



アメブロ2016

実録、アメブロリニューアル275日

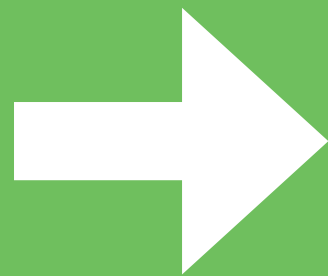
～ Webパフォーマンス編 ～

2017.2.25 Inside Frontend #1 @Yahoo
Kazunari Hara (@herablog)

アメブロ2016？

表示速度改善

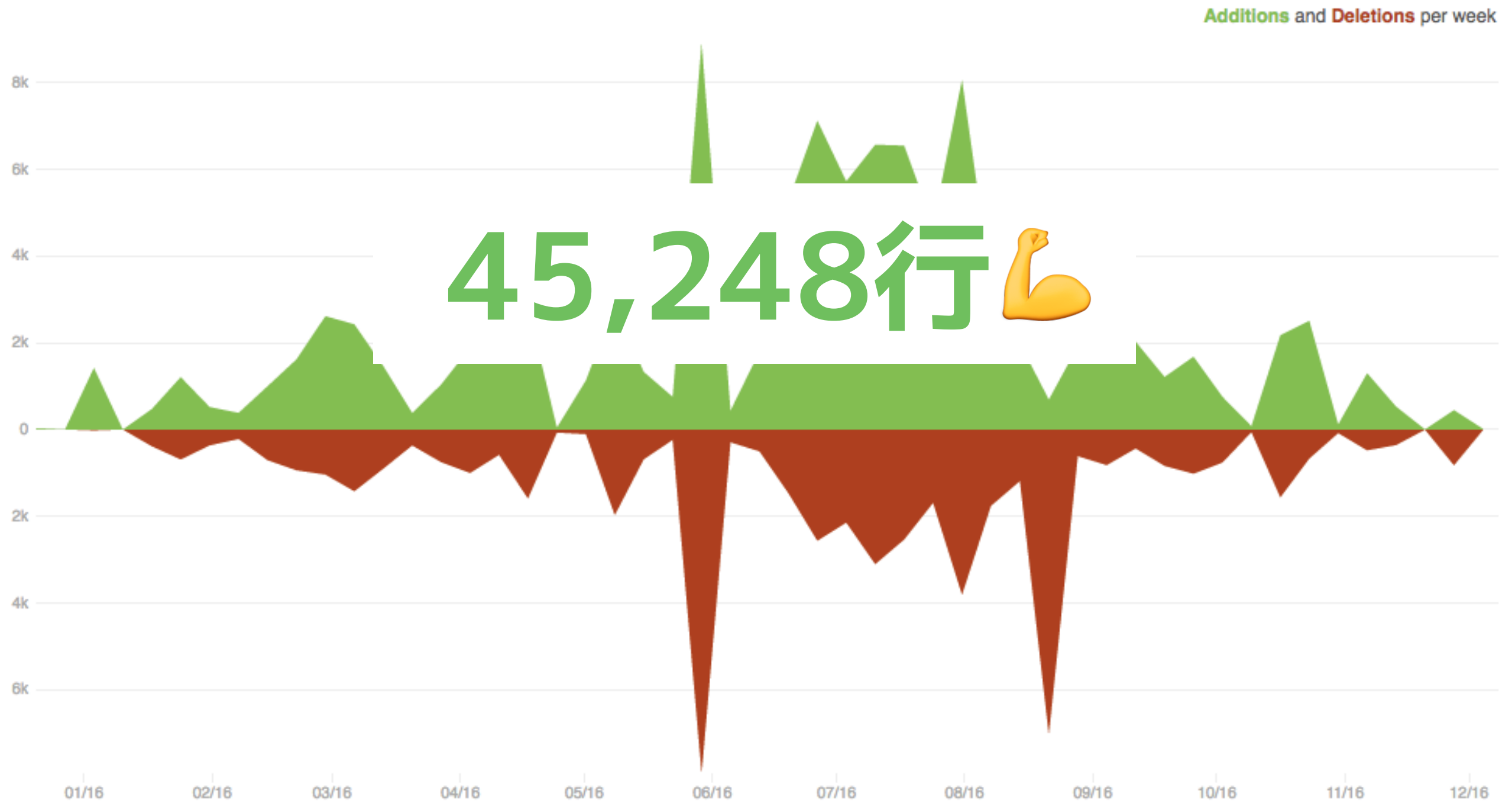
システム, UIのモダン化



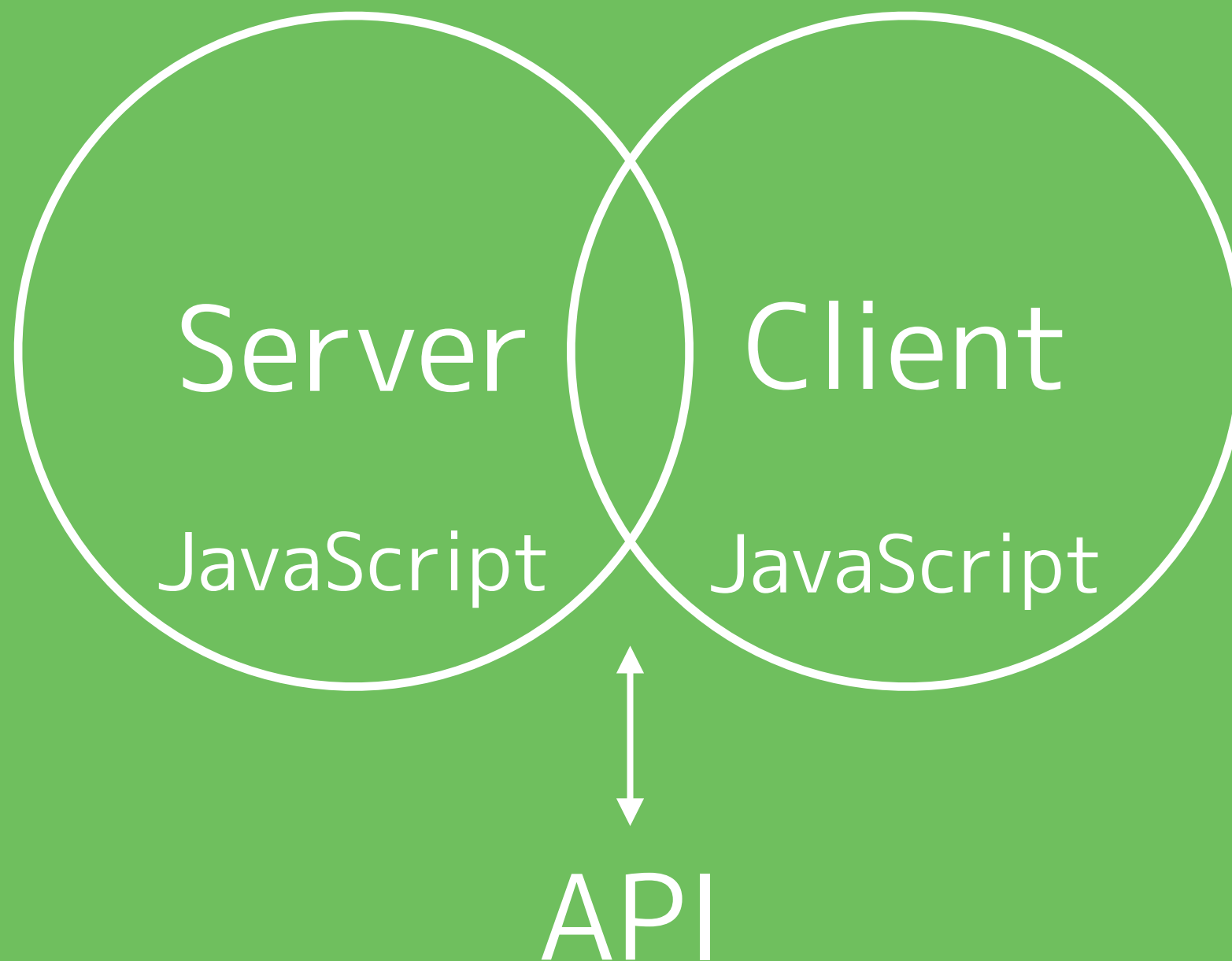
Amebaならではの
芸能人ブログ



アメブロ2016？



IsomorphicなWebで成果



IsomorphicなWebで成果

Pageviews

57.15%



Pages / Session

35.54%

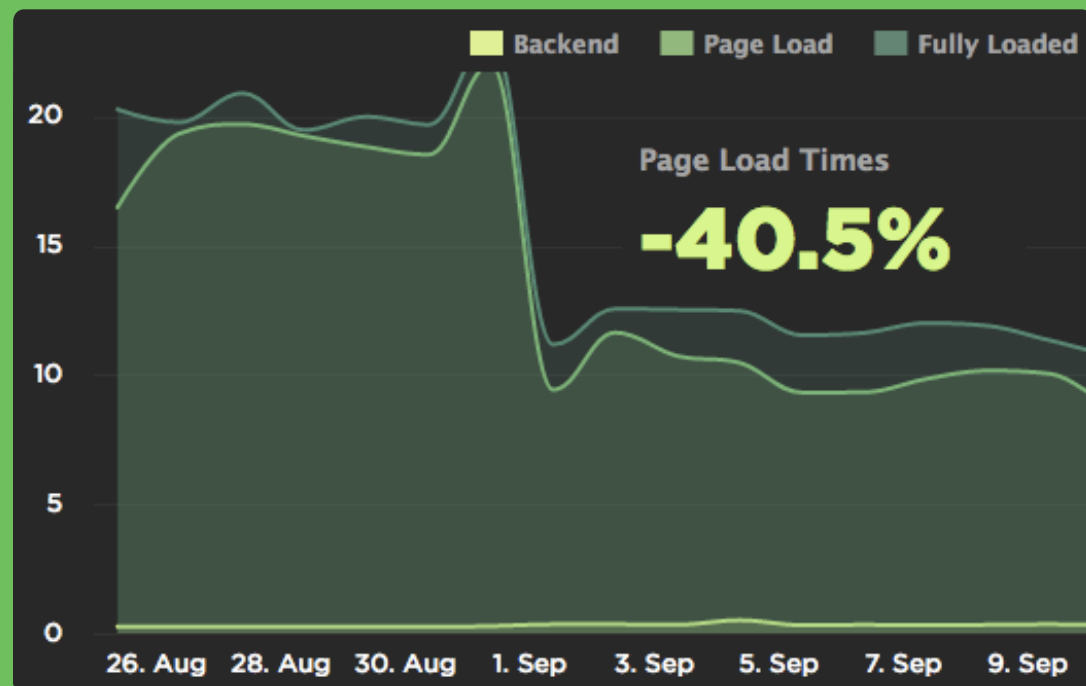
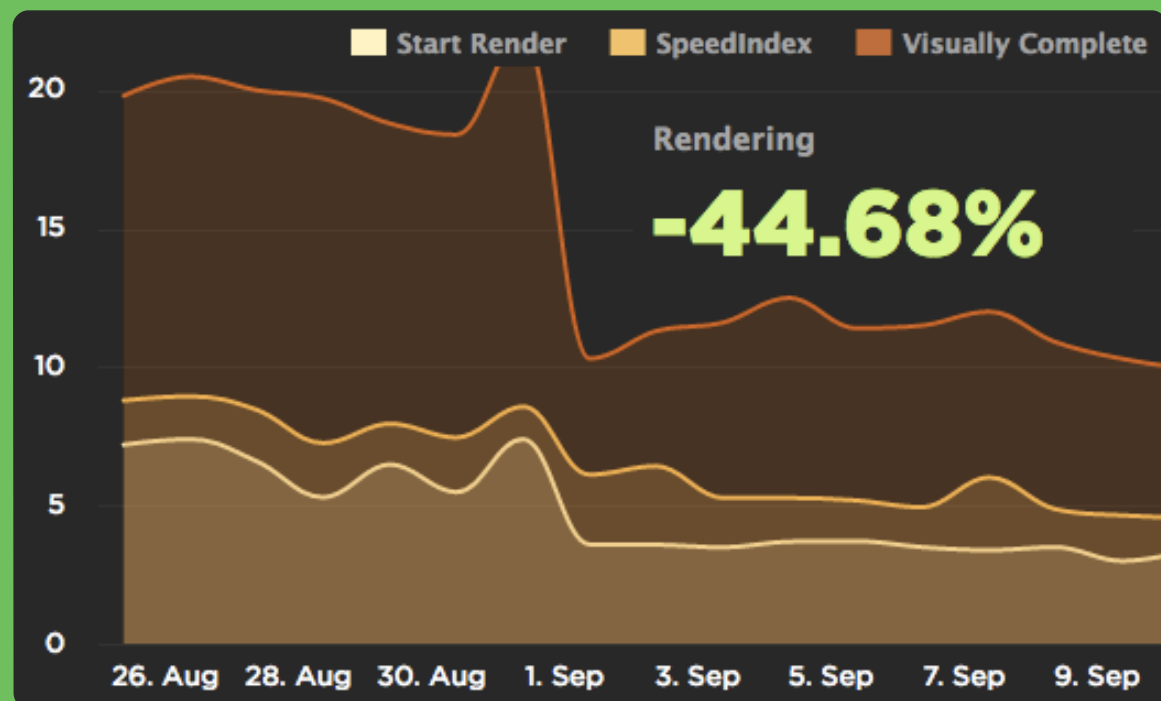
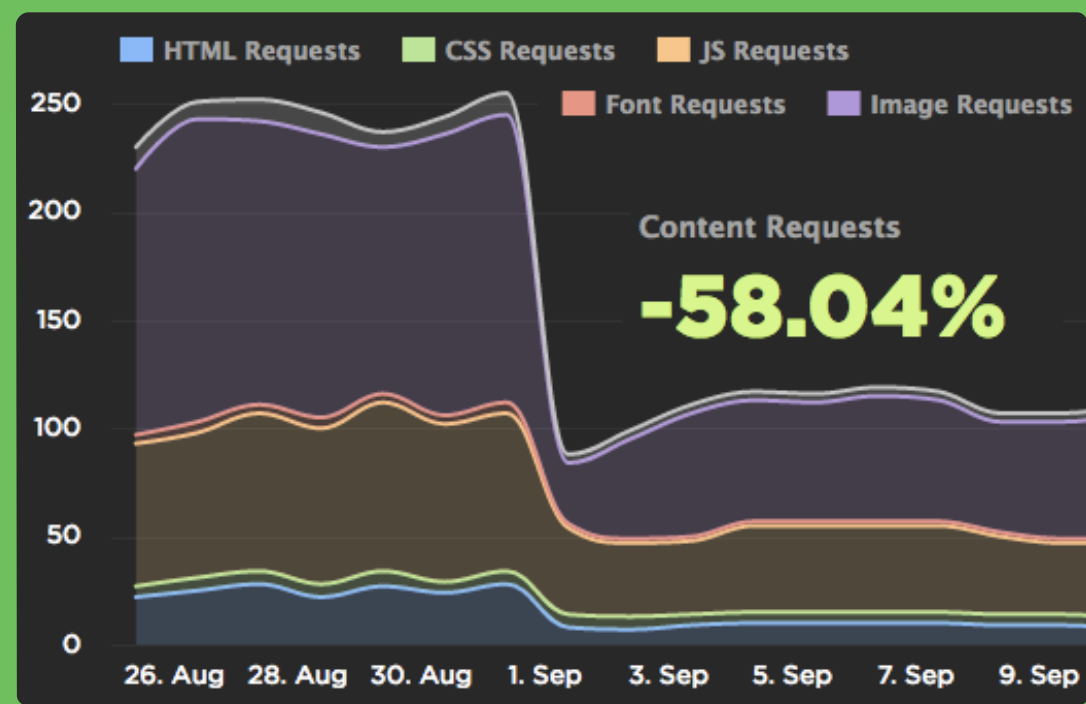
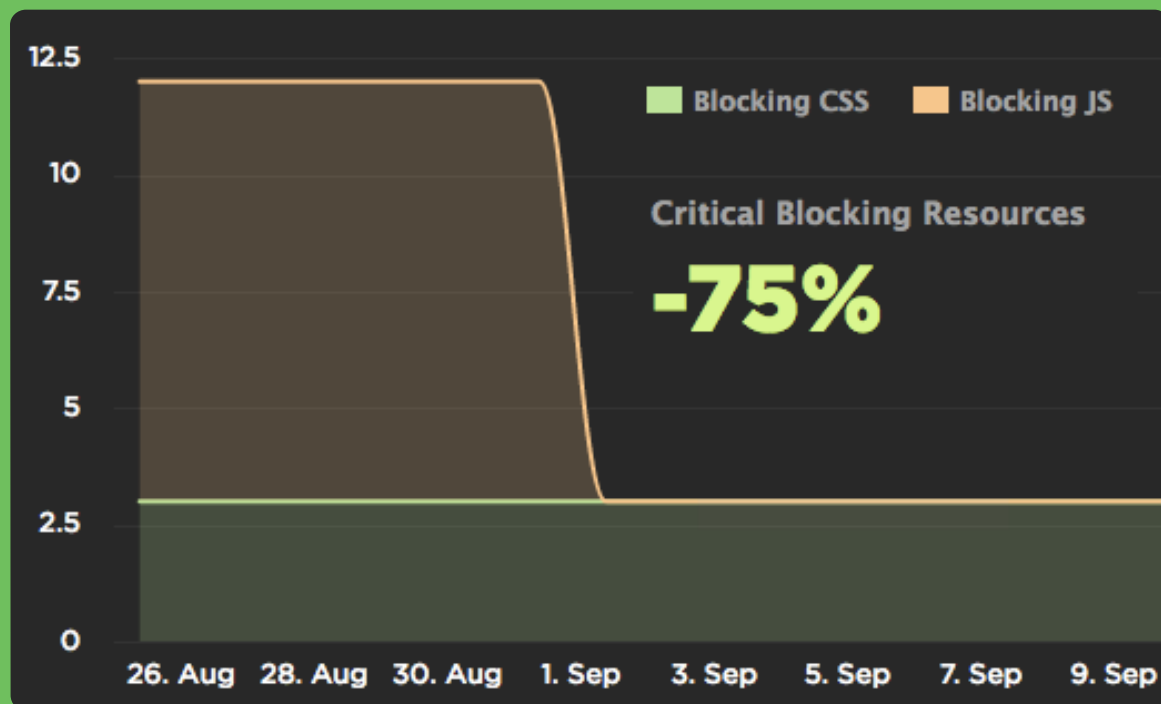


Bounce Rate

-44.44%



IsomorphicなWebで成果



採用技術はこんな感じ



BABEL



express



詳しくはブログを

アメブロ2016 ~ React/ReduxでつくるIsomorphic web app ~
<https://developers.cyberagent.co.jp/blog/archives/636/>

アメブロ2016 ~ Isomorphic JavaScriptの詳しい話
<https://developers.cyberagent.co.jp/blog/archives/3513/>

アメブロの大規模システム刷新と それを支えるSpring
<https://www.slideshare.net/nin2hanzo/spring-69237035>

アメブロ2016：インフラ編 ~大規模リニューアルの裏側~
<https://developers.cyberagent.co.jp/blog/archives/2653/>

Speaker

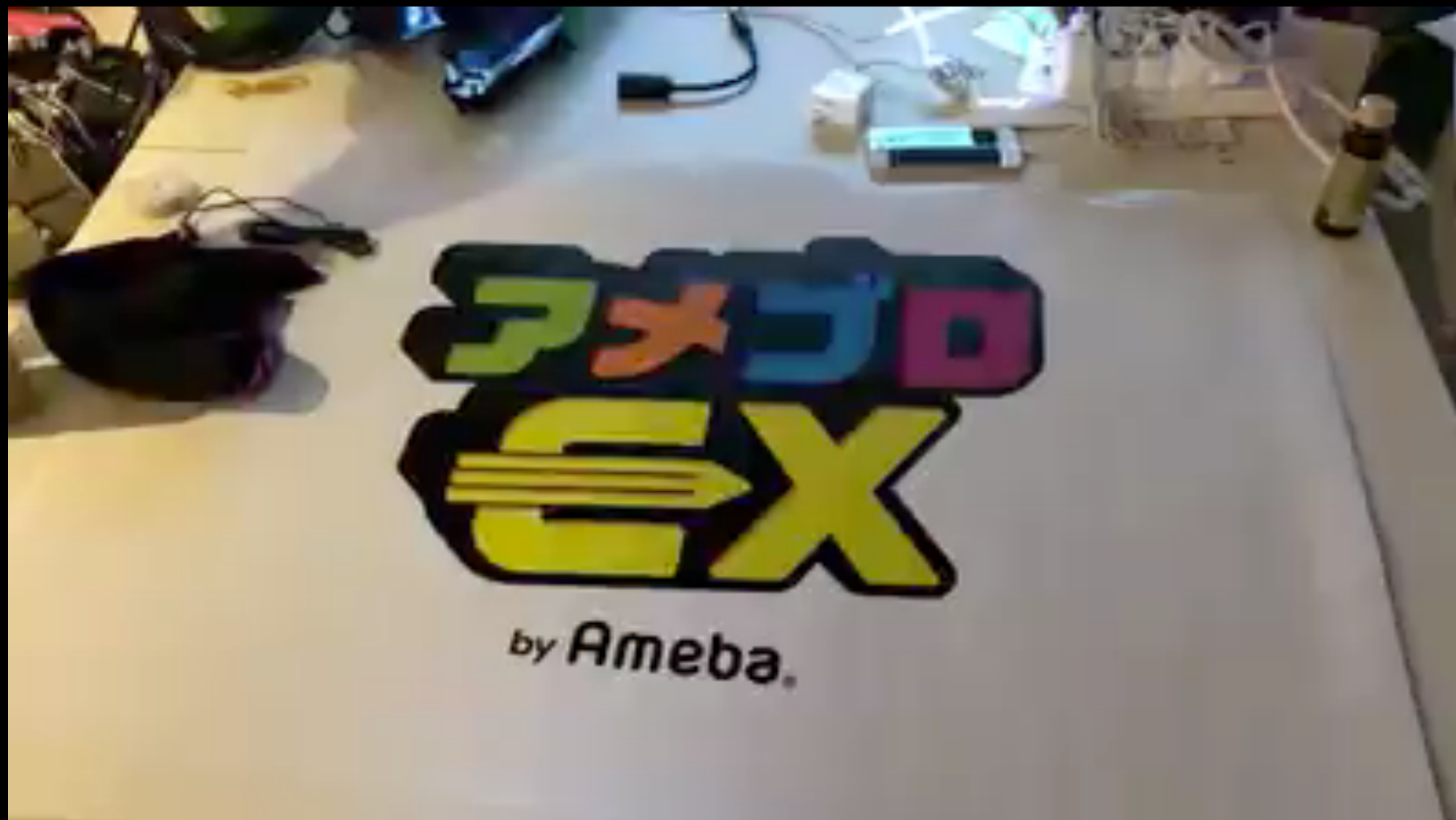


@herablog

Kazunari Hara

Shibuya Engineer

 **CyberAgent**®





アメブロ2016

実録、アメブロリニューアル275日

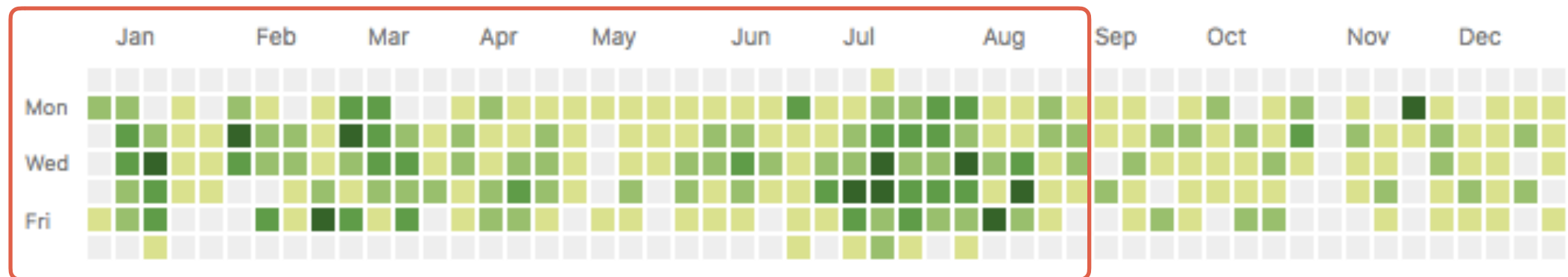
～ Webパフォーマンス編 ～

2016.1.1 ~ 2016.8.31

2016.1.1 ~ 2016.8.31

2,531 contributions in 2016

Contribution settings ▾



[Learn how we count contributions.](#)

Less More

Planning

Planning

GitHub Gist

Search...

All gists GitHub

New gist



hara-kazunari / ameblo_new_front.md Secret

Last active 6 months ago

Edit

Delete

Star

0

Code

Revisions 12

Embed

<script src="https://ghe.c



Download ZIP

アメブロ新システム構成

ameblo_new_front.md

Raw

コンセプト

- できるだけ普通(その時点で一番有名)のツールを使う
 - 情報が多い
 - 色んなひとが触れる
- フロントエンドサーバー
 - node
 - 表示に関わるところは誰でもさっと変えられるようにする
 - できるだけHTMLで返し、js, cssの容量減らす

The Standard

- エコシステムに乗っかる
- “Say hallo to the world”

Frontend Server (nodejs)

- 表示に関わるものをさっと変えられる
- 弊社数々のサービスで導入実績あり

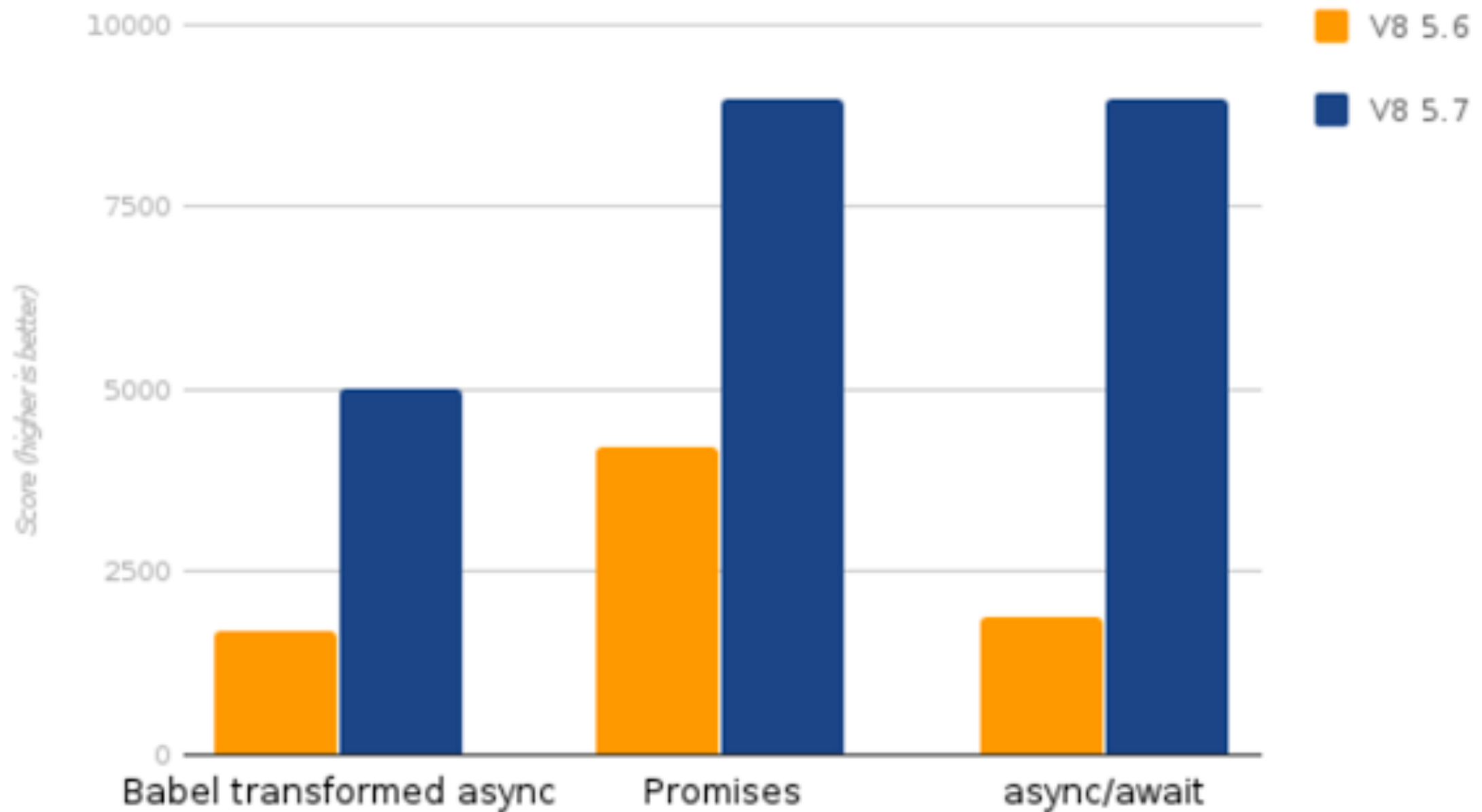


JavaScript

- React
- Flux -> Redux
- ES6(, 7)
- ESLint

V8 5.7

Async performance improvements in V8



mid-March 🤗

into it ?

 **targos** commented 16 days ago Member + 🗨️

@YurySolovyov 5.7 is still in beta phase. We will update to it when Chrome 57 is stable (should be mid-March)

👍 8

📖  **kevinsawicki** added a commit to electron/node that referenced this pull request 16 days ago

🔑   Bump NODE_MODULE_VERSION to 53 ... b649082

🏷️  **targos** removed the **build** label 15 days ago

 **targos** commented 15 days ago Member + 🗨️

ping @nodejs/v8 for the Windows build issue.

Basically the problem is that `deps\v8\src\v8_base_0.vcxproj` instructs to look for the inspector generated source files in `..\..\..\build\Release\obj\global_intermediate\src` but they are in fact in `..\..\..\Release\obj\global_intermediate\src`.

CSS

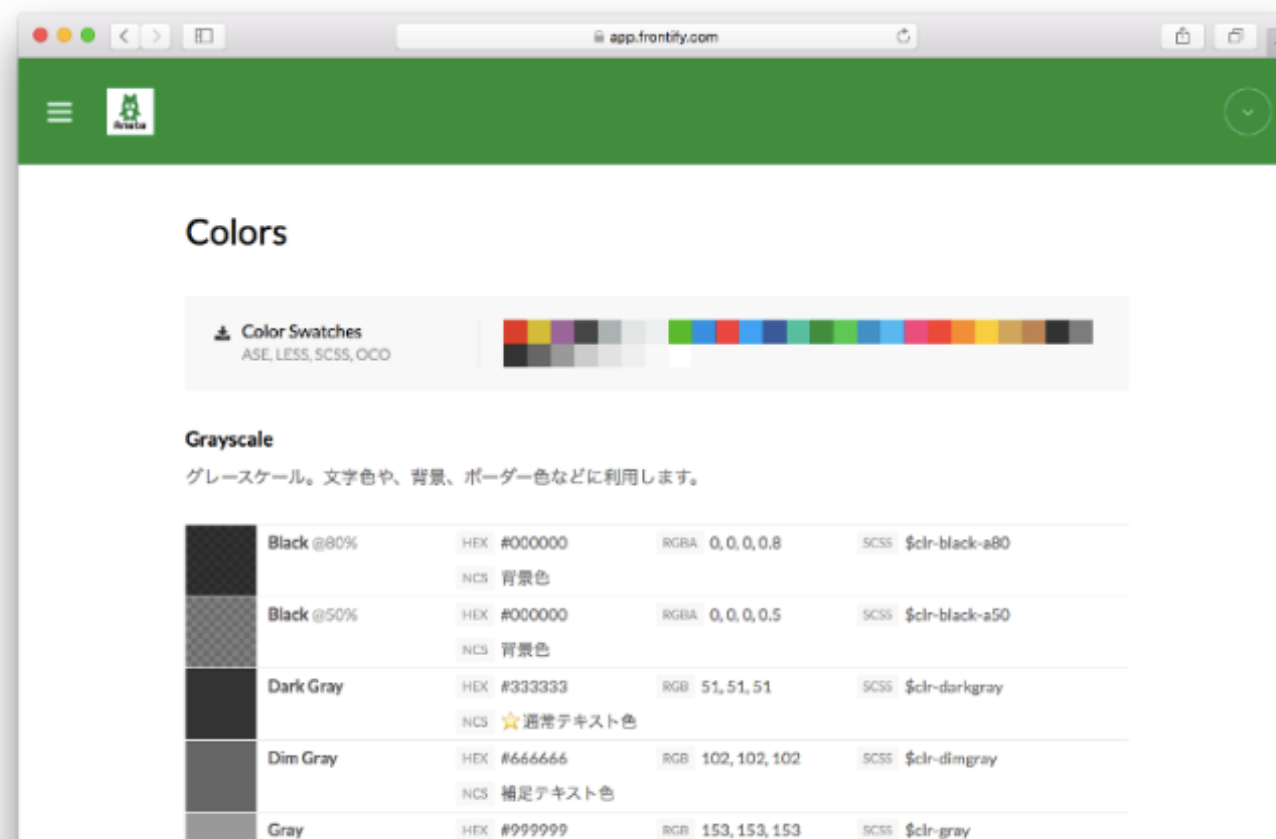
- PostCSS
- BEM -> CSS Modules
- stylelint
- Style guide -> Web

Style guide

Making Style Guides On The Web

We decided to make style guides on the web by using Frontify. Web services are more useful for **everyone** (designers, engineers, directors etc). It's easier to update and see than Sketch or generators.

SHARE



HTML

- モダンなMETA
- アクセシビリティ
- AMP

モダンなMETA



SlideShare

Search



Upload

Login

Signup

[Home](#)

Technology

Education

More Topics

For Uploaders

Be the first to clip this slide

モダンなMETA

```
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="keywords" content="">
<meta name="description" content="">
<meta property="fb:app_id" content="311629842256842" />
<meta property="og:locale" content="ja_JP" />
<meta property="og:title" content="「AWA」のデザインは..." />
<meta property="og:type" content="article" />
<meta property="og:url" content="http://ameblo..." />
<meta property="og:image" content="http://...png" />
<meta property="og:site_name" content="1 pixel" />
<meta property="og:description" content="こんにちは。AWAで..." />
<meta property="al:ios:url" content="jpameblo://ameblo/ca-1pixel/entry/12042291572" />
<meta property="al:ios:app_store_id" content="34942137" />
<meta property="al:ios:app_name" content="Ameba" />
<meta property="al:android:url" content="jpameblo://ameblo/ca-1pixel/entry/12042291572" />
<meta property="al:android:package" content="jp.ameba" />
<meta property="al:android:app_name" content="Ameba" />
<meta name="twitter:card" content="summary" />
<meta name="twitter:url" content="http://ameblo.jp/ca-1pixel/entry-12042291572.html" />
<meta name="twitter:title" content="「AWA」のデザインはどのようにして作られたのか。" />
<meta name="twitter:description" content="こんにちは。AWAで..." />
<meta name="twitter:image" content="http://stat.ameba.jp/user_images/20150622/10/ca-1pixel/52/38/p/o0600042013344659424.png" />
<meta name="twitter:app:country" content="JP" />
<meta name="twitter:app:name:iphone" content="Ameba" />
<meta name="twitter:app:id:iphone" content="34942137" />
<meta name="twitter:app:url:iphone" content="jpameblo://ameblo/ca-1pixel/entry/12042291572" />
<meta name="twitter:app:name:googleplay" content="Ameba" />
<meta name="twitter:app:id:googleplay" content="jp.ameba" />
<meta name="twitter:app:url:googleplay" content="jpameblo://ameblo/ca-1pixel/entry/12042291572" />
```

Photo by <http://matomo.nyu.edu/id/212748513458371593421275355045343030303>

24 of 26

Recommended



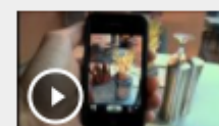
Up and Running with Twitter

lynda.com PREMIUM VIDEO



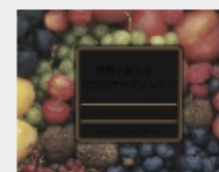
Up and Running with Pinterest

lynda.com PREMIUM VIDEO



Sharing Photos with Instagram

lynda.com PREMIUM VIDEO



CSS3 Design Recipe

Kazunari Hara

俺の、プレゼン構築法

Kazunari Hara

CA流 フロントエンドエンジ

アクセシビリティ (最初の一歩)



Speaker Deck

Search...

[Sign Up](#) [Sign In](#)



Kazunari Hara
5 Presentations

★ [Star this Talk](#) 1 Star

[Published in](#) [Technology](#)

[Stats](#) 7,910 Views

Share

[Twitter, Facebook](#)

[Embed](#)

[Direct Link](#)

[Download PDF](#)

AMP



川口春奈の「結婚しました」報告にネット騒然！その真相とは？ - シネマトゥデイ
www.cinematoday.jp ・ ... ・ 11月 ・ 24日

● AMP - 2016/11/24 - 女優の川口春奈（21）が21日、「結婚しました」と題してブログを更新し、ファンを驚かせた。



結婚しました | 川口春奈オフィシャルブログ
Powered by Ameba
ameblo.jp/entry-12221491370

● AMP - 2016/11/21 - やっとやっとてっちゃんと結婚式を、挙げる事が出来ました ... 川口春奈さんの読者になろう。ブログの更新情報が受け取れ ...

川口春奈、「結婚」をブログで報告 気になる相手はCM共演の「てっちゃん」/芸能 ...

AMP



Accelerated
Mobile
Pages
Project

2016/06/22

アメブロにおけるAMP

Colmun, Engineering, Web

Build

- gulp -> Webpack
- package.json

Performance Auditing

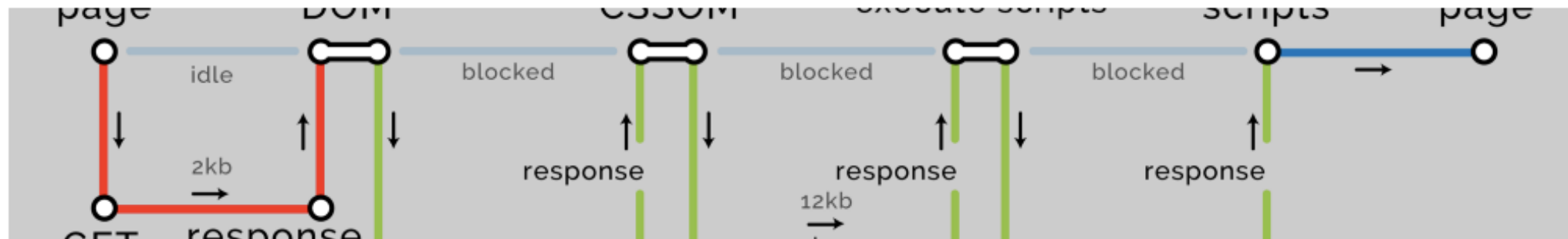
Browserの動きを理解する

- Critical Lengths
- Critical Files
- Critical Bytes

Browserの動きを理解する



Sanjay Purswani [Follow](#)
Software Engineer at comparethemarket.com
Feb 9 · 17 min read



Optimising the front end for the browser

Optimisation is all about speed and satisfaction.



201



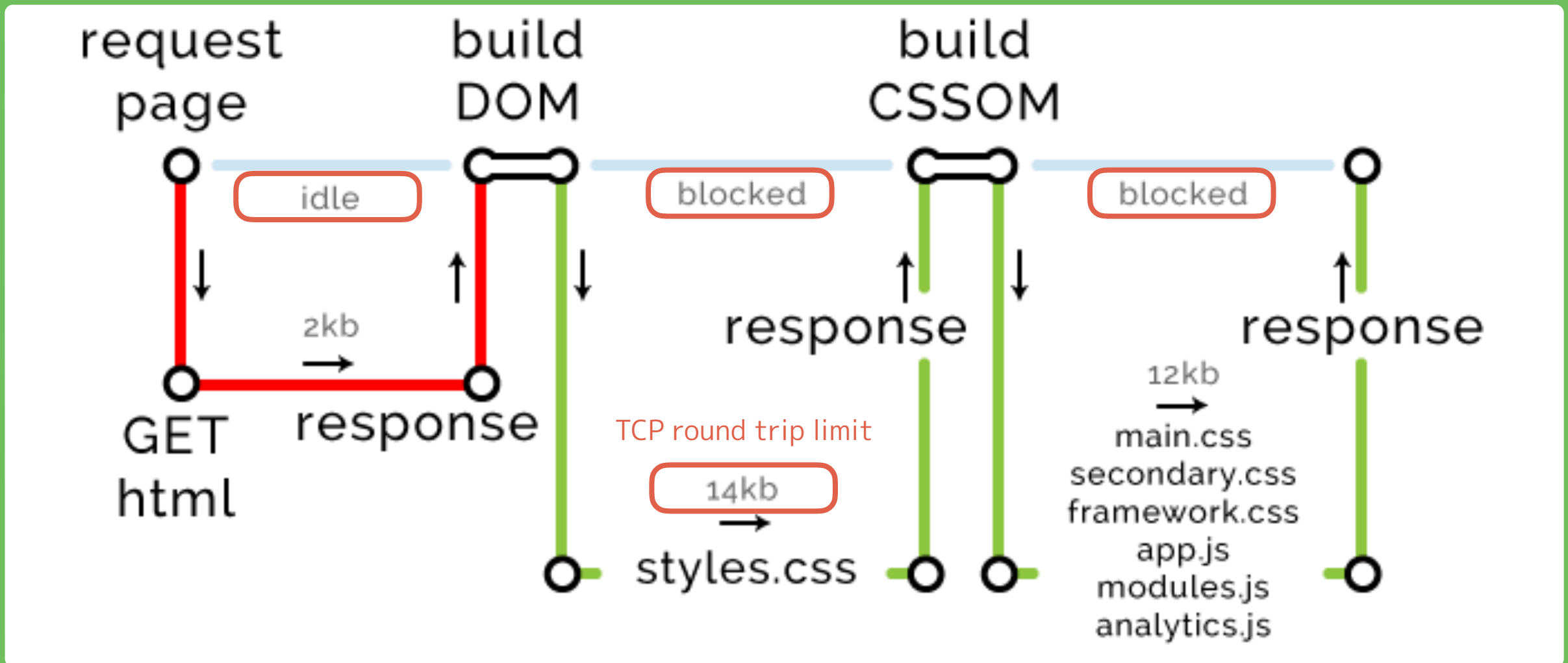
7



[Next story](#)
[Server Side Rendering with Create..](#)

Browserの動きを理解する

CRP Metrics: Length 3, Files 8, Bytes 28kb



Basics

Make your mobile pages render in under one second

- Keep **server backend time** to generate HTML to a minimum (**under 100ms**)
- **Avoid HTTP redirects** for the main HTML resource
- **Avoid loading blocking** external JavaScript and CSS before the initial render
- **Inline** just the JavaScript and CSS needed **for the initial render**
- **Delay or async load any JavaScript and CSS** not needed for the initial render
- Keep **HTML payload** needed to render initial content to **under 15kB compressed**

Performance Auditing with



Chrome Developer Tools



SpeedCurve

etc.

Performance Auditing

- 自分たちで調査・課題設定すること
- 継続できること



GUARDIAN BETA AND ARTICLE IN APPLE IPAD IN IRELAND LAST 1 MONTH ▼



RENDERING

Notes

Start Render

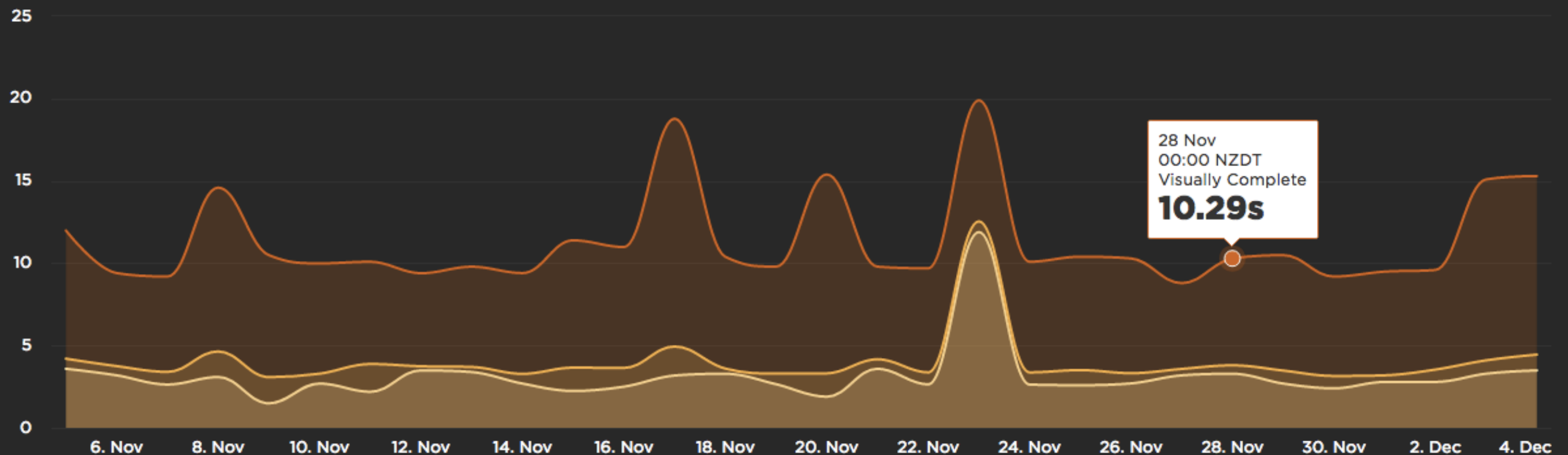
2.7s

SpeedIndex

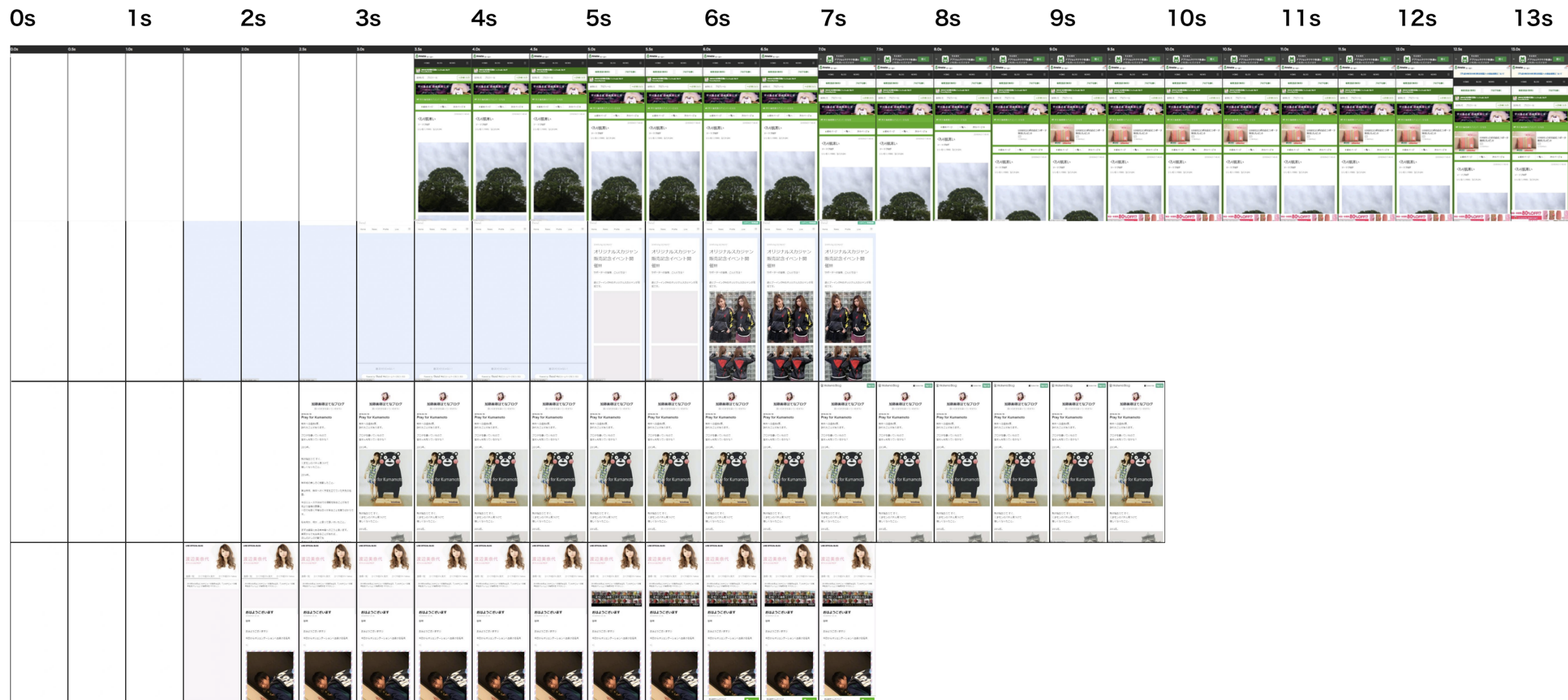
3.58s

Visually Complete

10.09s



Performance Auditing



Performance Auditing

- HTMLファイルサイズを減らす (71kB)
- Blocking Resourcesを減らす (8つ)
- JavaScriptの非同期読み込み
- CDNキャッシュの最適化
- Web Fontの統合 (3種類)
- いらない読み込みの削除
- 画像のCSS化

Goal

50% ↓

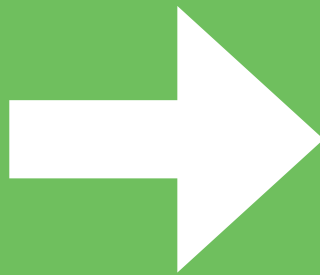
Architecture

Isomorphic Web App (SSR + SPA)

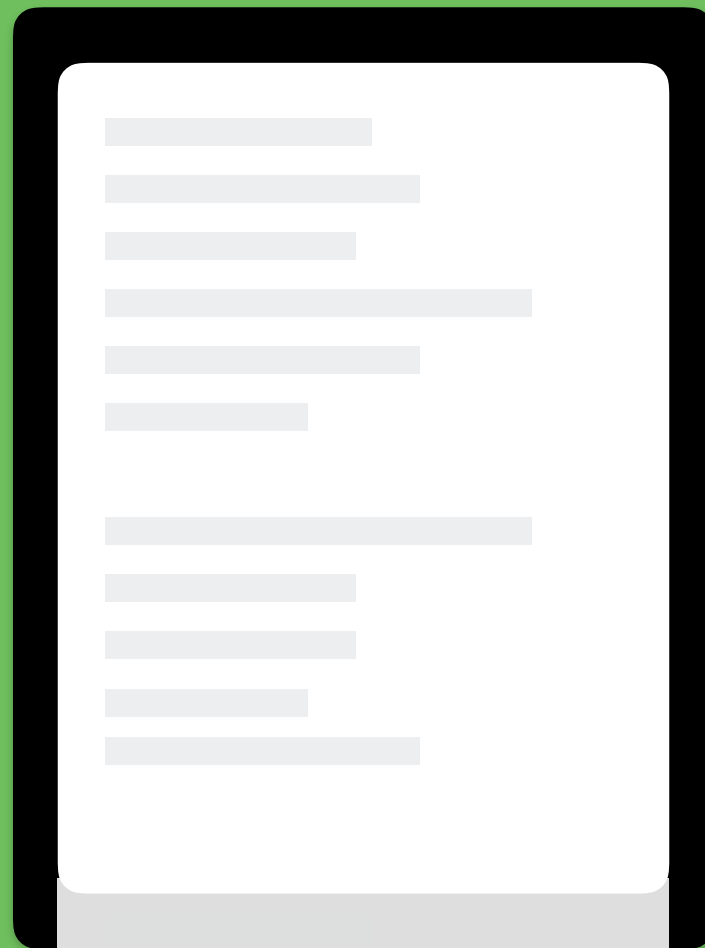
SSR

SPA

SPA



Lazy Load



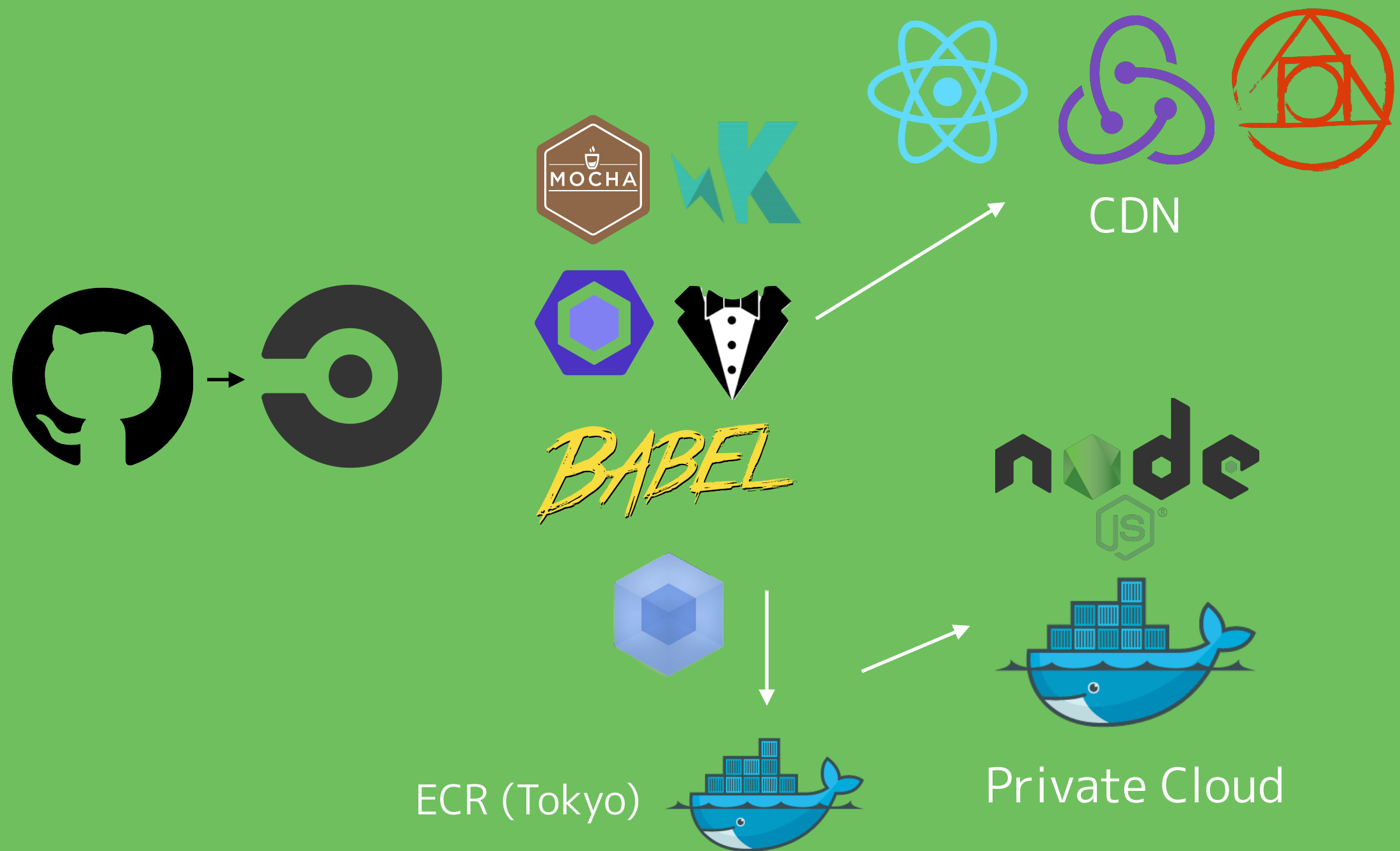
SSR

JS

Modern Workflow

- git-flow, Pull Request
- CIによる自動テスト, ビルド, デプロイ

Modern Workflow



Update Docs: README

tests.webpack.js

fix lint errors

6 months ago

README.md

tama-web

circleci passing

Ameba Blog Frontend:Next

What's the tama-web?

このリポジトリは、アメブロのフロントエンドアプリケーション、通称 `tama-web` です (☆∀☆)..

近年、モバイルの利用が増加し、Webアプリケーションへの要求も変化してきました。モバイルでは多くの人がネイティブアプリに慣れていて、タッチスクリーンを前提としたシンプルなデザインやストレスのない早い読み込み・表示が求められています。

Webでも同様にそれらを実現するためには、従来のサーバーサイドとクライアントサイドという垣根を越えて、トータルでのユーザー体験を考慮したWebアプリケーションを構築する必要があります。

このような状況を踏まえ、ブログのフロントエンドシステム刷新では以下の点の改善を目的としています。

一貫した開発 よいWebアプリケーションをつくるには、細分化された様々な技術を統合し、より効率的に開発しなければなりません。それぞれがバラバラな状態では技術的・時間的ロスが大きくなってしまいます。tama-webでは、IsomorphicなJavaScript(後述します)でサーバー・フロントサイドのコードを共有することや複数のリポジトリに分かれていたJavaScript, CSSを統合すること、staticファイルの自動バージョンング・デプロイなどを通じて開発効率の向上をはかります。そのうえで、ユーザー体験に責任をもっ

Update Docs: CONTRIBUTING.md

51 lines (35 sloc) | 2.88 KB

Raw

Blame

History



Contributing to tama-web

このプロジェクトに変更を加える際には、まず最初にこのドキュメントに目を通してください。

Pull Request Process

tama-webでは、ほとんど[git-flow](#)を採用しています。Forkはしない方式なので注意をしてください。

JIRA or GHE issue

JIRAチケット or GHE issue を確認します。アメプロでは基本的にJIRAを利用していますが、開発だけで完了する場合はissueだけでOKです。

Branch

作業ブランチを切り作業をします。ブランチ名には特に決まりはありませんが、通例 `feature/` , `fix/` から始めます。

Commit

作業が終わったらコミットをします。コミットメッセージは日本語でも英語でもかまいません。変更した内容がわかりやすいメッセージを記入してください。コミットメッセージにはチケット番号やissue番号を紐付けておくとよいです。

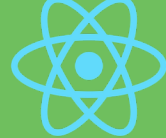
基準をOSSに合わせて開発

Do

Performance Auditing

- HTMLファイルサイズを減らす (71kB)
- Blocking Resourcesを減らす (8つ)
- JavaScriptの非同期読み込み
- CDNキャッシュの最適化
- Web Fontの統合 (3種類)
- いらない読み込みの削除
- 画像のCSS化

HTMLファイルサイズを減らす

- SSR + Lazy Load  
- 約1/7に削減 (71KB -> 9.8KB)

Blocking Resourcesを減らす

- バンドル 
- 8から3つに削減

JavaScriptの非同期読み込み

- SSR + バンドル + async属性

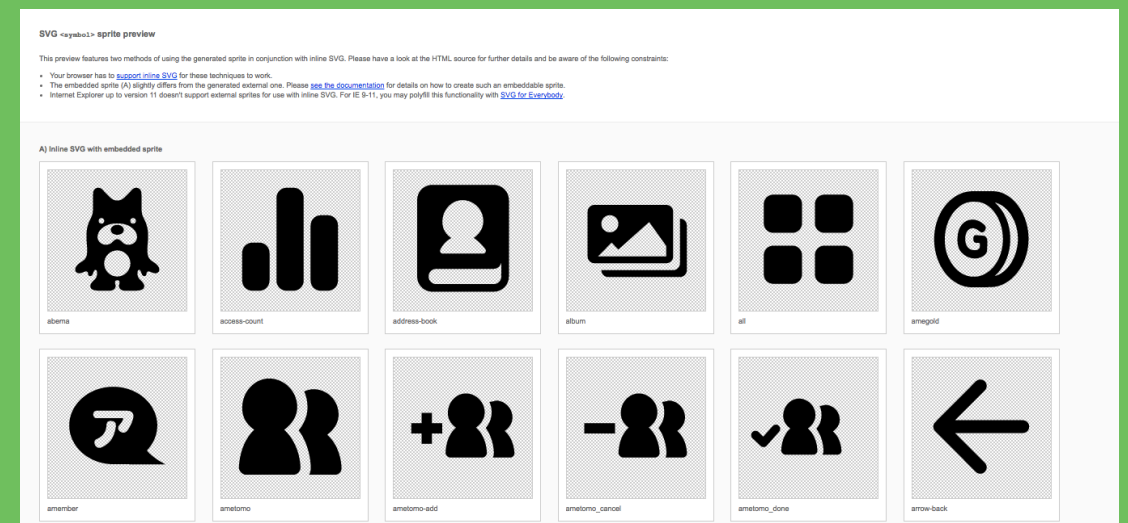
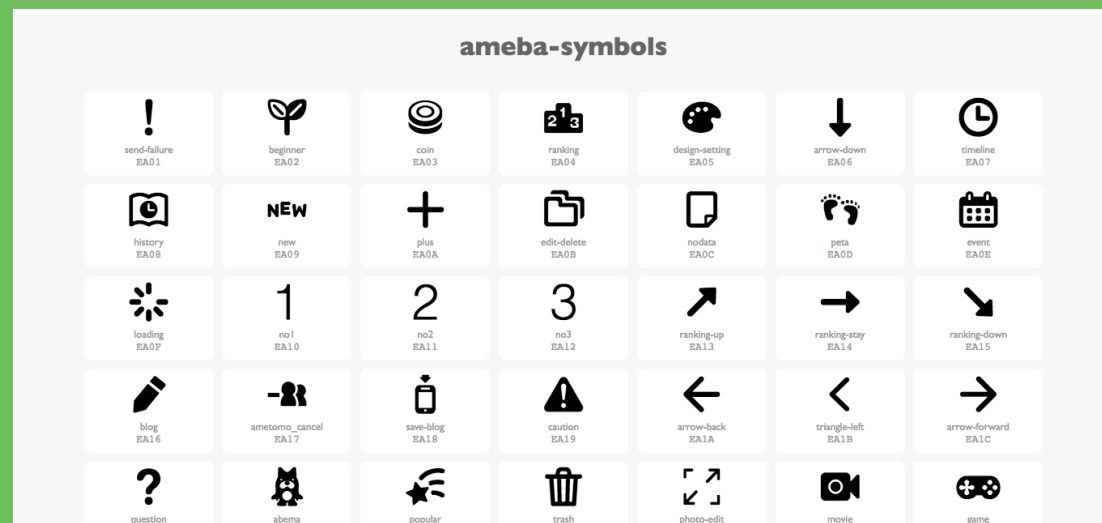
CDNキャッシュの最適化

- バージョニングしているので
長時間のキャッシュが可能

Web Fontの統合

- 各サービスで作成していた
- 共通利用できるように単体サービスとして運用

Web Fontの統合



Web Font (WOFF2, WOFF, TTF)

SVG Sprite

いらない読み込みの削除

- 既存のコードを読みながらリストアップ

画像のCSS化

- 画像でなくていい表現のCSS化

アプリをダウンロード

No more ガタンッ



本番のようにリリース

- リリースフローも固めていく

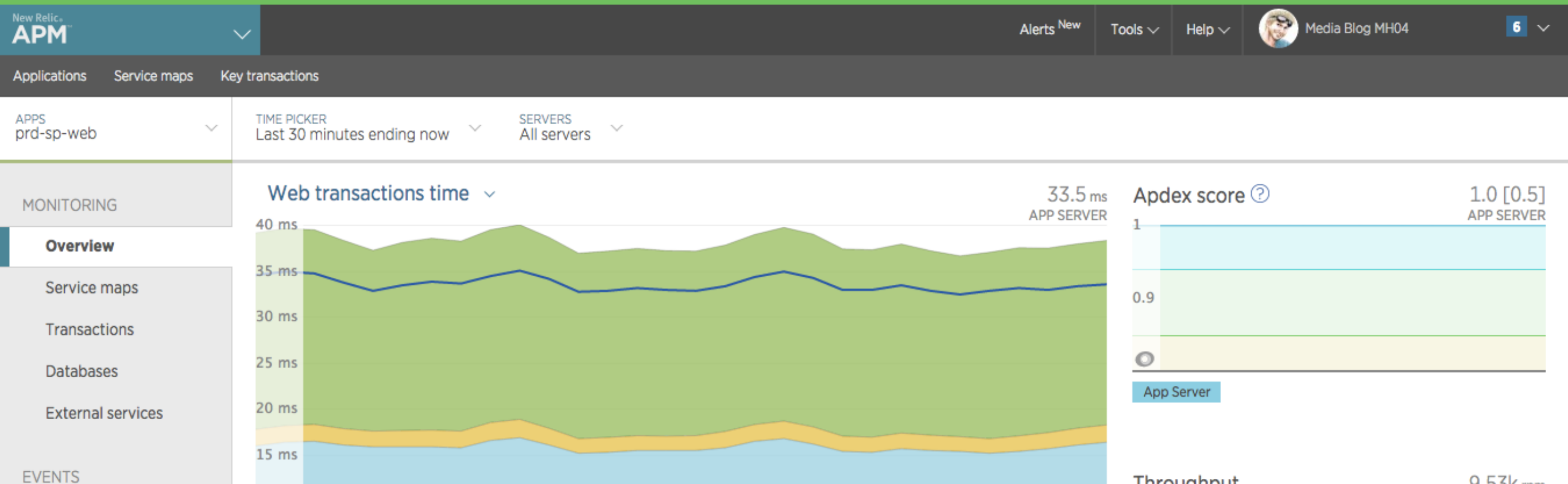
| | |
|---|--|
| v0.1.0
Closed on Apr 7, 2016 ⌚ Last updated 11 months ago
担当者: コウ | 
100% complete 0 open 16 closed
Edit Reopen Delete |
| v0.2.0
Closed on Apr 18, 2016 ⌚ Last updated 10 months ago
担当者: コウ | 
100% complete 0 open 26 closed
Edit Reopen Delete |
| v0.3.0
Closed on Apr 22, 2016 ⌚ Last updated 10 months ago
担当者: コウ | 
100% complete 0 open 17 closed
Edit Reopen Delete |
| v0.4.0
Closed on May 13, 2016 ⌚ Last updated 10 months ago
リリース担当者: @hara_kazunari | 
100% complete 0 open 21 closed
Edit Reopen Delete |
| v0.5.0 |  |

Testing

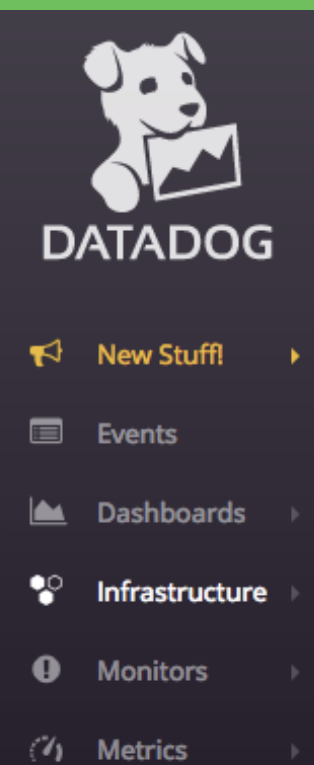
負荷試験

- 過去のアクセスログを元に検証
- Over 10,000 rps

負荷試験



負荷試験

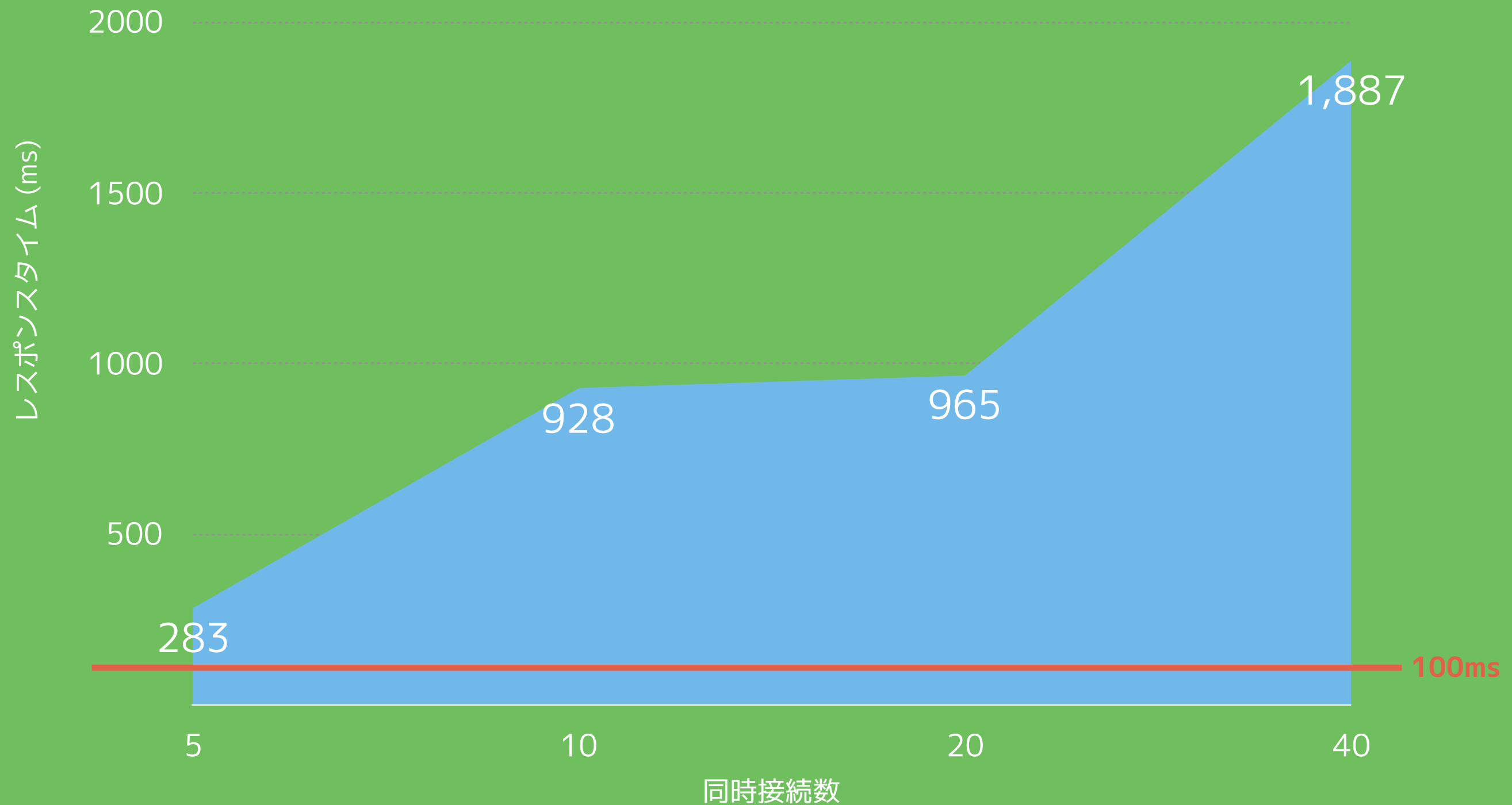


Filter by Group hosts by tags Fill by: % CPU utili... avg Size by: --

Your infrastructure



結果は悪かった



使い物にならない

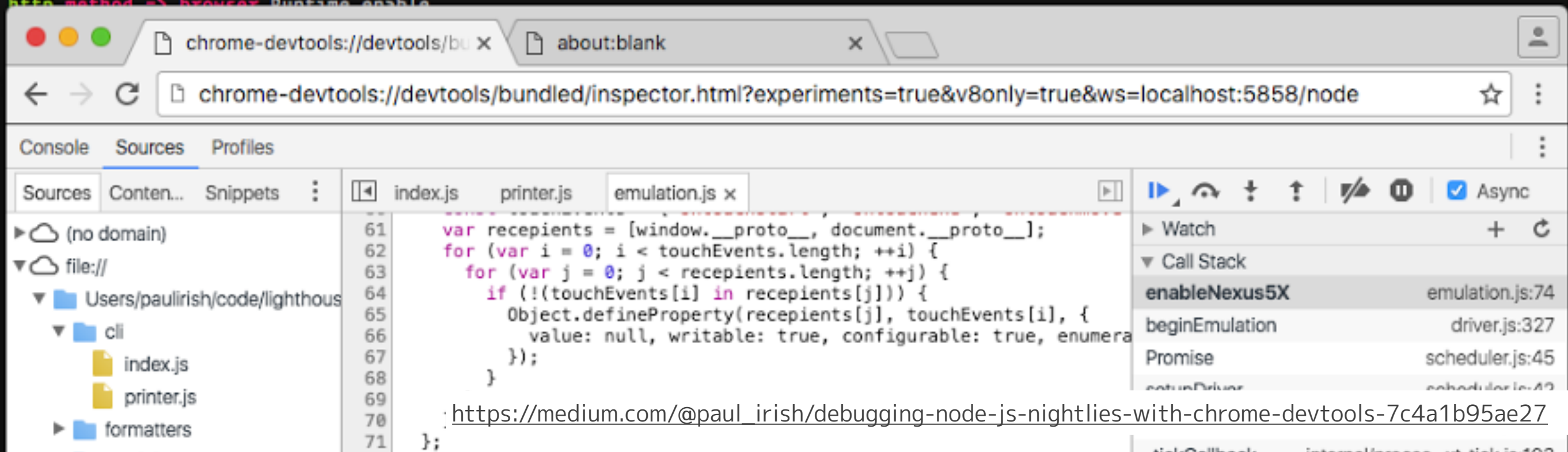


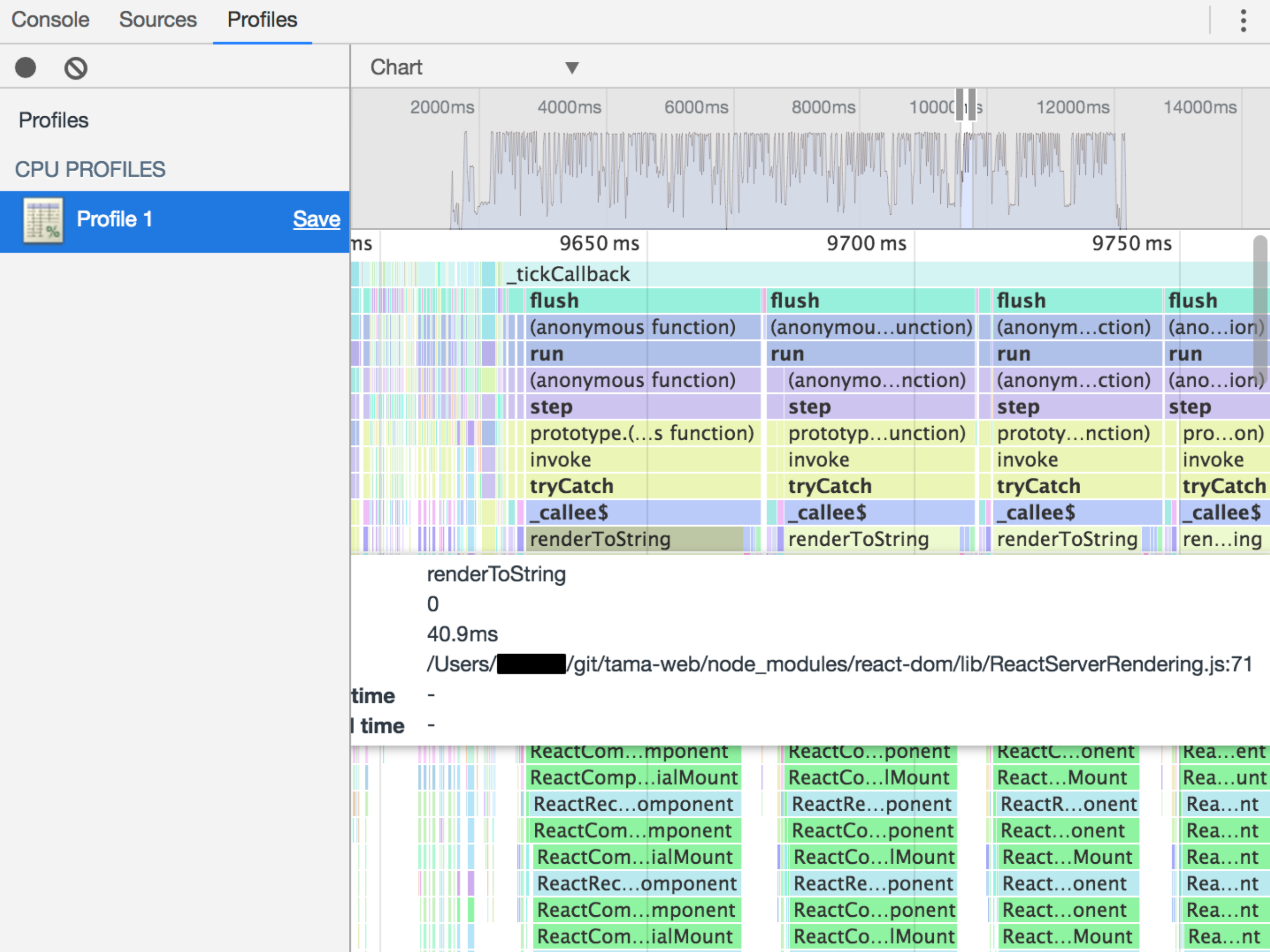
Turning

Node.js with Chrome DevTools

```
▶ node --inspect index.js
```

```
~/code/lighthouse on (master)
▶ ~/code/node-w-v8-inspector/out/Release/node --inspect --debug-brk cli/index.js https://nodejs.org/en/
Debugger listening on port 5858. To start debugging, open following URL in Chrome:
  chrome-devtools://devtools/remote/serve_file/@521e5b7e2b7cc66b4006a8a54cb9c4e57494a5ef/inspector.html?experiments=true&v8only=true&ws=localhost:5858/node
Debugger attached.
process.on(SIGPROF) is reserved while debugging
http_method => browser Runtime enable
```





renderToString() 遅い問題

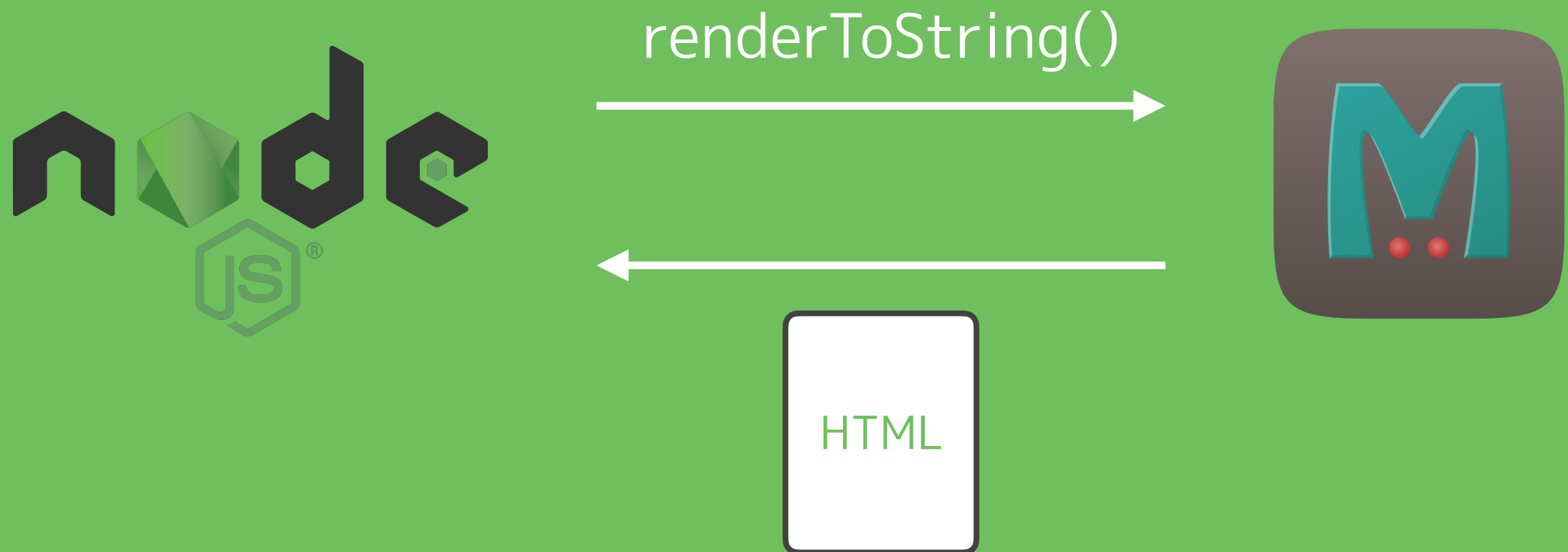
Caching

- 参照 > 更新
- 基本的に静的ドキュメント

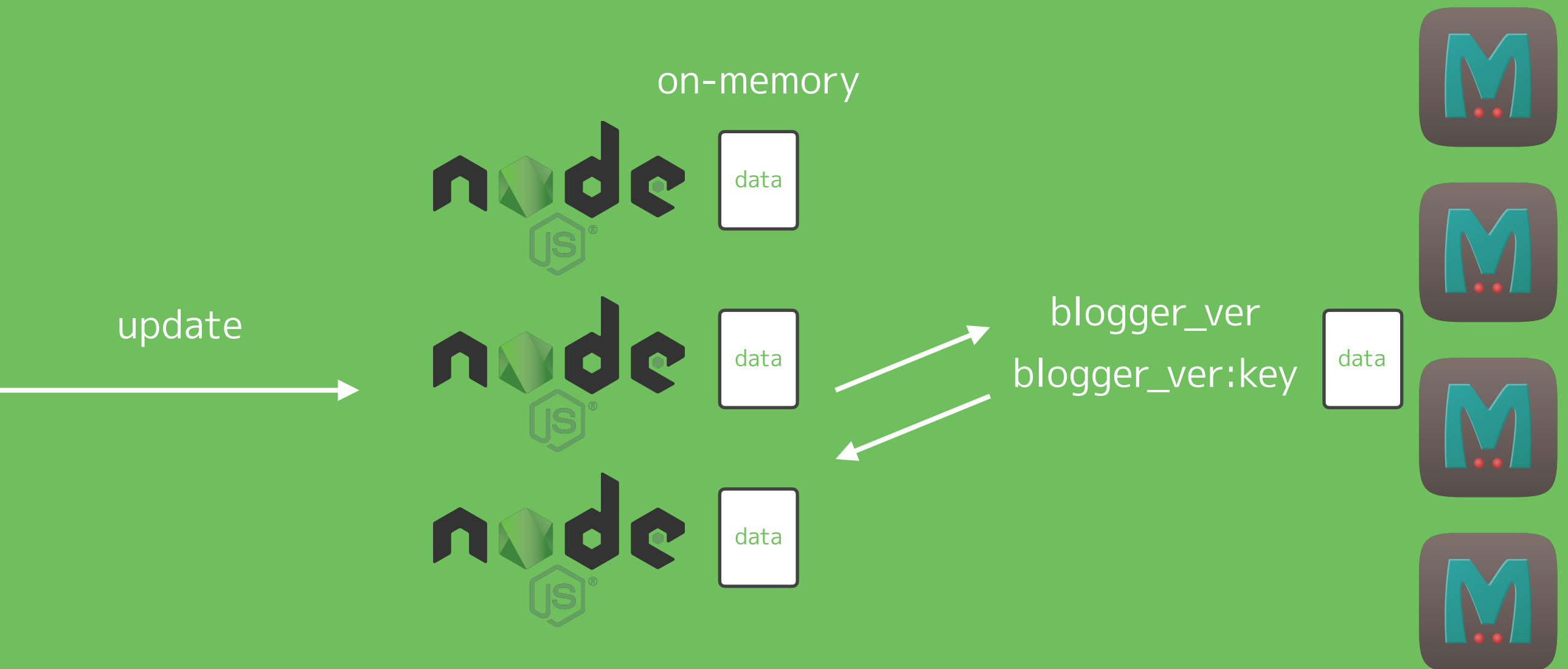
Cache Hit Rate



Caching



Caching (二層式)



Redis vs Memcached

- RedisのEviction発生によるパフォーマンス低下
- データ量大・高アクセス時の単純なキャッシュではMemcached



Under 100ms

/:amebald#GET

Track as key transaction



App performance

Historical performance

Map ^{Beta}

App server histogram



60 %

40 %

20 %



25 ms

50 ms

75 ms

100 ms

125 ms

150 ms

175 ms

200 ms

225 ms

React Fiber Architecture

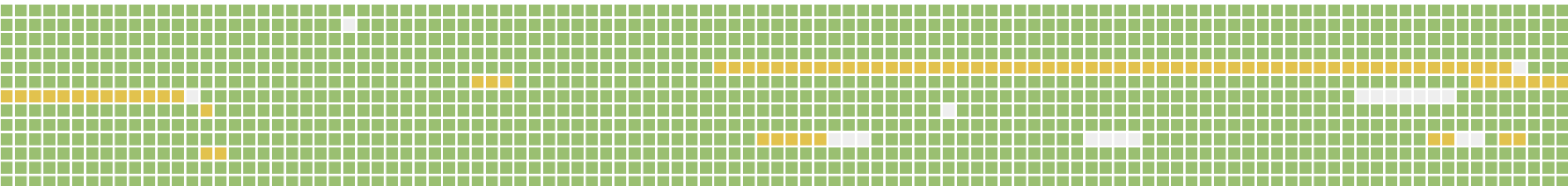
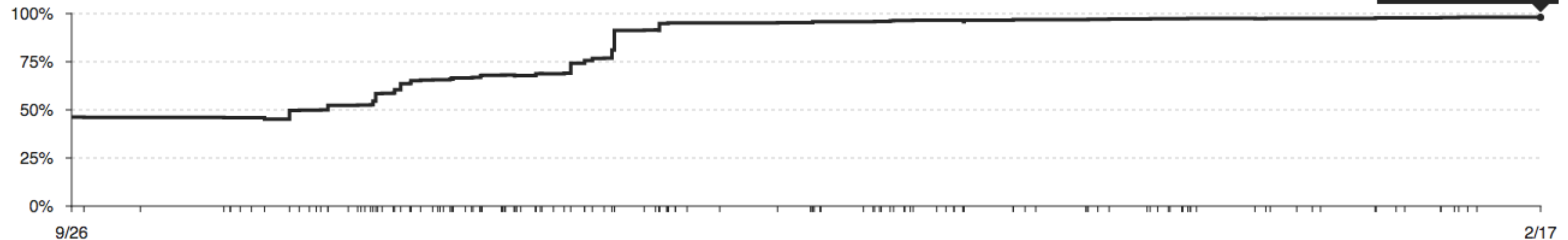
1640 of 1671 unit tests passing

NO

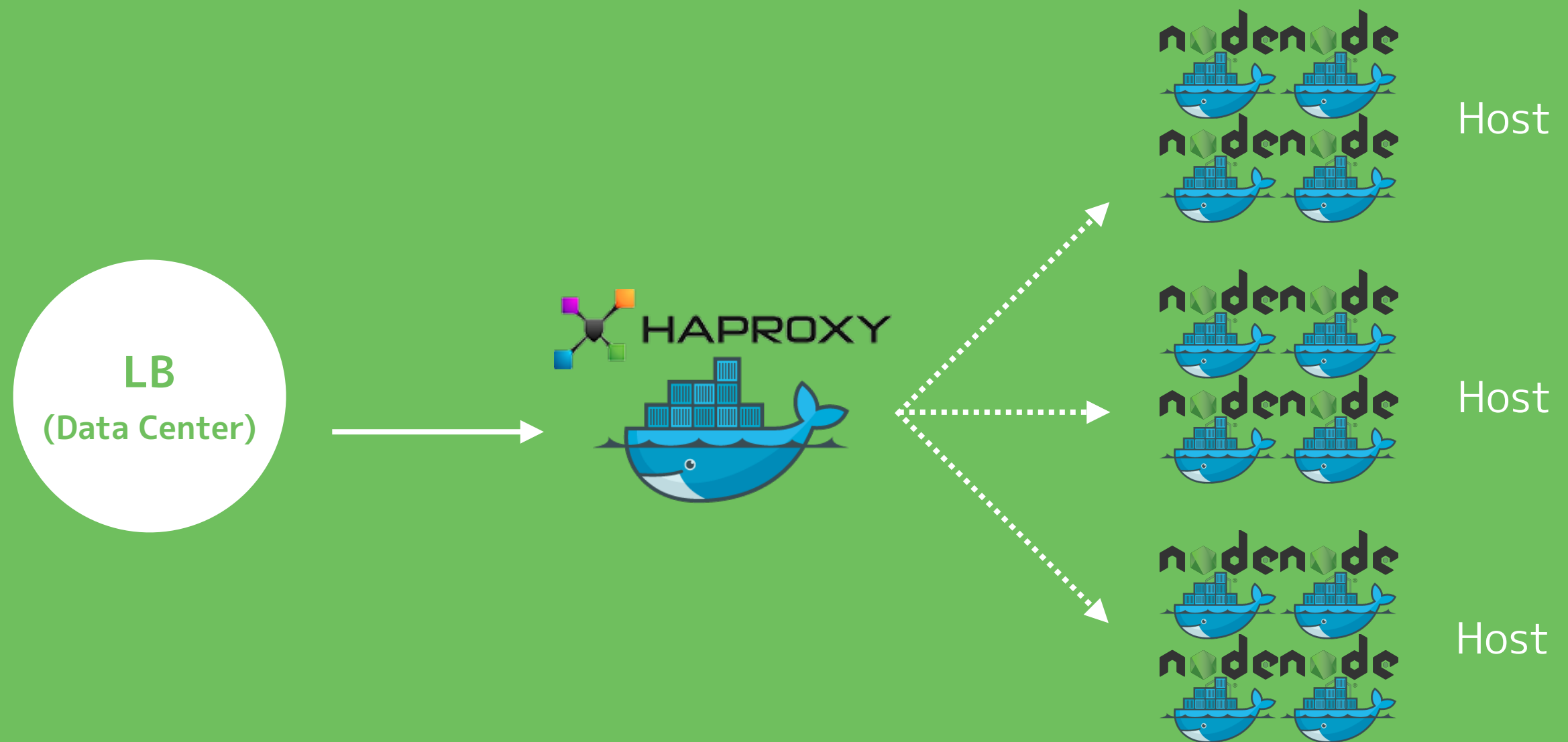
but this page was rendered with it ☒
98.1% of unit tests passing, 138 warnings left to fix

2017-02-17 02:06:48
→ bc2702f

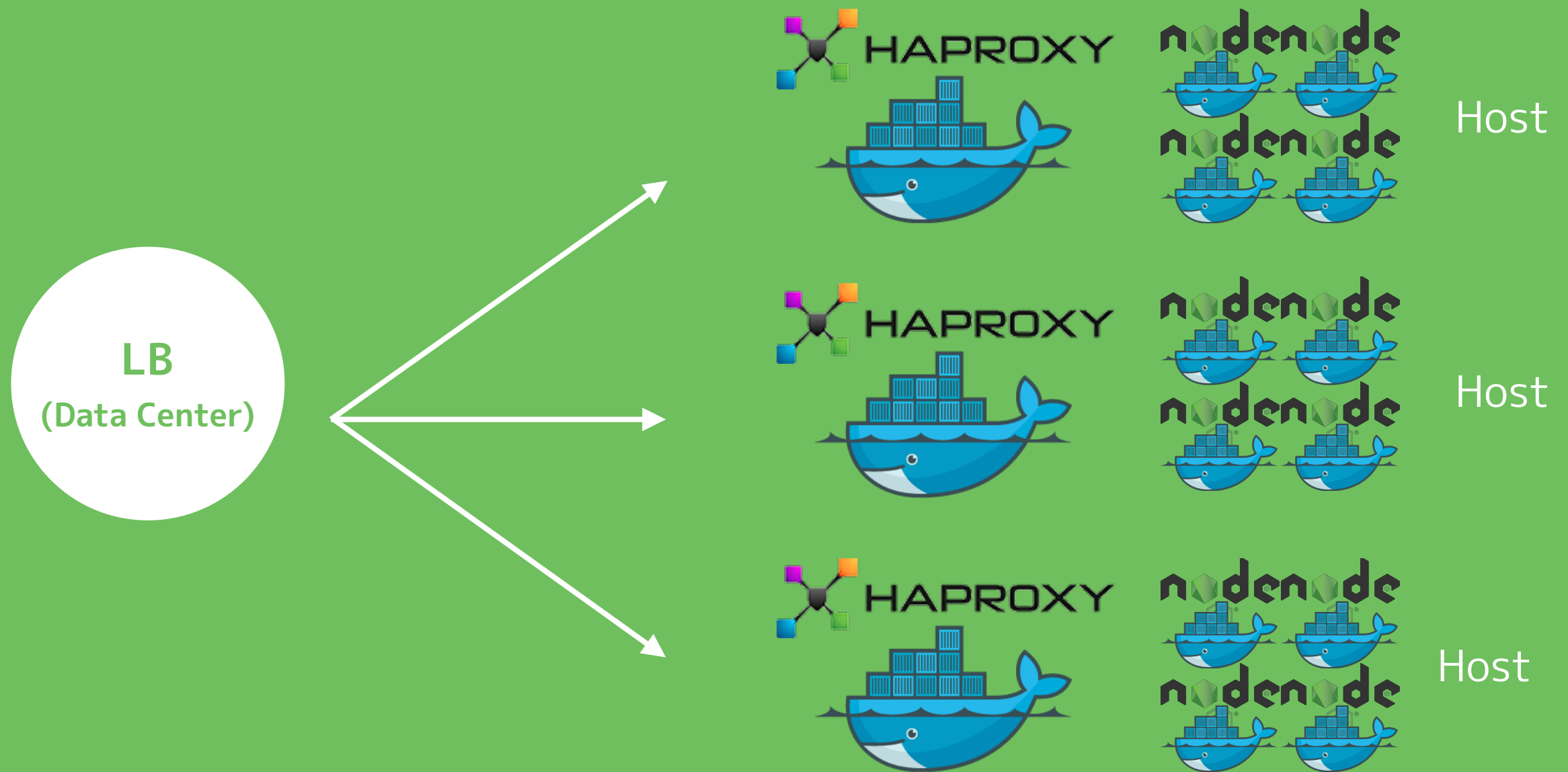
98.1% (1640 / 1671)



Infra Structure



Infra Structure



リリースは難産だった…

- 段階リリース、うまくいかない
- Slackに知らない人が入ってくる
- 段階リリース、うまくいかない
- Slackにエライ人が入ってくる

 **hawthorne_luke** 8:15 PM
60
 **AXIS** @AXIS125c
アメプロなんか変わったな
Twitter | Aug 31st, 2016 at 8:25 PM

 **hattori_takuya** 8:51 PM
おお、80%
internal-apiの負荷は普段とあまり変わらないですね

 **hawthorne_luke** 8:53 PM


 **hara_kazunari** 8:53 PM
[@hawthorne_luke](#) では次のマイルストーン 100% にしますか？
 8
internalもみつ

 **hawthorne_luke** 8:55 PM
いっちゃいますね！
85
 5

 **hawthorne_luke** 9:07 PM
9可
 5  1

 **hara_kazunari** 9:19 PM
[@hawthorne_luke](#) あともうひとoshi、がんばりますか...!

 **hawthorne_luke** 9:19 PM
95

 **hara_kazunari** 9:19 PM
はやw

 **hawthorne_luke** 9:19 PM ☆
後一個！
 3  2 

 **kou_hin** 9:20 PM
全台5500rpsぐらいです

 **hawthorne_luke** 9:22 PM
internal-apiに優しいのかな、すごいLA低い気がしますけどw

 **hara_kazunari** 9:23 PM
おおw
同じプロガー見る分にはかなり優しいはず

 **hawthorne_luke** 9:30 PM
1  
 7  4  2  2  2  2



2016.08.31 🚀

“Premature optimization is the root of all evil.”

— Donald Knuth



教訓

- 刷新は大変だった
- そのぶん知見も得られた
- 老害にならないようにがんばらなくちゃいけない 

One more thing...



Front-End Performance Checklist 2017

CODING

CSS

HTML

JavaScript

Techniques

DESIGN

Web Design

Responsive

Typography

Inspiration

MOBILE

iPhone & iPad

Android

Design Patterns

Front-End Performance Checklist 2017 (PDF, Apple Pages)

By [Vitaly Friedman](#) December 21st, 2016 [Checklists, Performance](#) [15 Comments](#)

Meet the new [Sketch Handbook](#), our brand **new Smashing book** that will help you master all the tricky, advanced facets of Sketch. Filled with **practical examples and tutorials in 12 chapters**, the book will help you become more proficient in your work. [Get the book now →](#)

Advertisement

[Become a sponsor!](#)

EARN YOUR MS IN
**INFORMATION
DESIGN AND
STRATEGY**
Northwestern

[Study Completely Online](#)

Get
**RESPONSIVE
HTML5/CSS3**
CODED BY PROS!
[psd2html.com](#)

[PSD2HTML.com](#)

Smashing Newsletter

Subscribe to our email newsletter for useful tips and valuable resources, sent out every second Tuesday.

email address

Subscribe

234,898

Subscriberspowered by [MailChimp](#)

Service Worker, HTTP/2, CDN,
srcset, client hints,
resource hints, HPACK,
AMP, Instant Atricle etc.

Lighthouse

Lighthouse

Version: 1.2.3

BETA

Progressive Web App

Best Practices

Performance Metrics

Fancier stuff

Results for: <https://pwa.rocks/>

Generated on: 12/14/2016, 6:46:04 PM PST



96
100

Progressive Web App

These audits validate the aspects of a Progressive Web App.

App can load on offline/flaky connections

Ensuring your web app can respond when the network connection is unavailable or flaky is critical to providing your users a good experience. This is achieved through use of a [Service Worker](#).

- ✓ Has a registered Service Worker ?
- ✓ URL responds with a 200 when offline ?

Page load performance is fast

Users notice if sites and apps don't perform well. These top-level metrics capture the most important perceived performance concerns.

98 First meaningful paint: **1329.2ms** (target: 1,600ms) ?

100 Perceptual Speed Index: **478** (target: 1,250) ?

First Visual Change: **337ms**

Last Visual Change: **1343ms**

<https://github.com/GoogleChrome/lighthouse>

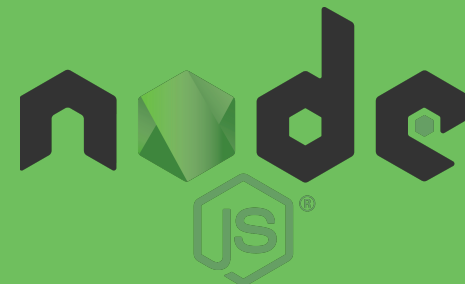
フロントエンドでできること



HTML



CSS



フロント、がんばってこ💪

Ask Me Anything



HOU Bin



herablog



アメブロ2016

実録、アメブロリニューアル275日

～ Webパフォーマンス編 ～

2017.2.25 Inside Frontend #1 @Yahoo

Kazunari Hara (@herablog)