# Certified Variables

Why do you trust data obtained from a blockchain? Well, all transactions and the subsequent changes to smart contract state made their way through the blockchain consensus protocol, which guarantees correctness as long as the underlying trust assumptions hold. But verifying correctness based on the consensus protocol is tedious: A client has to download and validate the blockchain data. Even in the case of more efficient mechanisms such as **Bitcoin's SPV** or **Ethereum's light clients**, clients still have to perform significant amounts of work, such as downloading and validating block headers. This makes it difficult for applications with restricted uptime and resources, such as mobile or web applications, to operate on blockchain data without defaulting to centralized intermediaries.

The Internet Computer is different: Using **chain-key cryptography**, the Internet Computer can generate **digital signatures** that can be validated with a single, permanent public key belonging to the Internet Computer. Unlike with traditional digital signatures, however, the private key material *never* exists in a single place. It is always securely distributed between many different nodes, and valid signatures can only be generated when the majority of these nodes cooperates in a cryptographic protocol. A client application only has to embed the Internet Computer's public key, and can immediately validate all certified responses it receives from the Internet Computer, without putting any trust into the particular node it received the response from.

The Internet Computer's certification feature is exposed to canisters through *certified variables*. From an application perspective, certified variables can be set during an update call to a canister, when the canister changes its state during a transaction that went through consensus. The certificate can then be read in a subsequent query call, so the canister can respond to a client's request in a trustworthy way but without incurring the additional delay of consensus. Certified variables also underlie many of the Internet Computer's advanced features such as **certified assets** and **Internet Identity**.

More technically, each canister can specify a single 32-byte value that will be certified by the subnet. Well-known concepts such as **Merkle trees** or, more generally, **authenticated data structures** can be used to extend the certification from this single 32-byte value to arbitrary

amounts of data. Libraries such as **certified-map** make the feature easily accessible for developers.



[How Internet Computer Responses Are Certified as Authentic]

[Certified Variables & Assets]