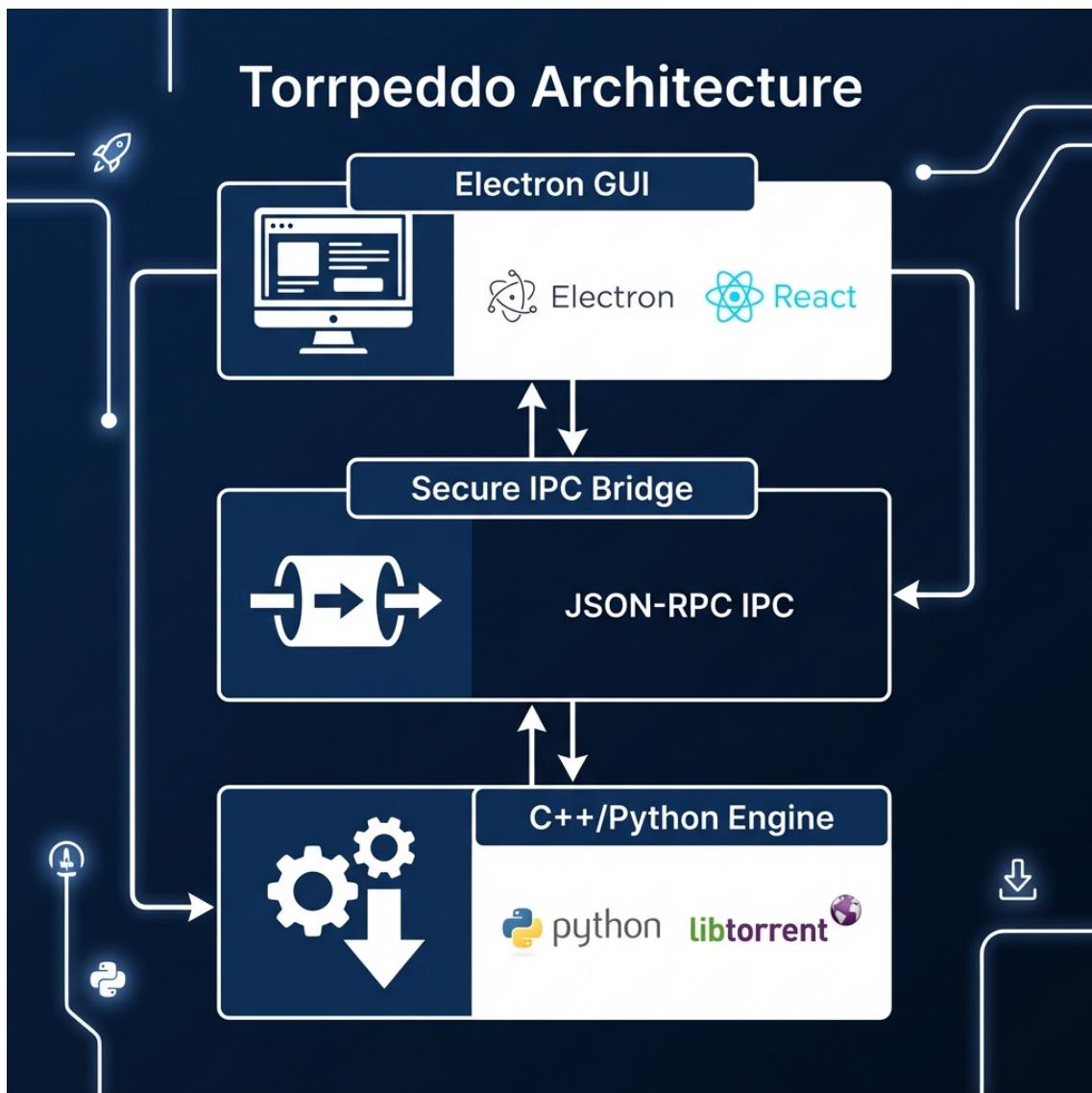


# TORRPEDDO PROJECT BOOK



## Executive Summary

Torpeddo is an industrial-grade, premium torrent client designed for the modern desktop. Built primarily with Python and the Electron framework, Torpeddo 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

Torpedo 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 for building cross-platform desktop applications using JavaScript. It consists of a runtime (for system integration) and a UI library.

Benefits for Torpedo:

- Visual Excellence
- Components to...

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

What is IPC?

IPC, or Inter-Process Communication, is a mechanism to share data and synchronize processes between different parts of a system. It connects the Electron front...

Implementation: Secure JSON-RPC

Communication interface

Why this approach?

- Decoupling:...

touching the UI.

### **3. Backend Engine: Python & libtorrent**

The Core: libtorrent with Python Bindings

At the heart of Torrpedo's performance is the libtorrent library. While the underlying C++ code is complex, the Python bindings utilize the official libtorrent API, making it easier for developers to work with the bridge layer.

Multi-threaded Performance:

- Engine Level Parallelism
- I/O, network polling, and file operations are handled by multiple torrent threads.

---

## **Development Process & Methodology**

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

1. Language Choice
2. Platform Integration
3. API Design
4. Testing and QA

*Torpedo Technical Documentation*

(c) 2026 Torpedo Team. All rights reserved.