ERC Standards Reference

The Ethereum Request for Comments standards you can build on, and how to choose between them.

VERSION	LAST UPDATED	STATUS
0.1	November 17, 2026	Draft

1. What ERCs are

An ERC is an application-level standard that defines an interface, so that wallets, exchanges, marketplaces and other contracts can work with your contract without knowing anything else about it. Neo Smart Chain runs the same EVM, so the same ERCs apply here as on Ethereum and on layer 2 networks such as Base and Robinhood Chain.

Use a standard whenever one fits. Standard contracts are recognised automatically by wallets and explorers, and they are far easier to audit than custom code.

2. Standards at a glance

Standard	Purpose	Use it for
ERC-20	Fungible tokens	Currencies, stablecoins, governance and utility tokens
ERC-721	Non-fungible tokens	Unique items, tickets, certificates, identity
ERC-1155	Multi-token standard	Games and collections that mix fungible and unique items
ERC-4626	Tokenized vaults	Yield, lending and staking shares
ERC-2612	Permit	Gasless token approvals using signatures

ERC-165	Interface detection	Letting other contracts discover what yours supports
ERC-2981	NFT royalties	Telling marketplaces what royalty to pay
ERC-1967	Proxy storage slots	Upgradeable contracts
ERC-5267	EIP-712 domain	Wallets reading signing domains
ERC-4337	Account abstraction	Smart accounts and sponsored gas

> ERC-4337 and any feature that depends on shared infrastructure such as an EntryPoint contract or a bundler needs network support. Availability on Neo Smart Chain: [ERC-4337 availability]. Check before building on it.

3. ERC-20: fungible tokens

Every unit of an ERC-20 token is identical and interchangeable. The standard defines balanceOf, transfer, approve, allowance and transferFrom, plus the Transfer and Approval events.

- Use OpenZeppelin ERC20 instead of writing your own. Add only the extensions you need, such as ERC20Permit, ERC20Burnable, ERC20Votes or ERC20Pausable.
- Choose decimals deliberately. 18 is the common default, and 6 is common for dollar-pegged tokens. Wallets and exchanges read this value, so never change it after launch.
- Do not rely on approvals being set to an exact amount. Frontends should request the amount needed, not unlimited approval.
- Document any admin power: minting, burning, pausing, blacklisting or upgrading. Holders and reviewers expect to find it in your documentation.

4. ERC-721 and ERC-1155: collectibles and items

ERC-721 gives each token a unique ID and a single owner. ERC-1155 manages many token types in one contract and supports batch transfers, which lowers gas cost for games and large collections.

- Return metadata from tokenURI (ERC-721) or uri (ERC-1155) as a JSON document with name, description and image fields.
- Host metadata and images at a stable address. Content-addressed storage keeps the data from changing without notice.

- Use safeTransferFrom and implement the receiver interfaces if a contract needs to hold tokens, so tokens cannot be locked by accident.
- Implement ERC-2981 if you want marketplaces to know your royalty, and remember that enforcement is up to each marketplace.

5. ERC-4626: vaults

ERC-4626 standardises contracts that accept an ERC-20 asset and issue shares in return. A single interface covers deposit, mint, withdraw and redeem, so aggregators can integrate every vault the same way.

- Protect against the first-depositor inflation attack. Use a virtual offset, as the OpenZeppelin implementation does, or seed the vault at deployment.
- Round in favour of the vault, not the user, when converting between assets and shares.
- Do not use a vault share price as an oracle without a manipulation check.

6. Permit, signatures and EIP-712

ERC-2612 lets a holder approve spending with a signature instead of a transaction. This removes the separate approval step and lets an application sponsor gas. It relies on EIP-712 typed data, whose domain includes the chain ID, so a signature made for another chain cannot be replayed on Neo Smart Chain.

- Always read the chain ID from the chain, not from a constant, so signatures stay valid if the chain ID differs between testnet and mainnet.
- Use deadlines on every signature and track nonces to prevent replay.
- If you deploy the same token on several chains, each deployment must use its own domain.

7. Choosing a standard

If you are building	Start with
A currency or stablecoin	ERC-20 with ERC-2612 permit
A governance token	ERC-20 with ERC20Votes
A collection of unique items	ERC-721, plus ERC-2981 for royalties
A game with many item types	ERC-1155

A yield or staking product	ERC-4626 over an ERC-20 asset
An upgradeable system	ERC-1967 proxy through a UUPS or transparent proxy

Before launching any token, read the listing guide to see which documents and metadata the Neo Smart Chain registry expects.

This document is for information only and is not financial, legal or tax advice. Highlighted items in square brackets are still to be confirmed by the issuer. The latest version is always available at neosmartchain.com/developers/erc-standards