10 YEARS OF NODE.JS

CHRONICLES OF NODE.JS
COMMUNITY & ENTERPRISE

@matvi3nko

Nikolay Matvienko

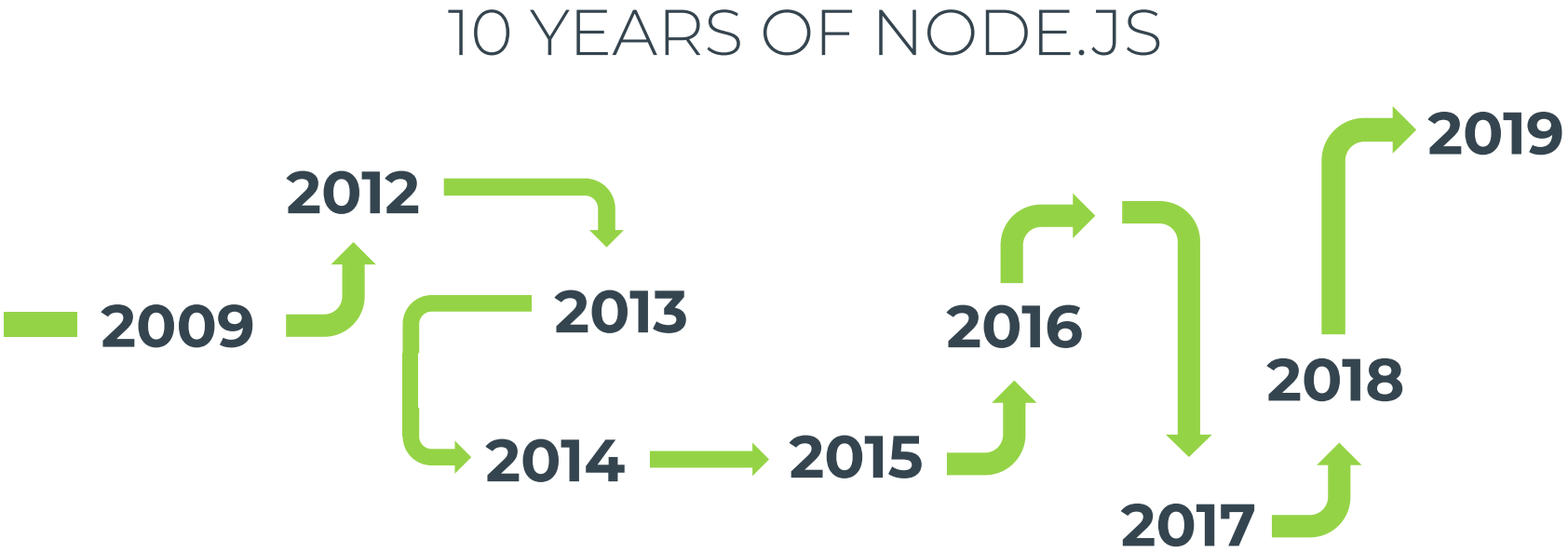
Senior Software Engineer and Node.js expert at Grid Dynamics

Node.js diagnostics, performance improvement consultant.

You can find me at twitter.com/**matvi3nko**

github.com/**matvi3nko**

PLAN


10 YEARS OF NODE.JS
2009
2012
2013
2014
2015
2016
2017
2018
2019

PREHISTORY

Dan Shaw - Node.JS and the web platform
https://youtu.be/thCFBOs54AE

2000 – 2009

HTML5

2012

THE FIRST NODE.JS SUMMIT

https://www.nodesummit.com/videos/

2012

THE FIRST NODE.JS SUMMIT

https://www.nodesummit.com/videos/

2013

node summit san francisco

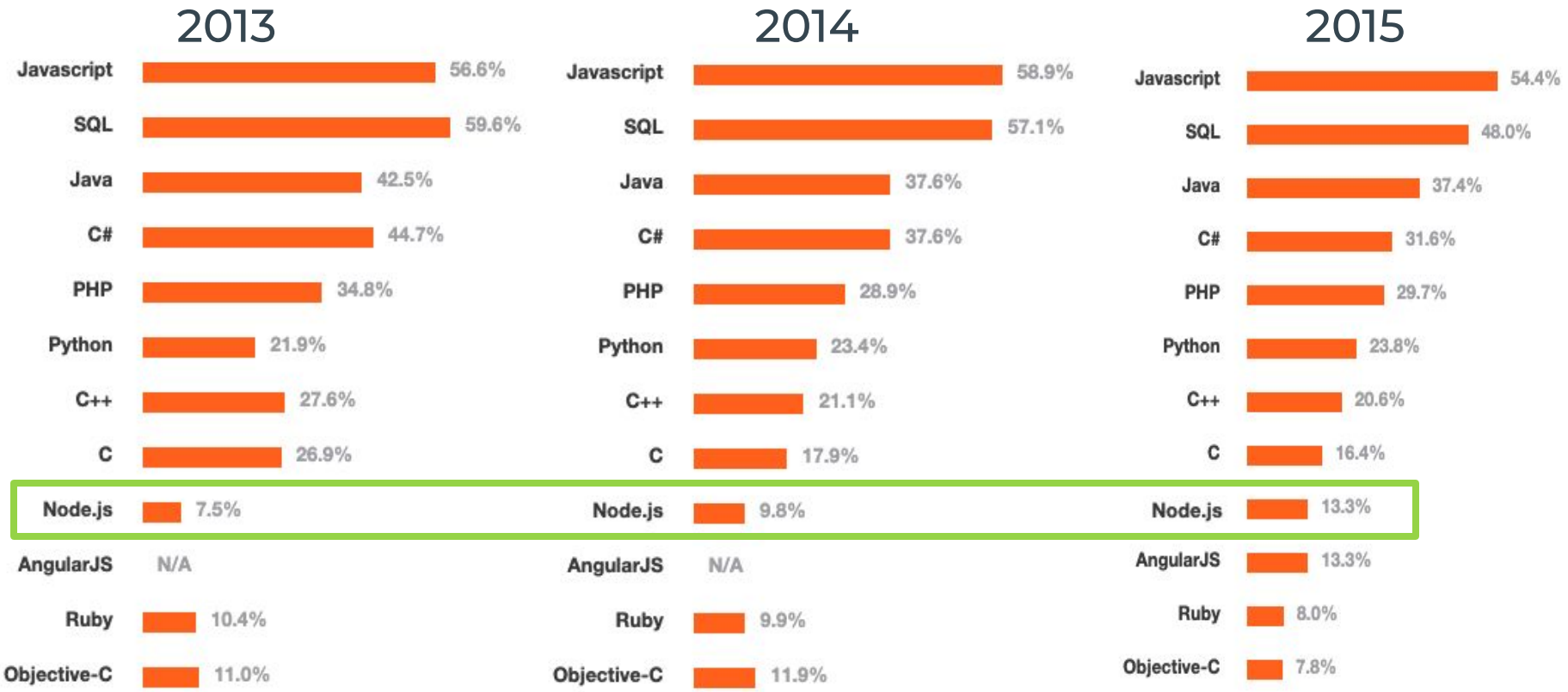
Node.js Helps NASA Keep Astronauts Safe and Data Accessible

https://foundation.nodejs.org/wp-content/uploads/sites/50/2017/09/Node_CaseStudy_Nasa_FNL.pdf

> Not surprisingly Estes' Node.js system rapidly won hearts and minds of NASA's engineers and developers.

Node.js Brings Big Data to the Cloud, so Astronauts Can Live and Work in Space

MOST POPULAR TECHNOLOGIES (stackoverflow)




SMART TV
SAMSUNG


BLACK FRIDAY
extra
SMART TV

Online Only Black Friday and Thanksgiving 2014 Sales (Figures are in Billions)

	2010	2011	2012	2013	2014
Black Friday	$0.648	$0.816	$1.042	$1.198	$1.505
Thanksgiving	$0.407	$0.479	$0.633	$0.766	$1.009

https://bestblackfriday.com/blog/black-friday-2014-sales-figures-numbers-recap/

Retail sites crash under weight of online Black Friday shoppers

Lowe's website goes down temporarily on Black Friday

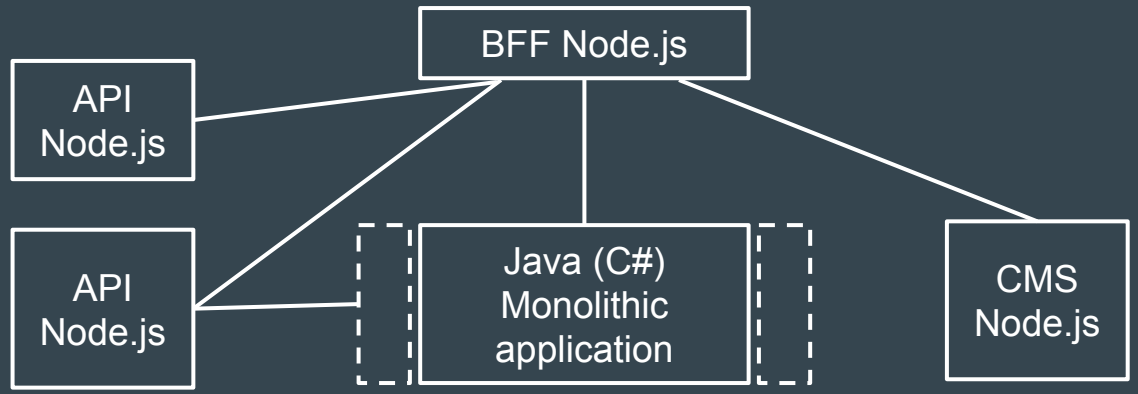
Black Friday – Cyber Monday Weekend: Mobile Traffic Soars But Overall Sales Down

J.Crew's web[illegible]ustomers try to shop Black Friday deals, and experts say it could have cost the company over $700,000 in sales

Facebook's ad platform has crashed — causing chaos just days before Black Friday

YEAR OF DIGITAL TRANSFORMATIONS WITH NODE.JS

2014



- ISOMORPHIC
- FAST RELEASE CYCLES
- HIGHT APPLICATION
- **PERFORMANCE** & **UPTIME**

JOYENT
SPLIT
FORK IO.JS
NO NODE.JS SUMMIT

https://blog.risingstack.com/what-is-nodejs-used-for-the-2015-nodejs-overview-report/

Hapi

"Originally developed to handle Walmart's **Black Friday scale"**

https://hapi.dev/

Express

FRAMEWORKS

Custom framework

IoC, MIddlewares, Adapter, Strategy, ...

(like NestJS now)

Node.js Design Patterns 2nd

https://www.oreilly.com/library/view/nodejs-design-patterns/9781785885587/

I DEVELOP ON NODE.JS

WENT-LIVE / PRODUCTION

2015

AWS Lambda

Popularity in the world:

Java – 36%

C# – 30%

PHP – 25%

Node.js – **17%**

Backend for frontend:
Node.js
caught up with
PHP

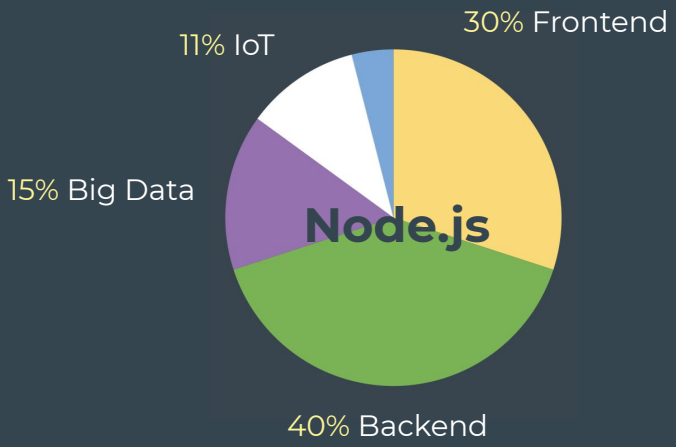
→2016

WIDELY USED AND MOSTLY IN ENTERPRISE AND **WEB DEVELOPMENT**

At Node Summit 2016:

- Netflix
- Google
- IBM
- Nasa
- Disney
- Walmart Labs
- eBay
- ...

shared success stories of switching to Node.js



and 40% of all PROJECTS USE MESSAGE QUEUES

https://nodejs.org/static/documents/2016-survey-report.pdf https://blog.risingstack.com/node-js-developer-survey-results-2016/

RESULTS OF USING NODE.JS

100% Business Availability, even with Extreme Load

Around 1.5 billion dollars are being spent online in the US on a single day on Black Friday, each year.

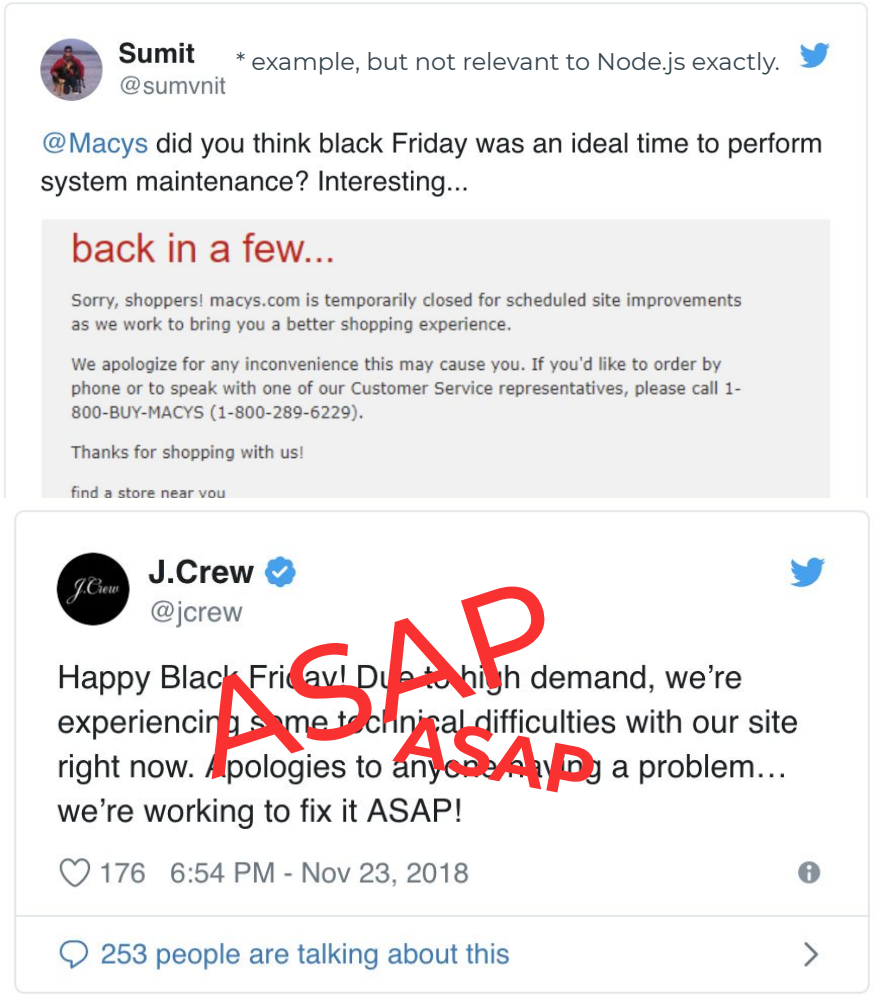
It is crucial that your site can keep up with the traffic. This is why **Walmart**, one of the biggest retailers is **using Node.js to serve 500 million pageviews on Black Friday, without a hitch.**

When **PayPal** started using Node.js, they reported an **2x increase in productivity** compared to the previous Java stack. How is that even possible?

https://blog.risingstack.com/digital-transformation-with-the-node-js-stack/

WITH GROWTH OF PROJECTS:

1. BEST PRACTICES OF BIG PROJECTS
2. ASYNCHRONOUS / EVENT LOOP
3. DEVELOPERS AND TECH LEADERS ARE MORE INTERESTED IN TOOLS and ENTERPRISE TOOLS



WITH GROWTH OF PROJECTS:

1. BEST PRACTICES OF BIG PROJECTS
2. ASYNCHRONOUS / EVENT LOOP
3. DEVELOPERS AND TECH LEADERS ARE MORE INTERESTED IN TOOLS and ENTERPRISE TOOLS

LACK OF TOOLS EFFICIENT IN PRODUCTION

PROFILING
D scripts, System profilers

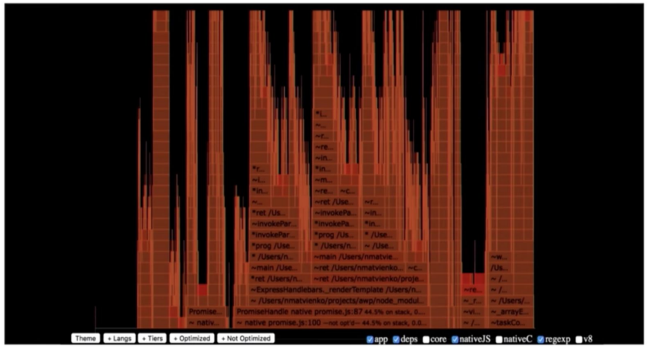
THE MOST ACUTE PROBLEM IN 2016 –
DEBUGGING
81.97% used console.log
https://blog.risingstack.com/node-js-developer-survey-results-2016/

2017

TONS OF ANONYMOUS FUNCTIONS

HEAVY FLAME GRAPHS

Пример 5. 0x flame graph приложения.



https://youtu.be/_qzFJ2MPVWQ

“Grokking Asynchronous Work in Node.js - Thorsten Lorenz” https://youtu.be/8Xoht4J6Jjw

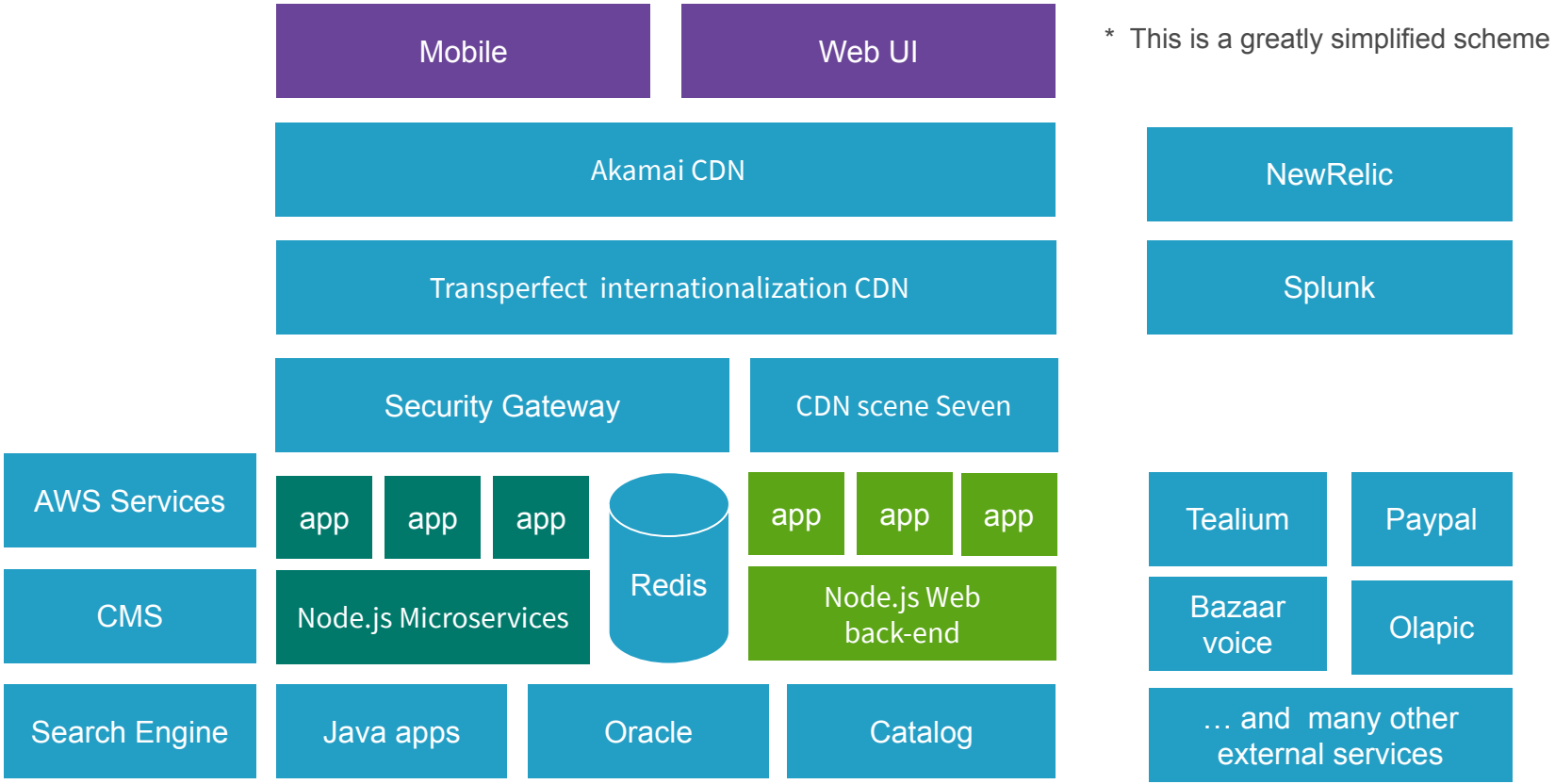
V8 API CHANGES:

1. INSTRUMENTS or NODE UPGRADE
2. WE HAD TO KNOW JIT
3. APPLY OPTIMIZATIONS IN CODE
4. PROFILE LIBRARIES

0x FLAME GRAPH

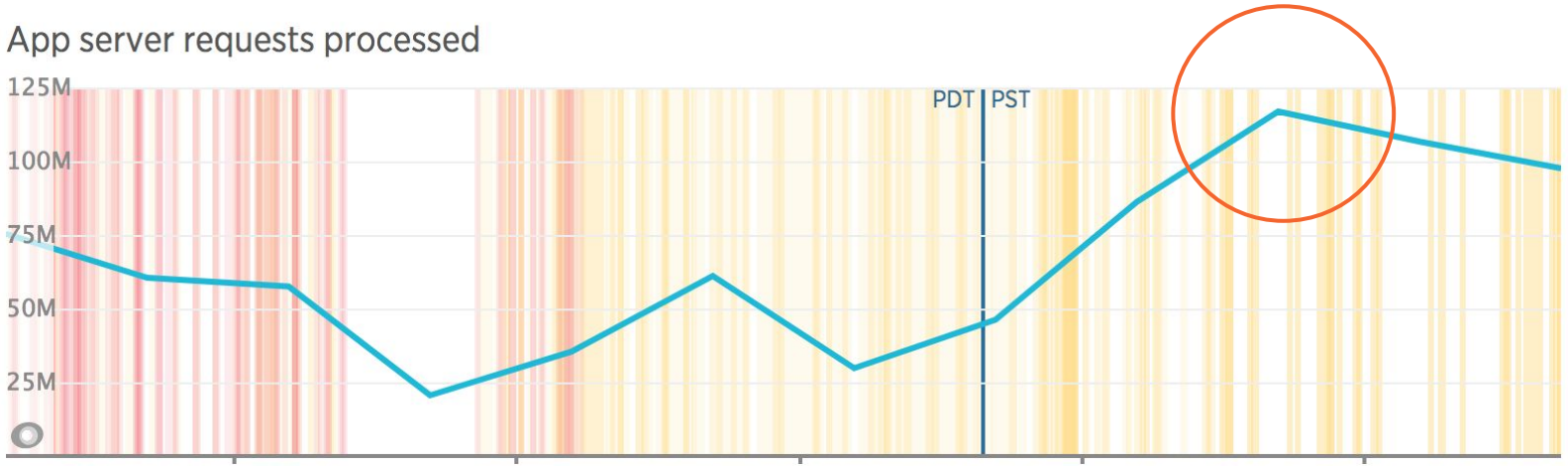
LLNODE

Node.js in Enterprise Architecture

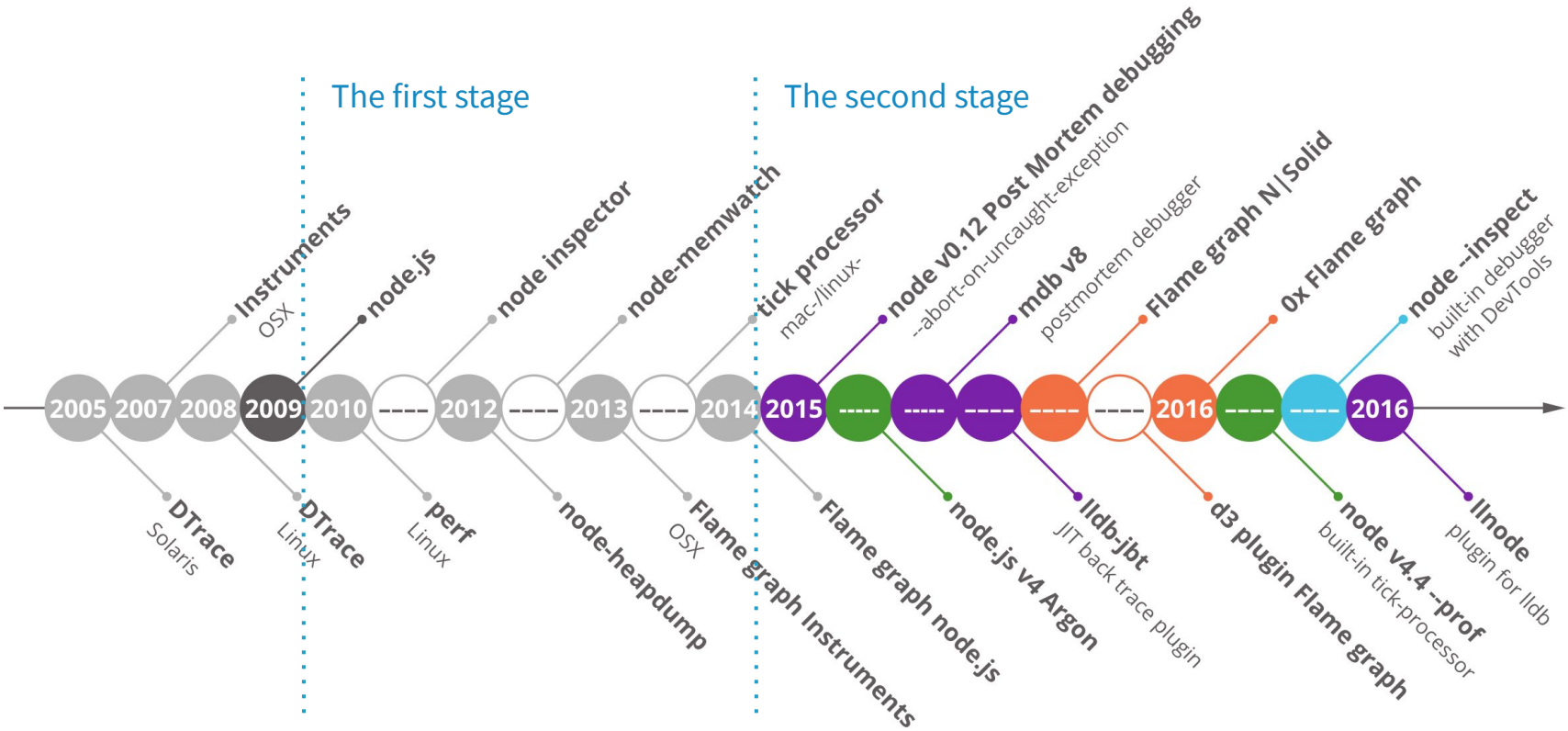


War room

Black Friday = ~60% of the annual income.



Stages of Node.js diagnostics evolution before 2017



Case 1. Product reservation.

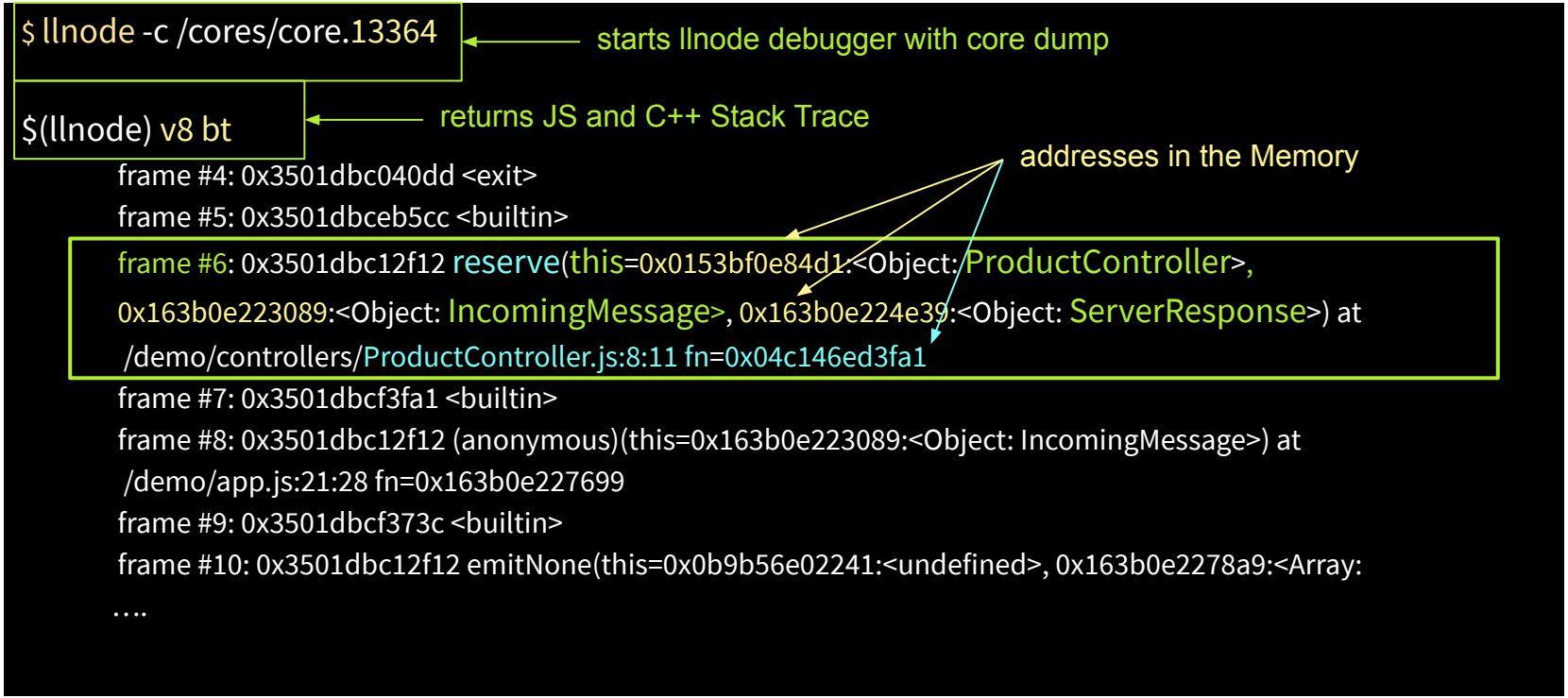
```
module.exports = class ProductController {
  constructor (reservationService) {
    this._reservationService = reservationService;
  }

  reserve (req, res) {
      const { id, storeId } = req.cookies['profile'];
      const rewards = req.cookies['rewards'];
      const { products } = req.body;

      this._reservationService.reserve(id, storeId, rewards.id, products)
        .then(data => {
            return res.send(data);
        });
  }};
```

Throws an Error when rewards is undefined

Step 1. Get the Stack Trace and find the last JS function



Step 3 result. Request object

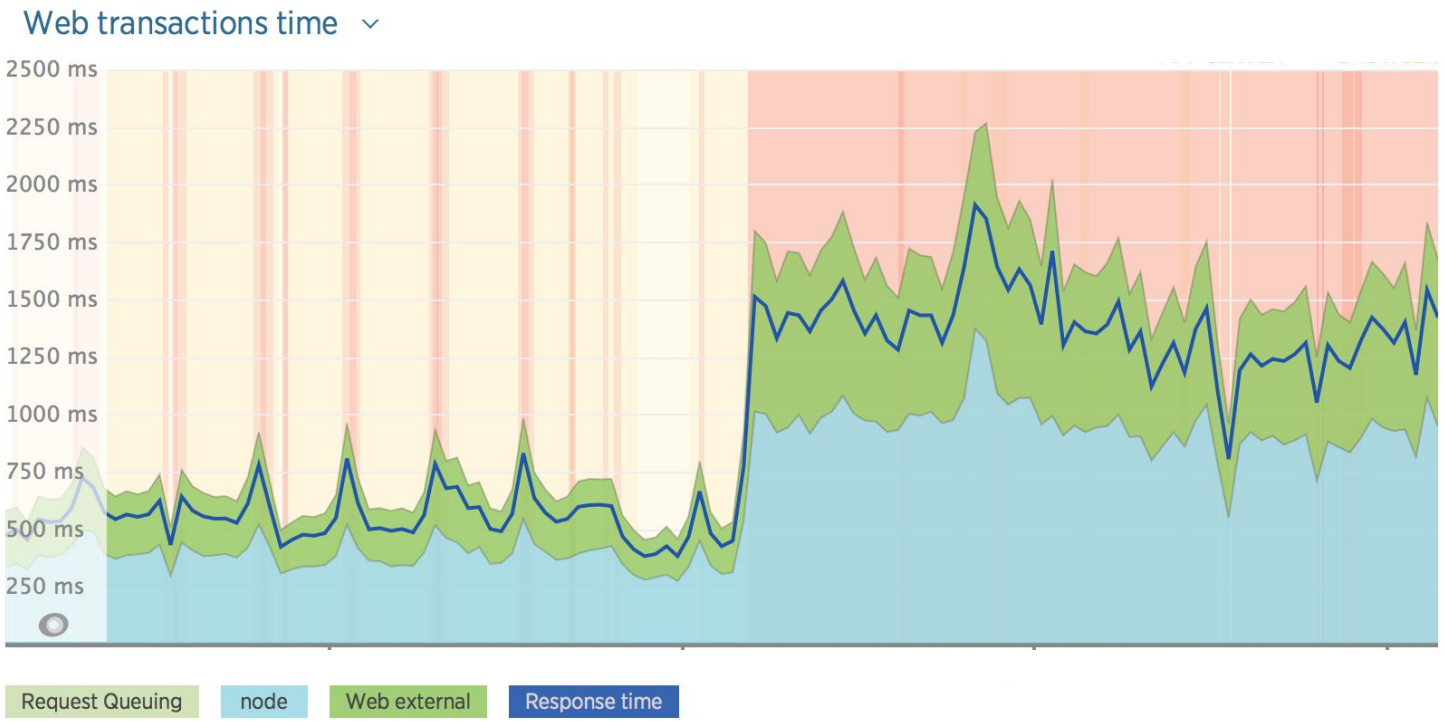
```
$(llnode) v8 inspect 0x163b0e223089

0x163b0e223089:<Object: IncomingMessage properties {
    ._events=0x163b0e223739:<Object: Object>,
    .socket=0x163b0e21e4e1:<Object: Socket>,
    .complete=0x0b9b56e022c1:<true>,
    .cookies=0x5c4b0e863755:<Object: Object>,
    .headers=0x163b0e223d11:<Object: Object>,
    .(external)=0x163b0e223069:<String: "/product/reserve">,
    .method=0x04c146ed42a1:<String: "POST">,
    .statusCode=0x0b9b56e02211:<null>,
    .statusMessage=0x0b9b56e02211:<null>,
    .body=0x163b0e22aee1:<Object: Object>}>
…
```

```
$(llnode) v8 inspect 0x5c4b0e863755

0x5c4b0e863755:<Object: Object properties {
    .profile=0x3d76d2184b41:<Object: Object>,
    .rewards=0x163b0e22af51:<undefined>}>
```

2. The response of the Node.js web server has grown from ~300 to ~1500 ms



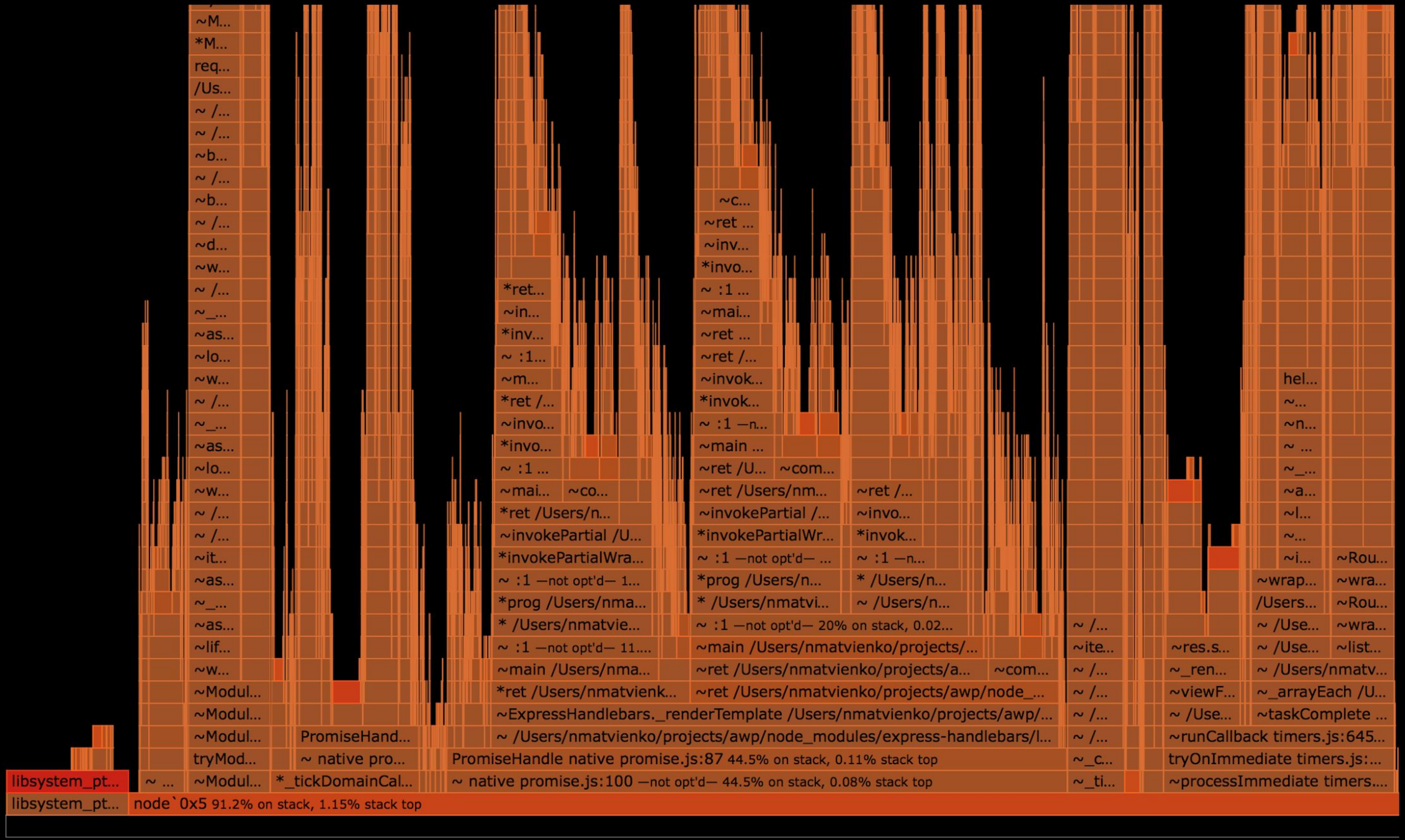
Profiling: node --prof(iler) + apache bench/JMeter

node --prof app.js and node --prof-process isolate-0x-v8.log > profile.txt

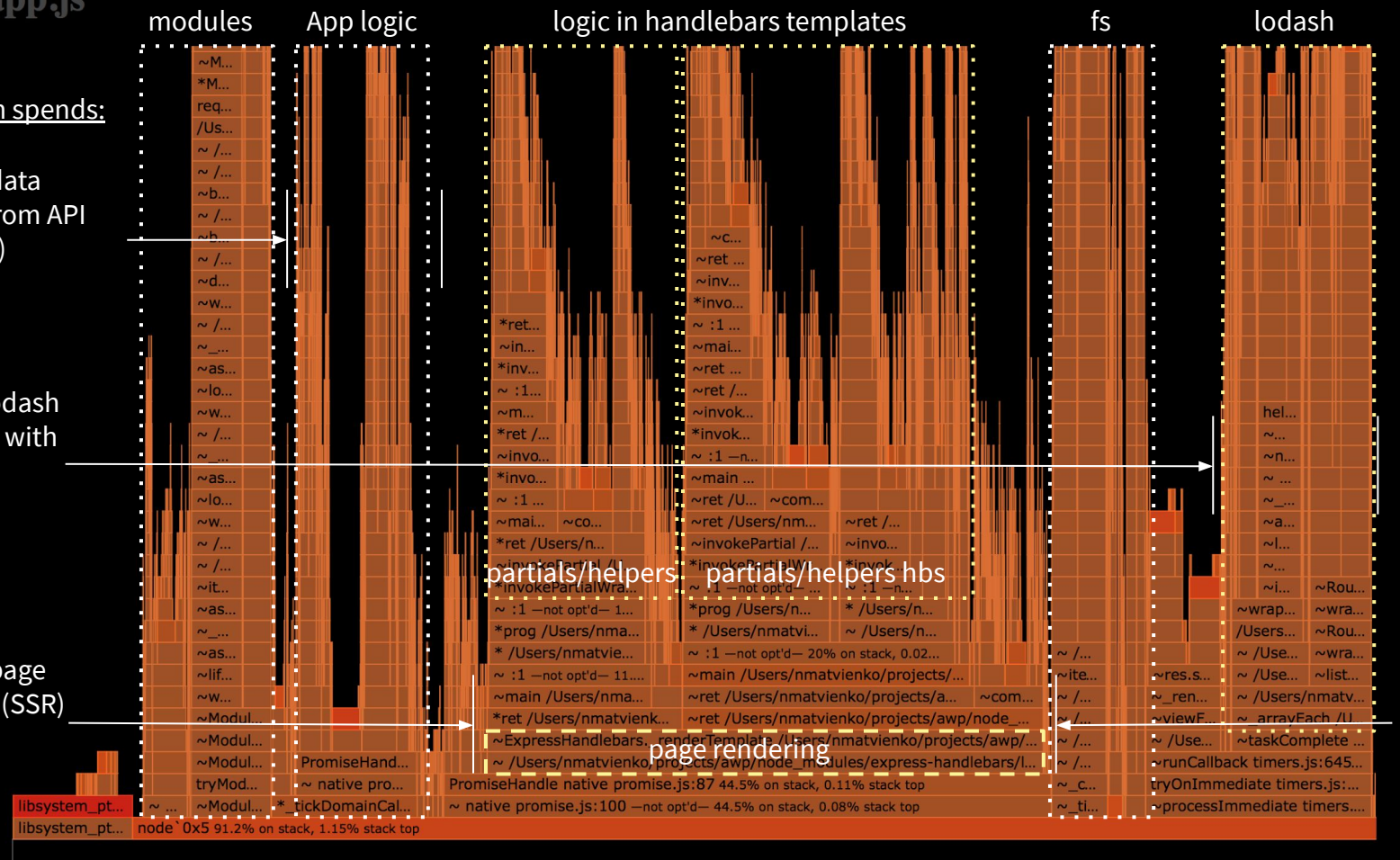
```
[Summary]:
   ticks  total  nonlib   name
    2710   18.9%   19.0%  JavaScript
   13348  58.7%   58.9%  C++
    561     2.5%    2.5%   GC
[JavaScript]:
     112    0.5%    0.5%  Builtin: CallFunction_ReceiverIsNotNullOrUndefined
       4    0.0%    0.0%  LazyCompile: *emit events.js:136:44

      42  15.2%   15.2% presentation/handlebars/helpers/urlBuilder.helper.js:51:37
       2    0.0%    0.0%  LazyCompile: ~substr native string.js:324:22
...
  [C++]:
    1292    5.7%    5.7%  node::ContextifyScript::New(v8::FunctionCallbackInfo<v8::Value> const&)
...
```

0x app.js


libsystem_pt...
node`0x5 91.2% on stack, 1.15% stack top
~ native promise.js:100 —not opt'd— 44.5% on stack, 0.08% stack top
PromiseHandle native promise.js:87 44.5% on stack, 0.11% stack top
~ /Users/nmatvienko/projects/awp/node_modules/express-handlebars/l...
~ExpressHandlebars._renderTemplate /Users/nmatvienko/projects/awp/...
*ret /Users/nmatvienk...
~ret /Users/nmatvienko/projects/awp/node_...
~main /Users/nma...
~ret /Users/nmatvienko/projects/a...
~main /Users/nmatvienko/projects/...
~ :1 —not opt'd— 20% on stack, 0.02...
*_tickDomainCal...
~ native pro...
PromiseHand...
tryMod...
~processImmediate timers....
tryOnImmediate timers.js:...
~runCallback timers.js:645...
~taskComplete ...
~_arrayEach /U...
~ /Users/nmatv...

0x app.js

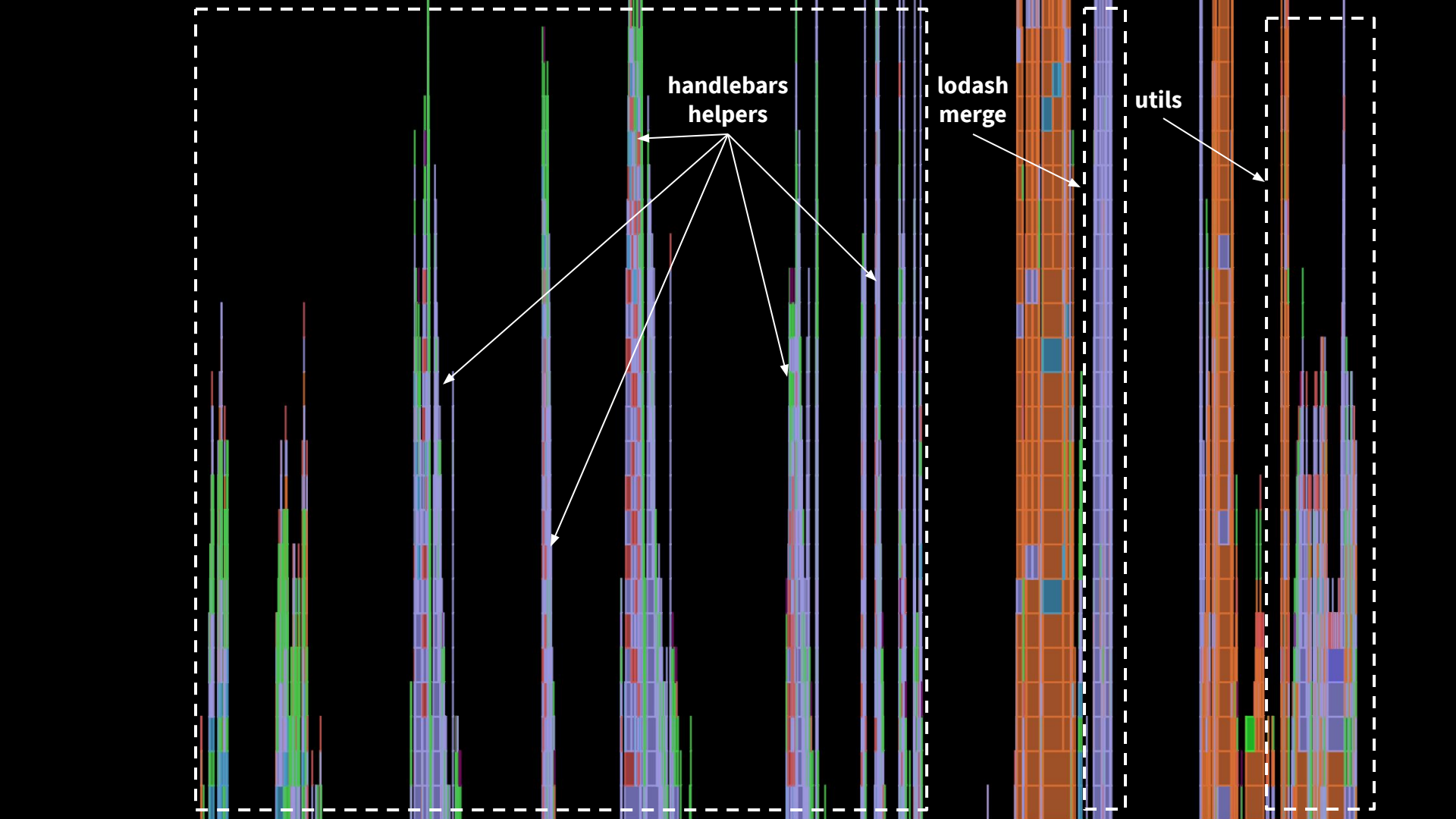


Application spends:

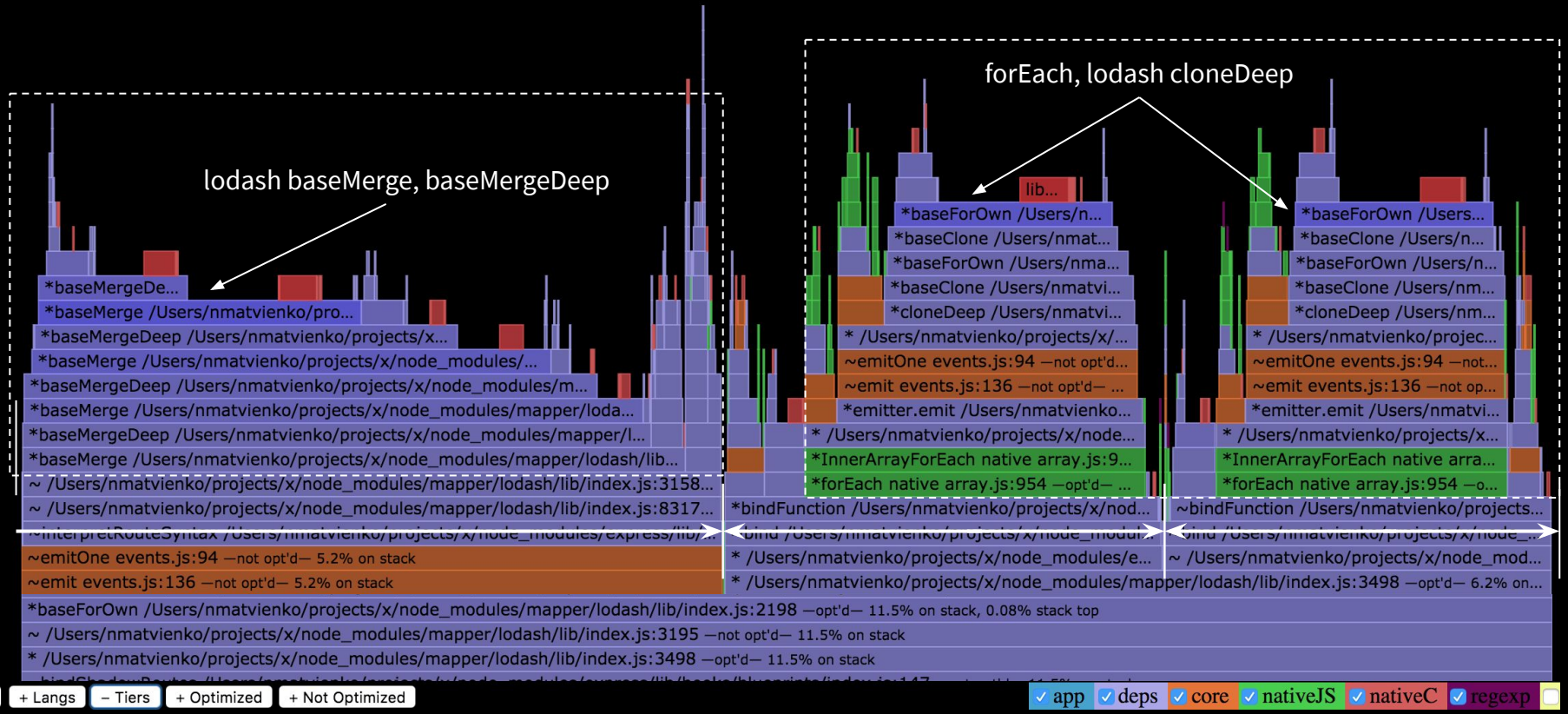
~10% on data retrieval from API (Promises)

~20% on lodash operations with data

~65% on page rendering (SSR)


handlebars
helpers
lodash
merge
utils

Data processing

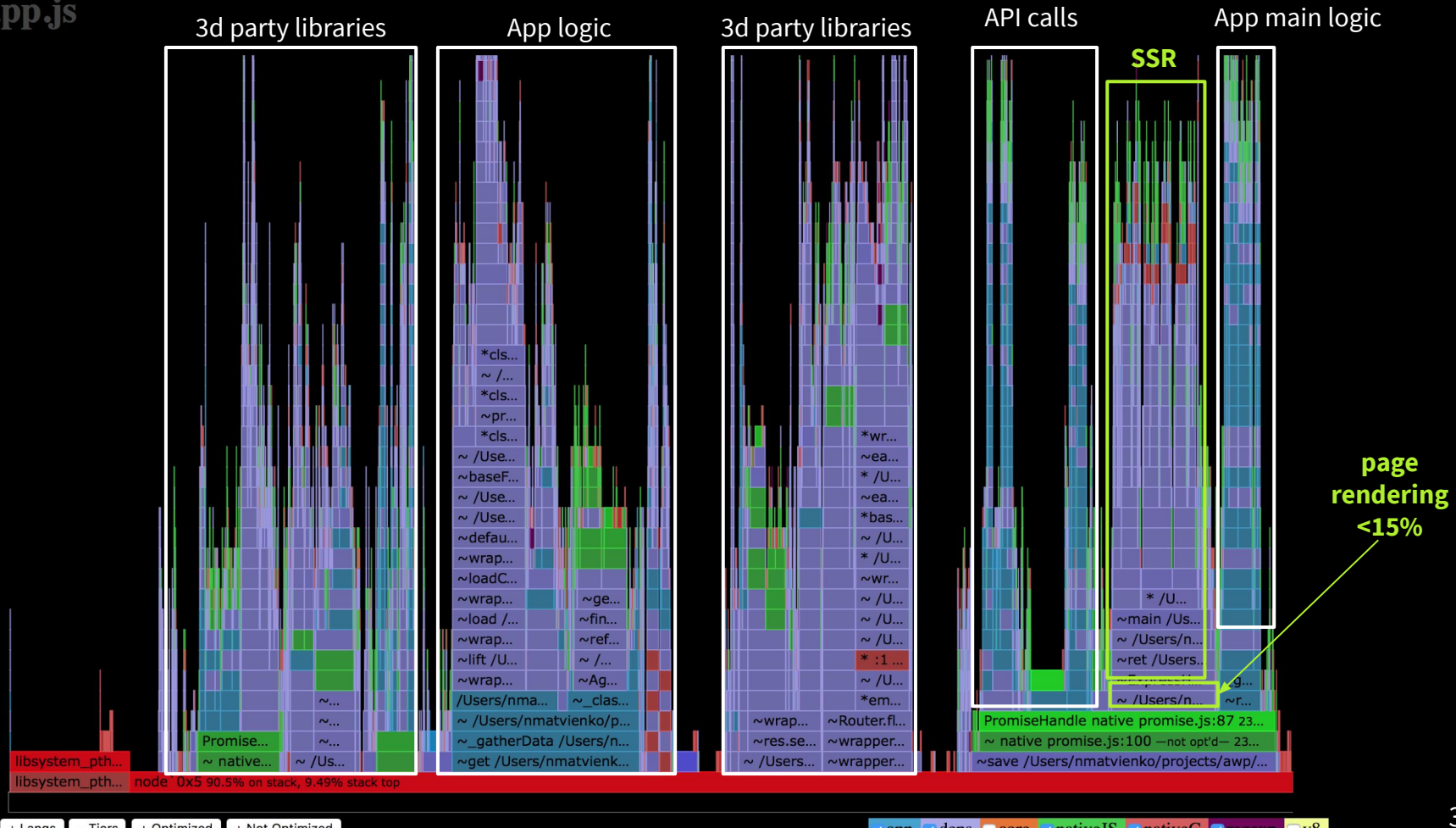


Performance improvements

After performance profiling with perf, DTrace, 0x and bcc utils we:

- Reduced CPU intensive operations:
    a. lodash objects merge, deep clone
    b. long cycles
    c. JSON parse
- Removed logic from view templates and helpers
- Reviewed and updated npm dependencies
- Gradually implemented React SSR

0x app.js



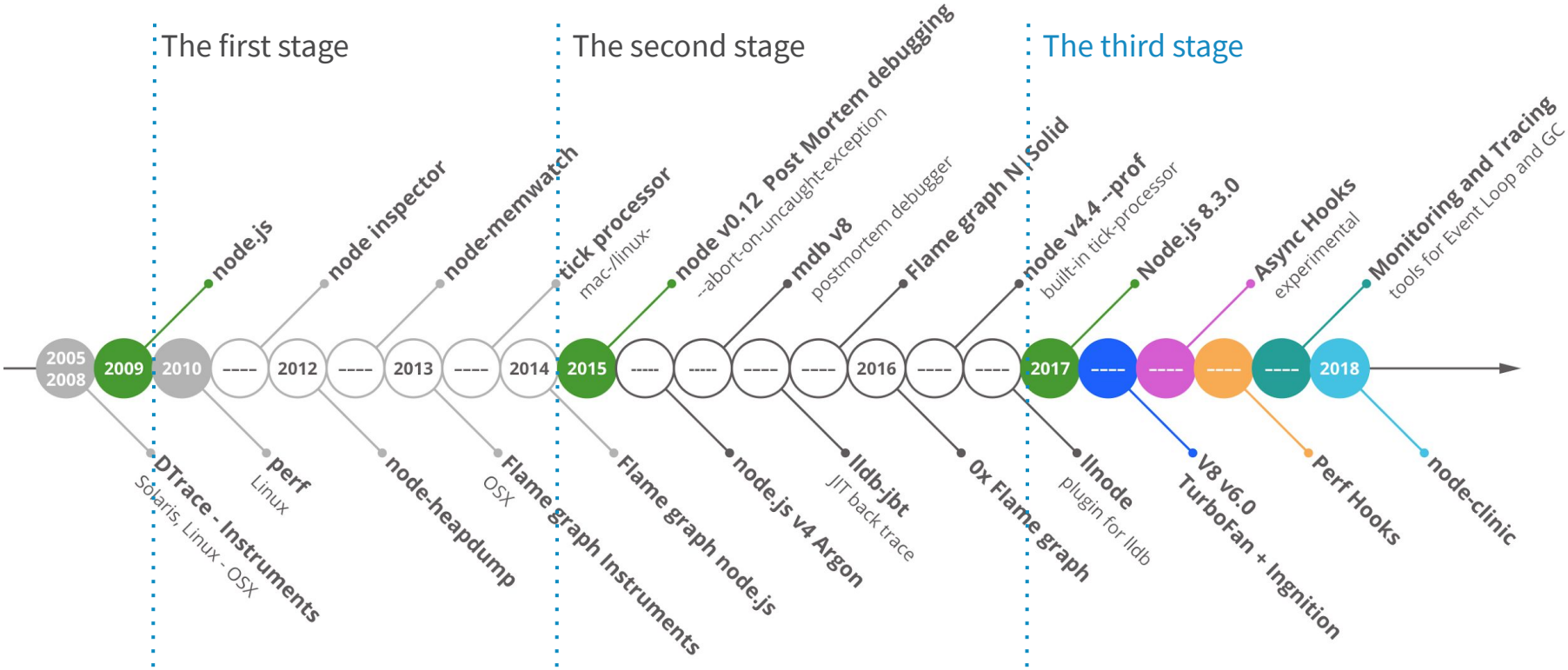
Theme | + Langs | – Tiers | + Optimized | + Not Optimized

✓ app ✓ deps ☐ core ✓ nativeJS ✓ nativeC ✓ regexp ☐ v8

Tips

1. Performance should be a part of requirements.
2. Measure performance before embedding third-party libraries and after
3. Collect measurement results archive
4. Profile on staging/pre-production environments
5. Monitor diagnostic tools performance impact

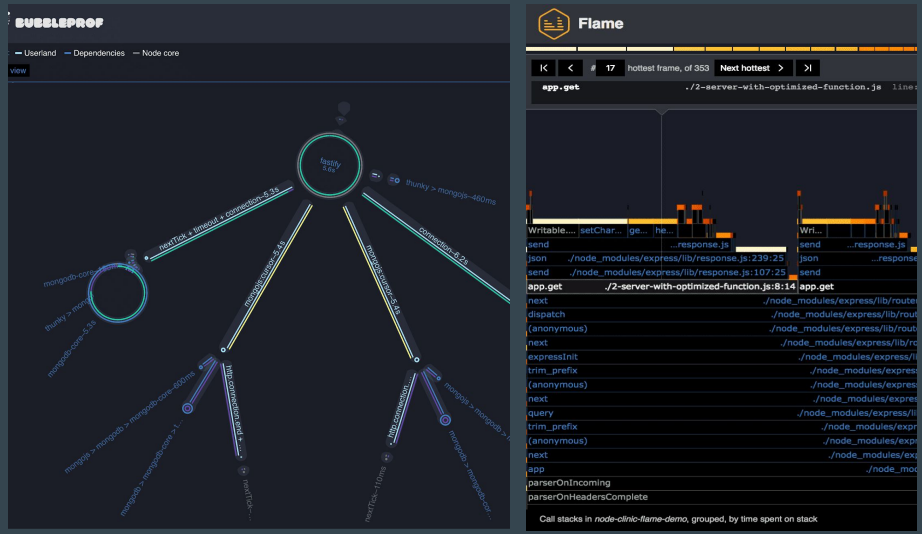
The third stage of Node.js diagnostics evolution



AUTOMATED DIAGNOSTICS TOOLS:

Node-clinic – Clinic.js (2019)

https://clinicjs.org/



"I initially didn't use Clinic.js because I thought that I could get all the results with other tools. Boy was I wrong! I was also really amazed by the UI and responsiveness of all the Clinic.js tools"

- Nikolay Matvienko, Software Engineer at Grid Dynamics

SURVEY 2017

USAGE APPLICATION

Node.js users span a variety of industries

https://foundation.nodejs.org/wp-content/uploads/sites/50/2017/11/Nodejs_2017_User_Survey_Exec_Sum.pdf

THE YEAR OF MULTITHREADED NODE.JS PLATFORMS

1. Napa.js
   https://github.com/microsoft/napajs
2. AliOS-nodejs
   https://github.com/alibaba/AliOS-nodejs/wiki/Workers-in-Node.js-based-on-multithreaded-V8
3. Ayo
   https://github.com/ayojs/ayo

USER FEEDBACK SESSION WG → 2018

WORKER-THREADS

RELEASE & FIRST USAGE

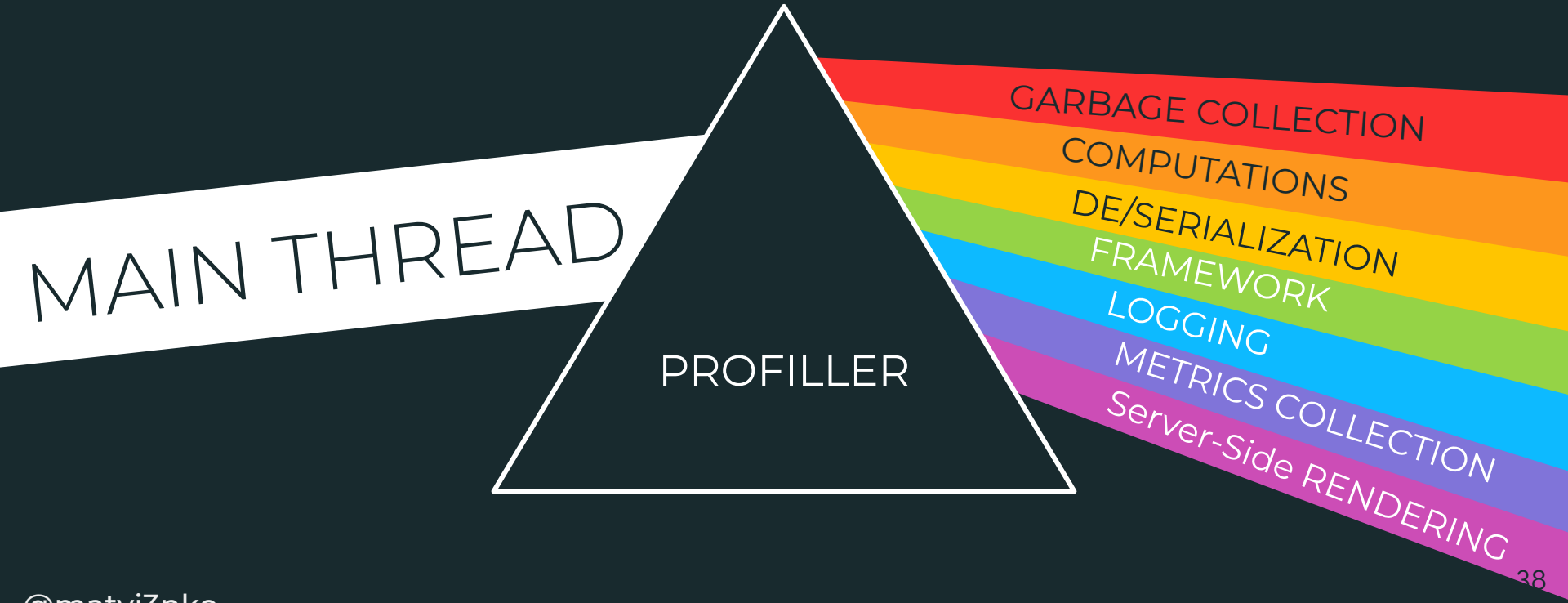
Patric-robot & Anna

Decomposition of the Main Thread in Node.js
https://youtu.be/Mfz1_blLl9Q

WORKER

[FEATURE REQUEST]

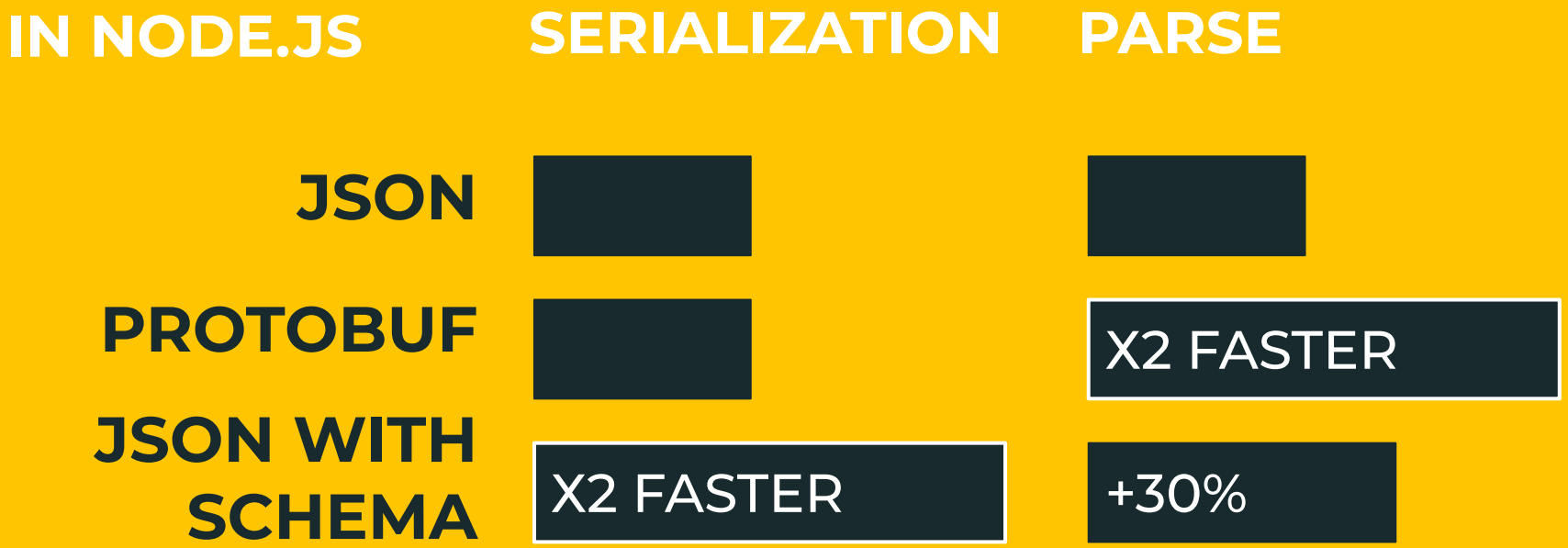
DISPERSION


MAIN THREAD
PROFILLER
GARBAGE COLLECTION
COMPUTATIONS
DE/SERIALIZATION
FRAMEWORK
LOGGING
METRICS COLLECTION
Server-Side RENDERING

EACH REQUEST

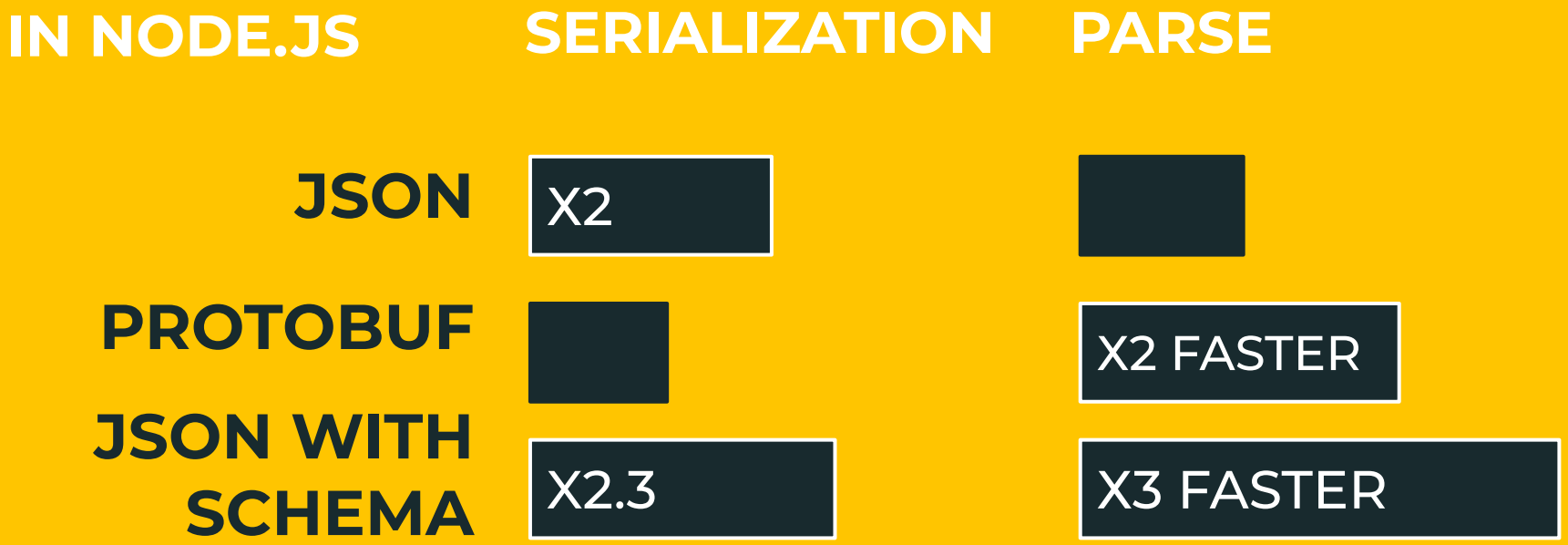
USERS PAY FOR LOGGING, GC AND METRICS COLLECTIONS ... WITH THEIR TIME.

NUMBERS



*Average values in JavaScript environment.
See libraries in resources.

STRINGS

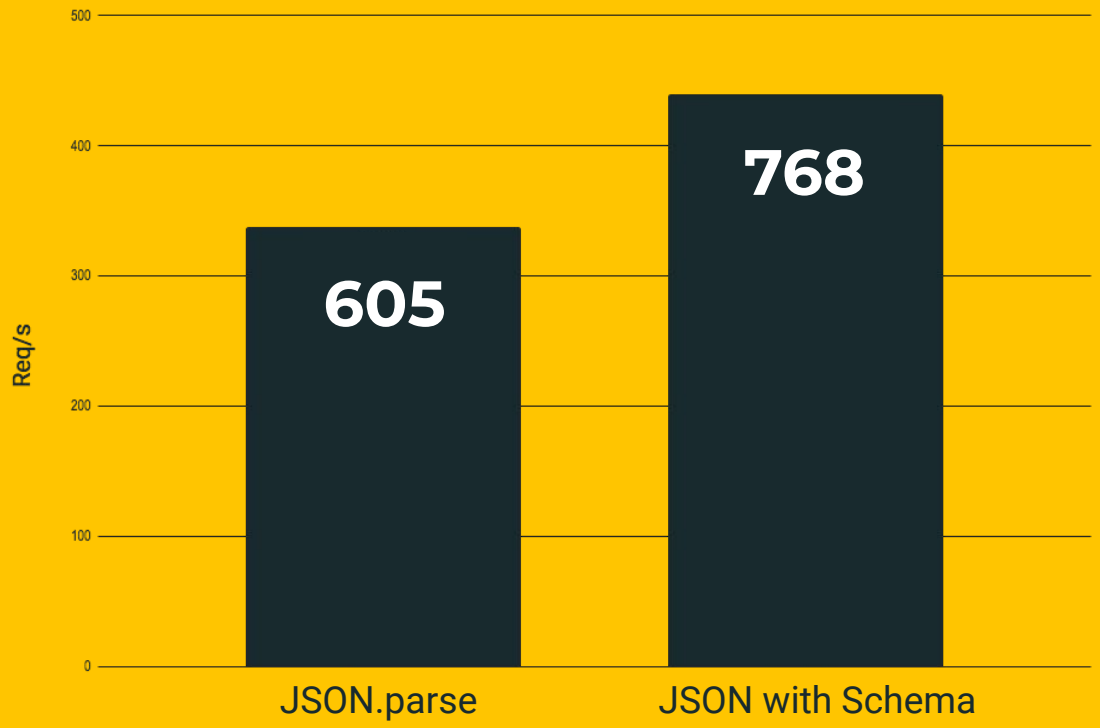


*Average values in JavaScript environment.
See libraries in resources.

JSON.PARSE CHANGES RESULT

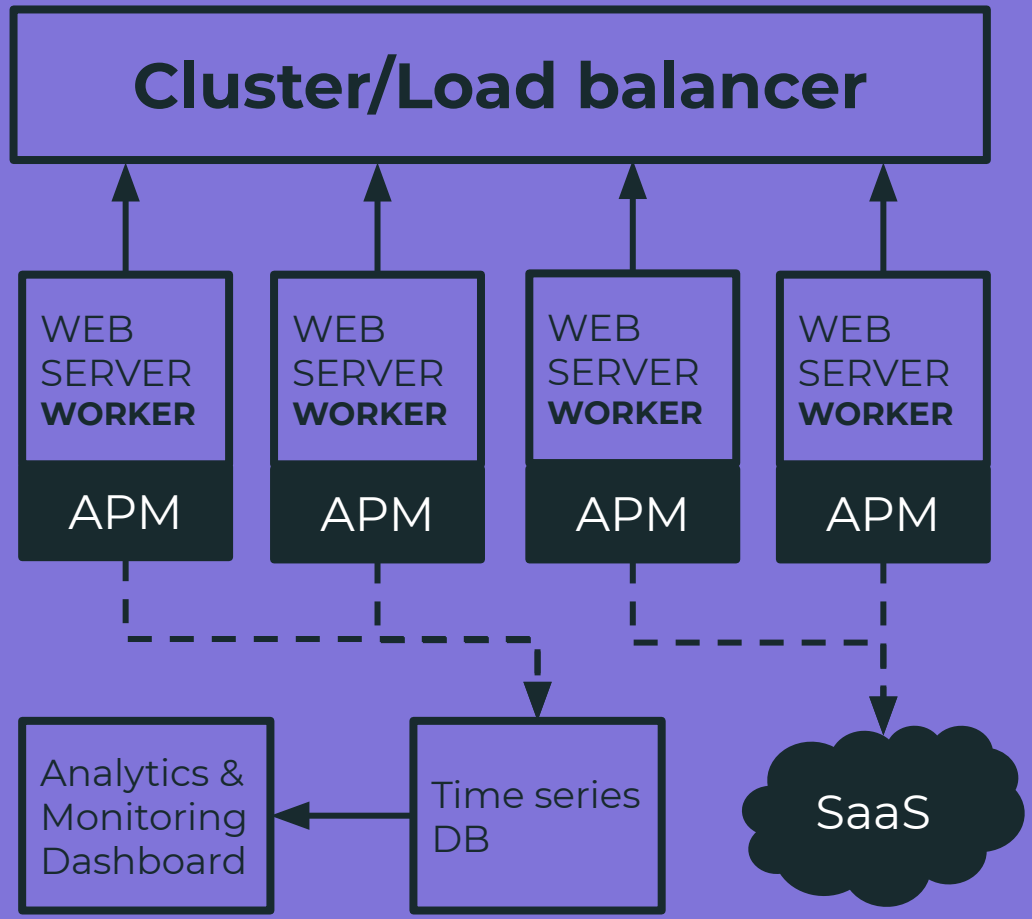
+27%

MORE REQ/S
WITH
FAST-JSON-STRINGIFY
&
JITSON/
TURBO-JSON-PARSE

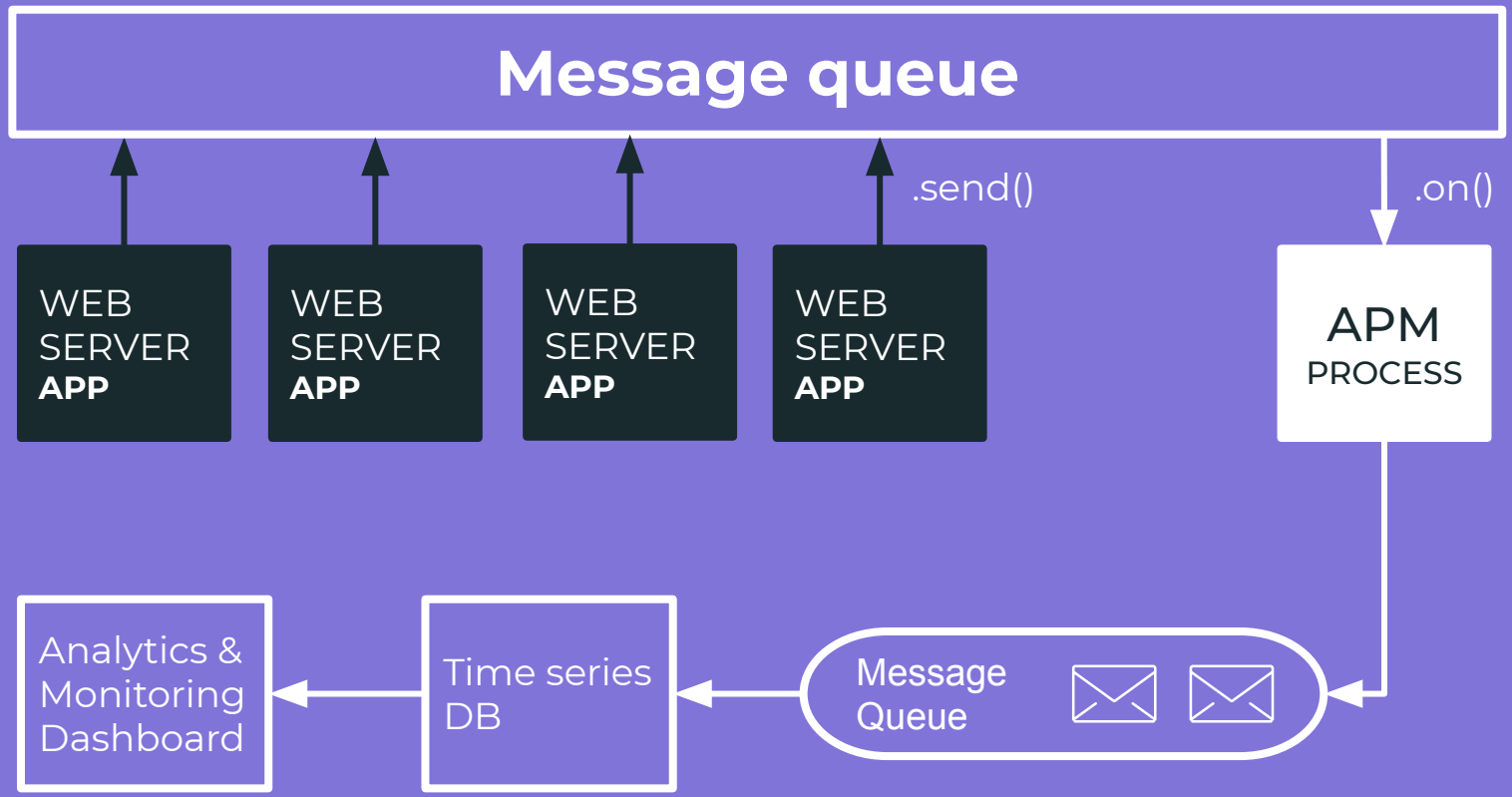


IN-PROCESS APM AGENT

THE APM AGENT PROBLEMS
ARE APPLICATION PROBLEMS



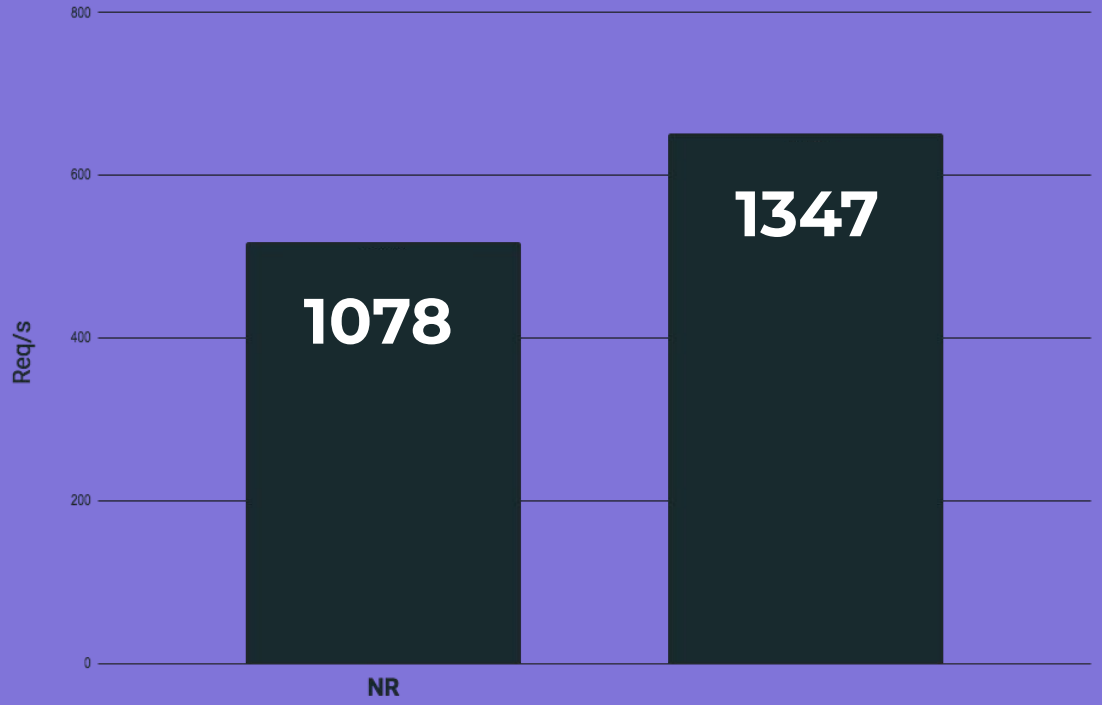
OFF-PROCESS APM AGENT



OFF-PROCESS MONITORING RESULT

+25%

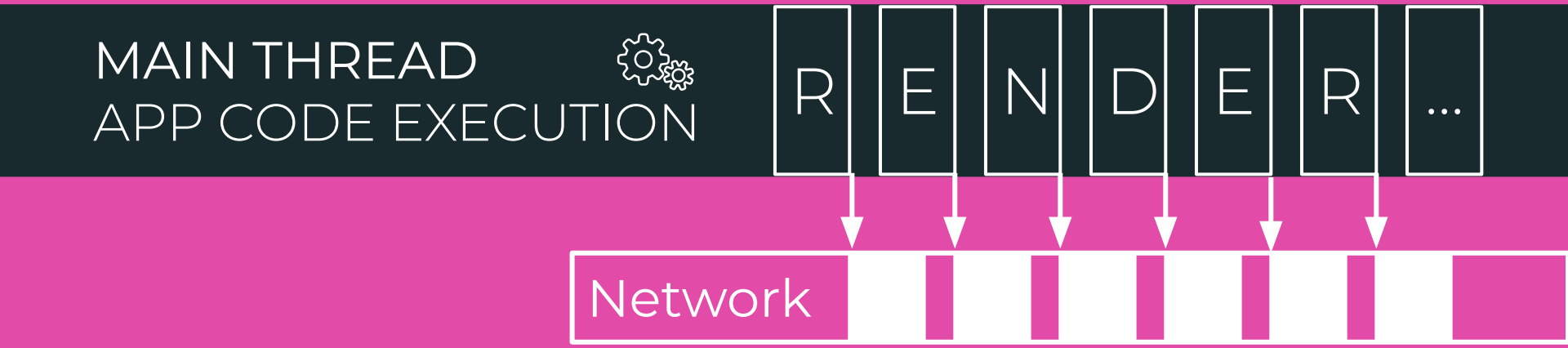
MORE REQ/S
WITH
OFF-PROCESS
METRIC AGENT



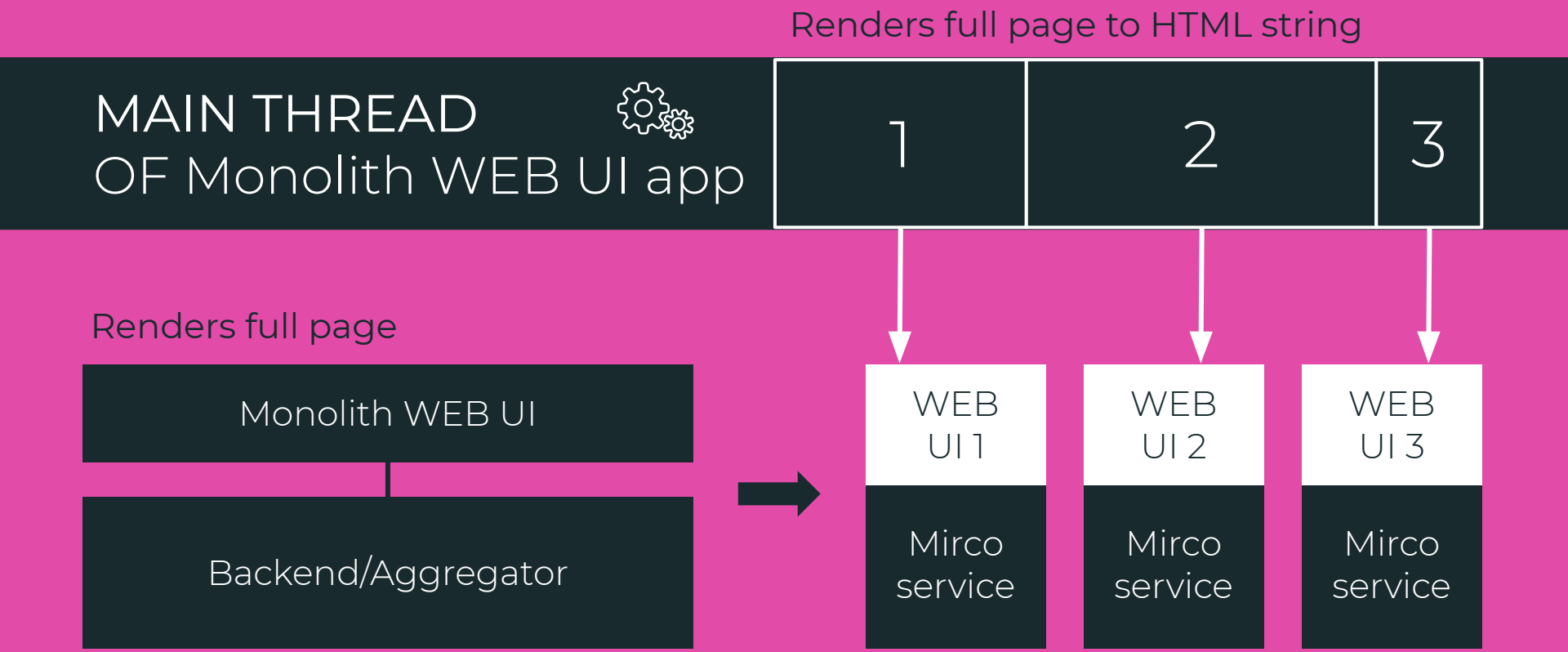
STREAMING SERVER-SIDE RENDERING

```
renderStream.pipe(res, { end: 'false' });
renderStream.on('end', () =>
{response.end('</div></body></html>'); });
```

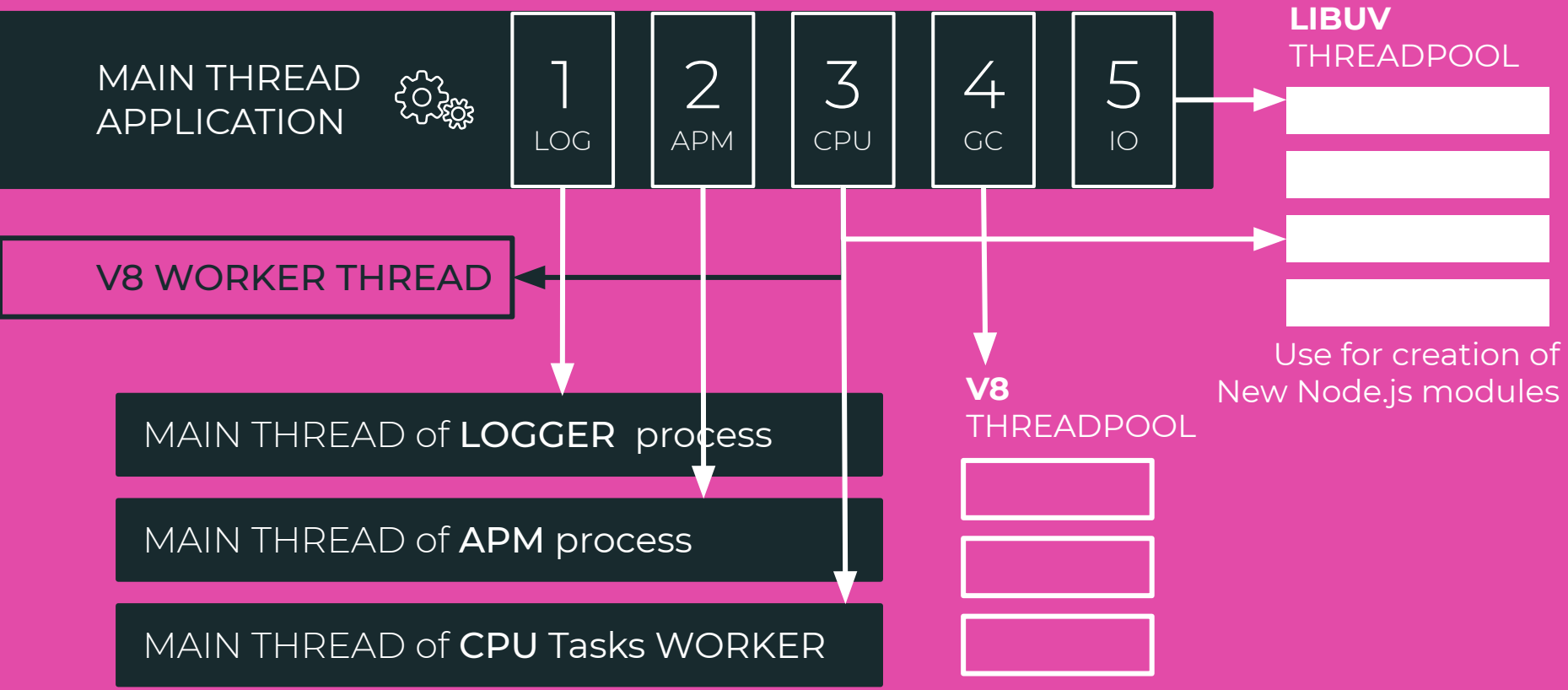
Asynchronous execution in STREAM with **REACT 16**



MICRO FRONTENDS



DECOMPOSED MAIN THREAD



REACT 16

2X

* Average value.

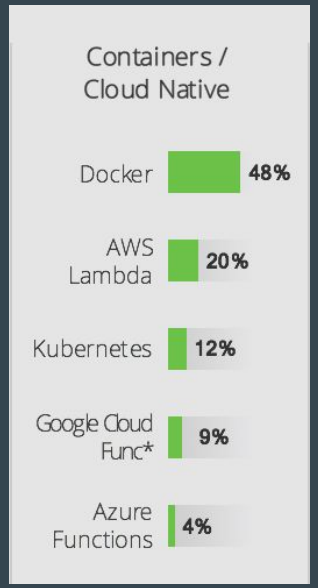
THROUGHPUT

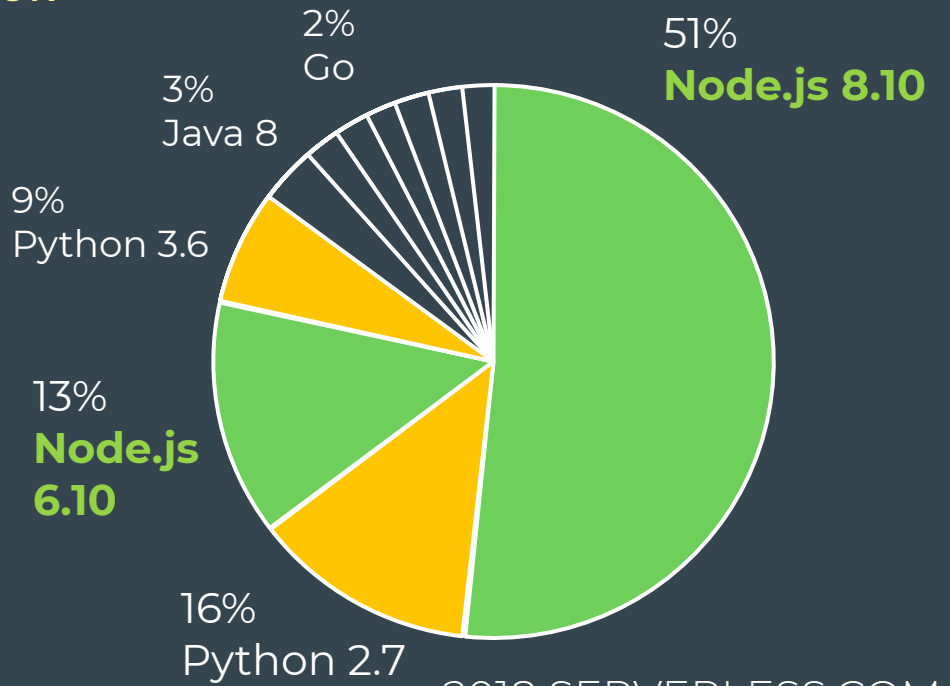
FAAS NOT VULNERABLE FOR
CPU intensive tasks

~75%
DEPLOYS TO CLOUD

and 33%
IS FAAS / SERVERLESS

BIG DATA ANALYTICS





2018 SERVERLESS.COM

https://blog.risingstack.com/takeaways-node-js-survey-2018/

2018

SERVERLESS COMPUTE SERVICE

AWS **LAMBDA**

SERVERLESS FUNCTION

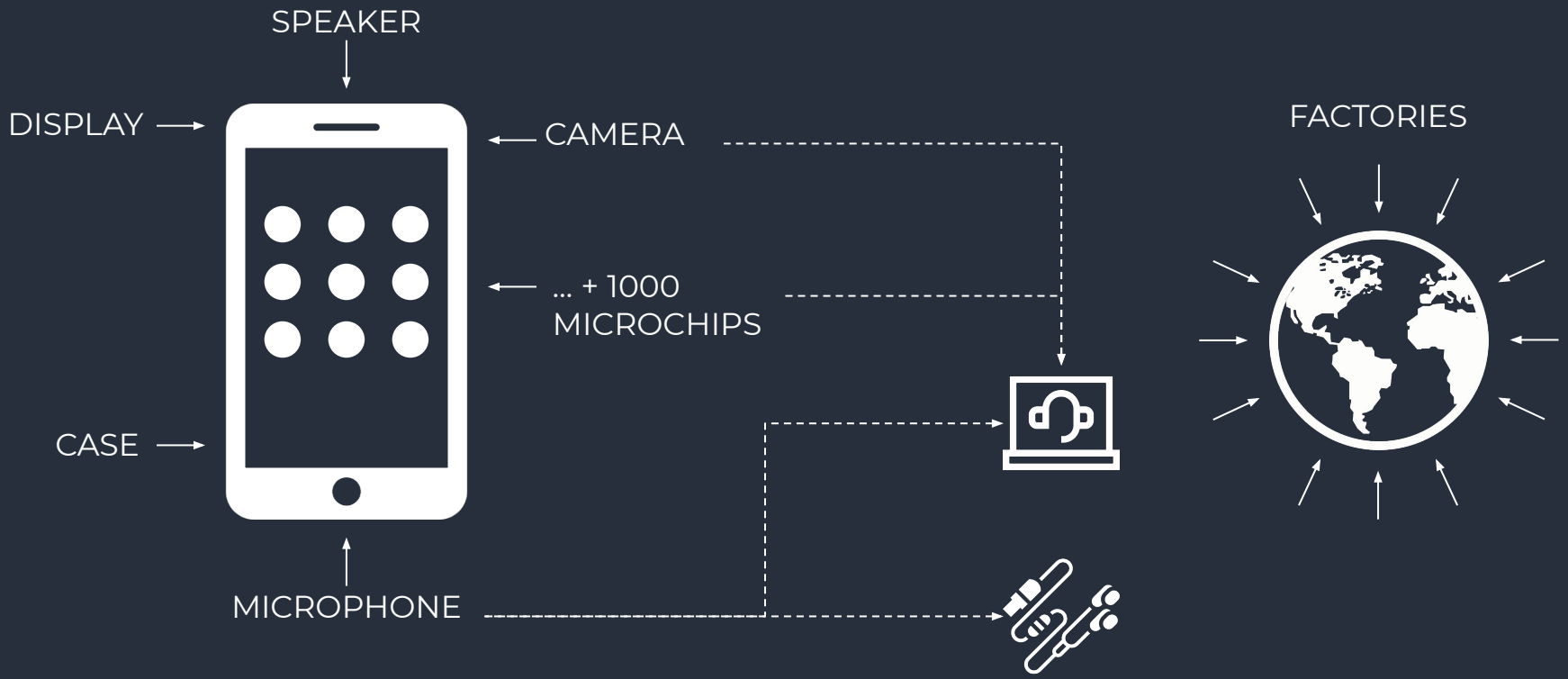
```
export const handler = async (event) => {
 const data = event.Records[0].body;

 // - TRANSFORM data
 // - WRITE to DB or
 // - PUT TO QUEUE/STREAM
 return 'success';
};
```

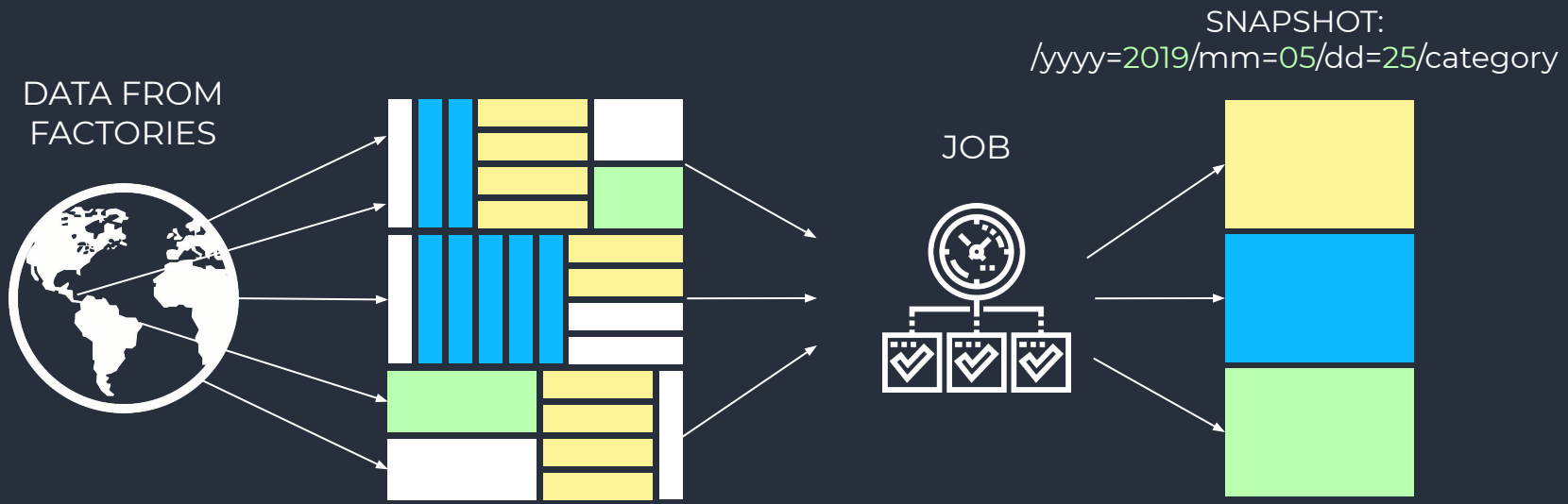
- CHOOSE YOUR CODE LANGUAGE
- NO INFRASTRUCTURE TO MANAGE
- TRIGGERED BY EVENTS
- HIGH SCALABLE
- STATELESS
- COST-EFFECTIVE

* Symbol in the presentation

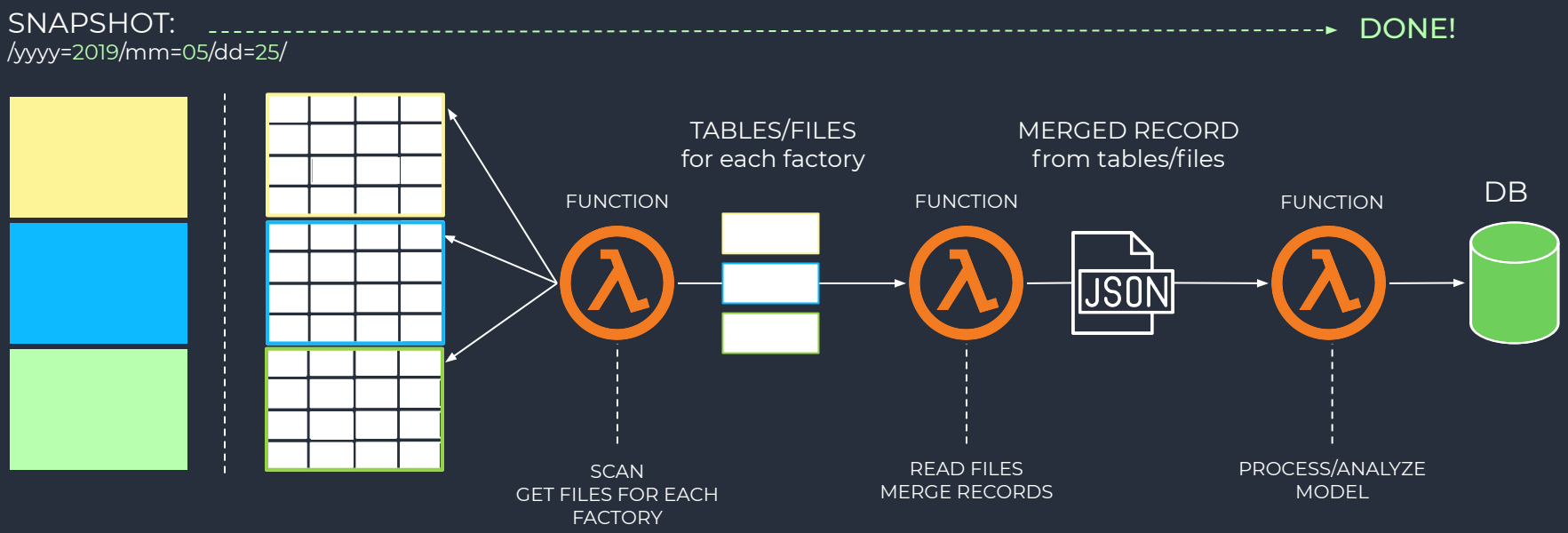
EXAMPLE


SPEAKER
DISPLAY
CAMERA
... + 1000
MICROCHIPS
CASE
MICROPHONE
FACTORIES

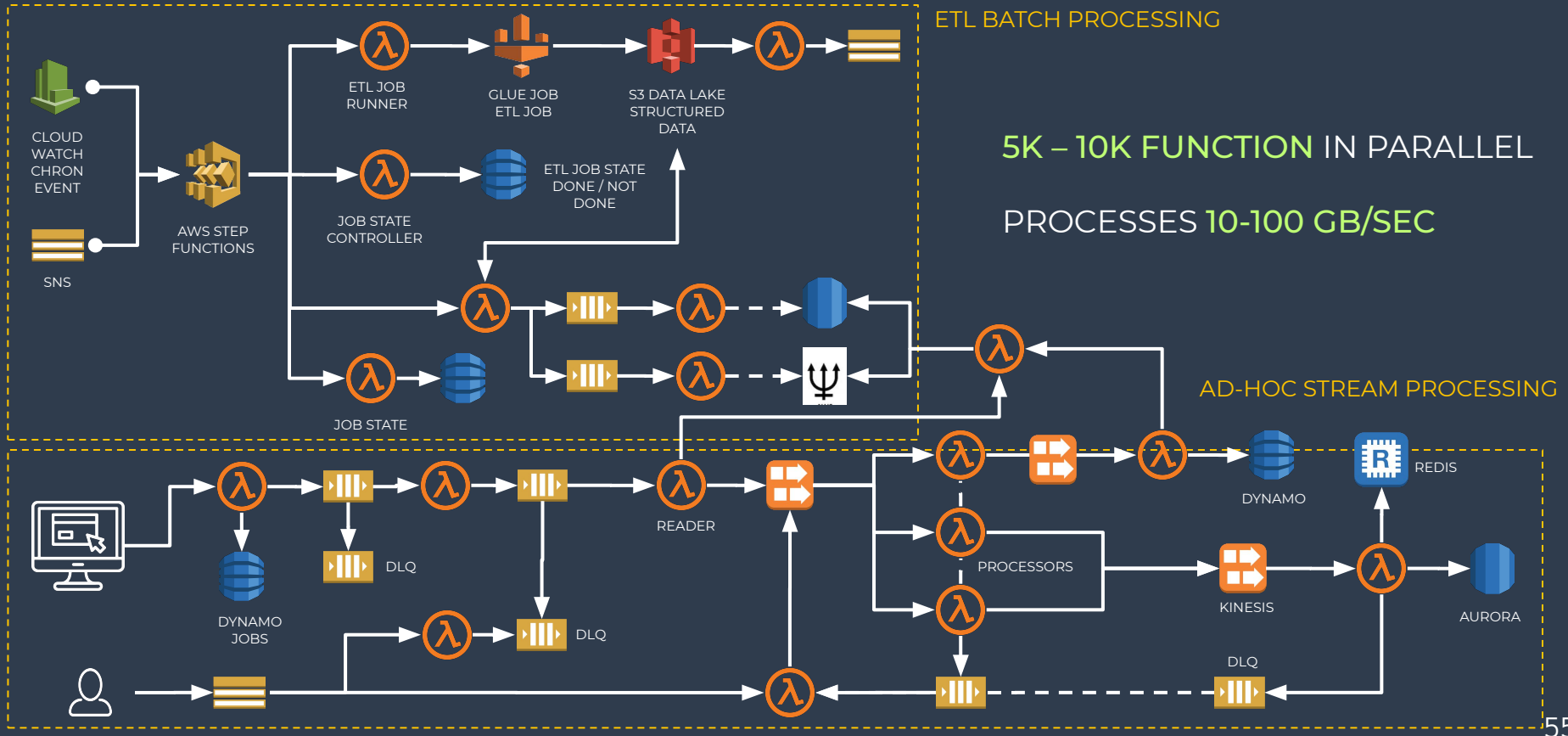
BATCH PROCESSING



BATCH DATA PROCESSING

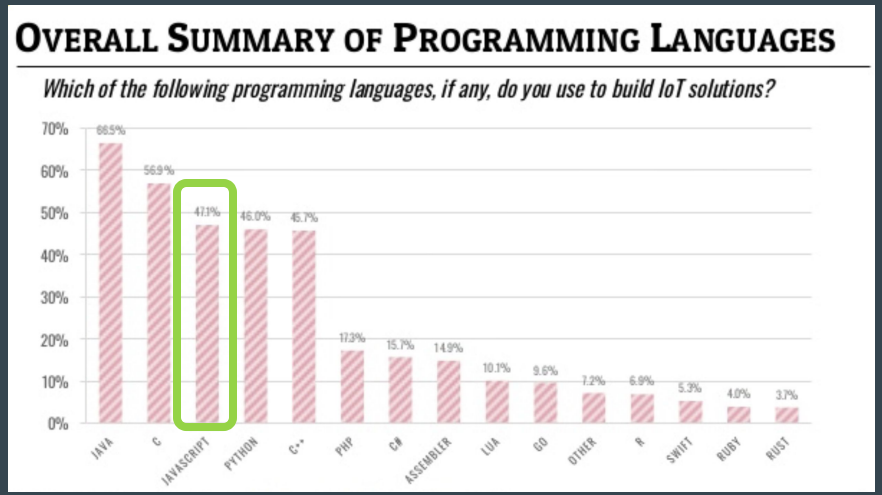


BLUEPRINT

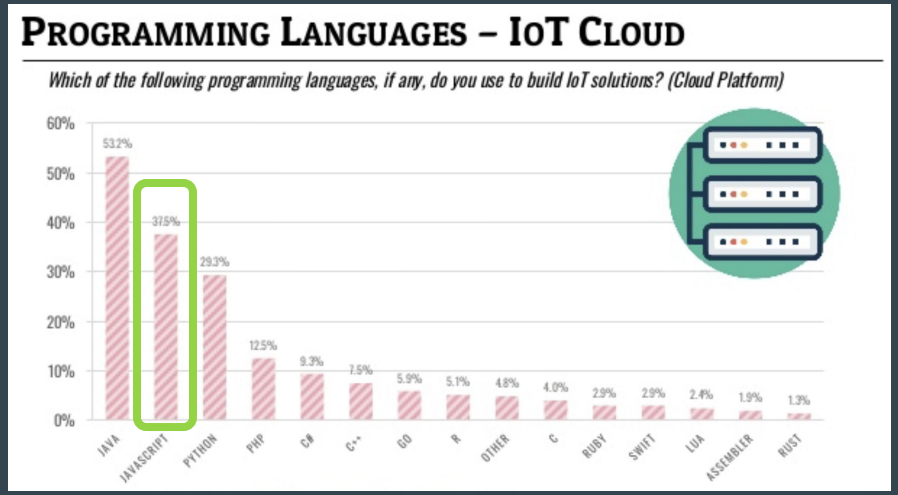

ETL BATCH PROCESSING
CLOUD WATCH CHRON EVENT
SNS
AWS STEP FUNCTIONS
ETL JOB RUNNER
GLUE JOB ETL JOB
S3 DATA LAKE STRUCTURED DATA
JOB STATE CONTROLLER
ETL JOB STATE DONE / NOT DONE
JOB STATE
5K – 10K FUNCTION IN PARALLEL
PROCESSES 10-100 GB/SEC
AD-HOC STREAM PROCESSING
DYNAMO JOBS
DLQ
DLQ
READER
PROCESSORS
DYNAMO
REDIS
KINESIS
AURORA
DLQ

IoT PLATFORMS

THE 3RD PLACE OVERALL



THE 2ND PLACE ON CLOUD



IoT Developer Survey 2018. Eclipse Foundation

https://www.slideshare.net/kartben/iot-developer-survey-2018

& NEXT 2019

- Startup time increased + new proposals
- Memory snapshots (core userland)
  https://github.com/joyeecheung/talks/blob/master/openjs_collab_summit_201905/bootstrap-of-node-core.pdf
- Diagnostics tools in the core
- TypeScript integration discussion

1st place most popular framework tool, instruments

Node.js 50% (.Net 38%, .Net Core 24%)

1st place most wanted technology

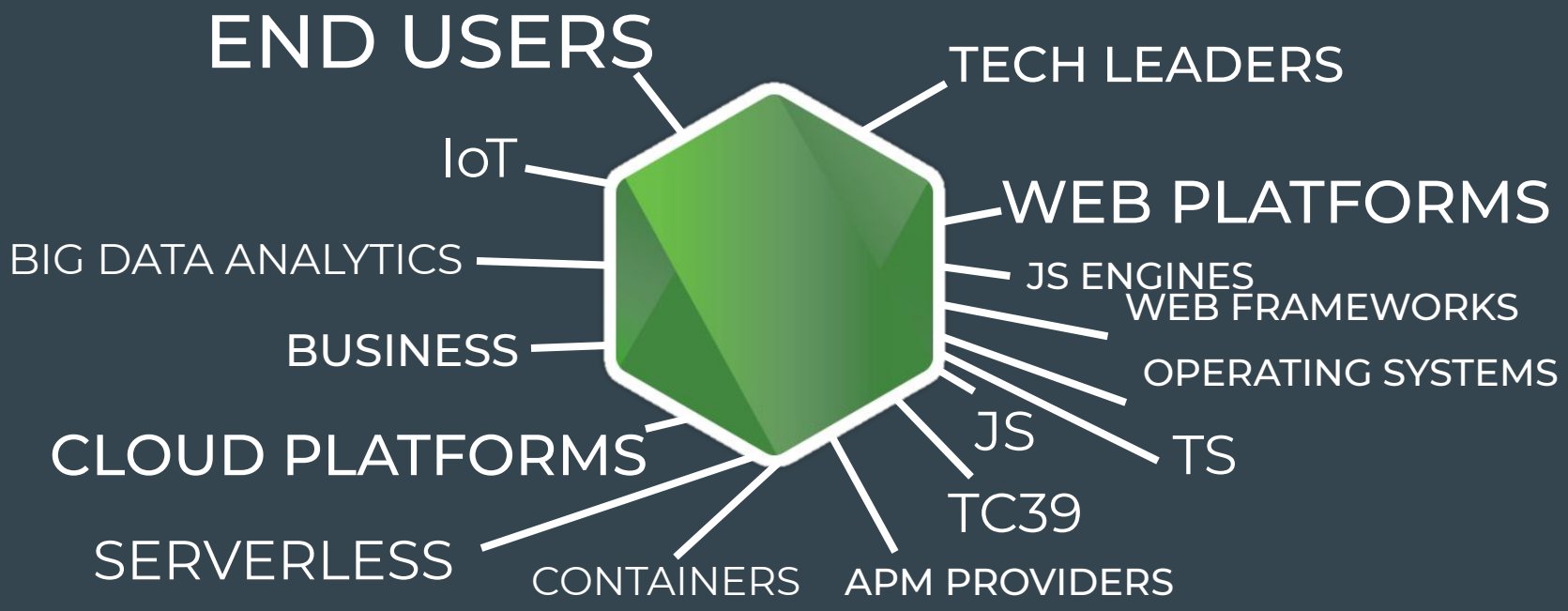
Node.js 17% (TensorFlow 16%, React Native 13%)

https://insights.stackoverflow.com/survey/2019

GD CLIENTS


News Corp
PayPal
Pinterest
UBER
Capital One
Moz://a
Angie's list
LOWE'S
Joyent
Linked in
Walmart
THE NEW YORKER
Zappos
Google
intuit
Goldman Sachs
GROUPON
CBS
NETFLIX
GoPro
The New York Times
Go Daddy
Microsoft
Symantec
DOW JONES
SIEMENS
ADP
NASA
cisco
NORDSTROM
amazon
Medium
ebay
airbnb
Yandex
Johnson Controls
SONY
QUALCOMM
WiX
citi
DuckDuckGo
IBM
redhat
LogMeIn
Quizlet
SAMSUNG
SKYCATCH
coursera
fitbit
CONDÉ NAST
TARGET
Trello
yammer
change.org
BUSTLE
intel
YAHOO!
OpenTable
RBS
Aol.
LendingClub
Cigna
pwc
flickr
SAP
Fidelity
Quad
simplereach
Koding
NodePing
MuleSoft
despegar.com
asana
shutterstock

DIRECTION AND FUNCTIONALITY



THANKS!

Nikolay Matvienko

matvi3nko@gmail.com

Twitter.com/**matvi3nko**

github.com/**matvi3nko**