Providing Access Control and Copy-on-Write for Content Distribution Network Assets

Researching Copy-on-Write solutions

Lukas Klingsbo <lukl8671@student.uu.se>

Department of Information Technology

Uppsala University

April 4, 2016

UPPSALA
UNIVERSITET

Outline

1 Background

- Uprise
- CDN
- BattleBinary
- Problem
- Copy-on-Write
- Related Work
- Solutions Idea

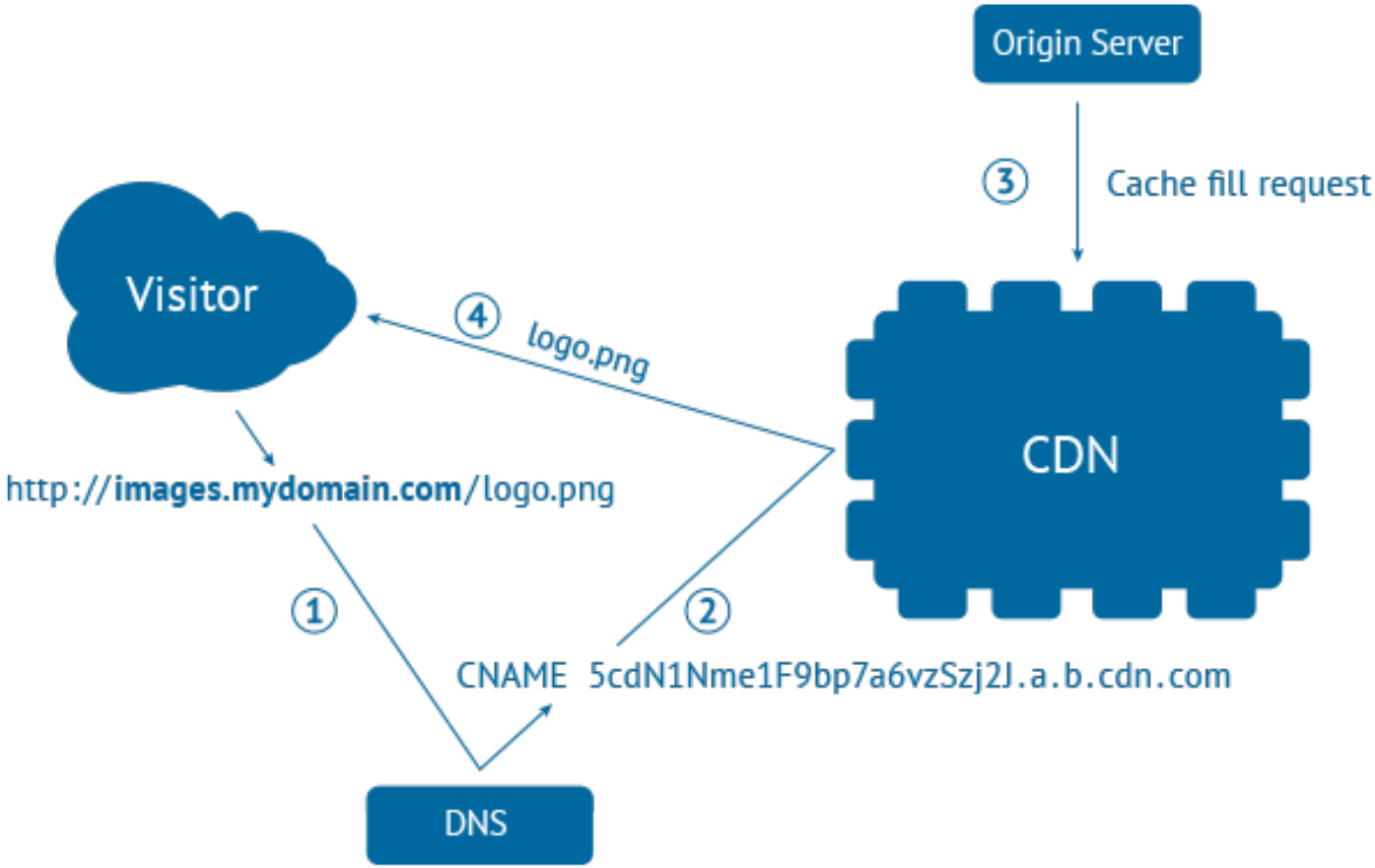
2 What has been done

3 How to proceed

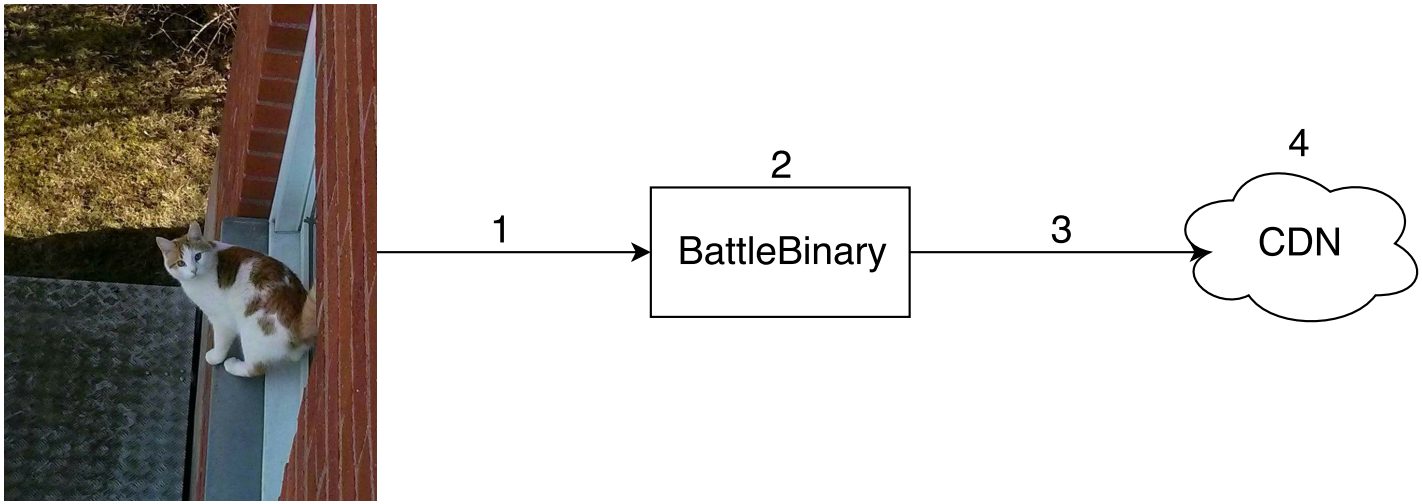
UPRISE

GHOST

Content Distribution Network (CDN)

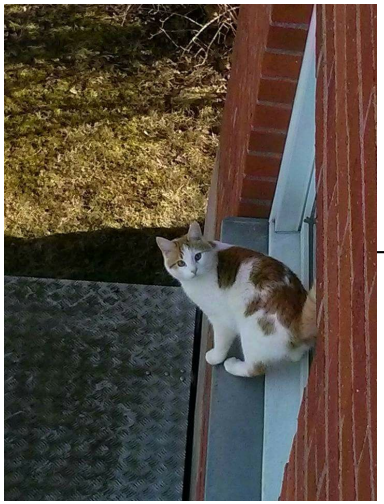


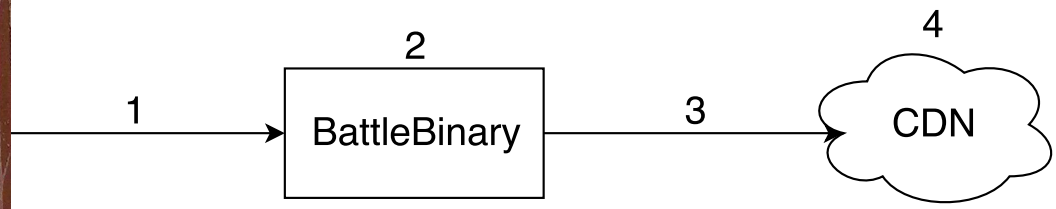
BattleBinary



- 1. Image is uploaded to BattleBinary
- 2. Filename + Part of file's hash = Asset Identifier
- 3. Upload image to CDN
- 4. Image is available to everybody with link `http://ea.akamaihd.net/cat-f1ee0283b6accd6.jpg`

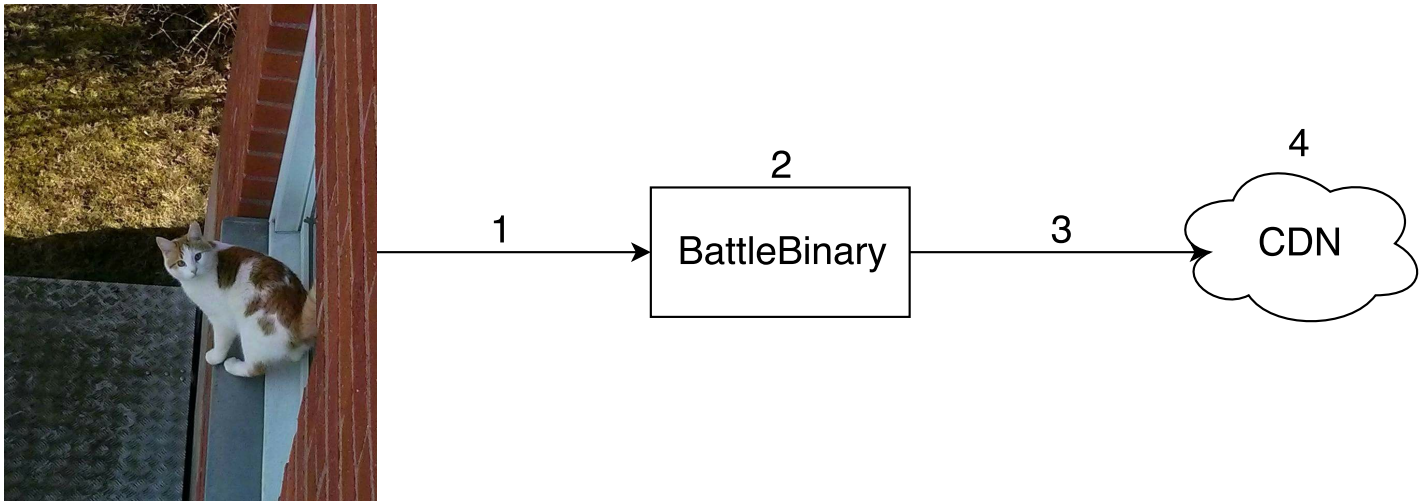
BattleBinary





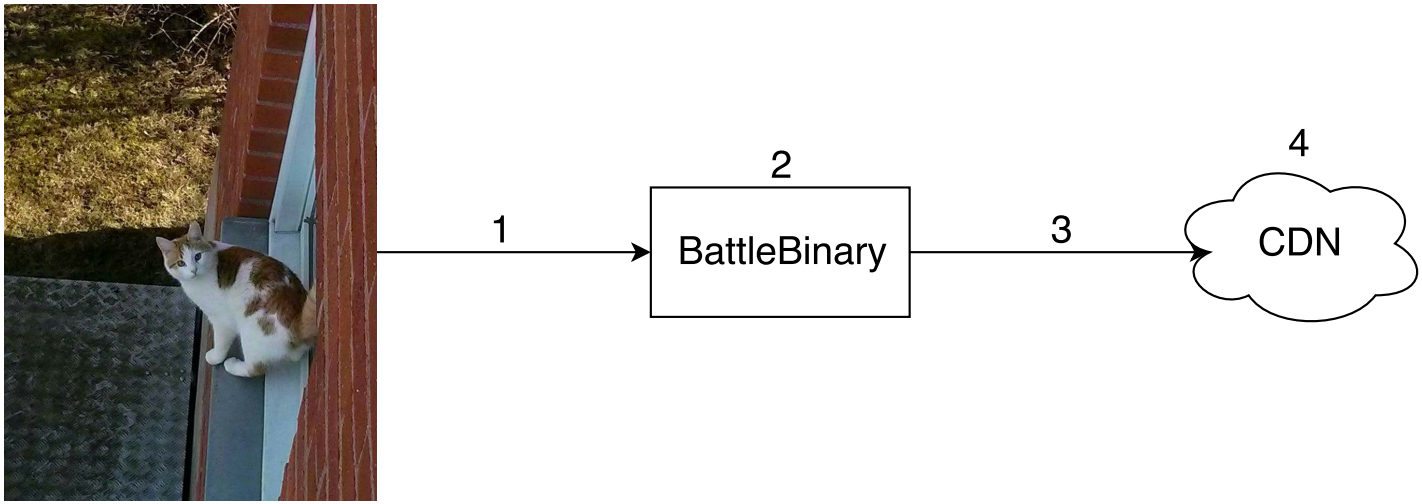
- 1. Image is uploaded to BattleBinary
- 2. Filename + Part of file's hash = Asset Identifier
- 3. Upload image to CDN
- 4. Image is available to everybody with link `http://ea.akamaihd.net/cat-f1ee0283b6accd6.jpg`

BattleBinary



- 1. Image is uploaded to BattleBinary
- 2. Filename + Part of file's hash = Asset Identifier
- 3. Upload image to CDN
- 4. Image is available to everybody with link `http://ea.akamaihd.net/cat-f1ee0283b6accd6.jpg`

BattleBinary



- 1. Image is uploaded to BattleBinary
- 2. Filename + Part of file's hash = Asset Identifier
- 3. Upload image to CDN
- 4. Image is available to everybody with link
  `http://ea.akamaihd.net/cat-f1ee0283b6accd6.jpg`

Problem

- insecure
- impractical
- wont scale

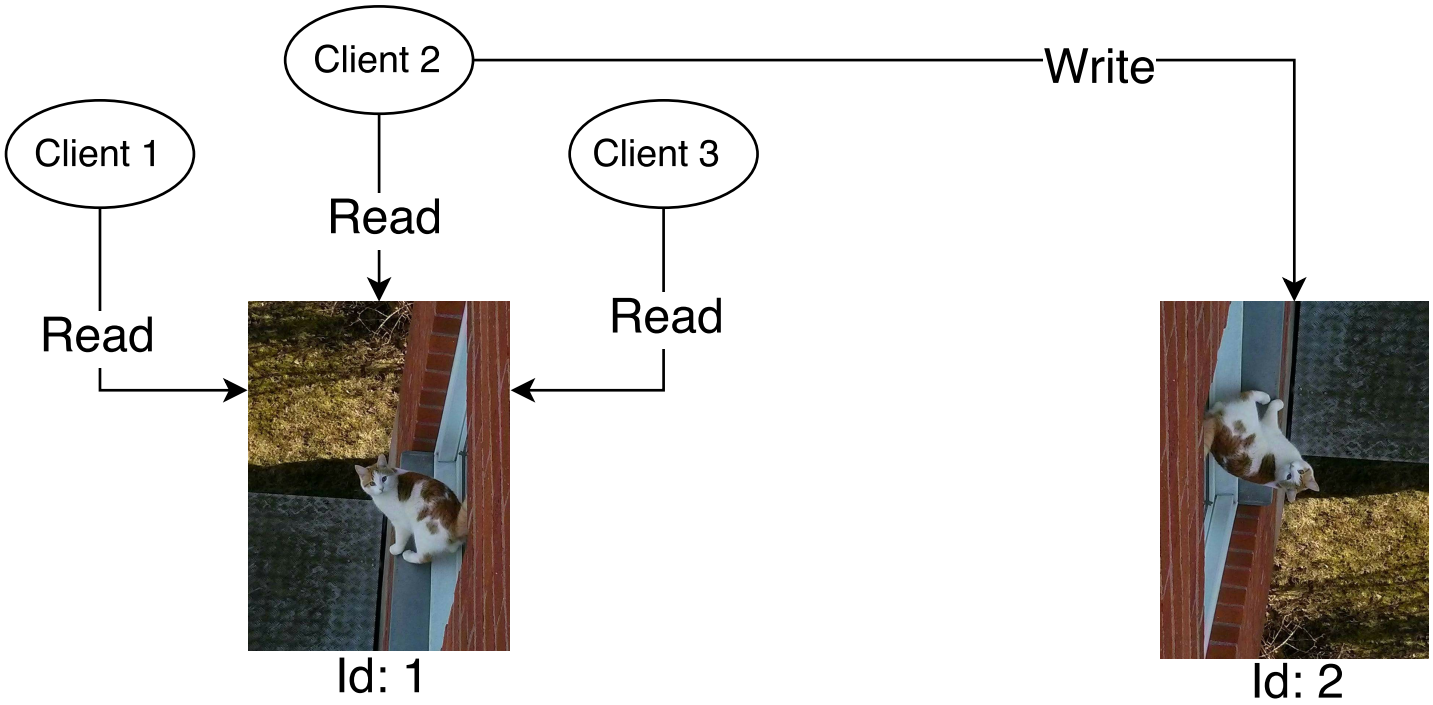
Problem

- insecure
- impractical
- wont scale

Problem

- insecure
- impractical
- wont scale

Copy-on-Write



*IDs are GUIDs in real implementations

Effects of COW

- No accidental incremental changes or race conditions
- No locks needed $\rightarrow$ Scalability
- Take snapshots of system in constant time
- Needs some form of garbage collection/awareness

Effects of COW

- No accidental incremental changes or race conditions
- No locks needed $\rightarrow$ Scalability
- Take snapshots of system in constant time
- Needs some form of garbage collection/awareness

Effects of COW

- No accidental incremental changes or race conditions
- No locks needed $\rightarrow$ Scalability
- Take snapshots of system in constant time
- Needs some form of garbage collection/awareness

Effects of COW

- No accidental incremental changes or race conditions
- No locks needed $\rightarrow$ Scalability
- Take snapshots of system in constant time
- Needs some form of garbage collection/awareness

Related Work

- Mach kernel
- Btrfs
- Programming Languages

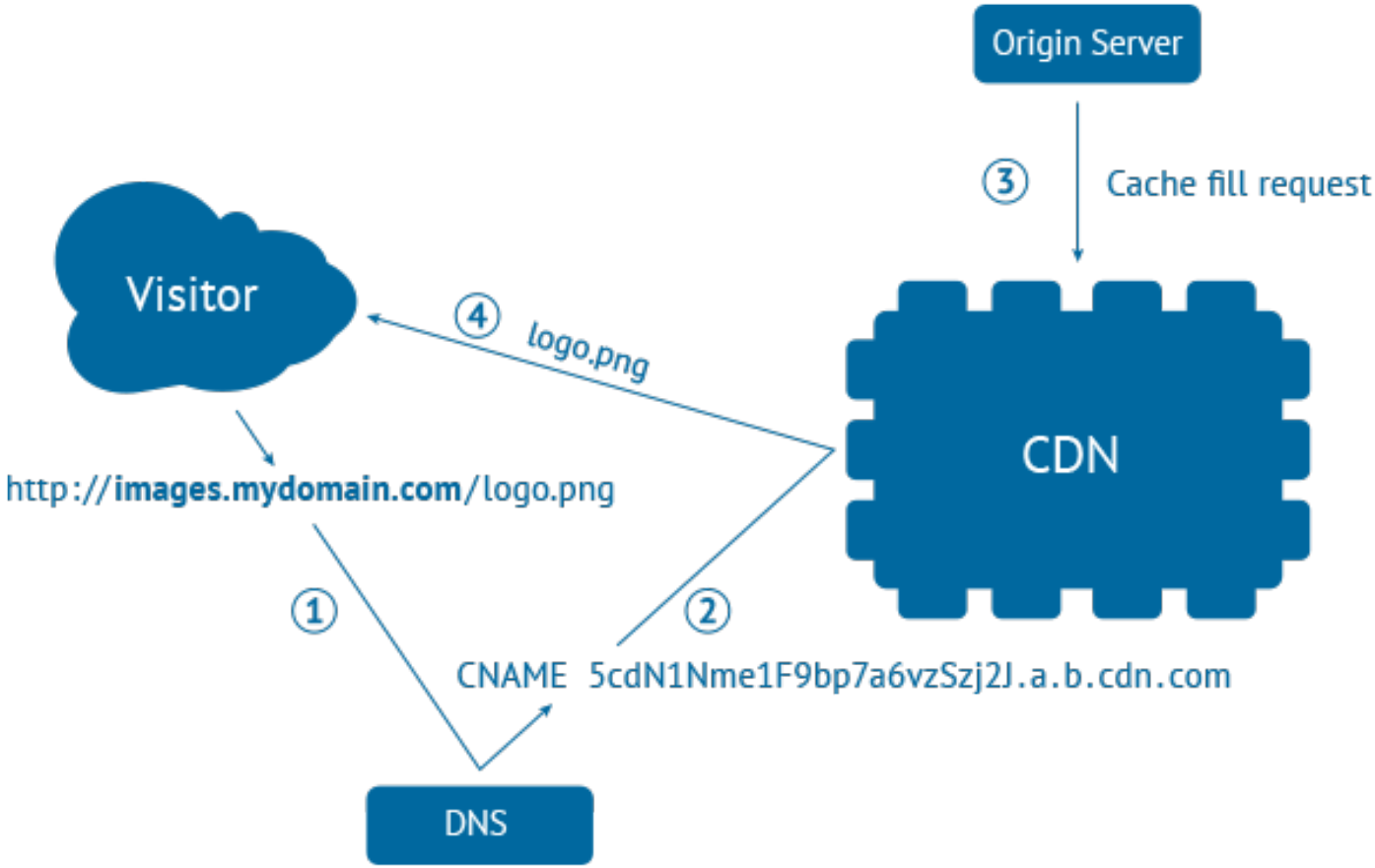
Related Work

- Mach kernel
- Btrfs
- Programming Languages

Related Work

- Mach kernel
- Btrfs
- Programming Languages

Solutions Idea



Example Operation

(a) Tree T_p (b) T_p cloned to T_q

BTRFS: The Linux B-Tree Filesystem, O. Rodeh et al

What has been done

- Researching Copy-on-Write implementation
- Model checking implementation idea in JPF
- Implemented the fundamental back-end parts

What has been done

- Researching Copy-on-Write implementation
- Model checking implementation idea in JPF
- Implemented the fundamental back-end parts

What has been done

- Researching Copy-on-Write implementation
- Model checking implementation idea in JPF
- Implemented the fundamental back-end parts

How to proceed

- Comparing Copy-on-Write implementations
- Fully implement and test the system
- Writing more on the report...

How to proceed

- Comparing Copy-on-Write implementations
- Fully implement and test the system
- Writing more on the report...

How to proceed

- Comparing Copy-on-Write implementations
- Fully implement and test the system
- Writing more on the report...