



VIGO

The Fieldbus Management System

for

Windows '95/98/NT/W2K

Windows XP

by

Palle Bergmann Mortensen

PROCES-DATA A/S



VIGO

- VIGO is a Fieldbus Management System,
- PC's running under MS Windows '95/98/NT/W2k and Windows XP.
- The application for VIGO is within distributed process automation systems, where one or more Fieldbuses are used.
- VIGO is an “operating system”, for handling the Fieldbus tasks.



The task of VIGO

- Is to provide a uniform interface between standard programmes in PC's,
 - such as Excel spreadsheet, Visual Basic, Delphi, C++ etc.
- and variables in Fieldbus modules (nodes).
- These variables are identified by a unique name (identifier).



The task of VIGO

- To hold information about the location and type of each variable identifier,
 - Such as the node address, internal address, offset, the data type etc.
- To communicate through different Fieldbus interfaces.
- Handle the queuing problems that occur in a multi-task environment.



The task of VIGO

- To keep track of which tools can be used with the various data objects
- These tools may be:
 - configuration tools, compilers, assemblers etc.
- To provide information to compilers and assemblers about variables.
- Ensure that Global variables only need to be declared once.
- To provide an editor, to describe the plant.

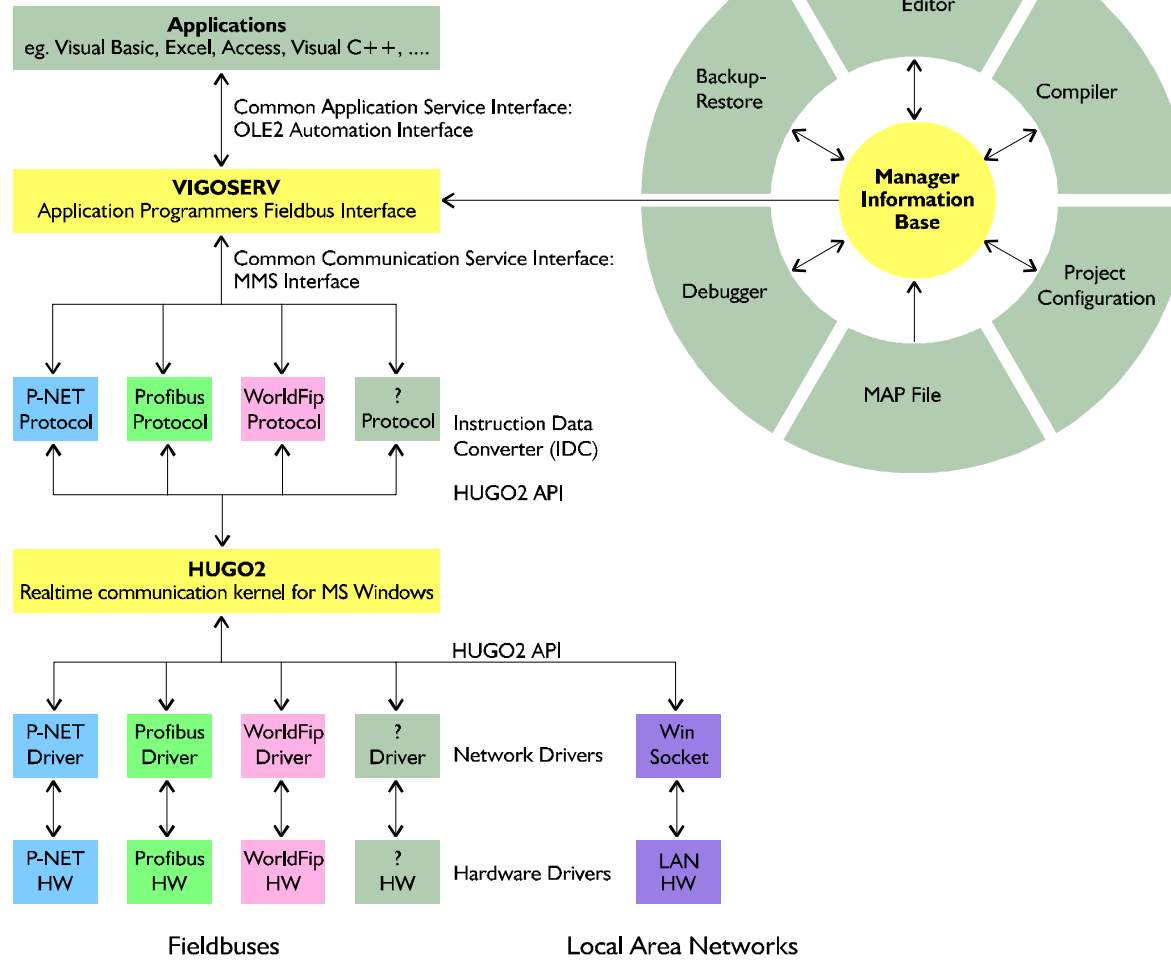


VIGO features

- The user does not need to consider the variations in different communication protocols, data conversion or addressing methods.
- The user need only to interface to one system, independent of Fieldbus type.
- From the users point of view, all these tasks are handled by VIGO.



VIGO

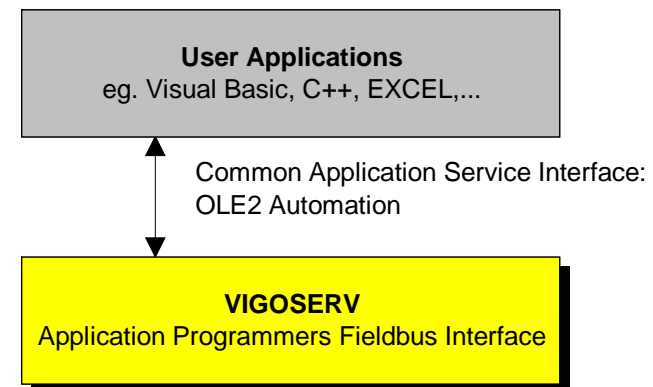




VIGOSERV

The Application Programmers Fieldbus Interface

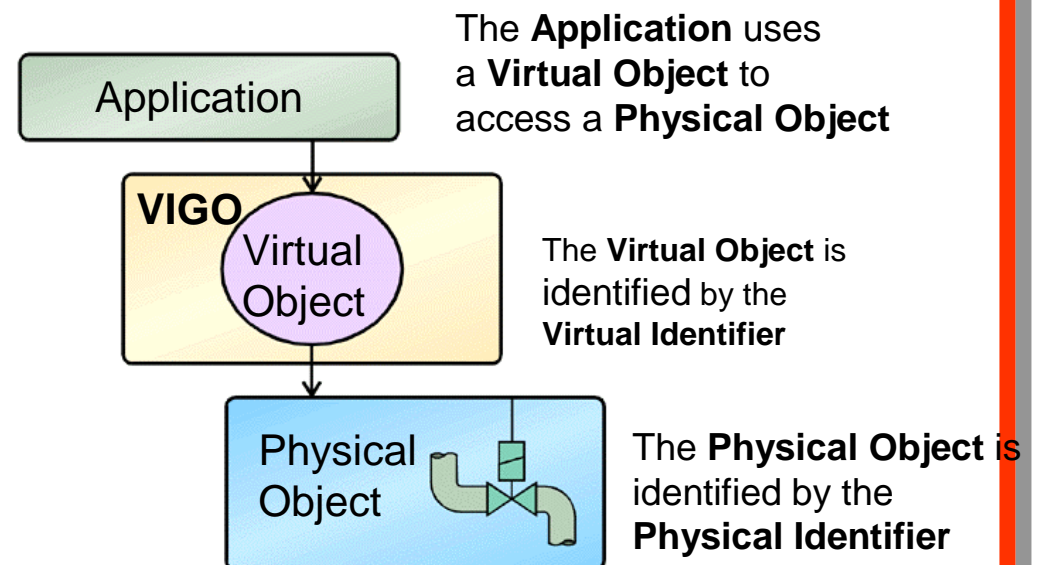
- VIGOSERV, provides a simple interface to standard program packages.
- VIGOSERV is an OLE Automation Server, a part of **Object Linking and Embedding (OLE2)**, a Microsoft Windows facility, for real-time data exchange.
- Standard program packages: Visual Basic, Delphi and Visual C++, spreadsheets, databases etc.





VIGOSERV

- VIGOSERV operate on a physical object via a Virtual object.
- The Virtual objects are created by user applications
- The virtual object is made to point to the physical object, by means of the physical identifier - a unique name.
- The physical identifier is defined in the Manager Information Base (MIB).

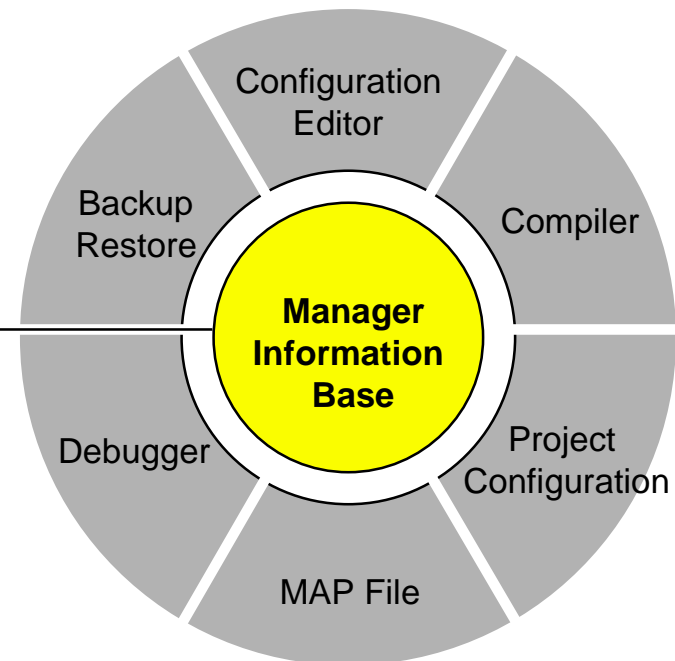




Manager Information Base

- VIGO includes a Manager Information Base - MIB.
- The MIB is used to describe the whole Fieldbus control system.

VIGOSERV
Application Programmers Fieldbus Interface



- A Fieldbus system is constructed of a number of Fieldbus Nodes.



Manager Information Base

- When VIGOSERV requests information from the MIB, using a global identifier, the MIB collates all necessary information about the physical object.
- The MIB describes:
 - how data is structured,
 - how different data elements are related,
 - where data is stored,
 - who has access to that data.



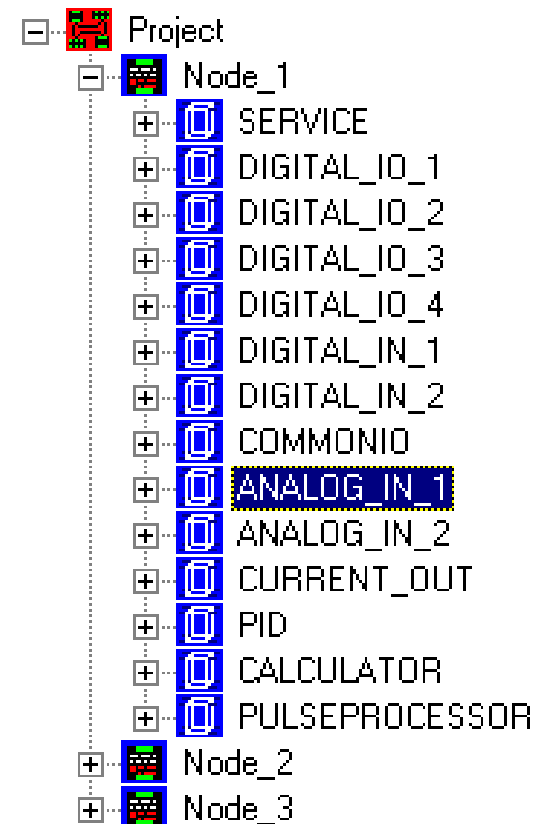
The MIB browser

- The purpose of the MIB browser is
 - to monitor the contents of the MIB,
 - to illustrate the structure of the system.
- The browser function is handled via the MIBOCX as an OLE Control Extension.
- This browser displays a tree-structure, in a similar way to standard file managers.
- The MIB elements are:
 - Project, Nodes, Variables, and Types.



The MIB browser

- This OCX control can be directly called by an object oriented programming language supporting this feature (Visual Basic, C++, Delphi...)
- The OCX control can be included in the users own programmes





The right mouse button within the MIBOCX

- Selecting a Node using the right mouse button results in a menu list relevant for a Node.
- Selecting a Project gives another menu list relevant for a Project.
- The MIB contains all the information required to access a physical object.

Monitor
PD Download
Screendump
Set P-NET Node Address
Show PD4000 controller

Properties



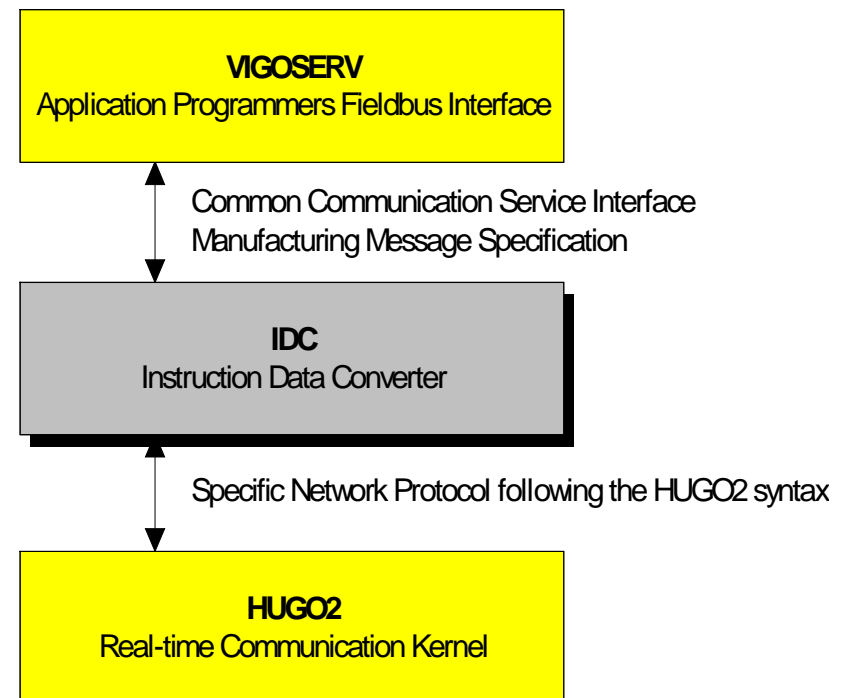
Instruction Data Converter, IDC

- The purpose of VIGO is to have a common interface to any Fieldbus.
- VIGOSERV defines a Common Communication Service Interface.
- A plant can be built, using different Fieldbuses.
- Different Fieldbuses uses dissimilar data formats, and services.
- For each Fieldbus it is necessary to convert to/from various services and data formats.



Instruction Data Converter, IDC

- This conversion is performed by Instruction Data Converters, IDC's.
- One IDC for each Fieldbus system.
- The services between VIGOSERV and the IDC, is based on ISO 9506 (MMS).
- The IDC is a Dynamic Link Library (DLL).



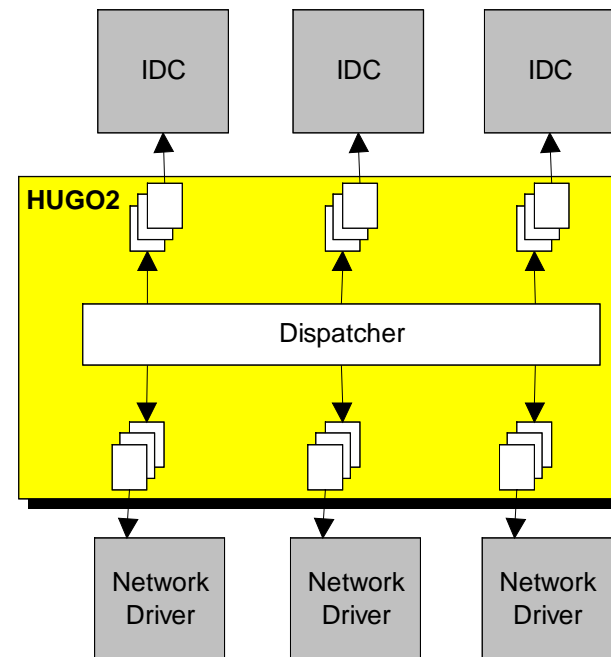


HUGO2

Real-time Communication Kernel

The task of HUGO2 is:

- to manage the queuing and routing of messages,
- to establish error handling procedures,
- and to handle interrupts at different levels.



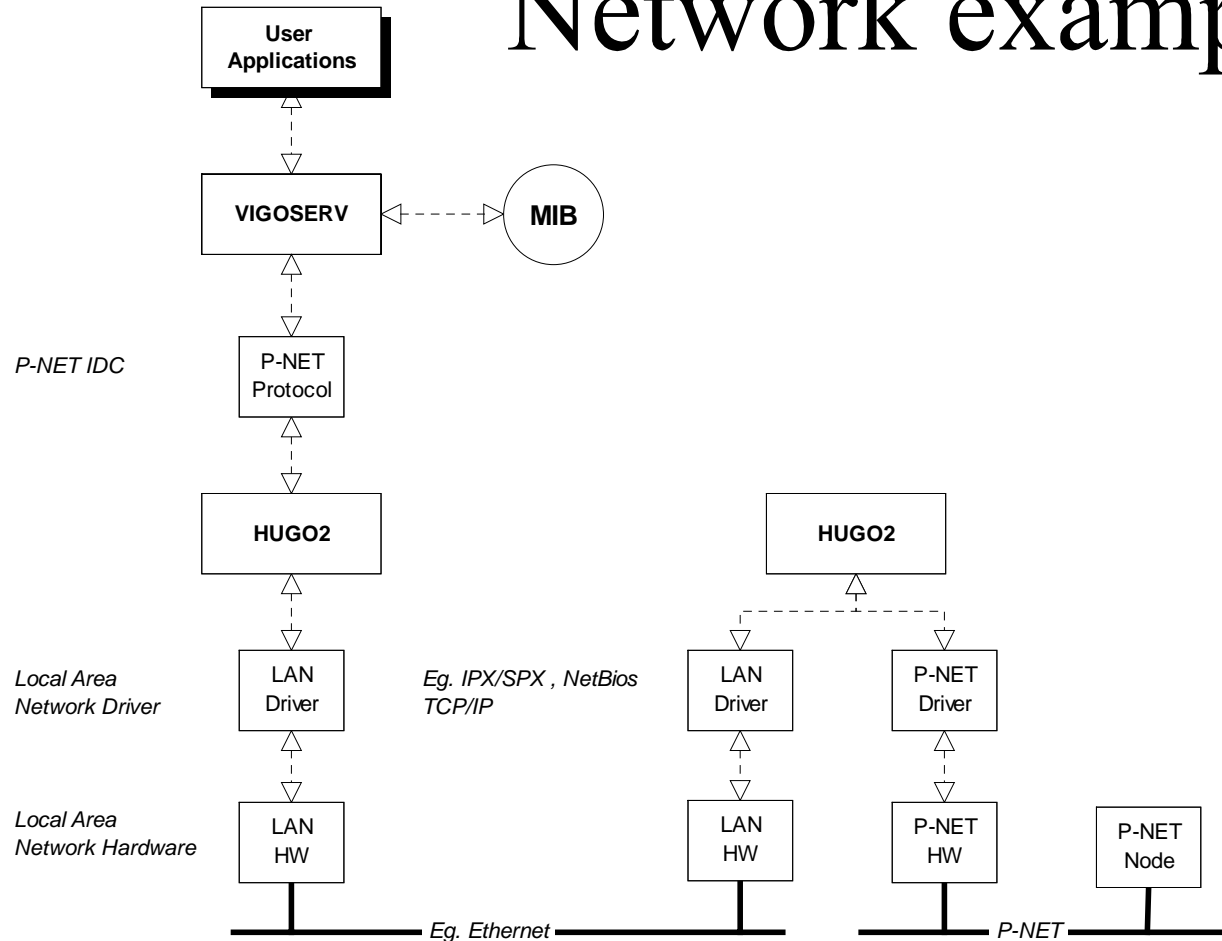


Network Driver

- The network Drivers connects HUGO2 hardware drivers,
 - eg. P-NET or IP.
- Three network types can be connected to HUGO2:
 - Fieldbuses,
 - Local Area Networks (LAN)
 - and Wide Area Networks (WAN).
- The LAN and the WAN types are used for transporting messages.
- Fieldbuses interpret the contents of what is being sent.



Network example.

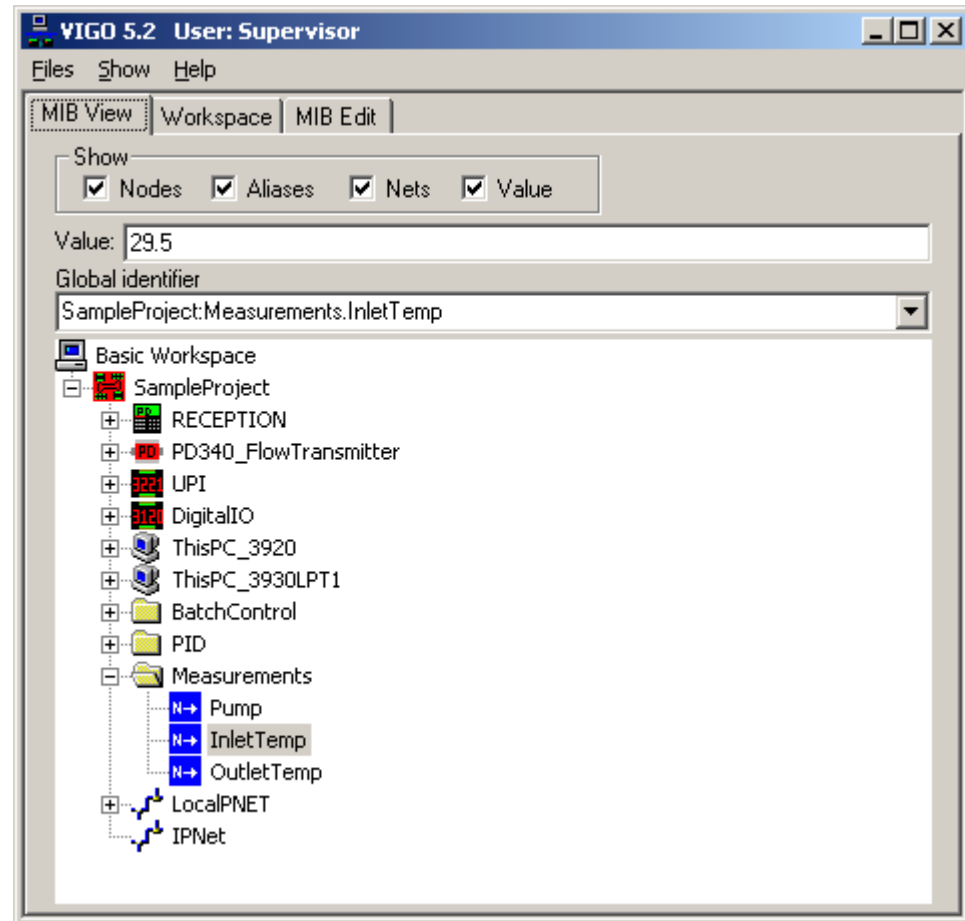




The VIGO main Window

Configuration of VIGO includes:

- selection of network drivers.
- the driver setup
- selection of the project in the MIB.
- and VIGO access conditions





Access to a variable

Step 1: Create a virtual object

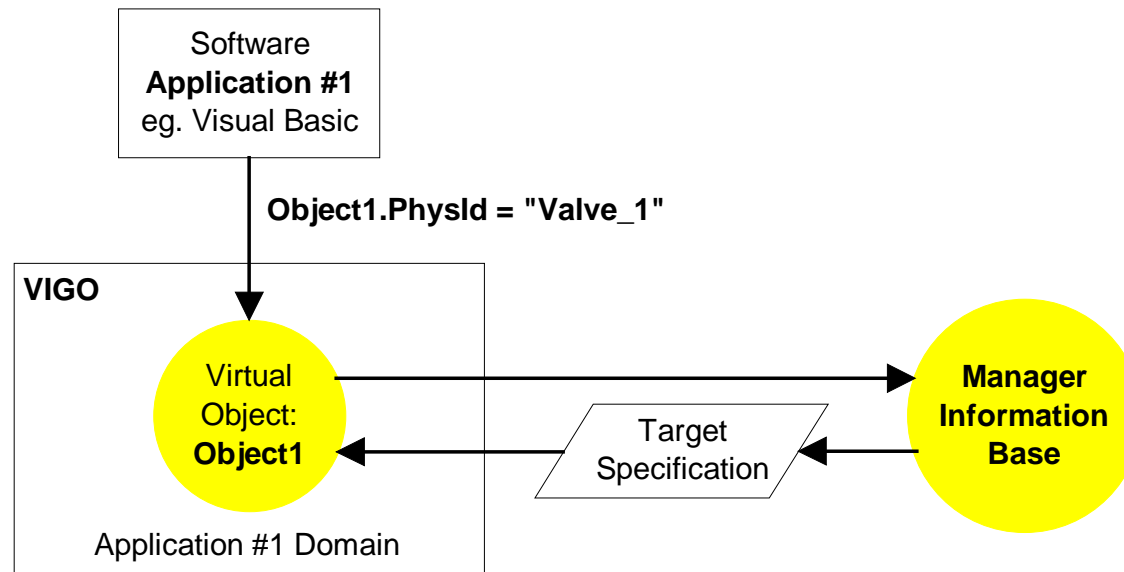
- An example where the virtual object is called "Object1".

set Object1 = CreateObject("VIGO.Std")



Access to a variable

Step 2: Make the virtual object point to the physical object.





Access to a variable

Step 3: Operate upon the physical object.

- Read or Write to the variable.

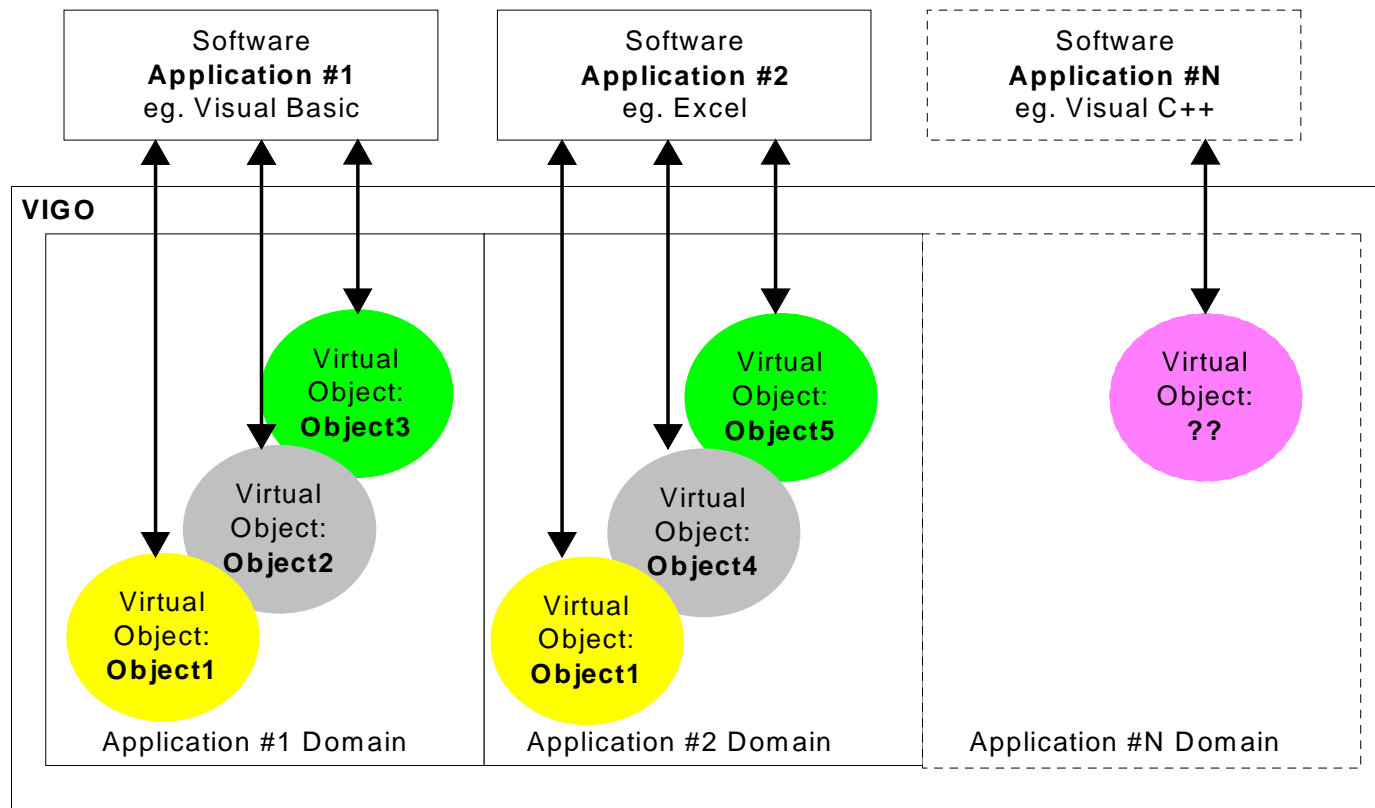
Read example:

- The valve state is read using the following code:

X = Object1.Value



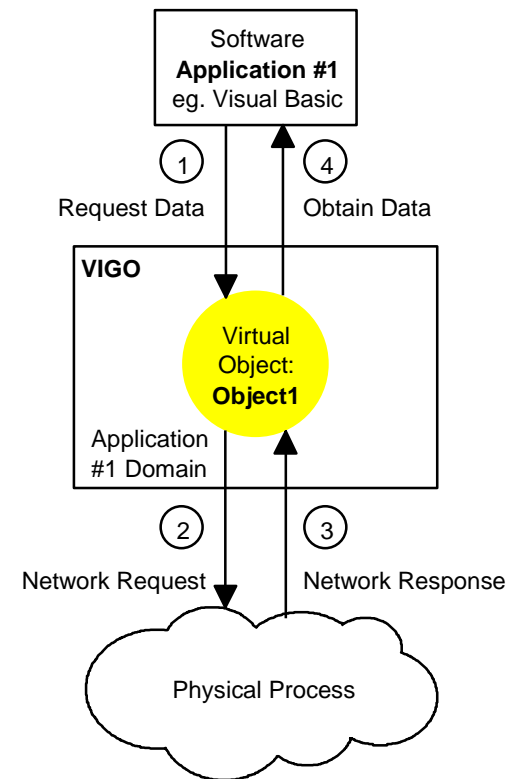
Application domains and shared physical objects





Direct-access

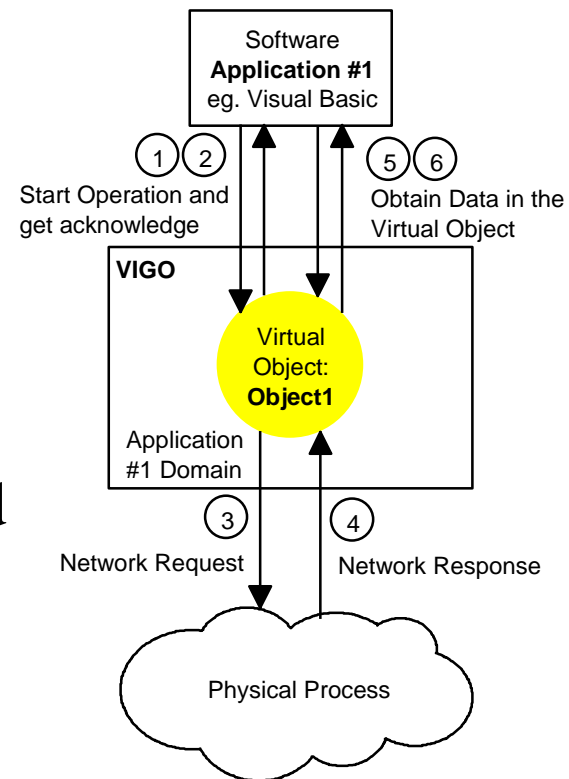
- Direct-access sends a command via the network, waits for the result,
- and then returns to the application when data is obtained from the physical process system.





Buffered-Access

- Buffered-access, sends a command via the network,
- VIGOSERV returns immediately to the application.
- After a while, VIGOSERV will receive the data.
- It is now up to the application to read the result.
- Buffered-access uses two Methods: *DoRead* and *DoWrite*.





Error handling

- Error handling in VIGO follows the OLE Automation Exception rules.
- It is possible to disable and enable Exceptions.
- When a response is received, the ErrorCode Property holds the error code if an error occurs.
- The application can then request a translation of this into a text string, by reading the ErrorString Property.
- The errortext is automatically translated to the default language that is used on the PC.
- The error may have occurred from within VIGOSERV, an IDC, a network or a Fieldbus.



VIGO Tools

Monitor

File Edit Options Help

Simulation Project | Sample Project

SimulationProject:UPI.Service.DeviceID.DeviceNumber	Word	<input checked="" type="checkbox"/>	3221
SimulationProject:UPI.Digital_IO_1.OutTimer	Real	<input checked="" type="checkbox"/>	-842,51
SimulationProject:UPI.Analog_In_1.AnalogIn	Real	<input checked="" type="checkbox"/>	22,35
		<input type="checkbox"/>	
SimulationProject:DigitalIO.Service.DeviceID.DeviceNumber	Word	<input checked="" type="checkbox"/>	3120
SimulationProject:DigitalIO.Service.DeviceID.Manufacturer	String	<input checked="" type="checkbox"/>	Proces-Data DK

Set P-NET Node Address

File Help

Node Identifier:
SampleProject:UPI

Serial Number
9727640PD

Node information

Node Number: 64 (\$40)

Device Type: 3221

Manufacturer: Proces-Data DK



Uniform user interface

The screenshot displays the VIGO 5.2 software interface. The main window, titled "VIGO 5.2 User: Supervisor", has a menu bar with "Files", "Show", and "Help". Below the menu bar are tabs for "MIB View", "Workspace", and "MIB Edit". A "Show" section contains checkboxes for "Nodes", "Aliases", "Nets", and "Value". A "Value:" field shows "False". Below this is a "Global identifier" dropdown menu with "MODBUSProject:ModbusNode.coilstatus[1]" selected. A tree view on the left shows the project structure: "Basic Workspace" containing "SampleProject" and "MODBUSProject". "MODBUSProject" contains "ModbusNode", which contains "PortA" and "coilstatus". "coilstatus" contains a list of coils from [1] to [10].

Overlaid on the main window is a "Monitor" window titled "Monitor - C:\Program Files\VIGO\DynamicData\monitor\default.mon". It has a menu bar with "File", "Edit", "Options", and "Help". The window displays a table of monitored variables:

Variable	Type	Value
SampleProject:UPI.Digital_IO_1.FlagReg[7]	Boolean	<input checked="" type="checkbox"/> False
SampleProject:Measurements.Pump	Boolean	<input checked="" type="checkbox"/> False
SampleProject:Measurements.InletTemp	Real	<input checked="" type="checkbox"/> 29,5
MODBUSProject:ModbusNode.coilstatus[1]	Boolean	<input checked="" type="checkbox"/> True
MODBUSProject:ModbusNode.reportslaveid.slaveid	Byte	<input checked="" type="checkbox"/> 0



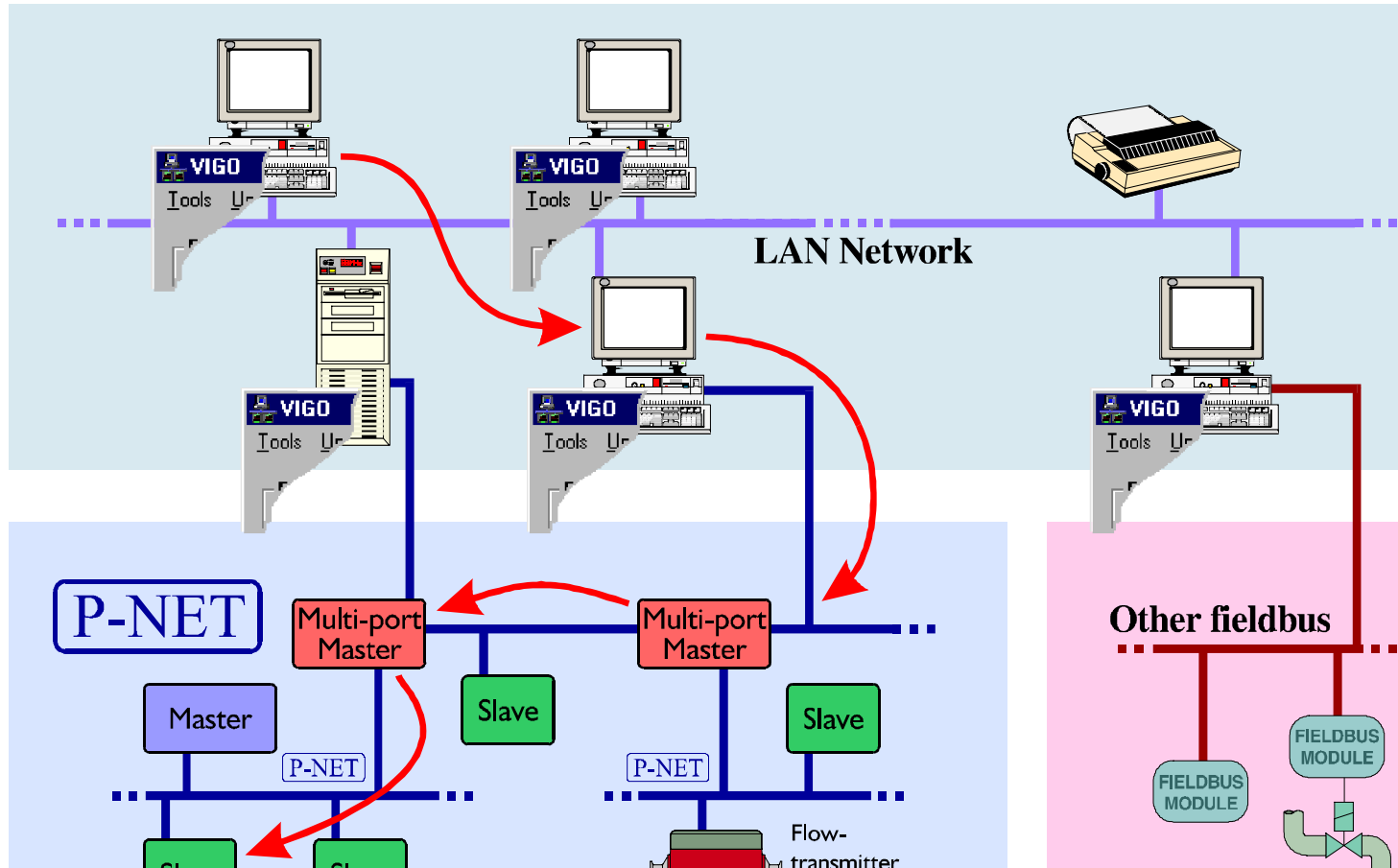
Support for SCADA

Currently, the following SCADA systems supports VIGO:

- Visual VIGO
- FactoryLink ECS
- InTouch
- LabView
- i-FIX
- IGSS
- PRO-view



VIGO and more Networks






www.proces-data.dk


The PROCES-DATA homepage. - Microsoft Internet Explorer


Eiler Rediger Vis Foretrykne Funktioner Hjælp


Tilbage Fremad Stop Opdater Startside Søg Foretrukne Oversigt Post Udskriv


Adresse <http://www.proces-data.dk> Gå Links


 **PROCES-DATA A/S** [HOME](#)


 **PROCES-DATA PRODUCTS**


 **VIGO FIELDBUS SOFTWARE**

 **PROCES-DATA A/S COMPANY PROFILE**

 **THE P-NET FIELDBUS**

 **NEWS!** **NEWS**

 **EXTRA!!** **PRESS - RELEASES!**



PROCES-DATA A/S - the leader in technology and know-how in the area of Process Control and Fieldbus Technology, welcomes you to our Homepage.

Document: Done Internet

PROCES-DATA A/S