NATO CA2X2 FORUM 2026 · ROME · 30 SEPTEMBER 2026

A data-driven geospatial testbed for long-range maritime cyber-physical experimentation

The operational environment is kept as a queryable spatial state in a GIS database. The 3D scene becomes an optional view of it.

Giacomo Longo · Enrico Russo · Alessio Merlo

CENTRO ALTI STUDI DIFESA

OUTLINE

What this talk covers

01 Context and problem: why cyber experiments need environments that look like data, not like scenery.

02 The geospatial-first idea and the requirements it must satisfy.

03 Architecture and implementation: world model, synchronization, sensor pipeline, standards.

04 Case study on the Ligurian coast and the outcomes measured there.

05 Summary and roadmap.

Security experiments on operational vessels

Naval platforms are complex, expensive and operationally critical. Running attack and defence experiments on systems in service carries practical and regulatory risks that make access impossible or inadvisable.

The standard answer is a cyber-range built around a 3D bridge simulator: controlled, repeatable, no operational consequences. This is the most widely adopted solution today.

Ranges of this kind reproduce the network and the onboard systems faithfully. The environment they run in, however, is inherited from the training simulator the range is built on.

The gap between training and testing

A bridge simulator is built to look and behave correctly to a human operator, and for operator training that standard is enough: the trainee's judgement is what is being exercised. Behind the display glass, the data flows can be simplified or proprietary, because nobody on the bridge ever sees them.

Security testing judges the system, and systems do not look at pictures. An intrusion detection system, a radar tracker or an ECDIS consumes protocol data: NMEA sentences, AIS reports, radar video. Testing it requires those protocols, complete and on the wire, and an authoritative state to compare every output against.

What a security testbed needs: accurate and coherent sensor data, systematic scenario variation, controlled simulation time, and a ground truth to verify every output.

Limits of the scene-centric testbed

01 Every experiment couples to one simulator's internal representations. Extending it to new sensors, other mathematical models or non-visual propagation, such as electronic countermeasures, means working inside that engine.

02 The world ends where the modelled scene ends. Cyber tests often need long routes and wide areas of operation.

03 Scenario logic is fixed when the scene is built. Varying traffic, sensors or degraded sensing systematically is difficult.

04 Sensing depends on the rendering loop. Execution is forced to real time, and a GPU stays in the loop even when nobody is looking at the picture.

Where this work sits

Physical and hybrid testbeds. eMIR with LABSKAUS, and the Cyber-SHIP Lab, integrate real bridge hardware to gain fidelity, at the cost of scalability and safety constraints.

Fully virtual testbeds. MaCySTe reproduces the vessel network and onboard systems in software, driving the Bridge Command simulator on a Ligurian Sea scenario. It is the benchmark for this thesis.

Sensor-level threat research. Catalogues of radar attacks and countermeasures exist, and radar intrusion detection has been shown to depend on terrain morphology: jagged coastlines are the harder case.

This work. Environments generated from geospatial data, following the direction the automotive simulation community took with OpenStreetMap, brought to the maritime domain.

The world as a queryable spatial state

SCENE AS WORLD MODEL	→	DATABASE AS WORLD MODEL
The world is the rendered scene		The world is the spatial state in the database
Sensors read the render loop		Sensors query the state
Geography is what was modelled by hand		Terrain and coast come from open data: TINItaly DEM, OpenStreetMap
Visualization is mandatory		Visualization is one client among others

Visualization remains available, but it is no longer required for sensor computation or experiment execution.

What the architecture must guarantee

Data fidelity	Terrain and infrastructure from the national DEM and OpenStreetMap, so sensor output corresponds to charted reality.
Geographic scalability	A new scenario in a new area arrives as a dataset in a standard geospatial format. Importing it extends the world.
Modularity	Sensor suites and threat models assembled per vessel and per scenario, from interchangeable components.
Time control	Runs faster or slower than real time, pausable and steppable, because nothing depends on rendering.
Cross-sensor coherence	One queryable world state behind every sensor view: radar, ECDIS and GNSS describe the same coast at the same moment.
Ground truth	The state itself is the record of what happened, so any sensor output can be verified against it.

One world model, interchangeable services

vessel dynamics · traffic · sensing services · network interfaces · protocol adapters · logging · visualization (optional)

SYNCHRONIZATION LAYER · EVENT-DRIVEN CLOCK · PUBLISH/SUBSCRIBE

DATA · INVARIANT LAYERS

Terrain elevation and coastal infrastructure, imported from open geospatial data into the GIS database. The world is what the tables say.

STATE · ENTITY REGISTRY

Every tracked object is ⟨ID, properties⟩ with a time-ordered identifier. Positions, headings and speeds update as the run proceeds, and the current world can be queried at any moment.

The prototype runs beside MaCySTe

MACYSTE · REFERENCE TESTBED

3D-driven maritime cyber range
Ownship dynamics
NMEA position and heading
AIS traffic of surrounding vessels

→ **GEOSPATIAL LAYER**

PostgreSQL with PostGIS holds the world
NATS carries clock and data
Open3D raytraces terrain and structures
Python, asyncio, Podman

→ **OUTPUTS**

ASTERIX CAT 240 video, UDP multicast
NMEA RARSD sentences
Live radar overlay in OpenCPN

The representative service is a primary radar computed from the database: it queries the spatial state, includes coastal structures and vessels inside the radar envelope, and emits CAT 240 video with no render loop in the path.

What the world model holds

Every object a scenario cares about has a record: the ownship, every vessel in the traffic, the buoys and the landmarks. Each record identifies the object and carries whatever properties the scenario gave it.

The coast itself is data. Terrain elevation comes from the national cartography, buildings, piers and breakwaters from the open world map, and importing a dataset brings a new stretch of coast into the testbed.

What moves lives beside it. Positions, headings and speeds update as the run proceeds, and the whole state can be queried at any moment, to see what is true in the simulation or to verify an output against it.

Clock and messaging

One clock drives the whole run and tells every component what time it is in the simulation. Because nothing depends on rendering, a run can pause, step or change pace without breaking a sensor, and long runs stay accurate.

Components talk through a message bus. Each subscribes to what it needs and publishes what it produces, and the bus keeps the history, so any run can be replayed afterwards.

A sensor can join or leave a running experiment without the rest noticing, which is what makes the sensor suite configurable per scenario.

A sensor layer beside existing simulators

The radar service is driven by what a bridge already produces: NMEA sentences for position and heading, AIS reports for traffic. Any simulator that emits NMEA can drive it, so the testbed can run beside an existing simulator and simulate just the sensors, without replacing anything.

From the antenna position, rays are cast through the terrain and the structures: 4096 azimuths, about 500 elevation angles each, dense near the horizon. For every direction this yields the ranges at which coast, cliffs and buildings return an echo, and it is recomputed each 500 m of movement.

On every clock tick, the service emits the picture the radar would be painting: echo intensities along the range axis, packed into the 512-byte scanlines of the CAT 240 format.

Protocol-native outputs

The testbed does not invent its own interfaces. Position and heading go out as NMEA 0183 sentences, traffic as AIS messages per ITU-R M.1371, and the radar picture as ASTERIX CAT 240 video over UDP multicast.

The protocols are implemented in full. Long AIS payloads are fragmented and reassembled exactly as real transceivers do, and the video follows the CAT 240 packet layout used by radar distribution infrastructures.

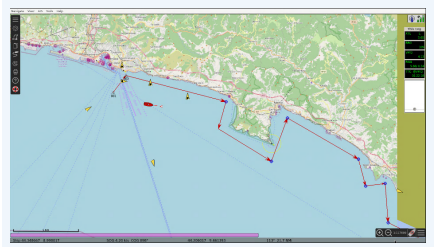
Existing ECDIS and radar tooling consumes the outputs directly: OpenCPN overlays the radar picture on the chart in real time. A testbed that speaks the protocols of the exercise infrastructure joins it without bespoke adapters.

Ligurian coast: Genoa to La Spezia

About 60 nautical miles between two major ports. La Spezia also hosts the Navy's main battle fleet and one of its arsenals. Portofino and the Cinque

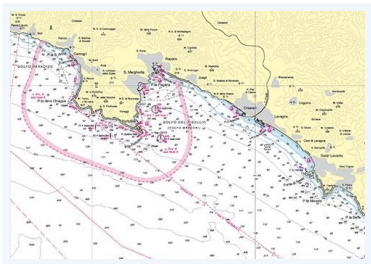
Terre are the conspicuous points along the route, the landmarks a radar operator would use to fix position. The scenario assumes reduced visibility.

With optical bearings unreliable, radar, ECDIS and GNSS are what remains to hold a fix.

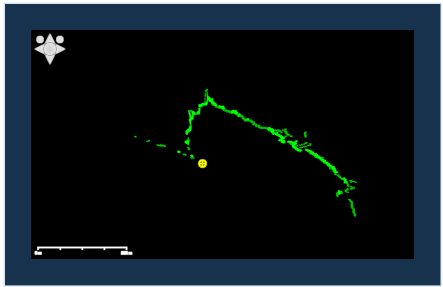


TEST ROUTE, ECDIS VIEW (OPENCPN)

Radar video against the chart



NAUTICAL CHART, GULF OF TIGULLIO



RADAR VIDEO, COMPUTED FROM THE DATABASE

Portofino is recognisable in the radar picture, in the shape the chart leads a navigator to expect. Because the world model is the database, sensor output can be checked against it directly: every echo traces back to a row in the world state. A different area, traffic or sensor setup is a different dataset, so scenarios can be varied and repeated under controlled conditions.

Scalability and deployability

Standard servers

Sensor computation never touches the render loop and holds no GPU: where one is present, it stays free for AI workloads such as intrusion detection models and synthetic traffic.

Controlled time

Runs go faster or slower than real time, pause and step, as the experiment requires.

Parallel execution

Several experiments run side by side on the same infrastructure.

Replaceable sensors

Sensors can be enabled, disabled, swapped or run at different fidelity, and new protocol outputs can be added without redesigning the testbed.

The terrain and infrastructure datasets already cover all of Italy. Extending to a new area of operations is a data import.

Summary

01 The world model is a spatial database, and sensors compute from it directly, so their output can be checked against the world that produced it.

02 Scenarios are datasets and configuration: areas, traffic, sensor setups and degraded sensing vary without rebuilding scenes, and runs repeat under controlled conditions.

03 Nothing in the pipeline requires rendering. The testbed runs on plain servers, in parallel, on controlled time, and new services attach as additional components.

04 Outputs are protocol-native, so the testbed can join existing CAX and exercise infrastructures without bespoke adapters.

Next steps

01 Quantitative comparison against MaCySTe as the baseline, and information-content metrics on the sensor output.

02 ECDIS and GNSS models on the same world state, for consistency checks across the whole navigation suite.

03 Radar attacks and electronic countermeasures on the sensor path, for intrusion detection studies over realistic coastlines.

04 Attaching the sensor layer to other simulators that already emit NMEA and AIS, beyond the MaCySTe reference case.

05 Mapping the world model onto NATO federation standards, HLA with NETN entity modules, to join distributed exercises as a federate.

06 Terrain and structures published as OGC 3D Tiles, the open standard for streaming geospatial scenes, for any visualization client.

Thank you. Questions?

A data-driven geospatial testbed for long-range maritime cyber-physical experimentation

NATO CA2X2 FORUM 2026 · ROME · 30 SEPTEMBER 2026

Giacomo Longo · giacomo.longo@unicasd.it · Centro Alti Studi Difesa