SecHack365

自分の手元からプロトコルの壁を超えて、もっと自由につながる分散型SNS YojoHan

開発駆動コース仲山ゼミ 新矢将宗

実現したいもの

そのプラットフォーム、ずっとそこにあると思っていませんか？

テキストコミュニケーション・SNSは日常の記録や思考、思い出、人間関係の蓄積の場として非常に有用 × サービス終了やアカウント凍結など、ユーザーが関与できないままコンテンツが消失してしまうリスクがある

⇨ 自分のコンテンツは消えないように自分で持っていたい
SNSとして世界中の人とコミュニケーションしたい

分散型SNSはその実現手段の一つとなるが、制約も存在する

- 一般的なSNSと比べてユーザーの絶対数が少ない
- 様々なプロトコルによるネットワーク(連合)が存在し、それらを跨いだやりとりはごく限られる
- 個人でサーバーを立てるには必要なリソースがやや重たい

分散型SNSとは？

分散型SNSの特徴

一般的なSNS
- サーバーは企業などが管理
- 専用アプリやWebから利用

分散型SNS
- 誰でもサーバーを立ててその管理者になれる
- 末端ユーザーからの操作感は一般的なSNSとほぼ同じ

サーバー間でコンテンツをやり取りする際、各々が複製して保存する
⇩
自分のサーバーが存在する限りコンテンツは消えない

サーバ間のやりとりには秘密鍵を用いて署名を付加することでなりすましを防ぐ

分散型SNSにおけるプロトコル

分散型SNSには仕様が異なる様々なプロトコルが存在し、原則として同一プロトコル内でネットワークを構成する
YojoHanではActivityPubとNostrという2つのプロトコルを扱う

ActivityPub
Push方式, RestAPI, Http署名
Mastodon, Misskeyなどで使用

Nostr
Pull方式,WebSocket, Schnorr署名
Damusなどで使用

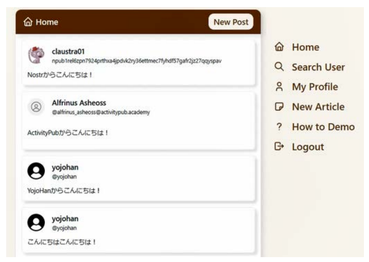
分散型SNSをベースに、より広く自由なコミュニケーションを

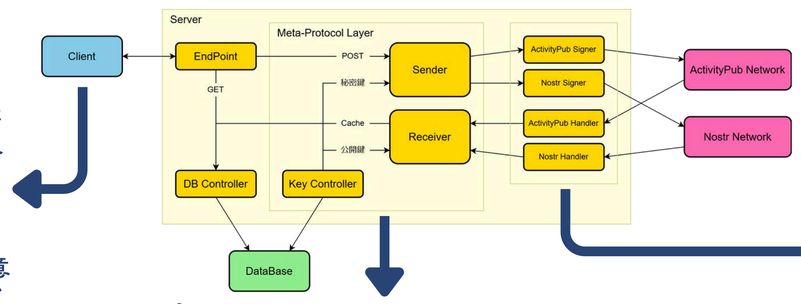
メタ・プロトコルレイヤーによる複数プロトコル対応

ユーザー体験の統一

- 対応プロトコルのコンテンツを一つのタイムラインで表示
- プロフィールやフォローの扱いも同一フローで共通化

⇩
ユーザーはプロトコルのことを意識せず、普通のSNSと同じ感覚で利用できる





メタ・プロトコルレイヤー

分散型SNSプロトコルを扱う時は必ずこのレイヤーを経由
- 署名・検証・連合とのやり取りを抽象化・並列化
- 各プロトコルの鍵はデータベースに保存
- 公開鍵や秘密鍵を各プロトコルの処理へ渡す
- 連合から受け取って共通化されたオブジェクトをデータベースにキャッシュ

⇩
エンドポイント側から複数のプロトコルをまとめて扱う

拡張性に優れた設計

- 追加実装・拡張が容易
  - 各プロトコル独自の部分をinterfaceに則って実装するだけで、既存のコードに手を加えず新たなプロトコルに対応
- プロトコル固有部分の並列化により、対応プロトコルが増えてもオーバーヘッドがほぼ一定

異なるプロトコルのオブジェクトを共通化

セキュリティへの貢献

秘密鍵の紛失リスク削減

プロトコルによっては秘密鍵そのものをユーザー自身が管理

秘密鍵が漏洩するとなりすましが可能

分散型SNS全般における鍵管理のデファクトスタンダードは未だ存在しない

YojoHanでは全てのプロトコルの秘密鍵をアプリ側で管理
ユーザーは普通のSNSと同じようにIDとパスワードのみで利用

➡ 秘密鍵の管理をユーザーから隠蔽
紛失のリスクを大きく減らす

YojoHanを介した連鎖的な本人証明

各プロトコルのIDをYojoHan内で紐づけ

一部のプロトコルで策定されている本人証明
例) Nostrのドメイン認証

➡ これらを組み合わせることで
連鎖的に、より強い本人証明が可能

短文SNSと長文Blogのゆるやかなつながり

SNS
さくっと書く
たくさん流れてくる
⇔
Blog
じっくり書く
一つずつ読む

ユースケースが異なる・同一UIでの共存は困難
どうにかして共存できないか？

SNSにおけるコミュニケーション手段「リプライ」
×
Blogにおけるコミュニケーション手段「コメント」

➡ 独立したBlogと分散型SNSを組み合わせた
ゆるやかなつながりを実装

YojoHanについて

個人での運用を意識

便利な機能群をあらかじめ同梱

自動定期バックアップ
クラウド/ローカル両対応

セットアップ・ログ監視
補助スクリプト

依存関係の最小化

- クリーンアーキテクチャによる依存性逆転
- 外部パッケージも最小限に
  - 根幹をなす暗号計算とSQLパッケージ以外ほぼ全てフルスクラッチで実装

➡ 外部パッケージアップデートに起因する
YojoHanのアップデート頻度を削減

YojoHanで使用している外部パッケージが
2024年の間にアップデートされた回数は全パッケージ合計で21回
一般的なWebアプリケーションと比較しても非常に少ない

実装で苦労したポイント

機能テストの外部への依存

機能を実装できているか否かは実際の連合で試すしかない
- ローカルやテストでは問題ないのに実環境では動かない
- あるインスタンスとの間では問題ないのに別のインスタンスとの間では動作しない
- テストデータが連合にずっと残ってしまうことも

言語仕様によるHash値の変化

JSONエンコード時、言語によってバックスラッシュやHTML特殊文字のエスケープの挙動が異なる場合がある

⇨ オブジェクトのHashや署名の値が変わってしまう

Http署名の未定義仕様

ActivityPubではRFC未定義のパラメータを使用
- Draftでは定義されている
- インスタンスによってあるパラメータがMUSTだったりそうでなかったり

そして、新しいテキストコミュニケーションの世界を見たい

SecHack365での一年

一人で25記事
アドベントカレンダー
を書く

他コースとの交流
知識と興味の広がり

習慣化が身に着いてきた

単方向のActivityPub
サーバーを実装

APIサーバーの
フレームワークを
フルスクラッチ実装

複数プロトコル
対応を目指した
技術検証・設計

統一された
タイムラインの
実装

ActivityPub,Nostrの
署名・検証を実装

より広いテキスト
コミュニケーション
対応への試み

YojoHanの先に
何があるのか
を考える

動機の芽生え
分散型SNSへの興味

Nostrなど様々な
プロトコルの存在を知る

Http署名の実装に苦戦
RFCや他の実装を読む

暗号理論やデジタル署名
について調べまわる

各種スクリプトの実装

SecHack365開始(6月) → 中間発表(11月) → 成果発表(2,3月)

NICT National Institute of Information and Communications Technology

"YojoHan"の名前の由来

手を伸ばせばその部屋の中のもの全てに手が届く、コンパクトだが必要なものは全て揃っているという意味を込めて、ミニマルな居住空間を意味する「四畳半」から取っています。

今後の展望

SecHack365でできたこと

既存の分散型SNSよりも広い範囲のテキストコミュニケーションへ2つのアプローチを示すことができた
- 「相手」を広げる（複数プロトコル対応）
- 「種類」を広げる（短文SNS×長文Blog）

⇩ より広く、自由に

これからやりたいこと

- メタ・プロトコルレイヤーの画像・リアクション対応
- 対応プロトコルの拡張
- リリースして自分以外の人に使ってもらう