zusätzlich zur DCOM-Security implementieren oder kann nur OPC-Security anbieten.

Die *OPC Security Specification* hat in der Vergangenheit keine große Verbreitung erfahren. Aus diesem Grund wird auf eine umfangreiche Darstellung der Spezifikationsgegenstände zugunsten anderer Punkte verzichtet. Die Bedeutung von Sicherheitsvorkehrungen hat in den letzten Jahren sehr zugenommen. Der auf die DCOM Security limitierte Schutz vor unautorisierten Zugriffen auf OPC Server und deren Daten wurde dementsprechend zunehmend als Schwachpunkt der Classic OPC Technologie gewertet. Security spielt bei OPC UA daher von Anfang an eine sehr große Rolle. Dem Thema Security wird in der *OPC UA Specification* mit einem eigenen Teil 2 eine hohe Bedeutung beigemessen.

2.3 OPC Unified Architecture

2.3.1 Einführung

2.3.1.1 Never Touch a Running System – Wofür ein neues OPC?

OPC ist unbestritten einer der erfolgreichsten De-facto-Standards seit der Erfindung des Computers. Mit weit über 20.000 OPC Produkten von über 3.500 verschiedenen Herstellern und weltweit vielen Millionen Installationen OPC-basierter Produkte in der Fertigungs- und Prozessindustrie, Gebäudeautomation und vielen anderen Branchen ist OPC der unangefochtene Standard für den interoperablen Austausch von Daten zwischen Softwareanwendungen unterschiedlicher Hersteller. OPC ermöglicht die Automatisierung des Datentransfers von einem Anlagenbereich zu einem anderen. OPC Schnittstellen bilden eine komfortable und leistungsfähige Verbindung von Automatisierungskomponenten mit Steuerungshardware und Feldgeräten und überbrücken die Unterschiede heterogener Automatisierungswelten. Die OPC-Technologie wird heute praktisch für alle Arten der Datenerfassung, der vertikalen und horizontalen Datenintegration und des Datenmanagements eingesetzt. OPC ist das entscheidende Bindeglied für HMI/SCADA-Systeme zur Prozessvisualisierung, für Prozessleitsysteme und PC-basierte SPSen zur Steuerung von Prozessen ebenso wie für MES- und ERP-Systeme zur Anbindung an unterlagerte Automatisierungskomponenten. Über die OPC-Schnittstelle werden längst nicht mehr nur Prozessdaten oder einzelne Parameter übertragen. Ganze Warenwirtschaftsdokumente, Parametersätze, Steuerungssequenzen oder Antriebsprogramme werden über OPC transportiert.

OPC ist weltweit etabliert, erfolgreich und bewährt. Was also motiviert die OPC Foundation zu der Einführung der *OPC Unified Architecture*? Ist OPC UA ein „neues OPC"? Wird OPC UA Classic OPC ablösen? Welche Vorteile gegenüber Classic OPC bietet die neue Unified Architecture? Diese und weitere Fragen werden in den folgenden Abschnitten beleuchtet.

2.3.1.2 Zehn Gründe für OPC UA

Basierend auf den OPC-Erfahrungen, technologischen Veränderungen und Trends in den mehr als 12 zurückliegenden Jahren seit den Anfängen der OPC-Technologie sowie vielen Wünschen und Anregungen von OPC-Herstellern und -Anwendern werden nachfolgend zehn Gründe vorgestellt, die die Motivation für eine ganz neue Technologiegeneration, die *OPC Unified Architecture* bilden:

1. Abkündigung von COM/DCOM

Microsofts Basistechnologie COM bildet die Grundlage für den automatisierten Austausch von Daten zwischen Classic OPC-Anwendungen. Distributed COM (DCOM) ermöglicht die rechnerübergreifende Kommunikation zwischen OPC-Clients und OPC-Servern. Die weltweit rasante Ausbreitung des Windows Betriebssystems und der Einzug von Windows-Rechnern in die Automatisierung waren ideale Bedingungen für eine schnelle Verbreitung der OPC-Technologie. Anfang 2002 bringt die Firma Microsoft ihr neues .NET Framework auf den Markt und kündigt DCOM ab. Dies bedeutet keineswegs, dass DCOM nicht weiterhin in zukünftigen Versionen von Windows Betriebssystemen unterstützt würde – die Voraussetzung für den Einsatz existierender OPC-Komponenten wie auch anderer DCOM-Anwendungen bleiben erhalten – doch resultierte daraus, dass die Basistechnologie von Classic OPC nicht mehr weiterentwickelt und damit früher oder später unweigerlich veraltet sein würde. Mit .NET führte Microsoft ein Framework ein, welches XML und Web Services als Basistechnologie verwendet. Mit *OPC XML-DA* brachte auch die OPC Foundation 2003 eine erste OPC-Spezifikation auf den Markt, die XML und Web Services anstelle von COM/DCOM verwendet. Die Erfahrungen bei der Implementierung von XML-DA-Komponenten unter Windows, aber auch Linux und anderen Nicht-Windows Betriebssystemen waren vielversprechend, wenngleich die OPC-Kommunikation zunächst auf Prozessdaten (*Data Access*) beschränkt war und die Datendurchsatzgeschwindigkeit des XML-Protokolls SOAP für viele Automatisierungsaufgaben unzureichend war. Doch ein Anfang war gemacht. Der erste Schritt in eine OPC-Technologie ohne COM/DCOM war erfolgt. Doch war damit auch der Bedarf an einer leistungsfähigen Möglichkeit geweckt, Prozessdaten, Ereignisse und auch historische Daten auszutauschen.

2. Grenzen von DCOM

Mit COM/DCOM führte Microsoft in den 90-er Jahren eine Reihe von Möglichkeiten ein, die schnell von Anwendern geschätzt wurden, sei es im privaten Bereich auf dem Heimcomputer oder im industriellen Bereich auf dem Windows-Rechner als Automatisierungskomponente: Copy&Paste, Drag&Drop, Linking&Embedding. Technologien, die es ermöglichen, Daten komfortabel zwischen verschiedenen Windows-Anwendungen auszutauschen, Grafiken oder Rechenkalkulationen in Textdokumenten einzubetten und vieles mehr. Mit DCOM steht auch die komplette Kommunikationsinfrastruktur mit allen erforderlichen Sicherheitsdiensten wie Authentisierung, Autorisierung oder Verschlüsselung zur Verfügung. DCOM-Sicherheitseinstellungen regeln die Befugnisse für den Zugriff auf Daten und Programme auf einem anderen Rechner. Doch gleichzeitig erweisen sich gerade diese DCOM-Sicherheitseinstellungen als große Herausforderung für Inbetriebnehmer und Betreiber von Projekten mit rechnerübergreifender OPC-Kommunikation. Die Einstellung funktionsfähiger DCOM-Einstellungen ist sehr anspruchsvoll und erfordert ein hohes Maß an Spezialwissen. So müssen unter anderem die Zugriffsrechte, die einem Benutzer beim Windows Login erteilt werden und die DCOM-Sicherheitseinstellungen aufeinander abgestimmt werden. Sehr häufig suchen Inbetriebnehmer und Systemintegratoren den schnellen Erfolg, indem sie auf allen vernetzten OPC-Rechnern sehr großzügige Zugriffsrechte vergeben und damit den Schutz vor unerwünschten Zugriffen von außen entfernen. Ein solches Vorgehen kollidiert mit den Sicherheitsanforderungen der IT-Abteilungen und riskiert letztendlich Schäden durch Unachtsamkeit oder Sabotage. DCOM-Sicherheitseinstellungen sind häufig ein „Show Stopper" der sonst sehr einfach konfigurierbaren OPC-Kommunikationsbeziehungen; sie sind der Spitzenreiter in der Statistik der Gründe für Supportanfragen bei Herstellern von OPC-Produkten. Eine weitere Grenze von DCOM stellen sehr lange und nicht konfigurierbare Zeiten für die Erkennung von Unterbrechungen einer Kommunikationsverbindung dar. Wird die Verbindung zwischen einem OPC-Client und einem OPC-Server auf einem entfernten Rechner, der beispielsweise die Daten einer SPS akquiriert, unterbrochen, können viele Sekunden vergehen, bis der OPC-Client über diese Störung informiert wird. Derartige Reaktionszeiten sind für die meisten industriellen Anwendungen unzureichend. Die Grenzen von DCOM sind die wesentlichen Schwachpunkte der sonst so erfolgreichen OPC-Technologie. Eine populäre und weit verbreitete Strategie zur Umgehung der DCOM Begrenzungen ist das sogenannte Tunnelling, welches DCOM komplett umgeht und eigene Sicherheitsvorkehrungen sowie eine eigene Verbindungsüberwachung vorsieht (siehe Abschnitt 4.4.3).

3. Firewall übergreifende OPC-Kommunikation

Sehr früh sind die Möglichkeiten einer rechnerübergreifenden OPC-Kommunikation in Automatisierungsprojekten genutzt worden. Die Vorteile liegen auf der Hand: der Zugriff auf entfernte OPC-Server ist absolut transparent, d. h. aus Sicht des Anwenders gibt es keinerlei Unterschiede zwischen dem Zugriff auf

lokale OPC-Daten oder auf OPC-Daten auf entfernten Rechnern. Mit geeigneten DCOM-Sicherheitseinstellungen können viele Kommunikationsteilnehmer innerhalb eines Intranets unkompliziert und komfortabel über OPC vernetzt werden. Doch wie sieht es mit einer OPC-Kommunikation außerhalb eines Intranets, d. h. über das Internet aus? Der Austausch von Daten in Automatisierungsprojekten über das Internet ist nur bei entsprechenden Sicherheitsvorkehrungen sinnvoll. Der Einsatz einer Firewall für den Schutz vor unautorisiertem Zugriff auf die Daten einer Automatisierungsanlage ist hierbei absolut zwingend. An dieser Stelle kommt es zu einer weiteren Begrenzung der Classic OPC-Kommunikation durch DCOM: DCOM benötigt für die Ausführung der Kommunikationsdienste sehr viele Ports. Ports sind ein Teil einer Telegrammadresse, der Datensegmente einem Netzwerkprotokoll zuordnet. Fest definierte Ports sind z. B. Port 80 für HTTP, Port 443 für HTTPS oder Port 21 für FTP. DCOM benötigt mehrere Ports zum Aufbau einer Verbindung, zur Authentisierung, zum Senden von Daten und für einige andere Dienste. Die zu nutzenden Ports werden von DCOM zufällig festgelegt. Wenn diese Ports nicht verfügbar sind, sucht DCOM automatisch andere. Für eine DCOM-Kommunikation über eine Firewall müssen somit eine Vielzahl von Ports in der Firewall geöffnet werden. Jeder geöffnete Port in einer Firewall ist ein potenzielles Angriffsziel für Hacker und stellt eine Sicherheitslücke dar. Kommen Network Address Translation (NAT) basierte Firewalls zum Einsatz, lässt sich gar keine OPC-Kommunikationsverbindung mehr herstellen, da DCOM mit der Adressumsetzung nicht klarkommt. Es lassen sich somit keine sicheren DCOM-OPC-Verbindungen über die Grenzen einer Firewall herstellen. Zur Überwindung dieser DCOM-Begrenzung beim Einsatz von Classic OPC-Produkten, ist OPC Tunnelling eine anerkannte Strategie (siehe Abschnitt 4.4.3).

4. Einsatz von OPC auf Nicht-Windows Plattformen

DCOM ist in allen Windows Betriebssystemen enthalten. Die beinahe „Allgegenwart" der Microsoft-Plattformen auch in der Industrie ist einerseits einer der Gründe für die rasante Verbreitung von OPC. Andererseits begrenzt DCOM den Einsatz der OPC-Technologie auf Windows. In weiten Bereichen der Industrie stellt dies keinen Begrenzungsfaktor für die Verbreitung von OPC dar. Doch gibt es Branchen, wie z. B. die Informationstechnologie, in welchen Windows Betriebssysteme eher selten anzutreffen sind. In der IT sind vielfach UNIX- oder Linux-Systeme im Einsatz. Aber auch im Bereich der Automatisierung gibt es Branchen, in welchen der Einsatz von Windows als Betriebssystem kategorisch ausgeschlossen wird. Sehr häufig sind dies konservative Branchen wie Chemie oder Pharmazie, in der die Kurzlebigkeit der Windows Betriebssystemversionen, die zum Teil erheblichen Unterschiede zwischen Windows NT, 2000, XP, Vista und Windows 7, aber auch sicherheitsspezifische Faktoren ein Hinderungsgrund für den Einsatz der Microsoft-Betriebssysteme darstellen. Ein dritter Bereich, in dem Windows mit Ausnahme von Windows CE oder embedded XP praktisch keine Rolle spielt, ist der Embedded-Bereich. In diesem Sektor ist seit einigen Jahren ein starker Trend festzustellen: Automatisierungsgeräte werden immer kleiner und

kompakter, gleichzeitig aber auch immer intelligenter und mit immer leistungsfähigeren Prozessoren ausgestattet. Immer komplexere Anwendungen werden direkt in Feldgeräte, Steuerungen, Bedien Panels u. a. Geräte eingebettet, die mit VxWorks, QNX, embedded Linux, RTOS oder anderen Embedded-Betriebssystemen ohne DCOM ausgestattet sind. Integrationskonzepte mit OPC scheitern hier, da die für OPC erforderliche Technologiebasis DCOM in den Embedded-Systemen fehlt.

5. Leistungsfähige OPC-Kommunikation über Web Services

Mit der 2003 veröffentlichten *OPC XML-DA Specification* zeigte die OPC Foundation erstmalig einen Weg aus der Abhängigkeit von Windowsplattformen und aus den Begrenzungen durch DCOM auf. Zahlreiche *OPC XML-DA* Produkte demonstrieren heute die Möglichkeiten einer Web Service basierten OPC-Technologie. In der Gebäudeautomatisierung, Energietechnik, Prozessindustrie und anderen Branchen werden XML-DA-Produkte auf Unix-Rechnern, Linux-Servern, aber auch eingebettet in Feldgeräten mit Linux oder anderen Embedded-Betriebssystemen eingesetzt. Unter Einsatz eines Wrappers, der die XML-DA-Daten und -Dienste auf DCOM DA abbildet und umgekehrt, können XML-DA-Produkte problemlos zusammen mit DCOM-OPC-Produkten eingesetzt werden. XML-DA ermöglicht somit Plattform-übergreifende Integrationskonzepte unter Einsatz von OPC. Für den Transport der Daten wird bei XML-DA das XML-Protokoll SOAP genutzt. Die OPC-Kommunikation erfolgt dabei durch Austausch von http-Telegrammen mit in Textformat kodierten OPC-Daten. Für das Lesen und Schreiben von OPC-Daten ist daher ein zeitaufwändiger Prozess für das Erstellen von http-Telegrammen und die Konvertierung der OPC-Daten in Textformat bzw. das Auspacken von http-Telegrammen und Rückkonvertieren von Text in das ursprüngliche OPC-Datenformat erforderlich. Die Datendurchsatzgeschwindigkeit einer XML-DA-Kommunikation ist um den Faktor fünf bis sieben geringer als die Datendurchsatzgeschwindigkeit einer DCOM DA-Kommunikation. Dies ist für viele Automatisierungsaufgaben deutlich zu wenig. Die Möglichkeiten einer Web Service-basierten OPC-Kommunikation sind vielversprechend, gleichzeitig besteht die Notwendigkeit einer viel höheren Datenübertragungs-Performance.

6. Einheitliches Datenmodell

Eine Anforderung des Marktes an die OPC Foundation, insbesondere aus dem Bereich der Prozessindustrie und der Gebäudeautomation, ist eine bessere Integration von Alarmen und historischen Daten in den Adressraum eines *Data Access Servers*. Bis heute sind drei verschiedene OPC-Server – *Data Access*, *Alarms&Events* und *Historical Data Access* erforderlich, um z. B. den aktuellen Wert eines Temperatursensors, das Ereignis einer Temperaturüberschreitung und den historischen Mittelwert der Temperatur zu erfassen. Zudem sind die Objektmodelle von DA, AE und HDA sehr verschieden. Wenngleich es möglich ist, alle drei Objektmodelle in einer OPC-Anwendung zu implementieren, ist dennoch der Zugriff auf

den Wert eines DA-, AE- oder HDA-Objektes sehr verschieden. Der Zugriff auf Prozessdaten, Ereignisse und historische Daten auf so unterschiedliche Art und Weise ist für den Anwender sehr zeitaufwändig. Eine Vereinheitlichung der drei Objektmodelle würde zu einer deutlichen Vereinfachung sowohl für Hersteller von OPC-Produkten, als auch für System-Integratoren und Anwender führen.

7. Unterstützung komplexer Datenstrukturen

Bereits in den frühen Jahren der OPC-Technologie wurden Wünsche an die OPC Foundation herangetragen, neben den einfachen Datentypen auch strukturierte Datentypen zu unterstützen. Eines der Haupteinsatzgebiete von OPC ist das Bedienen und Beobachten von Geräten, die über serielle Kommunikationsprotokolle oder Feldbusse vernetzt sind. Für das Auslesen von Prozessdaten und Zustandsinformationen bzw. das Schreiben einzelner Bedienparameter sind einfache Datentypen wie Byte, Integer, Real oder Arrays von diesen absolut ausreichend. Für die Konfiguration von Geräten werden jedoch Datentypen benötigt, mit welchen komplexere Datenstrukturen inklusive der Bedeutung der Datenstrukturelemente, von einem OPC-Client über einen OPC-Server an ein Gerät geschrieben werden können. Die Struktur dieser Konfigurationsdaten hängt von dem Gerätetyp und dem Hersteller ab. Viele Feldbus-Organisationen wie die PROFIBUS Nutzerorganisation, Fieldbus FOUNDATION, CAN in Automation und andere, haben eigene Gerätebeschreibungsformate für eine einheitliche, standardisierte Möglichkeit der Gerätekonfiguration definiert. Mit der *Complex Data Specification* (siehe Abschnitt 2.2.8) hat die OPC Foundation eine Möglichkeit geschaffen, komplexe Datenstrukturen zu beschreiben, einzelne Komponenten mit ihrem (einfachen) Datentyp zu unterscheiden und die Zusammenhänge zwischen den einzelnen Komponenten abzubilden. Damit ist es möglich, beispielsweise Gerätebeschreibungen von Feldbusorganisationen abzubilden. OPC-Produkte, wie sie heute am Markt vorzufinden sind, haben jedoch bis auf sehr wenige Ausnahmen die *Complex Data Specification* nicht implementiert. Dies ist wahrscheinlich darauf zurückzuführen, dass diese Spezifikation sehr spät, d. h. mehrere Jahre nach Einführung der *Data Access Specification* definiert wurde und zum Zeitpunkt der Veröffentlichung bereits Tausende OPC-Produkte installiert waren. Der Wunsch nach realer Unterstützung komplexer Datenstrukturen sowie einer Möglichkeit, weiterführende Beschreibungen zu einem Datenpunkt zu hinterlegen, bleibt somit weiterhin bestehen.

8. Prozessdatenkommunikation ohne Datenverlust

Data Access wurde ursprünglich definiert, um Client-Anwendungen zyklisch den aktuellen Zustand von Prozessdaten mitzuteilen. Die Frequenz dieses zyklischen Aktualisierungsvorgangs konfiguriert der Anwender über die *UpdateRate*. Ändern sich die Prozessdaten häufiger als sie vom Server gemäß der konfigurierten *UpdateRate* an den OPC-Client transferiert werden (können), kommt es zum Informationsverlust. Störungen in der physikalischen Kommunikationsverbindung

zwischen einem OPC-Client und einem entfernten OPC-Server führen gemäß der *Data Access Specification* zum Abbruch der Kommunikation. Eine Fortsetzung der Kommunikation zwischen Client und Server nach Beseitigung der physikalischen Kommunikationsstörung kann nur durch ein komplettes Neuaufsetzen der OPC-Verbindung erfolgen. Datenänderungen, die sich während der Kommunikationsstörung ergeben haben, konnten nicht an den OPC-Client transferiert werden und sind verloren gegangen. Für die meisten *Data Access* Projekte, wie z. B. zur Aufzeichnung von Trends, Beobachtung von Prozessen oder zur Prozessvisualisierung sind derlei Datenlücken unkritisch. OPC wird aber zunehmend in Anwendungsgebieten mit kritischeren Anforderungen eingesetzt. So hat sich beispielsweise die OPC-Technologie in Branchen wie der Chemie oder Pharmazie etabliert, in welchen Daten lückenlos aufgezeichnet werden müssen. Möglich ist dies, indem Hersteller-spezifische Erweiterungen implementiert wurden. Dies sind Verbindungsüberwachungen mit schneller Erkennung von Kommunikationsabbrüchen, automatischer Wiederaufbau nach Verbindungsabbrüchen, Datenpufferung in *Data Access Servern*, Store&Forward- sowie Redundanzkonzepte. Da diese durchaus sinnvollen Erweiterungen aber nicht in den Classic OPC-Spezifikationen festgelegt sind, sondern von Hersteller zu Hersteller unterschiedlich realisiert wurden, besteht ein großer Bedarf an einer allgemeingültigen, interoperablen OPC-Festlegung.

9. Mehr Schutz vor unautorisiertem Datenzugang

Im Zuge des Trends zu immer mehr Ethernet-basierter Kommunikation in der Automatisierung, wachsen Automatisierungsnetzwerke und Office-Netzwerke immer mehr zusammen. Dies eröffnet auf der einen Seite neue Möglichkeiten der vertikalen Integration. Die Daten auf der Prozessebene können von einem OPC-Server zur Verfügung gestellt werden und ohne zusätzliche Verkabelung von MS Excel mit einem OPC Client Plug-In dargestellt oder von einer OPC Client Applikation in einer Datenbank archiviert werden. Auf der anderen Seite bringen derartige Integrationskonzepte neue Sicherheitsrisiken mit sich. Werden keine speziellen Vorkehrungen getroffen, kann die Sicherheit einer Anlage durch unbefugte und ungewollte Zugriffe oder Datenmanipulationen gefährdet werden. OPC wird auch immer häufiger in Fernwartungs- und Fernbedienungskonzepten eingesetzt. Auch hier sind erhöhte Anforderungen an die Sicherheit der Anlagen vor unerwünschten Eingriffen von außen zu treffen. Im Zuge der Zunahme von Cyberkriminalität, Spionage und Sabotage spielt IT-Sicherheit heute eine immer bedeutendere Rolle und sind dementsprechend auch beim Einsatz von OPC höhere Anforderungen an die Sicherheit gestellt. Ohne herstellerspezifische Vorkehrungen kann Classic OPC diese Sicherheitsanforderungen nicht erfüllen.

10. Unterstützung von Methodenaufrufen

In vielen Anwendungen ist nicht nur das Lesen und Schreiben von Werten wichtig, sondern auch das Abarbeiten von Befehlen wie z. B. das Starten oder Stoppen eines Antriebs oder die Durchführung des Downloads einer Datei auf ein Gerät. Die *OPC Commands Specification* (siehe Abschnitt 2.2.7) definiert Möglichkeiten des Ausführens von Kommandos bzw. des Aufrufs von Methoden wie z. B. das Starten oder Stoppen eines Antriebs, das Abarbeiten eines Programms etc. Die *OPC Commands Specification* existiert als Draftversion und wurde bis zu dem Beginn der Arbeiten an der *OPC Unified Architecture Specification* nicht mehr fertig gestellt. Der Bedarf an Methodenaufrufen über die OPC-Schnittstelle bleibt bestehen und ist als Anforderung an die neue *OPC UA Specification* eingegangen.

2.3.1.3 Entstehung und Ziele von OPC UA

Erste Überlegungen zu der neuen OPC-Architektur gehen bereits auf das Jahr 2003 zurück, als die *Alarms&Events* Arbeitsgruppe der OPC-Foundation an der nächsten Generation der AE Spezifikation und ihrer Migration zu Web Services arbeitete. Diese Überlegungen führten Ende 2003 zur Gründung einer ganz neuen Arbeitsgruppe. Zunächst war es das primäre Ziel dieser Arbeitsgruppe, den Zugriff auf Prozessdaten (*Data Access*), Alarme und Ereignisse (*Alarms&Events*) sowie historische Daten (*Historical Data Access*) so auf Web Services umzusetzen, dass dieser auf einheitliche Art und Weise möglich ist. Eine Architektur der Vereinheitlichung, die *Unified Architecture*, kurz OPC UA, war geboren. Mitarbeiter von 30 zum Teil führenden Industrieunternehmen arbeiteten unter der Regie der OPC Foundation über einen Zeitraum von über fünf Jahren in unzählbaren Stunden an der neuen OPC-Architektur. Neben der Portierung von Classic OPC auf Web Services und des „Unifiyings" von DA, AE und HDA, sind viele weitere Anforderungen an das neue OPC UA hinzugekommen, die die OPC Foundation in Marktuntersuchungen und zahlreichen Befragungen von OPC-Anwendern, System-Integratoren und Herstellern in Erfahrung gebracht hat. Daraus resultierend hat die OPC Foundation folgende Leitlinien und Hauptziele definiert:

- **„Keep it simple":**
  Die Anwendung der UA Technologie in Form von UA Komponenten soll trotz der Vielzahl an funktionalen Anforderungen und Komplexität einfach sein.
- **„Evolution" statt „Revolution":**
  Terminologie, Objektmodelle und die wesentlichen Kommunikationsprinzipien von Classic OPC sollten weiterbestehen; Investitionen in die Entwicklung von Classic OPC-Produkten sollen durch deren weitere Verwendbarkeit geschützt werden.
- **Plattformunabhängigkeit und Skalierbarkeit:**
  nicht mehr DCOM als Technologiebasis, sondern eine Service-orientierte Architektur (SOA) für den Einsatz der OPC-Technologie auf der IT Ebene oder in Embedded-Systemen

- **Zugriffsschutz:**
  Schutz vor Spionage, Sabotage und Fehlern aufgrund von unachtsamen Verhaltens
- **Datensicherheit:**
  robuste Architektur, zuverlässige Kommunikationsmechanismen, Redundanzkonzepte und weitere Maßnahmen zur Vermeidung von Datenverlust
- **starke Performance:**
  schlanker, leistungsfähiger Datentransport zur Erfüllung höchster Performanceanforderungen

Mit OPC UA wurde nicht nur eine neue Version des Schnittstellenstandards OPC geplant. OPC UA ist die Vision von einer „globalen" Interoperabilität, einem standardisierten Datenaustausch zwischen Softwareanwendungen, unabhängig von welchem Hersteller sie stammen, in welcher Programmiersprache sie entwickelt wurden, auf welchem Betriebssystem sie laufen oder an welchem Ort sie sich befinden.

2.3.1.4 Neue Möglichkeiten mit OPC UA

OPC UA ergänzt den existierenden OPC-Industriestandard um wesentliche Eigenschaften wie Plattformunabhängigkeit, Skalierbarkeit, Hochverfügbarkeit, Internetfähigkeit und weitere. Insbesondere die Plattformunabhängigkeit und Skalierbarkeit ermöglichen die Realisierung ganz neuer, kostensparender Automatisierungskonzepte. Embedded-Feldgeräte, Prozessleitsysteme, Speicherprogrammierbare Steuerungen, Gateways oder Operator Panels können schlanke OPC UA Server Implementierungen enthalten, die direkt auf Betriebssysteme wie embedded Linux, VxWorks, QNX, RTOS oder andere portiert wurden. Ein separater Windows PC für den OPC-Server, der bisher den Zugang zu den Daten auf Geräten mit Nicht-Windows Plattformen bot, ist nicht mehr erforderlich.

OPC UA Komponenten können aber auch in informationstechnischen Systemen eingesetzt werden, in Enterprise Resource Planning (ERP), Produktionsplanungs- und -steuerungs-Software und anderen E-Business-Anwendungen mit Unix-Betriebssystemen wie Solaris, HP-UX, AIX und anderen.

Diese viel breitere Einsetzbarkeit der OPC UA Technologie ermöglicht die Realisierung ganz neuer vertikaler Integrationskonzepte. Durch Kaskadierung von OPC UA Komponenten können Informationen sicher und zuverlässig von der Fabrikhalle bis in das Produktionsplanungs- oder ERP-System transportiert werden. Dabei werden embedded UA Server auf der Feldebene über Client- und Server-fähige UA-Komponenten auf der Automatisierungsebene mit integrierten UA-Clients in ERP-Systemen auf der Unternehmensleitebene verbunden. Die jeweiligen UA-Komponenten können dabei geografisch verteilt und ohne weiteres durch Firewalls voneinander getrennt sein.