

# パフォーマンスはなぜ重要なのか？

目先の UX(?) にとらわれないクライアントサイド開発の本質

2017/04/22 名古屋マークアップ勉強会 The Final 基調講演

@ahomu - Ayumu Sato



おはようございます  
@ahomuです

# @ahomu

- ▶ 佐藤 歩 (さとう あゆむ)
- ▶ 名古屋と渋谷を往復中の Web ディベロッパー
- ▶ 最近の関心は #webperf と #a11y と人事考課
- ▶ CyberAgent, Inc. または AbemaTV, Inc.
  - 技術本部 Web Initiative Center 主宰 / メディア統括 技術戦略室

# アジェンダ

- ▶ パフォーマンスの重要性
- ▶ パフォーマンスの基本的な「考え方、捉え方」
- ▶ パフォーマンスとクライアント環境の「多様性」
- ▶ クライアントサイド開発における「品質」の重要性



リンクをタップした後  
何秒まで耐えられる？



Tweet link

スクロールが重いサイト  
じっくり見ていられる？

# 大手検索エンジンと広告収益の事例

<https://conferences.oreilly.com/velocity/velocity2009/public/schedule/detail/8523>

- ▶ サーバーの応答を遅延させて収益への影響を測る実験
- ▶ **0.5秒の遅延 – 1.2% の減益**
- ▶ **1.0秒の遅延 – 2.8% の減益**
- ▶ **2.0秒の遅延 – 4.3% の減益**

# 2011年アメリカ大統領選の事例

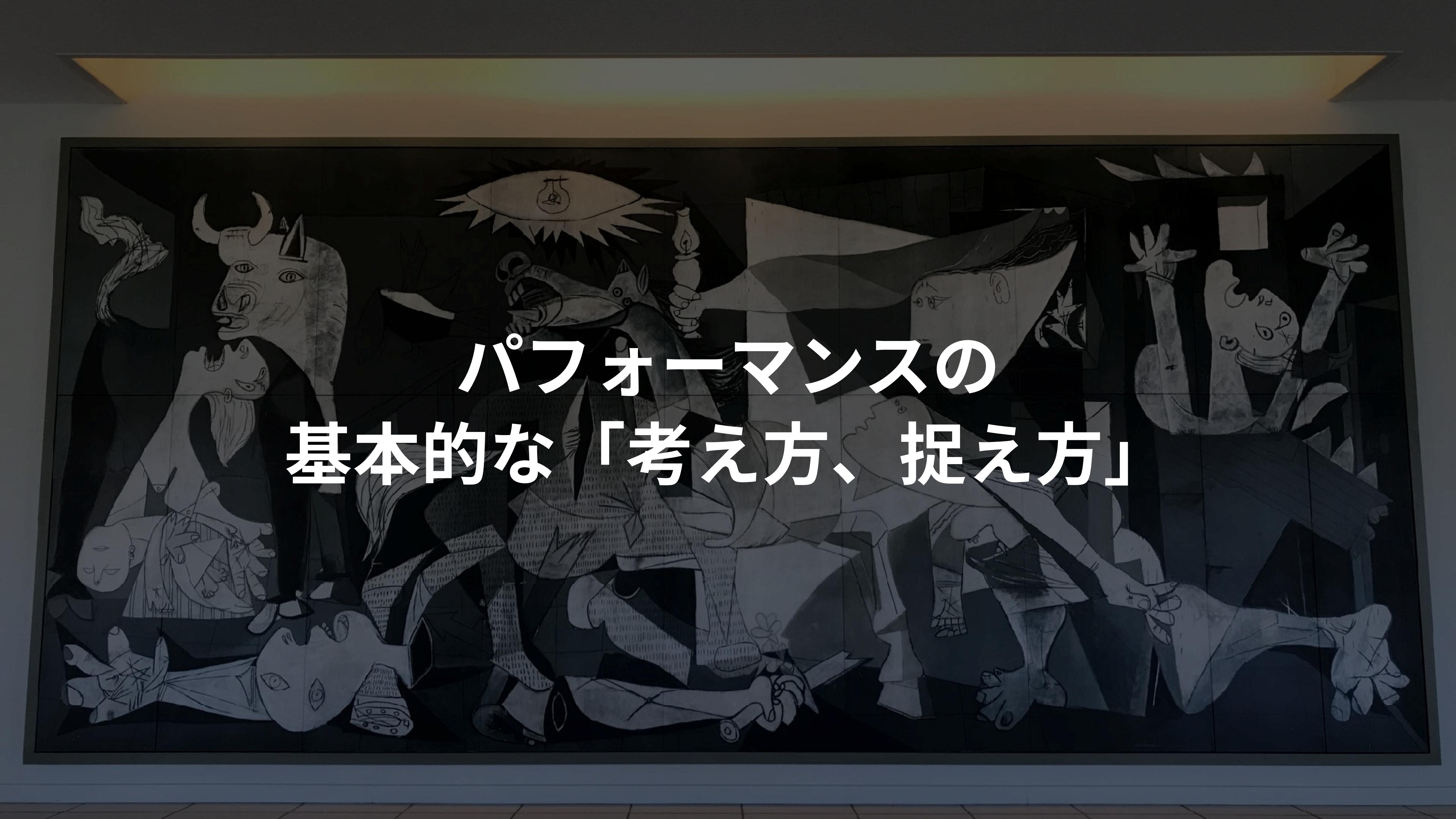
<http://kylerush.net/blog/meet-the-obama-campaigns-250-million-fundraising-platform/>

- オンライン寄付金プラットフォームの改修
- Web ページの表示を **60%** 高速化
- 寄付のコンバージョンレートが **14%** 向上
- 最終的に総額 11 億米ドルの寄付を獲得

# ソフトウェアダウンロードの事例

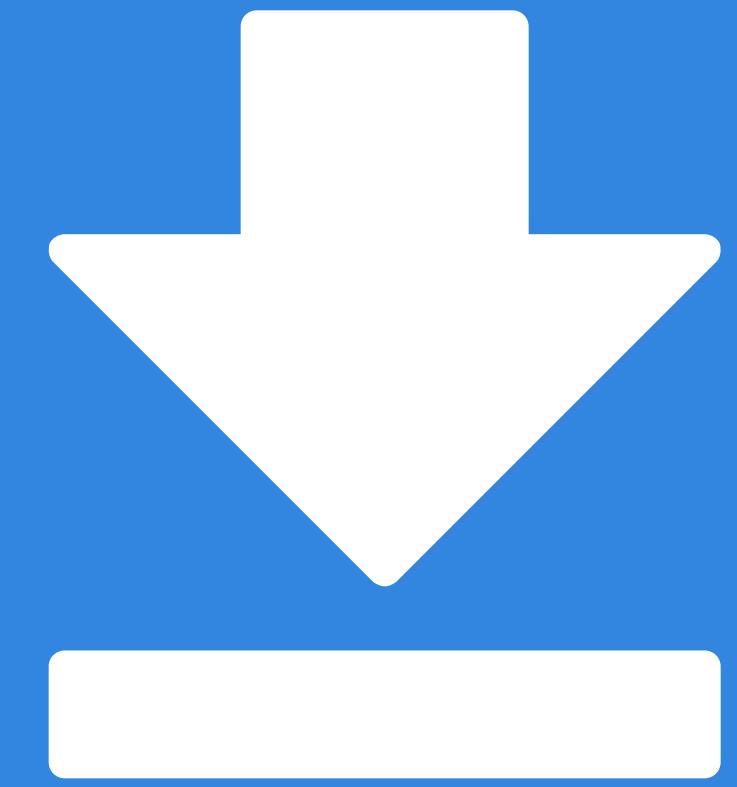
<https://blog.mozilla.org/metrics/2010/04/05/firefox-page-load-speed-%E2%80%93-part-ii/>

- Mozilla による Firefox のダウンロードキャンペーン
- ランディングページの表示を **2.2 秒** 高速化
- ダウンロードのコンバージョン率が **15.4%** 向上
- コメント欄で Addy Osmani が喧嘩売ってて笑える



# パフォーマンスの 基本的な「考え方、捉え方」

評価軸は大きく分けて2つ



ページロード



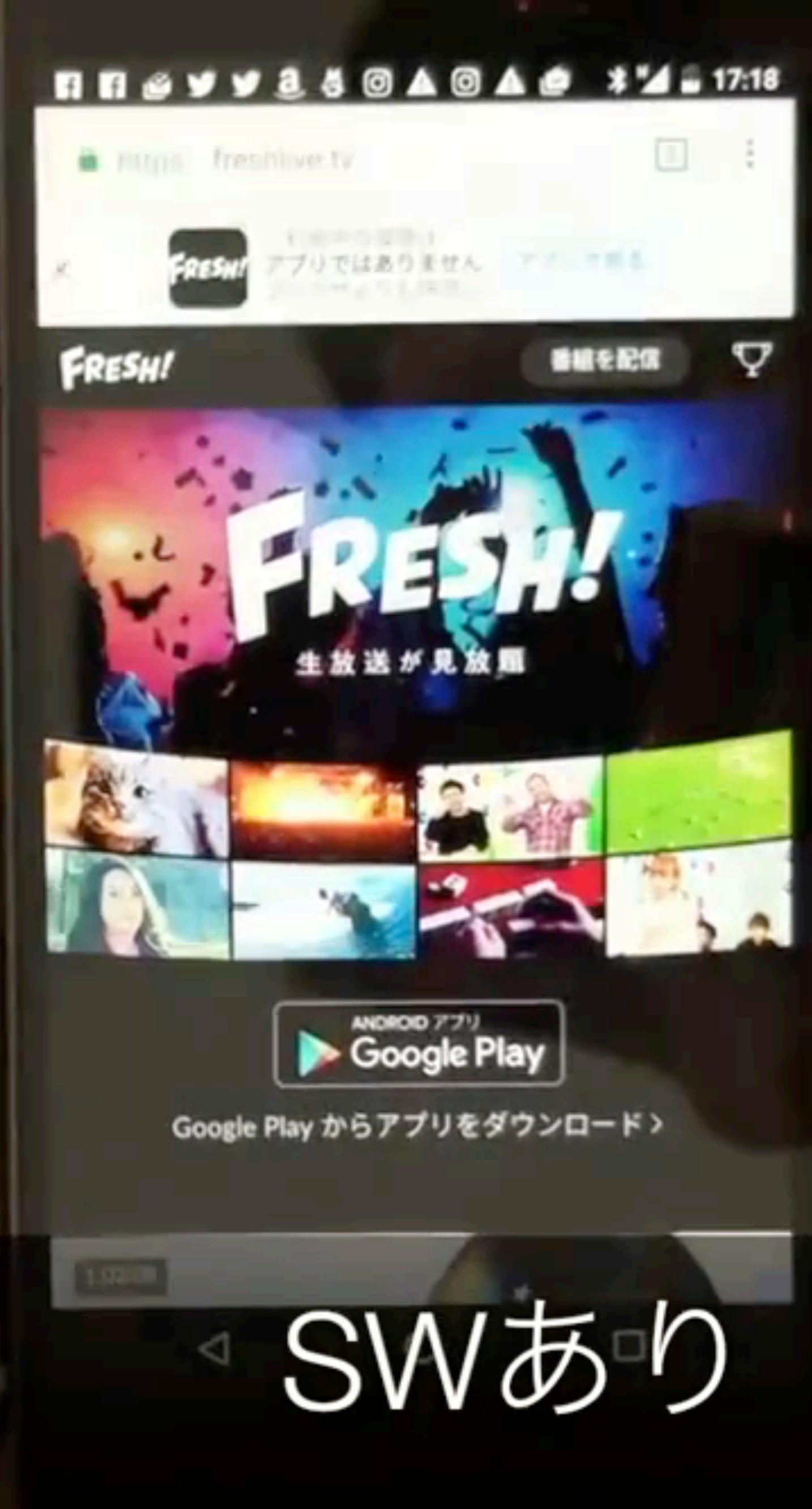
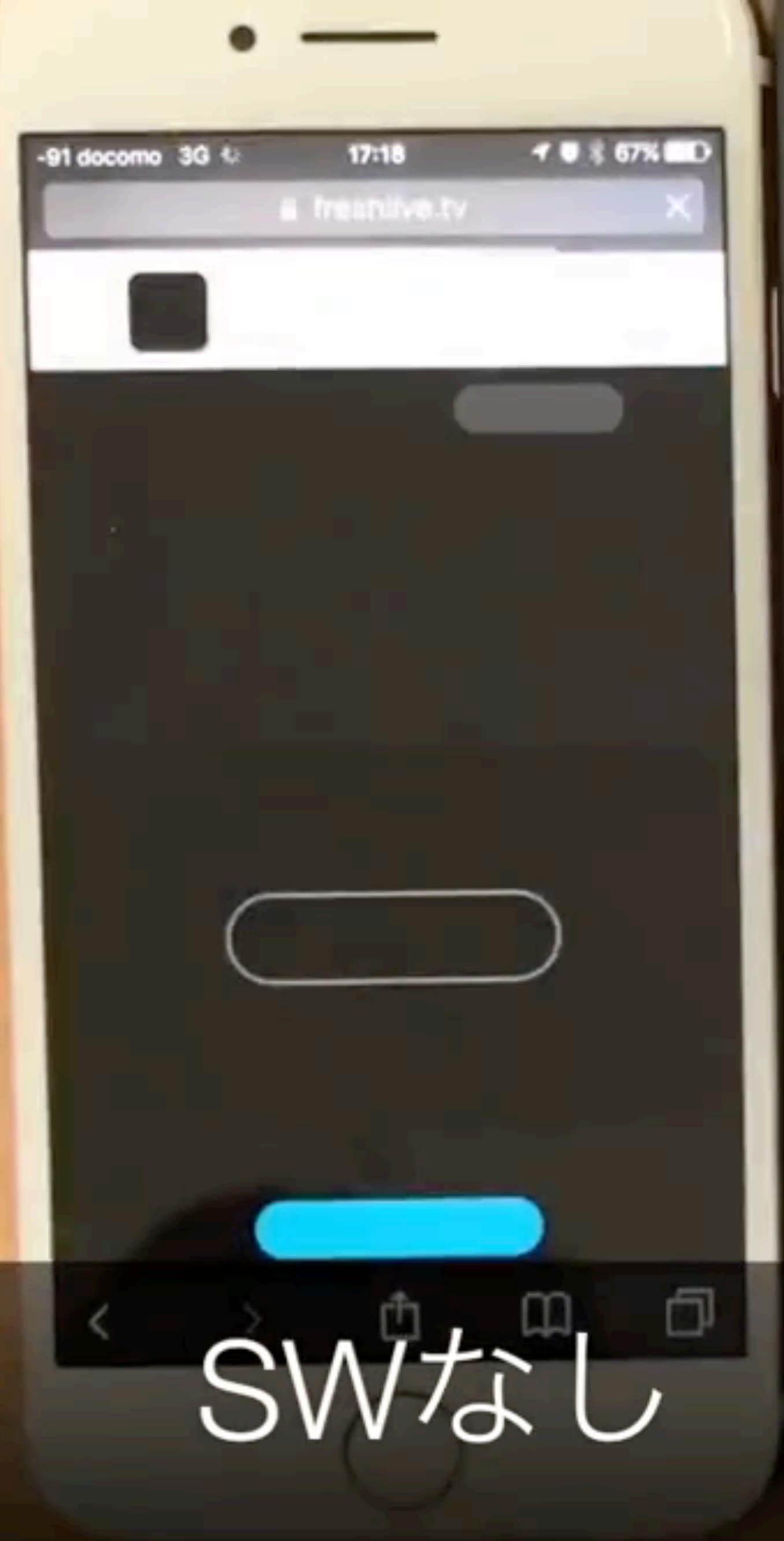
ランタイム

# ページロード

▶ ナビゲーションの開始からページが表示されるまでの速度

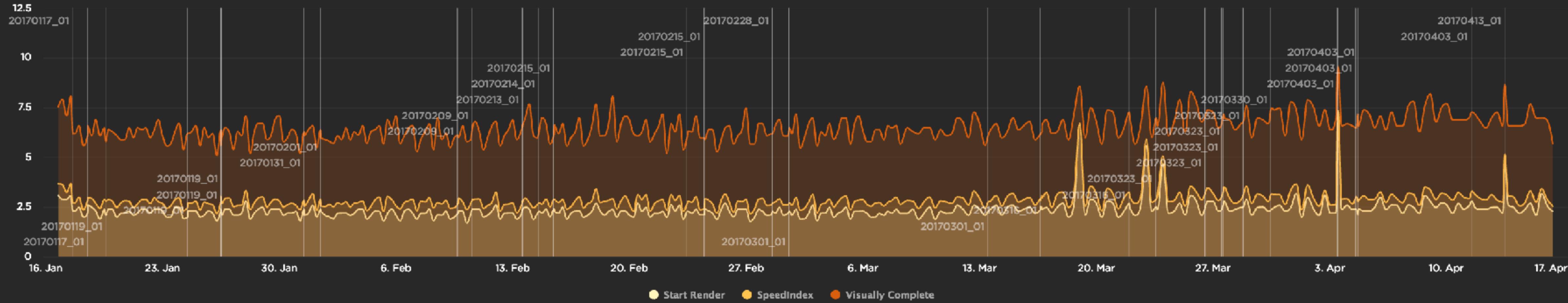
- クリティカルレンダリングパスの最適化
- 配信リソース(画像など)の数とファイルサイズの最適化
- HTML返却(≒サーバーサイド処理)の高速化 etc...

04:19



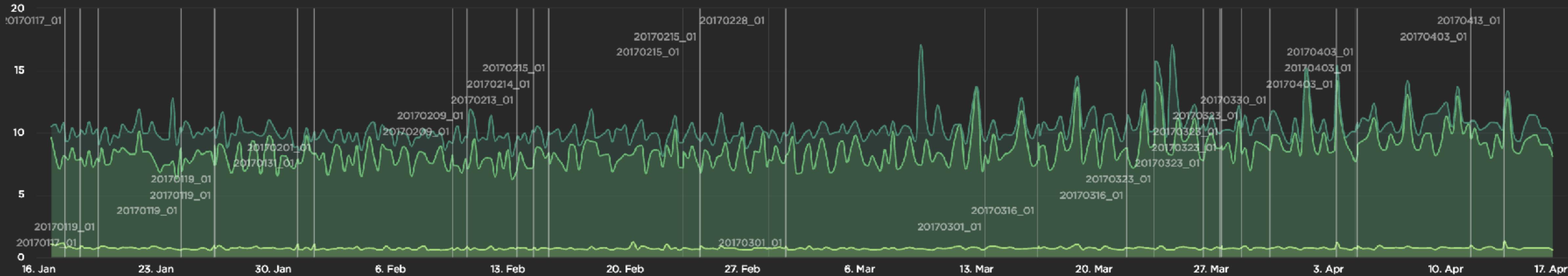
**Start Render**

**2.29s**



## Backend

0.71s



## SPEEDINDEX

FRESH!

**2.9s**

#2 38% slower

YouTube

**2.1s**

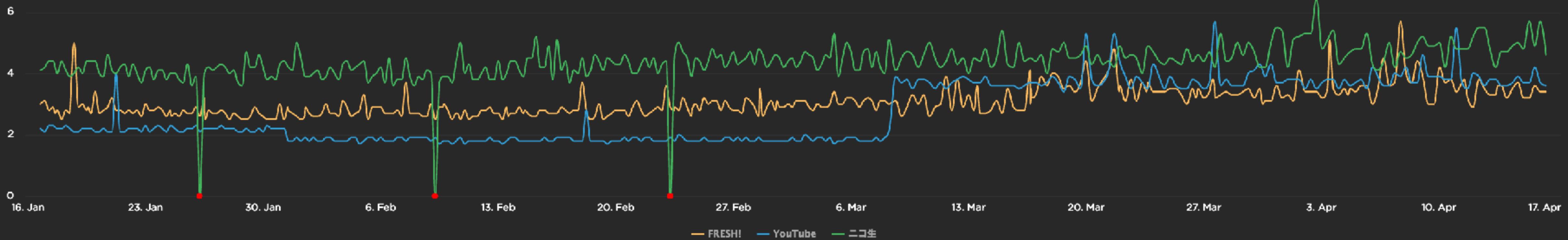
#1

ニコ生

**4.4s**

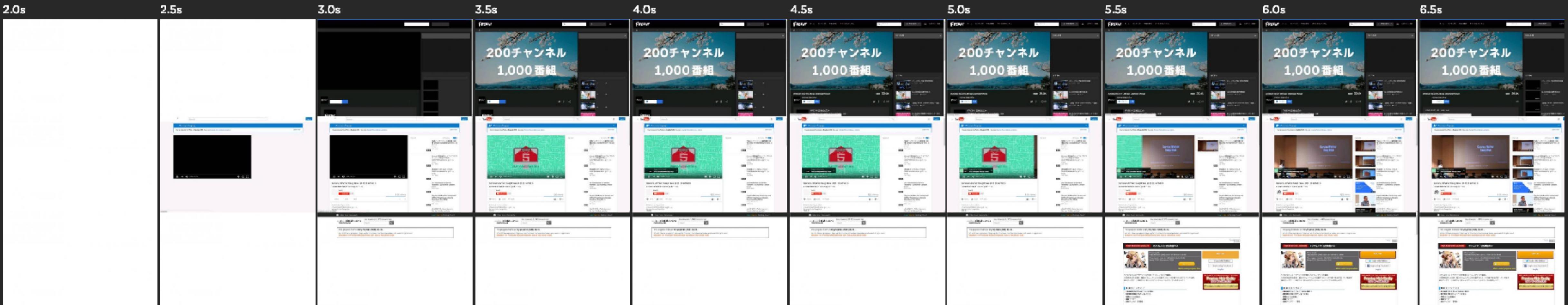
#3 110% slower

B



## FILMSTRIP

OPTIONS



# ランタイム

## ▶ ページが表示された後の UI の応答速度

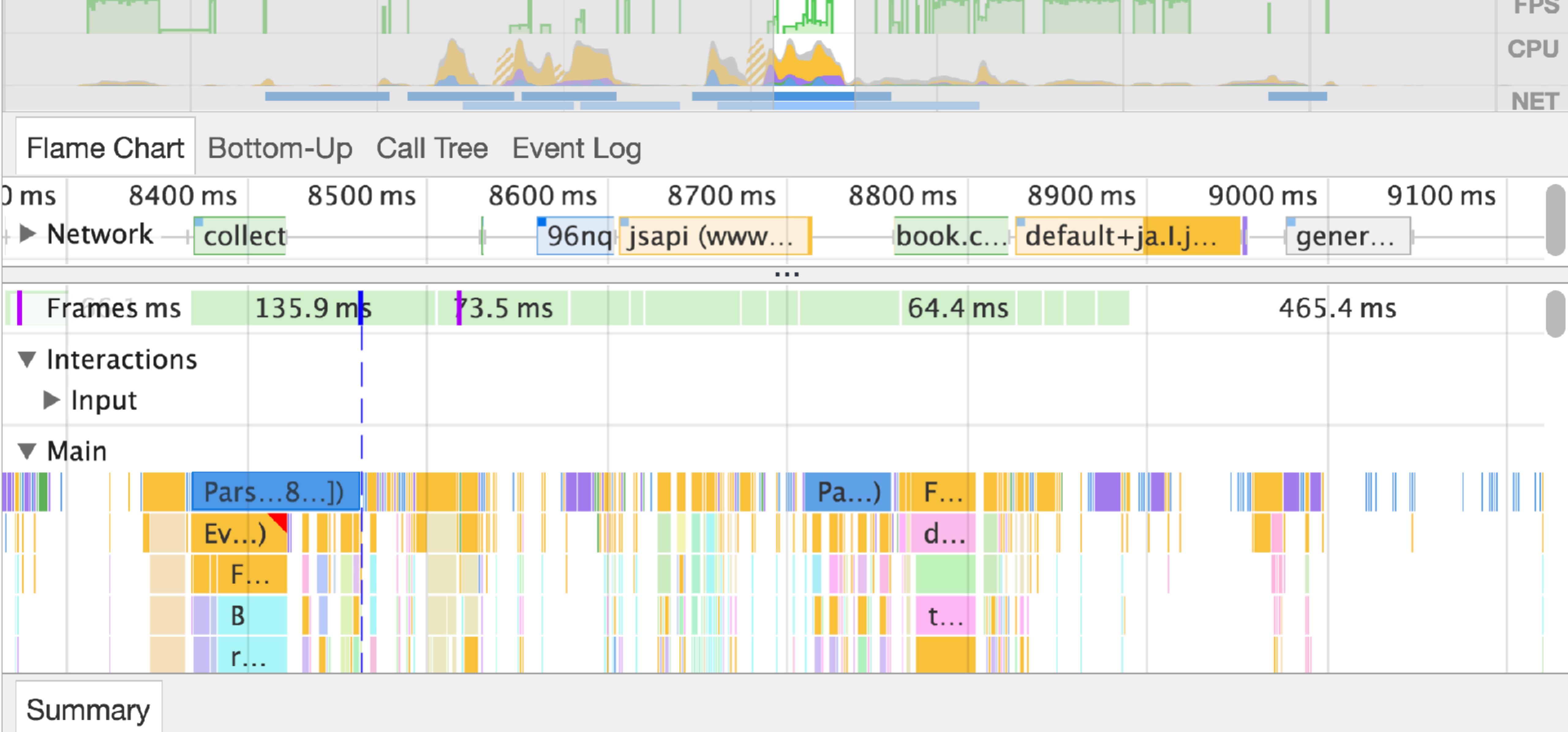
- FPS (1秒あたりの画面の更新回数) の最大化 or 反応の高速化
- グラフィック的なレンダリング負荷の軽量化
- 画面更新をさまたげるスクリプト処理の除去 etc...

**15**

**30**

**60**

**120**



16.1ms █ Loading (self)

1.0ms █ Loading (children)

65.4ms █ Scripting

# パフォーマンスを捉えるための計測・調査

- ▶ 改善するときは因果関係を正確に捉えることが最も重要
- ▶ 開発時のベストプラクティスの積み重ねも大事ではあるが…
- ▶ 合い言葉は「早すぎる最適化は諸悪の根源」
- ▶ 色々な計測・調査の方法を覚えることは真実に近づく訓練

どれくらいなら適切な応答速度と言えるのか？



UI のレスポンス

**100ms**

アニメーション1f

**10ms (16ms)**

アイドル中の1処理

**50ms**

ページロード

**1,000ms**

# なぜ Web ページは遅くなるのか？

- ▶ 誰も遅いものを作ろうとは思っていないはずだが…？
- ▶ 理想的な条件しか見ていない？
- ▶ 速くする方法がわからない、時間がない？
- ▶ そもそも関心が無い（趣味）？

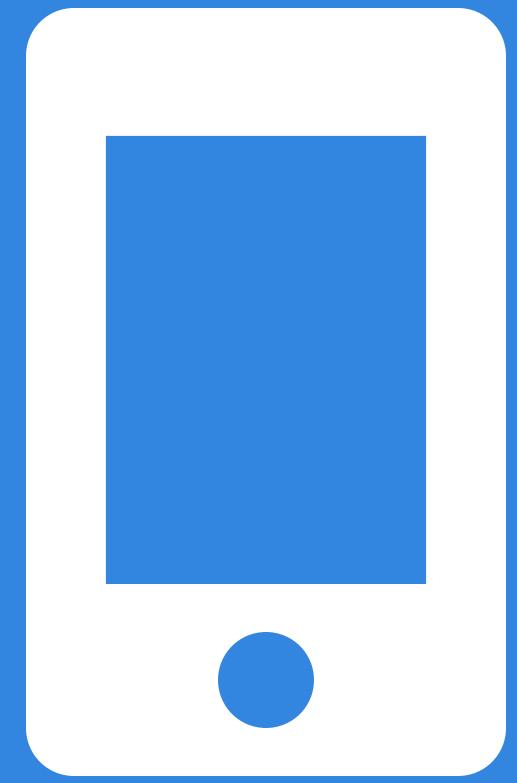
クライアント環境の多様性が  
甘く見られてしまっている



# パフォーマンスと クライアント環境の「多様性」

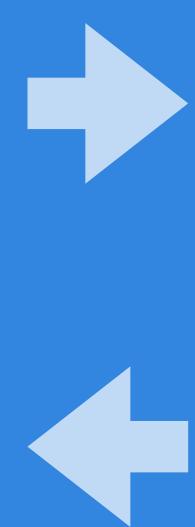
# クライアント環境の変数 (ほんの一部)

- ▶ デバイス (処理性能、スクリーンサイズ、操作方法)
- ▶ ブラウザ、OS (実行性能、ユーザーが追加した拡張機能)
- ▶ ネットワーク (オフィス、自宅、公衆Wifi、モバイル)
- ▶ ユーザー (年齢や個人差による認知特性の差異)



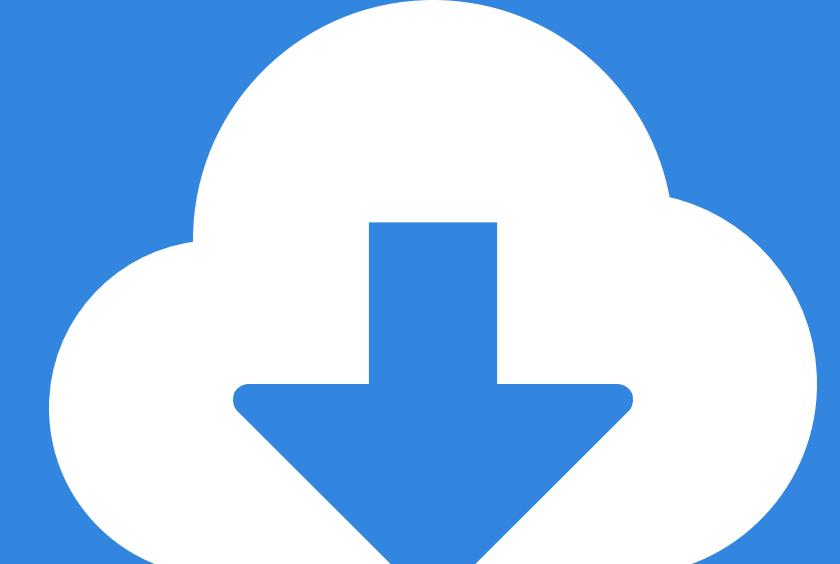
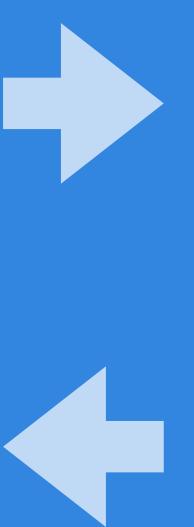
**Client**

スペックによる処理遅延



**Connection**

接続のレイテンシー



**Network**

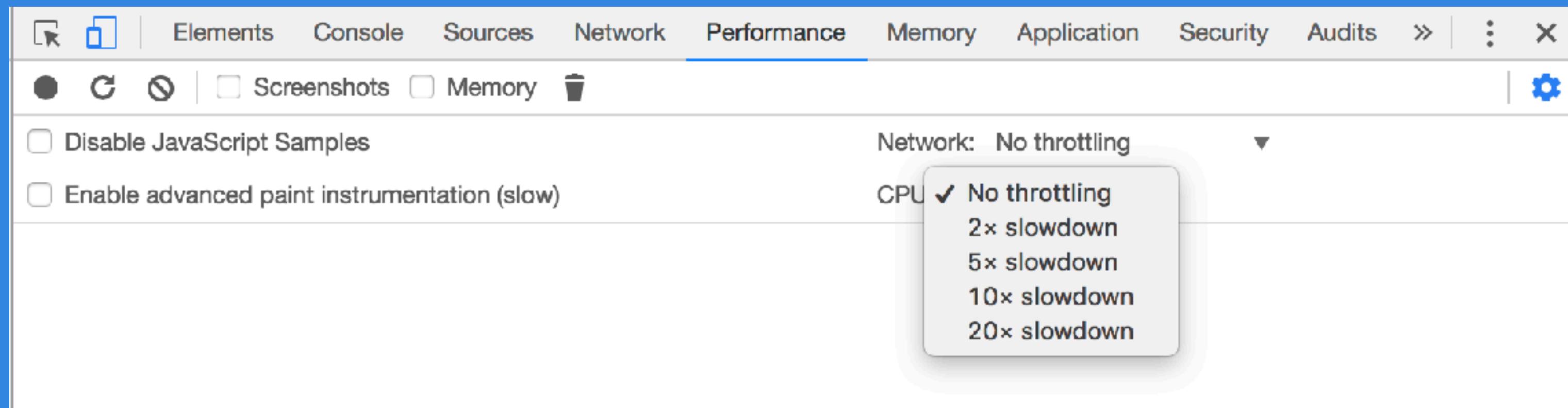
機器間のオーバーヘッド



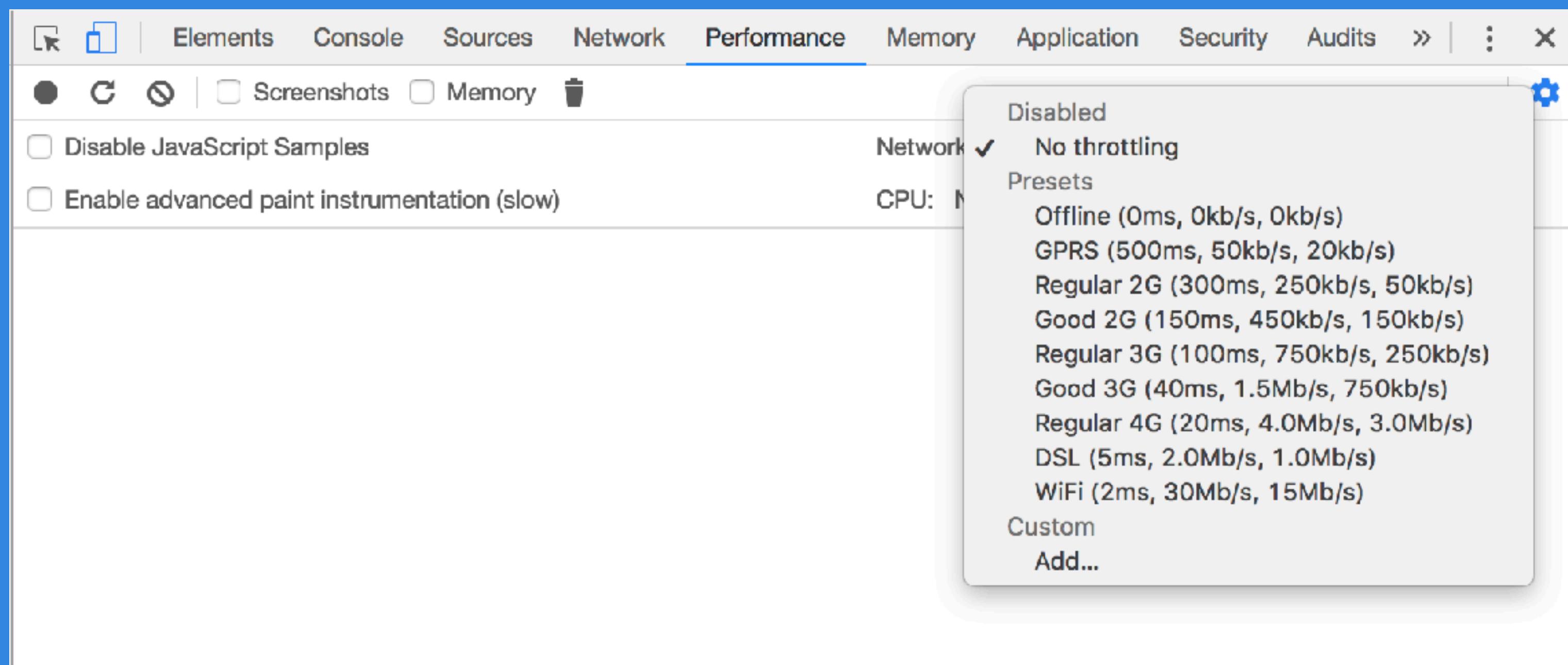
**Server**

データベースなどの遅延

## CPU のスロットリング（処理性能のエミュレート）

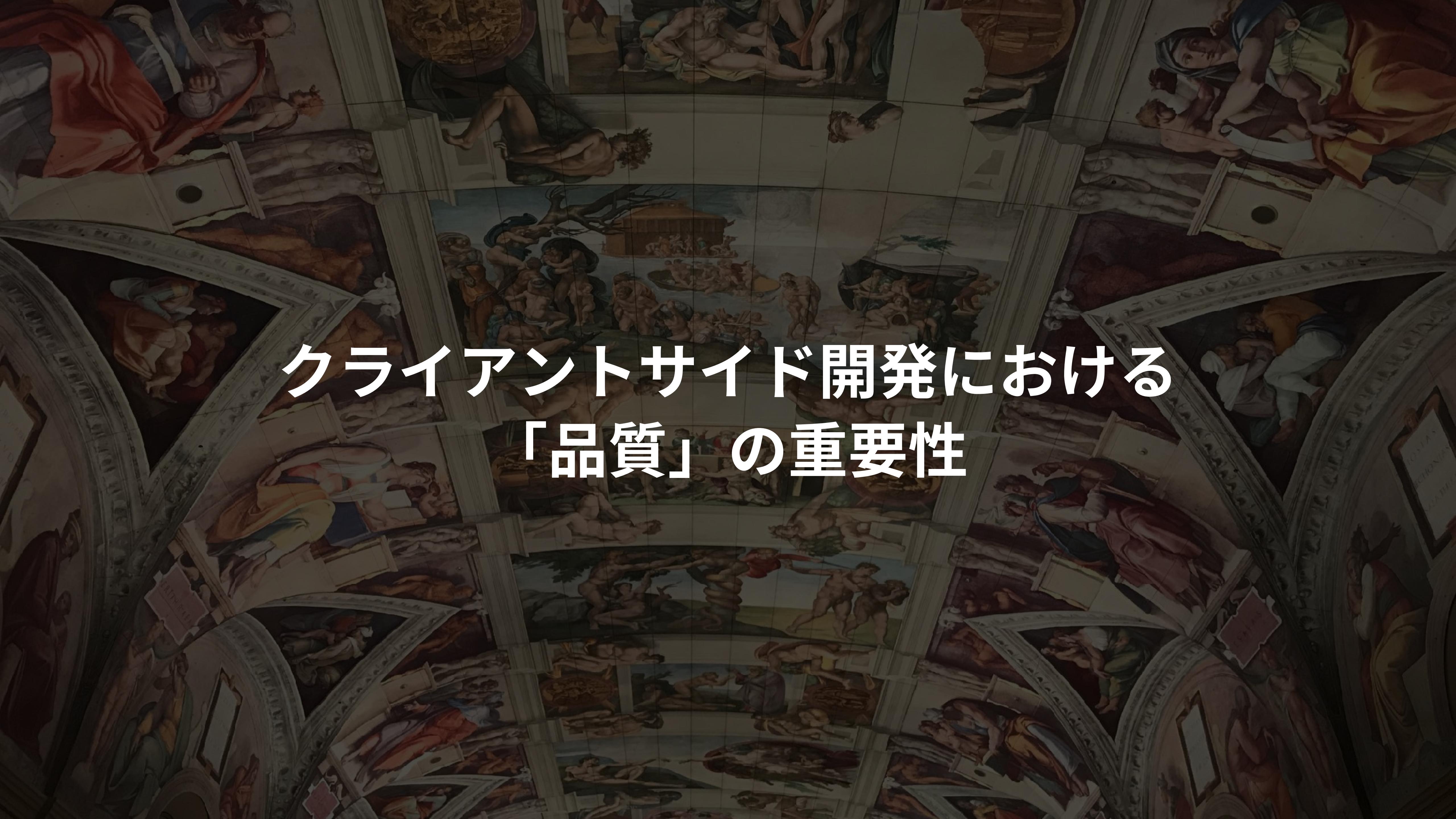


## ネットワークのスロットリング（帯域性能のエミュレート）



あらゆる多様性の中に  
“遅い”、“使えない”が潜む

”使えない”はユーザ一体験として  
サーバー障害と同じくらい深刻な問題



# クライアントサイド開発における 「品質」の重要性

目指すべきビジネスゴール

要求されるプロダクトの見た目、機能

プロダクトとして使いものになる品質



目先の UX (?)

理想



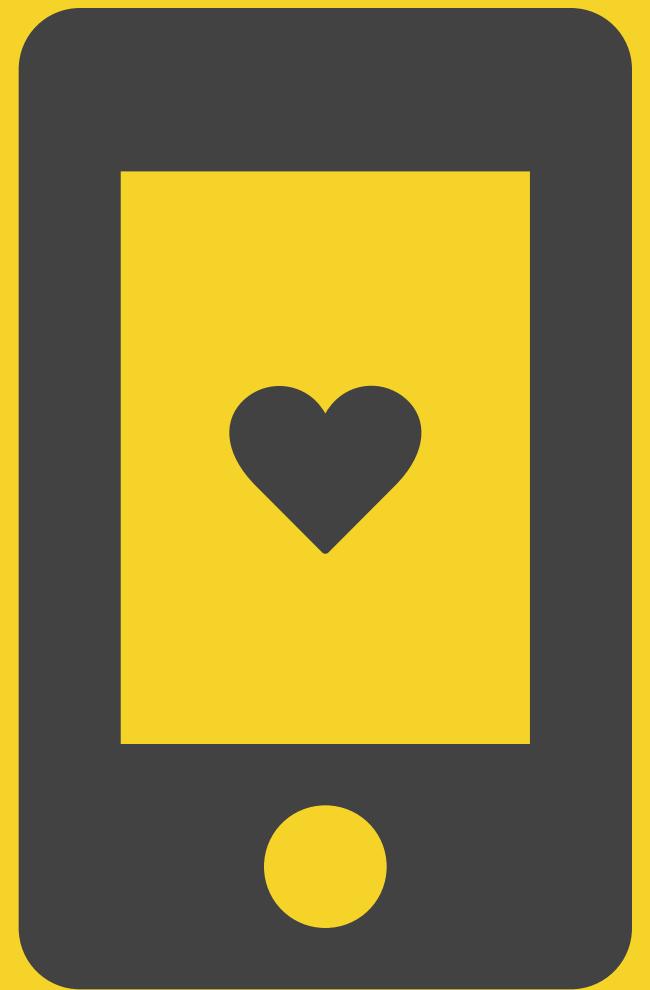
カッコいい

現実



クソ重い

理想



現実



画期的な UI

考慮不足で使いづらい

※開発者がHPを削って何とかしようとしたけど何ともならなかった



—人人人人人人人—  
> 最高の UX <  
—γ^γ^γ^γ^γ^γ^γ—

- ページがいつまでも開かない
- 開いたけどスクロールが重すぎる
- キーボードで操作できない
- どこが操作できるのか分からない
- なんかもうやだ

## 目指すべきビジネスゴール

Responsive Web Design

見た目、機能のクロスブラウザ

パフォーマンス

アクセシビリティ

セキュリティ

ユーザビリティ etc...

## 要求されるプロダクトの見た目、機能

## プロダクトとして使いものになる品質

# クライアントサイド開発の本質的な役割

- ▶ クライアント環境の多様性を十分に考慮して品質を高める
- ▶ 機会損失を減らし、**プロダクトの価値を最大化**する
  - 無限に IE 対応しようという意味ではない
  - もちろん理想はカッコ良くて使いやすくて速いこと ✨

まとめ

# 言いたかったこと

- ▶ パフォーマンス（とアクセシビリティとセキュリティ）は何よりも尊い
- ▶ 多様な環境でプロダクトの価値を担保するのが我々の本質
- ▶ 目先の UX(?) より 「ちゃんと使える」という体験がまず重要
- ▶ 出版されたら買ってください

# 新しい技術も品質を高めうる

- 新しい技術やライブラリはより効率的な方法の可能性
- 目的だけが大事なのではない
- 目的への自らの仕事の価値(品質)を最大化することも大事
- その為の手段もまた同様に大事なのでドンドンやろう

# Thank you ❤

🏠 <http://aho.mu>

🐦 [@ahomu](https://twitter.com/ahomu)

🐱 [github.com/ahomu](https://github.com/ahomu)