Decentralized Internet Archive
using the Cothority framework

Nicolas PLANCHEREL

Prof. Bryan FORD
Eleftherios KOKORIS KOGIAS
Kirill NIKITIN

Decentralized and Distributed Systems (DEDIS)
School of Computer and Communication Sciences (IC)
École Polytechnique Fédérale de Lausanne (EPFL)

EPFL, Master Thesis oral presentation February 2018

Outline

- Motivation
- Description
- Evaluation And Discussion
- Demo
- Conclusion

MOTIVATION

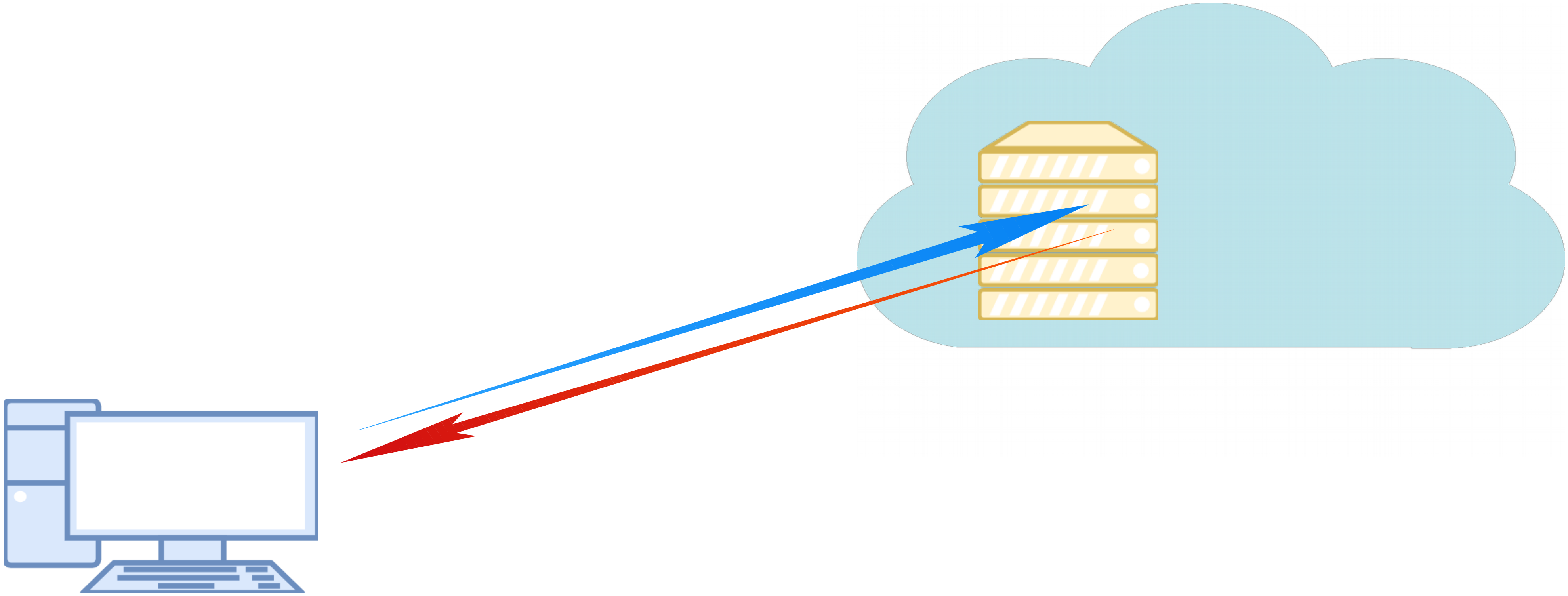
Motivation

Objectives

- Create a censorship resistant internet archive
  - Archiving avoiding tampering or deletion (by one or a small collusion of entity)
  - Store only relevant content
  - Possiblity to check integrity once archived
  - Consider that the censor can try to add, modify or delete data

Motivation

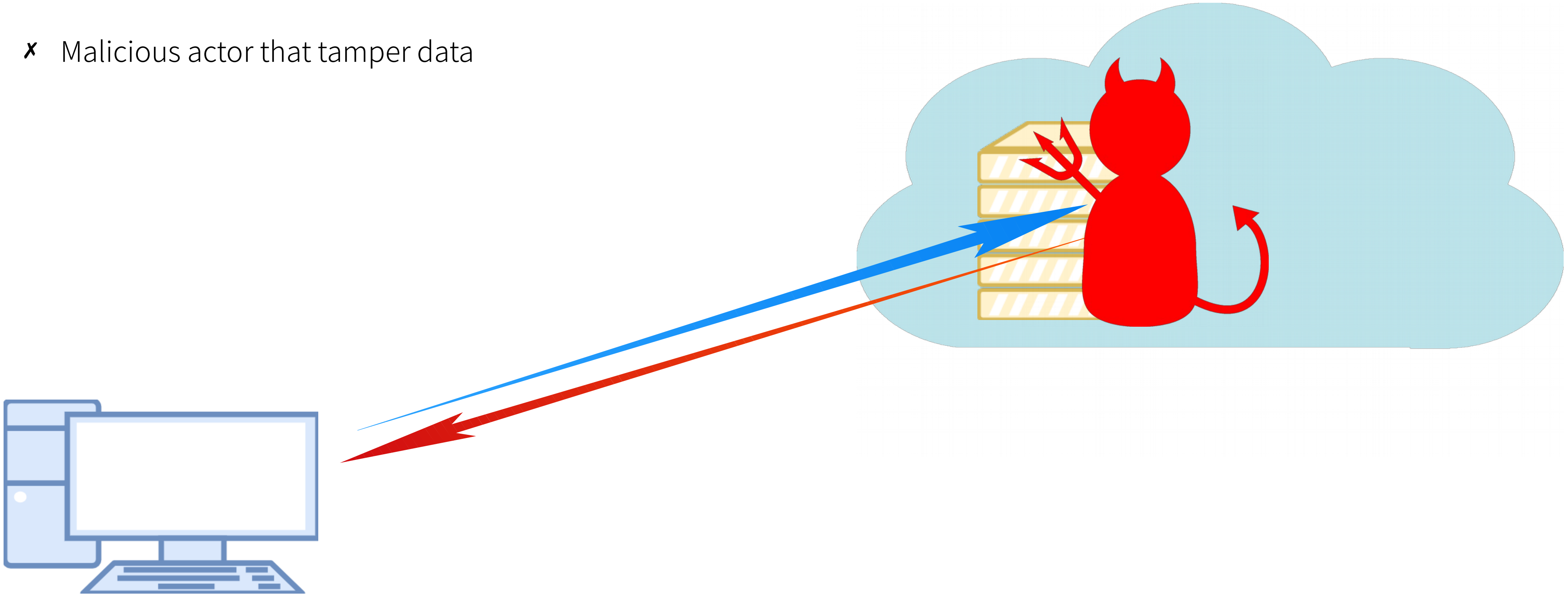
Context - Centralized Internet



Motivation

Context - Centralized Internet

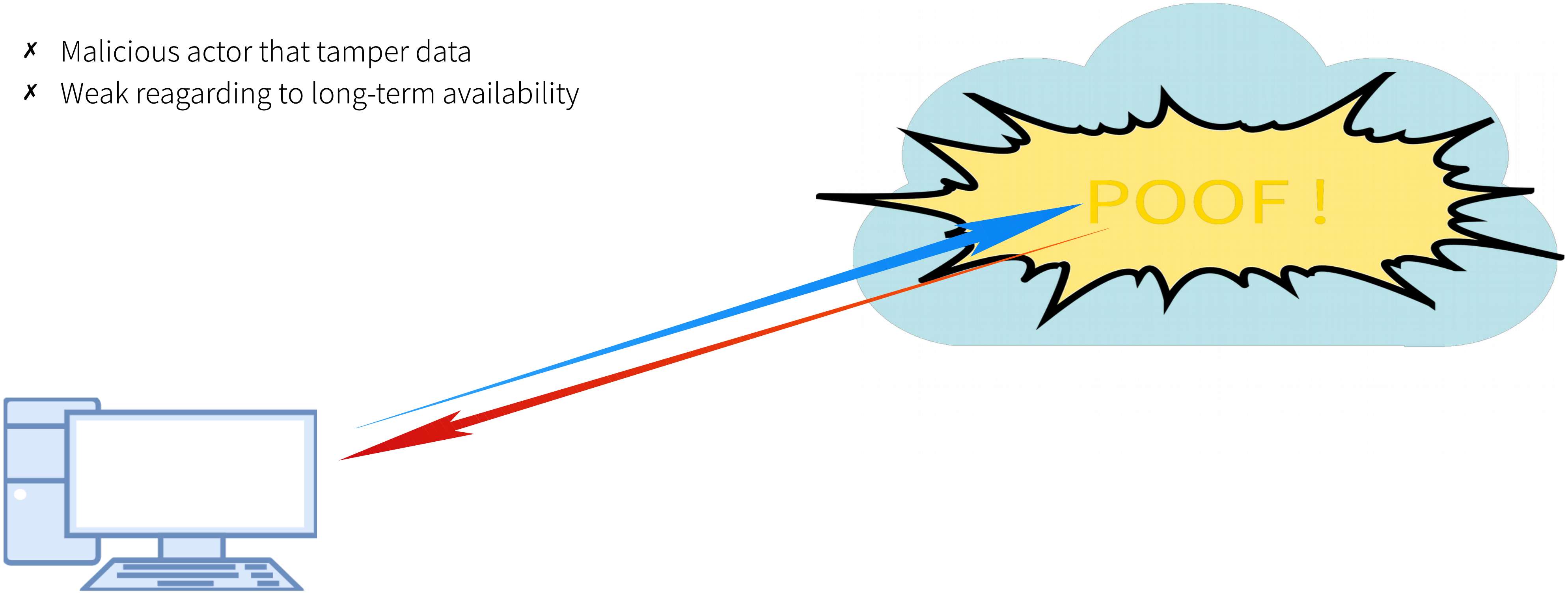
- ✗ Malicious actor that tamper data



Motivation

Context - Centralized Internet

- ✗ Malicious actor that tamper data
- ✗ Weak reagarding to long-term availability

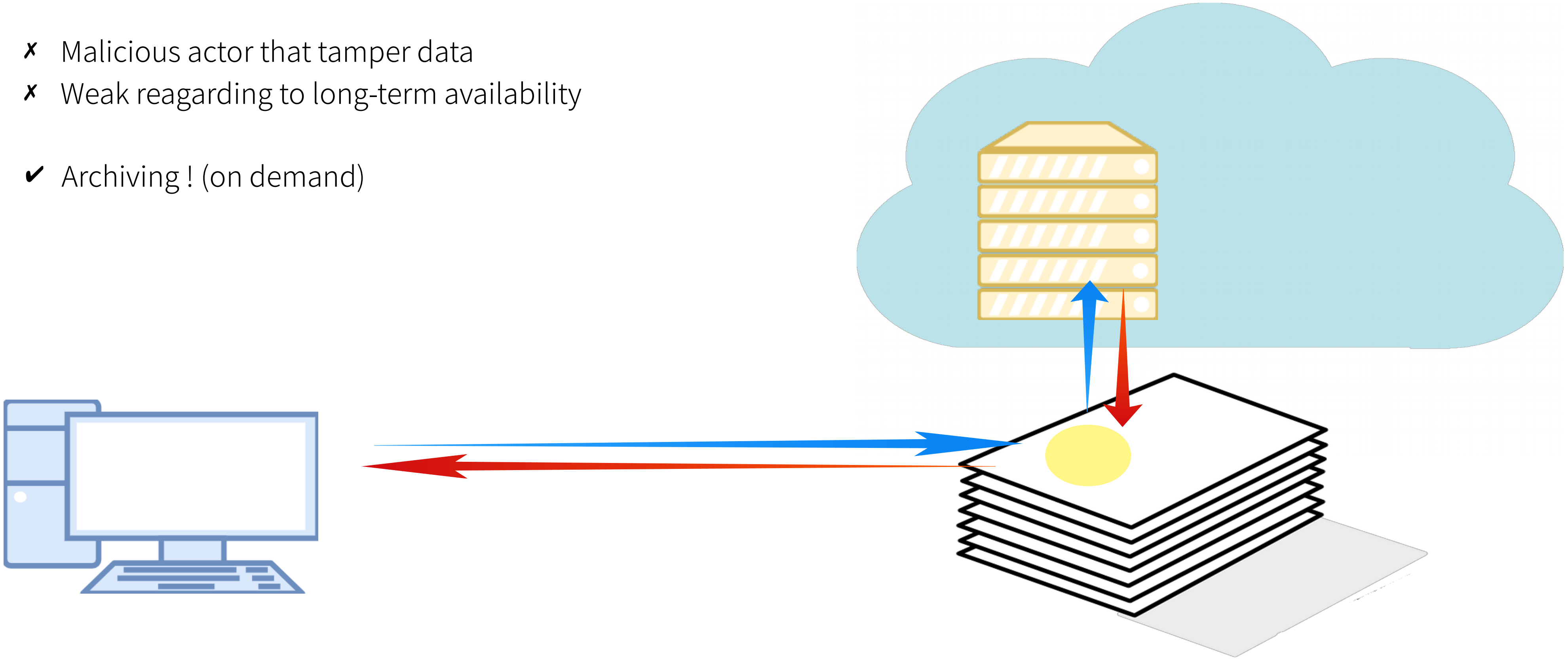


Motivation

Context – Centralized Internet - Archive.org

- ✗ Malicious actor that tamper data
- ✗ Weak reagarding to long-term availability

- ✔ Archiving ! (on demand)

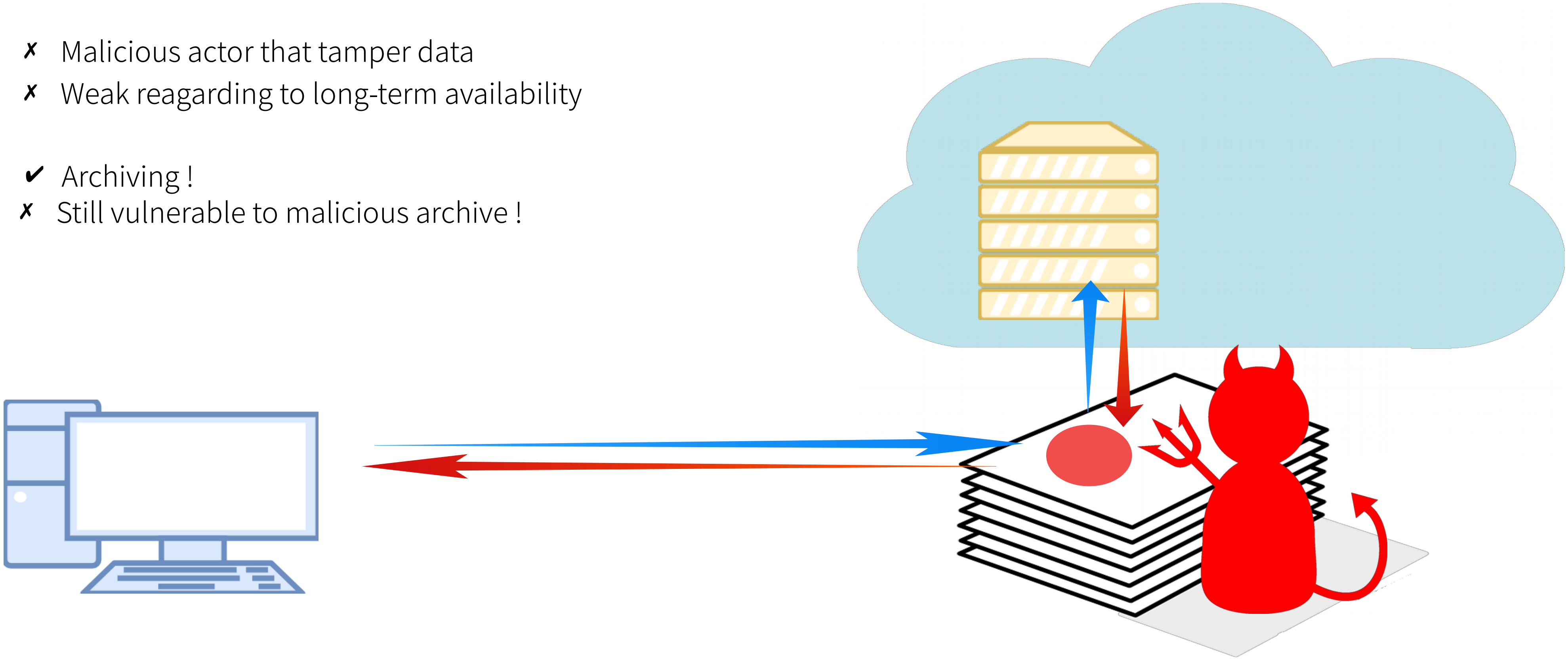


Motivation

Context – Centralized Internet - Archive.org

- ✗ Malicious actor that tamper data
- ✗ Weak reagarding to long-term availability

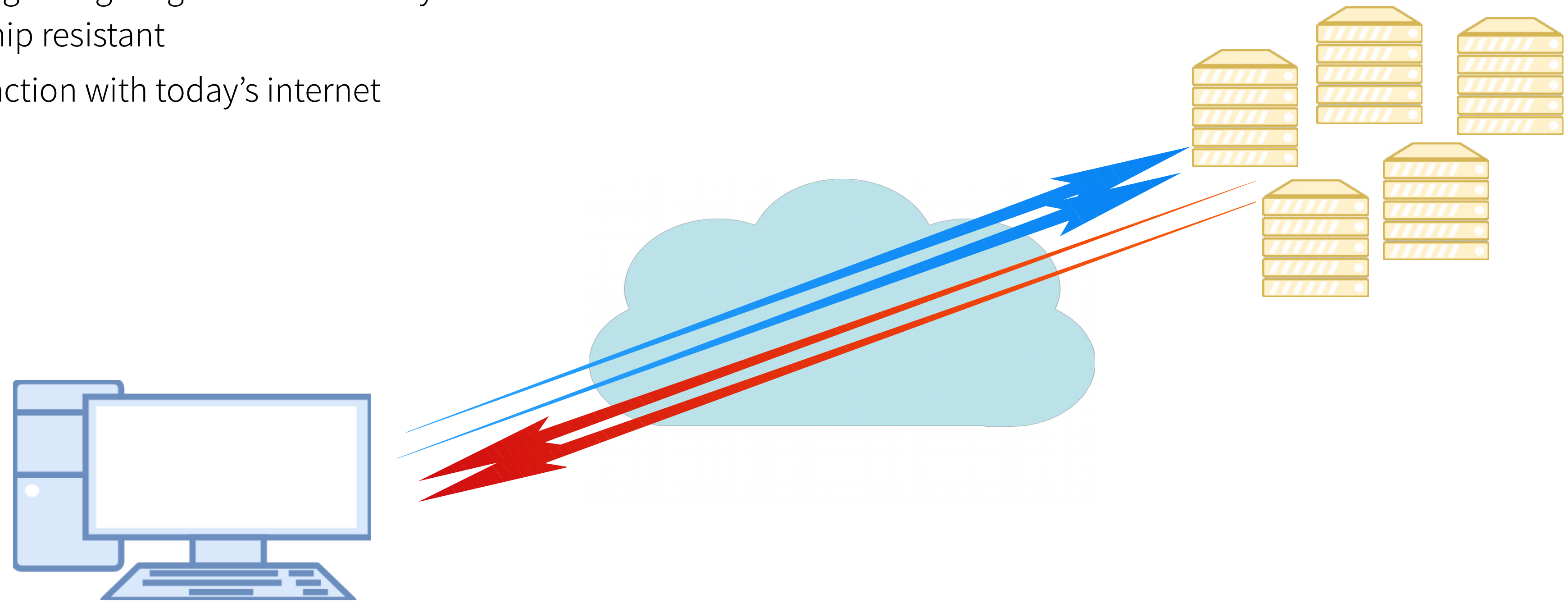
- ✔ Archiving !
- ✗ Still vulnerable to malicious archive !



Motivation

Context - Decentralized Internet – ZeroNet

- ✔ Distributed By Design !
- ✔ Strong regarding long-term avalilability
- ✔ Censorship resistant
- ✗ No interaction with today’s internet



Motivation

Overview

- Centralized Internet is vulnerable to censorship
  - Malicious actor
  - Deletion and Tampering
- Solutions exsits but still have weaknesses
  - Centralized : Archive.org
  - Decentralized : ZeroNet
- So we developed a Decentralized Internet Archive

DESCRIPTION

Description

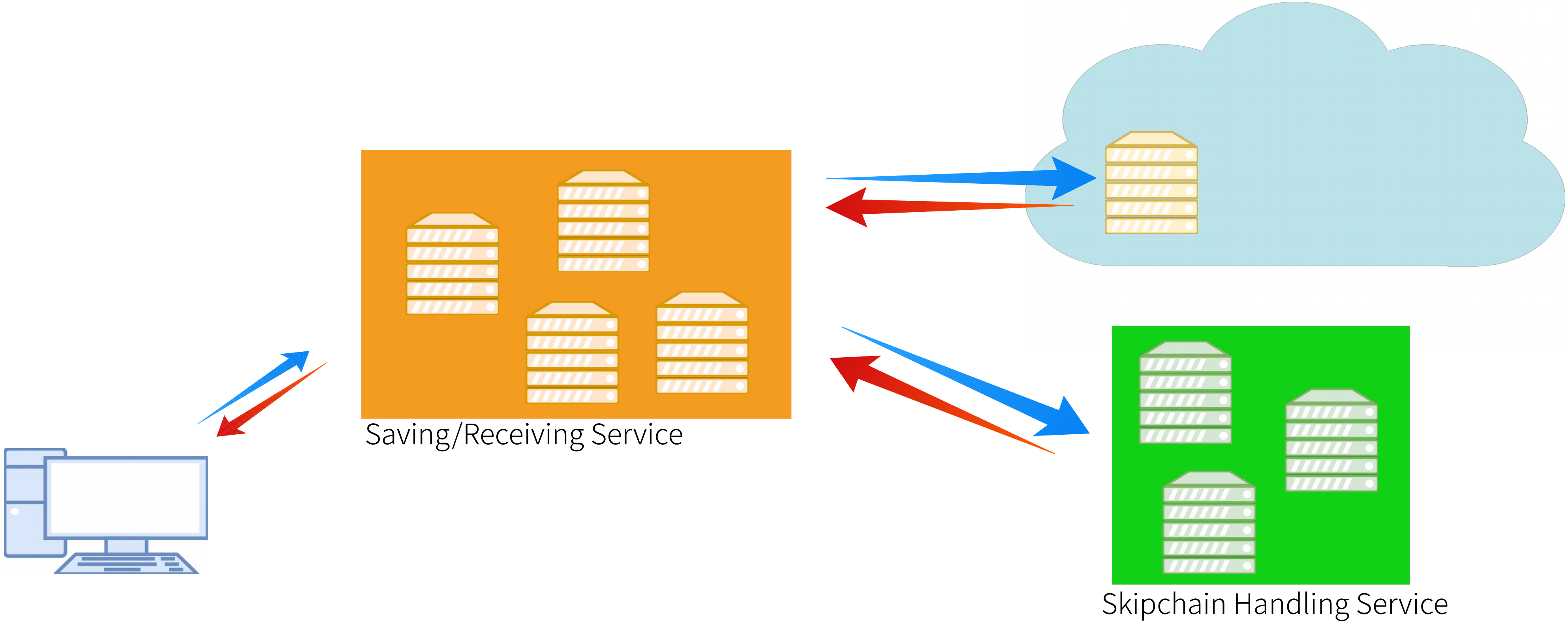
Objectives

- Create a censorship resistant internet archive
  - Avoid Tampering using decentralized storage system : Skipchain
  - Filter content by reaching a consensus on the content of the webpage
  - Using the CoSi Service of the Cothorithy framework (collective signature)
  - Avoid adding malicious data using a trusted reference to make a consensus on

- Operations
  - Save
    - Consensus on the content of the webpage
    - Collectively Sign the common subset of the page
    - Store the signed page on the skipchain
  - Retrieve
    - Get the correct signed page
    - Verify the signature

Description

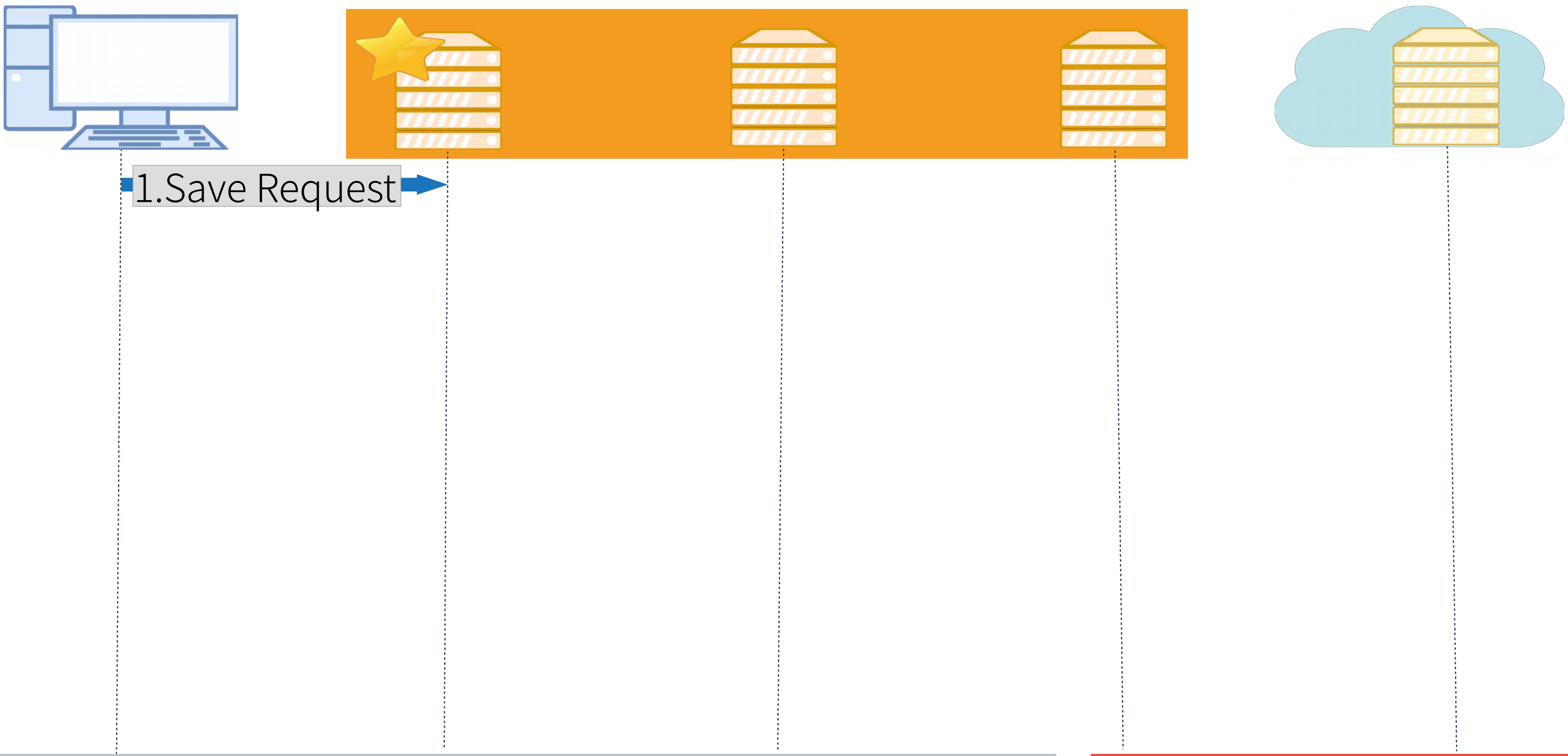
High-Level



Description

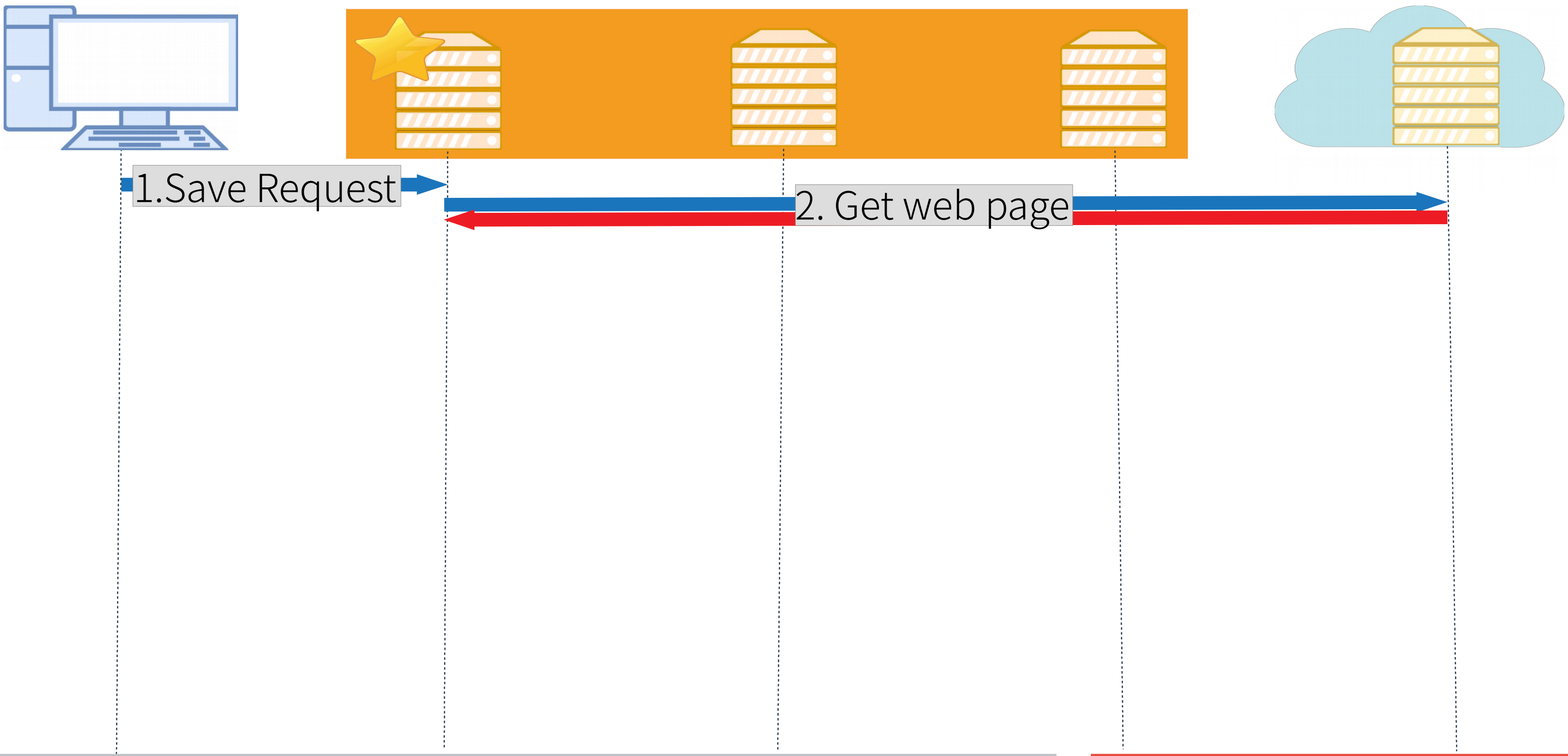
Saving (with a tree-based consenus protocol)

1. Request web page



Description

Saving (with a tree-based consenus protocol)

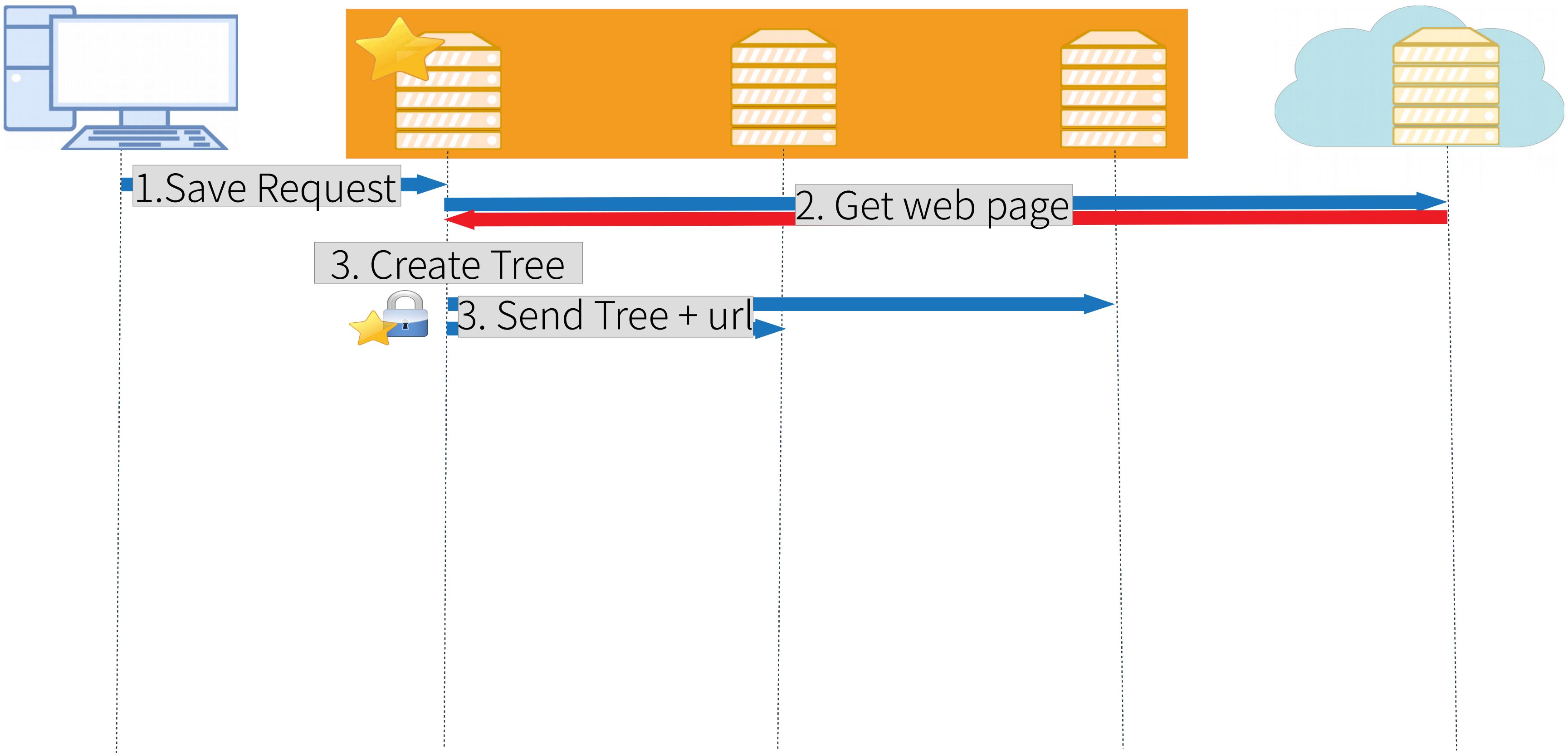


1. Request web page
2. Leader get page

Description

Saving (with a tree-based consenus protocol)

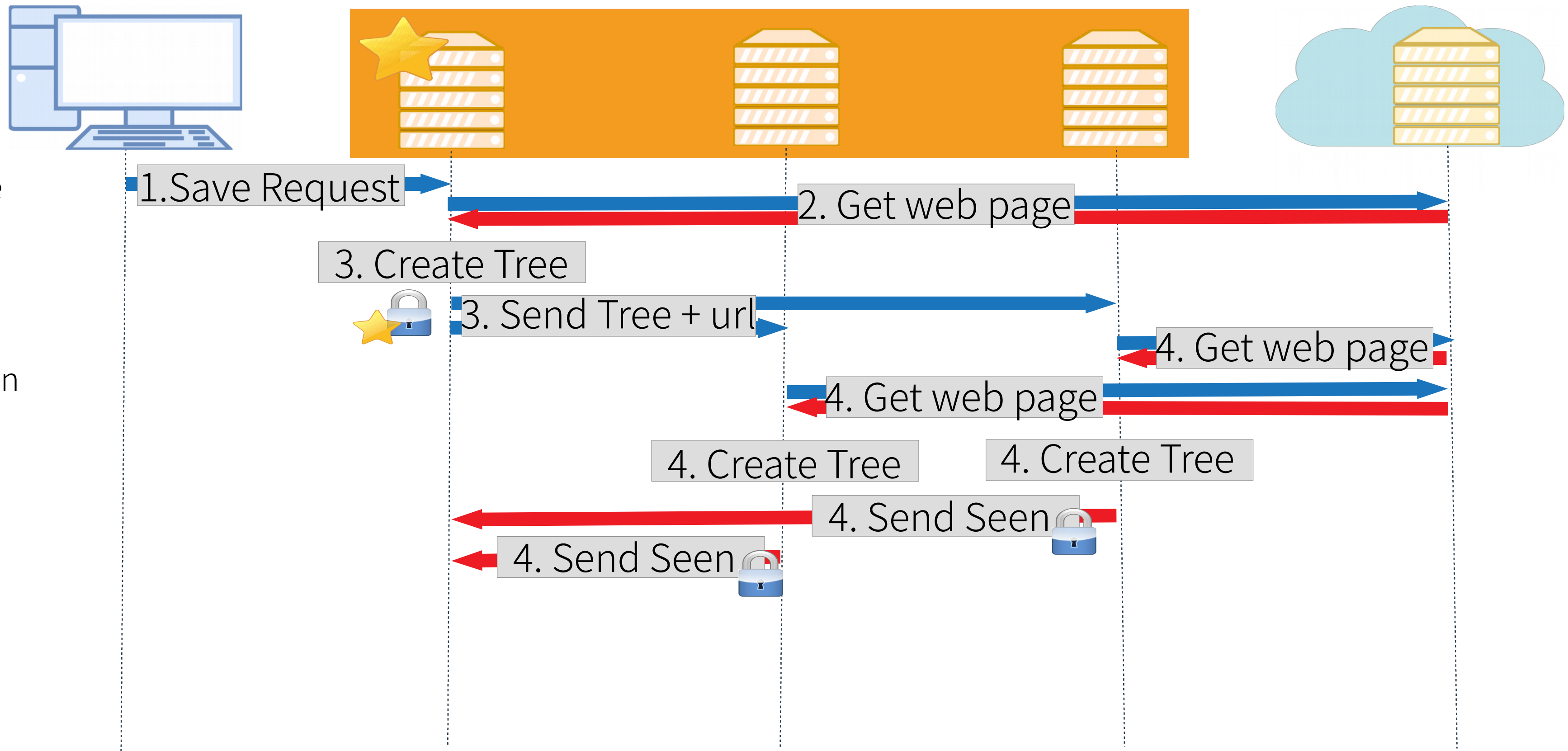
1. Request web page
2. Leader get page
3. Leader transmit url and tree



Description

Saving (with a tree-based consenus protocol)

1. Request web page
2. Leader get page
3. Leader transmit url and tree
4. Every child get own tree and send common parts

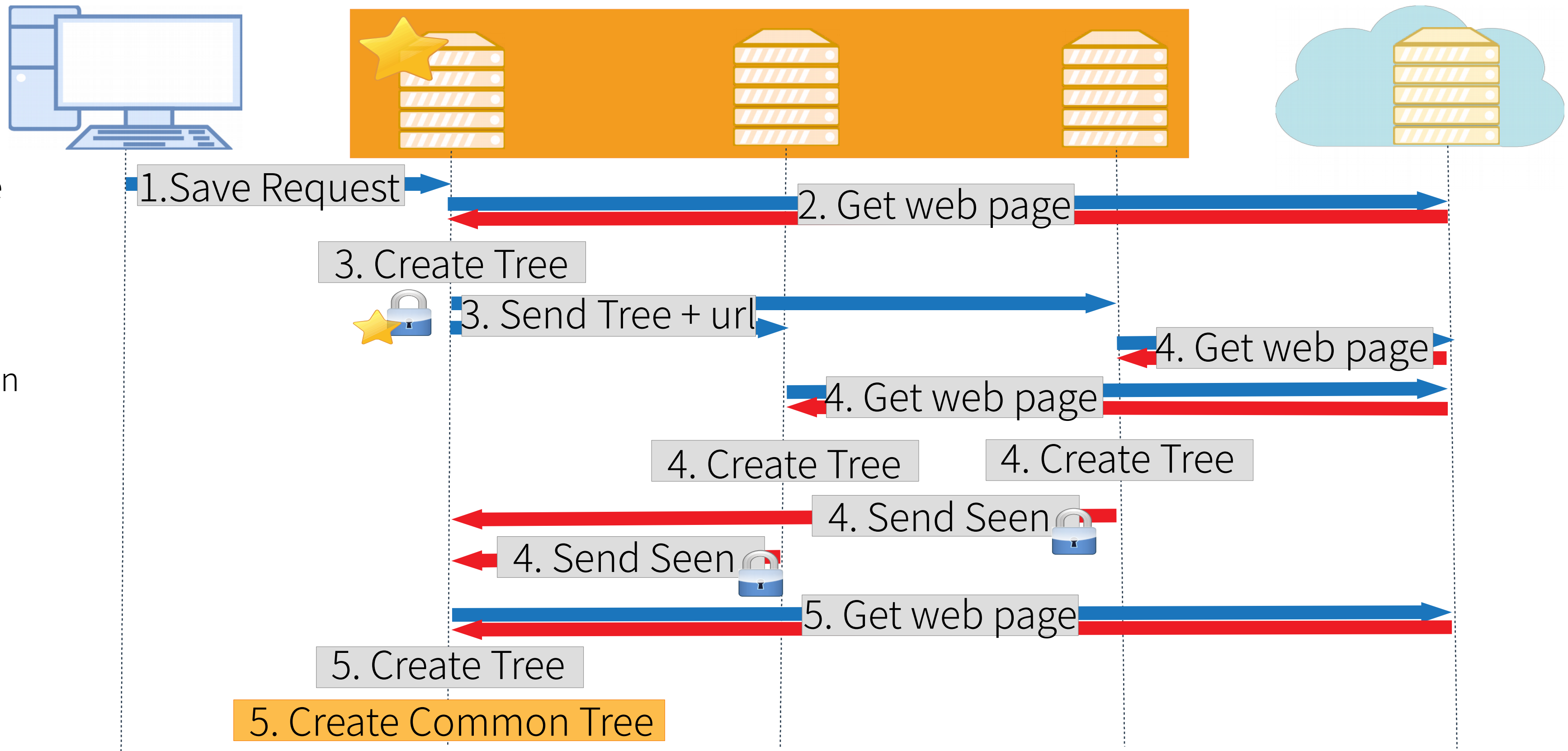


Description

Saving (with a tree-based consenus protocol)

1. Request web page
2. Leader get page
3. Leader transmit url and tree
4. Every child get own tree and send common parts

5. Leader create common tree



Description

Saving – Creating the HTML tree

```html
<!doctype html>
<html lang="en">
  <head>
      <meta charset="UTF-8">
      <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
      <h1>DECENARCH</h1>
  </body>
</html>
```

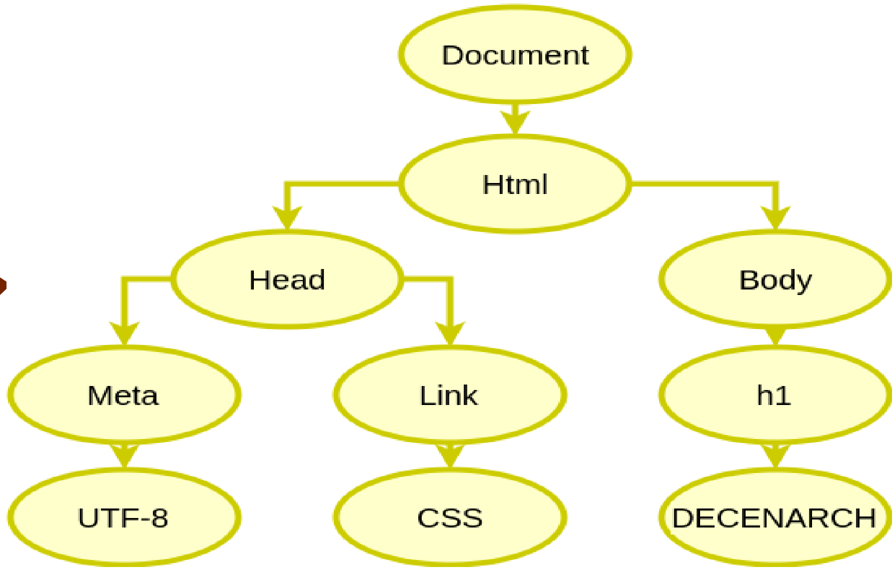
- Get Html Code

- A web page consists of
  - An html code text
  - Additional Data
    - Images
    - CSS file(s)

Description

Saving – Creating the HTML tree

```html
<!doctype html>
<html lang="en">
  <head>
      <meta charset="UTF-8">
      <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
      <h1>DECENARCH</h1>
  </body>
</html>
```

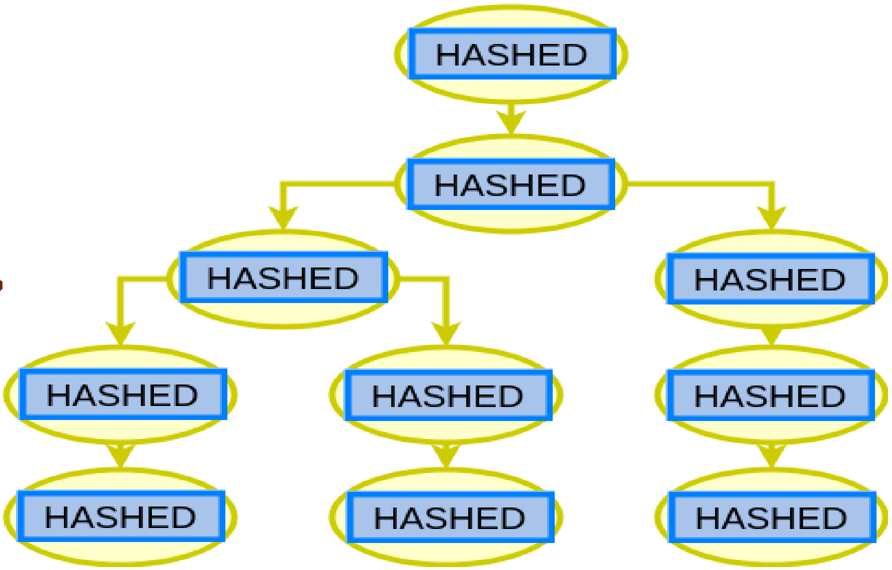


- Get Html Code
- Infer Html Tree from code

Description

Saving – Creating the HTML tree

```html
<!doctype html>
<html lang="en">
  <head>
      <meta charset="UTF-8">
      <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
      <h1>DECENARCH</h1>
  </body>
</html>
```



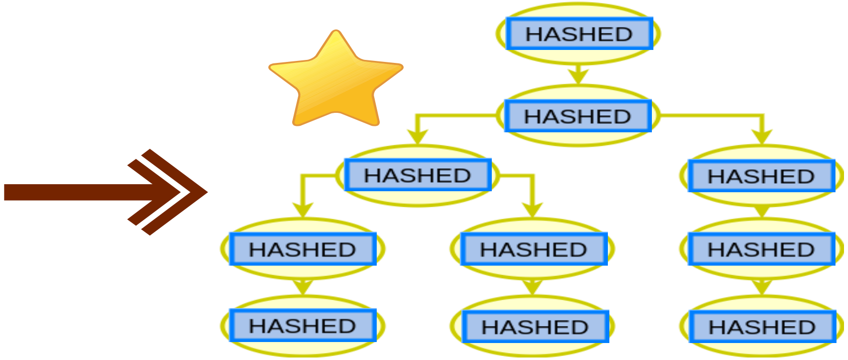
- Get Html Code

- Infer Html Tree from code
- Hash the data of every node individually

Description

Saving – Signing the HTML tree

```html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
      <h1>DECENARCH</h1>
  </body>
</html>
```

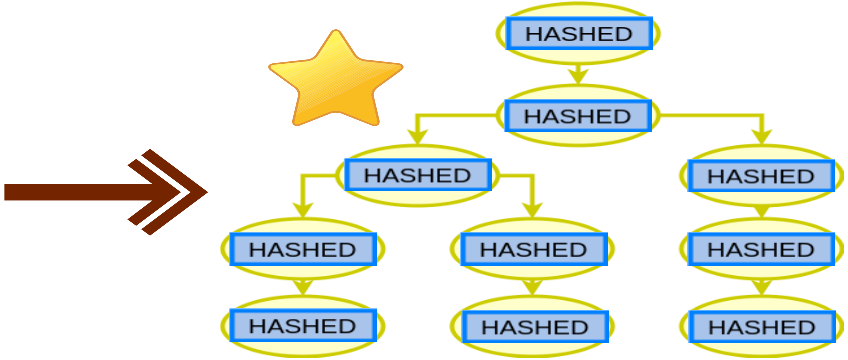


- Leader’s MasterTree

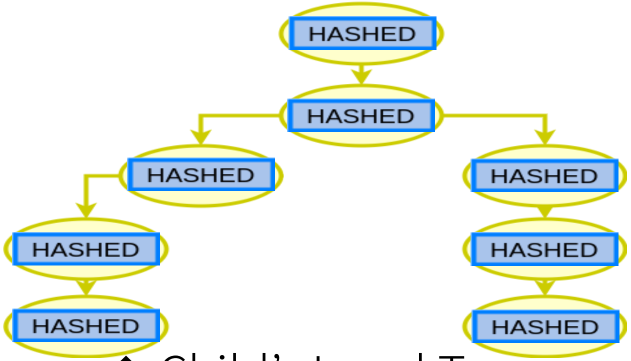
Description

Saving – Signing the HTML tree

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
     <h1>DECENARCH</h1>
  </body>
</html>
```



- Leader’s Master Tree

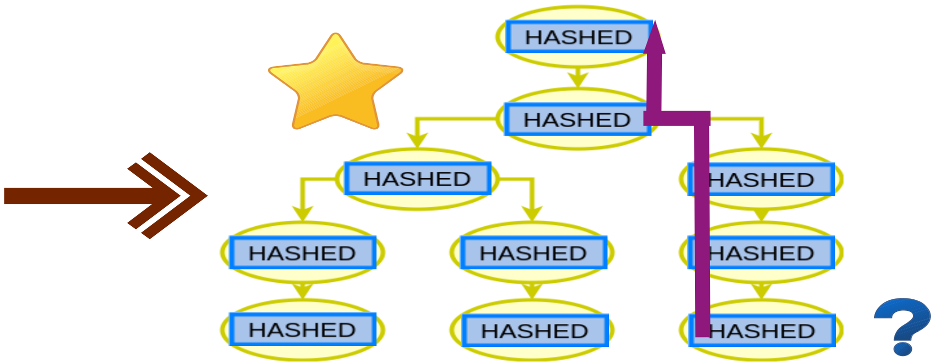


- Child’s Local Tree

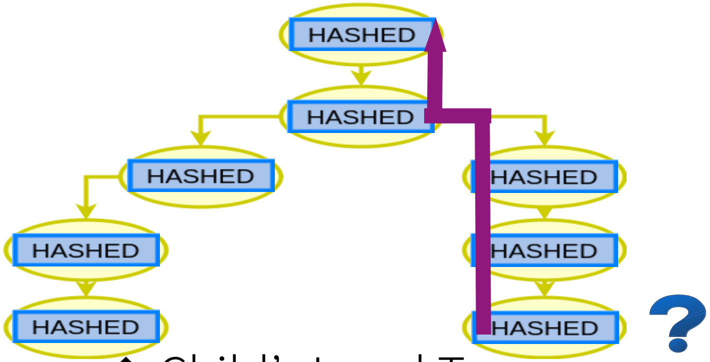
Description

Saving – Signing the HTML tree

```html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
     <h1>DECENARCH</h1>
  </body>
</html>
```



- Leader’s MasterTree

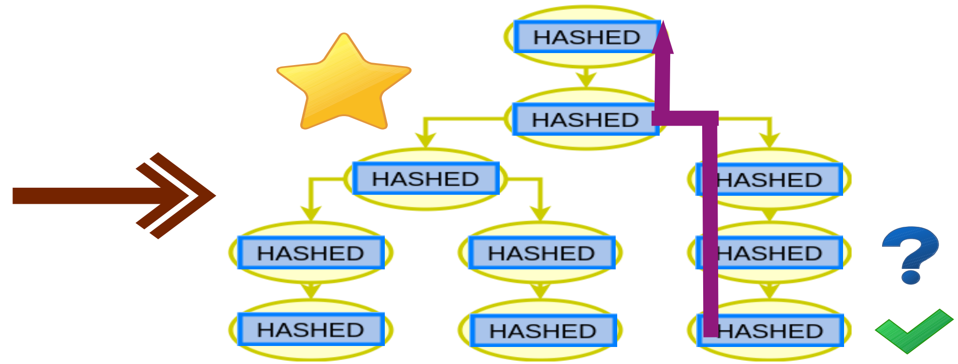


- Child’s Local Tree

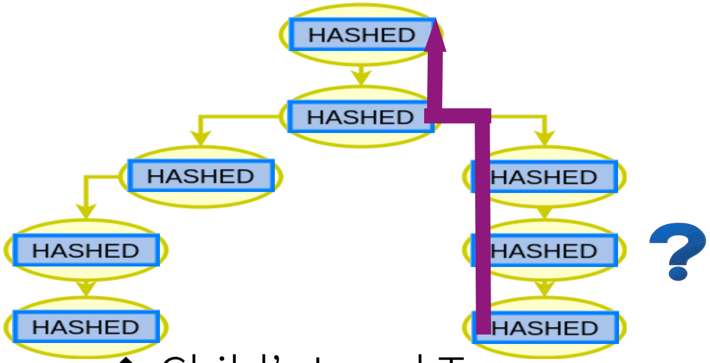
Description

Saving – Signing the HTML tree

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
     <h1>DECENARCH</h1>
  </body>
</html>
```



- Leader’s MasterTree

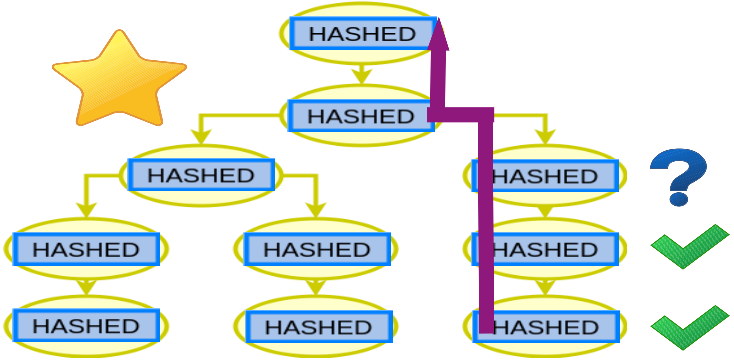


- Child’s Local Tree

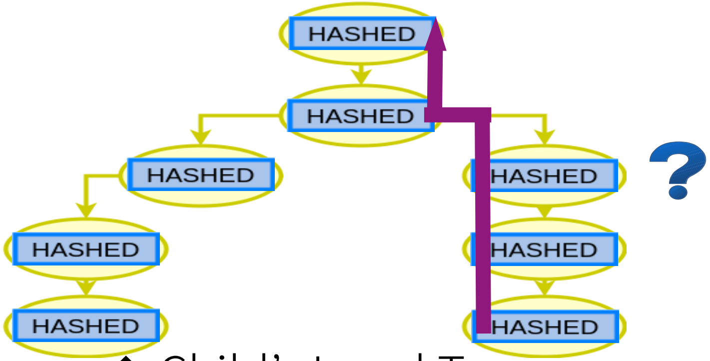
Description

Saving – Signing the HTML tree

```html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
     <h1>DECENARCH</h1>
  </body>
</html>
```



- Leader’s MasterTree

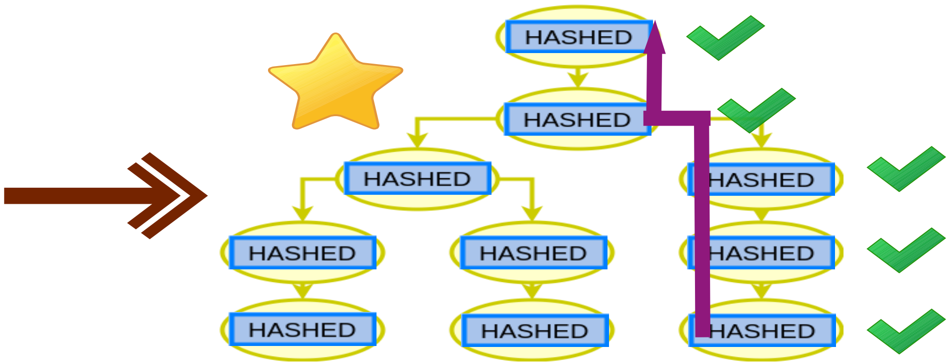


- Child’s Local Tree

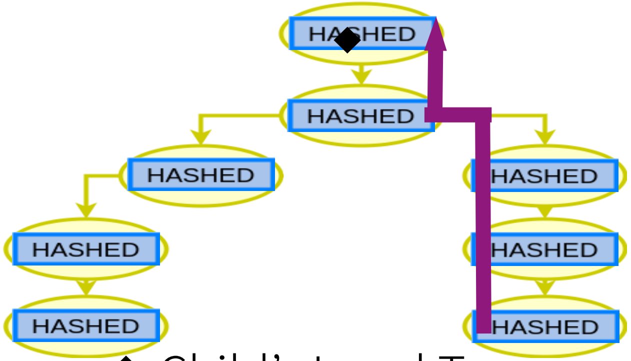
Description

Saving – Signing the HTML tree

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
      <h1>DECENARCH</h1>
  </body>
</html>
```



- Leader’s MasterTree

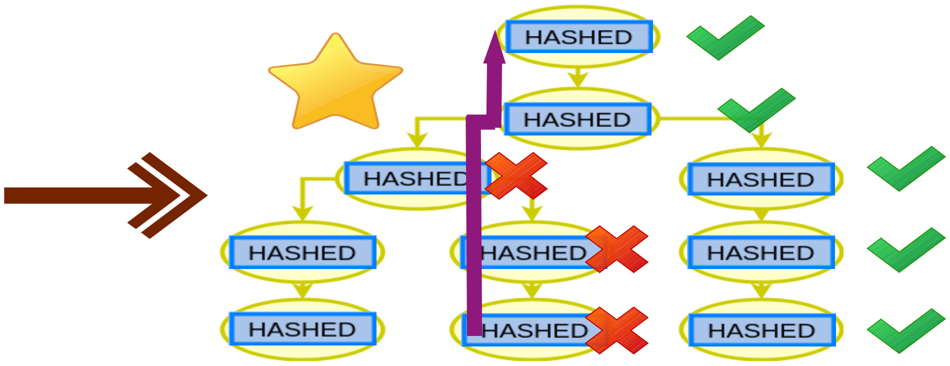


- Child’s Local Tree

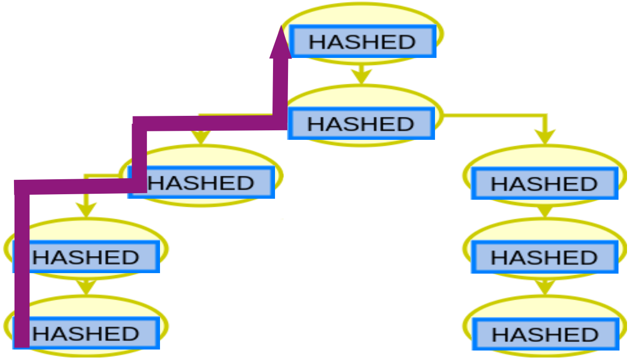
Description

Saving – Signing the HTML tree

```html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
      <h1>DECENARCH</h1>
  </body>
</html>
```



- Leader’s MasterTree

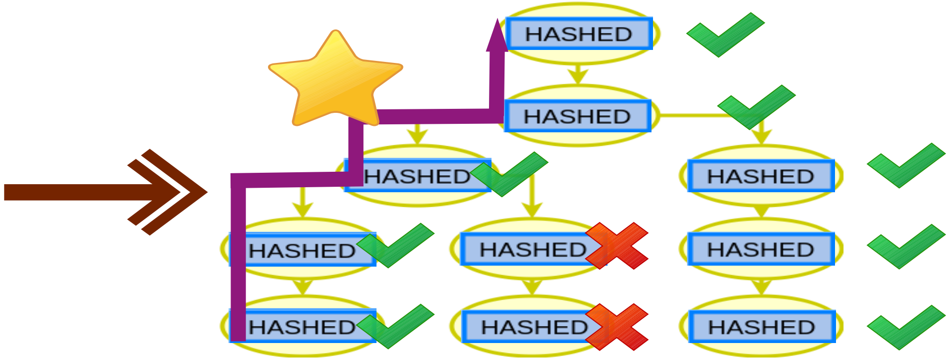


- Child’s Local Tree

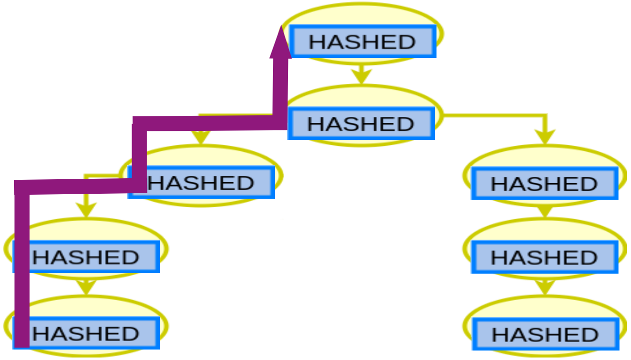
Description

Saving – Signing the HTML tree

```html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <h1>DECENARCH</h1>
  </body>
</html>
```



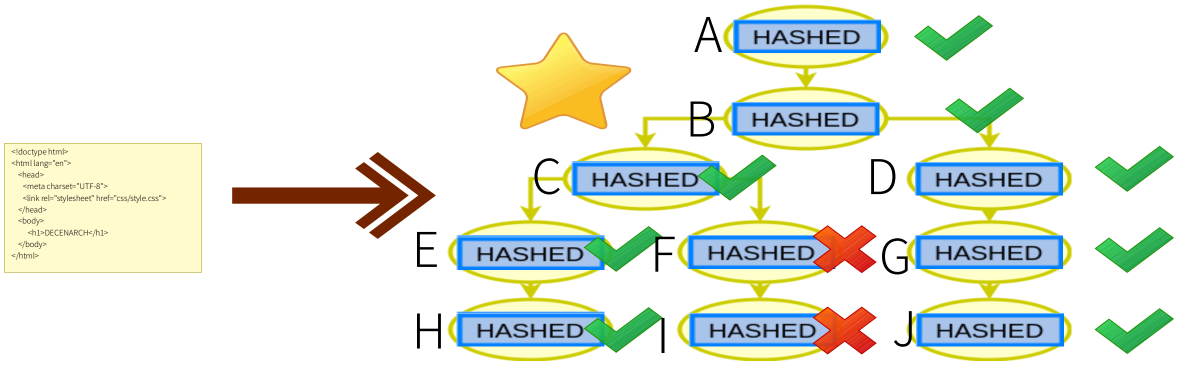
- Leader’s MasterTree



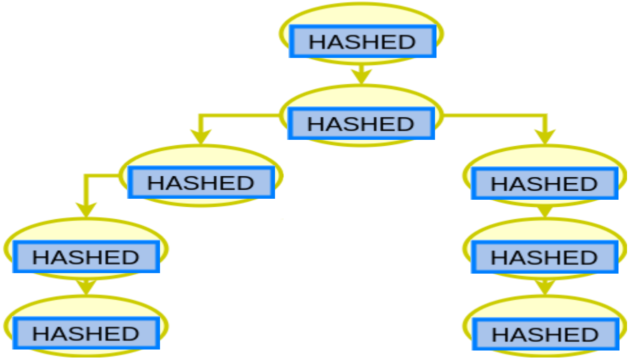
- Child’s Local Tree

Description

Saving – Signing the HTML tree



- Leader's MasterTree



- Child's Local Tree

- Nodes in BFS order
- Seen array
  $1_A\ 1_B\ 1_C\ 1_D\ 1_E\ 0_F\ 1_G\ 1_H\ 0_I\ 1_J$
- Signature
  $\mathrm{sign}(h_A+h_B+h_C+h_D+h_E+0+h_G+h_H+0+h_J)$

Description

Saving - Aggregation

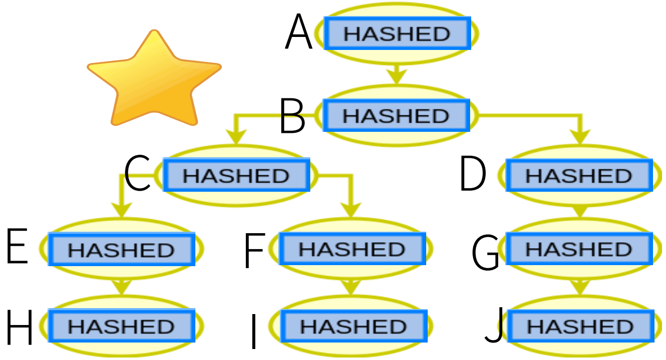
- Seen arrays

$$1_A\ 1_B\ 1_C\ 1_D\ 1_E\ 1_F\ 1_G\ 1_H\ 1_I\ 1_J$$

$$1_A\ 1_B\ 1_C\ 1_D\ 1_E\ 0_F\ 1_G\ 1_H\ 0_I\ 1_J$$

$$1_A\ 1_B\ 1_C\ 0_D\ 1_E\ 1_F\ 0_G\ 1_H\ 1_I\ 0_J$$

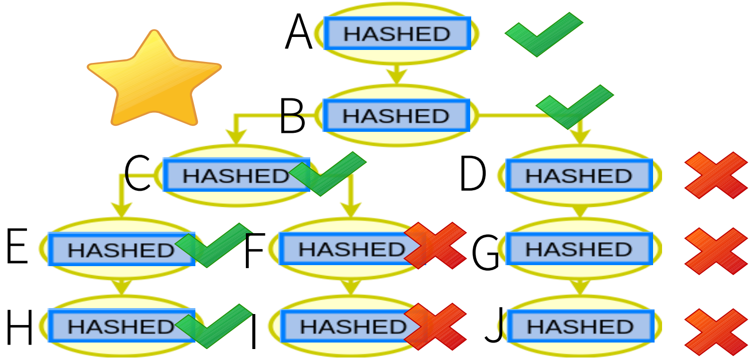
$$\Sigma\, 3_A\ 3_B\ 3_C\ 2_D\ 3_E\ 2_F\ 2_G\ 3_H\ 2_I\ 2_J$$



- Leader’s MasterTree

Description

Saving - Aggregation



- Leader’s MasterTree

- Seen arrays

$$1_A\ 1_B\ 1_C\ 1_D\ 1_E\ 1_F\ 1_G\ 1_H\ 1_I\ 1_J$$
$$1_A\ 1_B\ 1_C\ 1_D\ 1_E\ 0_F\ 1_G\ 1_H\ 0_I\ 1_J$$
$$1_A\ 1_B\ 1_C\ 0_D\ 1_E\ 1_F\ 0_G\ 1_H\ 1_I\ 0_J$$
$$\Sigma\, 3_A\ 3_B\ 3_C\ 2_D\ 3_E\ 2_F\ 2_G\ 3_H\ 2_I\ 2_J$$

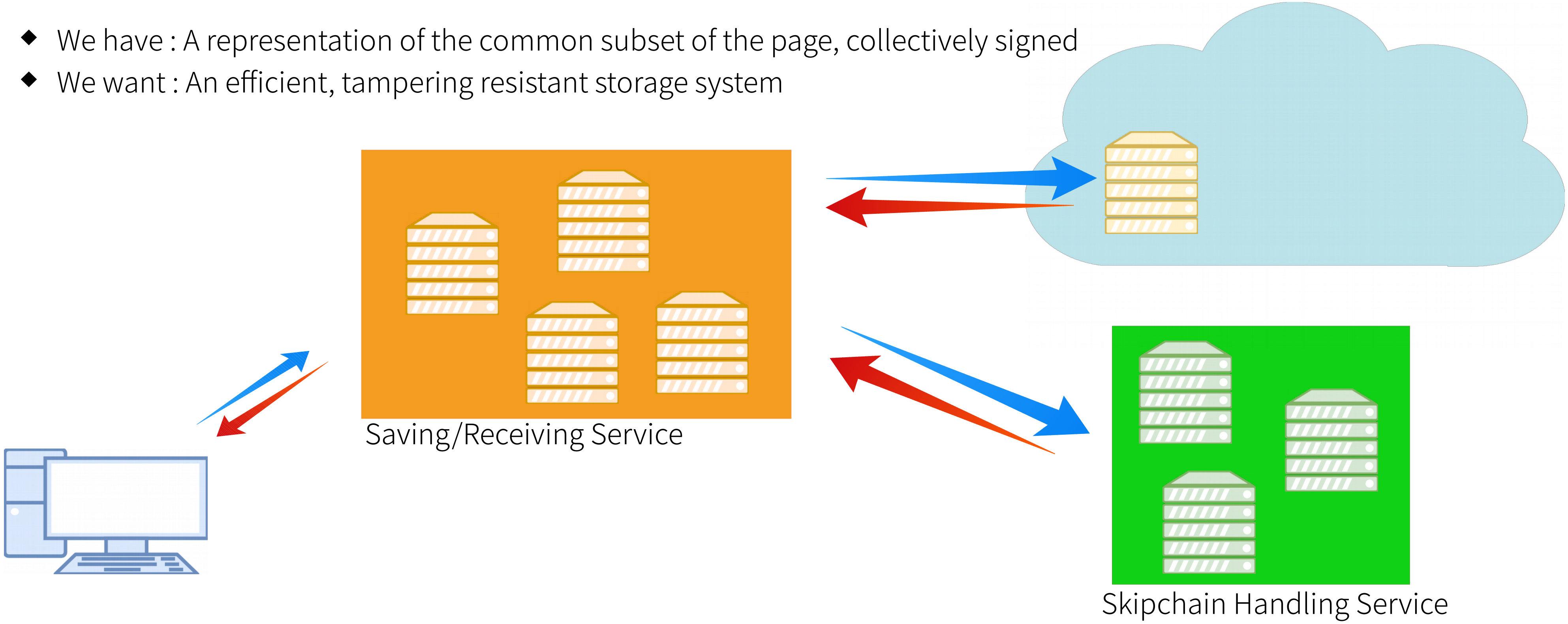
- Keep A,B,C,E,H

  Output html code collectively signed

Description

Handling the Skipchain

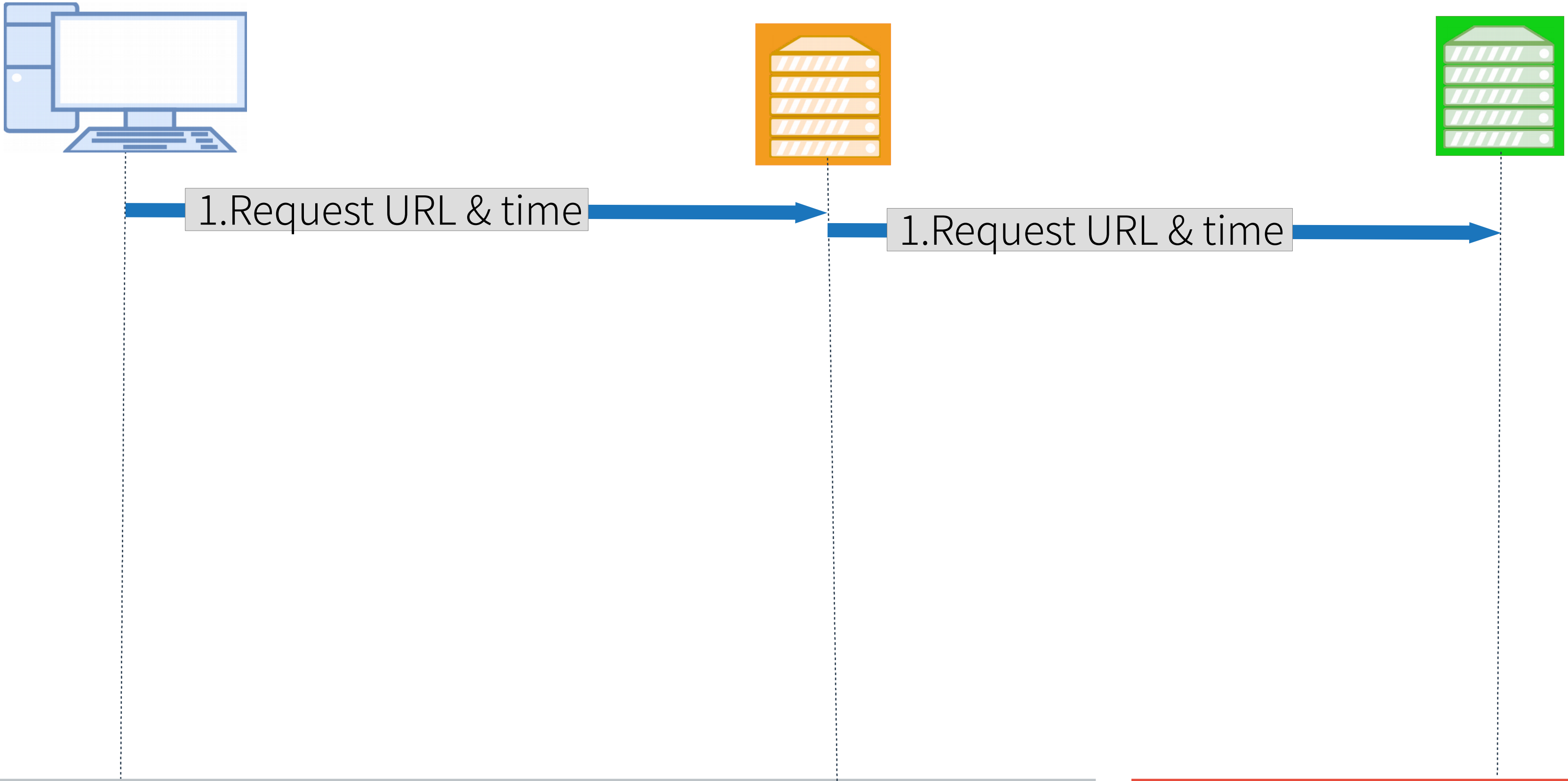
- We have : A representation of the common subset of the page, collectively signed
- We want : An efficient, tampering resistant storage system



Description

Retrieving the archived web page

1. Send new data.

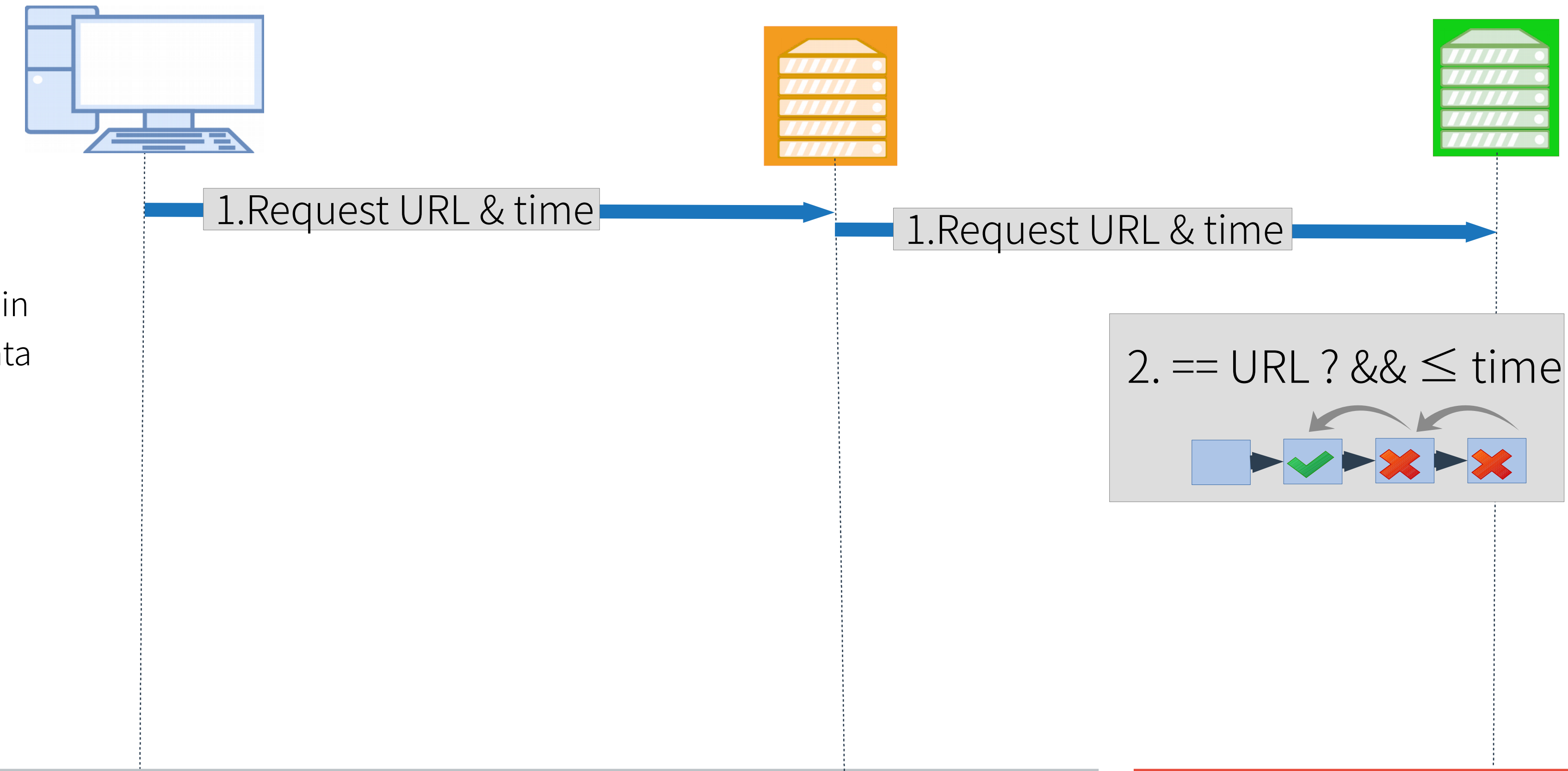


Description

Retrieving the archived web page

1. Send new data.

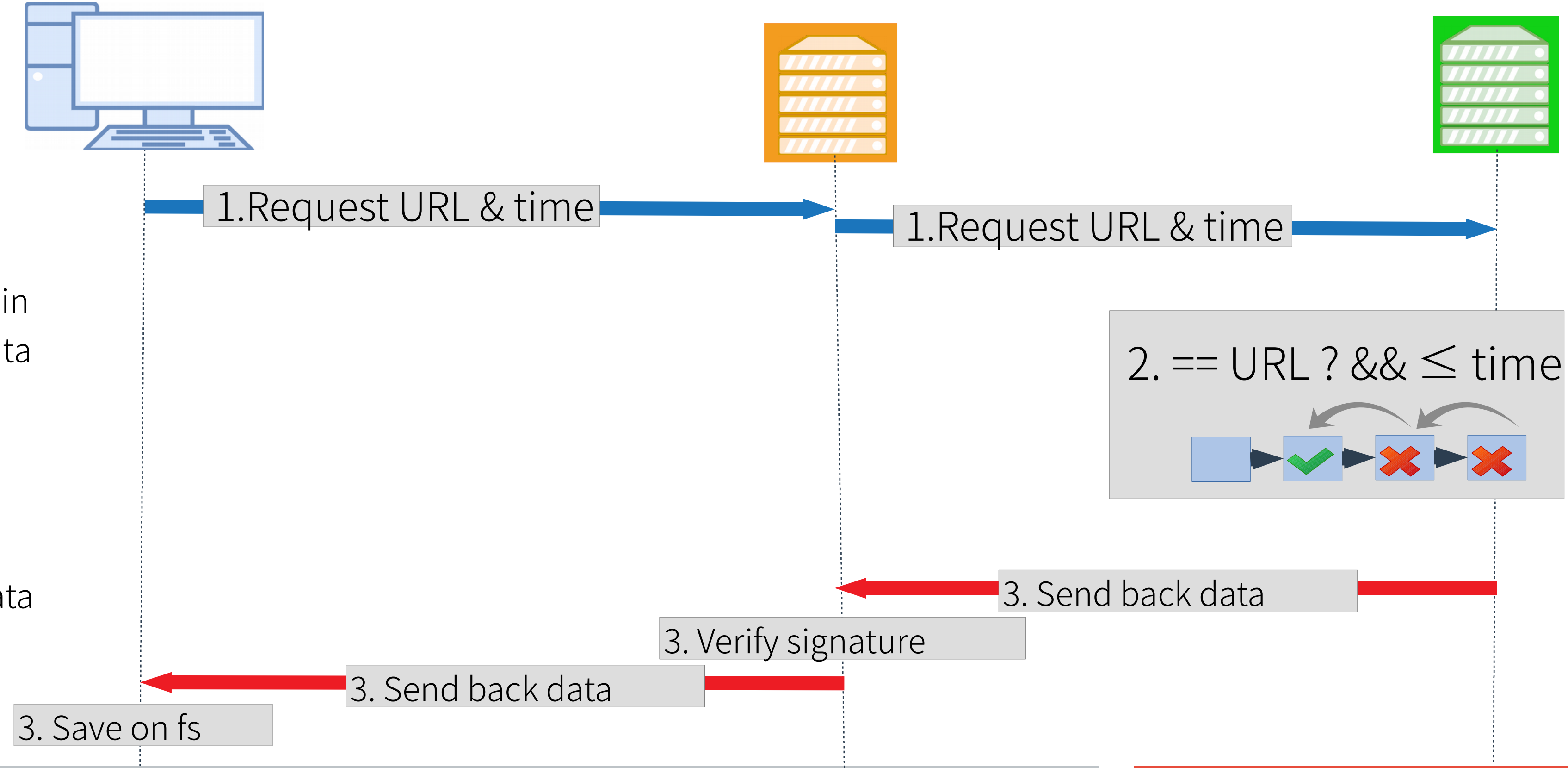
2. Look in blockchain
   to find correct data



Description

Retrieving the archived web page

1. Send new data.
2. Look in blockchain to find correct data
3. Send back the data

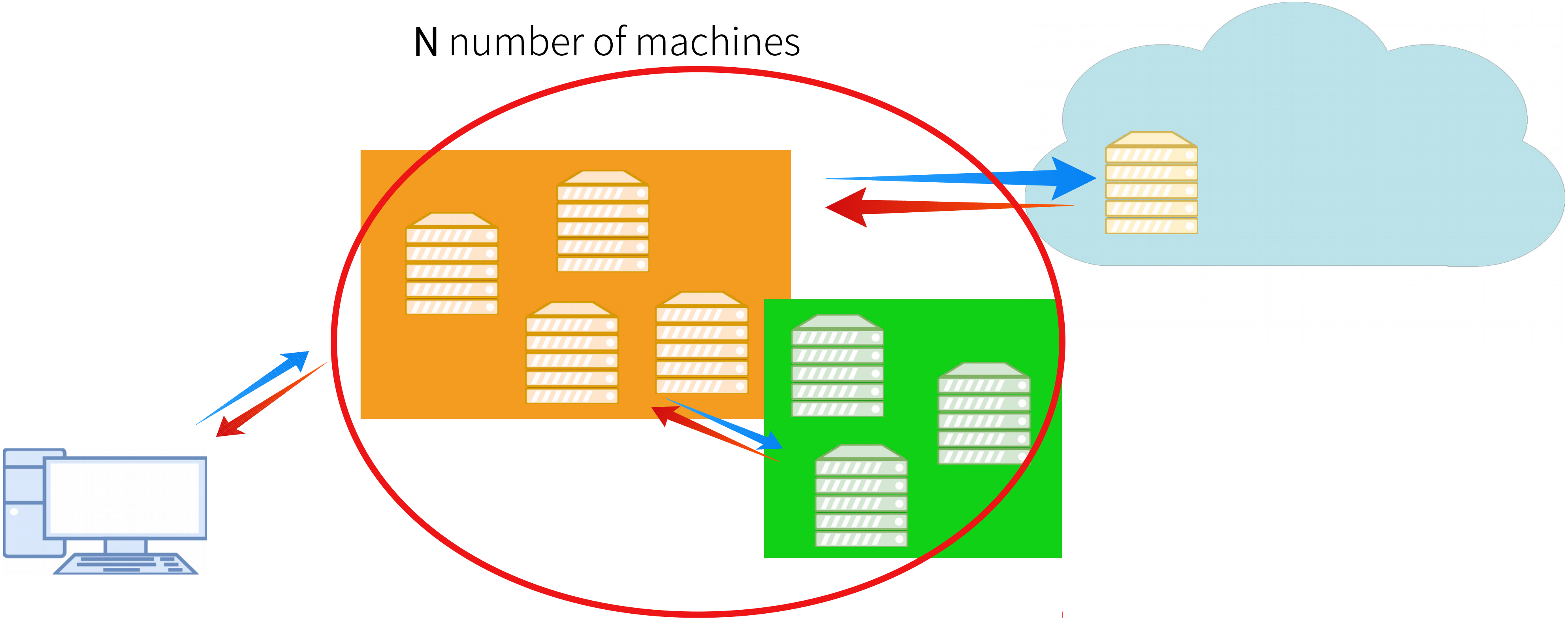


EVALUATION AND DISCUSSION

Evaluation and Discussion

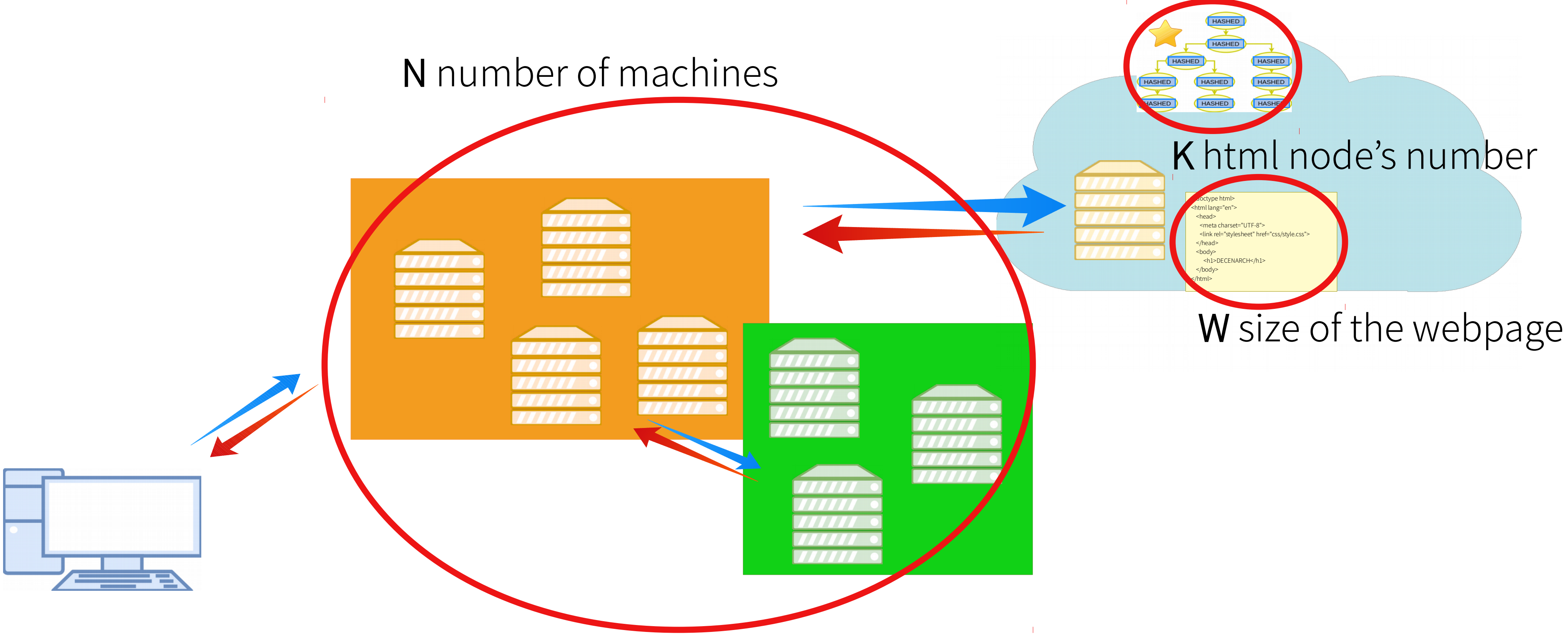
- Does it scales in terms of
  - Bandwidth use ?
  - Time complexity ?

- The ‘trusted leader’ constraint


N number of machines

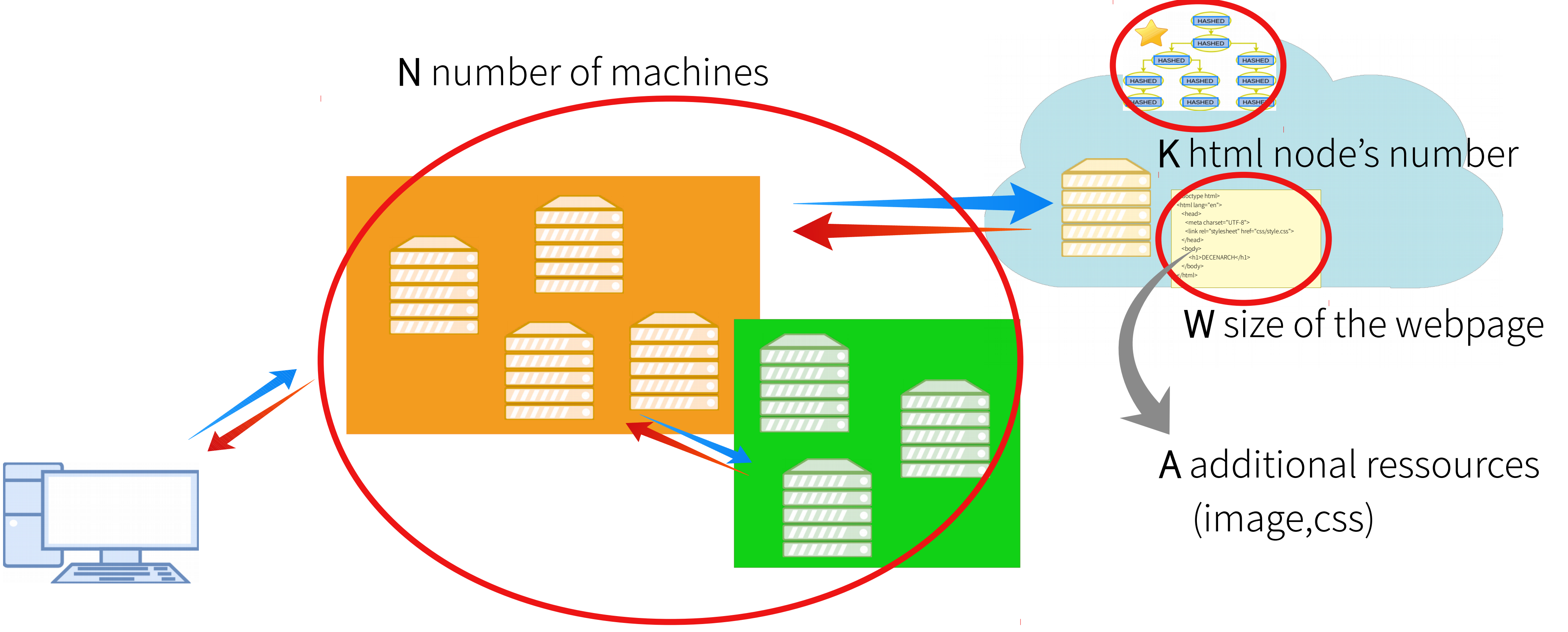
Evaluation and Discussion

Evaluation - Theory



Evaluation and Discussion

Evaluation - Theory



Evaluation and Discussion

Bandwidth

- Variables :
  - N number of machines.
  - W size of webpage.

- Bandwidth use is linear $O(N \cdot W)$
  - $N + 1$ request to the distant server of size $O(W)$
  - Finite total number of message of size $O(W)$

Evaluation and Discussion

Evaluation - Theory

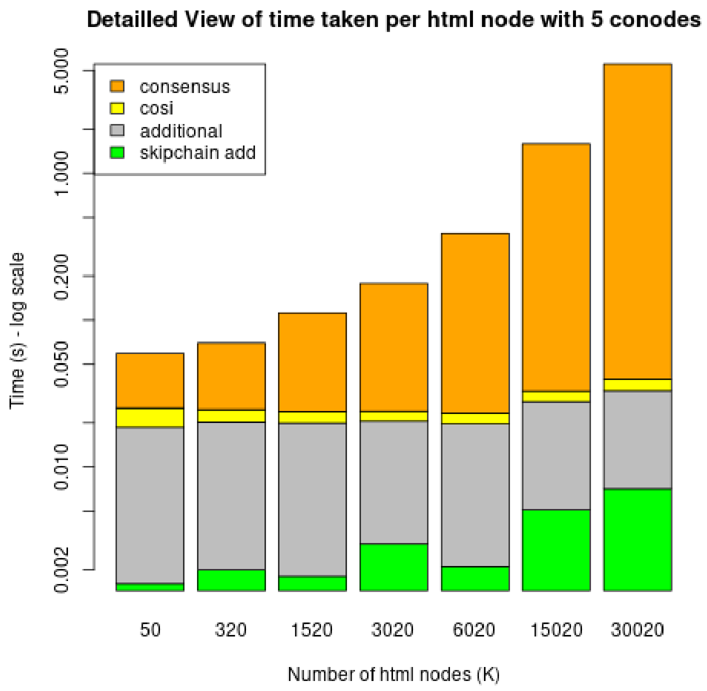
- Variables definitions :
  - N number of machines.
  - K html node's number.
  - A time cost of handling additional data (image,css) on one machine.

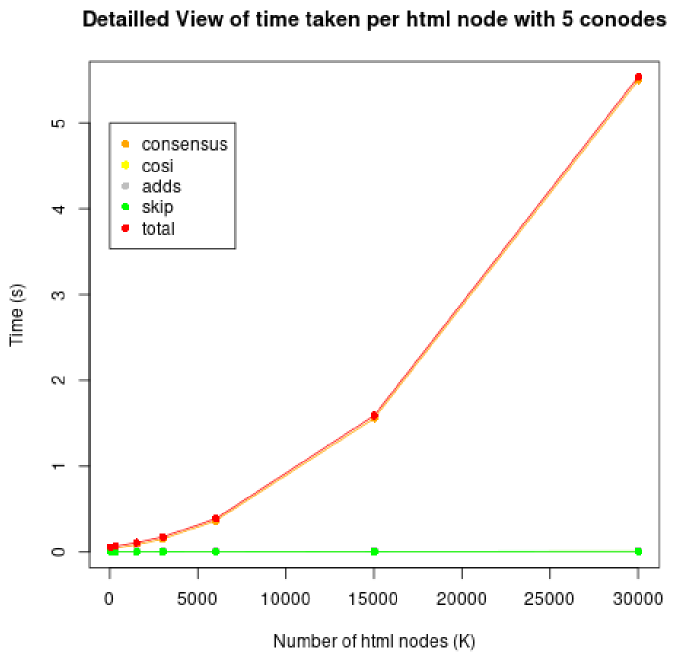
- Overall save time complexity is polynomial $O(N \cdot K^2 + (1+A) \cdot N \cdot K + N)$
  - Tree comparaison and aggregation is in $O(N \cdot K^2)$
  - Handling the additional data of the web page is in $O(A \cdot N \cdot K)$
  - Storing the website is in $O(N \cdot K)$
  - Collective signing is in $O(N)$

Evaluation and Discussion

Evaluation - Simulations

- Standardized Website
- Html Tree Node increase

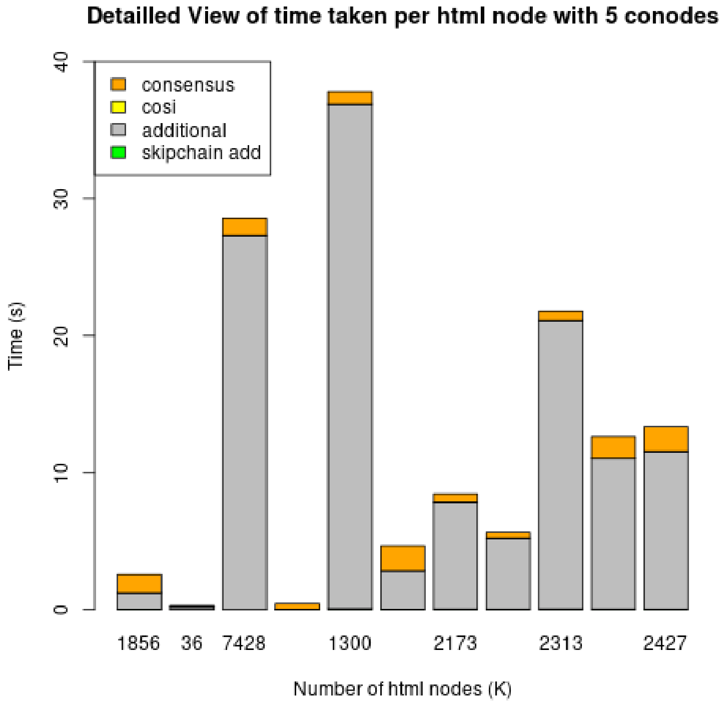




Evaluation and Discussion

Evaluation - Simulations

- Real-Life Website
- Html Tree Node increase

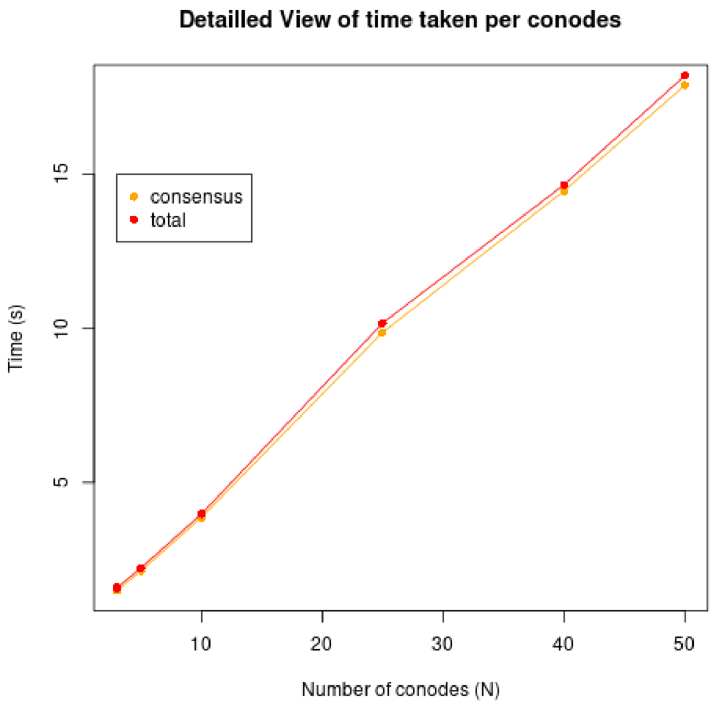


- Main time component :
  Handling the additional data.

Evaluation and Discussion

Evaluation - Simulations

- Standardized Website
- Conode nbr increase



- Main time component :
  The consensus
- Seems linear but require a larger simulation

- Why the trusted leader ?

- Why the trusted leader ?

- Why the tree structure ?
  - Keep a valid html document anytime.
  - Granularity.

Evaluation and Discussion

Discussion

- Why the trusted leader ?

- Why the tree structure ?
  - Keep a valid html document anytime.
  - Granularity.

- Why a reference ?
  - Union of Tree is NP.
  - Undeterministic matching, depends on order.

DEMO

"Anything that can go wrong will go wrong".
- Murphy's Law

Demo
Ain’t nobody got time for demo

Conclusion

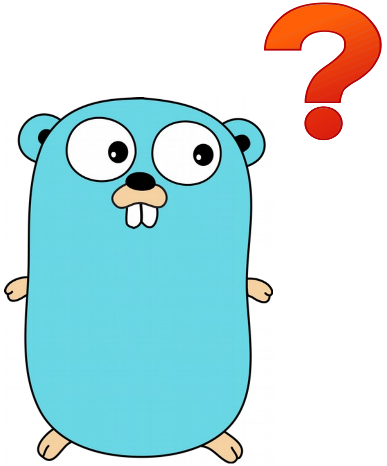
- Decentralized Internet Archive
  - Tree-based consensus with largest common subset
  - Decentralized storage with skipchain
  - Has a polytime complexity in $O(K^2 \cdot N)$

- Improvements ?
  - Storage Management
  - Additional Data filtering
  - Finer granularity
  - Confidentiality

Conclusion

- Decentralized Internet Archive
  - Tree-based consensus with largest common subset
  - Decentralized storage with skipchain
  - Has a polytime complexity in $O(K^2 \cdot N)$

- Improvements ?
  - Storage Management
  - Additional Data filtering
  - Finer granularity
  - Confidentiality



Reference

- [gopher] Takuya Ueda, https://github.com/golang-samples/gopher-vector
- [Master Thesis] Plancherel Nicolas 2018, Decentralized Internet Archive, EPFL