



Team Torrent (Group 25)

CISC 322/326 Assignment 3: Presentation

Kodi: Enhancement Proposal

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 Link: <https://youtu.be/QmdF1bCxjPk>

Introduction to our Project

Primary Topics:

- 1) Enhancement Proposal
- 2) Functional and Non-Functional Requirements
- 3) Current State of the System
- 4) Implementation Plans
- 5) SAAM Analysis
- 6) Use Cases
- 7) Testing Plans
- 8) Introduced Risks
- 9) Conclusion



Kodi Enhancement Proposal: Subtitle System

- Automatic and user-shared subtitles.
- Can automatically generate subtitles for videos using a machine learning subsystem.
- Users can upload saved subtitles to a remote server.
- Subtitles will be selected from the same menu used for local subtitles.

Functional Requirements

UI: A friendly user interface for easy activation and interaction.

Subtitle Generator: A ML model trained to generate subtitles from audio.

System Storage: Capacity to store, edit, download and reuse generated subtitles.

System Integration: Integration with Kodi's current architecture, allowing users to activate subtitle generation easily.

Non-Functional Requirements

Performance: Fast subtitle generation with low latency

Scalability: Handle an increasing number of users with subtitle generation requests

Reliability: Accuracy in subtitle generation across genres and audio qualities

Security: Measures to protect user data, especially if any personalization is involved

Maintainability: Easy updates and improvements without disrupting overall functionality

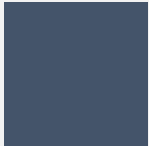
Current State of the System



Currently, Kodi subtitles can be hard-coded in a file, or have a reference subtitle files such as an SRT file.

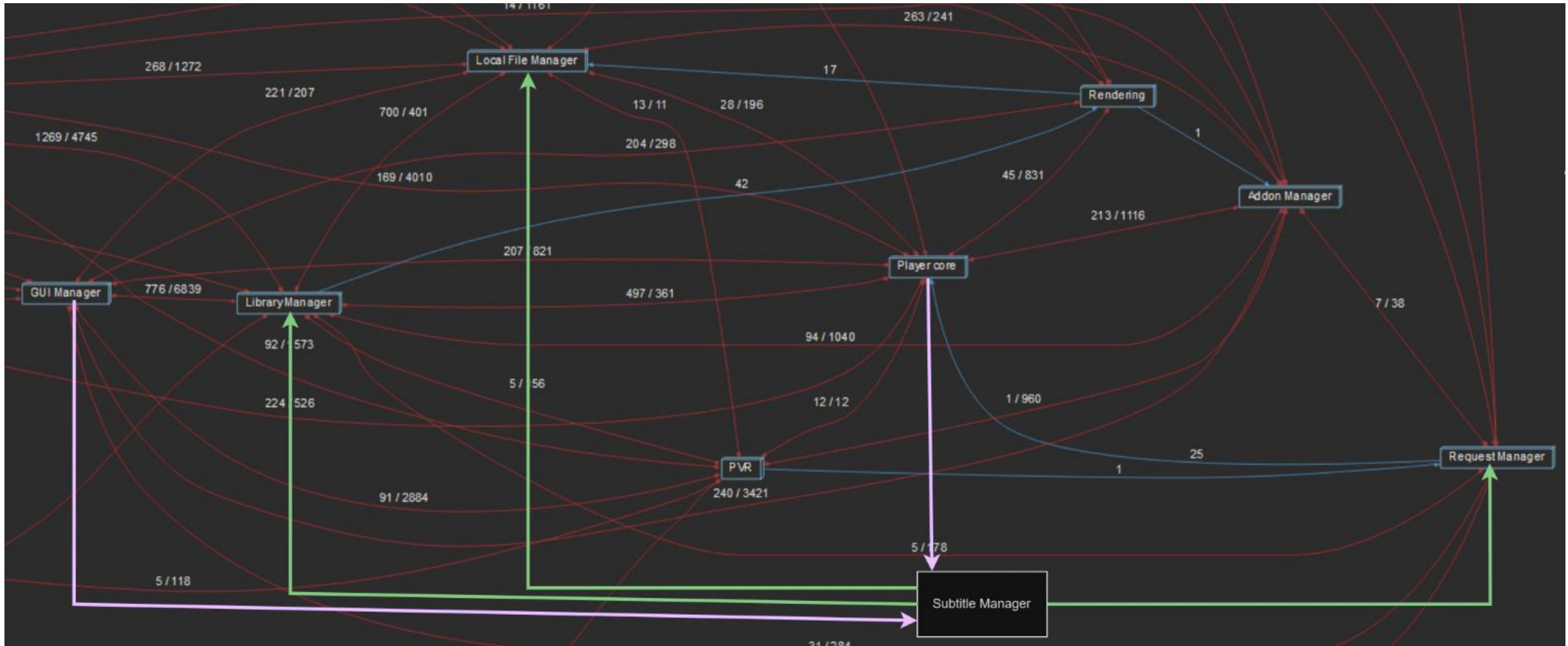


Kodi has a plug in called "Open Subtitles" that has access to a database of 7 million subtitles for popular TV and movies.



Both of these options have problems, finding media with built-in subtitles is challenging, and subtitles for an unpopular or local video most likely do not exist.

Implementation Plans



SAAM Analysis

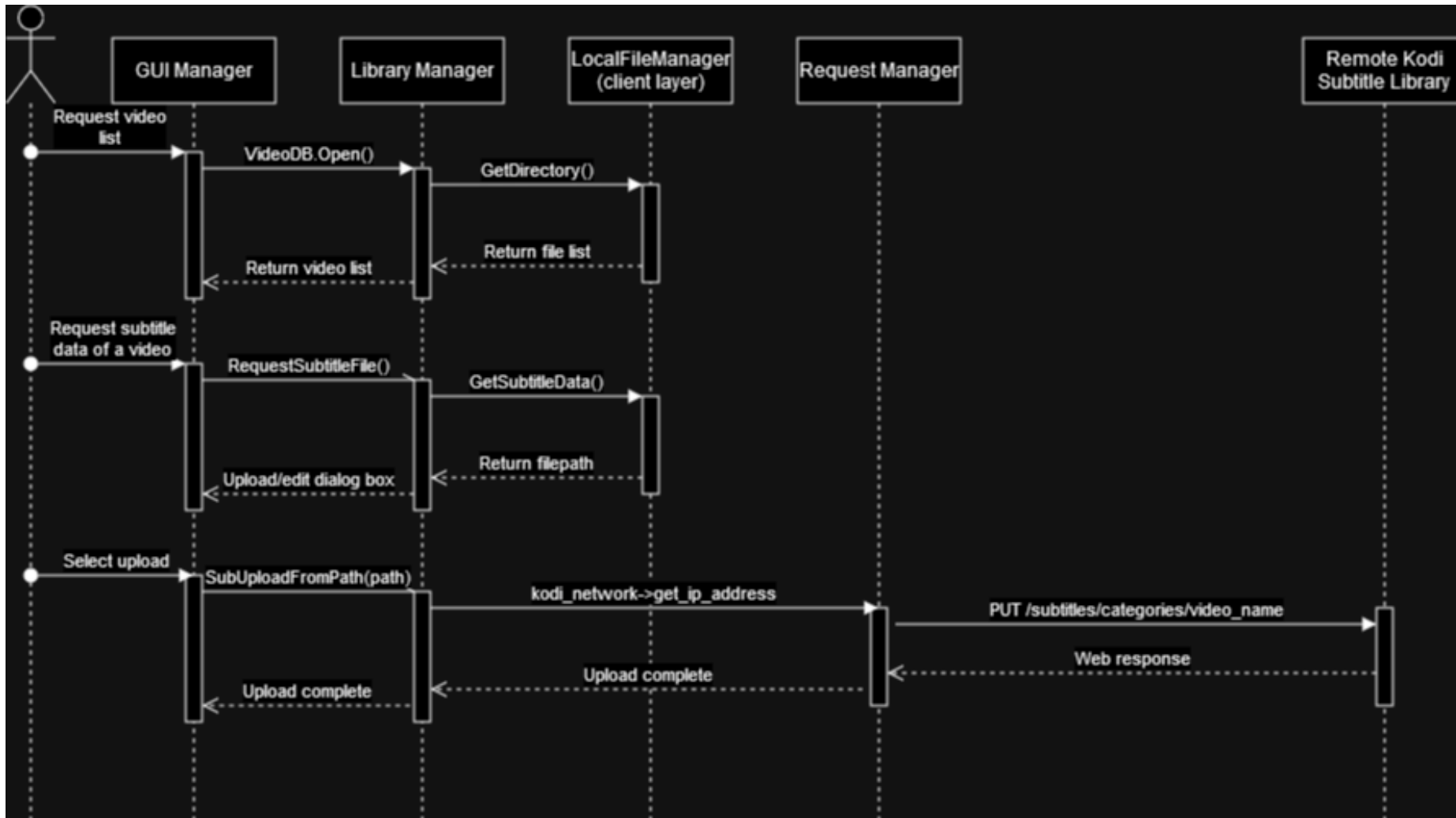
Stakeholders	Primary NFRs
Users	<p>Performance – Users want subtitles to be generated as quickly as possible.</p> <p>Cost – The cost to the user should be as low as possible, if not zero, since Kodi is currently free software.</p> <p>Accuracy – The subtitles generated need to be as accurate as possible to be useful to the user.</p>
Kodi Developers	<p>Scalability – Currently, Kodi is highly scalable because it is all locally computed on the user’s machine. Developers would prefer to keep it that way to keep complexity down.</p> <p>Maintainability – Developers want the software to be as easy to maintain as possible.</p> <p>Testability – Ease of testing is desirable for developers so they produce bug-free code.</p>
Kodi Foundation	<p>Cost – The Kodi Foundation is funded by donations from users, and as such wants to keep their costs very low.</p> <p>Maintainability – Keeping the software maintainable will help attract developers from the open-source community.</p> <p>Security – Security is a high priority for the Kodi Foundation, both to maintain their image and integrity as well as protect their assets.</p>

NFR	Implementation 1 (Local)	Implementation 2 (Cloud)
Performance	Highly dependent on the characteristics of the end user’s machine. Faster computers will produce subtitles much faster.	Quick results can consistently be computed regardless of the user’s machine. Internet speed is the only factor, as audio needs to be uploaded.
Cost	Free for both parties, aside from development time (which does not differ between the two implementations).	One party will have to pay for server time to generate the subtitles. Either the Kodi Foundation will absorb it, or it will be passed down to the end user as a “premium” option.
Accuracy	Different models may need to be selected for slower machines, which produce less accurate results.	Likely very high, assuming high performance cloud architecture is used.
Scalability	High. There is no change compared to current Kodi versions.	Limited. Additional server resources will have to be purchased if more users begin using the service.
Maintainability	Somewhat diminished, as the library used for subtitle generation will have to be periodically updated.	Greatly diminished. Not only will the library need to be updated, but server infrastructure must be maintained and monitored.
Security	Unchanged. No additional data is sent from the user to Kodi.	Potential risks. Audio files sent to the subtitle generation server may contain malicious code.

Use Case 1

- Sequence diagram of use case 1 – a user uploads subtitles to a central server for user sharing.

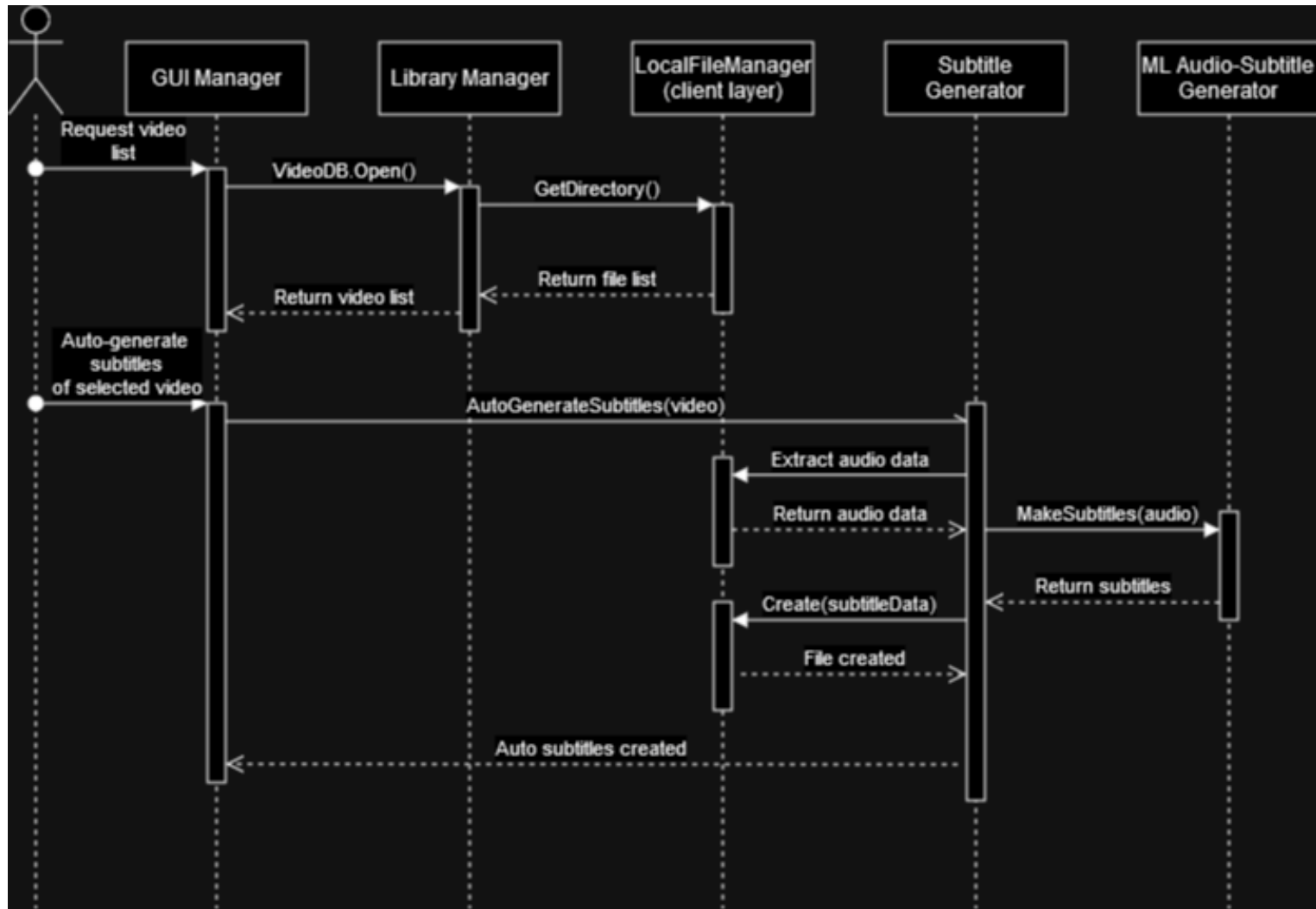
Use Case 1 Sequence Diagram



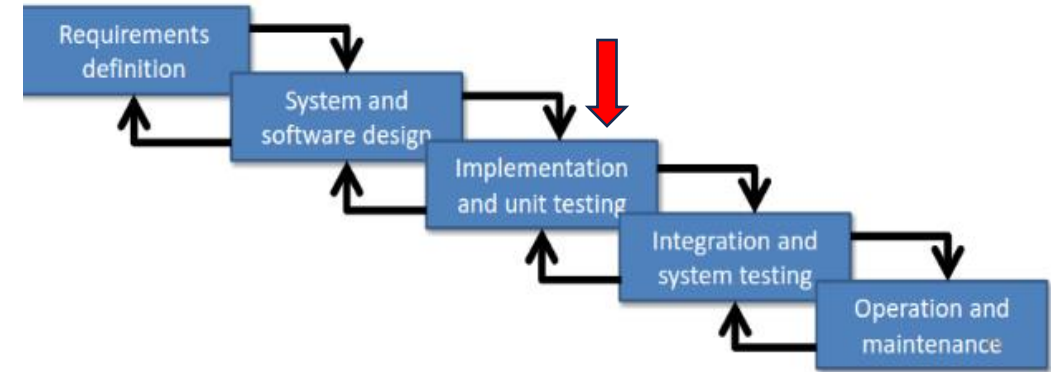
Use Case 2

- a user automatically generates subtitles for a video using the built-in ML subtitle generator.

Sequence Diagrams for Use Case 2



Testing Plans



Functional Testing:

- **Black Box Test (Output Coverage):** Compare generated subs vs transcript -> If accuracy +99% then pass
- **Black Box Test (Input Coverage):** SQL Injection can't be performed on search bars
- **White Box Test (Decision Mutation):** Decision generation when no language preference is given -> must show English subs
- **Integration Test:** Compare generated subs vs displayed

Non-Functional Testing:

- **performance, reliability, usability, and security**

Plan:

1. Add new unit tests for subtitle gen component
-> Evaluate functionality
2. Run regression tests
-> Validate system integration
3. Tests are automated + report generation

Potential Risks



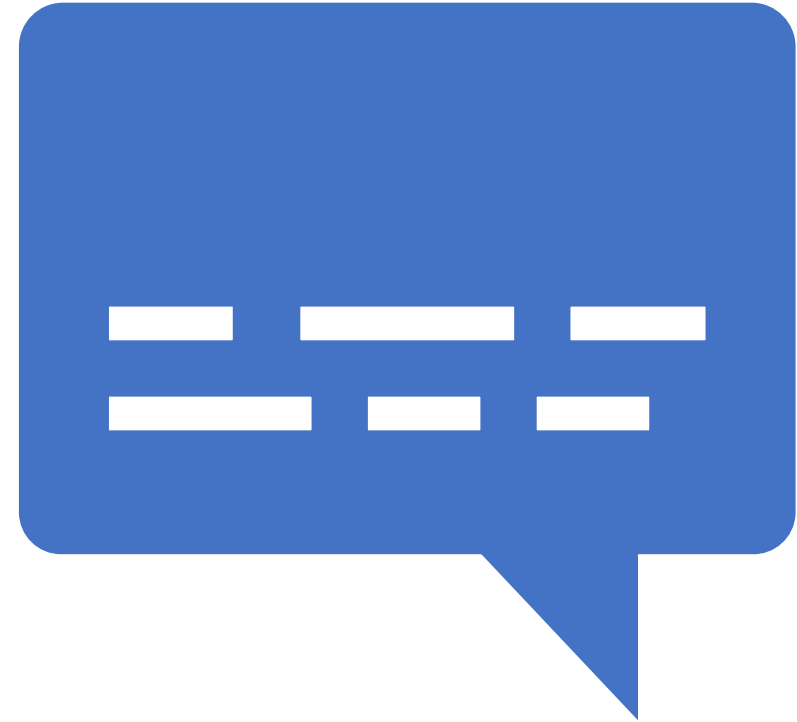
Security Flaw: Injection Attacks in User-Uploaded Subtitle Files



Performance Risk: Resource hogging during ML subtitle generation

Lessons Learned and Conclusion

- **ML Subtitles for Kodi:** Proposed ML-driven subtitle generation for Kodi.
- **Cloud Sharing:** Users upload/download subtitles on a central server.
- **Architecture Choice:** Favored pipe & filter for practical system integration.
- **Dependency Map:** Mapped module dependencies for insights into system integration.
- **Efficiency Priority:** SAAM analysis supports the first implementation for seamless integration.





References

Original References

- [1] *English subtitles*. Subtitles. (n.d.). <https://www.opensubtitles.org/en/en%20>
- [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].

New A3 References

- [1] OpenAI, Introducing Whisper, <https://openai.com/research/whisper/> (accessed Dec. 5, 2023).
- [2] OpenAI, “Openai/whisper: Robust speech recognition via large-scale weak supervision,” GitHub - Whisper, <https://github.com/openai/whisper> (accessed Dec. 5, 2023).
- [3] Abdeladim-S, “Abdeladim-S/subsai:  subtitles generation tool (web-UI + CLI + python package) powered by openai’s whisper and its variants .