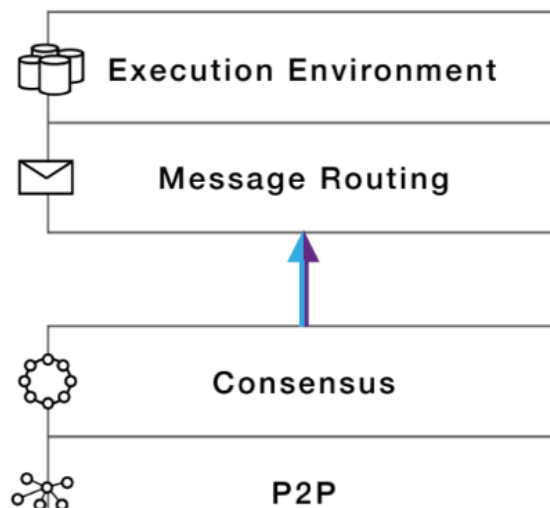


Core Internet Computer Protocol – Overview

A subnet hosts canister smart contracts and executes *messages* sent to them either by users or other canister smart contracts (which may be hosted on the same or another subnet). Messages on the IC are analogous to transactions on other blockchains. Messages addressed to a canister smart contract are executed by the nodes on the corresponding subnet by running the Wasm code of the canister. Canister code execution updates the canister state. In order to keep the state on the subnet nodes on which a canister is hosted in sync, it must be ensured that every node executes the same messages in the same order, i.e., fully deterministically. This is the core of the blockchain-based replicated state machine functionality realizing the IC.

Each node on the Internet Computer runs a replica process. The replica process is structured in a layered architecture consisting of the following 4 layers:

1. Peer-to-peer
2. Consensus
3. Message routing
4. Execution



The 4-layer architecture of the Core IC Protocol

The Peer-to-Peer layer is responsible for accepting messages from users and exchanging messages between nodes in a subnet. The consensus layer lets all the nodes on the subnet to agree on the messages to be processed, as well as their ordering. The message routing layer picks up the finalized blocks from consensus layer and routes the messages in the blocks to appropriate canisters. The execution layer deterministically executes canister code on the messages received from the messaging layer.

The upper two layers realize **deterministic execution** of the block of messages for a round received from the lower two layers, on each node of the subnet. At the beginning of a round, all (honest) nodes hold the same state, representing the replicated state of the subnet (which includes the current state on all canisters hosted on that subnet). By executing the messages of the next block received from consensus in a completely deterministic manner, it is ensured that the state after executing the messages of the block is the same on each node due to the determinism in execution.

Canister smart contracts can communicate with each other by sending messages, regardless of whether they are hosted on the same or different subnets. The IC core protocol handles both the inter-canister messages sent locally, i.e., on the same subnet, between canisters, as well as inter-canister messages sent across subnets, so called **XNet messages**. Local inter-canister messages do not need to go through consensus, while XNet inter-canister messages do (making the former more efficient in terms of throughput and incurring less latency).

The core IC protocol heavily relies on [chain-key cryptography](#) for its operation. A key component of chain-key cryptography is [chain-evolution technology](#), which facilitates the long-term operation of the IC, such as allowing new nodes to easily and securely join a subnet.