

The background of the slide features a large, faint watermark of the Uppsala University seal. The seal is circular with a sunburst in the center, surrounded by the Latin text "HIGRATIACADEMIAE" at the top and "VERITAS NATURAE" at the bottom.

# Access Control for Content Distribution Network Assets

Researching Copy-on-Write solutions

---

Lukas Klingsbo <luk18671@student.uu.se>

Department of Information Technology  
Uppsala University

April 20, 2016



# Outline

---

Background

Problem

Perius

## 1 Background

- Uprise
- CDN
- BattleBinary
- Copy-on-Write
- Related Work

## 2 Problem

- Private Content
- Solutions Idea

## 3 Perius

- Model
- Implementation



## Background

Uprise

CDN

BattleBinary

Copy-on-Write

Related Work

## Problem

## Perius



GHOST



# Content Distribution Network (CDN)

## Background

Uprise

CDN

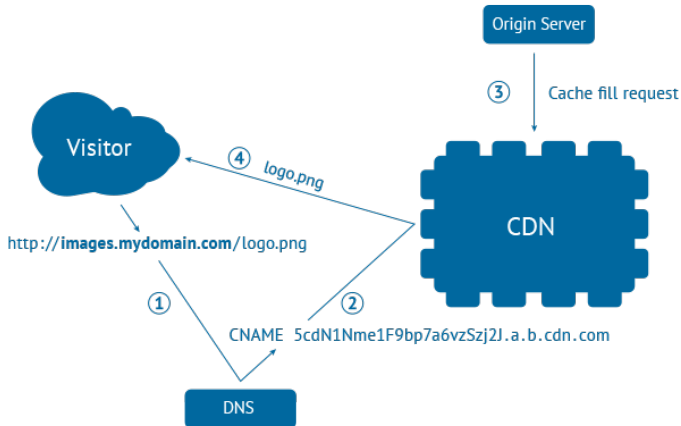
BattleBinary

Copy-on-Write

Related Work

## Problem

## Perius





# BattleBinary

## Virtual file system to organise CDN assets

### Background

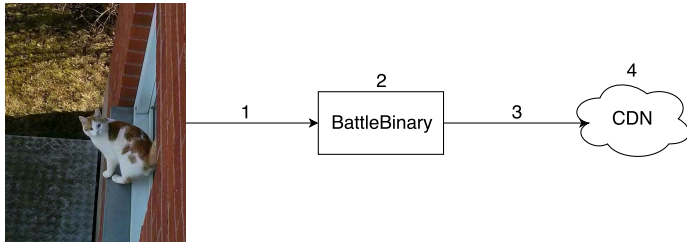
Uprise  
CDN

BattleBinary

Copy-on-Write  
Related Work

### Problem

### Perius



- 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

## Virtual file system to organise CDN assets

### Background

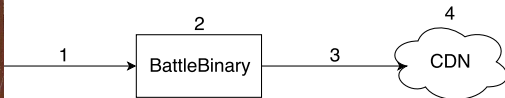
Uprise  
CDN

BattleBinary

Copy-on-Write  
Related Work

### Problem

### Perius



- 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

## Virtual file system to organise CDN assets

### Background

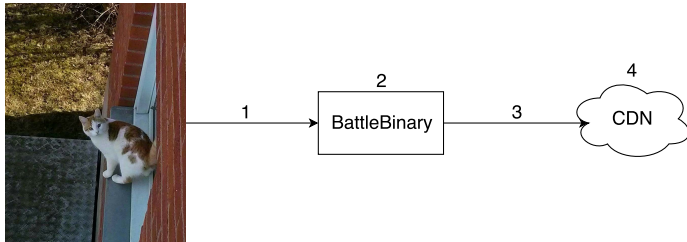
Uprise  
CDN

BattleBinary

Copy-on-Write  
Related Work

### Problem

### Perius



- 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

## Virtual file system to organise CDN assets

### Background

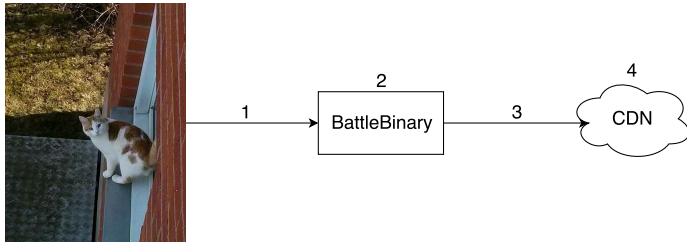
Uprise  
CDN

BattleBinary

Copy-on-Write  
Related Work

### Problem

### Perius



- 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

---

## Background

Uprise  
CDN

BattleBinary

Copy-on-Write  
Related Work

## Problem

## Perius

- A management system for CDN assets

What Uprise wanted:

- Possibility to handle private assets
- Migration to their current technology stack
- Features like snapshots and branching



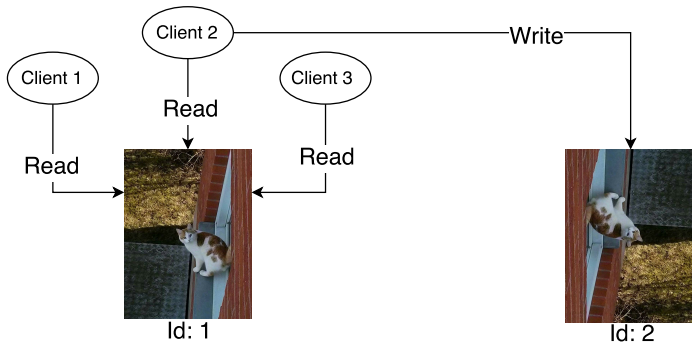
# Copy-on-Write

## Background

Uprise  
CDN  
BattleBinary  
Copy-on-Write  
Related Work

## Problem

## Perius



\*IDs are GUIDs in real implementations



# Effects of COW

---

## Background

Uprise

CDN

BattleBinary

Copy-on-Write

Related Work

## Problem

## Perius

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



# Effects of COW

---

## Background

Uprise

CDN

BattleBinary

Copy-on-Write

Related Work

## Problem

## Perius

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



# Effects of COW

---

## Background

Uprise

CDN

BattleBinary

Copy-on-Write

Related Work

## Problem

## Perius

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



# Effects of COW

---

## Background

Uprise

CDN

BattleBinary

Copy-on-Write

Related Work

## Problem

## Perius

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



# Garbage

## Background

Uprise

CDN

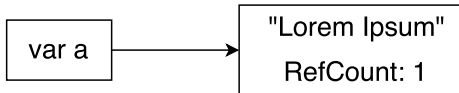
BattleBinary

Copy-on-Write

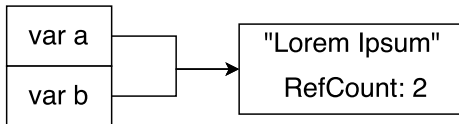
Related Work

## Problem

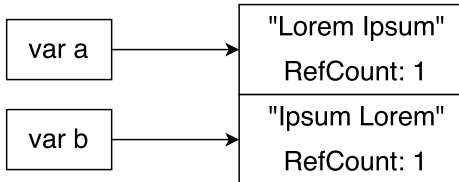
## Perius



**var b = a**



**b = "Ipsum Lorem"**





# Related Work

---

## Background

Uprise  
CDN  
BattleBinary  
Copy-on-Write  
Related Work

## Problem

## Perius

- Mach kernel
- Btrfs
- Programming Languages





# Related Work

---

## Background

Uprise  
CDN  
BattleBinary  
Copy-on-Write  
Related Work

## Problem

## Perius

- Mach kernel
- Btrfs
- Programming Languages



# Related Work

---

## Background

Uprise

CDN

BattleBinary

Copy-on-Write

Related Work

## Problem

## Perius

- Mach kernel
- Btrfs
- Programming Languages



# Problem

---

## Background

## Problem

Private Content  
Solutions Idea

## Perius

- Insecure
- Impractical
- Wont scale
- Lacks necessary features like access control and snapshots



# Problem

---

## Background

## Problem

Private Content  
Solutions Idea

## Perius

- Insecure
- Impractical
- Wont scale
- Lacks necessary features like access control and snapshots



# Problem

---

## Background

## Problem

Private Content  
Solutions Idea

## Perius

- Insecure
- Impractical
- Wont scale
- Lacks necessary features like access control and snapshots



# Problem

---

## Background

## Problem

Private Content  
Solutions Idea

## Perius

- Insecure
- Impractical
- Wont scale
- Lacks necessary features like access control and snapshots



# Private Content

---

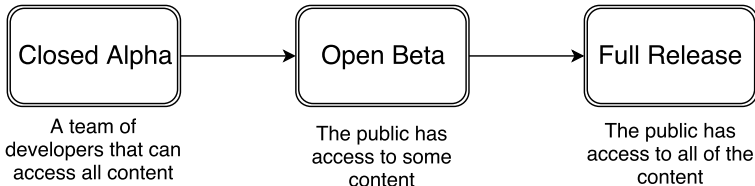
Background

Problem

Private Content

Solutions Idea

Perius





# Solutions Idea

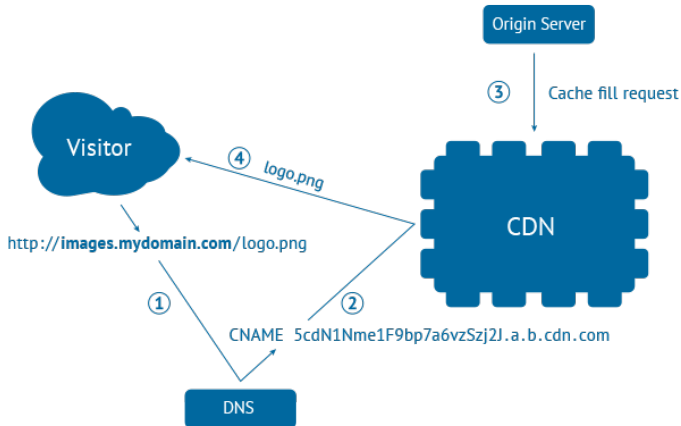
## Background

## Problem

Private Content

Solutions Idea

## Perius







# Example Operation

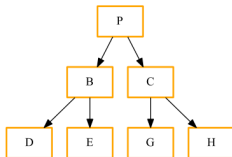
## Background

## Problem

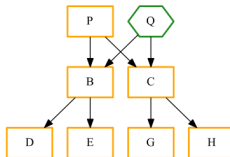
Private Content

Solutions Idea

## Perius



(a) Tree  $T_p$



(b)  $T_p$  cloned to  $T_q$

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




# Virtual Filesystem


Background


Problem

Perius

Model  
Implementation

 Projects Search About






GameName

Assets

Secret

 krafta.jpg

Winter art future

Videos

Images

Cat.jpg

animal cat




logo.jpg


logo

Filename: krafta.jpg

Filename:	krafta.jpg
Dimensions:	1920x1080
Size:	264.57 KB
Last modified:	07/04/2016, 14:43:46
Tags:	<div>future art winter</div>

CloudFront

<a href="http://perius.se:81/projects/5706554b98b798008105fd5a/files/5706560298b79">http://perius.se:81/projects/5706554b98b798008105fd5a/files/5706560298b79</a>	
Akamai	
<a href="http://perius.se:81/projects/5706554b98b798008105fd5a/files/5706560298b79">http://perius.se:81/projects/5706554b98b798008105fd5a/files/5706560298b79</a>	
S3	
<a href="http://perius.se:81/projects/5706554b98b798008105fd5a/files/5706560298b79">http://perius.se:81/projects/5706554b98b798008105fd5a/files/5706560298b79</a>	



krafta.jpg

Edit content

Download

Delete



# Model

---

Background

Problem

Perius

Model

Implementation

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



# Model

---

Background

Problem

Perius

Model

Implementation

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



# Model

---

Background

Problem

Perius

Model

Implementation

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



# Model

---

Background

Problem

Perius

Model

Implementation

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



# Model - Sets

---

Background

Problem

Perius

Model

Implementation

Set	Name	Relation	Type
<b>C</b>	Containers	$\mathbf{C} \supseteq C \ni c$	$\mathbf{C} = P_{fin}(P_{fin}(C \cup M))$
<b>M</b>	Content	$\mathbf{M} \supseteq M \ni m$	$\mathbf{M} = P_{fin}(M)$
<b>F</b>	Files	$\mathbf{F} \supseteq F \ni f$	$\mathbf{F} = P_{fin}(F)$



# Model - File Creation

---

Background

Problem

Perius

Model

Implementation

## Rule of Inference

$$\frac{C \ni c \quad A[u, c] \quad m \notin c \quad \neg \text{readOnly}(c)}{(C \cup \{c\}, F) \xrightarrow{u, \text{create}(m, c)} (C \cup \{c \cup \{m\}\}, F \cup \{m.\text{file}\})} \quad (1)$$





# Persistent Storage

---

Background

Problem

Perius

Model

Implementation



`{ name: mongo, type: DB }`

- Unaware of Copy-on-Write
- JSON (sort of) documents stored instead of tables



# Stack

Background

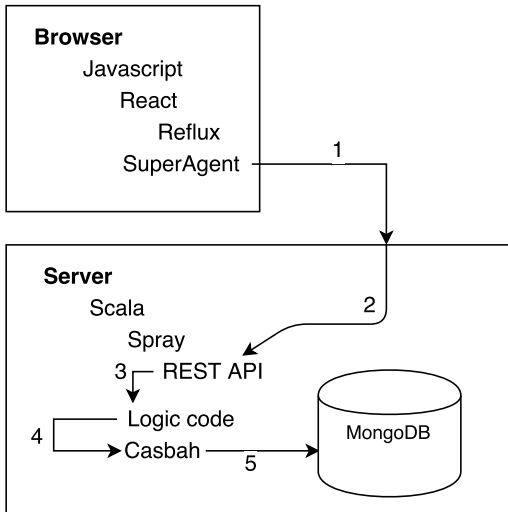
Problem

Perius

Model

Implementation

**Front-end  
(Perius-front)**





# Security Settings

---

Background

Problem

Perius

Model

Implementation

- **Public** - The content can be reached by anybody
- **Protected** - The content can only be reached by a range of IP addresses
- **Private** - The content can only be reached by users with a signed cookie



# Security Settings

---

Background

Problem

Perius

Model

Implementation

- **Public** - The content can be reached by anybody
- **Protected** - The content can only be reached by a range of IP addresses
- **Private** - The content can only be reached by users with a signed cookie



# Security Settings

---

Background

Problem

Perius

Model

Implementation

- **Public** - The content can be reached by anybody
- **Protected** - The content can only be reached by a range of IP addresses
- **Private** - The content can only be reached by users with a signed cookie



# Snapshots (and branching)

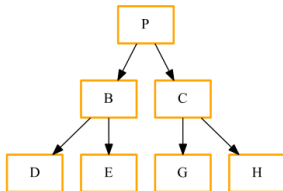
Background

Problem

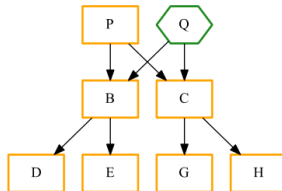
Perius

Model

Implementation



(a) Tree  $T_p$



(b)  $T_p$  cloned to  $T_q$



# Load Testing

---

Background

Problem

Perius

Model

Implementation

- **Rigorous Load Testing was done on the back-end**
- Wrk and Apache Bench was used
- Result: About 60000 clients/back-end node before congestion



# Load Testing

---

Background

Problem

Perius

Model

Implementation

- Rigorous Load Testing was done on the back-end
- Wrk and Apache Bench was used
- Result: About 60000 clients/back-end node before congestion





# Load Testing

---

Background

Problem

Perius

Model

Implementation

- Rigorous Load Testing was done on the back-end
- Wrk and Apache Bench was used
- Result: About 60000 clients/back-end node before congestion



# Scalability

---

Background

Problem

Perius

Model

Implementation

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



# Scalability

---

Background

Problem

Perius

Model

Implementation

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



# Scalability

---

Background

Problem

Perius

Model

Implementation

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



# Open Source

---

Background

Problem

Perius

Model

Implementation

- `https://github.com/spydon/perius`
- `https://github.com/spydon/perius-front`



Background

Problem

Perius

Model

Implementation

Questions?



Thank you for listening