Glossary

Access policies

An access policy is a set of rules that determines how users can interact with data within the PHEMI system. Asingle access policy rule lists some number of *attributes* that might be applied to a data item (for example, "SECRET"), the allowed action for that attribute (for example "READ"), and the *authorization(s)* a user must have to perform the specified action (for example, "SECRET-CLEARED"). It is also possible to include environmental properties in a rule; for example, to allow an action only when the user is logged on from a mobile device.

Attributes

All raw data and derived data stored in the PHEMI system can be tagged with data attributes. Data attributes are matched with the system user *authorizations* in *access policies* that specify what actions can be taken on what data, by whom.

Authorizations

Authorizations are properties that can be assigned to system users. Authorizations are matched in *access policies* against *data attributes* to allow fine-grained control over how users may interact with data in the system.

Cell

A cell is the unit of security for data within the PHEMI system. For *raw data*, the cell is the binary representation of the submitted document (for example, image or text) considered together with the data's *metadata*. For *derived data*, a cell is each derived data item, considered together with its metadata.

Code library

A code library is a package of executable code that is included in a *DPF archive*. Whether the code is source or compiled depends on the coding language. Code libraries must be portable and self-contained; that is, all dependencies required for the *DPF* to function must be bundled inside the library, in the appropriate way, for whatever language is being used.

Dataset

A dataset is a set of *derived data items* prepared for export from the PHEMI system to the consuming system. The available derived data items is determined by the *DPF* for the *data source*. Data items in a dataset can be selected from across multiple data sources and DPFs.

Data source

A data source represents an external source of data to be ingested into PHEMI Agile. Data Sources define data ownership, privacy, retention rules and other properties of the source data. These definitions are used to create metadata for every piece of data that is ingested into the system, enabling the tracking and management of all data in accordance with the properties.

Derived data

Derived data is data that is extracted from the *digital assets* that result when *raw data* is *ingested* into the system and tagged with *metadata*. Derived data is produced by a *Data Processing Function* acting on a set of digital assets. The set of derived data items can be searched, further processed, or exported from the system.

Digital asset

A digital asset is any piece of data stored in the system that has *metadata* properties defining how the data is to be managed. An item of *raw data* input into the system (for example, an X-ray image or XML document) is a digital asset when considered together with its metadata, as are derived data items.

DPF

DPF stands for Data Processing Function. A DPF is an executable piece of code that supplies the instructions for parsing a *digital asset* (for example, a log message or medical report) into *derived data* (such as a temperature reading or blood glucose measurement). The output of a DPF is structured elements, which can include type properties (for example, INT or STRING) and can include attributes (for example, SECRET or IDENTIFIABLE). A DPF is "registered" with a specific data source, by uploading the *DPF archive* during data source configuration.

DPF archive

A DPF archive is the code comprising a Data Processing Function. A DPF archive takes the form of a ZIP file archive, where the ZIP file contains a *manifest file* and a *code library*. To register a DPF with a data source, its DPF archive is uploaded during data source configuration.

Ingestion

Ingestion is the process by which data is brought into the PHEMI system. During ingestion, the system applies *metadata* to the incoming data, transforming the *raw data* into *digital assets*. To have raw data ingested, you configure the *data source* to specify the data policy, the ingest schedule, and the *Data Processing Function* to be used.

JSON

JSON stands for JavaScript Object Notation. JSON is a lightweight data-interchange format that is easy for humans to read and write and easy for machines to parse and generate. JSON is used in the body of several REST requests in the PHEMI *RESTful API*.

Key-value pairs

A key-value pair is a set of two linked data items: a key which uniquely identifies some item of data, and the data value itself.

Logical row

When a collection of *key-value pairs* from a single document are grouped together, the result is a logical row.

M2M

M2M is a way of referring to machine-to-machine interfaces, used in machine-to-machine communication.

Manifest file

A manifest file is a *JSON* file specifies the output of a *DPF*. With the *code library*, the manifest file forms the *DPF archive* that is uploaded to register the DPF with a *data source*. The manifest file should include the properties of the DPF along with the details of each *derived data item* to be generated.

Metadata

Metadata is information about a piece of data. In the PHEMI system, metadata is information about how a given piece of data is to be managed. When a piece of *raw data* is *ingested* into the PHEMI system, information from the connection together with information configured for the *data source* is used to create a variety of metadata properties that are stored with the raw data. For example, the system uses the timestamp from when the object was ingested together with the retention policy in the data source configuration to generate a time-to-live metadata property for the item. When a piece of raw or derived data is associated with metadata, it is considered a *digital asset*.

MongoDB

(From "hu**mongo**us"). MondoDB is an open-source document database.

Personally Identifiable Information (PII)

The concept of Personally Identifiable Information, or PII, is a legal concept used in US privacy law and information security to mean information that can be used on its own or with other information to identify, contact, or locate a single person, or to identify an individual in context. When thinking about PII, it is important to distinguish legal requirements to remove attributes uniquely identify an individual from a general technical ability to identify individuals. Because of the versatility and power of modern re-identification algorithms, together with the amount of information freely available from all sources, the absence of PII data does not guarantee that de-identified data cannot be used, perhaps in combination with other data, to identify individuals.

Privacy-level attributes

Privacy-level attributes are attributes that characterize the privacy level of a data item. The PHEMI system includes predefined privacy-level attributes designed to apply to data domains where privacy is important:

- **IDENTIFIED**. The data contains *Personally Identifying Information* that potentially identifies an individual. Examples of information of this type include name, Social Insurance Number, and date of birth.
- **DE-IDENTIFIED**. The data contains IDENTIFIED information that has been masked or encrypted.
- **NON-IDENTIFIED**. The data is not identifying in and of itself. Examples of this type of information include weight or favorite food.

Raw data

In the PHEMI system, raw data is files, objects, records, images, and so on that are submitted for ingest into the system. Raw data is stored exactly as received, along with the metadata generated for it on ingestion.

REST, RESTful API

An application programming interface (API) based on Representational Statement Transfer (REST) architectural style. RESTful applications use HTTP requests and associated methods (POST, PUT, GET, and DELETE) to create, update, read, and delete data.