We can eliminate free will by

1. requiring ALL valid Candidate Transactions to be used in the next block mined, and
2. placing them into the block in a deterministic order.

To accomplish this, we force all Candidate Transactions to request to be included in a specific Block Number.

Candidate Transactions will either be accepted into the requested block or discarded.

Discarded Candidate Transactions must be resubmitted and target a subsequent Block Number.

This eliminates any paid prioritization of Transactions.

It is expected that Per-Transaction fees are based solely on the size (number of bytes) of the Transaction data.

This also eliminates arbitrary confirmation delays across transactions.

An added benefit is a reduction in peer-to-peer bandwidth utilization since shared transaction candidates need to traverse the network only once and nodes can then accurately build the corresponding blocks locally - without the need to communicate the final blocks themselves.

Blocks during high-volume periods may become arbitrarily large.

Fees may be adjusted - perhaps based on the running average of Block length.

Fee adjustments must be deterministic and predictable over reasonable time scales in order to have the intended effect of load leveling on the network.

How can we Improve the Ethereum Blockchain?

1. The use of a Turing-Complete engine for smart contract programmability means that the operation of the blockchain may be non-deterministic.

Non-determinism means that the operation of particular contracts cannot be guaranteed at the time of their creation and the participants cannot know with absolute certainty what they are agreeing to.

Obscuring the operation and effects of a Contract is exactly the opposite of behavior that should be expected of a smart contract.

2. The use of Accounts to store a history of cryptocurrency transactions means that those transactions are not private and that the web of Pay and Spend actions associated with the Accounts are public knowledge

The use of Accounts means that the cryptographic key required to access an Account for Spending will typically be used more than once, thus violating a basic tenet of cryptographic security.

The use of unified Accounts also defeats the security of single-use identifiers provided by Hierarchial Deterministic (HD) Wallets.

3. Allowing Gas Limits to affect the operation – or even the validity – of transactions makes it essentially impossible to determine the effects of a Contract at the time of its creation or instantiation.

Try separating the Smart Contract Evaluation from the Mining Operations, possibly using trusted signatures.

Use only sequential programming operations to perform validation within actual Transactions.

Allow deterministic pseudo-random operations to select the output destinations of a transaction, thus permitting secure obfuscation of the payment web.

How can we Pay Bulk Storage and Bandwidth Providers?

Blockchain implementations traditionally incorporate the concept of Full Nodes, meaning servers that store complete copies of the blockchain and participate as Miners to validate and add new Blocks to the chain.

Ultimately, we want to implement true blockchains that incorporate huge quantities of bulk data in a fully secure and transparent manner.

This cannot be accomplished by trying to adapt existing file-oriented stores such as the Interplanetary File System (IPFS) because the data items in question bear essentially no relationship to traditional files.

Organizations may want to participate in, and profit from, the storage and communication of bulk data without making the commitment of becoming a Block Producer (Miner).

Conversely, other organizations may choose to become Block Producers and commit to the timely Mining of blocks but find it overwhelming to maintain the traditional physical infrastructure.

Thus, Block Producers could recruit Stake Holders to invest and share in the profits of Block Production, and Networked Storage Providers could recruit Block Producers to subscribe to their services based on promises of security, redundancy and reliability.

The key point here is the separation of Storage and Networking tailored specifically to the needs of Blockchain operations from the Bonded Proof-of-Stake operations of the Block Producers (Miners).

The shared-stake Production described here is a loose parallel to the Bitcoin Mining Pool concept – holders of tokens may invest in “shares” of a Producer Bond Stake and share in the resulting Producer bounty.

There is no requirement that any single “node” or participant in the operation of the Blockchain actually be in possession of a complete copy of the entire blockchain.

Indeed, the essentially random hashes used to identify individual fixed-length Segments of the overall blockchain make it possible for individual Storage nodes to preferentially, deterministically, store random selections from the complete blockchain.

Allowing Nodes to store deterministic subsets of the blockchain can enable more effectively directed, simultaneous network traffic.

The accurate delivery of data Segments is ensured through the ability to verify that the hash of the data within the Segment matches the handle used to retrieve it from the network storage.

Communications reliability will be ensured through independent, parallel connections using multiple Storage/Network providers.

Storage/Network providers are expected to serve not only Block Producers (Miners) but also other clients that need the ability to access the blockchain.

Client blockchain access could be used to create Oracles, Monitors, Analysis Tools, etc. without the need to ever receive and store the actual bulk data or complete blockchain.

Smart Contracts and the Blockchain Judiciary

There is no reason for the evaluation of smart contracts to be handled by block Miners.

All that is really needed is for the miners to validate the signatures on the candidate transactions.

Signatures may include those of members of a trusted judiciary.

The definition of trusted judiciary, and membership in that group, may be stored as elements on the blockchain.

This allows the programmatic execution of the contract code to be handled (and paid for) independently of the very fast processing of blockchain mining operations.

The function of the trusted judiciary is to execute the program code describing a smart contract - on request - and to provide a signed certificate of the results.

The trusted judiciary may become a cottage industry with dynamic price structure similar to SSL Certificate Authorities on the Internet.

As part of the contractual agreement between parties that create the transaction that becomes the smart contract, the parties specify the terms of the Trusted Judiciary that will evaluate the contract.

The name of the Trusted Judiciary is similar to a Jurisdiction in common law.

The name of the Trusted Judiciary is a Handle that implies a set of independent Signing Authorities (think: individual judges or courts).

The terms of the contract specify the voting structure of the judges within the Trusted Judiciary.

This allows the participants in the Contract to specify *a priori* whether the Contract must be agreed to by (for example) a single Judge, a quorum of the Court, Unanimous decision, etc.

This brings the expected 'consensus' concept to the smart contracts, but relieves the real-time blockchain operations of the burden of evaluating (sometimes lengthy) contract code.

Blockchain mining operations simply verify the signatures of the Trusted Judiciary and the terms of the Transaction.

This allows all transactions on the blockchain to use straightforward linear verification scripts, including (if included) verification of the signatures of the Judiciary.

This structure allows the actual membership within the Trusted Judiciary to change with time as new Judges are added or old ones removed in a separate set of transactions on the Blockchain.

There may be many independent Judiciaries on the blockchain, perhaps devoted to evaluating different types of contracts written in different programming languages or devoted to different tasks.

Note that this is a considerably different concept from using an off-chain Oracle as a “Judge-as-a-Service” for dispute resolution as part of existing hybrid smart contracts.

Concerning Persistent Oracles, Events, Triggers and Timers

Oracles act as sources of new information from outside the blockchain that is made available as input to transactions.

Frequently it is necessary that transactions occur automatically under certain conditions or at specific intervals.

It is expected that the services that automate the operation of transactions will be wholly independent of the services that append the transactions to the blockchain and those that distribute the blocks throughout the network.

This automated, persistent operation consumes resources and may be expected to have certain performance and reliability standards.

In order to meet these goals the smart contracts or their beneficiaries should be expected to pay a fee to the provider.

Notably, the provider is performing a service even in the absence of trigger conditions or new transactions being added to the blockchain.

Periodically executing off-chain services should be activated, paid for and terminated by specifically crafted Transactions placed on the blockchain.

The transaction that establishes the request for off-chain services should describe the desired functionality and service level as well as the terms of payment and warranty expectations.

Think the Unix Way

Store values a Printable ASCII Text.

Never use binary values.

Never use Fixed-Width fields - Always use field and value delimiters.

Use Name/Value pairs where possible.

Never depend on the order of fields.

Make data records Human-Readable where possible.

Never assign special meaning to certain numeric values:

Use enumerated text instead

DO NOT OPTIMIZE STORAGE FORMATS

In the long run, saving a few bits will not matter.

In any case, more effective compression will be provided by dedicated Storage and Networking layers.

Concerning Interest-Bearing Cryptocurrency

Gold and Bitcoin are alike in that their only increase or decrease in value is due to Arbitrage — i.e., an exchange with other value representations.

Gold or Bitcoin under a mattress do not create more Gold or Bitcoin.

Loans of Fiat Currency (via Banks, Bonds or Contracts) may yield more Fiat through Arbitrage in the Stock or Bond markets, for example.

Slán-Coin™ is different.

Slán-Coin™ holdings may be invested - via a Bond - in the operation of the Slán-Chain™.

In exchange for this investment, a valuable service will be provided:

Transactions will be accepted, verified and added to the Slán-Chain™.

Fees are collected in order to post Transactions to the Slán-Chain™.

The collected Transaction fees will be apportioned to the Bond holders and thence to the Investors.

This effectively represents on-chain Interest paid for the temporary use of the Slán-Coin™ investment.

Features:

1. No minimum investment
2. Negligible Transaction Fees
3. Slán-Coin™ investments may be aggregated into Pools and used by the operators of Server Nodes.
4. These Server Nodes are the fast, reliable network connections that host the Block Producers.
5. Block Producer Pools may compete to offer different Interest Rates to Investors
6. Everything is on-chain. I.e., no Arbitrage or off-chain Exchanges are involved
7. No hardware, Software or other participation is required of the Investor

Blockchain Comparisons

Satoshi's Way	Brian's Way
Create a Cryptocurrency	Create a Blockchain
	Establish value by Offering Unlimited Storage
	Establish Value by Offering Reliable Networking
	Establish Value by Offering Smart Contracts
Add a Blockchain for Security	Add Cryptocurrency to pay Fees for Services
Create New Bitcoin through Proof-of-Work	Create new Slán-Coin™ through Block Production
	Support On-Chain Investments that pay Interest in Slán-Coin™
Limit Inflation through Asymptote in Block Mining	Limit Inflation through Asymptote in Block Production
Establish External Value through Arbitrage Exchanges	Establish External Value through Arbitrage Exchanges
	Establish On-Chain Escrow to pay Annuity for Long-Term Data Storage
Limit Growth with: Escalating Proof-of-Work, Limited Block Size and Limited Transaction Rate	Do Not Limit Growth in any way
Bitcoin has No Inherent Value	Slán-Coin™ has Inherent Value derived from Storage, Networking and Contract Services
Storage Networking Computation are all Expenses - borne by Participants who are NOT COMPENSATED for their services.	

Concerning Slán-Chain™ Escrow

Normal transfers of Slán-Coin™ require validation of the Amount and Signature of the Payer.

"Escrow Account" is a property of the Chain which has no private key and requires no Signature.

Funds transfer to and from Escrow are handled ONLY by hard-coded rules in the Producer Software.

Initial Slán-Coin™ creation (akin to Mining in Bitcoin) is an Asymptote paid from Escrow.

Slán-Chain™ operation (Bonded Proof-of-Stake) is done through Transactions against Escrow.

Perpetual Data Storage fees are paid into Escrow.

Fees paid into Escrow can be viewed as the purchase of an Annuity.

The continuous background audit of Storage and Network Service Provider (SNSP) performance establishes an Annuity amount that is paid from Escrow to each individual SNSP.

Slán-Chain™: Forward and Backward Linked Blocks

Typical Blockchains establish security and immutability through links in newer Blocks pointing to immutable properties of previous Blocks.

Slán-Chain™ is different:

The Bonded-Proof-of-Stake lottery ensures that everyone will know ahead of time the identities of the possible Signers (Producers) of specific future blocks.

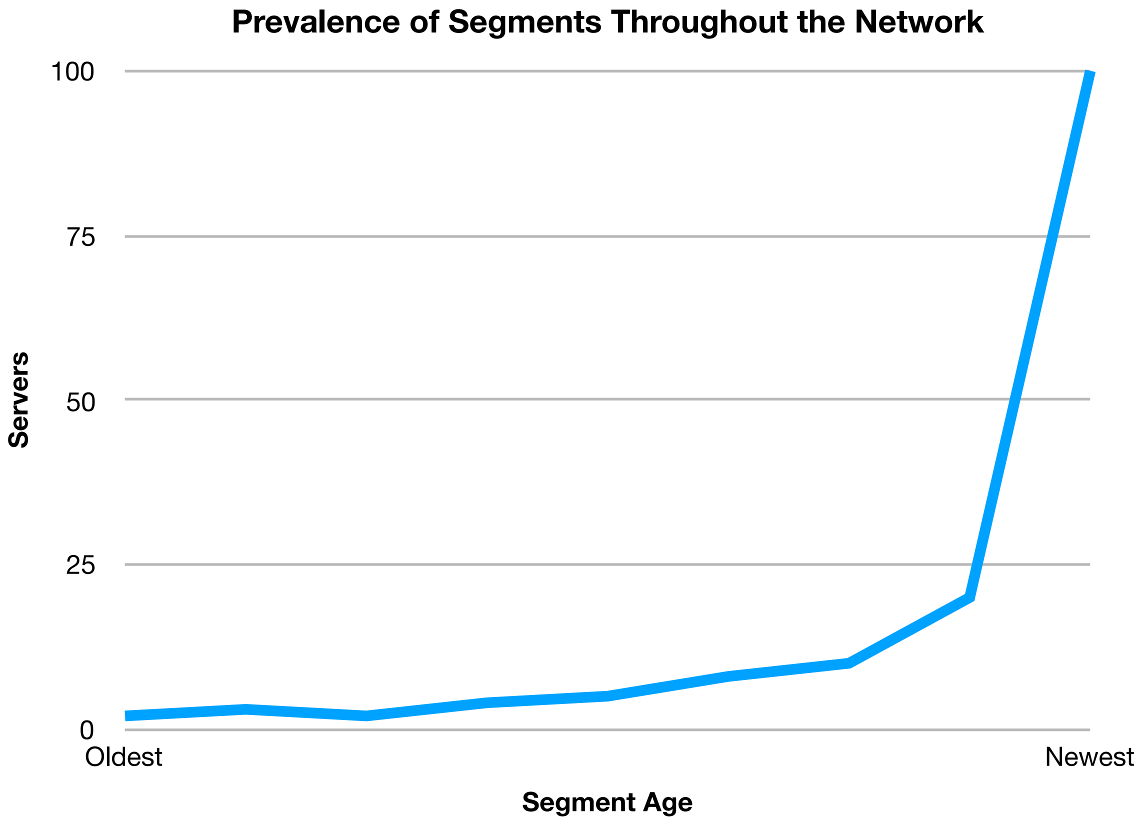
This means that the Slán-Chain™ will have forward and backward signed links identifying the Producer of a given block.

This is in addition to the standard Backward links that establish the order and immutability of the Blocks within the Chain.

These Bond Signatures are verified by the hundreds of cooperating Block Producers participating in the Slán-Chain™.

All operation of the Slán-Chain™ proceeds as rapidly as possible with no wasted computational effort and minimum operating cost.

Storage Segment Replication



Lots of people access recent Segments so they will be present on many Nodes (Servers)

Normally, few people look at older Segments, *however* -

Some old Segments are of continued interest and are widely present.

We use the Storage Network Audit as a background operation that randomly sweeps ALL segments

This ensures that the number of extant copies of any particular Segment *never reaches zero*.

Segments may occasionally appear inaccessible, perhaps due to excessive network delays.

Error Correction Coding (ECC) is automatically invoked as needed during any Storage Access.

Segments will therefore be *recreated* as needed during Audit sweeps and ensure future availability.

Envision a “Holographic Data Store” with essentially no locality of data:

> If you can point to a single Chip or Drive or Data Center and say “that is *the* location of my bits” you are doing it wrong.

How Does the Slán-Chain™ Get Started?

There must be a Root Chain Image that assigns initial Slán-Coin™ to a sufficient number of different Holders to enable the Bond Lottery to establish sufficient Block Producers for operation of the Chain.

It is not necessary that Block Producers use different Server hardware.

Hundreds ofBlock Producers (Bonded participants) may be hosted by a handful of Amazon (AWS) Server instances, for example.

Block Producers are just unique Accounts - with sufficient Slán-Coin™ to put up a Bond - who have actually placed a Bond (Proof-of-Stake) transaction onto the Slán-Chain™.

Features:

1. Discovery Protocol to locate other SNSPs. Preferably Anonymous and unpredictable.
2. Connect to multiple SNSPs to form a randomized web of Network Connections.
3. Request Storage Segment (via a Bloom Filter) to get the current Chain End.
4. Begin sharing Candidate Transactions
5. Lurk on the Network long enough to gain confidence enough to place a Bond for Production.

On startup, Block Production will probably consist almost entirely of Slán-Chain™ Bond Transactions.

The SNSPs will publish the availability of external APIs to allow connections to external parties who wish to initiate Transaction, Store or Retrieve data, create Smart Contracts, etc.

Concerning Slán-Chain™ Failure Modes

Several failure modes are possible and each require thoughtfully designed remedies.

1. Initial Startup. Creating the first connections and consensus.
2. Too few Bond Bids to meet the *ommer* requirement of the Bond Lottery.
3. Network Partition or Chain Fork.

Recovery from Failure

SNSP Discovery Protocol finds the largest Segment and Block Number.

There must be a mechanism to prevent spoofing and establishing a consensus.

Network Time Synchronization occurs and Heartbeat Propagation begins.

Bids are Proposed and Propagated filling all *ommer* slots for upcoming Bond Lotteries.

> The normal Lottery process relies on the hash of a block following the close of bidding to randomize the winners of the lottery. This must be simulated during startup.

Validation of initial Bids consists only of eliminating duplicates and checking static funds availability.

No non-Bid Transactions are allowed until normal chain operations resume.

The first new Transactions are expected to be (as needed) Network Connection (bandwidth) fees between SNSPs as the network connections are (re)established.

Once the Chain is running, new Transactions from outside will be accepted.

Concerning Transactions

There are several specific types of Transactions that are supported by the Slán-Chain™.

1. Cryptocurrency Payment with optional Data Payload.
2. Third-party Request for Selected Data from the Blockchain.
3. Performance Audit of Storage and Network Service Provider (SNSP).
4. Bonded Bid for Block Production - A Bond into Escrow.
5. Bonded Bid Lottery Results - List Winning Bidders and Return of Escrow to non-winning Bidders.
6. Allocation of Escrowed Bonds among *ommers* who Produced valid Blocks.
7. Payment of Annuity Fees to Storage and Network Service Providers (SNSPs)

Connections between SNSPs allow Segment Request and Response over established, pre-paid channels with only occasional new Transactions on the Slán-Chain™ for Audit and Connection Payment required.

Connections also support the exchange of:

1. Candidate Transactions
2. Bids by Bonded Producers
3. Data Storage Segments independent of Slán-Chain™ Operations
4. SNSP Background Replication and Audit.

We need some kind of SNSP Discovery Protocol.

Block Producers (or SNSPs) must pay to connect to the network:

1. To prove legitimacy with a signed Transaction.
2. To ensure good behavior and fair payment for network traffic.

How can a new player get invited to the party?

UTXOs or Accounts? Neither.

Bitcoin introduced the concept of Unspent Transaction Outputs (UTXOs). Every Bitcoin Transaction creates one or more outputs with rules concerning how they can be used. Every output may be used as an input to a future transaction exactly once, if the correct conditions are met (i.e. proper signatures).

All Bitcoin Miners maintain a list of every Transaction that has yet to be spent (UTXO) in order to be able to locate and verify the value when it is used as input to a potential new Transaction.

Bitcoin uses intelligent Wallet Applications (off-chain software with access to the entire Blockchain) to locate the possible multiple UTXOs that are spendable by that Wallet's private keys. These UTXOs may then be used as inputs to the next Transaction that the user initiates.

Conversely, Ethereum introduced the concept of Accounts which use a data structure that includes a running balance (and other data) for every Account Identifier that has ever existed. Every Ethereum miner must maintain a copy of this Balance Ledger structure.

Ethereum Wallets need only maintain the access keys for a particular Account and the running balance is immediately available from the network. This immediate access comes with the penalty that the current balance and all trading partners are published for the world to see.

Slán-Coin™ is Different:

The inherent ability to look up every instance of a particular data value via a linked list structure allow us to safely create a true Credit / Debit architecture.

Therefore we can instantly approve any Credit to a particular ID.

The rules prevent any account balance (chain of credits and debits) from becoming negative.

This allows us to approve any Debit after tracing the transaction chain backwards over recent Credits and Debits until we see a balance sufficient to cover the current proposed Debit. It is not necessary to find the Actual Balance - we only need to know that the current Debit will not drive the balance negative.

Slán-Chain™ Operating Parameters

Several parameters must be chosen to establish the operating constraints of the Slán-Chain™.

1. Block Production Rate.
2. Window for posting Candidate Transactions for a given Block.
3. Adaptable suggested Transaction Size to allow a reasonable expectation of Propagation to all Block Producers within the window.
4. Number of Transaction Inputs to allow a reasonable expectation of full Validation between the close of the window and the posting of the specified Block.
5. Value of Bond required in the Bonded Proof-of-Stake.
6. Block Delay between lottery Block and Block Produced by the first winner.
7. Number of parallel Lottery *ommers*.
8. Annuity Rate paid for Storage
9. Annuity Rate paid for Bandwidth usage
10. Block Production Bounty and Asymptote Function (if any)
11. Size of Bloom Filters used for byzantine consensus for Candidate Transactions and Segments.

We must be especially careful in the design of any self-adjusting (feedback) parameters.

Prevent conditions that drive a parameter to a limit, or unbounded value.

Prevent conditions that allow the introduction of oscillations.

Prevent conditions that allow stepwise changes or undamped excursions.

This means: **Apply Well-Designed Control System Principles**.

Storage and Network Operating Parameters

Certain parameters establish the guidelines for operation of the peer-to-peer Networking and the Distributed, Redundant Storage.

1. Size of physical Storage and Network Blocks.
2. Hash Algorithm for Segment identification.
3. Rate of Background Replication and Audit.
4. Number of Connections to Peer SNSPs.
5. Size of Bloom Filters used for Storage Announcements and Requests.

These parameters must not limit the future growth of the Network Bandwidth or Storage Capacity.

Searches, Single Instance Storage and Data Compression

Transactions include several types of Data Fields:

Inputs - Payer's ID, Amount, Signature

Scripts - Sequence of Transaction Rules for Validation and Dispersal of Funds

Outputs - Recipient's ID and Amount

Arbitrary Data - Unique Identifier of the Hash of an (unlimited length) block of Data

each of which may consist of:

Identity Handles - Unique Identifier used to associate Payers and Recipients

Signatures - Cryptographic verification that the Sender approved this Amount as payment

Values - Numeric amount of value with no predefined, arbitrary limits to scale or precision.

Indices - Method of selecting options or selected Data as input to a Script or Contract

Timestamps - Standardized method of allowing or restricting Transactions to certain times

Each of these fields may contain

Actual Values of arbitrary size (number of bits), and

May be referenced multiple times on the Blockchain

We need a design that assigns a handle to the first instance of the value when it occurs on the chain, and refers to the value using the handle on all subsequent uses.

This provides an inherent form of data compression. Arbitrarily large data values are reduced to the size of a reference handle. Ideally, the size of a handle will be dynamic and grow no faster than the size of the Blockchain itself.

Further, each instance of a handle may be stored in the form of a linked list.

This allows us to rapidly locate every instance of a particular value anywhere on the chain.

———

The handle assignment mechanism with its first instance concept implies the ability to rapidly look up any data value in order to determine its handle.

The rapid lookup is accomplished using a binary search of a specialized B-tree structure based on the hashes of the Actual Values.

This works because the order of the Actual Values is not important. We need only the ability to

1. Determine the existence of an Actual Value in the system, and
2. Add an new [Actual Value, Handle] pair to the system.

Since we are using hashes of the Actual Values we can expect the B-tree to remain generally level as it grows. This obviates the need for special code or radical operations on the tree.

It is expected that full copies of the "current" B-tree will be periodically appended to the Slán-Chain™ in the form of an operational Transaction. This will allow new participants to access the required structure without needless storage queries. Participants will build internal working B-trees based on changes after this snapshot and will independently verify the next proposed snapshot Transaction.

“Colored Coins” or Create-Your-Own Token

Independent cryptocurrencies sharing the Slán-Chain™ infrastructure. Analogy is poker chips in a casino. Total all the red ones together, and all the white ones together. But to turn red ones into white ones you have to use an exchange. Two questions arise: 1) How long does it take to “make change”, and 2) How much are the fees?

Advantages of the Slán-Chain™ include the sharing of the consensus servers, reliable networking, unlimited storage, rapid Transactions and trusted escrow.

The Slán-Chain™ technique of Transaction Approval and Balance Lookup can be used for Alt-Currencies.

The particular Alt-Currency is identified in the Transaction and verified within the Script.

We define an inherent, on-chain Exchange mechanism with other Slán-Chain™ Alt-Currencies.

The Slán-Chain™ itself acts as the trusted escrow agent for the settlement, ensuring that both parties actually possess the tokens being exchanged. The on-chain escrow is a variation of the method used to ensure that the Bonded Proof-of-Stake consensus confirms all Slán-Chain™ Transactions.

“Spot prices” are established, enforced and guaranteed to be honest by the Slán-Chain™ consensus.

Since we must be able to pay Transaction and Connection fees in Slán-Coin™, we must also act as an Exchange for Colored-Coin value into Slán-Coin™.

Thus, all Slán-Chain™ Alt-Currencies are inherently fungible and available for exchange.

Exchange rates are established by actual Transactions that occur on the Slán-Chain™. No arbitrary valuation exists.

It may be desirable to establish a Slán-Chain™ fee for the exchange service since exchange Transactions are slightly more complex to compute than simple Transactions. This should be a fixed fee per transaction, payable in Slán-Coin™, and apportioned to Block Producers.

Colored Coins should be able to implement:

1. Independent definitions of “Mining” to control inflation, etc.
2. Allow Dividends for holders-of-record.
3. We may include the ability to do splits and reverse-splits for more convenient valuation.

The problem with Ethereum’s (for example) implementation of Colored Coins (Tokens) is that the individual Coin ecosystems are totally isolated from each other and use the Blockchain ONLY to implement their idea of Smart Contracts to perform Transactions that move value between Accounts using the particular Colored Coin. This means that moving value between different types of Coin (different cryptocurrencies) requires the involvement of an Exchange - even though all the Coins are already on the Ethereum blockchain. These Exchanges will implement their own settlement rules, will establish unknown, unknowable and arbitrary actual exchange rates, will make no use of the Blockchain to maintain integrity, will collect a commission and will be subject to fraud. As if all that wasn’t bad enough, the “totally isolated Colored Coin” concept gets violated anyway, because you must have some ETH (from somewhere) in order to pay the Gas for the Transactions involving the Colored Coins on the chain.

The Meaning of Currency

Many of the terms that we routinely use will benefit from a bit of historical perspective. Please take a moment and enjoy this excerpt.

> “One of the English was using a funny word yesterevening — ‘currency.’ Do you know it?”
>
> “It is the quality that a current has. They speak of the currency of the River Thames, which is sluggish in most places, but violent when it passes under London Bridge. It is just the same as our word Umlauf — running around.”
>
> “That is what I supposed. This Englishman kept discoursing of currency in a way that was most fraught with meaning, and I thought he was speaking of some river or drainage-ditch. Finally I collected that he was using it as a synonym for money.”
>
> “Money?”
>
> “I’ve never felt so dense! Fortunately, Baron von Hacklheber is visiting from Leipzig. He was familiar with the term — or quicker to decypher it. Later I spoke with him in private and he explained all.”
>
> “What an odd coinage.”
>
> “You are too witty for your own good, girl.”
>
> “The Englishmen cannot get away from this topic. Their relationship to money is most peculiar.”
>
> “It is because they have nothing but sheep,” Sophie explained. “You must understand this if you are to be their Queen. They had to fight Spain, which has all of the gold and silver in the world. Then they had to fight France, which has every other source of material wealth that can be imagined. How does a poor country defeat rich ones?”
>
> “I think I am supposed to say ‘the grace of God’ or some such —”
>
> “If you please. But in what form is the grace of God manifested? Did piles of gold materialize on the banks of the Thames, as in a miracle?”
>
> “Of course not.”
>
> “Does Sir Isaac turn Cornish tin into gold in an alchemical laboratory in the Tower of London?”
>
> “Opinions differ. Leibniz thinks not.”
>
> “I agree with Baron von Leibniz. And yet all the gold is in England! It is dug up from Portuguese and Spanish mines, but it flows, by some occult power of attraction, to the Tower of London.”
>
> “Flows,” Caroline repeated, “flows like a current.”

Sophie nodded. “And the English have grown so used to this that they use ‘currency’ as a synonym for ‘money’ as if no distinction need be observed between them.”

Caroline said, “Is this the answer to your question — how does a poor country defeat rich ones?”

“Indeed. The answer is, not by acquiring wealth, in the sense that France has it —”

“Meaning vineyards, farms, peasants, cows —”

“But rather to play a sort of trick, and redefine wealth to mean something novel.”

“Currency!”

“Indeed. Baron von Hacklheber says that the idea is not wholly new, having been well understood by the Genoese, the Florentines, the Augsburgers, the Lyonnaise for many generations. The Dutch built a modest empire on it. But the English — having no other choices — perfected it.”

“You have given me new food for thought.”

Neal Stephenson. The System of the World (The Baroque Cycle Book 3). HarperCollins.

Slán-Chain™ Bonded Auctions

Basic Transactions on any Blockchain involve a transfer of coin value from one party to another.

More complex Transactions may involve multiple senders and/or multiple recipients.

The rules for apportioning the funds are contained in the Script that accompanies each Transaction.

Cryptocurrencies also require a mechanism to **auction a transaction** to the highest or lowest bidder.

Specifically, we wish to be able to handle these cases:

1. Seller has an item that he wishes to sell to the highest (unknown) bidder
2. Buyer wishes to purchase an item from an (unknown) seller for the lowest price

Unlike simple Transactions, the transfer of value is to or from a party that is unknown at the time of the initial offer. There must be a trusted agent to mediate the connection between these parties. The Slán-Chain™ implements the necessary protocol to manage the escrow of value and ensure the integrity of the ultimate transaction.

Essentially there are three steps:

1. **Post an Offer** to buy or sell. Describe the item, set a price range and a Closing time. Offers include the Public Key of the originator (since the Transaction must be signed by the sender).

2. **Post Bids**. Each will refer to the initial offer and set a bid price and additional information. Bids are added to the Blockchain to provide a permanent, immutable record of the honesty of the auction. All bid prices are withdrawn from the bidder's account and held in escrow until the close of the Auction. Multiple bids from the same bidder are handled by the Slán-Chain™ - recent bids "replace" the older ones and the escrow is computed appropriately. The bonds are held in escrow for all bidders until the close of the auction.

Bids include the Public Key of the originator (since the Transaction must be signed by the sender). The "additional information" can take the form of public or private text messages. This allows a public chat visible to all Slán-Chain™ Observers, or private messages (encrypted with the recipient's Public Key). Use cases include sending the mailing address for a package, for example.

3. **Close the Auction**. Slán-Chain™ settles the Transaction and returns the escrow of losers. The auction closes immediately when a bid hits the upper limit specified in the Offer. The first Bid that hits the limit is the winner; all others are losers and their escrow is returned. If the Bid never reaches the upper ("Buy it Now") limit, the auction closes with the first Transaction after the published Closing time.

Bids can be changed or withdrawn at any time prior to the close of the auction by simply creating a new transaction that will override the previous one from the specific bidder. The Slán-Chain™ handles the maintenance of the correct escrow balance based on the current bid.

The auction itself can be withdrawn by the originator, prior to close, by posting an overriding Transaction. The Slán-Chain™ returns the escrow balances to all bidders.

Using Slán-Chain™ Escrow for Bid-Ask Settlements

Two anonymous parties wish to reach an understanding for an exchange of token value.

One makes an offer to sell in the form of a Quantity of Tokens (value) and a range of Asking prices.

The other makes an offer to buy in the form of a Quantity and a range of Bidding prices.

The Bid and Ask offers may be made in either order, are individual Transactions on the Slán-Chain™, and remain viable until either withdrawn or settled.

Every Transaction involving either Bid or Ask transfers the sender's Token value into Escrow on the Slán-Chain™, thus ensuring that the sender actually has the necessary value and preventing the sender from double-spending, bidding on overlapping offers, or fraudulently promising future payments.

Off-Chain Exchanges: Dangerous But Necessary

In order to move Value on to and off of the Slán-Chain™ it is necessary to involve a trusted third party to act as an Exchange.

For example, in order to "Buy one USD worth of Slán-Coin™", I must:

1. Find a trustworthy entity who currently owns some Slán-Coin™ and would like some USD.
2. Negotiate an Exchange Rate with her to establish how many Slán-Coin™s I will receive.
3. Send her my $1 USD by some (existing) banking or USD funds transfer mechanism.
4. Ensure that she has received the funds.
5. Wait while she creates a Slán-Chain™ transaction that conveys the agreed Slán-Coin™ to me.

The problems associated with this are many:

1. Trust vs. Transparency

   Trust that the quoted Exchange Rate and Fees are fair and equitable.

   Trust that the Exchange actually has the promised Slán-Coin™ available.

   Trust that the Exchange will acknowledge receipt of my funds.

   Trust that the Exchange will honor the negotiated Exchange Rate.

   Trust that the Slán-Coin™ will be paid promptly.
2. No single-source Audit of agreements or performance
3. Incentivizes misbehavior of the Exchange since there is no penalty for simply lying.
4. Exchange holdings on the Blockchain are vulnerable to the compromise of their Private Key.
5. Exchange Transactions are monitored and taxed.

There is a Big Difference Between

"**We *Won't* Cheat You**"

and

"**We *Can't* Cheat You**"

Transaction Frequency on the Slán-Chain™

The Slán-Chain™ requires an adaptable, self-regulating method to define the windows for accepting Candidate Transactions and posting Produced Blocks to the chain.

It is expected that the processing and network communication speed will increase with time; conversely the number of Transactions targeting the Blockchain will also increase.

Keeping these opposing forces in balance and ensuring that new Block Producers can join the network with reasonable expectations of success will require a built-in feedback system.

As with all aspects of the Slán-Chain™, this feedback mechanism must rely only on on-chain data and must be able to be unambiguously evaluated by all participants.

The use of Timestamps included with the Signature of every Block Produced enables all participants to determine the amount of time that it took for the Candidate Transactions to propagate to the Producers and the amount of time it took the Producers to verify and format the particular Block.

Since we always have multiple Block Producers reaching a consensus concerning each block, we also have a representative sampling of these combined Production Delays.

This representative sampling allows for the computation of the Mean Delay and Standard Deviation for each block, which can be correlated with the size of each block.

Our goal is to automatically adjust the Candidate Transaction Acceptance Windows and the target Block Production rate so that we achieve the fastest performance based on an average of the production rate of the last several Blocks.

The Window-size adjustments are intended to allow for the (hopefully rare) case where a network or server failure causes Block Production to be delayed or not Produced at all.

The important thing is to achieve a generally consistent Block Production rate given varying network speed, computation speed and Candidate Transaction rate.

In addition, the ability to unambiguously assess Slán-Chain™ and Network performance allows new Block Producers to "play along" with the network - operating in a shadow mode - for a period of time. This allows them to develop confidence in their computing and networking reliability prior to putting up a Bond to become a Block Producer.

Bitcoin is only able to adjust Work Requirements to achieve an (arbitrary) one block per 600 seconds.

The Slán-Chain™ is able to optimize Production Rate to allow for ever-increasing Network Speed and Transaction Volume.

Slán-Chain™ Escrow Mechanism

The Slán-Chain™ Escrow Mechanism is a specialized account that temporarily holds value in the form of cryptocurrency across multiple Transactions.

The most important thing about Escrow is that there is **No Private Key** associated with it.

All other "Accounts" have a public address that receives value and a Private Key for spending value.

The rules of the Slán-Chain™ and the consensus of Block Producers guarantee the integrity of the Escrow.

The Escrow Mechanism allows users to safely and securely put up a Bond with the full expectation that an unknown third party will perform as expected or the Bind will be refunded.

The Trusted Entity is the Slán-Chain™ itself — not a third party.

The Escrow Mechanism is the core of the operation of the Slán-Chain™ and is required to be incorruptible in order for the Bonded Proof-of-Stake to ensure consensus among Block Producers.

We leverage the incorruptibility of the Escrow system to enable:

1. Trusted Auctions.
2. Colored Coin Exchanges.

The lack of any cryptographic key associated with the Escrow function means that the Slán-Chain™ can be completely decentralized and has no single-point weakness.

By NOT requiring private keys to sign Escrow payments we are able to

1. ensure that all Block Producers and Observers are able to verify the correctness of Transactions involving Escrow payments.
2. eliminate the vulnerabilities that would exist with Private Key Management, including delays, distribution, protection, revocation.
3. ensure that all escrowed funds are either legitimately spent or refunded at a particular time, preventing funds from being "lost in limbo".
4. ensure transparency and audit-ability of all Transactions.

Slán-Chain™ Value-Added Services

There are many opportunities for off-chain services that may generate revenue independent of the Slán-Chain™ itself. Thus, the Slán-Chain™ is a basic enabling technology for previously difficult or impossible services.

The Slán-Chain™ provides a secure, trusted and transparent foundation for these new services.

1. Transaction Gateway and Wallet Support
2. Exchanges
3. Smart Contract Judiciary
4. Auction Houses
5. Data Archives and Searches
6. AI and Deep Learning
7. Universal ID and Protected Personal Information
8. Bonded Proof-of-Stake Pools and on-chain interest-bearing or dividend-yielding investments

Creation of Colored Coins

Colored Coins (Alt-Tokens) on the Slán-Chain™ are created with specialized versions of Bonded Exchange Transactions.

1. The creator builds a Transaction that describes the new Coin by giving it a text name and description, and an initial quantity. The Slán-Chain™ will assign a unique handle (cryptographic hash) by which the Coin will be referenced in all future Transactions. The initial quantity of newly minted Coin will be assigned to the ID of the creator.

2. The creator may use any normal (Unilateral) Transactions to pay others with this new Coin.

3. Anyone possessing this new Coin may also use any normal (Unilateral) transaction to pay others with this new Coin.

4. Anyone possessing this new Coin may use Bid-Ask (Exchange) Transactions to convert this new Coin to other Colored Coins, or to Slán-Coin™.

5. The creator may build a new creation Transaction referencing the handle of the Coin. The Slán-Chain™ will verify the signature of the creator and ensure that the ID matches the original creator of the particular Coin. Freshly minted Coin in the specified quantity will be added to the creator's ID. This is the equivalent of a mining operation in other cryptocurrencies and requires the verified signature of the original creator.

6. The creator may "LOCK" the new Coin by signing a Transaction that mints exactly zero new Coin. This LOCK will prevent any additional Coin from ever being minted. Specifically, the signature verification phase described in (5.) above will cease to honor any signatures. This means that the new Coin can never be devalued by its creator and that holders are protected even in the case that the creator's Private Key is compromised.

The value of Colored Coins is established only by market (Exchange) forces; they have no inherent or fixed value. This means that Colored Coins may not be used to pay Transaction fees on the Slán-Chain™.

All Transactions on the Slán-Chain™ require fee payment in Slán-Coin™, in addition to whatever Colored Coin value is transferred.

This mechanism ensures that Colored Coins implemented on the Slán-Chain™ incur as liitle cost as possible and are secured by the full integrity of the Slán-Chain™ itself.

Because Coin creation, transfer and exchange is easy and inexpensive it will be convenient for applications such as managing company stock.

Care should be exercised to ensure that Transactions are directed to the correct handle for the desired Coin. Handle lookups using name and description suffer from the same problem that "look-alike domains" have when using the Internet Domain Name Servers.

The Slán-Chain™ will verify that the Coins referred to in an Exchange Transaction exist and that both parties have the claimed quantity, but the system cannot verify that these Coins are the ones you *intended* to acquire. *Caveat Emptor.*

Slán-Chain™ Transaction Types

Every Transaction supported by the Slán-Chain™ must be able to be understood and verified by the servers that add it to a particular Block. The Transactions must then be able to be searched in an expeditious manner to allow future Transactions to make use of the results. Furthermore, Observers of the blockchain must be able to recognize and understand the meaning of these Transactions.

We design a very limited number of Transaction Types that support the intended uses of the Slán-Chain™ and provide a trusted foundation for new or unforeseen applications.

1. **The Genesis Transaction**. This Transaction allocates previously non-existent Slán-Coin™ to a large number of different IDs. There is only one Genesis Transaction. It is necessary to establish a large number of IDs with sufficient Slán-Coin™ to be able to post the Bonds required to implement the Bonded Proof-of-Stake consensus mechanism that is the foundation of the Slán-Chain™.

2. **Block Production Bid Transaction**. A Production Bid is an offer for the originator to act as a Bonded Block Producer for a future Block on the Slán-Chain™. This implements Bonded Proof-of-Stake.

3. **Payment Transactions**. A payment Transaction verifies the Sender (who signs the Transaction) has a sufficient balance and transfers the specified amount to one or more Recipients. Recipient IDs are not verified and do not need to exist previously. In order to spend the funds received, the Recipient must have the ability to cryptographically sign a Transaction with the matching ID.

Therefore, care should be exercised to ensure the validity of the IDs. Such validation is not the province of the Slán-Chain™ and incorrect usage may result in the permanent loss of value.

4. **Auction Transactions**. A payment to or from a Transaction Originator that is matched with a previously unknown other party. Auction Transactions establish the description and intent of the Auction, or withdraw the Auction from the Slán-Chain™.

5. **Bid Transactions**. A Bid Transaction establishes the legitimacy of the Bidder and the value of her Bid. The Bid value is held in escrow by the Slán-Chain™ until the bid is revised, withdrawn, or the auction ends.

6. **Coin Creation Transactions**. Coin Creation consists of establishing a Name, Description, Quantity, Handle and Owner of a new Alt-Coin. The Owner of the new cryptocurrency (Coin) may create more (minting) using additional Creation Transactions, or may permanently LOCK the Coin to prevent any possibility of more of that Coin ever being created on the Slán-Chain™.

7. **Exchange Transactions**. An Exchange Transaction offers to exchange a quantity of one Coin for a quantity of another Coin. The Slán-Chain™ matches previous or future offers and ensures the rapid, fair and secure settlement of the Transaction. Exchange Transactions that have not settled may be revised or withdrawn by their Originators. The Slán-Chain™ ensures the existence of sufficient funds from both parties and holds the funds in escrow to prevent any possibility of double spending.

8. **Data Transactions**. All Transactions may have an (effectively unlimited) amount of auxiliary data stored with them. Typically, this may simply be comparable to the MEMO field on a check, or document images for future reference. This auxiliary data may be encrypted, or it may be expressly designed to be searchable. In any case the fees paid to the Slán-Chain™ by the originator of the Transaction will be based on the size of the Transaction. Large amounts of auxiliary data will therefore incur larger fees. Some Transactions will be intended only to add the immutable record of the auxiliary data to the Slán-Chain™. These are Data Transactions, and are unique in that they specify no Recipients or Script for distributing Coin value.

Slán-Chain™ Escrow Payments

The Slán-Chain™ Escrow mechanism is used to support several features that require a Trusted Agent.

1. Bonded Proof-of-Stake that enables the consensus mechanism for Block Production.
2. Payment for the Block Producers that verify and organize Transactions on the blockchain.
3. Payment for perpetual Storage and reliable Network connections via an Annuity.
4. Auction and Exchange services where the Buyer and Seller are Verified but Anonymous.

Each of these escrow-related Transactions begins with an explicit, signed payment of a particular Coin amount into the Slán-Chain™ escrow account. The value is held until one or more complementary Transactions require payment from Escrow.

The important thing here is that payments FROM escrow are authenticated by the Slán-Chain™ rules themselves, verified by Block Producer consensus and added as the last item to the current Block as an unsigned Escrow Payment Transaction. I.e., the Signature that validates the Escrow Payment is not a signature on the Transaction itself but rather the verified Signature of the Block Producer on the entire Block.

The rules for constructing a Block - including the verification of Candidate Transactions, the ordering of Transactions in the Block and the creation of the Escrow Payment Transaction - are all deterministic and designed to ensure that all Block Producers and Observers will independently generate the same results and therefore the same Block Hash.

This consensus among Producers on the formation of the next Block is the feature that ensures the integrity of the Slán-Chain™.

Concerning Stalled Block Production

Under normal operating conditions a large number of stakeholders bid for the right become a Block Producer. From each auction, several of these bidders will be deemed winners and assigned to produce a particular future Block. These Bonded Producer auctions occur continuously and ensure that there are a number of Bonded Block Producers assigned to every Block.

As long as a quorum of Bonded Block Producers correctly generate identical Blocks, the system runs as intended. The multiple Producers are referred to as *ommers*. One *ommer* is deterministically selected as the one which will be the "official" Next Block on the Slán-Chain™ and is awarded the highest Block payout. The other *ommers* are rewarded with lesser payments.

Block Production can stall if a quorum of Block Producers o not agree on a particular Block.

This situation can occur if there is a particularly sharp increase in Transaction volume or Transaction size that exceeds the ability of the network to reach the necessary Producers within the allocated window.

Widespread network outages or partitioning of the network into subsets with insufficient Producers online can also cause stalls.

Correcting Block Production stalls is handled by a retry mechanism.

Specifically, the Block Production window is reopened. This allows all Block Producers to accept Candidate Transactions that might not have arrived in time and caused Production discrepancies. The width of the operating windows is increased to help prevent repeated stalls due to traffic peaks.

By establishing dynamically adjustable windows based on network capacity, and an adjustable number of Bonded Producer Auction winners, and an adjustable required quorum of *ommers*, it is expected that the Slán-Chain™ will be immune to chain forks or takeover attacks involving less that half of the available Proof-of-Stake Bondholders.

Blockchain Construction

Candidate Transactions are presented from outside users and initially Verified by a Storage and Network Service Provider (SNSP). Candidate Transactions always include the number of the Target Block intended to contain the Transaction. The initial Verification ensures the existence and current accessibility of all the inputs (including currency amounts) required by the Transaction. The SNSP also ensures that all items required for other SNSPs to successfully Verify the Candidate Transaction are available during the processing window for the Target Block number.

Candidate Transactions are identified by their Hashes and sent to all networked SNSPs.

SNSPs sort Candidate Transactions into Target Blocks ordered by their Hashes. This is the first step toward building a Candidate Block.

After the close of the window for the new Block, the SNSPs reevaluate the Transactions in the order that they will appear in the Block. The validation at this point will be exhaustive and Transactions may fail - for example, due to double-spending by multiple Transactions within this Block. Candidate Transactions that fail to verify at this point are simply deleted from the Block. Failed Transactions will have to be resubmitted by their originating user through the full process.

As the Candidate Transactions are Verified, the SNSP creates an ordered list of Escrow Payment Transactions required by the Transactions within the Block. These Escrow-related Transactions are appended to the new Block. The Hash of the completed Block is used to identify it as a Block Candidate.

Block Producers use the Block Candidate Hash and apply their ID and Signature to create a Candidate Block. It is these Signed Block Hashes that are shared among SNSPs and are used to establish consensus among Block Producers concerning the legitimacy of the Block Candidates.

Data-Store Construction

The Slán-Chain™ Data-Store ensures that the data elements used repeatedly on the Blockchain need only reside at a single location, and be stored only once, within the Slán-Chain™ address space.

There are two fundamental requirements for dealing with Blockchain data:

1. Determining if a particular data item is already part of the Blockchain, and
2. Finding all references to the individual item.

These are especially important as we envision the need to find infrequently used items within a potentially unlimited amount of Blockchain data.

The "single location" concept has the added advantage that a short Handle may be used to refer to otherwise large blocks of data. Examples include IDs and Hashes, but not Signatures.

The second requirement is easily handled: By referring to data elements by their Handles we also create the ability to form linked lists of references to those Handles. These linked lists allow very rapid, incorruptible searches for all uses of a particular ID.

The first requirement is more difficult, as it involves both looking up a particular data item to determine its existence within the Blockchain but also determining its handle. This is accomplished using a specialized form of binary tree, or B-Tree. Using a B-tree structure ensures that any desired item can be located with a minimal number of accesses to the Data-Store and that that number will scale manageably as the size of the Data-Store grows into the future. Of particular interest is the fact that the percentage of the B-Tree required to be accessible within the memory of any individual SNSP will shrink with time and not place undue burden on these Providers.

Another interesting observation concerning the use of a B-Tree to lookup Hashes is that the inherently random nature of Hashes ensures that the B-Tree will remain generally level - i.e., the worst-case depth of any branch will be essentially the same throughout the tree. This means that no special software considerations will be needed to perform leveling operations and that the access speed and performance of the B-Tree will be statistically predictable into the future.

Further, the vast majority of nodes within the B-Tree will remain unchanged once they are created. Changes to the tree structure required by the continuous addition of new data elements will affect an ever-decreasing percentage of the entire tree. In keeping with the philosophy of all Blockchain technologies only the updated portions of the B-Tree need be added to the Data-Store.

It is necessary to have a table that can be used to find the most recent reference to a particular Handle within the Blockchain. This allows the linked list of references to be traced in its entirety. The size of this table will correspond to the total number of Handles on the Blockchain, thus it grows linearly with the size of the Blockchain.

As with all of these structures, the "working set" or portion of the Data-Store that must be held by any SNSP at any given time shrinks with time. Furthermore, the shared production and consensus mechanisms required by the Slán-Chain™ ensure that the active participants will share a similar working set and can expect to be able to ask each other for elements that they might not have within their own local store.

In order to facilitate new SNSP participants and Observers rapid congruence with the Slán-Chain™ it will be advisable to preiodically post complete copies of the B-Tree and Handle Table to the Data-Store. Implemented correctly, this will result in only the recent changes being posted to the physical Store.

Statistical Sharding

Previous sections have described the creation of Blocks on the Slán-Chain™, the single-instance storage of data elements and searching for occurrences of those elements. This section describes the physical storage of the resulting Data-Store and the Segmentation that allows distributed, redundant storage and peer-to-per networking access to the Slán-Chain™.

The most important feature of any Blockchain and the Slán-Chain™ Data-Store in particular is that it is "Append Only". That is, any data written to the Data-Store is indelible and will never change. New data can be added, possibly updating items with a new "current value", but all previous values will remain in the Data-Store and available for inspection as required.

This "write-once" property means that the Data-Store can be written to a series of Segments and that, once written, each Segment will remain unchanged forever. The Slán-Chain™ defines a fixed Segment size and uses this as the basic unit of permanent Storage and of Network communication. In addition, each Segment is cryptographically Hashed to assign a unique Segment Name. The use of hashing ensures that any Segment, requested by Segment Name, cannot be forged or otherwise corrupted.

Several specialized Segments are added to the physical implementation of the Slán-Chain™ Store.

1. Index Segments - contain a list of Logical Base Addresses and Segment Names to allow locating the Segment containing any desired portion of the Slán-Chain™

2. Forward Error Correction (FEC) Segments - contain redundant bits to allow recreation of missing or corrupted Segments.

In some distributed storage systems breaking data into chunks such as this is referred to as Sharding; the Slán-Chain™ uses the term Segmentation.

FEC Segments are inserted periodically as the Data-Store grows. FEC Segments provide overlapping protection to previously added Segments such that any Segment can be recreated even in the event that multiple nearby Segments are unavailable. This Error Correction capability ensures that it is never necessary to be able to retrieve a specific Segment - any Segment can be rebuilt given time to access adjacent Segments.

Index Segments are hierarchical in nature. This allows a tree of interlinked Index Segments to provide the resources to find the Name of the Segment containing any Logical Address on the Slán-Chain™.

The most recently created Index Segments contain the Logical Addresses and Names of the most recent Segments added to the Data-Store. This has the effect of allowing any Observer or Participant to be able to gain access to the Slán-Chain™ from scratch by simply requesting the most recent Index Segment. This Root Request is a fundamental capability of all Storage and Network Service Providers (SNSPs) and is provided in their connection announcement and authentication protocol. Receiving consistent announcements from multiple SNSPs ensures that customers (users) only connect to authentic Providers.

As time progresses the number of Segments stored on the hardware of any particular SNSP will exceed to capacity of that server. The choice of which Segments to discard will be made on a Statistical basis. The SNSP will use one or more randomly selected Bloom Filters to choose which Segments to retain. This technique ensures that different peer-to-peer servers and connections between them will be likely to possess different subsets of the entire Data-Store. In addition, each SNSP can publish to the peers the Bloom Filters it is using. Thus, when a request for a particular Segment Name is received, each SNSP can make an "educated guess" as to which Network Connection is most likely to yield the Segment.

Slán-Coin™ Exchange Rules

The Slán-Chain™ provides inherent, on-chain support for cryptocurrency Exchange among all Tokens implemented on the Slán-Chain™. This is accomplished by making use of the Slán-Chain™ Escrow mechanism. Escrow Transactions are generally **more secure** than any other Transactions on the Blockchain since they are based on the consensus mechanism of the Slán-Chain™ itself and are not subject to the possibility of forged signatures or the loss of Private Cryptographic Keys.

Slán-Chain™ Exchanges allow a Bid-Ask type auction of one cryptocurrency for another. This may include Slán-Coin™ or any other on-chain token.

The holder of Coin A may offer to sell quantity $\mathbf{Q_A}$ coins for quantity $\mathbf{Q_B}$ of Coin B using a Transaction of the form **A**→**B**. The Slán-Chain™ Block Producer will ensure that the quantity $\mathbf{Q_A}$ is actually owned by the holder.

During the Escrow Management phase of the Production of the current Block, the Coin A quantity $\mathbf{Q_A}$ will be moved to the Slán-Chain™ escrow and will no longer be the property of the original holder.

The Block Producer will then examine the Slán-Chain™ looking for all recent, pending Transactions of the form **B→A**. The most recent offer with compatible quantities (if found) will be used to create a settlement Transaction. Settlement Transactions pay the quantities of Coin A and B to their new holders from the Slán-Chain™ escrow. The details of handling all the possibilities will be documented elsewhere. In particular, the meaning of "recent, pending" Transactions is intended to ensure that all operations are deterministic, that extremely long-term pending offers are handled correctly and that the behavior of the Slán-Chain™ is correct even during periods of high transaction rates.

In addition, provision is made to allow settlement of offers which include ranges of quantities (instead of the fixed amounts shown in the previous example), as well as discount offers for larger quantities.