Embedded Networking Solutions

Twincomm

Bye bye protocol stacks

Outsourcing Embedded Industrial Networking

Kurt van Buul
Twincomm

About Twincomm

D&E event 2018

Twincomm

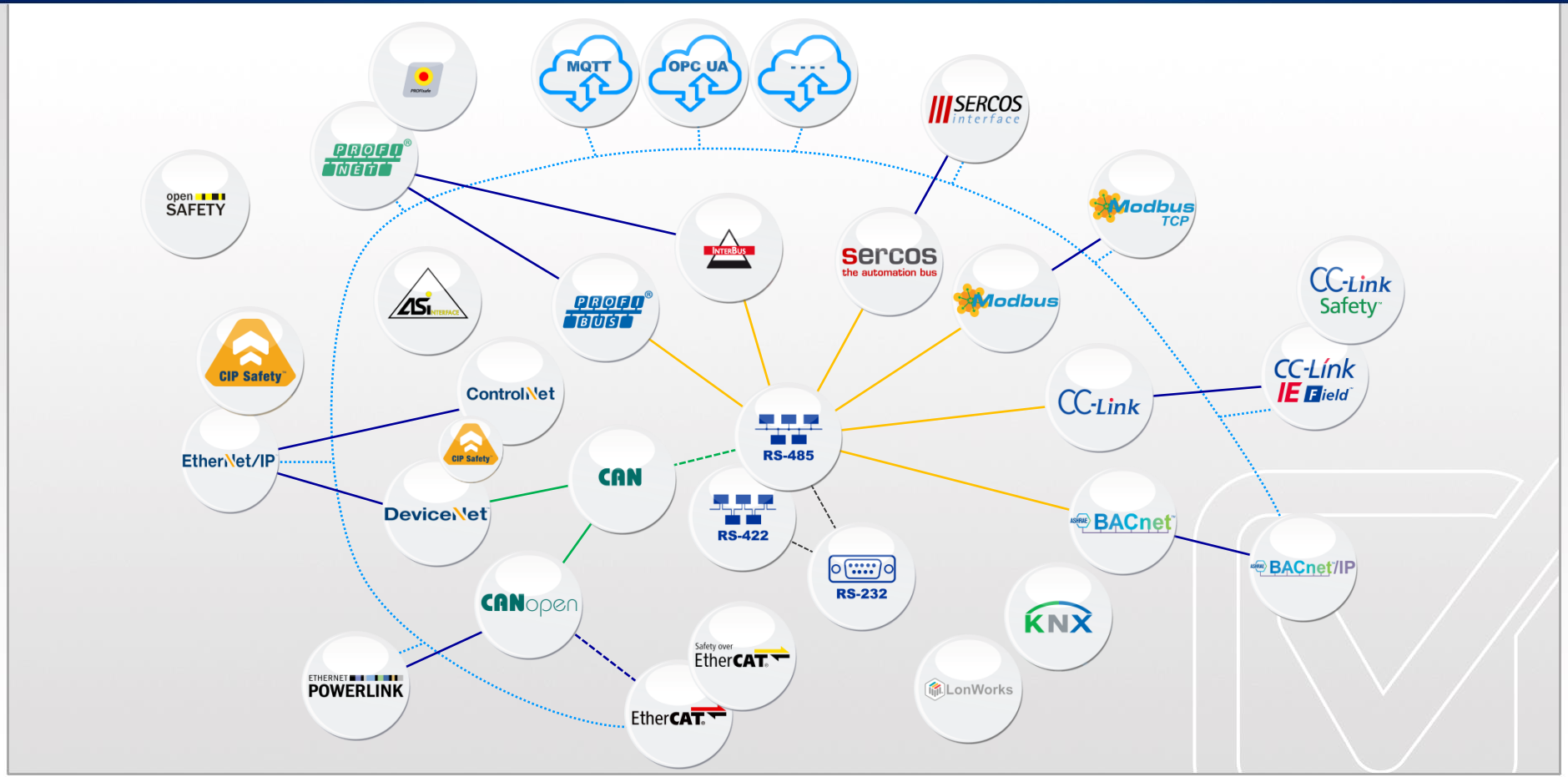
Embedded Networking Solutions

Our products and services enable embedded communication in machines, to systems and between components

PROFIBUS · EtherCAT · EtherNet/IP · ETHERNET POWERLINK · Modbus · sercos the automation bus · CC-Link · BACnet · PROFINET

CANopen · SAE J1939 · DeviceNet · ControlNet · lin · FlexRay · CAN · CAN FD

Fieldbus & Industrial Ethernet

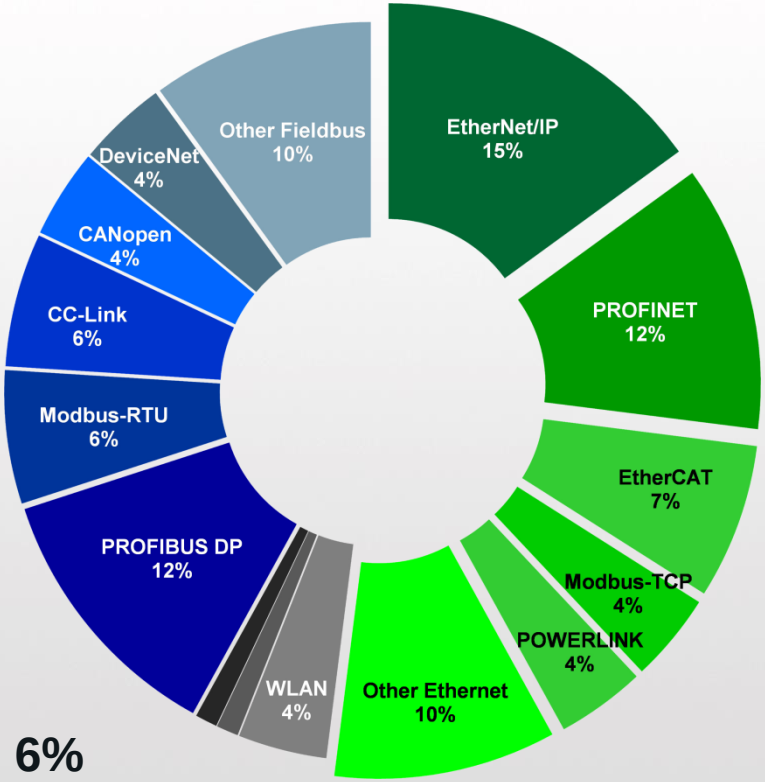


Industrial network shares 2018

Industrial Ethernet 52%

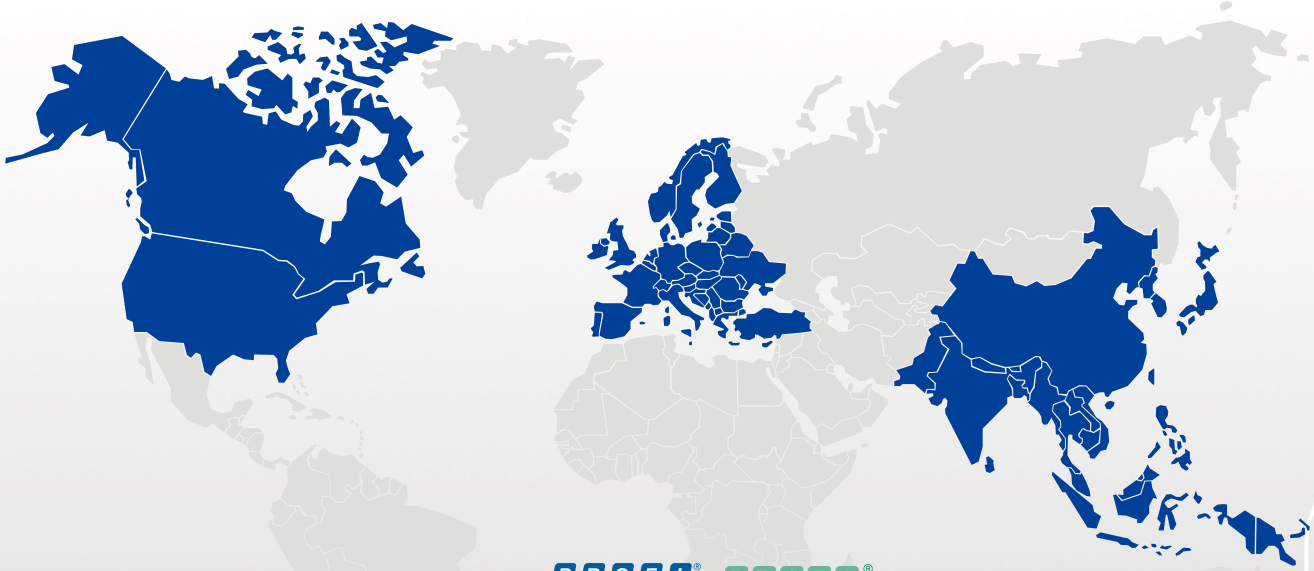
Fieldbus 42%

Wireless 6%



Source: HMS

Geography



EtherNet/IP
EtherCAT

PROFIBUS PROFINET
EtherCAT

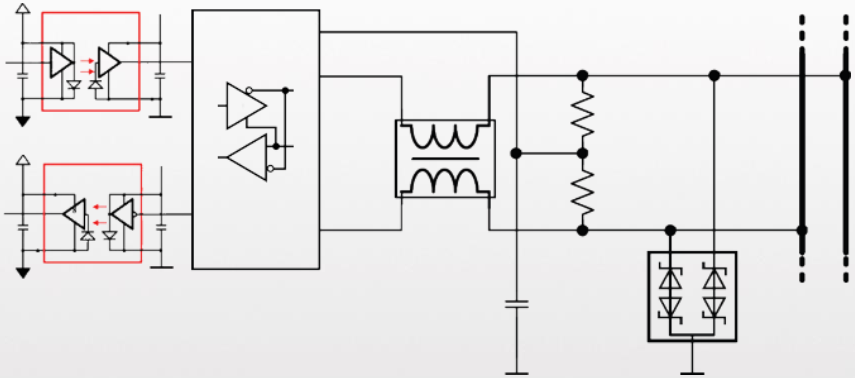
CC-Link IE

Modbus

Fieldbus & Industrial Ethernet

Hardware

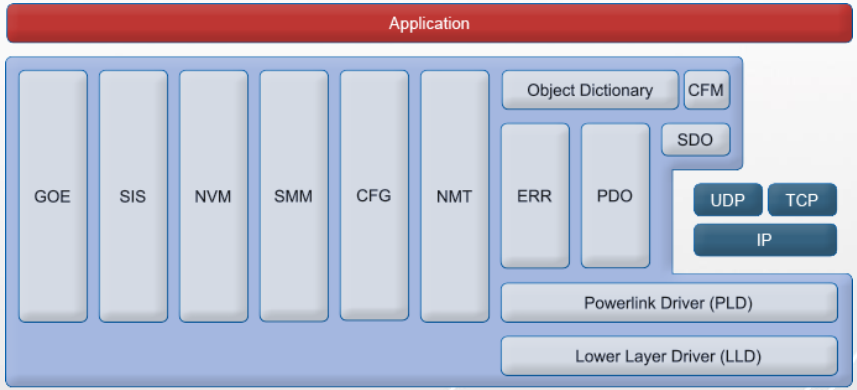
Understanding physical communication



- Every protocol has different physics
- Protection at the right level
- Development time and effort
- Certification

Software

Understanding the protocol

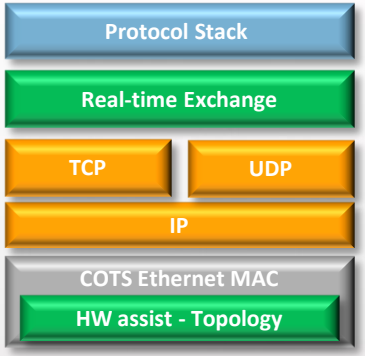


- Protocol Stack + Server, E-mail, FTP, …
- Develop, Buy or Open Source
- Licences
- Certification

Industrial Ethernet - IE handling

Standard TCP/IP

Architecture 1

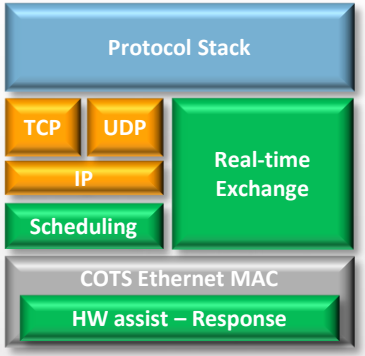


Standard Ethernet TCP/IP

- TCP/IP
- Modbus TCP
- EtherNet/IP
- Etc.

Software by-passing

Architecture 2

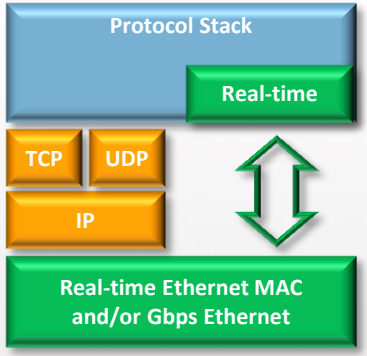


SW by-passing or traffic scheduling

- PowerLink
- PROFINET IO (RT)
- Etc.

Hardware assisted

Architecture 3

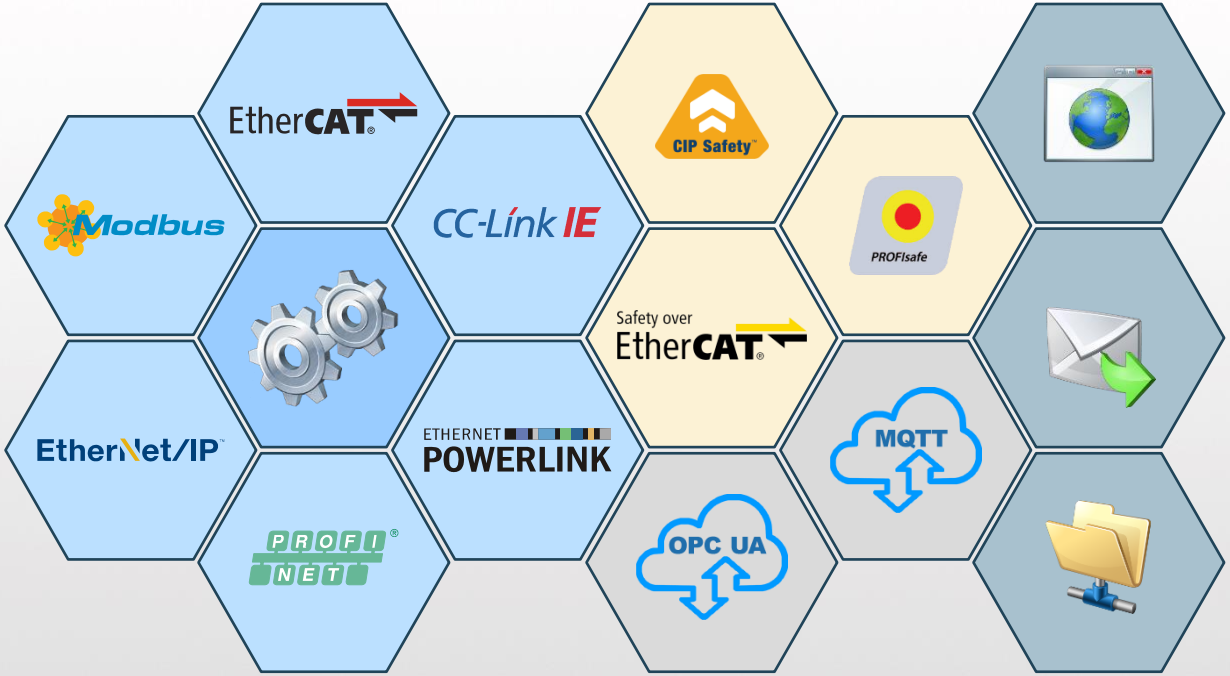


Hardware Assisted Real-time

- PROFINET IO (IRT, DFP)
- EtherCAT
- SERCOS III
- CC-Link IE (Gbps)
- etc.

Industrial Ethernet - IE handling

Multiple-protocol - Software stacks



Software is not standardized

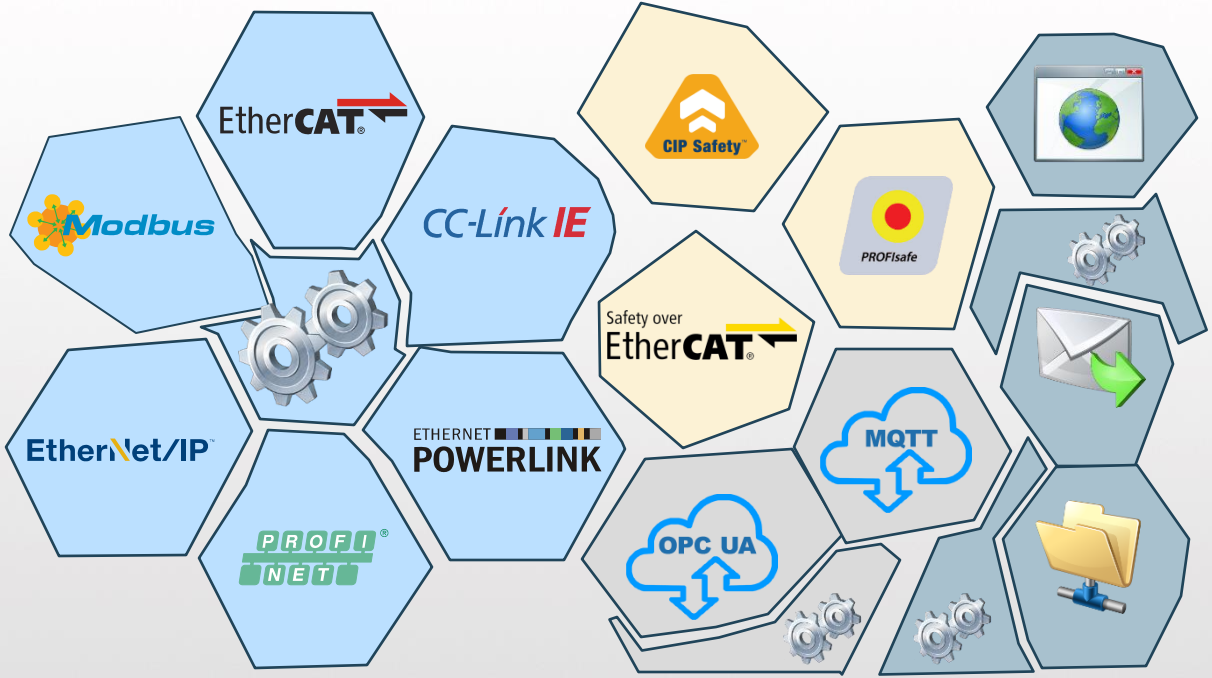
- Different vendors
- Different structures
- Different interfaces
- Different drivers
- Different releases

Special attention

- Non-TCP/IP protocols
- Real-time priority
- Software interference

Industrial Ethernet - IE handling

Multiple-protocol - Software stacks



Software is not standardized

- Different vendors
- Different structures
- Different interfaces
- Different drivers
- Different releases

Special attention

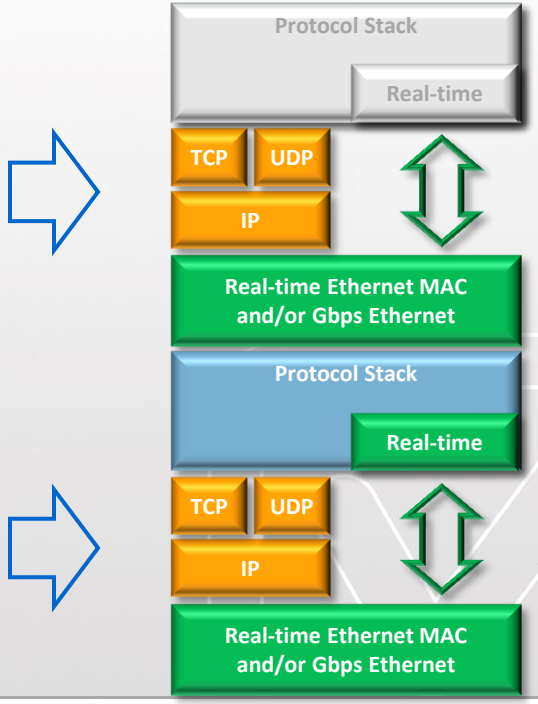
- Non-TCP/IP protocols
- Real-time priority
- Software interference

Multi-protocol software-development is a complex task!

Industrial Ethernet - IE handling

Principle system solution

	Type	Example	Network
	ASIC	Beckhoff ET1100	Fixed 1
	Protocol MCU	TI Sitara AM3357	Fixed 5
	MCU with FPGA	Xylinx Zinq	Flex ±10
	Brick	HMS ABCC B40	Flex ±20
	Module	HMS ABCC M40	Flex ±20



Industrial Ethernet - IE handling

Principle system solution

	Type	Example	Network	Stacks	Certified	Business Initial Cost pu
	ASIC	Beckhoff ET1100	Fixed 1	No	No	⊙⊙⊙⊙ O
	Protocol MCU	TI Sitara AM3357	Fixed 5	No	No	⊙⊙⊙ O
	MCU with FPGA	Xylinx Zinq	Flex ±10	No	No	⊙⊙ OO
	Brick	HMS ABCC B40	Flex ±20	Build-in	Pre	⊙ OO
	Module	HMS ABCC M40	Flex ±20	Build-in	Pre	⊙ OOO

Outsourcing Embedded Industrial Networking

Implementation *"as a service"*

- Common hard- & software
- Complete design-in
- Full support
- Short lead-time

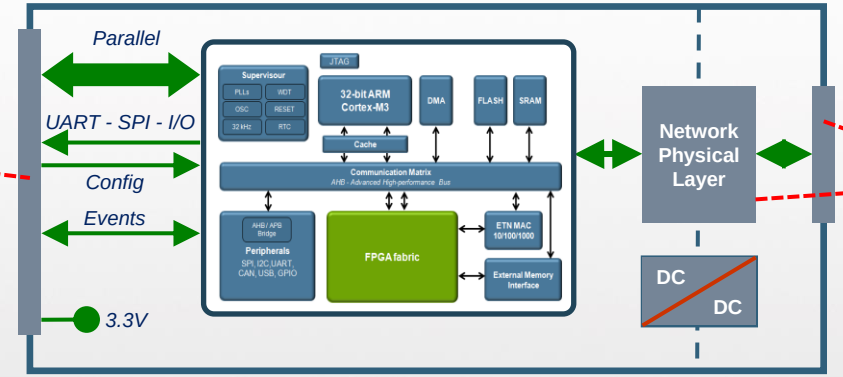
Reliability of third-party

- Pre-certified and certification guarantee
- Latest protocol variants and updates
- Long-term partnership
- Future-proof in development and existence

Under the hood – Block diagram

General hardware and application-interfacing

Common application Interface



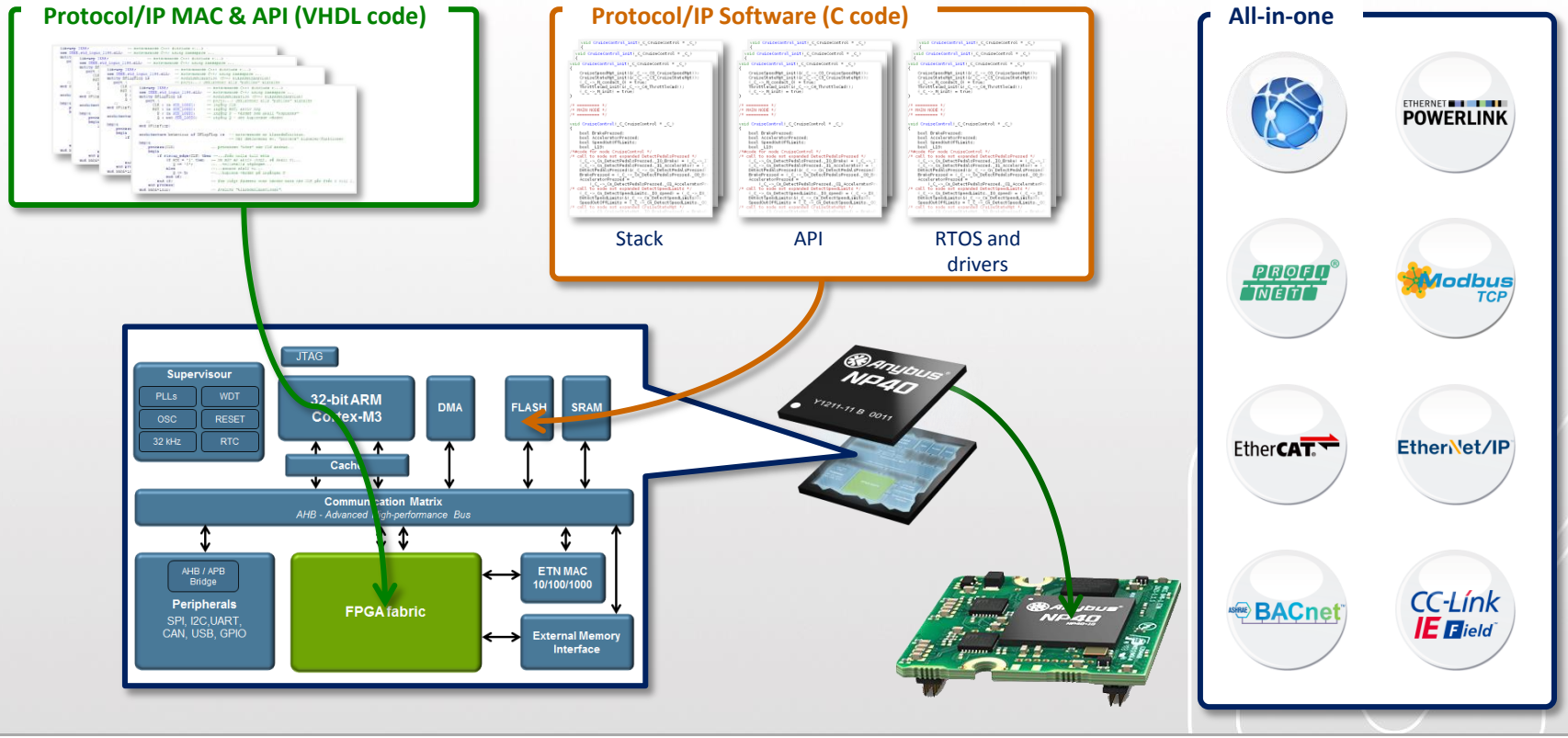
Network specific

Interfacing

- Parallel (8-/16-bit)
- SPI (20 Mhz)
- Asynchrone UART (625 kb/s)
- Stand-alone I/O (12,5 MHz)

Under the hood – Network processor

Programmable hard- & software



Common software

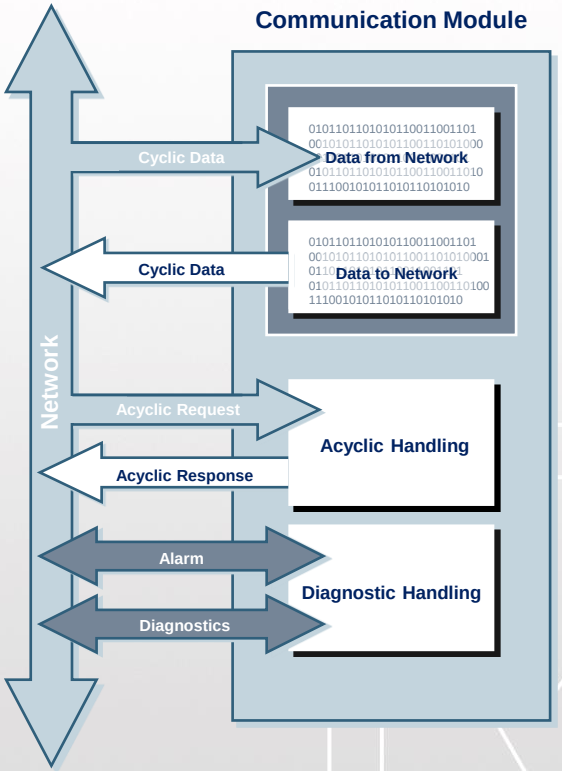
General network communication

Acyclic data handling
- Read or Write Request/Response
- Identification
- Configuration

Data exchange
- Real-time Process Data
- Continuous updating

Alarm
- Warnings & errors

Diagnostics
- Status
- Heart beat



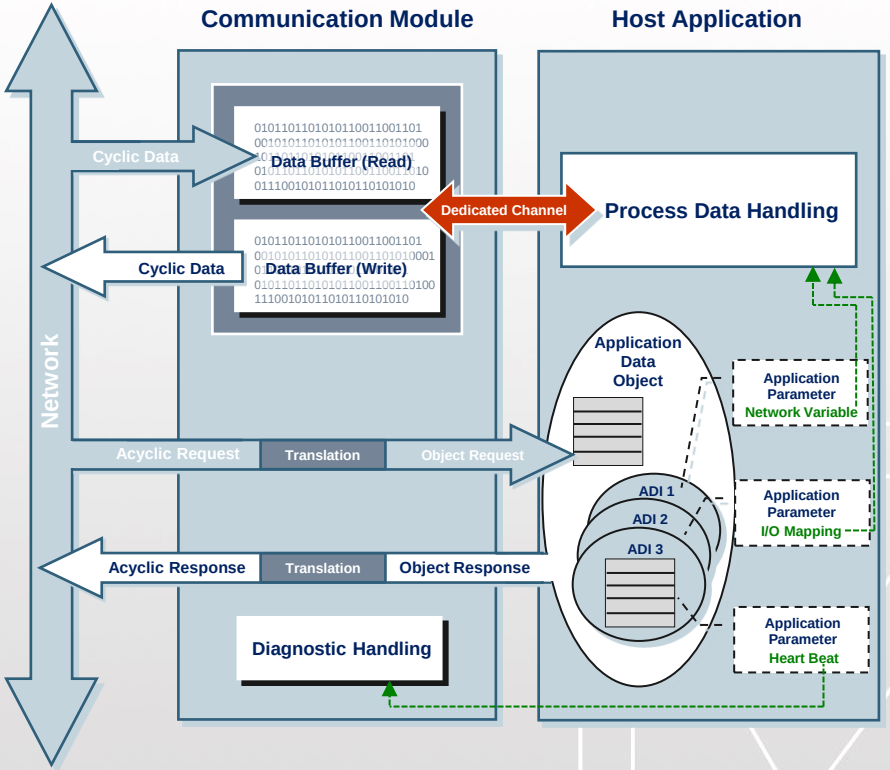
Common software

Software interfacing process

Data objects
- Parameter definitions
- Network variables
- I/O-mapping
- Configuration

Object requests (R/W)
- Passed to application
- Processed
- Response replied

Process data
- Buffered in module
- Continuous updating



Common Software

Software Interface

OS-Independent C-Driver

Application Data Object

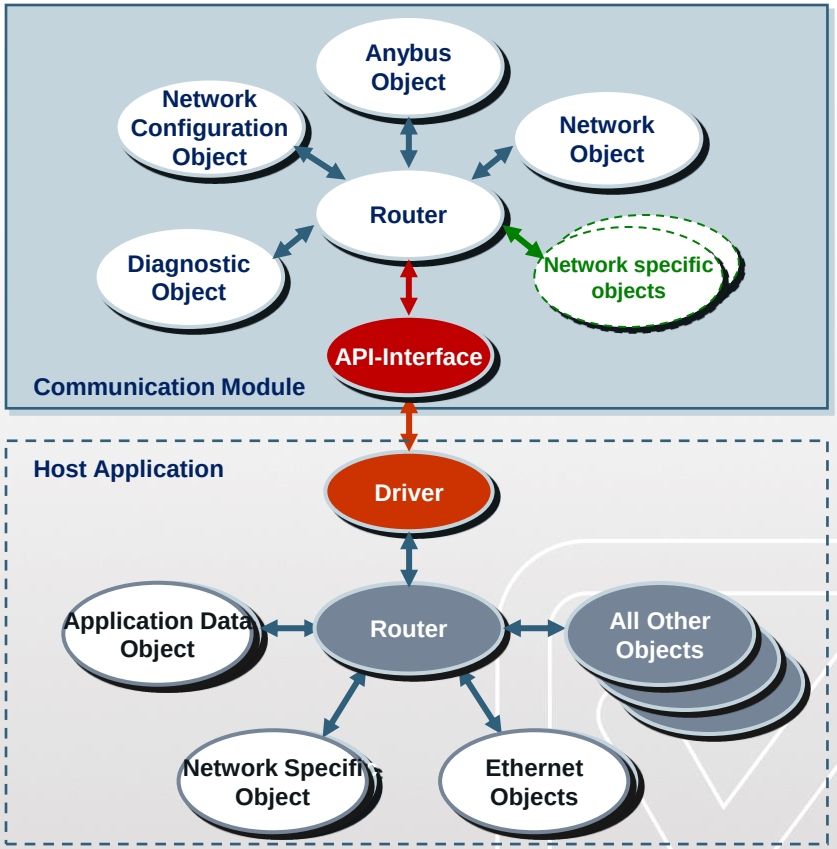
- Name, Data type, Total elements
- Value: Actual, Max, Min & Default

Network Specific Object

- Only when applicable
- Vendor ID, Product ID
- Specific Configuration Data

IT - Industry 4.0 - IIoT

- E-mail client, Web-server, FTP
- File-system & Firmware download
- TCP/IP socket, MQTT & OPC UA



Example - ADI Definition

Tower Light with 5 colors

Two Push buttons

Temperature Gaugel

ADI #100
Name: TowerLights
Array of 5 Boolean
Read

ADI #200
Name: PushButtons
Array of 2 Boolean
Write

ADI #201
Name: Temperature
Signed Integer
Write

PROFINET
Slot
Subslot
Index

PROFIBUS
Slot
Index

EtherNet/IP
Class
Instance
Attribute

Modbus
Coils
Inputs
Registers

EtherCAT
ETHERNET POWERLINK
CANopen
PDO
SDO
(sub-)Index

ADI Definition - Host-side (asynchrone data)

```
/*-----------------------------------------------------------------------------------------------------------------------------------------------------------
** 1. Instance | 2. Name | 3. DataType | 4. NumOfElements | 5. Bits| 6. ValuePtr | 7. ValuePropPtr | 8. StructPtr | 9. GetADIPtr | 10. SetADIPtr
**-----------------------------------------------------------------------------------------------------------------------------------------------------------
*/

const AD_AdiEntryType APPL_asAdiEntryList[] =
{
{ 100, "TowerLights", ABP_BOOL,    5,  READ+MAP,  { { &fTowerLights, NULL } } NULL, NULL, &fctSetLight, },
{ 200, "PushButtons", ABP_BOOL,    2,  WRITE+MAP, { { &fPushButtons, NULL } } NULL, NULL, NULL,         },
{ 201, "Temperature", ABP_SINT16,  1,  WRITE+MAP, { { &iTemperature, NULL } } NULL, NULL, NULL          }
};
```

#	Member	Description
1	Instance	ADI Instance number (1-65536)
2	Name	ADI Name in ASCII
3	DataType	ADI Data Type (Bool, Byte, Word, Quad, Char, (sign/unsigned wgen relevant)
4	NumOfElements	Number of elements in an array
5	Bits	Entry desription; Read, Write, Mappable (may be ored)
6	ValuePtr	Pointer to local value variable
7	ValuePropPtr	Pointer to local value properties (Max, Min & Default)
8	StructPtr	Pointer to structure in case of self-defined Data Type
9	GetADIPtr	Pointer to a service routine called when getting the ADI
10	SetADIPtr	Pointer to a service routine called when setting the ADI

ADI Definition - Network-side (real-time cyclic data)

```
/*-----------------------------------------------------------------------------
** 1. Instance | 2. Direction | 3. NumofElements | 4. Start index |
**-----------------------------------------------------------------------------
*/
const AD_DefaultMapType APPL_asAdObjDefaultMap[] =
{
{ 100, PD_READ,   AD_DEFAULT_MAP_ALL_ELEM, 0 },
{ 200, PD_WRITE,  AD_DEFAULT_MAP_ALL_ELEM, 0 },
{ AD_DEFAULT_MAP_END_ENTRY }
};
```

#	Member	Description
1	`Instance`	ADI Instance number (1-65536)
2	`Direction`	Directory of map
3	`NumOfElements`	Number of elements to map, `AD_DEFAULT_MAP_ALL_ELEM` = all available elements
4	`Start Index`	Element start index within an array or structure, when #3 is not default

Outsourcing Projectwise

Complete and reliable solution

- Common hard- & software design
- Do support all major protocol functionalities
- Fully compatible with all PLC’s and master controllers
- Multi network-protocol support within one development

Project

- Focus on information exchange instead of data communication
- Reduce development time up to 70% when carefully planned
- Planning all desired protocols ahead
- Fast design-in but total lead-time depending on certification

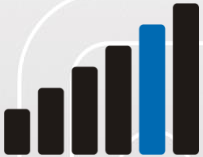
Outsourcing Businesswise

Business case

- Mix of protocols, fieldbusses and IE
- Depending on total-cost-of-development
- Internal-costs (or cost places) versus external cost
- Business policies

Optimal volumes

- Based on general experience
- Modules: Appr, volume 50 - 1500 units annually
  Bricks: Appr. volume 100 - 2000 units annually



Anybus CompactCom

Over 20 Fieldbus & Industrial Ethernet protocols covered

Open-frame modules

PC Interface

Brick

Thanks for your attention!

Twincomm
de Olieslager 44
5506 EV Veldhoven
the Netherlands

T +31-40-2301.922
E welcome@twincomm.nl
I www.twincomm.nl