

# Infrastructure Models

This article describes the Circle infrastructure models so you can choose which approach to use for your development.

[Suggest Edits](#)

This article delves into the options available for wallet infrastructure models and helps you decide based on your end-users' needs. The focus is on understanding the critical characteristics of each model to guide developers in selecting the right approach for their application.

Video Tutorial: Understanding the Types of Programmable Wallets

Watch this video for an insightful overview of the infrastructure models available with our Programmable Wallets. Learn the distinctions and advantages between user-controlled and developer-controlled wallets and how each model can be leveraged to suit your application's requirements.

[undefined](#)

## Wallet Infrastructure Model Types

Infrastructure models govern who controls the assets in a given wallet. Businesses and individuals need to understand the characteristics of each type of infrastructure to determine the correct wallet for their needs.

Circle Programmable Wallets offer two infrastructure models: user-controlled wallets and developer-controlled wallets. Each model provides distinct advantages to suit different requirements.

### User-Controlled Wallets

User-controlled wallets grant full autonomy to app users, enabling them to have complete control over their wallets. Users set up a PIN code for authorization and have sole access to their private keys. This ensures privacy and prevents unauthorized access or movement of assets. The user-controlled wallet setup prioritizes security while providing a familiar and seamless Web3 user experience.

Start building user-controlled wallets with the [quickstart guide](#) !

Limitations:

- Mobile SDKs are native and do not support frameworks like React Native, Flutter, Apache Cordova, or Ionic.

### Developer-Controlled Wallets

Developer-controlled wallets allow developers to retain control over their users' wallets while streamlining user interactions with the Web3 ecosystem. These wallets facilitate various blockchain interactions on behalf of the users, such as sending and receiving digital assets, smart contract interactions, and NFT minting. With developer-controlled wallets, users can enjoy the benefits of the blockchain without needing to understand its complexities or manage private keys.

Get started with the [Developer-Controlled Wallets quickstart](#) .

Limitations:

- There are no app restrictions for Developer-Controlled Wallets.

## Choosing your Infrastructure Model

The critical distinction between user-controlled and developer-controlled wallets is which party controls the ability to access the private key used to interact with the blockchain.

User-controlled and developer-controlled wallets can make transactions, interact with smart contracts, and sign messages. In the case of a user-controlled wallet, the user controls these abilities. In the case of a developer-controlled wallet, the developer has this control.

## Why choose User-Controlled Wallets?

User-controlled wallets give your app's users access and control over the underlying private keys of their wallets. Circle provides a Web3 user experience, using a PIN code mechanism and a set of security questions for account recovery, abstracting the complexity of seed phrases from your users. Additionally, Circle uses best-in-class security approaches to power its Programmable Wallets with secure multi-party computation.

## PIN Code Responsibility for User-controlled wallets

Users invoke usage of their private keys through their specified PIN code, which is then used to encrypt and decrypt access to the private keys upon invoking a transaction on the blockchain. Your users will be ultimately responsible for accessing their wallets, so they must understand the need to remember their PIN code and the answers to their security questions.

With user-controlled wallets, your users have access to the speed and security of the blockchain but with a much more familiar user experience.

## Why choose Developer-Controlled Wallets?

Developer-controlled wallets allow you to create and manage asset flows on behalf of your users, providing them with easy-to-use access to Web3 functionality. These wallets allow you to control virtually all blockchain interactions for your users, like sending and receiving digital assets, smart contract interactions, and NFT minting.

By using developer-controlled wallets, your users do not need to understand the complexities of the Web3 ecosystem or worry about losing their assets because of a lost seed phrase to access the speed of the blockchain.

This article explores the various options available for wallets, taking into account both the account type and the infrastructure model. The infrastructure model determines who controls the assets in the wallet, while the account type determines the underlying structure of the wallet itself. Understanding these concepts is essential for developers to make informed decisions that align with their specific use cases. Updated 3 days ago

### What's Next

- [Account Types](#)
- [Table of Contents](#)
- - [Wallet Infrastructure Model Types](#)
- - - [User-Controlled Wallets](#)
- - - [Developer-Controlled Wallets](#)
- - [Choosing your Infrastructure Model](#)
- - [Why choose User-Controlled Wallets?](#)
- - - [PIN Code Responsibility for User-controlled wallets](#)
- - [Why choose Developer-Controlled Wallets?](#)