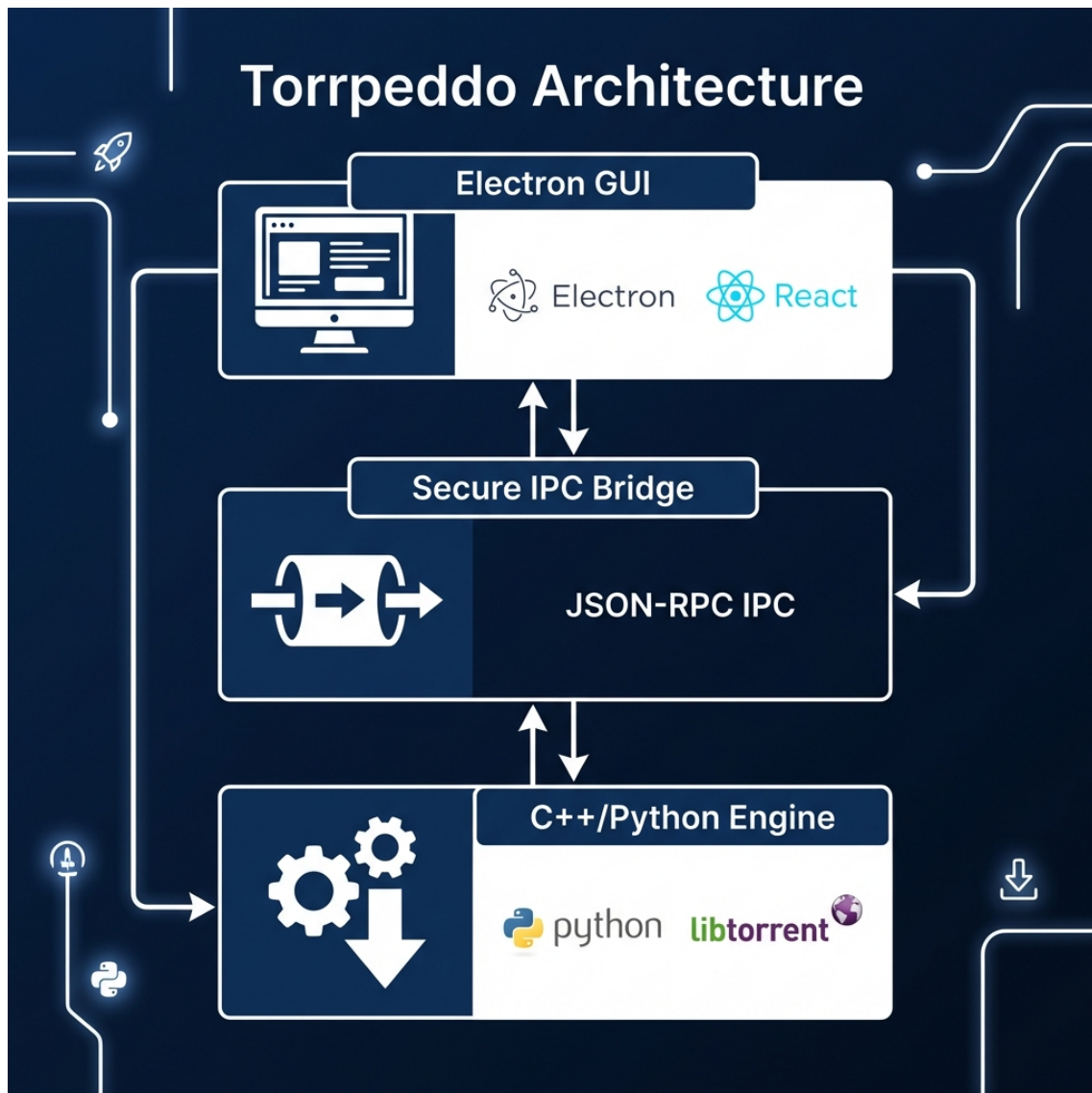


TORRPEDDO PROJECT BOOK



Executive Summary

Architectural Deep Dive

Torpeddo follows a decoupled architectural pattern, separating the presentation layer from the core logic and network engine. This is achieved through three primary layers:

1. Frontend: Electron Framework

Benefits for Torpeddo include native desktop and web components, Cross-Platform Compatibility across Linux, Windows, and macOS, and access to OS features like file system and network capabilities.

2. The Bridge: IPC (Inter-Process Communication)

Implementation: The frontend communicates with the backend via a custom IPC protocol over stdio. Backend logic is executed in a Python child process, with requests converted into JSON strings and responses parsed back into Python objects.

Why this approach? - Decoupling: The engine can be updated, debugged, or even replaced without touching the UI. - Security: The backend runs as a separate process, providing a layer of isolation. - Performance: High-speed communication with minimal overhead compared to HTTP-based local servers.

3. Backend Engine: Python & libtorrent

Multi-threaded Python backend leveraging libtorrent for internal thread pooling and parallel processing. Utilizes Python's `TorrentModule` for efficient data handling and includes safety features like memory management and error handling, ensuring stability and security from the DHT or peers.

Development Process & Methodology
