Providing an Access Control Layer for Content Distribution Networks

Lukas Klingsbo

September 21, 2015

Abstract

TODO: Abstract
Keywords:

Contents

1 Introduction

Developing larger projects containing static content usually involves using a Content Distribution Network to be able to scale to a large user base. The commercial Content Distribution Networks are usually fairly easy to use, the content that is to be used in a project is usually simply uploaded and then directly available to the public. For secret content this can be a problem and that is what this thesis is about. This work examines ways of enforcing access control on content and groups of content in the form of views. A system was developed to make the underlying theory work in practice.

2 Related Terminology

2.1 Technologies

2.1.1 React

React is a JavaScript library for building user interfaces. React uses both its own virtual DOM and the browser's, this makes it able to efficiently update dynamic web pages after a change of state through comparing the old virtual DOM with the resulting virtual DOM after the state change and then only update the browser's DOM according to the difference between the virtual DOMs [1].

2.1.2 Flux

2.1.3 Scala

Scala is a multi-paradigm programming language. It most commonly runs on the JVM and compared to Java it supports most functional programming features at the same time as it supports object oriented programming [2].

2.1.4 JSON-RPC

2.1.5 TODO: Insert persistent storage here

TODO: Write down related terminology, if any

2.2 Abbreviations

2.2.1 CDN

Content Distribution Network - Replicates content to several servers, usually spread out geographically, and serves

3 Background

3.1 About Uprise

Uprise is a company based in Uppsala, Sweden.

3.2 The current system

Maybe write about battlebinary?

3.3 Problem description

Having

Battle Binary

What is needed * Security layers * Views * Virtual file structure * Versioning of content * Multi project support * Auth and audit logs * Users

4 Methods for determining implementation details

This chapter introduces the different methods used to determine how the new system should be implemented, which DBMS it should use and how the estimation of long term scaling was done.

5 Security of the system

5.1 Authorization

5.2 Audit logs

6 Resulting system

6.1 Scalability

7 Discussion

8 Summary

8.1 Conclusions

8.2 Future work

References

[1] A. React, "Javascript library for building user interfaces," 2014.

[2] M. Odersky, P. Altherr, V. Cremet, B. Emir, S. Micheloud, N. Mihaylov, M. Schinz, E. Stenman, and M. Zenger, "The scala language specification," 2004.