International Journal of Engineering Innovation & Research
Volume 3, Issue 2, ISSN: 2277 – 5668

Software as Service Delivery Model for OPC Based Applications in Cloud Computing

Peter Peniak
Department of Control and Information Systems
Faculty of Electrical Engineering, University of Žilina, Slovakia
Email: peter.peniak@conti.sk

***Abstract* – The primary aim of the paper is to offer recommendations and to create a model for integration of OPC based applications within Cloud with Software as a Service delivery model. The main part of the paper describes the possibilities of the original OPC specification with respect to Cloud computing. The paper suggests to use dedicated OPC gateway with transformation of DCOM protocol to Cloud based protocols. Usage of OPC UA and dedicated OPC gateway for DCOM based traffic is essential part of the article.**

***Keywords* – Cloud Computing, IaaS, SaaS, OPC, OPC UA, PLC, SCADA.**

I. Introduction

Cloud computing is a model for enabling on-demand network access to a shared pool of configurable computing resources, such as servers, storage systems and application services. The Cloud computing defines three delivery models: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [1], [2].

OPC has become a worldwide standard for data and information exchange among process supervisory control systems, PC-based controls, MES (Manufacturing Execution System) systems and ERP (Enterprise Resource Planning). The OPC was used merely to replace proprietary communication drivers for interfacing SCADA systems and visualization programs to the process peripherals. It is no longer used for transmitting only process data. Entire ERP documents, parameter sets, control sequences, control programs, bill of materials and video signals are transported via OPC. As de facto standard reached broad usage and it is expected as major communication standard in future communication of heterogenic systems that are platform independent [4].

This article deals with Software as a Service delivery model for OPC applications as an integration platform between information systems such as ERP and MES and distributed control systems (DCS). In this model, any consumer can use an OPC client based application via Cloud, but does not control the operating system, hardware or network infrastructure on which it's running [3], as shown in Figure 1. According to Figure 1, MES system can be implemented in Cloud Computing with SaaS. It can use OPC client to get needed information from process control systems (PLC_1,PCL_2). In addition, SCADA systems might be used to transform process data and provide a visualization of process control in a centralized control room (supervisor room).

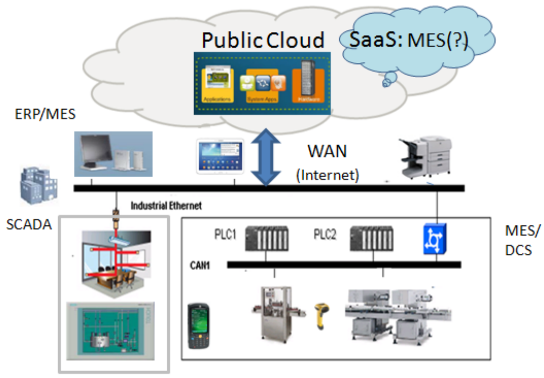

Fig.1. Public Cloud and SaaS for MES systems

II. OPC

Original OPC specification is closely linked to Microsoft's operating system and derived from Microsoft's OLE technology (Object Link Embedding) as "OLE for Process Control". The technology OLE was later superseded by the Component Object Model COM and by Distributed COM (DCOM).DCOM, as an object-oriented RPC system which enables remote procedure calls and communication of COM-based applications over a network. The OPC architecture is based on client/server model. The server part is implemented on the distributed control systems such as PLCs (DCS) and client is implemented on PCs/hosts, providing communication services to application (Visual Basic, C++) , as illustrated in Figure 2 [4]. The communication between OPC client and server is based on IP and DCOM protocol. Two different applications ($Application_1$, $Application_2$) are connected with OC client and have an access to process control data via defined variables and DCOM protocol.

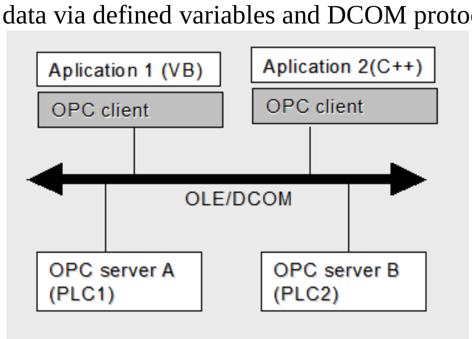

Fig.2. OPC architecture

A. *Data access (DA)*

This service is based on OPC Data Access Server, which is comprised of several objects: the server, the group, and the item. All access to data (reading, writing) is managed via an OPC Group object that "contains" the OPC item. The OPC client can configure the rate that is used by OPC server to provide the data changes. Each item has associated {Value, Quality, Time Stamp}. OPC client uses "OPC Custom Interface" to get access to OPC Groups and items.

B. Alarms & Events (AE)

The service provides mechanisms for notification of OPC clients in case of the occurrence of specified events and alarm conditions. A condition is a named state of the OPC Event Server, or its contained objects. An event is a detectable occurrence at device which is significant to the OPC Server.

C. Historical Data Access (HAD)

It is a dedicated service for simple trend data and raw data storage provided in the form of {Time, Value, Quality} or complex data and analysis such as average values, minimums and maximums etc.

The model of OPC client-server protocol is shown in Figure 3. OPC client can be used by several applications, which could be based on several programming languages, such as C++, or Visual Basic. The client communicates with OPC server via OPC Custom Interface. The process data from devices are transferred to OPC server via given Fieldbus protocols (for example CAN) and mapped to defined variables (Item $_1$,.. Item $_n$) that are accessible by OPC client. The variables can be organized logically to groups (Group$_1$). OPC server has to maintain a data buffer for process data to enable subsequent updating of variables due to different communication capabilities of Fieldbus protocol and OPC communication.

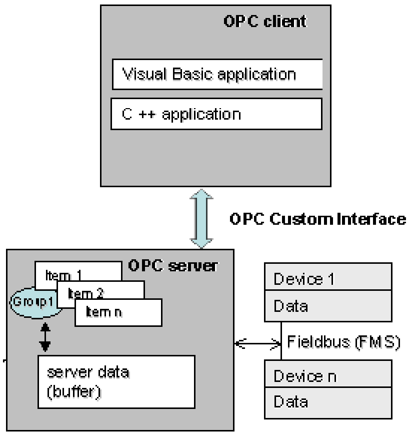

Fig.3. Model of OPC client –server protocol

III. OPC UA

OPC Unified Architecture (OPC UA) is the new, a platform-independent, standard for data communication in process automation. To eliminate dependency on Microsoft, it is based on the open technologies such as the XML and Web services (SOA). Therefore OPC UA will allow the secure and reliable transport of data and information from the factory floor through to the production execution systems such as MES and enterprise resource planning systems -ERP. In contrast to the previous concept with up to three different OPC servers (DA, AE, HDA) only one OPC server, the UA server, is needed. It simplifies semantics and overall implementation [5].

UA servers can be scaled in their functions, size, performance and the platform. The server properties are defined in its profile and can be queried by the client. For embedded systems with limited memory capacities, slim UA servers with a small set of UA services can be implemented. At the company level, in contrast, where memory resources are not that important, very powerful UA servers can be used with the full functionality. The data of a UA server can be published in different formats Binary, XML format and the standard format of an organization or manufacturer. OPC messages are by default transmitted as

1. SOAP messages,
2. XML text,
3. UA binary format for data transfer via TCP/IP.

This enables OPC UA applications to achieve a DCOM-like throughput performance (UA binary) or to enable huge data transfers between ERP/MES and process control (XML). UA Security governs the authentication of clients, ensures data integrity and authorization within OPC communication. The OPC UA uses several Web service security standards such as

- WS-Security,
- WS-Trust and
- WS-Secure Conversation.

OPC client can use security rules like password authentication, the exchange of digital signatures and full encryption of the OPC messages. OPC UA service framework can be considered as a complex due to the following features:

1. Query capabilities –the optional feature to search for data in complex systems
2. Publish mechanism- allows the logical callback to asynchronously send notification messages to a client containing data changes or event data without establishing a real backward channel
3. Connection establishment -requires establishing a secure channel. On top of the secure channel a session has to be created. The secure channel provides security on the transport level

IV. SAAS DELIVERY MODEL FOR OPC

According to the previous chapter, the integration of OPC applications to Cloud does not require significant adjustments in case of OPC UA. However, the original specification with DCOM protocol is not SaaS "friendly" In theory, all relevant OPC applications could be moved to the cloud with limited impact on operations, provided the infrastructure offers adequate reliability, uptime and response time. There are several recommendations which could be taken into consideration OPC in cloud:

1. OPC based applications shall be implemented in Cloud and provided to end users via SaaS delivery model

(MES/HMI/SCADA).

2. Control systems, such as PLCs and Soft PLCs of production machines, production lines are supposed to be implemented locally, as close as possible to controlled systems with appropriate OPC servers.
3. The end devices shall be divided to two groups (GroupA, GroupB), based on their possibilities to work directly with OPC UA and cloud "friendly" protocols, or legacy OPC DCOM specification.
4. All the devices dependent on OPC DCOM (GroupB) have to be connected to Cloud via dedicated OPC gateway in order to translate DCOM traffic to the requested Cloud protocols, such as XML and SOAP.
5. SaaS service has to be established in accordance with the defined SLA (Service Level Agreement), consisting the requested response time TSLA [ms], bandwidth PSLA [Mbps], and number of concurrent users N(n).

Finally, it may be useful to create a simplified model for integration of OPC applications within Cloud and SaaS delivery model. In accordance with above considerations, the proposed model is shown in Figure 4. There are illustrated two group of devices (GroupA, Group B) according to embedded support of Cloud protocols. Typically, the older process control devices, such as PLCs and sensors, do not support newer specification OPC UA and cannot be used easily in SaaS delivery model, which depends on Web based communication. However, DCOM communication would be still possible via IaaS delivery model, which is more preferred in private clouds and do not fit with target SaaS model.

The dedicated OPC gateway is proposed to provide protocol conversion from DCOM to SOAP, or XML messages. In addition, the OPC gateway has to map all assigned OPC Servers to assigned "virtual" OPC UA Proxy servers, which mirror the variables of OPC DA service to OPC client application through cloud. Each OPC based application can read or write data from associated OPC server, either directly from OPC UA server, or indirectly via OPC gateway with protocol conversion DCOM/XML or SOAP.

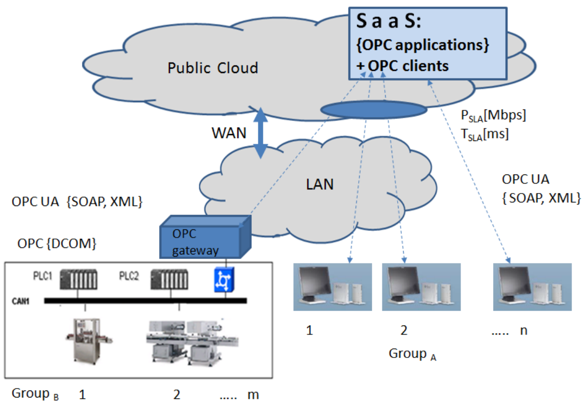

Fig.4. Proposal for delivery model of OPC via SaaS

OPC gateway, as shown in Figure 5, can be described by the expressions (1), (2) and (3), where each server is represented by assigned group of variables (Var_{ij}), containing value ($Value_{ij}$), quality flag and Time stamp. Generally, the gateway can support more logical OPC servers (i=1,..,n). Each OPC server represents a separate device or control system.

$$\{OPC\ UA\ Server\ Proxy_{i}\} = \{OPC\ Server_{i}\} \quad (1)$$

$$OPC\ Server\ i = \{Var_{i1}, Var_{i2}, \ldots, Var_{im}\} \quad (2)$$

$$Var_{i1} = \{Value_{i1}, Quality_{i1}, Time\_Stamp_{i1}\} \quad (3)$$

The conversion from specific OPC DCOM server {OPC Server $_i$} to OPC UA Proxy server is performed by specific driver software. In case of need, the driver can be directly replaced by OPC client translation layer represented by gateway default common OPC server {OPC server $_0$}, with index i=0.

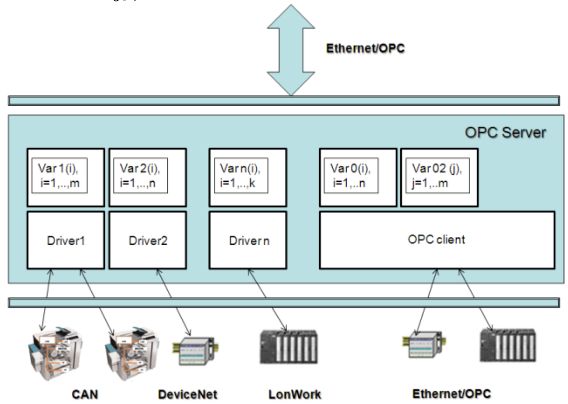

Fig.5. OPC gateway

There are two critical SLAs that are essential for OPC and integration with process control. Firstly, the response time cannot deviate from agreed value (T_{req}) more than defined difference (Δt), see Equation 4. Secondly, an overall bandwidth of WAN connection (P_{SLA}) has to include the required bandwidth of all $Group_A$ devices (n), plus the bandwidth of conversed protocol traffic for $Group_B$ (P_{Proxy}), as defined in (5).

$$T_{SLA} = T_{req} \pm \Delta t \quad (4)$$

$$P_{SLA} = \sum_{i=1}^{n} P_i + P_{Proxy} \quad (5)$$

V. Conclusion

The paper dealt with Cloud computing and possibilities to use Software as a Service for providing of OPC based applications and integration of distributed control systems with process control. Cloud computing shall support visualization applications (SCADA) and information systems used in Manufacturing (MES) as well.

The proposed model is based on Software as a Service delivery model, which is generally used for provisioning of applications via Cloud based protocols, such as REST and SOAP. The main focus was paid on creation of the model, which could be used for integration of the control systems, supporting the newest OPC UA specification and as well as the legacy OPC DCOM specification, which is still broadly used by vendors. Therefore dedicated OPC gateway has been proposed in order to provide the protocol conversion between DCOM protocol and Cloud based protocols (SOAP) that are requested by Cloud computing. OPC gateway has to represent each legacy system, with original OPC specification and DCOM protocol, as logical OPC UA server that is visible for Cloud and OPC UA client implemented via SaaS.

In addition, well defined Service Level Agreements regarding WAN connection to the Cloud provider are prerequisite for provisioning of OPC based application within Cloud Computing and Software as a Service delivery model.

REFERENCES

[1] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond,M. Morrow, "Blueprint for the Intercloud –Protocols and Formats for Cloud Computing," Internet and Web Applications and Services, 2009. ICIW '09. Fourth International Conference, p. 1-3.

[2] M. Giriraj, S. Muthu, "A Cloud Computing Methodology for Industrial Automation and Manufacturing Execution System," Journal of Theoretical and Applied Information Technology, Publisher, 2005 - 2013 JATIT

[3] T. Seitz, "SAP ERP in the Cloud," Oracle WhitePaper, Available: http://www.oracle.com/us/solutions/sap/database/sap-erp-cloud-352626.pdf

[4] N., P. MAHALIK, "Fieldbus technology, Industrial network standard for Real – Time Distributed Control", Springer, 2003

[5] P. Peniak, M. Franekova, "Virtualization MES systems integrated within distributed control systems," ATP Journal Plus - No. 2 2013, ISSN 1336-5010, pp. 74-77.

AUTHOR'S PROFILE

P. Peniak

was born in Pov. Bystrica, Slovakia on May 10, 1968. He received Ph D. in Telecommunication from Univerzity of Žilina in 1999. He is currently host assistant at Department of Control and Information Systems, Faculty of Electrical Engineering, University of Zilina. His research interests are MES systems and their applications for mission critical systems and process control.