



Music streaming service meets capacity, availability and no-latency needs with Varnish Enterprise

Case Study:

# Music streaming disruptor



---

# Varnish supports delivery of real-time, lightning-fast metadata from cache

---

## Background

One of the original disruptors of the music industry changed how we consume and think about music.

The original platform pioneered peer-to-peer sharing technology and evolved into a fully licensed subscription-based streaming service.

The corporate entity behind the service offers various types of music streaming, including own-branded music streaming products for telcos and other partner deals as well as a radio interface for Sonos devices where songs are delivered from the service's backend.

## The challenge

### Managing and serving massive data and metadata

The music streaming service has a large database that contains massive volumes of metadata – essentially all of the critical information about tracks, artists, and customer data. Millions of songs and accompanying metadata was once almost prohibitively expensive to query in real time, but with cloud technology, this has changed.

To take advantage of faster, less expensive technology developments, the company wanted to move away from trying to serve data from a single, giant relational database. To do this, they needed to create a subset of the database to serve from different caches. When an album is queried with its list of songs, users want to see the track listing/song information – and this needs to happen instantly.

The company's senior manager of engineering explained, "We previously ran very big machines and used the open source version of Varnish for caching. We ran into efficiency issues when running multiple instances on one machine. We moved some of this to Google Cloud Platform (GCP) as part of our cloud migration and wanted to simplify it. Ease of use is critical on our end, while speed of content delivery is critical for the customer. Varnish is the most logical place to look for optimization, as it is the layer closest to the customer."

## Music streaming disruptor at a glance

### Organization

- Offers fully licensed music streaming services, including own-branded music streaming products for telcos and a radio interface for Sonos devices

### Challenge

- Manage and serve millions of media objects/metadata in cache at massive scale
- Reduce load on backend and avoid outages, ensure high availability
- Improve user experience with high-performance content delivery and blazing-fast response times

### Varnish Enterprise

- Varnish Enterprise for caching, Massive Storage Engine (MSE), Varnish High Availability (VHA), Varnish Controller, Varnish Broadcaster
- Professional Varnish support



## The solution:

### Varnish for high-speed backend API service at scale

Because Varnish Cache was already in place, it made sense to continue with Varnish and expand on its capabilities with Varnish Enterprise in creating a high-speed, high-performance backend API service that would provide catalog data for the media the service hosts.

“We had an explicit set of success criteria for this project,” the company’s engineering manager continued. “This included setting up Varnish High Availability (VHA), a transition from the Varnish Administration Console (VAC) to the Varnish Controller, a move from old to new servers and the ability to discontinue the use of Google marketplace, the use of Massive Storage Engine (MSE) to handle previous capacity issues, the addition of new servers into the Broadcaster feature to enable pre-warming, and an overall reduction in load.”

Ease of use is critical on our end, while speed of content delivery is critical for the customer. Varnish is the most logical place to look for optimization, as it is the layer closest to the customer

*- Senior Manager, Engineering,  
Music streaming disruptor*

To make this happen, Varnish Enterprise helped the team make the move to get up and running quickly. The company’s previous Varnish champion and engineer who had initially set up the open-source version had left the company, which was another reason for seeking expert migration support and expertise on the Varnish Configuration Language (VCL).

“It was very easy to get running, and everything worked flawlessly, particularly as the Varnish Enterprise team was there to assist every step of the way.”





## Results:

### Speed, efficiency and backend protection with caching

As a user-facing service, high-speed, near-instant content delivery is an essential requirement for the streaming service. Improving the response rate for requests instantly improved with Varnish Enterprise because traffic no longer had to make round trips to the service's data center, which added at least a 20 millisecond latency. By moving the Varnish cluster into GCP, latency was reduced to just a few milliseconds, which was much more efficient on the front end.

At the same time, even though there was still latency on the backend because the service's origin was still on-prem, it was masked by a near 85% cache hit rate on that cluster. A senior engineering manager stated, "It was urgent to get Varnish Enterprise implemented because of capacity issues with our on-prem servers. Getting Varnish in place meant that we could avoid outages at the same time as improving the cache hit rate. Although we did not measure empirically, this was light years beyond how things worked before."

It was very easy to get up and running and everything worked flawlessly, particularly as the Varnish Enterprise team was there to assist every step of the way

- Senior Manager, Engineering,  
Music streaming disruptor

### The future with Varnish

"Varnish has been rock solid and stable, and we have encountered no issues. We have discussed how we might continue and expand on the relationship, for example, when we move more services from on-prem to GCP. We could, for example, use the Controller for managing different VCL. We may also begin to explore using Varnish for caching with some internal processes as well."





Los Angeles - Paris - London  
Stockholm - Singapore - Karlstad  
Dusseldorf - Oslo - Tokyo

**VARNISH**  
SOFTWARE  
[www.varnish-software.com](http://www.varnish-software.com)