# CS486C – Senior Capstone Design in Computer Science

## Project Description

| **Project Title: Cloud-Based Planetary Ephemerides** | |
|---|---|
| **Sponsor Information:** | Kelvin Rodriguez, Computer Scientist<br>USGS Astrogeology<br>krodriguez@usgs.gov<br><br>Amy Stamile, Computer Scientist<br>USGS Astrogeology<br>astamile@usgs.gov<br><br>Christine Kim, Computer Scientist<br>USGS Astrogeology<br>chkim@usgs.gov<br><br>Adam Paquette, Computer Scientist<br>USGS Astrogeology<br>acpaquette@usgs.gov |

### Project Overview:

Sensor models are an essential component in the processing of planetary imagery for NASA; they convert from sensor coordinates (i.e., charged-couple device (CCD) coordinates) to other spatial reference systems (SRS), e.g., longitude and latitude coordinates. They allow scientists to accurately geolocate imagery, reconstruct topology, convert from different planetary reference frames (e.g., coordinates relative to Mars, Earth, or the Sun), photogrammetrically adjust for positions and orientations, and project images onto planetary bodies, to name a few use cases. Image Support Data (ISD) are required to create sensor models in software. These data are the image's interior orientations (e.g., the sensor's focal length, CCD size, distortion coefficients, field-of-view), exterior orientations (e.g., instrument position, velocity, rotation, and angular velocity), and identifying metadata (e.g., instrument name, image ID, creation time). Without ISD, scientists cannot create planetary maps supporting NASA missions such as the Europa Clipper mission, launching in 2024 to research if Europa can support life.

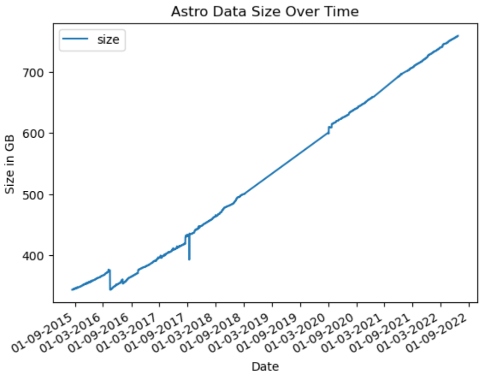

Figure 1: Size of USGS Astro's online SPICE data store since 2015. It steadily increases as SPICE data is created for new and existing missions by about 4GB a month.

USGS software writes ISD for an image in a plain text JSON-formatted ISD file. These files range from a few kilobytes to a few megabytes of data. ISD files, in turn, are derived from NASA's SPICE system, a dataset totaling over a terabyte of data. Currently, if users of our software want to generate an ISD file to instantiate a sensor model, they must download some portion of the SPICE dataset. This problem is a massive roadblock for new users which is only getting worse every year (Fig. 1).

To solve this problem, the capstone team will create a service on top of existing USGS software that can enable users to request ISD over the web. They will not have to understand the SPICE or ISD implementation details, that software already exists and is open source (https://github.com/DOI-USGS/ale). The goal is to create a web service that stands on top of this library and accepts requests for ISD (e.g., Flask, FastAPI). Some of the challenges are:

1. What is the minimum amount of data you can send over the web to generate an ISD?
2. The returns can sometimes be somewhat large JSON-formatted plain text. What can be done to reduce the data footprint? Consider binary conversion, compression, or some combination of the two.
3. Scientists often create ISD in large quantities, sometimes thousands of images at a time in High Powered Computing environments. Consider how this service could scale to thousands of simultaneous requests.
4. Generating ISD can be time-consuming as it requires querying large amounts of SPICE. Memoization can speed this up. What is the best way to cache these inputs? How do you handle cache invalidation when the SPICE data store updates?
5. We use AWS for our web services. A prototype should be able to run on AWS free-tier services. Consider a design that is AWS-friendly.

Success in this project will enable USGS and NASA scientists to process images at a large scale on both on-prem and cloud-enabled clients without being burdened with large amounts of data being downloaded. This will also lower the barrier of entry for future research students entering planetary science.

A summary of the requirements:

- A Python-based web service that accepts planetary image labels and SPICE kernel quality parameters via web requests. This can be a RESTful service that dispatches to ALE.
- A memoization mechanism that enables the instant retrieval of ISD on the backend.
- A new format that allows for reduced ISD data footprint which can be unpacked to an ALE-supported ISD format.
- A cache validation routine. The routine should invalidate ISD when SPICE data is updated, but only those affected.
- Stretch goal 1: Create a queryable format for the ISD cache.

### Knowledge, skills, and expertise required for this project:

- Experience with Python required.
- Basic knowledge of command line interfaces.
- Familiarity with the basics of RESTful interfaces and web services.
- Familiarity with database management systems, Redis, or similar services for querying or caching data.

### Equipment Requirements:

- A computer with Python and networking support. Linux, Mac, and Windows are all fine.

- A free-tier AWS account for stretch goals.

**Software  and other Deliverables:**

- MVP: A GitHub repository with code for a RESTful service under the CC-0 license. Repo should also include all documentation needed to run and maintain the service.
- MVP: A mechanism to quickly stand up the service on another computer such as Docker for user testing.
- MVP: Design and architecture documentation on the software.
- MVP: A well-written report detailing the design and implementation of the product in a complete, clear, and professional manner.
- MVP: Demo showing the web service running on free-tier AWS account with successful requests coming from a laptop.
- MVP: Cache validation service to invalidate the ISD database or cache based on SPICE update events.
- Stretch: A queryable data store of image ISD that can be downloaded or downloaded from.