Team Torrent (Group 25)

CISC 322/326 Assignment 1: Presentation

Kodi: Conceptual Architecture Analysis

Aselstyne, Alex (alex.aselstyne@queensu.ca) Lead.

Dinari, Daniel (20dd29@queensu.ca) Pres.

Nagel, Jake (20jn29@queensu.ca)

Peterson, Jack (21jrp10@queensu.ca) Pres.

Pleava, Ryan (20rcp5@queensu.ca)

YouTube Video Link: https://www.youtube.com/watch?v=Z7A151OGE0U

Introduction to Kodi

- Free and open-source multimedia player.
- Originally developed for the Xbox (2001), initial release in 2003.
- Later ported to most popular platforms.
- Disassociated from Xbox in 2014.

Introduction to our Project

- Document Kodi's conceptual architecture abstractly

- 5 Primary Topics
  1) Kodi Subsystems from the Kodi Wiki
  2) Derivation Process
  3) Describing the Use Cases and Sequence Diagrams
  4) Describe the Architectural Styles
  5) Describe the Overall Conceptual Architecture
  6) Conclusions

Kodi Architecture from Kodi Wiki

- Divided into 4 primary layers
    1. The presentation layer, which handles all GUI and user interaction tasks.
    2. The "business" layer, which does the computation (rendering, local media hosting, plugin computation)
    3. A data layer, which handles the loading of media from local hard disks and networks.
    4. And a client layer, which provides a framework for communication with the host operating system.
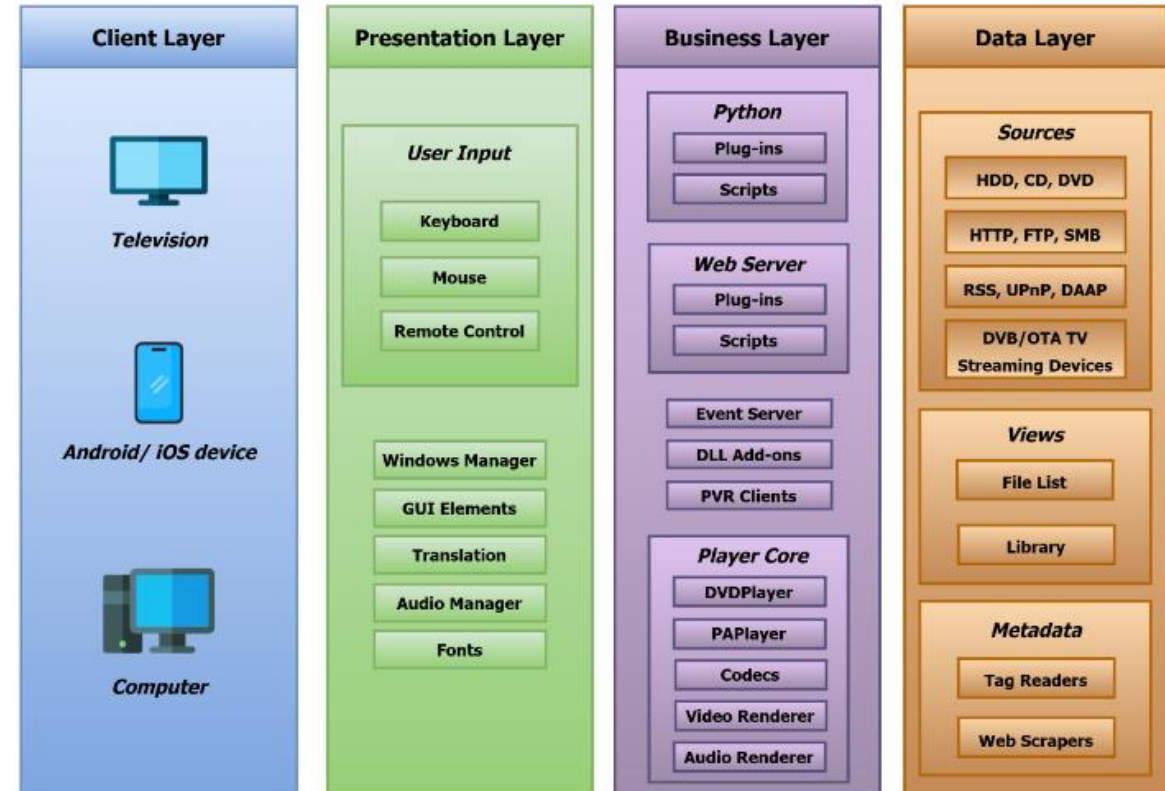
Derivation Process

- Came into agreement about overall architecture initially.

- Came up with separate 2 use cases, chosen due to their wide use of components, and large differences.

- Performed use case in the Kodi app, and matched ideas from the use case to the Kodi Wiki Architecture.

- Utilized 1 writer, and the rest of the group contributed, made sequence diagrams agreed upon, and consistent across ideas.

Use Case 1

- User selecting and playing a video
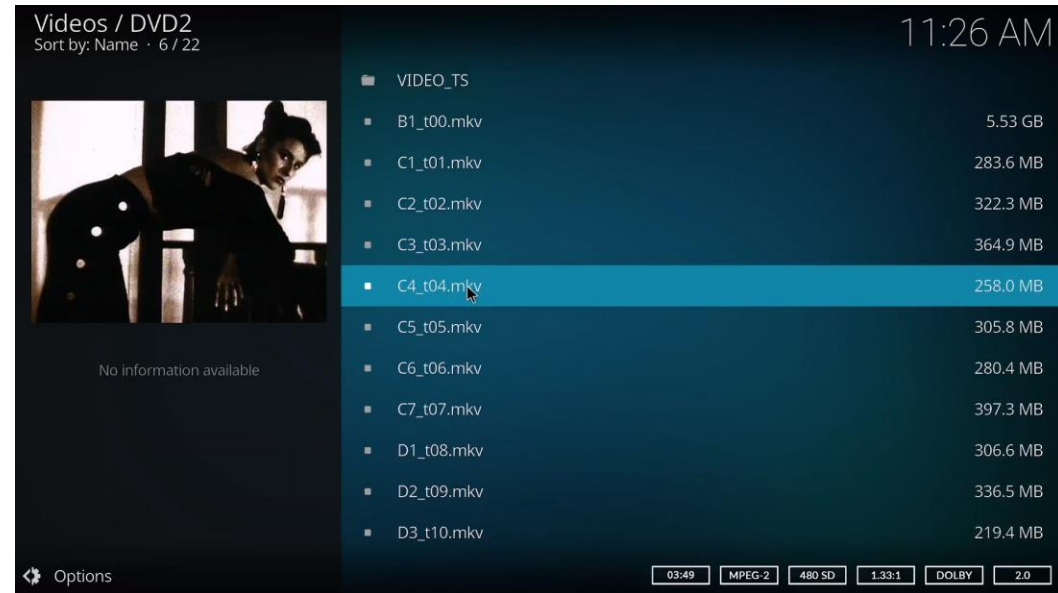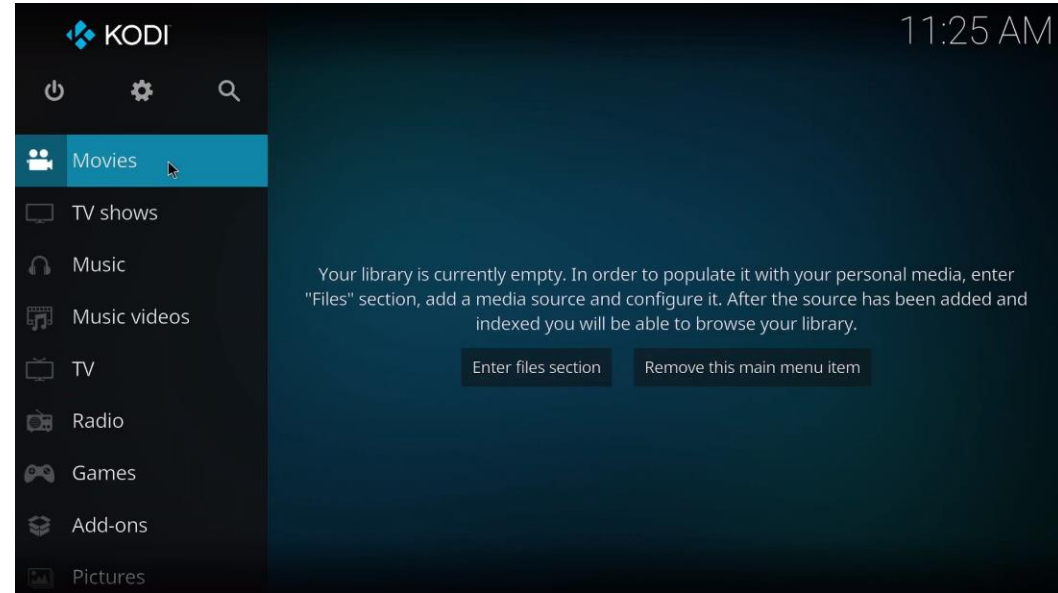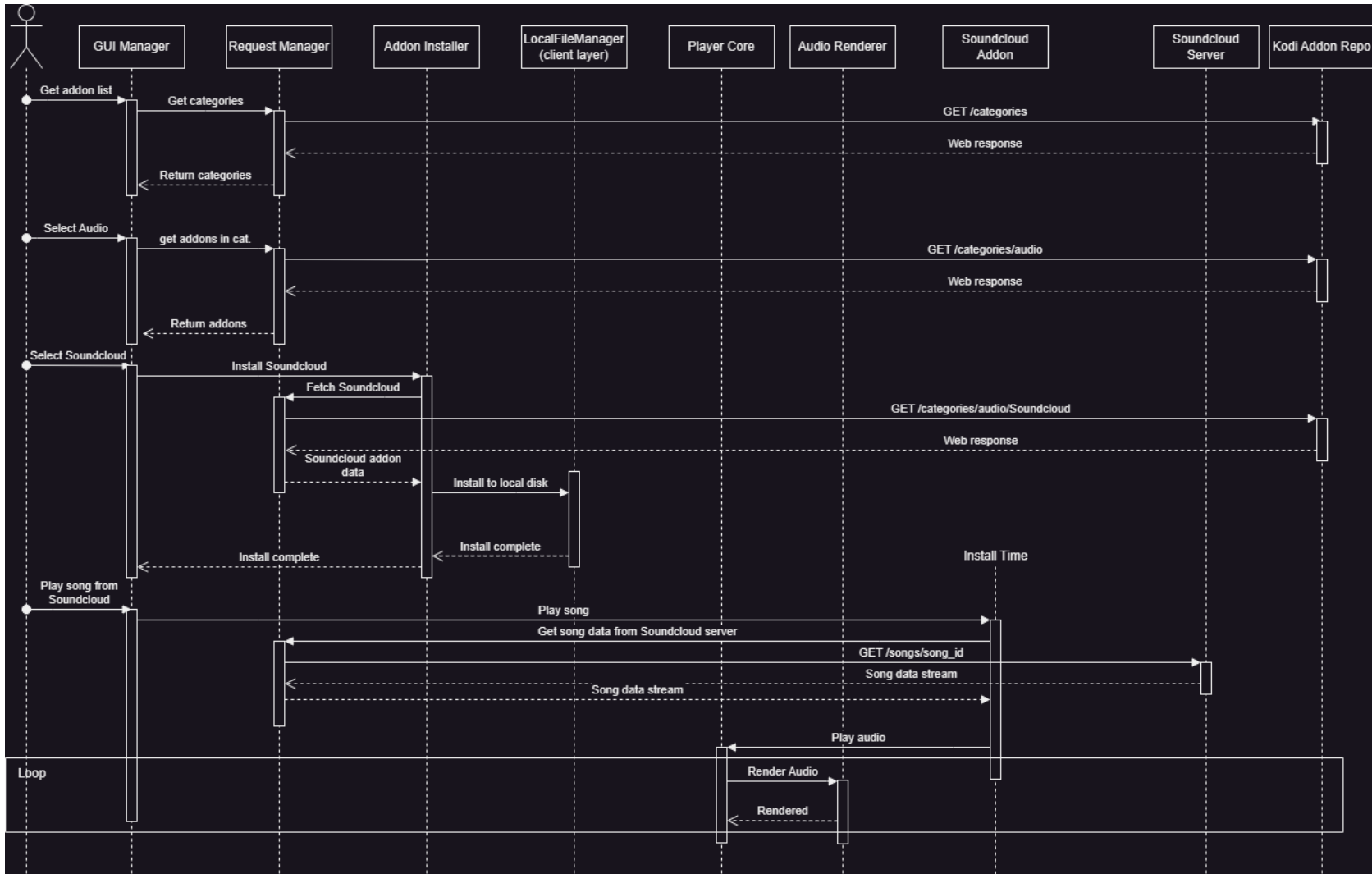
Use Case 1 Sequence Diagram
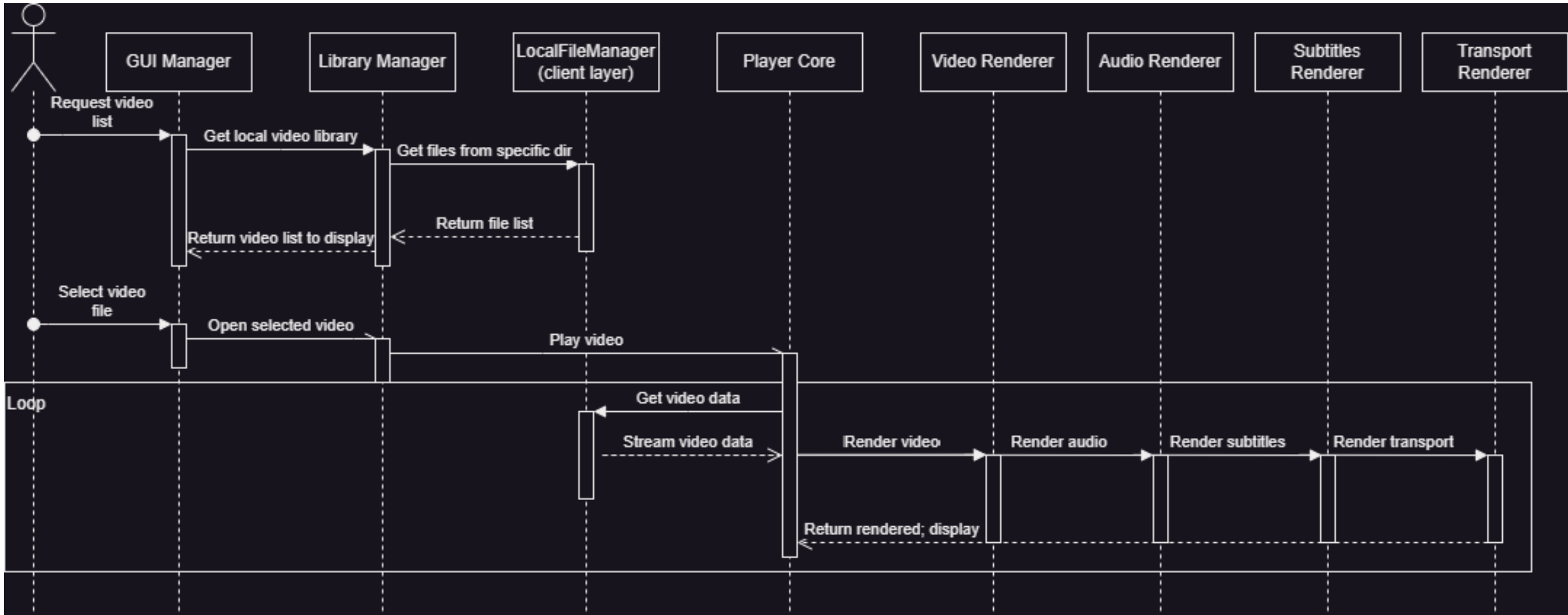
Use Case 2

- User selects an addon from the available list, installs it, and then uses it to play a song from a remote server

Sequence Diagrams for Use Case 2

Architectural Styles Used by Kodi
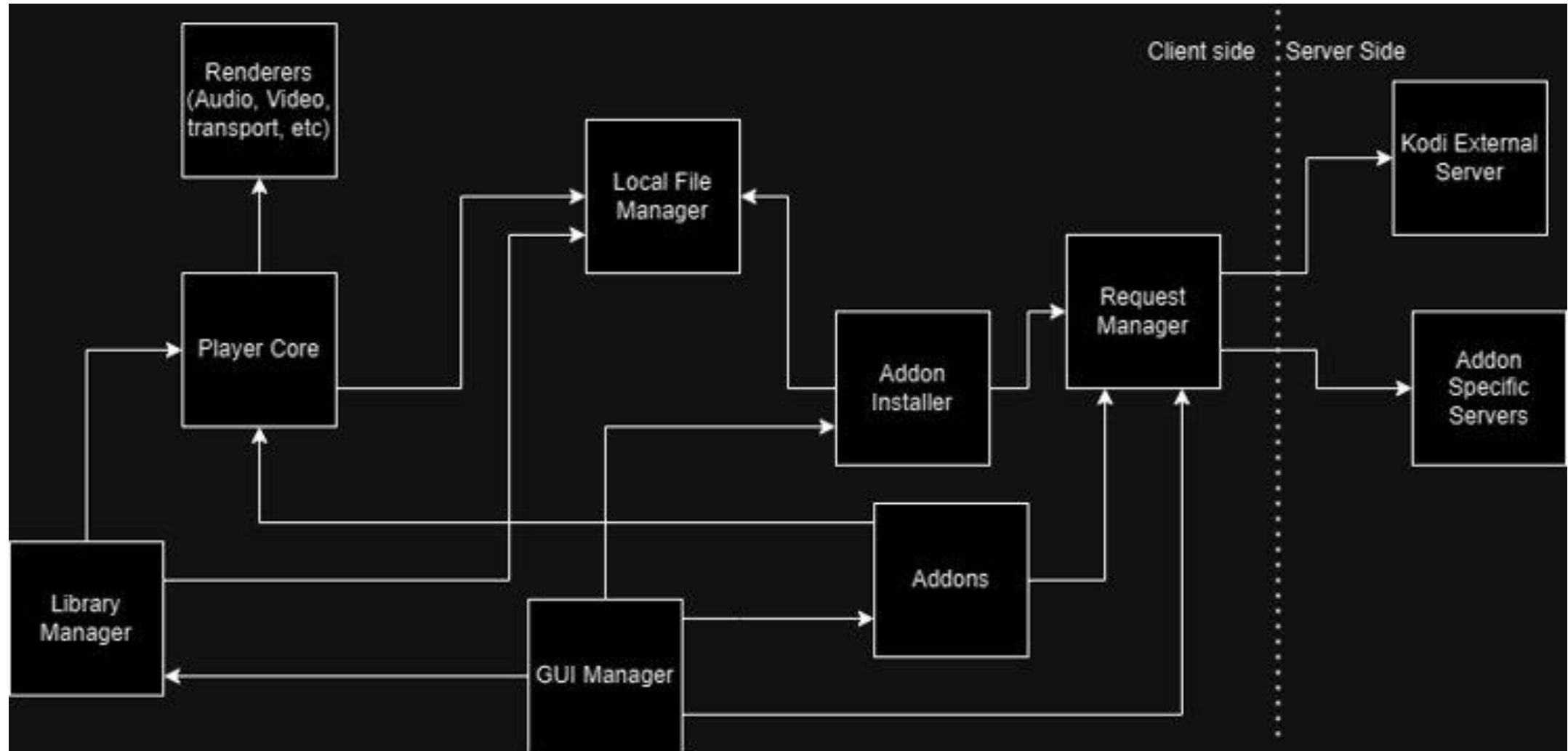
Pipe and Filter

Video/Audio Renderer

Repository Style
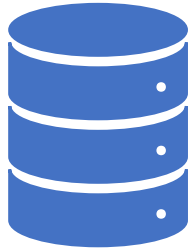
All media accessed locally or remotely

Client-Server

Kodi's servers host addons that can be installed by each Kodi instance

Conceptual Architecture and Component Diagram
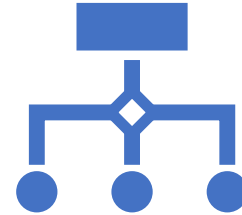
Data & Control Flow

Data Flow

Flows from supported file format into displayed media.

Passes from file to rendering pipelines, synced and verified by player core.

Then delivered to user.

Control Flow

Primarily managed by the GUI. GUI sends requests to other components in the system.

Other components then process the request and respond accordingly (e.g. play a video, display local files, etc.).

Lessons Learned and Conclusion

- Time Management
- Sequence diagram usage is important for creating a component diagram.
- Performing use case in the application makes understanding easier.
- Balancing personal contribution, and group contribution.
- Kodi is designed in an efficient way.
- Kodi is a very versatile and well-designed software.

References

[1] "About Kodi," Kodi.tv. [Online]. Available: https://kodi.tv/about/. [Accessed: 22-Oct-2023].

[2] Kodi.wiki. [Online]. Available: https://kodi.wiki/view/Architecture#Business_Layer.[Accessed: 22-Oct-2023].

[3] "Kodi," Github.io. [Online]. Available: http://delftswa.github.io/chapters/kodi/. [Accessed: 22-Oct-2023].

[4] "Kodi Foundation," Kodi.tv. [Online]. Available: https://kodi.tv/about/foundation/. [Accessed: 22-Oct-2023].

[5] Kodi.wiki. [Online]. Available: https://kodi.wiki/view/History_of_Kodi. [Accessed: 22-Oct-2023].

[6] "Pipe and filter," Berkeley.edu. [Online]. Available: https://patterns.eecs.berkeley.edu/?page_id=19. [Accessed: 22-Oct-2023].