Intro:

IC is a new layer2 sidechain that we've developed and released in order to help BTC scale and be much more easily accessible and usable for billions of people. It is a real-time layer that uses the bitcoin network as a settlement layer and allows a user to deposit (peg-in), withdraw (peg-out) and transfer BTC instantly with nearly 0 fees (starting at 1 sat fees). While there exists payment channels (Lightning network) and a layer 2 settlement sidechain (Liquid network), a realtime layer 2 sidechain is lacking, which is the domain that this project seeks to serve. A realtime sidechain is needed to cover some of the drawbacks and limitations of the lightning and liquid networks.

Here is a very bird's eye view of where this protocol fits in:


Settlement
Layer 1
SWIFT
Layer 2
WELLS FARGO
CHASE
Liquid
by Blockstream
RSK
citi
BANK OF AMERICA
VISA
venmo
InstaChain
Zelle
Cash App
Real-time

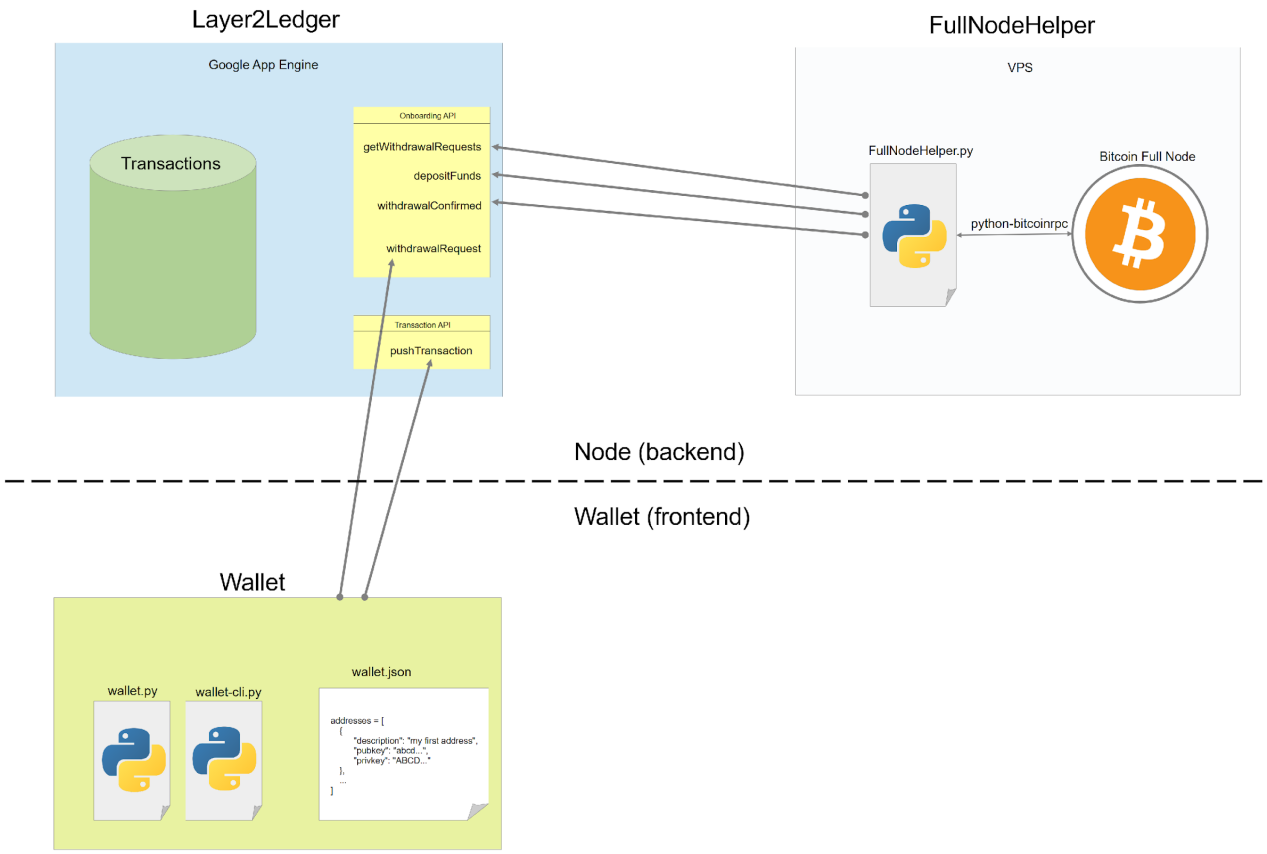
In the right is a very simplified visualization of the fiat payment infrastructure, and in the left is the (relatively very young) BTC infrastructure. As you can see financial applications are built on top of layers on each other, since this allows a diverse range of products each with their own benefits and uses cases. For example, apps like Venmo and Cash App allow easy and cheap payments, something that previously was costly to do between bank accounts. In this infrastructure, IC roughly corresponds to the Venmo/PayPal/Cash of the fiat world. Just like how from you bank account you can deposit and withdraw to & from Venmo, from IC you can also deposit and withdraw to the BTC network.

Use Cases:

The primary use cases of this protocol are to scale and enable worldwide adoption of BTC by providing instant and near-zero BTC transfers, however there are also many other use cases. Such a protocol would be ideal for remittances, micropayments, and merchant transactions (buying coffee) since the transaction fee is a flat amount.
One of the major benefit over the lightning network is easier and faster onboarding, meaning that if you want to get funds to a million users, you can do so in only one layer 1 transaction: deposit BTC to you layer 2 address once, then send a million layer 2 transactions to a million users. This can obviously be scaled to any amount of users and any value of funds. The same procedures on the lightning network would require a million layer 1 transactions, since it takes at least one transaction to establish a channel. Another advantage over the lightning network is that you can receive IC transactions to you address offline, just like regular cryptocurrencies.

Architecture:

Now that we've covered the high level overview of the protocol, here is the protocol level design of it:



There are two components, the node software and the wallet software. You can think of the node as the "backend" and the wallet as the "frontend". The node contains the layer 2 ledger, the database where all the layer 2 transactions are stored (and related tables) and makes available all the APIs that are required to transact on it. There are 3 basic types of layer2 transactions: transfer, deposit, and withdrawal. An IC account is just a public key/private key pair that users can generate through the wallet. The public key is the crypto 'address' that is used as the source and destination of a transaction, and the private key is kept in the wallet for signing the transaction. This is slightly different compared the BTC transaction model since BTC used the UTXO model, where the source is a transaction id + output number and the (one or more) destination is a public key hash. The IC model is more similar to an account based model like Ethereum where the source and destination are public keys.
The node itself comprises two parts, the user-facing Layer2Ledger project, which contains the APIs and the ledger database, and the FullNodeHelper project which is responsible for managing withdrawals and deposits by working with a BTC full node. The Layer2Ledger is ran as a serverless Google Cloud Project, while the FullNodeHelper is ran on a VPS on the same machine as a BTC full node, using the bitcoin-rpc API to interact with the BTC network. This is a protocol and not an 'app' or 'product', meaning anyone can run a node and start processing transactions; there is no permission required from ANYONE to run a node on this protocol, neither the project developers/management, nor existing nodes, nor any bank or other organization. All one needs is a bitcoin full node. The wallet can connect to any node, including having a balance on more than one node.

FAQ:

Is this custodial? Yes, the node has the private keys where the btc are deposited.
Can the devs censor transactions or block nodes? No. The transaction is transmitted by the user and processed by the node. The protocol devs prevent anyone from running a node or a wallet.
How much are the fees? There are no set fees specified by the protocol; technically it is determined by the node (nodes can reject the transaction if the user does not pay fees or the fee is too low), but the free market rate will always be slightly higher than the cost to process the transaction, which at this point is far lower than 1 sat.
What is the custodian method? Again, this is not specified by the protocol and is up the node operator. The keys can be held in a single account, a m-of-n multisig, a federated multisig, or any other method.
Can this sidechain be used for other cryptocurrencies? Yes! It is currently built on top of bitcoin, but can be easily ported to work on, for example, Eth or stablecoins. All you need to do is run an ethereum full node and handle deposits and withdrawals, everything else should remain the same.
How many transactions/second can the protocol process? Each node can handle about 60 trx/sec, however since each node is an independent ledger, the total throughput of the protocol is the sum of the throughput of all the nodes.
What is the transaction latency? Between 10-20 milliseconds. The reason it is this fast is because there is no consensus protocol and nodes do not need to broadcast a transaction to other nodes.
Do I need to have an account to use this network? No, all you need is a public/private key pair, which can be generated by the wallet. The wallet requires python 3.6 as well as some python libraries.
Is this anonymous? IC is pseudonymous just like BTC, meaning the source/destination are public keys that are not tied to any identity. You do not need to sign up or create any account to use it. However the ledger is public, so anyone can see all the transactions including deposits and withdrawals.
Is the mainnet version out? Right now, there is only a testnet version running. However the only difference between testnet and mainnet is the version that the btc full node daemon (bitcoind.exe) is running. To enable mainnet, bitcoind would have to be run on mainnet mode. There is no timeline on when to enable it; it could as soon as now but we want to have the testnet version testnet in production at least a little before launching the mainnet. This is not a technical roadblock, but rather a community one.
Is there a GUI or web wallet? As of this moment, the only way to use it is the python command line wallet, however once the wallet is ported to javascript, there will be a web as well as a desktop and mobile wallet (most likely through react).
Does this sidechain support smart contract or DeFi? No, this is only for payments, though DeFi capabilities are probable in the future.
Is there a roadmap? Currently, the biggest priority is maintaining a stable, bug free mainnet node out, then a UI wallet. After that it will probably be a community discussion.
Is there a token for this project? No. This is built directly on btc, there is no native token for this project.

How can I invest in this project? As of now, can either donate directly to the projects btc address. There is a possibility that I might put this project on gitcoin, from where you will be able to support us.