Generic Architecture of an IoT Digital Twin

Tony de Souza-Daw
School of Business, Engineering and Construction
Melbourne Polytechnic
Preston
tonydesouza-daw@melbournepolytechnic.edu.au

***Abstract*— Recent developments in the digital realm have created a need for a consistent, global approach to represent parameters of an object or thing in the real world, known as the physical object digital's twin. This paper proposes an architecture for digital twins that is compatible with current technologies, standards and support customs developments and owners' parameters, security and privacy considerations. The design uses current, scalable technologies as it supports digital twins' parameters and information.**

Keywords—Digital Twin, DOI, RAIN RFID, IoT

I. Introduction

Digital representation of real-world objects has been developed and solved for a number of different applications such as [1], [2], [3], etc. IoT devices are highly individual, application-specific, developed by small and large businesses and may or may not be networked or accessible over the Internet, accessible all the time, reliable and may use non-standard communication protocols. Many Digital Twins' characteristics have been captured in these surveys [4], [5], [6]. The highly flexible nature of IoT devices with many developers and non-standard nature of IoT makes uniform digital representation difficult to achieve.

II. Background

Digital twins representation has been achieved for purpose-built applications, some are described below:

Domain Name System – solves the problem of translating human-friendly names to IP addresses e.g. myserver.mydomain to 192.168.20.3 [7]

IP addresses – enable communication over a network e.g. home computers to a web application server on the other side of the world [8].

MAC – media access control address, enables unique identification on layer 2 of the OSI model communication [9].

DOI – Digital Object Identifier System, maps objects (movies, music, documents, etc) to numbers for inter-operable international identifiers [10].

GS1 – GTIN – Global Trade Item Number, including UPC barcode is found on product with 8, 12, 13 or 14 digits lengths [11].

GLN – Global Location Number (13 digits, identifies location stores, manufacturing centers, warehouse, corporate headquarters)

SSCC – Serial Shipping Container Code

EPC Tag Data – Electronic Product Code, Gen 2 RFID user memory data, control info, tag manufacturer.

ISO/IEC 15459 – license plates in barcodes

ISO/IEC 15961-2 Data Constructs Register includes packet objects, tag data profiles and mapping tables [12].

MIB – Management Information Based, a tree like structure to hold configuration parameters about a device such as in SNMP or windows registry [13].

Others barcodes [14] include EAN-13, UPC-A, EAN-8, ITF-14, ISBN, ISSN, Code 39, Code 128, Code 11, GS1-128 (UCC-128/EAN-128), SSCC, Codebar and QR Code or other technologies like RFID such as RAIN [15].

III. Applications of Digital Twins

There are many applications of digital twins; from having a digital representation of significant sites such as base towers in telecommunication, oil & gas industry plants and equipment, power generation industry equipment to maps and landscapes for the survey & construction industry. A digital twin of high-end, complex, expensive equipment may maintain a record of engineering materials such as specifications, configurations, part list and operating capacities; historical information such as maintenance, procedures and tests; and operating parameters such as current input and output data (volume, speed and pressure), consumables, emissions constrain, performance and benchmarks. Some clear advantages that digital twins can offer, for example, operation analytics to help determine preventive maintenance and reduce the over reliance on highly skilled staff performing at maximum performance all the time. In addition, predictive analytics to feed into visualizer, models of normal, ad-normal and disaster studies to create improvement, investment and development strategies.

There is a significant push to transfer everyday items such as personal items, travel bags, artwork, computers and wearable devices to have a digital twin of its own. In order to support the research and development into the devices, such as where and how much the device is used and thus determine where to advertise to new customers and personal advertisement of upgrade or upsell accessories. Digital twins advantage also includes the authenticity in a blockchain or having insurance contact information in case of recovery of a crime, or proof of ownership and ownership history for famous memorabilia such as a cricket hat worn by a famous player or an artwork painted by a famous artist centuries ago.

Futuristic applications include an object or service making transactions with the owners, custodians or users and the object or service subsequently making transactions with a business. For example, the owner of an airport luggage can renew its insurance or increase its capacity limit with the luggage, the luggage in turn, organize the payment to an airline. The luggage could determine if it is lost via its own GPS location and keep an accurate real-time location. Also, a warning beacon to insurance or police if it believes it is being stolen, ended up at the wrong address which the owner has not approved.

IV. Risks of Digital Twins

Creating and using digital twins comes with its own security risks. An owner may want to share information about an object with other devices or people to help find, fix and

upgrade however this information should not fall into the hands of a serial killer, stalker or blackmailer. Another example, someone selling their laptop second hand wants to make sure the laptop is securely wiped and the original owner's identity (and inherent responsibilities) removed. Moreover, a digital twin creates another attack surface that cyber criminals may use to exploit vulnerabilities for ransom, espionage or terrorism.

Some serious concerns about connecting information particularly, if it has never been connected to the Internet before and it allows parameters to be configured. Any universal design of digital twins must take security and privacy.

V. Digital Twin Universal Design Considerations

A table of design considerations are described in Table 1.

Table 1 Digital Twin Design Considerations

Q. Should the Digital Twin Universal Design directly use other protocols (e.g. such as TCP/IP model)? No. Although may be easier to translate initially, fixed to a third-party policy creates a dependency on third-party that needs to be consistently upgraded. Hence, an IoT digital twin architecture should not directly be dependent on other protocols as it is best to remain independent and not fixed to a protocol. Avoiding complications as seen with TCP/IP in comparison to OSI.
Q. Should the Digital Twin Universal Design have predefined values? No. Arguments to have predefined parameters such as name, protocols, etc have inherent complications as too many IoT protocols/devices are non-standard.
Q. Should the Digital Twin Universal Design have the ability to update configuration values? Yes Parameters should be able to be updated in real-time, not just static information.
Q. Should the Digital Twin Universal Design have inherited security? Yes. Blockchain, Digital Signatures and Certificates should all be used with IoT devices. Without inherent security, the Digital Twin should only transmit data like a satellite continuously transmitting data.
Q. How customers can use it after they brought it? Flexible. Customers should be able to change some configurations such as removal of personally identifiable data, but not change manufacturer specifications or model numbers.

The design considerations have already been solved in many different implementations as those mentioned in the background. Hence, using existing technology is preferred and will increase the chance of adoption.

VI. Digital Twin Universal Proposal

The proposal of taking a physical object or service and digitize it into a Digital Twin architecture is shown in Figure 1.

Access through a web browser can bring up digital properties of digitalize objects. These objects are hosted on web servers and accessible via an XML file that is being updated by an IoT server elsewhere. The IoT server communicates with IoT devices to update parameters such as sensors, audio, video, etc. The broker communicates with the IoT device and the network protocol (e.g. IPv4 or IPv6). The IoT Server allows access to historical transactions and data by storing and updating a blockchain.

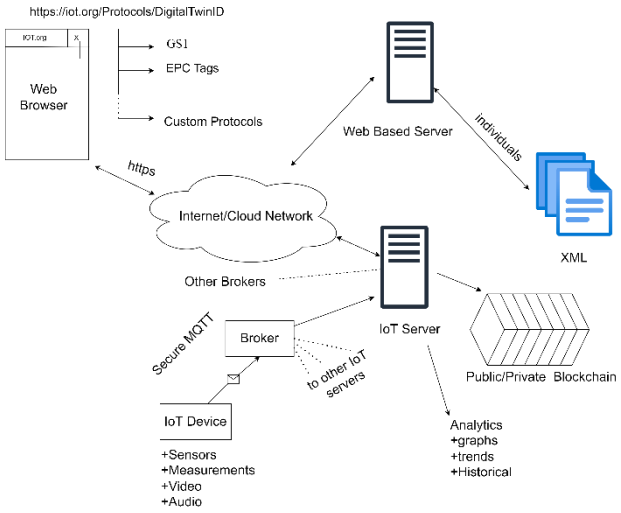

Figure 1 Digital Twin Architecture Proposal

Each section of the Digital Twin Architecture is described hereafter in more detail.

A. Web Browser Access

Web Browser can read http/https and many can display XML. This enables the display of XML features in addition to web browsers plugins to automate or entire tools to be built on top of web browsers. This is a very powerful feature, that enables customs and user-specific features to view and possibly control IoT devices through web browsers. Web browsers can also have applications and macro built on top to enable greater functionality of IoT devices.

Similar to DOI the *iot.org* domain will be indexed through the Domain Name System (DNS). DNS records can be requested from the iot.org server which forwards the appropriate records of the iot.org location of the files. The requester now has the location and can request the desired XML file. Any hijacking or masquerading of servers should be detected and prevented at the DNS trusted certificate authority to prevent IoT requests to a malicious server.

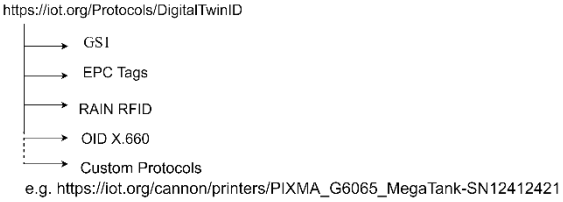

Figure 2 IoT Digital Twin Naming Convention

The domain iot.org will be the generic domain as shown in Figure 2, identifying that the universal resource is an internet of things resource. The existing protocols such as GS1, EPC tags, OID X.660 can be identified using the slash notation. E.g. https://iot.org/UPC/ 036000291452 will bring up *Oojami - Time Is Now (Music CD)* description and attributes related to that barcode. As performed by the website https://www.barcodelookup.com/036000291452.

Similar to the GS1 notation of website slash product code as described [11] and the GS1 URL [16]; the slash notation can be used in a custom protocol such as …/manufacturers/product line/device-Serial Number. This enables the manufacturer to create and expand their own custom directory notation. The sub-folders enable sorting at the manufacturer's discretion. Hazard information, datasheet and product information can all be identified in the manufacturer part of the IoT XML file can be in the forms or URL links or hosted in the XML file itself. It is envisaged that critical safety information is kept on the digital twin's XML file and product information and user manuals are linked to the manufacturer's websites. Individual items (an instance) such as serial numbers are separated with hyphens to access an instance of the product, whereas without the hyphen will bring up the descriptors from the manufacturer. Similarly, the IoT resource can be hosted locally on a local IoT domain for example in-house manufacturing or remotely over the Internet once an instance of a product is in the wild.

An object that uses more than one protocol can and should be resolved on both protocols. For example, an object that has a SSCC and GS1-128 will be resolved independently on their protocols and their attributes on that particular barcode will be show at the manufacturer level descriptors, generic descriptors for their product.

iot.org/GS1-128/(01)95012345678903(3103)00123

iot.org/SSCC/(00)104467001282329809

An instance of a device with a serial number will need to be located before its instance can be accessible. Once an instance of the device is located, the individual instance private attribute may identify that it also has parameters for other protocols and is able to access those parameters through the IoT instance.

Repeating, the slash notation indicates the number system employed, for example, a 96-bit binary string, converted to decimal for the purposes or accessing its digital twin over an intranet.

B. Creation and Destruction

The aforementioned described how manufacturers and product owners can create IoT based digital twin, which has all the generic attributes and fields at the date of manufacturing. Such as barcodes IoT slashes indicate a product line. Batch numbers and used by dates are done independently to barcodes. New products will require a new IoT product code, end of life products IoT codes will be discontinued after 6 months. Enabling the manufacturer template to create an XML file 6 months after the end of life has been determined. Although, their data still exists on an instance digital twin as per the storage plan. Hence, there is no need to consider how a product is created or destroyed on the manufacturer's side. However, once a product is distributed, brought or sold, then an instance of a product digital twin can be created. An instance can be created at a transaction point (distribution, brought, sold). Apart from the sale, the manufacturer will present the new owner with details on how to create an instance. An instance will need to use the manufacturer product key (access to the manufacturer's product digital twin template) and the product IoT code to produce an instance, a pseudorandom generator code, salted by the manufacturer product code and manufacturer's public key to produce a receiptID and issued by an IoT authority see Figure 3. The new owner will need to create an IoT account. As part of a creation account, the owner will need to create an IoT asymmetric public and private key see Figure 4. Use your private keys and the receiptID and the manufacturer public key to create an instance of the product. The receiptID can only be used once and expired. Hence, the owner can not create more than one digital twin of the same product (which could eventually result in insurance fraud, etc). If a receiptID was created inadvertently, (customers change their mind in the middle of a transactions, system crashes, etc) the receiptID is tagged as used and can not be used to create a digital twin. Only successful transactions leading to good receiptID can be used to create a digital twin. Items that are returned to the place of purchase for a refund must determine whether a digital twin instance exists by simply querying the IoT code. If a digital twin instance has been created, the new owner field must be changed to the store (the new owner) or probably the easier option is to reset to faculty settings (remove private components). The new owner can relinquish ownership over an item, as it may be passed to a new owner or discarded. If passing to the new owner, the original owner can remove all of their personal fields (eg. Reset to faculty settings) or change them to the new owner via a receiptID process. Resetting to faculty default would require a new receiptID to be used to issue the new owner. Transferring ownership requires a receiptID, as the end customer may have an ownership account but not a merchant account (can't sell), hence the store may provide a change of ownership through its stores for up to a fixed period of time (e.g. 3 years). For example, a high-end jewellery store may issue a receiptID for buying an engagement ring, giving it to your future bride and issuing another receiptID to transfer ownership after marriage.

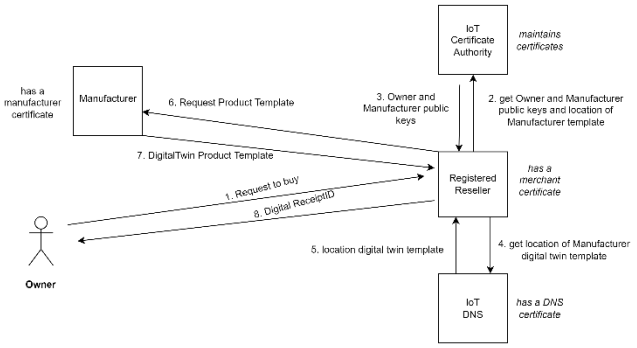

Figure 3 Purchasing a Product

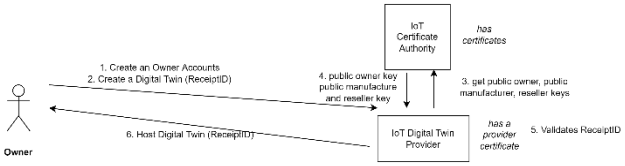

Figure 4 Creating a Digital Twin

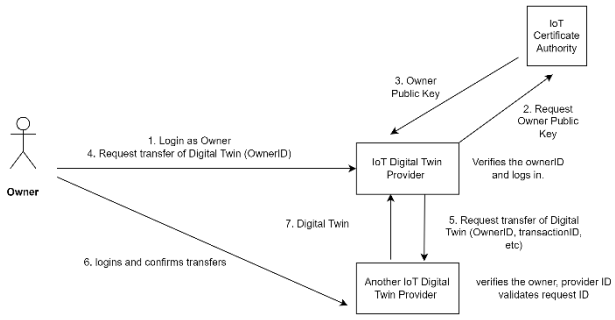

Figure 5 Transferring Digital Twin Providers

An owner can transfer its digital twin hosting provider to another hosting provider by creating an account on the second provider and requesting a transfer, which the owner needs to login to the first provider to accept the transfer as per Figure 5

C. Resolving Digital Twin ID

As previously discussed one digital twin can resolve to multiple protocols. Access the same digital twin via two protocols such as:

iot.org/EAN/(01)95012345678903(3103)00123-SN12341234

iot.org/UPC-A/9501234567890-SN56785678

In this case, the same product is sold with EAN and UPC for different parts of the world and can access the same digital twin using the above. A symbolic link (soft link) is created to point to an existing digital twin if a digital twin instance has already been created. In general, an instance would only be created for either EAN or UPC as it is unlikely a product would be brought to one country and then transfer and brought again to a country that does not recognize the original standard. Symbolic links are stored and distributed by the IoT DNS authority. Simply another entry is added to the IoT table with a different IoT code to point to the same URL.

Using unique codes such as serial numbers should not allow duplicate IoT codes to be created. A search of the new IoT code should be conducted before issuing on as part of the creation process to prevent this error from occurring. However, if there are duplicates, IoT codes can be combined. The code with the higher IoT domains is considered the original one with the duplicate being placed in the components part of the XML file (not deleted). The duplicate link now is redirected to the original IoT code. This requires both the owners are the same or in an agreement.

D. Storage Plan

The storage of an IoT digital twin, as the twin is an XML file with data embedded, some twin will be kilobytes other twins with audio-visual will be gigabytes. Similar to current hosts of web pages, the cost will be based on storage and time duration.

Web Servers can be hosted locally or externally. Hosted locally, internal would keep IoT transaction costs to a minimum. IoT code from a registered publisher may charge a fee for every time the IoT digital twin URL relocates as the IoT register publisher needs to keep the URL/IoT database up-to-date.

Blockchain storage keeps the transactions of data being updated by the IoT device through a broker and an IoT Server. The Blockchain would be funded by a hybrid or private IoT publishers. Records of globally created events must be stored on the blockchain. There are two fundamental deployments that need blockchain: (1) The creation, transactions and destruction of goods that are worth significant amounts including artwork, memorabilia, etc. (2) IoT devices where the data generated is worth more than the device, for example, weather sensors that combine measurements are subsequently sold to other countries to make their weather models work.

The first application should use blockchain as a method of validating and recording ownerships and transactions, consortium companies that make up the public trusted blockchain would include insurance companies, data host providers, high-end retail stores, high-end manufacturers, wholesalers and transport companies.

The second applications require the storage of data generated by a device. Placing data that is generated very quickly on a blockchain network will become very expensive very quickly, especially for video networks. This blockchain should be stored more locally in a historical blockchain. As the blockchain increase almost beyond the storage capacity, a hash is taken off the whole chain, and only the previous and current block is kept while the chain is hashed and stamped into the current block as the blockchain rebuilds with new data. The consortium that makes up this private blockchain would be the owner, data providers, suppliers and customers.

E. XML File

Instead of downloading a document or file such as the DOI it displays the current parameters in the form of an XML. The protected part of the XML tree structure is defined by the manufacturer of the IoT. The protected properties cannot change and are inherited by the manufacturer's product as shown in Figure 6. Owners of an instance of a product may change the public and private properties, by adding more properties or changing their values. The XML digital twin stores the current values. Previous values are stored on a private or public blockchain.

The protected attributes could include multiple standards such as OID X.660 or EPC tags and the digital twin XML should contain them all (if they exist), completed at the time of creation. Other attributes that should not change such as the serial number should also be located in the protected area. The three main roots of protected, public and private are all created when the digital twin is created. The manufacturer completes the protected part, including any barcodes (GS1) information, serial numbers, licenses or even a user guide that is immutable and protected against any change. Their public variables should be in a standard format such as those distributed by GS1 [17]. The public root contains what an owner or user like to add or remove such as the owner's identification in case the item is lost. Public keys for exchanging data or communicating with other devices.

The private root contains information only the owner can see. This may contain personal information such as passport ID, the hash of the private key, session keys for decrypting communication with asymmetric encryption with the public key, etc. Other private information like secret questions and answers for resetting the private keys and descriptions such as "home address" or links with email accounts and one-time pads to reset or enable accounts usage.

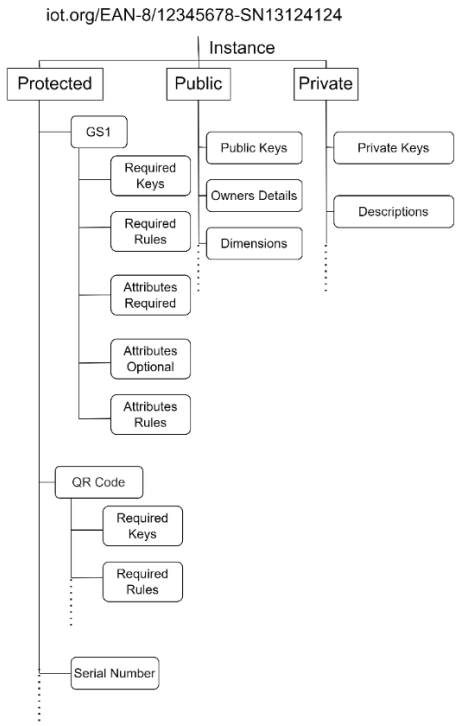

Figure 6 XML Tree Structure of a Digital Twin

Accessing the private root variables is only through the web server using a multiple factor authentication such as mobile phone SMS or physical token and a password.

F. Sharing

At the time of creation, a digital twin owner data can be pre-filled from the owner's account or left blank for anonymity. The IoT devices use the MQTT standard to transmit and share data as depicted in Figure 7.

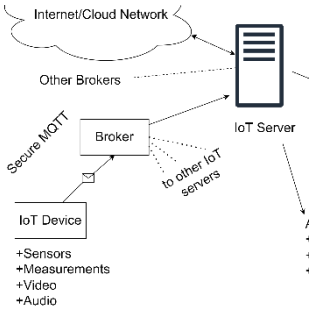

Figure 7 MQTT IoT Data Acquisitions and Messaging

The IoT devices collect data from numerous sensors and send that data to a broker, who then distributes that data to subscribers. There can be multiple brokers and multiple subscribers. An IoT device could be a subscriber as well as a collector/generator of data. To connect to the outside world an IoT server needs to be implemented. The IoT server connects the IoT network to the IP network the IoT server keeps the Web Server that hosts the digital twin up-to-date with the current state of the digital twin. The IoT Server can also perform analytics, graph trends, etc. The broker can act as a locator time and location stamp the data to the IoT Server, so the whereabouts of the IoT device are known.

IoT devices like sensors can generate minimal data, but audio-visual devices can generate large quantities of data. The data is buffered at the IoT device and broker but stored on the IoT Server and the database blockchain, backend. Storage is agreed upon when registering an IoT device to the IoT Server. The broker informs the IoT Server an unknown IoT device is trying to register. The IoT Server administrator can permit (automatically depending on the type of IoT) to the server. At the point of registration, the IoT server then determines a data plan.

The broker sends data to the IoT Server to update the current state, by either putting all the states on the blockchain, or only the changes (incremental) since the last the full state has been stored. IoT Server has a billing system to provide data storage and bandwidth allowance. IoT Server has a traffic shaper that limits an IoT broker's bandwidth to the IoT Server and storage capabilities of IoT data history on the server. The data history is paid via the maximum amount, after that it will start to override the previous data. The blockchain database is optional. Data stored on the blockchain is agreed upon via the blockchain network. (For example, only IoT devices that pay a premium would have their data stored on the blockchain).

Real-time mission critical analysis and automation should deploy on the IoT server. For real-time analysis, the IoT Server should also host the broker, such that data is collected and ciphered as soon as it arrives without transmitting to another party. This deployment strategy still allows for containerization but with minimum overhead.

G. Subcomponents

Any device nowadays comprises of several key components. A digital twin as a representation of an object is likely to be manufactured from components. For example, a car will have an engine, windows, doors, wheels, oils, etc. Individually, these parts that make up the car and then consumables such as petrol, oil/gas, filters, etc could also have digital twins. The digital twin current parts can be added to its XML file. For example, Figure 8 shows a headlight instance that has its own digital twin and a wheel that points to a manufacturer template. An actual instance of a part inside another part could be for smart technologies, such as a window that now has light sensors, rain sensors, augmented speed and maps, infrared, heater, shade band and night vision to automatically control the car's headlights, windshield wipers, enable see in the dark and proximity sensor or reflect infra-red heat.

```
<Car>
        <Public>
        <Parts>
                <Headlight> iot.org/jaylec/headlights/12342PR-SN123412
                </Headlight>
                <wheel>
                 iot.org/Advanti/Wheels/SA15
                </wheel>
       </Parts>
       ....
        </Public>
      ....
</Car>
```

Figure 8 Digital Twin subcomponents

H. *Travelling Suitcase Scenario*

A common scenario pose around IoT visionary is the traveling suitcase; a suitcase boards an airplane and reports scan in and out of arrival and dispatch as it transits through airports. If the suitcase ended up missing a flight or boarding the wrong flight the digital twin when updated could alert the owner. Figure 9 shows a diagrammatical view of the traveling suitcase scenario.

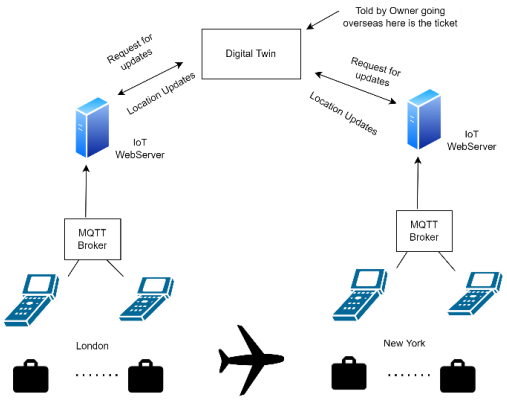

Figure 9 Travelling Suitcase Example

In this example, the suitcase is scanned in when arriving at an airport and scanned out at dispatch when boarding a plane. The scanner notifies the MQTT broker, in turn, reports the updated IoT server with the suitcase ID (e.g. Shipping ID) which then reports it to the digital twin over the Internet. However, firstly, the digital twin needs to tell which IoT web server to monitor for its ID such that the server can setup a trigger event for it. The digital twin knows which server IoT web server to update as it has been told by the owner.

If the suitcase arrives at the wrong airport, it is not expected to be at, the scanner notifies the MQTT, in turn, notifies the IoT web server. The webserver does not have any triggers for the event and places it in a queue to check for the existence of a digital twin and to update if found. In this case, the IoT website, if found a digital twin using an IoT DNS name (iot.org/…/modelNo-serialNo) can then notify the digital twin instance. If no digital twin is found, the notification is dropped.

I. *Implementation Roll Out Strategy*

Implementation of a mass implication technology such as this can be complex. The first course of action is to purchase the iot.org domain and partner with a large manufacturer and identifier provider such as Unilever and GS1. The manufacturer can provide the digital twin manufacturer details and hence start to populate the iot.org/…/… with digital twins. From there, instances can be created at the manufacturers and shipped to the destination and the digital twin is populated supporting insurance and preventing fraud. Hopefully, from there more manufacturers will create IoT.org digital twins and manufacturer enabling customers the transferal of ownership. The final stage will allow retailers to create receiptIDs for customers to create Digital Twins of their own internet of things.

VII. CONCLUSION

The world of Internet of Things has created the connected digital twins. The proposal showed how digital twins can be globally connected and their resources accessible which enables ownership, tracking and reporting.

References

[1] F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S. C.-Y. Lu and A. Y. C. Nee, "Digital twin-driven product design framework," *International Journal of Production Research,* vol. 57, no. 12, pp. 3935-3953, 2018.

[2] J. Wang, L. Ye, R. X. Gaop, C. Li and L. Zhang, "Digital Twin for rotating machinery fault diagnosis in smart manufacturing," *International Journal of Production Research,* vol. 57, no. 12, pp. 3920-3934, 2019.

[3] R. Söderberg, K. Wärmefjord, J. S. Carlson and L. Lindkvist, "Toward a Digital Twin for real-time geometry assurance in individualized production," *Toward a Digital Twin for real-time geometry assurance in individualized production,,* vol. 66, no. 1, pp. 137-140, CIRP Annals.

[4] K. Lim, P. Zheng and C. Chen, "A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives.," *J Intell Manuf,* vol. 31, p. 1313–1337, 2020.

[5] Y. Wu, K. Zhang and Y. Zhang, "Digital Twin Networks: A Survey," *IEEE Internet of Things Journal,* vol. 8, no. 18, pp. 13789-13804, 2021.

[6] B. R. Barricelli, E. Casiraghi and D. Fogli, "A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications," *IEEE Access,* vol. 7, no. doi: 10.1109/ACCESS.2019.2953499., pp. 167653-167671, 2019.

[7] I. E. T. Force, "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework," 8 2010. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc5890. [Accessed May 2022].

[8] I. S. Institute, "RFC: 791 Internet Protocol Darpa Internet Program Protocol Specification," Sept 1981. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc791.

[9] I. S. A. IEEE., ""Guidelines for Use of Extended Unique Identifier (EUI), Organizationally Unique Identifier (OUI), and Company ID (CID)" (PDF).," 2017. [Online]. Available: https://standards.ieee.org/wp-content/uploads/import/documents/tutorials/eui.pdf. [Accessed May 2022].

[10] ISO, "ISO 26324:2012(en)," 2012. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:26324:ed-1:v1:en. [Accessed May 2022].

[11] G. AISBL, "GS1 General Specifications," 1 2022. [Online]. Available: https://www.gs1.org/genspecs. [Accessed May 2022].

[12] I. J. 1. 31, "ISO/IEC 15961-2:2019," 2019. [Online]. Available: https://www.iso.org/standard/43631.html. [Accessed May 2022].

[13] IEEE, "IEEE8021-PAE-MIB DEFINITIONS," 2001. [Online]. Available: https://www.ieee802.org/1/files/public/MIBs/IEEE8021-PAE-MIB-200101160000Z.mib. [Accessed May 2022].

[14] B. Australia, "Barcode1 Australia," Barcode1 Australia, 2022. [Online]. Available: https://barcode1.com.au/types-of-barcodes/. [Accessed May 2022].

[15] R. RFID, "Rain RFID Homepage," [Online]. Available: rainrfid.org/. [Accessed May 2022].

[16] GS1, "GS1 Digital Link Standard: URI Syntax," 2022. [Online]. Available: https://www.gs1.org/docs/Digital-Link/GS1_Digital_Link_Standard_URI_Syntax_r_i1-2-1_2022-02-08.pdf. [Accessed May 2022].

[17] GS1, "GS1 XML," March 2022. [Online]. Available: https://www.gs1.org/standards/edi/gs1-xml. [Accessed May 2022].