

The logo for the 2024 embedded VISION SUMMIT is centered on the left side of the slide. It features a white octagonal background with a colorful, multi-layered border in shades of purple, blue, green, yellow, and orange. The text "2024" is at the top, "embedded" is below it, "VISION" is in large, bold, dark blue letters with a gradient, and "SUMMIT" is at the bottom in a smaller, dark blue font.

2024
embedded
VISION
SUMMIT®

Build a Tiny Vision Application in Minutes with the Edge App SDK

Dan Mihai Dumitriu
Chief Technology Officer
Midokura (Sony Group)

Sony's AITRIOS™

Challenges of Embedded Development

Case for Polyglot Development

Dev Tools & SDK

Example Applications

Conclusion

* AITRIOS and AITRIOS logos are the registered trademarks or trademarks of Sony Group Corporation or its affiliated companies.

Targeting solution developers for various vertical applications.



Smart Home



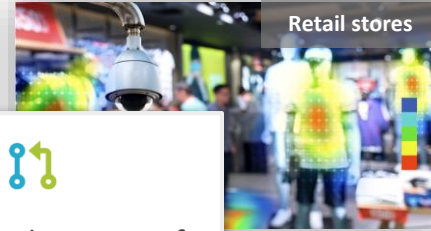
Bring to market easy-to-use sensing devices



Advanced vehicle assistance



Low barrier of entry for solution developers



Retail stores



Agile development of sensing applications



Smart Cities



Low operational cost of vision sensing apps



Polyglot Development



Marketplace to connect AI Developers & Solution Developers



Smart Manufacturing

Challenges of Embedded Development

Safety on hardware without MMU (i.e. MCUs)

- Ensure memory safety
- Dynamic loading and linking
- Replacing modules at runtime

Enable solution developers

- Reduce the development effort
- Enable portability across hardware platforms
- Provide a channel for distribution via marketplace

Enable OEM ecosystem using Android model

- Give away the device OS and/or runtime
- Lower the R&D costs of device manufacturers
- Enable marketplace model by application standardization
- Access to marketplace raises value of hardware product

Pain points of traditional embedded/IoT devices

Development

- Embedded development is difficult in C
- No standard component model, so customization cost is high

Maintenance

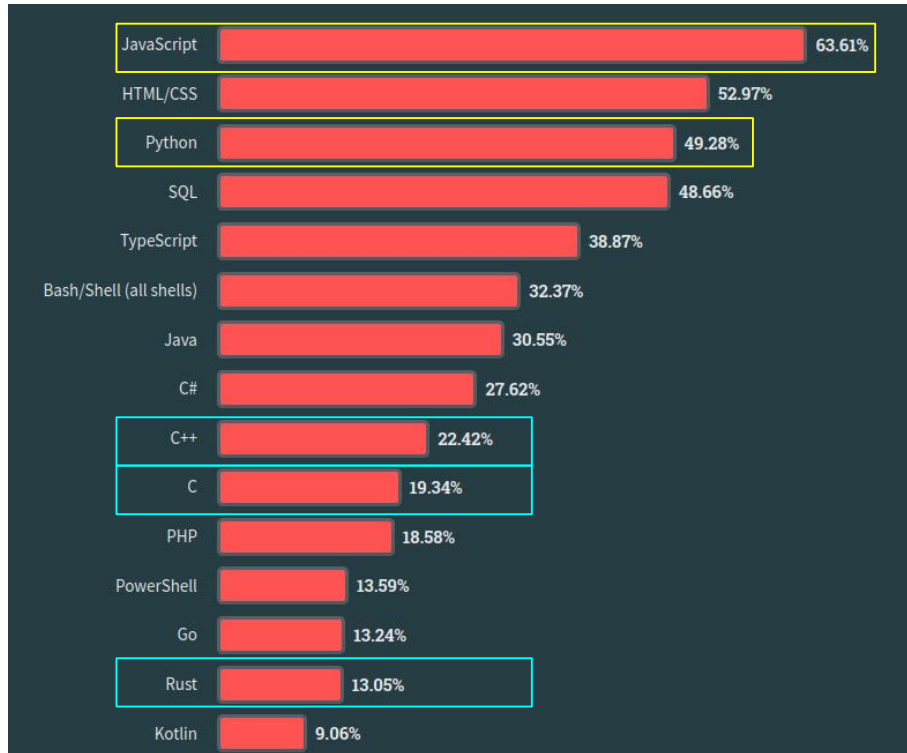
- Difficult or impossible to change functionality after deployment

Security

- Many instances of compromised IoT devices
- Vulnerabilities cannot all be found before shipping
- #1 source of security vulnerabilities is memory bugs

Why Polyglot Development?

Most popular programming, scripting, and markup languages



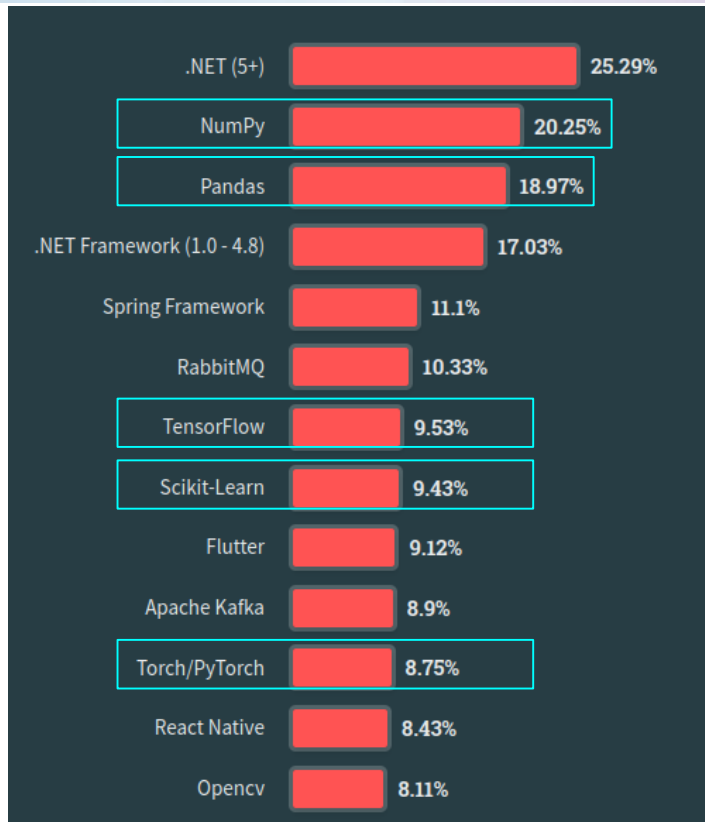
JavaScript and Python are far more popular than C++, C, and Rust

Embedded systems are primarily written in C

There is a mismatch...

Data extracted from <https://survey.stackoverflow.co/2023>

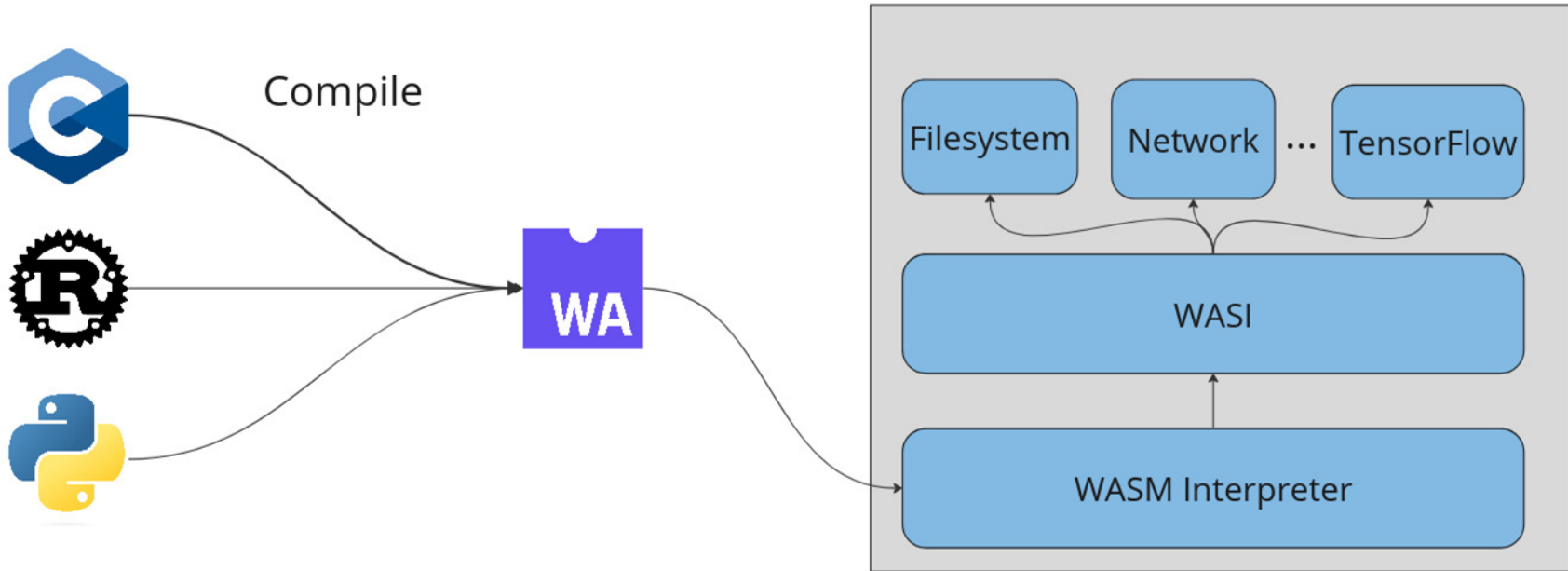
Most popular AI frameworks are Python



Most frameworks related to data or AI are Python-specific

- NumPy
- Pandas
- PyTorch

Decouple from language, target architecture & OS



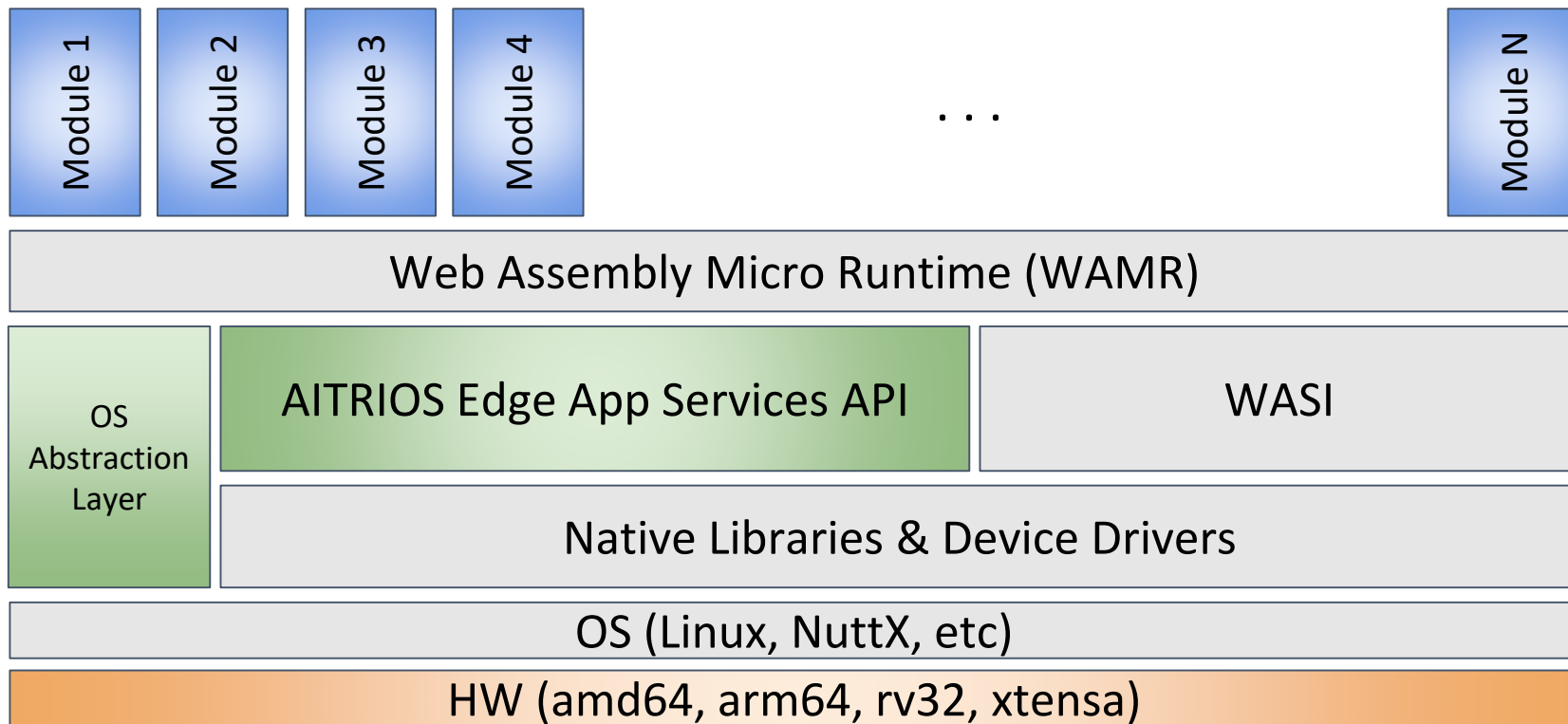
WebAssembly System Interface (WASI): standardized and secure way to access system resources

EVP Edge Stack

EVP (Edge Virtualization Platform)

- Like Kubernetes, but for tiny IoT devices
 - Lifecycle management of workloads on IoT devices
- EVP agent is like Kubelet
 - Leverages WebAssembly Micro Runtime (WAMR)
 - Strong isolation of modules
 - Secure even on RTOS
- Communicates with EVP backend via MQTT

EVP device stack



Edge app SDK: APIs

WASI-Sensor

- Read Image
- Configure (e.g., frame rate)

WASI-NN

- Load model
- Run inference

Communication

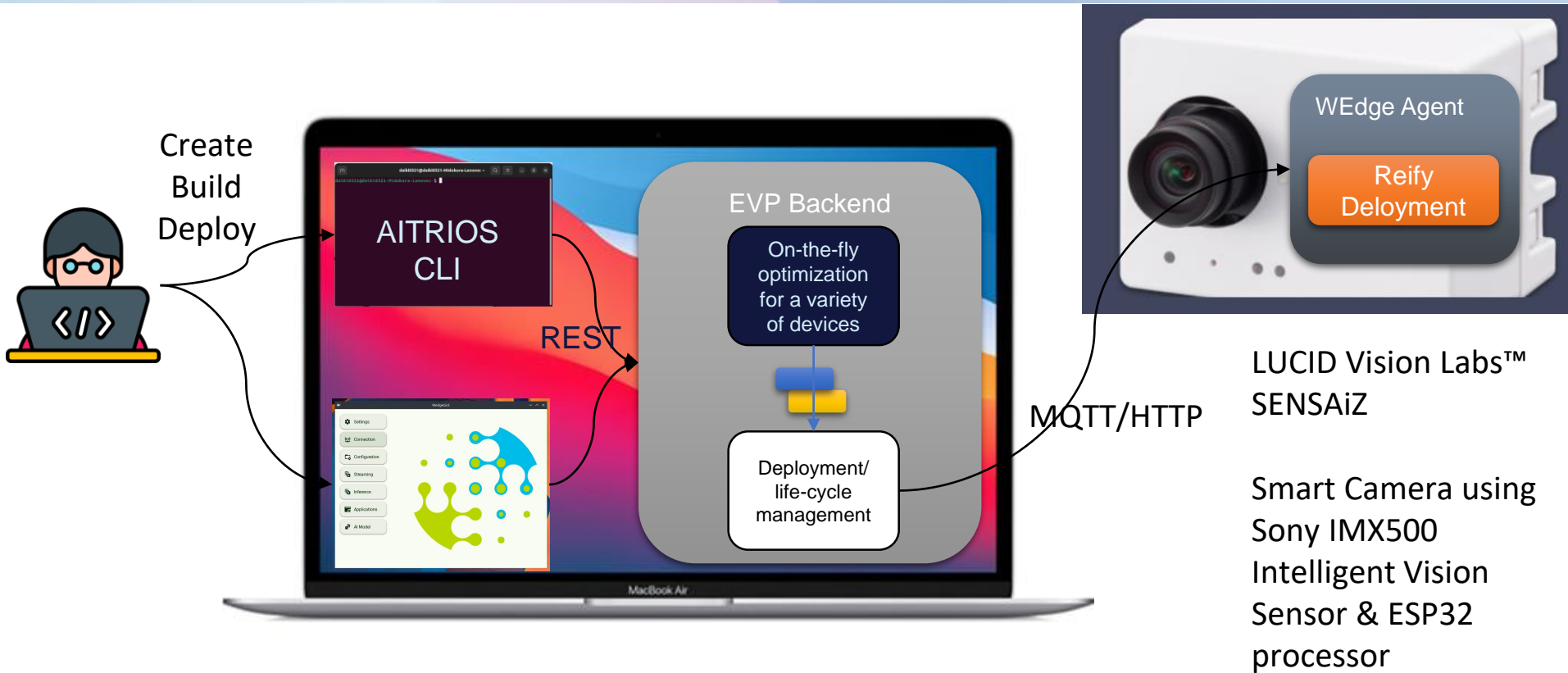
- Telemetry to EVP backend
 - via MQTT
- Device to device
 - Sockets

Data Storage

- Local DB
- Blob storage (HTTP PUT/GET/POST)

AITRIOS Edge App Dev Tools

Local developer setup

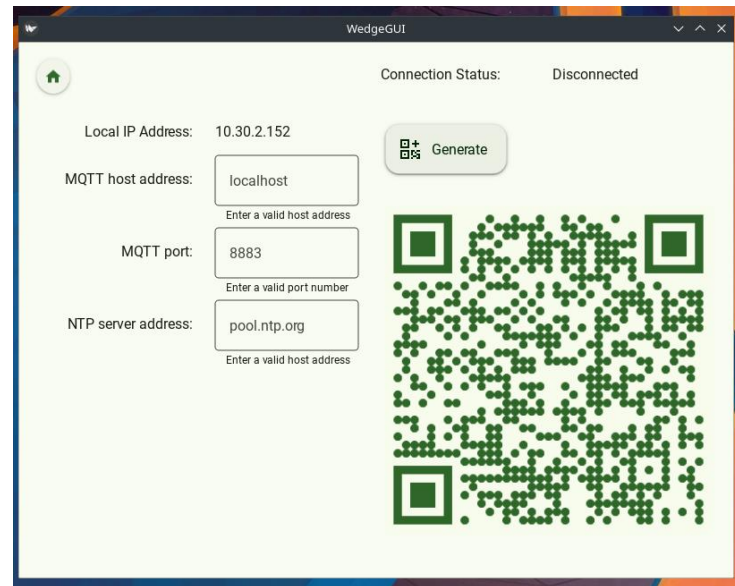
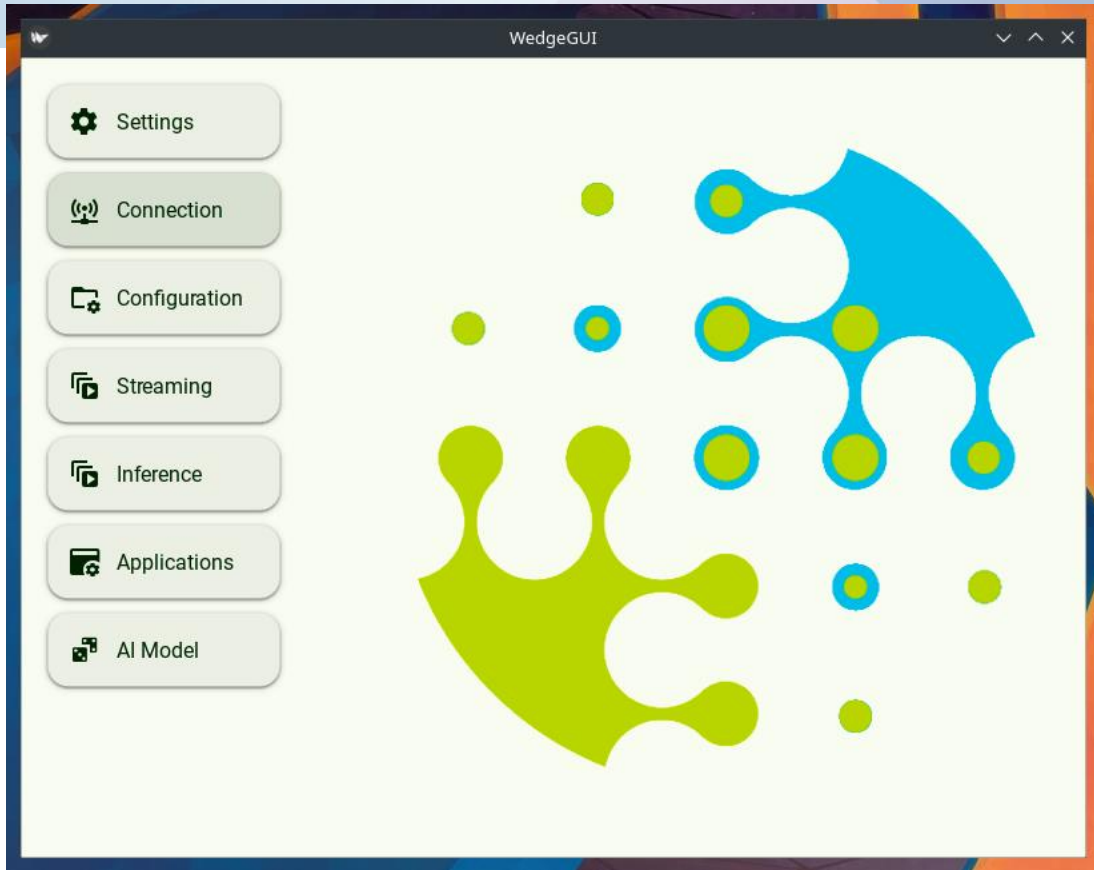


AITRIOS developer tools

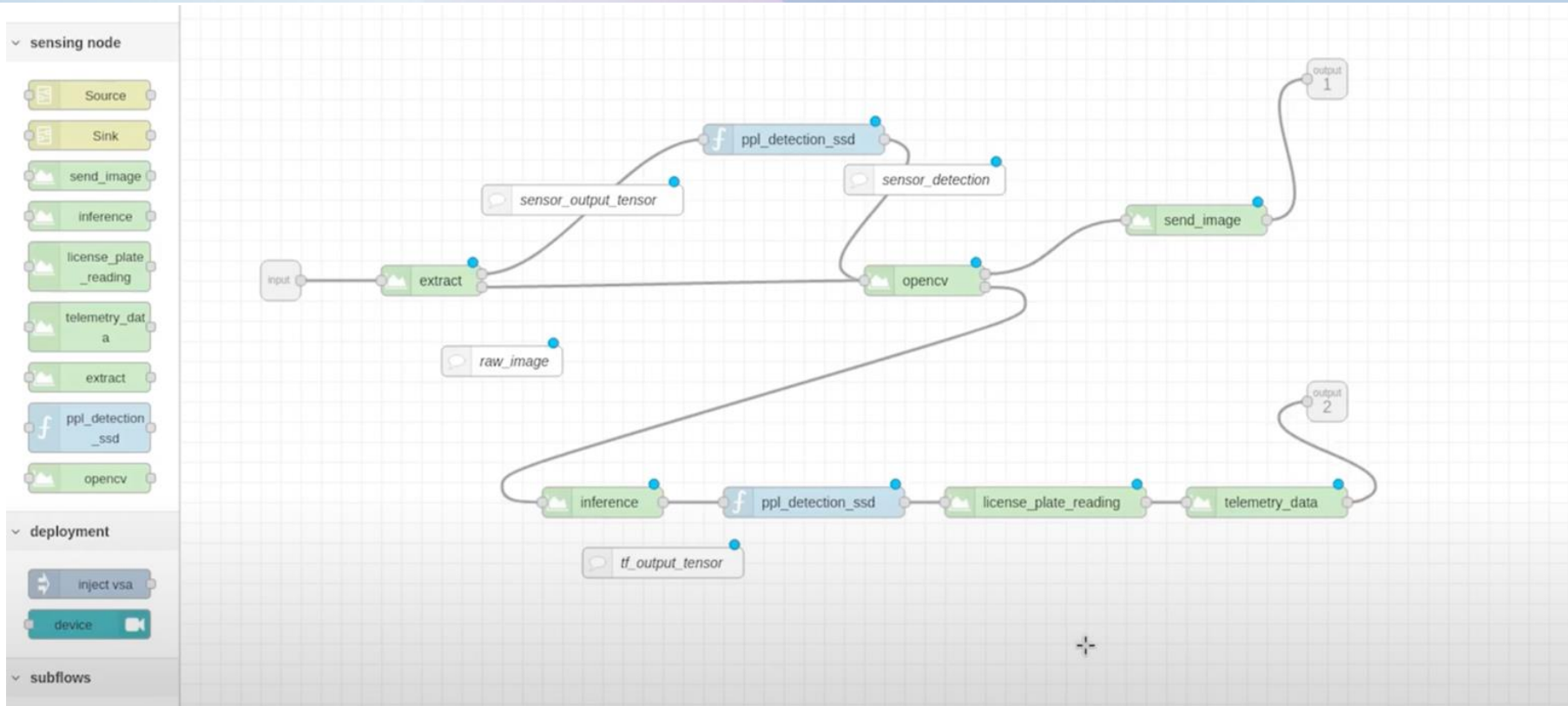
- EVP-in-a-Box (Backend)
- EVP CLI
 - Create module project
 - Build module
 - Register module
- EVP GUI
 - Manage device
 - Create edge application
 - Do test inference



EVP GUI

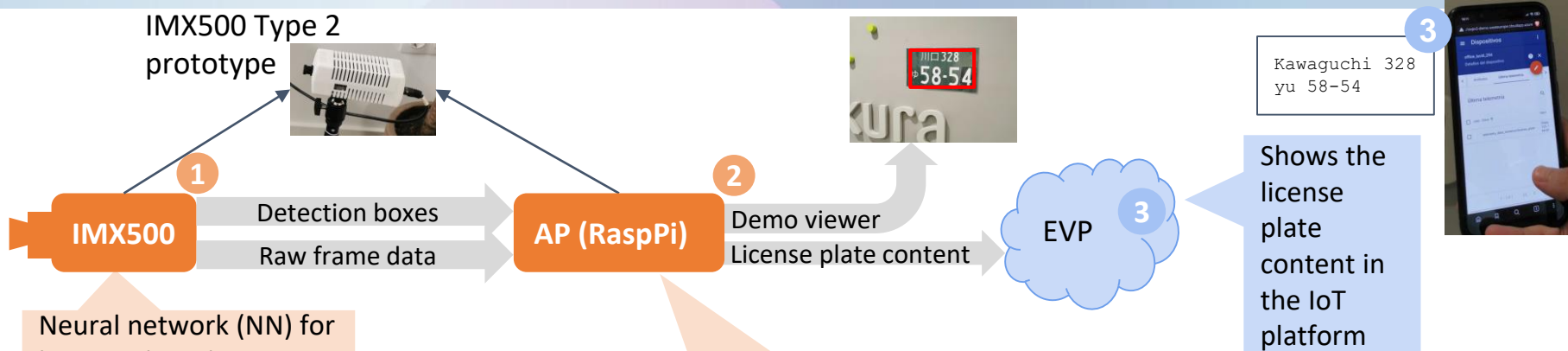


Visual programming

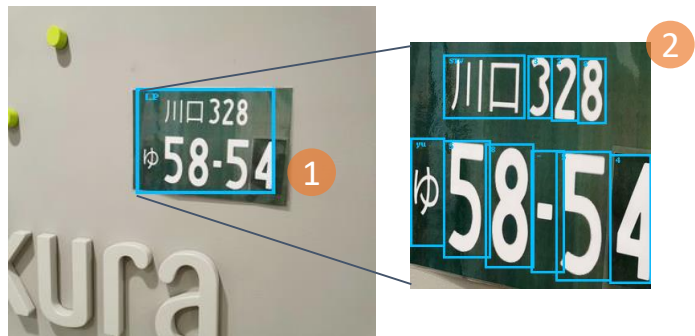


Example Edge App

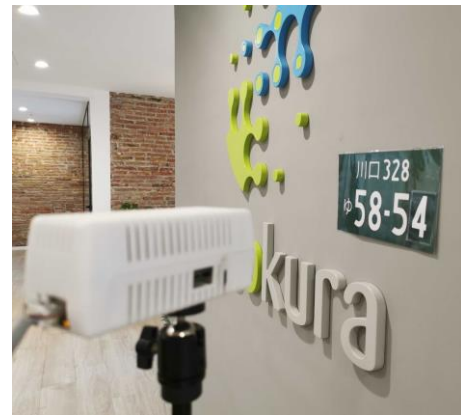
License plate reading (LPR) edge app



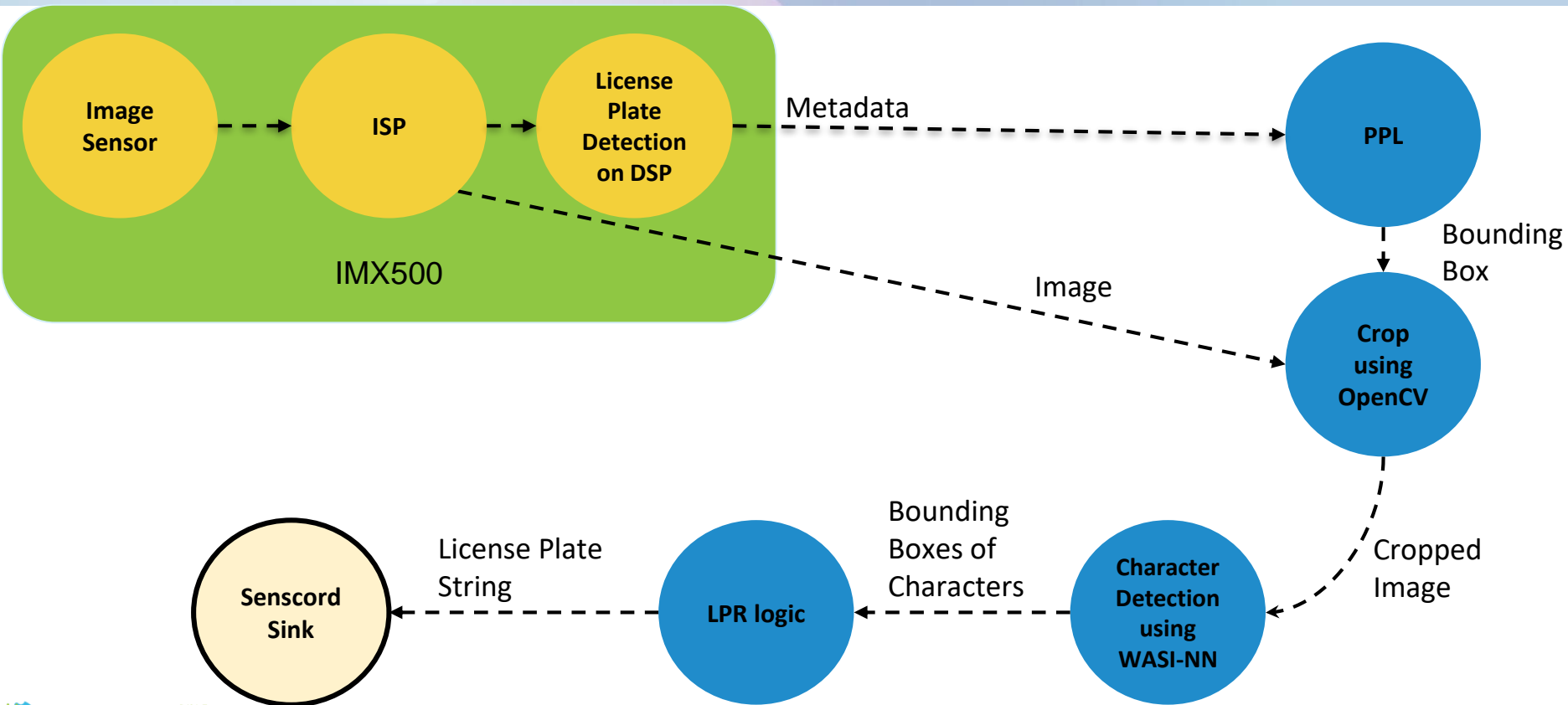
Neural network (NN) for license plate detection (LPD)



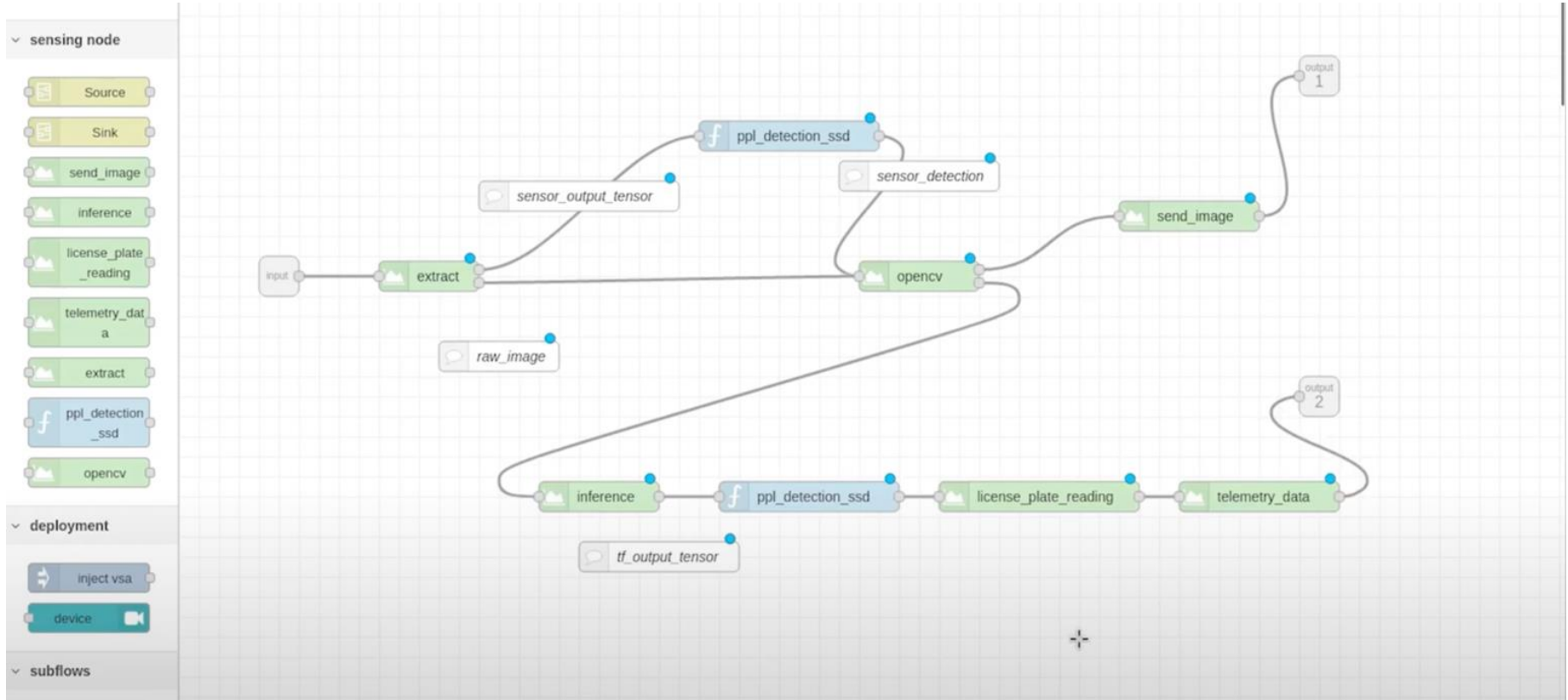
- Creates high-resolution crop of the detection
- Applies 2nd NN to detect characters/numbers of license plate (LPR network, in tf-lite runtime)
- Reads license plate content from character detections



License plate reading data flow



License plate reading data flow



License plate reading demo result

The interface is divided into two main sections. The left section is a block diagram on a grid background. It features a vertical toolbar with various functional blocks under 'common' (inject, debug, complete, catch, status, link in, link call, link out, comment) and 'function' (function, switch, change, range, template, delay, trigger, exec, filter). A block labeled 'opencvimage viewer' is connected to a camera icon. The viewer displays a license plate image with a red bounding box around the text 'Sakai 979, tsu 33-21'. The right section is a 'MyDevice' interface with a dark blue header. Below the header is a navigation bar with tabs: 'Details', 'Attributes', 'Latest telemetry', 'Alarms', 'Events', 'Relations', and 'Audit Logs'. The 'Latest telemetry' tab is active, showing a table with the following data:

<input type="checkbox"/>	Last update time	Key ↑	Value
<input type="checkbox"/>	2022-11-29 18:19:05	telemetry_data_instance/license_plate	Sakai 979, tsu 33-21

Below the table, a large white box displays the text 'Sakai 979, tsu 33-21'.

- Polyglot development for embedded systems is important
 - AI developers use Python
 - Embedded developers use C
- WebAssembly is a great abstraction layer
 - High performance
 - Multi-platform
 - Scales up and down
- Edge Apps can be built by anyone

Further information

Midokura

<https://www.midokura.com>

AITRIOS

<https://www.aitrios.sony-semicon.com/en/>

IMX500

<https://developer.sony.com/develop/imx500/>

Demo video

<https://bit.ly/mido-vsa-demo>

Questions



midokura ミドクラ

A Sony Group Company