OBJECT ORIENTED ANALYSIS AND DESIGN

ITE1007

Review 3

Title:

Expedia – An Agile Model, a Case study

Slot: C2+TC2

Submitted To: Thippa Reddy G

Team:

Aradhya Mathur – 17BIT0146
Sankalp Jain – 17BIT0136
Sidharth Raj Miglani – 17BIT0145
Satin Sunil Jain – 17BIT0113
Ankush Kumar – 17BIT0119

ABSTRACT

Expedia, Inc. is the world's largest online travel company with an extensive brand portfolio that includes some of the well-known online travel brands. Collectively, Expedia covers virtually every aspect of researching, planning, and booking travel – from choosing the best airplane seat, to reading personal travel reviews of hotels, to planning what to do in a destination once you arrive. In this project we aim to research and focus on the methods used by I-transition and AWS to improve Expedia's platform performance and provide a great experience for its customer. Both of these companies follow Agile model of development where I-transition tried to build up the front end of the client and AWS focused more on the server-side development. This provides greater adaptability to frequently changing scope and ensures that client business grows successfully.

ITRANSITION

1. Need

Expedia powers bookings for many of the world's leading airlines and hotels, top consumer brands, high traffic websites, and more than 10,000 private label partners (affiliates) through Expedia® Affiliate Network (EAN) globally. However, the templates of the EAN solution provided neither consistent user experience in terms of functionality and UI, nor a competitive differentiator to new market entrants. Parallel to moving onto a unified platform in a SOA environment, Expedia teamed up with Itransition to revamp the EAN solution to redeem the leading position and boost revenues generated by the EAN line of business.

2. Challenges

The project was part of a larger complex program to revamp all main customer products. The program was executed by several teams across different continents and it required significant efforts for coordination. Moreover, these products in their turn should have been integrated with a wider information ecosystem (Global Distribution Systems' property databases, other services, etc), many parts of which had problems with legacy code and interfaces. Therefore, the project team had to overcome a number of interrelated challenges:

- **Complex decision making**. Being an international corporation, the customer had a complicated, formalized process of approving requirements and technical solutions, including discussing those at the top management level.
- **Dependency on other distributed teams.** We worked simultaneously with a US-based technical team on the customer side responsible for improving existing services and creating API for new features. To coordinate the efforts we introduced joint planning and regular status tracking activities.
- **Troubles with legacy code**. The US team faced serious troubles with legacy code and a complex, time-consuming process to change API for most cases, which delayed the delivery of some functionality to the Itransition team.

- **Change of the main technology**. After several months of work, the customer asked to abandon Dojo as the main client JavaScript framework and to migrate to YUI 2.0.
- **Changing requirements and project scope**. To tackle the problem of sudden twists in product requirements, the Itransition team designed a formal change management process integrated with the process of requirements supply to the development team.

3. Solution

- **Technologies Enabling Specific Business Functions**

  From the functional standpoint the application consists of 2 parts:

  - the customization panel (used by affiliates);
  - the front-end part (used by consumers).

  There were different requirements to those two UI parts therefore different UI technologies were chosen to implement them, but both used the same codebase for the back-end services.

- **New High-Performance Scalable Platform**

  Itransition accomplished a very impressive feat of creating and launching a completely new template into production in under a year! All of the feedback received from Expedia, including the company's President and CEO, and also EAN affiliates were nothing but positive. In addition to custom application development, Itransition's team provided maintenance and support services.

4. Key achievements:

- Improved and simplified booking path for a range of travel products;
- Improved page template loading times compared to the existing ones;
- Improved design customization and discoverability of customization options by affiliates;

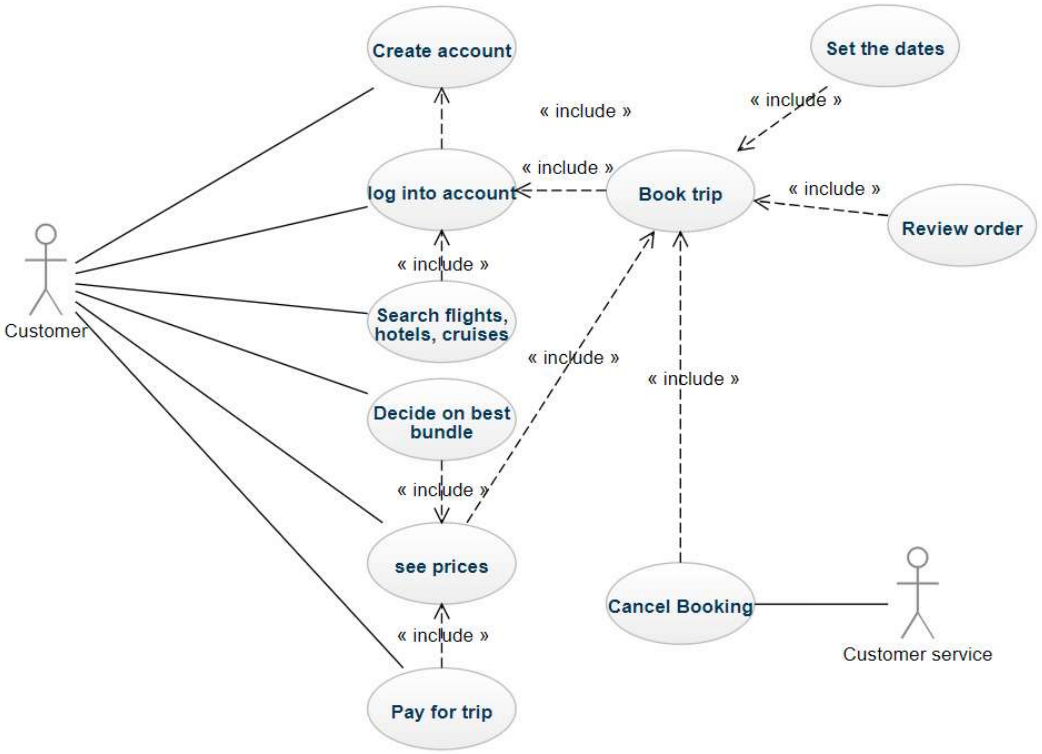
- Provided affiliates with a greater degree of flexibility on design and layout, as well as functional customization;
- Enabled Expedia to easily migrate existing affiliates onto the new template;
- Boosted user experience of consumers;
- Improved overall affiliates' satisfaction with the EAN services;
- Ensured high performance and scalability – a capacity to support over 10,000 affiliates.

5. Benefits

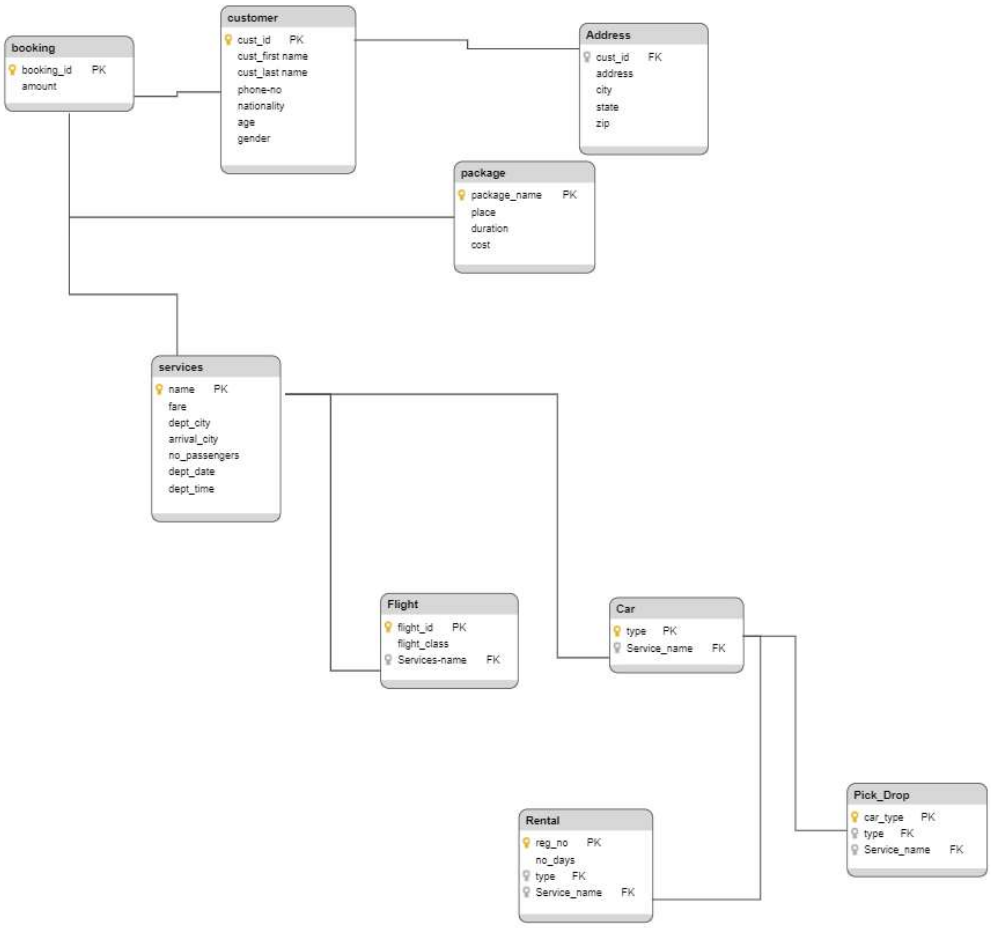
The advantages of the new templates help grow the EAN business:

- Single scalable and robust platform;
- Flexible branding capabilities;
- Consistent and contemporary user experience;
- Best in class UX principles (separation of content from presentation, semantic code, interaction/ presentation/ content layers);
- Adherence to the leading world's standards (W3C XHTML 1.0, CSS2, CSS3);
- Graded browser support and graceful degradation;
- Design provides maximum accessibility;
- Easily localizable templates available in more than 33 languages.

6. USE CASE DIAGRAM



7. CLASS DIAGRAM



AWS

1. Challenge

Expedia Group is committed to continuous innovation, technology, and platform improvements to create a great experience for its customers. The Expedia Worldwide Engineering (EWE) organization supports all websites under the Expedia Group brand. Expedia Group began using Amazon Web Services (AWS) in 2010 to launch Expedia Suggest Service (ESS), a typeahead suggestion service that helps customers enter travel, search, and location information correctly. According to the company's metrics, an error page is the main reason for site abandonment. Expedia Group wanted global users to find what they were looking for quickly and without errors. At the time, Expedia Group operated all its services from data centers in Chandler, AZ. The engineering team realized that they had to run ESS in locations physically close to customers to enable a quick and responsive service with minimal network latency.

2. Why Amazon Web Services

Expedia Group considered on-premises virtualization solutions as well as other cloud providers, but ultimately chose Amazon Web Services (AWS) because it was the only solution with the global infrastructure in place to support Asia Pacific customers. "From an architectural perspective, infrastructure, automation, and proximity to the customer were key factors," explains Murari Gopalan, Technology Director. "There was no way for us to solve the problem without AWS."

3. Solutions

- **Launching ESS on AWS**

  "Using AWS, we were able to build and deliver the ESS service within three months," says Magesh Chandramouli, Principal Architect. ESS uses algorithms based on customer location and aggregated shopping and booking data from past customers to display suggestions when a customer starts typing. For example, if a customer in Seattle

entered "sea" when booking a flight, the service would display Seattle, SeaTac, and other relevant destinations.

Expedia Group launched ESS instances initially in the Asia Pacific (Singapore) Region and then quickly replicated the service in the US West (Northern California) and EU (Ireland) Regions. Expedia Group engineers initially used Apache Lucene and other open source tools to build the service, but eventually developed powerful tools in-house to store indexes and queries.

By deploying ESS on AWS, Expedia Group was able to improve service to customers in the Asia Pacific region as well as Europe. “Latency was our biggest issue,” says Chandramouli. “Using AWS, we decreased average network latency from 700 milliseconds to less than 50 milliseconds.” Figure 1 demonstrates the ESS typeahead suggestion service running on AWS.

- **Running Critical Applications on AWS**

  By 2011, Expedia Group was running several critical, high-volumes applications on AWS, such as the Global Deals Engine (GDE). GDE delivers deals to its online partners and allows them to create custom websites and applications using Expedia Group APIs and product inventory tools.

  Expedia Group provisions Hadoop clusters using Amazon Elastic Map Reduce (Amazon EMR) to analyze and process streams of data coming from Expedia Group’s global network of websites, primarily clickstream, user interaction, and supply data, which is stored on Amazon Simple Storage Service (Amazon S3). Expedia Group processes approximately 240 requests per second. “The advantage of AWS is that we can use Auto Scaling to match load demand instead of having to maintain capacity for peak load in traditional datacenters,” comments Gopalan. Expedia Group uses AWS CloudFormation with Chef to deploy its entire front and backend stack into its Amazon Virtual Private Cloud (Amazon VPC) environment. Expedia Group uses a multi-region, multi-availability zone architecture with a proprietary DNS service to add resiliency to the applications.

Expedia Group can add a new cluster to manage GDE and other high volume applications without worrying about the infrastructure. “If we had to host the same applications on our on-premises data center, we wouldn’t have the same level of CPU efficiency,” says Chandramouli. “If an application processes 3,000 requests per second, we would have to configure our physical servers to run at about 30 percent capacity to avoid boxes running hot. On AWS, we can push CPU consumption close to 70 percent because we can always scale out. Fundamentally, running in AWS enables a 230 percent CPU consumption efficiency in data processing. We run our critical applications on AWS because we can scale and use the infrastructure efficiently.”

- **Using IAM to Manage Security**

  To simplify the management of GDE, Expedia Group developed an identity federation broker that uses AWS Identity and Access Management (AWS IAM) and the AWS Security Token Service (AWS STS). The federation broker allows systems administrators and developers to use their existing Windows Active Directory (AD) accounts to single sign-on (SSO) to the AWS Management Console. In doing so, Expedia Group eliminates the need to create IAM users and maintain multiple environments where user identities are stored. Federation broker users sign into their Windows machines with their existing Active Directory credentials, browse to the federation broker, and transparently log into the AWS Management Console. This allows Expedia Group to enforce password and permissions management within their existing directory and to enforce group policies and other governance rules. Additionally, if an employee ever leaves the company or takes a different role, Expedia Group simply make changes to Active Directory to revoke or changes AWS permissions for the user instead of inside of AWS.

- **Standardizing Application Deployment**

  The success of the ESS and GDE services sparked interest from other Expedia Group development teams, who began to use AWS for regional initiatives. By 2012, Expedia Group was hosting applications in the US East (Northern Virginia), EU (Ireland), Asia Pacific (Singapore), Asia Pacific (Tokyo), and US West (Northern California) Regions. Expedia Group Worldwide Engineering culled best practices from these initiatives to

create a standardized deployment setup across all Regions. As Jun-Dai Bates-Kobashigawa, Principal Software Engineer explains, “We’re using Chef to automate the configuration of the Amazon Elastic Compute Cloud (Amazon EC2) servers. We can take any AWS image and use scripts stored in Chef to build a machine and spin up an instance customized for a team in just in a few minutes.”

The team consolidated all AWS accounts under one AWS account and provisioned one Amazon VPC network in each Region. This allows each Region to have an isolated infrastructure with a separate firewall, application layer, and database layer. Expedia Group applies Amazon EC2 Security Group firewall settings to safeguard applications and services. Amazon VPC is completely integrated into Expedia Group’s lab and production environments. “The Amazon VPC experience for the developer is totally seamless,” says Bates-Kobashigawa. “Developers use the same Active Directory service for authentication and may not even know that some of the servers that they log onto are running on AWS. It feels like a physical infrastructure with its own subnets and multiple layers, and it’s also easy to connect to our on-premises infrastructure using VPN.” Expedia Group uses a blue-green deployment approach to create parallel production environments on AWS, enabling continuous deployment and faster time-to-market. “One of our metrics for success is the reduction of time to deploy within our teams,” says Gopalan. “We use this method to launch applications pretty quickly compared to a traditional deployment. Moreover, reducing the cost of a rollback to zero means we can be fearless with deployments.”

4. The Benefits

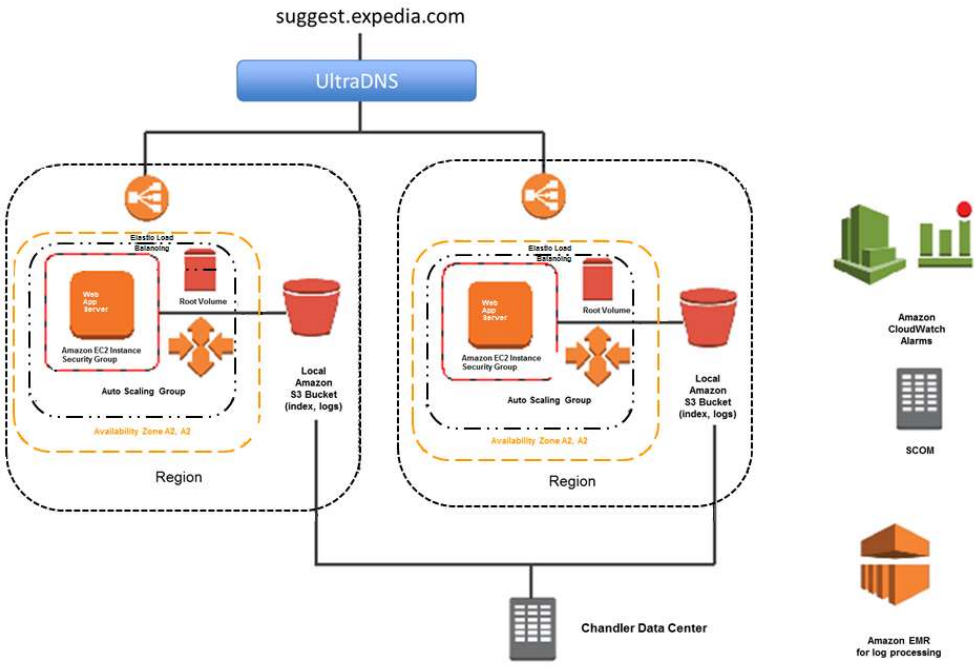
Expedia Group uses AWS to develop applications faster, scale to process large volumes of data, and troubleshoot issues quickly. By using AWS to build a standard deployment model, development teams can quickly create the infrastructure for new initiatives. Critical applications run in multiple Availability Zones in different Regions to ensure data is always available and to enable disaster recovery. Expedia Group Worldwide Engineering is working on building a monitoring infrastructure in all Regions and moving to a single infrastructure.

Generally, teams have more control over development and operations on AWS. When Expedia Group experienced conversion issues for its Client Logging service, engineers were able to

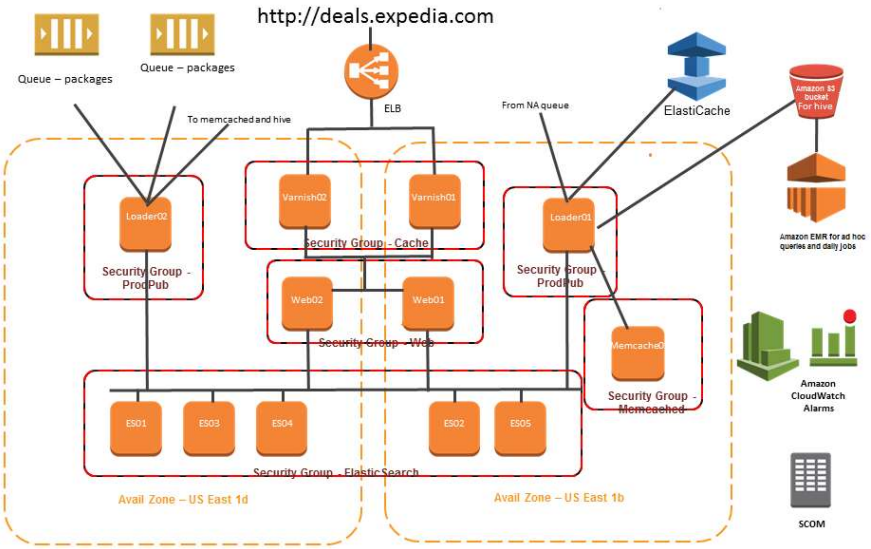
track and identify critical issues within two days. Expedia Group estimates that it would have taken six weeks to find the script errors if the service ran in a physical environment.

Previously, Expedia Group had to provision servers for a full-load scenario in its data centers. “To deploy an application using our on-site facility, you have to think about the physical infrastructure,” Bates-Kobashigawa explains. “If there are 100 boxes running, you might have to take 20 boxes out to apply new code. Using AWS, we don’t have to take capacity out; we just add new capacity and send traffic to it.”

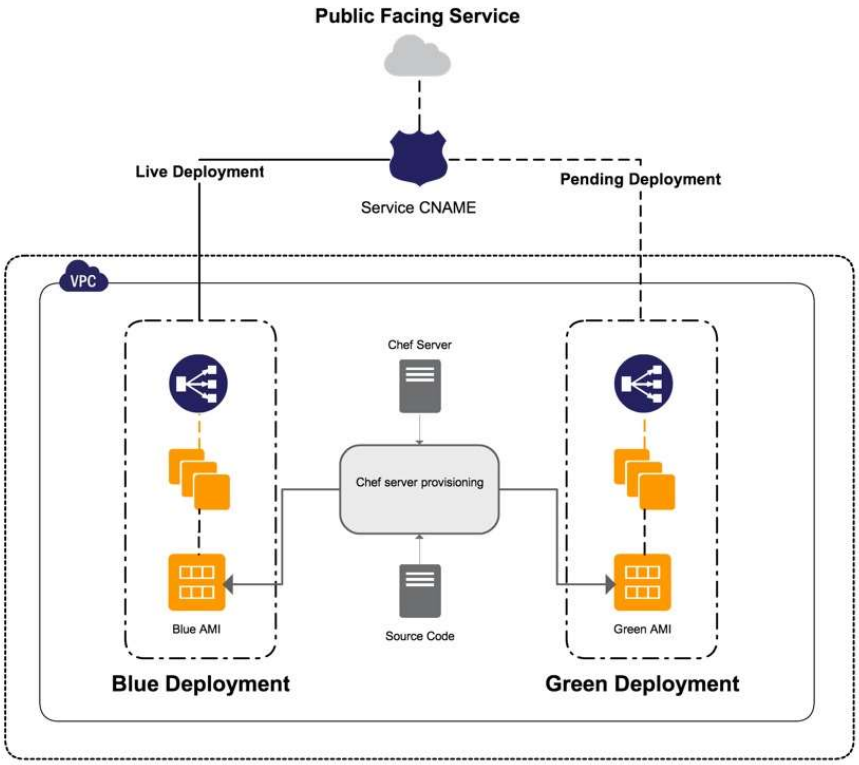
Expedia Suggest Service Architecture on AWS



Expedia Group Global Deals Engine Architecture on AWS



Expedia Group Standard Deployment Architecture on AWS



CONCLUSION

From this case study we can infer that ITransition and AWS played an important role in making the overall experience better for both the customers as well as the affialtes. The frontend upgrade ensured that the user does not feel overwhelmed by the information on the website and can easily navigate to his/her section. Here are the PSI metrics and Time to Page Usable metric to show how iTransition was able to successfully achieve this:

Release	Mobile PSI Change	Desktop PSI Change	Implementation Changes
R1	–	–	• **Inlined Critical CSS** for the above the fold content. • Removed **render blocking ads script** used to detect if ad-blocker is enabled. Instead, loaded it using AJAX.
R2	+14	+8	• Increased **browser caching time**, for partner team resources on the homepage
R3	0	0	• None
R4	-10	-2	• PSI scores dropped due to **uncompressed resources** from changes in internal libraries from other teams
R5	-10	-13	• PSI scores dropped due to **unoptimized images** in a new deals feature. Enabled **gzip compression** of resources for svgs in Storefront-web and Global Controls repo. • Moved subset of images from **media server to s3**, and set the browser cache time to 1 week.
R6	+4	+5	• **Image optimization** using Thumbor for images in Top deals section and other UI libraries. • Enabled **API compression** on more Ajax calls.
R7	+16	+12	• Applied above optimizations to the new features. • Target goal for PSI achieved.

Version	Time to Page Usable	Changes
R0	2.8s	• Image optimization and improving browser caching for some homepage modules. • Enable gzip wire compression for some required API calls
R1	2.7s	• Inlined critical CSS • Moved more importance, critical javascript to load before some ancillary site features .
R2	2.4s	• Cleaned up and removed redundancy from advertising markup on the homepage, saving ~130k or about 15%.
R3	2.2s	• Moved a significant amount of javascript responsible for rendering non-critical features from being synchronous to async, saving another 135k from page size.
R4	2.1s	• Reduced the size of the page by removing legacy unused markup, removing 5k from the mobile site. • Removed 12k worth of code that wasn't needed on mobile devices.

Along with handling server side problems such as Latency and Deployment, they also increased their security by shifting to AWS. Moreover, they were able to de-centralise their system organisation and easily scale up as the needs increased.

Baseline	**Page Usable (mobile tp50)**
Original Site on the legacy platform	7s
Improvements	**Page Usable (mobile tp50)**
Migrate codebase to standalone AWS application	4s
Rewrite main search wizard with a custom, fast rendered version instead of the common version used in all contexts.	3s
Reduce use of server side library JSP tags in favor of plain HTML.	2.8s
Remove unseen markup previously used for instant tab switching	2.6s
Inline critical CSS, and lazy load lazy load the rest	2.5s
Templatize elements on page, reduce page size by 10%	2.4s

Moreover, the entire process of revamping the model was done using agile technique which ensured that any future changes can be incorporated very easily, unlike the previous version which would take a lot of time of the developers to incorporate those changes.