Electronic devices based on unbiased decision methods in order to protect device memory

El OuerkhaouiAsmâa
[1]School EMG (EcoleMarocained'Ingénierie Rabat Morocco
elouerkhaoui.asmaa@gmail.com

Abstract

There are several factors that cause the corruption of data integrity through its flow on any kind of implementation such as redundancy, installation etc. The factor in question studied in this paper is different, it is the transactional process on which the sending and the reception of the data is based.
It is on this axis that we have innovated by moving to so-called atomic decisions.
The abundant resources offered today in any implementation are infinite but it is not because the resources are abundant that this guarantees the integrity of the data when it circulates and when it is modified/updated.
Electronic devices are today part of our daily lives. We rely on them to get live information and sometimes very sensitive information. But, synchronizing data between two or more separated Electronic devices is not enough studied to guarantee data integrity. Electronic devices will always have limits, one of them is memory and data transfer speed.
This paper proposes a solution to synchronize data we provide to our electronic devices to make the right decision with available memory and data speed transfer.

KeywordsSegregation responsibility, quantum decision theory, neuroScience, electronic devices, NeuroQuantology, cloud computing, cyber physical system.memory management, booting system, transfer speed, artificiel internet of things, data transfer protocols.

DOINumber:10.14704/nq.2022.20.8.NQ44887 **NeuroQuantology 2022; 20(8): 8652-8661**

INTRODUCTION

When it comes to synchronizing data a from device A to device B. How can we be sure that data a does not transform into value b where a is different from a?

In the process of synchronization, the end point can be corrupted because it's a human decision that triggered the synchronization final decision. Add to this, we all like to schedule this task and let automatic processes manage it. Since human decisions are managed in some areas with automatic processes, we propose that we replace transactions with quantum decisions. This concept is good because we all know how exhausting it is to start again because the automatic process fails.

The targeted areas are enormous, add to this, we need to schematize the concept and to do this we need to go from generic needs to specific one to focus on this schematization.

STATE OF THE ART

Managing risky choices: the concept of uncertainty Using Quantum decision Theory.

QDT describes a decision maker's choice as a stochastic event occurring with a probability that is the sum of an objective utility factor and a subjective attraction factor. QDT offers a prediction for the average effect of subjectivity on decision makers, the quarter law.

Digital identity is defined as a technological link between a real entity and virtual entities. It allows the identification of the individual online as well as the connection of this one with all the virtual communities present on the Web.

What is the aime of this solution?

Protect users data from sensitive decisions.

What are the goals of this solution?

Allow users to use electronic devices and guarantee that their sensitive data will fit their decision wherever their data will be transferred.

Synchronize data between Electronic devices

The main issue when it comes to synchronizing data between electronic devices is memory and data speed transfer. When it comes to synchronizing data it's a transaction process triggered by a human decision and automatically handled by machines. Two entities are thus included: humans and machines and the link is not atomic due to the process which is a transaction. It's obvious that a list of well known let's call them accidents may lead to the consequence that data sent is not data received and this is a problem when data is sensitive, when a user wants to synchronize his data, he needs to free his mind of these matters even if he works with a bunch of electronic devices.

There is a list of protocols used to synchronize data between electronic devices. This article will make a brief description of them because this is not the core of our paper.

The aim of this paper is to replace transaction processes with atomic ones which are QDT : Quantum decision theory.

What is QDT? QDT describes a decision maker's choice as a stochastic event occurring with a probability that is the sum of an objective utility factor and a subjective attraction factor. QDT offers a prediction for the average effect of subjectivity on decision makers, the quarter law.

We applied this theory in data synchronization between electronic devices since the synchronization decision is and will always be triggered by a human decision.

On the Internet, the highest data transfer speed reached today is 319 Terabytes per second. Researchers in Japan's National Institute of Information and Communications Technology recently set a new world record for the world's fastest internet speed at 319 Terabytes per second (Tbps). The long-haul transfer of data took place over 3,001 kilometers. A study presented at the International Conference on Fiber Optic Communication in June 2021 has revealed that Japanese engineers recently broke the world record for the highest internet speed, achieving a data transfer speed of 319 terabits per second (Tb/s).

In terms of protocoles to transfer data between electronic devices, MQTT protocol takes one message per second per publisher. In this way the number of publishers (and connections) also gives the global message throughput, for example 100.000 publishers produce 100.000 messages per second through 100.000 MQTT connections. Add to this, MQTT has been designed to save the battery of mobile devices as much as possible. MQTT

eISSN1303-5150

consumes 11 times less power to send messages and 170 times less to receive messages than HTTP. MQTT is also 93 times faster than the HTTP protocol. With MQTT, we have the possibility to set the QoS (Quality Of Service), that is to say that for each message sent we can choose how the broker should manage it.

QoS0: The message sent is not stored by the Broker. There is no acknowledgment of receipt. The message will be lost if the server or client is shut down. This is the default mode

QoS1: The message will be delivered at least once. The client returns the message until the broker sends an acknowledgment of receipt in return.

QoS2: The broker saves the message and will forward it until it is received by all connected subscribers.

Beside all of these, MQTT offers the persistence of messages on a Topic. Messages are kept on the Broker. It is also possible to manage the subscription/publication rights for each Topic. Also, it is possible to secure the transport of messages in SSL /TLS as well as by identification of the user (username and password), finally, topics and the tree structure are created on the fly. There is nothing to configure on the Broker. It is the "publisher" who creates the topics tree at the time of publication.

Another protocol is IP Internet Protocol which allows data to be sent between devices via the internet. And as we saw, 319 Terabytes is the Highest speed reached today using optical fiber.

As we see, the issue takes two parameters into consideration: memory and speed. The core of our paper is to avoid any massive transfer, it's not because the speed transfert on the Internet is gigantic means we will use it at its maximum. We will explain in the following paragraphs that the solution we propose to save data integrity is precisely by combining transfer speed and memory.

In the concrete application we made, we synchronized data between three microcontrollers which send random data such as temperature, humidity etc. We used sensors to get these values. Add to this, we used a laptop as a hub and the cloud to store data as a data warehouse, we lay on cloud computing to manage data.

Synchronize data between electronic devices.

Data management process:

We created an extension which measures the size in bytes of the message sent/received by electronic devices. We also created a TCP/IP packet generator.

The packets are sent in the cloud according to an atomic law by the IP protocol towards a central point which manages the data based on the rules of cloud computing.

To apply our solution we used three microcontrollers:

An Arduino UNO

An ESP8266

A Wemos D1 ESP8266

As a hub, we used a laptop equipped with wifi and USB port to be capable of communicating with the microcontrollers. Add to this we used a bunch of sensors to send/receive data such as temperature, humidity, etc which change depending on the sensor's location .

As a data warehouse, we used the cloud. We mixed Microsoft cloud and Google cloud to store data to take the benefits of both of them.

The sensors used provide random information such as temperature, humidity, etc. These values are formatted in a json array which is called a message, this message has the following format:

```
{
  “temperature” : 25.6,
  “humidity” : 78
}
```


eISSN1303-5150

The transition to the cloud must imperatively use the IP protocol and it is at this level that we apply QDT by regulating the transfer speed to adapt to the message following an atomic law (no massive transfer is allowed) because this is how data is corrupted. To send/receive data we lay on REST API.

Figure 1. Wemos D1 ESP8266 microcontroller.

Figure 2. Arduino UNO.

Figure 3. ESP8266.

To obtain a data value that continuously changes according to the location of the sensor, we chose for our experiment several sensors (humidity sensor, temperature sensor, gases sensors, etc.).

*e*ISSN1303-5150

Figure 4. MQ135 Gas sensor linked to ESP8266 Microcontroller.

The values published in the broker by the sensors follow a pattern that we will detail later. The goal is to circulate information on the electronic devices and check the integrity of the data even if you go through a bunch of electronic devices. For this, we use a simple LCD display to display the data on one of the microcontrollers.

Figure 5. LCD Display linked to ARDUINO UNO Microcontroller.

Finally, all electronic devices need to be powered, we have two types of power. By battery or by USB port depending on the microcontrollerFinally.

Figure 6. Battery power for the arduino microcontroller.

Figure 7. USB cable power for Wemos D1 and ESP8266 microcontrollers + serial cable to send and receive data to Arduino UNO.

The data published in the broker by the sensors are first conveyed to the broker.
The messages are generated based on the following schema:

***e*ISSN**1303-5150

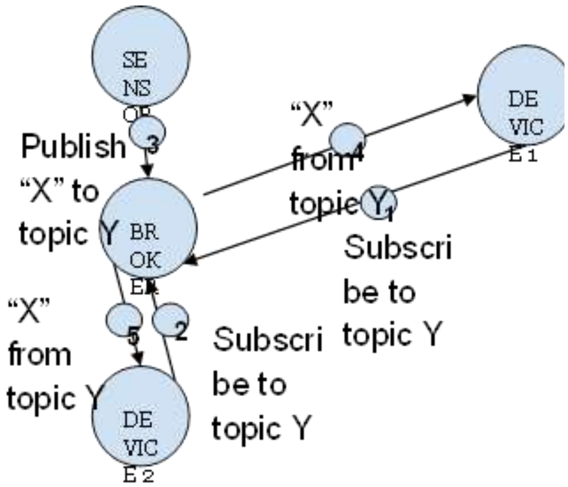

Figure 8. Generating messages process using sensors captured values.

As we explained earlier, the MQTT has a possibility of three quality of services. The following schema explain more clearly the difference between the three:

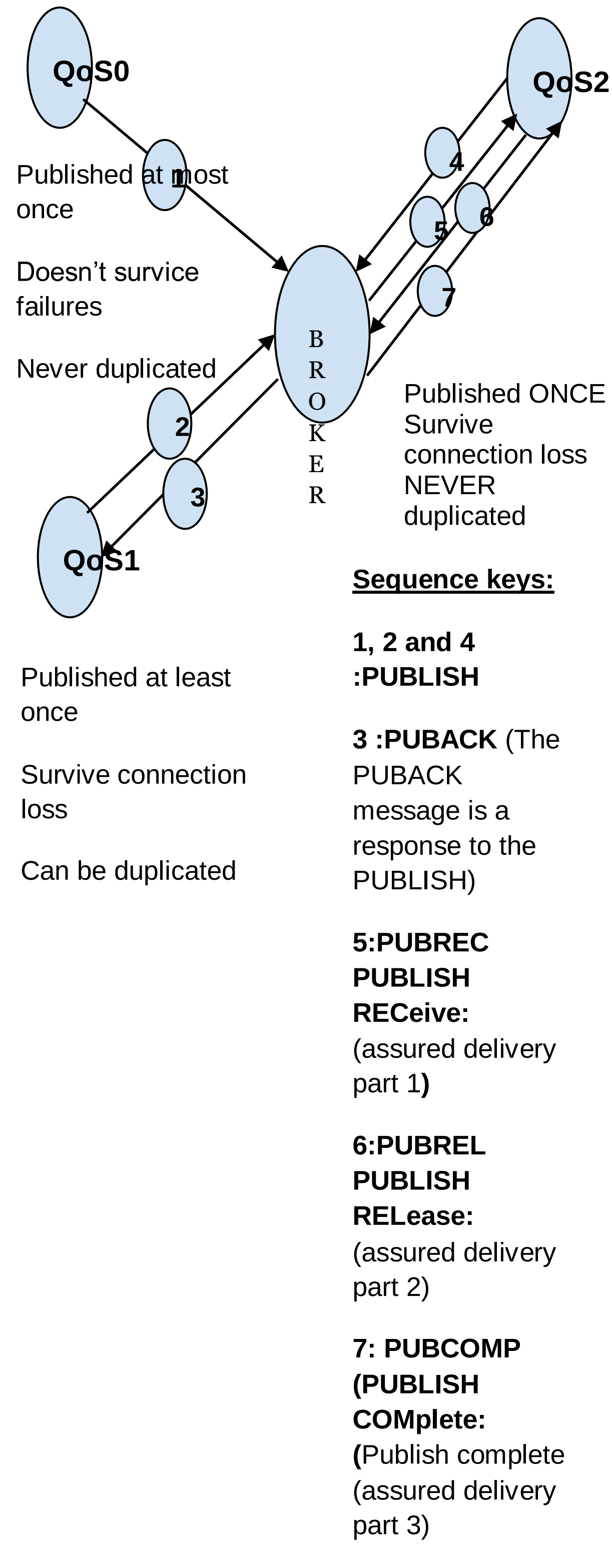

Figure 9. MQTT quality of services characteristics.

eISSN1303-5150

We based our work on QoS2 for its characteristics.

To keep the message updated, we need to have a central point to keep data well updated. In order to do that, data is centralized in a data warehouse in the cloud and the sensors are given input data regularly.

Here is our cloud computing strategy: The broker (server) is the central hub of our MQTT network. In our use case, the broker is local (based as a hub on our laptop).

To go from local to the cloud, we use a client/server strategy where the client is the used hub (broker) and the server is the data warehouse built on the cloud.

We lay on REST API to send JSON array to server.

It is a question of knowing the size of the message to be sent from the client to the server and to use REST API in an atomic way and no longer transactional.

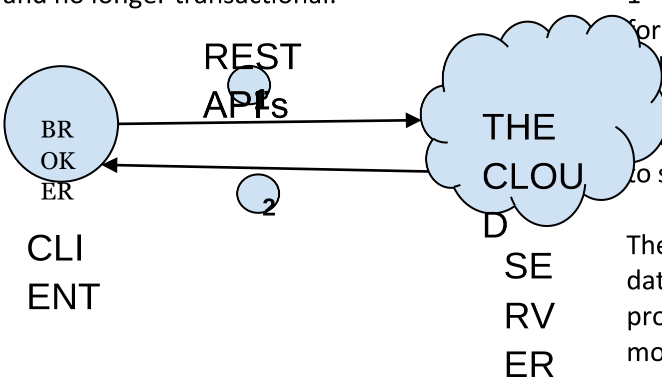

Figure 10. Client/server strategy used to send and receive data to the cloud.

RESULT OF APPLIED USE CASE

To test data integrity we follow the variationof data through the three electronic devices we used. We control three parameters (memory, transfer speed using IP protocol and transfer speed using MQTT protocol)

No massive transfer is allowed. The transfer speed of both MQTT and IP protocol must fit each other to follow an atomic law.

Study of a typical example

JSON:

```
{
        "temperature": 25.6,
   "humidity": 75%,
}
```

To send the message atomically, we must regulate the transfer speed between the MQTT protocol and the IP protocol.

The size of the message taken as example is measured using an extension we developed which gives the size of the message in byte. The transfer speed to the cloud using IP protocol is then calculated regarding this size.

We summarize the process in the following diagram:

1- Generating the message using the JSON format

Use our integrated extension to calculate size in bytes.

Use our calculator to fix IP transfer speed to send the message atomically.

The phase of updating and redistribution of data is done in a classic way, only the process of sending and receiving data is modified.

Use case and flow diagram

This paragraph is an overview of our method applied to a simple use case which is following the variation of gas value and a flow diagram to summarize the flow and the set of the dynamic relationships in our system.

First we need to subscribe our ARDUINO UNO, Wemos D1 and ESP8266 NodeMCU to the broker (remember, the broker in our use case is a hub installed locally on our laptop, this hub will be used as a client and the cloud will be considered the server, the strategy is well explained previously).

The MQ135 publish the CO_2 value in ppm (particle per million) to proper topic 'CarbonDioxide'

The Wemos D1 and the ESPRESSIF8266 have both a WI-FI network SSID and password. The Arduino UNO allows data to be transmitted using a serial cable.

The ID address or hostname of MQTT broker, the network port, the username and the corresponding password are kept to be used when the broker will turn from a hub to a client to send/receive data to the cloud.

We use MQTT-Explorer to explore our message queues, delete retained topics, and see what is happening on our broker. In fact, MQTT-Explorer allows the following:

- Visualize topics and topic activity
- Delete retained topics
- Search/filter topics
- Delete topic recursively
- Publish topics
- Plot numeric topics
- Keep a history of each topic

MQTT-explorer shows the whole picture of the message queue. It allows the integration of new services, IoT devices in a network. It subscribes to all topics on a MQTT-Server and displays a message queue hierarchy, allowing drill-down to the topics of interest.

Figure 11. A simple subscription to the topic CarbonDioxide using MQTT-explorer.

Figure 12. A simple visualization of the published message to the topic CarbonDioxide using MQTT-explorer.

To send/receive messages to the cloud, we intervene at the level of the transport layer of the ISO model, for reminder in networks, the so-called transport layer constitutes the fourth layer of the OSI model. This layer includes all the protocols responsible for error management and network flow control.

The protocols in use today in this layer for the Internet all originated in the development of TCP/IP. In the OSI model the transport layer is often referred to as Layer 4, or L4, while numbered layers are not used in TCP/IP. The best-known transport protocol of the Internet protocol suite is the Transmission Control Protocol (TCP).

To generate TCP traffic with the right port with the right delay between packets we propose a delay calculator to fix the number of ms between two packets.

Using a network connection, TCP allows a server and client pairs to exchange messages by using data segments packaged inside of data requests and responses.

TCP 3-way handshake:

1- SYN : Client Synchronization step

2- SYN/ACK: Server synchronization step

3-ACK: Client responds to the server that it recognizes.

Hping3 is a terminal application for Linux that will allow us to easily analyze and assemble TCP / IP packets. Unlike a conventional ping that is used to send ICMP packets, this application allows the sending of TCP, UDP and RAW-IP packets. Along with the analysis of packets, this application can also be used for other security purposes, for example, to test the effectiveness of a firewall through different protocols, the detection of suspicious or modified packets, and even protection against attacks. DoS of a system or a Firewall.

QoS2 is the level of service used to apply our approach in this simple use case. It is the highest level of service in MQTT. This

*e*ISSN1303-5150
www.neuroquantology.com

level guarantees that each message is received only once by the intended recipients and that the message is never duplicated. Thus, QoS2 is the safest and lowest quality of service level.

We offer our own traffic signaling system to control the entire process of sending and receiving data. As a reminder, the packet goes from a simple packet under MQTT protocol to a TCP packet. The ports included in our sending and receiving system include a security part managed by CA certificate. The ports used are port 1883 for the MQTT protocol and port 443 for the TCP/IP protocol. Keep it safe for data integrity from A to Z as Amazon says.

DISCUSSION AND CONCLUSIONS

As we see, our project is a Industry 4.0 project, it lays on its four component:
CPS: Cyber physical Systems
IoT : Internet Of Thing
Cloud Computing
Big Data
The core of this digitalization is very significant and has an enormous benefit to guarantee data integrity. We apply atomic law using the QDT theory to transfer data between used entities. This technique can be applied to many other use cases.
Data used can also be generic as long its integrity is protected.
The promoted technology is generic, it only requires a selection of used variables. Future work will focus on applying this technique on sensitive data in order to guarantee to the user that he can use as many devices as he wants and that his devices are completely synchronous and guarantee his data integrity.
In conclusion, the QDT theory demonstrates through its many use cases that access to information resources is not what guarantees the integrity of data but rather the decision as to its use.

REFERENCES

[1] A. Brandenburger, P. La Mura, «Team decision problems with classical and quantum signals» arxiv:1107.0237v3 [quant-ph] 22 Jan 2015.
[2] M. Biswajeeban, A. Kertesz, «The use of MQTT in M2M and IoT Systems: A survey» IEEE Access November, 2020.
[3] Guo, Le Jiang, Feng Zheng, Ya Hui Hu, Lei Xiao et Liang Liu. « Analysis and Research of Cloud Computing Data Center ». Applied Mechanics and Materials 427-429 (septembre 2013): 2184–87.
[4] Aouat, Asmaa, El Abbassia Deba, Abou El Hassan Benyamina et Djilali Benhamamouch. « Deployment in Cloud Computing ». International Journal of Distributed Systems and Technologies 11, no 1 (janvier 2020): 27–37
[5] Rylander, Andrea. « Cloud computing: Knowledge expectation and reality : Cloud computing: Knowledge expectation and reality ». Student thesis, Uppsala universitet, Institutionen för informationsteknologi, 2017.
[6] Datla, Dinesh. « Wireless Distributed Computing in Cloud Computing Networks ». Diss., Virginia Tech, 2013.
[7] Bielik, Valerii, YuriyMorozov, and Mykola Morozov. "Sensors in Cyber-Physical Systems Based on Android Operating System." Advances in Cyber-Physical Systems 6, no. 2 (December 2021): 83–89.
[8] Grebeniak, Andrii, Eugene Miyushkovych, and Yaroslav Paramud. "Digital Interfaces in Cyber-Physical Systems." Advances in Cyber-Physical Systems 2, no. 1 (March 2017): 6–10.
[9] Mykyychuk, Mykola, SvyatoslavYatsyshyn, Bohdan Stadnyk, and Yaroslav Lutsyk. "Smart

eISSN1303-5150

Measuring Instruments for Cyber-Physical Systems." Advances in Cyber-Physical Systems 1, no. 1 (February 2016): 7–15.

[10] Rawat, Danda B., and Chandra Bajracharya. Vehicular Cyber Physical Systems. Cham: Springer International Publishing, 2017.

[11] Greer, Christopher, Martin Burns, David Wollman, and Edward Griffor. Cyber-physical systems and internet of things. Gaithersburg, MD: National Institute of Standards and Technology, March 2019.

[12] Sharma, Nonita, L. K. Awasthi, Monika Mangla, K. P. Sharma, and Rohit Kumar. Cyber-Physical Systems. Boca Raton: Chapman and Hall/CRC, 2022.

[13] G. Nain, F. Fouquet, B. Morin, O. Barais, J-M. Jezequel « Integrating IoT and IoS woth a component-based approach » Published in the 36th EUROMICRO Conference on software engineering and advanced applications IEEE October 2010

[14] Sanchez, David, OswaldoSolarte, Victor Bucheli et Hugo Ordonez. « Evaluating The Scalability of Big Data Frameworks ». Scalable Computing: Practice and Experience 19, no 3 (septembre 2018)

[15] Phoomikiattisak, Ditchaphong, etSaleem N. Bhatti. « End-To-End Mobility for the Internet Using ILNP ». Wireless Communications and Mobile Computing 2019 (avril 2019): 1–29.

[16] Zhen, Xiao Qiong, Zhen Dong Yao et Wen Bin Wang. « Scheme Design for Communication between Network Port and Serial Port Based on ARM ». Advanced Engineering Forum 6-7 (septembre 2012): 941–47.

[17] Lobzhanidze, AleksandreZengWenjun. "Building hybrid multicast by combining IP and application layers". Electronic books. Electronic dissertations, Columbia, Mo. : University of Missouri--Columbia, 2007

[18] Yara, Ahmad. « Preventing Vulnerabilities and MitigatingAttacks on the MQTT Protocol ». Student thesis, Örebro universitet, Institutionen för naturvetenskap och teknik, 2020.

[19] Fakhraddin, Haider. « Toward IoT : Implementation of WSN based MQTT Data Protocol ». Student thesis, Linnéuniversitetet, Institutionen för datavetenskap och medieteknik (DM), 2019.

[20] Rodríguez, Alejandro, Lars Michael Kristensen et Adrian Rutle. « Formal Modelling and Incremental Verification of the MQTT IoT Protocol ». Dans Transactions on Petri Nets and Other Models of Concurrency XIV, 126–45. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019.

[21] V. Lampkin, W. Tat Leong and al Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry. ISBN-10: 0738437085 ISBN-13: 9780738437088