LAMP vs. MEAN: Which stack is right for you?

May 02, 2019 | 6 min read

Christoph Heike

> This is a guest post by Christoph Heike – "I've been developing web applications for over 10 years and currently run a web-development agency in Bonn, Germany. I'm also involved in an Amazon-based tech startup. My goal is to always deliver clean, sustainable and high performant software solutions. Connect with me on LinkedIn."

A web stack is a collection of software or technologies that are used to build a web application. Choices are plenty, but picking one can be hard.

When chatting with co-workers, developers or customers, suggestions for what technologies and stacks to use couldn't be more different. When I started off as a web developer, I went the usual way at that time: learning HTML & CSS, exploring some PHP – and of course MySQL. That was, if you were not using Java or ASP.NET, the technology stack of that time. Whether you wanted to host a blog, a bulletin board or become an image hoster – you would more than often need these things: Linux, Apache, MySQL and PHP (LAMP).

Here is a detailed overview of LAMP and the relatively new, MEAN stack, which are currently the most popular open source web stacks and a brief overview of other stacks. Whichever stack you choose, Bitbucket works with them all.

LAMP

LAMP delivers a strong platform for developing and hosting large, performant web applications. With the biggest and oldest community, countless libraries and tools, you get great support and will find developers quite easily.

Its integral components are:

Linux (OS)

Apache (Webserver)

MySQL (Data persistence)

PHP (Programming language)

There are also some derivatives of this stack:

- LAMP (with Perl or Python instead of PHP)
- LAMP (with MongoDB instead of MySQL)
- WAMP (Windows as OS)
- MAMP (Mac OS X as OS)
- XAMPP (Any OS + Perl or PHP + FTP Server)
- LAPP (PostgreSQL as database)

Pros:

LAMP is kind of the dinosaur of web development, used by hundreds of thousands of companies and therefore maintained and supported very well. With endless modules, libraries and add-ons available you can adapt it to your company's needs.

Being Linux based, you will find help for any topic in the large open source community. MySQL is a very reliable and scalable solution. PHP is in version 7 and is also supported by a mature and big community. PHP is also very fast and integrates well with the rest of the stack.

You can control the server and decide which versions and software you install, so you don't have to depend on the client's browser. Best for if you have lots of server-side tasks.

Cons:

Because it's easy to learn, there are a lot of developers out there who are not following best practices and building garbage apps. Starting with PHP is easy, but mastering it is hard. This is also true for security in these PHP apps. Some would also describe it as a script language instead of a real programming language because it's not strongly typed and not pre-compiled. I'd recommend diving in deeper into pros and cons of PHP, Python or Perl.

As for MySQL, other options are becoming more mature. NoSQL databases like MongoDB are popular among enterprises today due to it's scalability. Plus, pure JavaScript Stacks like MEAN gain more traction every year and new developers might not be interested in learning all of the LAMP's skills.

MEAN

Compared to LAMP, the MEAN stack is fairly new. One of its biggest differences is that MEAN is not dependent on a specific operating system – Node.js takes care of server-side execution. The MEAN Stack is especially recommended for JavaScript enthusiasts – as it uses JavaScript at all levels. This also makes it preferred by new developers.

MongoDB is a popular and flexible document based, NoSQL database, compared to MySQL's relational database system. Angular helps build progressive and modern web apps.

Its components are:

MongoDB (Data persistence)

Express.js (server-side application framework)

Angular.js (client-side application framework)

Node.js (server-side environment)

This stack has some derivatives too:

- MERN (React instead of Angular)
- MEEN (Ember.js instead of Angular)

Pros:

Using JavaScript as the primary programming language is a huge advantage. Everything can be set up quickly and done in JS, which makes it much easier to find developers, and LAMP developers typically know JavaScript as well. MongoDB is very popular for its easy schemaless data persistence and is faster than MySQL if you have a lot of read requests. The fact that Angular is maintained by Google is also a big plus. It receives new releases and functions on a constant basis. Another huge advantage is the ability to easily build mobile or desktop apps, for example with Ionic. Code and components can easily be reused or added.

Cons:

Like all new technologies, MEAN's glamour is creating some hype. Developers fall for this hype and build their apps in JavaScript, just because it's trendy. Many of these libraries and frameworks are quite new, and new versions get released quickly, so maintaining your app can become quite a hassle. Since many technologies disappear after a few years, sustainability can become an issue. It's also harder to maintain a clean code base and follow best practices as your app grows. Further, you have to rely on the client and the client's available technologies e.g. if you are targeting IE users, embedded systems or low end PCs, there may be usability issues.

A few other stacks to consider:

WISA

Windows Server / IIS / Microsoft SQL Server / ASP.net

Not open source, but all components are from Microsoft, so it should work seamlessly.

LAMP (With MongoDB)

Linux, Apache, MongoDB, PHP

NoSQL Databases like MongoDB can also be used in a classic LAMP environment.

Ruby Stack

Ruby/Ruby on Rails/RVM (Ruby Virtual Machine) / SQLite

This stack is losing popularity. Ruby on Rails was an often used framework once, and thus the whole stack.

Java+Spring

Preferred by large enterprises and shied by indie developers for its complexity, Spring offers an entire full-stack framework written in Java.

Django Stack

Python / Django / Apache / MySQL

The Django framework is loved by Python developers, delivers performance and is often referred to as an easy to learn stack.

Which stack is used more frequently?

It's hard to compare the popularity of stacks, but you can use Google Trends to compare programming languages and get a feel for what people are searching for. As the chart below shows, JavaScript is searched for more than PHP right now.

I'd recommend checking development trends using Google's trend tool from time to time.

I'd also suggest diving deeper on databases (SQL vs. NoSQL) to gain a basic understanding of the two concepts and make a choice.

So, how do you pick a stack?

Picking a stack depends on many factors. If you are a developer or project owner, here are a few questions to ask yourself.

- What kind of web-application am I planning to create?
- What is its expected lifetime?
- What technologies are available at my customer's/client's /cat's/... infrastructure?
- How easy is it to find developers to maintain the application?

Let me give you an example. Let's say you have a website for listing used cars. It was developed using a LAMP stack a while ago. But your website lacks a back-end for used car dealers, where they can manage their listings on your website. Depending on your company size, time and budget, you have to ask yourself all the above questions. If you have a small team, it might make sense to extend your existing application in the LAMP environment. Since your developers know the ecosystem and it'll be much faster. If you have time and resources to spare, you could take another approach and extend your existing LAMP application with an API. Later, your team could focus on developing a small, standalone (M)EAN application, that can easily be maintained, improved with new features and released using a much faster cycle.

Another example: You want to build a newsletter platform, where people can sign up, upload mailing lists, compose mailings and so on. You could of course use MEAN, but you have large scale and high traffic potential. It may make more sense to use a LAMP stack as your foundation, since Linux, MySQL and Apache provide a stable, scalable environment with lots of community support for any thinkable problem. You will also have lots of server-side tasks and cronjobs and will encounter mailing topics like SMTP and so on. I would recommend a Linux environment customized to your needs in this case.

Here is a summary of things to know/consider.

MEAN	LAMP
– Single code base (JavaScript) – Popular for modern web apps and hybrid apps – Supported by large companies like Google – Better for apps where a lot of the logic happens on the client's side – Harder to maintain long term due to the rapidly evolving javascript ecosystem – Best for progressive web apps	– Better for large applications – More mature, huge community – Well-established application frameworks like Symfony, Zend, Laravel – Easier to follow standards and easier to keep code clean

If you are new to programming and web development, ask yourself:

- What is the easiest to learn for you and your team?
- What technologies are trending and which will win in the long run?
- If open source, could you imagine contributing to the project?
- Which technologies will serve you personally in the long term?

A great resource for finding answers on JavaScript technologies is the StateOfJS project – https://stateofjs.com/ – it's a project that conducts a survey every year asking thousands of developers on their opinions on current technologies and salary.

Christoph Heike

From error to resolution with Sentry and the Atlassian stack

This post was written by Rahul Chhabria from Sentry, our

Setting up a basic server in Hapi.js

This post is for those who understand the basics of NodeJS, Javascript, and MVC. It's also

Find & fix errors faster with Rollbar + Bitbucket

[This guest post is written by Jesse Gibbs, Head of Product at Rollbar. Jesse is a developer turned product manager who...

marketplace partner. Find and fix errors in your code faster with...

May 06, 2020 | 6 min read

Rahul Chhabria

more suited for professionals who...

June 15, 2020 | 6 min read

Kanika Sud

January 06, 2017 | 6 min read

Raj Sarkar

Bitbucket

Technical Support
Documentation
Plans & pricing
What is Version Control?
Bitbucket writing program

Resources

Technical Support
Documentation
Plans & pricing
What is Version Control?
Bitbucket writing program
