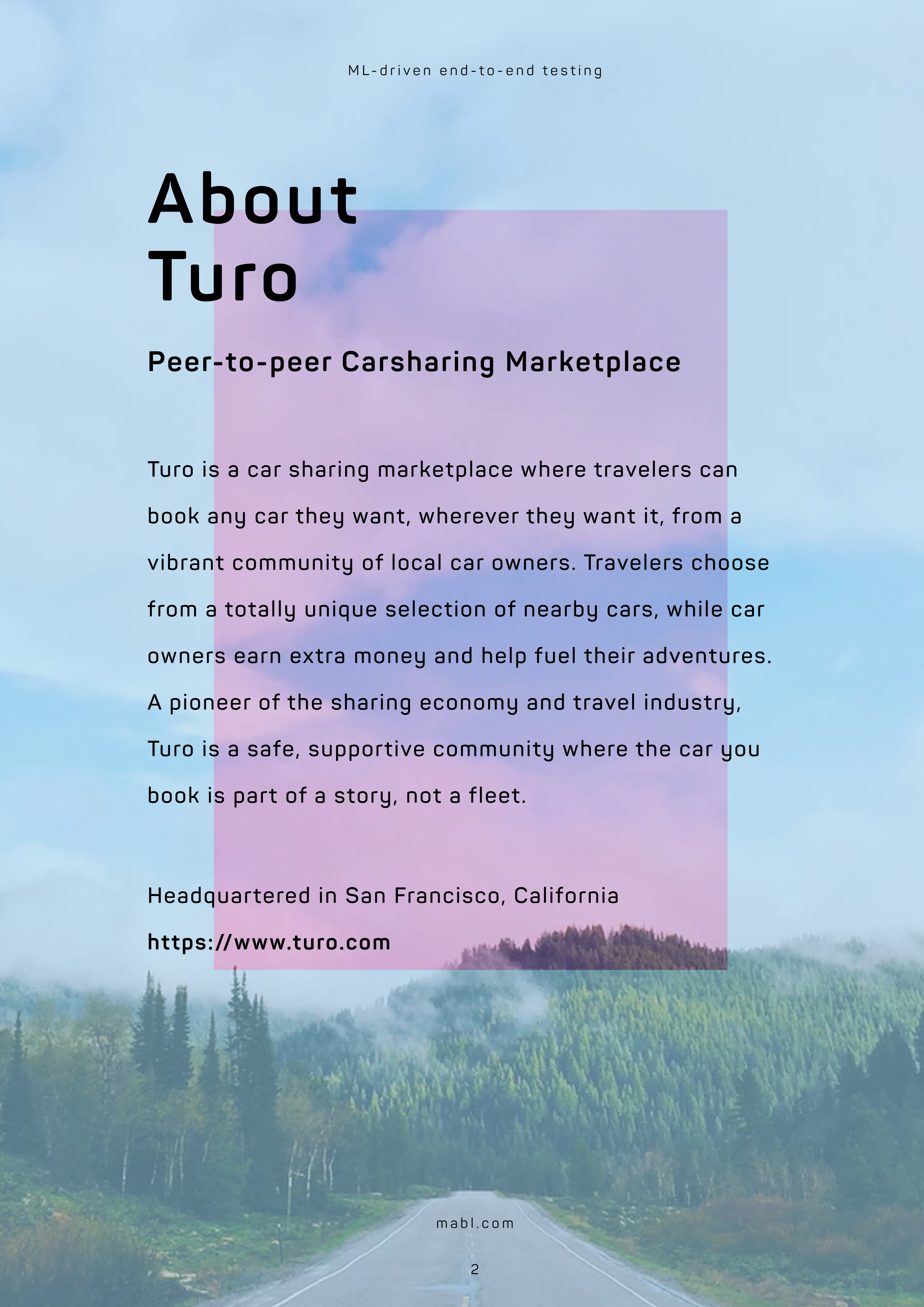
mabl

TURO

case study



About Turo

Peer-to-peer Carsharing Marketplace

Turo is a car sharing marketplace where travelers can book any car they want, wherever they want it, from a vibrant community of local car owners. Travelers choose from a totally unique selection of nearby cars, while car owners earn extra money and help fuel their adventures. A pioneer of the sharing economy and travel industry, Turo is a safe, supportive community where the car you book is part of a story, not a fleet.

Headquartered in San Francisco, California

https://www.turo.com

Describe mabl in 3 words:

"Pain free testing."

Jeff Zupka,
Engineering Director, Turo

Meet Jeff.

Jeff manages a team of 17, but Turo has 45 developers across their Boston, San Francisco, and London locations. He and his team are focused on expanding Turo internationally, and into new markets and verticals.

Jeff's team likes to deploy as quickly and as frequently as possible, shipping code 6-10 times a day. But the large development team only has two dedicated QA engineers. Therefore, Jeff's developers have a lot of ownership in writing tests, making sure that any new code is very well tested, and has automated tests that are passing. The QA team upholds 3 core values to support the development team: Have tests that are 1) easy to write, 2) fast, and 3) reliable.

The Challenge

> *Jeff and his team spent months setting up their own end-to-end testing infrastructure, but still had flakey tests. With mabl, Jeff was able to set up a reliable test suite for an untested admin interface in just 15-30 minutes.*

Jeff's team had an end-to-end testing platform, but found it hard to work with, and the tests unreliable. They decided to evaluate a different platform which took many fruitless months, between the selection of technology, configuration, and writing of bare-minimum tests.

After this big investment, they were still faced with unreliable, failing tests that were hard to diagnose.

The team would lose confidence in tests they didn't see the value of, and as a result, wouldn't invest the proper amount of time and effort into writing tests. The uphill battle prevented Jeff's team from testing their critical internal admin interface, which is used by several teams, as they only had the resources to test the most important customer-facing portions of their app.

Life After mabl

> *"It took maybe 30-min to an hour tops to create tests for our internal admin interface, and... we're getting all the benefits of offloading the maintenance and execution of tests, and the management of all the internal resources to mabl, so the team doesn't have to worry about them."*

What excited Jeff the most when he was exposed to mabl was her codeless nature, and the promise of being easy and intuitive enough to be used by anyone. With Jeff's small QA team, it's essential that they maximize their effectiveness, and being able to create tests with mabl's Trainer browser extension allows the development teams - and even support and customer success teams - to create tests for important user flows in the Turo app.

In a short amount of time, Jeff's team was able to add tests to their internal admin interface, which was lacking in test coverage. They also have tests triggered regularly via webhooks for two of the most important flows in the Turo customer-facing app: listing a car, and booking a car.

He also gets value out of the link crawler which checks for performance regressions, broken links, and broken JavaScript. mabl's ability to self-heal tests adds an unprecedented level of reliability. He notes: "when test failures happen, they're real failures that are actionable, which [his] team can respond to."

Moving forward

mabl is going to be a long term solution at Turo, and they have many plans for expanding its usage at the company.

After prioritizing and revamping the main flows that they're testing in the Turo app, they plan to add more complex flows, such as account modifications. The team will also add these flows to their software delivery pipeline so they kick off after every deploy. Jeff notes that even running tests at that frequency, mabl's cost is amazingly low.

They plan to learn how to apply mabl to other areas of the business, and other key workflows at the company. With the guidance of the QA team, they'll roll mabl out to other teams, such as the customer success and support teams.

> *"mabl really aligns with our values of being easy, fast, and reliable… It's easy to use and it really makes it easy to add tests in places that are actually really important to be testing that we just haven't had tests, or just haven't invested a lot of time in."*

mabl

mabl.com

Get a mabl demo