Rogue Robots

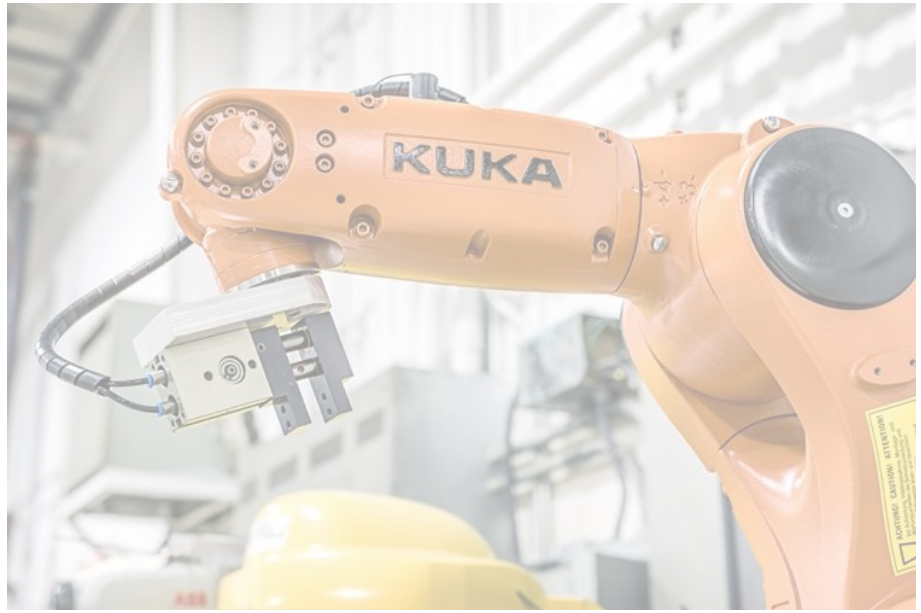
Violating the three laws of robotics

Davide Quarta, Marcello Pogliani, Mario Polino,
Andrea Maria Zanchettin, Federico Maggi, **Stefano Zanero**

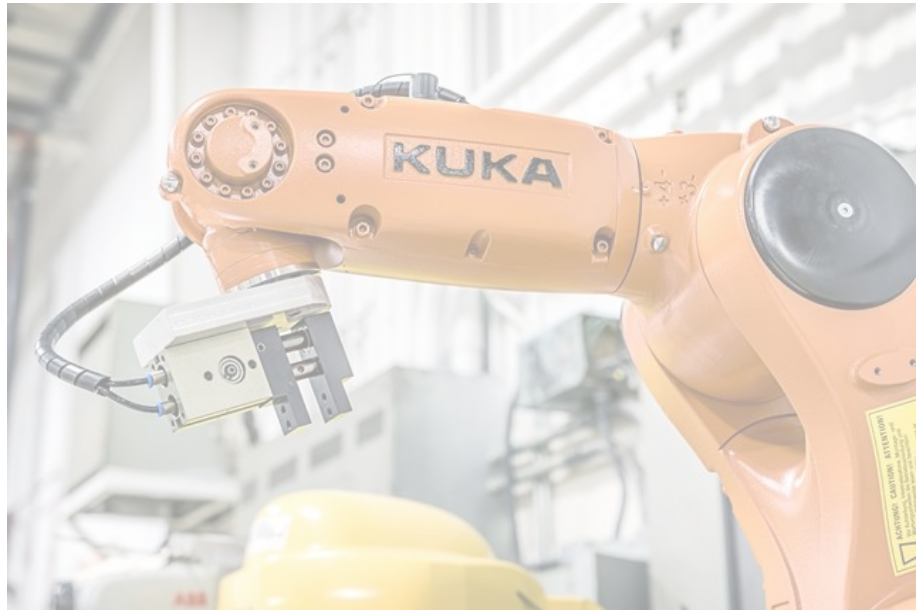
What are Industrial Robots?

What are Industrial Robots?

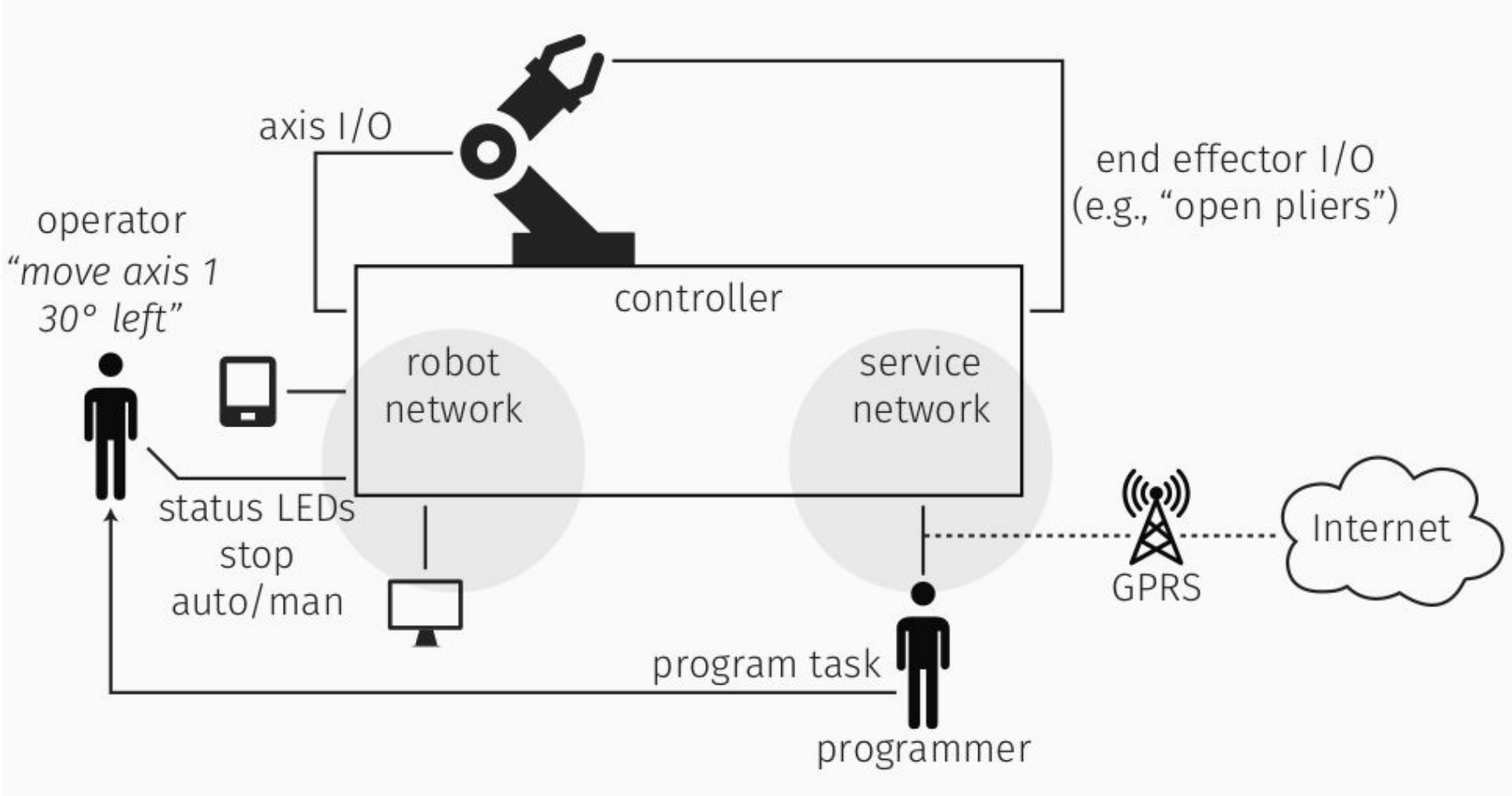
What are Industrial Robots?



What are Industrial Robots?



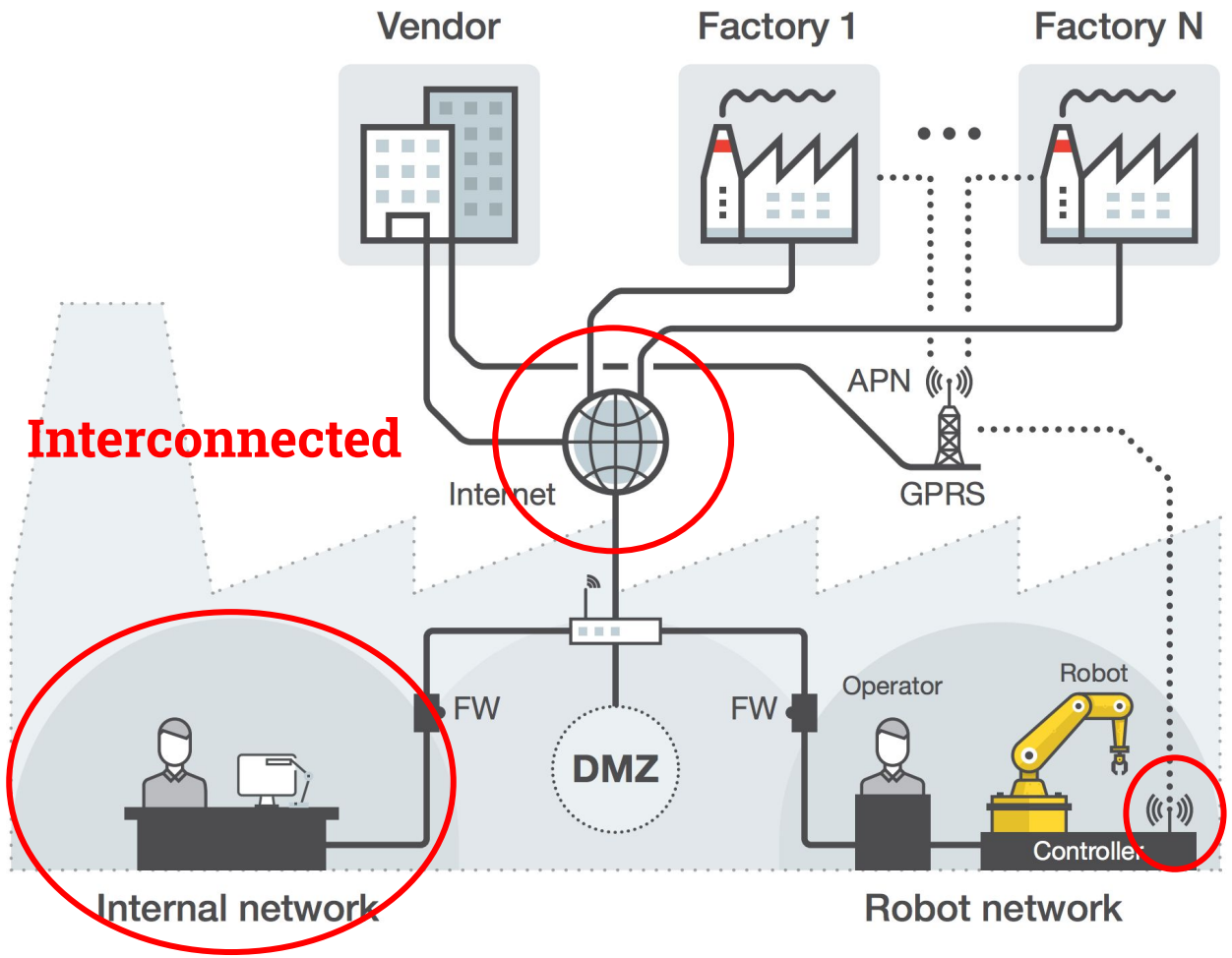
What are Industrial Robots?



What are Industrial Robots?

"automatically controlled, **re-programmable**, **multipurpose** manipulator programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications" (ISO 8373)

The “Smart” Factory



Awareness (or lack thereof)

Survey: Robot users vs. system security

50 domain experts—users interviewed: 20 answers

- 28%* access control policies not enforced
- 30% robots accessible over Internet
- 76% never performed a pentest
- > 50% not a realistic threat

* some users did not answer all the questions

Connected Robots

Connected Robots

- **Now:** mainly monitoring & maintenance (ISO 10218-2:2011)
- **Near future:** active production planning and control
  - ABB, and other vendors, expose REST APIs
  - … up to the use of mobile devices to issue commands
- **Future:** app stores (“industrial” robotappstore.com?)

Industrial Robots are *already* meant to be connected

17.4 Sending PDL2 commands via e-mail

The user is allowed to send PDL2 commands to the C4G Controller Unit, via e-mail. To do that, the required command is to be inserted in the e-mail title with the prefix 'CL' and the same syntax of the strings specified in SYS_CALL built-in. Example: if the required

.fm

l Functionality

command is ConfigureControllerRestartCold, the
the e-mail title: 'CL CCRC'.

The authentication is performed by inserting a tex
c4gmp program (on a PC), in the message bod
system identifier ($BOARD_DATA[1].SYS_ID), th
the required command, the user login and pass
inserted into the message body, and it will work a

17.3 Sending/receiving e-mails on C4G Controller

A PDL2 program called "email" is shown below ("email" program): it allows to send and receive e-mails on C4G Controller.

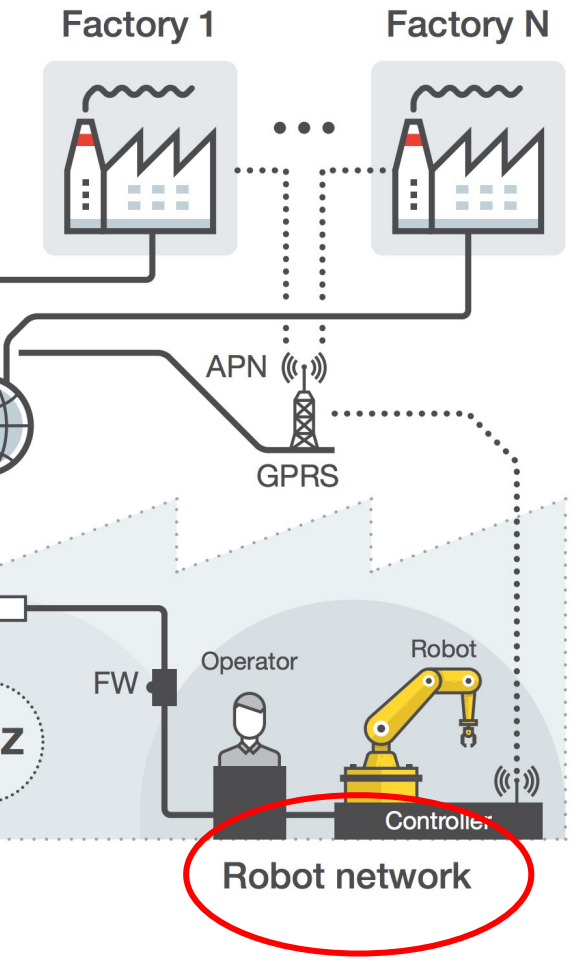
DV4_CNTRL Built-In Procedure is to be used to handle such functionalities.

See DV4_CNTRL Built-In Procedure in Chap. BUILT-IN Routines List section for further information about the e-mail functionality parameters.

17.3.1 "email" program

```
PROGRAM email NOHOLD, STACK = 10000
CONST ki_email_cnfg = 20
  ki_email_send = 21
  ki_email_num = 22
```

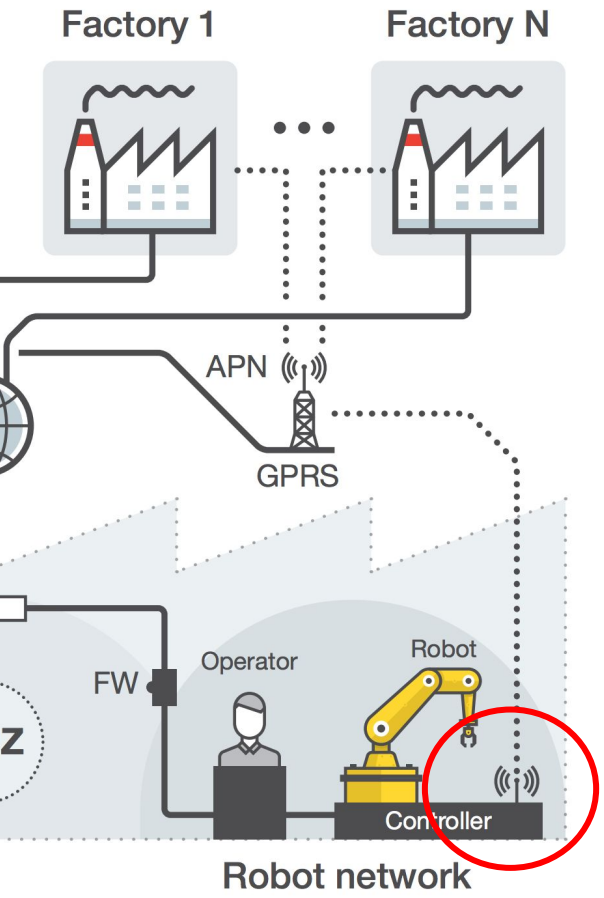
Remote Exposure Robots



Search	Entries	Country
ABB Robotics	5	DK, SE
FANUC FTP	9	US,KR,FR, TW
Yaskawa	9	CA,JP
Kawasaki E Controller	4	DE
Mitsubishi FTP	1	ID
Overall	**28**	**10**

Not so many! Maybe we're not screwed!

Remote Exposure Industrial Routers



Brand	Exposed Devices	No Authentication
Belden	956	
Eurotech	160	
eWON	6,219	1,160
Digi	1,200	
InHand	883	
Moxa	12,222	2,300
NetModule	886	135
Robustel	4,491	
Sierra Wireless	50,341	220
Virtual Access	209	
Welotec	25	
Westermo	6,081	1,200
TOTAL	83,673	5,105

Finding exposed routers is kind of expected, right?

(maybe not with authentication disabled, but let's put this aside for a moment)

Vulnerable industrial routers

Information Disclosure

- Verbose banners (beyond brand or model's name)
- Detailed technical material on vendor's website
  - Technical manual: All vendors inspected
  - Firmware: 7/12 vendors

Outdated Everything

- Kernel, compilers and libraries
- e.g., OpenVPN 2.0.5 (2004+) built on Dec 19 2016
- e.g., OpenSSL 0.9.7d (2004)

Vulnerable industrial routers

Weak/known/static credentials

- Null/blank default passwords (available on manuals)
- Weak hashing functions for passwords (easily cracked)
- OpenVPN with static key

Insecure Web Interface

- No input sanitization
- Critical system? Let's just copy paste code from a blog!

Bottom line

Robots and industry-grade networking equipment are remotely accessible, and vulnerable

What could possibly go wrong?

Let’s dig a little bit deeper

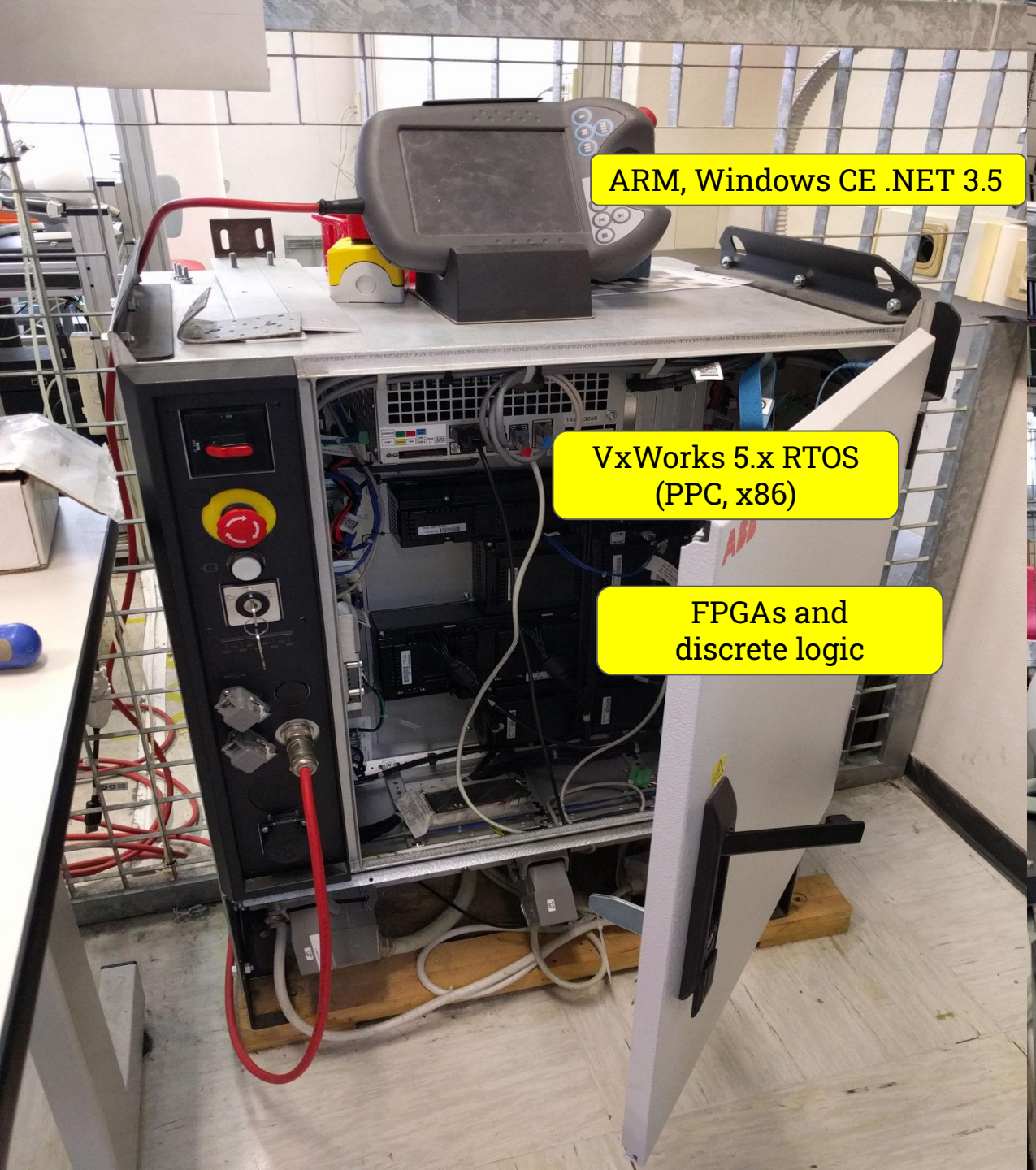
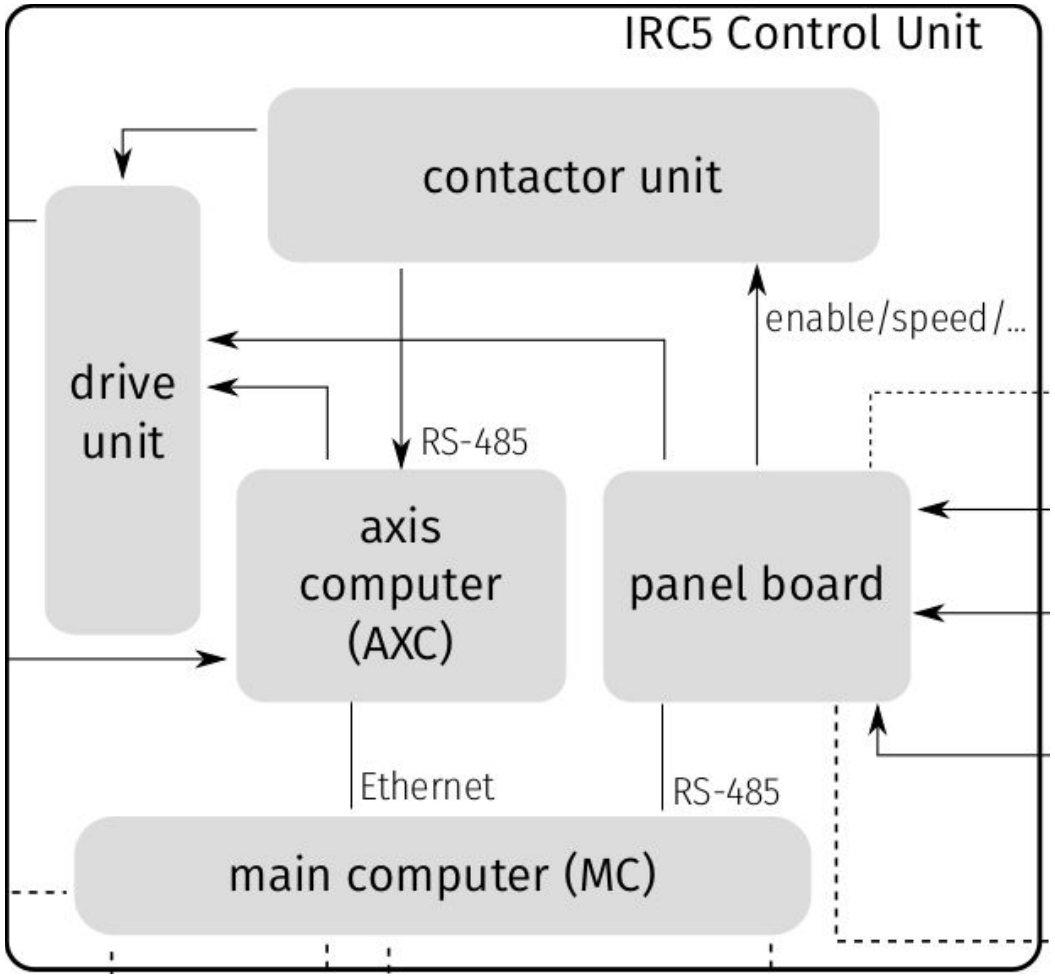
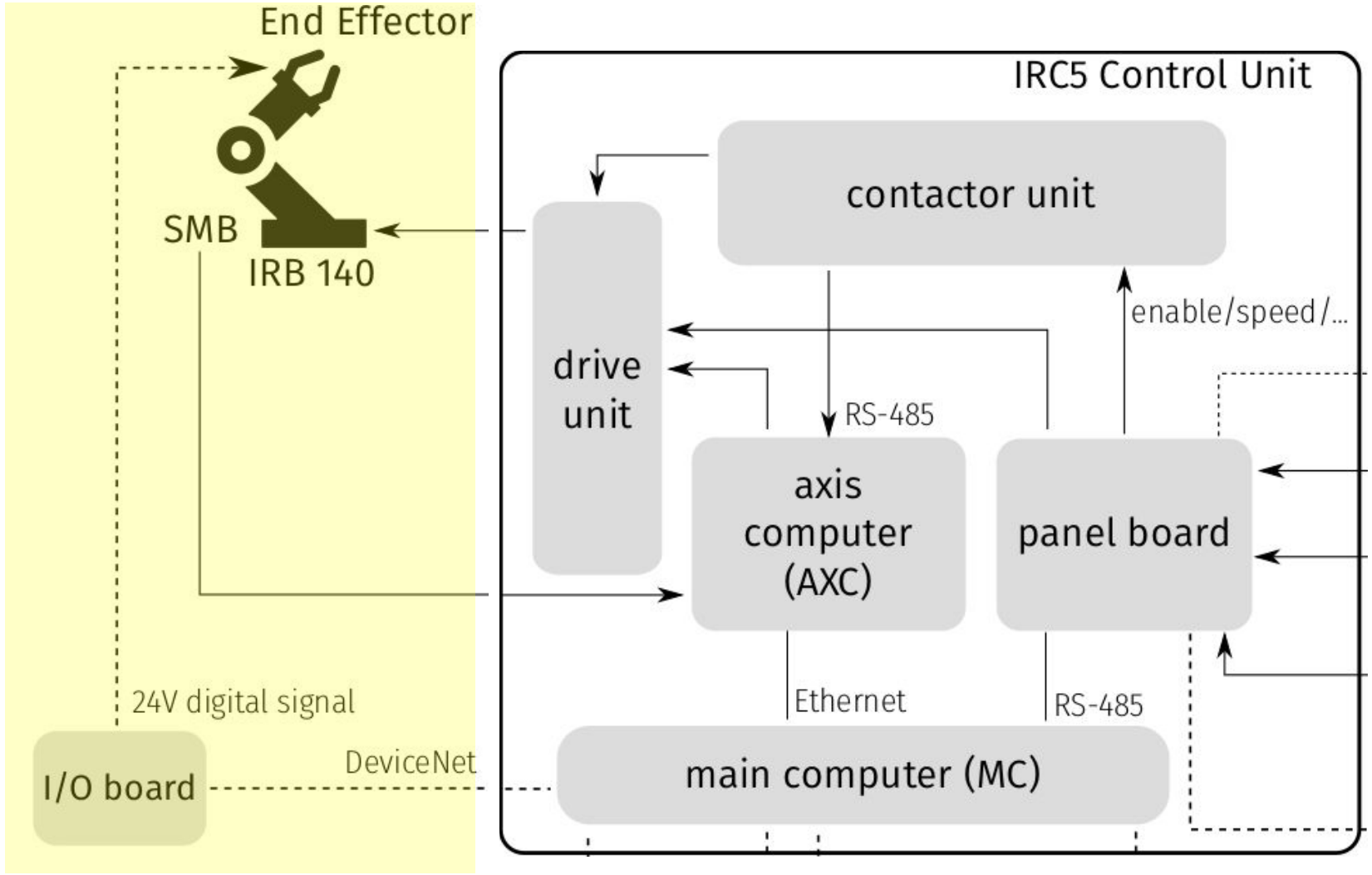

ARM, Windows CE .NET 3.5
VxWorks 5.x RTOS
(PPC, x86)
FPGAs and
discrete logic


ABB
ABB


IRC5 Control Unit
contactor unit
drive unit
enable/speed/...
RS-485
axis computer (AXC)
panel board
Ethernet
RS-485
main computer (MC)


End Effector
SMB
IRB 140
IRC5 Control Unit
contactor unit
drive unit
enable/speed/...
RS-485
axis computer (AXC)
panel board
Ethernet
RS-485
24V digital signal
I/O board
DeviceNet
main computer (MC)

End Effector

SMB

IRB 140

IRC5 Control Unit

contactor unit

drive unit

RS-485

enable/speed/...

axis computer (AXC)

panel board

RS-485

24V DS

e-stop

24V DS

motors on/off

24V DS

auto / manual

Ethernet

RS-485

24V digital signal

I/O board

DeviceNet

main computer (MC)

FlexPendant

Ethernet

Ethernet (LAN port)

Ethernet (service port)

factory LAN

service LAN

RS232

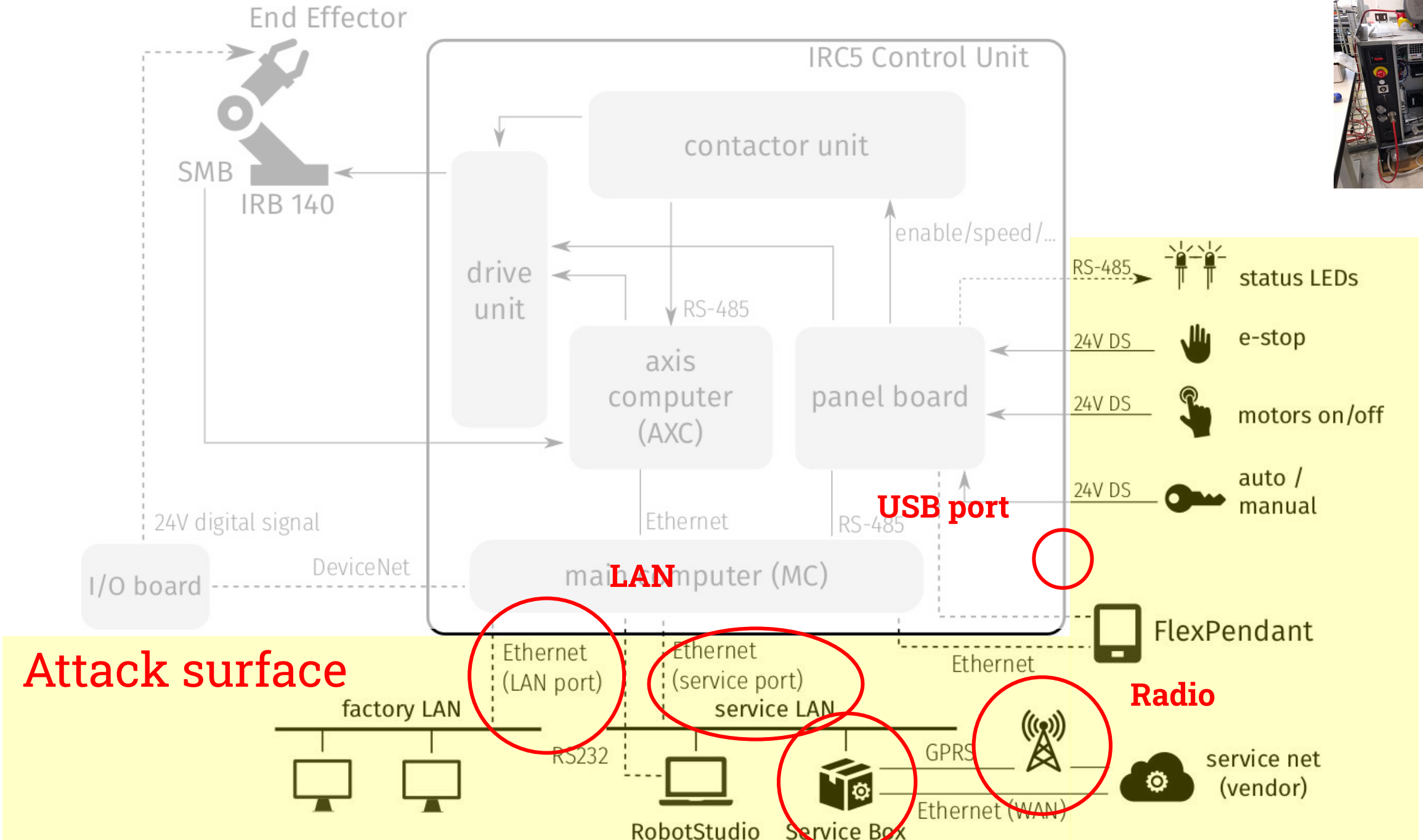
GPRS

service net (vendor)

Ethernet (WAN)

RobotStudio

Service Box



Attack surface

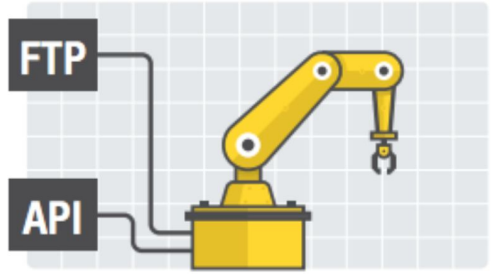
Vulnerabilities

- **BOF leading to RCE** ABBVU-DMRO-124641
- **BOF in FlexPendant** ABBVU-DMRO-124645
- **BOF in /command endpoint** ABBVU-DMRO-128238
- **Command Injection** ABBVU-DMRO-124642
- **Authentication bypass** ABBVU-DMRO-124644

Full Controller Exploitation

1. Using static credentials
   FTP PUT /command/command.cmd

1. (Alternatively) DHROOT RobAPI request
   (no auth) with buffer overflow exploit



2. FTP PUT /command/command.cmd
   script: “shell-uas_disable”

→ AUTH is now disabled

3. FTP PUT malice.dll

→ FP/MC will load malicious library at next boot

4. FTP PUT /command/command.cmd
   script: “shell reboot”

→ FP/MC will reboot

5. malice.dll will call home
   (C&C functionality)

→ **Robot controller is now under attacker’s control**

Aside: Timeline

06/17/2016 > Reported vulnerabilities

06/21/2016 > Contacted by Cyber-Security Manager (Robotics)

10/14/2016 > Reported new vuln (Industrial Router)

08/29/2016 > Updates about fixed vulnerabilities

11/04/2016 > ABB Published Advisory

12/20/2016 > external vendor fixes Industrial Router vuln

We can fully compromise the robot's computers.

What now?

What does it mean to attack a robot?

Let's think about what a robot does

- I/O **Accuracy**
  - **Read** precise values
  - **Issue** correct/accurate commands
- **Safety**
  - Never harm humans
  - Correctly inform operator
- **Integrity**
  - No damage to the robot

Let's think about what a robot does

- I/O **Accuracy**
  - Read precise values
  - Issue correct/accurate commands
- **Safety**
  - Never **harm** humans
  - Correctly **inform** operator
- **Integrity**
  - No damage to the robot

Let's think about what a robot does

- I/O **Accuracy**
  - Read precise values
  - Issue correct/accurate commands
- **Safety**
  - Never harm humans
  - Correctly inform operator
- **Integrity**
  - No **damage** to the robot

Let's think about what a robot does

- I/O **Accuracy**
  - Read precise values
  - Issue correct/accurate commands
- **Safety**
  - Never harm humans
  - Correctly inform operator
- **Integrity**
  - No damage to the robot

Robot-specific Attack:
Digital-borne violation
of any of these requirements

Let's think about what a robot does

- I/O **Accuracy**
  - Read precise values
  - Issue correct/accurate commands
- **Safety**
  - Never harm humans
  - Correctly inform operator
- **Integrity**
  - No damage to the robot

Do these three thing remind you of something?!

Let's think about what a robot does

- I/O **Accuracy**
  - Read precise values
  - Issue correct/accurate commands
- **Safety**
  - Never harm humans
  - Correctly inform operator
- **Integrity**
  - No damage to the robot

A robot may not injure a human being or, through inaction, allow a human being to come to harm

A robot must obey the orders given it by human beings except where such orders would conflict with the First Law

A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws

Attack PoCs

1. **Accuracy Violation** PID parameters detuning
1. **Safety Violation** User-Perceived Robot State Alteration
1. **Integrity Violation** Control-loop alteration

PoC 1 Accuracy Violation

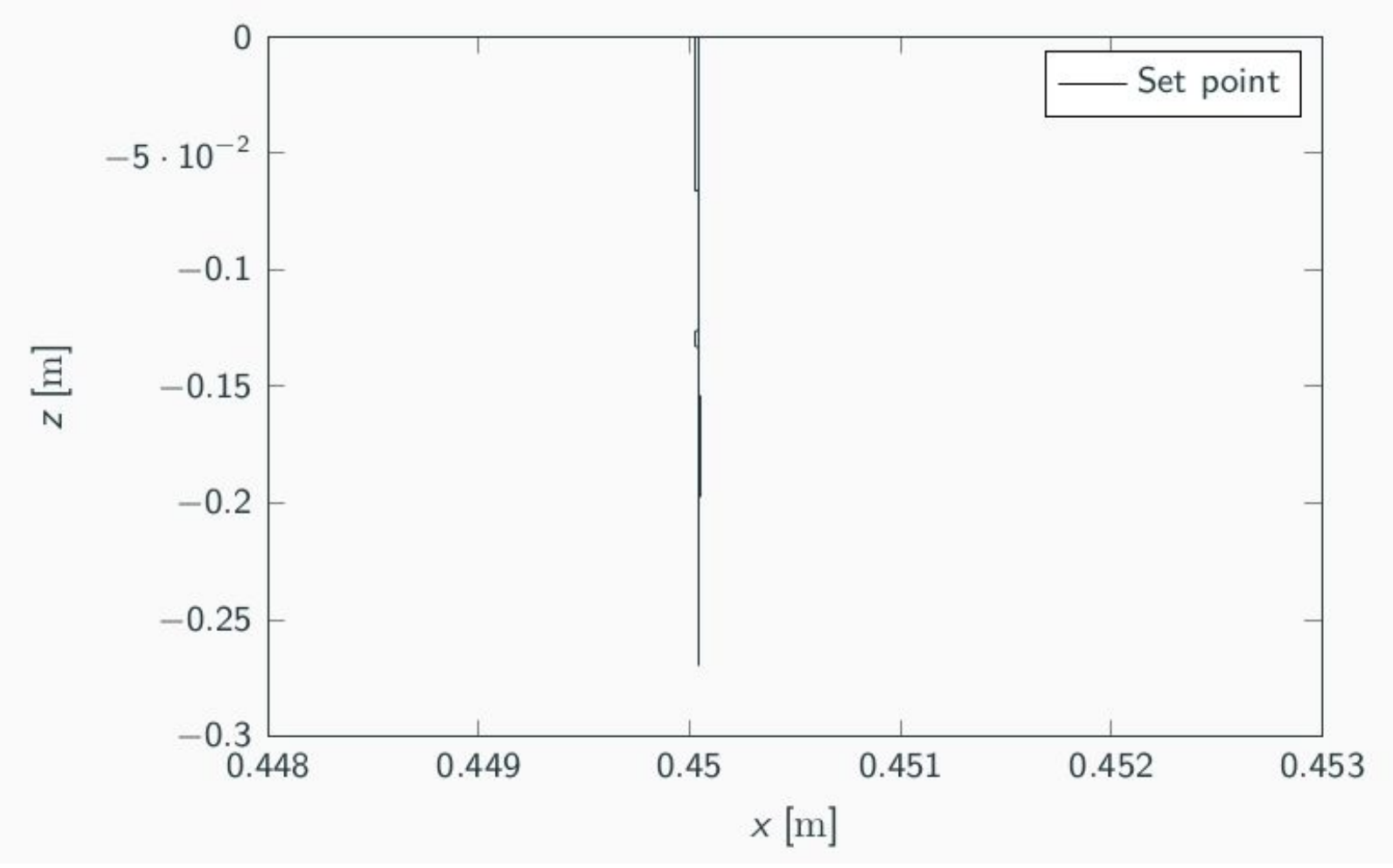
Attack Consequences

- defects in the end product
- structurally undermine the workpiece (e.g., car body)

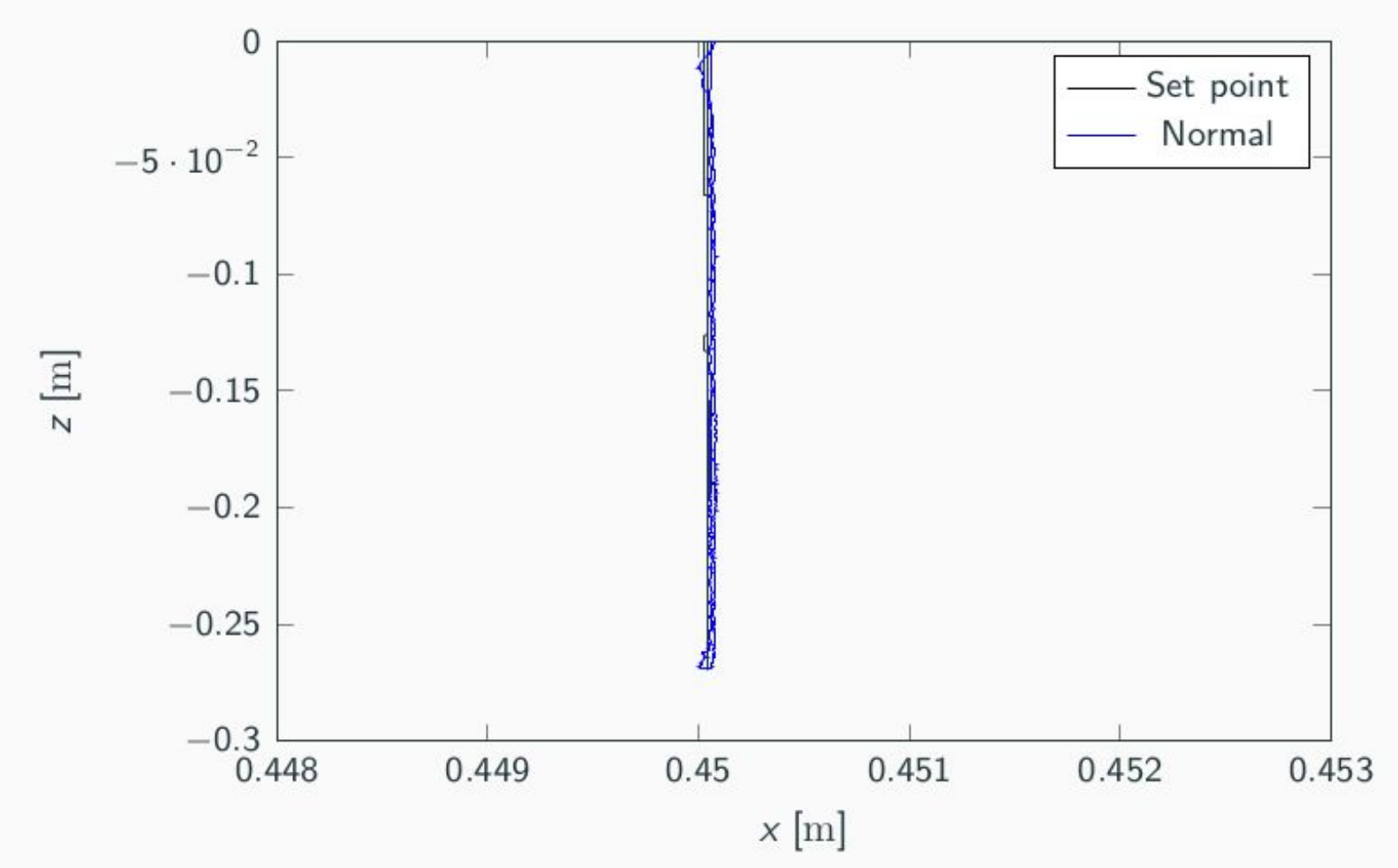
How:

- joints' angular position is PID controlled
- PID parameters are loaded at runtime from files
- obfuscation is broken
- plenty of remote code execution or file-write vulns

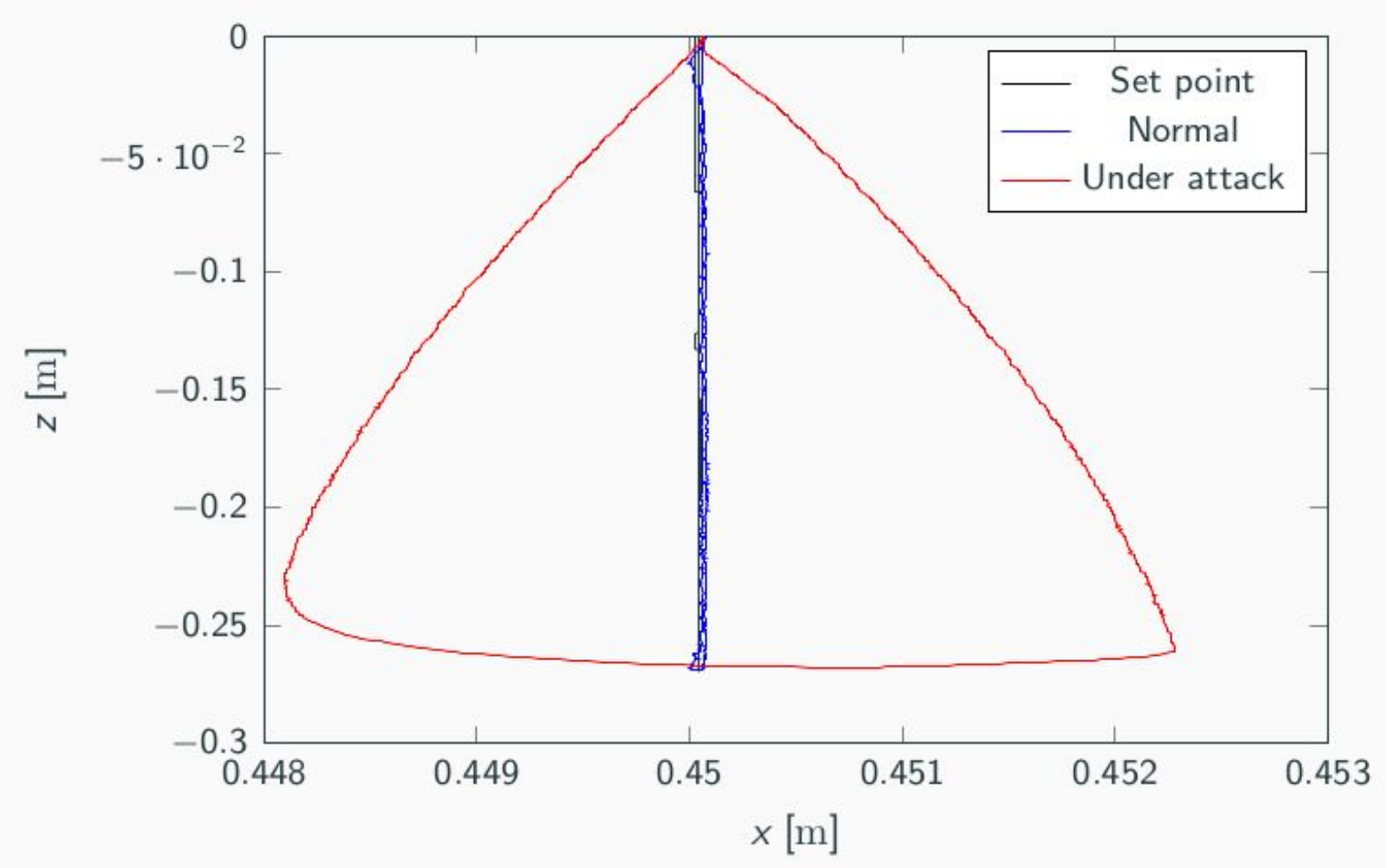
PoC 1 Accuracy Violation

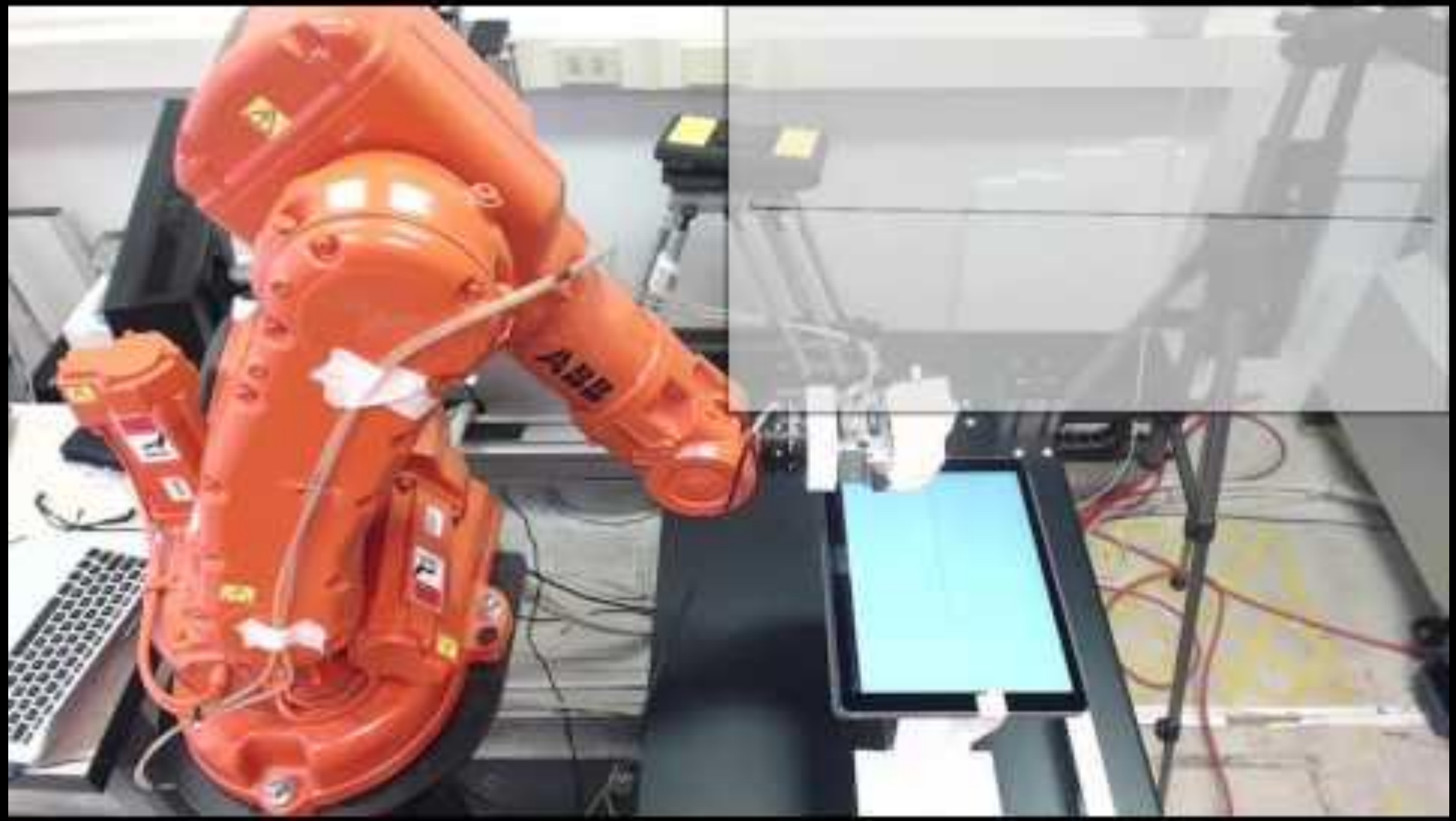


PoC 1 Accuracy Violation



PoC 1 Accuracy Violation





ABB

Attack PoCs

1. **Accuracy Violation** PID parameters detuning
1. **Safety Violation** User-Perceived Robot State Alteration
1. **Integrity Violation** Control-loop alteration

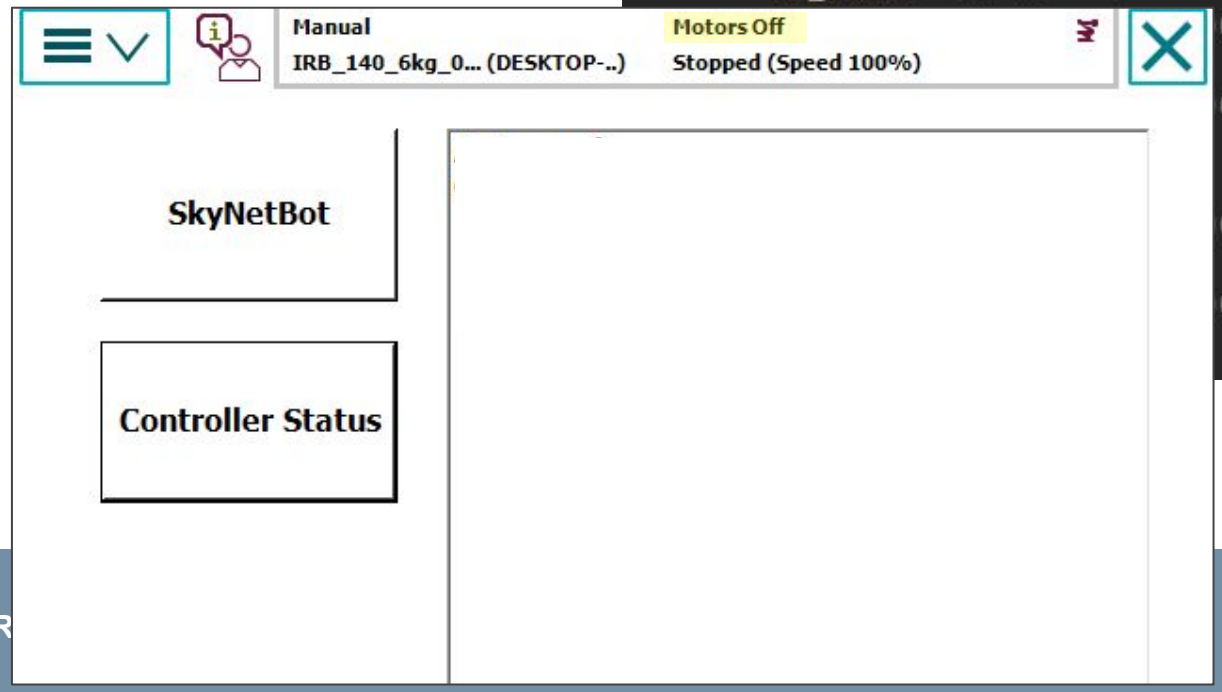

ABB
Auto
Production
APP42_P17_2 (192.168.8.217)
Motors On
Stopped (2 of 2) (Speed 100%)
Production Loop
Pre-menu
Menu
Post-menu
Waiting for order
Pre-part
Part
Post-part
Aborting
16:08
2013-06-17
Setup
Service
Info
Production Screen
E-STOP

PoC 2: Safety Violation

Malicious DLL

```
        IL_025c:  /* 03   |                    */ ldarg.1
        IL_025d:  /* 6F   | (0A)000028         */ callvirt   instance class [System.Drawing/*23000
() /* 0A000028 */
        //IL_0262:  /* 02   |                    */ ldarg.0
        //IL_0263:  /* 7B   | (04)0000A8         */ ldfld      string ABB.Robotics.Tps.Controls.St
        ldstr "Motors Off"
        IL_0268:  /* 02   |                    */ ldarg.0
        IL_0269:  /* 7B   | (04)0000B2         */ ldfld      class [System.Drawing/*23000007*/]Sys
        IL_026e:  /* 02   |                    */ ldarg.0
                                                0000B0      */ ldfld      class [System.Drawing/*23000007*/]Sys
                                                            */ ldloca.s   V_1
                                                000169      */ call       instance int32 [System.Drawing/*23000
                                                            */ conv.r4
                                                            */ ldloca.s   V_1
                                                0000DF      */ call       instance int32 [System.Drawing/*23000
                                                            */ conv.r4
                                                0000AD      */ callvirt   instance void [System.Drawing/*230000
```

Teach Pendant

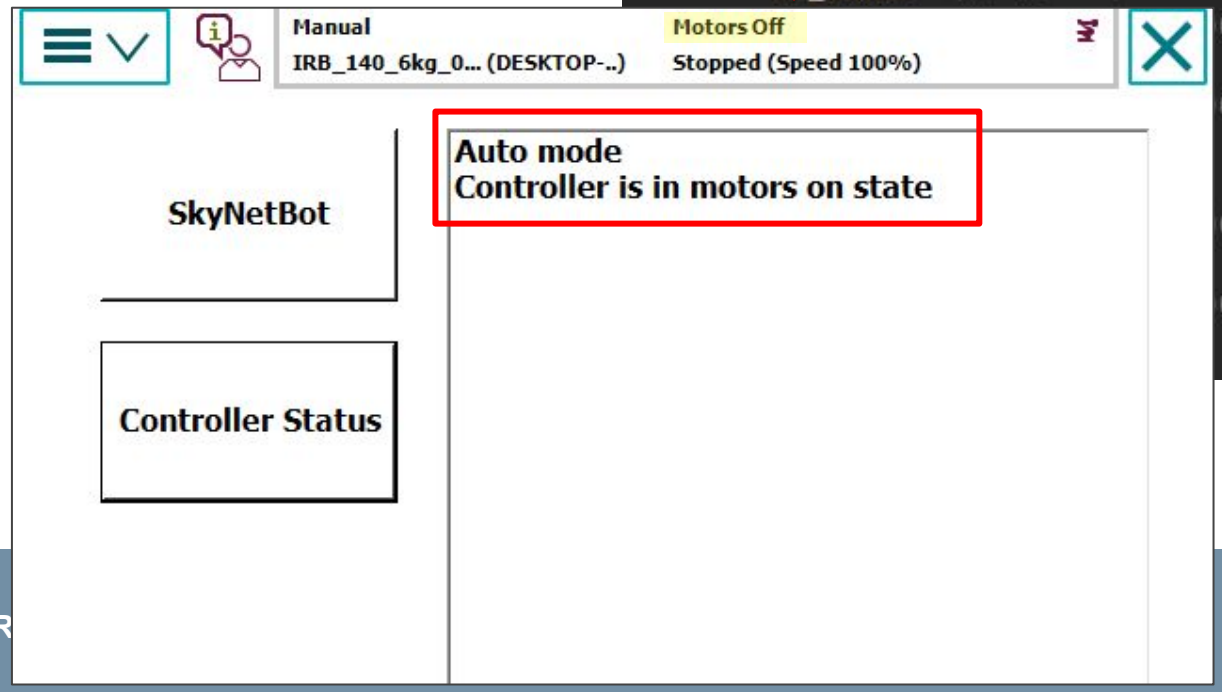


PoC 2: Safety Violation

Malicious DLL

```
        IL_025c:  /* 03   |                    */ ldarg.1
        IL_025d:  /* 6F   | (0A)000028         */ callvirt   instance class [System.Drawing/*23000
() /* 0A000028 */
        //IL_0262:  /* 02   |                    */ ldarg.0
        //IL_0263:  /* 7B   | (04)0000A8         */ ldfld      string ABB.Robotics.Tps.Controls.St
        ldstr "Motors Off"
        IL_0268:  /* 02   |                    */ ldarg.0
        IL_0269:  /* 7B   | (04)0000B2         */ ldfld      class [System.Drawing/*23000007*/]Sys
        IL_026e:  /* 02   |                    */ ldarg.0
                                                 0000B0      */ ldfld      class [System.Drawing/*23000007*/]Sys
                                                             */ ldloca.s   V_1
                                                 000169      */ call       instance int32 [System.Drawing/*23000
                                                             */ conv.r4
                                                             */ ldloca.s   V_1
                                                 0000DF      */ call       instance int32 [System.Drawing/*23000
                                                             */ conv.r4
                                                 0000AD      */ callvirt   instance void [System.Drawing/*230000
```

Teach Pendant



PoC 2: Safety Violation (?)

"The operational mode displayed at the teach pendant is for information only and is not part of the safety system. **Entering the safeguarded space in automatic mode will always lead to a protective stop regardless of the status information on the FlexPendant** since there are **mandatory regulations requiring that the safeguarded space shall be established by perimeter guarding**.

The safety system that implements safety functions is in accordance with EN ISO 10218-1:2011 and has been evaluated according to EN ISO 13849-1:2008 ‘Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design‘. As required by EN ISO 10218-1:2011, the safety functions achieve performance level d and category 3."

Custom Physical Protections, if any (despite regulations)

Fwd: [redacted] Researchers hijack a 220-pound industrial robotic arm

[redacted] to [redacted]

[redacted] has long had a robotics program and laboratories with larger robot arms than the one shown. These were the kind of robot arms where the lab floor had a red line to show the swing distance - inside that line and you could be struck by the arm, potentially fatally. Some of the early models were controlled by PCs connected to the corporate network. When powered down, the arms and their controllers were supposed to be safed. However, the COTS computers had a wake-on-LAN function. The internal security folks ran nmap with ping and happened to include the robotics labs' LAN. The PC woke up, automatically ran the robotics control program, and the arm extended to full length and swung around its full arc. This was witnessed by workers in the lab who, fortunately, were behind the red line.

Attack PoCs

1. **Accuracy Violation** PID parameters detuning
1. **Safety Violation** User-Perceived Robot State Alteration
1. **Integrity Violation** Control-loop alteration

PoC 3 Integrity Violation

Attack Consequences

- Robot’s arm collapse on itself
- Motors substantially damaged

Quite a risky PoC

- Same as PoC 1: Remote code execution to alter configuration files
- Showed configuration files to domain expert, who confirmed the effect

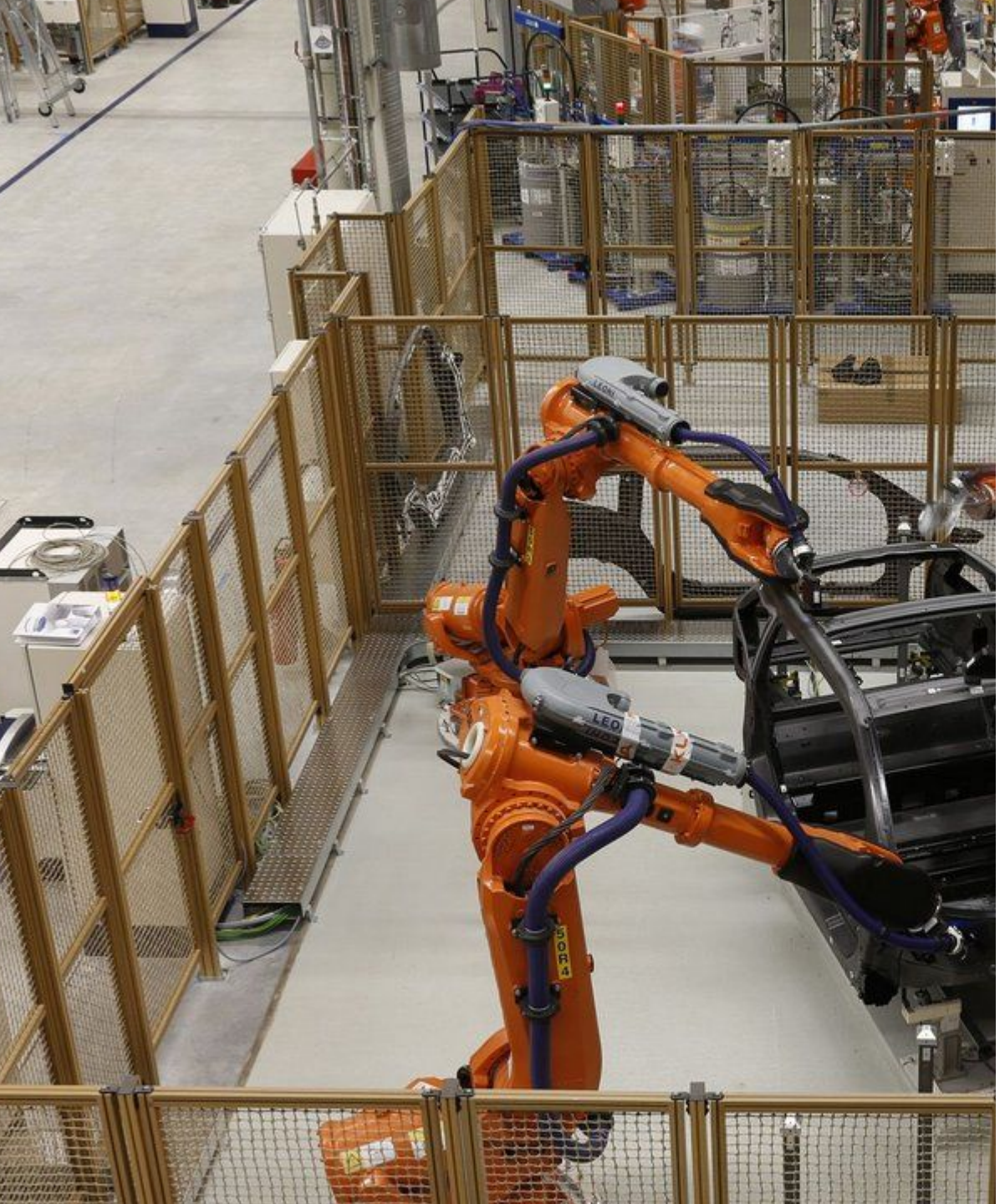
Let's generalize

- **Control-loop Alteration** (PoC 1)
  - alter movements at servo level
- **User-Perceived Robot State Alteration** (PoC 2)
  - fake state reported to user
- **Robot State Alteration** (similar to PoC 2)
  - switch manual -> auto
- **Production Logic Tampering** (PoC: easy)
  - upload custom programs
- **Calibration Parameters Tampering** (PoC 3)
  - fiddle with sensors or motors

From Attacks to Threat Scenarios

- Production Plant Halting
- Production Outcome Alteration
- Physical Damage
- Unauthorized Access
- Ransom requests to disclose micro defects (*oh, I could ransom that, too!*)

Looking Forward



Future Challenges

New standards, beyond safety issues

Attack detection and hardening

Secure collaborative robots & HRI

Thank you!

stefano.zanero@polimi.it · @raistolo

more info ~> http://robosec.org

Credits

Davide Quarta, Mario Polino, Andrea Zanchettin - Politecnico di Milano

Federico Maggi - Trend Micro's FTR

Marcello Pogliani - now @ Secure Network Srl

POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

TREND
MICRO™