CMaaS - Content Management as a Service

an outlook on future Content Management Systems and Services

Authors:
Cataldo Mega, IBM Deutschland Research & Development GmbH,
Kathleen Krebs, University Hamburg
Frank Wagner, University Stuttgart
Norbert Ritter, University Stuttgart
Berndhard Mitschang, University Stuttgart

1 Introduction

The constantly progressing penetration of IT in the private and in particular within business areas produces an immense and rapidly rising flood of data in which Rich Media represents the predominant portion. The vision of the paperless office publicized many years ago is delayed again and again, due to the enormous volume of data and the nontrivial technical hurdles to overcome when it comes to manage it. This is the dilemma that Content Management Systems today are struggling with. Today we can foresee that next generation Enterprise Content Management (ECM) systems will be dominated by effort to virtualize the content repository interface at all layers. The virtual electronic archives (e Archive) of the future will be built out of basic content services (components) individually composed to a custom ECM solution for each respective customer or better each individual tenant. Thus the ECM System of the future must cope with the request to support massive multi-tenancy. As a consequence the underlying IT-infrastructure must provide also massive scalability at an affordable price. Both will become possible thanks to newest Grid and Dynamic Workload Management (dyn-WLM) technology. New trends in IT Hardware technology indicates, that these goals can only be achieved by means of exploiting scale-out on commodity hardware, configured dynamically and provisioned on-demand. Thus we can say Content Management is provided as a Service – CMaaS. The future alternative to the normal way of purchasing and installing a CM product will be the 'on-line offer' of CM as a service based on Service Level Agreements (SLA) and only for the duration of service consumption. Enterprises must recognize and address requirements for these new trends and adjust accordingly. Themes like legal compliance, electronic discovery, and document retention management will still remain but must be adopted to the new environment and new class of users.

Today what we have are ECM systems capable to process very high document ingest rates, that support distributed full text indexing, and allow forensic search such to support litigation cases. Tomorrow all of the above must be provided at lowest cost with respect to archive management and administration to potentially a large population of millions of users organized in communities of interest. One approach is to introduce a virtualized ECM system interface where the key content repository components are wrapped into a set of tightly coupled stateful service entities, such to achieve scale-out on a cluster of commodity hardware that is automatically configured and dynamically provisioned. By doing so we believe, one can leverage the strength of Relational Database Management Systems (RDBMS) and Full Text Search Engines (FT) in a managed clustered environment with minimal operational overhead.

2 Enterprise Content Management Concept

Today one can choose from many commercial electronic archiving solutions that handle unstructured content [Gartner]. In addition to the commercial systems we must also consider open source technology that is utilized and developed. P2P[33] file sharing systems like BitTorrent[1] and Gnutella in a sense can also be seen as specialized content management systems. The latter focus on one specific job, i.e. file sharing, and that job is done very well. What is remarkable is their scale-out characteristics and automatic topology reconfiguration capability.

Interestingly enough, some earlier P2P implementations of Gnutella had scaling problems because during query each node flooded its query requests to all peers. The demand on each peer would increase in proportion to the total number of peers, quickly overrunning the peer's limited capacity. More recent P2P systems like BitTorrent scale well because of the lessons learned. Here demand on each peer is independent of the total number of peers. In addition, there is no centralized bottleneck, so the system may expand indefinitely (scale-out) without the addition of supporting resources other than the peers themselves. In the event of peers joining or leaving the configuration, the system is also able to re-configure itself in an automatic fashion.

Now, despite of all valuable features available in P2P systems, they do lack some important capabilities that are mandatory for ECM systems. For example: There is virtually no content management framework with an underlying repository infrastructure and application integration layers. Content management is reduced mostly to filesystem semantics. Support for content retrieval is purely via hash key value lookup. They lack the ability to allow content retrieval via meta data search. The latter is a service that implies the existence of a document data model. If wanted, this service must be added by some other means, such as: a catalog database or a full text engine. For this purpose many do use the Google MapReduce programming model and the Hadoop platform[2] from the Apache Lucene project. Both approaches strive for distributed data processing around the construction and management of cluster of full text indexes. But in the end, Peer-to-Peer lacks important storage management functions, a capability that is pivotal to ECM systems and fundamental to satisfy hierarchical storage management needs. Never the less this is the area were ECM system can learn from P2P systems.

Traditionally, the most common design approach adopted in ECM systems to cope with scale was a scale-up approach on large multi-processor systems. Measurements [14][15][23] done on clusters and grid systems indicate that scale-out might be more cost effective and affordable if administration and maintenance overhead can be kept to a minimum. Given that P2P systems do expose this capability one approach is to merge the best of both worlds.

---

1 http://www.bittorrent.com

2 http://lucene.apache.org/hadoop/

As always, in order to better understand how to refactor current ECM systems we must collect and understand the future functional and non-functional requirements from on-line communities and detail their typical activities and data flows. The goal is to define a representative workload model which can be used to sketch new ECM environments.

2.1 ECM System Architecture and Design

ECM systems are meant to help manage unstructured information and data and their related work processes. They are concerned with the routing of in- and out-bound information flows and managing the user's workspace. In this context, information is indexed and classified for later post-processing.

The next Figure outlines the architecture of an ECM system.

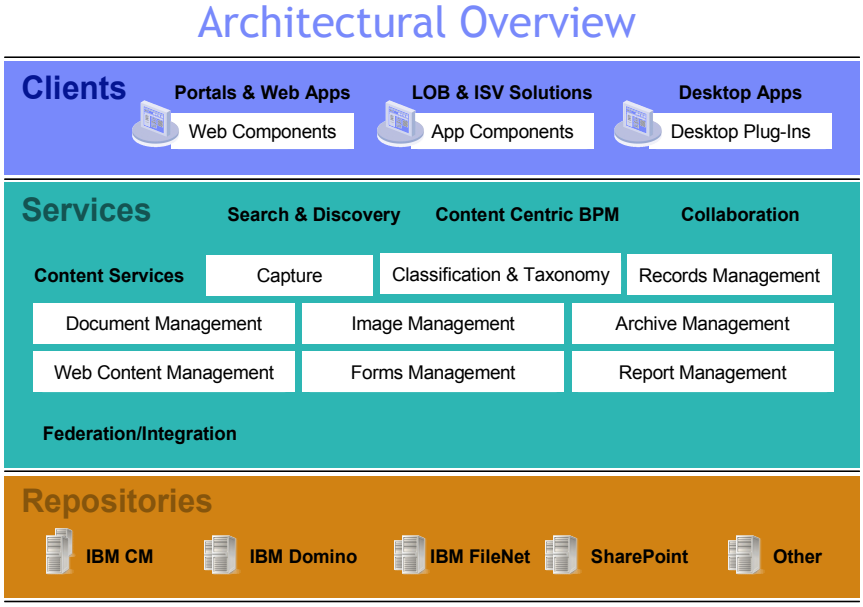

Figure 1: Enterprise Content Management System Architecture Overview

The layer in the middle represents the central part and actually defines an ECM system. This layer addresses the business needs of enterprises to deal with their unstructured data assets. It consists of: 1) Basic Content Services, 2) Search&Discovery, 3) Business Process Management (BPM) and 4) Collaboration Services. The main theme is to manage the content lifecycle from creation to disposition. The actual repository is the core of this architecture, shown at the bottom of the figure below. The repository is accessed via the the so called 'Repository Abstraction Layer' (RAL). The RAL ensures that the access to the actual physical repository is seamless, thus all upper layer services become agnostic with respect to the documents physical storage location. If more than one physical repository is accessed then the RAL also represents the Federation and Integration layer. Studies show that, due to legacy reasons,

companies in general have more than one repository in production. That is why federation and integration is key to every ECM infrastructure.

The top most, upper layer is the Clients Application Layer where end user access is facilitated via custom UI or off-the-shelf business applications.

In summary, the core constituents of ECM systems are:

1) The repository and storage systems that are responsible for storing and managing the actual electronic content in its original and other derived formats, so called renditions. In the core, the system catalog is included, that stores and manages the meta data, based on an optimized data model. Search support is realized via relation, full text search and XML query services, at least.
2) Basic Content Services handle and manage the document specific semantic. The information discovery services, the so called e-Discovery functions, are needed for supporting compliance needs in companies. Given the size of the collections archived, e-Discovery must be tailored and tuned towards the ability of handling of very large result lists produced by compliance investigations and its related forensic queries.
3) And finally, a business process management framework that allows the orchestration of individual business process flows.

In this context, electronic archive management (EAM) can be seen as the central service that a Content Management system must provide such to satisfy the specific needs of an enterprise to comply with **corporate governance.**

2.2 Electronic Archiving Management Systems

In the remainder of this paper we want to restrict our focus on electronic archive management as EAM specialized subclass of ECM systems, which lends itself much better to risk a prognosis to predict how ECM will be in the near future.

The key functional components of an EAM system are:

1) Content storage, archiving and indexing services
2) Retentions and Records Management services
3) Search & e-Discovery services

Content storage and archiving is tailored towards consistent management of the corporate data assets combined with storage optimization. Intelligent archiving is usually done automatically on a regular basis according to a set of predefined corporate rules.

Consistent and true content management becomes alive if documents are treated as business records, integrated in business processes and if corporate governance is enforced. This means that the relevant information contained in business related documents, messages, emails, chats, blogs, simple files and other new information artifacts, must be made accessible

throughout the enterprises to everyone with a business need and the right security and privacy constraints in place. In addition, today legislative regulations also mandate the preservation of the chain of custody, which means businesses must prove that requested information was preserved and can be produced in case of litigation within a finite time. For an ECM system, all these regulations must be translated into the set of archive rules, which then dictate the kind of long-term archiving adopted for every single record managed.

With such an electronic archive in place, e-Discovery becomes the framework targeted to support compliance officers to find and retrieve information when ordered to do so. Producing information in the context of compliance must be understood as producing 'all' relevant information not just a portion of it. A Google search like result list that is extrapolated to a guessed, possible exact number of hits is not feasible in this case. To satisfy a court order, 'all' relevant documents must be produced, possibly more but not less.

With the advent of Web 2.0 we are pushing the edge also for Enterprises with respect to the richness of collaboration and communication. Knowing this it is clear that the amount of information shared within and between the collaborating communities will increase, which translates into the need for new ways of dealing with this more complex and demanding environment.

2.3 Non-functional Requirements for EAM Systems

With the above said, we want to focus on the non-functional aspects derived for EAM systems. More specifically, the performance and scalability goals, which we want to formulate as follows:

"The *designing of an EAM system must put major focus on the performance and scale aspect such to efficiently handle an unknown and variable number of documents, ingested at highly variable rates by a possibly very large population of end users or principals*"

This goal can be translated into the following three key performance requirements for the EAM system:

1. Support high throughput and low response times under normal production conditions
2. Have the ability to dynamically adapt to the current load situation by acquiring additional resources or releasing unused ones when appropriate to the design goal to match the load variations and minimizing system power
3. Be reliable and available such to achieve the required business continuity goal

2.4 Typical Workloads of EAM Systems

The main purpose of an EAM system is to function as the information hub of business relevant unstructured data. Ideally, the Enterprise Content Management System tracks and manages all business content that enters, traverses and exits the enterprise. Email is a good example but other data types are becoming more dominant like: Instant Messaging (IM), Chats, Blogs and WiKis. With an ECM in place, data can be processed in a controlled, common way

independent of data type and location. This means: if necessary the information is captured, parsed, interpreted, and, if found relevant, finally archived in a format determined by specific business rules which is the core competency of an EAM system.

From an electronic archive workload perspective document ingest processing is the predominant macroscopic operation. At a finer, more granular level, the necessary processing steps during the ingestion and indexing of business data in to the repository are:

1. **Unload / Extract / Crawl:**
   The first step is to determine a document source. Then a document crawler is scheduled to connect to the chosen source server using appropriate credentials. Document extraction is then started according to set of predefined storage and archive rules.
2. **Identify:**
   From within the ingest work queue, unique hash IDs are generated using some sort of Hash algorithm like SHA-1 or SHA-256.
3. **De-duplicate:**
   Hash IDs are used to check if the document or message itself or part of it is already stored in the system. If the document is not already in the archive then it is archived. Document duplicates would normally not be archived.
4. **Classify:**
   Text classification is an important step in a compliance scenario. Documents with content that does not comply with the rules have to be intercepted. Typically a supervisor then decides if the document can be forwarded to its destination or needs to be sent back to its originator or sent to a supervisor for further actions.
5. **Decompose:**
   If a document is eligible for archiving then it is also full text indexed. For this purpose a pure text representation is created from header, body and attachments before it is passed to the full text index engine.
6. **Transcode:**
   In the case that a document contains rich media attachment, then their plain text is extracted by means of a text extraction step. For example, plain text would be extracted from PDF or from Microsoft Office documents.
7. **Tag/Annotate**:
   The so extracted plain text can now be further processed by linguistic methods used to augment the text information, before it is full text indexed and ready for e-Discovery.
8. **Transform&Archive:**
   Finally the document is stored in the archive in a format in order to provide 100% format fidelity with its original. Sometimes different renditions of the original document would also be created on a needed basis, for example: Word -> PDF/A, HTML, TIFF etc...

Besides document ingestion, the other important activities to mention are: Index & Search, Retrieval ,e-Discovery, Retention, and Disposal. In analogy to the ingest, all these other core EAM activities can as well be decomposed in finer intermediate steps. In the end, the actual workload pushed onto the system will be the sum of a mix of primitive archive operations.

2.4.1 EAM Workload Definition

At this point we can define the EAM workload based on the functional areas specified above. By workload, we mean a representative EAM workload which consists of a representative mix of primitive EAM operations derived from key use case scenarios and activity flows. A typical list of primitive operations exposed by EAM systems are: logon, logoff, create, retrieve,update and delete documents, folders and links as well as operations related to search and discovery of information within the repository. In general there would be a finite number of these primitive operations (OP1 ... OPn) that are executed in a specific order and with a predefined frequency. A 'Document Ingest Activity' performed by an insurance clerk might look like.

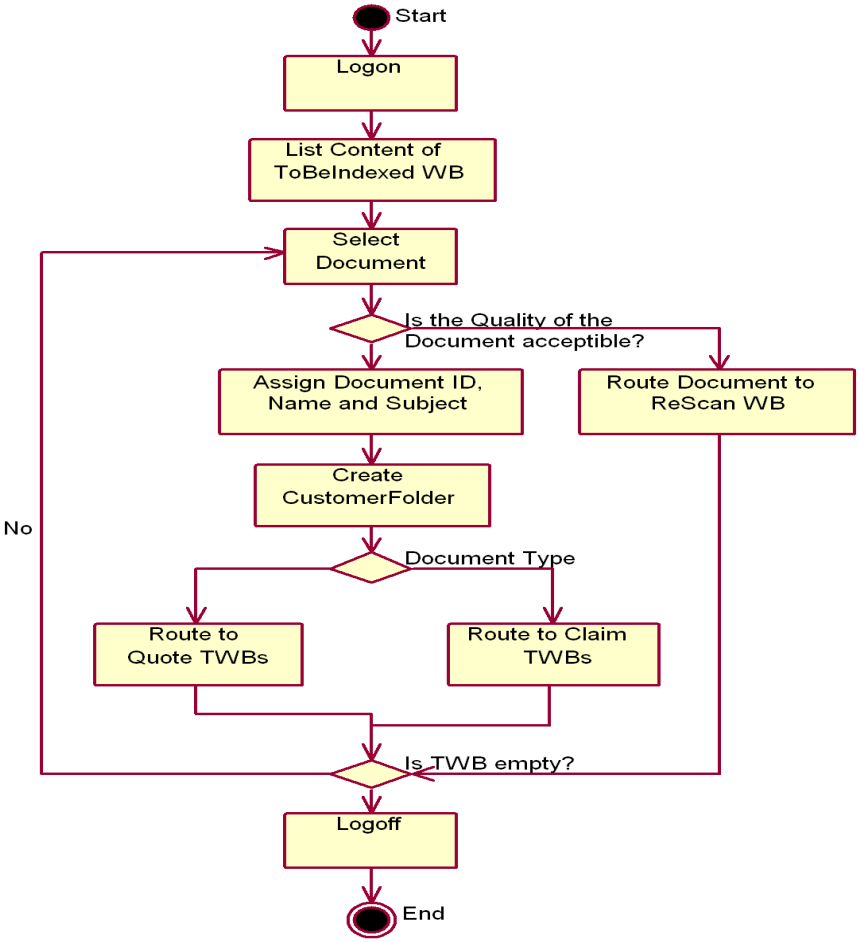

Figure 2: Document ingest activity and related operation

At a conceptional level we can now define the EAM workload in the following way: The overall system load generated by the main EAM activities i.e. ***Ingest, Search, Retrieval ,e-Discovery, Retention, Disposal*** can be synthesized into following definition:

- ***System_Load*** *is the* "*Sum of a representative mix of primitive archive operations OP1 … OPn issued against the repository interface layer with the given distribution as dictated by the respective activity.*

  *'EAM Work_Load****(Ingest, Search, Retrieval ,e-Discovery, Retention, Disposal)'***

As an example, in the case of a pure email archive system, the message ingest operations would contribute with with 80 % to the total workload. Whereas the the other operation related to: ***Search, Retrieval ,e-Discovery, Retention, Disposal respectively,*** would only contribute ~20% or less to the total workload. The exact set of weight factors for a specific EAM system in production would have to be determined empirically and adjusted over time in order to reflect change in application behavior.

Note: Due to retention rules, archived documents get deleted after a certain period of time(i.e. 7..12..90 years) at approximately the same rate at which they were originally ingested.

When analyzing this workload, one also finds another interesting fact with respect to the EAM data model. In almost all EAM solutions the document related meta data is very static. As a matter of fact, the portion of meta data that changes is small and is usually kept under version control logic. This means, archived documents usually do no:-) t change nor do they get updated very often, therefore the document meta data can be treated as immutable throughout the documents life cycle until its disposition. In practice, even deletion of archived documents is a rare operation, as companies most often hesitate to delete archived information because they fear to loose valuable information. As a consequence, given the amount of archived data to be considered, most documents will very likely never be retrieved either. In the end, many collections are basically managed until the end of the imposed retention period, before they then get disposed. [3]

It is because of these aspects that the design of some specific classes of EAM solutions can be optimized by mean of specialization. And the most prominent area to consider for specialization is around the ingest process, with discovery, and records management being second in importance.

[3] One trend that can be observed with on-line communities at hand, is the approach of granting access to these collections to on-line communties sort of special interst groups, be it internal or external.Then the idea is that in a collaborative effort these communities would mine the informaton and extract knowledge out of it, such to repurpose its use or create new business assets. see 32Peer-to-Patent: Community Patent Review

3 Evolving EAM Systems for future needs

3.1 Limitations in EAM Systems

Globalization and a 24/7/365 on-line system availability requirement to serve a world wide active user community, system load is increasing at much higher rates than expected. Demand for more load and flexibility in integrating new services is changing the way how traditional EAM systems are used and it is here that they are falling short. One way out of this dilemma is to evolve and enhance the system design continuously. In this chapter, we will first show some of the current limitations, then motivate how scale-out and services orientation paired with dynamic provisioning can help remove those limitation to better serve current and future needs.

When comparing the current document archiving and e-discovery workloads against the general purposes content management workloads one can clearly identify commonalities such as: storing and indexing content, as well as search and retrieval of the latter. Document life cycle management by means of Business Process Engines is a common shared task as well. Interestingly enough, the new upcoming on-line platforms like Yahoo BOSS, [33]and AMAZON EC2[31] for example, expose the same patterns. What is different with the new on-line platforms is the mix, sequence and weight with which the on-line processing steps are performed: i.e. document ingest, indexing, tagging, classification and text analytics is much more accentuated.

And the main design reasons for that are:

1. **Single central physical catalog.**
   An approach with a central physical catalog based on a pure relational model can only scale as far as the underlying RDBMS scales. Usually this system exposes scale-up characteristics. This means, the ECM system scales as far as the underlying single multi-processor machine scales. Using multi-threaded multi-core machines helps but does not remove the root cause.

2. **Serialization of processing flow.**
   A good example of process serialization happens when document unique Ids are generated. Usually the central catalog server of an ECM system monopolizes the creation of unique document identifier in the system. This means the catalog server when generating the requested UUID becomes also the gating factor of the system. The ingest process stalls if this component is too busy and can't catch up with the demand.

3. **Order of processing flow:**
   Wrong processing order of the individual ingest steps can be observed sometimes.

We have noticed that in many cases ECM systems archive the documents before further processing, and then in a second step they retrieve the already archived document back into memory for indexing, tagging and categorization. By doing so, too many copy operation are introduced and the archive becomes again a funnel that serializes the processing process.

4. **Mixed relational and full text data model.**
   The email data model is partially relational and partially full text search enabled. This leads to very expensive temporary join operations between both RDBMS and Full Text engine.

5. **Static deployment topology.**
   The physical EAM system is subject to a static deployment topology with a fix routing table.

6. **Static resource provisioning.**
   Fix resource provisioning and manual system installation and configuration hinders cost-effective provisioning.

It is with respect to all these services that one must satisfy the requirements of system scale, manageability at an affordable price. We believe that with scale-out on commodity hardware, and automatic re-configuration of components, these goals can be achieved. The way to boost and enhance general purpose EAM systems will come from mixing the best of open source and legacy technology including: Grid, P2P and dynamic provisioning techniques.

3.2 Scale in EAM Systems

3.2.1 Definition of Throughput

Throughput is the amount of work that a specific system is able to process in a given period of time. Its meaning is similar to that of capacity, and the two are often used as synonyms.

3.2.2 Definition of Scalability

Scalability is a non-functional system characteristic that reflects a number of intriguing issues. From literature it is known, the characteristics of scalability guarantees that the system behaves similarly under varying usage conditions like spikes in workload and managed data volumes. It is clear that a scalability analysis has to target system component and execution traces in order to uncover existing bottlenecks mostly in communication overhead, data sizes, and algorithmic performance. But before we do that let us define scale by using the following commonly accepted concepts and terms.

Scale can be seen as the cost function per unit of throughput produced.

Scalability, as a property of the CM system, is generally difficult to define and in any particular case it is necessary to define the specific requirements for scalability on those parameters

which allow characterizing the system under consideration. The key parameters to use to define scalability are:

1. **Load scalability:**
   The ability for a distributed system to easily expand and contract its resource pool to accommodate heavier or lighter loads.

2. **Administrative scalability:**
   Defined as the ease with which a system or component can be modified, added, or removed; to accommodate changing topologies such to enable a variable number of organizational units to easily share a logically single physical distributed system.

The latter is a typical requirement formulated by shared IT organizations in large enterprises.

From a technical stand point scalability is a significant issue for EAM systems, as it must co-ordinate and orchestrate many different components for example: the physical machine resources, networking, databases used, Web Application Servers and the actual archiving and indexing solution components. In our case the scalability of an EAM system is defined in following way:

> **Definition:**
> **"***An EAD system is said to be scalable if its workload throughput improves proportionally to the capacity of available system resources. More stringently it is said to scale if it is suitably* ***efficient****, sufficiently* ***easy*** *to administer and lastly also* ***affordable*** *when applied to ‚large' situations which means. a large input data set or large number of participating nodes in the case of a distributed clustered systems.*"

On the other side an ECM System is said not to scale if it fails to achieve these goals when the workload quantity increases. For our case the key metrics to measure scale are derived from a representative workload typical for ECM systems i.e.:

1. The number of documents archived per archive service instance used (store throughput).
2. The number of documents that are full text indexed per time unit per index processes in use (indexing throughput),
3. The response time of a search operation with respect to the number of concurrent users issuing queries (response time)
4. The aggregated cost per storage unit and lastly
5. The administrative cost of the IT infrastructure used per unit of work

3.2.3 Enhancing EAM Systems for Scala-Out

Scaling an EAM system can be done in two directions: vertically (scale-up) by using larger machines, and horizontally (scale-out) [45.] by adding more machines. Scale up is in our approach supported to some extend by the multi-threaded ingestion. A major objective is scale out with respect to the total system size i.e. the total amount of aggregated electronic content an EAM system can manage and the overall throughput performance characteristics with respect to ingest, index and query response times.

Next we will focus and discuss design factors that limit or enhance its system scalability. As listed before, we found that there were few but key factors that limit system scalability in a EAM system. It is for those factors that we have designed special enhancements for email archiving and e-discovery systems. Here is what we think will boost scalability.

1. **Avoid single central physical catalog** Aim at scale-out using a cluster of commodity machines instead of highly optimized large SMP systems. The central physical catalog is replaced by a central logical but physically distributed catalog, mapped onto a cluster of RMDBS and Full Text Indexes.

2. **Avoid processing queue serialization.** Generate document UUID from the document content itself using a SHA-1 or SHA-256 algorithm. By adopting this approach every ingest node can generate the document IDs independently, allowing scale-out to happen. [11.][18.][21.]

3. **Avoid wrong processing order:** Documents should be processed right after they are extracted from its source system. As they are loaded into memory and as soon as the decision to archive it is made. All other processing steps should be done in parallel. Redundant data copy operations must be avoided or minimized.

4. **Avoid mixed relational and full text data model.** Avoid joins between RDBMS and Full Text search engine at all costs. If necessary we issue a query twice by applying the 80/20 rule. This can be accomplished by splitting the ECM data model in a mutable (relational) and an immutable (Full Text) part.

5. **Avoid static deployment topology.**
   With dynamically changing operational boundaries one must introduce an automatic re-configurable deployment topology. This is best done using DHT [31.]based P2P technologies. [17.][28.][31.]

6. **Avoid static resource provisioning.**
   Introduce dynamic resource provisioning based on a particular application resource models and implement a SLA cost model. [24.]

3.2.4 Scalable EAM System Design

Armed with the set of best practices we can now evolve our approach in to an EAM system design that removes the scalability limitations by means of introducing a system wide scale-out capability. As you might recall, the core EAM components were the: a) System Catalogue, b) The Full Text Index service, and c) the storage subsystem. The magic on how to introduce a scale-out capability for the 3 EAM components is by means of abstraction and virtualization. This approach allows to introduce an indirection level creating a single logical entity to be distributed over a cluster of physical entities. The key technology use in our system is key space partitioning approach based on a DHT [31.] algorithm.

For example, the meta data stored in the system catalogue is crucial to all EAM operations. Its consistency and transactional integrity is very important, still instead of storing it in one single database one can store it in a cluster of IBM DB2 relational databases. Similarly, some of the meta data can be stored in a full text index (FT) managed by a text search engine. A FT engine provides an efficient and effective mechanism for searching on unstructured and semi-structured data and is therefore well suited for electronic documents retrieval. In our implementation we are using a cluster of Apache Lucene FT engines[1].

One problem when integrating structured and unstructured search is the necessary effort to join the results from both systems, and complexity to estimate this effort. This problem can be avoided by restricting searches on the immutable part of document meta data to the search engine only. The mutable part of the meta data instead is stored in RDMS. Usually, the amount of information stored in the database is several orders of magnitude lower than the information stored in the full text index. With an enhanced EAM data model meta data can be stored both in the Database and in the FT, and with a somewhat specialized search logic one can avoid joins between the two systems. The benefit is to exploit the strength of both engines (SQL & FT) but avoid their weaknesses.

Horizontal scalability at the storage systems level one can use P2P file sharing or a cluster file system technology.

3.2.5 Scale-out of the FT Index

Lucene already has some build in support for parallel processing thus scale-up is already supported. Scale-out can be added by splitting the single physical index into multiple segments, which can be worked on in parallel. In this way it is possible to perform several ingests tasks independently and in parallel. Synchronization is only necessary when the new segment are integrated into the global index. The separation into multiple segments opens many possibilities for the maintenance of the FT index. By using the key space partitioning approach, random distribution is achieved when adding new documents to an arbitrary index segment. In this case, to find a document, all index segments have to be searched. The aim of an explicit partitioning scheme is to be able to exclude index segments for the processing of a query if they can't produce any result reducing the overall system load.

3.3 Enhancing EAM Systems for SOA

3.3.1 Electronic Archive Management Infrastructure

IT infrastructures are becoming ever more complex and thus are getting also more difficult to administer. A goal is to provide to the administrators more **automatism** in order to make the installation,configuration and maintenance effort of the individual components easier and less time consuming. Or even better, one would want to build EAM services in a way that it can itself auto deploy and auto configure as required. This approach could be used for single production systems up to whole autonomous infrastructures.

Within this context we talk about an „Efficient Computing Architecture". In such an architecture all layers: Application, Infrastructure-Software, Virtualization and Hardware would be monitored and metered such to ensure that contracted service level agreements (SLA) and

policies are consistently fulfilled. With these mechanism in place an efficient provisioning of the respective components can be secured and delivered. If for example, a CPU is detected to be running for some time at higher regimes, lets say above 80% then the SLA policy can be evaluated and if necessary a new Core, CPU or a complete new node could be provisioned. How such an „efficient computing architecture“ in the concrete case of an EAM system might look like is shown in the following figure.

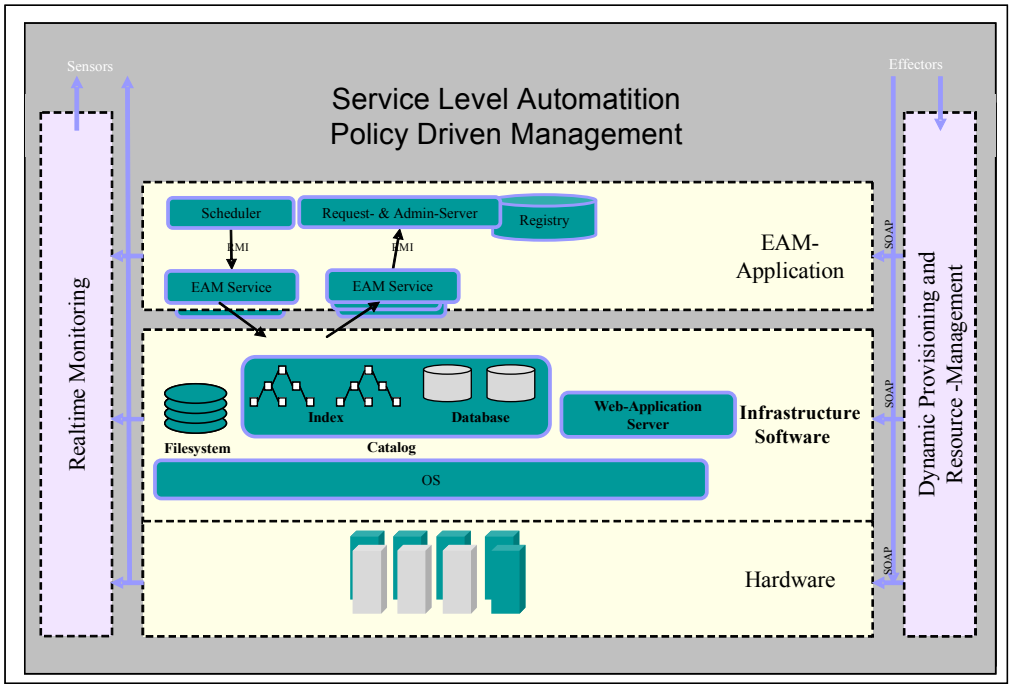

Figure 3: Design approach for an „efficient Computing Architecture“ for EAM

In the upper layer we have the document index, archive and management software. The middle layer constitutes the Infrastructure software layer including the OS with the software components necessary to run the Applications: Databases, Index and Web-Application Server. The filesystem is part of this layer and is considered a component on its own, and should be managed in an independent way. The virtualization layer can handled within the infrastructure software layer itself. At the bottom lower layer we have the hardware components like the Servers, network and storage systems. For the implementation of such an approach it is necessary to transform all hardware and software components into manageable resources. The granularity with which these components are created is easy to determine, as every software product becomes naturally a manageable resource given that they must be treated as black boxes. The same is true for all the hardware components like: Core, CPU, and Node. At the EAM Application level cutting components with the right boundaries is much more difficult and we will explain why in the next chapter 3.3.2.

In order for a component to become a managed resource there is the need to introduce a service interface wrapper for each of it. At runtime, the lifecycle of the individual service instance will be driven via this interface. How and if the SOA concept does represent a meaningful concept in this context will be explained in Section 3.3.2 What is important to understand it that these prerequisites are the necessary condition for an EAM system to be able to orchestrate at runtime resources in an autonomous fashion. By gathering in real-time information about each component and its state, resources can be efficiently used. And this is true for both hardware as well as software resources. The theme we are addressing with this topic is what we call the: „dynamic provisioned Workload Management" infrastructure. An explanation of the underlaying concepts will be given next.

3.3.2 Effectively using SOA in ECM Systems

The Service Oriented Architecture (SOA) concept is both widely accepted and also very controversially discussed. The adoption of SOA together with the abusive use of the Web-Service technology using XML as the transport can lead quickly to a performance killer. On the other side there is the advantage of true platform independence and great flexibility for applications to deal with it.

For EAM systems, Web-Services consumed via SOAP can be used without a big performance penalty at least for services that deal with the management aspect of the system. By doing so one does exploit the advantage of platform independence and ease of use, when dealing with the integration of new components with the infrastructure. The use of SOA technology within the core EAM application on the other side is only partially advisable given the amount and type of data that has to be moved around. Here the performance and scalability characteristics would have to be balanced with respect to the cost of a greater specialization when exploiting platform and environmental specific features.

3.4 CMaaS - Content Management as a Service

The new hype for on-line infrastructures is to support the *Software as a Service* (SaaS) business model. Similarly, we postulate that Content Management as Service - CMaaS is an attractive way to go. CMaaS is not meant to be a service offered in the traditional sense of SaaS but rather an infrastructure service offered to providers more like a service that facilitates the hosting other EAM services. In this sense a provider becomes a tenant and a tenant becomes a provider. The next chapter will explain how this higher level of SaaS abstraction will gain momentum with Web 2.0 becoming the market place and on-line communities becoming the consumers.

Now in analogy to SaaS, the contract between a CMaaS client and provider will be based on quality of service and service level agreements (SLA). In the specific case of the electronic archiving for compliance, the client rents repository space, adds specific services as needed and pays for its true utilization. For example: long term archiving of x Tera Byte of data, retention management, full text indexing, e-discovery and disposition.

As we described before the infrastructure would be custom tailored to fit the exact need not more not less. During runtime the system is then monitored and resource consumption is

metered. In case where SLA would be violated then autonomous measures are triggered to re-configure the production system such to comply with the specific SLA. Resource and audit reports are produced to constantly protocol the chain of events and generate related statistical data. The advantage here is that clients pay only for resources they actually consume.

Compared with the significant up-front expenditures traditionally required to obtain software licenses and purchase and maintain hardware, either in-house or hosted, this approach frees clients from many of the complexities of capacity planning. It also transforms large capital expenditures into much smaller operating costs, and eliminates the need to over-buy "safety net" capacity to handle periodic traffic spikes. Thus this is an implementation of a *„pay per use model“*.

One other advantage of this approach is that EAM services become affordable for smaller firms to buy. Traditionally smaller companies in the SMB market are not able to pay for the administrative cost of managing large ECM production systems with the needs for two or more administrators. The proof that this concept already works and that it gaining more and more traction comes from initiative like the Amazon Infrastructure Services [31], or Yahoo! Search BOSS™: BOSS (Build your Own Search Service) [33].

4 ECM Market Trends and Outlook

One of the most intriguing observations on the Web is the way how new communities almost spontaneously and in an ad hoc fashion become alive and active. The key catalyst influencing this trend is a common interest that expressed through collaboration. The trend manifests itself be means on how, common shared artifacts (some sort of 'content and new data types') are shared via some sort of on-line repository. Examples are the known on-line market places to exchange, share and trade. Almost everything can be obtained, be it real or virtual goods or anything else of interest to anyone. The most popular traded 'object' though is information like: programs, applications, pictures, films, more exotic artifacts like second life characters (avatars) and related accessories will follow.

In this context it is not a surprise that the idea to form a new community often comes as a natural consequence of the launch of new products created specifically tailored to satisfy the needs of a specific community or a more broader audience. Thus a positive loop is generated that drives the growth of the ecosystem.

One very successful case is the one were the iconic companies Apple and Nike teamed up to create a new product the "iPod-compatible Nike footwear" ([8.])their first jointly produced product: the Nike+iPod Sport Kit[5.]. In this case a community of runners share their running experience, statistical data, GPS data, music and everything else as needed via a hosted on-line community place. Another good example is the relationship of the on-line communities formed around the game consoles and the game console manufactures. Here whole new Ecosystems have formed with respective market places with intense trading activities. The Sony on-line store [42.]as an example is as an early adopter of this model. Others are following as the communities are growing in number and size and so becoming so a more attractive market for new products.

Another more indicative example (as of July 2008) in this area is the emergence of a new product created from bonding an interactive game and a music label. The new **"product"** created is the pairing of the game 'Guitar Hero'[19.], and songs from music label Activision[2.]. Here we can observe the birth of an on-line product from different owners or providers motivated by a community of music aficionados facilitated by a group of providers which are investing a substantial amount of money, in this case Vivendi and Activision. The Reuters [35.]article reads: *„The complex deal will give Vivendi a 52 percent stake in a new industry giant called Activision Blizzard with annual revenue of $3.8 billion, rivaling that of Electronic Arts Inc … the world's biggest independent game publisher. The Commission said for "all categories of game software, the combined firm would continue … Activision is riding high on the success of games such as "Guitar Hero", "Call of Duty" and "Tony Hawk" but has lacked an offering in the on-line role-playing area, dominated by "World of Warcraft" from Vivendi's Blizzard Entertainment."*

Here a clear trend can be seen, and we believe this trend will be accelerated by content management services and tools that make hosting and creating an on-line presence easy and affordable to everyone. The motto is "every one can become a provider or a tenant or both, everywhere any time".

A trend setter with this respect is Amazon with its new portfolio of on-line software and infrastructure service offerings like: on-line stores, on-line repositories, or on-line 'data centers' which can be designed and deployed on the fly, rented at an affordable price for as long as the new service is offered or survives. Using the new Amazon on-line service offering that promise is that no upfront investment in IT infrastructure and basic software stack is necessary. Within hours a new 'Elastic Cloud EC2'[4.] can be generated ready for development, selling or trading. We are talking about the Amazon aStore: The on-line Virtual Store initiative:From there one can read …"What is aStore? aStore is a new Associates product that gives you the power to create a professional on-line store, in minutes and without the need for programming skills, that can be embedded within or linked to from your website."

Amazon Web Services:[4.] based on the on-line Virtual Storage and IT Center using the Amazon Elastic Cloud EC2[4.] and Amazon Simple Queue Service SQS [4.] Service. Here we read: … "*Now, Amazon is making an even greater stretch -- selling storage, computing power and other behind-the-scenes data center services. The venture, which Amazon expects will grow into a significant business segment, could help keep the company strong if retailers get hit by an economic downturn. More broadly,* Amazon Web Services,*[4.] as the business is called, could improve chances for a new generation of Web startups by slashing how much they spend up front on costly infrastructure."*

With all these said one obvious conclusion is that significant business opportunities come from leveraging and re-purposing enterprise class data, next.

4.1 CMaaS - Content Management as a Service more than a thought

Future ECM service providers will demand for open on-line platforms to avoid their data and services being locked in. It is because of this that standards for data, industry data/process models, and programming technologies on the Web are emerging to improve openness and collaboration in tandem with security and privacy.

We do see that Enterprises have a vital interest that interactive, on-line community platforms will become an alternative choice for developers to 'compose & orchestrate' new content management solutions. As a starting point a few selected areas will be used to experiment first but as the technology matures the whole enterprise will be opened up to a world wide audience. In this context new repository technologies inside and outside enterprises are emerging to support Web based community platforms in the areas of:

- Communities of interest and collaborative intelligence
- Enterprise 2.0 [3.]- the open and collaborative Enterprise
- Massive Multi-tenancy

From a infrastructure stand point the biggest challenge is with the managing aspect of massive multi-tenancy as a consequence of the collaborative and distributed hosted environment. In order for the Content Management infrastructure to be able of consistently support an ease of administration and maintenance current ECM technology must be adjusted and new key components developed. We anticipate that, in this highly open and ever changing on-line market place the area of special interest is with respect to: security, data privacy, retention management, and the mandatory audit capabilities. It is within this context that Content Management needs to mature and become a ubiquitous on-demand *„pay per use"* service.

4.2 Communities of Interest and Collaborative Intelligence

What are the emerging technologies for future CMaaS Systems. This is best answered by anticipating the usage scenarios of future on-line communities. Therefore we ask: What do communities do, once formed? The answer is they:

- Share common interests and needs
- Share information and content
- Increase information sharing to known and unanticipated users
- Identify data/service assets (i.e. files, databases, & information services)
- Make data/service visible, accessible, understandable (tagged and discoverable)
- Define shared vocabularies & taxonomies
- Register semantic & structural meta data
- And much more …

And where does this lead us towards to? The answer might be the open Enterprise – Enterprise 2.0?

4.3 Enterprise 2.0 - The Open Enterprise

The foundation of this new business initiative is the open exchange of services and data. The Enterprise adoption of CMaaS platforms starts from non-mission critical areas, often initiated outside the data center, and will lead into heterogeneous environment composed of legacy, on-premise and off-premise active content solutions, dictated by the need for the services to be global and omnipresent.

Service oriented computing allows the integration of heterogeneous legacy applications be it on-premise or off-premise but running on on-line content management platforms. It is this aspect that will ensure that the future content management services platform will benefit from communities by means through the community ecosystems.

The most recent example, the Amazon eCommerce Web platform shows what "eCommerce for everyone" [4.]means. Here huge amount of data about users and merchandise is collected, analyzed, stored and managed. This can clearly be seen as an act of building communities around content and data. Typically these are:

- Community of stores: contribute data to Amazon, leverage data from Amazon to build their stores
- Community of developers: leverage the data and web services from Amazon to build new services

In a way this can be seen a 'positive loop' that drives continuous hyper growth through the simple fact of the existence of the community itself. In the same spirit we see here an opportunity for Enterprises to leverage and re-purpose their high value data in a controlled way. The pattern to follow is the use of the CMaaS framework as the enabling on-line platform into new business opportunities.

In finance, for example banks could facilitate risk-data aggregation, anonymization, and predictive analysis with community participation and in doing so they would facilitate the creation of new opportunities to banks which are member of a community of banks. In analogy, many more types of enterprise data can be leveraged via CMaaS on-line platforms to drive new business and growth.

4.3.1 New Business Models are evolving

New business implies also new business models or enhancements of the old one. The question is what and how is traded on-line, or gets acquired and paid. Yesterday the software license model was based on concurrent users, today software licenses are based on CPU consumption. Tomorrow, with the emerging of Software as a Service - SaaS paradigm, service quality is based on service level agreements – consequently the SLA might be the metric of choice with content being one billing parameter in the list.

An interesting example of this species is Yahoo! Search BOSS™ [47.].See also Yahoo BOSS – The Next Step in our Open Search Ecosystem [47.].BOSS (Build your Own Search Service) is Yahoo!'s open search web services platform. Aiming at fostering innovation in the search industry. With BOSS, developers, start-ups, and large Internet companies can build and launch web-scale search products that utilize the entire Yahoo! Search index as the hosting platform. With BOSS enterprises have access to Yahoo!'s powerful infrastructure and investments in crawling and indexing, ranking and relevancy algorithms. Enterprises can combine their unique assets and ideas with the Yahoo search technology assets. Thus this combination makes Yahoo BOSS a platform for the next generation of search innovation, serving hundreds of millions of users across the Web that are passionate about building open platforms.

The typical SLA for this kind of service among others are: Queries Per Day , usage charge for blending of proprietary and Yahoo! Search Content , monetization and White-Label.

4.3.2 Monetary Models i.e. Revenue Sharing

A monetization[45.] platform is the enabling framework that facilitates on-line partners to jointly participate in the economics of jointly developed and shared products. The on-line platform must implement a simple integrated payment model that regulates certain implementation and exclusivity requirements from a monetary stand point. The overall business motto of the Enterprise 2.0 can therefore be synthesized into:

- Open Applications via the Web
- Open Service
- Open Data
- Open Collaboration

i.e. a real Information on Demand – IOD initiative.

4.4 Massive Multitenancy

In the above scenario the key entity around which everything revolves is the community member. That is the individual, member of one or multiple communities that keeps the community alive and in motion. Every single entity can be seen as a tenant that has rented its on-line presence from CMaaS platform providers. As previously discussed in the Yahoo BOSS example before. These tenants vests different roles: the consumer, the provider the contributor becoming a tenant is a matter of a subscription to a community via an on-line platform.

It is because of this circumstance that new, more complex requirements for on-line Content Management arise. Given the size of the on-line population we have to deal with managing the complexity of massive multi-tenancy. By that we mean that there is a need for a CMaaS architecture that enables adequate isolation for sharing resource among tenants. The goal is a cost-effective, optimal scalability for large number of tenants by exploiting the synergy of the shared data center infrastructure. What makes massive multi-tenancy extremely difficult is the trade off between security, privacy, cost, and individual customization requirements. Because of the importance of these concept, let us define the taxonomy with which we will use to define and classify multitenancy (MT) models.

4.4.1 Definition of Multitenancy [33]

For the sake of a better understanding we want to differentiate between the dynamic and the static aspect of multitenancy. This leads to, two types of MT tiers that must be considered:

1. Execution Tier (ET) [Fehler: Referenz nicht gefunden]
2. Data Tier (DT). [Fehler: Referenz nicht gefunden]

Within the first tier we want to further classify the multitenancy into 4 different runtime isolation levels E-I to E-IV for each tenant. Respectively been the: 1) Application level, 2) Middle-ware level, 3) OS container level and 4) VM level of isolation. In the first case, the isolation of tenants among each other, is achieved through application logic. The runtime environment is completely shared. In the second case, isolation is at application level and we have one application per tenant.

The third case describes an environment where isolation is achieved at OS container level and finally the forth case is physically sharing the hardware but logically providing completely independent and isolated runtime production systems based on Virtual Machines or Logical Partitions. The latter exposes the best isolation layer but also the highest configuration complexity.

Multitenancy: Execution Tier	
E-I: Application level MT	E-II: Middle-ware level MT
E-III: OS Containers level MT	E-IV: VM level MT

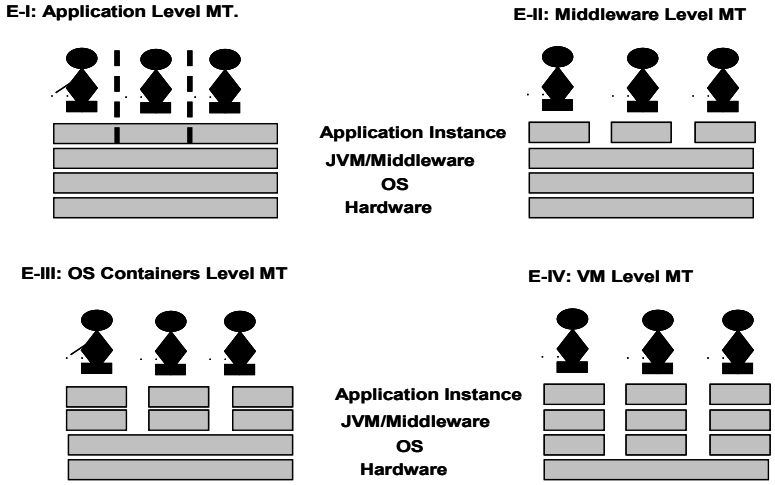

Figure 4: Multi-tenancy levels at execution time

Within the second, the data tier, we can further classify the multi-tenancy into 4 types: D-I to D-IV.

Multitenancy: Data Tier	
D-I: Shared tables, hidden Tenant ID.	D-II: Shared DB, separate tables.
D-III: Separate Databases shared DB Server.	D-IV: Separate Database Servers

In the first case D-I, all tenants are managed by means of one single shared data model that has been built in the distinction between the individual tenants. Segregation of data is achieved by hidden tenant Ids and businesses logic implemented in the application layer. In the second case D-II, the database model is more elaborated by the introduction of a set of separate tables related to each individual tenant.
Case 3 D-III,depicts the case were every tenant has its own schema but shares the Database server. And finally in case 4 D-IV, here one Database server is assigned to each tenant. D-IV is the best form of data segregation, but has also the highest administrative overhead.

What was said so far is summarized in the next figure below.

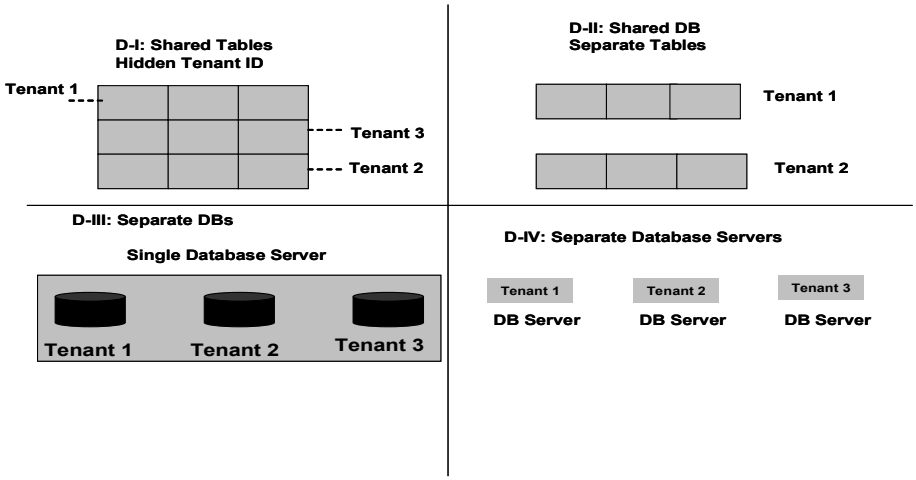

Figure 5: Multitenancy levels at the data tier level.

Thus we get a 2-dimensional plane that allows us to catalog content management systems based on their characteristics and according to the two most important variables to consider:

a) Degree of isolation
b) Complexity of configuration.

As one can see the latter translates directly into the aspect of administrative burden that an on-line platform poses and the capabilities that it must expose in the respective context.

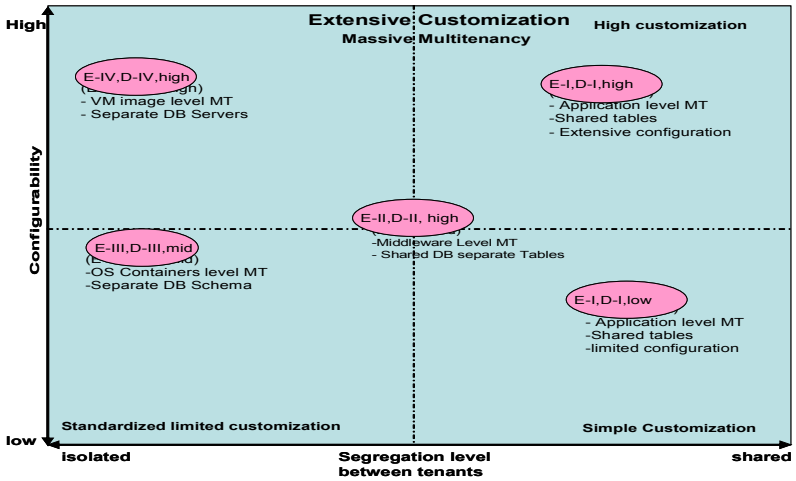

Figure 6: Managing massive multitenancy in the MT plane

This information is visualized in the diagram above.
The four quadrant above, lists the two key characteristics for managing massive multi-tenancy, which are the segregation level between tenants and the increase in complexity of configurability by number of tenants. For example, CMaaS systems in the lower left quadrant would provide high isolated runtime systems with some degree of configuration effort. Where as CMaaS systems in the in the right lower quadrant provide both less runtime isolation and simple to handle configuration. Moving up and right runtime increases the levels of sharing and more configurability at the expense of more administrative effort.

4.5 New CMaaS Reference Model

With these trends in mind we anticipate that new technologies are required to support a value/ risk optimization framework for opening enterprise data to communities of interest and collaborative intelligence. The aim will be to allow for in vivo software engineering, and massive multitenancy.

We envision an evolution of the CMaaS platform landscape for ECM Solutions with off-premise shared component, but with the ability to accommodate massive multi-tenancy that relying on the efficiency of Mega Data Centers. In doing so, a positive loop is generated that drives the growth of the ecosystem in which data is openly shared through the platform form community contributions that are the motor of the positive loop.

At this point and in this context we ask our self, what is Content Management as a Service all about? The answer could be: *"CMaaS is an open, highly extensible on-line platforms that is capable to enable continuous enrichment of innovative content and data services such to allow the acquisition of informational insights from user and client communities"*. If embraced Enterprises will be able to leverage their data to establish their on-line presence and in the same time profiting from the drive of business growth based on new business models. Thus we can summarize that the key system characteristics for a future CMaaS platform are:

1. Developing and deploying CMaaS on-line platforms must be easy. From concept to deployment it should take only hours.
2. It must facilitate the development of ECM solutions by means of composing from basic components, orchestrated to function together by a dynamic provisioning manager assisted at runtime by a dynamic workload manager.
3. CMaaS on-line platforms must enable accessibility and extensibility at the interface level, but exhibit lock-in of function and data.
4. It must adopt a service orientation paradigm that allows the integration of heterogeneous legacy, on-premise and off-premise ECM solutions. Assuring content services and data exchange is open and not locked-in.

With these characteristics in mind we can try to define a CMaaS Reference Model.

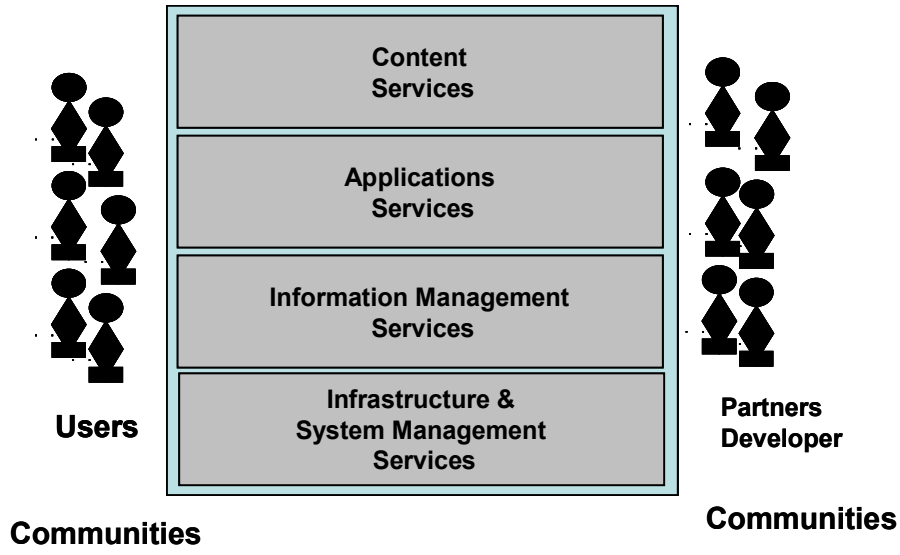

Figure 7: Suggested enhanced CMaaS Reference Model

The goal of: continuous enrichment of innovative content and data services that facilitates informational insights from user and developer communities.
Enterprises can so leverage their data to establish on-line platforms and drive new business growth by enabling accessibility and extensibility at the interface level. With this we mean:

- Accessibility: How easy data is accessed through user interface/batch interface/API
- Extensibility: How easy it is configured, customized and extended.
- Affordability: How expensive an individually customized CMaaS infrastructure is.

Thus we might conclude that given the distributed nature of the on-line infrastructure, scalability will inherently be based on a scale-out model not on scale-up for electronic archive and ultimately also content management systems. The new on-line ECM solutions will focus on core functionality of managing data/content throughout its lifecycle and facilitating collaboration among communities and their members by providing at the right time the right data, vital to their core business. From this we can dare the prognosis that future CMaaS system must have the following capabilities and characteristics.

First there is the need of an Enterprise Data Architecture that feeds and fosters the on-line ecosystem. By this we mean, a well suited software services framework for opening the enterprise data to trusted communities with the benefit of optimizing value and minimizing risk.
Next, at the functional level future ECM systems must expose a collaborative intelligence that enables exponential growth of applications developed by the community on the platform.

Other people by using these applications will gain insights about how to improve, reducing time to value. What is needed is an Hybrid Integration Model, required for integration of on-premise content services applications. The whole must by complemented by a sound authentication and entitlement model in order for each system to be identified, allowing the auditing of transactions. Most importantly it must also provide an isolation model in which each domain is protected against attacks from other domains but that allows mediated communications for the exchange of information between members of domains in a transparent and seamless fashion. Figure 6 below shows an attempt to visualize a highly distributed application landscape, were different tenants consume their acquired services provided transparently by different providers in the Web 2.0 Cloud. For example, the idea is that Amazon would provide via the *Amazon Simple Storage Service* the, a) EC2 repository to store and archive electronic documents, 2) Yahoo delivers the search service. Where documents would flow from the Amazon EC2 repository to the Yahoo search service (BOSS) by means of a, 3) Amazon *Simple Queue Service* (SQS).

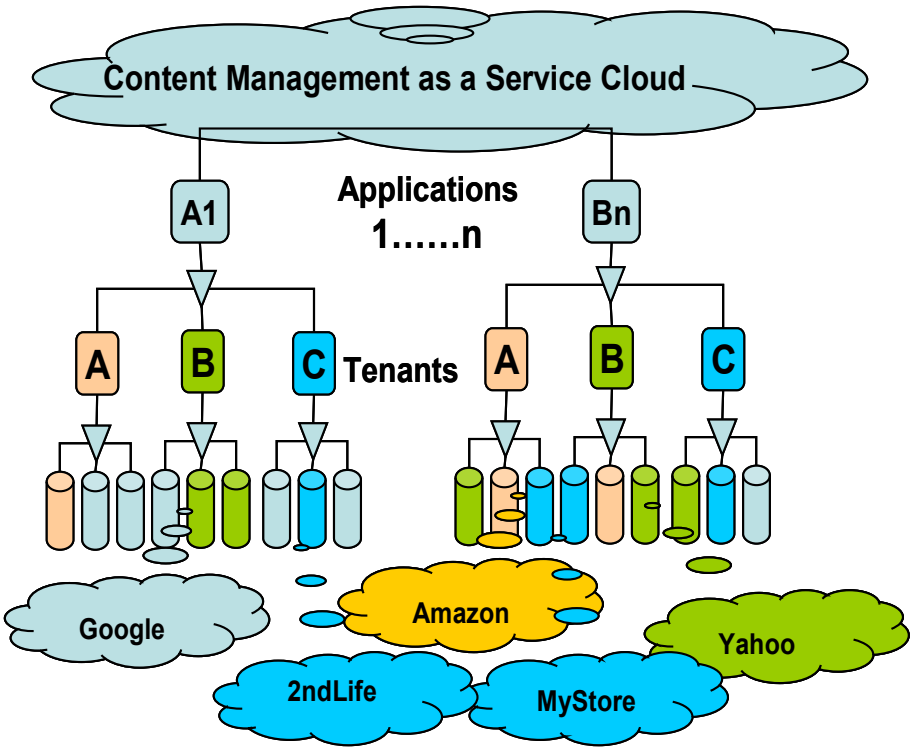

Figure 8: Massive Multi-Tenancy in the CMaaS Cloud

Somewhat orthogonal to the above mentioned system capabilities, from an administrative stand-point the CMaaS infrastructure would have to provide the ability of defining a common quality of services (QoS) and service level agreements (SLA) governance across platforms. Complemented by QoS and SLA management standards regarding integrity, compliance, and risk. The inherent distributed nature of service consumption will also push the need for service problem determination automation in the cloud with innovative way of problem resolution and dispute management. Both syntactic and semantic interoperability at process, data, UI and service composition level will require new types of synchronization at federation level. Services will be Web delivered or as an alternative put into an appliance. Companies will have to provide organization and maintenance constructs for content in order to help discover and identify high value enterprise data assets. And last but not least tools are needed to evaluate, analyze business processes & IT impacts.

5 Conclusion and Outlook

In our paper we introduced and discussed the design of current and future electronic archive management systems – EAM. The new on-line EAM solutions will focus on core functionality of managing data/content throughout its lifecycle and facilitating collaboration among communities and their members by providing at the right time the right data, vital to their core business. Given the distributed nature of the on-line infrastructure, configurability, flexibility and scalability will be the main characteristics. We believe that scale will inherently be based on a scale-out model not on scale-up. Especially when dealing with high-end, high scalable electronic document archiving and discovery services. We think new design approaches will be built upon an innovative mix of open source and legacy technology. Current trends suggests that scale-out is achieved by means of abstraction and virtualization, allowing an indirection level to be introduced for creating a single logical archive entity to be distributed over a cluster of physical entities. The key technology used might be a key space partitioning approach based on DHT [31.] algorithms where content is addressed via a simple URI. Future EAM system will require an Enterprise Data Architecture that feeds and fosters the on-line ecosystem. This architecture is based on a well suited software services framework that opens enterprise data collections to trusted communities with the benefit of optimizing value and minimizing risk. At the functional level future ECM systems must expose a collaborative intelligence that enables exponential growth of applications due to the collaborative effort of the communities.

6 References

1. Apache, Muse
2. Activision: Activision
3. AIIM 2008 Enterprise 2.0
4. Amazon eCommerce Web platform: eCommerce for everyone" *Amazon Elastic Compute Cloud Amazon SimpleDB Amazon Simple Storage Service Amazon Simple Queue Service*
5. Apple: Nike+iPod Sport Kit
6. John Bace & Debra Logan, "The Costs and Risks of E-discovery in Litigation", Gartner, December 1, 2005
7. Base One. "Database Scalability - Dispelling myths about the limits of database-centric architecture", 2007.
8. Businesses Week: Business week
9. *Dearing, G.* (2008): Top 5 Content Management Trends for 2008, on-line: http://www.infor mationweek.com/blog/main/archives/2008/01/top_5_content_m.html, Stand: 16.01.2008, Abruf: 02.07.2008.
10. Demir Barlas & Tamina Vahidy. "The Email Glut", Line56, January 24, 2006
11. E. Brewer. "Combining systems and databases: A search engine retrospective". In M. Stonebraker and J. Hellerstein, editors, Readings in Database Systems. MIT Press, 4th edition, 2004.
12. S. Chaudhuri, U. Dayal, and T.W. Yan. "Join queries with external text sources: Execution and optimization techniques", Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, pages 410–422, 1995.
13. K. Chen. "IBM DB2 content manager v8 implementation on DB2 universal database: A primer". Technical report, IBM, 2003.
14. *Barbara Churchill, Linda Clark, Jonathan Rosenoer and Fritz von Bulow,* The **impact of electronically stored**. **information on corporate legal** and. **compliance management**:. An **IBM point of view**. October 2006. By *IBM Corporation*
15. J. Dean, S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In OSDI'04: 6th Symposium on Operating Systems Design and Implementation, 2004.
16. C. DiCenzo, K. Chin. "Magic Quadrant for E-Mail Active Archiving". Gartner, 2007.

17. Ghodsi, Ali (2006) *Distributed k-ary System: Algorithms for Distributed Hash Tables*. Doctoral thesis, KTH - Royal Institute of Technology.
18. J. Gray, A. Reuter. „Transaction Processing: Concepts and Techniques", Morgan Kaufmann Publishers, 1993.
19. Guitar Hero: Guitar Hero'
20. *Gentzsch, W. et al.* (2005): Self-Adaptable Autonomics Computing Systems: An Industry View, in: Database and Expert Systems Applications, 2005. Proceedings of the 16th Interational Workshop on Database and Expert Systems Applications (DEXA'05), 2005, S. 201-205.
21. *Heueis, R.* (2008): Digitale Kommunikation: Der Einfluss intramedialer Kontextinformationen auf den Aufwand im E-Mail-Management, 2008.
22. Hausheer, D. Stiller, B.Comput. Eng. & Networks Lab., Eidgenossische Tech. Hochschule, Zurich, Switzerland; Design of a distributed P2P-based content management middleware
23. IBM Corporation, IBM Content Manager
24. IBM Corporation,, IBM Dynamic Infrastructure for mySAP Business Suite
25. C. Mega, F. Wagner, B. Mitschang. „From Content Management to Enterprise Content Management", In Datenbanksysteme in Business, Technologie und Web (BTW), 2005.
26. B. McLean, P. Elkind. "The smartest guys in the room – The amazing rise and scandalous fall of Enron", Portfolio, 2004.
27. Peer-to-Patent: Community Patent Review
28. Petar Maymounkov and David Mazieres. Kademlia: A Peer-to-peer Information System Based on the XOR Metric
{petar,dm}@cs.nyu.edu
29. Michael, M.; J.E. Moreira, D. Shiloach, R.W. Wisniewski (March 26, 2007). Scale-up x Scale-out: A Case Study using Nutch/Lucene. *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007.*. IEEE International. Retrieved on 2008-01-10.
30. .José E. Moreira, Maged M. Michael, Dilma Da Silva, Doron Shiloach, Parijat Dube, Li Zhang, Scalability of the Nutch search engine June 2007 **ICS '07:** Proceedings of the 21st annual international conference on Supercomputing **Publisher:**ACM
31. OpenDHT, publicly accessible distributed hash table (DHT) service
32. J. Plotkin, "E-mail discovery in civil litigation: Worst case scenario vs. best practices", white paper, kvsinc.com, April 2004.
33. The Radicati Group, Inc., Taming the Growth of Email – An ROI Analysis, White Paper by The Radicati Group, Inc.

34. The Radicati Group, Inc., An Overview of the Archiving Market and Jatheon Technologies, by The Radicati Group, Inc. September 2006
35. Reuters: The Reuters
36. Symantec Corporation, "Symantec Enterprise Vault 7.0 Introduction and Planning", 2006.
37. 26.G. Thickins, "Compliance: Do no evil – critical implications and opportunities for storage", Byte and Switch Insider, 2(5), 2004.
38. U.S. Department of the Interior, "It's in the mail: Common questions about electronic mail and official records", 2006.
39. Troy Werelius. "Trends in Email Archiving", Computer World Storage Networking World Online, August 21, 2006
40. Hao Yu; Moreira, J.E.; Dube, P.; I-hsin Chung; Li Zhang , Performance Studies of a WebSphere Application, Trade, in Scale-out and Scale-up Environments Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International Volume ,
41. *Scott, J./Globe, A./Schiffner, K.* (2004): K. Jungels and Gardens: The Evolution of Knowledge Management at J.D. Edwards, in: MIS Quarterly Executive, 2004, Nr. 3/1 vom 03.2004, S. 37-52.
42. Sony: The Sony on-line store
43. *Wagner, F. et al.* (2008a): ADBIS-Paper, Finnland 2008.
44. *Wagner, F. et al.* (2008b): ICD-Paper, Italien 2008.
45. Wikipedia, the free encyclopedia – Scalability see also Amdahl's Low, DHT Monetization Scalability Multitenancy
46. W. W. Yung, "Explore the IBM mail management and compliance solution". developerWorks, 2005.
47. Yahoo! Search BOSS™: BOSS (Build your Own Search Service) - Yahoo BOSS – The Next Step in our Open Search Ecosystem