**Architecture Overview**

The Modular Media Streaming Suite is a refactored, extensible media player system built using structural design patterns to overcome the monolithic limitations of the legacy codebase. It supports multiple media sources, dynamic plugins, composite playlists, runtime renderer switching, and stream caching. Key components include:

**Media Sources:** Unified via the Adapter pattern with a MediaSource interface. Adapters like LocalFileSource, HLSSource, and RemoteAPISource handle local files, HLS streams, and remote APIs, allowing polymorphic data fetching without duplication.

**Feature Plugins:** Enabled by the Decorator pattern on the MediaProcessor interface. A BasicMediaProcessor handles core processing, while decorators (SubtitleDecorator, EqualizerDecorator, WatermarkDecorator) allow runtime stacking for features like subtitles, audio equalization, and watermarking.

**Playlists:** Managed through the Composite pattern with a MediaItem interface. FileItem represents individual media (leaves), and Playlist composes multiple items (including nested playlists), enabling uniform playback of single files or hierarchical structures.

**Rendering Strategies:** Decoupled using the Bridge pattern. The MediaPlayerAbstraction (e.g., StandardMediaPlayer) separates playback logic from implementations like HardwareRenderer and SoftwareRenderer, supporting runtime switches via setRenderer().

**Caching:** Implemented with the Proxy pattern in RemoteStreamProxy, which wraps remote sources for lazy caching, improving performance on repeated fetches without altering client code.

The system emphasizes modularity and the open-closed principle, facilitating additions like new sources or decorators. The entry point is ModularMediaPlayer, now featuring an interactive console menu for user-driven assembly of playlists, decorator selection, proxy usage, and renderer toggling, with playback orchestrated through MediaPlayerAbstraction. This design ensures scalability for evolving requirements in media streaming.