Enhancement Proposal for Kodi

Aidan Wolf: 20atw@queensu.ca
Aryan Chalwa: 20acd10@queensu.ca
Marcus Secord: 20ms130@queensu.ca
Nick Levy: 18npl1@queensu.ca
Ronald Sun: 20rs80@queensu.ca
Udochukwu Ojeh: 20uwo@queensu.ca

CISC 322/326: Software Architecture
TA: Eric Lams
Professor Karim Jahed

5 December, 2023

Abstract

Some of the information in this report references “The Concrete Architecture of Kodi", by Group 14: BIPimbap.[1] This report aims to improve Kodi by proposing a new feature. The proposed enhancement for Kodi is the integration of smart home devices with the Kodi software. Smart home devices have become very popular over the last few years with the introduction of devices such as Google Home and Amazon Alexa. As such, the integration of said devices is beneficial to those who already own smart homes, or those with accessibility needs that struggle with the current input methods such as someone who has Parkinsons. This report examines the current state of Kodi and how it relies on basic input devices such as keyboard and mouse, giving more motivation for the enhancement. It also covers how the enhancement affects other components and the quality attributes of Kodi such as performance and maintainability. To further realize the enhancement, two methods of implementations are discussed alongside impacts on stakeholders' NFRs, interactions with the current conceptual architecture and potential risks they could introduce. The first method of implementation uses Kodi's web server to connect to the smart home through the user's network as well as the user input component to process the voice commands. The alternate method of implementation is a hardware based approach. It would be implemented with a smart hub ecosystem, making voice control automation systems more accessible compared to pairing the user's peripherals. The first implementation is a better option as a hardware based approach as it requires users to purchase it. This report serves as a valuable resource for developers who are interested in the continued growth of Kodi.

Introduction

This comprehensive report delves into the prospect of enhancing the Kodi software through the integration of smart home devices. It explores the unique possibilities and considerations involved in bridging Kodi's entertainment hub with the ever growing realm of smart home technologies. The report provides a detailed overview of the proposed enhancement, outlining the key objectives and motivations behind integrating smart home functionalities into Kodi. Furthermore, it delves into the current state of Kodi, detailing the software's reliance on traditional input methods and highlighting the potential for smart home integration to revolutionize user accessibility and interaction. Throughout the report, various aspects of the implementation are explored, including interactions with existing Kodi features, effects on quality and design considerations, plans for testing, and potential risks. There are two implementation approaches described in the report, the main approach and an alternative approach. Potential stakeholders, non-functional requirements, and the impact each implementation strategy may have on Kodi's architecture and its users are assessed. The findings emphasize the importance of upholding system maintainability, evolvability, testability, security, and performance while integrating smart home functionalities. An implicit invocation architectural style was deemed to be necessary for the ease of implementation of smart home devices into the overall Kodi architecture. Moreover, the report also addresses potential risks, such as compatibility issues, security vulnerabilities, and privacy concerns, offering insights into the challenges and complexities inherent in the integration of smart home devices. A comparison is drawn between the primary implementation and an alternative approach, weighing the advantages and disadvantages of each method. While the main implementation prioritizes

[1] https://aryanchawla1.github.io/cisc326/a2.html

existing smart home device integration, the alternative proposal focuses on a simplified, hardware-based Kodi system to cater to a less tech-savvy audience. In summation, this report serves as a comprehensive guide to understanding the intricacies and implications of integrating smart home functionalities into Kodi, highlighting potential challenges, risks, and necessary steps in pursuit of a more integrated and immersive media player experience.

The Enhancement and Motivation

With smart homes becoming more integrated with the surrounding technologies around them in households, residents have become more accustomed with voice control, a much more efficient and intricate method to declare commands. Recently, smart home systems integrate with playing music, control lights, and create grocery lists for users. Sooner or later, smart home systems will integrate with every aspect of a home, creating an opportunity for all media playback to be accessible by voice control.

Additionally, not all consumers who desire playing their media content have the same capabilities. People with disabilities: such as blindness, paralysis, Parkinson's, and more, struggle with the average means to play media on their devices. Their solution is voice control. Thus, consider the implementation to allow media content to be played and managed via audio commands.

To capitalize on this opportunity, and also create a solution for consumers, an enhancement for Kodi is presented. Allowing voice commands to be registered by smart homes, and then navigate and operate Kodi like a remote or keyboard and mouse would. Integrating the Smart Home: such as Echo and Home, with the Kodi software will allow audio commands to be used to interface with the GUI and perform actions that would regularly be used by input devices. Not only would the same actions be replicated by voice commands, but rather, human-like communication would be accommodated to allow easier navigation.

The Smart Home enhancement is well motivated. Hands-free operation reduces physical strain and multi-tasking for users. This means that rather than navigating via a remote or keyboard and mouse which requires physical access and coordination would rather be completed through voice commands. Furthermore, with voice commands, language navigation would allow more intricate and conversational commands, like second nature. This would allow easier navigation, more accommodating accessibility, and more context within commands.

Current State of Kodi

The current state of Kodi primarily relies on traditional input methods.

- Mouse and keyboard for computer and desktop users
- Touchscreen for mobile devices and/or tablets
- Remote control for TVs and other big screens
- Controllers for Xbox or other systems like that for playing games, etc.

Though these are the most common ones, it does say that they can include more through emulators that emulate the peripherals of its game platforms, in which case things like light guns, and flight simulator joysticks can also be used. The target for the Smart Home Ecosystem is to take over the less intensive controls, like navigating the screen, choosing which media to play, adjusting volume, etc. Game controls cannot be overtaken due to its complexity. However, that is not to say that each command results in one input. The reason for the Smart Home is that in a single command, it can save you several inputs, suppose jumping from login into directly playing a movie. Outside of Kodi, the inputs in the client can be customized to match with other elements of Kodi. For instance, when a song is requested through voice, Kodi starts playing, and the RGB lighting in your room starts to go off, or when a movie is requested, the movie starts and the lights in your room start to dim. As such, the current state of Kodi relative to the enhancement is preserved, and other features are just added on.

Interactions With Other Features

Implementing this enhancement will require the interaction with many Kodi components and features. However, some changes will be needed to improve and accommodate the new enhancement as well. This change will require interaction from the presentation layer to the data layer, and the layers in between, interacting with a plentitude of features.

Referring to the Kodi Architecture, the user input module accommodates Keyboard, Mouse, and Remote Control. With voice controls, the smart home would interact with the user input component and a new supported input device. Sadly, this new input device can't be directly translated into a keyboard, mouse, or remote control device input, but rather needs handling on its own for GUI elements and more. This means that navigation within the Kodi software would be selecting GUI elements to interact with rather than using a cursor. Thus, the enhancement would interact with Kodi's current input device features.

With the event server following a publish subscribe architecture style, new events caused by the audio commands would need to be accommodated. The player core, which handles the video playback, has some specific GUI and interaction capability features. The enhancement would also interact with this feature.

Finally, the views and files list, containing all the media files will interact with the smart home system. If the audio command directly asks to play content, the views component with the file list and library would handle the query. This would interact with Kodi's feature to store all the media in one place.

It is imperative to make it clear the smart home is accommodating the voice control input handling, and sending commands to Kodi to handle. With this in mind, a connection between the Kodi and the smart home must be established before any voice control would work. Thus, some sort of pairing component needs to be developed as a feature to allow communication between the smart home and Kodi.

Effects on Quality, and Design Considerations

Integrating the smart home ecosystem directly into Kodi will involve making significant modifications to the core codebase, and as such, it will make a direct impact on maintainability, evolvability, testability, and performance.

Maintainability: modifying the core codebase of Kodi presents a large risk. The maintainability of the system will heavily depend on how optimized the implementation is. Careful documentation becomes crucial to help future developers understand the modifications were made and how it was integrated with existing code. Version control, which is for sure already in use, will allow developers to understand the evolution of the code base. Logging errors and testing will all play an important aspect of maintainability.

Evolvability: ensuring that the integration is well-designed and modular, it could be easier to evolve the smart home functionality along with regular system updates. This can also position Kodi as a modern, evolving media center, with regular updates to support new Smart Home devices. However, rapidly changing Smart Home technologies (with its own updates) might cause compatibility issues with existing installations.

Performance: the overall user experience should be enhanced without affecting performance if the functionalities are implemented efficiently. Properly optimized code with async processing can help mitigate performance concerns. However, if poorly optimized, it can increase resource usage caused by continuous background processes, or latency in connection, etc. This concern arises from the consideration that the ecosystem might need to continuously monitor the activities of the external devices.

Testability: with a clear understanding of what kind of interactions the Smart Home functionalities would have with the current Kodi architecture, testing becomes more manageable. Well defined subsystems and components can be tested individually, but also ensure to test regressively as some dependencies might exist. Compatibility testing would likely be more challenging as there is a wide range of devices and platforms that should be compatible.

Some best practices to keep in mind when designing are as follow:

- Modular Design: Integrate the new features through modular components, minimizing the dependencies on existing Kodi subsystems.
- Compatibility Handling: Regular updates to the Smart Home will be necessary to ensure compatibility with the latest standards to the devices.
- User Configuration: Allow users the choice to enable or disable the ecosystem features based on their preferences to avoid unnecessary resource consumption.
- Thorough Testing: Ensures that everything works as expected.
- Documentation: Helps explain all the processes and implementation of the integration.
- Community Involvement: Leveraging Kodi’s committed community to gather feedback and address the issues promptly. This can also lead to valuable contributions/improvements.

Plans for Testing

Testing the impact of interactions resulting from the integration of smart home functionalities into Kodi’s architecture necessitates a comprehensive approach. Integration testing is crucial to ensure the smooth communication and interoperability between the modified subsystems and directories affected by the smart home device integration. This entails conducting tests to validate that the updated or newly introduced functionalities align harmoniously with existing components, ensuring their compatibility within the Kodi architecture. Additionally, usability testing is required post-integration to gauge how intuitively users can interact with the incorporated smart home features within the Kodi environment. Finally, security testing is necessary to identify and mitigate potential vulnerabilities arising from the modification of Kodi’s directories to implement smart home devices. This testing involves conducting vulnerability assessments and scrutinizing the data handling mechanisms of the smart home devices to reinforce Kodi’s resilience against potential privacy and security risks.

Main Implementation

For the first implementation of the proposed enhancement, Kodi will connect directly to the smart home device through the user's network. The smart home will be able to communicate with Kodi through an API for Kodi smart home integration. Any voice commands the user says to their smart home (assuming it is a command for Kodi) will be sent to Kodi where the user input component will use speech recognition to determine an event based on the input. The Kodi Core will then broadcast the event (implicit invocation) for the rest of the system to handle.

Stakeholders, NFRs, and Impact

When it comes to the addition of smart home integration support for devices such as Google Home and Amazon Echo, there are a few stakeholders that are impacted. These stakeholders include the developers of Kodi, users with smart homes devices, smart home device companies, and non tech savvy customers or customers with accessibility needs.

The first and most obvious stakeholder are the developers of Kodi. The developers of Kodi want their software to provide a good experience to their users, as well as know realistic expectations of the implementation of said feature. The NFRs that the developers may have are related to maintainability and security. The developers of Kodi are concerned with maintainability as they need to have automated tests that are run periodically to ensure all systems still work. A NFR could be that all tests will be automated and can be run in less than 6 hours. Another important aspect they will be concerned about is security. They need to ensure that new features do not also create new vulnerabilities that puts their user's personal information at risk. As such, a NFR might be that all smart home devices must be authorized and validated before interacting with Kodi. With this implementation of the feature there are some impacts that it may have on the NFRs. Maintainability can be impacted as it requires external devices (smart home) to be able to properly test. This may require additional equipment and new ways of implementing automated tests. Security can be impacted as vulnerabilities on the smart home side could affect Kodi. It may be hard to develop around vulnerabilities that are not part of their system. For example, if a smart home device had a vulnerability and allowed for a hacker to access it, the device could act as an entry point into the Kodi system.

The developers of various smart home device companies would also have concerns with the smart home integration feature being added. The important types of NFRs that they would have are portability and security. They would have very similar reasoning and NFR pertaining to security as they cannot control vulnerabilities in Kodi. They would also want the feature to be portable as using platform specific code could result in some smart home devices not working properly or at all with Kodi. A NFR for this could be that the program will be able to connect with android smart home operating systems as well as apple smart home operating systems. This means that the developers would have to take into consideration the wide variety of smart home devices on the market and implement the feature so that most if not all devices are compatible.

The users of Kodi are another major stakeholder for this feature, without them there is no reason to implement this feature. The important types of NFRs that users have interest in is security, usability, and performance. As mentioned previously, security is a major concern in terms of user privacy. The users want to ensure that their data is not mishandled as well as the smart home microphone not being constantly active. Another major factor for users is to have a very easy setup process as well as continued use once the smart home device is connected. A NFR for this could be less than 5% of users report issues with using their smart home device with Kodi. Users also want the system to be very responsive and react quickly to their inputs. In terms of performance, the users may have a NFR such as 80% of the time the system should receive the signal from the smart device and respond with the intended command from the user in less than 5 seconds. In terms of impact, the enhancement can affect the NFRs relating to security as data is transferred through the user's network. If the transfer is not secure or properly handled it could lead to the data being stolen.Since the integration of smart home devices will use voice commands from the user, the usability can be impacted as the developers have to account for various ways that different users say commands for the same functionality. For example one user may say "pause video" and another says "stop movie", while different, they still refer to the same functionality. The performance of the feature is impacted by one major issue, network speeds. Since the feature relies on the user's home network, the rate at which Kodi receives input from the smart home device is dependent on the network speed. This could also be affected by the amount of traffic on the user's network. This makes the NFRs regarding performance harder to manage as the performance is not only dependent on Kodi, but the user as well.

Effects on the Conceptual Architectures, Directories and Files

The addition of a Smart Home Event Server component within Kodi's conceptual architecture plays an instrumental role in enabling seamless integration with smart home devices. Located within the Kodi Core, this component serves as a bridge between Kodi's functionalities and the interconnected smart home environment. By facilitating the publication and subscribing of events to smart home devices, the Smart Home Event Server component establishes a cohesive communication framework. It allows for a bidirectional flow of information, ensuring that Kodi can trigger actions through smart home devices, and react to external responses from said devices. Through the addition of the Smart Home Event Server component, Kodi can manage and orchestrate interactions with smart home devices, expanding upon the media player's home theater capabilities.

In the context of the User Input subsystem of the Skinning Engine within Kodi's architecture, input from smart home devices follows a structured process. When smart home devices generate commands or triggers, these inputs are transmitted through the home network, where they subsequently seek to register within Kodi. Kodi's User Input subsystem plays a pivotal role in managing and handling this input. A directory called 'smarthomedevices' located in the 'input' directory of the User Input subsystem handles processes related to any given smart home device.

Furthermore, the GUI subsystem of the Skinning Engine is affected, as it is used to display notifications or alerts to inform users about any status changes or incoming commands from the smart home devices. This ensures that users remain informed and can promptly respond or interact with their smart devices as needed without leaving the Kodi environment. It also changes the way the UI is navigated as it cannot be done spatially like it would be with a remote. With voice commands, the UI is navigated by saying the names of elements or functions within Kodi.

The alteration of the Window Manager within the Skinning Engine was an essential augmentation due to smart home device integration in Kodi. By incorporating smart home notifications into the Window Manager subsystem, it ensures that users can receive timely and relevant alerts or status updates from their connected smart home devices directly within the

Kodi user interface. Users can be promptly informed about changes or events in their smart home ecosystem while engaging with media, or navigating through Kodi. This alteration to Kodi’s second-level subsystem transforms Kodi into a comprehensive hub where users can not only enjoy their media content, but also stay informed about their smart home environment, ensuring a more integrated user experience.

Overall, the Skinning Engine Subsystem serves as the mediator between the smart home devices and the user within the Kodi environment, effectively managing, interpreting, and presenting the input from these devices.

Finally, the extension of the Metadata subsystem to accommodate smart home data notably enhances the functionality of the smart home device integration. Traditionally responsible for organizing and storing media-related information, the Metadata subsystem has been augmented to incorporate data pertinent to smart home devices and their statuses. This change allows users to access comprehensive information about their smart home ecosystem within Kodi. By integrating this additional information, the Metadata subsystem can further enhance the user experience by offering a comprehensive view of relevant smart home device data alongside the user's typical view of their media content.

With the expansion of Kodi’s functionality to integrate smart home devices into its ecosystem, several pivotal directories within its concrete architecture face potential changes and adaptations.

Inside the Metadata subsystem, directories related to 'infoscanner', 'tags', and 'dbwrappers' are affected. These directories manage metadata acquisition, processing, and organization within Kodi. Integration of smart home devices requires handling metadata specific to smart home content, potentially impacting these directories.

Regarding the Web Servers subsystem, the 'network' directory will be influenced, as it deals with networking functionalities related to web services and communication within Kodi. Integration with smart home devices necessitates changes in networking protocols and functionalities to accommodate communication amongst the smart home ecosystem.

Finally, directories like 'contrib', 'utils', and 'threads' in the Utilities subsystem could potentially be affected, as these directories provide foundational and supportive functionalities to the entire Kodi system. With the integration of smart home devices, there might be a need for additional utilities or changes in foundational functionalities to accommodate smart home interactions.

In summary, as smart home functionalities are integrated into Kodi’s established architecture, the prospect of adapting directories becomes apparent. The affected directories, spanning Content Management, Interfaces, and Utilities represent critical components involved in Kodi’s overall functionality, but changes to these directories are inevitable in the endeavor of seamlessly integrating smart home functionalities to Kodi while maintaining its core architecture’s integrity.

Potential Risks and Limitations

The integration of smart home functionality within Kodi’s ecosystem offers a myriad of conveniences and expanded capabilities. However, this convergence of Kodi’s entertainment hub and home devices creates a set of potential risks that necessitate careful consideration and preventative measures.

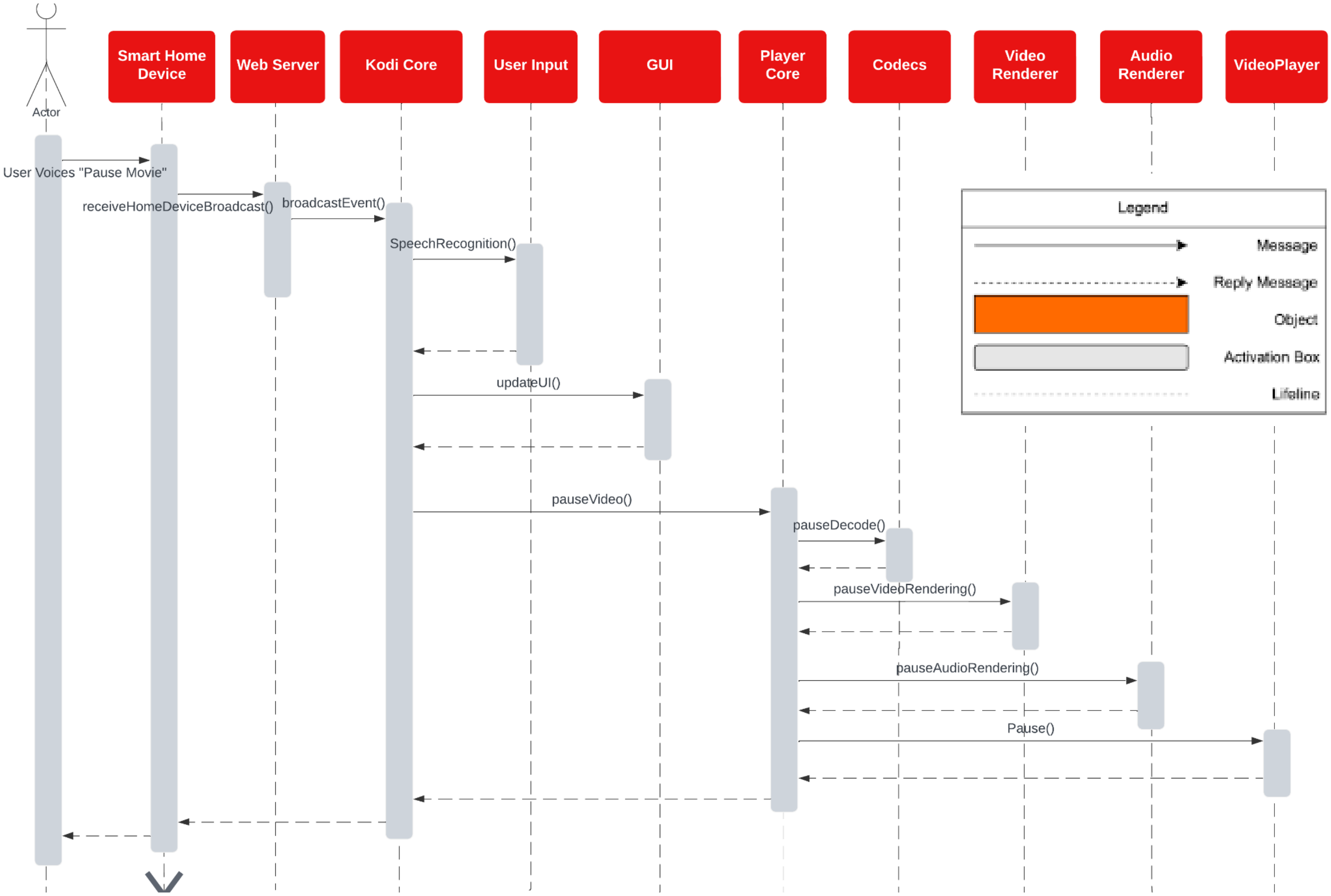
The integration of smart home devices into Kodi's framework introduces potential challenges in terms of maintaining compatibility and ongoing functionality across a range of smart devices. As new smart home devices and technologies that are constantly updated continuously emerge in the market, maintaining compatibility and ensuring seamless integration with Kodi becomes a perpetual chore. Changes or updates to the smart home device might necessitate frequent updates to Kodi to sustain interoperability. This ongoing need for updates and adaptations poses a threat to maintaining the functionality and compatibility of smart home features within Kodi, requiring consistent monitoring, testing, and adjustments to ensure a smooth and uninterrupted user experience across a wide array of smart home devices. Failure to keep up with these updates could result in device incompatibility or diminished functionality, impacting the ability of users to fully utilize the smart home integration within Kodi.

Moreover, the integration of smart home capabilities into Kodi's framework could potentially introduce performance issues, particularly in scenarios where the network experiences latency issues. As Kodi interacts with various smart home devices through the user's network, Kodi's performance might be susceptible to fluctuations, and delays in processing commands or receiving data from smart home devices could occur under these circumstances, impacting the responsiveness and overall user experience within Kodi.

Regarding security, another potential risk associated with the integration of smart home features into Kodi involves the privacy implications stemming from increased data sharing between devices. As Kodi interacts with various smart home devices, there is a heightened exchange of data, potentially including sensitive information about user preferences and usage patterns. This raises concerns about user privacy, as there is no guarantee that this information is being handled securely or appropriately. Furthermore, smart home devices can act as an entry point into Kodi, meaning vulnerabilities in the smart home software could result in access to Kodi. Measures should be put in place to ensure that devices connected are authorized and verified before having access to interact with Kodi beyond connecting.

Use Case and Sequence Diagram

The case shown below highlights how the Kodi system interacts with each component when the user pauses a movie using smart home voice controls. The user will voice the action "Pause Movie", which the smart home device will record and transfer to the Kodi web server component through the user's network. The web server will broadcast an event back to the Kodi core where it will broadcast an event for voice control input. The Skinning engine will register for this event as it contains the speech recognition component where the voice input will be converted to a command/event that Kodi Core can broadcast for the entire system. Kodi Core communicates with the Skins Engine, in order to update the UI to show that the video is paused. Finally, the Kodi Core will need to communicate with the Player Core component through an event to stop decoding, rendering and video playback.



Alternative Implementation

An alternative measure would be to implement a smart hub ecosystem, with this we make the smart home/voice control automation system much more accessible as it does not rely on the user having up to pair peripherals. It would allow the user to access Kodi smart home system without having technological barriers such as no smart tv plugin (Amazon Firestick), always-on mic to pick up ambient audio, and avoid having to go through a computer. This introduces a potential user base with less tech experience, that could use the optimized automated system without having to build Kodi and understand its intricacies which makes it much more appealing accessibility wise).

Stakeholders, NFRs, and Impact

It would function as a hardware deliverable unit that uses a simplified, optimized Kodi system to make automated video/audio calls such as “Hey Kodi, play *Insert Movie Name*” and this would search to find the optimal stream to match the user's needs. Our main push for this innovation is to satisfy the NFRs, that would make for a more accessible, operable and scalable version of the system. The device would have fast processing and response time with intelligent replies that are said in a timely manner (less than 3 sec on average) to ensure non-functional requirements for customers are satisfied. The device could connect to other smart home devices via a smart home ecosystem similar to the phillips hue devices; as opposed to the network via computer idea previously proposed. This would allow for low-power, low-bandwidth, but long range searching removing the network requirements from the first proposed implementation. The device could be plugged into any display devices,(i.e. tv, monitor) and have an integrated mic to pick up on ambient audio along with a speaker.

This simplified Kodi system would operate using low-level commands that could use automated responses to relay basic information (date/time), play music/audio or control connected peripherals (smart home devices). It would allow the user to use the streaming service for audio/video and control smart home devices but would be a lower-level system when interacting as opposed to the full system that can be accessed, which still could be accessed via an account online to add additional features. Users may have NFRs associated with how fast a media source can be found and selected in the automated process. Developers would have to focus on the automated calls to find the optimal stream, along with ensuring smart-home control/integration is seamless within the system, mainly focusing on performance and responsiveness. Software architects have to iron out the logistics of input handling and structuring of the simplified system that would be presented with the device along with how it would handle more advanced integration (add-ons); they mainly focus on satisfying deployability, efficiency and reliability.

Effects on the Conceptual Architectures, Directories and Files

For the integration of the Smart Home System, we first analyze the existing conceptual architecture to establish possible paths it would take throughout its lifetime. To process initial input/user input we make use of the speechRecognitions() method within the User Input subsystem of Skins Engine to process user audio inputs. We then make use of an additional subsystem in Interface called Automate Selection to automate stream client selection and this would depend on the Stream Client subsystem. We would add an additional subsystem to Player Core, called SmartHome Player to control the state of smart home devices. The response from these messages will tell our enhanced Audio Renderer, what the audio output should be. The subsystem is enhanced as additions on it are made for it to be able to comprehend the response from the system of the users audio input and output a respective response. It plays this audio through the smart device using PA Player.

Potential Risks

Integrating a smart home feature into Kodi, particularly aimed at less tech-savvy users, brings a unique set of challenges and risks. Simplifying the system to enhance user-friendliness might inadvertently reduce security measures. This reduction in security could create significant vulnerabilities, making the system an easier target for hackers. Such vulnerabilities could lead to unauthorized access to both smart home devices and personal data, posing a serious security risk.

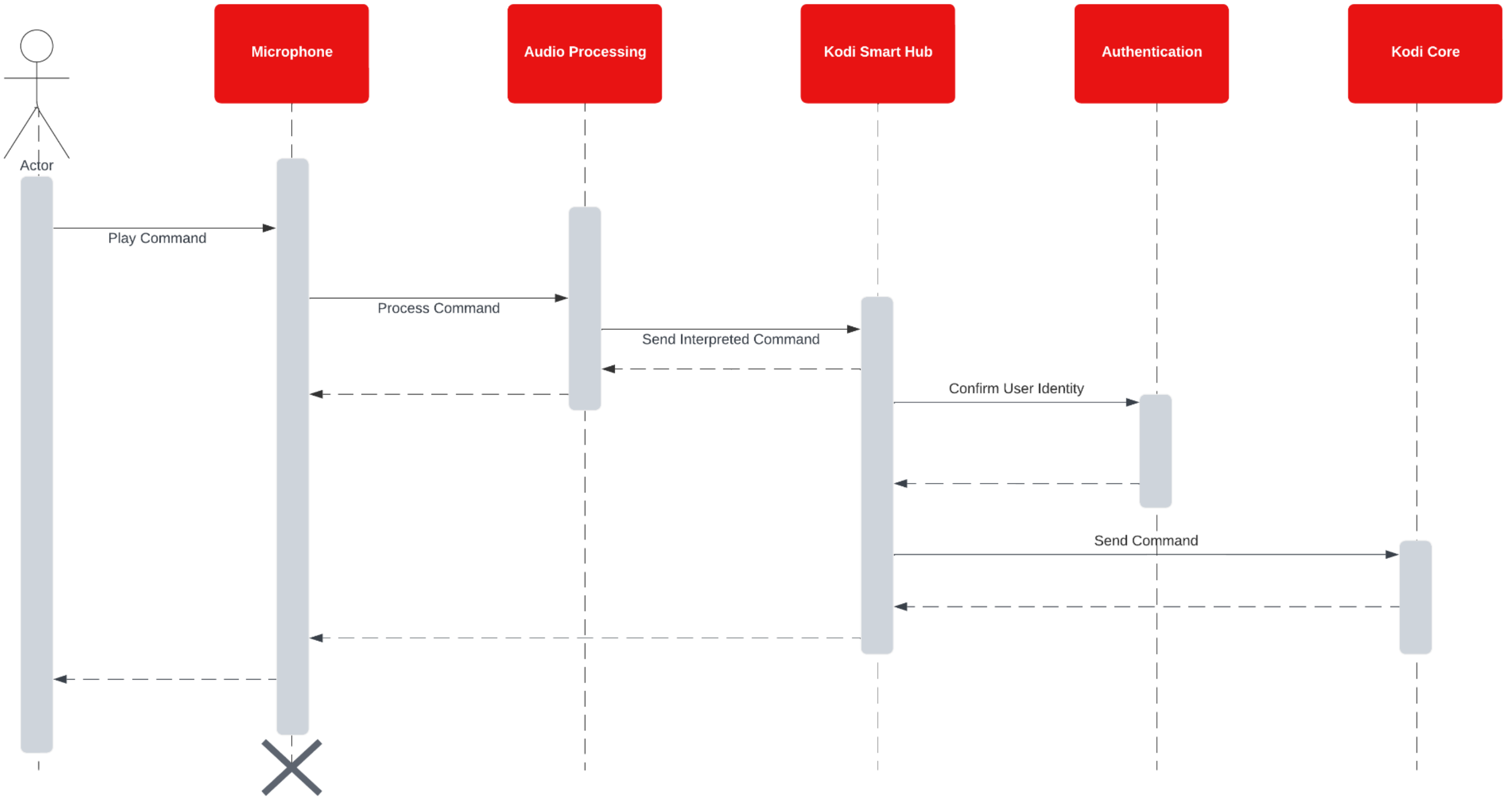
In addition, privacy concerns are raised due to features like integrated microphones and constant ambient audio monitoring. Users might be unaware of the extent of data collection, raising the risk that this data could be accessed by unauthorized parties or used inappropriately. In continuation, the system's reliability and performance are crucial. The ability to control various smart home devices and stream content efficiently depends heavily on the system's software and hardware capabilities. Any inadequacies in processing power or software glitches could lead to poor performance, leading to user frustration.

Compatibility with a wide range of smart home devices is another concern. Users may discover that some of their devices are not supported by the system, which limits its overall utility. Furthermore, targeting a less tech-savvy audience increases the risk of user error or misconfiguration, which might unintentionally compromise the system's security or functionality.

The system's heavy reliance on a stable internet connection adds another layer of complexity. Any disruption in internet service could affect both the streaming and smart home control features. Lastly, the development and maintenance of such a system, which is intended to be both simple for users and versatile in function, could be a costly and resource-intensive endeavor. This aspect must be carefully considered in the planning and ongoing support of the system.

Sequence Diagram

The sequence diagram above shows how the user would use voice commands to interact with Kodi through the Kodi Smart Hub. The audio processing and interpretation of commands is done through the Kodi Smart Hub rather than the main system. The Kodi Smart Hub then verifies the user before sending the command to the Kodi Core to broadcast the event.



Comparison of Implementations

	Main	Alternate
Advantages	- Does not require additional payment if the user already owns a smart home device - Easier to develop, sticks with current developer mindset - Users that already own a smart home device do not require any new knowledge to operate it	- More secure. Do not have to rely on security and trust of smart home companies. - Does not rely on user having up-to-par long range network to interact with devices - Does not require user to have ambient audio mic or sufficient speakers (contained within unit)
Disadvantages	- Performance can be limited by user's network speeds - Smart Home Device may not be compatible - Vulnerabilities on smart home devices could affect Kodi.	- Requires hardware alongside Kodi - Development of hardware does not align with open-source developer mentality - It may be pushing the wrong product to the wrong demographic (high-technical software for low-level customer) - Security risk associated with always-on mics, relies on a secure system that can not record inputs; just learn from them. (no chance for data leak)

The preferred implementation method is the network based approach. This not only aligns with Kodi's open-source developer mentality, but does not require any additional payment from users which allows the feature to be more accessible to users, especially those who already have smart home devices.

Conclusion and Lessons Learned

The integration of Smart Home Ecosystem functionality into Kodi will mark a visionary leap towards a more immersive and streamlined entertainment experience. Achieving this requires responsibility, such as the need to address challenges, privacy concerns, limitations, amongst many other aspects that we have highlighted over the course of this presentation. We learned over our research that Smart Home integration with other products already exists and contains a plentitude of documentation that can help us implement our enhancement. Lastly, it is important to recognize Kodi's rich repository of add-ons, as we discovered that some of our original ideas were already implemented and flourishing within the diverse and innovative landscape of existing add-ons. In summation, this report provides a foundational roadmap outlining the challenges and considerations necessary for a successful integration. It serves as a guide, emphasizing the need for meticulous planning, rigorous testing, and continuous adaptation to ensure a smooth integration of smart home functionalities within the Kodi framework.

Data dictionary

Implicit Invocation (Publish-Subscribe): An architectural style where the main idea is that a central system broadcasts events for individual components to register for and complete.

Smart Home Device: An electronic device, generally connected to other devices or networks via different wireless protocols that can operate to some extent interactively and autonomously.

Version Control: A class of systems responsible for managing changes to computer programs or other collections of information.

Naming Conventions

API: Application Programming Interface

GUI: Graphical User Interface

NFR: Non-Functional Requirement

UI: User Interface

References

Chawla, A., Levy, N., Ojeh, U., Secord, M., Sun, R., & Wolf, A. (2023, November 18). *The Concrete Architecture of Kodi*. Group 14: BIPimbap. https://aryanchawla1.github.io/cisc326/a2.html