Distributed Hosting

A Blockchain Based Distributed Hosting Engine

Alex Oberhauser, Stavros Champilomatis and Maria Efthimiadou

Copyright © 2015 Alex Oberhauser, Stavros Champilomatis and Maria Efthimiadou

PUBLISHED BY PUBLISHER

TODO

Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the "License"). You may not use this file except in compliance with the License. You may obtain a copy of the License at `http://creativecommons.org/licenses/by-nc/3.0`. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

First printing, April 2015

Contents

I Introduction

1 Introduction 7
1.1 Hosting Paradigm Shift 7
1.2 Use Cases 7

II Specifications

2 Blockchain Specification 11
2.1 Transaction Definition 11
2.1.1 UPDATE Transaction Definition 11
2.1.2 CLONE Transaction Definition 12
2.2 Block Definition 12

3 Protocol Specification 13
3.1 Blockchain Overlay Network 13
3.2 Data Overlay Network 13
3.3 Hosting Algorithm 13

III Appendix

Bibliography 17

Index 19

Introduction

1 **Introduction** 7
1.1 Hosting Paradigm Shift
1.2 Use Cases

1. Introduction

1.1 Hosting Paradigm Shift

1.2 Use Cases

II Specifications

2 Blockchain Specification 11
2.1 Transaction Definition
2.2 Block Definition

3 Protocol Specification 13
3.1 Blockchain Overlay Network
3.2 Data Overlay Network
3.3 Hosting Algorithm

2. Blockchain Specification

2.1 Transaction Definition

This section defines transactions and their relationship to each other and to external data. It also defines a (de-)serialization format to and from the binary format that is transferred over the wire. In Figure 2.1 a chain of transactions, with relevant fields, is shown. This example includes three hosts that have cloned the page and in total three updates.

Figure 2.1: Transaction chain example for one page.

2.1.1 UPDATE Transaction Definition

Update transactions can only be created by the page creator. Additional to the proof of ownership they also include a reference to the new content, with related checksums.

TRANSACTION HEADER	
TransactionID	Double SHA256 hash of this transaction.
Parent	The previous update *TransactionID*
ScriptSig	**TODO Future Work:** Proof here ownernship of the hostname and all requirements that come with this update.
Hostname	The tor hidden service onion hostname of this page. Tor hidden services are needed to be able to host page behind firewalls and NATs. Additional they protect the identity of the content owner and more importantly of clone hosts.
CONTENT SECTION	
Snapshot	The hash value of the content. This hash value is used to find all peers in the Distributed Hash Table (DHT) that share currently this snapshot. The DHT is bootstrapped by using all active hostname in the blockchain. For increased compatibility the MAGNET URI scheme is used.
SnapshotChecksum	Checksum of the Snapshot content, used to verify if the downloaded data.
ResponseChecksum	Checksums for all endpoints of this page. Makes each page verifiable and hence could be served by a random node. If the returned content does not match the here checksum the response is invalid.
Flags	**TODO Future Work:** Could be used for differentiation between incremental and full updates, but also indicating that this page is not hosted anymore and could be deleted.

2.1.2 CLONE Transaction Definition

Clone transactions are created and propagated by all hosts that start hosting a page or if they receive a new update transaction. This transactions always link back to exactly one update transaction.

TransactionID	Double SHA256 hash of this transaction.
Parent	The previous update *TransactionID*
ScriptSig	**TODO Future Work:** Proof here ownernship of the hostname and all requirements that come with this update.
Hostname	The tor hidden service onion hostname of this page. Tor hidden services are needed to be able to host page behind firewalls and NATs. Additional they protect the identity of the content owner and more importantly of clone hosts.
Flags	Possible values are: *CLONE* when new update was received or *PURGE* if the page is not hosted anymore.

2.2 Block Definition

3. Protocol Specification

3.1 Blockchain Overlay Network
3.2 Data Overlay Network
3.3 Hosting Algorithm

III

Appendix

Bibliography 17

Index 19

Bibliography

[08] "Bitcoin: A peer-to-peer electronic cash system". In: *Consulted* 1 (2008), page 2012.

Index

B

Block Definition 12
Blockchain Overlay Network 13

D

Data Overlay Network 13

H

Hosting Algorithm 13
Hosting Paradigm Shift 7

T

Transaction Definition 11

U

Use Cases 7