A Novel Blockchain Oracle Implementation Scheme Based on Application Specific Knowledge Engines

Shuai Wang[1,2,3], Hao Lu[1,4], Xingkai Sun[1,2,3], Yong Yuan*[1,2]*(Corresponding author)* and Fei-Yue Wang[1,2,3]
1. The State Key Laboratory for Management and Control of Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
2. Qingdao Academy of Intelligent Industries, Qingdao 266109, China
3. University of Chinese Academy of Sciences, Beijing 100049, China
4. School of Computer Science & Technology, Beijing Institute of Technology, Beijing 100081, China
{wangshuai2015, hao.lu, xingkai.sun, yong.yuan, feiyue.wang}@ia.ac.cn

***Abstract*—In blockchain ecosystems, an Oracle is a service tool which provides real-world data for smart contracts and other blockchain applications. At present, there are several Oracle implementation schemes, e.g. centralized Oracles, decentralized Oracles, and hardware Oracles. However, these schemes typically suffer from single source of data and low scalability. Application Specific Knowledge Engine (ASKE) is an integrated topic/application-centered knowledge portal that supports effective information retrieval and analysis. Inspired by ASKE, in this paper, we propose a novel Oracle implementation scheme. The proposed scheme can realize multi-source data extraction and analysis, then working prototypes are demonstrated to show the validity of the scheme.**

***Index Terms*—Oracle, blockchain, application specific knowledge engine (ASKE)**

I. Introduction

Blockchain is a deterministic, closed system [1], [2]. At present, it can only obtain data inside the chain, but not the real-world data outside the chain, which means that the blockchain is separated from the external world. However, the execution of blockchain smart contracts require external trigger conditions [3]. When trigger conditions rely on information outside the chain, these information need to be written into blockchain first. This process requires the Oracle to input real-world data into the blockchain. This is because the execution results of smart contracts must be consistent, so smart contracts are not allowed to actively conduct network calls (since the external data obtained by different blockchain nodes may be different, even the data obtained by the same node at different times may also be different), otherwise the execution results will be uncertain.

The real-world data or event that is sent to the blockchain as a transaction via Oracle will serve as a deterministic input that triggers the logical judgment of smart contracts. The Oracle can provide a variety of data, such as the arrival time of flights for the flight delay insurance smart contracts, the random number for the gaming smart contracts, and the asset price in the financial derivative smart contracts. Therefore, Oracle is the only interface for data interaction between blockchain and the external world, and thus plays an important role in the construction of blockchain ecosystems.

The Oracle's workflow is: when smart contracts have data requirements for the external world, the requests are first sent to Oracle (Oracle usually also appears as a smart contract), Oracle will obtain the real-world data through certain methods, then the data is returned to smart contracts within the blockchain. In this process, the most important thing is to ensure that Oracle gets the right data and the data has not been tampered with.

Currently, there are three types of Oracles. (1) Centralized Oracles. Such Oracles specify the data source by the users, and then the Oracles go to the corresponding data source to extract information (e.g. Provable[1]). Oracles do not interfere with the choice of data source and the accuracy of data source itself. However, centralized Oracles are usually operated by a single organization, so there are risks of centralization and Single Point of Failure (SPoF) [4]. (2) Decentralized Oracles. Such Oracles are not controlled by a single institution and there is no risk of SPoF. According to whether human participation is required, the decentralized Oracles can be divided into prediction markets Oracles (e.g. Augur[2]) and Layer-2 Oracles (e.g. Chainlink[3]). The former obtains data through group intelligence [5] and voting mechanism [6]; the latter collects data through a group of off-chain nodes, and the collected data is aggregated to get a final data. (3) Hardware Oracles. Such Oracles are usually data collectors on the Internet of Things, such as sensors in traceability systems and medical devices that collect various medical data, etc. The widespread use of blockchain technology in the Internet of Things field will spawn a large number of hardware Oracles in the future.

However, the current mainstream Oracle implementation schemes have the following problems: Firstly, the data source is single. Users need to specify a single data source (such as a webpage), but if the data source itself is false or maliciously

This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101400, and the National Natural Science Foundation of China under Grant 61533019 and Grant 71702182, and also in part by the Open Project Program of State Key Laboratory of Software Development Environment under Grant SKLSDE-2018KF-06.

[1] Provable. https://docs.provable.xyz/

[2] Augur. https://www.augur.net/

[3] Chainlink. https://chain.link/

978-1-7281-2853-5/19/$31.00©2019 IEEE

tampered, the data returned by Oracle is also wrong. Secondly, the intelligence is not high. Data can not be collected, cleaned, and analyzed intelligently. Lastly, the scalability is not good, and the performance will drop sharply as the number of participating nodes increase. With the popularity of blockchain applications, a large number of data query services are required in the future, so it is necessary to design a new Oracle implementation scheme that is automated, fast and accurate.

In this paper, we propose a novel new Oracle implementation scheme based on Application Specific Knowledge Engines (ASKE) [7]. ASKE is an information acquisition and analysis framework that can effectively collect and unify open source information in specific domains, and integrates multiple data analysis methods to analyze the collected data in multiple dimensions. With ASKE, users do not have to specify a single data source, and the web crawlers will automatically crawl the relevant authoritative websites to collect the required data. The data is aggregated off-chain to obtain the final result, then final result is returned to blockchain smart contracts. This scheme paves a new way for multi-data source Oracle implementation.

The remainder of this paper is organized as follows: Section II introduces the concept and system architecture of the application specific knowledge engines (ASKE). Based on ASKE, Section III proposes a novel Oracle implementation scheme, and two working demos are presented. The limitations and future works are discussed in Section IV. Section V concludes the paper.

II. An Introduction to Application Specific Knowledge Engines (ASKE)

Application Specific Knowledge Engine (ASKE) is an integrated topic/application-centered knowledge portal, which supports effective information retrieval and analysis. The basic idea of ASKE is to create a series of web crawlers [8] (also called spider agents, which is a program that automatically fetches web documents), that can collect specific information from heterogeneous data sources over the Internet and establish a semantic data repository, then uses a Knowledge Configuration File (KCF) to specify topics, keywords, searching sequences and schedules for query processing. The characteristics of ASKE are user specific, application specific, and domain specific [9]. As shown in Fig. 1, ASKE consists of five modules: data acquisition module, data repository, domain-specific ontology, data analysis module, and data visualization module [10]. The data acquisition module can effectively acquires various types of open source information through techniques such as web crawling [11], deep network collection, dynamic network collection, and data filtering. The data repository consists of two layers, i.e. the bottom layer and the upper layer. The bottom layer stores the raw data obtained by data acquisition module, and the upper layer stores more specific domain-related information extracted through domain ontology and data processing. Domain-specific ontology usually include classes (concepts), individuals (instances), attributes, inheritance, and relationships between classes and individuals [12]. The data analysis module performs in-depth analysis and mining of the collected open source data. The data visualization module uses visualization tools to present the analysis results.

The development of ASKE mainly consists of the following two phases:

(1) Data repositories construction. This phase aims to build data collection that is comprehensive and relevant. It includes three sub-phases, namely, data collection, data preparation and data silo. Data collection uses a module called Resource Identifier to locate domain relevant data resources (e.g. the sites that are known to provide specific contents, web directories that are dedicate to gathering related sites and web pages in certain applications), and web crawlers are utilized to crawl data. In data preparation sub-phase, data classifier is used to categorize all the collected data. And a parser and an indexer are applied to build up indices, lexicon library, and searchable databases for data collection. Lastly, semantic data repositories are constructed based on texture documents and structured databases via ontology developer and metadata extractor [13].

(2) Searching by KCF. After semantic data collection is completed, searches are conducted by KCF to help users to accurately and easily locate required information.

In ASKE, how to implement the Resource Identifier and web crawlers is very important. Ref. [9] proposed a recursive collection-building procedure, which combined both manual selection and automatic web-crawling methods. According to [9], a limited number of seed URLs (authority sites) are first identified through careful and systematic manual selections. After the high-quality resources have been identified, the data collection is automatically done by web crawlers. It is worth noting that to make sure the fetched pages are relevant to certain domains, the crawlers are limited within particular hyperlinked resources from the Resource Identifier. To fetch more pages in shorter time, crawlers can work in parallel.

III. Working Prototype

A. Workflow of the Proposed Oracle Implementation Scheme

For specific tasks in certain domains (such as sports competitions, weather prediction, political elections, etc), we ask domain experts to specify some authoritative websites (such as government websites, official websites or portal websites). For data query requests from blockchain smart contracts, the proposed Oracle implementation scheme first determines which domain the request belongs to, then ASKE is utilized to extract information from authoritative websites (for the designated websites, ASKE can achieve accurate information extraction through customized wrapper), the results are analyzed and aggregated to get a final result. Finally, the result is returned to the blockchain, as shown in Fig. 2.

B. Working Demo

Since the proposed scheme is still a preliminary study, we make a simple demonstration using football match results query and weather query as examples. First, some authoritative websites are selected in advance (for football match results query, we choose CCTV.com, Sina Sports, Tencent Sports,

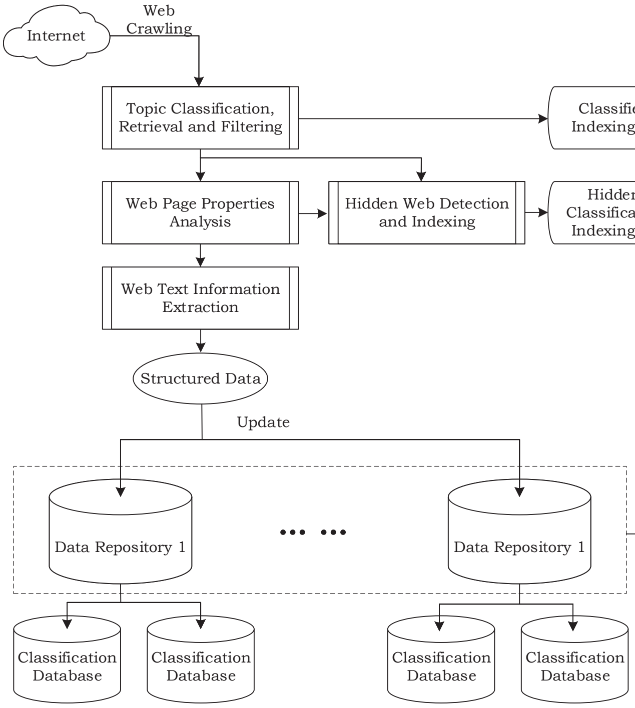

Internet
Web Crawling
Topic Classification, Retrieval and Filtering
Classified Web Indexing Library
Web Page Properties Analysis
Hidden Web Detection and Indexing
Hidden Web Classification and Indexing Library
Web Text Information Extraction
Structured Data
Update
Data Repository 1
Data Repository 1
Classification Database
Classification Database
Classification Database
Classification Database
User Query Input
Query Understanding
Query Expressions
Domain-specific Ontology
Data Analysis and Retrieval
Query Results/ Data Visualization

Fig. 1. Application Specific Knowledge Engines (ASKE).

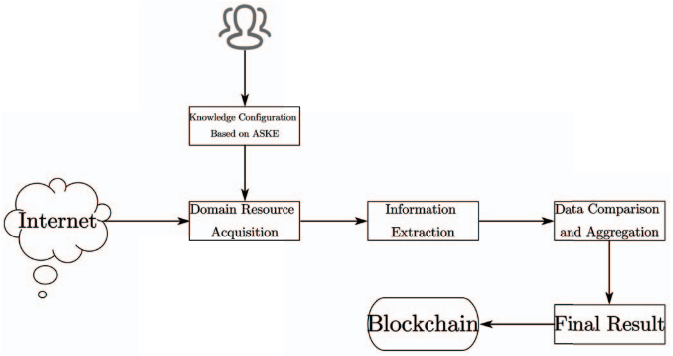

Knowledge Configuration Based on ASKE
Internet
Domain Resource Acquisition
Information Extraction
Data Comparison and Aggregation
Blockchain
Final Result

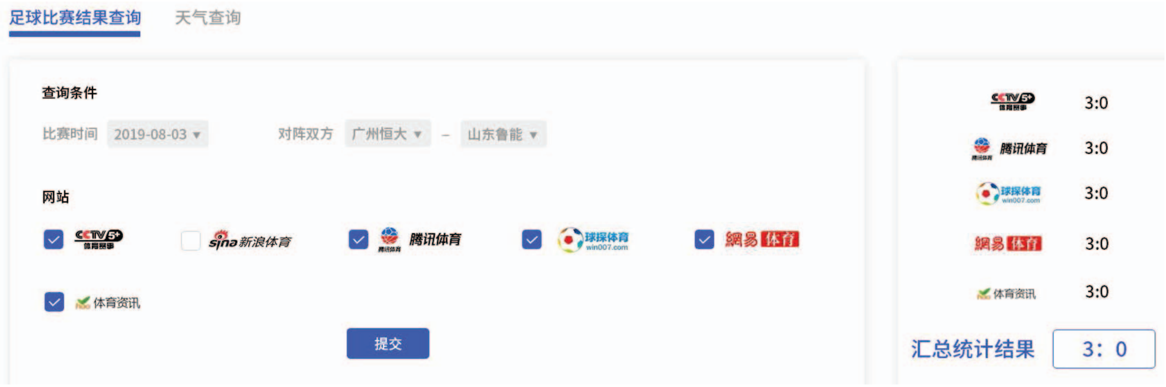
Fig. 2. Workflow of the proposed Oracle implementation scheme.

win007.com, NetEase Sports, etc. as authoritative websites; for weather query, we choose weather.com, tianqi.2345.com, Tencent Weather, Sohu weather, etc. as data sources). After user specifies the query conditions, the web crawlers in ASKE automatically collect the data on the websites. The returned data are compared, analyzed and aggregated to get a final result. The working demos are shown in Fig. 3 and Fig. 4.

IV. The Future Works

On the basis of the above preliminary work, we plan to do the following works in the future:

- Accurate information extraction [14] from heterogeneous data sources. Webpages are sometimes semi-structured and mixed with free text. For the authoritative websites in ASKE, the customized wrappers can achieve good information extraction performance. However, in the future, the Oracle will include more heterogeneous data sources for intelligent data collection, so it is impossible to build wrappers for all data sources manually. Thus, we need to study the processing mechanisms for semi-structured and unstructured webpages. For semi-structured webpages, the wrapper is constructed by automatic learning and the information extractor is trained by statistical learning. For unstructured webpages, the maximum entropy HMM information extraction method can be used to train the extractor. At the same time, the modification and updating of the websites also make the manual way unable to respond in time. In order to effectively track the changes of websites and improve extraction efficiency, automatic learning is used to construct the wrapper of authoritative websites. Through collecting typical webpages as samples for tagging, and by inductive learning [15] to obtain the extraction rules, the automatic customization of wrappers can be realized. Besides, by comparing the changes of webpages structure before and after, the parameters of wrappers are automatically adjusted to track the changes of the webpages. On this basis, efficient information extraction from heterogeneous data sources can be realized.
- Authenticity Proofs. An important requirement for Oracle is to ensure that the data it returns to the blockchain has not been tampered with. So Oracle needs to provide the authenticity proofs for the returned data. Our current Oracle implementation scheme does not provide this kind of proof. Next, we may refer to the TLSNotary Proof,



查询条件

比赛时间 2019-08-03 ▾　　对阵双方 广州恒大 ▾ – 山东鲁能 ▾

网站

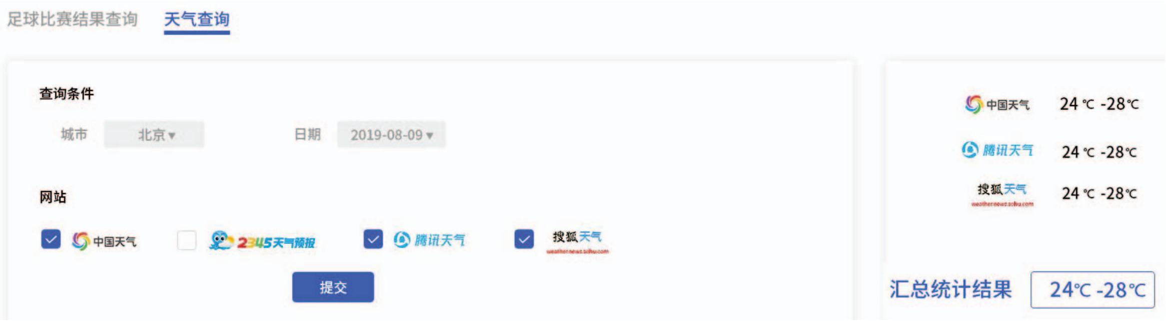
☑ CCTV5+ 体育赛事　□ sina 新浪体育　☑ 腾讯体育　☑ 球探体育 win007.com　☑ 网易体育

☑ 体育资讯

CCTV5+ 3:0
腾讯体育 3:0
球探体育 3:0
网易体育 3:0
体育资讯 3:0

汇总统计结果 3：0

Fig. 3. Football match results query.



查询条件

城市 北京 ▾　　日期 2019-08-09 ▾

网站

☑ 中国天气　□ 2345天气预报　☑ 腾讯天气　☑ 搜狐天气

中国天气 24 ℃ -28℃
腾讯天气 24 ℃ -28℃
搜狐天气 24 ℃ -28℃

汇总统计结果 24℃ -28℃

Fig. 4. Weather query.

Android Proof, or Ledger Proof that is used by Provable.

- Although the proposed scheme implements the multi-data source information retrieval, the system itself is still centralized. In the future, we consider building a Decentralized Autonomous Organization (DAO) [16] to allow more nodes to participate in the Oracle service ecosystems. Besides, we can use the reputation mechanism [17] or Token incentive mechanism [18] to promote the healthy operation of DAO.

V. Conclusion

In this paper, we propose a novel Oracle implementation scheme based on application specific knowledge engines (ASKE). ASKE is a topic/application-centered information retrieval and analysis system with characteristics of user specific, application specific, and domain specific. The proposed scheme can extract multi-source data from authoritative websites and working demos show the effectiveness of the system. Towards the end, limitations and future works are discussed, and many aspects need to be refined in the future.

References

[1] Y. Yuan, and F. Y. Wang, “Blockchain and cryptocurrencies: model, techniques, and applications,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 9, pp. 1421-1428, Jul. 2018.

[2] F. Wang, Y. Yuan, C. Rong *et al.*, “Parallel blockchain: an architecture for CPSS-based smart societies,” IEEE Transactions on Computational Social Systems, vol. 5, no. 2, pp. 303-310, Jun. 2018.

[3] S. Wang, L. Ouyang, Y. Yuan *et al.*, “Blockchain-enabled smart contracts: architecture, applications, and future trends,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, Early Access, doi: 10.1109/TSMC.2019.2895123, Feb. 2019.

[4] Z. Xin and C. Shouping, “Study on insulation detection method of electric vehicles based on single point of failure model,” in *2016 11th International Forum on Strategic Technology (IFOST)*, Novosibirsk, 2016, pp. 191-194.

[5] C. Yang, O. Flak and M. Grzegorzek., “Representation and matching of team managers: an experimental research,” IEEE Transactions on Computational Social Systems, vol. 5, no. 2, pp. 311-323, June 2018.

[6] X. Yang, C. Liang, M. Zhao *et al.*, “Collaborative filtering-based recommendation of online social voting,” IEEE Transactions on Computational Social Systems, vol. 4, no. 1, pp. 1-13, March 2017.

[7] F.-Y. Wang, G. Lai, and S. Tang, “An application specific knowledge engine for researches in intelligent transportation systems,” in *The 7th International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, 2004, pp. 841-846.

[8] G. H. Agre and N. V. Mahajan, ”Keyword focused web crawler,” in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, Coimbatore, 2015, pp. 1089-1092.

[9] G. Lai, Q. Zhang, D. Wen *et al.*, “A prototype of the next-generation journal system for ITS: academic social networking and media based on web 3.0,” IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 3, pp. 1078-1087, Sept. 2012.

[10] H. Lu, X. Sun, W. Liu *et al.*, “ASKE-based open source intelligence collection and analytics – an example for key chinese intelligence journal during 2008-2012 ,” Journal of Command and Control, vol. 1, no. 3, pp. 254-262, 2015.

[11] K. Lin, Y. Chang, C. Shen *et al.*, “Leveraging online word of mouth for personalized app recommendation,” IEEE Transactions on Computational Social Systems, vol. 5, no. 4, pp. 1061-1070, Dec. 2018.

[12] D. L. McGuinness and F. Van Harmelen, “OWL web ontology language overview,” W3C recommendation, 2004, 10(10): 2004.

[13] A. Preece, I. Spasic, K. Evans *et al.*, “Sentinel: a codesigned platform

for semantic enrichment of social media streams," IEEE Transactions on Computational Social Systems, vol. 5, no. 1, pp. 118-131, Mar. 2018.

[14] W. Li, D. Cheng, L. He *et al.*, "Joint event extraction based on hierarchical event schemas from FrameNet," IEEE Access, vol. 7, pp. 25001-25015, 2019.

[15] Z. Wang, K. Araki and K. Tochinai, "Word segmentation method based on inductive learning and segmentation rule," in *2008 International Symposium on Computational Intelligence and Design*, Wuhan, 2008, pp. 95-98.

[16] "What is a DAO." Accessed: Aug. 2, 2019. [Online]. Available: https://blockchainhub.net/dao-decentralized-autonomous-organization/

[17] M. E. Marrakchi, H. Bensaid, M. Bellafkih and H. Bensaid, "Intelligent reputation scoring in social networks: use case of brands of smartphones," in *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, Mohammedia, 2016, pp. 1-6.

[18] F. Wang, J. Zhang, R. Qin *et al.*, "Social energy: emerging token economy for energy production and consumption," IEEE Transactions on Computational Social Systems, vol. 6, no. 3, pp. 388-393, Jun. 2019.