# A Secure Interoperable API Wrapper Tunnel for Integration of GIS-Based ICS-IIoT and Digital Twins in Industry 4.0 Clouds

HASAN TARIQ[1], SHAFAQ SULTAN[2]
[1]Department of Electrical Engineering, College of Engineering,
Qatar University, Doha, QATAR
[2]Faculty Of Education, Allama Iqbal Open University
Islamabad, PAKISTAN

*Abstract:* - Industry 4.0 is the most disruptive revolution engulfing all the digital transformation from sensors nodes to cloud applications. A sustainable and secure Industry 4.0 ecosystem is a core challenge for stakeholders and state agencies addressed in this work by layered API tunnel interoperability instead of conventional cybersecurity methods and techniques. In this work, an interoperable application programmer interface (API) wrapper is being proposed to manage heterogeneous cloud architectures using Eucalyptus, Apache CloudStack, and OpenNebula. A comprehensive design and implementation plan from things layer to user interface has been accomplished in this work. Infrastructure as a Service (IaaS) is being automated using Ansible from Web Servers Layers through Common Gateway Interface (CGI) scripting. The Web Servers are being managed using WebSockets by micro–Web Servers Gateway Interface (uWSGI) from cloudstack managers (CSM) API. The Platform as a Service (PaaS) for big data, web servers, and machine learning platforms is accomplished using uWSGI from CSM. The public and private clouds are being managed using individual APIs. The wrapper, proposed in this work enables the big challenge of Industry 4.0 for realizing interoperability, scalability, heterogeneity, and automated cloud integration by functioning at the Software as a Service (SaaS) Layer.

*Key-Words:* - Industry 4.0, GIS, Digital Twins, CloudStack, IIoT, SCADA, uWSGI, IaaS, PaaS, SaaS, IoT, and Wrapper

Received: July 26, 2021. Revised: February 22, 2022. Accepted: March 27, 2022. Published: April 21, 2022.

## 1 Introduction

In secure urban scale, cloud managed services [1] and applications interoperability, scalability, heterogeneity, and automated cloud integration have always been a dreamed opulence. Urban applications [2] and processes challenges resulted in many systems, architectures, platforms, frameworks, and mechanisms. The canvas of environmental monitoring, infrastructure resilience, energy optimization, transportation routing, and health ubiquity can only be addressable by the Internet of Everything (IoE) in smart city level tasks [2-4].

By the start of 2020, the number of connected machines is estimated to reach more than 34 billion [3-4] more than fourfolds of the entire global population. The journey [5] from the open geospatial consortium (OGC) to InfraGML has a major gap in sensor to geo-dashboard interoperability. The work SHM-UCM implementation in [6] solved the interoperability problem by manual scripting that overcasts the manpower and technical resource in urban scale implementation by bulking the laborious jobs. Furthermore, the convergence [6-8] of application layer protocols MQTT, XMPP, CoAP, and AMQP with device libraries and information exchange template formats like JSON and XML again needs file-level parsing to assist tools like Ansible and Jenkin for network automation. Different simulation models and implementations were proposed over time [9]. Taking into account of UrbanSense to SEMSim hyper-converged implementation a plethora of gaps and peculiarities can be observed like information-centered gateways automation from sensors to application front-end and interoperability of the UrbanSense platform and SEMSim clouds [10, 11]. A micro-Web Server gateway interface (uWSGI) is needed that can automate the Infrastructure as a Service (IaaS) skeleton for network and REST API milestones using WebSockets [12]. The massive data produced in urban scale cloud operations [13] needs to be managed for creating, reading, updating, and deleting (CRUD) purposes in Big Data interfacing with platforms such as Hadoop Distributed File System (HDFS) [13-15]. The work in [16] proposed geospatial intelligence enablement using machine learning that needs real-time sensor data storage optimization and access for machine learning. In [17, 18], Software as a Service (SaaS) applications' automation needs IaaS access control to assist big data and machine learning capabilities. The public cloud tools for Azure, AWS and any third party need APIs interoperability, and the same stands also for open-source private clouds like Kaa, Kubernetes, and OpenStack [19-22]. An API processing wrapper is needed that can facilitate inter-cloud APIs to work while controlling the uWSGI and CGI at the same time [21, 23]. This is proposed, as a novel contribution, to this work as an interoperability API wrapper (IAW). The heterogeneity enablement using a single interoperable API wrapper is the proposed solution.

## 2 Problem Formulation

The connection of people, processes, systems, frameworks, and platforms is a slugfest task. Moving from instrument to SaaS requires in-depth integration wrappers programming [24]. This work has four module structures that map the functionality and applications in one layout. The topmost layer Business as a Service(BaaS) interacts with Thing as a Service using IAW.

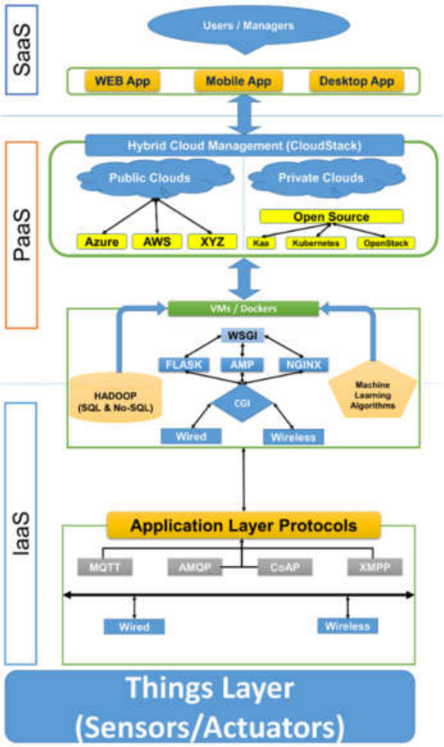


Fig 1. IAW Model Architecture Bridges TaaS-BaaS Functional Gap

Fig. 1 presents the complete concept and implementation block diagram enabling a BaaS-TaaS interaction. The addition of only 4 interface layers can perfectly map Industry 4.0 and BaaS-TaaS workflow architecture and frameworks with OSI architecture from software, hardware, and communication prospect. Following are the four layers of this work as a single tunnel:

A. Level 0 Things-IaaS API Wrapper
B. Level 1 IaaS-PaaS API Wrapper
C. Level 2 PaaS-SaaS API Wrapper
D. Level 3 TaaS-BaaS API Wrapper

This work enables the integration of the OSI model and the IoT standard architecture and framework by filling the fundamental gaps. The cost, effort, and time can be minimized by optimizing the OSI skeleton mapping over the Industry 4.0 skeleton as a one-turnkey solution.

## 3 Problem Solution

The gaps in the problem definition presented in fig 1 were addressed using a four-layer interoperable API wrapper (IAW) that can be integrated and deal with the physical cloud infrastructure as well as a PiL-MiL/HiL-SiL unified Digital Twin architecture. The effort and time can be minimized by optimizing the OSI skeleton mapping over the Industry 4.0 skeleton as a one-turnkey solution.

### 3.1 Level 0 Things-IaaS API Wrapper

Step 0, the ICS network architecture realization consists of $N_{\text{ICS}}$ sensor/actuator nodes with Industrial Communication Network Bus ($B_{ICN}$) with a topology $T_{ICS}$ as standard ICS and for a specific Process Control System (PCS) as $T_{ICS\text{-}PCS}$, Manufacturing Execution System (MES) as $T_{ICS\text{-}MES}$, Batch Processing Systems (BPS) as $T_{ICS\text{-}BPS}$. Let's consider an ICS-IIoT system with N number of PLCs connected in topology $ICS_{IIoT\text{-}PLCs}$ with gateway $ICS_{IIoT\text{-}Gateway}$ having X interfaces for GIS-SCADA Server $S_{IIoT}$ for ICS cloud $C_{ICS\text{-}IIoT}$. The ICS-IIoT integration in a GIS Digital Twin is a system and when joined with an interoperability chain of Software in Loop (SiL), Hardware in Loop (HiL), Process in Loop (PiL) and Model in Loop (MiL) workflow. The maximum number of IIoT nodes (PLCs, VFDs, HMIs, etc) for a GIS-based ICS-IIoT in a $ICS_{IIoT\text{-}Gateway}$ concerning the maximum number of packets $ICS_{\text{Packets}}$, for a given gateway, are given as follows

$$ICS_{\text{Packets}} = \sum(ICS_{IIoT-Gateway}(\mathrm{I})) + \sum \Delta G_{ICS-IIoT}\} \tag{1}$$

where $\Delta G_{ICS\text{-}IIoT}$ is the change in the geospatial patch of ICS system deployment.

## 3.2 Level 1: IaaS-PaaS API Wrapper

The Things-IaaS has to be added with network and web-scripting capabilities like *JhonnyFive* (i.e. JS), *Bash*, *PowerShell*, and Python. Furthermore, *Ansible* is a basic open-source IT infrastructure automation tool that allows easy application deployment, intra-service orchestration, cloud provisioning, etc.

The round-trip time (*RRT*) for JSON and XML based data transfer for a given DevOps deployment YML playbook be $T_{ICS\text{-}YML\text{-}RRT}$ for a single variable for any sensor/actuator in PLC node $N_{PLC} = \{N_{PLC1}, N_{PLC2}, \ldots N_{PLCN}\}$ for a specific GIS-based ICS sitemap base is the difference between the time to send (TTS) and application constraint channel tunnel allocation time $T_{YML\text{-}Ch\text{-}Alloc}$ at a given geographical area under $G_{ICS\text{-}IIoT\text{-}Obs}$ observation as:

$$T_{ICS\text{-}YML\text{-}RRT} = T_{YML\text{-}Ch\text{-}Alloc}(G_{ICS\text{-}IIoT\text{-}Obs}) - T_{YML\text{-}TTS}(\Delta G_{ICS\text{-}IIoT}) \quad (2)$$

In $B_{ICN}$, every packet from for a unit actuator or sensor connected to PLC for a specific $C_{ICS\text{-}IIoT}$ data loop for PiL will be based on the equation (2) and vary for different $T_{YML\text{-}Ch\text{-}Alloc}$.

## 3.2 Level 2: PaaS-IaaS API Wrapper

The interaction of Keras-Tensorflow using Docker, MapReduce-Hadoop using VM at WSGI is possible by a python API. This API would be using the *pyvmomi* library [13] with *ESXi* Server or *VSphere*. Furthermore, *pyvmomi* reacts with *PyDoop* to interact with *MapReduce* to optimize and control *Hadoop* operations. To allow the Cloud to interact with Sensors we propose to use a Python-PHP-*WebSockets* API. The architecture model is given in Fig. 2. In Fig. 2, a complete python-focused RPC-API implementation is displayed that uses RPCs to handle cloud APIs using python *CloudMesh*. *ARISTA eAPI* connects uWSGI with *vSphere* and *ESXi* clients assisted by *PyDoop* can be used to interact with *Hadoop*. Secondly, *gunicorn* could be used with *Connexion* to make the best out of *Tensorflow* and *Keras* deep learning interfaces inside *Docker* (an Enterprise Application Container Platform). *PyOne* handles XML-RPC API and *Python JSON-RPC* API handles JSON and routes XML and JSON through a *REST API* in the overall uWSGI architecture. Three key nodes with bottlenecks exist in the automated ICS infrastructure: a) A CGI, the bottleneck in $B_{ICN}$ between the PLCs, RTUs, and SCADA; b) uWSGI bottleneck between Docker and the VMs for Digital Twins to emulate the entire infrastructure; c) *ARISTA eAPI* core, the bottleneck between uWSGI and CloudStacks. This created a situation where the entire throughput $TP_{ICS\text{-}IIoT}$ became dependent on the *ARISTA eAPI* core and uWSGI. For a given $G_{ICS\text{-}IIoT\text{-}Obs}$, there must a exist a minimum value for $T_{ICS\text{-}YML\text{-}RRT}$ for the sum of all allocated channels under $T_{YML\text{-}Ch\text{-}Alloc}$ as well as their effective areas for a set of unique interfaces $I = \{I_1, I_2, \ldots I_N\}$ given as:

$$TP_{ICS\text{-}IIoT} = \frac{\text{size}\{\Sigma i\, G_{ICS-IIoT-Obs}(i) + \Sigma i \Delta G_{ICS-IIoT}(i)\}}{T_{ICS-YML-RRT}} \quad (3)$$

Higher the number of dynamically allocated $T_{ICS\text{-}YML\text{-}RRT}$ less will the throughput of PiL-MiL chain in ICS and its digital twin respectively.

## 3.3 Level 3: TaaS-BaaS API Wrapper

The user-centered automated operations and processes are the conscience of Industry 4.0 based architectures. The maximum throughput of ICS $TP_{ICS\text{-}IIoT}$ can be calculated from PiL-MiL chain for $T_{ICS\text{-}YML\text{-}RRT}$ for all the gateway interfaces termed as the gateway mesh for a specific YML in the current state of the ICS Cloud. Based on ICS-YML and the RPC-APIs the total channel capacities based on the $C_{Gateway\text{-}YML\text{-}Ch\text{-}Alloc}$ data rate size divided by the sum of channel-specific round time trips are given as:

$$TP_{ICS\text{-}IIoT} = \frac{size\{\sum_i C_{Gateway-YML-Ch-Alloc}(i)\}}{\sum_i T_{ICS-YML-RRT}(i)} \quad (4)$$

Precisely it is based on giving access to users to control the instrumentation below through parametric flexibility to achieve the production goals [25].

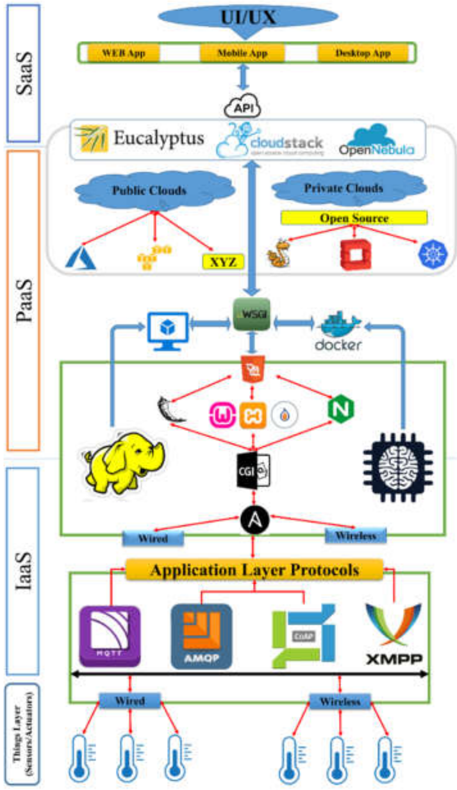


Fig 2. IAW Implementation for Industry 4.0 based ICS-IIoT Cloud

In fig 2, a clear formulation has been shown that enables a user to interact from a user interface (UI) for the price paid to have a respective user experience (UX). Furthermore, it can be elaborated as:

1. Any premium user will have cloud-level business services supported by machine learning and big data regardless of the cloud being used.
2. Any enterprise user will have the webserver portal to invest in his needed platforms to complete the task in a cloud he/she is part of.
3. Any basic user will have mobile apps or web clients to only interact with the factory instrumentation underneath.
4. The ICS-IIoT is always a multiple input and multiple output systems in a PLC/DCS and SCADA systems for Digital Twin integration it must be a super-set of the model predict controllers (MPCs) as MIMO-MPCs for KPIs control using PiL/MiL integration presented in fig 3.

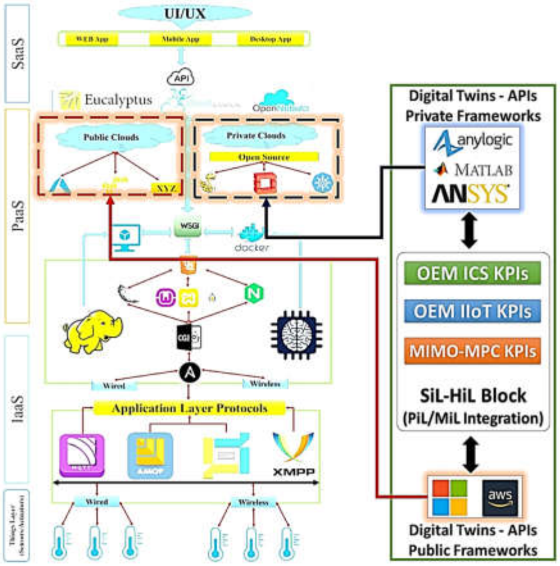


Fig 3. PiL/MiL based Integration of ICS-IIoT in Public and Private Cloud Digital Twins

In fig 3, it can be observed that at the factory/plant level a private cloud-based Digital Twin with local MiL/PiL in MATLAB-ANSYS and a public cloud-based Digital Twin with public MiL/PiL in any of Microsoft Azure or AWS will run continuously.

.

## 4 A Case Study of Smart Factories Cluster for Beverages Production using ICS-IIoT

The ICS-IIoT for the industrial internet of things (IIoT) gateway developed in our past work [26] has been capitalized as a case study.

### 4.1 Problem Statement

Supervisory Control and Data Acquisition (SCADA) systems are Industry 4.0 compliance production automation and integration systems that use IIoT. With the ever-varying production requirement and technological upgrades in the Beverages production ecosystem due to innovation in process analytical technology (PAT), 70% of the factories have shared facilities. The challenge in these shared facilities is the remote configuration and pre-production trial due to multi-OEM ISA equipment based on IEC 61131/61499 (PLC/DCS) for ANSI/ISA-88. The mega inter-factory production flow and inter-OEM technology interoperability challenges cost millions of dollars and that raises the costs and production time of ISO 9001 certified standard. Furthermore, the production technologists have to back up and test the automation codes for a single PLC 30% of the time. For multiple PLCs, this problem multiplies in magnitude.

### 4.2 Problem Statement

Capitalizing the proposed research, a python-based GIS YML to address this interoperability heterogeneity the implementation process is explained below:

1. The $\text{SCADA-IIoT}_{LOC}$ {(Lat1, Lon1), (Lat2, Lon2)} class calls one constructor for every $\text{Production}_{Schema}$ and created a 2D plane with a spatial diagram.
2. The production effective Basemap instance for production $\text{Production}_{BaseMap\text{-}Schema}$ is a vector file that was created for maximum outer boundary values of {(Lat1, Lon1), (Lat2, Lon2)}.
3. For every production application $\text{Production}_{Application}$, the $\text{Production}_{Schema}$ defined production infrastructure simulation will be performed for pre-production for control/mathematical models over the relevant PLC catalog numbers for 3 PLCs in three different smart factories: a) PLC1 at factory 1 (F1), PLC2 at factory 2 (F2), and PLC3 at factory 3 (F3).
4. The pre-trial of production was simulated in SCADA-IIoT Server and then an ANSI/ISA-88 compliance online models in loop (MiL) simulation was performed using IAWT ICS-IIoT with GIS Playbook Clustering.
5. The MiL quality assurance was verified for FDA CFR-21 as $G_{CSMI}$ instance; for a set of unique Production AI libraries $\text{Lib}_{Production\text{-}AI} = \{\text{Lib}_{Production\text{-}AI\text{-}1}, \text{Lib}_{Production\text{-}AI\text{-}2}, \ldots \text{Lib}_{Production\text{-}AI\text{-}N}\}$ the test $\text{Unit}_{Production\text{-}Test}$ for performed as per figure 4 comprehended V-Model.

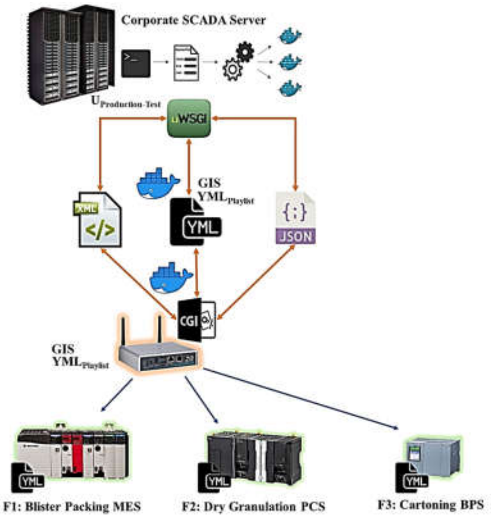


Fig 4. Smart Factories Production System Infrastructure for Geo-Spatial Production Application Implemented using IAWT ICS-IIoT GIS in Python

6. After the validation of the working configuration and end-to-end the IAWT ICS-IIoT for FDA CFR 21 in F1, F2, and F3, the GIS $\text{Production}_{Playbook}$ ($\text{SCADA-IIoT}_{LOC}$, GCSMI, ALBS, Gateway ID, IP, MAC) was created for every production gateway node connected to the production schema relevant PLCs.
7. As a fail-safe GIS cluster, a Production IAWT ICS-IIoT MiL docker was packaged and transferred to every $\text{EAI-IIoT}_{Gateway}$ for $\text{Production}_{Application}$.

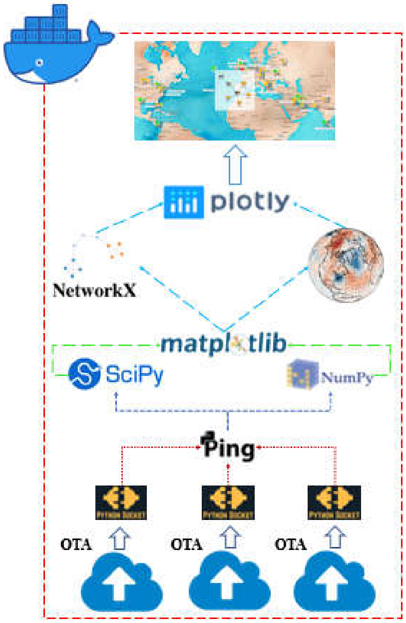


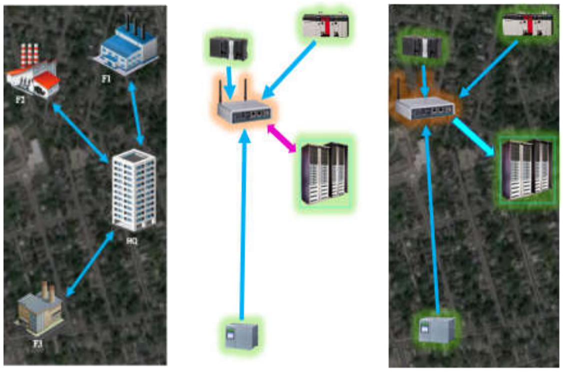
Fig 5. IAWT ICS-IIoT MiL Docker Packaged as BackUp

8. After the OTA by the backup of the last working firmware and configuration of the PLCs the $\text{EAI-IIoT}_{Gateway}$, existing QoS was recorded by using python library PyPing a python command sent a ping after ever Ack_Flag=TRUE to pool the production topology.
9. In IAWT ICS-IIoT MiL Docker, the pings were stored as real-time UNIX clock format variables using SciPy and statistically processed as YML playbook with arguments as ($\text{Production}_{Packet}$, $\text{Production}_{BaseMap\text{-}Schema}$, $\text{EAI-IIoT}_{Gateway}$ Data, Production Topology, Acknowledgement Flag) at UNIX time t.
10. The Plotly data frame is formed by mapping $\text{Production}_{Topology}$ over $\text{Production}_{BaseMap\text{-}Schema}$.
11. The arrow colors and dots are shown in Figs. 3 and 4 are plotted as it is with the same semantic descriptions in the final GNA results from ours.
12. The arrow colors and dots are shown in Figs. 3 and 4 are plotted as it is with the same semantic descriptions in the final GNA result.

These 12 operations ensured the successful implementation of IAWT ICS-IIoT and deployment of the GIS Playbook over the Edge AI IIoT Gateway.

## 4 Results and Discussion

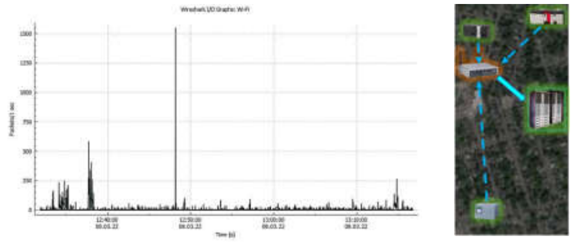
The systematic stepwise implementation of the 12 IAWT ICS-IIoT MiL based on procedures presented in Sections 2 and 3 enabled the QoS results. Steps 1 and 2 mapped created geospatial charts displayed in figure 6 given below.



a. HQ at Google Earth b. Spatial Layout c. Mapped Layout

Fig 6. IAWT ICS-IIoT Operations 1 and 2 for HQ MiL Deployment

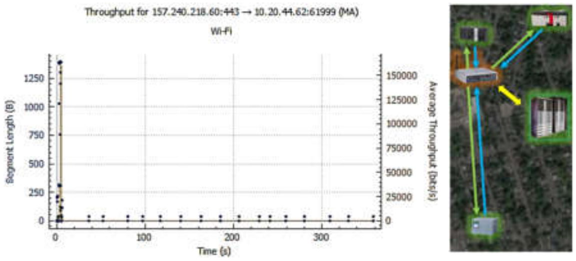
The results followed the proposed methodology steps sequence given in section 2 for regional computations. The second step was virtual circuit socket creation and start transmission and establishing a link between HQ SCADA-IIoT Server and IIoT Gateway and Factories.



a. Wireshark I/O Graph b. OTA BackUp

Fig 7. ICS-IIoT GIS results for Factory Sites to Gateway, $G_N$ I/O Graph for Wireshark I/O Graph for SCADA-IIoT$_{Address}$

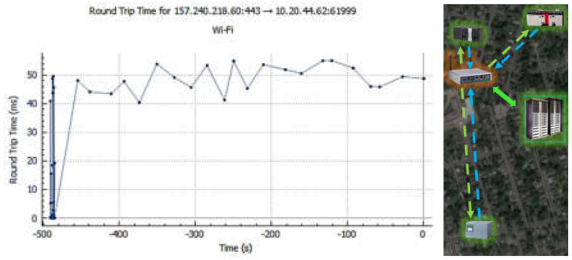
In Fig. 7, the 3 PLCs backups to the IIoT gateway and its archiving at the start of the process is exhibited. The maximum 1535 packets/second transmission took place from factory sites to IIoT Gateway i.e. 49,120 bits/second as per table 2. Only OTA backup data F1, F2, and F3 were transmitted upward in the hierarchy. For 100 sensors/actuators at a single PLC with 100+100 floating-point 32-bit variables per PLC per site, theoretically segment length needs to be stabilized to the lowest value after the GIS has been deployed on the gateway.



a. Wireshark Segment Length Trace b. IAWT MiL

Fig 8. Reduction in Throughput after the deployment of GIS implementation. for Factory Sites to Production$_{Gateway}$

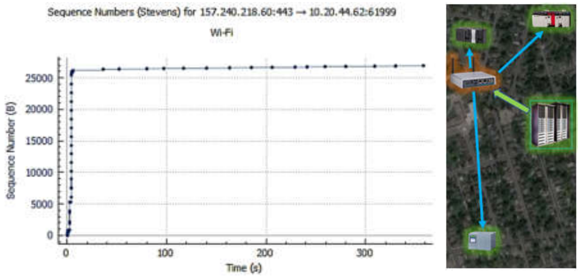
From (5) the WiFi$_{MSS}$ achieved a segment length of 1388 bytes for 7 seconds only is less than the MTU, which means that the Production$_{Playbook}$ was deployed without using the maximum capacity of IIoT network capacity. Achieving event response below MTU at such a low data rate was the very competitive benchmark. The less than 20 Bytes segment length will result in lower round trip times (RTTs).



a. Wireshark Trace for GIS Round Trip Time b. $G_N$ Ping

Fig 9. RTT results for successful deployment and stable production Operations based on ICS-IIoT GIS results for Factory Sites to $G_N$

In fig 9, the pre-settlement time between (40 to 56) ms was a better docket porting performance range for ISA equipment based on IEC 61131/61499 (PLC/DCS) for ANSI/ISA-88.



a. Wireshark Sequence Numbers Trace b. MiL Deployment

Fig 10. Relative Stationarity in Sequence Numbers for Factory Sites to $G_N$

In Fig. 10, the stationarity of sequence numbers exhibits consistency in acknowledgment. After 26000+ sequence numbers, the IAWT ICS-IIoT data stream achieved maturity as reflected by a horizontal line. The GIS events generation stopped 27000th token at 360 seconds means not all possible $V_i$ updates and event responses are expected.

## 4 Conclusion

This work has enabled a secure poly-lithic API wrapper tunnel to enable secure Industry 4.0 deployment without the usage of conventional cybersecurity tools like SSL, TSL, OAuth, firewalls, and encryption. Infrastructure has been secured with the help of heterogeneity and interoperability tunnel. The results can be summarized in 3 key findings: 1) Systematic interoperability is the basic constraint for dense heterogeneous installation to work successfully using inter-sector wrappers; 2) The APIs have to talk to each other to finish the end-to-end chaining; 3) Global or Urban user-centered applications must have a UI and UX flow to interact with the lowest instrument in the Industry 4.0 entrusted business service. The cloud automation APIs and the wrapper API constitute one of the prime achievements that have enabled gigantic transformations of IoTs for stocks and businesses.

## 5 Future Recommendations

In our current research, this effort contributed to beverages industry, in future this research can be improved and some possible gaps that may exist in this research by utilizing this research as base for:

1. AI-based Digital Twins for Industry 4.0
2. AI-based Digital Twins for Computer Integrated Manufacturing Robotic Work Cells.

3. GIS-Based ICS-IIoT Digital Twins with Spatio-Temporal Topology Control of Production Flows in Packaging in Pharmaceuticals.
4. Geo-AI for optimization of Digital Twins in Process in Loop and Model in Loop for Batch Processing Systems.

*References:*

[1] Daniel Zehe et al, SEMSim Cloud Service: Large-scale urban systems simulation in the cloud. *Simulation Modelling Practice and Theory*, September 2015.

[2] Stella Vetova, Big Data Integration and Processing Model. *WSEAS Transactions on Computers*, vol. 20, pp. 82-87, 2021

[3] Farid Touati, Hasan Tariq, Damiano Crescini, and Adel Ben Mnaouer, (2018). Development of Prototype for IoT and IoE Scalable Infrastructures, Architectures and Platforms. *Ubiquitous Networking. UNet 2018. Lecture Notes in Computer Science*, vol 11277. Springer, Cham.

[4] Zaheer Khan et al, An architecture for integrated intelligence in urban management using cloud computing. *Journal of Cloud Computing: Advances, Systems and Applications*, 2012.

[5] Abdullah A. Wardak, Interfacing C and TMS320C6713 Assembly Language (Part II). *WSEAS Transactions on Computers*, vol. 20, pp. 74-81, 2021.

[6] Hasan Tariq, Farid Touati, M. Al-Hitmi, Adel Ben Mnaouer, and Damiano Crescini, A Real-time Early Warning Seismic Event Detection Algorithm using Smart Geo-spatial Bi-axial Inclinometer Nodes for Industry 4.0 Applications. *Applied Sciences,* 2019.

[7] Wang Jianhong, Ricardo A. Ramirez-Mendoza, Application of Interval Predictor Model Into Model Predictive Control. *WSEAS Transactions on Systems*, vol. 20, pp. 331-343, 2021.

[8] Hasan Tariq, Farid Touati, M. Al-Hitmi, Adel Ben Mnaouer, and Damiano Crescini, Design and Implementation of Information Centered Protocol for Long Haul SHM Monitoring. *IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems*, 2019.

[9] Luca Tamburini et al, Electronic and ICT Solutions for Smart Buildings and Urban Areas. *Renewable and Alternative Energy: Concepts, Methodologies, Tools, and Applications*, 2017.

[10] Hasan Tariq, Farid Touati, Mohammed Abdulla E Al-Hitmi, Anas Tahir, Damiano Crescini, and Adel Ben Mnaouer, Structural Health Monitoring and Installation Scheme Deployment using Utility Computing Model. *2nd European Conference on Electrical Engineering and Computer Science*, 2018.

[11] Eman Emad, Omar Alaa, Mohamed Hossam, Mohamed Ashraf, and Mohamed A. Shamseldin, Design and Implementation of a Low-Cost Microcontroller-Based an Industrial Delta Robot. *WSEAS Transactions on Computers*, vol. 20, pp. 289-300, 2021

[12] Hasan Tariq, Farid Touati, Mohammed A. E. Al-Hitmi, Damiano Crescini, and Adel Ben Mnaouer, Design and Implementation of Programmable Multi-parametric 4-degrees of Freedom Seismic Waves Ground Motion Simulation IoT Platform. *15th International Wireless Communications & Mobile Computing Conference,* 2019.

[13] Luigi Maxmillian Caligiuri, Antonio Manzalini, Quantum Hypercomputing and Communications: Overview and Future Applications. *WSEAS Transactions on Computers*, vol. 20, pp. 247-257, 2021.

[14] Hasan Tariq, Abderrazak Abdaoui, Farid Touati, Mohammed Abdulla E Al-Hitmi, Damiano Crescini, and Adel Ben Mnaouer, An Autonomous Multi-variable Outdoor Air Quality Mapping Wireless Sensors IoT Node for Qatar. *IEEE International Wireless Communications & Mobile Computing Conference,* 2020.

[15] Merve Nur Cakir, Mehwish Saleemi, Karl-Heinz Zimmermann, Dynamic Programming in Topological Spaces. *WSEAS Transactions on Computers*, vol. 20, pp. 88-91, 2021

[16] Hasan Tariq, Abderrazak Abdaoui, Farid Touati, Mohammed Abdulla E Al-Hitmi, Damiano Crescini, and Adel Ben Mnaouer, Design and Implementation of Multi-Protocol Data Networks Interface Detector in Heterogeneous IoTs. *IEEE International Conference on Informatics, IoT, and Enabling Technologies,* 2020.

[17] Kellee Farris, Subhashini Ganapathy, and Mary Fendley, Presenting Trends in Petrochemical Process Control Systems. *WSEAS Transactions on Computers*, 2224-2872, Volume 19, Art. #24, pp. 194-200, 2020.

[18] Hasan Tariq, Abderrazak Abdaoui, Farid Touati, Mohammed Abdulla E Al-Hitmi, Damiano Crescini, and Adel Ben Mnaouer, A Real-time Gradient Aware Multi-Variable

Handheld Urban Scale Air Quality Mapping IoT System. *IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems,* 2020.

[19] Aydin Teymourifar, Ana Maria Rodrigues, Jose Soeiro Ferreira, Geographically Separating Sectors in Multi-Objective Location-Routing Problems. *WSEAS Transactions on Computers*, Volume 19, 2020, Art. #13, pp. 98-102, 2020.

[20] Hasan Tariq, Farid Touati, Mohammed A. E. Al-Hitmi, Damiano Crescini, and Adel Ben Mnaouer, Design and Implementation of Cadastral Geo-spatial IoT Network Gateway Analyzer for Urban Scale Infrastructure Health Monitoring. *10th Annual Computing and Communication Workshop and Conference,* 2020.

[21] Stanislav Bovchaliuk, Sergii Tymchuk, Sergii Shendryk, Vira Shendryk, The Fuzzy Control Automation Architecture of Parallel Action for the Intelligent Smart Grid Networks. *WSEAS Transactions on Computers*, Volume 19, 2020, Art. #3, pp. 21-25, 2020.

[22] Hasan Tariq, Abderrazak Abdaoui, Farid Touati, Mohammad Abdullah Al Hitmi, Damiano Crescini, and Adel Ben Mnaouer, Real-time Gradient-Aware Indigenous AQI Estimation IoT Platform. *Advances in Science, Technology and Engineering Systems Journal Vol. 5, No. 6, 1666-1673,* 2020.

[23] Muneer Bani Yassein, Omar Alzoubi, Saif Rawasheh, Farah Shatnawi, Ismail Hmeidi. Features, Challenges and Issues of Fog Computing: A Comprehensive Review. *WSEAS Transactions on Computers*, Volume 19, 2020, Art. #12, pp. 86-97, 2020.

[24] Hasan Tariq, Abderrazak Abdaoui, Farid Touati, Mohammed Abdulla E Al-Hitmi, Damiano Crescini, and Adel Ben Mnaouer, Design and Implementation of a Multi-Parametric Geo-Seismic Realization Engine for Programmable Mechatronic IoT Geo-Mechanics Simulators. *International Journal of Geology,* 2019.

[25] Roumen Trifonov, Slavcho Manolov, Georgi Tsochev, and Galya Pavlova, Automation of Cyber Security Incident Handling through Artificial Intelligence Methods. *WSEAS Transactions on Computers*, Volume 18, 2019, Art. #35, pp. 274-280, 2020.

[26] Hasan Tariq, Abderrazak Abdaoui, Farid Touati, Mohammed Abdulla E Al-Hitmi, Damiano Crescini, and Adel Ben Mnaouer, IoT/Edge Structural Health Monitoring System as a Life-Cycle Management tool for SDG-11 using Utility Computing Platform. *WSEAS TRANSACTIONS on COMPUTERS, 2019.*

## Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

**Author Contributions: Please, indicate the role and the contribution of each author:**
Example
Hasan Tariq performed the conceptual study, research methods, and case study (sections 1-3).
Shafaq Sultan has organized the manuscript and formatting and done the write-up (sections 4-5).

## Sources of funding for research presented in a scientific article or scientific article itself

Self-Funded.

## Creative Commons Attribution License 4.0 (Attribution 4.0 International , CC BY 4.0)