

卒業論文 2019 年度（令和 01 年度）

Hyper Launcher  
ホットキー型アプリケーションランチャーの  
利便性を高める研究

慶應義塾大学 環境情報学部

佐藤 佑哉

増井俊之研究会

2020 年 1 月

卒業論文 2019年度（令和01年度）

# Hyper Launcher

## ホットキー型アプリケーションランチャーの 利便性を高める研究

### 論文要旨

ホットキー型のアプリケーションランチャーはキーの入力のみで特定のアプリケーションを起動することができるとしても強力なユーティリティソフトウェアであるが、使いこなすのが難しく活用している人は少ない。このタイプのランチャーが広く活用されていないのは、その設定の煩雑さや設定を記憶しなければならないというハードルの高さにある。これらの問題は工夫次第でまだまだ改善できる余地があると思われる。

本研究では、既存のアプリケーションランチャーの種類や特徴について調査した上で、より強力で扱いやすいホットキー型のランチャーを実現するためのインターフェースを提案し、その発展性について考察する。

### キーワード

ランチャー, インターフェース

慶應義塾大学 環境情報学部

佐藤 佑哉

# 目 次

<b>第 1 章 序論</b>	<b>1</b>
1.1 背景 . . . . .	2
1.1.1 アプリケーションランチャーの種類と比較 . . . . .	2
1.1.2 既存のアプリケーションの問題点 . . . . .	4
1.2 本研究の目的 . . . . .	5
1.3 本論文の構成 . . . . .	5
<b>第 2 章 システムの提案</b>	<b>7</b>
2.1 設計 . . . . .	8
2.2 基本操作 . . . . .	8
2.2.1 アプリケーションの登録と起動 . . . . .	8
2.2.2 アプリケーションの登録変更と解除 . . . . .	8
2.2.3 複数アプリケーションの登録と操作 . . . . .	9
<b>第 3 章 実装</b>	<b>13</b>
3.1 システム構成 . . . . .	14
3.2 クライアントアプリケーション . . . . .	14
3.2.1 Electron . . . . .	14
3.2.2 AppleScript . . . . .	15
3.3 データストア . . . . .	15
<b>第 4 章 評価</b>	<b>16</b>
4.1 . . . . .	17
<b>第 5 章 考察</b>	<b>18</b>
<b>第 6 章 結論</b>	<b>19</b>
<b>謝辞</b>	<b>20</b>
<b>参考文献</b>	<b>21</b>

## 図 目 次

1.1	Dock . . . . .	2
1.2	Dock 内のメニュー . . . . .	3
1.3	Spotlight . . . . .	4
1.4	PMenu . . . . .	5
1.5	Snap . . . . .	6
2.1	Hyper Launcher . . . . .	9
2.2	ドラッグアンドドロップによる登録 . . . . .	10
2.3	追加ボタンから登録 . . . . .	10
2.4	ドラッグアンドドロップによる追加 . . . . .	11
2.5	登録の変更 . . . . .	11
2.6	登録の解除 . . . . .	12
3.1	システムの構成 . . . . .	14

## 表 目 次

# **第1章　序論**

本章では、本研究の背景と目的、及び本論文の構成について述べる。

## 1.1 背景

アプリケーションランチャーはオペレーティングシステムに標準で搭載されているだけでなく、サードパーティ製としても多くのソフトウェアが開発、公開されている。ホットキー型のランチャー以外にも異なるインターフェースを持ったランチャーが数多く存在しており、それぞれ異なった利点/欠点がある。

### 1.1.1 アプリケーションランチャーの種類と比較

アプリケーションランチャーには以下のような種類があり、その特徴に合わせて様々な方法でアプリケーションを起動することができる。

### (1) パレット型

画面上の一部に固定されたパレット型のエリアにアプリケーションを登録し、マウスによるクリックで起動するタイプのランチャー。macOSにおけるDock（図1.1）がこれにあたる。デスクトップに常駐していることから簡単にアクセスでき、誰でも使いやすいものとなっている。しかしそのエリアは限られており、多くのアプリケーションを登録しようとすると、ボタン数を増やしたりそれぞれを小さく表示したりする必要がある。基本的にその数が増えれば増えるほど操作性が低下するため、数個から数十個の頻用するアプリケーションのみを登録して使用するのが推奨される。



図 1.1: Dock

(2) メニュー型

上述したDock等にメニューを設け、マウスを使用して登録したアプリケーションを起動できるようにするタイプのランチャー。ショートカットキーと組み合わせて使用されるものもある。複数のアプリケーションを一つにまとめることで場所を節約できるだけでなく、階層化によって自分の使いやすいように整理することもできる。しかし、項目や階層が増えれば目的のアプリケーションに辿り着くまでの操作ステップは増えることになる。

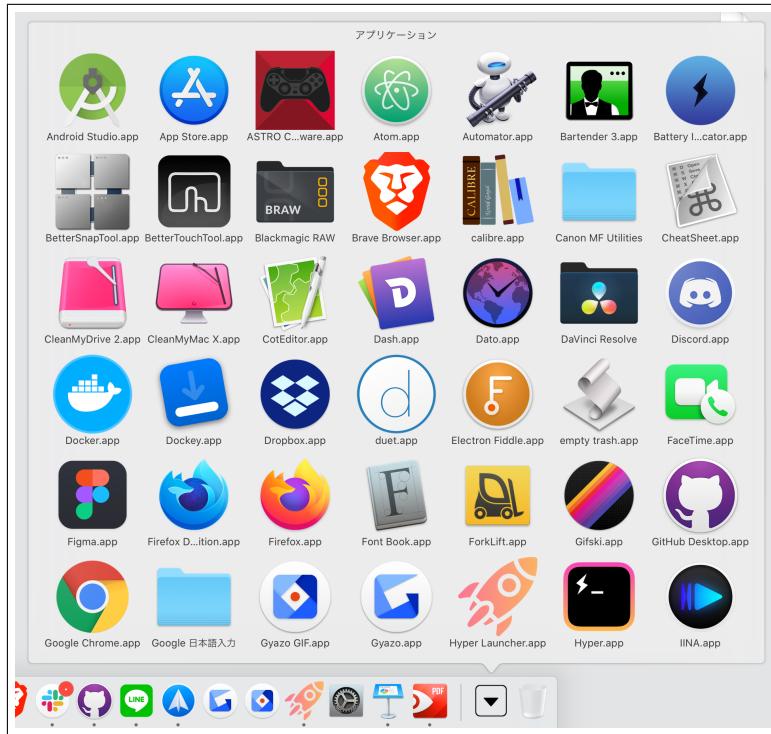


図 1.2: Dock 内のメニュー

### (3) 検索型

アプリケーションの名前を入力することで、対象のアプリケーションを起動するタイプのランチャー。macOS における Spotlight がこれにあたる。検索メニューは使用するときのみ表示されるため使用していない時は場所をとらず、キーボードのみの操作で完結しているのも利点の一つである。もちろんマウスと組み合わせて使用することもできる。また大抵の場合インクリメンタルサーチが導入されているため、名前を全て覚えていなくても起動することができる。しかし汎用性が高い反面、使い時は毎回ある程度のキーボード入力が必要となってしまうというのが欠点である。

### (4) ホットキー型

单一のキーもしくは複数のキーの組み合わせを入力するだけで、登録したアプリケーションを起動できるタイプのランチャー。今回着目しているのがこのタイプである。上で挙げたものとは違い、このタイプは標準で搭載されていないことがほとんどである。したがって自分にあったサードパーティ製ソフトウェアを探す必要がある。画面上に情報を表示する必要がなく場所を取らないことに加え、一発で特定のアプリケーションを起動できるため、とても強力なランチャーである。しかし設定が面倒であったり、どのキーにどのアプリケーションを登録したのか覚えておく

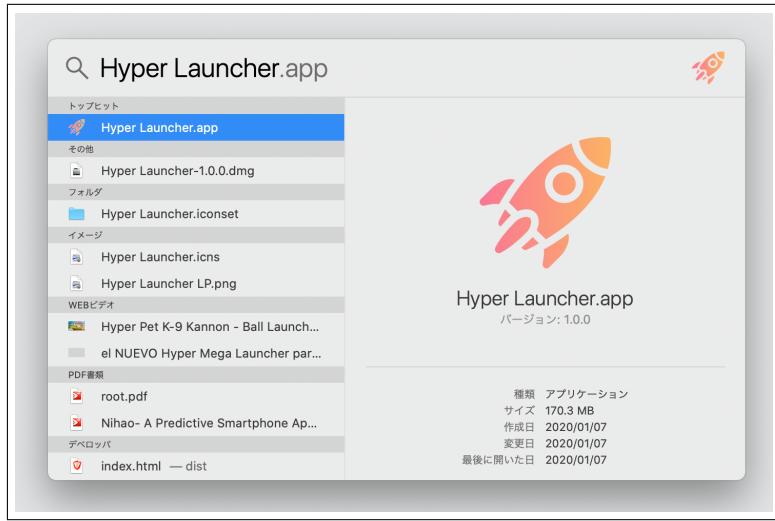


図 1.3: Spotlight

必要があつたりと、初心者には使いにくいタイプだとされているのも事実である。これこそが標準として機能が提供されていない理由の一つだと考えられる。

### 1.1.2 既存のアプリケーションの問題点

先述した通りホットキーを利用したアプリケーションランチャーは標準で搭載されておらず、使用者自ら使いやすいものを探す必要がある。しかしその種類は豊富とは言えず、突出した特徴があるわけでもない。例として筆者が日常的に使用してきた2つのアプリケーションランチャーを示す

#### (1) PMenu

PMenus(図1.4)はホットキーとメニューを利用したランチャーで、アプリケーションだけでなく様々なファイルをコマンドで操作することができる。使い勝手はいたって普通で可もなく不可もなしと言ったところである。対象に自分の好きなホットキーを割り当てられるのは利点であるが、キーを登録するために3ステップも必要になる。コンテキストメニューを利用する主な方法とされているため、ホットキーに関しては最低限の機能が実装されているだけと言える。

#### (2) Snap

Snap(図1.5)はホットキーを利用したランチャーで、macOSのDockを生かした機能が備わっている。SnapではDockに登録されたアプリケーションに左から順に数字を割り当てていき、その数字と設定したモディファイアキーとの組み合わせでアプリケーションを起動できるようになっている。これにより面倒な設定をすることなくホットキーの恩恵を授かることが可能となる。た

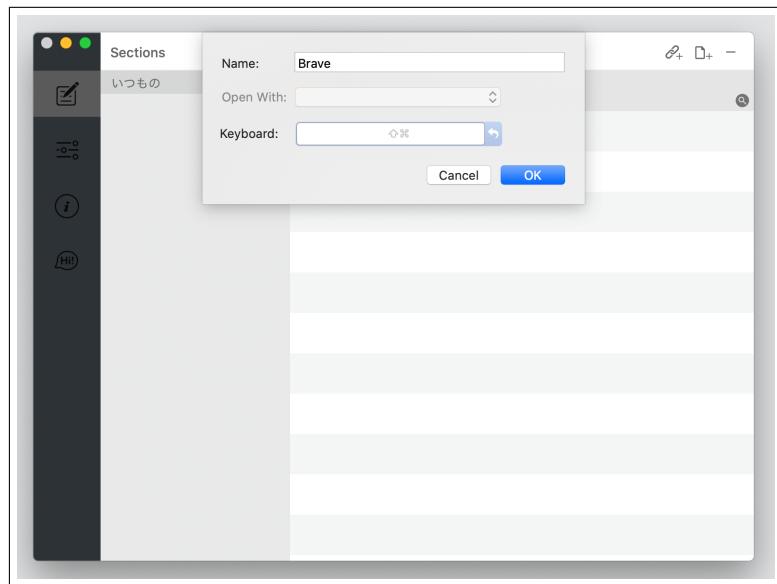


図 1.4: PMenu

だし Dock のサイズは有限であり画面上に場所もとるため、多くのアプリケーションを登録するというのではありません。

## 1.2 本研究の目的

既存のホットキーを利用したアプリケーションランチャーにおける不便を解消し、より強力でより多くの人に使いやすいシステムを開発することが本研究の目的である。

## 1.3 本論文の構成

第1章では、本研究における背景と問題意識、目的について述べた。

第2章では、第1章で述べた問題意識を踏まえ、システムを提案する。

第3章では、本論文で提案するシステムの詳細な実装について述べる。

第4章では、

第5章では、システムの考察を行う。

第6章では本研究を総括する。

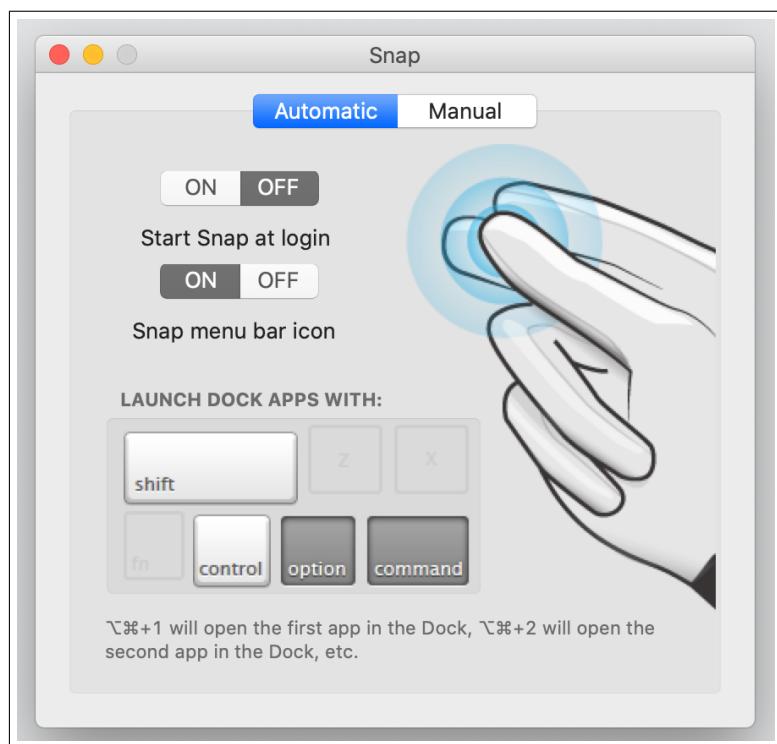


図 1.5: Snap

## **第2章 システムの提案**

本章では、アプリケーションランチャーについての背景を踏まえ、ホットキーを利用した新しいランチャーシステム「Hyper Launcher」を提案する。

## 2.1 設計

既存のホットキー型アプリケーションランチャーに見られる不便を解消するため、

1. アプリケーションの登録及びその変更が容易にできる
2. 設定が覚えやすい
3. より強力に操作できる

を実現するランチャーシステム「Hyper Launcher」を設計した。具体的には

1. アプリケーションの登録や変更をドラッグアンドドロップによって簡単に行えるようにする
2. 使用するホットキーの組み合わせを9つに制限し、そのキーに対してアプリケーションを登録することで、可能な限り認知不可を下げられるようにする
3. 単一のキーの組み合わせに対して複数のアプリケーションを登録できるようにする

というアプリケーションを実装した。

## 2.2 基本操作

自身の開発環境を考慮し、アプリケーションはmacOS上でデスクトップアプリとして利用できることとする。起動画面を図2.1に示す。画面は9つのセクションに分けられており、予めそれぞれにホットキーが指定されている。

### 2.2.1 アプリケーションの登録と起動

任意のアプリケーションをそれぞれのセクションにドラッグアンドドロップすることで簡単に登録することができる。また、タイトル横のプラスボタンを押すことでアプリケーション選択画面が出現し、そこからも登録することができる。後は指定されたキーを入力するだけでアプリケーションを起動することができる。(図2.4, 2.3)

### 2.2.2 アプリケーションの登録変更と解除

アプリケーションに紐付けるキーを変更したい時は、対象のアプリケーションを目的のセクションへドラッグアンドドロップするだけで操作が完了する。また、アプリケーションにホバーした際に右側に表れるバツボタンを押すことで登録を解除することができる。(図2.5, 2.6)

### 2.2.3 複数アプリケーションの登録と操作

既にアプリケーションが登録されているセクションにおいて、同じように新しいアプリケーションを追加するだけで複数のアプリケーションを登録することができる。アプリケーションは表示されている順番によって優先順位が定められており、キーを入力する度にその優先順位に基づき循環して起動するようになっている。その優先順についてもドラッグアンドドロップするだけで変更することができる。

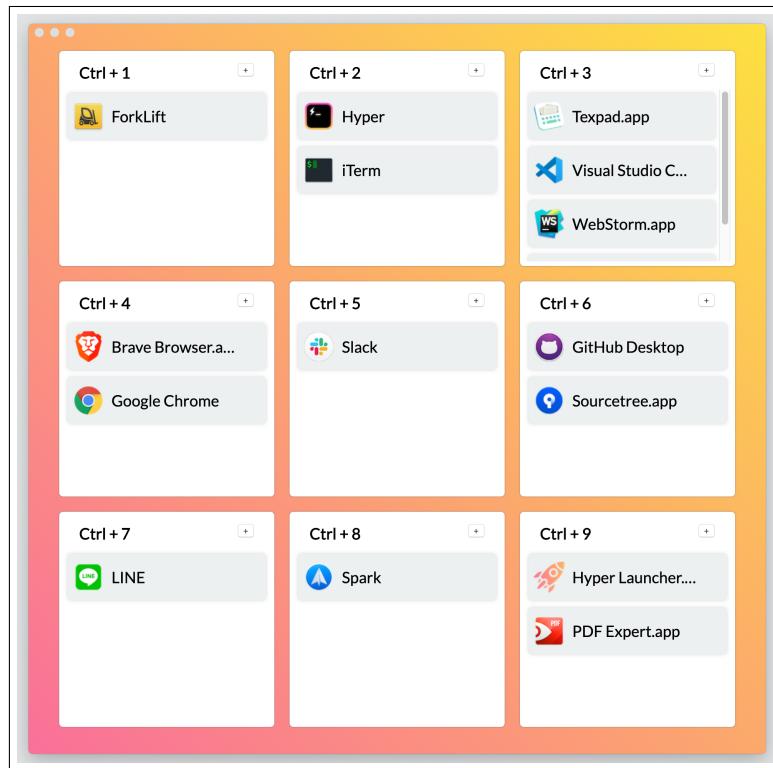


図 2.1: Hyper Launcher

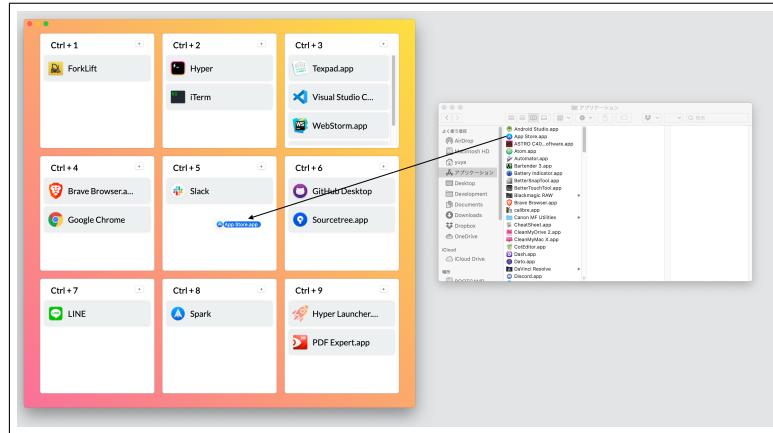


図 2.2: ドラッグアンドドロップによる登録

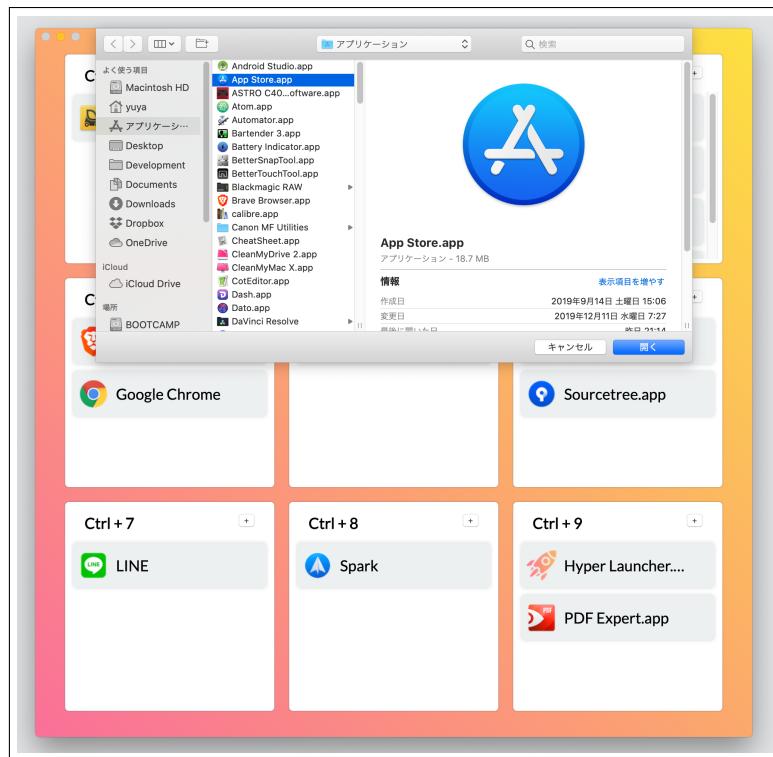


図 2.3: 追加ボタンから登録

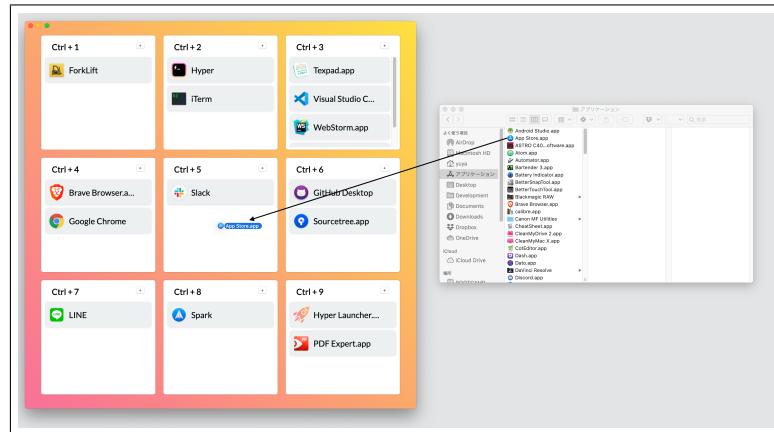


図 2.4: ドラッグアンドドロップによる追加

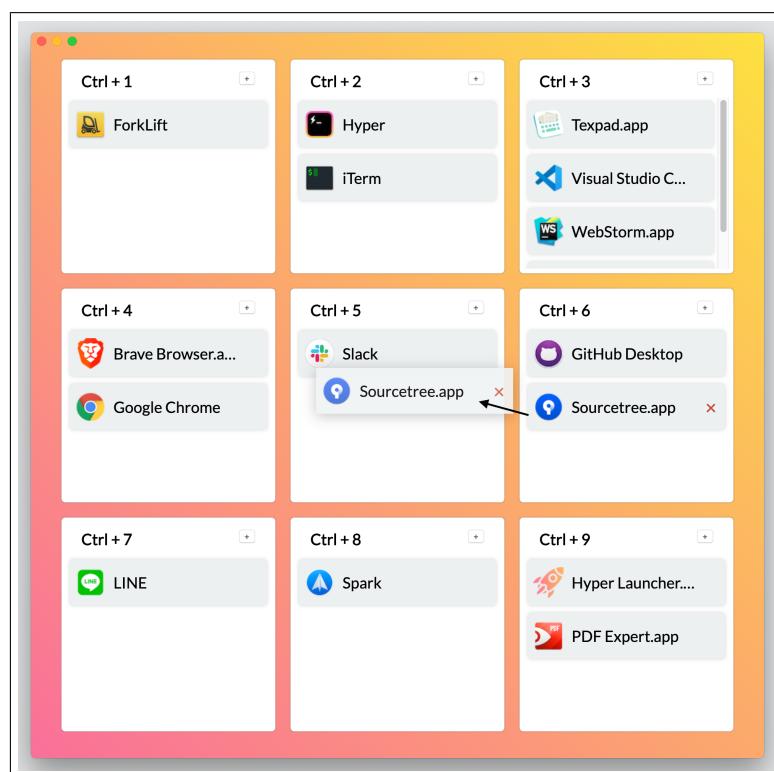


図 2.5: 登録の変更



図 2.6: 登録の解除

## 第3章 実装

本章では、第2章で述べたシステムの設計を受け、Hyper Launcher の実装について述べる。

### 3.1 システム構成

Hyper Launcher はユーザーが実際に操作するためのクライアントアプリケーションと、登録したデータを保存しておくためのデータストアから構成される。構成図を図 3.1 に示す。

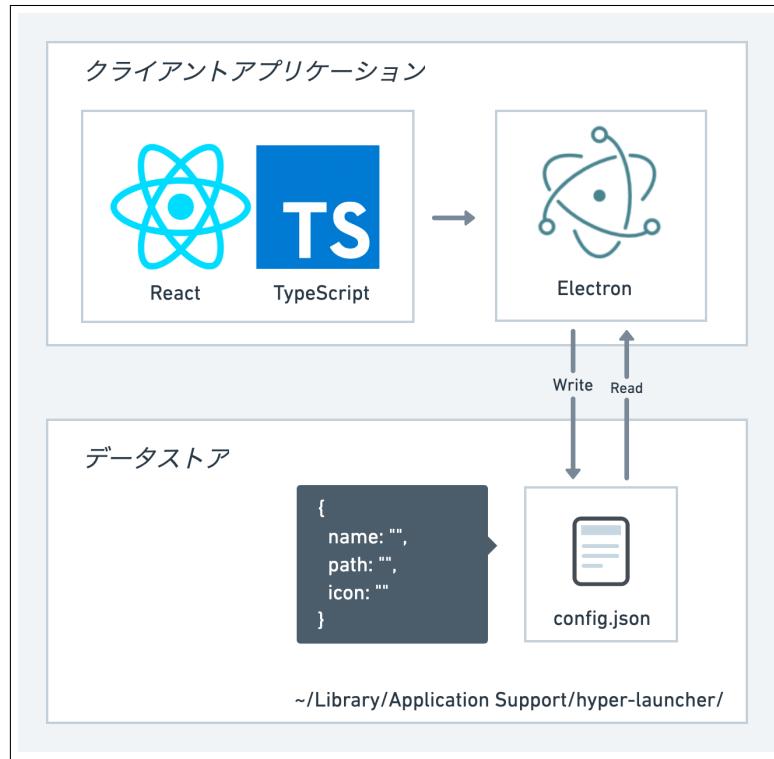


図 3.1: システムの構成

### 3.2 クライアントアプリケーション

クライアントアプリケーションは TypeScript 及び React などの Web 技術によって実装されており、macOS のデスクトップアプリケーションとして動作する。

#### 3.2.1 Electron

Electron は HTML、JavaScript、CSS を用いてクロスプラットフォームなデスクトップアプリケーションを作成することができるソフトウェアフレームワークである。Electron 自体は Chromium と Node.js を使用しており Web 技術のみで開発が完結するのが利点である。Electron の API ではグローバルなキーイベントのハンドリングを行えることに加え、アプリケーションの起動やアクティブ化なども操作できるため、ランチャーを作るにあたってとても有用なフレームワークである。

### 3.2.2 AppleScript

Web 技術のみでは難しい部分については AppleScript によって実装した。macOS には osascript と呼ばれるシェルスクリプトが存在し、これを利用することで Node.js から AppleScript を呼び出すことができる。AppleScript を使用することで、起動中のアプリケーションや特定のアプリケーションがアクティブかどうかといった情報を取得することができるようになる。これによってデスクトップアプリケーションとして十分な機能を実装することができた。ただし実行するために毎回子プロセスを立ち上げるため、何も考えずに使用してしまうとラグが発生してしまうこととなる。

## 3.3 データストア

ユーザーが登録した情報は、Electron が定義するユーザーデータ領域（/Library/Application Support/hyper-launcher/）に JSON ファイルとして永続化されている。データのやり取りが全てローカルで完結するため、オフラインでも使用できようになっている。データはホットキーの数字をキーにし対応する値として配列を持ったオブジェクトになっており、それぞれアプリケーションの名前、パス、そして base64 エンコードされたアイコンの 3 つを保持している。これにより単一のキーに対して複数のアプリケーションを登録することが可能となっている。以下にその例を示す。ただし icon の base64 文字列と中間部分は長くなるため省略した。

ソースコード 3.1: config.json

```
1 {
2   "shortcut": {
3     "1": [
4       {
5         "name": "ForkLift",
6         "path": "/Applications/ForkLift.app",
7         "icon": "*****"
8       }
9     ],
10    "2": [
11      {
12        "name": "Hyper",
13        "path": "/Applications/Hyper.app",
14        "icon": "*****"
15      }
16    ],
17    ...
18
19    "9": [
20      {
21        "name": "Hyper_Launcher",
22        "path": "/Applications/Hyper_Launcher.app",
23        "icon": "*****"
24      }
25    ],
26  },
27}
28 }
```

## **第4章 評価**

本章では、Hyper Launcher を実運用した結果および評価について述べる。

## 4.1

## **第5章 考察**

## **第6章 結論**

# 謝辞

## 参考文献

- [1] 著者名：文献名，書誌情報，出版年。
- [2] ほげ山太郎，ほげ山次郎：ほげほげ理論の HCI 分野への応用，ほげほげ学会論文誌，Vol.31, No.3, pp.194-201, 2009.
- [3] Taro Hogeyama, Jiro Hogeyama: The Theory of Hoge, *The Proceedings of The Hoge Society*, 2008.