In this paper:
Three major disruptions to ECM

*Challenge*: Scale ECM data model beyond the limits of SQL

*The Nuxeo Solution*: Hybrid NoSQL content repository engine

SQL or NoSQL? Nuxeo lets you decide

*Appendix*: Nuxeo’s JSON document-based data model

While scaling content storage is easy, thanks to content-addressable storage, big disks and cloud stores, *scaling the content data model is hard*, and will only get *much* harder, due to three major trends disrupting Enterprise Content Management (ECM) technology.

This paper explains why these three disruptions may result in SQL databases no longer providing sufficient scale for your ECM data model. You will also discover how Nuxeo, an elegant open source platform for rich content management, *has solved this challenge with a hybrid NoSQL approach.*
Introduction:
Are you ready for the three disruptive changes in ECM?

One of the world’s largest providers of communications and entertainment services wanted to create a customer-facing video on demand (VOD) repository. The repository needed to provide customers with 24/7 access to movies and TV series, while allowing them to easily find videos based on actors’ names, film rating, genre and hundreds of other metadata fields per video.

Behind the scenes, the VOD application must have advanced workflows to manage every aspect of its video library, such as automatically converting videos into select formats based on complex business logic. The videos are also enormous - up to 80GB each.

Having had poor previous experiences with proprietary ECM software, the company was interested in open-source technologies, which led them to evaluate Nuxeo as their application development platform. It was also clear that the sheer numbers of customer queries, videos added or removed from the library, and metadata fields required for all videos, contracts and other asset information would go beyond the level of scale possible using a SQL database.

Read on to see how Nuxeo made this next level of massive scalability possible.

Enterprise Content Management (ECM) software enables managers to easily and securely work with all of the information assets in an organization - documents, images, video, schematic drawings, spreadsheets and much more - to successfully complete key business processes. This simple definition of ECM assumes a lot of critical functionality is provided, including:

- **Access, view, read, modify and create content**, with full **security**, across all channels and devices - web and mobile
- Manage the **full lifecycle state** of every piece of content
- Enable teams to **collaborate** on related content (e.g., case management)
- Define **automated workflows** with customized business logic to deliver content, review and approve documents, manage versioning, etc.
- Enable the creation of **highly customized content-driven applications** that match unique business processes

To make this functionality possible, a true ECM platform must allow the creation and modification of relationships across content using rich metadata, maintained as a highly flexible data model.
While scaling content storage is easy, thanks to content-addressable storage, big disks and cloud stores, scaling data about the content is hard - and will very quickly get much harder - as three game-changing ECM trends continue to disrupt the ECM market:

1. **Content is now a magnitude larger in size and throughput**
   - from 1M to 1B content objects
   - mobile and ubiquitous network access
   - 10x increase in average file size from high definition media

2. **Content is now highly complex objects and not flat metadata sets**
   - data models are constantly evolving
   - content lives in complex objects with dynamic, nested properties
   - workflows are dynamic, adaptive processes

3. **Content types and delivery channels are increasing exponentially**
   - pictures and videos are becoming core to many applications
   - content delivery channels are very varied (mobile devices, cloud apps)

The most successful content-driven organizations will be those with advanced ECM systems capable of keeping up with these transformational changes.

This tech brief will inform you why scaling ECM data is challenging, as well as when and why relational SQL databases will not provide enough scale.

You will also discover how the Nuxeo Platform, an open source end-to-end system for creating, testing, deploying and managing content-based applications, has solved the ECM scalability challenge by offering a hybrid NoSQL repository engine. Using the open source technologies MongoDB and Elasticsearch, Nuxeo offers massive scalability for any content-driven application, well beyond what is possible by any other ECM offering today.
Challenge: Scale the ECM data model beyond the limits of SQL

While typical document management tools often rigidly define a document as a single file, a Nuxeo document can be a highly complex content object. Each content object can contain multiple documents and binary types, along with all related attributes (metadata and hierarchy), which you are free to define and modify to match your business needs.

To provide some context, let’s review how the Nuxeo core repository works in a pure SQL implementation (see chart below).

Making the most of SQL

Nuxeo stores the actual content (binary streams) into a BLOBstore, while storing its content attributes separately in the Visible Content Store (VCS). The VCS stores the content attributes and references and links to the content, as well as versioning, security access controls, content lifecycle states and workflow details.

The Visible Content Store is a SQL-based storage system that manages all metadata in multiple tables requiring joins. This is by design, as multiple tables provide better overall performance than
a *single key-value table*. Multiple tables provide faster access to large objects. Although access is slower for smaller objects, Nuxeo’s use of multiple tables enable the use of the SQL optimizer of the relational database in use, for creating fine tuned indexes and otherwise leverage the relational database engine’s capabilities.

The Nuxeo Platform is therefore designed to drive the best performance and scalability possible from SQL databases used to store and manage all metadata for the ECM data model.

While a SQL-based implementation of the Nuxeo Platform works well for the vast majority of our customers, it is not without limitations. Scaling of the content metadata storage with SQL will hit a wall as the use case becomes more challenging:

- **Scaling reads (query/fetch) becomes difficult** as queries become increasingly complex. Large document objects with complex, multiple attributes will require many joins to many tables. Queries requiring the full content object eventually become unwieldy. This *impedance mismatch* requires caching and lazy loading in the Nuxeo application.

- **Scaling writes becomes difficult** as writes become increasingly concurrent and complex, resulting in multi-table locking. An update to a single field of content metadata will require multiple writes to multiple tables and rows. The sheer numbers of rows and tables requiring writes will also increase dramatically depending on versioning requirements, the number of workflows and their degree of complexity.

- **Scaling reads and writes does not become difficult; it becomes impossible**, because ACID transactions cannot be distributed. Reads and writes must compete for resources.

The results from Nuxeo’s own SQL database benchmarking research (see chart, below) show that as write operations are added, they increasingly compete with read operations for resources and performance degrades significantly.
**When SQL is not enough**

The ECM use cases that prove most challenging for SQL databases to handle generally consist of some combination of such conditions as:

- 500 or more complex queries per second
- Daily batch updates impacting 100,000 or more documents
- Complex data models generating over 200 tables
- Over 20 million documents in storage
- A full audit history of all user activity and content versioning for several years is required

If these conditions do not exist in your Nuxeo-powered application, then your SQL database repository will continue to provide excellent performance at enterprise scale, while also providing ACID transaction functionality.

For those remaining use cases where SQL databases fall short, Nuxeo now offers an alternative Hybrid NoSQL option for an unprecedented level enterprise scalability and performance.

**The Nuxeo Solution: Hybrid NoSQL repository engine**

Nuxeo’s Hybrid NoSQL option utilizes MongoDB as repository storage backend, in tandem with Elasticsearch, which is now tightly integrated within the Nuxeo Platform (see chart, below).
MongoDB is not a drop-in replacement for platforms or applications built with a relational SQL database; however, the Nuxeo Platform is a fully pluggable system. Nuxeo’s storage of metadata, hierarchy relationships and CRUD operations is a distinct layer of abstraction, no matter what underlying storage method is used - greatly easing the task of making necessary infrastructure changes to integrate MongoDB.

The Nuxeo development team created a new document-based store (DBS) abstraction (see figure, right) to replace the platform’s SQL-based Visible Content Store, with no impact to the rest of the Nuxeo Platform and its underlying code. The DBS stores each Nuxeo document type (content object) as a fully dynamic and configurable JSON-formatted document (for further details, please see the Appendix below).

How hybrid NoSQL approach solves data model scaling challenges

- Scaling reads (query/fetch) becomes much easier as queries can be powered primarily by Elasticsearch instead of SQL. Elasticsearch also acts as a distributed query and caching engine for the repository, providing exceptionally fast reads at linear scale.

Recent Nuxeo tests revealed how dramatically the SQL impedance mismatch effect impacts content querying and fetching at heavy volume. Benchmarking the same Nuxeo application, document processing was 15x faster than the fastest SQL database implementation:

- **Process 20,000 documents**
  - 700 documents/s with SQL backend (cold cache)
  - 6,000 documents/s with MongoDB / mmapv1: x9
  - 11,000 documents/s with MongoDB / wiredTiger: x15

- **Process 100,000 documents**
  - 750 documents/s with SQL backend (cold cache)
  - 9,500 documents/s with MongoDB / mmapv1: x13
  - 11,500 documents/s with MongoDB / wiredTiger: x15

- **Process 200,000 documents**
  - 750 documents/s with SQL backend (cold cache)
  - 14,000 documents/s with MongoDB/mmapv1: x18
  - 11,000 documents/s with MongoDB/wiredTiger: x15
**TECHNOLOGY BRIEF**

- **Scaling writes becomes much easier**, as MongoDB is optimized for writing; even more so if MongoDB is also used as a native repository store via GridFS. Recently, the Nuxeo Platform, using MongoDB as its repository backend, was benchmarked with *5x faster bulk import over the fastest SQL-based backend*, under high concurrency loads or high-availability bulk loading scenarios:

  ![Diagram](image)

  - **Import 20,000 documents**
    - 750 documents/s with SQL backend
    - 1,400 documents/s with MongoDB/mmapv1
    - 3,200 documents/s with MongoDB/wiredTiger

  - **Import 100,000 documents**
    - 550 documents/s with SQL backend
    - 1,250 documents/s with MongoDB/mmapv1
    - 3,150 documents/s with MongoDB/wiredTiger

  *about 5x faster!*

However, multi-document writes present a new challenge, as NoSQL databases are not ACID compliant, but rather provide BASE functionality. This remaining challenge is addressed below.

- **Scaling reads and writes becomes not only possible, but easy.** Nuxeo’s benchmarking research using MongoDB as a backend confirms that write operations no longer compete with read operations, and are no longer blocked as they are with SQL:
"One more (tricky) thing": Handling multi-document transactions in NoSQL

MongoDB manages transactions only on a per-document basis. This is not an issue for the many Nuxeo Platform operations which involve only one document type; nor does it pose a problem at the Nuxeo core repository level, even when running large batch updates affecting many documents.

However, there are cases where MongoDB’s single document transaction management does present challenges at the Nuxeo Platform application level, where a single transaction may require the modification of multiple documents:

- **Workflows** are an obvious example, in which one end user action will typically update multiple documents as well as node data
- **Nuxeo Automation Chains**, a series of operations designed to run iteratively based on select business logic, are also transactional in nature
- **Transaction rollbacks**. The processing of a workflow, automation chain or other multi-document transaction typically must either fully complete, or be fully rolled back in the event of any error during the transaction

The Nuxeo Platform addresses the multi-document challenge by implementing additional transaction controls at the application level (see figure, right) including:

- tracking a transient state for documents, making changes to objects in memory
- flushing changes to the database only when there is another operation that expects those changes to be present
- making transactional rollbacks possible using an undo log of the application level transaction

Because multi-document operations must be performed as a transaction only in some cases, the Nuxeo Platform will reap the performance benefits of the non-transactional nature of MongoDB most of the time, while providing a sufficient level of transaction for multi-document writes.
**Nuxeo hybrid NoSQL approach in action…**

The video on demand (VOD) application for a leading communication technology company mentioned at the beginning of this tech brief was, basically, a compilation of all of the use cases discussed so far that are problematic when using a SQL database to store the content data model:

- **Very large content objects, as in hundreds of fields of metadata** (e.g., dublincore, Asset Distribution Interface (ADI) specification, etc.)

- **Lots of metadata fields subject to daily updates**, such as the “exhibition window” during which a video is licensed for on-demand viewing (which varies by customer location)

- **Multiple types of self-proliferating content**; for example, new format versions of each video must transcoded automatically based on workflow-based business logic with metadata. Other required content includes multiple thumbnail images for each video and detailed licensing contracts

After a multi-round competitive evaluation process of several open source ECM tools, the company concluded that “Nuxeo was going to allow us to be the most successful,” proceeding with the Nuxeo as its ECM development platform of choice.

Presently, the project is in an initial phase in which the VOD application is being offered to select B2B customers:

- **Cloud-based Nuxeo-powered application utilizes MongoDB and Elasticsearch**

- **Links to a content distribution network (CDN) used for binary storage of large digital assets, particularly HD movies (20-100GB), managed by Nuxeo Platform**

- **All metadata, access controls and custom workflows** are managed within the Nuxeo Platform

As the VOD application expands, it will eventually become the video backbone for the company going forward, with Nuxeo will be at the ready to enable more functionality. This will soon include expanded metadata requirements, including offering bundling options of select videos for purchase, to offer new customer value.
Conclusion: SQL or NoSQL? Nuxeo lets you decide

The Nuxeo Platform provides an end-to-end platform for creating, testing and deploying a vast spectrum of business applications, customized to precisely match an organization’s unique needs. We equip developers with a content repository, workflow engine, data model and APIs that are modular, flexible, and massively scalable.

Nuxeo gives you the choice of SQL or NoSQL repository backends based on your review and assessment of which option makes the most sense for your application. To make that decision:

- **Analyze the level of scale your application will require**
  - Align your SQL/NoSQL database tradeoffs as they make sense for your application; e.g., data availability, performance versus consistency, single point of failure

- **Select your backend based on your needs analysis**
  - Nuxeo’s own published benchmarks, with complete test environment details, help you make an informed decision
  - Ability to configure MongoDB is included as a bundle with the core Nuxeo distribution

- **Test, test, TEST (!) your application using your backend of choice**
  - Nuxeo makes application testing easy and automated
  - Customers have access to the same testing resource toolkit as used by our own developers, with full documentation

The Nuxeo team is constantly working to enable the most demanding ECM data models and application requirements will be supported at a level of massive enterprise scalability that other tools simply cannot match. By offering MongoDB as a NoSQL repository backend alternative when SQL just won’t do, the Nuxeo Platform is now positioned to be the highest performing ECM for the widest possible range of use cases.

Learn More: Next Steps…

- **Download the Nuxeo Platform**, or **start a free online trial**.
- **Visit the MongoDB for Nuxeo page**.
- **Visit the Elasticsearch for Nuxeo page**.
Appendix: Nuxeo’s JSON document-based data model

The new Nuxeo DBS (MongoDB) stores all attributes as JSON-formatted documents, which are ready to do some very heavy lifting for your business applications:

- **All JSON documents in Nuxeo are fully configurable and dynamic**, capable of supporting highly complex, customized document types (content objects), capable of supporting thousands of metadata, nested data structures, subdocuments and more
- **Each document type is a single JSON document**, maintained as a single collection.
- **Each JSON document also includes references to all related binary files** in the JSON digest. Note, the binary files themselves are still stored in the same BLOBstore as with SQL.

The following sample excerpts from a JSON document shows how Nuxeo dynamically computes and maintains its JSON documents to enable relational-style queries while also setting and enforcing all security permissions with further notes provided:

```json
{...
  "ecm:parentId": "52a7352b-041e-49ed-8676-32...",
  "ecm:ancestorIds": [
    "00000000-0000-0000-0000-000000000000",
    "1506bf11-4cb0-4ad4-94e3-ab0c1672c6c0",
    "399afabc-1eb5-4650-9049-b59d6da8d989"...
  ],
...

"dc:title": "My Document",
"dc:contributors": [ "bob", "pete", "mary" ],
"cust:Address": {
  "street": "1313 Mockingbird Lane",
  "city": "Los Angeles",
  "state": "CA"...
}...

"my:attachedFiles": [
  { "name": "doc.txt",
    "length": 1975,
    "mime-type": "plain/text",
    "data": "0111fefdc8b147380e56f3e568115"
  },
  { "name": "doc.pdf",
    "length": 29344,
    "mime-type": "application/pdf",
    "data": "20f42df3221d61cb3e6ab8916b248216"...
} ]}
```

In order to manage hierarchy (parent-child relationship):
- each document is given a **parentId** system attribute
- to manage a query on path, **ancestorIds** system attribute is also computed automatically by the Nuxeo framework, listing all ancestors of each document

These sample attributes are stored as one might expect for a JSON document, including:
- regular properties (e.g., **dc:title**),
- list properties (**dc:contributors**),
- complex properties as lists of JSON subdocuments (**cust:Address**, **my:attachedFiles**)

Continued on next page...
About Nuxeo

Nuxeo provides an extensible and modular Enterprise Content Management platform that enables architects and developers to easily design, build, test and deploy content-driven business applications. Designed by developers for developers, the Nuxeo Platform offers modern technologies, a powerful plug-in model and extensive packaging capabilities. It comes with ready-to-use Document Management, Digital Asset Management and Case Management packages. 1000+ organizations rely on Nuxeo to run business-critical applications, including Verizon, Electronic Arts, Sharp, FICO, the U.S. Navy, and Jeppesen, a Boeing Company. Nuxeo is dual-headquartered in New York and Paris.