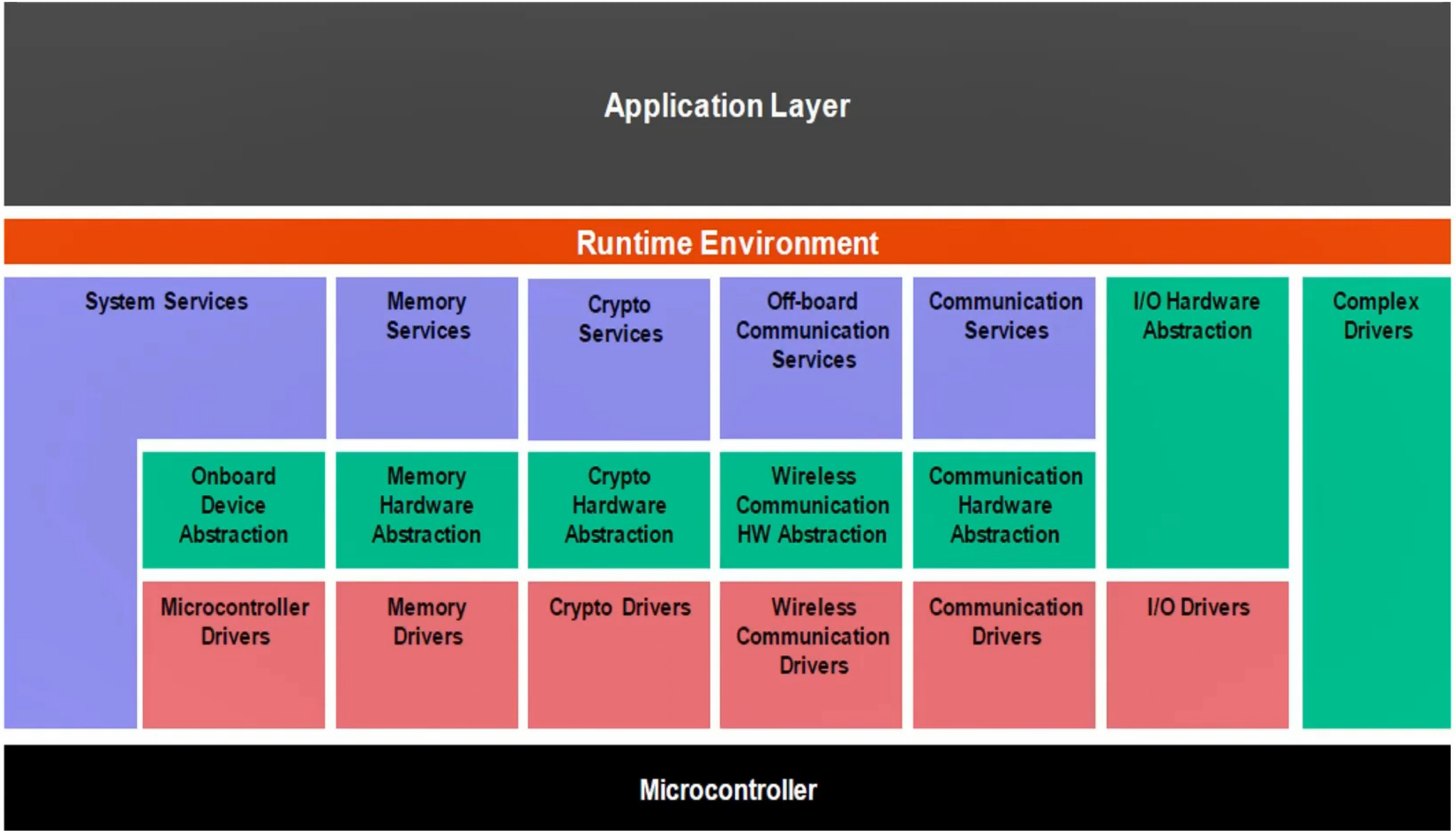
ECU ABSTRACTION LAYER


Application Layer
Runtime Environment
System Services
Memory Services
Crypto Services
Off-board Communication Services
Communication Services
I/O Hardware Abstraction
Complex Drivers
Onboard Device Abstraction
Memory Hardware Abstraction
Crypto Hardware Abstraction
Wireless Communication HW Abstraction
Communication Hardware Abstraction
Microcontroller Drivers
Memory Drivers
Crypto Drivers
Wireless Communication Drivers
Communication Drivers
I/O Drivers
Microcontroller

ECU Abstraction Layer

- Interface layers and external drivers are located in the ECU Abstraction Layer
- An Interface module contains the functionality to abstract from modules that are architecturally placed below
- It abstracts from the hardware realization of a specific device and provides a generic API to access a specific type of device-independent of the number of existing devices of that type and independent of the hardware realization of the different devices.
- E.g.: CAN If module provides a generic API to access CAN communication networks independent on the number of CAN Controllers within an ECU

DRIVER

A driver Contains the functionality to control and access an Internal or external device.

Internal Devices are located in the Microcontroller. Example

1. Internal EEPROM
2. Internal CAN Controller
3. Internal ADC

Internal Driver located in the Microcontroller.

EXTERNAL DRIVER

External Devices are located on the ECU Hardware outside of the Microcontroller . Examples are

1. External EEPROM
2. External Watchdog
3. External Flash

External driver Located in the ECU Abstraction Layer. It access the External Devices via drivers of the Microcontroller Abstraction Layer

The ECU Abstraction Layer Consist of following parts

1. I/O Hardware Abstraction
2. Communication Hardware Abstraction
3. Memory Hardware Abstraction
4. On Board Device Abstraction
5. Crypto Hardware Abstraction

The ECU Abstraction Layer is highly dependent on the ECU hardware and provides abstractions for

Communication Hardware Abstraction:

- CAN Interface (CanIf)
- CAN Transceiver Hardware(CANTrcv)
- Ethernet Interface (EthIf)
- Ethernet Transceiver Driver (EthTrcv)
- FlexRay Interface (FrIf)
- FlexRay Transceiver Driver (FrTrcv)
- LIN Interface (LinIf)
- LIN Transceiver Driver (LinTrcv)
- TTCAN (Ttcanif)

Memory HW Abstraction:

1. Memory Abstraction Interface
2. Flash EEPROM Emulation
3. EEPROM Abstraction

Onboard Device Abstraction:

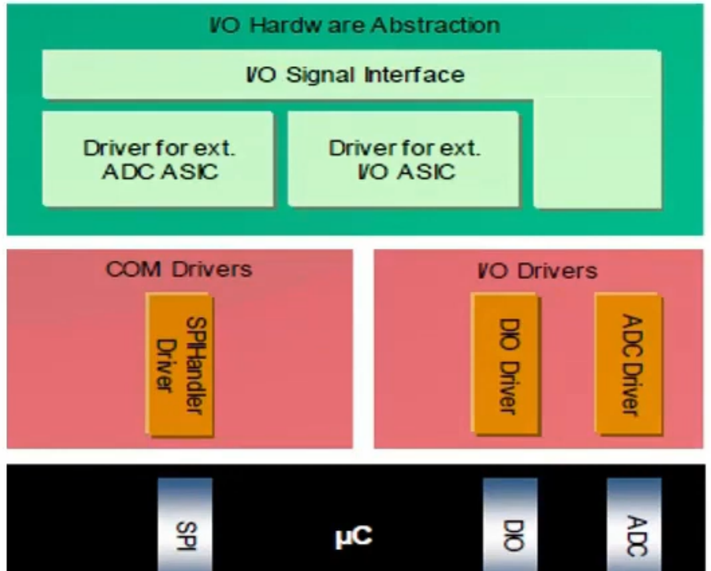
1. Watchdog Interface

I/O Hardware Abstraction:

1. I/O Hardware Abstraction

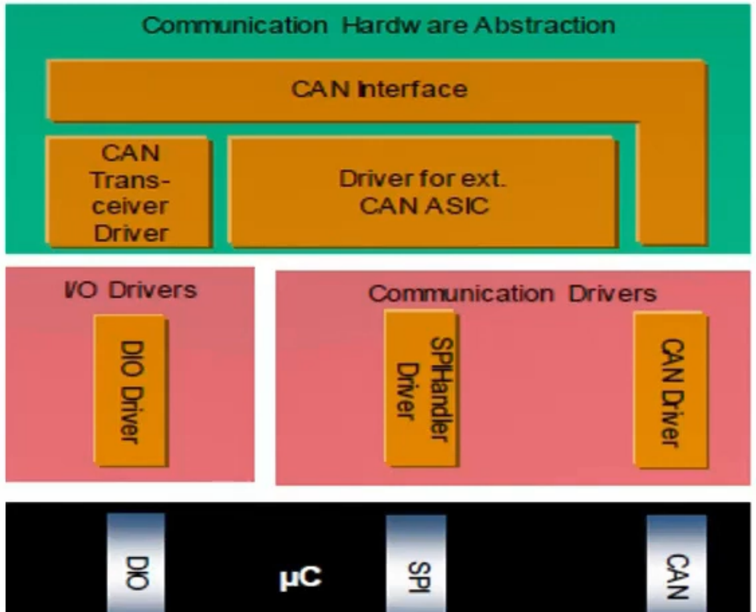
I/O HARDWARE ABSTRACTION LAYER

- The I/O Hardware Abstraction is a group of modules which abstracts from the location of peripheral I/O devices (on-chip or on-board) and the ECU hardware layout (e.g. µC pin connections and signal level inversions).
- The I/O Hardware Abstraction does not abstract from the sensors/actuators.



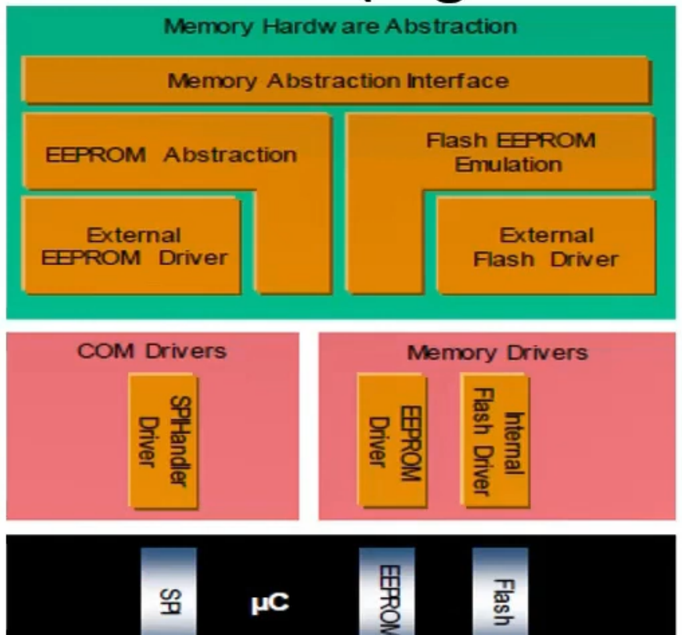
COMMUNICATION HARDWARE ABSTRACTION

- The Communication Hardware Abstraction is a group of modules which abstracts from the location of communication controllers and the ECU hardware layout.
- For all communication systems a specific Communication Hardware Abstraction is required (e.g. for LIN, CAN, FlexRay).



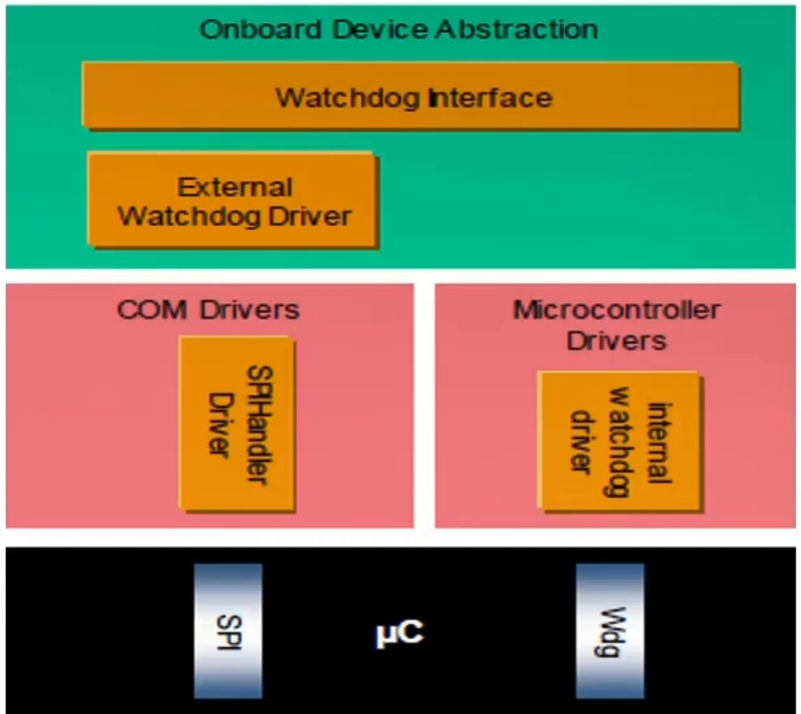
MEMORY HARDWARE ABSTRACTION

- The Memory Hardware Abstraction is a group of modules which abstracts from the location of peripheral memory devices (on-chip or on-board) and the ECU hardware layout.
- The memory drivers are accessed via memory specific abstraction/emulation modules (e.g. EEPROM Abstraction).



ON BOARD DEVICE ABSTRACTION

- The Onboard Device Abstraction contains drivers for ECU onboard devices which cannot be seen as sensors or actuators like internal or external watchdogs.
- Those drivers access the ECU onboard devices via the µC Abstraction Layer.



CRYPTO HARDWARE ABSTRACTION

The Crypto Hardware Abstraction is a group of modules which abstracts from the location of cryptographic primitives (internal- or external hardware or software-based).