ORNL/TM-2024/3621
M2CR-22OR0403052

FY 2024 Multidimensional Data Correlation Platform: Unified Software Architecture for Advanced Materials and Manufacturing Technologies Data Management and Processing

Stephanie Cooper
Gina Accawi
Abby Barnes
Wes Foreman
Greyson Jones
et al.

September 2024

OAK RIDGE National Laboratory

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2024/3621
M2CR-22OR0403052

Advanced Materials and Manufacturing Technologies Program

FY 2024 MULTIDIMENSIONAL DATA CORRELATION PLATFORM: UNIFIED SOFTWARE ARCHITECTURE FOR ADVANCED MATERIALS AND MANUFACTURING TECHNOLOGIES DATA MANAGEMENT AND PROCESSING

Stephanie Cooper
Gina Accawi
Abby Barnes
Wes Foreman
Greyson Jones
William Halsey
Vladimir Orlyanchik
Vincent Paquit
Jeffery Price
Luke Scime
Zackary Snow
Annika Wille
Amir Ziabari

September 2024

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-BATTELLE LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AD	Active Directory
AMMT	Advanced Materials and Manufacturing Technologies
API	application programming interface
CUI	Controlled Unclassified Information
DOD	US Department of Defense
EDM	electrical discharge machining
JSON	JavaScript object notation
LPBF	laser powder bed fusion
M:M	many-to-many
MDDC	Multidimensional Data Correlation
MDF	Manufacturing Demonstration Facility
MVC	model–view–controller
ORNL	Oak Ridge National Laboratory
QR	quick response
TCR	Transformational Challenge Reactor

ABSTRACT

The Advanced Materials and Manufacturing Technologies (AMMT) program is exploring a data-driven approach called the Multidimensional Data Correlation (MDDC) platform to demonstrate the use of additive manufacturing for fabricating components for nuclear applications. One of the program's main scientific goals is to use data to provide a greater understanding of manufacturing outcomes to improve the performance, reliability, and lifespan of nuclear components. If successful, this approach will lead to the development of standards for the certification and qualification of additively manufactured components, which are needed by industry for broader adoption of these technologies.

To support this objective, this work package is building a data management platform to record, index, analyze, and make available the manufacturing data generated by all members of the AMMT program. In FY 2023, this work package conceptualized the architecture of the unified software platform, centered around the concept of operations and trackables, to capture the data. This year, the project implemented this vision and deployed a functional platform at the Oak Ridge National Laboratory (ORNL) Manufacturing Demonstration Facility (MDF). This report describes its implementation and functionalities and provides an example of its usage with laser powder bed technology.

This report marks the completion of FY 2024 milestone M2CR-22OR0403052: Demonstration of the Use of the New Digital Platform at the MDF on a Laser Powder Bed System.

1. INTRODUCTION

The MDDC platform is the data-driven initiative sponsored by the AMMT program to support the development of a novel approach for certification and qualification of additively manufactured components for nuclear applications. The success of the MDDC relies strongly on the creation and availability of comprehensive datasets that can support multiple areas of development across each manufacturing process that is considered. This ambitious project requires collective contributions from various AMMT members to ensure the relevance and completeness of the dataset that is needed to support conclusions and advance this field of manufacturing.

To facilitate this data collection effort, this work package focused on building a custom data management platform over the past 2 years. The first year was dedicated to conceptualizing a platform that ensures that the relationships between manufacturing steps, feedstock materials, postprocessing, and testing and characterization are preserved. The platform also ensures that the associated data streams can create a digital twin of the fabricated components. The second year is described in this report. The second year focused on the implementation and testing of the platform using data from laser powder bed additive manufacturing technologies.

This report first provides an overview of the design of the new digital tool, Damara Tern, in Section 2, mixing information from FY 2023 and FY20 24 activities. Section 3 demonstrates the implementation of the new platform using data from powder bed systems to complete this milestone.

2. CONCEPT OF THE NEW DIGITAL PLATFORM DATABASE AND INTERFACE FRAMEWORK

This section was, in part, extracted from a previous report [1]. The authors updated the section when needed to reflect recent changes. The authors thought it was appropriate to have this section in this report as well for context and easy comparison with the concept and implementation.

Damara Tern is the new framework developed to access the MDF Digital Platform data. Its primary function is to structure and organize the metadata collected along the digital thread and to serve as a library, allowing users to save, explore, and retrieve the data associated with each operation and component (i.e., trackable) involved in the manufacturing process. The framework succeeds the MDF Digital Tool and related relational database initially deployed under the Transformational Challenge Reactor (TCR) program for accessing and exploring the data stored within the MDF Digital Platform. Although its predecessor successfully serves this function and enables component tracking during the manufacturing and testing processes, it relies on a printer-centric model that induces specificity and poses significant limitations in terms of extensibility and automation. To overcome these limitations, the Damara Tern framework relies on a generic and flexible operation-centric structure that accommodates a diverse range of machines, manufacturing and testing operations, and trackable components. Additionally, the framework aims to enhance search and view functionalities that reflect the digital thread for each manufacturing process.

2.1 IMPLEMENTATION CONSIDERATIONS

The main objective of the MDDC framework is to provide a unified data management platform that is accessible to all members of the AMMT program. This framework should enable (1) nationwide tracking of physical components and digital assets across multiple sites and (2) the creation of a database adhering to the findable, accessible, interoperable, and reusable best practices to support the research and development activities of the program. In contrast to the implementation for the TCR program, the platform described here requires using a data management strategy that preserves the integrity of the digital thread of each component across multiple physical locations. This important requirement resulted in the following considerations.

- **Database location:** Data will not be hosted at a single location, nor will there be replicas of the complete database at each national laboratory. Instead, AMMT participants will host a database of the information they will produce, and those databases will be interconnected using a cross-referencing mechanism described in Section 2.2.

- **Database access:** Initially, users will access the database using credentials provided by the host national lab. As new data exchange and linkage functionalities of the platform are developed, web tools will be provided that allow direct access to data regardless of geographic location.

- **Information retrieval:** Rarely, users will need access to the entire database to work on scientific problems. In this case, they will retrieve subsets of the data based on scoped queries, for which application programming interface (API) functionalities will be developed to make intelligent and focused queries to the database.

- **Knowledge extraction and data visualization:** The selected Django framework provides access to numerous Python libraries for advanced data visualization and processing. Domain experts will be expected to create their own data processing pipelines to access the database and format the data for their specific applications.

2.2 OPERATION-TRACKABLE DATA MODEL

The data model employed for the development of the framework revolves around the concepts of operations and trackables, which are defined as follows:

- **Operation:** Any action performed via the use of a machine (e.g., print, cure, heat treatment, blue light scan, microscopy analysis, tensile test) or human interaction e.g., (procurement, annotation, registration, manipulation). Operations can be performed prior to, during, or after the manufacturing process itself to extend the data context.

- **Trackable:** Any physical or digital component that can be subjected to an operation (e.g., parts, builds, feedstock materials).

This model relies on the manufacturing process flow and provides the necessary framework for storing and organizing metadata collected along the digital thread. The database is implemented along this model, in which each physical or digital component (i.e., trackable) undergoes a series of tests or actions (i.e., operations). Each operation supports the collection of metadata or substantial in situ data and can lead to the creation, alteration, combination, or transformation of the trackables.

By collecting the data at each operation level and preserving the traceability of relationships between the trackables and applied operations, recreating the digital thread for each trackable becomes possible. This pathway reflects the entire manufacturing process and is accompanied by the comprehensive context of collected data and metadata gathered throughout the entire process.

Figure 1 gives an example of the operation/trackable-based model of the digital thread.

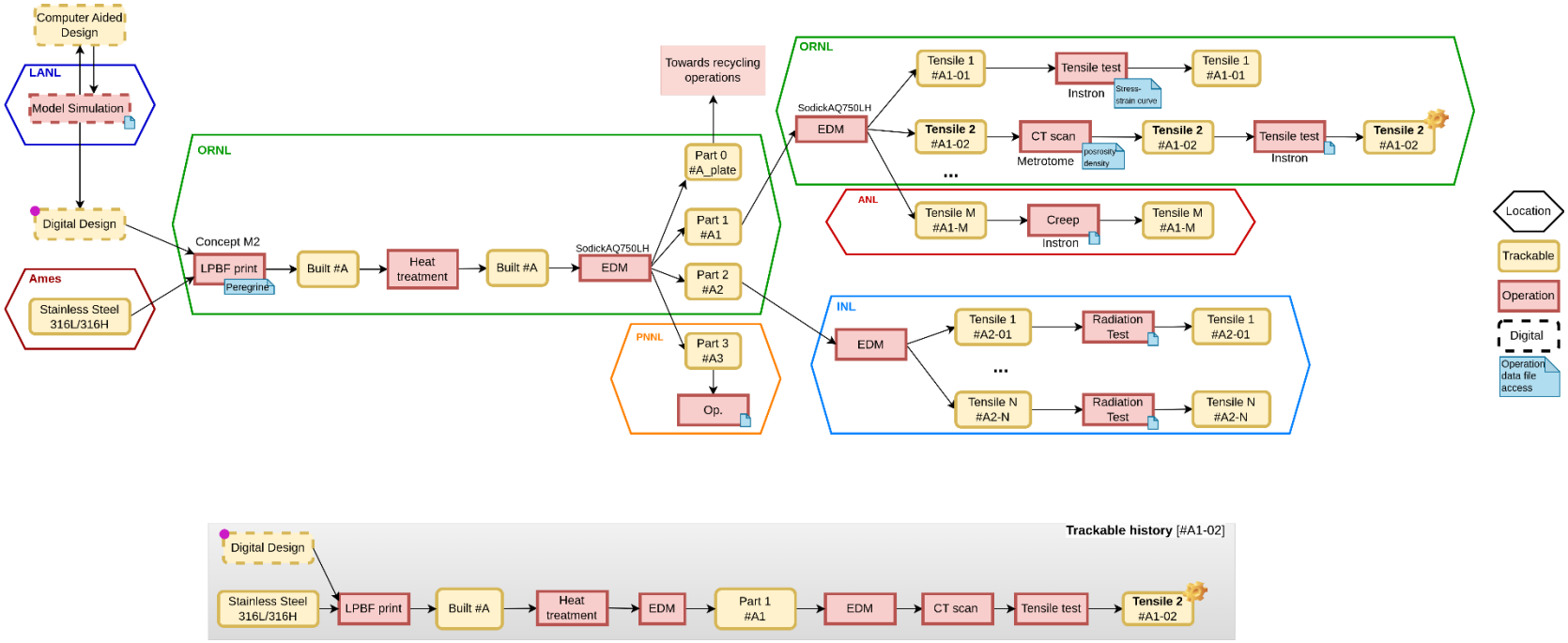


Figure 1. Representation of a multisite digital thread in the context of the operation-trackable data model. In this example, the trackable specimen "Tensile 2" history of operations and parent trackables are highlighted. The trackable's entire creation context can be retrieved by retracing this history and gathering the data and metadata collected along each operation. EDM stands for electrical discharge machining, and LPBF stands for laser powder bed fusion.

2.3 DAMARA TERN FRAMEWORK: SOFTWARE AND TECHNOLOGY

The Damara Tern framework combines the underlying MDF metadata database and a web interface for exploring, entering, and accessing data. It is implemented using the Python Django web framework. Django websites rely on a database and use model files that describe the data and provide automatically generated database-access APIs. To preserve the portability of the metadata collected for the Digital Platform, this project used distinct databases (or schemas) to store the Django-specific data and the data collected for the Digital Platform. This section presents an overview of the MDF database structure used for the Digital Platform metadata collected and the Django project architecture.

2.3.1 Database Implementation

The original database for the Digital Platform was printer-centric, with each build recorded in a machine-specific PostgreSQL schema. Although this setup allowed for detailed, machine-specific data, it lacked extensibility and automation, requiring manual schema creation for each new machine. To overcome this limitation, the new database architecture built upon the operation-trackable data model described in Section 1 follows a semistructured approach, using generic fields in tables and JavaScript object notation (JSON) fields for fields specific to equipment, materials, and operations.

The database is implemented using PostgreSQL, which was chosen initially because of the development team's existing knowledge and experience with it. Although alternatives such as MariaDB or MySQL were considered, PostgreSQL's JSON capabilities have since proven valuable for performance enhancements. Although switching to another relational database is feasible with minor modifications, the authors of this report recommend continuing with PostgreSQL for its benefits.

Figure 2 presents an overview of the database structure divided into five logical blocks: Operation, Trackable, Machine, Related Data, and People and Affiliation. Each block comprises the main tables responsible for providing information within its designated scope.

- **Operation:** Represents performed operations, including their types and specificities that define the contexts in which they are carried out.
- **Trackable:** Encompasses catalogued physical or digital components subject to operations, along with the types and specifications used for their categorization.
- **Machine:** Enumerates printers and other pieces of equipment employed for operations, including contextual details such as calibration and dedicated features.
- **Related Data:** Involves information pertaining to substantial process data (e.g., images, videos, documents) stored within file systems or accessible remotely via uniform resource locators (URLs). These data are essential for retrieval and access purposes.
- **People and Affiliation:** Encompasses all information regarding users, institutions, and projects required to manage data access and affiliations.

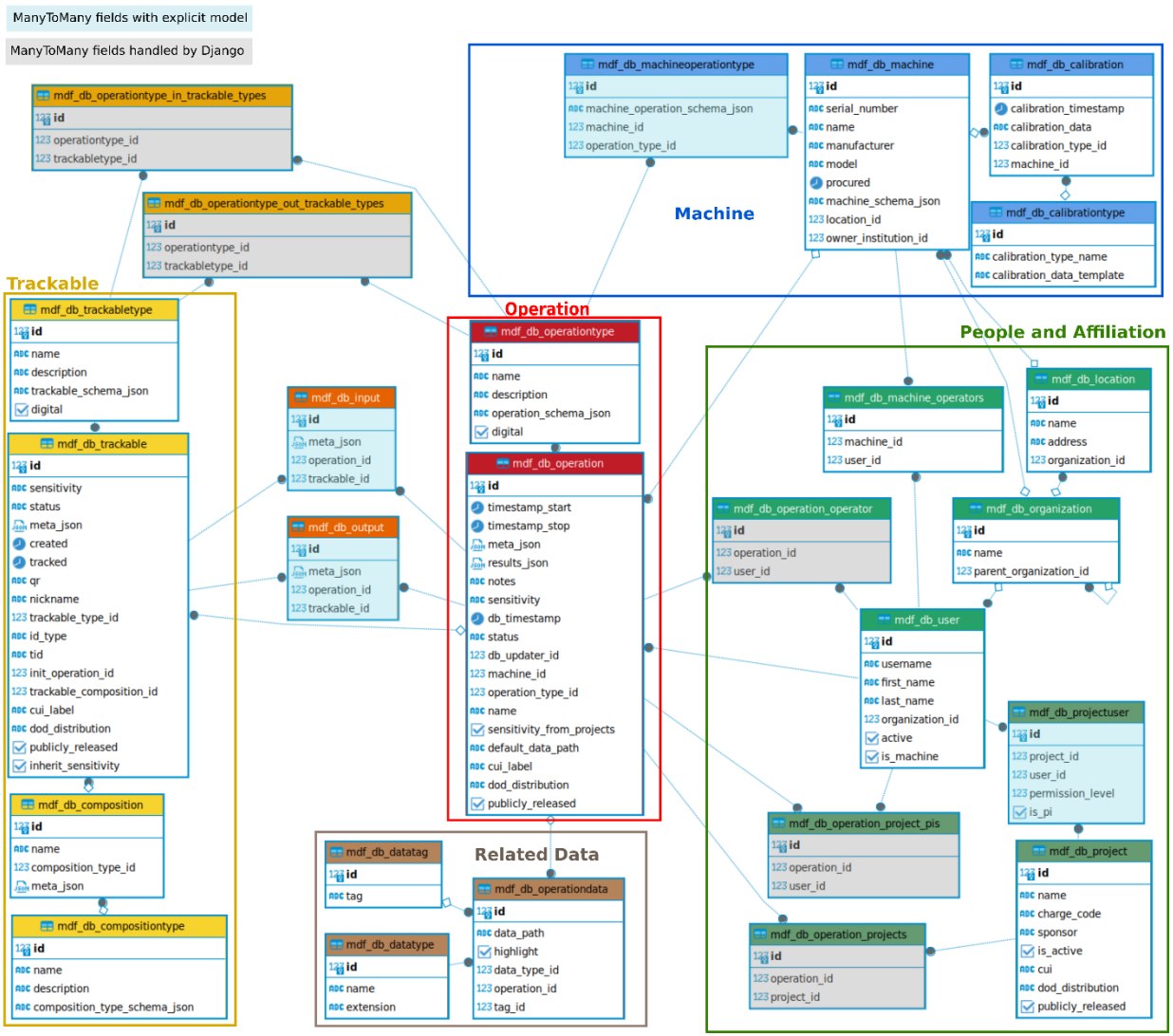


Figure 2. Overview of the database structure organized into five logical blocks: Operation, Trackable, Machine, Related Data, and People and Affiliation.

The following is a list of the primary tables and their functions corresponding to each logical group. We use colored text to help the reader find the corresponding database tables in Figure 2.

- **Operation:** Stores operation records and related metadata.
- **Operation_Type:** Names and describes the supported operation types and provides form templates for operation-specific metadata collection.
- **Trackable:** Stores trackable records and related metadata.
- **Trackable_Type:** Names and describes trackable types and provides form templates for trackable-specific metadata collection.
- **User:** Lists the MDF Digital Platform users.
- **Organization:** Lists the institutional organizations for the users.
- **Location:** Defines locations and addresses for organization and vendor sites.
- **Project:** Lists the projects and sponsors supporting the purchases and manufacturing operations.
- **Machine:** Identifies machines and equipment, along with their related metadata.
- **Calibration:** Records details of the machine and equipment calibration over time.
- **Calibration_Type:** Provisional table used to categorize calibrations and provide form templates for calibration-specific metadata collection.
- **Related_Data:** Records the data paths, URLs, or references to the collected data stored remotely or on a file system.
- **Data_Type:** Defines the data type referenced by the Related_Data entries.
- **Data_Tag:** Provides additional contextual tags to categorize the data referenced by the Related_Data entries.

As mentioned previously, Django interfaces with the database through tables and field definitions declared within a model file (models.py). Database structure alterations can be made by modifying the Django model and running migrations to propagate the changes. If not previously instantiated, the entire database structure can be created by running an initial migration. Although the Django model interfaces with the database, the structure itself was designed as a stand-alone component and does not require the use of the Django framework. The stored data could be saved or exported separately and possibly accessed via a different interface.

In addition to the main tables listed in this section, the database contains several association tables that store the many-to-many (M:M) relationships required by this data model design. Simple tables (highlighted in gray in Figure 2) are not explicitly declared in the Django model but are managed by the framework upon an M:M field declaration. Each of these tables includes three fields for each entry: a primary key ID and a foreign key for each of the tables involved in the relation. Other association tables (Input, Output, Project_User, and Machine_Operation_Type) have been explicitly declared to include additional information along with the M:M relationship.

2.3.2 Django Architecture

Django was chosen to implement the Damara Tern user interface because of its out-of-the-box features, versatility, utilization of Python as the implementation language, and proven success in prominent projects (e.g., Instagram, Mozilla, National Geographic). It is based on a model–view–controller (MVC) architecture, and the project is built upon a file structure organized into “apps.” Each app is thoughtfully designed to partition the project by core functionalities. The current project structure, illustrated in Figure 3, consists of five custom Django apps: mdf_db, operation, trackable, home, and api (note that api is not shown in Figure 3). Each app is discussed individually in the following sections.

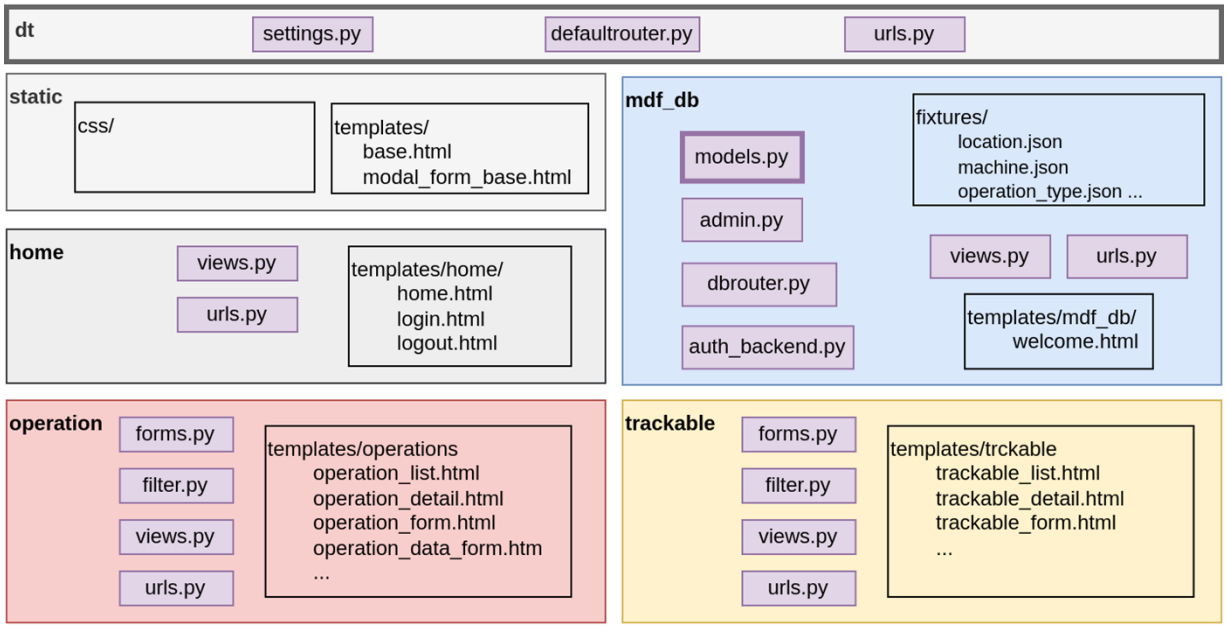


Figure 3. Damara Tern web interface project structure. The files are structured around the concept of apps, organizing the project by core functionalities.

2.3.2.1 mdf_db

The mdf_db app handles the interfacing and connection with the metadata database described in Section 2.3.1. It provides the Python declarations for the database tables and fields, along with database-level constraints and validation rules. As the component responsible for the data structure API, the other apps relying on the data stored (operation and trackable) depend on it. This app also defines the list fields and forms accessible through the admin interface and supports the ORNL Lightweight Directory Access Protocol (LDAP) authentication backend. Table 1 presents the file structure within the mdf_db app. It lists the main files contributing to the Damara Tern implementation and describes their purpose.

Table 1. mdf_db file structure

File	Description
admin.py	Defines the models and fields accessible through the admin interface. All the tables and fields a user wishes to interact with in the admin menu should be declared here.
auth_backend.py	Defines the authentication back end, including the LDAP authentication.
models.py	Defines the database structure presented in Section 3.1. Declares the tables and fields, defines all metadata of the database, and determines the behavior of each field.
fixture/*.json	Provides Django fixture, or the predefined set of data that can be imported into the database. Warning: The elements come with predefined primary keys. A fixture import in a nonempty database is likely to overwrite records.
dbrouter.py	Defines the database router for the model (required because this project uses several databases). The router ensures that the model objects are properly stored/retrieved from/to the mdf_data database rather that from/to the Django default database.

2.3.2.2 operation

The operation app encompasses all the functionalities and classes required to manage the operation entries of this database model. This app includes various forms for creation, views for listing and displaying, and search filters that allow navigation, viewing, and retrieval of the operations and their related stored data.

Table 2. operation file structure

File	Description
views.py	Defines the operation views, including the operation list, and details views presenting operation content, create and update views to access content creation forms, and OperationData views used within modal form content creation and to link the data related to operation. It also includes a set of non-class-based views for related data display, can download functionality, and is used for dynamic HTMX updates.
forms.py	Provides the forms definition for Operation and OperationData creation.
filter.py	Defines the operation filter used within the operation list view.
urls.py	Provides the view-URL mapping for the operation app.
templates/operation/*	Provides all the templates for Operation and OperationData content (e.g., operation_detail.html, detail_content.html, operationdata_view.html, operation_list.html, related_data.html, mat1.html), and the forms (e.g., operation_form.html, operationdata_form.html, operationdata_link.html, operationdata_modalform.html, operation_delete.html). Some templates, such as related_data.html, are referred to by the trackable app.
templatetags/extras.py	Defines custom template tags for the operation templates.

2.3.2.3 trackable

The trackable app provides similar functionalities to the operation app, including creation forms, list and detail views, and filter-based search capabilities tailored for the management of Trackable entries.

Table 3. trackable file structure

File	Description
views.py	Defines the trackable views, including the trackable list, detail, creation, and update views. Also provides the quick response (QR) code scanner view and non-class-based views for dynamic HTMX updates.
forms.py	Provides the forms definition for the Trackable creation.
filter.py	Defines the trackable filter used within the operation list view.
urls.py	Provides the view-URL mapping for the trackable app.
templates/trackable/*	Provides all the templates for Trackable content (e.g., composition_detail.html, trackable_detail.html, trackable_list.html) and forms (e.g., trackable_form.html, scan_form.html).
templatetags/custom_filters.py	Defines custom template tags for the trackable templates.

3. DESIGN OF THE NEW DIGITAL PLATFORM DATABASE AND INTERFACE FRAMEWORK

Damara Tern is the name of the new framework for accessing the MDF Digital Platform data hosted at the MDF (main page displayed in Figure 4). Following the concept described in Section 2.2, the development team has created a web-based platform for submitting and retrieving data generated during the program. This section describes each web page using individual subsections. This report provides detailed explanations of their functionalities and how they connect with the concepts of operations, trackables, and the digital thread overall.

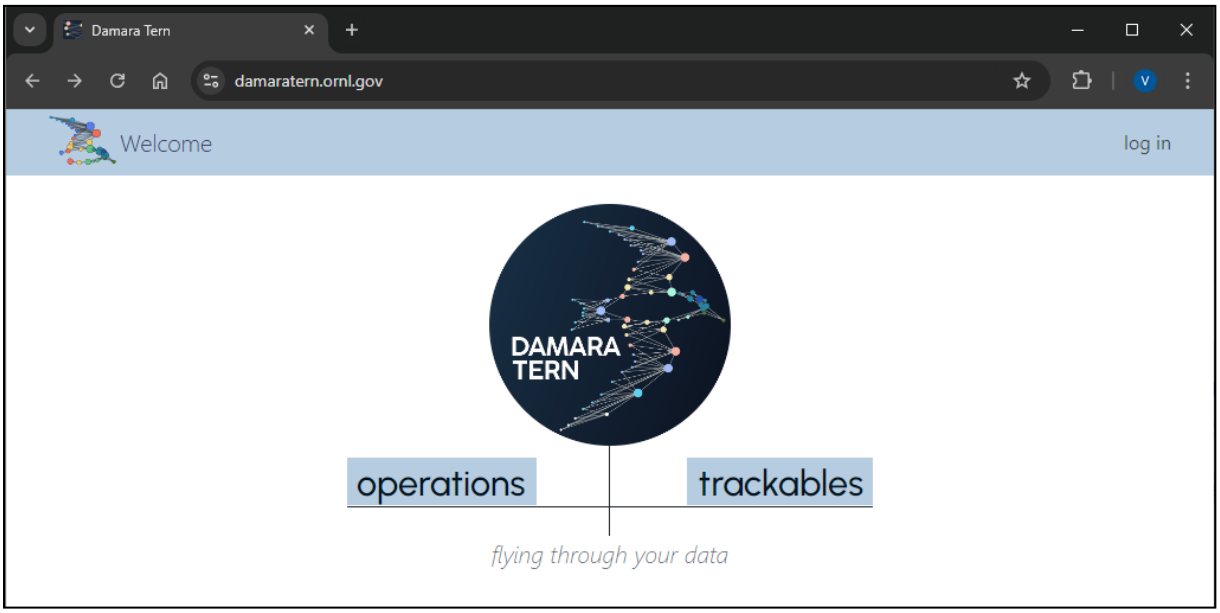


Figure 4. Main page of the digital platform called Damara Tern.

3.1 LOGIN PAGE

The MDF Digital Platform was designed to offer flexibility and adaptability for the various manufacturing processes investigated at the MDF, other national labs, and close collaborators from academia and industry involved in the AMMT program. Data produced across the program will have varying levels of sensitivity and access requirements based on the nature of the work, making access control essential. The concept of projects will be discussed in Section 3.3, but first, the login page provides prescreening of users, allows for the creation of sessions that restrict access, and tailors the information displayed on each web page or web form. This login page is shown in Figure 5.

Currently deployed at ORNL, the login credentials use Active Directory (AD) functions to validate user access. At this stage, only limited information is pulled from AD, such as users' names. AD also contains important details regarding user credentials, IT access levels, and export control access, which will be integrated into future versions of the platform. For now, the login process checks only whether the user has valid AD credentials and is a member of a project. If either condition is not met, the user is denied access; otherwise, they are directed to the page described in the next section.

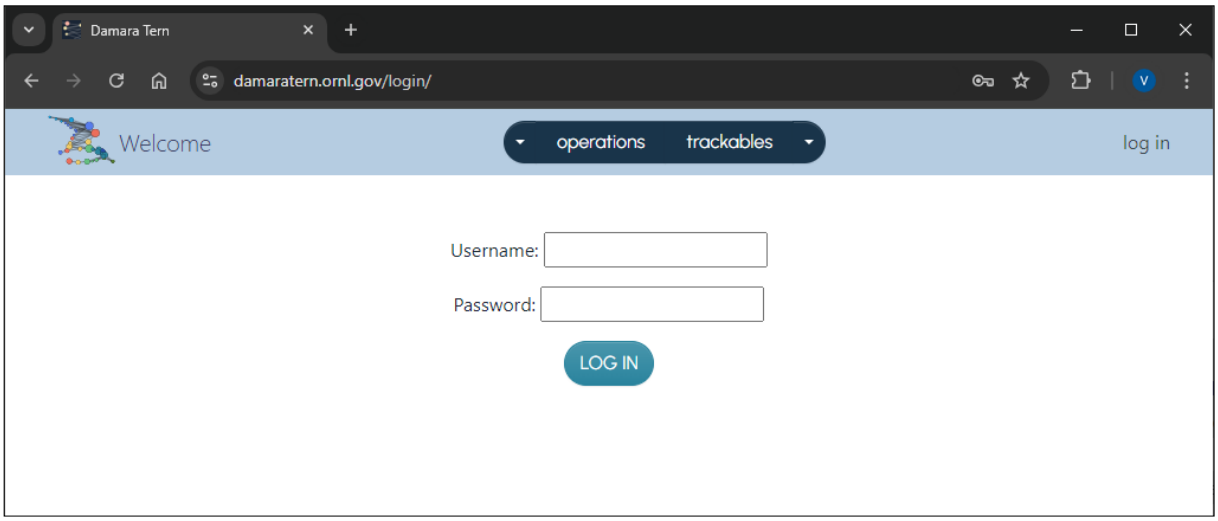


Figure 5. Login page: Traditional login page requires only a username and password to connect to the platform. The credentials used for login must be part of the AD and linked to a project in the Martian database.

3.2 HOME PAGE

Once the user is authenticated, the home page (see Figure 6) displays additional links with self-explanatory names to quickly guide the user to the relevant information. In the top banner, the username is displayed next to the welcome message. On the right-hand side, four options are available.

- Admin: Access to the admin page is provided only to users with admin privileges. If the user has these privileges, they can access server configurations and manage the database; otherwise, their request will be denied. Because the admin page contains sensitive information that could pose a security risk if exposed, its details are not covered in this report.

- Projects: Redirects the user to a web page listing the projects they have access to. This page is described in more detail in Section 3.3.

- Feedback: Redirects the user to a web page where they can submit comments and report bugs to the development team. This page is described in more detail in Section 3.4

- Logout: Option to close the current user's session.

The middle of the page displays several links for accessing operations and trackables specifically for searching and exploring the database, as well as for creating new entries. Trackables also offer the option to use a QR code search to quickly access linked data.

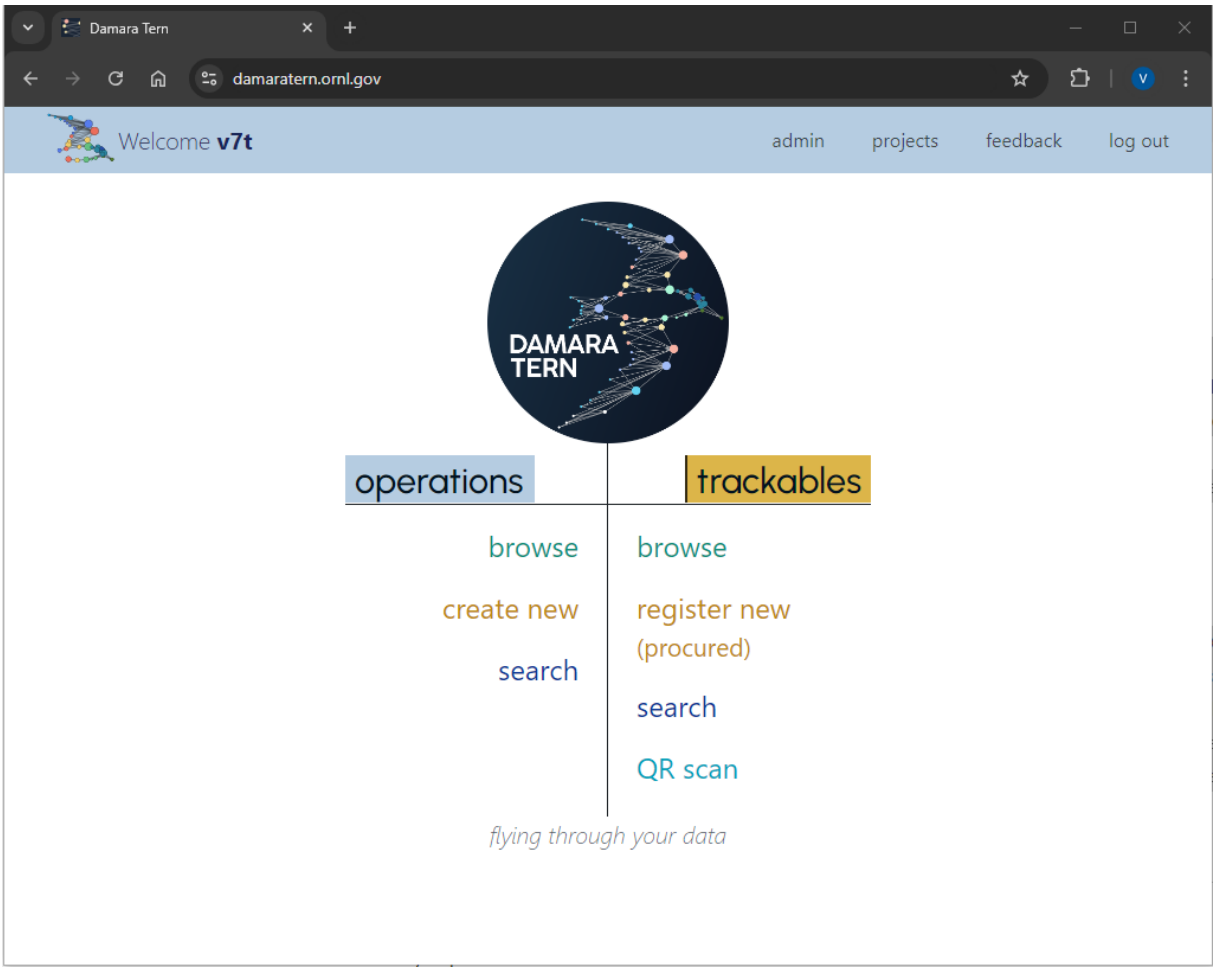


Figure 6. Home page.

3.3 PROJECT PAGE

By selecting the project page option (displayed in Figure 7), authenticated users can view a list of all the projects in the database. Because the database already contains hundreds of entries, users can filter the list to display only the projects they are members of by clicking the “My Projects” button. Pressing the “All Projects” button will reset the filter, and the web page will display all the projects on the server to which the user has access.

Notice also that two drop-down menus for “Operations” and “Trackables” are now present in the top bar. Both menus offer similar options compared with the ones found on the home page.

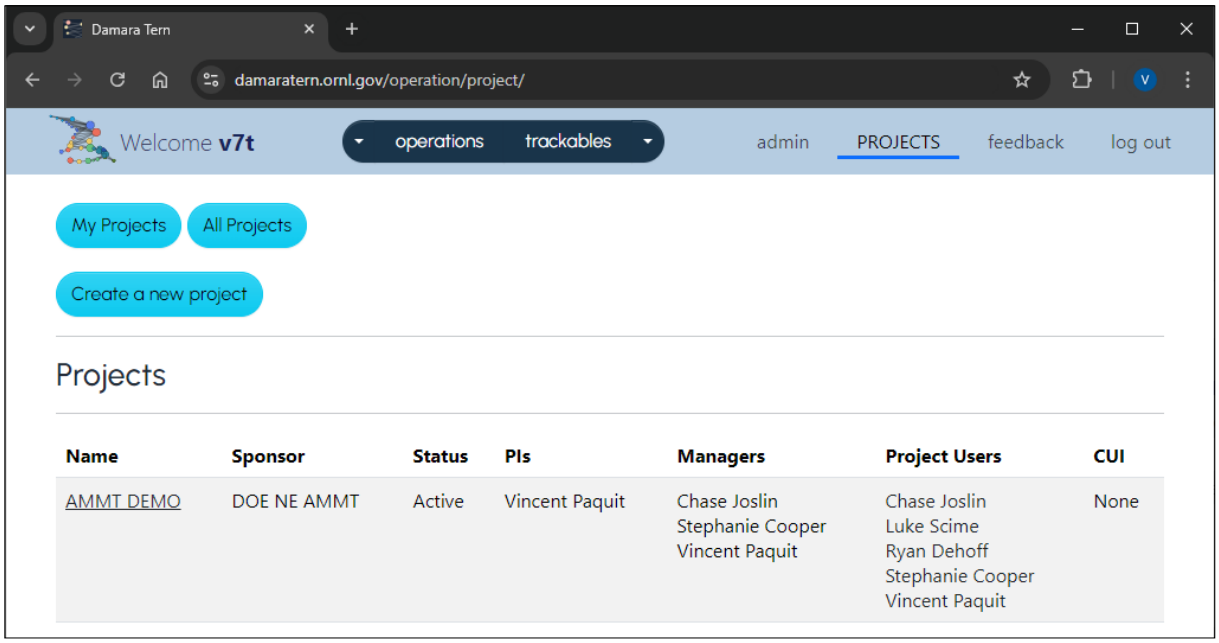


Figure 7. Projects listing page. In this particular example, the user selected the "My Project" option that returned only one entry.

The third option allows for the creation of a new project. By selecting this option, the user is redirected to the page shown in Figure 8. Although the form is self-explanatory, three key elements are important to note.

- Controlled Unclassified Information (CUI) and US Department of Defense (DOD) distribution statements: To ensure proper handling of the sensitivity of different projects, users are required to provide appropriate markings to identify the sensitivity level associated with each project.

- Add users: There are no limitations to the number of users that can be added to a project. However, only users approved to access the MDF Digital Platform can be added to a project. These users will already be selectable in the drop-down menu that is shown on the left side of the page under the user category. If a user is not present in the provided list, they will have to be added to the database by the administrator. At this stage, this platform does not offer a functionality to invite users from the interface. This feature will be added in a future release.

- User permissions, which has three control levels:
  - Manager—Has full control over the project, except for deleting it (only the admin is authorized to delete projects).
  - User—Can add information and data to the project.
  - Viewer—Can only view the content of the project.

Once the project is created, it is added to the list on the first page. If selected, a new page will display all information related to this project.

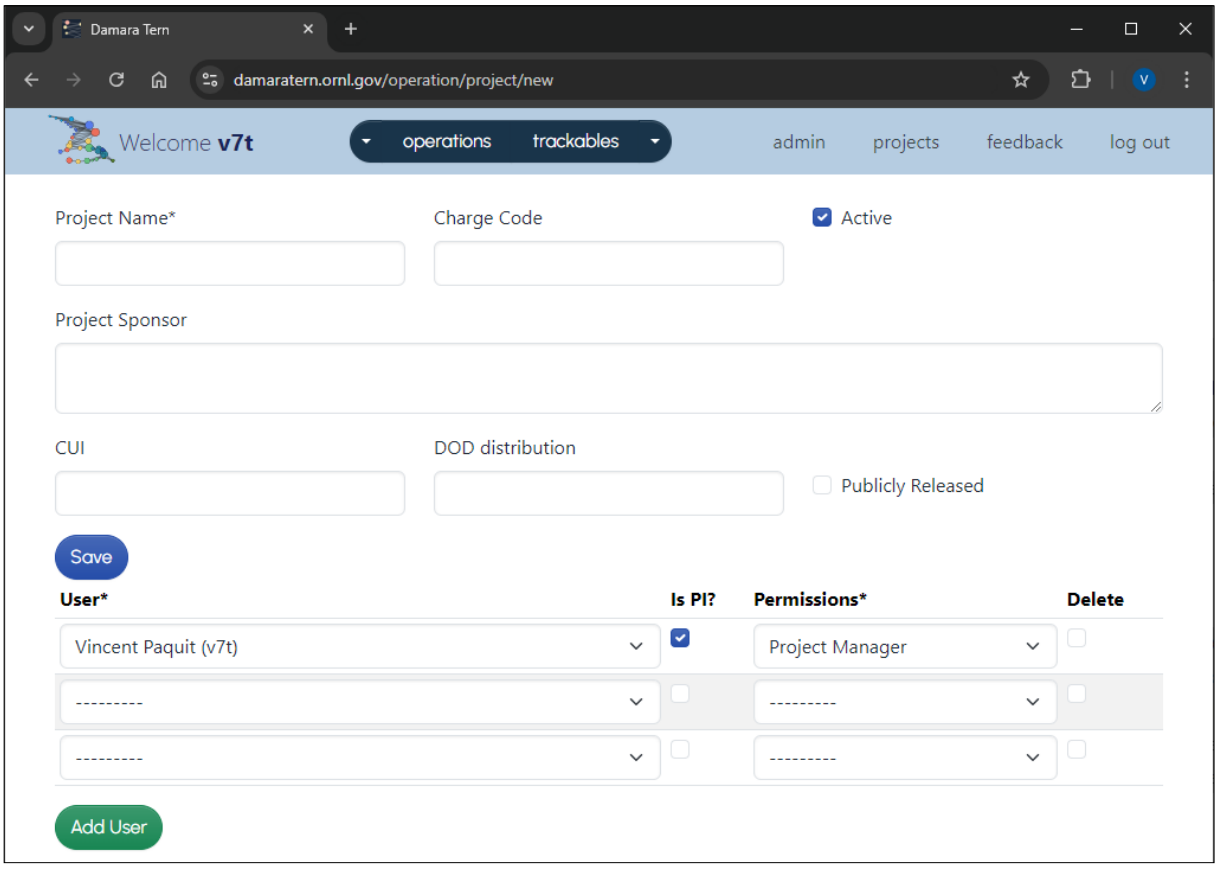


Figure 8. An example project creation page.

3.4 FEEDBACK PAGE

The MDF Digital Platform is currently under active development, making it important to provide a simple way for users to report bugs or suggest new features. Bugs are common in the early stages of a project, such as page rendering issues across different web browsers or unexpected behavior of a functionality in certain scenarios. User experience is also crucial for the platform's success. This work aims to incorporate users' expectations and feature requests into the work plan as quickly and easily as possible. The simple feedback submission form shown in Figure 9 fulfills this function. Users can select "bug report" or "feedback report" from the dropdown menu, choose the location within the website, and then provide details in the comment text box. The developers have already received valuable submissions from initial alpha testers and have been able to address each one promptly. This process is crucial for developing a platform that meets users' needs.

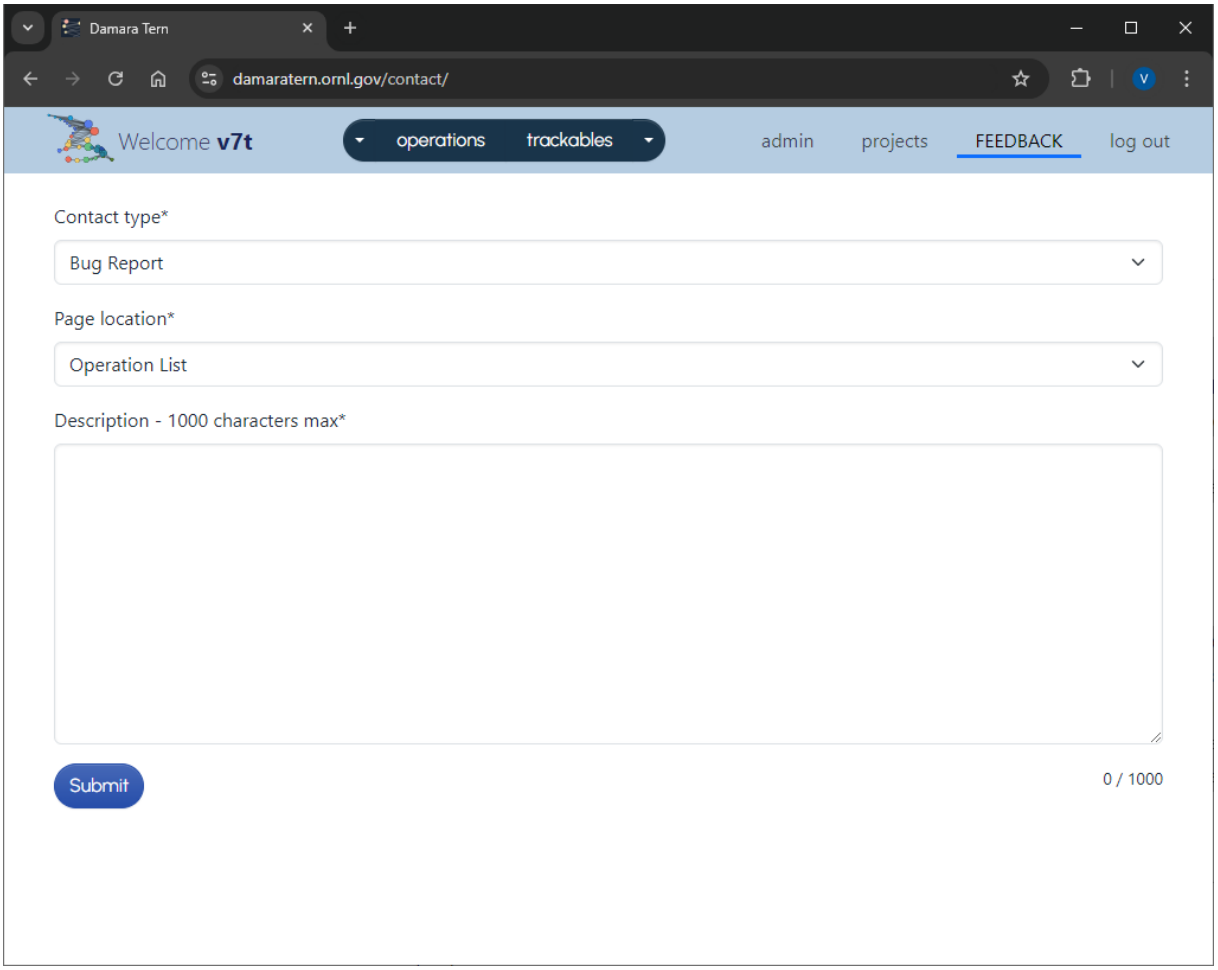


Figure 9. Feedback page. This page can be used to report bugs, submit comments, or suggest improvements and new features.

3.5 OPERATIONS PAGE

By default, the Operations page, shown in Figure 10, lists all the operations present in the database. These operations can be filtered using the search function, which is accessible by clicking the search button at the top right of the page. A pop-up window, shown in Figure 11, provides users with filters to select specific datasets. Currently, the search function is based on common filters. The development team is planning to release a search feature based on data content—for example, allowing users to query the database with natural language questions such as "List all failed LPBF operations" or "Show all printed geometries in 316SS that are 15 cm tall." Figure 12 shows an example of a web form to create a new operation. The form to create an operation requires the user to provide more information.

- Operation type: The drop-down menu will provide multiple options. Only those options can be selected, and if a new one needs to be added, the administrators of the system are the only ones allowed to do it for now.

- Machine: A similar drop-down menu provides options. Once again, if a new one needs to be added, the administrators of the system are the only ones allowed to do it for now.

- CUI and DOD distribution statements: There are special cases in which the operation but not the project is sensitive (or vice versa). This option allows users to make the distinction, but the sensitivity can also be inherited from the project automatically.

- Three trackable buttons: these will assume the functionalities described in Section 3.6.

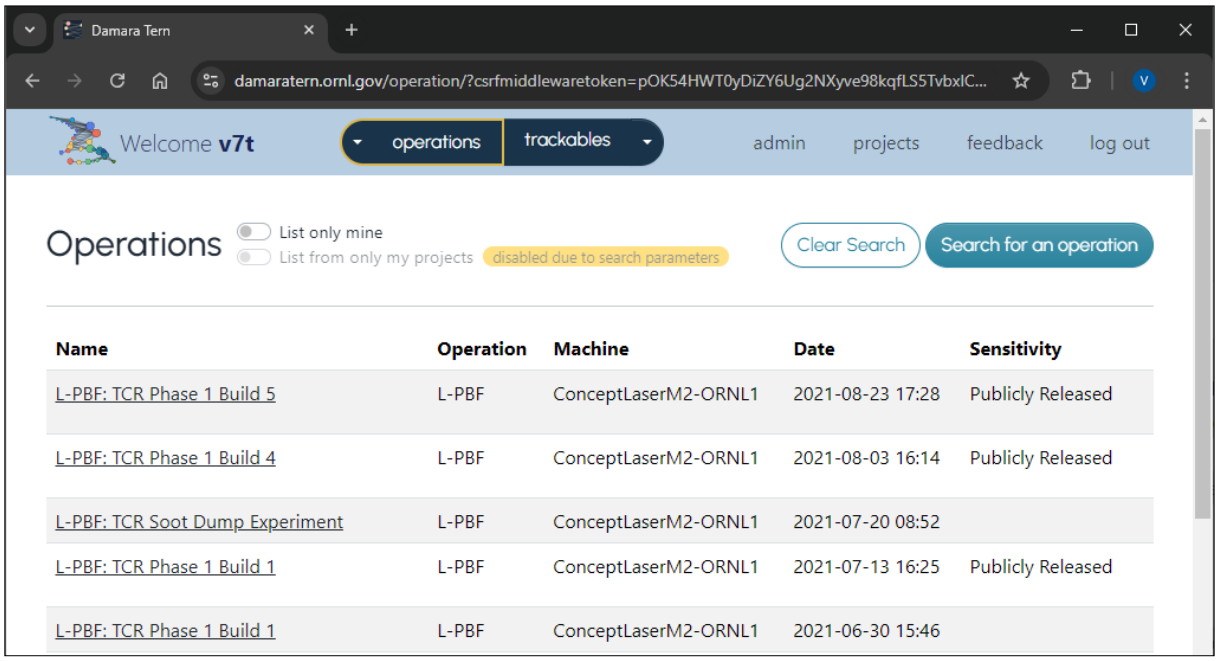


Figure 10. List of operations available in the database. Only a few entries are shown for clarity.

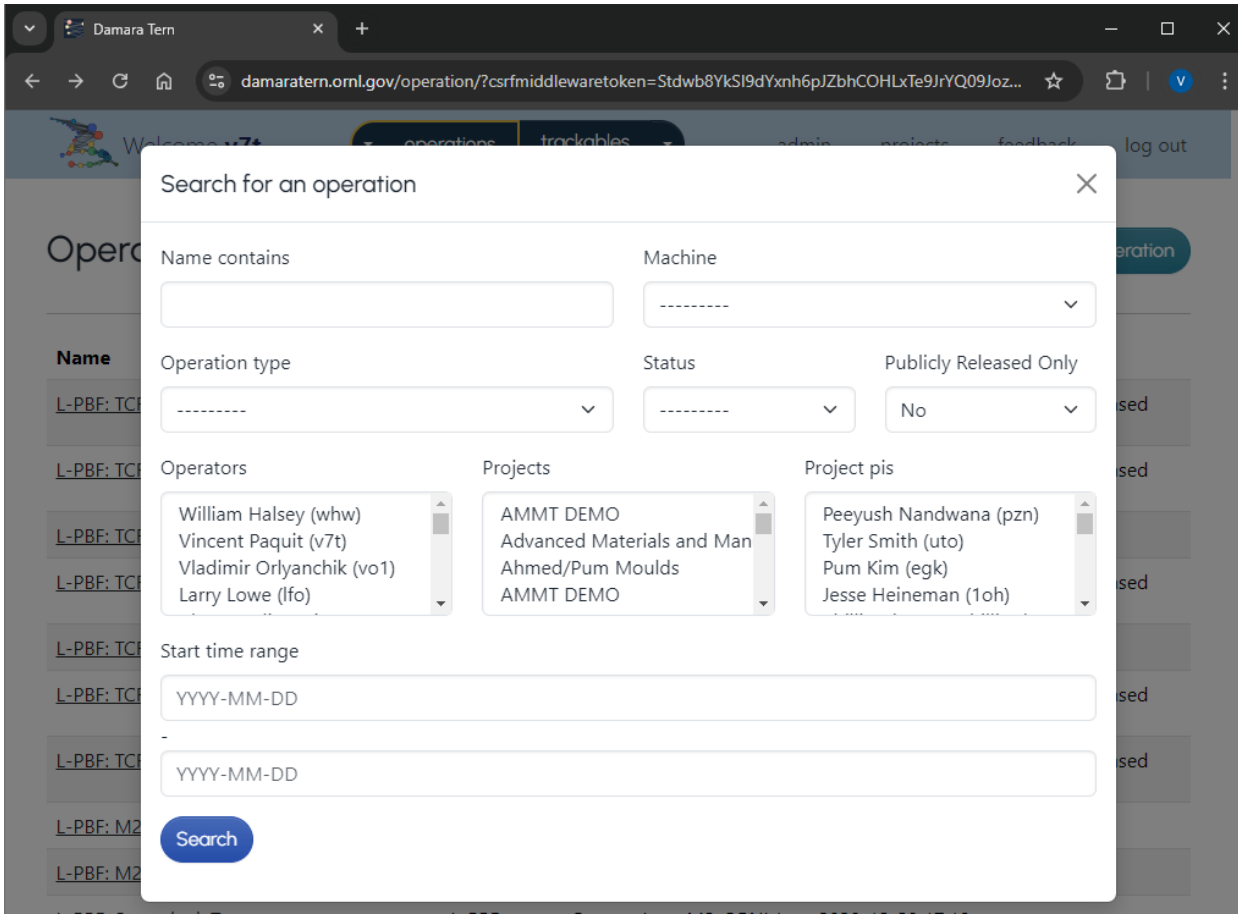


Figure 11. Dialog box to search for specific operations using filter-based queries on the database.

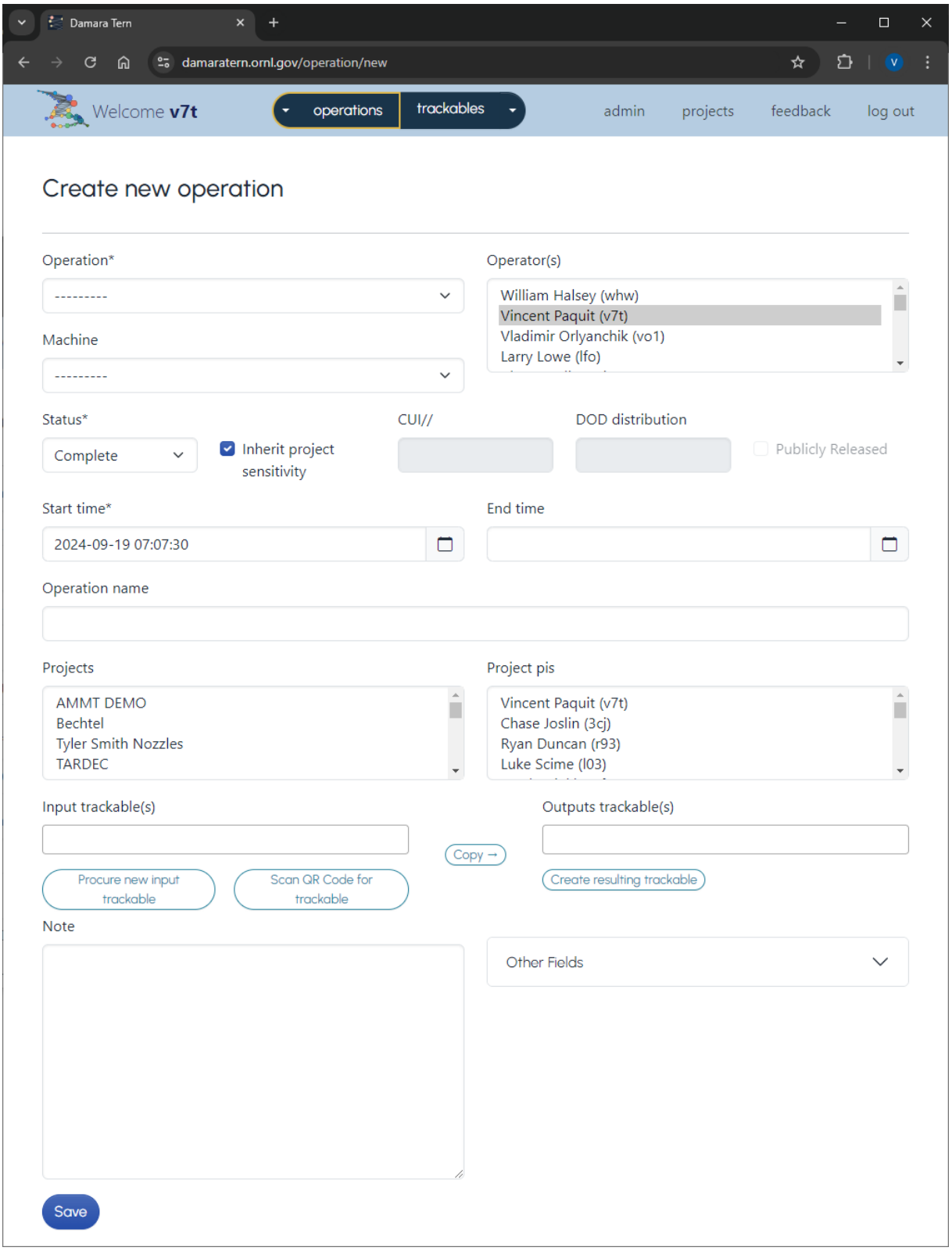
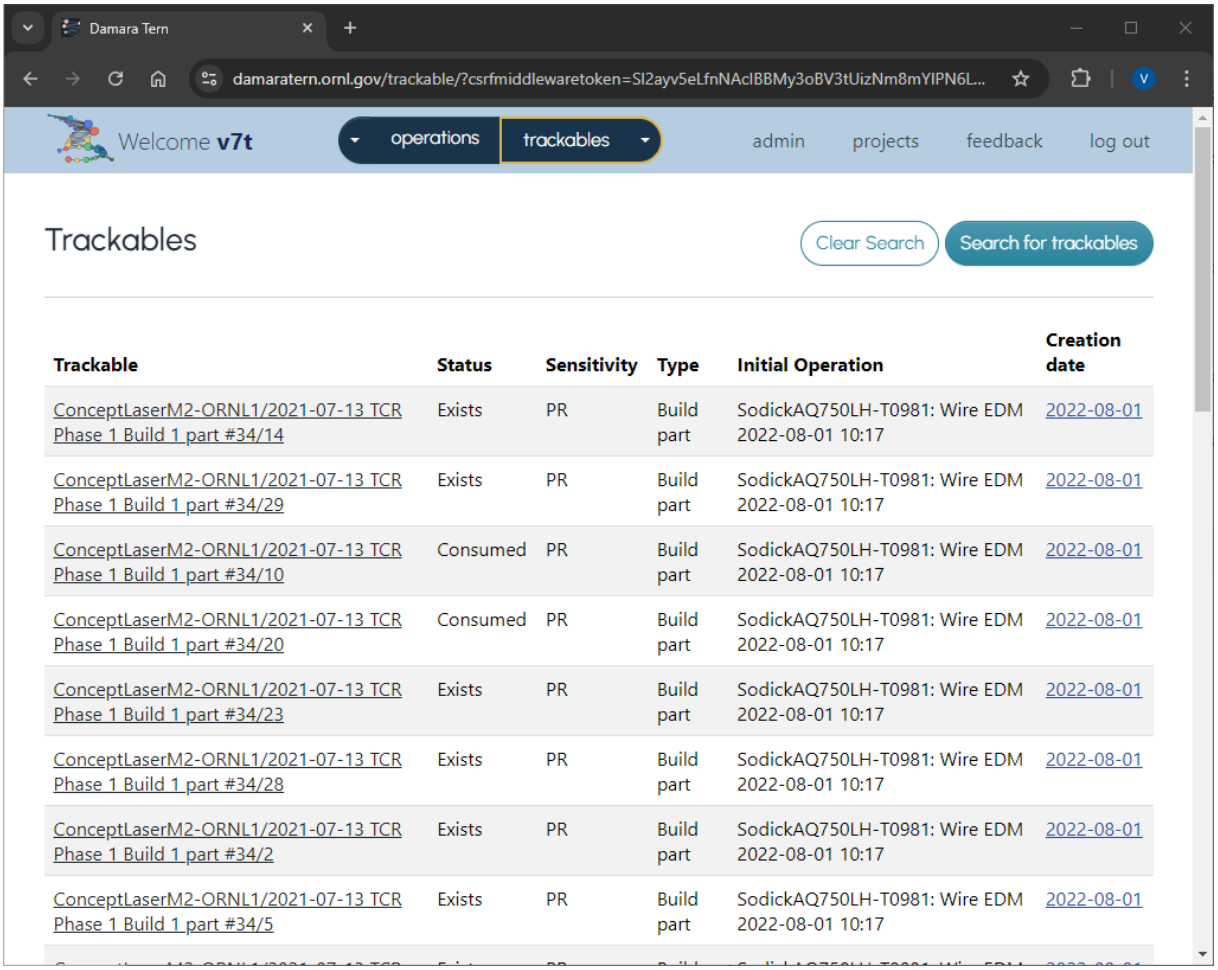


Figure 12. Example web form to create a new operation.

3.6 TRACKABLE PAGE

All three trackable pages will have the same functionalities as the operations page described in Section 3.5. These are presented in Figure 13, Figure 14, and Figure 15. The only difference is the introduction of the QR code string to generate a code for the trackable. This QR code is a feature specific to trackables and does not exist for operations.



Trackable	Status	Sensitivity	Type	Initial Operation	Creation date
ConceptLaserM2-ORNL1/2021-07-13 TCR Phase 1 Build 1 part #34/14	Exists	PR	Build part	SodickAQ750LH-T0981: Wire EDM 2022-08-01 10:17	2022-08-01
ConceptLaserM2-ORNL1/2021-07-13 TCR Phase 1 Build 1 part #34/29	Exists	PR	Build part	SodickAQ750LH-T0981: Wire EDM 2022-08-01 10:17	2022-08-01
ConceptLaserM2-ORNL1/2021-07-13 TCR Phase 1 Build 1 part #34/10	Consumed	PR	Build part	SodickAQ750LH-T0981: Wire EDM 2022-08-01 10:17	2022-08-01
ConceptLaserM2-ORNL1/2021-07-13 TCR Phase 1 Build 1 part #34/20	Consumed	PR	Build part	SodickAQ750LH-T0981: Wire EDM 2022-08-01 10:17	2022-08-01
ConceptLaserM2-ORNL1/2021-07-13 TCR Phase 1 Build 1 part #34/23	Exists	PR	Build part	SodickAQ750LH-T0981: Wire EDM 2022-08-01 10:17	2022-08-01
ConceptLaserM2-ORNL1/2021-07-13 TCR Phase 1 Build 1 part #34/28	Exists	PR	Build part	SodickAQ750LH-T0981: Wire EDM 2022-08-01 10:17	2022-08-01
ConceptLaserM2-ORNL1/2021-07-13 TCR Phase 1 Build 1 part #34/2	Exists	PR	Build part	SodickAQ750LH-T0981: Wire EDM 2022-08-01 10:17	2022-08-01
ConceptLaserM2-ORNL1/2021-07-13 TCR Phase 1 Build 1 part #34/5	Exists	PR	Build part	SodickAQ750LH-T0981: Wire EDM 2022-08-01 10:17	2022-08-01

Figure 13. List of trackables available in the database. Only a few entries are shown.

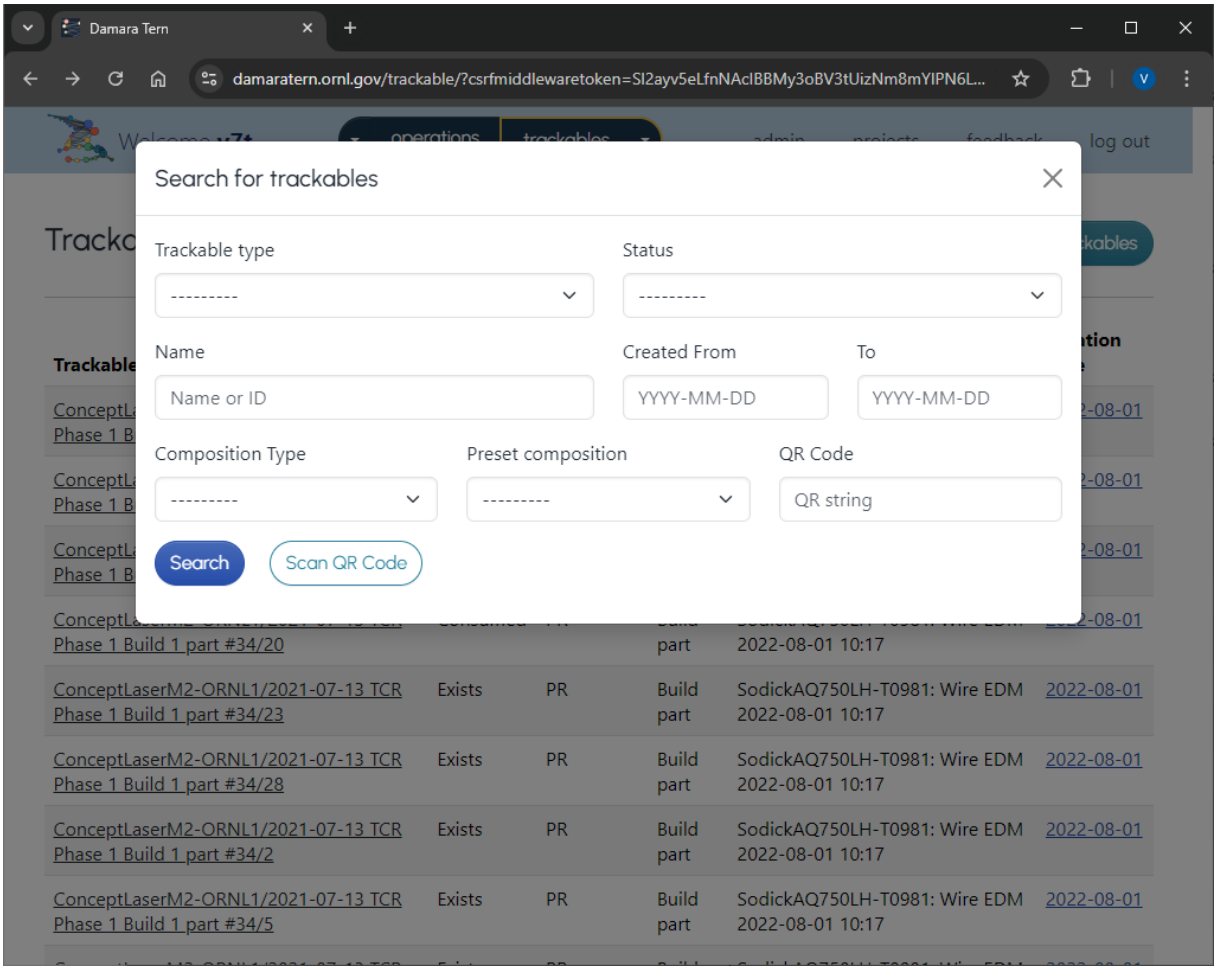


Figure 14. Dialog box to search for specific trackables using filter-based queries on the database.

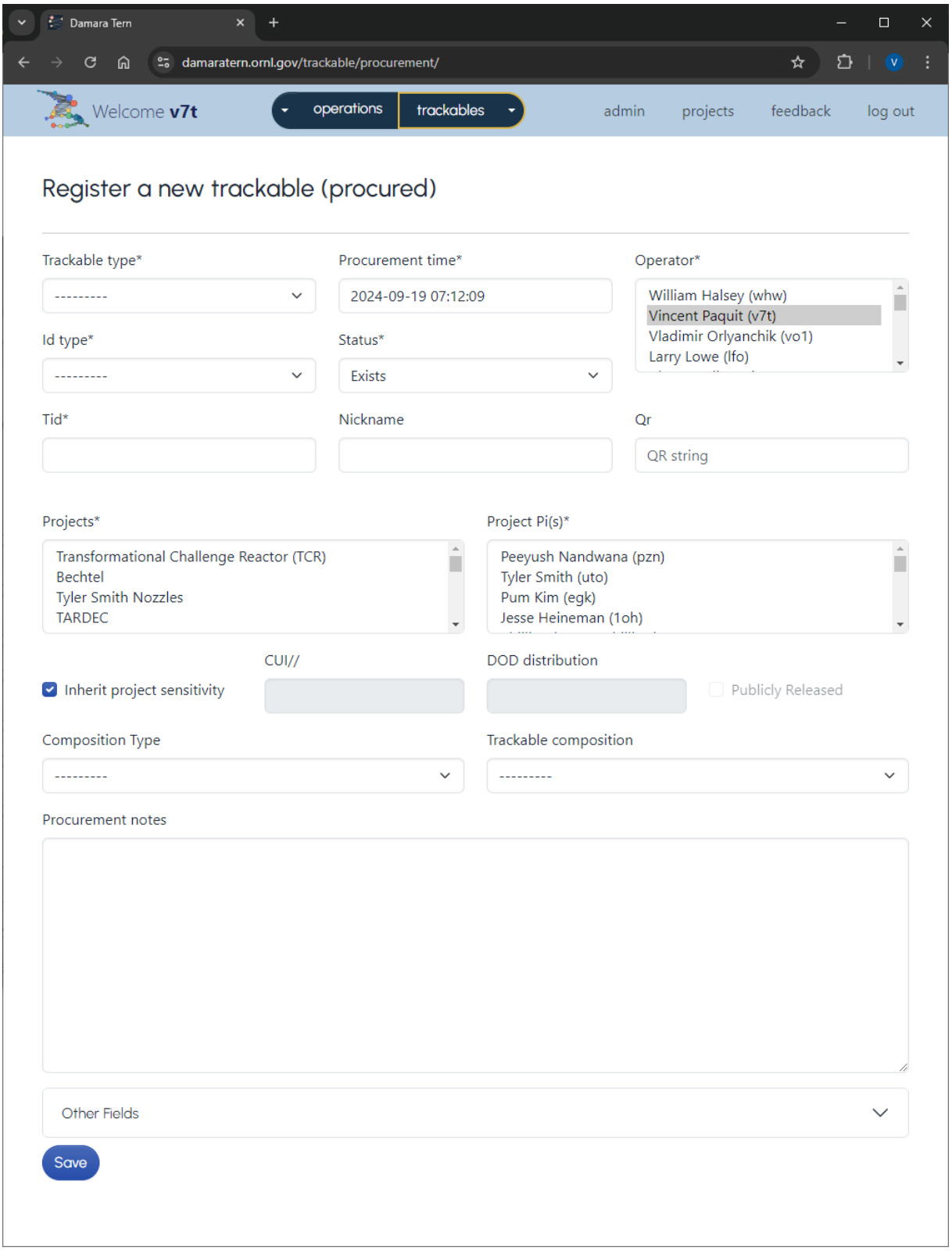


Figure 15. Web form to create a new trackable.

3.7 SELECTED OPERATION INFORMATION PAGE

After selecting a specific operation from the list, the user is redirected to the detailed page for that operation (see Figure 16 and Figure 17). This long page provides access to all information recorded in the database, starting from the top of Figure 16 with the title, followed by a block of metadata. Below this information are two clickable links: one for input trackables and one for output trackables. Currently, the page has one input and one output trackable, but multiples of each are possible. The following block of text titled “Notes” contains information provided by the operator of the machine, who in this case recorded a series of errors that would explain the unsuccessful outcome of this build.

Figure 17 shows the additional data displayed on the page. The “Add Related Data” section, located toward the top left, provides three options for adding extra data to the operation: (1) the “file” button allows the user to select a file directly from their local computer, (2) the “URL” button lets the user select an online resource, (3) and the “link uploaded files” button lets the user select files that are already present on the server where Damara Tern resides. Just below these options is a list of all data associated with the operation. These data are displayed either with featured images selected by someone working on the project or as a bulleted list of individual datasets. Users have the option to view these datasets online or download them locally.

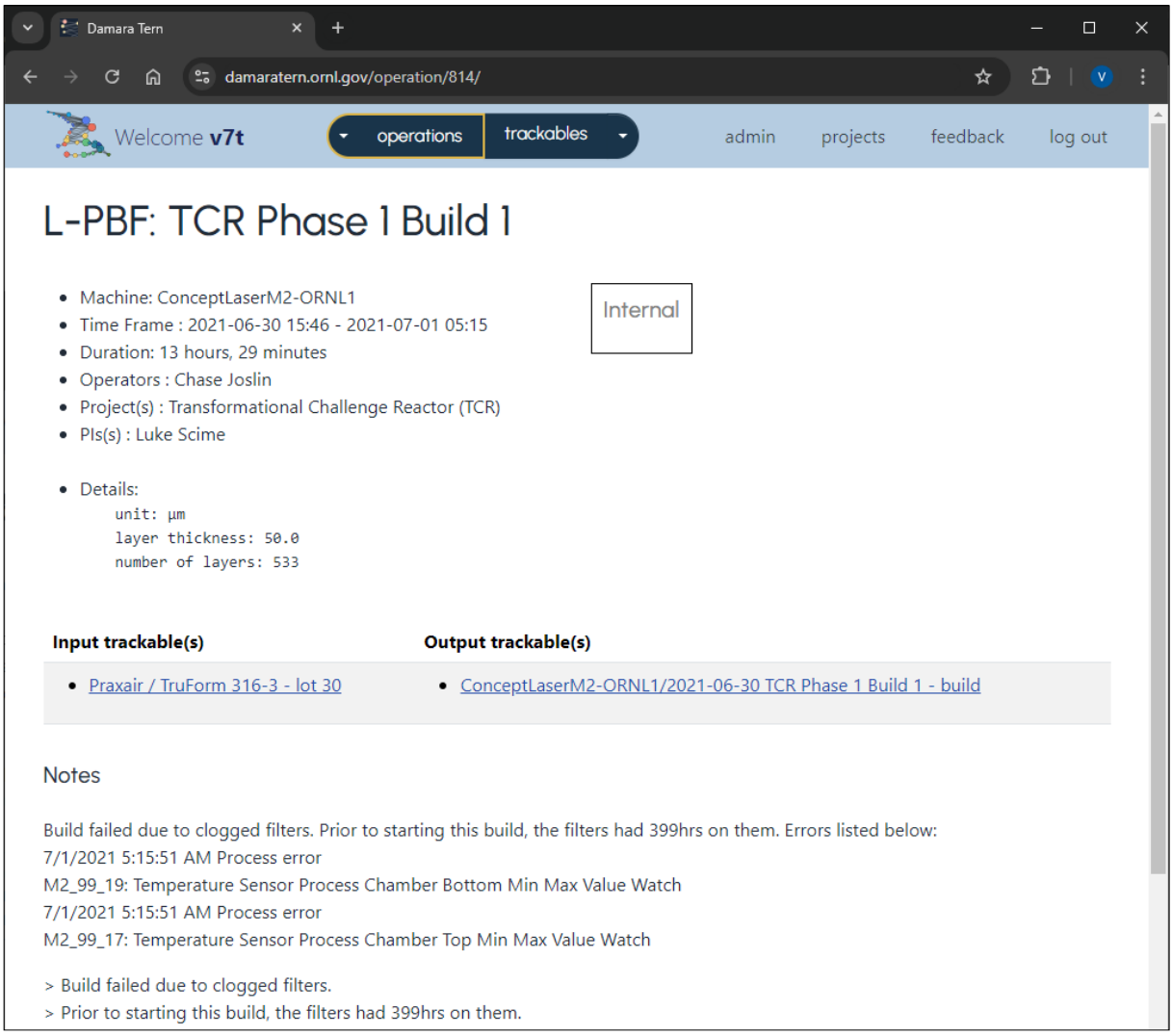


Figure 16. Summary page for a selected operation.

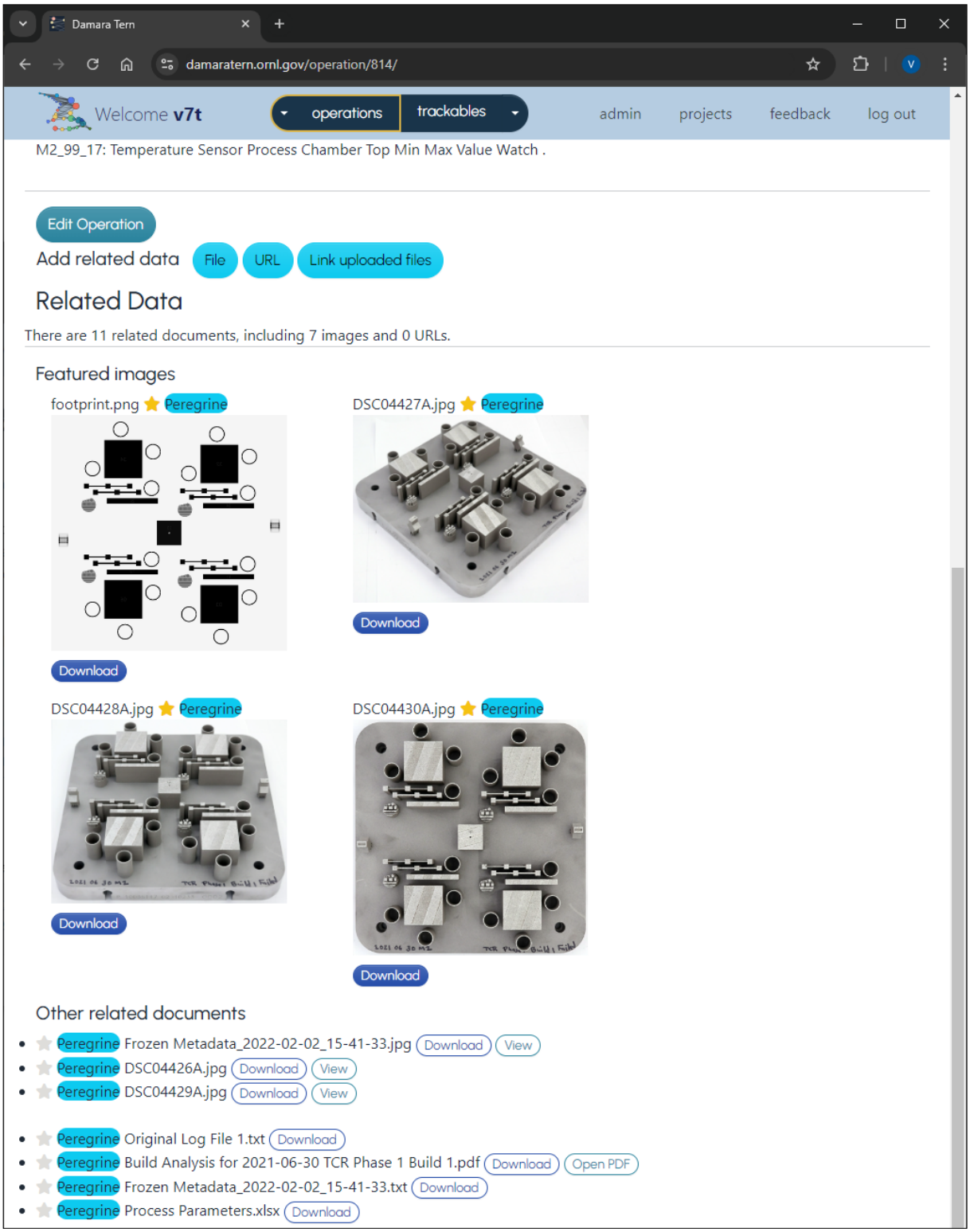


Figure 17. Summary page for a selected operation (continued).

3.8 SELECTED TRACKABLE INFORMATION PAGE

After selecting a specific trackable from the list, the user is redirected to the detailed page for that operation (see Figure 18 and Figure 19). This page is designed similarly to the Selected Operation Information page but has some differences:

- A QR code is provided to be placed on the sample to easily retrieve this record.
- A historical list of operations is attached to this sample—in this case, the sample was created using an electrical discharge machine and then tested.
- In Figure 19, the trackable history offers a different way to explore the digital thread of a component using a graph-based representation connecting operations and trackables.

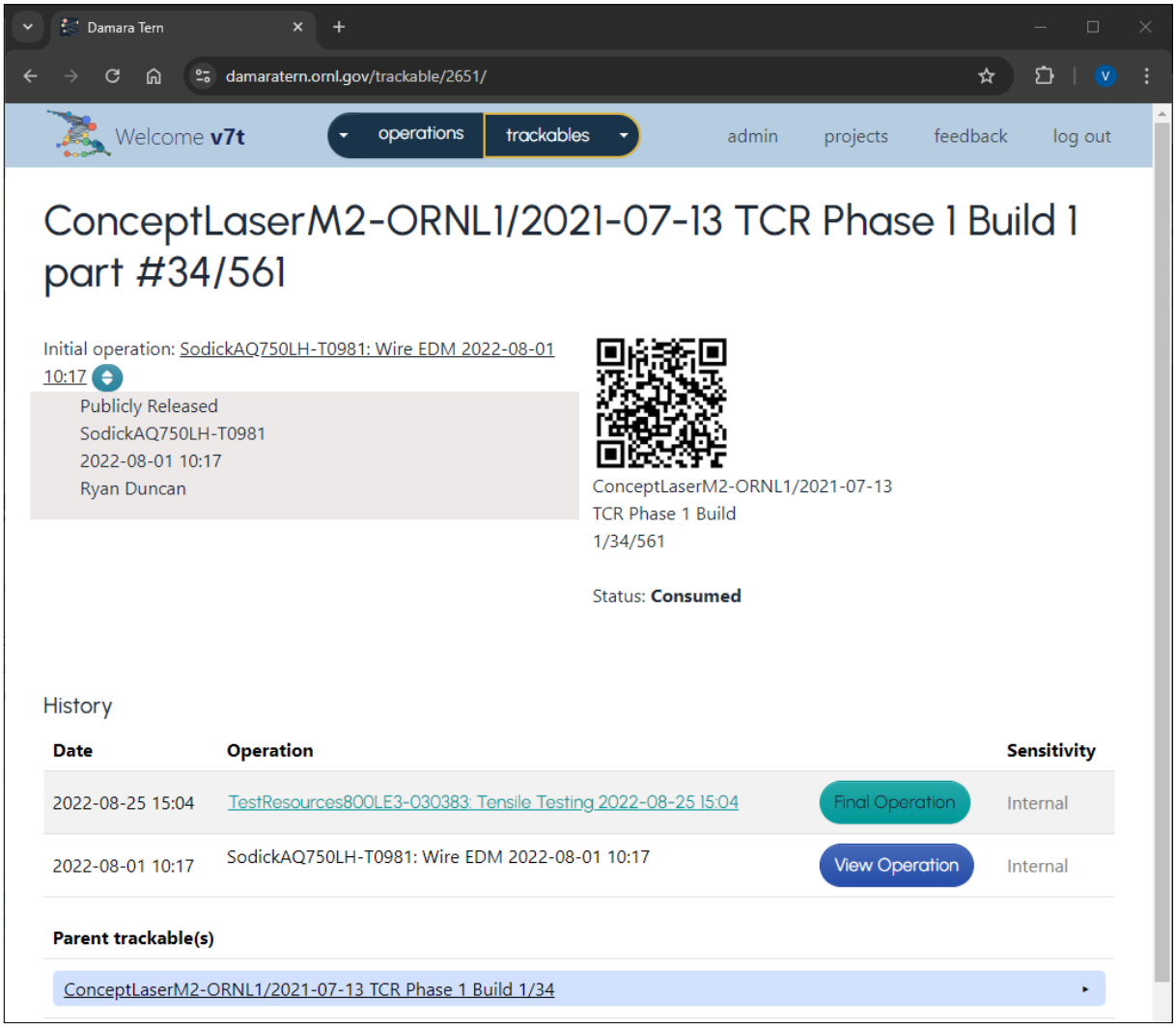


Figure 18. Example selected trackable information page.

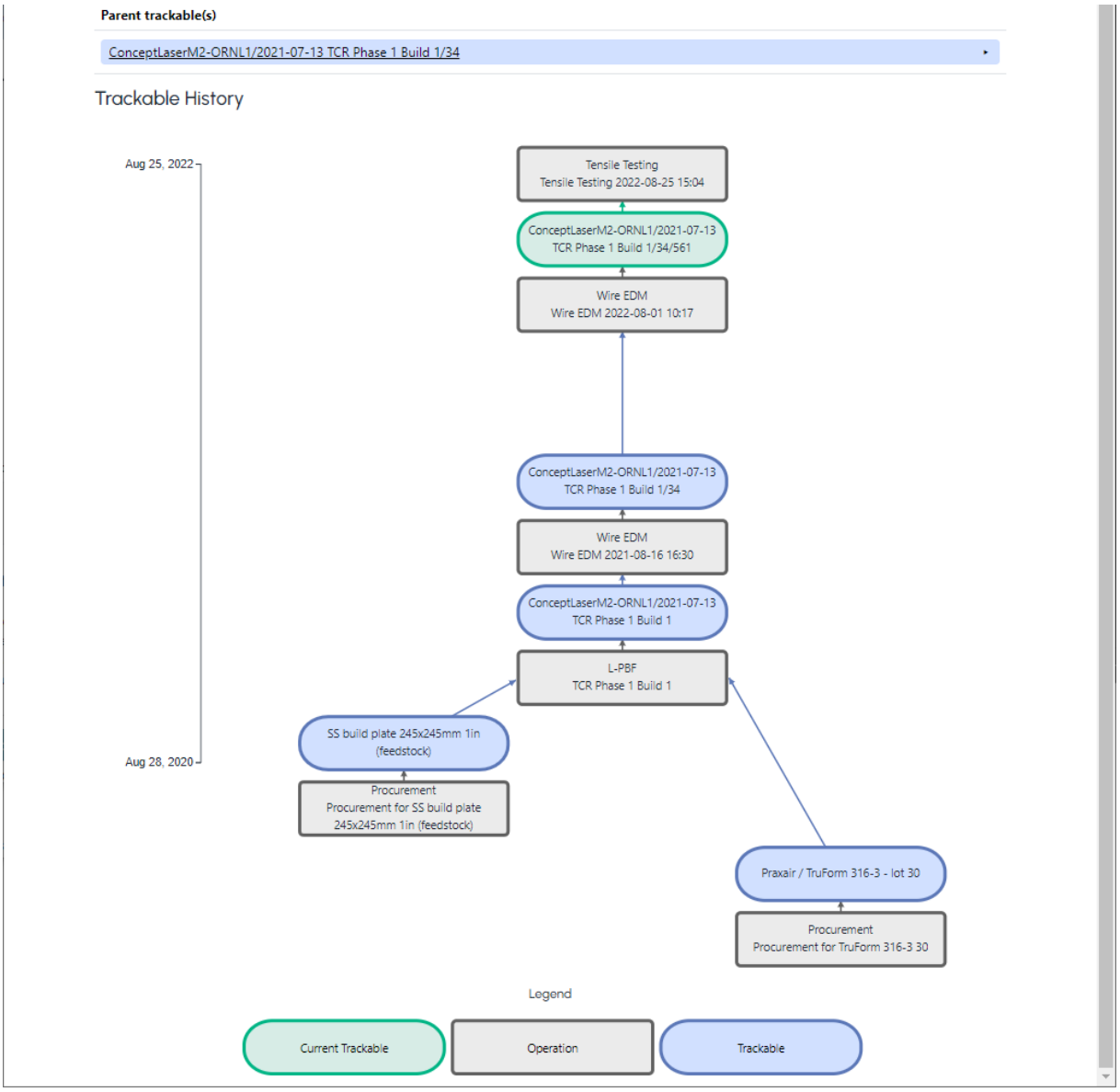


Figure 19. Trackable history displayed using a graph-based representation, with the current trackable highlighted in green. In this example, by following the downward links, users can trace the steps that led to the creation of this tensile sample.

3.9 DAMARA TERN API

Since 2020, the Peregrine software tool has been configured to communicate with the legacy MDF database via several API services running on the MDF Digital Tool. At a high level, the purposes of the existing services are to (1) synchronize metadata between the Peregrine user interface and the web portal and (2) maintain the linkage between individual trackable objects (e.g., tensile samples) and their associated in situ sensor data. To support the transition from the legacy MDF Digital Tool to the new Damara Tern platform, several changes have been made to the Peregrine codebase.

First, all hard-coded API end-point URL have been replaced with dynamically configurable user settings, and the code has been modified so that both the legacy MDF Digital Tool and Damara Tern can be supported simultaneously. This modification will allow a phased transition, switching over to Damara Tern one powder bed printer at a time. Second, significant configurability has been exposed to the user so that modifying the database metadata format will no longer require corresponding code changes on the Peregrine end. This change will also enable future compatibly between Peregrine and user-created, alternate MDF Digital Platforms and database structures provided that certain design rules are followed. Finally, initial work has begun allowing Peregrine to transmit authentication tokens to Damara Tern to maintain proper data access controls. Table 4 summarizes the current status of the relevant API services.

Table 4. The descriptions and statuses of each API service that Peregrine currently uses to communicate with the legacy MDF Digital Tool. Once each service is complete, Peregrine can be transitioned to use Damara Tern at ORNL.

Name	Description	Status
server_status_endpoint	Allows Peregrine to confirm that it can establish an active connection to Damara Tern.	COMPLETE
machine_list_endpoint	Returns to Peregrine the list of powder bed printers that currently exist in the Damara Tern database.	COMPLETE
pull_metadata_endpoint	Returns to Peregrine the metadata for a specific powder bed printing operation as a formatted JSON (JavaScript object notation) blob.	IN PROGRESS
announce_new_files_endpoint	Allows Peregrine to alert Damara Tern to the existence of new files on the file system (i.e., server) that should be indexed for display on the web portal.	NOT STARTED
push_trackable_promises_endpoint	Allows Peregrine to transmit to Damara Tern the number of trackable parts (and child parts) that could be produced from a specific powder bed printing operation.	NOT STARTED
new_builds_endpoint	Returns the list of powder bed printing operations that have been created via the Damara Tern web portal but that have not yet been linked to a Peregrine build instance.	NOT STARTED

4. CONCLUSION

The work package aims to establish a common digital manufacturing discipline among all participants of the AMMT program to improve the performance of nuclear components. In FY 2023, this project analyzed the manufacturing workflow and digital thread associated with additively fabricated components to develop a new data management strategy based on the concepts of operations and trackables. This approach ensures the preservation of data integrity and the links between data collected by AMMT members. This year, this project built the data management platform and demonstrated its functionality using laser powder bed data, achieving the requirements of this milestone.

This report marks the completion of FY 2024 milestone M2CR-22OR0403052: Demonstration of the Use of the New Digital Platform at the MDF on a Laser Powder Bed System.

5. REFERENCES

[1] Cooper, Stephanie, Ryan Dehoff, William Halsey, Vladimir Orlyanchik, Vincent Paquit, Luke Scime, Zackary Snow, and Amir Ziabari. 2024. *FY23 Multi-Dimensional Data Correlation Platform: Unified Software Architecture for AMMT Data Management and Processing*. ORNL/TM-2023/3093, Oak Ridge National Laboratory. DOI: 10.2172/2439894.