

# OS と Web ブラウザを統合したローカル LLM 支援型 オペレーティングプラットフォームの開発

## —OS と Web をつなぐ LLM プラットフォーム—

### 1. 背景

今日の Web ブラウザは Web 技術の発展により、単なる文書閲覧ツールではなく、多様なアプリケーションを動作させる重要な実行基盤へと発展してきた。Web アプリケーションの普及によって、日常の多くの作業が Web ブラウザ上で完結するようになった。

一方で、現在の Web ブラウザはセキュリティ上の理由からローカル OS とは意図的に隔離されており、ローカルのファイルシステムや周辺機器へのアクセスは限定的である。そのため、ユーザーは依然として、ローカルのネイティブアプリケーションと Web アプリケーションを別々のものとして扱わなければならない。ユーザーは Web アプリとローカルアプリの間を手作業で行き来する必要がある、非効率かつミスが生じる原因となっている。

また、近年急速に発展している大規模言語モデル（Large Language Model、以下 LLM）についても同様である。現在の高性能な LLM の多くはクラウド上で動作しているため、ローカルと連携した作業を行うためには機密データをサーバに送信する必要があり、コスト面での問題が発生している。

これは企業や団体において重大な情報漏洩リスクや心理的ハードルとなっている。また、日常的な PC 操作をクラウド上のサーバーで動作し、ネットワーク経由で利用する LLM であるクラウド LLM で支援・自動化しようとした場合、利用量に応じたランニングコストの増大も無視できない課題である。

### 2. 目的

本プロジェクトの目的は、Floorp Web ブラウザを中核として、ローカル OS 環境と Web ブラウザ環境の分断を解消し、両者を横断的に扱える次世代の LLM プラットフォーム「Floorp OS」を開発することである。Floorp Web ブラウザとは浅野が開発者を務める Firefox ベースの Web ブラウザである。高度なカスタマイズ性を特徴としており、広く利用されている。

この目標を実現するため、本プロジェクトでは以下の 3 点を軸として開発を進める。

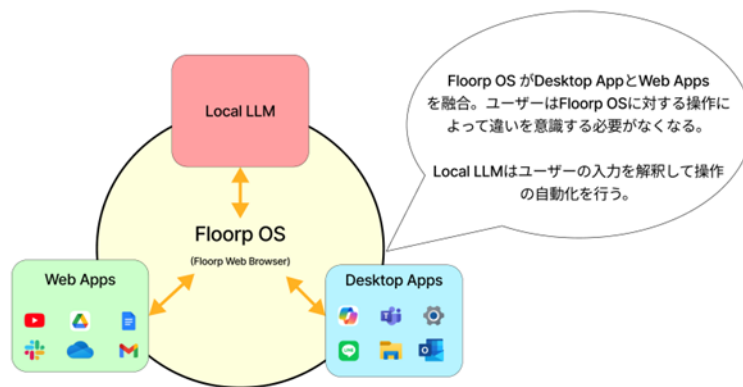


Figure 1: プラットフォームのコンセプト

1. ブラウザを単なる閲覧ソフトではなく LLM 活用基盤にする  
 現在、Web ブラウザは日常作業の大部分を担う存在でありながら、ローカル環境へのアクセスが制限された閲覧ツールに留まっている。本プロジェクトでは、Figure 1 に示すように、ブラウザを OS 全体を制御するハブ（Operating Platform）へ拡張し、Web と OS の境界をなくす LLM 活用基盤を構築する。
2. 自然言語で Web とローカルを横断して操作できるようにする  
 自然言語による指示を元に、ローカルのファイル操作と Web サイト操作を横断して自動実行する。これにより、分断されていた両者をシームレスに繋ぐユーザー体験を実現する。
3. ローカル LLM で完結する安全な実行環境をつくる  
 現在のクラウド型 LLM は、機密情報の漏洩リスクや API 利用コストの増大という課題を抱えている。本プロジェクトでは、ユーザーの PC で実行されるローカル LLM をプラットフォームの標準機能として活用することで、機密性の高いローカルファイルや業務データを外部サーバに送信せずに処理できる環境の実現を目指す。また、LLM が実行する自動化ワークフローについても、権限を細かく管理できる実行環境の上で動作させることで、セキュリティとプライバシーに配慮した安全な LLM 実行基盤を構築する。

### 3. 開発の内容

本プロジェクトでは、Web とローカル OS に分断されていた処理を、自然言語による指示から一連の操作として実行できるようにするため、Floorp OS の中核基盤を新たに開発した。この基盤は、ユーザーの指示に対して実行時に逐次判断を重ねるのではなく、まず実行手順を明示的なワークフローとして生成し、その内容を事前に定めたいうで静的に解釈しながら必要なプラグインを順次呼び出して処理を実行する。また、これを Floorp Web ブラウザに統合することで、ブラウザ操作とローカル OS 操作を一つの流れとして扱える構成を実現した。

#### 1. ワークフロー実行基盤の開発

LLM が生成したワークフローを実行する独自のワークフローランタイムである Saphillon を開発した。Saphillon (サフィオン) は Floorp OS の中核として機能する LLM オートメーション基盤である。これにより、Web ページの操作、ローカルファイルの処理、外部機能の呼び出しなどを、自然言語の指示から一連の処理として実行できるようにした。

## 2. プラグインランタイムとプラグインシステムの設計

ワークフローから利用できる機能を拡張可能にするため、プラグインランタイムを実装するとともに、プラグインシステムを設計した。ブラウザ制御、ファイルシステム、検索、光学的文字認識 (OCR)、ローカル LLM 連携などの機能をプラグインとして提供し、用途に応じて機能を追加できる構成とした。

## 3. サンドボックスと権限管理による安全な実行

ワークフローとプラグインはそれぞれ独立したサンドボックス環境で、システムと隔離して実行する設計とした。さらに、権限に基づいてアクセス可能な機能やリソースを細かく制御することで、柔軟性を保ちながら安全に動作する実行基盤を構築した。

## 4. Floorp Web ブラウザおよびユーザー向け UI との統合

Floorp Web ブラウザ側にも改修を行い、生成したワークフローからブラウザを直接操作できる仕組みを実装した。あわせて、自然言語入力からワークフローの生成・実行・結果確認までを扱うユーザー向け UI を整備し、Web とローカル OS を横断する処理を分かりやすく利用できるようにした。

## 4. 従来の技術 (または機能) との相違

従来の Web ページ操作やローカル OS 上の処理を自動化する仕組みは、ブラウザ内の操作、ローカルアプリケーションの操作、LLM による判断をそれぞれ別個に組み合わせる必要があり、導入や運用が複雑になりやすかった。この点、本プロジェクトで開発した Floorp OS は、ブラウザを中核に据え、Web ページ操作とローカル OS 上の処理を一つの流れとして横断的に扱える基盤を実現している。

また、実行時にその場で逐次判断を重ね続けるのではなく、まず実行手順を明示的なワークフローとして生成し、その内容を事前に定めたいうで処理を順次実行する構成を採ることで、実行内容を確認しやすく、再現性を確保しやすくしている。

さらに、ローカル LLM を活用できることに加え、ワークフローとプラグインをサンドボックスで隔離し、権限に基づいて動作を制御することで、安全性を考慮した実行環境を実現した。加えて、Floorp Web ブラウザの内部に改修を加えることで、拡張機能や外部ツールだけでは扱いにくいブラウザ内部の状態取得や、ログイン中のセッションを活かした動的ページの操作にも対応しやすい構成としている。

そして、ダウンロード・インストール・起動までの一連の導入プロセスをブラウザ側に組み込んでおり、ユーザーはブラウザからインストールボタンを押すだけで導入できる。これらの点で、Floorp OS は、Web とローカルを安全かつ一体的に

扱いながら、導入のしやすさと実用性を両立した点において、従来の仕組みと大きく異なると言える。

## 5. 期待される効果

本プロジェクトによって期待される効果は、Web とローカル OS に分断されていた作業を一つの流れとして扱えるようになることで、日常業務や開発作業の効率を大きく向上できる点にある。従来は、Web 上で情報を確認し、その内容をもとにローカルファイルを編集し、別のアプリケーションで結果を反映するといった手順を人手で往復する必要があったが、Floorp OS ではこれらを自然言語による指示から一連の処理として実行できるため、作業時間の短縮や操作ミスの低減が期待される。また、生成された実行手順をワークフローとして事前に確認し、その内容に沿って処理を順次実行する構成により、何が行われるかを把握しやすく、利用者が安心して自動化を活用しやすくなる。さらに、同様の処理を再利用しやすくなることで、繰り返し作業の効率化や業務手順の属人化の軽減も期待される。

加えて、ローカル LLM を活用し、ワークフローとプラグインを権限管理付きのサンドボックス環境で動作させることにより、機密性の高い情報を外部サーバに送信せずに処理できる環境の実現が期待される。これにより、企業や研究機関など、情報管理上の制約からクラウド型 AI を導入しにくかった場面でも活用の可能性が広がる。また、ブラウザから容易に導入できる構成や、用途に応じて機能を追加できるプラグインシステムを備えることで、個人利用から業務利用まで幅広い場面で活用しやすい基盤になると期待される。これらの点から、Floorp OS は、Web とローカルを横断する PC 操作をより安全かつ効率的に支援し、新しいユーザー体験と実用的な自動化の可能性を広げることが期待される。

## 6. 普及（または活用）の見通し

本プロジェクトの普及や活用の見通しとしては、まず、日常的な PC 作業の多くがすでに Web ブラウザを起点として行われていることから、Floorp OS は既存の作業環境に比較的自然的に組み込みやすいと考えられる。とりわけ、Web 上での情報収集とローカルでの文書作成、ファイル整理、開発作業などを往復する場面において、その有用性が発揮されることが期待される。また、ブラウザから導入できる構成や、ローカル LLM を活用しながら権限管理の下で安全に実行できる設計は、個人利用に加えて、機密情報を扱う企業や研究機関などにおいても活用の可能性を広げる要因になると考えられる。

今後、プラグインストアや開発者向け SDK の整備が進めば、用途に応じた機能拡張がより容易になり、開発、事務、調査、研究支援など多様な領域へ活用範囲が広がることが期待される。さらに、評価基盤の拡充やマルチモーダル LLM への対応、システム全体の安定性向上を進めたうえでベータ版の一般公開に至れば、実際の利用者からのフィードバックを通じてユースケースや UI/UX を継続的に改善できるようになる。その結果、Floorp OS は研究段階のプロトタイプにとどまらず、

Web とローカル OS を横断して活用できる実用的な LLM プラットフォームとして発展していくことが期待される。

#### 7. クリエータ名（所属）

- 高橋 侑大（東洋大学 情報連携学部 情報連携学科）
- 浅野 凌輔（東洋大学 情報連携学部 情報連携学科）
- 鮎澤 颯人（電気通信大学 情報理工学域 III 類）

#### （参考）関連 URL

- Floorp OS Web サイト: <https://ja.floorp.app/floorp-os>
- Floorp Web ブラウザ GitHub リポジトリ: <https://github.com/Floorp-Projects/Floorp>
- Floorp OS Core GitHub リポジトリ: <https://github.com/Floorp-Projects/Floorp-OS-Automator-Backend>
- Sapphillon GitHub リポジトリ: <https://github.com/Sapphillon/Sapphillon>