Bitland: A Peer-to-Peer Digital Property Deed System

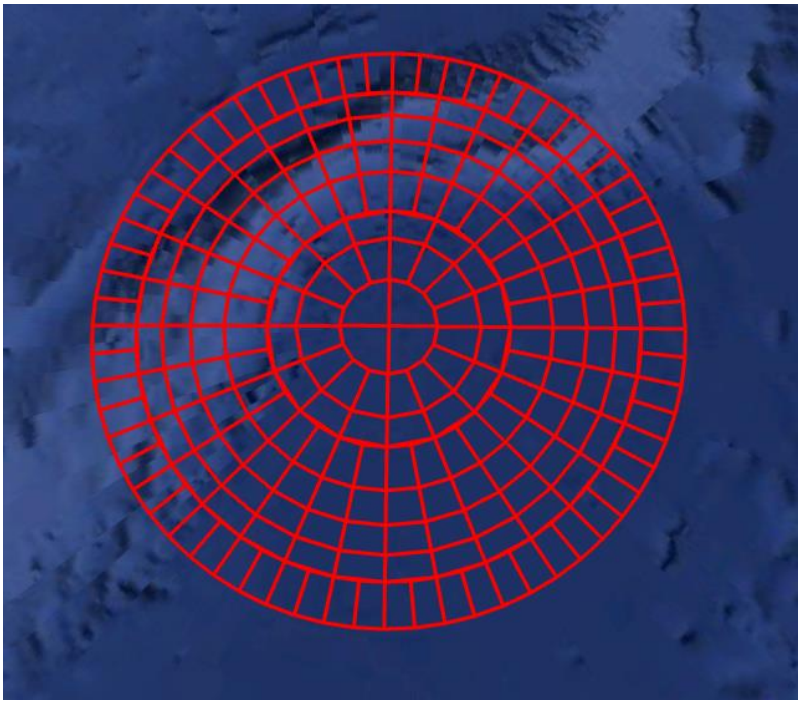
By: Brett Wood

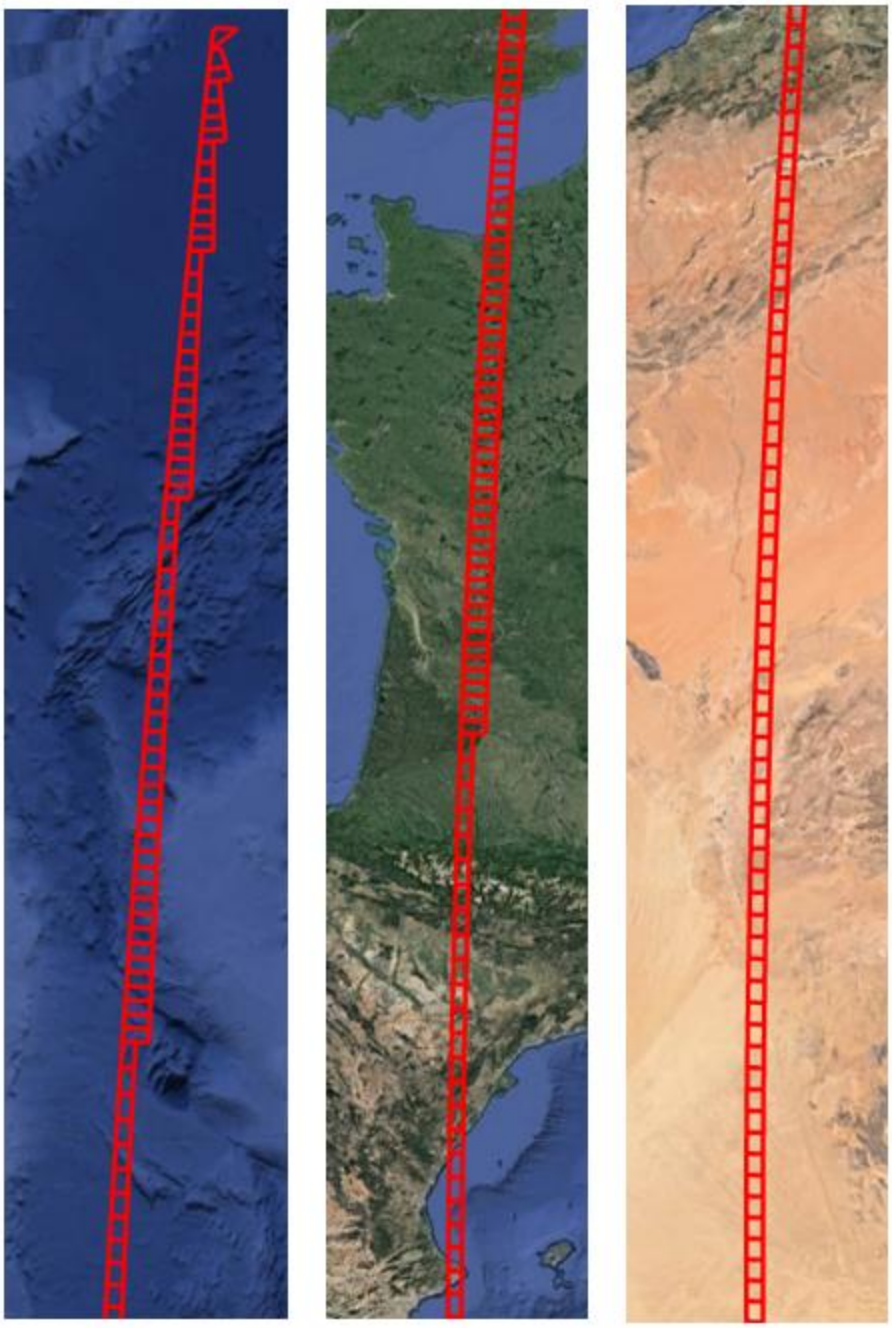
Abstract: In the same way that Bitcoin has proven useful for some to store savings and transact in a peer-to-peer network with no centralized authority, others may find a use for a decentralized, cryptographically secure proof of digital "land" ownership. It could be for digital rights of metaverse land parcels or maybe it could be adopted to replace analog deeds in physical world. One would want to familiarize themselves with the Bitcoin whitepaper before reading this as it borrows and builds on many of the concepts found there.

Introduction: There are countless examples of digital worlds being created: Second Life, Minecraft, World of Warcraft, Facebook's new "Metaverse", and many others. In each of these cases, the user invests significant time building in the ecosystem but ultimately doesn't own anything independently of the publisher of the world.

The other theme that you will see as part of the whitepaper is the use of Bitcoin as the source of truth for the system. The timing of the system is tied more to Bitcoin block time than UTC time (for example: difficulty adjustment is based on timing of Bitland blocks relative to Bitcoin blocks). And the most trust-minimized transactions possible in the system are based on truth being conveyed through the Bitcoin blockchain (there are some transaction types which can be set up to settle based on actions taking place in Bitcoin).

Land Polygons: It is a tricky problem to divide the surface of a sphere into polygons. The way it was approached in Bitland is to pick a target area in square meters on Earth – in this case 400,000,000m^2 was picked. First create 4 polygons at poles, extending far enough towards the equator such that the areas equal the desired area. Let's focus on north pole for the rest of this walkthrough. For the next polygons, work south, extending the edge of the first layer of polygons until they reach the desired area. If the north-south edge is more than double the east-west edge, then split the starting edge in half and instead create 2 polygons extending down from the original one. In the case of the second row, it actually double-splits, so four polygons come out of the original pole polygons (see first image, mapped onto Earth). Looking at the first image, you can see the next split at layer four, then another at layer eight. The second image shows one column of this continuing down through the equator.





This process results in 1,275,336 landbase polygons on the planet. Initially Bitland will only support one planet (mapped to Earth), but there is no reason the protocol couldn't be extended to other planets if useful. Perhaps in the future a planet that is soon to be settled could be mined initially on Bitland! When settlers reach the planet, they could fork Bitland chain and have their new chain start at that point.

These initial polygons are quite large. In subsequent transactions, users of the system can subdivide landbase transactions into smaller polygons. See the Transactions section for more details.

Mining: As Bitcoin had the problem that one cannot fairly map the existing system of money and ownership into the new system, Bitland faces the same problem. Rules for current ownership of land

vary widely by jurisdiction and don't exist in some places. So Bitland proposes borrowing the mining concept from Bitcoin to distribute the digital rights to land.

Bitland blocks will follow a similar cadence to Bitcoin, targeting one Bitland block being mined for every Bitcoin block. The miner can claim a landbase transaction by picking from one of the available landbase polygons as described in the Land Polygons section. At inception, the eight blocks at the poles are the available landbase polygons to choose from. As landbase polygons are mined, they open up any adjacent polygons as valid to mine in subsequent blocks.

Difficulty adjustment works similarly to Bitcoin except that rather than targeting a clock time, a Bitcoin time is targeted. The system targets that the last 2,016 Bitland blocks should be lined up with 2,016 Bitcoin blocks.

Blocks: Blocks are conceptually identical to Bitcoin. A block is made up of a set of transactions. Typically, a block will be one landbase transaction and n other transactions. Initially, like Bitcoin, blocks will be limited to 4MB of size. The block header consists of the following information:

1. Block version
2. Previous block
3. Merkle root of transactions in the block
4. UTC time
5. Difficulty bits
6. Best Bitcoin block hash
7. Bitcoin block height
8. Merkle root of last 64 Bitcoin blocks to ensure same chain
9. Miner's bitcoin address (see the miner fee explanation for why this is needed)
10. Nonce

Transaction: As with other parts of the Bitland system, conceptually transactions are very similar to Bitcoin. But I'd argue this is the part of the system which specifically has more complexity (not necessarily a good thing) than Bitcoin due to:

- The non-fungible nature of land shapes vs. coins
- The logic around miner and transfer fees given no native token in the system
- The ability to put claims on land outputs to avoid permanent land loss

Before diving into the structure of a transaction, it will be useful to cover the concepts of miner fees, transfer fees, and claims. These are the key concepts that allow for trust-minimized transactions between exchanging parties as well as miners.

Miner Fees are used as incentive for a miner to include a transaction in the block it is mining. Miner fees are specified in a transaction as the number of satoshis (1/100,000,000 of a bitcoin) to be paid to the miner's specified Bitcoin address within a specified number of Bitcoin blocks from when the transaction is included in a Bitland block. To ensure payment of the miner fee, a transaction can include a "collateral" output type which becomes spendable by the miner if the miner fee is not paid. When a miner fee is paid before the "due date", and has ten confirmations on the Bitcoin chain, the collateral output is spendable by the output address.

Example: Bob wants to transfer square A of land to Sue. He broadcasts it in a transaction with a miner fee of 20,000 satoshis payable within 1000 blocks. He posts a collateral UTXO of the top left 1% of his land square. The transaction is included in a block, but Bob then proceeds to forget about the satoshis he owes the miner. When 1000 Bitcoin blocks expire (plus a 10 block buffer) the collateral land now is spendable by the miner and no longer by Bob.

Transfer Fees are similar to miner fees but are targeted at minimizing trust between the parties exchanging land. Transfer fees are specified in a transaction as the number of satoshis to be paid to the transaction sender's specified address within a specified number of Bitcoin blocks. To ensure payment of the transfer fee, all the outputs become spendable by the "failover address" if the transfer fee is not made. The failover address is the first input address.

Example: Bob wants to transfer square A and B of land to Sue. They agreed that Sue would pay 50,000,000 satoshis for the transfer. Bob creates and broadcasts a transaction with inputs A and B being transferred to Sue's address or addresses and he includes a transfer fee of 50,000,000 satoshis payable to his bitcoin address included in the transaction within 2000 blocks. Sue decides to back out of the deal and does not complete the payment on the Bitcoin network. After 2000 Bitcoin blocks (plus 10 block buffer) squares A and B are spendable again, both by address A since it is the failover. Conversely, if Sue had transferred the funds after 950 Bitcoin blocks, she now takes control of the land starting at Bitcoin block 960.

Claims prevent against land being "lost" forever. Unlike Bitcoin, where lost coins are not an issue for the health of the network, lost land would lead to a deterioration of the usefulness of the system because gaps would emerge. Claims guard against this, by giving speculators the ability to claim an output that they think may be lost. A claim is made by specifying the desired output as a claim in the input of a transaction. The claim "stake" is the amount of miner fee satoshis. If a claim has already been made on an output, the new claim must stake its claim for at least 50% higher fee. To invalidate a claim, the output holder has ~1 human year – 52,500 Bitland blocks from the time of miner fee being validated – to move the output in a transaction. If the output is not moved in 52,500 blocks, it becomes spendable by the claim holder. Each additional claim on an output resets the counter to 52,500 blocks.

Now that these concepts are covered, we can describe the transaction structure. A transaction consists of a version, inputs, outputs and contingencies.

Version is the transaction version. This allows for multiple types of transactions to be interpreted by the validation. Currently there are only 2 versions:

- Version 1: landbase transaction
- Version 2: standard transaction

Inputs are the polygon outputs going into the transaction. In the case of a landbase transaction, there are no inputs. There are five types of inputs in the initial version of Bitland:

1. Standard transaction
2. Transacting a collateral output if the miner fee of its parent transaction was not paid
3. Making a claim on a UTXO (there can only be one claim per transaction)
4. Transacting an output if the transfer fee was of its parent transaction was not paid
5. Transacting a successful claim

Outputs are the polygon outputs created by the transaction. For all non-landbase transactions, the unioned input shape must equal the unioned output shape and a check on area must also be done to ensure that polygons are not double spent into the outputs. Output types are:

1. Landbase (which would be the only output in the transaction)
2. Standard
3. Collateral output
4. Output of a claim (polygon matches the input claim, this is the spendable output if the claim is successful)

Contingencies are the metadata associated with miner and transfer fees.

1. Miner fee in Satoshis
2. Miner fee blocks (less than or equal to 12,096 blocks)
3. Transfer fee in Satoshis
4. Transfer fee blocks (less than or equal to 12,096 blocks)
5. Transfer fee bitcoin address

Peer-to-peer Networking: There is essentially no difference in the information being shared in Bitland relative to Bitcoin. Nodes share blocks and transactions with each other as they receive them and can supply information about historical blocks when asked by a peer.