

EC ↔ *Master*

EC ↔ *Engineer*

EC ↔ *EngineerWeb*

EC ↔ *Simulator*

Software and Tools for
EtherCAT® Master Controller Development

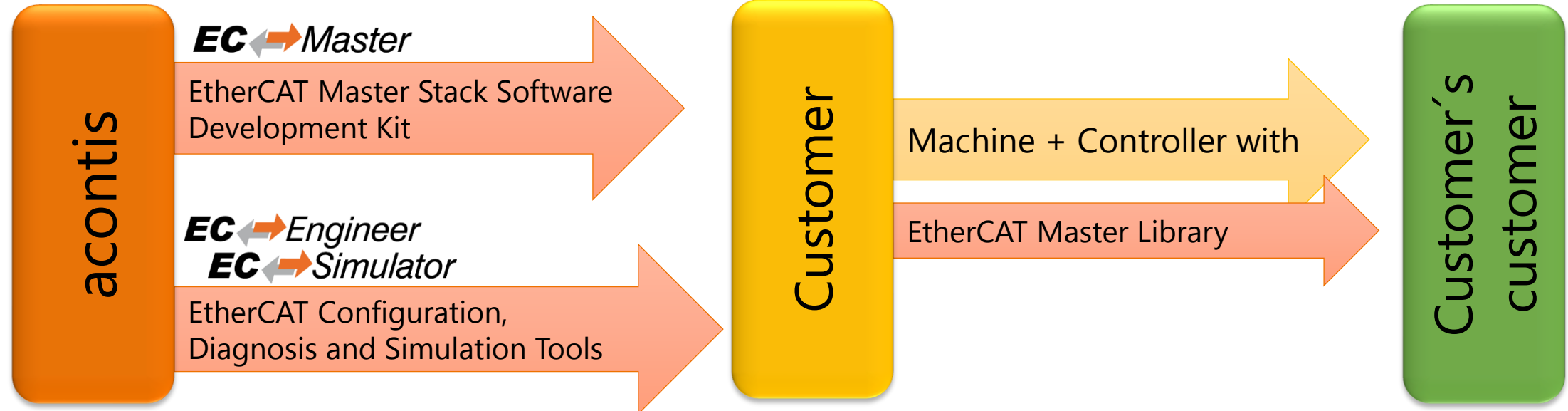
EtherCAT Software Solutions for Machine Builder Controller

EC ↔ **Master**

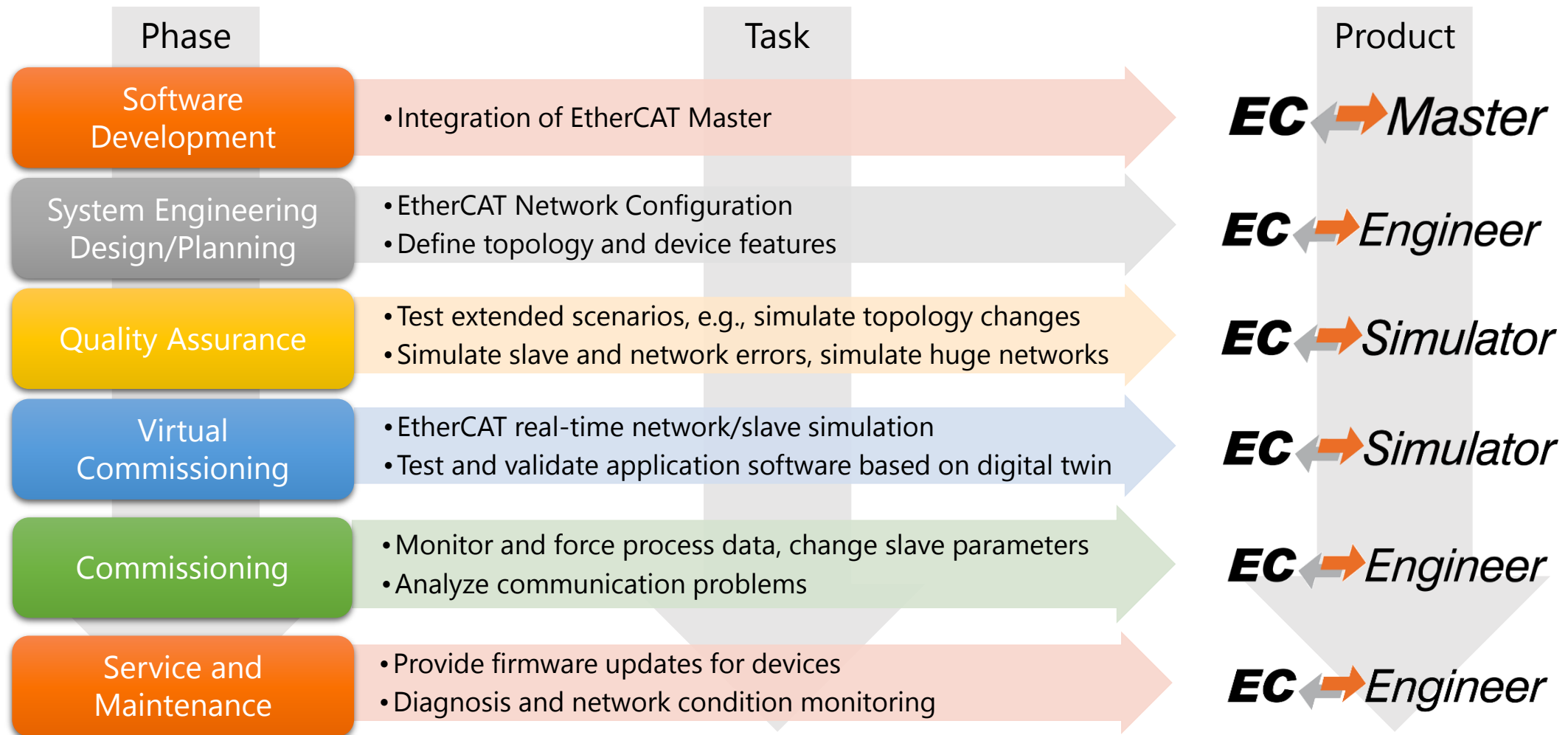


Machine Builders

- Motion Control, CNC, Material Handling
- Surgical Robotics, Simulators
- Test benches, Semiconductor tooling



Machine Builder Controller Development Life Cycle

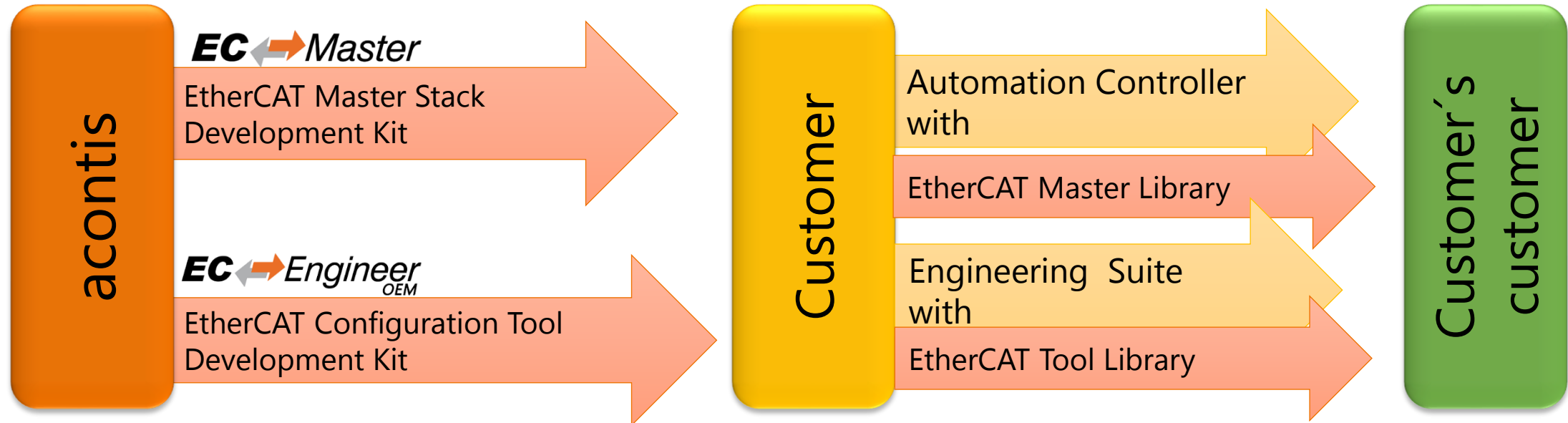


EtherCAT Software Solutions for Automation Controller Manufacturers

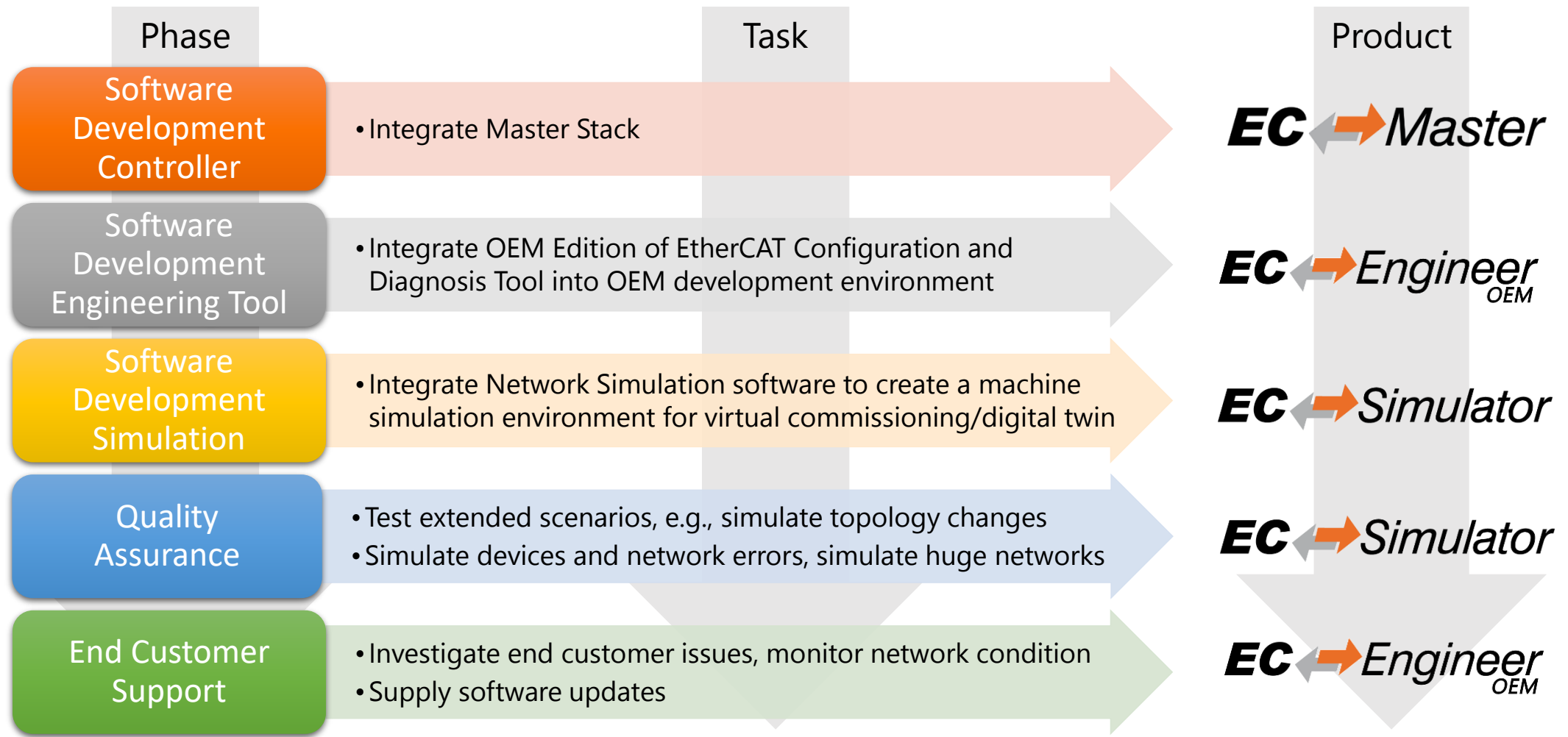


Automation Controller Manufacturers

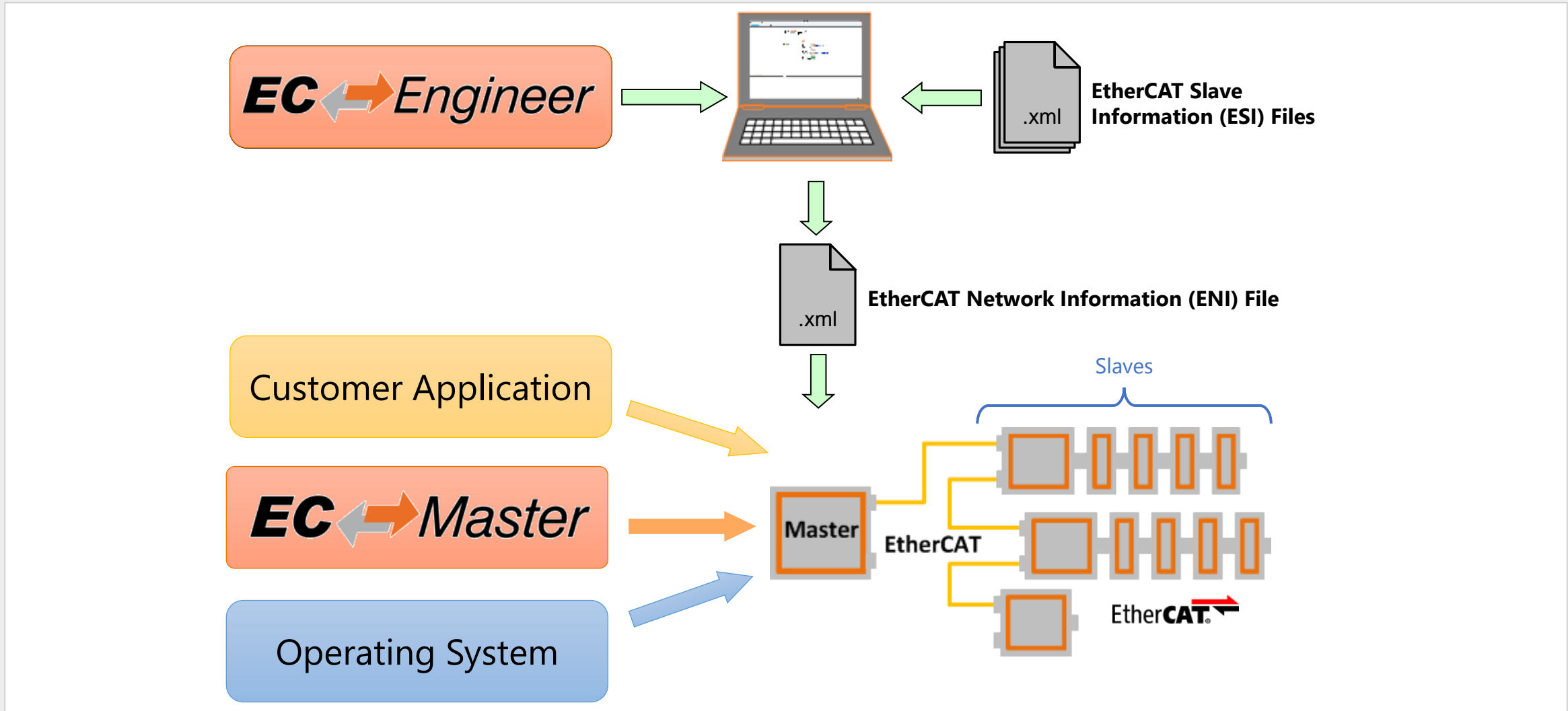
- Programmable Logic/Automation Controller (PLC/PAC)
- Motion Controllers (MC), Measurement Controllers
- HMI with controller



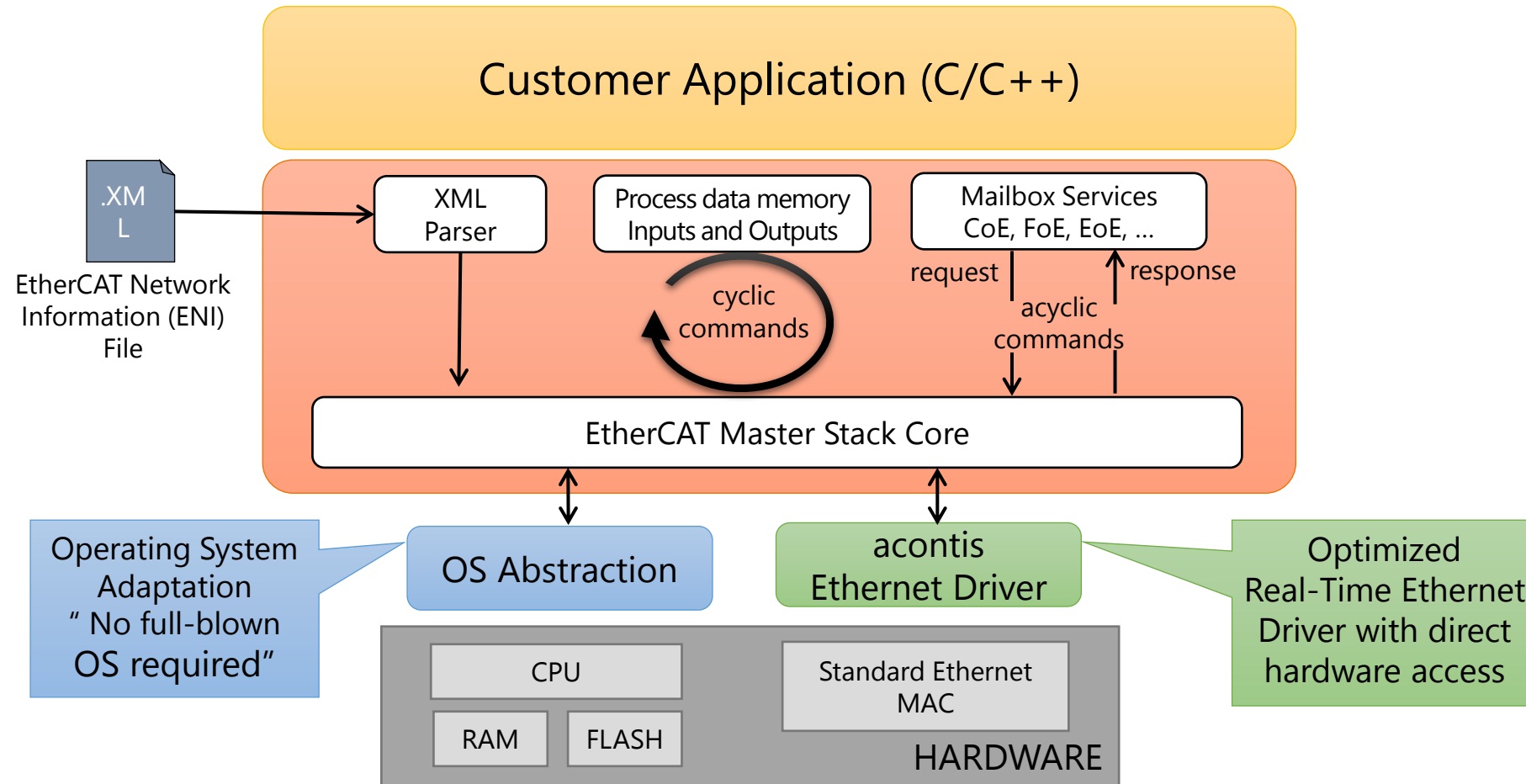
Automation Controller Manufacturers Development Life Cycle



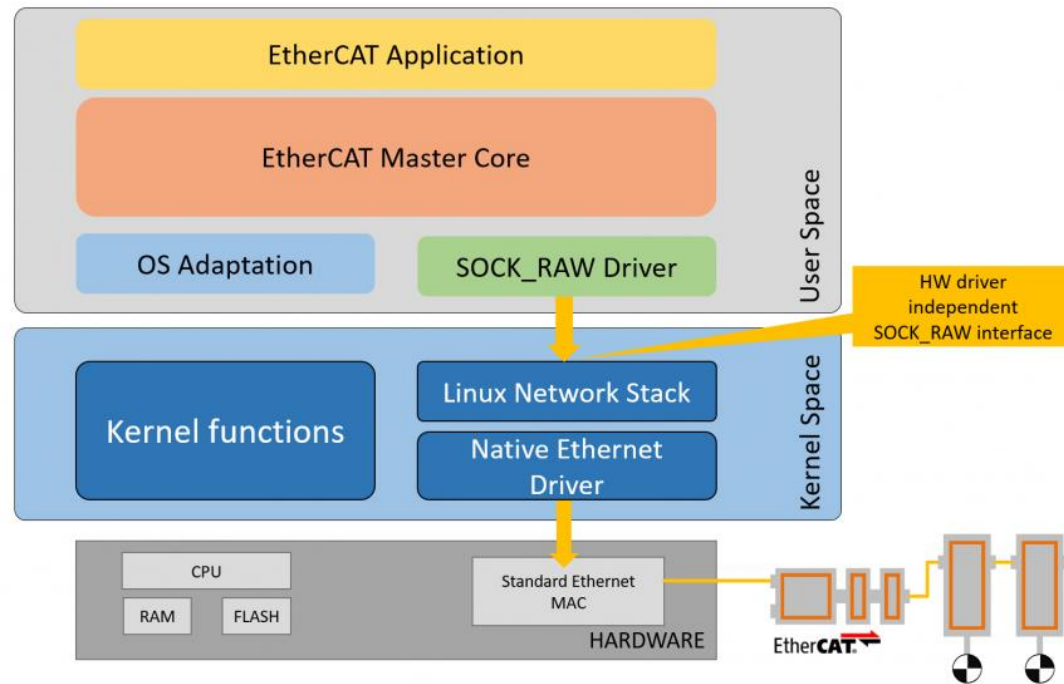
EtherCAT System Architecture



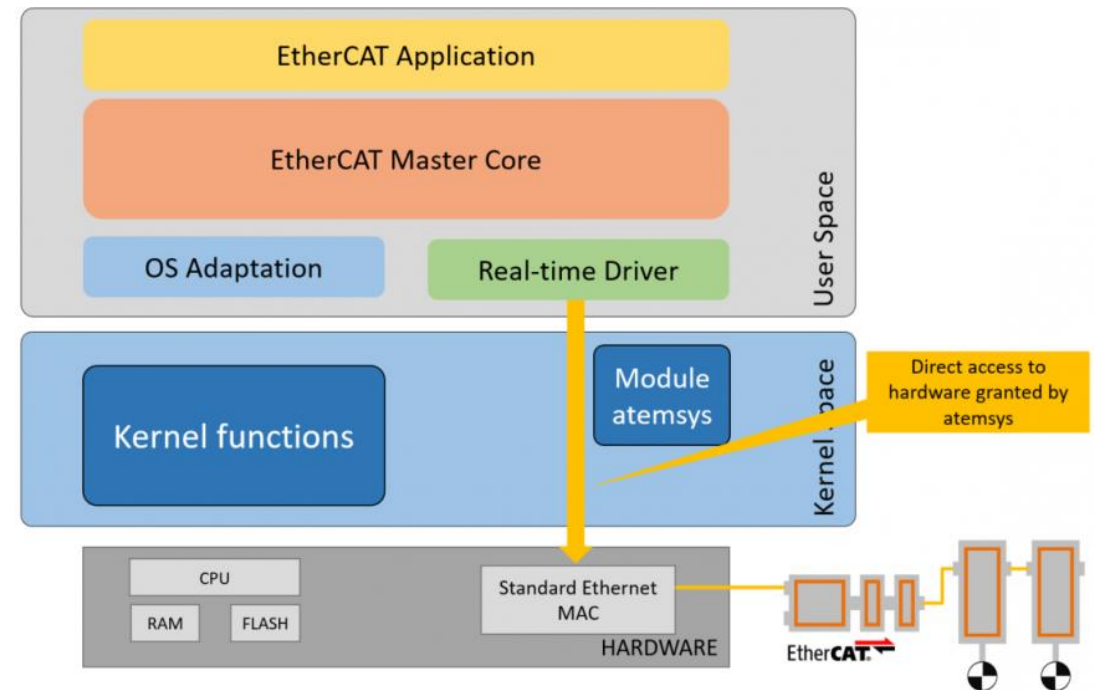
EC-Master Software Architecture



Architecture 1: Linux Network Driver



Architecture 2: acontis Real-time Ethernet Driver



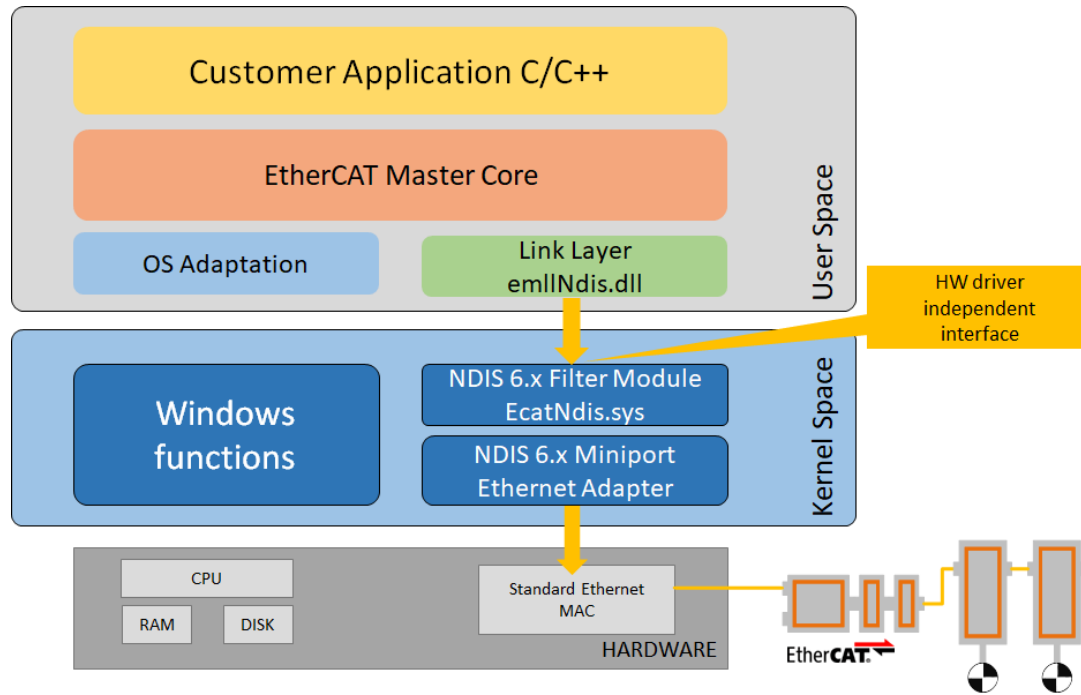
More Information: <https://www.acontis.com/en/ethercat-master-linux.html>

EC-Master Architecture on Windows (1)

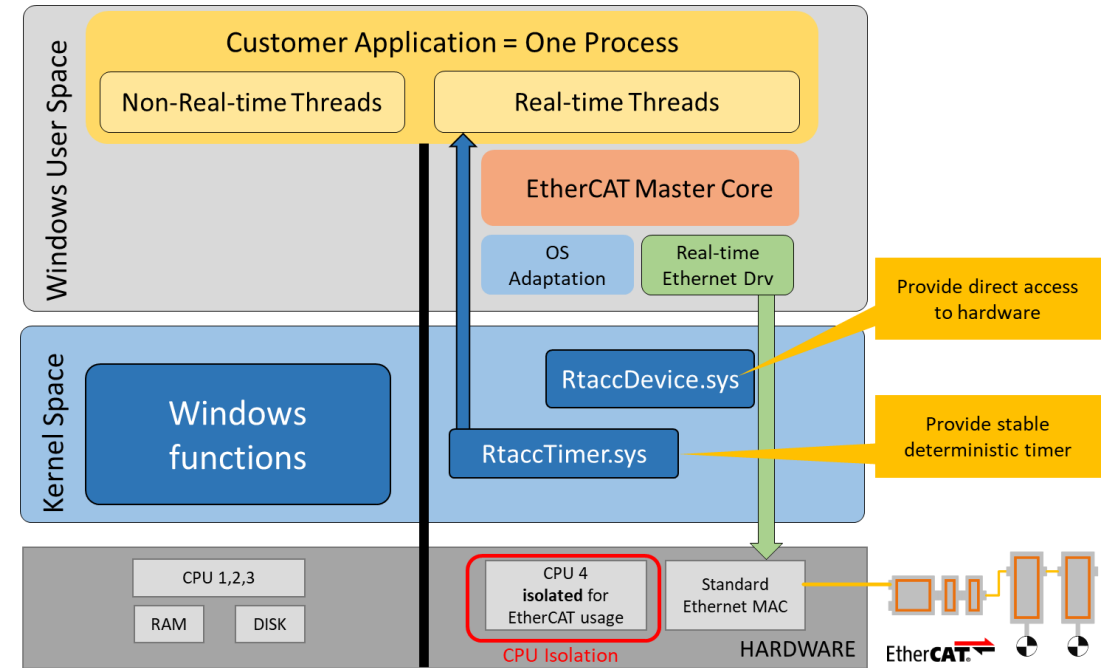
Solution 1 Off-the-Shelf & 2 CPU Isolation



Architecture 1: **Off-the-shelf**
 with acontis NDIS Filter Driver
 No reliable cycle time: ≥ 4 ms



Architecture 2: **CPU Isolation**
 with acontis RtaccWin
 Guaranteed cycle time: ≥ 1 ms



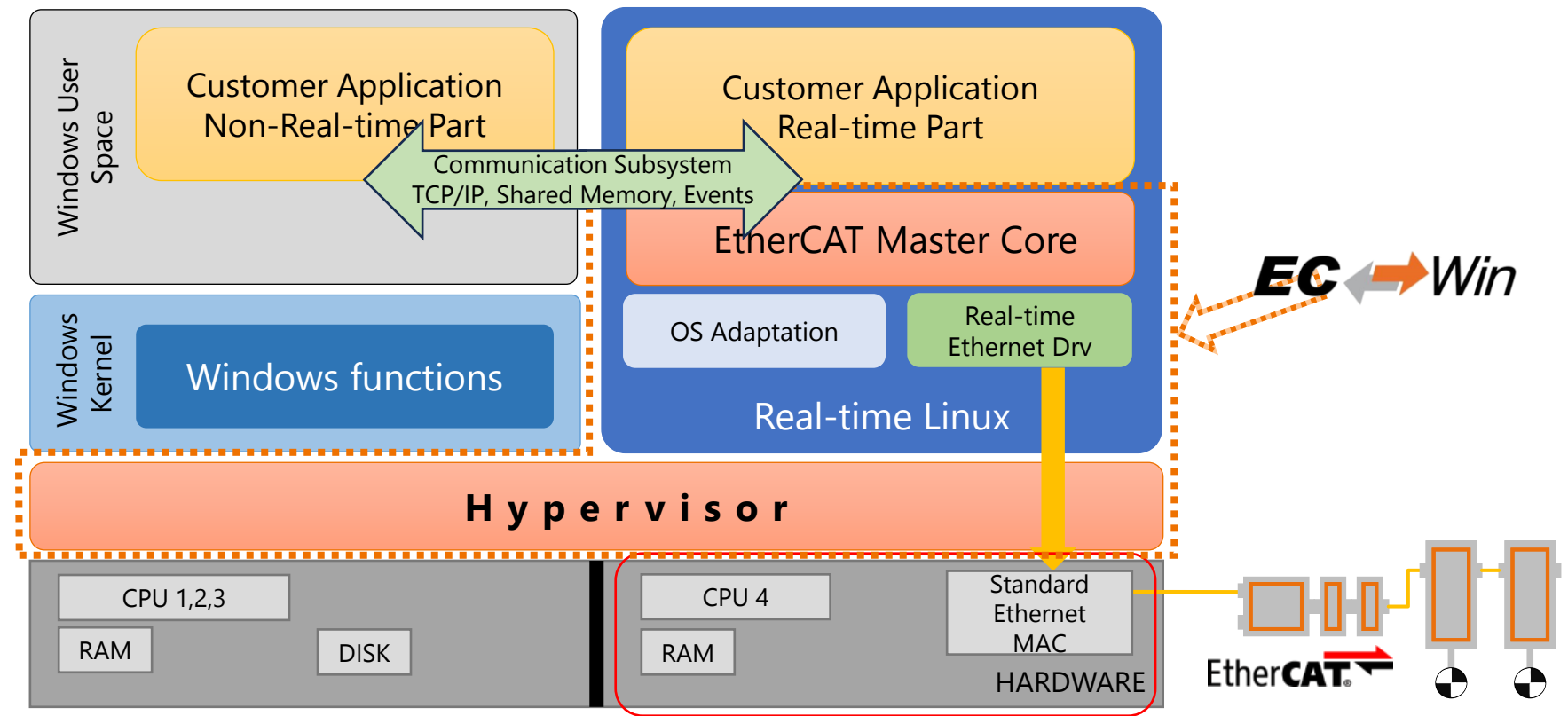
More Information: <https://www.acontis.com/en/ethercat-master-linux.html>

EC-Master Architecture on Windows (2)

Solution 3: Full Isolation, based on Hypervisor



Architecture 3: **Hypervisor** with hardware separation, enabling RT-Linux & Windows on one computer
Split real-time and non-real-time part
Controller App and Master stack run in Real-time Environment with **up to 10kHz Cycle**



EC-Master Out of the box for many Platforms

EC ↔ **Master**

OS Abstraction



Ethernet Controllers

Intel Pro/1000

Xilinx GEM

Realtek Gigabit

NXP
FEC, eTSEC

Renesas RZ Family

TI Sitara
CPSW, ICSS-PRU

Intel Elkhart Lake

SMSC 9218

Beckhoff CCAT

and more

X86
32-Bit

X64
64-Bit

ARM
32-Bit

Aarch64
64-Bit

PowerPC

EC-Master Available for Many Platforms

EC ↔ **Master**

25 Operating Systems

22 Ethernet controller families

5 CPU Architectures

> 90

Combinations

Class B Core

- Compare network configuration
- Cyclic process data exchange
- Slave to slave communication
- Mailbox protocols CoE, SoE
- Mailbox protocols EoE, FoE
- Mailbox protocols AoE, VoE


Class A Core

- All Class B Features
- **Distributed Clocks with master synchronization**

Feature Packs = Options

Cable Redundancy, Hot Connect, Superset ENI,
External Synchronization, EoE Gateway, Master Redundancy, ...

EC-Master broad CPU support

 BROADCOM	BCM2711 - Raspberry Pi 4 (Cortex-A72)
	XMC4800 (Cortex M4)
	Core-i Atom Atom® x6000E (Elkhart Lake) Altera Cyclone V Celeron, Xeon
 NVIDIA	Jetson TX2 (Quad Cortex-A57)
	i.MX6 (Cortex-A9) i.MX7 (Cortex-A7), iMX RT1064 (Cortex M7) i.MX8 (Cortex-A53) Layerscape 1021A QorIQ P-Series MPC8548 PowerQUICC
 Qualcomm	QRB5165 with Kryo 585 CPU (Octo Cortex-A77)

 RENESAS	RIN32, RZ-T, RZ-N, RZ-G (Cortex-A55) RZ-A (Cortex-A9)
 Rockchip 瑞芯微电子	RK3328 (Quad Cortex A53) RK3399 (Dual Cortex-A72) RK3588s (Quad Cortex-A76) RK3568 (Quad Cortex-A55)
 ST life.augmented	STM32MP1 (Dual Cortex-A7) STM32H7 (Cortex M7) STM32F769 (Cortex M7)
 TEXAS INSTRUMENTS	Sitara AM335x (Cortex-A8) Sitara AM437x (Cortex-A9) Sitara AM57xx (Dual Cortex-A15) Sitara AM64x (Dual Cortex-A53) Jacinto TDA4VM (Dual Cortex A72 / Quad Cortex R5F)
 XILINX	Zync-7000 (Dual Cortex-A9) Zynq UltraScale+ (Quad Cortex-A53)

Hardware Requirements

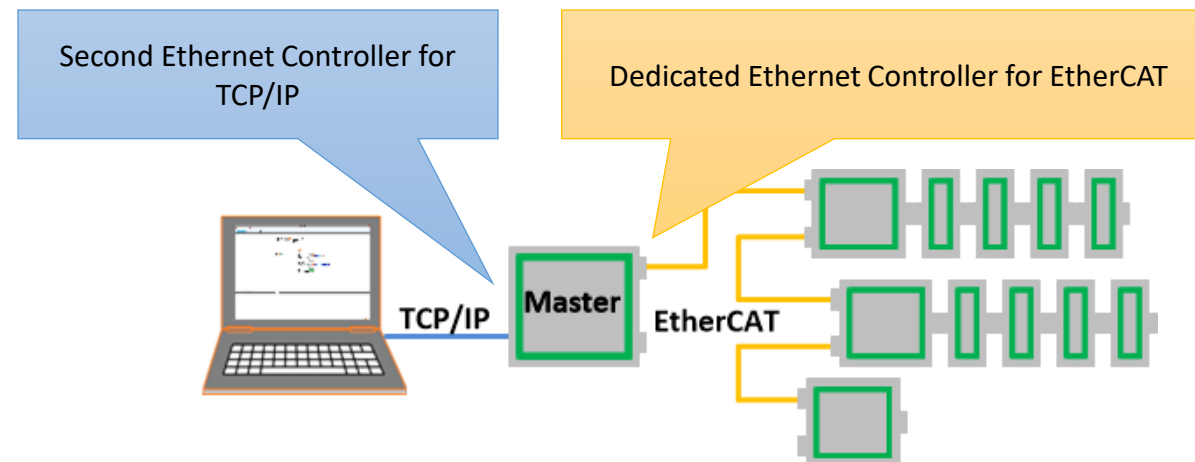
32 Bit or 64 Bit CPU

Footprint

ROM (FLASH or DISK): Between 400 and 900 Kbyte

RAM: Starting with 300 Kbyte up to several Megabyte.

Highly depending on the number and type of slaves!



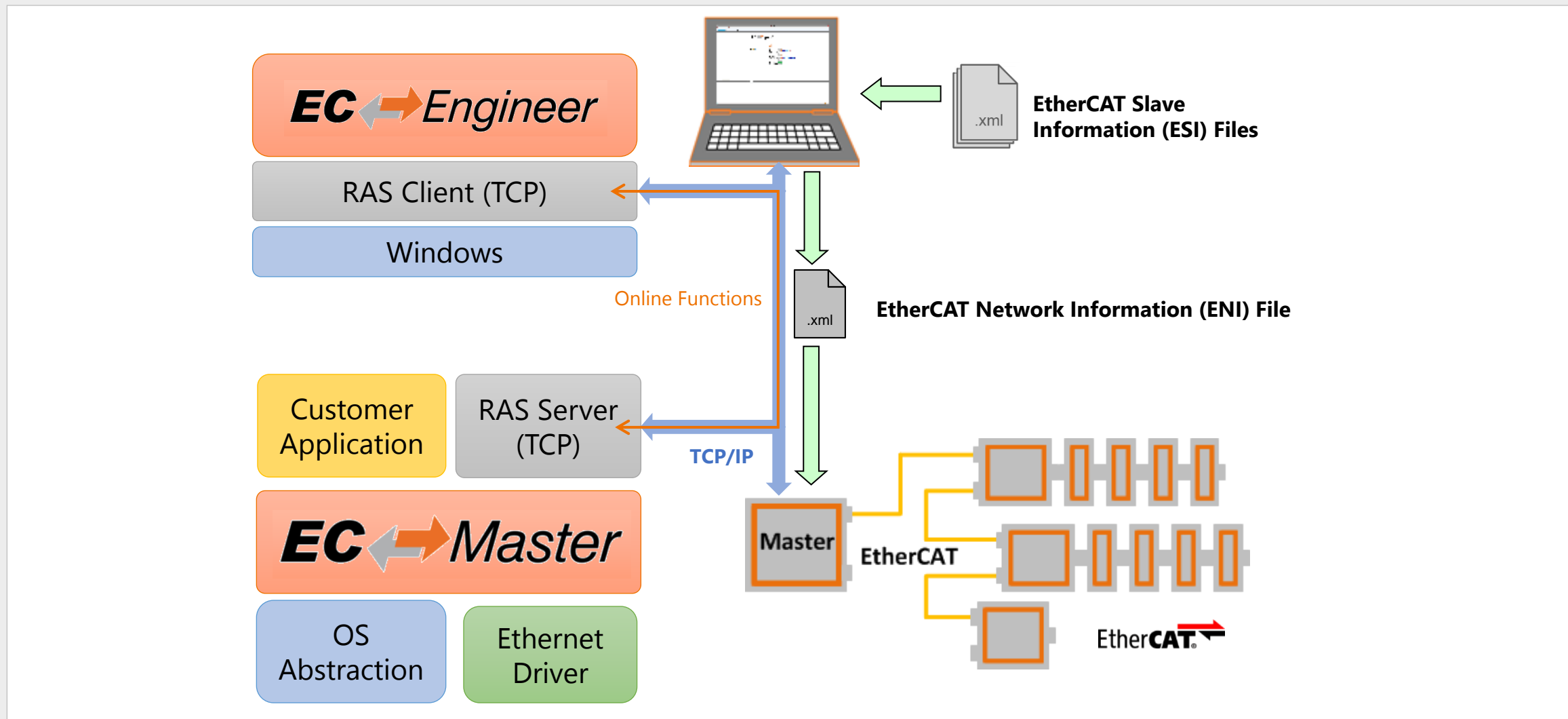
- **Out-of-the-box** for the most popular operating systems
 - ⇒ Get it running on your system in one day!
- **Reliable** and **robust** implementation
 - ⇒ Field proven in several 100000 systems per year!
- Sophisticated **diagnosis** functions
 - ⇒ Detect state change problems and frame loss errors easily
 - ⇒ More than 300 different error codes
- High **performance** and **hard real-time**
 - ⇒ Low CPU load due to acontis real-time Ethernet drivers
- **Easy** to integrate
 - ⇒ Various example applications and comprehensive user manuals

- EC-Master Operating Systems and Real-time Ethernet Drivers
<https://www.acontis.com/en/os.html>
- EC-Master user manual and quick start guide
<https://developer.acontis.com/ec-master>
- Request for EC-Master feature pack slides
- Request for EC-Master technical details slides
- Request for evaluation software
<https://www.acontis.com/en/ethernetcat-support-eval-request.html>

EC  *Engineer*

EtherCAT Configuration and Diagnosis Tool

EtherCAT System Architecture



Operating Modes

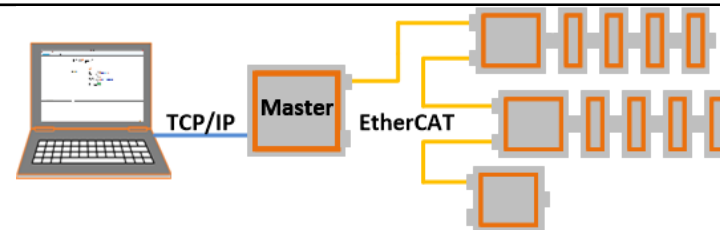
Offline **Configuration:**
(In the Office)



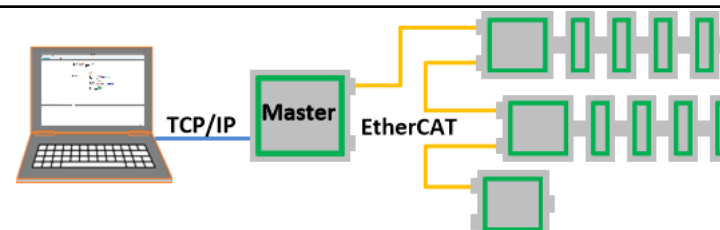
Online **Configuration:**
Slaves connected to
Engineering System



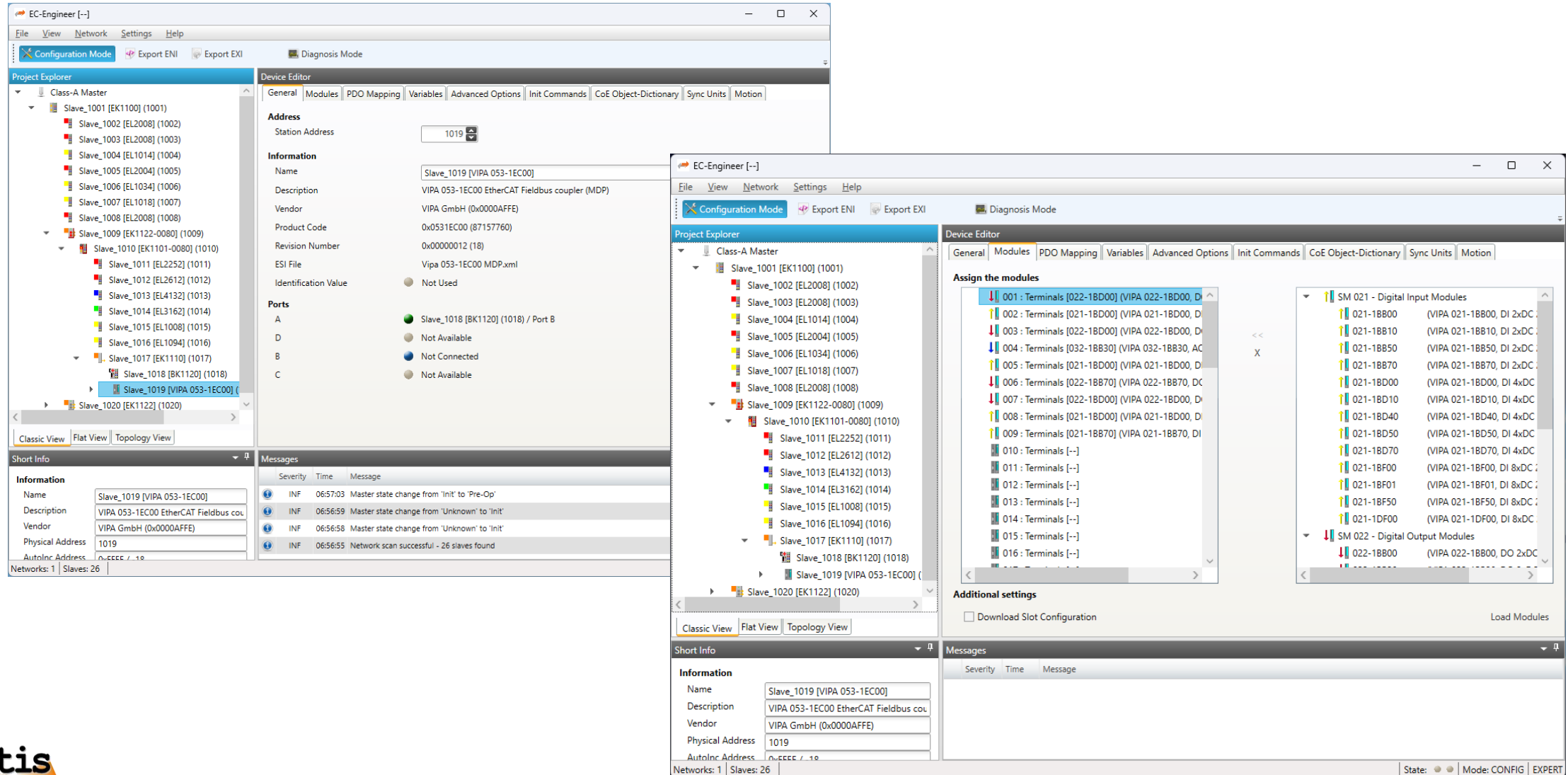
Remote **Configuration:**
Slaves connected to
Target System



Remote **Diagnosis:**
Slaves connected to
Target System



Build a configuration in less steps

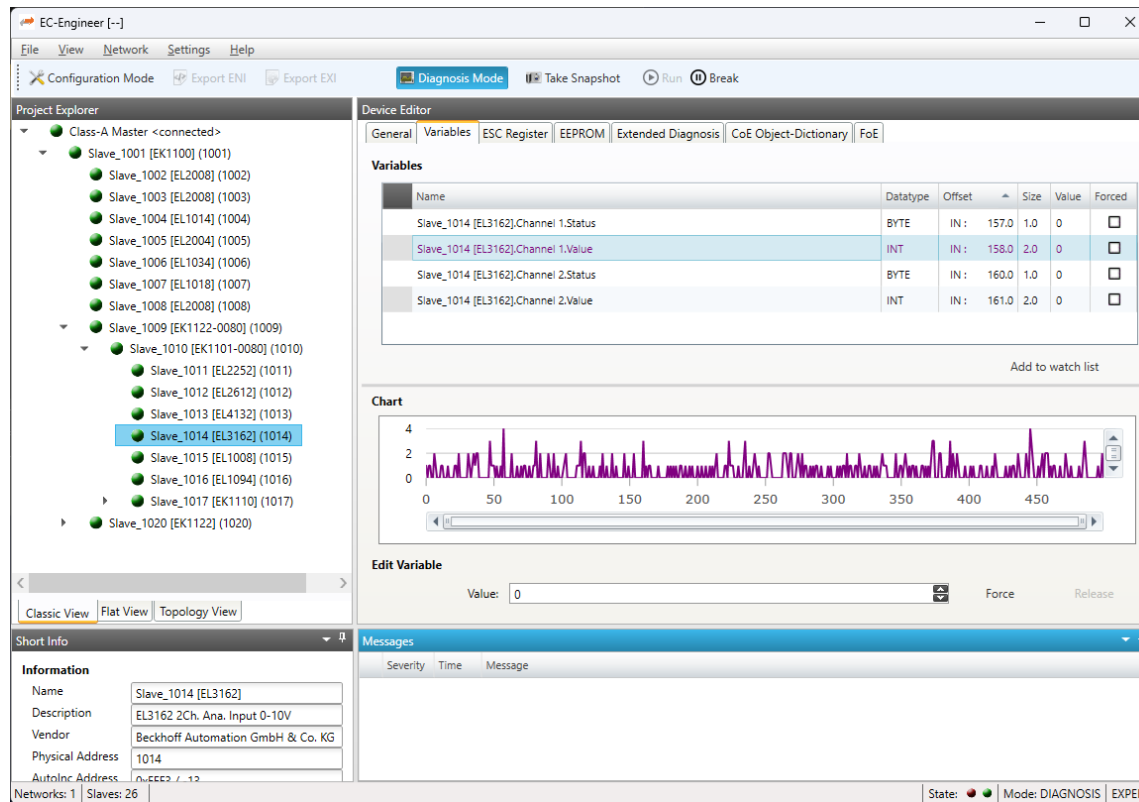


The screenshot displays the EC-Engineer software interface in Configuration Mode, showing the configuration of a Class-A Master device. The interface is divided into several panes:

- Project Explorer:** Shows a tree view of the network configuration, including a Class-A Master and 20 slave devices (Slave_1001 to Slave_1020).
- Device Editor:** Provides detailed configuration for the selected device (Slave_1019 [VIPA 053-1EC00]).
 - General:** Station Address: 1019.
 - Information:** Name: Slave_1019 [VIPA 053-1EC00], Description: VIPA 053-1EC00 EtherCAT Fieldbus coupler (MDP), Vendor: VIPA GmbH (0x0000AFFE), Product Code: 0x0531EC00 (87157760), Revision Number: 0x0000012 (18), ESI File: Vipa 053-1EC00 MDP.xml, Identification Value: Not Used.
 - Ports:** A: Slave_1018 [BK1120] (1018) / Port B; B: Not Available; C: Not Available.
- Messages:** Shows system messages such as "Master state change from 'Init' to 'Pre-Op'" and "Network scan successful - 26 slaves found".
- Assign the modules:** A list of modules to be assigned to the device, including terminals (e.g., 001: Terminals [022-1BD00] (VIPA 022-1BD00, D)) and digital input/output modules (e.g., SM 021 - Digital Input Modules).
- Additional settings:** Includes a checkbox for "Download Slot Configuration" and a "Load Modules" button.
- Short Info:** Provides a summary of the device configuration, including Name, Description, Vendor, Physical Address, and Autolnc Address.

The interface also includes a menu bar (File, View, Network, Settings, Help) and a toolbar with options like Configuration Mode, Export ENI, Export EXI, and Diagnosis Mode. The status bar at the bottom indicates "State: [] Mode: CONFIG | EXPERT".

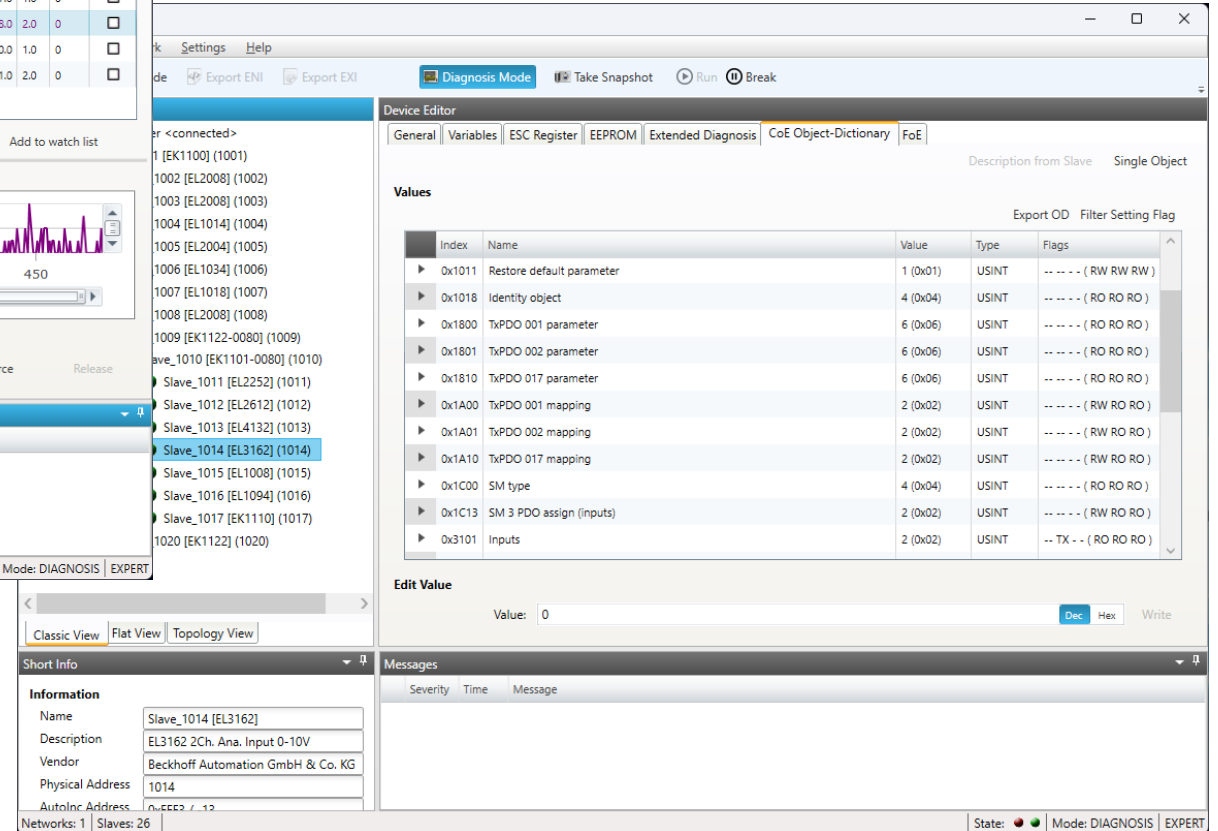
Comprehensive diagnostic: Monitor and force process data



The screenshot shows the EC-Engineer software interface in Diagnosis Mode. The Project Explorer on the left lists a hierarchy of devices, with 'Slave_1014 [EL3162] (1014)' selected. The Device Editor in the center shows the 'Variables' tab with a table of process data:

Name	Datatype	Offset	Size	Value	Forced
Slave_1014 [EL3162].Channel 1.Status	BYTE	IN : 157.0	1.0	0	<input type="checkbox"/>
Slave_1014 [EL3162].Channel 1.Value	INT	IN : 158.0	2.0	0	<input type="checkbox"/>
Slave_1014 [EL3162].Channel 2.Status	BYTE	IN : 160.0	1.0	0	<input type="checkbox"/>
Slave_1014 [EL3162].Channel 2.Value	INT	IN : 161.0	2.0	0	<input type="checkbox"/>

Below the table is a 'Chart' showing a signal waveform over time (0 to 450). The 'Edit Variable' section shows the current value is 0, with 'Force' and 'Release' buttons.

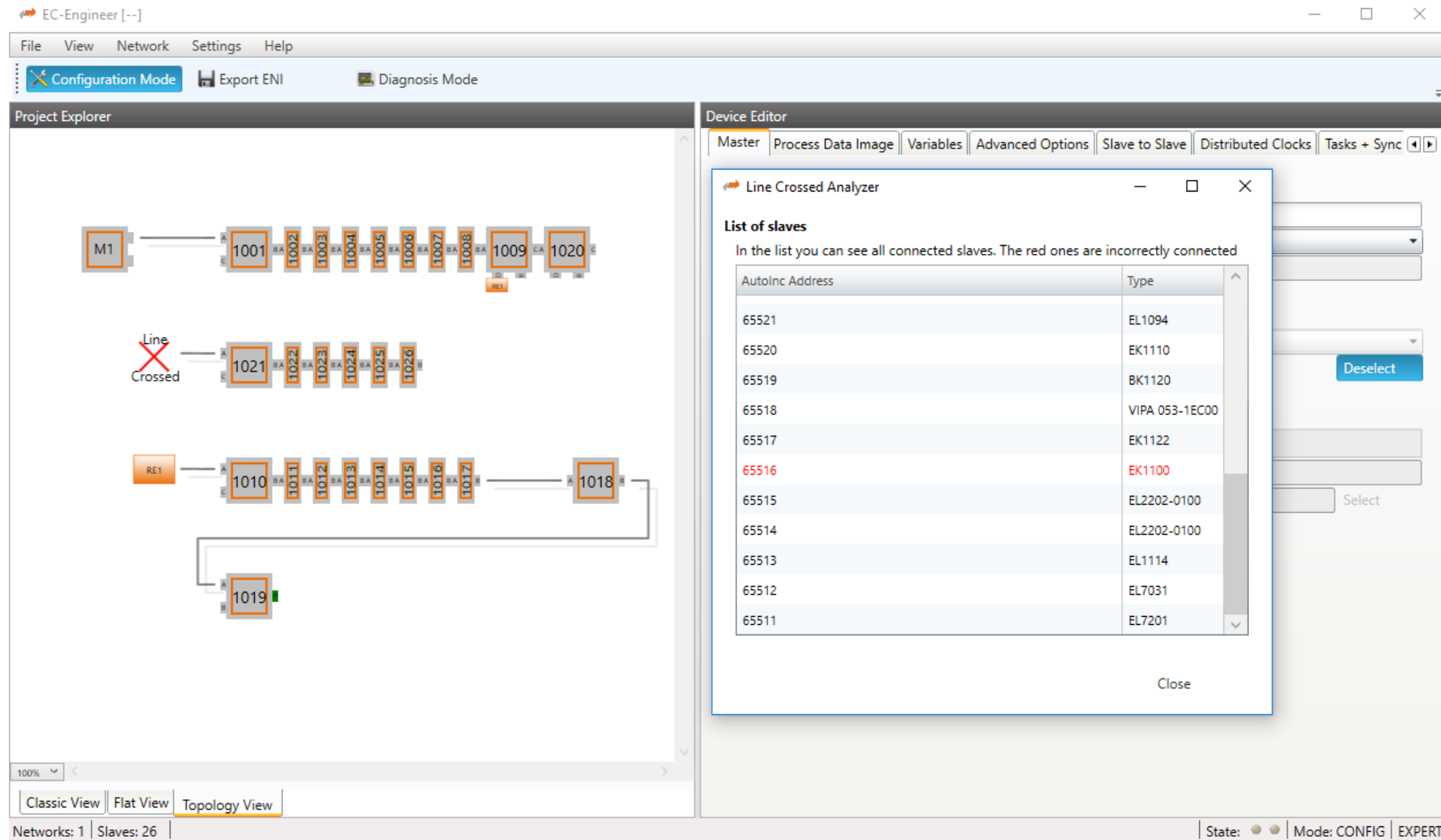


This screenshot shows the 'Values' view of the Device Editor. It displays a table of parameter values:

Index	Name	Value	Type	Flags
0x1011	Restore default parameter	1 (0x01)	USINT	... (RW RW RW)
0x1018	Identity object	4 (0x04)	USINT	... (RO RO RO)
0x1800	TxPDO 001 parameter	6 (0x06)	USINT	... (RO RO RO)
0x1801	TxPDO 002 parameter	6 (0x06)	USINT	... (RO RO RO)
0x1810	TxPDO 017 parameter	6 (0x06)	USINT	... (RO RO RO)
0x1A00	TxPDO 001 mapping	2 (0x02)	USINT	... (RW RW RW)
0x1A01	TxPDO 002 mapping	2 (0x02)	USINT	... (RW RW RW)
0x1A10	TxPDO 017 mapping	2 (0x02)	USINT	... (RW RW RW)
0x1C00	SM type	4 (0x04)	USINT	... (RO RO RO)
0x1C13	SM 3 PDO assign (Inputs)	2 (0x02)	USINT	... (RW RO RO)
0x3101	Inputs	2 (0x02)	USINT	.. TX .. (RO RO RO)

Below the table is an 'Edit Value' section with a value of 0 and buttons for 'Dec', 'Hex', and 'Write'.

Comprehensive diagnostic: Powerful “Line crossed” detection



The screenshot displays the EC-Engineer software interface. The main window shows a network topology with a master node M1 and several slave nodes (1001-1020, 1021-1026, 1010-1017, 1018, 1019). A red 'X' and the text 'Line Crossed' are positioned over the connection between nodes 1021 and 1022. The 'Device Editor' window is open, showing the 'Line Crossed Analyzer' tab. This window contains a table of slave nodes, with the entry for AutoInc Address 65516 (EK1100) highlighted in red. The status bar at the bottom indicates 'Networks: 1 | Slaves: 26' and 'Mode: CONFIG | EXPERT'.

Project Explorer

Device Editor

Master | Process Data Image | Variables | Advanced Options | Slave to Slave | Distributed Clocks | Tasks + Sync

Line Crossed Analyzer

List of slaves

In the list you can see all connected slaves. The red ones are incorrectly connected

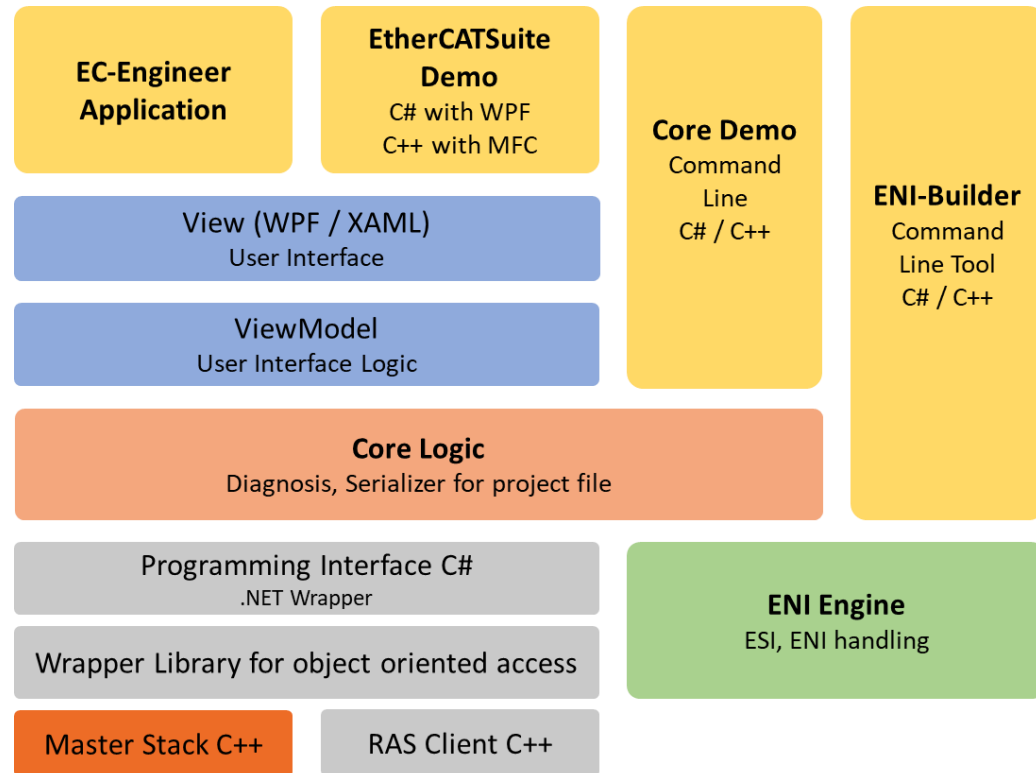
AutoInc Address	Type
65521	EL1094
65520	EK1110
65519	BK1120
65518	VIPA 053-1EC00
65517	EK1122
65516	EK1100
65515	EL2202-0100
65514	EL2202-0100
65513	EL1114
65512	EL7031
65511	EL7201

Close

Classic View | Flat View | Topology View

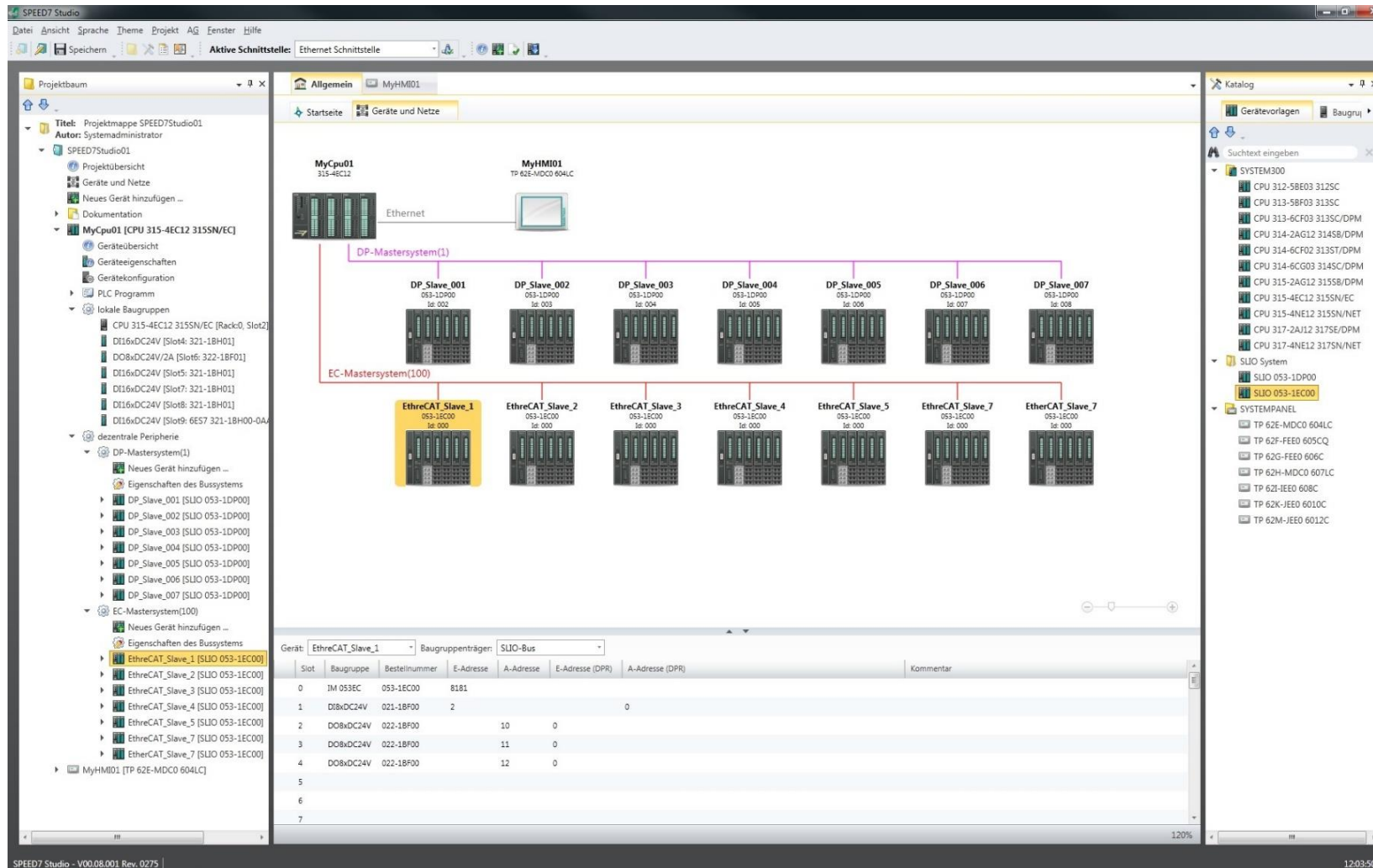
Networks: 1 | Slaves: 26 | State: ●● | Mode: CONFIG | EXPERT

EtherCAT Configuration Tool Software Development Kit (SDK)



- SDK for adding EtherCAT configuration and diagnosis features to an (existing) customer engineering tool
- Modular architecture offers various levels of integration
- Available for Windows and Linux
- Graphical User Interface (GUI) based on WPF for Windows
- Graphical User Interface (GUI) based on Web technology for Windows and Linux

Example: Integration into PLC programming tool YASKAWA SPEED7 Studio



The screenshot displays the YASKAWA SPEED7 Studio interface. The main window shows a network topology diagram with a central 'MyCpu01' (CPU 315-4EC12) connected via Ethernet to 'MyHM01' (TP 62E-MDC0 604LC). Below this, a 'DP-Mastersystem(1)' is connected to seven 'DP_Slave' units (001-007). An 'EC-Mastersystem(100)' is connected to seven 'EtherCAT_Slave' units (000-006, 007). The left sidebar shows the project tree, and the right sidebar shows the component catalog. At the bottom, a table provides detailed parameters for the selected 'EtherCAT_Slave_1'.

Slot	Baugruppe	Bestellnummer	E-Adresse	A-Adresse	E-Adresse (DPR)	A-Adresse (DPR)	Kommentar
0	IM 053EC	053-1EC00	8181				
1	DI8xDC24V	021-1BF00	2		0		
2	DO8xDC24V	022-1BF00		10	0		
3	DO8xDC24V	022-1BF00		11	0		
4	DO8xDC24V	022-1BF00		12	0		
5							
6							
7							

- **Easy** to use - **modern** design
 - ⇒ Build a configuration in less steps
 - ⇒ Only reasonable settings and options are visible, expert settings visible only if required
- **Powerful online functions** together with EC-Master
 - ⇒ Network scan local & remote, compare configured and found slaves (network mismatch view)
 - ⇒ Access to states, variables, object dictionary, ESC register, EEPROM, etc.
 - ⇒ Figure out the location of communication errors (bad cables and connectors, vibrations, etc.)
- **EtherCAT Master Information** (EMI) file for specifying master device features
 - ⇒ The configuration tool offers only supported features of the selected EtherCAT controller
 - ⇒ E.g. maximum number of EtherCAT SubDevices or process data size or cycle time limitations
- **Software Development Kit** available
 - ⇒ Adjust to customer needs or integrate into customer engineering environment

- EC-Engineer tutorials and user manual
<https://developer.acontis.com/ec-engineer>
- Request for EC-Engineer technical details slides
- Request for evaluation software
<https://www.acontis.com/en/ethernetcat-support-eval-request.html>

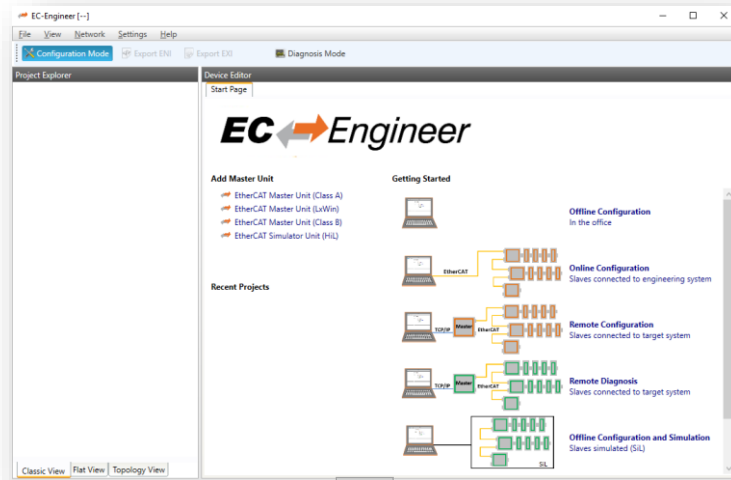
EC  **Engineer** **Web**

Web Edition

EtherCAT Configuration and Diagnosis using a Browser

EC-Engineer vs EC-EngineerWeb

EC Engineer

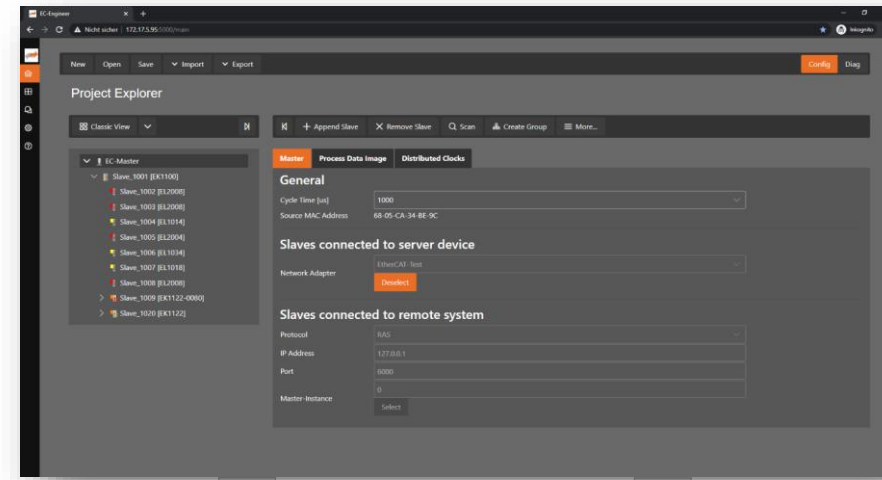


Windows

x86 32-Bit

x64 64-Bit

EC EngineerWeb



Windows

x64 64-Bit

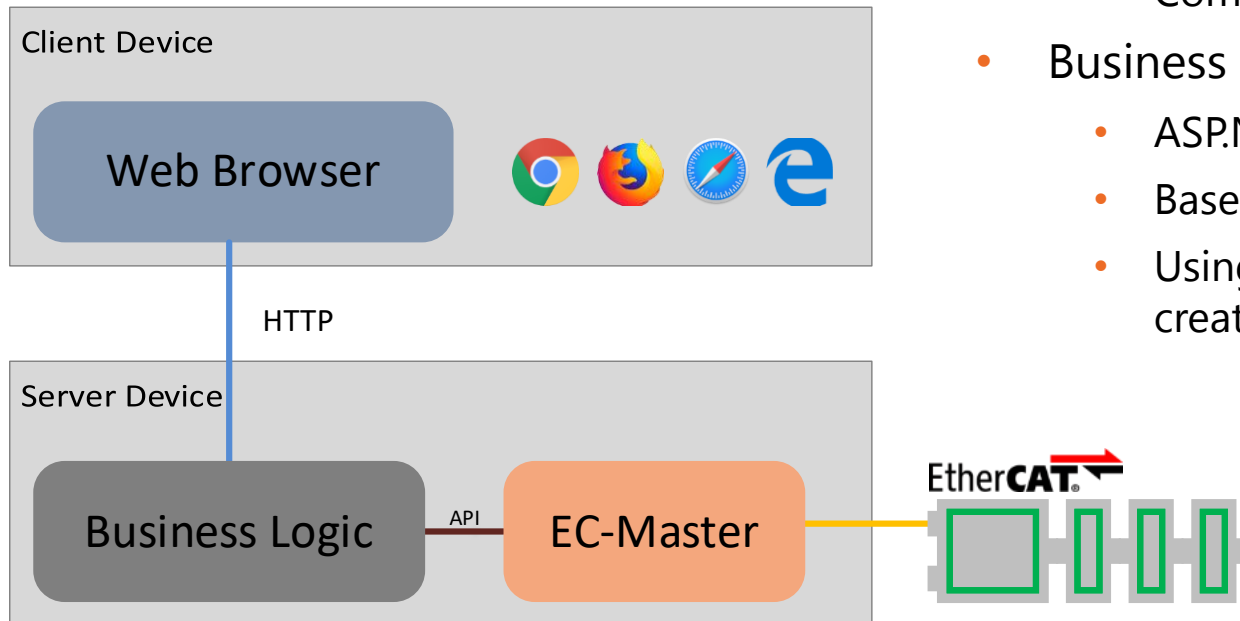
Linux

ARM 32-Bit
ARM 64-Bit

EC-Engineer Web Overview

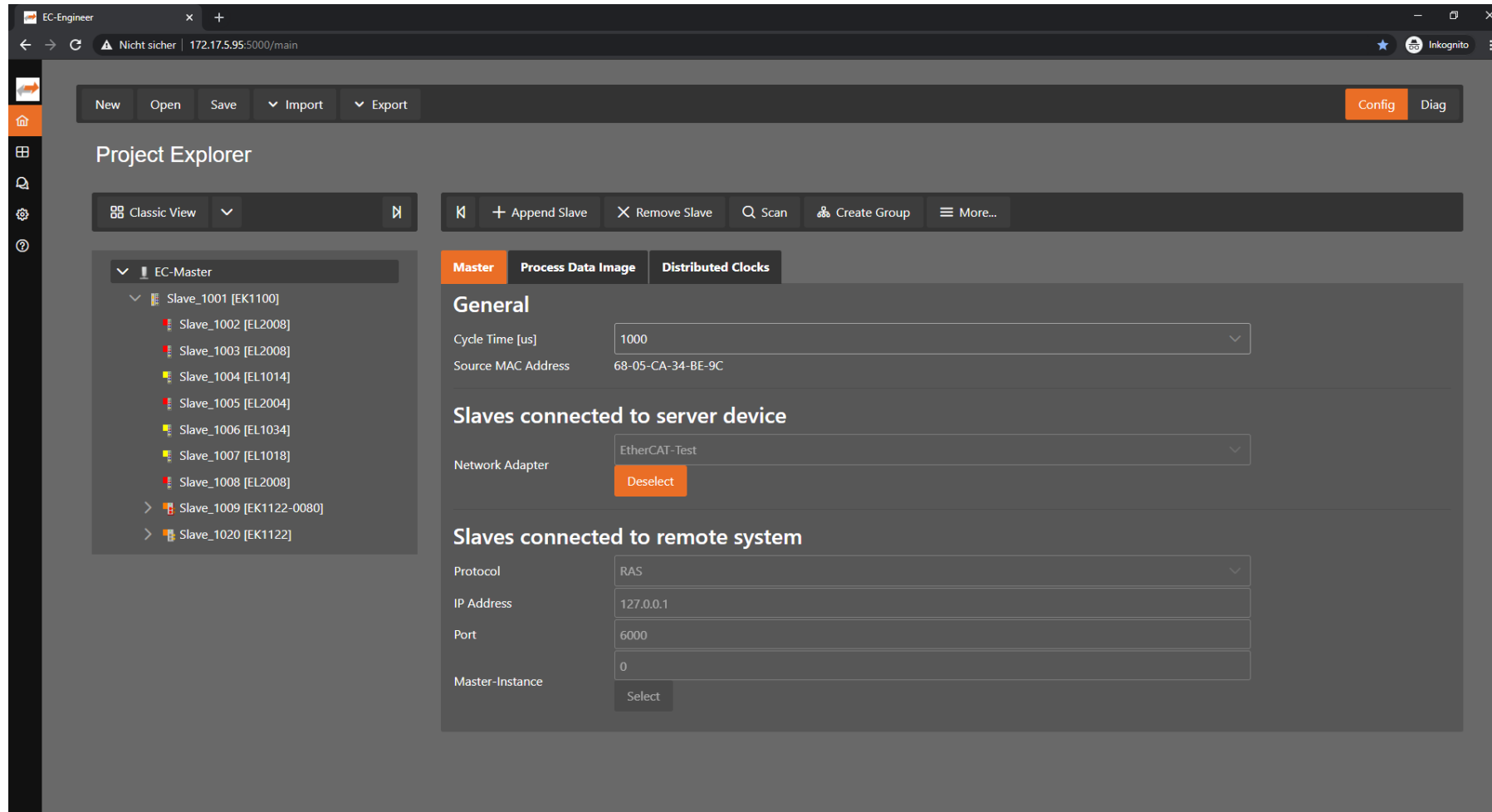
EtherCAT configuration and diagnosis tool using a standard browser as user interface:

- Web Browser with HTML5 and JavaScript support
 - Desktop computer
 - Mobile devices (tablet, smartphone)
 - Communication to backend via HTTP
- Business logic (RACE)
 - ASP.NET Core Web Application
 - Based on Microsoft .NET Core 3.1
 - Using the same, well proven core logic and ENI creation algorithm as EC-Engineer

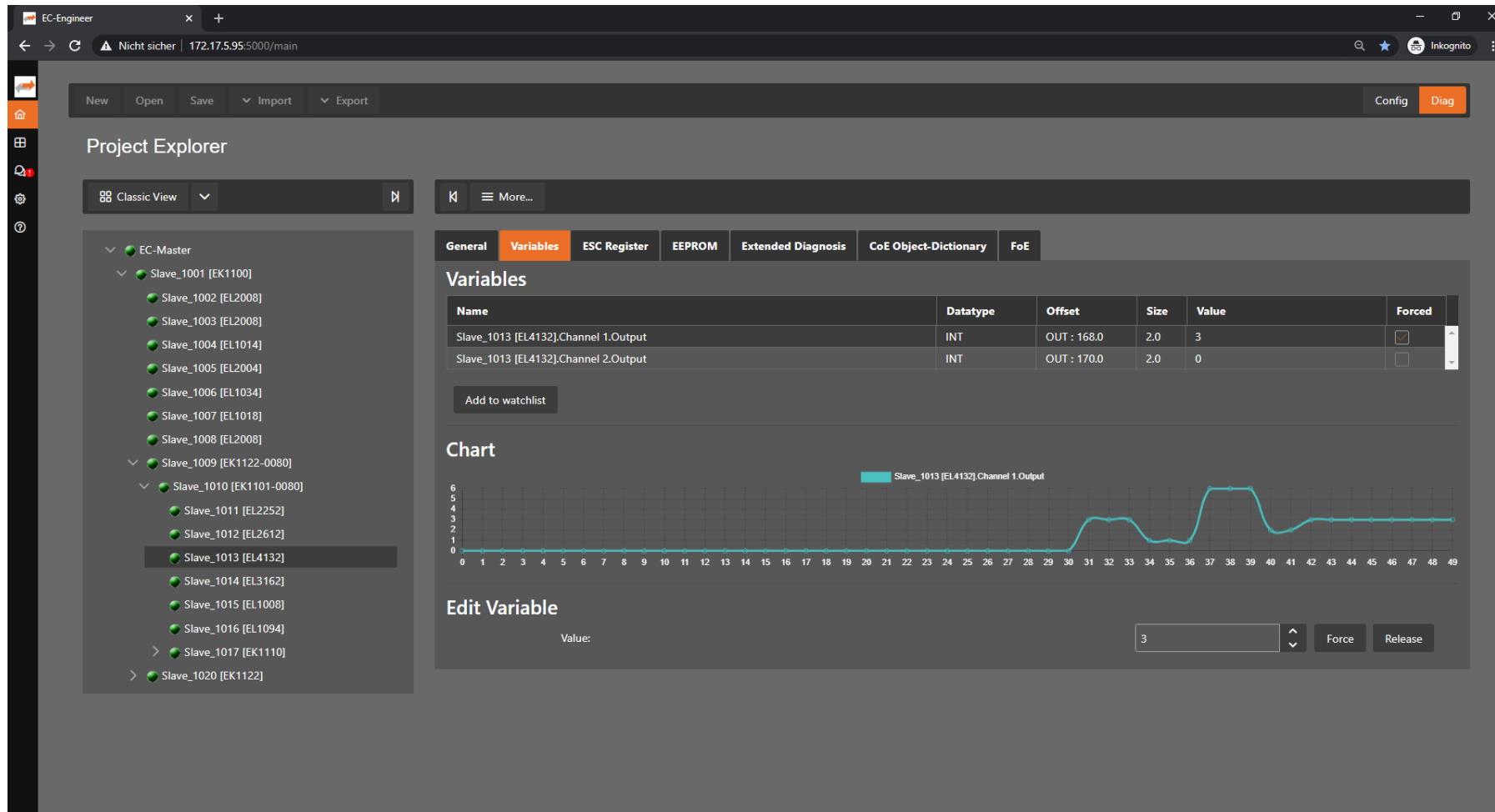


Configuration Mode

Full features including MDP etc.



Sophisticated Diagnosis Features Monitoring and forcing of Variables etc.



The screenshot displays the EC-Engineer software interface. On the left is the Project Explorer showing a tree structure of nodes including EC-Master and various Slave nodes (Slave_1001 to Slave_1020). The main area is divided into several sections:

- General** tab is selected, showing the **Variables** section with a table of monitored variables.
- Chart** section displays a line graph for the variable "Slave_1013 [EL4132] Channel 1 Output".
- Edit Variable** section allows for manual input of the variable's value, currently set to 3, with "Force" and "Release" buttons.

Name	Datatype	Offset	Size	Value	Forced
Slave_1013 [EL4132].Channel 1.Output	INT	OUT : 168.0	2.0	3	<input checked="" type="checkbox"/>
Slave_1013 [EL4132].Channel 2.Output	INT	OUT : 170.0	2.0	0	<input type="checkbox"/>

The chart shows a signal that remains at 0 until approximately x=30, then rises to a peak of 6 at x=38, and then fluctuates between 2 and 4 for the remainder of the time period shown.

- EC-Engineer Web user manual
<https://developer.acontis.com/ec-engineer>
- Request for EC-Engineer Web technical details slides
- Request for evaluation software
<https://www.acontis.com/en/ethernetcat-support-eval-request.html>

EC  *Simulator*

EtherCAT® Network Simulation

Run an EtherCAT controller with a simulated network

Master Software Development



Run the Master application without real slaves
Comfortable Debugging of complex topology/slave type scenarios

Fieldbus "Master-Device" Test



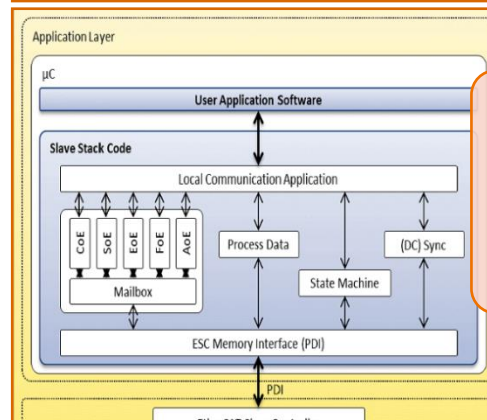
Replace manual test procedures by automatic procedures.
Implement enhanced tests

Virtual Commissioning



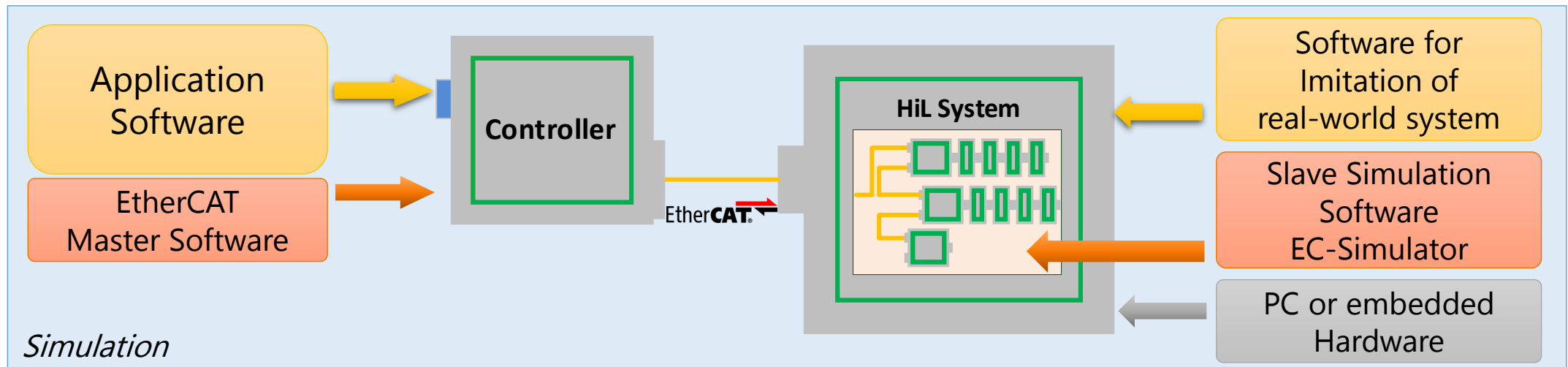
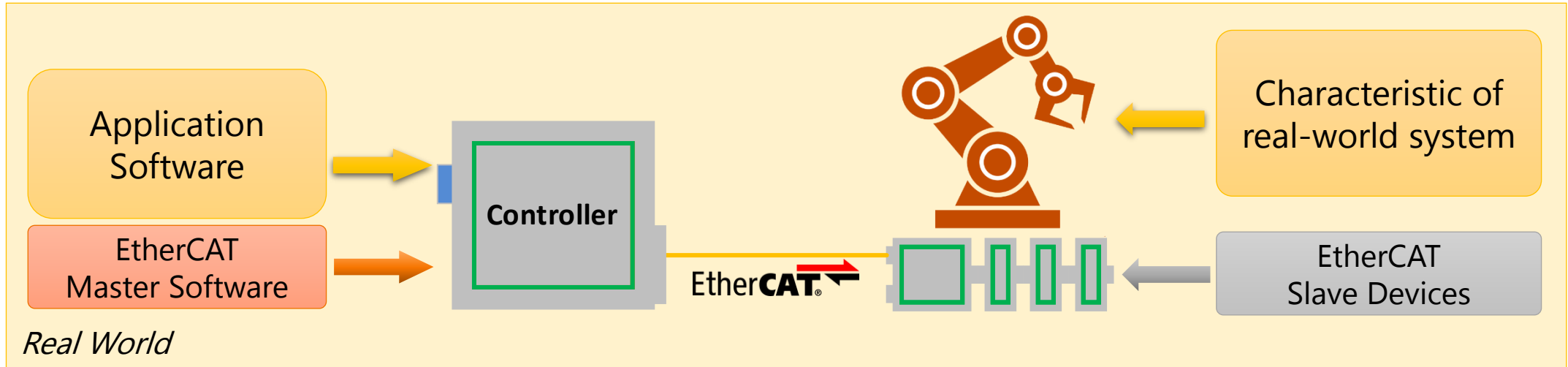
Develop and test applications based on a software emulation of the machine logic

Slave Firmware Development



Develop slave firmware before hardware is available

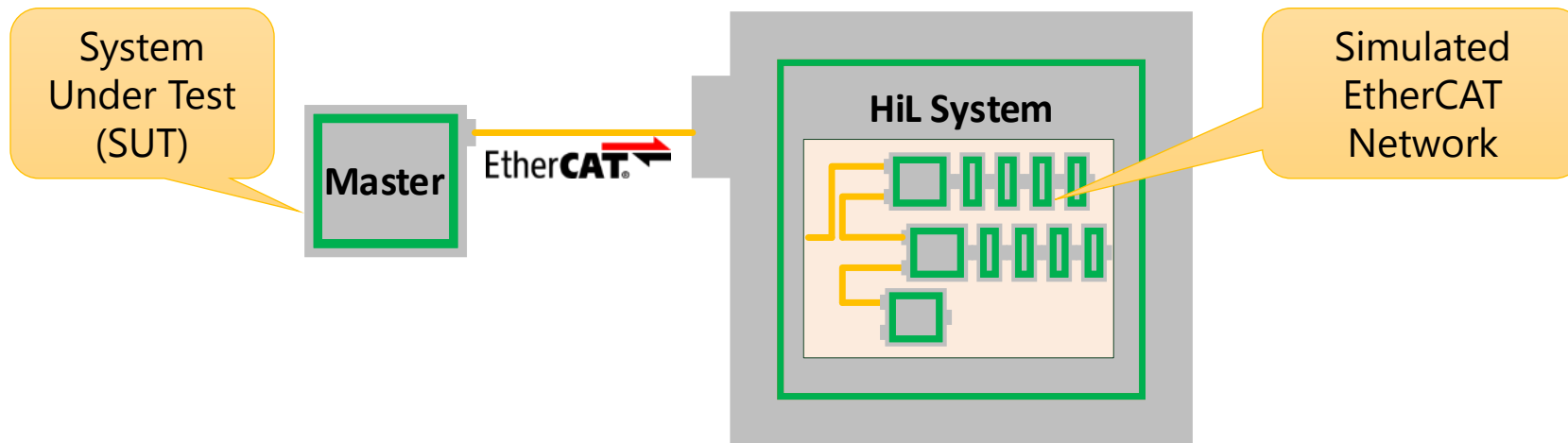
General System Architecture



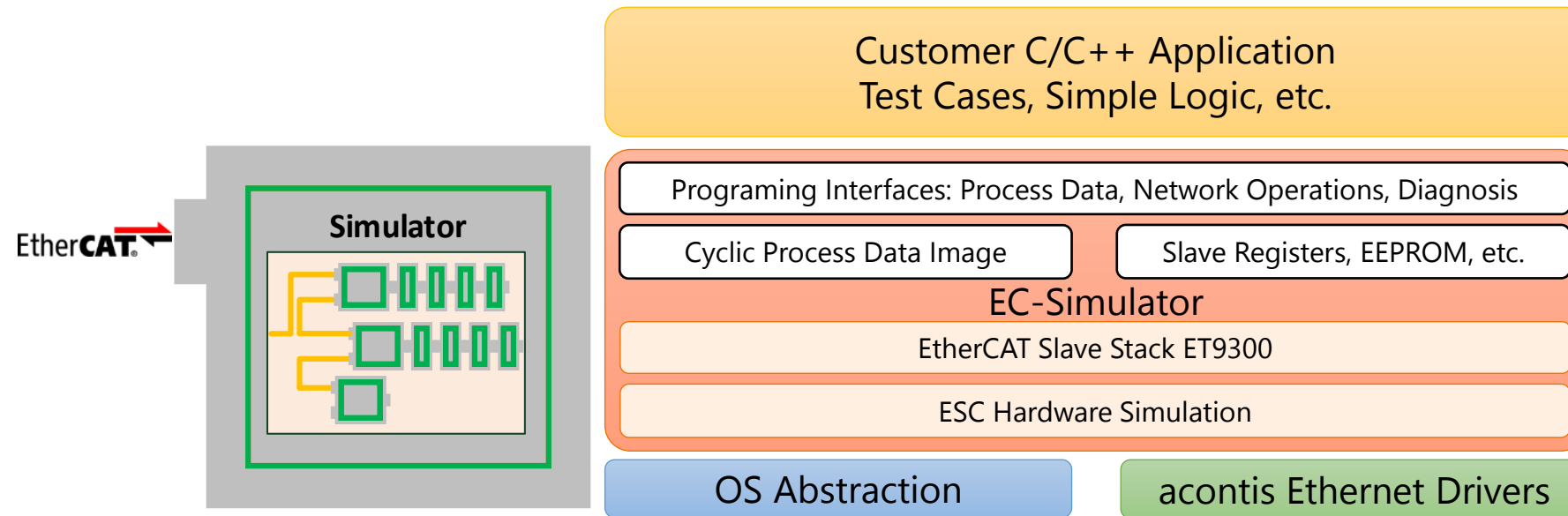
Hardware-in-Loop (HiL) Simulation

All slaves are simulated

- The System-Under-Test (SUT) is communicating via an EtherCAT cable with the EC-Simulator software running on an external hardware, the HiL System.
- HiL System doesn't require special hardware, a standard Ethernet interface can be used
- In this setup the original application together with the EtherCAT Master can be tested



HiL Simulation: Software architecture



- Support for various operating systems, e. g. Linux and Windows
- High performance due to the acontis Real-time Ethernet drivers
- Simulation of EtherCAT Slave Controller (ESC) hardware in software
- Implementation of Beckhoff EtherCAT Slave Stack ET9300
- EC-Simulator offers a "C" language application programming interface (API)

Use Case: Virtual Commissioning

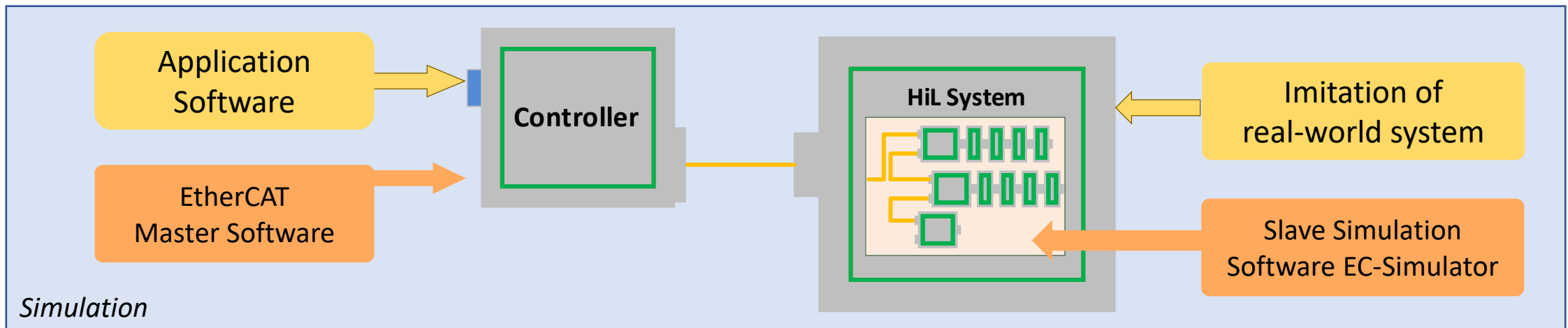
EC ↔ Simulator

- EC-Simulator together with Digital-Twin Software
- Test and optimize the application during early engineering-stages, without real existing target hardware.
- Test error scenarios which are dangerous and/or lead to damages



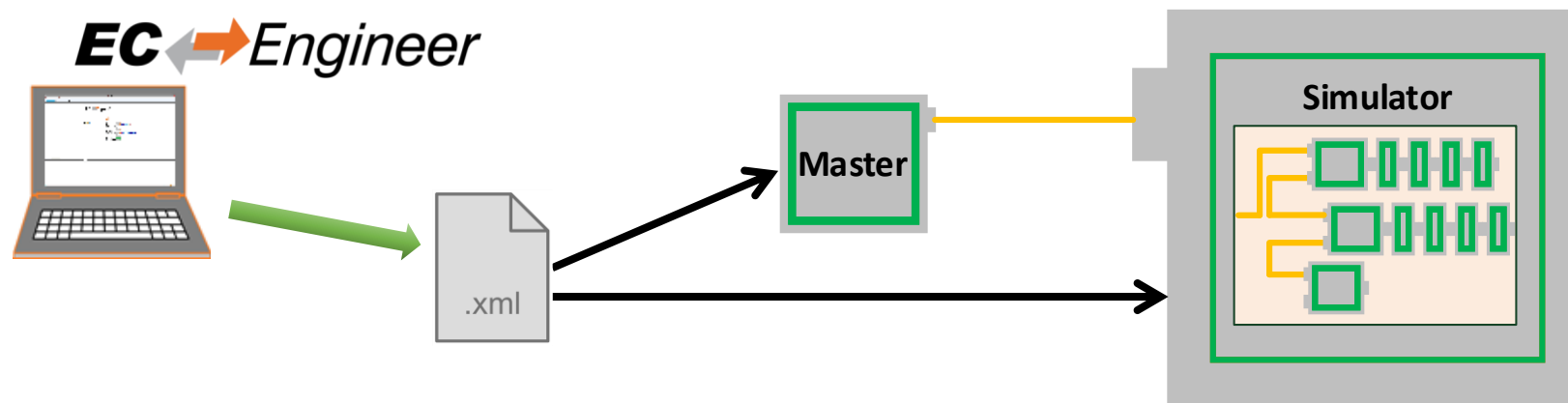
Third Party Software for
Digital Twin, Virtual Commissioning

EC ↔ Simulator

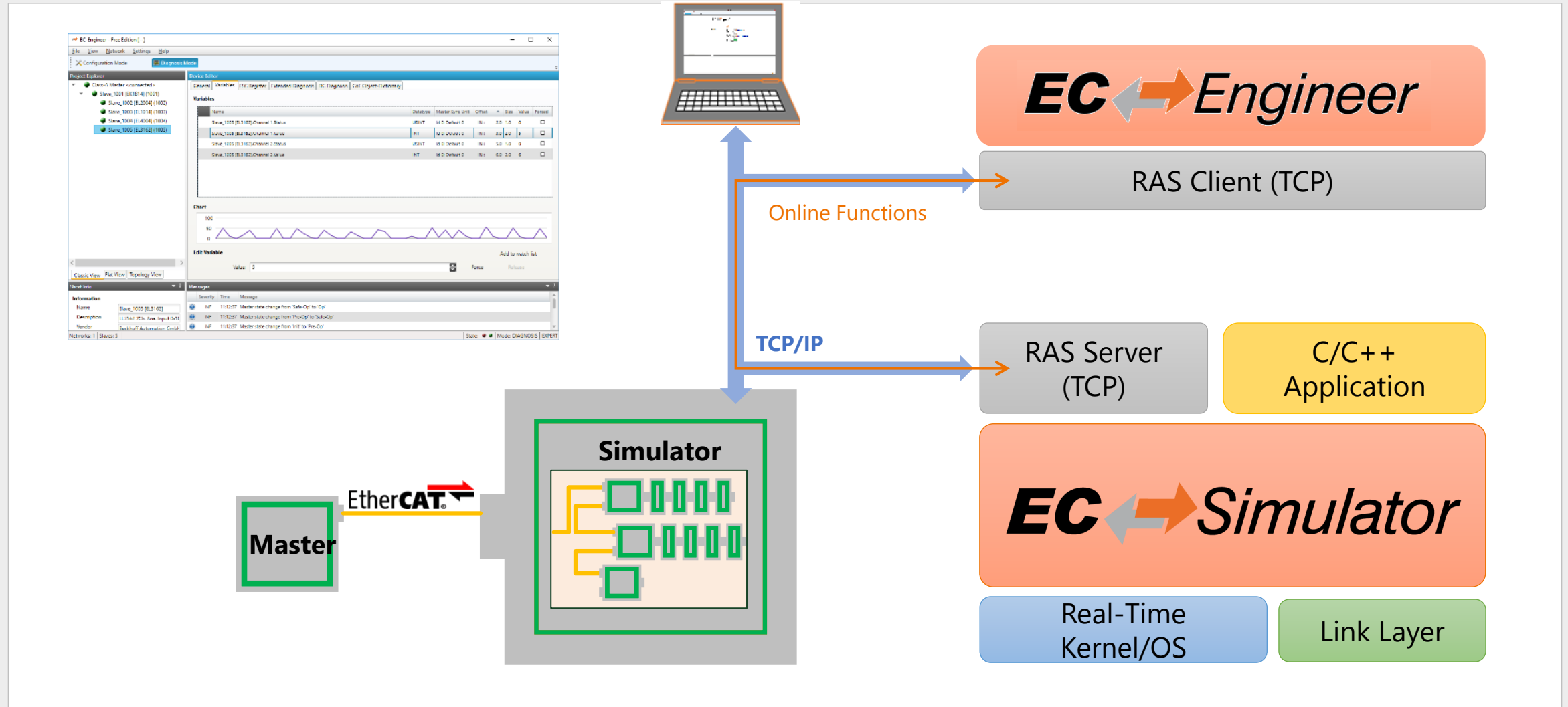


Configuration based on ENI file

- Standard EtherCAT Network Information (ENI) file is used to configure the EC-Simulator software
- ENI can be exported by EC-Engineer or any other configuration software, e. g. Beckhoff TwinCAT, supporting it.



Diagnosis with EC-Engineer



Simulate Power Loss, Link Loss and Frame Errors with EC-Engineer

Device Editor

General Variables ESC Register EEPROM Extended Diagnosis DC Diagnosis CoE Object-Dictionary FoE Simulator

State Machine

Current State:

Slave Power

Change the power condition of the slave. After a power cycle the slave is in INIT.

Power off Power on

Slave Connection

Change slave connection. Unplug or change connection to previous slave. Does not power off the slave.

Connect to Slave Address: Port:

Disconnect Connect

CRC Error

Generate a CRC error at a specific port (once or probability).

Port: Probability (%):

Set once Set random Reset random

Link Loss

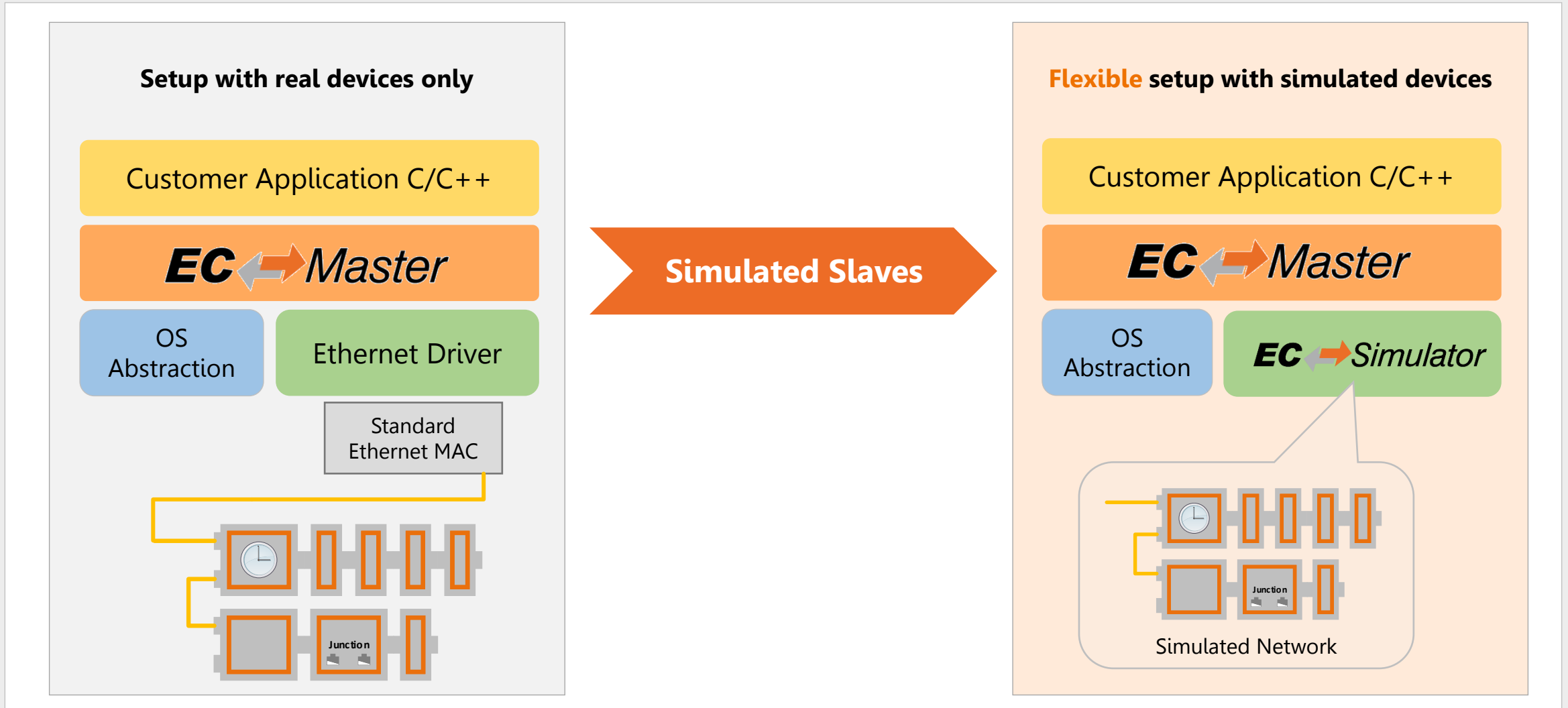
Generate a Link Loss at a specific port for a specific time (once or probability).

Port: Down time (s): Probability (%):

Set once Set random Reset all ports

- Power down/up a specific slave
- Create CRC errors on a specific port (by random)
- Simulate test cases like “Unplugging a cable”

Software-in-Loop (SiL) Simulation



- **Hardware-in-the-Loop:**
 - ⇒ Simulate a complete EtherCAT network on a PC or embedded system running Windows, Linux, QNX, etc.
- **Software-in-the-Loop:**
 - ⇒ Simulate a complete EtherCAT network on the EC-Master controller
- Comprehensive functions to **simulate errors**
 - ⇒ Including broken cable, wrong cabling, slave failures
- EC-Simulator **API** is mainly **equal** to EC-Master API
- Integration of **own slave firmware** possible

- EC-Simulator product intro video
https://youtu.be/5ToJh7gJ_Go
- EC-Simulator tutorials and user manual
<https://developer.acontis.com/ec-simulator>
- Request for EC-Simulator technical details slides
- Request for evaluation software
<https://www.acontis.com/en/ethercat-support-eval-request.html>