CISC 322/326 Assignment 3
Architectural Enhancement

Dec 2, 2024

Group name: '; DROP TABLE groups; –

Xavier Awadalla - 21xpa@queensu.ca

Christopher Gil - 21cagi@queensu.ca

Aaron Rivest - 21alr18@queensu.ca

Craig Tylman - 21crt6@queensu.ca

Sam Tylman - 21sdt6@queensu.ca

Felix Xing - 21fx2@queensu.ca

1.0 Document Goals

The goal of this Document is to Propose a new feature/enhancement for ScummVM, describe in detail what it is, how it would function and how it would be integrated into ScummVM's conceptual architecture. Next, we will perform a SAAM analysis to discuss who the stakeholders are and which non functional requirements for the enhancement would be important for them. An alternative implementation will also be considered for our enhancement.

1.1 Introduction and Overview

Enhancement description

The enhancement that we have chosen is an in-app hub for users to browse and download free-to-download games and modifications directly through the ScummVM GUI. Currently, this must be done by searching various websites and forums for ScummVM-supported modifications, so a native hub would skip this step and provide a convenient location for all downloadable content. The hub would be hosted on a server-side content delivery network (CDN) that the client makes requests to in order to render its content in a convenient user interface.

Enhancement consequences:

With the addition of an in-app game hub, there will be a handful of consequences for the overall system:

- System maintainability is the major drawback because the in-app hub will require higher server costs and some manual labour for maintenance.
- On the other hand, system evolvability will generally benefit from this enhancement as the in-app hub will serve as a great way to monitor system evolution.
- Testability will also be greatly improved as users can give feedback and report bugs directly through the hub via reviews and forums.
- Finally, overall system performance would not be significantly affected by this enhancement.

Alternate implementation:

Since one rigid method of implementation is not always a good idea when proposing an enhancement to a new piece of software, we will also provide an alternative way to implement the proposed enhancement.

We will also discuss its consequences in much the same way, yet still briefer than how we discussed the original implementation.

SAAM analysis:

The proposed implementations for the enhancement involves 3 distinct stakeholders: End users, ScummVM developers and game developers/modders. For end users, ease of use and performance are important. The enhancement should provide users with a fast, intuitive and seamless way to install independent games and game mods without external searches. Developers should prioritize maintainability, making sure that the enhancement is well-documented and modular. Developers should also consider compatibility to keep ScummVM's diverse community growing. Finally, independent developers especially value scalability and security, requiring a platform safe from cyber threats and capable of handling increased traffic.

We will also briefly make some concluding remarks at the end, as usual.

2.0 Enhancement Description

Although ScummVM functions well after importing pre-acquired games, the software is ultimately useless without first obtaining a supported game. While many of these games must be purchased digitally, the ScummVM website provides download links to several freeware titles that are directly set up for use with the software.

Our proposed enhancement for ScummVM is the addition of an in-app game hub that allows users to browse and install new adventure games without having to search on external websites. Games would be listed just as they are on ScummVM's website, and selected ones would be immediately downloaded and imported into ScummVM.

Alongside freeware games, many talented ScummVM users have also created their own installable content, ranging from modifications ("mods") and translations of existing games to original fan-made adventures. This content would be uploadable to our in-app hub, eliminating the need to search through obscure websites or forum posts in search of download links.

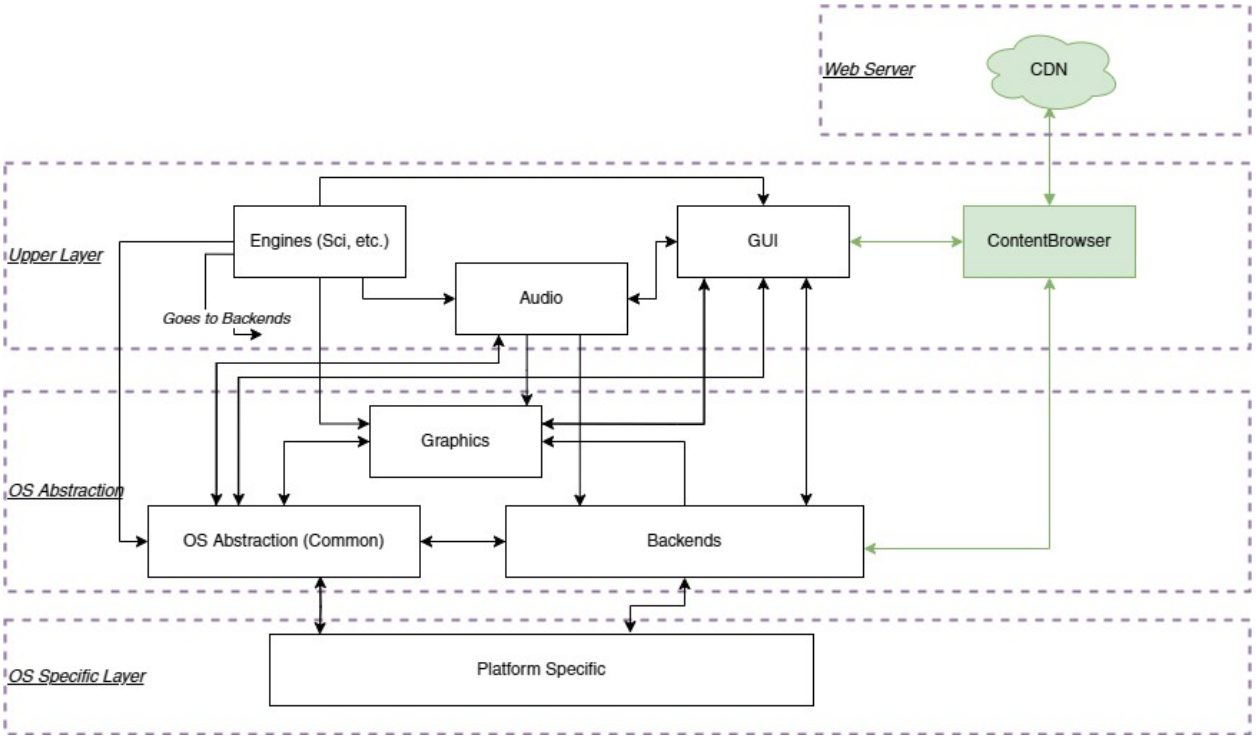
All downloadable content would be hosted on a dedicated content delivery network (CDN), with any new games and mods requiring staff approval before being listed on the in-app hub.

2.1 Enhancement Integration

Integrating an in-app hub into ScummVM would mostly require work on the software's frontend. Once content is fetched from the server, it must be displayed in an intuitive interface that shows all relevant information and loads any assets needed (e.g. thumbnails and other preview images). Most of this work could be done directly from the GUI component, but it would be especially useful to extend it into a dedicated ContentBrowser component that fetches server-side information and manages installed content.

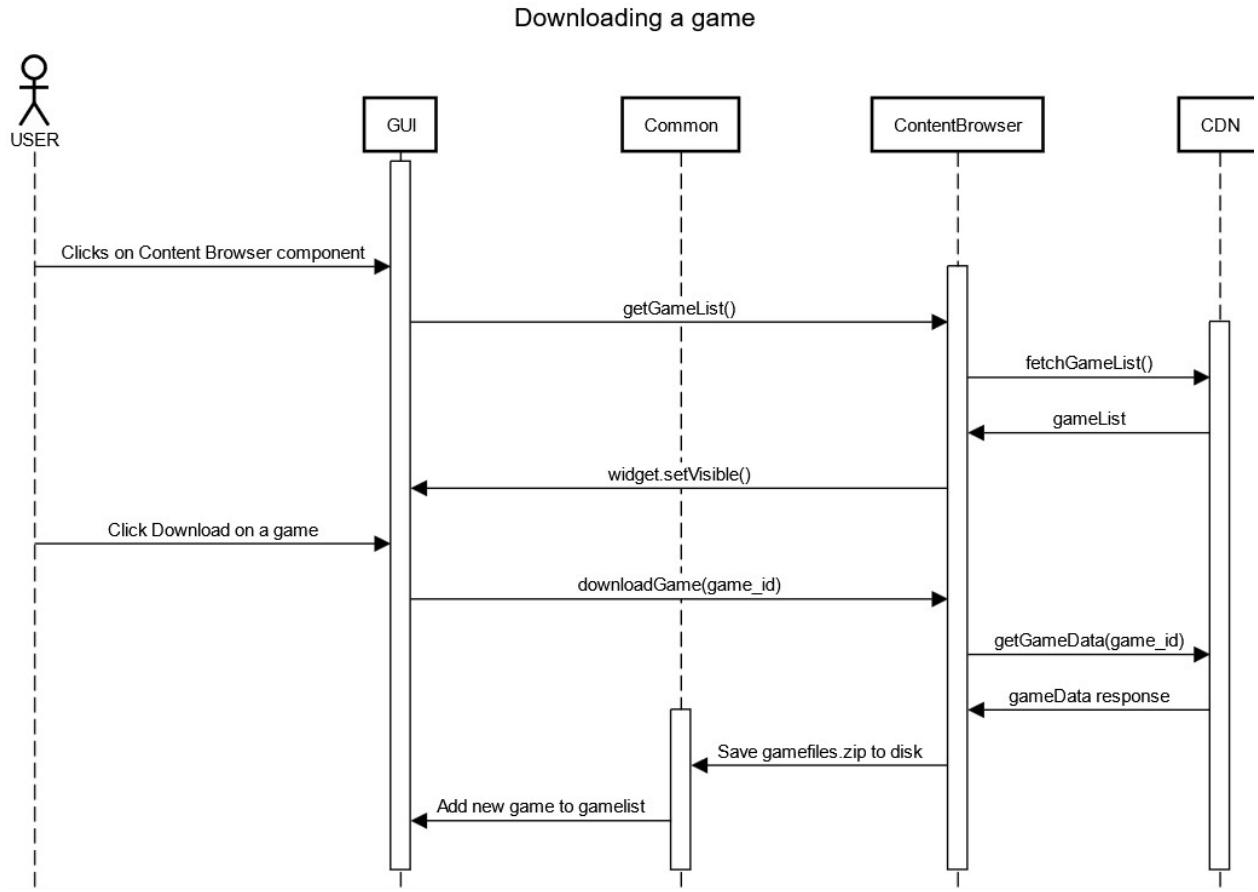
The ContentBrowser component would depend on the GUI component, which would display the downloadable content in a new section of the ScummVM launcher. It would also depend on the Backends component in order to install the content onto the user's device and set it up as if it were imported manually.

With our enhancements, the new architecture for the software would look like this:

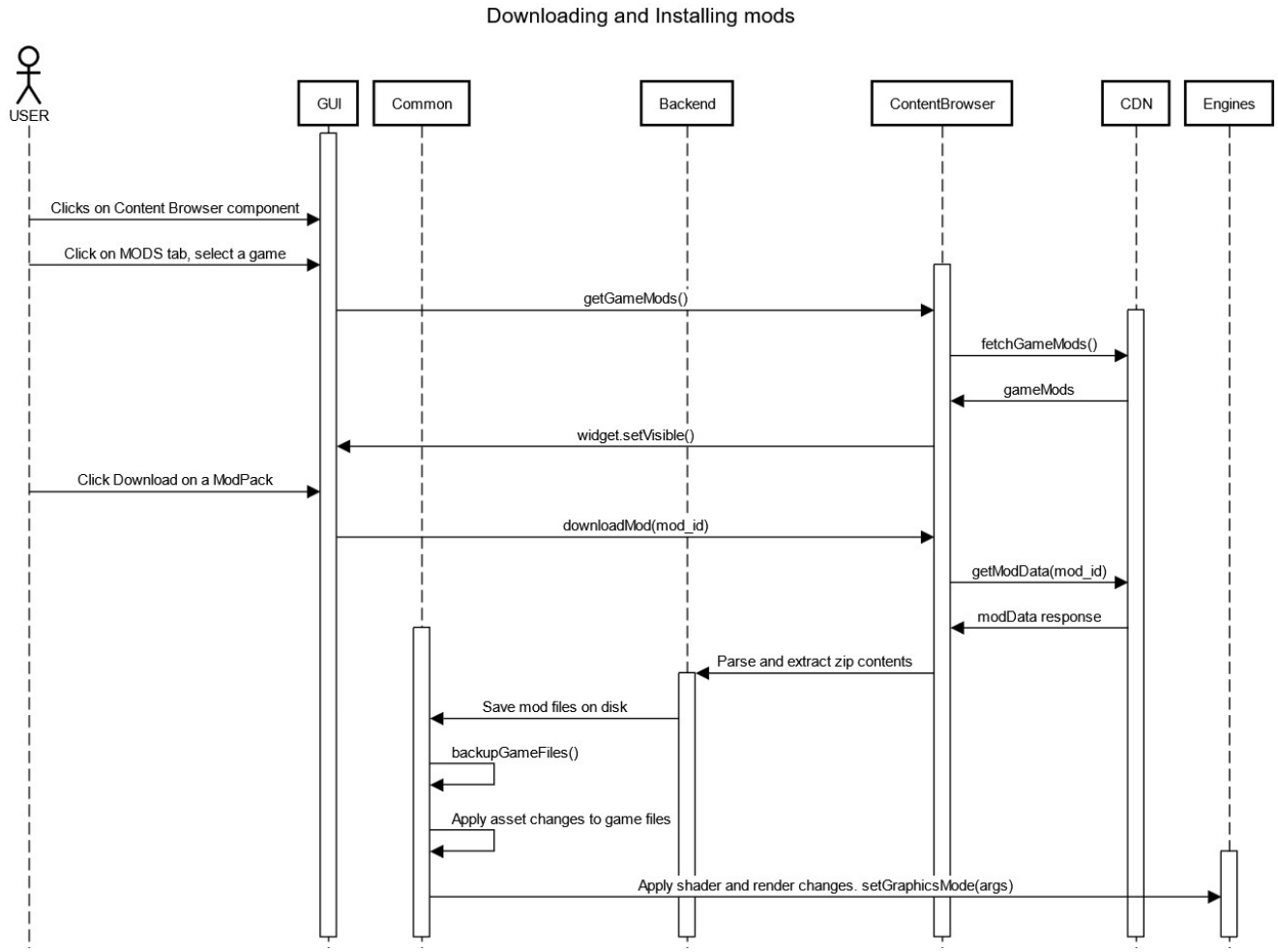


The CDN is not directly part of the ScummVM software, but rather hosted on an external website that the ContentBrowser component makes requests to. This includes retrieving content listings, navigating between pages, and downloading their data.

2.2 Sequence Diagrams



The above diagram illustrates the use case “Downloading a game”. It begins by having the user click on the new enhancement, the Content Browser component on the GUI. This makes the component fetch the list of available games from the CDN, and then displays it on the GUI. The user can then download a game from the list, which causes ContentBrowser to send another request to the CDN, which responds with the game data, packed in a ZIP file. ContentBrowser sends the downloaded game to Common in order to save it to local storage, and it then adds it to the list of installed games and becomes visible in the GUI.



This sequence diagram illustrates the use case “Downloading and installing mods” with the new Content Browser (CB) enhancement. It begins with the user clicking on the MODS tab of the CB component, selecting a game from the list of installed games, which will cause CB to fetch a list of community-made mods for that game from the CDN, and display it to the GUI by using widgets. The user can then download a modification, which will make CB request the mod data from the CDN, download it packed in a zip file, send it to Backend to extract it, parse the files, and send them to Common, which will save them in the appropriate directories (e.g. make separate folders for shaders, animations, image assets). Common will create a backup of the original game files as a safety feature, and it will then apply the changes from the mod by replacing any assets (images, animations, sounds) as well as other low level settings, like setting a custom graphics mode to the Engine. It may also replace certain function calls or game behaviours during runtime.

2.3 Enhancement Consequences

Maintainability:

One major upside to our idea for an in-app game hub, is that it strongly improves the ability to moderate mods. Requiring staff approval for new mods ensures that only

compatible and safe material is added. This maintains system integrity and prevents issues caused by poorly implemented code. Developers can still choose to host their mods on websites unassociated with the game hub, however, if they choose to have their mods officially approved, they will reach a larger audience of players that are more inclined to use them.

On the downside, this enhancement will lead to increased server costs due to the fact that users will now be downloading the game files directly from ScummVM's servers. Additionally, staff would need to spend time staying on top of updates and new mod releases.

Evolvability:

Overall, having an in-app game hub would greatly benefit system evolvability. In our A1 deliverable, we already mentioned that a possible evolution direction for ScummVM would be to start working towards properly archiving the many historic games that ScummVM supports. This enhancement by itself almost entirely achieves this goal by archiving and hosting downloads on ScummVM's servers where there is no risk of the sites going down.

Another benefit is that having all of the supported games and mods easily accessible through this game hub, makes it trivial to track that rate of ScummVM's evolution. All one would have to do is search in the game hub to see all of the games and mods as well as their release dates for ScummVM.

Testability:

Interestingly, having an in-app game hub can potentially improve testability for every supported game and mod on ScummVM. Similar to how Steam has a community and review page/forum attached to each of its games on the store, our enhancement could have the same type of thing where users can leave comments, guides and reviews about particular games on the same page where players would go to download the game. These discussions or reviews can help give crucial testing information to ScummVM staff or mod developers including bugs, gameplay flaws or just overall thoughts and suggestions for the games.

System performance:

Due to the fact that this enhancement is done almost entirely on the front end and in a web server that is separate from ScummVM's internal architecture. Our enhancement should not significantly affect the system performance.

3.0 Alternative Implementation: Do one thing and Do it well

Although integrating this new component into ScummVM using its original components is in fact realistic. However, in practice, that implementation will involve modifying legacy

codes and making large changes to the already complicated structure of ScummVM. Therefore, introducing a new program alongside ScummVM can be a more practical alternative to the original solution.

Before introducing the solution, some pros and cons need to be explained. A big problem of shipping another application to assist the existing ScummVM is compatibility. There are some wonderful cross-platform and FOSS application building frameworks on the market, but there is no guarantee that they will be able to support all platforms that ScummVM supports. For some rarely seen platforms today, like PS2, the mod manager and online game downloader will not be available to them. The biggest advantage will simply be easy development and way less work on maintenance. Mod loading and game downloading in a big context is nothing more than file operations (mostly done in %APPDATA%\ScummVM\scummvm.ini on Windows) or memory injections. Those 2 use cases can be easily done using a cross-platform development framework like QT and Cheat Engine (Not cross platform but FOSS, it can be used as a reference).

Integration:

There is no need to do any integration, or more precisely internal integration, with ScummVM. The enhancement will be implemented as a new executable (application) that only modify the config files of ScummVM and do memory injection when possible.

3.1 Alternative Implementation Consequences

Maintainability:

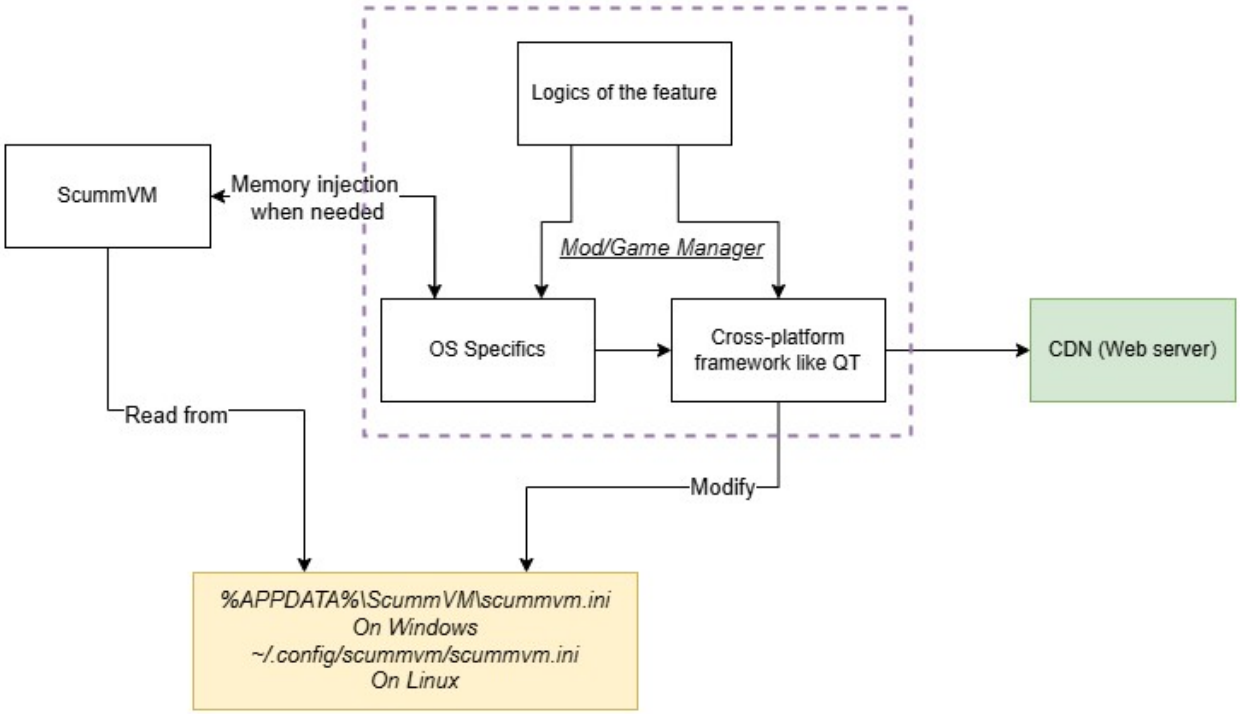
This will most likely be the same as the original implementation with an additional benefit that does not increase the overall complexity of ScummVM itself. However, this implementation will carry out the cost of managing one more repository which might make it harder to be organized. Another possible problem will be the change of ScummVM file usage pattern. The mod/game manager must be in sync with the ScummVM roadmap.

Testability & Evolvability:

The alternative implementation will carry out all benefits from the original one and will allow it to evolve itself without interference with the ScummVM's core functionalities. This will also make the testing easier compared to the original implementation since they can be tested separately. The problem as mentioned before will be the lack of support on rarely seen platforms like PS2. It is possible to extract the GUI component from ScummVM and use it to build a new application, but it will make it overly complicated.

Performance:

Since the enhancement and its implementation will have nothing to do with the ScummVM codebase. There will be zero impact on the performance of ScummVM at all.



4.0 SAAM Analysis

This section identifies the major stakeholders in this proposed enhancement and outlines their key non-functional requirements (NFRs) using a Software Architecture Analysis Method (SAAM) approach. For each of these stakeholders, we will provide the implications of both the main and alternative implementations for the new enhancement on their respective NFRs.

4.1 Stakeholder 1: End Users

End users of ScummVM are the primary stakeholders who will benefit from the proposed enhancement. With the proposed in-app game hub, these users will experience an improved user experience, gaining easier access to a wide variety of games, including free games, fan-made content, and user mods.

Ease of Use

The ease of use is a critical non-functional requirement for the end users. Regardless of the implementation, the user interface should be intuitive and easy to navigate. The primary goal here is for users to be able to easily find, download, and install games and mods with minimal effort. Confusing interfaces would directly frustrate users, diminishing their experience with ScummVM.

Main Implementation: Obtaining games and modifications for said games is now significantly easier, especially since the user will not need to exit ScummVM for most cases.

Alt. Implementation: Having a separate application to download and run would create a slightly higher barrier for entry for our users, but this also has the added benefit of making the feature entirely optional for ScummVM users.

Performance

The game browsing, installation, and import processes must be smooth and responsive. Delays in these functions can cause frustration, especially when users are eager to quickly access new content. The game hub should be optimized for fast content retrieval, downloading, and importing, without significant lag or slowdowns, ensuring that end users can enjoy their gaming experience without interruption or frustration.

Main Implementation: The new features should have a minimal effect on overall performance, even with the introduction of needing to maintain responsiveness with online networking on top of hardware. Additionally, even if the user is not interacting with the feature, asynchronous fetching of data may contribute to the performance footprint of the app (both internet- and hardware-wise).

Alt. Implementation: The performance footprint of an entirely new application will likely have a higher performance impact on the operating system than an integration for the end user due to possible memory injections externally. Additionally, optimizations made to the original ScummVM application will not apply.

4.2 Stakeholder 2: ScummVM Developers

The core developers of ScummVM are responsible for maintaining and enhancing the platform. This group includes those directly involved in programming, fixing bugs, and integrating new features into ScummVM. They play a major role, ultimately implementing the enhancement for the next release. While the feature offers significant benefits to users, developers will need to ensure that the addition does not disrupt the existing codebase or the platform's stability. They will also need to make decisions about the technical feasibility of the feature and its long-term maintainability.

Maintainability

Since ScummVM is a complex and mature platform with a significant existing codebase, adding the game hub should not complicate future maintenance or introduce issues into existing features. The new code must be well-documented, modular, and easily extensible. This assists the developers in keeping the system managed long-term without excessive overhead. If the integration leads to a fragile or unmanageable codebase, it could increase the time spent maintaining the system, negatively affecting both developers and users in the long run.

Main Implementation: Keeping the new features secure and stable will require some contributors with expertise in networking and cybersecurity, but otherwise the feature should have little impact on the general maintainability of the original system by programmers.

Alt. Implementation: Any issues with the external application will not have any negative effects on ScummVM itself and vice versa. Any bugs that may be introduced will be isolated and thus more easily fixable without having to examine too deeply the interactions between the two.

Compatibility

The enhancement needs to be compatible with various operating systems, devices, and configurations that ScummVM supports. Introducing the enhancement requires testing and adjustments to guarantee it works smoothly across all platforms, including Windows, macOS, Linux, and even more obscure platforms like mobile operating systems. This compatibility ensures that this feature can be enjoyed by ScummVM's user base in its entirety.

Main Implementation: Due to direct integration with other ScummVM features, the new feature should be compatible on all platforms that ScummVM is otherwise available to. It will not be convenient for those without or who choose not to use internet connectivity, but this is not really a direct impact for the contributors of the project.

Alt. Implementation: As discussed in the proposal, compatibility might be an issue that impacts development. The application may not be supported on systems that support ScummVM itself, and having to make compatibility patches for an entirely new application could be cumbersome.

4.3 Stakeholder 3: Modders and Developers of Fan-Made Games

Modders and developers of fan-made games will be a significant group of stakeholders as they will be directly impacted by the enhancement. Many of these creators are passionate about contributing new content to the ScummVM community, including mods, translations, and entirely new fan-made games. This enhancement would provide these creators with an easier, more centralized platform to share their creations with users, enabling a more dynamic and engaging community.

Security

Security is an important non-functional requirement for independent developers. When developers are uploading content, they need assurance that the platform is protected against cyber threats. Conversely, ScummVM's developers need assurance that the independent developers aren't uploading malware, copyrighted or otherwise undesirable material to the hub. This reliability fosters trust between the ScummVM community and the ScummVM developers, encouraging them to share their work without concerns for the integrity of their contributions.

Main Implementation: Since we may not always want to personally vet everything that is published on the new enhancement, and since MITM (Man In The Middle) attacks are always a possibility with content distribution, considering our new UI as a possible attack vector by/against users, modders and other content producers is very important.

Alt. Implementation: No additional considerations beyond a separation of security features between the two applications.

Scalability

Scalability is a vital non-functional requirement for game content creators. As more users download and contribute to the game hub, the system must scale efficiently to handle increased traffic and large volumes of content without performance degradation. If the platform becomes sluggish or unreliable as more creators contribute, this could discourage modders from uploading their creations or make the hub less attractive to users. Ensuring that the hub can accommodate growth while maintaining performance is essential to the long-term success of the enhancement.

Main Implementation: The scalability of the new system is not a new issue. Server/network infrastructure will be a new concern, but the scalability will ultimately be determined by dependencies between the enhancement and the main ScummVM architecture. In the perspective of modders and other content producers as mentioned above, security considerations must be made before we scale the permissibility of uploads to the application.

Alt. Implementation: Ditto, but the scalability of the application itself is no longer bound by the ScummVM internal architecture. New features can be added and changed more freely, as well as any changes to the network hardware infrastructure also being detached from the original application.

4.4 SAAM Conclusion

Based on the above analysis, we have a conclusion that is a little more complicated than saying only one of these implementations would be appropriate. Each has its own list of advantages and disadvantages, and what way to weigh these and decide which is better lies ultimately upon the contributors of ScummVM

The main implementation gives less concerns for performance and compatibility, being a direct integration, at the potential sacrifice of scalability and maintainability.

The alternative implementation is, while certainly an undertaking as it requires the development of an entirely new application, also better in some ways. It gives users a choice of whether or not they wish to interact with our content distribution enhancement or ‘do it the old-fashioned way’, as well as freeing itself from the original and much older codebase, but possibly brings additional concerns for hardware performance and compatibility between devices.

Regardless of their individual advantages and disadvantages, both implementations of the enhancement have shown themselves to be valid, and choosing will depend on what the ScummVM user base, content creators and contributors find more necessary out of these considerations.

5.0 Conclusions and Lessons Learned

The proposed in-app hub for ScummVM offers significant benefits. This includes increased system evolvability, improved testability and improves the ability to moderate mods. However, this enhancement offers challenges such as higher server costs and increased maintenance demands. The ScummVM contributors would need to plan carefully to address these drawbacks. The alternative implementation would be a separate program alongside ScummVM. This provides a practical approach which would mitigate the risk associated with modifying the existing codebase, while maintaining similar benefits.

This enhancement must meet the diverse needs of stakeholders, such as providing seamless functionality for users, maintainability and compatibility for ScummVM developers and scalability and security for independent developers.

One key lesson learned from this report is how adding a relatively independent feature to an existing codebase risks muddying the architecture of the entire project. Another critical lesson learned from our research is that there are usually conflicts between developers and users while a new enhancement needs to be implemented. An easy-to-use fully integrated solution might be extremely attractive to users, but the additional complexities added will likely make developers less motivated. As a result, we were forced to consider the importance of balancing user experience with long-term maintainability.

5.1 Data Dictionary

- **Mod** - Short for “modification”, represents a user-made edit to a game that changes its functionality by adding new features or modifying existing ones
- **CDN** - Content Delivery Network, a network of geographically dispersed servers designed to cache and deliver content closer to end users.
- **FOSS** - Free and Open Source Software, software which is free (to download and modify) and open sources, meaning that the software’s source code is public.
- **GUI** - Graphical User **I**nterface, a form of user interface that allows users to interact with electronic systems through graphical icons and visual indicators.
- **SAAM** - Software Architecture Analysis Method, a method of analyzing the architecture of a piece of software by identifying stakeholders and non-functional requirements (NFRs)
- **NFR** - Non-functional requirements, is a requirement that specifies the operation of a system, rather than specific behaviours
- **PS2** - PlayStation 2, a game console developed by Sony Computer Entertainment and released on March 4th, 2000. Succeeded by PlayStation 3 in 2006.
- **MITM** - ‘Man **I**n The Middle’ refers to an attack in cybersecurity wherein the attacker inserts themselves between two parties (for example, hijacking file distribution between the providers of a given software and their end users).

5.2 Sources

“ScummVM”, https://www.scummvm.org/ - *Where almost all information came from*

“ScummVM Forums”, https://forums.scummvm.org/ - *Community made content and modifications*

“Cheat Engine”, https://github.com/cheat-engine/cheat-engine - *Application focusing on modding games and applications for personal use*

“QT”, https://www.qt.io/ - *A popular free and open source C++ framework for building GUI applications*