A Make-or-Break Decision:

Choosing the Right Serialization Technology for Revenue, Operational, and Brand Success

Introduction

The pharmaceutical industry is on the verge of a critical turning point: as serialization introduces massive amounts of data into the supply chain, the industry must abandon traditional technologies that were simply not designed for a challenge of these dimensions. Companies who continue to rely on traditional solutions – most notably, relational database management systems (RDBMS) - will unequivocally fail, jeopardizing their ability to move product through the supply chain, their revenue, and their brand reputation.

With serialization, data volumes will increase by multiple orders of magnitude – 100 to 100,000 times larger – posing an "Internet scale" problem. Solutions based on traditional application architectures, for enterprise-class problems involving many fewer transactions, will not be able to scale to meet the demands. The combination of a massive database, high transaction volume, and the time required for operational processes demands a paradigm shift in technology.

In this whitepaper, we will review the depth of the challenge that serialization introduces; where relational databases can contribute, and where they cannot; and how a move to NoSQL – a database technology relied on by other industries to solve internet class problems - in a native cloud architecture is ultimately what's required to maximize performance and enable the pharma industry to successfully meet the serialization challenge.

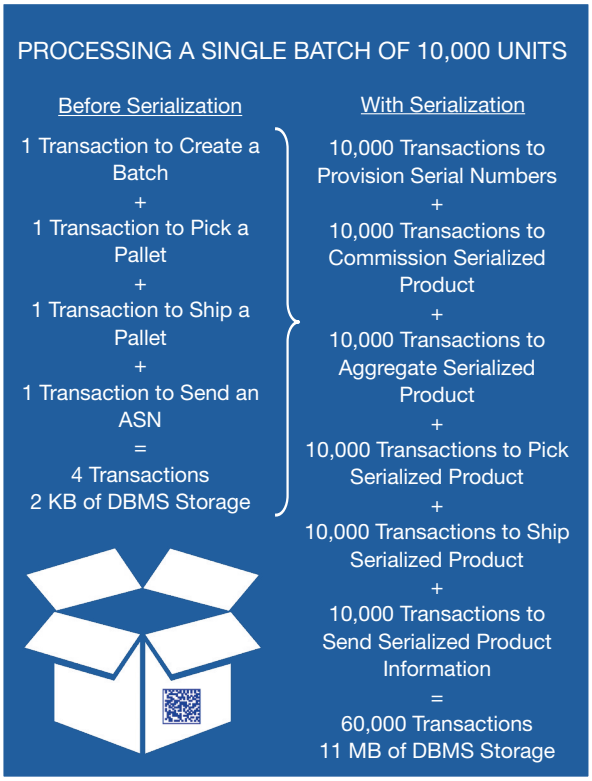
Table of Contents

The Serialization Challenge: From Batches to Unique Identifiers

By early 2019, track and trace laws will govern more than 75% of the world's prescription drug supply. At the heart of track and trace is serialization: the application of a unique identifier to each individual saleable unit. The process of serializing prescription drug products and tracking each event that happens to them throughout the supply chain will introduce a tremendous volume of new data.

Today, the pharma industry is accustomed to working on batches, as they do in ERP operations or lot level processing. Shipping a batch of 10,000 units requires that you process four simple transactions. The batch must be created; the pallet must be picked, and then shipped; and an advanced shipping notice (ASN) must be sent.

Once serialization comes into play, with the need to manage the unique identity of every single product, the shipment of that same batch will require the processing of at least 60,000 database transactions in order to update the data that corresponds to every single unit. Provisioning serial numbers will require 10,000 transactions, and commissioning the batch will be 10,000 more. And new tasks that didn't even exist before, such as aggregation or shipment notification, introduce thousands of new transactions. The bottom line is that the four transactions you complete today to ship 10,000 units will be – at minimum – 60,000 transactions in a serialized world. And as the volume of transactions expand, so do the information processing requirements: they will be 15,000 times more, and the size of the database will be 5,500 times larger, for the same volume of products you ship today.



Amidst all this growth, your manufacturing and warehouse processes march on. You still need to produce the same volume of product and get it out the door at the same speed in order to maintain operational efficiencies. Implementing a serialization solution that can manage these massive amounts of data and transactions is essential to protecting those efficiencies.

Relational Databases: Where They Thrive, And Where They Fail

Relational database management systems are data stores that organize records by their type, and understand and maintain the relationships between those records. Introduced in the late 1970s, they are a staple tool in many businesses arsenals. Relational databases are very good at solving enterprise-class problems, particularly ones that involve reporting and analytics.

As data grows to extreme sizes, it's very easy to outstrip the capacity of a relational database, and performance degrades.

Maintaining performance in a relational database centers on keeping the number of rows at a reasonable level, sized according to the hardware that you have available. Typically, all the processing of a relational database goes through a single machine. As data grows to extreme sizes, it's very easy to outstrip the capacity of a database, and performance begins to degrade. Database administrators are forced to invest a lot of time and effort managing the number of rows in their database to try to maintain performance. If they don't, they run the risk that performance degrades so far that the database stops working altogether. Their only other option in these cases is to scale up, acquiring bigger hardware and more powerful machines.

How Serialization and EPCIS Factor In

A key aspect of the serialization challenge is not just the pure volume of serial numbers companies will generate, but the events that need to be associated with each number and then referenced each time a serial number is in play. For instance, if you want to decommission a serial number, your solution must locate not only the number but everything that has happened to it thus far - commissioning, aggregation, disaggregation from a container – so it can understand that serial number's current state in order to change it again. Retrieving serial number event information requires accessing multiple indexes and tables, which increases the number of transactions to 100,000 times more than processing non-serialized events.

The Electronic Product Code Information System, or EPCIS, is a management system to track all of the events – the who, what, where, when, and why – associated with serialization. Traditionally, EPCIS repositories have been implemented in relational database structures. When EPCIS is built on RDBMS, though, the relational database cannot scale to perform the constant look-ups that are associated with tracking the events for each serial number activity.

Data Archiving: What You Need to Know

Because the speed of RDBMS queries slows down as the amount of information grows, solution vendors that rely on relational databases will require that you regularly purge and archive serial number data to keep the database at a manageable size. Archiving means that you store large volumes of data in an offline system, and only have a subset of your data available to you. That subset could be defined by time, or it could also be defined by size and have variable time. Either way, you do not have access to all the data you may need on hand and readily available.

Managing the archiving process is challenging for any business but for pharma companies managing serialization requirements, returns, and verification inquiries, archiving is a non-starter. According to McKesson, the largest US wholesale distributor, more than 225,000 units per day are involved in saleable returns across the US pharma supply chain. Each time one of those occurs, returns processors will need to verify product serial numbers with you. In addition, under DSCSA, wholesale distributors expect verification to be processed in real-time, and they will process millions of returns in a year. In either scenario, if you are only keeping the last six months of data in your serial number repository and you get a request involving data from a year or two ago, you now must locate that data in an offline archive and return it to the database, requiring you to build exception processing into your business processes.

For pharma companies managing serialization requirements, returns, and verification inquiries, archiving is a non-starter.

Operational Impact

Ultimately, your relational database has to keep up with your transaction load. If transactions are coming in faster than the serialization solution can handle, you will jeopardize your business processes. If, for instance, shipment transactions are sent to your solution at the end of the shift when those shipments were prepared, an RDBMS solution may take hours to process them. That delays when shipments can leave the facility and – if you can't book revenue until that shipment leaves – it has a financial impact to your business. There is also a potential customer satisfaction issue if shipments are consistently late. You might not be able to ship product because you haven't generated the appropriate documentation or downstream feeds that you have to send to your partners. This all assumes that the RDBMS solution will finish processing shipments at all – the greater risk is that it ultimately just does not work. In these types of architectures, the risk of "timing out" is greater.

If all you have is a relational database to solve the problems that are placed in front of you, it's as if all you have in your toolbox is a hammer, and the entire world looks like a nail.

If It Doesn't Work, Why Would Anyone Use It?

Serialization represents a daunting new challenge to the pharma industry, and to the solution providers who serve it. If relational databases aren't equipped to meet the challenges of serial number storage and processing, why would any vendor base their solution on one, and why would any pharma company buy it?

The answer is simple and to some degree, it's also human nature: it's what they use today, and what they know. If all you have is a relational database to solve the problems that are placed in front of you, it's as if all you have in your toolbox is a hammer, and the entire world looks like a nail.

Solving Internet Scale Problems With NoSQL

TraceLink began exploring alternate technologies more than five years ago, as soon as we realized that relational databases would not scale to meet serial number storage and processing needs. It quickly became clear that NoSQL was the preferred approach.

NoSQL is a non-relational database widely recognized for its high scalability and performance that remains near-constant, regardless of whether you have a relatively empty data store or one with hundreds of billions to trillions of rows and records. It is used by Facebook, Twitter, Google and other companies that manage hundreds of millions to hundreds of billions of transactions daily to address internet scale problems.

The reason that NoSQL databases can scale is that they can be highly distributed. Rather than relying on a single large machine, a highly distributed database splits the records across many separate machines, with each one managing a subset of the data. This follows a basic maxim of internet technologies: scale out by adding more machines, rather than scaling up by using bigger machines. For example, a Google data center runs hundreds of thousands or millions of relatively small machines, rather than a few big mainframes. The power of internet technologies is that it provides a single view of the data even though that data is physically distributed across many different machines.

NoSQL is used by Facebook, Twitter, Google and other companies that manage hundreds of billions of transactions daily to address internet scale problems.

Optimizing for Operational Processing and Performance

TraceLink has architected the Life Sciences Cloud to use a NoSQL database - in combination with a distributed data model and parallel processing with elastic computing in the Amazon Web Services (AWS) cloud - to achieve virtually limitless scale and continuous data availability. Because of NoSQL's ability to scale, TraceLink has never had to purge data out of our serial number repository: it contains serial numbers that were commissioned last week, and serial numbers that were commissioned four years ago. All serial number data, regardless of age, is available to query in real time.

This combination of tools makes TraceLink's approach unique and delivers unrivaled results. Our solution performs consistently, irrespective of the size of the serial number repository: we have near constant time regardless of whether we're looking at a company with just a couple serial numbers or a company with hundreds of millions. The time it takes to fetch, update, or operate on those serial numbers is basically the same. We have achieved real world performance of 150,000 serial number reads per second and 10,000 writes per second.

Many Machines Makes Light Work

TraceLink utilizes both Amazon Web Services' elastic computing fabric as well as distributed processing to further maximize performance. Through AWS, TraceLink can provision machines on demand to meet peak loads, and then dynamically shrink down. This allows us to add whatever processing power in real-time as needed. In addition, we distribute work across up to thousands of multiple machines, chunking it up into batches to create many smaller tasks and achieve high throughput. This enables a substantial request, such as shipping a large number of products, to be broken into smaller workloads so that many computers can achieve it in parallel.

TraceLink has achieved real world performance of 150,000 serial number reads and 10,000 writes per second.

The Only Proven Solution

By combining the strengths of NoSQL, distributed data, elastic computing, parallel processing, and other technologies in the Amazon Web Services cloud environment, TraceLink has built the only solution for pharma serialization that has already proven it can scale, and perform at a pace to keep companies' operational processes moving forward without impediment. The following milestones have been achieved and can be demonstrated by TraceLink:

- **Import** 100K serial numbers in under 1 minute
- **Provision** 100K serial numbers in under 10 seconds
- **Commission** 100K serial number in under 1 minute
- **Aggregate** a 100K batch of serialized products in under 1 minute
- **Ship** 100K serialized products in under 1 minute
- **Generate** an EPCIS message for a shipment in under 1 minute
- **Provide** a sub-second response for an API from warehouse operations to TraceLink
- **Read** 150,000 serial numbers per second
- **Write** 10,000 serial numbers per second for 100s of billions of serial numbers across all workloads for all customers simultaneously

As your business evaluates prospective serialization vendors, ask for a demonstration of these same tasks on a database containing several years of accumulated data. A demonstration on a clean, brand new database – or a pilot based on a limited production run that doesn't reflect your daily volumes – proves nothing in terms of performance and, in fact, should raise a red flag about that solution's ability to keep up at operational scale.

RDBMS vs NoSQL: At-a-Glance

RDBMS	NoSQL
Built to address enterprise-class problems with smaller numbers of transactions.	Designed to address internet-scale problems with billions to trillions of rows and records.
Ill-suited for operational challenges: performance degrades as data size grows.	Widely recognized for high scalability and performance that remains near-constant, regardless of data store size.
Relies on a single large machine to manage all processing.	Distributes workload across many machines to enable limitless scalability.
Requires archiving data to an offline system to maintain performance, rendering large subsets of your data unavailable in real-time.	Because performance remains constant regardless of data store size, archiving is never required.
Archiving scenario makes it impossible to manage supply chain returns and verification inquiries in real-time.	Serial numbers from 4 weeks or 4 years ago are immediately available for returns, verification inquiries, or any other reason.
Slow performance times impact operations as large transaction volumes take a long time to process, or cause the database to time out.	Demonstrable, real world performance of 150,000 serial number reads per second and 10,000 writes per second.
No proof it can keep pace with transaction load at operational scale in real-world businesses.	Additional proven performance stats from customers live in production.

Serialization solution vendors SAP, Axway, RfXcel, and Frequentz have all built their platforms on RDBMS technology. TraceLink utilizes NoSQL.

Meeting the Serialization Challenge and Eliminating Risk

Both the DSCSA serialization deadline and the European Union Federated Medicines Directive are fast approaching. If you do business in either region, your transaction volumes will ramp up steeply, and immediately. There will be no grace period to safely gauge if your solution will scale.

If you choose a solution provider who is putting all their eggs in the relational database basket, you will not succeed. RDBMS is a proven technology for some challenges, but not for serialization, where it must operate on large volumes of data at operational speed unrivalled by anything the pharmaceutical industry faces today.

TraceLink is wholly focused on compliance and the pharmaceutical supply chain, and has been serializing product for customers in production for more than 5 years now.

We initially tackled the challenge when volumes were low, and most deadlines were distant. When we discovered that RDBMS technology failed under the strain of serialization demands, we had time to research, build, test, and performance-tune a solution that meets serialization demands. NoSQL in the AWS environment, in concert with other key technologies, has been proven by TraceLink and all of our customers who are live in production, shipping serialized product. It meets the challenge today, and will scale to meet it in ten years, allowing your business to operate without risk, and grow without limitation.