Bitcoin Integration

The Bitcoin integration on the Internet Computer makes it possible for the first time to create Bitcoin smart contracts, that is, smart contracts in the form of canisters running on the Internet Computer that make use of real bitcoin. This integration is made possible through two key components.

The first component is **chain-key signatures**, which enables every canister to obtain ECDSA public keys and get signatures with respect to these keys in a secure manner. Since Bitcoin addresses are tied to ECDSA public keys, having ECDSA public keys on a canister means that the canister can derive its own Bitcoin addresses. Given that the canister can request signatures for any of its public keys using the **IC ECDSA interface**, a canister can create Bitcoin transactions with valid signatures that move bitcoins from any of its Bitcoin addresses to any other address.

The second component is the integration with Bitcoin at the network level. The Internet Computer replicas have the capability to instantiate a so-called ***Bitcoin adapter***, a process external to the replica process. In a first step, the Bitcoin adapter collects information about nodes in the Bitcoin peer-to-peer network and, once sufficiently many Bitcoin nodes are discovered, it connects to 5 randomly chosen Bitcoin nodes. Since each replica in the subnet performs this operation, the entire subnet has many, mostly distinct connections to the Bitcoin network. The Bitcoin adapter uses the standard Bitcoin peer-to-peer protocol to get information about the Bitcoin blockchain. Each Bitcoin adapter keeps track of the full Bitcoin block header chain.

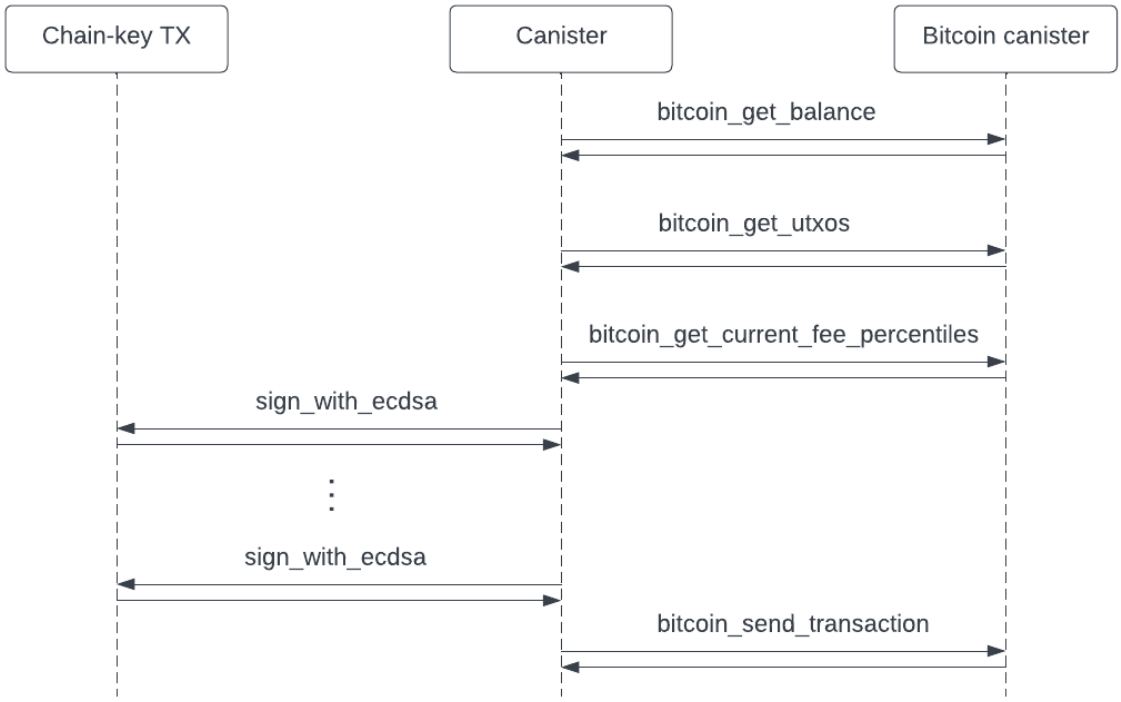
At the same time, the Bitcoin adapter communicates with the replica process to learn about the current Bitcoin state inside the replica. If the Bitcoin adapter learns that a Bitcoin block has not been made available to the replica yet by comparing the block header hashes provided by the replica against its locally available block header chain, the Bitcoin adapter requests the next missing block from the connected Bitcoin nodes and forwards it to the replica upon receipt.

Inside the replica, Bitcoin blocks received at the Networking layer are packed into IC blocks and processed in the Consensus and Message Routing layers and finally made available to the ***Bitcoin canister*** in the Execution layer. The Bitcoin canister is a canister running in a system

subnet whose purpose is to provide Bitcoin-related functionality to other canisters. In particular, it keeps information about the Bitcoin blockchain state and makes this information accessible to other canisters, such as the balance and unspent transaction outputs (UTXOs) for any address. Additionally, the fees of the most recent Bitcoin transactions that were put into blocks can be requested from the Bitcoin canister as well.

The Bitcoin canister also offers the last piece of crucial functionality: It provides an endpoint for canisters to send Bitcoin transactions, which are made available on the Networking layer where they are forwarded to the Bitcoin adapter. The Bitcoin adapter in turn advertises the transactions to its connected Bitcoin peers and transfers the transaction upon request. Since each replica in the subnet performs this step, every transaction can be dispersed quickly in the Bitcoin network.

The **IC management canister interface** provides access to all Bitcoin integration endpoints. Their use is illustrated in the following sample flow:



In this figure, a canister first requests the balance and then the UTXOs of a Bitcoin address. Next, it calls the fee endpoint to get recent fees. Lastly, the canister builds a Bitcoin transaction using some of the UTXOs as inputs. For each input, the ECDSA API is called to obtain the required signatures. Finally, the transaction is submitted.

Bitcoin integration wiki page.

Bitcoin canister source code.

Motion Proposal 20586.

Bitcoin integration goes live.