VA Enterprise Design Patterns
Interoperability and Data Sharing

Data Storage

OFFICE OF TECHNOLOGY STRATEGIES (TS)
OFFICE OF INFORMATION AND TECHNOLOGY (OI&T)

VERSION 1.0
DATE ISSUED: JUNE 2016

APPROVAL COORDINATION

Gary E. Marshall
137891

Digitally signed by Gary E. Marshall
137891
DN: dc=gov, dc=va, o=internal,
ou=people,
0.9.2342.19200300.100.1.1=gary.marshall
@va.gov, cn=Gary E. Marshall 137891
Date: 2016.06.15 09:10:48 -04'00'

_____

Gary Marshall
Director, Technology Strategies, ASD

PAUL A.
TIBBITS
116858

Digitally signed by PAUL A. TIBBITS
116858
DN: dc=gov, dc=va, o=internal,
ou=people,
0.9.2342.19200300.100.1.1=paul.tibbit
s@va.gov, cn=PAUL A. TIBBITS
116858
Reason: I am approving this document.
Date: 2016.06.15 12:53:33 -04'00'

_____

Paul A. Tibbits, M.D.
DCIO Architecture, Strategy, and Design

REVISION HISTORY

Version	Date	Approver	Notes
1.0	6/7/2017	Nicholas Bogden	Revised sections 1 and 2, fleshed out Section 3 based on vendor input, and incorporated team and stakeholder feedback in.

1 INTRODUCTION

The Department of Veterans Affairs (VA) is developing standardized approaches to deployment and management of reusable data storage capabilities to support data architecture objectives in VA's Enterprise Architecture (EA). VA will provide an array of data storage options to all projects as part of its adoption of Enterprise Shared Services (ESS) in accordance with the Enterprise Technology Strategic Plan (ETSP).

This Enterprise Design Pattern (EDP) guides project teams to criteria for selection of data storage technologies, as well as criteria for re-platforming legacy data to more current technologies.

1.1 Business Need

Approaching data storage as an enterprise service contributes to the overarching goal to connect VA business needs with supporting IT capabilities. Specifically, an enterprise approach to data storage will address the following concerns:

- Establish and maintain consistent and compliant security baselines for data storage technologies
- Build a unified, interoperable, hybrid data layer for EA that supports seamless access to and sharing of data

- Increase cost efficiencies for data storage acquisition, provision, deployment, and management
- Transition VA from a system-centric environment to a service-oriented and data-centric environment
- Facilitate migration of VA's systems, applications, workflows, data, and other digital assets to the cloud

Also, individual data, application, and system owners need an enterprise data storage strategy to address their own concerns. Architects and developers building a new application or workflow have many potential options in terms of database management systems (DBMS) and products. Selecting the most appropriate and cost-effective solution(s) for their business and functional needs can be difficult.

Data and system owners dealing with datasets on legacy technologies face additional challenges. They may find it difficult to determine whether or when to re-platform their data to a new technology, (particularly in a virtualized environment). If they have decided they want or need to re-platform, they may have difficulty justifying the processes' significant cost.

The proposed solutions in other VA EDPs related to Interoperability and Data Sharing[1] involve or require a service-oriented approach to data storage.

1.2 Approach

An initial component of an enterprise-wide data storage strategy involves providing projects with a standardized approach to data storage selection. This will help individual system and data owners make better decisions regarding:

- What type of data storage platform(s) to use for a particular set of requirements
- Whether, why, and when to re-platform data from legacy technologies

The proposed approach to data storage selection applies to choosing one or more technologies for re-platforming data and to service-oriented data storage. It can also be used to select cloud storage systems or articulate requirements to cloud service providers (e.g., through a service-level agreement).

This document guides technology prioritization and selection criteria for the One- VA Technical Reference Model (TRM).

---

[1] Refer to 3.2 Hybrid Data Access (HDA) and 3.4 Enterprise Data Analytics.

2 CURRENT CAPABILITIES AND LIMITATIONS

2.1 Capabilities

VA is moving toward more centralized, consistent, enterprise-level management of data stores. VA has instituted the VA Data Inventory (VADI) and Data Architecture Repository (DAR), inventories for data stores and metadata, respectively. VA's regional data centers shape and influence adoption of storage technology within the Department. The Business Intelligence Service Line (BISL) and warehouse governance boards drive technology prioritization and acquisition for analytics.[2]

The TRM provides the beginnings of an approach to technology selection for data storage and other purposes. It is a component within the overall EA that establishes a common vocabulary and structure for describing the information technology (IT) used to develop, operate, and maintain enterprise applications. The TRM serves as a technology roadmap and tool for supporting the Office of Information & Technology (OI&T).

Anyone can submit products, technologies, or standards to the TRM site for review by subject matter experts (SME) from the VA pillars. The SMEs evaluate submissions based on such criteria as:

- Functionality, sufficient to meet consumer's needs
- Appropriateness for and compatibility with the VA network
- Availability and strength of security controls[3]

Reviews also take into account how long a particular solution will be supported by its developer or vendor.

At the end of the review process, the technology is mapped to the VA Decision Matrix. The matrix displays the current and future VA IT position regarding different releases of a TRM entry.[4] These decisions are based upon the information available as of the decision date. The consumer of this information has the responsibility to consult the organizations responsible for the desktop, testing, and production environments to ensure that the chosen technology

---

[2] At the time of this writing (March 2016), they do not yet manage analytics VA-wide, but 3.4 Enterprise Data Analytics establishes them as the future enterprise service provider for all Department analytics.

[3] Reviewers from VA's Office of Information Security (OIS) ensure that prospective technologies support compliance with applicable information security policies and standards, including those published in VA Handbook 6500.

[4] The decision matrix displays the decision status of the technology for the four quarters of the current year, past year, and next year.

solution will be supported. Any "major.minor" version of a solution that is not listed in the VA Decision Matrix is considered unapproved for use.

There are six color-coded TRM decision statuses in the decision matrix defined in Table 1.

<p align="center">TABLE 1: TRM DECISION MATRIX LEGEND</p>

Color	Decision Status
White	**Approved**: The technology[5] has been approved for use.
Yellow	**Approved w/ Constraints**: The technology can be used within the constraints specified in the technology's TRM entry.
Gray	**Unapproved**: The technology has not been approved for use. It may only be used if a waiver signed by the Deputy Chief Information Officer (DCIO) of Architecture, Strategy, and Design (ASD) based upon a recommendation from the Architecture and Engineering Review Board (AERB) has been granted to a project.
Orange	**Divest**: The technology is currently in use at VA, but VA has decided to retire it and transition to an alternative solution. All projects currently utilizing the technology must plan to transition away from the technology, and the technology may not be used for new projects/systems. Additional information on when the entry is projected to become Unapproved may be found on the Decision tab for the specific entry.
Black	**Prohibited**: The technology is not (currently) permitted to be used under any circumstances.
Blue	**Planning/Evaluation Constraint:** The technology is currently being evaluated, reviewed, and tested in controlled environments. Use of this technology is strictly controlled and not available for use within the general population. If a customer would like to use this technology, they are directed to work with the local or Regional OI&T office and contact the appropriate evaluation office.

Most system and data owners select data storage technologies from among those labeled "Approved" or "Approved w/ Constraints." Some system and data owners may use unapproved technologies, provided they obtain a waiver.

Approved technologies have the native capacity to comply with VA functional, interoperability, performance, business, and security requirements. Technologies that are Approved w/Constraints can only meet these requirements under certain conditions, or with particular add-ons and compensating controls. See Figure 1 for examples of technologies with compliance issues, and the constraints and compensating controls that might be required by the TRM.

---

[5] The TRM uses the generic term "technology" to refer to all TRM entries, whether they are actually technologies or standards.

DBMS #1
- **ISSUE:** Does not natively support Federal Information Processing Standard (FIPS) 140-2 encryption.
- **CONSTRAINT:** Must be used with an add-on to provide capability for FIPS 140-2 encryption.

DBMS #2
- **ISSUE:** Uses proprietary technology that does not interconnect with most other systems.
- **CONSTRAINT:** Deploy with middleware to facilitate interoperability with other databases.

DBMS #3
- **ISSUE:** Does not easily support mirroring, striping, or virtualization, which makes it difficult to implement business continuity or failover solutions.
- **CONSTRAINT:** Do not use for mission-critical data or high-performance applications.

DBMS #4
- **ISSUE:** Does not support encryption or granular access control.
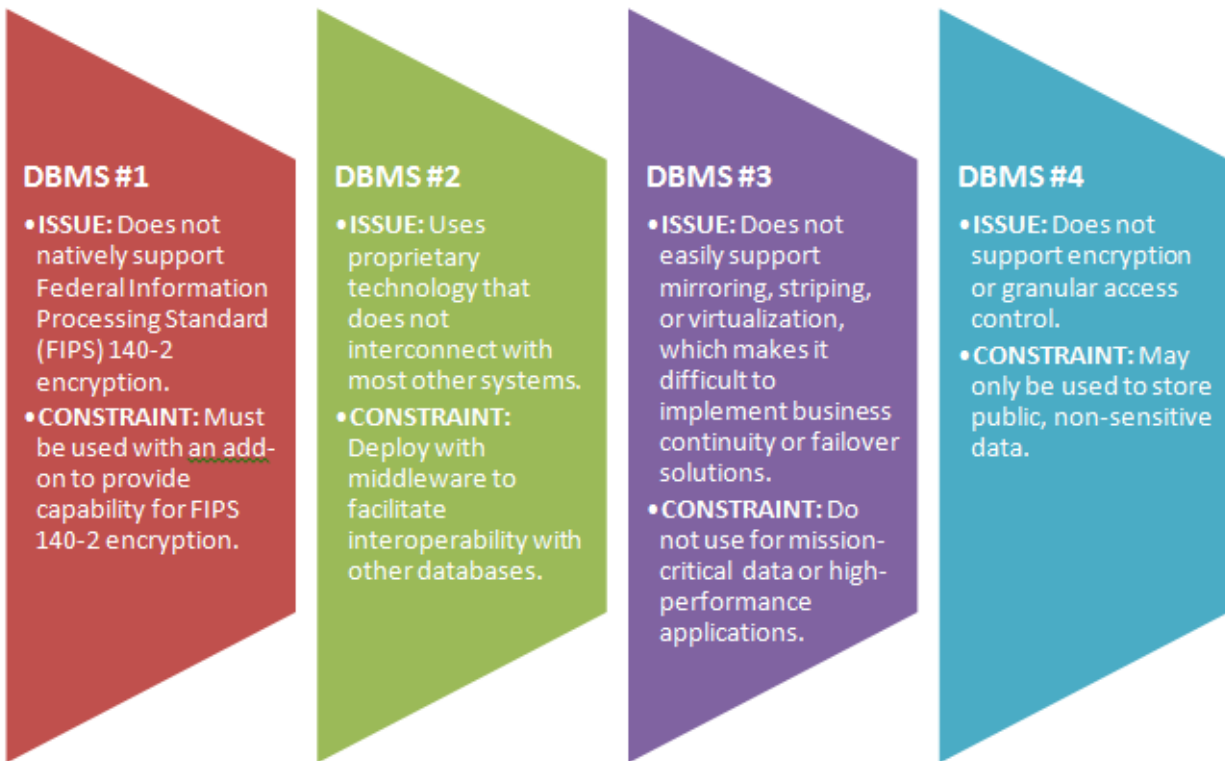- **CONSTRAINT:** May only be used to store public, non-sensitive data.

FIGURE 1: EXAMPLES OF TECHNOLOGIES THAT ARE "APPROVED W/ CONSTRAINTS" IN THE TRM

Categories of TRM data storage technologies are listed in Section 3.4.

2.2 Limitations

Until recently, VA lines of business (LOB) and offices acquired and managed their own assets with minimal Department-wide guidance or control. The TRM has helped to mitigate this issue by establishing and articulating baselines for data storage solutions. Curators of the TRM struggle to prioritize and evaluate the large number and variety of solutions to be cataloged.

VA system and data owners lack coherent guidelines for how to re-platform data from one technology to another, as well as established criteria for when to re-platform data. While re-platforming may be desirable in many cases, it is a costly and difficult prospect that necessitates a careful approach. Also, if data owners identify a need to re-platform data, they may have difficulty conveying that sense of urgency to those who control their budgets.

VA will not be able to implement a service approach to data storage until it addresses these gaps with appropriate guidance.

3 FUTURE CAPABILITIES

This section contains guidelines and criteria to re-platform data from legacy technologies to new systems using more current technologies. This helps project managers and sponsors articulate their business needs and justifications for making this transition. Most of this guidance comes from existing VA sources or industry best practices. The guidance is consistent with VA's intent to leverage cloud services per the ETSP, in that it encourages moving data from legacy technologies to platforms that are better suited to virtualized/cloud and service-oriented architectures.

In addition, this section presents consistent, mission-driven approaches for selecting appropriate data storage types for a current or emerging business purpose. The approach is informed by factors such as data temperature, data structure, and security and functional needs (described below). This approach to technology selection will be used whether the business purpose is for a new dataset or an existing dataset being re-platformed. It will also drive any automated policies or business processes for sorting different types of data input into the appropriate storage platform(s).

Refer to Appendix B for descriptions of data storage types, their characteristics, and potential applications and use cases.

3.1 Re-Platforming Data

Re-platforming data means migrating data from its original hosting platform to one with a different DBMS. The new platform may or may not be the same storage type as the original one. Some examples of re-platforming are:

- Moving data from one relational database management system (RDBMS) to another RDBMS from a different vendor
- Hosting data originally contained in an RDBMS in a new system with a columnar database management system (CDBMS)
- Taking data from a hardware platform and hosting it with the same RDBMS, but in a virtualized/cloud environment

The guidance in this section focuses on identifying and justifying the need to perform the first two types of re-platforming.

Data owners and stewards will make, or at least be involved in, the decision to re-platform a VA dataset. Unless and until a particular dataset has an officially designated and active owner/steward, it is not a candidate for re-platforming.[6]

Data should only be considered for re-platforming if:

1. The data is still useful, and
2. Redundant or similar datasets do not exist elsewhere, or are not usable.

If there is an alternative source of the same data available – particularly in an authoritative data source (ADS) – the target data source will be merged or reconciled with that alternative source. The original dataset may still be archived, but it will be retired from active use.

Beyond that, criteria for re-platforming data stems from the determination that the technology currently in use cannot meet new requirements or demands of the enterprise's future direction. Such determinations may be made in the TRM: any data stored on a technology that is soon to be listed as "Divest" or "Unapproved" will be re-platformed to a new technology.

Other criteria for re-platforming are that:

- The logical access patterns for the data have changed and cannot be supported by the current technology implementation, inhibiting the workflow in which the data is used
- Application performance requirements cannot be met
- Security requirements cannot be met
- Resourcing technology changes and maintenance is a foreseeable challenge (e.g., because the vendor will no longer support the solution)
- The current implementation does not support a service-oriented architecture (SOA) or data as a service (DaaS)
- The technology is not compatible with contemporary industry and de facto standards for data access and data transport[7]
- The solution does not have multi-core processing capabilities

Data will also be re-platformed if the application it supports is slated to be retired, replaced by a new application, or re-designed.

---

[6] The need for designated owners/stewards is addressed in 3.2 Utilizing Enterprise Identities and 3.3 Enterprise Data Analytics.
[7] To include Structured Query Language (SQL), JavaScript Object Notation (JSON), and Representational State Transfer (REST).

3.2 Data Storage Universal Requirements

All data storage technologies must meet baseline functional and security requirements to be approved for use (either as-is or with constraints) in the TRM. In practice, this means they must support certain security and functional capabilities to be used in certain ways – or to any extent at all – within VA.

Technologies approved "as-is" support these capabilities completely and natively. They can be used without modification or compensating controls. Technologies approved with constraints either provide only partial support for these capabilities or require some type of add-on (e.g., a compensating control) to enable them.

A technology that cannot fully meet the security or functional capability requirements detailed in Sections 3.2.1 and 3.2.2 may still be "Approved w/ Constraints." Even "Unapproved" technologies may be used with a waiver. In both cases, the technology must be deployed with compensating controls, used in specific circumstances or environments (but not others), or both. For example, one type of database that does not support granular access control may be suitable for non-sensitive data but not for protected health information (PHI). The constraint on this technology is that it can only be used for non-sensitive data.

System architects, developers, and data owners will make a good-faith effort to:

- Use technologies that are Approved or Approved w/Constraints for their data, systems, and applications
- Obtain the requisite AERB waiver when they must use Unapproved technologies
- When deploying technologies that are Approved w/Constraints or Unapproved, implement all required/recommended compensating controls

3.2.1 *Security Capabilities*

All data storage technologies used at VA will support the necessary security capabilities to fulfill statutory, regulatory, and agency requirements, especially those in the VA 6500 Handbook.

These security capabilities are:

- Support FIPS 140-2 encryption for data at rest and data in transit. Any database technologies without this capability are Unapproved in the TRM
- Observe operating system security for data stored in-memory
- Enable the use of hardware-accelerated encryption
- Enable the use of key management technologies to facilitate encryption at scale

- Enable same level of security for on-demand accessible and archived data

Cloud-based data storage solutions will also include controls to maintain the security of data in a multi-tenant environment. VA will exercise preference for data storage technologies that have these controls built in (as opposed to requiring that they be added on).

All data storage solutions will support Role-based Access Control (RBAC) at some level, although different data storage types naturally support RBAC at different levels of granularity. Relational databases usually support granular access control that will, for example, allow a specified client[8] to read some fields in a record but not others.[9]

The same is not necessarily true of other types of databases, which may only be able to apply access control at the file, document, folder, or collection level. In some cases, the application that uses the database can be used as a compensating control to provide more granular access control than the database itself will support. While this approach is feasible, it is generally more cumbersome and less reliable than using a database with built-in granular security controls. This may present an unacceptable risk for databases that contain, or will contain, sensitive data such as personally identifiable information (PII) or personal health information (PHI).

3.2.2   Functional Attributes

The critical functional attributes for data storage technologies within VA are:

- Support for common industry access standards, notably SQL and RESTful services
- Support for parallel and distributed processing, as well as management
- Can be virtualized or containerized
- If cloud-capable (which is preferred): deploy, manage, and interoperate on-premise, off-premise, or as a hybrid, and include multi-tenant awareness
- Enable separation of concerns between business logic and data access logic
- Support for modern hardware advances or for a roadmap that includes the disruptive and compelling features

3.3 Mission-Driven Selection Criteria

Other criteria for selecting data storage types will be driven by the mission, business, and application requirements of a particular data use case. This section focuses on criteria for selecting one or more data storage types based on:

---

[8] In this context, a user, application, or process.
[9] Integration with enterprise authorization services, including those that support other forms of access control such as Attribute-Based Access Control (ABAC), are addressed in the EDP for Enterprise Authorization (draft).

- Data temperature
- Complexity
  - Data structure
  - Query structure
- Concurrency

Some of these criteria are also referenced in Section 3.4 of the Enterprise Data Analytics EDP.[10]

The response surface in Figure 2 plots a number of possible storage types in terms of data temperature (X-axis) and structural complexity (Y-axis). The options shown are not an all-inclusive list of the types of data stores addressed in this document.
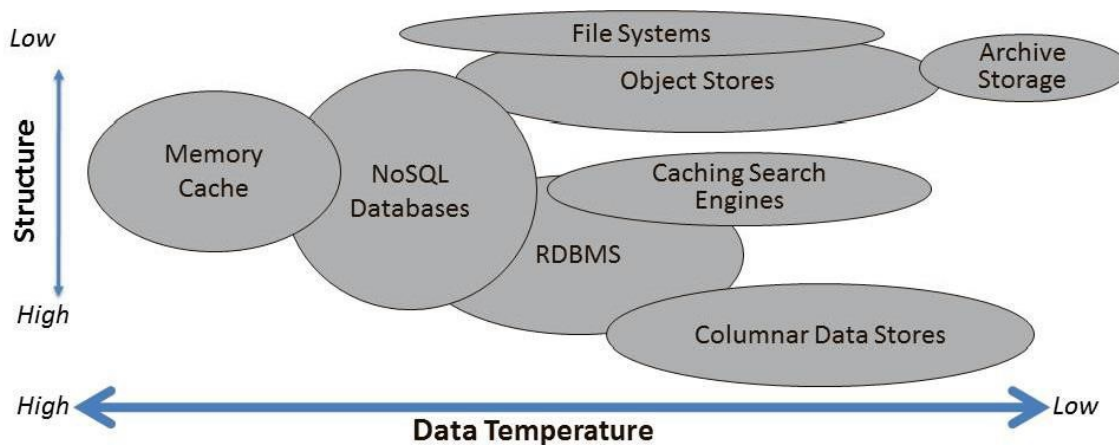

FIGURE 2: RESPONSE SURFACE FOR SELECTING DATA STORAGE TYPE

In some circumstances, a data, system, or application owner will need to employ more than one data storage type to meet the needs of a use case. However, it is a best practice to limit the number of types or solutions used to a few options with a broad set of features.

Refer to Appendix B for a list and description of data storage types, including their respective advantages, drawbacks, and potential scenarios for use.

3.3.1   Data Temperature

Data temperature is a critical consideration in selecting appropriate data storage technologies. Temperature – hot, warm, or cold – is the combination data characteristics shown in Figure 3:

---

[10] http://www.techstrategies.oit.va.gov/docs/designpatterns/Interoperability_and_Data_Sharing_Design_Pattern-Enterprise_Data_Analytics_2015_02_17_v1.0.pdf.

Latency
- Amount of time for a query to be executed on the database (milliseconds, seconds, minutes, hours)

Volume
- Maximum quantity of stored data used in a typical processing operation (MB, GB, TB, PB)

Item Size
- Size of individual data elements or data sets ingested into the database in a given instance (B, KB, MB, TB)

Request Rate
- Frequency at which operations are performed on/with the data (Low, Medium, High, Very High)

Durability
- Degree to which committed data transactions will survive permanently (Low, Medium, High, Very High)

FIGURE 3: CHARACTERISTICS THAT FACTOR INTO DATA TEMPERATURE

The temperature of data is determined with respect to a particular use case and not the source(s) or structure of the data itself. A system or application may need to store or process data at multiple temperatures to support different kinds of use cases or workloads. For example, data from a medical monitoring device has a different temperature with respect to each of the following analytic use cases:

- **Hot:** Alert clinicians to potentially dangerous changes in a patient's vital signs, so they can respond quickly
- **Warm:** Track changes in a patient's condition over the course of days or a week to evaluate how they are responding to medication
- **Cold:** Track long-term outcomes for similar populations with similar conditions

This is also an example of data cooling in temperature over time. The initially hot data is used for alerting. After it cools down to a warm temperature, it is used for tracking a patient's health over time. It cools down still further into cold data used for a years-long population study.

Data temperature tends to correlate with cost per unit of storage: the hotter the data temperature, the higher the cost per unit. Conversely, lower data temperatures incur a lower cost per unit of storage.

Data structure does not factor into data temperature. Table 2 details the temperatures and characteristics for different types of data storage.

TABLE 2: DIFFERENT DATA STORAGE TYPES AND THEIR TEMPERATURES

	Memory Cache	NoSQL	RDBMS	Caching Search Engine	MapReduce (HDFS)	Columnar Database	Object Storage	Archive Storage
Average Latency	ms	ms	ms, sec	ms, sec	sec, min, hrs	ms, sec (by size)	ms, sec, min (by size)	hrs, days
Data Volume	GB	GB-TB	GB-TB	GB-TB	GB-PB (by # of nodes)	GB-PB	GB-PB	GB-PB
Item Size	B-KB	KB-MB	KB-MB (by row size)	KB-MB	MB-GB	KB-GB	KB, GB, TB	GB-TB
Request Rate	Very High	Very High	High	High	Low-Very High	Low-Very High	Low-Very High	Very Low
Durability	Low-Moderate	Very High	High	High	High	Very High	Very High	Very High
Temperature	Hot		Warm			Cold		Frozen

- **"Hot"** data is typically associated with high-speed, low-volume stream input where the window of opportunity to take action on information gleaned from the data is very small.
- **"Warm"** data is associated with transactional or (relatively) small-batch input that is queried every few minutes, hours, or days. Warm data is somewhat more durable than hot data.
- **"Cold"** data is typically associated with long-term data that may be searched or analyzed with complex queries. In cold data, accuracy and consistency (rather than speed) is the priority.
- **"Frozen"** data applies solely to archiving. Frozen data is kept to fulfill backup, retention, and data security requirements. It may never be accessed at all, unless it is needed for business continuity, legal discovery, or forensic purposes.

In addition, certain data temperature ranges tend to correlate with certain types of data ingestion:

- **Stream:** Constant flows of data from sources such as sensors, Internet of Things (IoT) devices, clickstreams, and mobile devices. Tends to correlate with hot data.
- **Transactional:** Periodic inputs/updates from manual inputs or individual database reads/writes. Tends to correlate with warm (sometimes cold) data.
- **Files:** Regularly scheduled, high-volume transfers of batches, logs, or objects. Tends to correlate with cold data.

3.3.2 *Data Structure Complexity and Query Structure Complexity*

Data temperature is the first criteria for data storage selection. The other criteria are related to the complexity of the data's structure and the complexity of any queries run on it.

Figure 4 is a decision grid that compares data storage options in terms of data structure complexity (Y axis) and query structure complexity (X axis).[11]
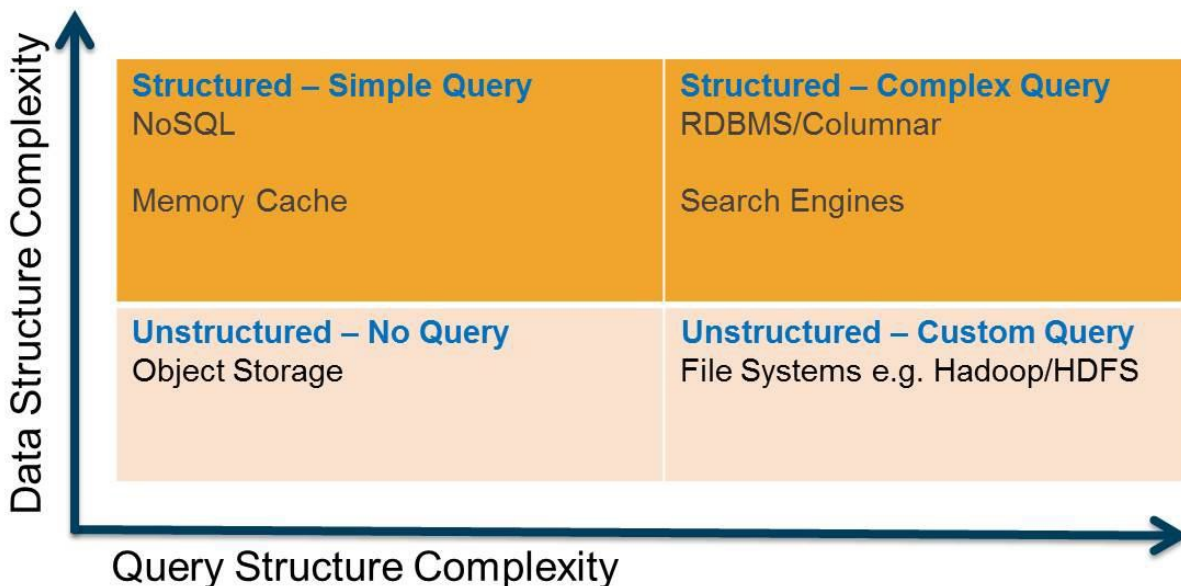
FIGURE 4: DECISION GRID FOR EVALUATING STORAGE OPTIONS

Different data storage types fall in different quadrants of the grid, depending upon the relative level of structure and query complexity they support. Options in the same data temperature

---

[11] The example decision grid in this figure contains storage types for multiple data temperatures, but the decision grid used in an actual selection process will only include options from a single temperature range.

range may fall into different quadrants, and options in different temperature ranges may fall into the same quadrant.

The combination of data temperature and complexity for a given use case will be used to determine and select the most appropriate type or types of data storage for that use case.

3.3.3   Concurrency

In order to guarantee atomicity, consistency, isolation, and durability (ACID) properties in databases, DBMSs must carefully manage changes to data through concurrency controls. Concurrency controls ensure that data currently being edited in a client is locked until the edit is complete.[12]

While almost all databases have concurrency controls, they may address concurrency in different ways. Some databases allow multiple clients to simultaneously read and write different data on the same file, object, table, or other entity. Other, less granular databases will allow multiple simultaneous reads of a particular entity, but only allow one client at a time to write. Still others will completely lock access to an entity in such a way that it can only be read or edited by one client at a time.

The concurrency requirements of a use case are one of the critical considerations in selecting an appropriate data store for that use case. If a use case involves multiple simultaneous transactions on a particular table of records, the data store selected for the use case needs to support multi-client concurrency. If a use case involves a chain of review and editing for a document, the data store for that document should restrict access to whoever is editing that document at a given time.

3.4 Alignment to the One-VA Technical Reference Model (TRM)

This section provides examples of TRM-approved tools that are currently used, or may be used, in the VA Information and Analytic Ecosystem. Future data storage capabilities are bound by approved technologies and standards cataloged in the TRM.

The TRM does not have categories for some of the data storage types listed in Appendix B.

---

[12] That may mean a change being committed to the database, or a termination of the editing process without any changes being saved.

Tool Category	Example Approved Technology
Columnar DBMS	N/A[13]
In-Memory Database/Memory Cache	N/A
Non-Relational Database	FIS-GTM
Object-Oriented DBMS	• Cache – 2012.1, 2012.2, 2014.1.3 • Cache Management Portal SQL Interface - 2012.2, 2013.1, 2014.1 • Cache Objects – 2012.2 • Oracle Database – v12.1.x
Relational DBMS	• Microsoft SQL Server – 2012 • Oracle Database – v12.1.x

3.5 Alignment to Veteran-Focused Integration Process (VIP)

The Veteran-focused Integration Process (VIP) is a Lean-Agile framework that services the interest of Veterans through the efficient streamlining of activities that occur within the enterprise. The VIP framework unifies and streamlines IT delivery oversight and will deliver IT products more efficiently, securely and predictably. VIP is the follow-on framework from Project Management Accountability System (PMAS) for the development and management of IT projects, which will propel the Department with even more rigor toward Veteran-focused delivery of IT capabilities.

All projects will select appropriate data storage technologies to address user stories and Epics that capture the business needs for an IT solution. The initial planning for data storage solutions will occur prior to Critical Design #1 in accordance with this EDP and VA Directive 6551 (Appendix D). Projects will define TRM-approved data storage technologies as part of its solution architecture, which is validated by stakeholders and the AERB prior to Critical Decision

#2. Projects will have the flexibility to change data storage solutions via iterative releases (referred to as "Builds" in VIP) and lessons learned gathered during sprint retrospectives.

4 USE CASES

The following use cases demonstrate the application of the capabilities and recommendations described in this document.

---

[13] Although the TRM does not currently include any approved columnar DBMS tools, vendors of widely used (and TRM-approved) relational databases are now developing and releasing columnar DBMS products.

4.1 Multiple Storage Types: Hybrid Data Access Involving Structured and Unstructured Data

4.1.1   Purpose

Many applications, systems, and use cases will require only one type of data store. Some use cases will involve using more than one type of data, and therefore more than one type of data store. For example, a certain workflow may involve both structured and unstructured data, as the application in the following use case does.

4.1.2   Assumptions

- The application in question is an operational transaction register that tracks Veteran appointments with Benefits Counselors
  - "Warm to cold" data use case
  - System of record
  - Requires granular security to protect personally identifiable information (PII)
- All users are able to log in to the application successfully, using VA's external Single Sign-On (SSOe) or internal Single Sign-On (SSOi) as appropriate
- The application's databases and any authoritative data sources (ADS) used are:[14]
  - Part of the VA EA data layer described in the HDA design pattern
  - Transparently accessible through authoritative information services and the Enterprise Create, Read, Update, Delete (eCRUD) logical wrapper
- All data in the transaction is encrypted in transit and at rest using FIPS 140-2 encryption

4.1.3   Use Case Description

1. A Veteran uses a web interface or app to make an appointment with a Benefits Counselor at a local office. The information the Veteran provides includes:
   a. Data elements selected from a list or limited set of options:
      i.   Date and time of appointment
      ii.   Name of a Benefits Counselor
      iii. Type of benefit the Veteran is interested in
   b. A written note with any additional questions or concerns the Veteran has
2. The information submitted by the Veteran is written to the application's database back-ends, as appropriate:
   a. All items selected from lists or limited sets of options are entered into a durable RDBMS record with an appointment ID, where they are appropriately secured

---

[14] The application uses IDs that correlate with the Master Veteran Index (MVI), for both Veterans and Benefits Counselors serving them.

        i.   Data elements containing PII are masked with encryption and access is limited to a few users, such as the Veteran and the Veteran's Benefits Counselor

       ii.   Data elements containing less sensitive information, such as the date, time, and location of the appointment, are accessible by other clients that need to know about the Benefits Counselor's availability

b. The note is entered into a less durable semi-structured data store and tagged with the same appointment ID as the associated DBMS record, as well as an ID that correlates to the Veteran's record in MVI

       i.   The note is encrypted so that only users authorized to access the corresponding DBMS record are able to see it, and only the Veteran can edit the note

3. The Benefits Counselor looks up his appointments through a staff-facing interface in the application

   a. He can read (in all data stores):

       i.   All the appointments that have his name and ID in the "Counselor" column in the structured RDBMS record

       ii.   All notes associated with those appointments (i.e., with the same appointment ID)

   b. He can write (in the relational data store):

       i.   Whether the appointment was completed, canceled, or rescheduled

       ii.   Whether a follow-up appointment was scheduled (Yes or No)

       iii. The appointment ID of any follow-up appointment.

   c. In the semi-structured data store, he can write his own notes on the appointment

       i.   These are tagged with the appointment ID and an ID that correlates to his record in MVI

       ii.   He and his supervisor can access this note, but the Veteran cannot

4.2 Multiple Storage Types: Same Data, Different Temperatures

4.2.1 Purpose

As noted in section 3.2.1, the same data from the same source can be stored and employed in multiple analytic data flows, each with a different temperature.[15] This use case describes an application that collects heart rate data and uses it at three different temperatures – hot,

---

[15] Refer to the glossary for a definition of data flows, and/or to Data Sharing and Interoperability 3.4: Enterprise Data Analytics for an in-depth explanation.

warm, and cold – for different purposes. The practice described here is also known as tiered analytics.

4.2.2 Assumptions

- Data is collected using either a dedicated heart rate monitor or a fitness tracking device that incorporates a heart rate monitor
- Once it has been collected and stored, heart rate data is not editable:
  - No one can change it, although authorized VA users can attach metadata to it to facilitate analysis
  - Data is automatically deleted from a work stream once it reaches an incompatible age (different for each data temperature)
- Analytic processing for this application occurs as described in Data Sharing and Interoperability 3.4: Enterprise Data Analytics.
- All other aspects of the solution are consistent with the Internet of Things (IoT) EDP (draft)

4.2.3 Use Case Description

1. A Veteran with hypertension wears a heart rate monitor that tracks his heart rate and heart rate patterns from second to second
2. The data from the heart rate monitor is streamed back to VA for analysis[16]
3. Once received, the streaming data is sequenced and sent into three distinct analytic pathways at different temperatures:
   a. Hot: Data is stored in a memory cache for streaming analysis, and evaporates after a few minutes
      i. The data in the memory cache is analyzed for signs that the patient is about to have (or may be having) a hypertensive crisis
      ii. When the device detects certain anomalies, it sends an alert to the patient and the local VAMC through Veteran-facing and clinician-facing apps
   b. Warm: Data is stored in a time series database that tracks samples of the Veteran's heart rate (and average resting heart rate) taken several times per day over a period of days or weeks
      i. This data is used by clinicians to track trends in the Veteran's heart rate and evaluate the effectiveness of prescribed medications and lifestyle changes to manage the Veteran's hypertension

---

[16] The data is streamed either directly to VA or through an authorized service provider incorporated with VA's Analytic Ecosystem.

c. Cold: Anonymized time series heart monitor data from a large number of Veterans – with new metadata attached – is copied to a semi-structured data store

    i. This data, collected over a multi-year period, will be processed with machine learning tools as part of a long-term population study

APPENDIX A. SCOPE

The purpose of this EDP is to guide selection of a data storage type for a particular business purpose. This includes both new applications and existing datasets being transitioned from legacy technology. The scope consists of:

- Criteria for re-platforming data from legacy technologies
- Data storage as a service approach
- Selecting data storage/database technologies
- Cost per unit of storage considerations
- Using data temperature, data structure complexity, and query structure complexity to inform technology selection
- Appropriate/recommended use cases for different types of data storage, including different types of NoSQL databases

The following concepts are outside the scope of this design document:

- Data storage technology criteria and baselines already covered by the TRM[17]
  - Security capabilities (e.g., granular access control, FIPS 140-2 encryption, compensating controls)
  - Open standards and interoperability
  - Network functionality
  - Minimal performance requirements
- Data formatting or modeling
- In-application, in-device, or embedded data storage/databases
- Infrastructure and hardware design specifications
- Vendor-specific products

Intended Audience

The primary audience for this document consists of VA stakeholders who make architectural, acquisition, and management decisions regarding data storage technologies. These include decision makers in the following groups:

- TRM team
- IT centers
- Business Intelligence Service Line (BISL)
- Warehouse governance boards

---

[17] These are briefly addressed in Section 3.2, but not explored in detail.

This document is also intended for those in leadership roles who can establish governance mechanisms, policies, and standards related to data storage selection.

Document Development and Maintenance

This document was developed collaboratively with internal stakeholders from across the Department and included participation from VA OI&T, Product Development (PD), Office of Information Security (OIS), ASD, and Service Delivery and Engineering (SDE). Extensive input and participation was also received from VHA, VBA and the National Cemetery Administration (NCA). In addition, the development effort included engagements with industry experts to review, provide input, and comment on the proposed pattern. This document contains a revision history and revision approval logs to track all changes. Updates are coordinated with the Government lead for this document, which will also facilitate stakeholder coordination and subsequent re-approval depending on the significance of the change.

APPENDIX B. DATA STORAGE TYPES

The subsections below describe different types of data storage/databases, which are organized into three separate categories. The list of categories and types includes:

- Structured (schema-driven)
    - Relational Database
    - Columnar Data Stores
    - Online Analytical Processing
    - Time Series Database
- Unstructured (schema-less)
    - File Systems
    - Document Database
    - Key-Value Store
- Semi-structured
    - File Systems
    - Document Database
    - Object-Oriented Database
    - Graph Database
    - Key-Value Store
    - Wide Column Data Store
- Other
    - Memory Cache
    - Caching Search Engine
    - Archive Data Store

Each category section provides a description of general characteristics, advantages, and disadvantages for most or all data storage types in that category. The data storage type subsections provide descriptions of specific characteristics, advantages, and disadvantages for that type. The data storage type subsections also briefly describe circumstances in which the particular data storage type is a good or poor fit for use.

The use cases in Section 4 provide examples of situations in which an application or workflow may require a combination of different data storage technologies.

Note that the distinctions between data storage types listed in this section are not universal. Some sources consider certain types (e.g., object-oriented and key-value data stores) to be subtypes of one another. While some of the data storage types listed below are similar, this

document – and the VA enterprise – will regard them as distinct types appropriate for different use cases.

Structured (Schema-Driven)

A schema defines the structure of data in a database. Any data written to the database must conform to the rules established in the schema. While strict schema controls limit the flexibility of structured databases, they also provide certain advantages:

- Predictable organization of data facilitates automated reading and writing
- Enforcement of data integrity and a high level of persistence with transactions, even across multiple entities
- Granular security and access control

Structured data stores also tend to have the following disadvantages:

- Only support predefined data values that fit the schema – lack the flexibility to handle other types of data
- Require a specific skillset to develop
- Require administration or stewardship to maintain, which contributes to high operational costs
- Can be difficult to make changes/alterations to the structure of the data
- Often rely on proprietary software which must then be purchased and upgraded

Relational Database

Relational Databases (RDBMS) store data in a highly structured, row/column-oriented, tabular format and are typically used in the warm data temperature range. It is perhaps the best-known and best-understood type of data storage, both within VA and elsewhere, making it the database of choice for much commercial off-the-shelf (COTS) software. RDBMS have all of the advantages and drawbacks typical of structured databases (as described in Section B.1). In addition, they have the following advantages:

- Ubiquity and familiarity, to include the ubiquity and familiarity of SQL
- Concurrency across multiple clients – it is possible for multiple clients to simultaneously access different parts of the same table
- Tabular format easily supports fine-grained security down to individual rows, columns, and data values

An RDBMS is typically used to support transactional[18] applications that process a large number of reads and writes of predetermined, highly structured data. It is a good fit for any use case that involves processing and storing input from forms (as many VA use cases do). When evaluating potential data storage types for a transactional use case, VA data and system owners are advised to consider an RDBMS first.

Columnar Data Store

A Columnar Data Store (CDBMS) is superficially similar to an RDBMS. While the RDBMS stores data in a tabular matrix of rows of individual entities and invariant columns (for entity attributes), the CDBMS stores data in a varying column-oriented format. As a result, CDBMS – unlike RDBMS – do not support complex or high-volume transactions well. Instead, they are optimized for analytics on large volumes of homogenous data, including aggregates and rollup queries. An increasing number of RDBMS COTS vendors now support columnar databases as well.

Because CDBMS do not support transactions well, VA data and system owners will use them primarily for analytics or long-term compressed storage of relational data. They may also use them to maintain records for lookup systems (similar to card catalogs), provided that they do not have to handle frequent writes or edits.

Online Analytical Processing

While RDBMS and CDBMS store data in a two-dimensional tabular format, Online Analytical Processing (OLAP) databases store and analyze data in three or more dimensions. As the name implies, they are intended for analytical use, typically with cool/cold data. They do not handle transactional data. The advantages of OLAP include:

- Support for granular access control at the level of an individual tuple (finite ordered list of elements)
- Allow multiple concurrent reads, though not multiple concurrent writes
- Store pre-calculated, multi-dimensional aggregates of data
- Provide rapid responses to aggregate queries

OLAP databases are more costly to design, develop, and implement than most other types of structured databases, and querying them requires specialized client tools. They are also subject to the same drawbacks as other structured databases.

---

[18] Transactional in the same sense as transactional input – that is to say, supporting periodic live input/updates made by clients or applications.

OLAP will be used for analytics that involve processing tabular data in three or more dimensions - for example, how many Veterans have lived in how many cities over a certain period of time. OLAP is not appropriate for transactional and operational data.

Time Series Database

A Time Series Database (TSDB), as the name implies, is designed specifically for time series data, which other types of structured databases often do not handle well. TSDB are optimized for tracking events or data changes (e.g. price, temperature.) across a period of time at set intervals. TSDBs can also be used to track data, such as memory utilization for performance monitoring or clickstreams on a website for human-computer interaction and interface design studies.

VA data and system owners will use TDSBs for:

- Transactional use cases where time or sequencing is an essential element of either the transactions themselves or properly tracking them
- Long-term, complex analysis of collected streaming data after it has cooled from hot[19] to warm or cold

Unstructured (Schema-Less)

Certain data storage types do not use a schema at all. Rather, their structure is determined (to varying degrees) by the applications or utilities used to read and write their contents. As a result they are much more flexible than structured databases and require little or no administration. This flexibility comes at the expense of certain capabilities that are guaranteed in structured databases – integrity, consistency, concurrency, and granular access control.

File System

File systems (or file-based data stores) contain files that are created and edited by one or more specific applications, which are responsible for the files' internal structure. The entities in file systems may be text documents, logs, messages, multimedia files, or any number of file types. Development of a file system database is not difficult: it is simply a matter of creating and storing files. File system databases also require little or no administration.

This ease of use entails the following drawbacks:

- Data integrity is consistent within a file, but not across files

---

[19] Hot, real-time data analytics use a memory cache (see Section [#]) as a data backend

- Files and versions may differ from file to file. Changes to one file do not automatically cascade/propagate to other files
- Write access to files is limited to an application working on a specific file instance at any particular time
- There is no common query engine or language

In addition, it is possible to set access controls at the level of a file folder, but anything more granular requires compensating controls. A surround engine (available in certain file database products) can provide more granular control – often at the "collection" level – but it is still container-based. Applying granular access control to the content of a particular document in a file database is also prohibitively difficult. Any client with access to a document can read *everything* in that document. It is possible to write redaction tags that will trigger applications to block certain redacted content, but that is very dependent on the application. It is also difficult to apply redaction consistently within or across files.

These characteristics make file systems a good fit for storing and providing access to files, but a poor fit for any use case that requires any of the following capabilities:

1. Support for high-write transaction volumes or multi-client concurrency
2. Consistency and concurrency across multiple entities (in this case, files)
3. Granular security at the level of individual entities or data elements/content within those entities
4. Open, commonly used standards to support access by a variety of applications, clients, or services

VA data and system owners will not use file systems for applications or workflows that depend on the four enumerated capabilities listed above. File systems will of course be used to store application files: some of these files may be linked (as supplements) to structured records (such as person records) in other databases. File system databases will also be used in certain analytic workflows that use semi-structured or unstructured data,[20] such as text analytics, pattern recognition, and machine learning.

Document Database

Document databases are similar to file systems – there may in fact be overlap between the two - but entities in document databases tend to have a more consistent content structure and file format. This is because the documents in document databases contain their own metadata/markup to indicate how content should be read and interpreted. The additional level

---

[20] To include those which use Binary Large Objects (BLOBs).

of consistency mitigates some of the shortfalls of file systems, while providing the flexibility to store unstructured data.

The advantages of document databases are:

- All data relevant to a transaction are contained within the associated document
- Can be developed within the confines of most applications without external support[21]
- Standards for structuring allow other solutions to consume the documents
- Self-contained versioning systems

These characteristics make document databases a good fit,even a necessity, for supporting transactions in asynchronous messaging systems. However, document databases have some disadvantages similar to file systems:

- No enforced relationship across files, making cross-file integrity and consistency difficult to achieve or maintain
- Security is implemented at the collection level, while access to data values/content is implemented at the document level
- No common query language for all document databases, although there are many popular ones
- Concurrency is limited to one client per document
- Data domain control is the responsibility of the application that writes/edits documents, not the database

From a VA enterprise perspective, document databases have many of the same limitations as file systems. The only possible exception is that they may be written and read by a variety of applications, provided they are all configured to use the same metadata, markup, and structuring standards.

Key-Value Store

Key-value databases are perhaps the simplest type of database currently available. They are fast and can quickly and massively scale without significant redesign. Their purpose is to store simple associative arrays of keys and records. They are in some ways analogous to dictionaries: in a dictionary, the word is the key and its definition is the associated record. Unlike relational databases, which rigidly structure the values in records, key-value databases do not require structuring of records at all – only the keys to which they are associated.

---

[21] However, some training may be required to produce adequately structured content for some document types – for instance, those that use Extensible Markup Language (XML).

Key-value databases have the following advantages:

- Very simple implementation
- Support small continuous reads and writes
- No fixed schema – there is no predefined format for records[22]
- Support for multiple user concurrency
- Cache read-only data to provide high-performance reads
- Each key-value pair requires a unique key name, which can interfere with grouping of related name pairs
- Only support "eventual" consistency

The eventual consistency issue is due to the fact that one cannot delete a key-value pair in the database: one can only deprecate the pair and put in a pointer to a replacement pair. This can eventually result in a chain of deprecated pointers referencing one another, which can slow performance.

Typical use cases for key-value databases include dictionaries, storing configuration profiles, embedded system databases, and serving as a temporary scratchpad for ephemeral data. They may also be used for:

- Managing session information for web applications
- High-performance in-process databases
- Enabling range queries and ordered processing of sorted keys
- Signposts to direct clients to other resources, including data sources

VA data and system owners may use key-value stores for any of the purposes described above. Key-value stores will not be used for storing large, complex datasets, or to store input for analytic operations.

Semi-Structured

Semi-structured data stores strike a balance between standardization and flexibility – although they achieve this balance in different ways, optimized for different purposes. Their schemata are often easier to modify or extend than those in structured databases.

Object-Oriented Database

In an Object-Oriented (O-O) database, data takes the form of objects (described by object-oriented programming) rather than, for example, the tabular data structure used in an RDBMS.

---

[22] Although a format can be applied and enforced, if desired.

These databases have all the capabilities of O-O programming languages, such as inheritance, encapsulation, and polymorphism. The databases themselves can be treated as objects by other databases and applications.

Advantages of O-O databases include:

- Structure allows for complex data relationships – more so than in relational databases
- Support for concurrent access to objects by a large number of clients
- Use of class methods supports media applications well, since O-O classes are responsible for correct programmatic interpretation of multimedia objects
- Faster access to data, since objects can be retrieved through direct, dedicated pointers; joins/searches are not needed
- At least some O-O databases support:
  - Systematic triggers and constraints for event-driven architecture (messaging)
  - Automatic object versioning

The drawbacks of O-O databases include:

- Require a specific skillset to develop
- Require administration or stewardship to maintain, which contributes to a high operational cost
- Often rely on proprietary software, which must then be purchased and upgraded
- No universally implemented querying language, although there are some commonly used ones

O-O databases are optimized for use cases in which there is a large amount of concurrent demand for one data item and associated items. VA will use O-O databases to support workflows with this type of use case. This EDP also recommends using O-O databases to support event-driven applications and workflows.

Network databases and key-value stores – described below – are also sometimes considered types of object-oriented databases. From the perspective of this document, however, they are considered to be distinct from object-oriented databases.

Graph Database

Current graph databases, also known as network databases, are a distinct variety of NoSQL databases that commonly express a node-and-arc data schema. Nodes signify real objects and arcs represent relationships between objects. Nodes may have multiple parent-child (or other) relationship types with other nodes.

Advantages of graph databases include:

- Representation of complex and flexible data relationships – more so than in relational or hierarchical databases
- More flexible data processing, since objects can be retrieved via relations; joins/searches are not needed
- Data representation/processing based on open standards
- Flexible data representation and evolution/maturation
- Charting of relationships facilitates data analytics

Drawbacks of graph databases include:

- Require a specific skillset to develop
- Require administration or stewardship to maintain, which contributes to operational cost
- Increasing complexity of data schema can incur increasing computational latency

Graph databases are best employed in use cases where there is a need to represent, track, or leverage complex relationships between entities; whether they are data values, organizations, people, or concepts. Some sample uses are:

- Tracking social networks, to include relationships between individuals and group memberships/affiliations
- Managing access and permissions based on roles, group memberships, and managerial or organizational relationships
- Supporting real-time recommendations and searches of highly related/linked data
- Master data management by tracking relationships between data sources or entities
- Meaningful integration of heterogeneous and distributed data sets/resources

VA will use graph databases for the purposes listed above, particularly to track relationships between people, whether they are internal staff or Veterans and beneficiaries.

Wide Column Data Store

Also called extensible record stores, wide column stores can hold very large numbers of dynamic columns, created for each row rather than predefined by a table structure.[23] They are essentially larger, extensible versions of key-value stores. They do not have fixed column names

---

[23] As in RDBMS.

or record keys, and may have billions of columns. Wide column data stores also support concurrency for multiple users.

The extensibility and flexibility of wide column databases makes them a good fit for reporting with large, non-normalized data sets, including streaming data that has cooled. This flexibility means that no guarantee of consistency or standardization exists in the database records.

Similarly, the lack of standardization in rows and columns within the database makes it difficult to provide granular (as opposed to all-or-nothing) access control.

Wide column data stores will not be used for any VA applications (transactional or analytic) that depend on standardized/normalized data or granular security. They may be used for analytic reporting that does not depend on normalized or completely accurate data (estimates, sentiment analysis, etc.).

Other

The data storage types in this subsection may potentially contain a combination of structured, semi-structured, or unstructured data, so they do not fall into the categories listed above.

Memory Cache

Memory caches, or "in-memory caches", are typically part of a device (such as a sensor or monitor) that feeds streaming data to an analytic system. Like random access memory (RAM), memory caches have a relatively high cost per unit of storage and low durability.

The typical use for a memory cache is in a hot analytic data flow that provides real-time or near real-time alerts based on input from sensors or monitors. These monitors may be anything from IoT devices to medical devices to network intrusion detection systems. The streaming input that monitors provide to the memory cache is analyzed using simple but high-frequency analytic operations. Regardless of the data source, the window of time in which to obtain analytic intelligence – and act on it – is small. The input remains in the cache for only a short time (no more than 24 hours, and often less) before being overwritten.

Within VA, memory caches will only be used for real-time and near real-time analytics, and will not store more than 24 hours' worth of input. Any streaming data to be used for more complex analytics in the warm-to-cold temperature range will be replicated to more durable storage, such as a NoSQL or time series database.

Caching Search Engine

Caching search engines are somewhat different from the other data storage types listed herein. They are metadata repositories built to support search functions. They usually incorporate metadata from a variety of organizational databases. VA system and data owners will use caching search engines to support search functions, especially for large audiences (VA-wide, external, etc.).

Archive Data Store

Archives are the only type of data store that reside in the "frozen" data temperature range. They provide high-durability, low-cost storage for data over long periods of time (years or decades) for data in a variety of formats. Data stored in archives is typically compressed, and must be uncompressed into a data store at a higher temperature range before it can be read or edited. As a matter of best practice, any data security controls present in operational systems are also applied to archival data (e.g., encryption, access controls).

The typical use case for an archive is storing data for extended periods of time for purposes of compliance, contingency measures, or both. Medical records, for example, are stored for compliance purposes.[24] System backups are stored for contingency. Security logs are stored for both compliance and contingency. Once archived, such data will rarely, if ever,be: when it is accessed, it is critical that it be faithfully restored to its original pre-archival state. This means that durability is the most important concern for archival storage technologies.

In addition to ensuring the durability of its archival data stores, and the integrity of their contents, VA will implement archival capabilities that reflect current industry best practices:

- Automated archiving of data based on time elapsed, data temperature/last access-based algorithms, or both[25]
- Transparent access to archival data through operational systems, such that the first byte of data from the archive file can be retrieved in 3-5 seconds

These capabilities will help VA better meet its compliance and availability needs while streamlining data management and reducing costs. They will, however, require keeping archival data in a medium more easily accessible than tape storage.

---

[24] Certain medical records, such as MRI scans, may be stored in archives because they are accessed only occasionally and are too large (therefore expensive) to keep in warm data storage.
[25] Automatic archiving rules and practices, whether they are time-based or temperature-based, will be consistent with VA's records management and archiving policies.

APPENDIX C. DEFINITIONS

This appendix provides definitions for terms used in this document, particularly those related to databases, database management, and data integration.

Key Term	Definition
Data Flow	Describes the lifecycle and movement of data in an analytic system with respect to a particular process or use case. A data flow begins with collection/ingestion from data sources and ends with the presentation of information extracted from the data using alerts, reports, visualization tools, applications, etc.
Client	In the context of this document, a client is a user, application, process, or some combination thereof that accesses and processes data.
Enterprise Creative, Read, Update, Delete (eCRUD)	The eCRUD service was initially created as part of the Veteran Lifetime Electronic Record (VLER) Data Access Service (DAS) project. eCRUD provides an interface that allows enterprise services to perform create, read, update or delete (CRUD) operations on data in the VA SOA data access layer/HDA solution. It also supports numerous adapters for data transformation, notification of data changes, and custom event handlers.
Ingest, Ingesting, or Ingestion	The entry of data into a process or data platform from an original data source, such as an application, operational data store, sensor, manual input, etc. There are essentially three types of data ingest: • Stream (sensor data, mobile) • Transactional (inidivual database reads/writes) • Files (batch transfers, logs, objects)
Not Only SQL (NoSQL)	Type of DBMS that structures data in a non-tabular/non-relational format. NoSQL database types include key-value, column-family, document, and network. Some NoSQL database can also store unstructured data.

APPENDIX D. ACRONYMS

The following table provides a list of acronyms that are applicable to and used within this document.

Acronym	Description
ABAC	Attribute-Based Access Control
ADS	Authoritative Data Source
AN	Analytics and Informatics
ASD	Architecture, Strategy and Design
BI	Business Intelligence
BIRLS	Beneficiary Identification Records Locator System
BLOB	Binary Large Object
CDI	Customer Data Integration
CDSS	Clinical Decision Support System
CDW	Corporate Data Warehouse
DaaS	Data as a Service
DAR	Data Architecture Repository
DAS	Data Access Service
DCPA	Disability Claims Processing Application
DEERS	Defense Enrollment Eligibility Reporting System
DGC	Data Governance Council
DoD	Department of Defense
EA	Enterprise Architecture
eHMP	Electronic Health Management Platform
EHR	Electronic Health Record
eMI	Enterprise Messaging Infrastructure
EPMO	Enterprise Program Management Office
ERP	Enterprise Resource Planning
ESS	Enterprise Shared Services
ETSP	Enterprise Technology Strategic Plan
FR	Field Reporting
GP	General Purpose
HHS	Department of Health and Human Services
HSRD	Health Services Research and Development
IAM	Identity and Access Management
IoT	Internet of Things
IPT	Integrated Project Team
IT	Information Technology
LOB	Line of Business
MVI	Master Veteran Index
NCA	National Cemetery Administration

Acronym	Description
NLP	Natural Language Processing
NoSQL	Not Only SQL
OGC	Office of General Counsel
OI&T	Office of Information and Technology
OIS	Office of Information Security
OLAP	Online Analytical Processing
O-O	Object-Oriented
PHI	Protected Health Information
PID	Person Identifier
PII	Personally Identifiable Information
POC	Point of Contact
RD	Health Services R&D
RDW	Regional Data Warehouse
SOA	Service-Oriented Architecture
SQL	Structured Query Language
SSA	Social Security Administration
TRM	Technical Reference Model
TSDB	Time Series Database
VADI	VA Data Inventory
VBA	Veteran Benefits Administration
VHA	Veteran Health Administration
VLER	Veteran Lifetime Electronic Record
XML	Extensible Markup Language

APPENDIX E.   REFERENCES, STANDARDS, AND POLICIES

This EDP is aligned to the following VA OI&T references and standards applicable to all new applications being developed in the VA, and are aligned to the VA Enterprise Technical Architecture (ETA):

#	Issuing Agency	Policy, Directive, or Procedure	Purpose
1	VA	VA Directive 6551 http://www.techstrategies.oit.va.gov /docs/designpatterns/ 6551dir16.pdf	Establishes a mandatory policy for establishing and utilizing Enterprise Design Patterns by all Department of Veterans Affairs (VA) projects developing information technology (IT) systems in accordance with the VA's Office of Information and Technology (OI&T) integrated development and release management process, the Veteran-focused Integration Process (VIP).
2	VA OIS	VA 6500 Handbook	Directive from the OI&T OIS for establishment of an information security program in VA, which applies to all applications that leverage ESS.
3		VA Data Inventory (VADI) http://vaausdarapp41 /ee/request/h ome	VADI is the authoritative source for VA Data Store metadata. VADI allows users to search for and navigate through VA Data Store metadata.
4		Data Architecture Repository (DAR) http://enterprise.metadata.va.gov/pl s/apex/f?p=DAR:1:50 1780167519508	DAR provides a means to catalog, search, report, and manage VA metadata via a web-accessible portal.