

# WAHTARI

End-to-End  
Deep Learning  
Ecosystem

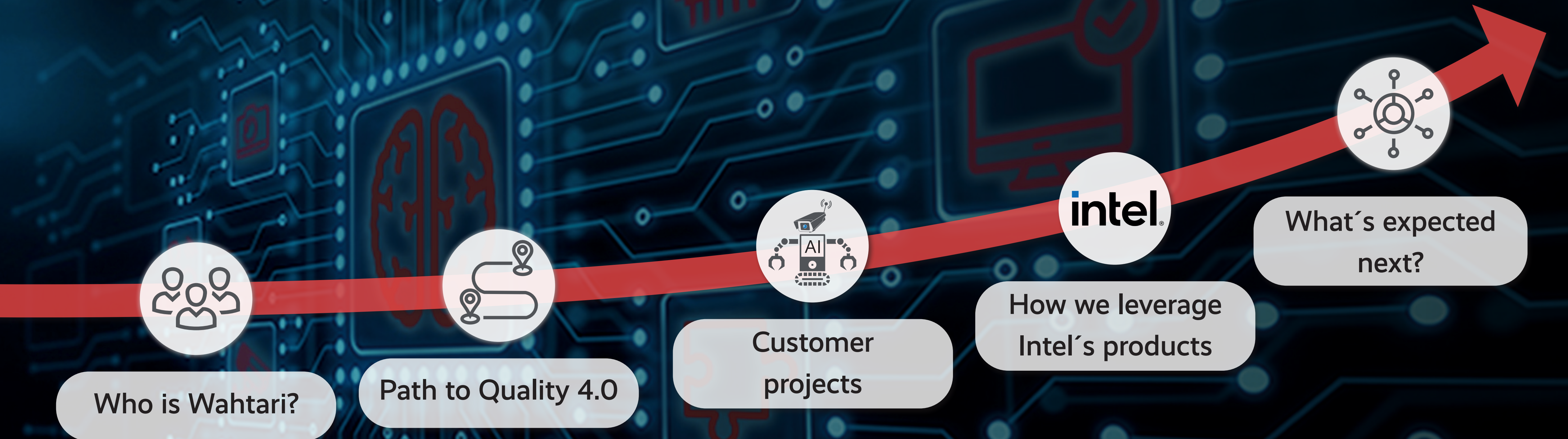
[wahtari.io](https://wahtari.io)



Today's speaker:  
Sebastian Borchers



# Agenda





Who is Wahtari?

**WAHTARI**

# Founder

## Roland Singer



- Founder of Manjaro Linux OS
- IT-Security, Linux & Go Enthusiast
- PCB Engineering & Electronics
- Deep Learning Researcher

## Marwin Gambel



- Founded Startup for Synthetic Fuel Production
- Hardware-Design & CNC Engineering
- Business Development & Marketing
- CEO at Wahtari

## Sebastian Borchers



- Co-Founder of BavarIT
- Microservices, Linux & Go Enthusiast
- Software Engineering
- Deep Learning Researcher



Founded in mid 2019

### Some of our partners



### Some of our customers

LEONI HITACHI

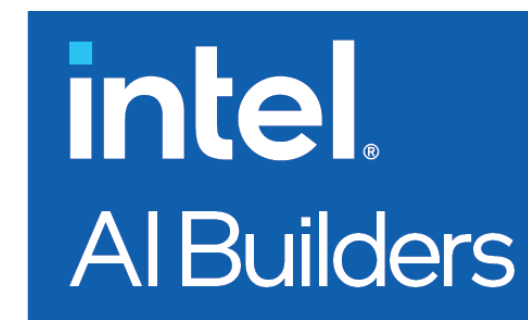
NELSKAMP



### Wahtari is sponsor of



### Wahtari is member of



INCEPTION  
PROGRAM

WAHTARI



# Path to Quality 4.0



# Quality 4.0

With further adoption of Industry 4.0, quality control becomes a key role for a true smart and fully automated industry. Our experience within the German industry has outlined the lack of quality control and quality automation in various crucial fields.



**The outlier in  
the Industry 4.0  
movement**

## **Our experience has outlined:**

- Customers have little to no quality guidelines & specifications
- Mostly human based quality inspection
- No fixed routines and missing smart procedures for instant reaction
- High barriers to entry
- Technology is still adapting to these new requirements



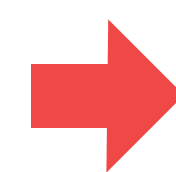
Industry 4.0 demands  
Plug & Play  
Deep Learning based  
Inspection Solutions

Deep Learning

Industrial Sector

Plug & Play

Machine Vision



With the rise of Industry 4.0 machine vision and smart inspection solutions must further evolve to keep pace with the rapid automation. Cutting edge research technology within the AI and deep learning field must meet the robustness and stability requirements of the industrial sector. Plug and play solutions with a straightforward integration process are key for a scalable transition to the next generation of quality inspection.



# How we solved it: All in one computer vision AI ecosystem

## Hardware

**Wahtari nCam**  
neural camera

**nLine**  
360 inline inspection

**nBox**  
High performance AI  
acceleration



## Software

**nGin**  
Smart AI engine for  
computer vision

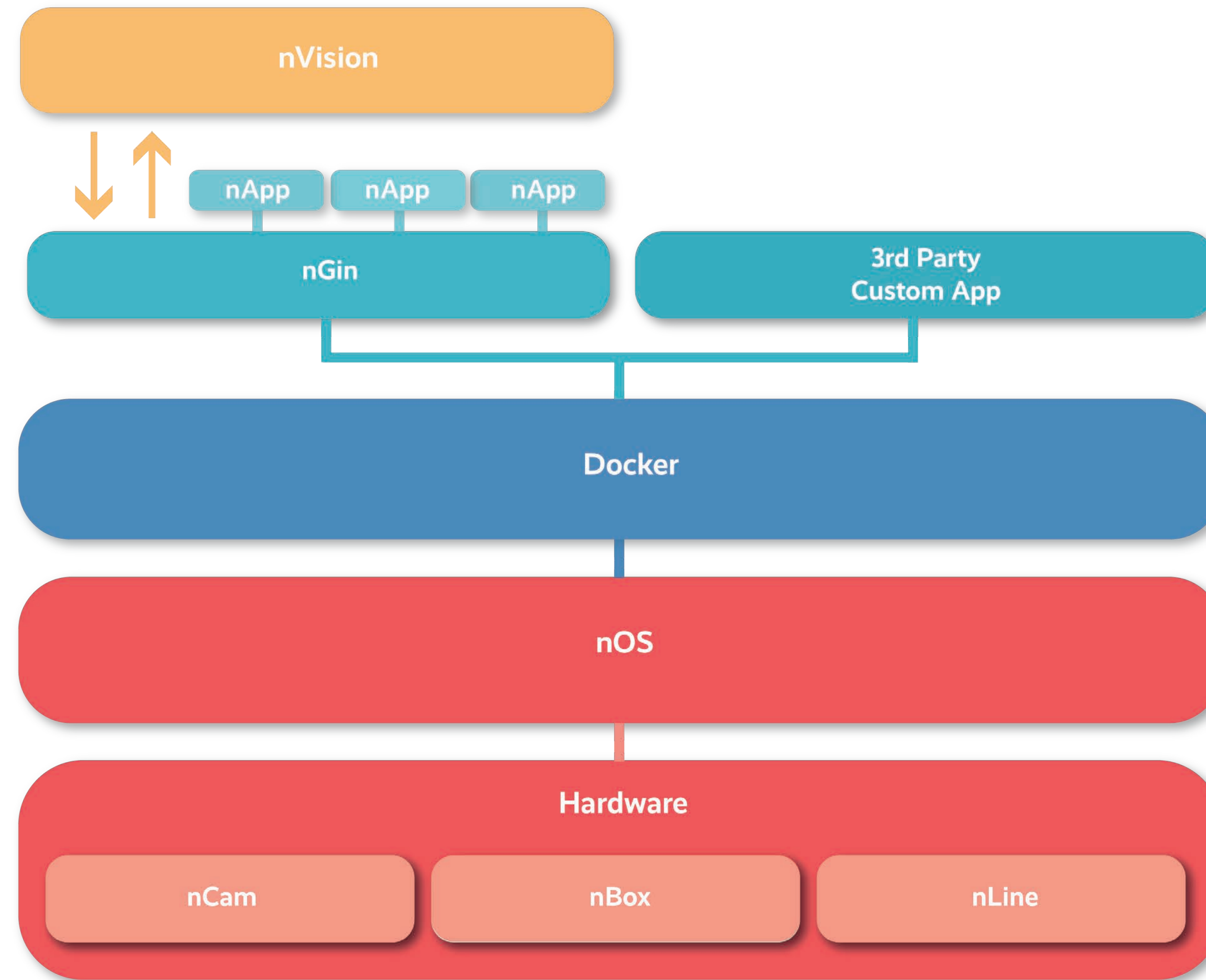
**nOS**  
secure and lightweight  
edge operating system

**nVision**  
Modular UI system

**WAHTARI**



# Ecosystem Overview





# Wahtari nCam Hardware



## Heater

- Smart automatic dual heater (-40°)
- Smart window heating
- Over-heating and low temperature protection

## Lens

- Motorized zoom & focus
- Motorized P-iris
- Switchable IR filter cut
- Various options: From near, mid and far range

## LED / IR

- High-power white LED (95 CRI)
- High-power IR (850nm)

## 4G

- Optional integrated 4G modem
- IP67 protected 4G antennas
- Edge ready with minimized installation overhead

## WIFI

- Optional integrated WIFI support
- IP67 protected WIFI antennas

## Hardened Aluminum Housing

- Up to IP67 protection
- Optional sun shield
- Multiple mounting options (ITS mounting & swivellink)
- IP67 M12 industrial IO connectors

## AI / VPU

- 3 Myriad X AI accelerators
- Up to 120 frames per second AI inference
- Efficient passive cooling
- Broad AI model support for various custom models
- Straightforward deployment with Wahtari nGin SDK

## Alvium Allied Vision

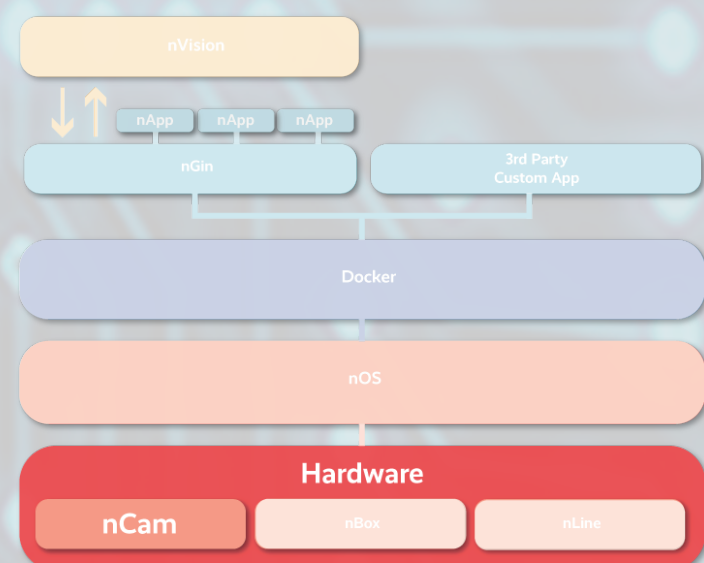
- Compatible with any Alvium camera modul
- UP to 150 frames per second
- Up to 20.4 MegaPixel
- Native support

## OS

- Small and secure operating system included
- Deploy custom Docker Apps
- Continuous security updates
- Simple deployment process
- Over-the-Air-Updates to any remote location
- End-to-End encrypted communication

## Computing

- Powerful x86\_64 Intel CPU
- Quad-Core 64bit 2 GHz
- Up to 128GB eMMC
- Up to 8GB RAM
- H.264 & VP8 hardware acceleration encoding

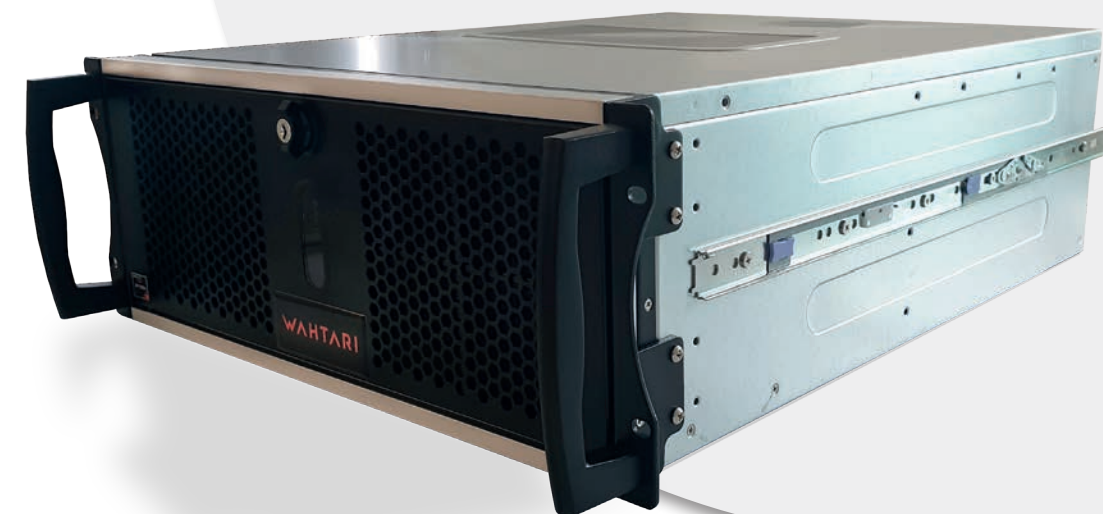


# WAHTARI



## nBox

High performance AI  
acceleration

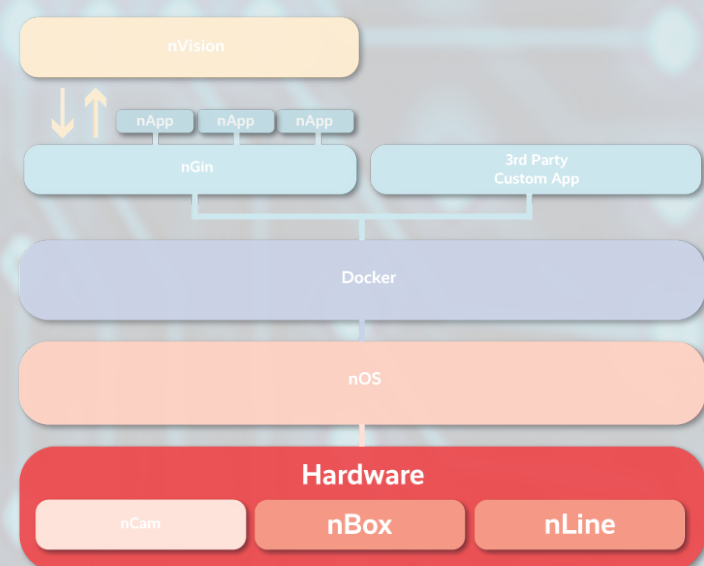


## nLine

360 inline inspection



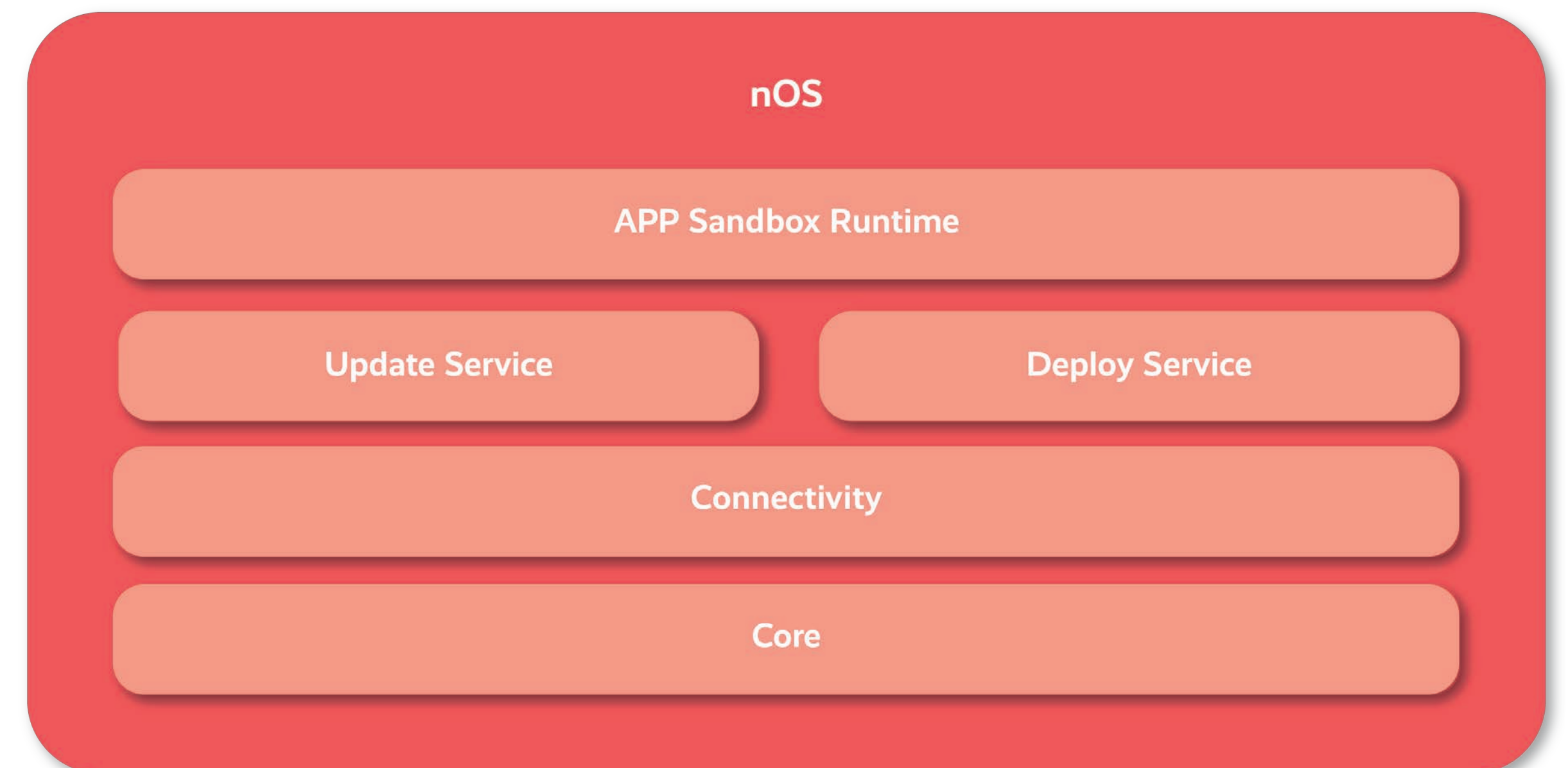
➔ Modular components allow easy creation of  
custom solutions



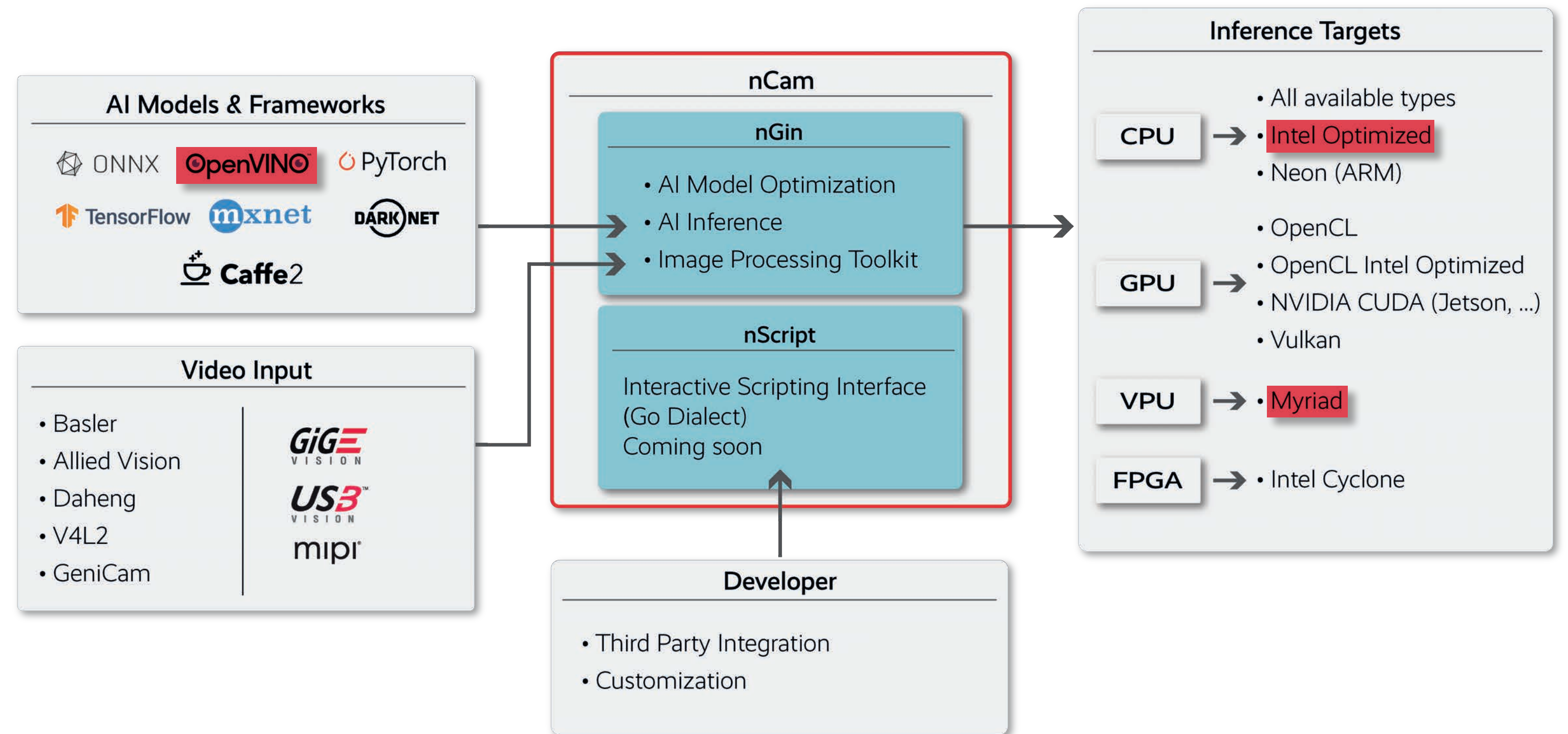


# nOS - IoT operating system

- Linux based
- Lightweight
- Core layers are read only
- Security first (signed updates)
- Bare minimum of packages







## AI model integration

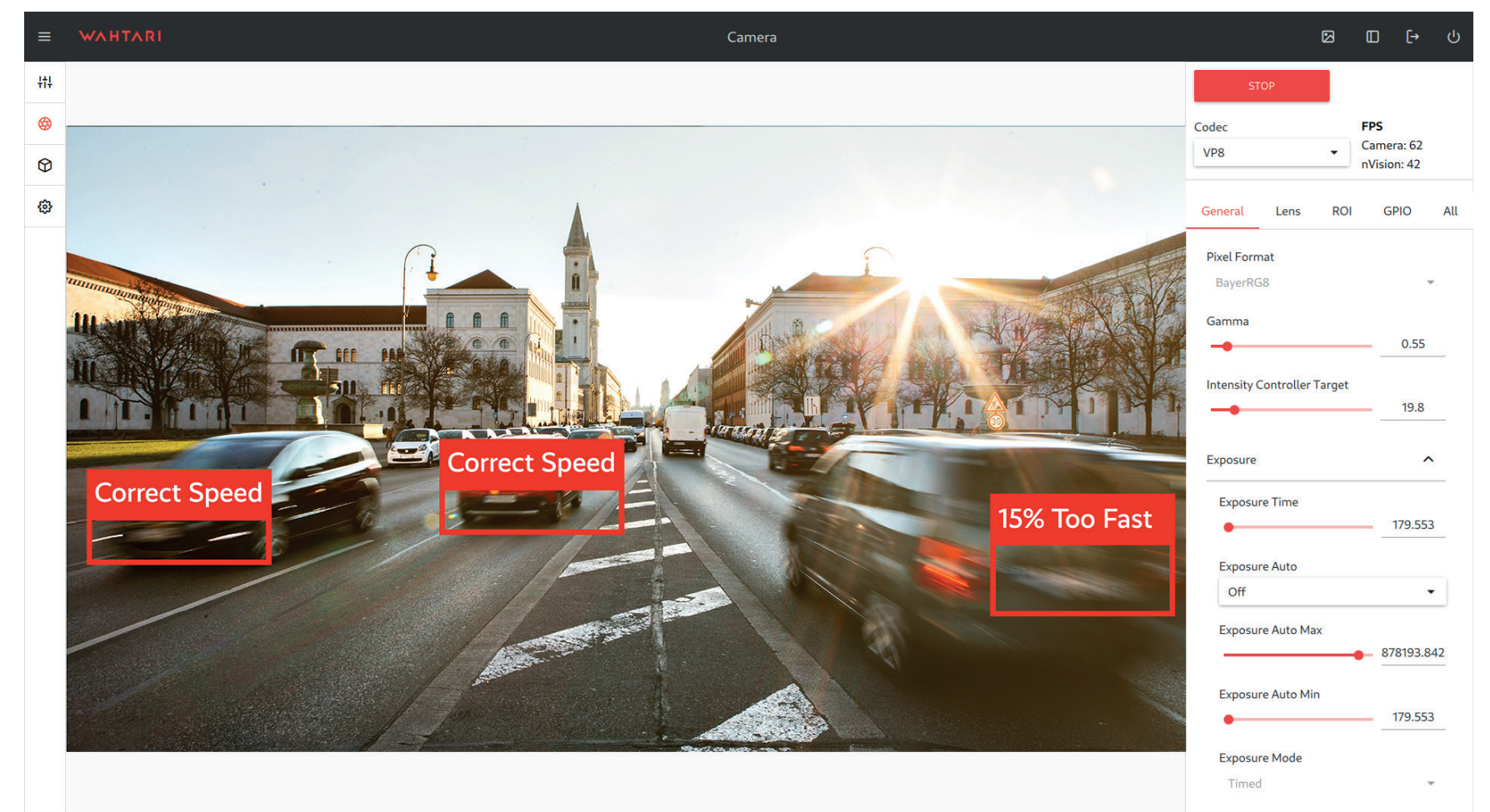
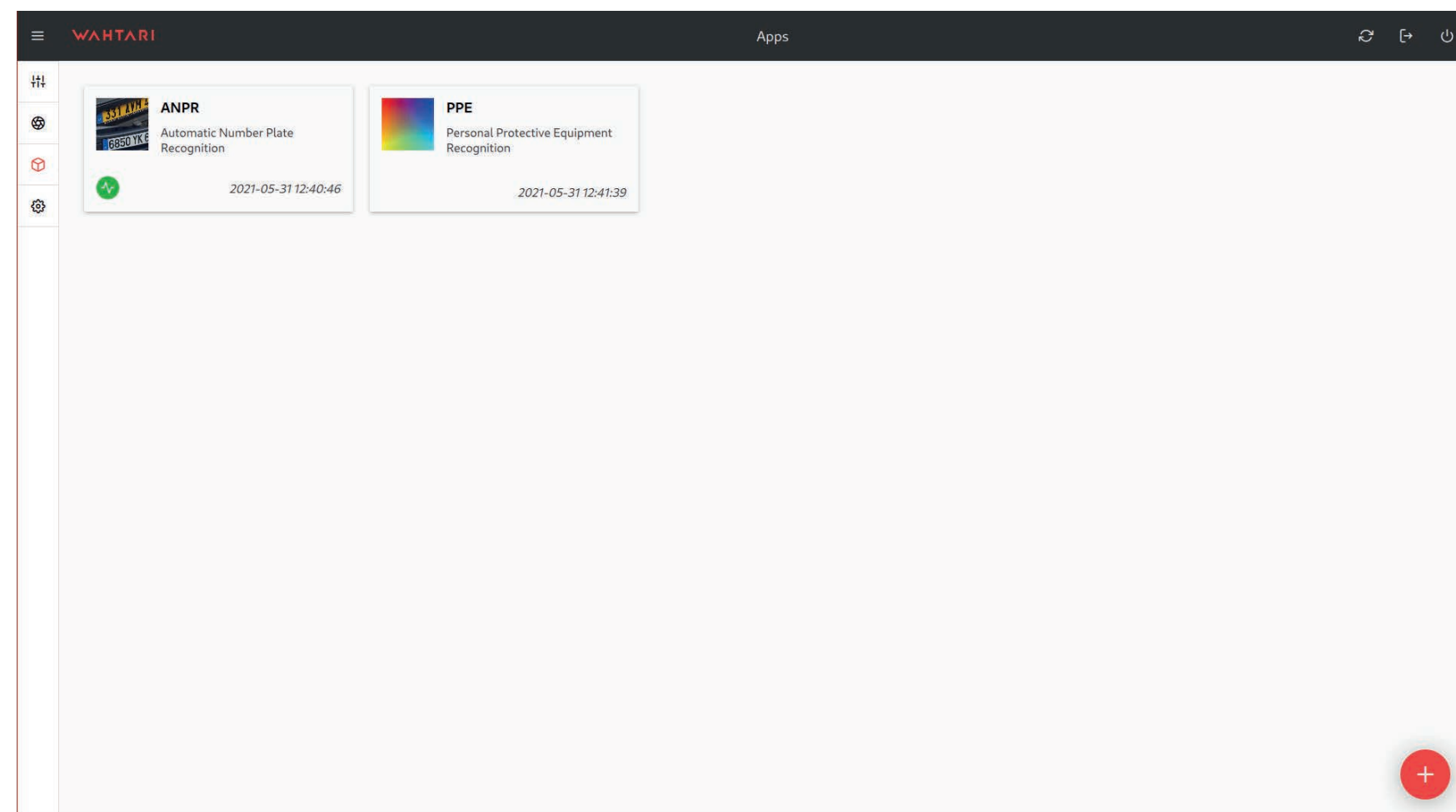
The smart neural engine for edge & cloud devices

- Powers nCam, nBox & nLine
- Machine vision SDK
- Traditional image algorithms
- AI inference engine
- Manages camera sensor, controller & VPU's





# nVision

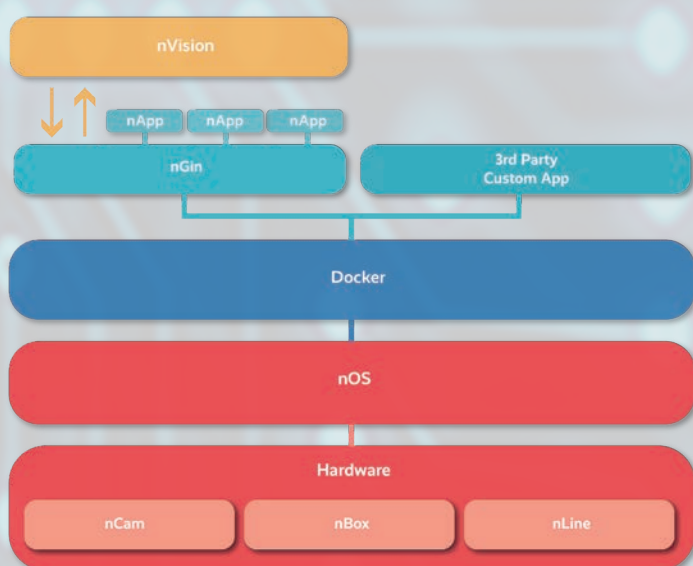
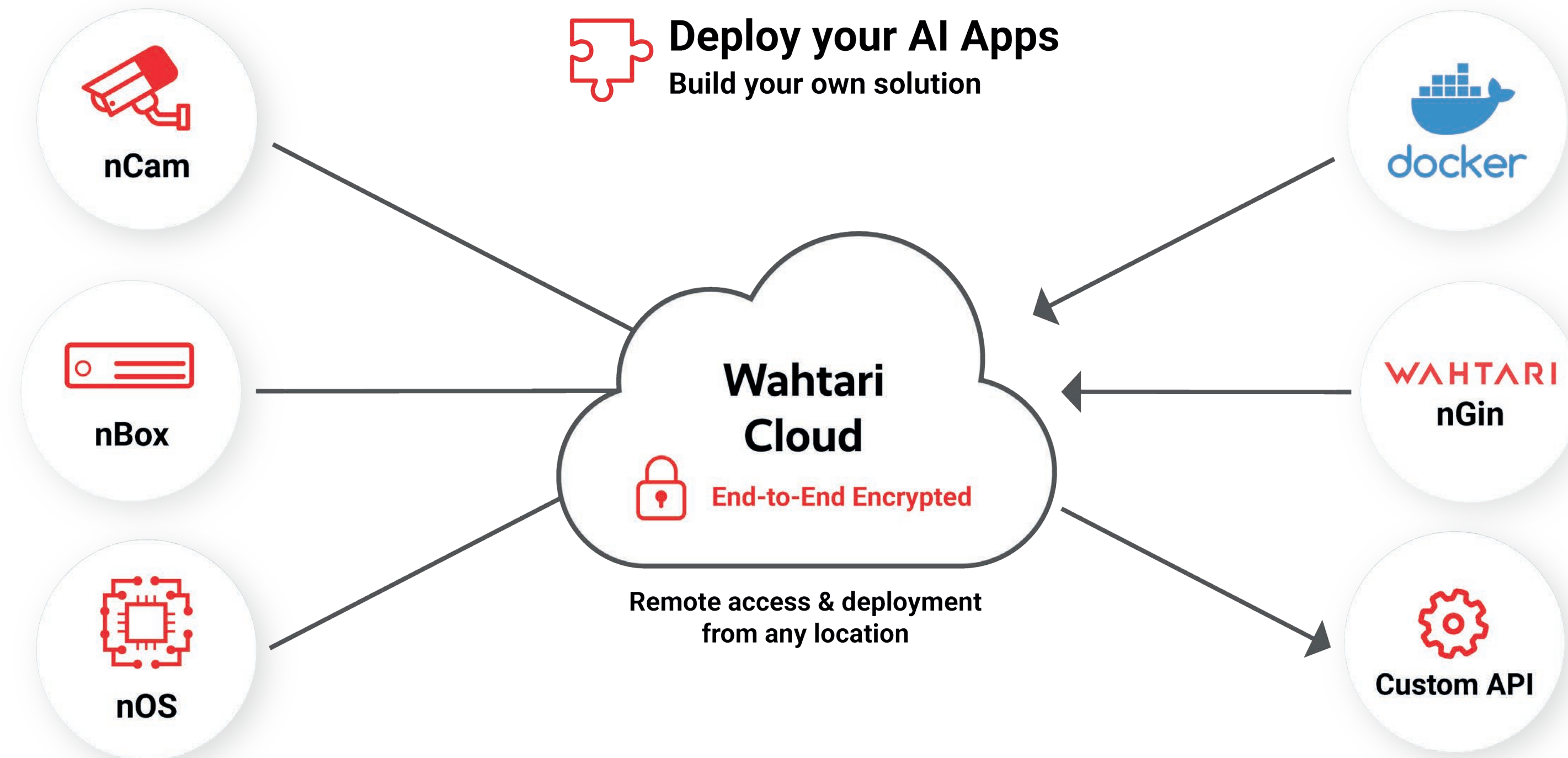


- Control all camera sensor features
- Manage & create Apps
- Overview system resources
- Adjust lighting, zoom/focus, IR...

# WAHTARI



# Wahtari nCam Software | Connection



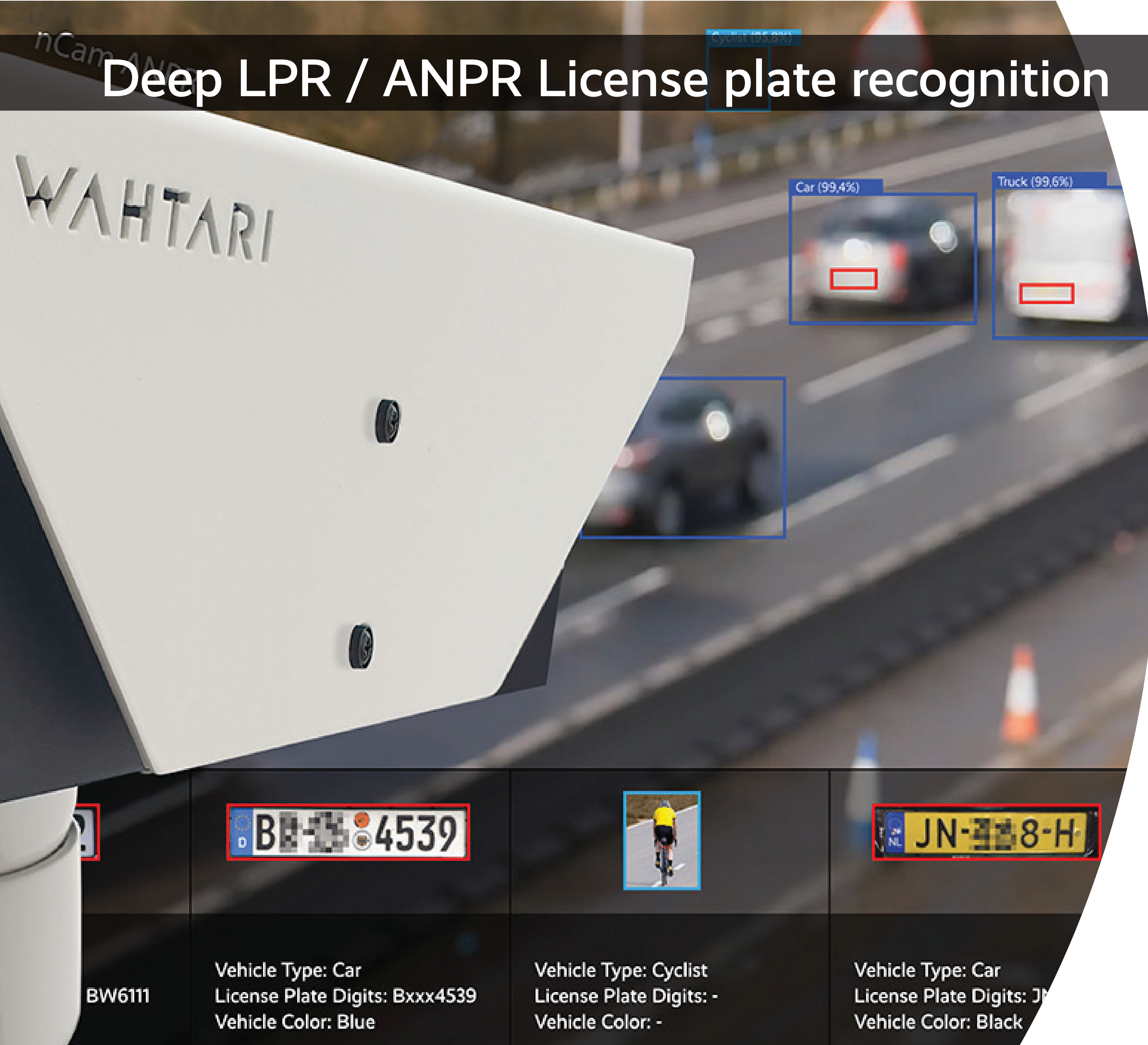
**WAHTARI**



# Customer projects



# Deep LPR / ANPR License plate recognition



## Scope

- Accurate identification of license plates and traffic management
- Diverse fields of applications:  
E.g automated payment, automated access control, measure traffic density, detect offenses

## Challenge

- Extreme viewing angles, light and weather conditions
- Non- unified standards and design of license plates

## Outcome

- Extreme high accuracy: Over 99%
- Highspeed detection within 50-100ms
- Onboard processing: Evaluation of the license plates within the camera

**WAHTARI**



# Inspection of FFP2 face masks

## Scope

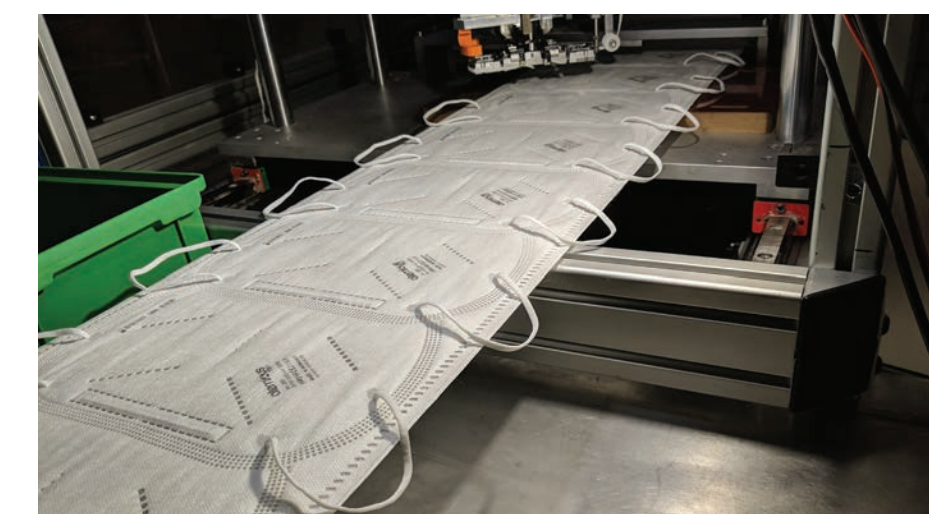
- Accurate detection and sorting of defective masks
  - Identify material flaws, imprint errors, ripped ribbons and falsely placed metal hooks
- No existing solution: human-based

## Challenge

- Very complex situation
- Masks are not evenly distributed /fixed

## Outcome

- Image analysis within 25ms
- Detection rate > 98%



**WAHTARI**



# Inspection of roof tiles

## Scope

- Accurate detection and sorting of broken tiles
- Identify cracks, spalling and discolorations
- Non existing solution: human-based

## Challenge

- Many different types / models
- No documented standards  
-> based on human intuition

## Outcome

- Image analysis within 25ms
- Detection rate > 99%



**WAHTARI**



# Inspection of conveyor belts

## Scope

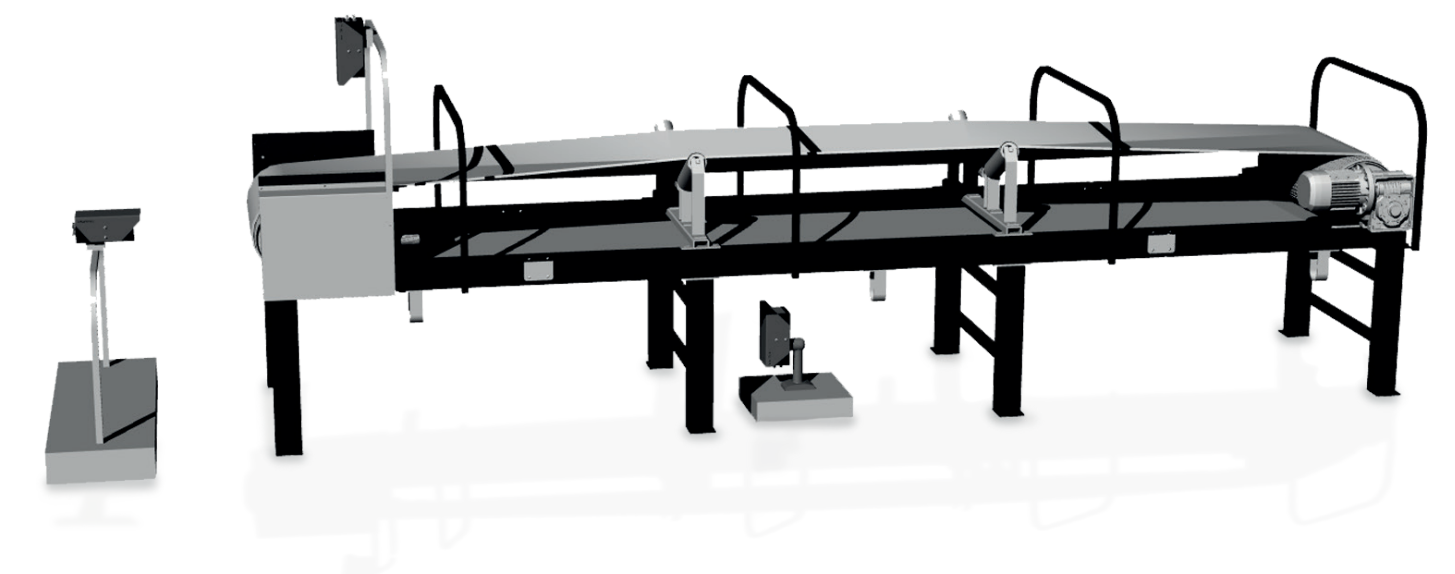
- Accurate detection of wear and tear on conveyor belts
- Identify cracks, breaks, cuts, holes, shreds

## Challenge

- Monitor thousands of kilometers
- Fluctuating visual conditions

## Outcome

- Highspeed detection:  $<15\text{m/s}$  belt speed
- Autonomous operation and alarm trigger



**WAHTARI**



# Bark inspection of tree logs



## Scope

- Detect residual bark
- Measure length

## Challenge

- Operating environment: change of seasons and varying light conditions
- Different types of bark
- Outdoor operation, dusty, foggy

## Outcome

- Highspeed detection:  $<15\text{m/s}$  belt speed
- Autonomous operation and alarm trigger

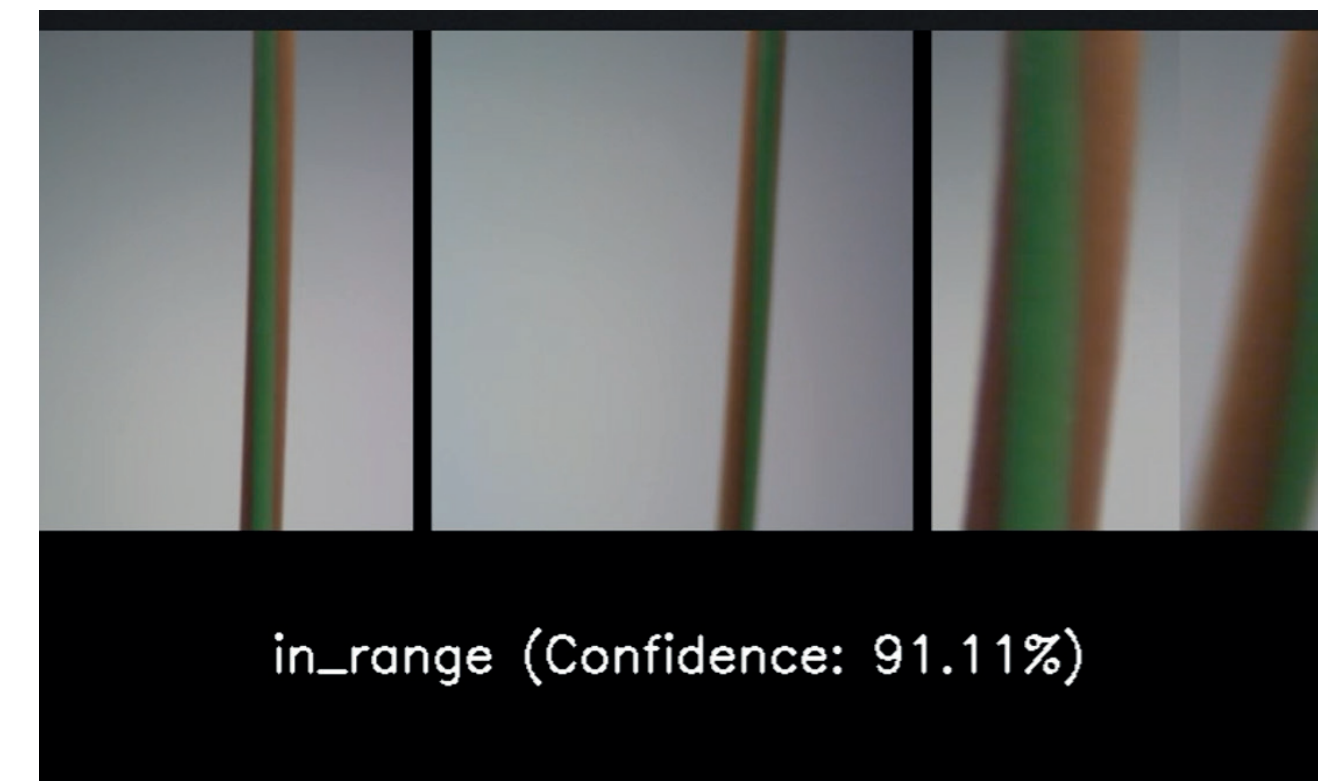
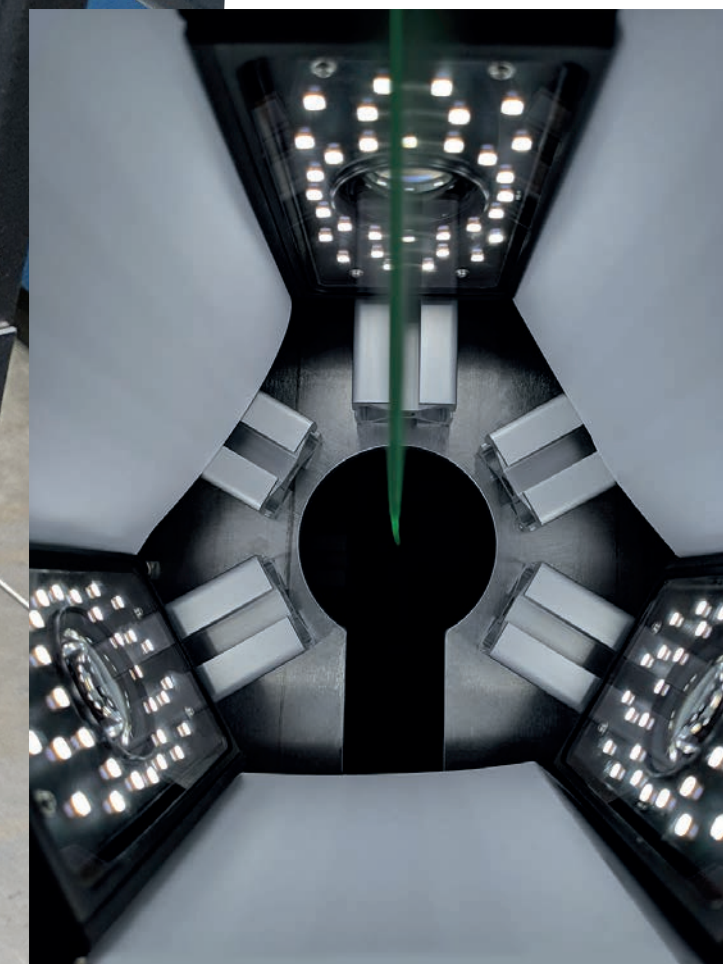




# Modular ecosystem makes custom solutions easy: nLine

## Quality control for tube- shaped products

- AI that learns exclusively with „gold-samples“
- 3 cameras for 360 degree coverage
- Essentially 3nCam mini + nBox
- Supports high manufacturing speeds of up to 1500m/min
- Simple UI & API
- No special employee training required





## How we leverage Intel's products



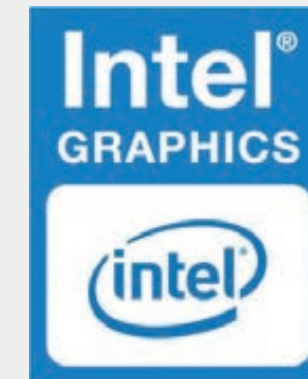
## Intel inside nCam

**CPU**



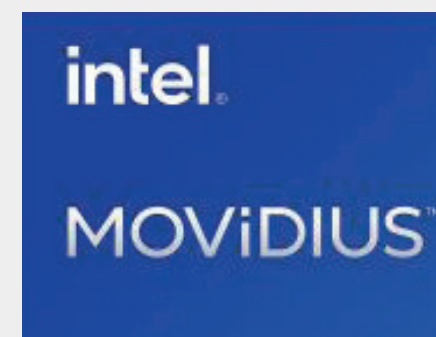
Intel Atom  
x86\_64 prozessor

**iGPU**



Video encoding

**VPU**



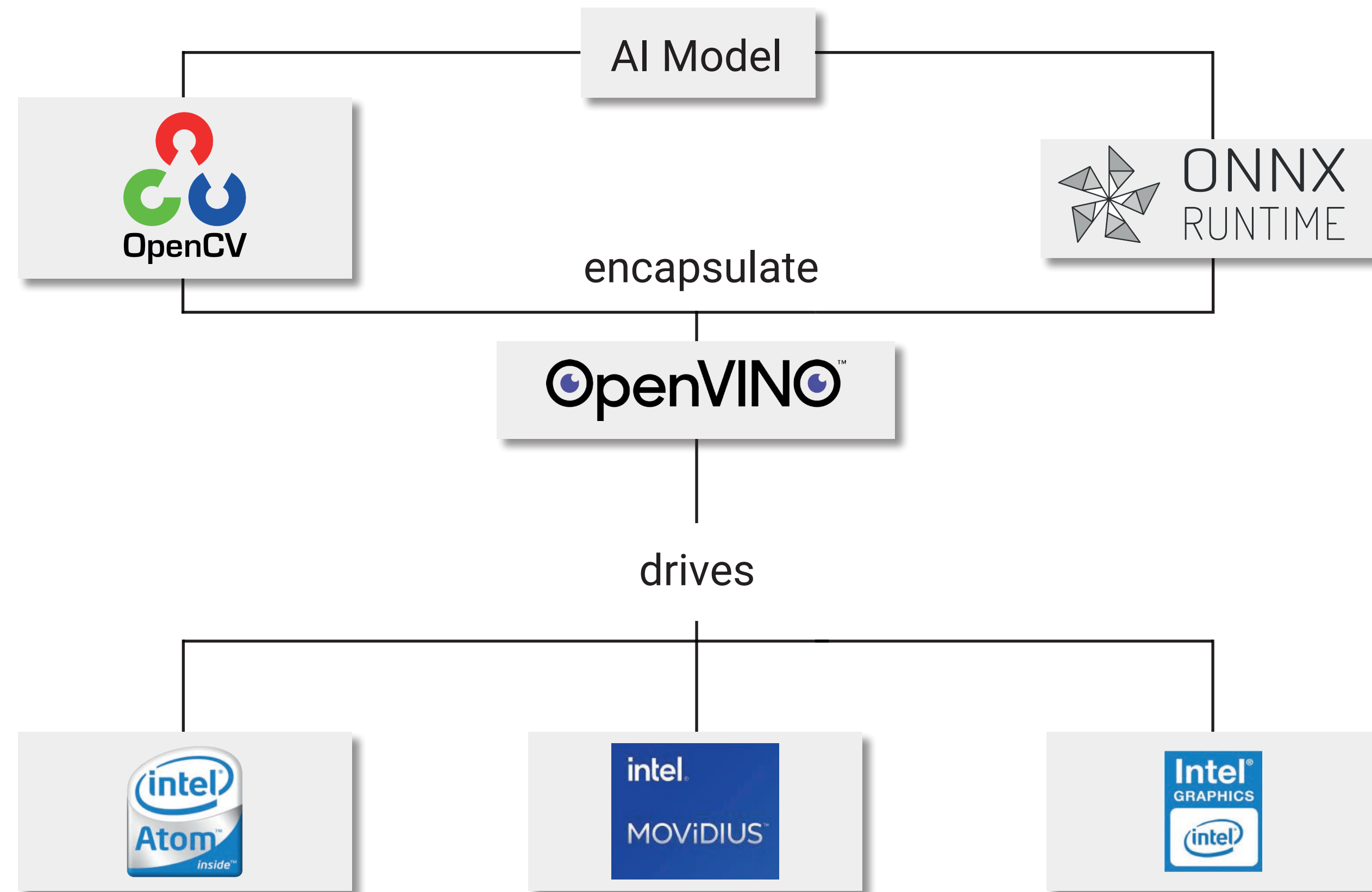
Intel AI Accelerators  
Movidius Myriad X

**AI model  
training**





# Intel software



- OpenCV offers traditional image and light ML operations
- OpenCV/ORT make loading AI model straightforward
- OpenVINO for maximizing performance



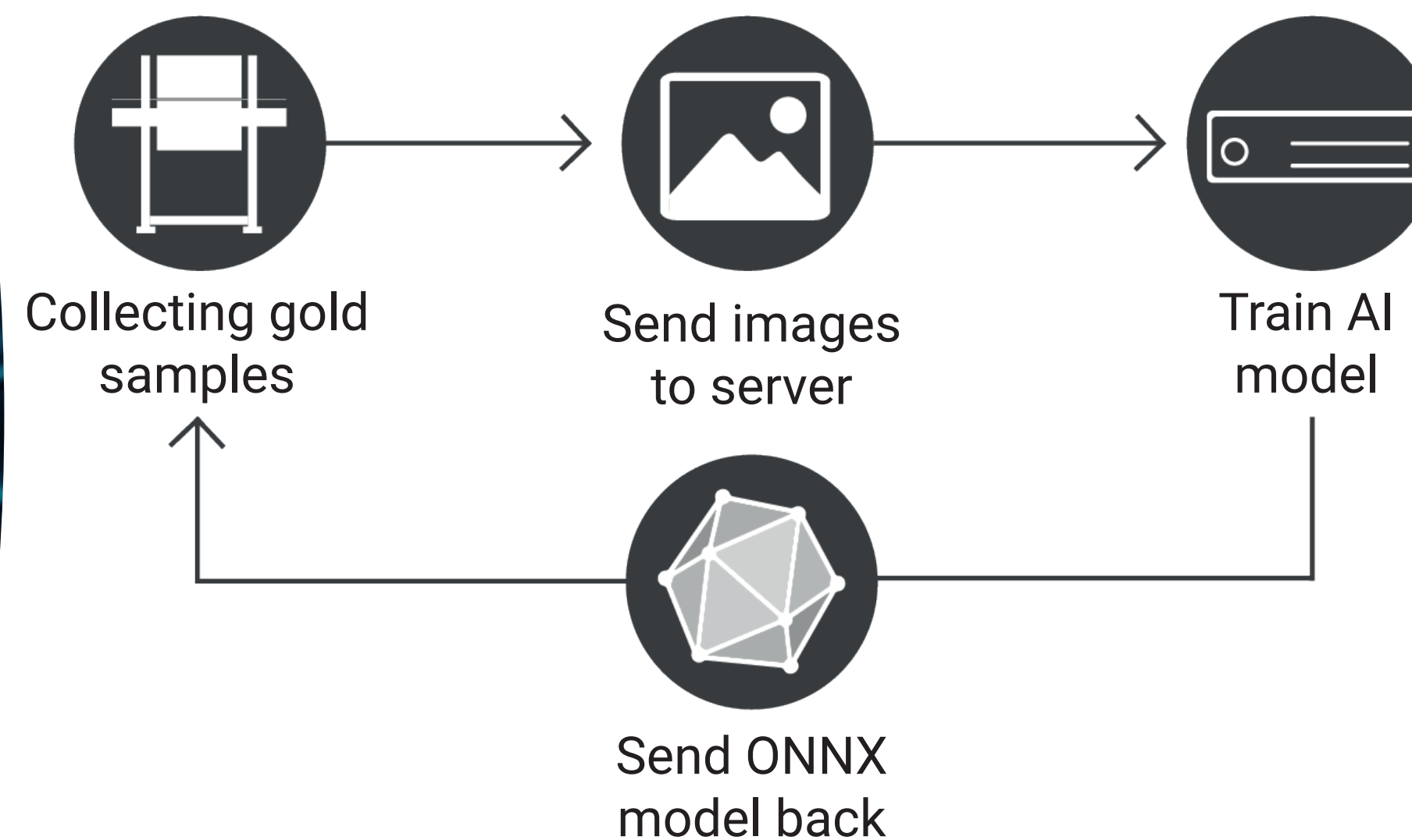
# OpenCV inference example

```
228 //#####//
229 /// Inference Routine ///
230 //#####//
231
232 // inference_routine reads in an endless loop frames off of the inference queue.
233 // It forwards each frame through its detection model and pushes the results to
234 // the inference result queue.
235 // Results are probably shown for a different frame than they have been created for.
236 // However, these slight deviations in timing are not important in this sample.
237 void inference_routine(string model_path, string config_path) {
238     cv::dnn::DetectionModel dm = cv::dnn::DetectionModel(model_path, config_path);
239     dm.setPreferableBackend(cv::dnn::DNN_BACKEND_INFERENCE_ENGINE);
240     dm.setPreferableTarget(cv::dnn::DNN_TARGET_MYRIAD);
241
242     Mat frame;
243     vector<int> classIDs;
244     vector<float> confs;
245     vector<Rect> boxes;
246
247     while (true) {
248         if (interrupted()) {
249             return;
250         }
251
252         {
253             unique_lock<mutex> lock(infQueueMx);
254
255             // Wait, if the queue is empty.
256             if (infQueue.size() == 0 && !infQueueCond.wait_for(lock, 100ms, [&]{ return infQueue.size() > 0; })) {
257                 // No frame available.
258                 continue;
259             }
260
261             // Frame is available.
262             frame = infQueue.front();
263             infQueue.pop();
264             if (frame.total() == 0) {
265                 // Ignore empty frames.
266                 continue;
267             }
268         }
269
270         // Do inference.
271         dm.detect(frame, classIDs, confs, boxes);
272
273         // TODO: show class and confidence as well?
274         infResQueueMx.lock();
275         // Drop oldest result when full.
276         if (infResQueue.size() == FPS) {
277             infResQueue.pop();
```



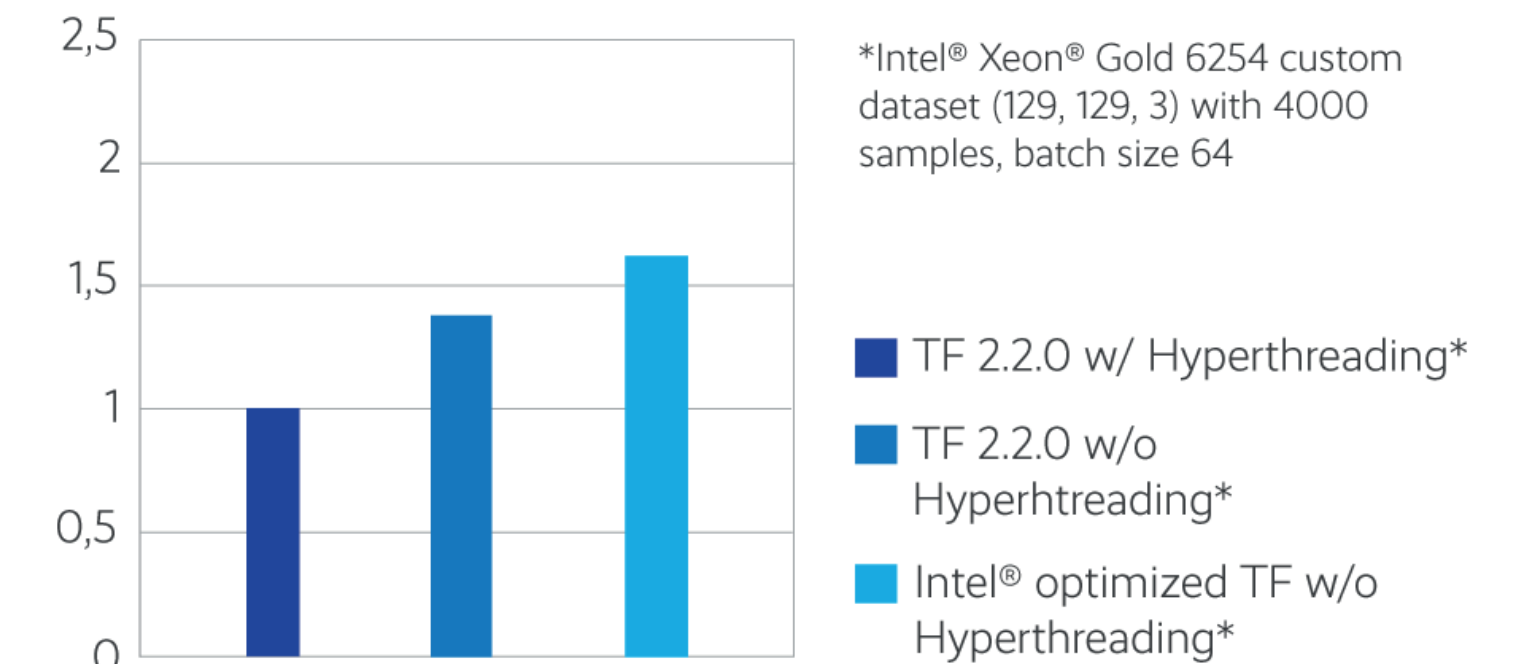
# Example: nLine

Training of AI models is computationally expensive  
Many manufacturing sites already have an IT infrastructure

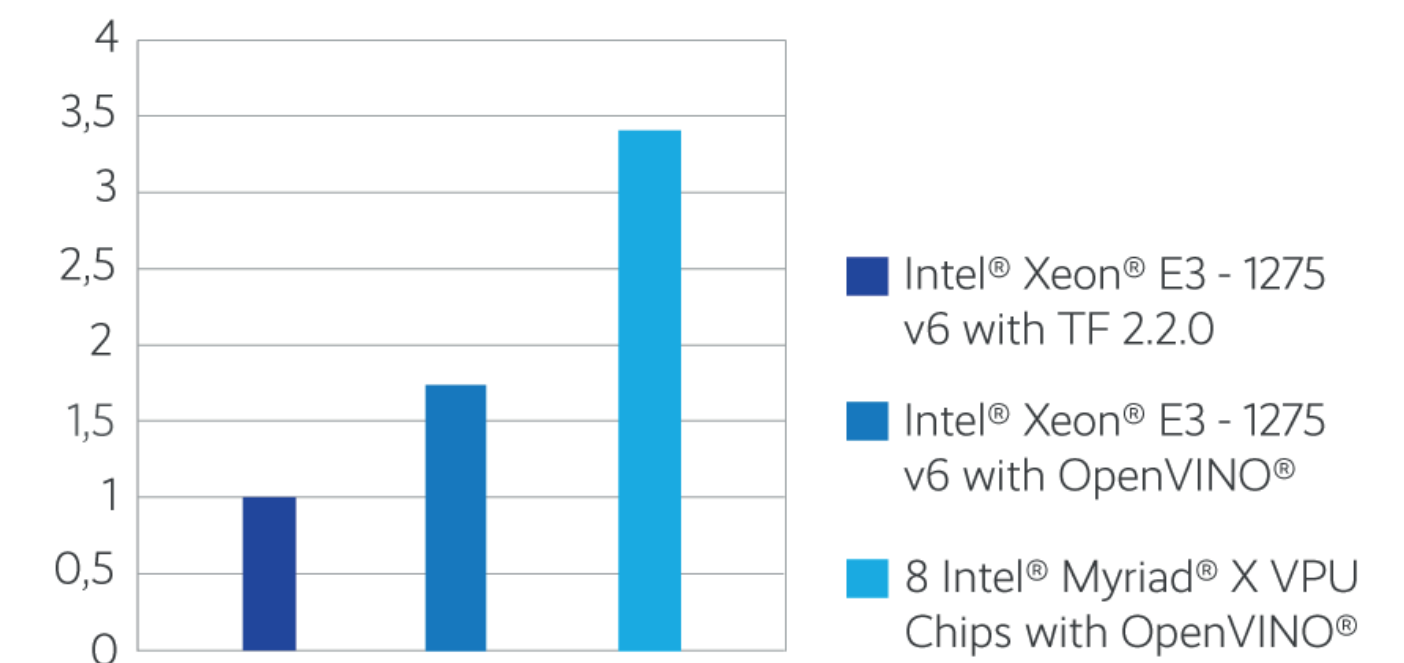


Powered by Intel® Xeon® and Intel® Movidius™ Myriad™ X VPU

Normalized training performance for avg. epoch  
Duration @ 4000 samples



Normalized inference performance for latency and FPS

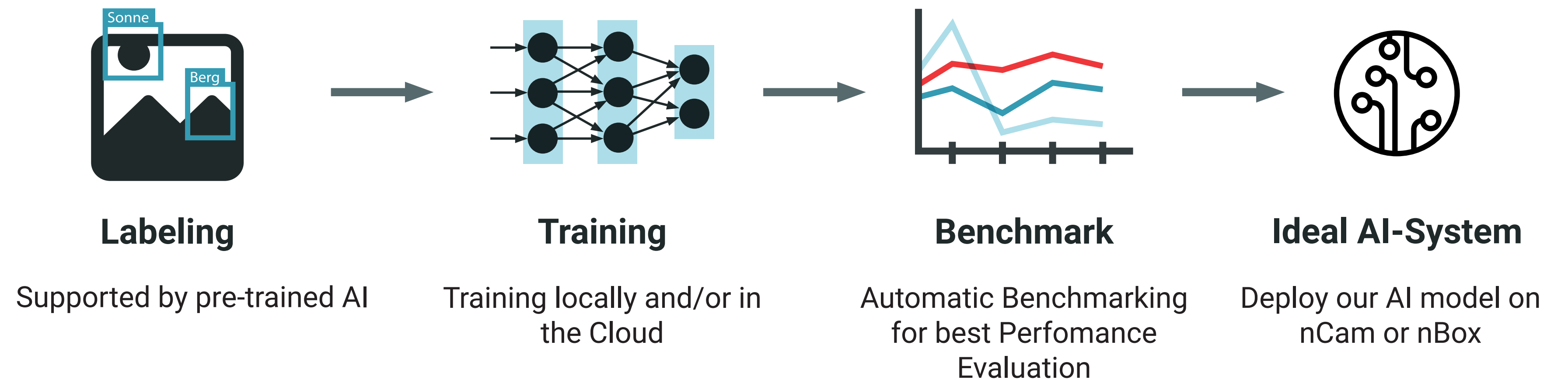




What's expected next?



# nLab Autonomous AI platform. Hybrid System



- Guides you through the DL Cycle
- AI supported Labeling
- Support of all major Frameworks
- Supports Intel Xeon Scalable
- Processors ( + Ice Lake)



# nGin Scripting: nScript

- Dynamic scripting language
- Arduino-like simplicity
- Performant (Go-Backend)
- Executed on camera
- Control camera pipeline
- Manage AI-Models
- Customize result API
- Minimize network traffic


```
1  /* nCam Script */
2
3  // Import the nGin module.
4  ngin := import("ngin")
5
6  // Load a detection model.
7  dt := ngin.NewDetectionModel("personDetection")
8  // Load a classification model.
9  cl := ngin.NewClassificationModel("faceClassifier")
10
11 // Define an API interface to publish the results.
12 api := ngin.NewHTTPJSONApi("https://result.server.com")
13
14 // Define a frame throttler.
15 th := ngin.NewThrottler(30)
16
17 // Set a handler that handles new frames from the camera.
18 // Throttle to not analyze every single frame.
19 ngin.OnFrame(func(frame) {
20     if th.Throttle(frame) {
21         return
22     }
23
24     dt.Detect(frame)
25 })
26
27 // Set a handler for the detection model.
28 dt.OnResult(func(result) {
29     if result.Ok {
30         cl.Classify(result.frame)
31     }
32 })
33
34 // Set a handler for the classification model.
35 cl.OnResult(func(result) {
36     if result.Ok {
37         api.Publish(result)
38     }
39 })
40
```



## Contact:

 [team@wahtari.io](mailto:team@wahtari.io)

 +49 89 21 54 3000

 Wahtari GmbH  
Münchener Str. 14  
85540 Haar  
Germany

 [wahtari.io](https://wahtari.io)

Follow us:  

