Modular and scalable RESTful API to sustain STAR collaboration’s record keeping

D Arkhipkin\textsuperscript{1}, J Lauret\textsuperscript{1} and P V Shanmuganathan\textsuperscript{2}

\textsuperscript{1}Physics Department, Brookhaven National Laboratory, Upton, NY 11973-5000 USA
\textsuperscript{2}Kent State University, Kent, Ohio 44242 USA

E-mail: arkhipkin@bnl.gov, jlauret@bnl.gov, sprashan@kent.edu

Abstract. STAR collaboration’s record system is a collection of heterogeneous and sparse information associated to each members and institutions. In its original incarnation, only flat information was stored revealing many restrictions such as the lack of historical change information, the inability to keep track of members leaving and re-joining STAR, or the ability to easily extend the saved information as new requirements appeared. In mid-2013, a new project was launched covering an extensive set of revisited requirements. The requirements led us to a design based on a RESTful API, back-end storage engine relying on key/value pair data representation model coupled with a tiered architecture design. This design was motivated by the fact that unifying many STAR tools, relying on the same business logic and storage engine, was a key and central feature for the maintainability and presentation of records. This central service API would leave no ambiguities and provide easy service integration between STAR tools. The new design stores the changes in records dynamically and allows tracking the changes chronologically. The storage engine is extensible as new field of information emerges (member specific or general) without affecting the presentation or the business logic layers. The new record system features a convenient administrative interface, fuzzy algorithms for data entry and search, and provides basic statistics and graphs. Finally, this modular approach is supplemented with access control, allowing private information and administrative operations to be hidden away from public eyes.

1. Introduction

The Solenoid Tracker At RHIC (STAR) [1] is one of the two experiments at the Relativistic Heavy Ion Collider (RHIC) [2] at the Brookhaven National Laboratory, USA. The STAR collaboration has over 600 scientists and engineers from 56 institutions spread over 12 countries (see figure 1 and figure 2). For the successful handling of STAR collaboration’s records, in addition to basic details of members and institution, historical changes of status such as leave date, re-join date, affiliated institution, authorship, expertise, council membership are also needed. Historical data of the collaboration plays a critical role in STAR’s internal logic such as generating author lists, summarizing talks assignment and assessing equal opportunity as well as calculating shift duties owed for each institution. With increasing needs, extending the stored records with additional details presents a unique challenge. The collaboration data becomes a more dynamically changing set of meta-data than initially anticipated, and keeping the history of all members from the dawn of STAR to today is equivalent to keeping and manipulating a large number of members and institutions with all associated meta-data. In the original record keeping design, only flat information was stored. Administrators were not equipped with a convenient
interface for record updates, and search capabilities were very constrained. Expert knowledge was required for manual database changes such as adding or removing fields. Strongly coupled design of the existing tools (with internal knowledge of the database schema) created situations where propagating changes to the database consistently to all STAR UIs became a huge and time-consuming task. In other words, the approach of using direct database access was neither flexible nor secure. Those tools included the following: the STAR Shift-Log, the STAR Shift-Signup UI, the core collaboration Phone Book as well as a private UI to maintain a fair talk allocation amongst members and institutions.

**Figure 1.** World map of STAR collaborators. The map is generated from our actual records of the collaboration institutions in good standing.

**Figure 2.** Percentage of collaborators from participating countries. The statistics can also be generated by continents.

2. Loosely-coupled Tiered Architecture for new STAR’s records

2.1. Requirements and new design

With the problems identified in our introduction, we derived the following requirements. First, we would like to have a flexible database design that allows us to reshuffle or extend the record structure as the need for the new set of fields or attributes appears. Second, we need a reliable business logic layer for centralized data processing, decoupling clients from the internal database structure and unifying record bookkeeping for all STAR services. In short, the UIs should not be knowledgeable of the database schema, following standard database design practices, but rely on an API returning the requested and required information. We also expect to have a cross-platform interface (or API), which will enable efficient access to the historical record database for all STAR services in the *Online* and *Offline* domains. Well-preserved historical data allows us to recover snapshots of the state of our collaboration records at any desired time. Third, the new design should allow us to consolidate and anticipate for the emerging requirements as requested by the collaboration – this apparently open ended requirement had as guidance to leverage past extension of records, all centralized around adding additional attributes to members or institutions. We also required special attention to the fact that some information may be private or semi-private (hence, implying a service knowledgeable of AA that is, minimally Authentication and Authorization). For example, users may be able to update their own records while their council members control whether they are authors or not; cellular phone numbers may be semi-private and used for shift on-call personnel but never accessible unauthenticated; administrators would have full access. Those three requirements were the basis for a new design.

The proposed approach was to implement a loosely-coupled three-tier architecture composed of the following tiers: a data storage tier based on an Entity–Attribute–Value storage model, a
business logic tier which includes a RESTful API, and a presentation tier. Figure 3 shows the detailed overview of our new design. In the sub-sections below we will describe each tier in more details, starting with the improved data storage architecture.

![Loosely-coupled Tiered Architecture of new STAR's records](image)

**Figure 3.** Loosely-coupled Tiered Architecture of new STAR’s records.

### 2.2 Refactored database storage using EAV data model

Figure 4 shows a simplified representation of the collaboration’s database schema, redesigned using an EAV (Entity–Attribute–Value) data model[3], which fits best our growing collection of sparse meta-data. With about 20 pre-existing fields per record in the old setup, and an immediate possibility of adding half a dozen fields to the existing tables in the near future to support internal (semi-private) information, the EAV model seemed well suited – previous fields would be propagated while (new) fields would have a default or null value. Hence, the EAV data model satisfies additional storage requirements (slower than linear growth upon adding new fields), and upgrading our former flat storage to a highly flexible column-like container for the meta-data is made easy. The refactored database schema is subdivided into two groups, institutions and members, sharing similar internal structure (see figure 4 ). For members and institutions, field properties (attributes in the data table) are described in the conventional-type **Fields** table. Modified EAV-style data tables carry the actual data of members and institutions, while a fourth column stores the event timestamp, helping us preserve the historical records. The first column serves as the foreign key (or ID) of a member or an institution. The second column is the attribute, which is the field table’s primary key. The first and the second columns together make the primary key in the data table. The third column has the actual value of the field of a member or an institution.

Historical records are separated in two additional EAV-style tables, one of each for members and institutions. For easy access and efficiency of row handling, data tables are further divided into three subgroups based on data type of field: string table, integer table, and date table. A separate **Field Group** table is introduced to enable the categorization support of the fields.

### 2.3 Business Logic Tier

This tier uses the meta-data provided by the storage tier, and features generalized procedures related to meta-data processing. It defines the rules for the centralized processing and transformations of the sparse meta-data, its interactions and modules for various workflows,
including the layer which provides compatibility with the legacy clients. In addition, this tier encapsulates and aggregates statistics for the meta-data stored by the EAV model.

**Figure 4.** STAR collaboration database.

**Figure 5.** Structure of RESTful API.

### 2.4. Scalable RESTful API

Representational State Transfer (REST)[4] is an architectural design model which allows to develop scalable and performant services using a set of simple web-friendly interfaces and protocols. The introduction of the RESTful design concepts also allows for portability of service components, which is an important criteria as we expect our software to be used for the next ten years or more (based on the long-term facility planning guidelines). The RESTful API of the new system was developed to decouple various STAR services from the actual data storage, and to ensure unified data access across multiple platforms and technologies commonly used in the STAR software. In our new architecture model, all data produced or consumed by STAR services is encoded in JavaScript Object Notation (JSON)[5]-formatted messages and transmitted over HTTP(S) channels. This makes the information equally available for servers, desktops, web-based applications, or mobile devices’ consumption. Figure 5 describes the resources, methods, and parameters used in STAR’s API. Resources **Members** and **Institutions** describe the members and institutions associated with the STAR collaboration (where **Member Fields** describes fields associated with member tables and **Institution Fields** contains descriptors for the fields associated with institution tables). Our implementation includes four basic methods **CREATE**, **GET**, **UPDATE**, and **LIST**, which are the common methods used to implement the RESTful service model. Next, we briefly outline some STAR services relying on the historical records database.

### 3. Selected STAR services leveraging the REST API

#### 3.1. Phonebook

As part of the upgrade process, a web-based administrative Graphical User Interface (GUI) was developed for the STAR Phonebook using the RESTful API described above. The updated Phonebook interface features a comprehensive set of administrative tools (see Figure 6) allowing to modify the storage fields, making the addition or removal of fields and attributes easy, intuitive, and convenient. A global authentication authorizes administrators to access all data and fields. Data export and statistical display of members and institutions (see figure 1 and figure 2) are possible (spreadsheet exports are popular for the STAR management’s internal need to review records, population distribution in a high demand from agencies, etc.). In addition, the old public Phonebook was re-interfaced with the new API, providing the same basic functionalities to the collaboration as well as extending its features by empowering our...
users with advanced search algorithms such as fuzzy match or a more consistent display of foreign names by supporting scripts and logographs by leveraging Unicode names. Since our new design supports historical records, recovered from past years “flat” table snapshots converted to the new data model, STAR collaborations author list can be generated for any particular time period (e.g. answering the question of “Who were STAR authors in 2005?”) using the historical records database.

3.2. Shift sign-up
Participating in shift work is a great way to learn about the operation and capabilities of the STAR detector and to contribute to STAR’s scientific program. The Shift signup interface (see figure 7) has been designed to automatically provide helpful information, recognize what positions shifters are qualified to sign up for and keep track of institutional dues and accounting. Currently, a STAR shift crew consists of a Period Coordinator, Shift Leader, two Detector Operators and a Run-time Assistant. The Shift Signup application is relying on the upgraded historical record database to determine the list of potential shifters and their qualifications for the available slots. This application provides an estimate of the number of required shift slots for each institution per year based on the number of authors per group and keeps track of the online experts along with other accounting features (such as their contact information and the current on-call expert list, now automated).

3.3. Talks statistics
The most ambitious UI is the STAR Talks Statistics web-based application (see figure 8). This UI provides to all STAR members and the STAR Talks Committee (STC) statistical data on the institutional and personal share of presentations, talks and poster contributions. It is using the historical record database to calculate the raw allocation as well as the author-normalized percentages for talks given at international conferences and regional meetings. For committee members, there is an option to compare statistics associated to selected members and institutions, so conference talks could be assigned to the under-represented groups promoting fair visibility for all members. For the junior members of the collaboration, typically students and postdocs, the Talks Statistics also displays the odds to get a talk assigned to them as a function of STAR institution they belong to in a convenient sorted tabular format (highest to lowest odds). The interface also features statistical information by grouping (gender, regional grouping, seniority, etc.).

4. Conclusions
We presented an upgrade of the STAR record system and its redesigned approach using a tiered architecture. This new design features extensible database design, a scalable RESTful API and a single stack data access technology for all platforms. The refactored database schema is using an EAV data model and has improved our historical record storage. The overall system’s performance satisfies the set of emerging requirements. Our approach allows us to easily integrate historical records to the schema, and allows extending the set of attributes or fields with minimal to no downtime of any services.

Decoupling the presentation layer from business logic and database increases its versatility, such as making it accessible from any device or platform. The old UI was in fact ported to the new service providing extended functionality with the same reliability to which the collaboration was accustomed.

As part of the reshape a web based GUI was provided for an easy use by the administrators, along with the public interface for the collaborators. Now, administrators have an intuitive and full access and control of record fields. They can add, remove, set attributes and provide access control to the collaborators (the AA features are currently partially implemented and will be the
focus of our future work). Furthermore, an extended set of add-on features is provided via the public interface, such as statistical display of collaboration records, advanced search capabilities (standard, fuzzy, conditional, etc.) providing unprecedented capabilities to the collaboration and great consistency across many previously disconnected UIs. Finally, the transition is now considered complete – it is worthwhile to note that as each UI was phased in and interfaced with the new service, we suffered no downtime.

Figure 6. Administrator view of web based phonebook GUI.

Figure 7. STAR shift-sign up page and part of shift statistics of year 2015.
Figure 8. All the talks given by STAR institutions in year 2014. The red bar shows the talk allocation renormalized by the number of authors.

Acknowledgments
This work was supported by the Office of Nuclear Physics within the U.S. Department of Energy’s Office of Science.

References
[1] The Solenoid Tracker At RHIC (STAR) experiment [Online] URL http://www.star.bnl.gov/
[2] M Harrison, T Ludlam and S Ozaki 2003 Nucl.Instrum.Methods Phys.Res.A 499 235–244
[3] Entity-attribute-value model [Online] URL http://en.wikipedia.org/wiki/Entity-attribute-value_model
[4] RESTful web services [Online] URL http://www.ibm.com/developerworks/webservices/library/ws-restful/
[5] The JSON Data Interchange Format URL http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf