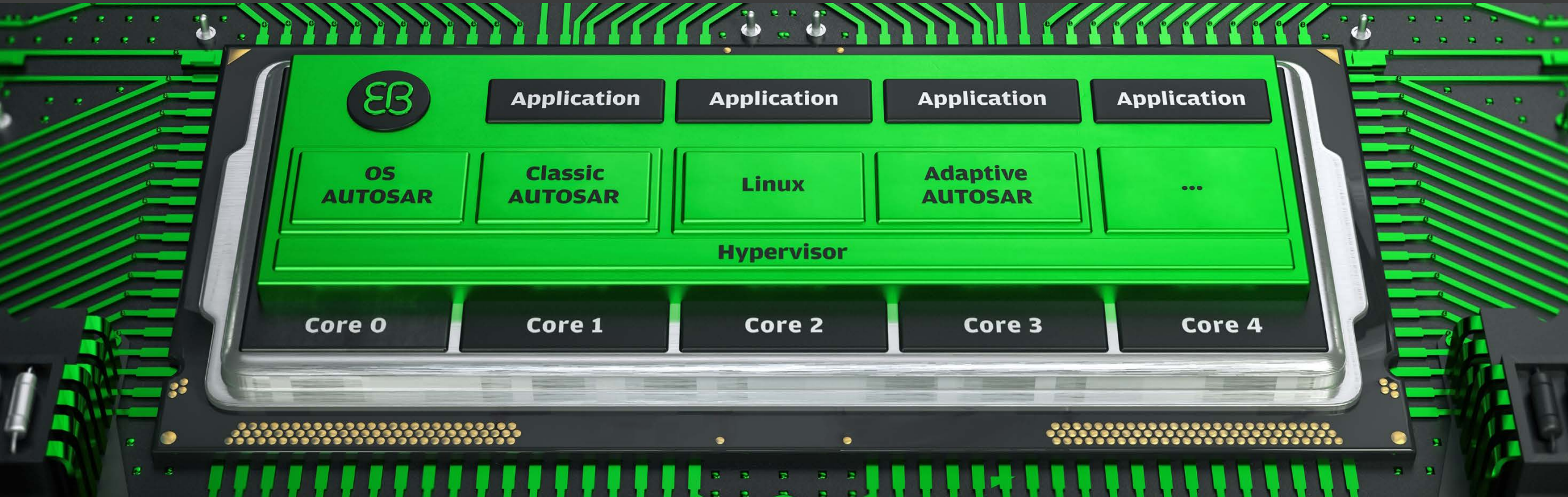


# Automotive Ethernet for Virtual Machines

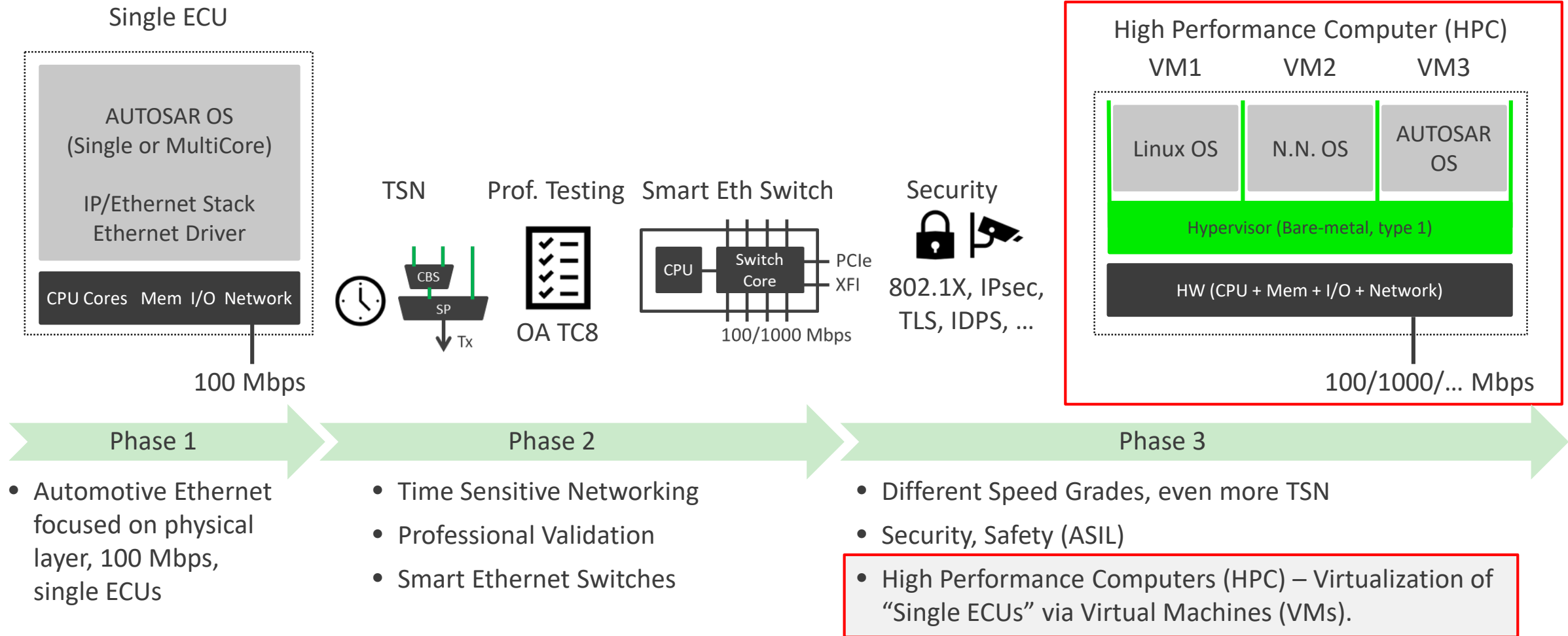


Elektrobit

2018 IEEE-SA Ethernet & IP @ Automotive Technology Day – London  
Dr. Michael Ziehensack, Elektrobit

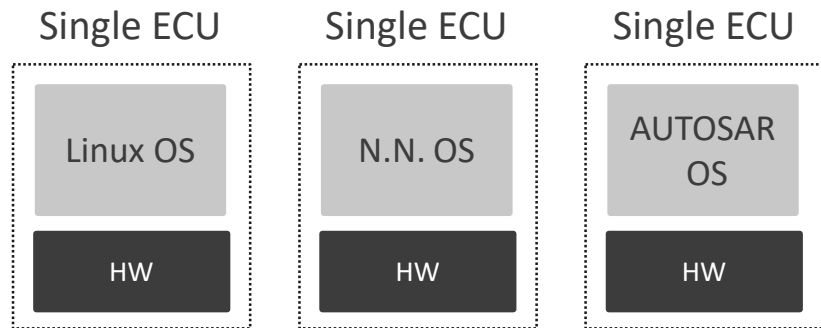


# Automotive Ethernet – Evolution Phases



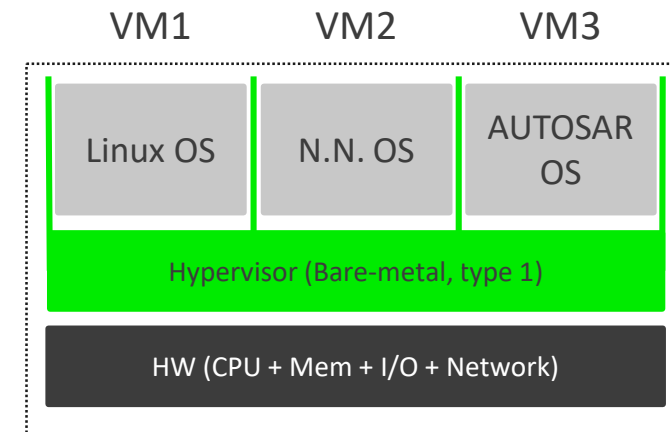
# High Performance Computers (HPC), Virtual Machines (VM)

## Single ECUs



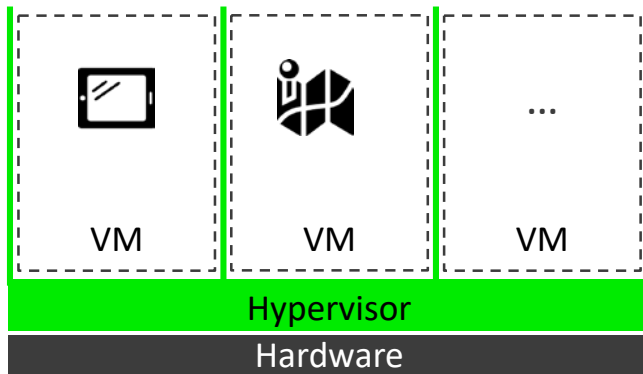
- one Operating System (Single or MultiCore) with exclusive access to the Hardware (CPU, Memory, I/O, Ethernet Controller)

## High Performance Computer (HPC)



- A HPC runs multiple OS safely and securely in parallel by using a Hypervisor
- Hypervisor has exclusive access to the hardware and provides **Virtual Machines (VMs)** as isolated, virtual duplicates of the hardware
- VMs allow to run an OS unmodified, just as it would execute directly on the real hardware

# Benefits of Virtualization (HPC, VMs)



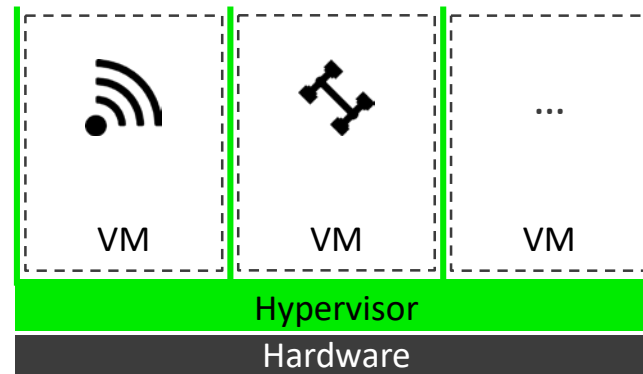
## ECU Consolidation

- Powerful SoC instead of multiple MCUs: performance controllers enable consolidation of multiple in-car applications on a single device (SW from different supplier) ...

Your Benefit

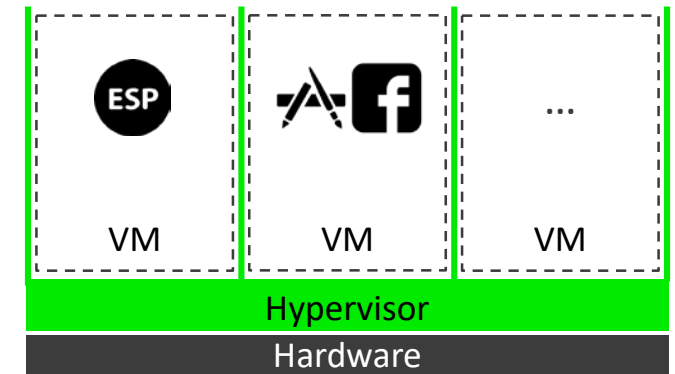


... requires less space, less heating, reduced costs as some parts are not required multiple times



## Network Separation

- Growing Car-2-X connectivity requires secure separation of out-bounded connections to the in-vehicle network



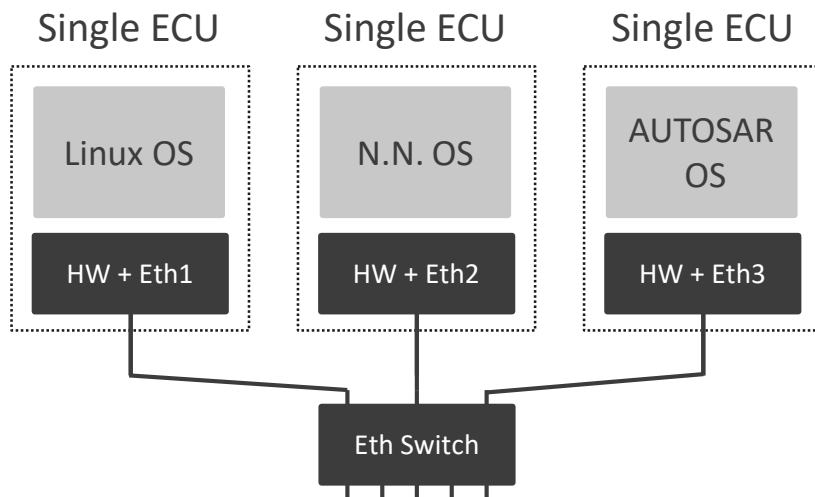
## Mixed Criticality Systems

- Usage of real-time and general-purpose operating systems in parallel, safety separation and isolation



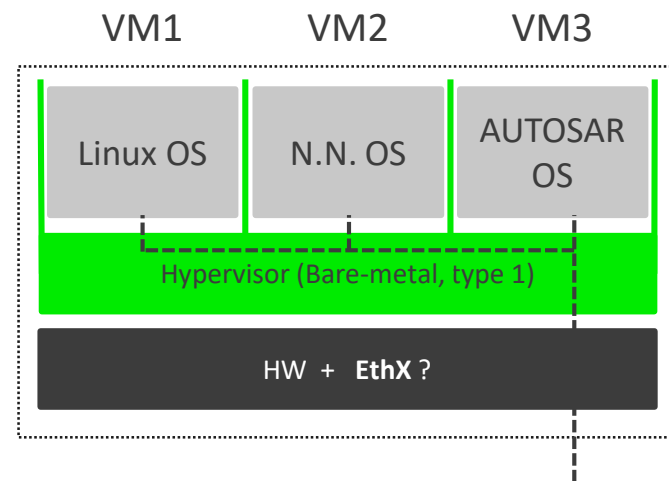
# Networking requirements for Virtual Machines

## Single ECUs



- Single ECUs, each with an Ethernet controller, are connected via an Ethernet Switch

## High Performance Computer



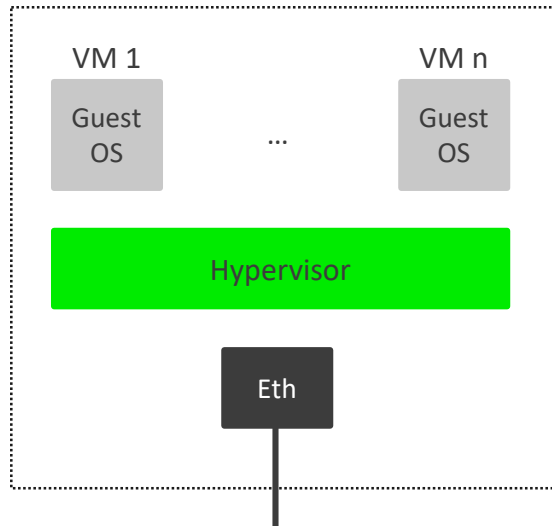
- High Performance Computer (HPC) combine multiple, Single ECUs as VMs in one device.
- For the HPC Ethernet hardware “EthX” different variants exist (see next slide)

## Requirements for VM comm.

- **01:** VMs need to communicate with the network (**VM-Net**) and each other (**VM-VM**) in a transparent way
- **02:** VM communication shall support Quality of Service (**QoS**)
- **03:** VM communication shall provide a synchronized **time** base to all VMs
- **04:** VM communication shall provide Freedom from interference at the comm. interface (**Safety**)
- **05:** VM Communication shall be protected against read/write access and suppression from other VMs (**Security:** Integrity, Availability, Confidentiality)

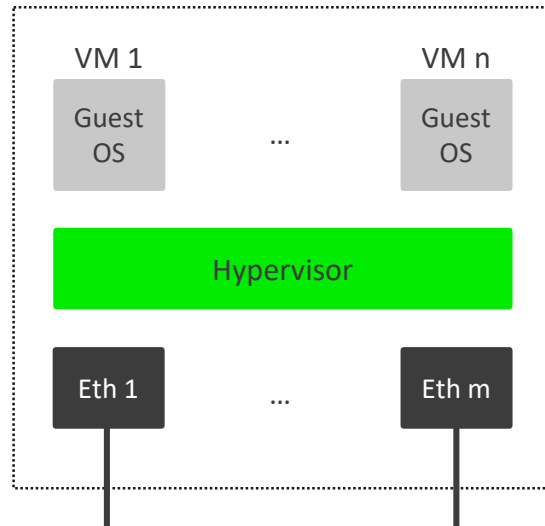
# HPC HW variants for connection to Automotive Ethernet

## V01: Single Eth Interface



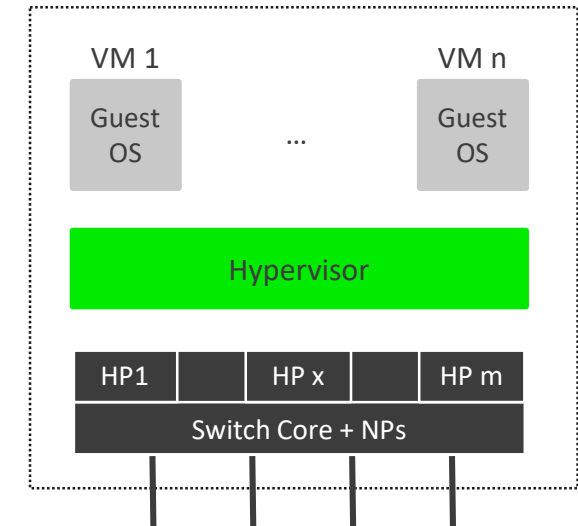
- HPC with single Ethernet HW interface which need to be shared between VMs
- In this case the Ethernet Controller typically provides some extensions, e.g. multiple receive and transmit queues.

## V02: Multiple Eth Interface



- HPC with two or more Ethernet HW interfaces.
- Requires multiple ports at an external Ethernet Switch

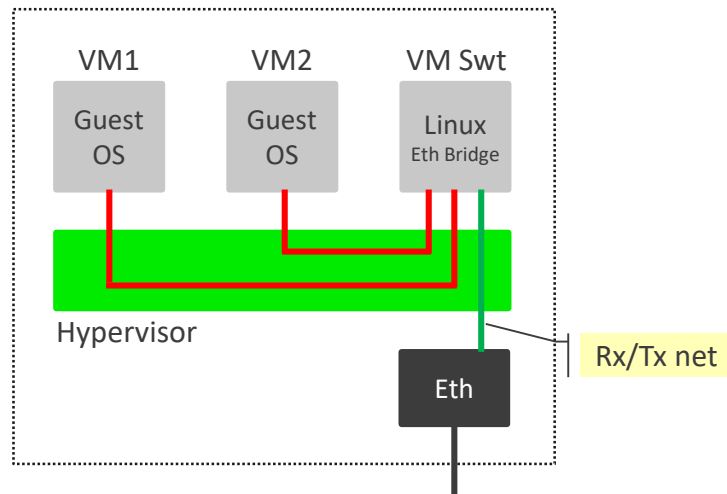
## V03: Integrated Eth Switch



- HPC with integrated Ethernet Switch HW:
  - Provides internal ports (Host Ports), i.e. Buffer-Descriptor based Interface for VMs
  - Provides external ports (Network Ports) to connect other ECUs

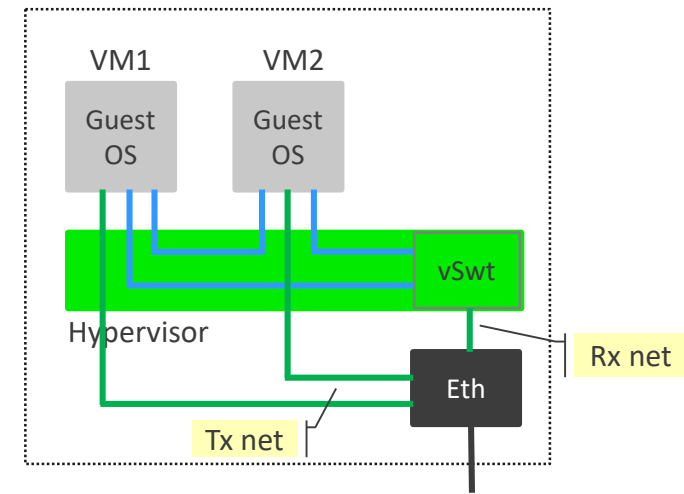
# HPC HW variant V01: Single Eth interface

## V01a: Virtual Switch in Linux VM



- additional Virtual Machine “VM Swt” as Virtual Switch, running Linux and providing a Linux Ethernet Driver and Eth Bridge
- VM Swt has exclusive access to the Ethernet HW, data to/from the network are transferred to the VM Swt via Eth-DMA
- Linux Bridge (Kernel Mode) forwards data between the bridge ports, data to/from VMs are forwarded via CPU copy

## V01b: Virtual Switch in Hypervisor

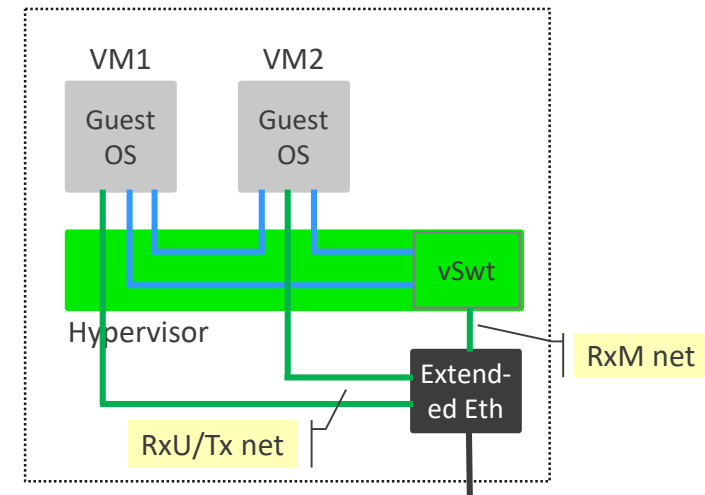
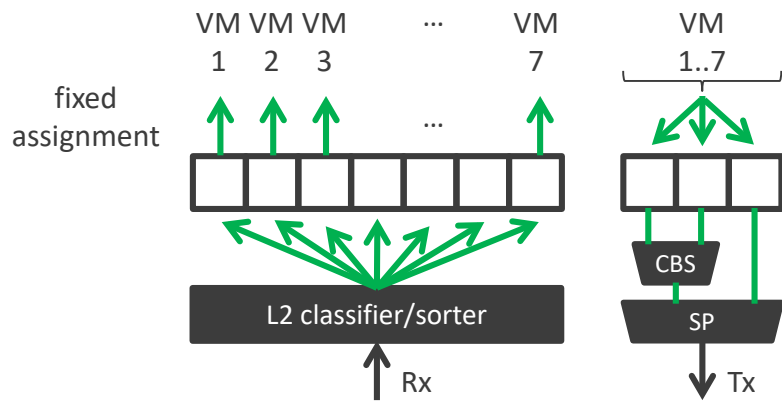


- A virtual Switch module is running in the Hypervisor or in an AUTOSAR OS executed on a dedicated (safety) Core
- The virtual Switch module contains an Ethernet Driver and has exclusive access to the Ethernet HW
- Eth-DMA is used to forward data to/from the VMs and the network, RT-DMA is used to forward data between VMs

Data path: — CPU copy — Eth-DMA — RT-DMA

# HPC HW variant V01: Single Eth interface

## V01c: Virtual Switch in Hypervisor, extended Ethernet HW



- provides multiple Rx queues with a L2 classifier and sorter
- Incoming frames are **classified** (e.g. destination MAC address) and **sorted** into a dedicated Rx queue per VM (fixed assignment)
- Shared Tx queues for QoS

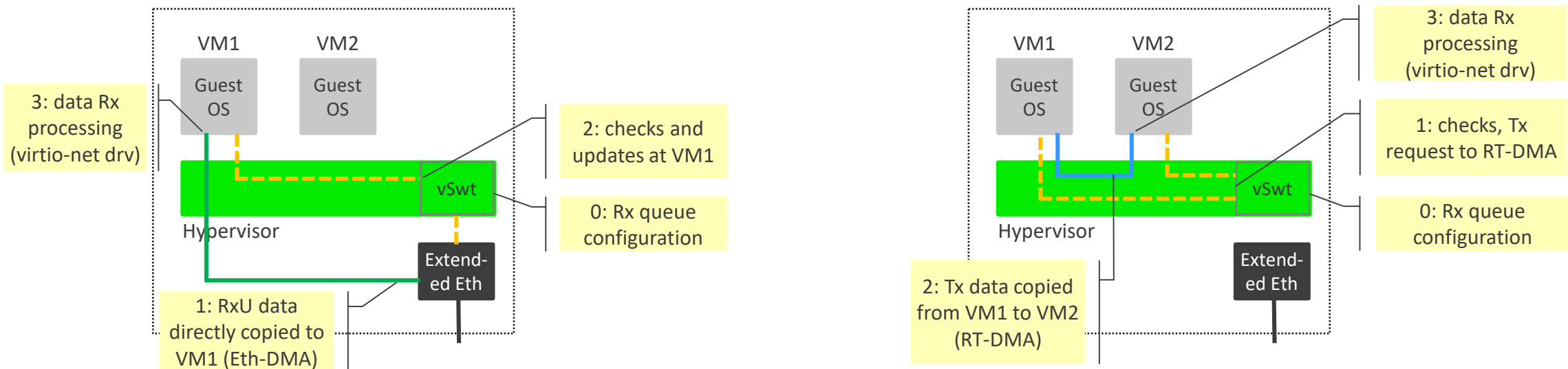
- Same as variant **V01b with optimization of the RxU path:**
- Reception of unicast frames is directly done by the Eth-DMA by using a **dedicated Rx queue for each VM**
- Multicast frames still need to be received via the vSw module (Eth-DMA, then RT-DMA)

Data path: — CPU copy — Eth-DMA — RT-DMA — RxU ... Rx unicast — RxM ... Rx multicast



# HPC HW variant V01: Single Eth interface

## V01c: Virtual Switch in Hypervisor, extended Ethernet HW



- **Example Sequence A:**  
Reception Unicast from the network to VM1
- Control path shown as dotted line

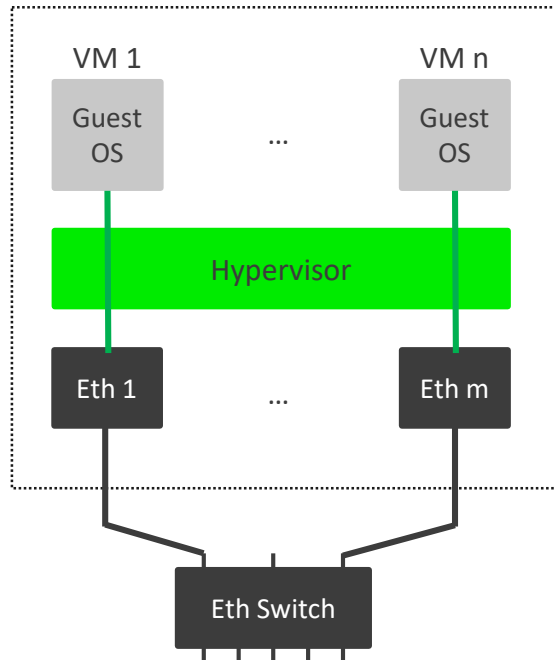
- **Example Sequence B:**  
Data transmission from VM1 to VM2
- Control path shown as dotted line

Control path: - - -

Data path: — CPU copy — Eth-DMA — RT-DMA

RxU ... Rx unicast RxM ... Rx multicast

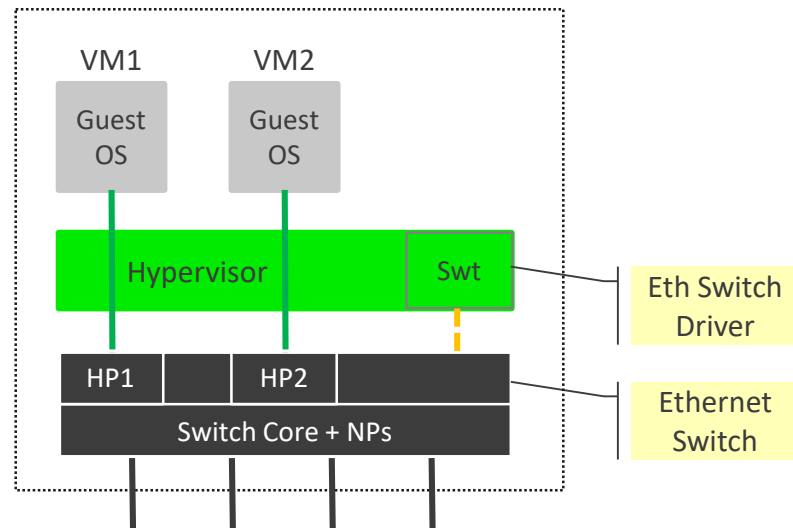
# HPC HW variant V02: Multiple Eth interfaces



- HPC with two or more Ethernet HW interfaces.
- Requires multiple ports at an external Ethernet Switch
- Hypervisor assigns a dedicated Ethernet Controller to each VM (1:1)
- Traffic between VMs goes via an external Ethernet Switch
- In case the number of VMs is higher than available Ethernet interfaces, some VMs can share an Ethernet interface by using variant V01.

Data path:  CPU copy  Eth-DMA  RT-DMA

# HPC HW variant V03: Integrated Eth switch

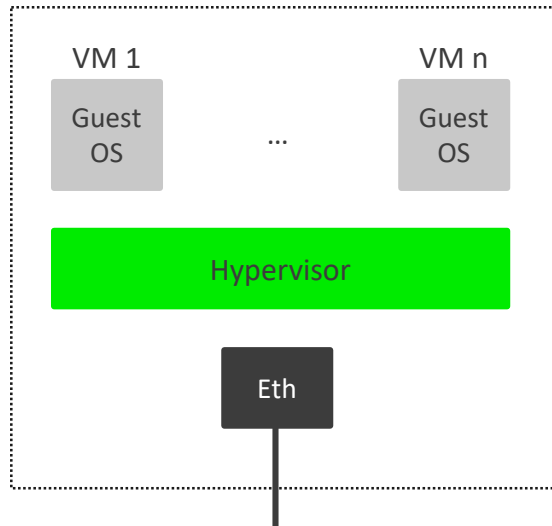


- HPC with integrated Ethernet Switch HW:
  - Provides internal ports (Host Ports), i.e. Buffer-Descriptor based Interface for VMs
  - Provides external ports (Network Ports) to connect other ECUs

- Hypervisor assigns a dedicated Host Port to each VM (1:1)
- Switch core forwards data between Host Ports (internal traffic), between Host Ports and Network Ports, and between Network Ports.
- The configuration and control of the switch is done via a Switch driver module running in the Hypervisor or in an AUTOSAR OS on a (safety) Core.
- There are also HW variants that do not provide a complete switch core, but specific HW accelerators for packet processing (such as a programmable packet forwarding engine) that can be used to recreate a switch.
- In case the number of VMs is higher than available Host Ports, some VMs can share a Host Port by using variant V01.

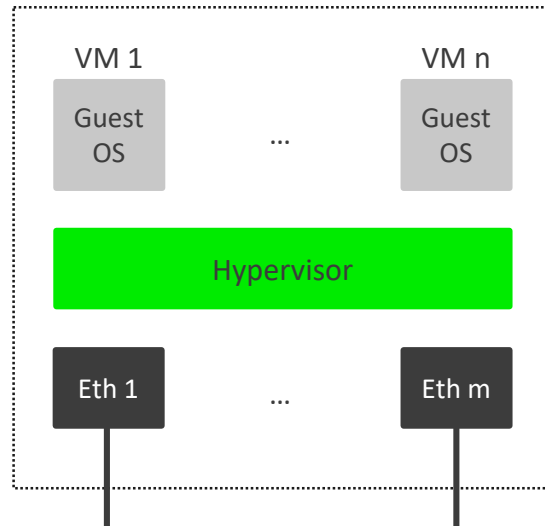
# HPC HW variants – Comparison

## V01: Single Eth Interface



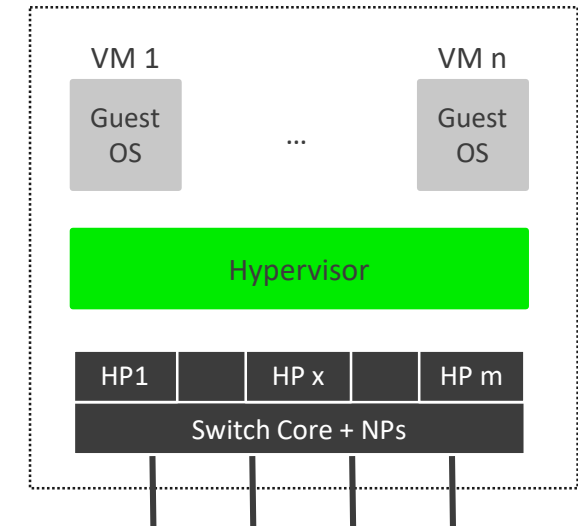
- HW: simple
- SW: Virtual Switch module, flexible
- Perf: low to high (depends on SW)

## V02: Multiple Eth Interface



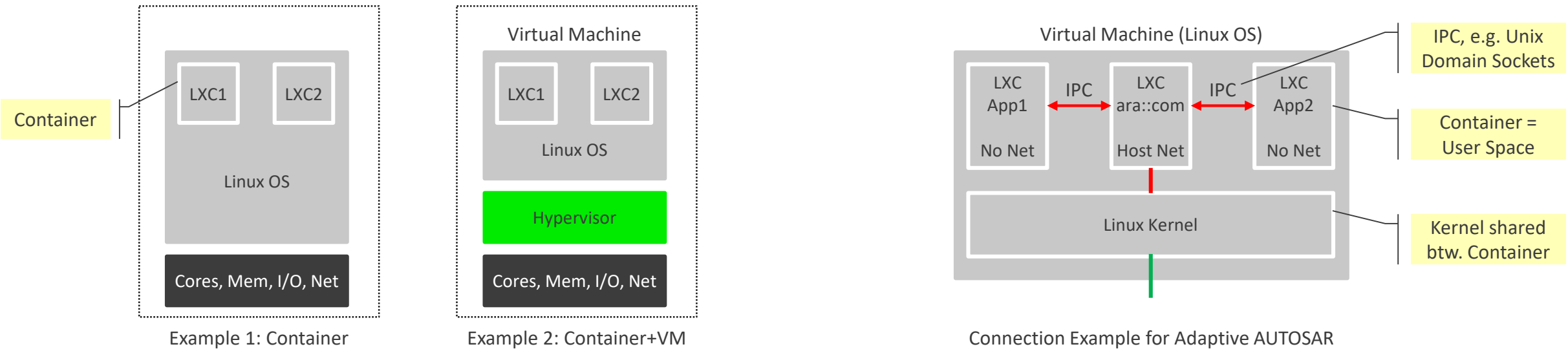
- HW: simple to complex (depends on number of VMs)
- SW: none
- Perf: high

## V03: Integrated Eth Switch



- HW: complex
- SW: Switch driver for config and control
- Perf: very high

# A further aspect: OS-level virtualization – Connection of Containers



- **OS-level virtualization** provides isolated user-space instances (**Container**) that share the same Linux kernel.
  - A process running inside a Container only sees the Containers content and the devices assigned to the Container.
  - Containers are typically faster but not as secure as HW-Virtualization. Can be used as an alternative (Ex. 1) or in addition to VMs (Ex. 2).

- Relevant **connection types** for Container
  - **Bridge**: the Container gets a virtual Ethernet interface that is connected to the bridge at the host, gets its own MAC and IP address (appears as separate physical machine on the network, in case the Container shall not be seen on the network, NAT can be used)
  - **Host**: the Container directly uses the existing interface of the Host, i.e. MAC and IP address of Host is used, typically used for Automotive SoC

— CPU copy

— Eth-DMA

# Summary

- **High Performance Computer** provide Virtualization supporting ECU Consolidation, Network Separation, Mixed Criticality Systems and **become a key part of the E/E architecture**.
- There are different variants to connect **Virtual Machines** and **Containers** of High Performance Computers to **Automotive Ethernet** depending on the hardware capabilities of the HPC.
- To ensure a high speed connection to the VMs **optimizations** for the data path are of crucial importance.
- Please join us at booth #11 (**Elektrobit**) if you want get more details of our optimized **implementations** to connect Virtual Machines to Automotive Ethernet.



## Thank you for your attention !!

---

### Author information

Dr. Michael Ziehensack, Elektrobit

VP, Car Infrastructure Software

[michael.ziehensack@elektrobit.com](mailto:michael.ziehensack@elektrobit.com)

# Get in touch!



Elektrobit

[sales@elektrobit.com](mailto:sales@elektrobit.com)  
[www.elektrobit.com](http://www.elektrobit.com)

