



How Digibee achieves velocity with scale



The ability to test SUT's without the burden of all the dependencies was an issue we wanted to solve.

Digibee is a fast-growing SaaS technology vendor that helps companies integrate their APIs and boost their digital business. Their no-code API studio allows companies to quickly build out ideas into digital experiences that accelerate time to market. As their customer base grows, they have placed increasing emphasis on system uptime and stability. An always-on API integration layer is critical to the businesses they serve, and ensuring availability is their main priority.

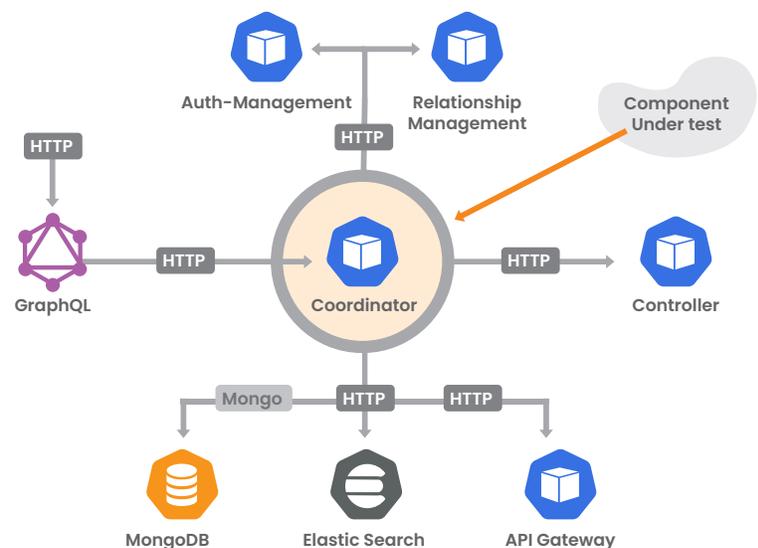
Challenge

As Digibee grows internationally from its initial customer base in Brazil into the US market, their engineering teams need to maximize development velocity. However due to the nature of their assembly platform, the myriad of API connections, containers, and growing complexity, ensuring quality at scale through conventional means has remained a challenge.

At the core of their challenge in ensuring scale is their Coordinator service, which receives various inbound requests, and requires working backends in order to present and structure customer functionality. UI testing against this Coordinator is ineffective and slow, particularly because its functionality is so far away from a highly variable UI application layer, thus, end-to-end environments are expensive to establish. Moreover, existing functionality must constantly be maintained while ensuring new features will work as designed.

“Just on the first run of a regression traffic replay we found a defect in how our API's were returning ID's. Although it was not affecting user experience yet, it was indeed a problem that needed to be addressed. That demonstrated how quick to value the use of SpeedScale could be.”

Peter Kreslins, Founder & CTO



Speedscale understands the relationship between inbound calls and the needs of resulting service dependencies.

Solution

Digibee runs Kubernetes in part due to its microservice-friendly platform, but also because of its inherent self-regulating, self-scaling capabilities. They have doubled their engineering team in the past year and are looking to build it out further. After hearing about Speedscale, Peter Kreslins, CTO chose to address Digibee's growing complexity and velocity needs with Speedscale.

By listening to the inbound/outbound traffic of the Coordinator, Speedscale understands the relationship between inbound calls and the needs of resulting service dependencies. Speedscale can then generate the variety of inbound calls and mock the multitude of backends from the cloud data it accumulates in seconds. By auto-generating diffs of recorded good traffic vs. new functionality, Digibee can quickly and automatically validate if any contracts have been broken, or if customer-facing features will be affected.

On just their first run of a regression playback they found a defect in how their APIs were returning IDs. Although it did not affect user experience in the short term, it was indeed a problem that needed to be addressed before the functionality was scaled.

After trying out Speedscale in this pilot, the solution was rolled out to the rest of the engineering organization to allow independent generation of traffic replays. Integration into CI/CD pipelines for immediate, repeatable API regression and performance feedback is currently underway.

Why Speedscale

Digibee needed a capability that allowed fast updates to their platform which supported Kubernetes and an extensive suite of APIs. Before Speedscale, testing through the UI and manual API test scripts were the only viable validation solutions -- both of which were tedious, unscalable and low coverage.

Two full-time employees were spending much of their time writing scripts or manually testing the application, which were deployed in expensive and complex end-to-end environments. With Speedscale's traffic replay framework, they could quickly auto-generate traffic replays and mocks that isolate their Coordinator service, automatically exercise APIs in a variety of conditions, and then return thorough reports of its performance SLIs and regression errors.



Benefits

- Teams can understand API architecture/dependencies, and navigate the complexity of their application without consulting documentation, senior engineers or architects.
- Ability to exercise SUTs without the burden of all the dependencies. Digibee's platform is very intricate and quickly putting together the whole architecture for testing purposes was an issue they wanted to solve.
- Deep visibility into API headers, cookies, security and body to reduce Mean Time to Discovery (MTTD) and Mean Time to Resolution (MTTR)
- Teams can move fast and get quick, thorough automated validation of builds in line with each build/delivery cycle.
- Digibee gets a one-stop-shop for API observability, traffic replay generation, mocking, regression and performance validation.

