Semantic Web 14 (2023) 1167–1207
DOI 10.3233/SW-233409
IOS Press

Quantifiable integrity for Linked Data on the web

Christoph H.-J. Braun * and Tobias Käfer
Institute AIFB, Karlsruhe Institute of Technology (KIT), Germany
E-mails: braun@kit.edu, tobias.kaefer@kit.edu

Editor: Elena Demidova, University of Bonn, Germany
Solicited reviews: Marvin Hofer, ScaDS.AI, Germany; Two anonymous reviewers

Abstract. We present an approach to publish Linked Data on the Web with quantifiable integrity using Web technologies, and in which rational agents are incentivised to contribute to the integrity of the link network. To this end, we introduce self-verifying resource representations, that include Linked Data Signatures whose signature value is used as a suffix in the resource's URI. Links among such representations, typically managed as web documents, contribute therefore to preserving the integrity of the resulting document graphs. To quantify how well a document's integrity can be relied on, we introduce the notion of trust scores and present an interpretation based on hubs and authorities. In addition, we present how specific agent behaviour may be induced by the choice of trust score regarding their optimisation, e.g., in general but also using a heuristic strategy called Additional Reach Strategy (ARS). We discuss our approach in a three-fold evaluation: First, we evaluate the effect of different graph metrics as trust scores on induced agent behaviour and resulting evolution of the document graph. We show that trust scores based on hubs and authorities induce agent behaviour that contributes to integrity preservation in the document graph. Next, we evaluate different heuristics for agents to optimise trust scores when general optimisation strategies are not applicable. We show that ARS outperforms other potential optimisation strategies. Last, we evaluate the whole approach by examining the resilience of integrity preservation in a document graph when resources are deleted. To this end, we propose a simulation system based on the Watts–Strogatz model for simulating a social network. We show that our approach produces a document graph that can recover from such attacks or failures in the document graph.

Keywords: Web technologies, Linked Data, Social Linked Data (Solid), signatures

1. Introduction

How do we know whether data on the Web can be trusted if, after the publication of data on the Web, the agent controlling the published resource can modify or delete the resource at will? No commonly accepted mechanism ensures that the published data is not altered after it has been first made accessible on the Web. In addition, declaring authorship of information on the Web is not required by design: There is no standardised mechanism to incentivise agents to publish information that they would want to be held accountable for. In other words, it is hard to build trust on the Web, as anybody can publish anything on the Web without being held accountable; and later modify or delete it. In this paper, we want to work towards trust starting with the notion of integrity, and present a decentralised technical solution that makes interlinked data tamper-evident and at the same time authorship traceable, while incentivising agents to contribute to the integrity of the link network.

*Corresponding author. E-mail: braun@kit.edu.

The notion of integrity has several meanings [37]: While integrity can mean the adherence to high (e.g. moral) standards, which we cannot check for data on the Web, we apply a technical definition, where integrity refers to an "unimpaired condition" [37]. Yet, it can be argued that with traceable authorship, people publish data that adheres to higher standards, as they can get held accountable, thus we also indirectly address the former point. This paper is now about quantifying the degree of the unimpaired condition of data that can be verified to be from a certain user. By knowing that the data is unlikely to have changed and knowing by who the data is, we achieve reliability and thus trustworthiness.

That information on the Web is reliable forms the foundation of many applications: In e-commerce, a recent study [1] by McKinsey & Company, a management consulting firm, found that more than half of consumers from the Millennial generation always search for background information prior to buying a fashion product. In supply chains, item documentation offers significant benefits regarding traceability of faulty products, and thus increased efficiency and reduced disruptions [22]. In research, e.g. regarding COVID-19 vaccination intention, information from experts and health authorities as well as local newspapers have a positive impact, while alternative information sources and misleading claims pose a noticeable threat to vaccination intention [23]. In social applications, chats and forums in all flavors, interaction is built on the sharing of posts, comments or messages. Without these resources being reliable, accountability on the Web is missing and human interaction may suffer from fake news or even personal harassment. For example, with Twitter, a centralised micro-blogging platform, posts are not editable and the idea of the possibility of editing posts sparked fierce discussion in the Twitter community.[1] A mechanism to publish data on the Web would with quantifiable integrity and accountability would thus benefit a wide variety of domains.

As a solution, one may be tempted to envision a trusted third party handle all data publication and access control, similar to social media content with Facebook as the third party. This approach not only compromises data sovereignty but simply transfers the problem to the central agent. No such central agent is trusted by all agents on the Web. As an alternative solution, one may be tempted to envision a decentralised approach, e.g. a public blockchain to store hashes of the published data as proof of existence, which does not rely on trust in one specific agent. Such a system is for example explored in [8]. However, the blockchain approach introduces an additional system that has proven to be very expensive and inefficient due to the consensus mechanism. Our approach on the other hand provides a mechanism to publish trustworthy data on the Web only using Web technologies, without compromising data sovereignty or introducing an additional system.

Our work can be seen in context of ongoing and heatedly debated standardisation efforts at the W3C around Verifiable Credentials (VCs),[2] Linked Data Signatures (LDS),[3] and RDF Canonicalisation and Hashing (RCH).[4] Parts of the technologies in those efforts shall contribute to tackle the outlined problems. Our work can also be seen in a longer history of attempts to (re-)decentralise identity on the web [26,36], where the trend has been for long in the other direction, to the hindsight of people such as Jack Dorsey, the founder of Twitter, who regrets to have contributed to this trend.[5]

In this paper, we present an approach to publish Linked Data on the Web with quantifiable integrity, only using Web technologies, and in which rational agents are incentivised to contribute to the integrity of the link network. As such, the foundation of our approach is the Web architecture [30] and Linked Data [6]. The Solid Protocol [12] allows for adding access control. Our approach then consists in the following contributions:

- The notion of self-verifying resource representations, which we first showcased in [10], where the RDF that describes a resource is signed digitally using Linked Data Signatures (LDS)[3] and RDF-star,[6] and the signature is made part of the corresponding information resource's URI, thus forming a *Signed URI*.
  If the description changes, the change becomes evident as the hash of the description does not match the hash in the URI any longer. If a Signed URI is part of another signed resource's description, those links in the

[1] https://twitter.com/elonmusk/status/1511143607385874434
[2] https://www.w3.org/TR/vc-data-model/
[3] https://w3c.github.io/lds-wg-charter/
[4] https://www.w3.org/2022/05/04-proposed-rch-wg-charter/
[5] https://twitter.com/jack/status/1510314535671922689
[6] https://www.w3.org/2021/12/rdf-star.html

descriptions form a directed acyclic *document graph*, in which changes to one resource representation need to get propagated to all those linking to the representation. Thus, transitively the integrity of the graph can get established.
- The definition of trust scores to quantify how well a resource representation's integrity is preserved by the document graph's link structure.
- The definition of heuristics that help an agent determine to which other Signed URI to link under resource constraints.
- The evaluation of different graph metrics as trust scores, using which we show that Kleinberg's hubs and authorities [32] can serve as a trust score that lets agents, who want to maximise the trustworthiness of their own documents, contribute to the integrity preservation via the document graph's link structure.
- The evaluation of different heuristics for agents to determine the next link to set, using which we show that our *Additional Reachability Strategy (ARS)* outperforms other heuristics.
- The evaluation of the whole approach by looking at the resilience of the integrity preservation of the document graph in the presence of deleted resources and resources going offline (failures as imagined by Guéret et al. in [24]), using which we show that the graph can indeed recover.
- A simulation system based on the Watts–Strogatz model [50] and situated in a social network setting for conducting some of the evaluations.

The paper is structured as follows: In Section 2, we give the foundations for our work including the foundations of the methods used in the evaluations. In Section 3, we survey related research. In Section 4, we provide a running example, illustrate the overall architecture and introduce some helpful terminology. In Sections 5, 6, 7 and 8, we introduce our approach. In Section 9, we evaluate our approach regarding constructive agent behaviour, heuristic score optimization performance and score resilience in the document graph. In Section 10, we conclude.

2. Preliminaries

In this section we present the technological underpinnings of our contributions, which build on Linked Data and related technologies. Moreover, we build on approaches to analyse the web, which we present subsequently.

2.1. Linked Data

This section clarifies the notion of Linked Data in the context of the paper. The Linked Data Principles form the technological foundation which is build on and extended by the Social Linked Data (Solid) project. The Linked Data principles outline a lightweight standard to semantically describe and publish data in a decentralised fashion on the Web. Defined by Sir Tim Berners-Lee in [6], the four principles are:

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).
4. Include links to other URIs. so that they can discover more things.

2.1.1. URI, resource and resource representation

Adhering to the Linked Data principles, we use Uniform Resource Identifiers (URIs) [7] as names for things which can be virtual, physical or abstract. The thing that is identified by a URI is also called a *resource*. A resource's state may be described by the resource's *representation* [21] which is retrieved when the resource's URI is dereferenced. We define a resource's representation according to [21]:

Definition 1 (Resource Representation)**.** A resource representation is a sequence of bytes, described by representation metadata. The representation of a resource is retrieved upon dereferencing the URI identifying the resource.

When dereferencing a resource's URI yields a representation, the resource is called an *information resource*. A resource where no representation can be retrieved is called to a *non-information resource*. Typically, non-information resources are described along side with information resources in an information resource's representation. For example,

```
<https://timbl.solidcommunity.net/profile/card#me>
```

is a non-information resource identifying Tim Berners-Lee. Dereferencing the URI yields the representation of the *corresponding information resource*

```
<https://timbl.solidcommunity.net/profile/card>
```

i.e. the profile card where one might find information on the non-information resource. More formal definitions are provided in Section 2.1.3. As representations of information resources that are available on the Web are commonly referred to as Web documents, the terms *document*, *information resource* and *information resource representation* are used synonymously in the remainder of the paper, similar to [27].

A URI [7] is structured as follows:

```
scheme : authority / path [ ? query ] [ # fragment ]
```

Square brackets indicate optional parts. Query and fragment are optional parameters that are included when necessary or desired for the resource to be served as intended. The authority and path identify the address at which the resource is hosted. The scheme determines how the subsequent characters are to be interpreted. For example, a `http:` scheme denotes that interaction with the resource via the HTTP protocol may be possible.

2.1.2. HTTP

For interacting with a resource or, more specifically, its representation, we use the Hypertext Transfer Protocol (HTTP) [19]. With HTTP, two parties communicate with each other: a client, the requesting party, and the server, the responding party. We use the *HTTP POST* method to create and the *HTTP GET* method to retrieve a resource's representation as specified in [20]. For example, when accessing a URI with HTTP GET, the client initiates the communication with the host server identified from the authority and path of the URI. The server processes the request and responds by serving a representation of the requested resource to the client. The resource is served in a data format that has been negotiated between client and server during the communication. One family of such formats is described by the Resource Description Framework (RDF).

2.1.3. RDF

We create resource representations using graphs expressed according to the Resource Description Framework (RDF) [17]. Formal definitions in the realm of RDF are adopted from and stay close to the definitions of [27].

RDF specifies a graph-based data model: An RDF graph is defined as a set of triples. A triple consists of a *subject*, *predicate*, and *object*. In each position, a URI may identify the associated resource. In subject and object position so-called blank nodes may serve as graph-local identifiers. In object position, so-called literals represent values, e.g. numbers or strings. Literals, blank nodes, and URIs are called RDF terms. RDF terms, triples, and graphs are formally defined as follows:

Definition 2 (RDF Terms, Triple, Graph)**.** The set of RDF terms consists of the set of URIs $\mathcal{U}$, the set of blank nodes $\mathcal{B}$ and the set of literals $\mathcal{L}$. An RDF triple t is defined as $t = (s, p, o) \in (\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L})$. An RDF graph G is a finite set of RDF triples; $\mathcal{G}$ denotes the set of all RDF graphs. The set of all distinct URIs from a triple t is denoted by $uris(t)$, and given a graph G, $uris(G)$ denotes the set of all distinct URIs from G.

For example (using CURIEs for prefix abbreviations, # denoting a comment), the RDF graph

```
# a graph consisting of one triple
@prefix timbl: <https://timbl.solidcommunity.net/profile/card#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/>.

        timbl:me a foaf:Person .
```

indicates that Tim Berners-Lee is a Person. Such representation may, for example, be retrieved upon dereferencing the URI of the profile card, the information resource mentioned in Section 2.1.1.

Definition 3 (Information Resource, Dereferencing)**.** Let $\mathcal{U}_\mathcal{I} \subseteq \mathcal{U}$ be the set of all URIs identifying information resources. The set of all non-information resources is defined as $\mathcal{U}_\mathcal{N} = \mathcal{U} \setminus \mathcal{U}_\mathcal{I}$. The function $deref : \mathcal{U}_\mathcal{I} \mapsto \mathcal{G}$ models dereferencing an URI and returns the resources' representation, i.e. an RDF graph.

As described in Section 2.1.1, a non-information resource is typically associated to an information resource. This association is also referred to as *correspondence*.

Definition 4 (Correspondence)**.** The function $co : \mathcal{U} \mapsto \mathcal{U}_\mathcal{I}$ associates a resource with its information resource. For inputs from $\mathcal{U}_\mathcal{I}$, co behaves as the identity function.

For example,

```
<https://timbl.solidcommunity.net/profile/card#me>
```

the non-information resource identifying Tim Berners-Lee has the corresponding information resource of

```
<https://timbl.solidcommunity.net/profile/card>
```

An RDF graph may include URIs that correspond to information resources other than the graph's corresponding information resource itself. In other words, the graph links to other information resources whose representations, i.e. RDF graphs, may link to more information resources and so on. Thus, a directed graph of interlinked information resources, i.e. "Web documents", is formed. Such graph is thus referred to as *document graph*.

Definition 5 (Document Graphs)**.** Let the set of information resources referenced in an RDF graph G be defined as $refs_{\mathrm{IR}}(G) = \{co(u) | u \in uris(G)\}$. A document graph G_D is a directed graph whose vertices are information resources, $V(G_D) \subseteq \mathcal{U}_\mathcal{I}$, and whose edges are references from a representation to an information resource, $E(G_D) = \{(v,u) | v \in V(G_D) \wedge u \in refs_{\mathrm{IR}}(deref(v))\}$. Let $reach(v)$ denote the set of all information resources reachable by graph traversal from the input information resource v, i.e. by iterative application of $refs_{\mathrm{IR}} \circ deref$. The transitive closure of document graph G_D, denoted by ${G_D}^+$, is a graph with vertex set $V({G_D}^+) = V(G_D)$ and edge set $E({G_D}^+) = \{(v,u) | v \in V({G_D}^+) \wedge u \in reach(v)\}$.

To reiterate in other words: The terms *document*, *information resource* and *information resource representation* are used synonymously (as mentioned in Section 2.1.1). This is especially intuitive when considering Web documents. A Web document is an information resource and its representation is the document's content which in turn comprises the document itself. So when a URI identifies a Web document, that document is retrieved upon dereferencing the URI. Links in that document link to other Web documents, thus forming the document graph.

2.1.4. RDF-star

RDF-star [28] (sometimes also RDF*) extends the RDF abstract syntax to support easier modelling of statements about other statements. In particular, RDF-star allows for quoting triples, i.e. referencing without asserting, using additional notation of `<< triple t >>`. For example,

```
@prefix : <#> .
@prefix profile: <https://timbl.solidcommunity.net/profile/> .
@prefix timbl: <https://timbl.solidcommunity.net/profile/card#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/>.

        profile:card :includes << timbl:me a foaf:Person >> .
```

indicating that the profile card includes the statement that Tim Berners-Lee is a Person. At the same time, the "outer" statement does not make an assumption about the truth-value of the quoted statement. It only asserts that the profile card includes such a statement. Whether or not Tim Berners-Lee is actually a Person is not asserted. Quoted triples are referentially opaque (in contrast to reified triples), i.e. the quoted triple is considered one entity instead of a composition of its single terms.

2.2. Social Linked Data – the solid protocol

Tim Berners-Lee's endeavor to re-decentralise the Web manifests in the Social Linked Data (Solid) Project [36, 43]. The goal is to decouple applications from the data they consume, to break up centralised data silos and to give users back control over their data. To this end, the Solid Protocol [12] specifies a set of conventions and recommendations, building on the Linked Data Principles and existing W3C standards. For managing data in a decentralised manner and for building applications consuming this distributed data, the protocol connects mature Web technologies like the Resource Description Framework (RDF), RESTful HTTPS APIs borrowing from the Linked Data Platform [46] and an adaptation of Open ID Connect [16].

Users or, more general, agents store their data as web-accessible resources in so-called *Pods*, personal online data storages, where they control who has access to what part of their data. Pods are basically personal Web servers with user-defined access control. Each agent controls one or more Pods, where among its other data the agent's personal profile information is stored. An agent is identified by its *WebID* [44] which points to the location of the agent's profile information. Using its WebID, the agent is able to authenticate to a Pod when accessing a resource. On the Pod, user-defined *access control lists (ACL)* [11] determine if the accessing agent is authorized to perform its desired action (read, write, append or defining access control).

Additionally, agents are able to exchange messages using *Linked Data Notifications (LDNs)* [13]. Each Pod exposes an inbox, where any agent can post LDNs to, e.g. to request access to a specific resource or to announce the creation of a new resource.

2.3. Linked Data Signatures

Linked Data Signatures (LDS) are a way of modelling a cryptographic signature of an RDF graph or dataset. The distinguishing factor from a typical signature of a plain string or byte array is that the LDS is RDF syntax agnostic. It does not matter in which specific RDF syntax (Turtle, JSON-LD, Quads, . . .) the resource representation is available. The syntax of the original representation which the signature value has been calculated on does not need to match the RDF syntax of the representation to be verified. Any specific RDF syntax produces the same hash and thus can be verified with the information modelled by the LDS.

An LDS includes information on the calculation of the signature value. It specifies, e.g.,

- the *canonicalization algorithm* used to normalize the RDF graph, e.g. Hogan's algorithm [29]
- the *message digest algorithm* to calculate the hash, e.g. SHA-256 [38]
- the *signature algorithm* to sign that hash, e.g. Elliptic Curve Digital Signature Algorithm (ECDSA) [2] with curve P-256, such that the signature value can be verified
- the *cryptographic key* (or a link to the key) to be used for verification
- the *signature value* to be verified

In addition, the LDS typically includes when and by whom the signature was created. LDS are not yet standardized, i.e. there exists no recommendation on how to connect the signed data to the LDS. Nonetheless, LDS are listed as a valid signature scheme in the Verifiable Credential (VC) data model [47], a W3C recommendation. We examine the usage of LDS in the VC data model in Section 2.4. We present the usage of LDS in our approach in Section 5.1; for an example see Listing 2.

2.4. Verifiable Credential (VC) data model

The W3C recently released the Verifiable Credential (VC) data model [47] as a recommendation for sharing verifiable claims. The VC data model focuses on modelling credentials and claims in particular. Various options for signature schemes are mentioned, including LDS among other non-linked-data schemes. We summarize the most important aspects of the VC data model to our understanding from [47].

The VC data model is composed of three main components: the claim, the credential (metadata) and the proof. These components are explained best with a simple example:

- The statement that Alice is a student of a university is called a *claim*.

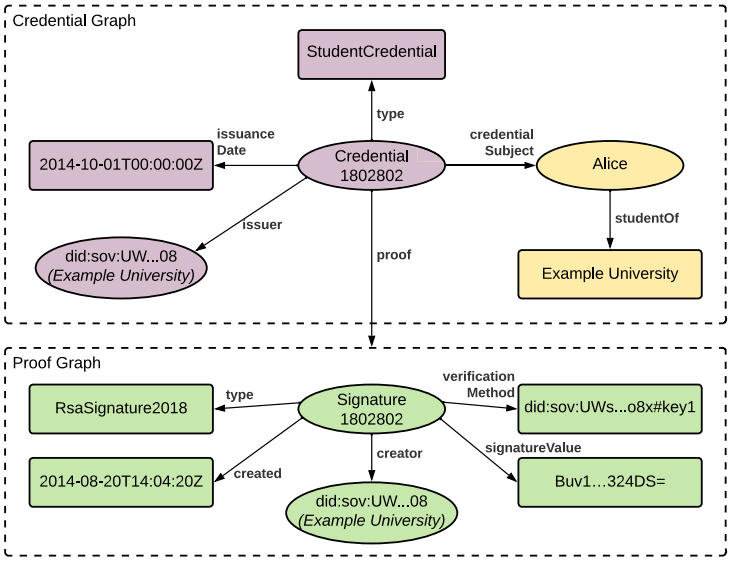

Fig. 1. A verifiable credential and its components in the style of [47]: *claim* (yellow), *credential (metadata)* (rose) and *proof* (green).

- The university accrediting Alice being a student is called a *credential*.
- The university's signature on the credential is called a *proof*.

All three elements of claim, credential and proof form a VC. The VC data model is defined in JSON-LD to allow for adoption of in both JSON and RDF-based implementations.

Regarding the data model from an RDF perspective, the VC data model relies on RDF datasets, i.e. a composition of RDF graphs. Claims and associated credential metadata form the so-called *credential graph*. The credential graph links to the so-called *proof graph* formed by the credential's proof information, i.e. the digital signature. In the credential graph, claims represent the actual statements which the VC is created for. The credential metadata represent information on the credential itself, e.g. which agent or entity issued the credential, when the credential was issued, which type the credential is of and what the subjectof the credential is. The proof graph consists of all information associated with the digital signature, e.g. an LDS, that asserts the credentials validity. Figure 1 illustrates the structure of a VC: With the VC data model, the credential links to the proof, i.e. the signature. There does not exist a link from the signature to the data that is signed.

Listing 1 shows a VC in JSON-LD format using a Linked Data Signature. Just from looking at the data, it is impossible to directly extract which algorithms for canonicalisation, digest and verification should be used. These are defined by the type of the signature, i.e. Ed25519Signature2020, and need to be looked up at the corresponding specification.

2.5. Analysis of the web

The Web is comprised of interlinked Web resources, e.g. human-understandable Web pages and machine-readable Web documents like RDF graphs and datasets. In this paper, we focus on a document graph on the Web as specified in Definition 5.

2.5.1. Hubs & authorities

The notion of hubs and authorities have originally been defined in the context of the HITS algorithm [32] for calculating a relevance score for pages and documents on the Web. The definition of hubs, authorities and their corresponding scores are adapted from the original definitions found in [32]:

```
{
"@context": [
            "https://www.w3.org/2018/credentials/v1",
            "https://example.org/education/v1"
],
"type":        [ "VerifiableCredential", "StudentCredential" ],
"name":          "Student ID Card",
"issuer":        "did:sov:UWsdevKh4dgjjv5SFbRo8x",
"issuanceDate": "2022-02-04T17:31:41Z",
"credentialSubject": {
            "name":       "Alice",
            "studentOf": "Example University",
            "type":     [ "Student", "Person" ]
},
"proof": {
    "type":                "RsaSignature2018",
    "created":             "2014-08-20T14:04:20Z",
    "creator":             "did:sov:UWsdevKh4dgjjv5SFbRo8x",
    "verificationMethod": "did:sov:UWsdevKh4dgjjv5SFbRo8x#key1",
    "signatureValue":      "Buv1y5q2M3XKaxup3tmzN4DRFTLVqpLMweBrSxMY2x
                            HX5XTYVQeVbY8nQAVHMrXFkXJpmEcqdoDwLWx24DS="
}
}
```

Listing 1. Example of a verifiable credential

Definition 6 (Hubs, Hub Score)**.** Hubs are documents that reference highly ranked authorities. Let the hub score be

$$hub(u) = \sum_{v \in \mathcal{U}_{\text{out}}} auth(v)$$

where $\mathcal{U}_{\text{out}}$ is the set of all documents which u links to.

A document's hub score is the sum of all the authority scores of documents it points to. Not only does the hub score take the number of referenced documents into account, but it also reflects the degree to which a document references authorities, i.e. other documents that are referenced by other hubs. In other words, highly ranked hubs exhibit a considerable overlap in the sets of documents they link to.

Definition 7 (Authorities, Authority Score)**.** Authorities are documents that are referenced by highly ranked hubs. Let the authority score be

$$auth(u) = \sum_{v \in \mathcal{U}_{\text{in}}} hub(v)$$

where $\mathcal{U}_{\text{in}}$ is the set of all documents which link to u.

A document's authority score is the sum of all the hub scores of the documents that point to it. Not only does the authority score take the number of documents referencing the document into account, but it also reflects the degree to which a document is referenced by hubs, i.e. other documents that reference other authorities. In other words, highly ranked authorities exhibit a considerable overlap in the sets of documents that reference them.

2.5.2. PageRank

The PageRank [39] is a score that assesses the popularity of Web pages. The idea of the PageRank can be formulated as *"Popular pages link popular pages"*. The definition of the PageRank is adapted from its original definition as presented in [39] and its calculation via the power method [4]:

Definition 8 (PageRank)**.** The PageRank of a document u is calculated iteratively until convergence by

$$r_{i+1}(u) = \frac{1-d}{n} + d \sum_{v \in \mathcal{U}_{\text{in}}(u)} \frac{r_i(v)}{\deg^+(v)}$$

where d is the damping factor of the random surfer, $\mathcal{U}_{\text{in}}(u)$ the set of all documents which link to u and i the number of iteration. The power method provides an alternative to the iterative method:

$$r = \left(dM + \frac{1-d}{n}E\right)^i z = \mathcal{M}^i z \quad (i \longmapsto \infty)$$

where d is the damping factor of the random surfer, M the transition probability matrix, i the number of iterations and z a probability distribution, e.g. the identity vector.

In each iteration the PageRank of each document is updated based the PageRank of documents linking to the document. The random surfer is introduced to solve the drainage of PageRank by sinks, i.e. documents with no out-going links [39]. The random surfer jumps from any document to another document in the graph based on a predefined damping factor d. Additionally, the random surfer jumps from a document without outgoing links to any other document in the document graph, by default with uniform probability.

2.6. Analysis of agent behaviour on the web

2.6.1. Watts–Strogatz model

The Watts–Strogatz model [50] generates graphs that exhibit the so-called *small-world properties* of short average path lengths and high clustering of graph nodes. One representative of such small-world networks are social agent networks [50]. The Watts–Strogatz model is thus used in Section 9.3 to create agent networks based on which agent interaction is modelled in the evaluation simulations.

The small-world networks generated by the Watts–Strogatz model are a type of network whose structural properties, e.g. shortest path length or clustering, lie in between the properties of regular graphs and random graphs. In a regular graph, each node is connected to the same number of other nodes, while in a random graph, each node is randomly connected to other nodes. To achieve such structural middle ground, the model's graph generation is based on *rewiring* connections between nodes of an initially regular graph. Dependent on a specified probability, each connection between two nodes is rewired.

To create a network using the Watts–Strogatz model, the number of nodes, their number of connections and the rewiring probability need to be specified. A regular graph is generated with the specified number of nodes where each node is connected to the specified number of other nodes. Then, the rewiring procedure is conducted per connection between two nodes in the graph. The thus created graph can be interpreted as a social network of agents, where the graph nodes represent agents and the connections between nodes represent agent interaction, e.g. being friends or chatting with each other. Using such networks, agent interaction can be modelled.

2.6.2. Basic concepts from Game Theory

We examine the behaviour of agents on the Web that is induced by our proposed approach. To analyse more complex agent behaviour, e.g. when agents interact repeatedly, we take inspiration from the analysis of repeated games in Game Theory. We use this technique in Section 9.1.3 to analyse agent behaviour when there is no obvious answer to how an agent may behave given a specified approach to trust scores. This section summarizes the foundations of repeated games in game theory, based on [42].

Agent behaviour is assumed to be rational with the goal of maximising a favorable payoff in a game. The way agents choose to behave within the rules of the game is defined as *a strategy*. Each agent has its own set of strategies. A game is thus defined as a set of agents and their set of strategies, i.e. their possibilities to play the game according to the rules. Each combination of strategies across all agents, i.e. a scenario that results from the combination of the agents' behaviour, is associated with an outcome for each agent. Each agent tries to maximise its payoff (also referred to utility) received by an outcome. The fact that an agent maximises its own payoff exclusively affects the

		Agent 2	
		C	*D*
Agent 1	*A*	(3, 1)	(2, 2)
	B	(3, 2)	(2, 3)

Fig. 2. A payoff matrix where agent 1 is indifferent between its strategies *A* and *B*, and agent 2 has a dominant strategy *D* over its other strategy *C*, resulting in the two Nash equilibria *(A,D)* and *(B,D)*.

decision making of all other agents. A state where no agent is able to receive a better payoff be deviating from its current strategy, given that the strategies of the others remain unchanged, is called a *Nash equilibrium*.

An agent's *dominant strategy* is a strategy that yields the best outcome for that agent independent of the other agents' strategies. There exists no incentive for that agent to deviate from its dominant strategy regardless of what other agents decide to do. If for each agent there exists a dominant strategy, the resulting Nash equilibrium is called a *Nash equilibrium in dominant strategies*.

A common tool to represent agents' payoffs, in a game of two agents, for each strategy they may choose provides the *payoff matrix*. Agent 1's possible strategies are denoted by the rows and agent 2's possible strategies are denoted by the columns. In each tile, the first element of the tuple denotes the potential payoff for agent 1 and the second element of the tuple denotes the potential payoff for agent 2. As illustrated by Fig. 2, dominant strategies as well as Nash equilibria can be identified by simple inspection.

In a game that is played once, called a *stage game*, all agents choose their strategy once and receive the corresponding outcome. A *repeated game* is a repetition of a stage game, where outcomes are added up. The agents need to take the impact of their current action on future actions of other players into account. Repeated games can be categorised into *finitely repeated games* and *infinitely repeated games*. In a finitely repeated game, the expected payoff of an agent's strategy is the sum of outcomes in each round. In an infinitely repeated game, the expected payoff of an agent is calculated by discounting future outcomes:

Definition 9 (Payoff in an infinitely repeated game)**.**

$$\text{payoff} = \sum_{t \geqslant 0} \delta^t u_i(x_t)$$

where δ is the discount factor, t the index of iteration and $u_i(x_t)$ the utility or payoff received from an outcome x_t.

3. Related research

Our approach aims to contribute integrity of Linked Data on the Web. Hence, we survey related work in the area of Linked Data integrity and Web analysis.

3.1. Linked data integrity

Integrity of Linked Data is typically verified by calculating and comparing cryptographic hashes of the underlying RDF graphs. Our approach relies on Hogan's algorithm [29] for canonicalising and hashing RDF graphs as it is able to handle blank nodes most gracefully. Various alternative algorithms for generating cryptographic hashes of RDF graphs have been presented [5,14,34,45,48].

Trusty URIs [34,35] aim to make digital resources verifiable, immutable and permanent by extending the usual URI scheme with a cryptographic hash of the resource as a URI suffix. A resource is directly verifiable from its Trusty URI as an accessing agent can already expect a specific content to be served from the hash in the URI. However, our approach builds on and extends the conceptional idea of Trusty URIs with digital signatures to *Signed URIs*.

Nanopublications [33] aim at publishing data on the Web as verifiable, immutable, and permanent digital artifacts. The approach uses Trusty URIs such that links among nanopublications contribute to data integrity. With only hash

values ensuring integrity of data, however, authorship of publications is not preserved. The centralised yet distributed nanopublication-server-network ensures publications' discoverability, permanence and immutability. However, in our approach, which is fully decentralised, each agent may host their own server, e.g. a Solid Pod, to self-sovereignly make their resources available at *Signed URIs*.

Similar in data model and serialization to Trusty URIs, a draft specification by the Internet Engineering Task Force (IETF) for *Hashlinks*,[7] cryptographically secured hyperlinks, aims to enable detection of unexpected changes to the resource. For computing the required hashes however, the application of the SHA-2 algorithm on the bytes of a resource representation is proposed, which easily considers two isomorphic RDF graphs different just because they have been serialised differently. Thus, in our approach, as a major difference, we apply the algorithm of Hogan [29] to produce RDF graph hashes that consider isomorphism.

The application of asymmetric cryptography to compute signatures of RDF graphs to secure communication among agents is presented in [31]. The integrity of messages are preserved through calculation of digital signatures over the message content. By adding the previous messages to the new message's content, a chain of iteratively signed messages is created in a local RDF graph. However, our approach builds on interlinked documents are distributed across the Web. Additionally, our approach considers confidentiality of data through access control in Solid Pods.

The W3C recently released the Verifiable Credential (VC) data model [47] as a recommendation for sharing verifiable claims. One possible signature scheme in the data model is the LDS, among other non-linked-data schemes. The VC data model focuses on modelling credentials and claims in particular whereas we concentrate more on the signature and less on content modeling. Thus, we do not use the VC data model in particular but use the more broadly applicable LDS for providing verifiability for any Linked Data and not just credential models. LDS are not yet standardized and, in fact, were in recent discussion: A proposal for standardization of LDS[8] has sparked fierce discourse within the community, albeit recognizing the need for standradized RDF canonicalization and hashing.

Resource Integrity Proofs data model [3] aims to enable hash-based integrity verification of digital resources. The authors suggest to include integrity information like hashes and algorithms in the resource itself as meta-data. They also propose a combination with distributed ledger technology to create persistent proofs of existence of resources stored outside of the distributed ledger. Approaches following such a strategy are examined in [18,49] and for example applied in a simple supply chain use-case in [8,9]. In contrast, our approach does not use a distributed ledger to achieve data integrity. Instead, our approach relies on digital signatures in URIs and their transitive nature in a distributed document graph on the Web.

3.2. Web analysis

In the realm of Web analysis, and as mentioned in Section 2.5, PageRank [39] and HITS [32] are two well known methods for calculating a relevance score for pages and documents on the Web. More specifically, they analyse the link structure of the graph that is formed by interlinked Web pages. Our approach founds its trust scores on the notion of hubs and authorities from [32].

An alternative algorithm for dynamic and collaborative computation of page-rank-based scores of Web pages distributed in a P2P network is presented in [41]. It is extended in [40] with statistical methods for detecting and accounting for malicious agent behaviour during the score calculation. An open Web is assumed that lacks mechanisms to ensure data integrity or confidentiality unlike to our approach that allows for both integrity verification and privacy of information.

A general framework for propagation of trust and distrust in a directed graph of entities is introduced in [25]. Prediction of trust scores is based on individual trust scores for and of entities. The framework focuses on a reputation network of agents with individual scores to quantify trust in other people. Only focusing on trust as a given, underlying mechanisms to achieve such trust are not examined. In contrast, our work provides a technical approach to achieve trust in the integrity of documents on the Web.

[7] https://tools.ietf.org/html/draft-sporny-hashlink-06

[8] https://lists.w3.org/Archives/Public/semantic-web/2021Oct/0020.html

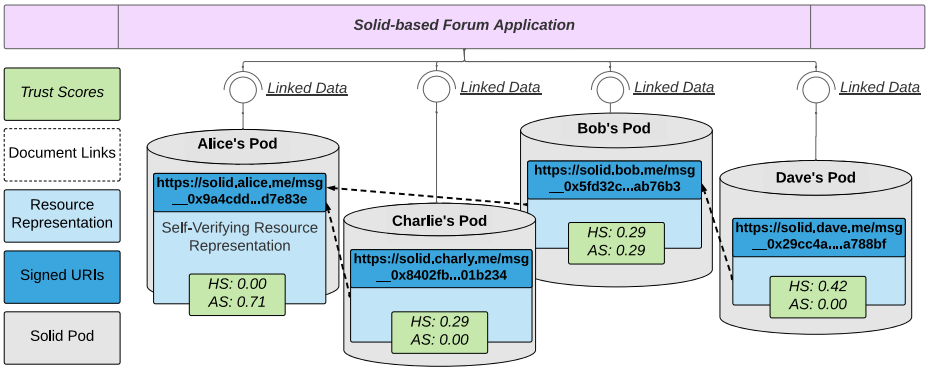

Fig. 3. An illustration of the system architecture: user's store their interlinked resources in their corresponding pods. The solid-based application consumes the RDF resource representation via HTTP as linked data. Each document exhibits a hub score (HS) and authority score (AS), calculated on the transitive closure of the document graph. Assume for this figure that Dave joined the group chat of the others and comments the latest post, created by Bob in reply to Alice, ignoring Charlie's post.

Analysing the Web graph, [24] examines the robustness of the Web of data. More specifically, a method to analyse and improve the physical and semantic structure of the Web is presented. We took inspiration for the evaluation from the method presented in [24], which laments that there are too few links on the Web. Our approach however incentivises the creation of links between documents, thus contributing to the robustness of the document graph. We use their method of targeted attacks is used to quickly disconnect the document graph during the evaluation of the proposed approach.

4. Exemplary architecture and terminology

Assume a decentralised social network on the Web, e.g. a Solid-based micro-blogging application where agents create and interlink posts of social banter, as illustrated by Fig. 3. These posts are interlinked by nature of the underlying social application: A post may reference another post as a comment or a simple reference. Each agent stores its posts in its own Pod where access control on the posts can be defined as desired by the agent. Not all posts may be available to everyone, access may be restricted to a certain group of agents.

A post is a resource with a self-verifying representation (presented in Section 5): It is described in RDF including a Linked Data Signature and is available at a Signed URI, which includes the signature's value as a suffix. Upon dereferencing the Signed URI, the integrity of the resource and the corresponding creator can be verified. The resource is said to be modification-detectable, or *tamper-evident*.

A *modification* to a resource refers to the resource being updated "in-place", i.e. the representation is updated but it is still available at the same URI. With a *modification*, there is no statement made if the resource representation can still be verified. In contrast, with an *update* of a resource, the verification of its representation still yields a valid result. An *update* of a resource thus refers to a new resource with corresponding self-verifying resource representation being created.

Consider a group of agents forming a social agent network: Alice, Bob, Charlie and Dave. Alice, Bob and Charlie form a group where they share posts only they interact with, i.e. respond to. Dave is a close friend of Bob but does not know or interact with the others.

We call such assemblies of interacting agents *interaction channel of size n*: An interaction channel of size n is an abstract space where a group of n agents actively interacts with each other, e.g. create and interlink posts. An interaction channel may be comparable to an online chat room of n actively participating agents. In the remainder of this work, we distinguish between interaction channels of size two, i.e. a pairwise agent interaction channel, and interaction channels of size greater than two, i.e. a multi-agent interaction channel.

The number of passively consuming, i.e. posts reading but not creating, agents is irrelevant to the definition of an interaction channel. In other words, it does not matter to the interaction channel if the posts are publicly available

or are confidential, i.e. only readable by actively participating agents. Instead, this decision is a design choice of the social Web application (and ultimately of the agents themselves). We investigate the effects of the decision on confidentiality as an parameter in our final evaluation of our approach (presented in Section 9.3).

Assume Charlie creates a new post in the multi-agent interaction channel with Alice and Bob. Agents are notified about a newly created document using Linked Data Notifications (LDNs). Of course, other means of advertising a newly created resource are also possible but not applied here. Independent from such options, resources are discoverable by Graph Traversal.

As Charlie's post is a comment to a post created by Alice, Charlie's post includes a link to Alice's post. We refer to such a link, which exists because of the natural linking of resource contents, as a *content-based link*. As the created resources have self-verifying representations, links among resource contribute to their integrity (presented in Section 6): In our case, Alice would update her resource, then, she would have to ask Charlie to also update her resource, such that the update becomes undetectable. The more resource representations link to a particular resource, the more resource representations need to be updated and, presumably, the more users would need agree to do so. The social Web application may desire an even more interlinked Web of resources in order to protect against unwanted undetectable resource updates (presented in Section 6.2). To this end, it may require a specific number of links to be included in any resource. The application may even impose additional constraints on a resource, e.g. which resources to link or which shape it should fit. If a resource does not satisfy the application's requirement it is considered untrustworthy by the application and thus ignored. We focus on the requirement of additional links, which we call *application-based links*. To put a number on the integrity, i.e. trustworthiness, of resources, the social Web application may consider calculating a score, which we call *trust score* (presented in Section 7). It captures on how well the application deems the resource to be secured against unwanted undetectable updates. We assume that agents using the application would like their resources to be considered well-secured, i.e. trustworthy. Based on that assumption, agents may include additional links in their resources that do not necessarily contribute to the content of the resource but mainly to the resource's scores as considered by the application used. With these additional links, agents aim to increase the trust scores of their resources (presented in Sections 9.1 and 9.2). We thus call such links *score-based links*. We investigate the effects of an application deciding to simply rely on content-based links or to apply trust scores as an parameter in our evaluation (presented in Section 9.3).

As the agents continue to collaborate on the social Web application, a Web of resources with self-verifying representations is created.

5. Self-verifying resource representation

We introduce self-verifying resource representations as we presented them in [10]: A resource is identified by an URI. Dereferencing that URI yields an RDF representation of the corresponding resource. A self-verifying resource representation consists of two components:

1. an RDF resource representation, which must include an LDS
2. a Signed URI, which the resource is made available at

Such self-verifying resource representation is also called modification-detectable, or *tamper-evident*: Any modification to the resource's representation becomes evident by comparing the hash calculated from the retrieved representation and the hash derived from the signature provided in the Signed URI. At the same time, the agent that created the corresponding LDS is authenticated.

5.1. Linked data signature using RDF-star

To sign the RDF representation of a resource, a Linked Data Signature (LDS) is created:

1. the RDF representation is canonicalised, e.g. using Hogan's algorithm [29].
2. the message digest of the canonical RDF representation is calculated, e.g. using SHA-256 [38]
3. the hash of the canonical RDF representation is signed, e.g. using the Elliptic Curve Digital Signature Algorithm (ECDSA) [2] with curve P-256.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dc:   <http://purl.org/dc/terms/> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix sec:  <https://w3id.org/security#> .
@prefix sioc: <http://rdfs.org/sioc/ns#> .

_:signature a sec:Signature;
    dc:created "2022-02-04T10:39:31.981Z"^^xsd:dateTime;
    dc:creator <http://solid.charly.me/profile/card#me>;
    sec:publicKey <https://solid.charly.me/public/keys/main#key>;
    sec:canonicalizationAlgorithm "Hogan2017";
    sec:digestAlgorithm "SHA-256";
    sec:digestValue "9fcefe...74cc8e"^^xsd:hexBinary;
    sec:signatureAlgorithm "ECDSA";
    sec:signatureValue "8402f...01b234"^^xsd:hexBinary;
    sec:proofOf
     << <#post> sioc:content     "My Blog post about Solid Pods." >>,
     << <#post> sioc:has_creator <https://solid.charly.me/profile/card#me> >>,
     << <#post> sioc:reply_of    <https://solid.alice.me/msg__0x9a4c...e83e> >>.

<#post> sioc:content     "My Blog post about Solid Pods.";
        sioc:has_creator <https://solid.charly.me/profile/card#me>;
        sioc:reply_of    <https://solid.alice.me/msg__0x9a4c...e83e>.
```

Listing 2. Example of a self-verifying resource representation, in our example available at *https://solid.charly.me/msg__0x8402f...01b234*. It is comprised of the asserted triples (see `<#post>`) and its corresponding linked data signature (see `_:signature`) using RDF-star. Hash and signature values are abbreviated for readability

The specific algorithms used are to be documented in the LDS, as illustrated by Listing 2, such that a verifying agent is able to verify the LDS.

LDS are listed as one possible signature scheme in the Verifiable Credentials (VC) data model [47], which is heavily influenced by the JSON-LD syntax. Examining, from a pure RDF perspective, how LDS are defined and used in the VC data model, we took issue in

(a) the use of RDF datasets, which have underspecified semantics [51], and
(b) the assertion of claimed statements, which are a result of the use of Named Graphs [15]

To avoid these issues, we model LDS using RDF-star: RDF-star allows for quoting triples, i.e. referencing without asserting. We argue that the LDS only provides meta-information on the signed triples and does not necessarily have to assert those triples. This may be useful, for instance, when the truth-value of the signed triples can change over time: a digital student id card becomes invalid once the student graduates. The signature, however, is always valid as it makes no statement about the truth-value of the signed triples.

5.2. *Signed URIs*

A *Signed URI* is a URI that includes the hexadecimal value of the resource representation's signature as a suffix, delimited by two underscores, e.g. for Listing 2:

https://solid.charly.me/msg__0x8402f...01b234

The suffix ends the hierarchical part of a URI that does not have a query part. It is similar to the idea of *Trusty URIs* [34,35] in that the URI's suffix can be used for integrity verification. But in contrast to Trusty URIs, instead of using the hash, the signature value is appended to the URI. For verification,

1. we first remove the signature suffix from the base URI prior to resolving relative URIs in the RDF graph.

2. We then check if the signature value is verifiable using the quoted triples and the specified algorithms.
3. Next, we check if all triples quoted by the Linked Data Signature are in fact asserted.

The latter check is optional in cases where assertion of signed triples is irrelevant.

An agent accessing a resource at a Signed URI can not only verify the content of the RDF representation from its Linked Data Signature but also expect that specific content signed by a specific creator to be served. Signed URIs allow agents to reference a specific resource in a specific state from another resource's representation.

6. A web of self-verifying resource representations

In a Web of self-verifying resource representations, links among representations contribute the integrity of the resources' representations as those links include the signatures of the corresponding resources. When a resource's representation changes, its signature value changes, the existing Signed URI mismatches, and the modification becomes detectable. For an undetectable update of a resources' representation, a new Signed URI is to be created and all resources that transitively reference the original resource have to be updated with respectively. In turn, those resources need to be updated with new Signed URIs as well and so on. Since on the Web, resources are typically under control of many different users, all those other users whose resources link to the original resource have to agree on such updates. Otherwise, *some* evidence will remain.

Thus, the more interconnected resources are, the more resources need to be iteratively updated. Additionally, the more agents are involved, the more agents need to be convinced to put up with the effort to update their resources. This way, contributions to such a Web contribute to preserving the integrity of existing self-verifying resource representations.

The Web of resource representations is a directed graph, where the resources[9] are the vertices and the links between the resources are the edges of the graph. This graph is referred to as the *document graph* as specified in Definition 5. Remember that the terms *document*, *information resource* and *information resource representation* are used synonymously (as mentioned in Section 2.1.1).

6.1. The document graph

Using self-verifying resource representations, this document graph implicitly contributes to the integrity of its documents by making them discoverable via graph traversal and subsequently verifiable for any agent. The document graph forms a directed acyclic graph (DAG). No cycle exists in the document graph; consider a trivial example: Document (a) is referenced by document (b) by a link using (a)'s Signed URI. So, (b)'s signature includes (a)'s signature. If document (a) was to reference (b), (a)'s signature would include (b)'s signature and, transitively, its own signature. This is not probable as this would indicate a collision in the hash or signature functions.

The documents can still be modified or deleted by the controlling agent. When an agent is to update a document, the modification of the existing content entails an update of the Signed URI. To do so unnoticed by other agents is only possible when the original document has not yet been referenced by other documents from other agents as agents traversing the graph expect a specific content when dereferencing the original Signed URI. If the document already has been linked from another agent's document, it is not possible for the first agent to modify its document unnoticed. Instead, the agent needs to create a new document and truthfully announce it as an updated version. It is then at the decision of agents whose documents reference the original to reference the new version in their documents accordingly or to keep the original reference. The iterative nature of document updates becomes apparent.

Undetectable document updates remain possible if and only if all agents whose documents reference the original document directly or transitively update their documents accordingly to reference the respective newly created documents. Moreover, all involved agents need to delete their original documents such that no trace of the original document sub-graph remains. If only one agent keeps one document that includes a reference to some original document, and if the agent is able to provide a replica of that document, the agent can prove that there has been an iterative update process.

[9] More specifically, their corresponding information resources.

6.2. Social applications on such a web

With self-verifying resource representations, any resource modification or update is detectable as long as at least one agent decides to not delete the references to the original representation. Only if all agents involved agree to the update, no evidence of the original resource may remain. Considering our exemplary application environment, i.e. a typical public online forum where agents interact in multiple public groups (see Section 4), this would mean that if all agents of one public group chat agree to an update, no evidence of the original messages persists.

The applications will abstract away most of the technicalities regarding a resource update, even a agreed non-evident one. For example, regarding the simple publication of a document at a *Signed* URI, the user will not notice at all that there is some magic happening in the background that ensures the document's integrity. Moreover for a document update, the application will also handle and abstract away setting links to the new and links to the old document appropriately and as desired. For a user, such details regarding publication and updates are hidden.

While such agreed-upon non-evident updates may be acceptable for some applications, others may not desire such behaviour due to stricter data integrity requirements. In such applications, preserving the integrity of resources is the overall objective. To this end, it may require a specific number of links to be included in any resource representation. The application may even impose additional constraints on a resource representation, e.g. which resources to link or which shape it should fit. If a resource representation does not satisfy the application's requirement it is considered untrustworthy by the application and thus ignored. We focus on the requirement of additional links, which we call *application-based links*.

As a dense document graph contributes more to the integrity of the documents than a sparse document graph, an application may impose an additional semantic constraint on the application-based links: The documents referenced by the application-based links should be "unrelated" to the document at hand. In our example, a post in a specific interaction channel should reference posts from other interaction channels as it is not directly related to those other posts.

By connecting otherwise unrelated documents, the number of agents involved with a specific document is potentially increased. With more documents connected and more agents involved, non-evident document updates become more difficult: All references of the original need to be discovered iteratively by the involved agents. The document graph needs to be searched "upstream" for incoming references of any document to be updated. In large-scale document graphs, the discovery of all such documents and references may become impractical. In addition, the corresponding agents need to be requested to update their documents accordingly. Moreover, some of these agents may be indifferent towards the update process, i.e. those agents who are involved due to the application-based links. Since updating their documents accordingly and subsequently searching for and notifying all other referencing agents would impose quite the effort on them, these agents have *ceteris paribus* no incentive to update their document. By introducing application-based links, the effort for non-evident document updates, i.e. the chance that some evidence of that process remains, is increased. To put a number on the achieved integrity, i.e. trustworthiness, of resources, the social Web application may consider calculating a score, which we call *trust score*.

7. Quantifying integrity of resource representations on the web

The following terms regarding document integrity are introduced: A document's *integrity preservation* describes how well its integrity is preserved by the document graph. A document's *integrity contribution* describes how much it contributes to the integrity of the document graph. Subsequently, the quantification of document integrity refers to measuring the effort needed for a document update in terms of number of documents to update. Due to the iterative nature of digital signatures, the transitive closure G^+ of a document graph G is considered.

Definition 10 (Integrity Preservation, Integrity Contribution)**.** Document u's in-degree $\deg^-(u)$ in G^+ is a measurement of u's integrity preservation. Document u's out-degree $\deg^+(u)$ in G^+ is a measurement of u's integrity contribution.

The quantification of document integrity provides a basis for trust in documents, where trust refers to the corresponding document's security against undesired access, modification or deletion as perceived by an agent. When

a document's integrity is highly preserved, it can be assumed that the document is unlikely to be updated without evidence and hence can be relied on. The integrity preservation of a document can serve as an indicator of how trustworthy an agent would consider that document.

We introduce the notion of *trust scores*:

Definition 11 (Trust Score, Trustworthiness, Gravitas)**.** Trust scores are document-specific values that provide a basis for interpretation regarding trustworthiness and gravitas of that particular document. The trustworthiness of a document describes how well that document may be trusted by an agent. The gravitas of a document describes the document's influence on the trustworthiness of a second document.

In general, trust scores provide a qualitative ordering instead of a quantifiable measurement of document integrity. While quantitative measurements of document integrity can be interpreted as trust scores, trust scores may additionally rely on other properties. These additional properties may not impact the integrity of a document directly but may be relevant to the integrity of the document graph, e.g. the interconnectedness of documents within the graph.

We introduce trust scores based on hubs and authorities and a practical heuristic strategy to optimize the scores. We will show in Section 9.1 that optimizing hub- and authority-based trust scores results in an incentive for agents to constructively contribute to the document graph and its integrity preservation. In Section 9.2, we provide experimental evidence that the proposed optimization heuristic outperforms alternative heuristic strategies.

7.1. Hub- and authority-based trust scores

Adapting the notion of hubs and authorities from [32], hub score and authority score can be interpreted as trust scores for resources with self-verifying representations on the Web.

Recall the definition of the hub score (Definition 6): A document's hub score is the sum of all the authority scores of documents it points to. The hub score of a document can be interpreted as to how much it contributes to preserving the integrity of the document graph, receiving more weight from highly ranked authorities. The hub score of a document can hence be interpreted as its gravitas. Not only does the hub score take the number of referenced documents into account, but it also reflects the degree to which a document references authorities, i.e. other documents that are referenced by other hubs. In other words, highly ranked hubs exhibit a considerable overlap in the sets of documents they link to.

Recall the definition of the authority score (Definition 7): A document's authority score is the sum of all the hub scores of the documents that point to it. The authority score of a document can be interpreted as to how well its integrity is preserved by the document graph, receiving more weight from highly ranked hubs. The authority score of a document can hence be interpreted as its trustworthiness. Not only does the authority score take the number of documents referencing the document into account, but it also reflects the degree to which a document is referenced by hubs, i.e. other documents that reference other authorities. In other words, highly ranked authorities exhibit a considerable overlap in the sets of documents that reference them.

Due to the iterative character of digital signatures, hub and authority scores are calculated on the document graph's transitive closure. Figure 4 illustrates the distribution of scores within a document graph.

As documents that are deep in the DAG exhibit a significant overlap in the set of documents referencing them on the transitive closure of the document graph, they exhibit a high authority score. As such documents do not link many other documents, their hub score is low. Inversely, as documents that have been recently been added exhibit a

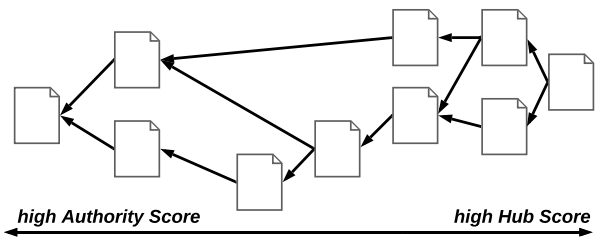

Fig. 4. A document graph with qualitative distribution of the trust scores calculated on its transitive closure.

significant overlap in the set of documents they link to on the transitive closure of the document graph, they exhibit a high hub score. As such documents are not linked to by a lot of other documents, their authority score is low.

This interdependence of hub and authority scores fosters interconnectivity among the documents. It is also the reason why a document that is referenced by a lot of documents with a low hub score does not necessarily receive a high authority score. Since the referencing documents are not well connected with other documents, and hence exhibit the low hub score, the resulting authority score of the referenced document will be rather low. The interpretations of gravitas and trustworthiness based on hubs and authorities can thus be described as follows: A document's gravitas is dependent its own interconnectedness and the interconnectedness of the documents it links to. A document's trustworthiness is dependent on the interconnectedness of the documents it is linked by. Or short: *"Trust is based on the interconnectedness of documents."*

8. Score optimisation and agent behaviour

Trust scores allow agents to determine which resources they would like to rely on in an application context, i.e., which resources are more trustworthy than others to be relied. When a resource is considered not trustworthy enough, the resource would simply not be considered for usage by an agent. Trust scores can thus serve as a mechanism to application designers for formulating incentives regarding agent behaviour: When agents rely on trust scores, agent behaviour may be influenced towards optimising these trust scores. In an application context for example, an agent would like its documents to be highly trustworthy such that other agents using the application consider the documents trustworthy enough to be used. The agent would like the trustworthiness of its documents to be reflected by the trust scores specified by the application. To this end, an agent would exploit the trust scores as much as possible to make its own documents seem most trustworthy.

Of course, an agent can always choose to rely on other trust scores than the ones defined by the application or to ignore them entirely. However, some trust scores may be more suitable than others in specific application contexts. Application-defined trust scores establish a common ground for agents interaction based on the recommendation of the application designer. Moreover, when an agent relies on different trust scores than the rest of the agents, the agent would optimise its chosen trust scores instead of the commonly used ones. This will reduce the trustworthiness of its documents from the perspective of the other agents. When an agent chooses to ignore the trust scores of its own documents, it does only effect the trust scores of its own documents. It does not have negative effects on the trust scores of other agents.

Application designers can leverage the characteristics of trust scores to incentivise participation and contribution to the integrity of and trust in their application. Assuming agents that optimise the trust scores of their documents to maximise their trustworthiness, agent behaviour is influenced depending on which trust score is chosen by the application designer.

For an agent, the main question regarding the optimisation is the following: *"Which document or documents should a new document reference, such that its own trust scores, i.e. trustworthiness or gravitas, are high."* With a high trustworthiness of the just published document, the agent has already reached its goal. With a high gravitas of the just published document, the agent can include a reference to another document such that that document's trustworthiness is increased.

This allows an application designer to choose the application's trust scores such that contributing to the integrity and trust in the document graph leads to high trustworthiness or gravitas of newly created documents. Subsequently, an optimisation rule can be deducted from the choice of trust scores. That is, which resources an agent should reference in a new resource to maximise the trust scores. Such references are thus a specific type of *application-based links* that is referred to as *score-based links*.

8.1. General score optimization

For an agent, the main question regarding score optimization is the following: *"How can I make my documents regarded as trustworthy (enough), i.e. high authority score, for other agents to rely on them?"* With hub- and authority-based trust scores, however, it is not possible to have a newly published document with high authority

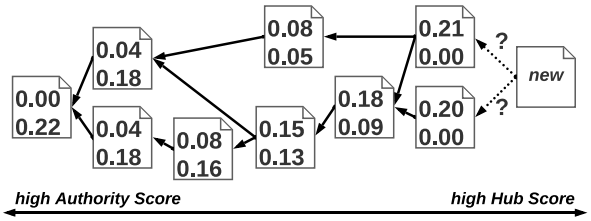

Fig. 5. A document graph with trust scores calculated on its transitive closure. The top value indicates the hub score, the bottom value indicates the authority score of a document. For a new document, it remains the question which existing documents to reference best.

score as no incoming links exist. Instead, with a high gravitas of the just published document, the agent would be able to increase the trustworthiness of already existing documents referenced from the new document. When the gravitas of a new document is maximised, it contributes most to the trustworthiness of referenced documents.

Looking at Fig. 5, it is quite obvious that documents with an in-degree of zero, that reference all the other documents transitively, exhibit a high hub score. Subsequently, those documents contribute most to the authority score of the documents deeper in the DAG.

For an agent publishing a new document, this means that the new document should reference as much other documents as possible. This way, it contributes most to the authority score of the referenced documents, and especially, to the referenced documents of the agent itself. The easiest way to achieve this, is by simply referencing all documents with an in-degree of zero from the new document. Thereby, the new document references all documents in the DAG transitively. In other words, all documents in the transitive closure of the document graph are referenced.

8.2. Heuristic optimization in restricted environments

However, including references to all documents with an in-degree of zero may not an option, e.g. when considering a large-scale document graph like the Web. The decision problem arises *which document or documents to reference best*; based on the information available to a particular agent, i.e. the documents that the agent has access to. Similarly, Web applications, e.g. our exemplary forum application from Section 4, may require resources to have a specific shape, where to number of outgoing links is restricted. For example, an application designer wants to leverage the integrity preserving nature of Signed URIs but, at the same time, needs to balance this desire with the responsiveness and performance of the web application. To this end, the allowed number of outgoing links is limited to a use-case specific number. As a specific example: Ten links may be handled by the application in under 2 seconds which may be the maximum of response delay desired for the application.

As brute-forcing the optimal choice is infeasible at large-scale and simply not allowed in restricted application environments, we propose the *Additional Reachability Strategy (ARS)* as an optimization heuristic to iteratively choose the next best document to reference. Assume a new document v being added to the document graph. Iteratively, ARS chooses as next reference the existing document which provides the highest number of not yet reachable documents from v. Implicitly, already reachable vertices are excluded since they yield an additional reachability of zero. In Section 9.2, we compare ARS to other potential heuristic strategies and show that ARS outperforms it competitors.

9. Evaluation

We evaluate three aspects of our approach:

First, we evaluate our proposed approach to trust scores which induce agent behaviour towards optimising such scores. We assess the strategy that an agent would choose to *maximise the trustworthiness* of its own documents, given an approach to trust scores. We evaluate three possible options: Degree-based, PageRank-based and hub- and authority-based trust scores. We show that hub- and authority-based trust scores provide a dominant strategy and thus an incentive for agents to contribute to the document graph and its integrity preservation.

Second, we evaluate our proposed Score Optimisation Strategy for finding *the next best link* for a new document. We assess four decision heuristics, i.e. the Initial Max Hub Score (Init HS), the Iterative Max Hub Score (Iter

HS), the Initial Reachability (IRS), and the Additional Reachability (ARS). We compare the performance of these heuristics using four performance metrics, i.e. precision, recall, optimal set rate and optimal subset rate. We show that ARS exhibits significantly better performance than other heuristic score optimization strategies.

Third, we evaluate our approach in our exemplary application environment. We assess how the authority score distribution within the document graph changes when iteratively documents are deleted until the graph becomes weakly disconnected. We further examine extend to which the score distribution is able to recover upon reconnection of the graph. We introduce a model of our exemplary application to simulate an agent network in a social Web application based on the Watts–Strogatz model [50]. We present a method to simulate agent activity within the social application, i.e. creating and interlinking resources. We then artificially delete resources until the graph becomes disconnected. The agents make an effort to connect the graph again and continue their usual behaviour. We define two evaluation parameters which yield five simulation cases. On one hand, we examine the influence of access control, i.e. the case where resources in an interaction channel are visible and thus linkeable only be members of the interaction channel, and transparency, i.e. the case where resources in an interaction channel are visible and thus linkeable to any agent. On the other hand, we examine the influence of different link structures within the document graph, i.e. a case where only content-based links exist, a case where only score-based links exist and a case where both types of links exist. We show that authority scores in a document graph are able to recover from the graph becoming weakly disconnected, even in access-controlled environments.

9.1. Choice of trust scores based on the induced agent behaviour

Trust scores allow agents to determine which resources are more trustworthy than others to be relied on for usage in an application context. Subsequently, agent behaviour may be influenced towards optimising these trust scores. Trust scores can thus serve as a mechanism to application designers for formulating incentives regarding agent behaviour. The evolution of the document graph underlying an application can thus be described based on the induced agent behaviour.

We first present three options for trust scores. Then, we define our methodology for evaluation. And lastly, we provide the results of our analysis.

9.1.1. Mathematical modeling of trust scores

Three approaches to trust scores are examined: The first approach produces simple degree-based trust scores. The second approach is based on the PageRank [39]. The third approach is adapting the notion of hubs and authorities of [32]. The approaches to trust scores can be summarized by their motto as follows:

- Degree-based approach: *"Integrity is trust."*
- PageRank: *"Highly trusted documents reference highly trusted documents."*
- Hubs and Authorities: *"Trust is based on the interconnectedness of documents."*

An overview of the iterative formula for modelling trust scores according to the three approaches is provided in Table 1.

Degree-based approach When integrity preservation is the only factor influencing trust in information resources, the quantitative measures of resource integrity may be interpreted directly as trust scores. The in-degree of a document is then interpreted as the trustworthiness of the document. The higher the in-degree of a document, the more trustworthy that document is. The in-degree of a document is not affected by the in-degree of referencing documents but only the existence of the reference. Thus, the gravitas of a document is equivalent to the existence of a reference to another document, i.e. it is 1 if a reference exists and 0 if there no reference exists.

Table 1
Overview of mathematical modelling of trust scores (* iterative formula after convergence)

Trust Scores	Approaches to Trust Scores of a document u in G^+		
	Degree-based	PageRank-based (*)	HITS-based (*)
Gravitas(u)	1	$\frac{r(u)}{\deg^+(u)}$	$hub(u) = \sum_{v \in \mathcal{U}_{\text{out}}} auth(v)$
Trustworthiness(u)	$\deg^-(u) = \lvert U_{\text{in}}(u) \rvert$	$r(u) = \sum_{v \in \mathcal{U}_{\text{in}}(u)} \frac{r(v)}{\deg^+(v)}$	$auth(u) = \sum_{v \in \mathcal{U}_{\text{in}}} hub(v)$

PageRank As an alternative use case for the PageRank, [39] explicitly states that the PageRank may be an indicator of the trustworthiness of a web page. The idea is adopted for documents published on the Web with self-verifying representations. The interpretation of the PageRank as a trust score can be formulated as *"Highly trusted documents reference highly trusted documents"*. When a highly trustworthy document references another document, the gravitas of that document leads to additionally increased trust in the referenced document. Highly trusted documents contribute with their gravitas to the trustworthiness and thus gravitas of other documents. In the PageRank-based approach, the trustworthiness as well as the gravitas of a document is determined by its PageRank.

Recall the definition of the PageRank (Definition 8). From the iterative calculation formula, the converged PageRank $r(u)$ of a document u is interpreted as the trustworthiness and $\frac{r(u)}{\deg^+(u)}$ as the gravitas of document u. The gravitas of a document is reduced the higher the out-degree, i.e. the more other documents are referenced by that document.

The notion of a random surfer and its damping factor d is not directly compatible with the interpretation of the PageRank as a trust score. While for the popularity of a Web page it may be acceptable to assume that an agent randomly jumps to another web page, it is hard to find a reasonable interpretation of the random surfer regarding trust scores. Hence, when interpreting the PageRank as a trust score, the damping factor should specified such that the random surfer is prohibited from jumping to a random document in the graph. Then, the random surfer is only allowed to jump from a sink document to a random document in the graph. Since this is done with uniformly distributed probability, a base value is distributed to all documents in the graph. This can be interpreted as a base trust level that is present in the document graph, even on documents that have no incoming links. This way, the problem of sinks is solved and the PageRank calculation results in reasonable values. Accepting the interpretation of the random surfer as a base trust level, the PageRank may serve as a trust score. The PageRank reflects the trustworthiness of a document and the sum of its gravitas at the same time.

Hubs and authorities Adapting the notion of hubs and authorities of [32], hub score and authority score can be interpreted as trust scores for documents published on the Web with self-verifying representations.

Recall the definition of hubs (Definition 6) and authorities (Definition 7): A document's hub score is the sum of all the authority scores of information resources it points to. The hub score of a document can be interpreted as to how much it contributes to preserving the integrity of the document graph, receiving more weight from highly ranked authorities. The hub score of a document can hence be interpreted as its gravitas. A document's authority score is the sum of all the hub scores of the information resources that point to it. The authority score of a document can be interpreted as to how well its integrity is preserved by the document graph, receiving more weight from highly ranked hubs. The authority score of a document can hence be interpreted as its trustworthiness.

Obviously, hub scores and authority scores are coupled: The trustworthiness of a document is determined by the gravitas of documents referencing that document. Inversely, the gravitas of a document is determined by the trustworthiness of other documents referenced by that document.

This interpretation of gravitas may seem counterintuitive at first glance because a document's own trustworthiness does not seem to influence its gravitas. Yet, the hub score of a document is dependent on the authority score of the referenced documents which in turn is dependent on the hub score of the documents referencing them. The hub score of a document is implicitly dependent on other hubs referencing the same documents. Hence, highly ranked hubs exhibit a considerable overlap in the set of documents they link to. The interdependence of the scores therefore fosters interconnectivity among the documents.

Referencing a lot of documents alone does not automatically make a document a highly ranked hub. It is referencing a lot of documents that other hubs reference as well. Only if a document is well interconnected with the other documents, it is able to aggregate the enough gravitas to contribute considerably to another documents trustworthiness. Similarly, a document referenced by one hub is also not automatically a highly ranked authority. Only if it is referenced by more other hubs, its authority score will increase. [32] describes the behavior of hubs mutatis mutandis as *"Hubs pull together authorities and allow to rule out unrelated pages of large in-degree"*.

The interpretations of gravitas and trustworthiness based on hubs and authorities can thus be refined as follows: A documents gravitas is dependent its own interconnectedness and the interconnectedness of the documents it links to. A documents trustworthiness is dependent on the interconnectedness of the documents it is linked by.

9.1.2. Methodology: Formal analysis

We assess the suitability of the proposed approaches by formal analysis. In particular, we examine the extend to which the modeling of the trust scores induces agent behaviour that contributes to the integrity of the document graph. We assume that an agent tries to maximise the trustworthiness of its own documents and, to this end, includes or excludes links to other documents in new documents it creates. This includes the assumption of rational agents.

The goal of the analysis to find a strategy for an agent to maximise the trustworthiness of its documents. We first try to find a dominant strategy by formally analysing the calculation formula of each trust score approach. Recall Section 2.6.2: A strategy is called dominant if it defines an optimal behaviour for an agent regardless of the behaviour of other agents. As all the agents have the same action set, and thus same strategies, finding a dominant strategy for an agent results in finding a Nash equilibrium in dominant strategies. In the case that we are not able find a Nash equilibrium in dominant strategies that way, we try to find any Nash equilibrium using a Game Theoretic approach: The iterative interaction of agents is represented as a repeated game where agents may collaborate to achieve an overall better outcome for oneself.

9.1.3. Results

Degree-based approach "Integrity is trust."

This approach relies fundamentally on Definition 10: Trustworthiness is only based on the incoming references of a document. Because the trust score of a document is not tied to its out-degree, there exists no score-inherent incentive to include outgoing links in a document. This means that there is no score-inherent incentive to reference other agents documents and to contribute to the integrity and trust of the document graph. Yet, from a game theoretic perspective, agents may want to collaborate and reference each others documents to increase the trustworthiness of their documents.

Formal Analysis. An agent can only influence the trustworthiness of its own documents by documents it has created itself as $\mathit{Trustworthiness}(u) = \deg^-(u) = |U_{\text{in}}(u)|$ is based on incoming links to a document. The agent is unable to directly influence the gravitas received from other agents' documents as the gravitas is constant, $\mathit{Gravitas}(u) = 1$. This is effectively an incentive for an agent to spam new documents with links to its own documents. Which, in itself is not a bad thing as it effectively contributes to the integrity of those documents. If the application recognises the spam, it may choose to exclude the documents from the calculation of the score for other agents. When other agents reference the "spam" documents, the documents may simply become part of the document graph. However, there is no incentive for agents to link to other agent's documents based on the definition of the trust scores alone.

Game Theoretic Assessment. Despite the lack of a score-inherent incentive to contribute to the integrity of the document graph, agents may want to collaborate and mutually reference each others documents to increase the trustworthiness of their documents. Consider the following example of two agents. It suffices to examine a two agent game as a game with more agents would just result in parallel pairwise versions of this game. In a nutshell, the game resembles a tit for tat between two agents. Each agent has already published a document, i.e. agent 1 published document 1 and agent 2 published document 2. Each agent is to publish a new document, i.e. agent 1 publishes document 3 and agent 2 publishes document 4. The agents can choose between the same set of strategies when publishing their documents. Strategy O refers to an agent only referencing its own existing document. Strategy B refers to an agent referencing both existing documents. Figure 6 depicts the contribution of the newly published documents to the trustworthiness of documents 1 and 2, i.e. (T_1, T_2), in a payoff matrix. It suffices to look at the minimal example of with one existing and one newly created document as additional existing documents or more newly created documents would only positively contribute to the agents payoffs. Of course, when considering n existing documents, one could image any number of placed links as a valid strategy thereby creating derivatives of the two strategies examined in our example. However, the overall intuition of the argument remains unchanged.

		Agent 2	
		O	B
Agent 1	O	$(1,1)$	$(2,1)$
	B	$(1,2)$	$(2,2)$

		Agent 2	
		O	B
Agent 1	O	(n_1, n_2)	$(n_1 + n_2, n_2)$
	B	$(n_1, n_2 + n_1)$	$(n_1 + n_2, n_2 + n_1)$

Fig. 6. The payoff matrix of the degree-based example ($n_i = 1$) with n_i being the number of existing documents of agent i.

Consider the following case: The new document of agent 1 references the existing documents of agent 1 and 2, i.e. agent 1 chooses strategy B, while the new document of agent 2 only references the existing document of agent 2, i.e. agent 2 chooses strategy O. The resulting payoff is $(1, 2)$. Document 1 receives a contribution to its trustworthiness of 1 from the link of document 3 and document 2 receives a contribution to its trustworthiness of 2 from the links of documents 3 and 4.

Both agents are indifferent between strategies O and B because they receive the same payoff regardless of their own action. Instead, their payoff depends only on the other agent's strategy. In the stage game, all agents are indifferent regarding their own strategy. Thus, we have no dominant strategy for an agent but found two Nash equilibria (O, O) and (B, B).

Considering a repeated game, which corresponds to agents repeatedly adding new resources to the document graph. The agents are able to collaborate to achieve an overall better outcome. In a finitely repeated game, it is best to collaborate to achieve the outcome of (B, B) in each period. If one agent deviates, the other agent will also deviate and simply stop referencing the other agents documents. This way, the deviating agent is effectively punished by the other agent. In an infinitely repeated game, the same punishment mechanisms exist Recalling the calculation formula (Definition 9), the expected overall payoff for each agent in the agreement for (B, B) in each period is

$$\text{payoff} = 2 + 2d + 2d^2 + \cdots = 2 + 2\frac{d}{d-1} \tag{1}$$

where d is the discount rate. When agent 1 deviates, its payoff is

$$\text{payoff} = 2 + d + d^2 + \cdots = 2 + \frac{d}{d-1} \tag{2}$$

because agent 2 will stop referencing agent 1's documents. Since the payoff when adhering to the agreement is higher for any discount factor d deviating is discouraged. Thus, it is always best to collaborate to achieve an overall better outcome, i.e. a higher trustworthiness of an agent's own documents. When an agent deviates, the agent is punished by the other agents through isolation.

Implications on Graph Evolution. It is best for agents to mutually reference each others documents such that the trustworthiness of all documents is increased. In a document graph, this translates to simply referencing all documents transitively. This is achieved by including references to all documents with an in-degree of zero in a newly published document.

To achieve the long term best outcome, agents have to check if the other agents adhere to the strategy repeatedly. That is, to check, if the other agents were to reference a particular resource and if they did so. This imposes quite the effort on each agent. Moreover, it might not be desirable or even possible to reference all documents with an in-degree of zero. Agents have to choose which documents to reference with their newly published documents. In this case, enforcement is not so straightforward because agents have to decide if a non-reference of their document is intentionally malicious or just imposed by the limitation of references. There is the possibility of the equilibrium in agent behaviour to collapse, and with it the constructive contributions to the document graph. The degree-based trust scores may become obsolete when every agent references their own document and/or other agents' documents randomly.

Conclusion. While Degree-based trust scores provide no score-inherent incentive and no dominant strategy for agents to collaborate and connect their documents, the incremental creation of the document graph allows for collaboration incentives to emerge. Degree-based trust scores exhibit the potential to incentivise agents to contribute to the document graph in a constructive manner. Moreover, degree-based trust scores provide the agents with an effective enforcement mechanism: If an agent does not cooperate, the agent is isolated and practically excluded from the agent network. However, limitations exist: Degree-based trust scores suffer from link farming. Although link farming attacks can be detected and the responsible agent isolated, it is an additional consideration for application designers to take into account. Moreover, when only a limited number of references is allowed or practically feasible, the equilibrium in agent behaviour may collapse leaving the construction of the document graph to a more random behaviour of agents. This may render degree-based trust scores obsolete. Hence, degree-based trust scores

can serve as the foundation for application design but their limitations as well as the possibility of them becoming obsolete need to be taken into account.

PageRank *"Highly trusted documents reference highly trusted documents."*

The trustworthiness of a document is only based on incoming references which can not be influenced by the publishing agent itself (except for link farming). The gravitas of a document is also increased with more incoming links. This provides an trust score inherent incentive for link farming. The gravitas of a document is reduced the higher the document's out-degree. This means that referencing other documents is discouraged and results in an self-focused agent behaviour. PageRank-based trust scores punish the inclusion of more outgoing links and thus punish contributing to the integrity and trust of the document graph.

Formal Analysis. Recall document u's $\mathit{Trustworthiness}(u) = r(u) = r_u = \sum_{v \in \mathcal{U}_{\text{in}}(u)} \frac{r_i(v)}{\deg^+(v)}$ after convergence and with the damping factor $d = 1$, the sum of Gravitas of documents linking to document v. Also recall document v's $\mathit{Gravitas}(v) = \frac{r(v)}{\deg^+(v)}$. An agent is not able to directly influence linkage of other agents' documents, but can only determine which documents are linked by its own documents. Thus, to maximise the trustworthiness of its own documents, the agent has to maximise the gravitas of its own documents. Then, the agent is encouraged to only link its own documents and not link any additional documents as with increasing $\deg^+(v) \longrightarrow \infty \Rightarrow \mathit{Gravitas} \longrightarrow 0$. Thus, it is a dominant strategy for an agent, to only link its own documents and not link any other agent's documents as this would "waste" a fraction of the document's gravitas. This fraction would be attributed to some other agent's document instead of the agent's own documents. Consider the following example of the graph adjacency matrices A and B and their corresponding transition probability matrices M_A and M_B. Note that the transition probability from documents with no outgoing links is uniformly distributed across all documents which is due to the random surfer from the definition of the PageRank (see Section 9.1.1).

$$
\begin{aligned}
A &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, & M_A &= \begin{pmatrix} 1/4 & 1/4 & 1 & 0 \\ 1/4 & 1/4 & 0 & 1 \\ 1/4 & 1/4 & 0 & 0 \\ 1/4 & 1/4 & 0 & 0 \end{pmatrix} \\
B &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, & M_B &= \begin{pmatrix} 1/4 & 1/4 & 1/2 & 0 \\ 1/4 & 1/4 & 1/2 & 1 \\ 1/4 & 1/4 & 0 & 0 \\ 1/4 & 1/4 & 0 & 0 \end{pmatrix}
\end{aligned}
\tag{3}
$$

With the power method, i.e. $r_{k+1} = Mr_k$ with r_0 an arbitrary initial vector and M the transition probability matrix until convergence, it follows that the PageRank of document 1 is lower in graph B than graph A due to the additional link from document 3 to document 2. So, when documents 1 and 3 belong to the same agent, it would be better for that agent to not include the link from document 3 to 2. We provide the formal proof for this intuition in Appendix A.

The PageRank does not provide an inherent incentive for agents to collaborate and contribute to the document graph in a constructive way. Moreover, it is a dominant strategy to not collaborate. The PageRank is thus not suitable as foundation for designing applications based on trust scores.

Hubs and authorities *"Trust is based on the interconnectedness of documents."*

To increase the trustworthiness of a document, it needs to referenced by preferably multiple highly ranked hubs. This way, agents are incentivised to create hubs such that they can increase the trustworthiness of their own documents. Contribution to the integrity and trust of the document graph is thus incentivised when considering hubs and authorities as the foundation of trust scores.

Formal Analysis. Referencing highly ranked hubs leads to a new document becoming a highly ranked hub itself. More formally, [32] shows that the hub score vector h is the principal eigenvector of the hub matrix $H = AA^T$ with A being the adjacency matrix of the graph. Moreover, the principal eigenvector h can be approximated using the

power method as the unit vector resulting from the convergence of

$$h = \left(AA^T\right)^k z \quad \text{with } k \longmapsto \infty \tag{4}$$

where k, the number of iterations, and z, the identity vector.

Using the power method, the reasoning behind the definition of hubs (and authorities analogously) becomes apparent. Consider two exemplary graphs and their corresponding adjacency matrices A and B. In the first graph, node 3 and 4 both link to node 1 and 2, while in the second graph node 4 additionally links to node 3.

$$\begin{aligned} A &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}, \qquad H_A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 2 \end{pmatrix} = AA^T \\ B &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \qquad H_B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 3 \end{pmatrix} = BB^T \end{aligned} \tag{5}$$

In a hub matrix H, the values on the diagonal H_{ij} $(i = j)$ indicate the out-degree of the node, while the other values H_{ij} $(i \neq j)$ indicate the number of nodes commonly linked to by node i and node j. So when using the power method, it becomes intuitive that nodes with a high out-degree as well as a high number of nodes commonly referenced by other nodes exhibit high hub scores.

As an example, consider nodes 3 and 4 and their corresponding rows in the hub matrices. Since both nodes link to the same nodes, i.e. 1 and 2, entries $H_{34} = H_{43} = 2$. Both nodes exhibit an out-degree of 2. In the second graph however, node 4 references node 3, thus having an out-degree of 3. Consider the hub scores of nodes 3 and 4 after the first iteration with an initial $z = (1, 1, 1, 1)^T$ after normalisation:

$$\begin{aligned} H_A z_0 = (0, 0, 4, 4)^T &\rightarrow z_1 = (0, 0, 1/2, 1/2)^T \\ H_B z_0 = (0, 0, 4, 5)^T &\rightarrow z_1 = (0, 0, 4/9, 5/9)^T \end{aligned} \tag{6}$$

In the first graph, both nodes exhibit the same hub score due to linking to the same nodes. Quite intuitively, the hub score of node 4 is higher in the second graph than in the first graph due to the higher out-degree. It is also higher than the score of node 3. This illustrates that a node linking to the same set of nodes as another node will have the same hub score as that node. Moreover, a node that links to the same set of nodes as another node and additionally links that note itself, will exhibit a higher hub score than that node. Subsequently, when considering transitivity, a node only needs to reference the particular other node to automatically achieve a higher hub score than that node. Considering the second graph of the example, node 4 only links to node 3 in the graph's transitive reduction while simultaneously achieving a higher hub score than node 3.

From this reasoning, we conclude the following lemma:

Lemma 1. *In the transitive closure of a DAG, a higher hub score is achieved when referencing a hub directly instead of only referencing the authorities referenced by that hub.*

We provide formal proof for Lemma 1 in Appendix B.

Game Theoretic Assessment. In the context of interpreting the hub score as a trust score, it follows from Lemma 1 that it is sufficient to reference existing hubs instead of authorities to achieve a higher hub score than existing hubs on the transitive closure of the document graph. In other words, contribution to the document graph is incentivised, and in particular referencing existing hubs. It follows that *all hubs should be referenced to ensure a hub score higher than any existing hub and thus impact on authority score of other documents*. Practically, that means referencing all available documents on the transitive closure of the graph, which is achieved by referencing all documents with no incoming links.

When an agent wants to increase the authority score and thus trustworthiness of its own documents, it is best to publish a new document that references many documents, preferably with a high hub score. Even when other agents choose to only reference specific documents, it still remains best for an agent to reference as many documents as possible. This strategy is independent from the behaviour of the other agents and as such a dominant strategy.

Implications on Graph Evolution. The high hub score documents are typically "newer" documents, i.e. documents with an in-degree of zero, that reference many if not all already existing documents. This means, that when a document is added to the document graph, it should reference many high hub score documents to become a hub itself and subsequently a valuable target to reference for new documents. This way, the documents will receive more incoming references as new documents are added, thereby gaining authority score. In turn, documents will slowly loose hub score as new documents are added because they link to more other documents and thus exhibit a relatively higher hub score. As the document graph evolves over time by incremental additions, a document will slowly lose its hub score but gain authority score instead. With each new document referencing a particular document, its gravitas is incrementally transformed into trustworthiness.

Interpreting hub and authority score as trust scores for gravitas and trustworthiness not only imposes a short term incentive to create a hub to increase the trustworthiness of own documents, but also provides a long term incentive to create a hub due to hubs being transformed to authorities as the document graph evolves. With the document graph incrementally growing, the trust scores maintain their meaningfulness. Hub and authority based trust scores incentivise agents to contribute to the document graph in a constructive manner which allows the document graph to evolve without its underlying trust scores becoming obsolete over time.

Even when the number of references for newly published documents is limited, the incentive remains to gain a high hub score. There does not exist an incentive for agents to deviate from their dominant strategy, unlike with degree-based trust scores.

Conclusion. The notions of hubs and authorities and their corresponding scores based [32] are well suited to be interpreted as trust scores in document graphs. Moreover, they provide an inherent incentive and dominant strategy for agents to contribute to the document graph in a constructive fashion keeping the graph and its underlying trust scores intact. Limiting the number of references of new documents does not affect agent behaviour when using hub and authority based trust scores.

9.1.4. Summary

Degree-based trust scores can serve as the foundation for application design but their limitations as well as the possibility of them becoming obsolete need to be taken into account. PageRank-based trust scores provide no incentive for agents to collaborate and contribute to the document graph constructively. Thus, they are not suited as the foundation for trust scores. Only Hub and Authority scores are well suited as trust scores. They provide an inherent incentive to contribute to the document graph constructively, while not suffering from any limitation that are present for the other approaches. Based on this assessment, we choose to examine applications that are designed using trust scores based on hubs and authorities.

9.2. Score optimisation strategies in restrictive application environments

In this section, we evaluate[10] our choice of the heuristic optimisation strategy for application environments that prescribe a finite number of links between resource representations. We examine multiple heuristic strategies for finding *the next best link* for a new document. We show that the proposed optimization heuristic ARS outperforms alternative heuristic strategies.

We revert from the Web-oriented terminology of documents and references to the graph theoretical equivalents of vertices and edges. Auxiliary terminology is introduced:

Source set: A set of vertices with an in-degree of zero.
Leaf set: A set of vertices with an out-degree of zero.

[10]The corresponding code is available at https://github.com/uvdsl/thesis_decision_strategy_performance.

9.2.1. Mathematical modeling of the decision process

Let G^+ denote the transitive closure of the unmodified graph $G = (V, E)$. A new vertex v is added to the graph. Let $G' = (V', E')$ with $V' = V \cup \{v\}$ and $E' \supseteq E$. Let $(G')^+ = (V', (E')^+)$ denote the transitive closure of G'. The decision process modeled as follows:

Problem: Find $u \in V$ s.t. max $hub_{G^*}(v)$ given $G^* = (V', E' \cup \{(v, u)\})^+$
Optimisation crit.: Hub Score $hub_{G^*}(v)$ with $G^* = (V', E' \cup \{(v, u)\})^+$
Decision state: Set of vertices reachable by v, i.e. $\{w|(v, w) \in E^+\}$
Decision: Select $u \in V \setminus \{w|(v, w) \in E^+\}$
Decision criterion: Dependent on decision strategy

Initial Max Hub Score (Init HS): $hub_{G^+}(u)$
Iterative Max Hub Score (Iter HS): $hub_{(G')^+}(u)$
Initial Reachability (IRS): $\deg^+_{G^+}(u)$
Additional Reachability (ARS): $ar_{(G')^+}(u) = |\{w|(u, w) \in (E')^+\} \setminus \{w|(v, w) \in (E')^+\}|$

The *Initial Max Hub Score Strategy (Init HS)* chooses the vertex with the highest hub score based on the graph prior to the addition of v. The *Iterative Max Hub Score Strategy (Iter HS)* chooses the vertex with the highest hub score based on the current decision state, i.e. it iteratively considers already existing edges from v. The *Initial Reachability Strategy (IRS)* chooses the vertex with the highest number of not yet referenced vertices reachable, i.e. reachability, based on the graph prior to the addition of v. The *Additional Reachability Strategy (ARS)* chooses the vertex with the highest additional reachability based on the current decision state, i.e. it considers vertices already reachable from v.

Given a decision state, each decision strategy determines a set of vertices that it deems best as a potential next reference using its corresponding decision criterion. The set of vertices proposed by each of the decision strategies is referred to as the *option set* of that strategy given a decision state. Each vertex in the option set is deemed equally well-suited as a next reference; a random vertex from the option is chosen.

9.2.2. Methodology: Experimental analysis

The performance of the decision strategies is assessed by conducting an experimental analysis. Given a probability p that an edge exists between two vertices, a DAG with a fixed number of vertices n is randomly generated. There is no restriction on the graph other than it being a DAG. We examine such graphs with 20, 50 and 100 vertices at an edge probability between 0.95 and 0.05 by steps of 0.05. That is: $n \in \{20, 50, 100\}$ and $p \in \{0.95, 0.90, 0.85, \ldots, 0.15, 0.10, 0.05\}$. For each combination (n, p), 400 random DAGs were generated. All measurement values are averages over those 400 graphs.

An additional vertex v is introduced as the basis for the decision process. Possible[11] decision states are calculated. The larger the source vertex set, the more options to choose from exist. The number of source and leaf vertices is typically high in sparse graphs, which often result from random generation with a low edge probability p. Regarding this analysis, there exist 2^k decision states, with k being the number of source vertices in a graph. For every decision state, each decision strategy proposes an option set based on the corresponding decision criterion. The strategies' performance is compared against the optimal option set as ground truth.

We evaluate the decision strategies' performance with two approaches: First, we evaluate the proposed option sets on set-level. We evaluate a strategy's ability to find all optimal options, i.e. equality comparison. Additionally, we evaluate a strategy's ability to provide at least one optimal option without proposing a non optimal option, i.e. subset comparison. However, these option set comparisons lack qualitative expressivity regarding how good the option set is compared with the optimal option set as any set with one non-optimal candidate is discarded. Second, we evaluate the proposed option sets on element-level using precision and recall to assess the internal quality of the proposed option set. We evaluate a strategy's ability to propose all optimal options using recall, and the ability to propose optimal options rather than non-optimal using precision. Precision and recall allow for better understanding of how well a strategy is able to find optimal options as opposed to an optimal (sub-)set.

[11] Not all vertices need to be considered: Leveraging Lemma 1, it is best to reference hubs themselves instead of the documents referenced by those hubs. It is thus sufficient to consider the source vertex set.

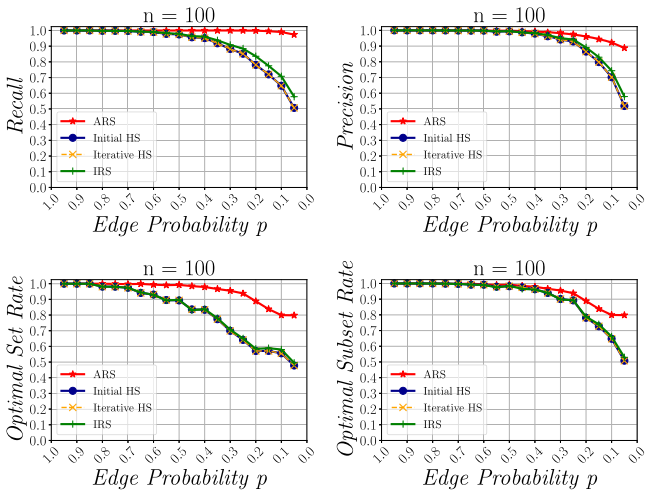

Fig. 7. Performance in random graphs with 100 vertices: ARS (red), IRS (green), IterHS (orange), InitHS (blue).

9.2.3. Results

In dense graphs, the difference in performance between strategies is marginal. In sparse graphs, ARS exhibits significantly better performance than the other strategies. ARS is able to propose option sets with high precision and very high recall. Even when considering subsets or set equality, ARS outperforms the other strategies significantly. Results of the experimental analysis suggest that ARS provides a well-performing heuristic. As an exemplary illustration of the results,[12] Fig. 7 provides an overview over the strategies' performance in graphs with 100 vertices and all performance measures. The performance of ARS is indicated by the red-starred line, which is the top line in all diagrams.

9.3. Trust score distribution: Score resilience in a document graph

In this section, we evaluate[13] how trust scores evolve in a Web, where our approach is applied. More specifically, we evaluate the resilience of trust scores in a document graph. A simulation of interacting agents is designed using the Watts–Strogatz model [50] to resemble an online social network, as in our example forum application from Section 4. The agents incrementally create a document graph in different interaction channels, i.e. two-agent interaction and multi-agent interaction channels. The behaviour of the authority score distribution, i.e. trustworthiness, in the document graph is examined. In particular, the change in scores from documents being deleted, such that the corresponding document graph becomes weakly disconnected, and recovery of scores from such disconnection through agent interaction is analysed.

9.3.1. Evaluation cases overview

We investigate the influence of document availability, i.e. a varying link structure within the graph, and of document confidentiality, i.e. varying access control, on the behaviour of the authority score distribution: First, *only content-based links* are created between documents of the same interaction channel (case A1). The link structure is not affected by any access control rules or documents being publicly accessible. Second, we assess the same document graph but now only consider *score-based links*, i.e. links that have been added to the documents to optimise the trust scores (cases (A2, C1) and (A2, C2)). Here, access control limits the number of available documents for linking. Third, we consider the same document graph but look at both types of links at the same time (cases (A3, C1) and (A3, C2)). Figure 8 summarizes the cases considered for the analysis of the document graph.

[12] Results of graphs with 20 and 50 vertices are available in Appendix C.

[13] The corresponding code is available at https://github.com/uvdsl/thesis_graph.

		Confidentiality	
		Access Control	**Transparency**
Availability	**Content-based links**	Case (A1)	
	Score-based links	Case (A2,C1)	Case (A2,C2)
	Both types of links	Case (A3,C1)	Case (A3,C2)

Fig. 8. An overview of examined evaluation cases.

9.3.2. Evaluation design

For each of the described cases, a simulation is conducted with a four-phase structure:

1. *Construction* of agent network and interaction model.
   Using the Watts–Strogatz model [50], we generate an agent network where two types of interaction channels exist: Two-agent-interaction of directly connected agents; and multi-agent-interaction of each agent with all its directly connected agents. Agent interaction is expressed in form of interlinked documents with connections based on the considered simulation case. The result of this phase is an agent network where each agent has access to certain interaction channels.
2. *Interaction* of agents creates a baseline document graph.
   A round-based system is established: In each round, agents become active with fixed probability and act simultaneously. Each active agent randomly chooses an available interaction channel and publishes a new document. The document includes a link to the last document of the channel, i.e. a content-based link, and/or links to all other latest messages of all channels the agent has access to, i.e. score-based links, depending on the simulation case. The result of this phase is a baseline document graph.
3. *Deletion* of documents to weakly disconnect the graph.
   The document deletion process is inspired by the robustness analysis of the Web graph of [24], in particular, the deletion of documents through targeted attacks using *betweenness centrality*. The betweenness centrality of a document is the number of shortest paths between two documents passing through that document. The more shortest paths pass through a document, the more it contributes to the transitive integrity preservation in the document graph. By removing the document with the highest betweenness centrality, the integrity preservation within the graph is reduced most. Moreover, these documents are typically members of cut sets of small size, thus leading to the graph becoming weakly disconnected without deleting more documents than needed. The result of this phase is a disconnected document graph with at least two components.
4. *Recovery* by reconnecting and extending the document graph.
   As agents aim to reference as many other documents as possible with new documents, new documents published will naturally reconnect the graph. As an initial effort, each agent adds one new message to the disconnected graph, referencing all documents that are not yet or not anymore referenced by another document and that are accessible by the agent. This initial addition to the disconnected graph creates a reconnected graph. With continuing agent interaction, the reconnected graph is extended based on the same interaction model created in the construction phase. The result of this phase are the reconnected document graph and the extended document graph.

The four phases are repeated using the Monte Carlo method to account for randomness in generation models used. Finally, the change in distribution of authority scores along the four steps is analyzed. The overall structure of the simulations is equal across all examined cases. Only the link structure between documents differs between the simulations dependent on the specific case.

9.3.3. Simulation parameters

For the simulations, the number of agents is set to 50. In agent network generated by the Watts–Strogatz model, each agent is connected to 4 neighboring agents with a probability of 0.25 that the connection to a neighboring agent is rewired. Regarding the interaction phase, agents are activated in a round with a fixed probability of 0.1. The activation probability was chosen to represent large-scale networks where multiple agents are active simultaneously. An activated agent chooses a interaction channel it publishes a message in with equal probability from all channels

available to the particular agent. The interaction phase is completed after 100 agent activation rounds. The deletion of documents based on betweenness centrality is dependent on the created graph. For recovery, the initial effort of all agents to reconnect the graph is fixed. Each agent publishes one new document referencing all documents that are not referenced by other documents. The following extension of the reconnected graph is similar to the interaction phase: Agents are activated with fixed probability of 0.1 and randomly choose an available chat to publish a new message. The extension interaction lasts only 10 rounds as opposed to the creation interaction phase's 100 rounds. The number of iterations of the Monte Carlo method is set to 1000 to achieve an overall reasonable representation of the documents' authority score distributions for the four document graphs, i.e. baseline, disconnected, reconnected and extended document graph.

9.3.4. Results

We provide a high-level summary of our results. We visualised the results for all evaluation cases using boxplots depicted in Fig. 9. Note that in case (A1), we observe a drastic change in trust score distribution which is caused by the initial effort of agents to reconnect the graph. More specifically, documents added after that point are able to contribute to more documents' authority score than previously. This is out-of-the-ordinary for this scenario and thus must be taken with a grain of salt.

In general, the analysis of the document graph shows that the authority scores of documents suffer severely when the graph becomes weakly disconnected. This is reflected by the authority score distribution exhibiting a strong positive skew[14] while being overall shifted towards lower values in the disconnected graph. Documents of the second

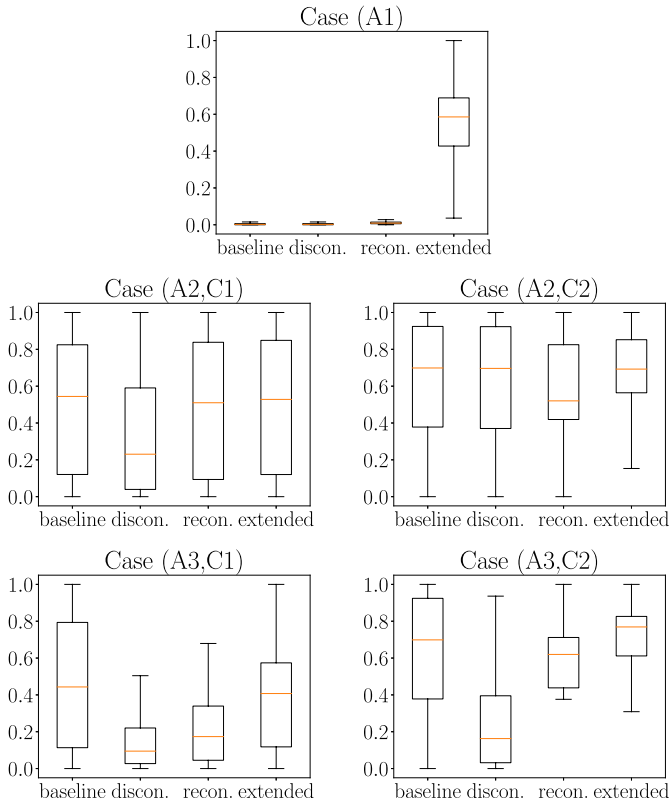

Fig. 9. Boxplots of authority scores for case (A1) (top-row), case (A2, C1) and (A2, C2) (mid-row) and case (A3, C1) and (A3, C2) (bottom row) for the baseline, disconnected, reconnected and extended graph.

[14] Exception for case (A2, C2) where the components of the disconnected graph seems to exhibit a similar link structure as the baseline graph. The shift in scores is delayed until the graph is reconnected.

component in the graph no longer contributing to the scores in the first component. The two components equalize in authority score levels, which are typically lower than in the baseline graph. With new documents reconnecting the graph, authority scores in the graph start to normalize to baseline levels. Extending the reconnected graph further brings the distribution closer to the baseline for all cases. The authority score distribution is able to recover to a certain extent depending on the simulation case.

Availability: Comparison of link structures Content-based links produce sparse graphs. Score-base links produce decently connected graphs with quick recovery potential. Having both types of links produces dense graphs that recover more slowly. To achieve disconnection, more documents need to be removed in better connected graphs than in sparse graphs. Similarly, more documents are needed for recovery in better connected graphs than in sparse graphs. This is due to dense graphs being less sensitive to one documents being deleted or added. Since it takes more documents in dense graphs to achieve disconnection, well-connected graphs show a higher sensitivity to disconnection regarding their authority score distribution than sparse graphs. Similarly, recovery takes more added documents in well-connected graphs than in sparse graphs. More documents removed equals more effect on the authority score distribution.

Confidentiality: Access control vs. transparency Transparency allows for very well-connected graphs. This is reflected by strong negative skews in the authority score distribution. Moreover, the authority score distribution is shifted towards higher values when recovering from graph disconnection. This indicates that the graph is connected such that quick recovery is possible. Reaching pre-disconnection levels, however, is not immediate. Confidentiality reduces the connectedness of a graph by fostering clusters of resources. This is reflected by a more spread out authority score distribution. In the conducted simulation, the observed distributions for score-based and both-link-type graphs were fairly balanced only showing a slight positive skew. This indicates that resources under access control may exhibit lower authority scores than with transparency, which is expected. The overall balanced authority score distributions seem reasonable enough to be practically relied on in an application context.

9.4. Discussion

We consider three aspects for further discussion with our proposed work. First, we touch on the fundamental assumption of small-world agent networks in our evaluation. Second, we revisit the topic of non-evident document updates. And third, we discuss the notion of trust within our approach.

First, regarding the evaluation of the resilience of trust score distribution (Section 9.3), we base our simulation on the Watts–Strogatz model [50] to create small-world networks. These types of networks typically occur in social networks in the real world. Therefore, our evaluation is geared towards applications in such agent networks. Other applications may be based on agent network that exhibit a different topology, possibly taken from real-world examples. Those applications are to be considered in future research.

Next, recall Section 6.2 where we describe that non-evident document updates are still possible in our approach. But this is only the case if (a) all involved agents agree to the update and (b) agree to delete all evidence of the original resources. Which documents may be linked to by other agents is subject to the applications design: Consider an application environment, for example a typical public online forum where agents interact in multiple public group chats, where only content-based links exist between documents. This would mean that if all agents of one public group chat agree to an update, no evidence of the original messages persists. While this may be acceptable for some applications, others may not desire such behaviour due to stricter data integrity requirements. In such a case, score-based links may help increase the number of involved agents including otherwise uninvolved ones. Depending on the applications design, and incentive induced by the trust scores on agents, this may be enough for a specific case. In other cases, additional measures, such as a central or distributed link registry may be required to achieve the design goals of the specific application.

Last, our notion of trust refers only to structural integrity of an RDF document – we do not consider content-level inconsistencies, i.e. contradicting statements. Our notion of trust does not cover the truthfulness of statements in a document. In some applications, however, trust score could give decision indication when different documents contain contradicting statements. The decision which document and subsequently its statements to accept as normative may be guided by trust scores. This may be similar to heaviest chain decision rule in bitcoin, depending on

the application. For other applications, this indication will be misleading, e.g. when documents exhibit high trust scores due to their age but at the same time contain obsolete statements, e.g. about the age of a person. So, there are application-dependent design goals and decisions to consider.

10. Conclusion

We presented an approach to share Linked Data with quantifiable integrity on the Web. As our infrastructure, we rely on Web technologies, such as RDF and the Solid Protocol. Using Linked Data Signatures and Signed URIs, self-verifying resource representations provide the foundation for a document graph on the Web whose link-structure contributes to preserving its integrity. With self-verifying resource representations, an agent is able to verify the document's integrity, i.e., that it has not been modified since it was published, and the document's authenticity, i.e., that it was created by a specific author. To quantify how well a document's integrity is preserved, and thus, how well an agent may rely on that document, we introduced the notion of trust scores proposing an interpretation based on hubs and authorities. Optimising the scores incentivises agents to contribute to the document graph in a constructive and integrity preserving manner. When the general strategy of simply referencing all documents transitively is not applicable, we introduced ARS, the Additional Reach Strategy, as a heuristic alternative. We discussed our approach in a three-fold evaluation: First, we took a look at different graph metrics as trust scores We show that trust scores based on hubs and authorities induce agent behaviour that contributes to integrity preservation in the document graph. Other graph metrics where only suitable with significant caveats or not suitable at all. Next, we compared ARS to different heuristics for agents to optimise trust scores. We show that ARS outperforms other potential optimisation strategies. Last, we proposed a simulation system based on the Watts–Strogatz model for simulating a social network. In this social network, agents interact with each other thereby creating a document graph. We evaluated our approach by examining the resilience of integrity preservation in such document graph when resources are deleted. We show that our approach produces a document graph that can recover from such attacks or failures in the document graph.

With our research, we hope to promote tamper-evidence of Linked Data on the Web and thus to contribute to the vision of a more decentralised and more reliable Web.

Appendix A. Proof for page rank punishing additional linkage

Proof. The PageRank r is the principal eigenvector of $\mathcal{M} = (dM + \frac{1-d}{n}E)$ [4]. Moreover, the principal eigenvector r can be approximated using the power method as the unit vector resulting from the convergence of

$$r = \mathcal{M}^k z \quad \text{with } k \longmapsto \infty \tag{7}$$

where k, the number of iterations, and z, the identity vector. As we choose $d = 1$, we only have to look at $r = M^k z$. Additionally, since matrix potentiation is only repeatedly executed matrix multiplication, it suffices to examine $k = 2$ to cover all $k \geqslant 2$.

Let further $U = \{1, 2, \ldots, n\}$ be the set of all documents in the graph. Let $U_o \subseteq U$ denote the set of documents of a particular agent o. Consider the adjacency matrix A:

$$A = (a_{ij}) \quad \text{with } a_{ij} = \begin{cases} \{0, 1\} & \text{if } i > j \\ 0, & \text{otherwise} \end{cases} \text{ and } i, j \in U \tag{8}$$

And the corresponding transition probability matrix M:

$$M = (m_{pq}) \quad \text{with } m_{pq} = \begin{cases} a_{qp} / \sum_{s \in U} a_{qs} & \text{if } \sum_{s \in U} a_{qs} > 0 \\ 1/n, & \text{otherwise} \end{cases} \text{ and } p, q \in U \tag{9}$$

Then, the matrix potentiation results in:

$$M^2 = M' = \left(m'_{pq}\right) \quad \text{with } p, q \in U$$
$$\text{where } m'_{pq} = \sum_{k \in U} m_{pk} m_{kq} \tag{10}$$

And the resulting page rank (if the power method was terminated after this iteration):

$$r' = M'z = (r_p) \quad \text{with } p \in U$$
$$\text{where } r_p = \sum_{q \in U} m'_{pq} = \sum_{q \in U} \sum_{k \in U} m_{pk} m_{kq} \tag{11}$$

Consider a document $z \in U_o$ and a second document $x \notin U_o$. We assume that document z links to documents from U_o but no additional documents.

$$\sum_{s \in U} a_{zs} > 0 \tag{12}$$

We compare the following cases:

1. $a^{\lambda}_{zx} = 0$, the case where $a_{zx} = 0$, i.e. there exists no link from z to x, denoted by λ.
2. $a^{\gamma}_{zx} = 1$, the case where $a_{zx} = 1$, i.e. there exists a link from z to x, denoted by γ.

It follows:

$$\sum_{s \in U} a^{\gamma}_{zs} = 1 + \sum_{s \in U} a^{\lambda}_{zs} \tag{13}$$

and for the connection from z to x:

$$m^{\gamma}_{xz} = \frac{a^{\gamma}_{zx}}{\sum_{s \in U} a^{\gamma}_{zs}} = \frac{1}{\sum_{s \in U} a^{\gamma}_{zs}} > \frac{0}{\sum_{s \in U} a^{\lambda}_{zs}} = \frac{a^{\lambda}_{zx}}{\sum_{s \in U} a^{\lambda}_{zs}} = m^{\lambda}_{xz} = 0 \tag{14}$$

and for the connection from z to any other document $p \in U \setminus \{x\}$:

$$m^{\gamma}_{pz} = \frac{a^{\gamma}_{zp}}{\sum_{s \in U} a^{\gamma}_{zs}} = \frac{a_{zp}}{\sum_{s \in U} a^{\gamma}_{zs}} \leqslant \frac{a_{zp}}{\sum_{s \in U} a^{\lambda}_{zs}} = \frac{a^{\lambda}_{zp}}{\sum_{s \in U} a^{\lambda}_{zs}} = m^{\lambda}_{pz}$$
$$\text{since } a_{zp} = a^{\gamma}_{zp} = a^{\lambda}_{zp} \quad \forall p \in U \setminus \{x\} \tag{15}$$

where $\forall p \in U \setminus \{x\}$ the delta is:

$$\begin{aligned} m^{\gamma}_{pz} - m^{\lambda}_{pz} &= \frac{a_{zp}}{\sum_{s \in U} a^{\gamma}_{zs}} - \frac{a_{zp}}{\sum_{s \in U} a^{\lambda}_{zs}} \\ &= \frac{a_{zp}}{1 + \sum_{s \in U} a^{\lambda}_{zs}} - \frac{a_{zp}}{\sum_{s \in U} a^{\lambda}_{zs}} \\ &= \left(- \frac{a_{zp}}{(\sum_{s \in U} a^{\lambda}_{zs})(1 + \sum_{s \in U} a^{\lambda}_{zs})} \right) \leqslant 0 \end{aligned} \tag{16}$$

Revisiting a document p's PageRank (equation (11)), we can extract terms related to documents z and x, i.e. terms that change between cases λ and γ:

$$\begin{aligned} r_p &= \sum_{q\in U} m'_{pq} = \sum_{q\in U}\sum_{k\in U} m_{pk}m_{kq} \\ &= \sum_{k\in U}\left(m_{pk}\sum_{q\in U} m_{kq}\right) \\ &= \sum_{k\in U\setminus\{x,z\}}\left(m_{pk}\left(m_{kx}+m_{kz}+\sum_{q\in U\setminus\{x,z\}} m_{kq}\right)\right) \\ &\quad + m_{px}\left(m_{xx}+m_{xz}+\sum_{q\in U\setminus\{x,z\}} m_{xq}\right) \\ &\quad + m_{pz}\left(m_{zx}+m_{zz}+\sum_{q\in U\setminus\{x,z\}} m_{zq}\right) \end{aligned} \tag{17}$$

In particular, the terms change between cases λ and γ where z appears as in second position of the subscript, i.e. $m_{kz}\ \forall k \in U$. All other terms remain unchanged between cases. In addition, it is worth noting that $m_{xx} = m_{zz} = 0$ as there may not exist (self-)loops in the document graph, which is a DAG. Moreover, as we consider the case where $a_{zx} = 1$ may hold true. It follows that $a_{xz} = 0$ and thus $m_{zx} = 0$ (again due to the DAG). We further consider z to be a newly created document, i.e. to have no incoming links: $\sum_{q\in U} m_{zq} = 0$. Existing documents, i.e. documents that may have incoming links, may not be modified anyway.

It follows for the delta in PageRank of a document p between cases:

$$\begin{aligned} r_p^\gamma - r_p^\lambda &= \sum_{k\in U}\left(m_{pk}^\gamma \sum_{q\in U} m_{kq}^\gamma\right) - \sum_{k\in U}\left(m_{pk}^\lambda \sum_{q\in U} m_{kq}^\lambda\right) \\ &= \sum_{k\in U\setminus\{x,z\}} \left(m_{pk}\left(m_{kz}^\gamma - m_{kz}^\lambda\right)\right) + m_{px}\left(m_{xz}^\gamma - m_{xz}^\lambda\right) \\ &= \sum_{k\in U\setminus\{x,z\}} \left(m_{pk}\left(m_{kz}^\gamma - m_{kz}^\lambda\right)\right) + m_{px}m_{xz}^\gamma \end{aligned} \tag{18}$$

We show that the sum of agent o's documents' PageRank will not increase in the case γ compared to the case λ:

$$\begin{aligned} \sum_{p\in U_o}\left(r_p^\gamma - r_p^\lambda\right) &= \sum_{p\in U_o}\left(\sum_{k\in U\setminus\{x,z\}} \left(m_{pk}\left(m_{kz}^\gamma - m_{kz}^\lambda\right)\right) + m_{px}m_{xz}^\gamma\right) \\ &= \sum_{p\in U_o}\left(m_{px}m_{xz}^\gamma + \sum_{k\in U\setminus\{x,z\}} \left(m_{pk}\left(m_{kz}^\gamma - m_{kz}^\lambda\right)\right)\right) \\ &= \sum_{p\in U_o}\left(\sum_{k\in U\setminus\{x,z\}} \left(m_{pk}\left(m_{kz}^\gamma - m_{kz}^\lambda\right)\right)\right) + \sum_{p\in U_o}\left(m_{px}m_{xz}^\gamma\right) \\ &= \sum_{k\in U\setminus\{x,z\}}\left(\left(m_{kz}^\gamma - m_{kz}^\lambda\right)\sum_{p\in U_o} m_{pk}\right) + \sum_{p\in U_o}\left(m_{px}m_{xz}^\gamma\right) \end{aligned}$$

$$\overset{\text{using (16)}}{=} \sum_{k\in U\setminus\{x,z\}} \left(\left(-\frac{a_{zk}}{(\sum_{s\in U} a^{\lambda}_{zs})(1+\sum_{s\in U} a^{\lambda}_{zs})} \right) \sum_{p\in U_o} \frac{a_{kp}}{\sum_{s\in U} a_{ks}} \right) + \sum_{p\in U_o} \left(\frac{a_{xp}}{\sum_{s\in U} a_{xs}} \frac{a^{\gamma}_{zx}}{\sum_{s\in U} a^{\gamma}_{zs}} \right) \tag{19}$$

At this point, we distinguish two cases:

The case (I) where $\sum_{p\in U_o} a_{xp} = 0$, i.e. there exists no link from x to any p of agent o.
The case (II) where $\sum_{p\in U_o} a_{xp} > 0$, i.e. there exists a link from x to at least one p of agent o.

For case (I), it follows directly that

$$\sum_{p\in U_o} \left(r^{\gamma}_p - r^{\lambda}_p\right) = \sum_{k\in U\setminus\{x,z\}} \left(\left(-\frac{a_{zk}}{(\sum_{s\in U} a^{\lambda}_{zs})(1+\sum_{s\in U} a^{\lambda}_{zs})} \right) \sum_{p\in U_o} \frac{a_{kp}}{\sum_{s\in U} a_{ks}} \right) \leqslant 0 \tag{20}$$

For case (II), we continue

$$\begin{aligned}
\sum_{p\in U_o} \left(r^{\gamma}_p - r^{\lambda}_p\right) &= \sum_{k\in U\setminus\{x,z\}} \left(\left(-\frac{a_{zk}}{(\sum_{s\in U} a^{\lambda}_{zs})(1+\sum_{s\in U} a^{\lambda}_{zs})} \right) \sum_{p\in U_o} \frac{a_{kp}}{\sum_{s\in U} a_{ks}} \right) \\
&\quad + \sum_{p\in U_o} \left(\frac{a_{xp}}{\sum_{s\in U} a_{xs}} \frac{a^{\gamma}_{zx}}{\sum_{s\in U} a^{\gamma}_{zs}} \right) \\
&\overset{\text{using (13)}}{=} \frac{1}{(1+\sum_{s\in U} a^{\lambda}_{zs})} \left(\sum_{k\in U\setminus\{x,z\}} \left(\left(-\frac{a_{zk}}{(\sum_{s\in U} a^{\lambda}_{zs})} \right) \sum_{p\in U_o} \frac{a_{kp}}{\sum_{s\in U} a_{ks}} \right) + \sum_{p\in U_o} \left(\frac{a_{xp} a^{\gamma}_{zx}}{\sum_{s\in U} a_{xs}} \right) \right) \\
&\overset{\text{using } (a^{\gamma}_{zx}=1)}{=} \frac{1}{(1+\sum_{s\in U} a^{\lambda}_{zs})} \left(\sum_{k\in U\setminus\{x,z\}} \left(\left(-\frac{a_{zk}}{(\sum_{s\in U} a^{\lambda}_{zs})} \right) \sum_{p\in U_o} \frac{a_{kp}}{\sum_{s\in U} a_{ks}} \right) \right. \\
&\quad \left. + \sum_{p\in U_o} \left(\frac{a_{xp}}{\sum_{s\in U} a_{xs}} \right) \right)
\end{aligned} \tag{21}$$

We see that the extend of the delta is determined by how well-connected the document graph is, i.e. how many connections from z to p via k exist and how well-connected x within the document graph is. We thus distinguish the two extreme cases:

The case (II.a) where agents link all the documents regardless which agent they belong to.
The case (II.b) where agents only link documents that belong to themselves (which is our intuitive argument).
Note that case (II) entails that there must be at least one link from x to some $p \in U_o$, i.e. $\sum_{p\in U_o} a_{xp} > 0$.

For simplicity, we ignore that the document graph is a DAG but still forbid self-loops.

For case (II.a). For simplicity, we dissect the equation starting with the second summand:

$$\sum_{p\in U_o} \left(\frac{a_{xp}}{\sum_{s\in U} a_{xs}} \right)) = \frac{|U_o|}{n-1} \tag{22}$$

And for the first summand (remember that $\sum_{s\in U} a^{\lambda}_{zs} = n-2$ as self-loop and link to x are excluded):

$$\begin{aligned}\sum_{k\in U\setminus\{x,z\}}\left(\left(-\frac{a_{zk}}{(\sum_{s\in U} a^{\lambda}_{zs})}\right)\sum_{p\in U_o}\frac{a_{kp}}{\sum_{s\in U} a_{ks}}\right) &= \sum_{k\in U\setminus\{x,z\}}\left(\left(-\frac{a_{zk}}{(\sum_{s\in U} a^{\lambda}_{zs})}\right)\frac{|U_o|}{n-1}\right)\\ &= \frac{|U_o|}{n-1}\sum_{k\in U\setminus\{x,z\}}\left(-\frac{a_{zk}}{(\sum_{s\in U} a^{\lambda}_{zs})}\right)\\ &= -\frac{|U_o|}{n-1}\frac{n-2}{n-2}\\ &= -\frac{|U_o|}{n-1}\end{aligned} \tag{23}$$

Substituting in the equation:

$$\begin{aligned}\sum_{p\in U_o}\left(r^{\gamma}_p - r^{\lambda}_p\right) &= \frac{1}{(1+\sum_{s\in U} a^{\lambda}_{zs})}\left(\sum_{k\in U\setminus\{x,z\}}\left(\left(-\frac{a_{zk}}{(\sum_{s\in U} a^{\lambda}_{zs})}\right)\sum_{p\in U_o}\frac{a_{kp}}{\sum_{s\in U} a_{ks}}\right)+\sum_{p\in U_o}\left(\frac{a_{xp}}{\sum_{s\in U} a_{xs}}\right)\right)\\ &= \frac{1}{(1+\sum_{s\in U} a^{\lambda}_{zs})}\left(-\frac{|U_o|}{n-1}+\frac{|U_o|}{n-1}\right)=0\end{aligned} \tag{24}$$

For an agent o, there is no PageRank to be gained from including the additional link in this case. There exists no incentive for the agent to include the additional link.

For case (II.b). We dissect the equation starting with the second summand:

$$\sum_{p\in U_o}\left(\frac{a_{xp}}{\sum_{s\in U} a_{xs}}\right)) = \frac{1}{\sum_{s\in U} a_{xs}} \quad \text{where } \sum_{s\in U} a_{xs} \geqslant 1 \text{ due assumption of case (II.b).} \tag{25}$$

And for the first summand:

$$\begin{aligned}\sum_{k\in U\setminus\{x,z\}}\left(\left(-\frac{a_{zk}}{(\sum_{s\in U} a^{\lambda}_{zs})}\right)\sum_{p\in U_o}\frac{a_{kp}}{\sum_{s\in U} a_{ks}}\right) &= \sum_{k\in U\setminus\{x,z\}}\left(\left(-\frac{a_{zk}}{(\sum_{s\in U} a^{\lambda}_{zs})}\right)\sum_{p\in U_o}\frac{a_{kp}}{\sum_{s\in U} a_{ks}}\right)\\ &= \sum_{k\in U_o\setminus\{x,z\}}\left(\left(-\frac{a_{zk}}{|U_o|-2}\right)\sum_{p\in U_o}\frac{a_{kp}}{|U_o|-1}\right)\\ &= \sum_{k\in U_o\setminus\{x,z\}}\left(\left(-\frac{1}{|U_o|-2}\right)\frac{|U_o|-1}{|U_o|-1}\right)\\ &= \left(-\frac{|U_o|-2}{|U_o|-2}\frac{|U_o|-1}{|U_o|-1}\right)=-1\end{aligned} \tag{26}$$

Substituting in the equation:

$$\begin{aligned}\sum_{p\in U_o}\left(r^{\gamma}_p - r^{\lambda}_p\right) &= \frac{1}{(1+\sum_{s\in U} a^{\lambda}_{zs})}\left(\sum_{k\in U\setminus\{x,z\}}\left(\left(-\frac{a_{zk}}{(\sum_{s\in U} a^{\lambda}_{zs})}\right)\sum_{p\in U_o}\frac{a_{kp}}{\sum_{s\in U} a_{ks}}\right)+\sum_{p\in U_o}\left(\frac{a_{xp}}{\sum_{s\in U} a_{xs}}\right)\right)\\ &= \frac{1}{(1+\sum_{s\in U} a^{\lambda}_{zs})}\left(-1+\frac{1}{\sum_{s\in U} a_{xs}}\right)\leqslant 0\end{aligned} \tag{27}$$

Examining equation (27) yields that the better connected document x is outside of documents from agent o, the more it hurts agent o's documents' PageRank to include the additional link, which is in line with our intuitive argument.

From equations (20), (24) and (27) we conclude that there exists no incentive for an agent to include links to documents other than its own. Although it is not a dominant strategy, i.e. the best option in all cases independently of what the other agents do, the agent is at least indifferent towards the alternative strategy of referencing other agents' documents. As the inclusion of such additional links may require some effort by the agent, the agent may simply choose not to endure this effort for an outcome that is at best not better than without that effort. If all agents follow this reasoning, no agent will include additional links to other agents' documents, there exists no incentive to include additional links (see equation (20)) and thus we found a Nash equilibrium. The PageRank does not provide an inherent incentive for agents to collaborate and contribute to the document graph in a constructive way. The PageRank is thus not suitable as foundation for designing applications based on trust scores. □

Appendix B. Proof for Lemma 1

Proof. Consider the adjacency matrix A of a DAG's nodes in topological ordering:

$$A = (a_{ij}) \quad \text{with } a_{ij} = \begin{cases} \{0,1\} & \text{if } i > j \\ 0, & \text{otherwise} \end{cases} \text{ and } i, j = 1, .., n \tag{28}$$

The corresponding hub matrix H is calculated by

$$\begin{aligned} AA^T = H = (H_{ik}) \quad & \text{with } i, k = 1, .., n \\ \text{where } H_{ik} = \sum_{j=1}^{n} a_{ij}a_{kj} = & \sum_{j=1}^{\min\{i,k\}-1} a_{ij}a_{kj} \end{aligned} \tag{29}$$

Trivially, it holds that $H \geqslant 0$ due to $A \geqslant 0$. Moreover, $H = AA^T$ is symmetric.

Assume two nodes with indices p and q where $p < q$ and every node j that is linked to by p is also linked to by q:

$$a_{pj} \leqslant a_{qj} \quad \forall j, p < q \tag{30}$$

It follows directly from equation (29):

$$\begin{aligned} \sum_{j=1}^{p-1} a_{pj} = H_{pp} \leqslant H_{qq} = \sum_{j=1}^{q-1} a_{qj} \\ \sum_{j=1}^{p-1} a_{pj} = H_{pp} = H_{qp} \end{aligned} \tag{31}$$

And especially:

$$H_{pk} \leqslant H_{qk} \quad \forall k \tag{32}$$

The hub score vector h is calculated by $h = (H)^k z$ with z being the identity vector. Since matrix potentiation is only repeatedly executed matrix multiplication, it suffices to examine $k = 2$ to cover all $k \geqslant 2$. We recall $H = AA^T$

is symmetric.

$$H^2 = H' = \left(H'_{ik}\right) \quad \text{with } i,k = 1,..,n$$
$$\text{where } H'_{ik} = \sum_{j=1}^{n} H_{ij} H_{kj} \tag{33}$$

It follows for nodes p and q in H' with equation (32):

$$H'_{pk} \leqslant H'_{qk} \quad \forall k \tag{34}$$

For the calculation of the hub scores h_p and h_q from hub score vector h follows:

$$h = H' \cdot z$$
$$\sum_{j=1}^{n} H'_{pj} = h_p \leqslant h_q = \sum_{j=1}^{n} H'_{qj} \tag{35}$$

Under the general assumption of (30), it holds that the hub score of node p is at least equal or higher than the hub score of q.

Equation (30) provides two specific cases to examine: The edge cases of strict equality and the case of (marginal) inequality, i.e. q links to (incrementally) more nodes than p.

Consider the *case of equality* for equation (30). Following the same steps as before results in equality of equation (35), i.e.

$$\begin{aligned} & a_{pj} = a_{qj} \quad \forall j,\ p < q \\ & \overset{(29)}{\Rightarrow} H_{pk} = H_{qk} \quad \forall k \\ & \overset{(33)}{\Rightarrow} H'_{pk} = H'_{qk} \quad \forall k \\ & \overset{(35)}{\Rightarrow} h_p = h_q \end{aligned} \tag{36}$$

For the case of equality, it holds that the hub scores of node p and q are equal.

Consider the *case of marginal inequality*, where node q links to incrementally more nodes than p. This increment leads to strict inequality in equation (35), i.e.

$$\begin{aligned} & a_{ps} < a_{qs} \quad \exists s,\ p < q \\ & a_{pj} = a_{qj} \quad \forall j \neq s \\ & \overset{(29)}{\Rightarrow} H_{pt} < H_{qt} \quad \exists t \\ & H_{pk} = H_{qk} \quad \forall k \neq t \\ & \overset{(33)}{\Rightarrow} H'_{pt} < H'_{qt} \quad \exists t \\ & H'_{pk} = H'_{qk} \quad \forall k \neq t \\ & \overset{(35)}{\Rightarrow} h_p < h_q \end{aligned} \tag{37}$$

Equation (37) proves that a node which has an additional increment link over another node while otherwise linking the same other nodes will exhibit a higher hub score than that node. This includes the case where q links additionally to p itself. Considering the transitive closure of the DAG, when q links to p, q transitively links to all nodes linked to by p and additionally to p itself. By equation (37), node q will exhibit a higher hub score than node p, which proves Lemma 1. □

Appendix C. Supplementary results of decision strategy performance

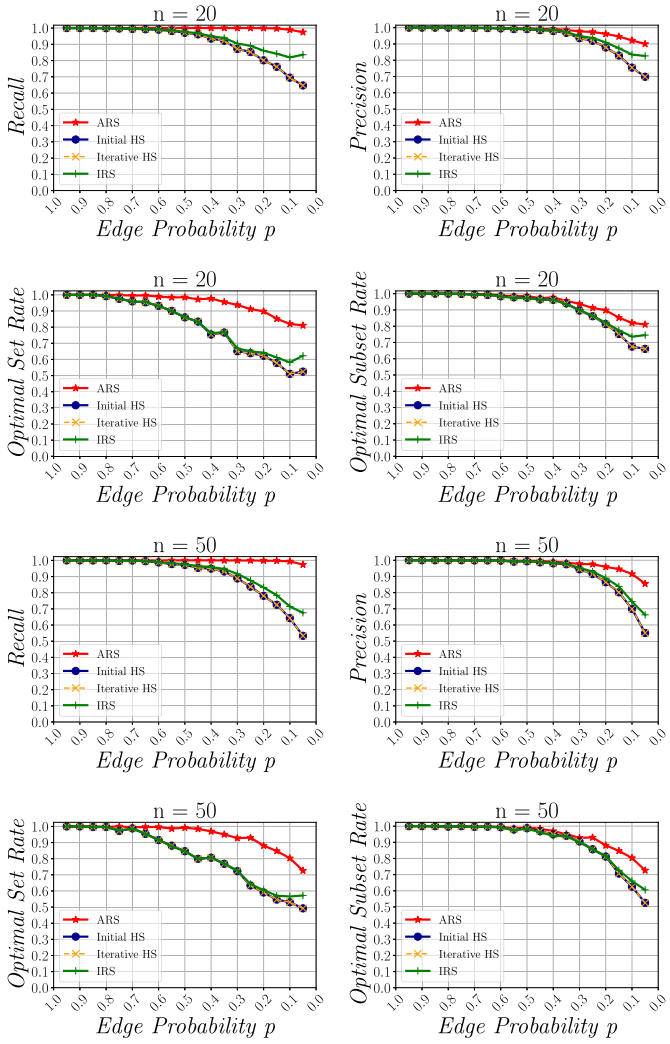

Fig. 10. Decision strategy performance in random graphs with 20 and 50 vertices. ARS in red, IRS in green, IterHS in orange, InitHS in blue.

References

[1] I. Amed, A. Balchandani, M. Beltram, A. Berg, S. Hedrich and F. Rölkens, What radical transparency could mean for the fashion industry, *McKinsey & Company* (2019), https://www.mckinsey.com/industries/retail/our-insights/what-radical-transparency-could-mean-for-the-fashion-industry.

[2] American National Standards Institute, Inc., *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, X9.62, ANSI, 2005.

[3] G. Annan and K.H. Dufy, Resource integrity proofs, 2018, https://nbviewer.jupyter.org/github/WebOfTrustInfo/rwot7/blob/master/final-documents/resource-integrity-proofs.pdf.

[4] A. Arasu, J. Novak, A. Tomkins and J. Tomlin, PageRank computation and the structure of the web: Experiments and algorithms, in: *Proceedings of the Poster Track at the 11th International World Wide Web Conference (WWW)*, 2002, pp. 107–117.

[5] J. Arias-Fisteus, N.F. García, L.S. Fernández and C.D. Kloos, Hashing and canonicalizing Notation 3 graphs, *Journal of Computer and System Sciences* **76**(7) (2010), 663–685. doi:10.1016/j.jcss.2010.01.003.

[6] T. Berners-Lee, Linked data, 2006, https://www.w3.org/DesignIssues/LinkedData.
[7] T. Berners-Lee, R. Fielding and L. Masinter, Uniform resource identifier (URI): Generic syntax, Internet Standards Track document, IETF, 2005, https://www.ietf.org/rfc/rfc3986.txt.
[8] C. Braun and T. Käfer, Verifying the integrity of hyperlinked information using linked data and smart contracts, in: *Proceedings of the 15th International Conference on Semantic Systems (SEMANTiCS)*, M. Acosta, P. Cudré-Mauroux, M. Maleshkova, T. Pellegrini, H. Sack and Y. Sure-Vetter, eds, Lecture Notes in Computer Science, Vol. 11702, Springer, 2019, pp. 376–390. doi:10.1007/978-3-030-33220-4_28.
[9] C. Braun and T. Käfer, Verifying the integrity of information along a supply chain using linked data and smart contracts, in: *Proceedings of the Posters and Demo Track at the 15th International Conference on Semantic Systems (SEMANTiCS)*, M. Alam, R. Usbeck, T. Pellegrini, H. Sack and Y. Sure-Vetter, eds, CEUR Workshop Proceedings, Vol. 2451, CEUR-WS.org, 2019, http://ceur-ws.org/Vol-2451/paper-07.pdf.
[10] C.H.-J. Braun and T. Käfer, Self-verifying web resource representations using Solid, RDF-star and signed URIs, in: *Proceedings of Posters & Demos at the 19th European Semantic Web Conference (ESWC)*, 2022.
[11] S. Capadisli, Web access control, Editor's draft, W3C, 2022, https://solid.github.io/web-access-control-spec/.
[12] S. Capadisli, T. Berners-Lee, R. Verborgh and K. Kjernsmo, Solid protocol, version 0.9.0, W3C Solid CG, 2021, https://solidproject.org/TR/protocol.
[13] S. Capadisli and A. Guy, Linked data notifications, W3C Recommendation, W3C, 2017, https://www.w3.org/TR/ldn/.
[14] J.J. Carroll, Signing RDF graphs, in: *Proceedings of the 2nd International Semantic Web Conference (ISWC)*, Lecture Notes in Computer Science, Vol. 2870, Springer, 2003, pp. 369–384. doi:10.1007/978-3-540-39718-2_24.
[15] J.J. Carroll, C. Bizer, P.J. Hayes and P. Stickler, Named graphs, *J. Web Semant.* **3**(4) (2005), 247–267. doi:10.1016/j.websem.2005.09.001.
[16] A. Coburn, e. Pavlik and D. Zagidulin, Solid-OIDC, Draft Community Group report, W3C, 2022, https://solid.github.io/solid-oidc/.
[17] R. Cyganiak, D. Wood and M. Lanthaler, RDF 1.1 concepts and abstract syntax, W3C Recommendation, W3C, 2014, https://www.w3.org/TR/rdf11-concepts/.
[18] J. Eberhardt and S. Tai, On or off the blockchain? Insights on off-chaining computation and data, in: *Proceedings of the 6th European Conference on Service-Oriented and Cloud Computing (ESOCC)*, F.D. Paoli, S. Schulte and E.B. Johnsen, eds, Lecture Notes in Computer Science, Vol. 10465, Springer, 2017, pp. 3–15. doi:10.1007/978-3-319-67262-5_1.
[19] R. Fielding and J. Reschke, Hypertext transfer protocol (HTTP/1.1): Message syntax and routing, Internet Standards Track document, IETF, 2014, https://www.ietf.org/rfc/rfc7230.txt.
[20] R. Fielding and J. Reschke, Hypertext transfer protocol (HTTP/1.1): Semantics and content, Internet Standards Track document, IETF, 2014, https://www.ietf.org/rfc/rfc7231.txt.
[21] R.T. Fielding and R.N. Taylor, Architectural styles and the design of network-based software architectures, PhD thesis, 2000, AAI9980887. ISBN 0599871180.
[22] V. Gaur and A. Gaiha, Building a transparent supply chain, *Harvard Business Review* (2020), https://hbr.org/2020/05/building-a-transparent-supply-chain.
[23] V. Gehrau, S. Fujarski, H. Lorenz, C. Schieb and B. Blöbaum, The impact of health information exposure and source credibility on COVID-19 vaccination intention in Germany, *International Journal of Environmental Research and Public Health* **18**(9) (2021), https://www.mdpi.com/1660-4601/18/9/4678. doi:10.3390/ijerph18094678.
[24] C. Guéret, P. Groth, F. van Harmelen and S. Schlobach, Finding the Achilles heel of the web of data: Using network analysis for link-recommendation, in: *Proceedings of the 9th International Semantic Web Conference (ISWC)*, P.F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J.Z. Pan, I. Horrocks and B. Glimm, eds, Lecture Notes in Computer Science, Vol. 6496, Springer, 2010, pp. 289–304. doi:10.1007/978-3-642-17746-0_19.
[25] R. Guha, R. Kumar, P. Raghavan and A. Tomkins, Propagation of trust and distrust, in: *Proceedings of the 13th International World Wide Web Conference (WWW), WWW'04*, Association for Computing Machinery, New York, NY, USA, 2004, pp. 403–412. ISBN 158113844X. doi:10.1145/988672.988727.
[26] H. Halpin, Decentralizing the social web – can blockchains solve ten years of standardization failure of the social web? in: *Proceedings of an International Workshop on the Future of Decentralized Governance at the International Conference on Internet Science (INSCI)*, S.S. Bodrunova, O. Koltsova, A. Følstad, H. Halpin, P. Kolozaridi, L. Yuldashev, A.S. Smoliarova and H. Niedermayer, eds, Lecture Notes in Computer Science, Vol. 11551, Springer, 2018, pp. 187–202. doi:10.1007/978-3-030-17705-8_16.
[27] A. Harth and S. Speiser, On completeness classes for query evaluation on linked data, in: *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*, J. Hoffmann and B. Selman, eds, AAAI Press, 2012, http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/5114.
[28] O. Hartig, P.-A. Champin, G. Kellogg and A. Seaborne, RDF-star and SPARQL-star, W3C Draft Community Group report, W3C, 2022, https://w3c.github.io/rdf-star/cg-spec/editors_draft.html.
[29] A. Hogan, Canonical forms for isomorphic and equivalent RDF graphs: Algorithms for leaning and labelling blank nodes, *ACM Trans. Web* **11**(4) (2017), 22:1–22:62. doi:10.1145/3068333.
[30] I. Jacobs and N. Walsh (eds), Architecture of the World Wide Web, volume one, 2004, W3C Recommendation, https://www.w3.org/TR/webarch/.
[31] F. Kleedorfer, Y. Panchenko, C.M. Busch and C. Huemer, Verifiability and traceability in a linked data based messaging system, in: *Proceedings of the 12th International Conference on Semantic Systems (SEMANTICS)*, ACM, 2016, pp. 97–100. doi:10.1145/2993318.2993342.
[32] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, *Journal of the ACM* **46**(5) (1999), 604–632. doi:10.1145/324133.324140.

[33] T. Kuhn, C. Chichester, M. Krauthammer, N. Queralt-Rosinach, R. Verborgh, G. Giannakopoulos, A.N. Ngomo, R. Viglianti and M. Dumontier, Decentralized provenance-aware publishing with nanopublications, *PeerJ Comput. Sci.* **2** (2016), e78. doi:10.7717/peerj-cs.78.
[34] T. Kuhn and M. Dumontier, Trusty URIs: Verifiable, immutable, and permanent digital artifacts for linked data, in: *Proceedings of the 11th European Semantic Web Conference (ESWC)*, Lecture Notes in Computer Science, Vol. 8465, Springer, 2014, pp. 395–410. doi:10.1007/978-3-319-07443-6_27.
[35] T. Kuhn and M. Dumontier, Making digital artifacts on the web verifiable and reliable, *IEEE Trans. Knowl. Data Eng.* **27**(9) (2015), 2390–2400. doi:10.1109/TKDE.2015.2419657.
[36] E. Mansour, A.V. Sambra, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulnaga and T. Berners-Lee, A demonstration of the solid platform for social web applications, in: *Proceedings of the 25th International World Wide Web Conference (WWW)*, J. Bourdeau, J. Hendler, R. Nkambou, I. Horrocks and B.Y. Zhao, eds, ACM, 2016, pp. 223–226. doi:10.1145/2872518.2890529.
[37] Merriam-Webster, Integrity, 2022, https://www.merriam-webster.com/dictionary/integrity.
[38] National Institute of Standards and Technology, Secure hash standard (SHS), FIPS 180-4, NIST, 2015.
[39] L. Page, S. Brin, R. Motwani and T. Winograd, The PageRank citation ranking: Bringing order to the Web, Technical report, 1999-66, Stanford InfoLab, 1999, Previous number SIDL-WP-1999-0120, http://ilpubs.stanford.edu:8090/422/.
[40] J.X. Parreira, D. Donato, C. Castillo and G. Weikum, Computing trusted authority scores in peer-to-peer web search networks, in: *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval*, ACM International Conference Proceeding Series, Vol. 215, 2007, on the Web (AIRWeb) at the World Wide Web conference (WWW), http://airweb.cse.lehigh.edu/2007/papers/paper_108.pdf.
[41] J.X. Parreira, D. Donato, S. Michel and G. Weikum, Efficient and decentralized PageRank approximation in a peer-to-peer web search network, in: *Proceedings of the 32nd International Conference on Very Large Data Bases*, U. Dayal, K. Whang, D.B. Lomet, G. Alonso, G.M. Lohman, M.L. Kersten, S.K. Cha and Y. Kim, eds, ACM, 2006, pp. 415–426, http://dl.acm.org/citation.cfm?id=1164164.
[42] H. Peters, *Game Theory*, 2nd edn, Springer Texts in Business and Economics, Springer, Berlin, Heidelberg, 2015. doi:10.1007/978-3-662-46950-7.
[43] A. Sambra, E. Mansour, S. Hawke, M. Zereba, N. Greco, A. Ghanem, D. Zagidulin, A. Aboulnaga and T. Berners-Lee, Solid: A platform for decentralized social applications based on linked data, Technical report, MIT CSAIL & Qatar Computing Research Institute, 2016.
[44] A. Sambra, H. Story and T. Berners-Lee, WebID 1.0 – web identity and discovery, W3C Editor's Draft, W3C, 2014, https://www.w3.org/2005/Incubator/webid/spec/identity/.
[45] C. Sayers and A.H. Karp, Computing the digest of an RDF graph, Technical report, HP, 2004, http://www.hpl.hp.com/techreports/2003/HPL-2003-235R1.pdf.
[46] S. Speicher, J. Arwe and A. Malhotra, Linked data platform 1.0, W3C Recommendation, W3C, 2015, https://www.w3.org/TR/ldp/.
[47] M. Sporny, G. Noble, D. Longley, D.C. Burnett, B. Zundel and K. Den Hartog, Verifiable credentials data model, W3C Recommendation, W3C, 2021, https://www.w3.org/TR/vc-data-model/.
[48] A. Sutton and R. Samavi, Integrity proofs for RDF graphs, *Open Journal of Semantic Web (OJSW)* **6**(1) (2019), 1–18, https://www.ronpub.com/ojsw/OJSW_2019v6i1n01_Sutton.html.
[49] A. Third and J. Domingue, LinkChains: Exploring the space of decentralised trustworthy linked data, in: *Proceedings of the Workshop on Decentralizing the Semantic Web 2017 at the 16th International Semantic Web Conference (ISWC)*, CEUR Workshop Proceedings, **1934**, CEUR-WS.org, 2017, http://ceur-ws.org/Vol-1934/contribution-06.pdf.
[50] D.J. Watts and S.H. Strogatz, Collective dynamics of 'small-world' networks, *Nature* **393**(6684) (1998), 440–442. doi:10.1038/30918.
[51] A. Zimmermann, RDF 1.1: On semantics of RDF datasets, W3C Working Group Note, W3C, 2014, https://www.w3.org/TR/rdf11-datasets/.