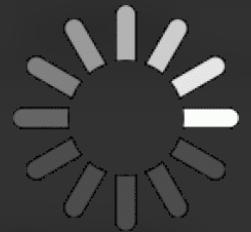


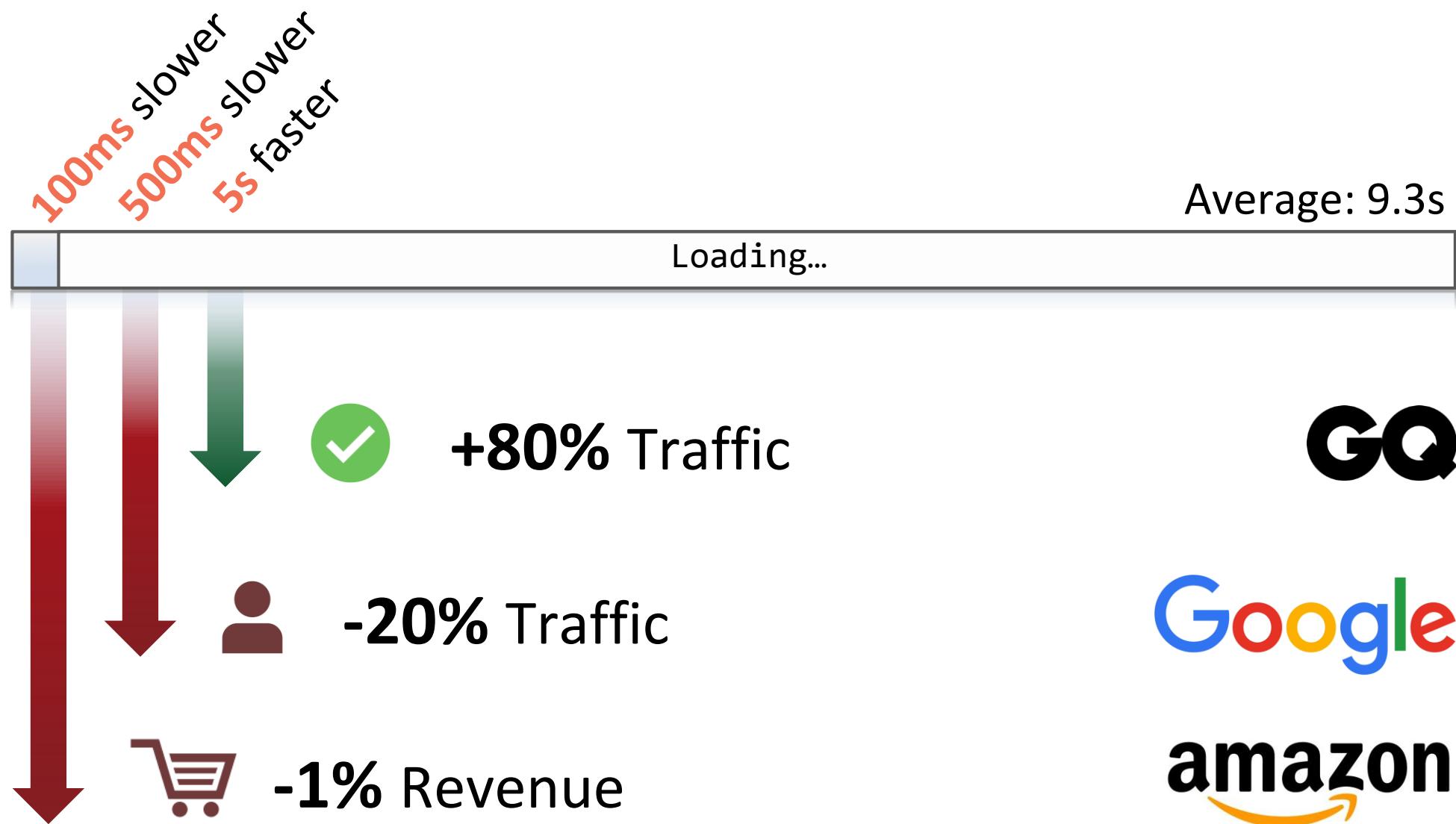


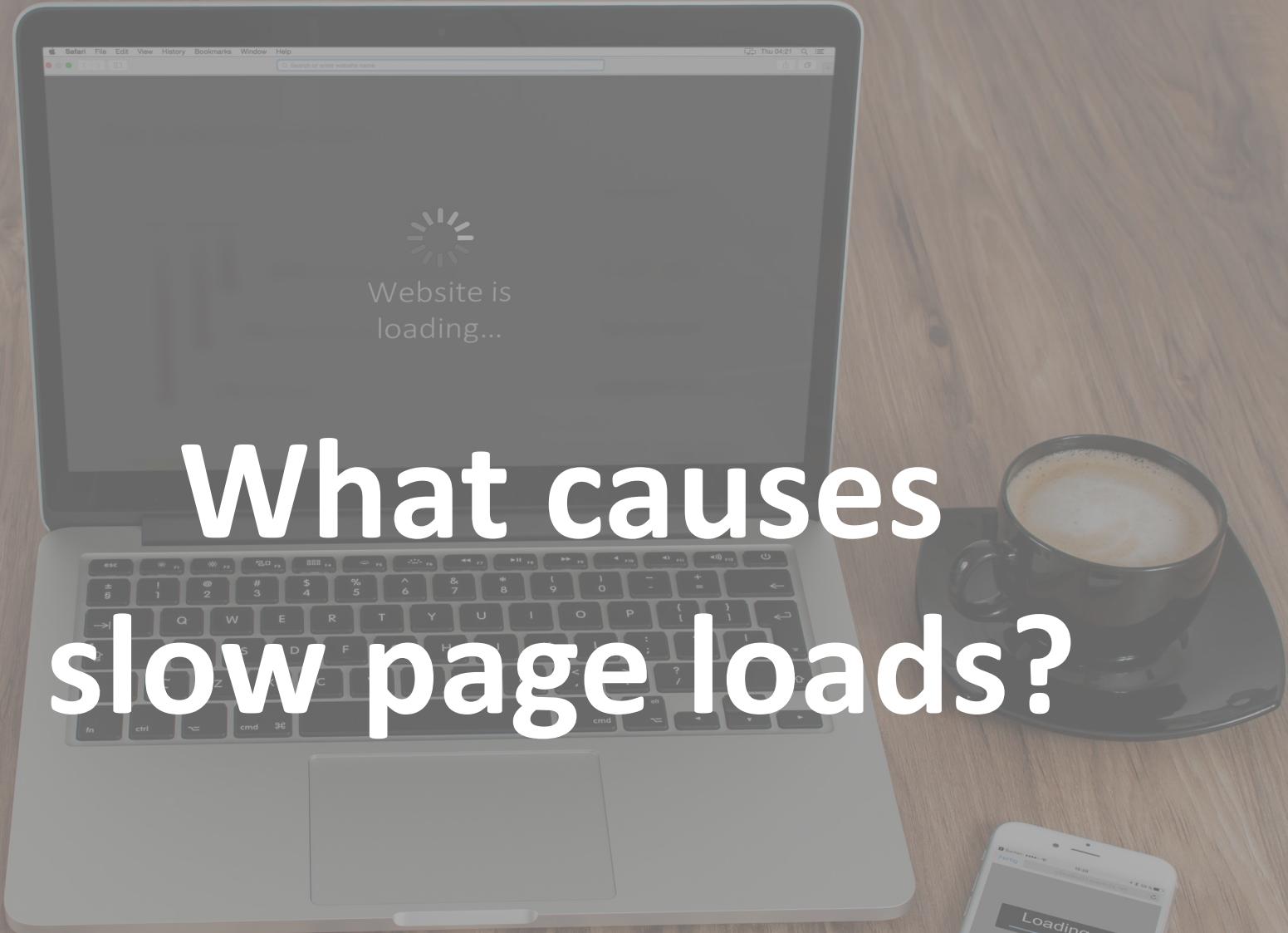
**Make the web faster.**



Presentation  
is loading

# Page Load Time = Money

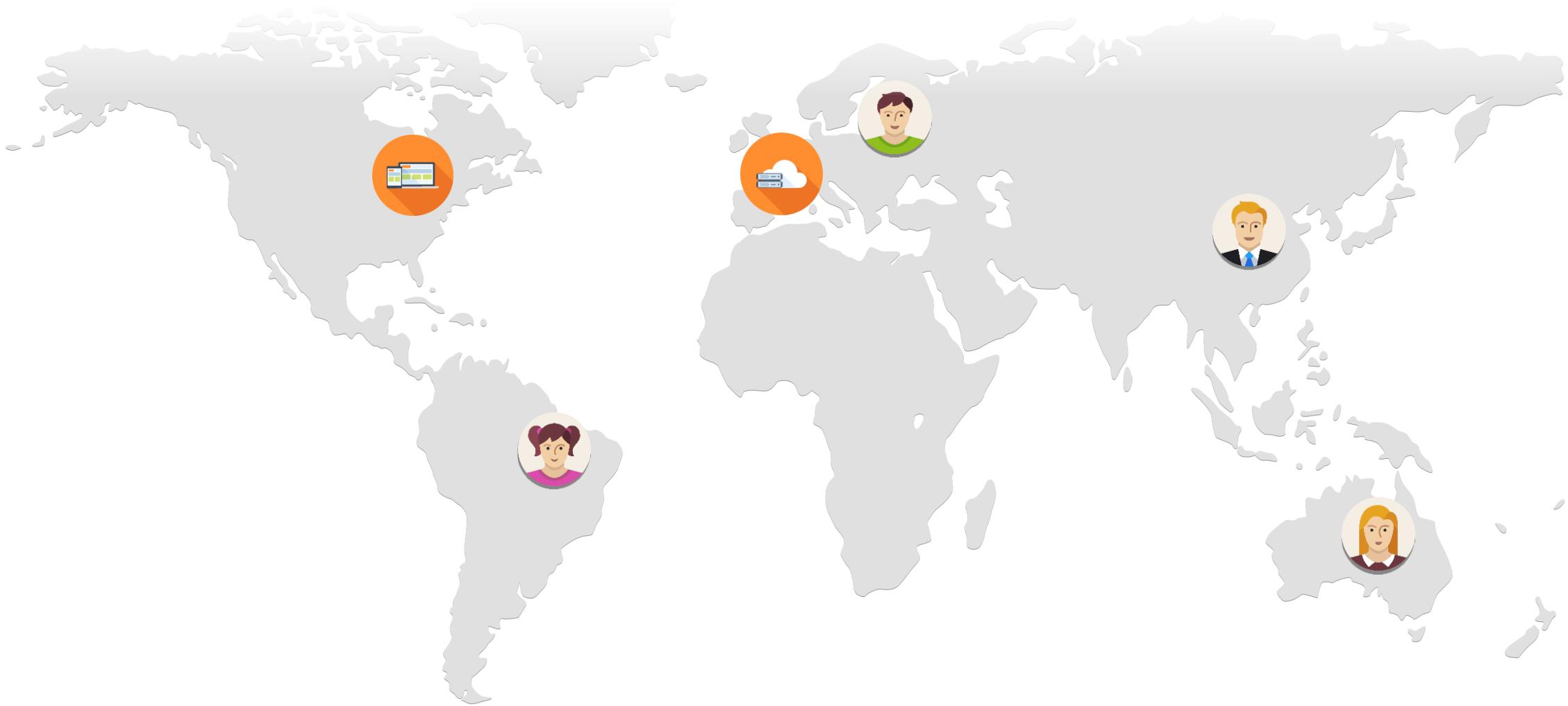




# What causes slow page loads?

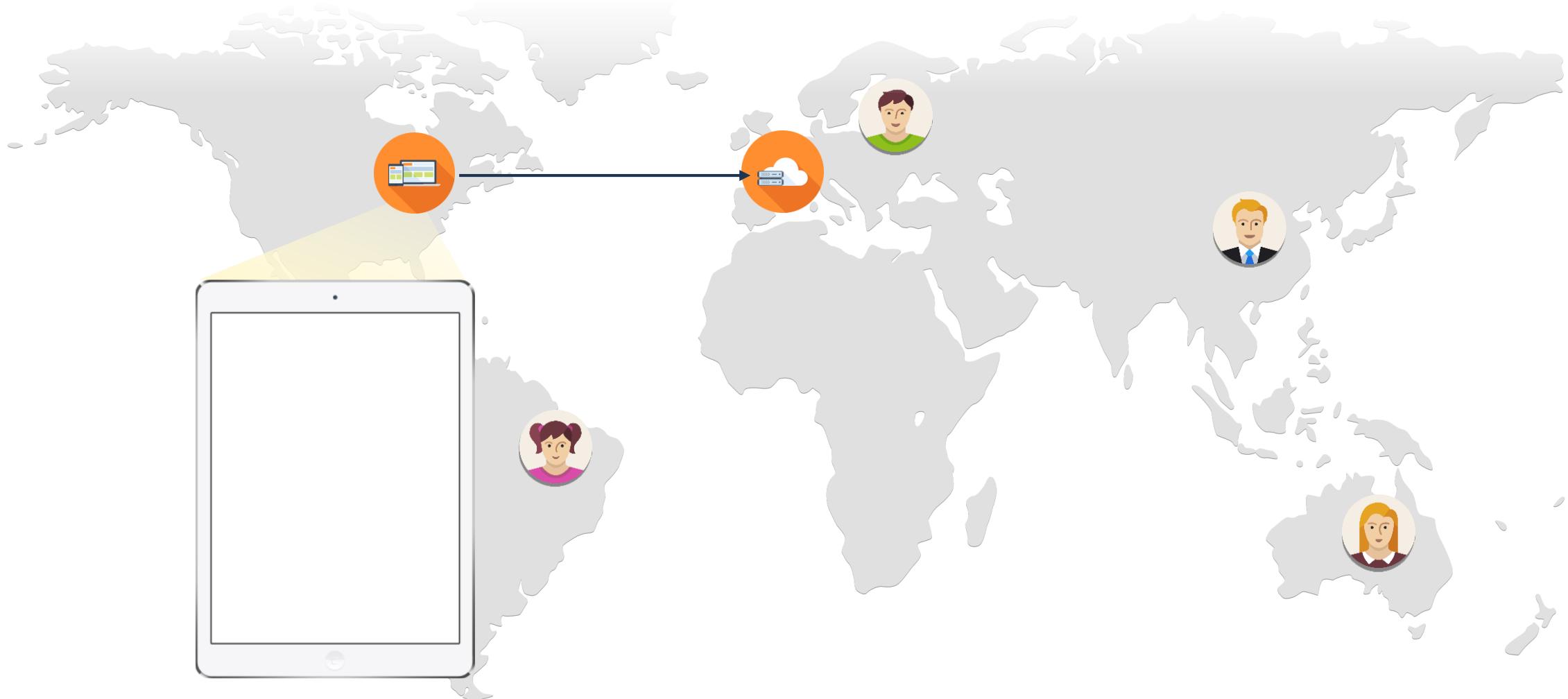
# There are 2 performance problems.

This is what makes websites **slow**.



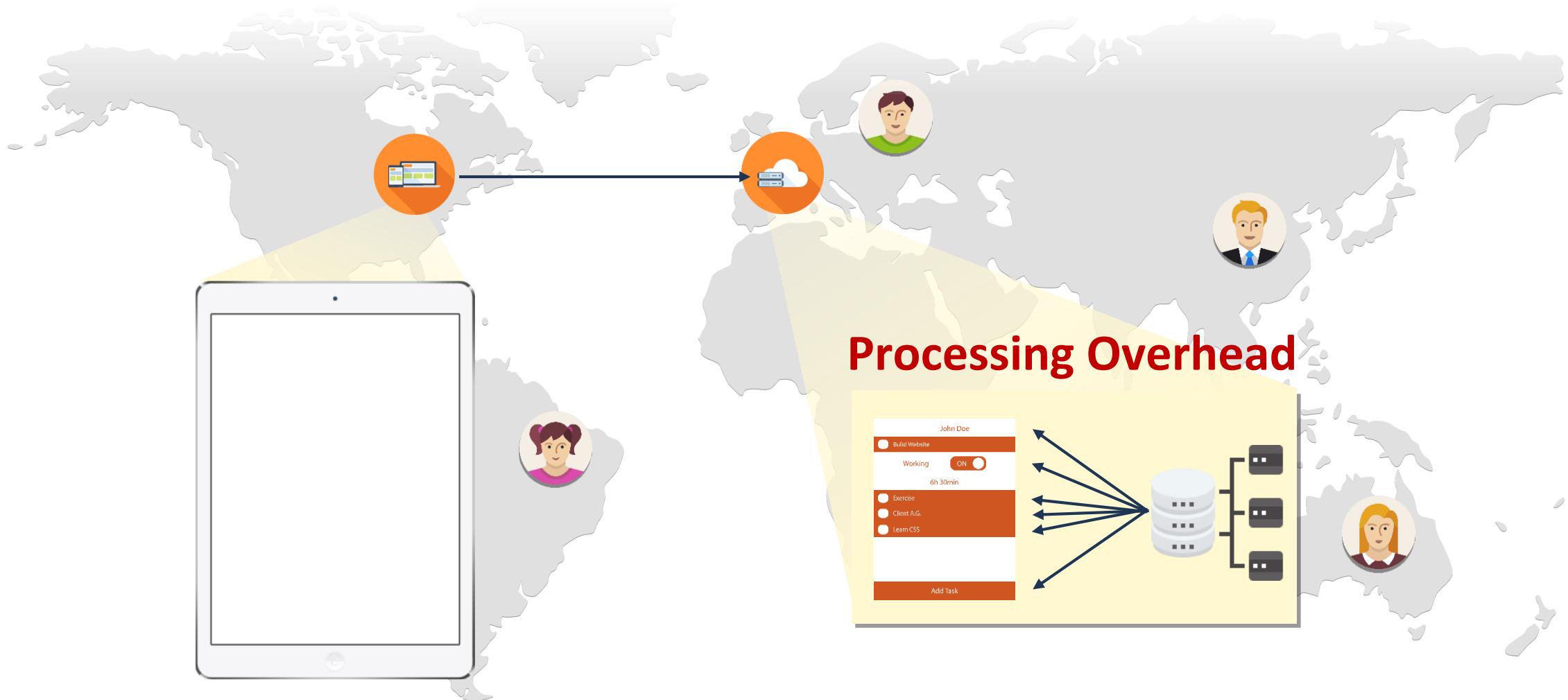
# There are 2 performance problems.

When a user visits a website, the **backend** is requested.



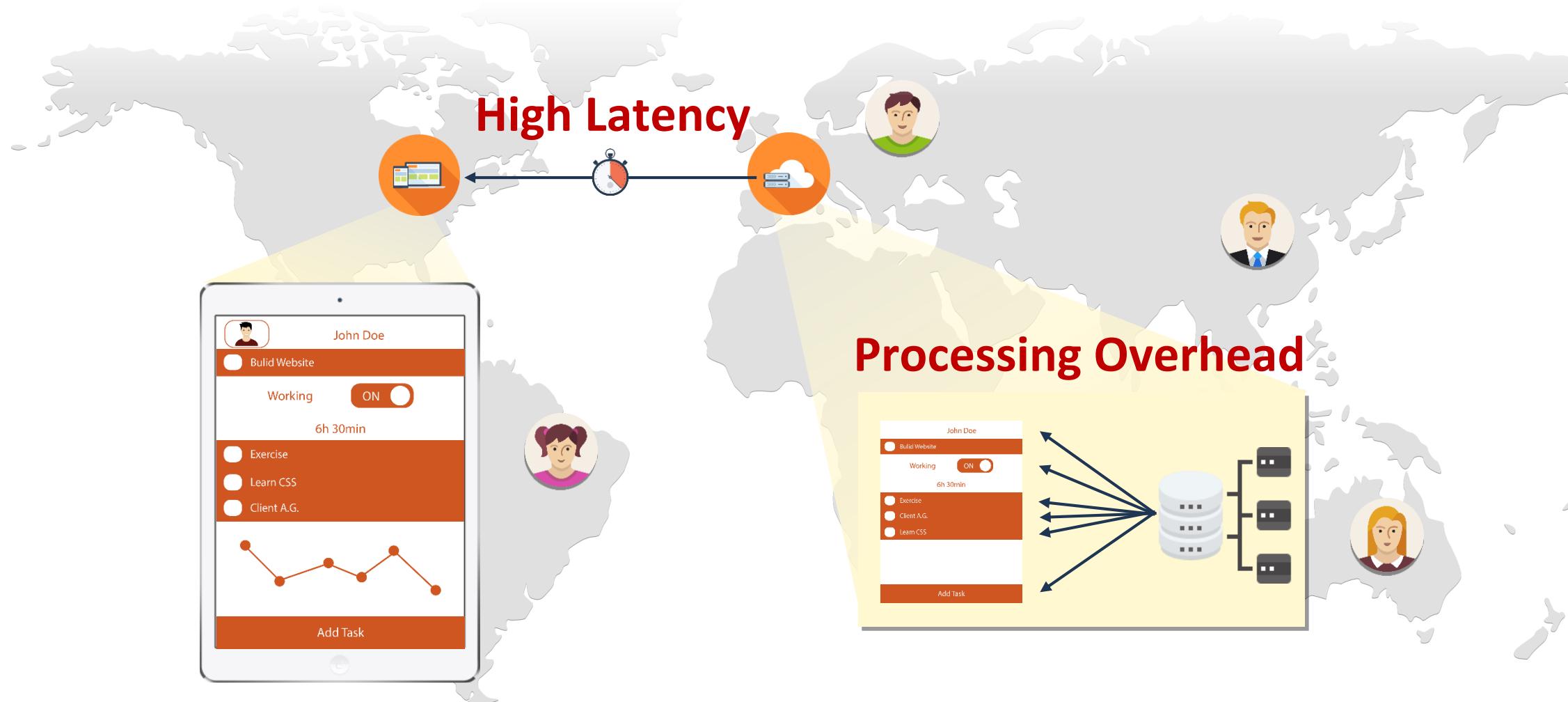
# First bottleneck: the backend.

The backend renders the site, causing **processing** overhead.



# Second bottleneck: the network.

>100 resources are sent to the user, causing **network delay**.



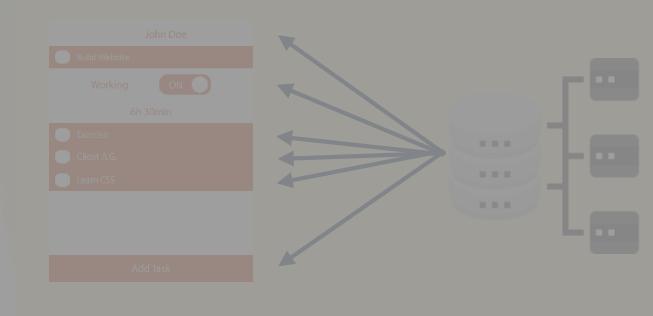
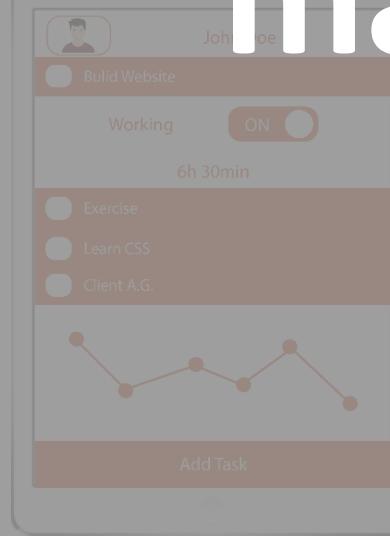
# There are 2 performance problems.

>100 resources are sent to the user, causing **network delay**.

High Latency  
**How can we**

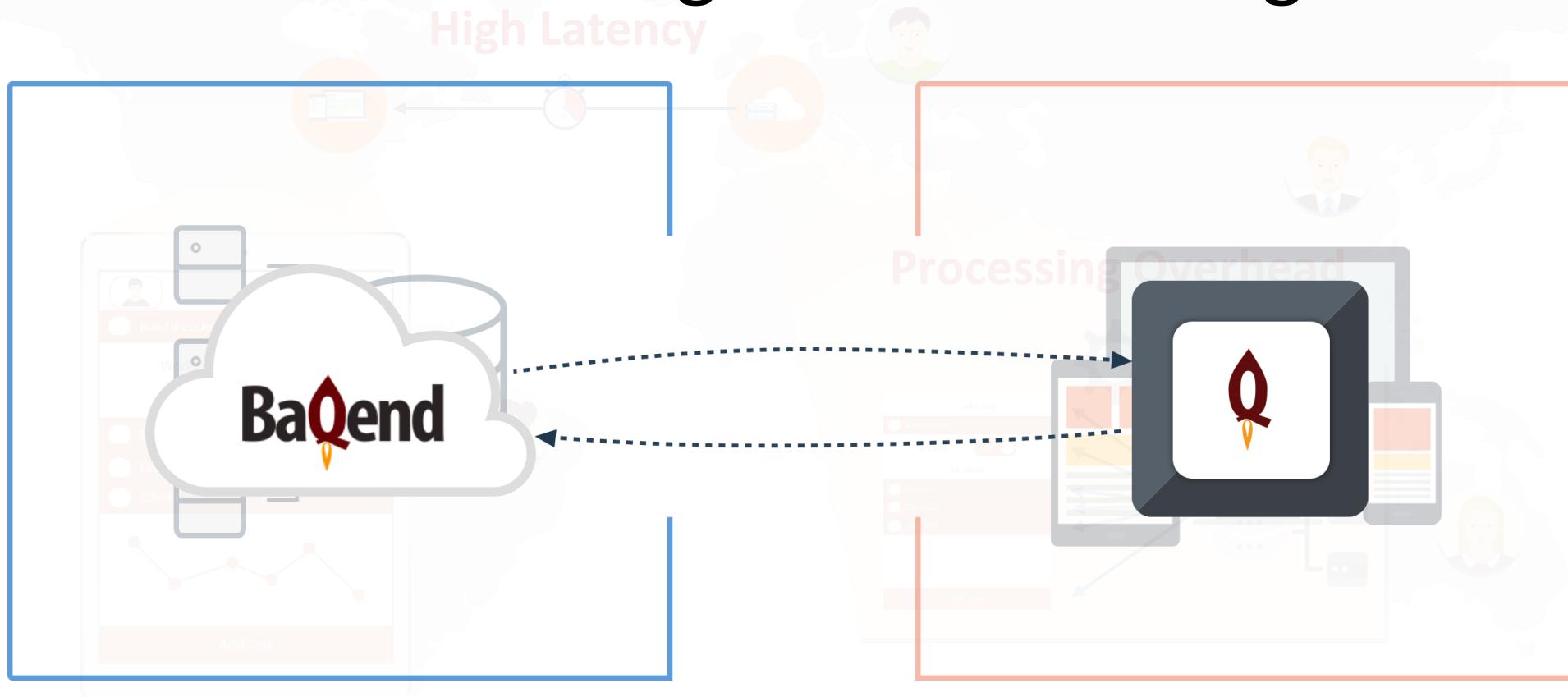
**make this FASTER?**

Processing Overhead



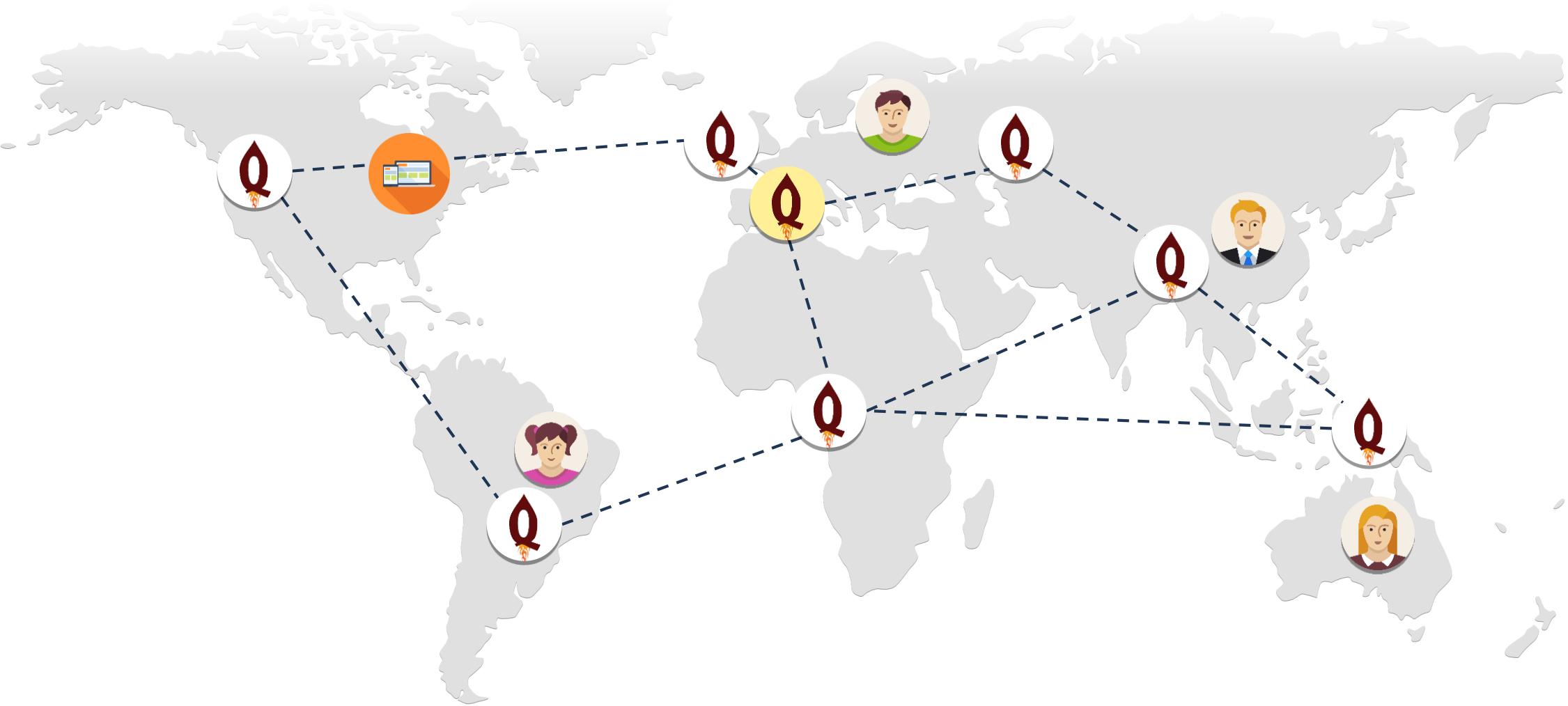
# Speed Kit

## Making **Existing Websites** Load Fast through Better Caching.



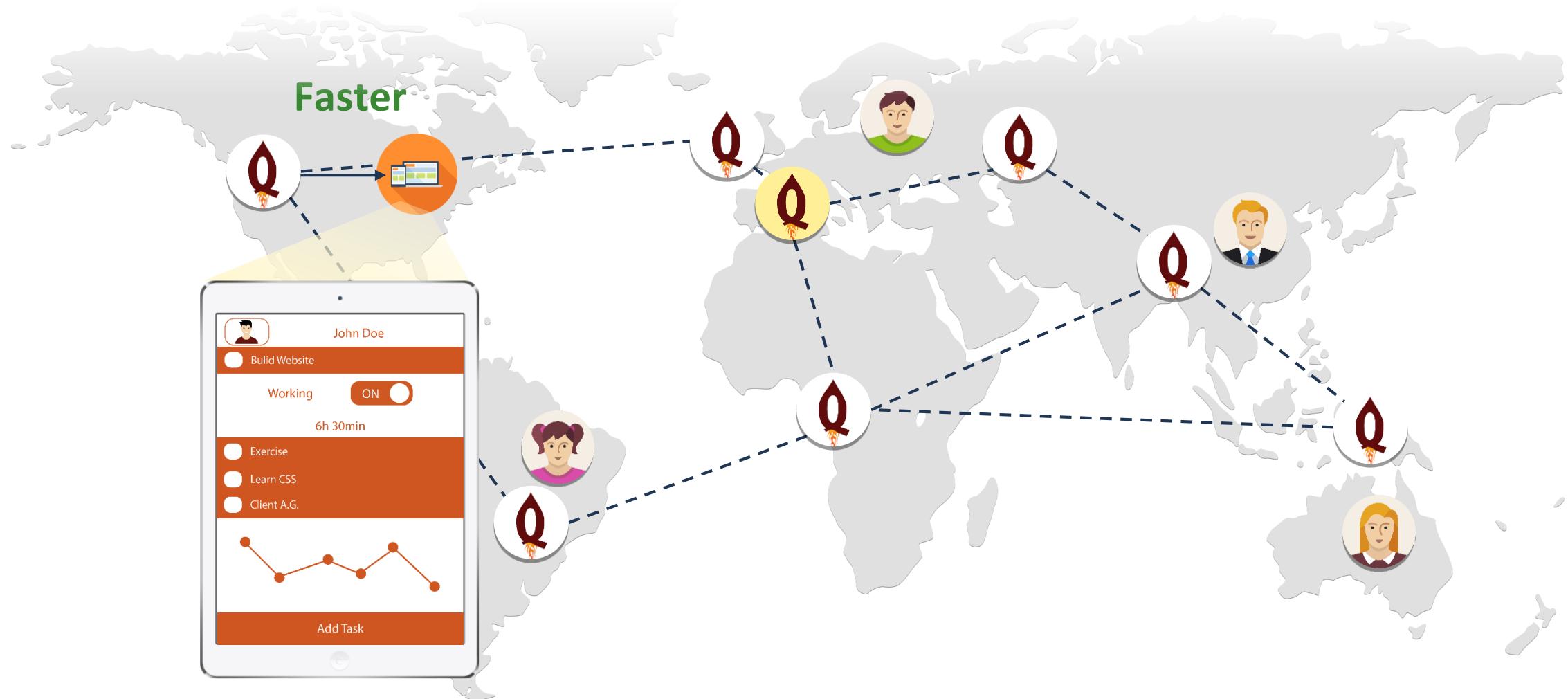
# This is what Speed Kit does.

Baqend brings data **closer to users**.



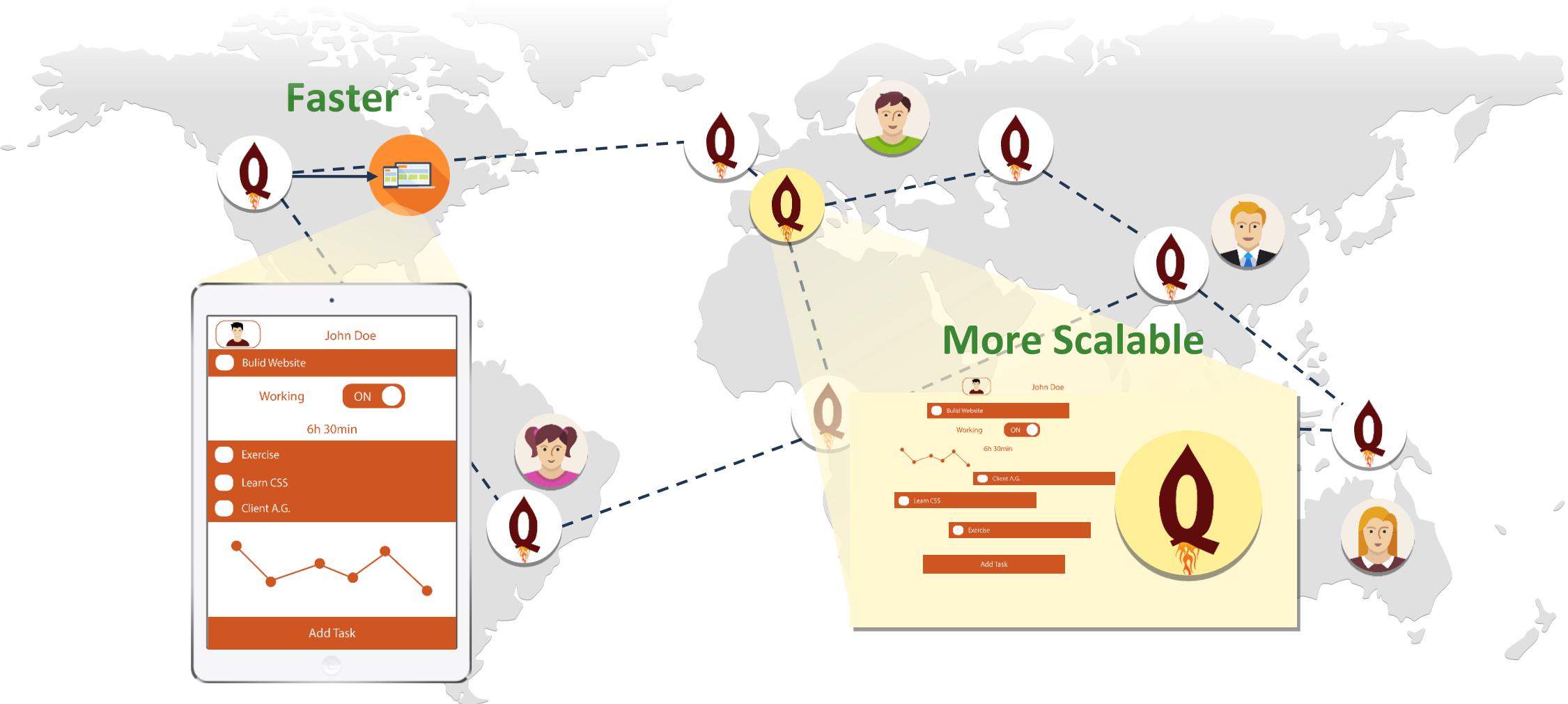
# Speed Kit delivers the site **faster**.

Data is automatically served from the nearest **cache**.

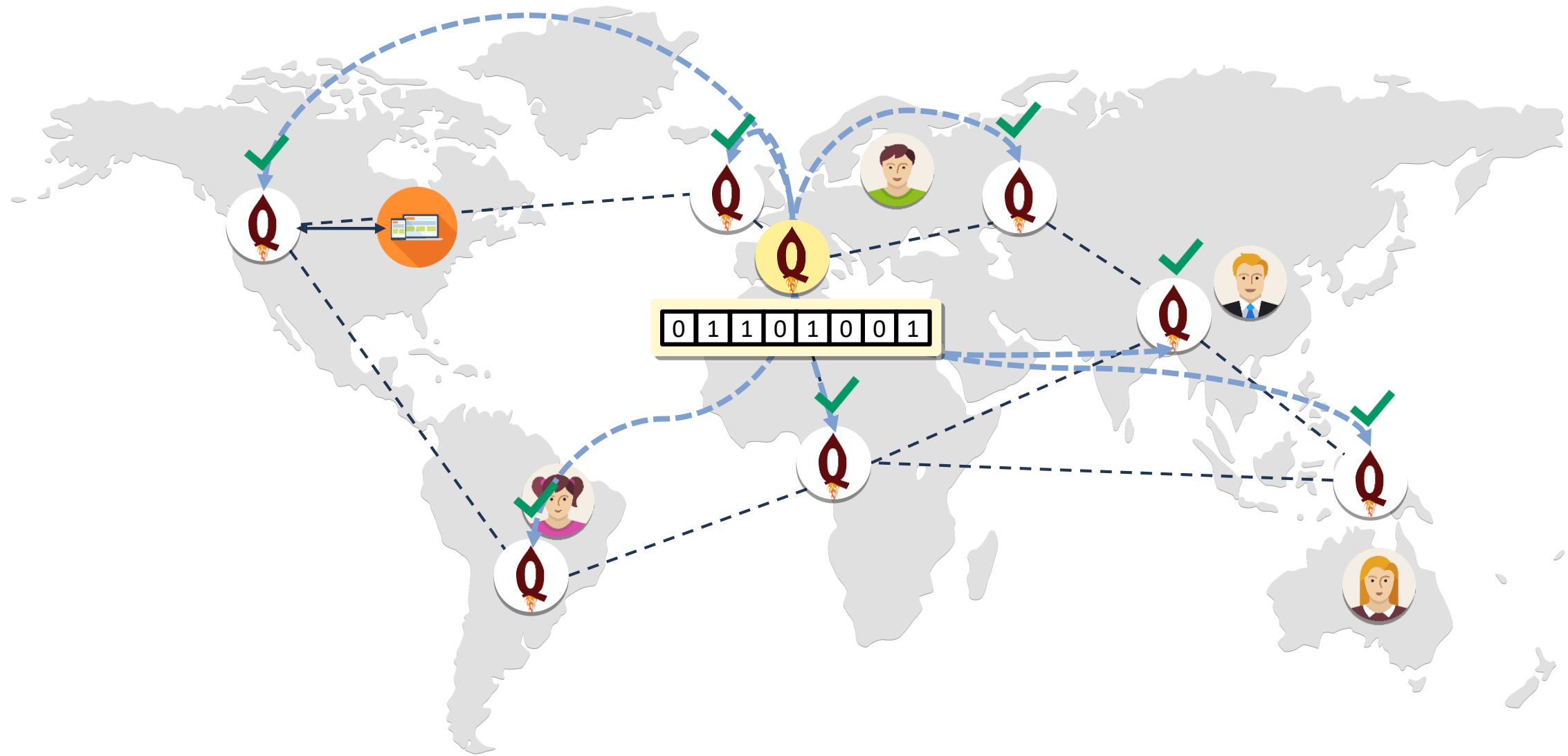


# Speed Kit helps scaling to more users.

The backend can now sustain heavier load.



# Data is kept **consistent**.



# Data is kept **consistent**.

## Novelty: caching dynamic data.

Backed by

## 30 man-years of research.

F. Gessert und F. Bücklers, „Performanz- und Reaktivitätssteigerungen von OODBMS vermittels der Web-Caching-Hierarchie. Bachelorarbeit, 2010.

M. Schaarschmidt, F. Gessert, und N. Ritter, „Towards Automated Polyglot Persistence“, in BTW 2015.

S. Friedrich, W. Wingerath, F. Gessert, und N. Ritter, „NoSQL OLTP Benchmarking: A Survey“, in 44. Jahrestagung der Gesellschaft für Informatik, 2014, Bd. 232, S. 693–704.

W. Wingerath, F. Gessert, S. Friedrich, N. Ritter „Real-time stream processing for Big Data“, *Big Data Analytics It - Information Technology*, 2016

F. Gessert, W. Wingerath, S. Friedrich, N. Ritter “NoSQL Database Systems: A Survey and Decision Guidance”, *Computer Science - Research and Development*, 2016

F. Gessert, F. Bücklers, und N. Ritter, „ORESTES: a Scalable Database-as-a-Service Architecture for the Cloud“, in CloudI 2014 – 4

F. Gessert und F. Bücklers, „ORESTES: ein System für horizontal skalierbaren Zugriff auf Cloud-Datenbanken“, in Informatiktage 2013, 2013.

F. Gessert und F. Bücklers, „Kohärentes Web-Caching von Datenbankobjekten im Cloud Computing. Masterarbeit 2012.

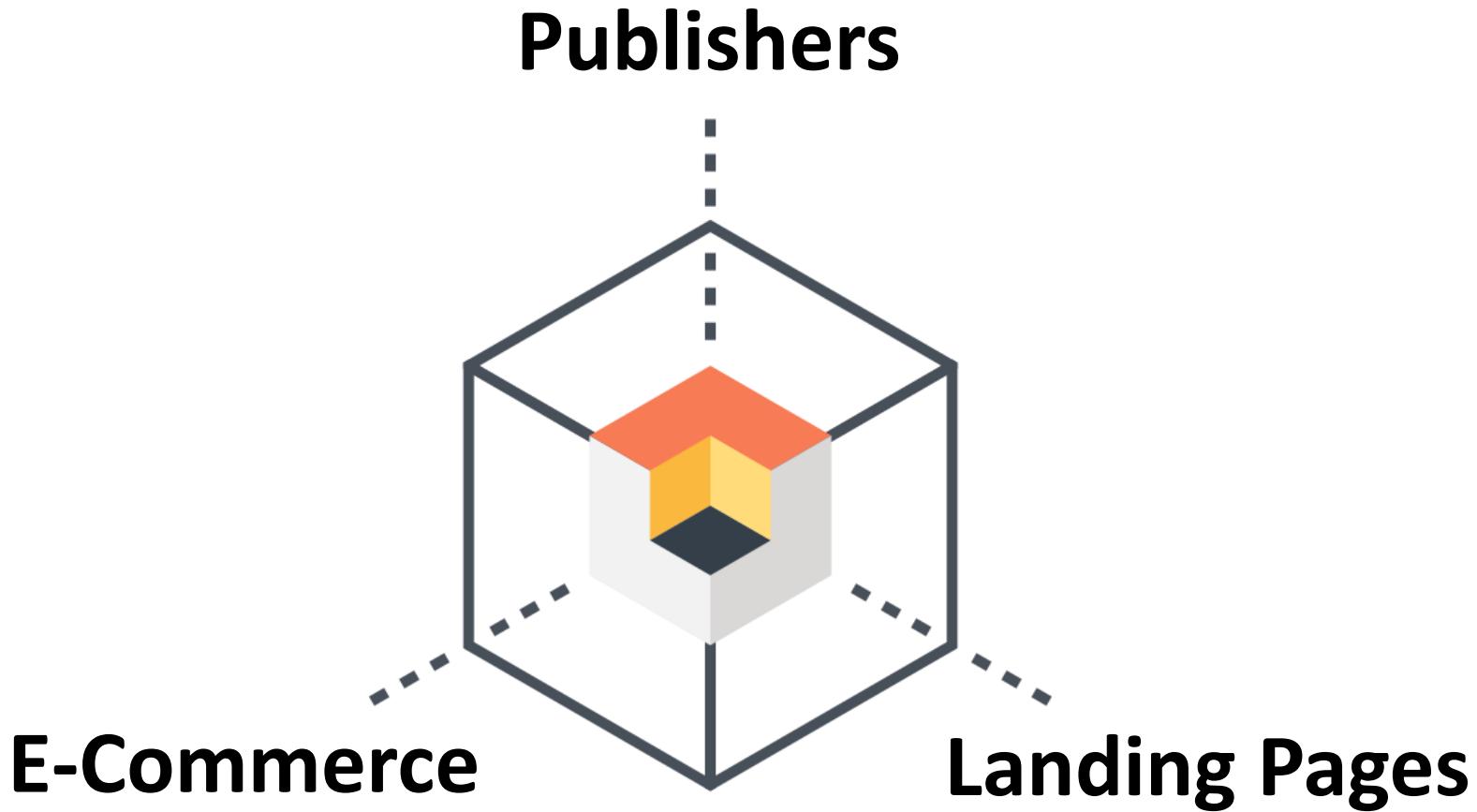
W. Wingerath, S. Friedrich, und F. Gessert, „Who Watches the Watchmen? On the Lack of Validation in NoSQL Benchmarking“, in BTW 2015.

F. Gessert, „Skalierbare NoSQL- und Cloud-Datenbanken in Forschung und Praxis“, BTW 2015

F. Gessert, N. Ritter „Scalable Data Management: NoSQL Data Stores in Research and Practice“, 32nd IEEE International Conference on Data Engineering, ICDE, 2016

F. Gessert, N. Ritter „Polyglot Persistence“, *Datenbank Spektrum*, 2016.

**Speed Kit is independent of the tech stack.**



# Works for Publishers.

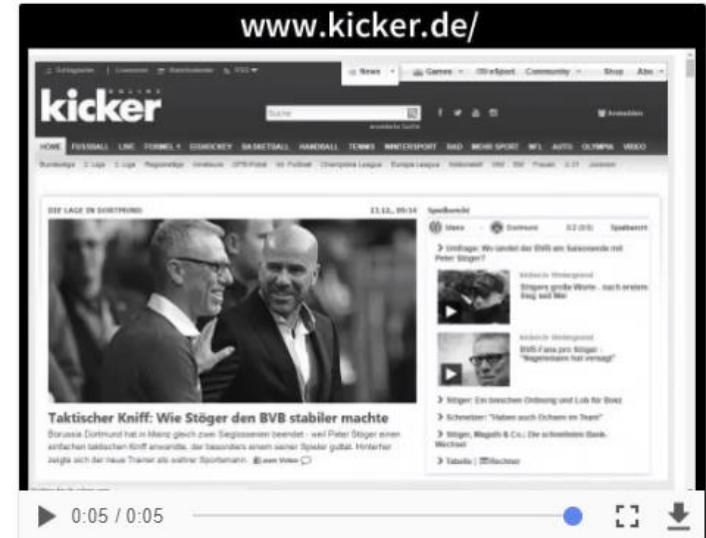
kicker.de

Your Website without Speed Kit



3754ms	1st Meaningful Paint	1941ms
3675ms	Speed Index	1877ms
814ms	Time To First Byte	28ms
7147ms	DOMContentLoaded	5637ms
17344ms	FullyLoaded	11761ms
3.8s	Last Visual Change	2.7s

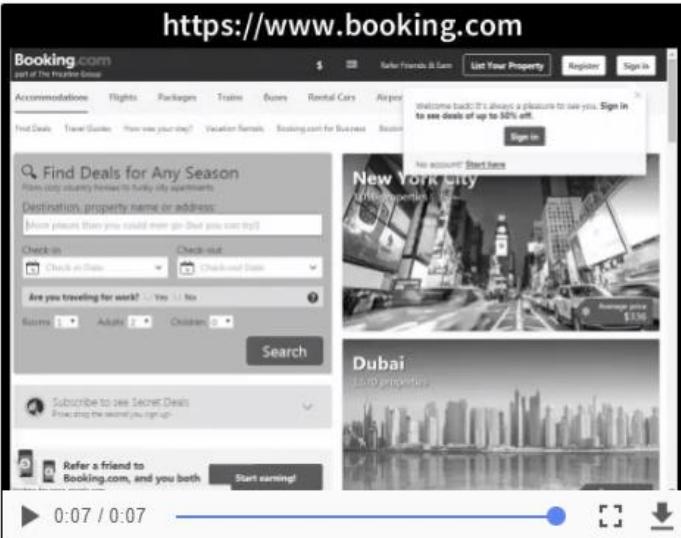
Your Website with Speed Kit



# Works for Portals.

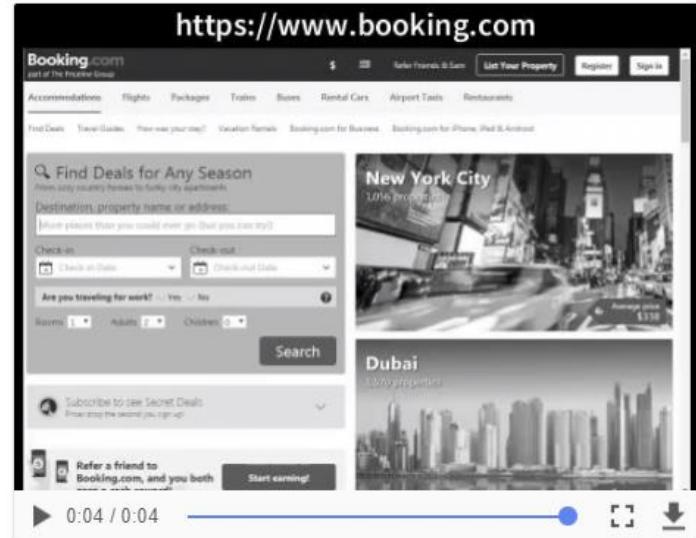
## booking.com

Your Website without Speed Kit ↗



2.05x Faster	1402ms	Speed Index	685ms
3.24x Faster	1280ms	1st Meaningful Paint	395ms
30.38x Faster	790ms	Time To First Byte	26ms
1.81x Faster	1620ms	DOMContentLoaded	894ms
1.15x Faster	4239ms	FullyLoaded	3698ms
3.34x Faster	4.3s	Last Visual Change	1.3s

Your Website with Speed Kit ↗



# Works for Landing Pages.

coca-cola.com

Your Website without Speed Kit ↗



3707ms	Speed Index	1303ms
3578ms	1st Meaningful Paint	562ms
77ms	Time To First Byte	29ms
3557ms	DOMContentLoaded	726ms
6314ms	FullyLoaded	4165ms
4.4s	Last Visual Change	2.2s

Your Website with Speed Kit ↗



# Works for E-Commerce.

alibaba.com

Your Website without Speed Kit ↗



2889ms	Speed Index	1341ms
3218ms	1st Meaningful Paint	2296ms
1544ms	Time To First Byte	17ms
2441ms	DOMContentLoaded	1274ms
5884ms	FullyLoaded	3642ms
3.5s	Last Visual Change	2.4s

Your Website with Speed Kit ↗



# Works for Brands.

braun.de

Your Website without Speed Kit ↗



	2721ms	Speed Index	527ms
5.16x Faster			
6187ms	1st Meaningful Paint	643ms	
9.62x Faster			
659ms	Time To First Byte	21ms	
31.38x Faster			
7317ms	DOMContentLoaded	786ms	
9.31x Faster			
7898ms	FullyLoaded	3151ms	
2.51x Faster			
7.4s	Last Visual Change	1.7s	
4.41x Faster			

Your Website with Speed Kit ↗

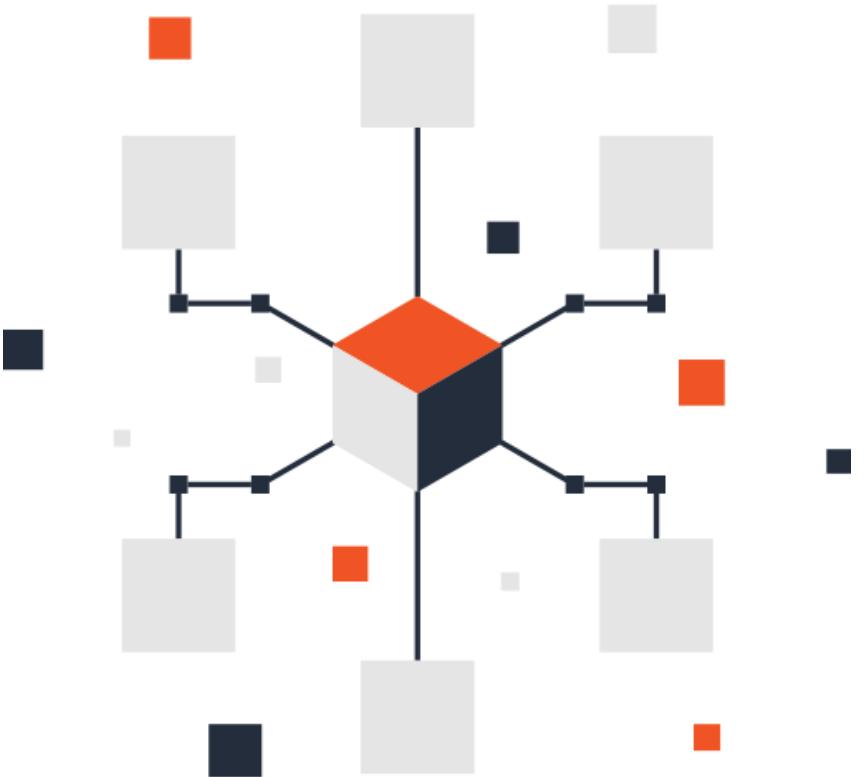


# Does it work for You?

www.example.com

Go

test.speed-kit.com



Which **business impact**  
do these improvements  
have?

# How much is being **2.2s faster** worth?

Publishers and Ad-driven Businesses

+16% Revenue

(PLT Original - PLT Speed Kit) / (19000 - 5000)

[DoubleClick study \(p. 3\)](#) "The Need for Mobile Speed" based on 4,500 real websites



E-Commerce

+22% Revenue

(PLT Original - PLT Speed Kit) \* (1 / 100)

[Amazon study \(p. 10\)](#) "Make Data Useful" using A/B tests on the Amazon shop





“

Every **1s** of page load improvement  
equals a **2% conversion** rate increase  
for Walmart.com

”

Cliff Crocker, Aaron Kulick, and Balaji Ram, “Real User Monitoring at Walmart.com: A Story in Three Parts,” 2012.



“

Staples.com shaves **1s** from load time,  
improves **conversion rate by 10%**.

”

Cliff Crocker and Steve Skroce, “How to Measure Revenue  
in Milliseconds,” 2014.

# AliExpress



AliExpress reduced load time by **36%** and recorded a 10.5% increase in orders and a **27% increase in conversion rates** for new customers.



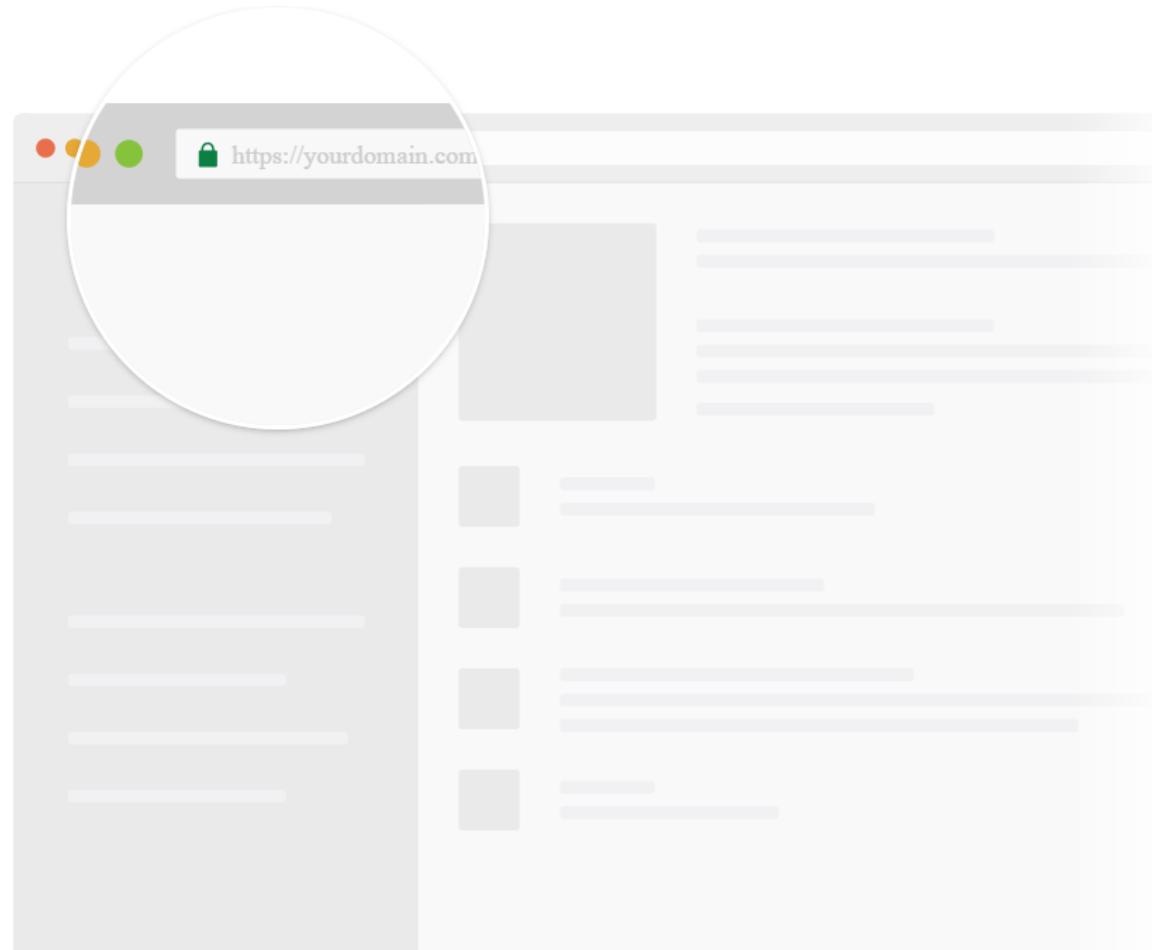
Alibaba presentation at Akamai Edge,  
“Future of Commerce,” 2016.



**Adding Speed Kit to your site.**

## 1. Configure Domain

Set which sites/URLS  
Baqend should accelerate  
(white- and blacklist,  
dynamic blocks).



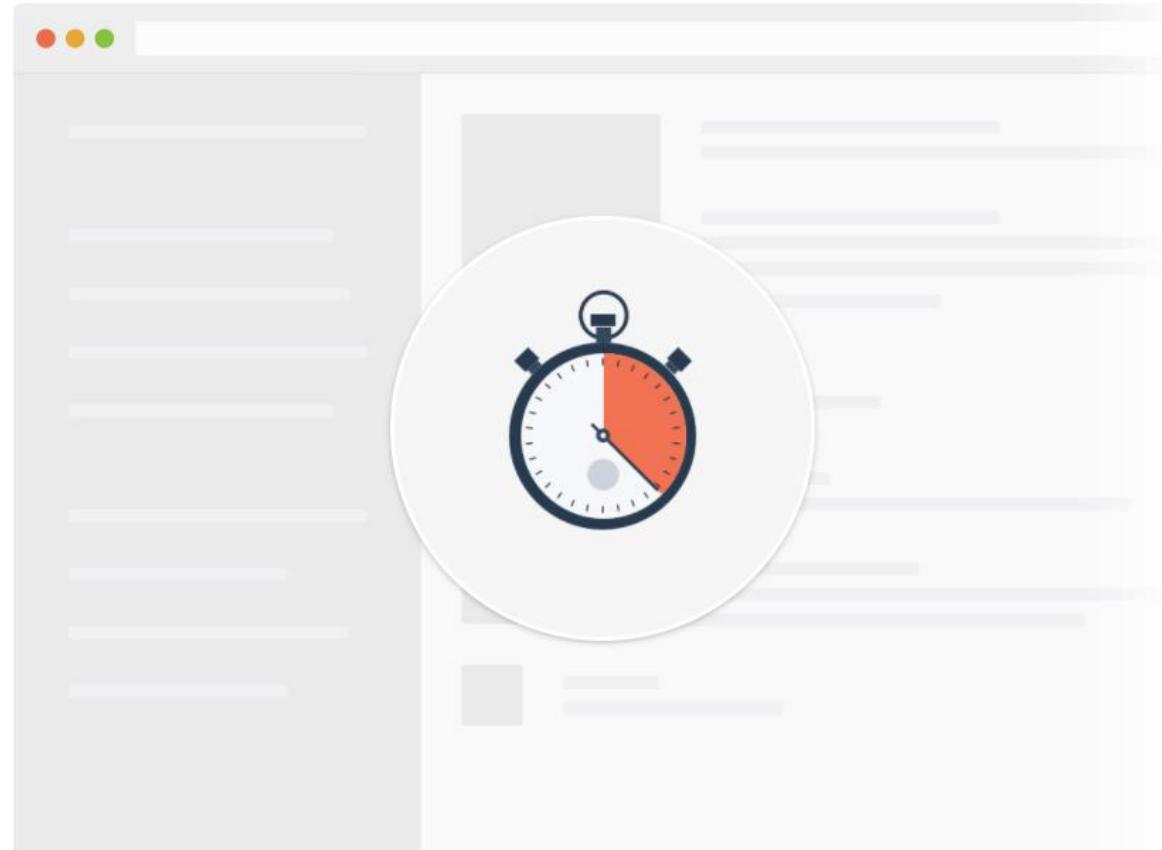
## 2. Include Code Snippet

Add Speed Kit to your website's HTML and upload the Service Worker.



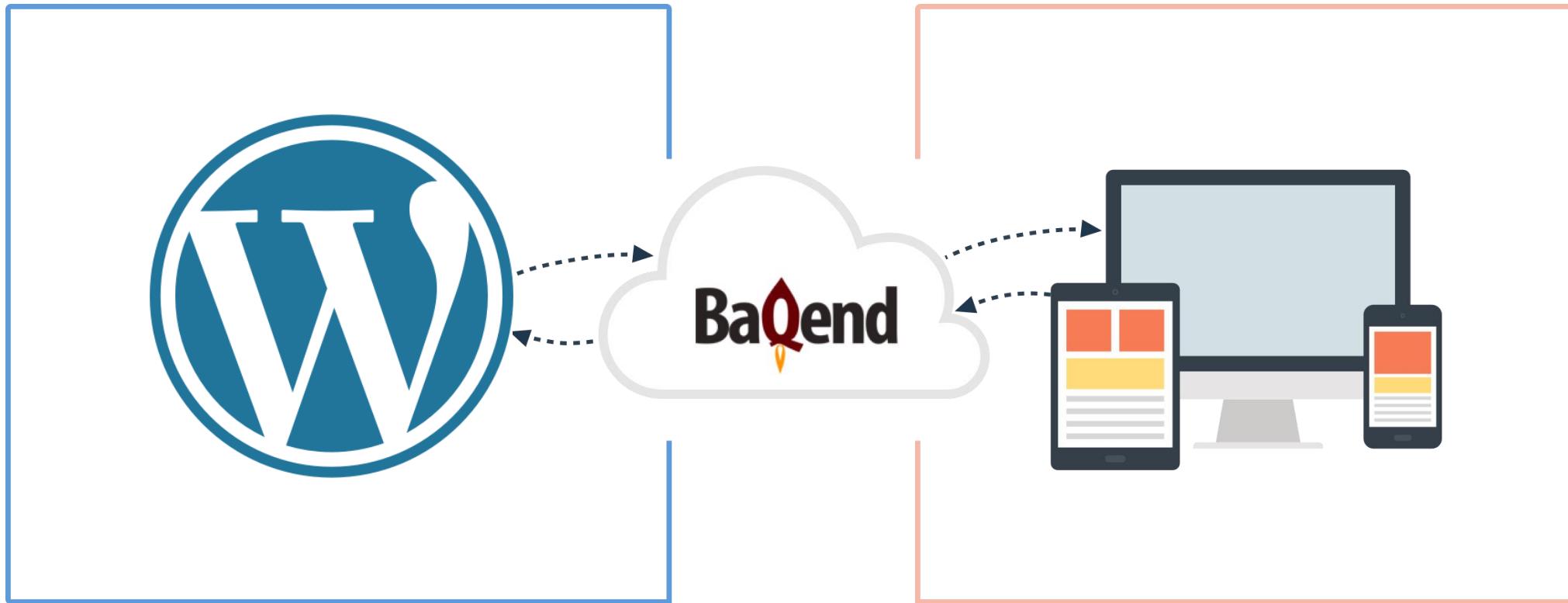
### 3. Speed Kit is Active

Speed Kit intercepts requests  
and serves them through  
Baqend's infrastructure.

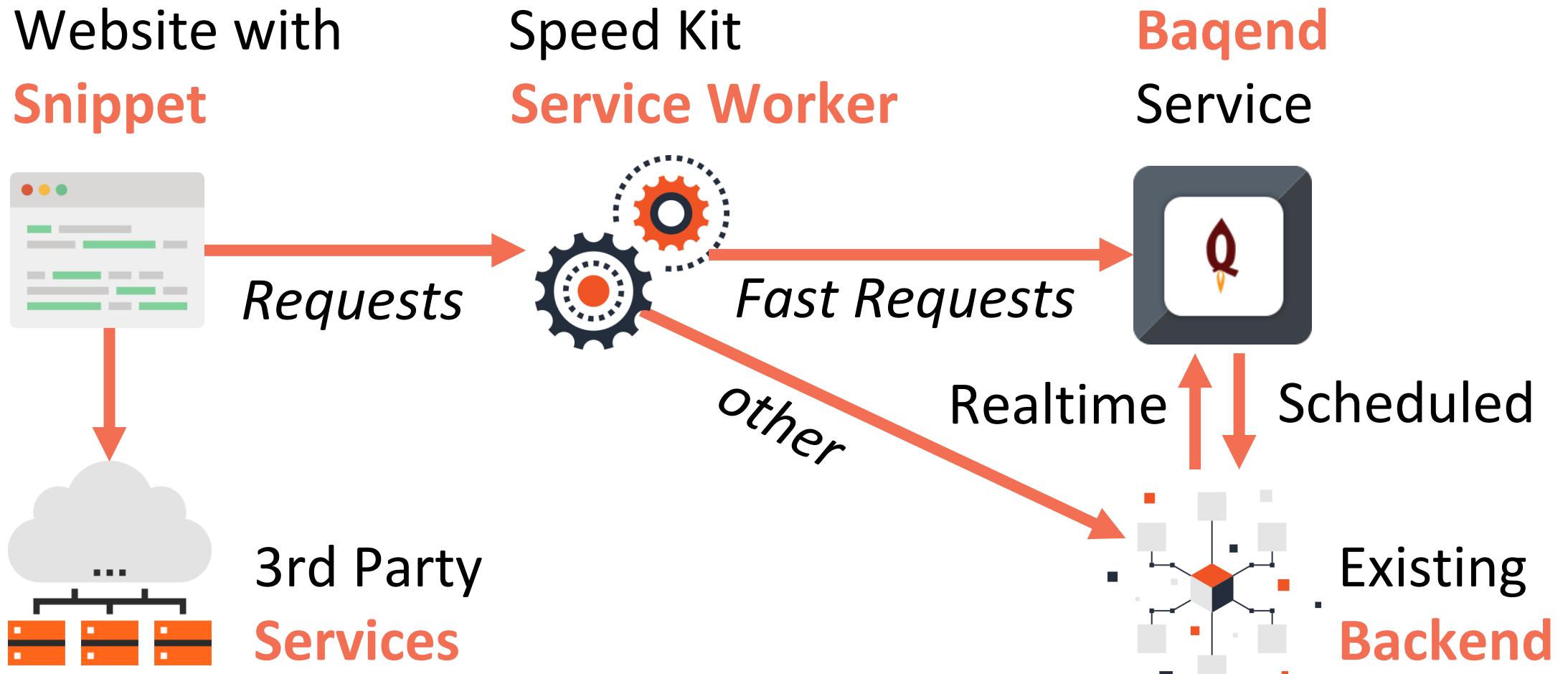


# WordPress too slow?

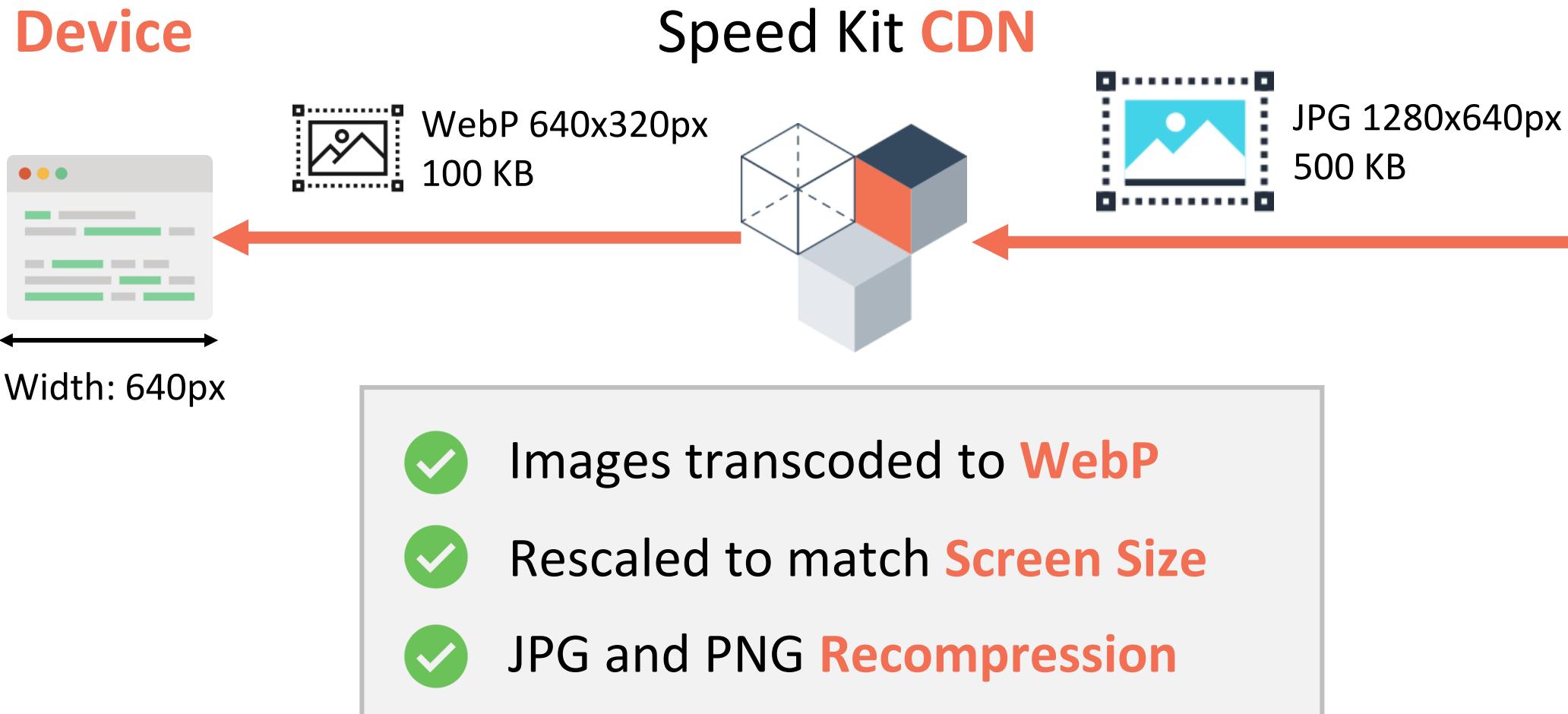
Install the official Speed Kit plugin.



# How it works under the hood



# Optimized & Cached Images



# How is Speed Kit different from **CDNs**?

Better  
**Performance**



Simpler  
**Integration**



**No** Vendor  
**Lock-in**



Caching dynamic data  
& in the browser

Just include the  
Speed Kit Snippet

Toggle Speed Kit on/off  
at any time

# It works for them.



**golem.de**  
IT-NEWS FÜR PROFIS



**fussballdaten**

**~2.5× Faster**

**~2.1× Faster**

# Speed Kit Integration In Detail

2 easy ways to **refresh data**

**Baqend Service**



Realtime ↑ Scheduled ↓



Existing **Backend**

**Scheduled:** Cron Job

**Realtime:** Manual, API Call

HTML    CSS    Javascript    Feed    Audio    Video    Image    Font

- URL Prefixes/Regexes
- Arbitrary Queries on url, eTag, lastModified, contentType, mediaType

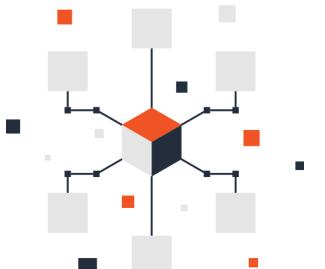
# Speed Kit Integration In Detail

2 easy ways to **refresh data**

**Baqend Service**



Realtime ↑ Scheduled ↓

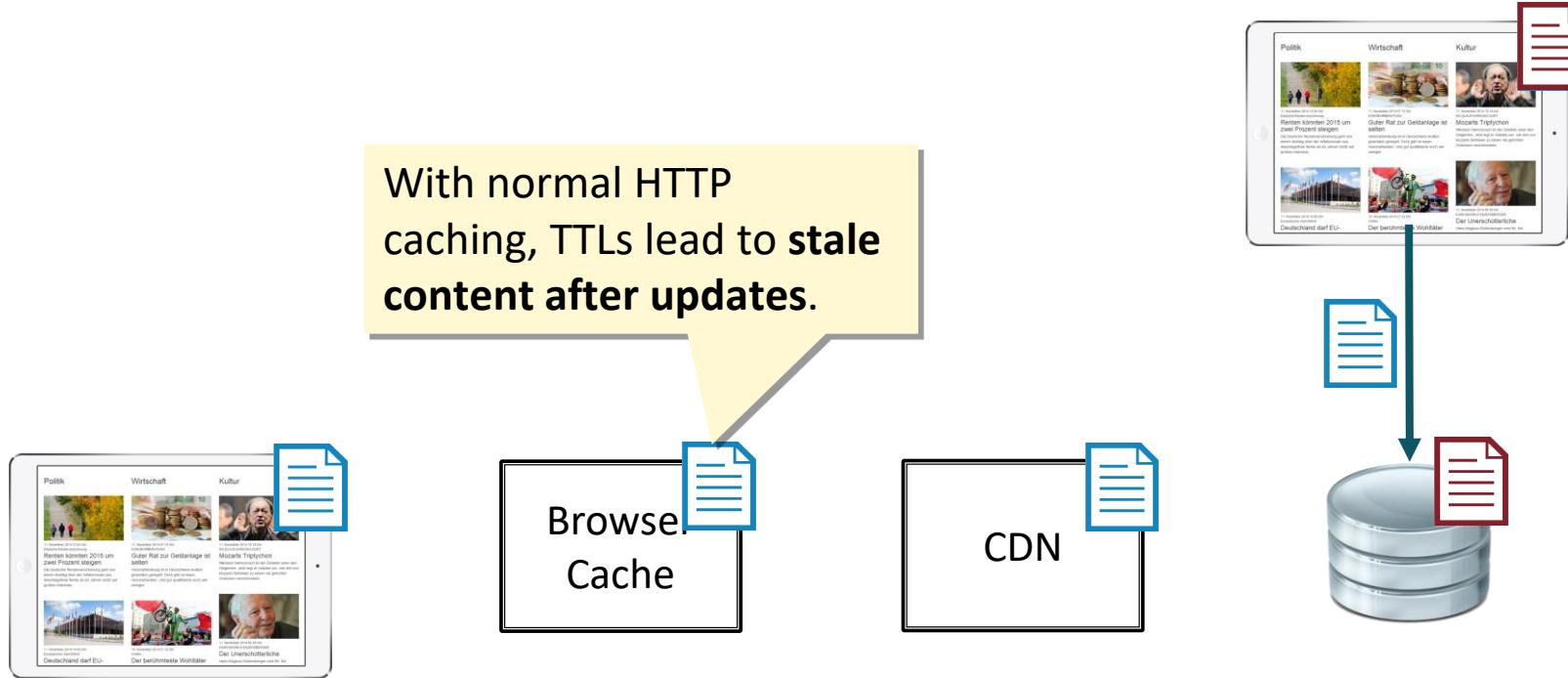


Suggested default is scheduled  
refreshing with:

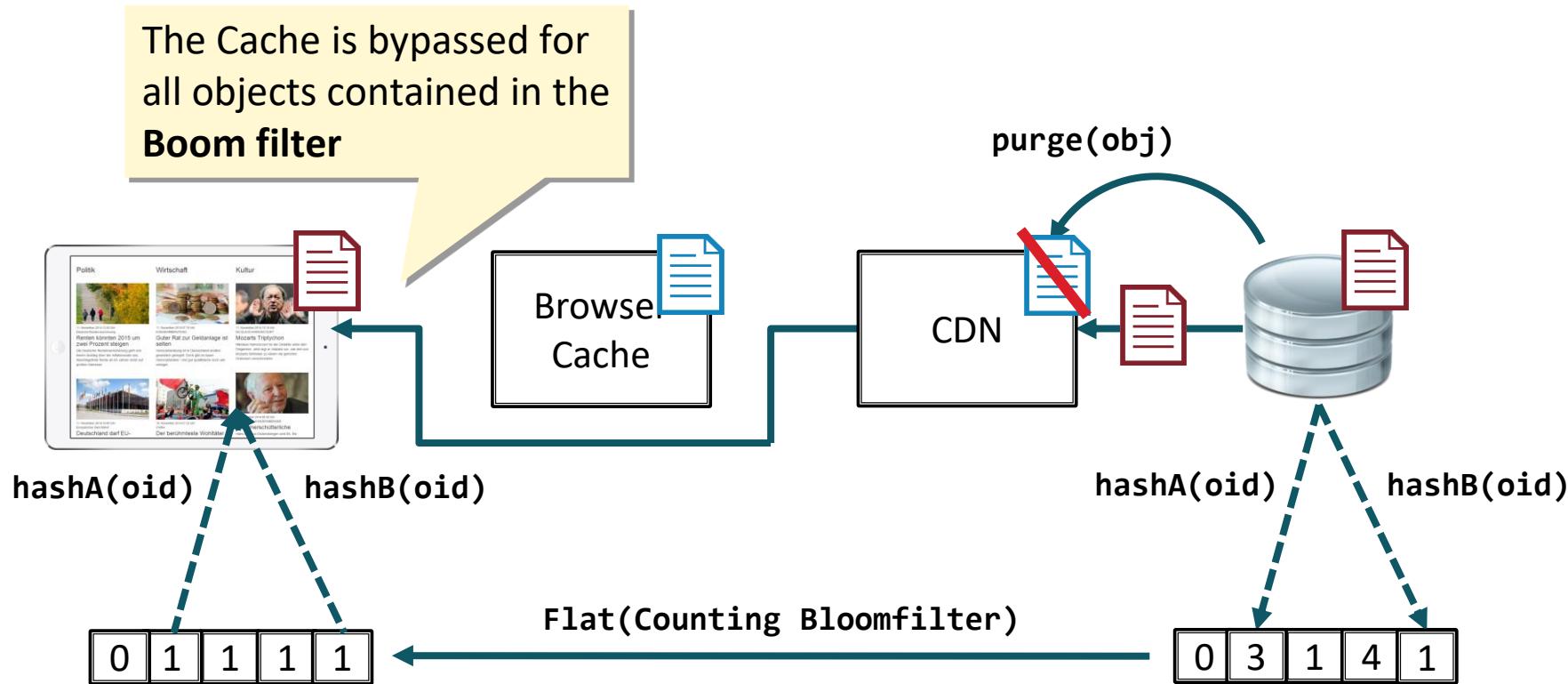
- **Landing Pages** every **1min**
- **Other HTML** every **10min**
- **Assets** every **1h**

Existing **Backend**

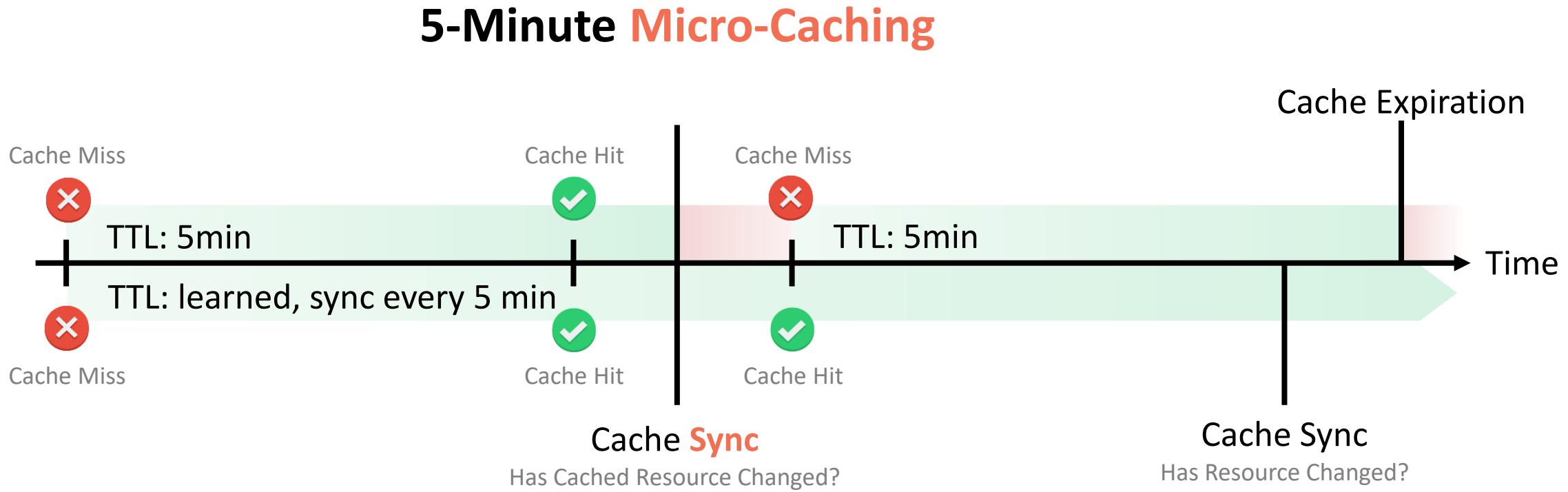
# How Baqend solves Cache Coherence



# How Baqend solves Cache Coherence

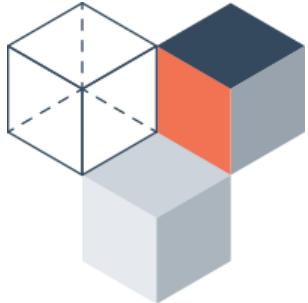


# How is this different from Micro-Caching?



**Speed Kit's Caching**

# Other Optimizations



## Caching the Uncacheable

Global cache invalidations in 150 ms



## Network Optimizations

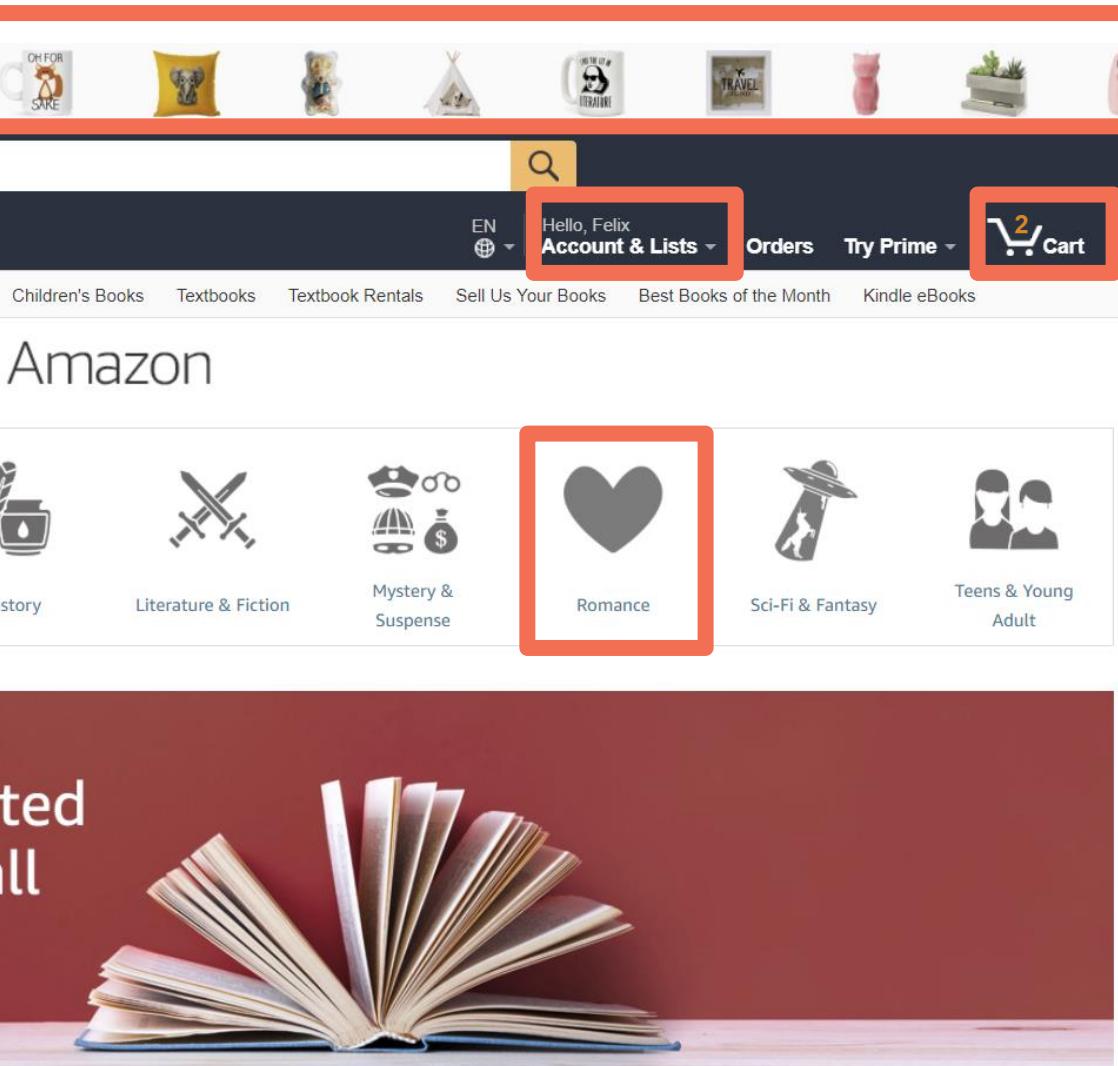
Optimized HTTP/2  
Tuned TLS  
TCP Multiplexing



## Speed up 3rd Party Dependencies

Cache External Resources

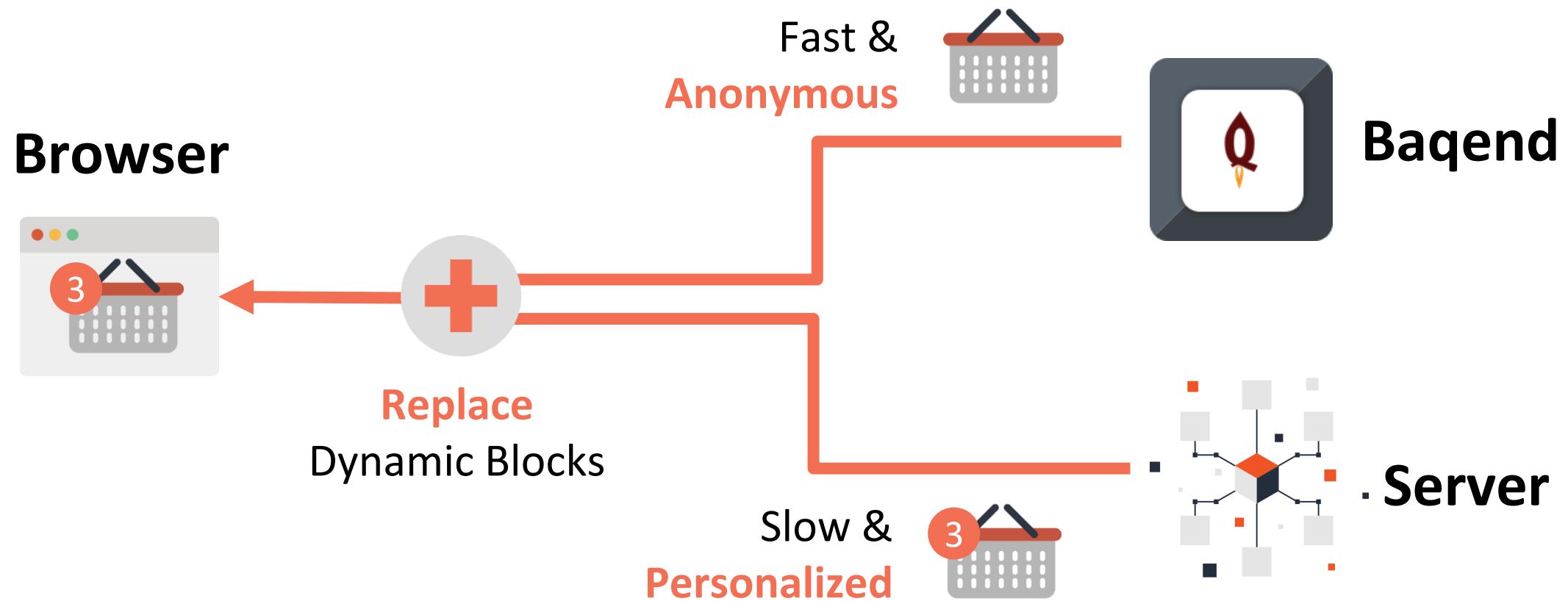
# Caching User- or Segment-specific Content



Speed Kit's **Dynamic Blocks** for shops, A/B tests, etc.:

1. Blocks **defined** by query selector.
2. Speed Kit loads a fast anonymous & the personalized view in **parallel**.
3. Dynamic blocks are **replaced**.

# Solution: Dynamic Blocks



# Transform sites into **Progressive Web Apps.**

**When Offline:**



→ Website becomes **offline-capable PWA.**

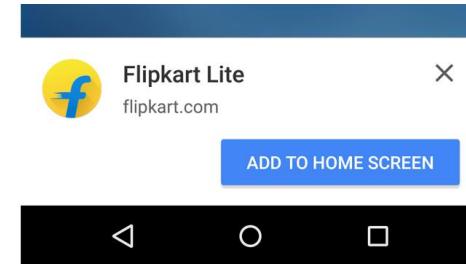
# Transform sites into Progressive Web Apps.



Fast **Loads**  
through Caching



**Offline** Mode with  
Auto-Sync



Add-to-**Homescreen**  
and **Push**

→ Website becomes **offline-capable PWA**.



# Dead Simple Pricing.

$\#Requests + \#Bytes$   
= *Pay-as-you-go*

# Requests

First 150K Requests	free
Next 10M Requests	0.34€ / 10K Requests
Next 140M Requests	0.25€ / 10K Requests
Next 360M Requests	0.20€ / 10K Requests
Over 500M Requests	Contact Sales

# Traffic

First 4 GB	free
Next 200 GB	0.34€ / 1 GB
Next 3000 GB	0.27€ / 1 GB
Next 7000 GB	0.25€ / 1 GB
Over 10 000 GB	Contact Sales

# Speed Kit in Summary.



**50-300% Faster  
Page Loads**



**Works with  
Current Site**



**Improves  
Scalability**



**Automatic  
Offline Mode**



Make page speed your  
competitive advantage.

[www.baqend.com](http://www.baqend.com)