Research on Social Networks Based on Blockchain

Yingyu Huo [1,*], Yong Zhong [1,2]
[1]*Electronic and Information Engineering School, Foshan University, Foshan, China*
[2]*Foshan Data Science Society, Foshan, China*
**Corresponding author.*

Abstract

The centralized structure of current social networks makes users lose controls of their private contents so that their personal privacy and digital rights can not be protected. BlockSN, a social network based on blockchain which uses the decentralization and immutability of blockchain to realize security and self-control of private data, is presented in this paper. BlockSN can share and propagate digital contents by internal incentive mechanism of blockchain, and realize united programming language by the powerful and expressive logic language Active-U-Datalog. The research can help the protection of user privacy and intellectual property, and help the propagation and share of digital content in social networks, which can develop commercial applications in social networks that bring excellent application values. The model structure, formal definition and operation mechanism of the network is discussed, and implementation and application of BlockSN are analysed. At last, an application example shows the feasibility of such social network based on blockchain.

Keywords: *Blockchain, social network, decentralization, logic language*

I. Introduction

In recent years, social networks, such as Facebook, twitter, Renren, microblog and so on, has become one of the most popular internet applications and owned lots of users. However, most of the existing social networks are centralized structures, and its social network management organizations are responsible for making rules, storing, managing and distributing content centrally. Users can not manage and control their own content which they create to get corresponding rewards, and their privacy and intellectual property rights are difficult to be protected. Blockchain system [1], due to its security features of decentralization and tamper-resistant, is paid attention to in the research of social network to solve the problem of centralized social platform, such as Obsidian Messenger, Nexus, Indorse, Synergeo, Steemit and domestic ONO based on social network or social media.

The research on social application based on blockchain has also received extensive attention in the academic circles. Fu et al. [2] applied blockchain-based trusted computing in social networks, and achieve decentralized privacy protection of social networks by proof of credibility. Chakravorty et al. [3] proposed a user-centered blockchain to control, track and confirm the rights of digital content in social media networks. Chen et al. [4] proposed a model of using blockchain to limit the spread of large-scale rumors in social networks, and Zhang et al. [5] proposed a model and protocol of the protection of private health information by blockchain in mobile social networks. Arquam et al. [6] studied the establishment of a secure and trusted framework based on blockchain and the realization of information sharing securely in online social networks. Most of the current researches focus on how to achieve information sharing securely, privacy protection, trusted computing and other fields by blockchain in social networks, and lack of research on social network model and mechanism based on blockchain architecture.

To solve the above problems, this paper proposes a blockchain social network model BlockSN, which uses a strong expressive logic language to implement the contract mechanism. The model structure, formal definition and operation mechanism of BlockSN are discussed, and the implementation and application of BlockSN are analysed and demonstrated.

II. Social Network Model based on Blockchain (BlockSN)

2.1 Component of BlockSN

The blockchain is regarded as a distributed database, in which the records are organized in the form of blocks. Distributed consensus is a rule accepted by blockchain nodes maintance to manage the construction and operation of blockchain [7].

BlockSN is a social network based on the blockchain, which can realize the security and self-control of personal content by the decentralization and non tamperability of the blockchain. It can also achieve the sharing and dissemination of content through the internal incentive mechanism of the blockchain. The Model of BlockSN is shown in Figure 1.

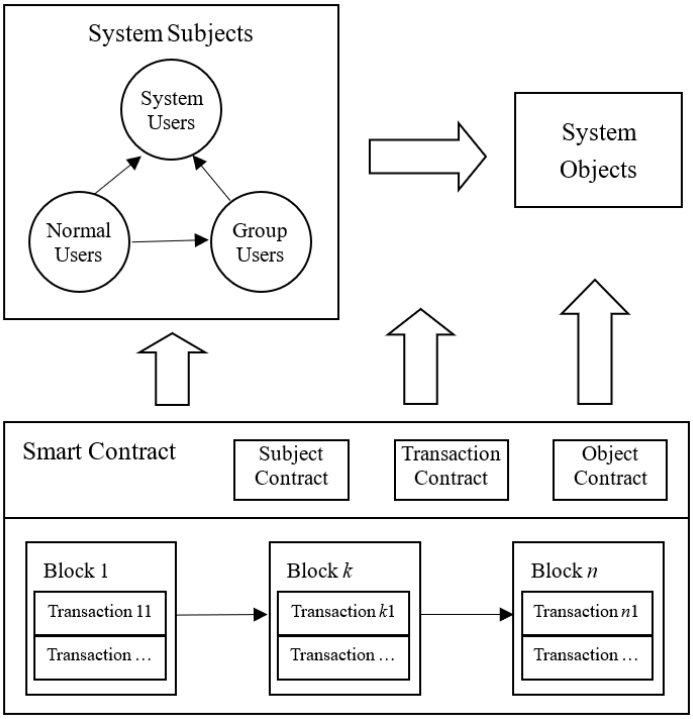

Fig. 1: Logic model of BlockSN

2.1.1 System subject

The system subject includes ordinary users, group users and system users. Group users are composed of ordinary users. Each system subject will generate a corresponding smart contract which is deployed on the blockchain when it is created. Such smart contract is called subject contract, and it is unique for each subject. Once the subject contract is created, it can not be withdrawn or tampered with, which effectively protects the subject's personal data and privacy.

2.1.2 System object

The system object can be all kinds of independent entity content of social network, such as chat content, articles, comments, notes and sorts of pictures, videos, audio, etc. The object belongs to a single or multiple subjects, and its subordination is an important part of the subject contract.

System subject and system object are system entities.

2.1.3 Operation

The operation is the behavior of the system subject to the system entity, such as trading tokens, changing the subject's attributes, establishing relationships, publishing articles, writing comments, etc. This model realizes the operation of the subject to the entity through the behavior of the subject to the entity's attributes.

2.1.4 Smart contract

The smart contract includes the subject contract that is generated from the system subject, the object contract that controls the use of the system object, and the transaction contract that carries on the dialogue and transaction between the subjects.

2.1.5 Transaction and block

The transactions are formed by the operations of the system user to the system entity, which are stored in the blocks with an ordered chain by the form of token transaction.

2.1.6 Attribute set

The attribute set includes all attributes, the characteristics of entities, smart contracts, blocks and blockchains, of a single entity.

2.2 Formal Definition

For the system subject, users can join a user group to become one of the members of the group. The blockchain social network is regarded as a system group and all users and user groups are members of such system group, and there is no inclusion relationship between user groups. Therefore, the system forms a hierarchical relationship. In the model of RuleSN [8], the subject hierarchy is defined as follows.

Definition 1. (Subject Hierarchy) Subject hierarchy SH is a quadruple (S_{user}, S_{sys}, $S_{usergroup}$, $\leq_{SH}$) where

(1) S_{user} is the user identifier set, S_{sys} is the system group identifier set, $S_{usergroup}$ is the user group identifier set, $\leq_{SH}$ is partial order relation.

(2) $\forall u \in S_{user}$, $\forall g \in S_{usergroup}$, $u \leq_{SH} g$ if and only if u is the member of g.

(3) $\forall v \in S_{user} \cup S_{usergroup}$, $\forall sys \in S_{sys}$, $v \leq_{SH} sys$.

Definition 2. (Subject Relation) Subject relation UR is a triple (S, E, $\sum$) where

(1) $S = S_{user} \cup S_{usergroup} \cup S_{sys}$ is the identifier set of all system subjects, O is the system objects identifier set, $E = S \cup O$ is the system subjects identifier set.

(2) $\sum = \sum_{s\_s} \cup \sum_{s\_o} = \{t_1, t_2, t_3, \ldots, t_n\}$ represents the set of relation types, where $t_i \in \sum (1 \leq i \leq n)$ is the relation type. The set of relation types is divided into tow categories S_S and S_O, where S_S defined as $S \times \sum_{s\_s} \times S$ is the relationship type between system subjects and S_O defined as $S \times \sum_{s\_o} \times O$ is the relationship type between the system subject and the system object.

Definition 3. (Attribute Set) Attribute set of the system entity E expressed as Δ_E is an attribute set of current value by a binary (EN, EV) where EN and EV represent the predefined attribute name set and value range set of E respectively.

The current value of a single attribute is also expressed by functions, such as $name$ (u) = "Wang Lin" and age (u) =25.

Definition 4. (Transaction) Transaction TR is a quadruple (S, E, P, Δ), where

(1) S is the system subject identifiers set, and E is the system entity identifier table.

(2) P is the operation set of the system subject to the system entity.

(3) Δ is the attribute set of the entity.

If user s_1 transfers 10 tokens to s_2, the following transaction that the value of the attribute *Account* of s_1 will subtract 10 while which of s_2 will corresponding increase 10 will be generated:

(s_1, s_2, *transfer*, (*Account*, Vs_1-10), (*Account*, Vs_2+10))

When finished, the transaction will be saved in the block orderly.

Definition 5. (Contract Policy) Contract policy CP is a quadruple (UR, Δ, TR_1, TR_2) where

(1) UR is the subject relation set, Δ is the attribute set, and TR_1 and TR_2 are the transaction sets.

(2) For any contract strategy (ur_1, Δ_1, tr_1, tr_2), ur_1, Δ_1, $tr_1 \vdash tr_2$, that means the conditions of ur_1, Δ_1 are satified and the transaction tr_1 happens will lead to the occurrence of the transaction tr_2.

$$(s_1, s_2, "friend"), Acount(s_1) > 10, (s_1, s_2, agree, (Account, 10)) \vdash (s_1, s_2, transfer, (Account, V_{s_1} - 10), (Account, V_{s_2} + 10)) \quad (1)$$

The above contract strategy indicates that if s_1 and s_2 are friends and the attribute value of s_1's account exceeds 10, the trasaction that s_1 agrees to transfer 10 tokens to s_2 and the transaction of transfer will occur meanwhile.

Definition 6. (Smart Contract) Smart contract SC is a binary (CP, Δ), where CP is the contract policy set and Δ is the entity attribute set.

Definition 7. (Block) Block BL is a finite sequence <tr_1, ..., tr_n>, where tr_i ($1 \le i \le n$) is a single transaction, and $n \le \delta$ where δ is the maximum transaction capacity of the block.

Definition 8. (Blockchain) Blockchain BC is a finite sequence <bl_1, ..., bl_n> where

(1) bl_i ($1 \le i \le n$) is a single block.

(2) $\forall bl_i$ ($1 \le i \le n$), *ord* (bl_i) = i, and *ord* is the sequence attribute of a block. That is, the sequence attribute value of the block should be the same as the position of the block in the blockchain sequence.

(3) $\forall$ bl_i ($2 \le i \le n$), *hash*(bl_{i-1}) = *link* (bl_{i-1}), where *hash* is the hash function and *link* is the link attribute of the block, that is, the blockchain is joined by the hash value of the previous block.

Definition 9. (Blockchain Social Network System) Blockchain social network system *BlockSN* is a six tuple system (*UGH*, *O*, *UR*, *TR*, Δ, *SC*, *BC*) where

(1) $UGH = (U, G_{sys}, G_{user}, \le_{UG})$ is the subject hierarchy.

(2) O is the set of objects.

(3) *UR* are the relationships sets between users and between subjects and objects.

(4) *TR* is the set of transactions.

(5) Δ is the attribute set of entity, transaction, block and blockchain.

(6) *SC* is the set of smart contracts.

(7) *BC* is a blockchain.

2.3 Operation mechanism

The operation mechanism of the model is illustrated by the process of user *A* using BlockSN.

(1) When logging in BlockSN for the first time, user *A* needs to register a new user by following the policy of the new user registration contract in the system subject contract.

(2) After the new user registering successfully, the system subject contract generates the unique subject contract of user *A* which includes the user's public key and related attributes, contract policy. Such contract is deployed on the blockchain and will not be changed again. The user who wants to change the subject contract must re-register.

(3) The trade of the new user's registration which are saved in the block in the form of transactions can't be changed again.

(4) User *A* can independently make and use the sharing rules and the benefit sharing mechanism for the objects generated by itself such as chat records, articles, comments, notes and various kinds of pictures, videos, audio and generate the object contract which can also be solidified on the blockchain. Files with large amount of data can generate hash codes which are stored on the blockchain. The encryption key for the objects that need encryption can also be managed by the subject contract to ensure the security.

(5) For the transaction of the objects in BlockSN, user *A* can also generate a contract for transaction.

2.4 Incentive mechanism

The incentives of BlockSN include:

(1) The contribution of computing power. If users contribute their computing power to the platform, they will be rewarded by the blockchain mining incentive mechanism and get tokens.

(2) System management. It is the management incentive of system users to users and platform. Because BlockSN can no longer get profits through information monopoly, selling users' attention, inducing users' behavior and fixed-point advertising push, just like traditional social networks, system users will get a certain token as revenue for the management of normal users and platforms according to the rules.

(3) Content sharing. Users can make revenue rules of digital rights and obtain token through copyright revenue by sharing and using their own digital content.

(4) Social activity. Users can get some tokens by participating in social activities on the behaviour which can improve system activity and attention such as posting, popularity ranking, etc.

2.5 Unified programming language

Currently, blockchain programming mostly adopts imperative language, such as Solidity or Go language used by

Ethereum. Against the shortcomings of imperative language, Idelberger et al. [9] proposed to adopt declarative programming language, such as smart contract language based on logic, and we propose to use logic language Active-U-Datalog[10] as the programming language of smart contract and propose smart contract model Logic-SC[11]. Active-U-Datalog is a update Datalog program with active rules, whose predicate atoms include update atoms $\pm p(t_1, t_2,\ldots t_n)$ representing insertion and deletion, which can be regarded as a special instance of constraint logic programming (CLP) syntactically. Active-U-Datalog program $P=IDB\cup EDB\cup AR$, in which the extensive database EDB is composed of a set of ground atoms representing the state of the database, and the intensive database IDB defined by the following formal rules:

$$h \leftarrow u_1,\cdots,u_k,l_1,\cdots,l_m \tag{2}$$

Where h is arbitrary atom, l_i $(0\le i\le m)$ is arbitrary literal, and $u_i(0\le i\le k)$ is arbitrary update atom.
Active rule set AR is defined by the following rules:

$$u_1,\cdots,u_k \leftarrow e_1,\cdots,e_h,l_1,\cdots,l_m \tag{3}$$

Where $u_i(0\le i\le k)$ and $e_i(0\le i\le h)$ are arbitrary update atoms, and $l_i(0\le i\le m)$ is arbitrary literal.

In the smart contract model Logic-SC, the contract SC is composed of *CONST*, *EDB*, *IDB*, *TR* where

(1) *CONST* is a constant set.

(2) *EDB* is an extensive database.

(3) IDB is an intensive database.

(4) AR is an active rule set.

In smart contract, the constant set and the extensive database constitute the attribute set of entities, and the intensive database and the active rule set constitute the contract policy set.

In BlockSN, subject contract uses Logic-SC model as subject contract model and Active-U-Datalog logic language as unified authorization evaluation, as shown in Fig. 2.

```
1. Contract (
2. userA,
3. {PublicKey≡''81AEF78……B976'; date≡'03/08/19'}, //CONST
4. {name('userA'), token(100), object('textA', hashidA), object('videoB', hashidB), price('textA', 5), price('videoB', 10) }, //EDB
5. { watch(UserB, Video)←price(Video, P), transfer(UserB, P), +onlineuser(UserB);
6.   copy(UserB, Text) ←price(Text, P), transfer(UserB, P), CALL(UserB, {+}), +onlineuser(UserB)
7.   transfer(UserB, P)←addToken(P), CALL(UserB, {+object(Text, Hashid), +price(Text, 5),SAVE(Text)})
8.   addToken(P)←token(N), M=N+P, -token(N), + token(M)  }, //IDB
9. {+visitedtimes(Y), -visitedtimes(X)←+onlineuser(User), visitedtimes(X), Y=X+1; }, //AR
10.      )
```

Fig. 2: Sample of User A's Subject Contract

The subject contract is composed of attribute set, that is, extensive database (EDB). Line (3) is the attribute set including the public key, creation date, subject name, token number of the subject, and the two objects of text *textA* and video *videoB* and their hash codes owned by the subject, and the tokens needed to use the two objects. The rule

of line (5) indicates that a user needs to pay the token for watching the video of a subject. The predicate *transfer* (*UserB, P*) initiates the transaction of transferring *P* tokens from *UserB* to the subject and returns true only after the transaction is successful and the rule also generates an operation *onlineuser* (*UserB*) to insert the current user. The rule of line (6) is the token transfer rule, which is executed to add *P* tokens to *UserA* and subtract *P* tokens from *UserB*. The system predicate *CALL* (*UserB*, {*decToken* (*P*)}) submits the transaction {*decToken*(*P*)} to the subject contract logic program of *UserB* for execution. The rule of line (7) states that a user needs to pay the token for copying text *textA* to his space. The system submits the transaction {+*object* (*Text, Hashid*), +*price* (*Text*, 5), *SAVE* (*Text*)} to the subject contract logic program of *UserB* for execution, inserts the object ID and its price and saves the text of them in the local space by the system predicate *SAVE* (*Text*). The rule of line (8) is adding token to the subject. The rule of line (9) is an active rule, which is triggered by inserting current user to update the subject's visited times predicate *visitedtimes*.

III. Realization

Combining the centralized management mechanism of traditional social networks and the decentralized mechanism of blockchain, BlockSN adopts the mechanism of the centralized user management and the decentralized personal data management, which realize the separation of user management and data management. It not only retains the needs of user audit, personal real identity and responsibility in social networks, but also meets the needs of users to protect personal data, secure independence and benefits share.

Data storage includes two methods, storing the subject contract and the user's key data directly on the blockchain to make them tamper-resistant, and storing the files with large amount of user data in the distributed file system that the hash values of them and the keys of the files needed to be encrypted are saved in the blockchain by the subject contract to prevent tampering and illegal use. The whole implementation structure of BlockSN is shown in Fig. 3.

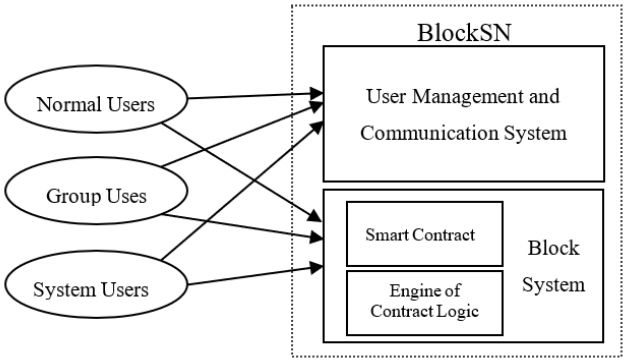

Fig. 3: The Whole Implementation Structure of BlockSN

Ethereum[12] is a popular blockchain system in recent years. Its Turing complete smart contract increases the programmability of Ethereum. The design of its account mode can reduce the storage space, and its built-in storage with persistent state facilitates the storage of user attributes and data. Therefore, BlockSN uses Ethereum as the underlying blockchain system which user accounts integrate with Ethereum's user accounts.

The subject contract in BlockSN uses the logic language Active-U-Datalog with evaluation algorithm and rule engine as the programming language, and the rule engine of it can be deployed in Ethereum.

The storage of files with large-volumn data and digital content adopts the interstellar file system (IPFS) which is a content-addressable and peer-to-peer hypermedia distribution protocol to create a network transmission protocol

that can store and share files persistently and distributedly. The characteristics of IPFS such as permanence, decentralized storage and files share are suitable for the storage of large amount of data in BlockSN. Hence capacity files in BlockSN is stored by IPFS. The implementation of BlockSN using Ethereum and IPFS will be introduced in another article.

IV. Realization

How BlockSN works will be illustrated by the process of the user *UserB* using the object *UserA*. The subject contract of *UserA* is shown in Fig. 2. After *UserB* being registered its subject contract is generated as shown in Fig.4.

```
1. Contract (
2. userB,
3. {PublicKey≡''FFFF35……B267'; date≡'03/08/19'}, //CONST
4. {name('userb'), token(100) }, //EDB
5. { addToken(P)←token(N), M=N+P, -token(N), +token(M)
   decToken(P)←token(N), N>=P, M=N-P, -token(N),
    +token(M) }, //IDB
6. { }, //AR
7. )
```

Fig. 4: Sample of User B's Initial Subject Contract

In Fig.4, line (3) is a constant set which includes the constants such as the public key and creation date of the subject. Line (4) is the extensive database which is consist of subject name and token number. The rules in line (5) and line (6) add and subtract a certain amount of tokens from the account of *UserB* respectively.

(1) If *UserB* wants to watch *UserA*'s video, he will submits the transaction {*watch* (*UserB, 'videoB'*)} to the logical program of *UserA*'s subject contract for execution. And then the subject contract of *UserA* deduces 10 tokens for the fee from the subject contract of *UserB* and allows *UserB* to watch the video.

(2) If *UserB* wants to copy *UserA*'s text *textA*, he will submits the transaction {+*object* (*'textA'*, *hashidA*), +*price* (*'textA'*, 5), *SAVE* (*'textA'*)} to the logical program of *UserA*'s subject contract for execution. Then the subject contract of *UserA* deduces 5 tokens for the fee from the subject contract of *UserB*, inserts *textA*'s object ID, hash value, and price into *UserB*'s subject contract, and saves *textA* in IPFS.

When the above steps have been executed, *UserB*'s subject contract will be updated as shown in Fig.5.

```
1. Contract (
2. userB,
3. { PublicKey≡''FFFF35……B267'; date≡'03/08/19'},    //CONST
4. { name('userb'), token(85), object('textA', hashidA), price('textA', 5) }, //EDB
5. { addToken(P)←token(N), M=N+P, -token(N), +token(M)
6.   decToken(P)←token(N), N>=P, M=N-P, -token(N), +token(M) }, //IDB
7. { }, //AR
8. )
```

Fig. 5: Sample of User B's Refresh Subject Contract

V. Conclusion

Aiming at the problems of traditional social networks such as excessive centralization, difficult protection for user privacy and data content, difficult share of benefits and so on, this paper proposes a social network based on blockchain BlockSN, explains its subject management, object management and incentive mechanism, discusses the mechanism of the subject contract based on the model Logic-SC, states the implementation, and gives an example for application. BlockSN attains the security and self-control of personal content through the decentralization of structure and the non tamperability of content, and effectively promotes the sharing and dissemination of digital content by the internal incentive mechanism of blockchain.

In future we will make deeper study on the methods and crucial technologies to develop BlockSN using Ethereum and IPFS.

Acknowledgement

This work was supported by grants from Guangdong Educational Science Planning Project under Grant No. 2018GXJK200.

References

[1] Melanie Swan, "Blockchain: blueprint for a new economy," O'Reilly Media, Inc. 2015.
[2] Fu Dongqi, Fang Liri, "Blockchain-based trusted computing in social network," IEEE International Conference on Computer & Communications. IEEE, 2017.
[3] Antorweep Chakravorty, Rong Chunming, "Ushare: user controlled social media based on blockchain," International Conference on Ubiquitous Information Management & Communication. ACM, 2017.
[4] Chen Yize, Li Quanlai, Wang Hao, "Towards trusted social networks with blockchain technology," 2018.
[5] Zhang Jie, Xue Nian, Huang Xin, "A Secure system for pervasive social network-based healthcare," IEEE Access, vol. 4, pp. 9239-9250, 2016.
[6] Md Arquam, Anurag Singh, Rajesh Sharma, "A blockchain based secure and trusted framework for information propagation on online social networks." Social Network Analysis and Mining, vol. 2021, pp. 49.
[7] Kiktenko E O, Pozhar N O, Anufriev M N, et al., "Quantum-secured blockchain," Quantum Science & Technology, vol. 3, no. 3, pp. 035004-035011, 2017.
[8] Ma Li, Tao Lixin, Gai Keke, Zhong Yong, "A novel social network access control model using logical authorization language in cloud computing," Concurrency & Computation Practice & Experience, vol. 14, no. 29, pp. 1-17, 2017.
[9] Florian Idelberger, Guido Governatori, Régis Riveret, Giovanni Sartor, "Evaluation of logic-based smart contracts for blockchain systems," Rule Technologies. Research, Tools, and Applications. Springer International Publishing, vol. 2016, pp. 167-183.
[10] Elisa Bertino, Barbara Catania, Vincenzo Gervasi, Raffaet àAlessandra, "Active-U-Datalog: integrating active rules in a logical update language," ILPS '97: International Seminar on Logic Databases and the Meaning of Change, Transactions and Change in Logic Databases, vol. 1996, pp. 107-133.
[11] Hu Jingwen, Zhong Yong, "A method of logic-based smart contracts for blockchain system.," Proceedings of the International Conference on Data Processing and Applications, Guangdong, China, vol. 2018, pp. 58-61.
[12] Nicola Atzei, Massimo Bartoletti, Tiziana Cimoli, "A Survey of attacks on ethereum smart contracts (SoK)," International Conference on Principles of Security and Trust. Springer, Berlin, Heidelberg, vol. 2017, pp. 164-186.