Hyland™

ARTICLE

Designing a modern platform architecture for content services

A clear path to transition from legacy information management systems

Introduction

Most Enterprise Content Management (ECM) systems were architected during an age when mobile phones were the size of bricks, when Miami Vice pastel suits were all the rage and when Joe Montana was winning Super Bowls. So is it any wonder that these systems are not able to cope with the modern, digital-first environment we find ourselves in today?

This article explores some of the modern technologies currently available that can be leveraged to get more value from enterprise data and content — technologies that had not even been thought of when most legacy ECM systems were built.

Individually, these technologies can deliver huge advancements in how you create, manage, process, access, analyze and secure information within your enterprise. When deployed together, they create a modern content services platform (CSP) that can empower innovation in the modern era.

Let's get to know Hyland's Nuxeo Platform, a cloud-native CSP that uses artificial intelligence to drive efficiency.

Build or buy?

Many organizations consider the build vs. buy question. To help you answer the question, this article details the various components that a modern organization needs to consider when building, or buying, a solution to manage its information assets.

One of the key questions faced today by CIOs, Chief Architects and other IT leaders within all industries is whether to build their ideal information management architecture themselves, or to purchase and deploy solutions from outside vendors. The natural inclination for many at first is to build it. However, creating a custom information environment is time- and resource-intensive — and it often fails, due to the complexity and skills required to create a home-grown information environment that addresses current needs while having the flexibility to adapt as business and technologies evolve.

For IT leaders considering their options or seeking to shift from the build-it path, this article will provide you with insights on the key architectural components to consider when evaluating vendor offerings for delivering a future-proof information management environment.

If you do choose the build route, each of these components will need to be configured, integrated to other components, linked to standardized security mechanisms, then tested for scalability, robustness, user-acceptance and ease of use.

If you consider the "buy" route instead, each section of this document has a "Nuxeo Platform specifics" sidebar highlighting how the Nuxeo Platform uses the component or technology in that section.

We're excited to share how Nuxeo Platform has put its own spin on the most innovative and useful technologies of today, whether that involves how we use NoSQL databases to provide the ultimate flexibility and scalability, or how our cutting-edge work with low-code development allows users to rapidly develop and deploy custom applications.

Throughout 15 years of software development, over 1 million lines of open source code and thousands of hours have gone into creating this modern content services platform.

A modern content services architecture

Modern CSP architecture requires many pieces, all needing to work in harmony. In addition to being scalable and secure, it must also enable you to better manage and derive value from your organization's information assets.

There are six key architectural features and concepts that form the core of a modern content services architecture:

- Cloud-first
- Cloud-native
- NoSQL database
- Low-code application development
- REST API enabled
- Open source

Each of these are explained below, and the Nuxeo Platform difference sidebars illustrate how they come together to serve as the architectural underpinnings of the Nuxeo Platform.

6 key architectural features

1. Cloud-first

Many companies want to prioritize the use of cloud computing resources and services rather than relying on traditional on-premises data centers. The main reasons for choosing cloud environments are:

- Agility and scalability
- Cost efficiency
- Innovation and modernization
- Risk management and compliance
- Improved collaboration

Agility and scalability

The concepts of 'agility' and 'scalability' are central to the cloud-first approach, and they encompass several key aspects:

- **Rapid deployment and flexibility:** Companies can launch new applications or expand existing ones without the lengthy procurement and setup process associated with traditional on-premises infrastructure. This flexibility is crucial for responding to market changes or business opportunities promptly.
- **Elasticity:** Cloud services are inherently elastic, meaning they can be scaled up or down with minimal effort. This elasticity is essential for handling fluctuating workloads. For instance, during peak business periods or unexpected spikes in demand, cloud resources can be automatically increased to maintain performance levels.

Cost efficiency

A cloud-first model shifts the financial burden from capital expenses for hardware and infrastructure to operational expenses. Companies pay for computing resources as they use them, rather than investing large sums upfront for purchasing and maintaining hardware.

Also, with cloud services, the responsibility for maintenance, upgrades and hardware failures falls to the cloud provider. This reduces the need for in-house IT maintenance teams and cuts down on the overhead costs of running data centers, such as power, cooling and rental space.

In short, a cloud-first approach provides a cost-efficient solution by converting fixed costs to variable costs, reducing overhead, optimizing resource usage and benefiting from the economies of scale that cloud providers offer.

Innovation and modernization

The cloud-first approach is intrinsically linked to driving innovation and modernization within organizations for several reasons:

- **Access to cutting-edge technologies:** Cloud providers invest heavily in the latest technologies, from machine learning, to IoT and Big Data Analytics. Adopting a cloud-first policy allows businesses to leverage these advanced tools without significant investment in R&D
- **Faster experimentation:** The ability to quickly spin up and tear down resources in the cloud reduces the time and cost associated with experimentation. Businesses can test new ideas and iterate rapidly, fostering a culture of innovation
- **Continuous improvement:** Cloud platforms are continually updated, providing access to the latest features and improvements without additional cost or effort for the businesses that use them
- **Global scalability:** The cloud enables organizations to deploy services worldwide with ease, modernizing their reach and operations on a global scale
- **Modernizing legacy systems:** By moving to the cloud, businesses can modernize legacy applications, improving performance and security while adding new capabilities and driving new business value.

Risk management and compliance

This aspect is a critical component that addresses security, legal and regulatory requirements. In a cloud-first approach, several key points are addressed:

- **Shared responsibility model:** Cloud providers take on the responsibility for the infrastructure's security, while customers are responsible for securing their data within the cloud. This model can help organizations manage risks more effectively as the providers are equipped with sophisticated security measures that may be beyond the reach of individual organizations
- **Regulatory compliance:** Cloud service providers often comply with a wide range of international and industry-specific standards and regulations, such as GDPR, HIPAA and PCI-DSS. Leveraging cloud services can help organizations meet these compliance requirements more easily as they benefit from the provider's certifications
- **Data protection:** Cloud providers invest heavily in security technologies and practices, including encryption, firewalls, intrusion detection systems and multi-factor authentication, which are critical for protecting data and managing risk
- **Disaster recovery and business continuity:** The cloud's distributed nature allows for robust disaster recovery plans. Data can be backed up in multiple locations geographically, ensuring business continuity in case of localized physical disasters
- **Regular security updates:** Cloud providers regularly update their infrastructure with the latest security patches and software updates. This ensures that the cloud infrastructure is less vulnerable to new threats, reducing the risk of security breaches
- **Advanced threat detection:** Cloud services often include advanced threat detection capabilities that use machine learning and artificial intelligence to identify and respond to security threats in real time
- **Audit and reporting facilities:** Cloud platforms provide comprehensive logging and reporting tools that enable organizations to audit and monitor their environments for any unusual activity, which enhances their ability to manage risk.

Cloud-first does not mean "cloud only"

Some applications or data may remain on premises, due to regulatory considerations for example. The goal of a cloud-first strategy is to consider cloud solutions as the preferred option and to leverage the benefits of cloud computing wherever feasible. A flexible solution that can handle such hybrid environments is required.

The Nuxeo Platform Difference

Nuxeo Platform is designed to align with multiple types of cloud first strategies, from buying a full CSP service fully managed by Hyland to running your own cloud native CSP in a public private cloud.

By relying on open-source technologies that are both available as managed services on most cloud infrastructures and as software that you can run yourself, Nuxeo Platform enables organizations to fully leverage the potential benefits of the cloud. In other words, unlike legacy solutions retrofitted for the cloud, Nuxeo Platform does not just run in the cloud on provisioned virtual machines.

Being extremely flexible, Nuxeo Platform is typically deployed in a managed container environment and supports scaling up or down, high availability and disaster recovery across Cloud regions. When deployed on the cloud, it leverages the managed services for database, file storage, search index, message queues, artificial intelligence, serverless computing, and more. When deployed on-premises, it leverages the equivalent software such as MongoDB, Elasticsearch/OpenSearch, Kafka and MinIO to name a few.

As an extensible platform, it can handle data anywhere — whatever the load — with no limits in terms of storage or connection, and enables organizations to leverage the same application for workloads on the cloud and sensitive workloads not suitable for the cloud.

Nuxeo Platform is cloud-native, designed for the cloud from day one. It inherently supports a cloud-first strategy and takes advantage of cloud scalability, elasticity and flexibility, unlike platforms that have been retrofitted for the cloud.

- **Elastic scalability:** Nuxeo Platform can automatically scale resources up or down based on demand, which is a fundamental characteristic of cloud-first applications. This ensures that performance remains consistent as the workload grows
- **Independent component scalability:** Nuxeo Platform can scale components independently (database, index engine, storage, transformation engines, etc.) and update them without downtime, which is a hallmark of cloud-first applications
- **Containerization:** Nuxeo Platform leverages containerization technologies like Docker and orchestration systems like Kubernetes, which are essential for cloud deployments, allowing for rapid scaling and management of services
- **API-first design:** Nuxeo Platform provides a robust set of APIs, ensuring that it can integrate smoothly with other cloud services and systems — which is critical for a seamless cloud ecosystem
- **Multi-tenant capabilities:** Nuxeo Platform is built to support multi-tenancy, which is essential for serving multiple customers from a single cloud instance while keeping their data separate and secure
- **Cloud-native storage:** Nuxeo Platform makes use of cloud object storage, which is infinitely scalable and more cost-effective than traditional file storage methods, ensuring that it can handle extensive digital asset repositories
- **Hybrid:** Nuxeo Platform supports hybrid environments, where parts of the system in the cloud, and other is on-premises
- **Backups and disaster recovery:** Nuxeo Platform includes features for backups and disaster recovery, leveraging the cloud's distributed nature to enhance data protection and availability
- **Cloud optimization:** Nuxeo Platform is optimized for cloud resources utilization, ensuring cost-effective operation by minimizing wasted resources and adapting to the load dynamically
- **Built-in Compliance and Security:** Nuxeo Platform implements cloud security best practices and compliance measures to meet the stringent requirements of a cloud-first environment.

By focusing on these cloud-centric features and capabilities, Nuxeo Platform ensures that it is fully equipped to support organizations that prioritize a cloud-first strategy, providing them with a robust, scalable and secure content services platform.

2. Cloud-native

One of the problems with many legacy software applications today is that they were built 10, 20 or even more than 30 years ago, and were designed as a single — often referred to as monolithic — piece of code. This causes a number of issues. For example:

- Updates, even minor ones, have to be applied to the whole program — a costly, time-consuming process.
- If one part of the software is heavily loaded, then the whole system slows to accommodate it.
- These systems try to build everything into one system — for ECM this means capture, workflow, analytics, reporting and so on. This results in a growing system footprint, leading to longer installation times, debugging processes, support requirements and increased difficulty in re-engineering to leverage new technologies like cloud and mobile.

So, what's the alternative?

Despite being touted as a concept since early 2005, modular software architecture has only really taken off with the adoption of the cloud in the 2010s. The concept of decoupling the specific functional elements of a software application has allowed software design to break free of the shackles of single executables to create flexible, interconnected, collaborative sets of integrated modules.

In the early days these modules were typically created by one vendor. But over time, as open systems and restful APIs have become more prevalent, standardized components have been created for everything from single sign-on authentication, to databases, to visualization.

This approach is often called cloud-native — not because it requires cloud-hosting to be used, but because most cloud services are built using this method.

Cloud-native encompasses a lot more than just signing up with a cloud provider and using it to run your existing applications. It affects the design, implementation, deployment and operation of your application.

The Cloud Native Computing Foundation defines cloud-native as: "… computing [that] uses an open source software stack to be:

1. Containerized

Each part (applications, processes, etc.) is packaged in its own container. This facilitates reproducibility, transparency and resource isolation.

2. Dynamically orchestrated

Containers are actively scheduled and managed to optimize resource utilization.

3. Microservices-oriented

Applications are segmented into microservices. This significantly increases the overall agility and maintainability of applications.

Let's explore each of these architectural elements in detail.

Containerized

Containers package software with everything needed to execute it into a single package — for example, a Java VM, an application server and the application itself. This container is then executed in a virtualized environment to isolate the containerized application from its environment.

The main benefits of this approach are that the application becomes independent of the environment, and that the container is highly portable. The same container can be executed on a development, test or production system. If the application supports horizontal scaling, multiple instances of a container can be started or stopped to add or remove instances of the application based on user demand.

Quite simply, the container has everything your application needs — you just need to start it.

Orchestration

Deploying an application — with all of its associated library and reference dependencies — in a container is the first step to solve the deployment challenges faced by traditional software builds. But to benefit fully from a cloud-native platform, there are additional requirements.

Running multiple application nodes will require you to:

- Monitor the system
- Trigger the startup or shutdown of containers
- Ensure all required configuration parameters are in place
- Balance the load between active instances
- Ensure security and collaboration between containers

Orchestration systems provide the tools to automatically react to any unexpected changes in system load, and to streamline the management of cloud-native systems.

Microservices

Cloud-native applications are typically built as a system of microservices.

This architectural style implements a system of multiple, small applications (or microservices), each of which work together to provide the overall functionality of your system. Each microservice delivers one specific piece of functionality, has clearly defined inputs and outputs, is developed and operated by a relatively small team and, most importantly, has a published and managed API.

The use of microservices has a number of benefits:

1. The delivery and deployment of large, complex applications are broken down into smaller, more manageable pieces, each of which:

- Are smaller and faster to test
- Can be deployed independently
- Enable you to organize the development effort around multiple teams, each of which owns and is responsible for one or more single service. Each team can develop, deploy and scale its services independently of the other teams.

2. Each microservice is relatively small, making it:

- Easier for a developer to understand, extend and maintain
- More productive for developers using a simpler and faster integrated development environment
- Faster to start, making developers more productive and speeding up deployments

3. Elimination of long-term commitment to a specific technology stack, meaning that:

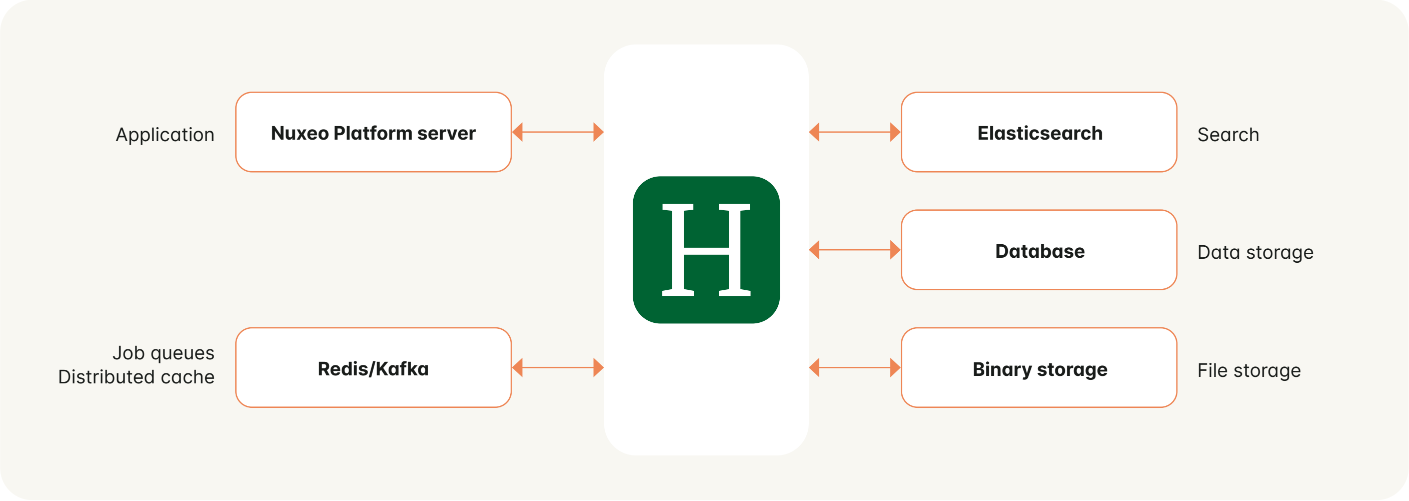
- When developing a new service, the developer can pick the best technology stack for the microserver
- When making major changes to an existing service, it can easily be rewritten using any new technology stack

The Nuxeo Platform difference

Nuxeo Platform was designed as a cloud-native solution from day one and has been built to:

- Make the best use of the massively scalable NoSQL to store and manage metadata
- Utilize tools such as Elasticsearch to enhance the usability and flexibility of finding content
- Scale independently and horizontally to provide the most cost-effective enterprise-grade performance

Nuxeo Platform seamlessly combines these elements, delivering the system architecture you desire without requiring the millions of labor hours needed for a DIY build.



3. NoSQL database

A relational database (RDBMS) organizes data into fields, tables and their associated relationships. However, in the real world, not all data fits naturally into a tabular structure (think of unstructured content, trees and graph structures). An RDBMS works around this limitation by introducing artificial structures such as data in normalized form and parent-child records. These techniques shoehorn nontabular data into a tabular, record-oriented format.

As a result, the structure of the records in a table (also known as the schema) must be the same. For data that includes optional values, this creates tables with many empty columns — and subsequently, a lot of wasted space. We can work around this by breaking up records into subparts (or subtables), which hold the subfields, and then do cross-table joins when querying to create complete records. However, this creates additional work every time data is requested, even for relatively simple queries.

A second result is that schema changes are expensive. To amend even one record with a new field, that new field needs to be added to every single record in the table.

By contrast, in a document-oriented database (or a NoSQL database) each document is stored independent of the other, so there is no wasted space for optional fields and adding a new field to one document does not affect any of the other documents. In addition, there are several other benefits of using NoSQL databases over their relational cousins in a modern content services platform.

"As-is" content

NoSQL databases support storing data "as-is." Key-value stores (RDBMS) give you the ability to stores simple data structures, but document-based NoSQL databases provide you with the ability to handle a wider range of structures, including flat or nested entities.

In the microservices world, data needs to be shared and communicated between microservices securely and consistently. This communication happens via message transfer. Typically, the data takes one of these formats:

- A binary object to be passed through a set of layers
- An XML document
- A JSON document

Being able to handle these formats natively in a NoSQL database decreases the amount of code required to convert from the source data format to the format that needs storing. This reduces the complexity of the system and increases the speed of operation.

Unstructured text support

The vast majority of data in enterprise systems is unstructured. NoSQL databases handle indexing of unstructured text either as a native feature or via an integrated set of services. Nuxeo Platform uses the industry standard Elasticsearch to provide this capability.

Being able to manage unstructured text increases the amount of information that can natively be managed and can help organizations make faster and better decisions. For example, advanced use cases include support for multiple languages with faceted search, fuzzy and similarity-based search functionality, as well as enterprise-grade scalability and speed. Advanced features also include support for dictionaries and thesauri.

Flexibility

Because of the schema-agnostic nature of NoSQL databases, they're very capable of managing schema changes. This means that if your business now requires additional information to be stored against a particular entity, this can be accommodated without issue.

If a document structure changes, these indexes allow organizations to use the information immediately rather than having to wait for several months before you can test and rewrite systems. Given the increasingly rapid rate at which businesses need to adapt, this flexibility in the underlying information architecture is critical.

Simultaneous and multiple data structures

Many applications need simple object storage, whereas others require highly complex and interrelated structure storage. NoSQL databases provide support for a range of data structures:

- Simple binary values, lists, maps and strings
- Related information values can be grouped in column families
- Highly complex parent-child hierarchical structures

Reduced reliance on "magic" queries

Structured Query Language (SQL) is the predominant language used to query relational database management systems. Over the years, structuring queries so they perform well has become more of an art than a science, with complex multi-table joins being the norm in order to get anything done. The optimal design of those queries has provided many SQL engineers with well-paid roles for many years.

Although NoSQL databases support SQL access, they only do so for compatibility with existing applications, such as business intelligence (BI) tools. NoSQL databases support their own access languages that can interpret the data being stored, rather than requiring a relational model within the underlying database.

Application developers don't need to know the inner workings of databases before using them. NoSQL databases empower developers to work on what is required in the applications instead of trying to force relational databases to do what is required.

This developer-centric mentality to the design of databases and their access via application programming interfaces (API) is one of the key reasons why NoSQL databases have become very popular among application developers.

Scalability on commodity hardware

NoSQL databases automatically manage the partitioning of a database across several servers. This means that as data storage requirements grow, the infrastructure can be extended by adding (relatively) inexpensive servers. These servers are added to the existing database cluster, making them work as a single data service. This is known as horizontal scaling.

Contrast this to the relational database world, where new, more powerful and thus more expensive hardware is required to scale up (aka vertical scaling).

One of NoSQL's major assets? Its ability to provide high availability and low-cost scalability by using inexpensive hardware and storage via its database.

Metadata matters

A key aspect of any information management platform is how metadata is handled. Numerous benefits are achieved by using a NoSQL database to manage metadata.

Firstly, the flexible nature of NoSQL means that schema changes can be readily accommodated, versus the rigid nature of many traditional ECM schemas. No longer does a schema change require major updates to the entire system.

Enterprises today manage a wide variety of content with increasing focus on rich media in addition to text and pdf content. The ability for NoSQL to manage content "as-is" delivers simplicity and speed. In addition, the ability for NoSQL to scale massively on commodity hardware provides enterprises with a cost-effective way to address the increasing volume of content and data required to be managed in today's enterprises.

Finally, NoSQL enables a break from traditional document focused systems — meaning that businesses with cases, matters, patients and more complex content-driven processes and workflows are able to make use of the power and flexibility of managing information securely and efficiently.

The Nuxeo Platform difference

Nuxeo Platform fully embodies the power of NoSQL databases to deliver benefits such as “as-is” content storage, massive scalability and simultaneous data structures — but the most important, from a content services standpoint, is metadata flexibility.

The ability to dynamically change the underlying schema of a content services platform is both new and extremely significant. In traditional ECM systems, schemas are fixed and require intensive and costly scoping exercises upfront to define and create the schemas necessary for a specific project. If that schema ends up being wrong or needing an amendment, the whole underlying structure of the relational database would need to be changed. The flexibility of NoSQL allows changes to be made to the content schema dynamically, allowing the Nuxeo Platform CSP to be as agile as the organization it supports. This, combined with the ability to provide multiple views or facets on information and content, creates a key differentiator for platforms like Nuxeo Platform that deploy on NoSQL.

Super scalable
One billion assets benchmark

Nuxeo Platform performance at a glance

1B
Assets (images, docs, videos)

40k
Documents imported/second

5x
Faster bulk import than any SQL database

30x
Faster than the competition

4. Low-code application development

User requirements within the business world are more challenging than ever before. Largely fueled by a rise in consumer technologies providing simple user interfaces with rich cross-app integration (like mobile and tablet apps), business users now expect the same level of simplicity and ease of use within the enterprise. These capabilities cannot be delivered using the traditional methods of software application development, like the waterfall method.

The new model for building applications is based around a low-code, rapid design and prototyping environment — a user-friendly capability to quickly create solutions that integrate content, data, external services and workflows that actually solve real challenges faced by the modern knowledge worker.

This style of app development can only be achieved if built on top of a strong foundation — one where business logic can be combined with consistent and reliable data. With that foundation in place, however, low-code development can deliver significant benefits.

Stakeholder engagement and satisfaction

Low-code platforms allow rapid app creation and updates, allowing developers to share fully functional features with stakeholders within days or even hours. This dynamic and rapid cycle fosters more active engagement and provides a clear, ongoing sense of progress and responsiveness.

In addition, something as simple as delivering an application that uses the domain-specific language or terms that an end-user is expecting can reduce the time taken to learn a new system and reduce resistance to change.

Lower risk, higher ROI

With a low-code platform, all the security, cross-platform support and data integration capabilities have already been built. As a result, developers and others can focus on solving business problems with rapid prototyping and visual delivery.

Ongoing delivery

App development does not end with a successful launch. Updates, feature improvements and bug fixes all need to be considered. A low-code platform provides the ability to make complex updates and deliver new features in minutes rather than days, reducing the impact on other projects in the development queue.

Reduction of the IT skills gap

The intuitive, visual editors in low-code platforms reduce the dependence on highly skilled developers in order to realize value. Years of specific programming language knowledge isn’t required, which allows more developers to spend more time contributing to business-critical projects and innovation. In some cases, even nontechnical stakeholders can learn to build their own prototypes. Waiting for IT to deliver solutions is a thing of the past.

Quick change and innovation

Custom coding an application is time-consuming and labor-intensive. Low-code platforms enable developers to innovate faster and easily create new functionality or customize features in days or weeks, rather than months or years. This shift allows developers to continually focus on delivering innovation and new value to the business instead of fixing bugs or maintaining defunct, outdated systems.

Mobile delivery

The need for mobile apps drives new and innovative approaches to app development. Low-code platforms provide developers and organizations with toolsets that enable them to develop and deliver applications that delight staff and customers quickly and consistently.

The Nuxeo Platform difference

One of the downfalls of ECM solutions was their fixed, and often arcane, interfaces. This was so pronounced that a separate industry, known as case management, was created in an attempt to solve the problem. From a user perspective, they really do not care — all they're interested in is the system providing a simple-to-use interface that combines data, content and workflows to allow the user to get their job done.

However, the dilemma here is that individual roles within an organization (and even individual users) tend to work in different ways. This degree of personalization was never supported with ECM systems, but the low-code development approach provided by modern CSPs like Nuxeo Platform delivers this.

Hyland's low-code Nuxeo Platform enables the delivery of unique, personalized interfaces built specifically for departments, teams or even individuals. Solutions are formed using reusable components, assembled in a visual environment, and connected together to deliver the perfect combination of data, content and process.

A single, monolithic interface cannot meet the needs of multiple users within the enterprise. However, a low-code development platform that can create several dedicated solutions built on top of a single information foundation can address this, and many more needs.

Business agility

Low-code development

Cloud-native

Open source

APIs & integration

Containerized delivery

Public/private On-premise hybrid

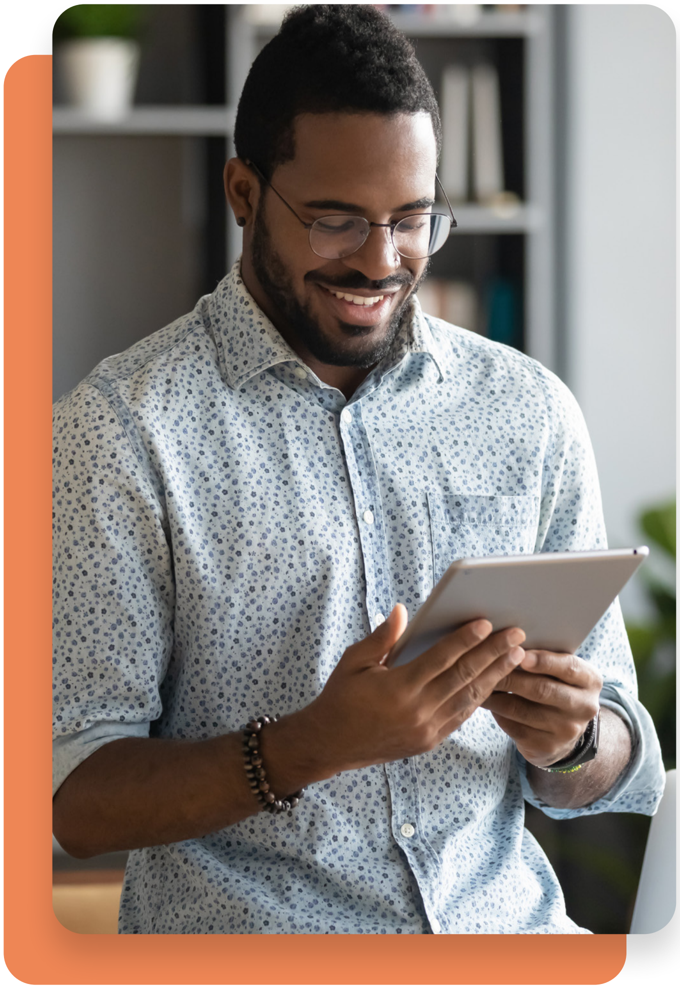
Mobile-enabled

5. REST APIs

One of the core elements of a cloud-native architecture is the concept of microservices — connected, modular, specific function entities that facilitate dynamic and modern software design. As previously discussed, microservices enable the standardized interconnection of applications and other microservices, both on-premises and in cloud environments. But to do that, these microservices need a structured way to communicate with each other. This is achieved via application programmers' interfaces or APIs.

An API is a defined set of ways in which to communicate with and control an application, microservice or any software entity. This typically includes ways to control variables and parameters, create new instances, execute functions and so on.

There are numerous benefits to providing and utilizing REST APIs including:



Separation of client and server

The REST protocol completely separates the user interface from the server and the data storage. This improves the portability of the interface to other platforms, increases the scalability of projects and allows the different pieces of any component to evolve independently.

Language independence

A REST API adapts to the type of syntax or platforms being used, which gives considerable freedom when changing or testing new environments within the development. With a REST API, you can easily have PHP, Java, Python or Node.js servers.

Future-proofing

From a content services perspective, REST APIs serve two purposes. First, they enable future-proofing of the platform — allowing integration to the best-of-breed tools of today, then the ability to swap them for the best-of-breed tools of tomorrow as they are developed. Second, they allow the platform to become a microservice host of its own — enabling parts of the platform to be exposed to other consuming entities such as websites, applications, mobile apps or other content microservices.

The Nuxeo Platform difference

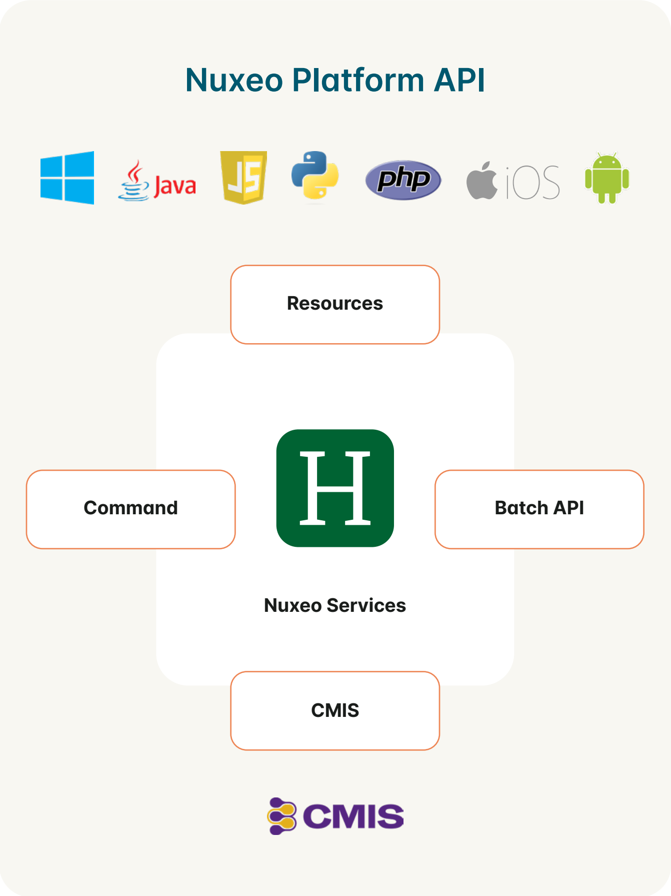
Every new piece of functionality in Nuxeo Platform is created "API-first," which means functionality is built from the ground up to be connected, documented, and available throughout all aspects of the platform. This approach fully embodies the open nature of Nuxeo Platform.

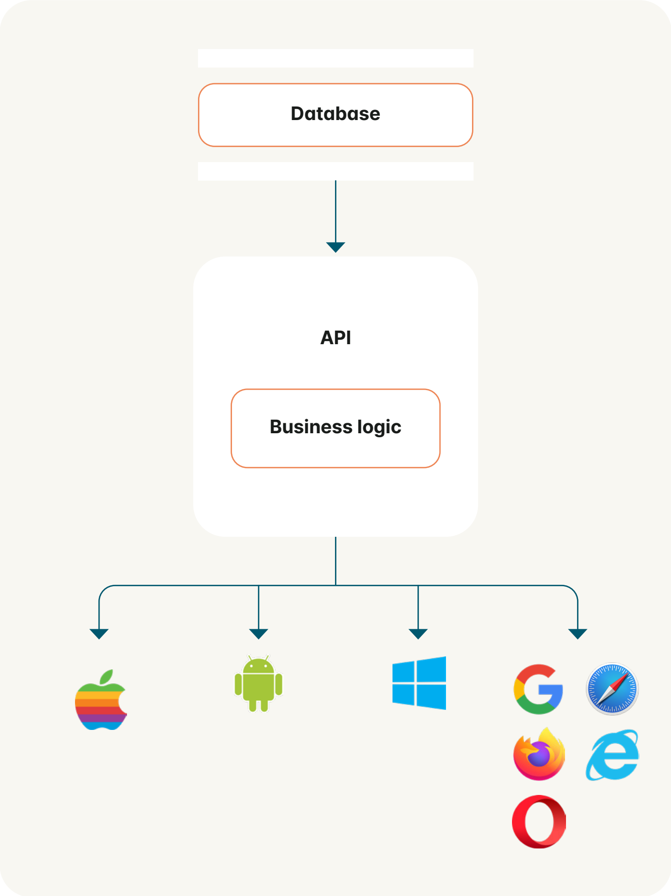
Nuxeo Platform has been an advocate of open APIs for many years and this extends to the use of REST-APIs for both consumption and delivery of services.

For consumption, Nuxeo Platform can make use of any REST-API enabled application, service or tool. This method is used to integrate to numerous pieces of the overall Nuxeo Platform solution including AI, digital signature and records management capabilities.

For delivery, any content, data or workflow from within the Nuxeo Platform can be made available via a REST-API. This includes services from external services that have been consumed by Nuxeo Platform, providing developers with the ability to "roll their own" APIs within the platform and expose, promote and potentially monetize them.

> We built the Nuxeo Platform from the ground up to be as flexible and extensible as possible and we want to provide that same high level of flexibility and extensibility through the API. This means that our API must be dynamic and composable.
>
> **Thierry Delprat, CTO, Nuxeo Platform**





6. Open source

Once upon a time, open source was thought of as free, indie, unsecure software. Those days are over. Open source is now a recognized business model for the delivery of software that adheres to the highest coding standards. Open source now means developer-enabled software that opens its source to the world for inspection.

Enterprises looking to make smart use of open source software will find plenty of reasons to do so, such as:

Community

Enterprise open source solutions often have thriving global communities around them. They are bound by a common drive to support and improve a solution that both the enterprise and the community benefit from (and believe in). United around improving these solutions, they introduce new concepts and capabilities faster, better and more effectively than internal teams working on proprietary solutions.

Transparency

Open-source code means you get full visibility into the code base as well as all the discussions about how the community develops features and addresses bugs. In contrast, proprietary code is largely produced in private by a few people and may come with unforeseen limitations and other unwelcome surprises. With open source, you're protected against lock-in risks and can see exactly what you're getting.

Reliability

The reliability of open source code tends to be superior because there are more eyes on it. The output tends to be extremely robust, tried and tested code. In fact, open source code now powers about 90% of the internet and is being rapidly adopted across major enterprises for this reason.

Better security

Open source software code is often more secure than proprietary code because it is much more thoroughly reviewed and vetted by the community (and any issues that do arise tend to be patched more diligently). Despite often being a point of hesitation for enterprises looking to adopt open source software, concerns about security are largely unfounded.

Everyone's doing it

Many large enterprises are now implementing open source solutions and often making corporate policies promoting the use of open source, bringing the strength of their resources to the communities that support open source solutions.

In addition to these advantages, open-source software has the long-term viability to outlast proprietary developers that come and go. And, thanks to supportive communities that are energized to continually introduce innovations, open source software remains at the forefront of advancing technology as a whole, meeting enterprise needs as they evolve going forward.

The Nuxeo Platform difference

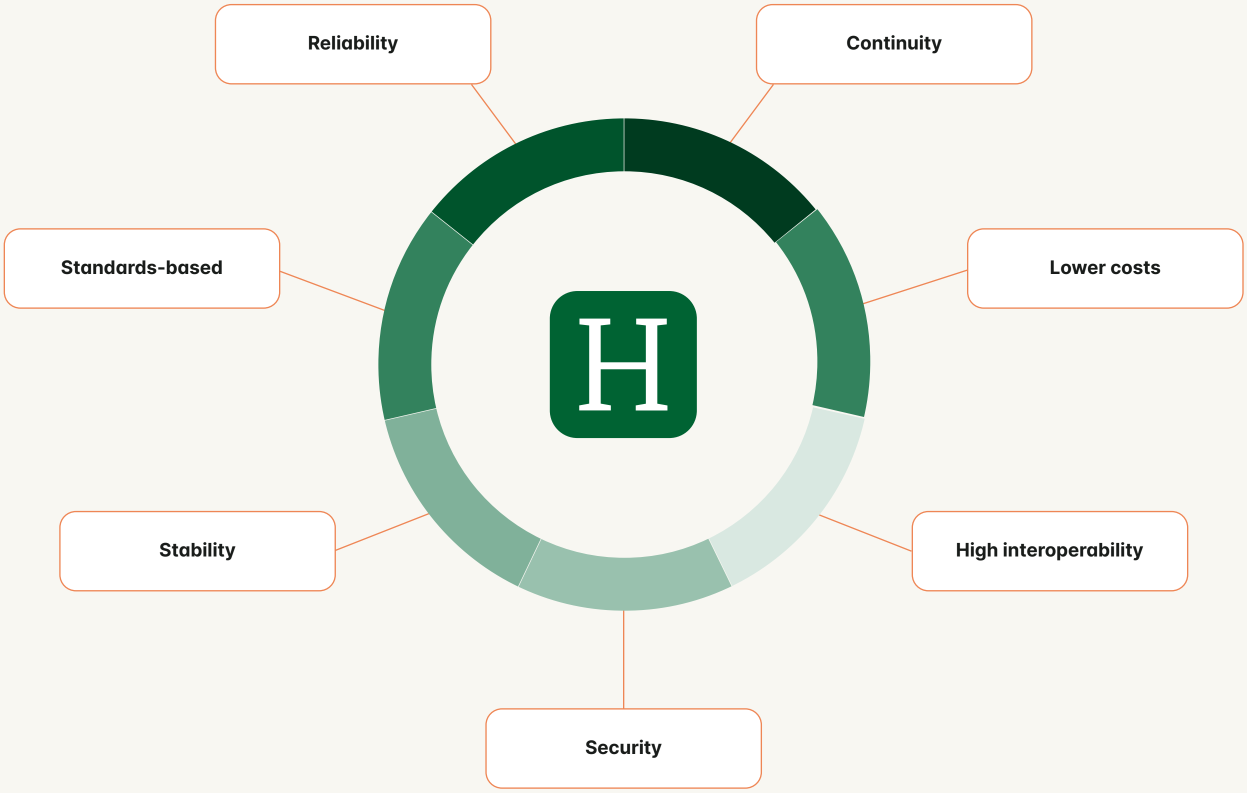
There are a number of open source solutions available within the ECM and CSP market. However, many of these use open source solutions as a marketing tool more than as a fundamental underpinning. Nuxeo Platform is different for these reasons:

1. One code base

Nuxeo Platform delivers only one version - there is no "community" version or "commercial" version, just one code base for all. Hotfixes are made available to subscribed clients 90 days before they are published to the community. This means that absolutely anyone can download the platform and start making use of it-for free. The Nuxeo Platform generates income by providing access to package distribution and a cloud- based configuration tool called Nuxeo Studio that enables developers to fast-track configuration and deployment, as opposed to working directly with the platform via the API.

2. Open kitchen

For Nuxeo Platform, open source means being held to the highest standards by the developer community. To that extent, we operate an open kitchen approach, providing full access to all of our source code — to anybody. In the same way that chefs allow diners to look into an open kitchen to see exactly how their food is being prepared, Nuxeo Platform's open kitchen software development facility provides a similar experience to the developer community.



Nuxeo Platform architecture

Nuxeo Platform brings the strongest elements of a modern content services architecture together into one platform. Hyland's Nuxeo Platform is equipped with industry-leading CSP capabilities. The platform:

- Is cloud-native, with a state-of-the-art modular component architecture
- Has a database-agnostic architecture and can be deployed on NoSQL or traditional SQL databases
- Leverages innovative and best-of-breed technologies such as Elasticsearch, containerization and AI
- Provides a low-code development capability to drive rapid prototyping and delivery of customized, user-driven applications
- Provides REST API capabilities for both consuming and providing microservices
- Is a true open source solution
- Features multitiered security controls and capabilities

These architectural underpinnings combine to deliver a modern CSP that delivers a full set of features and capabilities, including:

- System security and authentication
- User management
- Multiple, customizable user interfaces (including mobile)
- Native content viewers
- Flexible and scalable metadata schema
- Indexing and search
- Versioning
- Audit logs
- Workflow engine
- Connectivity to numerous repositories
- Records and retention management and much more

Conclusion

This article has highlighted some of the significant architectural elements available to software developers and architects today. It has also explored how Nuxeo Platform has taken these separate elements and combined them to create a uniquely modern, scalable and flexible content services platform.

Nuxeo Platform:

- Combines the latest technology components to deliver a full set of content services, such as: Workflow management, integration, document management, records management, case management, mobile and tablet access, knowledge management, digital asset management and more
- Provides comprehensive extensibility via APIs and microservices for both consumption and delivery of content services
- Integrates with existing systems to enable access to information residing in native solutions via content federation and/or migration
- Provides security and audit tracking for all content and data
- Includes cloud-native capabilities to deliver scalability, ease of deployment and flexibility — all on commodity hardware
- Enables organizational agility by being as dynamic and flexible as your business

Nuxeo Platform has been developed by technologists and system architects, for technologists and system architects. **Designed to save you time, money and headaches — Nuxeo Platform really is the CSP you would have built yourself.**

Hyland™