

Technical Disclosure Commons

Defensive Publications Series

June 2021

SSH Connections In Chat Applications

Daniel Clarke

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Clarke, Daniel, "SSH Connections In Chat Applications", Technical Disclosure Commons, (June 16, 2021)
https://www.tdcommons.org/dpubs_series/4379



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

SSH Connections In Chat Applications

ABSTRACT

Many operating systems and web browsers provide a terminal application to execute SSH commands and interact with servers. SSH interaction has notable overhead, especially when the user needs to run a quick command or two. This disclosure describes techniques that add SSH connections as an additional class of communication to the unified communication mechanisms included within operating systems, web browsers, and applications such as chat and email.

KEYWORDS

- Secure Shell Protocol (SSH)
- Terminal
- Unified communication
- Chat application
- Messaging application
- Cloud server
- Internet of Things (IoT)

BACKGROUND

Increasingly, various computing applications operate within the cloud, requiring users to manage the growing volume and complexity of communication between cloud servers and applications and the lightweight clients used to interact with them. These interactions usually take place via the Secure Shell Protocol (SSH) that is commonly used to connect securely to remote servers. Many operating systems and web browsers provide a terminal application to execute SSH commands and interact with servers in the cloud.

Current mechanisms for SSH connections permit users to store some settings for various connections to reduce some of the burden and overhead of initiating the connection and interacting with the server. While the stored settings are useful, the SSH interaction still has notable overhead, especially when the user needs to run a quick command or two. Moreover, the interaction can still feel awkward, especially compared to other communication such as chat interactions with other people or bots via messaging or chat applications.

Many operating systems, web browsers, and applications include mechanisms that attempt to unify a user's communications with external entities. For example, tabs within email and chat applications enable users to unify their email and chat communications within a single application. Such chat tabs can include communication with humans (individuals/groups), bots (e.g., customer service), and computational representations of physical assets, such as meeting rooms. However, these mechanisms for unified communication do not currently support the inclusion of SSH communication with public or private cloud servers and other machines and devices in the network, such as Internet of Things (IoT).

DESCRIPTION

This disclosure describes the addition of SSH connections as an additional class of communication to the unified communication mechanisms included within operating systems, browsers, and applications, such as chat or email.

A user can add and configure an SSH connection with a server within the desired app similar to adding any other interacting entity such as people, bots, rooms, etc. The configuration can specify the same settings as would be specified within a dedicated SSH application or a terminal. The user's communication with a server via SSH can then be shown and managed in a tab within the application along with the user's other interactions.

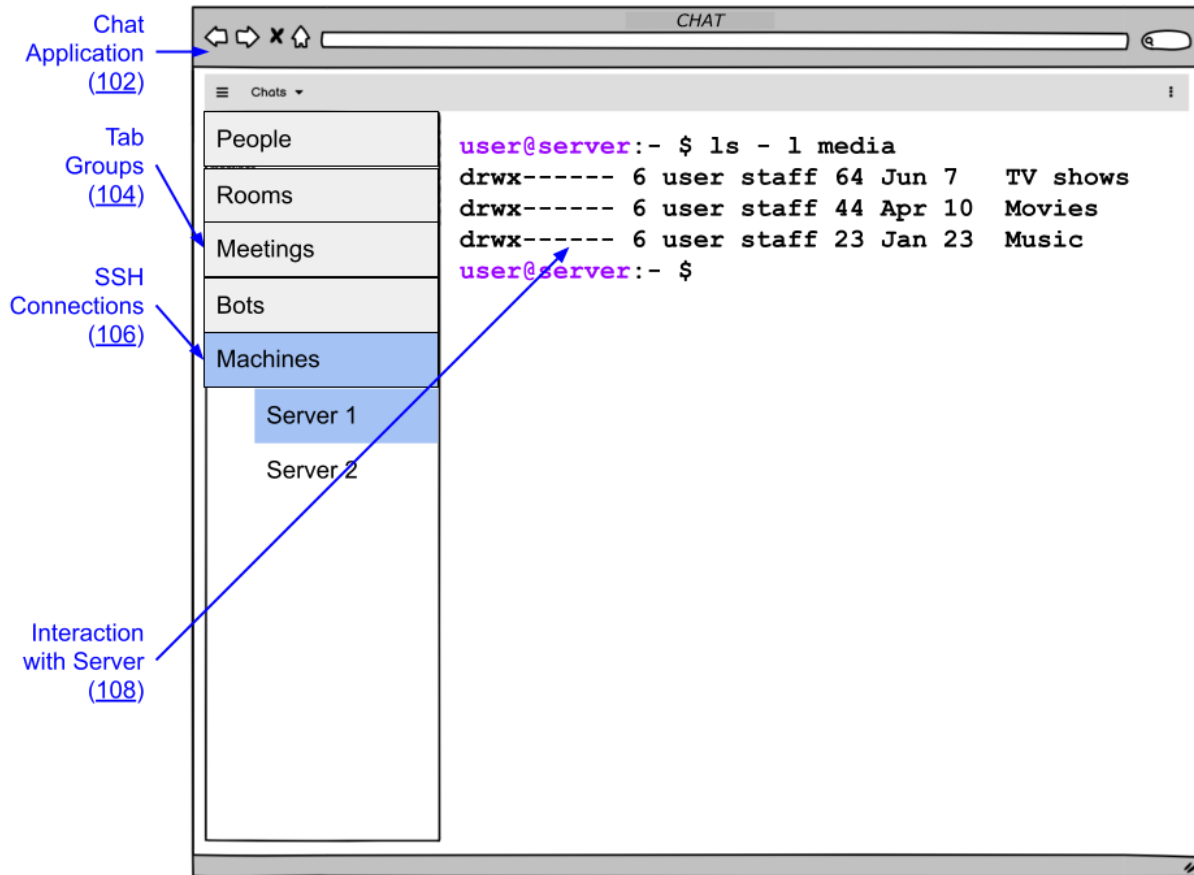


Fig. 1: SSH interaction with a server as a tab within a chat application

Fig. 1 shows an example of operational implementation of the techniques described in this disclosure. A user connects via SSH to a remote machine in the cloud (Server 1) within a chat application (102). For example, the chat application may be a standalone application or a browser-based application. SSH connections (106) are added as a separate group of tabs within the application, along with various other types of chat tabs (104), such as people, rooms, meetings, and bots. The user can then click the corresponding chat tab to interact with the server via SSH (108) by typing commands just as within a terminal. In Fig. 1, the user connects to a media server and obtains a list of the types of media stored on the server by executing a file listing command.

Such integrated applications can be particularly useful in a workplace setting in which engineers, developers, and other technical staff frequently use SSH to interact with machines and devices while also communicating with others such as peers, management, customers, etc. The integration also provides a unified communication mechanism for interacting with various servers and devices within a smart home.

The described techniques can be implemented within any operating system, browser, application, or platform that includes communication with other entities. The techniques can be used to communicate with any machine or device that supports interaction via SSH. Implementation of the techniques enables users to unify SSH interactions within their other communications, thus improving speed, intuitiveness, and productivity.

CONCLUSION

This disclosure describes techniques that add SSH connections as an additional class of communication to the unified communication mechanisms included within operating systems, web browsers, and applications. With integration of SSH connections, the user's communications with a server via SSH are shown and managed in a tab or other interface within the application along with the user's interactions with other entities such as people, bots, rooms, etc. Implementation of the techniques enables users to unify SSH interactions within their other communications, thus improving speed, intuitiveness, and productivity.