Legal Entity Identifier blockchained by a Hyperledger Indy implementation of GraphChain

Mirek Sopek[1], Przemysław Grądzki[1], Dominik Kuziński[1], Rafał Trójczak[1,2], and Robert Trypuz[1,2]

[1] R&D Team, MakoLab S.A.
ul. Demokratyczna 46
93-430 Łódź, Poland
sopek@makolab.com
[2] The John Paul II Catholic University of Lublin,
Faculty of Philosophy
Al. Racławickie 14,
20-950 Lublin, Poland

Abstract. The main idea behind GraphChain is to use blockchain mechanisms on top of an abstract RDF graphs. This paper presents an implementation of GraphChain in the Hyperledger Indy framework. The whole setting is shown to be applied to the RDF graphs containing information about Legal Entity Identifiers (LEIs).

Keywords: Hyperledger, Hyperledger Indy, GraphChain, semantic blockchain, LEI, GLEIS

1 Introduction

In this paper are presenting the idea and propose the implementation of Blockchain based data management system that could be used for the Global LEI system (GLEIS). The system preserves all the benefits of using RDF graph data model for the representation of LEI system reference data, including the powerful querying mechanisms, explicit semantics and data model extensibility with the security and non-repudiation of LEI as the digital identifiers for legal entities. To achieve such features we combined the solutions previously invented for the GraphChain[6] with one of the frameworks of the Hyperledger project, namely Hyperledger Indy.

The idea of combining blockchain technology with the Semantic Web principles has been proposed in [2,9,3]. An approach similar to the presented in this paper, albeit not for LEI system, can be found in [4], where Flex Ledger – a graph data model and a protocol for decentralised ledgers – was presented.

We will first introduce the basics of the Legal Entity Identifier (LEI) and the basics of GLEIS - Global LEI System and an ontology we have developed for the LEI system called GLEIO[8,]. Then, after short introduction to Blockchain technology we will explain the rationale of using Blockchain for LEI system and why the idea of GraphChain is important for the goals of our work. Finally, we will present how the use Hyperledger Indy helps to achieve the goals and present some implementation details behind the proposal.

2 Legal Entity Identifier and its ontology

2.1 Legal Entity Identifier

A Legal Entity Identifier, LEI in short, is a 20-digit, alpha-numeric code. It is based on the ISO 17442 standard. It is intended to be the key and unique legal entity reference number enabling straightforward identification of legal entities participating in financial transactions throughout the whole world.

Each LEI is connected to the legal entity reference information that is specified and standardised by means of Common Data File (CDF in short) formats[3]. Currently there are two levels of the LEI reference data. Level 1 provides information about the official name of a legal entity, its legal and headquarters' addresses and the LEI registration data such as LEI initial registration date, next renewal date, etc. Level 2 specifies information about relationships of a legal entity with other legal entities.

The management of the global LEI system (GLEIS) is coordinated and supported by the Global Legal Entity Identifier Foundation (GLEIF)[4]. GLEIF accredits some external organisations to operate within the GLEIS as issuers of LEIs [5]. LEI issuers are called Local Operating Units (LOUs).

For the sake of further analysis, it is important for us to understand the whole process of LEI registration and maintenance. At the beginning of the process, through self-registration, the registering legal entity supplies its reference data (such as its legal name, address or the business identifier in the jurisdiction of its legal registration). The LOU verifies the reference data with local authoritative sources and issues an LEI. Every day each LOU reports all the LEIs with their reference data to GLEIF by means of XML Concatenated Files (standardised by CDF formats as was described above). Also GLEIF publishes daily updated XML Concatenated Files including all current LEIs, their reference data and description of relationships between legal entities.

Summarising, in the whole process of LEI registration and maintenance we may distinguish the following roles:

- users (legal entities) – requesting for LEIs and supplying their reference data
- LOUs – responsible for issuing LEIs and verifying the reference data provided by legal entities
- GLEIF – the organisation that accredits LOUs, collects and publishes all LEI reference data daily.

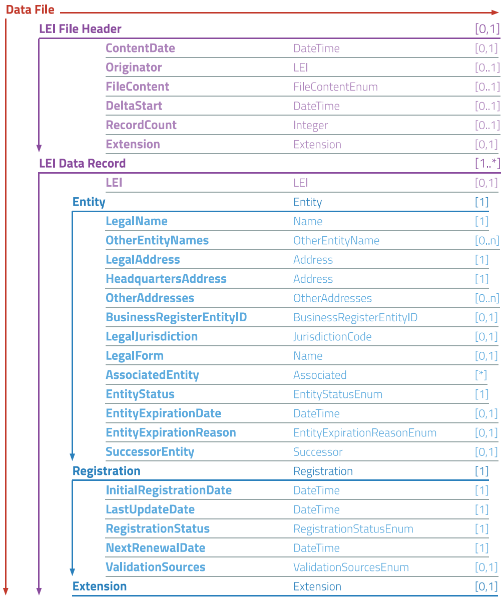
[3] https://www.gleif.org/en/about-lei/common-data-file-format

[4] www.gleif.org

[5] We shall not describe in details the whole process of accreditation by which GLEIF evaluates the suitability of organizations seeking to operate within the GLEIS as LOUs, see https://www.gleif.org/en/about-lei/gleif-accreditation-of-lei-issuers/accreditation-process.

2.2 Global LEI Ontology

LOUs report LEIs and their reference data by means of XML files compliant with the CDF format. For instance the XML schema for LEI level 1 consists of LEI File Header and LEI Data Record as depicted in figure 1.



Data File		
LEI File Header		[0,1]
ContentDate	DateTime	[0,1]
Originator	LEI	[0..1]
FileContent	FileContentEnum	[0..1]
DeltaStart	DateTime	[0..1]
RecordCount	Integer	[0..1]
Extension	Extension	[0,1]
LEI Data Record		[1..*]
LEI	LEI	[0,1]
Entity	Entity	[1]
LegalName	Name	[1]
OtherEntityNames	OtherEntityName	[0..n]
LegalAddress	Address	[1]
HeadquartersAddress	Address	[1]
OtherAddresses	OtherAddresses	[0..n]
BusinessRegisterEntityID	BusinessRegisterEntityID	[0,1]
LegalJurisdiction	JurisdictionCode	[0,1]
LegalForm	Name	[0,1]
AssociatedEntity	Associated	[*]
EntityStatus	EntityStatusEnum	[1]
EntityExpirationDate	DateTime	[0,1]
EntityExpirationReason	EntityExpirationReasonEnum	[0,1]
SuccessorEntity	Successor	[0,1]
Registration	Registration	[1]
InitialRegistrationDate	DateTime	[1]
LastUpdateDate	DateTime	[1]
RegistrationStatus	RegistrationStatusEnum	[1]
NextRenewalDate	DateTime	[1]
ValidationSources	ValidationSourcesEnum	[0,1]
Extension	Extension	[0,1]

Fig. 1. Visualisation of XML schema for the CDF format of level 1

In the paper [8] we have described some advantages of semantic representation of LEIs – such as precise semantics, flexible extensibility, global persistent identifiers and inference – in contrast to the current XML representation. Moreover, we have proposed Global LEI Ontology (GLEIO in short)[6] that is not only compliant with CDF Format but also allows for representation of changes of LEI and related LEI reference data over time. GLEIO provided meaning for LEI data, shaped the LEI triple store and allowed for SPARQL queries[7].

The methodology and the whole process of GLEIO creation was described in [8, section 2.2]. A need for expressing change of LEIs and their reference data in time was as a very important modelling requirement while creating GLEIO. The changes we had in mind considered for instance a change of LEI registration status or the change of the legal address of an entity. As described in section 2.1 GLEIF publishes the XML concatenated file daily. So every day we have a new snapshot of all LEIs. GLEIO allows for explicit representation of the LEIs'

[6] `http://lei.info/gleio/`

[7] `https://lei.info/sparql`

snapshot changes by means of manifestations. A manifestation of an entity is a complete picture of the entity in a given time stamp (snapshot). Each LEI may have one or many (linearly ordered) manifestations.

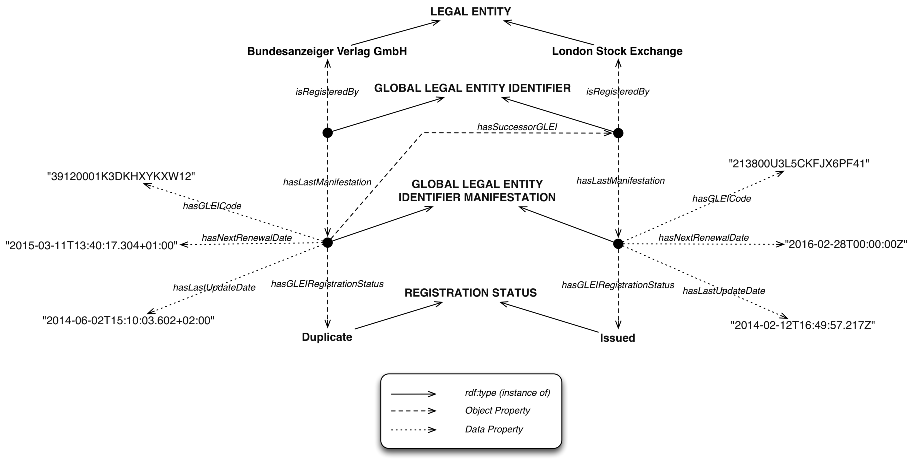


Fig. 2. Variable entity and its manifestations

GLEIO is part of SaaS (Software as a Service) application accessible at: `http://lei.info` – allowing for storing, displaying and integrating information about LEIs (It is also informational and educational portal about the GLEIS). Every day our LEI application reads a complete concatenated XML file and creates new manifestations for those LEIs that have changed since the last check had been performed (for instance if the status of LEI registration has changed from "Duplicate" to "Issued", then the LEI will gain a new manifestation – see figure 2).

Since we have eventually decided to create a chain of graphs in the blockchain spirit (our motivation is explained in the forthcoming section), manifestations have not been needed any more. Its role/function have been replaced by transactions/blocks of data. Also the linear order of manifestations has been replaced by the order of transactions in the ledger chain. A new GLEIO without manifestations can be browse here: `https://lei.info/voc/`.

3 Towards blockchained LEIs

3.1 A short introduction to blockchain

The history of blockchain started ten years ago when Satoshi Nakamoto published "Bitcoin: A Peer-to-Peer Electronic Cash System" [5].

Blockchain is commonly associated with cryptocurrencies and cash transactions. But it was just the first phase of its evolution named “Blockchain 1.0” in [7]. The next blockchain evolution phase, Blockchain 2.0, proposed applications that offered smart contracts and could deal with complex financial products such as: bonds, loans or mortgages. Nowadays, we have Blockchain 3.0 offering applications going “beyond currency, finance, and markets—particularly in the areas of government, health, science, literacy, culture, and art.” [7]

Blockchain is usually defined by referring to its structure and function. In ([1]) we find the following general definition:

> A block chain is a type of database that takes a number of records and puts them in a block (...). Each block is then ‘chained’ to the next block, using a cryptographic signature. This allows block chains to be used like a ledger, which can be shared and corroborated by anyone with the appropriate permissions.

Since we are still dealing with relatively young technology it is very difficult to classify the variety of blockchains. However there are two criteria that make it a lot clearer. These criteria are following blockchain deployment styles: public/private and permissioned/permissionless [10].

Public blockchains are open for everyone who would like to join them. They are also at least “read open”. The private ones have limited access that is controlled by selected nodes. These nodes can control whether the blockchain is read-only or of read-write access.

In the permissioned blockchains there are parties that can assign different permissions/rights to the clients concerning the kinds of transactions they can carry out. Permissioned blockchains can also form “federated” or “consortium” blockchains. They may give the public right to read the blockchain. Permissionless blockchains do not have different permission levels.

Bitcoin, the first implementation of blockchain, is public and permissionless. We can observe the tendency to name only these sort of implementations “blockchains”. Other ones (i.e. private or permissioned blockchains) are more commonly named “distributed ledger technologies”. Sometimes “distributed ledger technology” names also public and permissionless blockchains.

In the permissionless and public blockchains the process of transaction validation is open for everyone. In the permissioned blockchains a node must satisfy certain criteria to gain permission for transaction validation. In the permissioned and private blockchain this means to become a member of the consortium.

3.2 Why using blockchain for LEI

From the very beginning of its existence GLEIF has not been thought of as a hub for collecting LEIs and their reference data. Its primary, and in fact only role, was management of “a network of partners, known as the LEI issuing organizations, to provide trusted services and open, reliable data for unique legal entity identification worldwide.”[8].

[8] See `https://www.gleif.org/en/about/this-is-gleif/`

Now keeping in mind what we have written in section 2 about GLEIS, it is evident that GLEIF together with a network of LOUs and legal entities interested in getting LEIs creates a distributed system that can be adequately modelled as a permissioned (consortium) blockchain. From the high level view the proposed in this paper a new LEI system assumes the use of permissioned blockchain with three types of roles (see figure 3):

- User nodes and registars – the nodes participate in the global blockchain as passive users; they can see all the data stored in it, but can't create nor edit anything. Registars may have additional ability to "provisionally" add new LEIs to the system. However, such newly added LEIs are not visible on the system until the LOU nodes confirm them through the "Proof of Authority" mechanism.
- LOU nodes – the nodes having all the properties of the Registration nodes plus the capacity to confirm the new or modified LEIs as valid.
- GLEIF – the node possessing all permissions

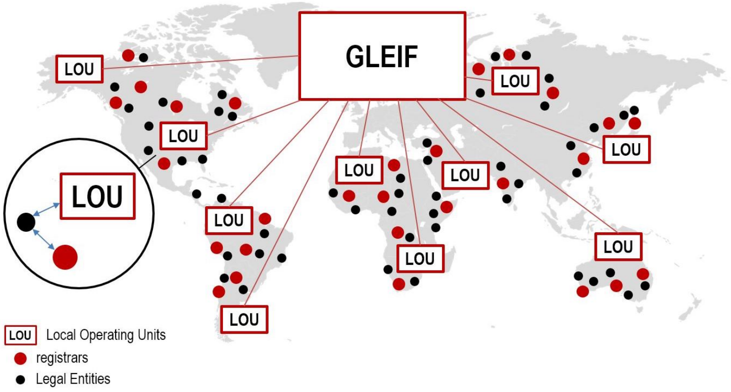


Fig. 3. GLEIS architecture

Such an architecture of the new LEI system enables thousands of registration authorities from multiple countries to participate in the new LEI creation, opening path for the true global adoption of the system.

There are many advantages of using blockchains (or distributed ledgers) in modern business applications. For the identification purposes (which is the primary function of the LEI system) the most important benefits are: non-repudiation of identities and transactions, immutability of LEI data, decentralisation of LEI issuing and distribution process, lowering the LEI issuing costs, transparency to internal stakeholders and regulators, resilience to system failures, efficient replication mechanisms, far-reaching democratisation of digital identifiers generation, ability to restrict generation of identifiers to authorised agents or institutions, diversification of targets: institutions, legal and real persons, datasets and devices.

3.3 GraphChain

In [6] we presented an idea of GraphChain – a linearly ordered collection (a chain) of cryptographically secured named RDF graphs on which all nodes eventually agree. [6] describes the first preliminary (PoC) realisation of GraphChain. We described OWL-compliant GraphChain ontology that specifies all the structural, invariant elements of the GraphChain and defines their basic semantics. We have also presented some general mechanisms for calculating a digest of the named RDF graphs and some simple (naive) network mechanisms that are responsible for the distribution of the named RDF graphs among the distributed peers and for achieving the consensus.

The main idea behind GraphChain is to use blockchain mechanisms on top of an abstract RDF graph data model. In the next section we will describe how to build a distributed ledger of LEI graphs shaped by GLEIO ontology.

3.4 LEI in Hyperledger Indy

For our purposes, we have chosen Hyperledger Indy, as it is a distributed ledger built to be used for decentralised identity management[9]. The key assumption of Indy architecture is that the content (in our case LEI) data is never written to the ledger. This makes it the ideal choice for realising our idea of using blockchain mechanisms on top of triple store of LEIs.

Hyperledger Indy is a public and permissioned blockchain – Indy permits registered members to manage (write) their self-sovereign identity and everyone to read the content of the blockchain.

The ledger is maintained by the nodes, which run Plenum Byzantine Fault Tolerant Protocol, i.e. (a consensus protocol based on Redundant Byzantine Fault Tolerant) to agree on the order of transactions in the ledger. Pairwise Pseudonymous Identifiers and Decentralised Public Key Infrastructure (using asymmetric key cryptography) guarantee full privacy, prevent identity correlation and ensure that connections between the members of the system (nodes and clients) are established in secure, encrypted manner.

Another fundamental feature of Indy is using Decentralised Identifiers (DIDs)[10] – a new type of identifiers for "self-sovereign" digital identity. DIDs, as the primary keys on Indy ledger, enable long-term digital identities requiring no centralised registry services. DIDs are stored in the Indy ledger as NYM records. Our first idea was to put LEI graph as attribute(s) added to NYM records corresponding to legal entities by using `ATTRIB` transaction. But there are a few problems related to the current implementation of Hyperledger Indy:

1. the maximum size of the data (represented as JSON) added as a attribute is limited to `5 * 1024` bytes

[9] https://www.hyperledger.org/projects/hyperledger-indy

[10] https://w3c-ccg.github.io/did-spec/

2. permissions assigned to roles[11] are not the same as permissions required by LEI system
3. attributes are not "searcheable" – there is no implemented request to execute full-text search based on the content of attributes

The second idea, eventually realised by us, was to add a new transaction type: `ADD_LEI`. The transaction type is in fact "enhanced `ATTRIB`" transaction free of the limitations mentioned above. Whereas the raw data from `ATTRIB` transaction is stored in a separate attribute ledger—the key-value database (currently RocksDB), where the key is the hash of raw data and value is raw data[12]—our `ADD_LEI` transaction keeps data in the external searchable triplestore (see figure 4). In `ADD_LEI` transaction, the key is constructed using prefix "`http://lei.info/`" concatenated with the hash of raw data. It forms IRI that is used as the identifier of the named graph:

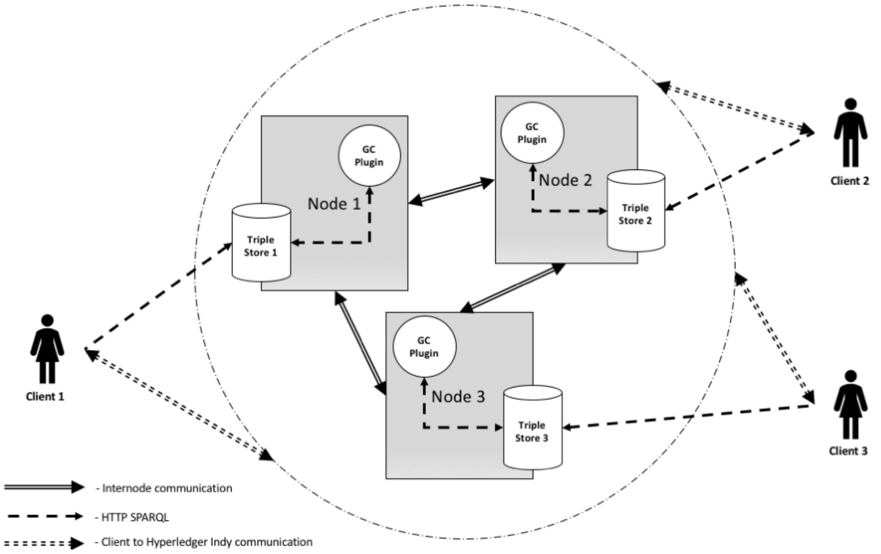


Fig. 4. The architecture of Hyperledger Indy with Triple Store

```
http://lei.info/<I-HASH>
```

Functionality of the `ADD_LEI` transaction is implemented as "GC plugin" inside the Indy nodes (see figure 4). The plugin performs the following tasks:

[11] https://docs.google.com/spreadsheets/d/1TWXF7NtBjSOaUIBeIH77SyZnawfo91cJ_ns4TR-wsq4

[12] https://github.com/hyperledger/indy-node/blob/stable/docs/transactions.md#attrib

– validates ADD_LEI transactions,
– checks users and nodes permissions to carry out transactions,
– reads and writes named graphs from/to the triple store.

Triple stores are external to the Indy nodes and allow clients to perform SPARQL queries.

3.5 Interwoven hash verification

In our framework, LEIs data are stored in the triple store within the named graphs. The named graphs' IRIs are made up as a concatenation of prefix "http://lei.info/" and the value of the Interwoven hash function[13] "<I-HASH>" calculated for the content of the named graph. The IRIs have the following shape: http://lei.info/<I-HASH>.

Since LEI data can be accessed directly from a triple store of a node through HTTP SPARQL query, one may wonder how to prove integrity of data in the triple store. Below we describe how to validate an LEI graph for a legal entity having legal name "EXAMPLE COMPANY LEGAL NAME".

First run SPARQL query that returns IRI of the named graph:

Listing 1.1. Get IRI of the graph

```
PREFIX l1:<http://lei.info/voc/l1/>
SELECT ?g
WHERE {
   GRAPH ?g {
      [ l1:hasLegalName "EXAMPLE COMPANY LEGAL NAME"] .
   }
}
```

The result will be IRI of the form "http://lei.info/<I-HASH>". Then run SPARQL Query that returns the content of the named graph:

Listing 1.2. Get the content (triples) of the graph "http://lei.info/<I-HASH>"

```
CONSTRUCT { ?s ?p ?o }
WHERE
 {
  GRAPH <http://lei.info/<I-HASH>>
  { ?s ?p ?o } .
}
```

In the next step, extract <I-HASH> part of the named graph IRI and compare it with the value of the Interwoven hash function calculated for the content of the graph http://lei.info/<I-HASH>. If they are not equal – verification fails, else check, if the <I-HASH> value is stored in Hyperledger Indy using request:

```
send GET_LEI ihash=<I-HASH>
```

If it's missing – verification fails. If they are equal, we have certainty of both the existence of an entity identified by such an IRI and the veracity of the retrieved data.

[13] The algorithm for calculating interwoven hash was described in [6, section 4.3.3].

4 Conclusions

In this paper we have presented the idea and the implementation details of the proposed Blockchain based LEI system. This idea assumes the use of our GraphChain solution for the storage of LEI reference data and Hyperledger Indy for distributed, secured identification mechanisms and consensus about data.

As a future work we consider adding the meta-data of extracted from Indy's transactions (such as time stamp, sequence number, etc) to the triple store.

As a challenge also remains the question of storing data, which size exceeds the limit of the message size `128 *1024` (`MSG_LEN_LIMIT`[14]). As an option we consider cutting a line-based serialisations of RDF graphs, such as for example N-Triples, into sub-graphs that would fit the limit of size. Of course this solution would require a description of the sub-graphs in a way enabling their later "glueing" into the original graph.

Another benefit from the proposed architecture is its potential alignment to the digital identification system proposed by Sovrin Foundation [15]. The foundation system realizes the idea of Self-Sovereign Identity. The combination of such model of digital identity and the LEI has already been considered by us for the future of the LEI system which could enable legal entities to undeniably and uniquely identify themselves in digital space.

Acknowledgements. Our research was conducted in the project number 01.01.01-00-0982/16 funded by the National Centre for Research and Development under the Smart Growth Operational Programme.

References

1. UK Government Chief Scientific Adviser. Distributed Ledger Technology: beyond block chain. Technical report, 2016.
2. M. English, S. Auer, and S.Domingue. Blockchain technologies & the Semantic Web: A framework for symbiotic development. In J. Lehmann, H. Thakkar, L. Halilaj, and R. Asmat, editors, *CS Conference for University of Bonn Students*, pages 47–61, 2016.
3. Elena García-Barriocanal, Salvador Sánchez-Alonso, and Miguel-Angel Sicilia. Deploying metadata on blockchain technologies. In Emmanouel Garoufallou, Sirje Virkus, Rania Siatri, and Damiana Koutsomiha, editors, *Metadata and Semantic Research*, pages 38–49, Cham, 2017. Springer International Publishing.
4. Victoria L. Lemieux and Manu Sporny. Preserving the archival bond in distributed ledgers: A data model and syntax. In *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, pages 1437–1443, 2017.
5. Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Technical report, 2008.

[14] `https://github.com/hyperledger/indy-plenum/blob/master/stp\_core/config.py`

[15] `https://sovrin.org/`

6. Mirek Sopek, Przemyslaw Gradzki, Witold Kosowski, Dominik Kuziski, Rafa Trójczak, and Robert Trypuz. Graphchain: A distributed database with explicit semantics and chained rdf graphs. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 1171–1178, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.
7. Melanie Swan. *Blockchain: Blueprint for a New Economy.* O'Reilly Media, Inc., 1st edition, 2015.
8. Robert Trypuz, Dominik Kuzinski, and Mirek Sopek. General legal entity identifier ontology. In Oliver Kutz, Sergio de Cesare, Maria M. Hedblom, Tarek Richard Besold, Tony Veale, Frederik Gailly, Giancarlo Guizzardi, Mark Lycett, Chris Partridge, Oscar Pastor, Michael Grüninger, Fabian Neuhaus, Till Mossakowski, Stefano Borgo, Loris Bozzato, Chiara Del Vescovo, Martin Homola, Frank Loebe, Adrien Barton, and Jean-Rémi Bourguet, editors, *Proceedings of the Joint Ontology Workshops 2016 Episode 2: The French Summer of Ontology co-located with the 9th International Conference on Formal Ontology in Information Systems (FOIS 2016), Annecy, France, July 6-9, 2016.*, volume 1660 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.
9. Hector Ugarte. A more pragmatic Web 3.0: Linked Blockchain Data. Technical report, https://www.researchgate.net/publication/315619465_A_more_pragmatic_Web_30_Linked_Blockchain_Data, March 2017.
10. Shermin Voshmgir and Valentin Kalinov. Blockchain, a beginners guide. Technical report, BlockchainHub, September 2017.