Open Access Library Journal
2024, Volume 11, e11261
ISSN Online: 2333-9721
ISSN Print: 2333-9705

An Approach for Smart Gateways to Collect Data and Automate It on Block-Chain

Jingli Mu, Xiaodong Wang, Qiang Liu, Ran Li, Kun Wang, Lei Zhu, Hongbin Chang, Jiabao Lei*

Technology R&D Center, China Energy Digital Intelligence Technology Co., Ltd., Beijing, China
Email: *2712198662@qq.com

How to cite this paper: Mu, J.L., Wang, X.D., Liu, Q., Li, R., Wang, K., Zhu, L., Chang, H.B. and Lei, J.B. (2024) An Approach for Smart Gateways to Collect Data and Automate It on Block-Chain. *Open Access Library Journal*, **11**: e11261.
https://doi.org/10.4236/oalib.1111261

Received: Jan 25, 2024
Accepted: August 18, 2024
Published: August 21, 2024

Abstract

Blockchain is a technology that has received a lot of attention in recent years, employing a distributed consensus mechanism that ensures the credibility and integrity of data through the verification and approval of multiple parties [1] [2]. Due to its decentralization, immutability, security, reliability, and traceability, blockchain is widely used in various fields such as finance, logistics, and healthcare. Currently, common methods of uploading to the blockchain include interfacing business systems to the blockchain, data lake to blockchain, and manual uploading, among others [3]. These methods have brought significant inconvenience to the uniform management of data uploading across systems. Moreover, these methods do not involve directly collecting source data from endpoint devices to the blockchain, leading to a "vacuum" period for data credibility control between the time data is collected and uploaded to the blockchain. During this period, there is a risk of data tampering, and the security, integrity, and authenticity of the data cannot be effectively guaranteed. Therefore, this paper proposes a solution for automatically collecting and uploading carbon emission data from thermal power plants directly to the blockchain, aimed at ensuring the "last mile" before data is uploaded to the blockchain is secure, trustworthy, and immutable.

Subject Areas

Blockchain Technology

Keywords

Blockchain Technology, Industrial Internet Gateway, Data Acquisition, Spacetime Positioning

1. Introduction

This project will take a thermal power plant as a pilot, integrating various technologies such as terminal device identity management, real-time data acquisition of equipment, spatiotemporal positioning, and blockchain to carry out the technical feasibility of direct data collection from end-side terminals and on-chain intelligent device preparation. It aims for subsequent blockchain application scenarios to achieve isolation between humans and equipment, and between humans and data through direct data collection and on-chain processes, thereby eliminating the mode of manual data collection, significantly reducing the possibility of human intervention in the data, saving labor costs, and improving the efficiency of data on-chain. This furthers the construction of a trusted data source. At the same time, through feasibility studies, it is expected to form a data acquisition product line with on-chain functionality applicable in the thermal power scenario.

2. Ease of Use

2.1. Data Cochain

Data cochain refers to the process where users encrypt their own data and upload it to the public network side-chain, and form a data index on the public trust chain. Once the data is on the chain, aside from the decryption authorized by the owner's Data-key, no individual or organization has the capability to access your data, thereby ensuring absolute data security.

2.2. Data Acquisition Gateway

The data acquisition gateway is an industrial equipment data collection and conversion device. It integrates data collection, protocol analysis, and edge computing, remote control, and cloud access into one smart device. It is suitable for data collection and remote management of various types of equipment, solving problems related to industrial equipment IoT, data collection, and remote monitoring.

By combining the technologies of data on-chain and data collection gateways, the direct collection of business data and its automated on-chain process are realized. This saves costs and improves efficiency with a certain degree of usability.

3. Methodology

The main content of this study includes a feasibility study of the technology.

3.1. Spatiotemporal Location of Data

In this project, the Beidou chip is used to add spatio-temporal positioning capabilities to the terminal. Using the most advanced Beidou technology, data can be precisely located within 50 meters, and the timing of data to users can be precise within 0.1ms. Therefore, users can quickly and conveniently monitor all business data acquisition's precise location and time through the browser. Real-time data monitoring and transmission under spatio-temporal interaction enhances the credibility of the terminal and data.

3.2. Data Directly Uploaded to Blockchain

In this project, the smart data collection and blockchain terminal will directly collect data through semiconductor gas sensors, infrared gas sensors, electrochemical gas sensors, sampling equipment, sample preparation equipment, sample transfer equipment, and laboratory equipment. At the same time, the collected data will be automatically uploaded to the blockchain within the terminal, avoiding cumbersome intermediate processes, reducing costs, and increasing efficiency. Through the data collection function, these business data can be aggregated to form an internal big data platform for the group, performing data mining and analysis, and serving users accurately. Through the direct blockchain upload function, it avoids manual intervention in data collection and ensures the whole process of data retrieval and blockchain upload is secure, trustworthy, and tamper-proof.

3.3. Intelligent Data Collection Blockchain Terminal

In this project, each of the aforementioned functions will be developed and tested one by one. Once all functions are implemented, we will explore the feasibility of integrating these two major functions into the gateway device, assess whether each function is affected in interaction, whether performance is impacted, and whether the data flow logic meets the design requirements, etc., in order to evaluate if the terminal can meet the basic requirements of performance standards and complete functionality, thus achieving technical feasibility.

4. Experimental

4.1. Direct Data Collection (Taking Location Data as an Example)

Through the connection and interaction between the positioning module (chip)/data acquisition module and the gateway, the chip's position and time information are available in real-time. Devices used include the edge gateway (Guoneng Industrial Internet Intelligent Gateway D600), positioning module (Quectel EC20), and data acquisition sensor (NEPRICD-2000). Taking the interaction between the gateway and the positioning module as an example, this describes the method of direct data acquisition onto the chain. Method: According to the official manual of the EC20 module, it is understood that the corresponding AT control command channel for the module is /dev/ttyUSB2, as shown in Figure 1.

Connect to the module's corresponding AT command channel through serial communication, and then it is possible to read information and data from the module through AT commands, as shown in Figure 2.

Through AT commands, it is possible to query the network status, signal quality, GNSS information, and other parameters; by reading the positioning information through AT commands, it will be organized into the module's own format and sent to the device through serial communication, as shown in Figure 3.

Module VID and PID	USB Drivers	Interface Number	Device Names	Functions
EC20-CE/ EC25 series/ EG25-G/ EG25-GL/ EM05 series: VID: 0x2c7c PID: 0x0125	USB serial option	0	/dev/ttyUSB0	DIAG
		1	/dev/ttyUSB1	GNSS
		2	/dev/ttyUSB2	AT command
		3	/dev/ttyUSB3	Modem
EC21 series/ EG21-G: VID: 0x2c7c PID: 0x0121 **EG91 series:**	GobiNet	4	usb0 /dev/qcqmi0	USB network adapter Configure the type of USBnet interface as RmNet by **AT+QCFG="usbnet",0**

Figure 1. EC20 module control command.

```
fd = open("/dev/ttyUSB2", O_RDWR | O_NOCTTY | O_NONBLOCK);
if(fd<0)
{
    printf("EC20:cann't open ttyUSB\n");
    return 0;
}
init();
```

Figure 2. AT command.

该命令可用于获取定位信息。执行该命令前，必须通过命令 **AT+QGPS** 打开 GNSS。如果定位失败，将根据对应情况返回**+CME ERROR:** <errcode>，设置后立即生效。

AT+QGPSLOC 获取定位信息	
测试命令 AT+QGPSLOC=?	响应 +QGPSLOC: <UTC>,<latitude>,<longitude>,<HDOP>,<altitude>,<fix>,<COG>,<spkm>,<spkn>,<date>,<nsat> OK
设置命令 AT+QGPSLOC=<mode>	响应 +QGPSLOC: <UTC>,<latitude>,<longitude>,<HDOP>,<altitude>,<fix>,<COG>,<spkm>,<spkn>,<date>,<nsat> OK 若错误与 ME 功能相关，则返回： **+CME ERROR:** <errcode>
参考	

Figure 3. Obtaining positioning information (1).

In the code, instructions described in the manual are sent to obtain data such as location information, as shown in Figure 4.

```
int get_gnss()
{
    int ret;
    int i = 0;
    char send_buf[50] = "AT+QGPSLOC?\r\n";
    ret = write(fd, send_buf, 15);
    if (ret < 0)
    {
        //printf("send error!!\n");
        return -1;
    }
    return 0;
}
```

Figure 4. Getting location information (2).

Finally, follow the data sequence in the instruction manual, and data parsing can obtain the required positioning information; the interaction method between the gateway and the data collection module is similar to that between the gateway and the positioning module. Since it is not the core innovation of this patent, it will not be elaborated on further.

4.2. Data Cochain

Taking the positioning module as an example, the data format obtained is shown in Figure 5.

```
data:39.949627N;116.394210E;24-1-8 11:38:43 addr:127.0.0.1:65432 count:38
data:39.949673N;116.394227E;24-1-8 11:39:13 addr:127.0.0.1:65432 count:38
data:39.949683N;116.394233E;24-1-8 11:39:43 addr:127.0.0.1:65432 count:38
data:39.949683N;116.394227E;24-1-8 11:40:13 addr:127.0.0.1:65432 count:38
data:39.949680N;116.394218E;24-1-8 11:40:43 addr:127.0.0.1:65432 count:38
data:39.949688N;116.394227E;24-1-8 11:41:13 addr:127.0.0.1:65432 count:38
data:39.949688N;116.394242E;24-1-8 11:41:43 addr:127.0.0.1:65432 count:38
data:39.949683N;116.394242E;24-1-8 11:42:14 addr:127.0.0.1:65432 count:38
data:39.949680N;116.394242E;24-1-8 11:42:44 addr:127.0.0.1:65432 count:38
data:39.949680N;116.394242E;24-1-8 11:43:14 addr:127.0.0.1:65432 count:38
data:39.949650N;116.394180E;24-1-8 11:43:44 addr:127.0.0.1:65432 count:38
```

Figure 5. Data cochain.

As shown in the figure above, the Linux system within the gateway has read the data transmitted by the positioning module. Taking the first line as an example, "data" represents the positioning data, where 39.949627N represents latitude, 116.39420E represents longitude, and 24-1-8 11:38:43 represents the current date and time. "addr" represents the IP address, which supports the subsequent blockchain program to listen to the positioning data.

After receiving data, the gateway can directly invoke the contract SDK to go on-chain within the gateway according to the on-chain mechanism of the State Energy Blockchain. The State Energy Chain SDK acts as a bridge between the business module and the State Energy Chain, supporting bi-directional TLS authentication and providing a secure and reliable encrypted communication channel. The task of the on-chain contract SDK is to ensure that business data is transmitted to the blockchain accurately, reliably, and traceably. The implementation is as follows.

Implementation method: First, the on-chain contract is written, as shown in Figure 6.

```
// 新建存证对象
func NewFact(latitude string, longitude string, time int32) *Fact {
    fact := &Fact{
        latitude: latitude,
        longitude: longitude,
        time:     time,
    }
    return fact
```

Figure 6. Creating a new evidence object.

Creating a new proof of existence object, latitude, longitude, and time, is exposed externally for users to call the contract to complete the on-chain operation of the above three parameters.

```go
// 对外暴露 find_by_latitude 方法，供用户由 SDK 调用
//export find_by_latitude
func findByLatitude() {
    ctx := NewSimContext()
    // 获取参数
    latitude, _ := ctx.ArgString("latitude")
    // 查询Json
    if result, resultCode := ctx.GetStateByte("fact_json", latitude); resultCode != SUCCESS {
        // 返回结果
        ctx.ErrorResult("failed to call get_state, only 64 letters and numbers are allowed. got key:" + "fact" +
", field:" + latitude)
    } else {
        // 返回结果
        ctx.SuccessResultByte(result)
        // 记录日志
        ctx.Log("get val:" + string(result))
```

Figure 7. Contract method (1).

After the contract is written, it can be installed and compiled on a business chain within the National Energy Blockchain Enablement Platform. Then, you can directly invoke the contract to go on-chain. The part of the contract SDK code is shown in **Figure 7** and **Figure 8**.

```go
fmt.Printf("data:#{string(data[:n])} addr:#{addr} count:#{n}\n")
latitude_globe = []byte(string(data[0:4]))
longitude_globe = []byte(string(data[7:12]))
dateAndTime_globe = []byte(string(data[15:18]))
fmt.Printf("latitude:#{string(latitude_globe[:])}\n")
fmt.Printf("longitude:#{string(longitude_globe[:])}\n")
fmt.Printf("DateAndTime:#{string(dateAndTime_globe[:])}\n")
```

Figure 8. Contract method (2).

The test results are shown in **Figure 9**, the positioning data was successfully uploaded to the blockchain, and the process of uploading measurement data to the blockchain is similar, so it will not be reiterated.

```
DateAndTime:3122812302
2024/01/08 13:47:42 invoke contract failed, [code:4]/[msg:]

root@RK356x-GNXK:/userdata# ./main
data:39957N;116372E;23122812302 addr:127.0.0.1:65432 count:26
latitude:3995
longitude:11637
DateAndTime:;23
2024/01/08 13:55:59 invoke contract failed, [code:4]/[msg:]

root@RK356x-GNXK:/userdata# ./main
data:39957N;116372E;23122812304 addr:127.0.0.1:65432 count:26
latitude:3995
longitude:11637
DateAndTime:231
invoke contract success, resp: [code:0]/[msg:]/[contractResult:result:"399511637"
gas_used:9980371 contract_event:<topic:"topic_vx" tx_id:"17a863d3aaf0d114cadce3c90
b3541e48f3da5448d1a4a429458b6ae0f2ec1c6" contract_name:"demo1" contract_version:"v
1.0" event_data:"3995" event_data:"11637" > ]
QUERY claim contract resp: message:"SUCCESS" contract_result:<result:"{\"latitude\
":\"3995\",\"longitude\":\"11637\",\"time\":231}" gas_used:8106572 > tx_id:"17a863
d3ce36b677cabb928398838b20f19cf6ac68ab43ad9d0f3b386c191cb8"
```

Figure 9. Test result.

5. Conclusion

In conclusion, it can be considered that the technology research for direct sourcing to blockchain is feasible. After the mature product of the smart data collection blockchain terminal is launched, it has a certain degree of universality and can undergo adaptive modifications at a lower cost, therefore, it can be widely used in a variety of production scenarios. At the same time, the smart data collection blockchain terminal itself is small in size and can be placed in cramped spaces, further enhancing adaptability and competitiveness. Finally, due to its low manufacturing cost, large-scale production and application will greatly improve the efficiency of blockchain integration and reduce production costs in various fields.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Christidis, K. and Devetsikiotis, M. (2016) Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*, **4**, 2292-2303. https://doi.org/10.1109/ACCESS.2016.2566339

[2] Sim S.-H. and Jeong, Y.-S. (2021) Multi-Blockchain-Based IoT Data Processing Techniques to Ensure the Integrity of IoT Data in AIoT Edge Computing Environments. *Sensors*, **21**, Article 3515. https://doi.org/10.3390/s21103515

[3] Watanabe, H., Fujimura, S., Nakadaira, A., *et al.* (2016) Blockchain Contract: Securing a blockchain Applied to Smart Contracts. *IEEE International Conference on Consumer Electronics* (*ICCE*), Las Vegas, 7-11 January 2016, 467-468. https://doi.org/10.1109/ICCE.2016.7430693