

From Full Stack Engineer to CTO, Part 2

Author: Andrew Kenney

Claps: 62

Date: May 3, 2018

Klaviyo likes having T-shaped engineers with broad technical skills and experience and subject matter expertise in certain dimensions. Although we don't have job openings titled "Full Stack Engineer" we do recruit for the skillset and often see candidates apply with backgrounds or aspirations of being Full Stack Engineers. These candidates may have come from an accelerator program, been self taught, or have a formal computer science degree or online curriculum under their belt. Many have several years of experience at a startup building web products or apps. The best candidates often have an excellent portfolio of side projects they have built - be they mobile apps connecting to API backends, hackathon projects integrating exciting new technologies, or web apps leveraging the latest JS and frontend techniques to build delightful interfaces.

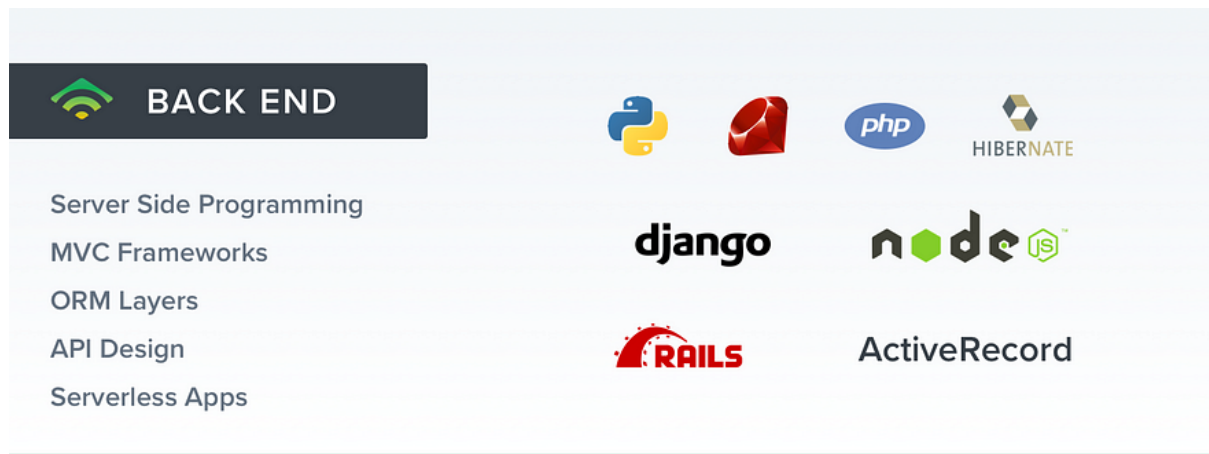
When we are evaluating these engineering candidates we like to assess their aptitude for full stack engineering and determine both where their existing expertise lies, as well as which technical skills they're passionate about learning or mastering. This is more than just a determination on whether as a full stack engineer you prefer the frontend or backend. We like to go deeper than that. Full stack is not as simple as making a basic CRUD app and the depth matters both for us growing cross-functional high-performing teams as well as reliably shipping exceptional products. In order to better understand an individual's strengths we break down full stack engineering into several logical stacks of technical expertise and separation of concerns.



The first stack is the **frontend stack**. In this almost everyone is agreed to have frontend engineering include HTML, CSS and Javascript and for a focus on UX, Design and product experience. The specific flavors of JavaScript may vary as well as candidates' aptitude for writing raw modern JavaScript or using a framework. Key engineering skills such as understanding build toolchains, responsive design and browser testing best practices are critical. Overall, however, frontend engineering is fairly straightforward even if the landscape is evolving

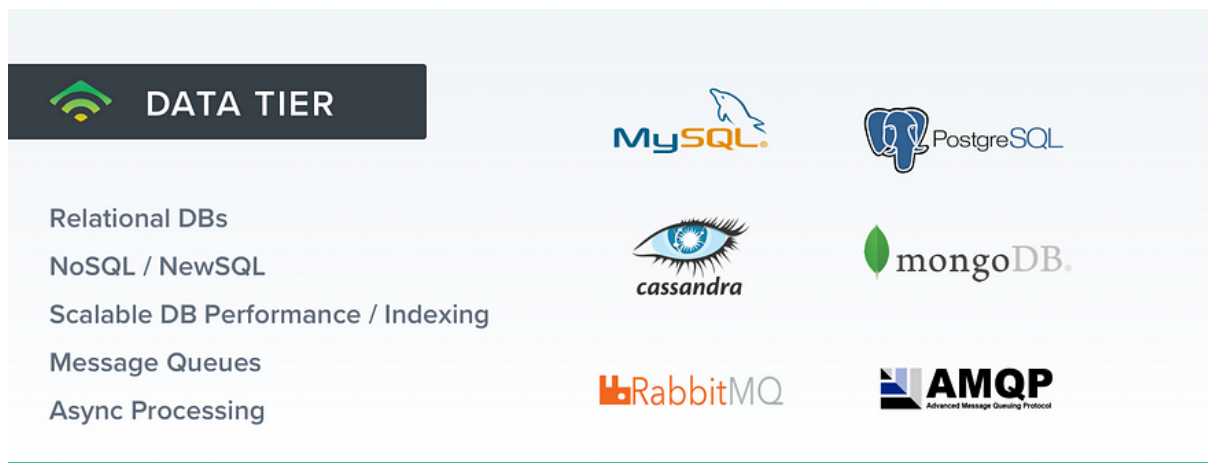
rapidly. Klaviyo itself utilizes React and Redux and is passionate about consistent UX and building reusable components as evidenced by us investing in a component library and style guide.

In frontend-focused candidates we look for a proven ability to learn and adopt best **practices and write clean code over dogmatic adherence to any particular frontend framework**. The best frontend engineers have an eye for detail and endlessly advocate for delightful and intuitive user interfaces and consistency across applications. They also are pragmatic and not quick to chase the latest shiny framework.



Going down one layer is the **backend stack**. When most people think “full stack” they tend to stop here, and as such what **backend engineer** conveys is often muddled. We prefer to think of backend technologies as the programming languages and web frameworks which make up modern SaaS architectures. This includes dynamic languages such as Python, PHP, and Ruby or even compiled languages like Scala, Go and others. It also includes popular frameworks such as Django or Rails meant to accelerate development efforts. We also include various skill sets in the mix such as utilizing an ORM for database querying or building and integrating with APIs. The backend stack usually contains the relevant business logic and as such is the brains of the app.

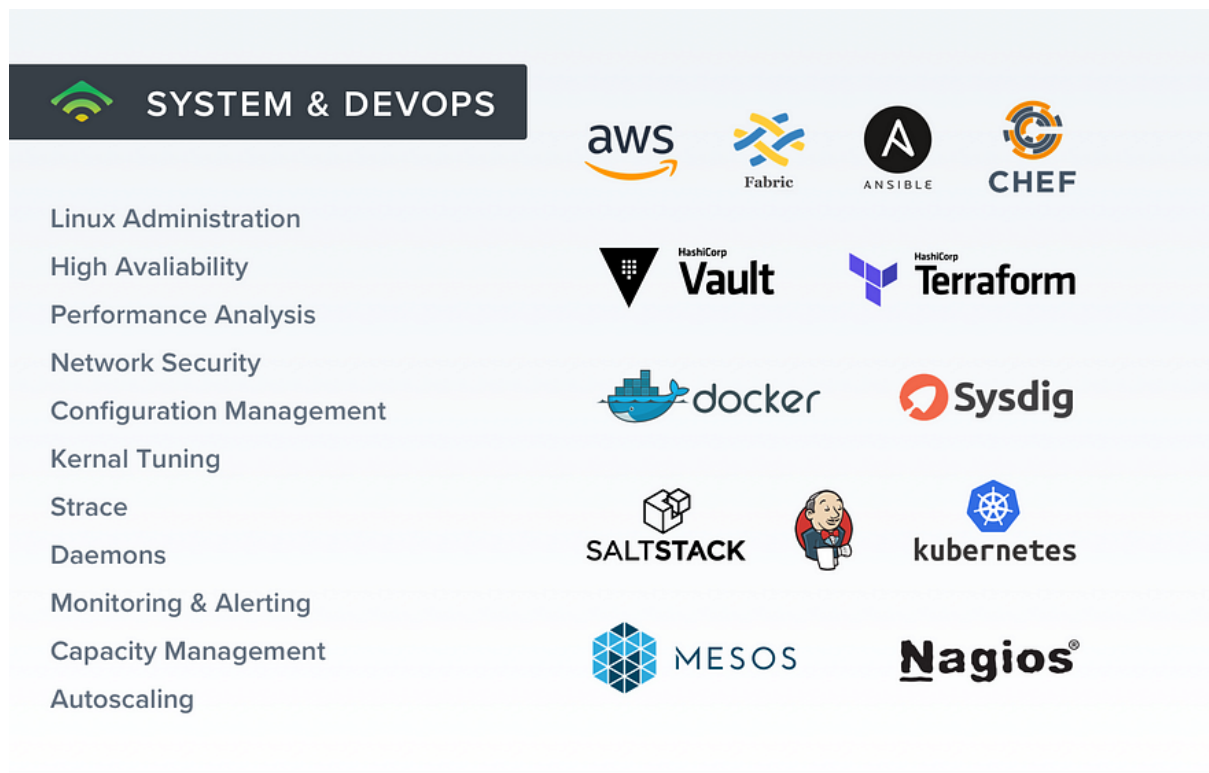
What separates great backend engineers from their peers, however, is not just their experience slinging APIs or coding with a particular framework, but whether they really know how the technologies they use work under the hood. The best backend engineers not only have experience with multiple languages and frameworks but may have written their own framework. They recognize the benefits and tradeoffs provided with any solution and pick the right tool for the job. They actively read the source code and technical docs for their chosen framework and the underlying programming language and may even have contributed extensions or modules or bug reports to the cause. Ultimately they prefer to treat no system as a black box and they value open and extensible architectures.



Many backend developers build data-intensive applications but more often than not they use an ORM without a deep understanding of how it really works nor how their data is organized. This means that unless their applications experience explosive growth they may never acquire the skillset to build applications that can truly scale. Because Klaviyo is focused on processing and extracting value from extremely large amounts of data in real-time we deeply value expertise with the “**data stack**” and “**distributed systems stack**”. We expect developers here to have experience setting up and configuring databases from scratch. They should understand how indexes work, how data is laid out on disk, the latency of various operations and the performance tradeoffs from database schema decisions.

Engineers with deep data tier expertise also understand not only why in memory solutions such as memcache or Redis are an integral part of an app, but how to model various data structures efficiently and achieve significant performance improvements by keeping data in RAM. The best data stack developers are also well versed in various NoSQL or NewSQL solutions such as MongoDB, Cassandra and others. They understand these systems are not a silver bullet and know when a non-relational database adds tremendous value to a project and is suitable for long term development. We also look for experience in distributed processing and engineers having a grasp of parallel processing and leveraging queueing systems such as RabbitMQ, SQS or even pub/sub systems in a data system such as Redis. A single machine does not scale and the most reliable systems are scale horizontally with commodity machines.

Most engineers only have cursory knowledge of this data tier because they have never had the need or curiosity to solve scalability or performance issues. However, there is no magic here and developers that really know the cardinality of their data and the latency of operations for data in memory, on disk or over the network really do prove themselves as they help to design scalable systems. Engineers at Klaviyo are expected to model these data characteristics in the technical specs for any system we build and we provide engineers with ample opportunity to deeply understand the interactions between high performance systems and stateful data stores.



The lowest layers of the stack and the necessary foundation of applications are the **systems & cloud stack**. This includes everything from systems administration, to the OSI 7 layer model and high availability to network security and configuration management best practices in the cloud. Traditionally, an operations team or even DevOps team may have consolidated most of this expertise and been a gatekeeper for production releases. However, at Klaviyo we believe DevOps is an engineering culture and not a team. All of our teams share operational responsibility and work to grow their systems expertise as they learn to launch ever more reliable and scalable systems.

Engineers with a passion for systems treat the building blocks of cloud services as an erector set of possibilities, and learn that the first rule of cloud is to assume that everything eventually fails. They seek to gain depth on Linux processing itself – daemonization to stracing™s processes and debugging core dumps, to optimizing kernel TCP settings.

In order to build and scale SaaS products investing in every layer of the stack is necessary. Full stack engineers are incredibly valuable because their versatility allows them to continually unblock their team and tackle any obstacle or velocity impedance ahead of them. When you™re a young startup you never know exactly what technical challenge you may need to tackle next. Thus, a CTO must be able to take on any technical initiative. At Klaviyo we look to talent that can operate in the same fashion, so they may one day be a CTO of their own company.

FRONT END

HTML / CSS / JS

Functional Programming

Design & UX

Browser Testing

Build Tools

CDNs & FE Performance

Responsive Design



BACK END

Server Side Programming

MVC Frameworks

ORM Layers

API Design

Serverless Apps



ActiveRecord

DATA TIER

Relational DBs

NoSQL / NewSQL

Scalable DB Performance / Indexing

Message Queues

Async Processing



SYSTEM & DEVOPS

Linux Administration

High Availability

Performance Analysis

Network Security

Configuration Management

Kernal Tuning

Strace

Daemons



Layers of full stack engineering