The Move Book

The Object Model allows for variable transaction execution paths, depending on the object's ownership type. The transaction execution path determines how the transaction is processed and validated by the network. In this section, we'll explore the different transaction execution paths in Sui and how they interact with the consensus mechanism.

At its core, blockchain technology faces a fundamental concurrency challenge: multiple parties may try to modify or access the same data simultaneously in a decentralized environment. This requires a system for sequencing and validating transactions to support the network's consistency. Sui addresses this challenge through a consensus mechanism, ensuring all nodes agree on the transactions' sequence and state.

Consider a marketplace scenario where Alice and Bob simultaneously attempt to purchase the same asset. The network must resolve this conflict to prevent double-spending, ensuring that at most one transaction succeeds while the other is rightfully rejected.

However, not all transactions require the same level of validation and consensus. For example, if Alice wants to transfer an object that she owns to Bob, the network can process this transaction without sequencing it with respect to all other transactions in the network, as only Alice has the authority to access the object. This is known as the fast path execution, where transactions accessing account-owned objects are processed quickly without the need for extensive consensus. No concurrent data access -> simpler challenge -> fast path.

Another ownership model that allows for fast path execution is the immutable state . Since immutable objects cannot change, transactions involving them can be processed quickly without the need to sequence them.

Transactions that do access shared state - on Sui it is represented with shared objects - require sequencing to ensure that the state is updated and consistent across all nodes. This is known as the execution through consensus , where transactions accessing shared objects are subject to the agreement process to maintain network consistency.

Lastly, it is important to mention that objects owned by other objects are subject to the same rules as the parent object. If the parent object is shared , the child object is also transitively shared. If the parent object is immutable, the child object is also immutable.

Concurrency Challenge

At its core, blockchain technology faces a fundamental concurrency challenge: multiple parties may try to modify or access the same data simultaneously in a decentralized environment. This requires a system for sequencing and validating transactions to support the network's consistency. Sui addresses this challenge through a consensus mechanism, ensuring all nodes agree on the transactions' sequence and state.

Consider a marketplace scenario where Alice and Bob simultaneously attempt to purchase the same asset. The network must resolve this conflict to prevent double-spending, ensuring that at most one transaction succeeds while the other is rightfully rejected.

However, not all transactions require the same level of validation and consensus. For example, if Alice wants to transfer an object that she owns to Bob, the network can process this transaction without sequencing it with respect to all other transactions in the network, as only Alice has the authority to access the object. This is known as the fast path execution, where transactions accessing account-owned objects are processed quickly without the need for extensive consensus. No concurrent data access -> simpler challenge -> fast path.

Another ownership model that allows for fast path execution is the immutable state . Since immutable objects cannot change, transactions involving them can be processed quickly without the need to sequence them.

Transactions that do access shared state - on Sui it is represented with shared objects - require sequencing to ensure that the state is updated and consistent across all nodes. This is known as the execution through consensus , where transactions accessing shared objects are subject to the agreement process to maintain network consistency.

Lastly, it is important to mention that objects owned by other objects are subject to the same rules as the parent object. If the parent object is shared , the child object is also transitively shared. If the parent object is immutable, the child object is also immutable.

Fast Path

However, not all transactions require the same level of validation and consensus. For example, if Alice wants to transfer an object that she owns to Bob, the network can process this transaction without sequencing it with respect to all other transactions in the network, as only Alice has the authority to access the object. This is known as the fast path execution, where transactions accessing account-owned objects are processed quickly without the need for extensive consensus. No concurrent data access -> simpler

challenge -> fast path.

Another ownership model that allows for fast path execution is the immutable state . Since immutable objects cannot change, transactions involving them can be processed quickly without the need to sequence them.

Transactions that do access shared state - on Sui it is represented with shared objects - require sequencing to ensure that the state is updated and consistent across all nodes. This is known as the execution through consensus , where transactions accessing shared objects are subject to the agreement process to maintain network consistency.

Lastly, it is important to mention that objects owned by other objects are subject to the same rules as the parent object. If the parent object is shared , the child object is also transitively shared. If the parent object is immutable, the child object is also immutable.

Consensus Path

Transactions that do access shared state - on Sui it is represented with shared objects - require sequencing to ensure that the state is updated and consistent across all nodes. This is known as the execution through consensus , where transactions accessing shared objects are subject to the agreement process to maintain network consistency.

Lastly, it is important to mention that objects owned by other objects are subject to the same rules as the parent object. If the parent object is shared , the child object is also transitively shared. If the parent object is immutable, the child object is also immutable.

Objects owned by Objects

Lastly, it is important to mention that objects owned by other objects are subject to the same rules as the parent object. If the parent object is shared , the child object is also transitively shared. If the parent object is immutable, the child object is also immutable.

Summary