



Implementing a FORTE network Layer

LEADING
INNOVATIONS

WWW.PROFACTOR.AT

Overview

- What is the problem?
- Configurable Network FBs
- Layer Design Pattern
- Configuration String
- Class Diagram
- Interface in C++
- Open/Close/Send/Receive

What is the Problem?

- IEC 61499 Compliance Profile for Feasibility Demonstrations only defines ASN.1 and TCP/UDP as network protocol
- Changing the network protocol in an IEC-61499 FB-Network
- Configuring the Network-FB with the 4DIAC-IDE (in future)
- Implementing a new Network Layer for the FORTE

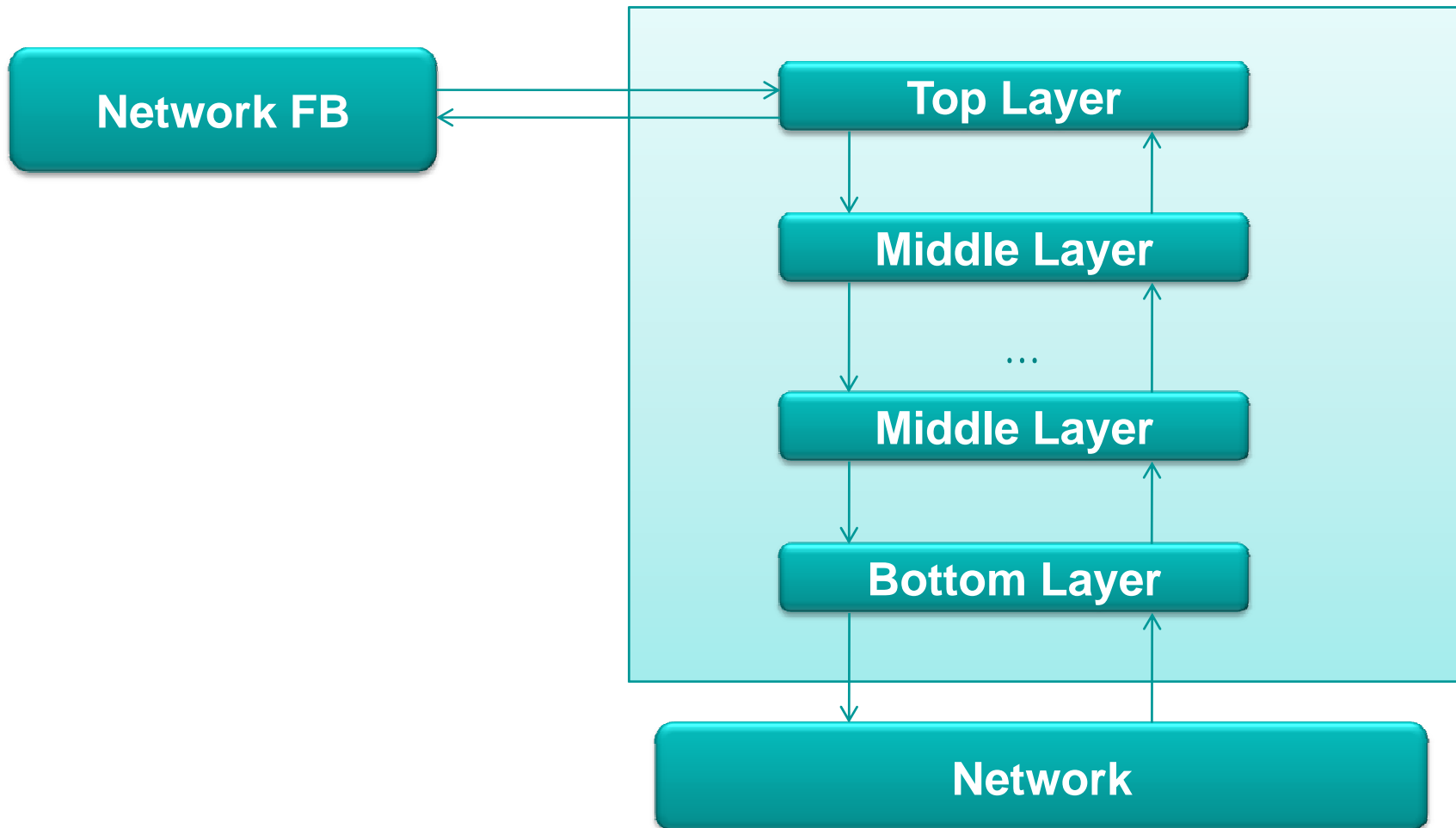
Configurable Network FBs

- Network-FBs are configured when initialized; network protocol is not associated with FB-type

- Use layer design pattern to improve reusability of protocols
 - Implement Serialization once
 - Implement Network access once
 - Implement data encryption once

- New protocols only have to implement one layer, less knowledge about FB-execution is needed

Layer Design Pattern

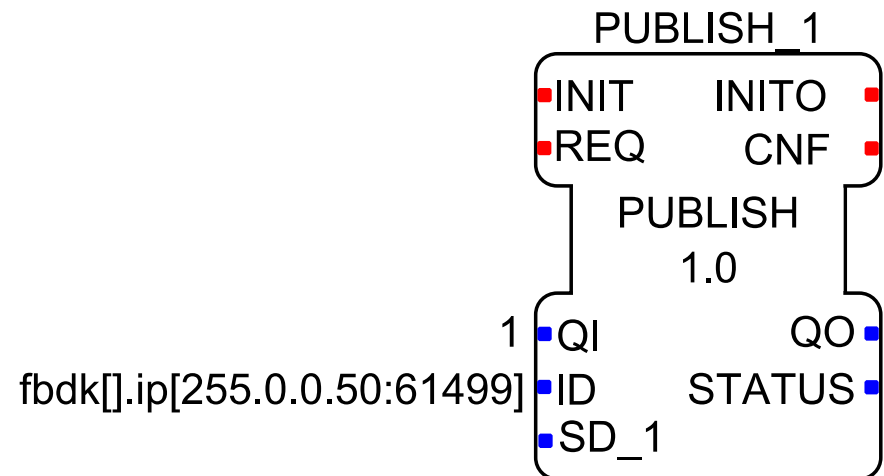


Configuration String

- Every layer is configured independently from other layers, at least one layer has to be specified
 - layer: string to identify the protocol
 - parameter: vendor specific string
 - ID: {protocol[parameter].}protocol[parameter]

Example:

Fbdk[].ip[255.0.0.50:61499]



Class Diagram (1)

➤ Function Block

- Can be called to process an interrupt
- Uses the LayerManager to create the layer Stack
- Has one or more layers which can be used to send/receive data

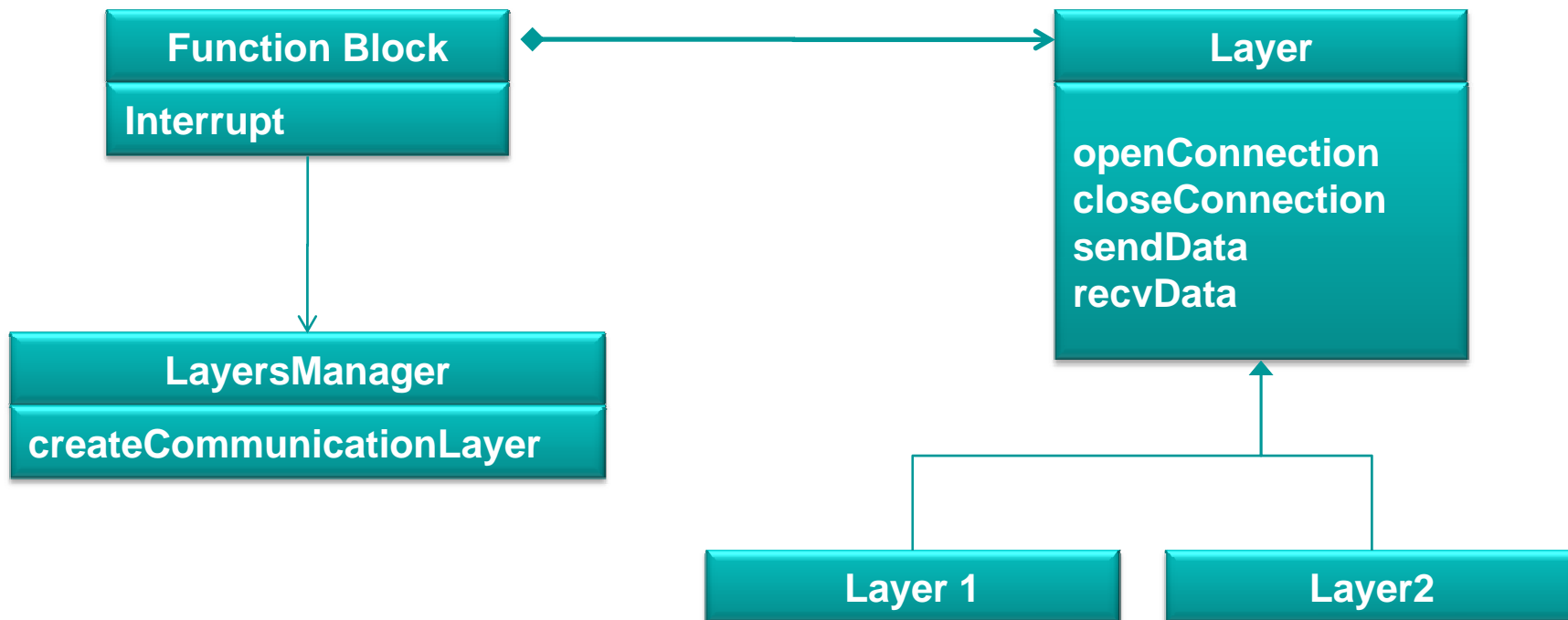
➤ LayerManager

- Knows all available Layers
- Creates Layers and initializes them

➤ Layer

- Can be opened, closed
- Can send/receive data
- Can call the FB to change from interrupt-Thread to FB-Execution-Thread

Class Diagram(2)



Interface in C++

- **EComResponse** openConnection (**char *pa_acLayerParameter**)
 - Depending on the layers functionality different things have to be performed here. This can range from doing nothing to establishing an TCP session.

- **void** closeConnection ()
 - Implementations of this function should perform the actions necessary for closing the layer and than call the closeConnection() of the bottom layer.

- **EComResponse** sendData (**void *pa_pvData, unsigned int pa_unSize**)
 - If necessary invoke bottom layer sendData functions.

- **EComResponse** rcvData (**const void *pa_pvData, unsigned int pa_unSize**)
 - This function is called for processing the received data from bottom to top. Therefore, if necessary invoke the top layer's receiveData function to hand on the processed data.

- **void** interrupt (**IComCallback *pa_poComLayer**)
 - Request process invocation from function block. When this function is called, the function block will call rcvData to process the new recieved data.

Open Connection (1)

➤ Network-FB

- gets event to open connection
- Calls Layer Manager to create and initialize all Layers

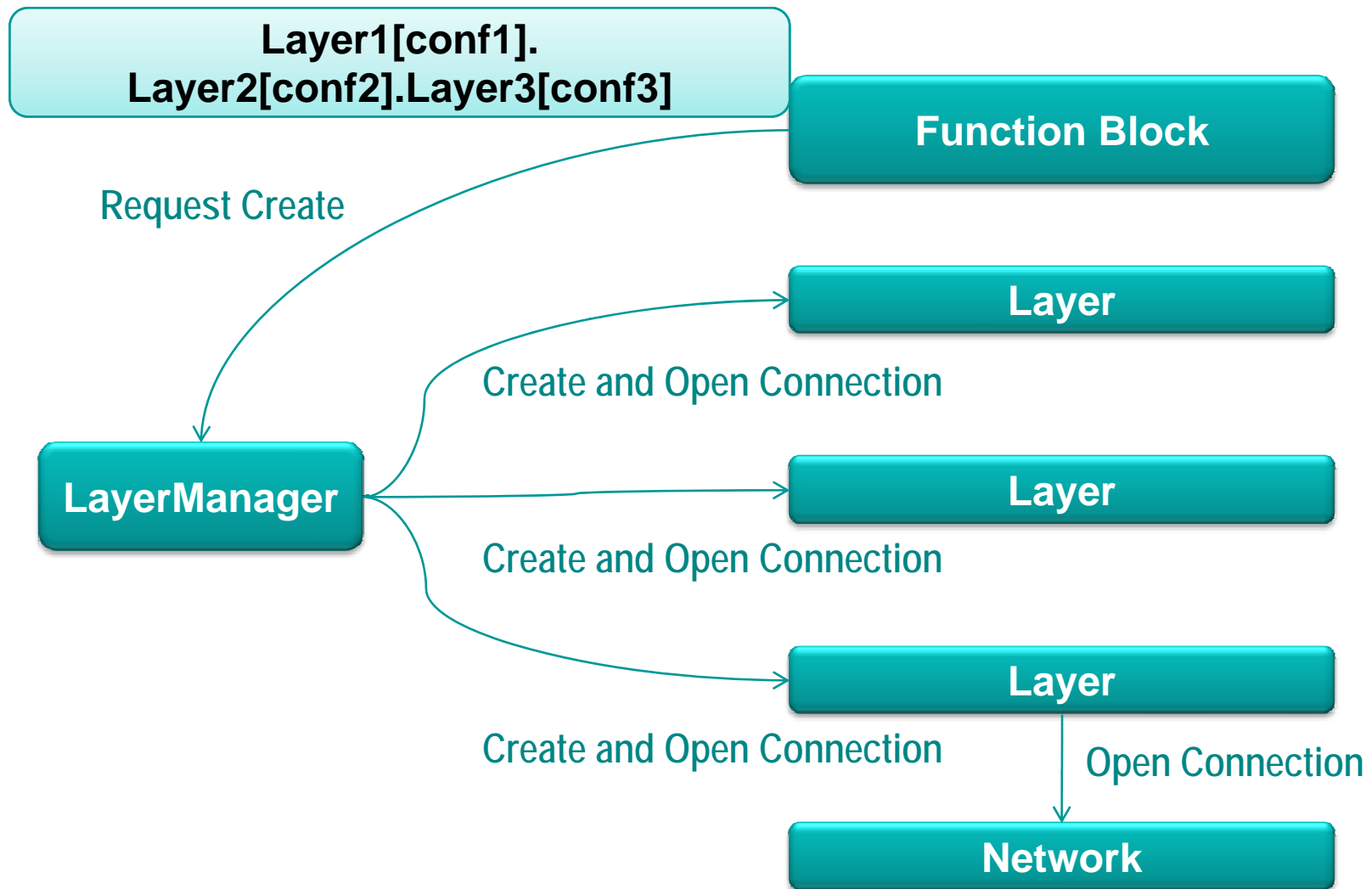
➤ Layer Manager

- Creates Layer and initializes it with the configuration string
- Goes from Top-Layer to Bottom-Layer

➤ Layer

- Does not know anything about other layers

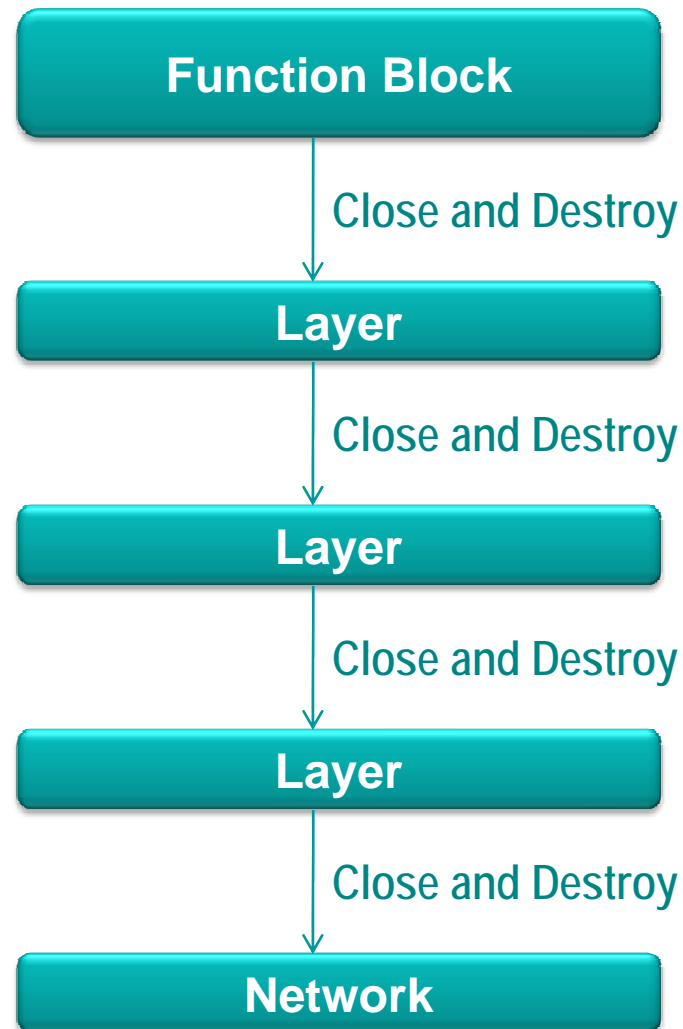
Open Connection (2)



Close Connection (1)

- Network-FB
 - gets event to close connection
 - Sends Top-Layer to close the connection
- Layer
 - Clean up own data
 - Has to close underneath Layer before or after clean up
- Bottom-Layer
 - Clean up own data
 - Has to close physical network connection

Close Connection (2)



Send Data

➤ Network-FB

- gets event to send Data
- calls Top-Layer: Layer will get a list of FB-Inputs

➤ Top-Layer

- Serializes data from FB-Inputs
- Sends data to Middle-Layer

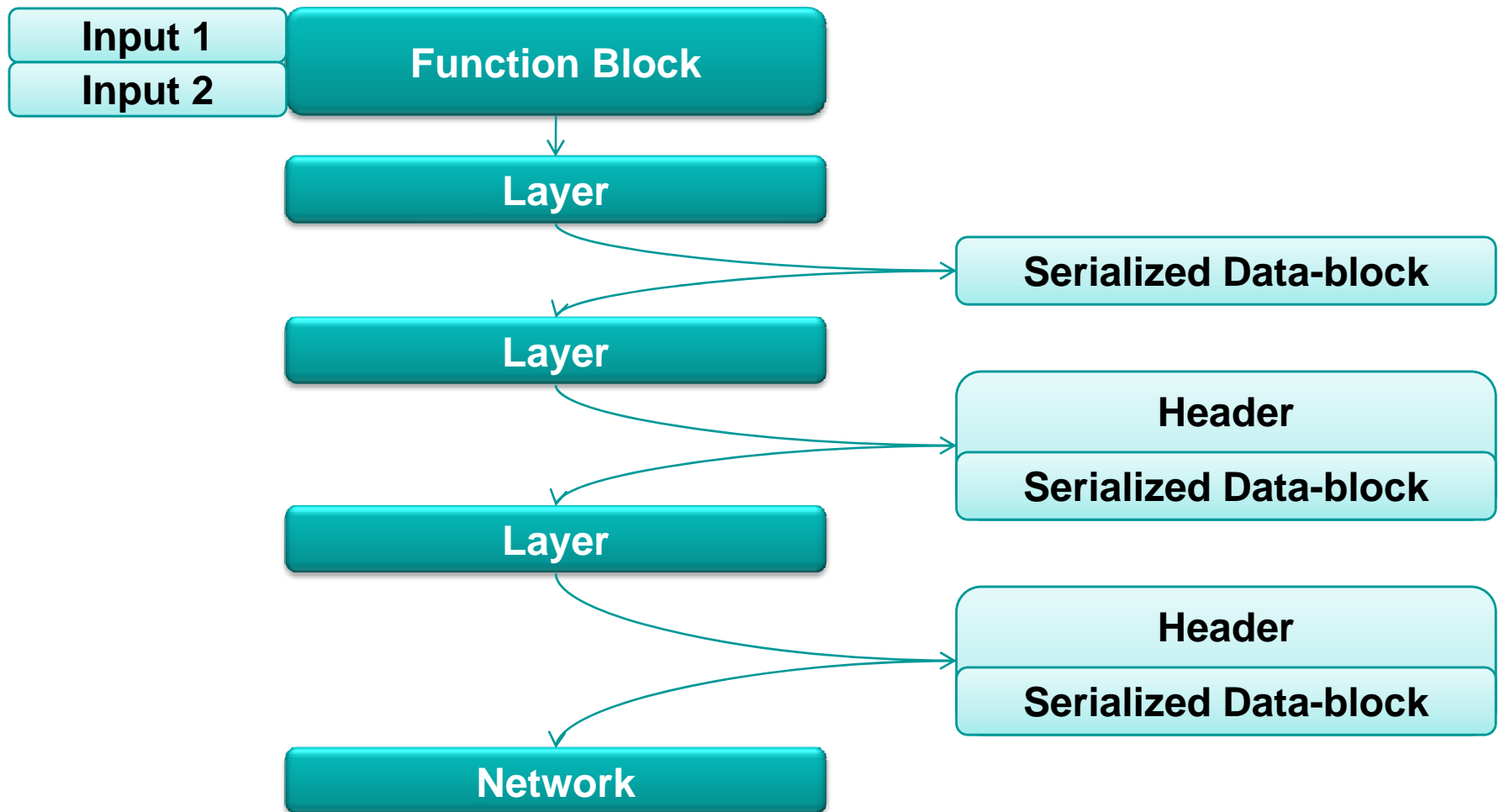
➤ Middle-Layer

- Can process data: e.g. encryption, compression
- Can add a header to data
- Sends data to Middle-Layer or Bottom-Layer

➤ Bottom-Layer

- Sends data to physical network

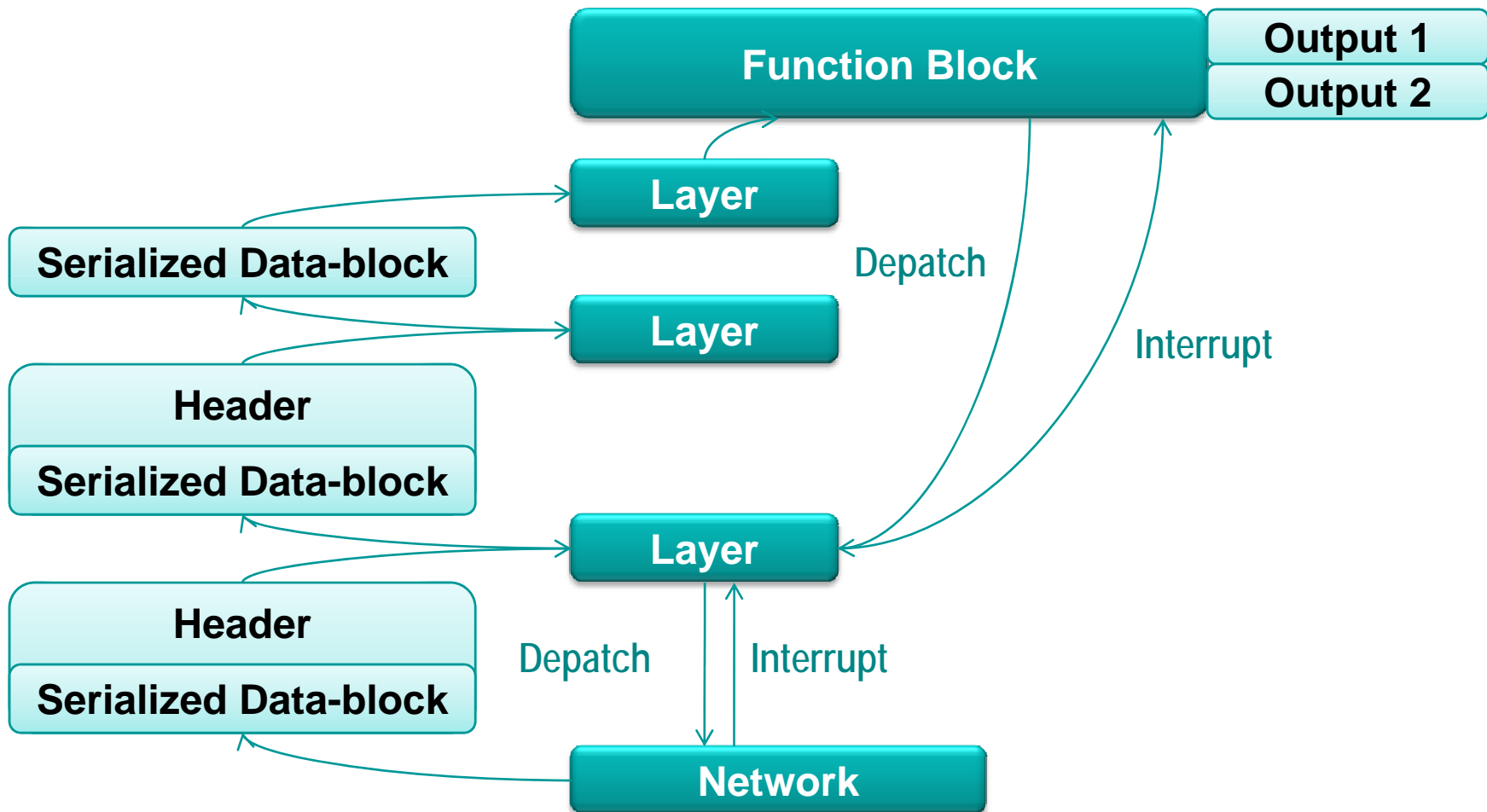
Send Data



Receive Data (1)

- Physical-Network (network-Thread)
 - Receives new data from Network
 - Informs Bottom-Layer
 - Waits until data are fetched from Bottom-Layer
- Bottom-Layer
 - Is informed from Physical layer (network-Thread)
 - Asks FB for execution (network-Thread)
 - Will be called by FB (fb-Thread)
 - Gets Data from Network and relays data to above layer (fb-Thread)
- Above-Layer
 - Has to decode, decrypt, uncompress data and calls above layer
- Top-Layer
 - Writes data do network-FB
 - Asks FB to send Event
- Network-FB
 - Dispatches interrupt request from Layer, needed to change thread-context
 - Sends event requested from Layer

Receive Data (2)



Thanks for your attention!

Contact Speaker
Michael Hofmann
PROFACTOR GmbH
Im Stadtgut A2
4407 Steyr-Gleink, AUSTRIA
michael.hofmann@profactor.at
www.profactor.at