

The Move Book

Transaction is a fundamental concept in the blockchain world. It is a way to interact with a blockchain. Transactions are used to change the state of the blockchain, and they are the only way to do so. In Move, transactions are used to call functions in a package, deploy new packages, and upgrade existing ones.

Every transaction explicitly specifies the objects it operates on!

Transactions consist of:

Transaction inputs are the arguments for the transaction and are split between 2 types:

Sui transactions may consist of multiple commands. Each command is a single built-in command (like publishing a package) or a call to a function in an already published package. The commands are executed in the order they are listed in the transaction, and they can use the results of the previous commands, forming a chain. Transaction either succeeds or fails as a whole.

Schematically, a transaction looks like this (in pseudo-code):

In this example, the transaction consists of three commands:

Transaction effects are the changes that a transaction makes to the blockchain state. More specifically, a transaction can change the state in the following ways:

The result of the executed transaction consists of different parts:

Transaction Structure

Every transaction explicitly specifies the objects it operates on!

Transactions consist of:

Transaction inputs are the arguments for the transaction and are split between 2 types:

Sui transactions may consist of multiple commands. Each command is a single built-in command (like publishing a package) or a call to a function in an already published package. The commands are executed in the order they are listed in the transaction, and they can use the results of the previous commands, forming a chain. Transaction either succeeds or fails as a whole.

Schematically, a transaction looks like this (in pseudo-code):

```
```bash Inputs: - sender = 0xa11ce
```

```
Commands: - payment = SplitCoins(Gas, [1000]) - item = MoveCall(0xAAA::market::purchase, [payment]) -
TransferObjects(item, sender) ```
```

In this example, the transaction consists of three commands:

Transaction effects are the changes that a transaction makes to the blockchain state. More specifically, a transaction can change the state in the following ways:

The result of the executed transaction consists of different parts:

## Inputs

Transaction inputs are the arguments for the transaction and are split between 2 types:

Sui transactions may consist of multiple commands. Each command is a single built-in command (like publishing a package) or a call to a function in an already published package. The commands are executed in the order they are listed in the transaction, and they can use the results of the previous commands, forming a chain. Transaction either succeeds or fails as a whole.

Schematically, a transaction looks like this (in pseudo-code):

```
```bash Inputs: - sender = 0xa11ce
```

```
Commands: - payment = SplitCoins(Gas, [ 1000 ]) - item = MoveCall(0xAAA::market::purchase, [ payment ]) -
```

```
TransferObjects(item, sender) ``
```

In this example, the transaction consists of three commands:

Transaction effects are the changes that a transaction makes to the blockchain state. More specifically, a transaction can change the state in the following ways:

The result of the executed transaction consists of different parts:

Commands

Sui transactions may consist of multiple commands. Each command is a single built-in command (like publishing a package) or a call to a function in an already published package. The commands are executed in the order they are listed in the transaction, and they can use the results of the previous commands, forming a chain. Transaction either succeeds or fails as a whole.

Schematically, a transaction looks like this (in pseudo-code):

```
``bash Inputs: - sender = 0xa11ce
```

```
Commands: - payment = SplitCoins(Gas, [ 1000 ]) - item = MoveCall(0xAAA::market::purchase, [ payment ]) -  
TransferObjects(item, sender) ``
```

In this example, the transaction consists of three commands:

Transaction effects are the changes that a transaction makes to the blockchain state. More specifically, a transaction can change the state in the following ways:

The result of the executed transaction consists of different parts:

Transaction Effects

Transaction effects are the changes that a transaction makes to the blockchain state. More specifically, a transaction can change the state in the following ways:

The result of the executed transaction consists of different parts: