Video Presentation

https://youtu.be/Y5eXtdTkRho?si=OzyFBJ15jTa0Ya78

A3: Architectural Enhancements

Group 9

Xavier Awadalla (Team Leader)
Craig Tylman (Presenter)
Christopher Gil (Presenter)
Sam Tylman
Aaron Rivest
Felix Xing

(group roles listed on next slide)

https://www.scummvm.org https://github.com/scummvm/scummvm

Group member roles

Xavier Awadalla
Descriptions of stakeholders, selection & description of NFRs

Aaron Rivest
Comparison of implementations, impact on NFRs and Stakeholders

Christopher Gil
Sequence diagrams, presentation

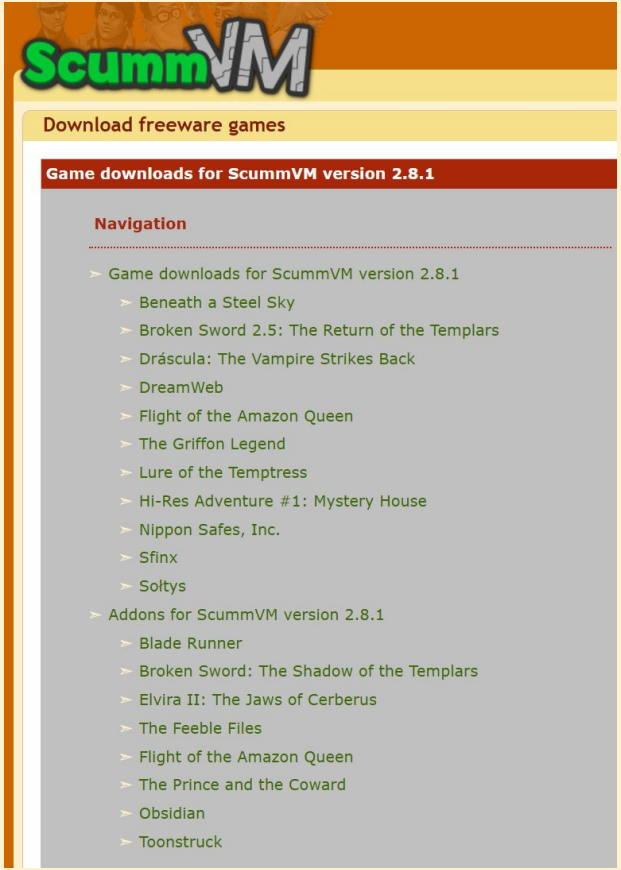
Craig Tylman
Description of enhancement & diagram, presentation

Felix Xing
Alternative implementations

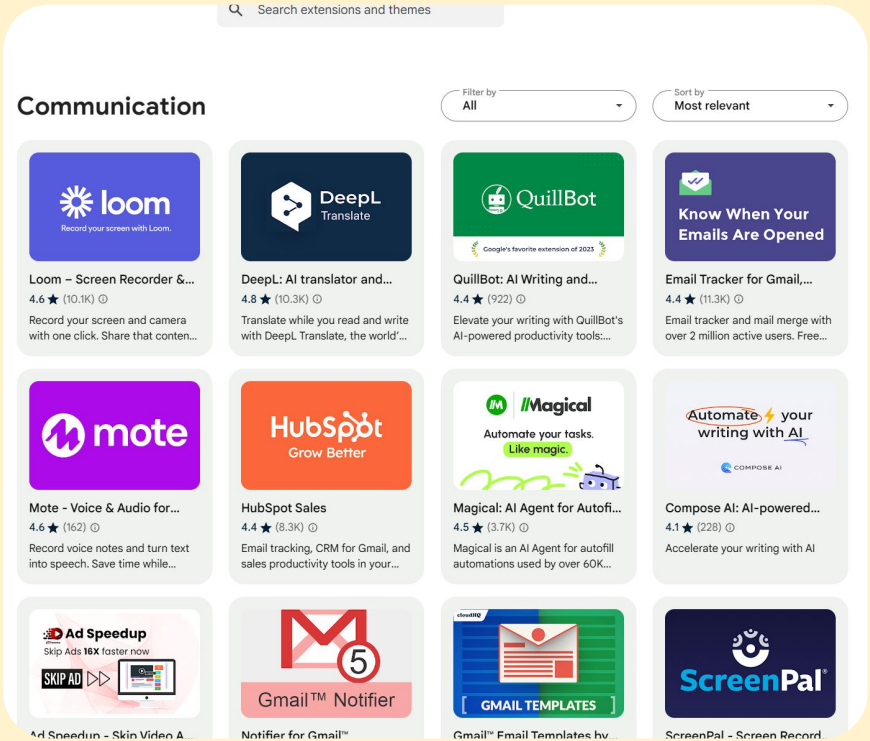
Sam Tylman
Enhancement Consequences

Enhancement Idea: Content Browser

- In-app hub to browse and download games and modifications
- Currently this done by searching various websites online, takes lots of time and effort.
- A native hub can skip many steps and save time
    - Serves as a convenient location for all downloadable content
    - Hosted server-side with a content delivery network
    - Client requests content, Content Browser obtains it and renders it
- Easy to use, and simplifies user workflow when downloading new games and mods by automatically handling file operations.



Exists for many well-known software



Content Browser: Consequences

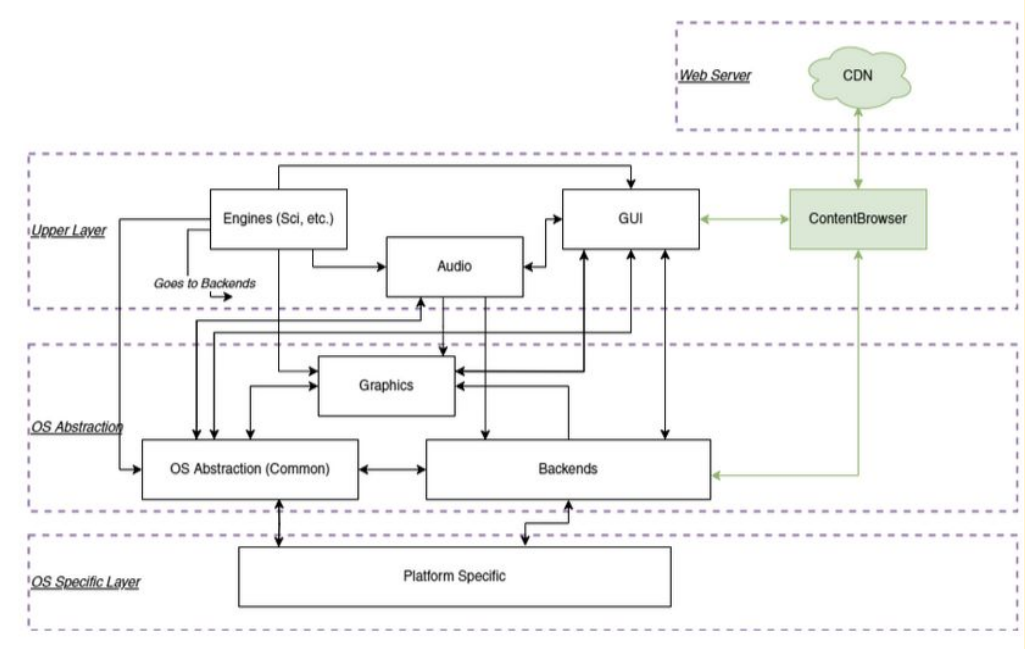
- Benefits to system evolvability and testability
  - Users can provide feedback and report bugs through the hub
  - Hub can be used to monitor system performance and evolution
- Drawbacks in system maintainability
  - More components to test, as more dependencies are created
  - Integration tests needed to ensure proper interactions between affected components
  - Increased chance of bugs and errors as codebase is increased
  - Higher server costs and maintenance
- Performance remains the same (client side). Server will need more hardware to handle the increased traffic
- No changes in architectural style. However, GUI and Backend may need some changes to properly process and render the new features of the Enhancement (properly save data to disk, display modslist to GUI).

Main Enhancement Integration

- Most changes are done on frontend
  - Content is fetched from server, processed and displayed
- A dedicated component “ContentBrowser” will handle fetching server data and manage content installation.
- ContentBrowser will depend on the GUI component to organize and display downloadable content information
- Backends will also be a dependency to install the contents into the user’s device storage (OS Dependent)

Main Enhancement Integration Diagram

- CDN isn't a direct component of ScummVM software, but included as ContentBrowser makes requests to it.
- ContentBrowser performs requests such as retrieving content listings, navigating pages, and downloading data & files.
- Backends will handle file and system operations based on the needs of ContentBrowser:
    - Saving games to local storage
    - Extracting, parsing and saving mods to local storage
    - Applying modifications to existing game assets and code
    - Changing Engine settings for rendering, shaders, audio



Main Enhancement Integration: Advantages & Disadvantages

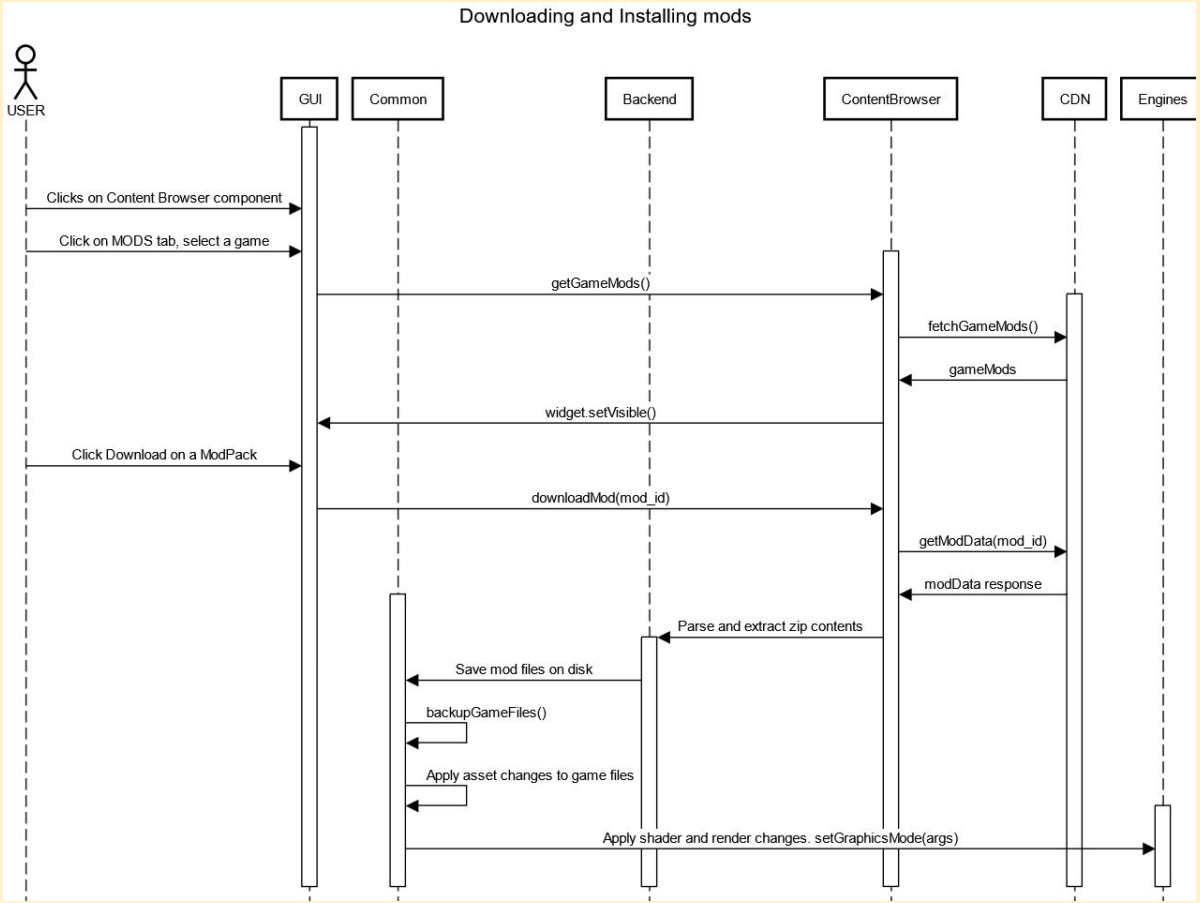
Advantages:

- Integration to existing ScummVM allows for re-usage of pre-existing components and libraries, reducing the need of developing such.
- Would be bundled together in ScummVM releases, rather than downloading a separate program.
- Increased compatibility & platform independence, as by including this within ScummVM, it could potentially be run in many platforms that are not necessarily PC only.

Disadvantages:

- Increased architectural complexity and potential for errors and bugs
- Increased development effort to integrate enhancement to existing legacy code, which could involve refactoring of legacy code in order to ensure proper operation.
- Risk of feature bloat and performance degradation: If implemented or optimized poorly, it could cause the GUI to be less responsive while waiting for server data.

Sequence Diagram: Downloading & Installing game mods



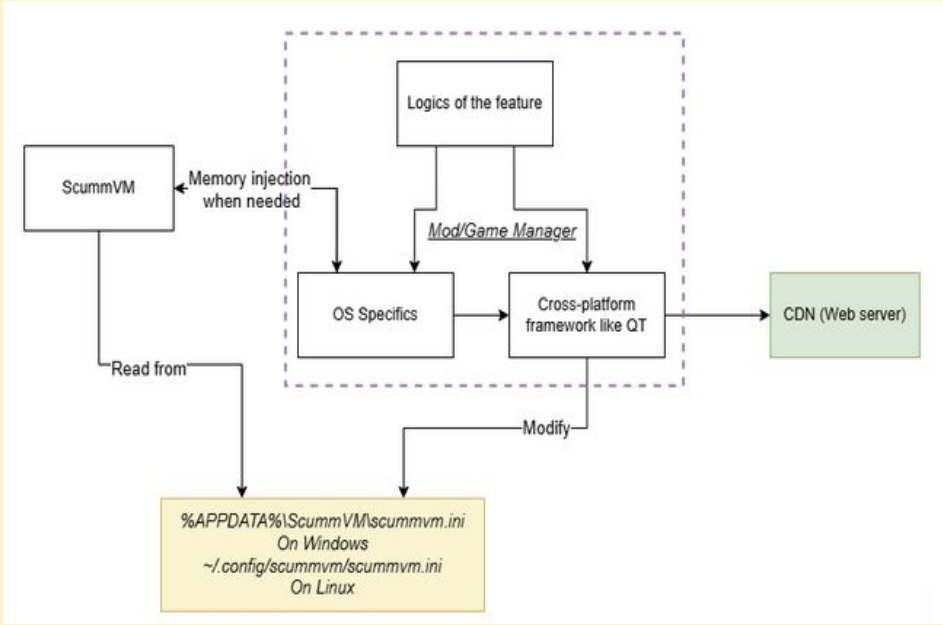
Alternative Implementation & Advantages

- A practical alternative to the original idea: introducing a new, separate program alongside ScummVM
- Biggest advantage of this would be easy development and less work on maintenance due to independence between both ScummVM and the new program.
    - Mod loading and game downloading is mostly file operations or memory injections performed by the new program. ScummVM performance remains unaffected as it operates separately.
- Another advantage: Barely any integration needed as the enhancement will be implemented as a new executable application. The ScummVM architecture remains unaffected.
- The program can evolve by itself without interfering with the core functionalities of ScummVM.

Alternative Implementation Diagram & Disadvantages

The Disadvantages:

- Reduced ease of use, as users would need to download and install a separate program.
- Reduced compatibility or increased development effort: Operating as a separate app would require extra effort to port it to other platforms than PC (e.g. PS2, Wii), or reducing the scope of compatibility support.
- Increased possibility of data corruption:
    - Async operations could access data in use by ScummVM and corrupt it
- Increased development effort, as many pre-existing libraries of ScummVM for file operation and OS abstraction would have to be re-written for the new program.



SAAM Analysis for both implementations

Stakeholder 1: End Users

- **Ease of Use:** Critical NFR for end users. GUI Should be intuitive and easy to use with minimal effort
  - **Main Implementation:** Significantly Easier to obtain games. No need to exit ScummVM as it can all be done within it.
  - **Alternate Implementation:** A separate application, a slightly higher learning curve for users. This also means users have to click off ScummVM to use the separate app.
- **Performance:** Game and mod browsing, download and install must be responsive. Lag and slowdowns damage user experience and cause frustration.
  - **Main Implementation:** New features should have reduced effect on performance. Maintain responsiveness and try to asynchronously fetch data while idle.
  - **Alternate Implementation:** Higher performance impact on the OS due to running separately and async memory injections. Any optimizations made to scummvm will not apply to it.

SAAM Analysis for both implementations

Stakeholder 2: ScummVM Developers

Maintainability: New features must not complicate future maintenance or introduce issues into the existing codebase.

- **Main Implementation**:
  - Requires well-documented, modular, and extensible code.
  - Needs contributors with expertise in networking and cybersecurity.
  - Minimal impact on overall maintainability if properly integrated.
- **Alternate Implementation**:
  - Issues with an external app are isolated from ScummVM's core system.
  - Easier debugging but requires maintaining a separate codebase.

Compatibility: Must work seamlessly across all platforms supported by ScummVM (Windows, macOS, Linux, mobile, etc.).

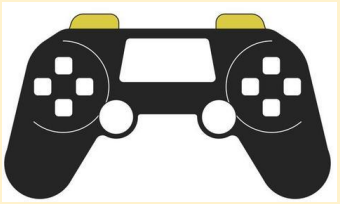
- **Main Implementation**:
  - Fully compatible with ScummVM's existing supported platforms.
  - Limited usability for systems without internet access but aligns with ScummVM's scope.
- **Alternate Implementation**:
  - Separate application might face compatibility issues on obscure platforms.
  - Maintaining compatibility patches for a standalone app adds development overhead.

SAAM Analysis for both implementations

Stakeholder 3: Modders and Developers of Fan-Made Games

Security: Secure platform fosters trust and encourages creators to share content.

- **Main Implementation**:
  - Protect against cyber threats, including potential MITM (Man-In-The-Middle) attacks.
  - UI considered as a potential attack vector; must ensure secure upload/download workflows.
  - Requires measures to prevent malware, copyright violations, or undesirable material.
- **Alternate Implementation**:
  - Isolating security measures to a standalone app simplifies management.
  - Security threats to one application do not impact the other.



Scalability: The platform must handle growing user and content creator activity without performance degradation.

- **Main Implementation**:
  - Scalability depends on dependencies between the new hub and ScummVM's architecture.
  - Growth requires balancing upload permissibility with robust security infrastructure.
- **Alternate Implementation**:
  - A separate application allows independent scalability and feature expansion.
  - Network hardware and infrastructure changes can be made without affecting ScummVM.

SAAM Conclusion

Complex Decision:

- No single implementation is definitively better; both have unique advantages and drawbacks.
- The choice depends on ScummVM contributors' priorities and user needs.

Main Implementation: Key Factors

- **Advantages**:
  - Direct integration reduces performance and compatibility concerns.
  - Works seamlessly with ScummVM's existing architecture.
- **Disadvantages**:
  - Potential trade-offs in scalability and maintainability.
  - Relies on modifying and integrating with an older, complex codebase.

Alternative Implementation: Key Factors

- **Advantages**:
  - Independent app frees the system from legacy code constraints.
  - Easier to scale and evolve independently from ScummVM's core.
- **Disadvantages**:
  - Requires significant effort to develop a standalone application.
  - May introduce hardware performance issues and device compatibility challenges.

Conclusions and Lessons Learned

- **SAAM Final Consideration**:
  - Both approaches are viable, and the choice depends on what contributors, users, and content creators value most: seamless integration vs. modular independence.
  - Main implementation selected, as it gives less concerns for performance and compatibility, with some sacrifice to scalability and maintainability.
- Overall, the in-app hub provides significant benefits to Scalability, evolvability, testability, and such, as mentioned before.
- The enhancement brings challenges such as higher server costs and maintenance demands. It must also meet diverse needs of stakeholders (users,developers,fan creators).
- **Lessons Learned**
  - Adding a relatively independent feature to an existing codebase can drastically affect the architecture of the entire project.
  - An easy to use solution might be extremely attractive to users, but the extra complexities will demotivate developers and could cause conflicts between them. Balancing user experience with long-term maintainability is key.

Thank you!

Analyzing the ScummVM code & structure