

List the strategic IoT edge computing portfolio projects for Helveticore

To build your credibility with network operators and GreenTech players in Switzerland and Estonia by mid-2027, your portfolio must not look like a list of academic projects. It must be presented as a catalogue of proprietary, industrialized technology bricks, ready to be deployed as integration mandates.

Here is the strategic structure of your R&D portfolio for Helveticore, built around your three major technical pillars:



## The Strategic IoT / Edge Computing Triptych (2026-2027)



### Project 1: EdgeStream-GW (IoT Gateway & Edge Analytics)

An ultra-lightweight industrial software gateway designed to process data streams at the network edge.

**Commercial Positioning: Energy efficiency and bandwidth optimization for smart meter networks (Smart Grid).**

**Technical Architecture:**

- Containerized Docker engine for "zero-configuration" deployment on an embedded Linux target.

- Asynchronous Python processing pipeline applying adaptive filtering and noise reduction to data frames.

- Ultra-lightweight local persistence to guarantee resilient operation in disconnected mode.

**Impact KPI (Key Indicator): A measured 75% reduction in data volumes transmitted to the Cloud and critical alert latency below 50 ms.**



### Project 2: Fog-Crypto Core (FPGA Hardware Accelerator)

A Hardware/Software co-design module dedicated to offloading critical computations and securing IoT nodes.

**Commercial Positioning: Protection of data integrity at the source for sensitive industrial infrastructures.**

**Technical Architecture:**

- Development of optimized logic blocks in SystemVerilog / Verilog for FPGA architectures.

- Low-level communication mechanism via high-speed buses (SPI / PCIe) to interface the accelerator with a host processor (embedded Linux board).

- Hardware implementation of encryption algorithms or real-time mathematical signal pre-processing.

**Impact KPI: A 10x reduction in main gateway CPU load and line-rate encryption of data streams.**



### Project 3: NetSentry-IoT (Active Network Cybersecurity Probe)

An autonomous software probe for anomaly detection and IoT protocol auditing.

**Commercial Positioning: Regulatory compliance (NIS2 / GDPR) and cyber hardening of distributed industrial sensor fleets.**

**Technical Architecture:**

- Packet analysis engine based on Scapy for deep inspection of lightweight protocol layers (MQTT, HTTP, Zigbee).

Traffic generator and attack-source simulator to evaluate the robustness of local industrial firewalls.

Behavioral analysis algorithm that locally detects identity spoofing and denial-of-service (DoS) attempts.

**Impact KPI: Detection and isolation of malicious packets in under 5 ms at the edge, with no dependency on a Cloud signature database.**



## Integration into your materials (GitHub / LinkedIn / Website)

For each project in this portfolio, your documentation (whether written in English or French on [helveticore.ee](https://helveticore.ee)) must follow the same engineering rigor:

**The Specifications Document (Requirements):** Present the industrial constraints (e.g. resilience to network outages, memory limitations).

**The Virtualized Test Bench (Testbench):** Explain how you validate your code by emulating, for example, 50 simultaneous energy meters using your local scripts.

**Performance Graphs:** Display latency and resource-consumption curves to prove the efficiency of your technology bricks.

### Module 7: Edge Deployment and Gateways with Raspberry Pi 4

- **7.1 Deploying the Local Broker:** Setting up the Raspberry Pi as the central communications hub. Students will install and configure the free, open-source **Eclipse Mosquitto** broker, which easily runs on single-board computers like the Raspberry Pi.
- **7.2 Building the Application Layer:** Installing and running **Node-RED** directly on the Raspberry Pi. Because Node-RED is a highly lightweight framework built on Node.js, it is perfectly suited for the Raspberry Pi, allowing students to build API routes and MQTT workflows using a drag-and-drop interface.
- **7.3 Handling Edge Memory Constraints:** Exploring the hardware realities of edge computing. Students will learn how to manage data storage when the Raspberry Pi loses internet connection, including how to configure devices to locally store telemetry data within limited memory constraints and automatically stream it to the server once the connection is restored.
- **7.4 Real-World Hardware Integration (Solar/Energy Case Study):** Utilizing the Raspberry Pi to communicate with external industrial hardware. Students will explore how a Raspberry Pi can run custom software (such as Venus OS) to connect via MQTT with solar energy components, such as Victron inverters and batteries, acting as a local orchestration hub for smart energy routing.