A Blockchain Infrastructure for the Semantic Web of Things

Michele Ruta*, Floriano Scioscia, Saverio Ieva,
Giovanna Capurso, Agnese Pinto, and Eugenio Di Sciascio

Polytechnic University of Bari, via E. Orabona 4, Bari (I-70125), Italy
name.surname@poliba.it

Abstract. The Semantic Web of Things (SWoT) improves the Internet of Things power by increasing resource representation capabilities through knowledge management and reasoning technologies adapted from the Semantic Web. This promotes information interoperability and decision autonomy. Nevertheless, trust and reliability issues remain basically unsolved. Large-scale, decentralized and dynamic infrastructures suffer from unpredictable volatility of nodes, which compromises resource availability. Trust and coordination are still difficult. Blockchain is increasingly used as a transactional data storage solution for distributed ledgers. It enables trustless collaboration by enforcing smart contracts and prevents data tampering by validating transactions through consensus protocols. This paper proposes a blockchain framework for SWoT contexts settled as a Service-Oriented Architecture. Nodes can exploit smart contracts for registration, discovery and selection of annotated services/resources. While semantic matchmaking enables relevant resource retrieval with logic-based ranking and explanation features, blockchain provides reliable transaction storage. A prototype has been developed by enhancing the standard *Hyperledger Iroha* framework. Application areas are discussed and experimental tests on a cluster of virtual nodes provide early insight on effectiveness, performance and scalability.

Keywords: Semantic Web of Things · Distributed transactional systems · Blockchain · Smart contracts · Service-Oriented Architectures · Semantic matchmaking

1 Introduction

The *Semantic Web of Things* (SWoT) [10] vision extends the Semantic Web initiative to the Internet of Things (IoT) by improving the representation capability of objects and environments through annotating them with semantically rich languages. Knowledge representation models and languages defined for the Semantic Web can provide the basic substrate for interoperable information modeling and sharing in the IoT. Machine understandability of adopted formalisms

* Corresponding author. Tel.: +39-339-635-4949; fax: +39-080-596-3410
SEBD 2018, June 24-27, 2018, Castellaneta Marina, Italy. Copyright held by the author(s).

allows applying automated inferences so that heterogeneous micro-devices, each conveying a small amount of information, can interact autonomously to provide high-level services to users, via decision support and task automation.

Anyway, IoT suffers from unpredictability of node and resource availability, due to the volatility of actors and appliances. This makes trust and coordination management difficult and these limits are inevitably inherited by the SWoT: their burden is particularly evident when reliable and trustworthy applications are needed. Blockchain technology is interesting from this perspective. In conventional distributed databases, a trusted intermediary is needed to guarantee irreversibility (*i.e.*, committed transactions cannot be altered or reverted) and prevent censorship (*i.e.*, all valid transactions are committed). *Blockchain* is a data structure and protocol for *trustless* distributed transactional systems: they avoid intermediaries by approving transactions through a distributed *consensus* protocol, which guarantees no single node –or small group of colluding nodes– can force the addition, removal or modification of data [2]. Blockchain could incorporate SWoT approaches providing interesting possibilities for large-scale distributed trustless systems. A SWoT blockchain basically amounts to a Service-Oriented Architecture (SOA) for regulating registration, discovery and selection operations. These tasks are intended as distributed and validated by consensus.

This paper proposes a framework for SWoT blockchain systems. A semantic-based resource discovery layer is integrated in a basic blockchain infrastructure, adding verifiable records for every single transaction. A distinguishing feature of the systems is *logic-based explanation* of discovery outcomes, grounded on non-standard inference services for semantic matchmaking [11]. The proposed system preserves fundamental blockchain features, also when size increases. Particularly, the effective and secure structure of the chain is capable of detecting erroneous or malicious changes on a transaction block, also in case of large amounts of volatile nodes. Annotations are registered as assets on the chain and non-standard inferences in [11] are executed by validating peers. The proposed framework has been implemented and tested on the *Hyperledger Iroha*[1] platform. It has been encapsulated in a cluster of *Docker*[2] containers to simulate a large SWoT infrastructure. Experiments on the framework assess the feasibility of the approach.

The remainder of the paper is reported hereafter. Section 2 recalls relevant background, while a functional and architectural description of the proposed framework is given in Section 3. Experimental results are provided in Section 4, followed by an overview of application areas for the approach in Section 5. Conclusion closes the paper.

2 Background

In what follows some relevant background is surveyed in order to make the paper self-contained.

[1] https://www.hyperledger.org/projects/iroha
[2] https://www.docker.com/

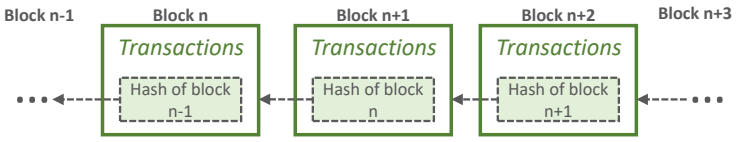

Fig. 1. Blockchain structure

2.1 Blockchain

The blockchain technology was born in 2009 with *Bitcoin*, an open source platform for electronic currency exploiting previous theoretical results on *Proof-of-Work* consensus algorithms [13]. Bitcoin relies on blockchain as a ledger of currency transfers. In a blockchain, transactions approved in a given time window are grouped in *blocks*. Figure 1 shows each block also contains a hash, computed from both the content of the block and the hash of the previous block. This forms a chain of blocks, preventing tampering even with old blocks without node agreement. Transactions are validated through consensus algorithms where all network nodes behave as peers, holding a copy of current blockchain status.

Many blockchain-based e-currencies have been proposed in latest years due to the worldwide success of Bitcoin. Furthermore, finance, industry and research communities have been increasingly experimenting with blockchain as a general-purpose distributed database enabling practical implementations of *Smart Contracts* (SCs) [12] *i.e.*, programs encoding and enforcing multi-party agreements and coordinated activities. While a trusted mediator is required in the original SC concept –restraining real-world deployments– blockchain consensus-based approach allows trustless collaboration of *Decentralized Autonomous Organizations* (DAOs) through parallel execution of SCs among network nodes. Every SC-enabled blockchain can thus be seen as a general-purpose application platform based on a distributed Virtual Machine (VM): emerging proposals include proprietary platforms (*e.g.*, *Ethereum*[3]) and standardization efforts, such as the *Hyperledger*[4] initiative steered by the Linux Foundation.

From a technical viewpoint, blockchain systems can be classified based on the following core design decisions:

– **Network access policy.** A blockchain network is *permissioned* if a white-list of allowed nodes exists and nodes are uniquely identified, or *permission-less* if any node can join at any time, even anonymously.

– **Consensus algorithm.** Permission-less systems require stricter consensus methods, such as Proof-of-Work, which is based on solving a cryptographic problem and guarantee data security unless a large portion of nodes is colluding to subvert the blockchain. Permissioned systems –where each node is accountable– may relax consensus constraints in order to reduce the computational load, by selecting simpler algorithms. State-machine replication protocols such as Byzantine Fault Tolerance (BFT) variants [13] are often adopted. The Iroha framework

[3] Ethereum Project: https://www.ethereum.org/

[4] Hyperledger: https://www.hyperledger.org/

adopted for the experiments of this paper uses a BFT consensus algorithm named *Sumeragi*.
– **Transaction model.** In typical blockchain systems, *assets* are registered and exchanged, so that at any time each node owns some assets in a given quantity. In the *Unspent Transaction Outputs* (UTXO) model, a transfer from A to B *consumes* (*i.e.*, deletes) records for A's spent assets and *produces* (*i.e.*, adds) new ones for B's received assets. In the *account-based* model, instead, every node has an account reporting all its assets, which is updated by transactions.
– **Smart contract language.** Blockchains can adopt any formalism for SC specification and execution, such as procedural (imperative) languages or logical (declarative) languages or automata [4]. Current proposals mostly adopt computationally complete programming languages, either existing (*e.g.*, Java in the *Iroha* framework of Hyperledger, exploited in this work) or created for the purpose (*e.g.*, Ethereum's *Solidity*).

Internet of Things (IoT) scenarios are expected to be among the main application areas in the near future. Blockchain applications in IoT contexts are mostly focused on supply chain [7] and Industry 4.0 [3]. The integration of Semantic Web technologies in blockchain has been discussed *e.g.*, in [3, 6], while [4] proposed *defeasible logics* for SC definition and implementation. Nevertheless, to the best of our knowledge this paper presents the first full blockchain framework integrating logic-based resource discovery.

2.2 Semantic Web of Things

Knowledge representation and reasoning in the SWoT are formally grounded on Description Logics (DLs), a family of logic languages in a decidable fragment of First Order Logic [1]. Basic DL syntax elements are: *concepts* (a.k.a. *classes*), standing for sets of objects; *roles* (a.k.a. *object properties*), linking pairs of objects in different concepts; *individuals* (a.k.a. *instances*), special named elements belonging to concepts. Each DL has a specific set of logical *constructors* for combining the above elements in concept and role *expressions*. Concept expressions can be used in *inclusion* (a.k.a. *subsumption*) and *definition* (a.k.a. *equivalence*) axioms, which model knowledge elicited for a given domain. A set of such axioms is called *Terminological Box (TBox)*, a.k.a. *ontology*. A set of individual axioms (a.k.a. *facts*) constitutes an *Assertion Box (ABox)*. TBox and ABox together make up a *Knowledge Base (KB)*.

The approach proposed in this paper leverages *semantic matchmaking*, *i.e.*, the retrieval of the most relevant resources for a given request, where both resources and requests are annotated with concept expressions w.r.t. a common ontology $\mathcal{T}$. Given a request R and a resource S, *subsumption* checks whether all features in R are included in S; *satisfiability* checks whether any constraint in R contradicts some specification in S. Unfortunately these classic inference services enable only a Boolean "full match or no match" approach. This is inadequate in complex scenarios, because full matches are rare and incompatibility is frequent when dealing with detailed resource descriptions. Therefore, non-standard inferences in [11] help to determine a semantic ranking of resources w.r.t. a request

and a logic-based explanation of outcomes:
– **Concept Contraction**: if request R and resource S are not compatible, Contraction determines which part of R is conflicting with S. If one retracts conflicting requirements in R, G (for *Give up*), a concept K (for *Keep*) is obtained, representing a contracted version of the original request, such that K and S are compatible w.r.t. $\mathcal{T}$. G represents "why" R and S clash.
– **Concept Abduction**: when R and S are compatible but S is not a full match for R, Abduction determines what should be hypothesized in S in order to completely satisfy R. The solution H (for *Hypothesis*) to Abduction is a concept representing "why" the subsumption relation does not hold. H can be interpreted as *what is requested in R and not specified in S*.
If R and S are incompatible, the matchmaking process uses Contraction to extract the compatible part K and then Abduction to obtain the required H_K for reaching a full match. Furthermore, *penalty functions* are computed based on the structure and number of elements in G and H_K [11], defining a well-founded *semantic distance* metric, which can be used to rank a set of resources by relevance (*i.e.*, semantic affinity) w.r.t. the request.

In SWoT contexts, computation resources of devices are strictly constrained and require careful software design. Adding more constructors makes DL languages more expressive, but leads to an increase in computational complexity of inference services, hence a tradeoff is needed. This paper refers to the moderately expressive *Attributive Language with unqualified Number restrictions* ($\mathcal{ALN}$) DL, which grants polynomial complexity to all the above standard and non-standard inferences.

3 Semantic-enhanced Blockchain for unpredictable scenarios

The proposed framework defines a semantic-based discovery layer built upon a standard blockchain system, retaining full backward compatibility.

3.1 Framework architecture

Figure 2 sketches the overall architecture, whose main features are reported hereafter.

– **Peer** agents registered in the blockchain are identified by their *public keys* and associated with *accounts*. Each agent can perform a semantic-based resource discovery in order to take ownership and transfer assets between accounts.
– **Assets** are annotated w.r.t. a domain ontology. They can represent physical or digital resources, as well as service instances. Peers can register assets through specific transactions, and annotations are stored in the blockchain.
– **Smart contracts** are enhanced by exploiting non-standard inference services described in Section 2.2. Particularly, each peer locally integrates an

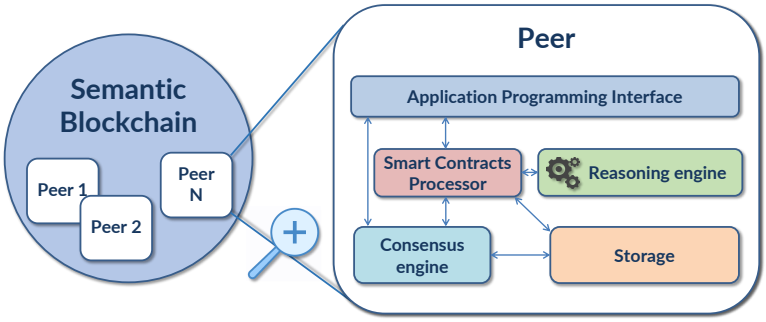

Fig. 2. Framework architecture

embedded matchmaking and reasoning engine, allowing semantic-based resource discovery [11].
- **Consensus engine** tracks and validates semantic-based transactions in a standard and transparent way.
- **Storage** for all transactions, including semantic ones, exploits *Merkle trees*, a data structure commonly adopted in blockchains for efficient detection of erroneous or malicious changes on a transaction block.

3.2 How to make smarter contracts: put semantics in the consensus protocol

The SWoT blockchain proposal is intended as a SOA enabling fundamental operations as registration, discovery, selection and final execution (payment). Such tasks are implemented as smart contracts, *i.e.*, their accomplishment is distributed and validation is reached by consensus. The distinguishing feature of the proposed approach w.r.t. basic blockchain infrastructures is the integration of advanced resource/service discovery grounded on semantic matchmaking. It allows computing a score measuring the semantic distance between the request and each available chain resource, both described w.r.t. the same domain ontology. This logic-based metric induces a relevance-ranked list of annotated elements w.r.t. the request. The non-standard inference services also provide a formal *explanation* of discovery outcomes, reinforcing user confidence in the discovery process. SOA primitives and corresponding SCs are outlined hereafter.

A. Resource registration. Multiple resource domains can coexist and be tracked in the same blockchain. Each domain is associated to a different ontology, providing the reference conceptual vocabulary for annotating resources. Resource instances are characterized by the following attributes:

- a Uniform Resource Identifier (URI) featuring the resource unambiguously (and possibly denoting its fruition endpoint, although this is not mandatory);

- a semantic annotation in Web Ontology Language (OWL 2)[5] describing the resource;
- the URI of the reference ontology, in order to cope with the co-existence of several resource domains in the same blockchain;
- a resource price: the framework supports any type of currency unit, pecuniary or otherwise (*e.g.*, in SWoT applications energy or time could be chosen).

By means of the `Registration` SC the owner node will register a resource as an asset on the blockchain storage, in order to make it available for discovery and usage.

B. Resource discovery. In order to search for a (set of) item(s), the requester randomly selects n peers and sends a multicast request specifying:

- URI of the reference ontology: this determines the resource domain, *i.e.*, the vocabulary used to express annotations of request and resources to be retrieved; the nodes receiving a request will not process resources annotated w.r.t. other ontologies in the semantic matchmaking;
- OWL semantic annotation of the request, specifying desired resource features and constraints;
- maximum price p_{max} the requester is willing to pay; resources with a price higher than this threshold will be skipped from matchmaking in order to reduce computational overhead;
- minimum semantic relevance threshold s_{min}, as a floating-point number in the $[0, 1]$ range, where 1 corresponds to a full match and 0 to a complete mismatch (both rare situations in realistic scenarios); after matchmaking, resources with a relevance score below this threshold will not be returned, as deemed irrelevant to the requester;
- maximum number of results r_{max} to be returned;
- requester's address.

The proposal adopts a *gossip-based* (a.k.a. *epidemic*) approach [5] to disseminate discovery requests and aggregate results as in Figure 3. This grants protocol simplicity and low computational overhead, which is a primary requirement in SWoT contexts. Nodes receiving the request execute semantic matchmaking of it with their own resources through an on-board matchmaking engine [11], implementing the non-standard inference services in Section 2.2. A list of at most r_{max} results satisfying both semantic relevance and price constraints is returned, ranked by relevance. Nodes also select other n random peers and forward the request. Nodes receiving forwarded requests behave in the same way, up to a search depth threshold m. Each queried node returns results directly to the original requester at the specified address.

C. Explanation. This is an optional step in a typical discovery process, invoked when a requester needs a justification of the matchmaking outcome. This

[5] OWL 2 Web Ontology Language Document Overview (Second Edition), W3C Recommendation 11 December 2012, http://www.w3.org/TR/owl2-overview/

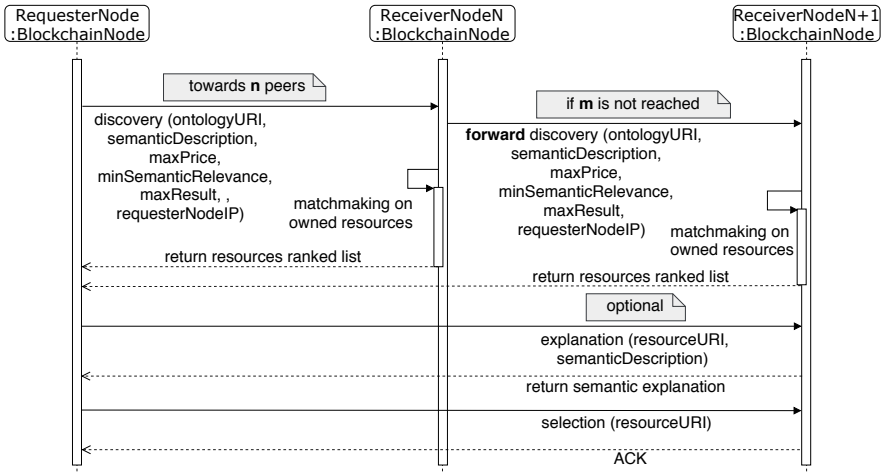

Fig. 3. Resource discovery and retrieval flow

can be useful, *e.g.*, to trigger a request refinement process [9], as it can make the requester aware of further features it did not include in its original request but it may be interested in. The requester node sends a unicast request containing: (i) the semantic annotation of the request; (ii) the URI of the discovered resource. The receiving node replies with the matchmaking outcome explanation, structured as: (i) semantic affinity score in the $[0, 1]$ interval; (ii) concept expressions of G and K from Concept Contraction and of H from Concept Abduction. G, K and H can be cached from the previous discovery step, if storage of the resource owner allows it.

D. Resource selection. After receiving all results –or just a subset, if the response delay of some nodes is greater than a fixed *timeout*– the requester selects the best resource(s) by means of `select` SC. A unicast message is delivered to the resource owner specifying the resource URI and a currency payment. The recipient answers with a properly usable resource representation: this depends on the actual kind of resource and meaning of the URI, *e.g.*, an interface endpoint to access a networked device or a further SC to be invoked. The proposal does not constrain resource fruition in any way, leaving application-specific details to the semantic annotation of the resources themselves.

The resource discovery and retrieval interaction sequence is shown in Figure 3. Each associated transaction is recorded on the blockchain for robustness, traceability and accountability purposes.

4 Performance assessment

In order to assess effectiveness and scalability of the proposed approach, an experimental evaluation campaign has been carried out starting from the *Iroha* framework from Hyperledger. The implemented prototype enhanced Iroha as in

what follows:
– the server API has been extended with support for semantic matchmaking;
– the SCs described in Section 3.2 have been implemented and the *Mini-ME* reasoning and matchmaking engine [11] has been integrated;
– *zlib*[6] compression library has been exploited to cope with the well-known verbosity of ontology languages as OWL.

In order to reach a quantitative performance analysis, small, medium and large scale chain scenarios have been considered, respectively with 50, 150 and 500 nodes. In each of them nodes have been split in two sets materially executing SC transactions with semantic-based discovery (see Section 3.2): *producers*, *i.e.*, providers of annotated resources, registered in the blockchain; *consumers*, *i.e.*, resource requesters.

The following experiment parameters have been set: (i) duration of 300 s; (ii) consumer/producer ratio of 0.1; (iii) 20 randomly-generated annotations per producer; (iv) each consumer sends a new randomly-generated request every 10 s; (v) each request can be forwarded to 4 nodes at most; (vi) a request is aborted if no match is found after 2 hops; (vii) the minimum threshold of semantic affinity is 0.9. Each scenario is executed several times by varying the following parameters: (i) the discovery timeout has been set to 2, 6 and 10 s; (ii) the explanation SC, described in Section 3.2, has been either enabled or disabled.

The experimental campaign has leveraged the adoption of *Docker* platform to deploy the testbed, by performing the following steps:
– the prototype has been compiled as a Docker image, to create all the scenarios;
– each node has been executed as a container instance of the compiled image;
– a Docker *Swarm mode* cluster has been deployed on 6 *VirtualBox* virtual machines running on a workstation[7], with an overlay network configured to allow communications among the Iroha nodes;
– the execution of experiments has been managed via the Docker API SDK[8].

The following performance metrics have been calculated: (i) average request processing time, both as all-out and split by task; (ii) average hit ratio per node, *i.e.*, percentage of requests with at least one resource satisfying the constraints within the given timeout. Experimental results are reported later on.

Time. Average turnaround times can be deemed as very low in experiments involving 50 and 150 nodes, as Figure 4 and Figure 5 show. the growth exhibits a linear trend, suggesting proper scaling. With 500 nodes, instead, absolute time reach the timeouts as depicted in Figure 6, due to the needed consensus about among a larger number of entities. Furthermore, in all the experiments the time of discovery process dominates the ones of explanation and selection phases. This could be due to the fact that matchmaking is the most computationally intensive task, despite the optimization of the adopted reasoning engine.

[6] http://zlib.net/
[7] Intel Xeon E5-2643 CPU at 3.30 GHz, 48 GB of RAM and Ubuntu 16.04 (64bit) operating system.
[8] https://github.com/spotify/docker-client

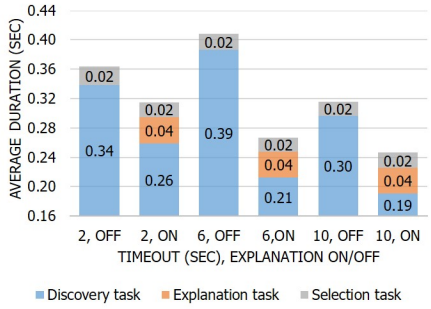

Fig. 4. Processing time for tasks on 50 nodes

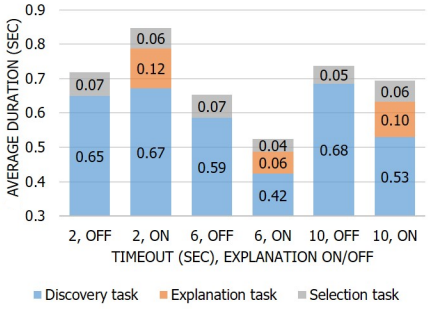

Fig. 5. Processing time for tasks on 150 nodes

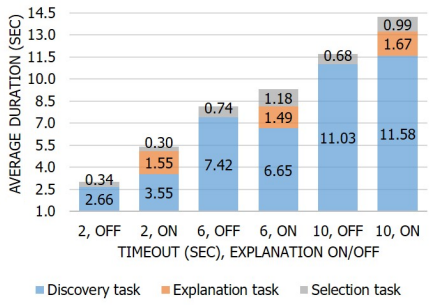

Fig. 6. Processing time for tasks on 500 nodes

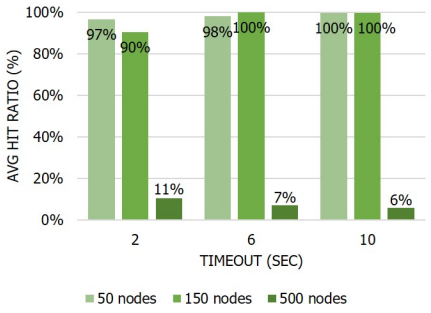

Fig. 7. Average hit ratio depending on timeout

Hit ratio. Figure 7 shows average results are closely related to the number of nodes. The best outcomes have been obtained in the small and medium scenarios with timeouts high enough. Conversely, in the 500 nodes case the average hit ratio is noticeably lower. The increased resource miss ratio is partially due to Docker Swarm deployment on a single host instead of adopting a more proper cluster computing environment: the large number of containers on the same host led to Docker resource contention issues affecting the CPU, file system and network load. Furthermore, the inherent complexity of consensus algorithms tends to increase at higher scales, leading to higher processing times and consequent increased probability of timeout expiration.

Early results supported the feasibility of the proposal, as performance is basically satisfactory for small-to-medium permissioned blockchains. Larger-scale scenarios could not be setup on the reference testbed due to the above limits with Docker Swarm deployment in a single-host environment. Testbed migration toward a computer cluster will be performed to re-evaluate semantic-enhanced blockchain performance in the same scenarios as well as to allow larger-scale simulations.

5 Applications

Semantic-enhanced blockchain systems enable discovery infrastructures for general-purpose machine-to-machine trustless marketplaces with minimal or no human intervention across multiple DAOs. This has several possible applications with potential transformation impact on relevant sectors.

– **Logistics.** Asset tracking and supply chain are among the most popular blockchain applications, due to the easy fit with existing industry standards. The simplest approaches rely on transactional ledgers for asset transfer [2]. Semantic-enhanced SC-based blockchains further allow any application logic to be implemented, and also support discoverable, composable and verifiable multi-step business processes in multi-party SOAs [8].

– **Industry 4.0**. IoT-based manufacturing benefits from blockchain technologies [2], granting not only a decentralized collaboration infrastructure, but also a ledger for process traceability of production and quality assurance. Semantic-based blockchain evolution can provide greater composition flexibility and rigorous process formalization.

– **Utility markets.** Energy, water and natural gas provisioning are increasingly relying on sensor networks, low-level digital control and high-level decision support. Semantic-enhanced blockchains can strongly support the *Smart Grid*, by providing both resource discovery and a robust ledger for contracts and payments, which are needed in large-scale peer-to-peer decentralized marketplaces.

– **Public sector.** Many public services can be made faster, cheaper and less error prone through process and data dematerialization. Blockchain technology can assist in the interfacing of the information systems of several independent branches and levels of the public administration. Furthermore, it plays the role of verifiable registry in property transfers as well as authentication and notary services. Semantic-based querying capabilities make information and functionalities more accessible to both citizens and decision-makers.

– **Financial services.** Traditional banks and finance institutions, private and public alike, are experimenting with blockchain technology to reduce operating costs of financial transactions management. Semantic-based approaches enable a marketplace of financial services, where atomic building blocks can be automatically discovered, compared and composed in order to provide the most suitable personalized solutions.

6 Conclusion

The paper proposed a framework redesigning resource discovery for SWoT scenarios thanks to an underlying blockchain infrastructure. Registration, discovery, selection and finalization tasks have been revisited as smart contracts with opportunistic and distributed execution, exploiting validation by consensus. Logic-based explanation of discovery outcomes is an important feature of the proposal, granted by non-standard inferences for request-resource matchmaking.

Future aims are basically directed to migrate the testbed toward a cluster of physical nodes, in order to remove bottlenecks biasing results and increase

simulation scale. Development of case studies within the envisioned application areas is at an early stage and will be completed to fully validate benefits and possible limitations of the proposal in realistic settings.

Acknowledgements

This work was supported in part by Italian PON project ERHA (Enhanced Radiotherapy with HAdrons).

References

1. Baader, F., Calvanese, D., Mc Guinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook. Cambridge University Press (2002)
2. Christidis, K., Devetsikiotis, M.: Blockchains and Smart Contracts for the Internet of Things. IEEE Access **4**, 2292–2303 (2016)
3. English, M., Auer, S., Domingue, J.: Block Chain Technologies & The Semantic Web: A Framework for Symbiotic Development. In: Lehmann, J., Thakkar, H., Halilaj, L., Asmat, R. (eds.) Computer Science Conference for University of Bonn Students. pp. 47–61 (2016)
4. Idelberger, F., Governatori, G., Riveret, R., Sartor, G.: Evaluation of logic-based smart contracts for blockchain systems. In: International Symposium on Rules and Rule Markup Languages for the Semantic Web. pp. 167–183. Springer (2016)
5. Jelasity, M., Montresor, A., Babaoglu, O.: Gossip-based aggregation in large dynamic networks. ACM Transactions on Computer Systems (TOCS) **23**(3), 219–252 (2005)
6. Kim, H.M., Laskowski, M.: Towards an Ontology-Driven Blockchain Design for Supply Chain Provenance (2016), https://ssrn.com/abstract=2828369
7. Korpela, K., Hallikas, J., Dahlberg, T.: Digital Supply Chain Transformation toward Blockchain Integration. In: Proceedings of the 50th Hawaii International Conference on System Sciences. pp. 4182–4191 (2017)
8. Norta, A.: Creation of smart-contracting collaborations for decentralized autonomous organizations. In: International Conference on Business Informatics Research. pp. 3–17. Springer (2015)
9. Ruta, M., Di Sciascio, E., Scioscia, F.: Concept Abduction and Contraction in Semantic-based P2P Environments. Web Intelligence and Agent Systems **9**(3), 179–207 (2011)
10. Scioscia, F., Ruta, M.: Building a Semantic Web of Things: issues and perspectives in information compression. In: Semantic Web Information Management (SWIM'09). In Proceedings of the 3rd IEEE International Conference on Semantic Computing (ICSC 2009). pp. 589–594. IEEE Computer Society (2009)
11. Scioscia, F., Ruta, M., Loseto, G., Gramegna, F., Ieva, S., Pinto, A., Di Sciascio, E.: A Mobile Matchmaker for the Ubiquitous Semantic Web. International Journal on Semantic Web and Information Systems (IJSWIS) **10**(4), 77–100 (2014)
12. Szabo, N.: Formalizing and securing relationships on public networks. First Monday **2**(9) (1997)
13. Vukolić, M.: The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In: International Workshop on Open Problems in Network Security. pp. 112–125. Springer (2015)