Nebras

CyberSecurityExpert

OT Cybersecurity

Fundamentals

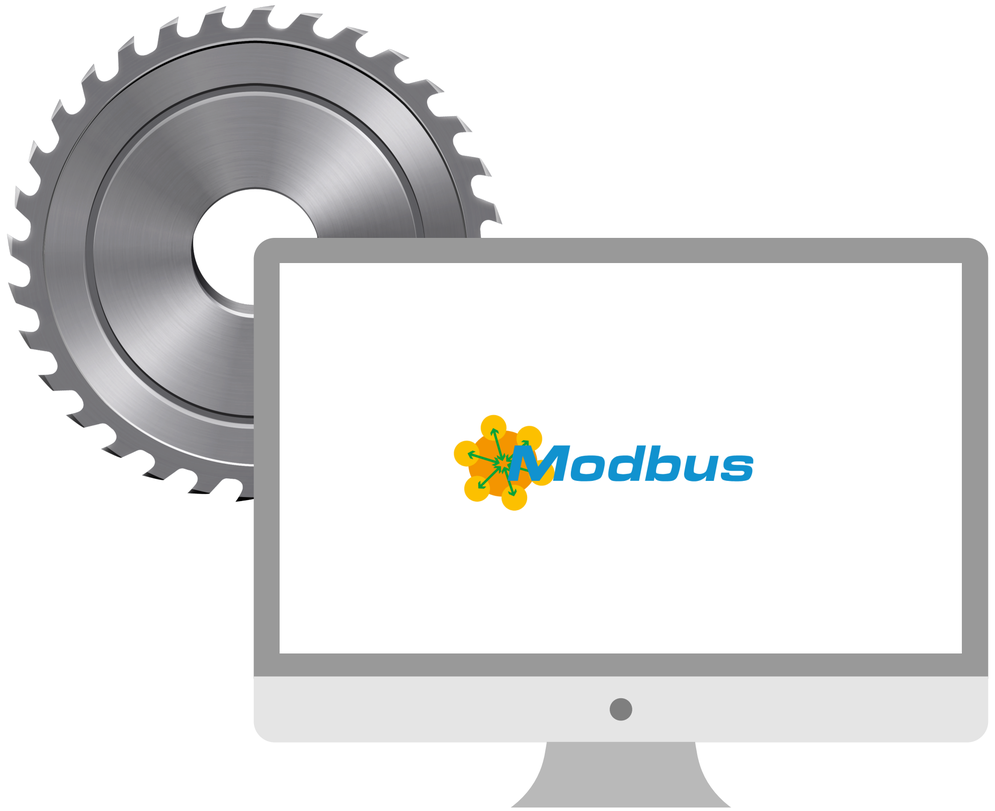
Nebras
CyberSecurityExpert

Introduction to OT

OT Cybersecurity Fundamentals

OT Communications & Protocols

What are OT protocols?
Modbus Protocol Technical Analysis
IETF RFC
Exercise 1: Modsim - Modscan
Exercise 2: Wireshark Analysis
Exercise 3: HMI using Winlog lite



Nebras
CyberSecurityExpert

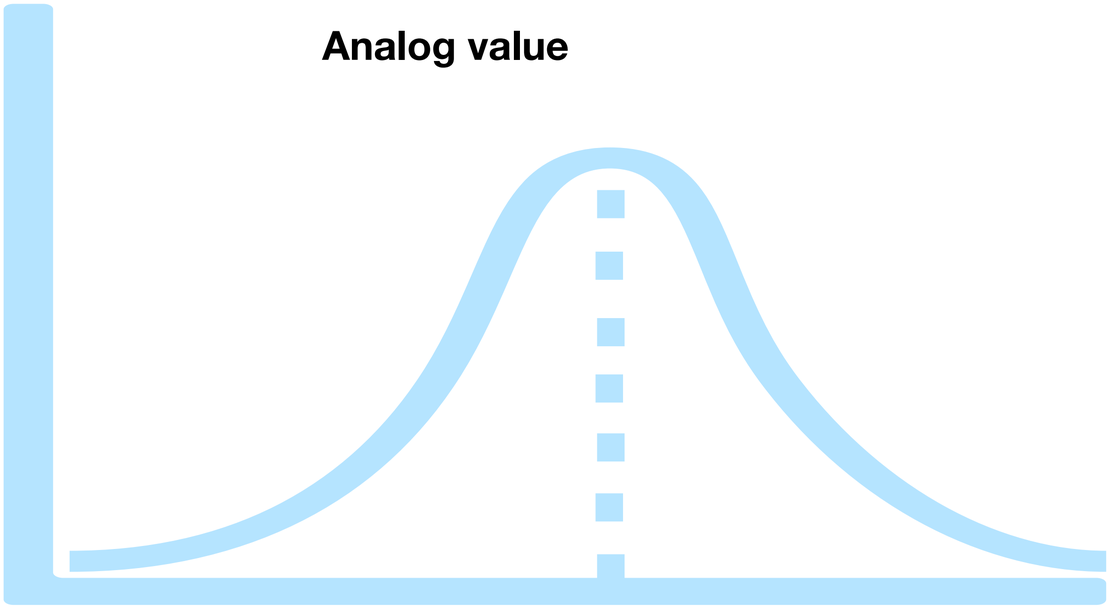
Recap on Sensors

To understand the OT communications protocols we need to understand what types of signals it carries.

Digital value

Binary output signal in the form of a logic “1” or a logic “0”, (“ON” or “OFF”)

Analog value

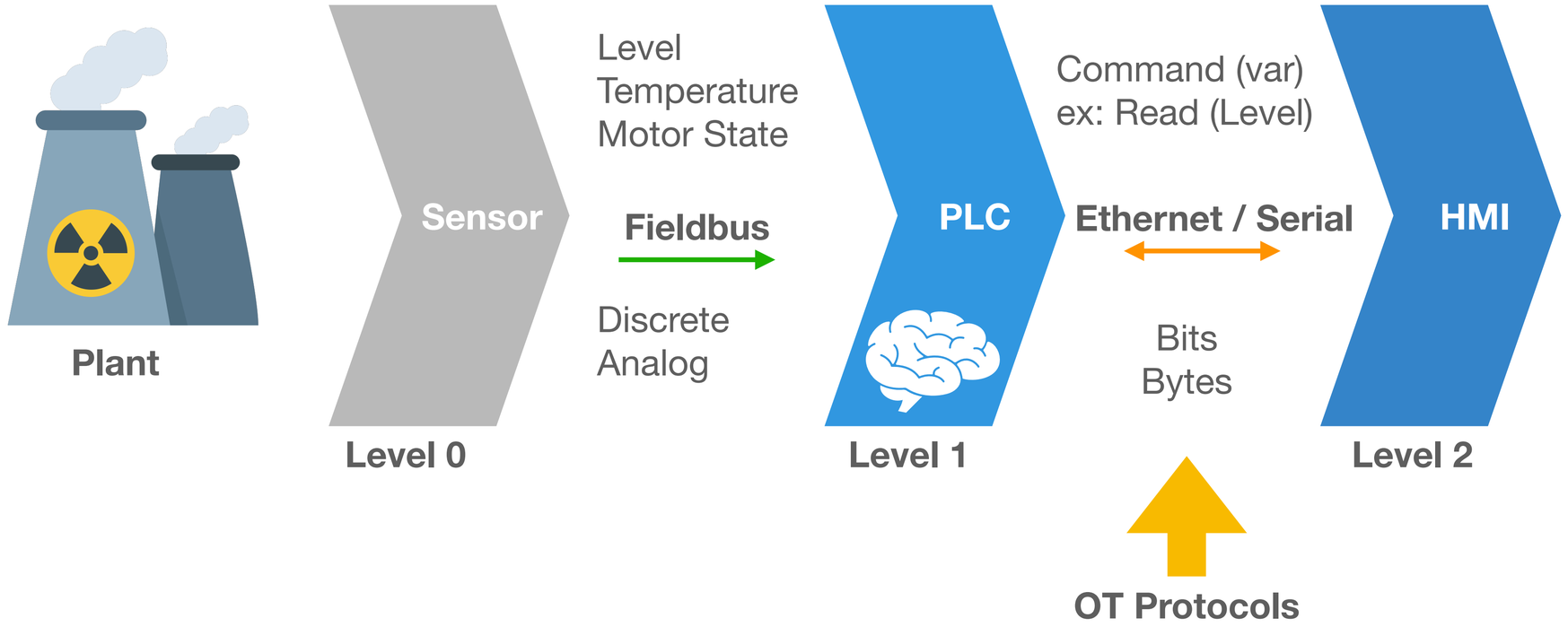


Produces a continuous output signal or voltage .

Nebras
CyberSecurityExpert

Recap on Supervise and Control

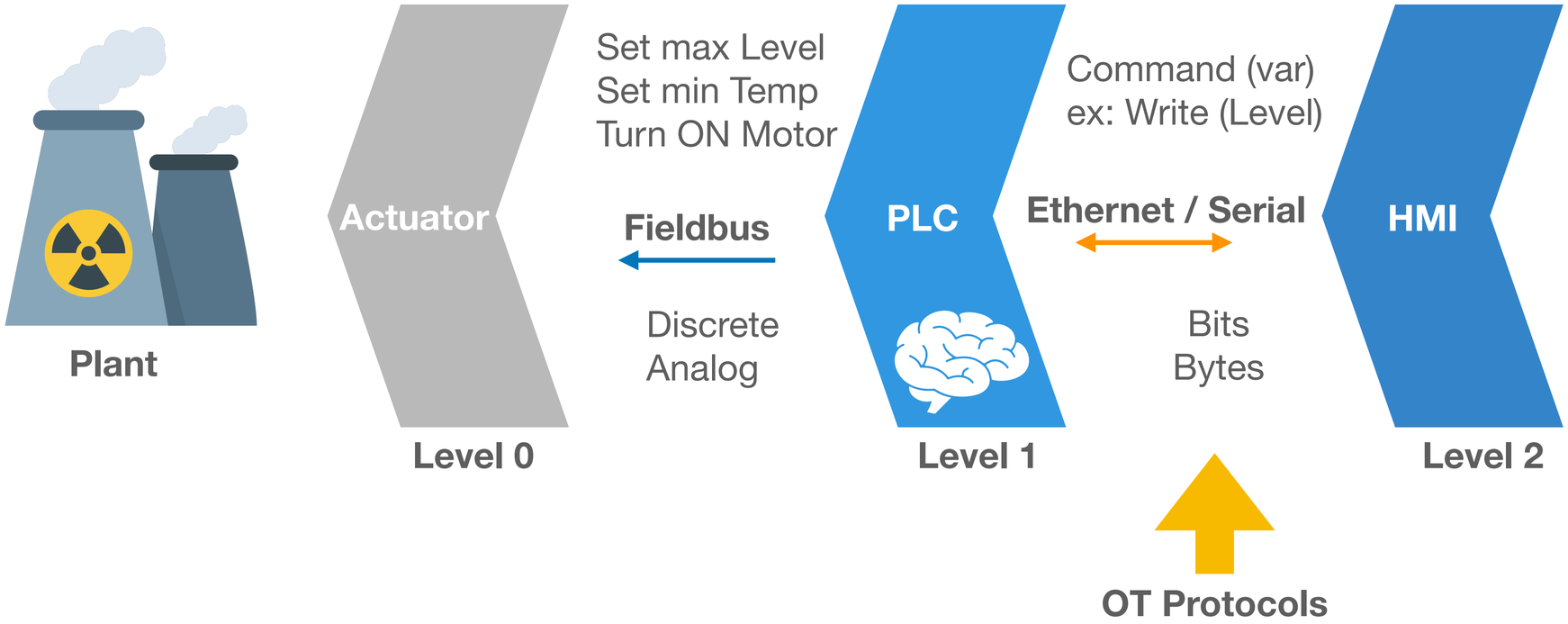
Supervise: Read Values coming from Input devices either discrete or analog (tags)
Control: Setpoints / control commands.



Nebras CyberSecurityExpert

Recap on Supervise and Control

Supervise: Read Values coming from Input devices either discrete or analog (tags)
Control: Setpoints / control commands.



Nebras CyberSecurityExpert

Operational Technology Protocols

Insecure by design!

Process Automation	Industrial Control Systems	Building Management Systems	Power System Automation	Automatic Meter Reading	Automobile/ Vehicle
CIP	**OPC**	BACnet	**DNP3**	IEC-61107	CAN
Modbus	OPC UA	LonTalk	IEC-60870-5	ZigBee	FlexRay
DNP3	OMG DDS	**Modbus**	**IEC-61850**	**Modbus**	LIN
EtherNet/IP	MTConnect	ZigBee	ICCP	…	…
HART	**Modbus**	…			
PROFINET					
GE SRTP					
…					

Internet Engineering Task Force (IETF) **www.ietf.org**

Responsible of the design and technical specifications of the protocols in the form of RFC (Request For Comments)

Insecure by design, lack authentication, authorization, and encryption.

Nebras CyberSecurityExpert

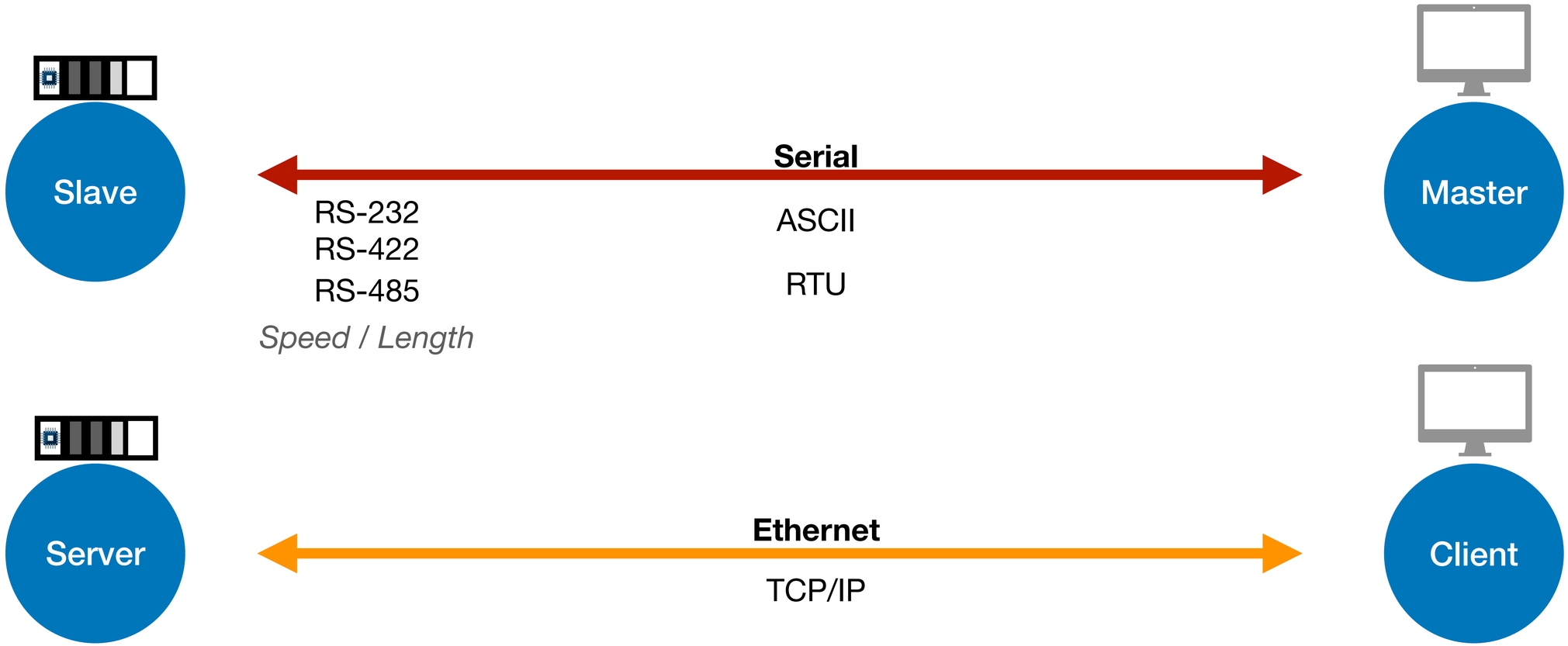
Modbus

About?	Designed in 1979 by Modicon (part of Schneider Electric) that invented the first PLC. It's an open standard, freely distributed and is widely supported by Modbus Organization.
What?	It communicates raw messages without restrictions of authentication or excessive overhead. It allows for efficient communications between assets based on request/reply methodology.
Protocol	An application layer messaging protocol (Layer 7 of OSI model). Request / Response protocol using three distinct Protocol Data Units (PDUs): (1) Modbus Request (2) Modbus Response (3) Modbus Exception Response

Schneider Electric MODICON

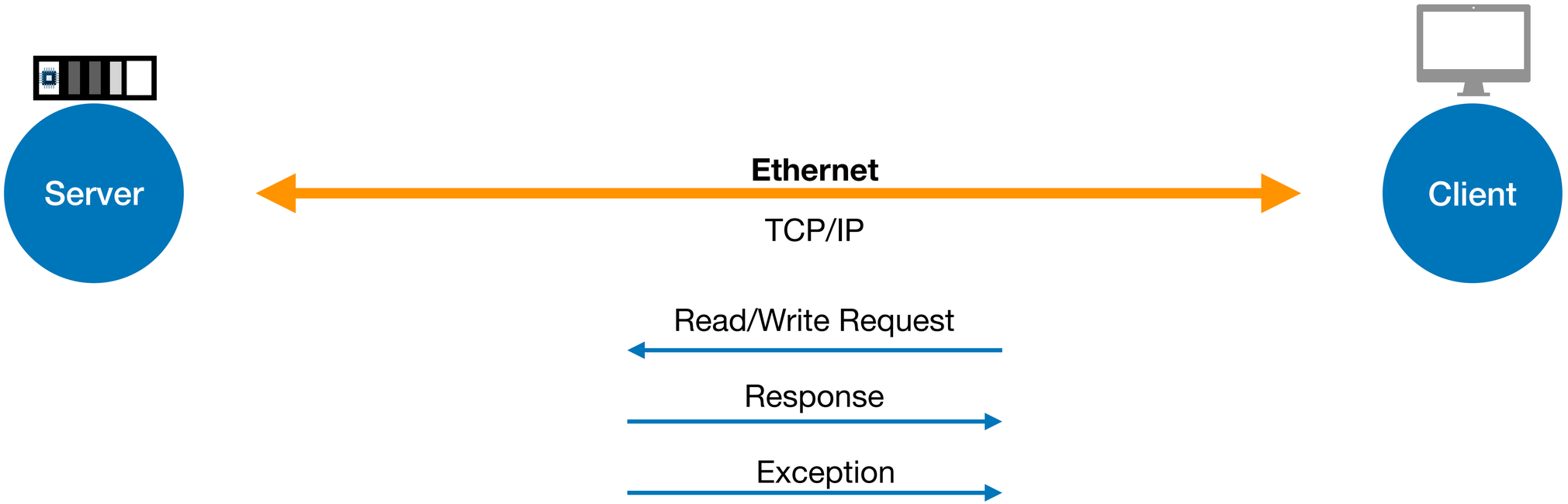
Nebras CyberSecurityExpert

Modbus Communications



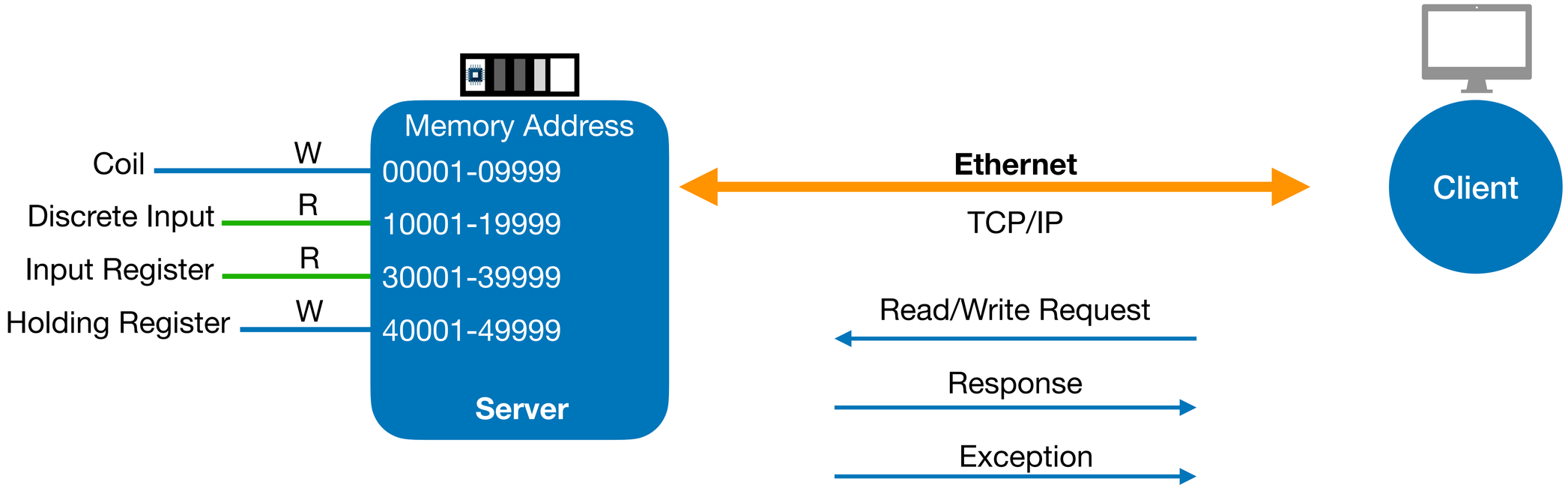
Nebras
CyberSecurityExpert

Modbus TCP Communications



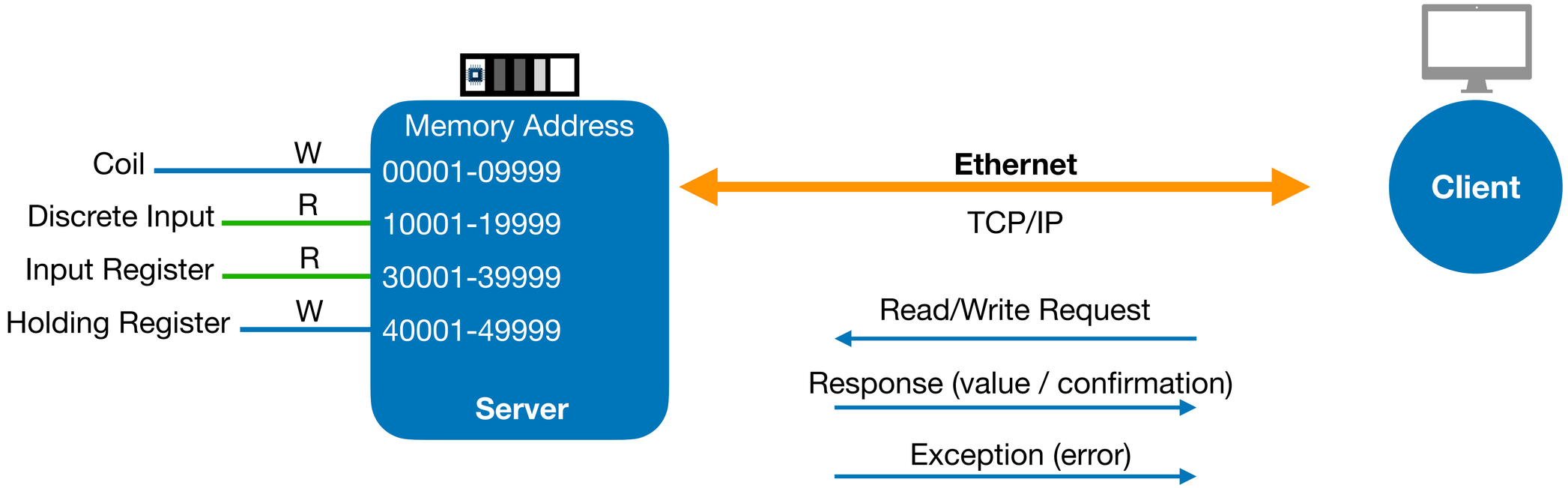
Nebras
CyberSecurityExpert

Modbus TCP Communications



Nebras
CyberSecurityExpert

Modbus TCP Communications



Discrete value (0 or 1)

Coil (Read / Write)

Discrete Input (Read only)

Registers (Numeric 16 bits)

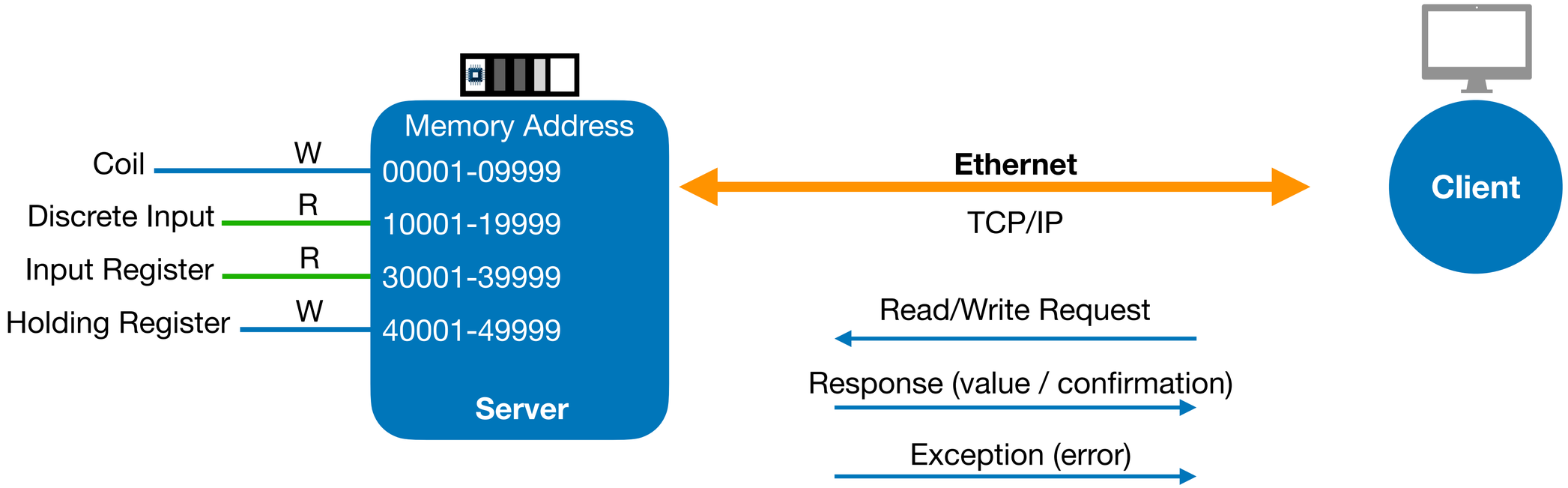
Input Register (Read only)

Holding Register (Read / Write)

9999 values for each type

Nebras CyberSecurityExpert

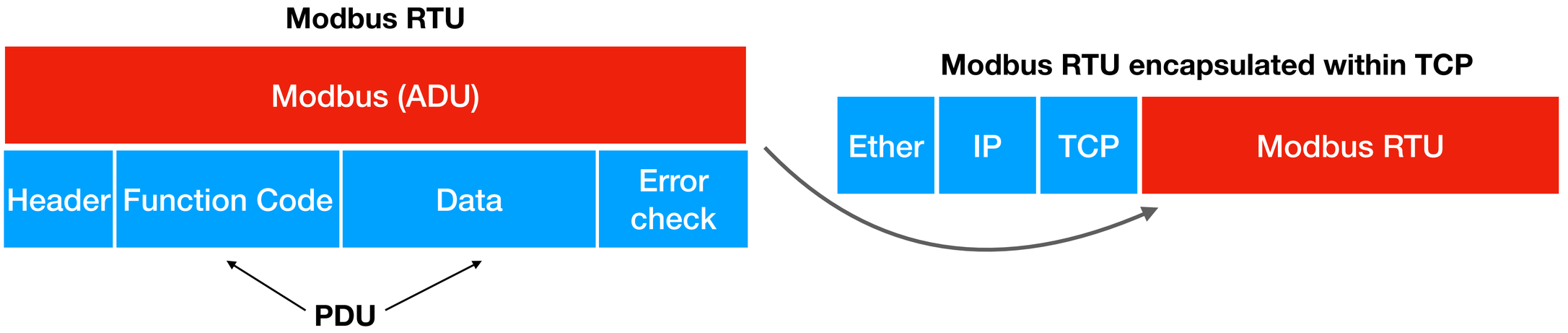
Modbus TCP Communications



Unit Identifier: Each Server has unique ID
Function Code: Each operation represented by specific code
Memory Address: Code (read / write) to which address?

Nebras CyberSecurityExpert

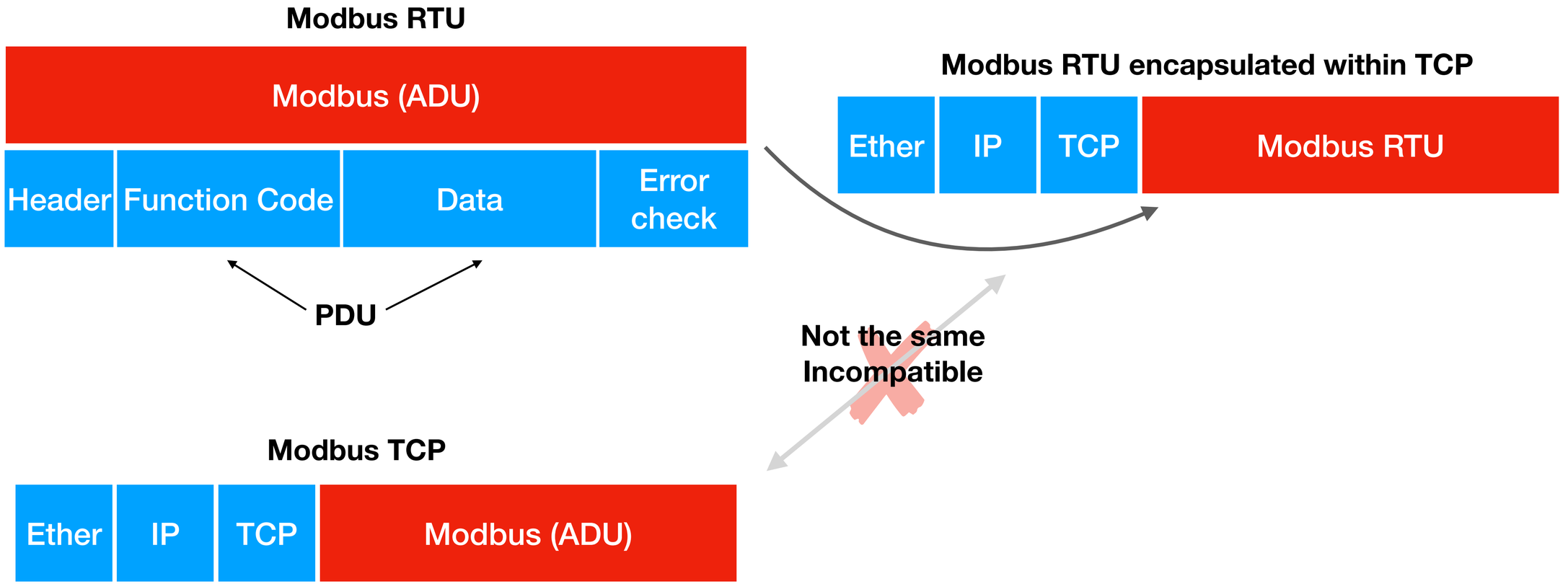
Modbus TCP Communications



ADU: Application Data Unit
PDU: Protocol Data Unit

Nebras
CyberSecurityExpert

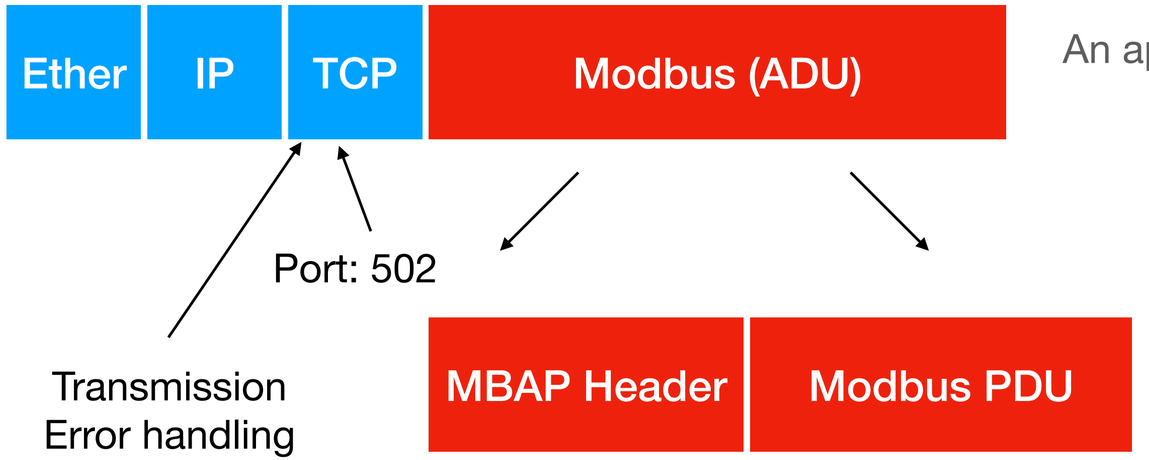
Modbus TCP Communications



ADU: Application Data Unit
PDU: Protocol Data Unit

Nebras
CyberSecurityExpert

Modbus TCP Communications



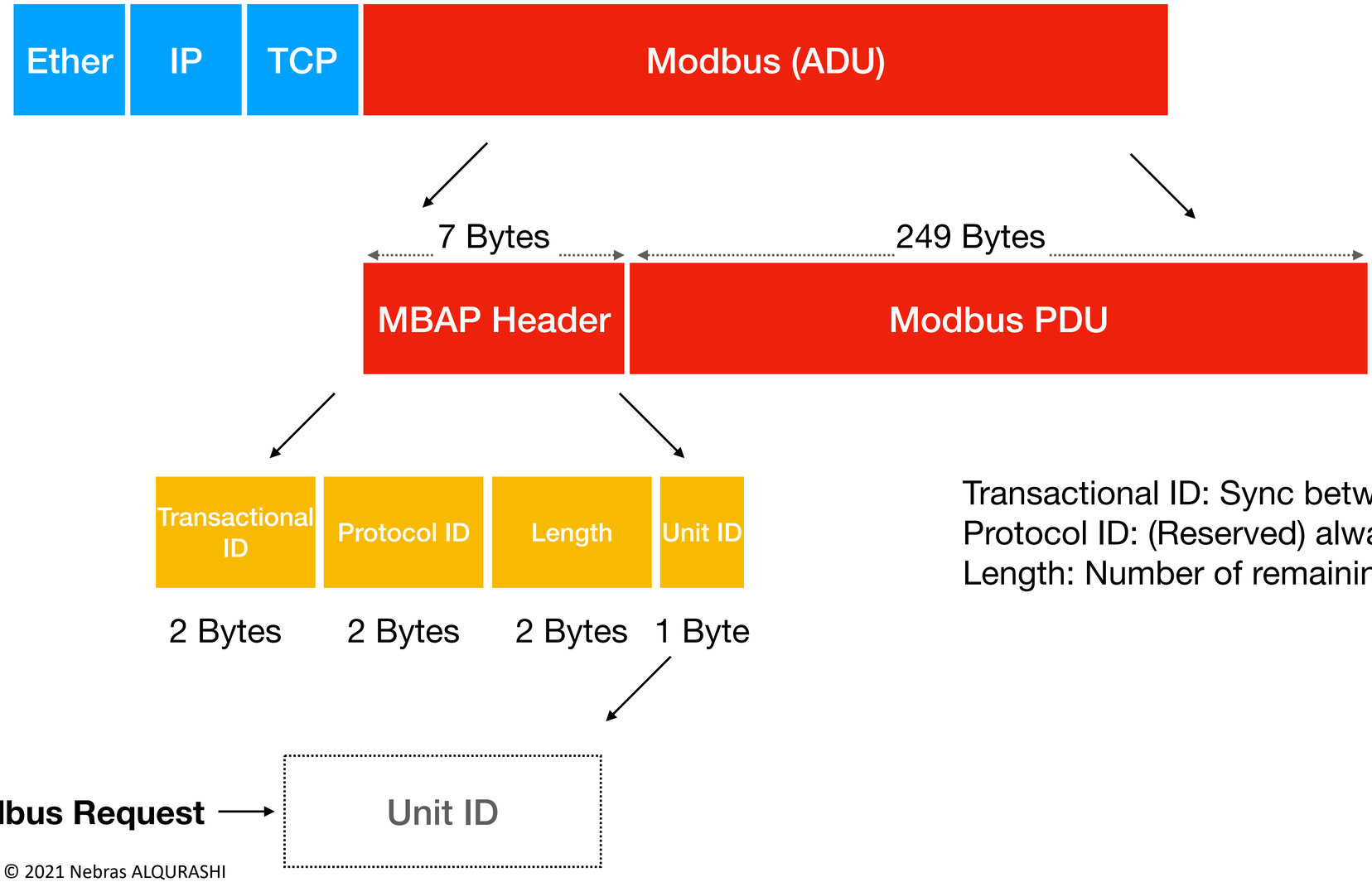
An application layer messaging protocol (Layer 7 of OSI model).

ADU: Application Data Unit
MBAP: Modbus Application Protocol
PDU: Protocol Data Unit

Nebras
CyberSecurityExpert

Modbus TCP Communications

Modbus TCP Request

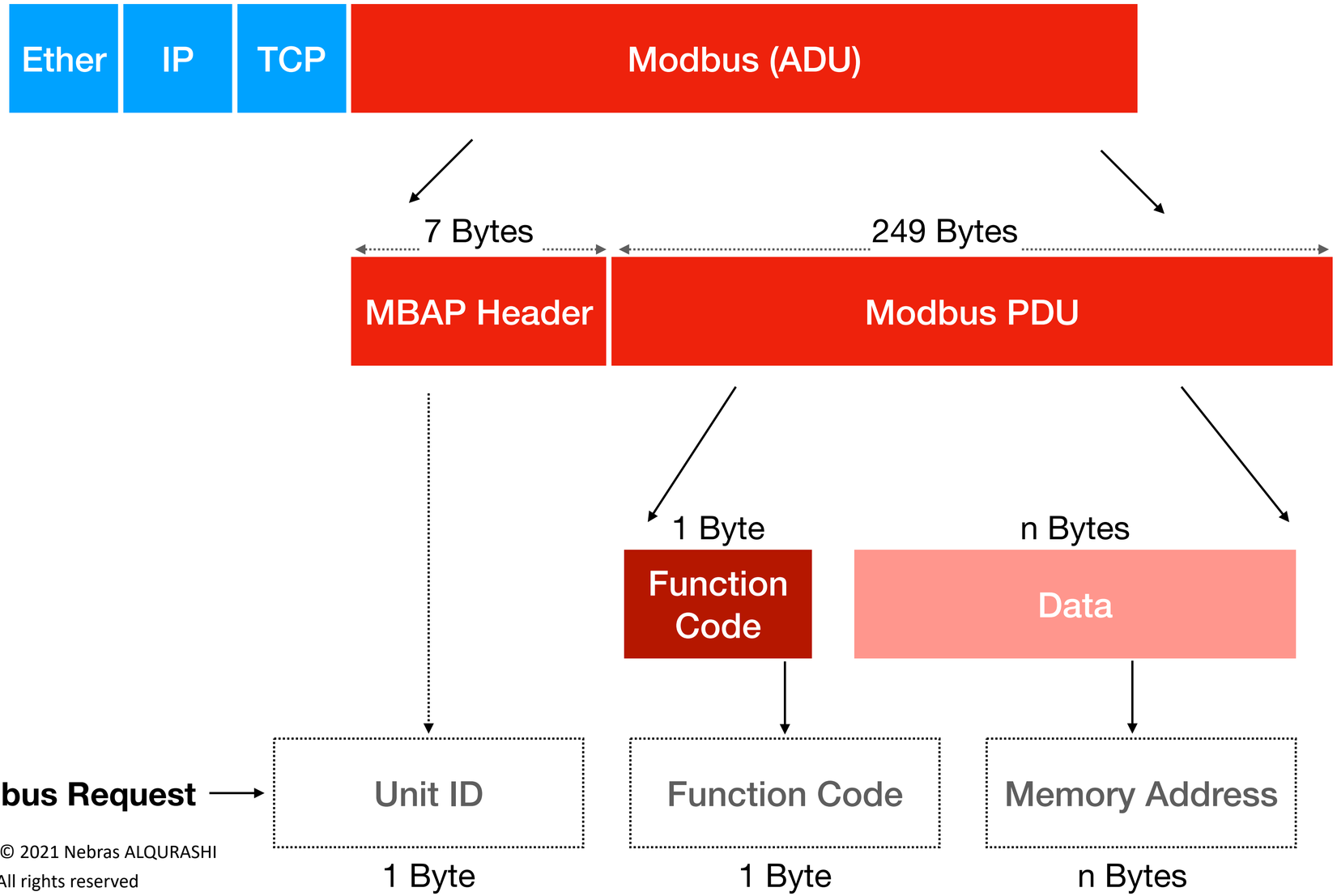


Transactional ID: Sync between messages (request/response)
Protocol ID: (Reserved) always 00 00 for Modbus
Length: Number of remaining Bytes in this frame

Nebras CyberSecurityExpert

Modbus TCP Communications

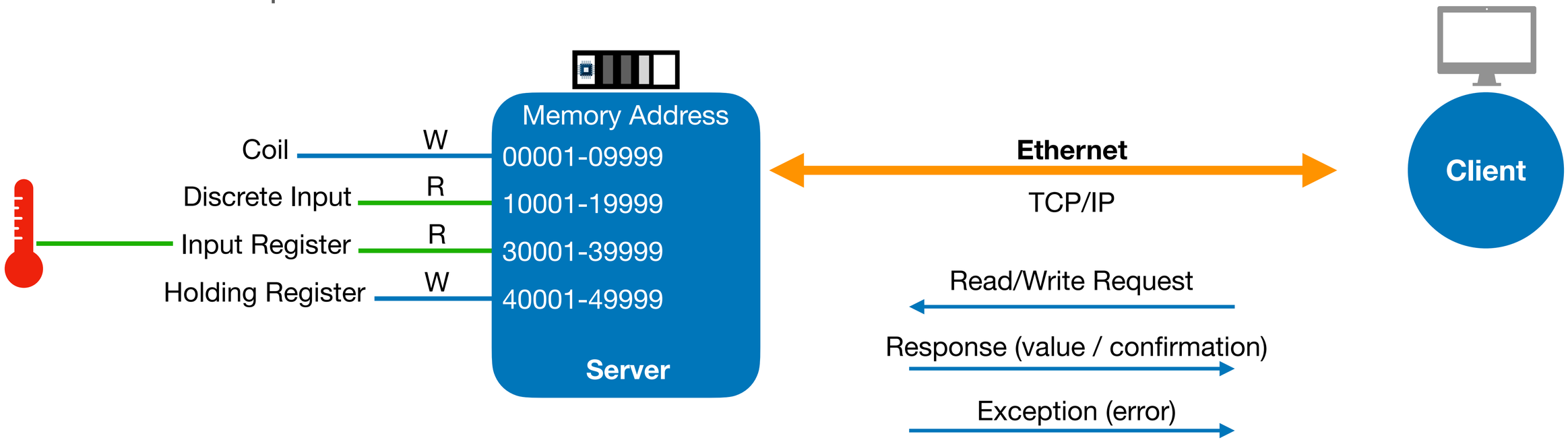
Modbus TCP Request



Nebras CyberSecurityExpert

Modbus TCP Communications

Modbus TCP Request



Example: Read Temperature at address 100 from PLC 10

Unit ID: (10) - Function Code: (Read Input Register) - Address (30100)*

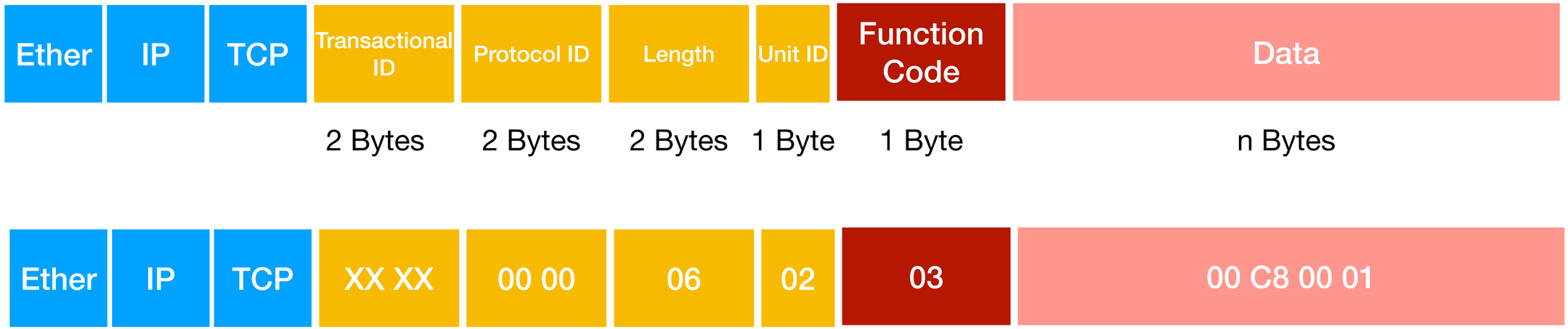
In Zero based addressing: (30099)*

** We will see that address can be:*
Start Address and length

Nebras CyberSecurityExpert

Modbus TCP Communications

Modbus TCP Request



Transactional ID	XX XX (Hex Reference)
Protocol ID	For Modbus always 00 00
Length	Remaining bytes after the length 0x6
Unit ID	To PLC of Unit ID 0x02
Function Code	0x03 (Read Holding Registers)
Memory Address	Start at: Hex (00C8) = 200 for a length of 0x1

Read Holding Register single value from the memory address 200 from PLC ID: 02

Nebras CyberSecurityExpert

Modbus TCP Communications

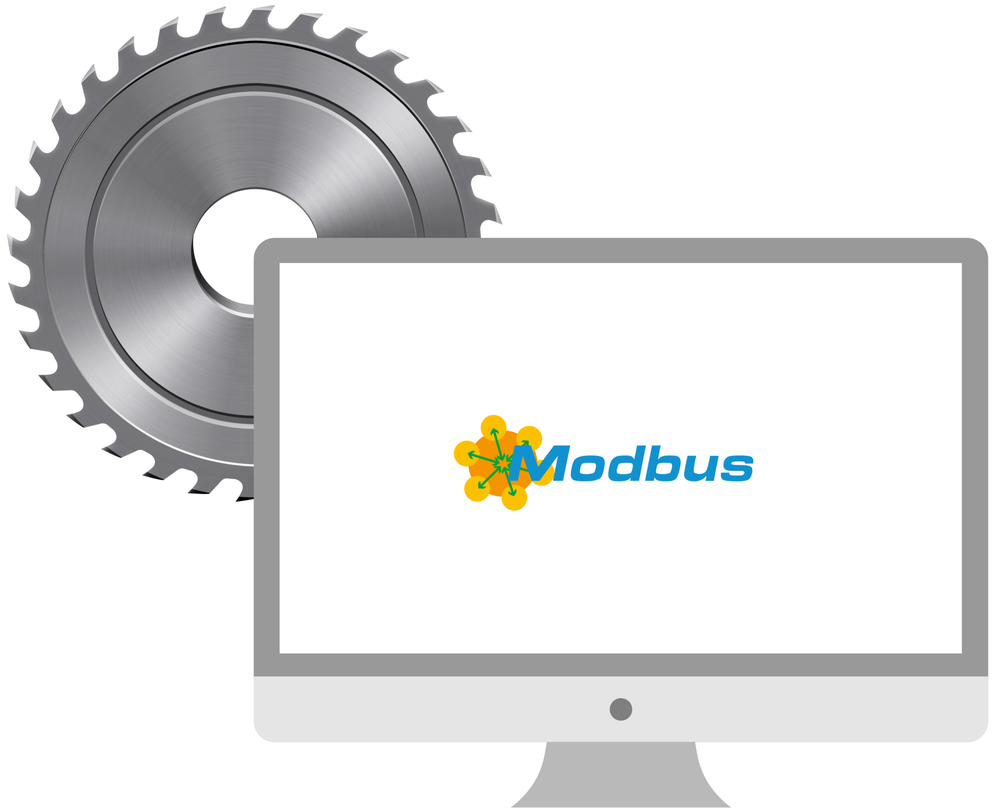
Common Modbus Function Codes

Value Type	Function Name	Function Code	Hex
Bits (0 / 1)	Read Discrete Inputs	2	0x02
Bits (0 / 1)	Read Coils	1	0x01
Bits (0 / 1)	Write Single Coil	5	0x05
Bits (0 / 1)	Write Multiple Coils	15	0x0F
Words (16 bit)	Read Input Register	4	0x04
Words (16 bit)	Read Holding Registers	3	0x03
Words (16 bit)	Write Single Register	6	0x06
Words (16 bit)	Write Multiple Registers	16	0x10

Nebras
CyberSecurityExpert

Quick Summary

What are OT protocols?
Modbus Protocol Technical Analysis
IETF RFC
Exercise 1: Modsim - Modscan
Exercise 2: Wireshark Analysis
Exercise 3: HMI using Winlog lite



Nebras
CyberSecurityExpert