UnetStack

Unlocking Underwater Connectivity

Exploring the UnetStack Framework for Subsea Communications and Networking

01 Introduction

The Internet and wireless technologies have reshaped our lives, and the marine and subsea domains are no exception. From autonomous ships and smart ports to telemetry buoys, marine robots, and sensors, a wave of advancement has emerged, enabled by terrestrial wireless networks. Underwater, this does not translate: many robots still rely on tethers, and subsea sensors store data for manual retrieval.

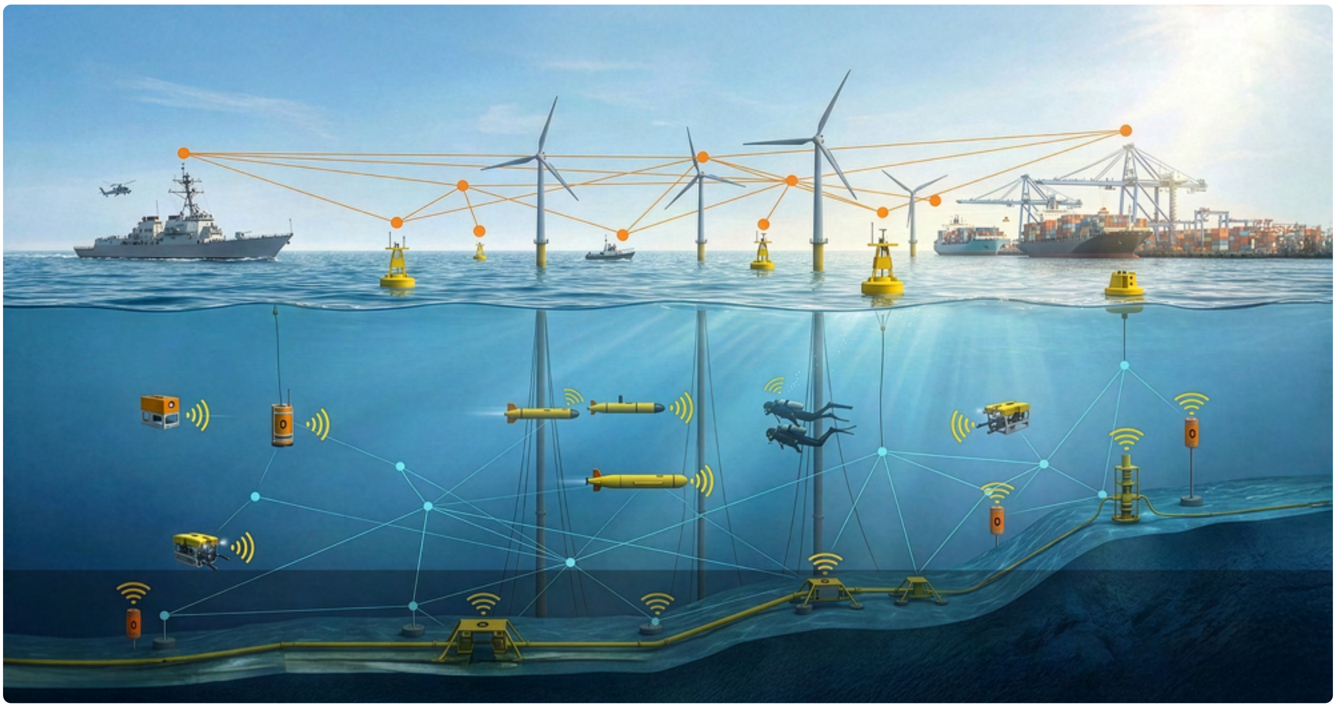

Figure 1. The future of underwater wireless

Terrestrial wireless does not work underwater: electromagnetic waves fade rapidly, except optically over short ranges. Acoustic links bring limited bandwidth, long latencies, and challenging channel conditions, rendering Internet technologies such as TCP/IP impractical over them. Underwater networks call for a whole new way of thinking. This motivated UnetStack, a software-defined framework for scalable, flexible underwater wireless networks that lets us extend the Internet underwater.

> UnetStack is a versatile software framework designed for underwater communication and networking. Offering a range of tools and technologies, it enables the development, configuration, and optimization of communication devices and networks beneath the waves.

02 What Is UnetStack?

UnetStack is a collection of technologies designed to extend communication networks underwater. Developed as part of the Unet project at the Acoustic Research Laboratory (ARL)[1], National University of Singapore, this agent-based network technology enables communication networks that span above and below water. Acoustic or optical links are served by dedicated modems, while RF, cellular, satellite, or wired connections use standard TCP/IP interfaces over UnetStack. Not every node needs to run UnetStack: nodes come under its umbrella, managed by it, with the network utilizing the best links available for seamless end-to-end connectivity.

This fully software-defined framework spans the physical layer to the application layer, along with positioning and localization, data analytics, and device and sensor management.

Figure 2. UnetStack capability wheel

1 Acoustic Research Laboratory, National University of Singapore, arl.nus.edu.sg

03 Who Is UnetStack For?

Different users need different things from underwater communication, and a rigid black-box modem serves none of them fully. Four audiences illustrate the range.

3.1 Acoustic Communication Research

Researchers developing communication techniques and network protocols need what no closed modem offers: access below the deck, transmitting arbitrary waveforms, running their own code inside the modem, and replacing pieces of the stack while keeping the rest.

UnetStack was created in a research laboratory and is built for exactly this way of working. Its open, modular architecture lets researchers work at whichever layer interests them: an interactive shell with Groovy scripting gives live access to every agent and parameter, extensible web dashboards handle configuration and monitoring. The Unet socket and gateway APIs, in Java, Groovy, Python, Julia, C, and JavaScript, support everything from socket-like data transfer to deep integration. Finally, since no technique should reach a sea trial untested, high-fidelity simulation is as essential to research as the hardware itself.

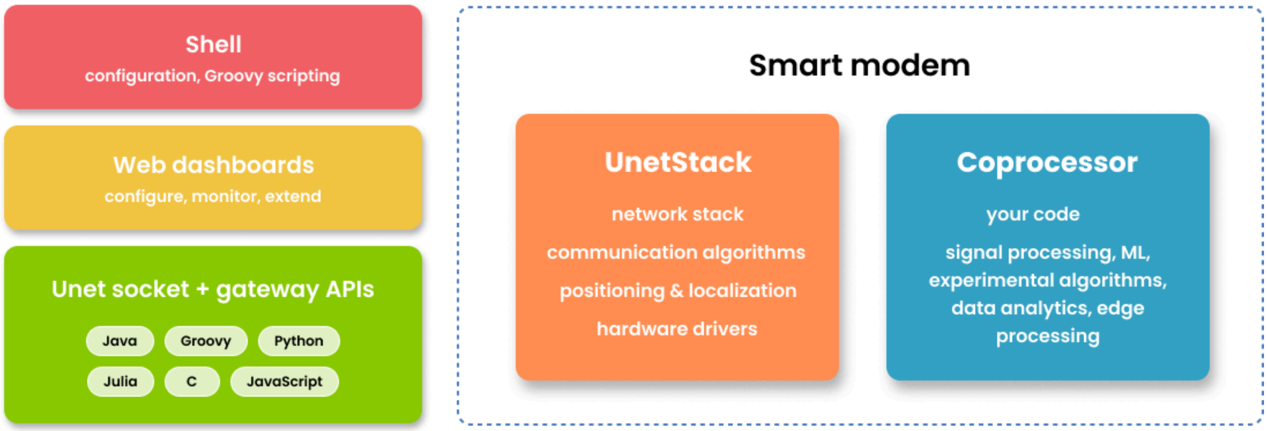

Figure 3. Research access to the stack. Every layer is reachable: run your code on the coprocessor, script it from the shell, or integrate through the APIs.

3.2 Underwater Networks

Many deployments need more than a point-to-point link: a network of many nodes with routing, medium access control, reliable transport, and remote management, spanning acoustic, optical, and above-surface links. UnetStack is a full network stack, so multi-node networking comes standard, as illustrated in Figure 1. The same software scales from a two-node link to a harbor-wide network, and deployed nodes can be reconfigured remotely, even over the acoustic link itself. And since repositioning misplaced nodes at sea is costly, topology and protocols are best validated in simulation first.

3.3 Marine Robotics

Autonomous underwater vehicles place unique demands on communication: links change constantly as vehicles move, several vehicles may share the channel, and every transmitted bit competes with mission endurance. Subnero's software package for marine robotics answers with dynamic TDMA for channel sharing, adaptive routing, and content-aware compression, turning a modem into a mission communication system. With vehicles and vessel time at stake, rehearsing multi-vehicle communication in physics accurate simulation before a mission is essential.

Figure 4. Marine robot communications

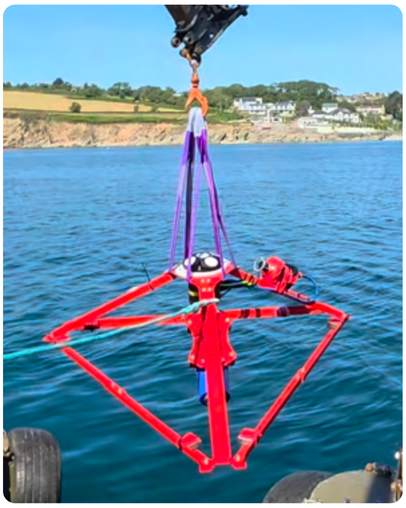

Figure 5. SWIS - ADCP Edition deployment

3.4 Subsea Sensing

Subsea sensors have historically stored data for manual retrieval, with months between measurement and insight. Inverting this takes more than a data pipe: data must be processed at the edge, summarized intelligently, and delivered to shore over a bandwidth-limited acoustic link, with the instruments manageable remotely. The Subnero Wireless Integrated Suite (SWIS)[2] is built on exactly these UnetStack capabilities: the same stack that moves data can also understand it.

04 The Gap, and How UnetStack Closes It

A typical acoustic modem is a sealed point-to-point data pipe, and that leaves every audience in Section 3 short. Researchers cannot get inside it: the physical layer is fixed, there is no way to run their own code, and every experiment stops at the vendor's interface. Network builders get a link where they need to build routing, transport, and management; everything above the modem becomes custom engineering. Robotics and sensing teams get raw bytes where they need application intelligence: channel sharing, prioritization, edge processing. And everyone shares the same two gaps: no faithful way to test a system before committing it to the water, and no path to extend or reuse the hardware they already own.

UnetStack closes these gaps with two families of offerings: a simulation ecosystem to validate everything before it hits the water (Section 5), and UnetStack-enabled modems that are open, flexible, modular, and customizable, from the bare digital core to complete smart modems (Section 6).

[2] subnero.com/products/swis/

05 Simulate First: The UnetStack Simulation Ecosystem

Underwater trials are expensive: vessel time, logistics, and weather windows make every hour in the water precious. The UnetStack simulation ecosystem is built for fidelity: simulations close enough to the real thing that the performance you see on the desk closely predicts the performance you will get at sea. The ecosystem is a ladder, each rung closer to reality, and you climb as far as your project needs.

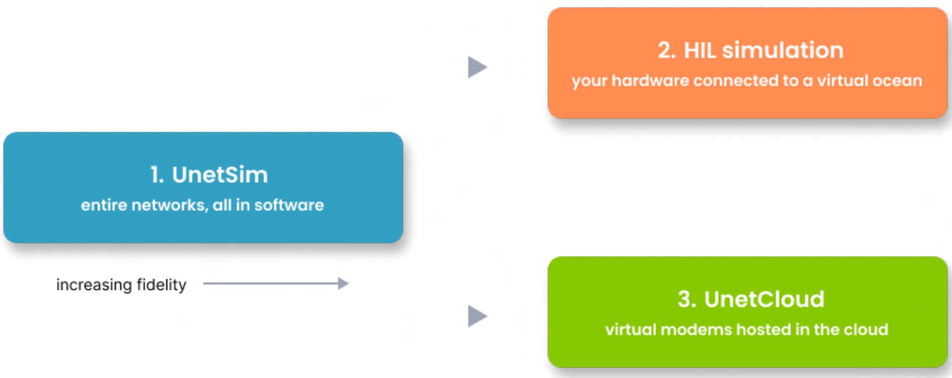

Figure 6. The simulation fidelity ladder

5.1 UnetSim

UnetSim simulates networks of many nodes, with real-time and discrete event modes for interactive testing, Monte Carlo evaluations, and protocol assessment, using an abstracted PHY suited to network and protocol work. The same agents that run in simulation run on real hardware, so protocols carry over to deployment without rewriting. UnetSim is free for research and education, with commercial use available under license (Section 8).

5.2 Hardware-in-the-Loop Simulation and the Virtual Acoustic Ocean

Where UnetSim abstracts the physical layer, HIL simulation exercises it, bringing the real system into the loop. It has two components: a simulated ocean, and the system dropped into it.

The ocean is the Virtual Acoustic Ocean (VAO)[3], a fully open source simulator that presents real acoustic signals to connected systems, so the actual modem software runs against the virtual ocean exactly as it would against the real one. Its channels can be physics-based, modeling acoustic propagation, or replay channels built from at-sea recordings, which take fidelity beyond what physics models alone can offer. Its interface protocol, UASP, is open source too[4], so any system that speaks it can join the loop.

The system in the loop can be a complete modem connected in place of its transducer, just the digital side of one such as the UnetCube, or entire nodes running in the cloud (Section 5.3). What you validate on the bench is exactly what enters the water: integration testing, protocol validation, acceptance testing, and operator training can all happen before a single vessel is booked.

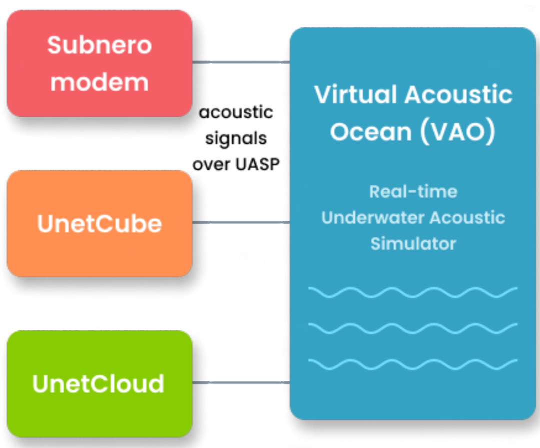

Figure 7. One virtual ocean, every level of fidelity: real modems and UnetCubes in the loop, or entire nodes simulated in the cloud.

3 VirtualAcousticOcean.jl, real-time underwater acoustic simulator, github.com/org-arl/VirtualAcousticOcean.jl
4 UASP (Unet Acoustic Streaming Protocol), github.com/org-arl/VirtualAcousticOcean.jl/blob/main/docs/uasp2-protocol.md

5.3 UnetCloud

UnetCloud brings real modem software to the cloud: full modem-grade UnetStack nodes hosted by Subnero, for developing and testing against the real stack, including within VAO, without owning hardware. Beyond starting integration before hardware arrives, running large virtual deployments, or sharing a test environment across sites, it unlocks something no sea trial can: simulating expected performance in waters that are hard to reach, in a site you have never operated in but are bidding your first job on, or in restricted waters, common in defense, where access may never be possible at all.

06 The Hardware: Smart Modems and the Digital Modem

Simulation carries a project a long way, but every deployment ends in hardware.

6.1 Smart Modems

When the goal is optimal end-to-end connectivity, Subnero's WNC series of underwater acoustic smart modems is the answer. They run the full UnetStack stack, deliver class-leading throughput and range, and come in three configurable frequency bands, LF, MF, and HF. Every capability described in Section 3 is available on, and field-proven on, these modems.

Figure 8. Subnero acoustic smart modems

6.2 UnetCube: The Digital Modem

UnetCube is a digital modem: exactly a Subnero smart modem without the analog parts. The digital hardware, UnetStack software, signal processing chain, and network stack are identical; the data acquisition modules, amplifiers, analog electronics, and transducer are absent, supplied by the user instead.

It serves two purposes: connected to VAO, it is the hardware for HIL simulation similar to actual modems (Section 5); paired with your own analog electronics and transducer, it is the core of your own modem.

Figure 9. Illustration of UnetCube in action

6.3 Building Your Own Modem Around UnetCube

Researchers with bespoke transmission systems, OEM vendors adding communication capability without extensive R&D, and operators of third party modems can all use UnetCube as the digital heart of their solution. Two applications cover most of them.

Building your own modem. Supply the data acquisition and analog side and UnetCube does the rest. Your hardware, ADC and DAC, amplification, and a transducer, connects to UnetCube over Ethernet using UASP; all your side needs is a small piece of translation software converting UASP to whatever your electronics accept. UnetCube performs all signal processing with the complete network stack above it, you keep full control of the hardware design, and since UASP is open source, the integration can be done entirely on your own, no support required.

Enabling network features on your existing modem. A typical modem gives you a point-to-point link and stops there. Pair it with a UnetCube, with a modem software driver, and everything UnetStack offers arrives on top of the link you already trust: routing, reliable transport, remote management, and the application capabilities of Section 3, much like adding TCP/IP on top of an Ethernet card. If your modem supports arbitrary waveform transmission and reception, UnetCube can even take over the signal processing itself; otherwise it sits above your modem's own physical layer. Either way, this is also the path to multi-modal networks combining acoustic, optical, and other links under one stack.

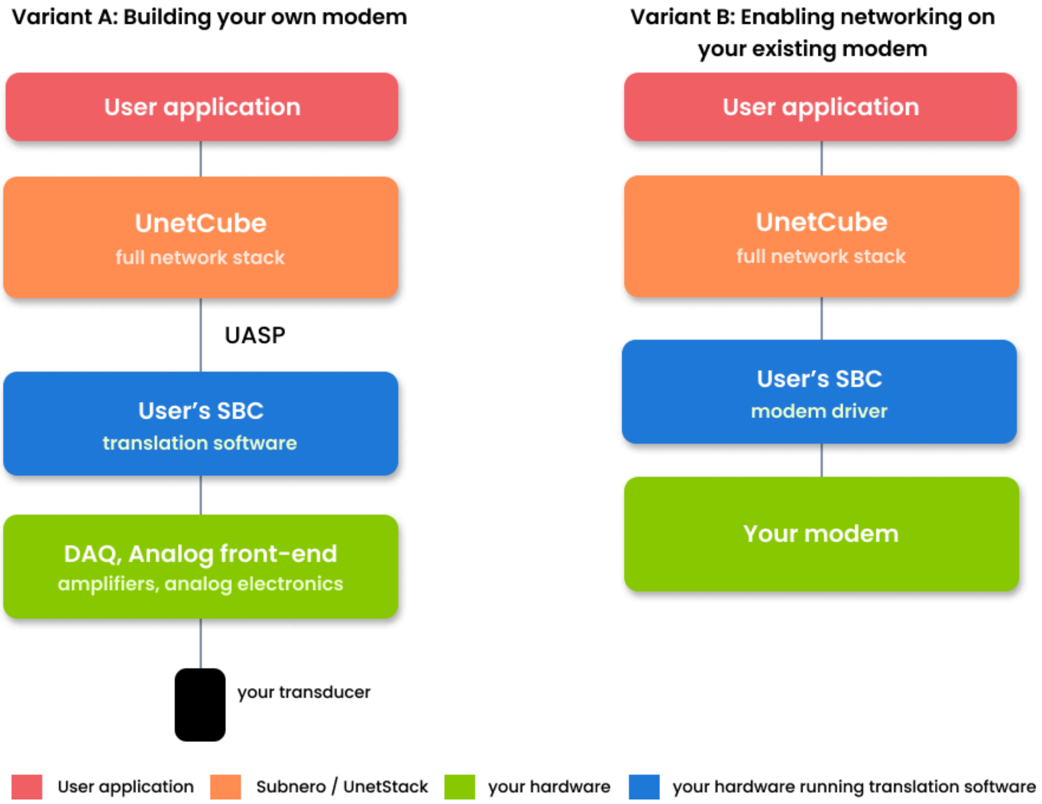

Figure 10. Two applications of UnetCube: supply the analog side and build a modem, or add a networking layer to a modem you already own.

> Note: Features such as coherent modulation schemes, two-way ranging, and positioning depend on the capabilities of the connected hardware.

07 Under the Hood: The UnetStack Architecture

Everything described so far rests on a deliberately simple architecture.

Messaging fabric (fjåge) + Unet framework. The lightweight, open source fjåge forms the agent-oriented foundation of UnetStack, using JSON over networked transports such as TCP. The Unet framework provides the core services, messages, agents, and APIs needed by UnetStack.

Agents. Everything above the foundation is an agent: the physical layer, the network stack, localization, analytics. Basic agents fulfill the essential requirements of typical underwater networks. Licensed agents provide the application-specific capabilities behind offerings such as the marine robotics package and SWIS (Section 8).

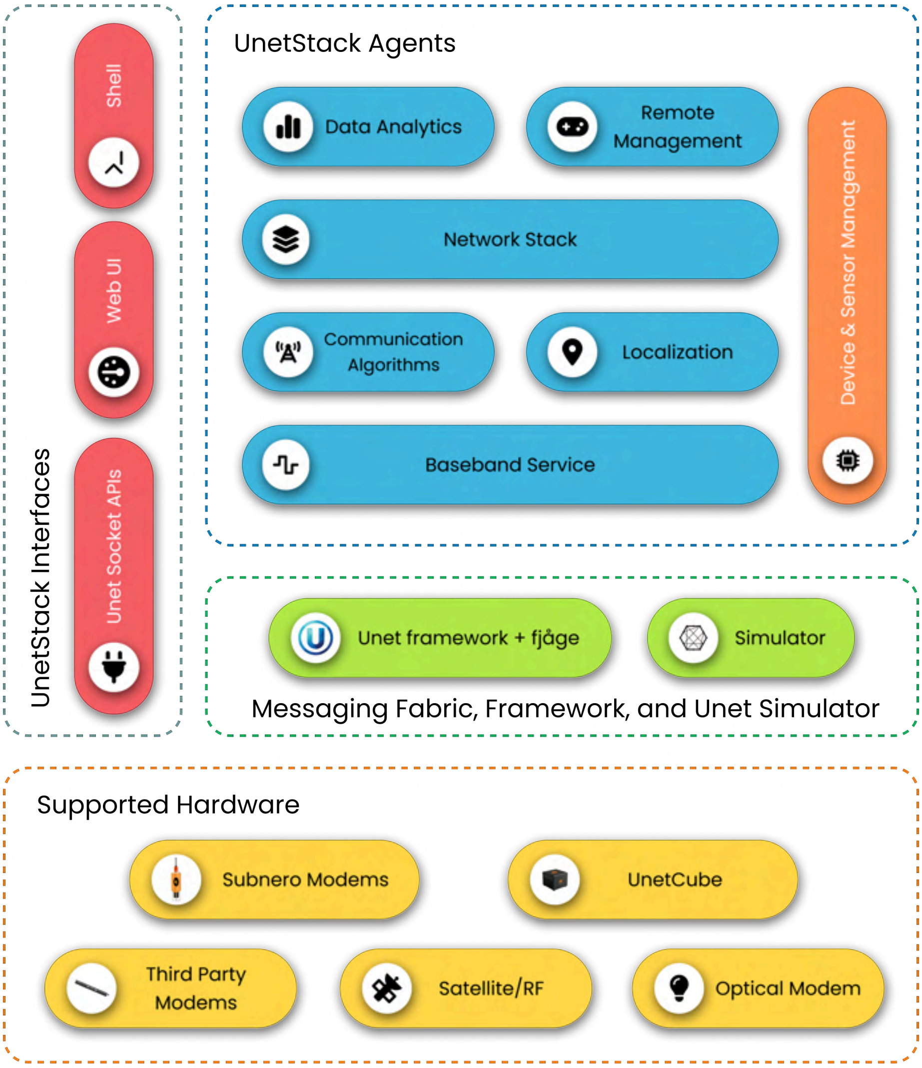

Figure 11. UnetStack component stack

08 UnetStack Editions

Behind every offering in this document sits an edition of UnetStack. UnetSim ships as the Community or Commercial edition, while Subnero smart modems, UnetCube, and UnetCloud all run the OEM edition. You never choose an edition directly, it comes with the offering, but what each contains tells you what your offering can do.

	Community	Commercial	OEM
Comes with	UnetSim, free for non-commercial use	UnetSim, commercial license	Subnero modems, UnetCube, UnetCloud
Basic agents	Yes	Yes	Yes
Physical layer	Abstracted	Abstracted	**Real signal processing**
Channel	Model-based	Model-based	**Real ocean or VAO**
Baseband service, device agent	Simulation-grade	Simulation-grade	**Full**
Device specific agents	No	No	**Yes**
Licensed agents	No	No	**Available**
Use	Non-commercial	Commercial	Per product/instance

Table 1. UnetStack editions

Community and Commercial are the simulation editions: UnetSim and the basic agents with an abstracted physical layer. Community is free from the UnetStack website for non-commercial use; Commercial is functionally the same, licensed for commercial use.

The OEM edition runs on real hardware and completes the stack for it: the full baseband service and device related agents replace their simulation-grade counterparts. It ships inside Subnero smart modems, UnetCube, and UnetCloud, and is what lets modems and UnetCube take part in HIL simulation. Licensed agents are available only with the OEM edition, since the applications they serve run on real hardware.

09 Conclusion

UnetStack meets each of its audiences where they are: researchers get access at every layer, network operators get a stack that scales from a single link to a multi-node deployment, robotics teams get mission-grade communication, and sensing programs get edge intelligence with SWIS.

Whatever the starting point, the path is the same: simulate with UnetSim, test real hardware-in-the-loop against the Virtual Acoustic Ocean, scale with UnetCloud, then deploy, on Subnero smart modems or a modem of your own built around UnetCube. From first simulation to final deployment, UnetStack is the natural foundation for extending networks underwater.

RESOURCES

UnetStack unetstack.net, project home, documentation, and Community edition download

UnetStack Handbook the definitive technical reference, read the handbook

Virtual Acoustic Ocean open-source simulator on GitHub

Subnero modems and solutions subnero.com/products

UnetCube in the UnetStack ecosystem, learn more

UnetCloud & HIL simulation learn more

Technical Resources subnero.com/technical-resources

Contact info@subnero.com

subnero.com
info@subnero.com
unetstack.net