

Feed Info

feed Id : Feed Id

feed Name : String?

indexable: Boolean

feed Certificate : Certificate

is Relayed: Boolean

last Version : Version?

Post Info

post Id : Post Id

feed Id : Feed Id

version : Version

post Name : String?

files: List<FileInfo>?

createdAt: DateTime?

bump Post InFeed: boolean?

post Info Signature: Signature

File Info

file Id : File Id

feed Id : Feed Id

post Id : Post Id

version : Version

fileName: byte[]?

type: FileUpdateType?

decryption Keys: List<DecryptionKey>?

file Data Signature: Signature

file Info Signature: Signature

Posts

since : Version?

to : Version?

feed Id : PublicKey?

post Id : Post Id?

posts: List<Post Info>

friendly Id?

signed by
source
or Trusted
Authority

encrypt? getHead
getTail

if no name file,
use title of index
file

use special name/
index file

UI-specific
really, not core

always set
for new Post

Default=false
for update

isFull Update: Boolean?

tags: List<String>
indexable: Boolean

Encrypt?
Delete all files

New/Full Update
Diff Update
Delete

encrypted
file name

always set
for new File

type: FileUpdateType?

decryption Keys: List<DecryptionKey>?

file Data Signature: Signature

file Info Signature: Signature

indexable: Boolean?

mime-type: String?

encrypt this
somehow

Запомни:
- ОБЕМ ХРАНИЛИЩА
• Б. Т. КАТОУ
- ТАРАС

Имена

feed private key, storage

popups/polls decryption keys

on the fly

Relay API

API hostname,
certificate,
etc

get Source Info(): Source Info

get Source Feed(): Feed Info

get Feed Info(feed Id): Feed Info

get Feed Updates(feed Id, since Version?)
: Posts

get Post Updates(feed Id, post Id, since Version?)
: Posts

getFile(feed Id, post Id, fileId, version?)

from Byte?, to Byte?)
+ getFileName
+ getDecryptionKey
+ getMimeType

Service API - Search

find Feed(query, in Name?, in Description?,
in Tags?)

find Post(query, feed Id?)

Service API - Authentication, Registration, User Management

Relay API - Relaying (pull preferred?)

- register "in" feed (pull feed from)
- register "out" feed (allowed to pull feed)
- subscribe for feed updates (websocket?)

Relay API - Feed Persistence

- push feed post for cloud persistence
- with decryption keys
- share post / add decryption keys

Decryption Key

decryption Key: byte[]

for: Public Key?

Важные модули:

- Post Creator (десктопное приложение JavaFX) - работа с локальными файловыми представлениями постов
 - создать новый пост из папки с файлами
 - создать обновление для существующего поста
- local Relay - REST API сервер
 - передает и принимает локалы
 - знает, какие локальные (момент передават), а какие внешние (момент приходит)
 - локальные - читает из локального файлового представления
 - удаленные
 - может проксировать Relay API
 - может такие материализовывать посты в локальное файловое представление
- Persistence API
расширенное API, подключающее управление материализацией.
- Post materialization - materialize Posts for Feed by default (yes/no/ask)
- De-materialize Post - force de-materialization, leave tombstone
- Materialize Post - force materialization (also materialize Files files for this post?)
- File materialization - materialize Files for Feed by default (yes/no/lazy/ask)
- De-materialize File - force de-materialization, leave tombstone
- Materialize File - force materialization
- Аутентификация
 - аутентификация сессии
 - Rate limit методами Proof Of Work - см. faroah
 - II Rate limit как Proof Of Work математически сложные методы

- Relay Viewer Web

- Это модуль, работающий как HTTP сервер и предоставляет Web UI.
- Работает как клиент для Relay API.
- Для локального Relay, управляет материализацией постов и взаимодействием с внешними Relays.
- Если Relay локальный, управляет локальной материализацией
- Этот уровень умеет декриптовать, знает о ключах (в т.ч. PKCS#11)

Дополнительные идеи:

- Post Creator Web

- Дополнение к Relay Viewer Web, позволяющее создавать посты и пушить их на Cloud Relay.

- Чат / Мессенджер

- Настройка, позволяющая собирать несколько потоков из взаимосвязанных чатов в чат / группу чатов.

Combined Feed View

Participants feeds

- Обучение

- Настройка, позволяющая запускать обучающие курсы

- Коллективы, лаймы

- Применение международного соглашения для доступа

- Запрос доступа к постам, файлам

- И пр., и пр.

Relay Viewer Web

- браузер постов - просмотр Markdown
 - MD индекс, тока головной страницы
 - специальные файлы name, index
 - ссылки внутри поста
 - ссылки на внешние посты
 - в своей ленте
 - в других лентах
 - гостевые и полные материализации файлов
(локальное сокращение криптованных файлов)
 - lazy
 - forced
 - скачивание декриптованных файлов / сохранение в папке
 - ! • открытие файлов внешними программами по HTTP
пользование

• Просмотр медиа-контента

1) Видео/аудиоплеер

- Противление видео/аудио
- Для видео
 - выбор субтитров
 - выбор аудиодорожки

- Для аудио

- слайд-шоу изображений

- Медиа

- пост как медиа

* возможность локальной монтируемости + персист

- мета-медиа, выложенный в себе медиа постов

Append-only
↓
локально монтируемые посты



• Просмотр документов

- pdf
- Office
- Read Era formats

- Get FeedUpdates - enne helay te nepemum, a nparupym, mo monem beprymus source info, omayga npemuna utipa
- Feed Info - post Id 0
- Don't make version UnixTime, that's a leak
- Folders in Post for Files, but also folders in Feed for Posts
 - File Folder (file manifest)
 - Post Folder (post manifest)
- Views
 - News View
 - Post
 - ↳ Folder
 - Tree View
 - Folder | Updated at
 - ↳ Post | Id | Created at | Updated at
- Local Feed Browser
 - No view
 - Files
 - Open external (file)
- Metadata
 - Decryption Key & Salt - file
 - Manifest
 - + CRC + fileId + fileName + folder + mimeType + updateType
 - + Data length / Padding

- Concatenate manifests and files
 - Avoid metadata leaks

- Binary diff / Stream diff format

- **Create Post** - from Folder

- **Update Post** - original Post Folder, updated Post Folder

- Post - Encrypted format

- Unencrypted format

- Не генерируются моменты, связанные с Апдейт / Бек

- Хранение кратких подтверждений

- Хранение списка подтверждений

- Кратки подтверждения

- Path на подтверждения

- Использование "Стандартное авторизованное приложение UserId" механизма

- Post

- /subscriber.keys

- /manifests

- /post

- /files

- subscriber.keys

- at least 1024 files (w/dummies)

- ↳ CRC / Algo / Key / Salt / File Id (encrypted)

- Post concatenated
Signature MF / Signature All / MFs / Files
- Ephemeral Post
 - Valid for (seconds)