BLOCKCHAIN NODE MANAGEMENT IN SMART GRID

Mila Ilieva-Obretenova

University of Mining and Geology "St. Ivan Rilski", 1700 Sofia; mila.ilieva@mgu.bg

ABSTRACT. Blockchain is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way. It is typically managed by a peer-to-peer network collectively adhering to a protocol for inter-node communication and validating new blocks. In Bulgaria were represented: the idea for application of blockchain in business environment, the blockchain technology – as a linked list with different key on every step, the structure of a node and the information models for a node management with a large granularity. The paper represents the second stage of project Blockchain (permissioned) in Smart Grid: representing of main classes managed objects; managed objects classes for blockchain network installation with accent on security and performance; detail the models for node management with accent on configuration. The models are designed for user interface developers, professors, and students. On the base of management functions managed objects classes are defined by the usage of UML (Unified Modelling Language). From the structure diagrams class diagrams are used. Class diagram describes objects types in the system and different types of static relationships among them. Represented are specific attributes of selected objects classes. The refined information model with more precise granularity of managed objects classes represents a step forward to development of user interface.

Keywords: Blockchain, distributed ledger, linked list, peer-to-peer network, transaction

УПРАВЛЕНИЕ НА БЛОКЧЕЙН ВЪЗЕЛ В SMART GRID

Мила Илиева-Обретенова

Минно-геоложки университет „Св. Иван Рилски", 1700 София

РЕЗЮМЕ. Блокчейн представлява отворено разпределено счетоводство, което може да записва транзакции между две страни ефективно, доказуемо и по постоянен начин. Блокчейн се управлява чрез мрежа „от точка до точка", колективно придържаща се към протокол за междувъзлова комуникация и валидираща нови блокове. В България бяха представени: идеята за прилагане на блокчейн в бизнес среда, технологията на блокчейн – като свързан списък с нов ключ на всяка стъпка; структурата на възел и информационните модели за управление на възел с едра гранулност. Статията представлява втори етап на проект Блокчейн (фирмена) в smart grid: представяне на общи класове управлявани обекти; на класове управлявани обекти за инсталиране на блокчейн мрежа с акценти върху сигурността и техническите характеристики; детайлизиране на моделите за управление на възлите с акцент върху конфигурацията. Моделите са предназначени за разработчици на потребителски интерфейс, преподаватели и студенти. На базата на функциите за управление се дефинират класове управлявани обекти, като се прилага унифициран език за моделиране (UML). От диаграмите на структура се използват диаграми на класове. Диаграмата на класове описва типовете обекти в системата и различните видове статични взаимоотношения между тях. Представени са специфични атрибути на избрани класове обекти. Усъвършенстваният информационен модел с прецизирана гранулност на класовете управлявани обекти представлява стъпка напред към разработване на потребителския интерфейс.

Ключови думи: Блокчейн, мрежа от точка до точка, разпределено счетоводство, свързан списък, транзакция

Introduction

Blockchain is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way (Iansiti, Lakhani, 2017). It is typically managed by a peer-to-peer network collectively adhering to a protocol for inter-node communication and validating new blocks (Raval, 2016). China suggests blockchain to become one of the first like networks, build and maintained from government (Stockton, 2020). Europe walks slowly as it creates international blockchain association with the aim regulatory clarity for the early participants (Chandler, 2019). In smart grid blockchain could guarantee effective energy distribution, keep low losses, high quality, and security of power supply. Therefore, node and network management of blockchain is timely and necessary. The international scientific community suggests different conceptual frameworks. For example, Ullah et al. (2019) suggest model for bidirectional communication in smart grid with blockchain, but without considering the life cycle of the network and its functional areas. Chren et al. (2018), Ge et al. (2019), Hrabovska et al. (2019), and Schwarcbacher et al. (2018) show different aspects of smart grid, but without blockchain. Krishna et al. (2020) represents blockchain application for analysis, but with no realization`s suggestions in different business environments. Nedyalkova (2014) represents methodical basics for assessment for internal audit, also in smart grid, but without blockchain. Agung (2020) suggests architecture of smart grid with blockchain nodes – Fig.1.

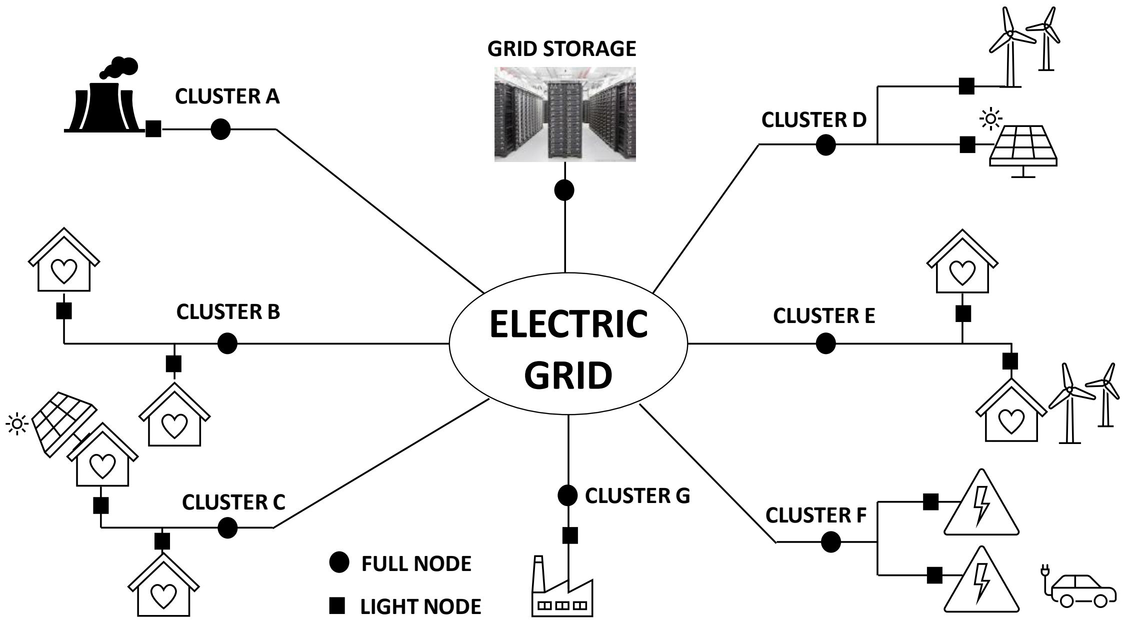

Fig. 1. Architecture of smart grid with blockchain nodes

Cluster A consists of government power plants, cluster B consists of residential buildings, clusters C and E consist of residential buildings too, but some can generate own electricity. Cluster D consists of private power plants, cluster F consists of public electric outlets and cluster G consists of manufacturing enterprise. Each cluster has own full node, and each subscriber (user/producer) has own light node. The grid storage doesn`t have light node because it doesn`t initiate transaction. Full nodes consist of blockchain copy and their basic functions include confirmation of transactions and consensus support with other full nodes. Full nodes are added to existing distribution points as step-down transformers at distribution lines to users. Light nodes are used in everyday transactions. They apply Simple Payment Verification (SPV) for transactions validation. They don`t consist of the full copy of blockchain, so they depend of full nodes. Light nodes are connected to or substitute the existing electricity meters. Management data for both nodes are missing in this model. In Bulgaria were represented: the idea for blockchain application in business environment (Dimov, Stratiev, 2018), blockchain technology – as a linked list with a new key on every step; node structure and information models for node management with a large granularity. The paper represents the second stage of project Blockchain (permissioned) in smart grid: representing of main classes managed objects; representing of managed objects classes for installation of blockchain network with accent on security and performance and detail of models for node management with accent on configuration. Models are purposed for user interface developers, professors, and students.

Methodology

On the base of management functions are defined classes managed objects applying Unified Modeling Language – UML (Fowler, 2004). From the structure diagrams are used class diagrams. A class diagram describes objects types in the system and the different types static relationships among them. Diagrams show also features and operations of classes and limitations for the way connecting objects. Features are one term, but they are represented with two quite different notations: attributes and associations. The notation for the attribute describes a distinct feature as text line (second row) in the rectangle, symbolizing the class. Association is a directed line between two classes and its direction is from class-source to class-aim. The name of the feature is set on the aimed end of the association with its majority. The aimed end of the association is connected to the class which is the feature type. The majority of a feature is a note for how many objects could complete the feature. Operations are actions which a class could realize. The “Part-of”-relationship between objects is shown with a rhombus and a line. In the paper are represented specific attributes of selected objects classes. Operations are not represented.

Results

1. Main managed objects classes

In this section are defined managed objects (MO) classes for the actors, the object of management and the way of management. The actors are: Network Operator – MO Operator, Transaction Provider (Network Subscriber) – MO NetworkSubscriber, Transaction Subscriber (Network User) – MO NetworkUser. Objects of management are: Blockchain Network – BCNetwork, Transaction Provider and Transaction Subscriber. Objects of management are managed by profile – MO Profile. Management elements are explained by functional areas on Open Systems Interconnection: MO NConfiguration, MO NMaintenance, MO NSecurity, MO NAccounting, and MO NPerformance. Managed objects for network are represented on Fig.2. In the next sections are identified the managed objects for network installation and for network nodes.

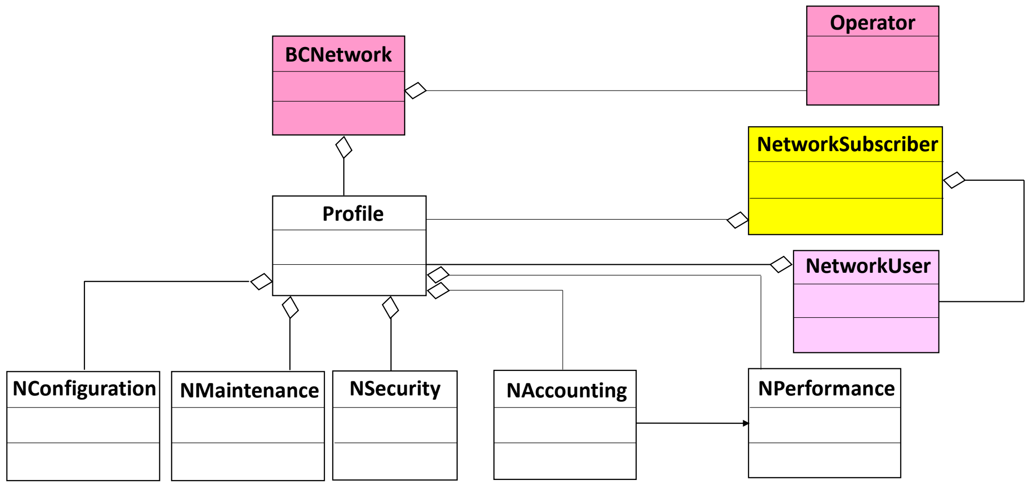

Fig. 2. UML diagram of main managed objects classes for blockchain network

2. Main managed objects classes for blockchain network installation

MO Operator represents the information for a person or usually organization, providing the network. MO BCNetwork represents a set of interconnected managed objects (logical and physical), able to transfer information. These objects have one or more general characteristics. In example they could be delegated to single user, producer or provider, and could relate to specific transaction network. A blockchain network could be embedded in another one. MO NetworkProfile consists of the whole information for a management of a blockchain network. MO BCNConfiguration represents the main configuration of blockchain. It consists of all available network elements. Attribute FullNode represents the full nodes in blockchain network. Attribute LightNode represents the light nodes in blockchain network. MO BCNMaintenance represents management information for network maintenance. MO BCNSecurity represents the rights on network security and the corresponding data. This object must consist of a root key, which enables the monitoring of all transactions in the platform. MO BCNAccounting represents the management data, connected to the financial aspect of network provision. MO BCNPerformance represents the storage and generalization of managed data of performance and usage of specific network. So, it is reached separation of users' identity e.g. separation of transactions and identities. Important attribute is network speed e.g. 15 transactions/sec, hard coded in the protocol. Or 1 transaction is completed for 67 milliseconds. For comparison SCADA sends information every 2-3 sec. MO CoopNetwork represents data for the network, with which is cooperated blockchain. MO SNetwork represents signaling network, used for communication between full and light nodes. MO BNetwork represents the network serving the energy transport (bearer network). MO Configuration represents the configuration of bearer network. Fig.3 shows UML diagram of managed objects classes for Network Installation.

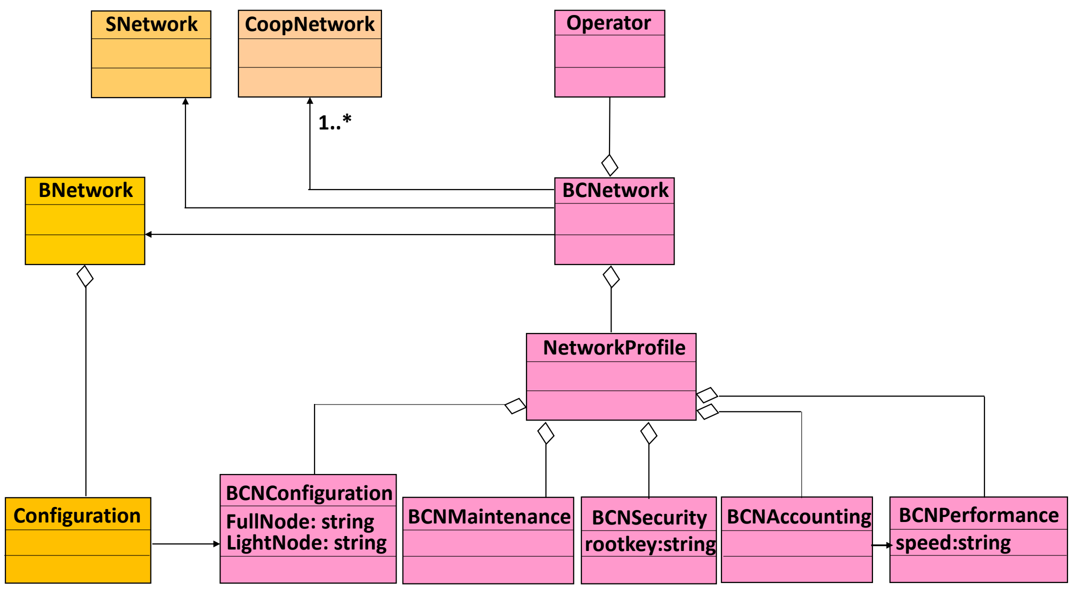

Fig. 3. UML diagram of managed objects classes for Network Installation

3. Managed Objects classes for network nodes

3.1. Managed objects classes for light node Transaction Switching Point (TSP)

Aim of this node is continuously balancing of generation and load and keeping the microgrid`s connection to the main grid stable. MO Switch represents information for switch in bearer network. MO SCF represents information about Supply Control Function, completed from switch in bearer network. MO BAccounting consists of data for accounting of bearer network. MO SCFConfig represents the configuration of specific SCF. MO RoutingTable represents the routing information of incoming requests in SCF. MO TSP represents management of a light node. MO TSF represents management information for a Transaction Switching Function. MO TSFConfig represents the configuration of specific TSF. MO TSFMainten represents the maintenance conditions of a TSF. MO TSFSecurity represents the security rules of TSF. MO TSFPerform represents performance parameters of TSF and their management. Attribute PowerTurn represents the ramp turn of wind generator when the wind power changes. The speed of turning is 1MW/min. Attribute NPeriods shows the number of periods, during which the node checks for output changes of wind power. One transaction execution is 67 milliseconds, the period by frequency 50 Hz is 20 milliseconds. So, the number of periods is 67/20 = 3.35~4. This node also creates its own forecasts of load and generation based on weather forecasts and intelligent processing of statistical load data. The forecasting models are updated in real time to compute the optimal settings for power end energy storage in the system. The TSFAccounting represents management information for accounting of TSF. MO TriggerTable represents information for all transactions` triggers in blockchain. Attribute PowerCh shows threshold of charge on renewable output, by which it is necessary to turn on or off, e.g. 1 MW. MO TriggerInfo represents the routing information, connected with possible directions for network access for specific subscriber. This object is also necessary for transactions, allowing subscriber several sources of power supply e.g. routing depending on time and weather. If the wind stops blowing, the system instantly compensates by discharging the batteries to the grid, as power from the main grid gradually increases to match demand. When the wind starts blowing again, the batteries immediately respond by absorbing a portion of the generated wind power. As they charge, the network turns off and the power supply is meted only from wind generator. The same is valid for solar parks. Attribute Manufacturer shows that in a specific moment the subscriber is turned on as a manufacturer. Attribute TSubscriber shows that in a specific moment the subscriber is turned on as a user. Attribute Renewable shows that in a specific moment the subscriber (could) receive energy from renewable. Attribute GridStorage shows that in a specific moment the subscriber (could) receive energy from grid storage. Attribute Network shows that in a specific moment the subscriber (could) receive energy from the network. This object belongs to the area Transactions management too. MO FSManager represents supporting mechanism for competitive realizations of blockchain transactions and transactions out of blockchain in a request. MO BCSManager represents interaction mechanism with TCF for transactions provision. It discovers events which might be reported to active transaction realization, and manage TSF resources, which must support transactions realizations. MO FEAManager represents mechanism for information transfer with other functional element (or node) by notifications. MO TCF will be represented in next section. MO BSManager represents a mechanism for return in requests, after the transaction have been completed (confirmed). MO NBCFManager represents requesting mechanism of transaction feature out of blockchain. MO TSFUsageLog represents the collected usage entries of TSF. MO TSFAccountingLog represents the collected accounting entries of TSF for transaction execution. Fig.4 shows UML diagram of managed objects classes for light node TSF.

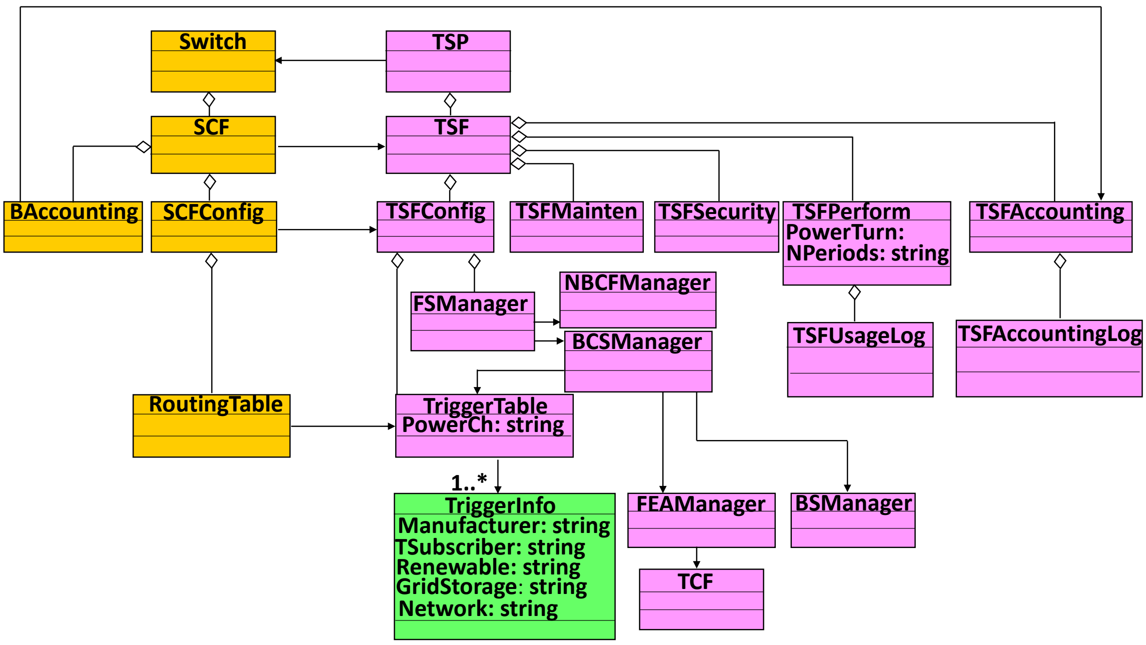

Fig. 4. UML diagram of managed objects classes for light node TSP

3.2. *Managed objects classes for full node Transaction Control Point (TCP)*

MO TCP represents information for node TCP. This node consists of three functions: TCF (Transaction Control Function), TDF (Transaction Data Function) and TRF (Transaction Resource Function).

3.2.1. Managed Objects Classes for function TCF. MO TCF represents information for control management over a transaction. MO TCFMaintenance represents maintenance conditions of TCF. MO TCFSecurity represents the security rules of TCF. MO TCFConfiguration represents the configuration of TCF. MO TCFPerformance represents the performance parameters of TCF and their management. MO TCFAccounting represents the management information for accounting of TCF. MO TCFPreventiveFunction represents the testing programs of TCF in normal working conditions. MO LogicExecutionEnvironment represents environment for execution of transaction logic with all included managers, programs, and data, e.g. Hyperledger. MO LogicExecutionManager represents information for functionality, processing and controlling of transaction execution. MO ProgramLibrary represents resource for storage of different programs in TCF. MO Program represents the description of a program with transaction logic (consensus), e.g. PoW (Proof of Work), PoSt (Proof of Stake), PoSe (Proof of Service) and PoI (Proof of Importance). This object belongs to area Transactions Management too. MO DataAccessManager represents information for storage, management, and access to shared information in TCF and for access to remote information in other nodes and functional elements by MO FEAccessManager. MO FEAccessManager and MO TSF were represented in section 3.1. Managed Objects Classes for light node Transaction Switching Point (TSP). MO TRF (Transaction Resource Function) and MO TDF (Transaction Data Function) will be represented in next sections. MO TCFUsageLog represents the collected entries for usage of TCF. MO TCFAccountingLog represents the collected entries for accounting of TCF for transaction execution. Fig.5 shows UML diagram of managed objects classes for full node TCP.

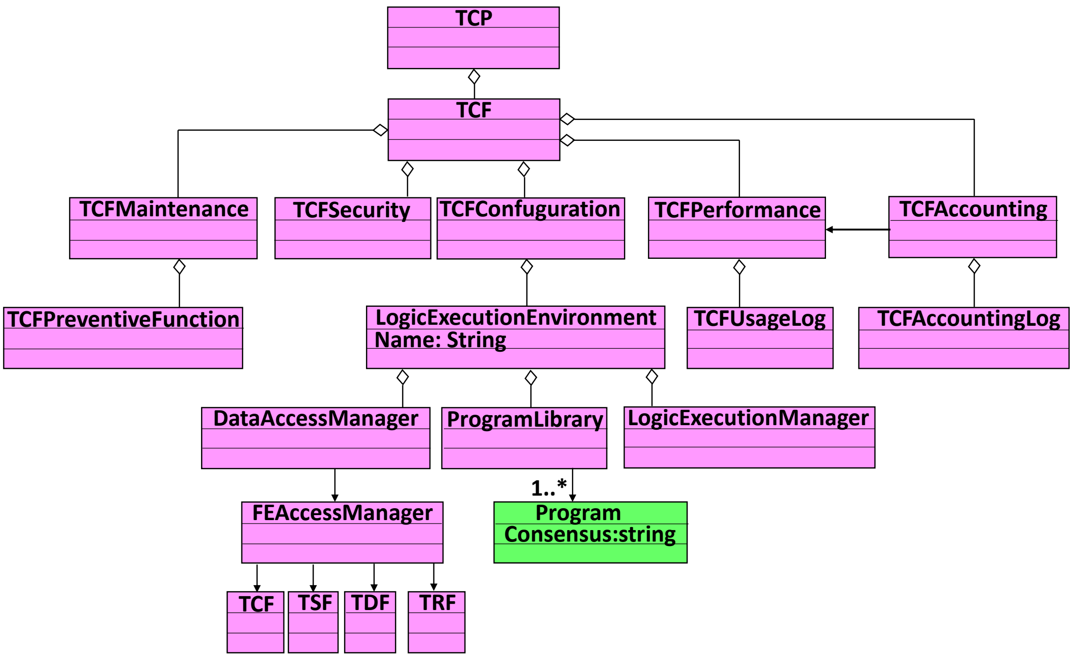

Fig. 5. UML diagram of managed objects classes for full node TCP

3.2.2. Managed Objects Classes for function TDF. MO TDF represents management information of data for transactions users. MO TDFConfiguration represents the configuration of a specific TDF. MO TDFPerformance represents the parameters for quality work of TDF and their management. MO TDFAccounting represents management information for accounting of TDF. MO TDFMaintenance represents maintenance conditions for TDF. MO TDFSecurity represents the security rules of TDF. MO TDFDataManager represents information for storage, management, and access to data in TDF. MO TDFDataBase represents information for the data base in TDF. MO FEAccessManager was represented in section 3.1. Managed Objects Classes for light node Transaction Switching Point (TSP). MO TDFData represents an entry in functional element TDF for specific transactions user, e.g. Received/Consumed/Rest/Sold Energy. MO Template represents the description of entry format for each transaction in blockchain. The last two objects belong to the area Transactions Management too. MO TCF was represented in the previous section. Fig.6 shows UML diagram of managed objects classes for function TDF.

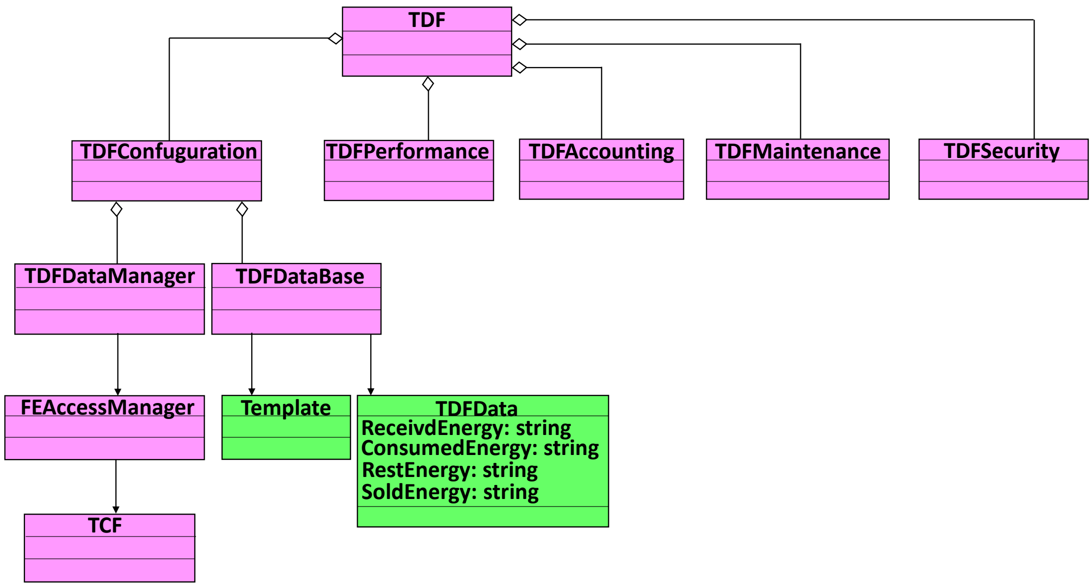

Fig. 6. UML diagram of managed objects classes for function TDF

3.2.3. Managed Objects Classes for function TRF. MO TRF represents management information for producer in transactions. MO TRFConfiguration represents the configuration of a specific TRF. MO TRFPerformance represents performance values of TRF and their management. MO TRFAccounting represents the management information of accounting in TRF. MO TRFMaintenance represents maintenance conditions for TRF. MO TRFSecurity represents the security rules of TRF. MO FEAccessManager was represented in previous sections. MO ResourceManager represents information for resources, managed from TRF. MO TRFUsageLog represents collected entries for usage of TRF. MO Resource represents description of specialized resource, used in a transaction. MO PowerPlant describes the data for the used power plants, e.g. thermal and nuclear. MO Renewables describes the data of used renewables, e.g. hydroelectric powerplants, solar parks, wind generators, generators with biomass etc. MO GridStorage describes the data of used electricity storages, e.g. pumped and battery storages. The last four objects belong to the area Transactions Management too. MO TCF was represented in section 3.2.1. Managed Objects classes for function TCF. Fig. 7 shows UML diagram of managed objects classes for function TRF.

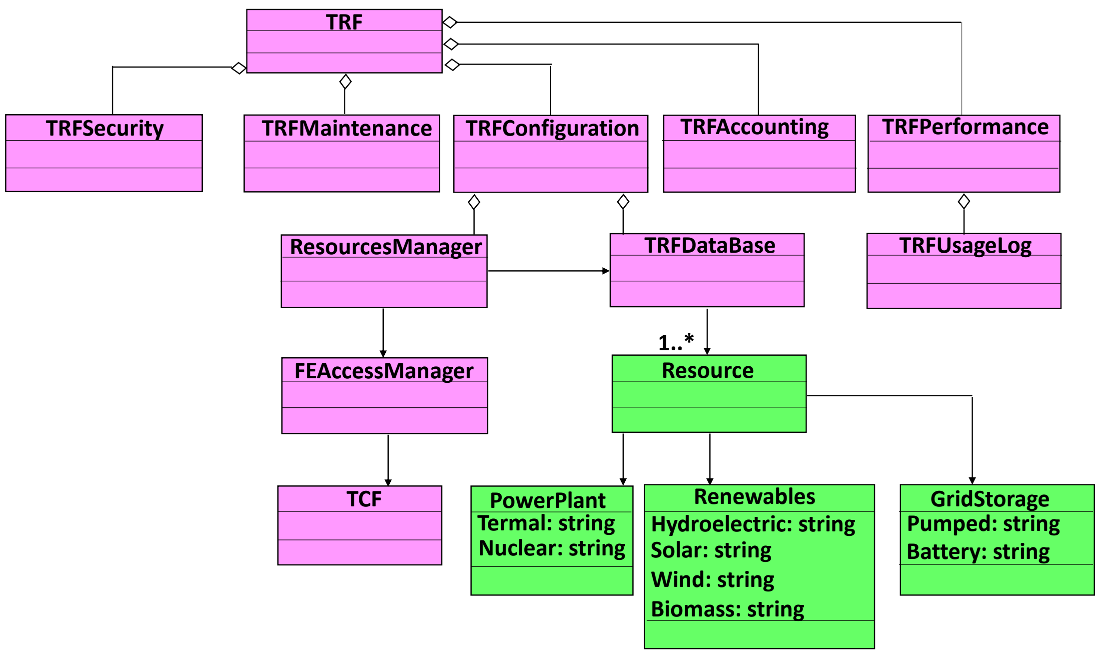

Fig. 7. UML diagram of managed objects classes for full node TRF

Conclusion

Deployment of blockchain is still in testing stage. Although the paper shows its advantage – specialized (permissioned) networks could be easily scaled because all confirmations happen in network. Considered is the question about separation of transactions and identity of users. A detail information model represents refined granularity of managed objects classes about Blockchain Network Configuration, Blockchain Network Security, Blockchain Network Performance, Performance of Transaction Switching Function, Trigger Table, Trigger Information, Logic Execution Environment, Program, User data and Resources - Power Plants, Renewables and Grid Storage. With these improvements the model represents a good base for user interface development. The permissioned blockchain has its disadvantages, but the long-term perspective forecasts deployment in the next line: business platforms, local, national, and international networks.

Acknowledgements. This research was supported by the Bulgarian projects under contract: **"**Study of the possibilities for application of DLT using blockchain in data sharing in mining activities and electricity production from biogas **"**MEMF-164.

References

Agung, A. A. G., Rini Handayani, 2020, "Blockchain in Smart Grid", *Journal of King Saud University – Computer and Information Sciences,* DOI: 10.1016/j.jcsuci.2020.01.002

Chandler, S., 2019, "EU launches International blockchain association bringing crypto one step closer to mainstream adoption", *Cointelegraph,* https://cointelegraph.com/news/eu-launches-international-blockchain-association-bringing-crypto-one-step-closer-to-mainstream-adoption

Chren, S. et al., 2018, "Reliability Data for Smart Grids: Where the real Data can be found", *Smart City Symposium Prague (SCSP)*

Dimov, E., Stratiev., N., 2018, "The Investment of Cryptocurrencies: A QUES for Earnings Lost", *University of Mining and Geology, Journal of Mining and Geological Sciences, Volume 61,* Part IV: Humanitarian Sciences and Economics, Sofia, ISSN 2535-1206, pp.33-36

Fowler, M., 2004, *"UML Distilled: A Brief Guide to the Standard Object Modeling Language". 3rd ed*. Addison-Wesley Professional

Ge, M. et al., 2019, "Data Quality Management Framework for Smart Grid Systems", *International Conference on Business Information Systems BIS 2019: Business Information Systems,* pp. 299-310, DOI: 10.1007/978-3-030-20482-2_24

Hrabovska, K. et al., 2019, "Smart Grid and Software Testing Process Models", *Smart Cities Symposium Prague (SCSP)*

Iansity, Marco; Lakhani, Karim R. (January 2017). "*The Through about Blockchain*". *Harvard Business Review. Harvard University.*

Krishna, G. J. et al., 2020, *"Smart Blockchain: A Formidable Combination of Analytics and Blockchain – A Survey", In book: Smart Systems Design, Applications, and Challenges,* DOI: 10.4018/978-1-7998-2112-0.ch003

Nedyalkova, P., 2014, "Methodological Foundations of the Assessment of Internal Audit", *Izvestia Journal of Economics, Varna, 2014, No.1,* pp. 65-80

Raval, Saraj (2016). "*What is a Decentralized Application?*". *Decentralized Applications: Harnessing Bitcoin`s Blockchain Technology.* O`Railly Media, Inc. pp. 1-2. ISBN 978-1-4919-2452-5. OCLC. 968277125

Schvarcbacher, M. et al., 2018, "Smart Grid Testing Management Platform" (SGTMP), *Applied Sciences 8(11):2278,* DOI: 10.3390/app8112278

Stockton, N., 2020, "China launches national Blockchain Network in 100 Cities", *https://spectrum.ieee.org/computing/software/china-launches-national-blockchain-in-100-cities*

Ullah, Z. et al., 2019, "Smart Grid Blockchain (BC) Conceptual Framework: Bi-Directional Models for Renewable Energy District and Utility", *15th International Conference on Emerging Technologies*