iTAG: Developing a Cloud Based, Collaborative Animal Tracking Network in the Gulf of Mexico

R.Currier[1], B. Kirkpatrick[1], C. Simoniello[1], S. Lowerre-Barbieri[2], J. Bickford[2]

[1]*Gulf of Mexico Coastal Ocean Observing System, Texas A&M University, College Station TX*
[2]*Florida Fish and Wildlife Conservation Commission, St. Petersburg FL*

***Abstract*— The tagging and tracking of aquatic animals using acoustic telemetry hardware has traditionally been the purview of individual researchers that specialize in single species. Concerns over data privacy and unauthorized use of receiver arrays have prevented the construction of large-scale, multi-species, multi-institution, multi-researcher collaborative acoustic arrays. A workshop[1] to understand the informational needs and concerns of potential contributors to an integrated animal telemetry network in the Golf of Mexico was held at the Florida Fish and Wildlife Conservation Commission (FWC), Florida Fish and Wildlife Research Institute in St. Petersburg, FL 29-30 May 2014. Input from workshop attendees provided a basic blueprint for the development of iTAG: Integrated Tracking of Aquatic Animals in the Gulf of Mexico. Key points made by attendees included solving the 'orphan/mystery' tag problem and ensuring machine-level data compatibility with the work being done by the Ocean Tracking Network[2].**

I. INTRODUCTION

The tagging and tracking of aquatic animals using acoustic telemetry hardware has traditionally been the purview of individual researchers that specialize in single species. Concerns over data privacy and unauthorized use of receiver arrays have prevented the construction of large-scale, multi-species, multi-institution, multi-researcher collaborative acoustic arrays.

A workshop to understand the informational needs and concerns of potential contributors to an integrated animal telemetry network in the Gulf of Mexico was held at the Florida Fish and Wildlife Conservation Commission (FWC), Florida Fish and Wildlife Research Institute in St. Petersburg, FL 29-30 May 2014. The workshop was sponsored by the Gulf of Mexico Coastal Ocean Observing System Regional Association (GCOOS-RA) and the Southeast Coastal Ocean Observing Regional Association (SECOORA).

Input from workshop attendees provided a basic blueprint for the development of iTAG: Integrated Tracking of Aquatic Animals in the Gulf of Mexico. Key points made by attendees included:

- Solve the 'orphan/mystery' tag problem
- Guarantee data privacy
- Establish a sense of community and collaboration
- Facilitate a regional approach to addressing bigger science problems
- Ability to monitor aquatic animals over larger spatial scales
- Ability to integrate detection data with oceanographic data
- Arrays that can be used to track multiple species
- Mechanism to share resources – data, equipment, expertise
- Integration with other telemetry networks
- Use of data for outreach and education

II. INITIAL DEVELOPMENT

Development of iTAG began using the Meteor[3] web framework. Meteor is based on Node.js and is a complete open source platform for building web and mobile apps in Javascript. We were particularly attracted to Meteor because of the built-in authentication module that allowed us to rapidly prototype a working user management model. As development progressed, however, the limitations of the provided authentication system became apparent. Customization was difficult, and the standard login procedure did not map well to the 'request to join/approval issued' model that iTAG required. We also had concerns about the compatibility of Meteor's only supported database, MongoDB with the animal acoustic telemetry metadata convention proposed by the Ocean Tracking Network.

Midway through the development process it became apparent that Meteor was not the optimal solution. We elected to change to the Flask micro-framework and the Python language, with PostgreSQL/PostGIS as the database. Twitter Bootstrap was selected as the User Interface library. Authentication and user management is provided by Stormpath, a cloud based security service. The iTAG application is hosted on a Digital Ocean droplet cloud server instance. With the new toolset in place we were able to rapidly progress to the point of a pre-beta release, with user management and 'orphan/mystery' tag tracking implemented along with the ability for researchers to enter receiver and

978-0-933957-43-5 ©2015 MTS

tag information. Initial feedback was been overwhelmingly positive. The privacy policy has been praised for its granularity: principal investigators can choose between three levels of privacy for all data and hardware:

- Completely private – viewable only by the PI
- Visible to iTAG members
- Visible to the general public

Privacy levels are symbolized by a three-color scheme: red for private, yellow for members-only and green for the general public. *(fig.1)*

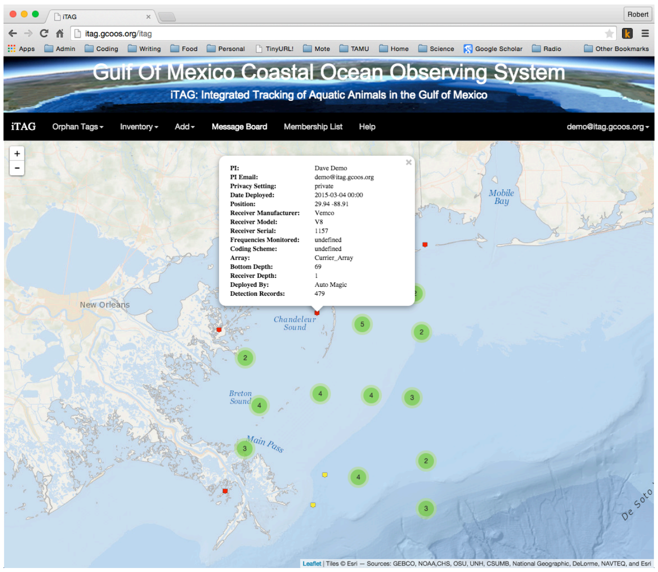
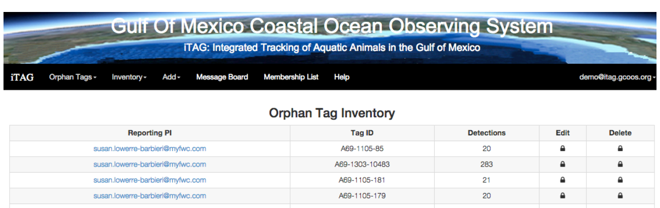


Fig. 1 Receiver pop-up box displaying receiver attributes

Orphan tags are displayed with the tag ID number and the date of detection. The email address of the reporting PI is hyperlinked allowing for one-click communication between a researcher searching for a missing tag and the reporting PI. The reporting PI has edit/delete capabilities for their orphan tags. Other member's tags are locked and unable to be changed or deleted. *(fig.2)*



Orphan Tag Inventory

Reporting PI	Tag ID	Detections	Edit	Delete
susan.lowerre-barbieri@myfwc.com	A69-1105-85	20		
susan.lowerre-barbieri@myfwc.com	A69-1303-10483	283		
susan.lowerre-barbieri@myfwc.com	A69-1105-181	21		
susan.lowerre-barbieri@myfwc.com	A69-1105-179	20		
demo@itag.gcoos.org	A70-999-9999	12		

Fig. 2 Orphan tag display for logged-in members

III. DEVELOPMENT LANGUAGE AND FRAMEWORK

We chose Python[4] as the development language. Python is widely used in the scientific and research communities and is beginning to become the de-facto standard for research programming. The selection of Python allowed for the use of Flask[5], a micro-framework for Python basked on Werkzeug[6] and Jinja 2[7].

Flask takes a minimalist approach to web application development. Flask does not include many features that other frameworks do such as database abstraction layers or form validation. Rather, Flask supports extensions that allow the developer to add only the needed functionality without being forced to import unnecessary code.

IV. DATABASE

PostgreSQL was selected as the database and PostGIS as the spatial database extender. The database schema adheres as closely as possible to that proposed in Payne et al. 2013[8].

Maintaining schema-level compatibility with the proposed metadata convention for animal acoustic telemetry will allow iTAG to share data with other projects that adhere to this standard. At the time of publication iTAG has implemented the receiver_deployment and tag_release schemas. All of the mandatory fields are included. Some optional fields are excluded and may be added at a later date.

```
                              Table "public.tag_release"
          Column           |           Type           |                      Modifiers
---------------------------+--------------------------+--------------------------------------------------------------
 id                        | integer                  | not null default nextval('tag_release_id_seq'::regclass)
 pi_email                  | text                     |
 pi_name                   | text                     |
 release_project_reference | text                     |
 datacenter_reference      | text                     |
 tag_device_id             | text                     |
 release_guid              | text                     |
 release_reference_id      | text                     |
 release_reference_type    | text                     |
 release_latitude          | double precision         |
 release_longitude         | double precision         |
 release_datetime_utc      | timestamp with time zone |
 expected_enddate          | date                     |
 tag_manufacturer          | text                     |
 tag_model                 | text                     |
 tag_serial_number         | text                     |
 tag_frequency             | text                     |
 tag_coding_system         | text                     |
 transmitted_id            | text                     |
 transmitter_name          | text                     |
 transmitter_type          | text                     |
 tag_programming_id        | text                     |
 privacy                   | text                     |
 the_geom                  | geometry                 |
Indexes:
    "tag_serial" UNIQUE CONSTRAINT, btree (tag_serial_number)
```

Fig. 4 Schema for tag_release table

V. MAPPING ENGINE

Development of iTAG began with Google Maps as the mapping engine. When we switched platforms from Meteor to Python/Flask we also switched mapping engines from Google Maps to the Leaflet[9] library.

Leaflet is an open-source JavaScript library for building detailed interactive maps. Leaflet is very lightweight. The entire library is contained in 33KB of JavaScript and is fully supported by a wide range of desktop and mobile platforms.

Selecting Leaflet as the mapping engine allowed us to use ESRI's Ocean Basemap and the PruneCluster[10] real-time marker clustering library.

VI. AUTHENTICATION

As noted in the introduction, a major reason for switching from the Meteor development platform to Python/Flask was the limited authentication features built into Meteor. Extending the authentication functionality would have a required a major investment in time and greatly complicated the process.

We tested several authentication frameworks and found them all to be complex and subject to errors in coding which would have potentially put iTAG at risk. Building one's own authentication framework is not recommended. After evaluating several candidates we elected to use the Stormpath authentication service.

Stormpath is a cloud-hosted user management applications programming interface (API) with multiple language support. Stormpath makes it easy for developers to authenticate, manage and secure users and roles. We were able to bring up the authentication service for iTAG in under an hour. Stormpath provided registration, account, group, directory and app management services as well as role-based access control and security workflows. By taking advantage of the Stormpath API we were able achieve sufficient granularity in the iTAG privacy settings to address the 'guarantee data privacy' concerns raised by the workshop attendees.

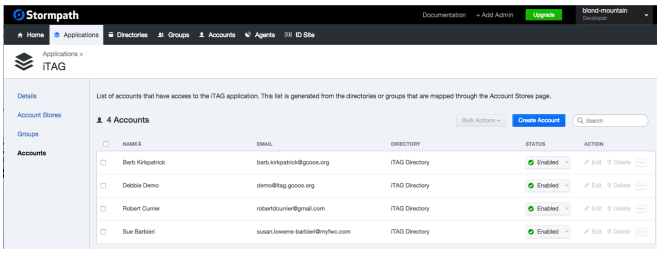


Fig. 5 Stormpath User Management Page

VII. Community Feedback

During the development process several informal meetings were held where iTAG was demonstrated to a group of scientists and technology professionals. Multiple video conferences with staff from Ocean Tracking Network (OTN) took place between November 2014 and May of 2015.

Based on the feedback from the informal meetings and OTN video conferences the corresponding author concluded that he did not have sufficient subject matter expertise to build iTAG into a system that the tagging community would embrace. It was decided that the author should spend some time engaged in a hands-on discussion of tagging practices and methodologies with active taggers to better enable him to understand the entire process.

On July 27th 2015 the corresponding author met with Dr. Sue Lowerre-Barbieri and Joel Bickford of the Florida Fish and Wildlife Conservation Commission (FWC), Fish and Wildlife Research Institute. (FWRI) Over the course of the day, Dr. Lowerre-Barbieri and Mr. Bickford took the corresponding author through the entire tagging process, starting with receiver and tag preparation and concluding with receiver retrieval and data downloads and exporting. At the end of the day the corresponding author was able to make significant changes to the data architecture and layout of iTAG based on information provided by Dr. Lowerre-Barbieri and Mr. Bickford.

VIII. Conclusion

At the onset of the development process we were uncertain if many of the significant obstacles to sharing data through a communal web site could be overcome. Data privacy was at the forefront of every discussion about iTAG. Implementing iTAG using the Stormpath authentication library allowed us to fine-tune the privacy settings and satisfy the requirements of the majority of stakeholders.

While the initial deployment of iTAG will be limited to orphan/mystery tag reporting we hope that after a trial period the tagging community will embrace iTAG and begin to share receiver and tag data. As the amount of data stored in iTAG grows we will begin testing Representational State Transfer (REST) communications between iTAG and OTN.

Acknowledgments

The authors wish to thank the Gulf of Mexico Coastal Ocean Observing System Regional Association (GCOOS-RA), Florida Fish and Wildlife Conservation Commission and the staff of the Ocean Tracking Network for their support and encouragement. The authors also wish to thank Stormpath for providing the Stormpath authentication library and Python bindings, without which iTAG could not have been built.

References

[1] http://gcoos.tamu.edu/?page_id=6714
[2] http://oceantrackingnetwork.org/
[3] https://www.meteor.com/
[4] https://www.python.org
[5] http://flask.pocoo.org/
[6] http://werkzeug.pocoo.org/
[7] http://jinja.pocoo.org/docs/dev/
[8] http://ioostech.googlecode.com/files/AAT%20Metadata%20Convention%20v1.2.pdf
[9] http://leafletjs.com/
[10] https://github.com/SINTEF-9012/PruneCluster