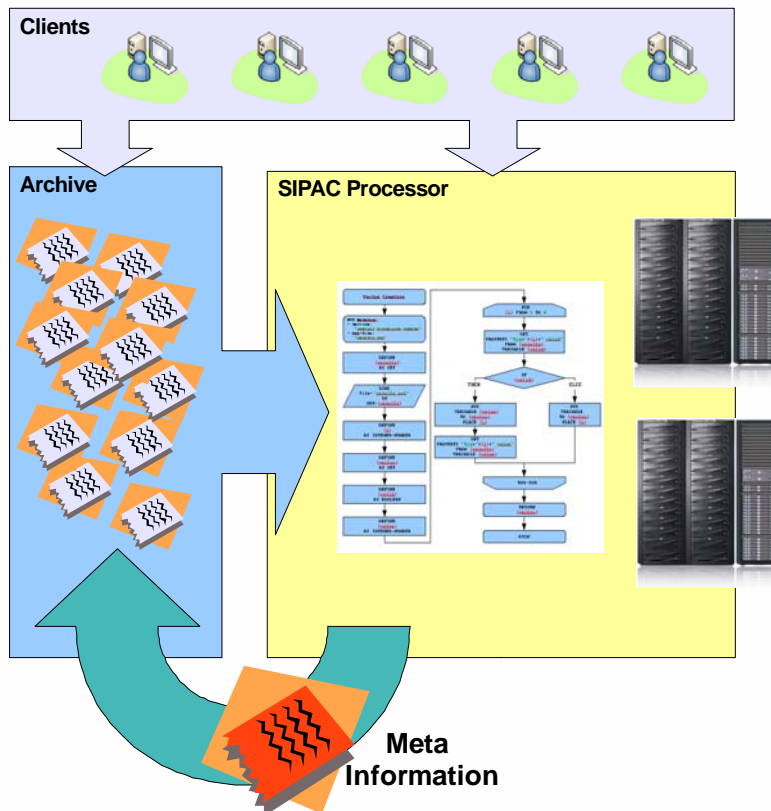




WIR VERSTEHEN DIE ZEICHEN DER ZEIT
KEEPING PACE WITH THE SIGNAL OF TIME



SIPAC

Signal Intelligence Processing for Analysis and Classification

Framework for Mass Data Processing with
Modules for Data Storage, Production
and Configuration

Introduction of SIPAC

SIPAC is a **framework to process mass data** in a well defined, configurable workflow. It offers solutions for processing of different types of data and provides quality assurance issues for processing systems. The framework allows for a high degree of flexibility in many regards (size, types of information to process, workflow) and is widely open for scalability.

Any type of data processing software can be embedded. If an executable file exists that can be called and parameterised from command line, its textual output can be turned in meta data (attributes referring to files). Another way is the execution of special processing software under control of a macro recorder and player software.

Key Features of SIPAC

- Framework for mass data processing and classification
- File oriented processing (Input: files, output: files with meta data)
- User configurable flows for processing sequence
- Server client concept (different configurations possible)
- Scalable and flexible
- Archive for storage of: signals, configurations, flows, other data
- User interface for operator
- Various applications available
- Configuration for speech classifiers:
 - Trainable classifiers
 - Training environment (MELANIE)
 - Corpora archive
- Configuration for customer specific tools:
 - Open interface for customer's programs

Today's Challenges

Today we are confronted with a huge amount of information transmitted over most different ways. Individuals busy with reconnaissance tasks cannot process these mass data. Methods **separating important from irrelevant information** are essential in a reconnaissance or investigation environment.

Automation of labour-intensive routine tasks is the key for solving the problem in an economical way. An important step to handle large data volumes is to **enrich raw information by additional meta data** in successive processes to support further evaluation steps, such as classification and retrieval. Lots of common software tools to contribute such meta data are available in the market, many even for free, however usually as standing alone programs needing interactive operation and delivering results in proprietary formats.

In the following, SIPAC is introduced by applications in which classifiers are incorporated. This is the most complex type of application. However, SIPAC can also contribute for effectiveness by file processing (like file identification), demodulators or decoders.

Solution: Automatic Processing of Mass Data

Sensors as receiving systems or human investigators intercept and collect different types of traffic and information (e.g. files of confiscated volumes), which can be classified.

According to the classification results, meta data are generated and assigned to the input files that are passed to further processing systems. The meta data may decide over the way of the input file through the SIPAC system, decide which steps are undertaken to gather more meta data or content information from the input file.

While the original file remains unchanged, the system evaluates its content and can generate:

- **meta data** for sorting and prioritisation all data
- files with **comprehensible information**, e.g. from language point of view
- files with **readable information**, e.g. demodulation

Different techniques can be applied to generate meta data. Especially methods of **file analysis**, **speech processing**, **text processing**, and **image analysis** are helpful, the profits of which are in regards of augmentation of efficiency and abilities. Here are to mention manifold common solutions for methods of content processing of files, text, audio, and images.

Undeniably, the fusion of uncorrelated sensor outputs, with all the resulting possibilities of the additional evaluation of technical parameters, opens extended abilities in **criminal investigations**, **military threat recognition** and in the long term observation.

Supporting programs for different subtasks conduct routine jobs by automatic operation and thus help the evaluation operators and analysts at jobs that can be done more efficiently by supplementing tools. Thus, information technology makes way for more intensive detail analyses, because routine jobs are done automatically.

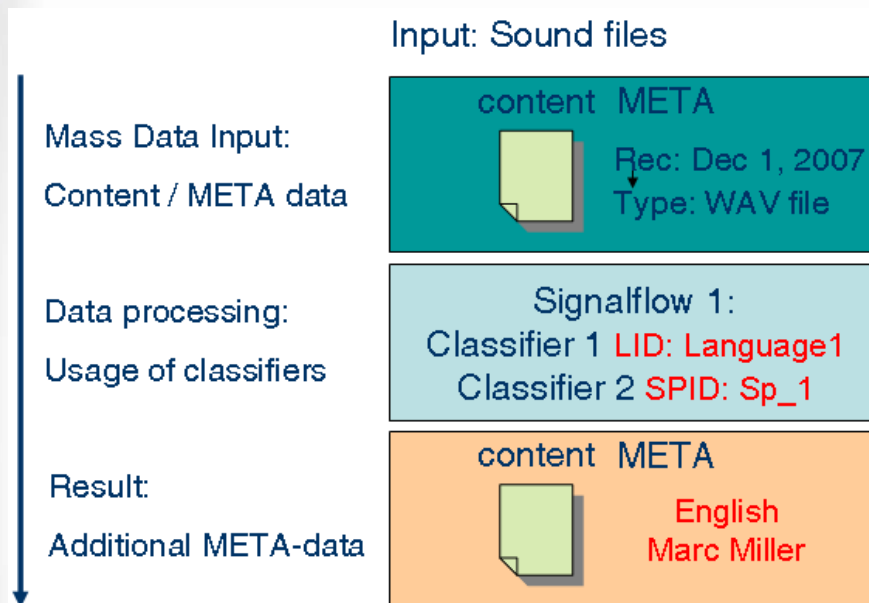
Overview of SIPAC

SIPAC is a **framework** to classify and manage masses of any type of electronically available information (“documents”), and to apply **processing** of **additional meta data** based on the contents of the input files.

It is a **building block system**, consisting of a data storage resource, a processing resource and clients, into which customer or Medav developed components can be embedded. This allows to provide solutions according to the needs and requirements of customers.

The **SIPAC process** starts with gathering masses of input data as files, which are enhanced by meta data according to their origin. Subsequent processing (signal flows) analyses the files by classifiers. The results have wide effect on the processing flow, which adds more and more meta data. All additional meta data are a result and are assigned to the input files. Therefore, the main focus of the SIPAC system is to **improve the information about the incoming signals**. Different classifiers or programs generate additional information. This information also may prepare and speed up subsequent analysis steps (e.g. search for all files with a specific language).

The following illustration shows an example:



Processing step	Content of processing step
Input	Sound files with content and basic meta data
Data processing	Computation of additional information (here: language identification and speaker identification)
Result	Additional meta data: language of the sound file and the speaker

The SIPAC system allows to embed **customer specific tools** and to apply them within a **homogenised processing context**. This concept can be used in very manifold ways, beginning with simple acquisition and collection of files up to the complex data or content information processing, defined by a program flow using special software.

Within SIPAC, documents are any data structures that can be stored as file and their meta data, independently of format and content.

SIPAC is **flexible** in regards of **operating systems** and **computing power**. With limitations in speed and performance, it can be run on a single computer, but it can also be spread over a computer cluster.

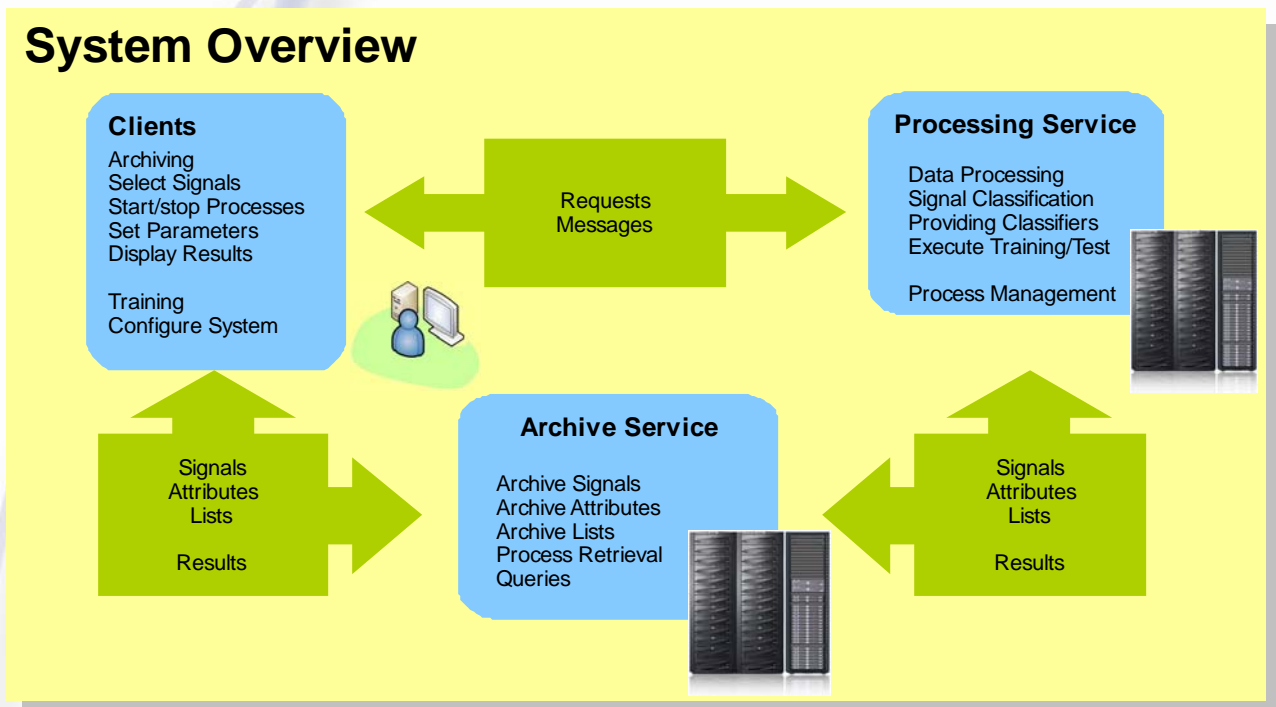
The SIPAC system consists of the following **basic components**:

- **Clients**
- **Archive**
- **Processor**

The SIPAC components are explained in the following sections, followed by a detailed description of the system functionality and sample configurations.

System Components

The following figure shows the structure of the system naming its components.



SIPAC (processing) service

The SIPAC service is the central processing unit of the system. After starting and passing input documents to it, the system processes them according to the active setup and signal flow. It runs the classifications, collects the results in the activity log file and returns them to the clients for display, to the archive server for storage, or customised output. The functionality depends on the used processing modules, beginning with basic functionality (such as file type recognition) up to complex processing modules developed either by MEDAV or the customer.

The concept of the service allows for distribution of the processing tasks to different machines of different platforms following a master/slave concept.

A list of processing modules is available on page 12.

A tool to configure signal flows is available.

SIPAC Clients

The clients run the graphical user interfaces for the processing and archive services. The system provides user interfaces for the different tasks and use phases: training of classifiers, configuring the signal flows, starting and controlling the production, viewing results and files and retrieving data. Also functions for system administration are available over the clients, such as editing flows, and managing not only the archive but also the complete system.

There are two sets of clients: the operator client for the operators who observe the daily classification processes, and an administrator/expert client for system setup and control.

The basic client comes with numerous masks and functions, which can be enhanced and extended depending on the functionality, the solution and customer requirements.

Archive service

The archive is the data storage of the system, and thus connecting all parts of the system. The archive stores the incoming data as well as processing and configuration information, and all results and intermediate results of processing.

For fast data access to the operators, it provides up-to-date means of retrieval, such as categorisation of data and full-text-search.

Interfaces

All components communicate by TCP/IP protocol. If distributed machines are in use, LAN connection is required accordingly.

Software Modules

The system provides the following software modules:

- **System Basic software:** consisting of archive and SIPAC server software. Optionally, the software is available as master and slave versions to distribute the required computation power over several machines.
- **Operator Workplaces/SIPAC Client Admin/SIPAC Client Operator:** client software to allow users to setup and control the system, to observe the processing and to view the results.
- **Processing modules** by MEDAV, as described below, or external modules.

Hardware Modules

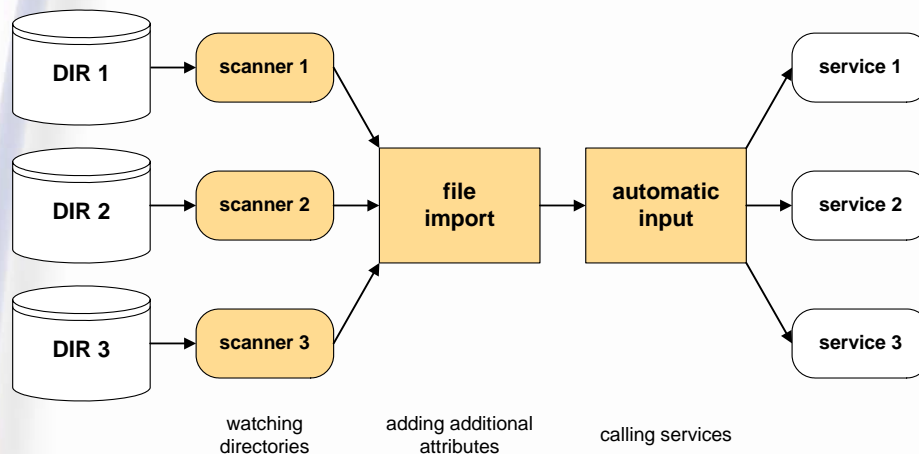
There are no specific hardware modules needed. The system can be run on a single computer as well as within a cluster consisting of as many computers as needed to fulfil the requirements in data processing within adequate speed.

SIPAC Modules in more detail

SIPAC Processing

The processing modules can be concatenated to a *signal flow*, the parts of which can be distributed over different computers. When passing through the flow, more and more meta data and perhaps content can be extracted by the modules, which is collected and posed to the operators.

The system either fetches a file from a given directory, or an operator imports it into the system, which then processes it according to the configuration. The configuration specifies which modules (*services* in SIPAC terminology) are applied to which type of file and is widely flexible.



A graphical configuration tool helps to setup and maintain the flow. Thereby, it is also possible to distribute the processing tasks to different computers within a network to achieve balanced computing load for high system performance on the one hand, on the other hand this flexibility allows the system to support processing modules only available for distinct platforms (for example: Windows or Linux).

SIPAC writes all processing information including results and intermediate results to logging files during production, and also can show relevant information directly on screen. So, it is possible for operating staff to react immediately to important results, and analysts can track back important information later.

The system is widely customisable to different specific inputs also by training. For example, if a certain speaker is searched and some training material is present, a speech classifier can be trained to detect exactly this speaker in all incoming audio files.

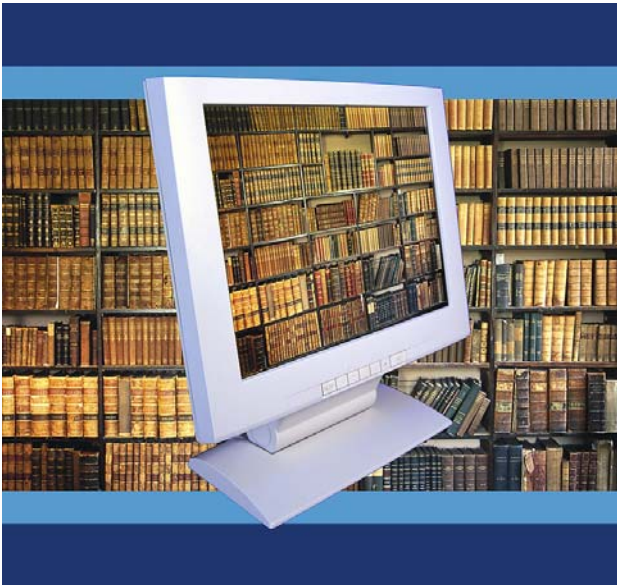
To achieve this flexibility, the usage of the system is divided into different phases:

- Training the classifiers (if needed)
- Developing a setup (= flow)
- Production of incoming files

However, SIPAC is not limited to classifying audio files, but supports different classification techniques. It allows the user to decide what classification technique is best in its environment and supports the user at the training and test of own classifiers. Another way is to use the general classification technique of Support Vector Machines (SVM) if suitable training data is available or extractable.

The system is available in variable extents, beginning with a small single workplace system up to systems distributed over different networking machines, partially manned and observed and other computers working stand-alone.

Archive service



The archive service permanently stores all kind of information and provides sophisticated means for information retrieval:

- it stores the incoming signals
- it manages all data and meta data upcoming during processing and production
- it stores different system parameterisations or setups (regardless whether they are currently in use).

In principle, any kind of file can be stored by the archive service.

The archive service processes search and retrieval requests from the client, for example,

if operators are searching for input signals or results.

The archive server is a stand-alone product which is used in numerous MEDAV solutions and thus can be used as link between the different applications.

The core functions of the archive server are:

- Saving and compressing incoming files as archive documents
- Restoring the files
- Labelling the documents with an internal unique ID for reference
- Adding meta data to documents and storing them separately as attributes
- Storing and assigning attributes to documents also if they added later on
- Checking incoming text documents for keywords and storing them in an internal index for fast full text retrieval
- Protection of documents against changes
- Deletion of documents can be prohibited ensuring archive integrity
- Assignment of documents to different types allowing to assign proper processor applications for the documents (for example, viewers or editors)
- User and password management
- Access documentation by log files
- Synchronisation of a local working directory with archived documents
- Summarising document sets (chunks) for storage on external media and vice versa
- Email interface for data and for remote control
- Capacity limited only by hardware
- Programming interface

Storing Wideband Signals

Wideband signal records can, depending on the bandwidth and the length of the record, demand very much disk space, need very much network resources for transfers and computing power for processing. Usually, a big amount of data must be processed within short time, mostly to detect and extract narrowband signals from the wideband signal recording. The strategies of the file archive as described above are not suitable for this type of information and would strongly obstruct other system activities.

Therefore, MEDAV decided to encapsulate storage services for wideband signal processing in the separate wideband acquisition service package, called *ReProS*, that also can be run on a separate server with its own network connection. Its task is to store and manage wideband signal data and to make them comfortably available to processes and users without unduly loading network and computer system.

ReProS can be combined with a common file archive system. Access coordination is performed automatically by the system.

Acquisition is performed by use of MEDAV Virtual Devices (VD), whereas, server control is done by the command set of the server. Cooperation of both is done automatically:

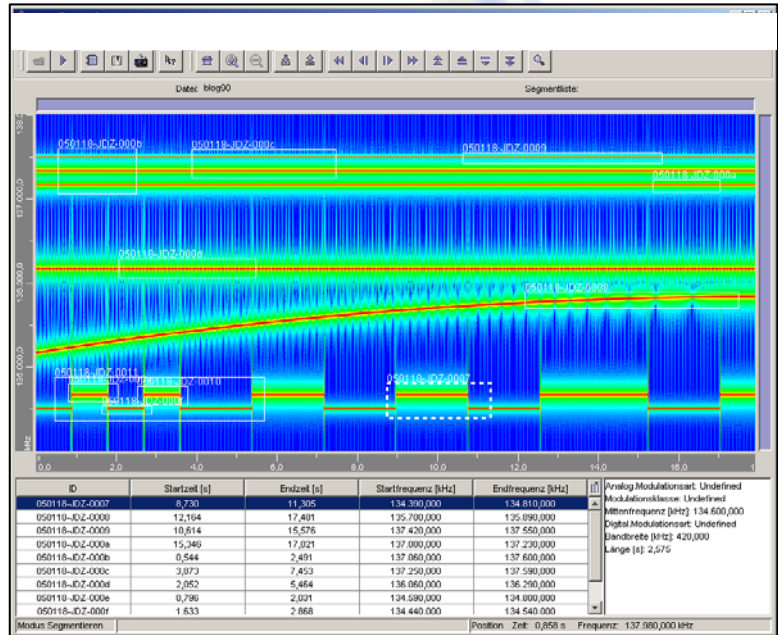
- To record a signal, the server invokes the VD "FileOutput" with the address information from where the signal is going to come. The data transfer is then conducted directly between server and data source in parallel or, if the network setup enables for this, out-of-line of other network traffic. The data are written directly to the target file system via the VD.

In parallel to the transfer, the server divides the signal into segments and computes spectrogram information from it, so that it can easily be visualised to users by graphics. In addition to that, navigation data (zoomed or adjacent areas of the signal or overviews in lower resolution) are computed and stored together with the signal to allow users to navigate through the signal fast and comfortably.

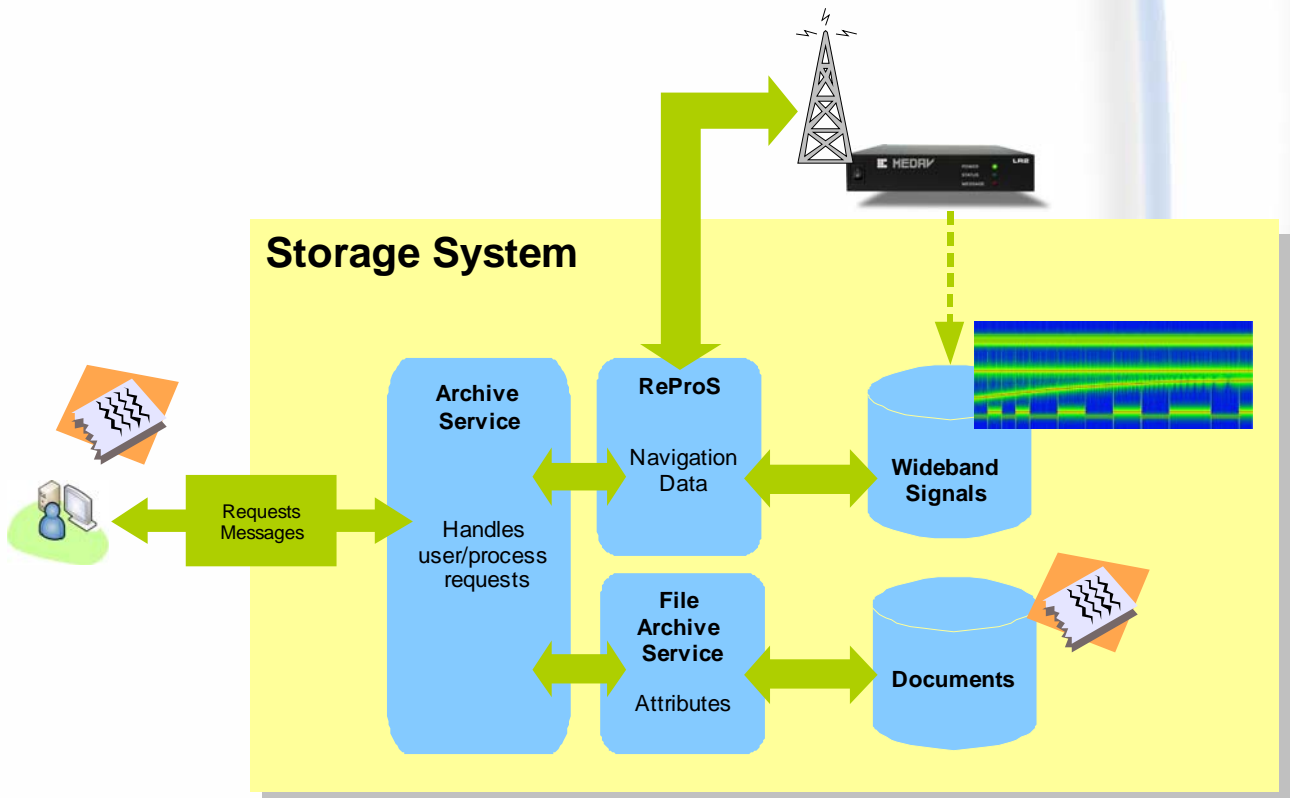
- To restore a signal, the server invokes the VD "FileInput", which picks the desired segments from the signal and navigation data and returns it to the enquirer. In this way, only a small relevant part of the wideband signal is needed to be transferred, while the complete wideband signal file may remain on the server.

Nevertheless, the user has still the chance to see signal overviews, or to request adjacent signal areas for closer examination.

The *ReProS* server provides a programming interface with a set of commands for control of storage and retrieval wideband data and is thus used in numerous MEDAV applications if they provide wideband components, however, can also be provided to interface third party programs and processes. Additionally, MEDAV provides with CCI-Offline a software program with graphical user interface, which is tailored to operate the acquisition server by a user and is embedded in different MEDAV systems and solutions.



The following figure gives an overview of the cooperation between the file archive service and *ReProS*.



Processing Modules Overview

The productivity of the system is based on the processing modules that are used, and the quality and quantity of training (so far training is needed for the processing modules). All processing modules are software modules.

The table below lists and briefly describes the currently available processing module packages. They are introduced in more detail in the sections afterwards.

Processing Modules for Audio File Analysis	
Training environment is included if applicable.	
Signal Enhancement	Improves the signal file quality in order to improve the results of subsequent classifiers.
Speech Detection	Identifies speech sections in audio files.
Language Detection	Identifies languages in which is spoken in audio files.
Speaker Detection	Identifies speakers who speak in audio files.
Topic Recognition	Recognizes topics about which is spoken in audio files.
Keyword Detection	Recognizes given keywords and phrases in audio files.
Gender Detection	Recognizes the gender of speakers in audio files.

Processing Modules for Image File Analysis	
OCR Methods	Recognises images with high textual content and extract textual content from images. Additional licence for external OCR software required. No training required.
HiTex	Trainable classifier to detect if an image contains a high textual content. The classifier is pre-trained, the training environment is part of the shipment, training data are available on request.
StegoBMP StegoJPG	Search for steganographic changes in image files. The classifier is pre-trained, the training environment is part of the shipment, training data are available on request.
ImageChecker	Special tests of different image formats (header information, attachments, insertions, filled areas, pads, etc.). No training required.
JPGLab PNGLab BMPLab	Support libraries for image analysis of JPG, PNG, and BMP files. No training required.
SVM-Library	Support library for Support Vector Machine (SVM) Classifiers. Training environment is included.

Diverse File Processing	
No training required.	
File Type Recognition	Identifies the file type, not only by file suffix but also through evaluation of the first bytes (header) or the body of the file. Additionally MIME type information is set.
Unpacking Of Archives	Unpacks archive files (ZIP, ARJ, RAR, LHA, UC2 etc.).
Matlab	External tool used for different purposes.

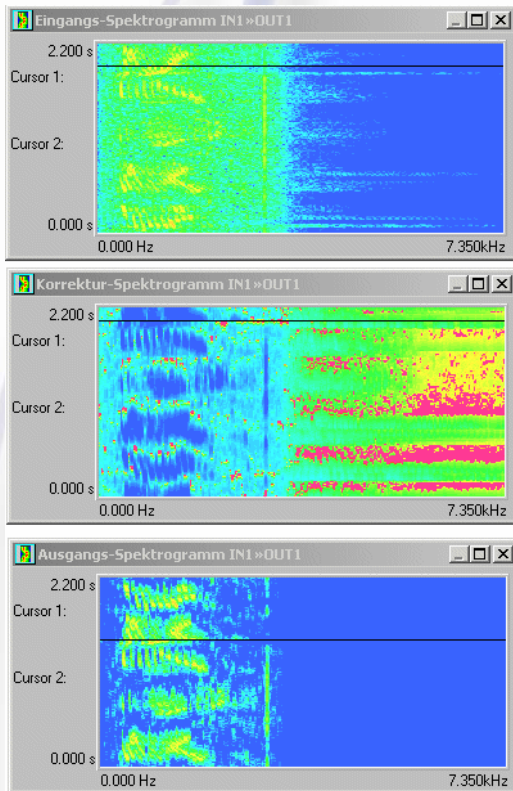
The processing modules are described in more detail in the following. On interest, also refer to the corresponding data sheets.

Note that processing modules provided/developed by the customer can be incorporated into the SIPAC system besides the processing modules offered by MEDAV. Ask MEDAV for details or support.

Processing Modules for Audio File Analysis

The following processing modules analyse audio files.

Signal Enhancement



The module for speech signal enhancement is a non trainable analysis tool, in order to filter the disturbing noises from the recorded language signal. The advantage is in the comfortable hearing quality, not in an enhanced recognition rate.

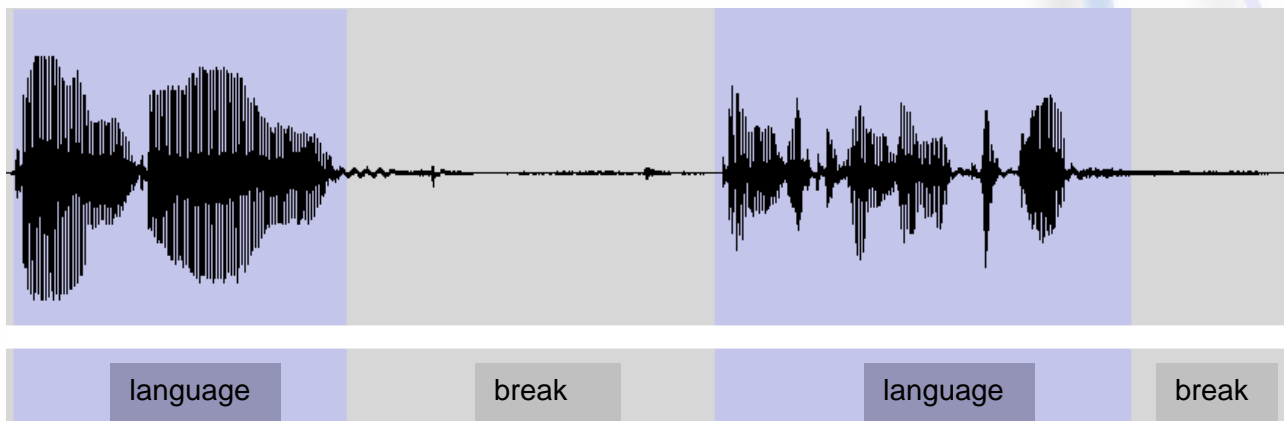
The signal enhancement takes place through dynamic and adaptive rejection of non speech signals, e.g. noises, continuous wave (CW-) emissions, constant single tone signals, chirp- and FSK- signals. Automatic Gain Control (AGC) - effects can be equalized.

In the adjoining example the sonogram of a noisy signal is represented in the upper image, in the middle image the sonogram for disturbance signal and in the lower image the image for remaining, enhanced speech is represented.

Speech Detection

The module for speech detection determines, whether the available file is a speech signal. The advantage of this classifier is in the automatic rejection of non-language segments in a file, e.g. speech pauses, data transfer or channel noises, so that downstream speech signal classifiers get an optimized signal.

The signals file can include one or several speakers or languages. The signal is analysed in segments of optionally simulated or automatically adapted lengths and a decision to speech/non-speech is performed for each segment. These decisions are documented in a protocol file.



The module includes two classification techniques:

- A parametric approach, realized as not to be trained, language independent, threshold determined. There, special signal characteristics are assessed, e.g. the signal energy and the normalized basis frequency. Its ability to surely classify bell sounds as non-speech-signals must be highlighted.
- A trainable classifier, which sub-classifies the signal through assessment of acoustic characteristic in speech and non-speech segments. The trainable classifier can lead to an enhanced recognition rate subsequent to its optimization for special applications against the parametric classifier – however connected with training expenses. This application is specially recommended for channels, where often bad language quality occurs.

Language Detection

The module for language identification supports the determination of the language prevailing in a speech sample from the list of trained alternatives. The speech samples to be assessed can include one or several languages. In a multi language signal a language is determined for each segment. This segment can be determined by the user or be set automatically. For each learned language the calculated probability is determined. The result is the name of the language with the highest probability, where rejections in case of insufficient security are possible.

By connecting modules for speech detection (module "speech detection"), automatically non-speech-segments are determined in the file and excluded from the analysis.

Three variants for language identification are offered, which are optimized in view of identification rate or processing speed:

Variants	Identification quality	Processing speed
SQ – Single Quant	good	excellent
MP – Multi Path	very good	very good
SP – Single Phone	excellent	good



Speaker Detection



The module for speaker identification supports in case of task to automatically determine the speaker in a speech signal. The speaker is either known by the training data or assigned as unknown. In that way e.g. the concerned files are assessed further in content or time profiles for activities of individual persons or person groups can be created. The verification of a speaker is possible.

The speech samples to be assessed can include one or several speakers, even multilingual. In a signal with several speakers a speaker per segment is defined for optionally simulating or automatically adapting segment lengths. The result of classification is the probability to have identified one or several speakers in the speech sample. Below a minimum probability for known speakers the speaker is classified as “unknown speaker” to the classifier.

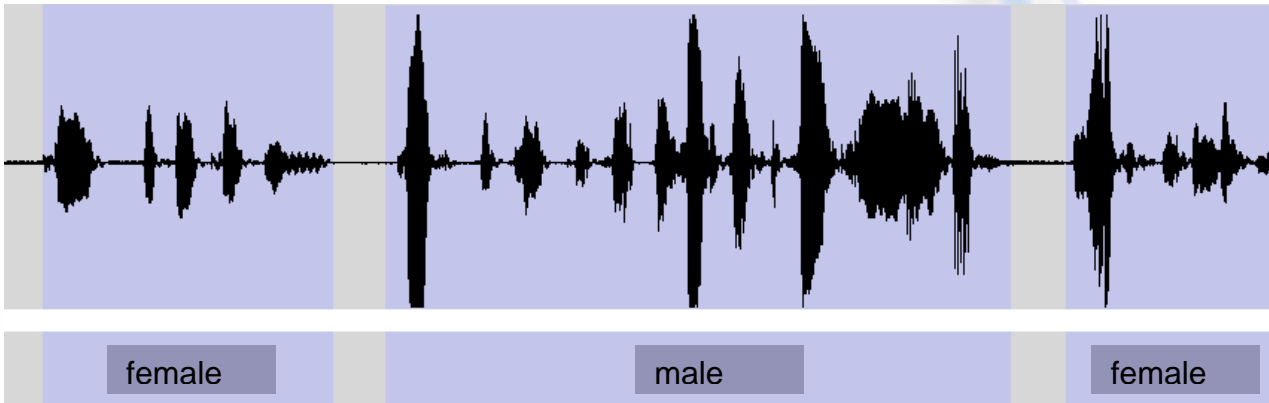


Gender Identification

The module for gender identification determines, whether the speaker is male or female. The advantage of this classifier lies in its very high processing speed in combination with a very good identification rate. In that way it is advantageous to reduce or sort the quantity of data, which is processed relating to calculating time of spending classifiers.

The speech samples to be assessed can include one or several speakers or languages. In a signal with several speakers or languages, the module defines the gender of the speaker for optionally simulated or automatically adapted segment lengths, in each case per segment.

The result of classification is the probability that the speaker is male or female.



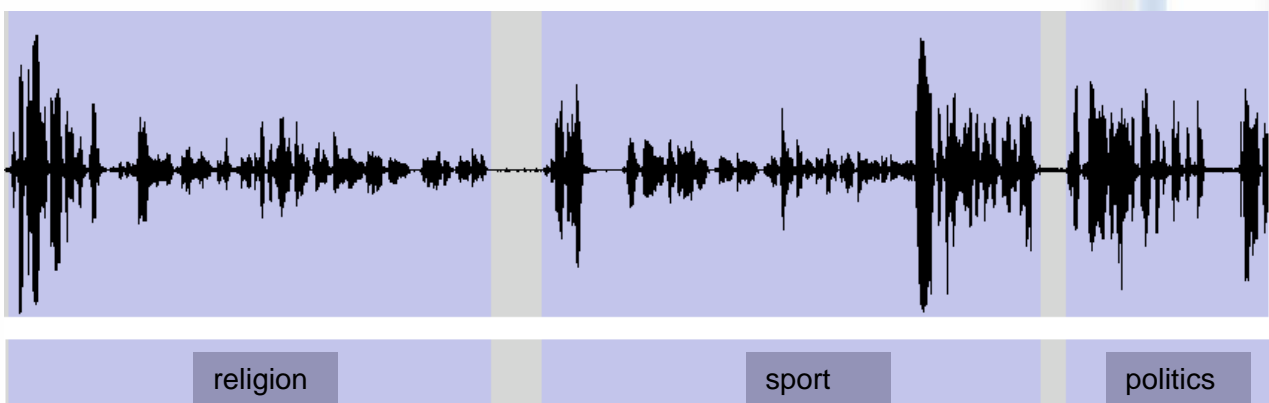
Topic Recognition (“Word Spotting”)

The module for key word and phrase identification addresses the automatic finding of given key words and phrases. A classifier is trained for the required languages. Additionally a list of keywords and phrases to be searched is created. The classifier searches these words and phrases in the speech signal. For an enhancement of the recognition rate the specification of a negative word list (“background words”) is possible, in order to reduce the mixing up of searched words with similar words and phrases.

The speech records to be processed can include one or several languages, where the key words and phrases are searched only in that language, as they are entered in the list.

The result of the classification is a label track with the recognized key words or the frequency of individual key words. Parameters for the sensitivity to identify a key word or a phrase can be given.

The words are entered in Latin characters; for Arabic we recommend to the Buckwalter transcript, for Chinese the Pinyin transcript. In other cases we recommend to transcribe words as closely to the pronunciation as possible.



Processing Modules for Image File Analysis

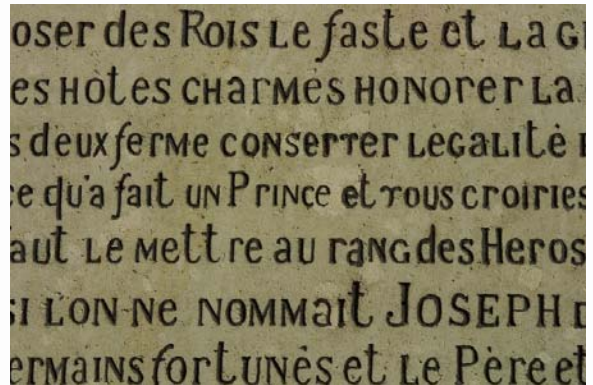
The following processing modules analyse image files.

Optical Character Recognition (OCR) Methods

The OCR module recognises raster graphic image with high textual content and extracts textual content from the images.

This is, for example, useful as a pre-stage for processing of scanned documents.

There is no training needed for this module.



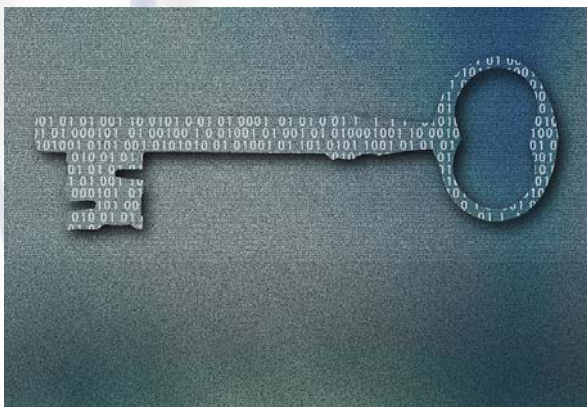
Searching for Text Information within Graphics

The *HiTex* classifier analyses raster graphic images for embedded text information.

Senders of spam email often try to bypass common spam filters by embedding text information in images, because this way the text information is harder to find and to analyse for keywords to decide whether the email carries spam. Today other spam filters cope with this situation applying OCR algorithms as known from common scanner software with moderate success usually.

The *HiTex* classifier developed by MEDAV, however, uses the proven method of feature based training and testing. It is a fast reliable graphics filter, which can, for example, be used to separate spam from other email, or to search graphics data bases for scanned documents, and for similar purposes.

Searching for Hidden Steganographic Information within Graphics



The *StegoBMP*, *StegoJPG* classifiers analyse raster graphic (bitmap) images for hidden information embedded by means of steganography, that is: hidden secret information is embedded in other data. The secret information can be extracted only by the intended recipient (or the sender) however other persons hardly notice the existence of the secret information.

Bitmap images are commonly used as container images for secret information because they generally provide sufficient redundancy that can be used to hide secret information without modifying the image content significantly. The presence of the embedded information remains invisible to the beholder.

The steganalysis classifiers use the proven method of training and testing. It is a fast and reliable classifier detecting bitmap images containing hidden information.

Searching for Irregularities within Graphics

The *ImageChecker* module searches image files of different formats for peculiarities as attachments, insertions, filled areas, pads, etc.

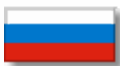
The module does not require training.

Processing Modules for Text File Analysis

Language Identification within Texts

This tool identifies the language in which a text is written. By default, the tool is familiar with a set of languages, from which it either selects the recognised language, or it rejects the file.

The module recognises certain words that are typical for the referring language, and on the statistical distribution of language-specific recurring n-grams. N-grams are specific character combinations, for example, the 4-grams of the word *combination* are: *comb*, *ombi*, *mbin*, *bina*, *inat*, *nati*, *atio*, *tion*. Each language provides specific portions of certain n-grams that allow to identify a language reliably.



The input for the module is a text file, and, if available, a character code specifying the character code used in the file. By default, the tool assumes UTF-8 (which applies for most files today) or tries to detect the used character code. The module returns the recognised language and the character code.

Key Term Extraction/Summary

This tool identifies single word key terms in a text, either by extracting the terms from the text, or by applying a filter bank of relevant terms to be searched for. The algorithm conducts a shallow morphosyntactic analysis of the text.

The input for the module is a text, a language, and, optionally, a recognized topic. It returns a list of terms that describe the document best (either as mark-up in the text, or as a list with links to the occurrences).

Entity Recognition

This tool identifies names (entities) in texts. Names can be of different type: persons, places, companies, institutions, dates, currencies. Other entities can be added.

The algorithm uses a dictionary of names and name indicators, for which it searches in the input text file.

The input for the module is a text, a language, (a list of name types to be recognized). It returns a list of names recognized, or alternatively a text containing mark-up for the different name types.

Term Substitution

This tool translates terms in texts from a source languages to a target language. Translation is performed applying a two-level approach:

- Level 1: Electronic Dictionary Source language. 30.000 terms, linguistically annotated. General terms (about 20.000), domain-specific terms, covering the areas of customer interest (about 10.000 special terminology).
- Level 2: Term Substitution. Term-based translation, insertion of key terms into foreign language text

Translation

This tool translates texts from a source languages to a target language. Translation is performed applying a three-level approach:

- Level 1: Electronic Dictionary Source language. 30.000 terms, linguistically annotated. General terms (about 20.000), domain-specific terms, covering the areas of customer interest (about 10.000 special terminology).
- Level 2: Term Substitution: in the source texts, a term-based translation takes place by insertion of key terms into foreign language text. The terms of a lexical field are substituted by a correct term in the target language.
- Level 3: Machine Translation: the hybrid translation engine architecture features foreign language morphology, bilingual dictionary lookup, and target language generation using statistical language-model.



An additional tool can be offered to add and modify dictionary entries.

Diverse File Processing

The following processing modules conduct analysis functions on different types of files.

File Type Recognition

The module detects the file type and the MIME type of an input file. To determine the file type, it does not only consider the file suffix but also evaluates of the first bytes (header) and/or the body of the file.

Unpacking Of Archives

The module unpacks archive files (ZIP, ARJ, RAR, LHA, UC2 etc.). The unpacked files can be forwarded to other modules for further processing.



Application Example

The approach, in which SIPAC is applied, is divided in three important phases:

- Training and Test
- Configuration
- Production

They are briefly described in the following.

Training and Test

Some processing modules need training before they can be applied. This is true especially for speech processing modules because even those must be adapted to their specific intent of use. For example: if the speaker identification classifier is intended to be used to identify a specific speaker, it needs training with speech samples of this speaker, so that the classifier can learn which speaker it has to identify. To ensure good training success, the classifier can then be tested using test material similar to the training material.

MEDAV provides extensive training and test environments on the base of a graphical user interface within in the SIPAC system, as well as command line oriented for automatic training purposes, for example MELANIE. Furthermore, MEDAV offers processing module training, especially for those processing modules that cannot be trained at the customer's site.

Configuration

Configuration specifies how the SIPAC system processes automatically the type of input file.

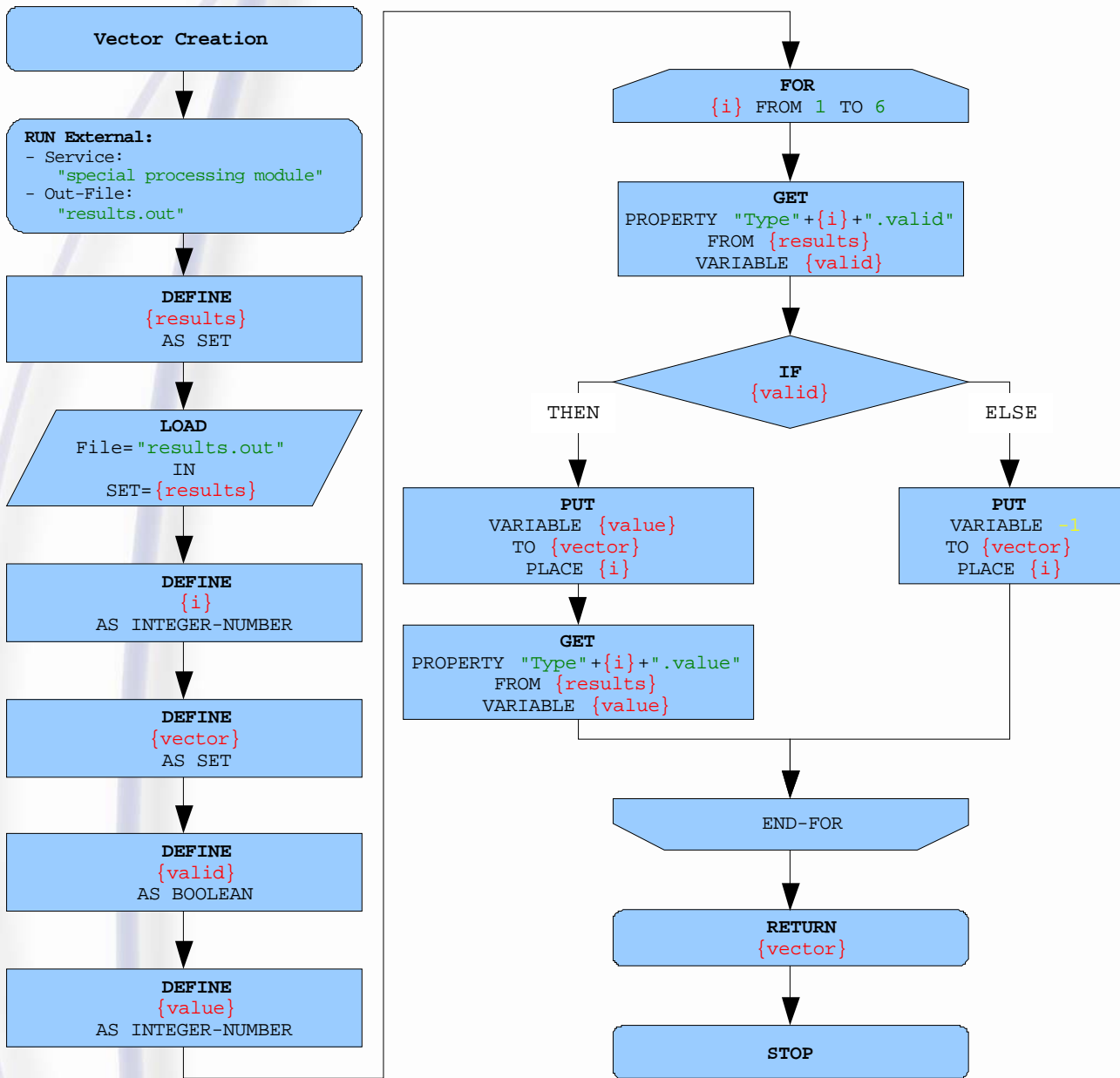
As there are different types of input files (audio and image files of different format and different sources), the SIPAC system can be configured to process them in individual ways. Examples could be:

- Extract all files from a ZIP archive, and process the files according to their file type.
- On audio files from source A apply speech enhancement, then try to detect the language, and if it's Arabic, try to detect speaker X.
- Try to detect keywords or find out the topic about which is spoken in audio files from source B.
- If an image file from source C contains embedded text, sort it out to a spam directory.

The SIPAC processes are organized in *nodes* that are concatenated to *flows*. Each input document is passed through from node to node according to the selected flow, and each process adds the information that it extracts from the input according to its task and algorithm as meta data into separate result files.

Therefore, for each type of input file, a flow can be defined, similar to a batch operation mode. Thereby, a graphical flow editor allows to setup the configuration: the steps and decisions of the chain are represented by blocks, that are concatenated and linked.

The diagram shows a sample flow:



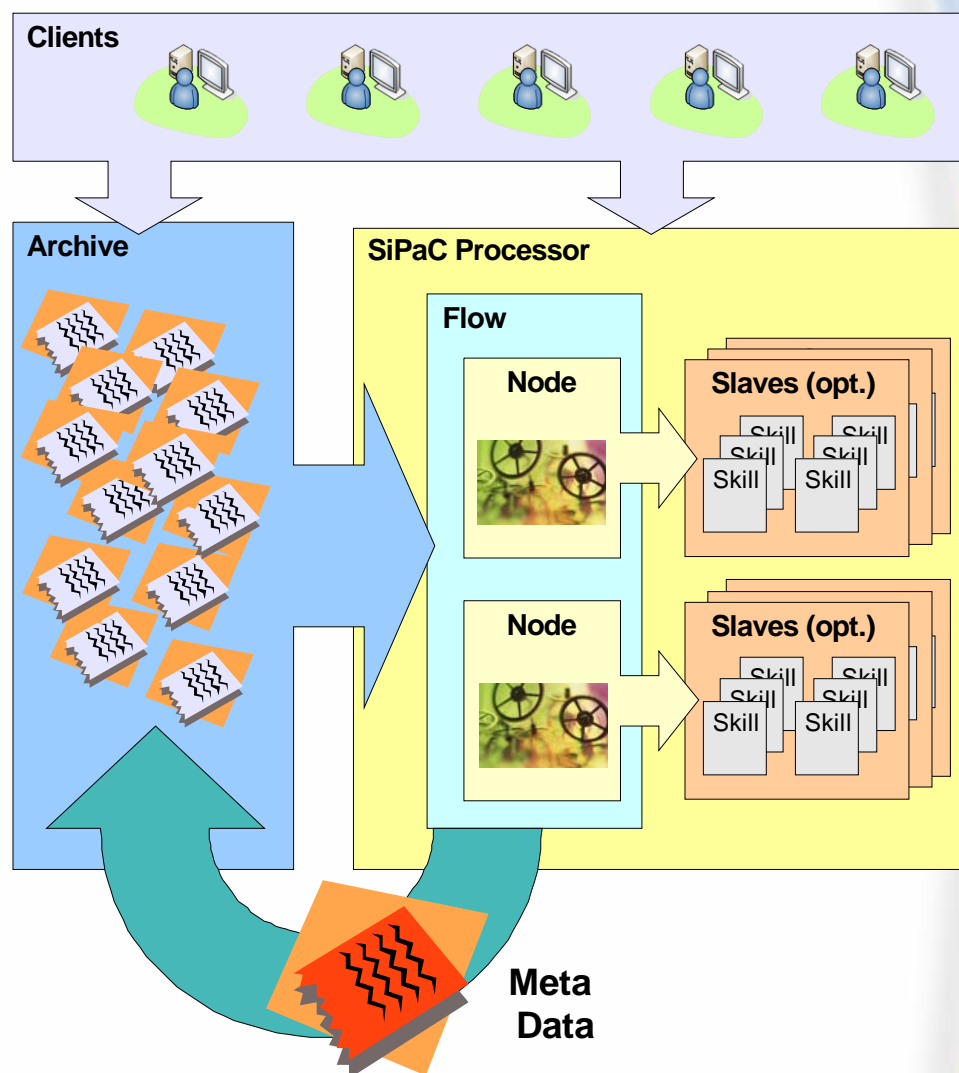
The result is an XML file containing the processing information. This file can then be transferred to the SIPAC server and be selected for production.

In the next steps, nodes and skills are created and distributed over the available slaves.

Flows concatenate the nodes to a processing sequence and always run on the SIPAC server. A node may address slaves, which can (but not necessarily must) be located on different computers, running different operating systems. The assignment of the slaves may be programmed firmly in the configuration, but can also be conducted dynamically during the process, depending on the availability of resources.

The slaves finally execute functions to process the input document. Each function is embedded in a *skill*, extracting information from the document and returning it as meta data. The meta data are collected and returned to the SIPAC service, where they can be stored by the archive service, are accessible to the client workplaces and thus to the operators.

The figure shows the principle by a configuration example:



Note that the figure shows an example only, and that it does not state, which process runs on separate computers.

Other as shown in the figure, the number of clients is not limited to five, the number documents in the archive, the number of nodes per flow, the number of slaves per node, and the number of skills per slave are not limited. Limitations result only from computing power and performance.

The SIPAC system manages the data occurring in the process, including meta data (“attributes”) and file contents. It is selectable whether or not results are stored in the archive.

The cooperation and location of flows, nodes, slaves and skills are subject of the configuration.

Production

Production is conducted in the following steps:

- The user passes one or more input files to the system or the system read them from a specified input directory.
- The input file is assigned to a service or a flow, that processes the file according to the configuration.
- The process is logged in an activity log file. The file can be viewed while the process is ongoing.
- Intermediate results and final results are stored in result files. Intermediate results can be viewed before the process is complete.

For production, the setup is activated on the SIPAC server and the acquisition environment stores its files into the SIPAC input directories. The input directories are observed by SIPAC and whenever an input file arrives, it is processed according to the setup. Alternatively operators can load a single file or list of files and feed it to a selected process.

Then, the process runs according to the configuration. The production results are stored in log files and can optionally be displayed in the graphical user interface of the client. Intermediate results of a flow's nodes that completed their process can be viewed before the flow is finished but while the other nodes are still running.

A final step is conditioning the result data, so that they are useful and meaningful.

Result output depends widely on the requirements of the operators, respectively on the type of expected results: this can be a modified output file in the archive, or the attributes (meta data) the system investigated on the input file. It may be sufficient to check these result files from time to time by full-text retrieval or checking the activity log files.

In other cases it might be good to see selected information in the moment when they are detected, or even to be alarmed or to pass this information to other, external systems that further process the result information. All this can be achieved by developing special result output programs that collects the desired information from the SIPAC processor and presents them, for example, on a webpage. Also customer specific modification of the SIPAC client may be worth considering.

By default, the client provides an archive view, showing all results files associated with an input document, which then can be easily called and viewed. Depending on the type of result, special viewers can be selected (assumed they are installed on the client computer).

Sample Configurations

The SIPAC system is available in different configurations: from very small extent with only one commercial computer running all parts of the software, up to large systems consisting of a farm of machines connected by LAN.

Two samples are described in the following sections.

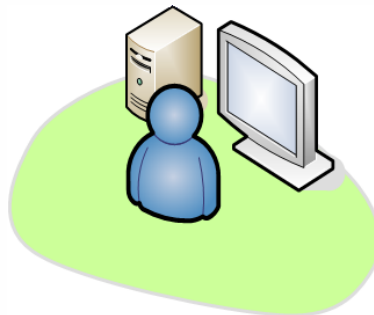
Standalone Solution

The standalone solution means that all parts of the system are located on a machine. The solution consists of:

- Basic Software Package, consisting of SIPAC Archive and Server
- Workplace Administrator/Expert, operator console for SIPAC Client and Admin User Interface Software
- Classifier packages (MELANIE, image processing, etc.)

The following tasks can be performed:

- File analysis and packer support
- Training and testing classifiers
- System Configuration and Management
- Production

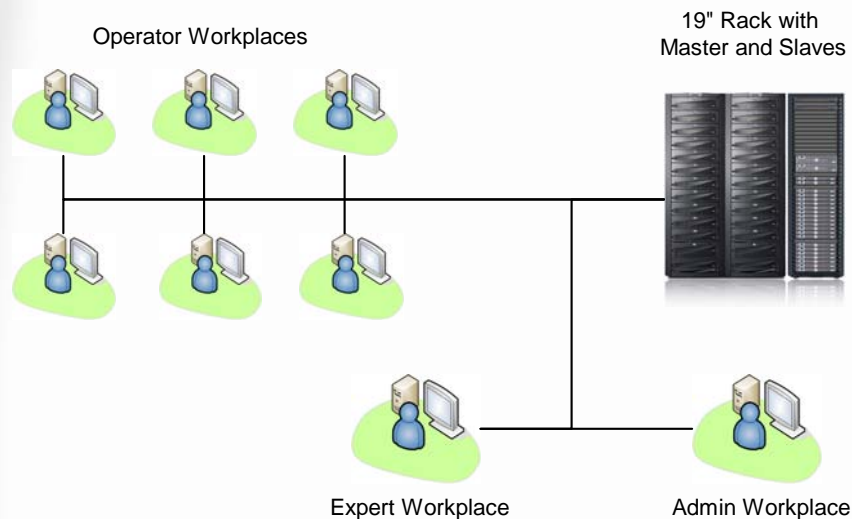


Maximum Configuration

The maximum configuration consists of a network of multiple PCs, and a number of operator and administrator workplaces:

- Basic Software Package, consisting of SIPAC Archive and Processing Services
- 19" rack with server PC, slave PC(s), speech production package
- Operator workplace(s) and SIPAC Operator Client software
- Classifier packages (MELANIE, image processing, etc.)
- Administrator/Expert Workplace: operator console for SIPAC Admin Client software

The following figure shows a sample configuration with all available components connected by LAN:



In principle, it is possible to add as many operator workstations and slave processors as desired.

The tasks at the distinct workplaces are:

- **Operator Workplace:** importing input documents, information retrieval, choosing workflows, starting and stopping processes and services
- **Expert/Admin Workplace:** collecting speech data, training and testing classifiers, changing configurations

The servers run archive services, as storing and managing data, execute the SIPAC services.

Technical Data

Components of the System	
Client Software	Operator and administrator user interface to control processes and servers.
Archive Server Software	Storing and managing data and meta data. Data retrieval on operator request.
SIPAC Processing Server Software	Executing processes, controlling and coordinating data flow.
SIPAC Slave	Executing processes (skills).

Requirements	
Hardware	<p>Commercial PC, operating system any if able to run Java applications. TCP/IP network as required.</p> <p>The number and computing power of the PCs depends on the stage of extension and customer requirements.</p> <p>A second system may be useful for testing and development purposes to avoid to interrupt production when the system is adapted to new demands and requirements.</p>
User requirements	<p>Operators viewing results need to be able to use graphical user interfaces. No programming skills are required.</p> <p>Administrators and experts maintaining the system and adding new processing modules need skills in common computer network maintenance and administration, and easy programming skills (Perl).</p> <p>Experts and Developers adding new SIPAC services to the system need software engineering and development experience (Java, Perl).</p> <p>MEDAV offers training and support for users, administrators, experts and developers.</p>
External Processing Modules	<p>Executable files with textual input and output over command line.</p> <p>Programs that can be executed under special processing software under control of a macro recorder and player software.</p> <p>Tools that use socket communication or CORBA.</p>

Our Goals



Technology

... in the development and in the company management is state-of-the-art and represents a high level.



Quality

... in all divisions of our company is considered as the pre-requisite for a risk-free and successful cooperation with our customers and business partners.



Position in the market

... is affected by the experience made in signal and information processing. We are ready to face up to the competition.



Service Available

... is comprehensive, complete and tailored to suit the requirements. As a systems vendor, we offer standard devices, systems and services.



Employees

... form the roots of the company and render the services necessary for maintaining and expanding the technical basis and a trustful cooperation.



Growth

... on a stable technical and economical basis at home and abroad is our aim.



Trust

... vis-à-vis our business partners and within the company is the basis of our business.



Compliance

... with excessive sensibility and compliance with German and international export regulations we act on a worldwide basis.

MEDAV GmbH

GRÄFENBERGER STRASSE 32 - 34
D-91080 UTTENREUTH

HOMBURGER PLATZ 3
D-98693 ILMENAU

TELEFON: +49-9131-583-0
FAX: + 49-9131-583-11
E-MAIL: info@medav.de
www.medav.de

w711od.089