CS 428 Freejournal Use Cases

daian2 — folayo2 — wzhang69

February 2015

Use Case Title

UC12: Validate data through direct API access

Use Case Brief

Actor

Third-party application developer using FreeJournal network

Brief

A third-party API developer or application user retreives documents from the FreeJournal network and validates and checks the timestamp, ranking of the document in the network, title, description, uploader integrity (digital signatures), and other document metadata programatically.

Casual Use Case

Actor

Third-party application developer using FreeJournal network

Scope

Subsystem: FreeJournal Back-End API

Level

Primary critical task

Story

A third-party API developer installs FreeJournal and its dependencies in developing their application. The developer or application user retreives documents from the FreeJournal network and validates and checks the timestamp, ranking of the document in the network, title, description, uploader integrity (digital signatures), and other document metadata programatically.

The developer bundles their applications with FreeJournal, either as an external dependency for the user to install or as a bundled library. End-users of the third party application then can perform tasks involving checking and browsing documents through this application.

Fully Dressed UseCase

Primary Actor

Third-party application developer looking to validate data stored on FreeJournal network

Goal in Context

Third-party developers can access and validate the integrity, timestamp, and rank of FreeJournal documents and related data in external applications using clear, well-defined APIs

Scope

Subsystem: FreeJournal Back-End API

Level

Primary critical task

Stakeholders and Interests

- Third-party developers: looking to increase the security of the document system
- Users: looking to gain assurance from outside developers of the integrity of the FreeJournal network and its data

Preconditions

- Developer installs the Journal library and dependencies
- Developer synchronizes their local FreeJournal installation with the network
- Developer properly configures FreeJournal to store keys, etc.

Minimum Guarantees

- All data returned by the FreeJournal API is valid FreeJournal network data

Success Guarantees

- Developer interacts with the FreeJournal network to retrieve and check the validity of documents
- Developer is able to bundle the FreeJournal API with their third-party application for end-users or provide installation instructions for FreeJournal as a dependency to users
- End uers are able to use third-party application to access FreeJournal network data

Trigger

- A third-party application user initializes the FreeJournal API to validate FreeJournal data

Main Success Scenario 12.1

1. Third-party developer or user makes API call to retreive FreeJournal document from the Storage Subsystem
2. System returns a FreeJournal document object with all known properties of the FreeJournal document
3. Third-party developer checks document timestamp
4. System returns last known timestamp and assurance level of document timestamp
5. Third-party developer makes request for rating of document
6. System returns rating of document by current FreeJournal users and document popularity
7. Third-party developer requests a vote in the document ranking

8. System returns all data necessary for third-party developer to vote on document
9. Third-party develper requests the creation of a trusted timestamp for the document
10. System returns all data necessary for creating a third-party timestamp on the Bitcoin network for the document

Note that this flow is also applicable to users of third-party applications, who can trigger all the scenarios of third-party application developers.

Extensions / Alternative Scenarios

12.2: Document is not found: system raises an exception for the third-party developer or application user
12.3: No timestamp is available for the document: system returns best guess with assurance level of 0
12.4: The API is unable to connect to other nodes in the FreeJournal network: the third-party application is told to try again later
12.5: The API is unable to access a needed dependency due to improper configuration: throw an exception prompting the installation of the appropriate dependency

Use Case Title

UC17: Vote on and rate documents

Use Case Brief

Actor

Document viewer browsing FreeJournal documents

Brief

A viewer of FreeJournal documents is able to vote on and rate documents that are interesting to them, both positively and negatively. Such viewers can also rank documents by the ratings of other users, and view ratings on all documents. The Bitcoin network is used for document ranking.

Casual Use Case

Actor

Document viewer browsing FreeJournal documents

Scope

System functionality

Level

Secondary critical task

Story

The viewers of FreeJournal view the documents by the leakers. The viewers can vote on the documents they are viewing by sending Bitcoin tokens. Users can vote both positively and negatively on the documents. A positive vote will increase document's rating and vice versa. Viewers can choose too rank the document by each document's overall rating. When viewer act to viewing the documents, developer should first check the document's raring information in order to determine the document's displaying rank. Ranking will be optional for users and will require a compatible Bitcoin client.

Fully Dressed UseCase

Primary Actor

Document viewer and power user contributing to overall network

Goal in Context

Users can vote on and both positively and negatively rank documents which they find to be quality, and view the ratings of other users on the same document. Bitcoin tokens are used to handle rating in a decentralized fashion

Scope

System functionality

Level

Secondary critical task

Stakeholders and Interests

- Document viewer: rate documents and rank them according to preference
- Document publisher: improve visibility of quality documents and incentivize quality publications
- Network operator: ensure documents being relayed and viewed are of high quality, mirror documents of high quality

Preconditions

- User installs the FreeJournal application or is using a gateway with the application installed
- User is viewing a document on the system
- User has a Bitcoin wallet separately installed, configured, and funded with a sufficient balance (around two US cents at the time of this document)

Minimal Guarantees

- All votes considered in ranking are real user votes
- No Bitcoin is lost to the network other than what is used to vote

Success Guarantees

- User's vote is registered in the system and affects ranking of other users on the network
- All votes considered in user rankings are authentic
- User is informed that their vote is registered

Trigger

- User selects to rate a document in the FreeJournal interface

Main Success Scenario 17.1

1. User selects a document to view, viewing the page in more detail
2. System displays a link to vote documents up or down
3. User clicks the link indicating their desired voting direction
4. System displays a prompt to the user asking the USD value of the tokens they would like to contribute to voting on the document
5. User enters the values requested by the system
6. System displays instructions for creating a Bitcoin transaction to timestamp the document
7. User uses compatible Bitcoin wallet to complete the transaction, registering their vote
8. System indicates to the user that vote has been registered with details of vote and receipt
9. User is able to sort other documents by rankings influenced by the vote, including the vote they recently submitted

Extensions / Alternative Scenarios

17.2 - User enters invalid value for token when prompted by system: user is prompted to reenter values
17.3 - User fails to complete the Bitcoin transaction as instructed by the system: system indicates to the user that votes have not been registered
17.4 - User is unable to connect to the Bitcoin network or is missing required prerequisites: system throws exception or instructs user to download prerequisites
17.5 - User decides to adjust vote amount or cancel voting altogether: system discards current voting process