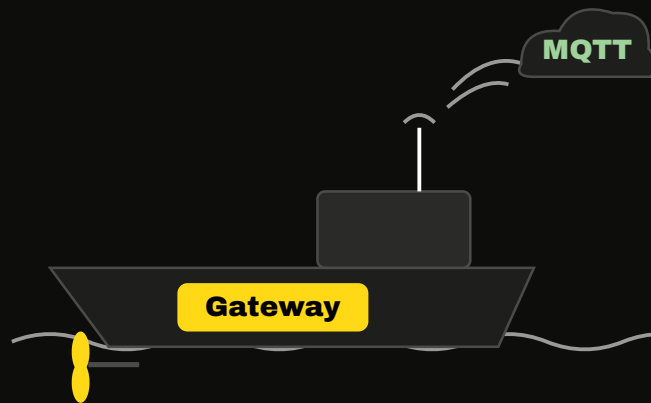


SECTOR: NAVAL PROPULSION

Propulsion data from every ship. Without touching the control.

A Linux PLC on board reads the propulsion control, listens to NMEA 2000 and sends the data to shore over MQTT.

[Read the interactive case study online ›](#)



Schematic illustration

Read only

toward the propulsion control: the gateway never sends a command

2 + 2

Ethernet and RS-485 ports for Modbus TCP and Modbus RTU

1

CAN board option to listen to NMEA 2000 or J1939

1 per ship

the same gateway and software image on every installation

THE CHALLENGE

Data locked inside the propulsion control.

Makers of propulsion systems (propellers, thrusters and their controls) install their equipment on ships that sail all over the world. The control system knows pitch, shaft speed, hydraulic pressures and alarms, but that data stays on board. To support the crew, plan service and design the next system, each maker needs the data on shore, from every vessel, without adding risk to a system that moves the ship. That calls for a separate gateway: it reads, and it never commands.

01

Read the propulsion control

Pitch, shaft speed, setpoints, hydraulic values and alarms over Modbus TCP or Modbus RTU.

02

Listen to the ship

Engine and navigation data from NMEA 2000 or J1939, with a CAN board.

03

Send to shore

MQTT through the router of the vessel, with a local buffer while the satellite or 4G link is down.

04

The same unit on every ship

One gateway and one software image, with a configuration file per vessel.



Read only, by design

The gateway has no write path to the propulsion control: Modbus reads only, CAN listening only, and its own network segment. Commands and alarms stay in the propulsion control and the alarm system of the ship.

HOW IT WORKS

Three jobs: read the propulsion, send to shore, listen to the ship.

Job 1 · Read the propulsion control

Poll, check, timestamp.

The gateway polls the propulsion control over Modbus TCP, or over Modbus RTU on RS-485, at a fixed rate. Each value is scaled, checked against its range and timestamped. If the control stops answering, the gateway marks the values as stale instead of sending an old number as if it were new. Go from harbour to transit, then lose the answer.

- Pitch, shaft speed, hydraulic pressure, alarms
- Scaling and a range check per register
- A stale flag when an answer is missing

Job 2 · Send to shore

Keep it at sea, send it when the link is back.

The gateway publishes the values over MQTT to a broker on shore, through the router of the vessel: satellite at sea, 4G near the coast. The link comes and goes, so every message is stored first on the gateway and sent in order when the link returns. Values go out when they change, plus periodic summaries, to save bandwidth. Lose the satellite, then get it back.

- MQTT over TLS to the broker on shore
- A local buffer while the link is down
- Send on change, plus summaries

Job 3 · Listen to the ship

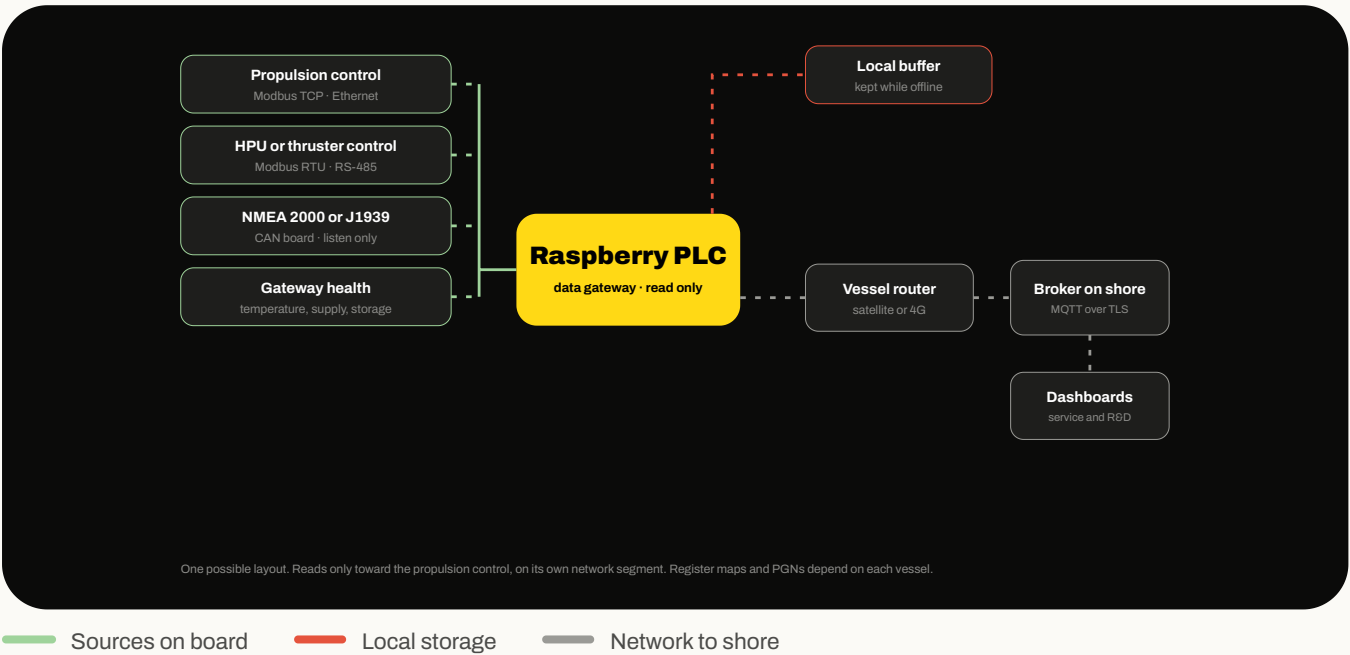
The data is already on the backbone. Listen, do not talk.

Engine speed, speed over ground and speed through the water are already on the NMEA 2000 backbone of many vessels. With a CAN board, the gateway listens to the parameter groups it needs (PGNs) and puts them next to the propulsion data, so pitch and power can be compared with the real speed of the ship. It only listens: it does not claim an address or send to the backbone. Compare two voyages.

- PGN 127488: engine speed
- PGN 129026: speed and course over ground
- PGN 128259: speed through the water

How do you get propulsion data from a ship to shore?

The propulsion control serves its values over Modbus TCP on Ethernet or Modbus RTU on RS-485. A RASPBERRY PLC CPU-Gateway FLEX polls them, and with a CAN board it also listens to NMEA 2000 or J1939. Python or Node-RED scales, checks and timestamps the values, stores them, and publishes them over MQTT to a broker on shore through the router of the vessel. The gateway sits on its own network segment and has no write path to the control.



Signals

Signal	Between	Type	Purpose
Pitch, shaft speed, setpoints	Propulsion control → gateway	Modbus TCP, read only	Operating profile of the propeller
Hydraulic pressure and temperature	HPU or thruster control → gateway	Modbus RTU on RS-485	Service planning
Alarms and events	Propulsion control → gateway	Modbus registers, read only	A timeline of events on shore
Engine speed	NMEA 2000 → gateway	PGN 127488, CAN, listen only	Compare with shaft speed and pitch
Speed and course over ground	NMEA 2000 → gateway	PGN 129026	Speed against power
Speed through the water	NMEA 2000 → gateway	PGN 128259	Hull and propeller condition, as a trend
All values, with time and quality	Gateway → broker on shore	MQTT over TLS	Service, support and R&D
Gateway status	Gateway → broker on shore	MQTT	Know that the gateway itself is alive

A Linux PLC with no local I/O: a gateway.



BOUGHT IN SERIES FOR THIS ROLE

RASPBERRY PLC CPU-Gateway FLEX

The Raspberry PLC without local I/O. In this sector it is bought in series, one per installation, to collect data and pass it on; the I/O stays in the propulsion control.

Role in this design	Polls Modbus, listens to CAN, buffers the data and publishes it over MQTT
Compute	Raspberry Pi 4, quad core 64-bit, 1.8 GHz, 2, 4 or 8 GB RAM
Ports	1 x Gigabit and 1 x 10/100 Ethernet, Wi-Fi, Bluetooth LE, 2 x USB 2.0, 2 x USB 3.0, 2 x RS-485
Expansion	One optional board: CAN, LoRa or 4G (LTE-M/NB-IoT or LTE Cat 4)
Software	Debian 64-bit preinstalled; runs Python, Node-RED and C/C++
Power and mounting	12 to 24 Vdc, UPS with supercapacitors, real-time clock, DIN rail TS35, IP20, 0 to 50 °C

This page does not claim a marine type approval from a classification society. Use the gateway for data and monitoring outside the essential control and alarm functions, and agree its place with the yard and the class society.

[See the CPU-Gateway FLEX ›](#)

No risk to the control

Reads only, its own network segment, no commands. The propulsion control and its alarm system stay as they are.

Built for a series

The same Linux image on every unit, with one configuration file per vessel.

Open software

Debian with Python and Node-RED: the maker owns the protocol code and can read every line of it.

SIMILAR PROJECT?

Tell us about your propulsion data. An engineer will answer.

Tell us which controls you need to read and over which protocol, whether there is an NMEA 2000 or J1939 network on board, how the vessel connects to shore, and how many installations a year you deliver. We will tell you how to build the gateway.

READ IT ONLINE

industrialshields.com/case-study-naval-propulsion-onboard-data-gateway-raspberry-plc

EMAIL

info@industrialshields.com

PHONE

[\(+34\) 938 760 191](tel:+34938760191)