

# Middleware für verteilte industrielle Umgebungen

## OLE for Process Control



# Übersicht

- ❑ Was ist OPC?
- ❑ Geschichte
- ❑ Spezifikationen
- ❑ Aktuelle Entwicklung
- ❑ Bewertung

Quellen: Thomas J. Burke, President OPC Foundation  
Lee Neitzel, Emerson Process  
Thomas Hadlich, ifak system GmbH  
J. Lange / F. Iwanitz, softing AG



# Was ist OPC?

- ❑ OLE for Process Control
- ❑ OPC: Openness, Productivity, Connectivity
- ❑ Definition von Softwareschnittstellen basierend auf Technologien der Microsoft- Welt
- ❑ Zugang zu Daten der Automatisierungstechnik (E/A an Feldbussen, SPS, Kompaktregler u.s.w.) erleichtern
- ❑ Alle namhaften Hersteller der Automatisierungstechnik bieten OPC- fähige Server und Clients an
  
- ❑ <http://www.opcfoundation.org>



# Was ist OPC? (ff)

- ❑ Jeder Client kann sich mit jedem Server verbinden
  - Ziel des interoperablen “Plug and Play”
- ❑ Kombination der besten Eigenschaften proprietärer C APIs
- ❑ Offener und publizierter Standard
- ❑ flexibel – Unterstützung unterschiedlicher Typen von Clients und Server
- ❑ effizient – optimiert für schnellen Datentransport
- ❑ Unterstützung aller Programmiersprachen
  - C, C++, VB, Java, HTML, DHTML
- ❑ für Internet geeignet



# Was ist OPC? (ff)

- ❑ Openness: Users are able to EASILY ACCESS real-time industrial plant floor DATA
- ❑ Productivity: OPC is an OPEN industry-standard. End-users and suppliers have to spend less time on integration issues
- ❑ Connectivity: End users will be able to use OPC Client applications (HMI, SCADA, MES, Custom ...) with a broad range of automation devices and systems

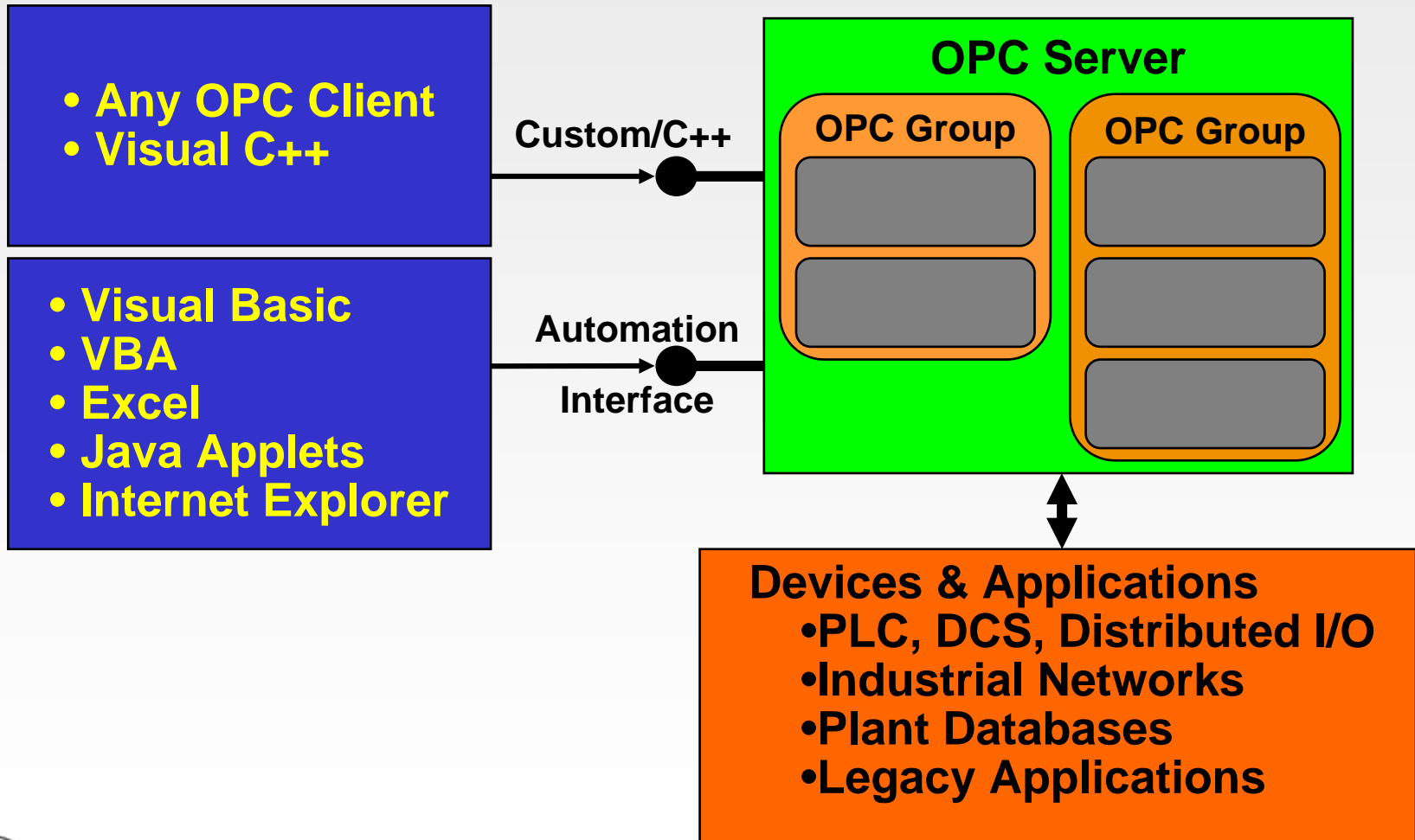


# Geschichte

- ❑ 1995 Entstehung der OPC Task Force:
  - Basierend auf Microsoft DCOM (OLE) Standard für Zugriff auf Echtzeitdaten
  - August 1996 OPC Specification Version 1.0
- ❑ 1996 Gründung der OPC-Foundation
  - 1997 Data Access Specification Version 1.0A
  - 1998 Data Access Specification Version 2.0
  - 1999 Alarm and Events Specification Version 1.01
  - 2000 Historical Data Access Specification Version 1.0
  - 2000 Security Specification Prerelease 1.0



# Spezifikationen



# Spezifikationen (ff)

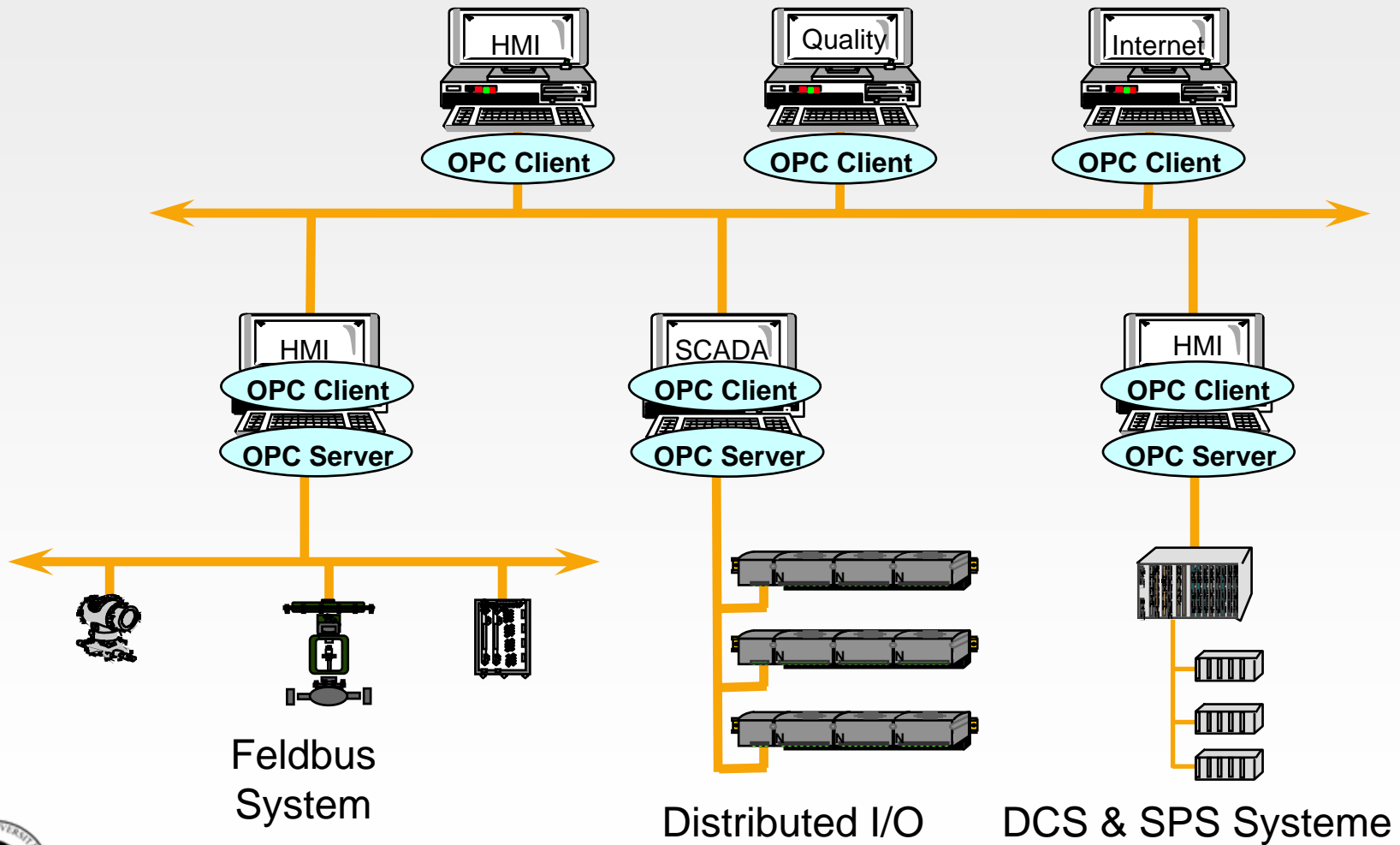
## □ Data Access Specification V 2.04

- Definition einer Schnittstelle zum Prozessdatenaustausch
- 1 oder mehrere Clients können mit dem Server verbunden sein
- Definition eines Namensraums und einer Objekthierarchie (Knoten und Blätter)
- Namensraum enthält alle Daten, beliebig gegliedert
- Datenpunkte verfügen über Attribute





# Spezifikationen (ff)



# Spezifikationen (ff)

## □ Alarms and Events Specification V 1.01

- Strukturieren, Übertragen und Quittieren von Ereignissen
- Gleiche Datenquellen für Data Access und Alarms and Events Server möglich
- Alarms and Events Server sendet nur Ereignis, keine Werte



# Spezifikationen (ff)

## □ Historical Data Access Specification

- Definition eines Adressraums und Objekthierarchie
- Zugriff auf historische Daten
- Unterscheidung in Roh- und aggregierte Daten
- Lesen, Schreiben, Verändern der Daten
- Implementierung von Trenddaten Server und Datenkomprimierungs- und Analyse-Server



# Spezifikationen (ff)

## □ Batch Specification Version 1.0

- Definition von Ergänzungen von OPC DA für Batchprocessing
- Batch Server unterstützt Browse-interface
- Unterstützung zusätzlicher OPC-Objekte und Interfaces
- Namensraum nach IEC 61512-1
- OPC-Items müssen immer lesbar sein



# Weiterentwicklungen

- ❑ Aktive Technical Sub-Committees
- ❑ OPC Data Access 3.0 (OPC DA)
- ❑ OPC Common IO
- ❑ OPC Complex Data Access (OPC CDA)
- ❑ OPC XML / .NET (OPC XML)
- ❑ OPC UPnP
- ❑ OPC Data Exchange (OPC DX)
- ❑ OPC Command Execution (OPC CE)
  
- ❑ **Aktuell: OPC Unified Architecture (OPC UA)**



# Ziele der Weiterentwicklungen

## □ Ziele von OPC **Complex Data Access**

- Bessere Unterstützung von Daten der Feldebene, die nicht auf die skalaren Datentypen von OPC abbildbar sind.
- Applikationen:
  - DCS, SPS & Feldbusse
  - CNC, Roboter
  - Maschinenüberwachung
  - ...
- Unterstützung zusammengesetzter Daten und Attribute
  - BLOB (images)
  - Arrays (e.g. Waveforms)
  - Compact Boolean Arrays (bit arrays)
  - Verschachtelte Strukturen



# Ziele der Weiterentwicklungen

## □ Anforderungen an OPC **Complex Data Access**

- Client muss komplexen Datentyp bestimmen können, der vom Server bereitgestellt wird
- Client muss zusätzliche Informationen zur Semantik erhalten
- Server stellt Dienste für zeitkonsistentes Lesen / Schreiben größerer „Datenböcke“
- Server erlaubt Erweiterungen für Suchen/Navigieren durch den Adressraum
- Kompatibilität zu bisherigen Lösungen
- Nutzung der Semantik von OPC XML



# Ziele der Weiterentwicklungen

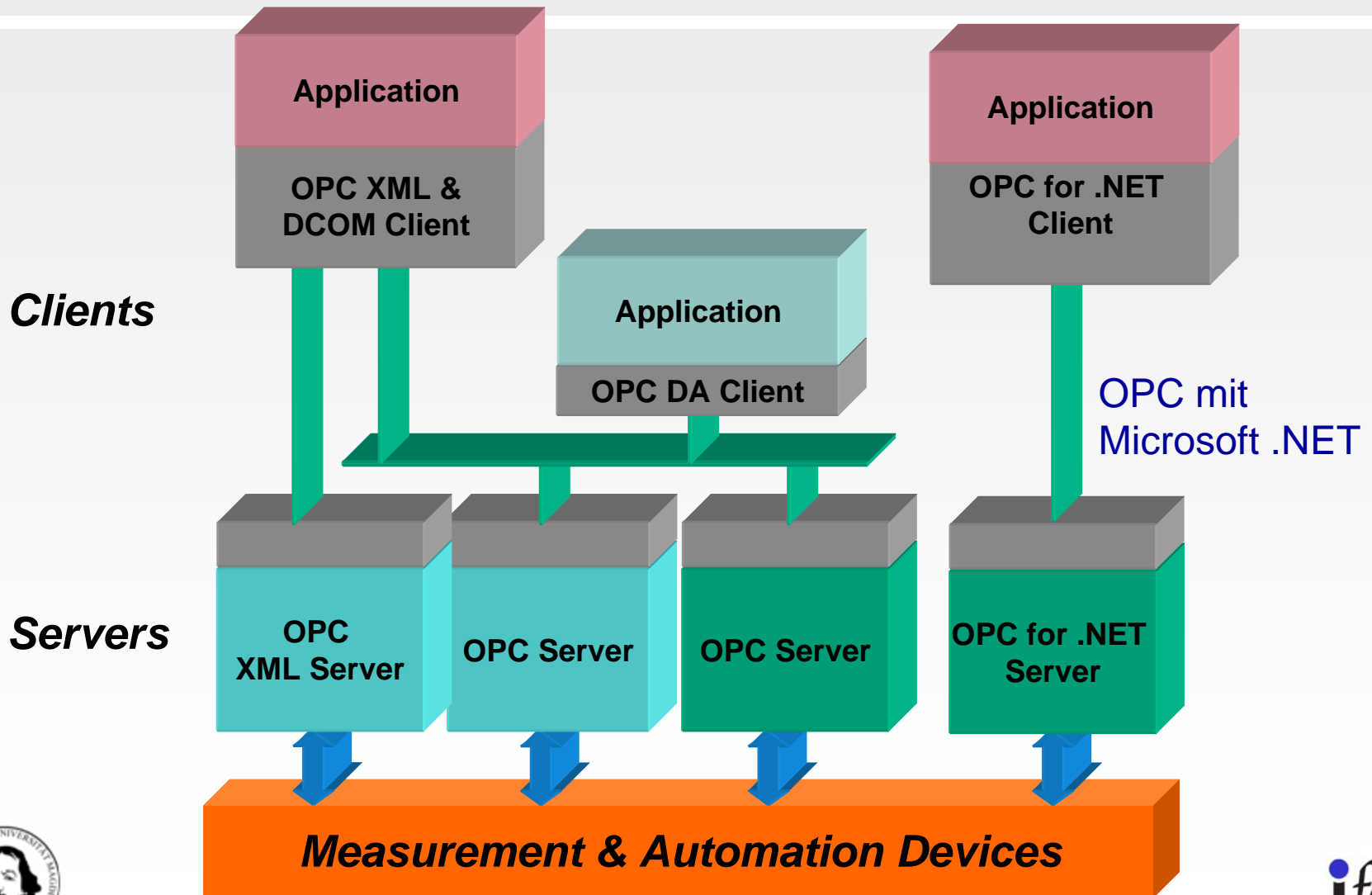
## □ Ziele von OPC XML / .NET

- Weiterführung der Arbeiten an der Interoperabilität von Anwendungen
- Vereinfachung des gemeinsamen Zugriffs auf Daten und deren Austausch auch auf einem höheren Level
- Entwicklung eines Austauschschemas für anfordernde und sendende OPC DA Datendienste
- Bereitstellung von Diensten für OPC Web Applikationen





# Kombinatorik der Technologien



# Ziele der Weiterentwicklungen

## □ OPC Data Exchange

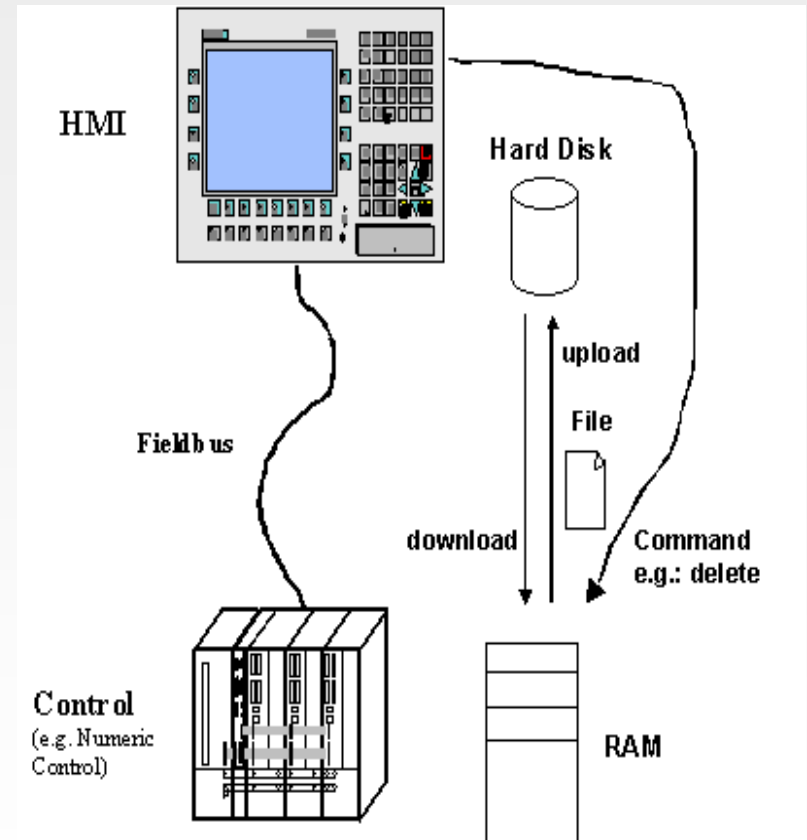
- Erweiterung von OPC Data Access
- Kommunikation zwischen OPC-Servern
- Kommunikation muß konfiguriert werden
- z.Z. Nutzung von (D)COM (Echtzeit)
- Nutzung von OPC XML geplant



# Ziele der Weiterentwicklungen

## □ OPC Command Execution

- Steuerung von Automatisierungsgeräten via OPC
  - PLCs
  - Komplexe Geräte (Drives, ...)
  - Ausführung von Aktionen



# Ziele der Weiterentwicklungen

## ❑ OPC Command Execution

- Proprietäre Lösungen, z.B. DriveServer-Spezifikation:
  - DS\_ParameterSet[0-9]+ (branch) /
  - Filename (R|W, stringvalue)
  - Action (R|W, {UPLOAD, DOWNLOAD, VERIFY, COMPARE})
  - State (R, {READY, RUNNING, UNDEFINED} )
  - Result (R, HRESULT)
- Probleme:
  - Speziallösung
  - Keine allgemeine Semantik
  - Keine Informationen für generische Clients



# Ziele der Weiterentwicklungen

## ❑ OPC Command Execution

### ❑ Vorteile:

- Standardstatusmaschine
- Zugang zur Kommandoausführung
- Ausführung von Kommandos
- Standard Informationsmechanismus
- Format in XML
- Erweiterbar

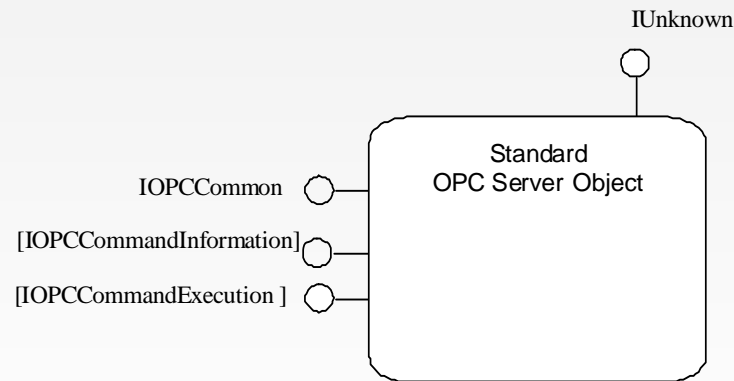
### ❑ Kommando:

- Aktion, die lange dauern, den Status des OPC Servers beeinflussen kann und auf die Daten des Gerätes wirkt



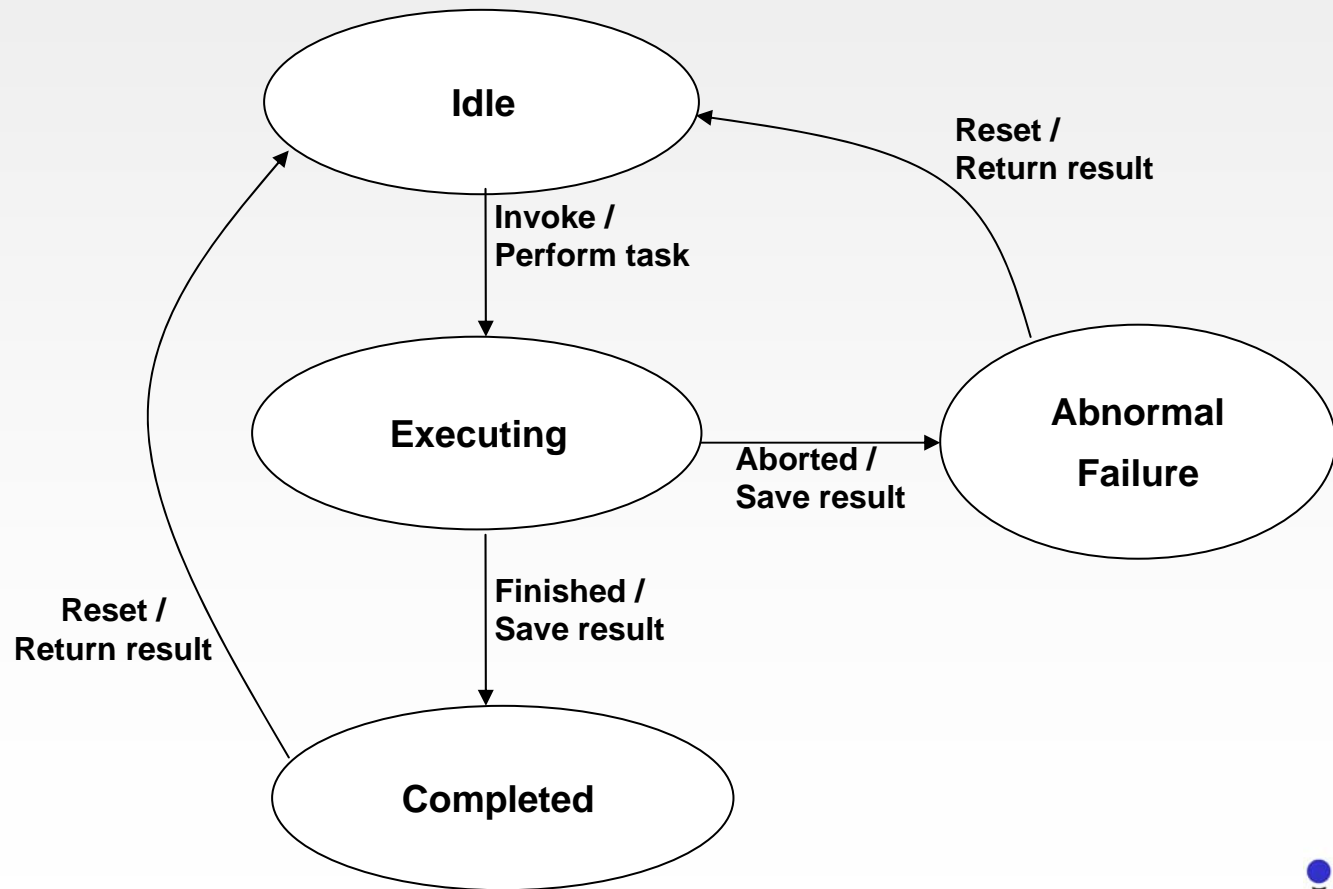
# Ziele der Weiterentwicklungen

## ❑ OPC Command Execution Interfaces



# Ziele der Weiterentwicklungen

## □ OPC Command Execution Statusmaschine



# Bewertung von OPC für die Automation

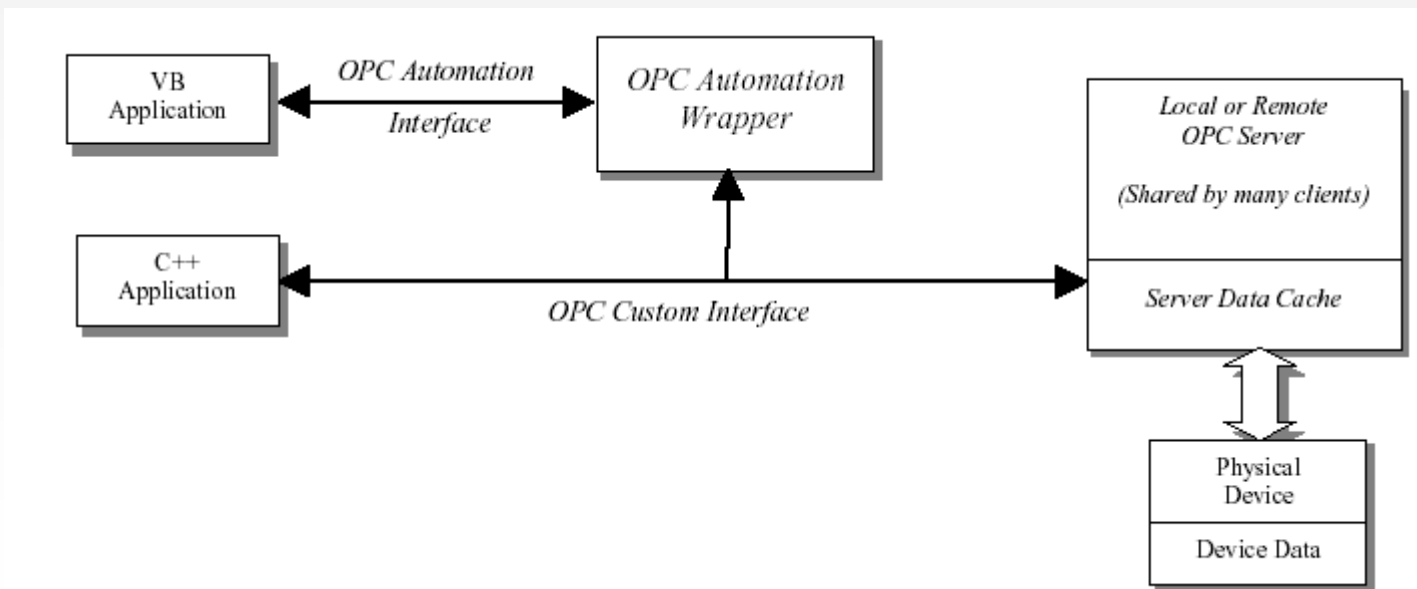
- ❑ Standard-Schnittstelle zwischen zwei Windows-Applikationen
- ❑ Größter Einfluss auf die automatisierungstechnischen Softwarebausteine
- ❑ Einfaches zusammenführen von Automatisierungsinself verschiedener Hersteller auf der Ebene der Prozessleittechnik
- ❑ Begünstigt die PC- basierte Steuerungs- und Regelungstechnik





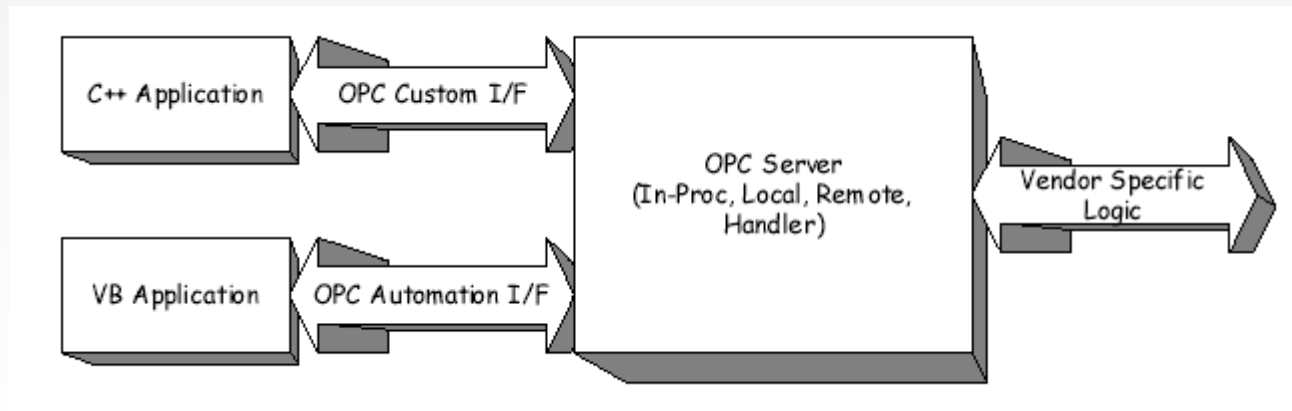
# Data Access 2.0x

- 2 Spezifikationen:
  - Custom Interface
  - Automation Interface

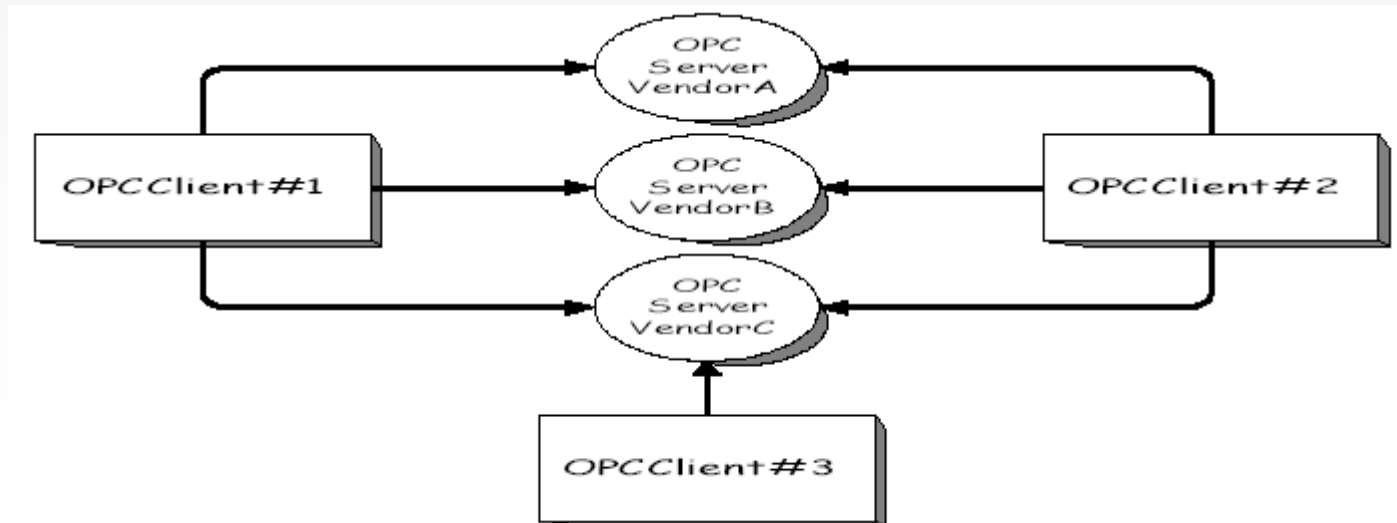
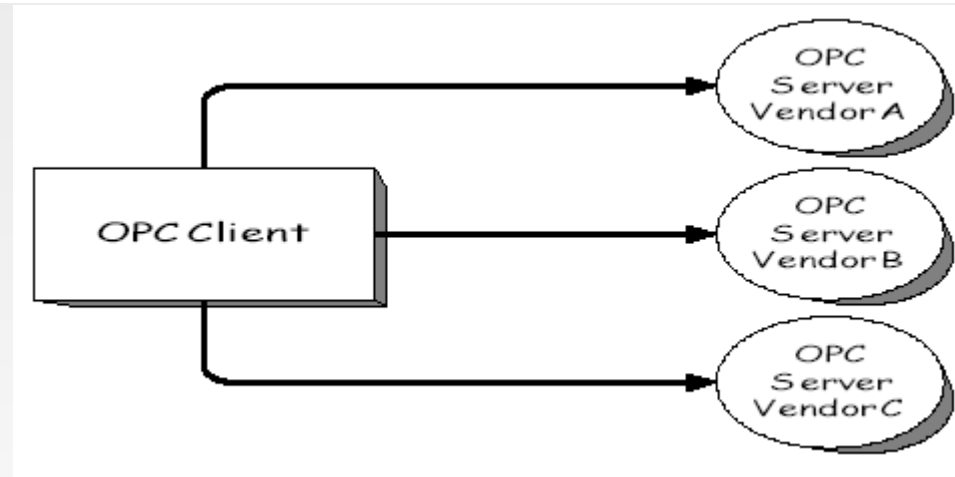


# Data Access 2.0x

- 2 Spezifikationen:
  - Custom Interface
  - Automation Interface

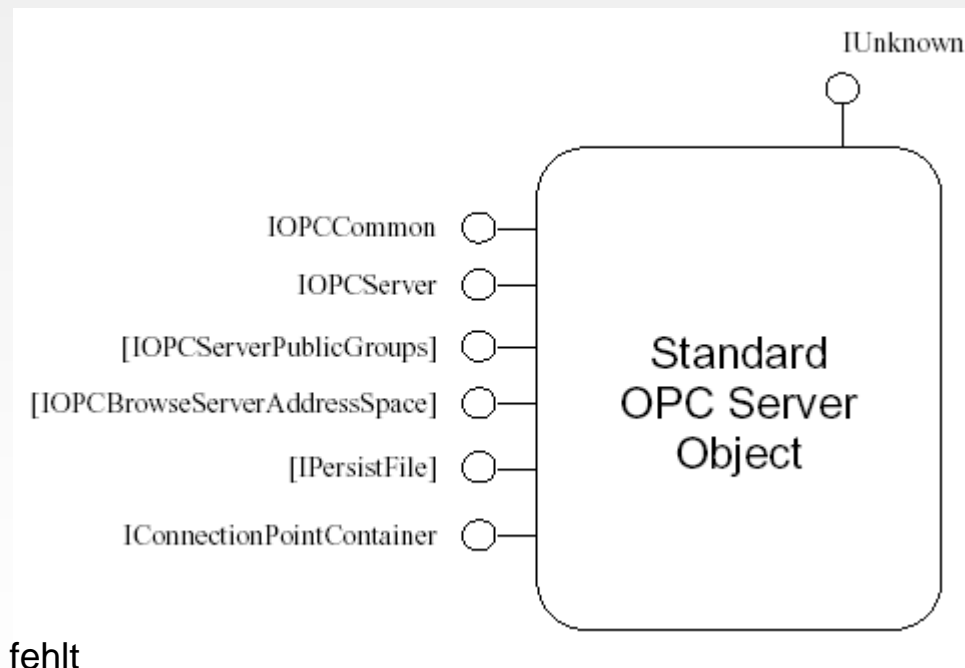


# Data Access 2.0x



# Data Access 2.0x

## ❑ Schnittstellen Standard Server Objekt



IOPCItemProperties fehlt



# Data Access 2.0x

## □ IOPCCommon

```
HRESULT SetLocaleID ( dwLcid );
```

```
HRESULT GetLocaleID ( pdwLcid );
```

```
HRESULT QueryAvailableLocaleIDs ( pdwCount, pdwLcid );
```

```
HRESULT GetErrorString ( dwError, ppString );
```

```
HRESULT SetClientName ( szName );
```



# Data Access 2.0x

## □ IOPCServer

```
HRESULT AddGroup ( szName, bActive, dwRequestedUpdateRate,  
                  hClientGroup,  
                  pTimeBias, pPercentDeadband, dwLCID,  
                  phServerGroup, pRevisedUpdateRate, riid,  
                  ppUnk );
```

```
HRESULT GetErrorString ( dwError, dwLocale, ppString );
```

```
HRESULT GetGroupByName ( szName, riid, ppUnk );
```

```
HRESULT GetStatus ( ppServerStatus );
```

```
HRESULT RemoveGroup ( hServerGroup, bForce );
```

```
HRESULT CreateGroupEnumerator ( dwScope, riid, ppUnk );
```



# Data Access 2.0x

## □ IConnectionPointContainer

```
HRESULT EnumConnectionPoints( IEnumConnectionPoints ppEnum );  
HRESULT FindConnectionPoint( REFIID riid,  
                             IConnectionPoint ppCP );
```



# Data Access 2.0x

## □ IOPCItemProperties

```
HRESULT QueryAvailableProperties ( szItemID, pdwCount,
                                  ppPropertyIDs,
                                  ppDescriptions,
                                  ppvtDataTypes );
```

```
HRESULT GetItemProperties ( szItemID, dwCount,
                             pdwPropertyIDs,
                             ppvData, ppErrors );
```

```
HRESULT LookupItemIDs ( szItemID, dwCount, pdwPropertyIDs,
                         ppszNewItemIDs, ppErrors );
```





# Data Access 2.0x

## ❑ IOPCBrowseServerAddressSpace (optional)

```
HRESULT QueryOrganization ( pNameSpaceType );  
HRESULT ChangeBrowsePosition ( dwBrowseDirection, szString );  
HRESULT BrowseOPCItemIDs ( dwBrowseFilterType,  
                             szFilterCriteria,  
                             vtDataTypeFilter,  
                             dwAccessRightsFilter,  
                             ppIEnumString );  
  
HRESULT GetItemID ( szItemDataID, szItemID );  
HRESULT BrowseAccessPaths ( szItemID, ppIEnumString );
```



# Data Access 2.0x

## ❑ IOPCServerPublicGroups (optional)

```
HRESULT GetPublicGroupName ( szName, riid, ppUnk );
```

```
HRESULT RemovePublicGroup ( hServerGroup, bForce );
```



# Data Access 2.0x

## □ IPersistFile (optional)

```
HRESULT IsDirty ( );
```

```
HRESULT Load ( pszFileName, dwMode );
```

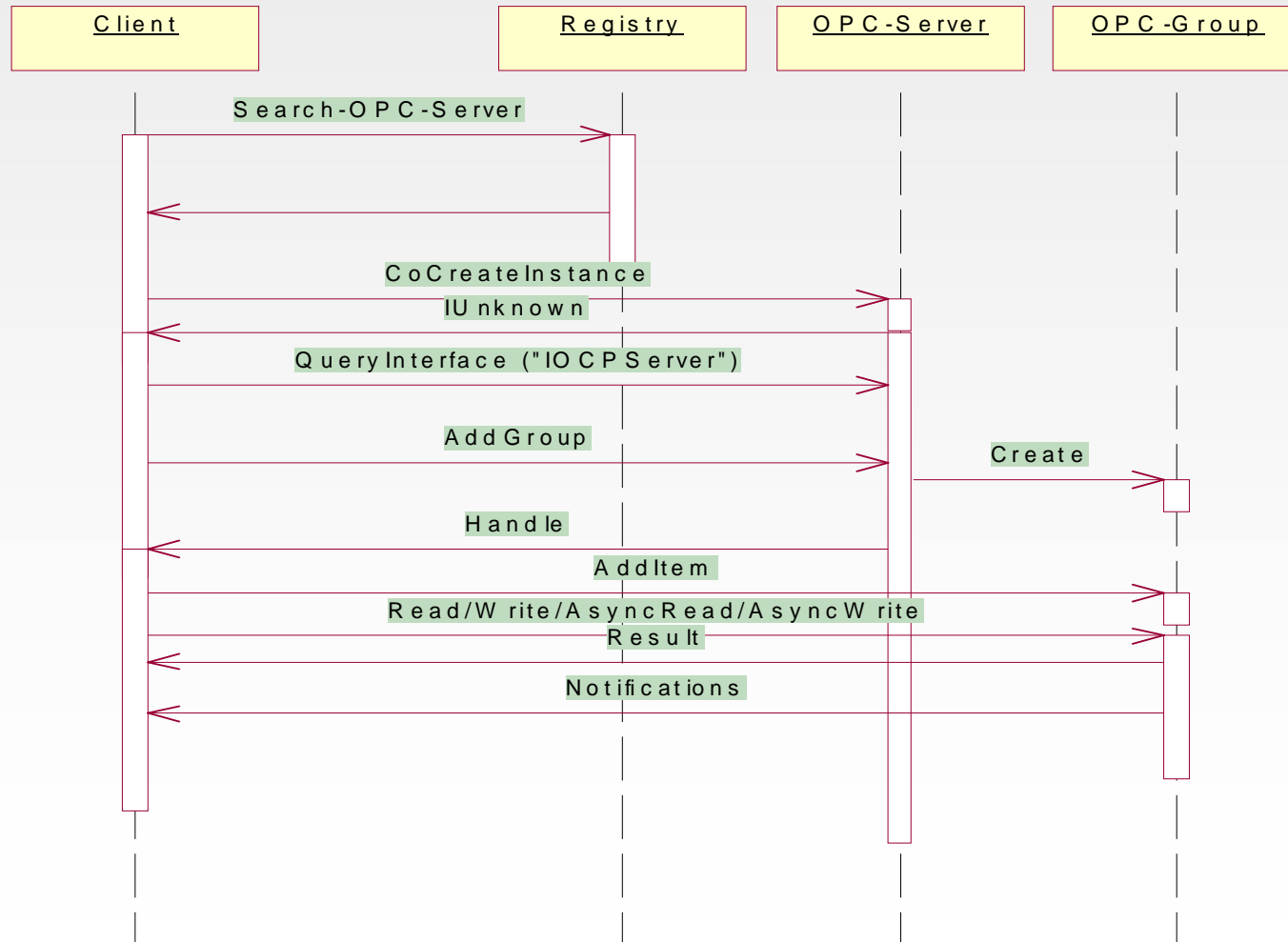
```
HRESULT Save ( pszFileName, fRemember );
```

```
HRESULT SaveCompleted ( pszFileName );
```

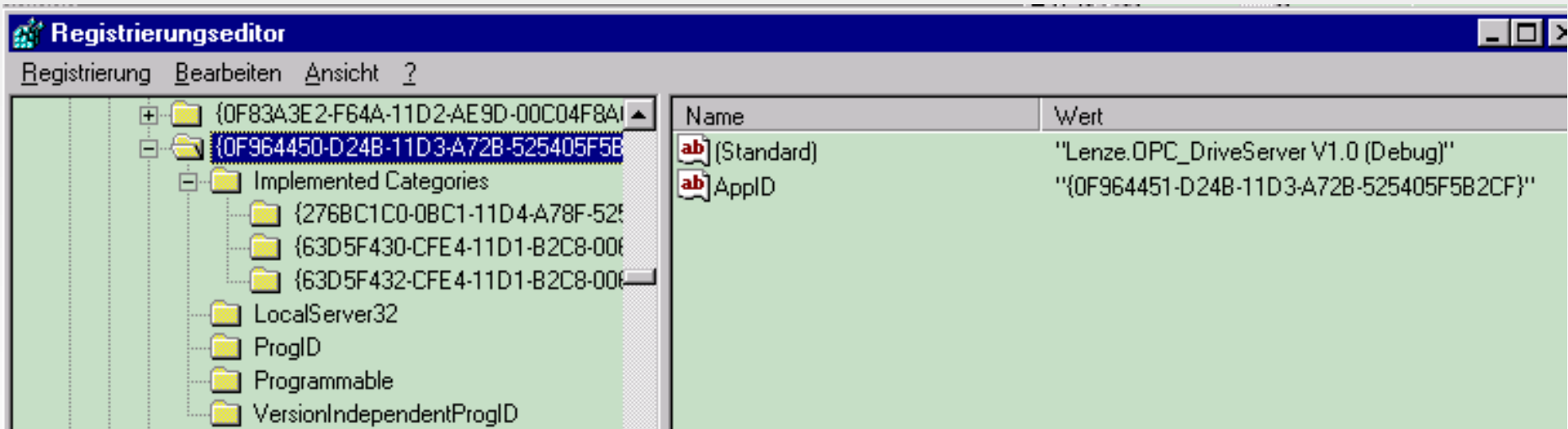
```
HRESULT GetCurFileName ( ppszFileName );
```



# Start eines OPC-DA Servers



# Registry Informationen



"OPC Data Access Servers Version 1.0"

CATID\_OPCDAServer10 = {63D5F430-CFE4-11d1-B2C8-0060083BA1FB}

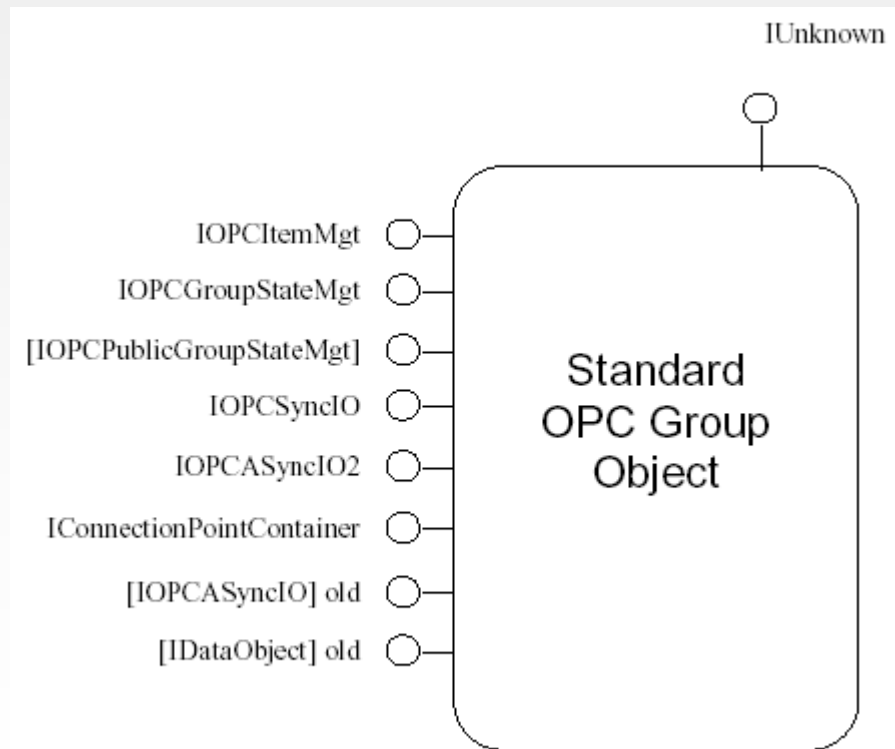
"OPC Data Access Servers Version 2.0"

CATID\_OPCDAServer20 = {63D5F432-CFE4-11d1-B2C8-0060083BA1FB}



# Data Access 2.0x

## ❑ Schnittstellen Standard Group Objekt



# Data Access 2.0x

## □ IOPCGroupStateMgt

```
HRESULT GetState ( pUpdateRate, pActive, ppName, pTimeBias,  
                  pPercentDeadband, pLCID, phClientGroup,  
                  phServerGroup );
```

```
HRESULT SetState ( pRequestedUpdateRate, pRevisedUpdateRate,  
                  pActive, pTimeBias, pPercentDeadband,  
                  pLCID, phClientGroup );
```

```
HRESULT SetName ( szName );
```

```
HRESULT CloneGroup ( szName, riid, ppUnk );
```



# Data Access 2.0x

## ❑ IOPCPublicGroupStateMgt (optional)

```
HRESULT GetState ( pPublic );
```

```
HRESULT MoveToPublic ( void );
```





# Data Access 2.0x

## □ IOPCSyncIO

```
HRESULT Read ( dwSource, dwCount, phServer, ppItemValues,  
              ppErrors );  
  
HRESULT Write ( dwCount, phServer, pItemValues, ppErrors );
```



# Data Access 2.0x

## □ IOPCAsyncIO2

```
HRESULT Read ( dwCount, phServer, dwTransactionID,  
              pdwCancelID, ppErrors );  
  
HRESULT Write ( dwCount, phServer, pItemValues,  
              dwTransactionID, pdwCancelID, ppErrors );  
  
HRESULT Cancel2 ( dwCancelID );  
  
HRESULT Refresh2 ( dwSource, dwTransactionID, pdwCancelID );  
  
HRESULT SetEnable ( bEnable );  
  
HRESULT GetEnable ( pbEnable );
```



# Data Access 2.0x

## □ IOPCItemMgt

```
HRESULT AddItems ( dwCount, pItemArray, ppAddResults,  
                  ppErrors );
```

```
HRESULT ValidateItems ( dwCount, pItemArray, bBlobUpdate,  
                       ppValidationResults, ppErrors );
```

```
HRESULT RemoveItems ( dwCount, phServer, ppErrors );
```

```
HRESULT SetActiveState ( dwCount, phServer, bActive,  
                        ppErrors );
```

```
HRESULT SetClientHandles ( dwCount, phServer, phClient,  
                          ppErrors );
```

```
HRESULT SetDatatypes ( dwCount, phServer,  
                      pRequestedDatatypes, ppErrors );
```

```
HRESULT CreateEnumerator ( riid, ppUnk );
```



# Data Access 2.0x

## □ IConnectionPointContainer

```
HRESULT EnumConnectionPoints ( IEnumConnectionPoints ppEnum );  
HRESULT FindConnectionPoint ( REFIID riid,  
                              IConnectionPoint ppCP );
```



# Data Access 2.0x

## □ IOPCDataCallback (Client)

```
HRESULT OnReadComplete ( dwTransid, hGroup, hrMasterquality,  
                          hrMastererror, dwCount, phClientItems,  
                          pvValues, pwQualities, pftTimeStamps,  
                          pErrors );
```

```
HRESULT OnWriteComplete ( dwTransid, hGroup, hrMastererr,  
                          dwCount, phClientItems, pErrors );
```

```
HRESULT OnCancelComplete ( dwTransid, hGroup );
```

```
HRESULT OnDataChange ( dwTransid, hGroup, hrMasterquality,  
                       hrMastererror, dwCount, phClientItems,  
                       pvValues, pwQualities, pftTimeStamps,  
                       pErrors );
```



# Data Access 2.0x

## ❑ IOPCShutdown (Client)

```
void ShutdownRequest ( szReason );
```



# Data Access 2.0x

## □ Datentypen

- „The VARIANT types VT\_I2, I4, R4, R8, CY, DATE, BSTR, BOOL, UI1 as well as single arrays of these types (VT\_ARRAY) are expected to be most commonly used (in part because these are the legal types in Visual Basic).“

