A Semantic Approach to Information Governance

Architecture Design And Blueprint Of Modern IG Systems

vorgelegt von
Dipl.-Phys.
Cataldo Mega
geb. in San Giogio Jonico, Taranto - Italy

von der Fakultät V - IPVS-AS
Universität Stuttgart
zur Erlangung des akademischen Grades

Doktor der Informatik
-Dr. rer. nat

genehmigte Dissertation

Promotionsausschuss:
Vorsitzender: Prof. Dr.-Ing. habil. Bernhard Mitschang, Head of Institute, Applications of Parallel and Distributed Systems, IPVS, Universität Stuttgart
Gutachter: Prof. Dr.-Ing. - Norbert Ritter, Universität Hamburg
Gutachterin: Prof. Carla Di Maggio (virtual assistant)
Gutachter: Prof. D
Tag der wissenschaftlichen Aussprache: XX. xxxx 2022

Stuttgart 2023

Zusammenfassung

In our paper we showed a possible evolutionary path of a n ECM systems, outlined their core functionality and the services they provide to businesses. We discussed a set of representative use case scenarios and defined the workload model from where we derived the blueprint of ECM systems their underlying architecture design. From our analysis we found that the repository framework consist of a globally distributed, heterogeneous infrastructure build from commodity hardware components deployed on- or off-premise or a combination of both. We also learned that the key non-functional characteristics of modern ECM systems are: flexibility, configurability and scale at an affordable cost. Current trends show these goals can be achieved by means of abstraction, virtualization and containerization thus allowing an indirection level to be introduced for creating a single logical ECM entity that is geographically distributed in nature. Many decision makers in larger Enterprises have mandated the restructure of their IT-Environment such to cut cost and increase productivity. We have identified the the key areas of outdated, inflexible and slow IT-Processes. Over the years these homegrown processes have become a major show stopper when it comes to provided more dynamic, flexible and fast resource allocation services. The established application deployment processes no longer satisfy demand in an 'Everything as a Service' (EaaS) world. The way out of this dilemma is the adoption of 'Cloud'-technology. The latter has industrialized the delivery of managed services by employing a new consumption model and thus showing a clear way out into a more efficient IT-Infrastructure.

Abstract

The 'Cloud' or to be more precise cloud computing has become a synonym for "faster, flexible, scalable and last but not least cost efficient". Many decision makers in mayor Enterprises are mandating the restructuring of their IT-Environment such to cut cost and increase productivity. For the latter, cloud implies the industrialization of delivery of content services which means "employing a new consumption model inspired by the consumer internet services. In this market place one business is offering a managed service to the other business, thus building a food chain for the B-B eco system. Cloud aims at: a) exploiting the economies-of-scale derived from an inherent massive multi-tenant environment, b) the ubiquity of the service and c) an increasing secure and trusted trading environment. In B-B the expectation is of an enterprise ready, replicable standard SaaS offering where costs are reduced to the actual consumption i.e. pay per use business model and security and privacy are guaranteed. Resource topology optimization is a mandatory capability for cloud systems.

Dedicated to ... Cataldo Mega

Acknowledgements

I would like to acknowledge the thousands of individuals who have coded for open-source projects for free. It is due to their efforts that scientific work with powerful tools is possible.

Table of Contents

List of Figures

List of Tables

Abbreviations

This document is incomplete. The external file associated with the glossary 'abbreviations' (which should be called `mythesis.gls-abr`) hasn't been created.

Check the contents of the file `mythesis.glo-abr`. If it's empty, that means you haven't indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can't be generated. If the file isn't empty, the document build process hasn't been completed.

Try one of the following:

- Add `automake` to your package option list when you load `glossaries-extra.sty`. For example:

  `\usepackage[automake]{glossaries-extra}`

- Run the external (Lua) application:

  `makeglossaries-lite.lua "mythesis"`

- Run the external (Perl) application:

  `makeglossaries "mythesis"`

Then rerun LaTeX on this document.

This message will be removed once the problem has been fixed.

Symbols

This document is incomplete. The external file associated with the glossary 'symbolslist' (which should be called `mythesis.syi`) hasn't been created.

Check the contents of the file `mythesis.syg`. If it's empty, that means you haven't indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can't be generated. If the file isn't empty, the document build process hasn't been completed.

Try one of the following:

- Add `automake` to your package option list when you load `glossaries-extra.sty`. For example:

  `\usepackage[automake]{glossaries-extra}`

- Run the external (Lua) application:

  `makeglossaries-lite.lua "mythesis"`

- Run the external (Perl) application:

  `makeglossaries "mythesis"`

Then rerun LaTeX on this document.

This message will be removed once the problem has been fixed.

1

Introduction

1.1 Background:

The cloud hype has reached the Enterprise. It is expected that the Web 3.0 the Internet of things and data will revolve around the B-B market needs. For the latter, cloud implies the industrialization of delivery of managed services this means “employing a new consumption model inspired by the consumer internet services (B-C).

Corporations have started to embrace and some have already adopted the ’cloud’ delivery model of managed services. Very much like the SOA approach,

One key area that must be looked at are the outdated, very often inflexible and slow resource allocation processes. Over the years these homegrown IT-Processes have become a major show stopper when it comes to a more dynamic and flexible resource allocation. The established deployment processes no longer satisfy demand in an ‘Everything as a Service’ (EaaS) world. The way out of this dilemma is the adoption of ‘Cloud’-technology. The latter has industrialized the delivery of managed services by employing a new consumption model and thus showing a clear way out into a more efficient IT-Infrastructure.

For example a multi-national bank, with its many different business models and related complex business scenarios that require elaborated workflows. Such a bank has legacy data bases, data warehouses and Enterprise content management systems to support their business units with all the processing of business data created and associated digital information collected. Independent of business model employed an Enterprise is obliged to comply with corporate and legal compliance and therefore is forced to implement a governance model for all information it is responsible for. Enterprises need to know what information they have, who is authorized to access it, how long the information needs to be kept and when it must be disposed.

These four fundamental business characteristics imply that a catalog record must always point to its data counterpart, data can’t be orphaned nor should there be unknown duplicates. Companies already use an infrastructure to support their business processes and also satisfy their business intelligence needs using data bases, data marts, content management systems and a heterogeneous, geographically distributed repository landscape. What has been missing in the past were new tools to better handle the sheer volumes of data and scale in processing power required.

We believe that the Cloud hype has brought fresh wind, new technologies and methods to boost Analytics and Discovery together with the promise to help cope the sheer volumes of information, the scale of processing power required and the benefit of flexible and cost-effective common infrastructure. We would like to show that Enterprise Content Management Systems and the 'Cloud' complement each other well to the benefit of every business that requires both content management and large scale analytics.

1.1.1 Goals

With this master thesis we want to investigate how the architecture design of legacy ECM Systems can be evolved such for being able to exploit todays cloud technologies in a non-disruptive way and thus overcoming current issues. We believe the benefits for ECM from embracing the 'Cloud': the exploitation of the economies-of-scale, an inherent support for multi-tenancy, and the ubiquity of the service offering in an increasingly secure and trusted environment. [1].

1.1.2 Expectations

Focus of this master thesis is to analyze the legacy architecture of ECM Systems, describe the shortcomings of the current design component by component and then develop a strategy to change, enhance and evolve the design in a way that allows the integration with current and future 'cloud technology' in a non-disruptive way.

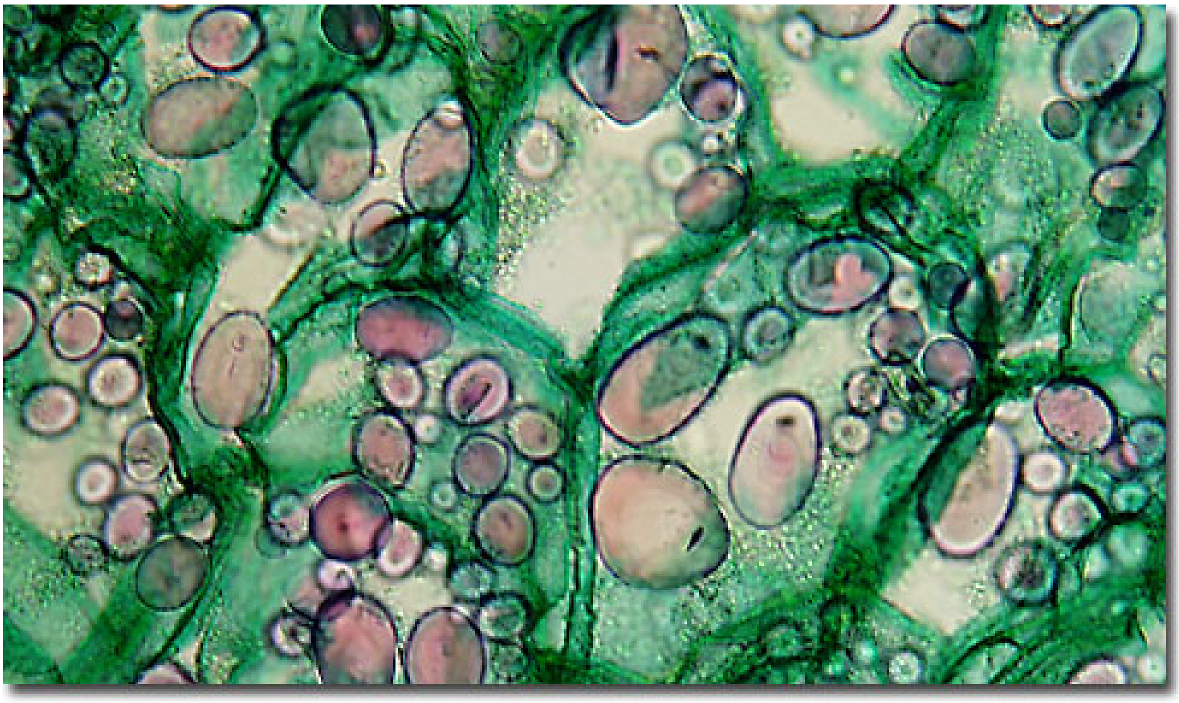

Figure 1.1: A common ECM system architecture design - The figure Molecular Expressions.

[1]Enterprise Content Management - ECM

The work should first analyze the current system architecture then introduce design aspects of dynamic deployment and automated operations and finally demonstrate the feasibility that legacy ECM systems can take advantage of cloud technology like:

- Continuous delivery & Integration
- Virtualization, and Containerization
- Automated Administration & Operations
- Cloud enabled Monitoring & Metering
-

In the second more practical part of this master work, a prototype ECM service should be developed, implemented and deployed into a cloud environment. As result a qualitative and quantitative evidence is to be shown of the optimized efficiency and minimize resource consumption compared to its legacy counterpart.

1.1.3 Approach

The detailed tasks are:

- Part I
  - Familiarize yourself with the current ECM system architecture design and its context
  - Perform a gap analysis relative to the new business requirements
  - Document the design problems derived and think possible solutions.
  - Research and classify current and future cloud technology. This means you should analyze and compare the different cloud technologies and / or products available on the market, company laboratories and research institutions. Here are examples of cloud products /technology: Openstack, IBM Red Hat / Openshift, Podman / Kubernetes, Puppet, AWS CloudFormation, Ansible, Chef, Terraform, Google Cloud Deployment Manager, Microsoft Azure Automation, Foreman, VMware vCenter Configuration Manager (VCM), Cisco Intelligent Automation for Cloud
  - Create a list of all possible candidates document their cloud technology and their key capabilities, detail the pro and cons.
  - Using all the above, sketch out the design changes required to satisfy function and non-functional requirements
  - Create a concept design of the new and revised system architecture, outline the operational extensions required for being able to integrate with the cloud environment(s) chosen
  - Consolidate the documentation so far and present your intermediate results
- Part II
  - Implement the prototype ECM service chosen

  - Verify and validate that deployment topology works as expected
  - Consider the integration points challenges and benefits
  - Monitor consumption and the operations of new IT-Infrastructure. Measure the efficiency and consumption figures and estimate the benefits gained with the new approach.
  - Consolidate the documentation so far and present your intermediate results

- Part III
  - Evaluate the results and consolidate documentation so far
  - Explain the new 'cloud' delivery model', the enhanced service offering and the benefits to the own business units inside the company and customers outside.
  - Final presentation and explanation if the final results

1.2 SI-Units

Please use siunitx-package: $1\,\mathrm{V} = 1\,\Omega\,1\,\mathrm{A}$

2

Definition of Enterprise Content Management (ECM)

AIIM - The Association for Information and Image Management defines Enterprise Content Management as: "Enterprise Content Management is the strategies, methods and tools used to capture, manage, store, preserve, and deliver content and documents related to organizational processes. ECM tools and strategies allow the management of an organization's unstructured information, wherever that information exists." [1] The essence of an ECM System consists therefore of a collection of infrastructure components that fit into a multi-layer model of technologies for efficiently handling the 'Information Lifecycle and Governance (ILG)' of 'mostly unstructured and semi-structured data'. As such, Enterprise Content Management is the necessary fabric of the overarching E-Business application area.

2.1 3.1 What is an ECM system all about?

The constantly progressing penetration of IT in the private and in particular within businesses has given rise to an immense volume of all type of data (rich media data) in which 'unstructured data' represents the predominant portion. The nontrivial technical hurdles to process and manage these volumes of information has forced companies in to an effort to adopt new technologies and methods, borrowed the hype on clouds that allows scale at an affordable cost. This has led the ECM world to move away from static, monolithic architecture designs towards an all virtualized and componentized content repository concept based on 'Software Defined Infrastructure' and 'Content Management as a Services' (CMaaS) models. The aim for ECM solutions architect is to employ a business model that allows to offer basic content services on-demand to ECM customers. Think of a 'Build your own ECM Application' BYOCA-model which in essence represents an individually composed solution for each customer or better each tenant. Thus the ECM systems must cope with the requirement to support massive multi-tenancy and to provide Content Management as a Service – CMaaS. Current trends in IT Hardware technology show that these goals can only be achieved by means of exploiting a scale-out approach using commodity hardware, which is configured in a programmatic fashion and provisioned on-demand. As a consequence the underlying IT-infrastructure must provide a massive scalability at an affordable price. By looking at today key cloud players we realize that both goals scalability and affordable price have become possible thanks to newest advances in dynamic workload management and cloud technology. Our conclusion is that today's design alternatives to the traditional way of purchasing a monolithic shrink wrapped ECM solution is to acquire ECM services based on Service Level Agreements (SLA) 'on-demand' and pay by 'consumption'. Very much like how we consume electricity for our appliances. We looked at typical ECM companies in the finance market and have observed that meanwhile these companies have recognized these trends and are starting to restructure their business processes. While key business areas like legal compliance, electronic discovery, and document retention management remain their production systems must be adjusted to fit in to the new environment and serve a new class of end users requirements.

2.1.1 3.1.1 Key functional requirements:

1. A heterogeneous, geographically distributed repository infrastructure framework based on storage subsystems with elaborated storage management capabilities that supports rich media data and multi-media indexing & search capabilities as well as a federated repository abstraction layer.

2. Repository Abstraction Layer - RAL - an abstraction and federation layer that ensures application services gain access to the actual physical repository in a seamless way and agnostic of the actual physical storage location.

3. Support for rich media and indexing & search of rich media

4. Provide e-Discovery services to support compliance investigations and its related forensic search and queries with the required performance at an affordable cost.

5. Provide a content centric workflow and document archive management framework.

6. Provide a flexible client application integration layer.

2.1.2 3.1.1 Key functional requirements:

The architecture design of an ECM system must put major focus on the performance and scaleability 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 in index with minimal time delay. Following three key non-functional requirements for the ECM systems:

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

3. The design goal is to match the load variations and minimizing system power

4. Be reliable and available such to achieve the required business continuity goals

3

ECM System Architecture and Design

3.1 Core Components of ECM Systems

Today one can choose from many commercial content management and archiving solutions that handle unstructured content [Gartner]. Despite the diversity of the many disparate ECM products on the market, they all share the following capabilities that make up a content management framework.

1. Content collection, data input and capture services
2. Content classification, meta data extraction and index services
3. A catalog database with an implied data and document model
4. Support for metadata indexing and search
5. Full Text indexing & search using clusters or full text engines (e.g. Solr et al) and distributed full text indexes.
6. Support for Multi-media indexing & search (i.e. Search by Image, Audio and Video content)
7. A heterogeneous repository infrastructure based on a storage subsystem with elaborated storage management functions i.e. the capability to satisfy hierarchical storage management needs and to hold rich media data.
8. Support of workflow and workflow engines
9. Resource management and administrative layers
10. Data Output and data delivery services
11. Flexible application and Business integration layers
12. Repositories consisting of the following system components

(a) Global Catalog employing a relational index, document model and administrative functions

(b) Content Resource Managers

i. One or more Hierarchical Storage Systems i.e. Block Storage, Object Storage, Off-line Storage (Tapes, Optical, Cloud)

ii. Local object catalog and local administrative functions

(c) Full Text Catalog that complements the Global Catalog

i. Cluster of Full Text Search Engines

ii. ull Text Index Storage Subsystems (Fast Block Storage)

(d) Core Content Services including:

i. Business Process Management (BPM) Collaboration Services.

ii. Content Collection Services

iii. Content Classification Services

iv. Content Delivery Services

v. Compliance Services

A. Discovery

B. eRecords Management

vi. Client Application Layer (Repository Abstraction Layer)

vii. Decision Support Services (DSS)

3.2 Outline of Enterprise Content Management Systems

Recall that we want to surface the differences and commonalities of an ECM system versus a Data Lake. For this we need to better understand what constitutes an ECM system by looking at typical use case scenarios and the details of typical user activities and the implied flow of data. This analysis is then used to derive the functional and non-functional requirements and help define a representative workload model for the one and the other system. The end goal of this exercise is to sketch the typical working environment for ECM systems and to compare it with that of Data Lakes.

We stated before that ECM systems are meant to manage the unstructured information their related work processes in a corporation. The main goal is to manage the content life-cycle from creation to disposition therefore ECM systems are concerned with the routing of in- and out-bound information flows, the integration with business applications and managing the end user's work-space. In this context, information must be indexed and classified for later processing and archiving. The next Figure outlines the architecture of an ECM system.

Figure 1- ECM System Architecture Overview

Figure 2 outlined the 3 relevant layers exposed by a typical ECM system which they are:

1. Repositories consisting of the following system components

(a) Global Catalog employing a relational index, document model and administrative functions

(b) Content Resource Managers
   i. One or more Hierarchical Storage Systems i.e.
   ii. Block Storage, Object Storage, Off-line Storage (Tapes, Optical, Cloud)
   iii. Local object catalog and local administrative functions

(c) Full Text Catalog that complements the Global Catalog
   i. Cluster of Full Text Search Engines
   ii. Full Text Index Storage Subsystems (Fast Block Storage)

(d) Core Content Services including:
   i. Business Process Management (BPM) Collaboration Services.
   ii. Content Collection Services
   iii. Content Classification Services
   iv. Content Delivery Services
   v. Compliance Services
      A. e-Discovery
      B. Records Management
   vi. Client Application Layer (Repository Abstraction Layer)
   vii. Decision Support Services (DSS)

Shown at the bottom of Figure 3 above one can see the content repository layer which is at the core part of an ECM architecture. The term repository depicts an abstract entity can be accessed via a so called RAL the -'Repository Abstraction Layer'. The RAL ensures that access to the actual physical repositories is location agnostic and seamless. The upper services layers when archiving and retrieving a document via the RAL do not have to know their physical storage location nor the communication protocol used by the RAL CRUD methods. As matter of fact investigation have shown, that companies due to business and legal compliance reasons must entertain more than one repository in production and that those repositories are in general geographically distributed. Thus the RAL layer not only is an abstraction but also a federation layer i.e. another key characteristic of ECM systems.

In summary the repository with its storage systems is responsible for storing and managing the actual electronic data and content, in its original and other derived formats created via transcoding facilities into renditions required to serve business processes from the upper layer.

The layer in the middle of Figure 4 relates to the core business needs of enterprises when it comes to deal with their unstructured data assets. It is the 'Content Services' layer, responsible to handle and manage the document according to specific business process logic.

These core services rely on a virtualized central catalog build upon a decentralized distributed database cluster with an implied document model that describes the persisted metadata which is required to support search and retrieval of business relevant data. This relational catalog is complemented by a distributed full text index and the relational search functions are complemented their full text search companions. In addition, very often we have also seen the requirement to support rich media data with multi-media indexing & search capabilities that utilize new and more advanced technologies to better deal with 'Search by Image', Search by Audio' and Search by Video' content.

Another trend that ECM system must satisfy is compliance and legal discovery. That is, the traditional relational catalog must be enhanced to also support information discovery services so called 'e-Discovery' functions which look more like forensic search then traditional SQL search functions needed to support legal and corporate compliance requirements as well as for creating new knowledge from crawling raw information and searching unknown patterns and the context around them. Given the size of the collections archived the e-Discovery business also must be tailored and tuned towards the ability of handling of very large result lists produced by compliance investigations and its related forensic queries with the required performance at an affordable cost.

Next we have a business process management framework that allows the orchestration of content centric business processes and their respective data flows complemented by an electronic archive management framework which can be seen as the central service that a Content Management system must provide such to satisfy the specific business needs and to comply with corporate and legal governance.

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

3.3 4.3 Workload Model of ECM Systems

A workload model is defined from key use case scenarios and describes the load distribution that the mix of operations performed by the user population and application push onto the system during normal production.

In this chapter we will use what we learned so far to look at the characteristics of content centric work processes and distil a representative mix of core operations that defines the typical ECM workload model which then can be used to compare against its Data Lake counterpart.

In the previous chapter we stated that the main purpose of an ECM system is to function as the information hub provisioning all relevant unstructured data to business processes. Ideally an Enterprise Content Management System tracks and manages all content that enters, traverses and exits the enterprise. Email is a good example, but any other data type can be used as well. As soon as the data is loaded the ECM system takes control and upon request it grants or denies access through its standard interfaces independent of data type and physical storage location. This means that information is captured, parsed, classified, interpreted and eventually archived in conformance to the class specific retention policies. From a workload perspective operations like ingest, indexing and disposition are the predominant ones but there are other operations that add also to the workload.

The key operations that contribute to the overall workload are:

1. Load & Indexing: During ingest the document is loaded in to the content repository and cataloged with the required metadata in the ECM catalog. The metadata record is then linked with the content object in the repository and guarded by a referential integrity constraint. Enforcing referential integrity between catalog and repository is necessary to avoid dangling pointer and orphan objects.

2. De-duplication: De-duplication becomes a mandatory feature when dealing with large collections in an enterprise. Team members usually share copies of document thus a

single document will be stored multiple time in an enterprise repository. Therefore once a document is stored or updated a periodic, asynchronous task will compute the content hash, look up the catalog trying to find match and then store the hash as part of the metadata record. If the look-up yields a hit then the content is removed from the repository the metadata record is updated with the storage location of the content object now shared by 2 or more records.

3. Full Text (FT) Indexing: A full text index complements the ECM catalog with relevant pieces of text extracted from the content object. The FT index complement the catalog and enhances overall search and retrieval. The FT indexing job is a scheduled with the task the periodically retrieves new or updated documents to feed the full text engine, which in turn extracts relevant business information, enriching it with context relevant tags and storing the extract it in the full text index.

4. Classification: Document classification is another important piece of required work, especially for compliance reasons. Documents and their content is always under scrutiny of corporate governance and legal compliance rules and must be classified according to corporate taxonomies. Those which do not comply have to be intercepted and re-directed and funneled into the appropriate processing queue.

5. Rendering & Trans-coding: The different business scenarios in a company often require a specific format of the document at hand for being able to work with it. Especially in cases were rich media is involved, either a document or an attachment to a document. For example, an office document that is rendered into web page or a PDF file. Or an image that is rendered into different image formats. Another good example are video files, for compliance reasons the video must be archived untouched in its original format, but for streaming it into the network or for working on it at video work places often a trans-coded version of the video file is required.

6. Content Centric workflow Content centric workflow relates to the load generated by industry specific business applications and the way they integrate with the ECM system. Despite the variety of different applications that might exist the mix of core operations that affects the repository can be reduced to the five CRUDS methods i.e. Create, Retrieve, Update, Delete and Search and the load they generate can be measured at the RAL layer.

7. Archiving: Archiving is the final state of document, which means the document has become immutable and is now stored as the original copy until it gets disposed.

8. Delivery: Active content i.e. content that is being actively processed by business processes is moved around and delivered to inside and external location creating its share of load on to the ECM system.

9. Disposition: Disposition happens when document have aged past their retention period dictated by a corporate or legal compliance policy. The act of disposing documents is called expunging and is a massive batch operation that can push an extensive load onto the ECM Catalog, Full Text Engine, the Storage Systems and the Network.

In the end, the actual workload pushed onto the system will be the sum of a mix of primitive archive operations.

3.4 The ECM Reference Model

A Content Management System delivers a set of data services that allow the processing of business information and the acquisition of informational insights from applications and users. With these characteristics in mind we can define the ECM Reference Model as follows:

Figure: 1 ECM System Reference Model

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 business value and business minimizing risk. 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. ECM systems must expose a collaborative intelligence that enables exponential growth of applications developed by the community on the platform.

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, and 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.

Thus we might conclude that given the distributed nature of the on-line infrastructure, scaleability will inherently be based on a scale-out model.

Somewhat orthogonal to the above mentioned system capabilities, from an administrative stand-point the Current ECM systems have 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 they expose a collaborative intelligence that enables growth of applications due to the collaborative effort of the internal and external communities.

Comparison between ECM Systems and Data Lakes Type ECM System Data Warehouse Data Lake Use scenarios

Table: Workload Model

System Architecture Component Data Base Catalog Full Text Index Workflow Engine eDiscovery Record Management Hierarchical Storage Application Component

4

Sum - Algorithm

```
procedure SUM( {x})
    y ← 0
    for i ← 1 : N^x do                    ▷ Time series {x} has length N^x
        y ← y + x(i)                      ▷ Summing up.
    end for
    return y
end procedure
```

Figure 4.1: Implementation of a algorithm for calculating a sum.

5
PGF-plots from python

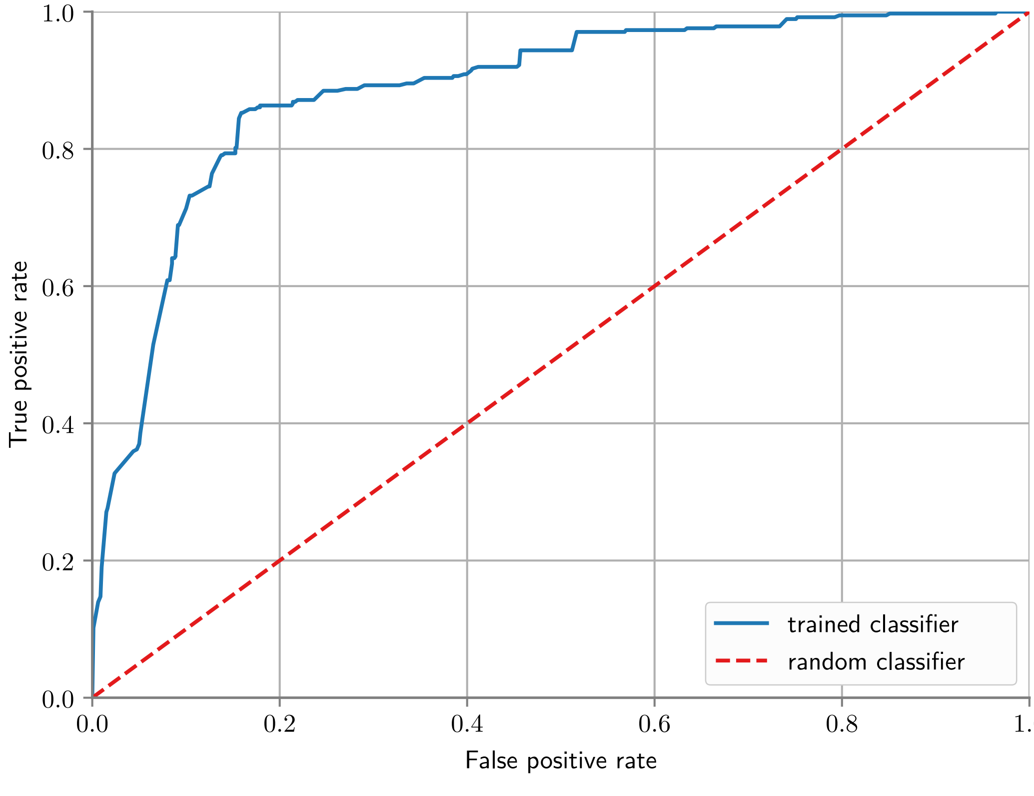

Figure 5.1: Example of using python to generate a pgf-figure which has the same fonts as the main latex document. Run python plot_exemplary_roc.py from the Python directory to generate the pgf-file.

6
Asymptote

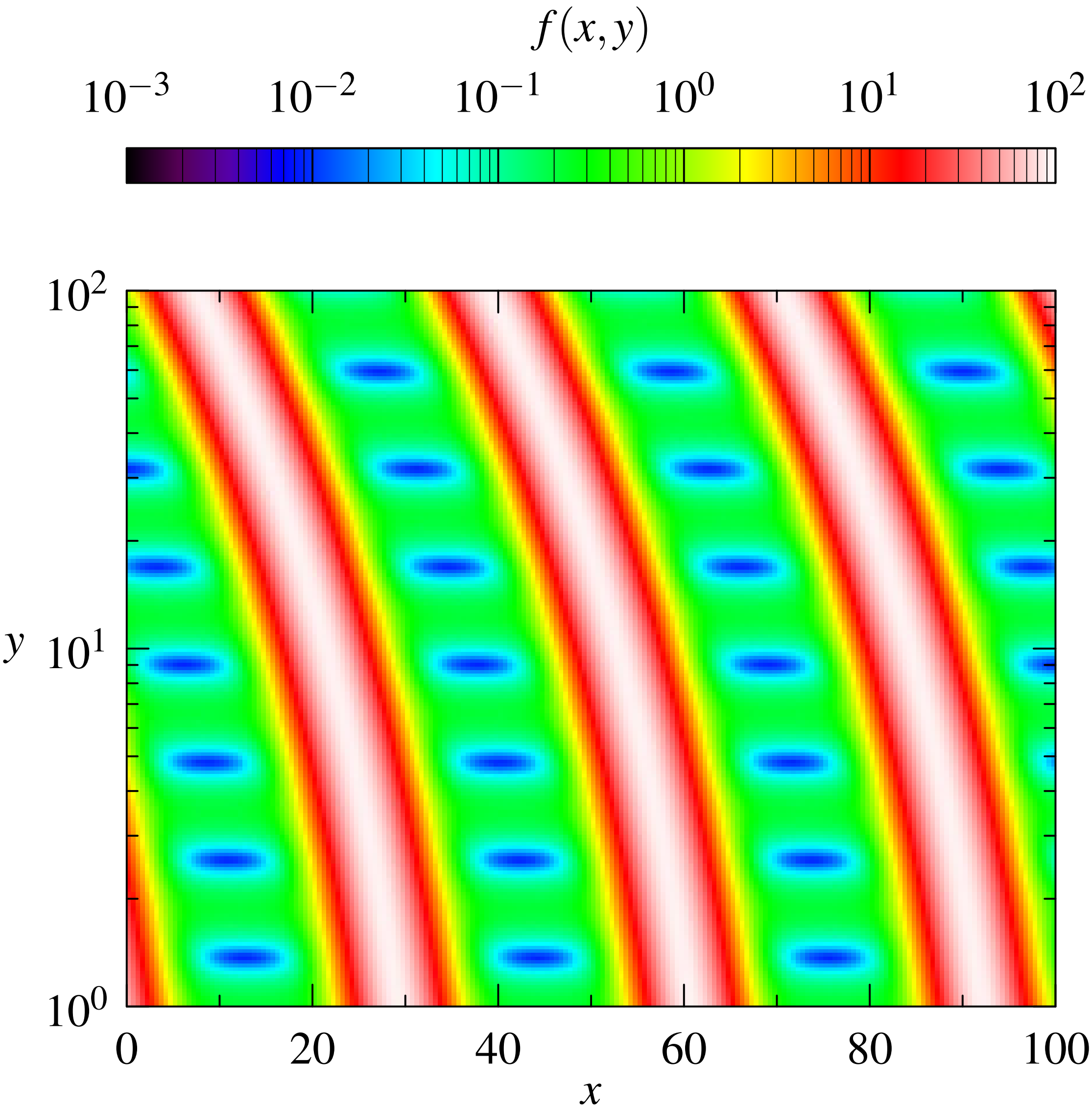

Figure 6.1: Example for plotting with asymptote

7 Discussion

8
Materials and Methods

A

Appendix A