Smart Contract Platform

Suggest Edits

The Smart Contract Platform (SCP) enables developers to utilize on-chain smart contracts in their apps intuitively. It allows them to facilitate the creation, deployment, and execution of smart contracts via the Developer Console or APIs, ensuring flexibility and ease of integration. The goal of SCP is to make it easy for developers to use smart contracts to get real-world utility into their apps.

Some examples of what developers can do using the SCP:

1. Deploying Custom Contracts:
2. Bring your custom contracts to life by efficiently deploying them onto the blockchain network, expanding the functionality and capabilities of your application.
3. Deploying NFT Contracts:
4. Easily deploy NFT contracts and programmatically mint unique tokens for end-users, enabling the creation of digital assets and collectibles.
5. Creating On-Chain Loyalty Programs:
6. Establish on-chain loyalty programs within your applications, allowing users to earn and redeem rewards seamlessly.
7. Interacting with DeFi Projects:
8. Effortlessly integrate with popular DeFi projects like Uniswap, enabling users to interact with decentralized exchanges and other financial services with just a few clicks.
9. Integrating Circle Contracts:
10. Seamlessly incorporate Circle contracts, such as CCTP (Circle Contract Transfer Protocol), into your application, providing secure and efficient transfer functionalities.

Explore Smart Contracts

The web3 services console allows developers to view the details of any smart contract. Developers can import a smart contract by adding the contract's address and chain. Once imported, developers can explore the contract, view the ABI functions, read the source code, and see all transactions associated with the contract.

Coming soon

We are working on introducing the Magic Decoder, which will decode common interfaces of a contract for you to use via the console or in your app. We are also developing contract event features and exporting code in your preferred integrated development environment (IDE).

Deploy Smart Contracts

Developers can deploy smart contracts using SCP by writing their contracts or using pre-vetted templates provided by Circle. To deploy a contract, developers need to create a Developer-controlled wallet using Circle's Prog wallets and use it to deploy the contract across any supported chains. Deployment can be done through the console in a no-code way or programmatically using APIs.

Deploying a Custom Contract

If you have already written a smart contract on an IDE, you can deploy it by providing the compiled bytecode and ABI. For console deployment, create a console wallet (a smart contract wallet) and use it to deploy the contract on the desired chain. Include the source code, ABI, and bytecode for API deployment in the request parameters.

Deploying with Templates

For developers unfamiliar with smart contract engineering, Templates provide code snippets to deploy contracts without writing any solidity code. These templates, curated by the Circle team and audited by third-party auditors, cover popular on-chain use cases. Fill in the required properties and deploy the contract. Templates can be deployed through the console or APIs.

Some templates that we support include:

- Token (ERC-20) contract, by Thirdweb
- NFT (ERC-721) contract, by Thirdweb
- NFT (ERC-1155) contract, by Thirdweb

What's next: We are continuously improving the deployment experience and will soon support using SCP via your favorite

IDE, deploying proxy contracts, and deploying gasless contracts. We will also be launching new templates

Manage Smart Contracts

Once deployed, you can use the console to manage your smart contracts. This includes analytics, calling contract functions, and changing ownership. The console provides an easy way to update and manage contracts post-deployment. Developers can access analytics such as transactions and events for contracts deployed using SCP via APIs.

Interact with Smart Contracts

Interacting with smart contracts allows you to integrate existing on-chain contracts into your applications. Import the contract and explore its various functions. You can add parameters and generate API resources to interact with the contract in a straightforward manner. Updated16 days ago

What's Next Head over to our quickstart and deploy your first smart contract! *Quickstarts * Table of Contents * * Explore Smart Contracts * * Deploy Smart Contracts * * * Deploying a Custom Contract * * * Deploying with Templates * * Manage Smart Contracts * * Interact with Smart Contracts