

contentgrid

Beyond content chaos,  
**clarity** with entities

# Start your path to clarity

**Problem statement:**

Modern content management systems (CMS) often replicate outdated, paper-based organizational structures. Leading to what's termed "content chaos". This chaos is worsened by using fragmented systems like SharePoint, shared drives, and email systems. Which leaves complicated governance, compliance, and clarity behind for the sake of individual productivity and small-scale collaboration.

**Innovative solution:**

ContentGrid is a new way of managing content that uses entities and relationships instead of folder-based systems. This model allows for a more intuitive and scalable way to organize content that aligns closely with business processes and needs.

**Technological advancements:**

The platform provides a sophisticated contextual model that extends beyond mere metadata to encompass relationships among entities, significantly enhancing the management and retrieval of information.

**Security and compliance:**

The platform is designed to meet high security standards, including ISO 27001 and GDPR, through a new model of access control that uses attributes and policies rather than roles. This approach makes governance easier and improves security.

**Business benefits:**

ContentGrid simplifies content management and improves the digital experience for customers and employees alike. It promises significant cost control through cloud migration, as well as architectural benefits like orchestration and scale-to-zero capabilities.

**Future vision:**

The white paper outlines a future where ContentGrid can facilitate more collaborative and intelligent content management practices, thereby enabling enterprises to remain ahead in a rapidly evolving digital landscape. The platform combines cutting-edge AI and database technologies to make content management more organized, secure, and efficient.

# Content Chaos, the natural state of information

Putting content under collective control is a growing challenge. With the adoption of cloud content services, the number of content systems in organizations continues to increase, including shared drives, sites, and email systems. With the migration of content to Sharepoint, small-scale collaboration is empowered. But governance, compliance, and collective clarity are still lagging behind. Where can I find my document? What were our agreements back then? Content chaos thrives.

The reality of agility is that it is difficult to adapt content systems to a changing business environment. Running document archives and team collaboration tools eats into tight IT budgets. Regrettably, the primary innovations that we introduce into ECM and content services are merely enhancements. Nobody, except for startups, has sufficient time to leverage generative AI and exciting new technologies.

On the other hand, your employees and customers expect to access their content 24/7 on demand, enjoy fully digitized content centric processes, and rigorous protection of their confidential and private information.

Therefore, we need a new way to organize our content that goes beyond websites, folders, types, and groups. Duplicating, tripling, or multiplying your customer's info in content metadata is a red flag for your content platform.

# Unlock Clarity!

In the past decades, there haven't been any really innovative approaches to context modeling practices. New and established players have recycled old concepts like folder structures, document types, and categories. They invented new weapons of mass distraction, such as sites and drives. Although useful for short-term, small team collaboration scenarios, they are not helpful for long-term collective information management. When I helped roll out Documentum at BASF in 1999, it had the same organizational principles as most ECM's today. Around 2005, open-source ECM software, like Alfresco and Nuxeo, became more flexible. Introducing aspects and the use of NoSQL databases to scale up expanding content repositories in document models. But we didn't see any significant changes in ECM.

Since then, we've noticed a difference between managing content with SharePoint or Microsoft 365 and the rest (the archive or repository). Online collaboration is enhanced by the ability to edit a document in parallel. The plates of collaboration, transactional content services, and records management have gotten separated.

Today, we think a new way of sharing content is needed to solve the chaos. The problem is, first and foremost, too much information, and more folders won't help. The way we model the context should be reconsidered to keep content. Using ideas from other fields can help us a lot.

Most 'creative' content produced by (collaborating) humans becomes static after a short, intense, and creative period. Consider a sales proposal, an R&D design, product documentation, commercial presentations, and other similar documents. After approval and exchange, the document is cast in stone and considered a record.



# Navigating the content space is easy with ContentGrid

Where folders and types fall short, we believe that entities and relationships will succeed. We need an intuitive context model with entities and relationships to tie it all together into a simple content map of the organization.

The way we store information should match your business as closely as possible. Entities and relationships accomplish precisely that.

## Folders proliferate content chaos

Folders have remained a fundamental organizational principle to this day.

They are used improperly to host access control structures, facilitating inheritance of Access Control Lists over large tree structures. Chaos is imminent while auditing is neglected. The use of folders as a foundation model to establish context for your content is a viable approach for small-scale, individual collaboration, however, it is not suitable for large-scale corporate repositories.

- Folders have a one-to-many (one folder contains many documents) conceptual model. This is simply too simple for the real world.
- Folders are flat, indiscriminating, and non-semantic. A folder looks a lot like another, and there isn't much information about how to use a document collection.
- Unless “contain”, folders have no direct connection to the contents they hold. We have no formal rules for how documents are “folded”.

Many times, folders are created without central governance, in an ad hoc fashion. This approach works well for short-term collaboration, such as creating a new proposal, product presentation, or brainstorming a new design. However, it leads to increasing chaos in long-term document repositories. Nobody cleans up because we fired the librarian decades ago. It was worsened by sites and drives. The enterprise repository has one root node, so you know where to start. In collaboration environments, we face multiple drives, sites, and teams that require content to be scattered across more context.

# Document types are fuzzy

Content is very heterogeneous by nature. A document is a snapshot of information in time. If you agree with this definition, it's no surprise that typifying documents is hard. Unfortunately, classes (types) do not align very well with the heterogeneous nature of many documents. They do match well with transactional documents, such as a proposal or purchase order. Because anyone can add anything they want. Of course, we have a set of transactional documents (a quote, a contract, an invoice, a deliverable, ...) that are closely related to well-defined processes. They align well with types. However, there is also the rest. Documents that are of type "document".

A collection of groundbreaking research titled "Researchers at the University of Cambridge propose Anchor AL: a unique machine learning method for active learning in unbalanced classification tasks" shows this problem of unbalanced classification. It's human to want AI to clean up the mess, but it's better to avoid it. Common sense is not outdated yet.

All of these rest documents are tied to rights that matter to your business. The ContentGrid platform allows for a lot of different classification combinations without the need for manual classification. However, it provides the opportunity to categorize a document into a specific type, such as a quote, with an amount, a product, and a customer, by introducing associations.

We had a problem in the past where we couldn't use multiple inheritance because we couldn't make a document both a project document and a meeting note. Thankfully, Alfresco introduced aspects to model this challenge. ContentGrid takes it a step further by reinstating entities and relationships as the optimal modeling mechanism, given its extensive and mature toolset.

Have we already mentioned our fondness for postures? The ecosystem is 49 years old and stronger than ever. Expect Postgres-based technology to make waves in analytical user cases in the future. ContentGrid will happily integrate to make the search experience even better.

# ContentGrid talks and thinks business

Let insurance experts look for contracts, claims and annexes, policies and customers, and from customers, back to all the documents about a customer. Their business applications do so already. Let HR and labor services talk about candidates, CV's, employment and skill assessments, and the relationships between them. Allow a financial institution to discuss accounts, investments, loans, stocks, bonds, and interest. Not only the entities, but also their relationships. When content leaves the process and gets classified, this valuable context is lost and unnecessary with a scalable corporate content memory.

All we need to construct a corporate content memory are these concepts and their relationships, skillfully reduced to the bare minimum context. All documents fall within their natural context. Your idea about the ideal minimal model will evolve as your platform evolves. It should facilitate, not sabotage, due to costs or risks. ContentGrid would typically be seen as a long-term memory and transactional service underlying a collaborative environment like Office/Microsoft 365 SharePoint. In the genesis phase of content, Microsoft 365 allows limiting the number of actors to a small group, which requires flexibility and concurrent access. The content is eventually placed in the long-term memory grid, ContentGrid.

## 1. Context for content: the metamodel

Any content service implements basic CRUD operations and provides a context for the content under its management. The legacy system relies on folders and paths to create that context. The document's context is basically the answer to the question: "where is the document?". Even though metadata can be attached to documents in folders, it's often not needed because the path doesn't provide much information. Guess what happens to non-required metadata? People look for the path of least resistance...

The search experience will never be the same: vector/similarity search and the sweet assistance of LLMs will start speaking the language of the user, customer, and human trying to master digital complexity. Picture your oldest customer. Image him or her using your system. We are not yet there, but we cannot rest until we are inclusive of people who do not speak Python as their second language.

With ContentGrid, context is more rich. We give a piece of content context by classifying it into an entity with attributes, as do other content services, but we also add the capability of associating content to other entities in your model. An incoming order comes from a customer. The order has some metadata (the product, quantity, price, PO number) and it has a relationship with the customer who ordered it.



## 2. ChatGPT for ContentGrid: let's get relational

Of course, I asked ChatGPT to make me a small ERD to show the model, and please note the “document” attribute of the order in the screenshot below, which is the PDF carrying the actual document. ChatGPT even got the number right: a customer can place multiple orders (fortunately).

```

-----
| Customer|      | Order |
|-----|      |-----|
| CustomerID (PK) | OrderID (PK) |
| Name           | CustomerID (FK)|
| Email          | OrderDate      |
| Phone          | TotalAmount    |
| Address        | Status         |
|               | Document       |
|-----|      |-----|

```

Continuing our little excursion, ChatGPT actually grasps why ContentGrid is the preferred method for arranging content. Let us trigger a next step:

**You**  
chatgpt, you just made me a relational model, about customers and orders. Could you try to make me a folder structure to model the fact that I could store orders for a particular customer?

**ChatGPT**  
Sure! You can organize the folder structure to model the storage of orders for each customer. Here's one way you could structure it:

```

markdown
Copy code

Customers
- Customer_ID_1
  - Orders
    - Order_ID_1
      - Order_Details (e.g., Order document, invoice, etc.)
    - Order_ID_2
      - Order_Details
    ...
- Customer_ID_2
  - Orders
    - Order_ID_1
      - Order_Details
    - Order_ID_2
      - Order_Details
    ...
...

```



There is no Nobel-prize winning insight here: a relational model is more expressive than a tree-based model. In 1970, Edgar F. Codd introduced the notion of relational databases, which brought forth the advantages of normalization. Since then, the relation model and the database technology that supports it have dealt with UML constructs, key-values, vectors, graphs, JSON structures, etc., and will in the coming years combine extreme transactional workloads with analytical capabilities. Long story short: Postgres and relational models are a safe bet for a long time.

Context doesn't have to be a vague concept. Consider customers, projects, renewals, and proposals. That is the context in which content lives. In relation to other concepts, either single or multiple. This model, which has fueled over a million applications worldwide and spawned numerous database systems, is the model that is most effective for content. It provides a context within which we can manage the lifecycle of our documents.

Let's take it one step further. I asked ChatGPT to integrate the concept of a product into the customer and order model. It gets messier.



ChatGPT

Apologies for the oversight. Let's correct that. Here's an updated tree-like structure that captures the relationship between customers, orders, and products:

```
markdown
Copy code

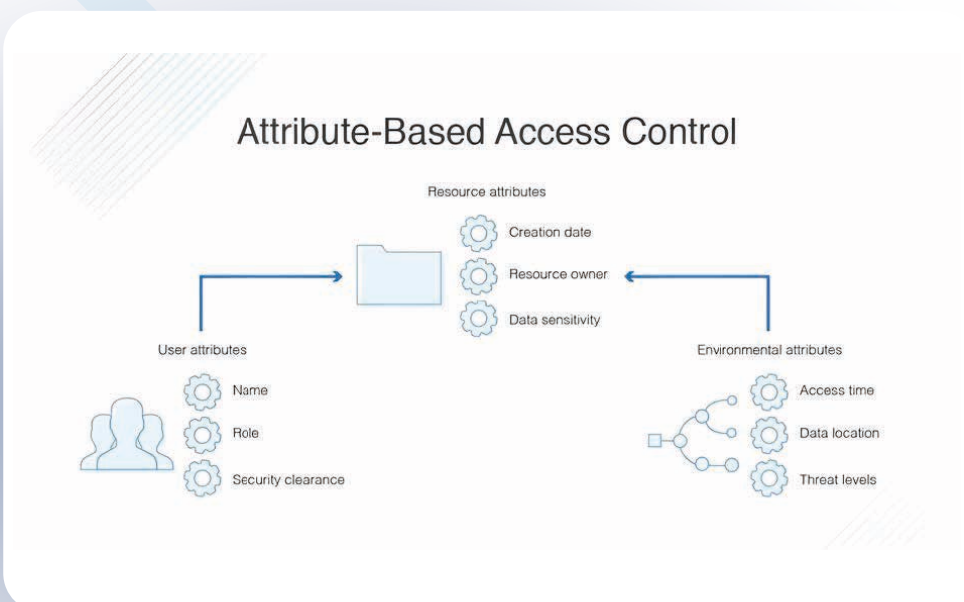
- Customers
- Customer_ID_1
- Orders
- Order_ID_1
- Products
- Product_ID_1
- Product_Details (e.g., Product information, specifications, etc.)
- Product_ID_2
- Product_Details
...
- Order_Details (e.g., Order document, invoice, etc.)
- Order_ID_2
- Products
- Product_ID_3
- Product_Details
- Product_ID_4
- Product_Details
...
- Order_Details
...
```

I guess the point is made. Folders don't scale cognitively. Neither technically: putting 10 million documents in folders that should contain 1000 children at most, leads already to 100.000 folders. Just for the sake of them. Just for access control inheritance. It is likely that using them at scale will create deadlocks in your database, even if you have purchased the expensive giga-edition.

### 3. Transformative

Current ECM's are identical to the paper world. The concept of how we create context for our content is shadowed by the physical organization we have adhered to since the invention of printing.

We refer to ContentGrid as a transformative content service platform because we transform the way context is created to support the full life cycle of content. In this change, we had to create new ways of controlling access by using attributes instead. By virtue of this “metamodel”, content can be retrieved and at the same time protected from unauthorized access.



We must consolidate content repositories, move to the cloud, and leverage new architectural benefits (orchestration, scale to zero) in order to control our costs. We need to improve our customers digital experience and comply with ISO 27001 and EU regulation for GDPR and qualified digital archives.

### How does it work?

After the analyst has defined the model (blueprint), ContentGrid generates the app (spring boot does the job behind the scenes). This app is unique to the customer and is made using her data and access model. The result is a generated API, application code, and a generic user interface that comprehends the model. When the model is changed, the app is re-generated, the conversion scripts are generated, and the app is updated without any downtime and with maximum preservation of backwards compatibility. Pilots are built in hours.

# Simpler, safer, smarter and scalable!

The essence of our business value has been condensed into four words that begin with the letter S. And maybe it's fair to say that there's a single S buried beneath the rest. Semantic: the semantics conveyed by your entities and relationships.

*We define the metamodel as the entities, attributes, and relationships necessary to manage the life-cycle of content. All are based on one unified and business-centric minimal model. This model is used by organizations to find, protect, and retain content in compliance with all relevant regulations.*

## Simpler

Make the system work for your business, not vice versa. The Simple in ContentGrid means that the website is intuitive to use. It is simple for users, analysts, and architects, but not for the vendor. We can make it simple, even if your business is Volatile Uncertain Complex and Ambiguous. Trying to reduce it to simple concepts like folders moves the complexity from the vendor to the user and the analyst.

ContentGrid only requires a relationship model to model context and control access. It facilitates a seamless progression of your model over time. Yes, we try to be backwards compatible, but at the same time forward-looking. We can make things as simple as possible by using proven concepts that are precise and complete to describe our business. Entities and relationships are a perfect fit to model the unique context for your business, as in any ERP, CRM or business application.

*A document is a short snapshot of information. In ContentGrid, documents are objects with binary content. Key business entities, such as the client, product, and supplier, are closely associated with them.*

## Safer

Access control is broken, says OWASP. We're all for OWASP's broken access control, but we're particularly worried about the current ECM systems. Role Based Access Control is often used in traditional systems, but the role explosion lurks in the shadows. Even though old-school role-based access control is too complicated for content protection, difficult to audit, and challenging to manage, vendors stayed compatible with role-based access control.

ContentGrid's decision to base the core contextual model for content on entities and relationships implies a modernization of access control towards attribute or policy-based access control. Changing access control is a delightful side effect of building the context for your content around entities and relationships. NIST has endorsed PBAC as the next-generation access control (Attribute-Based Access Control | CSRC). You can create any access control rule using the properties of the user (group memberships, attributes in AD) and the attributes of the entities that populate your business model to express fine-grained access control rules.

- Safer network policies are required (cilium based)
- Compliance with regulations (digital act, AI act? GDPR, NIS2, ...)
- ISO 27001 compliance, ensuring confidentiality, integrity, and accessibility

*Key use case: imagine that you store emails within your content app. You can match the current user with the To, From, or CC of the email, so anyone who is mentioned in the To, From, or CC can see the email. CC can see the email, with a single rule.*

In the introduction of this white paper, we mentioned the ad hoc basis upon which sharing of content is handled. In ContentGrid, this is handled differently. Remember that everything is metadata, and metadata drives all access control. As long as the metadata is correct, and the document is associated with the correct entities (customer, partner, supplier), sharing will be a matter of setting the attributes correctly and updating the access rules.

Furthermore, we have the capability to audit modifications in access control or automate sharing based on the metadata of the document. Sharing content requires setting metadata accordingly, and metadata changes can be audited. Sharing links, like in Sharepoint, is complex. Being able to share a link becomes another right to manage. When your external partners want to access a document in ContentGrid, they can use a social ID. This allows you to control and audit external access. Every access to content should be under a personal account, which can be validated and audited. Adding a watermark like "Hi Pierre, thank you for accessing this document on 9/3/2024" would help to encourage professional care and confidentiality while accessing the content.

## Scalable

ContentGrid scales in three dimensions: technology, cognition, and economically. Note that economic scaling means getting cheaper, not more expensive!

Scaling starts with simplicity. Customers should only get the resources they need for data structures and functions. Invoices, contracts, and proposals in ContentGrid require just one table per type of document. It goes without saying how much simpler it is to scale such a model compared to the “one model fits all” abstractions in common ECM systems, leading to very complex models. ContentGrid has invested into the necessary automation to modify a data model when required by the analyst, to gain efficiency in the run-time environment. Doing so allowed us to remove all the constructs necessary to keep the data model generic. The data model at CG is specific and straightforward. We believe that the overall performance gains and computing economies in run-time greatly outweigh the extra investment we made in automating model conversions.

Due to our decision to build upon the proven relational model, we were able to leverage exceptional tools such as Flyway on top of the world-class Postgres. Searches involving multiple attributes within an entity, such as all invoices sent to customer ACME in the year 2022, are executed on a single table and can utilize only one index. With modest hardware, this can handle billions and billions of entities. The simple data model that CG generates can easily be partitioned and/or sharded if we hit a scaling limit after billions of documents. Remember, any document entity can be contained in one table. With complex models that are designed to accommodate all customer cases at runtime, this becomes an engineering nightmare. With CG, it's simple.

Simplifying the structure to a basic relational database, ContentGrid greatly reduces the cloud resources needed to handle massive repositories. The run-time environment needs a number of Kubernetes pods that run the API and the general GUI. Combine this with a Postgres database, and you have all that you need to run your app, handling anything from a million to billions of documents, and storing the binaries in S3. Kubernetes takes care of high availability and scales automatically.



## Smarter

Today, all content service providers have an AI strategy and can incorporate some AI functionality into the content value chain. ContentGrid is a natural ally for AI, thanks to its real-world model with entities and relationships. The process of extracting entities, properties, and relationships is a discipline as old as artificial intelligence (NER). It has been reinvented with transformer models. BERT by Google is a pioneer, the GPT model is improving, and Hugging Face is a treasure trove of transformer models. The metamodel, which is based on entities and relationships, makes ContentGrid AI first.

Every day, new ways AI can change the way we handle content come to mind. ContentGrid is AI-friendly because its metamodel, which is basically an ERD diagram with entities and relationships, is so common, well-studied, and automated that AI can make your blueprint automatically. We can demonstrate a proof of concept just doing so.

Nonetheless, there are cases that hold greater significance than the creation of the blueprint of your CG application. This is a type of art form that is done only a few times a year or, in the case of a highly dynamic enterprise, every month. The number of requests and queries in daily usage dwarfs this.

AI can help in the creation or ingestion of new content. It can classify a document or content item, figure out its characteristics (for example, this document is a CV and the person applying for it is called Hillary Trump, who is 54 years old and knows ancient martial arts and Mediterranean cooking). It can also determine how the documents relate to other important things in your CG app. You often have master data in ERP, CRM, or your core business applications. We all have customers, products, and other information stored in accounting systems that ContentGrid can either mirror or import using foreign data wrappers. Extracting these entities in documents becomes a no-brainer using such master data.

Once content is classified, the system can help retrieve the document by interpreting your question (“you typed 2000, I assume you are searching for candidates born in that year”), leveraging vector search (“you searched for Trump, did you intend to search for a presidential candidate?”), and accepting natural language as a query format. “Hey ContentGrid, could you retrieve all the records from the congressional library, containing speeches of presidential candidate Trump in 2019”?

Due to the relational model that underpins all content, ContentGrid is capable of producing such lists for your organization, providing additional insights or historical retrievals. “Who were the customers known to our organization up to December 2016”? Expect PostgreSQL and ContentGrid to become full-blown analytical platforms in the years to come. Activate your archive.



# Cloud-native, open-source, and without a license key

We were astonished to discover that there is no relevant open-source content platform on the market anymore. The reason is simple: technology companies did not work out fair value sharing for the community, including the users. It's hard to say if an open-source community for ContentGrid will be successful. We can be sure that the service will be free for all to inspect, improve, and run in conformity with the agreed service-level objectives (SLO). The result runs on open-source components, is spring-based, and is PostgreSQL oriented.

The Cloud Native Linux Foundation understands the challenges and threats that new, cloud-native technology has to face. ContentGrid adopts:

- Kubernetes serves as an orchestration instrument, bringing self-healing and auto-scaling into our reach
- Versatile network control tools like cilium enhance network level security (zero trust)
- Content encryption via application, beyond storage-based encryption
- Rigorous authentication chains are used when linking cloud services (no content leaks)



# Value-based pricing in SaaS and Bring your own cloud

## Usage based

Pricing will be per use, not per user. We will move toward value-based pricing while we better understand the value of content (more sales, satisfied customers, break-through research, automation of knowledge intensive work, making proposals, etc. Value-based pricing allows you to divide the value fairly between all stakeholders. In the coming years, measuring value should become a subject of research. ContentGrid uses the three value drivers as a proxy for values:

- Number of documents retained
- Number of documents injected per month or year
- The number of queries that returned within three seconds

## Value drivers

We will find better value drivers. Someday, we will move to pricing schemes that reflect the time we save, the cost we avoid, and the new business we help develop because of the automatic generation of sales proposals, product documentation, quotes, etc.

## SaaS

Software becomes a service. You buy availability, performance, continuous and automatic upgrades to improve security and reduce cost. ContentGrid comes with hard and quantified commitments to availability (99.9%), performance (ingestion of 100 documents per second, 99% of queries return end-to-end results within 3 seconds) and durability (we guarantee 99.999999999% durability). The same as S3, but this for the metadata (S3) and the binary content (Postgres).

## No vendor lock-in

There is no alternative to open source if one desires to maximize their independence from their SaaS provider. There must be an exit strategy, and even more advantageous, a strategy to transfer the service completely under your control if you so desire. You should also not be locked into a cloud.

## Part of the solution

The applications will be unique to each customer. The days of the monolith, with its bewildering array of obscure and highly dependent parameters and configuration, are over. Many systems have a lot of parameters because the software vendor doesn't know how to set them up for your case. We envision a platform in which you build your app from your model, and all other configurations are part of the solution.

# Maturity is more than technology

Technology alone will not solve the issue of content chaos.  
A method and metrics are needed.

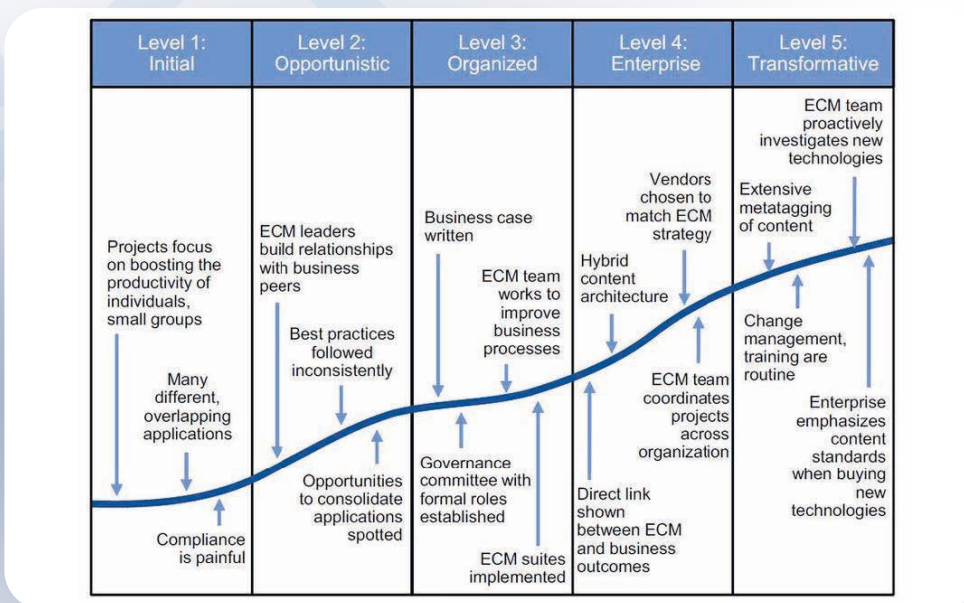
We believe that a solid approach can be built around four major steps:

- Switch to a simpler architecture
  - Scalability
  - Consolidating ECM systems and giving your business users one way to access content
- Promote entities that are involved in document types and improve
  - Data quality (metadata is correct and complete)
  - Search experience
  - Compactness of your data and document
- Automate to eliminate manual, repetitive and mind-numbing work
- Active content assets that become drivers of business success and a great customer experience

ContentGrid helps organizations reach higher levels of information maturity. Frequently, this is characterized by a reduced number of systems (scalable), increased control (audit, policy-based access control), predictability, resilience, and agility. If your systems are difficult to modify, upgrade, or extend, you have reached a plateau of maturity and an upgrade will not help you get out. Allow us to highlight a few important indicators of content maturity and suggest how ContentGrid can help organizations move up the ladder.

Maturity Level	ContentGrid	Phase
Organizations clearly identify the use cases of content and design an appropriate architecture, "understand how content powers the business and limits the risks"	Use business concepts to model the right context for your business	Migrate & consolidate (ContentGrid scales to billions!)
Adequate contextual model		Elevate
Corporate content adapts to new business environments. Automation of micro-processes		Automate & integrate (With microsoft 365)
Dynamic routing and content processes that run semi-autonomously		Activate

ContentGrid is a transformative platform (that can move organizations step by step until they have a top-notch content strategy), and the Gartner quadrant for ECM maturity can help us along the way.

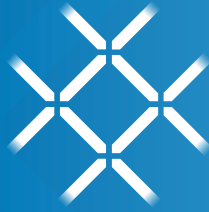


## Unlock clarity in Content Chaos

ContentGrid alone can't bring order to content chaos, but we think it's the best way to help our clients, partners, and co-creators understand their content processes and repositories, and rethink how they create context for content.

Finally, ContentGrid will allow customers to learn together without compromising privacy or confidentiality. This exciting new development, captured in the research domain of federated learning, will allow organizations to work together in domains that they gain from collaboration. All insurance companies benefit from detecting fraud, so why not use AI to learn together to make the world a better place? If HR companies or divisions of an HR group collaborate to enhance the matching of candidates and job offerings, we can enhance the business.

If ContentGrid can contribute to a greener world, but a simple and resource-conscious architecture, and maybe help to federated learning, we can support the only way to tackle the big challenges our planet faces in the future: by working together.



contentgrid

# Beyond content chaos, **clarity** with entities

## Join us in the discussion

Come join the conversation in the ContentGrid lab.  
Email [lab@ContentGrid.com](mailto:lab@ContentGrid.com), to reach out to us.

Perhaps ContentGrid is the transformative technology that you are seeking. But rest assured that, regardless of any product or service, thinking and learning together will advance content services and how we turn information into insight and experience.