



VAN

FP6/2004/IST/NMP/2 - 016969 VAN

Virtual Automation Networks

Work Package 2

Open Platform and System Architecture

Task 2.3

Application specific architecture profiles

Deliverable D02.3-1

Application specific Architecture for
Automation, VAN Profile for Process-
Manufacturing-Industry

Document type	: Deliverable
Document version	: V1.0
Document Preparation Date	: 25.7.2006
Classification	: Public
Contract Start Date	: 01.09.2005
Duration	: 31.08.2009



Project funded by the European Community
under the "Information Society Technology"
Programme (2002-2006)

Rev.	Content	Resp. Partner	Date
V0.1	Document structure	Sajdl, BUT	28/10/06
V0.2	Update document structure First input (Proxy Device)	Hundt, CVS	14/11/06
V0.3	Deadlines, chapter editors	Sajdl, BUT	15/11/06
V0.6	First official draft version	Sajdl, BUT	15/12/06
V0.7	Revision and updates	Sajdl, BUT	25/1/2007
V0.9	Added chapter Architecture refinement, Conformance classes, modified Conventions chapter	Sajdl, BUT	3/4/2007
V0.10	Added chapter 2 inputs, updated VAN-AD & VAN-AP with conformance class tables	Sajdl, BUT	16/4/2007
V0.11	Inputs to chapter 2, 3.5, 3.6, 4.1, 4.2, 4.4, 5.1	Sajdl, BUT	2/6/2007
V0.12	Use cases and updates	Sajdl, BUT	27/6/2007
V0.13	Second public draft version	Sajdl, BUT	6/7/2007
V0.14	Contribution from Aucoteam (chapters 2.5 and 5.2) and MCM (chapter 2.5 and 5.1)	Sajdl BUT, Calegari MCM, Tursch Aucoteam	23/7/2007
V0.15 (V1.0)	Final formatting, conclusion, executive summary, task level review	Sajdl, BUT	25/7/2007

Final approval	Name	Partner
Review Task Level	Ondrej Sajdl	BUT
Review WP Level	Axel Pöschmann	ifak
Review Board Level	Dr. Axel Klostermeyer	Siemens

Executive summary

This document concurs on deliverable D02.1-1 where's been defined general VAN device architecture. The main task of WP2.3 was to specify subsets of this architecture and to create so called architecture device profiles or shortly device profiles for selected devices. Moreover,

- in chapter 2 we described selected parts of general VAN device architecture which we found necessary to be specified more deeply. Also there is stated the necessity of these parts in device profiles. This parts and functionality was refined: Media gateway, MIB and SNMP, IP stack, safety, security and runtime tunnel.
- Chapter 3 describes general use case for all our device profiles and also introduces conformance class concept, which guarantee the scalability of functionality and minimal cost of real VAN devices. Chapter also contains definition of used terms.
- Chapter 4 defines device profiles based on conformance class concept. We've defined profiles and classes for VAN-AD, VAN-PD, VAN-AP and PnP Manager.
- Chapter 5 shows examples how our VAN device profiles can be used in real application. We've focused on examples from manufacturing and process industries because it is a domain of our participants (Aucoteam, MCM).

Contents

Executive summary	3
Contents.....	4
List of figures.....	7
List of tables	8
1 Introduction	9
2 Architecture refinement	10
2.1 Introduction.....	10
2.2 Media gateway	10
2.3 MiB + SNMP.....	10
2.4 Safety	10
2.5 Security.....	10
2.5.1 ACL.....	12
2.5.2 Firewalls.....	13
2.5.3 VPN, IPSec and OpenVPN	14
2.5.4 Web Service and Security	15
2.6 IP stack.....	17
2.7 Runtime tunnel	17
3 VAN device profile description.....	19
3.1 General use cases for VAN devices	19
3.2 Identification of VAN device profiles.....	20
3.3 Definition of used terms.....	21
3.3.1 VAN device profile	21
3.3.2 VAN application profile	21
3.3.3 VAN Access Point (VAN-AP).....	21
3.3.4 VAN Proxy Device (VAN-PD).....	21
3.3.5 VAN Plug&Play Manager	21
3.3.6 VAN Plug&Play Agent	21
3.3.7 VAN Virtual Device (VAN-VD).....	22
3.3.8 VAN Automation Device (VAN-AD).....	22
3.4 VAN device layer definition	22
3.5 Conformance classes concept	23
3.6 Conventions for device profile definition	24
3.6.1 Subchapters for each profile.....	24
3.6.1.1 General overview	24
3.6.1.2 Selection of conformance classes layers elements	24
3.6.1.3 Object model	29

4	VAN device profile definition	31
4.1	VAN Automation Device (VAN-AD)	31
4.1.1	General Overview	31
4.1.1.1	Implementation of Automation Device	31
4.1.2	Use case	32
4.1.3	VAN- AD Object selection	33
4.1.4	Object model	37
4.2	VAN Proxy	38
4.2.1	General overview	38
4.2.2	Use case	39
4.2.3	VAN- PD Object selection	40
4.2.4	Object Model	44
4.3	VAN Access Point	45
4.3.1	General overview	45
	VAN-AP Delivers:	45
	Interacting between VAN devices with VAN-FS regarding:	45
	VAN-AP Receives:	46
4.3.2	Use case	46
4.3.3	VAN-AP objects selection	46
4.3.4	Object Model	50
4.4	Plug&Play Manager	50
4.4.1	General overview	51
4.4.1.1	“Plug and Play” process	51
4.4.1.2	Plug&Play Manager functions	53
4.4.2	Use case	53
4.4.2.1	Example: PnP role-playing between a VAN Engineering station, VAN Automation Devices, and Fieldbus Automation Devices in a defined VAN domain	53
4.4.2.2	Classification of the VAN PnP roles	55
4.4.3	VAN-PnP Manager objects selection	55
4.4.4	Object Model	60
5	VAN Application Profiles Definitions	61
5.1	Manufacturing Industry Profiles	61
5.1.1.1	Manufacturing Device categories	61
5.1.1.1.1	Manufacturing field devices	61
5.1.1.1.2	Manufacturing supervisor devices	62
5.1.2	VAN Application profiles and Manufacturing devices	62
5.1.3	VAN CNC profile	63
5.1.3.1	Device description	63
5.1.3.2	Interfaces description	63
5.1.3.3	Protocols	63
5.1.3.4	Communication requirements	64
5.1.3.5	Object model	64
5.1.4	VAN RFID station profile	66
5.1.4.1	Device description	66
5.1.4.2	Interfaces description	66
5.1.4.3	Protocols	67
5.1.4.4	Communication requirements	67
5.1.4.5	Object model	67
5.1.5	VAN Robot controller profile	69
5.1.5.1	Device description	69
5.1.5.2	Interfaces description	69
5.1.5.3	Protocols	69

5.1.5.4	Communication requirements	69
5.1.5.5	Object model	70
5.1.6	VAN Supervisor unit profile	71
5.1.6.1	Device description	71
5.1.6.2	Interfaces description	71
5.1.6.3	Protocols	71
5.1.6.4	Communication requirements	71
5.1.6.5	Object model	72
5.2	Process Industry Profiles	73
5.2.1	Approach to the definition of the application profiles for the Process Industry	73
5.2.2	Difference between application profiles VAN Solutions in the process industry and in the manufacturing industry	76
5.2.3	Requirements for Process Industry	77
5.2.4	VAN requirements for Application profiles of the Process Industry	77
5.2.5	Device profile objects selection	78
5.2.6	VAN based central control for decentralised plants	82
5.2.6.1	Overview of the evaluation platform for process control specific test scenarios	82
5.2.6.2	Test scenario – closed loop control (bio reactor)	83
5.2.6.2.1	Device Description	84
5.2.6.2.2	Interfaces description	84
5.2.6.2.3	Protocols	84
5.2.6.2.4	Communication requirements	84
5.2.6.2.5	Object model	84
5.2.6.2.6	Required objects	85
5.2.6.3	Test scenario – safety loop control (gas storage)	86
5.2.6.3.1	Device Description	86
5.2.6.3.2	Interfaces description	86
5.2.6.3.3	Protocols	87
5.2.6.3.4	Communication requirements	87
5.2.6.3.5	Object model	87
5.2.6.3.6	Required objects	87
5.2.6.4	Test scenario – open loop control (power station)	89
5.2.6.4.1	Device Description	89
5.2.6.4.2	Interfaces description	89
5.2.6.4.3	Protocols	90
5.2.6.4.4	Communication requirements	90
5.2.6.4.5	Object model	90
5.2.6.4.6	Required objects	90
6	Conclusion	92
	Glossary	93
	References	95

List of figures

Fig. 2-1: Security blocks in the VAN Device Architecture.....	12
Fig. 2-2: The Current Web Service Protocol Stack	16
Fig. 3-1: General use case – an example how all VAN devices can cooperate.....	19
Fig. 3-2: Identification of VAN device layers.....	22
Fig. 3-3: Application of conformance classes – example for VAN-AP.....	23
Fig. 3-4: Object Model.....	30
Fig. 3-5: Legend of symbols used in device architecture	30
Fig. 4-1: VAN-AD.....	31
Fig. 4-2: Implementation of AD control system	32
Fig. 4-3: Use case for VAN-AD	32
Fig. 4-4: VAN-AD class A object model.....	37
Fig. 4-5: VAN-PD and VAN-VD device.....	38
Fig. 4-6: VAN Proxy Application Process	39
Fig. 4-7: VAN-PD use case	39
Fig. 4-8: VAN-PD class A object model.....	44
Fig. 4-9: VAN-AP as gateway (VA1, VA2) and router (VA3, VA4)	46
Fig. 4-10: VAN-AP Class A object model.....	50
Fig. 4-11: VAN Web Service without UDDI	51
Fig. 4-12: PnP scenario in a VAN domain.....	52
Fig. 4-13: VAN PnP roles with VAN Engineering	54
Fig. 4-14: VAN-PnP Manager object model	60
Fig. 5-1: VAN CNC object model.....	65
Fig. 5-2: VAN CNC object model.....	67
Fig. 5-3: VAN Robot controller object model.....	70
Fig. 5-4: VAN Supervisor unit object model	72
Fig. 5-5: Object model for VAN test scenario - gas storage leakage control.....	87
Fig. 5-6: VAN test scenario - parameter switching.....	89

List of tables

Table 2-1: The necessity of IP stack function blocks	17
Table 3-1: Description of profile conformance classes A, B and C	28
Table 3-2: Description of each column previous table	29
Table 4-1: VAN- AD device functions.....	31
Table 4-2: Main functions of VAN-PD	38
Table 4-3: Main functions of VAN-AP.....	45
Table 4-4: Classification of the VAN PnP roles to the device classes and function sets	55
Table 5-1: VAN Manufacturing Application profiles.....	63
Table 5-2: VAN CNC additional blocks	65
Table 5-3: VAN RFID station additional blocks	68
Table 5-4: VAN Robot controller additional blocks.....	70
Table 5-5: VAN Supervisor unit additional blocks	72
Table 5-6: Required VAN devices for process industry demonstrator	78
Table 5-7: Device profile object selection.....	81
Table 5-8: Profile objects for VAN test scenario Bio-reactor	85
Table 5-9: Profile objects for VAN test scenario - gas storage leakage control	88
Table 5-10: Profile objects for VAN test scenario – parameter switching	91

1 Introduction

According to the Open Platform for Automation Infrastructure specification, this task defines application specific architecture profiles of devices in different industrial branches such as manufacturing industries, process industries and communication systems. Especially the following issues were considered:

- the refinement of the architecture according planned demonstrators,
- the definition of mandatory and optional ASEs defined in task 2.2,
- the definition of a subset of ASE for special devices e.g. VAN-AP, VAN-PD, VAN-AD,
- the definition of device profiles for process and factory automation.

2 Architecture refinement

2.1 Introduction

This chapter shall describe selected parts of general VAN device architecture which we've found necessary to be specified more deeply.

2.2 Media gateway

The VAN Media gateway will be used for integration of specialized VAN network technologies in the future steps of the project. Purpose of this block is to connect different VAN technology layers (e.g. Bluetooth, UMTS, GPRS).

VAN Media gateway will be used to build up media gateway connections. There is no specific interface definition for common VAN cases. The interface specification has to be described for each case separately; depending on the implementation and is transparent for application communication.

Furthermore the media gateway will join different VAN segments, e.g. which uses different local IP-address ranges.

According to the VAN communication concept principles, which uses name based addressing; this means a tight communication to the VAN Domain ASE Object for resolving the IP-Addresses by using Web Services.

The original idea is to apply a name based routing process that connects domains with private IP addresses. The physical representation then is e.g. an application routing process. We are convinced that no company can introduce such functionality to IT. No company of the consortium has accordant marked power in this environment. Therefore, this functionality is not used by any device profile we defined in task 2.3.

2.3 MiB + SNMP

MiB is storage of engineering data. SNMP should serve to device monitoring. SNMP will not be used in VAN. No further definition needed. SNMP is recommended to include it because it intrinsically exists in all devices.

2.4 Safety

There are no specific requirements at this point. WP5 made its own architecture extension in deliverable D05.2-1 [VAND08].

2.5 Security

This section deals with the security aspects of the defined VAN device architecture. The VAN security issues have been taken in charge by the VAN WP6 Work Package. VAN WP6 activities are in their second crucial phase that is defined in [VAN07] to be «Definition of security mechanisms in industrial environment addressed by VAN; Catalogue of attack scenarios».

Here we want to investigate about how the VAN security is evolving, what are the main decisions that the WP6 has already taken so far, what is or what could be the impact of these decisions by the VAN device architecture and VAN device profile points of view.

The VAN device architecture was introduced in [VAN06a] and has been modified in [VAN06b].

The most important architecture refinements in the device object model are:

- Introduction of the ACL layer between the Communication Technology Standards group and the Web Services over Virtual Automation Network block.
- VAN Runtime Object Tunnel: direct connection to the Communication Technology Standards group

These changes are both related to the security layer that the “Web Services over Virtual Automation Network” block needs.

The Fig. 2-1 depicts the VAN device architecture model with all the VAN specific blocks highlighted: the main security blocks (4) are green painted.

The ACL (Access Control List) was directly introduced as an addition block and it is now considered a mandatory security feature in all VAN devices (see 2.5.1).

The Access Control List implements a security mechanism, based on a permission list, which is very device specific. This means that a software program that performs the ACL functionality has to be run on the device and that an ACL configuration database (a text file for example) has to be saved in the device: the ACL mechanism uses device resources and requires a management function to be up and running on the device because the permission list could be updated during runtime and it has normally to be downloaded from a management station in case of a device replacement.

It is clear that the ACL is not the only security measure that it has been selected for VAN devices and VAN networks, because the interoperability between devices that belongs to different VAN network segment, the use of WAN networks and especially the use of the Internet as a communication base for the remote Maintenance and the remote Engineering functions, require a global security strategy to be applied to VAN. Without a global analysis of the automation functions that will be implemented in VAN networks, it is difficult to understand how a particular security measure can influence VAN device designing and prototyping.

In this global security view a set of security mechanisms that can cover all the possible network and device vulnerabilities have to be chosen. The set of security mechanisms that has been identified by the WP6 to be suitable for VAN is:

- Access Control List,
- firewalls,
- Virtual Private Networks,
- Web Services security.

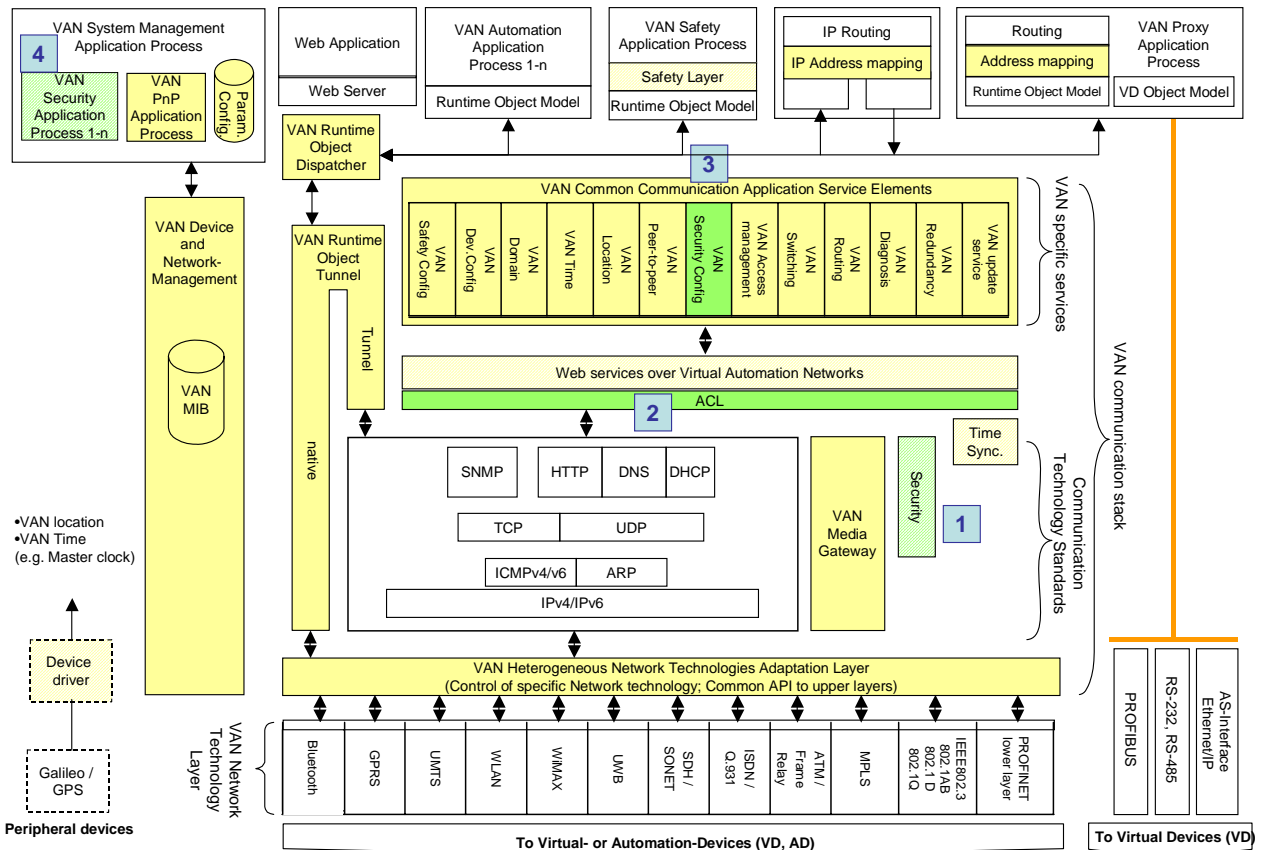


Fig. 2-1: Security blocks in the VAN Device Architecture

In the following sections these mechanisms will be analysed by the VAN device architecture point of view.

2.5.1 ACL

Every VAN device should have the option to take care for some connection security measures by itself. This can be accomplished by defining Access Control Lists (ACL), which is related to the Application Service Element (ASE) Security within the device architecture. So, every device, be it a VAN automation device or a VAN security device, can have access to a control mechanism, that allows or denies connections. ACL is a concept for computer or network security, which is used to enforce privileges of users or systems according to some existing security policies. It is a method to determine access rights to a specific object or service depending on certain identity aspects of the

request making communication partner. This list is a data structure that contains the entries for the different users or groups, which specify their rights to access the device.

For the VAN device, the entries of the ACL should be able to have the identification attributes:

- IP address / IP address range,
- name / name space,
- object,
- method,
- certification,
- external authentication.

Every communication partner requesting a connection should be checked for any single or any combination of these attributes and the related access rights. According to its security rules the device can then allow or deny the establishment of a connection or access to an object. If the device decides to deny the connection, it can either simply drop it without additional information, or it rejects the connection and send back comments. This annotation can span from a general message about the denial up to detailed information for the reason and the identification of the rejecting device.

The VAN engineering should provide functions to create such ACL and to load them into the VAN devices.

2.5.2 Firewalls

A firewall is a network software program running on a machine (software firewall) or a network stand-alone device (hardware firewall) that permits or avoids well defined types of communications to flow inside/outside a particular system, network or a bordered network area that is usually called DMZ (DeMilitarized Zone).

Firewalls are an important part of the overall VAN security architecture. The main limiting factor for the usage of firewalls in the automation area is the latency or the delay. This latency shall be investigated for freely available firewalls (pf) or for purchasable standard firewalls.

It shall be investigated how the firewall latency can be reduced by the usage of dedicated hardware. The work includes the specification of the kind of firewall roles which are necessary for VAN. Also included is the development of a firewall architecture which can be implemented in hardware.

When using firewalls we have to keep in mind a multi level network architecture with some different security zones (DMZ) in which a global network distributed system can be thought of.

Firewalls and IP filters can be deployed between any network levels for securing different zones with different security efficiency: in the level of the assured security and in the response time of the connections they have to keep running.

Generally, the lower the network level the fast has the firewall to be. Normally high network level connections or WAN connections don't need so faster firewall behaviours. First because the remote applications that run remotely don't usually need to be as fast as the network applications that are running within an automation cell or in a local network, and secondly because the throughput's bottleneck of a lower rate WAN point-to-point connection can always be found outside of the firewall device itself, so the introduced latency could be calculated to have a very small influence on the connection speed reduction. Note that this has nothing to do with any real-time behaviour of a remote connection: only the measured circle time/response time and its jitter are significant in any consideration about real-time communications.

A Firewall implemented on a dedicated hardware could be defined as a VAN device, a VAN-SID device as it was defined in [VAN06a] the Security Infrastructure Device category.

In a distributed and heterogeneous network environment like a generic VAN Domain, the automation device concept have to be enlarged to include any infrastructure element of the VAN Domain that can have an influence on any network distributed automation function.

As a possible use case that explains this need we can consider a VAN Engineering application (local or remote) that wants to implement a distributed automation function, and thus wants to make a link, between two devices that belong to two different VAN Segments that are connected through a hardware Firewall.

It should be clear that the Firewall configuration has an influence on the connection between the two end points, so the associated VAN application depends on the behaviour and the specific Firewall device configuration.

This thought can be as the same as for a possible VAN Maintenance function that have to be performed remotely on one of the two automation devices that are included in our application example.

The Firewall device has a set of configuration parameters that have to be defined (and have to be saved somewhere in a network computer) for every cross-border network application that have to be run in the VAN-Domain it is belonging to.

An example of a Firewall configuration parameter is the set of IP addresses that are permitted to pass thru the network border.

If we can add a new VAN Maintenance station (or a new VAN Engineering Client) that has its own network IP address, the Firewall configuration have to updated with this new address.

Firewall configuration shall be independent from the type of firewall. Therefore a configuration language for firewalls based on XML shall be developed. If that were true a unique VAN Firewall device profile could be defined to be used in VAN networks.

2.5.3 VPN, IPSec and OpenVPN

VPN stands for Virtual Private Network. VPN is the term used to refer to any device that is capable of creating a semi-permanent encrypted tunnel over the public network between two private machines or networks to pass non-protocol specific, or arbitrary, traffic. This tunnel can carry all forms of traffic between these two machines meaning it is encrypting on a link basis, not on a per application basis.

One of the key elements of VPNs is encryption. To protect sensitive or non routable data as it passes over the public Internet, we need to create a virtual private tunnel. This tunnel is built by encrypting the packets or frames and then encapsulating these in regular IP traffic between the two hosts or networks.

IPSec is a standard set of protocols and rules for their use that allow the creation of VPNs.

IPSec creates a secure tunnel by first using a handshake protocol called Internet Key Exchange (IKE).

IKE authenticates the end points of the tunnel to each other, and then follows a secure procedure to exchange the necessary information to create a more permanent tunnel using symmetric encryption. Once this tunnel is in place, any arbitrary traffic sent between these two end points will be passed through the protected tunnel.

IPSec was formalized by the IETF (Internet Engineering Task Force) to be used as a standard so the different VPNs vendor products will have been interoperable each other, but many problems practically arose, especially about the configuration complexity of IPSec and for its interaction with the Operating System kernel. Another problem to be mentioned here is that IPSec doesn't support NAT (Network Address Translation) so that is not generally usable for a device-to-device connection in a VAN Domain when one of these two devices is placed inside an automation cell.

A second and more successful approach that have been applied for creating VPN connections is the usage of the user-space SSL/TLS based VPN.

Nowadays most of the VPNs are SSL/TLS based and one of the most widespread SSL VPN is the OpenVPN, that is an open source product.

In a SSL VPN the connection between the two end points is established on a socket level (TCP), while in the IPsec VPN the link is established on the IP level.

This user-space OpenVPN way to build-up secured connections avoids complexity problems, provides more flexibility in porting to different operating systems and improve the overall security.

The OpenVPN was chosen as the VAN mechanism for the establishment of a secure remote connection between devices.

OpenVPN uses the UDP protocol and tunnels traffic over the Port 5000. There is a well known documented reason for the usage of the UDP protocol instead of the TCP.

We don't want to give here a detailed description for why the UDP protocol is used, but if we consider a TCP traffic that has to be tunnelled on an OpenVPN encapsulated TCP connection we have a TCP tunnel riding on top of TCP, and that can cause an instability in the flow control of the resulted link: TCP keep tracks of packet sequence and of packet loss and automatically resent a missing packet with an adaptive timeout mechanism.

This doesn't happen if we use UDP that doesn't check for lost packets.

From the VAN device profile point of view this means that all VAN devices which could be the end point of an OpenVPN tunnel connection must have the UDP protocol mandatory by default.

2.5.4 Web Service and Security

As described in this deliverable the communication between the devices and the access to the objects of the Application Service Element (ASE) should be organised via Web Services. As this communication technology lies within the application layer of the network model, it can interconnect different communication partners and the devices don't need to handle with the underlying protocols.

The communication takes place with SOAP messages, which are transported usually with HTTP, but other application layer transport protocols are also useable, like FTP or SMTP. From the security point of view this solution covers several aspects. With the use of these standard communication protocols, it is easy to configure existing firewalls. Furthermore, there exists already secured versions for these protocols, like HTTPS or FTPS, which can be used for an encrypted transfer of data. Web Services itself encompass a set of protocols, where some refers to its security, too.

Function	Specification, Standards, Description	Group
Business Domain Specific extensions	Various	Business Domain
Distributed Management	WSDM, WS-Manageability	Management
Provisioning	WS-Provisioning	
Security	WS-Security	Security
Security Policy	WS-SecurityPolicy	
Secure Conversation	WS-SecureConversation	
Trusted Message	WS-Trust	
Federated Identity	WS-Federation	
Portal and Presentation	WSRP	
Asynchronous Services	ASAP	Transactions and Business Process
Transaction	WS-Transactions, WS-Coordination, WS-CAF	
Orchestration	BPEL4WS, WS-CDL	
Events and Notification	WS-Eventing, WS, Notification	Messaging
Multiple Message Sessions	WS-Enumeration, WS-Transfer	
Routing / Addressing	WS-Addressing, WS-MessageDelivery	
Message Packaging	SOAP, MTOM	
Publication and Discovery	UDDI, WSIL	Metadata
Policy	WS-Policy, WS-PolicyAssertion	
Base Service and Message Description	WSDL	
Metadata Retrieval	WS-MetadataExchange	

Fig. 2-2: The Current Web Service Protocol Stack

The most important standard for security with Web Services is WS-Security. This communication standard was accepted by the Organization for the Advancement of Structured Information Standards (OASIS) in 2004 as Web Service Security v.10 (WSS v1.0). This standard is freely available [WSS04].

It describes an enhancement of SOAP messages to provide integrity and confidentiality's of the data and provides a general-purpose mechanism for associating security tokens with message content. It is intended to be used within other security models like PKI or Kerberos.

Within that standard an abstract message security model in terms of security tokens combined with digital signatures is specified. Message integrity is provided by XML Signature [W3C02a] in conjunction with security tokens to be able to detect modifications of messages. By using XML Encryption [W3C02b] in conjunction with security tokens to keep portions of a SOAP message confidential the aimed confidentiality will be reached. But for establishing a security context and for acquiring the keys additional mechanisms have to be used.

By implementing these security mechanisms, applications can engage in secure communication designed to work with the general Web Services framework.

It must be checked by the Security Work Package what standards are relevant for VAN and what is the status of the standards. Because the Web Services are quite new defined not all standards may be in a final state.

2.6 IP stack

The necessity of IP stack function blocks describes following table:

Block	Description/Discussion	Necessity in VAN profiles
IPv4/v6	Internet Protocol stack. All devices with Web Services rely on this because are based on HTTP and HTTP is based on TCP/IP.	Mandatory for all
ICMPv4/v6	ICMP is an integral part of IP and is implemented by each device with an IP stack. All devices with Web Services rely on this.	Mandatory for all
ARP	ARP is a core protocol of the Internet protocol suite and is typically used for error responses in IP datagrams or for diagnostic and routing purposes. Inseparable part of IP stack.	Mandatory for all
TCP	Transmission Control Protocol - one of the core protocols of the Internet protocol suite. The protocol guarantees reliable and in-order delivery of data from sender to receiver. All devices with Web Services rely on this.	Mandatory for all
UDP	User Datagram Protocol - UDP does not provide the reliability and ordering that TCP does. Compared to TCP, UDP is required for broadcast (send to all on local network) and multicast (send to all subscribers).	Mandatory for all
SNMP	Simple Network Management Protocol - is part of the Internet protocol suite and supports monitoring and control of network attached devices (router, server, switches) for any conditions in administration. In VAN it is used in local monitoring access.	Optional
HTTP	Hypertext Transfer Protocol - All devices with Web Services rely on this.	Mandatory for all
DNS	Domain Name System - stores information associated with domain names in a distributed database on networks. Because VAN is based on domain names, it should be mandatory for all devices.	Mandatory for all
DHCP	Dynamic Host Configuration Protocol – client/server protocol based on UDP which takes care about assigning IP address based on MAC address.	Mandatory for all

Table 2-1: The necessity of IP stack function blocks

2.7 Runtime tunnel

Runtime tunnel is a component which permits communication between the automation application (and its runtime object model) and the lowest layers of the stack (API and drivers of a particular

communication HW interface). To remind the VAN device architecture, please see Fig. 3.3, further in this document.

The idea is to preserve the functionality of the runtime object model of the existing automation technology (e.g. PROFINET IO) and overcome the shortcomings of it. From the runtime communication the most limiting feature is impossibility to communicate beyond boundaries of a single LAN.

Therefore, the communication has to be split into two passages, each with a different mission. The left one would be used for fast runtime communication with another remote device, wherever it is located. The right passage would be used for engineering, monitoring and for secure establishment of a host2host tunnel which would be used for a runtime communication of the left passage.

First, and apparently the most suitable seemed to be introducing RToUDP protocol. It has already been published in a specification of the PROFINET IO and is being worked on by Siemens in terms of implementation. RToUDP is capable of communication over LAN boundaries and has immensely low latency when passing the stack.

However, real-time behaviour was not the only examined aspect. From the security point of view, this approach was not sufficient. Implementation of RToUDP uses specialised NIC (ERTEC - EB200) without any direct connection to the intrinsic TCP/IP entities of the operating system. On the other hand, traffic generated in the ERTEC cannot be let into potentially insecure network as such.

The best agreed approach of securing VAN communication is OpenVPN tunnelling. The basic idea is that the tunnel ingress/egress has to find itself in a device (host2host tunnelling). Only this approach eliminates the most of the security threats introduced in T6.1 and T6.2. Using RToUDP implemented in ERTEC and binding it to an instance of an OpenVPN (which automatically means another NIC) is far from being straightforward and far from exploiting the dominant feature of the ERTEC (low latency of passing stack). Furthermore, not complete implementation and support of this technology is available at the moment.

As a solution, the agreed approach is to use intrinsic PROFINET IO without RToUDP. In runtime phase this means L2 communication. Tunnel using L2 (TAP) brings in less data overhead. This does not mean that RToUDP is out of scope of the project.

Real-time behaviour of the communication will not be therefore supported by a fast stack in an end device. However, it will be possible to influence flow classification in the OpenVPN instance in terms of QoS which is far more important. Further information on this topic is introduced in D04.3-1. Furthermore, security requirements will be fulfilled, which is important.

One of the latest findings regarding the availability of the complete PROFINET IO stack directs the attention of the TechPCC to investigations on Telecontrol profile of the PROFINET CBA specification, which could turn out to be even more suitable for the VAN purposes.

Other types of communication (monitoring, engineering), which do not require strict deterministic behaviour will be established over WS. This is the latter (right) passage through the runtime tunnel.

3 VAN device profile description

3.1 General use cases for VAN devices

Until the spreading of the VAN standard, a typical industrial configuration should allow the coexistence of VAN devices and traditional devices. The following scheme describes such general configuration and the way in which all our VAN devices can cooperate.

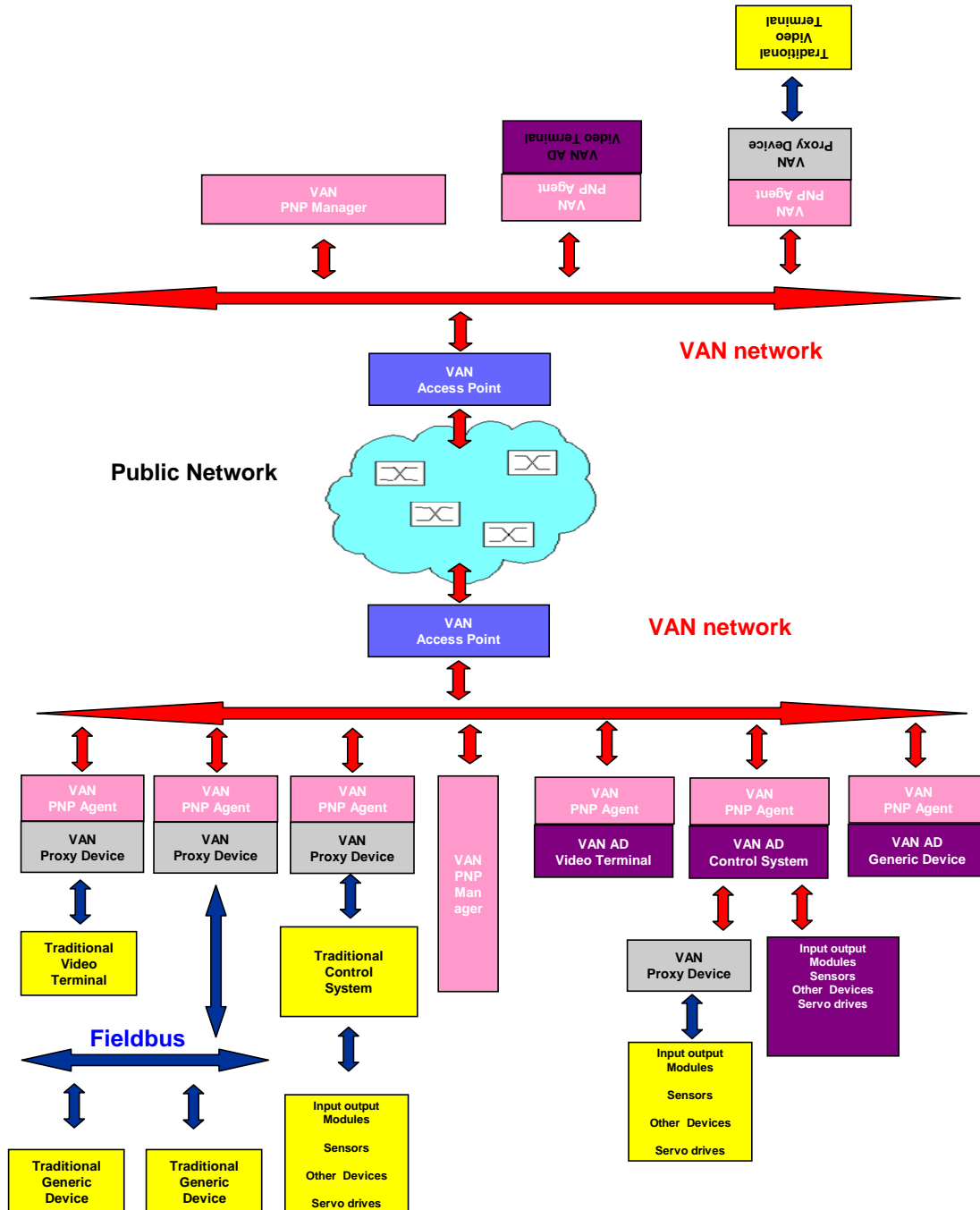


Fig. 3-1: General use case – an example how all VAN devices can cooperate

Here follows a legend of the used colours:

- Red arrows: VAN Network
- Blue arrows: traditional fieldbus
- Yellow block: traditional devices
- Violet block: VAN Automation Devices
- Grey block: Proxy Device
- Pink block: PnP Agents and Manager.
- Light blue: VAN Access Point

As can be seen from the diagram the use case shown is divided into two main parts: a generic automation system (the half below the public network) and a remote control network (the upper half of the diagram). Of course this is only an example, since on both the sides of the public network we could find control terminals and automation devices. The main communication stream flows through the VAN networks. Note that each device connected to the network requires the minimal VAN capabilities (Conformance Class A) as well as a PnP Agent.

At the startup the PnP Manager exchanges information with all the PnP Agents of the devices connected to the VAN Network, identifying them and their respective characteristics; in this way the PnP Manager can compile a list of all the available PnP Agents (note that there is a PnP Manager for each local VAN Network); given the particular importance of the PnP Manager we can suppose it is granted a C Conformance Class. Since we cannot expect to find only VAN Devices in an traditional automation system (at least in a first time), the presence of a VAN Proxy Device is required in order to allow seamless communication for all the devices missing the minimal VAN requirements; the VAN Proxy Device grants the device to which is linked with the minimal VAN functionality needed to connect with the VAN Network (Conformance Class A); if required, an entire local fieldbus network can be mirrored by a VAN Proxy Device. It is worth noting that also the sensors and servo drives must be equipped with VAN functionality (Conformance Class A or B) if they are to be connected to a VAN Control System. Any communication that has to move from a local VAN Network to another one through the public network will require a VAN Access Point capable (among the other things) to grant the security of the data.

3.2 Identification of VAN device profiles

Based on discussion across WPs and according shown use case, three device profiles were identified as basic and essential for the VAN:

- VAN-AD,
- VAN-PD,
- VAN-AP.

Additional in each device should be VAN PnP Agent FS included. In each IP address space one device (with VAN network capability and VAN PnP Manager FS) is required. Therefore is not necessary to define own profiles for this (see chapter 3.3.5 and 3.3.6)

3.3 Definition of used terms

3.3.1 VAN device profile

As is stated in DoW and in D02.2-1 task 2.3 should define subset of ASEs for special devices e.g. VAN-AP, VAN-PD, VAN-AD. This subset we call a device profile. It is subset of general VAN device architecture already defined in D02.2-1. Therefore, device profile defines only VAN communication functionality. Moreover, each profile specifies three possible levels of functionality called conformance classes (see chapter 3.5)

3.3.2 VAN application profile

Application profile specifies optional and recommended functionality for selected conformance class of VAN device profile according application requirements. It specifies concrete application functionality of one or more VAN devices.

3.3.3 VAN Access Point (VAN-AP)

A VAN Access Point defines an entity containing VAN-FS and connects VAN network segments but it does not contain an automation function or an automation application process. It can work as a gateway or router to automation devices which are members in a VAN application context (VAN Domain).

3.3.4 VAN Proxy Device (VAN-PD)

A VAN Proxy Device defines an entity containing VAN Function Set (VAN-FS) realizing the VAN Network capability and Application Proxy Function. It is a proxy to automation devices (VAN Virtual Devices) in a VAN application context (VAN Domain) which can be accessed within the VAN Domain name space.

With a VAN-PD an entire local fieldbus network can be connected to a VAN Network. Only those devices of the local fieldbus network that are "mirrored" into the VAN network are called VAN Virtual Devices. Each VAN-VD can be addressed separately within the VAN Network.

3.3.5 VAN Plug&Play Manager

PnP Manager is standalone VAN device.

The VAN PnP Manager also acts as a gateway to fieldbus systems to make the PnP specific fieldbus traffic available. The realization depends strongly on the corresponding fieldbus system.

The VAN Engineering uses the VAN PnP Manager as server which provides the current network situation (available devices, topology information, connection attributes, and so on).

A VAN PnP Manager has the general states un-configured and configured. An un-configured VAN PnP Manager has no information about the planned devices. As consequence an un-configured VAN PnP Manager scans a VAN domain without restrictions.

A VAN PnP Manager will be configured by the VAN engineering. A configured VAN PnP Manager has information about the planned devices. The domain scan can be restricted in this case.

3.3.6 VAN Plug&Play Agent

The PnP Agent functionality an embedded functionality in all device profiles. Each VAN device shall have the VAN PnP Agent functionality. This functionality comprises objects and services which are used for the address assignment, naming, identification, and announcement of a VAN device. These objects and services are used by the VAN PnP Manager to perform the VAN PnP functions.

The VAN PnP Agent shall contain a DHCP client and/or a service for setting the network address. For the naming also a service shall be available.

3.3.7 VAN Virtual Device (VAN-VD)

A VAN Virtual Device defines an entity without VAN network capabilities but with at least one automation function or one automation application process in the VAN application context.

3.3.8 VAN Automation Device (VAN-AD)

A VAN Automation Device defines an entity containing VAN services and protocol implementation and does contain at least one automation function or one automation application process in the VAN application context.

3.4 VAN device layer definition

According the definition in D02.2-2 chapter 3.3 a VAN device comprises three major layers:

- VAN Network Layer,
- VAN Communication Stack,
- VAN Application Layer.

Fig. 3-2 shows the classification of the VAN Device object model into these three major layers. These layers are used for profile definition template (see chapter 3.6.1.2)

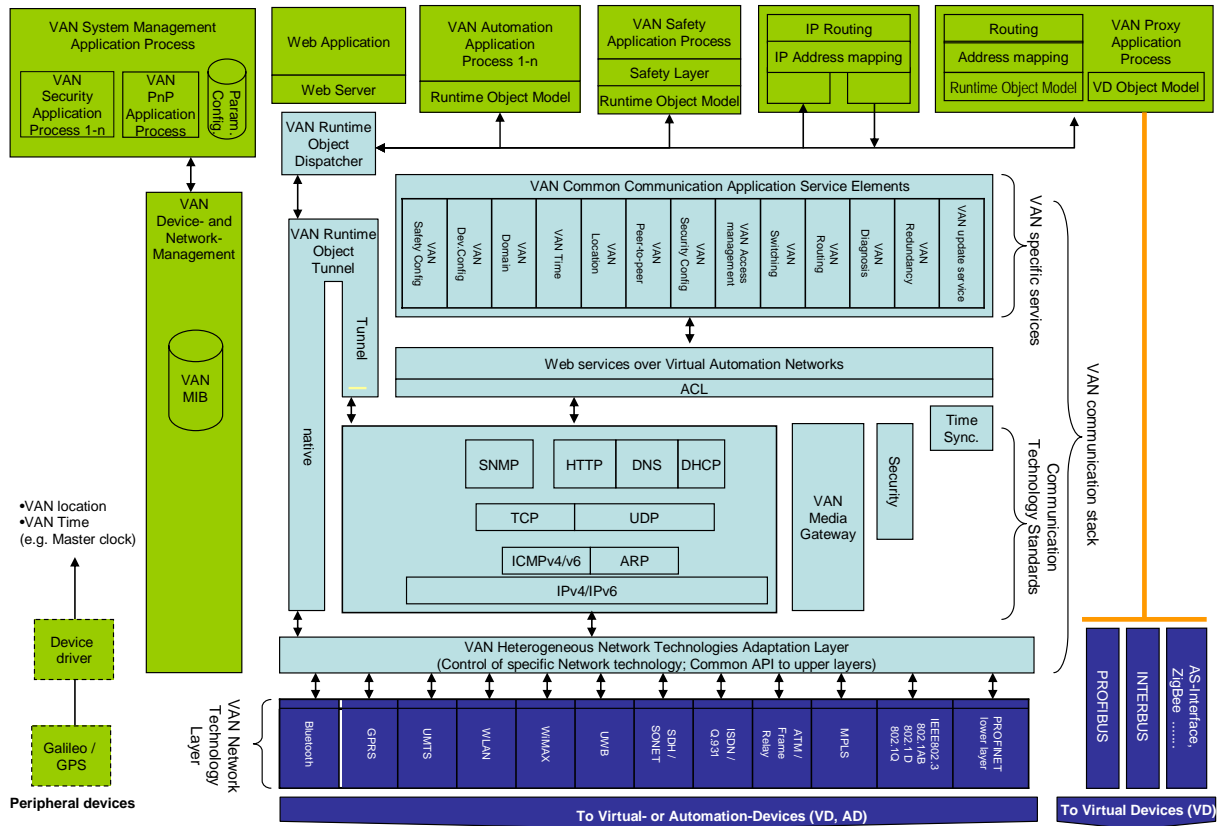
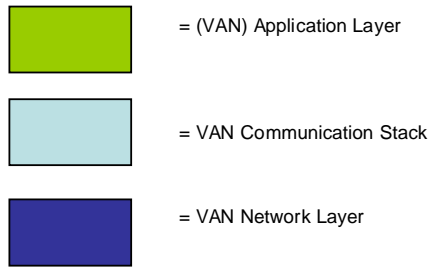


Fig. 3-2: Identification of VAN device layers



3.5 Conformance classes concept

The conformance class concept is introduced to guarantee a scalability of functionality and minimal cost for VAN devices. For this purpose tree conformance classes are defined. These are Class A for devices with the basic VAN functionality, Class B with advanced VAN functionality and Class C with sophisticated VAN functionality. VAN functionalities in this context means the VAN Network capability and not the automation function of the device.

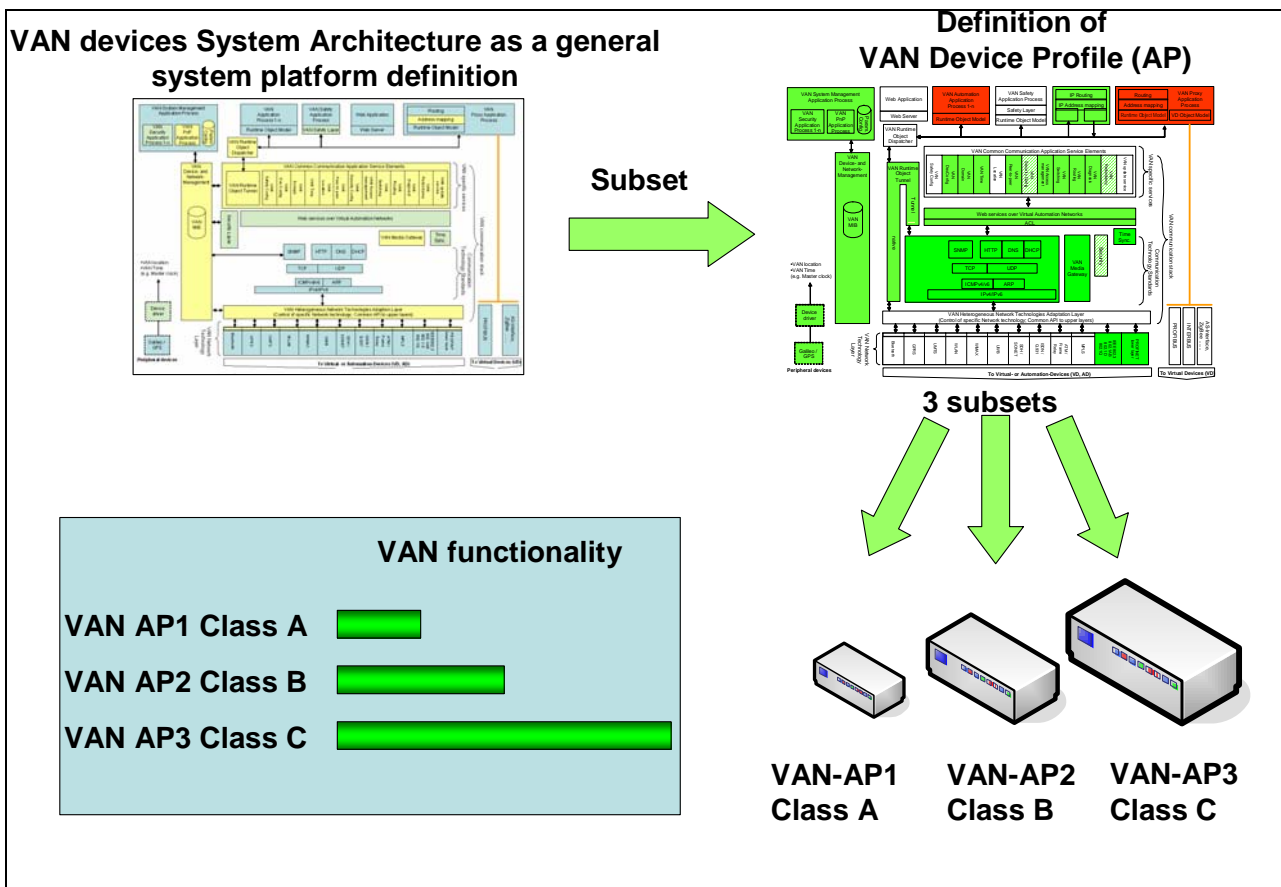


Fig. 3-3: Application of conformance classes – example for VAN-AP

Fig. 3-3 shows the basic idea behind the tree conformance classes for device profiles. Each device profile describes an object subset of the VAN devices which are necessary to realize its functionality in the VAN context. It is not in every case useful/necessary to have the complete functionality in each

device. For this reason the device conformance classes are introduced. Below you can find an overview about the three classes.

CLASS A

Conformance Class A defines the basic subset of VAN objects, which have to be implemented in a VAN device. Class A devices achieve the minimal requirements which are necessary to be visible and to communicate within the VAN context.

CLASS B

Conformance Class B defines the common subset of VAN objects, which should be implemented in a VAN device. Class B devices provide a common VAN functionality this includes all common functionality of their device profile.

CLASS C

Conformance Class C defines the convenience subset of VAN objects for a VAN device. Devices which belong to Class C provide the whole possible VAN functionality of their device profile.

3.6 Conventions for device profile definition

3.6.1 Subchapters for each profile

Each of the device profile definition should be divided in following subchapters:

- general overview,
- description of all layers with table,
- object model.

If it is necessary additional subchapters may be used for the definition of the device profiles.

3.6.1.1 General overview

The chapter General overview shall give an overview about:

- the required functionality should be covered by devices of the device profile,
- the properties of devices of the device profile.

Additional the chapter may be used to give an example for use and real implementation respectively of the device profile.

3.6.1.2 Selection of conformance classes layers elements

Table 3-1 should describe selection of conformance classes elements from general VAN architecture layers defined before: VAN Network Technology Layer, VAN Communication Stack and Application Layer.

Major layer	Feature/Object	Description	Class A	Class B	Class C	Remarks
VAN Network Layer	VAN Heterogeneous Network Technology Adaptation Layer	See D02.2-1 chapter 2.4.10				
	IEEE 802.3	Access to standard Ethernet network.				
	PROFINET lower layer	Standard PROFINET Driver				
	ATM / Frame Relay					
	ISDN / Q .931					
	SDH / SONET					
	UWB					
	WiMAX					
	WLAN					
	UMTS					
	GPRS					
Bluetooth						
VAN Communication Stack	VAN Runtime Object Dispatcher	Dispatches automation objects conveyed via the runtime object tunnel.				
	VAN Runtime Object Tunnel	The tunnel is an interface to the application layer.				
	Web Service over Virtual Automation Networks	Provides the WS of the VAN device				
	ACL	Access Control Layer assures the authorised access to objects in the VAN device.				
	VAN Common Communication Application Service Elements	For all ASE see API specification in D02.2.-2				
	VAN Domain	connection parameters (incl. VAN Runtime				

Major layer	Feature/Object	Description	Class A	Class B	Class C	Remarks
		Object Tunnel)				
	VAN Diagnosis	VAN specific diagnosis, status information of ASEs (e.g. config_status, VAN_device_statuses)				
	VAN Dev Config	Operating parameters of functional parts of a VAN component				
	VAN Access Management	provider access information, appropriate login information and mechanism to handle the login				
	VAN Update Service	firmware, version information, update rules, list of update objects				
	VAN Location	geographical (GPS, Galileo), logical (e.g. belongs to radio cell, local language ...)				
	VAN Security Config					
	VAN Switching	alternative path switching (e.g line redundancy, least cost routing)				
	VAN Peer-to-Peer	container for VAN internal communication				
	VAN Routing	routing between VAN subdomains				
	VAN Redundancy	low priority (optional) -> not to be realized in this step of the project (board decision)				
	VAN Time	time in PROFINET format, time synchronisation type, master/slave				

Major layer	Feature/Object	Description	Class A	Class B	Class C	Remarks
		function, synchronisation status				
	Security					
	TCP/IP Stack					
	SNMP	Only for monitoring purposes				
	HTTP	HTTP Protocol as carrier for SOAP				
	DNS					
	DHCP					
	ICMP v4/v6					
	ARP					
	IPv4/v6					
	TCP					
	UDP					
	VAN Device and Network Management					
	Time Sync	Realizes the time synchronisation between VAN Devices: <ul style="list-style-type: none"> - Within one LAN - Over InterLAN - or WAN 				
Application Layer	VAN Automation Application Process					
	VAN System Management Application Process	Controls all parts of VAN Communication Stack and Network Technology Layer				
	VAN Security Application Process					
	Parameter Config.	Is an kind of data bases for all				

Major layer	Feature/Object	Description	Class A	Class B	Class C	Remarks
		defined parameters.				
	VAN PnP Application Process	Is used for automatic configuration and identification of connected VAN-Devices.				
	VAN Proxy Application Process					
	Web Server					

Table 3-1: Description of profile conformance classes A, B and C

Column	Text	Meaning
Feature/Object	<text>	Name of the object within the architecture diagram
Description	<text>	Textual description of the object
Class A, B, C	m	Mandatory
	o	Optional
	r	Recommend
	f	Object is never present
Remarks	See<#>	Remarks are defined in the given sub clause, table or figure of this profile document
	-	No constraints other than given in the reference document (sub)clause, or not applicable
	<text>	The text defines the constraint directly, for longer text table footnotes or table notes may be used

Table 3-2: Description of each column previous table

If an ASE or VAN architecture block is optional in all three conformance classes, it needn't to be listed.

Differences between recommended and optional blocks

Optional blocks are an option which **can** be implemented in a VAN device of the conformance class if the device needs this functionality. A typical example for such an optional function is the time synchronisation because not all VAN devices need time synchronisation to fulfil their automation function. Hence this all blocks related to the time synchronisation are optional blocks.

In opposite to this the recommend blocks of a VAN device **shall** be implemented for devices of this class. Exceptions for the implantation can be done if the recommend block is not required for the application the devices are usually used for.

3.6.1.3 Object model

The subchapter Object model should give an overview about the selection of VAN Architecture elements.

Fig. 3-4 shows the VAN device architecture. This figure should be used to visualise the selections done in the above chapters. The conventions for the use are defined in the legend in Fig. 3-5.

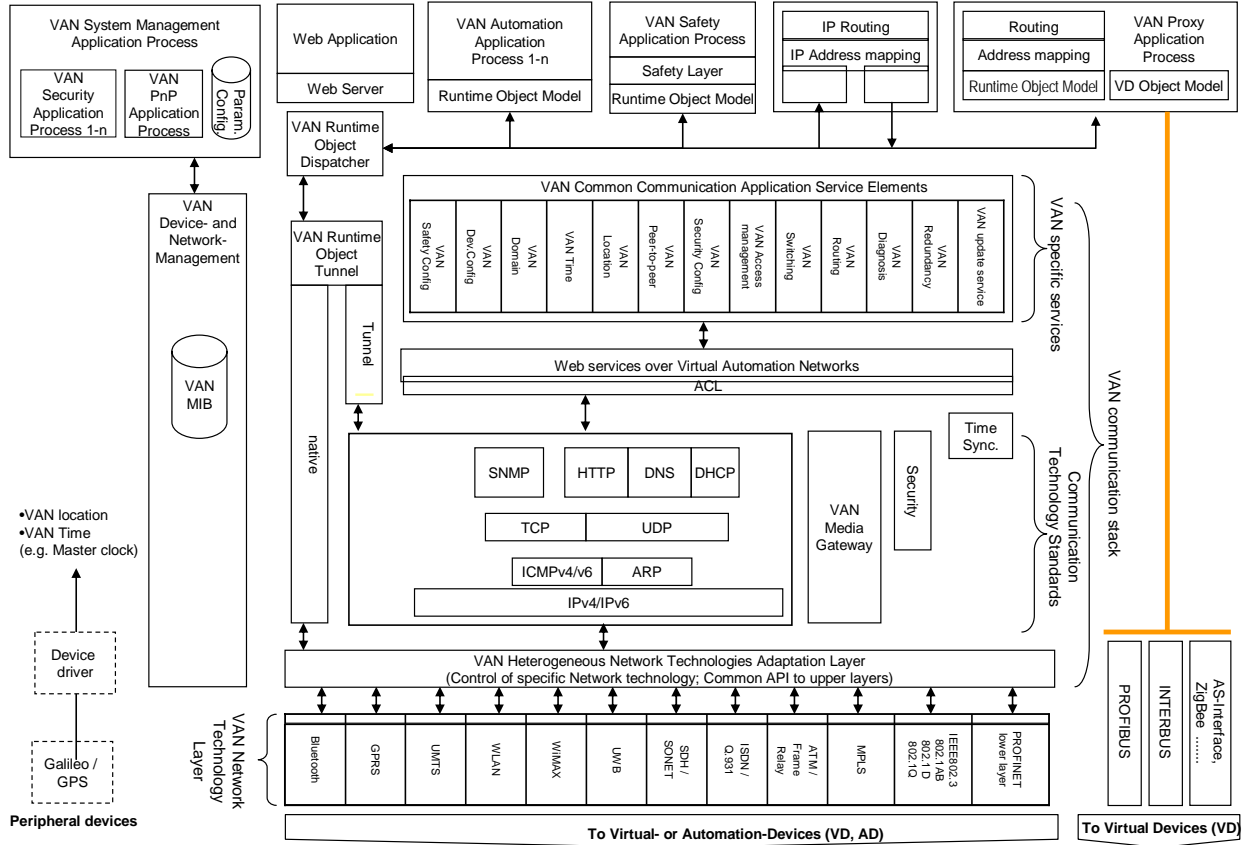


Fig. 3-4: Object Model

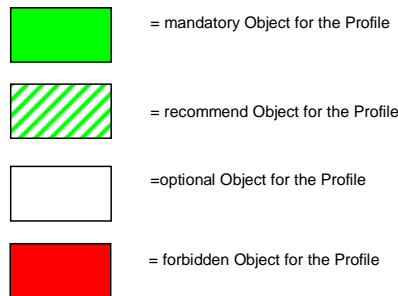


Fig. 3-5: Legend of symbols used in device architecture

4 VAN device profile definition

This chapter describes all needed device profiles. It includes VAN-AD, VAN-PD and VAN-AP. PnP Manager functionality is also defined. Engineering functionality cannot be described by profile and is described by WP8.

4.1 VAN Automation Device (VAN-AD)

4.1.1 General Overview

A VAN Automation Device is an entity which contains VAN services and protocol implementation and at least **one** automation function or automation application process in the VAN application context.

According to the definitions D02.2-1 chapter 1.2.3 a VAN-AD has to provided the main functions / properties selected in Table 4-1.

Device class / function set	Accessible over VAN	VAN-FS function set	VAN Domain name	Automation function	Proxy function	Security functions
VAN-PD	X	X	X	X	-	O

Table 4-1: VAN- AD device functions

Fig. 4-1 shows the two main functions of a VAN-AD consist of:

- the VAN Network capability and
- the Automation Function

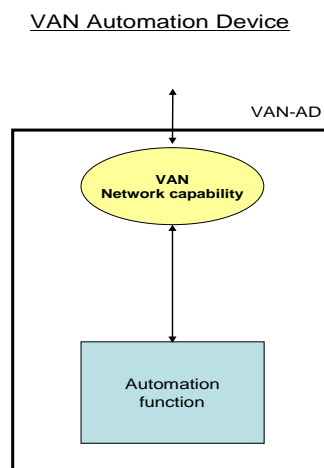


Fig. 4-1: VAN-AD

In this approach the connectivity to VAN network will be done by the VAN Network capability and the functionality will realized by the VAN Automation function.

4.1.1.1 Implementation of Automation Device

The following example shows a possible implementation of a PC based VAN-AD as extension of an existing solution.

In most cases in the state of the art, from a hardware point of view, control systems are PC based. In most cases the operating system is Windows, in other cases it is Linux. They are usually equipped with an Ethernet interface card. This allows implementing an Automation Device just by installing suitable hardware and software onto this architecture.

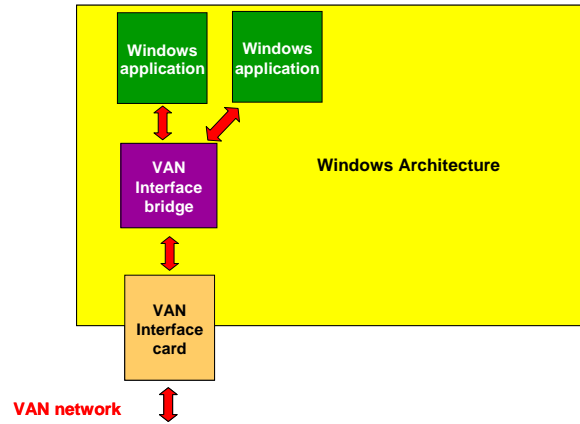


Fig. 4-2: Implementation of AD control system

A software module must be built, acting as a bridge among Windows applications and the VAN network, and managing VAN services and protocol implementation.

This way the control system includes VAN protocol, and at the same time contains its traditional automations functions, without the need of a separated proxy device. As a consequence it is compliant with the definition of VAN Automation Device.

4.1.2 Use case

Fig. 4-4 shows uses of VAN-AD based on the example 1 for the use of the VAN Runtime Object Tunnel, see chapter 9 of D02.2-2.

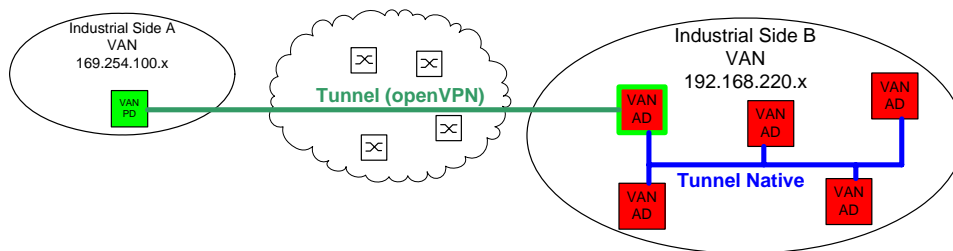


Fig. 4-3: Use case for VAN-AD

In this use case one VAN-AD communicates within the industrial side where it is located with other VAN-AD. The communication interface used for communication in this case is the native device driver interface of VAN Runtime Object Tunnel.

The second communication relation realized by the VAN-AD is the communication with a remote VAN Proxy Device via tunnelling interface of the VAN Runtime Object tunnel.

This is a typical situation for the use of VAN-AD for the time after the migration of VAN has been accomplished.

The devices of all tree device classes should be able to implement the describe uses case. For additional functionality like security mechanism, VAN Switching or VAN Time Synchronisation devices of the corresponding device class have to be used.

4.1.3 VAN- AD Object selection

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
VAN Network Layer	VAN Hetrogeneous Network Technologie Adaption Layer	See D02.2-1 chpater 2.4.10	m	m	m	The implementation of these Interface should consist of typical driver operations like open(), close(),read(), write() and iocontrol(). The implementation of the interface is local matter.
	IEEE 802.3	Access to Standard Ethernet Network.	m	m	m	The access over IEEE 802.3 to VAN communication enviornment
	PROFINET lower layer	Standard PROFINET Driver	m	m	m	Is the common access to the VAN communication environment
VAN Com- munication Stack	VAN Runtime Object Dispatcher	Dispatches automation objects conveyed via the runtime object tunnel.	m	m	m	Implementation is local matter and should be optimized to fit the vendor specific upper layer. Examples of using in combination with VAN runtime object tunnel see D02.2-2 Chapter 9.
	VAN Runtime Object Tunnel	The tunnel is an interface to the application layer.	m	m	m	The tunnel is divided into different levels one see API description D02.2. Example of using in combination with VAN runtime object tunnel see D02.2-2 Chapter 9.

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
	Web Service over Virtual Automation Networks	Provides the WS of the VAN device	m	m	m	The name based access via Web Service to the devices is one key concept of VAN. The WS should follow the W3C WS specifications.
	ACL	Access Control Layer assures the authorised access to objects in the VAN device.	m	m	m	The specification of the ACL functionality and use in VAN devices is topic of WP 6
	VAN Common Communication Application Service Elements	For all ASE see API specification in D02.2.-2	m	m	m	A minimal subset off ASE is need for each VAN device. ASE API description, [D02.2-2] page 40
	VAN Domain		m	m	m	Mandatory for all VAN Devices
	VAN Diagnosis		o	r	m	
	VAN Dev Config		m	m	m	Mandatory for all VAN Devices
	VAN Safety Config		o	o	o	
	VAN Update Sevice		o	r	m	
	VAN Security Config		o	r	m	
	VAN Access Management		o	r	m	
	VAN Switching		o	r	m	
	Security		o	r	m	The specification of the functionality and use in VAN devices is topic of WP 6
	TCP/IP Stack	Normal TCP/IP stack.	m	m	m	Defines TCP/IP Stack for the Web Service Access to the Device, see chapter Fehler! Verweisquelle konnte nicht gefunden werden..

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
	SNMP	Simple Network Management Protocol - is part of the Internet protocol suite and supports monitoring and control of network attached devices (router, server, switches) for any conditions in administration. In VAN it is used in local monitoring access	o	r	m	
	HTTP	HTTP Protocol as carrier for SOAP	m	m	m	
	DNS	Domain Name System - stores information associated with domain names in a distributed database on networks. Because VAN is based on domain names, it should be mandatory for all devices.	m	m	m	
	DHCP	Dynamic Host Configuration Protocol – client/server protocol based on UDP which takes care about assigning IP address based on MAC address.	m	m	m	
	ICMP v4/v6	ICMP is an integral part of IP and is implemented by each device with an IP stack. All devices with Web Services rely on this	m	m	m	
	ARP	ARP is a core protocol of the Internet protocol suite and is typically used for error responses in IP datagrams or for diagnostic and routing purposes. Inseparable part of IP	m	m	m	

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
		stack.				
	IPv4/v6	Internet Protocol stack. All devices with Web Services rely on this because are based on HTTP and HTTP is based on TCP/IP.	m	m	m	
	TCP	Transmission Control Protocol - one of the core protocols of the Internet protocol suite. The protocol guarantees reliable and in-order delivery of data from sender to receiver. All devices with Web Services rely on this.	m	m	m	
	UDP	User Datagram Protocol - UDP does not provide the reliability and ordering that TCP does. Compared to TCP, UDP is required for broadcast (send to all on local network) and multicast (send to all subscribers).	r	r	m	
	Time Sync	Realizes the time synchronisation between VAN Devices: <ul style="list-style-type: none"> - Within one LAN - Over InterLAN - or WAN 	o	o	o	The necessary attributes an the specification of the messages is done in D04.3
Application Layer	VAN Automation Application Process	Fulfils the automation function of the VAN-AD.	m	m	m	
	VAN System Management Application Process	Controls all parts of VAN Communication Stack and Network Technology Layer	m	m	m	
	VAN Security Application Process		o	r	m	The specification of the functionality and use in VAN devices is topic of WP 6

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
	Parameter Config.	Is an kind of data bases for all defined parameters.	m	m	m	
	VAN PnP Application Process	Is used for automatic configuration and identification of connected VAN-Devices.	o	m	m	
	VAN Proxy Application Process		f	f	f	A VAN-AD does not provide any Proxy Function for other Devices.
	Web Server		m	m	m	

4.1.4 Object model

The following figure shows the object model of a VAN-AD.

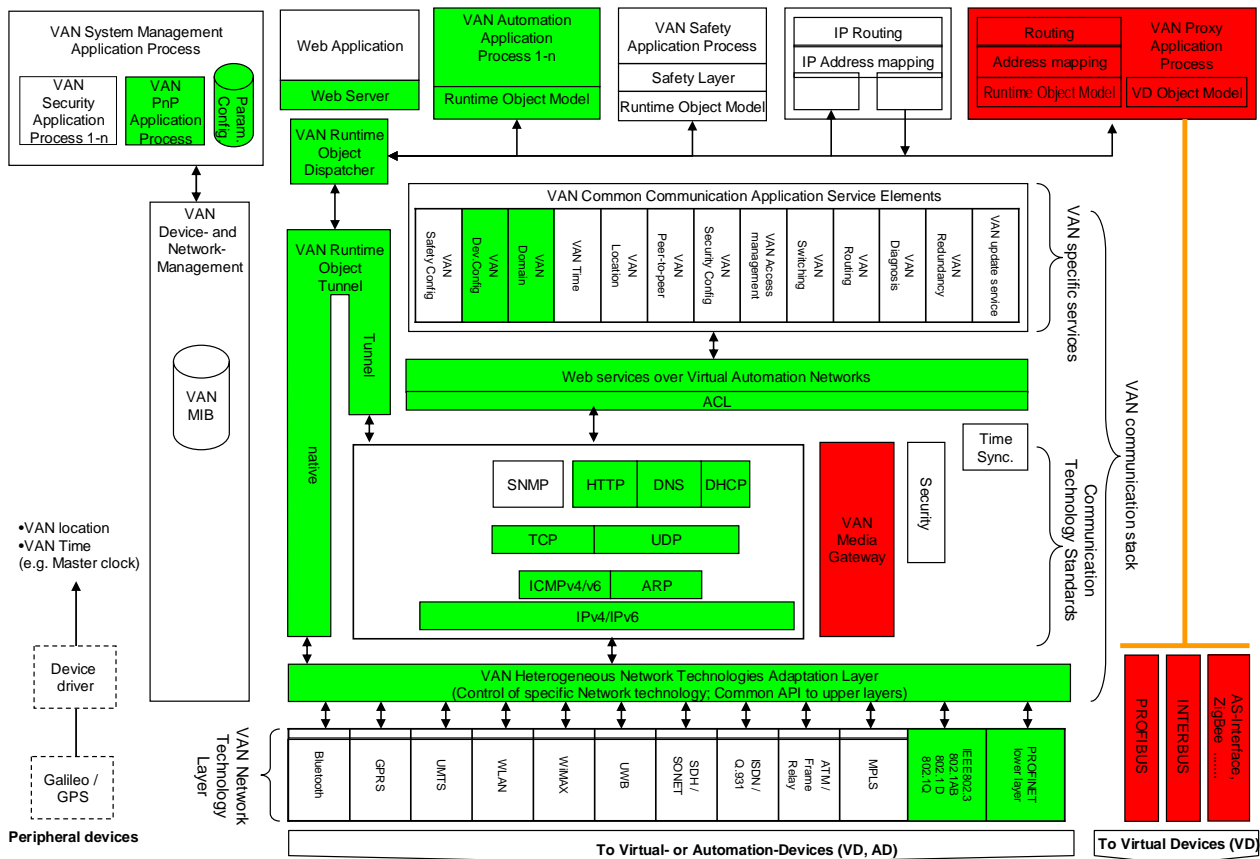


Fig. 4-4: VAN-AD class A object model

4.2 VAN Proxy

4.2.1 General overview

A VAN-PD is an entity that contains a VAN Function Set (VAN-FS) that is realising the VAN Network capability and a Proxy Application (Proxy Function).

According to the definitions of D02.2-1 chapter 1.2.3 VAN-PD provides the main functions shown in Table 4-2.

Device class / function set	Accessible over VAN	VAN-FS function set	VAN Domain name	Automation function	Proxy function	Security functions
VAN-PD	X	X	X	-	X	O

Table 4-2: Main functions of VAN-PD

With a VAN-PD an entire local fieldbus network can be connected to a VAN Network. Only those devices of the local fieldbus network that are “mirrored” into the VAN network are called VAN Virtual Devices. Each VAN-VD can be addressed separately within the VAN Network.

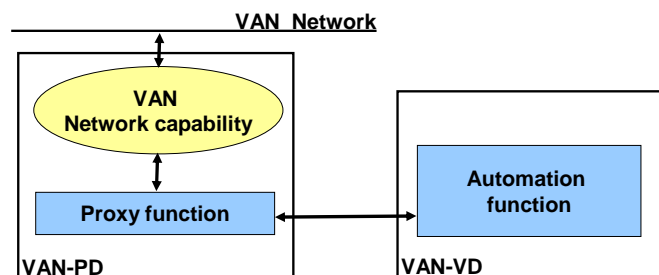


Fig. 4-5: VAN-PD and VAN-VD device

Fig. 4-5 shows the main function of an VAN-PD:

- the VAN Network capability,
- the Proxy Function,
- and the connection to an VAN-VD with its automation function.

VAN-PDs are used to visualize Automation device which are VAN aware but not VAN enabled (i.e. VAN-VD). These devices belong to networks which cannot handle the VAN communication and/or uses a different object model.

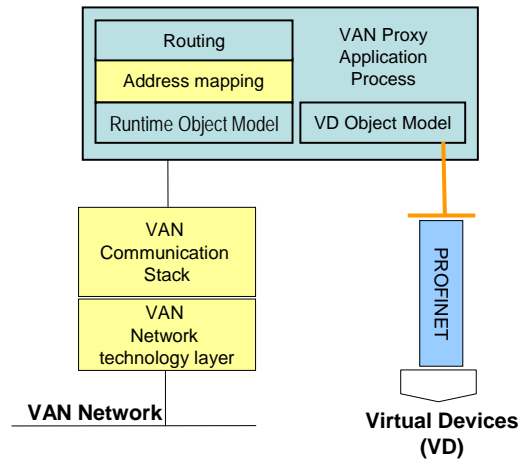


Fig. 4-6: VAN Proxy Application Process

To make devices aware of VAN the VAN Proxy Application Process of a VAN-PD is used, see Fig. 4-6. To realize this VAN Proxy Application Process translates the Runtime Object Model of the device into the VAN Runtime Object model and the VAN addressing to the local addressing of the device (e.g. Profibus addresses).

4.2.2 Use case

Fig. 4-7 shows uses of VAN-PD based on the example 3 for the use of the VAN Runtime Object Tunnel, see chapter 9 of D02.2-2.

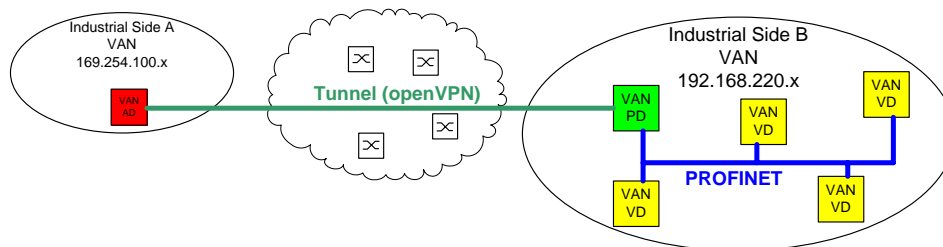


Fig. 4-7: VAN-PD use case

In this Use Case a VAN-PD communicates within the industrial side where it is located with VAN-VD over PROFINET.

The second communication channel realized by the VAN-PD is the communication with a remote VAN Automation Device via tunnelling interface of the VAN Runtime Object tunnel.

This use case shows a typical example for the migration time of not VAN Ethernet based field busses to VAN.

The devices of all tree device classes should be able to implement the describe uses case. For additional functionality like security mechanism, VAN Switching or VAN Time Synchronisation devices of the corresponding device class have to be used.

4.2.3 VAN- PD Object selection

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
VAN Network Layer	VAN Hetrogeneous Network Technologie Adaption Layer	See D02.2-1 chpater 2.4.10	M	m	m	The implementation of these Interface should consist of typical driver operations like open(), close(),read(), write() and iocontrol(). The implementation of the interface is local matter.
	IEEE 802.3	Access to Standard Ethernet Network.	M	m	m	The access over IEEE 802.3 to VAN communication enviornment
	PROFINET lower layer	Standard PROFINET Driver	M	m	m	Is the common access to the VAN communication environment
	Additional Bus system	e.g. Profibus	o	o	o	
VAN Com- munication Stack	VAN Runtime Object Dispatcher	Dispatches automation objects conveyed via the runtime object tunnel.	m	m	m	Implementation is local matter and should be optimized to fit the vendor specific upper layer. Examples of using in combination with VAN runtime object tunnel see D02.2-2 Chapter 9.
	VAN Runtime Object Tunnel	The tunnel is an interface to the application layer.	m	m	m	The tunnel is divided into different levels one see API description D02.2. Example of using in combination with VAN runtime object tunnel see D02.2-2 Chapter 9.
	Web Service over Virtual Automation Networks	Provides the WS of the VAN deviece	m	m	m	The name based access via Web Service to the devices is one key concept of VAN. The WS should follow the W3C WS

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
						specifications.
	ACL	Access Control Layer assures the authorised access to objects in the VAN device.	m	m	m	The specification of the ACL functionality and use in VAN devices is topic of WP 6
	VAN Common Communication Application Service Elements	For all ASE see API specification in D02.2.-2	m	m	m	A minimal subset off ASE is need for each VAN device. ASE API description, [D02.2-2] page 40
	VAN Domain		m	m	m	Mandatory for all VAN Devices
	VAN Diagnosis		o	r	m	
	VAN Dev Config		m	m	m	Mandatory for all VAN Devices
	VAN Safety Config		o	o	o	
	VAN Update Sevice		o	r	m	
	VAN Security Config		o	r	m	
	VAN Access Management		o	r	m	
	VAN Switching		o	r	m	
	Security		o	r	m	The specification of the functionality and use in VAN devices is topic of WP 6
	TCP/IP Stack	Normal TCP/IP stack.	m	m	m	Defines TCP/IP Stack for the Web Service Access to the Device, see chapter Fehler! Verweisquelle konnte nicht gefunden werden..
	SNMP	Simple Network Management Protocol - is part of the Internet protocol suite and supports monitoring and control of network	o	r	m	

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
		attached devices (router, server, switches) for any conditions in administration. In VAN it is used in local monitoring access				
	HTTP	HTTP Protocol as carrier for SOAP	m	m	m	
	DNS	Domain Name System - stores information associated with domain names in a distributed database on networks. Because VAN is based on domain names, it should be mandatory for all devices.	m	m	m	
	DHCP	Dynamic Host Configuration Protocol – client/server protocol based on UDP which takes care about assigning IP address based on MAC address.	m	m	m	
	ICMP v4/v6	ICMP is an integral part of IP and is implemented by each device with an IP stack. All devices with Web Services rely on this	m	m	m	
	ARP	ARP is a core protocol of the Internet protocol suite and is typically used for error responses in IP datagrams or for diagnostic and routing purposes. Inseparable part of IP stack.	m	m	m	
	IPv4/v6	Internet Protocol stack. All devices with Web Services rely on this because are based on HTTP and HTTP is based on	m	m	m	

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
		TCP/IP.				
	TCP	Transmission Control Protocol - one of the core protocols of the Internet protocol suite. The protocol guarantees reliable and in-order delivery of data from sender to receiver. All devices with Web Services rely on this.	m	m	m	
	UDP	User Datagram Protocol - UDP does not provide the reliability and ordering that TCP does. Compared to TCP, UDP is required for broadcast (send to all on local network) and multicast (send to all subscribers).	m	m	m	
	Time Sync	Realizes the time synchronisation between VAN Devices: <ul style="list-style-type: none"> - Within one LAN - Over InterLAN - or WAN 	o	o	o	The necessary attributes an the specification of the messages is done in D04.3
Application Layer	VAN Automation Application Process	Fulfills the automation function of the VAN-AD.	f	f	f	A VAN-PD does not provide any Automation Application functionality.
	VAN System Management Application Process	Controls all parts of VAN Communication Stack and Network Technology Layer	m	m	m	
	VAN Security Application Process		o	r	m	The specification of the functionality and use in VAN devices is topic of WP 6

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
	Parameter Config.	Is an kind of data bases for all defined parameters.	m	m	m	
	VAN PnP Application Process	Is used for automatic configuration and identification of connected VAN-Devices.	o	r	m	
	VAN Proxy Application Process		m	m	m	The Proxy Application Process is mandatory and the key functionality of all VAN-PDs.
	Web Server		m	m	m	

4.2.4 Object Model

The following figure shows the object

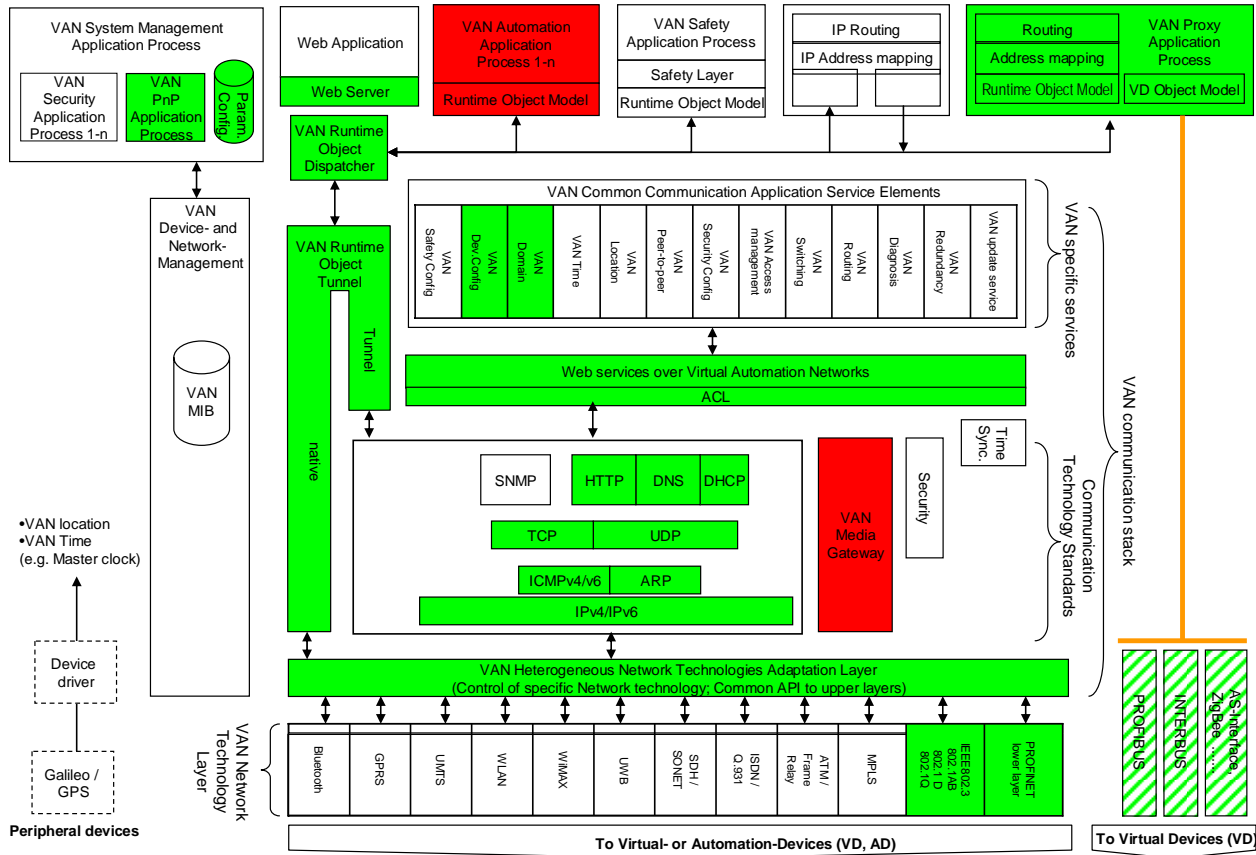


Fig. 4-8: VAN-PD class A object model

4.3 VAN Access Point

4.3.1 General overview

A VAN Access Point defines an entity containing VAN-FS and connects VAN network segments but it does not contain an automation function or an automation application process.

Note: It can work as a gateway or router to automation devices which are members in a VAN application context (VAN Domain).

Device class / function set	Accessible over VAN	VAN-FS function set	VAN Domain name	Automation function	Proxy function	Security functions
VAN-AP	X	X	X	-	-	X

Table 4-3: Main functions of VAN-AP

Any device participates on a VAN network by connecting through an integrated VAN Access Point (e.g. VAN Automation Device) or through a VAN Proxy Device. The VAN Access Point defines an entity containing VAN-FS which provides information and functions to other VAN-APs or engineering tools according to the following list:

VAN-AP Delivers:

- Capability of the Van infrastructure -> information / capability on the communication paths to the neighbour VAN device (capability of the runtime channels)

- Online (MIB) and Offline (description File) for Catalogue based “configurator”
- Conformance Classes (VAN device class and VAN device profile, Protocols, Realtime, ...)
- Diagnosis, Security, QoS, predictable response
- Online and Offline operation
- Time synchronization over VAN
- Provide Topology information
- Diagnostic interface for Engineering

Interacting between VAN devices with VAN-FS regarding:

- Communication runtime channel
- Diagnosis
- Topology scan
- Plug & Play of a VAN device, not an Automation Device
- QoS-Capability of the communication paths
- VAN-AP intermediate protocol
- Addressing concept
- Redundancy
- Time synchronizations
- Security functions
- Access Rights
- Nested VAN Networks (e.g. WLAN→ Public Network→ WLAN)

VAN-AP Receives:

- parameter and setup information from VAN engineering (VAN-EC) to guaranty for the VAN Connection
- Has a media independent access interface to engineering.

4.3.2 Use case

VAN-AP can work as a gateway or router to automation devices which are members in a VAN Application Context (VAN Domain). From the picture you can see example of gateway functionality - VA1, VA2. Routing functionality is illustrated by VA3 and VA4. Based on application needs conformance class should be selected.

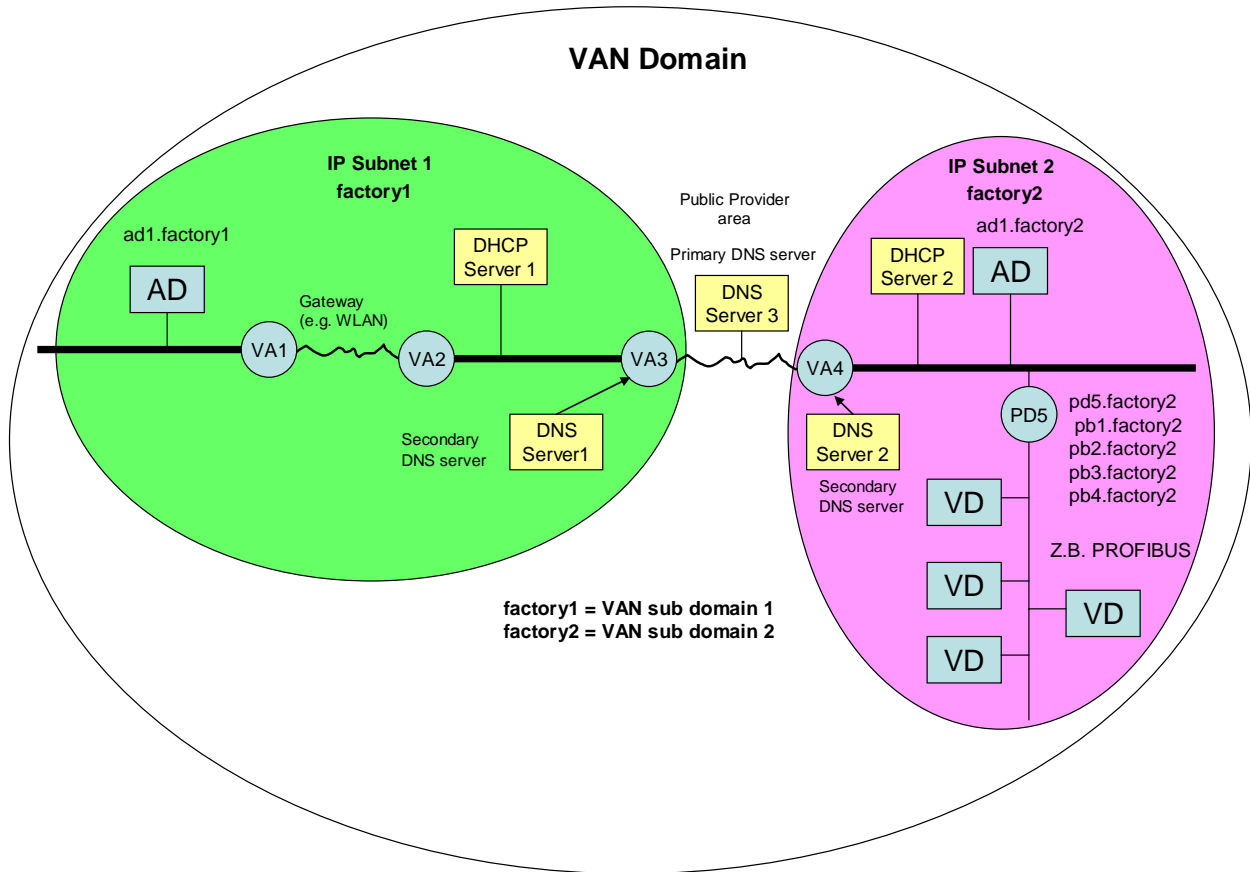


Fig. 4-9: VAN-AP as gateway (VA1, VA2) and router (VA3, VA4)

4.3.3 VAN-AP objects selection

Major layer	Feature/Object	Description	Class A	Class B	Class C	Remarks
VAN Network Layer	VAN Heterogeneous Network Technology Adaptation Layer	See D02.2-1 chapter 2.4.10	m	m	m	The implementation of these Interface should consist of typical driver operations like open(), close(),read(), write() and ioctl(). The implementation of the interface is local matter.
	IEEE 802.3	Access to standard Ethernet network.	m	m	m	The access over IEEE 802.3 to VAN communication environment
	PROFINET lower layer	Standard PROFINET Driver	m	m	m	It is the common access to the VAN

Major layer	Feature/Object	Description	Class A	Class B	Class C	Remarks
						communication environment.
VAN Communication Stack	VAN Runtime Object Dispatcher	Dispatches automation objects conveyed via the runtime object tunnel.	m	m	m	Implementation is local matter and should be optimized to fit the vendor specific upper layer. Examples of using in combination with VAN runtime object tunnel see D02.2-2 Chapter 9.
	VAN Runtime Object Tunnel	The tunnel is an interface to the application layer.	m	m	m	The tunnel is divided into different levels one see API description D02.2. Example of using in combination with VAN runtime object tunnel see D02.2-2 Chapter 9.
	Web Service over Virtual Automation Networks	Provides the WS of the VAN device	m	m	m	The name based access via Web Service to the devices is one key concept of VAN.
	ACL	Access Control Layer assures the authorised access to objects in the VAN device.	m	m	m	The functionality is topic of WP 6.
	VAN Common Communication Application Service Elements	For all ASE see API specification in D02.2.-2	m	m	m	A minimal subset of ASE is need for each VAN device.
	VAN Domain		m	m	m	
	VAN Diagnosis		o	r	m	
	VAN Dev Config		m	m	m	
	VAN Access Management		m	m	m	
	VAN Update Service		o	r	m	
	VAN Security Config		o	r	m	
VAN Switching		m	m	m		
VAN Peer-to-Peer		m	m	m		

Major layer	Feature/Object	Description	Class A	Class B	Class C	Remarks
	VAN Routing		m	m	m	
	VAN Redundancy		o	r	m	
	VAN Time		m	m	m	
	Security		o	r	m	The functionality is topic of WP 6
	TCP/IP Stack		m	m	m	Defines TCP/IP Stack for the Web Service Access to the Device.
	SNMP		o	r	m	
	HTTP	HTTP Protocol as carrier for SOAP	m	m	m	
	DNS		m	m	m	
	DHCP		m	m	m	
	ICMP v4/v6		m	m	m	
	ARP		m	m	m	
	IPv4/v6		m	m	m	
	TCP		m	m	m	
	UDP		m	m	m	
	VAN Device and Network Management		m	m	m	
Time Sync	Realizes the time synchronisation between VAN Devices: <ul style="list-style-type: none"> - Within one LAN - Over InterLAN - or WAN 	m	m	m	The necessary attributes an the specification of the messages is done in D04.3	
Application Layer	VAN Automation Application Process		f	f	f	
	VAN System Management Application Process	Controls all parts of VAN Communication Stack and Network Technology Layer	m	m	m	

Major layer	Feature/Object	Description	Class A	Class B	Class C	Remarks
	VAN Security Application Process		o	r	m	The functionality is topic of WP 6
	Parameter Config.	Is an kind of data bases for all defined parameters.	m	m	m	Discussion in Tech PCC which kind of storing mechanism will be provided by VAN devices.
	VAN PnP Application Process	Is used for automatic configuration and identification of connected VAN-Devices.	m	m	m	
	VAN Proxy Application Process		f	f	f	A VAN-AD does not provide any Proxy Function for other Devices.
	Web Server		m	m	m	

4.3.4 Object Model

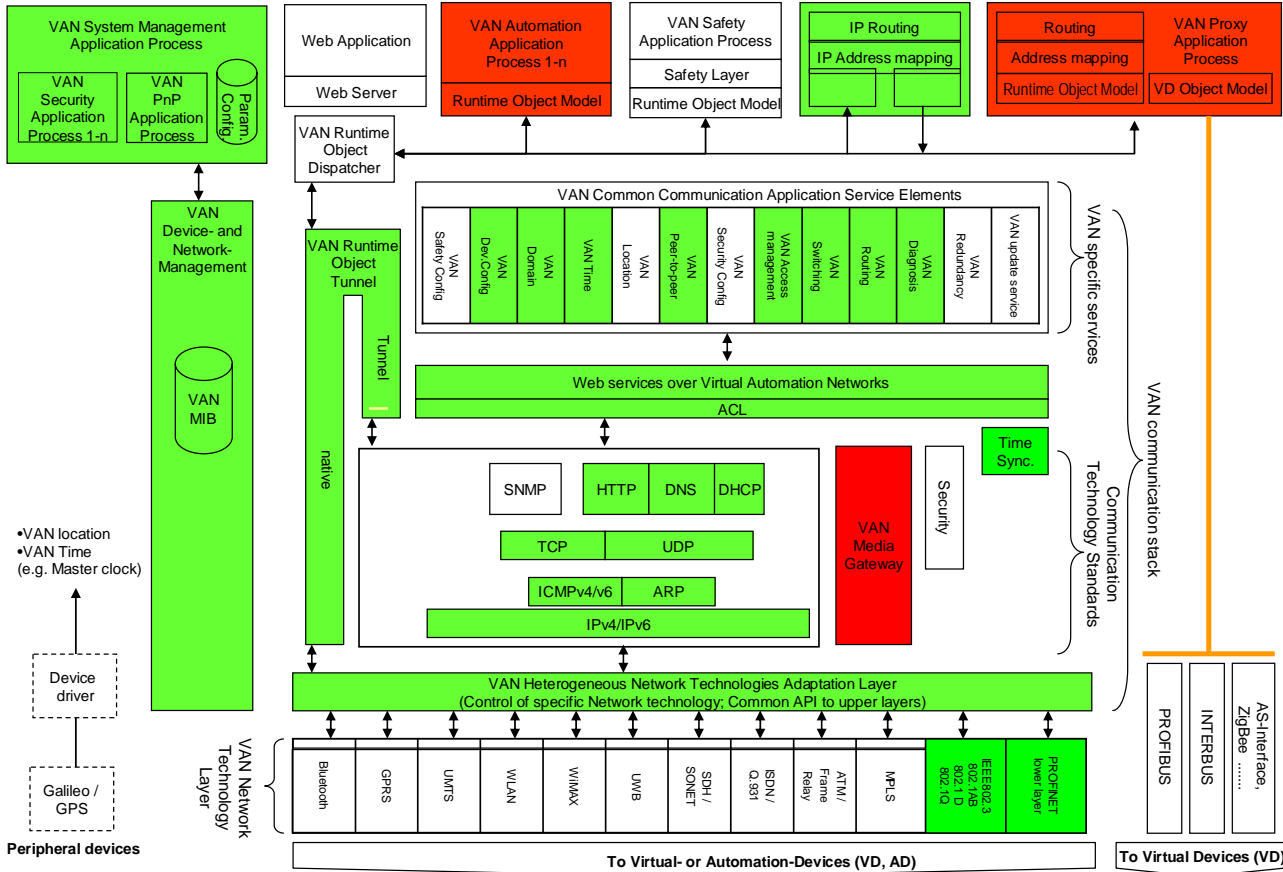


Fig. 4-10: VAN-AP Class A object model

4.4 Plug&Play Manager

Basis of the realization is the technology of the Web Services.

A Web Service is defined as a software system designed to support interoperable machine-to-machine interaction over a network. The services use for communication as a general rule SOAP-formatted XML coding (being in a manner of speaking the 'envelope').

The W3C specified WSDL as the description method for Web Services. In addition a Web Service should be interoperable according the WS-I (Web Service Interoperability Organization).

VAN will use a Service-Orientated Architecture (SOA). Since VAN introduces the VAN PnP functionality which allows to locate the distributed service providers, UDDI is not needed and therefore not used.

So the WSDL should not be used to discover metadata about VAN Web Services by a service registrar but only for the standardised description of the Web Service within VAN.

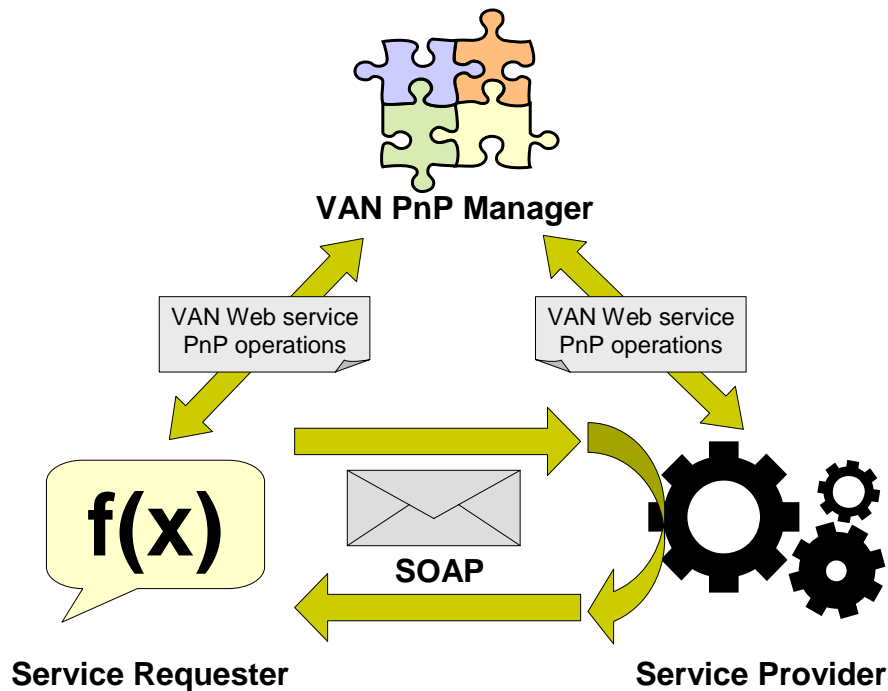


Fig. 4-11: VAN Web Service without UDDI

The WSDL describes the interface of the service, types, operations, access control and details to deployment in machine readable form. All information of the service access is concentrated in a standardised form.

4.4.1 General overview

4.4.1.1 "Plug and Play" process

In the deliverable 2.2-1 mechanisms were defined for "plug & play".

The PnP Application Process is important during the entire lifecycle of a VAN domain, especially for maintenance and engineering. Information about all containing devices must be collected and stored. To realize the plug and play process in a VAN domain different PnP roles are defined. A PnP-broker has to manage all available PnP-requests in its VAN domain. To get and set the information from and to the PnP-Requests and vice versa a PnP Broker uses services from several VAN objects like the Peer-to-Peer or the Device Configuration.

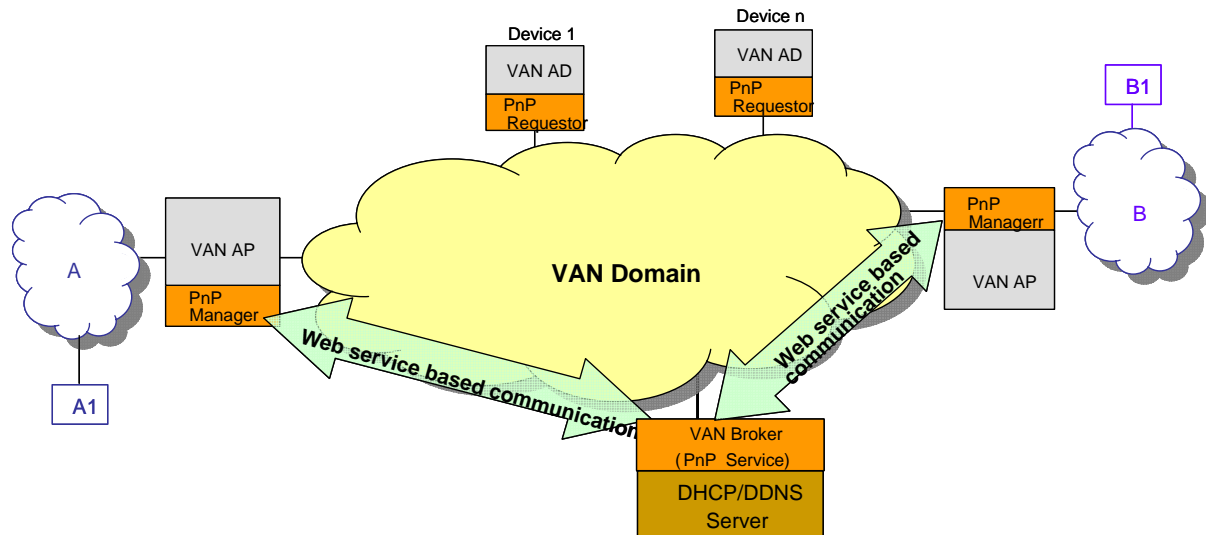


Fig. 4-12: PnP scenario in a VAN domain

In the deliverable 2.2-1 three different PnP roles are defined:

- PnP Manager (→Broker),
- PnP Agent (→Requestor) and
- PnP Negotiator (→Broker).

The PnP Manager respectively Broker is the central point in a VAN domain that keeps the knowledge about the VAN devices of this VAN domain and their status.

A PnP Requestor respectively Agent delivers its relevant device information to the PnP Broker via Web Services.

A Negotiator is a special PnP Broker which has the task to connect all PnP Manager which are located in different VAN sub domains. So a query e.g. for a VAN Automation device outside the own sub domain is forwarded from the PnP Manager to the known PnP Negotiator. For this the PnP Negotiator exchanges the relevant information with all known PnP Manager which are located in the other sub domains.

Since the PnP Negotiator is a PnP Broker with a special task, it is in the further elaboration considered as PnP Manager and not separately described. So only the two PnP roles PnP Requestor (Agent ?) and PnP Manager are further described.

For the description of the PnP Application Process also the PnP function set is important. This function set contains

- the “Network Addressing”,
- the “Knowledge Discovery” (Device Discovery),
- the “Service Presentation” and
- the “Network Control”

and is implemented in the PnP Requestors and PnP Brokers. The basic tasks of the function set are already described in the deliverable 2.2-1. This chapter describes how the different functions work in the PnP roles and which services from the different ASEs are used to realise the PnP Application Process.

If required the PnP Application Process can provide additional server addresses for instance an event drain for logging and alarm purposes.

4.4.1.2 Plug&Play Manager functions

The VAN PnP Manager parts are defined (D02.2-1 chapter 3.3) as active role to control and schedule the commissioning and maintenance of the VAN network and the automation task in a nearly automatic way.

A *VAN PnP function* shall solve certain tasks in a VAN network, e.g. that all participants of the VAN network are reachable. The *VAN PnP roles* have to implement special parts of the *VAN PnP functions* to solve this task.

Necessary functions Plug&Play Managers:

- Network Addressing Function - assigning of network addresses and names
- Knowledge Discovery Function - identification of devices
- Service Presentation Function - realizing of device services
- Network Control Function - controlling and scheduling

The elements which contain VAN PnP functions are highlighted. Other elements of the architecture certainly are used by the PnP elements such as the *VAN Runtime Object Tunnel*, some of the VAN Common Communication ASEs and of course the Web Services.

4.4.2 Use case

The VAN PnP Manager has an active role and executes the VAN PnP Functions. In each IP address space a VAN PnP Manager is required in order to capture all requested devices. The VAN PnP Application Process is part of the VAN PnP Manager.

The VAN PnP Manager also acts as a gateway to the fieldbus systems to make the PnP specific fieldbus traffic available. The current realization depends strongly on the corresponding fieldbus system.

The VAN Engineering uses the VAN PnP Manager as server which provides the current network situation (available devices, topology information, connection attributes, and so on).

A VAN PnP Manager has the general states un-configured and configured. An un-configured VAN PnP Manager has no information about the planned devices. As consequence an un-configured VAN PnP Manager scans a VAN domain without restrictions.

A VAN PnP Manager will be configured by the VAN engineering. A configured VAN PnP Manager has information about the planned devices. The domain scan can be restricted in this case.

4.4.2.1 Example: PnP role-playing between a VAN Engineering station, VAN Automation Devices, and Fieldbus Automation Devices in a defined VAN domain

In the following figure a VAN domain is represented which consists of two LANs. The LANs are connected via the internet using a public provider. Furthermore the VAN domain comprises a number of VAN automation devices, fieldbus automation devices, and a VAN engineering station. All VAN automation devices and fieldbus automation devices belong to the same automation application which is designed by the VAN engineering station.

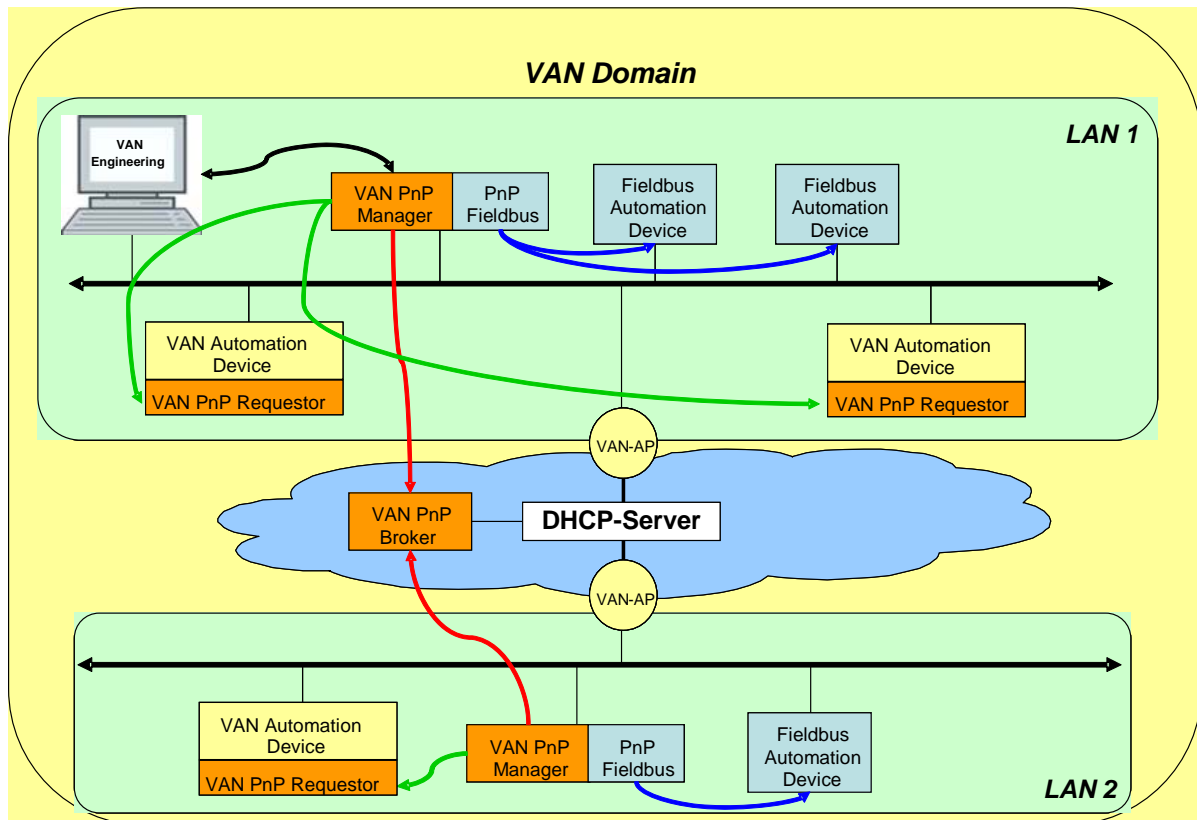


Fig. 4-13: VAN PnP roles with VAN Engineering

Relationship between the VAN Engineering and a VAN PnP Manager

In a first step the VAN PnP Manager shall capture all reachable devices and store them in an information base. The Engineering station anytime can read out the information base with the captured devices. Alternatively the VAN PnP Manager indicates new captured devices. In a second step the Engineering station loads the configuration on the VAN PnP Manager. The configured VAN PnP Manager shall observe all configured devices which belong to a special automation application during runtime. In case of a device failure the device can be exchanged without the engineering station and all connections will be automatically re-established.

Relationship between a VAN PnP Manager and a VAN PnP Requestor

The VAN PnP Manager is a central point in a defined address space and unambiguous able to be identified by a special name. The VAN PnP Manager shall manage a current list of all VAN PnP Agents in its responsible network. All VAN PnP Requests shall provide PnP relevant objects and services for the VAN PnP Manager.

Relationship between a VAN PnP Manager and an Automation Device

Automation Devices are conventional Ethernet based fieldbus devices. They are not equipped with the VAN PnP Agent functionality. Depending on the respective fieldbus system partial special PnP functionality is available. This special PnP functionality is re-used by the PnP fieldbus as part of the VAN PnP Manager. So the VAN PnP Manager acts as a gateway. The gateway utilizes the fieldbus PnP functions for the VAN Engineering. In this case the gateway works like the Fieldbus Engineering station to capture the corresponding Automation Devices.

The possibilities with relation to PnP are always limited by the special fieldbus system.

Relationship between a VAN PnP Manager and a VAN PnP Broker

The VAN PnP Broker connects all VAN PnP Managers in a VAN domain. Similar to the Domain Name Service a query for an Automation Device is forwarded from the VAN PnP Manager to the known VAN PnP Negotiator. The VAN PnP asks all known VAN PnP Managers which are located in the bordering networks.

4.4.2.2 Classification of the VAN PnP roles

The table below shows the integration of the VAN PnP roles into the VAN device classes and the classification of the functionality to the VAN function sets.

device classes	VAN PnP Manager	VAN PnP Requestor (Agent ?)	VAN Broker (Web Service !)
VAN-AP (VAN Access Point)	x	x	-
VAN-AD (VAN Automation Device)	-	x	-
VAN-VD (VAN Virtual Device)	-	-	-
VAN-PD (VAN Proxy Device)	-	x	-
VAN-SVD (VAN Server Device)	x (local network)	x	x (public network)
VAN-SID (VAN Security Infrastructure Device)	-	x	-
Function sets			
VAN-EC (VAN Engineering Client)	x (client side)	-	-
VAN-M (VAN Management)	x	-	-
VAN-FS (VAN Function Set)	x	x	x

Table 4-4: Classification of the VAN PnP roles to the device classes and function sets

4.4.3 VAN-PnP Manager objects selection

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
VAN Network Layer	VAN Hetrogeneous Network Technologie Adaption Layer	See D02.2-1 chapter 2.4.10	m	m	m	The implementation of these Interface should consist of typical driver operations like open(), close(),read(), write() and ioctl().

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
						The implementation of the interface is local matter.
	IEEE 802.3	Access to Standard Ethernet Network.	m	m	m	The access over IEEE 802.3 to VAN communication environment
	PROFINET lower layer	Standard PROFINET Driver	o	o	o	Is the common access to the VAN communication environment
VAN Communication Stack	VAN Runtime Object Dispatcher	Dispatches automation objects conveyed via the runtime object tunnel.	o	r	m	Implementation is local matter and should be optimized to fit the vendor specific upper layer. Examples of using in combination with VAN runtime object tunnel see D02.2-2 Chapter 9.
	VAN Runtime Object Tunnel	The tunnel is an interface to the application layer.	o	r	m	The tunnel is divided into different levels one see API description D02.2. Example of using in combination with VAN runtime object tunnel see D02.2-2 Chapter 9.
	Web Service over Virtual Automation Networks	Provides the WS of the VAN deviece	o	r	m	The name based access via Web Service to the devioces is one key concept of VAN. The WS should follow the W3C WS specifications.
	ACL	Access Control Layer assures the authorised access to objects in the VAN device.	m	m	m	The functionality is topic of WP 6
	VAN Common Communication Application Service Elements	For all ASE see API specification in D02.2.-2	m	m	m	A minimal subset off ASE is need for each VAN device.

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
	VAN Domain		m	m	m	
	VAN Dev Config		m	m	m	
	VAN Access Management		m	m	m	
	VAN Diagnosis		o	r	r	
	VAN Safety Config		o	o	o	
	VAN Update Sevice		o	r	m	
	VAN Security Config		r	r	m	
	VAN Switching		o	r	m	
	VAN Peer-to-Peer	container for VAN internal communication	o	r	m	
	VAN Routing	routing between VAN subdomains	m	m	m	
	VAN Location	geographical (GPS, Galileo), logical (e.g. belongs to radio cell, local	o	r	r	
	Security		r	r	m	The functionality is topic of WP 6
	TCP/IP Stack	Normal TCP/IP stack.	m	m	m	Defines TCP/IP Stack for the Web Service Access to the Device.
	HTTP	HTTP Protocol as carrier for SOAP	m	m	m	
	DNS	Domain Name System - stores information associated with domain names in a distributed database on networks. Because VAN is based on domain names, it should be mandatory for all devices.	m	m	m	
	DHCP	Dynamic Host Configuration Protocol – client/server protocol	o	r	m	

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
		based on UDP which takes care about assigning IP address based on MAC address.				
	SNMP	Simple Network Management Protocol - is part of the Internet protocol suite and supports monitoring and control of network attached devices (router, server, switches) for any conditions in administration. In VAN it is used in local monitoring access	o	r	m	
	ICMP v4/v6	ICMP is an integral part of IP and is implemented by each device with an IP stack. All devices with Web Services rely on this	m	m	m	
	ARP	ARP is a core protocol of the Internet protocol suite and is typically used for error responses in IP datagrams or for diagnostic and routing purposes. Inseparable part of IP stack.	m	m	m	
	IPv4/v6	Internet Protocol stack. All devices with Web Services rely on this because are based on HTTP and HTTP is based on TCP/IP.	m	m	m	
	TCP	Transmission Control Protocol - one of the core protocols of the Internet protocol suite. The protocol guarantees reliable and in-order delivery of data from sender to receiver. All devices with Web	m	m	m	

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
		Services rely on this.				
	UDP	User Datagram Protocol - UDP does not provide the reliability and ordering that TCP does. Compared to TCP, UDP is required for broadcast (send to all on local network) and multicast (send to all subscribers).	m	m	m	
	Time Sync	Realizes the time synchronisation between VAN Devices: <ul style="list-style-type: none"> - Within one LAN - Over InterLAN - or WAN 	o	o	o	The necessary attributes an the specification of the messages is done in D04.3
Application Layer	VAN Automation Application Process	Fulfils the automation function of the VAN-AD.	f	f	f	
	VAN System Management Application Process	Controls all parts of VAN Communication Stack and Network Technology Layer	m	m	m	
	VAN PnP Application Process	Is used for automatic configuration and identification of connected VAN-Devices.	m	m	m	
	VAN Security Application Process		o	r	m	The functionality is topic of WP 6 => is the ACL this Application Process?
	Parameter Config.	Is an kind of data bases for all defined parameters.	m	m	m	Discussion in Tech PCC which kind of storing mechanism will be provided by VAN devices. => The storing mechanism of the defined
	VAN Proxy Application Process		f	f	f	A VAN-AD does not provide any Proxy Function for other

Major layer	Feature/ Object	Description / Access to	Class A	Class B	Class C	Remarks
						Devices.
	Web Server		m	m	m	

4.4.4 Object Model

Subchapter includes a figure where the mandatory and recommend objects of the profile are highlighted

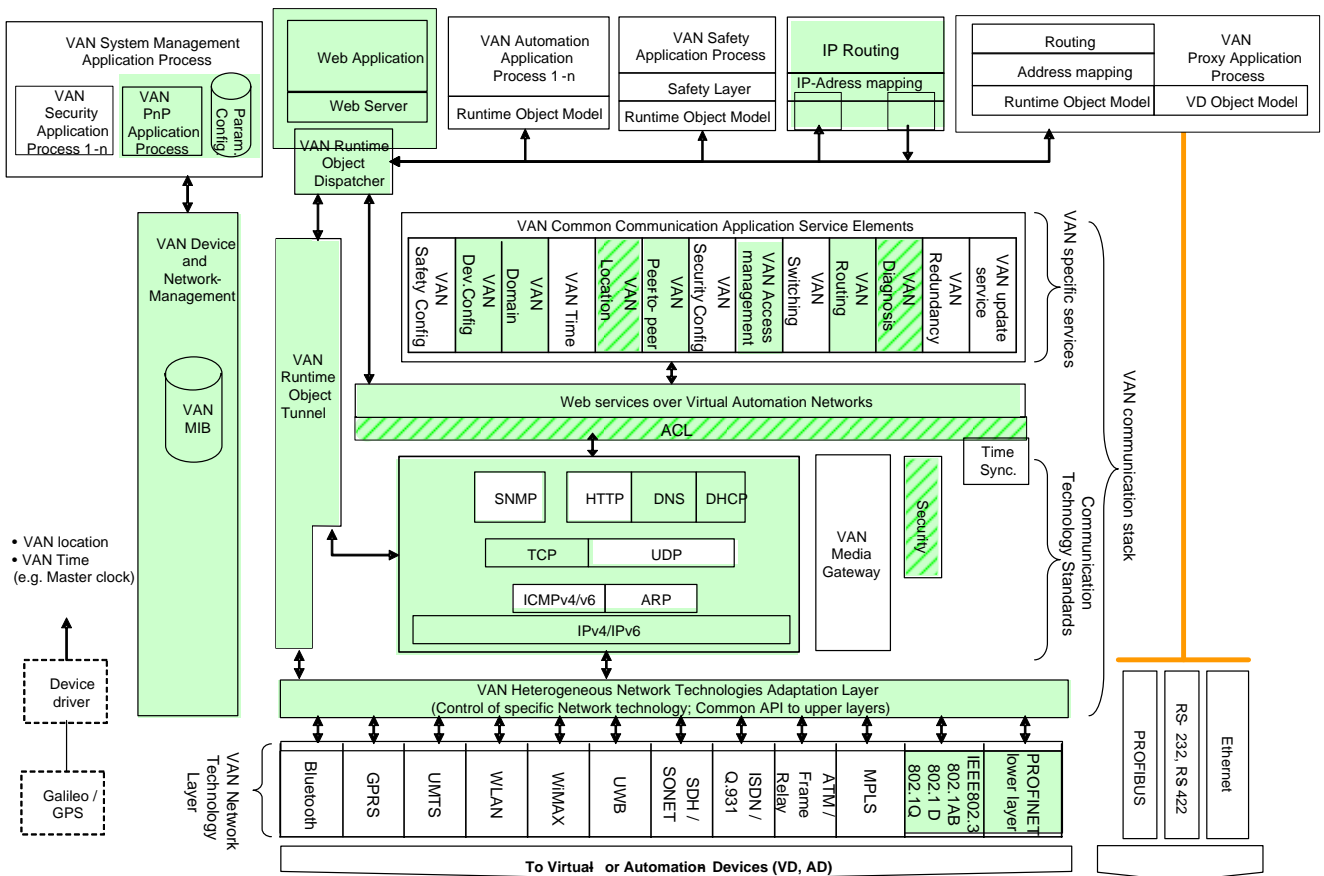


Fig. 4-14: VAN-PnP Manager object model

5 VAN Application Profiles Definitions

5.1 Manufacturing Industry Profiles

This section deals with application profiles definition in the flexible Manufacturing Industry.

A categorization about the different manufacturing devices is considered and two main groups are identified.

A list of manufacturing devices that are commonly used nowadays is then reported for the two identified categories.

Some manufacturing devices are then chosen from these lists and the VAN application device profiles for those are then defined.

The base for defining the VAN application profiles is the “VAN Device Profile Definition” that is reported in the chapter 4 of this document. We will always refer to it as a base in this section.

5.1.1.1 Manufacturing Device categories

In this section a list of devices that are commonly used in the manufacturing environment is reported.

A classification of “manufacturing field devices” and “manufacturing supervisor devices” is adopted. Using this type of categorization it is possible to show the different levels of IT devices deployment in modern manufacturing plants.

Devices that belong to the supervisor network level have been selected within an enlarged set of industrial and commercial IT devices because they generally perform functions that are not easy to find if we look only at the industrial device category.

The field devices list contains well-known traditional industrial devices.

Nowadays with the increasing diffusion of IT features into industrial devices and components the proposed classification could be considered more as logical or functional rather than environmental. Moreover the hierarchical architecture vision of a Manufacturing production system with a multi-agent cooperating structure is going to spread device intelligence over the field more than it has been since now.

5.1.1.1.1 Manufacturing field devices

Here is a list of well-known industrial automation devices that are used in the Manufacturing Industry. All have a field-bus interface available, but not all of them have an embedded Ethernet interface.

- ◆ Computer Numerical Control (CNC)
- ◆ Programmable Logic Controller (PLC)
- ◆ Robot controller
- ◆ Servo drive
- ◆ Positional transducer
- ◆ Digital I/O signal module
- ◆ Analogical signal module
- ◆ Operator panel

- ◆ Operator keyboard

CNCs, PLCs and Robot controller units are most suitable to be defined as VAN devices.

5.1.1.1.2 Manufacturing supervisor devices

Here we a list of device used in Manufacturing production plants that are connected to the Plant Ethernet Network.

- ◆ PC supervisor main unit
- ◆ PC operator terminal unit
- ◆ Web network camera
- ◆ Tool RFID station
- ◆ Ethernet 10/100 Mbit/s switch
- ◆ Point-to-point Wi-Fi wireless link

These devices have all a standard Ethernet interface. The Tool RFID station has also a field-bus interface. PC based devices are the most appropriate for VAN device implementations.

5.1.2 VAN Application profiles and Manufacturing devices

In the chapter 4 of this deliverable the VAN devices were defined to belong to these main categories:

1. VAN-AD VAN Automation Device
2. VAN-PD VAN Proxy Device
3. VAN-AP VAN Access Point Device

The VAN Manufacturing devices we are going to define and to describe here can be identified only in the first (1) device category.

This is due to the following reasons:

VAN-PD device profile can't be used because all the devices we have here implement their own automation function, while VAN-PD doesn't have any Automation function (see the VAN Proxy section in chapter 4).

VAN-AP device profile can't be considered in our application devices, because VAN-AP devices are units that connect sub-networks in a VAN Domain without performing any Automation function.

In the following sections we will describe four possible Manufacturing devices, that can be derived from the proposed VAN-AD device type, as VAN application devices.

For each automation device a VAN application profile is defined.

The four selected devices, which belong to the two category lists we introduced in the previous section, are reported in the following table along with their VAN device type and their selected Conformance Class (see chapter 4 about this concept).

VAN Application device	VAN Device	Conformance Class
VAN CNC profile	VAN-AD	C
VAN RFID station	VAN-AD	B
VAN Robot controller	VAN-AD	B
VAN Supervisor unit	VAN-AD	C

Table 5-1: VAN Manufacturing Application profiles

5.1.3 VAN CNC profile

5.1.3.1 Device description

CNC (Computer Numerical Control) devices are well known automation devices that are used in the Manufacturing Industry to control different type of machines, mechanical components, transportation system and in some cases an entire small production plant. Here we want to consider a very common use of this device as a Machine Tool control system.

The CNC device controls different sets of digital and analogical signals that come from a mix of low level field sensors and devices and that are used for various machine control functions, for instance: pushbuttons and lamps in the pushbutton panel, end of travel switches, spindle gears signals, brake and drive enable signals, tool change signals, and so on. The field signals are managed in different ways in the CNC; some of them are connected to the PLC system that is usually embedded in the CNC, some of them are handled by the Motion Control itself. Some of them may be high frequency signals (for instance signals used for command of an axis).

From a hardware point of view, most of current Computer Numerical Controls are PC based. In most cases the operating system is Windows, in other cases it is Linux. They are usually equipped with an Ethernet interface card for connecting the device to the Manufacturing plant network or to the Factory Network.

The CNC is then a device that could be connected to two different VAN Network segments in VAN Domain: a low level Network segment with hard real-time communication between field devices and a higher level Network segment with a soft real-time or no real-time communication.

5.1.3.2 Interfaces description

All the various field-bus systems can be used in a CNC device.

Here are some of them:

- PROFIBUS DP
- Multi Point Interface (MPI)
- PROFINet
- AS-Interface
- Ethernet-IP

These are well known standardized field-buses that are used in the Manufacturing Industry.

Non standard and very application specific interfaces are also present in the CNC device, but these are out of the VAN project scope.

5.1.3.3 Protocols

TCP/IP and all protocols provided by the PC operating system are usually available as a standard.

Any other protocol that is implemented on the field-bus side can be easily added.

5.1.3.4 Communication requirements

Communication requirements are reported here for the two main identified Network segments on which a CNC device can be connected.

Low level IO signals interfaces

Real-time communication is required.

Cycle time in the order of 10 ms, and response time in the order of 100 microseconds are required for ordinary digital binary signals.

Cycle time in the order of 1 ms, and response time in the order of a few microseconds are required for high frequency digital binary signals.

Higher level network interfaces

Several information can be exchanged to a higher level network host. Typical transfer data are:

- Status information for monitoring
- Commands from a higher level plant supervisor
- Parts-programs

In all these cases communication requirements are present, but not very strict: no real-time communication is required. Regarding status information, they are usually transmitted with a fixed but quite low frequency (Hz or less). The constraint for the transmission of part-programs is that the speed for transmission should be higher than the speed of the machine for execution. This way no stop-and-go of the machine is caused. This data transfer is then time dependent, but no real-time control is applied on it: a guaranteed Network availability with a certain minimum communication rate and a well defined memory buffer length are enough to let the machine working without stopping.

5.1.3.5 Object model

In this section the VAN Application profile of a CNC device is derived from the VAN-AD profile.

Class C VAN Automation Device object model is the most suitable to be applied here.

Some additional functions that are considered as "optional" in the VAN-AD model have to be added: VAN Safety, VAN Time.

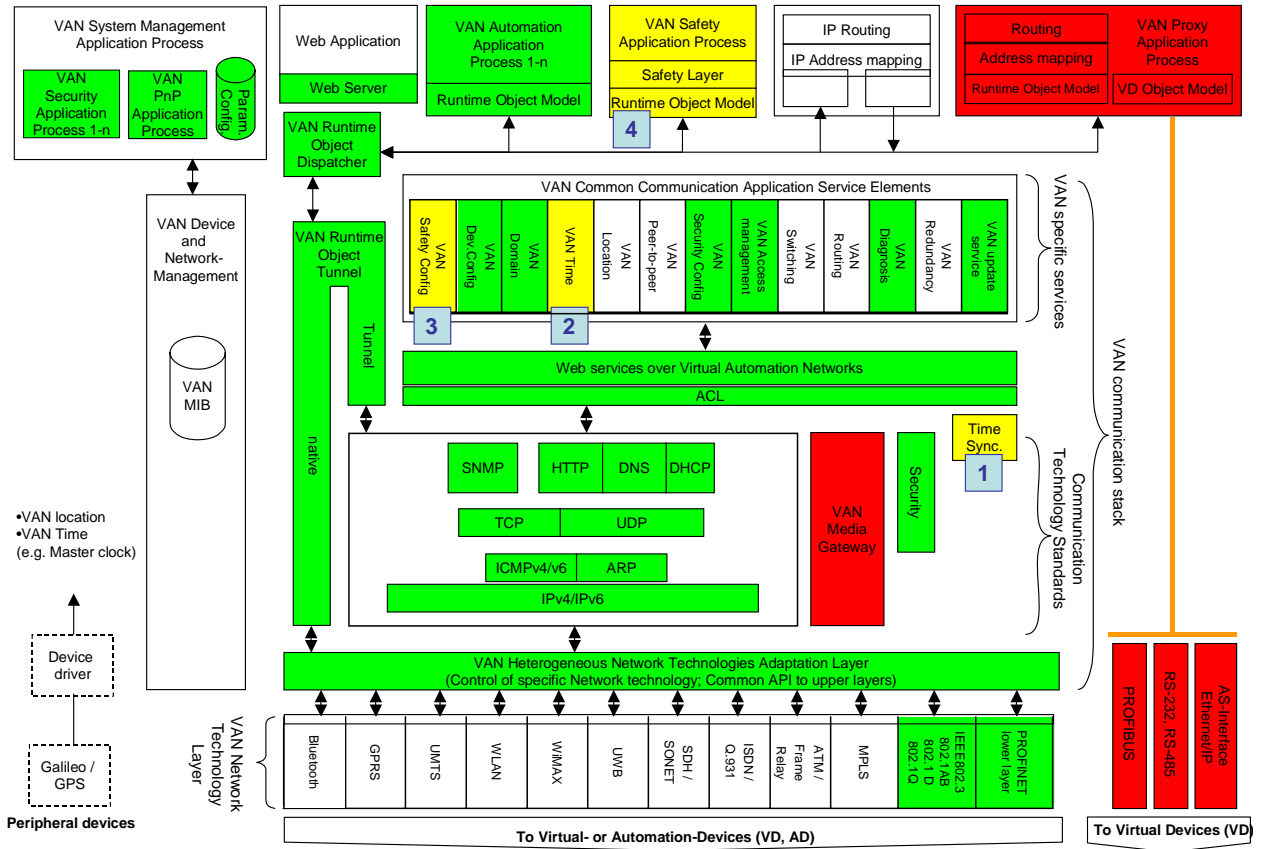


Fig. 5-1: VAN CNC object model

The following table lists “optional” VAN-AD Class C blocks that have been added to this device configuration.

N	Object	Description
1	Time Sync	Performs Time Synchronisation between VAN devices.
2	VAN Time	Time in PROFINet format, time synchronization type, synchronisations status.
3	VAN Safety Configuration	Configuration of the Safety layer/function.
4	VAN Safety Application Process	The specific Safety application.

Table 5-2: VAN CNC additional blocks

5.1.4 VAN RFID station profile

5.1.4.1 Device description

Some devices are used in the manufacturing production plants for the identification of resources. Typical mechanical resources needed for setting up a machine task are tools and parts or an aggregation of parts mounted on a fixture and carried by a pallet.

In an automated manufacturing plant these resources can be charged and temporary stored in different and not predefined positions, so their physical location varies continuously during the production time. Furthermore these resources can be moved between two or more different areas of the same plant and could be moved between different production plants. Automated controlled transportation systems perform moving of resources. Mechanical tools are normally preset outside of the plant, the factory or even the company when there is a third-party tools provider.

Normally these resources have some parameter values that describe and identify them. These values generally change during the normal production flow.

In this profile we consider a mechanical tools identification system where data change after any utilization of the tool, but it is possible to think a similar situation for a fixture identification system in which the status of parts (or a collection of parts) changes at any production step.

The collection of data that identifies a single tool resource is normally stored on a magnetic chip or tag that is attached to the tool. Nowadays identification systems in the manufacturing and logistics are evolving with many different RFID technologies, so different store devices may be used and a wide range of applications may arise in the near future. Substantially the various technologies that are available today differ on the distance that can be reached between the store element attached to the resource and the read/write interface unit. The applied store media technology and the amount of data to store in single unit are important factors to consider when a particular RFID system has to be chosen for a manufacturing application, but this is an aspect related to the environmental conditions and the way of utilization.

The device profile we want to submit here is the read/write RFID station that is used to store or read parameter data on the tool tag.

In a manufacturing plant one or more read/write RFID stations are normally present, for presetting or loading resources into the plant, but it is possible to foresee a higher utilization of identification stations in the near future.

With the nowadays technology evolution in this field it could be possible to localise elements in a defined space.

Using an identification and localisation technology the resource management could be completely designed and automated in a different way.

5.1.4.2 Interfaces description

These industrial devices have always a standard RS-232 or RS-485 interface. AS-Interface, Profibus-DP, Ethernet/IP and a more common Ethernet TCP/IP are also available as an option, one for each device model.

The RFID station can be considered as a VAN Automation Device (VAN-AD).

In some manufacturing production systems the RFID device may be used with a direct connection to a CNC device as a low level device.

The presetting station is always implemented on a dedicated PC and the engineering tools data are usually transferred through the enterprise network.

In mechanical tool identification system the data that are stored on the tag consist of tool parameters: geometrical, technological, etc..

A single data unit may be formalized easily as an XML entity: (see [VAN06b]).

5.1.4.3 Protocols

If using the Ethernet/IP or the standard Ethernet TCP/IP versions, the TCP/IP software protocol is directly implemented inside the device. DHCP and SNMP are also available.

Software utilities running on Windows are available for the device configuration and test.

Proprietary application protocols are used for the read/write operation of data.

A CRC of data is transmitted along with the data message for data integrity check.

The parameters configuration is possible using an EDS file.

5.1.4.4 Communication requirements

The application doesn't have particular requirements of real-time. Only a validation for a command message completion without errors is required.

5.1.4.5 Object model

In this section the VAN Application profile of the RFID station device is built using the VAN-AD class B object model.

The red and green blocks have the same meanings "mandatory" and "forbidden" as they have in the general VAN-AD object model, while yellow blocks are the optional and recommended blocks we can consider to be part of the particular Application profile we are defining here.

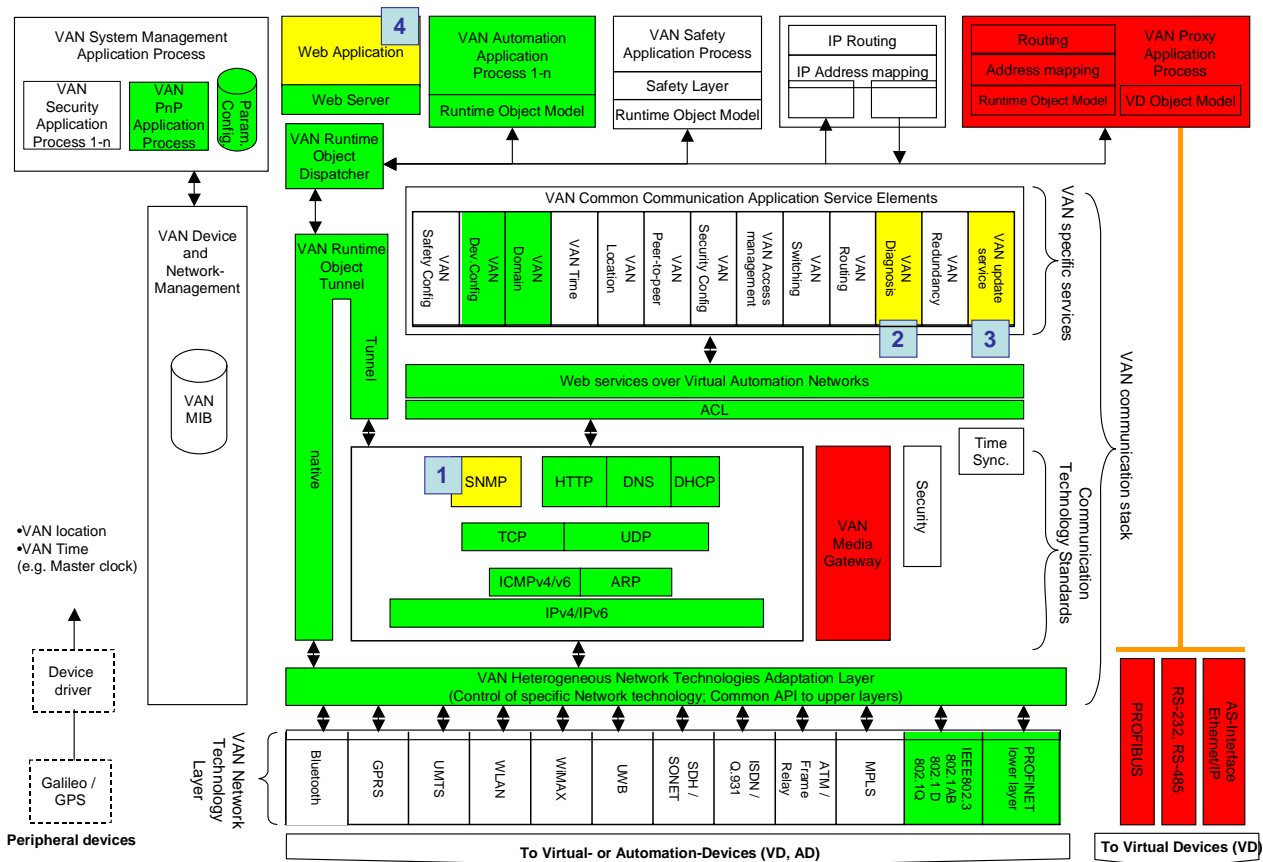


Fig. 5-2: VAN CNC object model

We don't have here to further detail the red and green VAN-AD class B blocks: see chapter 4 for descriptions and remarks about these blocks.

The optional and recommended VAN-AD class B blocks that are used in this Application profile are listed in Table 5-3.

The recommended SNMP block is considered here because our RFID Ethernet TCP/IP device has the SNMP protocol embedded in the firmware and it starts by default.

N	Object	Description
1	SNMP	The SNMP Protocol embedded in the device.
2	ASE VAN Diagnosis	VAN Diagnosis specific service.
3	ASE VAN update service	VAN update service specific function.
4	Web Application	Web Application integrated in the VAN-AD.

Table 5-3: VAN RFID station additional blocks

5.1.5 VAN Robot controller profile

5.1.5.1 Device description

This profile specifies the device controller of a robot that is used in many manufacturing applications for the manipulation of the mechanical parts that are produced by a flexible manufacturing system plant.

Parts are moved between predefined positions by the robot when needed, typically prior to start a machine task and when the task is completed, so parts go into the machine and came back after the working cycle is ended.

For setting up new activities or missions the controller must be programmed. Commands have to be sent to the controller for starting a mission; status, alarms and errors conditions have to be received from the controller for the robot unit management.

This type of device is part of the planned VAN Manufacturing demonstrator.

5.1.5.2 Interfaces description

A local attached hand held terminal is used for the robot setup phase.

An RS-232 interface is always available.

The Ethernet interface is integrated in the robot controller and this is nowadays the standard way to communicate with this device.

A PROFIBUS interface is an option.

5.1.5.3 Protocols

A TCP/IP protocol stack on the Ethernet interface is available. UDP protocols is used for messages.

File transfer is available using a proprietary software library that is based on the TCP/IP protocol stack.

The application protocol used for message exchange: mission start, mission end, status, alarms, etc., is proprietary and require a software driver.

5.1.5.4 Communication requirements

The set-up phase and the regular operation phase have to be considered differently.

The RS-232/RS-422 interface and the local attached hand held terminal have to be used for the set-up phase.

The Ethernet interface is used for normal operations.

Part program has to be transferred to the robot and be resident on there before starting a mission.

A part program may be used for a group of missions because it accepts a set of input parameters.

A mission is started by sending a command to the robot and specifying the associated parameters.

Robot configuration, status description and alarms are stored in the local controller memory that may be accessed using a proprietary protocol.

5.1.5.5 Object model

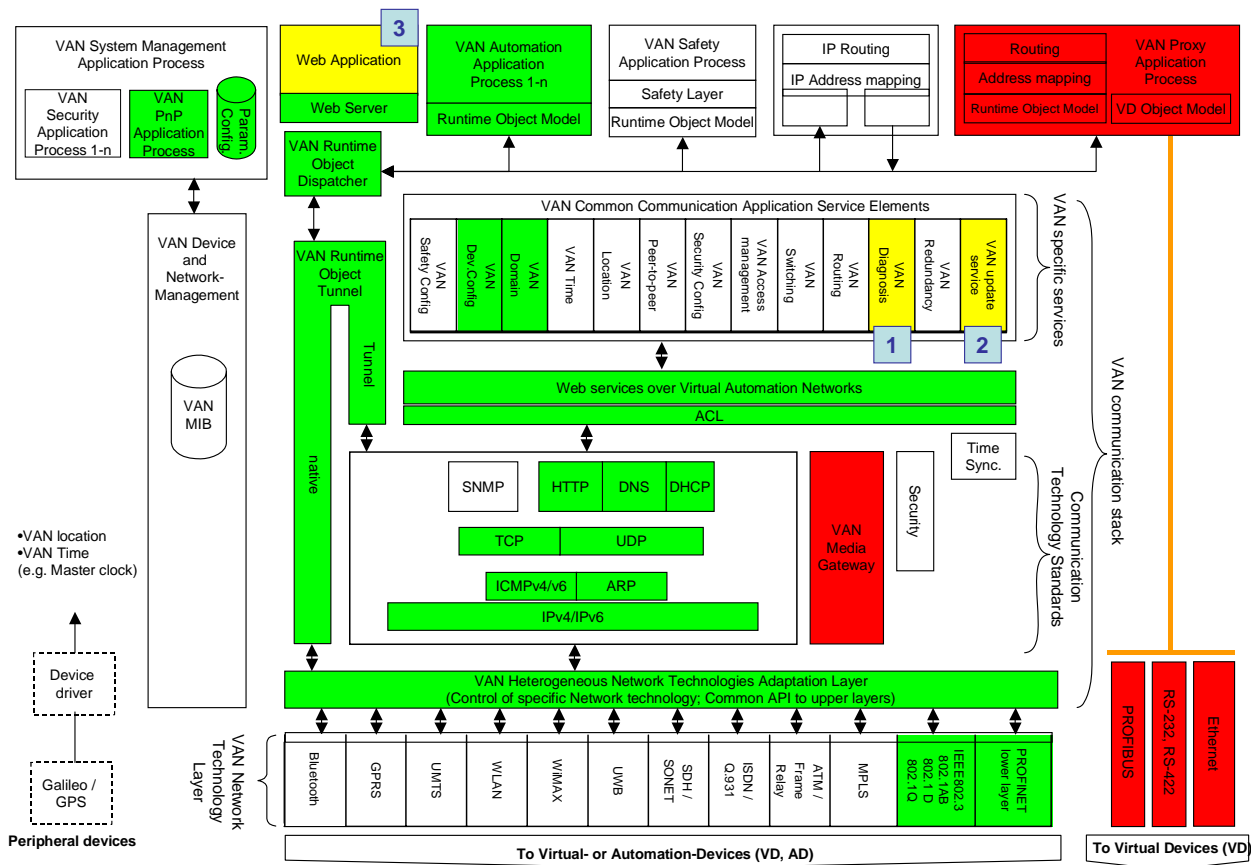


Fig. 5-3: VAN Robot controller object model

N	Object	Description
1	ASE VAN Diagnosis	VAN Diagnosis specific service.
2	ASE VAN update service	VAN update service specific function.
3	Web application	Web application integrated in the VAN-AD.

Table 5-4: VAN Robot controller additional blocks

5.1.6 VAN Supervisor unit profile

5.1.6.1 Device description

This profile defines a manufacturing supervisor control unit as a VAN device.

Generally speaking the mission of a plant supervisor is to logically manage all the different devices that belong to a manufacturing production plant, to carry out a predefined production plan, based on the availability of various sets of resources. Controlling of a single part production time, parts and tools moving optimisation, redistribution of resources, production load balancing, the whole production plant scheduling, production quality, etc., are only a few of the different aspects a plant supervisor can cover.

The supervisor physically interconnects real automation device like CNCs, PLCs, digital IO units, specific sensors, etc. that reside on one or more Ethernet networks.

The supervisor unit has an operating system like Windows, Unix or Linux and nowadays runs Java application software modules, to be more OS independent and more suitable for the Internet Web applications.

The supervisor PC unit has usually two or more Ethernet cards and can have different types of proprietary or standard interfaces.

Supervisor functions can be centralized in a hierarchical system architecture approach or can be distributed on many units over the network.

The same hardware and software modules are used for the different component units in a distributed architecture scheme, so a unique profile description can be adopted for this VAN device.

5.1.6.2 Interfaces description

The hardware of this device is a common industrial PC so all types of standard interfaces are used: RS-232, USB, Ethernet and so on.

This device is normally connected to the plant automation devices thru an Ethernet network.

A PSTN Modem connection have been traditionally used for the remote maintenance function, but nowadays a more convenient routed connection to the Internet via VPN networks is used instead.

Point-to-point Wireless links are also used, but normally they are implemented on external Wireless devices that are attached to the Ethernet network (Wireless Access Points) so no integrated WLAN interfaces are considered for this Application profile.

5.1.6.3 Protocols

TCP/IP on Ethernet is implemented with all type of services. Both TCP and UDP are used for command message exchange. FTP is used for file transfer, etc. .

PPP protocol is used for remote maintenance on dedicated Modem connections.

A Web Server is always running and the HTTP protocol is used.

Many proprietary application protocols are used for managing different types of devices that are directly attached to this unit, but these additional automation devices can't be formalized in the proposed set of VAN devices.

5.1.6.4 Communication requirements

No real-time communication constraints are required for the applications that are running on this device.

Single operation response time is usually in the order of units of seconds.

5.1.6.5 Object model

Fig. 5-4 depicts the VAN Supervisor unit as a VAN-AD Class C object.

Very few additional blocks are added to the selected source model to define this Application profile.

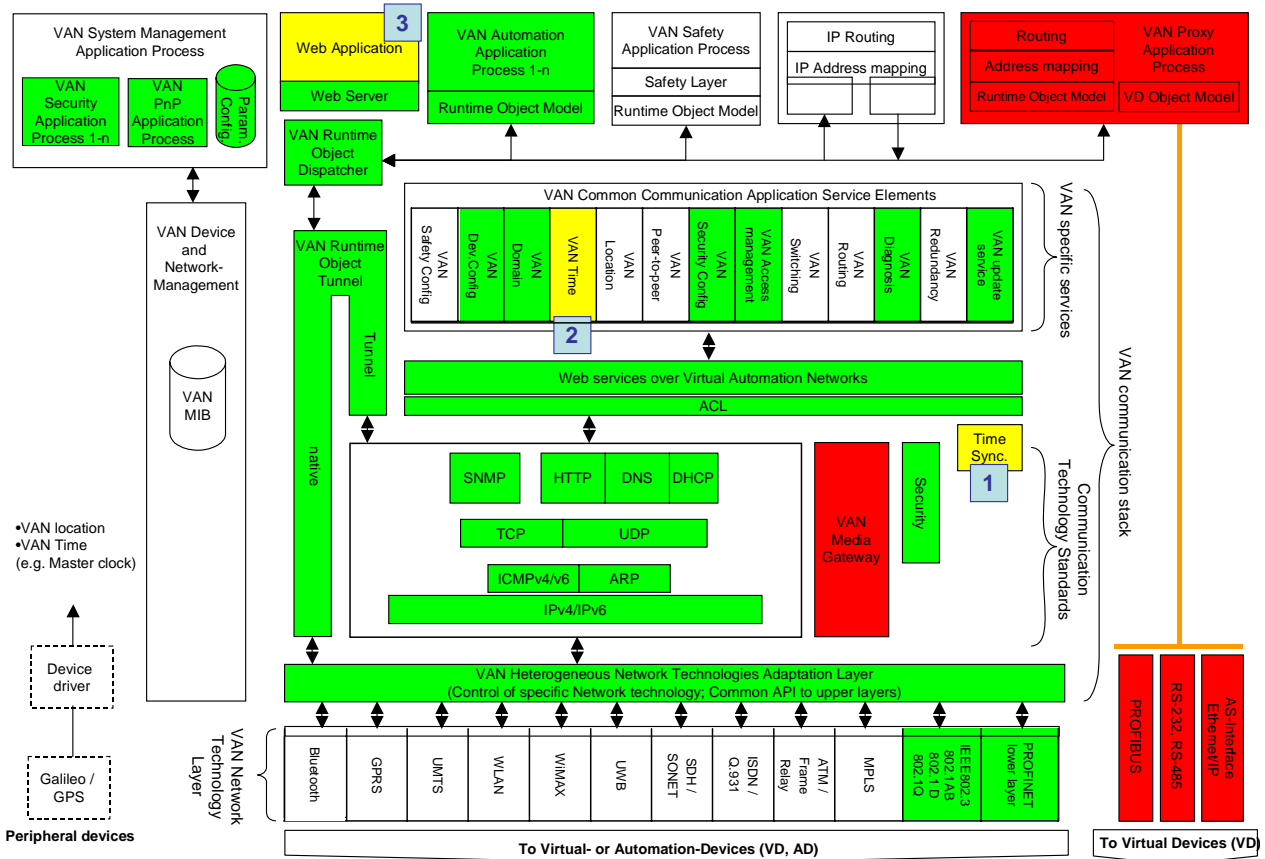


Fig. 5-4: VAN Supervisor unit object model

The following table lists optional blocks that are added in this case.

N	Object	Description
1	Time Sync	Performs Time Synchronisation between VAN devices.
2	VAN Time	Time in PROFInet format, time synchronization type, synchronisations status.
3	Web Application	The Web Applications running on this object.

Table 5-5: VAN Supervisor unit additional blocks

5.2 Process Industry Profiles

5.2.1 Approach to the definition of the application profiles for the Process Industry

The main content of the chapter 5.2 is to use the VAN device profile for the definition of application profiles in the demonstrator of the process Industry. The definition of application profile will be done in three steps:

- Description of the functional structures of the demonstrator use cases (Fig. 5.3 and Fig. 5.4) from the point of view of control engineering

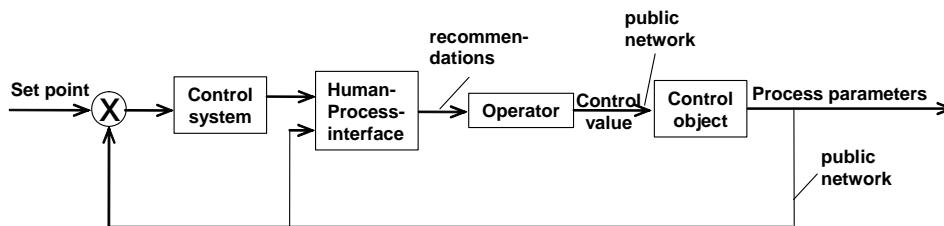


Fig. 5-1: Functional structure of the “open loop” systems for leakage control of gas storage and parameter switching in the power plant

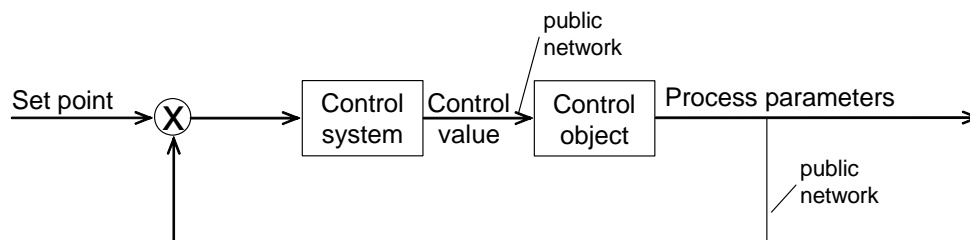


Fig. 5-2: Functional structure of the “closed loop” systems for foaming prevention

- Definition of technical structure of the central control of decentralized technological systems (Fig. 5.6)

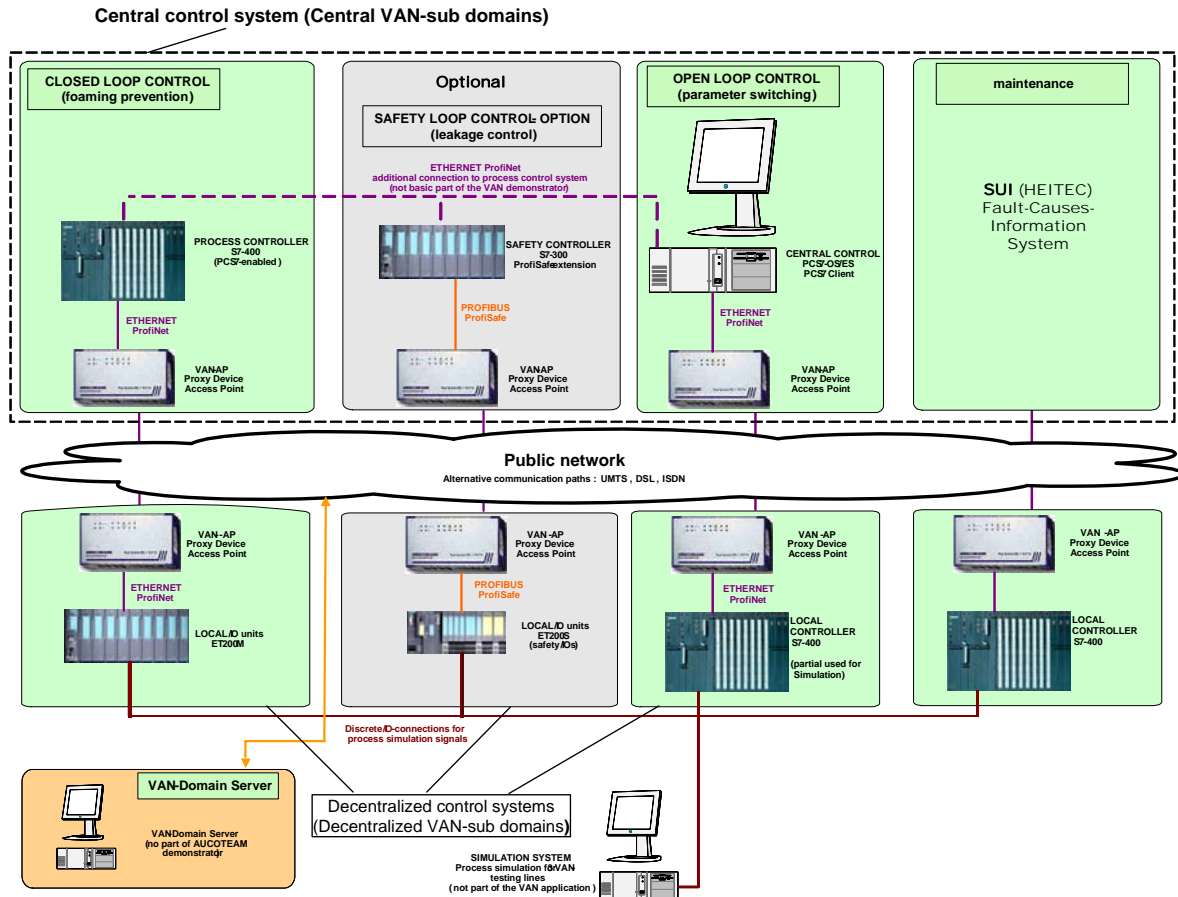


Fig. 5-3: Technical structure of the central control of decentralized systems

Realization of the technical structure using VAN-devices (Realization matrix – Fig. 5.7)

	Telecontrol Means	PROFINET IO Means
VAN PD	<p>Central Control System:</p> <ul style="list-style-type: none"> Hard PLC as PROFINET CBA Peers Difference transmission (cyclic scan) as input Control value as output <p>Bio Reactor</p> <ul style="list-style-type: none"> Actors/ sensors as PROFINET CBA Peers <p>Storage</p> <ul style="list-style-type: none"> Actors/ sensors as PROFINET CBA Peers Usage of today's safety devices <p>Power Plant</p> <ul style="list-style-type: none"> Actors/ sensors as PROFINET CBA Peers 	<p>Central Control System</p> <ul style="list-style-type: none"> HARD PLC as PROFINET IO Controller Cyclic transmission <p>Bio Reactor</p> <ul style="list-style-type: none"> Actor/ sensors as PROFINET IO devices <p>Storage</p> <ul style="list-style-type: none"> Actor/ sensors as PROFINET IO devices Safety integrated in VAN PD <p>Power Plant</p> <ul style="list-style-type: none"> Actor/ sensors as PROFINET IO devices
VAN AD	<p>Central Control System</p> <ul style="list-style-type: none"> Soft PLC as PROFINET VAN Telecontrol Peer Difference transmission (cyclic scan) as output <p>Bio Reactor</p> <ul style="list-style-type: none"> Actors/ sensors with VAN Telecontrol <p>Storage</p> <ul style="list-style-type: none"> Actors/ sensors with VAN Telecontrol No Safety Power Plant Actors/ sensors with VAN Telecontrol 	<p>Central Control System</p> <ul style="list-style-type: none"> Actor/ sensors as VAN Controller Cyclic Transmission <p>Bio Reactor</p> <ul style="list-style-type: none"> Actor/ sensors as VAN Devices <p>Storage</p> <ul style="list-style-type: none"> Actor/ sensors as VAN Devices No Safety <p>Power Plant</p> <ul style="list-style-type: none"> Actor/ sensors as VAN Devices

Fig. 5-4: Realization Matrix

With regards to the realization matrix (Fig. 5.7) the general structure of application profiles is shown in Fig. 5.8.

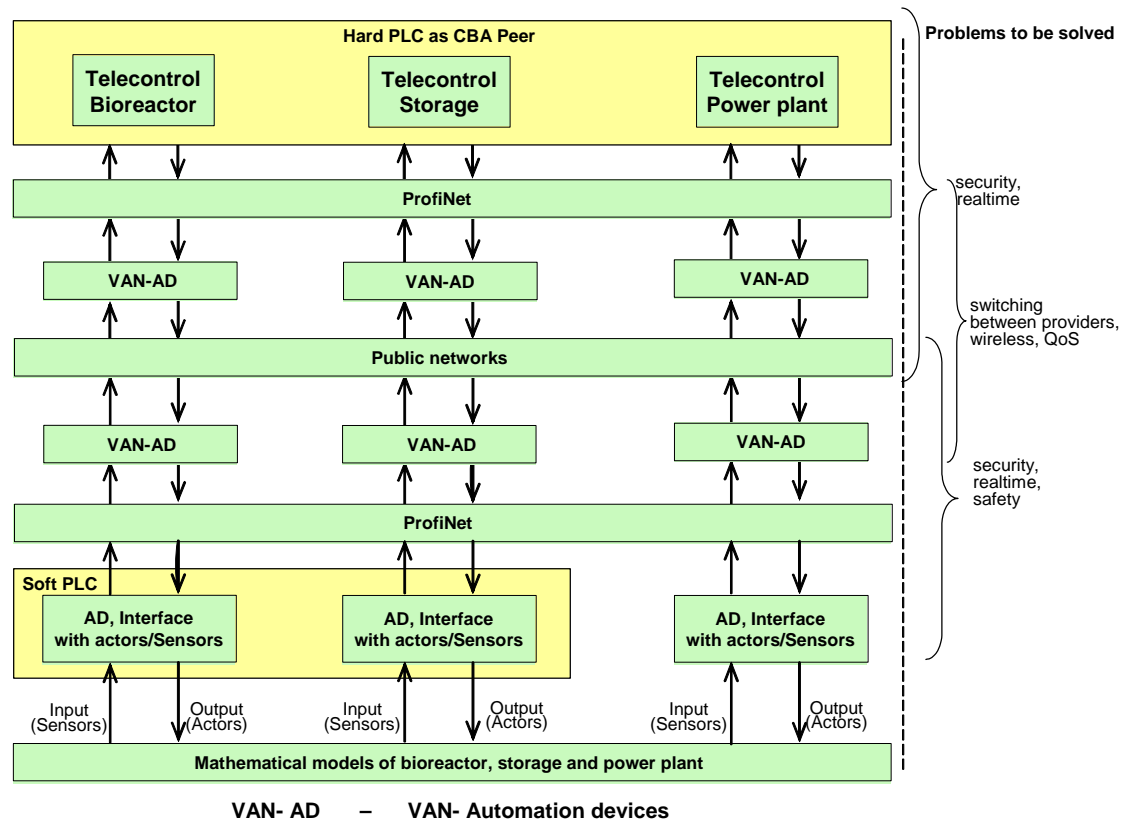


Fig. 5-5: General structure of application profiles for the VAN-based control system in the process industry

Important to note, that the basic of the definition of the application profiles is the object-oriented framework of internal modules (Components) to be defined in the Task 2.4. In this framework will be created a ASE Hierarchy. The same approach we use for the definition of application profiles.

A process monitoring and control system is a distributed system in which modular control structures are managed and executed in the most efficient way possible. This applies in particular to the local (process-level) area. The process control system should make it easy for the user to work with a modular system in a similar way to a system based on discrete devices.

One characteristic of a modular system is that a module-with all its parts, data records, connectors, and methods-is treated as a unit. In Process control, from the communications standpoint, the module can be identified by a single reference.

Such an object-oriented modular concept can be implemented in either of two ways: by realizing the modules as flat, completely autonomous objects, or by realizing them as entities belonging to a class hierarchy.

Methods using discrete devices correspond to the first approach. Each compact controller (software) has its own data storage, its own interfaces, and its own logic for implementing the PID algorithm, for example for foaming prevention in the bioreactor. In a software-type concept, which follows the second approach, the modules are realized as instances of module types; this corresponds to a special, one-level abstraction scheme.

When it is realized in the functional system, each module is therefore split into two objects, a type object and an instance object. The type object contains the processing methods, a description of the data structure, and a description of the connector structure. The instance object (entity) contains the module name, the individual status information, and a reference to the type object.

This functional structure implies two properties that are significant for module handling in process control systems:

- Type objects and entities can be created and loaded separately
- More than one entity can be defined for one type object

Figure 5.8 shows for example realizations of three controller modules. All three use the common type object "PID controller", modules TA 12L1, TA 14F1, and TA3T2 are three entities.

As you can see the hierarchical structure of software implementation of three modules corresponds with the ASE Hierarchy (Task 2.2).

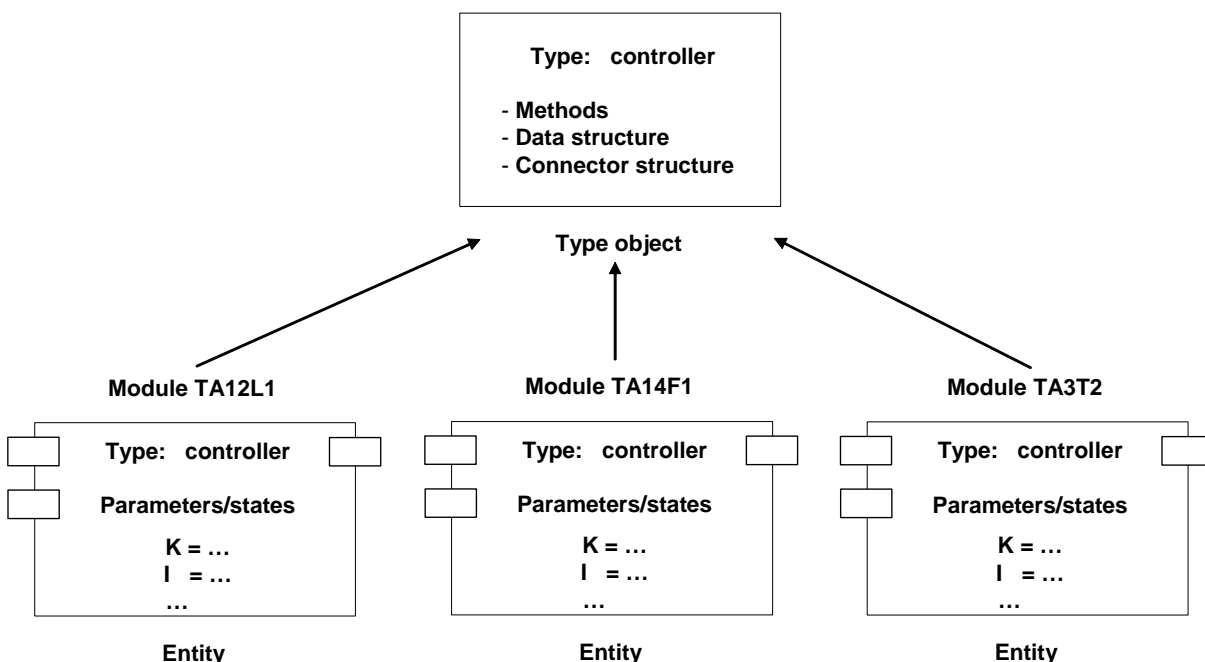


Fig. 5-6: Software implementation of three modules of the type "controller"

A major task of the process control system is to manage and dynamically perform communications in the VAN-network of modules.

5.2.2 Difference between application profiles VAN Solutions in the process industry and in the manufacturing industry

Specific properties of process industry (Different from the manufacturing industry) are:

- continuous processes
- control tasks are defined as compensation of correctable
- and uncorrectable disturbances (Stabilisation, optimisation, safety control)
- control activities in the case of disturbances only,

Specific properties of manufacturing industry are:

- the machines cannot operate without the control system
- the control systems are an integrated part of machines

Continuous activities of control systems, not only in the case of disturbances

5.2.3 Requirements for Process Industry

Requirements to central control system with integrated VAN-solutions are:

- Real-time requirements
- Security requirements
- Safety requirements
- Combination of mixed wireless and wired communication (specification of interfaces)
- Combination of mixed private and public communication
- Scalability of security and real-time
- Guaranteed availability

The Costumer benefits of centralized control of distributed technological plants using heterogeneous network in the process industry are:

- Know- how of one operator or maintenance engineer can be used for the control of a great number of plants without time-delay
- Model based control system can be used for a great number of plants
- The costs of centralized control system can be divided in the number of decentralized technological plants
- Integration of operator training systems (OTS) in the centralized control system
- Estimated benefit in the process industry (Steel industry and in the Waste industry) - 30% raising of the profit
-

5.2.4 VAN requirements for Application profiles of the Process Industry

Classification for Application profiles:

VAN Integration Level I (VIL I)	VAN minimum functionality	- stage 1
VAN Integration Level II (VIL II)	VAN basic functionality	- stage 2
VAN Integration Level III (VIL III)	VAN advanced functionality	- stage 3

Communication Profiles for Process industry demonstrator			
Feature Devices	VIL I	VIL II	VIL III C
Dynamic DNS Server	m	m	m
Proxy Device (VAN-PD)	m	m	m
VAN Automation Device (VAN-AD)	m	m	m
Access Point (VAN-AP)	o	r	m
VAN Virtual Device (VAN-VD)	o	r	m

Legend:

- m = mandatory
- r = recommend
- o = optional
- f = forbidden


 = minimum for process industry demonstrator

Table 5-6: Required VAN devices for process industry demonstrator

According to the definition in D02.2-2 chapter 3.3 a VAN device consists of three major layers:

- VAN Network Layer,
- VAN Communication Layer,
- VAN Application Layer.

5.2.5 Device profile objects selection

Major layer	Feature/Object	Description	VIL I	VIL II	VIL III	Remarks
VAN Network Layer	VAN Heterogeneous Network Technology Adaptation Layer	See D02.2-1 chapter 2.4.10	m	m	m	The implementation of these Interface should consist of typical driver operations like open(), close(),read(), write() and ioctl(). The implementation of the interface is local matter.
	IEEE 802.3	Access to standard Ethernet network.	m	m	m	The access over IEEE 802.3 to VAN communication environment
	PROFINET lower layer	Standard PROFINET Driver	r	r	m	It is the common access to the VAN communication environment.
	ATM / Frame Relay		o	o	o	
	ISDN / Q .931		o	o	o	

Major layer	Feature/Object	Description	VIL I	VIL II	VIL III	Remarks
	SDH / SONET		o	o	o	
	UWB		o	o	o	
	WiMAX		o	o	o	
	WLAN		o	o	o	
	UMTS		o	o	o	
	GPRS		o	o	o	
	Bluetooth		o	o	o	
VAN Communication Stack	VAN Runtime Object Dispatcher	Dispatches automation objects conveyed via the runtime object tunnel.	o	r	m	Implementation is local matter and should be optimized to fit the vendor specific upper layer. Examples of using in combination with VAN runtime object tunnel see D02.2-2 Chapter 9.
	VAN Runtime Object Tunnel	The tunnel is an interface to the application layer. 3 levels and interface to the application layer	o	r	m	The tunnel is divided into different levels one see API description D02.2. Example of using in combination with VAN runtime object tunnel see D02.2-2 Chapter 9.
	Web Service over Virtual Automation Networks	Provides the WS of the VAN device	m	m	m	The name based access via Web Service to the devices is one key concept of VAN.
	ACL	Access Control Layer assures the authorised access to objects in the VAN device.	o	r	m	The functionality is topic of WP 6.
	VAN Common Communication Application Service Elements	For all ASE see API specification in D02.2.-2	m	m	m	A minimal subset of ASE is need for each VAN device.
	VAN Domain	connection parameters (incl. VAN Runtime Object Tunnel)	m	m	m	
	VAN Diagnosis	VAN specific diagnosis, status information of ASEs (e.g. config_status, VAN_device_status)	o	r	m	
	VAN Dev Config	Operating parameters of functional parts of a VAN component	o	r	m	

Major layer	Feature/Object	Description	VIL I	VIL II	VIL III	Remarks
	VAN Access Management	provider access information, appropriate login information and mechanism to handle the login	m	m	m	
	VAN Update Sevice	firmware, version information, update rules, list of update objects	o	r	r	
	VAN Location	geographical (GPS, Galileo), logical (e.g. belongs to radio cell, local language ...)	o	r	r	
	VAN Security Config		r	m	m	
	VAN Switching	alternative path switching (e.g line redundancy, least cost routing)	m	m	m	
	VAN Peer-to-Peer	container for VAN internal communication	o	o	o	
	VAN Routing	routing between VAN subdomains	m	m	m	
	VAN Redundancy	low priority (optional) -> not to be realized in this step of the project (board decision)	o	r	m	
	VAN Time	time in PROFINET format, time synchronisation type, master/slave function, synchronisation status	o	r	r	
	Security		o	r	m	The functionality is topic of WP 6
	TCP/IP Stack		m	m	m	Defines TCP/IP Stack for the Web Service Access to the Device.
	SNMP	Only for monitoring purposes	m	m	m	
	HTTP	HTTP Protocol as carrier for SOAP				
	DNS					
	DHCP		m	m	m	
	ICMP v4/v6		o	o	r	

Major layer	Feature/Object	Description	VIL I	VIL II	VIL III	Remarks
	ARP		o	o	r	
	IPv4/v6		m	m	m	
	TCP		m	m	m	
	UDP		f	f	f	
	VAN Device and Network Management		o	o	r	
	Time Sync	Realizes the time synchronisation between VAN Devices: <ul style="list-style-type: none"> - Within one LAN - Over InterLAN - or WAN 	o	r	m	The necessary attributes an the specification of the messages is done in D04.3
Application Layer	VAN Automation Application Process		m	m	m	
	VAN System Management Application Process	Controls all parts of VAN Communication Stack and Network Technology Layer	o	r	m	
	VAN Security Application Process		o	r	m	The functionality is topic of WP 6
	Parameter Config.	Is an kind of data bases for all defined parameters.	o	r	m	Discussion in Tech PCC which kind of storing mechanism will be provided by VAN devices.
	VAN PnP Application Process	Is used for automatic configuration and identification of connected VAN-Devices.	m	m	m	Parameter are located within the ASEservices and parameters are used by the PnP.
	VAN Proxy Application Process		m	m	m	A VAN-AD does not provide any Proxy Function for other Devices.
	Web Server		o	m	m	

Table 5-7: Device profile object selection

5.2.6 VAN based central control for decentralised plants

5.2.6.1 Overview of the evaluation platform for process control specific test scenarios

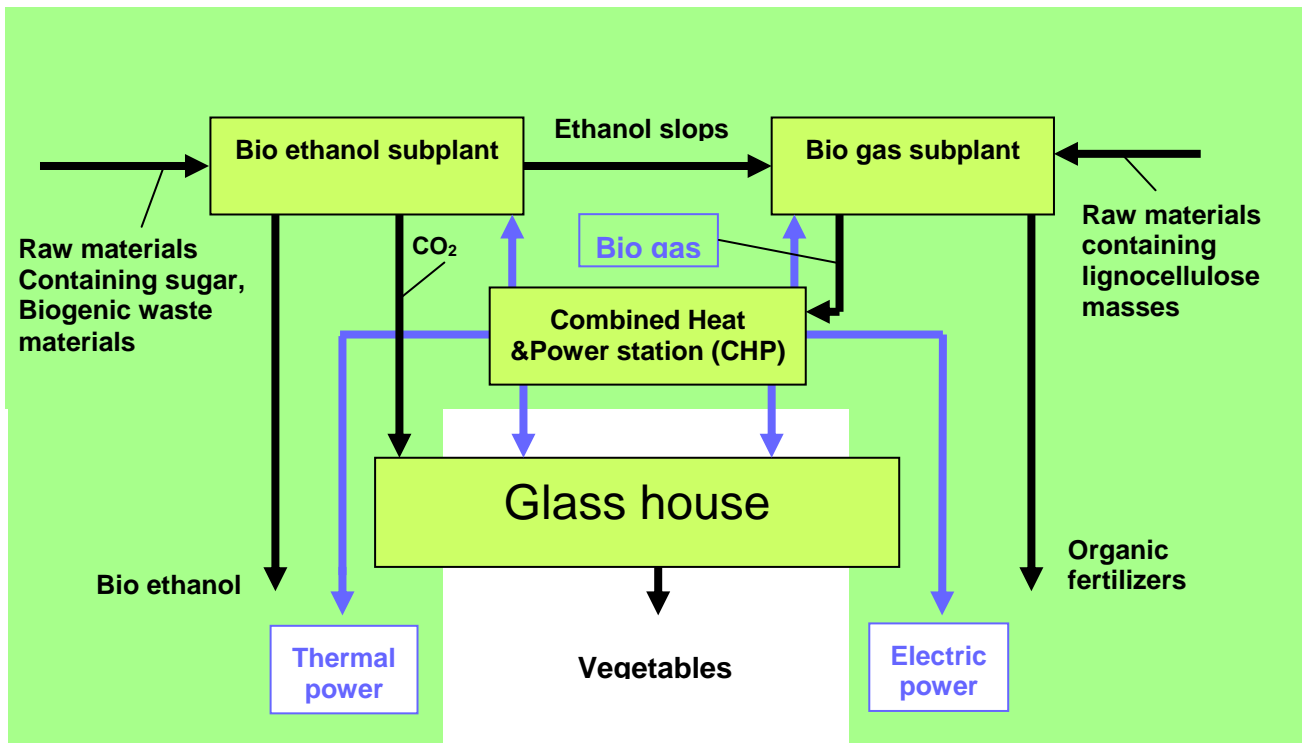


Fig. 5-7: Technological flow sheet of combined biotechnological plants

The test configuration for VAN validation and test platform regarding the process industry is a combination of biotechnological plants (Fig. 5-7:), that contains various plants:

- A bioreactor that operates to transform and degrade biogenic waste material and produces gas
- A gas cleaning and storage plant
- A combined heat and power station for the production of heat and electric power

A test of the VAN functionality in a real industrial domain seems to be too dangerous and is not supported by most of the plant operators. For this reason the VAN evaluation process for the process industry will be spitted in two steps:

- STEP1: **Test of VAN solutions using simulated technological plants**
- STEP2: **Application of VAN solutions in real technological plants after successful evaluation with simulated processes**

The VAN evaluation platform for process industry specific tests uses a PC based process simulation system, that has to be connected to a Siemens PLC via a normal (non VAN) Profibus interface. A specialised software system simulates the real process and manipulates the I/O system of the host PLC. The process simulation for not Profibus-enabled devices will be realised via simple I/O connections (analog/digital) between the "simulation PLC" and the other VAN-integrated I/Os.

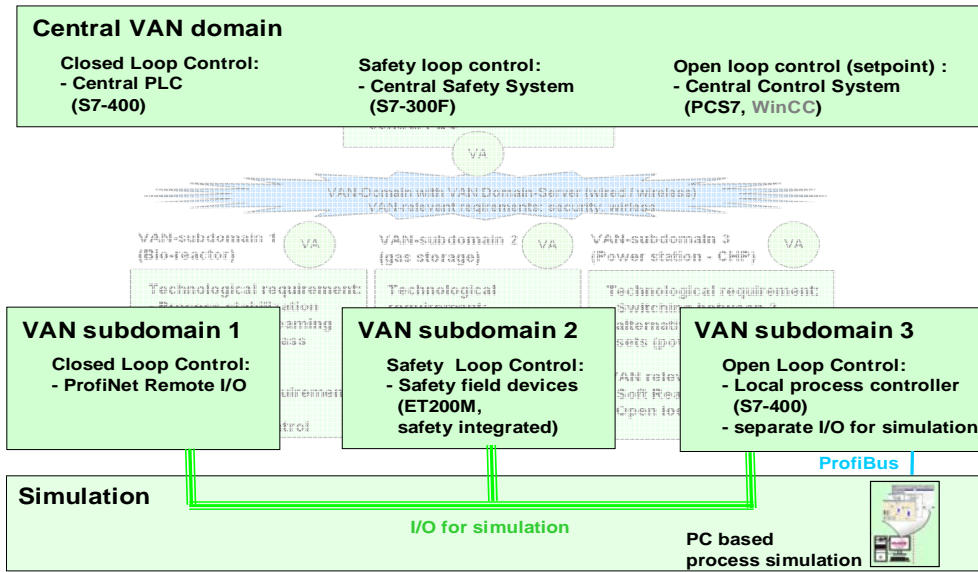


Fig. 5-8: VAN evaluation platform for process industry with process simulation system

5.2.6.2 Test scenario – closed loop control (bio reactor)

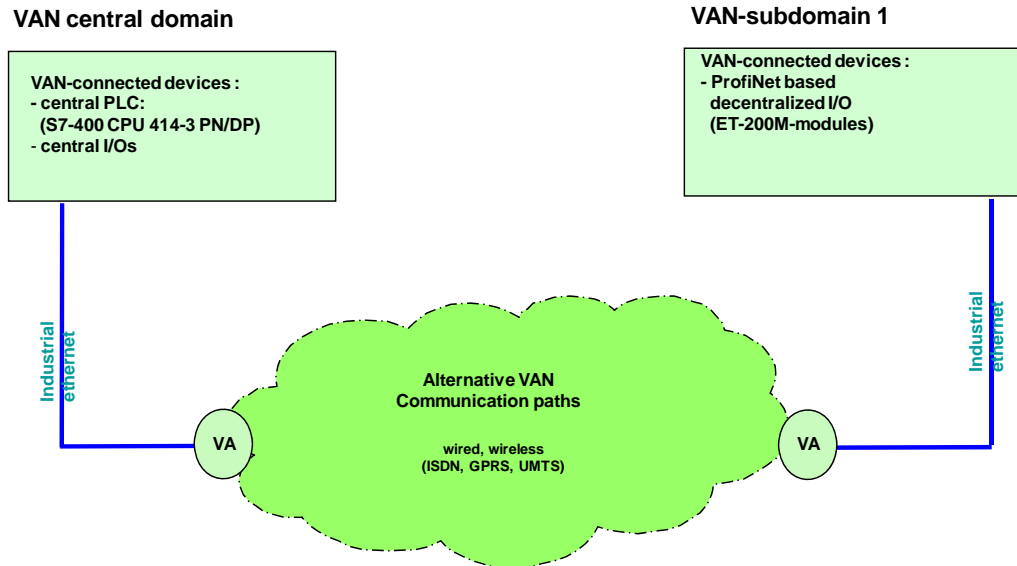


Fig. 5-9: VAN test scenario - bio reactor

5.2.6.2.1 Device Description

This profile specifies the VAN communication path for stabilisation of a decentralised local process with decentralized VAN enabled I/Os and a central process control unit. This central PLC will be used to control several local bio reactors.

The main task of the central PLC is the process stabilisation of the local process by using a closed loop.

This use case is an element of the proposed VAN-demonstrator for the process industry.

The VAN Access Point (VA) on the field site can be a VAN Automation device alternatively.

5.2.6.2.2 Interfaces description

- Local attached remote I/Os
- Ethernet interface
- Alternatively: UMTS, GPRS, ISDN

5.2.6.2.3 Protocols

- TCP/IP on Ethernet interface (ProfiNet, Industrial Ethernet)

5.2.6.2.4 Communication requirements

5.2.6.2.5 Object model

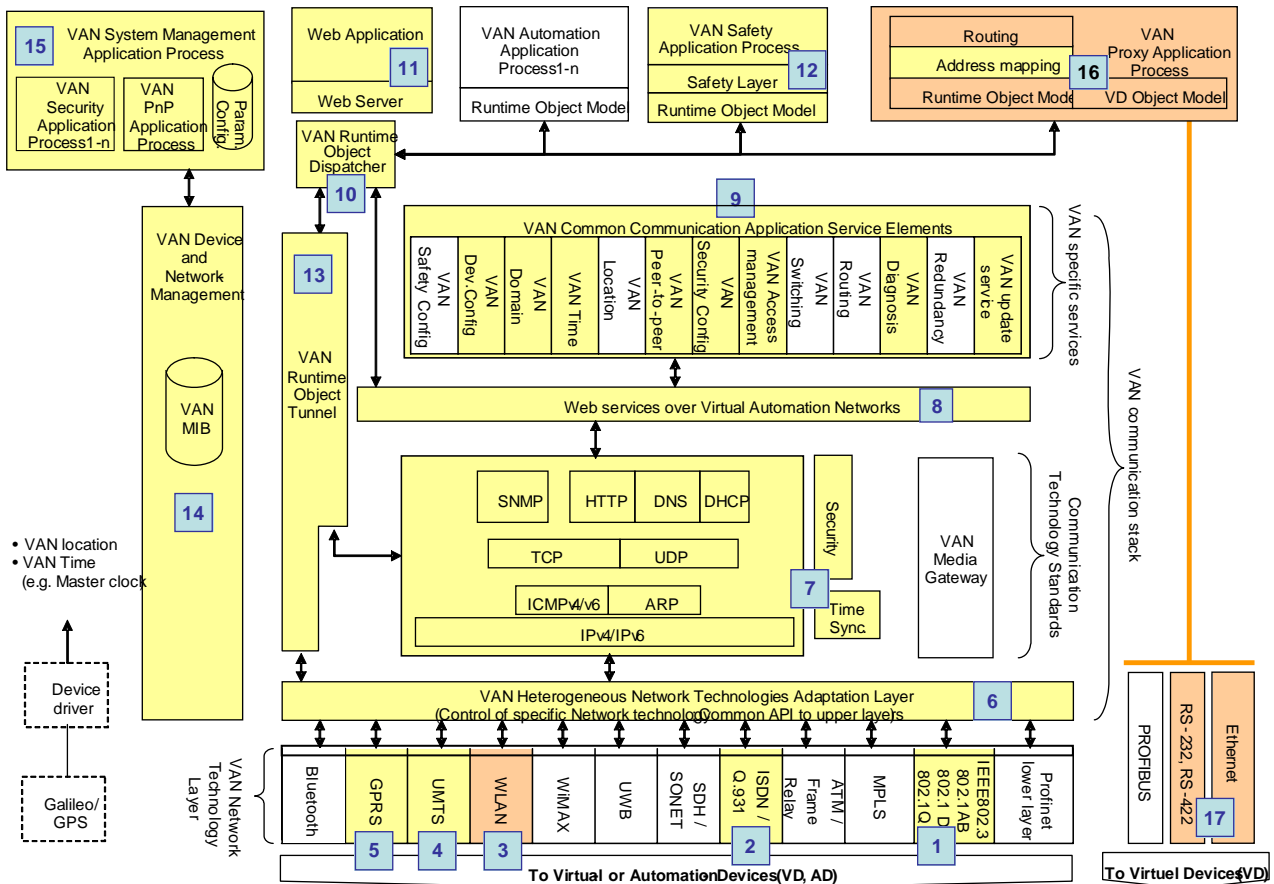


Fig. 5-10: Object model for VAN test scenario - bio reactor

5.2.6.2.6 Required objects

The following table lists device objects highlighted in the VAN device architecture diagram of *VAN test scenario Bio-reactor* (Figure 5-12)

N	Object	Description
1	PROFINET lower layer	PROFINET interface
2	IEEE 802.3	Standard 10/100 Mb/s Ethernet interface.
3	ISDN	Alternative communication path for public communication channels, supported by the VAN-AP (switching event, depending on the QoS-Level of the communication path)
4	WLAN	Wireless LAN interface, optional in this use case, for local connected VAN-devices
5	GPRS / UMTS	Alternative communication path for public communication channels, supported by the VAN-AP (switching event, depending on the QoS-Level of the communication path)
6	VAN heterogeneous network technologies adaptation layer	Supplies a common API between the Ethernet or the WLAN layer and the IP stack.
7	Communication technology standards	IP stack, time synchronisation protocols, security protocols.
8	Web Services over VAN	Web interface to the different VAN ASE specific services and to the VAN runtime dispatcher.
9	VAN common communication Application Service Elements (ASE)	VAN specific services: update, diagnosis, access management, security configuration, peer-to-peer, time, domain, device configuration.
10	VAN runtime object dispatcher	Dispatches automation objects to the VAN proxy application process.
11	Web application and Web server	Web server integrated in the VAN-PD section.
12	VAN application process	VAN test szenario Bio-reactor
13	VAN runtime object tunnel	The transfer of runtime data from the communication layer to the proxy application and vice-versa.
14	VAN device and network management	Provides a MIB database information collected on the device: configuration, runtime parameters, etc. so the object can be managed via the SNMP protocol.
15	VAN System Management Application Process	VAN PnP Application Process (essential in VAN), configuration parameter, security application process.
16	VAN proxy application process	The proxy implementation elements.
17	VAN virtual device interfaces	Different types of possible VAN virtual device interfaces: Ethernet, RS-232/RS-422. (optional in this use case, for local connected VAN-devices)

Table 5-8: Profile objects for VAN test scenario Bio-reactor

5.2.6.3 Test scenario – safety loop control (gas storage)

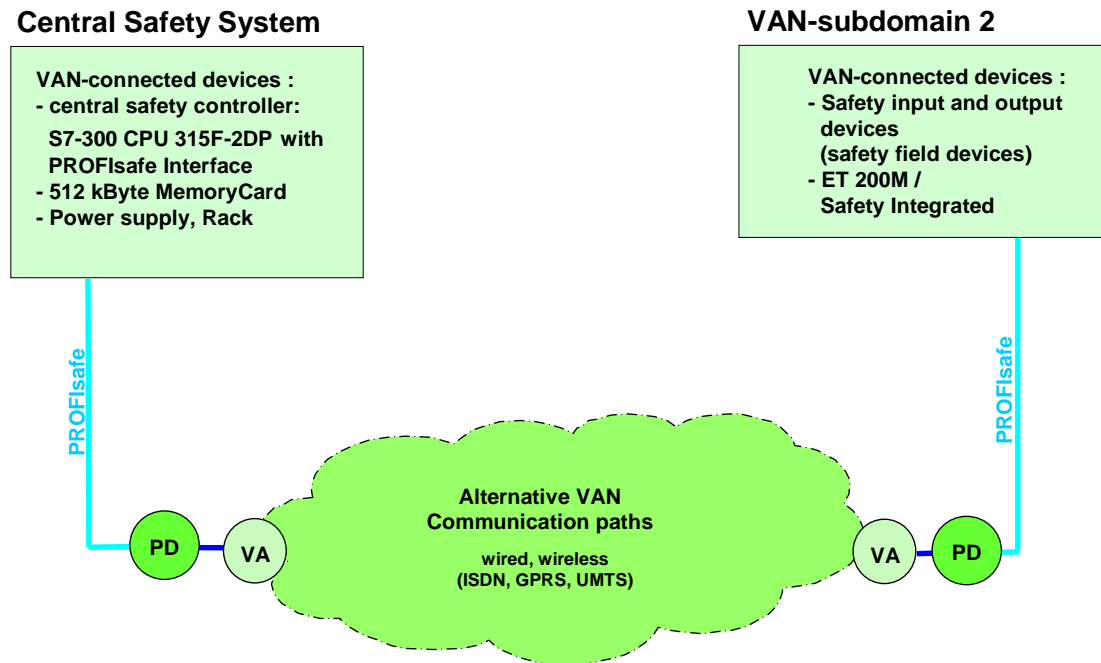


Fig. 5-11: VAN test scenario - gas storage leakage control

5.2.6.3.1 Device Description

This profile specifies the VAN communication path for the control of a safety loop, used in the gas transportation and gas storage system.

The VAN use case uses a decentralised local safety I/O-device for data acquisition and safety cut-off. The main safety controller with the safety application is located central and communicates with the local safety equipment via VAN connections. The main task of the central PLC is the process stabilisation of the local process by using a closed loop.

This use case is another element of the proposed VAN-demonstrator for the process industry.

The VAN Access Points (VA) and the VAN Proxy devices (PD) can be joined into a VAN Automation device alternatively.

5.2.6.3.2 Interfaces description

- Local attached remote safety I/Os
- Ethernet interface
- Alternatively: UMTS, GPRS, ISDN

5.2.6.3.3 Protocols

- TCP/IP on Ethernet interface (VAN)
- PROFISafe

5.2.6.3.4 Communication requirements

5.2.6.3.5 Object model

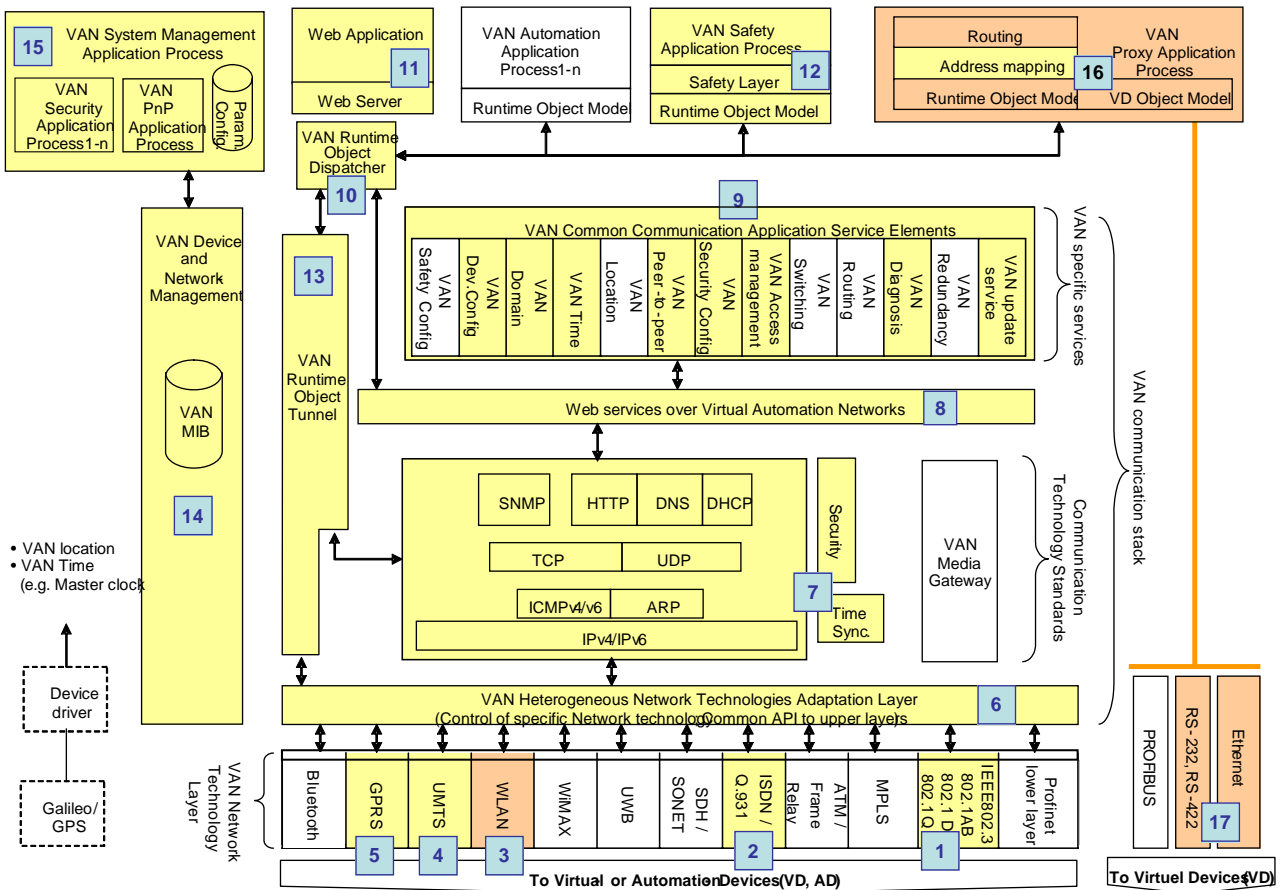


Fig. 5-5: Object model for VAN test scenario - gas storage leakage control

5.2.6.3.6 Required objects

The following table lists device objects highlighted in the VAN device architecture diagram of VAN test scenario - gas storage leakage control Fig. 5-5.

N	Object	Description
1	IEEE 802.3	Standard 10/100 Mb/s Ethernet interface.
2	ISDN	Alternative communication path for public communication channels, supported by the VAN-AP (switching event, depending on the QoS-Level of the communication path)

N	Object	Description
3	WLAN	Wireless LAN interface, optional in this use case, for local connected VAN-devices
4	GPRS	Alternative communication path for public communication channels, supported by the VAN-AP (switching event, depending on the QoS-Level of the communication path)
5	UMTS	Alternative communication path for public communication channels, supported by the VAN-AP (switching event, depending on the QoS-Level of the communication path)
6	VAN heterogeneous network technologies adaptation layer	Supplies a common API between the Ethernet or the WLAN layer and the IP stack.
7	Communication technology standards	IP stack, time synchronisation protocols, security protocols.
8	Web Services over VAN	Web interface to the different VAN ASE specific services and to the VAN runtime dispatcher.
9	VAN common communication Application Service Elements (ASE)	VAN specific services: update, diagnosis, access management, security configuration, peer-to-peer, time, domain, device configuration.
10	VAN runtime object dispatcher	Dispatches automation objects to the VAN proxy application process.
11	Web application and Web server	Web server integrated in the VAN-PD section.
12	VAN safety application process	VAN test scenario - gas storage leakage control
13	VAN runtime object tunnel	The transfer of runtime data from the communication layer to the proxy application and vice-versa.
14	VAN device and network management	Provides a MIB database information collected on the device: configuration, runtime parameters, etc. so the object can be managed via the SNMP protocol.
15	VAN System Management Application Process	VAN PnP Application Process (essential in VAN), configuration parameter, security application process.
16	VAN proxy application process	The proxy implementation elements.
17	VAN virtual device interfaces	Different types of possible VAN virtual device interfaces: Ethernet, RS-232/RS-422. (not used in the idea of this use case, for local connected VAN-devices)

Table 5-9: Profile objects for VAN test scenario - gas storage leakage control

5.2.6.4 Test scenario – open loop control (power station)

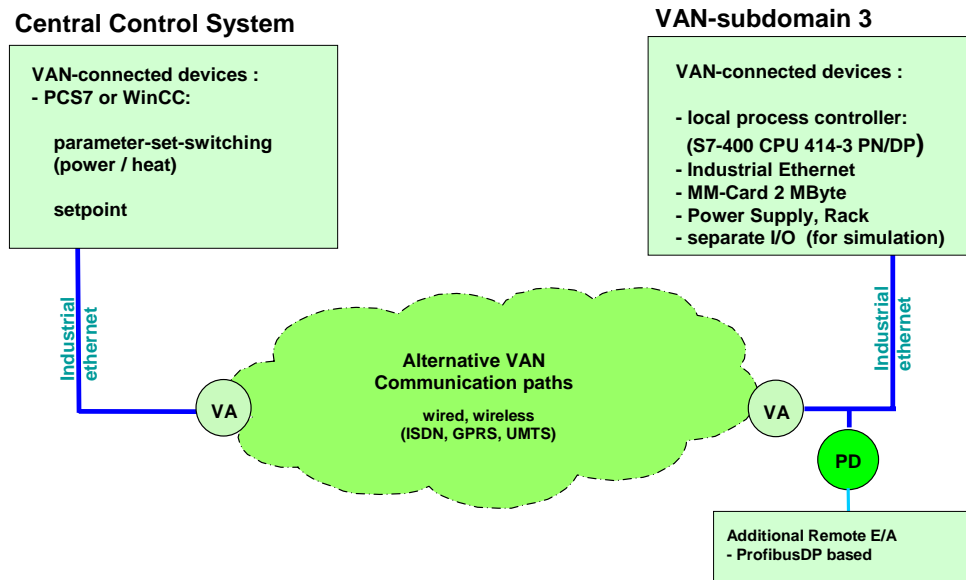


Fig. 5-6: VAN test scenario - parameter switching

5.2.6.4.1 Device Description

This test scenario specifies the VAN communication path for a parameter switching of a CHP. The source of the parameter switching event is either a manual operation from the operator on the central control system (manual mode) or environment affected (VAN-connected temperature sensor).

The VAN use case uses a decentralised local PLC for the control of the CHP, a local sensor for the acquisition of temperature data and a central process control system. The local VAN subdomain is connected with the central process control system via VAN connections (wired / wireless).

Optional this test scenario integrates VAN proxy functionality.

This use case is another element of the proposed VAN-demonstrator for the process industry.

The field-site VAN Access Point (VA) and the VAN Proxy device (PD) can be joined into a VAN Automation device alternatively.

5.2.6.4.2 Interfaces description

- Local attached PLC
- Ethernet interface

- Alternatively: UMTS, GPRS, ISDN

5.2.6.4.3 Protocols

- TCP/IP on Ethernet interface (VAN)
- Profibus

5.2.6.4.4 Communication requirements

5.2.6.4.5 Object model

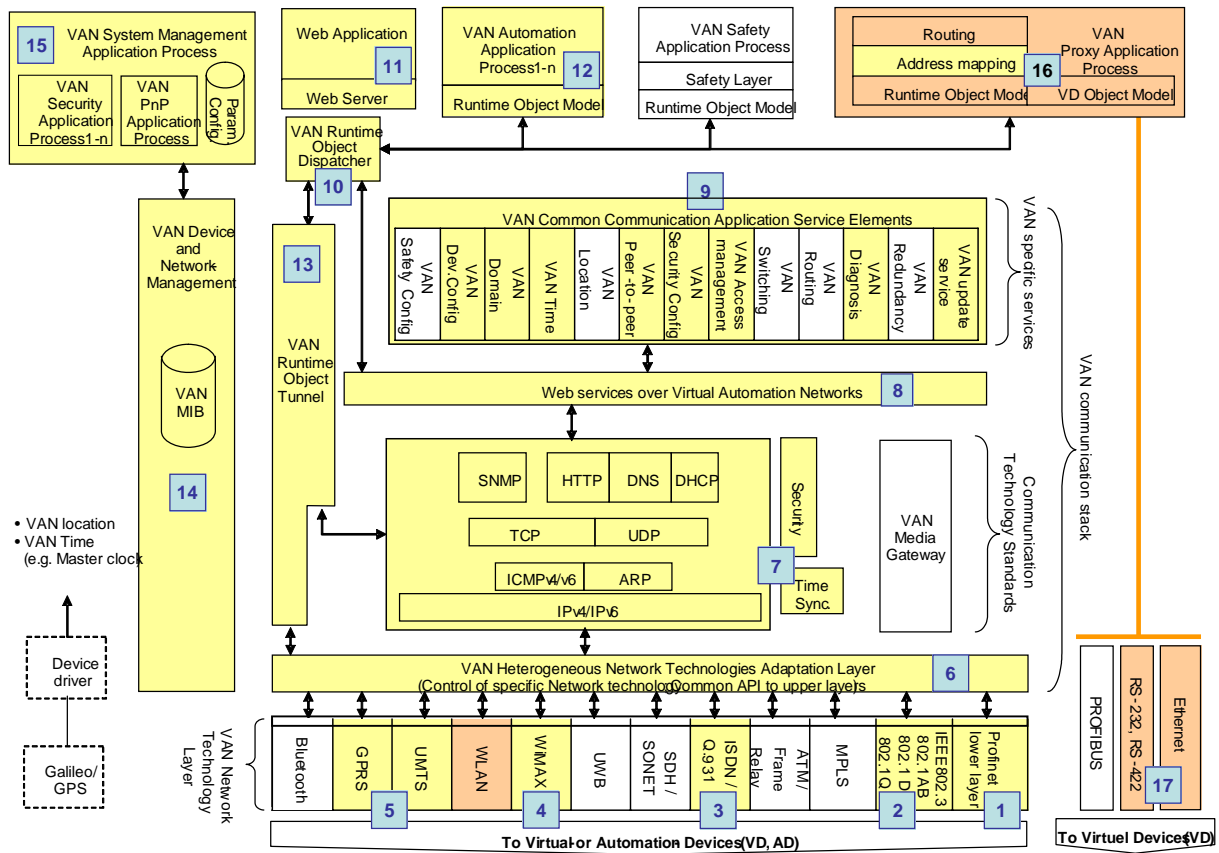


Fig. 5-12: Object model for VAN test scenario - parameter switching

5.2.6.4.6 Required objects

The following table lists device objects highlighted in the VAN device architecture diagram of *VAN test scenario – parameter switching* Fig. 5-12:

N	Object	Description
1	PROFINET lower layer	PROFINET interface
2	IEEE 802.3	Standard 10/100 Mb/s Ethernet interface.
3	ISDN	Alternative communication path for public

N	Object	Description
		communication channels, supported by the VAN-AP (switching event, depending on the QoS-Level of the communication path)
4	WiMax	Worldwide Interoperability for Microwave Access (optional in this use case, for connected VAN-devices in future use cases)
5	GPRS / UMTS	Alternative communication path for public communication channels, supported by the VAN-AP (switching event, depending on the QoS-Level of the communication path)
6	VAN heterogeneous network technologies adaptation layer	Supplies a common API between the Ethernet or the WLAN layer and the IP stack.
7	Communication technology standards	IP stack, time synchronisation protocols, security protocols.
8	Web Services over VAN	Web interface to the different VAN ASE specific services and to the VAN runtime dispatcher.
9	VAN common communication Application Service Elements (ASE)	VAN specific services: update, diagnosis, access management, security configuration, peer-to-peer, time, domain, device configuration.
10	VAN runtime object dispatcher	Dispatches automation objects to the VAN proxy application process.
11	Web application and Web server	Web server integrated in the VAN-PD section.
12	VAN application process	VAN test scenario – parameter switching
13	VAN runtime object tunnel	The transfer of runtime data from the communication layer to the proxy application and vice-versa.
14	VAN device and network management	Provides a MIB database information collected on the device: configuration, runtime parameters, etc. so the object can be managed via the SNMP protocol.
15	VAN System Management Application Process	VAN PnP Application Process (essential in VAN), configuration parameter, security application process.
16	VAN proxy application process	The proxy implementation elements.
17	VAN virtual device interfaces	Different types of possible VAN virtual device interfaces: Ethernet, RS-232/RS-422. (optional in this use case, for local connected additional VAN-devices)

Table 5-10: Profile objects for VAN test scenario – parameter switching

6 Conclusion

After the definition of the general architecture of a VAN device, the topology and APIs of a VAN network, this document describes subsets of general VAN device architecture – VAN architecture device profiles. To identify and to define these profiles was the main goal of the task.

But firstly we had found that it is necessary to refine selected architecture functional blocks. We needed to decide whether these parts are necessary in device profiles or it is just optional functionality. The results are in chapter 2. Media gateway, MIB and SNMP, IP stack, safety, security and runtime tunnel were refined and we identified necessity of these parts for our device profiles.

Then we identified which device profiles are needed to define for all VAN. We also considered that the same device can have different functionality based on application needs. That's why we introduced conformance class concept. Each device profile describes an object subset of a general VAN device which is necessary to realize its functionality in the VAN context. It is not in every case useful/necessary to have the complete functionality in each device. For this reason three device conformance classes were introduced. We also defined the template for device profile description. This is the content of chapter 3.

In chapter 4 we defined three conformance classes for each device profile. VAN-AD, VAN-PD, VAN-AP and PnP Manager were described and defined. How to use each device profile illustrates a relevant use case.

Chapter 5 shows examples how our device profiles with selected conformance classes can work in possible real VAN application. Many example application profiles were introduced for manufacturing (chapter 5.1) and process industry (chapter 5.2).

Our works should be one of the fundamentals for implementation of real VAN devices by consequential task 2.4.

Glossary

ACL	Access Control List
API	Application Programming Interface
AS-Interface	Fieldbus system
ASE	Application Service Element
CNC	Computer Numerical Control
CRC	Cyclic Redundancy Check
DHCP	Dynamic Host Configuration Protocol
DoW	Description of Work
EDS	Electronic Data Sheet
Ethernet/IP	Fieldbus system
FTP	File Transfer Protocol
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol with SSL
IO	Input Output
IP	Internet Protocol
IT	Information Technology
PC	Personal Computer
PLC	Programmable Logic Controller
PPP	Point to Point Protocol
PROFIBUS	Fieldbus systems
Profibus-DP	Fieldbus system
PSTN	Public Switched Telephone Network
RFID	Radio Frequency Identification
RS-232	EIA standard for serial interfaces
RS-422	EIA standard for serial interfaces
RS-485	EIA standard for serial interfaces
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SSL/TLS	Secure Socket Layer / Transport Layer Security
TCP/IP	Transmission Control Protocol / Internet Protocol
UDP	User Datagram Protocol
USB	Universal Serial Bus
VAN	Virtual Automation Network
VAN-AD	VAN Automation Device

VAN-AP	VAN Access Point
VAN-PD	VAN Proxy Device
VPN	Virtual Private Network
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
XML	eXtensible Markup Language

References

- [VAN06a] Virtual Automation Networks – Deliverable D02.2-1 – *Topology Architecture for the VAN virtual Automation Domain* – 2006
- [VAN06b] Virtual Automation Networks – Deliverable D02.2-2 – *VAN Open Platform API-Specification* – 2006
- [VAN07] Virtual Automation Networks – *Annex I – “Description of Work” Month 13-30* – Version 1.4 – 2007
- [VAN08] Virtual Automation Networks – Deliverable D05.2-1 – *Requirements specification; Description of runtime architecture, engineering and safety mechanisms - 2006*
- [WSS04] Web Service Security: SOAP Message Security 1.0, OASIS Standard 200401, March 2004, www.oasis-open.org
- [W3C02a] W3C: XML-Signature Syntax and Processing, W3C Recommendation February 2002, <http://www.w3.org/TR/xmlsig-core/>
- [W3C02b] W3C: XML Encryption Syntax and Processing, W3C Recommendation December 2002, <http://www.w3.org/TR/xmlenc-core/>