i2b2 implemented over SMART-on-FHIR

Nicolas Paris, MSc.[1,2,3], Michael Mendis[4], Shawn Murphy, MD, Ph.D[4], Christel Daniel, MD, Ph.D[1,3], Xavier Tannier, Ph.D[3,5] and Pierre Zweigenbaum, Ph.D[2]
[1]WIND-DSI, AP-HP, Paris, France; [2]LIMSI, CNRS, Université Paris-Saclay, Orsay, France; [3]INSERM, UMR_S 1142, LIMICS, Paris, France; [4]Partners Healthcare In., Boston, MA, USA; [5]Sorbonne Universités, UPMC Univ Paris 06, France

Abstract

Integrating Biology and the Bedside (i2b2) is the de-facto open-source medical tool for cohort discovery. Fast Healthcare Interoperability Resources (FHIR) is a new standard for exchanging health care information electronically. Substitutable Modular third-party Applications (SMART) defines the SMART-on-FHIR specification on how applications shall interface with Electronic Health Records (EHR) through FHIR. Related work made it possible to produce FHIR from an i2b2 instance or i2b2 to store FHIR datasets. In this paper, we extend i2b2 to search remotely into one or multiple SMART-on-FHIR Application Programming Interfaces (APIs). This enables the federation of queries, security, terminology mapping, and also bridges the gap between i2b2 and modern big-data technologies.

[PZ]for i2b2?

Introduction

Learning Health Systems aim to maximize the potential of large-scale, harmonized data from variable, quickly-developing digital sources such as Electronic Health Records (EHRs), which are emerging as powerful tools to facilitate discoveries that can improve health. Data heterogeneity is one of the critical problems in analyzing, reusing, sharing or linking datasets. With the development of platforms enabling the linking and federation of phenome, genome and exposome data across sites in US,[1,2] Europe[3,4] or at international scale[5] a key challenge is to define harmonized access to heterogeneous EHR-based data.

i2b2 is the de-facto open-source medical tool for cohort discovery and allows healthcare practitioners to easily subset patient data to address research questions. i2b2 has been described as being used by more than 200 hospitals[6] over the world, and the recent migration of i2b2 to GitHub has facilitated development work. The tool is flexible, supporting its own star schema and ontology model as well as exploiting alternative information models such as PCORnet[7] and the OMOP common data model[8] without requiring changes to the underlying data. Many initiatives have extended the primary goal of cohort discovery, providing functionality to carry out statistical analysis in place, as well as federated queries over multiple centers, and even genomic analytics.[9,10] Three well-known tools that extend the i2b2 functionality in this way include SHRINE, INSITE, tranSMART, and TRINETX[9,11–13].

i2b2 and derived solutions do have room for improvement. For example, in terms of *data variety*, federation tools such as SHRINE, INSITE, and TRINETX are inconsistent in terms of their terminology mapping processes[14]. Where mapping details are provided[2], they are time consuming and software specific[15]. In terms of *freshness of the data*, Extract Transform Load processes (ETL) feeding traditional relational databases supported by the i2b2 tools (e.g. postgreSQL, Oracle, MSSQL) are resource consuming, taking considerable amounts of time, maintenance, and disk-space. Though ETL procedures are still feasible these days, the emergence of high-throughput healthcare data and the "Internet of Things" demands the development of new approaches that allow data to be queried in place (i.e. directly within EHRs) or in optimized, dedicated places such as RDF triple stores or NO-SQL databases. The time delta due to data migrations and transformations poses problems of *data veracity* because the source data is susceptible to be modified in the interval, and multiple transformations are error prone. In terms of *data volume*, data such as omics, exposomics, imaging or free text notes are more and more produced at hospital level while they still are challenging to store and therefore to analyse. In order for the data to be analyzed properly and efficiently, specialized and dedicated technology are required. While there have been several engineering attempts to create i2b2 based data-warehouses solutions that work with technologies other than traditional relational databases[16], the cost to create such interfaces is high. The i2b2 star schema model is highly optimized for fast retrieval of lists of patients matching criteria, but it is not intended for statistical analytics or data exploration[17]. Although there are some bridges with other common data-models such as OMOP, the architecture is still based on RDBMS[7]. The emergence of new technology is faster

[PZ]which architecture?

than i2b2's ability to exploit it. In terms of *software accessibility*, for example, physicians spend time switching between applications, writing their login and password credentials again & again. Providing these users with the "one login/multiple application" paradigm would optimize the time spent on the computer and thus improve patient care.

The solution explored in this work is to bring the latest accomplishments of the Health Level Seven (HL7) Fast Healthcare Interoperability Resources (FHIR) community to i2b2. In particular, to bring the flexibility, the extensibility, the standardization, and the interoperability efforts to i2b2. In the domain of patient care, several large-scale efforts have been underway for over a decade with the goal of specifying both the structure and the semantics of patient clinical information in a manner that enables computable semantic interoperability between diverse systems. Although there is no consensus in the medical informatics community regarding a standard patient information model, FHIR specifications are gaining interest and show promise to mitigate the classic site-specific data mapping problem. FHIR is built on lessons[18] from previous standards, including the Reference Information Model (RIM) which became an ISO standard in 2003, and the Clinical Document Architecture designed to express a single clinical document as a message using HL7 version 3 RIM classes. FHIR specifies a RESTful API to access resources. Several initiatives facilitate the adoption of FHIR, including the Argonaut project[19] and the Clinical Information Modeling Initiative (CIMI)[20]. SMART-on-FHIR[21] is an open, standards based technology platform that enables innovators to create apps that seamlessly and securely run across the healthcare system. Using EHRs or data warehouses that support the SMART standard, patients, doctors, and healthcare practitioners can draw on this library of applications to improve clinical care, research, and public health. SMART improves the user experience with regard to login details in the way OAuthentication does for many websites, and there exist currently more than 50 applications that are able to consume FHIR resources in a consistent way.

[PZ]of whom?

FHIR and SMART-on-FHIR appear to be good candidates for overcoming several of i2b2's architectural weaknesses. Several studies have explored how to bridge i2b2 and FHIR. One approach[22] aims at allowing mobile phones to push FHIR resources into the i2b2 star schema. Other approaches[23,24] allow an existing i2b2 instance to supply their star schema data through a FHIR-API, allowing SMART-on-FHIR applications to run on top of i2b2. In contrast to this prior work our proposal does the exact opposite, in that we build a general interface that allows i2b2 to consume data from any FHIR endpoint. This approach enables clinical datasets to be queried by exploiting FHIR search, Terminology Mapping[25] and SMART Oauth2 security[26] specifications. The aim is not only to bridge the gap between patient care and research communities, but also to open i2b2 to new and improved data types, as well as security and interoperability management in the context of scalable solutions for cross-border and cross-domain networking of data. The ultimate goal of the architecture presented in this paper is to allow multiple institutions to quickly and effectively engage in massive, international cohort discovery studies.

Methods

To meet our objectives, the existing i2b2 traditional query search module (i2b2crc) code source is extended to meet the SMART-on-FHIR API specifications and the FHIR search specifications. Figure 1 shows the overall architecture and how the three-tier i2b2 application integrates with three remote institutions. The figure shows how an i2b2 application gives access to users in a SMART-on-FHIR application modality. In this context, users log-in to any SMART application or EHR system just once, and get access to a set of specialized applications, such as i2b2. The architecture allows queries to be combined over multiple endpoints: zero to one i2b2 star schemas and/or zero to many SMART-on-FHIR APIs.

[PZ]Je n'en vois que deux connectées ?

Figure 2 is a detailed UML sequence diagram. The scenario describes a user who runs a query over an i2b2 instance containing both data in its star schema and data present in multiple remote FHIR-API endpoints. The orange arrows represent the specific new methods introduced in this work. The user first logs into the system with his or her login credentials, which are verified by the i2b2 project management cell (i2b2pm). The i2b2pm lets the user choose from an i2b2 project list according a set of defined roles. After building and running a multiple panel query across different medical domains, the user gets back a patient cohort set by picking concepts from the i2b2 ontology. Some domains are linked to the FHIR endpoints, others are linked to the local star schema. At first, the i2b2crc conducts a patient-set lookup in its local database. Then, for the FHIR-based concepts, the system loops over the following steps against the SMART-on-FHIR API. The i2b2pm first get its Oauth2 authentication credentials because it is known as a trusted application within the SMART-Auth layer. The i2b2crc new FHIR query builder then produces the HTTP query

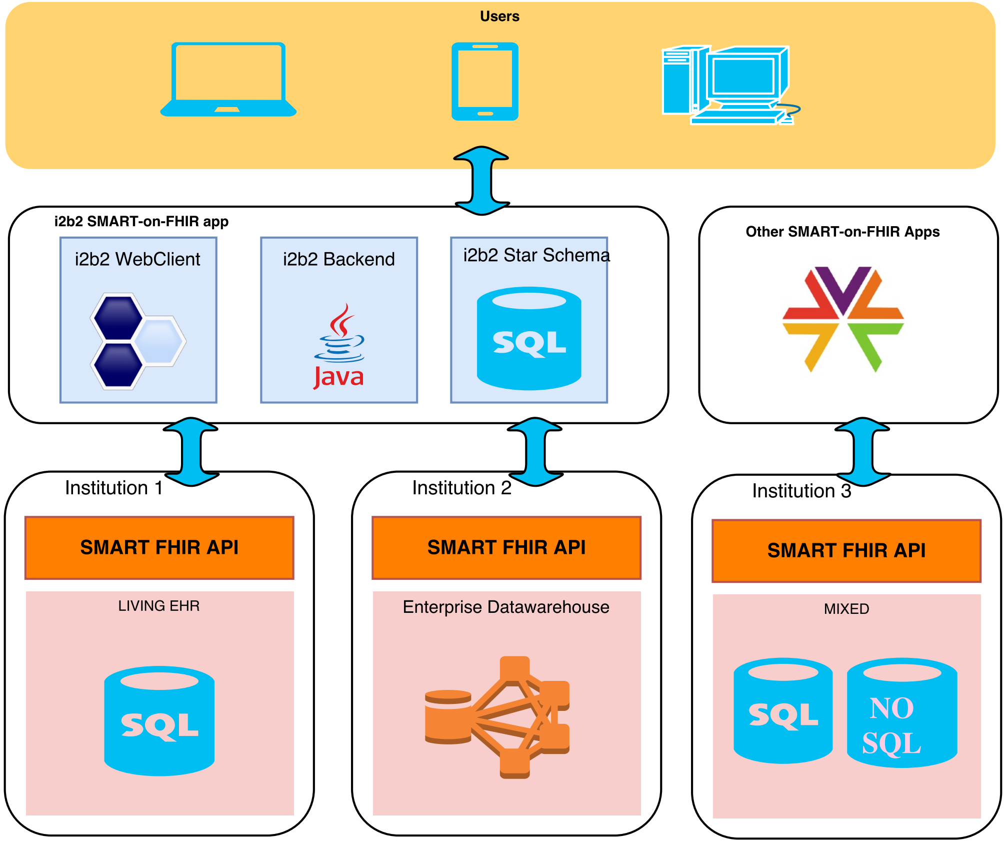


Figure 1: Overall Diagram

according to the FHIR-search specifications and makes an HTTP call to the FHIR-API. This query is extended with coding synonyms defined in the FHIR ConceptMap resources (terminology server part). The resulting query is then translated by the FHIR layer in the local database query dialect to fetch the results. The result is transformed into a FHIR json bundle only containing the information needed (patient_ids in this case). A parsing step extracts the patient_ids. They are then mapped to a unique i2b2 identifier and pushed into a CRC temporary table that integrates all the results. Once looping is done, the i2b2crc applies a new security step to only keep those patients available for the i2b2 project selected by the user. The patient cohort set is finally returned to the user.

[PZ]expanser

HTTP request	Description
GET <FHIR-API>/<Resource> ?_elements=<elements>&code=<codes> &date=gt<date_inf>&date=lt<date_sup> &<custom_filter>	Retrieves chosen <elements> from resources optionally matching a date range or/and a list of <codes> or/and a <custom_filter>
GET <FHIR-API>/ConceptMap ?target-code=<codes> &target-system:in=<code-system>	Retrieves all codes that are mapped to <codes> & <code-system>

Table 1: Index of HTTP request templates

FHIR-search: FHIR search specifications[25] describe how to communicate with a FHIR-API to get back a set of resources matching the criteria in an HTTP query. The present work exploits only the possibility to fetch one type of resource per query. This is sufficient because the traditional i2b2crc allows combining multiple filter predicates by processing each one separately and then uses a deliberation step based on temporary tables. The new i2b2crc query builder replaces the SQL queries acting over the star schema to fetch record identifiers (ID) with HTTP calls to a

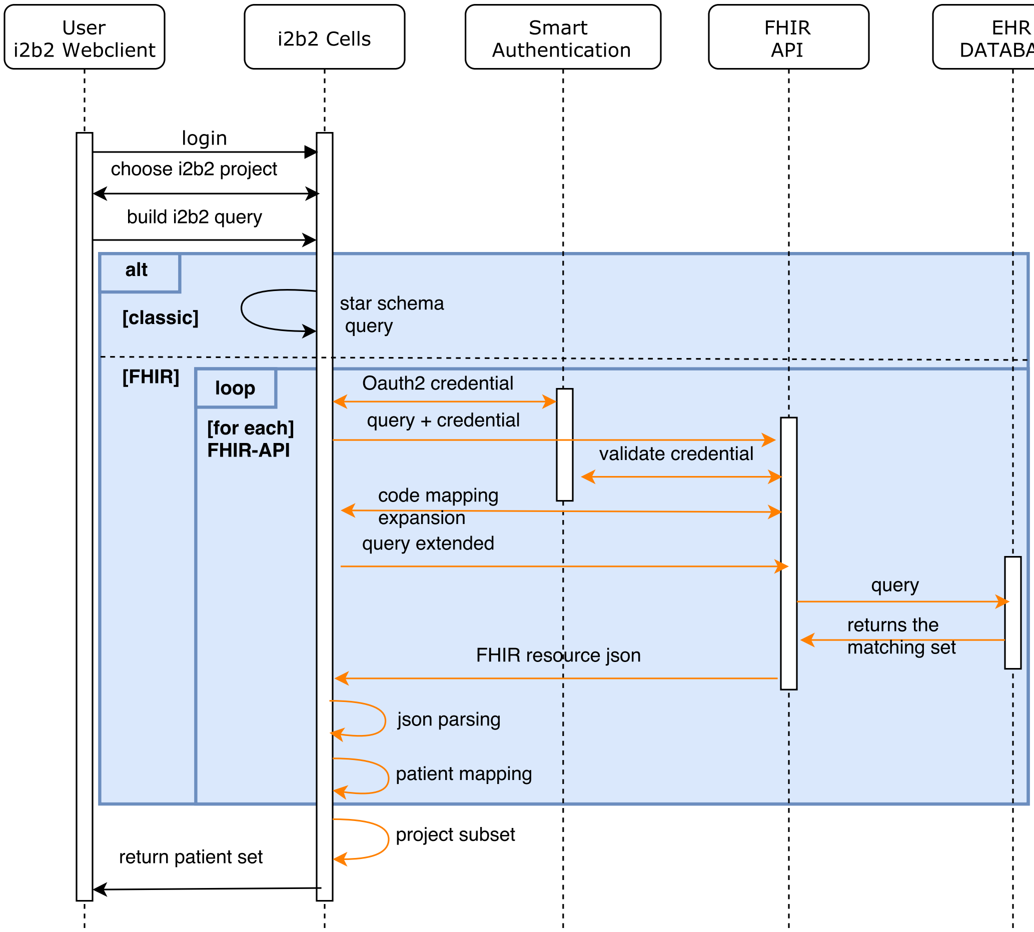


Figure 2: UML Sequence Diagram

FHIR-API. The latter is then translated into the backend database system. The HTTP calls enabled in this design are presented in Table 1. The first row is the general template that is used, and is compared here with the SQL syntax (SELECT, FROM, WHERE):

SELECT: The <elements> pattern lists the resource elements that are returned by the FHIR-API. Depending on the user's choice, patient ID, encounter ID, instance ID or date are retrieved, to respectively provide i2b2 "same patient", "same encounter", "same instance", or "temporal queries" features. The way the i2b2crc retrieves those information from a given resource is described into the i2b2 FHIR config YAML file (see Figure 3).

FROM: The <Resource> pattern is supposed to be replaced by any existing FHIR standard resource, or any profiled resource (modification of the standard to meet the local institutions constraints). In order to let the user point to the right FHIR resource, the i2b2 traditional ontology table has been reused and populated with the needed information. Table 2 describes how to store the information into the "c_tablename" column.

WHERE: Both patterns <date_inf> and <date_sup> allow filtering the data based on the date range defined by the user at run time. The <custom_filter> allows to combine a predefined pattern, such as data status, or a user-defined constraint by value query when enabled by filling the "c_metadataxml" column. The <codes> pattern can optionally specify a list of codes(e.g: SNOMED, LOINC...) by populating the i2b2 ontology "c_basecode" column.

[PZ]"value query" n'est pas clair pour moi

[PZ]"codes" plutôt que "coding"? Ou sinon "coding systems"?

FHIR-mapping: The second row of Table 1 describes the HTTP query template to enable terminology mapping.

ontology table columns	Description	Example
c_basecode	FHIR code_system / code pipe separated	FHIR:http://loinc.com|1234-5
c_tablename	Resource / Profile pipe separated	Observation|ObservationAphp
c_metadataxml	An xml describing datatype (numeric, free text or enumerated) and measure units	cf: i2b2 documentation
c_dimcode	an optionnal additional filter	active=true&status=final

Table 2: i2b2 ontology adapted for FHIR

```
version: dstu3
Patient:
    patientUriPath: $.resource.id
    patientUriField: id
Observation:
  - patientUriPath: $.resource.subject.reference,
  - encounterUriPath: $.resource.context.reference
  - instanceUriPath: $.resource.id
  - datePath: $.resource.effectiveDateTime
  - patientUriField: subject
  - encounterUriField: context
  - instanceUriField: id
  - dateField: effective
[...]
```

Figure 3: i2b2-FHIR YAML configuration file sample

The i2b2crc cell is allowed to fetch synonym codes from the FHIR-Terminology server. As part of the ConceptMap[25] resource, FHIR links a source code to a target with a set of semantic "equivalences" such as "equivalent" or "narrower" that characterize the way they relate to each other. The program fetches each mapping pair and only keeps the "wider", "subsumes", "equal", and "equivalent" semantic equivalence sources. The i2b2-FHIR code expansion exploits this mechanism to query over distinct code systems.

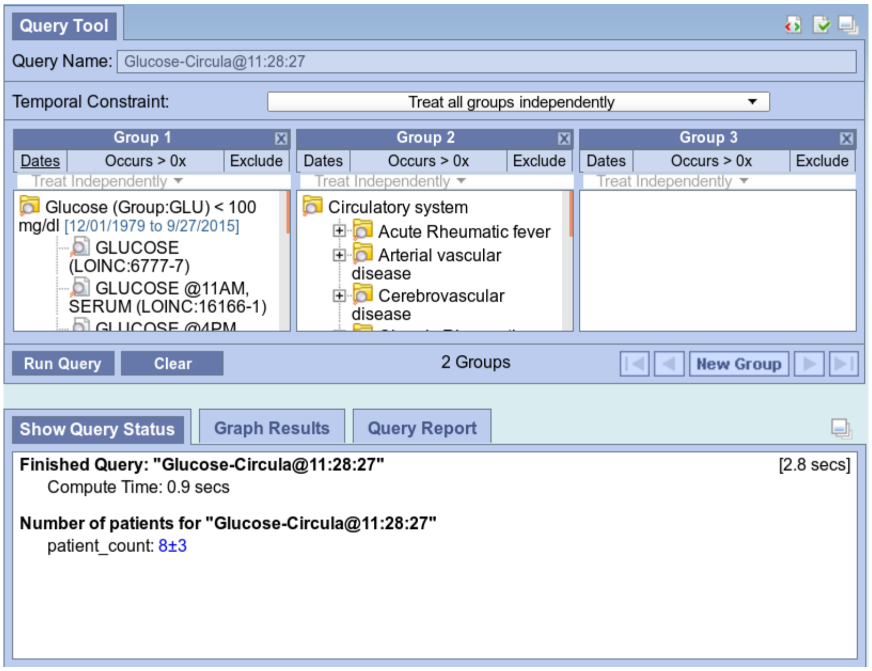
Outcome measurement: In order to test the FHIR DSTU3 resources compatibility coverage, the HAPI FHIR[27] test server has been used as an endpoint since it contains useful demo datasets with fictitious patient data. The benchmark comparing traditional i2b2 and FHIR-i2b2 was carried out with the same i2b2 "observation_fact" table containing 140 million records in a postgreSQL 9.6 instance. The first is based on a 1.7 i2b2 instance. The FHIR-i2b2 has been set up by implementing HAPI-FHIR server on top of the observation_fact table into an Apache Tomcat 9 webserver, and has been accessed via the FHIR-i2b2 prototype. A FHIR-i2b2 big-data benchmark has been set up by implementing an HAPI-FHIR server on top of the "chartevents" MIMIC-III[28] table multiplied by 15, and stored in an Apache HIVE2 table distributed over a 5-computer cluster in the Optimized Row Columnar (ORC) format distributed over HDFS. All software used: i2b2, HAPI-FHIR, postgreSQL and Apache Hive is open-source licensed.

[PZ]The first what?

Results

Implementation Status: The design presented below is implemented at 70%. To date, the new i2b2crc query builder is able to query on both a star schema and one remote FHIR endpoint simultaneously. Logical relations between selection criteria represented as multiple i2b2 webclient panels are also possible. The constitution of a "patient_set" can be constrained by dates, values and measurement units and by one or multiple codes. The code expansion based on FHIR terminology mapping is also implemented. A living demo is deployed[29] and a screen-shot presented in Figure 4. The first panel 1 query searched into HAPI FHIR test server for patients with LOINC glucose codes having value lower than 100ml/dl in a year range from 1979 to 2015 and is mixed with the second panel searching for patients having a diagnosis related to circulatory system within the star schema. The resulting "patient_set" contains about eight patients.

[PZ]first panel 1?



Query Name: Glucose-Circula@11:28:27

Temporal Constraint: Treat all groups independently

Group 1: Glucose (Group:GLU) < 100 mg/dl [12/01/1979 to 9/27/2015]
GLUCOSE (LOINC:6777-7)
GLUCOSE @11AM, SERUM (LOINC:16166-1)
GLUCOSE @4PM

Group 2: Circulatory system
Acute Rheumatic fever
Arterial vascular disease
Cerebrovascular disease

Finished Query: "Glucose-Circula@11:28:27" [2.8 secs]
Compute Time: 0.9 secs

Number of patients for "Glucose-Circula@11:28:27"
patient_count: 8±3

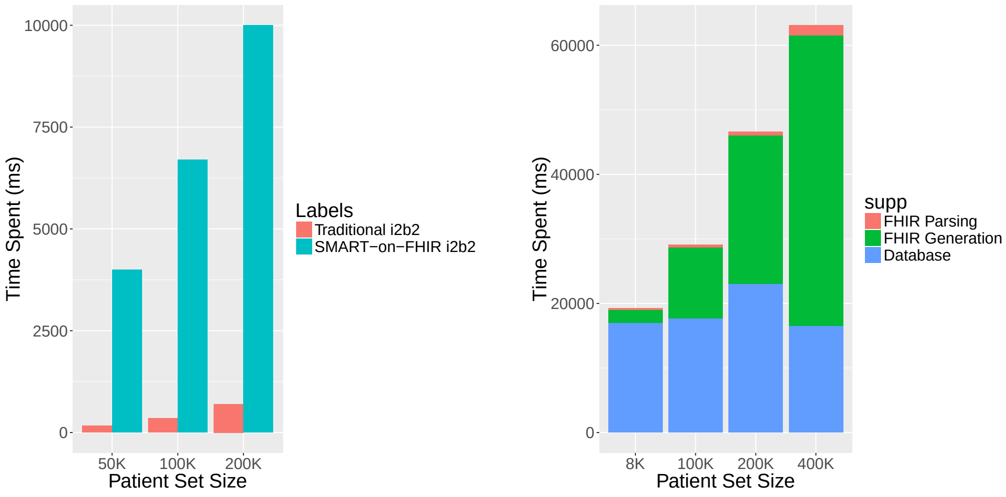
Figure 4: i2b2-FHIR online demo screen-shot

Performance: The performance of i2b2-FHIR has been benchmarked (Figure 5a) versus a traditional i2b2 instance based on the star schema with the same amount of data and configuration (140 million records). The different patient set size corresponds to different criteria selection. The histogram shows traditional i2b2 is 20 times faster than the i2b2-FHIR version. The difference can be explained by the additional steps involved: the fetched result-set is transformed into a json bundle, sent over the network and then parsed. The performance factor tends to decrease with the number of patients matched. The second benchmark (Figure 5b) experimented connecting to an Apache HIVE table on a big-data platform containing 5 billion actual physiological patient data points. The results show that the time spent is under the minute and compatible with i2b2 promises. Moreover, the bar-plots show that the major bottleneck is the FHIR json generation step, mostly the part to produce the json (networking transfert is actually pretty fast). Such quantities of data have never been described to be handled by i2b2 before, since here we approach traditional RDBMS volume limitations. While the traditional i2b2 outperforms the FHIR based i2b2 on modest datasets, the latter opens new perspectives by enabling connections with specialized and optimized database systems.

i2b2 feature coverage: The i2b2 querying feature covers filtering patients facts by code, values, dates, thoughtpatient history, within an encounter temporal window or even a free sequence of events. By adding new temporal table mechanisms, the present work allows all of those features. Thus it does not limit the existing set of functionalities. The i2b2-FHIR configuration file Figure 3 contains information about the FHIR-API instance, such as its version, and how the resources are implemented. Depending on the kind of cohort set the user wants to extract, patient ID, encounter ID, instance ID or dates are retrieved from the FHIR-API thanks to a jsonPATH description. This then allows to populate the CRC temporary tables. This is how i2b2 deliberation mechanisms can be populated, and the set built. Moreover the i2b2-FHIR implementation keeps backwards compatibility and does not impose FHIR-API usage for implementers that would not require it.

[PZ]thought? through?

Security: A security layer has been proposed and implemented into the existing i2b2crc. A new i2b2 table allows to define which patient are part of which i2b2 project. This security layer is important because with one endpoint containing all patients records, it allows to create multiple projects with subset allowing multiple views on the dataset. In terms of performance, the table might be vertically partitioned and split by project, in order to get stable performance as the number of projects increases. This mechanism is both compatible with traditional i2b2 and i2b2-FHIR and has been deployed in production at the AP-HP hospitals where it handles more than 200 distinct projects. The Oauth2



(a)

(b)

Figure 5: (a) Traditional versus i2b2-FHIR performance comparison (on a 150M postgreSQL table). (b) i2b2-FHIR performance (on a 5B Hive table)

security layer has not yet been implemented. The implementation will be inspired by C3-PRO[22] and SMART-on-FHIR[23] that recently succeeded in.

[PZ]succeeded in what?

Extensibility: The FHIR access layer has been tested over the HAPI FHIR test server for all resources at least referring to a patient (68 resources), and has a complete resource coverage. To date, the query builder is compatible with the current FHIR DSTU3 version. In the future, it will maintain compatibility with each FHIR release, and also backward compatibilities. The FHIR version of each endpoint is set up in the configuration file (Figure 3). The query builder handles the FHIR extensibility, local profiled resources or even local new resources. Moreover, the design allows to filter based on FHIR extensions, thanks to the i2b2 ontology table "c_dimcode" with its custom filters. The results suggest that the design is flexible enough to query multiple centers with different FHIR implementations at the same time.

[PZ]to filter what?

Interoperability: The FHIR-ConceptMap expansion has been implemented. A set of test mappings have been produced and populated into HAPI-FHIR as a proof of concept. The HTTP query described in Table 1 (row 2) allows to fetch the equivalent codes. While there is still room for improvement, the results open the way for massive and collaborative concept mapping, with a FHIR compatible terminology server. Interoperability is also derived from the FHIR standard resource definition. However, the ability to derive from them and build Profiled Resources is handled by the i2b2-FHIR YAML configuration flexibility together with the i2b2 ontology table, as they are designed to be adapted.

Discussion

FHIR abstraction allows designing a mixed architecture based on living EHR and big-data storage to leverage massive and unstructured clinical data. One can choose the best technology depending on the expected usage and local specificity of the data. This flexible design allows implementers to define their own i2b2 ontologies. Finally, an i2b2 federation over FHIR is able to bridge multiple FHIR implementations at the same time. The querying benchmarks showed that performance was not really an issue. There is also room for improvements there. High level performance improvement includes parallelizing the queries over both each hospital endpoint and each i2b2 panel. In order to manage heterogeneity in endpoint performance correctly, solutions such background queries have been proposed by SHRINE and could be reproduced. Low level performance improvement includes using more optimized code at both i2b2 and FHIR endpoint side and at each step of the process: json building, json compression, json parsing. While the FHIR resource subset feature is already used to reduce the size of the bundles, we also will give feedback to the FHIR community to propose a more compact option. Finally the recent FHIR specifications around the GraphQL API language[30] lets us envisage both query optimizations and extentions in the future.

By leveraging access to big-data technologies, this opens a new-area of specific solutions (such as temporal-series, text-mining, distributed, graph databases) to manage the diversity, variety and volume of healthcare data such as Genomics[31], Imaging, or physiological waveforms monitoring. The abstraction provided by the FHIR layer allows plugging new text specific technologies based on Apache Lucene, such SOLR & Elastic search. This will allow clinicians to mine text as simply as a modern search engine does. Moreover, the interoperability gain from the FHIR interface let envisage to query multiple centers the same way (FHIR Concept Mapping ability) on real-time data (FHIR API direclty plugged on top of EHR). The security was enforced and allows multiple sub projects to access subsets of the whole patient database. This addresses the patient research opposition and checks that studies only access data needed and permitted.

Several modules have been implemented, some aspects of the design have only been tested as separate modules. The roadmap provides for the development of multiple SMART-on-FHIR endpoints access, Oauth2 implementation, performance improvements and also the release of i2b2 as part of the SMART-on-FHIR apps. Once satisfied with the results, the system should be available in the next releases of the core i2b2. While all resources containing patient reference where tested, there is a need to propose a general mapping between traditional i2b2 objects (patient, visit, provider, observation) and FHIR specific resources (Organization, HealthcareService, Patient, EpisodeOfCare, Condition, Procedure, Medication, MedicationRequest, Observation, DiagnosticReport, ClinicalImpression...). A general algorithm to translate FHIR terminologies into i2b2 ontology will also be investigated, and implemented as complementary software. Last but not least, as a standard way to represent concept mapping, the FHIR ConceptMap resource is a great candidate to centralize and share collaborative work and tools for this major purpose.

Conclusion

By bridging current modern solutions in the field of medical data, this work paves the way for improvements that address current Learning Health Systems challenges. The challenges of *data federation*, *data interoperability*, *data freshness* and *data security* can benefit from both i2b2 experience and FHIR simplification. The challenges of *data volume* and *data variety* of medical datasets are indirectly addressed by the FHIR-API abstraction that makes possible the use of powerful and dedicated technology.

In the end, while a tool that is able to bridge international institutions together is likely to emerge, concept mapping between many institutions remains to be done. Since these institutions are likely to use different languages, different granularities and different concepts and practices, this remains a challenge to be addressed. While ontology matching is an old research area, it is an area that still presents significant challenges to overcome.

[PZ]all = these institutions?

References

1. Omri Gottesman, Helena Kuivaniemi, Gerard Tromp, W. Andrew Faucett, Rongling Li, Teri A. Manolio, Saskia C. Sanderson, Joseph Kannry, Randi Zinberg, Melissa A. Basford, et al. The electronic medical records and genomics (eMERGE) network: past, present, and future. *Genetics in Medicine*, 15(10):761–771, Oct 2013.

2. Andrew J. McMurry, Shawn N. Murphy, Douglas MacFadden, Griffin Weber, William W. Simons, John Orechia, Jonathan Bickel, Nich Wattanasin, Clint Gilbert, Philip Trevvett, et al. SHRINE: Enabling nationally scalable multi-site disease studies. *PLoS ONE*, 8(3):e55811, Mar 2013.

3. Georges De Moor, Mats Sundgren, Dipak Kalra, Andreas Schmidt, Martin Dugas, Brecht Claerhout, Töresin Karakoyun, Christian Ohmann, Pierre-Yves Lastic, Nadir Ammour, et al. Using electronic health records for clinical research: The case of the EHR4CR project. *Journal of Biomedical Informatics*, 53:162–173, Feb 2015.

4. Brendan C. Delaney, Vasa Curcin, Anna Andreasson, Theodoros N. Arvanitis, Hilde Bastiaens, Derek Corrigan, Jean-Francois Ethier, Olga Kostopoulou, Wolfgang Kuchinke, Mark McGilchrist, et al. Translational medicine and patient safety in Europe: TRANSFoRm—architecture for the learning health system in europe. *BioMed Research International*, 2015:1–8, 2015.

5. George Hripcsak, Jon D. Duke, Nigam H. Shah, Christian G. Reich, Vojtech Huser, Martijn J. Schuemie, Marc A. Suchard, Rae Woong Park, Ian Chi Kei Wong, Peter R. Rijnbeek, et al. Observational health data sciences and informatics (OHDSI): opportunities for observational researchers. *Studies in health technology and informatics*, 216:574, 2015.

6. I. S. Kohane, S. E. Churchill, and S. N. Murphy. A translational engine at the national scale: informatics for integrating biology and the bedside. *J Am Med Inform Assoc*, 19(2):181–185, 2012.

7. Jeffrey G Klann, Aaron Abend, Vijay A Raghavan, Kenneth D Mandl, and Shawn N Murphy. Data interchange using i2b2. *Journal of the American Medical Informatics Association*, 23(5):909–915, Sep 2016.

8. i2b2-omop documentation. `https://community.i2b2.org/wiki/display/OMOP/OMOP+Home`. Accessed: 2017-09-28.

9. Elisabeth Scheufele, Dina Aronzon, Robert Coopersmith, Michael T. McDuffie, Manish Kapoor, Christopher A. Uhrich, Jean E. Avitabile, Jinlei Liu, Dan Housman, and Matvey B. Palchuk. tranSMART: an open source knowledge management and high content data analytics platform. *AMIA Summits on Translational Science Proceedings*, 2014:96, 2014.

10. i2b2-transmart website. `https://avillach-lab.hms.harvard.edu/`. Accessed: 2017-09-28.

11. G. M. Weber, S. N. Murphy, A. J. McMurry, D. MacFadden, D. J. Nigrin, S. Churchill, and I. S. Kohane. The shared health research information network (SHRINE): A prototype federated query tool for clinical data repositories. *Journal of the American Medical Informatics Association*, 16(5):624–630, Sep 2009.

12. insite website. `https://www.insiteplatform.com/`. Accessed: 2017-09-28.

13. trinetx website. `https://www.trinetx.com/`. Accessed: 2017-09-28.

14. Yannick Girardeau, Justin Doods, Eric Zapletal, Gilles Chatellier, Christel Daniel, Anita Burgun, Martin Dugas, and Bastien Rance. Leveraging the EHR4CR platform to support patient inclusion in academic studies: challenges and lessons learned. *BMC Medical Research Methodology*, 17(1), Dec 2017.

15. Rob Wynden, Mark G. Weiner, Ida Sim, Davera Gabriel, Marco Casale, Simona Carini, Shannon Hastings, David Ervin, Samson Tu, John H. Gennari, et al. Ontology mapping and data discovery for the translational investigator. *Summit on Translational Bioinformatics*, 2010:66, 2010.

16. Shicai Wang, Ioannis Pandis, Chao Wu, Sijin He, David Johnson, Ibrahim Emam, Florian Guitton, and Yike Guo. High dimensional biological data retrieval optimization with NoSQL technology. *BMC Genomics*, 15(8):S3, Nov 2014.

17. D. Firnkorn, S. Merker, M. Ganzinger, T. Muley, and P. Knaup. Unlocking Data for Statistical Analyses and Data Mining: Generic Case Extraction of Clinical Items from i2b2 and tranSMART. *Stud Health Technol Inform*, 228:567–571, 2016.

18. S. T. Rosenbloom, R. J. Carroll, J. L. Warner, M. E. Matheny, and J. C. Denny. Representing knowledge consistently across health systems. *Yearbook of Medical Informatics*, 26(01):139–147, Aug 2017.

19. Argonaut wikipage. `http://argonautwiki.hl7.org/index.php?title=Main\_Page`. Accessed: 2017-09-28.

20. CIMI website. `http://www.opencimi.org/`. Accessed: 2017-09-28.

21. SMART-on-FHIR website. `http://docs.smarthealthit.org/`. Accessed: 2017-09-28.

22. Pascal B. Pfiffner, Isaac Pinyol, Marc D. Natter, and Kenneth D. Mandl. C3-PRO: Connecting researchkit to the health system using i2b2 and FHIR. *PLOS ONE*, 11(3):e0152722, Mar 2016.

23. Kavishwar B Wagholikar, Joshua C Mandel, Jeffery G Klann, Nich Wattanasin, Michael Mendis, Christopher G Chute, Kenneth D Mandl, and Shawn N Murphy. SMART-on-FHIR implemented over i2b2. *Journal of the American Medical Informatics Association*, page ocw079, Jun 2016.

24. Abdelali Boussadi and Eric Zapletal. A fast healthcare interoperability resources (FHIR) layer implemented over i2b2. *BMC Medical Informatics and Decision Making*, 17(1), Dec 2017.

25. FHIR specifications. `https://www.hl7.org/fhir/`. Accessed: 2017-09-28.

26. SMART-on-FHIR specifications. `http://docs.smarthealthit.org/`. Accessed: 2017-09-28.

27. HAPI-FHIR website. `http://hapifhir.io/`. Accessed: 2017-09-28.

28. AE Johnson, TJ Pollard, L Shen, LW Lehman, M Feng, M Ghassemi, B Moody, P Szolovits, LA Celi, and RG Mark. MIMIC-III, a freely accessible critical care database. *Sci Data*, 3:160035, May 24 2016.

29. i2b2-FHIR demo website. `http://34.205.31.28/webclient/`. Accessed: 2017-09-28.

30. FHIR graphql specifications. `https://build.fhir.org/graphql.html`. Accessed: 2018-01-03.

31. Gil Alterovitz, Jeremy Warner, Peijin Zhang, Yishen Chen, Mollie Ullman-Cullere, David Kreda, and Isaac S. Kohane. SMART on FHIR genomics: Facilitating standardized clinico-genomic apps. *Journal of the American Medical Informatics Association*, page ocv045, Jul 2015.