

Chainlink Functions Service Responsibility

Chainlink Functions provides access to trust-minimized compute infrastructure that allows you to retrieve data and run computation. Because the service is highly-flexible and runs offchain, both developers and Chainlink service providers share responsibility in ensuring that operation and performance match expectations.

Developer responsibilities

Developers who implement Chainlink Functions in their code and applications are responsible for several components.

Data

- **Data quality:**When using Chainlink Functions, developers are responsible for ensuring that any external data they retrieve meet the data quality requirements for their applications. This responsibility applies to data retrieved via APIs or other data retrieval mechanisms. You must ensure that the data sources you consume through Chainlink Functions are accurate, reliable, secure, and not at risk of manipulation by malicious actors. When possible, use multiple data sources to reduce single points of failure and manipulation risks.
- **Data availability:**Developers must ensure that the data sources and APIs that they use with Chainlink Functions meet the fault-tolerance and availability requirements for their applications. Node operators are geographically distributed, so developers must ensure that APIs do not have geographic or other restrictions that would prohibit node operators from retrieving data. When possible, use redundant data sources to reduce the risk that your applications cannot execute due to unavailable data. In situations where data isn't available, ensure proper error handling.
- **Data privacy and ethics:**Developers are responsible for determining what data they have a right to use with Chainlink Functions and ensuring that they use that data ethically. Developers must ensure that their Chainlink Functions code does not expose their private information or the private information of users without proper consent. Do not use data that you are not authorized to access. Your Chainlink Functions code and your application must ensure that private or sensitive information remains secure.

Code

- **Code quality and reliability:**Developers must execute code on Chainlink Functions only if the code meets the quality and reliability requirements for their use case and application.
- **Code and application audits:**Developers are responsible for auditing their code and applications before deploying to production. You must determine the quality of any audits and ensure that they meet the requirements for your application and any code that runs on Chainlink Functions.
- **Code dependencies and imports:**Developers are responsible for ensuring the quality, reliability, and security of any dependencies or imported packages that they use with Chainlink Functions. Review and audit these dependencies and packages.

Secrets

caution

Chainlink Functions is still in BETA. The use of secrets in your requests is an experimental feature that may not operate as expected and is subject to change. Use of this feature is at your own risk and may result in unexpected errors, possible revealing of the secret as new versions are released, or other issues.

- **Self-hosted secrets:** Developers are responsible for securing self-hosted secrets, monitoring unauthorized access, auditing permissions, and ensuring that secrets are available for retrieval by Chainlink Functions when executing code.
- **Secrets best practices:** For all types of secrets used with Chainlink Functions, developers must follow common best practices for managing secrets for applications. Developers are responsible for selecting, setting expiration time, monitoring, and rotating secrets to ensure the security of their applications and Chainlink Functions code.

Subscriptions

- **Subscription owner wallet management:**Developers must ensure the security of any wallets that own Chainlink Functions subscriptions or wallets that secure funds for subscriptions.
- **Subscription balances:**Subscription owners are responsible for maintaining the Chainlink Function balance that is necessary to fund Chainlink Functions requests and computation. Monitor your subscription balance and implement the necessary processes to fund your subscription balance at a level that meets your application's requirements.

Node Operator responsibilities

High-quality node operators participate in the Functions DONs using a configuration specified in the Chainlink software. As participants in these deployments, Node Operators are responsible for the following components of Chainlink Functions:

- Ensuring the proper configuration, maintenance, and monitoring of nodes participating in the Chainlink Functions DON.
- Storing encrypted secrets that developers provide using threshold encryption.
- Ensuring that transactions execute onchain in a timely manner and apply gas bumping when necessary.
- Selecting and properly employing blockchain clients to connect to supported blockchain networks.
- Maintaining continuous uptime and active participation in OCR consensus.
- Employ defensive measures to prevent unauthorized access to their Chainlink node deployments.
- Ensure that Chainlink node deployments are running the latest software versions.
- Responding to important communication from Chainlink Labs or from other node operators in a timely manner.