

YouTube Link

...and yes, with the right add-on, you
can watch in in Kodi.

Copy this link:

<https://youtu.be/jgF3fQ20Z7M>

or just click [here!](#)



Upon Further Review: Kodi and Letterboxd

Group 2 - ArchAngels

Presentation Designer, Narrator, Report Writer- Bianca Bucchino: 20blb@queensu.ca

Architectural Analyst, Report Writer - Adam Clarke: 21asc6@queensu.ca

Presentation Scripter, Report Editor/Writer - Christian Higham: 20csdh@queensu.ca

Architectural Analyst, Report Writer - Omar Ibrahim: 20omha@queensu.ca

Architectural Analyst, Report Writer - Owen Rocchi: 19omr6@queensu.ca

Group Leader, Architectural Analyst, Report Writer, Diagram Designer - Aidan Sibley: 20ajs18@queensu.ca

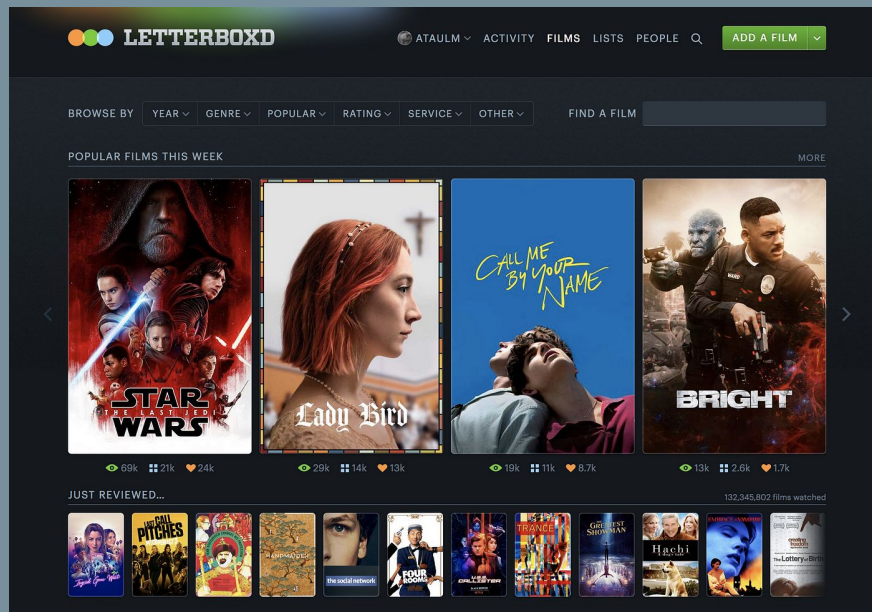
Presentation Overview

- Letterboxd Overview
- Proposed Enhancement
- Stakeholder Analysis
 - ◆ Identification of 3 Stakeholders & their NFRs
- Analysis of Current Architecture
- Proposed Changes for Enhancement
- Discussion of impact and risk of changes
- Evaluation of Architectural Options
 - ◆ Modular
 - ◆ Hard Integration
- Use Case & Sequence Diagram
- Conclusion
- Lessons Learned

Letterboxd Overview

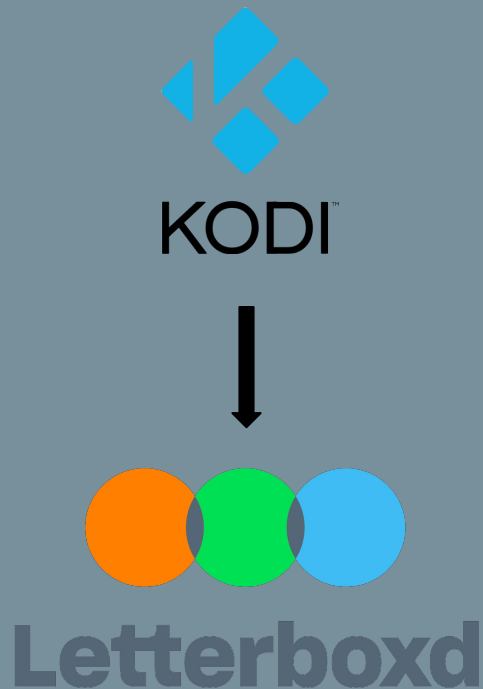
- For film enthusiasts
- Users can:
 - Rate & review movies
 - Organize movies
 - Get recommendations

Letterboxd GUI



<https://medium.com/@ataulm/this-is-letterboxd-57db4eeb6175>

Proposed Enhancement



- Give users features to Letterboxd within Kodi
- **Movie recommendations:**
 - Accessed via streaming or local files within Kodi
 - Watch history, relevant data sets, and addons passed to Letterboxd API
 - Effortless manner
- Who is affected...?

Stakeholder 1: End User

Goal:

Provide more utility in a Kodi user's search for something to watch.

Requirements:

- 75% of visible data must be loaded within 2 seconds
- Recommendation system must be accurate at least half of the time

Stakeholder 2: Open Source Developers

Goal:

New modifications to code must be in line with current tests and introduce new tests.

Requirements:

- New functions do not break old ones
- 95% of the code base must be present to give developers insight into how our additions work

Stakeholder 3: Letterboxd

Goal:

Ensure enhancement is appropriately used on both the front and back ends.

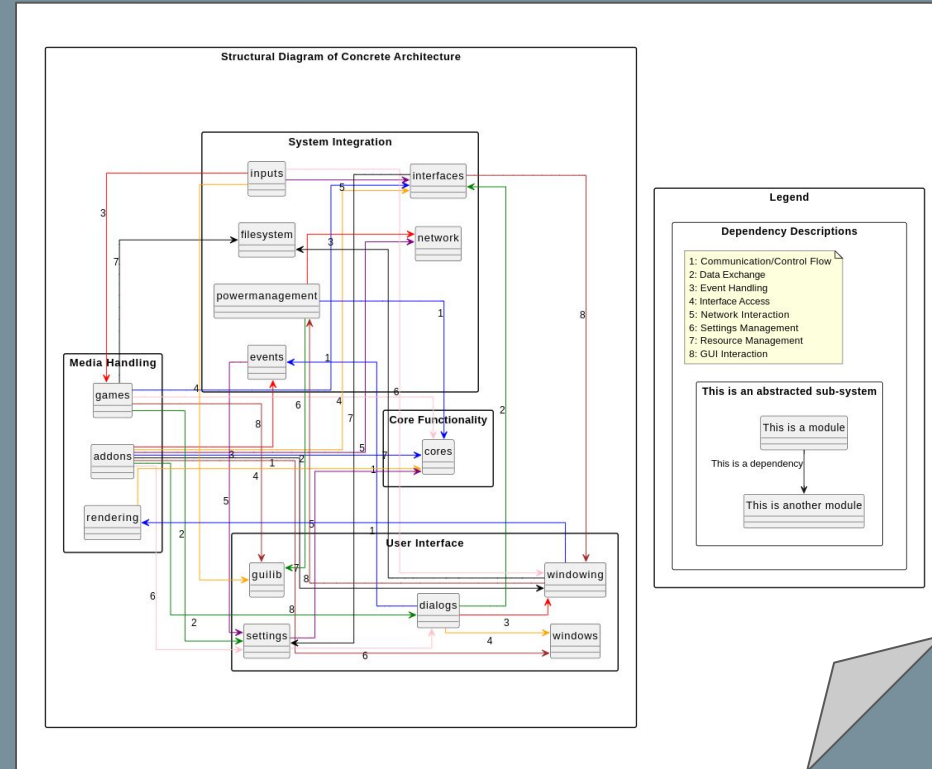
Requirements:

- Regular security analysis of API must occur
- Any new high-risk issues should be immediately addressed
- Ensure that the API's server load remains below 80% of capacity

Kodi Architecture Review

Current Kodi Architecture

- Repository Style
- Four Main Subsystems:
 1. User Interface
 2. Media Handling
 3. System Integration
 4. Centralized Core Functionality
- Software is efficient and adaptable for add-ons



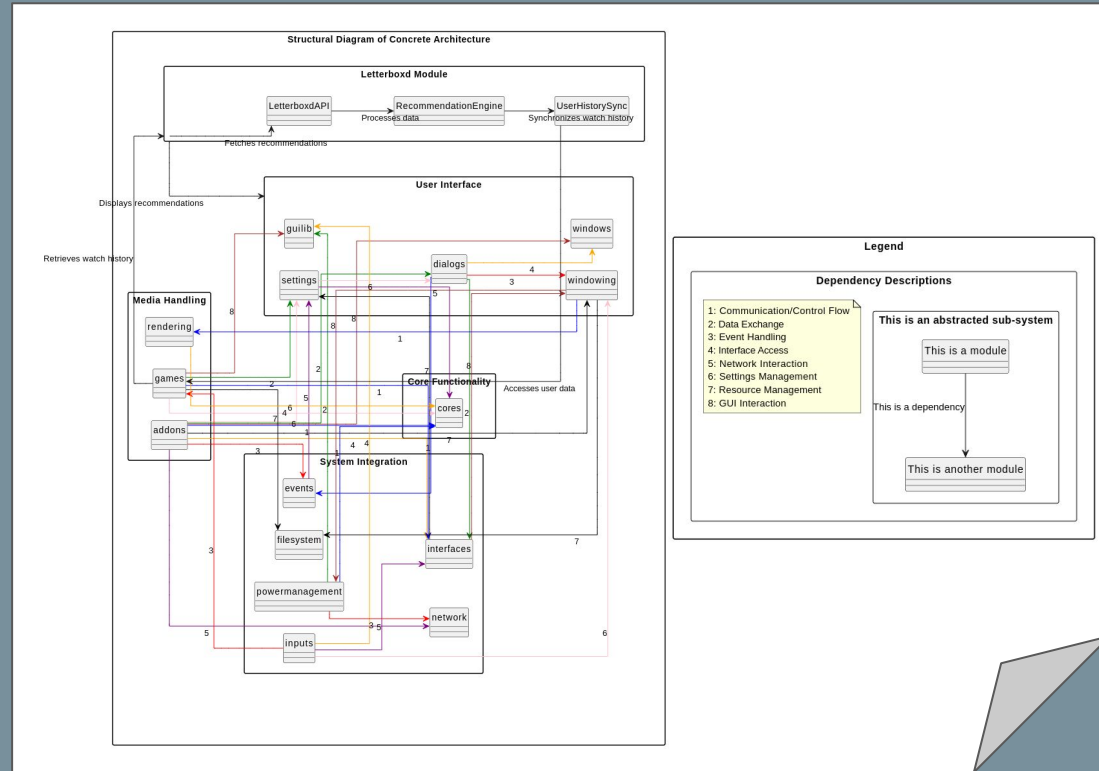
Proposed Changes for Enhancement

Let's utilize Kodi's adaptability!

Proposal:

- An entire new module for Letterboxd integration
- Media Handling Subsystem is given enhanced information on the files it streams & passes data to Letterboxd module
- Letterboxd module processes & feeds data to Letterboxd API
- Letterboxd module passes data to User Interface component

Updated Kodi Architecture to Support Enhancement



Impact and Potential Risks of Changes



- Decline in Performance
- Dislike of Recommendations
- Complexities of Integration and Documentation
- Letterboxd API
 - Availability
 - Reliability
- All result in an unsatisfactory user experience

Option 1: Modular Approach



The Letterboxd module becomes its own entity within Kodi that connects to the existing architecture. It receives data from Media Handling, sends it through to the Letterboxd API and displays recommended films with ratings and reviews via the User Interface Module.

Positives:

- Requires little modification

Negatives:

- May introduce performance, security, and availability issues for the API.

Option 2: Hard Integration

The recommendation system is folded into the media handling component and the recommendations are sent straight to its associated subcomponent within the User interface module.

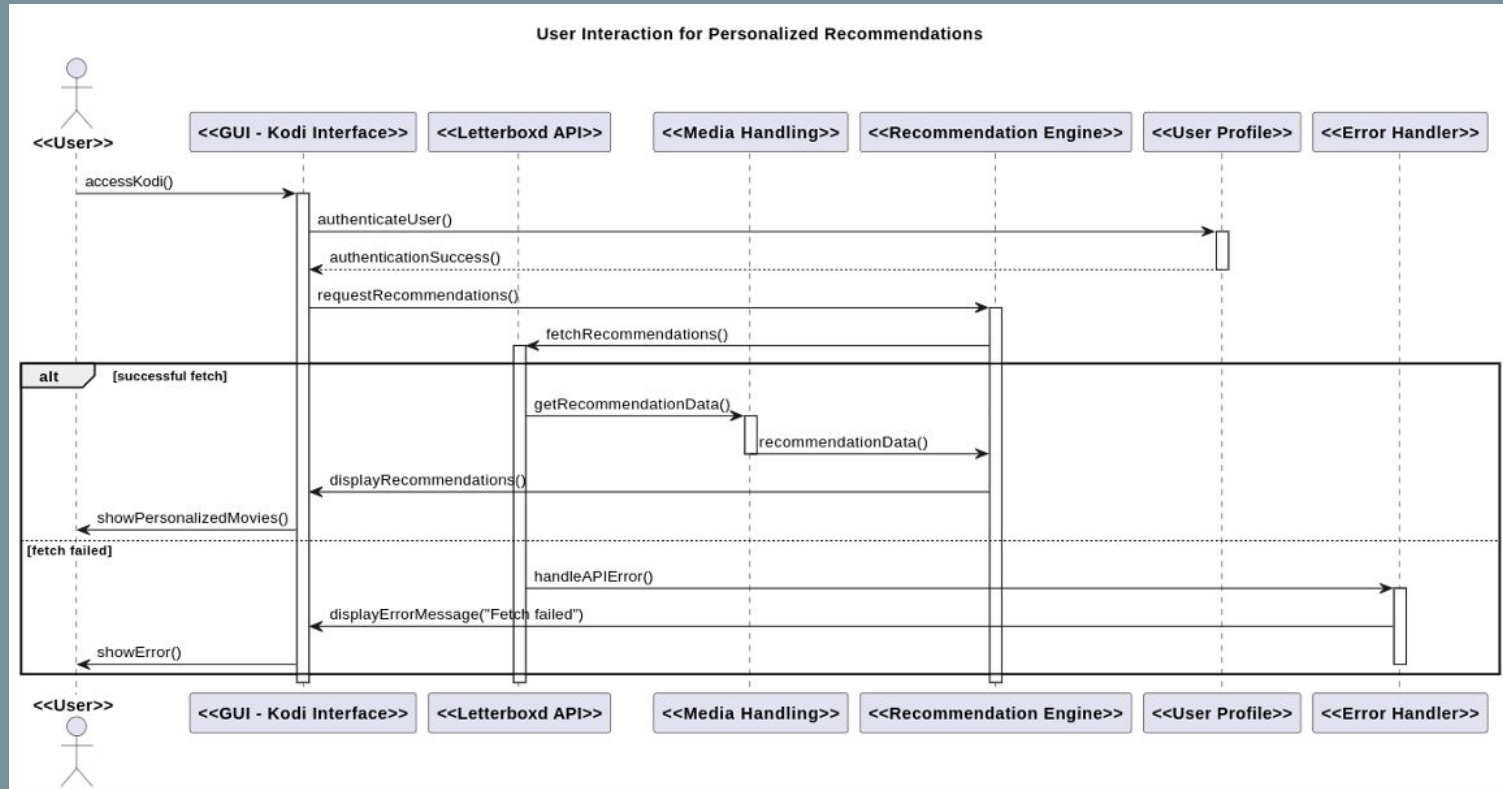
Positives:

- More efficient processes

Negatives:

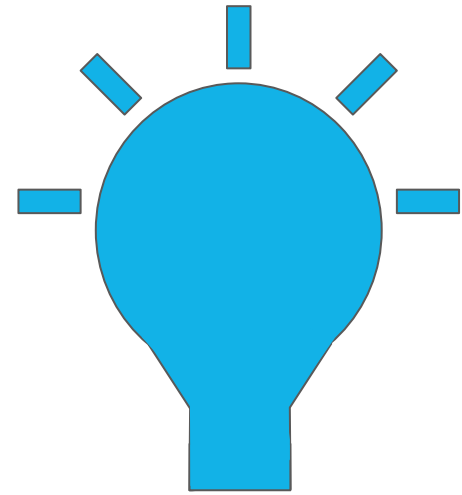
- Any bug introduced will more broadly affect Kodi
- Software becomes more complex

Use Case & Sequence Diagram



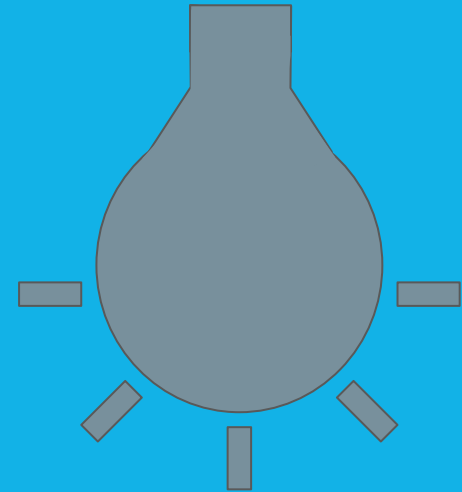
Lessons Learned

- Various benefits and drawbacks of integrating new, unplanned features into existing software
- Gained insight into the values and challenges of working around concrete architecture



- Have not tried to program our enhancement into Kodu
- Are not 100% sure our method of integration is possible
- All ideas are theoretical

Limitations



To Summarize

- Two ways to integrate a movie recommendation system using Letterboxd API:
 1. Modular Approach
 2. Hard Integration
- Considered stakeholders and their requirements
- Potential Risks and Impacts of Enhancement with our method of integration
- Discussed a use case for our enhancement

