Nancy Grace Roman Space Telescope (Roman) Technical Report

Title: An Introduction to the Roman Data Monitoring Tool	Doc #: Roman-STScI-000704, SE-01 Date: October 31, 2024 Rev: -
Authors: William Schultz, Justin Otor, John Wu, Jesse Averbukh, Richard Cosentino, Gisella de Rosa, Tyler Desjardins, Sierra Gomez, Jonathan Hargis, Bradley Sappington, Sanjib Sharma, Megan Sosey Phone: (410) 338-6420	Release Date: December 17, 2024

1 Abstract

The Roman Data Monitoring Tool (RDMT) is a software system that will enable the Roman Space Telescope instrument team to rapidly monitor the calibration data quality of the Wide Field Instrument (WFI) during on-orbit operations. The RDMT comprises two applications: the RDMT-local, which will monitor WFI calibration reference files after their publication to the Calibration Relational Data System on-premises at the Space Telescope Science Institute, and the RDMT-cloud, which will monitor calibrated detector-level science data to diagnose peculiarities and systematics in the slope images in the Amazon Web Services commercial cloud. The RDMT-local and RDMT-cloud will store data quality metrics in databases and saved files, and will send automated notifications to the Roman instrument team if the monitored metrics exceed pre-defined thresholds or exhibit anomalous trends. Due to the large size of Roman data, the RDMT is designed to be lightweight and computationally efficient, empowering Roman Science Operations Center instrument scientists to review the scientific data quality for processed WFI detector-level data products and share the information with the broader community.

Operated by the Association of Universities for Research in Astronomy, Inc., for the National Aeronautics and Space Administration under Contract #80GSFC19C0054

2 Introduction

The Nancy Grace Roman Space Telescope, hereafter referred to as Roman, will image large swaths of sky with the Wide Field Instrument (WFI). The WFI's 18 detectors combine to form an effective 302 Megapixel camera. The wide field of view and Roman's rapid survey speeds will produce > 10 Tb of uncalibrated data every day of the nominal 5 year mission. The Science Operations Center (SOC), located at Space Telescope Science Institute (STScI), is responsible for processing Roman data during operations (Ferguson et al. 2022). Calibration reference files are created for use in the science data calibration pipelines via the WFI Reference Files Pipeline (RFP). Validated reference files are submitted to and published in the Calibration Reference Data System (CRDS) for archival storage and reference. A subset of the created reference files are used by the Exposure-Level Pipeline (ELP) to calibrate raw ramp data from a single detector (Level 1 data products) into detector slope images (Level 2 data products) on-premises. After the ELP has generated the Level 2 science data, the High-Level Pipeline (HLP) combines the detector-level data products and resamples the images onto a uniform grid across the sky (Level 3 data products), and also produces higher level science data (Level 4 data products). Level 3 and Level 4 products will be created and stored in the STScI Amazon Web Services (AWS) commercial cloud. All told, the SOC must ensure the efficient and sufficient calibration of what is an unprecedented data volume (compared to past NASA missions) across multiple platforms.

As an SOC product, the Roman Data Monitoring Tool (RDMT) will verify that reference files and Level 2 data are sufficiently calibrated and stable for scientific use. Although the schema and structure of reference files will be verified and validated during delivery to CRDS, these requirements do not guarantee the files can sufficiently calibrate Level 1 products. For example, issues with scattered light or $1/f$ noise may not impede the creation of calibration reference files or the data pipelines, but they may impact scientific analyses and warrant further attention.

The RDMT cannot directly monitor the data quality of reference files or Level 2 files and will thus compute and utilize summary statistics, including detector-wide and localized means and variations, as data quality metrics. By comparing data quality metrics against predetermined thresholds, the RDMT will enable SOC instrument scientists to rapidly identify cases where these metrics deviate from expected trends or baseline values. Throughout this report, we refer to such cases as "anomalies." We additionally refer to the process of computing data quality metrics and evaluating them against predetermined tolerances and their previous values as monitoring, checking, or assessing the "quality" of the data.

2.1 SOC Scope of Work

Under the Roman SOC Contract, the SOC is charged with building, maintaining, and operating a WFI data monitoring tool for use by instrument scientists during operations. This capability fills a critical gap in the previous plan for monitoring WFI data. While WFI health and engineering monitoring is primarily the responsibility of NASA's Goddard Space Flight Center (GSFC), science quality assessment falls under the purview of the Roman science teams (e.g., the project infrastructure teams, or PITs).

Between these two functions, however, there is a need for this monitoring tool to perform further quality checks on prompt data products after they are processed through the SOC Data Management Subsystem (DMS) pipeline. For example, we need to ascertain that the data meet basic imaging quality metrics such as the expected point spread function (PSF) shape or overall stability, expected background levels, and detector bias levels. Though this tool is primarily intended to monitor prompt products, instrument scientists may also employ it in tandem with more specialized tools to perform retroactive quality checks on subsets of reprocessed data products.

Appendix B of the SOC Contract Offer Volume discusses three relevant technical assumptions regarding the development and operation of the RDMT:

104. The SOC will reuse existing software from the Hubble/Wide Field Camera 3 (WFC3) and Webb quick-look tools as the foundation of the Roman data monitoring tool.

105. The SOC data monitoring tool will make use of existing data only, including DMS pipeline products and catalogs, instrument/data product metadata keywords, and telemetry and engineering health data as available from GSFC or other mission partners. No new data products will be developed specifically for this tool.

106. The SOC data monitoring tool will operate in an automated fashion, requiring staff intervention only as problem images/outliers are identified.

Monitoring is broadly defined as the statistical characterization of an observation or the tracking of a statistical metric over time. The choice of the statistical metric (and any associated computational complexity) should be determined based on the instrument/detector characteristic being monitored. There are three classes of quantities that can determine the quality of data calibration:

- *Detector characteristics* such as gain, read noise, bias level, dark current, etc.
- *Instrument characteristics* such as focal plane temperature, etc. GSFC is responsible for monitoring the WFI health and engineering as part of the Mission Operations Center (MOC).
- *Observation characteristics* such as PSF stability, guide star metrics, etc.

The RDMT will monitor both detector and observation characteristics. When anomalies arise, the RDMT will notify instrument staff for manual follow-up. Combining these metrics with instrument characteristics allows for better understanding of anomalous results. Unlike past Quicklook tools developed for HST and JWST, the RDMT will carry out this process completely autonomously in order to handle Roman's exceptional data volume.

Though the immediate notifications will only be delivered to STScI instrument scientists, the RDMT will support formal procedures for problem reporting to the broader community during operations in the form of reports on detector and observation characteristics. These reports will be available on a public-facing platform. The format and cadence of the reporting is still under development.

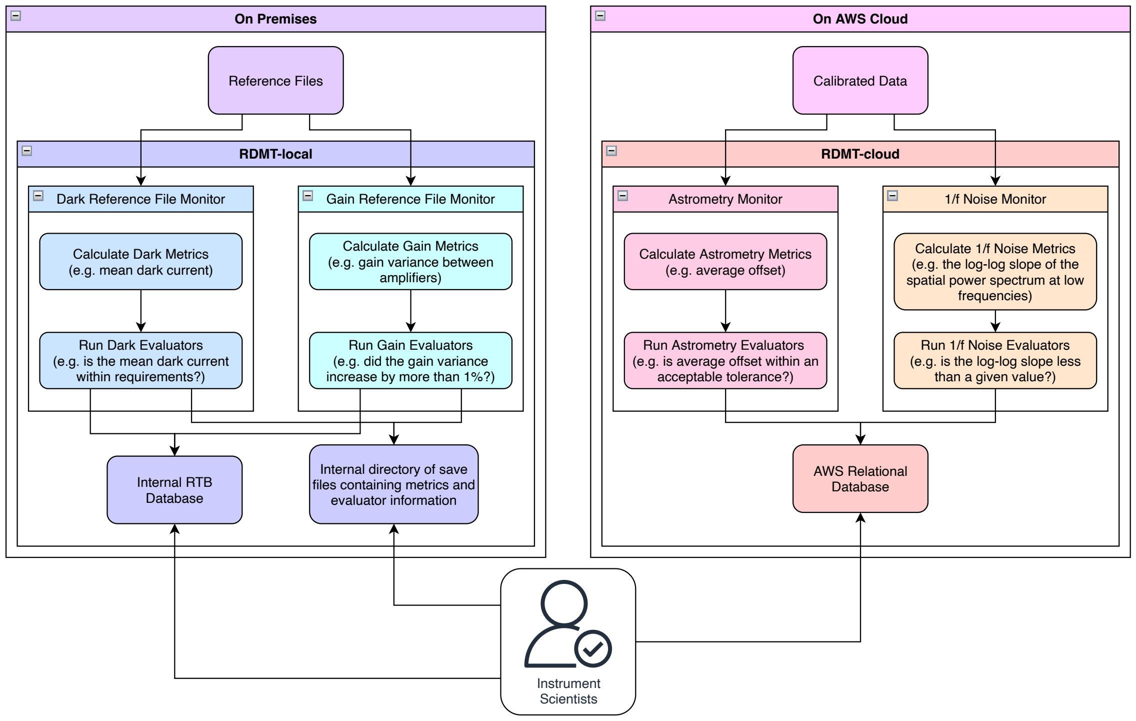
On Premises
Reference Files
RDMT-local
Dark Reference File Monitor
Calculate Dark Metrics (e.g. mean dark current)
Run Dark Evaluators (e.g. is the mean dark current within requirements?)
Gain Reference File Monitor
Calculate Gain Metrics (e.g. gain variance between amplifiers)
Run Gain Evaluators (e.g. did the gain variance increase by more than 1%?)
Internal RTB Database
Internal directory of save files containing metrics and evaluator information
On AWS Cloud
Calibrated Data
RDMT-cloud
Astrometry Monitor
Calculate Astrometry Metrics (e.g. average offset)
Run Astrometry Evaluators (e.g. is average offset within an acceptable tolerance?)
1/f Noise Monitor
Calculate 1/f Noise Metrics (e.g. the log-log slope of the spatial power spectrum at low frequencies)
Run 1/f Noise Evaluators (e.g. is the log-log slope less than a given value?)
AWS Relational Database
Instrument Scientists

Figure 1: **High-level overview of the entire RDMT architecture.** The left panel shows the on-premises RDMT-local, which ingests reference files upon their creation and calls the appropriate monitors. The monitors then calculate the metrics and compare them in the evaluators before saving the outputs. The AWS Cloud infrastructure (right panel) mimics this structure in the RDMT-cloud, which is triggered on the Level 2 products created after the ELP processing. The bottom of the diagram highlights how instrument scientists will query the databases and/or saved files to determine the calibration quality of the data products.

3 RDMT Design

The RDMT builds on the successful Quicklook applications developed for the Hubble and JWST missions (Bourque et al. 2017, 2020). Of particular note, the WFI H4RG-10 detectors operate comparably to the H2RG detectors employed by the JWST NIRCam, NIRISS, and NIRSpec instruments, so, where overlap exists, the RDMT reference files and Level 2 monitors will be based on similar monitors in the JWST Quicklook Application (JWQL; Bourque et al. 2020).

The RDMT will monitor reference files and Level 2 science data, the respective inputs and output of the ELP. Since reference files will be stored on-premises, we design one “arm” of the RDMT, the RDMT-local, to operate on-premises, monitoring the files after they are created and delivered to CRDS. Monitors that are meant for Level 2 SOC data products will execute as part of the other arm, the RDMT-cloud, when those products are transferred to the cloud-based Archive. Figure 1 presents the overall approach and data flow for the

RDMT.

These two arms of the RDMT function as independent and automated applications that produce different characterizations of WFI data at different stages of calibration. The RDMT-local focuses on monitoring calibration reference files for issues at the detector level (e.g., gain, read noise, bias, dark current). The RDMT-cloud focuses on monitoring scientific data at the observation level (e.g., PSF shape and stability or guide star metrics). More complex artifacts like snowballs and dragon's breath, which fall under the scope of the RDMT-cloud, may also be monitored on a best-effort basis. Section 3.1 covers both arms' structure in greater depth.

The RDMT will notify instrument science teams about any issues in the data. Select functionality of the tool and all algorithms used therein will be available to the general public. We will also deliver automated reports that summarize outputs of the data monitors on a best-effort basis. The format and contents of these reports are still under development.

The current levels of development for the RDMT-local and RDMT-cloud are discussed in Sections 4.4 and 5.1, respectively.

3.1 Key Functionality

The science calibration pipeline software, which is part of the ELP, applies reference files to Level 1 ramps to produce Level 2 slope products. The RFP is integrated with the RDMT-local in order to monitor the quality of newly-created reference files. The RDMT-local will automatically compute metrics on pixel, amplifier, and detector scales to identify changes and trends in the characteristics of these reference files and store these metrics in a database. The specific list of statistics that quantify the characteristics that will be tracked has not been finalized. The RDMT will utilize the result from Thermal Vacuum (TVAC) testing and the expertise of other SOC staff to finalize a list of summary statistics that fully quantify potential anomalies that may arise during operations.

The RDMT-cloud will compute detector and pixel-level metrics for each Level 2 slope image created by the ELP in order to quantify the observation characteristics of each science exposure. It includes a database for metrics storage and functionality for automated notifications. Notifications from the RDMT-local and the RDMT-cloud may arrive at different times because reference file creation is decoupled from the execution of DMS science calibration pipelines. Instrument scientists will have access to both RDMT systems to facilitate root cause analysis of problematic data.

Figure 1 highlights the symmetric infrastructures of the RDMT-cloud and RDMT-local. The details within this figure are explained in Sections 4 and 5.

3.2 Nomenclature

We define three terms to describe the fundamental components that comprise each arm of the RDMT. They are as follows:

- *Metrics* are lower-dimensional data products that quantify the monitored data. The idea behind the metrics is to ensure that the quality of the higher-level data and reference files can be accurately addressed quickly without inspecting each pixel. Some examples include the mean and median dark rate across the detector, the 99th percentile of the gain across an amplifier, and the average astrometric offset across a set of Gaia stars.

- *Evaluators* are boolean comparisons that determine if the metrics indicate anomalies in reference files or the calibration of the science data (true if the evaluation passes, false if it fails). They are agnostic to the individual metrics they validate by comparing the metrics to dynamic reference values. Some examples include ensuring a metric does not exceed a set value or does not change significantly from one observation to another.
- *Monitors* are the runners that make up the RDMT itself. Each monitored data product and/or specific component of those data products will have its own monitors. The monitors calculate their unique set of required metrics and then compare those metrics using a suite of evaluators. Each monitor is also responsible for notifying instrument staff about any potential problems and managing the database and saved files. The example monitors discussed herein include the dark reference file monitor, the astrometry monitor, and the $1/f$ noise monitor.

This nomenclature will be used throughout this text and is referenced in both arms of the RDMT. Figure 1 highlights each arm of the RDMT in a simple schematic of its general shape. Each arm of the RDMT will maintain a separate set of monitors that will be triggered based on the type of ingested file. These monitors each house the infrastructure to calculate the required metrics and then compare them with the necessary evaluators to assess their data quality. Each monitor then writes useful outputs to a database and produces saved data products both locally and in a centrally accessible location. Instrument scientists will use the RDMT databases and saved quantities, alongside other project data (e.g., engineering data), to analyze calibration quality and manage any problems the RDMT identifies with the released data products.

4 RDMT-local

The main goal of the RDMT-local is to assess and track the quality of the products used to calibrate the WFI science data. This quality assessment will include monitoring trends from reference file-derived quantities like bias levels/structure, dark current, gain, dead/bad pixels, read noise, and reference pixels. These quality assessments will culminate in notifications to instrument scientists when anomalous results are identified. These notifications will be accompanied by reports that detail the values of each metric and how they compared to their accompanying standards. Though these reports will be made publicly available, the exact process of dissemination is still being finalized. In quantifying these characteristics, the RDMT-local also starts the process of gathering data to be used for generating new reference files for the reprocessing of the Level 2 files in the planned Roman data releases.

In the following sections, we describe the RDMT-local in more detail. Section 4.1 provides an overview of the execution procedure of the RDMT-local. The current plan for the tool's structure is detailed in Section 4.2, and Section 4.3 provides information about the software infrastructure and products of the RDMT-local. The current state of development is briefly summarized in Section 4.4.

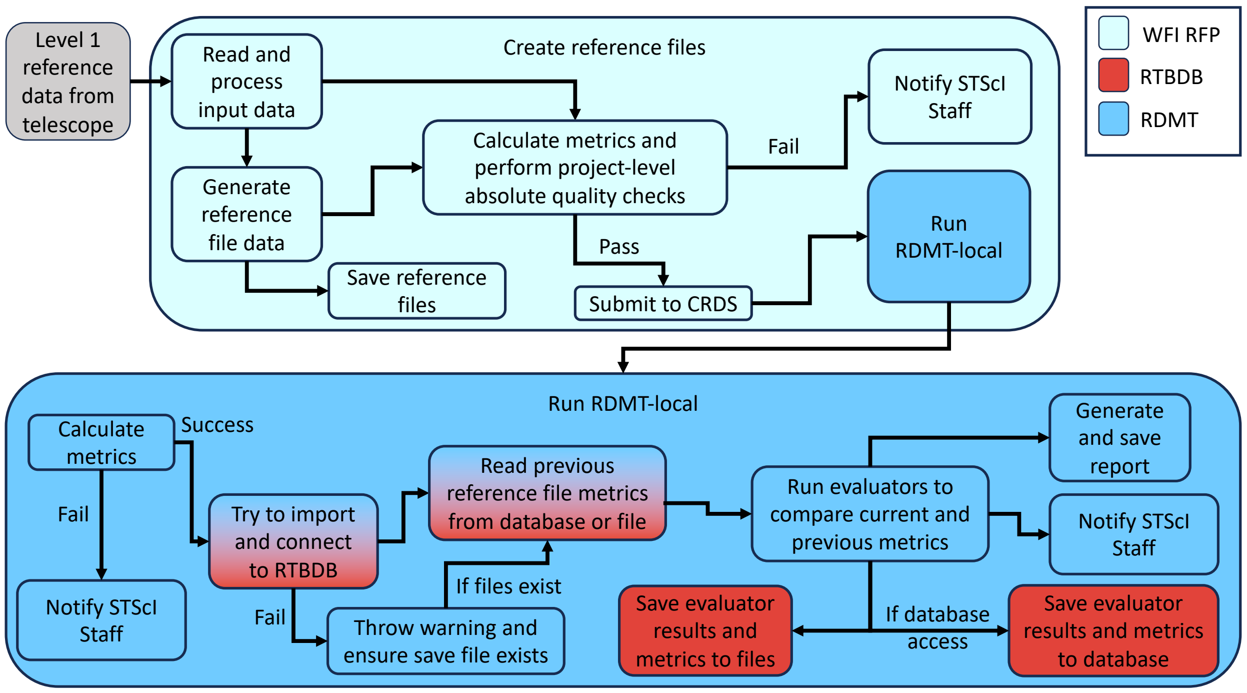
Level 1 reference data from telescope
Create reference files
Read and process input data
Generate reference file data
Save reference files
Calculate metrics and perform project-level absolute quality checks
Fail
Pass
Notify STScI Staff
Submit to CRDS
Run RDMT-local
WFI RFP
RTBDB
RDMT
Run RDMT-local
Calculate metrics
Success
Fail
Notify STScI Staff
Try to import and connect to RTBDB
Fail
Throw warning and ensure save file exists
If files exist
Read previous reference file metrics from database or file
Run evaluators to compare current and previous metrics
Generate and save report
Notify STScI Staff
Save evaluator results and metrics to files
If database access
Save evaluator results and metrics to database

Figure 2: **RDMT-local architecture and integration with the RFP.** We present a logic chart for the WFI RFP (light blue) and the RDMT-local (darker blue). The process starts with the creation of reference files in the RFP. After they are submitted, the RDMT calculates metrics to assess the data quality, and then evaluates the metrics against standards and attempts to save the results in both the local Roman Telescope Branch Database (RTBDB, red) and save files. At the end, a report is generated and posted for the community to view and notifications are sent to internal instrument scientists, both of which detail the results of the data quality analysis.

4.1 RDMT-local Architecture

The logical decisions connecting the WFI RFP and the RDMT are shown in Figure 2. The WFI RFP reads in Level 1 calibration data from the Archive and processes them into the appropriate reference files. The RFP performs its own set of quality checks to avert catastrophe (i.e., catching a problem that could break the calibration pipeline or make the calibrated Level 2 data unusable). To maintain coherent versions of reference files for the entire instrument, the RFP creates reference files for all 18 WFI detectors before submitting new files to CRDS. After the delivery is accepted, the RFP passes the filepaths to the RDMT-local to run a more thorough quality assessment for the instrument scientists. This order of operations was chosen to ensure the RDMT-local will never impede the RFP's ability to deliver reference files. As such, should the RDMT-local identify issues with a new reference file, instrument scientists would address them in a subsequent release.

Though we expect the reference files to be publicly delivered on weekly-to-monthly timescales (depending on detector stability during commissioning and the first year of the mission), we expect to receive calibration data on a weekly basis. Thus, SOC instrument scientists could create intermediate reference files utilizing subsets of the incoming calibration data to characterize shorter-term detector performance. If these intermediate reference files are created, the RDMT-local could then be used to better track the trends of the calibration quality. Formally-delivered reference files will only be checked for consistency with previously-delivered reference files. The intermediate products may be compared to previous formal deliveries to potentially improve the reference files for the planned Roman data releases.

4.2 RDMT-local Components

The RDMT-local codebase consists of Python scripts developed and maintained by the Roman Telescope Branch (RTB), a subsection of the SOC, and version controlled on RTB's GitLab project. Although the RDMT-local code will primarily be run via a call from the RFP after the creation and delivery of any reference files, it is also designed as a stand-alone package. The code is organized into separate modules for each reference file type. Each module is highly configurable by a combination of default and user-defined configuration files. Though the RDMT-local is designed for automated use by the SOC during operations, it is also flexible enough to be used by instrument scientists or community members on their own computers to evaluate data quality metrics of WFI reference files. Execution options are controlled by flags in configuration files to enable custom workflows.

The foundational components of the RDMT-local are the metrics which quantify the pixel-level data of the reference files into lower dimensional quantities. These metrics span three categories: detector-wide quantities, readout amplifier-specific quantities, and super-pixel quantities. The super-pixels allow for the 4096×4096 detectors to be down-sampled, and thus still provide spatial information but reduce the data volume. In the first version of the RDMT-local, we collapse 128×128 detector pixels into a single super-pixel and thus collapse the 4096×4096 detectors into 32×32 super-pixel arrays. Some examples of currently implemented metrics used to quantify the dark reference files include the mean dark current, dark current percentiles (e.g., 50^{th} or median, 90^{th}, and 99^{th}), and the number of dead, hot, and warm pixels. Each metric is calculated at all three size categories (detector, amplifier, and super-pixel), resulting in 1,057 values for each metric quantity for each detector for each reference file. Though this results in many metric values per reference file, data volume estimates suggest that less than 500 GB of metrics will be generated over the nominal 5-year mission.

These metrics are measured against reference values, which are configurable requirements or standard values, in a set of evaluators. The expected evaluators are grouped into two types: "absolute" and "relative" evaluators. Absolute evaluators compare the metric values to fixed values set by the configuration files (e.g., confirming if a certain requirement is met), while relative evaluators compare the new metric to older metrics or a trend in the previous metrics (e.g., verifying whether the fractional difference between two metrics lies within a given tolerance).

For the first version of the RDMT-local (v0.1.0), we have implemented two absolute evaluators (`mission` and `soc`) and one relative evaluator (`change`). The `mission` evaluator compares the metrics against mission requirements to confirm that the calibration products do not exceed the operational requirements. The `soc` evaluator compares the metrics against a suite of values determined to be concerning by the SOC. These values can change as the

Metric Name	Value	Mission Req	SOC Req	Change Req
detector_dark_current_mean	0.005	1.0	1.00	0.11
detector_dark_current_percentiles				
50%	0.005	0.51	0.51	0.11
95%	0.006 65	0.7	0.70	0.11
99%	0.007 33	0.99	0.99	0.11
detector_num_dead_pix	16 891	1 000 000	100 000	0.11
detector_num_hot_pix	33 636	1 000 000	100 000	0.11
detector_num_warm_pix	50 393	1 000 000	100 000	0.11

Table 1: **Example RDMT-local report table of detector metrics.** This table shows detector metrics and their reference values for a simulated dark reference file taken from CRDS.

instrument and algorithms used by the science calibration software evolve and will likely be established during commissioning. The mission requirements are already determined and will only be modified if there is an unexpected change in the instrument's behavior. The `change` evaluator checks the relative change of a metric compared to the previous metric and verifies that the fractional difference between the two is within a specified tolerance. Default values for these standards will be set by the SOC, but they will be configurable, allowing users to change their values for their own purposes.

4.3 Using the RDMT-local Software

Each reference file type will have its own monitor that will house the framework for the entire quality assessment. Monitors maintain a unique set of metrics relevant to their corresponding reference file type, and all monitors will send notifications to instrument scientists after the analysis has been finished and generate reports that will be publicly available. The notifications contain information summarizing the results of each evaluator, highlighting any failures that occur. Notifications are distributed directly to instrument staff promptly after the reference files are analyzed, allowing the SOC to stay up to date with any anomalies in the newly created files.

Monitor reports are also automatically generated at the end of every evaluation. The final scope of the reports is under consideration, but currently they list all the calculated metrics, the standards used in each evaluator, and highlight any metrics that failed evaluation. Tables 1 and 2 show examples of what the dark reference file detector-level and amplifier number of dead pixel tables look like after processing a simulated dark reference file from CRDS (`roman_wfi_dark_0949.asdf` from context `roman_0063.pmap` uploaded August 7th, 2024). The reports also display figures showing the spatial distribution of the super-pixel metrics (see Figure 3 for an example of a number of dead pixels per super-pixel figure). The reports will be hosted on a public-facing platform so the community can view the calculated data quality metrics and learn about any anomalies in the calibration data products.

In addition to notifications and reports, the monitors save the computed metrics and

Amp	Value	Amp	Value	Amp	Value	Amp	Value
1	533	9	513	17	543	25	568
2	531	10	515	18	515	26	496
3	501	11	472	19	515	27	543
4	542	12	525	20	522	28	530
5	579	13	516	21	527	29	547
6	552	14	516	22	527	30	543
7	524	15	543	23	486	31	552
8	542	16	563	24	530	32	480

Table 2: **Example RDMT-local report table of dead pixel counts.** This table shows the number of dead pixels per amplifier for WFI01 of a simulated dark reference file taken from CRDS.

evaluation results to ECSV files (machine and human-readable files supported by `Astropy`) and to a local, private RTB database (RTBDB). Internal instrument scientists will use the database to query histories of metrics and evaluations when preparing for the planned Roman data releases. In normal use, the RTBDB will also be used to run the relative evaluators (e.g., the `change` evaluator). The ECSV files will make it easier to share evaluation results with team members outside STScI, as the local database is restricted to members of RTB. The SOC will consult with the Science Support Center (SSC) at IPAC and the PITs to define the scope and nature of data sharing of the ECSV files. To allow the community to recreate the output of the RDMT-local without using SOC resources, the tool contains the infrastructure to make new files or connect to a separate database.

The RDMT-local uses existing RTB Microsoft SQL servers and computational resources of the WFI RFP, and thus will not need any additional resources. All calculated and saved values should result in < 500 GB for the nominal 5-year mission, which can easily be accommodated on-premises.

4.4 Development Status

Version 0.1.0 of the RDMT-local has been released to a private GitLab repository for internal SOC staff. This version primarily includes the development of the coding infrastructure to allow for more reference file monitors to be added easily. The current monitor infrastructure includes the ability to distribute notifications to a private Slack channel, create Markdown reports for each file reviewed, and update a suite of tables in the local SQL database, the RTBDB. Currently as a proof of concept, only the dark reference file monitor has been implemented with a minimal suite of metrics discussed in the previous section. For version 0.2.0, we plan to finalize the infrastructure of the codebase, develop a suite of regression tests, and implement our integration with the RFP. During the infrastructure updates, we hope to compile a comprehensive list of eventual metrics to monitor for each reference file type that will be implemented in subsequent releases.

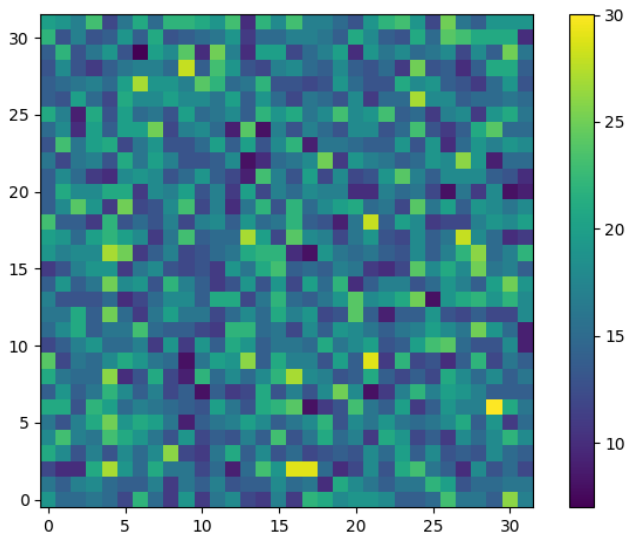

Figure 3: **Example RDMT-local report figure of dead pixels per super-pixel.** We show the number of dead pixels per super-pixel for a simulated dark reference file of a single detector taken from CRDS.

5 RDMT-cloud

The RDMT-cloud assesses whether metrics related to observation characteristics of slope images produced by the ELP are consistent with the SOC's expectations for successful image calibration. The Level 2 calibration data is hosted by DMS in the AWS cloud, so the RDMT-cloud application is also hosted there to decrease latency in image analysis. The AWS environment also offers greater scalability in computing resources and cost efficiency for the RDMT-cloud's specific use case. This arm of the RDMT functions as an automated pipeline composed of several AWS services. It activates when new slope images appear in the DMS Archive and enlists multiple monitors to evaluate them based on observation metrics of interest. Like the RDMT-local, it then saves the results to a final database and will alert instrument scientists when anomalous entries appear.

Section 5.1 is a high-level description of the current state of RDMT-cloud development, including brief summaries of the prototype astrometric and $1/f$ noise monitors included with the tool's first release. Section 5.2 explains the procedures used to simulate reference images for the monitors to analyze.

The rest of Section 5 functions as a more detailed investigation of the tool's AWS infrastructure for those interested in the design choices that have informed the tool's development. Section 5.3 lays out the STScI procedure for moving cloud applications into a production environment, while Section 5.4 discusses the sandbox environment in which we have prototyped the RDMT-cloud. Section 5.5 describes the process for adding new monitors to our AWS pipeline. Section 5.6 details the platforms and public visibility of the RDMT-cloud's components. Section 5.7 takes a closer, sequential look at the AWS resources that make up the RDMT-cloud pipeline. Finally, we provide peak operational cost estimates for a fully realized RDMT-cloud in Section 5.8.

5.1 Development Status and Current Monitors

Version 0.1.0 of the RDMT-cloud is currently undergoing code reviews in private repositories hosted on multiple platforms. (See Section 5.6 for more on the rationale behind splitting the version control platforms.) As detailed in Section 5.7, this version includes a fully functional AWS pipeline and prototype implementations for the aforementioned astrometric and $1/f$ noise monitors. After the completion and release of the next version (0.2.0), the application logic scripts that the monitors use to evaluate new images will be made publicly available.

The prototype astrometric monitor compares the expected positions of known stars from a simulated catalog, sourced from the Gaia catalog, to their observed positions in a new image. It estimates the latter by modeling PSFs to point sources found in the image and returns the root mean square offset of matching PSF centroids from their simulated coordinates.

The current $1/f$ noise monitor takes an image's power spectrum by rearranging its pixels into a time series, taking a fast Fourier transform of the result, and judging whether the power measured at the lowest frequency shows evidence of $1/f$ noise removal. It returns a boolean that indicates a successful removal when the power at that frequency falls below an image-specific threshold.

In version 0.2.0, we plan to make our existing monitors more efficient, add high-priority monitors as identified by the RTB Calibration block, and develop an infrastructure as code solution that will allow instrument scientists to run the pipeline outside of the AWS console. (See Section 5.6 for more information on the last goal.)

5.2 Simulating Data

In anticipation of operational data, the RDMT-cloud currently ingests scenes simulated by Roman I-Sim, a WFI simulator supported by STScI. We maintain collections of reference images for each monitor. The astrometric monitor's simulated images are based on a scene that features around 300,000 point sources (e.g., known Gaia stars) and extended sources (e.g., simulated galaxies using GalSim) per detector. There is a corresponding image for each detector, making for 18 in total. All images are classified as Level 2 data products and use the F158 filter. We save the scripts and software environment used to generate the images on STScI's Central Store file server.

The $1/f$ noise monitor's simulated images are initially created in an identical manner to those from the astrometric monitor, with the same initial conditions and number of end products. The $1/f$ noise monitor's simulated images then go through an extra step where they are injected with $1/f$ noise. Our sample $1/f$ noise comes from the difference between rate images of darks from just before and after the Roman pipeline's Improved Roman Reference Pixel Correction (IRRC) step. We inject the images for detectors WFI13-18 with 50 times that difference, WFI07-12 with 25 times that difference, and leave WFI01-06 untouched in order to test the monitor's response to three levels of $1/f$ noise.

We also upload each monitor's associated images to the cloud via the Amazon Simple Storage Service (S3), AWS' object storage solution. Each monitor has its own bucket where the latest versions of each image are stored.

5.3 Cloud Application Deployment

Version 0.1.0 of the RDMT-cloud is prototyped in a sandbox environment within the STScI AWS cloud presence. In order for the different AWS services we use to communicate, we must set up an instance of the Amazon Virtual Private Cloud (VPC), an AWS service for creating private networks, as well as the subnets (i.e., controlled-access IP addresses) over which they can transfer information. We define Security Groups that allow inbound traffic from member services to pass while firewalling content from other sources. Finally, we set up an endpoint that serves as the only point of entry to the sandbox for outside content.

Further development of the RDMT-cloud application will include standard processes and reviews established by the STScI Cloud Center of Excellence (CCoE). These include a formal Systems Development Life Cycle (SDLC), such that version 0.2.0 will be created and tested within development and test isolated environments. Before launch, we will create an `operations` environment that will complete the RDMT-cloud's infrastructure. Though Section 5 primarily focuses on the sandbox environment, we provide peak operational cost estimates for a fully-realized RDMT-cloud in Section 5.8.

In the next six months, we expect the RDMT-cloud to satisfy the CCoE Cloud Readiness Checklist, receive approval to set up a development instance of the AWS infrastructure, and leave the sandbox environment.

5.4 Cloud Application Development

To ensure consistency between various monitor runs, we must create a software environment that defines which versions of our dependencies to install. We attach the environment to a function from AWS Lambda, an event-driven and on-demand service for running prewritten scripts. Though it is possible to natively attach a software environment to a Lambda function via a "Lambda Layer," the entire function cannot exceed either 250 MB uncompressed or 50 MB compressed. The astrometric monitor depends on Python packages like Astropy and Photutils whose larger installation sizes are incompatible with that limit.

The workaround is to locally configure our desired software environment as a Docker container image and upload it to the Amazon Elastic Container Registry (ECR), an AWS service for hosting Docker containers in the cloud. A Lambda function can be built from a container image as large as 10 GB, [1] which is more than enough storage for RDMT's current needs.

The sandbox environment imposes security constraints that prevent direct uploads to ECR, so we must make use of multiple services in order to deploy containerized images as Lambda functions. After uploading the Docker image from a local user's computer to S3, we copy it into an instance of Amazon Elastic Compute Cloud (EC2), AWS' on-demand computing resource. Once it arrives, we decompress it and push the full image to a dedicated ECR repository. We will simplify this process in the future, as shifting from the sandbox to development, test, and production environments will allow properly-credentialed instrument scientists to upload Docker images directly to ECR without the need to involve S3 and EC2.

For now, we use the sandbox's aforementioned VPC, Security Group, and the VPC endpoint that allow outside access to an S3 bucket in order to upload a locally-built Docker

[1] https://docs.aws.amazon.com/lambda/latest/dg/images-create.html#images-reqs

image. Then, we attach an AWS Identity Access and Management role, analogous to a user profile with optional security access controls, that allows access to S3, EC2, ECR, and Lambda to an EC2 instance equipped with an EC2 Instance Connect Endpoint. We connect to the EC2 instance via the AWS console and the AWS Command Line Interface to move the Docker image from S3 to EC2. Finally, we decompress and push the image to an ECR repository. Lambda functions for each monitor will pull the appropriate Docker image from ECR to build software environments where all the packages they require are present.

5.5 RDMT-cloud Monitor Infrastructure

For each monitor, we bundle the corresponding Docker container with the application logic the monitor will use to evaluate incoming files. We choose to write the application logic as Python scripts to maintain compatibility with packages commonly used for astronomical data analysis. These scripts use `Boto3`, the official package for interfacing with AWS services from Python scripts, to interact with the RDMT-cloud AWS environment. The monitor's application logic must ingest the new S3 image and produce a JSON object containing metrics of interest and an HTTP status code indicating a successful response.

Besides the necessary incorporation of `Boto3` and the format of the end result, the application logic can take whatever shape is needed to fit a target use case. For example, our current astrometric monitor depends on Photutils to perform PSF photometry on new images and Astropy's SkyCoord class to cross-match the empirical and expected locations of known guide stars, while the $1/f$ noise monitor relies on Scipy to calculate the image's power spectrum via a fast Fourier transform.

5.6 AWS Version Control

The RDMT-cloud infrastructure is composed of three types of files that call for different version control solutions:

- *AWS resource files*, which can be the scripts that are utilized by the pipeline during execution or that create and configure the resources in the pipeline from scratch.
- *Monitor application logic*, or the analysis script and supplementary files (e.g., `requirements.txt` and a `Dockerfile`) that define a software environment for an individual monitor. (Described in Section 5.5.)
- *Docker images* that bundle the monitor application logic files into a single `.tar` file. (Also described in Section 5.5.)

AWS resource files will remain private and, per CCoE suggestions, will be version controlled using AWS CodeCommit, a cloud-native and git-based version control platform. In the sandbox, hosting the repository on CodeCommit means that our other AWS resources can connect to it, unlike external version control websites (e.g., GitHub). In version 0.1.0 of the RDMT-cloud, this repository contains copies of the ancillary scripts attached to AWS resources in the RDMT-cloud pipeline, such as the Python script for the Lambda function that writes monitor outputs to RDS (see Section 5.7.3).

In version 0.2.0, the RDMT-cloud will be built in an infrastructure as code (IaC) framework and this CodeCommit repository will host the configuration files that function as its source code. An IaC implementation will allow instrument scientists to dynamically edit and programmatically deploy the RDMT-cloud either in part (to test individual components) or in full across multiple AWS environments without interacting with the AWS console. This will become especially useful once future versions of the RDMT-cloud expand to the multi-environment approach described in Section 5.3.

The monitor application logic is intended for public consumption. Since the Docker image configuration process described in Section 5.4 can begin outside the AWS environment, we choose to host these files on GitHub due to developer familiarity and richer functionality for conducting code reviews and managing notifications. GitHub is also more amenable to the public as an online platform since AWS announced earlier this year that CodeCommit is being discontinued for new users. [2]

Though they contain different files, this GitHub repository and the CodeCommit repository of AWS resource files will share version numbers and follow the same release schedule since both are integral parts of the full RDMT-cloud application. This GitHub repository is currently private as we finalize version 0.1.0 of the tool.

Finally, the Docker images are hosted in ECR so Lambda functions for each monitor can build and run them as part of the RDMT-cloud pipeline. Each monitor has its own ECR repository where new Docker images are uploaded whenever the monitor application logic is updated. Though these repositories are private, the application logic used to create them will become publicly available in the GitHub repository mentioned above.

5.7 RDMT-cloud Architecture and Operations

In this section, we walk through a full run of the astrometric monitor from the initial alert through the post to the final database in order to discuss each step in detail. Individual steps are grouped into logical units by subsection that follow the same progression as Figure 4, which visually depicts the end-to-end pipeline.

5.7.1 Data Ingest

We currently mock a notification from the STScI Data Management Divison's Amazon Simple Notification System (SNS) Topic with an SNS Topic of our own. SNS is a service for creating messages to be sent between AWS resources, and a resource can establish a Topic to which other resources can subscribe in order to receive those messages when they are emitted. Whether the posting of the message happens to the Topic happens manually (in the AWS console) or programmatically (through a Lambda test event), the message must contain fields that specify an image file's S3 bucket, S3 object key, and, crucially, the target monitor.

The alert sends a message to one of three input queues from the Amazon Simple Queue Service (SQS), a solution for processing messages between AWS resources. We divide the queues based on the target monitor mentioned in SNS Topic. One queue triggers the astro-

[2]https://thenewstack.io/aws-discontinues-git-hosting-service-codecommit/

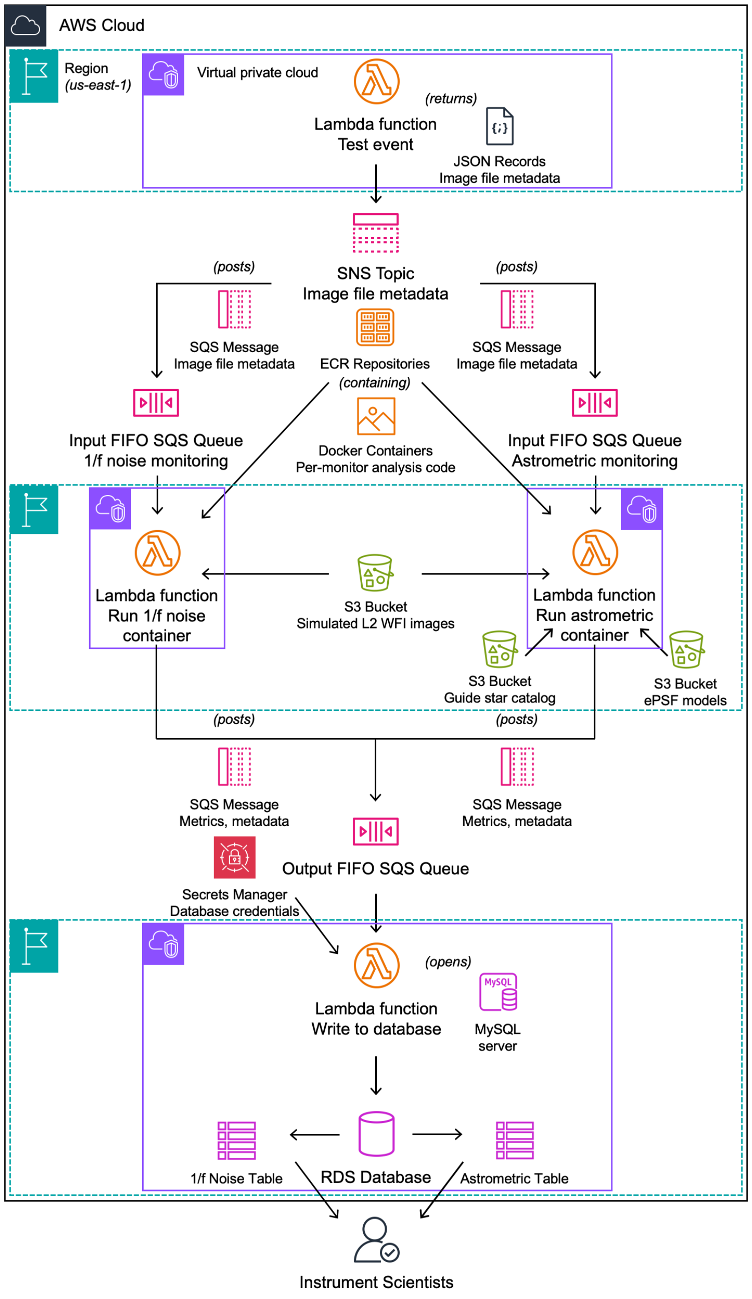

Figure 4: **RDMT-cloud architecture,** or an end-to-end depiction of the AWS pipeline for analyzing a Level 2 WFI image with the $1/f$ noise and astrometric monitors. The content of the initial alert decides which path, or which monitor, is triggered to evaluate an image and write the result to the matching table in the final RDMT database. Though monitors' analysis strategies and supplementary resource needs may differ, all monitors will run Docker containers pulled from ECR and read the indicated image from an official S3 bucket. All steps are automated and triggered by the completion of their immediate predecessor except the initial Lambda test event (which is a stand-in for an automated alert from DMS) and the examination of the final database by instrument scientists. All Lambda functions share the same virtual private cloud (VPC) that allows them to access connected resources through subnets and ingest or write data.

metric monitor's Lambda function, another triggers the $1/f$ noise monitor's Lambda function, and a third handles situations where no monitor is specified by not triggering either monitor. The queues process messages in first in, first out (FIFO) order to avoid confusion in situations where they receive a batch of notifications in short order.

The RDMT will subscribe directly to an SNS Topic from DMS regarding data availability in the Archive once the project progresses to a production environment. Though the structure of the Topic has not yet been finalized, we expect that it should include attributes that allow us to distinguish files (e.g., WFI observing mode and calibration level). Though the RDMT-cloud v0.1.0's monitors focus on L2 imaging data, we imagine that a fully-fledged RDMT-cloud may also monitor Level 2 spectra products and other levels of imaging data. The RDMT team will work with DMS to define an extensible SNS schema to direct these messages as RDMT's number of monitors continues to grow.

5.7.2 Image Analysis and Metrics

A new entry to a queue associated with an active monitor will trigger the appropriate Lambda function to analyze the new file via a Python script. These Lambda functions are built from the Docker images uploaded to ECR repositories that were described in Section 5.4.

The prototype astrometric monitor's Python script calculates the average offset, in arcseconds, between simulated stars' expected locations in a given WFI image and their apparent locations as derived from PSF photometry. Currently, the simulated stars are from a fixed catalog, but in the future they will either be the astrometric reference stars within each image or will be queried from the Guide Star Catalog. These decisions are still being finalized. Once the calculation is made, it sends a message to an output SQS queue that handles messages from either monitor. All messages to the output SQS queue will carry the file metadata from the SNS Topic that initiated the run and the values meant to be saved to the final database. For the astrometric monitor, these additional attributes are an integer describing the number of guide stars found in an image and a float representing their average astrometric offset from expectation. (The $1/f$ noise monitor produces a single Boolean describing whether it judges the reference pixel correction was successful.)

Once it has posted to the output SQS queue, the Python script must return a 200 status code to indicate a successful run. The script will also explicitly catch commonly-encountered errors (e.g., a timeout while attempting to fetch an image from S3 via `Boto3`) and include a procedure for retries should they occur instead of stalling the pipeline.

5.7.3 Monitor Outputs

Just as each monitor's Lambda function for image analysis is triggered by an input SQS queue, the Lambda function that writes monitor results to the RDMT-cloud database is triggered by the output SQS queue.

Although the image analysis code must use the `Boto3` API to send a message to the output SQS queue, the "write-to-RDS" Lambda function already comes with the SQS message's content bundled into its `event` parameter, so there's no need to explicitly call it from the queue. The message is also automatically removed from the queue after the successful execution of the "write-to-RDS" Lambda function.

File	offset [arcsec]	n_sources
r0000101001001001001_01101_0001_WFI01_cal.asdf	0.1459	35
r0000101001001001001_01101_0001_WFI02_cal.asdf	0.1194	52
r0000101001001001001_01101_0001_WFI03_cal.asdf	0.1081	38
...	...	...
r0000101001001001001_01101_0001_WFI18_cal.asdf	0.1513	44

Table 3: **RDMT-cloud Astrometric Table example.** The RDMT-cloud stores the filenames, mean astrometric offsets, and number of detected point sources in the Astrometric Table within RDS.

The output database is hosted in the Amazon Relational Database Service (RDS), the primary service for managing relational, tabular databases in AWS. After gaining the proper credentials from a secret stored in the RDMT-cloud's AWS Secrets Manager, the "write-to-RDS" Python script connects to the database via MySQL with `pymysql` and writes the latest results there. The database contains one table per monitor, so the "write-to-RDS" Lambda function targets the proper table based on the image file metadata present in the SQS message that triggered it. See Table 3 for an example of how the astrometric monitor's table might look in RDS.

Unlike the RDMT-local, the RDMT-cloud does not currently save any metrics or evaluators to files beyond the output database. We may add this functionality as we continue to expand the RDMT-cloud's capabilities in future releases.

5.8 AWS Cost Estimates

This section details expected costs for each component of the operations instance of the RDMT-cloud during the nominal 5-year mission. We aim to provide an estimate for the maximal monthly budget we expect to occur. To overestimate the number of invocations of the RDMT-cloud, we assume that we will run 20 monitors on all Level 2 files created within a 31 day month during a later executions of the Design Reference Mission (DRM) Galactic Bulge Time Domain Survey (GBTDS) near the end of the nominal mission. This is only an approximate estimate; the exact implementation of the GBTDS is subject to change.

The GBTDS will strain the maximum data volume that can be transmitted from the telescope to the SOC in a single day. In the DRM, each pass consists of 7 visits and is revisited on a 15-minute cadence, resulting in 18 new images (one per detector) created roughly every 2 minutes for upwards of 68 days straight. Though these observations may be interrupted by calibration campaigns and periods of data downlinking, we neglect any science data observing interruptions to predict the scenario with the largest number of science files produced to generate a maximum possible budget. The resulting rapid cadence would result in 401,760 Level 2 files created per month. We will use this upper bound for the expected number of files in our maximal costing analysis below. It is important to note that this estimate does not include monitoring of any reprocessed data produced during the data releases. At this moment, there is not a plan for monitoring data releases other than acknowledging that some monitoring of release products will be needed. These decisions

AWS Service	Maximal Monthly Cost	Typical Monthly Cost
RDS	$171 [a]	$171
Lambda	$4,016 [b]	$1,550 [c]
S3	$71 [d]	$36 [e]
SQS [f]	$8	$3
ECR [g]	$6	$6
CodeCommit [h]	$5	$5
VPC	$0	$0
Total	**$4,277**	**$1,771**

[a] Using a single `db.t4g.medium` instance with 100 GB storage with 300 GB of backup storage.
[b] Assuming 18 files created every 2 minutes for a 31 day month (e.g., during the GBTDS) with 20 monitors running on each file, or 8,035,200 Lambda function executions.
[c] Assuming 18 files created every 5 minutes for a 30 day month with 20 monitors running on each file, or 3,110,400 Lambda function executions.
[d] Assuming 3 TB of storage is used (which is unlikely to happen over the 5 year nominal mission).
[e] Assuming half of the 3 TB estimate is used. AWS S3 is a pay-as-you-go service so this is a decent estimate for the nominal mission.
[f] Assuming 2 requests per monitor run: one to trigger the monitor and another to store the information in the database.
[g] Assuming each monitor hosts three unique Docker images that are each 1 GB in size.
[h] Assuming 10 users need access to CodeCommit.

Table 4: Estimated maximal and typical monthly costs for the RDMT-cloud operations instance during the nominal 5-year mission.

will be made during the development of version 0.2.0 of the RDMT-cloud. As we get closer to launch, we will also revise this estimate to better reflect the maximum expected file generation per month.

Table 4 includes a summary of this budget and an approximation of the typical monthly costing of the operations instance assuming a new file is delivered every 5 minutes. A quote containing exact specifications for this budget can be found at the AWS Pricing Calculator. [3] We are only providing the budgets for the production instances of the RDMT-cloud because the costs of the development and testing instances will be dominated by the databases and will be ≃$200 each, which is much less than production.

We expect the majority of the RDMT-cloud's costs will be generated by RDS, S3, and, during operations, Lambda. The following estimates assume that all components operate in the AWS `us-east-1` Region, are accurate as of September 2024, and do not include any enterprise discounts that STScI receives through its contract with AWS.

RDS pricing depends on the selections made for storage capacity, computation capacity, and the volume of data egress from the AWS cloud.

- *Primary database storage:* Version 0.1.0 of the RDMT-cloud uses 20 GB of `gp2`-class General Purpose SSD storage with a MySQL database engine. Combined with the selections from other RDS categories, this makes it eligible for the AWS Free Tier. During

[3] AWS Pricing Calculator: https://calculator.aws/#/estimate?id=e3b639a9728f1c908d779730cc5f1d17f109f9f6

the mission, we expect to maintain development, test, and production databases. At that time, we anticipate we will upgrade the production database to 100 GB of `gp3` storage in order to comfortably fit tables for more monitors and increase read/write efficiency through `gp3`'s higher throughputs. [4] `gp3` storage costs $0.115 per GB per month. The development and operations databases will likely be much smaller (and thus cheaper) than their production counterpart.

- *Backup database storage:* AWS offers free backup storage up to the amount of primary storage allocated in the previous step. Afterward, backup storage costs $0.095 per extra GB per month. We don't anticipate needing more backup storage than 20 GB for version 0.1.0 of the RDMT-cloud. During the mission, we estimate that we will require 200 GB of backup storage for the production database.

- *On-demand database instance type:* Version 0.1.0 of the RDMT-cloud uses a `db.t3.micro` instance over a single Availability Zone (AZ) to handle computation. This is a lighter-weight and "General Purpose" [5] instance that is eligible for the AWS Free Tier. During the mission, we anticipate using a `db.t4g.medium` instance for the production database, which comes with more memory, a higher baseline for CPU usage, and, if needed, increased runtime at above-baseline CPU usage before incurring extra costs. [6] Additionally, we plan to use multiple AZs for the production database to introduce fault tolerance in unexpected scenarios where the primary database is not available. With a `gp3` volume, AWS allows up to 3,000 In/Out operations per second (IOPS) and 125 MiBps of throughput per month. Surpassing these thresholds in a given month costs $0.02 per IOPS and $0.080 per MiBps.

- *Data transfer:* Data transfer into RDS is free. Customers may transfer up to 100 GB of data out of an AWS environment (across all services and regions) to the internet for free. After 100 GB, data egress costs as much as $0.09 per GB. However, we don't expect the RDMT-cloud to challenge this limit in any given month.

Storage in Standard S3 buckets, which are the recommended class for frequently accessed data, costs $0.023 per GB per month for the first 50 TB of data. Version 0.1.0 of the RDMT-cloud's total storage needs are on the order of tens of gigabytes. We will need more storage in production, but our current maximal estimate of 3 TB of total storage still sits well within the first tier.[7] It is free to upload data, transfer data between S3 buckets in the same Region, and transfer data between AWS services in the same Region. Data egress is subject to the global 100 GB limit mentioned with RDS, though we don't expect to export data from S3.

ECR offers 50 GB of free storage per month for public repositories and otherwise charges $0.10 per GB of storage per month. The RDMT-cloud's ECR repositories are currently

[4] https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-storage-compare-volume-types.html

[5] https://aws.amazon.com/rds/instance-types/#General_Purpose

[6] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/burstable-credits-baseline-concepts.html#earning-CPU-credits

[7] 3 TB of storage would allow each monitor to output several super-pixel images into save files for every file that is checked.

private, but given that a typical Docker image size for both monitors is less than half a gigabyte, we expect ECR's contribution to the RDMT budget to be minimal. The private repositories in version 0.1.0 of the RDMT-cloud cost less than $2 per month. During the mission, if Docker image sizes remain the same, we can host up to five Docker images in each monitor's public repository and remain within ECR's free tier. It is free to transfer data into any repository and between ECR and other AWS services in the same Region. Data egress is not free, but we do not expect to regularly transfer data out of ECR.

CodeCommit is free for up to five users, 5,000 repositories, 50 GB per month of storage, and 10,000 Git requests per month. Though we expect the RDMT-cloud's repository count and associated storage to fall below this limit, we have assumed 5 extra users in the cost estimate in case the team grows to more than 5 developers. Each additional active user costs $1 per month and adds another 10 GB of storage and 2,000 Git requests.

SNS costs are calculated based on the number of notifications emitted and are free up to 1 million standard monthly requests. Subsequent requests cost $0.50 per million. While version 0.1.0 of the RDMT-cloud comes in well below the 1 million message threshold, the maximal GBTDS scenario of 401,760 files passing through 20 monitors generates closer to 10 million messages in a month. These costs will mostly be covered by DMS, as they will post the notifications. However, coordination between the RDMT team and DMS is needed to ensure that additional message filtering costs are not accrued. That coordination is currently underway.

AWS also permits up 1 million free FIFO SQS requests per month. FIFO requests cost, at most, $0.50 per million per month in subsequent tiers. Again, we don't expect to approach this limit with version 0.1.0 of the RDMT-cloud. The maximal GBTDS scenario does surpass the free limit since there are two SQS requests to a monitor's input and output queues for every SNS request. However, the relatively low rate makes SQS is another low-to-no-cost service for the RDMT-cloud.

The RDMT-cloud's Lambda functions share a virtual private cloud (VPC) that routes incoming and outgoing traffic through "Gateway" VPC endpoints. As long as all endpoints are of type Gateway, AWS does not charge for the time the VPC spends directing traffic or the amount of data it processes. (This is not the case for other types of endpoints and gateways.) While Gateway endpoints are only available for traffic to and from S3 and DynamoDB, this is workable for the RDMT-cloud because all other resources that need to communicate are already on the VPC.

Lambda costs are calculated based on the number of monthly requests and the amount of memory resources consumed during those requests. Requests cost $0.20 per million per month, and the first 6,000 petabyte-seconds of memory resources consumed per month cost $16.6667 per petabyte-second. If a typical monitor conducts its analysis on a single image file over 30 seconds and consumes an average 1 GB of memory per second, a run costs $0.0005, or 1/20 of a penny. Expanding to 1,000 runs per detector per month, as we might for version 0.1.0 of the RDMT-cloud, generates less than a penny of costs for the number of requests and $9.00 of computation costs while only consuming 0.54 petabyte-seconds of the first-tier allotment. In the GBTDS scenario of 20 monitors and 401,760 images, the requests generate less than $2.00 of costs while the computation costs grow to about $4,000. This maximal scenario remains within Lambda's first tier by consuming about 241 petabyte-seconds of memory resources.

6 Summary and Future Development

The RDMT is a critical tool for ensuring the calibration quality and usefulness of data from the Roman WFI. By enabling both on-premises and cloud-based monitoring capabilities, the RDMT provides a comprehensive solution for rapid detection and resolution of data quality anomalies throughout the processing pipeline. This tool will play an important role in maximizing the scientific return of the Roman mission by enabling efficient data calibration consistency assessments and rapid responses by the instrument team.

The RDMT is designed to rapidly monitor data metrics and produce automated notifications, but it is only one out of many applications that will enable the SOC to diagnose and remedy issues in the processing pipeline. To ensure the WFI's successful operations, instrument scientists will need to combine RDMT-local and RDMT-cloud outputs with other calibration software outputs and engineering data. On a best-effort basis, we hope to implement a single online dashboard that will serve as a "one-stop shop" for all the instrument scientists' needs. Using the engineering and calibration software outputs, the dashboard would combine information from both arms of the RDMT with the auxiliary data and allow for correlations to be easily identified.

Though all current functionality is intended to be operated and contained within STScI holdings, there is significant need for calibration analysis to be shared with the greater community. For example, the SSC has expressed interest in receiving notifications of poor calibration consistency prior to their processing of the spectroscopic mode WFI data. In addition to the SSC, the PITs would also benefit from receiving warnings about calibration quality prior to running Roman data through their unique pipelines and determining their own scientific data quality assessments. The recipients of such notifications need not stop at direct project stakeholders, but could be expanded to community members who would benefit from the ability to subscribe to the RDMT notifications.

Additionally, both arms of the RDMT are designed to be easily expandable, allowing for additional monitors and features to be added to the RDMT over time as new problems or ways of evaluating the calibration arise. We also anticipate that our SDLC will update as cloud security and software development best practices will continue to evolve according to recommendations from the STScI CCoE.

References

Bourque, M., Bajaj, V., Bowers, A., et al. 2017, in Astroinformatics, ed. M. Brescia, S. G. Djorgovski, E. D. Feigelson, G. Longo, & S. Cavuoti, Vol. 325, 397–400, doi: 10.1017/S174392131601276X

Bourque, M., Chambers, L., Cracraft, M., et al. 2020, in Astronomical Society of the Pacific Conference Series, Vol. 527, Astronomical Data Analysis Software and Systems XXIX, ed. R. Pizzo, E. R. Deul, J. D. Mol, J. de Plaa, & H. Verkouter, 364539

Ferguson, H., Sosey, M., Snyder, G., & Kolatch, E. 2022, Roman Science Operations Center Wide Field Instrument Data Processing Summary, Nancy Grace Roman Space Telescope Technical Reports, 18 pp.