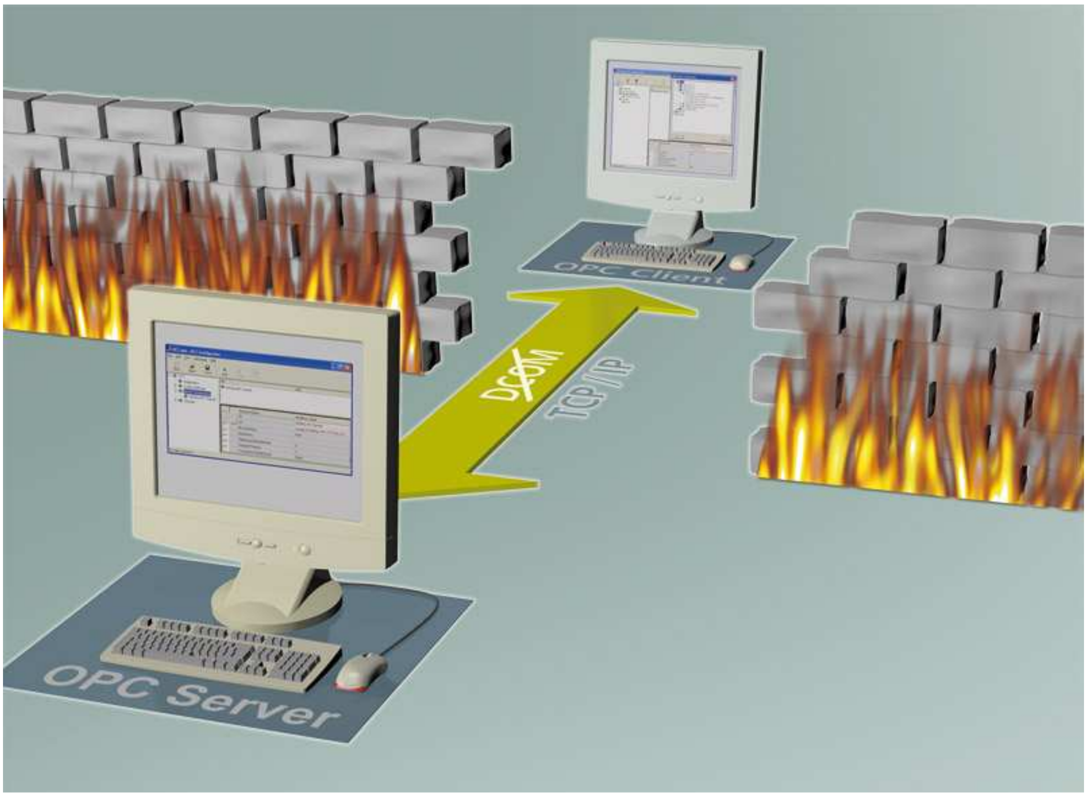

DCOM security settings present a major challenge for OPC communication across PCs

Part 3 of 8:

Ten Reasons for OPC UA

The second article of the OPC UA Series (SPS-Magazin, Issue 4) described the origin, development and objectives of OPC UA. The third part of our series of articles outlines ten key reasons that have led to the development of an entirely new technology generation, the OPC Unified Architecture. They are based, on the one hand, on the experiences with OPC as well as the technological changes and trends over the past 14+ years since the beginning of OPC technology. On the other hand, they also take into account many wishes and suggestions from OPC vendors and users.

1. Discontinuation of COM/DCOM

The automated exchange of data between Classic OPC applications is based on Microsoft's COM technology. As the Windows operating system rapidly became widely used all over the world and promoted the use of Windows computers in automation, it provided ideal conditions for driving the widespread adoption of OPC technology. In early 2002, Microsoft launched its new .NET framework and announced the discontinuation of DCOM. This did not mean that future Windows operating system versions would not support DCOM but, as a result of the discontinuation, the base technology of Classic OPC would not be further developed and sooner or later become obsolete.

2. DCOM limitations

With COM/DCOM, Microsoft introduced a set of features in the 90ies that were highly appreciated by both end users on home computers in the non-industrial segment and professional users who used Windows computers as automation components in industrial applications. These features include copy and paste, drag and drop, linking and embedding. DCOM also offers the complete communication infrastructure with all the necessary security, such as authentication, authorization and encryption. DCOM Security controls the access rights to data and programs on remote computers. But DCOM Security at the same time also presents a major challenge for setup engineers, system integrators and developers managing projects that involve OPC communication across PCs. Setting up DCOM Security correctly is a very complicated task and takes a lot of expertise. As a result, setup engineers and system integrators routinely choose to speed up the process by granting very broad access rights on all networked OPC computers and thus largely disabling the protection from unauthorized remote access. This shortcut collides with IT security requirements and, in the long run, risks damage caused by negligence or sabotage. DCOM Security settings are often a showstopper for the otherwise very easy to configure OPC communication relationships.

3. OPC communication across firewalls

The possibilities of OPC communication across computer boundaries were recognized very early in the automation industry. And this is where DCOM again limits Classic OPC communication. DCOM requires multiple ports for establishing a connection, for authentication, for transmitting data, and for a number of other services. Consequently, many ports have to be opened in a firewall to allow DCOM communication across it. Every open port in a firewall is a security gap and provides a potential target for hacker attacks. OPC Tunneling is a widely accepted strategy to solve DCOM's limitation of the use of Classic OPC products.

4. Use of OPC on non-Windows platforms

The 'omnipresence' of Microsoft platforms in industrial applications with DCOM as a component of the operating system was a major factor in promoting the rapid acceptance of Classic OPC. At the same time, integration concepts with OPC failed in areas in which other operating systems are employed. The IT industry, for example, often uses Unix or Linux systems. Automation, too, has application areas that categorically refuse to implement Windows operating systems. The embedded area is another area in which Windows hardly features (except for Windows CE or embedded XP). Here, complex applications are embedded directly in field devices, PLCs, operator panels and other devices running VxWorks, QNX, embedded Linux, RTOS or other embedded operating systems without DCOM. Integration concepts with OPC are doomed to fail in those areas because OPC needs DCOM as the technological basis, and this basis is missing in embedded systems.

5. High-performance OPC communication via Web Services

With the release of the OPC XML-DA Specification in 2003, the OPC Foundation for the first time showed a way out of the dependency on Windows platforms and the limitations caused by DCOM. Today, many OPC XML-DA products demonstrate the possibilities of Web Services based OPC technology. The data throughput of XML-DA communication, however, is slower by a factor between five and seven compared to that of DCOM DA communication.

This performance is significantly too slow for many automation tasks. The possibilities offered by Web Services based OPC communication are promising, but a much higher data transfer performance has to be achieved.

6. Unified data model

Until now, it takes three different OPC servers in Classic OPC – Data Access, Alarms & Events and Historical Data Access – to acquire, for example, the current value of a temperature sensor, the event of the temperature exceeding a preset limit and the historical average of the temperature. This makes it very time consuming for users to access process data, event and historical data in such different ways. Unifying the three object models would make things a lot simpler not only for the OPC product vendors, but also for system integrators and users.

7. Support of complex data structures

One of the main applications of OPC is the operation and monitoring of devices that are networked through serial communication protocols or fieldbuses. To configure devices, data types are needed that allow an OPC client to write complex data structures, including the meanings of the data structure elements, to a device via an OPC server. With the Complex Data Specification, the OPC Foundation has created a possibility to describe complex data structures. However, the vast majority of Classic OPC products on the market today has not implemented the Complex Data Specification, apart from very few exceptions.

8. Process data communication without data loss

Data Access was originally defined to cyclically inform client applications of the current state of process data. If disturbances occur in the physical communication link between an OPC client and a remote OPC server, the communication will be broken according to the Data Access Specification. Data changes that have occurred while communication was broken could not be transferred to the OPC client and were lost. This data loss is not critical with most Data Access projects, such as trend recording, process monitoring or process visualization. But OPC has increasingly penetrated application areas where requirements are more critical. For example, OPC technology has become established in areas such as the chemical or pharmaceutical industries, where data must be seamlessly recorded. What made this possible is that vendors have implemented specific extensions. They are based on connection monitoring systems that ensure a fast detection of broken communication, automatic reconnection if communication breaks, data buffering in Data Access servers, redundancy, and store & forward concepts. Useful as these extensions are, they have not been defined in the Classic OPC specifications and vary from vendor to vendor.

9. Increased protection against unauthorized data access

As a result of the growing trend towards Ethernet based communication in automation, the automation and office networks are intertwining. While opening up new possibilities of vertical integration, this type of integration concept involves new security risks. OPC is also increasingly used in remote maintenance and remote control concepts. Here, again, more stringent requirements must be met regarding the security of the installation from unauthorized access from the outside. With rising cybercrime, spying and sabotage, IT security is growing more and more important – and so are the requirements on security when using OPC. Without the proprietary precautions developed by vendors, Classic OPC cannot meet these security requirements.

10. Support of method calls

In many applications, it is not only the reading and writing of values that is important, but also the execution of commands, such as starting or stopping a drive or downloading a file to a device. The OPC Commands Specification defines possibilities to execute commands, but is only available as a draft version and was not taken into account for Classic OPC.

In Part 4 of the OPC UA Series we will look at the OPC UA Specifications in detail.

www.softing-ia.com

Authors:
Peter Seeberg, Product Marketing Manager, Softing Industrial Automation GmbH
Jürgen Lange, Area Account Manager Embedded Technology Products, Softing Industrial Automation GmbH

OPC Day Europe 2012

The OPC Foundation will be holding this year's OPC Day Europe on May 16, 2012. The event will take place at Endress+Hauser in Reinach, Switzerland, and focus on the use of OPC UA in process automation.

Series of Articles in SPS-MAGAZIN

Issue	**Date of Publication**	**Topic**
3	24.02.2012	OPC UA Status
4	16.03.2012	OPC UA: Origin, Development and Objectives
HMI Special	13.04.2012	Ten Reasons for OPC UA
5	10.05.2012	OPC UA Specifications
6	01.06.2012	OPC UA Companion Standards
7	30.06.2012	OPC UA Compliance Test
8	27.07.2012	OPC UA Toolkits
9	31.08.2012	OPC UA Outlook