Common Data Model Harmonization (CDMH) and Open Standards for Evidence Generation

Final Report

Prepared by:

The combined IBM and FDA CDMH Contract Team

December 31, 2018

Table of Contents

I. BACKGROUND AND PROJECT VISION

The Patient Protection and Affordable Care Act of 2010 created the Patient-Centered Outcomes Research Trust Fund (PCORTF) to help build the national capacity and infrastructure needed to conduct patient-centered outcomes research (PCOR), and to enable PCOR findings to be integrated into clinical practice. The aim of these efforts is to allow patients, providers, and caregivers to make more informed healthcare decisions.[1] Office of the Secretary (OS) PCORTF funds are made available annually through 2019. The Secretary of Health and Human Services (HHS) has delegated authority to the Office of the Assistant Secretary for Planning and Evaluation (ASPE) to coordinate and implement a strategic plan to invest these funds effectively.

With oversight provided by HHS ASPE, this PCORTF project is a collaboration of several HHS agencies including FDA, National Institutes of Health (NIH) (National Cancer Institute (NCI), National Center for Advancing Translational Sciences (NCATS), National Library of Medicine (NLM)), and the Office of the National Coordinator (ONC)). There is a desire to achieve a sustainable data network infrastructure, promote interoperability, and foster the creation of a Learning Health System (LHS) as laid out in the Connecting Health and Care for the Nation a Shared Nationwide Interoperability Roadmap[2]. This infrastructure will be achieved by mapping and transforming data across various Common Data Models (CDMs) and by leveraging open-source standards. By mapping various CDM data elements and leveraging existing PCORTF investments, it is feasible to reuse the data, methods and other resources from each network (e.g., PCORnet, Sentinel, *Observational Health Data Sciences and Informatics* (OHDSI), i2b2) thereby providing PCOR researchers with access to larger and more diverse types of Real World Data (RWD).

II. PROJECT GOALS

This project will build upon existing OS PCORTF investments and other HHS investments. While each of these investments help to make data accessible and understandable, each will address specific data sources, representing not just specific data models but also specific collection use cases and constraints.

Within this contract with FDA, team members will work towards achievement of the following goals:

- Development of a general framework (i.e., tools, processes, policies, governance and standards) for the transformation of various CDMs, curation, maintenance and sustainability.
- Assessment of the value of the developed CDM harmonization mechanisms by demonstrating research utility for safety evaluation of cancer drugs that use the body's immune system (PD1/PDL1 inhibitors) with a focus on patients with autoimmune disorders.

[1] https://aspe.hhs.gov/patient-centered-outcomes-research-trust-fund

[2] https://www.healthit.gov/policy-researchers-implementers/interoperability A Shared Nationwide Interoperability Roadmap version 1.0 [PDF – 3.7 MB]

- Leveraged open standards and controlled terminologies to advance Patient-Centered Oriented Research.
- Tested methods and tools developed by the collaborative on the universal CDM mapping and transformation approach.

III. PROJECT DELIVERABLES

The following deliverables are included as part of the IBM contact with FDA and help provide insight into the scope of work accomplished during Fiscal Year 2018.

Relevant Work Package	Deliverables
1. Project Management & Oversight	Kickoff meeting presentation
	Project Management Plan
	Project Status Meeting (Agendas/Minutes)
	Monthly Status Reports
	Monthly Financial Reports
2. Common Data Model Mapping Tool Development	Validate mappings developed between the CDMs and the target model.
	Document for a process for version control, whereby changes to source CDMs can be appropriately shared and incorporated in the target model.
	Create a plan for how to leverage HL7 FHIR, CDISC standards implementation.
	Draft and facilitate review of a white paper capturing the design process from NCATS for the emergent data architecture to allow transformation of various Common Data Models into the selected target model.
	Develop conceptual data model
	Develop logical and physical data models
	Spreadsheet with ASPE data elements mapped to the CDISC SDTM domains/variables
3. ETL Tool Selection Process & Installation	Conduct review of documentation of the tool selection process to automate the mapping process and provide feedback to the FDA and HHS Team.
	Complete installation of selected tool in the vendor development environment.
	Draft and finalize a white paper for the tool selection process to automate the mapping process.

Relevant Work Package	Deliverables
4. Use Case Validation & Pilot Implementation	Draft and finalize a white paper capturing the potential use of the following PCORTF investments: 1. Cross Network Delivery Service and 2. NIH CDE Repository based on material reviews and interviews with the program managers for these efforts.
	Implement selected tool, including configuration in the vendor environment and implementation of the mappings developed and reviewed in Task 2.
	Create two (2) standardized sets of data one (1) in a CDISC SDTM format (e.g., SDTM) and one (1) using a HL7 FHIR Resource format

IV. KEY COMPONENTS OF COMMON DATA MODEL HARMONIZATION

A. HARMONIZATION MAPPING APPROACH

The initial phase of the project dealt with harmonizing all the data elements from each of the chosen CDMH models. The models chosen were Sentinel v6.0.2, PCORnet v3.1, i2b2-ACT v1.3, and OMOP v5.2. During the project, it was decided to harmonize PCORnet v4.0 as well as some of the data partners had already or would be implementing this new version. NOTE: Towards the end of the project, it was discovered that the listing of data elements that we had for i2b2-ACT was incorrect which changed the final harmonization but did not affect the approach described here.

To harmonize all the disparate models, the BRIDG model was chosen as the common model. For each of the models, an initial BRIDG Mapping spreadsheet was created. Each model's data elements were imported as the Mapped Specification source. Then each data element was analyzed and mapped to an existing BRIDG element, if one existed. This analysis looked at the source model's definition of the element, any vocabulary that was provided for the field, as well as sample data to see exactly how the element was used. For each element that mapped to an existing BRIDG element, the BRIDG class, BRIDG element, and BRIDG mapping path were entered into the mapping spreadsheet. For model elements that did not map to an existing BRIDG element, a proposed element was entered. In some cases, these proposed elements were additions to existing BRIDG classes while in other cases a new BRIDG class was also proposed.

Once each of the models had been mapped to the BRIDG model, a Consolidated Mapping spreadsheet was created. This consolidated mapping spreadsheet combined all the mappings. There was a tab for each of the individual model mappings and one tab that listed the important information from each mapping. Review of the consolidated mappings was done to ensure that all mappings were consistent and that model elements that were mapped to the same BRIDG element were equivalent to each other.

Once the Consolidated Mapping spreadsheet had been finalized and reviewed and feedback incorporated back into the mappings, a CDMH Conceptual model was created in Enterprise Architect. This Conceptual model copied the mapped BRIDG classes and elements as well as any new BRIDG classes, associations, and elements. All the additions were marked with a CDMH stereotype.

The final Consolidated Mapping spreadsheet and the Conceptual model diagram were sent to the BRIDG team for review. For each of the additions, a rationale for the addition and a definition of the new element was presented. This review with the BRIDG team resulted in some changes to the proposed elements as well as some changes to the mappings as suggested by the BRIDG review team. Once the review was complete, the mappings and the Conceptual model were considered finalized and were submitted to the BRIDG team for incorporation into BRIDG v5.1. The consolidated CDMH semantics were harmonized with BRIDG R 5.0.1. At end the BRIDG Harmonization process, the CDMH project added 2 new classes and 17 new attributes to the BRIDG model. The BRIDG team released BRIDG 5.1 in March 2018 which incorporates the CDMH model semantics. BRIDG Release files can be downloaded from the BRIDG Website.

B. DATABASE DESIGN APPROACH

It was decided at the beginning of the project to build the database using a model-driven approach. Models would be created in Enterprise Architect and the database utilities present in Enterprise Architect would be used to generate Data Definition Language (DDL) files to create the physical database.

Starting with the BRIDG Conceptual model, a CDMH Logical model was created. This model simplified the datatypes and used model-friendly names as opposed to the BRIDG element names. A set of Code Table and Query tables were also added in the Logical model. Some of the simplifications were:

- Convert interval datatype elements into separate start and end elements
- Address parts were broken out into separate elements
- All codes had ‘Code’ or ‘Unit’ at the end of the names
- Any class specializations were collapsed into separate classes

As well, to deal with Oracle’s name length restrictions, class and element names were shortened to a maximum of 13 characters.

Once the Logical model had been validated, Enterprise Architect’s capabilities to create a Data model from a Class model were used to create the CDMH Physical model. The stock templates were modified to provide the following changes:

- Different string types were mapped to different sizes:
  - EN (names) became VARCHAR2(100)
  - II (identifiers) became VARCHAR2(50)
  - TEL (telephone/email) became VARCHAR2(50)
  - ST (normal strings) became VARCHAR2(255)
  - ST.LONG (long strings) became VARCHAR2(4000)

- Foreign key links were auto-generated from any Coded values to the CDMH code tables
- Database metadata was auto-added to every class (create_date, create_id, modify_date, modify_id)

Finally, once the Physical model had been created and validated, Enterprise Architect's capabilities to generate Oracle DDL were used. As with the data model templates, the stock DDL templates were modified:

- Add sequences to all primary key elements
- Add indexes on all foreign keys

The resulting Oracle DDL was passed on to the development team who used it to create the CDMH tables. Any changes that were needed were passed back and made in the CDMH Logical model and then the Physical model and DDL were then regenerated. This ensured that the Logical model, the Physical model, and the CDMH database were always kept synchronized.

C. DATA LOADING

The content below depicts the process of loading data provided in response to the originated query to the BRIDG data model.

The Adeptia ETL (Extract, Transfer, Load) Suite was selected as a software to be used for implementing the CDMH ETL process. Adeptia ETL is a graphical, easy-to-use data mapping solution used for aggregating data from multiple sources to populate databases and for business intelligence solutions. It provides a comprehensive solution that combines data transport with metadata management and data transformation capability.

Adeptia CDHM ETL workflow was developed to accommodate the task of loading data for Aggregate and Patient level queries. The application can create queries, translate vocabularies, load data and review the results by query requestor on a query-specific base. The entire process includes the following steps:

- Creating the query and translating vocabulary to specific data models;
- Creating and transferring (SFTP) query files;
- Loading the query response to staging area;
- Transforming received data to BRIDG vocabulary and loading data.

The data loading is completed in two steps: 1) loading the response data 'as is' to the Staging area and 2) translating the response to BRIDG vocabulary and loading the actual query results. Data Model-specific and BRIDG vocabularies (as well as corresponding mappings) are stored in the data base and are accessible by Adeptia ETL. The code mapping process integrated in the ETL allows for the transfer of vocabularies from one system to another. Populations of Data model-specific queries are then converted possessing all required values. Successful execution of each steps automatically updates query status, allowing user to oversee and manage the entire ETL process. While the current process

requires manual triggering of each task in Adeptia ETL tool, this will likely become fully automated in the next phase of the project.

D. QUERYING

The CDMH application has the capability to create and review customized queries for both Aggregate and Patient-level health data. The Adeptia Suite Web Forms capability was used to create the CDMH Query Builder and Query Manager components.

The developed Query Builder component allows the user to enter query name and description and specify the type of query, parameter and aggregation. The validation process checks the integrity of the query after all mandatory elements are entered. When the query is submitted, information is displayed in the Query Manager view.

The Query Manage component allows for saving, viewing and editing of the query, as well as the ability to review results. Once the query is submitted, it cannot be deleted from the queries list.

E. EXPORTING

The Query Manage component also has the capability to export data. Once the data loading process is completed and the query status is set to "Transaction Completed", the Aggregate-level query can be exported to a .CSV format.

Patient-level queries can be exported to .CSV format for all user-selected elements of the returned data. In addition, the system allows for exporting query results to SDTM format.

F. MAPPING CDMH TO STANDARDS

One of the objectives of the CDMH Project was to demonstrate the capability to export data in CDISC SDTM and HL7 FHIR data exchange formats. To accomplish this task, the CDMH database model semantics were mapped to CDISC SDTM 3.2 and HL7 FHIR Resources 3.0.1.

1. CDISC SDTM

One of the benefits of using BRIDG as the common model was the fact that BRIDG model is already mapped to the CDISC SDTM standard. This effort started with getting a BRIDG to SDTM mapping report and then reviewing an updating the mappings as need to provide the mapping for SDTM 3.2 This was a very detailed mapping effort. Since SDTM is primarily used to exchange clinical trial data, the mapping of clinical care use case-based CDMH repository model to SDTM involved making a few assumptions and setting default data points. The CDISC SDTM mappings were added to the CDMH Consolidated Mapping spreadsheet and were leveraged by the Adeptia team during transformation and export of CDISC SDTM.

2. HL7 FHIR

The BRIDG team had started an effort an effort to map the entire BRIDG model semantics to the HL7 FHIR artifacts under a FDA U01 Grant. Since some of the same team members were on the CDMH project, the BRIDG 5.1 to FHIR 3.0.1 work-in-progress mapping was leveraged for this effort. This CDMH

to HL7 FHIR mapping spreadsheet has three components - The CDMH subset of the BRIDG model (the conceptual model), the CDMH physical model and the HL7 FHIR 3.0.1 artifacts. CDMH specific comments were documented for the various mappings and gaps were identified. There were many instances identified during the mapping process where a CDMH semantic did not have an equivalent in HL7 FHIR artifacts. These were identified as GAPS. These gaps could be considered in the future as requirements for creation of extensions to the existing FHIR resources. The mapping document was shared with the ONC team of this effort. That team is currently in the process of developing CDMH FHIR profiles and creating the necessary extensions. This work is being done under the governance of the HL7 Biomedical Research & Regulation (BR&R) work group and is following the HL7 FHIR development processes.

V. PROJECT ACCOMPLISHMENTS

A. ARCHITECTURE DESIGN AND SOFTWARE DEVELOPMENT

Software architecture was developed and is being implemented which will allow investigators to build a single query for distribution to the four Common Data Model (CDM) networks (PCORNET, OMOP, ACT/i2b2, and Sentinel) and allow analysis of received results via a portal. This software was developed utilizing the Adeptia ETL software package and is currently housed in a cloud-based environment hosted by NCATS where it will continue to reside after the initial software development. Additional non-cloud implementation options are also being explored.

B. DATA MAPPINGS FOR TRANSFORMATIONS

As foundational work that powers the CDMH software architecture, alignment and gap analysis was performed to cross-map the data relationships between the CDMs, the transitional data model (BRIDG), and current releases of HL7 FHIR and CDISC STDM transport standards. These mapping documents, in and of themselves, will be useful tools for cross-CDM analyses, FHIR implementations using the CDMs, and relevant submissions to FDA using SDTM.

C. TESTING OF DATA MAPPINGS AND IMPLEMENTED ARCHITECTURE

To test and validate of the mapping and software implementation in the Adeptia-driven CDMH solution, healthcare data partners contracted by FDA will assess the CDMH functionality using a post-market research use case assessing the safety profile of oncology immunotherapy drugs on populations with autoimmune disorders. Testing will involve comparison of results generated by direct manual queries of data in different Common Data Models with results generated by the Adeptia-driven CDMH solution.

D. RESOURCES FOR PCOR RESEARCH COMMUNITY SUPPORT

To provide ongoing support for researchers of many types including PCOR researchers and those at NIH, FDA, and others, many created resources will be available to the public for use. In addition to the CDMH solution itself and supporting educational materials for use, access to the foundational cross-mappings

are available both in spreadsheet form as well as through public portals at NCI and NLM (currently under development).

E. PLANNING FOR FUTURE DEVELOPMENT AND SUSTAINABILITY

To ensure subsequent development of the CDM Harmonization solution, associated documentation, and artifacts, the CDMH Project team is actively exploring additional funding sources to provide resources for future work. Additionally, documentation of recommendations for data governance and future paths forward based on best practices and lessons learned are being developed as guides for best utilizing any new funding sources. Finally, while the initial hosting of the CDMH solution in the NCATS-owned cloud space will continue after initial development, additional implementation options, such as packaging the pre-configured CDMH solution for local research installation, are being explored for feasibility.

VI. LESSONS LEARNED

In this section, we describe lessons learned through the CDMH project and how we might carry this learning through to other projects.

A. PROJECT DELAYS DUE TO CONTRACT AGREEMENTS AND ETL TOOL PROCUREMENT

One of the main complications of the contract between FDA and IBM was the reliance on procuring Adeptia as the ETL tool. During the FY18 contract period, IBM was able to perform the initial functions of the statement of work – essentially, to work with Subject Matter Experts to develop the common data model mappings over the first half of the project, through March and April of 2018. At this point, the expectation was that the IBM and Digital Infusion technical teams would have been fully trained and ramped up on the Adeptia ETL tool to begin development and implementation activities through Summer 2018.

However, FDA and IBM had to wait for approval of the Adeptia contract, managed by Leidos on behalf of NCI, which was delayed several months in part due to Adeptia being relatively new in the government contracting space. By the time the Adeptia contract was awarded, the FDA, IBM, NCI and Digital Infusion teams had to scramble to condense a training, development and implementation period of only a few months.

It was essential that the FDA and IBM teams were able to work out a no-cost Project Change Request (PCR) to extend the contract through Calendar Year 2018 and were able to negotiate and prioritize the highest priority data models to be implemented first. Given resources plans to move on to other work beginning in October 2018, the team was able to achieve a commendable amount of work in November and December 2018 to still satisfy the main project requirements. Thankfully, the remaining implementation items that could not be completed per the FDA/IBM contract (e.g. I2B2/ACT implementation activities) can be carried out in a follow-on contract between the Digital Infusion and NCI Teams.

B. PROJECT DELAYS DUE TO TECHNICAL CONSTRAINTS

Even after the Adeptia tool was procured by NCI, there were significant blockers preventing the IBM technical team from being able to install and use the software.

First, access to the NCATS AWS servers was restricted for users with FDA/CDER issued laptops by FDA Firewall security settings. The attempt to get a Firewall exception turn out to be a long and complicated process – even after escalating the request up to the decision-making level, we were ultimately unable to make a compelling enough case to bypass the firewall. Several tickets were opened over several months to resolve the issue, but to no avail, in part due to complication with creation of non-static IP addresses for FDA developers. Finally, the workaround of accessing the server by not logging into the FDA VPN allowed development to continue, but this was something that could have been pursued weeks earlier, if it were made clear up front the team was not going to be given a waiver to bypass the Firewall.

Second, once Adeptia was procured and installed on AWS servers, there was an issue with the Java setting on FDA laptops, causing Java plug-in components such as Process Flow Designer and Data Mapper to not open properly. Numerous sessions with the FDA Helpdesk and the Adeptia product team were held to ultimately troubleshoot and resolve the issue prior to Product Training, but this came at a significant use of resources to resolve a seemingly simple problem.

C. CDISC SDTM MAPPING

One of the known challenges for exporting data from CDMH Repository to CDISC SDTM was the fact that CDISC SDTM is a data exchange standard for submitting clinical trial data to the FDA. The patient level data that CDMH had access to was coming from EHR's and was not research/clinical trial data. The project team made some assumptions and created default data for few data points to demonstrate the export to SDTM functionality. Moving forward. One recommendation is to work with CDISC and identify/develop new SDTM domains/variables that would support using SDTM as a potential data exchange in non-research use cases also.

VII. PROPOSAL FOR FUTURE WORK

As the CDMH project was ramping down in the second half of 2018, the FDA, IBM and NCI teams began to discuss follow-on applications to enhance the retrieval and process of Real-World Data (RWD) and Real-World Evidence (RWE). With remaining time and funding on the CDMH contract, one of the IBM team members began working on a prototype called "SMART on FHIR", which recognized HL7 FHIR as the established leader in the healthcare data interoperability space. The following sections glean on what was learned from CDMH, and how future teams (presumably, the team comprised of NCATS and Johns Hopkins resources for FY19) could view the prototype to gather requirements for future development work.

A. OVERVIEW OF THE FHIR PROTOTYPE

The approach implemented by the CDMH project involved establishing interfaces with four different providers/aggregators of healthcare data, so called CDMs (Common Data Model). Each CDM is implemented using its own data model and its own way of querying the data. This required building a translation layer for each of the providers using SQL as the "common ground". This approach has several shortcomings:

- Limited number of data providers. While the four CDMs have broad participation in the industry, it is still only a subset of the clinical data universe.
- CDM providers are intermediaries and they themselves depend on their members (hospitals and universities) providing them with the data. As a result, CDMs contain only a subset of clinical information available from a given EHR.
- Developing logic to translate queries to four different data models using different SQL dialects (or SAS as required by Sentinel) is time consuming and error prone, especially without direct access to the database where the resulting query is intended to run.
- Dependency on the low-level details of the physical data models. Physical data models tend to change substantially over time, so this approach will result in high maintenance cost. This affects the ability to reliably run the translated queries (the translation logic must be updated in case of changes) and performance/response time, in case if the underlying physical DB is changed (e.g., changes to its indexes).
- Slow response time from CDMs; effectively CDMH had to rely on the providers running the queries manually.

A more scalable and more general solution calls for a standard-based mechanism for obtaining healthcare data. Fortunately, there is now a clear leader in the healthcare interoperability space, the HL7 FHIR standard. It makes sense to use this standard for all future developments of CDMH or a successor of the CDMH project.

FHIR is a broad standard with a lot of flexibility and with deep support for customization and extensibility. For example, it supports different bindings, including JSON and XML. To achieve broad interoperability, a more strictly defined subset of FHIR must be used.

SMART on FHIR (https://smarthealthit.org/) provides such a subset. In a nutshell, SMART on FHIR has the following key characteristics:

- Based on JSON
- Standardizes on Argonaut FHIR profile (http://argonautwiki.hl7.org/index.php?title=Main_Page)
- Uses OAuth2 as the authentication/authorization mechanism. This is the key since it security has always been a very difficult problem in establishing interoperability in healthcare.

SMART on FHIR has gained a lot of momentum and it is currently supported by several major EHR vendors. Also, OMOP and i2b2 have already implemented support for SMART on FHIR (http://omoponfhir.org/, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5961782/).

Therefore, the goal of the prototype was to implement an interface with a SMART on FHIR back-end. Specifically, the prototype had to perform the following steps:

- Authenticate to the back-end
- Submit a query to select data from several different FHIR resource types
- Save the data locally or in a non-SQL DB.

To achieve these goals, the prototype had to support a SMART on FHIR dialect specifically designed for system-to-system interfaces, called FHIR Bulk Data Access

B. OVERVIEW OF THE FHIR PROTOTYPE

FHIR Bulk data access standard (https://github.com/smart-on-fhir/fhir-bulk-data-docs) is an effort/standard to make it easier to obtain FHIR-encoded data using APIs. It is intended for system-to-system scenarios, meaning that the client (a FHIR data consumer) represents another system (e.g. a CDMH back-end) and that the communication doesn't have to be initiated from a UI.

This is distinctly different from a "regular" SMART on FHIR use case where the client is an app running in a browser or on a mobile platform. A SMART on FHIR app is typically launched by the EHR itself. This distinction is mostly determined by the supported authentication flows. SMART on FHIR mandates the authorization code grant flow (https://auth0.com/docs/api-auth/which-oauth-flow-to-use) which relies on browser/HTTP redirects. The flow is documented in detail here: http://hl7.org/fhir/smart-app-launch/

FHIR Bulk uses so called "client credential flow" that does not require any UI or an end-user involvement. Like mutual SSL authentication, it relies on the certificate/key owned by the client. The certificate (public key) must be registered with the FHIR Bulk server before the client can start accessing it. Once this is done, the authentication request is simply signed by the client's private key and validated by the server using the pre-registered public key (https://github.com/smart-on-fhir/fhir-bulk-data-docs/blob/master/authorization.md).

The second characteristic of FHIR Bulk is that it's designed for asynchronous download of large volume of data. This is done via polling by the client. The client first submits an initial request for data, it then must poll the server periodically to get notified when the data is available for download. The notification response contains the information with the URLs to download data from. Each FHIR resource type has its own URL.

] Note that this is very different from the traditional synchronous HTTP API-based model and much more suitable for transferring large amount of data to the client.

C. WHAT WAS IMPLEMENTED?

The prototype was implemented in Python 3 using open source python libraries such as "request" for HTTP calls. "jwt" library was used for authentication (jwt is the token format used by OAuth2).

The core of the prototype is the client (bulk_client module) that implements all the steps required by the FHIR Bulk standard. The client performs the following steps:

- Authentication using a pre-configured key (https://auth0.com/docs/jwks), which is used as the credential to gain access to the server.
- Initial request with the provided query (see below)
- Periodic polling waiting for the fulfillment of the request by the server
- Parsing of the initial response. The response contains the URL for each of the requested resource type.
- Getting the actual data (FHIR resources).

The prototype's "main" module invokes the client and then simply saves returned resources in a directory on disk. Each resource instance is saved in its own file in JSON format. The "main" module runs from the command line and supports several command-line parameters, including FHIR resource types, where to save the data, etc.

For more details on how to run the client, please see the README file.

Note that a FHIR Bulk server returns the data in a new line - delimited JSON format (http://ndjson.org/) where each line is a valid JSON document (a FHIR resource instance). This allows for streaming the data without having to download the entire (well-formed) JSON document. Our client handles it. Each resource instance, however, is saved as a separate file by the prototype. Each resource type is saved in its own sub-directory.

The client was tested against the SMART Bulk Data Server (https://bulk-data.smarthealthit.org/index.html) provided by the SMART on FHIR organization. The server provides a good representative sample of FHIR resources based on the " Common Clinical Data Set". We tested querying for several different resource types, including Patient, Observation, etc.

D. HOW DO YOU QUERY THE DATA USING FHIR BULK?

The ability to query data using various criteria is obviously very important for CDMH or any similar effort.

FHIR Bulk currently supports only the ability to query by the resource type or by the date: https://github.com/smart-on-fhir/fhir-bulk-data-docs/blob/master/export.md#query-parameters

There is also an experimental support for a "_typeFilter" parameter that provides much more sophisticated querying capabilities (https://github.com/smart-on-fhir/fhir-bulk-data-docs/blob/master/export.md#example-request-with-_typefilter). However, "_typeFilter" does not seem to be currently implemented by the SMART Bulk Data Server and so it has not been tested from the prototype. Judging from the examples of the "_typeFilter" query, it would satisfy the requirements of the CDMH use case. An alternative that would require more effort is to download all the necessary resources constrained by the date and then perform further filtering on the CDMH side.