

ContentAnalyzer MailAnalyzer

Technical Whitepaper

StoneOne - The Web Service Factory





Introduction

Companies as well as public authorities have to handle a multitude of data: scanned files, emails or documents that have to be reasonably placed within the existing IT-infrastructure. Employees often need to categorize, index, transmit and archive documents to subsequent systems in an appropriate way. At the same time, they have to follow legal guidelines and numerous regulations.

The use of electronic archive systems for audit-proof storage of documents and information is quite common. Nonetheless, the effort to actually collect and relate this data still exists. In regard to state-of-the-art technology standards, incoming data should generally be processed automatically i.e. with the help of standardized taxonomies. By using suitable classifications and indexations, up-

coming documents can be processed in a legally secure way.

Currently the IT-market provides methods and product solutions that mainly process pre-sorted and rather well structured document types like bills or standardized documents used in transactions.

In contrast, unstructured information is seldom processed automatically. This is where ContentAnalyzer and its specification, EmailAnalyzer comes into play: it decreases the amount of effort needed to classify and index data used for different purposes. Already existing methods are not replaced completely. Instead, they are integrated into the overall solution using a component concept.

Approach

ContentAnalyzer and EmailAnalyzer have to satisfy the following requirements containing two sections:

- The classification of any given document to one or more classes
- The extraction of further information (index data, processing data)

The latter are needed to find specific files or documents within electronic archives quickly or to present essential information to a co-worker immediately. Determining document types and extracting information are connected: the classification of a document into the respective document class determines the number and characteristics of the index criteria needed.



Classification

The term „classification“ implies the integration of objects into a given structure, the so called taxonomy. In this context, taxonomy can be understood as a mono-hierarchical tree, i.e. every class except the root has exactly one upper class. Terms like classification scheme, filing system or filing plan are used synonymously.

If a document is being classified according to several aspects, the taxonomy consists of more than one dimension. Thus, a request (dimension content) can also come in as an email (dimension technical format) and have urgency 1 (dimension *priority*) at the same time.

The classification should file incoming documents according to the existing taxonomy or at least provide suggestions that accelerate manual classification. Companies often use identical classifying attributes and index criteria for the same purpose, which moreover cover the same kind of documents. This fact makes it possible to develop universal taxonomies that already cover a huge part of the company's requirements. Thus, only the customer-specific parts need to be adjusted.

Index evaluation

Extraction of „named entities“

The extraction of „named entities“ is an early stage of semantic and structured extraction. It aims at finding terms within the text and matching them under the given category.

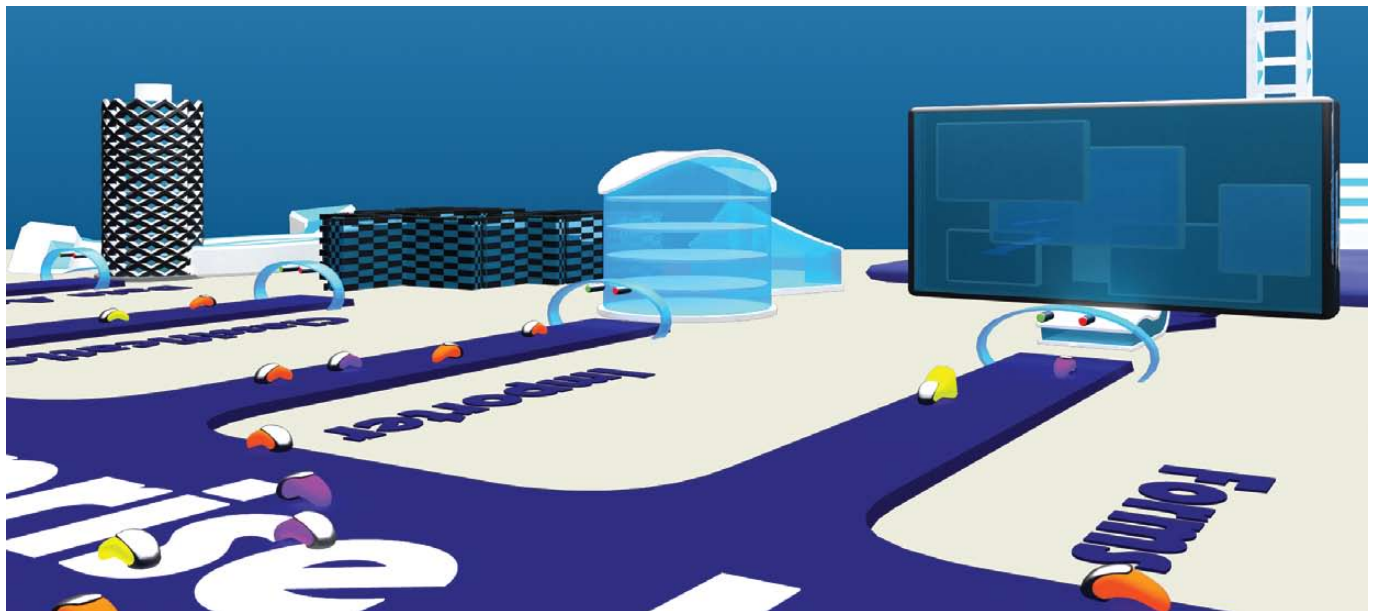
These categories are not names in the common sense of the word (i.e. places, personal names) but also contain terms like data, file numbers, names of departments etc.

The extraction data are the raw material for upcoming interpretation and structuring. They also simplify the search process. If possible, the linguistic context of the extraction data (sentence or word group) is determined as well. It often contains indicators for semantic interpretation.

Semantic and structured extraction

Based on the results of the previous steps and a system of rules, the granular information is subsumed under structures. Within this process a personal name, name of an organization, address and email address are classified under the data structure named „communication partner“.

Moreover, these complex structures are interpreted within the document's context. For example, the communication partner that had been neutral so far can now be identified as sender.



The next step includes the actual analysis, providing plenty of detailed information.

The starting point usually is the statistic analysis of the text. Based on known trained documents and suitable mathematical methods the object is being compared. Similarities to existing classes are determined. The result is a list of classes including the respective probability for every dimension of the used taxonomy.

Linguistic analysis provides „named entities“ like places, personal names or reference numbers. Moreover linguistic analysis allows the identification of relevant word groups (e.g. <place>, <date>) and pre-structured meta data (street, number, zip code, location). This process is based on a string operation on the basis of so-called token-formation, morphological information (stem and derivations) and the access to available catalogues. Additionally, semantic information concerning the respective word group and so-called tokens are provided (e.g. “Your contact <personal name>” indicates the sender).

Key word analysis verifies the occurrence of key

words (e.g. product names, laws) based on categorized lists, returning them in a standardized way.

Essential analysis concerning the extraction of detailed information is now completed. Next the gained information has to be structured and interpreted.. Now the detailed information has to be assigned using a system of rules. (Example: If a letterhead features a name right below the address, name and address are connected). Moreover, the already structured extraction data is connected to their specific meaning. (Example: If a name appears in the title and occurs in context to an address within the letterhead, this person is the addressee).

The exact classification of objects into the given taxonomy can now take place based on the found key words and the class hypothesis (already determined by the statistic analysis). Interpretation and structuring depend on the aim of the concrete case of application. The analysis of emails regarding compliance aspects requires different rules than the analysis for filing in terms of a base-taxonomy.

The detected results are being passed onto the subsequent systems in form of a XML-structure.



Architecture

The system architecture is based on the StoneOne Web Service Factory component framework, the so-called Enterprise Information Bus (EIB) (see also www.stoneone.de). The EIB serves the central control of all components and also governs the data flow among them. The standardization sim-

plifies the incorporation of new components as well as their replacement. EIB has been developed using JAVA, is scalable and designed for deployment within a distributed environment with high throughput. Thus, a huge amount of content can be processed easily by using several computers.

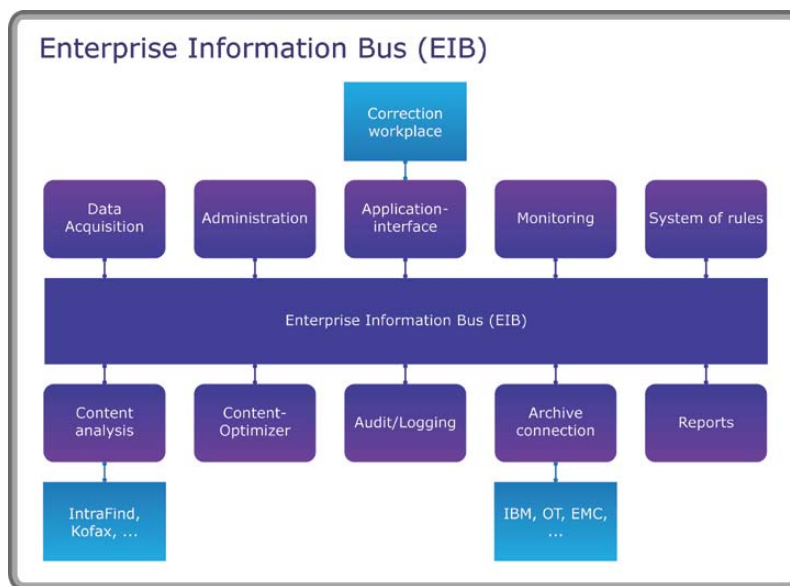


Image: Enterprise Information Bus

Regarding ContentAnalyzer a series of EIB-conform components, each with special functions, have been developed:

- Primary content analysis for statistic classification
- Optimizing content analysis on linguistic basis (ContentOptimizer)
- System of rules for subsumption and evaluation of both methods

Original or slightly modified EIB-components are used as means of integration into the given system environment:

- Data transfer
- Archive connection/Exporter
- Rules engine



Besides, EIB offers a series of supporting components to allow regular system operation.

These are:

- Administration
- Monitoring
- Audit/Logging
- Reports

For details see chapter Supportive and accompanying.

Object processing takes place as a rule-based run through the several components. Thereby the deliverables of each component are recorded and carried along within a cumulated data object. This data object is edited when processing ends and is then passed to the subsequent system. If needed for further analysis, the data object can be exported entirely as a XML-structure.

The processing order done by business and supporting components is in this case determined by an adjustable sequence control.

Special business components

Content analysis

Firstly, content analysis carries out a classification based on a statistic procedure.

Several software products available on the market have already proven their value for statistic analysis. ContentAnalyzer incorporates such a software product, encapsulates it and presents itself as an independent and completed EIB-component.

Primary content analysis uses the text of the given object as input. It delivers a list of possible classes, tagged with the determined probability in reference to the used taxonomy. This list serves as working hypothesis for following steps. The encapsulation of the software product and its connection to ContentAnalyzer allows the use of several products on the market with only a small adjustment effort.

Working method

Statistic analysis usually works based on the full text of a given document.

In order to start statistic analysis, the given text is divided into strings, normally based on defined separators. So-called anagrams are formed as an alternative or supplemental. Thus, possible OCR- or spelling mistakes have a noticeable smaller effect. This however happens at the expense of performance.

The system automatically analyzes the strings and connections within a huge, statistically relevant number of documents by means of a training run. The goal is to detect as many significant differences between the classes as possible. For this purpose, a sufficient amount of training data has to be classified manually. This process can be refined by additionally weighing individual strings. The result is a so-called training index.



Its quality can be determined by a training run using an appropriate test amount. For means of comparison, it has to be classified in advance. A possible goal is to file all documents of a given class into the same class, even if some wrong documents might slip in. On the other hand, it might be important that no document whatsoever is filed inaccurately. Both aims turn out to be mutually opposed. As long as no priority is given, the middle course is usually the best way to achieve a compromise.

New documents to be analyzed are – like described above – divided into strings and tested according to the existing training index.

If the occurring strings match the class-typical characteristics, ContentAnalyzer calculates and returns the probability for the document belonging to this class.

Using a (normally) adjustable threshold value, the final decision is made from the user's point of view. For string formation as well as for statistic use there exist several popular algorithms. In terms of statistics, so-called support-vector-machines (SVM) are currently used. Detailed information can be found on the internet using this search term.

ContentOptimizer

Optimizing content analysis (ContentOptimizer) initially accomplishes a linguistic analysis. Its base is a software product that has been developed in cooperation with the Deutsches Forschungszentrum für Künstliche Intelligenz (German research center for artificial intelligence).

a list of those „named entities“ that are relevant for extraction, i.e. word groups and partial structures including their position within the text. Indicators for later interpretations are partly included within the result.

The component uses the text of the given object in form of a data-stream as its input. It then provides

The following image showcases the process of linguistic analysis:

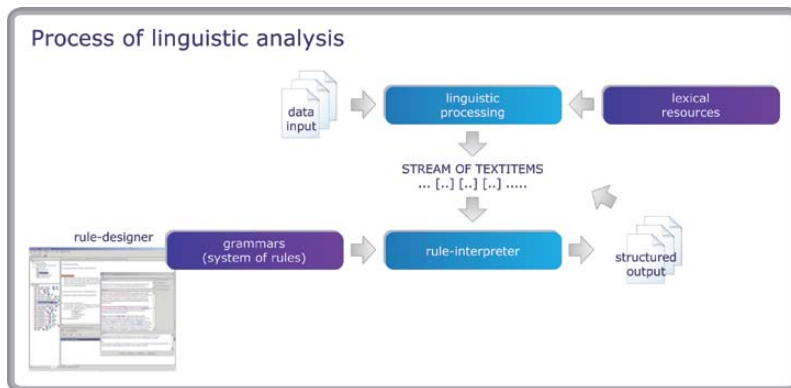


Image: Process of linguistic analysis



The input data stream first undergoes a linguistic analysis. Thereby it is separated into elemental text units, so-called tokens. The tokens have a limited number of meta data: value, position, and type-classification.

The separate tokens then are analyzed according to morphological aspects. Meta data like stem, part of speech, case, singular/plural etc. are determined.

Next, the question is whether the token appears in one of the available catalogues. If this is the

case, the information based on the key value is attached as meta data. Now the linguistic processing is completed.

Based on the meta data found by linguistic analysis, the token stream now is in focus.

By means of rules from the rule-interpreter, patterns for a token series are described. If a corresponding pattern is found, an output-structure is generated and inserted to the output stream.

The following example illustrates the approach:

```
;; hereby I/we apply for ...

request_03 :-

left side (requirement, pattern)

((morph & [STEM "hiermit", CSTART #cstart]
morph & [STEM "beantragen"]
token & [SURFACE gdet_actor]
((token & [TYPE slash])token & [TYPE back_slash])
token & [SURFACE gdet_actor])*

right side (action, output)

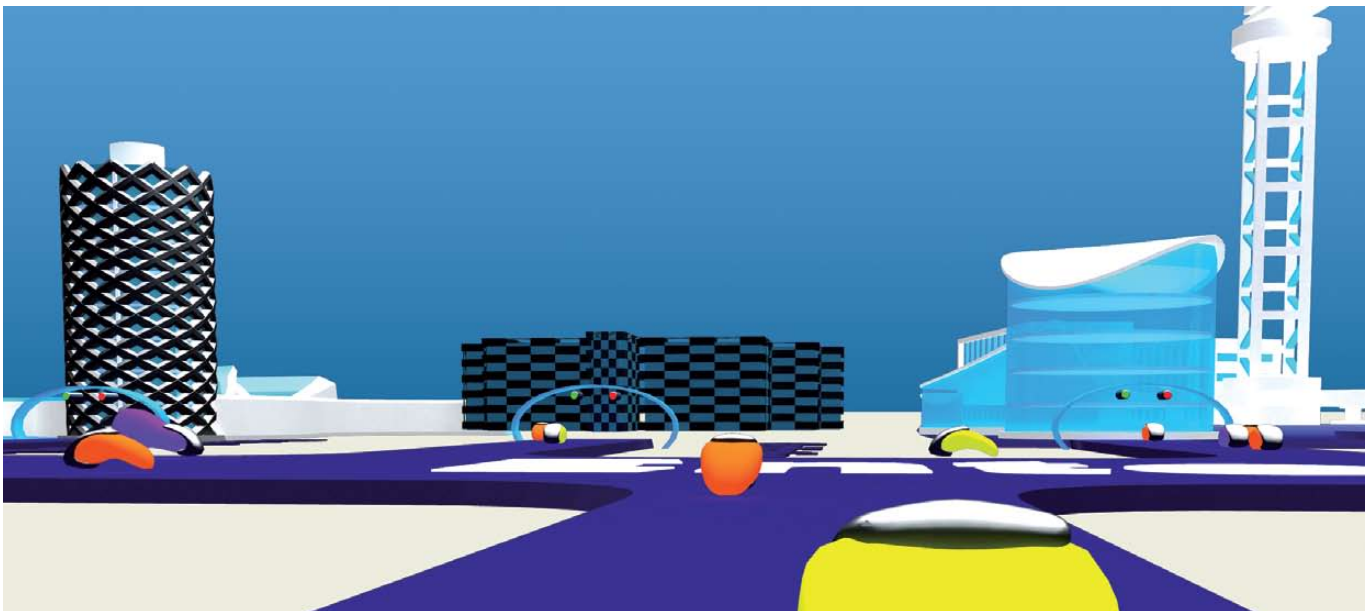
-> ne_class & [CLASS antrag_allgemein,
CONFIDENCE_CLASS medium, CSTART #cstart, CEND
#cend].
```

Image: Example of the approach

The result is a XML-structure that can be used in the following process by standardized means.

```
- <DISJ id="DIO">
- <MATCHINFO cend="63" cstart="43" end="19" id="MIO" rule="person_normal" start="18">
- <FS type="ne-person">
- <F name="CONFIDENCE">
<FS type="50" />
</F>
- <F name="SURFACE">
<FS type="Maximilian Mustermann" />
</F>
- <F name="CSTART">
<FS type="string" />
</F>
- <F name="CEND">
<FS type="string" />
</F>
- <F name="DESCRIPTOR">
<FS type="string" />
</F>
```

Image: The XML-structured result



Summary and evaluation of content analysis

A separate component consolidates and interprets the identification fragments that have been detected so far. It provides document classes and the matching index criteria together with the respective probability values.

This process is driven by the definition of respective rule catalogues which are now interpreted and processed by the EIB-intern rules engine. For every defined document class and every index structure there is a separate system of rules.

Building on the class hypothesis of statistic classification, the respective classes are meant to be assigned to the possible structured index data. At the same time, each class hypothesis is again evaluated.

Finally, the risen suggestions are again examined according to their plausibility. In accordance with the project definition, an analysis-result is generated.

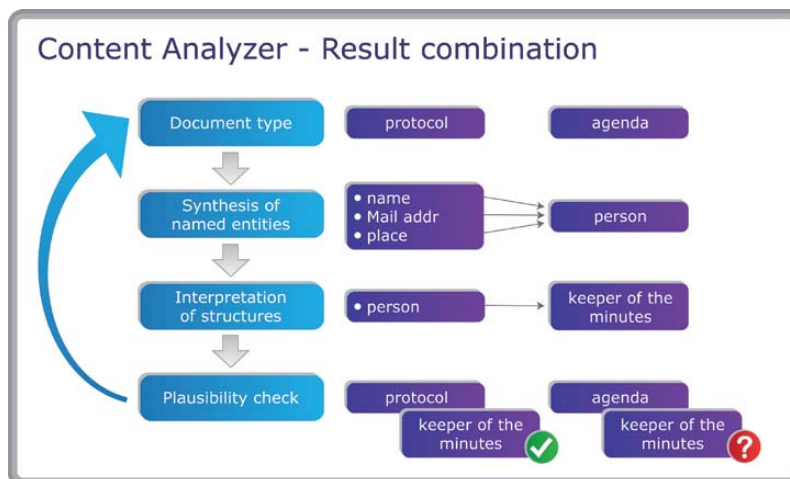


Image: Content Analyzer - Result combination

Specific methods of analysis

As mentioned in the introduction, there are perfected extraction tools for a series of special cases—formula analysis, invoice recognition etc. All of these tools can be encapsulated as an EIB-component

and used to identify document classes before or after statistic content analysis has been run. They can also be used for index identification before or after ContentOptimizer has completed its work.



Supportive and accompanying

StoneOne Web Service Factory's Enterprise Information Bus (EIB) provides a series of supporting and accompanying components that are used in varying degrees, depending on the current requirements. The following chapter explains these

components from the perspective of ContentAnalyzer resp. MailAnalyzer. For further information on the much more extensive functionality of EIB see www.stoneone.de.

Data Acquisition

Following methods are available for data transfer and analysis:

- Call of a web service done by an external system
- Polling to a defined directory
- Communication via database table
- Upload via an user interface (manual)

In any case the EIB-component Data Acquisition is the base. It transfers the objects from the system

environment and provides an internal queue for batch processing, if necessary.

If needed, format conversions and the extraction of general meta data follows (e.g. meta data of office documents, object size etc.) for the time being disregarding any special context. The result is the supply of a normalized internal object which is handed to EIB independent of the respective source format.

Data transfer

Once analysis is completed, the data object in form of a XML-file is transferred to the subsequent system for further processing. Because requirements and priorities might change with every new application, technical as well as textual customization is necessary for each specific project.

Concerning content, it often suffices to change the style sheet. A separate method – in the form of an

output-strategy- has only to be implemented seldomly.

The following technical methods are available:

- Collection/ delivery via web service
- Filing of an XML-file within a directory or within EIB-storage
- Listing results into a data table



Storage

Big applications require a safe storage for non-volatile data. The storage component allows the use of several storage systems like RAID, SAN, EMC Centerra etc. Almost any system can be connected like this.

ContentAnalyzer optionally uses storage component to store internally used data objects.

Rules Engine

ContentAnalyzer uses a defined number of partly hierarchical rules. These rules come from EIB's functional module Rules Engine and they are used to summarize and evaluate content analysis.

To cause a decision, a vector with input values is transmitted and a specific system of rules is requested.

Rules Engine administrates the system of rules and provides tools to formulate new rules as well.

Rules Engine provides the result in form of a vector, in most parts consisting of only one value. Alternatively, an operation can be initiated directly.

Monitoring/Maintenance

Monitoring signifies the graphical interface of component maintenance. The state of all components in the system is displayed in form of a traffic light. Based on that, further detailed information concerning the current status can be requested.

Additionally, an event history (start, stop, made adjustments, error messages and comments of the administrator) can be displayed.

Audit/Logging

All EIB-conformable components generate a series of system messages. These messages are stored by EIB.

Audit/Logging organizes this process, administrates the stored system messages and gives administrators access to an interface with filters and search options.



Billing/Tracking

EIB has originally been developed under the guideline „software as a service“. Hence, EIB provides a refined accounting system. This system:

- Generates invoices according to different methods of payment
- Provides detailed records of performance
- Automates the sending of invoices

Deployment

EIB is built for deployment in a multi server environment – this affects failure safety as well as load distribution.

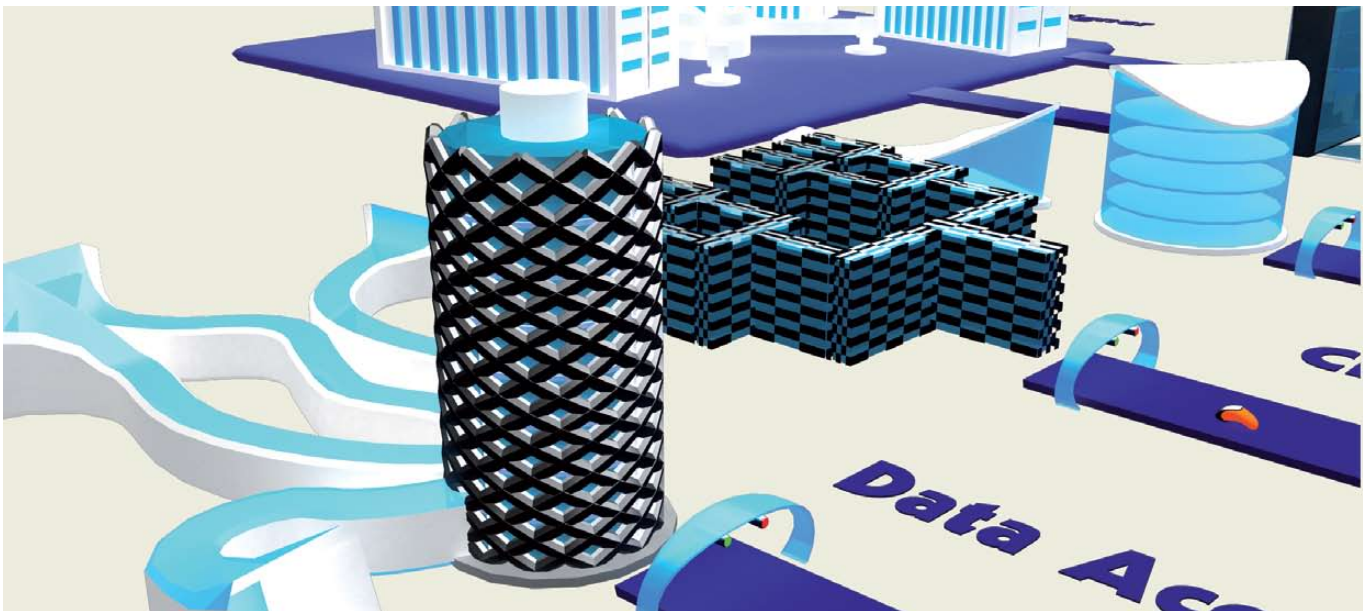
This produces the following necessity: The components and their matching data form parameterization can be distributed independently, following the given system configuration.

The component deployment supports the initial system installation and automatically generates a system-wide update of all corresponding components whenever a new application is added.

User-Administration

User-Administration administrates client- and user information centrally.

ContentAnalyzer uses this function to regulate access to i.a. data objects, taxonomy, training indexes and system of rules depending on the client.



MailAnalyzer

MailAnalyzer is a sub form of ContentAnalyzer. It is used to interpret and classify email-texts and – if given – of their attachments.

Its main application is the compliance of the regulatory requirements, i.e. correct taxonomy, classification into the respective project files. Those are prescribed by legislation and cannot be complied using only audit-proof storage and a full-text search.

Analysis of email-texts

MailAnalyzer is based on emails that already have been archived.

The actual email-texts are transmitted to MailAnalyzer as sheer text objects by the archive system, using the functionalities of data transfer (as described in chapter Data Acquisition).

Email-texts and attachments

Emails with attachments are usually called „compound document“. This kind of document always contains of two or more documents, the actual email-text and one or several attachments.

If the archive system transmits the attachments in text form and labels them correspondingly, the documents are processed as described in chapter Analysis of email-texts.

If attachments are transmitted in their source format, the texts are either extracted immediately (e.g. word.-documents) – using the application's

Other applications of MailAnalyzer are i.a. the correct and automatic forwarding of incoming emails to the actual addressee or to various connected systems. In doing this, the relevant meta data or the risk evaluation of incoming and outgoing emails is considered, including the possibilities of following actions. Next, the three variations of MailAnalyzer are described.

MailAnalyzer determines the document class and all connected index values, referring to the used taxonomy. It then transmits the evaluated data back to the archive system, using the export-interface (see chapter Data transfer). If needed, return values can be completed with information about the actual probabilities.

interface – or the attachments are transformed into a pdf-file and then extracted via OCR.

The processing of arbitrary attachments is far more complicated, because attached images or drawings cannot be classified or interpreted without their context. In this case, the project needs to be adapted.

Especially in the case of emails that have been stored in an archive system there already exist prototypical solutions. Their refinement will be done within the first projects.



General approach of MailAnalyzer

The extraction of meta data out of the mail server – e.g. like described above, to determine meta data or risk evaluation of incoming or outgoing emails including the possibilities of further actions – calls for adjustments affecting the project. Those adjust-

ments have to take place outside MailAnalyzer but rather focus on the used mail server.

Inside the mail server the following logical components are needed:

Mail-listener

Mail-listener supervises the income and output of emails within the server and initiates processing done by the EIB-based MailAnalyzer. At the same time it prevents output as long as there is no analysis result, for example it blocks incoming mails against any user interference. From a technological point of view it resembles a web service consumer using a web service provided by an EIB-component (here: Mail-Acquisition).

Web-Service-Provider

Web Service-Provider provides web services. These web services are used by an EIB-component (here: communicator) to return results of analysis and action codes.

Time-out-Handler

This component supervises the time for email processing done by the MailAnalyzer. In the case of timeouts, it carries out appropriate measures, thus making sure no emails are left.

Inside MailAnalyzer the following logical components are necessary:

Mail-Acquisition (on base of Data Acquisition)

This component offers a web service that is invoked to initiate email-processing.

Special-Communicator (on base of Application Interface)

This component embodies a web service consumer who transmits the results of analysis and/or processing instructions.

Mail client- Application (on base of Application Interface)

The component runs on a web server and provides applications that are used by the Mail clients on the base of a link inside the email. The link has been inserted automatically.

Storage

Storage deposits objects temporarily (e.g. for the time of analysis and the post processing of an email through the user). Additionally, this component can also be used as an interface for email archiving.



stone one
The Web Service Factory

StoneOne AG
Keithstr. 6
10787 Berlin
Fon: +49 (0)30 469 99 07 18
Fax: +49 (0)30 469 99 07 19
info@stoneone.de
www.stoneone.de

