Review Article

System-Independent Server-Side Infrastructure for IoT Applications

Vural Avşar[1*], Özgür Tamer[2*]

[1] Department of Mechatronics Engineering, Dokuz Eylül University, Orcid ID: https://orcid.org/0009-0005-2468-6232, E-mail: vural.avsar@gmail.com
[2] Department of Electrical and Electronics Engineering, Dokuz Eylül University, Orcid ID: https://orcid.org/0000-0002-5776-6627, E-mail: ozgur.tamer@deu.edu.tr
[*] Correspondence: avsar.vural@ogr.deu.edu.tr

(First received September 12, 2023 and in final form December 21, 2023)

Reference: Avşar, V., Tamer, Ö. System-Independent Server-Side Infrastructure for IoT Applications. The European Journal of Research and Development, 3(4), 381-389.

Abstract

Precision of production in industry is a hot area and companies are seeking for solutions to make data-drive decisions to optimize the processes of production and calculate gain/loss during manufacturing. Making a right decision with huge reports always takes time and executive or decision makers always seek user-friendly, automatised applications to help.
This paper offers a server-side infrastructure for IoT applications that targets to provide an end-to-end solution to smart factories on production lines. Solution designed as a middleware application in which northbound and southbound systems are connected to each other. Collected data is stored and integrated with BI tool to present gain/loss during production to executive management and same output can be used for MES-MOM operations. OPC-UA is selected as southbound protocol as it can be considered the most used protocol in industrial IoT. Moreover, solution can be enhanced with any other protocol with slight changes in configurations.

Keywords: *IoT, MES-MOM, Server-Side Application, OPC-UA*

1. Introduction

With recent developments on internet and telecommunication technologies, internet became vital in our daily lives. We are introduced different IoT (Internet of Things) applications every day. The evolution of the Internet led to continuous development in industry and revealed the concept of Industrial IoT (IIoT). It is about collecting data from an industrial infrastructure using smart sensors, actuators, real-time monitoring, analysing collected data communicating and exchanging data through Internet connections. The goal is to give industries greater efficiency, robustness, reliability, and high-performance infrastructure.

Parallel to the evolution of IoT technologies, several standardization initiatives have played a role in the IoT domain to design a common language for industrial applications. The most common protocols in the IIoT domain are MQTT, COAP, LWM2M, and HTTP. Protocols such as Modbus TCP or Profinet can provide real-time data, but cannot provide safe and reliable historical data, and protocols such as RTI DDS do not emphasise the integration of industrial equipment (Lin, 2019). OPC-UA is another standard that became popular recently and widely accepted in the industry even though it had already been defined as a standard long before. OPC stands for "Open Platform Communications" and UA stands for "Unified Architecture". It is a client-server protocol used for industrial communication between multiple devices and offers an abstraction that describes the kind of messages that can be exchanged between a client and a server (Karaagac, 2019).

Several studies for IIoT applications have been presented in the literature on different protocols. A web-based platform for OPC-UA integration in IIoT environment had already been developed. The main idea behind their application is to let web-users exchange data from servers with OPC-UA without any knowledge of the standard (Cavalieri, 2017). "OPC UA provides a future-oriented framework where innovative technologies and methods such as new transmission protocols, security algorithms, coding standards or application services can be incorporated into OPC UA while maintaining compatibility with existing products; for modeling capabilities, OPC UA information modeling framework can transform data into information, with object-oriented technology, even very complex multi-level structures can be modeled and extended." (Dantao, 2022). OPC-UA can provide many features that OOP can provide like data abstraction, encapsulation, etc.

According to research by S. Mantravandi, manufacturing execution systems (MES) became a key system in smart factory terminology and customization and reconfigurability of factory is a problem. They developed a framework to assess reconfigurability for smart factory development (Mantravadi, 2020). The framework allows manufacturers to respond market needs and changes quickly.

In addition to that, another study has proposed a management architecture that supports automatic discovery and aggregate of OPC-UA servers (Pu, 2022). Their platform enables a centralised configuration management method for the OPC-UA communication data stream. Their experimental result shows that the platform can realise the automatic discovery of multiple OPC-UA servers and mapping of information models.

When industrial IoT supported with cloud technologies, huge amount of data can be transferred to cloud and cloud-edge algorithms process the data efficiently. OPC-UA provides more advantages during data transmission from devices to cloud. Using cloud

computing platforms, time-taken operations on the server side can be easily processed on the cloud side, like image processing (Lin, 2019, Shi, 2020).
Secured communication is another important topic in IoT domain. Because IoT equipments generally plugged into internal network and do not use any secure method for communication. And without secure communication, the system will be open to vulnerabilities, and the system can be down with intentional actions. An OPC-UA protocol mutually authenticate themselves with certificates and generate ssh keys for bi-directional communication. OPC UA uses X.509 compliant certification and this certificate includes a digital signature. Based on certificates and keys, OPC UA provides application-level security (Kohnhauser, 2022).
In this paper, we propose a server-side infrastructure for the IIoT application that can be used to collect, exchange data with the OPC-UA protocol, store, and analyse the collected data to provide meaningful outputs for MES-MOM operations. According to our review of the literature, we have found many different developments with a single approach, such as the exchange of data only, the analysis or the focus on the reconfigurability of devices only. We could not find any solution that has a BI connection to provide meaningful outputs. Application is platform independent which means it can be run on any operating system. We implemented this solution in a packing station to print metadata -which is called lot details- for each pack coming from ERP system onto packs and collect metadata of those packs, like actual weight, etc.- from different PLC servers for each package on the production line. Collected data is analyzed and visualized with dashboards via BI tool as well to present executive managements. The main advantages of having this application are that it is a platform-independent server-side application and that it can be easily integrated with ERP systems. System-in-a-package has a BI tool also and users can easily prepare executive dashboards using that BI tool.

2. Problem Description

The HIPE factory (High Intense Polyethylene) manufactures polyethylene, and the filling machine fills this product into packs. Each pack must be 25 kg and based on customer needs, multiple packs can be shipped to customer premises. Order is created in SAP system, but all the information related with that is sent to packing station via hand-written order request form. As there is no integration between SAP and packing station; when new request order comes, operator types all the information into PLC servers manually referring the order request form.
Once production process starts, product is filled by filling machine and sent to production line. The pack is weighted on the line and moved to the metal detection phase. In this phase, the detector checks if any metal is mixed into the pack by mistake. If there is no metal, pack is moved to lot writer and all the details about the product is written to pack by laser. After that, packed product is moved to storehouse to be shipped.

Due to the accuracy of the filling machine, there can be deviation on total weightage of each pack, and this deviation on total weightage is not logged in any system. Thus, the deviation between expected production and actual production is unknown to anyone or system, and it causes profit/loss to the factory. As there is no integration between SAP and the packing station, in case of any typo error when putting the lot details into the packing system, different products can be packed and shipped to the customer. Once it is realized, production should be start again which causes delay and loss to factory.
All this loss/gain on total shipment and order-to-delivery process is requested to be logged and visualized with dashboards to present executive management by factory managers. As a solution to overcome this problem, a platform-independent server-side infrastructure is proposed and developed. This newly developed application can integrate with any third-party system and all metadata related with production process can be logged and visualised. The OPC-UA protocol is used to collect and exchange data between the PLC server and the application.

3. System Architecture

The client-server communication model is the basis of the OPC-UA protocol. OPC server provides a standardized interface to outside environment to provide information via services like read, write, or browse. OPC Client is used to connect to OPC Server and consume provided services to exchange data. In this manner, Siemens S7-1500 is selected as OPC Server in our solution. S7-1500 provides an OPC-Server and allows OPC access [10]. S7-1500 is used as centralised OPC Server that connects to subservers with any protocol that sub-servers allow. Device gateway in our system serves as OPC-Client to collect and exchange data from centralize PLC server. To simulate a server as an OPC server, the ProSys simulator is used to simulate a centralized server.
The application is designed to enable end-to-end connection between devices and northbound systems. The system architecture has four main layers as described in the following.

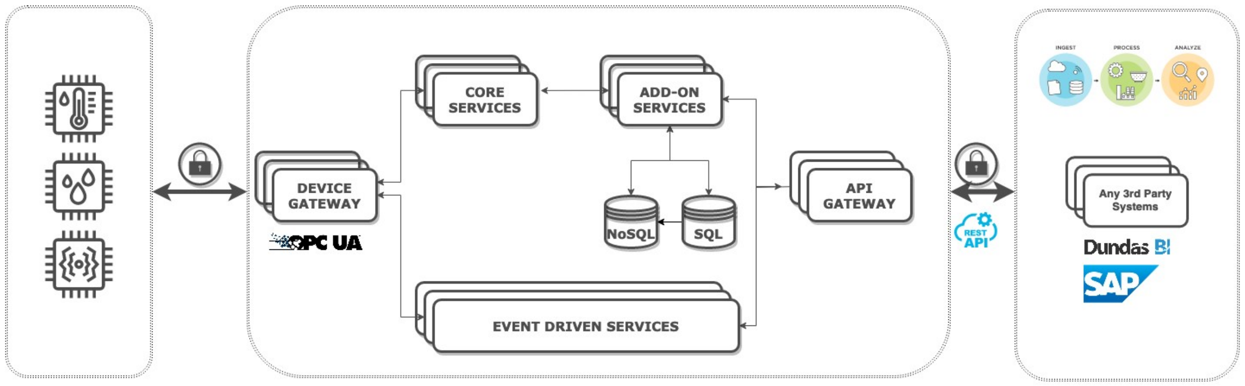


Figure 1: Solution Architecture

Core and add-on services are designed to allow collected data retention and save collected data in database. The objective of this approach is data collection, verification, and representation along with real-time processing. In this solution, collected data is manipulated and saved in relational database.
Dump data in database is consumed by Dundas BI Tool to create visual output to present executive management. The system is designed to add an n-number add-on service based on customer needs.
Event Driven services are used by third-party systems to exchange real-time data. Third-party systems can connect to the OPC Client directly to exchange data with PLC servers. API Gateway is used to communicate with third-party systems. Event-driven services and core/add-on services are consumed by this gateway and serve third-party systems with a secured protocol. API Gateway uses REST protocol for communication. Any protocol can be added to this gateway to serve other systems using newly added protocol. In the same way, any other adapter can be added to the device gateway protocol to communicate with PLC servers using an API other than the OPC-UA protocol.
The application solution is dockerized and deployed as a package. In addition to the three main layers described in Figure 1. Left box represents PLC servers to be connected. The box in the middle shows our application diagram, and the box in the right represents northbound systems that connect to PLC Servers through our application. ProSys OPC UA Simulator is also added to the container. System can be up and running quickly on any operating system.

3.1. System Features

System features can be grouped under six categories which are:

3.1.1. Telemetry Data Collection

This application allows collect/exchange and store telemetry data in reliable way. Real-time data can be saved in a relational/non-relational database or consumed by any third-party system with supported protocols directly without any manipulation.

3.1.2. Customization and Integration

The application can be integrated with the third-party system. As the application provides REST API to users, it can be integrated with any cloud systems using this REST API's as well. Any other protocol can be added to the API or device gateway to enable data exchange using different protocols.

3.1.3. Device Management

The application provides API for server-side applications to send RPC commands to devices to manipulate data.

3.1.4. Data Visualisation

With the help of the integrated BI Tool, users can easily visualise their stored data using built-in-line charts, digital and analogue gauges, maps, and much more. Based on the needs, the visualisations can be updated and reported to the specific group of people who has access.

3.1.5. Robustness

Both device and API gateways use the HTTPs protocol to provide secure communication. Data is encrypted with HTTPs protocol to increase robustness and security of data exchange.

3.1.6. System Independency

As application dockerized, docker technology provides system independency which means system can run on any operating system without any reconfiguration. Packed application contains all modules and dependencies that need to run our application.

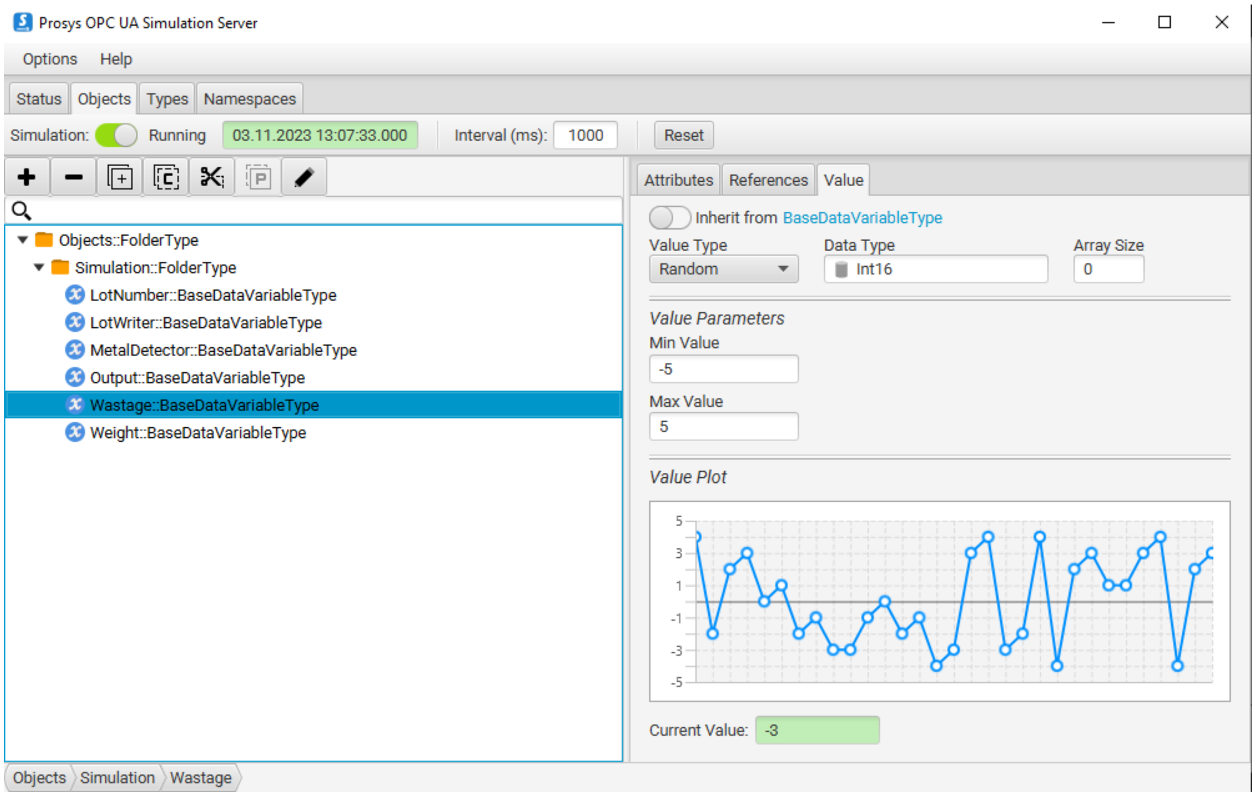
4. Simulation Results

As the aim of this solution is to provide an executive dashboard, a development environment is set for proposed application. All components are listed in Table 1 for a successful simulation.

Table 2: Platform for Development Environment

Platform	**Details**
ProSys OPC UA Simulation Server	Simulator to let Simulate Data
SQL Server Management Studio 19	Integrated Environment for managing databases
Docker Desktop	Platform to run Containerized Application
IntelliJ Community Edition	Integrated Development Environment for Coding
Dundas BI Tool (Trial Version)	Enterprise Platform for Visualization

There are five objects configured in ProSys Simulation Server as data (Figure-2). "LotNumber" is the shared attributes which can be updated by NB system to start manufacturing. When new lot detail submitted by SAP or any other northbound system with REST protocol, application updates the same data via device gateway in centralized server and collect mock data produced by simulation server from same gateway to save in relational database. Data in database is normalized before feeding BI Tool.



Prosys OPC UA Simulation Server

Options Help

Status Objects Types Namespaces

Simulation: Running 03.11.2023 13:07:33.000 Interval (ms): 1000 Reset

- Objects::FolderType
  - Simulation::FolderType
    - LotNumber::BaseDataVariableType
    - LotWriter::BaseDataVariableType
    - MetalDetector::BaseDataVariableType
    - Output::BaseDataVariableType
    - Wastage::BaseDataVariableType
    - Weight::BaseDataVariableType

Attributes References Value

Inherit from BaseDataVariableType

Value Type: Random
Data Type: Int16
Array Size: 0

Value Parameters
Min Value: -5
Max Value: 5

Value Plot

Current Value: -3

Objects > Simulation > Wastage

Figure 2: ProSys Simulation Server

'LotNumber' represents lot details to be printed to packages and initiator to start manufacturing. The objects 'LotWriter' and 'MetalDetector' are Boolean values. They represent if there is any metal detected in the packages and lot details printed on the packages on the line. "Weight" object is integer and represents total weight of each pack and "Wastage" represents the deviation of production. The simulator generates random values between -5 and 5 to simulate the waste of each pack.
Collected and saved data is consumed by Dundas BI via database created in SQL Server Management Studio 19 for below visualization.

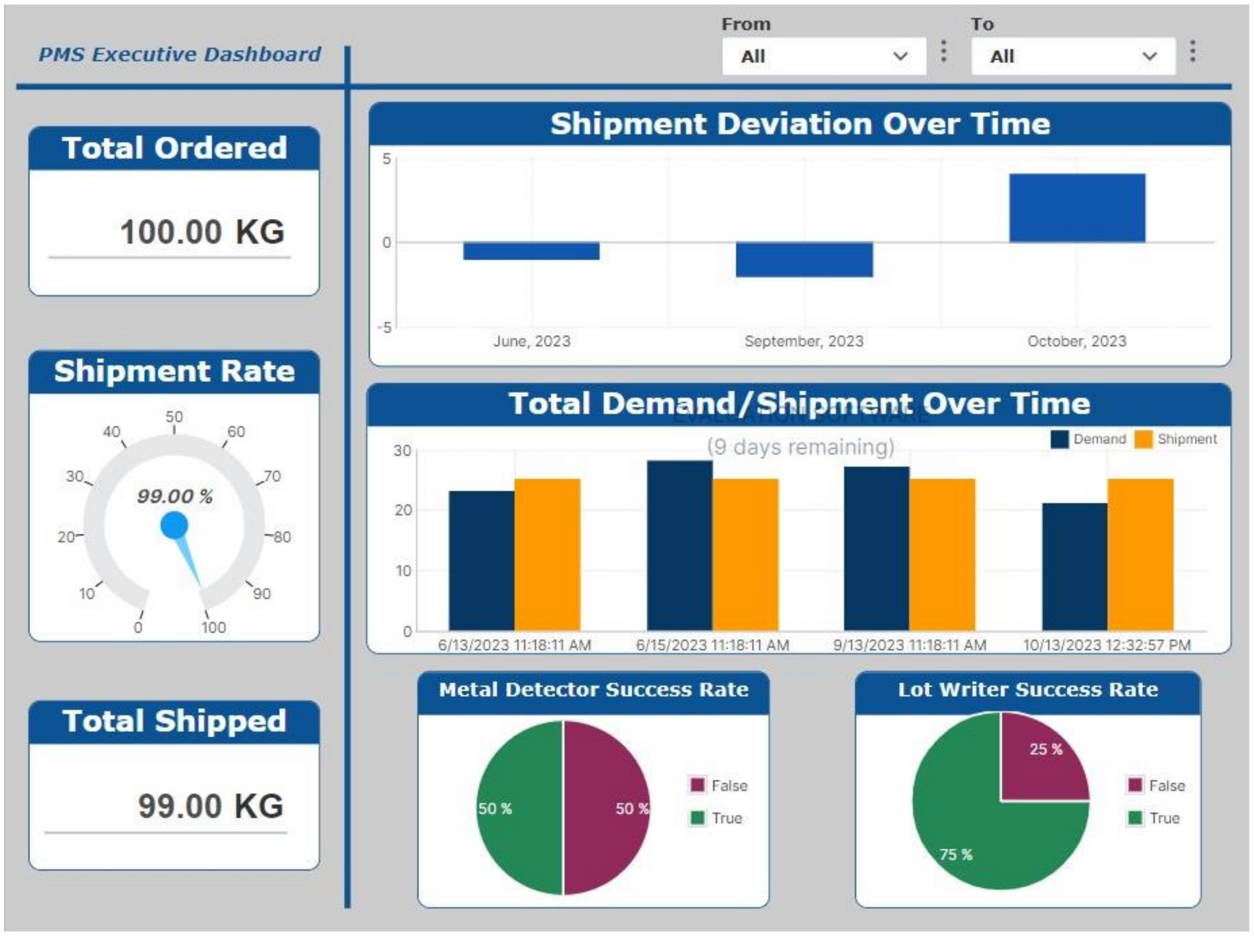


Figure 1: Dashboard for Executive Management

There are two main sections in the executive dashboard to provide details about total manufacturing. On the left side, we can see the "Total Ordered" field that shows the total value of requested manufacturing by northbound system in kilogrammes. The “Total Shipped" field populates the actual manufactured product in kilogrammes. The “Shipment Rate" shows the accuracy of manufacturing. According to Figure-3, the total ordered product is 100 kg, but the actual output is 99 kg. Therefore, the precision of the manufacturing is %99.

On the right side, we can see "Shipment Deviation Over Time" field which shows the deviation over months and next field "Total Demand/Shipment Over Time" shows both ordered and shipped value side-by-side. The last two fields at the bottom show the metal detector and lot writer performance rate. As per simulation, half of the packs contains metal inside to be removed and lot details printed to %75 of packed products.

End users can filter the result within the given date range. All charts are dynamically updated as per given criteria without refreshing the page.

5. Conclusion

The simulation results show that proposed application architecture is working as designed. Any third-party user can exchange data with servers which are using OPC-UA protocol directly or data can be collected in a database with or without manipulation. Collected data can be combined with data from third-party systems, and meaningful outputs are created and visualized using DundasBI Tool.
Application is designed to add additional services to enrich services system can provide. Current solution supports the OPC-UA protocol as a southbound protocol, but any other protocol can be added to the device gateway module. Likewise, the same additional protocol can be added to API Gateway to enable multiprotocol access.
The application uses Docker technology which can be deployed and run on any operating system. Thanks to the advantage of docker technology; in case of any environmental change, reconfiguration is not needed for this application. The same container can be deployed to the new environment quickly.

References

[1] Lin, H., & Hwang, Y. (2019b). Integration of Robot and IIoT over the OPC Unified Architecture. *2019 International Automatic Control Conference (CACS)*. https://doi.org/10.1109/cacs47674.2019.9024732

[2] Karaağaç, A., Verbeeck, N., & Hoebeke, J. (2019b). The Integration of LwM2M and OPC UA: An Interoperability Approach for Industrial IoT. *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. https://doi.org/10.1109/wf-iot.2019.8767209

[3] Cavalieri, S., Di Stefano, D., Salafia, M. G., & Scroppo, M. S. (2017b). A web-based platform for OPC UA integration in IIoT environment. *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. https://doi.org/10.1109/etfa.2017.8247713

[4] Dantao, H., Yanjie, G., & Daqian, X. (2022). Research on Key Technologies of OPC UA Standard and Test. 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC). https://doi.org/10.1109/IPEC54454.2022.977611

[5] Mantravadi, S., Srai, J. S., Brunoe, T. D., & Møller, C. (2020b). Exploring Reconfigurability in Manufacturing Through IIoT Connected MES/MOM. *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM*. https://doi.org/10.1109/ieem45057.2020.9309989

[6] Kohnhäuser, F., Grüner, S., & Heuschkel, J. (2022). Secure Onboarding of IIoT Devices using OPC UA. *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. https://doi.org/10.1109/etfa52439.2022.9921547

[7] Pu, C., Ding, X., Wang, P., & Yang, Y. (2022). Practical implementation of an OPC UA Multi-Server Aggregation and Management Architecture for IIOT. *2022 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing &Amp; Communications (GreenCom) and IEEE Cyber, Physical &Amp; Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*. https://doi.org/10.1109/ithings-greencom-cpscom-smartdata-cybermatics55523.2022.00099

[8] Shi, H., Niu, L., & Sun, J. (2020). Construction of the Industrial Internet of Things Based on MQTT and OPC UA Protocols. *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*. https://doi.org/10.1109/icaica50127.2020.9182598