twim

**YOUR PARTNER FOR
DIGITAL TRANSFORMATION**

WELCOME TO THE JAMSTACK

An introduction to a revolutionary

new web development architecture

The problem with the legacy web

**COMPLEX APPS
AND COMPLEX SERVERS**

A traditional website is actually a program that has to run on a web server at all times.

Running sites this way needlessly slows things down, provides way too many opportunities for attack, and is expensive to scale.

Web server
(like Apache)

Application
(like Wordpress)

Database
(like MySQL)
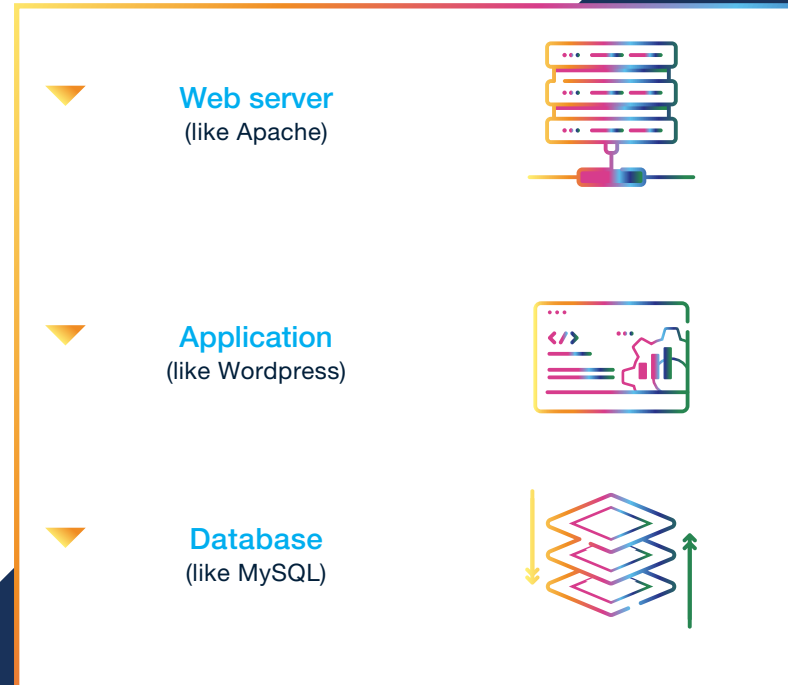
What the Jamstack delivers

**Faster
websites**
Optimize your site
with prerendering
and global delivery.

**More secure
infrastructure**
By design, Jamstack
architectures offer
fewer points of attack.

**High scale without
high complexity**
Optimize your site
with prerendering
and global delivery.

**Improved developer
experience**
Launch faster
with more productive dev
& deployment cycles

How it works

▶ **DECOUPLING THE FRONTEND FROM THE BACKEND**

Unlike the large legacy apps, Jamstack projects neatly separate the frontend pages and UI from the backend apps and databases. Freed from backend servers, the frontend can then be deployed globally, directly to a CDN.

▶ **PREBUILDING ALL PAGES FOR SPEED**

Before deployment, the entire front end is prebuilt into highly optimized static pages and assets. This happens in a build process (before being deployed / hosted).

▶ **DYNAMIC CONTENT VIA APIS**

The global front end uses Javascript and APIs to talk to backend services, allowing pages to be enhanced and personalized.

HTML   JS   PNG

High-performance static frontend

Builds

APIs

Build system (CI/CD)
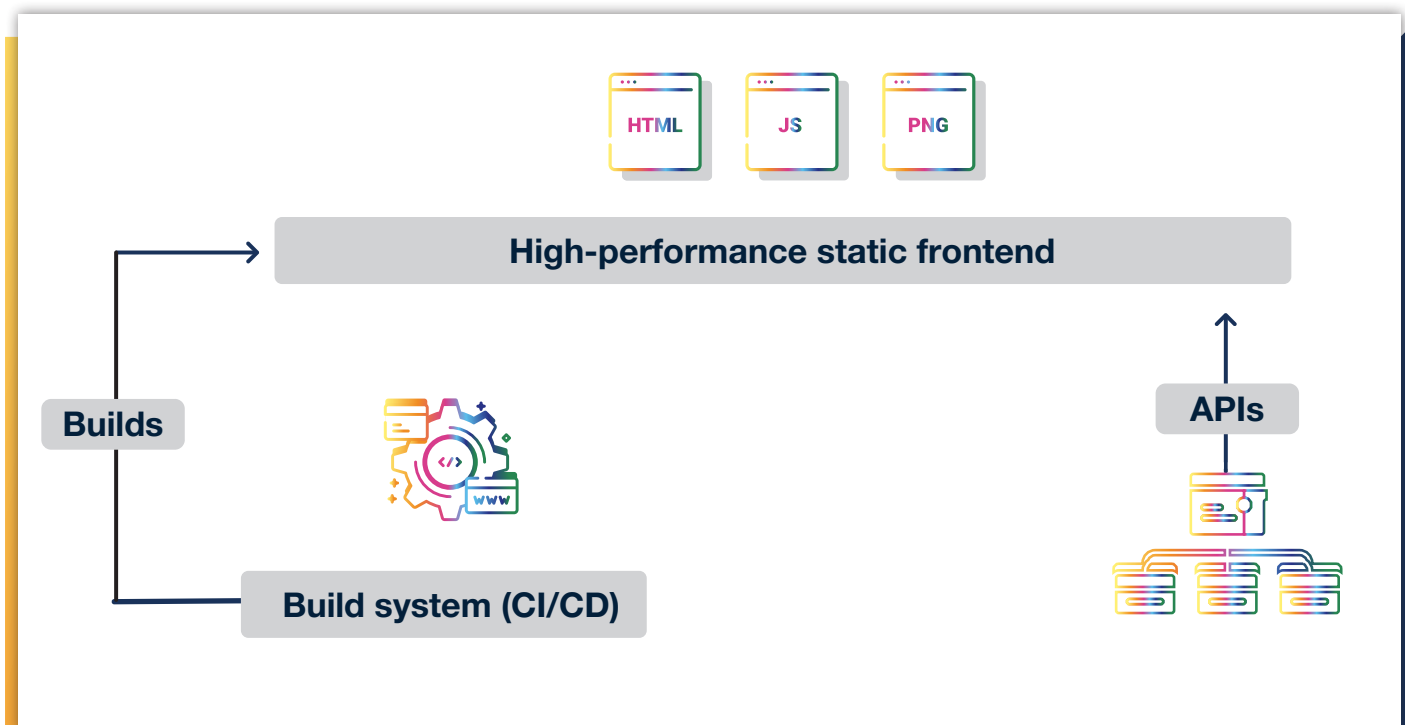
How it advances the web

RUNNING SITES ON WEB SERVERS → replaced by → **DEPLOYING SITES GLOBALLY**

A Jamstack deployment doesn't run on a traditional setup of origin servers. Instead, automated deployments are used to push sites directly to the Edge / CDN.

WEBPAGES RENDERED AT RUNTIME → replaced by → **WEBPAGES PRERENDERED FOR SPEED**

Rather than building content at runtime for each request, content is prebuilt and optimized during a build setup using a site generator and other build tools.

MONOLITHIC APPLICATIONS → replaced by → **APIS & MICROSERVICES**

Decoupling the front end from the backend allows for more modular development where we can take advantage of the huge ecosystem of third party tools to provide website functionality.

BYOT: Bring your own technologies

Progressive concept. Proven components.

The Jamstack doesn't change the technologies you use—Jamstack sites can be built in Javascript, PHP, Ruby, Python or virtually any language. It's not a collection of specific software, it's a set of best practices that combines:

MODERN BUILD TOOLS

With a build run every time a deploy happens, The Jamstack brings full support for linting code, transpiling javascript, compiling CSS, and optimizing html and assets.

GIT WORKFLOWS

The Jamstack ties deployments closely to a Git-based workflows. Git brings the rigor and safety of version control to web projects allowing support for large projects with numerous contributors.

CDN TECHNOLOGY

First deployed by large enterprises, the Jamstack democratizes the use of CDN technology to deliver web content from end points as close to users as possible.

Summary

Jamstack is extremely powerful as a model to replace the traditional web development approach and allows a small agile team to move at pace. It removes the need to worry about infrastructure and servers. It also encourages integration with other software-as-a-service vendors using interoperable APIs and moves away from backend systems requiring additional specialist teams.

Releasing, supporting and scaling Jamstack is trivial. This means high agility, fast speed-to-market, and better uptimes. Enterprise is slow moving and those that can adapt will get a competitive advantage.

Deepak Kumar, CEO twim GmbH

CLICK HERE FOR YOUR FREE PRESENTATION

twim

**YOUR PARTNER FOR
DIGITAL TRANSFORMATION**

info@twim.ch

Twim is an official Storyblok and Netlify partner.

B storyblok
PARTNER

netlify
JAMstack partner