Permissioned blockchains as ownership-registration systems

Master's Thesis

Robert Diebels
July, 2017

As part of a Master's graduate project at the Universiteit van Amsterdam

Supervisor: M. X. Makkes, Postdoctoral researcher, Vrije Universiteit

Abstract

Contents

Chapter 1

Introduction

Blockchain-technology, the technology underlying Bitcoin [7], promises efficiency improvements in a variety of industries [6]. As a result the amount of systems which implement blockchain-technology is steadily increasing [5]. Even-though the amount of systems is increasing, an objective comparison of the capabilities of these systems remains absent [5]. To resolve that absence this Thesis describes an experiment designed for such a comparison.

1.1 Permissioned blockchains

Blockchains which use a type of access control are called permissioned blockchains. Permissioned blockchains offer advantages over unpermissioned blockchains such as Bitcoin which allow anyone to participate. For example, as a form of access control a permissioned blockchain might require a participant to supply "real-world" identification. Identification of a participant lowers the chance a participant becomes malicious as there are bigger consequences to malicious activity.

Mitigation of malicious activities is why most unpermissioned blockchains do not perform well. As unpermissioned blockchains mitigate malicious activities by checking participant activities many times over. Contrary to unpermissioned blockchains, permissioned blockchains can avoid checking activities many times because they can assume participants meet demands such as identifying themselves. Avoiding checking activities gives permissioned blockchains significant performance increases.

The increased performance of permissioned blockchains makes them an interesting research subject. Particularly interesting would be researching performance of current market compared to one another and in-use systems. However, such comparative research has not received much attention from the scientific community [5].

1.2 A comparative experiment

This thesis describes a comparative experiment of 3 permissioned blockchains and 2 in-use systems. The in-use systems in question are owned by Sticht-

ing ter Exploitatie van Naburige rechten (SENA) and Bureau voor Muziek-Auteursrechten (Buma) Stichting tot Exploitatie van Mechanische Reproduktierechten voor Auteurs (Stemra) (Buma/Stemra), two Performing Rights Organizations (PROs). PROs represent rights-holders of musical works when acquiring payment for the use of musical works.

For the experiment the PRO use-case has been implemented in 3 permissioned blockchains. The implementations were then used in the experiment to compare their performance. To measure their performance each implementation was subjected to a load comparable to the PROs in-use systems. This allowed a comparison to be made between implementations and in-use systems.

1.3 Contributions

The primary contributions of this Thesis are:

- A mapping study on the current state of research on permissioned blockchains.
- Implementation of a music-industry use-case in several permissioned blockchains.
- Permissioned blockchain performance evaluation through a controlled experiment.

Chapter 2

Literature Survey

Examining the current body of knowledge on permissioned blockchains was done by conducting a mapping study [5] based on [10] and [9]. This chapter serves as a summary outlining the study's methodology, scope and findings. The full study is available on GitHub[1].

2.1 Methodology

A mapping study for software engineering as described in [9] consists of 5 steps; (1) a scope definition by a set of research questions, (2) conduction of a search based on defined search terms, (3) a search result screening by applying exclusion and inclusion criteria, (4) classification of results through abstract key-wording, (5) data extraction based on a set of data extraction items. Here the term "results" was broadened to include scientific studies and implementations of permissioned blockchains accommodating the study's research questions.

2.2 Scope

The scope of the study was limited by the definition of a set or research questions. Serving the purpose of examining the current state of permissioned blockchain literature, identifying existing permissioned blockchain implementations, assessing empirical evidence gathering in literature and practice and lastly serving to identify research of permissioned blockchains in the context of the music-industry. The following research questions detail the question and why it was of importance to answer for the purpose of this Thesis.

Which topics are addressed in current research on permissioned blockchains? Answering this question is important to assess which topics are being researched. This will help judge to what degree the graduate project can contribute to the field.

[1] `https://github.com/RobertDiebels/graduate-project/blob/master/dist/experiment.pdf`

What implementations of permissioned blockchains exist? Evaluating which permissioned blockchains exist and what flaws they have will help assessing the state of current permissioned blockchain technology. In turn this can create a framework for which measurements should performed when assessing a permissioned blockchain.

What experiments are being performed on permissioned blockchains? Assessing which experiments are performed on permissioned blockchains allows determining what the state of empirical evidence gathering in the field is. This will advance future experiments by providing an overview of which measurements are deemed valuable.

Is there any research on blockchains related to the music-industry? This question is relevant for the graduate projects' topic, relevancy and potential contributions to the field.

2.3 Results

The results of the survey showed that there is a focus on reporting on permissioned blockchains and improving the technology. With respectively 28% reporting and 48% suggesting improvements. Surprisingly only 43.8% of the research which suggested solutions or improvements did any experimentation and it was solely aimed at validating a suggested application or improvement. Search results for permissioned blockchain implementations revealed 7 blockchain implementations and 2 implementations with blockchain-like properties. A total of 8 implementations allowed for permissioning access.

2.4 Conclusion

Based on the survey's results we reach the following primary findings [5, p.16].

Blockchain research lacks comparative research Current literature is focused on improving several flaws in blockchain-technology. Such as privacy-preservation, consensus protocols, access control and more. The improvements suggested in these studies are then evaluated with the use of experimentation. This survey found no comparative research on improvements addressing the same flaw.

Blockchain research lacks research aimed at reproduction of results The results of this survey indicate that there is a lack of reproducing results of previous studies. In fact there was not one study aimed at reproducing results of other studies. The field could benefit from validating earlier research addressing and identifying flaws in experimental design.

Blockchain research lacks use-case specific research To validate the usefulness of blockchain technology it needs to be implemented and used. The survey identified several articles which suggest potential use-cases yet provide no implementation [1] [11].

Blockchain research lacks evaluation of current implementations The literature survey results indicate that implementations of blockchains have

not been objectively evaluated. Comparison of implementations could provide insights into general blockchain flaws and the suggestion of new improvements.

These findings indicate that the Thesis's aim is of value to the field and that several contributions can be made by performing a comparative experiment between current permissioned blockchains.

Chapter 3

Research methodology

This chapter defines the research methodology used in this Thesis. The sections in this chapter outline the graduate projects' hypothesis, research questions for evaluation of the hypothesis, an experimental design to answer said questions and how data gathered from the experiment is to be analysed.

3.1 Hypothesis

"Permissioned blockchains outperform current ownership-validation systems in terms of throughput and verification-speed."

3.2 Primary research questions

How well do blockchains perform in permissioned environments compared to current systems? This question is answered via a comparative experiment detailed in the next section. The comparative experiment is used to gauge permissioned blockchain performance compared to current systems and each other.

What measurements should be taken to compare permissioned blockchains? Comparing several permissioned blockchain implementations can only be done by comparing them based on a unit of measurement. Which units should be measured and why?

3.3 Experimental design

The experimental design described in this section is largely based on the design by [2, p.64]. The version of the design defined in this Thesis examines a PROs use-case implemented in several permissioned blockchains. It is useful to read chapter 5 to gain insight into the PRO use-case.

Measuring each blockchains performance is achieved by spawning Amazon EC2 instances doubling from 4 to 64 each run. A single instance runs 2 docker containers the first acts a a validating node the second generates transactions on the blockchain.

3.3.1 Variables

Independent The following independent variables are part of the experiment. A number of variables depend on external factors. I_{min_val} is dependent on the minimum number of validating nodes demanded by the protocol of a blockchain implementation and transaction-size is dependent on the Minimum Viable Data (MVD) of the use-case.

1. CPU speed (GHz)
2. RAM (GiB)
3. Number of validating nodes (N_{val}) (I_{min_val}, 64)
4. Number of faulty nodes (N_{fault})
5. Block-size (MB) (128, 32768)
6. Transaction-size (KB) (MVD)
7. Run-time (200 blocks)

Dependent The following dependent variables are to be measured for performance evaluation.

1. Throughput, transactions per second (Tx/s).
2. Latency, block-formation per second (Bf/s).

3.3.2 Deployment

Hardware Each node in the network will be hosted on a separate Amazon EC2 t2.medium instance. The t2.medium instance hardware specifications related to independent variables can be found in Table 3.1.

Independent variable	**Instance value**
Computing instance	t2.medium
CPU Speed	(0, 3.3) Ghz
RAM	4 GiB
Storage	Amazon EBS gp2
Volume Size	1 GiB - 16 TiB
Throuput Max	160 MiB/s[1]

Table 3.1: Amazon EC2 hardware specifications

Containerization All blockchain implementations are containerized using docker and their images are placed in the Docker Hub. The images will be made available at `https://hub.docker.com/u/robertdiebels/`.

[1] t2.medium has lower throuput then defined on the Amazon EBS page. How much is unclear.

3.3.3 Blockchain implementations

Inclusion In [4] recommendations are made for software engineering researchers to ensure their algorithms are reproducible. Those recommendations and the suggestion made in [3] to use docker containers serve as the base criteria for an implementations' inclusion into the experiment.

The recommendations used as inclusion criteria are I, II, IV, V, VII and X. A description of all recommendations from [4] can be found in the Appendix.

The implementations found in [5] will be evaluated and included in the experiment if found they fulfill the inclusion criteria.

3.4 Result analysis

3.5 Internal validity

Chapter 4

Implementation

This chapter describes implementation specifics of the graduate-project experiment. The implementation specifics are explained bottom-up. First, section 4.1 details the infrastructure setup for the permissioned blockchains. Second, section 4.2 details the application orchestration for the nodes in the blockchain network. Lastly, section 4.3 describes the specifics of the blockchain applications.

4.1 Infrastructure

The experimental design described in this Thesis uses Amazon AWS EC2 instances as nodes in the permissioned blockchain networks. Amazon offers a variety of tools to interact with the EC2 platform to launch instances. Such as its web console and the AWS cli for usage in a command prompt environment. These tools are slow in use as they require a user to launch instances one by one.

To resolve this issue several cloud provisioning and management tools have been created. These tools resolve the problem through configuration files. The files are read and executed to provision cloud providers, eliminating the need for users to launch instances one by one. The decreased effort in setting up cloud instances and the share-able nature of the configuration makes the tools attractive for use in experiments.

The infrastructure configuration files used in the graduate-project experiment were generated using Kubernetes Operations (Kops). Kops enables a user to create a Kubernetes cluster on several cloud providers. Kubernetes is a container orchestration tool which offers rapid deployment of applications containers. Containers are wrappers around applications and their configuration and they allow applications to be launched without worrying about configuration details. Kops is used to either launch the cloud instances and start Kubernetes or, generate configuration files for the cloud provisioning and management tool Terraform.

The experiment used the Terraform configuration files generated bye Kops to create Kubernetes clusters with fixed amounts of nodes. Terraform configuration files were chosen over running the cluster with Kops and another tool called Apache Brooklyn for several reasons; (1) ease of use, (2) reproduction

purposes, (3) project maturity, (4) state of documentation. Terraform has been in development for well over three years with an active developers community. Apache Brooklyn and Terraform boast the same features however during implementation Terraform in combination with Kops were significantly easier to work with and the state of Terraform's documentation is currently much better than that of Apache Brooklyn.

4.2 Application orchestration

During the experiment the permissioned blockchain applications were managed using Kubernetes. Kubernetes is an application container orchestration tool. Containers are wrappers around software applications and any configuration needed for an applications. Containers are often versioned which allows containers or rather the applications within them to be re-used with ease [3]. Kubernetes distributes application containers across nodes in a network and allows users to load-balance containers, start them, stop them and execute other commands within containers.

Currently there are several container application formats. For example, Linux Containers, RKT and Docker containers. The experiment uses Docker containers for the deployment of the blockchains applications. This was done for following reasons (1) Docker is the default container format for Kuberenetes, (2) Docker widely used throughout the industry, (3) Docker has excellent documentation and (4) some of the permissioned blockchains provide Docker containers as a means to setup the blockchain-network.

4.3 Permissioned blockchain applications

4.3.1 Hyperledger Burrow

Hyperledger Burrow is a blockchain node which serves as the default node in Monax permissioned blockchains. Monax is a company which offers tooling to easily create and manage blockchains. The Monax platform can be thought of as a layered system where (1) the top layer is an API gateway which allows external applications to call the system, (2) the middle layer is a permissioning layer, (3) the bottom is the blockchain which is being used to store data.

For the experiment the Monax tooling was used to create a blockchain-network configuration with Burrow nodes. These configurations were then loaded into the Kubernetes cluster. Each node in the Kubernetes contained two Docker application containers. The first container contained a Burrow node, the second contained a NodeJS application which wrote transactions onto the blockchain network.

4.3.2 Hyperledger Fabric

Hyperledger Fabric is a platform for distributed ledger solutions. The Hyperledger Fabric platform consists of a set of tools to operate a blockchain network. The network has several components (1) an ordering node responsible for connecting nodes to one another via a communication channel, (2) peer nodes which

evaluate transactions on the network and (3) client nodes which interact with the blockchain.

During the experiment 1 ordering node and a range of 4 to 64 peer nodes and client nodes were deployed. The ordering node and the peer nodes were deployed using Docker containers which Hyperledger Fabric offers. The client nodes used Fabric's NodeJS SDK to generate transactions.

4.3.3 Corda

Chapter 5

Use-case

This chapter describes the use-case under examination in this Thesis. Understanding the PROs use-case was achieved by interviewing several representatives of Dutch PROs. They were interviewed about current business processes and data-flow. Using their answers this chapter sets out to, clarify their use-case, identify entities operating with their business in section 5.2, model current PROs processes using Business Process Modelling (BPM) in section 5.3 and show how blockchains may benefit PROs in section 5.4.

5.1 Introduction

Under most legal systems in the world the creation of a work of art endows its creator with rights of ownership. Depending on the legal system this includes licensing other parties to use the work under a set of usage terms, often entailing monetary payments. In most cases a works' rights-holder wants assurance of payment by licensees. In the context of the music-industry ensuring payment is received is done by Performing Rights Organizations (PROs).

In the Netherlands PROs are by default non-profit foundations, the members of which are creators of musical works. Acting as a representative of a creator the foundation administrates and processes usage of works. Usage data is processed and rights-holders of a work receive payment.

5.2 Entities

This section specifies the entities which drive the PROs use-case. The entities consist of (1) rights-holders such as artists, composers and producers, (2) licensees such as the radio and television industry, the hospitality industry and Digital Service Providers (DSPs), (3) Performing Rights Organizations (PROs) who act as intermediaries between rights-holders and licensees.

5.3 Business Process Model

This section defines a BPM diagram of the PROs use-case. The diagram as shown in Figure 5.1 shows an abstraction of the processes within Buma/Stemra

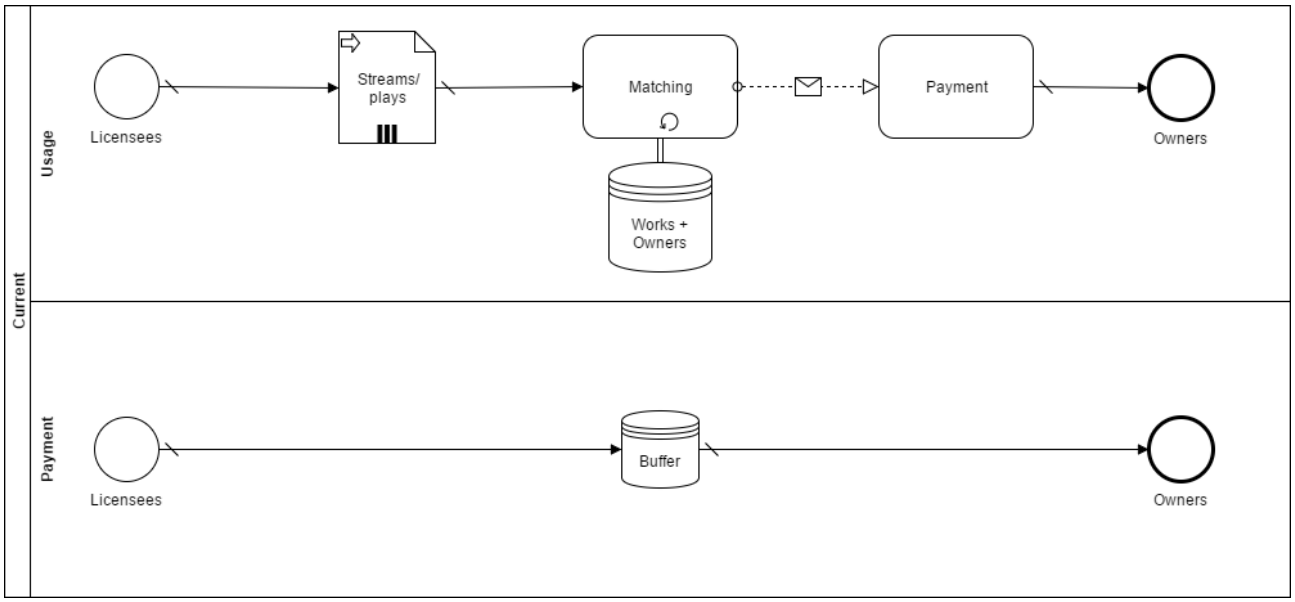

Figure 5.1: BPM of current PROs business processes.

and SENA.

The first lane in the diagram shows a process model in which (1) data is provided to PROs by licensees describing which works were played or streamed, (2) the works found in the data are identified through a matching process, (3) and a payment process for rights-holders. The second lane shows a parallel process were licensees transfer payment to a PRO. Payment is buffered and once the processes in the first lane are complete payments are made.

5.4 Blockchain

Blockchains can provide the record-industry with (1) A networked database for music copyright information (2) Fast, friction-less royalty payments (3) Transparency through the value chain (4) and access to alternative sources of capital [8, p.8]. The interviewed PROs agree with this observation. Stating that their primary motivations for using blockchains are improving operational transparency towards their members and the possibility of friction-less royalty payments.

Adjusting the model defined in Figure 5.1 for usage with a blockchain yields the model depicted in Figure 5.2. The model shows that the usage process is simplified by (1) allowing licensees to transfer usage data directly to the blockchain, (2) avoiding the matching process between PRO data and usage data (3) and subsequently, faster payment through a process tied into the blockchain. The payment process would be sped up significantly by avoiding the matching process though its structure would remain similar to the original. As instant payments would require owners and licensees to predetermine prices for usage. Considering current market conditions interviewed PROs found it unlikely that licensees would be willing to put a price on the usage of a single work. Their

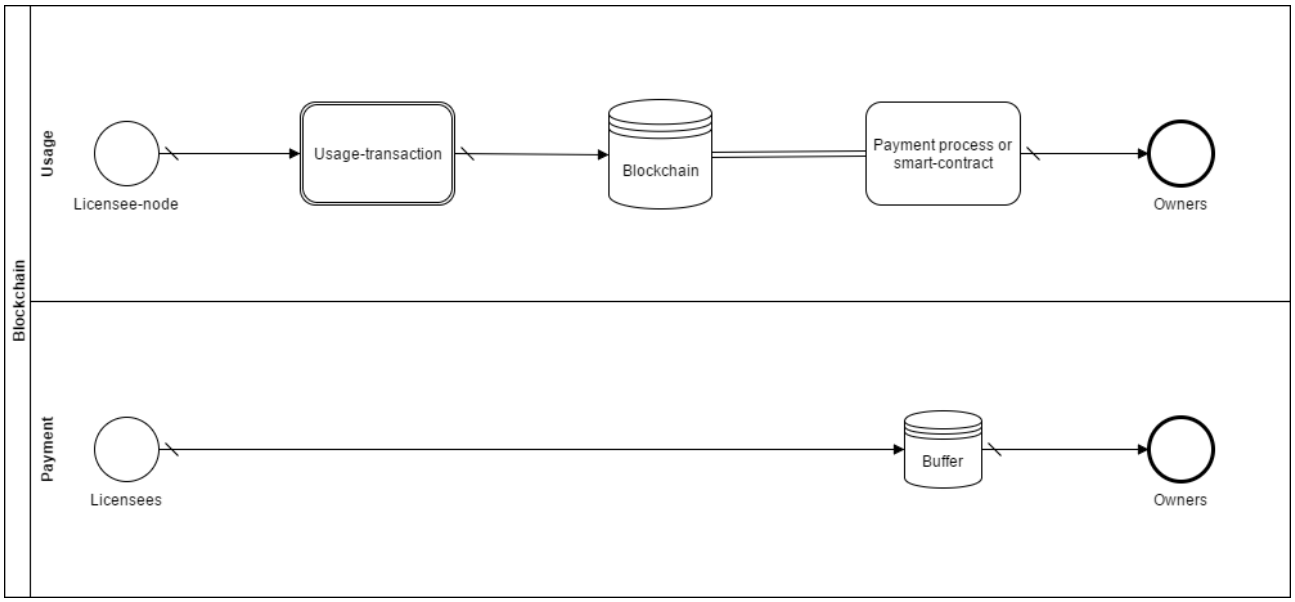

Figure 5.2: BPM of current PROs business processes.

reasoning was that larger DSPs would have to charge their users for usage of individual works. Their current business model involves a fixed monthly fee for all users, introducing pricing of individual works would complicate their model significantly.

5.5 Conclusion

The definition of the Performing Rights Organizations (PROs) use-case provided in this chapter shows that current PRO processes can be optimized by forgoing the need to match input data to stored data within the system. Blockchains can provide significant improvements to current processes. Most notable are the simplification, removal of obsolete processes and inherent properties of blockchains such as transparency of PRO operation.

Chapter 6

Evaluation

Chapter 7

Conclusion

Bibliography

[1] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. Medrec: Using blockchain for medical data access and permission management. In *Open and Big Data (OBD), International Conference on*, pages 25–30. IEEE, 2016.

[2] Ethan Buchman. *Tendermint: Byzantine Fault Tolerance in the Age of Blockchains*. PhD thesis, 2016.

[3] Jürgen Cito, Vincenzo Ferme, and Harald C Gall. Using docker containers to improve reproducibility in software and web engineering research. In *International Conference on Web Engineering*, pages 609–612. Springer, 2016.

[4] Tom Crick, Benjamin A Hall, and Samin Ishtiaq. " can i implement your algorithm?": A model for reproducible research software. *arXiv preprint arXiv:1407.5981*, 2014.

[5] Robert Diebels. Literature survey - ownership using persmissioned blockchains, 2017. Available at `https://github.com/RobertDiebels/graduate-project/blob/master/dist/survey.pdf`.

[6] Juri Mattila et al. The blockchain phenomenon–the disruptive potential of distributed consensus architectures. Technical report, The Research Institute of the Finnish Economy, 2016.

[7] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[8] Marcus O'Dair, Zuleika Beaven, David Neilson, Richard Osborne, and Paul Pacifico. Music on the blockchain. 2016.

[9] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *EASE*, volume 8, pages 68–77, 2008.

[10] Jesse Yli-Huumo, Deokyoon Ko, Sujin Choi, Sooyong Park, and Kari Smolander. Where is current research on blockchain technology?—a systematic review. *PloS one*, 11(10):e0163477, 2016.

[11] Guy Zyskind, Oz Nathan, and Alex Pentland. Enigma: Decentralized computation platform with guaranteed privacy. *arXiv preprint arXiv:1506.03471*, 2015.

Appendices

Appendix A

Reproduction recommendations

Recommendation	Description	Used as criteria
I	"[a paper] must describe the algorithm in such a way that it is implementable by any reader of that algorithm.". This recommendation is interpreted as the state of the documentation of an implementation.	Yes
II	"We recommend that code be published under an appropriate open source license...". This recommendation is interpreted as an implementation being open-sourced and open to modification for personal use.	Yes
III	"[we] recommend that basic programming and computational skills are taught as core at undergraduate and postgraduate level.". This recommendation was not used as an inclusion criteria. The aim the experiment is to compare implementations. Educational backgrounds are not considered.	No
IV	"The use of a principled, high-level programming language in which to write your software helps hugely with the maintainability, robustness and openness of the software produced.". Interpreted as an implementation being written in a commonly used high-level language in the industry. [Java, Go, C#, C++]	Yes

V	"Testing new complex scientific software is difficult – until the software is complete, unit tests may not be available. You should thus aim to link to/from publicly-shared code: shared code is inherently more test-able.". Interpreted as an implementations test infrastructure being clearly defined.	Yes
VI	"Code should always include links to papers publishing key algorithms and the code should include explicit relationships to other projects on the repository (i.e. Project B was branched from Project A).". Not used. Evaluating proper referencing in implementation code is not within the experiments' scope.	No
VII	"Providing the source code of the tool helps, of course. But you must also provide details of precisely how you built and wrote the software.". Interpreted as the presence of docker containers and build-documentation.	Yes
VIII	"Avoid creating new representations when common formats already exist. Use existing extensible internationally standardised representations and formats to facilitate sharing and re-use." Not used. Evaluation of representations within code is not within the scope of this experiment.	No
IX	"Benchmarks should be public. They should allow anyone to contribute, implying that the tests are in a standard format." Benchmarks are optional as the experiment sets out the evaluate performance in a self-defined use-case.	Optionally
X	"The Web and the cloud really do open up a whole new way of working...". Interpreted as deploy-able to cloud infrastructure.	Yes

Table A.1: Definitions of recommendations by [4] and their usage as inclusion criteria.