PHEMI Central Features

Draft: September 13, 2014  
Updated: October 15, 2014  
Update: November 24, 2014

Roy Wilds

**FOR INTERNAL USE ONLY**

# Release Schedule[[1]](#footnote-1)

The table below specifies the projected release schedule for PHEMI Agile and Central releases and the features contained. There are 2 major releases of each product per year (1.0, 2.0, etc.). The .5 releases are internal milestones that are not generally released.

PHEMI Central includes all Agile features, plus the specific ones listed. The External Component features are add-ons that are separately licensed. Initial delivery of the features listed will typically be limited in functionality, with improvements being continually developed moving forward.

Table : Release roadmap of features for PHEMI Agile and PHEMI Central. The .5 releases are internal milestones, not generally available releases. Patches, updates, etc. are not reflected on this roadmap. Release is intended to happen at the mid-point of the Quarter indicated.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Quarter** | **Agile** | | **Central** | | **External Components** |
| 14/Q4 |  |  |  |  | - Self-serve Forms Server |
| 15/Q1 | 0.5 | - Data Immutability  - Retention Policies  - Version Control  - Data Deduplication | 1.0 | - Cell-level Security  - Multi-projects | - HANA Data Integration |
| 15/Q2 | 1.0 | - Data Linking  - Backup & Recovery  - Native SQL Interface  - Data Dictionary | 1.5 | - Role Separation  - Data Anonymization | - Genome/VCF Analyser |
| 15/Q3 | 1.5 | - Audit Logs  - Version Rollback | 2.0 | - Master Patient Index  - Multi-tenancy | - Geospatial |
| 15/Q4 | 2.0 | - Encrypted Data  - *TBD* | 2.5 | - Consent Management  - ABAC | *TBD* |

# Shipping PHEMI Central/Agile

PHEMI Agile/Central software is shipped as either an Appliance, deployed on Amazon Web Services (AWS), or installed on a customers VMWare environment.

For all scenarios, PHEMI Professional Services is responsible for setting up the base cluster system, consisting of cluster node(s) and management node(s), so that PHEMI Agile/Central software can be installed & configured on the cluster.

# Features

This section contains a brief description of existing and future features for PHEMI Central/Agile, including the quarter release is planned for. Refer to the Release Schedule for information on which features belong to only Central. The feature descriptions span the initial version of the feature through to more advanced capability that will not be released until future development is completed.

## Version Control (Q1/2015)

Data Owners can specify that they want the ability to update/replace objects stored for a Data Source. This will provide an additional interface where they can specify existing object(s) in the system (by specifying metadata field(s)) and provide a new object to replace those existing objects. This feature enables a user to correct data that may have been mistakenly entered, or to update data with a newer version.

Replacement of an object has the old object still present in the system, but not viewed/accessed through normal system access (i.e. a Dataset Execution will not “see” the old data, nor will a search “find” it). The new object will have the Admin and Descriptive metadata the same as the old.

Data Owners and Auditors can query the system to retrieve the version history (i.e. all previous versions) for any object. This history includes the objects and metadata.

Replacement gets tricky when we have trails/links of the data that are also present in other tables/structures, such as the Index and Aggregate tables. Initially, we will only update/add the Aggregate, Index, and any other data structures (if applicable) with the new values (from the replacing object), but will not remove the replaced data from the Aggregate table. This means that the Aggregate table will become inaccurate (since it has the original and replaced data in it). If a user wishes to make the Aggregate data structures only reflect the “current data” (i.e. ignore versions that were replaced) then they must re-compute them (i.e. reprocess all data). Future iterations of this feature will address these gaps.

More advanced versions of this feature will allow the tagging of versions of object(s) in the system. The rollback feature (separate) describes the ability to specify which version of an object (or objects) should be treated as the “current” objects in the system.

## Rollback (Q3/2015)

Data Owners can specify a global “tag” that refers to a specific version of data in the system to be the current one.

Once Version Control supports tagging of versions of data then the rollback feature will permit global (i.e. across many objects) specification of which version of data should be accessed when performing computations on data and/or reading data from the system.

## Role Separation (Q2/2015)

This is a change to the interface to the system (via the GUI and underlying REST interfaces) involving the flow/processes that take place. Certain personas in the system will be required/restricted in what they can do for adds, moves, changes of objects in the system, such as Data Sources, Datasets, Access Policies, etc.

For example, when creating a Dataset, only a Privacy Officer persona will be permitted to set the status of the dataset to be Available. Initial creation of it will be restricted to Draft status. Once it has been made Available, then other personas can execute the Dataset.

## Cell-level Security (Q1/2015)

Programmers that are writing DPFs can apply attribute(s) to derived data using the DPF framework that PHEMI has. These attributes are referenced in Access Policies to determine what derived data can be accessed by users/tasks that read/access data in the system.

Initially this is used only for Derived Data, and is specified by Data Processing Functions (DPFs). Future versions of the feature will have visibilities for Raw Data as well. Additionally there will be Administrator controls for managing the set of visibilities that are available in the system (i.e. manage the set of visibilities, such as: IDENTIFIED, DE-IDENTIFIED, NON-IDENTIFIED).

## Attribute Based Access Controls (Q1/2015)

Privacy Officers and Administrators can create Access Policies that embody ABAC (Attribute Based Access Control) principles (such as shown in the initial implementation of the Access Policies). These policies are attached to datasets and determine how the data is presented to the user that is executing the dataset creation (i.e. materializing the dataset).

A future version of this feature will apply the same rules to the REST interface to data in the system. They will be configured on a per Data Source basis (i.e. using Attributes for objects/derived data stored for a Data Source).

Additional improvements will alter the policy creation and implementation to more closely align itself with the XACML, NIST, or other appropriate standards implementation of ABAC.

## Native SQL – Schema on Demand (Q2/2015)

DBAs can create Datasets that are tabular views of data that is stored in our key-value store, with rows corresponding to objects in the system, and columns being fields/columns for each row.

This tabular view can be queried in a variety of ways, such as exporting the contents of the view to a CSV, or placing it in an external database system (such as HANA, MySQL, Hive, etc.).

Future iterations of this will enable the dataset view(s) that are created to be directly interrogated using SQL without materializing the data in an intermediate system (i.e. in Hive or HANA). Additionally, we will include a JDBC or ODBC driver that can be connected to by external 3rd party tools to query data in the PHEMI system.

## HANA Data Integration (Q1/2015)

This feature is focused on enhancing the HANA integration with PHEMI.

Initially, DBAs can create Datasets that are virtual tables (i.e. like a VIEW in a RDBMS). These are effectively a schema (tables, columns, types) that are mapped/connected to the underlying system. Users can configure Datasets so that when the dataset is executed (i.e. actually populated/constructed) it will be materialized in HANA. Initially we treat HANA as just a RDBMS that we push data to.

The next iteration of this feature is to permit HANA to use SDA (Smart Data Access) to request data from PHEMI datasets that have been registered with a HANA system. Additional future features would have the federated queries include persona/identity information about the user issuing the query so that the appropriate Access Policy can be applied to the data.

## Data Anonymization (Q2/2015)

Programmers and Data Scientists can leverage a library of functions (either language agnostic, or in Python, then Java to start) that will be created for anonymizing data, and made available for DPF development. For instance, the masking of fields or truncation of data (e.g. truncate date to the year).

More advanced functions could use a secret key to encrypt identifiers, or create tokens that uniquely identify data without revealing information about the original value.

## Encryption of Data at Rest (Q4/2015)

Data Owners can specify that the data managed under a Data Source be Encrypted at rest. The Data Owner will use a secret key (e.g. password) to encrypt data that is stored. The Data Owner will specify which users (or what attributes they must have) to enable decryption of the stored data.

## Data Verification (TBD)

A DBA can configure when a system process will periodically compare the metadata checksum with the actual checksum of the raw data to detect if any corruption has occurred. Logs are generated to note any problems. This feature is analogous to the Data Retention enforcement feature.

A more advanced version of this allows checksum verification of derived data and data structures as well.

## Data Validation (TBD)

The ingestion process is augmented to allow users to pass in a checksum for the data that is being ingested. This is used to validate that the data stored in the raw data field is correct.

This is only done at ingestion time. Data Verification takes care of verifying the integrity of data after it’s been stored.

## Retention Policy Enforcement (Q1/2015)

A Data Owner can configure the retention policy associated to a Data Source (e.g. store for so long, or delete after). An Administrator configures when a system process is run periodically to enforce the retention policy by deprecating and/or deleting data from the system.

## Multi-projects (Q1/2015)

An Administrator can configure the Data Attributes, Access Policies, and Authorizations to prevent users/data from multiple projects accessing each others data.

## Multi-tenancy (Q3/2015)

A super-administrator can configure the domains (i.e. segregation of data, users, policies, logs, etc. in the system) so that all resources/objects associated with, for example, Company A are opaque to Company B. This is analogous to having different databases schemas in an RDBMS, with access to each database being completely segregated from others.

## Geospatial Querying (Q2/2015)

A Programmer can use the PHEMI DPF framework to tag data as being geographical (such as latitude, longitude). The PHEMI system will provide a REST interface allowing users to query the system by specifying geographical regions (i.e. a bounding box or other polygonal area) and the system will efficiently return the relevant data.

## Data Dictionary/Curation (Q2/2015)

Data Curation refers to the ability to catalog (i.e. via a Data Dictionary) and cleanse (i.e. standardize a date format) data.

Data Scientists can cleanse, curate, and transform data in the PHEMI system to increase the value of this data.

Initially, the PHEMI system provides users with a framework to write DPFs to perform this functionality, and can leverage a standard library that contains common functionality (such as a date cleanser/normalizer). The DPF emits fields that can be queried against (i.e. this is the cataloging) and which have values that have been computed in a specific way (i.e. this is the cleanse/map).

The role of a Data Dictionary that enables users to interactively work with the fields (i.e. derived data) that a DPF produces will be developed next.

Future versions of this feature will enable Data Scientists to interactively correct/fix data and specify rules to be applied (such as “if object is PDF and has attribute Y then apply operation A to extract value X”). Google’s OpenRefine provides much of this functionality and this feature will either incorporate the functionality or will integrate that actual code with the system.

## Backup and Recovery (Q1/2015)

Administrators can specify a backup resource (i.e. hardware/nodes) and schedule for the system to be backed up onto. This backup can be used to restore a PHEMI system to the state captured by the backup.

Advanced versions of this will enable remote backup (i.e. separate data center for disaster protection).

## Data Linking (Q2/2015)

A Programmer/Data Scientist can specify relationships between objects stored in the system, by indicating a set of objects (i.e. logical rows) and a descriptive label, and have this relationship stored in the system so that it can be efficiently queried by specifying the label, or any member of the set.

This feature is a way of describing relationships amongst objects in the system. For example, the user could identify all Echo98 documents stored in the system for a particular patient, and apply a label to this set that is the patient’s Name. The system could then be queried with a single Echo98 object, and based on the links that were defined; all objects in the set with that Echo98 document would be returned, along with the label (i.e. Name).

## Master Identity Index (Q2/2015)

This feature is a particular implementation of the Data Linking feature that provides helpful capabilities focused around managing a collection of objects that are related to a set of individuals.

A user can upload a list that is a set of identities (i.e. could be a list of PHNs). Objects that are then stored in the system can be tagged with elements of the set of identities. The system can be efficiently queried to retrieve objects for an identity. Furthermore, the identities can be used to protect data stored in the system via the use of visibilities.

## Consent Management (Q3/2015)

This feature combines with the Master Identity Index feature to enable object level control of how that object is to be:

* Stored (i.e. object level retention policy)
* Read/Accessed (i.e. visibilities)
* The object is also linked to the consent specification (could be rules and/or a document).

The ingestion of data has an additional parameter that allows this consent data to be attached, facilitating this behavior.

## Data Compression (TBD)

An Administrator can turn on Data Compression for a Data Source that will reduce the size of the raw data being stored by using compression techniques to reduce the overall footprint of the data. When data is read/accessed it will be de-compressed.

## D4M (TBD)

Performance of ad hoc queries across large genotype/phenotype datasets is a frustrating problem for researchers looking to explore their digital assets, searching for biomarkers.  PHEMI plans to evaluate, and potentially license and integrate the Dynamic Distributed Dimensional Data Model (D4M) capabilities developed by MIT Lincoln Labs.

## Bio-Informatics Processing (TBD)

Moving large genomic files to cluster computers for sequence pipeline operations can be a slow and inefficient process.  PHEMI is closely following bioinformatics research currently investigating the execution of sequencing pipeline algorithms within a big data system.

Separately, PHEMI will work with lead customers to explore ways to improve system performance by removing bioinformatics system bottlenecks. e.g. preprocessing aggregates.

## Graph Database (TBD)

Researchers struggle with organizing large quantities of heavily interconnected data.  Traditional database systems are insufficient to capture the relationship and rapidly evolving understanding of genes, proteins, regulatory mechanisms, etc. Graph database technology will become a powerful tool to explore and discover relationships within very large and complex genotype/phenotype datasets.

## Genomic Data Visualisation (TBD)

The area of genotype/phenotype analysis is a relatively new field.  PHEMI plans to work with Professor Tamara Munzer, an internationally recognized genomics visualization expert from the University of British Columbia Department of Computer Science.  Work under consideration includes the pre-processing of data in PHEMI Central to accelerate data visualization.

## Genomics EMR Integration (TBD)

Currently, genomic information is not captured in hospital EMRs, and most physicians are poorly equipped to deal with this information.  PHEMI plans to develop a series of features to feed the output from a genomic analysis into clinical EMRs to make the information accessible and actionable by clinicians.  Partnership with a third party Clinical Decision Support tool is anticipated.

# Personas

An individual may have 1 or more personae (and a persona may have more than one individual) depending on the size/complexity/stage of the organization.

Responsibilities for each persona are as follows.

## Data Owner

* Managing the on-boarding process for new data sources
* Implementing data sharing agreements,
* Implementing privacy policies (approved by the privacy officer) for all digital assets

## Data Scientist/Programmer

* Writing and maintaining data processing functions to curate digital assets.

## DBA/Data Architect

* Custodian of Digital Assets
* Approving new data processing functions.
* Approving data sets for Analysts
* Creating joined, verified, and finalized datasets to be consumed by Data Analysts (done in external integrated systems, such as HANA).
* Publishing final datasets for the Data Analyst (approved by the privacy officer)

## Data Analysts/Data Consumer

* The end user and consumer of information.
* Using the datasets produced by PHEMI Central.

## Administrator

* User accounts
* Managing Dataset creation and access settings
* Managing Attributes and Authorizations for protecting data in the system
* …

## IT/System Administrator (sysadmin)

* Integration with Identity, source information systems, user applications/services, analytic tools, etc.
* Monitoring and maintenance of system processes and hardware components
* Security (e.g. encryption)
* SLA (Hardware/software system availability, performance and capacity management)
* Backup, disaster recovery
* System monitoring and alarms
* System management and maintenance (e.g. ability to identify a node as not available, repair a disk, replace a node, add more nodes, remove nodes, etc.)

## Privacy Officer:

* Writing and enforcing privacy processes in the organization
* Writing and approving Access Policy rules
* Approving user attribute/settings
* Reviewing and approving dataset requests (i.e. change Draft to Available)
* Conducting privacy audits.

## Data Governance Committee

* Defining, enforcing and overseeing roles, policies and procedures at a system and a company level.
* Reviewing governance/audit reports

# Stories

This section describes some simple usage stories for PHEMI Central/Agile.

## Report Storage & Routing

*Under Development!*

The Widget company specializes in routing lab reports from laboratories that generate reports, based on patient samples, to physicians and other healthcare providers. They wish to use a platform that will enable them to scale cost-effectively as volumes grow, as well as deliver additional valuable services based on the information they store & route.

The reports are XML documents.

## Clinical & Genomic Storage Platform

This is a collection of stories outlining the types of functions the user wishes to use the solution for.

**Story 1**

I'd like store a collection of historical patient records as a new data source in PHEMI Central. The data is a single sheet in an Excel workbook, and there is a single column that contains the unique identifier for the patients. Moving forward, I'd like to create a form to facilitate the collection of the same data, but through a web-portal, and have it stored under the same data source (see Story 8).

I want to enable users to query for records in the system, based on the contents of the fields (e.g. find records with this unique identifier, or find all records where the age in the record is over 65). The queried records will be downloaded as an Excel workbook object.

The Administrator would like to setup controls to enable some users to see & query against sensitive data, while other users can not query against these fields, and will only see de-identified versions of the data. De-identified means that the value is masked.

For audit purposes, I want the Data Owner to be able to see a log of all queries against the system, as well as a record of all user creations/modifications so I can catch if someone gains access when they shouldn't have, and their query activity.

**Story 2**

I'd like to be able to store additional data sources in the system, and have them connected to a patient via the same unique identifier as in the patient record in story 1. When a user queries for data based on a unique identifier, they'll be able to select what data sources they want to query across, and the system will return a collection (as a .zip file) of all the objects stored in the system (assuming they have the necessary authorizations to do this query).

**Story 3**

I'd like to remove all data for a specific patient record in the system if they revoke their consent for me to store their data. This removal is to be recorded in the Audit Log and I need to be able to prove to a compliance officer that the data has been removed if needed.

**Story 4**

In addition to data sources from Excel Workbooks and Forms, I'd like to store arbitrary data for patients in the system. I want to specify some descriptive parameters, such as the: (i) Unique Identifier of the patient, (ii) Date the data was collected; that I can later on query against to retrieve the data.

**Story 5**

I'd like to develop a DPF to extract additional fields from the arbitrary data being stored in the system, and be able to query against this additional data.

**Story 6**

I'd like to update any data stored in the system in case I mis-enter something the first time. I want to specify query criteria to identify the object to be replaced, and then provide a new object to replace the old data. I want this activity recorded in the audit log.

**Story 7**

I want a DPF that processes and extracts data from Variant Call Format (VCF) files so I can query against variants that are encoded in the files I upload.

**Story 8**

I'd like to build a web-form that can collect data that is structured the same as the Excel Workbook data in Story 1, so I can transition from my historical method of collecting data via spreadsheets, to an online method of data collection. I want to query and interact with the system the same way, regardless of if the data came from a spreadsheet or a web-form. (note that it's up to the creator of the web-form and spreadsheet to ensure that a query makes sense across both sources of data).

Furthermore, I want to build the web-form without using any programming (i.e. WYSIWYG style), and I want the ability to edit and modify it in the future.

## Ministry Data Warehouse

*Under Development!*

The ministry of widgets wants to combine three different database systems into a single platform where they can standardize various representations of data, combine this data, and perform standard BI (Business Intelligence) analysis on the data.

System 1 contains information about applications the ministry receives from citizens/companies for developing land. Each application contains geographical information, and a variety of specifications about the use of the area.

# As an End User, How do I

(and who does it)

## Manage users, ensuring they get the correct authorizations and roles

An Administrator will use the Identity Management system to record the user. Each authorization granted will be recorded on which day it is effective, and if applicable, when the authorization expires.

## Determine what Attributes and Authorizations are needed for the data being stored

For all Data being considered for storage, identify the different sensitivities or properties that may factor into access control restrictions.

## Determine the Access Policies needed to ensure I conform with any (and all) regulations/compliance requirements

Have the Chief Privacy Officer (or equivalent) person involved.

## Use the Audit Log to demonstrate compliance

## Create and Manage a new Data Source

## Develop and deploy a Data Processing Function

## Create and Manage a new Dataset

## Remove (or replace) data that has been accidentally put in the system

## Configure the Dataset Destinations for the system

1. Subject to change based on customer requirements and priorities. [↑](#footnote-ref-1)