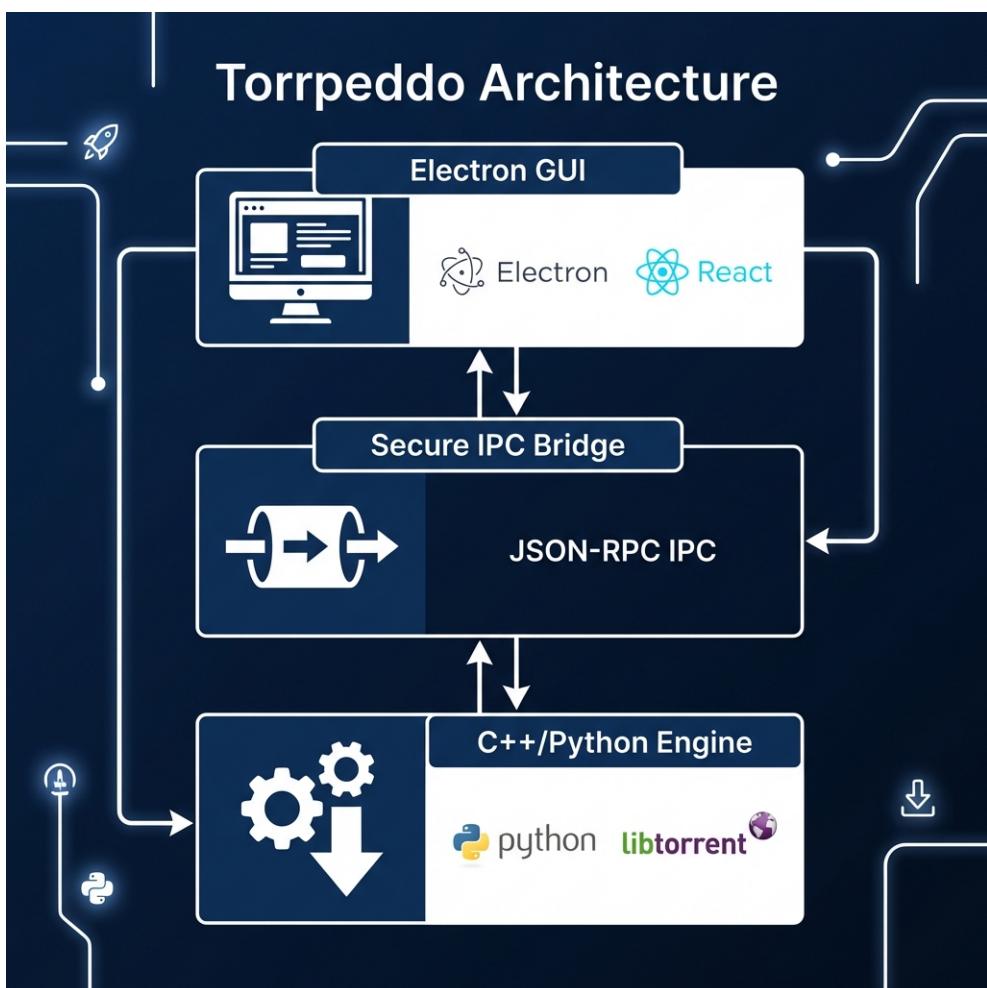


TORRPEDDO PROJECT BOOK



Executive Summary

Torrpeddo is an industrial-grade, premium torrent client designed for the modern desktop. Built primarily with Python and the Electron framework, Torrpeddo leverages the powerful `libtorrent` suite to offer a seamless, high-performance experience that bridges the gap between complex network protocols and professional user interfaces.

Architectural Deep Dive

Torrpeddo 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

What is Electron?

Electron is an open-source framework that allows developers to build cross-platform desktop applications using web technologies (HTML, CSS, and JavaScript) and the Node.js runtime.

Benefits for Torrpeddo:

- Visual Excellence: Leveraging web components to create a "native" UI.

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

What is IPC?

IPC, or Inter-Process Communication, is a mechanism for processes to share data and synchronize their operations. In Torrpeddo, it's used to create an IPC bridge to connect the Frontend and Backend.

Implementation: Secure JSON-RPC

Communication is handled via a secure JSON-RPC interface.

Why this approach?

- Decoupling: The engine and UI communicate through the IPC bridge, avoiding the UI directly touching the UI.

3. Backend Engine: Python & libtorrent

The Core: libtorrent with Python Bindings

At the heart of Torrpeddo is the libtorrent library, a C++ implementation. While the core logic is written in C++, Torrpeddo utilizes the Python Bindings to provide a seamless integration with Python.

Multi-threaded Performance:

- Engine Level: The `libtorrent` engine is highly optimized for multi-threaded operations, handling disk I/O, network polling, and the processing of multiple torrent fragments simultaneously.

Development Process & Methodology

The Torrpeddo project followed a "Platform-First" methodology:

1. Language Choice: Python was chosen due to its readability, ease of integration with libtorrent, and availability of many useful libraries.

(c) 2026 Torrpeddo Team. All rights reserved.