The Move Book

The Object Model in Sui can be viewed as a high-level abstraction representing digital assets as objects . These objects have their own type and associated behaviors, a unique identifier, and support native storage operations like transfer and share . Designed to be intuitive and easy to use, the Object Model enables a wide range of use cases to be implemented with ease.

Objects in Sui have the following properties:

Type: Every object has a type, defining the structure and behavior of the object. Objects of different types cannot be mixed or used interchangeably, ensuring objects are used correctly according to their type system.

Unique ID: Each object has a unique identifier, distinguishing it from other objects. This ID is generated upon the object's creation and is immutable. It's used to track and identify objects within the system.

Owner: Every object is associated with an owner, who has control over changes to the object. Ownership on Sui can be exclusive to an account, shared across the network, or frozen, allowing read-only access without modification or transfer capabilities. We will discuss ownership in more detail in the following sections.

Note that ownership does not control the confidentiality of an object — it is always possible to read the contents of an on-chain object from outside of Move. You should never store unencrypted secrets inside of objects.

Data: Objects encapsulate their data, simplifying management and manipulation. The data structure and operations are defined by the object's type.

Version: The transition from accounts to objects is facilitated by object versioning. Traditionally, blockchains use a nonce to prevent replay attacks. In Sui, the object's version acts as a nonce, preventing replay attacks for each object.

Digest: Every object has a digest, which is a hash of the object's data. The digest is used to cryptographically verify the integrity of the object's data and ensures that it has not been tampered with. The digest is calculated when the object is created and is updated whenever the object's data changes.

Summary

Further reading