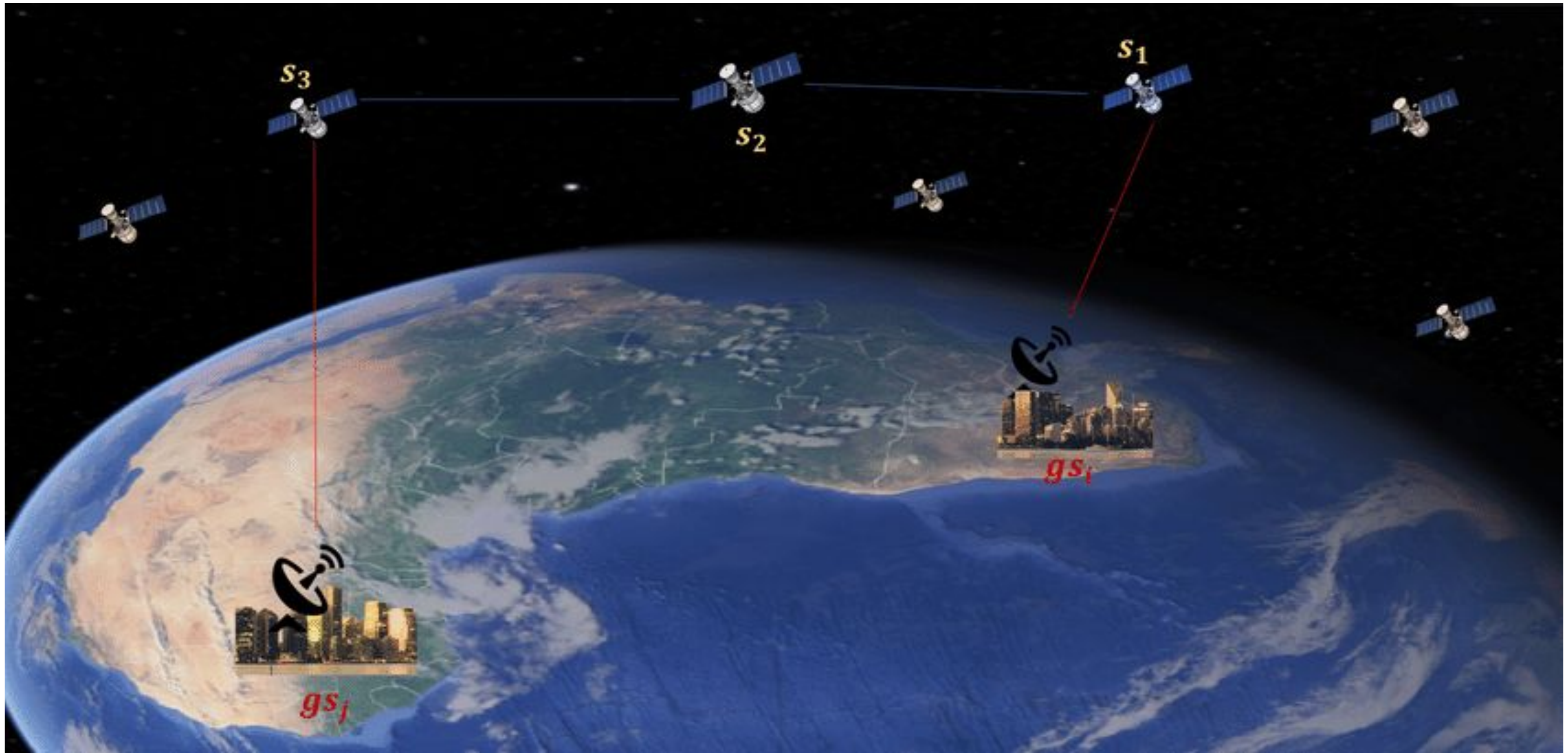
SATOPSMQ

GitHub

By: Hashir Jamil, Rafael Dolores, Walid Al Dari, Ruth Bezabeh, Stanley Ihesiulo, James Le, Youssef Hany

Supervisor: Dr. Regina S.K. Lee, PhD, P. Eng.

Satellite/Groundstation Communication



PROJECT OVERVIEW

Problem: Lack of open-source solutions that can process complex orders into a schedule that demonstrates satellite and ground station communications.

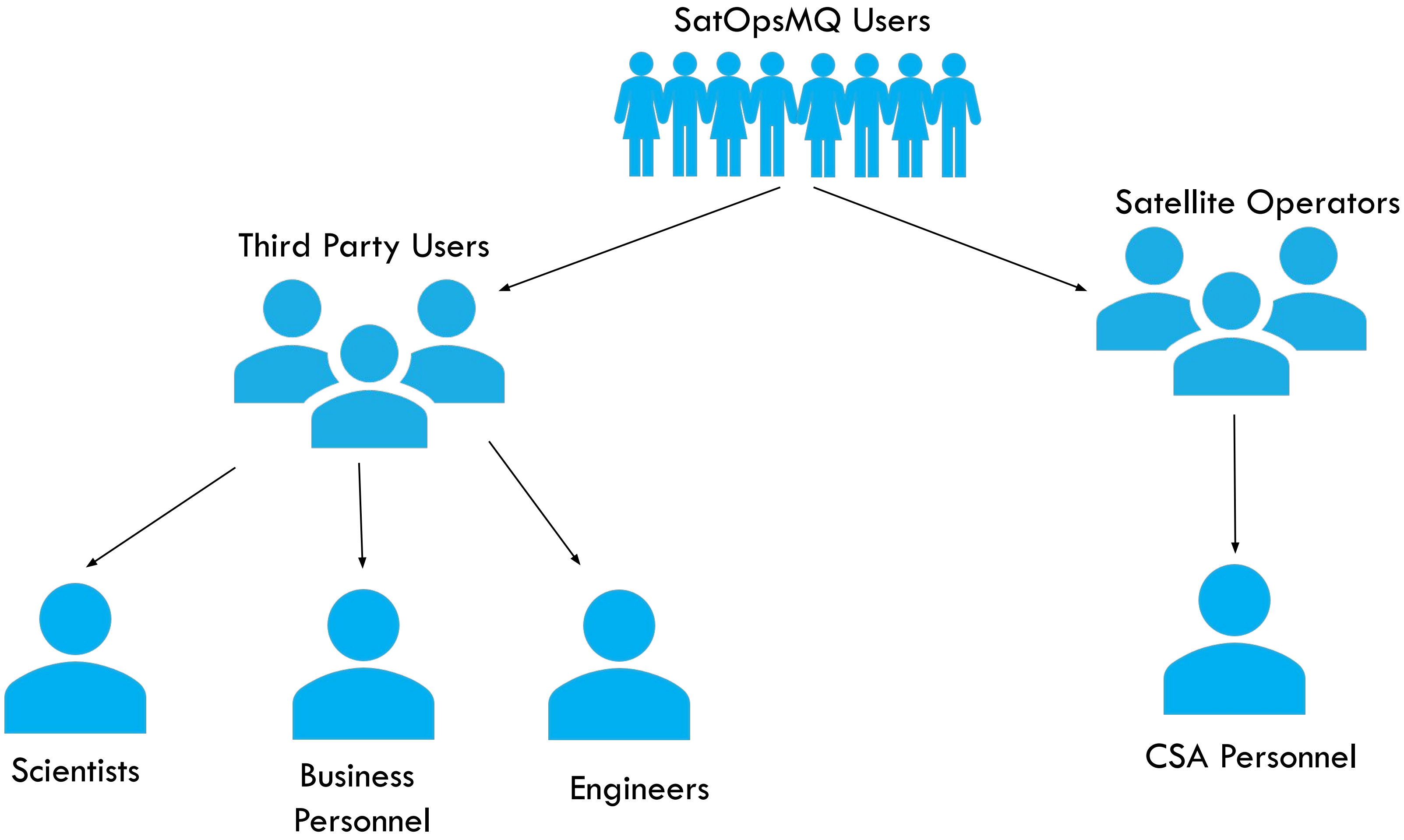
Solution: Canadian Space Agency sponsored project that aims to develop an open-source scalable web-app that can simulate mock ground station and satellite communications to process image, maintenance, and outage orders in a schedule format that is easy to understand.

Event-Driven,
Microservices
Architecture

Open-source
&
Explainable

Lightweight &
Scalable

SYSTEM USER HIERARCHY



SYSTEM INPUTS

Image Order

```
{
  "Latitude": -85.6439118846981,
  "Longitude": -52.678023392885535,
  "Priority": 1,
  "ImageType": "Low",
  "ImageStartTime": "2023-11-18T02:10:15",
  "ImageEndTime": "2023-11-18T07:18:07",
  "DeliveryTime": "2023-11-18T12:18:07",
  "Recurrence": { …  }
}
```

Maintenance Order

```
{
  "Target": "SOSO-1",
  "Activity": "OrbitManeuver",
  "Window": {
    "Start": "2023-10-08T06:26:47",
    "End": "2023-10-08T06:41:47"
  },
  "Duration": "900",
  "RepeatCycle": { … }
}
```

Outage Order

```
{
  "Target": "GroundStation-1",
  "Activity": "Outage",
  "Window": {
    "Start": "2023-10-08T06:26:47",
    "End": "2023-10-08T06:41:47"
  }
```

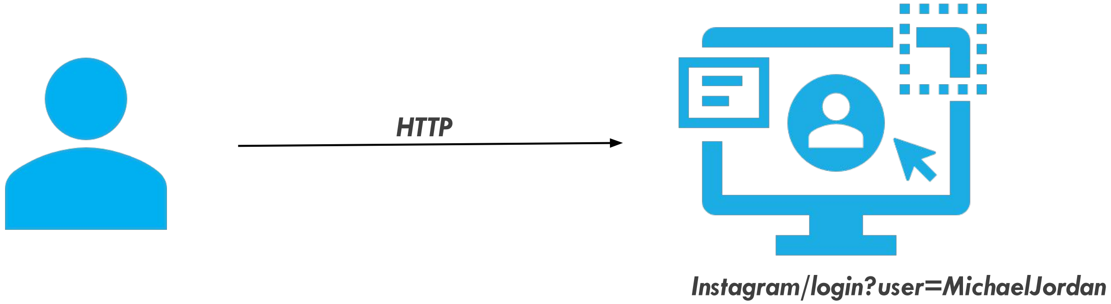
DEMO

Implementation

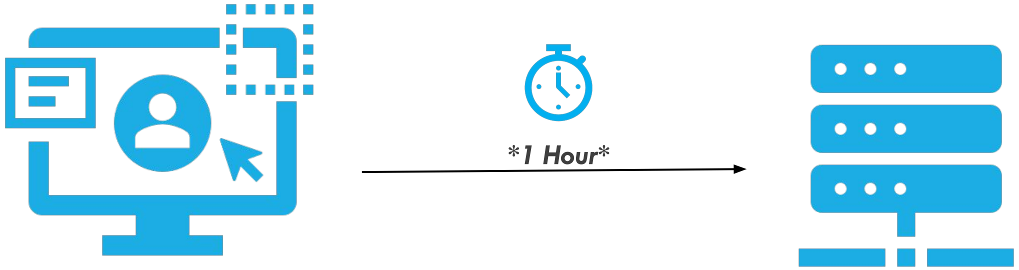
- Architecture
- Optimization Goals
- Scheduler Implementations
- Future Imrpovements

SYSTEM DEFINITIONS

API Call: a request made to a URL linked to a function in our software



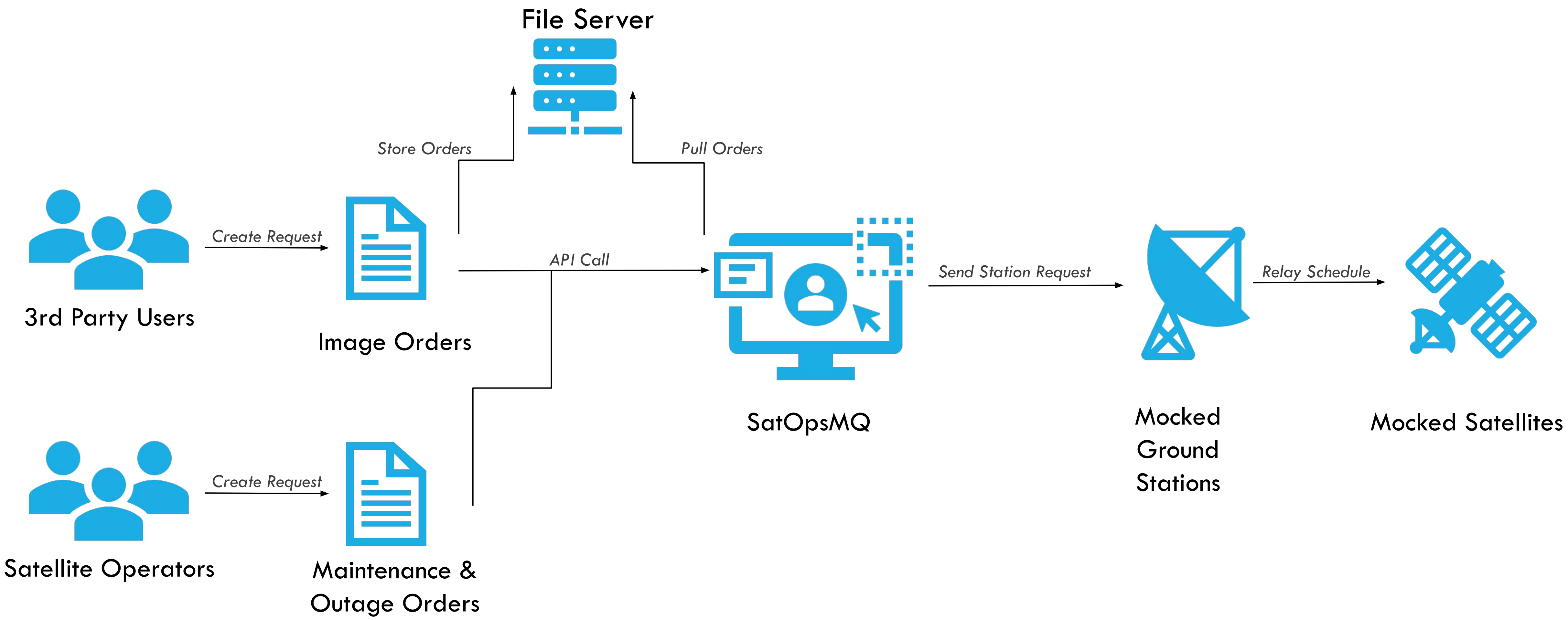
Pull Order: downloading image orders periodically from a remote server



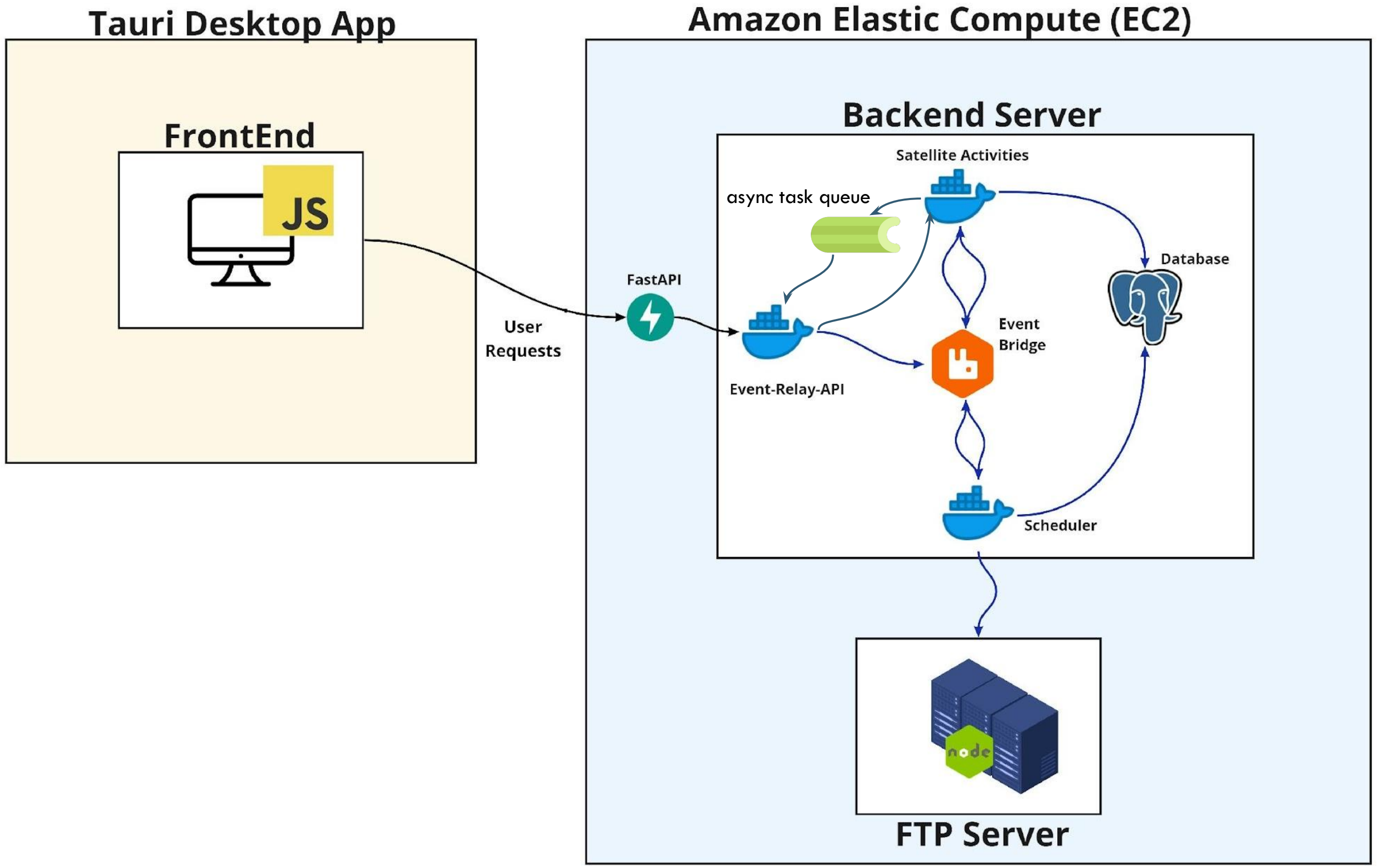
Ground Station: a radio telescope that sends & receives data from a satellite

SYSTEM OVERVIEW

Ground Stations and Satellites are APIs



ARCHITECTURE



Schedule Optimization Goals

- High Throughput
- High Resource Utilization
- Balanced Workload distribution

How we calculate a numeric score for how well a schedule is meeting these goals: Schedule Performance Metrics

SCHEDULING APPROACHES

Currently implemented approaches

- Knapsack Problem
- Heuristic Candidate Elimination

FUTURE IMPROVEMENTS

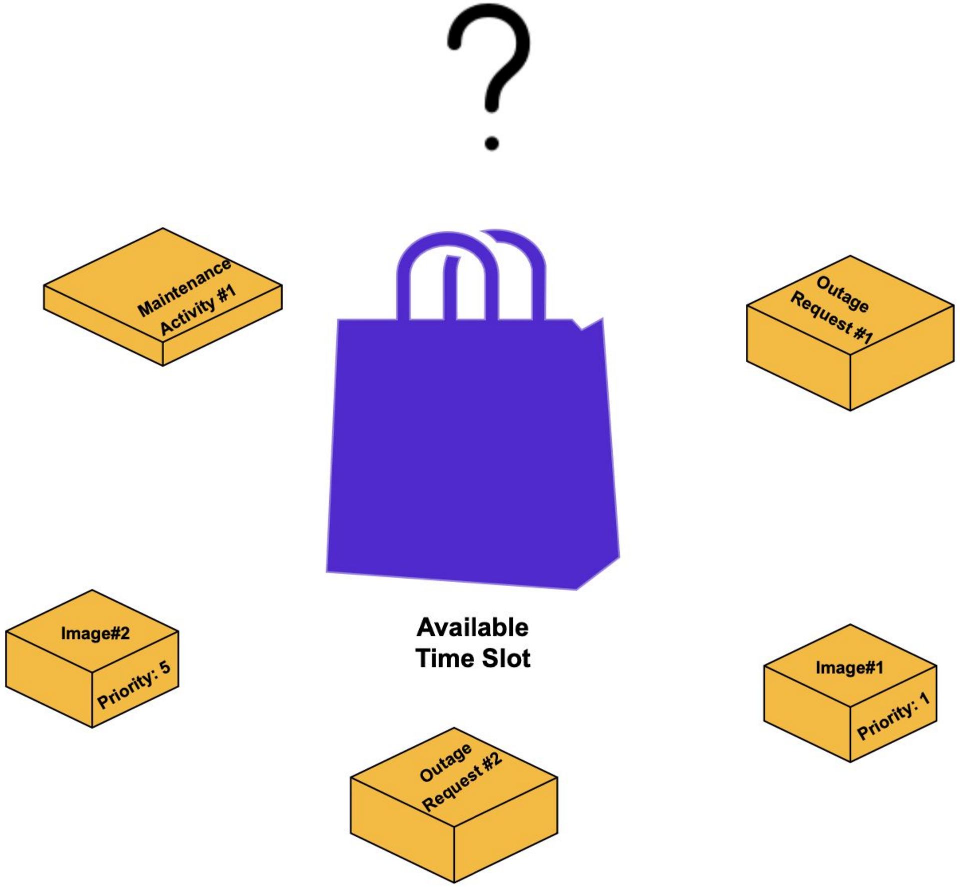
- Genetic Algorithm (Pending Completion)
- Particle Swarm Optimization (In the conceptual phase)
- Supporting Concurrent Scheduling (Pending Completion)

SCHEDULING APPROACHES - KNAPSACK

- The idea of the Knapsack problem is to determine the optimal subset of items to pack into a knapsack, such that the total weight does not exceed the knapsack’s capacity while maximizing the total value of the packed items

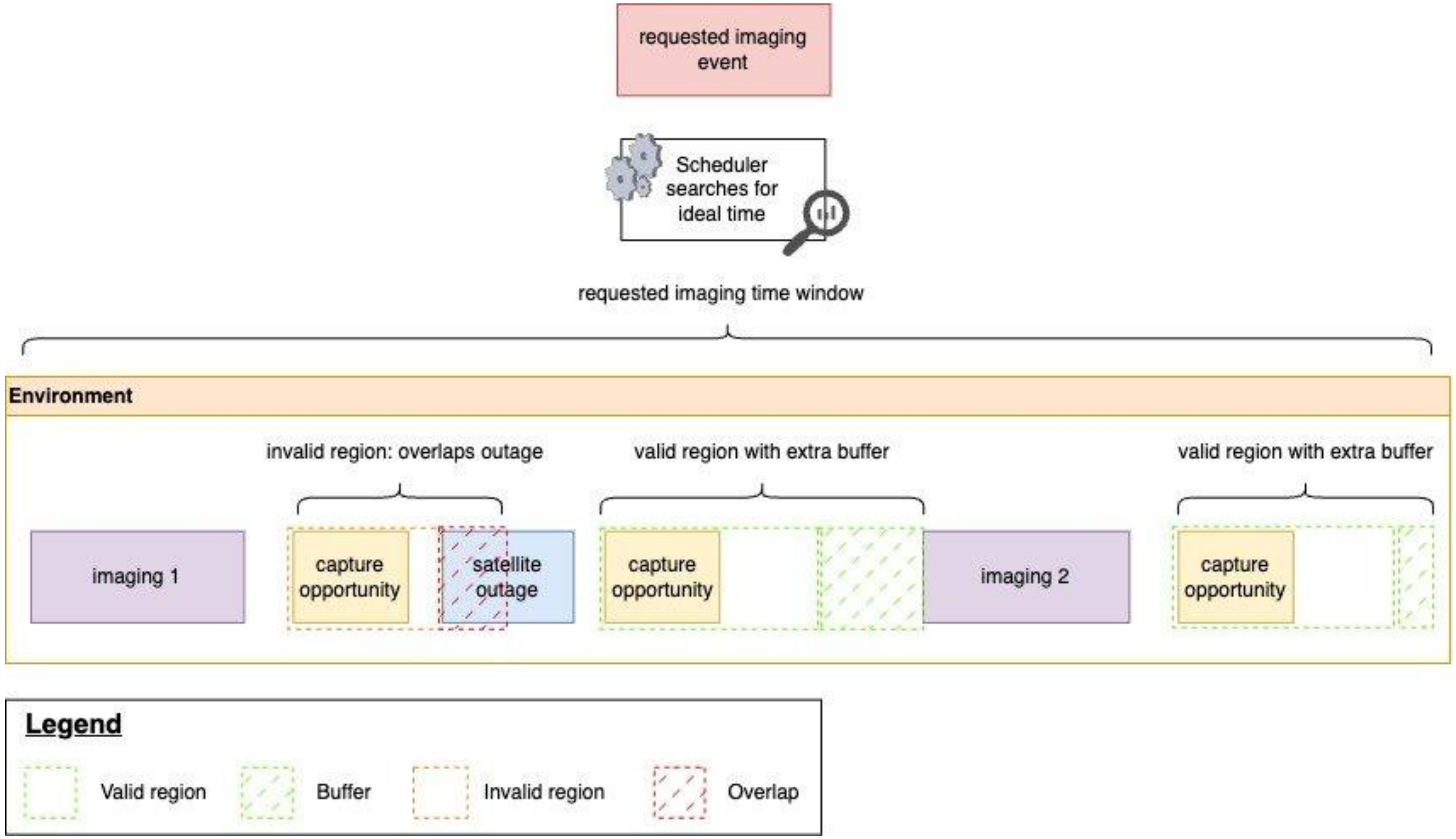
- In SatOpsMQ, the "knapsack" is the available time slots and bandwidth on the satellite or ground stations.

- Each satellite operation (imaging request, maintenance task, outage response) represents an "item" to be placed in the knapsack. These tasks have "weights" (resource requirements like time and bandwidth) and "values" (priority or importance of the task).



SCHEDULING APPROACHES – HEURISTIC CANDIDATE ELIMINATION

- Events stored in database.
- Valid gaps in time between events retrieved – candidate schedule slots
- Candidate schedule slots paired with transmission events to form candidate schedule plans
- Candidate schedule plans eliminated based on if it results in an invalid satellite state (e.g. uses more storage/power than available)
- Schedule plan selected using a uniform random distribution across candidates, for good workload distribution across satellites/ground-stations



FUTURE IMPROVEMENTS

- Genetic Algorithm (Pending Completion)
- Particle Swarm Optimization (In the conceptual phase)
- Supporting Concurrent Scheduling (Pending Completion)

FUTURE IMPROVEMENTS – GENETIC ALGORITHM

- Candidates generated from the heuristic approach are used to create a population of schedules (Pending Completion)
- Each of these schedules can be scored using modular, extensible, performance metrics. We developed 3 performance metrics modules so far that, for a given schedule, scores its: Resource utilization, Throughput, and Workload Distribution (Completed)
    - More information on how these performance metrics are calculated can be found here.
- Create mutation modules – modular and extensible components that mutate a schedule in a controlled way, given a mutation strength/temperature: for example, a mutation module that shifts the scheduled time of events, or one that swaps downlink contact ground-stations, e.t.c. (Uncompleted)
- Mutate an initial schedule population, select top-scoring schedules, and repeat for a given number of generations, until some stopping criteria is reached (Uncompleted)

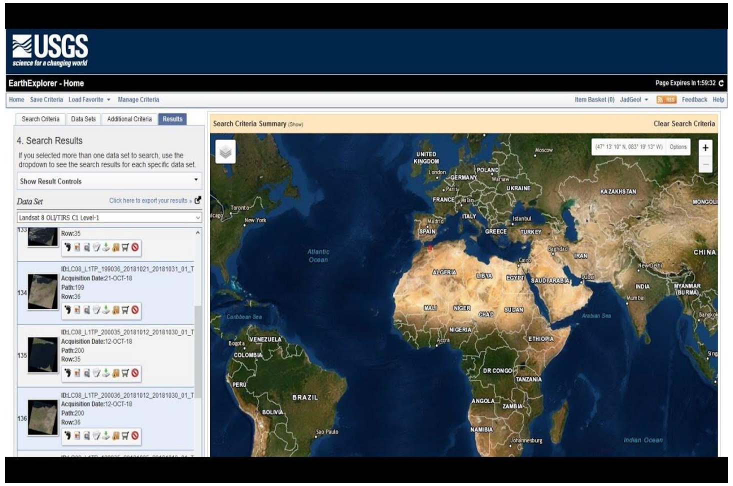
FUTURE IMPROVEMENTS – PARTICLE SWARM OPTIMIZATION (CONCEPTUAL PHASE)

- This approach builds on the mutations of the genetic algorithm approach. The position of particles are encoded as a vector containing the temperature/strength of the mutation modules in use
- The velocity vector defines how much the temperature/strength of the mutations change
- Performance score of schedule defines the optimization space
- Particle swarm optimization algorithm is performed to find a schedule optima

FUTURE IMPROVEMENTS – CONCURRENCY

- The system was built with concurrency in mind. However, a significant amount of work remains to support concurrently processing schedule requests.
- Concurrency support would allow for horizontal scalability of the scheduler system, allowing it to support a much higher order scheduling rate, and improving its throughput.

COMPETITORS IN THE MARKET


USGS
science for a changing world
EarthExplorer - Home
Home Save Criteria Load Favorite Manage Criteria
Search Criteria Data Sets Additional Criteria Results
Search Criteria Summary
Clear Search Criteria
Show Result Controls
Click here to export your results »

4. Search Results

If you selected more than one data set to search, use the dropdown to see the search results for each specific data set.

Data Set: Landsat 8 OLI/TIRS C1 Level-1

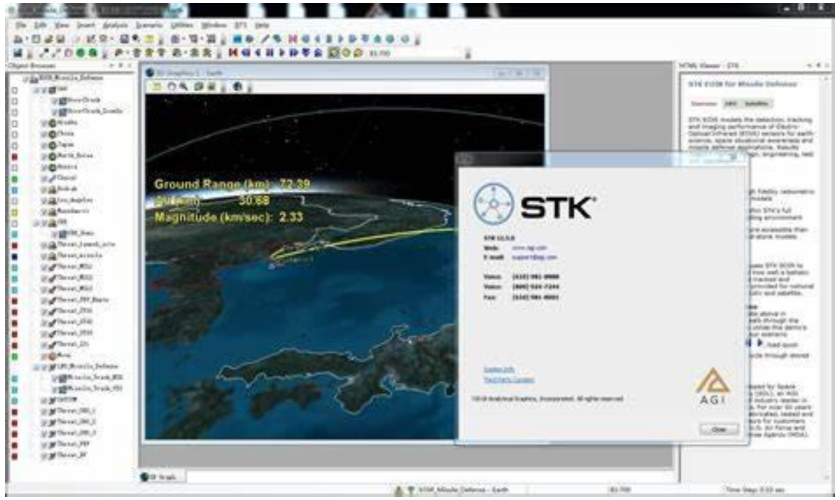
133 Row:35

134 ID:LC08_L1TP_199036_20181021_20181031_01_T
Acquisition Date:21-OCT-18
Path:199
Row:36

135 ID:LC08_L1TP_200035_20181012_20181030_01_T
Acquisition Date:12-OCT-18
Path:200
Row:35

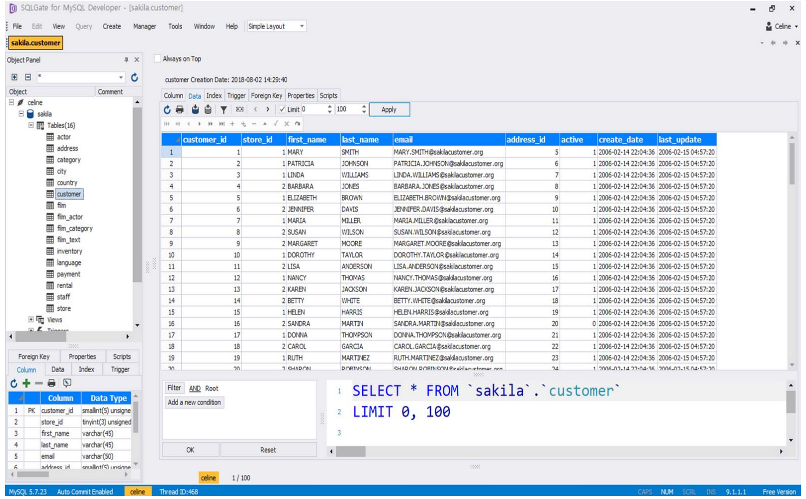
136 ID:LC08_L1TP_200036_20181012_20181030_01_T
Acquisition Date:12-OCT-18
Path:200
Row:36

Proprietary Software



Ground Range (km): 72.39
[illegible] 30.88
Magnitude (km/sec): 2.33

Simulations



customer Creation Date: 2018-08-02 14:29:40

☑ Limit 0 100

	customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
1	1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	1	2006-02-14 22:04:36	2006-02-15 04:57:20
2	2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	1	2006-02-14 22:04:36	2006-02-15 04:57:20
3	3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	1	2006-02-14 22:04:36	2006-02-15 04:57:20
4	4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	1	2006-02-14 22:04:36	2006-02-15 04:57:20
5	5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	1	2006-02-14 22:04:36	2006-02-15 04:57:20
6	6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	1	2006-02-14 22:04:36	2006-02-15 04:57:20
7	7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	1	2006-02-14 22:04:36	2006-02-15 04:57:20
8	8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer.org	12	1	2006-02-14 22:04:36	2006-02-15 04:57:20
9	9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer.org	13	1	2006-02-14 22:04:36	2006-02-15 04:57:20
10	10	1	DOROTHY	TAYLOR	DOROTHY.TAYLOR@sakilacustomer.org	14	1	2006-02-14 22:04:36	2006-02-15 04:57:20
11	11	2	LISA	ANDERSON	LISA.ANDERSON@sakilacustomer.org	15	1	2006-02-14 22:04:36	2006-02-15 04:57:20
12	12	1	NANCY	THOMAS	NANCY.THOMAS@sakilacustomer.org	16	1	2006-02-14 22:04:36	2006-02-15 04:57:20
13	13	2	KAREN	JACKSON	KAREN.JACKSON@sakilacustomer.org	17	1	2006-02-14 22:04:36	2006-02-15 04:57:20
14	14	2	BETTY	WHITE	BETTY.WHITE@sakilacustomer.org	18	1	2006-02-14 22:04:36	2006-02-15 04:57:20
15	15	1	HELEN	HARRIS	HELEN.HARRIS@sakilacustomer.org	19	1	2006-02-14 22:04:36	2006-02-15 04:57:20
16	16	2	SANDRA	MARTIN	SANDRA.MARTIN@sakilacustomer.org	20	0	2006-02-14 22:04:36	2006-02-15 04:57:20
17	17	1	DONNA	THOMPSON	DONNA.THOMPSON@sakilacustomer.org	21	1	2006-02-14 22:04:36	2006-02-15 04:57:20
18	18	2	CAROL	GARCIA	CAROL.GARCIA@sakilacustomer.org	22	1	2006-02-14 22:04:36	2006-02-15 04:57:20
19	19	1	RUTH	MARTINEZ	RUTH.MARTINEZ@sakilacustomer.org	23	1	2006-02-14 22:04:36	2006-02-15 04:57:20
20	20	2	SHARON	ROBINSON	SHARON.ROBINSON@sakilacustomer.org	24	1	2006-02-14 22:04:36	2006-02-15 04:57:20

```
SELECT * FROM `sakila`.`customer`
LIMIT 0, 100
```

Manual Operation

UNIQUE SELLING POINTS

1. Fully open-source system.
2. Transparency and explainability of optimization results is prioritized.
3. Allows for improvements to system by community, transparent change logs and issue tracking.
4. Can be used as an educational/tutorial for modern software engineering in the space systems domain.

ALPHA RELEASE TESTING OVERVIEW

CI/CD PIPELINE

UNIT TESTING

INTEGRATION TESTING

END-TO-END TESTING

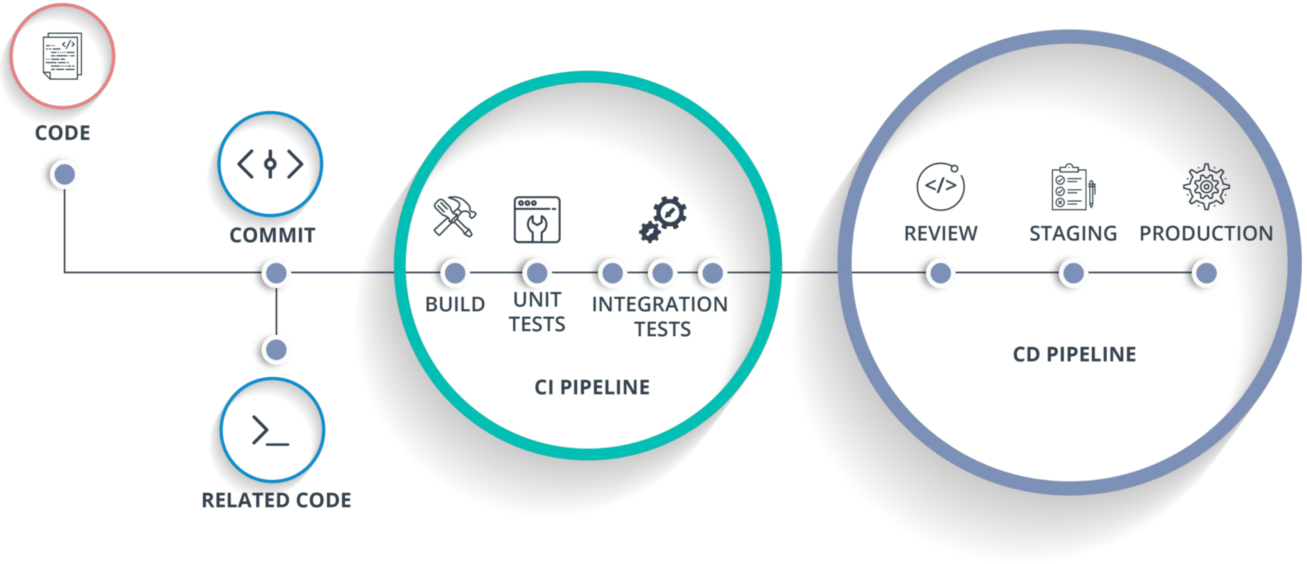
TESTING CHALLENGES

CI/CD PIPELINE

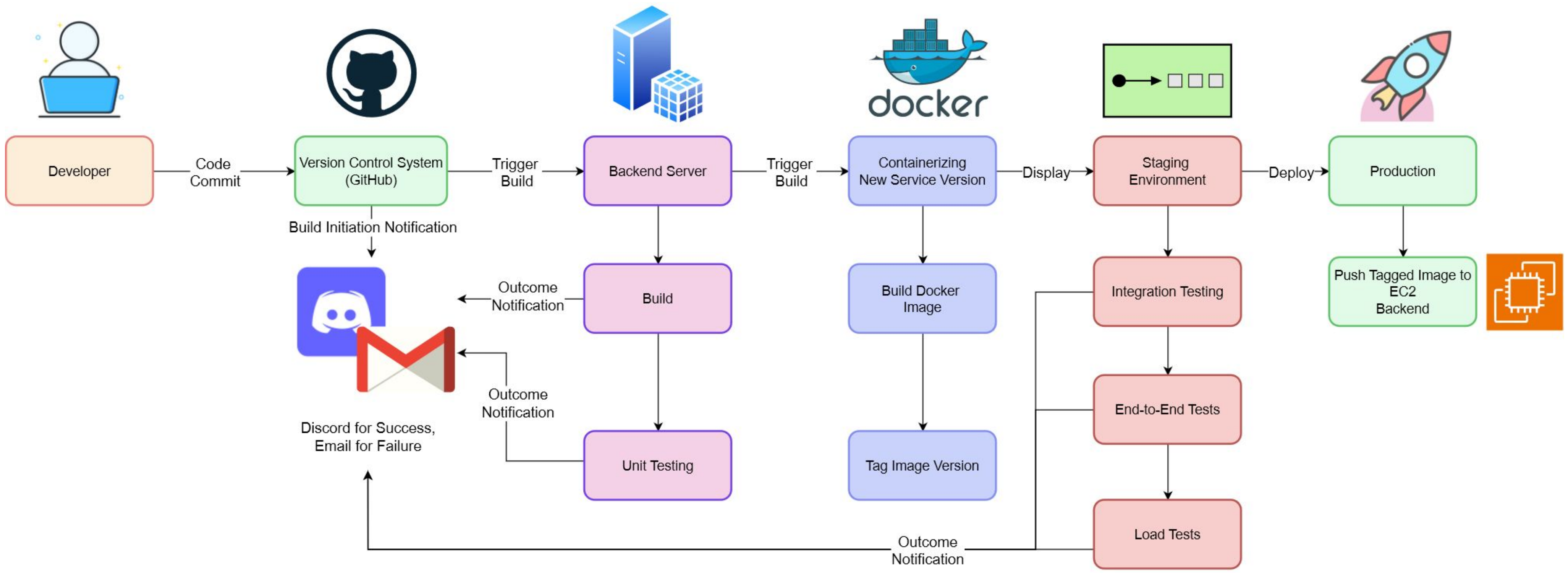
Definition: CI/CD is a practice that automates the process of integrating code changes and deploying applications.

Importance: Accelerates development cycles, improves code quality, and ensures reliable deployments.

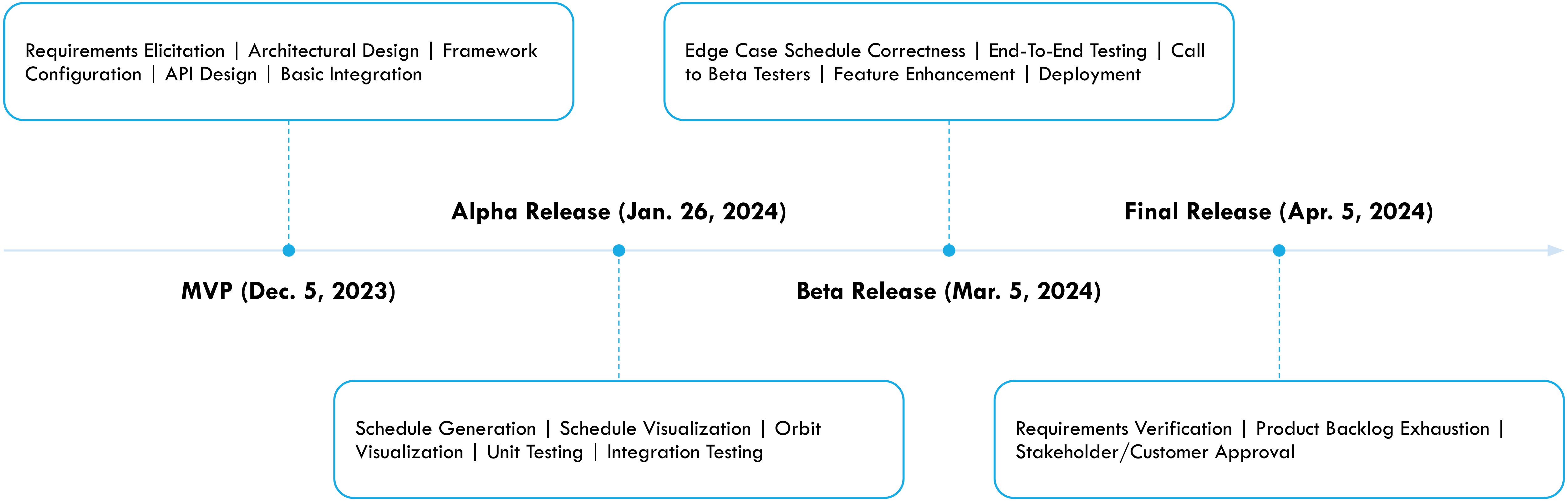
GitHub Actions: A platform for automating workflows directly within the GitHub repository.



OUR PIPELINE



PATH TO RELEASE



Requirements Elicitation | Architectural Design | Framework Configuration | API Design | Basic Integration

MVP (Dec. 5, 2023)

Alpha Release (Jan. 26, 2024)

Schedule Generation | Schedule Visualization | Orbit Visualization | Unit Testing | Integration Testing

Edge Case Schedule Correctness | End-To-End Testing | Call to Beta Testers | Feature Enhancement | Deployment

Beta Release (Mar. 5, 2024)

Final Release (Apr. 5, 2024)

Requirements Verification | Product Backlog Exhaustion | Stakeholder/Customer Approval

Code

https://github.com/Satellite-Operations-Services-Optimizer

The code is completely open-source, so you can do whatever at all you want with it :)

THANK YOU FOR LISTENING!

What are your questions and feedback?

We appreciate all input!