ScummVM Architecture Enhancement

Presented by Jacob McMullen and Reid Stobo

https://youtu.be/XqpOGOvbbtQ

Evan Kreutzwiser

Mike Stefan

Ryan Jacobson
(Group Leader)

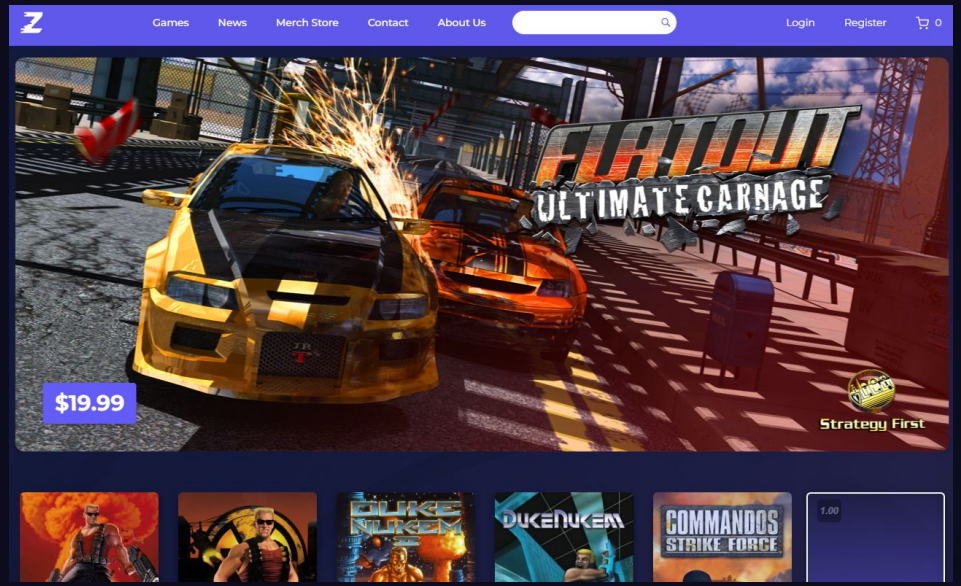
Jacob McMullen

Kashan Rauf

Reid Stobo

Team Orangutan 1F9A7

ZOOM PLATFORM

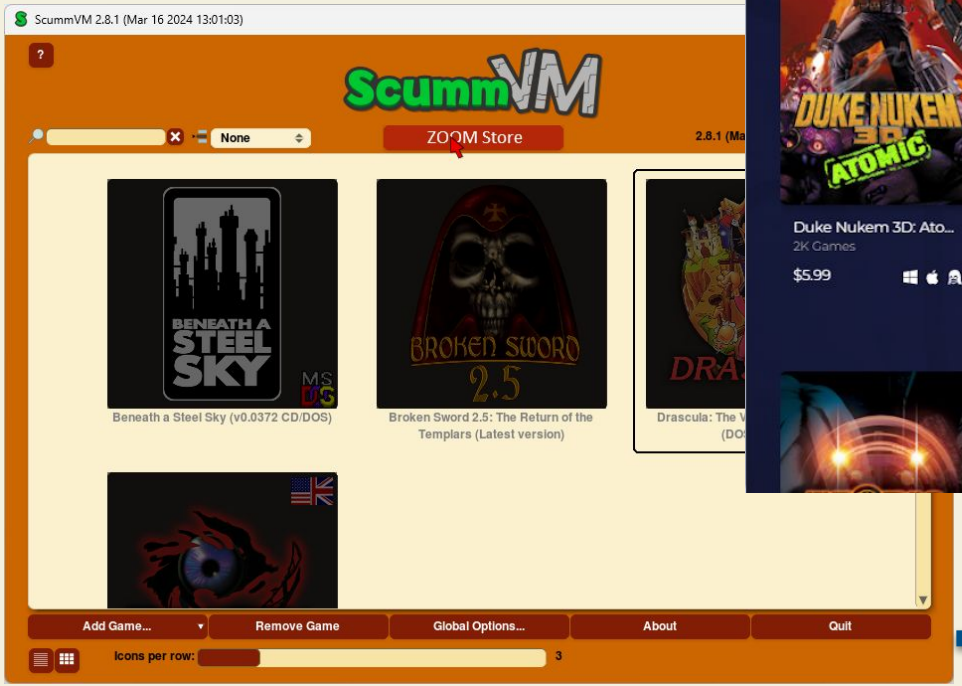


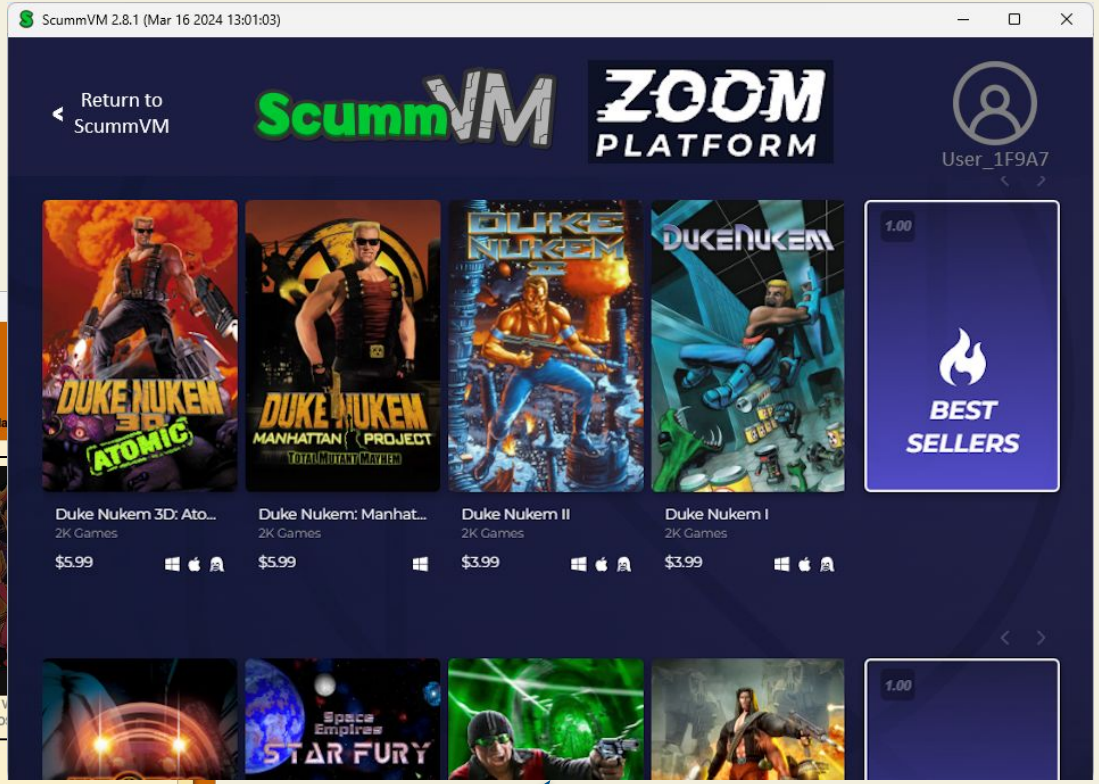
735
OFFICIALLY LICENSED GAMES

ALWAYS DRM FREE

SAFE & SECURE CHECKOUT

Proposed Enhancement: Integrated ZOOM Storefront





ZOOM Storefront opens from within ScummVM

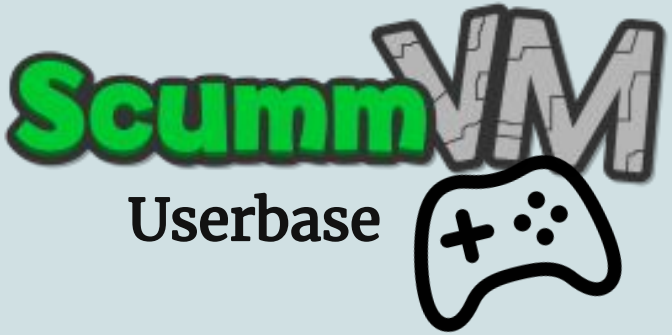
Stakeholders

- Volunteer developers
- Designed ScummVM to be highly modular
- Care most about modifiability and scalability

Stakeholders

- Retro game marketplace
- Proposed enhancement serves as a new market avenue
- Care most about supportability and security

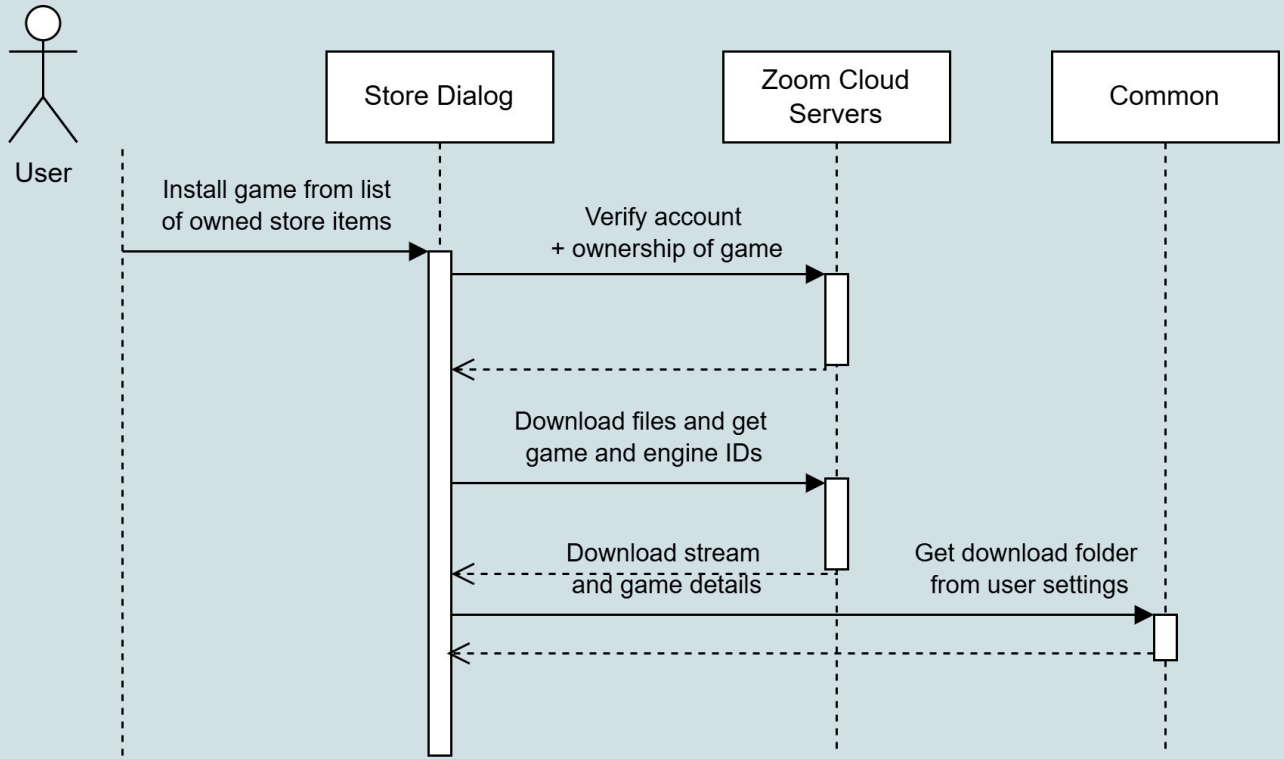
Stakeholders



- Diverse userbase
- Accesses many different games on many different operating systems
- Care most about availability, and portability

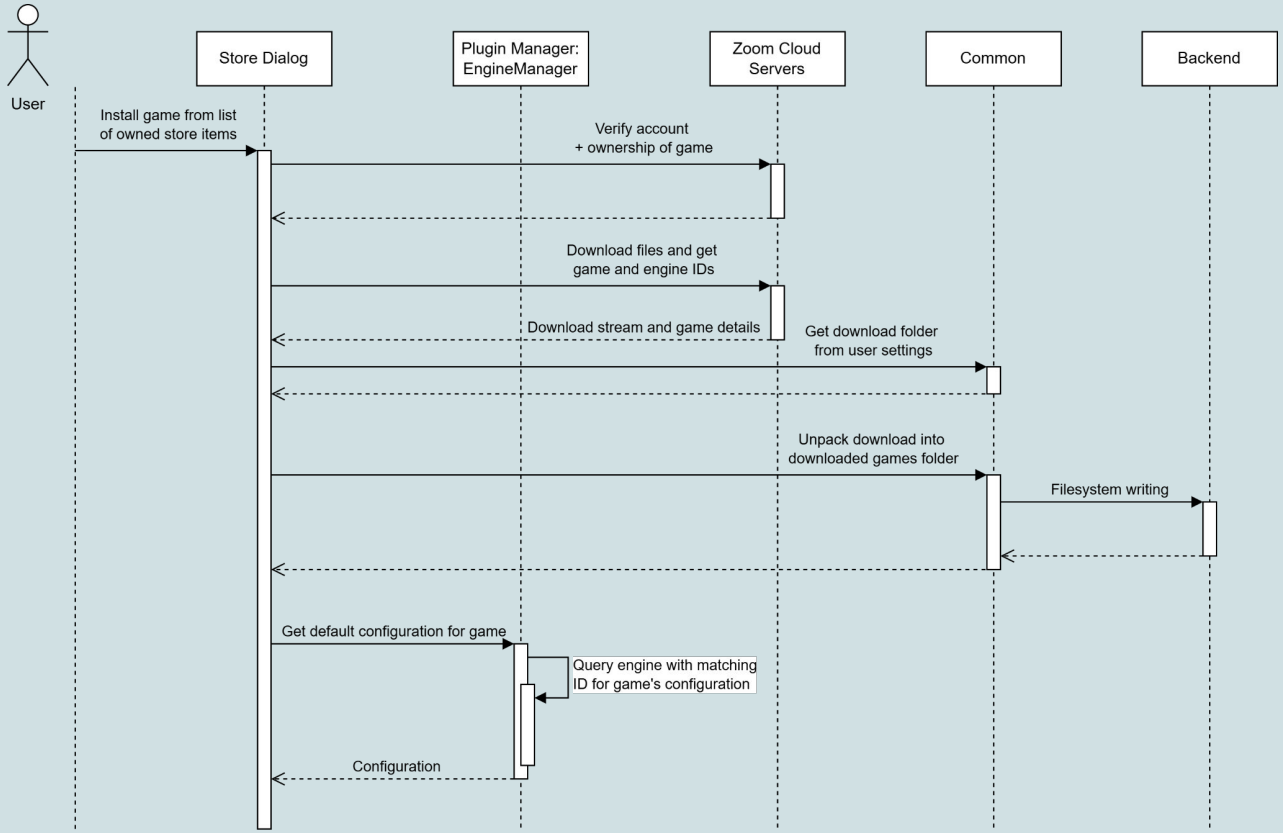
Use Case: Installing Game from Store

- Communication between ScummVM Store and ZOOM servers
- Common storage of user settings



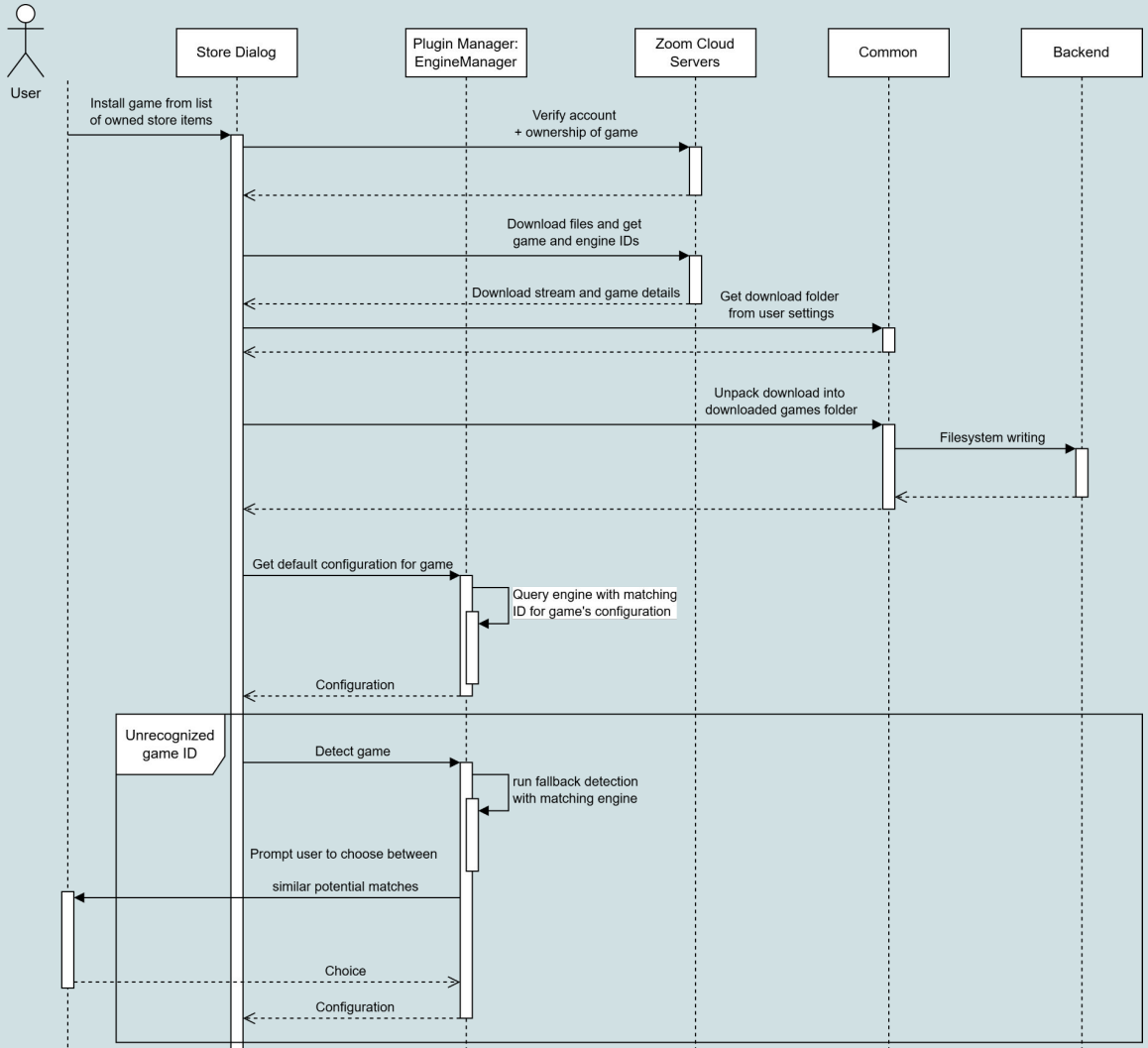
Use Case: Installing Game from Store

- Place game data in the correct location
- Load default config. on behalf of user



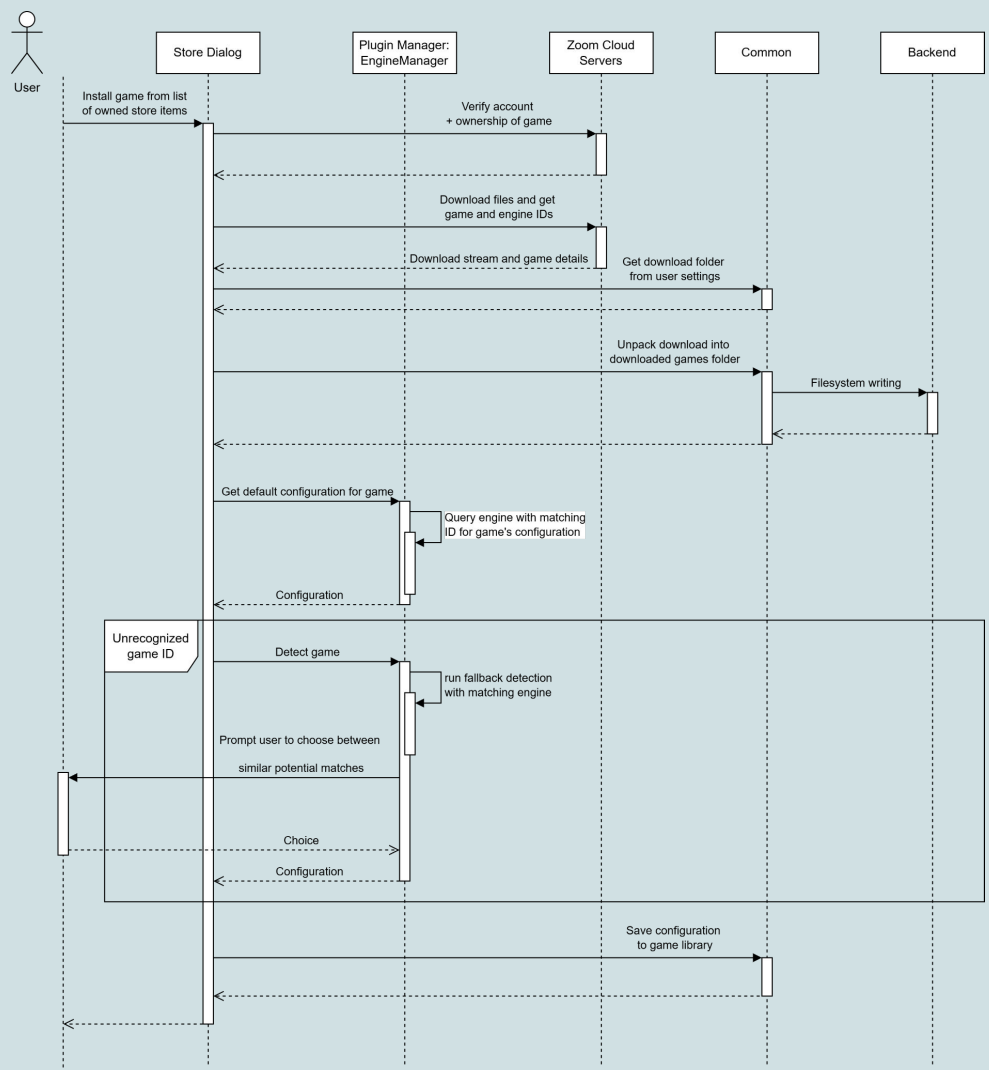
Use Case: Installing Game from Store

- If game unrecognised, use fallback detection
- Allow user final decision to confirm correct detection

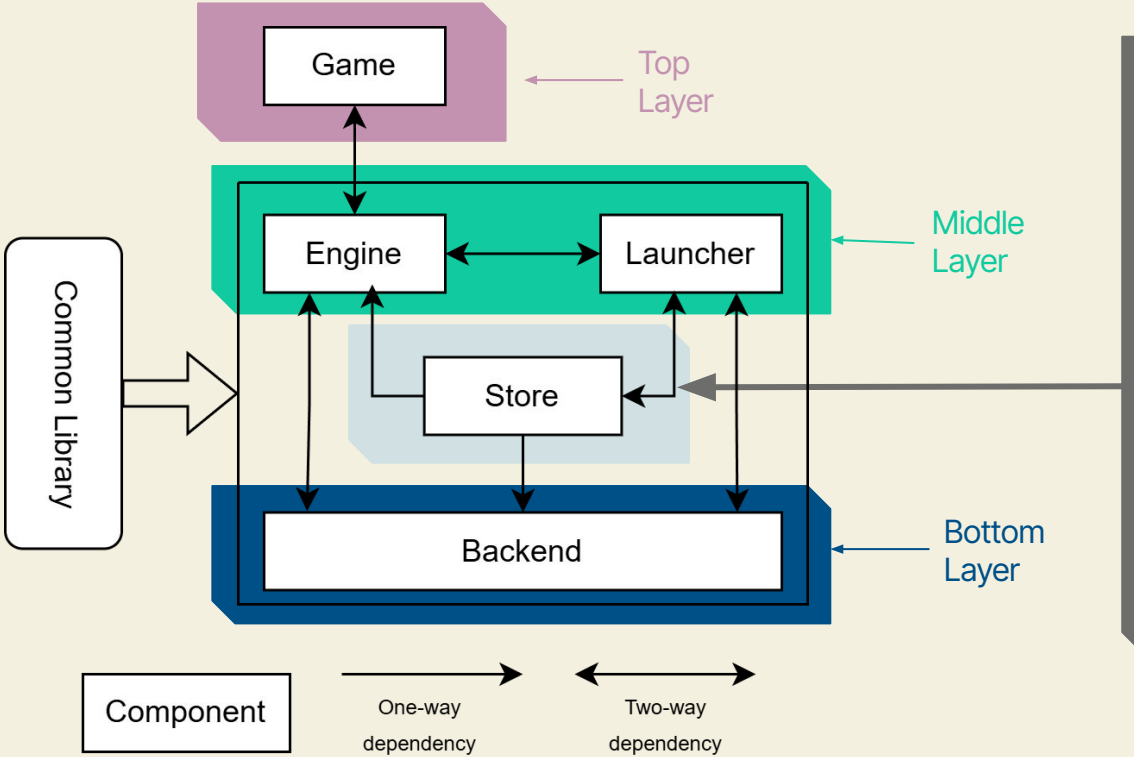


Use Case: Installing Game from Store

- Save config. to common
- Game is in users library, ready for launch



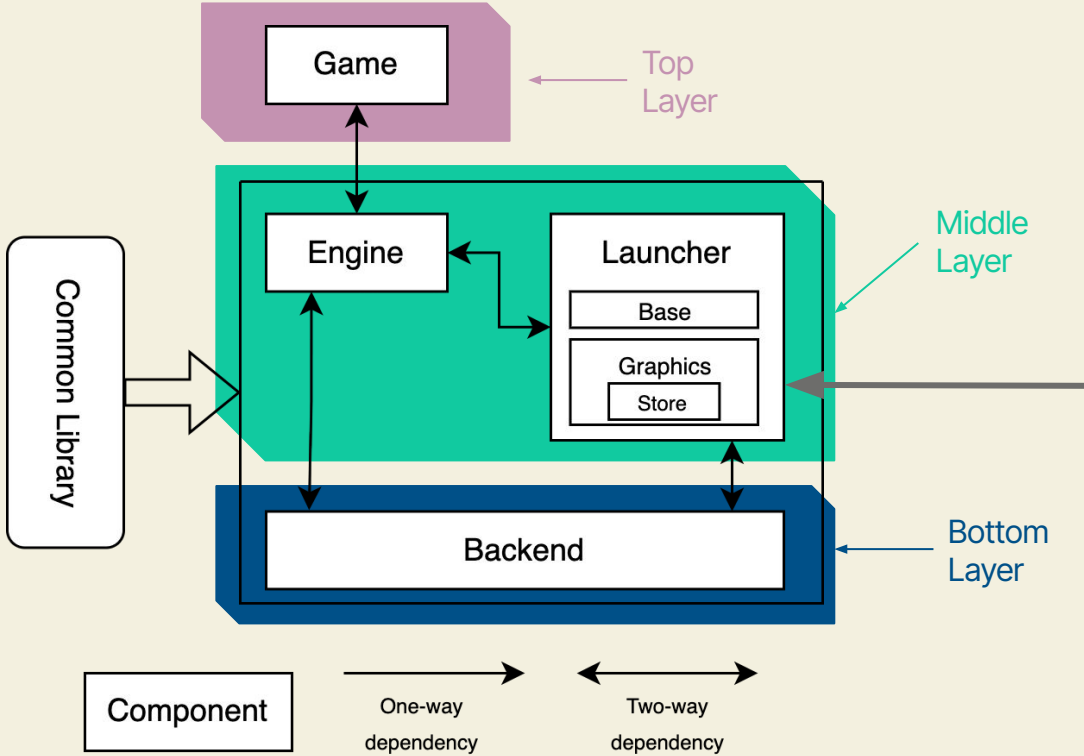
Implementation 1:
Storefront as Top-Level Component



Is the Store component part of the **middle layer**, or its own **in-between layer**?

Does it overcomplicate the layered architectural style?

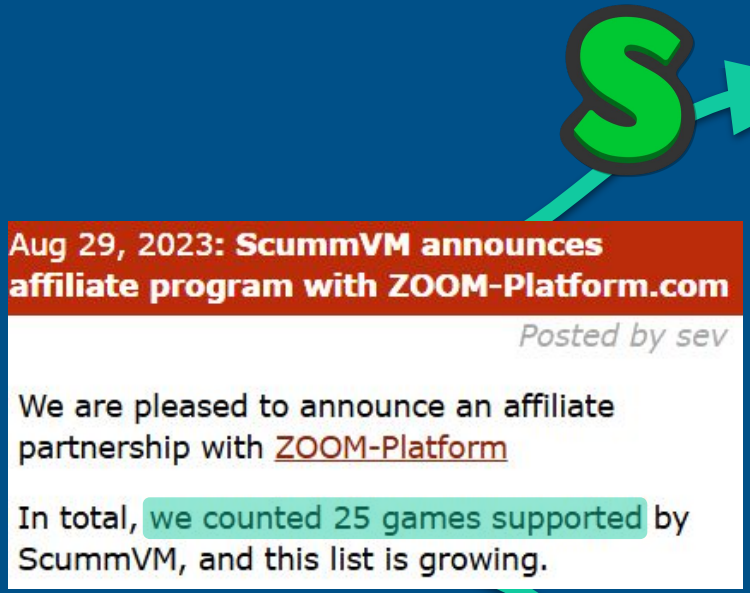
Implementation 2:
Storefront as GUI Subcomponent



We have preserved the **layered architectural style.** But have we made the launcher component too bloated?

Risk: Instability of Partnership

Limitation: Small Compatible Library

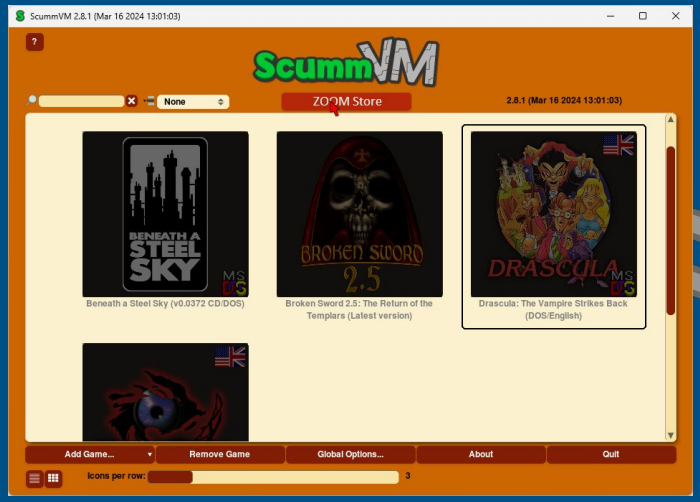


Develop for Existing ZOOM Games

Expand the Compatible Library

Negotiate Licenses for Existing ScummVM games

Risk: Introduction of Security Vulnerabilities

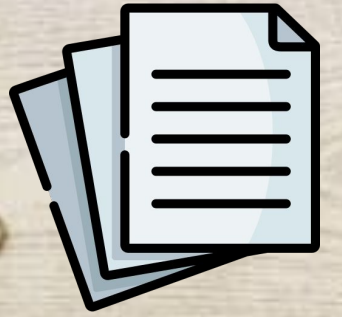


Payment Data

User Data

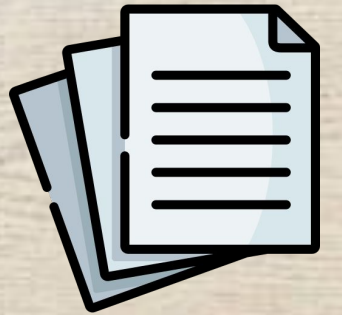
Sensitive user data must not be vulnerable through this new integration

Plans for Testing



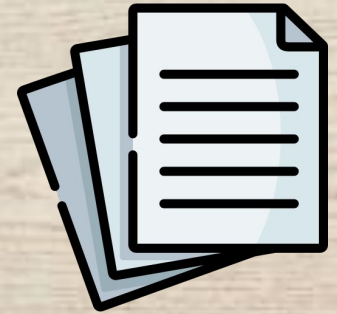
Unit tests for Systems which Support Storefront

- Test functionality of each component
- Test relations between components and storefront



Integration and End-to-End Tests

- Included in each set of unit tests
- Tests the various use cases found within system and enhancement

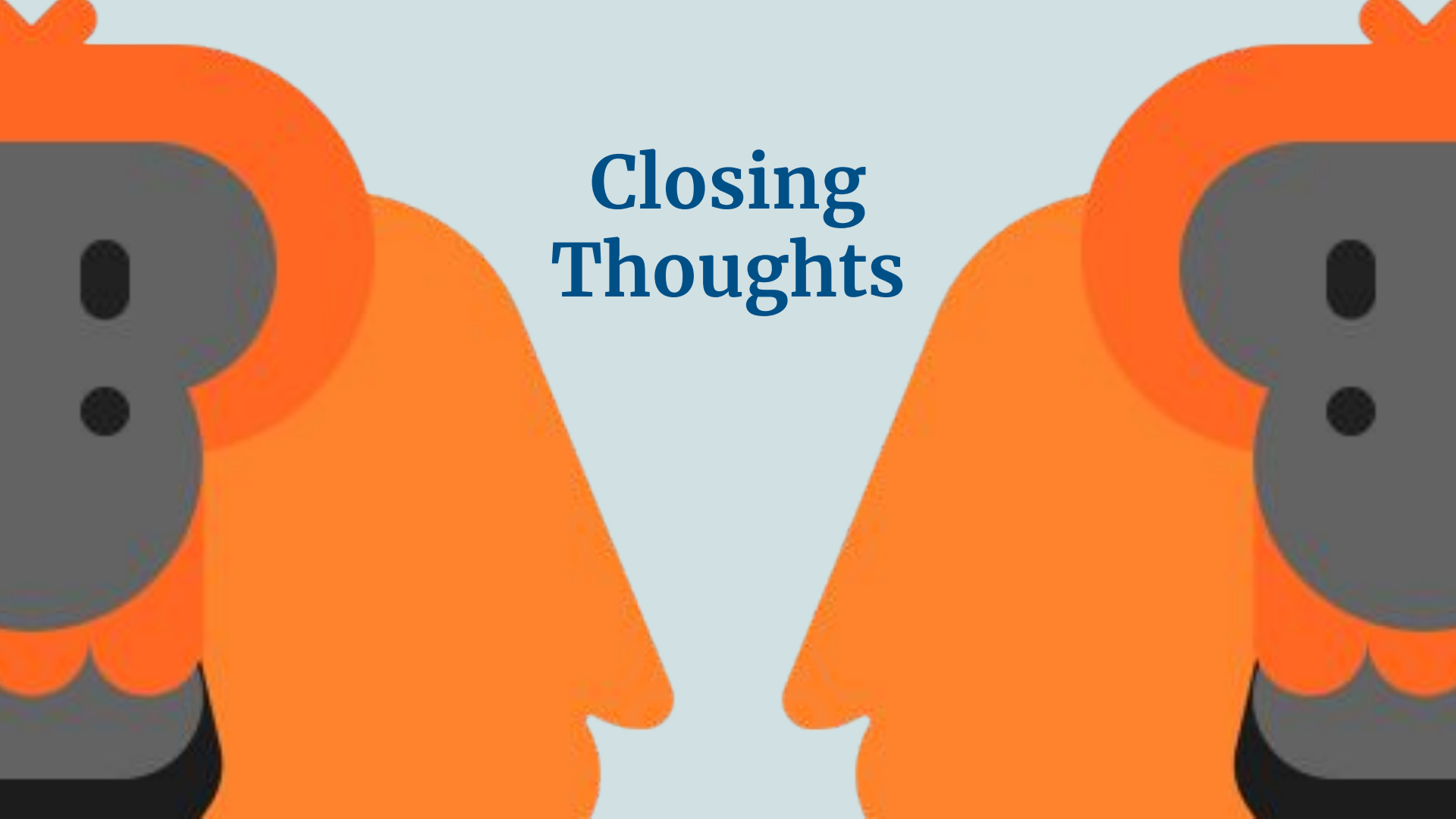


Unit tests for Systems which don't Support Storefront

- Test functionality of each component
- Test components operate the same as before storefront was introduced

Regression Test Suite

- Deployed after enhancement released
- Test if ScummVM and storefront are still performing as intended after launch



Closing Thoughts