

## I.T. Infrastructure Modernization Committee Report - July 2019

### **Background**

The Committee was formed by the Executive Committee in May, 2019 to investigate the state of the ARRL's Headquarters IT infrastructure and to develop a roadmap for its modernization and possible migration to cloud computing. In the process the Committee was tasked to investigate the opportunities for operational cost savings and improvements to redundancy and disaster recovery.

The ARRL operates several IT applications. These can be roughly categorized into public facing and enterprise applications. The former serves Membership and the general public. The latter services internal back office operations of the League. An investigation and evaluation of the League's applications are needed to determine the opportunity for cost savings, advantages in redundancy and a potential ease of the physical move of HQ staff, if this is ever needed. It could also support telework and improve the ease of accessibility by remote League volunteers and staff.

Currently the League runs most of its applications on-premises at ARRL HQ on purchased or leased servers.

#### Public facing applications:

- Main public facing Website
- Logbook of The World (LoTW) and Online DXCC
- Contest log submission, results, records
- Mass e-mail to Divisions and Sections

#### Enterprise applications:

- E-Mail
- File sharing
- Payroll/HR/Finance
- Association Management
- Learning management
- Publishing systems

Desktop computers and similar infrastructure were not considered and is outside the original scope of this project.

VoIP cloud phone service could be considered, but for now the Committee considers it to be outside the original scope of this project.

### **Initial meeting**

The Committee initially met on June 27 via teleconference and discussed some goals for the project as well as the League's current inventory. Present were Director Jairam, Director Holden, CEO Michel and IT Manager Keane.

Two possible approaches for a cloud migration were discussed at the initial meeting, and what it would take to implement either approach. These approaches are known as "Lift and Shift" and rearchitecting.

### **Lift and Shift:**

“Lift and shift” would entail migrating existing applications to the cloud without modification. Data and applications would be copied “as-is” and spun up on virtual cloud server instances.

**Pros:**

It would be not difficult to accomplish this migration. Very little re-architecting of applications would need to take place.

Applications would behave predictably as they would essentially be running the same way with the same code.

We would accomplish one of the goals of getting rid of physical hardware maintenance. This would eliminate things like hardware maintenance contracts and server network maintenance.

We could build in redundancy and scaling with demand load fairly easily.

**Cons:**

There could be little to no cost saving from this approach. Studies in other organizations have shown this approach to be the least effective cloud migration strategy. This would be due to the applications not being aware of the principal advantage of cloud infrastructure - being able to use only what computing resources you need.

We would end up with the same old applications and if they are due for retirement or upgrade, the effort to re-architect would need to be done anyway, thus duplicating effort.

IT support would still have to maintain server operating systems to some degree, negating the advantage of cloud computing of being able to destroy/re-create instances as needed.

**Re-Architecting approach:**

This approach would involve some re-development of ARRL applications to more efficiently utilize cloud infrastructure. This would take into account the availability of Software as a Service (SaaS), Platform as a Service (PaaS) as well as Infrastructure as a Service (IaaS).

**Pros:**

Running cost and performance would be better than lift and shift. For example, using a database service such as Amazon RDS would be significantly less effort than using a database running on a virtual server instance (EC2). Further cost savings could be realized by using serverless computing using lambda functions or similar. Lambda functions do not require a dedicated instance. They start up, take input, run, produce output and when done, exit memory.

**Cons:**

This approach could involve significant application re-development. This could tax existing resources or require new resources.

**Cloud steps already implemented by ARRL:**

E-Mail is on Office 365. This is a cloud-based email service from Microsoft to which many organizations have migrated. The cloud-based service saves the cost and complexity of running Microsoft exchange servers on-premises.

The Mintz and Hoke project proposes Software as a Service applications in its plan, rather than on-premise applications.

**Logbook of the World:**

LoTW is a perfect candidate for studying a migration, as it is public facing and can benefit from scaling.

Preliminary reports were prepared for the LoTW Committee with options for Database replacement. This included a limited cost comparison for Amazon AWS and Rackspace cloud services along with physical hardware replacement. Physical hardware replacement came in significantly less. However, this preliminary report appears to suggest only a “lift and shift” versus re-architecting. Utilizing a database such as Amazon RDS was apparently not considered.

Therefore, the Committee believes it would be prudent to explore this more in-depth to determine the costs and potential savings to be realized by using a properly re-architected application.

**Proposed next steps:**

The Committee needs to obtain a full inventory of all applications, average usage, and web traffic. We have already obtained a list of all servers; however, this limited information is not sufficient to permit an accurate costing analysis.. The Committee needs traffic numbers and average loads for the existing servers. This information is necessary to enable accurate cost comparisons. In particular, CPU load, traffic and database load are necessary for estimating the size of instances needed. We also need to consider peak demand such as after major contests, DXpeditions and special events. ARRL IT should be able to supply this.

Vice-Director McIntyre from the LoTW Committee has proposed a trial of replicating the LoTW database to a database service such as Amazon RDS and using that as the database for LoTW on a trial basis. These services have database connectors that are compatible with the most popular databases, therefore it would not be necessary to extensively re-code LoTW to run a trial. The feasibility of implementing such a trial should be studied. Such a study is especially pertinent and timely due to the planned development of Logbook 2.0 and online DXCC. We should also try to run a LoTW application in the cloud and run load testing to simulate production conditions. In light of this, we would like to propose adding Vice-Director McIntyre to the committee. Her expertise in software and I.T. as well as interfacing to the LoTW committee will prove valuable in this exercise.

The Committee should initiate an investigation of the costs, feasibility and benefits of migration of mass e-mail systems for email to Divisions and Sections to a SaaS platform. This would entail the use of commercial services such as Constant Contact or similar services. Such robust commercial email systems would permit the League to obtain metrics on its emails to Members; metrics that include opens, deletes, spam reports and similar. It would also help us more fully comply with anti-spam laws and future proof us in that regard. Constant Contact and others have special discounts for non-profits.

The Committee should ascertain whether source code management system such as Git is being used. Additionally, we propose that a cloud-based system such as GitHub be used to store code. GitHub is now free for basic use with private repositories included. For more users/collaborators, pricing starts at \$7 per month with enterprise starting at \$25 per month with 5 users included.

As part of a possible cloud migration, templatization should be investigated. Templatization stores server definitions in code permitting easy re-deployment.

An analysis and documentation of what is or what will already be moving to cloud platforms is also needed. This would include the Mintz & Hoke project. Such an analysis would avoid the duplication of efforts.

A cost analysis of moving each of the League's existing applications would be prepared and provided in a future report to the Board

**Conclusion:**

A great deal of investigation and analysis remains to be done, but the Committee believes it has a good roadmap. Cloud migration also presents an opportunity to start fresh in a green field. Cloud computing will require a change of mindset where one does not treat servers like pets that need to be fed, watered and to which one must pay constant attention. Instead, cloud computing allows the flexibility to create, run and dispose of resources when done. It allows more flexibility and no long-term commitments.

The majority of successful and well-run organizations have moved or are in the process of migrating all or a substantial portion of their IT operations to cloud computing. The cost savings of the maintenance and development of IT resources, increases in operational and personnel efficiency and security are just some of the reasons why companies have moved to the cloud with more of them moving daily.

The Committee believes the completion of its tasking will permit the League to properly evaluate its options and will permit the Committee to recommend a beneficial strategy for the League's IT future.

Respectfully

Director Ria Jairam, N2RJ (Chair)  
Director Matt Holden, K0BBC  
Vice-Director Kristen McIntyre, K6WX  
I.T. Manager Mike Keane, K1MK  
CEO Howard Michel, WB2ITX

## Glossary of terms used

- **Application** - a type of computer software that allows you to perform a specific task. Eg. - Microsoft Word, Logbook of The World (LoTW)
- **Architecture/Architecting** - the fundamental structures of a software system and the discipline of creating such structures and systems.
- **Back office** - an office or center in which the administrative work of a business is carried out, as opposed to its dealings with customers. In the case of applications, this refers to applications used to carry out these functions.
- **Cloud** - Software and services that run on the Internet, instead of local servers
- **Cloud server instance** - a virtual machine running in the cloud.
- **Customer facing** - business software products hosted on your systems used by your customers (members, advertisers etc) to interact with your company.
- **Database connector** - a facility that allows client software (eg. application) to talk to **database** server software, whether on the same machine or not. Eg. JDBC, ODBC
- **Database service** - a web service that makes it easier to set up, operate, and scale a relational database in the cloud. Eg. AWS Relational Database Service (RDS)
- **Disaster recovery (DR)** - an area of security planning that aims to protect an organization from the effects of significant negative events such as natural or man-made disasters. DR allows an organization to maintain or quickly resume mission critical functions following a disaster.
- **Infrastructure as a Service (IaaS)** - where a cloud provider hosts infrastructure components such as servers, storage and networking hardware and provides this on-demand to customers. Eg. Amazon Elastic Compute Cloud (EC2).
- **Lift and shift** - the strategy of removing workloads and tasks from one storage location and placing them in another, usually cloud-based, location.
- **Office 365** - Enterprise office software suite from Microsoft.
- **On-premises** - applications and servers hosted physically at the company's offices or other premises.
- **Platform as a Service (PaaS)** - a category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app.
- **Redundancy** - the inclusion of extra components which are not strictly necessary to functioning, but are there in case of failure in other components. Eg. redundant hard disks, redundant power supplies, redundant servers
- **Scaling** - the ability to increase the capacity of existing hardware or software by adding resources, including adding multiple server units to act as one logical unit.

- **Serverless computing** - a cloud-**computing** execution model in which the cloud provider runs the server, and dynamically manages the allocation of machine resources. eg. AWS Lambda
- **Software as a Service (SaaS)** - a method of software delivery and licensing in which software is accessed online via a subscription, rather than bought and installed on individual computers.
- **Source code management system (SCM) or Source Code Control System (SCCS)** - a system designed to track changes in source code and other text files during the development of a piece of software. This allows the user to retrieve any of the previous versions of the original source code and the changes which are stored. SCMs or SCCSs may also serve as code repositories which provide a central place to store and archive code.
- **Telework** - the practice of working from home or locations other than the offices of the company, making use of the Internet, email, and the telephone.
- **Templatization** - the practice of storing server definitions in files, usually in a source code management system. This allows server instances to be repeatedly created easily.
- **Virtual machine or virtual instance** - a computer file, typically called an image, which behaves like an actual computer but may exist completely in memory of another computer and may not run directly on physical computer hardware.