Technical Disclosure Commons

Defensive Publications Series

March 13, 2019

Integrated system for notes and lists

Amitabh Saikia

Integrated system for notes and lists

ABSTRACT

A flexible integrated system for management and maintenance of notes and lists is described. With user permission, notes and lists are stored by a central server, and can be updated using a variety of surfaces (e.g., virtual assistant) and providers (e.g., smartphone or desktop applications, browser-based interfaces, etc. A database coupled to the central server maintains a state of notes and lists associated with each user. Federated updates are utilized to keep the providers, surfaces, and the database in a synchronized state. A low-latency application program interface (API) is provided to enable two-way synchronization of lists and notes. Upon a change of state in the database, a notification that includes a token is provided through a publish/subscribe (pubsub) channel. The token can be utilized by the providers to access specific entries in the database. Updates at a provider's end are transmitted to the central server using a direct API call. A mapped identifier between a provider and the central server is utilized to keep database entries synchronized.

KEYWORDS

- pubsub
- publish/subscribe
- synchronization
- virtual assistant
- federated update
- note-taking

BACKGROUND

Notes and lists serve as an important reminder and productivity tool for users. With user permission, online services that provide note-taking/ list functionality store notes and list contents on a server. Users commonly use a variety of devices and interfaces to create, update, access, and delete such notes and lists. For example, a user may use online note-taking software, e.g., via a browser-based interface; a dedicated application (e.g., installed on a user device such as a smartphone or computer); a virtual assistant application; etc. For a quality user experience, it is important that users are able to access and update their notes and lists seamlessly and accurately irrespective of the device used.

DESCRIPTION

This disclosure describes an integrated system for management and maintenance of notes and lists. The system is flexible in terms of how notes and lists are stored, accessed, managed and synchronized. User are provided access to their notes and lists via interfaces provided by different providers (e.g., desktop or mobile software applications, browser-based applications or websites, etc.) and on surfaces (e.g., virtual assistants provided via a mobile device, on a home speaker, via a home appliance, via an in-car entertainment system, etc.). The interfaces enable users to create, access, update, and delete their notes and lists.

Fig. 1 depicts an integrated system (100) for management and maintenance of notes and lists.

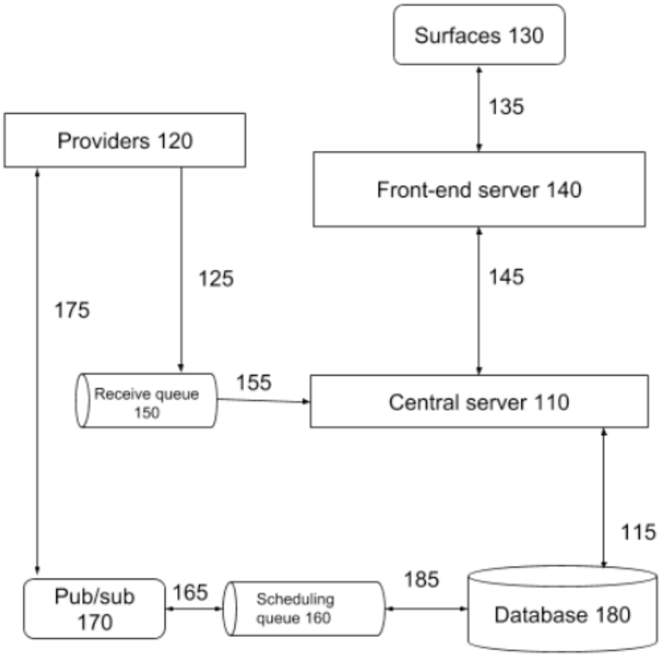

Fig. 1: Management of lists and notes across different providers and surfaces

Central server (110) is configured to receive requests for creation, access, updation and deletion of lists and notes from different providers (120) and/or surfaces (130). Database (180) is coupled (115) to the central server and maintains a state of each note and list associated with each user. The providers and surfaces can be the same as (or associated with) the entity operating the central server or can be third-parties. Federated updates are utilized to keep data stored by the providers, surfaces, and database in a synchronized state.

Access via a provider: Requests (125) for operating on a note or list by a user via a provider are routed (155) to the central server via a receive queue (150). Such routing enables reduction in perceived user perceived latency.

Access via a surface: Requests (135) for operating on a note or list by a user via a surface are routed (145) to the central server via a front-end server (140). The front-end server is configured to perform functions associated with particular surfaces. For example, the server can provide voice recognition functionality for a surface that permits a user to interact with notes using their voice, e.g., a home speaker, an in-car entertainment system, a virtual assistant on a smartphone or wearable device, etc.

A low-latency application program interface (API) is provided to enable two-way synchronization of updates of lists and notes between different surfaces/providers and the database. A scheduling queue (160) is provided outside a critical path (e.g., separate from paths that support user interaction). The scheduling queue synchronizes (185) with the database and is utilized (165) by a publish/subscribe channel (170) to synchronize state updates (175) with the providers.

Upon a change of state in the database, a notification that includes a token is provided through a pubsub channel, e.g., via a registered handler. The token can be utilized by the providers to access specific entries in the database. Updates at a provider's end are transmitted to the central server using a direct API call. A mapped identifier between a provider and the central server is utilized to keep database entries synchronized.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user's social network, social actions or activities, profession, a user's preferences, or a user's current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

A flexible integrated system for management and maintenance of notes and lists is described. With user permission, notes and lists are stored by a central server, and can be updated using a variety of surfaces (e.g., virtual assistant) and providers (e.g., smartphone or desktop applications, browser-based interfaces, etc. A database coupled to the central server maintains a state of notes and lists associated with each user. Federated updates are utilized to keep the providers, surfaces, and the database in a synchronized state. A low-latency application program interface (API) is provided to enable two-way synchronization of lists and notes. Upon a change of state in the database, a notification that includes a token is provided through a publish/subscribe (pubsub) channel. The token can be utilized by the providers to access specific entries in the database. Updates at a provider's end are transmitted to the central server using a direct API call. A mapped identifier between a provider and the central server is utilized to keep database entries synchronized.