Assignment 8.2 – Securing Microservices

Jonathan Roland

WEB-420

12/06/2020

Instructor: Prof. Krasso

To: Example Corporation CEO

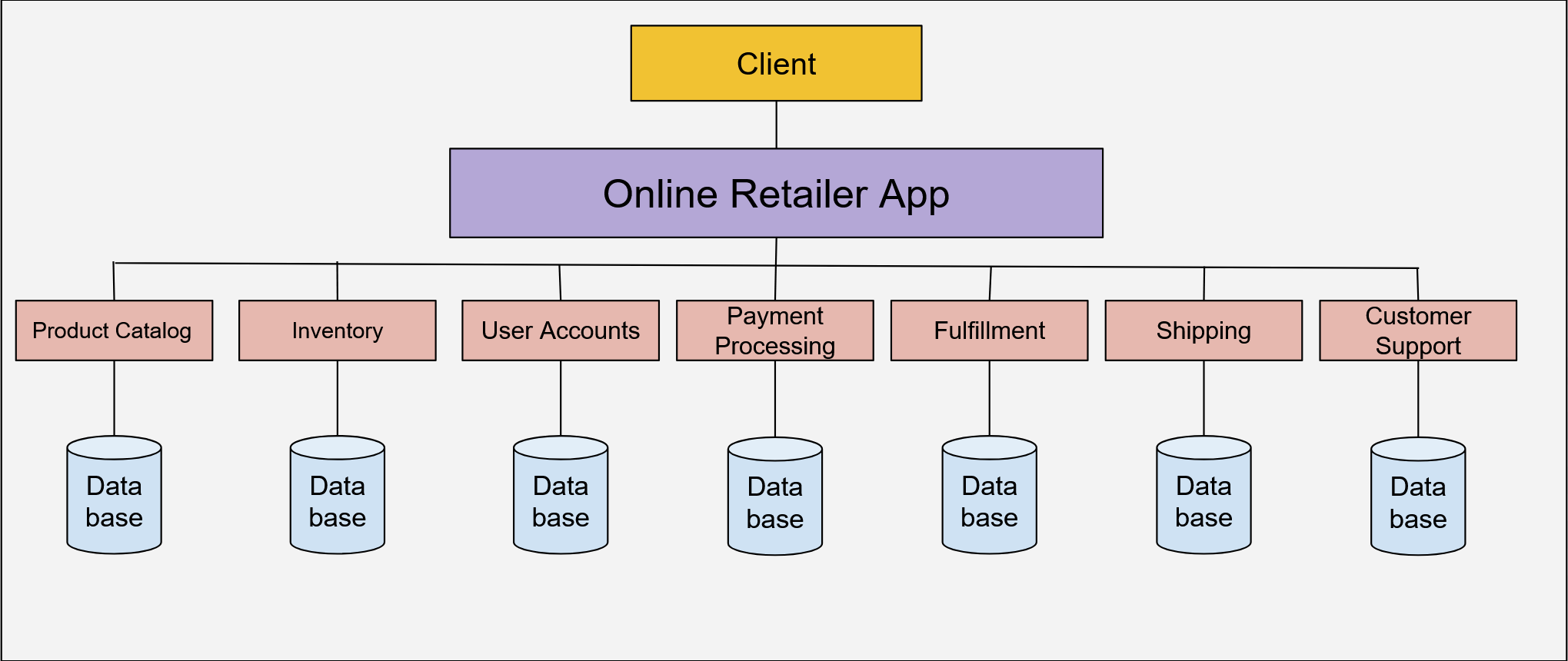
From: Jonathan Roland

Date: 6 December 2020

Subject: Securing Microservices

You may have heard the term ‘Microservices’ floating around the tech world lately. This approach to software architecture has significant advantages, but with new approaches come new avenues for bad actors to do damage. If we want to engage with this new way of building our applications, we must recognize our responsibility to keep them secure.

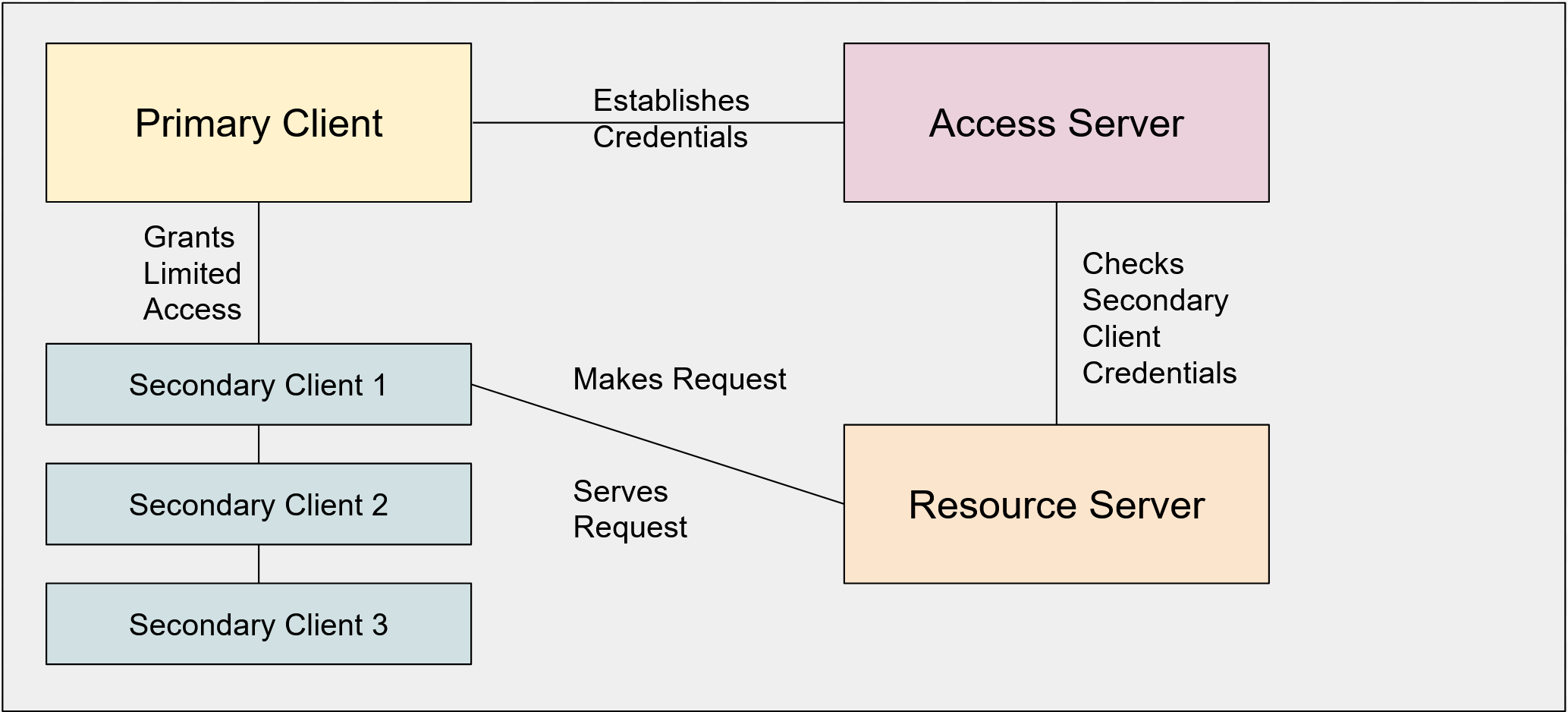
Microservices are the result of splitting up the areas of application functionality into narrowly focused responsibilities. Specifically, each microservice should focus on a single capability. In the example of an online retailer, we may have a division into microservices for the following concerns:

In a monolithic application, any changes or maintenance to any one of these systems risks potential impact on the whole system. By splitting these into microservices, small teams can maintain a narrow focus on their area of responsibility. Additionally, microservices can easily be swapped out for new vendors or versions. Another pair of advantages are speed and re-use of code: modular applications can be put together from existing microservice components like a budding young architect playing with a box of Legos.

A quick development cycle means a reduced window for security testing. Where monolithic applications would have months to thoroughly examine code, microservices applications may come together too quickly to allow such scrutiny. Another area of concern is surface area. Microservices applications are sprawling entities that communicate over internal and external networks to achieve their specialized tasks between components. This highlights a vulnerability area with so many communications needing to be transmitted rather than processed internally.

To address the concerns of quick development cycles we must adopt best practices for thorough security analysis before rushing to production. While the microservices model may allow us to save time in implementation, we cannot allow security review practices to be swept up in that current. As for network communication vulnerabilities, there are proven open standards we can examine to secure our communications.

The HTTP protocol provides for some basic obfuscation of security credentials, but in most cases that will not be enough. OAuth 1.0 and OAuth 2.0 provide better security implementations. The OAuth 1.0 standard allows clients to grant varied levels of access to accounts and applications on their behalf via an access server. When a server asks a subordinate client for credentials, the subordinate can refer the server to the access server. Essentially it says, “I don’t have the security credentials, but the primary client wrote me a permission slip to access that resource and you can find it at this access server”. If true, the resource server can verify this with the access server. This allows the client to withhold their actual credentials, revoke permissions for subordinate clients, and allows both the resource server and the primary client to forgo the need for credentials specific to that server. The OAuth 2.0 standard allows similar functionality but increases some areas of security. These include the length of time an authorization can be considered valid and accounting for the needs of non-browser client applications. Here we can see a basic model of OAuth:



Microservices offer significant advantages for speed of development, maintenance, code re-use and modularity of applications. The quick development time can easily shut out due diligence with security review steps, so we need to adopt a habit of mindfulness to that end. Considering the significantly distributed network communications that a microservices model implies, we must also make architectural choices early in development about which standard of security we will adopt. With these considerations in mind, we can successfully harness the power of microservices application architecture.

References

Singh, J. (2018, June 22). *The What, Why, and How of a Microservices Architecture*. Medium. https://medium.com/hashmapinc/the-what-why-and-how-of-a-microservices-architecture-4179579423a9

Robinson, R. M. (2015, October 7). *Microservices Are Powerful, but Pose New Security Challenges*. Security Intelligence. https://securityintelligence.com/microservices-are-powerful-but-pose-new-security-challenges/

Richardson, L., Amundsen, M., & Ruby, S. (2013). *RESTful Web APIs: Services for a Changing World* (1st ed.). O’Reilly Media.

Ramchandani, P. (2018, June 13). *What’s the difference between OAuth 1.0 and OAuth 2.0?* Packt Hub. <https://hub.packtpub.com/what-is-the-difference-between-oauth-1-0-and-2-0/>