Data Management Environment at the National Cancer Institute

Sunita Menon
Cancer Data Science Initiatives,
Frederick National Laboratory for
Cancer Research, Frederick, MD
sunita.menon@nih.gov

Eran Rosenberg
Cancer Data Science Initiatives,
Frederick National Laboratory for
Cancer Research, Frederick, MD
eran.rosenberg@nih.gov

Yuri Dinh
Cancer Data Science Initiatives,
Frederick National Laboratory for
Cancer Research, Frederick, MD
yuri.dinh@nih.gov

Dr. Zhengwu Lu
Bioinformatics and Computational
Science,
Frederick National Laboratory for
Cancer Research, Frederick, MD
zhengwu.lu@nih.gov

Prasad Konka
Cancer Data Science Initiatives,
Frederick National Laboratory for
Cancer Research, Frederick, MD
prasad.konka@nih.gov

Dr. George Zaki
Cancer Data Science Initiatives,
Frederick National Laboratory for
Cancer Research, Frederick, MD
george.zaki@nih.gov

Udit Sehgal
Cancer Data Science Initiatives,
Frederick National Laboratory for
Cancer Research, Frederick, MD
udit.sehgal@nih.gov

Sarada Chintala
Cancer Data Science Initiatives,
Frederick National Laboratory for
Cancer Research, Frederick, MD
sarada.chintala@nih.gov

Ruth Frost
Bioinformatics and Computational
Science,
Frederick National Laboratory for
Cancer Research, Frederick, MD
ruth.frost@nih.gov

Dr. Eric Stahlberg
Cancer Data Science Initiatives,
Frederick National Laboratory for
Cancer Research, Frederick, MD
eric.stahlberg@nih.gov

ABSTRACT

An efficient and cost-effective mechanism is required to store and manage the large heterogeneous datasets generated by high throughput technologies such as Next Generation Sequencing, Cryo-Electron Microscopy, and High Content Imaging. High-performance tier 1 storage is expensive, and the affordable tier 2 devices used standalone do not lend themselves well to discovering and disseminating datasets. The Data Management Environment (DME), a data management platform for storing and managing high-value scientific datasets, was developed at the National Cancer Institute to close this gap. DME addresses the long-term data management needs of research labs and cores at NCI per the FAIR [1] (Findable, Accessible, Interoperable, and Reusable) guiding principles for data management. It supports S3 compatible object store, as well as file system storage. DME uses iRODS [2] as the metadata management layer enabling virtualization of backend storage, replacement of storage providers with zero impact on users, and transparent migration of data across providers. The granular permissions scheme provided by iRODS coupled with DME's authentication and authorization mechanism enables researchers to share data with collaborators securely. This paper will provide an overview of the capabilities and architecture of the Data Management Environment and discuss how DME has leveraged iRODS to deliver enhanced data management and storage management capabilities.

Keywords

DME, iRODS, data management, scientific datasets, metadata, virtualization, data migration, object store

iRODS UGM 2022, July 5-8, 2022, Leuven, Belgium

INTRODUCTION

Research labs and cores utilizing high throughput instruments regularly generate data at terabyte and petabyte-scale. The data collected from the instrument is moved to a local drive or network-attached storage, from where it makes its way to one or more computational servers and analysis workstations. Copies of the raw and analysis data are made to secure them, resulting in the generation of multiple redundant copies along the processing path. Often, researchers share the data with one or more collaborators, who make more copies along new processing paths. New files are often added to these directories, or the original files are reorganized to align with the analysis performed. After a while, the provenance information of the initial dataset is no longer available, making storage space recovery extremely challenging. Staff turnover only adds to the problem. The data stays forever in the expensive tier 1 storage devices. Some of it is moved later to the infrequently accessed tier 2 devices like tape storage, which is much cheaper but needs heavy investment in time and effort to retrieve the data. Since this data is not annotated, further effort is required to search and locate what is needed. These limitations also prevent the sharing and processing of data in integration and analysis platforms for further study and research.

To solve this problem, we need to store the data in cost-effective, reliable storage (Figure 1) from where it is directly and easily accessible for reuse. The data needs to be secured from unauthorized access while at the same time being shareable with collaborators whenever required. It should be annotated with the appropriate provenance and domain metadata, easily searchable and downloadable. It should be migratable to other storage devices when the lifecycle of the current device has ended. The migration should be transparent to users so that there is no burden on learning to use a new interface or technology to retrieve the data and no impact on the bioinformatics pipelines and data analysis platforms that access it programmatically. It should be easy to tier the data to cold storage or dispose of it when the predetermined lifecycle of the data has ended.

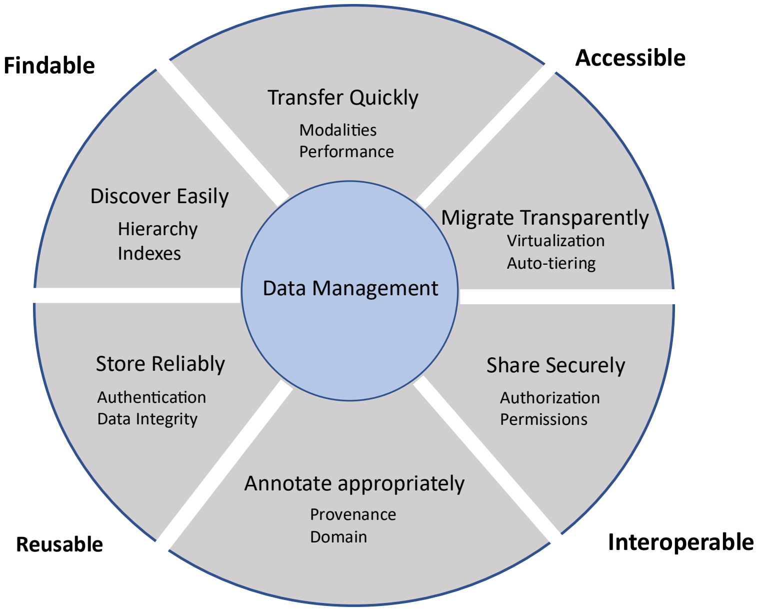


Figure 1. Data Management System Requirements

The Data Management Environment (DME) was developed to address these needs. DME is a metadata-based data management platform that provides secure, virtualized storage for high-value scientific datasets generated at NCI. Its reliable storage mechanism and the ease of accessing and sharing large datasets eliminate the need for users to maintain copies of datasets in their local or network-attached storage.

SYSTEM OVERVIEW

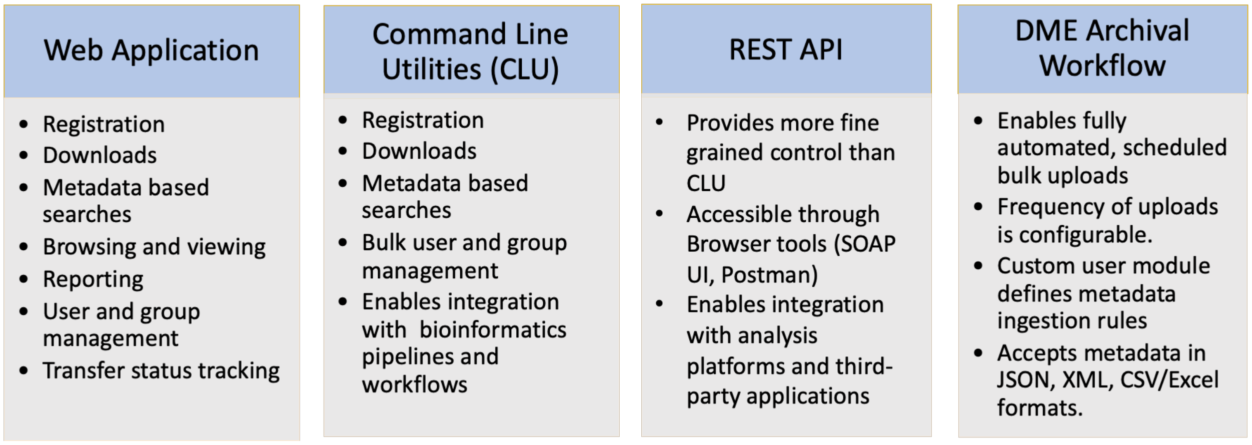
DME was designed to archive and share large, heterogeneous datasets. DME archives data to S3-based object stores that presently include Cleversafe [3], Cloudian [4], and Amazon S3 Glacier Deep Archive [5]. Support is also available for archiving to network file systems.

Data in DME is associated with provenance and domain metadata to enable the targeted discovery of datasets. DME performs all data management functions are performed through iRODS. These functions include the management of collections, data objects, user accounts, user groups, metadata, and permissions. Using iRODS for data management functions has enabled DME to perform storage virtualization, data migration, and data tiering transparently. These critical capabilities have provided users with a seamless data management experience, enabled secure data sharing, and significantly eased the IT functions required to manage large data volumes. It has also enabled DME to be domain agnostic, facilitating its broad adoption across NCI.

Interfaces

DME provides the following interfaces (Figure 2) for users to interact with the system:

- The DME web application enables users to easily browse, store, search and download data through an intuitive user interface. Transfer status screens enable detailed tracking of ongoing asynchronous bulk transfers. Users can create 'bookmarks' to enable quick access to desired collections. Other capabilities include user account management, group management, and reporting.
- The command-line utilities (CLU) provide shell commands for programmatic access from bioinformatics pipelines and workflows. CLUs can be used to store, search and download data. User account management and bookmark creation are also supported.
- The representational state transfer (REST) API suite provides fine-grained control of DME services and enables programmatic integration with data analysis platforms and external third-party applications.
- The DME Archival workflow supports users requiring regular bulk uploads. It enables fully automated archival of datasets. The workflow scans the source directories specified by the user at pre-configured intervals to locate the files to be archived. It then extracts the metadata from metadata input files based on the rules configured in a custom user module and uploads the annotations and the corresponding dataset to DME. Supported metadata file formats are JSON, XML, and Excel.



Web Application

- Registration
- Downloads
- Metadata based searches
- Browsing and viewing
- Reporting
- User and group management
- Transfer status tracking

Command Line Utilities (CLU)

- Registration
- Downloads
- Metadata based searches
- Bulk user and group management
- Enables integration with bioinformatics pipelines and workflows

REST API

- Provides more fine grained control than CLU
- Accessible through Browser tools (SOAP UI, Postman)
- Enables integration with analysis platforms and third-party applications

DME Archival Workflow

- Enables fully automated, scheduled bulk uploads
- Frequency of uploads is configurable.
- Custom user module defines metadata ingestion rules
- Accepts metadata in JSON, XML, CSV/Excel formats.

Figure 2. DME Interfaces

Authentication and Authorization

A DME user account is required to obtain access to the system. A user account can be created for a user only if the requested user identifier is present in the NIH Active Directory system. However, a user with an active DME account cannot access files or collections unless explicitly permissioned by the data generator.

To prevent the password and username from being sent for each CLU or REST API call, an access token is issued when the user successfully authenticates with the system using the token generator CLU or the authenticate API. The returned token can then be used for subsequent calls until it expires. The expiration period is configurable for a specific installation.

Transfer Modalities

Data can presently be uploaded from or downloaded to five endpoint types - Amazon S3 [6], Google Cloud [7], Google Drive [8], Globus endpoint [9], and the user's local file system. We continually evaluate and add new transfer modalities to make the system more broadly useable based on user feedback and NCI needs.

DME ARCHITECTURE

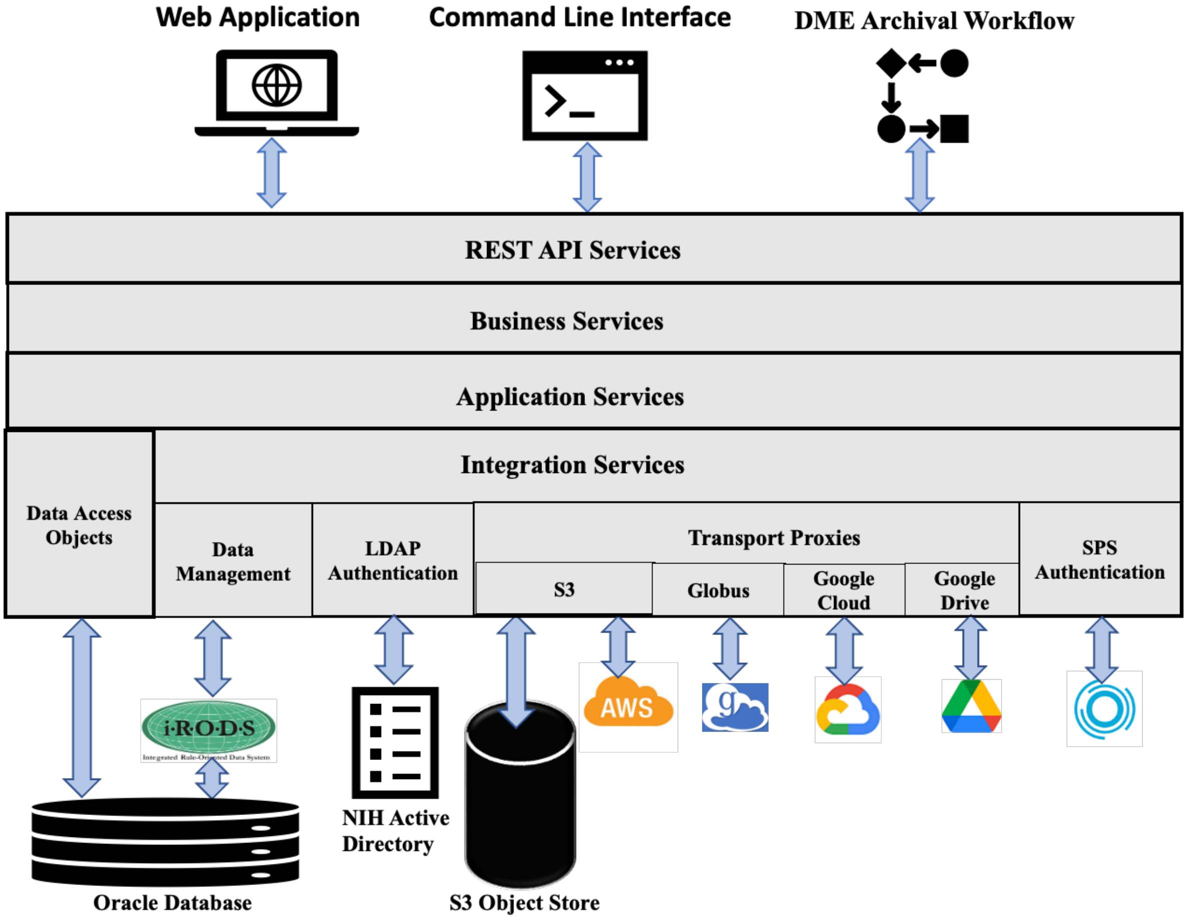


Figure 3. DME Logical Architecture

DME consists of the API server providing the platform core services, the DME web application that provides the graphical user interface, and the command-line interface (CLI) that is fronted by the command line utilities (CLU). The web application and the CLI/CLU communicate with the API server through the DME REST API. The API server includes 'schedulers' that perform various tasks in the background at separately configured intervals.

The DME production infrastructure consists of the following components:

- Tomcat 8 [10] server hosting the DME web application
- 6 API servers running on Apache ServiceMix [11]
- iRODS 4.2.9 server
- Oracle [12] 19c database server hosting the iRODS metadata database
- On-premises Cleversafe and Cloudian vaults
- Amazon S3 and S3 Glacier Deep Archive

All servers run on CentOS 7 machines with the default Java 8 installation. The scheduler is packaged as a separate ServiceMix feature enabling it to be deployed separately from the API server. The schedulers are distributed across the 6 API servers, enabling dynamic, horizontal scaling of services in response to user load.

We implemented the platform core services in a modular, layered architecture (Figure 3) to provide clean interfaces and separation of concerns, making it easy to maintain and scale the system. The services are implemented in Java using the Spring Framework. Each horizontal layer exposes its services through the layer's exported API, thus hiding the details of its implementation.

The REST API layer invokes the APIs exposed by the business services layer, which orchestrates one or more application services to deliver the requested service.

The data access objects (DAOs) connect to the Oracle database to write and read data to and from the DME tables and materialized views that have been set up to support reporting and other business requirements.

The integration services interfaces with external subsystems, enabling easy replacement of these subsystems without impacting the higher layers. It contains the following modules:

- LDAP authenticator authenticates the user credentials with the NIH Active directory.
- The Data Management module communicates with the iRODS server through the Jargon API [13].
- The Data Transfer Proxies manage the transfer of data to and from physical storage and the migration and tiering of data across S3 storage providers. The proxies include modules for interfacing with S3 providers, Globus, Google Cloud, Google Drive, and SPS [14]. This decoupling of data transfer functions enables easy replacement of the backend storage without impacting the data management functions.
- SPS authorization module provides the token received from third-party applications to the NIH authorization web service for verification.

DATA MANAGEMENT

Metadata is associated with both collections and data objects in DME. DME categorizes metadata as system and user metadata. System metadata is automatically captured in DME when a data object is created and cannot be added or modified by the user. It includes the file size, file archive location, checksum, data transfer type, data transfer status, and transfer date. The user metadata is provided by the user and consists of provenance and domain metadata. Provenance metadata is the same for all the user groups in DME and is collected for administrative and maintenance purposes, including determining the end of the data lifecycle. Domain metadata forms the backbone of efficient data discovery and is defined by the user depending on their data management workflow and the granularity of the searches required. The user metadata may be configured as mandatory or optional. The mandatory metadata is supplied during object registration and is validated during that time. The optional metadata can be added anytime during the data lifecycle and is not subject to validation. Most of the provenance metadata is mandatory.

DME provides flexibility to each Division/Office/Center (DOC) to define their data hierarchy (virtual folder organization) and metadata structure (attributes defined for each level in the hierarchy) in DME. These together constitute the metadata model, which consists of three JSON policy files structured as follows:

- Data hierarchy file: Specifies each collection type, whether it is container collection, and the parent of the collection

- Collection metadata validation rules: Specifies the attributes of a collection. The name, a brief description, the parent collection type, and the attribute type (whether mandatory or optional) are provided for each attribute.
- Data object validation rules: Specifies the attributes of a data object. The name, a brief description, the parent collection type, and the attribute type (whether mandatory or optional) are provided for each attribute.

A DOC can have multiple metadata models, one for each sub-group.

User and Group Management

DME has implemented REST APIs and command-line utilities to manage users and groups using IRODs. Users and groups can also be managed through the DME web application.

Group administrators and system administrators can create or delete users. They can retrieve all the users present in a group or all users having a specific role. A new user account can be created in DME only if that user has an active NIH Active Directory account. For creating the new account, only the NIH user identifier for that user is required. DME automatically populates the last name and first name from the NIH LDAP. Email notification is optionally sent to the user when the account is successfully created in DME. While adding a bookmark for a user, the user account is automatically created if it does not exist, and permissions are set on a file or collection for that user.

Group administrators and system administrators can create or delete groups. They can add or remove users to/from a group. They can search for groups and retrieve all the groups to which a user belongs. If metadata containing encrypted PII information is present, DME decrypts it for a user only if that user belongs to the DOC's DME security group.

Roles and Permissions

DME provides the ability to set the permission on all datasets, ensuring that only authorized users of the system can access the data. IRODS fine-grained permissions scheme, coupled with the authentication mechanism implemented by DME, has enabled highly secure data sharing with NIH collaborators.

The iRODS permissions scheme is applied to DME as follows:

- OWN permissions: The data generator uploading data to DME automatically gets OWN permissions for the file or collection. In the case of core facilities uploading data for clients, the data generator or group administrator grants OWN permissions to the data owner or the designee of the data owner.
- WRITE permissions: The data generator or data owner grants WRITE permissions to users who need to modify the metadata associated with the data asset.
- READ permissions: The data generator or the data owner grants READ permissions to researchers and collaborators who need to browse and download the data. Data can be shared with a larger audience by setting permissions for a group rather than individual users.

Permissions can be set on multiple files and collections simultaneously for a user. Additionally, multiple users can be permissioned to a file or collection.

DME uses the iRODS roles to manage the activities a user is permitted to perform. A role can perform all the tasks of the next lower role, in addition to the tasks described below:

- System Administrator (*rodsadmin*): This role is granted only to DME administrators. It enables them to create the metadata model for new onboarding DOCs and monitor the status of the data transfers initiated by users. System administrator privileges are required for data migration and tiering in DME. On request from the DOC, System Administrators may perform user or data management functions.
- Group Administrator (*groupadmin*): This role is granted to the data generators, lab managers, or bioinformatics analysts in a research lab or core. Group administrators perform data archival and user and group management functions for their DOC. They also set permissions on the collections and files for the researchers and collaborators requiring access.

- User (*rodsuser*): This is the default role assigned to researchers and collaborators working on a project. It enables them to view, search and download the files and collections they are permissioned to see. If they are provided OWN permissions to a collection, they can also register new sub-collections or files.

STORAGE VIRTUALIZATION

All users access data in DME through the logical path presented by iRODS. Users view the data through the data hierarchy they have defined. This hierarchy translates to a non-hierarchical prefix-based structure for the S3 bucket, with the logical path mapped to the key of the S3 object.

The physical path is stored as a metadata attribute of the data object, and the physical location and organization of data are transparent to the users. The metadata in DME is decoupled from the storage, enabling easy replacement of storage modalities. The storage provider URL is stored as a database configuration, and switchover of the S3 storage provider only involves changing this configuration.

Since all references to the data are made through the iRODS collections and data objects, changes to the backend storage have no impact on the systems and pipelines that are integrated with DME, enabling significant changes to the storage infrastructure with just a few minutes of system downtime.

DATA MIGRATION AND TIERING

As the data progresses through its lifecycle and gets used less frequently, moving it to a slower, cheaper storage is more cost-effective. Since the metadata is not attached to the data, this transition only involves setting up the lifecycle rules and storage class and invoking the appropriate S3 API to facilitate the transfer. Tiering REST APIs have been added in DME to enable tiering from Cleversafe and Cloudian to Glacier and Glacier Deep Archive. DME performs data retrieval from Glacier in two steps – it is first restored from Glacier to AWS S3 and then downloaded from AWS S3. An email notification is sent to the requesting user when the data has been restored to AWS S3.

Storage systems need to be replaced when they reach end-of-life or end of support, which requires the migration of all data to a new system. Using iRODs to manage the metadata separately has enabled seamless data migration from one storage provider to another. The virtual path presented by the data management layer and the metadata associated with a file or collection remains the same. Only the configuration parameters representing the file's physical location are modified. The Migration REST API enables the transfer of data from and to Cleversafe, Cloudian, or AWS S3. Further archiving to Glacier or Glacier Deep Archive can be performed by setting the appropriate lifecycle policies on the S3 buckets. All calls to retrieve the data after the migration will automatically fetch it from the new storage device, making the migration fully transparent to the user.

CONCLUSION

DME presently hosts over 4 Petabytes of data from 23 labs and cores across NCI, and the number is multiplying rapidly. The infrastructure has been scaled significantly over the past year to accommodate the growing demand. As the data continues to grow and DME gets leveraged by more and more stakeholders for data sharing and archival, the focus is on implementing innovative approaches to improve the performance further while adding new capabilities and enabling integrations with external platforms.

ACKNOWLEDGEMENTS

The authors would like to thank their colleagues at the Enterprise Information Technology group of the Frederick National Laboratories for Cancer Research for setting up the storage and infrastructure for this platform.

We are grateful to the various research labs and core facilities across NCI who have onboarded to DME and provided valuable feedback to improve the system.

This work has been funded in whole or in part with Federal funds from the National Cancer Institute, National Institutes of Health, under Contract No. 75N91019D00024. The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does the mention of trade names, commercial products, or organizations imply endorsement by the U.S government.

REFERENCES

[1] FAIR Principles, **https://www.go-fair.org/fair-principles**
[2] iRODS, **https://irods.org**
[3] IBM Cloud Object Storage, **https://www.ibm.com/cloud/object-storage**
[4] Cloudian, **https://cloudian.com/**
[5] Amazon S3 Glacier Storage Classes, **https://aws.amazon.com/s3/storage-classes/glacier/**
[6] Amazon S3, **https://aws.amazon.com/s3/**
[7] Google Cloud Storage, **https://cloud.google.com/storage**
[8] Google Drive, **https://www.google.com/drive/**
[9] Globus, **https://www.globus.org/**
[10] Apache Tomcat, https://tomcat.apache.org/
[11] Apache ServiceMix, **https://servicemix.apache.org/**
[12] Oracle Database 19c, **https://docs.oracle.com/en/database/oracle/oracle-database/19/index.html**
[13] Jargon core libraries, **https://github.com/DICE-UNC/jargon**
[14] Lightweight Directory Access Protocol, **https://ldap.com/**
[15] One Identity – Safeguard for Privileged Sessions, **https://www.oneidentity.com/products/one-identity-safeguard-for-privileged-sessions/**