# A Cloud based Solution in Hydrographic Data Processing: The Shift to a Web Centric Software Platform

Daniel B. Wright, *Process Researcher, Geopoint Solutions*, Charles E. Wright, *Principal Software Engineer, National Board of Medical Examiners*

***Abstract*— Currently available hydrographic data processing software is mostly limited to on premises installations, requiring annual licenses and a significant investment in hardware and data storage. This imposes hardware limitations on both the speed and capacity of processing large datasets. By leveraging the tools available through cloud computing, a Software as a Service (SaaS) data processing model is proposed. As implemented, many of the limitations imposed by on premises architecture are eliminated, but many new challenges are expected in bringing a SaaS processing solution to the field of hydrographic data processing. Using academically proven Open Source API's to build the conversion engine and create the requisite Bag output files, we will show how a cloud based solution accomplishes these tasks more efficiently and with a significant reduction in both time and cost over traditional on premise software. The model requires rigorous testing methodologies as well as the development of a secure and reliable web based interface. It will also be shown that the cloud architecture provides additional opportunities for the use of aggregated data to satisfy the evolving needs of chart producing organizations. With these concepts in mind, it is intended to demonstrate the functionality and benefits of the proposed processing system.**

***Index Terms*—Cloud Computing, Computer Security, Data Processing, Geographic Information Systems, Hydrography, Software as a Service, Software Architecture.**

## I. Introduction

THE motivation to design a web based application for the processing and storage of hydrographic data corresponds to the advances in integrated cloud computing environments. The goal of this project is to develop tools that utilize the benefits of this environment, with particular emphasis on the application development cycle, and the scalability of processing and storage resources. Some advantages of this approach are in a reduced time for new product releases, a system that can scale automatically to user demand, and the development of machine learning tools to quickly and accurately analyze very large data sets [1]. Other commonly acknowledged benefits of operating within a cloud based architecture include automated disaster recovery, reduced capital expenditures, increased team collaboration, document and data version control, higher level security, and a reduced environmental footprint [2]. However, managing the risks inherent in developing a new hydrographic data processing application, and the requirements of operating a secure and reliable user interface, present significant challenges. To manage this effectively, cloud infrastructure providers have developed tools which can be deployed that take advantage of the inherent efficiencies and mitigate the risks associated with operating in the cloud. In addition, there are significant opportunities to improve the data management pipeline, as well as develop tools for managing organizational efficiency and use of resources.

This project has as its primary goal to provide a cost effective and resource efficient means of processing and storing hydrographic data, while also creating a means to database bathymetry and other data used in the in the production of S-101 compliant navigation products. A predicate of this goal is the use of a web app that ingests hydrographic survey data and generates Bathymetrically Attributed Grid (BAG) files which can, at the producer's discretion, be utilized by chart producing organizations worldwide.

This paper describes the design of the project, the development process and the technology adopted. Details of the structure of a bathymetric database are also included and we conclude with what adoption of cloud technologies could mean for the future of hydrographic product development.

## II. Defining the Cloud: What we need to understand

The term "Cloud" and its many functionalities have numerous perceived definitions. For this paper, we rely on the National Institutes of Standards (NIST) definition of cloud computing (2011), which defines it as … *a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources*…. and *that can be rapidly provisioned and released with minimal management effort or service provider interaction* [3]." These elements represent the key utilization model of cloud resources. Further, NIST defines five essential characteristics and three service models as illustrated in Figures 1 and 2.

Submitted February 28, 2017, for the U.S. Hydro Conference, Galveston, TX . Daniel B. Wright is the Process Researcher for GeoPoint Solutions, Berwyn, PA 19312 USA (e-mail: dan.wright@geopointsolutions.us).

Charles E. Wright is with the National Board of Medical Examiners (NBME), Philadelphia, PA 80309 USA, (e-mail: CWright@nbme.org).

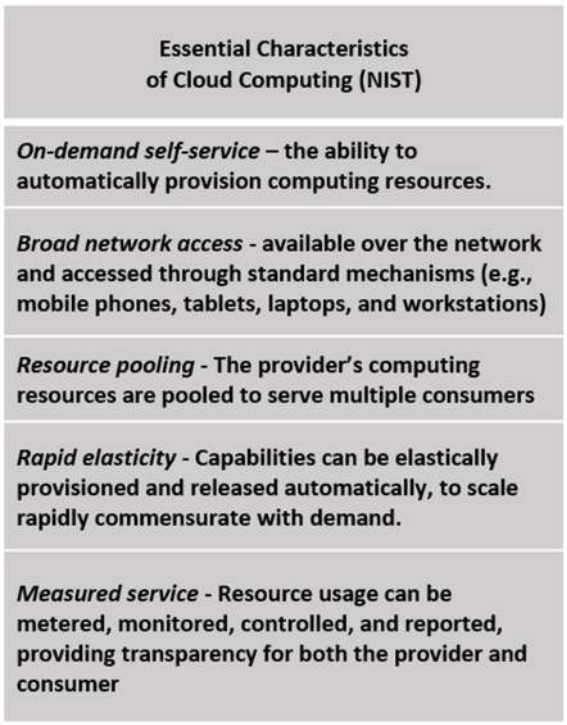


Fig. 1. Essential characteristics of a cloud computing model.

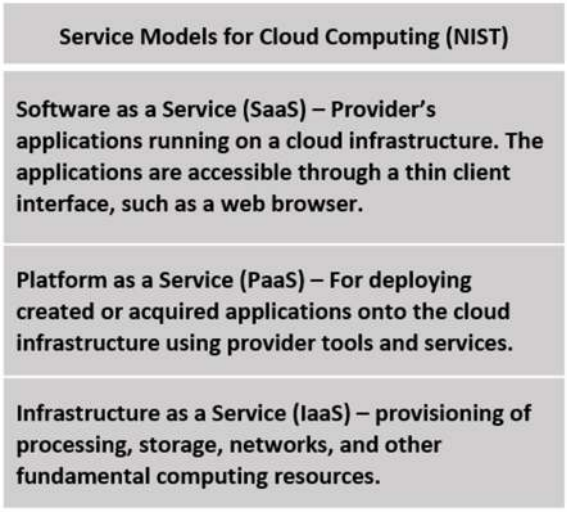


Fig. 2. Service Models for Cloud computing

The first and second mode of service models, SaaS and PaaS, are the most relevant to the processes described in this paper. It is also important to define the user roles in the context of both producers and consumers. In our context, an organization that collects hydrographic data, processes it through the web based application a defined as a data producer. A data consumer is any organization that uses the outputs of the processing, whether directly involved in the collection or not. As end users, they will also help to define and create new uses and representations of that data for their end users (i.e. Chart Producers). Of course, many organizations act in both roles, but a de-coupling of the inputs and outputs is required to define the specific functionalities of the application. In the design of the model, both roles use the SaaS user interface. As an App service provider, we operate primarily within the PaaS environment, since this is where the applications reside, and which utilizes the services available from cloud providers, as well as the infrastructure as part of that service.

## III. BASIC DESIGN OF A HYDROGRAPHIC CLOUD SERVICE

The goal of this project is to provide a client independent web based application that can perform the basic transformations of hydrographic data from "raw" collected data to an industry standard gridded format, entirely within a public cloud computing environment. In order to provide this service, an architecture needs to be defined. As much as possible, open source components have been utilized, as these elements are either produced by, or generally accepted by the academic and business community. These open source components also allow transparency of the code in many cases, as well as reducing the development resources required. In many cases, existing open source elements are well developed and provide a better user experience than developing tools in house. When open source is not a viable option, Microsoft Visual Studio is used as a collaborative development platform.

Industry research of the cloud services marketplace has identified many of the strengths and weaknesses of the major cloud providers, and the selection of a service remains a critical strategic decision. Independent research from Gartner, Inc. [4] has shown that only a few providers offer the full capabilities required to host a SaaS product of this scope. One of the key selection drivers results from the lack of portability, that is, a cloud service is not a commodity, and a SaaS product provisioned on one vendor's platform will not easily (if at all) transfer to another. This essentially locks the development and delivery to a single provider. In our case, we have selected Microsoft Azure to host the application, since their service evolved primarily as a PaaS provider, whereas other cloud providers have specialized in IaaS as their primary service. This, combined with the support provided through the Microsoft BizSpark program, has enabled the first stages of development of this project.

### *A. Architecture*

Systems architecture focuses on how the major elements in an application are used and interact with other major elements and components. The architecture components are primarily concerned with the public side of interfaces; details of the elements, the internal implementation, data structures and algorithms are all design concerns and function within the parameters of the user requirements to accomplish a specific task. [5]. An example of a cloud based system architecture for bathy data processing is shown in Figure 3.

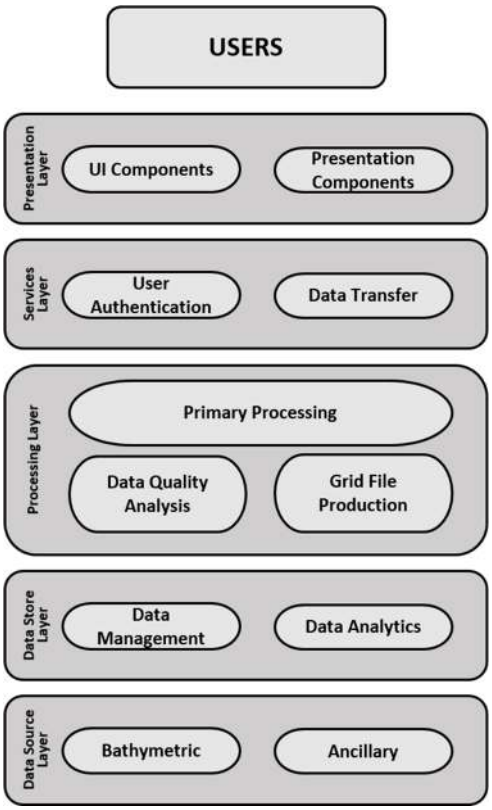


Fig. 3. Application architecture for Cloud computing.

### B. *Inputs, Outputs and Process Design*

In our current architecture, we limit the data inputs to the minimum required to produce a Bathymetrically Attributed Grid (BAG) file, as described by Calder et al., 2006 [6]. Due to the vast number of possible data types and configurations, the processing engine, as currently designed, will read only those data relevant to a sounding solution. Many modern multibeam systems produce data that are not directly relevant to the sounding solution, and by reducing the source data to only those that are required, there is a significant reduction in the bandwidth required to upload data for processing. This also has the effect of creating a clearly defined data stream for the processing algorithm [7].

A benefit of the cloud service model is that customized user configurations can be deployed and modified as needed without changing the fundamental architecture of the system. The user has the option to configure their data to conform to the defined input framework, or for those users that do not have the capabilities to modify or create the proper inputs, customized solutions could be deployed to the user as needed. Sample data files and vessel configuration from the user would be the basic requirements for this. For our simplified test case, we limit the input to the data relevant to the sounding solution contained in a Hysweep HSX file, a sound velocity profile in the CARIS format, and a single station tide file.

The processing engine is built using Object Oriented Design and utilizes the Python programming language. The availability of standardized formatting and math libraries such as numpy and pandas allows flexibility in design and accepted reliability of results. The downside is that this is not a language that is optimized for multi thread processing, and thus does not fully take advantage of the scalability of multicore cloud computing. Despite this limitation, it is possible to improve processing speed by scaling to multiple virtual machines and processing in parallel as described by Calder, 2013 [8].

Each module in the processing engine incorporates self-testing procedures that run with each job. Failures and out of bounds conditions can be immediately acknowledged and addressed in real time by the developer. In all cases, the developer is responsible for following best practices in developing a testing solution for each code module. Initially, the output of the processing engine is intended to produce only the required elements to populate a BAG file using the tools defined and provided by the Open Navigation Surface Working Group (ONSWG) in their format specification for a Bathymetrically Attributed Grid (Version 1.6) [9].

### C. *Accessibility, Elasticity and Storage Considerations*

Since use of web based applications is governed by the user's access to the internet, and this cannot be controlled by the SaaS provider, we leave it to the user to resolve any issues with connectivity. However, a distinction should be made between the connection requirements of accessing the system functions versus the connection speed required for transferring data. Since the user interface requires only a web browser and is platform agnostic, any thin client device (laptop, tablet, cell phone) can access the processing interface, with generally low bandwidth requirements. Data transfer represents a different concern, since speed and bandwidth will limit the transfer rate. In very general terms, 8hrs of unedited Reson 8125 multibeam and position/attitude data in HSX format takes about 15 minutes to transfer on a 50Mbs connection. If the files are reduced to contain only relevant data for the sounding solution, the time required could be reduced, or made over a less robust connection. Despite this limitation, the immediate benefit of transferring data to cloud storage is that it is immediately secured and protected with redundant backup. From a risk management perspective, this is a significant benefit.

Elasticity and scalability represent two of the most important features in the cloud computing environment. As per the NIST definition, elasticity allows the system to be "provisioned and released automatically" and to "scale rapidly commensurate with [user] demand". Scalability has two distinct modes, scaling up and scaling out. Scaling up allows the system to increase CPU capacity, memory, disk space, virtual machines (VMs), custom domains, certificates, staging slots, and allows this to all be done automatically. Scaling out refers to the number of instances running the application, that is, dedicated VM's running at any given time. In the system design, scaling out is reflected in the system's ability to accommodate the number of users at any given moment, while also accommodating the processing demands on the system. Most importantly, this can be provisioned to run automatically. This impacts both the user and the SaaS provider. For the data producer, there is no requirement to pre-provision computing or storage resources, since this is configured on an as needed basis

by the SaaS provider. As demand increases or decreases, the SaaS provider can respond with the appropriate level of temporary processing power and storage, and increase total storage capacity with little or no planning required.

Design and implementation of the cloud data storage and retrieval architecture represent one of the greatest challenges and opportunities for the SaaS provider, as well as the data producing and consuming organizations. As each new BAG file is created, an index can be generated which can then be used within a generalized reference system. With this in mind, there is an opportunity to create a path for potential consumers to all data that a producer chooses to make available. To accomplish this, a federated database system provides a viable alternative to managing a centralized data base system. In a federated system, each constituent database remains autonomous, and each data producer retains a unique entity [10]. If a data producer elects to share data in a federated system, it would acknowledge the existence of particular data set, ideally via the BAG metadata, which could then be retrieved via a common import/export schema. A data producer would always have the choice to make data either public or private, but with adequate incentive, previously unshared data could be made available to data consumers. Figure 4 is a representation of a federated access schema.

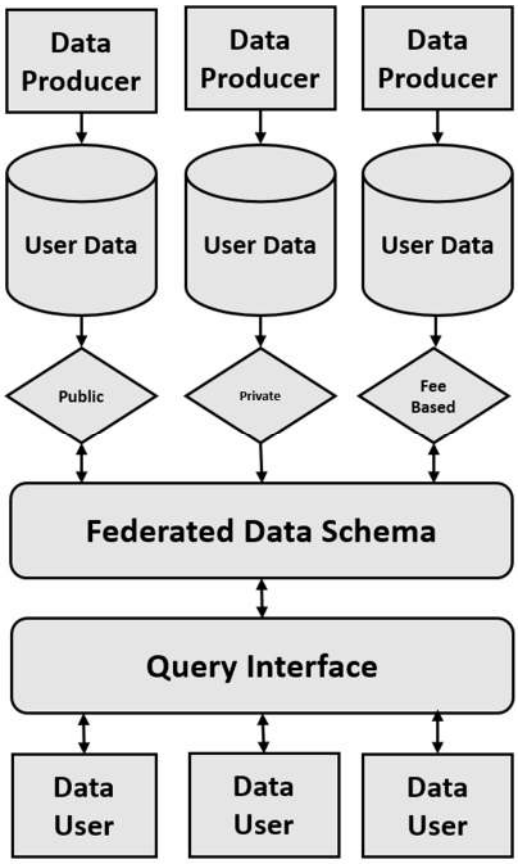


Fig. 4. Federated Data Base architecture for cloud computing.

Of particular interest is high resolution hydrographic data for use in S-101 compliant navigation products. BAG files represent the common certified representation of high resolution bathymetry for this purpose, and a generally available federated database could provide data producing organizations a way to distribute otherwise static information to potential consumers. Cloud computing serves as an ideal platform to achieve this goal, since it can provide the latent networking system to facilitate a common exchange platform. This could have the effect of making previously unavailable data accessible as a worldwide repository of hydrographic data.

Another important consideration for data producing organizations is that storage in the cloud resolves the issue of media obsolescence. Since it is the responsibility of the infrastructure provider to guarantee access to data over time, the burden of staying current is removed from the data producing organization, since the storage media is no longer a factor.

### *D. Security and Compliance*

For many organizations, security remains the most important factor in cloud service adoption, although this concern has declined as cloud providers have proven a high level of reliability [11]. In contrast, some smaller organizations have adopted cloud services for the *benefit* of added security and compliance, since these represent a significant investment in technology and expertise to manage effectively, and cloud providers have invested heavily in technologies that mitigate these risks.

In order to comply with national, regional, and industry-specific requirements of privacy and security, some cloud providers have incorporated compliance certifications into their infrastructure, essentially removing that task from the operational needs of the user. Of particular interest to the hydrographic community are the government specific compliances, for example FedRAMP and DoD (for US agencies), and ISO 27001 as well as many other country specific controls.

By design, many crosscutting security concerns are addressed by the use of authentication and encryption, as well as auditing and logging functions. Authentication provides the means of managing and controlling user identity and access to the applications, environment and data. Multi-factor authentication provides the highest level of security and is key requirement for maintaining a secure transaction environment. For uploading data between user devices and datacenters, and within datacenters themselves, standard transport protocols require encrypted communications and processes. For data at rest, (in storage) cloud providers can provide encryption capabilities that meet the highest available standard (AES-256). An example of standard security measures is shown in Figure 5.

Auditing and logging capabilities can serve multiple functions, and the ability to collect and analyze data across users in real time provides a viable and critical means of monitoring customer usage events that impact service. The ability to apply analytics to all elements of user interaction can benefit the development cycle as well, by streamlining the reaction time to trends in usage. Some PaaS providers have built visualization and monitoring tools into their core infrastructure

to provide these services, making it standard procedure to analyze these data in real time.

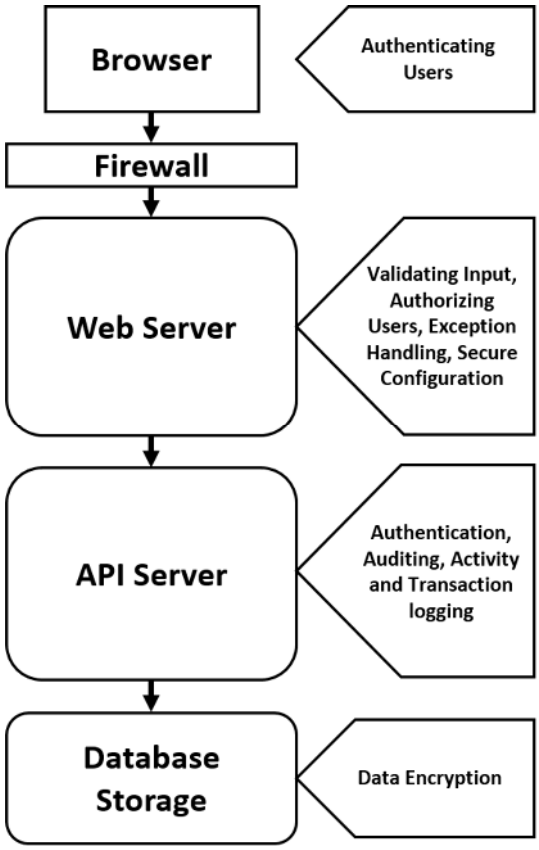


Fig. 5. Security schema for cloud computing.

## IV. CONTINUOUS DELIVERY, TESTING AND INTEGRATION

Producing code within a cloud infrastructure has inherent benefits to the producing organization as well as the user, including the ability to deploy software faster and with fewer defects. Principles and practices such as Continuous Delivery (CD) and Continuous Integration (CI) are ideally suited to a cloud platform and utilize automation and shared developer tools in the testing and delivery of products. The benefits of operating in a CD development cycle include improved customer feedback and satisfaction in terms of new features and maintenance, higher frequency of releases (six months to two weeks in some cases), improved quality and productivity through build/test automation, and reduced scope of releases. An additional benefit is the integration of Code Development and IT Operations (DevOps) which helps to reduce development "silos" where development occurs over long periods of time without connection to the operations environment [12].

The primary advantage of implementing these practices is that the code resides on the Web and API server and not on the user computer, which means the user has no requirement to update or re-install software. Testing can be done in parallel to running operations, and fully vetted before being deployed. Legacy code does pose some problems to implementation of CD and CI, since automated testing is part of the design, and some older open source API's are not well suited to this process.

## V. FUTURE PRODUCT INTEGRATION

As part of the system design, developing algorithms that look for anomalies first, then requiring the user to respond by editing the data, refocuses the emphasis away from visualization tools, and onto quantitative results. It remains common practice to first scan navigationally significant bathymetry (<50m) visually for systematic errors, using area based editing as well as time based (swath) editing tools [13] [14]. Although early studies using CUBE surfaces indicated many instances where manual or systematic editing was required prior to generating a surface (Calder, Smith, 2003) [15], with the ability to re-calculate CUBE surfaces of any size in near real-time, the rational for initial visual editing is greatly reduced. Through work done at UNH/CCOM, tools have been developed that automate some aspects of data analysis and Quality Control (QC) [16]. Developed as open source API's using Python, the HydrOffice Quality Control suite utilizes basic machine learning techniques to evaluate BAG files for predictable anomalies in a systematic way. These tools perform functions that are the first steps towards building algorithmic tools that could utilize machine learning practices on complete surveys.

The more we rely on visual editing, the less reliable and repeatable are the results. Our goal is to reverse the editing paradigm by creating an application that can load uncleaned, unedited data, generate a bag file and then offer a list of suggested editing parameters. Since the algorithmic tools to evaluate systematic errors have become viable practice, we intend to build on this expertise in ways only available in a cloud computing environment. This a fundamental shift in the way we evaluate multibeam data, but going back to the original intention of CUBE, why shouldn't we start by looking for exceptions in the final product before spending valuable resources doing manual edits? In developing a cloud based architecture, we can choose to invest our resources into analytical tools designed to identify exceptions, and our initial approach is to build only those editing tools necessary to meet the basic editing requirements. If re-processing a CUBE surface takes only moments, then the results can be observed and modified at will.

## VI. SUMMARY

Processing hydrographic data in a cloud environment can benefit from an architecture that supports a thin client user interface, cross cutting security, a federated data base system, and a continuous delivery model for code development and product support. It can allow organizations to take full advantage of the computing power and analytical tools just now becoming available, and help to develop a common data sharing and usage paradigm which addresses the needs of a range of end users of bathymetric data.

## ACKNOWLEDGMENT

Thanks go to the staff of the Microsoft Reactor, Philadelphia PA, for their advice and guidance, the Microsoft BizSpark Program for providing the computing resources used in this study, and the many others who have given valuable input and supported to this project.

## VII. REFERENCES

[1] J. Boers and L. Persson, "A View on Trends in Hydrographic Processing Software: In What Direction is Progress Headed?" *Hydro International*, vol. 20, no. 8, pp. 25–27, Nov./Dec. 2016.

[2] Sales Force UK & Ireland Blog. " Why Move to the Cloud? 10 Benefits of Cloud Computing." Posted Nov 17, 2016. [Online]. Available: https://www.salesforce.com/uk/blog/2015/11/why-move-to-the-cloud-10-benefits-of-cloud-computing.html

[3] Peter Mell and Timothy Grance. (2011, Sep.). Special Publication 800-145 The NIST Definition of Cloud Computing. National Institute of Standards and Technology [Online]. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[4] L. Leong, G. Petri, B. Gill and M. Dorosh, "Magic Quadrant for Cloud Infrastructure as a Service, Worldwide," Gartner, Inc., Stamford, CT, Tech. Rep. G00278620, Aug. 2016. [Online] Available: https://www.gartner.com/doc/reprints?id=1-2G2O5FC&ct=150519

[5] *Microsoft Application Architecture Guide, 2nd Ed.*, Microsoft Corp., Redmond, WA, 2009, pp. 03-*05. Available [Online]* https://msdn.microsoft.com/en-us/library/ff650706.aspx

[6] Calder B. et al., "The Open Navigation Surface Project" *International Hydrographic Review*, vol. 6, no. 2, pp. 1–10, Aug. 2005.

[7] J. Coleman, private communication, Feb 2017.

[8] Calder, Brian, "Parallel and Distributed Performance of a Depth Estimation Algorithm" (2013). Center for Coastal and Ocean Mapping. Paper 858. http://scholars.unh.edu/ccom/858

[9] Description of Bathymetric Attributed Grid Object (BAG), Version 1.6, Open Navigation Surface Working Group et al., Jun. 2016. Available [Online] http://www.opennavsurf.org/

[10] D. Heimbigner and D. McLeod, "A Federated Architecture for Information Management" *ACM Transactions on Office Information Systems*, vol. 3, no. 3, pp. 254–255, Jul. 1985.

[11] *2016 Future of Cloud Computing Survey*, North Bridge Growth Equity Venture Partners, Waltham, MA. http://www.northbridge.com/cloud-computing

[12] M. Leppanen et al, "The Highways and Country Roads to Continuous Deployment," *IEEE Software*, vol. 32, no. 2, pp. 64–72, Mar/Apr. 2015.

[13] C. LeBlanc, A. Roy, "CHS Atlantic Multibeam Processing Guide", Canadian Hydrographic Service, Atlantic Region, Feb. 2013. http://www.charts.gc.ca/documents/data-gestion/guidelines-directrices/sg-ld-2013-eng.pdf

[14] Field Procedures Manual, National Oceanic and Atmospheric Administration, Office of Coast Survey, pp. 154-155, April 2010 https://www.nauticalcharts.noaa.gov/hsd/docs/Field_Procedures_Manual_April_2010.pdf

[15] Calder, B. and S. Smith, "A Time/Effort Comparison of Automatic and Manual Bathymetric Processing in Real-Time Mode", Proc. US Hydro Conf. (Biloxi, MS), 2003

[16] M. Wilson, G. Masetti and B. Calder, "NOAA QC Tools: Origin, Development, and Future," presented at the Canadian Hydrographic Conference, Halifax, Nova Scotia, CA, May 16-19, 2016.

## AUTHOR BIOGRAPHIES

**Daniel Wright** earned his MSc (2004) in Marine Science from the University of Southern Mississippi and a Master's in Business Administration from Northeastern University (1990). After finishing his MSc at USM, he worked for NOAA as a Physical Scientist and aboard the NOAA ship Thomas Jefferson as Chief Hydrographic Survey Technician.

**Charles Wright** earned his MSc (2004) in Computer Science from Villanova University and has worked 25 years as a Software Engineer for the National Board of Medical Examiners (NBME) in Philadelphia.