

**Kon-CEPT**

## **OLE for Process Control**

OPC Data Access, OPC XML und  
Erfahrungen mit OPC

Mai 2008

# Inhalte

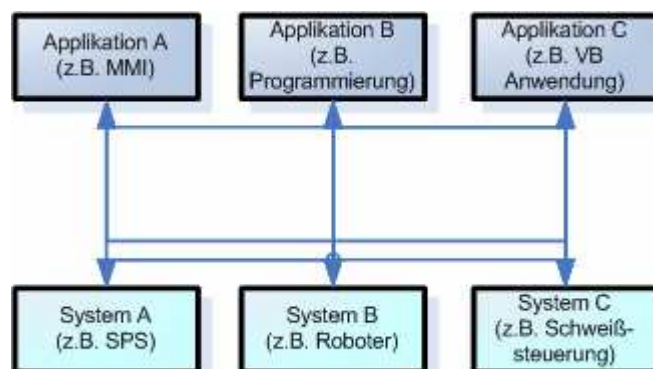
<b>1.</b>	<b>Einführung</b>	<b>3</b>
1.1.	Die Situation vor OPC	3
1.2.	OPC ermöglicht eine neue Struktur	4
1.3.	OPC Schnittstellen	5
<b>2.</b>	<b>OPC Data Access</b>	<b>5</b>
2.1.	Die logische Struktur für den Zugriff auf Daten	5
2.2.	Technischer Aufbau der OPC Infrastruktur	7
2.3.	Die Verwendung von OPC in MMS	8
<b>3.</b>	<b>Erfahrungen mit OPC</b>	<b>10</b>
<b>4.</b>	<b>OPC XML</b>	<b>13</b>
4.1.	Web Services	13
4.2.	Microsoft .NET	13
4.3.	Ausblick	14

## 1. Einführung

OPC (OLE for Process Control) ist ein Standard für die Kommunikation zwischen Windows Applikationen und Geräten auf der Automatisierungsebene. Dieser Standard wird von der OPC Foundation (<http://www.opcfoundation.org>) verwaltet und publiziert. Die OPC Foundation ist ein Gremium dem annähernd 200 Automatisierungshersteller, Softwarehersteller und Anwender angehören. Hierzu zählen alle namhaften Hersteller von Systemen, die auch im Automobilbau Anwendung finden.

### 1.1. Die Situation vor OPC

Folgende Abbildung zeigt die Problematik, aus der heraus OPC entstanden ist:



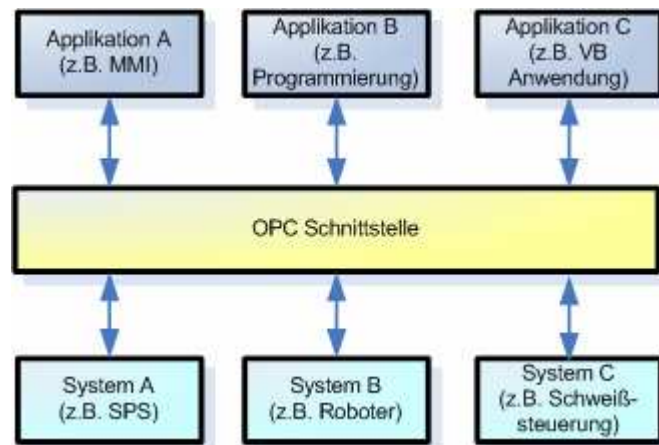
Jede Applikation muss für jedes Gerät auf der Automatisierungsebene, mit der es kommunizieren soll, einen spezifischen Treiber mitbringen, der das Gerätespezifische Protokoll speziell für dieses Gerät implementiert. Dies hat folgende Nachteile:

- Hoher Entwicklungsaufwand auf Seiten der Applikationsentwickler, da jeder Hersteller einer Applikation eigene Treiber für jedes Gerät entwickeln muss
- Probleme durch Treiber verschiedener Applikationen, die nicht nebeneinander auf dem gleichen PC laufen können, sind beinahe vorprogrammiert
- Hoher Installations- und Verwaltungsaufwand für den Anwender, der z.B. die IP-Adresse eines Geräts an mehreren Stellen verwalten muss

Diese Situation erinnert ein wenig an die Situation von Software, wie sie unter MS-DOS üblich war. Dort wurden für Microsoft Word für DOS mehr Disketten mit Druckertreibern ausgeliefert, als Disketten für das Programm selbst erforderlich waren.

## 1.2. OPC ermöglicht eine neue Struktur

Durch die Einführung von OPC ergibt sich folgende Systemstruktur:



OPC definiert eine Standardschnittstelle, die für die Verbindung zwischen beliebigen PC-Applikationen und beliebigen Geräten auf der Automatisierungsebene geeignet ist. Jeder Hersteller von PC-Software erstellt einen OPC Client, während jeder Hersteller für sein Geräte einen passenden OPC Server liefert, der das Gerätespezifische Protokoll implementiert und hinter der Standard-OPC-Schnittstelle verbirgt. Heute ist es selbstverständlich, dass ein Druckerhersteller für seinen Drucker einen Druckertreiber liefert, den alle Applikationen in der gleichen Weise verwenden. Genauso sollte es für einen Hersteller eines Automatisierungsgerätes selbstverständlich sein, für sein Gerät einen passenden OPC Server zu liefern.

OPC ermöglicht auch den parallelen Betrieb mehrerer OPC Server und Clients auf einem PC.

### 1.3. OPC Schnittstellen

OPC ist über die Jahre seit seiner Einführung gewachsen, sodass der Standard heute folgende Schnittstellen definiert:

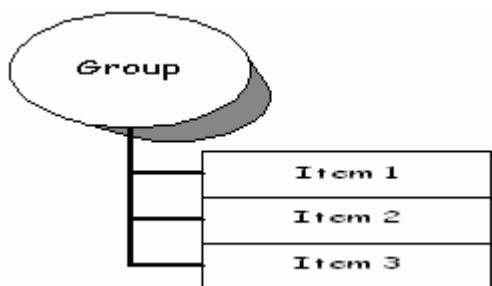
- OPC Data Access - dies war der erste Satz von Schnittstellen, der in OPC definiert war.
- OPC Alarms and Events
- OPC for Batch Applications
- OPC Historical Data Access
- OPC Security and Common Interfaces
- OPC XML

Von den meisten Herstellern wird heute OPC Data Access unterstützt, während andere Schnittstellen eher ignoriert werden. OPC XML ist noch zu taufisch, sodass derzeit noch keine diesem Standard entsprechenden Produkte am Markt erhältlich sind.

## 2. OPC Data Access

### 2.1. Die logische Struktur für den Zugriff auf Daten

Für den Zugriff auf Daten in Automatisierungsgeräten bietet OPC eine standardisierte logische Struktur an, die in folgender Darstellung illustriert wird:



Der Client kann eine beliebige Anzahl von OPC Gruppen erzeugen. Innerhalb jeder Gruppe kann der Client weiters eine beliebige Anzahl von Items erzeugen. Jedes Item entspricht einem Datum am Automatisierungsgerät bzw. einem Datenfeld. Jeder Item besitzt eine Reihe von Eigenschaften, die der Client für die weitere Verarbeitung verwenden kann.

Die wichtigsten Eigenschaften eines Items sind:

- Wert
- Qualitätsstatus (z.B. IO, unbekannt, Gerätefehler etc.)
- Zeitstempel
- Zugriffsrechte (lesen, schreiben, schreiben und lesen)

Der Client hat folgende Möglichkeiten, auf Items zuzugreifen:

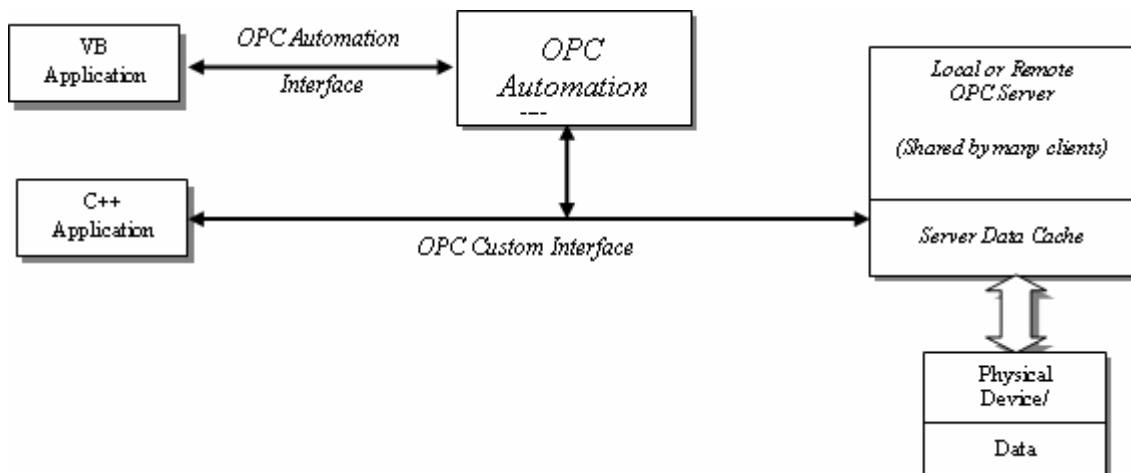
- Synchrones Schreiben und Lesen
- Asynchrones Lesen und Schreiben
- Abonnieren von Benachrichtigungen. Wenn der Client eine Benachrichtigung abonniert, so wird er vom Server benachrichtigt, wenn sich der Wert des Items ändert. Der Client kann zusätzlich eine maximale Datenrate (Updatezeit der Gruppe) und einen Bias für Änderungen von analogen Werten angeben

Auch wenn der Client im Falle eines Abonnements nur dann benachrichtigt wird, wenn sich der Wert eines Items ändert, bedeutet dies nicht automatisch, dass ein ähnlicher Mechanismus für den Datenaustausch zwischen OPC Server und Automatisierungsgerät verwendet wird. Es steht dem Server frei, das Automatisierungsgerät dennoch zu pollen, was bei den meisten SPS am Markt auch der Fall ist, da Änderungserkennung und unsolicited Messaging in der SPS zusätzliche Anwenderprogrammierung erfordert.

Items werden durch eine Kombination aus Access Path und Itemnamen identifiziert. OPC legt keine Syntaxregeln für diese Angaben fest. Die Art, wie Items bezeichnet werden, ist daher abhängig vom verwendeten Automatisierungsgerät und herstellerspezifisch.

## 2.2. Technischer Aufbau der OPC Infrastruktur

Die OPC Infrastruktur ist entsprechend folgender Abbildung aufgebaut:



Die Schnittstellen, die von OPC spezifiziert werden, entsprechen der Microsoft COM Spezifikation (Component Object Model). Hierbei stehen immer zwei Sätze von Schnittstellen zur Verfügung, nämlich "Custom Interfaces", die einen sehr performanten Datenzugriff von Applikationen, die auf C/C++ basieren bietet und "Automation Interfaces", die eine geringere Performance durch den mit ihnen verbundenen Binding Overhead anbieten, aber aus Applikationen wie Visual Basic oder VBA sehr einfach anzusprechen sind.

Die Festlegung auf das COM Modell hat den Vorteil, dass OPC Server aus Windows Applikationen sehr performant und mit geringem Aufwand ansprechbar sind. Der Nachteil, den man sich damit erkaufte, ist die Bindung an die Windows Plattform. Dies bedeutet, dass OPC mit Java oder mit Unix Plattformen nicht ohne weiteres funktioniert. Dies war einer der wesentlichsten Gründe, warum MMS auf einer Windows Plattform basiert.

Da OPC auch DCOM unterstützt (Distributed COM), muss der OPC Server nicht notwendigerweise auf dem gleichen PC installiert sein, wie der Client. Kuka verwendet dieses Konzept z.B. in seinen Robotersteuerungen, die über einen lokal installierten OPC Server verfügen, auf den eine Client Applikation auf einem PC remote über Ethernet TCP/IP unter Verwendung des DCOM Protokolls zugreifen kann.

## Zusätzliche Dienste

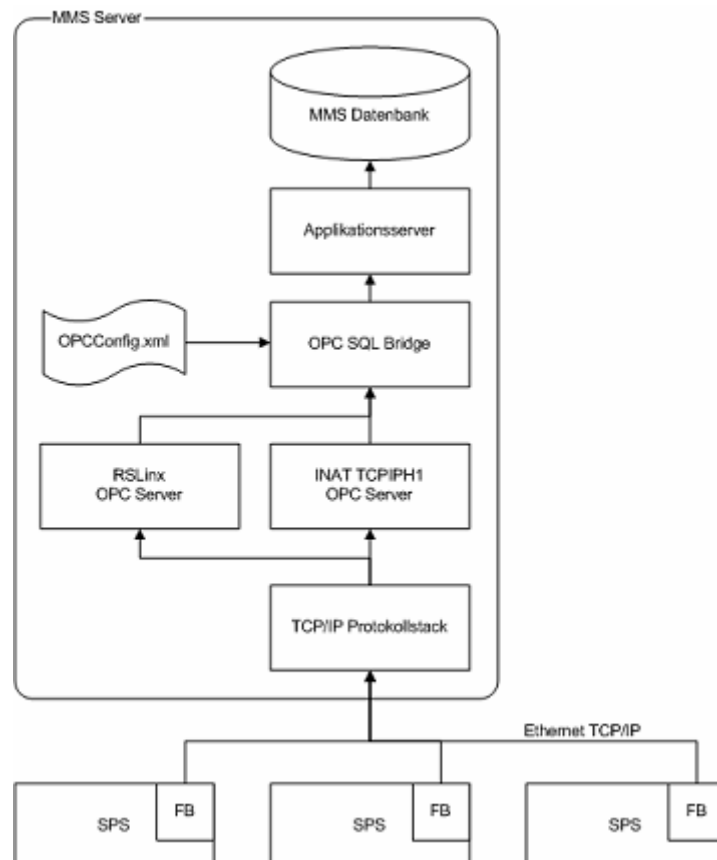
OPC bietet eine Reihe weiterer Dienste an, die für eine nahtlose Integration von Applikationen und OPC Servern verschiedener Hersteller erforderlich sind. Diese sind:

- Der OPC Server Enumerator OPCENUM. Dieser Dienst ermöglicht es Clients, herauszufinden, welche OPC Server auf dem PC installiert sind und wie diese aktiviert werden.
- OPC Security - Hierdurch wird sichergestellt, dass nur authentifizierte und autorisierte Clients auf OPC Server zugreifen können
- OPC Browser - OPC definiert einen Satz von Schnittstellen, die es einem Client ermöglichen, den Adressraum des Servers zu durchsuchen und mögliche Items dem Anwender z.B. in einer Auswahlliste zu präsentieren.

### 2.3. Die Verwendung von OPC in MMS

#### Datensammlung von SPS und Feldgeräten

Die Datensammlung von SPS und Feldgeräten ist nach dem folgenden Schema implementiert:



<b>Komponente</b>	<b>Beschreibung</b>
MMS Funktionsbaustein	Für die Abwicklung der Kommunikation mit dem MMS Server ist es erforderlich, auf den Stations-SPS für jede Station eine Instanz des MMS Funktionsbausteins aufzurufen.
TCP/IP Protokollstack	Die Datenkommunikation zwischen den Stations-SPS und dem MMS Server erfolgt ausschließlich über Ethernet TCP/IP. Hierbei können je nach Serverauslegung zusätzliche Dienste, wie HP Teaming für die Erhöhung der Verfügbarkeit durch eine redundante Netzwerkauslegung zum Einsatz kommen.
OPC Server	Für jede SPS Type, die im Projekt eingesetzt wird, ist ein passender OPC Server zu installieren. Die Aufgabe des OPC Servers ist es, systemspezifische Protokolle der einzelnen SPS-Hersteller zu kapseln und der Applikationssoftware eine Standardschnittstelle zur Verfügung zu stellen. In einigen Fällen kann der OPC Server auch auf einem anderen Rechner oder am Gerät selbst installiert sein. In diesem Fall wird mittels DCOM Remote auf den OPC Server zugegriffen (z.B. Kuka KRC2).
OPC-SQL-Bridge	Die OPC-SQL-Bridge ist ein Dienst, der die Verbindung zwischen der MMS Datenbank und den angebundenen OPC Servern herstellt. Die OPC-SQL-Bridge ruft Dienste der MMS Datenbank auf, um Daten in die Datenbank zu schreiben, oder Daten aus der Datenbank zu lesen und auf die angebundenen SPS Systeme zu transferieren.

Für eine Liste der von MMS unterstützten Automatisierungs- und Feldgeräten siehe die Beschreibung des MMS Systemaufbaus.

Die OPC-SQL-Bridge ist vollständig als Windows Dienst im .NET Framework (Sprache C#) implementiert. Sie enthält daher ein Assembly, das den Zugriff auf OPC Server kapselt und diese als eine Reihe von .NET Klassen präsentiert.

Die Konfiguration aller Komponenten, die für die Kommunikation mit Geräten auf der Automatisierungs- und Feldebene erforderlich sind, ist in der MMS Datenbank abgelegt. Der Administrator kann alle erforderlichen Einstellungen vollständig im MMS Web Client vornehmen, sodass er nicht mit zusätzlichen Werkzeugen konfrontiert ist und die Komplexität des Aufbaus vollständig vor ihm verborgen wird, sodass er nicht über spezifisches Know-How zu OPC verfügen muss.

### 3. Erfahrungen mit OPC

Unsere Erfahrungen mit OPC sollen an Hand diverser Aussagen und Mythen, mit denen wir immer wieder konfrontiert sind, diskutiert werden.

OPC basiert auf Windows und ist daher nicht stabil genug

Kon-Cept hat bisher einige Windows-basierte MMS Systeme unter Verwendung von OPC für den Datenaustausch installiert. Einige dieser Systeme sind seit mehr als einem Jahr ununterbrochen in Betrieb (d.h. ohne Neustart des Servers oder der Applikationen).

Wenn Windows Server professionell installiert und gewartet werden, so ist die Stabilität selbst für prozesskritische Anwendungen in der Automobilfertigung absolut ausreichend.

Vor Einsatz eines neuen OPC Servers muss dieser jedoch eingehend auf seine Eignung für den Anwendungsfall in MMS geprüft werden. Viele OPC Server sind nicht für den Betrieb in MMS geeignet, da sie nicht unter Windows Server oder auf Multiprozessormaschinen laufen.

## OPC ist langsam und bremst die Performance

Das größte System, das wir bisher installiert haben, hat folgendes Mengengerüst:

Beschreibung	Wert
Anzahl angebundener SPS	146
Anzahl angebundener PC-basierter Systeme	92
Anzahl Betriebsmeldungen je Tag	2,4 Mio.
Anzahl Fahrzeugbewegungen je Tag	220.000

Wie sich gezeigt hat, liegen die Flaschenhälse, die letztendlich die Performance des Systems begrenzen nicht in der OPC Schnittstelle, sondern in folgenden Komponenten:

- Die Performance der Datenbank ist ein entscheidender Faktor. MMS basiert auf unsolicited Messaging und muss unter weichen Echtzeitanforderungen auf Stationsmeldungen reagieren. Die Transaktionsleistung der Datenbank ist hier ein entscheidender Faktor, der die maximale Größe von MMS Systemen begrenzt.
- MMS verwendet unsolicited Messaging. Dies erfordert Zusatzaufwende durch Programmierung der Ereigniserkennung und der Sende-FBs in den Automatisierungsgeräten auf der einen Seite und erweiterte Queuingmechanismen im MMS Server, um mit Bursts umgehen zu können. Der Aufbau eines Systems dieser Größe, welches auf Polling von Daten basiert, würde das Netzwerk extrem belasten und würde auf Grund der langen Pollzyklen keine akzeptable Performance bieten.

Vorurteile über die Performance von OPC stammen aus Erfahrungen, die Anwender mit früheren Standards, die auf NetDDE basierten, gemacht haben. NetDDE ist mittels Windows Meldungen, die zwischen den Applikationswarteschlangen der beteiligten Applikationen ausgetauscht werden, implementiert und daher naturgemäß erheblich langsamer als OPC.

## OPC verbirgt die Eigenheiten von Automatisierungsgeräten vor dem Client

Diese Aussage ist leider nur bedingt wahr. Für die effiziente Implementierung von Systemen ist es erforderlich, die Protokollmechanismen, die der OPC Server für den Datenaustausch mit dem Automatisierungsgerät verwendet, zumindest im Prinzip zu verstehen. Entscheidend ist z.B. die Frage, ob Polling oder unsolicited Messaging verwendet wird.

OPC überlässt die Syntax für den Aufbau von Itemnamen, die angeben, welche Daten am Automatisierungsgerät ein Item im OPC Server repräsentiert, dem Gerätehersteller. Dementsprechend ergeben sich bei Einsatz verschiedener OPC Server subtile Unterschiede in der Behandlung von Datentypen und Abonnements. Auch die automatisierte Generierung von Itemnamen, wie sie in MMS erforderlich ist, um die Konfiguration der OPC Schnittstelle vor dem Anwender zu verbergen, ist spezifisch für verschiedene Server.

## OPC stellt nicht alle Funktionen bereit, die proprietäre Protokolle einzelner Hersteller bieten

Diese Aussage ist richtig. Praktisch jeder Hersteller bietet proprietäre Erweiterungen zu OPC an, deren Spezifikation oftmals nicht öffentlich publiziert ist. Die von OPC angebotenen Funktionen reichen für den Anwendungsfall in MMS Systemen jedoch aus.

## OPC ist auf Windows Systeme begrenzt

Manche Propagandaschriften behaupten, dass z.B. über DCOM Bridges ein Einsatz von OPC auch mit Unix Systemen möglich ist. Wir haben noch keine ernstzunehmende Applikation gesehen, die solch eine Komponente tatsächlich nutzt. Derzeit ist OPC auf Windows Plattformen festgelegt, was einer der Gründe für die Entscheidung, MMS auf Windows Servern zu entwickeln war. Durch neue Spezifikationen, wie OPC XML sollte dieses Problem in den nächsten Jahren gelöst werden.

## 4. OPC XML

### 4.1. Web Services

In den letzten Jahren wurden einige neue Standards in der Informationstechnologie publiziert, die von allen wichtigen Herstellern akzeptiert und auch tatkräftig unterstützt werden. Diese sind:

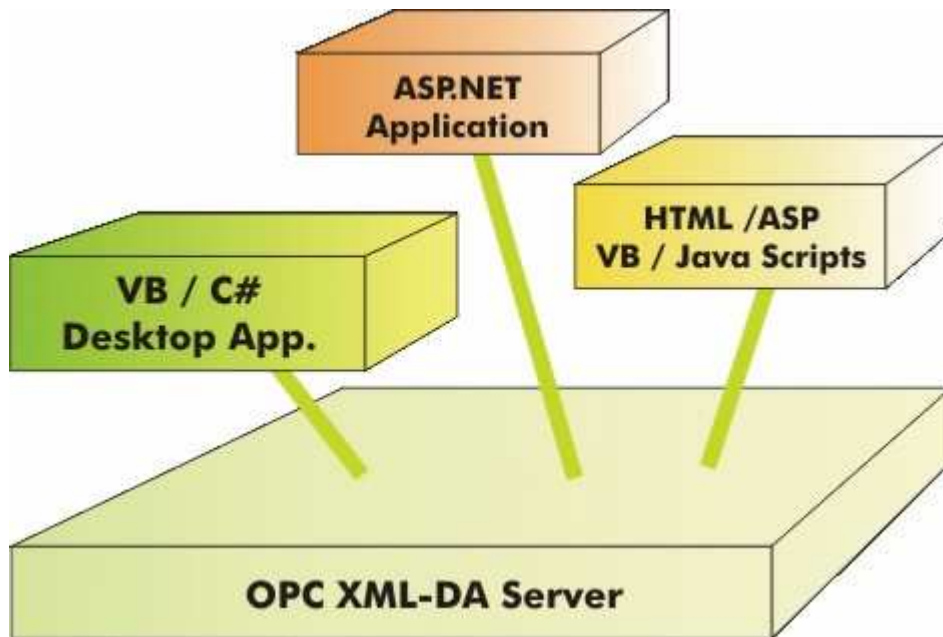
- XML
- XML Schema
- SOAP
- UDDI

Diese Standards werden generell unter dem Begriff "Web Services" zusammengefasst. Web Services ermöglichen den plattformübergreifenden Datenaustausch zwischen verschiedenen Systemen. Web Service Toolkits werden heute von allen wichtigen Herstellern angeboten und sind für Windows (.NET, Web Service Toolkit), Java und andere Umgebungen verfügbar. Web Services setzen keine spezielle Kommunikationsinfrastruktur voraus. SOAP Packets können über HTTP, plain TCP, SMTP oder Message Queuing Systeme, wie IBM WebSphere MQ transportiert werden. HTTP auf TCP/IP ist der derzeit gängigste Transportmechanismus.

Die Unterstützung von Web Services kann OPC zu einem allgemein einsetzbaren Standard machen und aus seiner Abhängigkeit von der Windows Plattform lösen.

### 4.2. Microsoft .NET

Die von Microsoft neu eingeführte .NET Plattform basiert nicht mehr auf COM (Component Object Model). Aus diesem Grund werden COM Schnittstellen in Kürze eine veraltete Technologie darstellen, sodass OPC auf eine neue Plattform gestellt werden muss. Da .NET eine sehr umfangreiche Unterstützung für Web Services anbietet, sind diese die klare Wahl als Basis für neue OPC Spezifikationen. Folgende Abbildung zeigt die neue Systemumgebung:



#### 4.3. Ausblick

Die aktuelle Version der OPC XML Spezifikation deckt im wesentlichen die gleichen Funktionen ab, wie die OPC Data Access Spezifikation. Es ist daher in Kürze mit OPC Servern zu rechnen, die diesen Standard unterstützen bzw. mit Wrappern, die OPC DA Server kapseln und eine OPC XML Schnittstelle zur Verfügung stellen. Kon-Cept wird die OPC XML Schnittstelle in Kürze in MMS als weitere Option für die Ankopplung an Anlagensysteme unterstützen.