SECURITY
LICENSING
PERFECTION IN PROTECTION

Security and Protection for Machine Learning

Axel Engelmann

Architect Protection Technologies – Wibu-Systems

Andreas Schaad

Professor of IT Security – University of Applied Sciences Offenburg

Where to find the accompanying audio

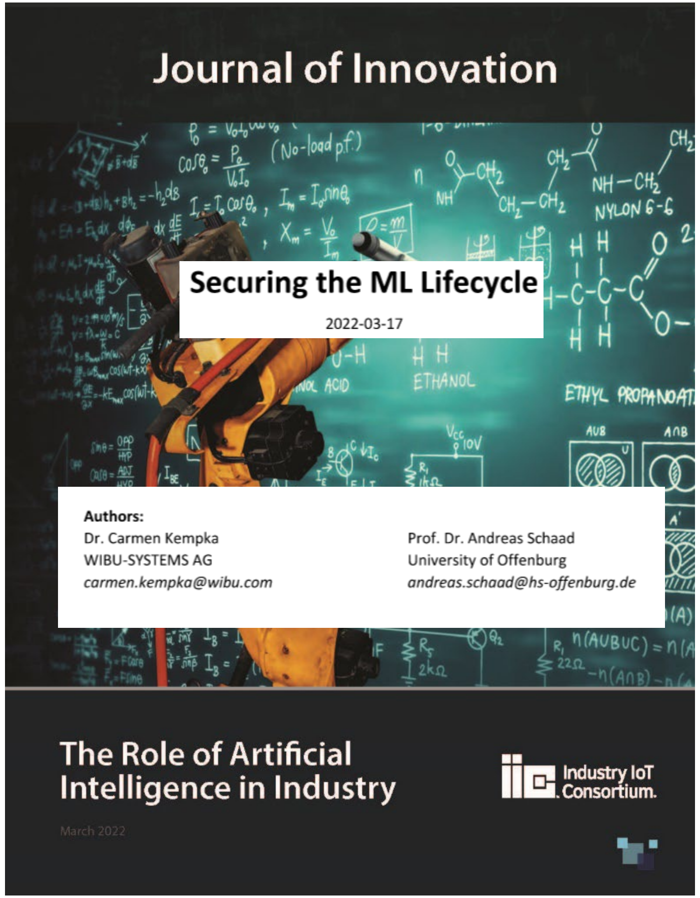
To access the on-demand replay of this masterclass, please visit

www.wibu.com/wibu-systems-webinars/security-and-protection-for-machine-learning/access.html

Introduction to Machine Learning

Machine Learning in a Nutshell

- In the widest sense, a specific field of Artificial Intelligence.
- Machine Learning comprises a set of techniques / tools that now complement our software development lifecycle.
- Why?
  - Can replace hard to maintain rulesets / imperative programming
  - Widely available computational frameworks
  - CPU power / Cloud platforms / Data
  - Available skillset increasing / part of Comp. Science curriculum
- But: Securing the ML Lifecycle is important!



Journal of Innovation

Securing the ML Lifecycle

2022-03-17

Authors:

Dr. Carmen Kempka
WIBU-SYSTEMS AG
carmen.kempka@wibu.com

Prof. Dr. Andreas Schaad
University of Offenburg
andreas.schaad@hs-offenburg.de

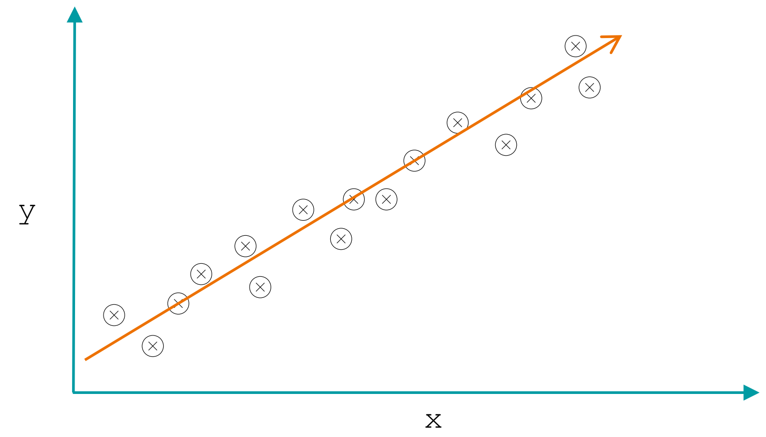
The Role of Artificial Intelligence in Industry

March 2022

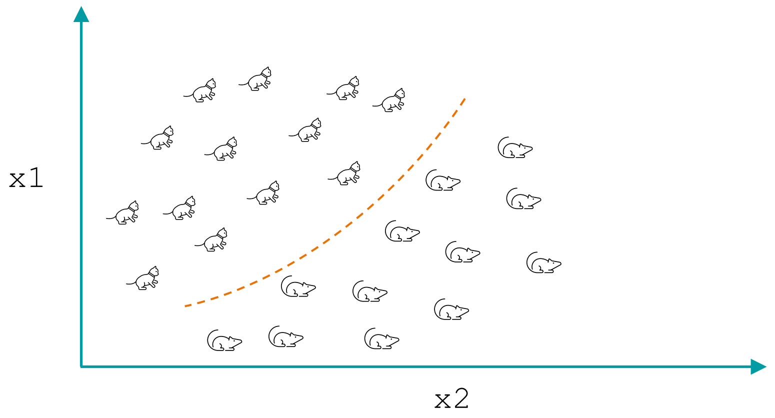
Industry IoT Consortium.

Machine Learning in a Nutshell

- Making predictions based on already known data

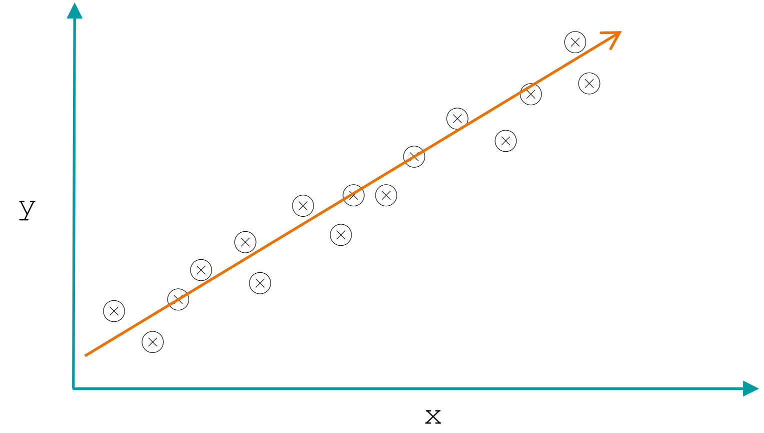


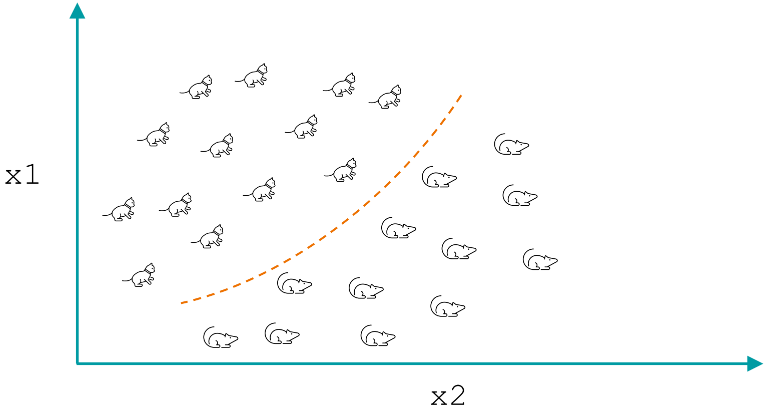
- Classifying new data based on known data



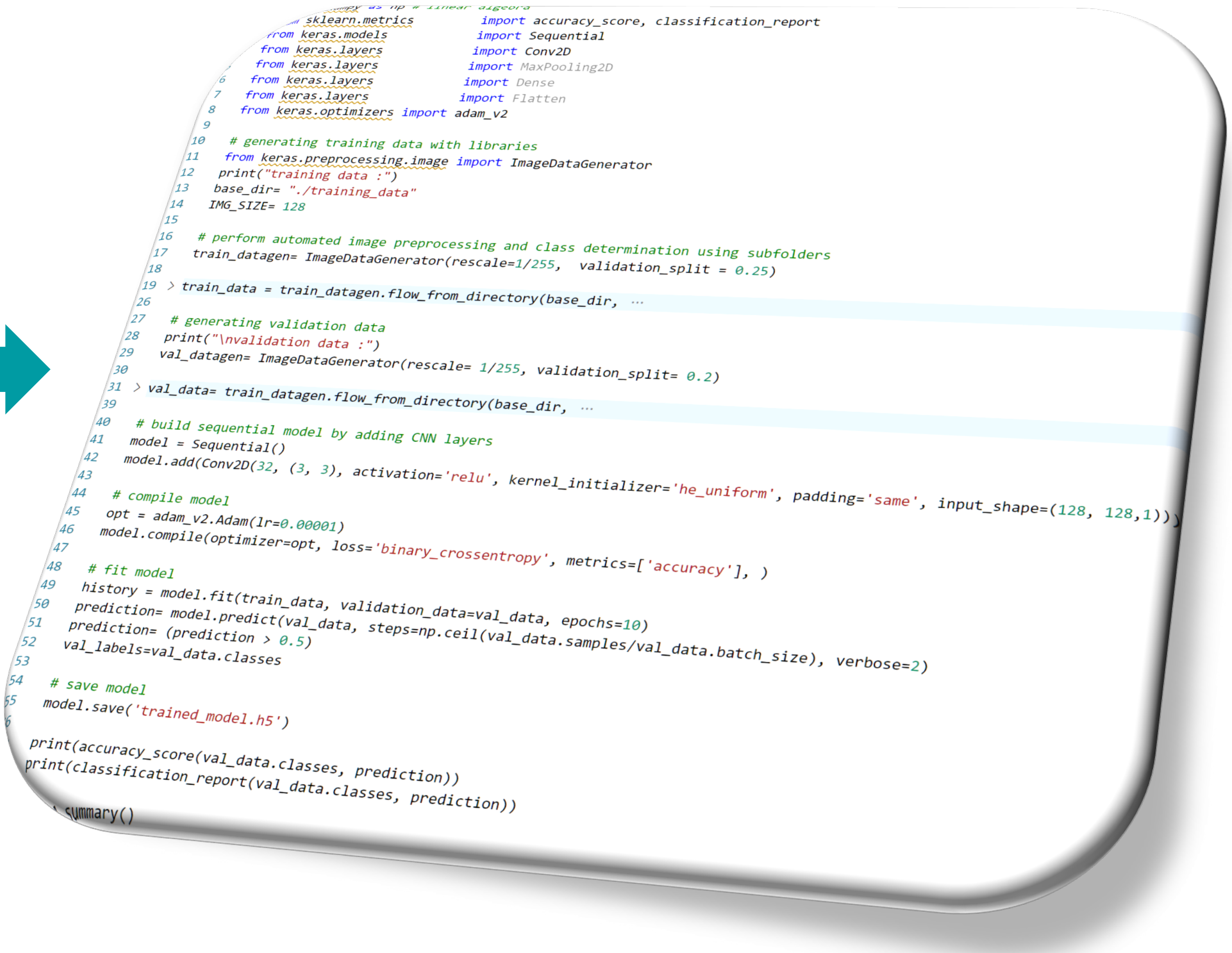
Examples

- Making predictions based on already known data
  - Financial Forecasting
  - Maintenance Prediction
  - Network Analysis
  - ….
- Classifying new data based on known data
  - Spam Filtering
  - Image Recognition
  - Intrusion Detection
  - …





Example



```
umpy as np # linear algebra
m sklearn.metrics              import accuracy_score, classification_report
from keras.models              import Sequential
from keras.layers              import Conv2D
from keras.layers              import MaxPooling2D
from keras.layers              import Dense
from keras.layers              import Flatten
from keras.optimizers import adam_v2

# generating training data with libraries
from keras.preprocessing.image import ImageDataGenerator
print("training data :")
base_dir= "./training_data"
IMG_SIZE= 128

# perform automated image preprocessing and class determination using subfolders
train_datagen= ImageDataGenerator(rescale=1/255,  validation_split = 0.25)

train_data = train_datagen.flow_from_directory(base_dir, ...

# generating validation data
print("\nvalidation data :")
val_datagen= ImageDataGenerator(rescale= 1/255, validation_split= 0.2)

val_data= train_datagen.flow_from_directory(base_dir, ...

# build sequential model by adding CNN layers
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(128, 128,1)))

# compile model
opt = adam_v2.Adam(lr=0.00001)
model.compile(optimizer=opt, loss='binary_crossentropy', metrics=['accuracy'], )

# fit model
history = model.fit(train_data, validation_data=val_data, epochs=10)
prediction= model.predict(val_data, steps=np.ceil(val_data.samples/val_data.batch_size), verbose=2)
prediction= (prediction > 0.5)
val_labels=val_data.classes

# save model
model.save('trained_model.h5')

print(accuracy_score(val_data.classes, prediction))
print(classification_report(val_data.classes, prediction))
summary()
```

WIBU SYSTEMS

The Machine Learning Lifecycle

1. Training Phase

Trained model is created.

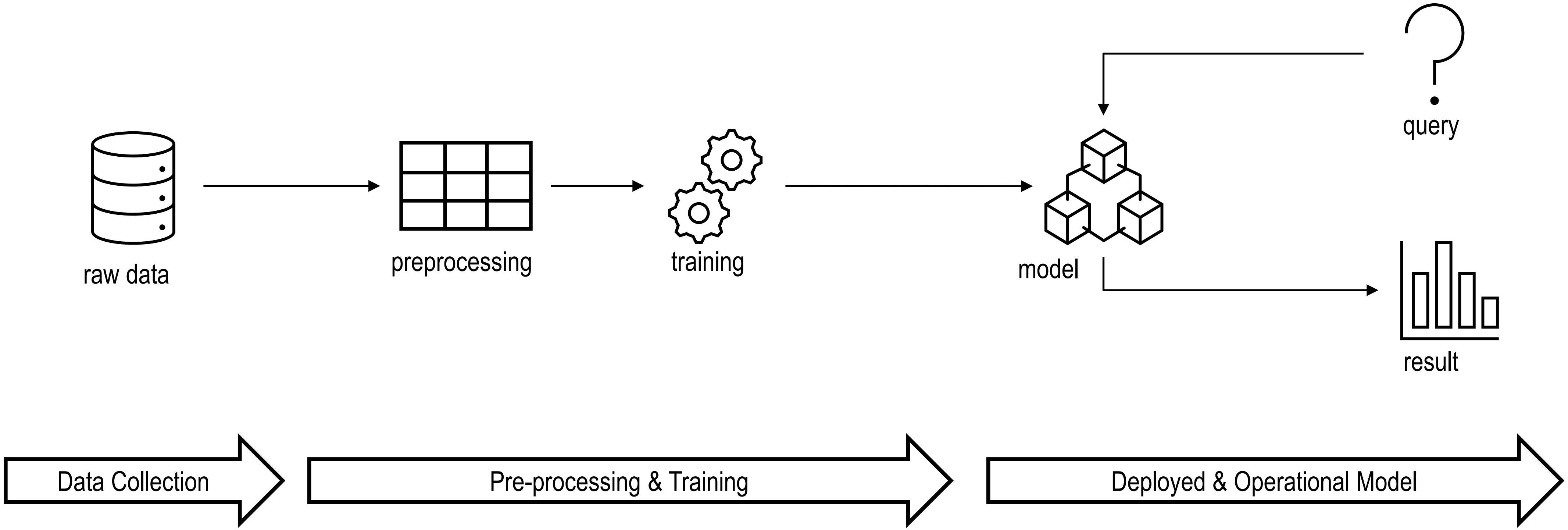
- Data collection
- Pre-processing & Feature engineering
- Training process using a framework code
- Outcome: Trained model

2. Inference Phase

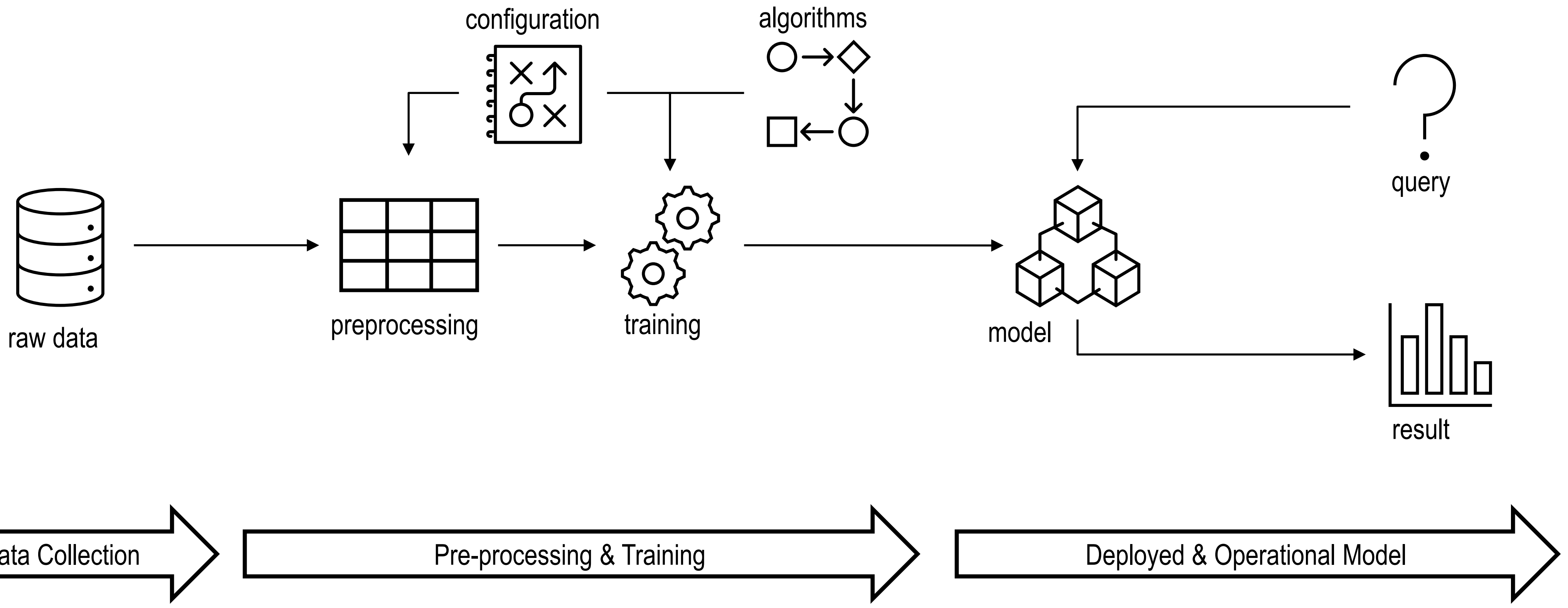
Trained model is used to predict results from new data. Cloud or offline usage.

- Input
- Pre-processing
- Prediction using trained model
- Outcome: Output (Prediction)

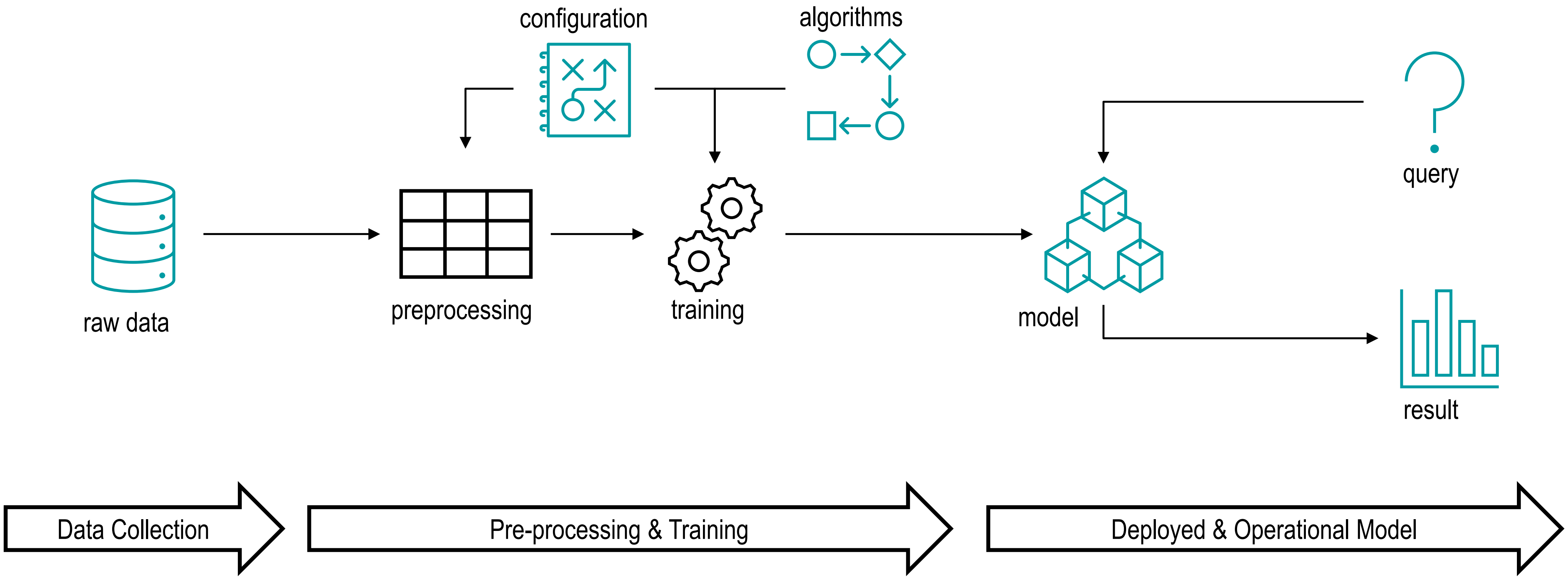
The Machine Learning Lifecyle



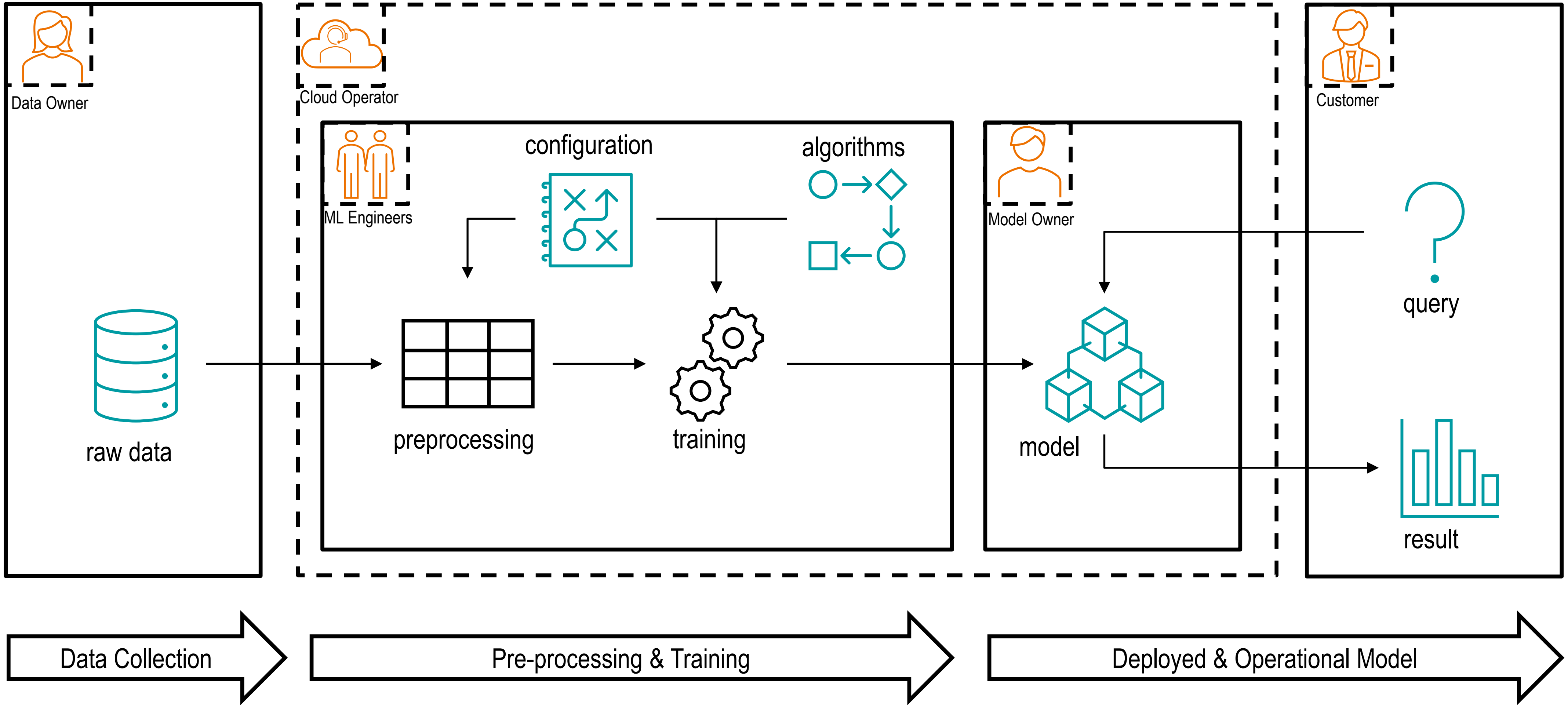
The Machine Learning Lifecycle



The Machine Learning Lifecycle

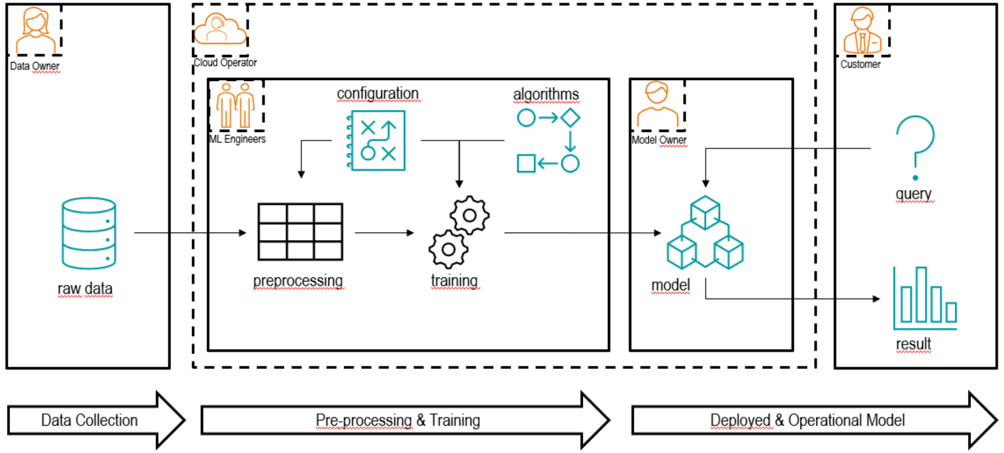


The Machine Learning Lifecycle

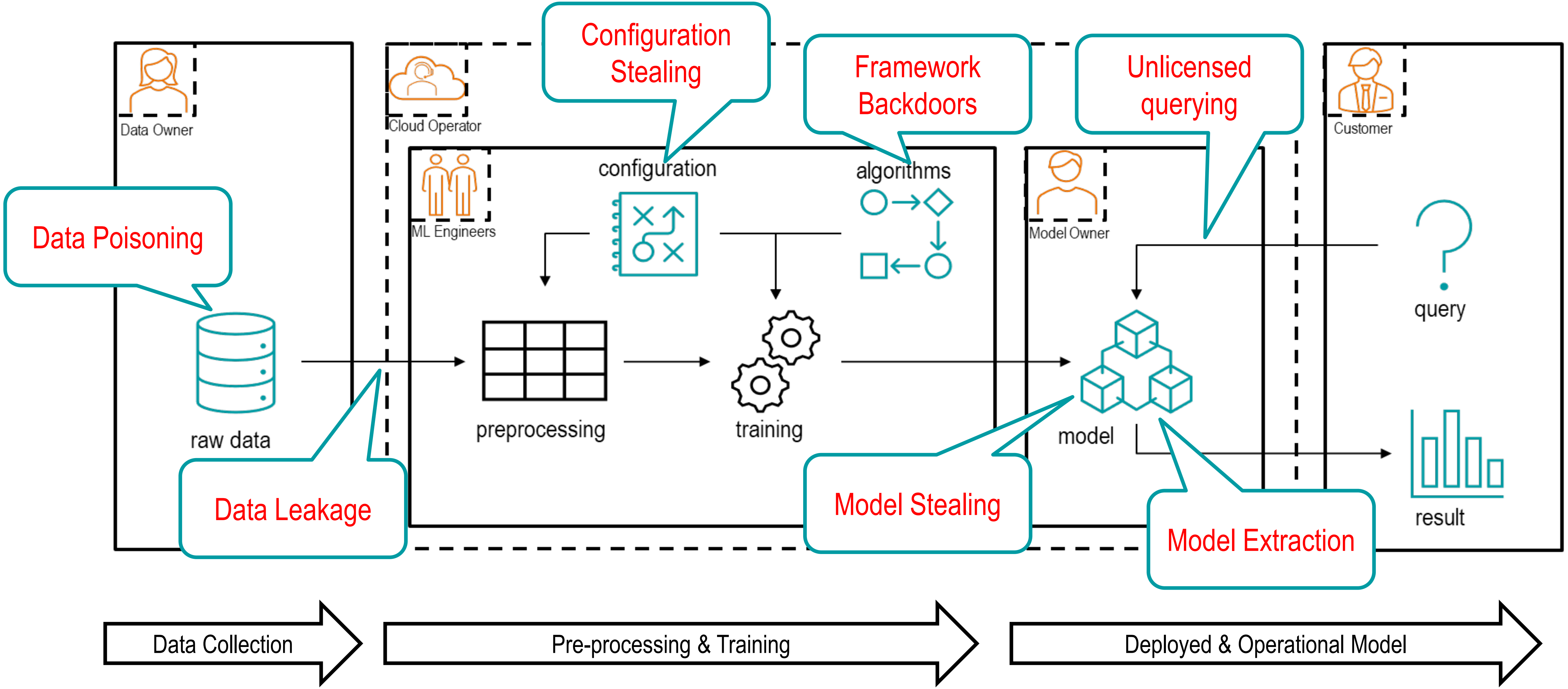


Summary

- What are the assets we need to protect?
  - Source / Training data
  - Training configuration
  - Licensed access to our trained model
  - Secure delivery of results
- …and many stakeholders with different access or licensing requirements



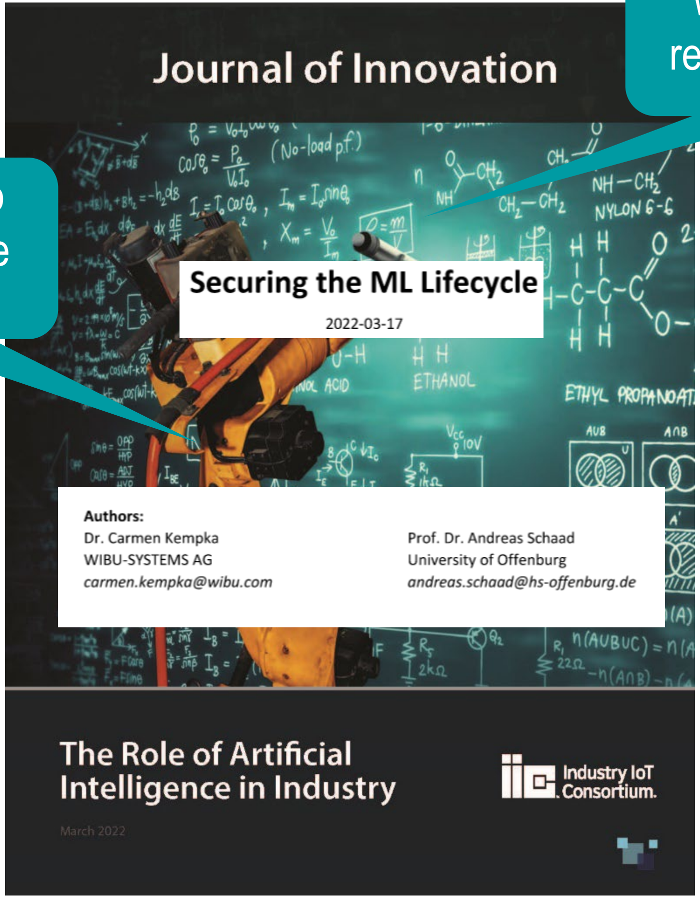
Attacking the ML Lifecycle



More Details

What do I need to consider to secure my ML pipeline?

What are known real-world attacks?



Journal of Innovation

Securing the ML Lifecycle

2022-03-17

Authors:
Dr. Carmen Kempka
WIBU-SYSTEMS AG
carmen.kempka@wibu.com

Prof. Dr. Andreas Schaad
University of Offenburg
andreas.schaad@hs-offenburg.de

The Role of Artificial Intelligence in Industry

March 2022

Industry IoT Consortium.

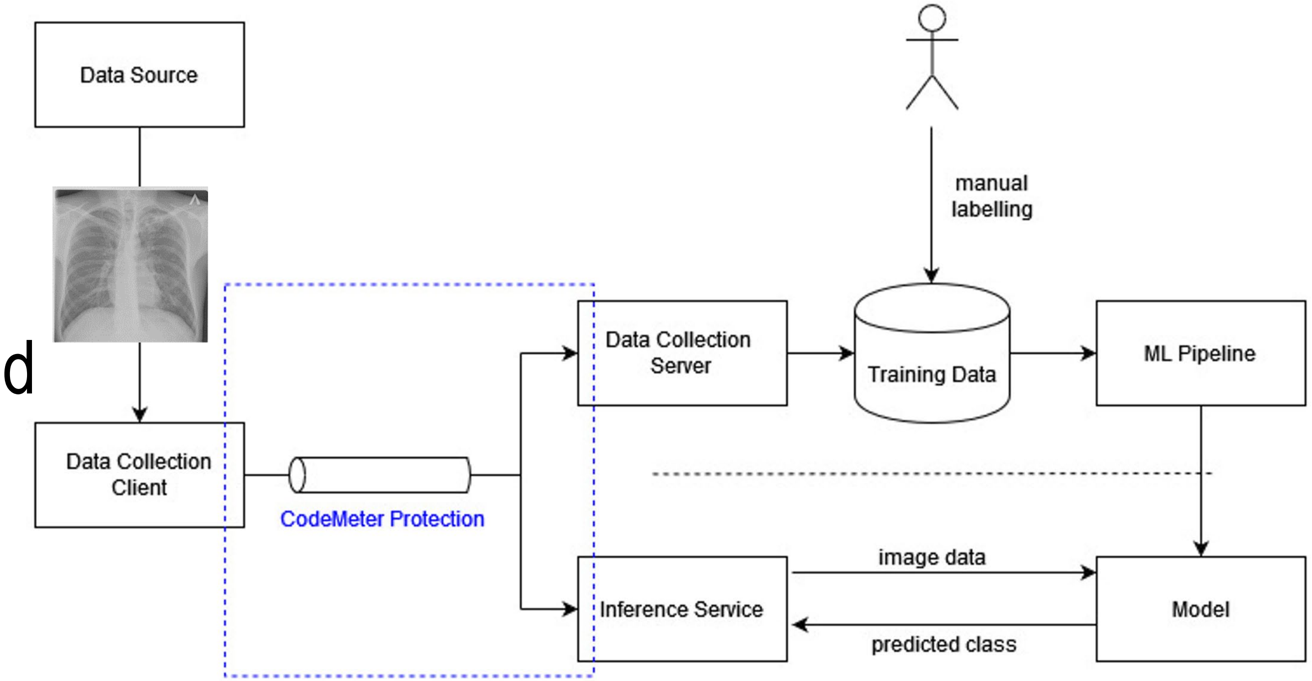
https://joom.ag/gdpd/p40

Threats

Phase	Description	Category (CIA)	Access needed
Training	Data Poisoning	Integrity	no
Training	Model Poisoning	Integrity	yes
Inference	Model Stealing	Confidentiality	yes
Inference	Model Replacement	Integrity, Availability	yes
Inference	Model Extraction	Confidentiality	no
Inference	Inference/Exfiltration Attacks	Confidentiality	no
Inference	Perturbation Attacks	Integrity	no
Both	Software Dependencies of ML System Exploit	Confidentiality, Integrity, Availiability	no

Proof of Concept

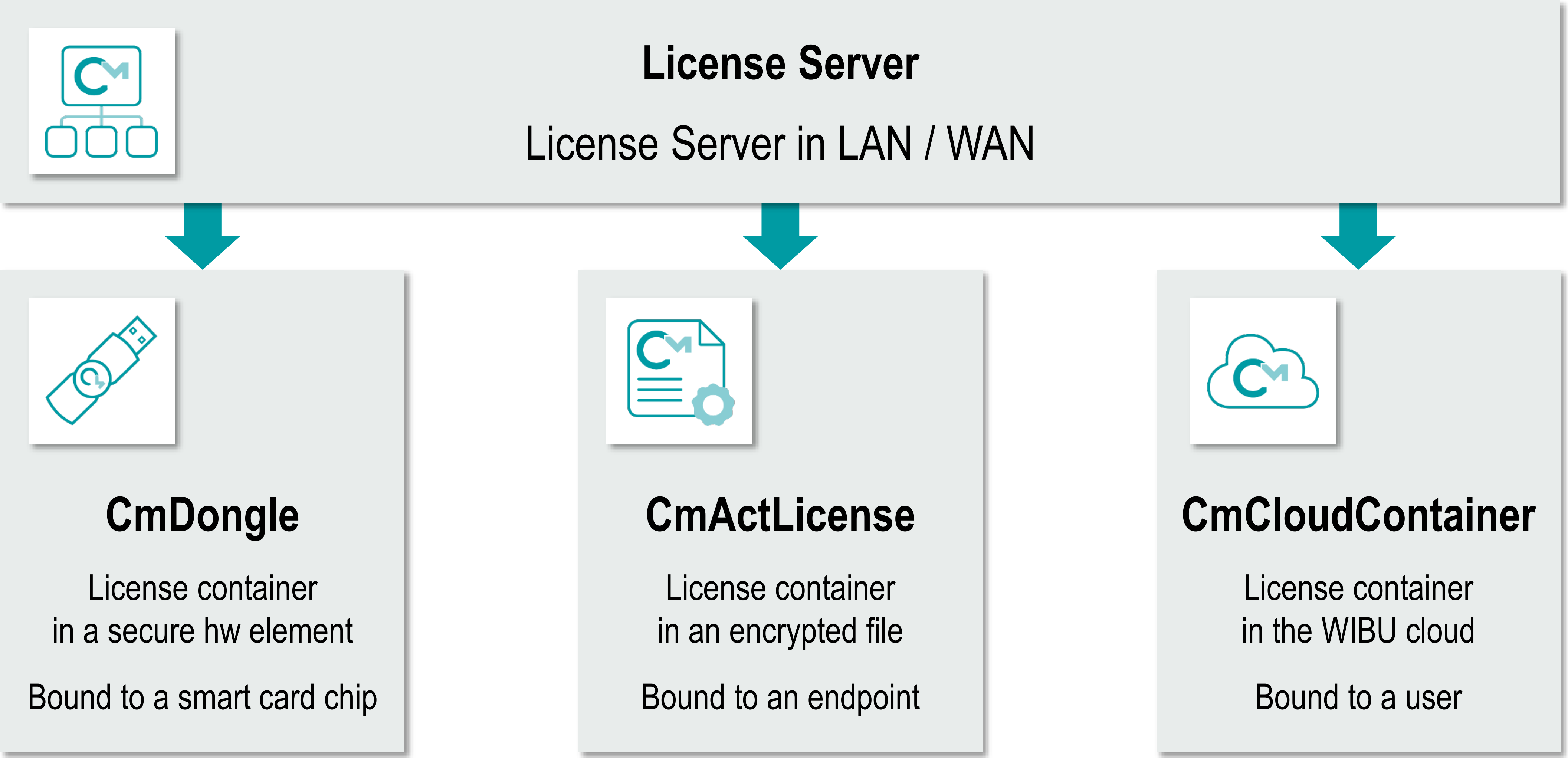
- Medical ML project
  - 3500 x-ray pictures
- Data transfer from source to ML environment already secured using CodeMeter
- Protection against data poisoning



- Today’s Demo: Protecting the training model against stealing

CodeMeter at a Glance

CodeMeter Licensing Systems



License Server

License Server in LAN / WAN

CmDongle

License container
in a secure hw element

Bound to a smart card chip

CmActLicense

License container
in an encrypted file

Bound to an endpoint

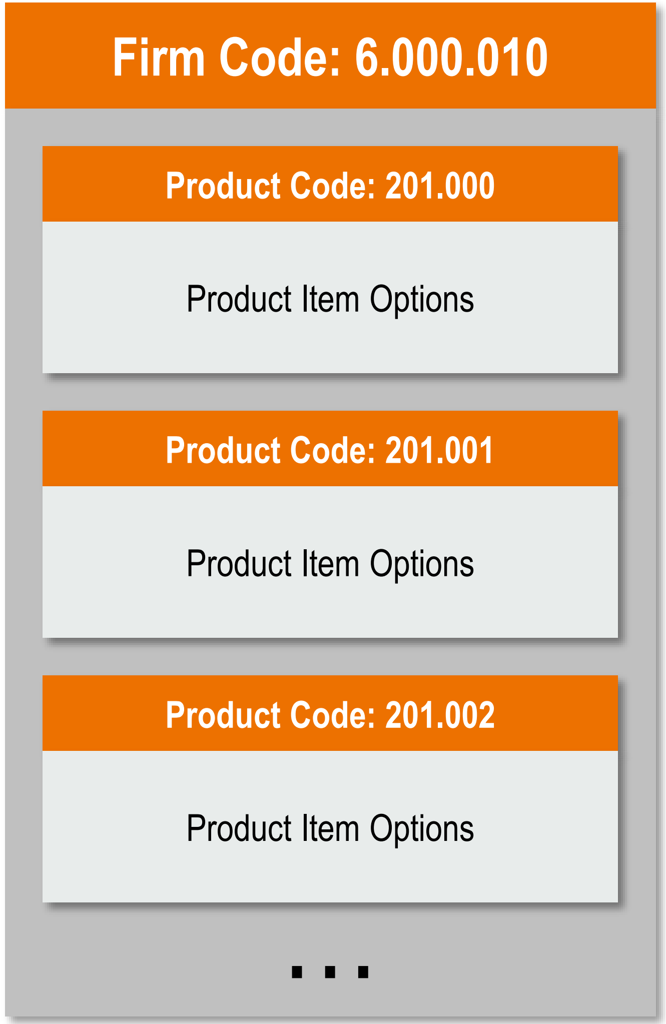
CmCloudContainer

License container
in the WIBU cloud

Bound to a user

License Entries

- License entry = Firm Code | Product Code
- Firm Code: issued by Wibu-Systems
- Product Code:
  - Defined by the software vendor
  - Per Option / Module / Feature
  - 4 bn. Product Codes (UInt32)
- Up to 2,000 Product Items per CmContainer
- Product Item Options: Each license can include combinable options



Firm Code: 6.000.010

Product Code: 201.000

Product Item Options

Product Code: 201.001

Product Item Options

Product Code: 201.002

Product Item Options

. . .

CodeMeter Protection Suite

Overview CodeMeter Protection Suite

	Windows	macOS	Linux	.NET	Python	JavaScript	Java	Android
Automatic Protection	1336-1000	1336-1200	1336-1300	1336-2000	1336-1700	1336-1800	1336-3000	1336-1500
Modular Licensing	1336-1001	1336-1201	1336-1301	1336-2001	1336-1701	1336-1801	1336-3001	1336-1501
IP Protection	1336-1002	1336-1202	1336-1302	1336-2002	1336-1702	1336-1802	1336-3002	1336-1502
CodeMoving	1336-1003	1336-1203	1336-1303	planned	1336-1703	1336-1803	1336-3003	1336-1503
File Encryption	planned	planned	planned	planned	1336-1704	1336-1804	planned	planned
Additional Targets	-	-	1336-135x	1336-205x	1336-175x	136-185x	planned	-

Modules of AxProtector

- Basic
  - Protection with one license list (0)
  - Encryption on method level
- Modular Licensing
  - Use of more than 1 license list out of license lists other than (0)
- IP Protection
  - Encryption without using CodeMeter licensing capabilities (fixed key)
- CodeMoving
  - Use of CodeMoving (CmDongle and CmCloudContainer)
- File Encryption
  - File encryption modus (AI models)

AxProtector Technologies – Overview

Binary-Mode AxProtector

- AxProtector Windows
- AxProtector macOS
- AxProtector Linux
- AxProtector Android

IL-Mode AxProtector

- AxProtector .NET
- AxProtector Python
- AxProtector JavaScript
- AxProtector Java

AxProtector Technologies – Functionality

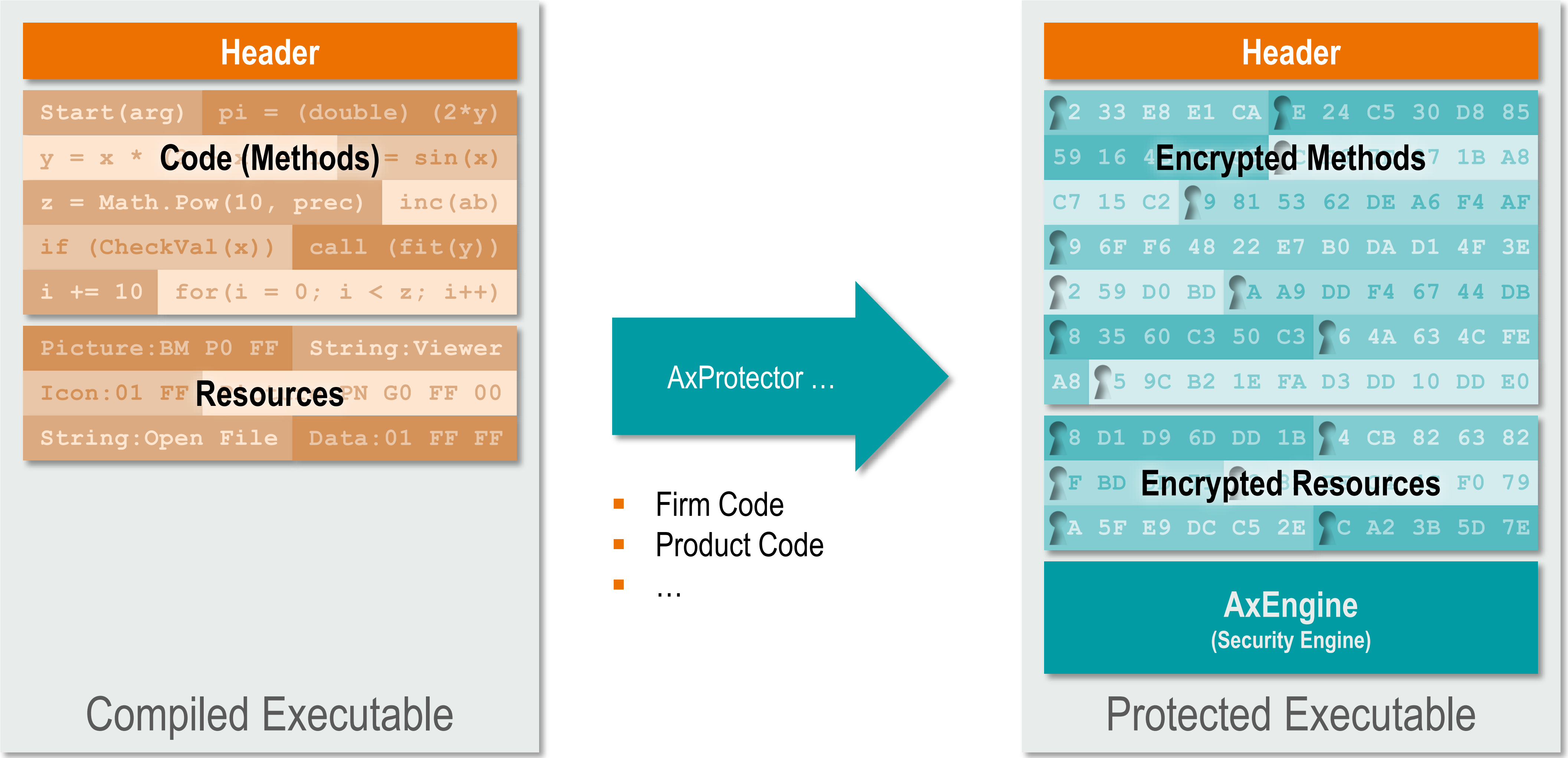
Binary-Mode AxProtector

- Encryption of the entire application as one blob
- Encryption on method level requires manual integration
- Complete decryption during startup (except for individual defined methods)
- No unpredictable performance impact during runtime

IL-Mode AxProtector

- Encryption of individual methods / classes as individual blobs
- Automatic encryption on method level
- Highest security thanks to on-demand decryption of every method
- Very small performance impact during runtime thanks to intelligent caching

Operating Principle of IL-based AxProtector



- Firm Code
- Product Code
- …

Demo

Demo - Recap

- Client / Server application
  - Image classification using a trained model
  - Prediction of a tuberculosis desease
- AxProtector Python
  - YAML files for encryption specification
  - Protection of Python server scripts
  - Protection of a trained h5 model
- Application only works with valid licenses

Many thanks for your kind attention

Europe:	+49-721-931720
USA:	+1-425-7756900
China:	+86-21-55661790
Japan:	+81-45-5659710

https://www.wibu.com

info@wibu.com