MoveApps-EarthRanger Integration for storage and management of processed movement data

# Overview & Scope

A recent collaboration between North Carolina Zoo, St Andrews University, DMP Stats and EarthRanger developed a data processing and analysis workflow in the *MoveApps* platform to process movement data from GPS-tagged Vultures. The analysis leverages the innate efficiency of Vultures in detecting large carcasses to identify mortality events, supporting the surveillance and conservation of endangered species in protected areas.

While the developed workflow met most of its requirements in terms of data processing and analysis, shortcomings were identified in the storage of processed data – particularly in maintaining the integrity of cluster metrics for long-lasting clusters during data updates.

The main goal of this this work package is to extend the developed MoveApps workflow by implementing a new *MoveApp*-*EarthRanger* integration to enable the permanent storage and updating of behaviour- and cluster-annotated tracking data outputted by upstream Workflow Apps. This integration can be split into three main tasks:

1. Retrieval of relevant historical (i.e. previously processed) movement data stored in the EarthRanger (ER) system
2. Combination of new and historical data, with focus on the correct updating of cluster-membership information
3. Upload of updated data back to the ER system

In addition to subject-level movement data storage, the current workflow will also need adjustments to enable the uploading of dedicated cluster-level data to the ER system. Current expectation is that the existing *EarthRanger Integration* app will have in place the required coding infrastructure to accommodate the sought functionality, but minor alterations to the current code are likely necessary.

This proposal includes an optional budget for collaboration on the writing-up of a scientific article describing the developed workflow.

Work and costings outlined here pertain specifically to DMP Stats. However, some of the tasks will depend on key inputs and collaboration from EarthRanger. ER-specific requirements to each task are detailed alongside their descriptions.

# Work Plan

## MoveApps-EarthRanger Tracking Data Storage

Aimed functionality is currently envisioned to be handled within a single MoveApp for optimal coding efficiency and robustness. However, implementation constraints may require dividing tasks into multiple MoveApps.

### Task 1 - App structure planning

Design App’s functionality and requirements, including:

* 1. Define the user interface, with clear and interpretable input parameters
  2. Specify input specifications and their validation rules
  3. Identify pre-processing requirements and expected outputs
  4. Define App’s code structure and relevant helper functions

### Task 2 - Implement the retrieval of historical tracking data from EarthRanger

The primary purpose of this step is to retrieve subject-level location data from historical records hosted by EarthRanger, within its ‘Observations’ class, using HTTP API requests. The fetched data should include:

* All observations annotated as members of currently active clusters (see **Task 3** below for the definition of “active” clusters)
* non-clustered observations recorded within a given time period relative to observations contained in the object passed on to the app as input.

In addition, the incorporation of a **spatial filter** into the API query is highly desirable to minimize data transfer costs. This filter would limit downloaded data to historical observations occurring near the location points in the App’s input data. However, implementing these filters depends on EarthRanger's agreement and development time, especially if spatial filtering is not currently supported within the ER system.

Within the MoveApp, the development of this task is expected to broadly include:

1. Constructing and Executing HTTP Requests:
   * Develop the data processing functionality required to build and execute the API queries (e.g. ensure JSON format compliance to API expectations)
   * Address data transfer limitations by supporting batched requests to handle large datasets efficiently.
2. Validation of HTTP responses
   * Implement validation checks to confirm the successful retrieval of queried data.
   * Ensure the returned data meets specified requirements (format, completeness, and content accuracy).
   * Develop mechanisms to identify and handle errors or inconsistencies in the API responses, with clear logging and feedback for troubleshooting.

The following text box specifies anticipated requirements from EarthRanger to enable the successful operation of this task.

|  |
| --- |
| Task 2: EarthRanger’s Requirements and Contributions |
| 1. ER’s ‘Observation’ class data   Enable the storage (and retrieval) of the following attributes via additional columns or a JSON list:   * + MoveBank-issued IDs (location event, animal, tag, deployment);   + Behaviour category   + Cluster ID   + Cluster Status (e.g. “active”, “closed”)   + Optional: addition of auxiliary subject- or track-level data, if feasible  1. MoveApps – ER’s API Integration    * Provide assistance in constructing the required HTTP requests for the ER API.    * Ability to query data based on specific attributes, including those outlined above under Task 2. |

### Task 3. Update cluster annotations and behaviours

This task focusses on updating observations in the historical data retrieved in Task 1 with newly processed data, provided as input to the app. The updating process will likely be based on the following logic:

1. Compare datasets, identifying new and repeated location events
2. Handle new locations:
   * If clustered, determine if they belong to existent “active” clusters or form new “active” clusters. Locations forming newly detected clusters are labelled with brand-new cluster IDs.
   * If non-clustered, bind them to the historical dataset as-is
3. Process repeated locations:
   * If new and old versions are both clustered, check for changes in cluster membership and update the cluster ID accordingly
   * If clustering states differ (e.g. historical version is clustered but new version is non-clustered), assign or drop cluster ID based on the latest state.
   * If behaviour classifications differ, overwrite old entries with behaviour categories provided in the newly processed data
4. Mark clusters as:
   * “active” if membership has changed at least once within a user-specified time period relative to the current app run time
   * “closed" if no changes in membership have occurred within the specified time frame.

Summary output(s) detailing the latest changes to the cluster membership and status, as well as updates to behaviour classifications, will be provided as App artifact(s).

### Task 4. Upload updated data back to EarthRanger

This step involves uploading the newly updated data back to the EarthRanger’s ‘Observation’ records via HTTP POST/PATCH requests to ER’s API. The implementation within the MoveApp will include steps similar to those outlined in **Task 2** and can be summarised as follow:

1. Data manipulation for upload, including the preparation for batched requests to handle limits to the volume of transmittable data.
2. Construction of the API request(s), including the conversion of tabular data into JSON format.
3. Validation checks to confirm successful uploading of data

|  |
| --- |
| Task 4: EarthRanger’s Requirements and Contributions |
| * Provide support for the construction of the HTTP requests for uploading the updated data to the ER system * Similar needs to those already outlined for Task 2 |

### Task 5. Error Management, Documentation, Testing and Refinement

To ensure App usability, robustness and overall reliability, the following measures will be undertaken during App development:

1. **Error Handling**: incorporate mechanisms to handle unexpected issues effectively. Provide clear, user-friendly error messages to help users resolve the most common issues.
2. **Automated testing**: develop and implement comprehensive automated testing procedures, including unit-test and app-level tests, to safeguard expected functionality, behaviour and integrity across various use cases.
3. **Documentation:** prepare detailed, user-friendly documentation for app users and developers, covering usage guidelines, setup instructions, and technical details to support future maintenance and enhancements.
4. **Beta testing and Refinement**: submit App to online Beta testing on the MoveApps platform to trial it under most common use cases. Collect feedback from testers on usability and potential areas of improvement. Refine the App to optimize performance and user experience.

## Support required modifications to current *EarthRanger Integration* MoveApp

As mentioned in the overview, additional adjustments to the existing workflow may be necessary to integrate changes introduced in this work package. Specifically, the *EarthRanger Integration* MoveApp might require modifications to enable the direct upload of cluster-level data (including cluster metrics and importance/priority) as "Event" records in the EarthRanger system, bypassing the current cluster-digestion process applied to uploaded vulture data.

These modifications are expected to involve relatively simple changes, such as altering specific input parameters (e.g., the "server" endpoint or "event\_type"). However, should deeper adjustments to the App’s source code become necessary (e.g. different JSON formatting), DMP Stats will provide the required development support at no additional cost.

At this stage, the scope of changes needed on the EarthRanger side to implement these modifications is unclear.

## Scientific Paper Co-authoring (Optional)

This optional add-on covers DMP’s contribution for drafting a manuscript detailing the workflow and methodologies employed. While lack of funding will not hinder DMP's participation in co-authoring the paper, providing financial support for this task would enable DMP to prioritize its resources, accelerating the manuscript's preparation and submission for publication.