

Research and Applications

Beyond compliance with the 21st Century Cures Act Rule: a patient controlled electronic health information export application programming interface

Dylan Phelan, MS¹, Daniel Gottlieb, MPA^{1,2}, Joshua C. Mandel, MD^{1,2}, Vladimir Ignatov, MFA¹, James Jones, MPhil¹, Brett Marquard, BS³, Alyssa Ellis, MS¹, Kenneth D. Mandl , MD, MPH^{*,1,2}

¹Computational Health Informatics Program, Boston Children's Hospital, Boston, MA 02215, United States, ²Department of Biomedical Informatics, Harvard Medical School, Boston, MA 02115, United States, ³Waveone Associates, Amherst, MA 01002, United States

*Corresponding author: Kenneth D. Mandl, MD, MPH, Computational Health Informatics Program, Boston Children's Hospital, 401 Park Drive, LM5506, Mail Stop BCH3187, Boston, MA 02215, United States (kenneth_mandl@harvard.edu)

Abstract

Objective: The 21st Century Cures Act Final Rule requires that certified electronic health records (EHRs) be able to export a patient's full set of electronic health information (EHI). This requirement becomes more powerful if EHI exports use interoperable application programming interfaces (APIs). We sought to advance the ecosystem, instantiating policy desiderata in a working reference implementation based on a consensus design.

Materials and Methods: We formulate a model for interoperable, patient-controlled, app-driven access to EHI exports in an open source reference implementation following the Argonaut FHIR Accelerator consensus implementation guide for EHI Export.

Results: The reference implementation, which asynchronously provides EHI across an API, has three central components: a web application for patients to request EHI exports, an EHI server to respond to requests, and an administrative export management web application to manage requests. It leverages mandated SMART on FHIR/Bulk FHIR APIs.

Discussion: A patient-controlled app enabling full EHI export from any EHR across an API could facilitate national-scale patient-directed information exchange. We hope releasing these tools sparks engagement from the health IT community to evolve the design, implement and test in real-world settings, and develop patient-facing apps.

Conclusion: To advance regulatory innovation, we formulate a model that builds on existing requirements under the Cures Act Rule and takes a step toward an interoperable, scalable approach, simplifying patient access to their own health data; supporting the sharing of clinical data for both improved patient care and medical research; and encouraging the growth of an ecosystem of third-party applications.

Key words: electronic health records/standards; health information interoperability/standards; digital technology.

Background and significance

The 21st Century Cures Act requires application programming interface (API) access to all elements of a patient's electronic health record (EHR) "without special effort."¹ It further states that EHR developers "will not take any action...that may inhibit the appropriate exchange, access, and use of electronic health information,"¹ a practice defined as information blocking.² It specifically defines, as an instance of information blocking, "Implementing health information technology in ways that are likely to restrict the access, exchange, or use of EHI with respect to exporting complete information sets or in transitioning between health information technology systems."¹

The Office of the National Coordinator of Health Information Technology (ONC) 21st Century Cures Act Final Rule implements several data-access provisions in the Cures Act.³ Since December 31, 2022, EHRs must support standard APIs for both one-patient-at-a-time and bulk export^{4,5} of a subset of EHI electronic health data called the US Core for Data

Interoperability (USCDI).⁶ Beginning December 31, 2023, EHR developers must implement functionality for patients to request and receive complete electronic copies of their EHI. Also, when switching to a different EHR system, providers must be able to export the full content of all of their patients' EHI. However, under the Cures Act Rule, EHR developers do not need to offer EHI export using a standardized API. The Cures Act Rule only requires that the export be provided in a computable, electronic format with documentation available via hyperlink.⁷ This decision apparently reflects ONC's cost/benefit analysis, but we believe the regulatory bar is too low to ensure national scalability.

Some EHR vendors released links to their EHI export documentation ahead of the requirement date. Epic published EHI Tables Export⁸ which uses Epic-specific relational-table formats, comprising more than 6500 different tab-separated value table formats and separate downloads for rich-text documents and images. AthenaHealth published athenaOne EHI Export⁹ which references different data elements across ambulatory and in-patient environments, extends the set of

core FHIR resources with custom definitions, and exports EHI as new-line delimited JSON (NDJSON) and HTML files. Although these services may at a technical level make data available, invoking these exports will burden patients with a complex workflow including manual file download and management that can vary across EHRs and healthcare institutions.

Objective

We sought to advance the ecosystem, instantiating policy desiderata in a working software prototype based on a consensus design. We formulate a model for interoperable, patient-controlled, app-driven access to EHI exports, instantiated in an open source prototype.

For patient-directed information exchange, the capabilities required under the Cures Act Rule would be more powerful if a patient could use an app to make an API request for their complete EHI. An app here refers to any user-authorized software that interacts with the API, and can include native or web-based software for any platform. There are myriad use cases for this new model of EHI Export, such as sharing healthcare data with a clinician, subjecting one's health data to artificial intelligence (AI) based processes to provide personalized decision support, and data donation for research and innovation.^{10,11} To that end, our team, SMART Health IT,¹² proposed and led development of a standards-based EHI Export API specification as part of the Health Level Seven (HL7) Argonaut FHIR Accelerator. This led to development of a consensus implementation guide (IG), which is a set of rules about how FHIR resources are used (or should be used) to solve a particular problem with documentation to support and clarify usage.¹³ From this IG we sought to develop our open source reference implementation, which is a standard or authoritative example of how a software specification should be implemented. This reference implementation consists of an EHI export server, a sample client web application for making EHI exports, and an interface for staff at provider organizations to manage and modify these export requests.

Methods

This work occurred as part of a cooperative agreement with ONC and is based on two prior successful technologies. The first is the SMART on FHIR API^{4,14,15} which standardizes authentication and authorization of third party apps to EHRs, thereby giving access to defined data resources represented as Fast Healthcare Interoperability Resources (FHIR), a modern format for representing health data. The second is SMART/HL7 FHIR Bulk Access (Bulk FHIR),⁵ an API that lets organizations transmit large datasets in FHIR. Because Bulk FHIR already implements an asynchronous request pattern for large-scale data on demand by authorized clients, this was our technical starting point.

To build consensus on the API specification in the form of a FHIR implementation guide (IG), we followed the 'recipe' of prior projects and began with a proposal to Argonaut,^{16,17} a consensus group launched in 2014 to support patient-oriented FHIR API recommendations and mandates. Convened by Health Level Seven (HL7), an ANSI accredited standards organization, Argonaut membership includes EHR vendors, healthcare providers, payors, technology companies,

and the SMART Health IT team. Every year, Argonaut selects priority FHIR standardization projects to be tested with operational open source reference models and/or EHR systems. This ensures that the proposed designs are limited to feasible solutions and that sample software is primed for adoption and utilization. Argonaut was the first of the "FHIR Accelerators."¹⁸

In 2022, with encouragement by the ONC, Argonaut members prioritized the EHI Export API project to develop a draft IG. IGs are used in the FHIR community to define use case specific specifications that build on the core FHIR specification. Development of the draft IG¹⁹ from the initial SMART team proposal occurred over a series of six meetings, along with interviews with four subject matter experts. Four EHR vendors, three health systems, and many other stakeholders participated. The guide incorporated: (1) SMART App Launch Authorization for access control; (2) the FHIR Async API leveraged by Bulk FHIR for asynchronous export management; (3) a DocumentReference profile to describe exported files; and (4) an optional filter step allowing patients to narrow their EHI request.

As a brief technical overview, the draft IG describes an EHI export flow where a client app uses the SMART App Launch API to connect to an EHI server. The client initiates an EHI export with an HTTP POST request at a kick-off endpoint, "[fhir base]/Patient/{id}/\$ehi-export." The EHI server returns an HTTP response with a "Content-Location" header, providing a URL that clients can poll for updates to the export request. Additionally, depending on its configuration, an EHI server may return a "Link" header with the "rel='patient-interaction'" value, pointing the client to an EHI export customization form for the user to complete before exports proceed. After the form is completed, the EHI server aggregates EHI in a way that suits its backend workflow. When the export is complete, the client app polling requests will yield responses with "200 OK" status codes, containing a JSON-formatted manifest in the response body. This manifest details the download location of every EHI-exported FHIR resource, aggregated and formatted as NDJSON. Any file using a proprietary format is wrapped in a FHIR DocumentReference resource for consistent presentation of metadata.

The four additional design goals for the API that emerged from the Argonaut development process were: (1) support integration with manual workflow steps such as review, redaction, or format conversion by medical records staff; (2) support vendor-specific capabilities and provider-specific forms allowing patients to filter or limit the volume of EHI exported; (3) support returning EHI data with consistent FHIR-based metadata wrapping a combination of standardized or proprietary export files; (4) support internal use of the API by IT or automated systems to stitch together data from multiple certified EHR systems. The anticipated user experience is that the patient signs into a web or mobile application and connects the app to the EHR of their healthcare provider. After initiating an export, the app tracks the export process and transmits or downloads the resulting files.

Results

To continue developing the EHI Export API standard, we created a reference implementation, a standard or authoritative example of how a software specification should be implemented. Our reference implementation, based on the

Argonaut IG, supports two user workflows as part of the overall EHI request. One is a patient requesting their EHI be exported from their healthcare institution and shared with a third party web application. Another is the medical records staff (“admins”) of a healthcare institution reviewing EHI export requests and manually adding EHI that cannot be retrieved by means of APIs, such as hospital records not yet integrated into an EHR. The architecture has three central components: a sample patient web application for requesting EHI exports, an EHI server for responding to EHI export requests, and an admin export management web application for managing requests (Figure 1).

EHI server

The EHI server component is a reference implementation for the EHI Export API’s server specification. It contains synthetic patient data created using Synthea²⁰ and includes the logic for creating and processing EHI export requests for a hospital. The server supports SMART on FHIR’s SMART Authentication and builds on the structure of the SMART/HL7 Bulk FHIR Access API’s response format. A patient requesting EHI export from our server will receive the USCDI elements in structured FHIR format, together with any additional data indexed using FHIR DocumentReferences. Our

server supports the management of EHI exports using both manual and automated workflows. This reference implementation serves as a template for FHIR server developers or EHR systems developers who want to support the EHI Export API in their own systems. Its functionality is demonstrated by our accompanying sample client web application.

Sample patient web application

Our patient-facing user interface (UI) component is a sample client web application that implements the EHI Export API’s client interactions. Consisting of a UI and its own backend server, this application allows users to create new EHI export requests, check on the progress of these exports, and download completed exports, containing synthetic patient data, directly onto their machines.

Patient workflow

Our sample patient web application simulates a patient collecting their EHI from one or more current providers to send to another provider for a second opinion. This patient workflow begins with users logging in. Successful login presents users with the screen in Figure 2, a list of existing exports they created, the status of those exports, a button for creating new export requests, and available interactions based on the

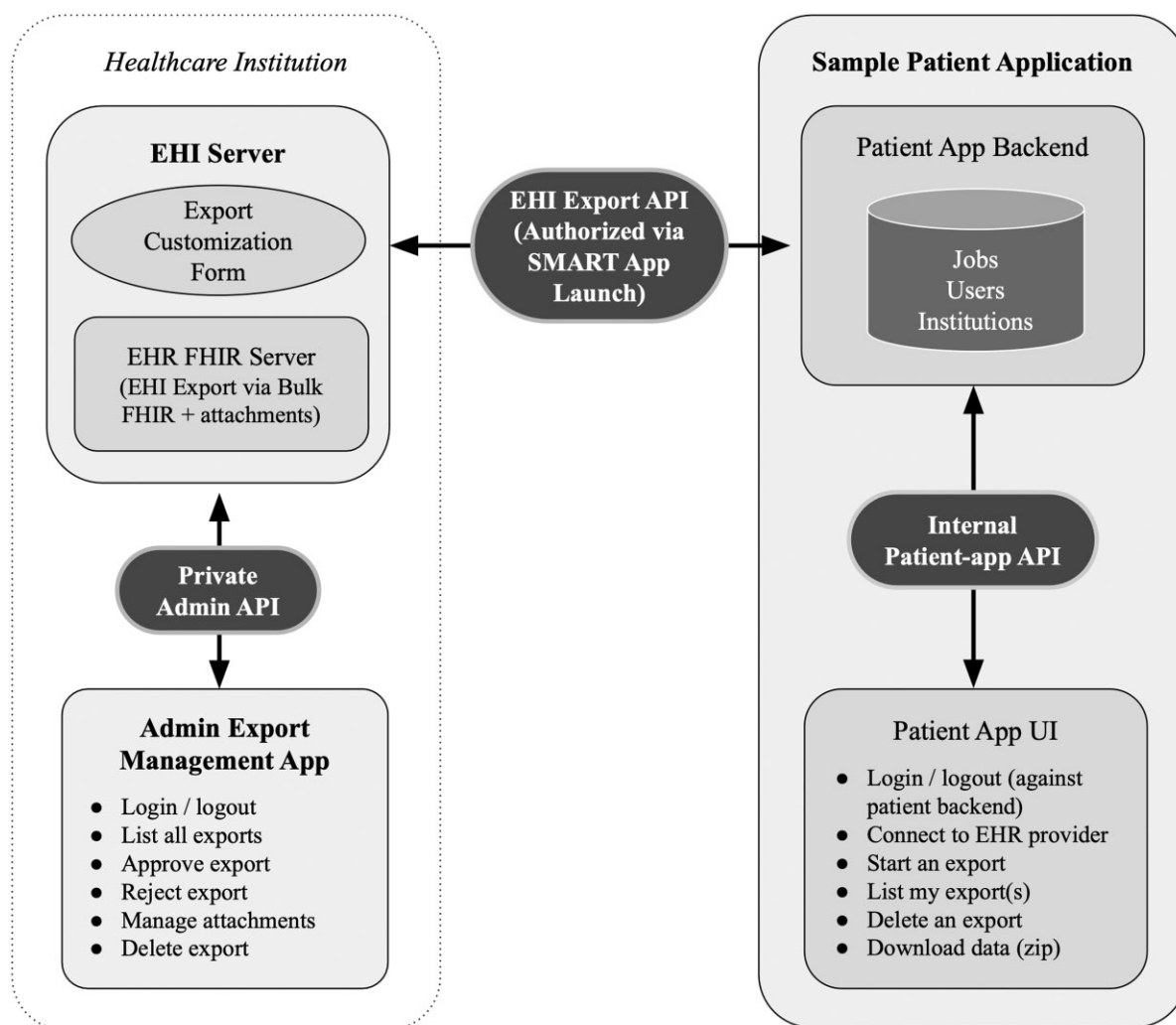


Figure 1. Architecture diagram visualizing relationships among the three central components of our reference implementation: EHI export server, sample patient-facing web application, and admin export management web application.

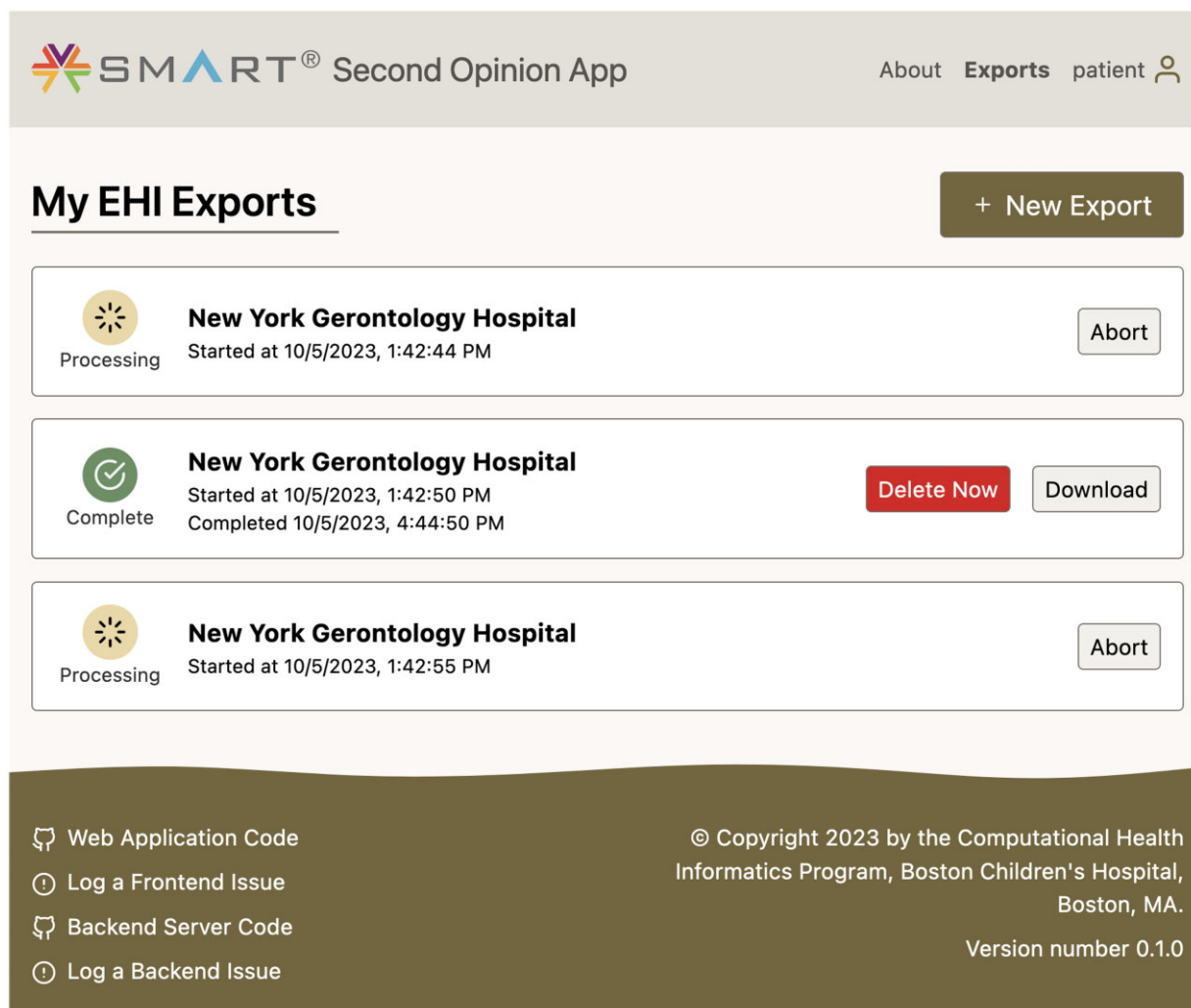


Figure 2. The patient-facing sample web application exports EHI data from a synthetic EHR system to send to others for a secondary health opinion.

status of the request. Users can abort “Processing” exports, delete “Rejected” exports, and download EHI from successfully “Completed” exports, with the additional option to delete “Completed” exports. This page will continue polling for export request updates at a server-recommended rate, so that any status changes are automatically presented in the UI.

Clicking “New Export” presents users with a list of institutions from which they can select those appropriate for their request. When they select an institution, the SMART App Launch process begins, allowing users to select a demo patient and authorize data access. Users are then redirected to complete the EHI server’s customization form (Figure 3). After submitting this form, users are redirected back to the export list in Figure 2—now showing one more export. As export requests are completed by hospital staff, patients will see status changes and a “Download” button for successful exports. Clicking the button initiates the download of the patient’s EHI bundle as a ZIP file. It contains both the automatically retrieved and manually attached EHI (see Admin Workflow) for the export request. Patients can then share this data with a clinician (or AI) to receive a second opinion. In the case of an app linked to the institution or a service providing the second opinion, patient sharing of their EHI could


take place entirely within the app, without requiring them to download their data.

Admin export management web application

Our admin export management web application demonstrates how medical records staff (“admins”) can view EHI export requests, add attachments with EHI data manually when necessary, and review them using custom EHI server API endpoints. Although some of these UI interactions are not prescribed by the EHI Export API IG, they allow exploration of organization-specific EHI export workflow choices, reflecting how export management applications would be implemented in practice.

Admin workflow

This admin workflow enables healthcare provider organizations’ internal work queue for managing EHI export requests, reflecting manual steps that real-world export implementations might include. After login with admin credentials, admin users see all exports being processed by the EHI server (Figure 4). Status indicators communicate where each request is in the export process, along with export metadata: the export’s ID, the patient whose data is being exported, when it


EHI Export Server

INFORMATION TO BE RELEASED

Please check all that apply, and optionally specify date ranges

Electronic Records

	From	To	Notes
<input type="checkbox"/> Clinic Visits	mm/dd/yyyy	mm/dd/yyyy	
<input type="checkbox"/> Lab Reports	mm/dd/yyyy	mm/dd/yyyy	
<input type="checkbox"/> Billing Records	mm/dd/yyyy	mm/dd/yyyy	
<input type="checkbox"/> Other Records	N/A	N/A	

Other Documents

	From	To	Notes
<input type="checkbox"/> Discharge Summary	mm/dd/yyyy	mm/dd/yyyy	
<input type="checkbox"/> Operative Reports	mm/dd/yyyy	mm/dd/yyyy	
<input type="checkbox"/> Pathology Reports	mm/dd/yyyy	mm/dd/yyyy	
<input type="checkbox"/> Radiology Reports	mm/dd/yyyy	mm/dd/yyyy	
<input type="checkbox"/> Photographs	mm/dd/yyyy	mm/dd/yyyy	
<input type="checkbox"/> Other	mm/dd/yyyy	mm/dd/yyyy	

AUTHORIZATION FOR RELEASE OF PROTECTED OR PRIVILEGED HEALTH INFORMATION

Please indicate if you give permission to release the following information if present in your record

- ☐ HIV test results
- ☐ Alcohol and Drug Abuse Records
- ☐ Details of Mental Health Diagnosis and/or Treatment
- ☐ Confidential Communications with a Licensed Social Worker
- ☐ Details of Domestic Violence Victims Counseling
- ☐ Details of Sexual Assault Counseling
- ☐ Genetic Screening
- ☐ Other(s):

Submit

Figure 3. The patient-facing customization form allows patients to specify what information is included in their EHI export. Note that the specification of customization forms' filtering logic is out-of-scope for the EHI Export API IG. While the IG and our reference implementation illustrate how these customization forms can be used, their use is optional and their implementation details are bespoke to each institution's needs and business logic.

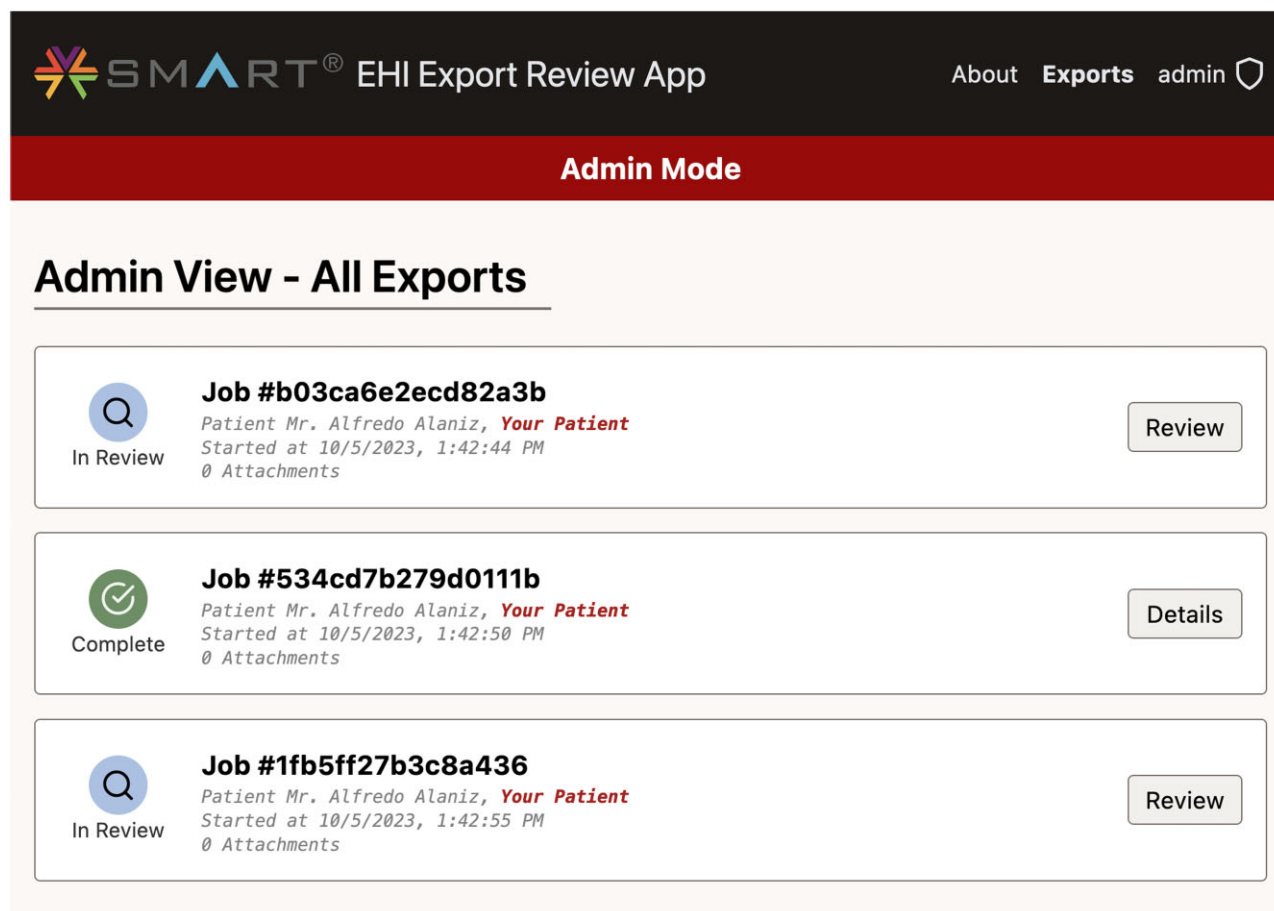


Figure 4. The admin-facing sample web application displays EHI export requests being processed, their statuses, and action buttons for medical record staff to review each request.

was started, and the number of manual attachments that were uploaded. Action buttons on the right side of the export allow users to review exports that have an “In Review” status or view details for exports that were already reviewed.

Clicking “Review,” admins see a UI displaying metadata and additional actions (Figure 5). Attachments can be uploaded via drag and drop or via the browser’s native file input element, and they can be removed afterwards. Data from the customization form are formatted as descriptive paragraphs above the upload component, cueing admins to review them for specific details of the patient’s record. Admins can reject exports, removing them from the EHI-server, or approve exports after relevant attachments have been added, enabling authorized clients to access their EHI. Upon completing the export review, users will return to the list of all exports. Clicking “Details” for reviewed exports will result in a read-only version of the UI, where status can no longer be changed and attachments cannot be modified.

Open source code

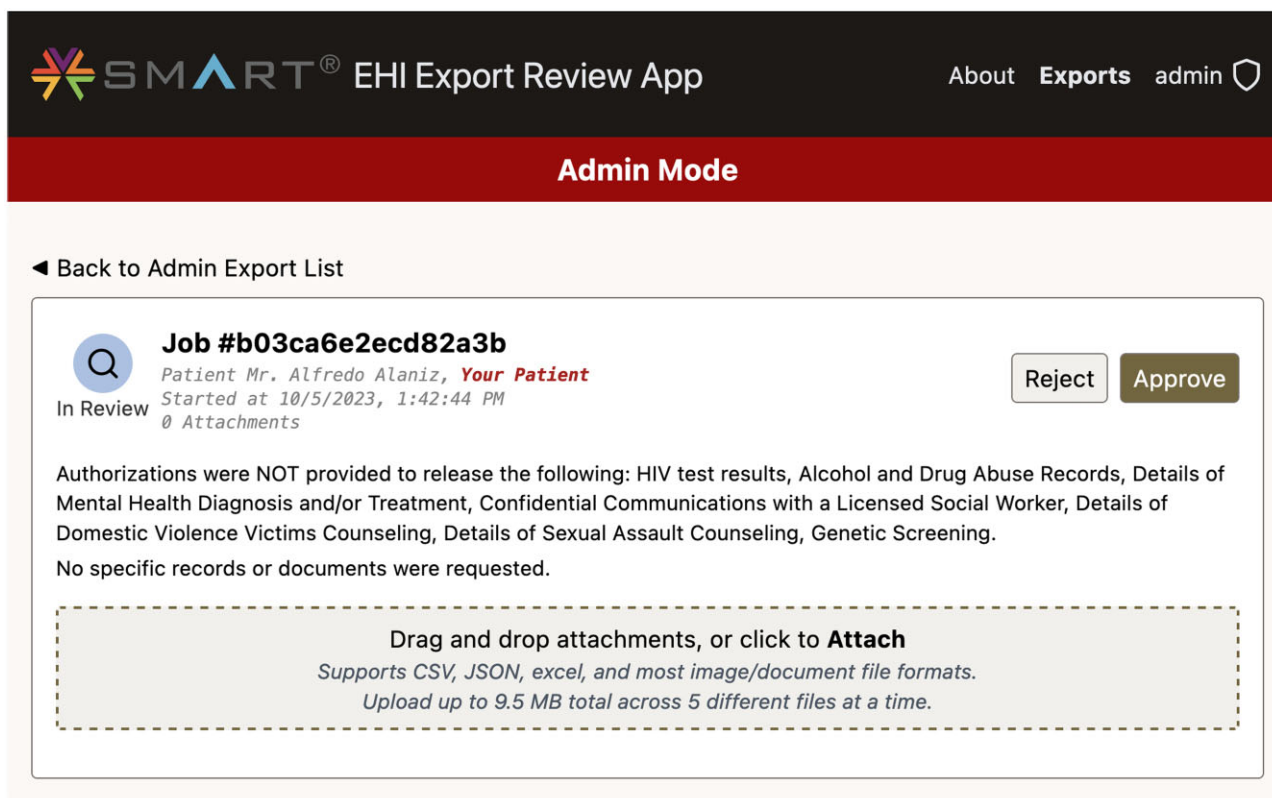
Our implementations are split across two code repositories. The EHI-Server repository, <https://github.com/smart-on-fhir/ehi-server>, contains code for our EHI server. The EHI-App repository, <https://github.com/smart-on-fhir/ehi-app>, contains code for our sample patient application as well as our admin export management application. This reference implementation demonstrates use of the EHI Export API and related user workflows, but is not intended for use in

production. The patient-facing application and hospital-facing admin application are in the same repository for development purposes; however, they are functionally distinct apps. All patient data are synthetic, generated from Synthea, <https://synthetichealth.github.io/synthea/> and do not correspond to real patients.

Discussion

We present a model formulation for substantive patient-centered data sharing on a national level. We prototyped an app designed for patient use that enable the complete transfer of EHI from any EHR through an API, as a way to encourage extension of the regulatory intent of the 21st Century Act Rule. Our aspiration is that by making an open source reference implementation of an EHI export API available, it will ignite active participation within the health IT sector, fostering the evolution of the design, its practical application in real-world scenarios, and the creation of apps that are aimed at patients.

The software is available, open source, free, and liberally licensed. Demo instances can be found online: the EHI server at <https://ehi-server.smarthealthit.org/fhir/well-known/smart-configuration>, the sample patient web application at <https://ehi-app.smarthealthit.org/>, and the admin export management web application at <https://ehi-app.smarthealthit.org/admin/>. We anticipate that release of these applications will spark engagement from the FHIR



SMART® EHI Export Review App About Exports admin

Admin Mode

◀ Back to Admin Export List

Job #b03ca6e2ecd82a3b
 Patient Mr. Alfredo Alaniz, **Your Patient**
 Started at 10/5/2023, 1:42:44 PM
 0 Attachments

In Review

Reject Approve

Authorizations were NOT provided to release the following: HIV test results, Alcohol and Drug Abuse Records, Details of Mental Health Diagnosis and/or Treatment, Confidential Communications with a Licensed Social Worker, Details of Domestic Violence Victims Counseling, Details of Sexual Assault Counseling, Genetic Screening.

No specific records or documents were requested.

Drag and drop attachments, or click to **Attach**
 Supports CSV, JSON, excel, and most image/document file formats.
 Upload up to 9.5 MB total across 5 different files at a time.

Figure 5. The admin-facing review page for an EHI export request. The related details page displays the same metadata but does not have an attachment upload input field or review-related action buttons.

community, as well as broader health IT communities, encouraging feedback on the tools and expanded interest in the EHI Export IG.

Providing EHI over an API is necessary to reduce patient burden for accessing and using their health data. Current EHI exports require patients to locate request forms and generated files, to manage the transfer of files that may be too large to fit in available storage space on a phone or laptop, and to use capabilities that can differ dramatically across health systems and institutions. This model of standardized API access allows mobile and web applications, like our sample app, to provide a user-friendly conduit between patients, who authorize release of their healthcare data, and a variety of EHI destinations such as providers of secondary clinical opinions, research studies, or AI-powered clinical recommendation tools. With an automated patient-authorized export API, these tools can lower barriers to patients maximizing the value of their EHI across multiple novel contexts.

Moving the EHI IG from its draft status to a formal HL7 publication is an iterative process driven by feedback and incremental improvement. An EHI Export API track at an upcoming FHIR Connectathon event could bring together health IT vendors and app developers to test the draft IG and reference implementations, providing important feedback to refine the IG.²¹ Early pilots of the EHI Export API reference implementation architecture across real world EHR systems—contrasted with our de novo server and web application—will also be critical to maturing standards for the use of this approach with existing systems. Subsequently, an HL7

ballot of the IG as a Standard for Trial Use²² would offer a forum for broader community input, with the eventual goal of formal publication of a specification intended to be binding on implementers.

Because the EHI Export API emphasizes vendor customizability, different healthcare institutions will need slightly different EHI server implementations. Some hospitals with completely digital patient records served by FHIR servers may only need to run a FHIR patient export to generate patient EHI. Conversely, hospitals with fragmented digital infrastructure may use manual processes—for example email-driven attachment uploads or manual drag/drop onto a file server—to collate patient EHI. Additionally, medical records staff may require anything from multi-step admin review with complex checks and verifications to no admin review at all, or something in between. The EHI Export API's flexibility supports all these scenarios, though it follows that real-world testing of our reference implementations across new environments will necessitate some customization.

Future work could include expanding our EHI server to support greater configuration options, demonstrating more real-world workflows in the sample patient application, adding data visualizations for patients, and augmenting research use by integrating EHI into data collection programs.

Conclusion

The Cures Act Rule requirement for EHI export capability in certified EHRs could be pivotal in the liberation of patient

EHI, further advancing patient access rights specified by the Health Insurance Portability and Accountability Act.²³ But without interoperable exchange requirements and an API-mediated EHI export, patient access to their full EHI will be cumbersome; programmatic use of EHI, for example contribution of data to a research project or EHI shared with a diagnostic service, will be hindered by disparate and cumbersome approaches to EHI access.

This new model of patient-controlled, API-mediated EHI export is intended to address those hindrances, building upon the already-mandated support for SMART on FHIR/Bulk FHIR APIs. This model is designed to ease the process for patients to access their health information, promote the exchange of clinical data to enhance patient care and support medical research, and stimulate the development of a thriving market for third-party applications.

We hope our reference implementation can serve as a foundational step toward ushering in a robust ecosystem around EHR data. This would empower patients with personal control over their healthcare data, a long sought after right,²⁴ and enable better decision-making informed by their comprehensive medical records. Achieving wide-scale adoption of these capabilities will necessitate coordinated support from care delivery systems, EHR vendors, and potentially standards organizations and regulators.

Author contributions

K.D.M. obtained funding. K.D.M., J.C.M., and D.G. conceptualized the study. B.M. oversaw development of the implementation guide. V.I. and D.P. developed the custom software that is the subject of study. D.P., D.G., J.C.M., J.J., and K.D.M. wrote the first draft. A.E. managed and coordinated the project. All authors edited and reviewed the draft.

Funding

This work was funded by 90AX0022 from the Office of the National Coordinator of Health Information Technology (ONC) and by U01TR002623 from the National Center for Advancing Translational Sciences/NIH.

Conflicts of interest

J.C.M. is employed by Microsoft Corporation. Boston Children's Hospital receives philanthropic contributions on behalf of Dr Mandl's laboratory from the SMART Advisory Committee with members including Microsoft, Cambia, Humana, and HCA Healthcare. K.D.M. holds equity in SMART Check-In. D.G. is Principal, FHIR and Healthcare Data Standards, for Central Square Solutions.

Data availability

All synthetic patient data were generated using [Synthea](#) and are publicly available at the URLs cited.

Code availability

All software and specifications are publicly available at the URLs cited.

References

- 21st Century Cures Act, Pub L No 144-255, 130 STAT. 1033. 2016. Accessed December 2023. <https://www.congress.gov/bill/114th-congress/house-bill/34/text>
- Information Blocking. Accessed December 2023. <https://www.healthit.gov/topic/information-blocking>
- Health and Human Services Department. 21st Century Cures Act: Interoperability, Information Blocking, and the ONC Health IT Certification Program. 2020. Accessed December 2023. <https://www.federalregister.gov/d/2020-07419>
- Mandel JC, Kreda DA, Mandl KD, et al. SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. *J Am Med Inform Assoc*. 2016;23(5):899-908.
- Mandl KD, Gottlieb D, Mandel JC, et al. Push button population health: the SMART/HL7 FHIR bulk data access application programming interface. *Npj Digital Medicine*. 2020;3(1):151-159.
- Office of the National Coordinator of Health Information Technology. United States Core Data for Interoperability (USCDI). Accessed February 12, 2023. <https://www.healthit.gov/isa/united-states-core-data-interoperability-uscdi>
- 45 CFR 170.315(b)(10)(ii)(B). Accessed January 23, 2024. [https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-D/part-170#p-170.315\(b\)\(10\)\(ii\)\(B\)](https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-D/part-170#p-170.315(b)(10)(ii)(B))
- Epic Systems. Epic's EHI Tables Export Documentation. Accessed August 17, 2023. <https://open.epic.com/EHITables>
- Athenahealth. Athenahealth's athenaOne EHI Export Documentation. Accessed August 17, 2023. <https://docs.athenahealth.com/athenaone-dataexports/>
- Taylor PL, Mandl KD. Leaping the data chasm: structuring donation of clinical data for healthcare innovation and modeling. *Harvard Health Policy Rev*. 2015;14(2):18-21.
- Gordon WJ, Gottlieb D, Kreda D, et al. Patient-led data sharing for clinical bioinformatics research: USCDI and beyond. *J Am Med Inform Assoc*. 2021;28(10):2298-2300.
- SMART Health IT. SMART Health IT. Accessed September 16, 2023. <https://smarthealthit.org/>
- ImplementationGuide—FHIR v6.0.0-cibuild. Accessed December 4, 2023. <https://build.fhir.org/implementationguide.html>
- SMART/HL7 Bulk FHIR Client. *Github*. Accessed August 4, 2023. <https://github.com/smart-on-fhir/bulk-data-client>
- Mandl KD, Mandel JC, Murphy SN, et al. The SMART platform: early experience enabling substitutable applications for electronic health records. *J Am Med Inform Assoc*. 2012;19(4):597-603.
- Health Level Seven. Argonaut Project Home. Accessed June 4, 2023. <https://confluence.hl7.org/display/AP/Argonaut+Project+Home>
- HealthITAnalytics. Argonaut Project Aims to Boost Standards, EHR Interoperability. Accessed April 4, 2023. <https://healthitanalytics.com/news/argonaut-project-aims-boost-standards-ehr-interoperability>
- EHRIntelligence. What Are HL7 Fast Healthcare Interoperability (FHIR) Accelerators? Accessed January 23, 2024. <https://ehrintelligence.com/features/what-are-hl7-fast-healthcare-interoperability-fhir-accelerators>
- Argonaut. EHI Export API Implementation Guide. *Health Level Seven*. 2022. Accessed August 16, 2023. <https://build.fhir.org/ig/argonautproject/ehi-api/>
- Walonoski J, Kramer M, Nichols J, et al. Synthea: an approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *J Am Med Inform Assoc*. 2018;25(3):230-238.

21. McKenzie L. Connectathons—FHIR—confluence. Accessed December 5, 2023. <https://confluence.hl7.org/display/FHIR/Connectathons>
22. McKenzie L. HL7 balloting. Accessed December 5, 2023. <https://confluence.hl7.org/display/HL7/HL7+Balloting>
23. Health Information Privacy Division. Individuals' Right under HIPAA to Access their Health Information 45 CFR § 164.524. Hhs.gov. 2016. Accessed Aug 7, 2023. <https://www.hhs.gov/hipaa/for-professionals/privacy/guidance/access/index.html>
24. Mandl KD, Szolovits P, Kohane IS. Public standards and patients' control: how to keep electronic medical records accessible but private. *BMJ*. 2001;322(7281):283-287.