



carwow connects auto dealers and buyers in a transparent marketplace, creating better buyer experiences and incremental sales for dealers.

Environment:

- Heroku
- Rails
- Microservices-lite, one app per region, multiple apps per country
- Kafka

Need:

- An observability service that allowed them to maintain deep visibility into their infrastructure as they scale it to meet their flexibility requirements
- A single interface to trace request performance and the interaction of systems across their architecture from beginning to end

Love:

"Before Honeycomb, we couldn't really observe what was going on across the entire system. Honeycomb Tracing makes everything transparent again.."

Tracing is a game-changing thing for us. We loved Honeycomb before, but this has sped up our ability to observe how our entire system behaves.

Honeycomb has really opened our eyes to how our system behaves, how we can make it more efficient.."

- Mike Wagg
Technical Director, carwow



Challenge:

As carwow's successful online car review and selling business expanded into larger markets in new regions, their engineering team needed deep, system-wide visibility into the performance impacts of interactions among the components of their rapidly evolving infrastructure.

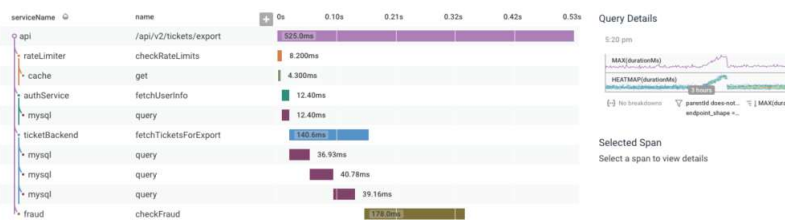
In scaling their systems to support growing traffic from one app to many apps and background jobs and events, carwow's engineering team has gained much-needed flexibility and agility, but as all systems architects and engineering leaders understand, increasing complexity leads to greater opacity:

Because the organization is growing quickly, it's becoming more difficult for one team of developers to have a good understanding of how the code they ship will interact with code shipped by other teams. Because the system is more loosely-coupled (which provides flexibility) than before, it's easier for new code to trigger jobs that have an escalating, ripple effect through the system.

"When there's an issue, New Relic can show the jobs running, the API requests but can't tell us the source of this thing that's happening right now, can't trace it back to a request, an earlier job in the chain."

Solution:

With access to the high-cardinality querying of wide events and deep tracing introspection supplied by Honeycomb, carwow can now find a request from a user, what messages it sent to Kafka, what jobs were queued, and rapidly arrive at the one request that caused a particular job to fire.



The engineers at carwow now know that when they see a problem in one system and see that it crosses multiple systems, they can use Honeycomb to follow the request through its entire life-cycle and understand the impact on different subsystems in the code. Honeycomb enables and turbo-charges cross-team collaboration to solve issues.

Results:

Having this visibility and collaborative power has allowed the carwow engineering organization to work together more effectively; they are able to quickly and easily find and eradicate inefficiencies and performance problems introduced by code making requests across components developed by different teams.