

## CUSTOMER STORY

# Digibee maximizes development velocity with Speedscale

“

Before Speedscale, testing through the UI as well as manual API test scripts were the only viable validation solutions—both of which were tedious, unscalable, and low coverage.



# Digibee maximizes development velocity with Speedscale



Read how Digibee uses Speedscale's traffic replay for API observability, service mocking, regression and performance validation to boost development while ensuring quality.

## Company

[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 grew internationally from its initial customer base in Brazil into the US market, their engineering teams needed 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 became a challenge.

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

## Evaluating Solutions

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 as well as 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 a lot of their time writing scripts or manually testing the application, which were deployed in expensive and complex end-to-end environments.

Furthermore, Digibee also needed a solution that could scale and grow alongside their business. The company doubled their engineering team in the past year and has plans to build it out further. After hearing about Speedscale, Peter Kreslins, CTO at Digibee, chose to run a pilot to address Digibee's growing complexity and velocity needs.

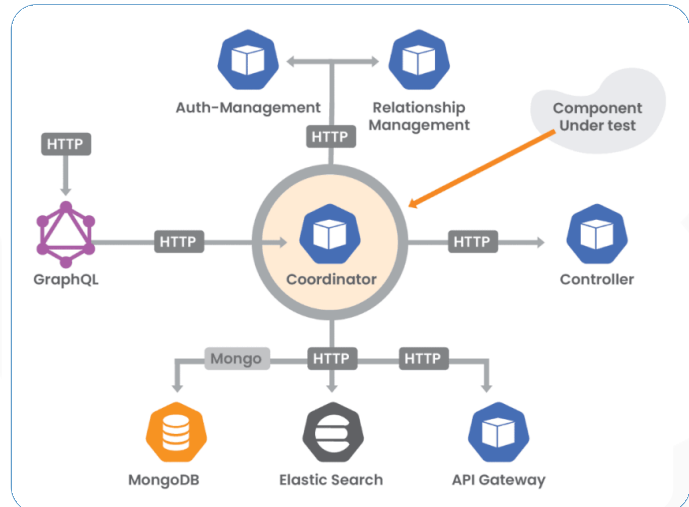


Before Speedscale, testing through the UI as well as manual API test scripts were the only viable validation solutions—both of which were **tedious, unscalable, and low coverage**.

## Using Speedscale

By listening to the inbound and outbound traffic of the Coordinator, Speedscale understood the relationship between inbound calls and the needs of resulting service dependencies. Speedscale could then generate the variety of inbound calls and mock the multitude of backends from the cloud data it accumulated.

By auto-generating diffs of [recorded good traffic vs. new functionality](#), Digibee quickly and automatically validated if any contracts had been broken, or if customer-facing features were affected.



### Feature Spotlight: Auto Diff View

The Speedscale Auto Diff View is an automated, fast way to validate you won't break anything in prod after an update.

[Read the Blog](#)



On 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.



On the first run of a regression traffic replay, we found a defect in how our APIs were returning IDs. 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



## Results & Future Outlook

Speedscale became Digibee's one-stop-shop for [API observability](#), traffic replay generation, [service mocking](#), regression, and performance validation. With Speedscale's traffic replay framework, Digibee 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. This led to the realization of some key benefits:

- Multiple, independent teams were able to understand API architecture and dependencies in a self-service manner, navigating 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 were able to move fast and get quick, thorough automated validation of builds in line with each build and delivery cycle

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.

### Feature Spotlight: Service Mocks

Service Mocks are a critical component of development and testing. Find out the Why and How behind our Kubernetes traffic-based mocks!

[Read the Blog](#)

