

Misskey: 分散型SNSプラットフォームの詳細分析レポート

1. Misskeyとは: 分散型SNSプラットフォームの全貌

1.1 Misskeyの基本的な定義と概念

Misskey(ミスキー)は、分散型マイクロブログサービスを実現するためのオープンソースソフトウェアである¹。注意すべき点として、「Misskey」は厳密にはソフトウェア自体の名称であり、特定のサービス名を指すわけではない。しかし、実際には「Misskey」ソフトウェアを利用して構築されたサービス全体、あるいは個々のサービスを指して単に「Misskey」と呼称するユーザーも多い¹。

このプロジェクトは、日本の開発者であるsyuilo(しゅいろ)氏(本名: 篠田 英司氏)によって2014年に開発が開始された²。当初はテキスト掲示板がメインのサービスであったが、短文投稿と時系列タイムライン機能が追加されると人気が高まり、徐々にマイクロブログとしての機能開発が中心となった⁵。

「Misskey」という名称は、開発者のsyuilo氏が当時聴いていたアーティストMay'nの楽曲「Brain Diver」の歌詞の一節に由来している²。また、Misskeyには「藍(あい)」と呼ばれる制服姿の猫娘のマスコットキャラクターが存在する²。

1.2 分散型SNSとしてのMisskey: 中央集権型との違い

Misskeyの最も重要な特徴は、「分散型」であるという点にある。これは、X(旧Twitter)やFacebookといった「中央集権型」SNSとは根本的に異なる構造を持つことを意味する。中央集権型SNSでは、特定の企業が単一のプラットフォームを運営・管理し、すべてのユーザーデータやルールを一元的にコントロールする⁷。

対照的に、Misskeyは特定の単一企業によって運営されるのではなく、世界中に存在する多数の独立したサーバー(「インスタンス」と呼ばれる)によって構成されるネットワークである⁴。各インスタンスは、それぞれ異なる個人やグループによって運営され、独自のテーマ、利用規約、コミュニティ文化、モデレーションポリシーを持つことができる⁴。ユーザーは、既存の多種多様なインスタンスの中から自分の興味や価値観に合ったものを選択してアカウントを登録することも、あるいは技術的な知識があれば自身で新たにインスタンスを立ち上げて運営することも可能である²。

さらに、Misskeyは「ActivityPub」というW3Cによって標準化された通信プロトコルを採用している⁷。これにより、異なるMisskeyインスタンス間はもちろんのこと、同じくActivityPubに対応しているMastodonやPleromaといった他の分散型SNSプラットフォームのユーザーとも、相互にフォローしたり、投稿を送受信したりすることが可能となる(この相互接続の仕組みを「連合」または「Federation」と呼ぶ)⁷。これらの連合されたインスタンス群全体から成る広大なネットワークは、「Fediverse」(Federation + Universeの合成語)という俗称で呼ばれている⁷。

この分散型の構造は、ユーザーに対して大きな「選択の自由」をもたらす。中央集権型プラットフォームの画一的なルールや運営方針に縛られることなく、自分に合ったコミュニティや環境を選べる点は大きな利点である⁵。しかし、この自由は同時に「コミュニティの断片化」という側面も持つ。数多くの

インスタンスが存在する¹⁰ことで、ユーザーベースは必然的に分散される。ActivityPubによる連合があるとはいえ、インスタンスごとに異なるルールや文化⁴が存在するため、プラットフォーム全体としての統一感や、巨大な中央集権型SNSが持つような大規模な情報拡散力は限定的になる可能性がある。また、数あるインスタンスの中から自分に最適な場所を見つけ出すこと自体が、新規ユーザーにとって一つのハードルとなり得る¹³。このように、Misskeyの分散性は自由と引き換えに、ある種の複雑さや分断も内包していると言える。

2. Misskeyの主要機能：豊富なインタラクションとカスタマイズ性

Misskeyは、標準的なマイクロブログ機能に加え、独自の多彩な機能を提供することで、ユーザーに豊かなインタラクションと高度なカスタマイズ性を提供している。

2.1 ノートとリアクション：コミュニケーションの核

Misskeyにおけるユーザーの投稿は、「ノート(Note)」と呼ばれる⁸。一つのノートには最大で3000文字まで入力することが可能であり、これはX(旧Twitter)の基本的な140文字制限や、Blueskyの300文字制限と比較して大幅に長い⁸。これにより、より詳細な内容や長文の考察などを投稿しやすくなっている。

ノートには、テキストだけでなく、画像、動画、音声ファイルといった様々な種類のファイルを添付することができる⁴。特筆すべき点として、添付するファイルごとに個別に「閲覧注意(NSFW: Not Safe For Work)」設定を行うことが可能であり、デリケートな内容を含むコンテンツの共有に配慮されている³。

Misskeyのコミュニケーションにおける際立った特徴の一つが、「リアクション(Reaction)」機能である⁵。これは、他の多くのSNSで見られるような単純な「いいね(Like)」ボタンとは異なり、ノートに対して様々な絵文字(標準のUnicode絵文字に加えて、各インスタンスの管理者が独自に追加した「カスタム絵文字」)を使って反応を示すことができる機能である。ユーザーは、他者のノートに対して、単なる肯定や賛同だけでなく、笑い、驚き、悲しみといった多様な感情やニュアンスを、絵文字を通じて気軽に、かつ具体的に表現することが可能となる¹⁰。

カスタム絵文字は、インスタンスの管理者が自由に追加できる⁸だけでなく、他のインスタンスで使われている絵文字を自分のインスタンスにインポートすることもできる²³。これにより、各インスタンスやコミュニティ内で独自の視覚言語や内輪ネタ、ミームなどが生まれやすく、コミュニケーションをより豊かで楽しいものにする効果が期待できる³。この点は、テキストチャットツールであるDiscordやSlackのリアクション機能に近い感覚と言える⁷。

このリアクション機能は、コミュニケーションの質そのものを変化させる可能性を秘めている。単なる「いいね」の数を競うような指標中心の文化から、より多様で繊細な感情表現が重視される文化へとシフトを促すかもしれない。ただし、留意点として、Misskeyのリアクション機能は、連合先の他のプラットフォーム(特にMastodonなど)では完全にはサポートされていない場合がある。例えば、MastodonではMisskeyからのリアクションが単なる「お気に入り(いいね)」として扱われたり²⁴、カスタム絵文字が表示されなかったりする²⁶ことがあり、プラットフォーム間でのコミュニケーションにおいては、意図したニュアンスが完全には伝わらない可能性がある。

もちろん、他のユーザーのノートを自身のタイムラインで共有する「リノート(Renote)」機能や、コメン

トを付けて共有する「引用リノート」、ノートに返信して会話を続ける(スレッドを形成する)機能も標準で備わっている¹⁰。

2.2 ドライブ、チャンネル、ウィジェット: 高度な機能群

Misskeyは基本的な投稿・閲覧機能以外にも、ユーザーの利便性や表現力を高めるための高度な機能を多数搭載している。

- **ドライブ (Drive):** ユーザーがアップロードした画像や動画などのファイルを一元管理できる機能⁵。ファイルはフォルダに整理して保存でき、過去にアップロードしたファイルをノート作成時に簡単に再利用することが可能である。ただし、利用可能なストレージ容量は、所属するインスタンスの運営方針や、ユーザーに割り当てられたロール(役割、例: 新規ユーザー、支援者など)によって制限が設けられている場合がある²⁷。
- **チャンネル (Channels):** 特定のテーマや話題について集中的に会話を行うためのグループ機能¹²。ユーザーは既存のチャンネルに参加したり、新たに作成したりできる。チャンネル内でのノートは、そのチャンネルをフォローしているユーザーのタイムラインにのみ表示され、通常のローカルタイムラインなどには流れないため、特定の話題に関心のあるメンバー間でのクローズドな議論に適している。ただし、現状では一度作成したチャンネルを削除する機能は提供されていない点に注意が必要である³。
- **ウィジェット (Widgets):** Misskeyのユーザーインターフェース(UI)上のサイドバーなどに配置できる、小型の情報表示・操作コンポーネント¹⁹。例えば、プロフィール情報、サーバー情報、時計、カレンダー、RSSフィードリーダー、特定のユーザーの投稿を表示するものなど、様々なウィジェットが用意されている。さらに、Misskey独自のスクリプト言語である「AiScript」⁵を用いて、ユーザー自身がオリジナルのウィジェットを作成することも可能であり、UIのカスタマイズ性を大幅に高めている。

これら以外にも、Misskeyには以下のような多様な機能が実装されている¹⁹。

- **アンテナ (Antenna):** 設定したキーワードや条件に合致するノートを自動的に収集し、専用のタイムラインとして表示する機能。
- **クリップ (Clip):** 複数のノート(自分のものも他人のものも可)をまとめてコレクションとして保存・管理できる機能。
- **ページ (Page):** ノートとは別に、プロフィールページのような静的なコンテンツページを作成・公開できる機能。
- **アンケート (Poll):** ノートに選択式のアンケートを付加し、他のユーザーから簡単に意見を収集できる機能。

これらの豊富な機能群は、Misskeyを単なるマイクロブログツール以上の多機能プラットフォームたらしめている。特に、ドライブによるファイル管理の容易さ、チャンネルによるコミュニティ内コミュニケーションの深化、ウィジェットやAiScriptによる高度なカスタマイズ性は、特定のニーズを持つパワーユーザーにとって大きな魅力となるだろう²²。

しかしながら、この機能の豊富さは、同時にMisskeyの複雑さにも繋がっている。X(旧Twitter)のような比較的シンプルなプラットフォームに慣れたユーザーや、SNS初心者にとっては、これらの多様な機能を理解し、使いこなすまでにある程度の学習コストが必要となる可能性がある²⁹。多機能性はパワーユーザーを引きつける一方で、カジュアルユーザーにとっては参入障壁と感じられるかもしれない。プラットフォームの普及拡大においては、この複雑さをいかに緩和するかが一つの課題となり得る。

2.3 MFMとUIカスタマイズ: 表現の自由度

Misskeyは、テキスト表現とユーザーインターフェース(UI)の両面で高いカスタマイズ性を提供し、ユーザーの表現の自由度を高めている。

- **MFM (Markup language For Misskey):** ノート本文、プロフィール、チャットメッセージなど、Misskeyの様々な場所で利用できる独自のマークアップ言語である⁷。MFMを用いることで、簡単な記法を使ってテキストに様々な装飾を施すことができる。例えば、文字を回転させたり、拡大・縮小したり、色を付けたり、太字や斜体にしたり、背景色を設定したり、さらには点滅や揺れといったアニメーション効果を加えることも可能である⁷。これにより、ユーザーは感情や強調したい点をより豊かに、視覚的に表現することができる。
- **UIカスタマイズ:** ユーザーはMisskeyの見た目を自分の好みに合わせて自由に変更することができる⁷。システム全体の色調やデザインを変更する「テーマ」機能が提供されており、標準で用意されているテーマ(ライトモード、ダークモードなど)を選択するだけでなく、ユーザー自身がCSSなどを編集して独自のテーマを作成することも可能である。さらに、「デッキUI」¹⁹と呼ばれる表示モードを選択すると、画面上に複数の「カラム」(特定のタイムライン、通知、リストなどを表示する列)を自由に配置し、自分にとって最も効率的で情報密度の高い作業環境を構築することができる。

MFMによるテキスト表現の拡張と、テーマやデッキUIによる柔軟なインターフェースカスタマイズは、Misskeyがユーザーの個性や創造性を重視していることの表れであり、他のSNSプラットフォームにはない独自の魅力となっている。

2.4 タイムラインの種類とプライバシー設定

Misskeyは、ユーザーが情報をどのように受け取り、また自身の情報をどのように発信・管理するかについて、多様な選択肢を提供している。

- **タイムライン (Timelines):** Misskeyには、表示されるノートの内容が異なる複数のタイムラインが存在する⁷。主なタイムラインは以下の通りである。
 - **ホーム (Home / HTL):** 自分がフォローしているユーザーのノートが表示される、最も基本的なタイムライン⁴。
 - **ローカル (Local / LTL):** 自分がアカウントを所有しているインスタンス(サーバー)に所属する全ユーザーの公開ノートが、フォロー関係に関わらず時系列で表示されるタイムライン⁴。そのインスタンスの「広場」のような役割を果たす。
 - **ソーシャル (Social):** ホームタイムラインとローカルタイムラインのノートが混合して表示されるタイムライン¹⁰。API上では「hybridTimeline」と呼ばれる³¹。
 - **グローバル (Global / GTL):** 自身が所属するインスタンスが連合している(接続している)他のすべてのインスタンスの公開ノートが表示されるタイムライン¹⁰。Fediverse全体の動向を把握するのに役立つ。API上では「globalTimeline」と呼ばれる³¹。
 - **メディア (Media):** 所属インスタンス内で投稿された、画像や動画などのメディアファイルが添付されたノートのみを抽出して表示するタイムライン¹⁰。

これらの多様なタイムラインを使い分けることで、ユーザーはフォロー中心の情報収集から、所属コミュニティ内の話題、さらにはFediverse全体の広範な情報まで、目的に応じてアクセスすることが可能となる。

- **公開範囲設定 (Visibility Settings):** ノートを投稿する際に、そのノートを誰に見せるかを細かく設定できる⁷。主な公開範囲は以下の通りである。
 - **パブリック (Public):** すべてのユーザー（ログインしていないユーザーを含む場合もある）に公開され、ローカルタイムラインやグローバルタイムラインにも表示される。
 - **ホーム (Home):** ホームタイムラインにのみ公開される。フォロワーはもちろん、リノートなどを通じてフォロワー以外のユーザーも閲覧可能となる。
 - **フォロワー (Followers):** 自分のフォロワーにのみ公開される。フォロワー以外は閲覧できず、リノートも許可されない。
 - **ダイレクト (Direct):** 特定のユーザーを1名以上指定して送信する、いわゆるダイレクトメッセージ(DM)に相当する機能。指定されたユーザーにのみ通知が届き、内容もそのユーザーしか閲覧できない。
- **その他のプライバシー機能:** 上記以外にも、ユーザーのプライバシーを守るための機能が用意されている。
 - **フォロー承認制:** 自分をフォローしようとするユーザーのリクエストを承認するかどうかを選択できる機能¹²。
 - **ミュート・ブロック:** 特定のユーザーの投稿を自分のタイムラインに表示させない「ミュート」機能と、相手からのアクセスやインタラクションを拒否する「ブロック」機能がある¹⁹。Misskeyのブロックは主に相手から自分を見えなくする機能であり、一般的なSNSのブロック（相互に見えなくする）を再現するにはミュートとブロックの両方が必要になる場合がある³²。
 - **ワードミュート:** 特定のキーワードを含むノートをタイムライン上で非表示（最小化）にする機能¹⁹。
 - **CW (Content Warning / 内容警告):** ノートの内容を隠し、警告文を表示する機能。閲覧者はクリックすることで内容を確認できる¹⁰。ネタバレやデリケートな話題に用いられる。

これらのタイムラインとプライバシー設定の組み合わせにより、ユーザーは情報収集の方法をカスタマイズし、自身の情報発信における公開レベルを柔軟にコントロールすることが可能となっている。特に「ローカルタイムライン (LTL)」の存在は、Misskeyのコミュニティ文化形成において重要な役割を果たしていると考えられる。多くのユーザーがLTLを積極的に閲覧し、そこでの交流が活発に行われているインスタンスが多い¹²。LTLは、そのインスタンス独自の雰囲気や話題、暗黙のルールが醸成される場となりやすく、所属メンバー間の連帯感やコミュニティへの帰属意識を高める効果がある³。これは、フォロー関係に基づいたホームタイムライン中心のコミュニケーションとは異なる体験を提供する。一方で、LTL中心の文化は、インスタンス内部の交流を活発にする反面、インスタンス外部との交流が相対的に少なくなる傾向や、新規参加者がそのインスタンス特有の文化に馴染むまでに時間を要する可能性も示唆している。ユーザーは、LTLの特性を理解し、そのインスタンスの「空気感」を尊重することが求められる³。

3. Misskeyを支える技術：分散アーキテクチャとActivityPub

Misskeyの分散型という特性と豊富な機能は、特定の技術基盤によって支えられている。ここでは、その根幹を成すインスタンスの仕組み、連合を実現するActivityPubプロトコル、そして用いられている具体的な技術スタックについて解説する。

3.1 インスタンス(サーバー)の仕組みと役割

前述の通り、Misskeyのネットワークは多数の独立した「インスタンス」によって構成されている。インスタンスとは、Misskeyのソフトウェアがインストールされ、稼働している個々のWebサーバーのことを指す⁷。ユーザーは、いずれかのインスタンスを選んでアカウントを作成し、そのインスタンスを拠点としてMisskeyのサービスを利用することになる²。

各インスタンスは、それぞれ独立した運営主体(個人または組織)によって管理されている。そのため、インスタンスごとに利用規約、プライバシーポリシー、モデレーション(投稿内容の監視・管理)の基準、許容されるコンテンツの範囲(例えば、成人向けコンテンツの扱いなど)、サーバー独自のカスタム絵文字、ユーザーに割り当てられるドライブ容量などが異なる⁴。つまり、どのインスタンスを選択するかによって、ユーザーが経験するMisskeyの環境は大きく左右されることになる。

インスタンスを新たに立ち上げる運営者は、まずサーバーの物理的または仮想的なリソース(CPU性能、メモリ容量、ストレージ容量など)を確保する必要がある。Misskeyを安定稼働させるためには、ある程度のスペックが推奨されており、特にメモリ容量は重要視される傾向にある³⁴。次に、インスタンスにアクセスするためのドメイン名(例: misskey.example.com)を取得・設定し³⁵、Misskeyソフトウェアをインストール・設定する。ユーザーがアップロードする大量のファイルを効率的に管理するために、サーバー本体のストレージとは別に、AWS S3互換などのオブジェクトストレージサービスを利用したり³⁶、サーバー負荷軽減やセキュリティ向上のためにCloudflareなどのCDN(コンテンツデリバリーネットワーク)を導入したりすることも一般的である³⁵。これらのサーバー運用には、当然ながら継続的な費用(サーバーレンタル代、ドメイン維持費、ストレージ利用料など)が発生する¹³。

このように、インスタンスはMisskeyの分散ネットワークを構成する基本的な単位であり、それぞれが独立した運営と特性を持つことで、プラットフォーム全体の多様性を生み出している。

3.2 ActivityPubプロトコルによる連合(Federation)

個々に独立したインスタンス群が、あたかも一つの大きなネットワークのように相互に接続し、ユーザー間のコミュニケーションを可能にしているのが、「連合(Federation)」の仕組みであり、その技術的な基盤となっているのが「ActivityPub」プロトコルである。

ActivityPubは、W3C(World Wide Web Consortium)によって標準化された、ソーシャルネットワークのためのオープンな分散型プロトコルである¹⁷。MisskeyはこのActivityPubを実装しており²、これにより異なるMisskeyインスタンス間でのユーザーの相互フォローやノートの送受信が可能になっている。さらに、ActivityPubはMisskeyだけでなく、Mastodon⁷、Pleroma⁷、Pixelfed、PeerTubeなど、他の多くの分散型ソーシャルソフトウェアでも採用されているため、Misskeyユーザーはこれらの異なるプラットフォームのユーザーとも(理論上は)シームレスに交流することができる⁷。

技術的には、ActivityPubによる連合は、各インスタンス(サーバー)間で、決められた形式(Activity Streams 2.0形式のJSONデータ)の「アクティビティ」(投稿、フォロー、いいね、リノートなどの操作を表す情報)を、HTTPリクエスト(主にPOSTリクエスト)として送受信することによって実現される¹⁷。例えば、あるインスタンスAのユーザー(例: @alice@instance.a)がノートを投稿し、そのフォロワーが別のインスタンスB(例: @bob@instance.b)やインスタンスC(例: @carol@instance.c)にいる場合、インスタンスAはインスタンスBとインスタンスCに対して、そのノートの内容を含むアクティビティ情報をHTTPリクエストで送信する。この処理を「アクティビティの配送(delivery)」と呼ぶ¹⁷。受信側のインスタンスは、受け取ったアクティビティ情報を解釈し、自身のユーザー(この例では@bobや@carol)

のタイムラインなどに適切に表示する。

このActivityPubによる連合機能によって、Misskeyは単一のインスタンスに閉じることなく、より広範な「Fediverse」と呼ばれる分散型ソーシャルウェブの一部として機能することができる。

しかしながら、このActivityPubによる相互接続性は、理論上の「可能性」と実際のユーザー体験との間にギャップが存在する場合がある点に留意が必要である。プロトコルレベルでの互換性はあっても、各ソフトウェアやインスタンスが実装している機能には差異があるため、完全な相互運用性が常に保証されるわけではない。前述の通り、Misskey独自のリアクション機能やカスタム絵文字は、Mastodonなど他のプラットフォームでは意図通りに表示されなかったり、異なる形で解釈されたりすることがある²⁴。また、新しいアカウントや小規模なインスタンスが、他のインスタンスから発見され、スムーズに連合関係を築くまでには時間がかかる場合や、技術的な問題、あるいはインスタンス管理者の意向によって連合が制限されたり、特定のインスタンスとの接続が拒否されたりする可能性もある¹⁰。したがって、ActivityPubは相互接続のための強力な基盤を提供するものの、プラットフォーム間の機能差や運用ポリシーの違いが、実際のユーザー体験においては摩擦や不整合を生む可能性があることを認識しておく必要がある。

3.3 技術スタック概観

Misskeyの開発には、現代的なWeb技術が積極的に採用されている。その主要な技術スタックは以下の通りである⁵。

- プログラミング言語:
 - **TypeScript**: バックエンド(サーバーサイド)とフロントエンド(クライアントサイド)の両方で全面的に使用されている。JavaScriptに静的型付けを導入した言語であり、コードの安全性や保守性、開発効率の向上に寄与している⁵。
- サーバーサイド:
 - **Node.js**: JavaScriptを実行するためのサーバーサイドランタイム環境。非同期I/O処理に優れ、高いパフォーマンスが期待できる⁵。
 - **Fastify**: 高速性と低オーバーヘッドを特徴とするNode.js向けのWebフレームワーク。HTTPリクエストの処理などを担当する¹⁷。
 - **Nest.js**: TypeScriptベースのNode.jsサーバーサイドアプリケーションフレームワーク。モジュール性や拡張性に優れ、特にDependency Injection(DI)などの設計パターンを導入するために利用されている¹⁷。
 - **Bull**: Redisをバックエンドとして利用する、Node.js向けの高性能なジョブキューライブラリ。ActivityPubのアクティビティ配送など、時間のかかる処理やバックグラウンドで実行すべきタスクを管理するために使用される¹⁷。
- データベース:
 - **PostgreSQL**: 高機能で信頼性の高いオープンソースのリレーショナルデータベース管理システム(RDBMS)。複雑なデータ構造の扱いに長けており、Misskeyの主要なデータ(ユーザー情報、ノート、フォロー関係など)の永続化に使用されている⁵。
 - **Redis**: 高速なインメモリKey-Valueストア。主に揮発性の高いデータ(セッション情報、キャッシュなど)の保存、リアルタイム通知のためのPub/Sub機能、ジョブキュー(Bullと連携)の管理、プロセス間通信のハブとして利用され、PostgreSQLへの負荷を軽減する役割も担っている⁵。
- フロントエンド:

- **Vue.js:** ユーザーインターフェースを構築するためのプログレッシブなJavaScriptフレームワーク。Misskeyでは最新バージョンのVue 3が採用されており、Composition APIなどの機能により柔軟なコンポーネント設計が可能となっている⁵。
- **Vite:** 次世代フロントエンドツール。高速な開発サーバー起動とHot Module Replacement(HMR)、最適化された本番ビルドを提供し、開発体験の向上に貢献している¹⁷。
- 独自技術:
 - **MFM (Markup language For Misskey):** 前述の通り、Misskey独自のマークアップ言語⁵。
 - **AiScript:** Misskeyに組み込まれた独自のスクリプト言語。ウィジェットやプラグイン、ゲームなどの開発に利用でき、プラットフォームの拡張性を高めている⁵。

開発思想としては、特定のイデオロギー(例えば、反中央集権主義など)に強く基づくというよりは、技術的な探求心や面白さを重視し、最新のWeb技術を積極的に取り入れていく姿勢が見られる⁵。リード開発者のsyuilo氏は、古いブラウザでの互換性を犠牲にしても、最新技術の導入によるメリット(新機能の提供、パフォーマンス向上など)を優先する場合があると言及している¹⁷。

この技術的な先進性はMisskeyの魅力の一つであるが、一方で開発リソースとのバランスという課題も示唆される。Misskeyの開発は主に少数のボランティア開発者によって支えられており²、特にリード開発者は多忙により、プルリクエストのレビュー、ドキュメント整備、Issue対応などに十分な時間を割けない状況にあることが吐露されている⁴²。また、過去には大規模なリファクタリング作業が、その複雑さとリソース不足から中断された経緯もある⁴³。最新技術を積極的に採用するという野心的な姿勢と、限られた開発リソースとの間には潜在的な緊張関係が存在する。これが、新機能の実装速度、バグ修正、ドキュメントの充実度、長期的な安定性などに影響を与える可能性がある。オープンソースプロジェクトとして持続的に発展していくためには、より多くのコントリビューターを惹きつけ、開発プロセスを効率化し、あるいは開発資金を確保するといった方策が、今後の重要な課題となるであろう⁴²。

4. Misskey利用のメリット・デメリット分析

Misskeyは、その分散型の性質と豊富な機能により、多くの利点を提供する一方で、いくつかの欠点や注意点も存在する。利用を検討する際には、これらの両側面を理解しておくことが重要である。

4.1 利点 (Advantages)

- 高い自由度とカスタマイズ性: ユーザーは、MFMを用いた多彩なテキスト表現⁷、テーマやデッキUIによる柔軟なインターフェース設定⁷、そしてドライブ、チャンネル、ウィジェット、AiScriptといった豊富な機能群⁵を活用し、自身の好みや目的に合わせてプラットフォームを深くカスタマイズできる。
- コミュニティ選択の自由: 中央集権型SNSとは異なり、ユーザーは自身の興味関心、価値観、許容できるルールなどに合致したインスタンスを選択することができる⁴。これにより、より快適で帰属意識を持てるオンライン空間を見つけやすくなる。
- 表現の自由度(相対的): 特定の巨大企業による一元的なコンテンツポリシーやアルゴリズムによるフィルタリングが存在しないため、各インスタンスのルール範囲内であれば、中央集権型SNSと比較して自由な表現や議論が行いやすい環境がある(ただし、インスタンスごとの

ルールは遵守する必要がある)¹⁰。

- プライバシー保護の向上: ユーザーデータが単一のサーバーに集中せず、各インスタンスに分散して管理されるため、仮に一つのインスタンスがサイバー攻撃を受けても、被害がプラットフォーム全体に及ぶリスクは低い⁹。また、詳細なノートのパブリシティ設定¹⁰やフォロー承認制²⁸など、プライバシーコントロールのための機能も提供されている。ただし、ダイレクトメッセージの内容は受信側サーバーの管理者が閲覧可能である点には注意が必要¹⁰。
- 高い耐障害性: 分散型アーキテクチャにより、一部のインスタンスが技術的な問題やメンテナンスで一時的に停止したとしても、他のインスタンスは影響を受けずに稼働し続けることができる¹⁰。プラットフォーム全体が単一障害点を持たない構造になっている。
- オープンソースであること: MisskeyのソースコードはAGPLライセンスの下で公開されており⁵、誰でもその内容を確認できるため、ソフトウェアの透明性が高い。また、バグ修正や機能追加など、開発に直接貢献することも可能である²。
- 広告表示の抑制: サーバー管理者が広告を表示する機能自体は存在する¹⁹が、多くのインスタンス(特に個人運営のもの)では、中央集権型SNSに見られるような収益目的の強制的な広告表示は少ないか、全くない場合が多い。

4.2 欠点 (Disadvantages)

- ユーザー数の相対的な少なさ: X(旧Twitter)などの巨大プラットフォームと比較すると、Misskey全体のユーザー数はまだ少なく²⁹、特定の情報(特に最新ニュースなど)の入手や、広範な人々との交流においては限界がある可能性がある³。
- 機能の複雑さと学習コスト: 豊富な機能は利点である反面、特にSNS初心者にとっては操作が複雑に感じられたり、すべての機能を理解・活用するまでに時間がかかったりする可能性がある²⁹。
- インスタンスへの強い依存性: ユーザー体験は、選択したインスタンスの運営方針、ルール、コミュニティ文化、サーバーの安定性、提供されるカスタム絵文字の種類やドライブ容量などに大きく左右される³。
- 運営の不安定性と持続性の懸念: 多くのインスタンスは個人がボランティアで運営しており、運営者の個人的な事情(経済的理由、時間的制約、健康問題、興味の喪失など)により、予告なくサービスが停止したり、最悪の場合データが消失したりするリスクが存在する³。大企業が運営するサービスのような持続性の保証はない¹²。
- インスタンス選びの難易度: 多種多様なインスタンスが存在するため、自分にとって最適なインスタンスを見つけ出すこと自体が、特にMisskeyを初めて利用するユーザーにとっては最初のハードルとなり得る¹³。
- 鍵アカウント機能の挙動の違い: X(旧Twitter)のような単純な「鍵アカウント(非公開アカウント)」機能は存在しない。同様の状態を実現するには、ノートのデフォルト公開範囲を「フォロワー」に設定し、かつ「フォローを承認制にする」設定を有効にする必要があり、やや手順が煩雑である¹²。
- 連合(Federation)の不完全性: ActivityPubで他のプラットフォームと接続できるものの、Misskey独自の機能(リアクション、MFM、カスタム絵文字など)が連合先で完全には再現されなかったり、意図しない表示になったりするケースがある²⁴。
- モデレーション基準のばらつき: 各インスタンスが独自のルールとモデレーション基準で運営さ

れているため、プラットフォーム全体として一貫したコンテンツ管理が行われているわけではない。そのため、一部のインスタンスでは不適切なコンテンツが流通しやすい環境が存在する可能性も指摘されている⁹。

- 公式スマートフォンアプリの不在: Misskeyには公式のスマートフォンアプリが存在しないため、モバイル環境で利用するには、Webブラウザ経由でアクセスするか、サードパーティ製の非公式クライアントアプリ(例: MissRirica, MissCatなど⁸)を利用する必要がある⁸。
- 開発リソースの制約: プロジェクトが主にボランティアによって支えられているため、開発速度、バグ修正、ドキュメント整備などがリソース(特に主要開発者の時間)によって制限される可能性がある⁴²。
- サーバーリソースの要求: 比較的軽量の動作を目指してはいるものの、ある程度のユーザー数を抱えるインスタンスを安定稼働させるには、それなりのサーバーリソース(特にメモリ)が必要となる場合がある³⁵。
- プライバシーリスク(ダイレクトメッセージ): ダイレクトメッセージ(公開範囲「ダイレクト」のノート)の内容は、送信先ユーザーが所属するインスタンスの管理者であれば技術的に閲覧が可能であるため、機密性の高い情報を送信する際には相手とそのインスタンスの信頼性を考慮する必要がある¹⁰。

メリット・デメリット比較表

項目	メリット (Advantages)	デメリット (Disadvantages)
自由度・カスタマイズ性	高い(MFM, UI, 機能豊富) ⁷	機能が複雑で学習コストが高い可能性 ²⁹
コミュニティ	自分に合ったインスタンスを選べる ⁴	ユーザー数が相対的に少ない ²⁹ 、インスタンス選びが難しい ¹³
表現の自由	中央集権型より制約が少ない傾向(インスタンス依存) ¹⁰	モデレーション基準がインスタンス毎に異なる ⁹
プライバシー	分散による漏洩リスク低減 ⁹ 、詳細な公開設定 ¹⁰	DM内容は相手サーバー管理者が閲覧可能 ¹⁰ 、鍵垢設定が煩雑 ¹²
安定性・永続性	耐障害性が高い(分散型) ¹⁰	個人運営インスタンスが多く、閉鎖・データ消失リスク ³ 、永続性の保証なし ¹²
オープン性	オープンソースで透明性が高い ¹	開発リソースに制約 ⁴²
他プラットフォーム連携	ActivityPubで連合可能 ⁷	連合先での機能互換性が不完全 ²⁴
モバイル利用	サードパーティアプリが存在 ⁸	公式スマホアプリがない ⁸
広告	少ないか無い場合が多い	-
サーバー運用	-	ある程度のサーバーリソースが必要 ³⁵

5. 他のSNSプラットフォームとの比較: Misskeyの位置づけ

Misskeyの特徴をより深く理解するために、他の主要なSNSプラットフォーム、特に同じ分散型SNSであるMastodonと、中央集権型SNSの代表格であるX(旧Twitter)と比較する。

5.1 Misskey vs. Mastodon

MisskeyとMastodonは、共にオープンソースの分散型SNSソフトウェアであり、ActivityPubプロトコルを実装している点で共通している⁷。両者とも、ユーザーは独立したインスタンス(サーバー)に参加し、異なるインスタンスや対応プラットフォームのユーザーと相互に交流することが可能である。しかし、その起源、機能セット、開発思想、UI/UXには顕著な違いが見られる。

- 起源と開発思想: Misskeyは2014年に日本の開発者syuilo氏によって開発が開始され²、特定の思想(例えば反中央集権)に強く依拠するのではなく、技術的な面白さや機能性を追求する形で発展してきた⁵。一方、Mastodonは2016年にドイツのEugen Rochko氏によって開発され⁴⁵、Twitterのような中央集権型プラットフォームへの対抗軸として、より明確に非中央集権的な思想を背景に持つ傾向がある。
- 機能セット: Misskeyは、絵文字リアクション⁸、MFMによる高度なテキスト装飾⁷、ファイル管理機能「ドライブ」⁵、UI上に配置できる「ウィジェット」¹⁹、柔軟な「デッキUI」¹⁹、スクリプト言語「AiScript」⁵など、ユニークで豊富な独自機能を多数搭載している²²。これに対し、Mastodonは基本的なマイクロブログ機能(投稿、返信、ブースト=リノート、お気に入り=いいね)を中心とした、比較的シンプルで、X(旧Twitter)に近い機能セットを持つとされる⁴⁷。
- UI/UX: MisskeyのUIは、テーマやウィジェット、デッキUIなどにより非常に高いカスタマイズ性を持つが、その多機能性ゆえに、特に初心者にはやや複雑に感じられる可能性がある²⁹。MastodonのUIは、よりシンプルで直感的、X(旧Twitter)に慣れたユーザーにとっては馴染みやすいデザインとなっていることが多い⁴⁷。
- 機能互換性: ActivityPubで連合しているにも関わらず、機能差による互換性の問題が存在する。特に、Misskeyの主要機能であるカスタム絵文字やリアクションは、Mastodon側では正しく表示されなかったり²⁶、単なる「お気に入り(いいね)」として処理されたりする場合がある²⁴。
- スマートフォンアプリ: Mastodonには公式のスマートフォンアプリが存在するが、Misskeyには公式アプリはなく、サードパーティ製のアプリを利用する必要がある⁴³。
- 開発経緯: MisskeyはMastodonよりも開発開始年が早い(Misskey 2014年⁵、Mastodon 2016年)、分散型プロトコルであるActivityPubに対応したのはMastodon登場後の2018年である⁵。なお、MisskeyはMastodonのフォーク(派生プロジェクト)ではなく、完全に独立したプロジェクトとして開発されている⁵。

総じて、Misskeyは多機能性、カスタマイズ性、そして独自の文化(リアクション、MFMなど)を重視するユーザーに適している一方、MastodonはシンプルさやTwitterに近い使用感を求めるユーザー、あるいはより明確な非中央集権志向を持つユーザーにとって魅力的な選択肢となり得る。どちらもActivityPubで繋がっているため、ユーザーは必ずしもどちらか一方を選ぶ必要はなく、両プラットフォームのユーザーと交流することが可能である⁴⁴。

5.2 Misskey vs. X (旧Twitter)

MisskeyとX(旧Twitter)の比較は、分散型SNSと中央集権型SNSの根本的な違いを理解する上で重

要である。

- 基本構造: 最大の違いは、Misskeyが多数の独立したインスタンスから成る「分散型」であるのに対し、Xは単一の企業(X Corp.)が運営する「中央集権型」である点にある⁷。これにより、運営主体、サーバーアーキテクチャ、ルール策定プロセス、データの管理方法などが全く異なる。
- 機能面の比較:
 - 投稿文字数: Misskeyは最大3000文字⁸と長文投稿に対応しているが、Xの基本制限は140字(日本語等)である(有料プラン等で拡張可能)。
 - 反応機能: Misskeyは多様な絵文字による「リアクション」⁷を採用しているのに対し、Xは「いいね」が基本である。
 - カスタマイズ性: MisskeyはUIや機能のカスタマイズ性が非常に高い⁷が、Xのカスタマイズオプションは限定的である。
 - タイムライン: Misskeyはホーム、ローカル、グローバルなど複数のタイムラインを提供する¹⁰が、Xは主にフォロー中のアカウントの投稿を表示するタイムライン(「フォロー中」タブ)と、アルゴリズムによる推奨投稿を含むタイムライン(「おすすめ」タブ)で構成される。
 - APIアクセス: Xは近年API利用に対する制限を強化する傾向にあるが、Misskeyはオープンソースであり、APIも比較的自由に利用可能である(ただし、インスタンスごとの制限は存在し得る)。
- 連合(Federation): MisskeyはActivityPubを通じて他の分散型SNSと相互接続が可能⁷であるが、Xは他のプラットフォームとの連合機能を持たない独立した(閉じられた)ネットワークである²¹。
- ユーザー規模と情報の性質: Xは世界的に圧倒的なユーザー数を抱え、ニュース速報やトレンド情報の拡散力が非常に高い。一方、Misskey全体のユーザー規模は比較的小さく²⁹、情報の性質もインスタンス内のコミュニティ中心の話題や、特定の趣味に関する交流が多くなる傾向がある³。広範な情報収集や不特定多数への情報発信においてはXが有利だが、クローズドなコミュニティでの深い交流や、特定の興味関心に基づく情報交換にはMisskeyが適している場合がある。

これらの比較から浮かび上がるのは、MisskeyがX(旧Twitter)の単なる「代替品」ではないという点である。両者は提供する体験や価値が根本的に異なる。Misskeyは、中央集権的なプラットフォームの制約から離れ、より自由なカスタマイズ、プライバシーへの配慮、そして特定のコミュニティへの帰属を求めるユーザーにとって魅力的な「異なる選択肢」を提供する²¹。Xが持つ広範なリーチや情報の即時性とは異なる種類の価値、すなわち、ユーザー主導のコミュニティ形成や、よりパーソナルで多様なオンライン体験の可能性を提示していると言えるだろう。したがって、Xからの移行を検討するユーザーは、MisskeyがXと全く同じ機能や体験を提供するわけではないことを理解し、自身の求めるものがMisskeyの特性と合致するかどうかを判断する必要がある³。

主要SNSプラットフォーム比較表

特徴	Misskey	Mastodon	X (旧Twitter)
構造	分散型 ⁷	分散型 ⁴⁸	中央集権型 ⁷

プロトコル	ActivityPub ⁷	ActivityPub ⁹	独自
最大投稿文字数	3000字 ⁸	デフォルト500字(インスタンスにより変更可)	140字(日本語等、有料プラン等で拡張可)
リアクション機能	絵文字リアクション(カスタム可) ⁷	いいね(お気に入り)が基本 ⁴⁶	いいね
UIカスタマイズ性	高い(テーマ, デッキUI等) ⁷	中程度(テーマ変更等)	低い
公式スマホアプリ	なし ⁸	あり ⁴³	あり
インスタンス概念	あり ⁷	あり ⁴⁹	なし
開発元/主要国	日本 (syuilo氏) ²	ドイツ (Eugen Rochko氏) ⁴⁵	アメリカ (X Corp.)

6. Misskeyのはじめ方: インスタンス選びからアカウント作成まで

Misskeyを利用開始するには、まず拠点となるインスタンス(サーバー)を選び、そこにアカウントを作成する必要がある。

6.1 自分に合ったインスタンスの見つけ方: 基準と探し方

Misskeyの分散型という性質上、どのインスタンスに参加するかはユーザー体験に大きな影響を与えるため、慎重な選択が推奨される。インスタンスごとに、コミュニティの雰囲気、主な話題(テーマ)、利用規約、許容されるコンテンツ、管理者のモデレーション方針、サーバーの安定性、独自に導入されているカスタム絵文字の種類、割り当てられるドライブ容量などが異なるからである⁴。

インスタンスを探す主な方法は以下の通りである。

- **Misskey Hub**のサーバー一覧を利用する: Misskeyプロジェクトの公式サイトであるMisskey Hubには、Fediverse上で発見されたMisskeyサーバーの一覧が掲載されている⁷。ここで、現在新規登録を受け付けているサーバーなどを検索できる。ただし、このリストは自動的に生成・更新されるものであり、各サーバーの詳細な情報(規約、文化、特色など)はほとんど記載されていないため¹⁸、リストから興味を持ったインスタンスがあれば、実際にそのインスタンスのWebサイトを訪問して詳細を確認する必要がある。
- 他の**SNS**やコミュニティで情報を得る: 既にMisskeyを利用している友人・知人に尋ねたり、X(旧Twitter)などの他のSNSで「#Misskey」「#インスタンス紹介」といったハッシュタグで検索したり、関連するオンラインコミュニティで情報を集めたりするのも有効な手段である。
- 興味のあるキーワードで検索する: 自分の趣味や関心事(例:「ゲーム」「イラスト」「プログラミング」「TRPG」など)に関連するキーワードと「Misskey インスタンス」を組み合わせることで、特定のテーマに特化したインスタンスが見つかる可能性がある¹²。

インスタンスを選択する際には、以下のような基準を考慮すると良いだろう。

- **テーマ・トピック**: そのインスタンスが特定の趣味や話題に焦点を当てている場合、それが自分の興味と合致しているか¹⁸。
- **利用規約・ルール**: 各インスタンスが定めている利用規約、禁止事項、コンテンツに関するガイドライン(特にNSFWコンテンツの扱い³)などを事前に確認し、自分が同意でき、快適に利用

できる内容であるか¹⁸。

- コミュニティの規模と雰囲気: ユーザー数が多く活気のある大規模インスタンス(例: Misskey.io³)を好むか、あるいは小規模で落ち着いた雰囲気のインスタンスを好むか。ローカルタイムライン(LTL)を覗いてみて、そのインスタンスの「空気感」が自分に合いそうかを確認するのも有効である¹²。
- 言語: 主にコミュニケーションで使用されている言語が自分の使用言語と一致しているか。
- 登録の可否: 新規ユーザー登録が現在受け付けられているか。一部のインスタンスでは、サーバー負荷軽減やコミュニティ管理のために、登録が招待制になっている場合がある³。
- 運営者とサーバーの安定性: 運営者の情報が公開されているか、運営歴はどのくらいか、サーバーが安定して稼働しているかなどを、可能な範囲で確認する。個人運営サーバーには前述の通り不安定性のリスクが伴う¹³。

インスタンス選びは重要ではあるが、Misskey(およびActivityPub対応SNS)には、アカウント情報を他のインスタンスに移行する機能(フォロー・フォロワーリストのエクスポート/インポートなど)も存在する(ただし完全な移行ではない)。そのため、最初に選んだインスタンスが合わなかった場合でも、後から別のインスタンスに移ることは不可能ではない¹²。もし選択に迷う場合は、最大規模で多くのユーザーが集まる「Misskey.io」²にまず登録してみて、Misskeyの基本的な操作感や雰囲気を掴んでから、改めて自分に合ったインスタンスを探すというアプローチも考えられる¹⁰。

6.2 アカウント作成手順と初期設定

参加したいインスタンスが決まったら、次はアカウントを作成する。一般的な手順は以下の通りである。

1. インスタンスへのアクセス: 選択したインスタンスのWebサイトにWebブラウザでアクセスする。
2. 登録開始: トップページにある「登録」「アカウント作成」「このサーバーに登録する」といったボタンやリンクをクリックする¹⁰。
3. 情報入力: ユーザー名(他のユーザーと重複しない、半角英数字などが一般的)、パスワード、メールアドレスなどの必要情報を入力フォームに記入する⁸。ユーザー名は後から変更できない場合が多いので注意が必要。
4. 規約同意: インスタンスの利用規約やプライバシーポリシーなどを確認し、同意のチェックボックスにチェックを入れる。
5. メール認証: 登録したメールアドレス宛に確認メールが送信されるので、メールを開き、本文中に記載されている認証用リンクをクリック(またはURLをコピーしてブラウザで開く)する。これによりメールアドレスの有効性が確認され、アカウント作成が完了する¹⁰。通常、SMS(電話番号)による認証は要求されない¹⁰。

注意点: 一部のインスタンス(特に大規模なもの)では、スパムアカウント対策などの理由から、VPN接続や使い捨てメールアドレス(捨てメアド)を利用した登録を禁止している場合がある(例: Misskey.io⁵⁰)。

アカウント作成後、Misskeyを快適に利用するために、以下の初期設定を行っておくと良いだろう¹⁰。

- プロフィール設定: 自分のアカウントページに表示されるアイコン(アバター画像)、ヘッダー画像、自己紹介文などを設定する。
- プライバシー設定:
 - フォロー承認制: 自分をフォローする際に承認を必要とするか設定する²⁸。非公開アカ

ウントに近い運用をしたい場合に有効。

- デフォルトの公開範囲: 新規ノート作成時のデフォルトの公開範囲(パブリック、ホーム、フォロワー、ダイレクト)を設定する²⁸。
- 通知設定: どのような場合に通知を受け取るか(メンション、リアクション、フォローなど)を細かく設定する。
- テーマ設定: Misskeyの見た目を自分の好みのテーマ(ライト、ダークなど)に変更する⁷。

Misskeyでは、一人のユーザーが複数のアカウントを異なるインスタンス、あるいは同一インスタンス内で(許可されていれば)持つことも可能である⁵¹。

7. Misskeyの成り立ちと開発コミュニティ

Misskeyがどのようにして生まれ、現在どのように開発・維持されているのか、その歴史とコミュニティの側面を探る。

7.1 開発の歴史: syuilo氏による創設から現在まで

Misskeyの歴史は、2014年に当時高校生だった開発者syuilo氏(篠田英司氏²)によって、個人プロジェクトとしてスタートしたことに始まる²。当初は、リアルタイム性の高いテキスト掲示板サービスとして公開された⁵。その後、タイムライン機能が実装され、ユーザー間で短文を投稿し合うマイクロブログ的な利用が主流となっていた⁵。

開発初期のUIは、複数のウィンドウがタイル状に並ぶ独特なものであったが、2015年頃にはより現代的なWebインターフェースに近い形へと変化し、2018年頃には現在のMisskeyの原型とも言える洗練されたUIが実装された⁶。この間、技術スタックも大きく変遷しており、開発当初のPHP、MySQL、jQueryといった構成から、Node.js、PostgreSQL、Redis、そしてフロントエンドにVue.js、言語としてTypeScriptを採用する現在のモダンな構成へと移行している⁵。

Misskeyの歴史における大きな転換点は、2018年に分散型SNSプロトコルであるActivityPubに対応したことである²。これにより、Misskeyは単体のサービスから、他のサーバーやプラットフォームと相互接続可能な分散型ネットワーク(Fediverse)の一員となり、その知名度とユーザーベースを大きく拡大させるきっかけとなった⁵²。

2023年には、X(旧Twitter)の買収やそれに伴う一連の混乱などを背景として、代替SNSを求めるユーザーがMisskeyに流入し、ユーザー数が急増するという現象も見られた²。また、同年には最大規模のインスタンスであるMisskey.ioが、運営基盤の安定化などを目的として株式会社MisskeyHQとして法人化されている²。

このように、Misskeyは一人の開発者の手によって産声を上げ、技術の進化や外部環境の変化に対応しながら、独自の機能と分散型アーキテクチャを持つユニークなプラットフォームへと成長を続けてきた。

7.2 オープンソースコミュニティと貢献方法

Misskeyは、AGPL 3.0というライセンスの下で公開されているオープンソースソフトウェア(OSS)である²。そのソースコードはすべてGitHub上のリポジトリで公開されており²、誰でも自由に閲覧、利用、改変、再配布することが可能である。

開発は、創設者であるsyuilo氏が中心となっているものの、現在では彼一人だけではなく、世界中の複数のコントリビューター(貢献者)が参加するコミュニティによって支えられている²。これらの貢献

者の多くはボランティアであり⁵、Misskeyというプロジェクトへの共感や技術的興味に基づいて開発に関わっている。

Misskeyプロジェクトへの貢献方法は多岐にわたる⁵。

- **コードによる貢献:** 新機能の開発、既存機能の改善、バグの修正など、TypeScriptを用いたプログラミングによって直接ソースコードに貢献する。プルリクエストはGitHubを通じて行われる²。貢献内容によっては報奨金が支払われる制度も存在する²。
- **議論への参加:** 新機能の提案、既存機能に関する意見交換、バグ報告などを、GitHubのIssueやDiscussionといった場で行う。
- **翻訳:** Misskeyのインターフェースやドキュメントを、日本語から他の言語へ翻訳する。翻訳作業はCrowdinというプラットフォームを利用して共同で行われている⁵。
- **ドキュメント作成・整備:** ユーザー向け、サーバー管理者向け、開発者向けのドキュメントを作成・更新し、情報のアクセス性を向上させる。
- **普及活動:** Misskeyの魅力を広め、新たなユーザー(ミスキストと呼ばれる)を増やす活動を行う。
- **寄付:** Misskeyプロジェクト本体や、個々のインスタンス運営者に対して、金銭的な支援を行う。これは、サーバー維持費や開発者の活動を支える上で重要な要素となる⁵。

このように、Misskeyはオープンソースコミュニティの力によって開発が進められている。このコミュニティ主導の開発モデルは、多様なアイデアやスキルを結集させ、プロジェクトに活気をもたらす原動力となっている。例えば、多言語対応などは、世界中の有志による翻訳協力によって目覚ましいスピードで進展している⁵。

しかし、このモデルには課題も存在する。開発リソースが主にボランティアの善意と時間に依存するため、貢献には波があり、プロジェクト全体の進行速度や安定性が、コミュニティの活性度や、特に中心的な開発者のキャパシティに大きく左右される側面がある⁵。実際に、リード開発者の多忙さがボトルネックとなり、プルリクエストのレビュー遅延やドキュメント整備の停滞といった問題も指摘されている⁴²。また、開発コミュニティ内でのコミュニケーションにおいて、言語の壁が障壁となるケースもある(後述するFoundKeyフォークの動機の一つ)²。Misskeyが今後も持続的に発展していくためには、貢献者が参加しやすい環境を整備し、開発プロセスを効率化するとともに、主要開発者の負担を軽減するような仕組み(例えば、より多くのメンテナの育成、あるいは安定した資金源の確保など)を構築していくことが、重要な課題となるだろう。

7.3 主要なフォークプロジェクト

Misskeyはオープンソースであるため、そのソースコードを基にして、独自の変更や機能追加を行った派生プロジェクト、すなわち「フォーク(Fork)」が多数存在する²。これらは、Misskey本体(本家)の開発方針とは異なる方向性を目指したり、特定の機能に特化したり、あるいは本家が抱える課題を解決しようとしたりする試みとして生まれている。主要なフォークプロジェクトとしては、以下のようなものが挙げられる。

- **Firefish (旧称: Calckey):** 2022年から開発されていたMisskeyフォーク。特にMastodonとの互換性を強化しており、一部のMastodon用クライアントがそのまま利用できるなどの特徴があった。Misskeyの「ノート」を「ポスト」と呼ぶなど、用語の変更も見られた。しかし、2024年9月にメンテナンスモードに入り、同年年末をもってサポートを終了することが発表されている²。
- **FoundKey:** ドイツの開発者Johann150氏によって開発されているフォーク。同氏はかつてMisskey本家の開発に参加していたが、主に日本語で議論が進むことによる言語の壁や、そ

の他の技術的な理由からフォーク開発を開始したとされる。Misskey Pagesの簡素化や広告機能の削除などの改善が行われている²。

- **Sharkey:** TransFem.orgによって2023年から開発されているフォーク。投稿の編集機能やMastodon APIへの対応などが特徴として挙げられている²。
- その他のフォーク: 過去には、Misskeyから機能を大幅に削減した軽量版とも言える「Dolphin」というソフトウェアも開発されたことがある⁶。また、本家Misskeyのコードベースに対して大規模なリファクタリング(コード整理・改善)を試みるフォークプロジェクトも存在する(ただし、開発者がモチベーション低下により中断している例もある⁴⁰)。

これらのフォークプロジェクトの存在は、Misskeyのソースコードがオープンであることの証左であり、Misskeyエコシステムの多様性と活力を示している。同時に、本家とは異なるニーズや開発思想が存在することも示唆しており、ユーザーにとっては、本家Misskeyだけでなく、これらのフォークから自分に合ったものを選択するという可能性も開かれている。

8. インスタンスごとの多様性: ルール、文化、特色

Misskeyの分散型という特性は、プラットフォーム全体に画一的なルールや文化が存在するのではなく、個々のインスタンス(サーバー)がそれぞれ独自の個性を持つことを意味する。

8.1 各インスタンスの独自性とコミュニティ形成

Misskeyのインスタンスは、単に技術的に分離されたサーバーというだけでなく、それぞれが独自の社会的なコミュニティを形成する場となっている⁴。インスタンスごとに、そこで主に交わされる話題のテーマ(例えば、特定のゲーム、アニメ、音楽ジャンル、学術分野、地域情報、あるいはTRPG¹²など)、集まっているユーザー層、主に使用される言語、そしてコミュニティ全体の雰囲気(例えば、非常に活発で賑やか³、落ち着いていて穏やか、専門的な議論が中心、あるいは非常にオープンで誰でも歓迎する雰囲気¹²など)が大きく異なる⁴。

特に、そのインスタンスに所属するユーザーの公開投稿がすべて流れてくる「ローカルタイムライン(LTL)」は、そのインスタンスの個性や文化、いわば「空気感」を最も強く反映する場所となりやすい³。LTLでの交流を通じて、ユーザーはそのインスタンスへの帰属意識を高め、独自のコミュニティ規範や暗黙のルール、共通のミームなどが形成されていく。

8.2 利用規約、モデレーション、テーマの重要性

このようにインスタンスごとに独自性が生まれる背景には、各インスタンスが独自の運営方針を持っていることがある。

- **利用規約 (Terms of Service):** すべてのインスタンスは、そのサーバーを利用する上でのルールを定めた独自の利用規約を持っている⁷。これには、禁止される行為(嫌がらせ、スパム、違法コンテンツの投稿など)、コンテンツに関するガイドライン(特に成人向けコンテンツ(NSFW)の扱いに関する規定はインスタンスによって大きく異なる³)、著作権に関する注意などが含まれる。ユーザーは、アカウントを作成する前に、そのインスタンスの利用規約をよく読み、内容に同意する必要がある¹⁸。
- **モデレーション (Moderation):** サーバーの管理者は、利用規約に基づいてインスタンス内のコンテンツやユーザーの行動を監視し、必要に応じて注意喚起、投稿削除、アカウント停止などの措置(モデレーション)を行う³⁸。モデレーションの基準や厳しさ、対応の迅速さなどもイン

スタンスによって異なる。

- テーマ・特色: 前述の通り、多くのインスタンスは特定のテーマや興味関心を持つユーザーが集まることを意図して設立・運営されている⁴。これにより、ユーザーは自分の趣味や専門分野について深く語り合えるコミュニティを見つけやすくなる。

したがって、Misskeyを利用する際には、「郷に入っては郷に従え」という考え方が重要になる³³。自分が所属するインスタンスの利用規約を遵守することはもちろん、そのコミュニティで形成されている独自の文化や暗黙のルールを尊重し、他のユーザーに配慮したコミュニケーションを心がけることが、トラブルを避け、快適なMisskeyライフを送るための鍵となる。

このインスタンスの多様性は、ユーザーにとって大きな魅力である。自分の興味や価値観にぴったり合った「居場所」を見つけることができれば、非常に満足度の高いオンライン体験が得られるだろう⁵。しかし、この多様性は同時に、ある種の「排他性」を生み出す可能性もはらんでいる。特定のテーマに特化したインスタンスでは、そのテーマと無関係な話題をLTLに投稿することが暗黙のうちに避けられるかもしれない³。また、強く結束した内部文化は、新規参入者や他のインスタンスのユーザーにとっては、理解しにくく、壁のように感じられることもあるかもしれない¹²。さらに、あるインスタンスでは許容される表現が、別のインスタンスでは規約違反とされるなど、ルールやモデレーション基準の違いが、連合を通じたインスタンス間の交流において摩擦を生む可能性もある¹²。このように、インスタンスの多様性が生み出す「コミュニティごとの快適な居場所」は、裏を返せば「コミュニティ間の壁」や「エコーチェンバー現象」を助長するリスクも伴う。Fediverse全体の健全な発展のためには、ユーザーが自身の所属インスタンスの文化を理解すると同時に、他のインスタンスの多様性を認め、尊重する姿勢が求められると言えるだろう。

9. 結論と展望: Misskeyの可能性と今後の課題

本レポートでは、分散型SNSプラットフォームMisskeyについて、その定義、機能、技術、利点・欠点、他SNSとの比較、利用方法、歴史、コミュニティ、インスタンスの多様性といった多角的な側面から分析を行った。最後に、Misskeyの独自性と魅力、そして今後の可能性と課題についてまとめる。

9.1 Misskeyの独自性と魅力の再確認

Misskeyは、日本発のユニークな分散型SNSプラットフォームとして、以下のような際立った特徴と魅力を持っている。

- 豊富な独自機能: 標準的なマイクロブログ機能に加え、絵文字リアクション、MFMによる高度なテキスト装飾、ドライブ、チャンネル、ウィジェット、AiScriptなど、他のSNSには見られない多彩でユニークな機能を提供している。
- 高いカスタマイズ性: UIのテーマ変更、デッキUIによるカラム配置、ウィジェットの追加など、ユーザーが自身の好みや使い方に合わせてインターフェースを柔軟にカスタマイズできる。
- 分散型とオープンソース: 特定企業に依存しない分散型アーキテクチャと、ソースコードが公開されたオープンソースソフトウェアであることが、透明性、耐障害性、プライバシー保護の向上に寄与している。
- コミュニティ選択の自由: ユーザーは、数多くの独立したインスタンスの中から、自身の興味や価値観に合ったコミュニティを選択できる。
- 独自の開発文化: 特定の思想に強く縛られることなく、技術的な探求心や遊び心を重視した開発が進められている⁵。

これらの要素が組み合わさることで、Misskeyは画一的な中央集権型SNSとは一線を画す、多様で

自由度の高い、ユーザー主導のコミュニケーション空間を提供している。

9.2 分散型SNSとしての将来性と課題

近年、中央集権型SNSプラットフォームにおけるプライバシー懸念、検閲リスク、運営方針の突然の変更などに対するユーザーの不満や不安が高まる中で、Misskeyのような分散型SNSへの注目が集まっている⁷。Misskeyは、プライバシーを重視するユーザーや、特定のテーマについて深く交流したいコミュニティ志向のユーザーにとって、有力な選択肢となり得る。将来的には、単なるSNSの枠を超え、よりオープンで自由な情報発信・交流のための社会的な「インフラ」として機能する可能性も秘めている⁵³。

しかし、Misskeyがそのポテンシャルを最大限に発揮し、持続的に発展していくためには、いくつかの課題を克服する必要がある。

- スケーラビリティと安定性の確保: ユーザー数が増加した場合のサーバー負荷への対応³や、特に個人運営インスタンスにおける安定した運営基盤の確保、資金調達、データ消失リスクへの対策などが求められる³。
- ユーザー体験(UX)の向上: 多機能性ゆえの複雑さを緩和し、初心者でも容易に利用開始できるようなガイドやUI/UXの改善²⁹。インスタンス間での体験のばらつきをある程度吸収する仕組み。そして、依然として公式のスマートフォンアプリが存在しない点⁸も、利便性の観点からは課題である。
- 開発リソースの確保と継続性: プロジェクトを支える開発コミュニティの活性化と、主要開発者の負担軽減、そして長期的に開発・メンテナンスを継続できる体制の構築が不可欠である⁴²。
- 持続可能なマネタイズモデル: 現在は主に寄付⁵や一部インスタンスの法人運営(例: MisskeyHQ²)に依存しているが、プロジェクト全体の持続可能性を高めるための、オープンソースの理念と両立するようなマネタイズ方法の模索も長期的には課題となり得る。
- モデレーションと安全性: 分散型であるがゆえに生じる、インスタンス間のルール・モデレーション基準の差異や、不適切コンテンツ、スパム⁹、あるいはプライバシー侵害⁵⁴といった問題への対策。Fediverse全体での協調も視野に入れた取り組みが必要となる可能性がある。

Misskeyは、そのユニークな機能と設計思想により、既存のSNSとは異なる価値を提供するプラットフォームである。上記の課題にコミュニティ全体で取り組み、ActivityPubによるFediverse内での連携を深めつつ、Misskeyならではの魅力を維持・発展させていくことができれば、今後ますます多様化・細分化していくであろうソーシャルメディア市場において、確固たる地位を築き、多くのユーザーにとって魅力的な選択肢であり続けるだろう。

引用文献

1. gihyo.jp, 4月 21, 2025にアクセス、
<https://gihyo.jp/article/2023/02/misskey-01#:~:text=Misskey%E3%81%AF%E3%80%81%E3%82%AA%E3%83%BC%E3%83%97%E3%83%B3%E3%82%BD%E3%83%BC%E3%82%B9%E3%81%AE.%E5%91%BC%E3%81%B6%E3%83%A6%E3%83%BC%E3%82%B6%E3%83%BC%E3%82%82%E5%A4%9A%E3%81%84%E3%81%A7%E3%81%99%E3%80%82>
2. Misskey - Wikipedia, 4月 21, 2025にアクセス、<https://ja.wikipedia.org/wiki/Misskey>
3. Misskeyの気になるサーバーをいろいろ見てきたのでレポ - note, 4月 21, 2025にアクセス

- ス、<https://note.com/volanstars/n/nf5f255d8c6bd>
4. 次世代Twitterと話題「Misskey」使い方 インスタンス？ 分散型SNS？ その基本を解説 - KAI-YOU, 4月 21, 2025にアクセス、<https://kai-you.net/article/86094>
 5. Misskeyについて | Misskey Hub, 4月 21, 2025にアクセス、<https://misskey-hub.net/ja/docs/about-misskey/>
 6. 歴史資料館 | Misskey Hub, 4月 21, 2025にアクセス、<https://misskey-hub.net/ja/about-us/history/>
 7. SNSが好きだけでもう疲れた時に会うためのMisskey | cloud.config Tech Blog, 4月 21, 2025にアクセス、<https://tech-blog.cloud-config.jp/2022-04-07-introducing-misskey/>
 8. 【話題沸騰中】Misskey(ミスキー)とは？ エンタメ性抜群な分散型SNS！ | 名古屋のホームページ制作なら WWG, 4月 21, 2025にアクセス、<https://wwg.co.jp/blog/31341>
 9. 分散型SNSとは？ メリットとデメリット、有名な4つのSNSを紹介 - PreBell, 4月 21, 2025にアクセス、https://prebell.so-net.ne.jp/tips/pre_24080601.html
 10. 分散型SNS Misskey(ミスキー)とは？ 始め方から使い方までを解説 - Value Note(バリューノート), 4月 21, 2025にアクセス、<https://www.value-domain.com/media/misskey/>
 11. Misskeyプロジェクト 公式サイト, 4月 21, 2025にアクセス、<https://misskey-hub.net/ja-ks/>
 12. TwitterからMisskeyへ移った感想 - わかばTRPG部, 4月 21, 2025にアクセス、<https://wakabatrp.com/blog/2023/07/27/twitter%E3%81%8B%E3%82%89misskey%E3%81%B8%E7%A7%BB%E3%81%A3%E3%81%9F%E6%84%9F%E6%83%B3/>
 13. TwitterからMisskeyへ移った感想 - わかばTRPG部, 4月 21, 2025にアクセス、<https://wakabatrp.com/2023/07/27/twitter%E3%81%8B%E3%82%89misskey%E3%81%B8%E7%A7%BB%E3%81%A3%E3%81%9F%E6%84%9F%E6%83%B3/>
 14. 分散型SNS「Misskey」への開発支援に関するお知らせ - エックスサーバー, 4月 21, 2025にアクセス、<https://www.xserver.co.jp/news/news-article.php?id=85>
 15. Misskeyアプリイメージを利用する - シンVPS, 4月 21, 2025にアクセス、https://www.shin-vps.jp/support/manual/man_server_app_use_misskey.php
 16. MastodonやMisskeyなどのFediverseサーバーの統計サイトFediDBが新UIのベータサイトを公開 | gihyo.jp, 4月 21, 2025にアクセス、<https://gihyo.jp/article/2023/04/fedidb-new-beta-site>
 17. 分散型SNS「Misskey」、それを支える技術スタック - gihyo.jp, 4月 21, 2025にアクセス、<https://gihyo.jp/article/2023/02/misskey-01>
 18. サーバー一覧 | Misskey Hub, 4月 21, 2025にアクセス、<https://misskey-hub.net/ja/servers/>
 19. さまざまな機能 | Misskey Hub, 4月 21, 2025にアクセス、<https://misskey-hub.net/ja/docs/for-users/features/>
 20. X(Twitter)の代わりになるSNSは？ 対抗サービスのユーザー数と特徴まとめ, 4月 21, 2025にアクセス、<https://gaiax-socialmedialab.jp/x-alternative-sns/20240301>
 21. Twitterユーザーに送るMisskeyのススメ〜最高に楽しいSNSへようこそ！ - note, 4月 21, 2025にアクセス、https://note.com/okayu_tar_gz/n/n59a941ae0e21
 22. Misskey Hub - Misskeyプロジェクト 公式サイト, 4月 21, 2025にアクセス、<https://misskey-hub.net/ja/>
 23. Misskeyのおひとり様インスタンスを建てる - Zenn, 4月 21, 2025にアクセス、

- <https://zenn.dev/see2et/articles/building-an-one-person-instance-of-misskey>
24. Misskeyとは？ - Join the Fediverse, 4月 21, 2025にアクセス、
https://joinfediverse.wiki/index.php?title=Misskey/ja&mobileaction=toggle_view_desktop
 25. Misskey系とMastodon系の違いのこと - 松浦はこ - FANBOX, 4月 21, 2025にアクセス、
<https://matsbox.fanbox.cc/posts/5776373>
 26. ネットワークに基づく分散型+中央集権SNS一覧(damus, nostr, Bluesky, AT Protocol, Mastodon, Misskey, Threads, ActivityPub ...) - Qiita, 4月 21, 2025にアクセス、
<https://qiita.com/gpsnmeajp/items/1413960dd2cf4488ccd5>
 27. 各ロールの機能について - Misskey.io サポート, 4月 21, 2025にアクセス、
<https://support.misskey.io/hc/ja/articles/6521352727695-%E5%90%84%E3%83%AD%E3%83%BC%E3%83%AB%E3%81%AE%E6%A9%9F%E8%83%BD%E3%81%AB%E3%81%A4%E3%81%84%E3%81%A6>
 28. Misskey.ioに来たばかりののユーザーが知っておくといこと - note, 4月 21, 2025にアクセス、
https://note.com/rrrb_n/n/n563dfd537695
 29. 【X(旧Twitter)の代替品！】分散型SNS「Misskey(ミスキー)」の使い方や特徴を徹底解説！, 4月 21, 2025にアクセス、
<https://inglow.jp/weblog/snsmarketing/misskey-explanation/>
 30. Markup language For Misskey | MisskeyHQ Wiki, 4月 21, 2025にアクセス、
<https://wiki.misskey.io/ja/function/mfm>
 31. チャンネル一覧 | Misskey Hub, 4月 21, 2025にアクセス、
<https://misskey-hub.net/ja/docs/for-developers/api/streaming/channel/>
 32. Misskeyに来た新規ユーザーが戸惑う要素を解説する記事 | れるらば - note, 4月 21, 2025にアクセス、
https://note.com/rrrb_n/n/n567ff2e57b51
 33. Mastodon(マストドン)、Misskey(ミスキー)、そしてFediverse(フェディバース)との付き合い方 - note, 4月 21, 2025にアクセス、
https://note.com/rrrb_n/n/n58724e1009f9
 34. 【Misskey】誰でも簡単にMisskeyサーバーの立て方を紹介！ | 揚げポテ知恵ログ+, 4月 21, 2025にアクセス、
<https://note.agepote.jp/it-sp-pc/misskey-instance-xservervps>
 35. Misskeyインスタンスを立てたときの諸々メモ | ぽけ手帳, 4月 21, 2025にアクセス、
<https://www.pokete.com/server-startup-misskey>
 36. Misskeyサーバーをちゃんと建ててみた #Docker - Qiita, 4月 21, 2025にアクセス、
<https://qiita.com/Soli0222/items/1a8f854706528b63a8e2>
 37. 君たちはどのSNSで生きるか - anopara, 4月 21, 2025にアクセス、
<https://anopara.net/post/240102-sns>
 38. 【鯖缶入門】Misskeyインスタンスを建てたらやっておきたい事 - Qiita, 4月 21, 2025にアクセス、
https://qiita.com/mai_llj/items/a1a4c65651af904cac60
 39. タグ Misskey に関するスライド | ドクセル - Docswell, 4月 21, 2025にアクセス、
<https://www.docswell.com/tag/Misskey>
 40. Misskeyをフォークしてリファクタリングしまくっている - Zenn, 4月 21, 2025にアクセス、
<https://zenn.dev/okayurisotto/articles/d1e4cbdcbf5725>
 41. サーバーの機能 - Misskey Hub, 4月 21, 2025にアクセス、
<https://misskey-hub.net/ja/docs/for-admin/features/>
 42. Misskey開発の現状について思いをしたためたもの - GitHubのGist, 4月 21, 2025にアクセス、
<https://gist.github.com/syuilo/1a8cc608b2ab7451151e409029a943bb>

43. MastodonやMisskeyとはどんなもの？ | AKIRA_デジタルデザイン - note, 4月 21, 2025にアクセス、https://note.com/ak_schweitzer/n/n233c7c487018
44. Twitterの移行先はここだ！ 類似SNS紹介 - Qiita, 4月 21, 2025にアクセス、<https://qiita.com/MiyuWaka725/items/9495e803073afbed49ce>
45. XServer SNSを使うとMisskeyやMastodon、Blueskyを簡単に始められるって本当？【実際に使ってみた】 | 揚げポテ知恵ログ+, 4月 21, 2025にアクセス、<https://note.agepote.jp/it-sp-pc/xserver-sns>
46. Mastodonにある程度慣れてきた人による、Twitterとの比較に関する所見 | KMY - note, 4月 21, 2025にアクセス、<https://note.com/kmycode/n/n8c3d762f84d9>
47. Twitterからの乗り換えが進む？ 今大注目の分散型SNSとは - マナミナ, 4月 21, 2025にアクセス、<https://manamina.valuesccg.com/articles/2580>
48. あなたは何を選んでる？ 3つの最新SNSサービスを調べてみた話, 4月 21, 2025にアクセス、https://niigata.insight-lab.co.jp/blog/sns_service
49. 次のTwitterはMastodonかMisskeyか!? Fediverse (フェディバース) な世界へ飛び込んでみる, 4月 21, 2025にアクセス、<https://k-tai.watch.impress.co.jp/docs/column/minna/1458691.html>
50. Misskey.io, 4月 21, 2025にアクセス、<https://misskey.io/>
51. Misskey Shareについて, 4月 21, 2025にアクセス、<https://misskeyshare.link/howtouse.html>
52. 歴史 - Misskey Wiki, 4月 21, 2025にアクセス、<https://wiki.misskey.io/ja/software/misskey>
53. 【Misskey開発者syuilo氏】「楽しんでほしい」ピュアな思いがMisskeyを生んだ。異様な熱狂を放つSNSの今までとこれから | レバテックラボ(レバテックLAB), 4月 21, 2025にアクセス、https://levtech.jp/media/article/interview/detail_268/
54. Misskeyを使って内輪向けのクローズドなSNSを構築してみようとしたけどうまくいかなかった話, 4月 21, 2025にアクセス、<https://zenn.dev/pesi/articles/0751aae26a631f>