

WAITING ROOM SOLUTION FOR ADIDAS.COM

Glasbergen T.

Hogeschool van Amsterdam, Amsterdam, Netherlands

Abstract. The aim of investigation is to analyze several Waiting Room solutions to meet the client expectations for a product sale. The objective of the study is to implement a waiting room solution whereby it's possible to randomly pick customers for sale without the need of a page refresh. The object is a Waiting Room with Akamai Services solutions. The subject is the architecture design for Waiting Rooms without refreshing the web pages, so customers choices can be memorized. The study used specific client requirements, standards of Akamai Services and basic design methods. The research is a qualitative research method to determine the best way to rebuild the waiting room application, based on desk- and field research.

Keywords: Akamai services, waiting room solutions, webpage refresh, architecture design.

Introduction. If a customer wants to buy a product at Adidas they see the Product Detail Page which is connected to the organization's backend. Whenever there is a Hype product on sale, a product with high demand and low supply, a Waiting Room page is served to the customers to protect the backend against heavy loads. For Adidas, the Waiting Room is a static html page which is refreshed every 30 seconds. On every page load a lottery is run for every customer waiting in line to determine if they can buy the product. Akamai is providing the technology to run this lottery called Visitor Prioritization Cloudlet [1], but there are other Cloudlet applications available as well that can protect a backend from overload [2].

The purpose of the work. The study is the implementation of a Waiting Room solution using Akamai services where the web page does not have to be refreshed.

Formulation of the problem. The Waiting Room page is refreshed every 30 seconds for all customers waiting in line. Adidas cannot show the customer anything other than a plain html page because of page load time, problems with data cache and the risk to overload the backend systems when API calls are made to retrieve data.

Solving the problem. The Waiting Room application is used to sell limited-edition products (hype products) on the website of Adidas. The demand is high, the supply is low and page traffic peaks during these sales. To prevent their backend from reaching the maximum capacity, Adidas has an online Waiting Room where customers can wait until they are randomly selected to go to the Product Detail Page (PDP) to make their purchase.

Customers that go to the Product Detail Page, shown in Figure 1, connect through one of Akamai's servers and based on probability settings like protocols, hosts, cookies, file extension, query string or geography the Visitor Prioritization Cloudlet, shown in Figure 2, determines if a user can pass the Waiting Room or not. When the browser page is refreshed every 30 seconds, Akamai runs the Visitor Prioritization lottery which determines if a customer gets the probability settings.

Using the Visitor Prioritization cloudlet is a stateful way of handling online traffic when the applications backend is in high demand.

Other Waiting Room solutions without the need of a page refresh are:

1. Stateless API Prioritization Cloudlet;
2. Stateless API Prioritization with stateful virtual endpoint polling;
3. Stateful Visitor Prioritization Cloudlet combined with Phased Release Cloudlet;
4. Stateful Visitor Prioritization with stateful virtual endpoint polling.

A mayor downside of involving another Cloudlet is that it adds more complexity and also maintaining more configurations. A virtual endpoint can be configured in a way that it runs the lottery, which is only an addition on existing configurations. The endpoint can be called in the background of a Waiting Room page, so a page refresh is not needed anymore.

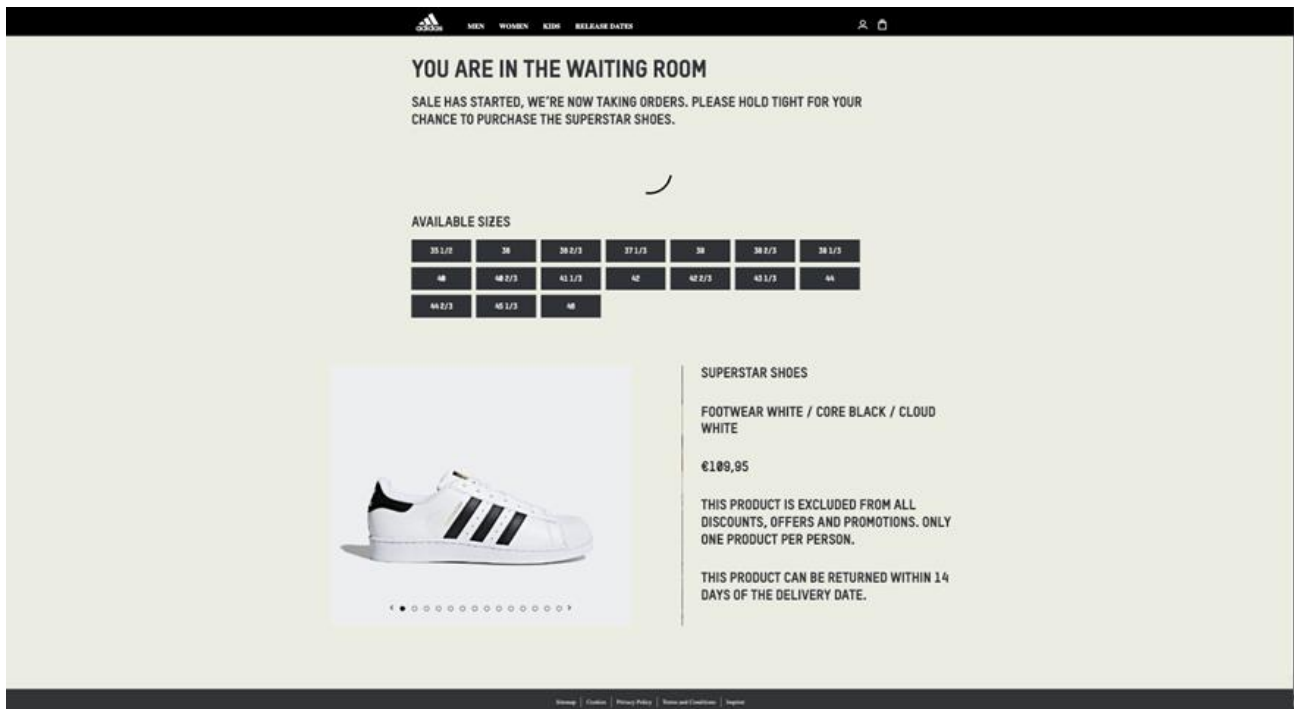


Fig. 1. Adidas Waiting Room page

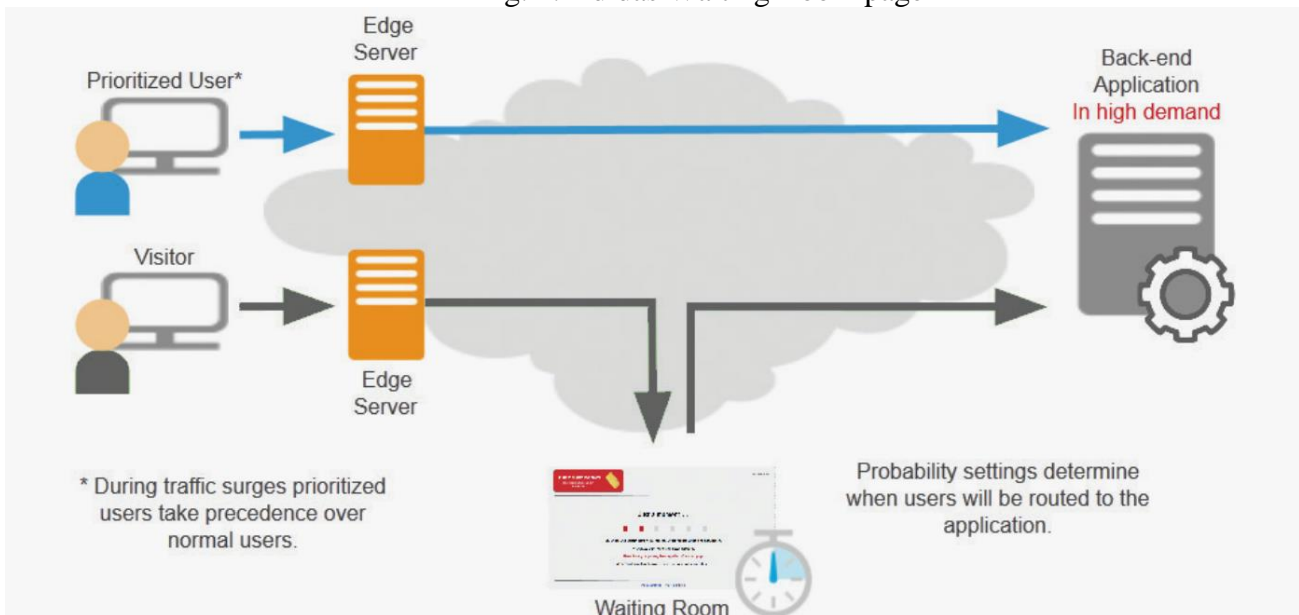


Fig. 2. Visitor Prioritization Cloudlet

Conclusion. Using the current solution Visitor Prioritization Cloudlet with a stateful virtual endpoint seems to be a powerful way to improve the Waiting Room solution. It will not only improve the customer experience, it will also allow adidas to build a new heavier UI, improve customer engagement and experiment more with improving the fairness of the Waiting Room.

References

1. Akamai Technologies. (2017). Visitor Prioritization Cloudlet. Retrieved. URL: <https://learn.akamai.com/pdf/VisitorPrioritization-CS.pdf>
2. Akamai Technologies. (2019a). Cloudlet Applications. Retrieved. URL: <https://www.akamai.com/us/en/products/performance/cloudlets/>