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

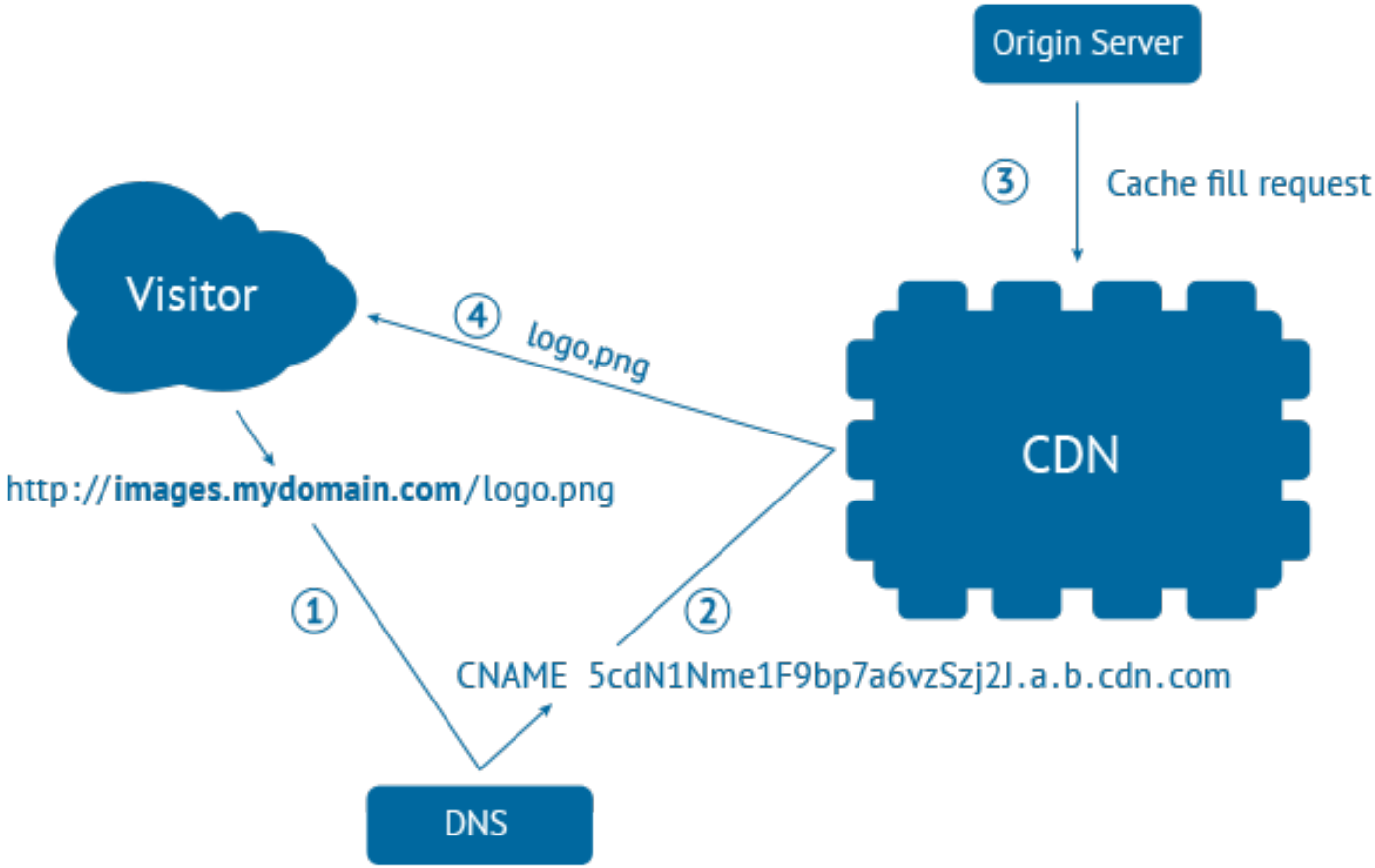
UPPSALA
UNIVERSITET

Outline

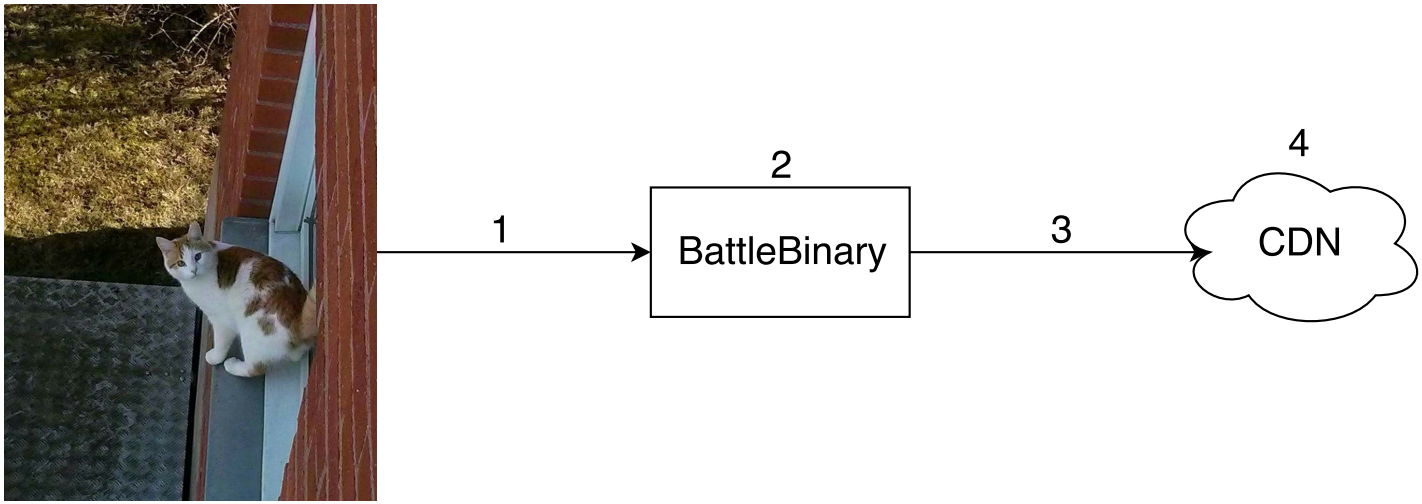
UPRISE

GHOST

Content Distribution Network (CDN)

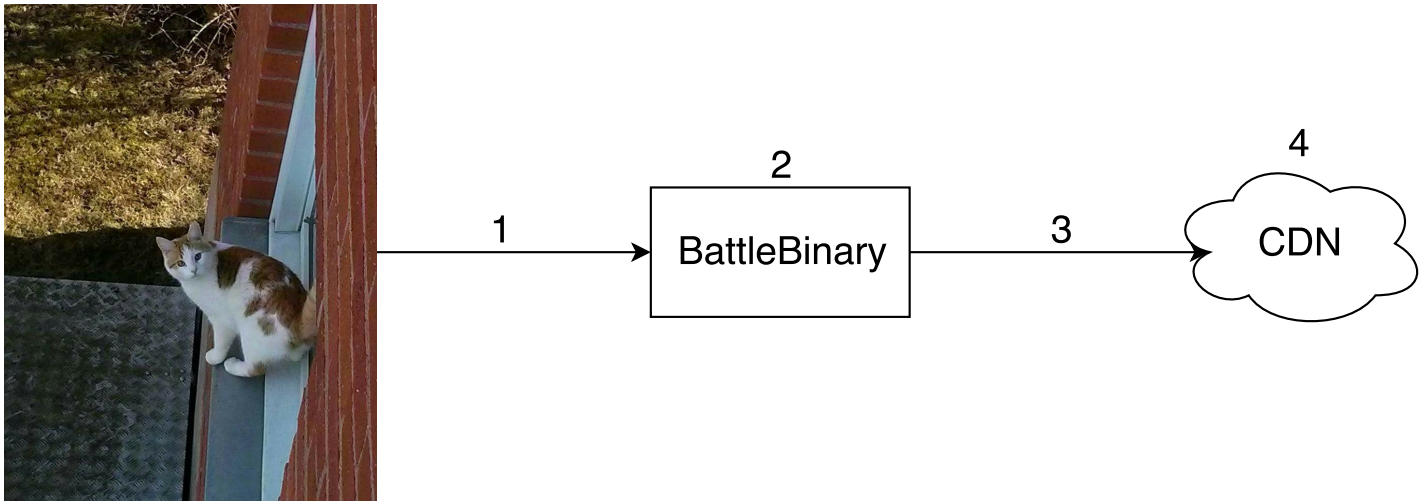


BattleBinary



- 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`

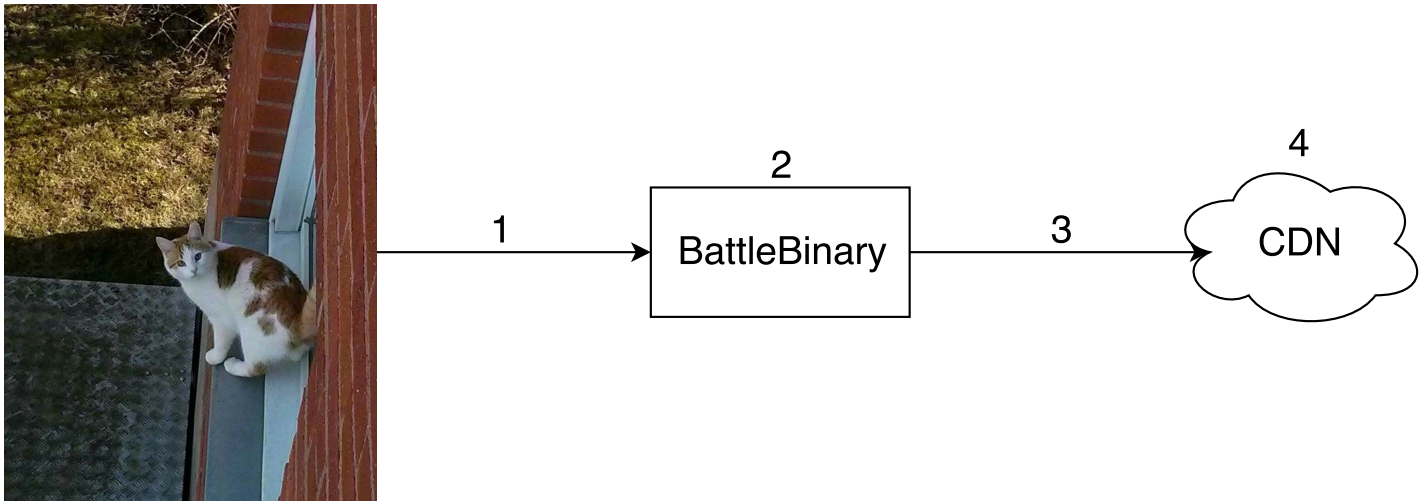
BattleBinary



- 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

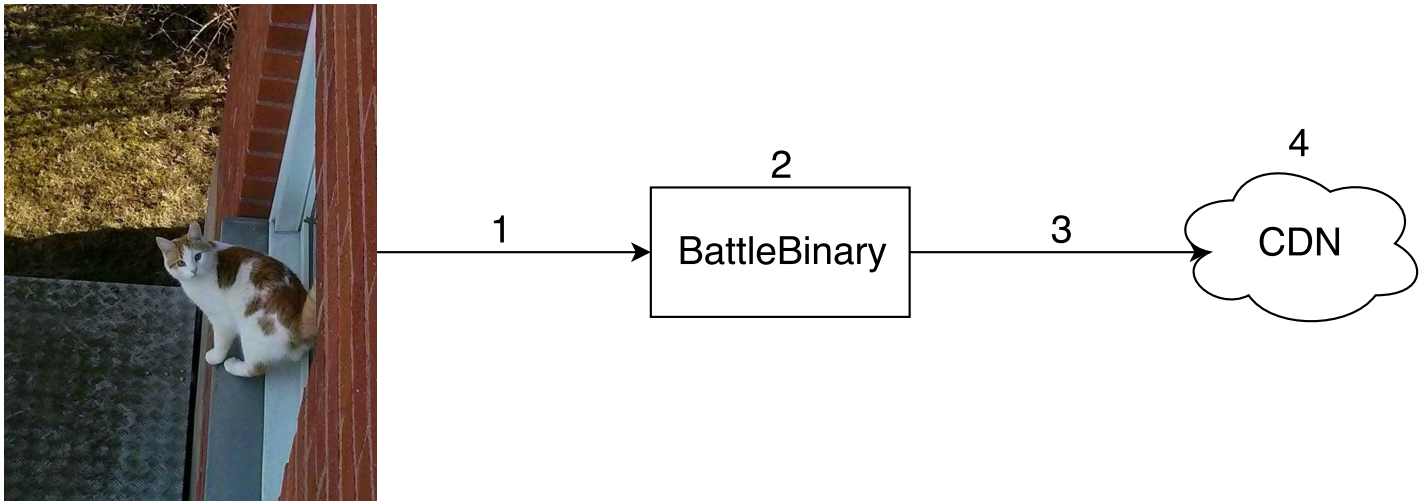
`http://ea.akamaihd.net/cat-f1ee0283b6accd6.jpg`

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

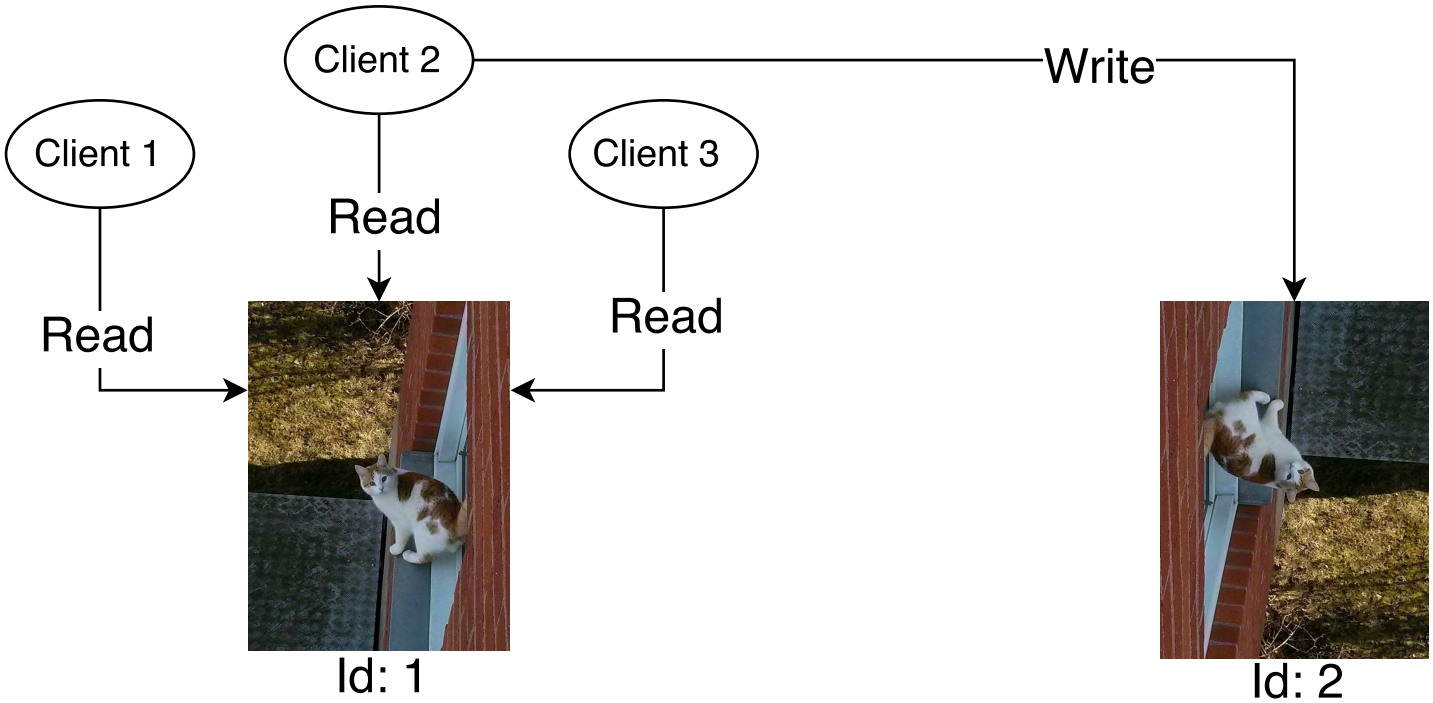
- Insecure
- Impractical
- wont scale

UPPSALA
UNIVERSITET

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

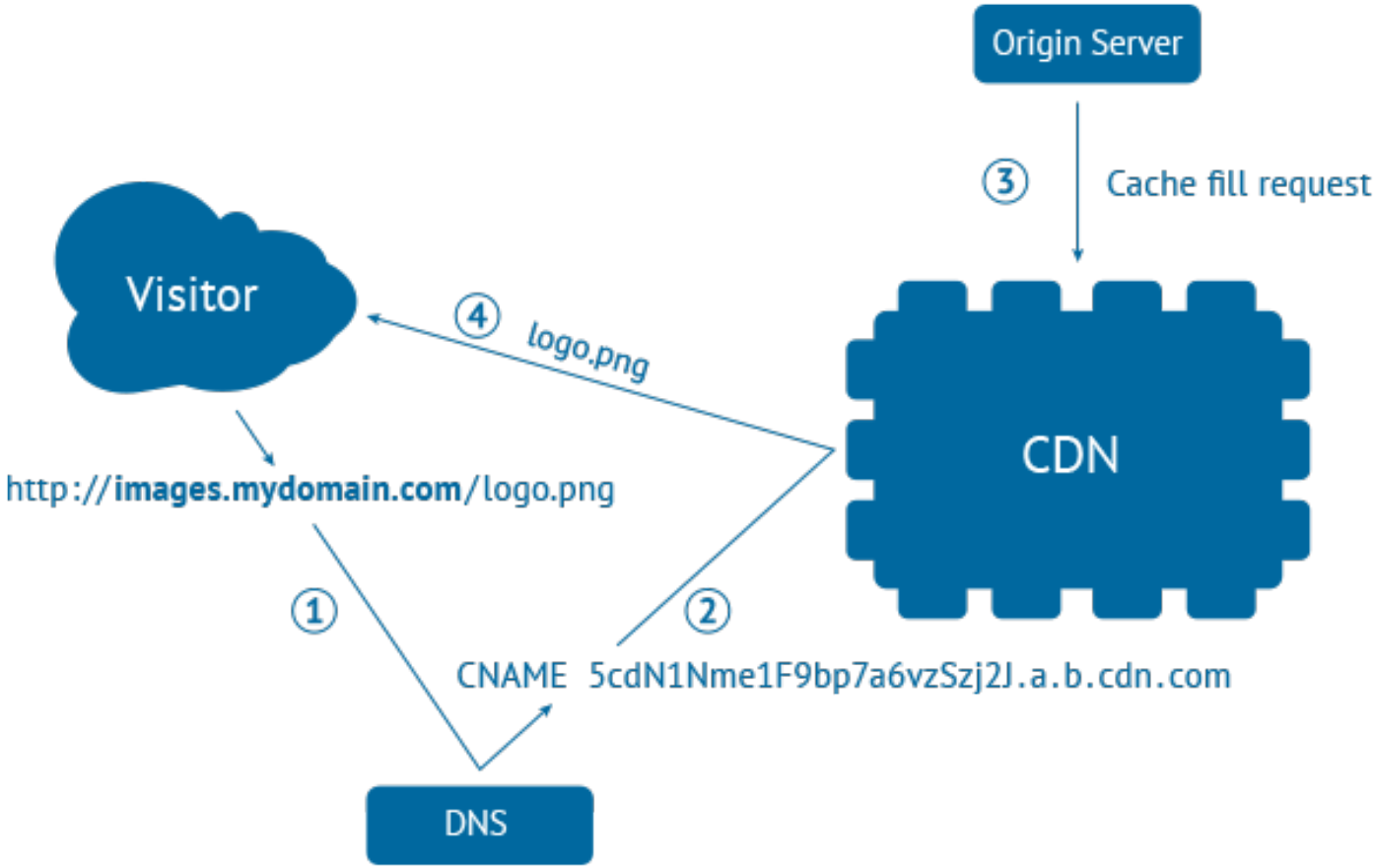
- Mach kernel
- Btrfs
- Programming Languages

UPPSALA
UNIVERSITET

Related Work

- Mach kernel
- Btrfs
- Programming Languages

Solutions Idea

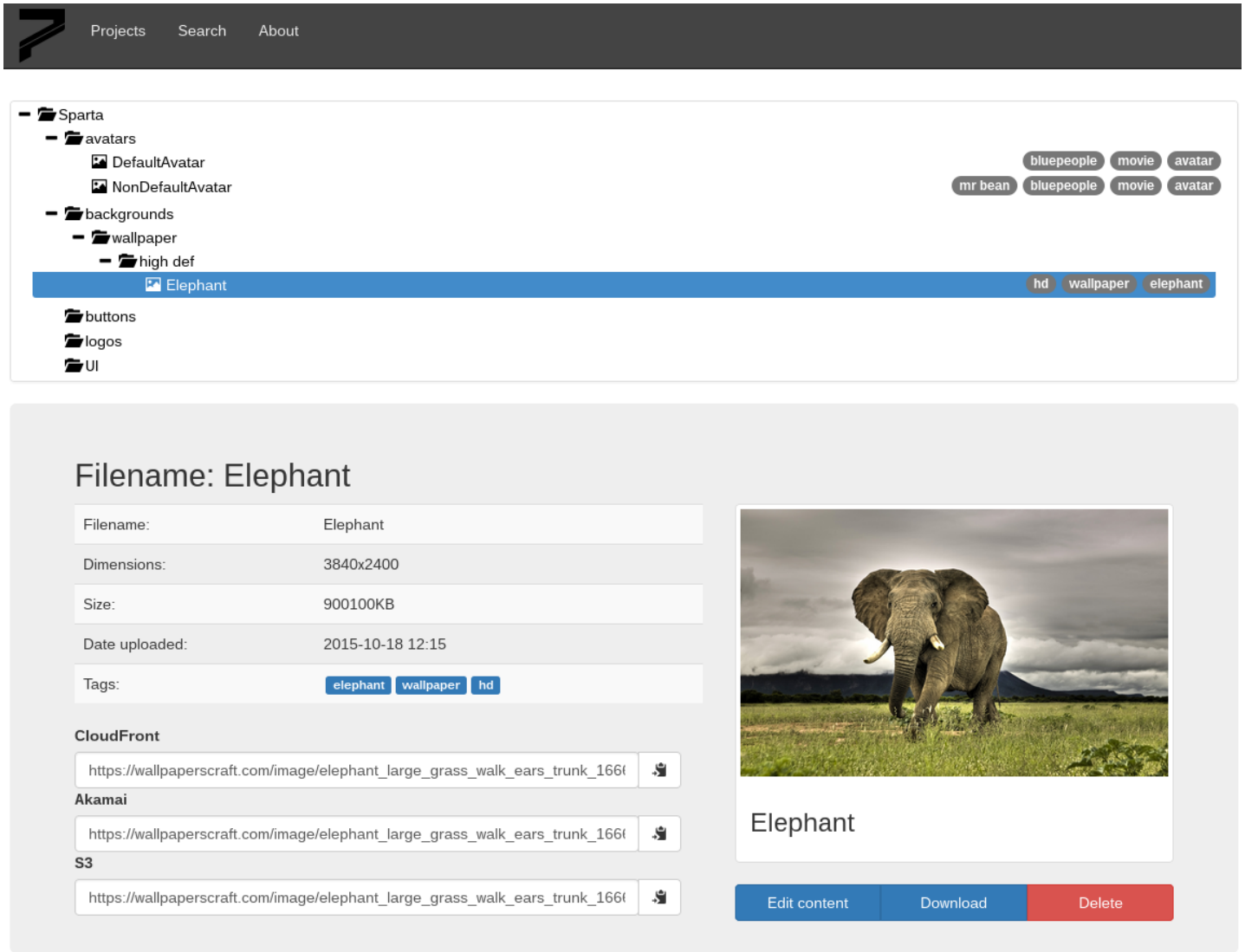


Example Operation

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

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

Virtual Filesystem



Model

- Describes the system that is to be implemented
- Model informally checked through JPF
- Inspired by Biba. et al

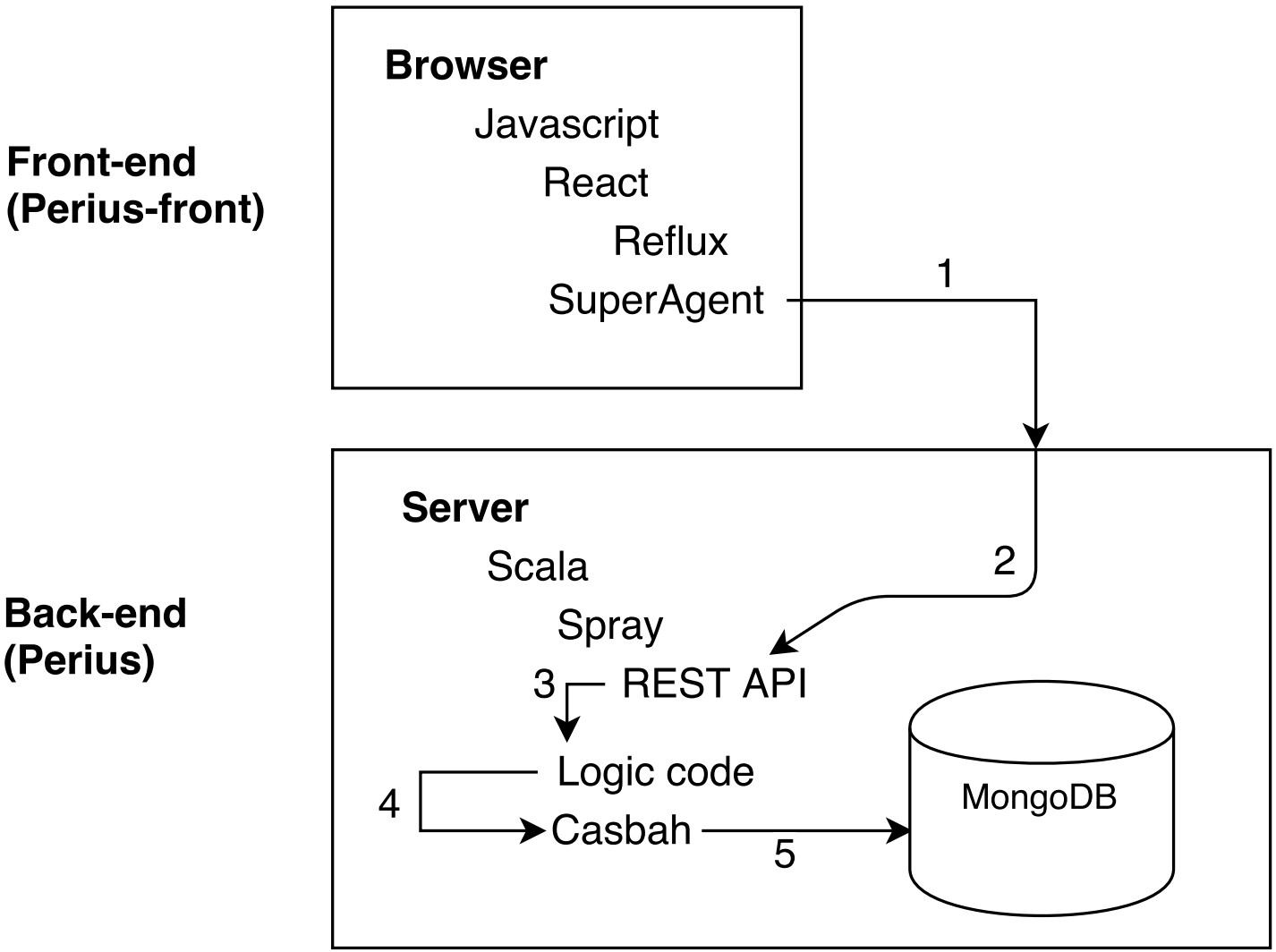
Model

- Describes the system that is to be implemented
- Model informally checked through JPF
- Inspired by Biba. et al

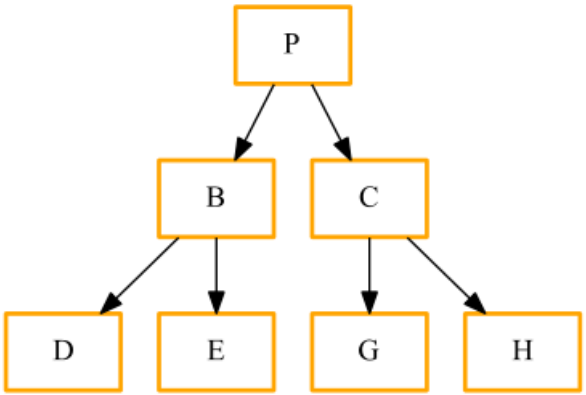
Model

- Describes the system that is to be implemented
- Model informally checked through JPF
- Inspired by Biba. et al

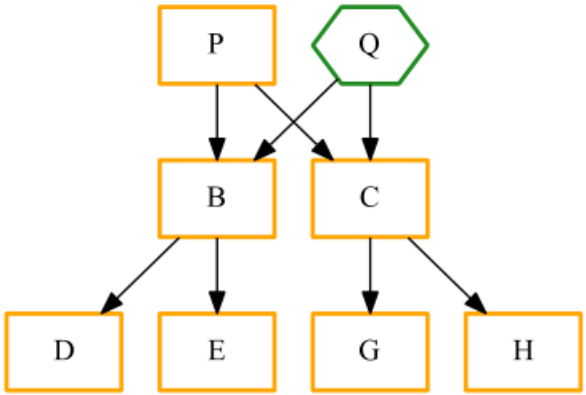
Stack


Front-end
(Perius-front)
Browser
Javascript
React
Reflux
SuperAgent
1
Back-end
(Perius)
Server
Scala
Spray
2
3
REST API
Logic code
4
Casbah
5
MongoDB

Snapshots (and branching)



(a) Tree T_p



(b) T_p cloned to T_q

Scalability

- Unlimited scaling on width
- Optional Non-Blocking I/O with Reactive Mongo
- Some heavy operations can be solved with caching

Scalability

- Unlimited scaling on width
- Optional Non-Blocking I/O with Reactive Mongo
- Some heavy operations can be solved with caching

Scalability

- Unlimited scaling on width
- Optional Non-Blocking I/O with Reactive Mongo
- Some heavy operations can be solved with caching