# OPC Server for W-IE-NE-R devices interfaced to CAN-Bus

**Zbigniew Hajduk, Jolanta Olszowska**

Institute of Nuclear Physics

Kraków

# Visit at the W-IE-NE-R Company

- Company visited in the end of July

  - ✓ **We have presented our needs in terms of integration W-IE-NE-R CAN-controlled devices into ATLAS DCS systems**
  - ✓ **Andreas Koester from W-IE-NE-R presented their CAN interfaces protocol and future plans**
  - ✓ **We have presented the proposal of writing the OPC Server dedicated to W-IE-NE-R CAN controlled devices as an undergraduate student diploma work supervised by J.Olszowska**
  - ✓ **Company agreed to send us one LV Power Supply with CANbus controller for the time of development**

# W-IE-NE-R PS's remote controlling options

- PL 500 Power Supplies Series
  - ✓ Manual Control Board with Display
  - ✓ RS 232 – mainly for service purposes (no complete manual description for protocol)
  - ✓ CANbus Interface – with Wiener specific Level-2 protocol
- PL 600 Power Supplies Series
  - ✓ Manual Control Board with Display
  - ✓ RS 232 (as above)
  - ✓ CANbus Interface (as above)
  - ✓ TCP/IP
  - ✓ WWW Interface

# W-IE-NE-R crates remote controlling options

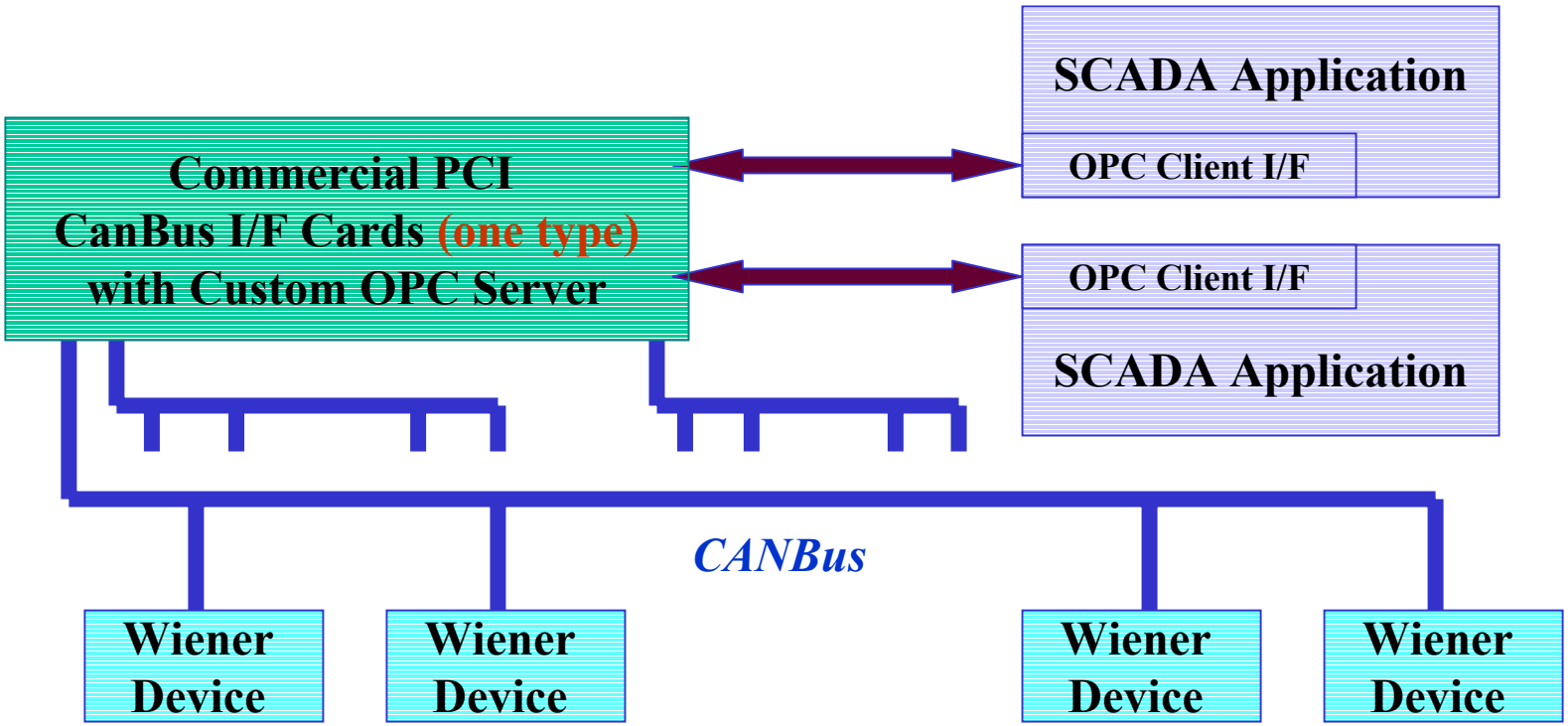
- NIM, CAMAC or VME crates

  ✓ Voltages, Currents, Fans speeds, Temperatures can be controlled by CANbus interface with Wiener specific Level-2 protocol

***The Level-2 CANbus protocol is the same for all of above W-IE-NE-R devices***

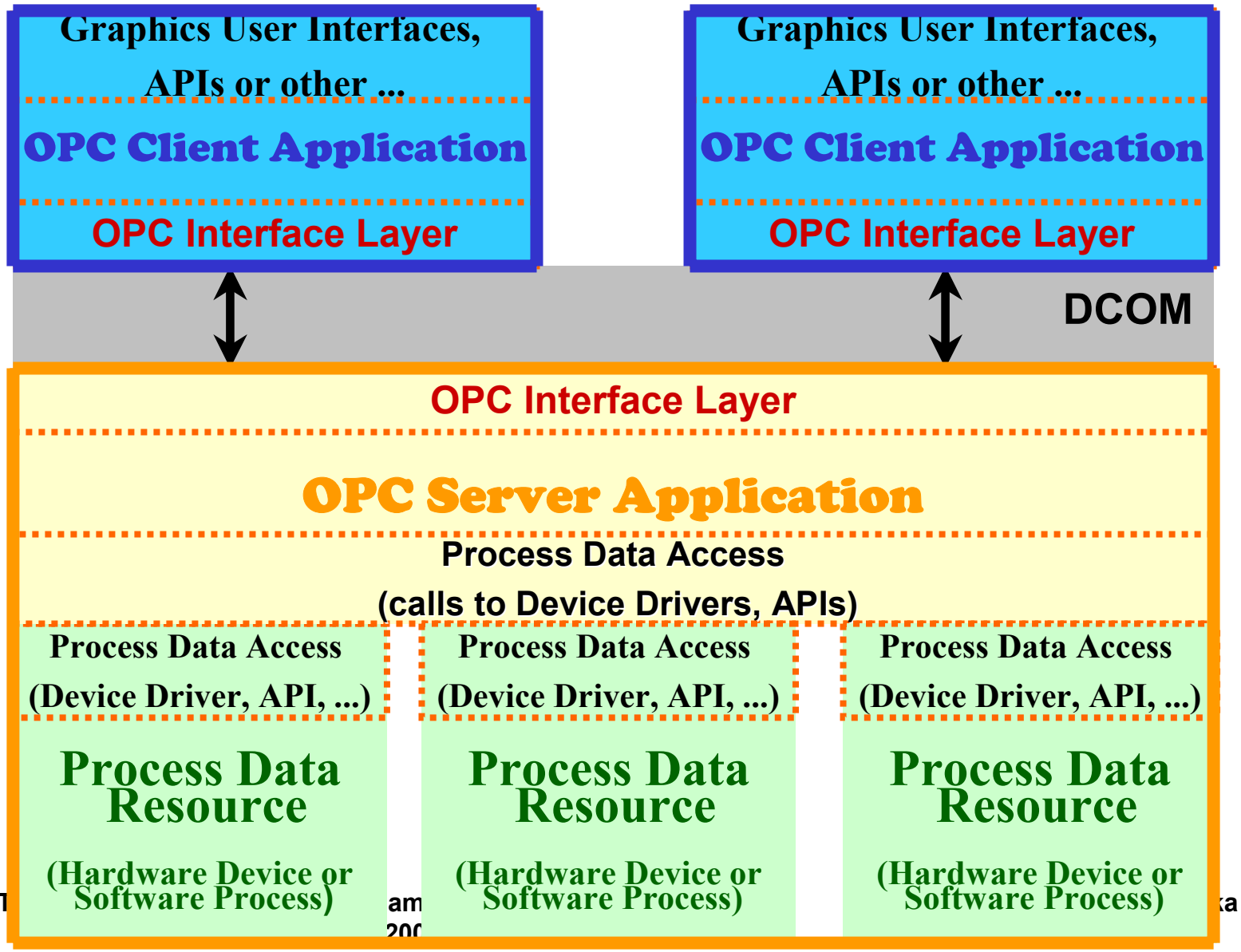
***CanOpen implementation is not forseen in the near future***

# Integration of W-IE-NE-R CANbus devices to SCADA

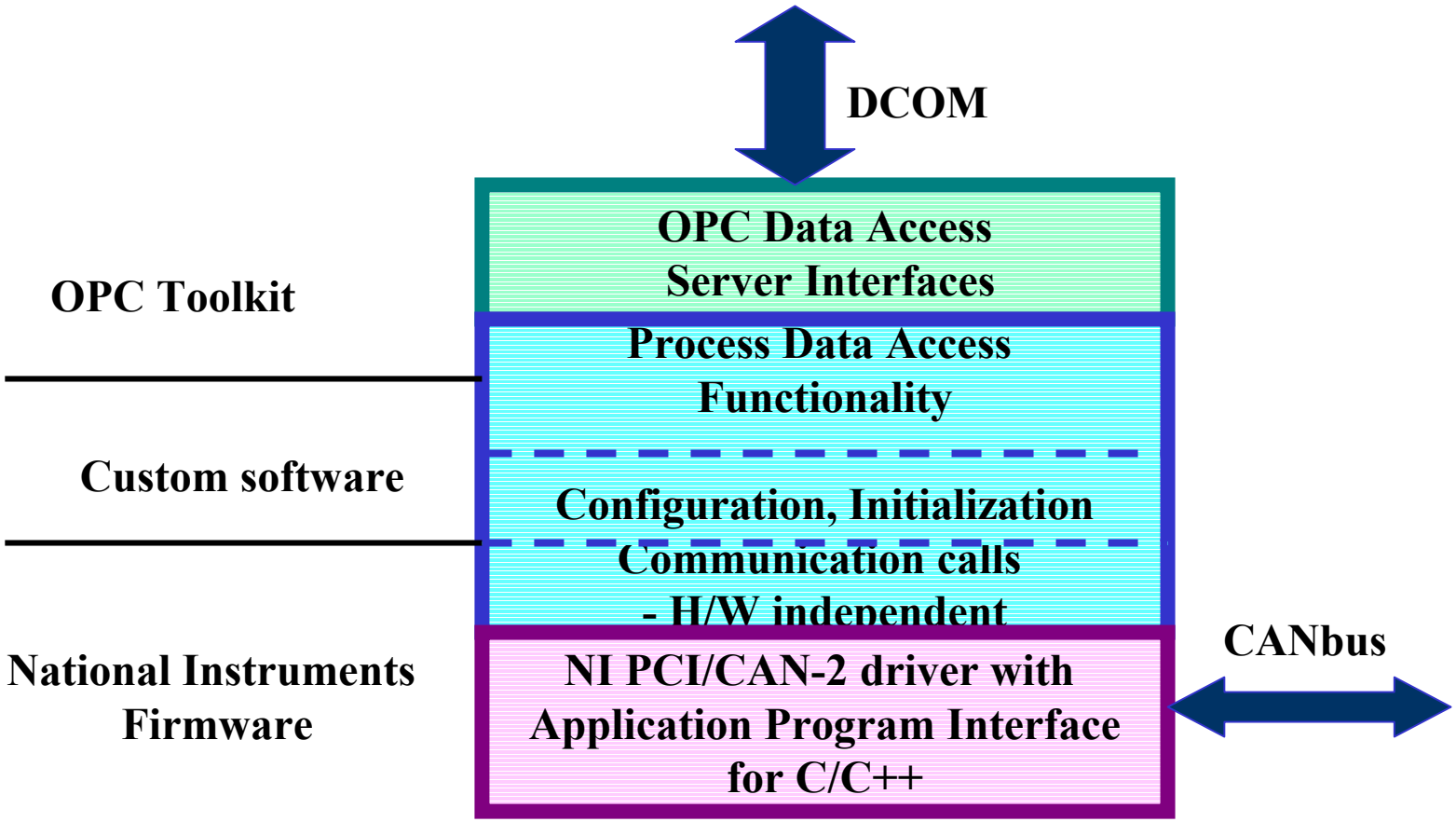



*"Wiener protocol" devices only*

# OPC Server – general layers



# Wiener OPC Server



# Custom OPC Server - work to do

- Process Data Namespace defining
  - ✓ Names, data types, attributes, tree structure ...
- Process Data to CAN Messages mapping
- H/W independent communication calls defining
  - ✓ Library for specific PCI/CAN interface developing (NI-PCI/CAN now)
- Initialization of drivers and other h/w initializations
- CAN networks configuration
  - ✓ Configuration application, editable configuration file, self-configuration option, devices ID strings setting
- Process Data access queues, events handling
  - ✓ Hardware resources access optimization
- Software "heartbeat" implementation

# Project status

- Project started
  - ✓ All hardware components and software tools needed to start the project completed
  - ✓ Dariusz Antonczyk started from getting familiar with hardware and software tools and code examples
  - ✓ He started with writing the configuration tool for the OPC Server and the set of calls for configuration and communication with NI-CAN PCI interface driver

**BUT**

  - ✓ We are not quite satisfied in progress in that project (partially due to holidays, and another student's duties)

**So**

  - ✓ We decided to involve postgraduate student to work in parallel to guarantee timely finish of work (early summer next year)