Packages

A Move package on Sui includes one or more modules that define that package's interaction with on-chain objects. You develop the logic for those modules using Move, which you then compile into an object. Finally, you publish your package object onto a Sui network. On chain, anyone can view your package contents and the logic it employs to manipulate other on-chain objects using a Sui network explorer.

After you publish a package object on chain to a network, it lives there forever. You cannot directly change the code of an on-chain package object. After a package object appears on chain, other packages can use the modules that the original package provides. Bad actors aside, imagine if someone changed the logic in their on-chain package to correct unintentional behavior that was overlooked during development. This would cause a ripple effect, changing the logic of every package that uses that module to perform in ways the developers might have never intended.

While you can't manipulate on-chain packages directly, you do have the ability to upgrade them. Upgrading on-chain packages provides a way to improve your code or add features without affecting packages that use the original package. When you upgrade a package, you're creating a new object on chain instead of modifying the original package. See Upgrading Packages to learn more about the process.

The Sui Client CLI upgrade command offers an approachable way to upgrade packages when the CLI active address owns the UpgradeCap object associated with those packages.

Using the Sui CLI is useful to get started with upgrades, or in the early stages of package development, but protecting the ability to upgrade a package on chain using a single key can pose a security risk for several reasons:

You can make a package immutable when it goes live to mitigate the single-key risk using the Move sui::package::make_immutable function to destroy its UpgradeCap . Making the package immutable, however, prevents future bug fixes and new features, which might not be practical or desired.

To protect your package from single-key risk on chain, see Custom Upgrade Policies .

Packages are immutable

After you publish a package object on chain to a network, it lives there forever. You cannot directly change the code of an on-chain package object. After a package object appears on chain, other packages can use the modules that the original package provides. Bad actors aside, imagine if someone changed the logic in their on-chain package to correct unintentional behavior that was overlooked during development. This would cause a ripple effect, changing the logic of every package that uses that module to perform in ways the developers might have never intended.

While you can't manipulate on-chain packages directly, you do have the ability to upgrade them. Upgrading on-chain packages provides a way to improve your code or add features without affecting packages that use the original package. When you upgrade a package, you're creating a new object on chain instead of modifying the original package. See Upgrading Packages to learn more about the process.

The Sui Client CLI upgrade command offers an approachable way to upgrade packages when the CLI active address owns the UpgradeCap object associated with those packages.

Using the Sui CLI is useful to get started with upgrades, or in the early stages of package development, but protecting the ability to upgrade a package on chain using a single key can pose a security risk for several reasons:

You can make a package immutable when it goes live to mitigate the single-key risk using the Move sui::package::make_immutable function to destroy its UpgradeCap . Making the package immutable, however, prevents future bug fixes and new features, which might not be practical or desired.

To protect your package from single-key risk on chain, see Custom Upgrade Policies .

Upgrading packages

While you can't manipulate on-chain packages directly, you do have the ability to upgrade them. Upgrading on-chain packages provides a way to improve your code or add features without affecting packages that use the original package. When you upgrade a package, you're creating a new object on chain instead of modifying the original package. See Upgrading Packages to learn more about the process.

The Sui Client CLI upgrade command offers an approachable way to upgrade packages when the CLI active address owns the UpgradeCap object associated with those packages.

Using the Sui CLI is useful to get started with upgrades, or in the early stages of package development, but protecting the ability to upgrade a package on chain using a single key can pose a security risk for several reasons:

You can make a package immutable when it goes live to mitigate the single-key risk using the Move sui::package::make_immutable function to destroy its UpgradeCap . Making the package immutable, however, prevents future bug fixes and new features, which might not be practical or desired.

To protect your package from single-key risk on chain, see Custom Upgrade Policies .

Using Sui Client CLI to upgrade packages

The Sui Client CLI upgrade command offers an approachable way to upgrade packages when the CLI active address owns the UpgradeCap object associated with those packages.

Using the Sui CLI is useful to get started with upgrades, or in the early stages of package development, but protecting the ability to upgrade a package on chain using a single key can pose a security risk for several reasons:

You can make a package immutable when it goes live to mitigate the single-key risk using the Move sui::package::make_immutable function to destroy its UpgradeCap . Making the package immutable, however, prevents future bug fixes and new features, which might not be practical or desired.

To protect your package from single-key risk on chain, see Custom Upgrade Policies .