18th International Conference on Space Operations, Montreal, Canada, 26 - 30 May 2025.

SpaceOps 2025, ID # 189

MARS RELAY OPERATIONS SERVICE (MaROS): ADAPTING TO THE NEEDS OF THE MARS RELAY NETWORK

Brandon Sauer

Jet Propulsion Laboratory, California Institute of Technology, USA, brandon.sauer@jpl.nasa.gov

Abstract

The Mars Relay Operations Service (MaROS) is a centralized ground system service managed and maintained by the Jet Propulsion Laboratory (JPL) that is used for the planning and coordination of relay telecommunications between spacecraft at Mars. This paper will discuss its functions and capabilities and describe its expanding list of subsystems and related services that have enhanced the relay operations process.

One fundamental change to the service was to abstract the responsibility of calculating view periods from predicted ephemerides away from the mission operators, and to instead rely on a new centralized subsystem called MPX (Metric Predicts Executive). Another substantial enhancement to the service came with the introduction of the Relay Telecom Predictor (RTP), a service that allows mission operators to compute data volume predictions for relay passes using mission-defined models, then feed those predictions back into MaROS to facilitate the selection of mission-optimal passes. Finally, a new subsystem, called "Where's My Relay Data", aims to help provide visibility to the relay downlink dataflow by aggregating monitoring data from various checkpoints in the downlink pipeline and displaying them to mission operators in a dynamic and customizable dashboard.

This paper will describe the latest capabilities as MaROS is continuously being positioned to manage the many challenges of the Mars Relay Network as it expands and evolve.

---

1. Introduction

Introduced in 2010, the Mars Relay Operations Service (MaROS) was created as a joint effort between the Mars Exploration Program (MEP) and the Multimission Ground System Services (MGSS) Program at the Jet Propulsion Laboratory (JPL) to bring structure and standardization to the otherwise disjointed and varied Mars Relay Network (MRN) [1, 2]. MaROS has been utilized by nearly a dozen spacecraft over the course of its existence, and while the key functions accomplished by its original design have remained at the forefront of its role in relay coordination, a plethora of repairs and enhancements have been implemented to support new technology and use cases presented by missions entering the network. To this point, over 8 Terabits of data have been returned from the surface of Mars over 35,000 relay "overflights" between landers and orbiters at Mars. Modern landers, such as the Mars Science Laboratory (MSL, Curiosity) rover [3] and Mars 2020 (Perseverance) rover [4], depend on relay for over 99% of the data they return to Earth, all of which is scheduled and coordinated through MaROS.

MaROS is a ground system software service that is managed, hosted, and maintained by JPL which allows mission operators of relay service users (typically lander missions) and relay service providers (typically orbiter missions) to negotiate the planning of relay communication at Mars. As a brief overview, MaROS supports the following high-level functionality:

- The strategic and tactical planning and negotiation of parameters that define how the spacecraft and radios involved should be commanded for each requested relay pass. MaROS also assists in the generation of scheduling and validation conflicts, which are raised if a relay request violates any number of constraints.

- The handling and transfer of data products between mission operators in both the forward-link (data transferred from Earth, through the orbiter, to the lander) and return-link (data transferred from the lander, through the orbiter, to Earth) directions. MaROS also calculates the predicted latencies in both directions to inform lander operators when they need to provide data to the orbiter operators via MaROS in the forward-link case, or when they can expect to receive their data from a relay pass in the return-link case.

- The monitoring and accountability of post-pass performance of relay overflights coordinated through the system via the delivery of various telemetry reports provided by both the relay service users and providers after the pass has occurred.

The MaROS system is focused around a centralized, relational database (SQL) that stores geometric relay view period information and planning data for relay overflights input by both the lander and orbiter mission operators. In addition to planning data directly related to overflights, the database also contains various other data points provided by mission operators that allow MaROS to perform the previously described functions, such as orbiter Earth-downlink window events, mission configuration settings, file transfer metadata, post-pass performance data, and much more. Interaction with the database is controlled via a Java-based web server that contains all of the business logic for extracting and accepting data. The web server provides an abundance of Representational State Transfer Application Programming Interfaces (ReST APIs) which are utilized by the MaROS Graphical User Interface (GUI) web application and the Python-based Command Line Interface (CLI) client (Figure 1). The GUI and the CLI are the main forms by which authorized users access and interact with MaROS. The CLI provides a great deal of extensibility and enables operators to integrate MaROS interactivity into their automated processes or other operations ground systems. The GUI, meanwhile, supplies a diverse set of more interactive and visual experiences for users to view and manipulate MaROS data.

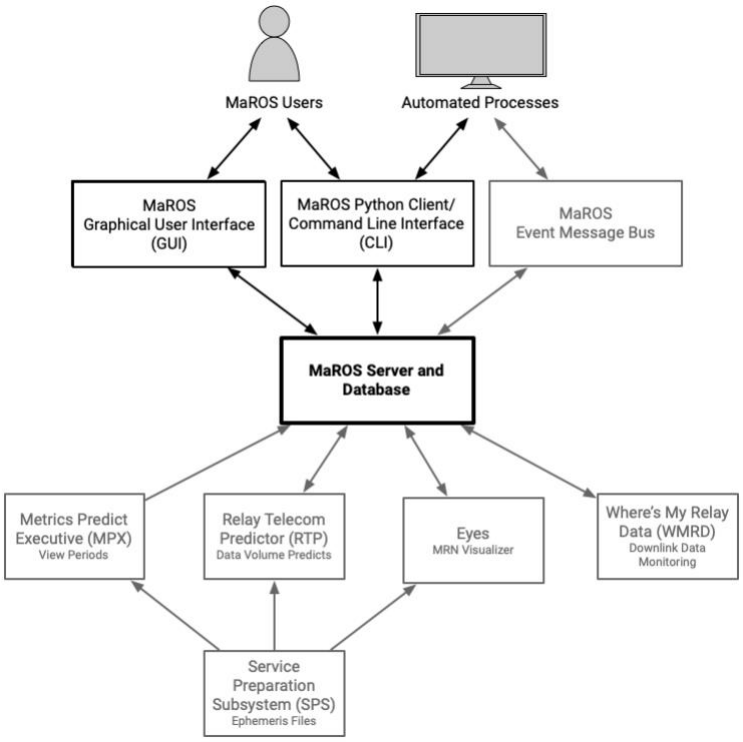

Figure 1: The MaROS "Ecosystem" of software services and tools

With over a decade of operational experience, and more lessons learned throughout the existence of the MRN [5], nearly every aspect of the MaROS system has required changes and additions of varying degrees to meet the needs of the MRN community.

Disclaimer—Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

2. Capability Updates

Since the last publication explicitly regarding MaROS functionality in 2014 [6], hundreds of software changes, small and large, have been incorporated into the service by the MaROS software team. Many of these changes repair bugs and discrepancies discovered in the system over time, and many more were identified collaboratively by users,

sponsors, and the development team to best support the needs of MaROS users. Described below are just some of the many significant enhancements that have been integrated into MaROS over the past decade, in roughly chronological order (excluding new subsystems, which are discussed in detail in further sections).

Generic Events—MaROS had originally supported the publications of only a discreet list of relay-related events types, such as orbiter downlink windows and relay keep-out zones, that are used internally for relay planning logic. Because this list of valid event types was finite, however, if a new event type needed to be communicated to the system that might affect relay coordination, changes to the MaROS codebase needed to be made to support these. The introduction of Generic Events allows for more flexible changes to the system, such as indication of spacecraft attitude durations by the MAVEN orbiter for relay support or UHF radiometric experiments between ESA's MEX and TGO orbiters. Such events are not directly related to the planning of overflights, but may have effects on relay coordination with other relay service users.

General File Transfer—In the same vein of General Events, the need for General File Transfer (GFT) was also added to support cases where operators of missions in the relay network may need to transfer files to one another that may have impacts on relay coordination. Such files are typically not binary files to be transferred via relay services between the spacecraft, but biproducts of relay operations such as doppler, attitude, or timing data.

Overflight Timeline—An overflight timeline visualizer was added to the MaROS GUI Overflight Search page to provide a visual representation of overflight opportunities and requests, as well as latency predictions associated with each overflight. The timeline also allows the user to switch the x-axis to Sols and Local Mean Solar Time (LMST) for each mission, display view period elevation as a gradient, select overflights to perform further actions, and more (Figure 2). Going forward, there is intent to add more events, such as the Generic Events and Orbiter Events, to the timeline in relation to overflights, as these are currently not visible elsewhere on the MaROS GUI.

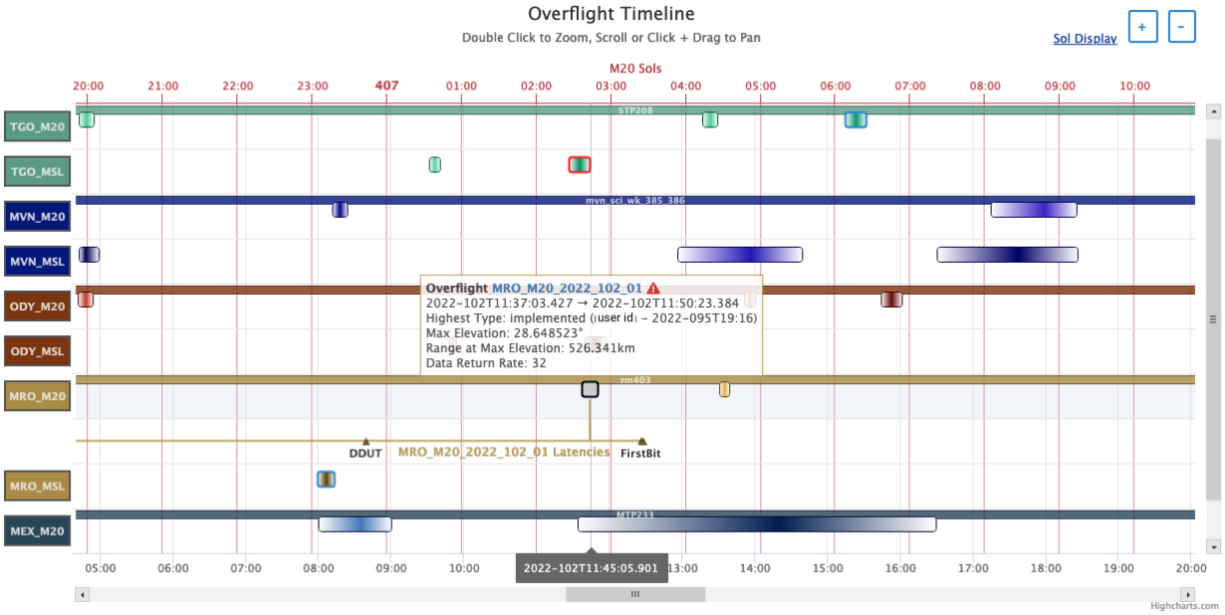

Figure 2: A sample screenshot of the Overflight Timeline from the updated MaROS GUI.

File Transfer GUI—After multiple mishaps occurred when submitting Forward Link files (files to be submitted from the orbiter to the lander) through the MaROS GUI, a replacement for the confusing interface was deemed necessary. Instead of uploading files, sending files to orbiter operators, and viewing previously sent files through the same page, the new design breaks these functionalities out into their own discreet interfaces. A "File Management" page lets mission operators upload, view, and delete Forward Link files for their mission's repository. The Forward Link Home page provides a clear view of all Forward Links initiated through MaROS, with up-to-date status information on the

progress of each file in its journey to the relay service user's ground system. Finally, the Forward Link Builder was converted to a multi-step "wizard" tool that walks mission operators through the process of selecting which files to transfer and when to transfer them (Figure 3). The interfaces for managing, sending, and receiving General File Transfers were implemented using similar design patterns for a cohesive user experience.

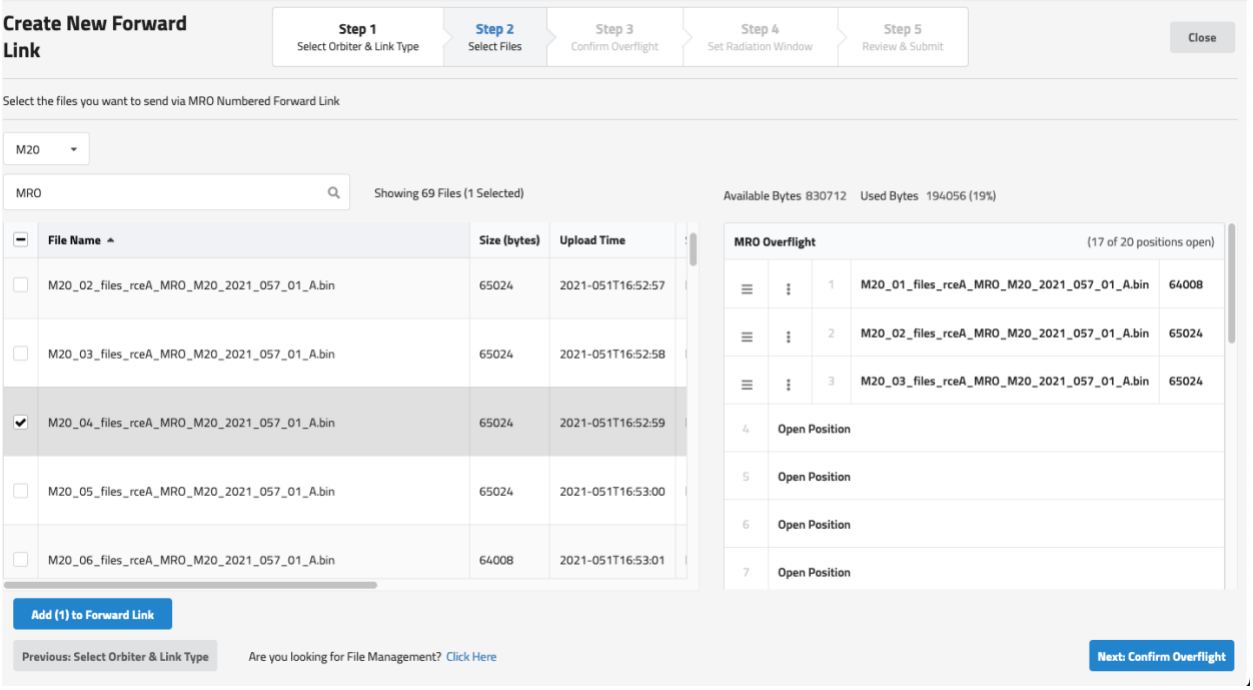

Figure 3: The Forward Link Builder wizard from the updated MaROS GUI.

Split/Shared Relay—When the InSight lander joined the relay network at Mars, its geographic proximity to the MSL rover required the two missions to sometimes coordinate splitting their relay opportunities. For example, if MRO was in view of both landers at roughly the same time, MRO could spend the first half of that view period communicating with MSL, and the second half with InSight. In support of this new requirement, MaROS introduced the determination of "Shared Relay Candidates", which indicates when two (or more) landers' view period geometry with a single orbiter overlaps. When the landers both request to perform relay over a pair of Candidates, they become "Shared Relay Partners" and are validated together along with a new Boolean "SHARED_RELAY" overflight parameter to ensure there is enough time between their scheduled comm windows for the orbiter to handle both links.

Cloud Migration—The MaROS database and server were originally hosted on machines at JPL. With the rise of adoption of Amazon Web Services (AWS) via GovCloud at JPL and many other services that had previously shared the costs of on-premise hosting migrating to the cloud, the idea of making the jump to AWS for MaROS became more appealing. Not only was it becoming more cost-effective and widely supported at JPL, but AWS offered a vast suite of ready-to-use services that could help to enable functionality previously out of reach for MaROS. After research, experimentation, and an eventual transition over to GovCloud in 2019, MaROS is now able to take advantage of AWS services such as Simple Notification Service (SNS) and Simple Queue Service (SQS) for an event driven message distribution system, able to be utilized both internally within MaROS and externally by users to allow for greater automation and extensibility potential. While JPL requires administrator permission for some AWS functionality for security reasons, self-service access to cloud resources greatly improves the flexibility of the MaROS development team to more quickly debug and resolve operational issues as opposed to dependency on separate System Administrators. As will be discussed throughout this paper, many of the new features and subsystems in MaROS are enabled by this migration to AWS. Outside of the new services MaROS can take advantage of, AWS also provides the benefit of higher reliability and availability. Though infrequent, MaROS is now able to remain functional even

when availability on-premise at JPL is compromised, such as during the Eaton fire in January 2025 when MaROS allowed MSL and Mars 2020 to send forward-link data to ESA orbiters despite JPL infrastructure outages [7].

Advanced Overflight Search—In MaROS, each mission defines their own custom overflight parameters that are negotiated between the relay service user and provider to determine the nature of a given relay link. The dynamic nature of these parameters, however, creates a challenge when it comes to building a comprehensive search interface for MaROS users to perform analysis of overflight planning data across the system. An Advanced Overflight Search feature was designed and implemented in MaROS, with the ability to select parameters from each mission and filter overflights by the values for each of these individual parameters. Due to the complex structure of MaROS's relational database and the sheer quantity of overflight data in MaROS, the resulting implementation relied on the introduction of a GraphQL database [8]. The GraphQL database acts like a cache of the SQL database, with all of the complex connections between SQL tables consolidated into a single JSON structure linked via a GraphQL schema. These structures, combined with the efficiency and flexibility of the GraphQL query language, allows the Advanced Overflight Search GUI to quickly perform highly dynamic queries over tens of thousands of MaROS overflights.

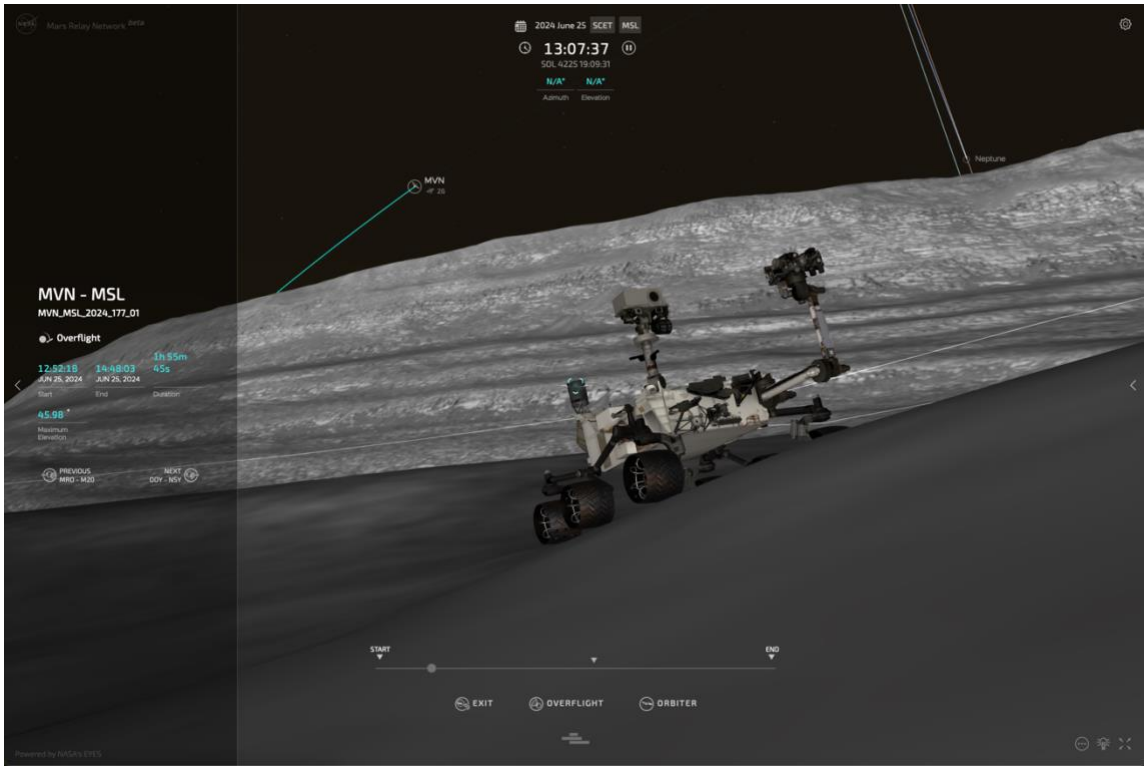

Figure 4: A screenshot of a MAVEN/MSL overflight from the lander's perspective in the Eyes on the MRN Visualizer

Integrating with "Eyes on the MRN"—Originally developed as an outreach tool that uses MaROS overflight data, the conception of the publicly available "Eyes on the MRN" 3D Visualizer [9] also became a helpful way for mission operators to visualize their spacecrafts' predicted and past interactions with others in the network. The tool was embedded into MaROS for users to quickly visualize an overflight in context with their planning parameters. The Eyes on the MRN tool is capable of displaying the overflight geometry from space or from the orbiter's or lander's perspective directly. An especially helpful capability was added in 2023 to support the visualization of local terrain when viewing relay from the lander's perspective (Figure 4), allowing operators to visually determine if a terrain occlusion may have had an impact on relay execution.

FDMA Support—Future Mars missions expect to introduce a new relay operations concept not yet supported by MaROS in the form of Frequency Division Multiple Access (FDMA). In short, FDMA will allow a single orbiter to perform relay with two landers simultaneously on different radio frequencies [10]. Similar to the inclusion of split relay when the InSight lander joined the network, support for FDMA will also require changes to the MaROS system in order to prevent scheduling conflicts from intentionally overlapping overflights when they occur on separate radio channels for FDMA-supporting orbiters. Work to support this feature was released for initial testing of FDMA use cases in MaROS in 2024, enabling FDMA-capable missions to be supported in the system at any time.

3. User Interface and Design

When introduced, the MaROS GUI took the form of a web application built using Adobe Flash. Naturally, as with nearly all web applications, software and technology advancements illicit modernizations and other maintenance efforts to sustain a useful tool. MaROS is no different, and in 2016 it was deemed that an upgrade was needed.

With the main goal of porting over and, when reasonable, modernizing existing functionality from the Flash GUI, a new web app was developed using primarily JavaScript (JS), JQuery, and PHP. While many of the ReST APIs that supplied the Flash GUI with connection to the server remained in use for the new GUI, others were re-designed and implemented with more modernized and standardized structures to better fit the needs of the new GUI. In 2019, an effort to further modernize the GUI began with the introduction of React.js [11], a highly popular Javascript framework that levies its component-based paradigm to unify implementation architecture and improve reusability of common components. As of this writing, about fifty percent of the MaROS GUI has been redesigned and refactored utilizing this React paradigm – which integrates relatively smoothly with existing JS/PHP pages – with all future GUI development planned to continue in React. The official deprecation of Flash support in all major web browsers in 2020 forced the deprecation of the MaROS Flash GUI, and hence forth, all GUI activity takes place on the new Javascript and React web application and all major features from the Flash GUI have been successfully transferred to the new GUI.

Also beginning in 2019, a User Experience/User Interface (UX/UI) Designer joined the MaROS development team on a consistent basis. Prior to this, GUI features for both the Flash and the initial JS/PHP GUI started with a description or rough drawing of the desired functionality by the product owner based on their understanding of the users' needs, followed by software developers building the feature, starting a cycle of varying degrees of iteration and rework before settling on a product that was satisfactory enough. Though the development team did meet with MaROS power users and stakeholders on a frequent basis, incorporating feedback from these meetings into a final product that met everyone's expectations was an arduous task lacking a formal process. With the addition of a dedicated designer to the team came a more structured process for streamlining MaROS GUI development.

To help ensure that user needs were properly met, discussions in the biweekly MaROS Users Working Group meetings revolved around higher level design product demos and discussions, while meetings with select individual users representing missions most impacted by the change were held to collect more detailed feedback. The act of creating more detailed UI design products and thorough documentation well ahead of the time of development, as well as having the designer available for consultation and clarification during the implementation phase has resulted in a noticeable reduction in blockages and rework for developers. With the introduction of thoroughly documented design frameworks and patterns, users benefit from a much more consistent user experience, and developers benefit from reusability of existing components and styling definitions.

4. Metric Predicts Executive

Providing frequent ephemeris predictions for relay service participants – and consequently updating view periods concerning each relay orbiter-lander pair – is paramount for the success of a relay network. In its initial years,

MaROS required relay service user operators to provide their own view period predictions to the system in the form of Lander Orbit Propagation and Timing Geometry (LOPTG) files [12]. With this mechanism, each lander was responsible for retrieving ephemeris predict files (in the form of SPICE Kernel, SPK files) [13], then calculating view periods for each orbiter and generating LOPTG files to supply to MaROS. This disjointed process resulted in a high potential for user error and inconsistency within the network.

In 2014, a plan was put forth to utilize the Service Preparation Subsystem (SPS) [14] – a service provided by the DSN that acts as a repository for various DSN planning files for all spacecraft that interface with the DSN, including spacecraft ephemeris predict files – for the purposes of automatically calculating relay view periods for all orbiter-lander pairs in the MRN. Development began on a new service, Metric Predicts Executive (MPX) – developed by the same DSN tool team responsible for SPS – which listens for new orbiter ephemeris predict files to be published to SPS, then utilizes the SPICE Toolkit API developed by JPL Navigation and Ancillary Information Facility (NAIF) [15] to perform the calculation of view period start ("rise"), end ("set"), range, elevation and azimuth angles. The time taken to calculate these periods varies depending on the applicable time range of the orbiter SPK, but typically takes around 10 minutes. Once MPX has completed its calculations, it notifies MaROS of the new view periods via an AWS SQS message, at which point MaROS queries the view periods via a ReST endpoint and processes the XML response to correlate view periods to overflights. Upon the updates to overflight geometry, MaROS automatically recalculates potential conflicts, data volume and latency predictions, and subsequently notifies involved MaROS users of any significant changes to their relay requests. This highly automated process greatly improves the consistency of view period updates across the system and reduces the onus on lander operators to calculate, verify, and propagate their own view periods into MaROS.

As with many automated and interdependent systems, reliability and transparency are of great concern to consumers and operators. The simplest solution for the MaROS operators to quickly detect errors between the two systems was to simply build a separate script that consistently monitors SPS for newly published SPKs, then notifies MaROS operators if the file has not automatically been ingested into MaROS within a reasonable amount of time. As a more advanced solution for providing transparency, the Location or Attitude File (LOAF) Visualizer (Figure 5) was added to the MaROS GUI in 2021, which provides users with the ability to view the status of ephemeris predict files (and attitude predict files, though in a more limited capacity) in SPS have been processed by MPX, MaROS, and RTP to

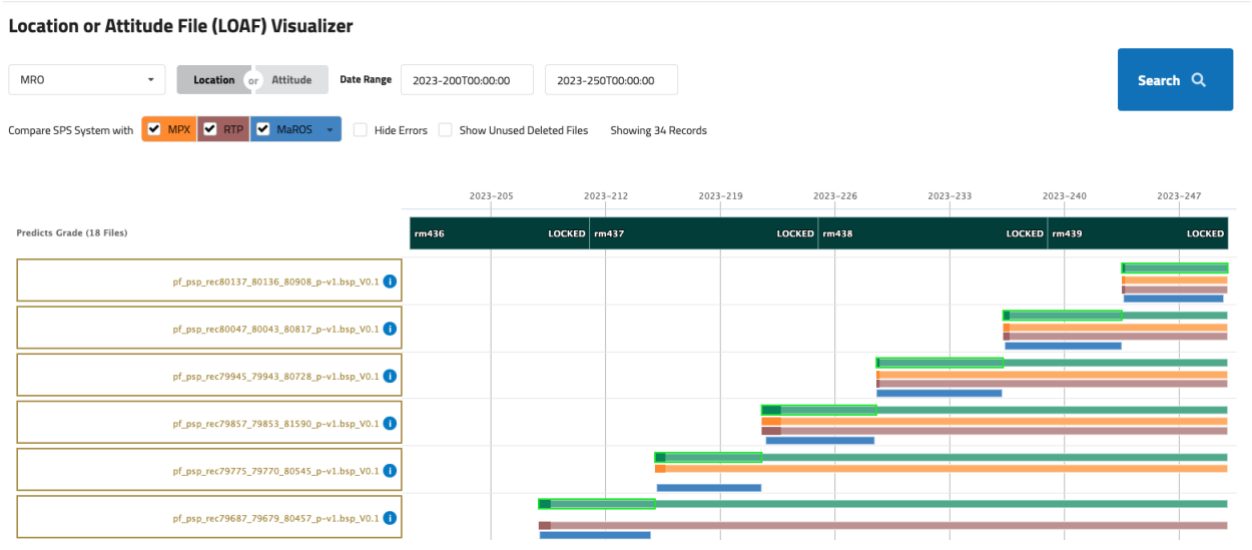

Figure 5: The Location or Attitude File (LOAF) Visualizer in the MaROS GUI, displaying MRO ephemeris file applicability windows and sync status in SPS, MPX, RTP, and MaROS.

ensure that all four systems are in sync with one another. LOAF also displays files based on applicable times on grades, which is useful when determining which file is the "best" prediction for a given time frame. While these solutions provide transparency to MaROS operators and users to make them aware of potential syncing issues between these systems, they don't directly resolve the cause of the issue.

In 2024, a new version of MPX was deployed which improved reliability in several ways. This new MPX uses a ZFS replication file syncing mechanism with the SPS ephemeris repository to reliably monitor new SPK file delivery. Also, this new MPX embraces AWS GovCloud technology by publishing to the SNS to indicate when view periods are ready for querying. SNS allows for various other event-based integrations for consumers such as MaROS, which utilizes a listener on AWS's SQS to trigger ingestion of view periods from MPX. The new backend has shown promising results and these improvements will provides a more reliable experience for MaROS users and operators alike (Figure 6).

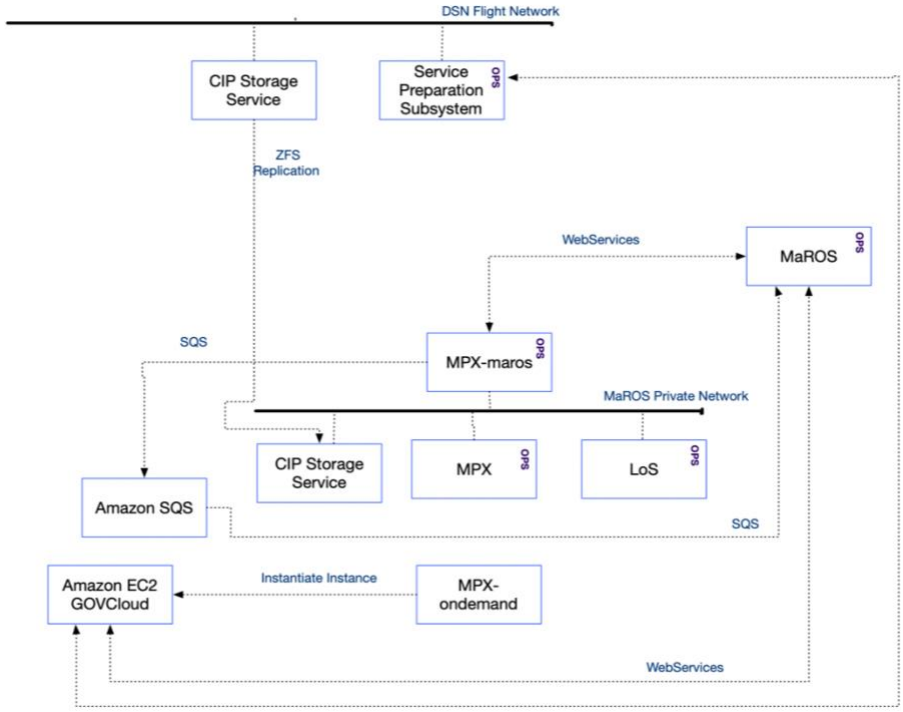

Figure 6: The Architecture of the "New" MPX Version, with relation to MaROS

Apart from reliability and transparency improvements, enhancements to support new relay use cases are also underway for the MPX development team. A new service called Line of Sight (LoS) utilizes not only orbiter and planetary ephemerides to calculate view periods, but also accounts for lander location, orbiter and lander attitudes, and local terrain to calculate view periods between any two objects in space. With this service, MPX now has an option to include rise and set times and elevation angles for view periods accounting for local terrain, where previous view periods were calculating assuming Mars to be a smooth surface. These terrain-adjusted view periods are now shown in MaROS alongside existing view period geometry and allow relay planners to account for potential interruptions due to terrain obstructions. Mission relay communications planners currently add a constant elevation and/or time buffer to the MPX rise/set times when scheduling requests that may be revisited for further accuracy with these new terrain-adjusted view periods to take advantage of every precious second of data transfer from the surface of Mars [16]. Because LoS is able to calculate view periods for any two objects in space, it is also very likely to be utilized for calculating view periods between two orbiters – a MaROS capability that is planned for implementation to support future MRN architectures.

5. Relay Telecom Predictor

One of the key metrics used for relay planning and scheduling is the predicted amount of data volume that can be transferred on each overflight. Typically the responsibility of mission telecom teams, there are various models used to make such predictions, and many factors that go into these models. As one might imagine, many of those factors are geometry-dependent – when are the rise and set for the overflight, what is the range from the lander to the orbiter, what is the attitude/orientation of the spacecraft involved, etc. Other factors are specific to the relay operation parameters of the involved spacecraft that are coordinated through MaROS – when will the spacecraft radios hail, what frequency will be used on this overflight, will the spacecraft communicate on a fixed data rate or will they use Adaptive Data Rate (ADR), etc [17]. The responsibility of tracking the variety of models used, combined with ever-changing geometry and MaROS overflight parameters for multiple spacecraft creates an organizational headache for each mission's telecom teams. Return link data volumes are fed back into MaROS by the relay service user operators for each overflight request as a single number – a predict, in Megabits, with no correlating metadata describing what information went into the calculation of that prediction. To complicate things further, the primary data volume models used require the calculation of Link-Power and Signal-to-Noise Ratio (LPSNR), a computation-heavy and time-intensive calculation [18].

In 2019, Curiosity and Perseverance telecommunications engineers had finally had enough of the disorganization, and a proposal was funded by Advanced Multimission Operations System office (AMMOS) within MGSS to build a service that can be used by all MRN missions for data volume predict calculation. The result, released initially in 2021, was the introduction of a new centralized service highly integrated with MaROS called the Relay Telecom Predictor (RTP). RTP has the following primary responsibilities:

- Execute mission-supplied black-box models for LPSNR and data volume predictions upon request.
- Split LPSNR and data volume calculations into a two-step process to allow time-consuming LPSNR calculations to be precalculated and used many comparatively quick data volume predictions.
- Utilize geometry file grades and applicability timeframes to automatically use the "best" geometry records as inputs for predicts for a given overflight, as well as the "best" corresponding model given the parameters provided in the request.
- Store LPSNR and data volume predict calculation inputs and outputs in a traceable and recallable way.
- Return pre-existing LPSNR and/or data volume prediction records if a request is submitted with identical inputs, rather than recalculating.

To achieve these goals, design for RTP began with its database. Much of the data that would track inputs and outputs for the predicts would be highly flexibly JSON strings to account for varying types of models. After some review of NoSQL database options were assessed, the decision was made to use a relational database (MySQL 8+), allowing the schema to enforce that each prediction record had the requisite traceability to its inputs. Because LPSNR computations are required for most data volume predictions, and because these computations are primarily geometry-dependent, it was determined that each LPSNR request would be tied to the four points of geometry – orbiter and lander attitudes and ephemerides, as input. If the desired geometry records are not specified in the LPSNR request, RTP will automatically determine which geometry records are "best" for the desired duration and create an "LPSNR Run" for each overlapping set of geometry records. Once the LPSNR has been completed for a duration, data volume requests are executed for each overflight within that duration using the parameters defined for the overflight within MaROS and are tied back to the LPSNR Run(s) that were used to calculate data volume. This provides an inherent link to the geometry, models, and parameters that were supplied as input to calculate the data

volume thanks mostly to the inherent schema of the relational database. The algorithms incorporated into the RTP server for determining the correct geometries and model definitions and comparing new inputs against existing records alleviates many of the other pain points experienced in the past.

The other biggest hurdle to overcome in developing the system was the act of running of the models once LPSNR or data volume predicts are requested. To allow for generic models to be run in the system, the models are run within Docker containers [19] (referred to internally as "engines") that contain all of the code and infrastructure needed to compute a predict. Many of the variables within the engine were abstracted out of the engine code and stored in the RTP database as "model constants", so that any changes to certain adjustable variables within the model would only require a database change rather than a redeployment of new code and to improve traceability. For an arbitrary example, if it was decided that MRO models should change from 0 dB to a 1 dB margin policy, a new model definition could be added to the database and all subsequent data volume predicts using that new model would use the new margin policy, while old data volume predicts would remain traced back to the old model with 0 dB. A standard file structure was described for the engine containers, consisting mainly of an input directory for geometry files, model definitions, and input parameters that would be passed in for a given run, and an output directory for capturing the resulting predictions along with any corresponding output metadata to be fed back into the RTP database.

Along with utilizing Docker engines, AWS Elastic Container Service (ECS) is used as the orchestrator to run the Docker containers as "Tasks" [20]. ECS allows for multiple engines to be running simultaneously, which is utilized by RTP for parallelization. Each engine type is paired with a "sidecar" container developed by the RTP development team which handles much of the overhead of interacting with RTP. When a request comes into the RTP server, it puts a message onto an SQS queue, which is picked up by a single, custom Task Manager task that spins up the correct engine for that request. The sidecar pulls inputs from RTP into the engine input directory before running the engine's "start" command. The sidecar does not have insight into the engine logic itself (i.e. "black-box"), then monitors the engine's output directory for a completed file, at which point it will send the predict result back to RTP's database and kill the job. This abstraction allows the engine developers to focus on the contents of their engine, without needing to worry about all of the overhead specific to how RTP works. It also allows the RTP development team to manage higher level changes in a single place and propagate those to all engines in the system.

Finally, the integration of RTP with MaROS yields many tangible benefits to relay planners and telecom engineers alike. Because RTP is now a centralized service, predicts can be automatically fired off from MaROS whenever any inputs to those predicts change. For example, when a new orbiter SPK file is delivered, MaROS automatically fires off LPSNR and data volume predict requests for each overflight effected by that SPK. Upon completion, MaROS updates the Current Best Estimate (CBE) data volume predict for each overflight, which is displayed to users in MaROS. Likewise, if a relay planner updates the parameters defined in an overflight request in MaROS, a new data volume prediction is computed right away and the CBE is updated. This automation ensures that the CBE data volume predict is always up-to-date with the latest inputs, making them trustworthy for relay planners without the need for telecommunications engineers to fire off these calculations manually. Because these predictions are all stored in RTP, a history of changes to data volumes for a given overflight can be collected and analyzed from a single source. The upfront computation of time-consuming LPSNR allows relay planners to trigger data volume predictions in real-time before submitting their relay requests. Support has been added to the MaROS GUI for users to make changes to an overflight request and see the effects of the change on the predicted data volume before submitting to assess the impact of their changes in real-time.

6. "Where's My Relay Data?"

Imagine, the operators of the Mars 2020 rover are expecting to receive data from a TGO relay pass that would influence the command schedule for the following day. According to MaROS's latency prediction, the data was supposed to be on the ground just before 9am local time, but it's now 9:30am and the data still hasn't arrived. Was there a problem sending the data from TGO to the ESTRACK ground station? Or maybe the data made it to the ground, but got caught up in the ground processing system somewhere? If so, where do the operators look first to get their data in time to commence their planning?

Though issues in the relay pipeline may not be terribly common, they occur often enough for frustrations in such a situation to mount, and many dollars to be spent trying to push the data through the pipeline in a timely manner. At worst, if the impacted data was critical for mission planning, missions may lose the most valuable resource of all: time on Mars. While it's virtually impossible to create a solution to completely prevent hang-ups from happening in the various downlink data pipelines used in the MRN, it seems plausible to provide enough transparency into the dataflow to help answer the question asked by mission operators: "Where's My Relay Data?" (WMRD). Design and development of WMRD began in 2022 and was initially released in 2024.

WMRD is a new subsystem within MaROS, funded as a joint effort between AMMOS and MEP, which takes the form of a customizable dashboard within the MaROS GUI comprised of widgets to show the real-time status of a

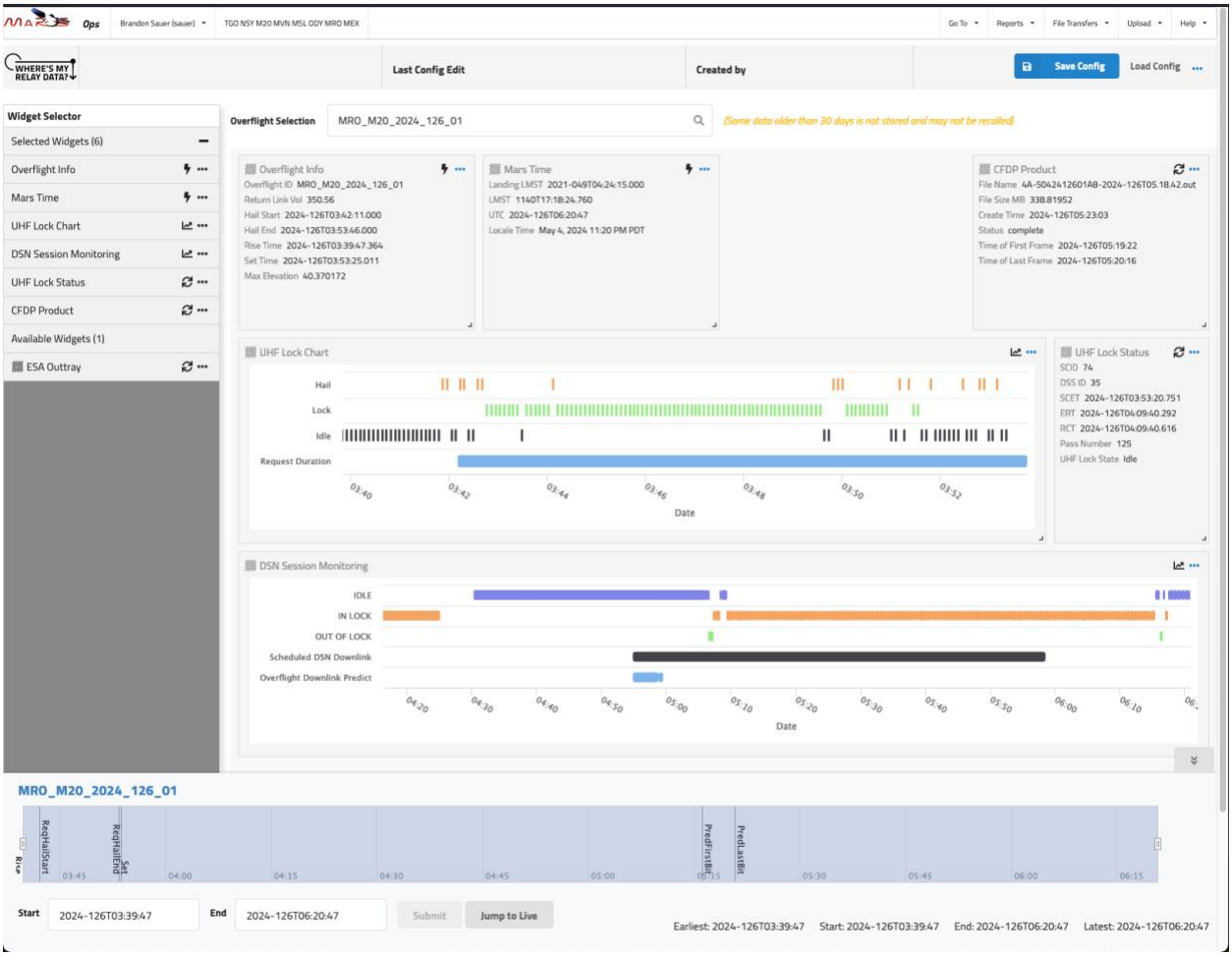

Figure 7: A sample screenshot of the Where's My Relay Data Dashboard in MaROS

relay data product as it passes through each data custody holder. The widgets in the WMRD dashboard show telemetry data reflecting the performance of the lander-to-orbiter UHF link, monitoring of the scheduled orbiter-to-ground link when relay data is expected to be downlinked, and the various gates the data passes through as it is processed on the ground before it makes it to the lander teams. Different widgets may pertain to particular data flows, for example, some widgets are specific to overflights involving ESA orbiters, which may transmit data through ESTRACK instead of the DSN, following different ground processing pathways along the way (Figure 7).

The largest hurdle identified for implementing this ambitious endeavor was figuring out how to aggregate data from the many data custody holders and gates along the dataflow into a single, centralized data source that the dashboard could pull from. The implementation of WMRD includes a centralized server, database, and other infrastructure designed to collect status data from custody holders. With a goal to provide flexibility and a low barrier to entry for data custody holders to publish their relay data information in real time, the server accepts status data in nearly any structure or format, along with an applicable time duration, mission(s), and data type. That raw status data is then routed to a post-processor specific to that data type, which standardizes the output into a format that can be requested by the dashboard for the various widgets. This is accomplished due, once again, to a variety of services offered by AWS. A Python Flask [21] server accepts raw data via a ReST API, stores the data as files into a Simple Storage Service (S3) repository, and creates a record with a reference back to the S3 location in an AWS DocumentDB instance. Upon accepting the raw data, a message is published to SNS that data of this type has been received. An AWS Lambda service detects messages specific to its corresponding data type, transforms the data into a format that will be useful to the dashboard, and writes that post-processed data back to a separate collection in the DocumentDB. Finally, the Flask server allows ReST API requests from the dashboard GUI to poll for post-processed data depending on which widgets are currently selected by the end user (Figure 8). This architecture, though complex, allows for the high flexibility needed to support data ingestion of various types from many different sources simultaneously.

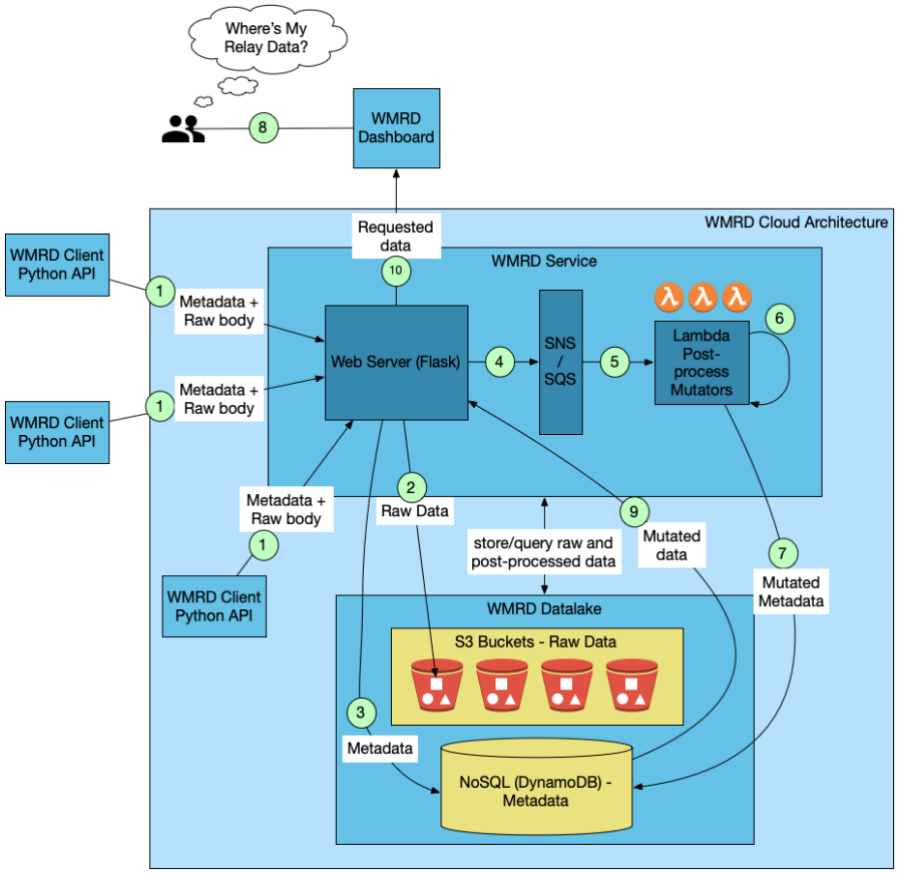

Figure 8: Architecture Design of "Where's My Relay Data" Implementation

As with all aspects of the MaROS GUI, each widget is being designed with user experience in mind, and feedback is being collected from MaROS and expected WMRD power users. With the component-based design of the dashboard and the flexibility of the underlying architecture, the hope is that WMRD will remain flexible to display downlink dataflows for future MRN missions or reflect any future changes to the present dataflow with relative ease.

7. Conclusion

MaROS has proven over its decade-plus lifespan to be an invaluable resource for participants in the Mars Relay Network. In a relatively short amount of time, the service has taken on multiple shifts in functionality as it adapts to advancing technology both at Mars and on Earth. Undoubtedly, more new challenges will present themselves in the future [22], and the cycle of advancements will result in continued changes to MaROS so long as relay at Mars continues to require hands-on coordination.

While the enhancements to MaROS described in this paper are specific to the experiences of the Mars Relay Network, the lessons learned from the changes required to the system are likely relevant to any relay network.

8. Acknowledgements

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

9. References

[1] "Mars Relay Operations Service (MaROS): Rationale and Approach", R. Gladden, SpaceOps 2010, Apr 2010, Huntsville, Alabama, USA.

[2] "Mars Relay Operations Service (MaROS): Managing Strategic and Tactical Relay for the Evolving Mars Network", D. Allard, R. Gladden, IEEE Aerospace Conference, Mar 2012, Big Sky, Montana, USA.

[3] "Relay Support for the Mars Science Laboratory Mission", C. Edwards et al, IEEE Aerospace, Feb 2013, Big Sky, Montana, USA.

[4] "Preparing the Mars Relay Network for the Arrival of the Perseverance Rover at Mars", R. Gladden et al, IEEE Aerospace, Mar 2022, Big Sky, Montana, USA.

[5] "Lessons Learned from the Mars Relay Network: Considerations for Future Relay Networks", R. Gladden et al, IEEE Aerospace, March 2024, Big Sky, Montana, USA.

[6] "Mars Relay Operations Service (MaROS): A Present Service Preparing for the Future", R. Gladden, SpaceOps 2014, Apr 2014, Pasadena, California, USA.

[7] "Eaton Fire", available at https://www.fire.ca.gov/incidents/2025/1/7/eaton-fire, provided by CAL FIRE, January 7th, 2025.

[8] "GraphQL", available at https://graphql.org/, provided by The GraphQL Foundation.

[9] "Eyes on the Mars Relay Network", available at https://eyes.nasa.gov/apps/mrn/, provided by Jet Propulsion Laboratory.

[10] Report of the Interagency Operations Advisory Group Mars and Beyond Communications (MBC) Architecture Working Group, Vol. 1, "The Future Mars Communications Architecture", Final Version, 22 Feb 2022.

[11] "React: The library for web and native user interfaces", available at https://react.dev/, provided by Meta Platform Inc.

[12] "Mars Relay Lander and Orbiter Overflight Profile Estimation", Michael N Wallick et al, NASA Tech Briefings, December 1st, 2012.

[13] "SPICE Data (SPICE Kernels)", available at https://naif.jpl.nasa.gov/naif/data.html, provided by The Navigation and Ancillary Information Facility (NAIF) at JPL.

[14] "Integrated Planning and Scheduling for NASA's Deep Space Network – from Forecasting to Real-time", M. Johnston and J. Lad, SpaceOps 2018, May 2018, Marseille, France.

[15] "SPICE Toolkit", available at https://naif.jpl.nasa.gov/naif/toolkit.html, provided by The Navigation and Ancillary Information Facility (NAIF) at JPL.

[16] "Playing Telephone Through Martian Rock: Assessing Terrain Occlusions to Enhance Telecom Predictions", E. Wiederhold et al, IEEE Aerospace, Mar 2023, Big Sky, Montana, USA.

[17] "Proximity-1 Space Link Protocol—Rationale, Architecture, and Scenarios", The Consultative Committee for Space Data Systems (CCSDS), December, 2013.

[18] "Mars UHF relay telecom: Engineering tools and analysis", B. Arnold et al, IEEE Aerospace, Apr 2012, Big Sky, Montana, USA.

[19] "Use containers to Build, Share and Run your applications", available at https://www.docker.com/resources/what-container/, provided by Docker Inc.

[20] "Amazon Elastic Container Service (ECS)", available at https://aws.amazon.com/ecs/, provided by Amazon Web Services Inc.

[21] "Flask", available at https://flask.palletsprojects.com/en/stable/, provided by Pallets, 2010.

[22] "Preparing the Mars Relay Operations Service for the Challenges Ahead", B. Sauer, M. Newcomb, SpaceOps 2025, Expected May 2025, Montreal, Québec, Canada.

10. Author Biography

Brandon Sauer *received a B.S. in Computer Science from Cal Poly Pomona. After spending much of his 10-year career at JPL as a software engineer in IT and in mission operations, he is currently the task lead for the Mars Relay Operations Service (MaROS) and other related relay software services.*