

Fee abstraction

You can use the FeeGrant module or Smart contract based fee abstraction to simplify the usage of your dApp as explained [here!](#)

FeeGrant

Secret also allows for Gas abstraction for users by leveraging the CosmosSDK [FeeGrant module](#) . This module allows one to submit transactions where a different wallet is paying the gas fees as long as they granted you a budget to do that.

Documentation to create FeeGrant functionality in your UI are [here](#) .

This tool is widely used in different Secret UIs (for ex [Secret dashboard](#)) and there is a community run FeeGrant faucet available for dApps to use. - [Faucet](#) - [Code](#)

Smart contract based

You can use Smart-contracts with Fees deposited in them to abstract fee usage and automate tasks for users - An example of this is the [Sienna rewards contract](#) which deposits users rewards into their wallet daily.

There is also the feature of "opportunistic execution" which allows additional privacy to perform automated actions by using leftover gas from the Gas evaporation function in the Secret compute module. - [Here is a great guide on opportunistic execution!](#)

Last updated 3 months ago On this page * [FeeGrant](#) * [Smart contract based](#)

Was this helpful? [Edit on GitHub](#) [Export as PDF](#)