Allsky Camera Network for Detecting Bolides
Milestone 4

Members
Tyler Turner, tturner2021@my.fit.edu
Vincent Quintero, vquintero2021@my.fit.edu
Jean-Pierre Derbes, jderbes2021@my.fit.edu
Charles Derbes, cderbes2021@my.fit.edu
Faculty Advisor/Client
Csaba Palotai, APSS, cpalotai@fit.edu

Progress Matrix for Milestone 4:

Task	Completion	To Do	Tyler	Vincent	Jean-Pierre	Charles
Implement UI	50%	Total integration of node controls Total database integration Adjustments req'd by researchers	10%	60%	0%	30%
Polish Server	99%	Modify some endpoints and move some services due to event detection moving to the node.	50%	20%	30%	0%
Polish Client	99%	Identify every service that requires an internet connection and come up with a way to intelligently run the node in offline mode.	30%	20%	20%	30%

Client hardware interaction	50%	Return hardware status when status endpoint is hit, camera calibration	30%	10%	50%	10%
Create setup process for node	100%	Done	20%	0%	10%	70%

Discussion of each accomplished task for the current Milestone:

- Task 1: Implemented the beginnings of a web based user interface enabling researchers to monitor and modify nodes as needed. The interface is set up to allow researchers to remotely customize the configuration/settings of individual nodes, such as setting certain time windows to record for events, what camera settings should be used, etc. A framework for the page layouts and interactions is made and some real-world data is already being populated. Additionally the interface allows researchers to more easily view collected events and evaluate if they even care about new events, or if they are not bolides.
- Task 2: Server now stores all of the configs for each node making it the sole source of truth for node configuration. Changes to the config are "PATCH" requested to the node which then updates its internal config which is referenced by services (such as video recording which on a config update needs to be able to reschedule the video recording to match the new start and stop times). Also added more endpoints for the UI to utilize such as the /analyze and /events endpoint. This also includes the functionality to read and write records to the database. Server serves weather data to the frontend using information from a particular node's config (latitude and longitude) and openweathermap.
- Task 3: Client polish ran into a roadblock because event detection had to be moved from the server to the client due to concerns with poor internet bandwidth on the client side. As a result more focus was put into making sure all events and videos make it to the server i.e. they are not being sent when the internet connection on the client side is down. IoT must also be reevaluated because users must be able to remotely connect to the box in situations where there is no internet connection in order to add or update wifi credentials or even change the network the box is using. The box only has one network interface which causes a lot of problems. One such problem is determining when it is a good idea to swap to the hotspot given that internet outages could be short lived. Also another important question is when the box is broadcasting the hotspot how does it check to see if known networks are available to connect to since that causes the control panel web page to be reloaded due to the wifi network swapping to check connection.
- Task 4: Basic testing for sensors is in place, and the node can now test its hardware on startup. GPS integration for accurate time and position is now done. This should work towards preventing hardware related issues from being "unsolvable" because they are assumed to be software related. The sensor testing code is also easily expandable (i.e you can add more sensors to be tested) since it is written in python and there are interfaces for easy python interaction with GPIO pins.

- Task 5: Node setup process is controlled via an ansible script (creates SSH keys, installs APT dependencies, installs and activates tailscale, sets up the python virtual environment, grabs repo etc.). Physical testing (beyond the basic software tests that are done on the pi) such as verifying the quality of the box (no leaks etc.) has been defined through a clear set of instructions (verify pin connections, verify box is sealed properly, etc.). Lastly software is written to test individual components to see if the pi is receiving data from them and to specifically verify that the camera is recording properly (this test is not checking to see if the logic behind events is "working" since that would be software related, this is just to see if the camera has been damaged, the image is blurry etc.).

Discussion of contribution of each team member to the current Milestone:

- Tyler Turner:
  - Assisted Vincent in making the UI functional by adding the backend functionality that is required for the front end. Created the UI demo that will be presented to the class. Implemented tools for the teams to improve code quality such as a lint and formatter. Implemented the trash can functionality that will delete unwanted videos from the server after a specified time period. Introduced the openweathermap api to use gps latitude and longitude to get the nodes geographical location such as city, state, or county while also getting the weather associated with that location.
- Vincent Quintero:
  - UI conception, frameworking, development. Worked with researchers to determine how they most prefer to interact with the site when analyzing events. Early integration of database to frontend. Support for event detection and segmentation design.
- Jean-Pierre Derbes:
  - Worked on camera video code (primarily the event detection and recording schedule). Previously video code had been implemented but due to the need to record events on the node this had to be redone. Worked on state management (config truth source is located on server, therefore the node needs to sync its config with the server config which can be problematic if the server, node, or connection is down). Changing config in the database on the server for the node now sends a config "PATCH" request to the node.
- Charles Derbes:
  - Worked on client connectivity and client design in order to make the client robust to internet outages and other failures. Original work division didn't really play out this milestone as most of the time was wrapped up in addressing the need to move event detection to the node. Worked on IoT (had to be reevaluated to integrate with the connectivity changes), particularly how and when the hotspot should be activated to minimize self inflicted outage prolongation.

Task Matrix for Milestone 5:

Task	Tyler	Vincent	Jean-Pierre	Charles
Client Hardware Interaction	50%	0%	25%	25%
Orbit, trajectory, velocity, mass (of bolide)	0%	0%	20%	80%
Client connectivity logic	10%	0%	20%	70%
Poster and e-book	25%	25%	25%	25%
Finish UI	20%	80%	0%	0%
Server and client testing	40%	30%	20%	10%
Video capturing and storage, move event detection to node	0%	0%	80%	20%

Discussion (at least a few sentences, ie a paragraph) of each planned task for the next Milestone:

- Task 1: Client hardware interaction involves making sure that all of the sensors and devices connected to the node can be verified as functional. It also involves communication with the devices i.e. to ensure emails are being sent out when there is water in the node, to ensure the camera can be properly calibrated, to ensure that the code works with a wide variety of different cameras should the researchers choose to upgrade etc. In order to move forward with trajectory and mass calculations we need to be able to calibrate the camera. Lastly, the hardware interaction being functional allows researchers to diagnose issues as hardware related rather than software related.
- Task 2: There is a feature request to be able to determine the position of a bolide given two videos of that bolide from different nodes. This poses multiple challenges such as knowing if the two nodes can see the same event (relates to guaranteeing the events

occurring at the same are of the same object, although doing the math can prove the even is not the same as long as the error is egregious), syncing the time of the event between the two nodes, figuring out the acceptable error in the calculation given the discrete nature of an image, calibration of the camera to understand where the camera is looking etc. Once you have the position of the bolide you can approximate the velocity by looking at the change in the position at different points in the event. This allows you to estimate a trajectory and thus approximate the orbital path of the bolide as it entered the atmosphere. Lastly with the light curve we can calculate the kinetic energy released by the bolide which combined with the velocity of the bolide allows us to determine the mass of the bolide which can be translated to volume given an estimated density.

- Task 3: Understanding how the node and the user control panel behaves in situations where there is no connectivity is important. Currently there is nothing in place to prevent outgoing requests from occurring with no internet connection causing the event not to be sent to the central server. In order to mitigate this we will have to implement a queue of requests that get processed by a background process that checks connectivity before each request is made. Connecting and disconnecting from the internet should also be atomic and the program should not rebroadcast the hotspot if a control panel request to connect to the wifi causes the hotspot to drop.
- Task 4: Interview researchers to understand limitations and improvements and run certain evaluation tests to determine performance. Create design diagrams and identify all features that were added. Ensure all requirements are met. Find a project banner image and make a video describing the project. Lastly, create the project e-book page.
- Task 5: The UI requires complete integration with the actual data that makes it useful. Currently there is minimal partial integration for both the database and the nodes. Both must be fully integrated such that the nodes can more or less be fully configured remotely from the frontend for whatever needs the researchers have. The database should be fully integrated to ensure researchers have the most current information possible on things like new events or node statuses. Database integration is critical to maintaining healthy node states and actually allowing the researchers to search and evaluate events.
- Task 6: We hope to catch up a bit so we can at least begin testing the client and server by targeting certain services (to make sure functionality works in general and in certain edge cases) and also just leaving the box out overnight to see if it properly grabs events that are then sent to the frontend. We also want to test to make sure that video backups are retrievable and that the researchers can modify on each node the amount of days to keep in backup.
- Task 7: Event detection must be moved to the node and it should be somewhat realtime. This poses problems due to long read times in OpenCV (of frames) causing frames to be dropped. Videos (of the entire night) must also now be backed up on the node and retrievable from the frontend, however transferring large videos could saturate the user's network so we have to test the transfers to see if they significantly impact the network. We also want to implement the ability for researchers to be able to choose the amount of days (up to 4-5 days) to keep video backups on the node.

Dates of meetings with Client during the current milestone:

see Faculty Advisor Meeting Dates below

Client feedback on the current milestone:

see Faculty Advisor Feedback below

Dates of meetings with Faculty Advisor during the current milestone:

- Jan 29, 2024
- Feb 12, 2024
- Feb 19, 2024

Faculty Advisor feedback on each task for the current Milestone:

- Task 1: Dr. Palotai especially likes the ability to quickly sort through events as they come in, the centralized UI will save researchers a lot of time.
- Task 2: Dr. Palotai agrees that it's easier to store all of the configs in the server and sync to the server rather than on each node.
- Task 3: Dr. Palotai agrees that missing events due to sending without connection should be prevented.
- Task 4: Dr. Palotai appreciates the ability to differentiate between hardware and software issues.
- Task 5: Dr. Palotai likes better quality assurance because it reduces the amount of complaints and requests for help from node stewards.

Faculty Advisor Signature: ________________________________ Date: ________