codecentric

Success Story

BetQuest

BetQuest

Aufgabe

- Technologiepartner für die Planung und Steuerung einer neuen Produktidee
- Entwicklung des interaktiven Quiz-Spiels BetQuest, bestehend aus einer hoch skalierbaren Microservice-Architektur und einer mobilen App
- Sicherstellen, dass zur medialen Einführung das System fehlerfrei ist und mit der maximal zu erwartenden Last zurechtkommt

Ergebnisse

- Eine Quiz-App für mehrere Zielplattformen
- Eine hochverfügbare Backend-Infrastruktur, die automatisiert skaliert, um täglich zur Sendezeit die auftretende Last zu verarbeiten
- Ein Redaktionssystem zur Pflege und Auswertung der Spielinhalte

Vorgehen

- Entwicklung eines ersten Prototyps in Form einer hybriden mobilen App
- Evaluierung von Technologien für ein elastisches Backend, das zur Sendezeit von Köln 50667 auf RTL2 problemlos mehrere hunderttausend Spieler bedienen soll
- Entwicklung der nötigen Komponenten in kleinen agilen Teams mit kurzen Feedbackzyklen
- Entwicklung und Durchführung von Lasttests von mehreren hunderttausend simultanen Spielern
- Betreuung bei der Produkteinführung

Projektziele

- Beratung und Umsetzung eines MVP für eine breite Produkteinführung
- Ein auf Betriebskosten optimiertes System

Die Demski Media ist ein Unternehmen aus der TV-Branche, das sich das Ziel gesetzt hat, eine neue Form der mobilen TV-Unterhaltung auf dem Markt zu etablieren. Die Demski Media mit Sitz in Köln ist hauptsächlich im Bereich der Postproduktion von TV-Inhalten tätig und war der Auftraggeber der codecentric AG.
In Form einer Partnerschaft mit der filmpool Entertainment, einem der größten TV-Produzenten in Deutschland, wurde für die Markteinführung ein täglich auf RTL2 ausgestrahltes und beim jungen Publikum beliebtes Scripted-Reality-Format, Köln 50667, gewählt.

BetQuest

Hinter dem Produkt BetQuest verbirgt sich eine interaktive Second-Screen-Anwendung, die den Zuschauer stärker an ein bestehendes TV-Format binden soll. Ziel ist es, den Zuschauer aktiv über die gesamte Sendezeit hinweg an dem Sendeinhalt zu beteiligen und somit einem bestehenden Sendeformat eine Art Live-Event-Charakter zu verleihen.
Mit Köln 50667 sollte in der ersten Phase bewiesen werden, ob ein interaktives Spielkonzept in diesem Umfeld gut funktioniert.

Die besonderen technischen Herausforderungen bestanden hauptsächlich auf Seiten des Backends, wo es ein verteiltes System zu entwerfen galt, das punktuell enorme Lastspitzen verarbeiten musste. Das Spielprinzip selbst ist eine Art Quizspiel, in dem den Spielern synchron zum ausgestrahlten Inhalt und zeitgleich auf den Endgeräten Fragerunden präsentiert werden sollen. Ein Spieleredakteur muss auf Basis einer produzierten Sendung den Inhalt der Frage- und Antwortenrunden in das System einpflegen. Weiterhin musste gewährleistet werden, dass während eines Spiels auf Anpassungen im Sendebetrieb reagiert werden kann.

Systemarchitektur

Ein Hauptaugenmerk bei der Systemarchitektur war eine elastisch-skalierbare Technologie samt Infrastruktur, die es erlaubt, das System in Zeiten, in denen kein aktives Spiel läuft, auf ein Minimum herunterzufahren und kurz vor den Spielen auf den benötigten Ressourcenbedarf wieder hochzuskalieren. Dieser Vorgang sollte weitgehend automatisiert und ohne Down-Zeiten für den Spieler erfolgen.
Die richtige Wahl der einzusetzenden Technologien war aufgrund der besonderen Anforderungen von zentraler Bedeutung. Deshalb wurden in einer frühen Phase schon eine Reihe von Prototypen entwickelt und gegen diese Rahmenbedingungen evaluiert.
Als Infrastrukturpartner fiel die Wahl auf die Amazon Cloud; die Services und die kostengünstige Flexibilität, die Amazon seinen Kunden in der AWS bietet, waren hierfür ausschlaggebend.

Als Basistechnologie für die clusterbasierte und Microservice-orientierte Anwendungsarchitektur

entschieden wir uns für das Java-basierte und reaktive Entwicklungsframework Vert.x, das für die Entwicklung und den Betrieb moderner verteilter Systeme eine Vielzahl von nützlichen Technologien bereithält.
Da auch die Persistierung der Spieldaten einer hohen Last ausgesetzt sein würde und man ebenfalls in der Lage sein wollte, flexibel und schnell zu skalieren, erschien Apache Cassandra für diesen Anwendungsfall als das optimale DBMS. Gemeinsam mit den DataStax-Experten der codecentric wurde in AWS ein leistungsfähiges Cassandra-Cluster aufgesetzt.
Unterstützt durch die Cloud-Experten der codecentric wurde das Know-how der Infrastruktur in Form eines strukturierten Ansible-Projekts zusammengelegt und somit nachvollziehbar und jederzeit wiederholbar abgebildet.

Mithilfe des ELK Stack von Elasticsearch wurde eine Lösung gewählt, die das System- sowie Anwendungslogging zentralisiert, um es in dieser dynamischen, verteilten Infrastruktur beherrschbar zu gestalten.

Um das Systemverhalten zur Laufzeit zu betrachten und Fehlverhalten frühzeitig zu erkennen, war weiterhin eine passende Monitoring-Lösung vonnöten.
Mit der APM-Lösung Instana von dem gleichnamigen Startup fanden wir eine Lösung, die sehr einfach zu integrieren war und unseren Anforderungen an ein modernes Monitoring entsprach. Da sich Instana noch in einer frühen Produkteinführungsphase befand, erhielten wir als einer der ersten Kunden die Chance, unsere Anforderungen und Wünsche gezielt in die Produktentwicklung einfließen zu lassen.
Das anfängliche Monitoring der Basissystemmetriken wurden nach und nach erweitert um Sensoren, Dashboards, Knowledge für unsere eingesetzten Technologien, wie der Java Virtual Machine (JVM), Docker, Cassandra, ngnix und Elasticsearch. Auch sorgte die dynamisch visualisierte 3D-Sicht auf das cloud-basierte Gesamtsystem für Transparenz und Zufriedenheit bei unserem Auftraggeber.
Auf Basis dieser Systemlandschaft konnten im Anschluss Last- und Performancetests beweisen, ob das System in der Lage sein würde, Spiele mit mehreren hunderttausend Spielern gleichzeitig zu verarbeiten. Auch hierfür bot sich AWS an – schnell stand ein Testcluster zur Verfügung, der das Backendsystem von BetQuest in verschieden Testszenarien an seine Grenzen brachte.

App

Hybrid oder native – diese Frage beschäftigte die App-Planung. Die Vorteile einer hybriden App-Entwicklung überwogen, technologisch gab es kaum Einbußen und man war in der Lage, mit einen kleinen Team und einem Entwicklungsstrang zu starten, um zeitnah und iterativ mit einem runden MVP (Minimum Viable Product) an den Start zu gehen.

Fazit

Wir als codecentric waren von der Produktidee von BetQuest schnell überzeugt und sahen für uns die Chance, eine für den Kunden moderne und herausfordernde Lösung zu entwerfen.
Mit dem Einsatz diverser Expertisen unserer Mannschaft war es möglich, eine schlanke und agile Entwicklung zu betreiben, um schnell und kostengünstig ein gewünschtes MVP anzubieten, das technologisch auf modernen und leistungsfähigen Infrastrukturkomponenten beruht.
Das Entwicklungsteam setzte sich selbstorganisiert zu den jeweiligen Anforderungen zusammen und umfasste während der Entwicklungsphase im Schnitt drei Entwickler. Über einen PO Proxy auf unserer Seite wurden mit dem Kunden und den Designern kurze Feedbackphasen vereinbart, sodass Wünsche nach Anpassungen oder neue Anforderungen zeitnah in das Produktbacklog einfließen konnten.

Markus Bonsch, Senior IT Consultant
Standort Solingen

"BetQuest versteht sich als Startup-Produkt, das mit einer innovativen Idee und interessierten Kunden einen neuen Markt der Unterhaltung erschließen möchte.
Die codecentric ist und war für uns in mehreren Belangen ein entscheidender Partner, der sich als verlässlicher Berater für die technischen Fragen und Herausforderungen anbot sowie uns auch in der fachlichen Schärfung des Produktes in Richtung MVP kompetent unterstützte.

Weiterhin konnten wir uns bei der Produktentwicklung voll auf die Professionalität der codecentric verlassen und ihr freie Hand in der Organisation und Umsetzung des Entwicklungsprozesses geben. Belohnt wurden wir mit einem hohen Grad an Transparenz und einem zu jedem Zeitpunkt passenden Entwicklungsstand."

Rainer Demski, Geschäftführer der Demski Media GmbH