CriticalMAAS Milestone 5 report – Macrostrat TA4 team

Daven Quinn and the UW–Madison / Macrostrat CriticalMAAS team

May 7, 2024

Contents

1 Report period

This **Milestone 5** report describes the the technical progress of the UW–Madison – Macrostrat team on the CriticalMAAS project, from March 30 – May 7 (Month 8). It sets final expected capabilities and integration plans for the end of Phase 1, based on the work completed to date and the codebases described in the Milestone 4 report in April 2024.

2 Research and technical progress

The UW–Madison – Macrostrat team has made significant progress on the CriticalMAAS project leading up to the 9-month Hackathon. The team has continued to develop the core Macrostrat codebase and extend it for project objectives, particularly around map ingestion, geologic entity characterization from the literature, and new APIs to provide targeted subsets of geologic map data. Additionally, we have continued to develop the Mapboard topology manager codebase for editing geologic maps. Taken together, these elements form the core of a system that can integrate and provide access to geologic maps to feed to TA3. During this period our effort has gone mostly to

- Streamlining data pipelines for map ingestion, feedback collection, and editing, in order to consolidate towards a final architecture for the Phase 1 system.
- Developing user interfaces for all project components that can be the basis for feedback from experts and other TA teams during the 9-month hackathon.

Here, we briefly summarize progress in several domains.

2.1 Macrostrat

We have continued to refine the Macrostrat core codebase to enable map ingestion and eventual standalone operation by the USGS and other partners. This has mostly consisted of refining Macrostrat's command-line utilities for database maintenance and migrations, upgrading Macrostrat's APIs, and consolidating the map ingestion pipelines that we demonstrated at the 6-Month hackathon for more efficient operation and better visibility of pipeline status.

Core system upgrades

We have begun explicitly breaking out database schema elements into separate subsystems that can be managed independently with commands such as `macrostrat db update <subsystem>`. This has enabled rapid iteration of schema designs in a modular fashion. This system has been used to experiment with data structures needed for map ingestion and literature knowledge assimilation. The `macrostrat up` command now starts a full Macrostrat stack, but it is missing the legacy v2 API and an ability to bootstrap the database from scratch. We will work on these missing elements towards the end of Phase 1; in the meantime, we do maintain both 'minimal' (only stratigraphic datasets and dictionaries; no

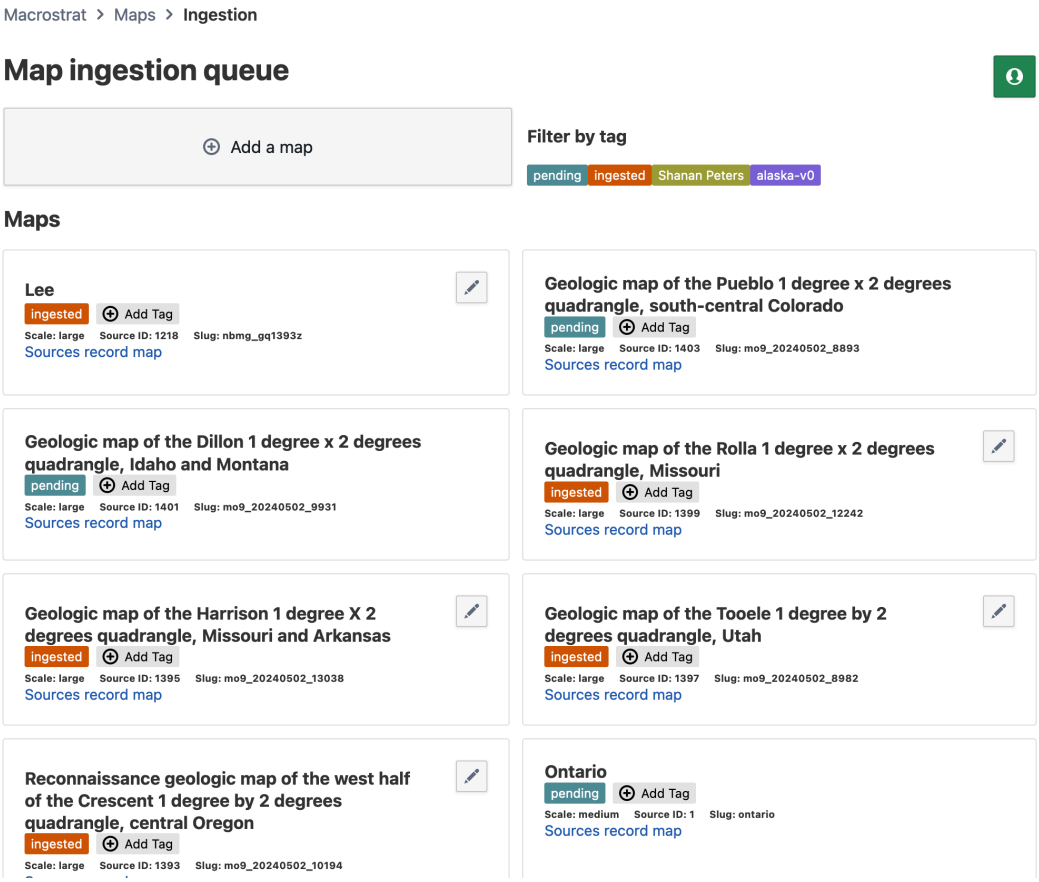

Figure 1: Map ingestion tracking overview

maps) and 'full' (all current and in process mapping) database dumps that can each be used to bootstrap the system.

Map ingestion pipeline

We have continued developing Macrostrat's map ingestion pipeline into a more approachable web application. In order to help deal with a with more tools to manage maps' lifecycle in the Macrostrat system (e.g., the ability to tag and add metadata to maps during the ingestion process). Additionally, we have made substantial progress acquiring already-vectorized maps from USGS, state geologic surveys, and other sources. We have begun to stage these maps into Macrostrat through automated pipelines. Several procedures that required multiple CLI commands to be run in sequence have been simplified and converted into automated 'daemon' processes. As a result

For many maps, significant "expert" work (requiring undergraduate-level understanding of geology) is still required to standardize legend entries to Macrostrat's minimum data requirements. Our new metadata tracking tools enable "triage" of tricky maps and the legend editing HITL itself is now broadly usable and efficient to support this process. We will demonstrate this tool to USGS during the 9-month Hackathon to seek feedback and testing; over the sum-

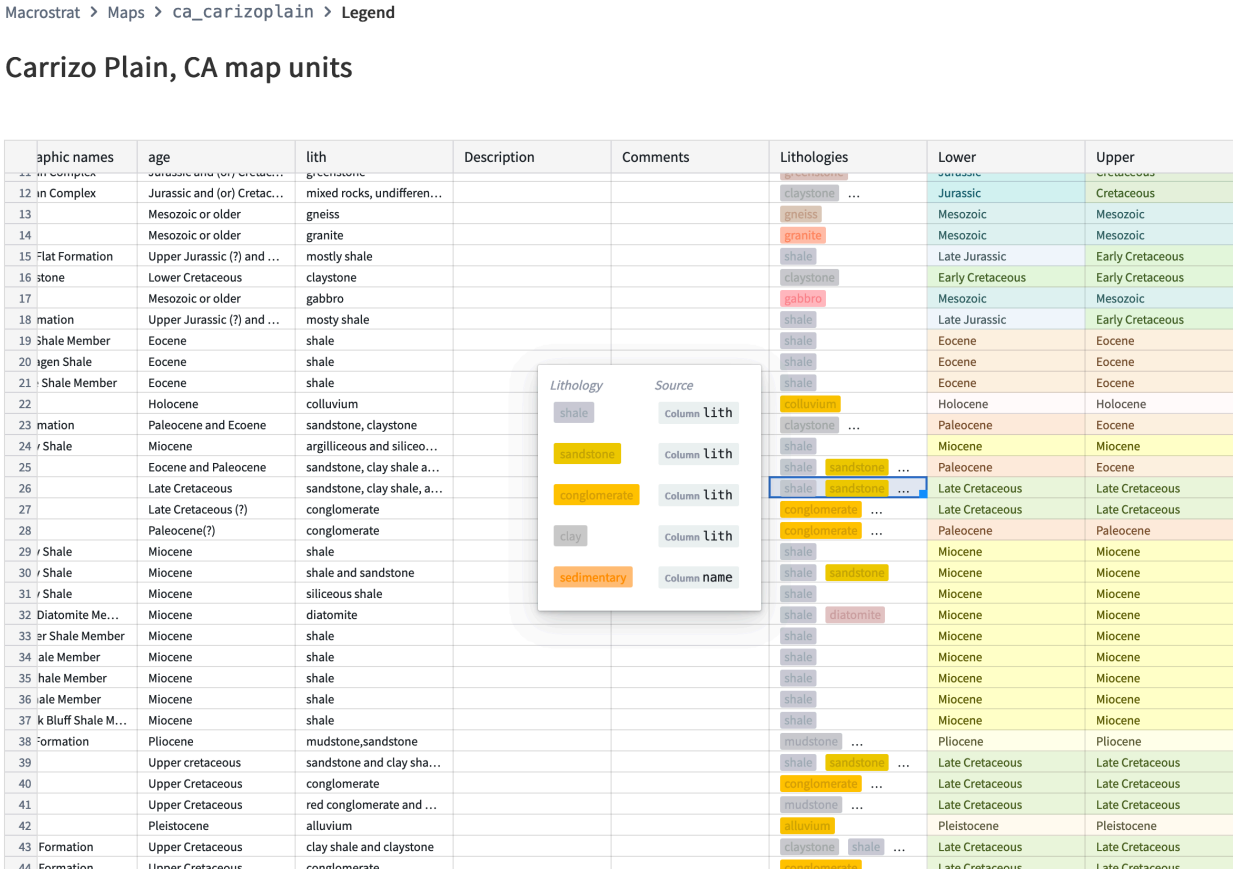

Figure 2: New table component for viewing and editing map legend entries

mer we will engage several student workers to enter legend information for new maps. This new pipeline is broadly similar to our pre-CriticalMAAS practices, but with many manual steps totally automated, enabling us to ingest maps much more efficiently. Once TA1 teams conform to the CriticalMAAS schema for legend extractions, we will be able to automate more of this process.

Standardized user interface elements

Macrostrat has long been committed to browser-based user interfaces, which are highly accessible and require no installation. One of the challenges we have faced in designing HITL components in a web context is the need to rapidly scaffold and prototype data manipulation tools with a consistent (and ideally mostly "self-documenting") user experience. One way we have addressed this is through creating standardized web components that can form elements of many different HITL workflows. We have recently finished standardizing some of our ingestion tables into React components that can be used for any table-based data editing task, with similar semantics to a spreadsheet across arbitrary data. Coupled with the PostgREST standardized API toolkit for PostgreSQL and lazy-loading approaches, we are able to provide such table interfaces over high-scale data, such as Macrostrat's entire catalog of geologic units

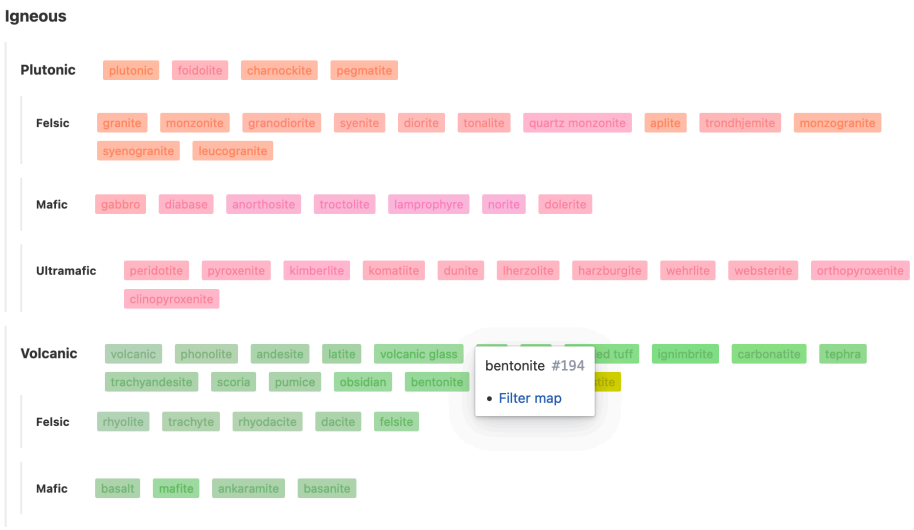

Figure 3: New visualization of Macrostrat's nested lithology dictionary

(>75,000 rows). We have structured this system so that it can easily integrate data editing and validation tools. These tools have increased our ability to rapidly scaffold new HITLs and will eventually underpin robust and consistent editing capabilities. These components are being developed in the web-components repository.

APIs for providing mapping to TA3

We are continuing to develop APIs and workflows for TA3 to access the Macrostrat database. We have produced a basic filterable tile layer that can be used to access Macrostrat data in a controlled fashion across large areas. The current filtering approach is solely based on an `AND` concatenated list of lithology elements, which our defined in our tree-based lithology dictionary. This has just been produced and we will discuss its integration with TA3 and MTRI. Other approaches to build a more complex filter may eventually be made available, depending on need. These filtering approaches will be made accessible via the [Macrostrat client library][macrostrat_client] produced by MTRI and recently extracted into its own codebase.

2.2 Map editing

We have made several upgrades to the Mapboard map editing system allowing it to work more effectively alongside Macrostrat's other services. The codebase has been expanded to allow for multiple map topologies to be managed in the same PostgreSQL database (in separate schemas). The Mapboard topology manager that forms the core of the system is now fully implemented as a Python module following modern Macrostrat coding standards. It has a full test suite with 39 passing tests. We have also added new features such as separate, nested mapping layers, and control of valid data types between layers (e.g., so surficial units can be separated from bedrock units). In advance of the 9-month Hackathon, we are bringing the Mapboard system into our Kubernetes platform to enable wider testing, with both QGIS and

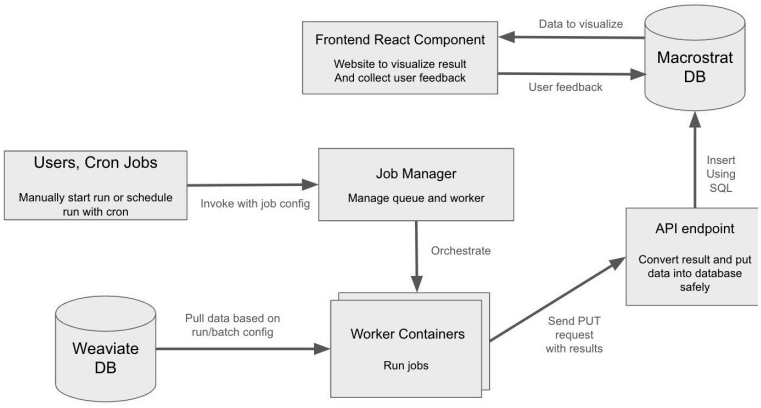

Figure 4: Entity canonicalization pipeline design

the Mapboard GIS iPad app as client interfaces. A web interface is also in development, but this may not be ready for demonstration at the Hackathon. Some essential elements like user tracking and authentication have not yet been integrated, but code and database models already in use within Macrostrat will be adapted for this purpose.

2.3 Geologic entity canonicalization

We have continued to develop a pipeline for geological entity canonicalization from the literature, based on xDD and both direct knowledge-graph transformer (KGG) and LLM-prompting graph construction approaches. The overall pipeline is described in the `macrostrat-xdd` repository, and the different modeling approaches are laid out in the `llm-kg-generator` and `unsupervised-kg` repositories. Progress during this period has focused on building the infrastructure for an integrated modeling framework that encompasses both approaches, can be run on demand, and adds results directly to the Macrostrat database for visualization. In advance of the 9-month Hackathon, this pipeline is being used to process results using both the LLM and KGG pipelines. We have accumulated model results for evaluation over several thousand paragraphs so far.

One of the challenges holding back the system is the lack of granular training data at the paragraph level (see Milestone reports 3 and 4), which has made evaluating recall and precision of each model difficult. To address this, we are building a feedback and training data capture component to allow experts to provide `strat_name -> lith -> lith_attr` relationships that can be used to train the model. The same interface will allow correction of extracted entities, and these user-provided links will also be useful as their own source of data. We are finalizing this interface to demo and solicit feedback at the 9-Month hackathon.

While the initial version of this pipeline works over documents from xDD, we hope to eventually be able to run these models over any text relevant to CriticalMAAS including mining reports, map pamphlets, and other sources. These tools may be eventually integrated into our map ingestion pipeline to pull structured unit characteristics from map unit descriptions.

2.4 Entity extraction and document tracking

We have expended significant energy in making documents available for TA2, which has resolved into the contents and data models of the CriticalMAAS Document Store being contributed to seed the CDR. At the 6-Month hackathon, there was significant interest in our COSMOS engine for extracting figures, tables, and other elements from documents in semantic context. We have improved the ability of the pipeline to run on demand over any collection of documents, not just those already in xDD. As a test before the 9-Month Hackathon, we are running the pipeline to segment the ~200 documents identified by Graham Lederer as relevant to CMMI deposit type classification (Slack message, April 23). These extractions will be relevant to a conversation (hopefully, at the 9-Month Hackathon) about how to contribute these extracted elements to the CDR and employ them in TA2 workflows.

3 Expected capabilities of Phase 1 system

Our progress on infrastructure and user interfaces during this period has allowed us to sketch the broad organization of our Phase 1 system. We expect the system to be locally runnable within a Docker compose environment with minimal and clearly-defined dependencies on outside APIs (e.g., Mapbox basemaps, CDR datasets). Here, we present the broad organization and expected capabilities of the Macrostrat system at the end of Phase 1. All codebases will be open-source and packaged within the system as Docker containers unless otherwise noted.

If we can achieve the expected capabilities of the Phase 1 system, the entire Macrostrat stack will be an independent component that can be set up where needed. Deployment in a "production" capacity by USGS may not be possible until the end of Phase 2 depending on integration requirements, but the system should be deployable for evaluation and integration testing at the very least. It is unclear at this moment what the context in which the program will want to run the system. Will it be in an cloud environment alongside the CDR or some other computing environment? Who at the USGS will need to operate the system? This does not need to be decided right now, but some indication will help guide the specific functionality and documentation we prioritize to make it easier to run.

3.1 Macrostrat core system

- A container stack that can be run using `macrostrat up` in a Docker compose environment.
- A set of command-line utilities for managing the database, running the map ingestion pipeline, and other tasks.
- A PostgreSQL database with a set of schemas for managing maps and stratigraphic information
- Various API and web frontend containers for accessing and editing the database
- Authentication with ORCID for user tracking and editing permissions
- Hooks for CDR push and pull as needed

3.2 Map ingestion pipeline

- HITL tools for managing the lifecycle of maps in the Macrostrat system, including metadata tagging and map prioritization
- Ingestion pipeline for vector geologic maps working over arbitrary sets of geologic data
- CDR-integrated ingestion pipelines for TA1 maps, making use of the TA1 schemas to pre-fill information
- Legend editing tools to bring TA1 and born-vector maps to Macrostrat and USGS data standards

3.3 Map provision to TA3

- A set of tile-based APIs for accessing Macrostrat geologic map data in aggregrate (all maps) or granular fashion (individual maps) with optional gap filling.
- A filterable API for geologic units matching specific criteria
- Standardized fields for paleolatitude, age, and lithology of units that can be easily integrated as TA3 input layers

3.4 Map editing

- "Mapboard" topologically aware map editor HITL that can load data to and from the CDR
- Ability to move between Macrostrat (space-filling) and Mapboard (topologically expanded) representations of geologic maps
- Support for QGIS, web, and Mapboard GIS iPad app for editing topologically aware maps (the Mapboard GIS app and server hooks required to communicate with it are not open-source).

This map editing app will be loosely coupled to the rest of Macrostrat, allowing it to be run over maps that are not yet ready for Macrostrat ingestion,

3.5 Geologic entity canonicalization

- A pipeline for extracting geologic entities from the literature and adding them to the Macrostrat database
- A feedback user interface for vetting entity extractions and providing training data for the canonicalization models
- A dataset of extracted and vetted rock descriptions that will become part of the Macrostrat database

Entity canonicalization pipelines have substantial system requirements relative to the rest of the system and are currently coupled to the xDD corpus and APIs. We cannot commit to having the entire system fully deployable by the government team by the end of Phase 1, since we are just getting to the stage of an integrated prototype now. However, we will ensure that

these services are loosely coupled to the rest of the system, so that they can continue to run in the xDD infrastructure and can process arbitrary text (e.g., from the CDR or Macrostrat's map ingestion pipeline). This will allow the system to be used to augment the Macrostrat database as long as the integration is prioritized. If USGS-run versions of these canonicalization pipelines become a program priority, we can commit effort to the required engineering in Phase 2.

4 Gaps

Broadly, we are making sustained progress towards our Phase 1 goals, and the biggest barrier to our success is our own ability to execute efficiently on planned HITL tools. This is often limited by the need for coordinated changes across Macrostrat's database, APIs, and web frontend components. However, this is a manageable problem that is well within our purview. Still, there are several areas where we have notable gaps that we are working to resolve or discuss with other teams.

4.1 Role-based access control

We have a basic authentication system for controlling access to Macrostrat's editing capabilities for feedback and data integration. However this system is still clunky and difficult to operate. We are pursuing several approaches to improve this system including:

- Integration with ORCID-based OAuth for user tracking and authentication (removing CILogon which is user-unfriendly)
- Better integration of identity with PostgreSQL user permissions and row-level security for editing
- A more robust system of ensuring that PostgreSQL roles are centrally defined alongside database views and tables
- Systems such as PostgREST that provide efficient connectivity between database access control and web services.

These systems are all operating within Macrostrat but the fully polished access management that will frame all of our HITL tools and allow cooperative work by USGS staff is a key need that we will address in coming months.

4.2 Connectivity of Macrostrat user interfaces

We are rapidly moving towards developing user interfaces and HITL tools for navigating, improving, and providing feedback for geologic maps. However, many of these components are isolated within different pages of Macrostrat's user interface codebase. One of our major goals has been making the Macrostrat system explorable and somewhat "self-documenting" for users. Examples of this type of integration include:

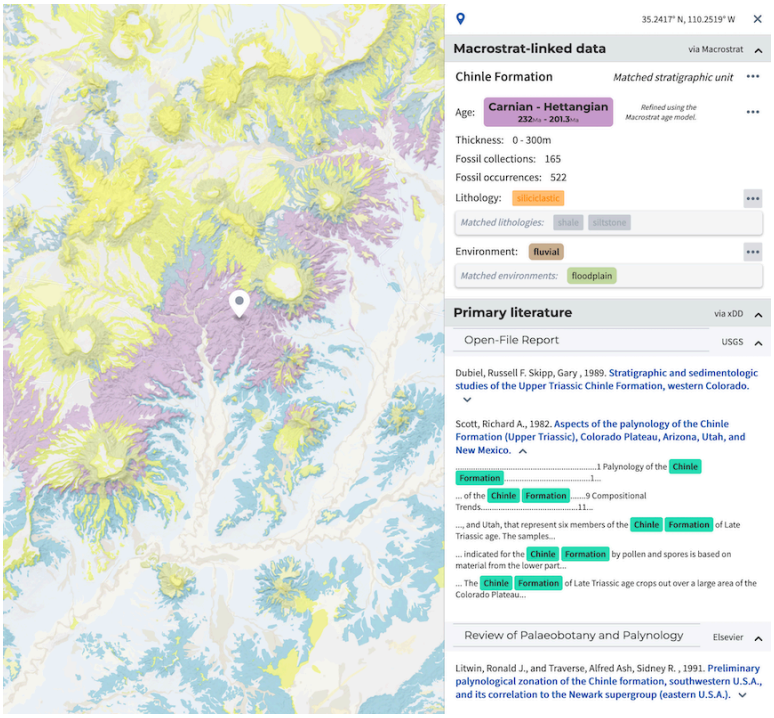

Figure 5: Macrostrat contextual data for a geologic unit that should lead the user to feedback and evaluation interfaces

- Exploring Macrostrat's main map interface should quickly allow jumping to evaluation and feedback on an individual map
- A paper mentioning a unit in Macrostrat's xDD panel should allow a user to quickly jump to linking unit attributes in the source paragraph to provide training data for canonicalization models.

These connections require both more linking in the user interface and, in some cases, new APIs to be developed. While much of this connectivity will not be built out at the 9-Month hackathon, we will continue to work on this over the summer.

4.3 Structurally complete products from TA1

Macrostrat's map ingestion systems have in the past been focused on high-quality, vetted map products with all necessary elements (i.e., points, lines, and polygons). Currently, TA1 output has not been synthesized across teams and problem domains; for instance, projected data is not easily available through the CDR. Jataware is working on this, but it has held up progress on integrating TA1 mapping products and will likely be a program pain point in the future.

4.4 Integration with TA2

We have struggled to understand the expectations and obligations for user interface development for TA2, especially in light of the shifting design of the CDR. On the data system side,

we have already developed components that turned out to be of little use to the program (e.g., the document store and TA1 Geopackage Format) and are reluctant to explore too deeply in directions that will be superseded by Jataware-originated designs. We have begun to discuss this problem with Jataware and others, and we anticipate that the month 9 Hackathon will be a good opportunity to make progress on specific plans around the development of HITLs themselves (not just CDR schemas). This is required to enable us to dedicate resources to these UI construction tasks if necessary.

5 Integration plans for the end of Phase 1

Our Phase 1 integration plans are focused primarily on ensuring that there is a clear path for TA1 data to be standardized and provided to TA3 via Macrostrat's ingestion pipelines and APIs. The goal of providing contextualized and attributed TA1 mapping for efficient use in CriticalMAAS will require two key integrations:

1. Integration with the CDR to ingest 'completed' or structurally correct TA1 maps for legend correction and processing. This will require TA1 to provide schema-compliant data and, ideally, the right set of HITL tools (both 'upstream' tools developed by Jataware and the Macrostrat map ingestion system to deconflict and finalize the structure of TA1 products.
2. Integration with TA3 to provide Macrostrat data in a way that is useful for their machine learning models and for interactive subsetting by experts. Much of this integration already exists but Month 9-12 will be focused on improving and streamlining these capabilities, and integrating them with other performers, especially MTRI.

Additionally, we will work to continue several integrations that secondary in importance but will bolster CriticalMAAS's chances of success:

1. Integration with xDD and entity canonicalization pipelines to provide descriptions of geologic units into the Macrostrat database, both from the literature and arbitrary text, and to link back to source material.
2. Integration with existing sources of vector-based geologic maps, via connections to NGMDB, state geologic surveys, and academic institutions, to bring more mapping into alignment in a way that will support TA3 as well as other uses of the mapping in the geologic community (e.g., exploration and spatial search for relevant data across organizations). In this, we will take seriously our obligations to link back to the canonical source for all mapping.

We look forward to discussing and perfecting these Phase 1 plans at the 9-Month hackathon and beyond.