BRING THE FHIR® INSIDE

Written by **Jeff Needham** • **MongoDB**

Data, Digital Transformation, and Why You Don't Need to Rip and Replace Legacy Systems

November 2021

This white paper was contributed by MongoDB and does not represent the views of HL7 International.

Achieving the goal of easily and securely exchanging patient data within and between different healthcare organizations, and with patients, is a continuing challenge for the healthcare industry.

Providers and payers adopting closed electronic health record (EHR) systems, data held in outdated legacy databases, and multiple incompatible data standards are just some of the issues preventing the free and secure flow of data.

But with HL7® Fast Healthcare Interoperability Resources (FHIR) cemented as the industry's new data standard, all that is about to change.

FHIR, a common schema and a set of API standards for exchanging healthcare data, performs a simple sounding but fiendishly complicated task: providing the highest level of data interoperability while also reducing the challenges that organizations face when trying to share and consume healthcare data.

In the United States, the steady, organic adoption of FHIR among healthcare organizations and technology vendors has accelerated due to growing regulatory pressure. Starting in 2020, the Centers for Medicare and Medicaid Services (CMS) set a rolling deadline for FHIR compliance, with fines for institutions that fall behind. As a result, for most U.S.-based healthcare providers, payers, and their technology vendors, the past few years have been a headlong race to adopt FHIR.

But with healthcare organizations either FHIR compliant or on the way to becoming so, what comes next? Is this the end of the interoperability journey?

ADOPTING HL7 FHIR IS JUST THE BEGINNING

FHIR is more than a check-box compliance event for the healthcare industry; adopting FHIR marks the beginning of a digital transformation journey for healthcare organizations.

FHIR was created with the intent of reducing the burden and friction associated with sharing and exchanging healthcare data across disparate parties and systems, schemas, and formats. In doing so, FHIR helps organizations bring compelling, insightful, and modern applications and experiences to patients, helping them live healthier lives.

Imagine applying that same advantage — the easy exchange of data — to your own organization's systems. Could you serve insights and alerts to patients in real time, triggered by a pharmacy claim or patient's connected device? Could you also augment those events with even deeper analytics insights and machine learning to make personalized wellness program recommendations? Or is this kind of insightful data locked away behind complex and rigid legacy systems or data warehouses, where accessing data in real time would simply cost too much money and time?

In addition to patient experiences, could you also help your organization's employees increase the quality of care by augmenting and automating rudimentary tasks such as data entry, capturing patient vitals in electronic systems and scheduling appointments?

THE LEGACY CHALLENGE

Improving business workflows and patient experiences hinges on the easy exchange of and real-time access to relevant data. Consider your own healthcare organization. How easy is it to:

- Access all of the data required to transform a business process?
- Extract data from legacy technology, particularly legacy relational databases?
- Combine different data formats to create meaningful and actionable insights and streamline new business processes?

The ability to execute on a digital transformation plan succeeds or fails depending on how you answer these questions. If the answer to all three is “not very easy,” then your organization, like many others, faces steep hurdles to digitally transform. You’re stuck, battling against a pervasive opposing force or “digital friction.”

To innovate and give patients the modern healthcare experiences they expect, healthcare organizations must first free themselves from the rigid data architectures associated with legacy hardware and monolithic patient record and care applications.

Even modern healthcare technologies rely on traditional data architectures, like relational database management systems (RDBMS), making change harder than it needs to be. They slow the rate of innovation, entrench a fear of failure, and complicate business requirements that didn’t exist when RDBMSes were invented, such as data privacy.

BRING THE FHIR INSIDE

Despite the challenge posed by legacy technologies, when healthcare organizations do decide to modernize their internal business systems and processes, inevitably a “buy vs. build” discourse emerges. The choice appears binary: Should the legacy systems in question be rewritten with in-house developers and custom software, or replaced entirely with one-size-fits-all commercial off- the-shelf (COTS) software?

This is a false dichotomy. The challenge is not in how to replace legacy systems and applications. As with interoperability and the adoption of FHIR, the drive to modernize healthcare’s internal digital systems is about accessing data in a timely manner and translating it into a single consistent representation, so that business systems, internal users, and ultimately patients can make sense of it.

FHIR offers your organization a head start, with a canonical schema that can be used to accelerate your internal modernization and transformation efforts. However, it may be possible that your systems are not FHIR compatible, or they may not conform to a shared data standard or format.

MongoDB offers a cost effective and incremental approach to modernizing legacy healthcare systems to take advantage of FHIR, one that doesn't require the wholesale "rip and replace" of legacy technology, or the upheaval and huge expense associated with purchasing COTS software alternatives.

APPROACHES TO MODERNIZATION PLANNING

There are three ways to modernize your applications to leverage FHIR:

- Data-driven
- Application-driven
- An iterative data and application approach

DATA-DRIVEN MODERNIZATION

This approach begins by moving data from the legacy system to the new environment before any FHIR APIs or microservices are provisioned to interact with that data. Even in its earliest phases, data-driven modernization is a big step forward over legacy systems because once you've moved your first data source into the new environment, you can leverage it immediately and start building modern applications on top of it.

Applications can write directly to the new environment without affecting the existing one. Once more writes are executed in the new environment than the old one, you can begin to dramatically reduce the footprint of the legacy system. By the time the last phases of data-driven modernization are implemented — when the new environment takes over the majority of the work and becomes the system of record — you can begin to retire legacy applications entirely. Data-driven modernization is often a preferred approach for making legacy data interoperable with FHIR, as it offers a comprehensive approach for facilitating access to an entire domain of data (provider, patient, diagnosis, etc.).

While there isn't a clear technical downside to this approach, it is often more "modernization" and not digitalization, as existing data structures are often carried over unchanged and therefore continue to inhibit new workflows and experiences. Therefore, some organizations may prefer an iterative approach, outlined below.

APPLICATION-DRIVEN MODERNIZATION

With application-driven modernization, all reads and writes from new applications and microservices are executed in the new data environment from the start. Existing traffic continues to route to the existing data store. The legacy system continues to operate unchanged. This enables new functionality to be introduced immediately, but it also introduces more complexity. Because application-driven modernization is an all-or-nothing approach, healthcare organizations must have a clear strategy for retiring the legacy applications in due course.

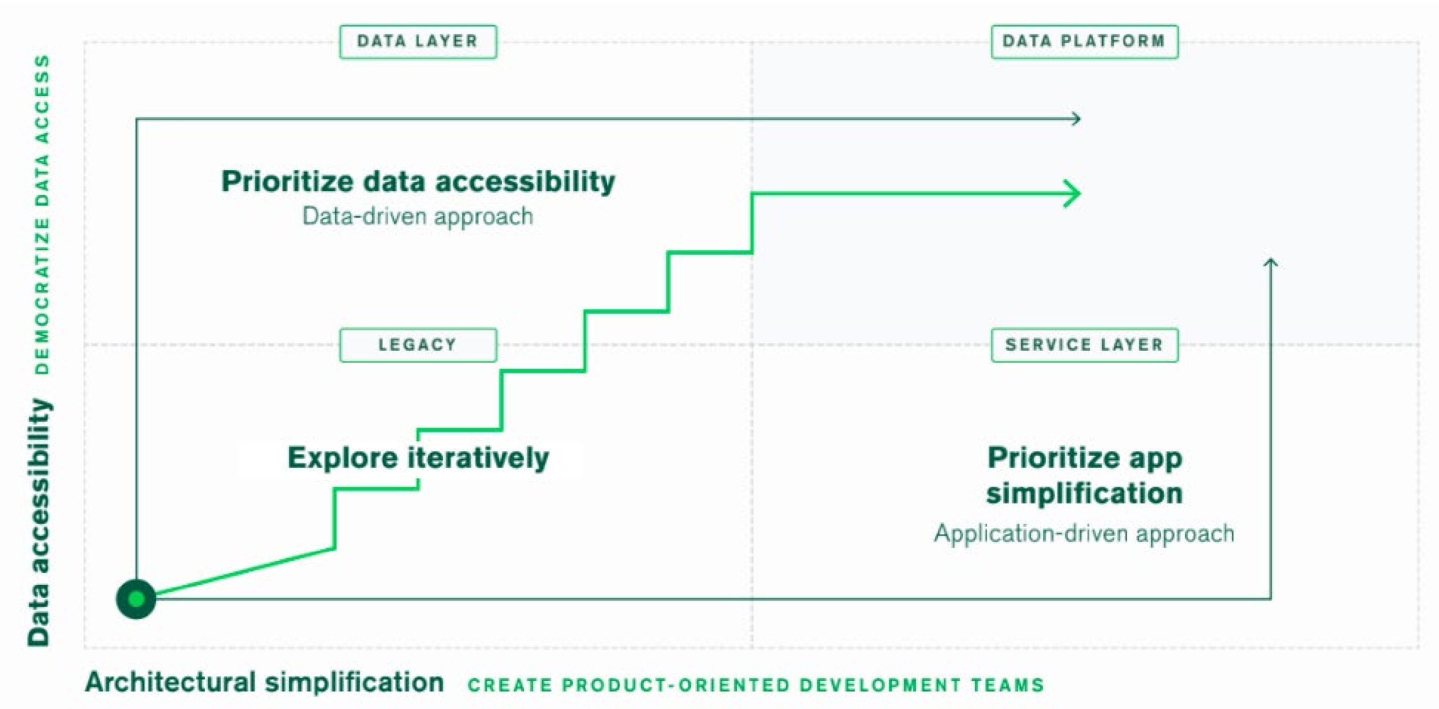
A caveat on this approach is that it works best in cases where an application in its entirety must interact with FHIR and is not an application that will likely be refactored in the near future.

ITERATIVE MODERNIZATION

Iterative modernization enables organizations to innovate while modernizing. This approach — the one MongoDB recommends — blends data- and application-driven approaches for incremental FHIR interoperability, starting with the least complex applications and objects and slowly progressing to more complex ones.

This one-step-at-a-time approach gives you the best of both worlds: You see immediate gains along the way but are not committing to a newly refactored environment right away. This minimizes risk while preserving data from the legacy systems.

THREE APPROACHES TO MODERNIZATION


DATA LAYER
DATA PLATFORM
Prioritize data accessibility
Data-driven approach
LEGACY
SERVICE LAYER
Explore iteratively
Prioritize app simplification
Application-driven approach
Data accessibility
DEMOCRATIZE DATA ACCESS
Architectural simplification
CREATE PRODUCT-ORIENTED DEVELOPMENT TEAMS

Legacy modernization is frequently perceived to be time-consuming, complex, and error-prone, but the reality is that it can be straightforward, predictable, and successful, allowing healthcare organizations to accelerate their digital transformation, deliver truly modern patient experiences, and support compliance with increasingly restrictive data-privacy regulations, all while minimizing risk.

MongoDB's iterative approach to modernization utilizes an Operational Data Layer (ODL), allowing healthcare organizations to incrementally detach from legacy systems, balancing performance and risk.

WHAT IS AN OPERATIONAL DATA LAYER?

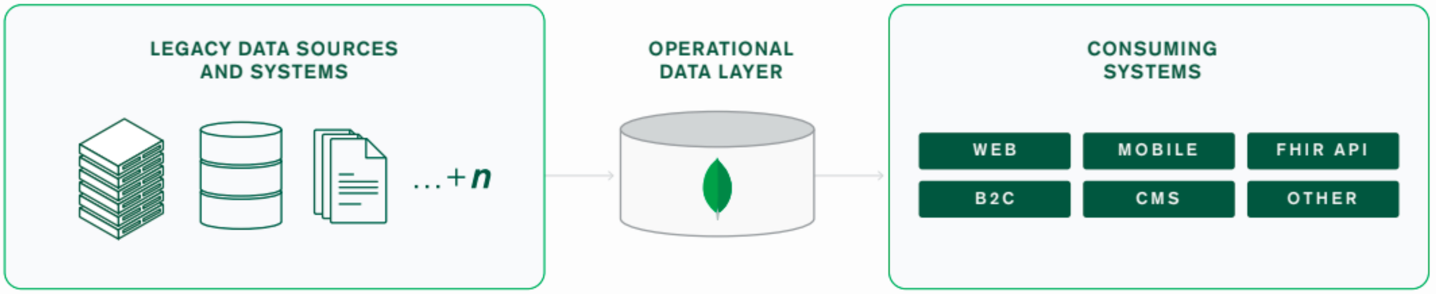
The ODL is the on-ramp for data that's being routed to the new environment. It performs the following functions:

- Centrally integrates and organizes siloed enterprise data
- Makes real-time data available to consuming FHIR-based APIs and applications
- Enables legacy modernization and data as a service
- Creates a single source of truth using the FHIR standard
- Enables real-time analytics and mainframe offload
- Allows for gradual refactoring (vs. rip and replace)
- Minimizes disruption when deploying to the cloud
- Serves legacy data to new applications without straining the legacy system
- Makes data immediately available for analysis and business intelligence
- Allows easy extensibility of data structures to support new use cases and specific requirements (supported by both MongoDB and FHIR)

Gradually, more reads and writes are routed to the new environment as the legacy systems are retired one step at a time.

By the final phase, all applications are provisioned in the new environment.

THE OPERATIONAL DATA LAYER (ODL)



MOVING DATA BETWEEN SYSTEMS

In order for your organization to take full advantage of the FHIR standard, you'll need to be able to interact with legacy system data that is not FHIR-schema compliant, and do so in real or near- real time.

While on-the-fly queries against legacy systems can be issued, the data still needs to be transformed to and from the FHIR schema in order to be compliant. It's often more practical to move a copy of that data into an ODL and transform it to be FHIR compatible so that real-time calls can be made with as little latency as possible.

Before moving any data from the legacy RDBMS to the ODL, you'll need to build temporary scaffolding to transition from the legacy system to the new environment.

The first part of the scaffolding uses connection services between the legacy systems and the new environment. Connection services are needed for three different types of data sources:

1. **Streaming interfaces:** real-time data, generally measured in seconds, milliseconds, or microseconds, that will be replicated between the legacy systems and the new environment simultaneously
2. **Service interfaces:** end-of-day and batch-processing actions that are common in older environments
3. **Specialty connectors:** connect to specific workloads, like Hadoop or Spark

Once you've established which connection service you need for each data source, you can begin building the intermediary layer that will bridge the RDBMS and other legacy systems to the new data architecture. With the modernize-while-innovating approach, the connection between legacy systems and the new architecture is the ODL.

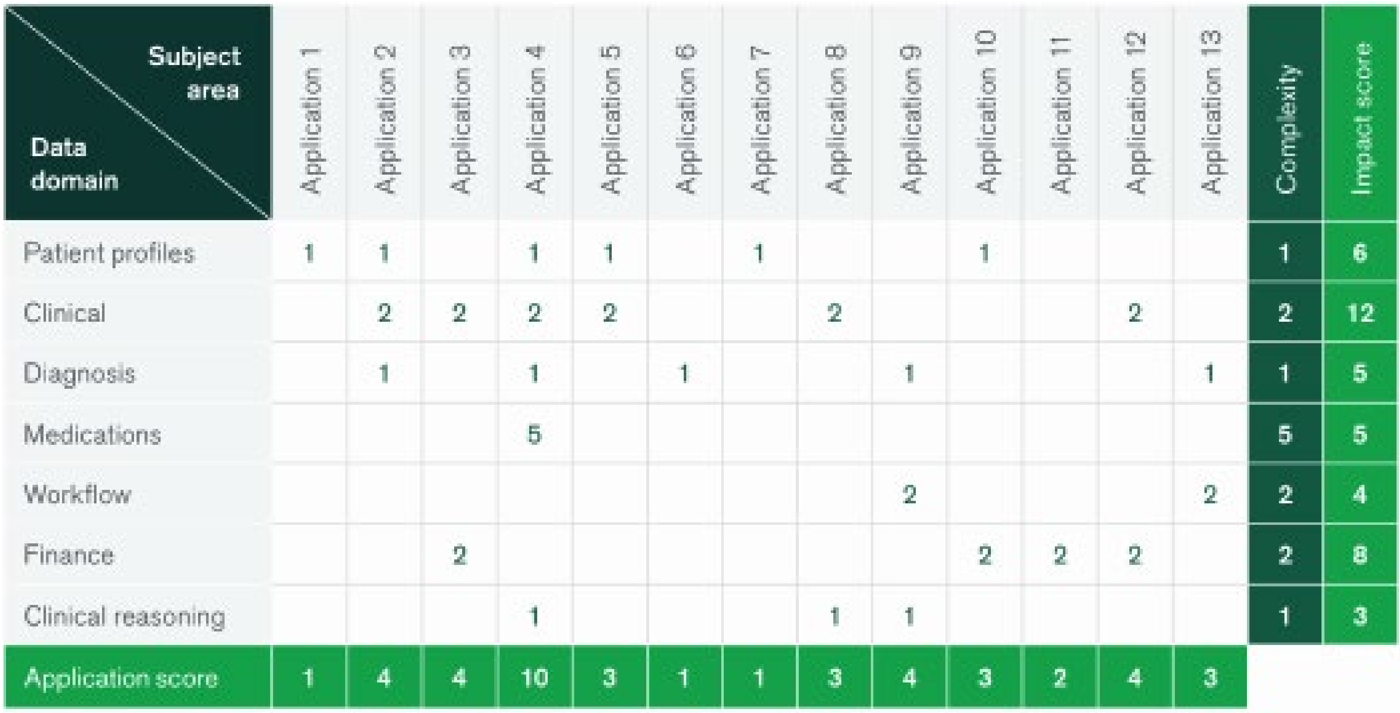
MODERNIZING WHILE TRANSFORMING WITH FHIR

MongoDB's iterative approach begins by identifying all objects in the application code and any applications that connect to them.

Each of these objects constitutes a data domain, which is a collection of values contained in an element. For instance, "patient profiles" is a data domain that includes details about patients expressed as values, such as how long they've been a patient, their medical history, or the type of medication they take.

Once you've identified the objects you're using, you can assign a complexity score to each object based on their properties, methods, collections, and other attributes. You can then identify each application that connects to a domain and rank them based on variables such as how mission-critical it is, how many patients or internal users rely on it, how many tasks it has to perform, and how complex those tasks are.

RANKING DATA DOMAINS AND APPLICATIONS



Subject area Data domain	Application 1	Application 2	Application 3	Application 4	Application 5	Application 6	Application 7	Application 8	Application 9	Application 10	Application 11	Application 12	Application 13	Complexity	Impact score
Patient profiles	1	1		1	1		1			1				1	6
Clinical		2	2	2	2			2				2		2	12
Diagnosis		1		1		1			1				1	1	5
Medications				5										5	5
Workflow									2				2	2	4
Finance			2							2	2	2		2	8
Clinical reasoning				1				1	1					1	3
Application score	1	4	4	10	3	1	1	3	4	3	2	4	3		

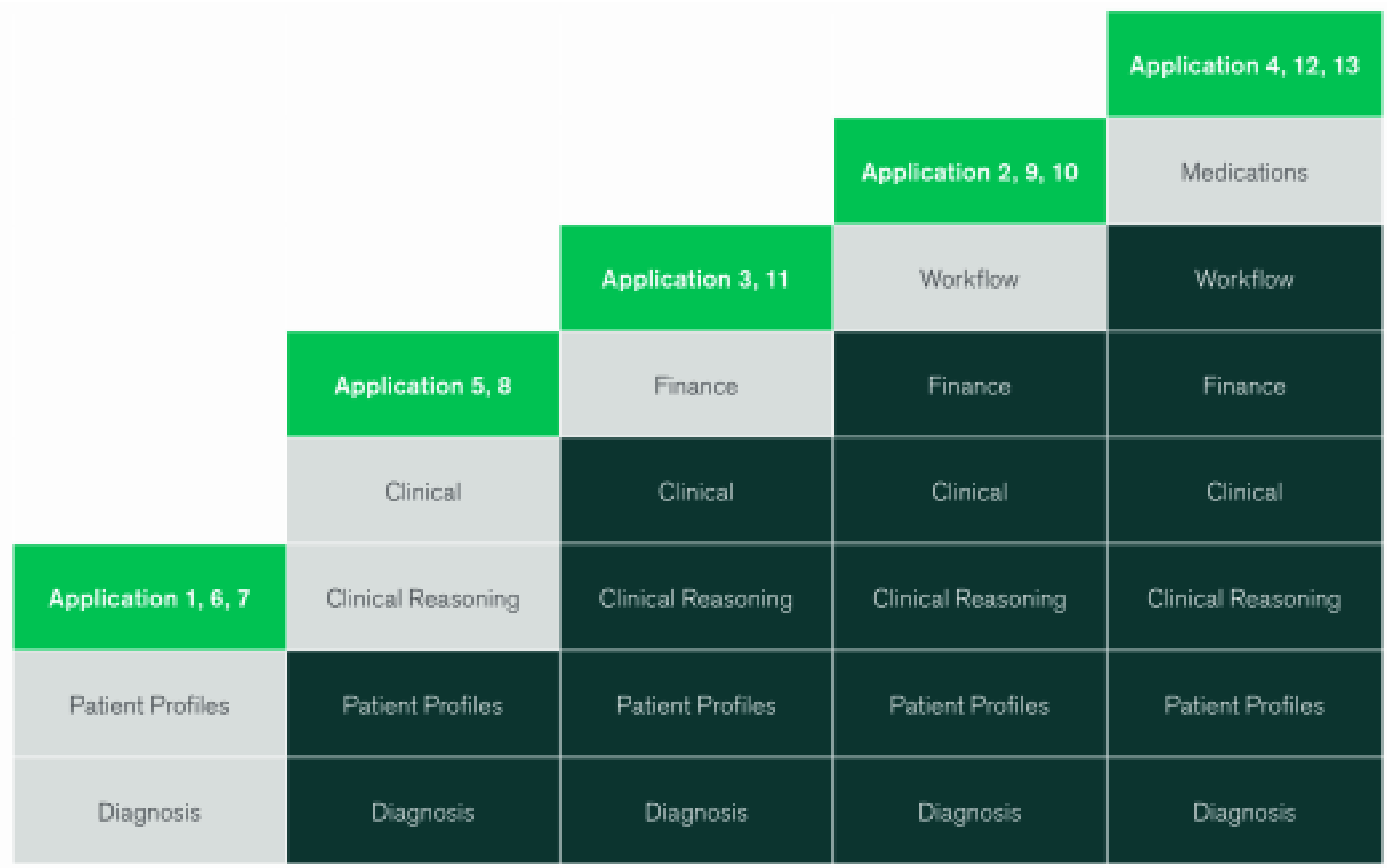
By ranking the data domains and applications by complexity, you can create a plan for moving each domain from the legacy system to the new architecture and rerouting applications to connect to the new domains, starting with the least complex data sources and gradually progressing to more complex ones.

In the graphic on the previous page, the data domains “patient profiles” and “diagnosis” each have a complexity score of 1 and are used by applications 1, 6, and 7, each with a complexity score of 1. These are perfect candidates to become the first sources of data migrated to the new architecture and the first applications refactored to connect to the new data domains.

Once you have a clear picture of all the objects and applications and have scored them based on the number of dependencies and their complexities, you’ll end up with a graph that shows the potential sequence and timing for moving objects and applications into the new data architecture.

This will be the basis for your five-phase iterative modernization plan.

POTENTIAL SEQUENCE FOR MODERNIZATION



				Application 4, 12, 13
			Application 2, 9, 10	Medications
		Application 3, 11	Workflow	Workflow
	Application 5, 8	Finance	Finance	Finance
	Clinical	Clinical	Clinical	Clinical
Application 1, 6, 7	Clinical Reasoning	Clinical Reasoning	Clinical Reasoning	Clinical Reasoning
Patient Profiles	Patient Profiles	Patient Profiles	Patient Profiles	Patient Profiles
Diagnosis	Diagnosis	Diagnosis	Diagnosis	Diagnosis

HEALTHCARE MODERNIZATION IN FIVE PHASES

With an ODL, scaffolding, and a map of your data domains and applications, you're ready to start the five incremental phases of modernization while still protecting existing patient assets and business-critical operations.

PHASE 1: Simple ODL

In the first phase of legacy migration, reads from the legacy mainframe are offloaded to the ODL. This reduces read traffic to the mainframe. The ODL provides high availability, improves performance, and handles long-running analytics queries. The ODL is interpreted directly by the application. It has a modern interface, and you can start building modern applications on the ODL.

PHASE 2: Enriched ODL

In the second phase, the ODL acts as an integration layer enriched with multiple data sources and metadata. At this stage, you can begin building microservices on top of your data. The ODL also serves as an operational intelligence platform for insights and analysis. The ODL offloads more reads from the source systems and enables more use cases than were previously possible, including a single customer view.

PHASE 3: Parallel write

In the third phase, reads and writes are performed concurrently on the source system and the ODL, either directly from application logic or through a messaging system, API layer, or other intermediary. This is also known as Y-loading or Y-storing. This phase lays the foundation for a more transformational shift of the ODL's role in the system architecture. In this phase, you can test the ODL to ensure functionality before using it as the primary system for writes.

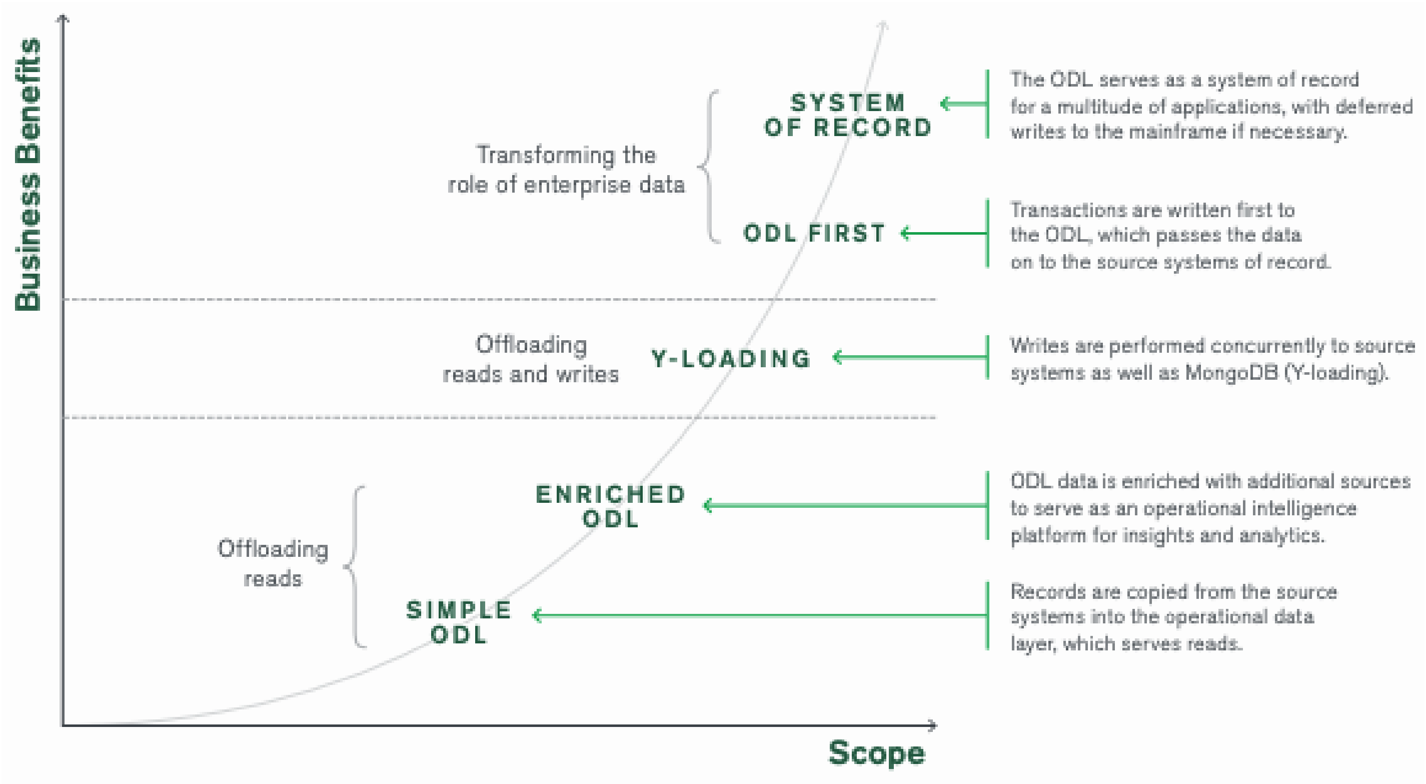
PHASE 4: System of transaction

In the fourth phase, transactions are written first to the ODL, then passed on to the legacy system if necessary. At this point, the ODL is the single source of truth. The secondary write to the legacy source can be accomplished with a change data capture system listening to the ODL or a similar system, such as MongoDB Realm Triggers.

PHASE 5: System of record

In the fifth phase, the ODL becomes the system of record for all consuming applications. The source system(s) can be decommissioned for cost savings and architectural simplicity.

THE FIVE PHASES OF HEALTHCARE MODERNIZATION



The ODL serves as a system of record for a multitude of applications, with deferred writes to the mainframe if necessary.

Transactions are written first to the ODL, which passes the data on to the source systems of record.

Writes are performed concurrently to source systems as well as MongoDB (Y-loading).

ODL data is enriched with additional sources to serve as an operational intelligence platform for insights and analytics.

Records are copied from the source systems into the operational data layer, which serves reads.

YOUR ROAD MAP TO DIGITAL TRANSFORMATION

For years, healthcare organizations have wrestled with the questions of whether, and how, to modernize their legacy mainframe systems. With the emergence of digital engagement strategies that require connected care, real-time transactions, analytics, and agile product development, legacy modernization has become a healthcare imperative.

The key to legacy modernization is creating a bridge between legacy systems and the new architecture, the ODL. This approach enables healthcare organizations to offload traffic away from costly legacy systems and, eventually, to rearchitect monolithic applications into a suite of microservices, while also exploiting FHIR data standards and utilizing FHIR compliance projects — not just for check-box compliance, but as a strategic starting point for a more modern data infrastructure.

Crucially, by deploying the ODL in phases, healthcare organizations can embark on their digital transformation journey iteratively, without the risk of an all-or-nothing, rip-and-replace approach.

This white paper was contributed by MongoDB and does not represent the views of HL7 International.

ABOUT THE AUTHOR

Jeff Needham is principal of Industry Solutions for Healthcare at MongoDB and brings more than 20 years of IT experience in large and small organizations. Prior to joining MongoDB, Jeff drove legacy transformation and legacy modernization efforts at Travelers Insurance as senior director of architecture and architected healthcare solutions for Aetna/CVS Health.