Design of a Wireless, Inexpensive Ocean of Things System for Volunteer Bathymetry

Brian Calder *Member, IEEE*

***Abstract*—Although much progress has been made in recent years to fully map the world ocean, only approximately 20% is adequately mapped to modern standards. Filling in the remainder must by necessity be a multi-modal effort, with traditional ocean mapping technologies such as crewed survey ships with multibeam echosounders being mixed with newer systems such as uncrewed, sail-powered mapping systems. Volunteer data from any ship with an echosounder can also be used, but although there have been commercial efforts in this field, most of these systems do not contribute data into the public arena and public entities have largely avoided this field due to complexities of costs, data processing, and uncertainty on how to handle the effort. This paper describes the design of an end-to-end system for managed volunteer bathymetric collection consisting of an inexpensive (∼$20) wireless "ocean of things" data logger for NMEA0183 and NMEA2000 data, associated firmware to manage the collection, a mobile device application to off-load, aggregate, and transfer the data into the cloud, and a cloud segment to process the data and submit it to an international data repository. All of the design has been released under an Open Hardware or Open Source license, allowing independent organizations to initiate data collection efforts without having to do any of the design work themselves. The goal is to provide a simple, approachable implementation, encouraging greater adoption of these ideas in hard-to-reach areas of the world with minimal effort on the part of the host organization.**

***Index Terms*—Internet of Things, Ocean of Things, Volunteer Bathymetric Information, Crowdsourced Bathymetry, Cloud-enabled Data Processing**

I. Introduction

THE bathymetry of the world ocean is a vital base layer in any scientific or industrial effort in the seas, but despite centuries of effort, estimates of the area mapped "adequately" (by some appropriate definition) still hover around 20% [1]. Partly this is due to modern expectations of data quality, uncertainty, and completeness, but it is also a function of the physical difficulty of mapping the oceans. The reference modern mapping system is the multibeam echosounder [2], which can map hundreds of simultaneous depth measurements in a wide swath perpendicular to the longitudinal axis of a ship at up to 40 measurement cycles per second, typically covering a swath of approximately four times the water depth. Even with the most advanced systems, however, coverage rate is limited by the speed of the ship through the water, the water depth, and environmental conditions. In addition, the oceans are large, and ships are (relatively) small, and expensive: day rates for professional ocean mapping assets vary from $20,000/day for small boats to $250,000/day for large, specialist ships such as ice-breakers. Consequently, to effectively map the entire ocean to modern standards by 2030 (the goal of the Seabed 2030 project [3]), a mix of traditional and innovative technologies will be required.

While the reference standard for mapping is the survey-grade echosounder, there is growing evidence [4] that data collected by volunteers using recreational-grade fish-finders or navigational echosounders can be used to provide auxiliary data in many areas. This data will likely never reach the level of uncertainty, completeness, and quality achieved by professionally acquired data, but with appropriate assessment of reliability [5] it can likely be used to fill in between authoritative data, or indicate where there appears to be a significant deviation from the known authoritative data, thereby triggering a new survey effort.

Collection of volunteer bathymetry (also, somewhat inaccurately [6], called "crowdsourced bathymetry") can therefore be a useful adjunct to more controlled survey effort, and has been proposed as an additional tool to fill data gaps. This idea is not new. A number of commercial projects currently collect data from individual users and aggregate the data to make bathymetric products, which are then typically offered back to the users, usually gratis (see Section II for more details). These projects do not usually, however, make this data available for scientific use; this can be due to financial concerns, liability issues, or even data protection rights. While there have been efforts to open these databases (e.g., as lower resolution gridded products for integration with other compilations), success has been limited and the fragmented nature of the projects means that the best effect of the data is not achieved.

What is required is a scalable, cost-effective, simple, unified data collection platform. Starting a new unified effort to collect data from scratch would be no small feat, however. The effort involved in setting up a data collection event in a local area (e.g., a harbor or local bay system) can be significant and technically complex, and would likely entail cost and technical demands beyond those which can be supported by an interested local party (e.g., a yacht club or coastal management group). There is also no guarantee that the data from such an effort would make it to a national or international database, which can often be a technically challenging process.

This paper aims to resolve these difficulties by providing a template for such a directed data collection event. It outlines a concept of operations for this effort, and provides detailed plans for the hardware, firmware, mobile application, and cloud segment software required. These supports recording,

B. Calder is with the Center for Coastal and Ocean Mapping and NOAA/UNH Joint Hydrographic Center, University of New Hampshire, Durham NH 03824, USA. E-mail: brc@ccom.unh.edu.

aggregation, upload, processing, and submission of data to an international database in a standard, accepted, format. The project therefore provides all of the resources required to implement a volunteer bathymetry data collection event. This would allow a local organization with limited resources to essay a volunteer bathymetry collection event using known and standardized methods, while also allowing for variations to suit local needs. Crucially, the standardized methods would ensure that the data is treated consistently, and would always be contributed to the same database in the same way, making global aggregation possible.

By providing these resources, the hope is that more local, smaller organizations (e.g., local port authorities, yacht clubs, smaller municipalities, or even smaller/developing nations) who might otherwise be put off by the complexity of the endeavor would be encouraged to attempt to collect and contribute this type of data, since most of the hurdles have already been considered and resolved. Herein, this is termed the "no excuses" principle: if the project provides all of the appropriate resources, the question should become "why not?" rather than "can we?"

The remainder of the paper outlines the core concept of operations underlying the proposed end-to-end system for data collection, manipulation, and submission, then describes the hardware data logger, its embedded firmware, the mobile device application used to collect the data, and the cloud-based processing and transport segment. Finally, a demonstration project is described with data from an eastbound transit of the USCGC Healy through the Northwest Passage.

II. Background

The concept of volunteer bathymetric information (VBI) has a long history. Many hydrographic offices, for example, have a mechanism to accept observations from mariners, often of obstructions or unexpected shoals. The number of potential observers, and the consequent volume of data, has also made this an interesting target for commercial interests, and more recently for projects like Seabed 2030. Consequently, there have been a variety of projects aiming to tap into the potential of volunteer observations. These can be broken into four main groups: closed commercial, commercial plus, open commercial, and open voluntary/academic.

Closed commercial VBI projects have been the most successful of the kind with respect to number of volunteers, data collected, and products made. Olex[1], for example, has collected significant amounts of data from fishing vessels, primarily in the north Atlantic and Arctic, using a "closed garden" approach: all observers collect data using software provided by Olex, and receive in return an aggregated dataset from all other observers. Olex acts as database, processor, and distribution system. The source database is proprietary, but analysis of aggregated grids released for scientific use [7] show good agreement with data from authoritative sources.

Similar projects sponsored by recreational equipment manufacturers such as Garmin[2], Navico[3], and Johnson Outdoors[4] have similar properties, and generally also follow the "closed garden" model. This is typically implemented by logging fishfinder or navigation echosounder data in a multifunction display chartplotter provided by the manufacturer, with data exchange happening when charts are updated. The systems typically provide a gridded overlay product based on contributed data, which can be displayed as an alternative to official charts, but only by authorized users. Release of data, even in aggregated form, is often limited by policy, economic, or commercial concerns, or by data rights legislation. Consequently, data from these projects are rarely, if ever, submitted to national or international databases.

Commercial Plus projects attempt to provide what might be considered a "freemium" model, where access to collected data is free for contributing volunteers, but the sponsoring company attempts to make a commercial offering from the data. This is typically done by aggregating the data into a product that can subsequently be licensed or sold. An early example of this was the ARGUS project [8], which provided users marine WiFi as an incentive to host the data collection hardware, and intended to aggregate data for eventual product creation. The TeamSurv project [9] had similar ambitions, and was able to offer data on a commercial basis for some areas in the English Channel for a time (the hardware component is still commercially available, but data services have ceased operations). Both projects, however, had difficulty scaling since they attempted to release loggers to the general community rather than focusing on a dense collection in a local area, and therefore were not able to provide sufficient amounts of data to feed the construction of a commercially viable data service. The Orange Force Marine[5] Mussel is a further, current, example of this type with a hardware component that attempts to telemeter data to shore in real time to assist in oceanographic prediction, and a cloud-based processing scheme with a data pipeline to the Great Lakes Observing System[6]. It remains to be seen how this system scales in practice.

The Open Commercial category consists of projects where a commercial, or occasionally non-profit, entity provides software and/or hardware to volunteer collectors, contributes the data to a national or international data, but (mostly, but not universally) has the intent of using the data for further products or services. Projects from SeaID [4], CIDCO [10], [11], and FarSounder[7] fall in this category, along with other providers of generic data loggers (e.g., YachtDevices[8]) who provide loggers, but limited services, leaving it to the user to find

[1] https://olex.no/index_en.html

[2] https://www.navionics.com/usa/charts/features/sonarchart, https://activecaptain.garmin.com

[3] https://www.c-map.com/social-map/, although note that Navico-CMAP have also started providing data to DCDB and promoting this concept to their users.

[4] https://humminbird.johnsonoutdoors.com/mapping/autochart-pc-north-america

[5] https://www.orangeforcemarine.com

[6] https://glos.org

[7] https://www.farsounder.com/blog/expedition-sourced-data-collection-program-progress-update

[8] https://yachtdevicesus.com/collections/nmea-converters-adapters/products/voyage-recorder-ydvr-0

out how to contribute the data. None of these projects has yet found a business model, however, and their practical reach has generally been quite limited. A notable exception is the collaboration project between Rosepoint Navigation Systems[9] and NOAA [12], [13] to allow users of the Coastal Explorer software to opt-in to contributing their data to the International Hydrographic Organisation's (IHO) Data Center for Digital Bathymetry (DCDB), which has contributed the majority of new data holding for volunteer data there. Atypically, the business model here appears to be not to have one: the data is provided freely if the user allows it. Since Coastal Explorer is a software-only product, however, a computer, laptop, or tablet is required, limiting potential volunteers to larger vessels.

The final category of projects are academic or open source efforts to mobilise volunteers. The Open Sea Map project[10] provides a small data logger and a mechanism to upload data; the DYNAMO system [14] provides both hardware and an extensive software environment to aggregate and process volunteer data (of many kinds besides bathymetry). Both systems attempt to collect data globally, and therefore have limited data density available in most areas. A project which perhaps provides a better model is the Great Barrier Reef mapping effort [15], where dive boats have been recruited to collect data in a relatively limited geographical area, and close interaction from the project managers—including rapid feedback of collected data products—has encouraged volunteer recruitment and retention.

These examples make it clear that while there is a desire for a strong volunteer data collection effort, and volunteers to be had, none of the current (and past) projects have been able to scale to global reach and access all market segments of potential volunteers (although it is questionable whether all segments are equally useful [6]). These observed problems in previous projects have guided and motivated the current work: a scalable platform that can be replicated many times, avoiding a bottleneck of a single organization; a low-cost hardware/software solution pre-configured to ensure that data reaches international databases in an agreed format with options for metadata; an open source approach allowing for adoption by multiple organizations and enhancement from the community including, potentially, commercial support; and focus on locally-supported dense data collection by the local inhabitants—who have the motivation—of a spatially-limited area. The expectation is that by resolving many of the issues that have limited or hindered current projects, more, and better, data may be collected in the future.

III. Concept of Operations

Instead of a general volunteer collection effort (i.e., trying to reach as many people as possible without consideration of locality), the proposal here is for a set of localized data collections, each one supported by an organization with an interest in the locale. The concept of operations (CONOP) here is to minimize the effort required of the volunteer data collectors and thereby encourage as much recruitment and retention of volunteers as possible.

In this model of operations, the Wireless Inexpensive Bathymetry Logger (WIBL) hardware units would be handed out by the support organization, which would be expected to commit one or more persons to the effort. The support person(s) would be expected to help with installation of the loggers, and would return to the ships on a regular basis to check that the loggers are still running, show the volunteers the aggregated data that they are collecting, and to offload new data from the loggers, ameliorating concerns about modifying the navigation suite in attaching the logger. This process therefore requires only that the volunteers provide a space for the (very small) logger and potentially, depending on data interface, 12V power, making participation almost effortless. (More interested volunteers could, of course, be encouraged to interact more with the support organization—for example by providing more extensive metadata—hopefully making their data more valuable, and allowing them to access their own data.)

The outline system, Figure 1, expects that a custom mobile device application will be used by the support person to interact with the logger. A general interface there allows the system to be interrogated and configured, for data to be transferred to the mobile device for subsequent upload, and for the firmware of the logger to be updated over the air, allowing for field maintenance and software updates.

Once aggregated on the mobile device, data from multiple volunteers is transferred to the cloud once better connectivity is available (it is possible that the volunteer ships may be remote or in under-developed regions without cell service close to the dock). While many potential implementations are possible, in the initial development the WIBL system uses the mobile device to upload the data into an Amazon Web Services (AWS) Simple Storage Service (S3) bucket, which in turn triggers a processing AWS Lambda (Python) script to unpack the data, recompute timestamps, and reformat the data as required for upload into the DCDB, hosted by the U.S. National Oceanic and Atmospheric Administration (NOAA) National Centers for Environmental Information (NCEI) in Boulder, Colorado. This intermediate data is stored to another S3 bucket so that it can be further manipulated, but the initial write triggers a second AWS Lambda to transfer the data directly to the DCDB using their Application Programming Interface (API) for data ingestion[11]. Metadata is added to the core data during processing (if available) in order to provide more information on the data collector.

The system is, by design, a template for how a particular implementation of a WIBL data collection would be structured: each local organization would start with the basic design and then adjust as required for their particular circumstances, resources, etc. The data flow path is therefore designed to be modular, with the S3 buckets acting as check-points where the particular implementation could insert different processing, storage, or quality control procedures as required. For example, a simple implementation might just copy the template

[9]https://www.rosepoint.com

[10]https://openseamap.org

[11]https://www.ngdc.noaa.gov/iho/

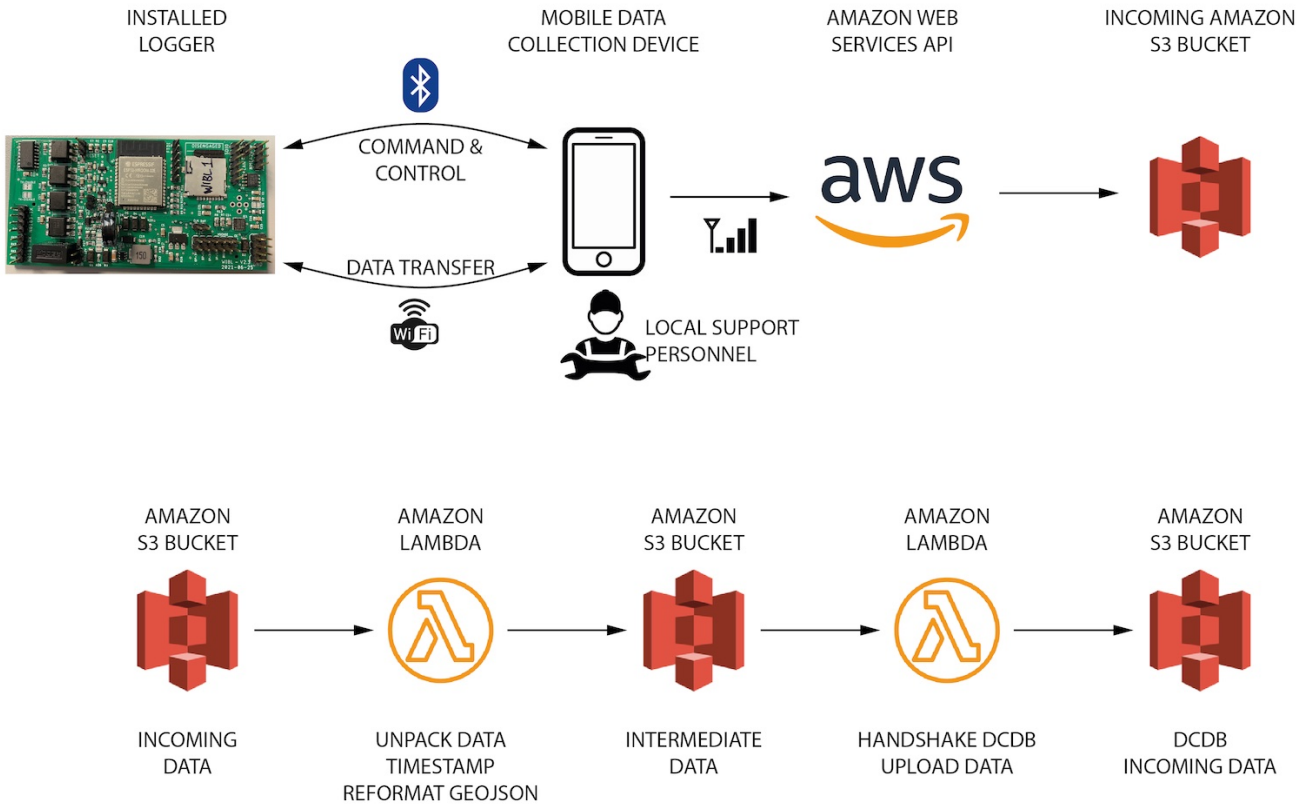

Fig. 1. Concept of Operations flow-path for data collected using the WIBL volunteer bathymetry logger system.

directly, while a more sophisticated implementation might add functionality to aggregate all of the available data into an intermediate product for volunteer feedback, or assess uncertainty and observer reputation to better qualify the data for further use.

This CONOP matches the requirements for an IHO Trusted Node, as proposed in the IHO guidance on VBI [16]. The Trusted Node (TN) model outlines a relationship between a data aggregator, typically an organization that provides software or hardware to record VBI, and the DCDB. The TN manages collection and aggregation of VBI, and agrees to format the data appropriately, provide minimal metadata, and manage the relationship with their volunteers. In turn, the DCDB makes available the upload API, issues upload tokens, and provides support to the TNs. The WIBL project provides the infrastructure to support this model, and has already been qualified to send data to the DCDB, making it easier for prospective TNs to set up their data path.

IV. System Design

The WIBL design is intended to be modular and portable so that different hardware implementations can be built to provide the same, or closely related, functionality. This allows individual implementations to innovate on the base design to support their own purposes. All of the details of the WIBL system are published through the project repository[12] under a permissive Open Hardware or Open Source Software license, respectively, including Geber files, pick-and-place positions, and a bill of materials for hardware production, full source code for firmware and mobile application[13], and Python code for the cloud processing segment.

A. Hardware Logger

In order to encourage wide adoption, the hardware segment of WIBL is designed to be as inexpensive as possible. The current implementation, Figures 2–3, is a straightforward design based on the low-cost ESP32-WROOM-32 embedded microcontroller System on Module (SoM), which provides a dual-core microcontroller, 4MB flash memory, and both Bluetooth LE and WiFi connectivity through a PCB antenna. Versions of the module with more flash are available, but the baseline model can be partitioned to provide two 1.9MB segments which are sufficient for firmware and an "over the air" firmware update space, and still have a small (∼200kB) space for non-volatile storage of configuration parameters in a virtual filesystem.

The logger is required to record position and depth information at a minimum, but must also establish a time reference for the received data. The most common interface for this information are those established by the National Marine Electronics Association (NMEA), either the older NMEA0183 (using RS-422 as the physical transport) or NMEA2000 (using CAN bus). The logger reference design uses two ESP32-internal hardware UARTs to support two channels of optoisolated transmit and

[12]https://bitbucket.org/ccomjhc/wibl.git

[13]https://bitbucket.org/ccomjhc/wibl-mobile-app.git

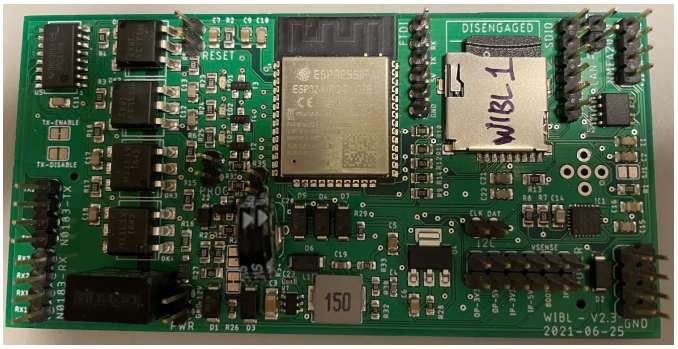

Fig. 2. Hardware for the current WIBL design, v. 2.4, showing the ESP32 module (center),optoisolated NMEA0183 interface (left), Micro SD card (right) storage, and NMEA2000 interface (far right). Power is conditioned from a 12V input through a switch-mode 5V regulator (for NMEA0183) followed by a 3.3V low-dropout (LDO) regulator for the remainder of the circuitry.

receive for NMEA0183 using an external transceiver, and an ESP32-internal CAN bus controller connected to an external transceiver for NMEA2000. As a speculative addition, a small six-degree of freedom MEMS motion sensor has also been integrated. Other data interfaces (e.g., NMEA over ethernet, or a NMEA to WiFi network bridge) are possible, but have been avoided on this initial implementation due to cost and complexity concerns.

Data is stored between download events on a Micro SD card, which can be adjusted in size to suit the project budget and expected time between download events. The prototype has been tested with cards up to 32GB. Faster cards (e.g., HC/Class 10) are recommended. The ESP32-internal SD/MMC controller is used at full speed (up to 40 MHz) to provide low-latency storage.

Power regulation for the logger is provided by a switch-mode power supply to regulate down from 12V input to the 5V supply required for the external transceivers. A 3.3V low-dropout (LDO) linear regulator is used for the ESP32, SD card, and motion sensor. A super-capacitor is included to provide a small back-up power source to allow the logger to shut down cleanly when power is removed, and power monitoring is provided to allow the ESP32 to identify this condition.

Recent production runs of the board indicate that small runs (of order five) boards can be made for approximately $20-25 each, and in larger volumes (of order fifty boards), the cost would reduce to approximately $8-10 each; further reductions are likely in larger volumes. Extra costs for connectors and enclosures would be expected to add perhaps $10 more per unit.

B. Logger Firmware

The logger firmware uses the Arduino framework for portability, relying on a NMEA2000 library[14] to drive the CAN bus transceivers, and an LSM6DS library[15] for the motion sensor. Native and Arduino support was used for Bluetooth LE, WiFi, SD card interface, and over-the-air firmware updates; custom libraries were developed for voltage detection, power and log file management, command interface, automatic polarity detection for RS-422 input, and status LED control. VSCode[16] and PlatformIO[17] were used for development and debug; all code is C++.

The firmware is structured as a one-time setup routine, followed by a run-loop that handles messages from the various interfaces, Figure 4. Measurements indicate that the run-loop executes in approximately 17–20 ms. All log data is serialized and written by a separate module to maintain coherency of the files on the SD card, while a configuration class is used to support key-value parameter storage, in this implementation in a small section of the flash memory on the ESP32 module which acts as a file system. Abstract base classes are used extensively in the firmware to allow for translation to other hardware designs.

The WIBL firmware implements a simple ASCII command language, and assumes that commands can be sent through a serial interface (debugging, development, initial programming), over Bluetooth LE (BLE) as a UART emulator, or via TCP/IP packets to a known port over WiFi. Responses are returned through the same interface to avoid cross-talk. In order to minimize radio pollution on the bridge of the host vessel, the mobile application uses BLE to configure and interrogate the logger, turning on the WiFi interface as required to transfer data. The command language includes features to configure which data sources are to be logged, the baud rates expected on the inputs, the WiFi and BLE parameters, etc. and to manipulate the log files (create new, erase, transfer, etc.). The logger can also store a user-supplied list of algorithms that should be run on the data (e.g., to de-duplicate depths, spike removal, etc.) in post-processing, and arbitrary metadata in JSON format to insert into the data files before transmission to the DCDB. Putting these in the logger allows for individual customization and avoids (cloud-based, centralized) database look-up in post processing. Data storage and transfer is binary using a custom file format for efficiency and compactness.

WIBL has no real-time clock, and therefore cannot natively timestamp data packets being received. The data streams have time information embedded, however, either in the form of NMEA0183 ZDA messages or NMEA2000 SystemTime packets, which are used to fabricate a preliminary estimate of time for real-time use. The firmware also records the millisecond elapsed time since module boot for each message logged (a roll-over of the 32-bit count will occur approximately every 49.7 days; this is detected and corrected in post-processing). This is not a reliable time source, but it is monotonic, which allows post-processing of timestamps.

C. Mobile Data Collection and Aggregation

The mobile segment of the system is used to aggregate data in the field, and provide transport to the cloud segment. Designed for the local technical support personnel installing and maintaining WIBL loggers in an area, but usable by interested

[14]https://github.com/ttlappalainen/NMEA2000.git, with the addition of ESP32 support from https://github.com/ttlappalainen/NMEA2000_esp32.git

[15]https://github.com/arduino-libraries/Arduino_LSM6DS3

[16]https://code.visualstudio.com

[17]https://platformio.org

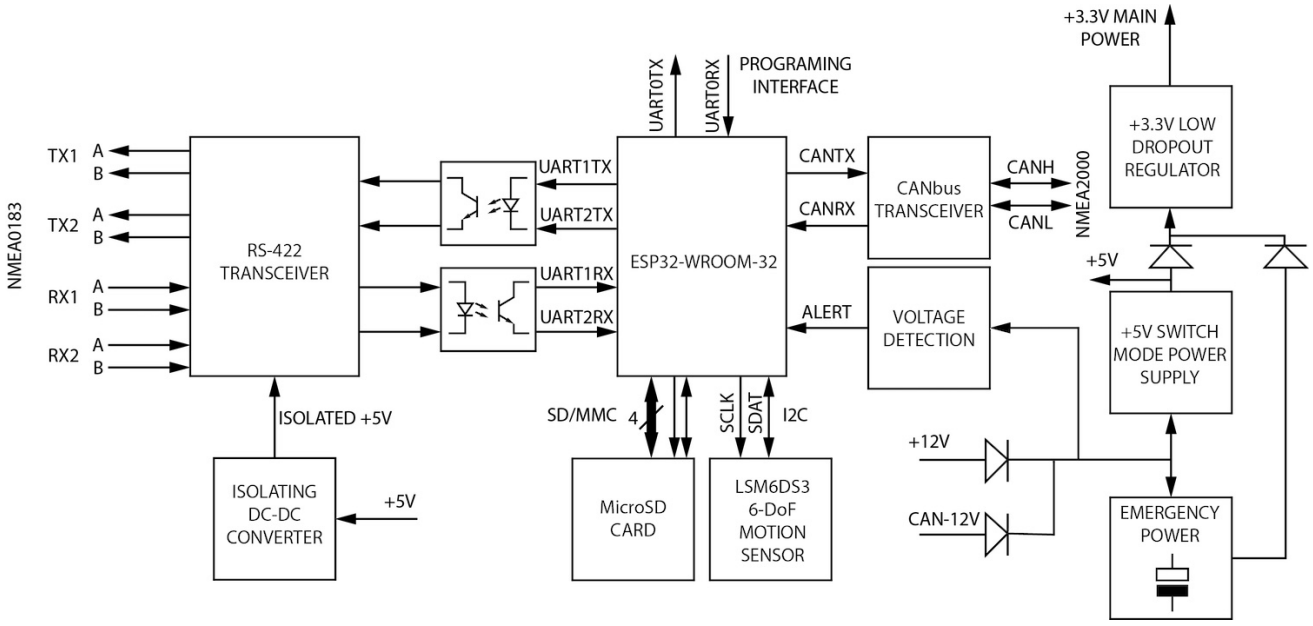

Fig. 3. Block diagram of the WIBL design, v 2.4, showing the major functional components. Full schematics and PCB layout are available in the project Git repository.

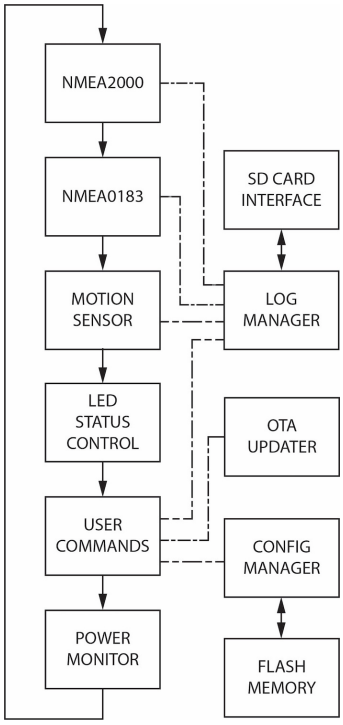

Fig. 4. Main run-loop for WIBL v. 2.4, with auxiliary modules for log files on SD card, over-the-air (OTA) firmware update, and configuration/parameter management.

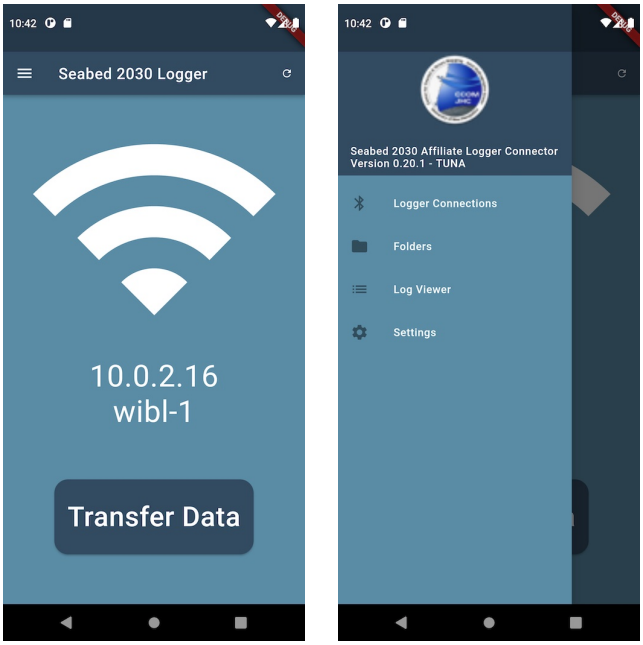

Fig. 5. Mobile application based display (left) and navigation page (right) used to configure and control loggers, aggregate data from multiple loggers, and upload to the cloud segment for post-processing.

volunteer collectors, the software (Figure 5) connects initially via Bluetooth LE to configure and control the logger, then switches to WiFi to off-load data more efficiently (although equivalent control can be conducted over WiFi if required). Data from multiple loggers can be aggregated on the mobile device before transmitting to the cloud when a better transport network becomes available. To avoid hardware and network costs and complexity, no provision was made for real-time transport of data to shore from the logger, although this could be added to the firmware if required (see Section VI for more details).

D. Cloud Processing and Transport

The cloud processing structure is by design modular so that particular implementations can adapt or expand the processing as required. Cloud-segment configuration can be automated

through desktop-based scripting ("infrastructure as code") to ease implementation complexity. In order to avoid having to manage server instances, the code is implemented as a pair of AWS Lambdas separated by an intermediate S3 bucket. AWS Lambdas are a "serverless" cloud service where user-supplied code (in this case in Python) is triggered to run under certain conditions, such as the upload of a file, or the notification of an event.

The first Lambda is triggered by the successful upload of a raw WIBL file into the appropriate S3 bucket, and carries out format translation, time-stamp generation, and metadata construction. The WIBL-specific binary data format is first unpacked, a time source is determined from the data, and then timestamps are interpolated for all positions and depths found based on the correspondence between elapsed time at the logger when the relevant time source messages were received, and the real time information contained in the messages. First order (linear) interpolation is used, and roll-over of the elapsed time is detected by observing that although the reception times may not be very accurate, they are monotonic so a reversal in elapsed time indicates roll-over. Subsequently, a second pass through the data interpolates timestamps for all positions and depths, and then interpolates the position information to the times of the depth observations. First order (linear) interpolation is again used, since the positions are typically delivered at 1 Hz. The processed data contains one observation per depth measurement, including time, position, and depth. Any recognized algorithm requests stored in the logger are then actioned to manipulate the interpolated data before the Lambda completes by converting the data into the required GeoJSON format [16], with associated metadata (stored in the binary file by the logger) that includes a unique identifier for the logger (and therefore the observer), a ship name, the logger make and version information, etc. If the logger has stored JSON metadata in the data file, it is included at this stage. The final result of the Lambda is stored into an intermediate S3 bucket to allow for further processing.

The time source algorithm allows for the possibility that information may be available from both NMEA2000 and NMEA0183 sources, and therefore examines the data during the first pass for available time information. In order of preference, the algorithm will use NMEA2000 System-Time, NMEA2000 GNSS, NMEA0183 ZDA, and finally NMEA0183 RMC.

The second Lambda is triggered by the intermediate file being generated, and manages the process of uploading the data to the DCDB. This is accomplished through upload to a specific URL provided by NOAA, including a recognition token, unique to the uploading implementation, provided by the DCDB for authentication and security. Files are given a unique name using a Universally Unique Identifier (UUID) algorithm.

V. Example Data Collection

Data was collected from the USCGC Healy during the HLY21TD expedition from Seward, AK to Nuuk, Greenland through the Northwest Passage in 2021. Due to the implementation of the data distribution on the ship, which used single-ended RS-232 rather than differential RS-422 as required by the NMEA0183 protocol, data was supplied to the logger by capturing network (UDP) broadcast packets of the NMEA strings for position, time, and depth on a laptop via WiFi; the data was then passed to the WIBL logger via a USB to serial converter. A specially constructed "pass through" mode in the logger's firmware was used to route this information out of one of the logger's NMEA0183 transmit channels. This channel was then looped back to the logger's input NMEA0183 channels using a pair of wires, allowing hardware capture and potential latency to be exercised. Data was provided by the ship at 1 Hz, although occasional network packet loss was observed which reduced this temporarily throughout the data capture. A total of 688,497 packets were observed, with 16,139 pairs separated by more than 1.5 s, indicating a total packet loss of 2.3%. The 99% quantile for inter-packet times was, however, 2.048 s and the 99.9% quantile 5.019 s, indicating that the most common loss was a single packet, although longer losses were observed. This is attributed to packet loss on the WiFi distribution network. The echosounding rate was, however, frequently below 1 Hz due to water depth, and therefore repeated depths were added to the data stream by the ship's broadcast systems to match the GNSS observation rate. The WIBL logger was configured to indicate that depth de-duplication was required at the processing stage in order to reverse the duplication before submission to the DCDB; the first of the set of repeated depths was preserved into the output data. After de-duplication, a total of 254,460 depth measurements were retained.

Due to communications limitations on the ship (as with many volunteer platforms), the data were downloaded and processed after the expedition. A total of 19 files (185.9 MB in 17 files of 10 MB, the maximum that the logger allows before swapping to a new file, one of 7.3 MB, and one of 8.6 MB) were transferred into the AWS S3 upload bucket, timestamped, de-duplicated, and then converted to GeoJSON for transmission to the DCDB ingestion point. The AWS processing Lambda was configured for a timeout of 30 s and memory allocation of 2048 MB. The average processing time was 0.945 s/MB (range 0.911–0.977 s/MB), with average start-up time of 1.071 s (range 0.968–1.243 s). The AWS run-time invoked four Lambdas for the 19 files, so the start-up time was amortized over multiple files in this instance, but for individual file uploads this cannot be guaranteed. Due to parallelization, however, the real-time processing burden was 64.39 s for the entire dataset, or 0.346 s/MB (speed-up 1.73). Clearly, there are significant advantages in batch upload of data. Average memory usage was 213.6 MB (range 183–236 MB), which was significantly lower than the configured memory for each invocation. Previous experience, however, had demonstrated that the allocated memory had significant effect of the overall run time of the Lambda, it was assumed due to swapping of code when the SciPy library layer was loaded into the Lambda. Testing with memory limits from 128 MB (the default) to 4096 MB demonstrated diminishing returns after 2048 MB, although further testing might be required for production implementations to ensure that this is consistent at scale. The

data were finally made available in a demonstration website by NCEI, Figure 6.

VI. Discussion

The WIBL project has demonstrated a minimal, but real-world, example of end-to-end collection, processing, and transmission of data. A number of questions remain unanswered, however. For example, although the current hardware prototype appears to be stable, it is unknown how well it will stand up to the rigors of the marine environment (e.g., when installed in a small-craft without environmental control unlike for the example data collection), which can be harsh. Longer, larger, and more widespread deployments are therefore required to prove the design in detail. It is also unknown whether the motion data from the on-board MEMS sensor is of sufficient value to warrant its place on the limited storage between download events. That is, it is expected that the sensor will provide at least an indication of the level of motion being experienced, which could be useful for uncertainty estimation, but it is not currently clear whether the quality of the data will be sufficient to provide motion corrections for the data.

Obtaining access to volunteer ships to install equipment, and to service it, can be difficult due to security concerns. Since the NMEA interfaces are part of the navigation suite of the ship, there can also be some reluctance to allow equipment to be attached to it. However, it is expected that all of the ships volunteering to collect data will be self-selecting with respect to these considerations; the question is then to what extent this restriction limits the number of potential observers. This remains to be seen. Alternatives to NMEA interfacing that might be less invasive would be to capture data over WiFi or Ethernet, and to use more modern protocols such as SignalK (http://signalk.org). This would require, however, that the volunteer collectors had this equipment available on the ship, which would again limit the pool of potential volunteers. This interface equipment is often expensive (e.g., 30 times the cost of the WIBL logger for one particular SignalK example), which would provide further barriers to entry. The project has therefore not pursued these options, although the ESP32 module used has both WiFi and a hardware Ethernet protocol engine for suitably motivated developers. Custom firmware, and in the case of Ethernet a physical layer interface (PHY), could be used to provide a NMEA to WiFi bridge, Ethernet interface, or SignalK implementation; some extension of flash memory in the ESP32 module might be required to support all of these functionalities.

Some models of VBI data collection (e.g., Orange Force Marine, ARGUS, DYNAMO) have proposed that loggers be enabled with ship-to-shore networking capabilities in order to off-load data either periodically, or in real time; reliable network transmission is emphasized [14]. The WIBL project has not pursued this option. First, the data transfer costs for getting data ashore from each logger would be much more than the cost of the logger over time, and the cost of the hardware to make the connection would also likely at least triple the production cost. Second, if the goal is to use ship systems to provide connectivity, relying on the ship to have infrastructure to allow for connectivity to shore would significantly reduce the number of volunteers who would be able to take part in data collection. Third, there is in practice little urgency in transmitting this type of data ashore: except in some very particular circumstances (e.g., in a narrow seaway [10]) rapid transmission between ships, or ashore, of volunteer data is unlikely to be useful. There is no expectation that volunteers will detect critical hazards (although this is possible), and therefore the added complexity of real-time transfer are not warranted. Finally, the IHO definition of volunteer data is that it is the result of observations taken in the course of normal surface navigation [16]. This is primarily a means to encourage the member states to support collection efforts, since many would not generally allow organized data collection (i.e., data collection with the specific aim of depth definition) without permits, for a variety of reasons. A system that supported real-time data transmission without oversight could then be considered in violation of this principle; to avoid potential complications, batch processing of aggregated data is considered safer. For all of these reasons, the WIBL project has not pursued the option of ship-to-shore transmission as part of the CONOP. Note, however, that the WiFi component of the ESP32 module can be put into "station" mode and joined to an existing network. Therefore, if a ship already has network connectivity, it would be possible to use it to transmit data. Putting the logger into station mode is part of the standard firmware; transmission to shore would have to be added by a suitably motivated implementation group.

By design, the WIBL system is modular and open source, allowing any group using the system to customize the code to their requirements. This would allow groups in jurisdictions where there is a requirement to report all volunteer data to the hydrographic office to add a delivery mechanism for reporting, for example, and in other cases would allow the group to add specific processing (e.g., uncertainty estimation, aggregation with other data, observer reputation estimation [5]). Unless modified by the sponsor group, however, the core code ensures that the data is transformed into the appropriate GeoJSON format required by the DCDB, and automatically triggers upload, ensuring that the data always flows to the international database in good order. One potential difficulty with this environment is that adding extra hooks to auxiliary processing would potentially diverge the implementation from the reference repository, which can make it more difficult to merge updates from the main development effort. A more modular structure would be beneficial, and a current development theme for the project is to provide for a state-machine model of processing using AWS Step Functions[18], which would facilitate more flexibility. This effort is mainly to allow for algorithms declared in the data stream to be executed automatically, but could also be used for implementation details. Currently, there is no agreed algorithm for processing of data of this kind, although some approaches [5] have been proposed. The project therefore opted to release the core implementation now to allow for feedback and adoption, and will consider adding processing modules as techniques

[18]https://aws.amazon.com/step-functions

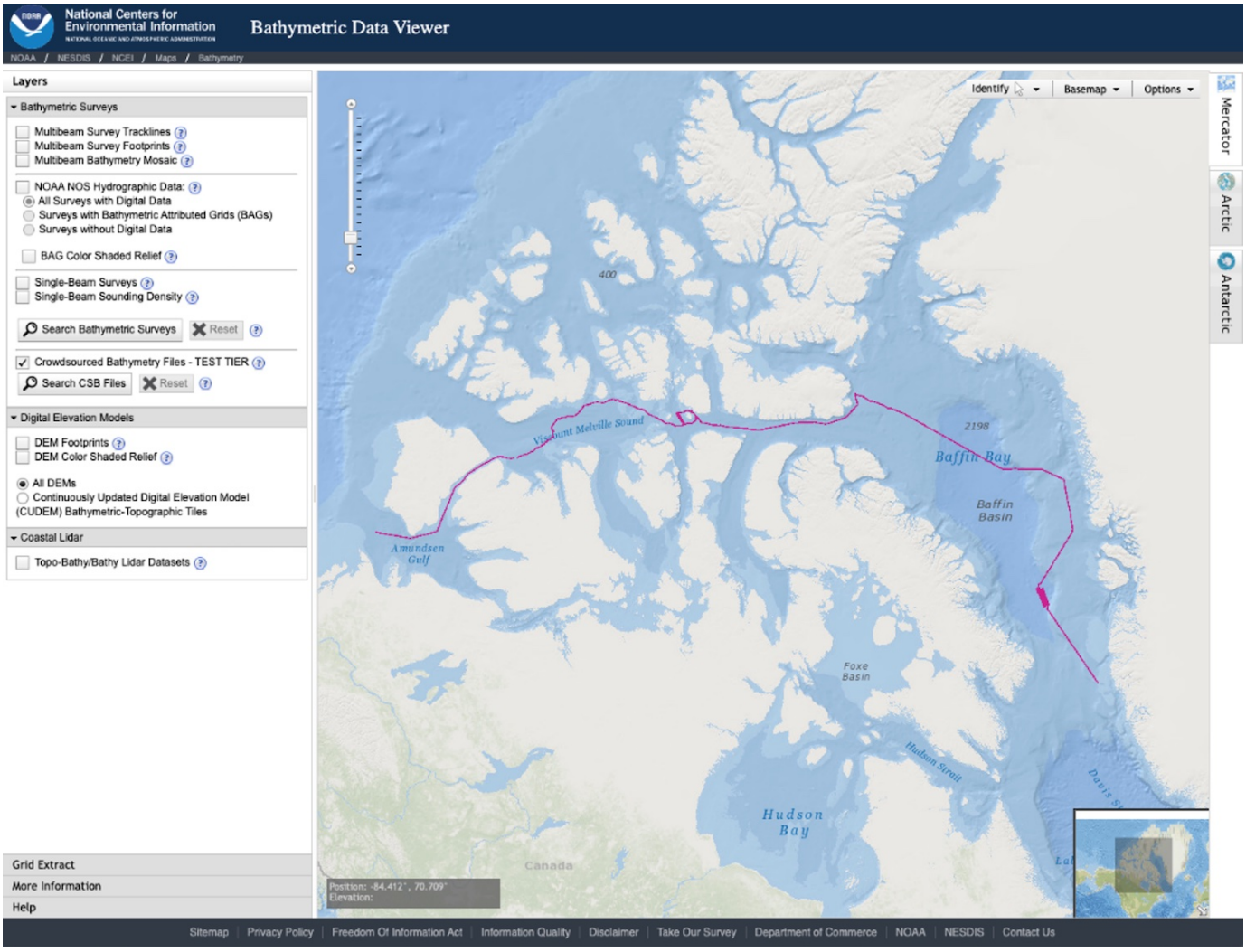

Fig. 6. Example data from the USCCGC Healy expedition, HLY21TD, through the Northwest Passage (west to east) as captured by WIBL and displayed in the DCDB volunteer data viewer, hosted by NCEI in Boulder, CO.

stabilize and improve.

It is not a goal of the WIBL project to be a data hub: the project is expected to be cloned for use in many locations, with each individual installation doing their own data collection and management. This underlies much of the design of the system, and is intended to allow data collections to scale in a federated (rather than centrally controlled) manner, allowing for more rapid scaling. The use of open source hardware and software also allows for much lower costs both for collection and processing, and lowers barriers to entry for even small groups (e.g., a local yacht club) that want to collect local data. Similarly, it is not a goal of the project to provide extensive data management and processing facilities. One of the purposes of the DCDB archive is to provide searchable and discoverable data, data extraction facilities, and (eventually) data processing; it is also considered a "hundred year", or long-term, archive. Relying on the archive to support these services significantly reduces development, test, and maintenance burdens on the project. The cloud segment is, however, modular so that a suitably motivated implementation could add data processing facilities if desired.

There is of course a fundamental complexity in setting up a data collection. For example, although the project provides all of the files required to have loggers manufactured, and "infrastructure as code" scripts to configure the cloud segment (a process currently being made easier by adoption of packaging and deployment tools such as Serverless[19]), it is likely that not all potential volunteer groups will want to build their own hardware, or would prefer not to manage data processing themselves. There are therefore opportunities for organizations, either commercial or non-profit, to provide these facilities as services, for example by packaging the cloud segment of WIBL and offering deployments of this package to interested volunteer groups (potentially with an enhanced user interface). In essence, the volunteer group would be renting an IHO Trusted Node. Similarly, it would be possible for an organization to manufacture WIBL, or WIBL-compatible, loggers at scale (therefore obtaining benefits of volume manufacturing in reducing overall costs) and then provide them as a package for volunteer groups. Clearly,

[19]https://www.serverless.com

costs would be important: a goal of the project is to make loggers that are essentially sufficiently cheap to give away. However, if this could be done, it might encourage more data collection efforts by smaller organizations who want a turn-key experience.

Although intended primarily to support WIBL loggers, the cloud segment of the project is a general-purpose processing framework for VBI, carrying out the sorts of tasks required by any logger of this type. It is therefore possible to retrofit other loggers into the scheme; the project includes log-file converters for both TeamSurv and YachtDevices loggers for this purpose. There are potential limitations to this conversion processing. For example, the TeamSurv device does not timestamp data, so relative times between data capture events are unknown and have to be approximated for interpolation; in addition, neither logger provides metadata associated with the platform. However, within these limitations, this facility can potentially allow for data loggers without a corresponding data infrastructure to be used for volunteer efforts. This is used, for example, in the Seabed 2030 Global Data Assembly Centre (hosted at Southampton Oceanography Center, UK) to provide data upload for collectors with no other IHO Trusted Node to accept their data.

VII. Conclusions

The WIBL project has demonstrated that it is possible to build (more than) minimally functional loggers for volunteer bathymetric information with hardware costs on the order of US$20 (2022) for a fully assembled system, essentially inexpensive enough to be handed out for free within a sponsored project. The system is also designed such that it can be cloned for use by even small groups that want to collect their own dense data in a localized area, and defines a concept of operations to support this. These features are expected to facilitate scaling of collection in a distributed manner, so that a single organization does not become a bottleneck. The system has been demonstrated through construction of hardware prototypes, collection of over 185.9MB of data, and submission of this data to the DCDB for archive. The totality of the system provides all of the requirements for an implementation to function as an IHO Trusted Node for volunteer bathymetry, although the individual implementation would have to go through the registration process to effect this. Given the standardized form of the WIBL system, however, particularly of the data processing, formatting, and metadata handling, this should be significantly simpler than qualifying a system *ab initio*.

WIBL provides an open model for volunteer data collection, allowing for individual groups to collect data to known standards and automatically transfer them to the DCDB for archive, while still maintaining control over use and subsequent processing of their own data, and having the ability to extend the processing path to meet individual requirements for data reporting or processing. The same cloud-based processing path can be used for other types of data loggers, so long as their data is converted first into WIBL format, which is well documented; the project already provides converters for two common loggers, and is readily extensible.

While implementation of a WIBL-supported data collection by a sponsoring group is well documented, not all groups will have the resources to build hardware and/or run cloud-based processing. Commercial or non-profit support of "Trusted Node as a Service" cloud implementations, or hardware logger packages are therefore expected, including potentially different, but hopefully compatible, hardware implementations. Doing so would provide a computing platform business model (a common and sustainable approach to scaling in the cloud), but more importantly strongly support many data collection events that would otherwise fall by the wayside for reasons of complexity. In turn, this would provide more and better data to projects such as Seabed 2030, with their critical goal of fully mapping our ocean planet.

VIII. Acknowledgments

The support of NOAA grant NA20NOS4000196 is gratefully acknowledged. The author would also like to thank the Captain and crew of the USCGC Healy, and especially the STARC support team for their help in acquiring data for the paper; the two teams of Computer Science undergraduate students (Christopher Schwartz, Hannah Dukeman, Taylor Roy, and Noah Perron in academic year 2019, and Jason Walerszak, Jason Worden, Connor Murphy, and Patrick Doherty in academic year 2021) who worked on the mobile and cloud segments; and Jenn Jencks and Georgie Zelenak at NCEI.

References

[1] J. McMichael-Phillips, "Seabed 2030 progress report," Nippon Foundation-GEBCO Seabed 2030, Tech. Rep., 2021.

[2] X. Lurton, *An Introduction to Underwater Acoutics: Principles and Applications*, 2nd ed. Springer Praxis, 2010.

[3] L. Mayer, M. Jakobsson, G. Allen, B. Dorschel, R. Falconer, V. Ferrini, G. Lamarche, H. Snaith, and P. Weatherall, "The Nippon Foundation–GEBCO Seabed 2030 project: The quest to see the world's oceans completely mapped by 2030," *Geosciences*, vol. 8, no. 2, 2018. [Online]. Available: https://www.mdpi.com/2076-3263/8/2/63

[4] B. Calder, S. Dijkstra, S. Hoy, K. Himschoot, and A. Schofield, "A design for a trusted community bathymetry system," *Marine Geodesy*, vol. 43, no. 4, pp. 327–358, 2020.

[5] B. Calder, "Estimating data and observer repution in mariner-volunteered bathymetry," *International Hydrographic Review*, vol. 25, pp. 77–96, May 2021.

[6] S. Hoy, "The viability of crowdsourced bathymetry for authoritative use," Univ. New Hampshire Center for Coastal and Ocean Mapping, Tech. Rep., May 2022.

[7] E. Novaczek, R. Devillers, and E. Edinger, "Generating higher resolution regional seafloor maps from crow-sourced bathymetry," *PLoS One*, vol. 14, no. 6, p. e0216792, 2019.

[8] M. Van Norden and J. A. Hersey, "Crowdsourcing for hydrographic data," *Hydro International*, 2012.

[9] T. Thornton, "TeamSurv—Surveying with the crowd," *Hydro International*, 2011.

[10] M. Rondeau and P. Dion, "Potential of a HydroBox crowd-sourced bathymetric data logger to support the monitoring of the Saint Lawrence waterway," in *Proc. Canadian Hydro. Conf.* Québec City, Québec, Canada: Canadian Hydro. Assoc., February 2020.

[11] J. Derochers, D. Ndeh, and K. Regular, "Crowd-sourced bathymetry in northern Canada," in *Proc. Canadian Hydro. Conf.* Québec City, Québec, Canada: Canadian Hydro. Assoc., February 2020.

[12] A. Reed, "Steps towards a true crowd: collecting bathymetry via electronic navigation systems," in *Proc. U.S. Hydro. Conf.* Hydro. Soc. Am., March 2017.

[13] A. Rosenberg, J. Jencks, E. Robertson, and A. Reed, "Mapping the gaps: Building a pipeline for contributing and accessing crowdsourced bathymetry data," in *AGU Fall Meeting*, Am. Geophys. Union. Am. Geophys. Union, December 2017.
[14] R. Montella, D. Di Luccio, L. Marcellino, A. Galletti, S. Kosta, G. Giunta, and I. Foster, "Workflow-based automatic processing for internet of floating things crowdsourced data," *Future Generation Computer Systems*, vol. 94, pp. 103–119, 2019.
[15] R. Beaman, "Crowdsourced bathymetry on the great barrier reef, australia," in *GEBCO Symposium*, GEBCO. GEBCO, November 2018.
[16] Int. Hydro. Org., "Guidance on crowdsourced bathymetry (B-12)," International Hydrographic Organisation, Tech. Rep., 2021.

Brian Calder (M'2002) received the M.Eng (Merit) and Ph.D. degrees in Electrical and Electronic Engineering from Heriot-Watt University (Edinburgh, Scotland) in 1994 and 1997, respectively. He joined the Center for Coastal and Ocean Mapping (CCOM) as a founder member in January 2000, where he is currently a Research Professor and CCOM's Associate Director. His research interests are primarily in the collection, quality assessment, and (especially) processing of bathymetric data.