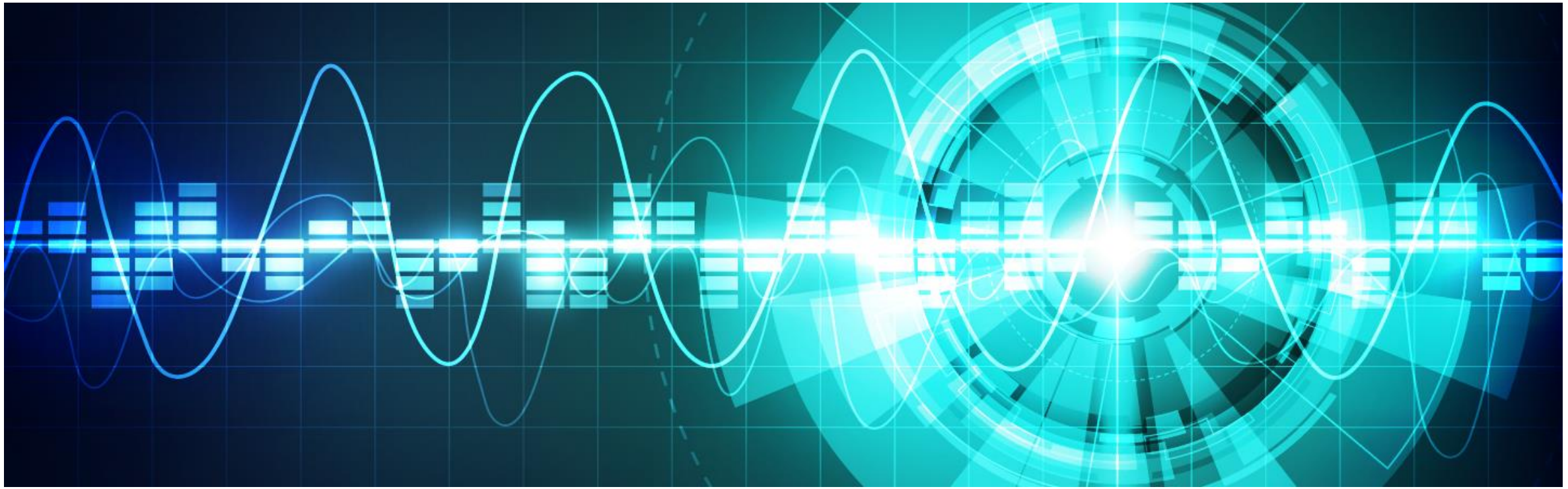
SPEAKER IDENTIFICATION

Who speaks?



AGENDA

- Application examples
- Categories of Speaker Identification
- Speaker Identification Task
- iVector Paradigm
- Demo
- Open Source

Use Case IFThisThenThat (IFTTT)

Use Case IFThisThenThat (IFTTT)

- Based on the recognition results, not only speaker recognition but also beyond, applications can be programmed to react on specific metadata. The result can be an alarming of the user, but also automatic sorting and filtering of content. Based on such a mechanism, monitoring applications which analyse and evaluate all public transmissions of a company spokesman, politician or other famous persons can be implemented.

Find audio Quotes

Find audio Quotes

- Metadata containing speaker information allows the search and retrieval of media files in which a certain speaker speaks.
- In cases where the metadata speaker information is available on segment level, a direct jump into the speakers segment is possible.
- In combination with speech recognition requests combining speaker and speech information can be made and thus enable the user to search for audio quotes.

Speaker Clustering

Speaker Clustering

- Speaker clustering assigns each segment of an audio signal to a group, depending on the speaker which speaks in each segment. The algorithm assigns every segment of a single speaker the same group. For each different speaker, different groups are used, thus there should be a correspondence between speaker and a group.
- The information of the speaker clustering results are only relevant within a media file and are not relevant for search requests. But the information, if visuallized, provide the user with additional means for navigation and enabling them to skip unrelevant speaker segments and speed up their work with the media file.
- Results from the gender detection can be used during speaker clustering to prevent the assignment of two segments into a single cluster, where the gender does not match.
- The speakers which are to be grouped do not need to be known beforehand.

Speaker Verification

Speaker Verification

- The task of verification checks if the claim that the speaker is a specific person. The difference to speaker recognition is, that you do not need to find likely possibilities of speakers which could match, but instead you know who the person is supposed to be and check that claim based on the speakers model known to the system.

Speaker Recognition

Speaker Diarization

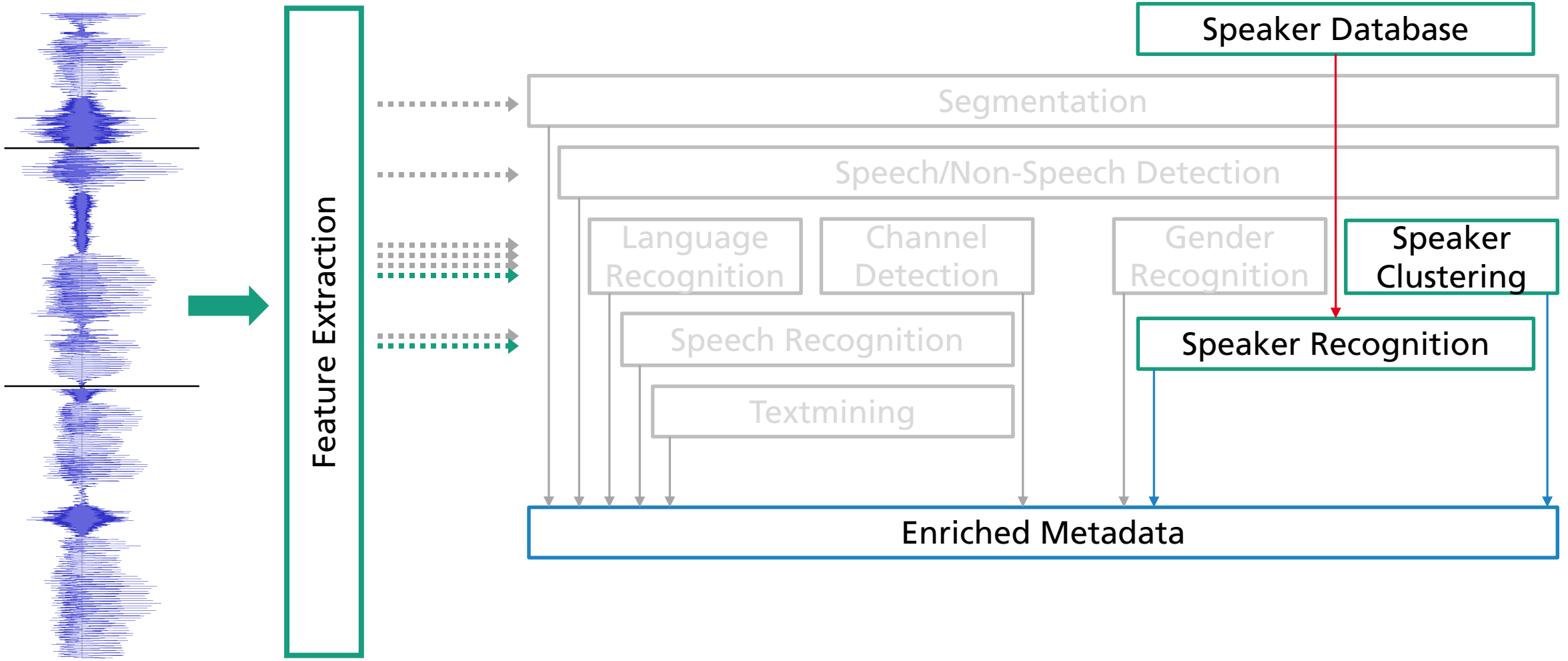
- The speaker recognition compares the speaker of a specific segment with all those speakers present in a speaker database. It enables the selection of either a very specific person or of labeling the segments speaker as „unknown“.
- The speaker information is usable across multiple media files and especially useful for search applications. It also enable the user to perform searches for audio quotes, by searching for speaker and transcript information.
- Instead of the speaker clustering results, the correct speaker can be displayed in a visualization.
- As the algorithm itself does not depend on a mapping between a speaker model and its identifier, its possible to use pure names, ids or links into different databases to identify a certain speaker. This makes it possible to store only relations to different databases to store all speaker related information.

Speaker „Tracking“

Sprecher „Tracking“

- Even without knowing, who a specific speaker is, i would like to know where i‘ve heard that person before..
- The speaker needs to be found in the database of existing material, even without having ever assigned a name to him – in opposite to normal search queries where the metadata already exists.

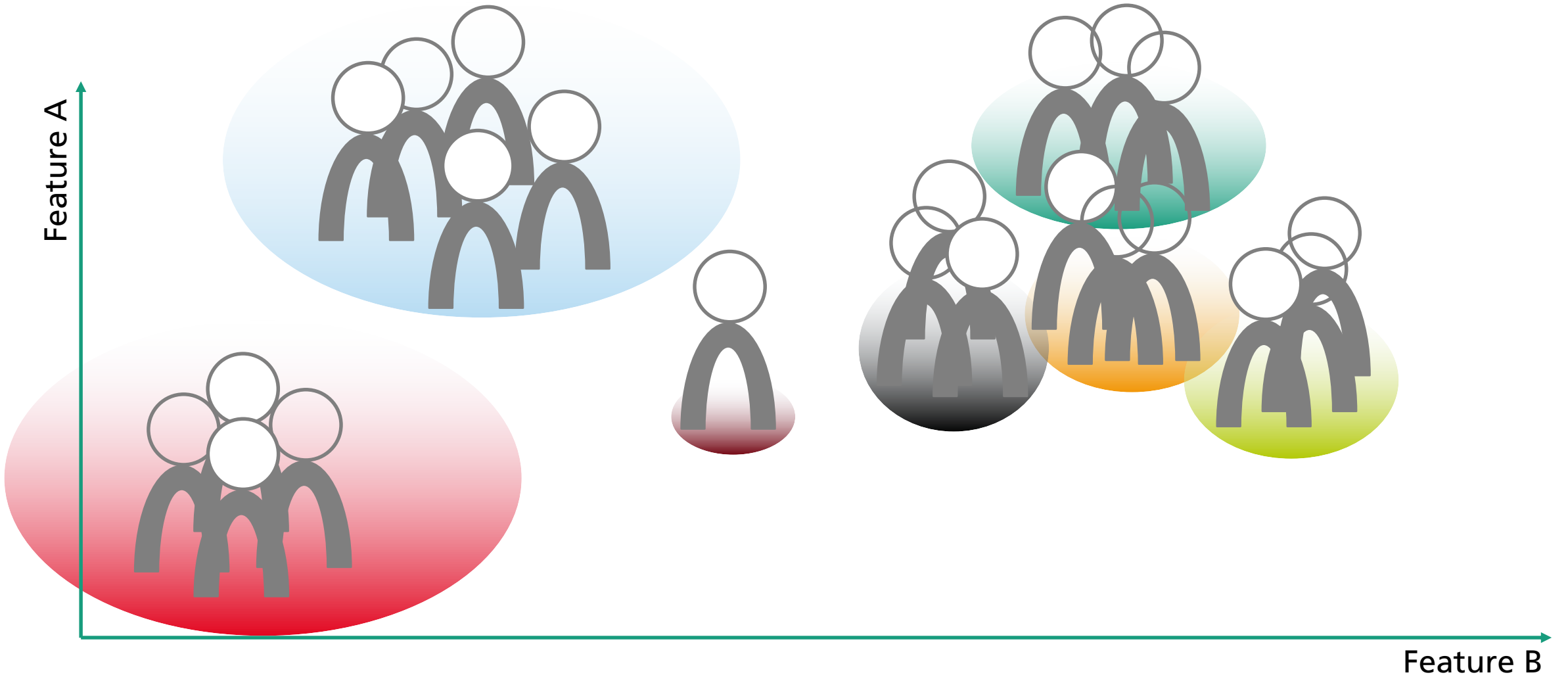
Architecture



Architecture

- Based on the segmentation and speech/non-speech results, the speaker clustering and speaker recognition can be performed.
- Speaker recognition can be separated into two steps. The generation of an audio fingerprint of a segments speaker and the comparison with reference speakers.

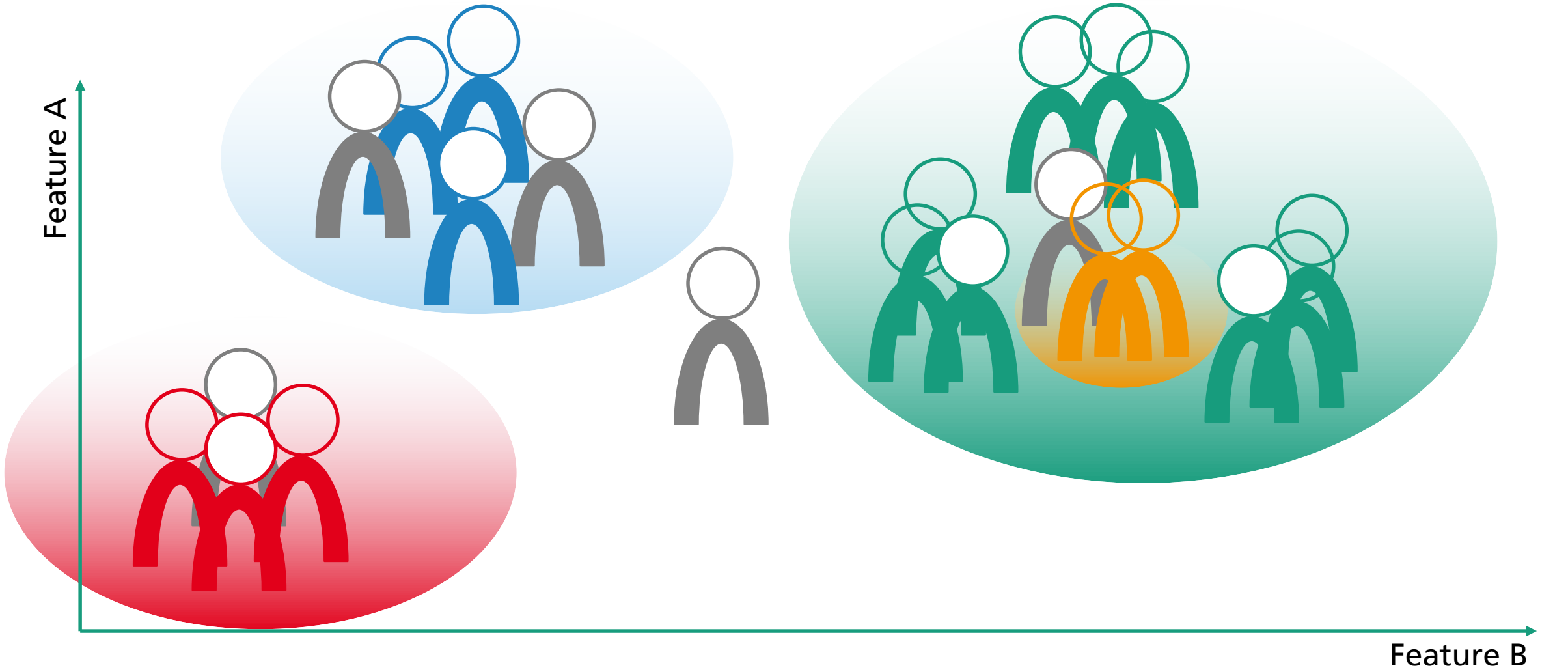
Speaker Clustering – Concept



Speaker Clustering

- The speaker clustering works only on the features of the different speaker segments, but does not have any knowledge about specific speakers and theire models.
- The graphic shows a two dimensional feature space with a possible clustering result. The present speaker segments are clustered into 7 groups.
- Colered areas highlight the areas of a certain group; in the center of these ellipses the probability of belonging to this group is the highest and decreases the higher the distance to the center is.

Speaker Recognition – Concept



Speaker Recognition – Concept

- Colered areas highlight the areas of a certain speaker; in the center of these ellipses the probability of belonging to this group is the highest and decreases the higher the distance to the center is.
- The training of the speakermodels is independent of the recognition of new speakers.
- In this example there is a big area of influence for the green speaker. During the training of this speaker data ws used, in which the variance of the speakers features was quite high. Due to that, the area of influence of the speaker green and yellow are overlapping, which will cause missclassifications.
- The unknown (grey) speaker in the middle does not fit to one of the existing speakers. To reflect this in a speaker recognition system, there need to be, in addition to models for specific speakers, a model which reflects speakers in general. This model is then used to decide between specific speakers and unknown speakers.

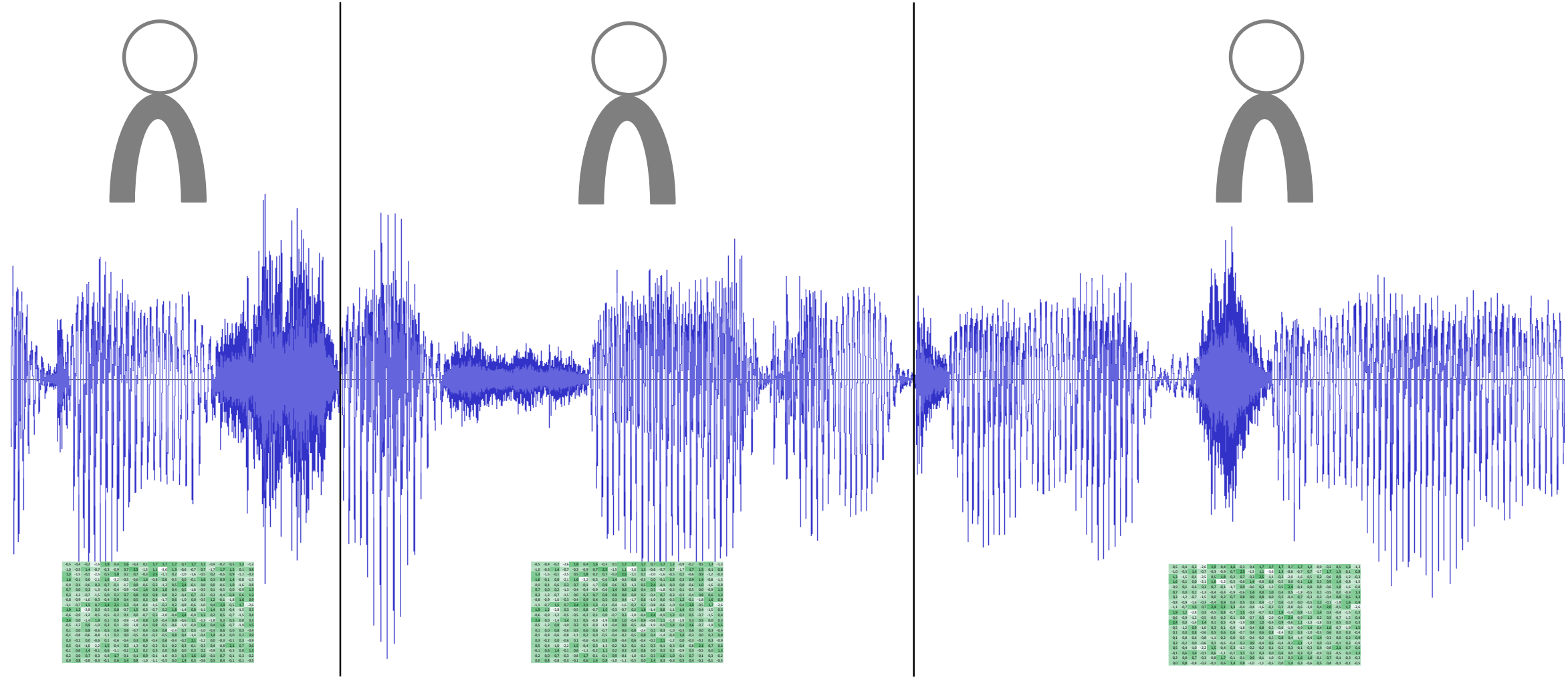
Features – iVectors

-0,5	-0,4	-0,2	-2,6	1,9	0,4	1,6	-0,3	0,1	1,7	1,7	1,7	0,7	1,7	1,2	-0,9	-0,2	0,5	1,3	-1,3
-1,0	-0,5	1,4	-0,7	-0,3	-0,9	0,7	2,5	-1,5	1,1	-3,6	1,3	-0,6	-0,7	0,7	-1,7	1,7	1,5	0,1	0,8
1,3	-1,5	-0,1	-2,5	0,5	1,8	0,2	0,7	-0,3	2,5	-1,1	0,2	-2,0	-1,6	-0,5	0,2	-0,6	0,9	-1,2	-0,3
1,6	-0,1	0,0	-2,1	1,6	-3,2	-0,5	-0,6	1,4	-0,8	0,6	-0,5	0,0	-0,1	1,6	0,3	0,9	1,4	-0,8	-1,5
-0,9	0,1	-0,6	0,3	0,7	-0,1	-1,7	0,9	-0,6	0,3	-1,3	0,5	2,4	-0,1	0,0	0,0	-0,6	1,0	-1,6	-0,8
0,7	0,0	0,2	-1,3	-0,4	-0,4	-0,9	-0,6	1,4	0,8	1,0	0,4	0,5	-1,8	-0,5	0,1	-0,5	0,0	-0,9	1,3
0,3	-1,2	-0,7	-1,5	0,0	0,2	0,7	0,8	0,9	0,8	0,0	0,2	-0,4	0,7	-0,3	-0,3	-0,4	0,8	0,6	1,3
-0,8	-0,9	-1,6	-0,3	-0,4	0,9	0,4	0,5	0,3	0,4	-1,7	0,6	-1,0	0,0	-0,5	1,2	-0,1	-1,8	1,6	0,9
-1,1	-0,7	1,5	0,7	2,4	2,1	1,3	-0,4	-0,4	-1,6	-0,2	0,2	-0,8	-0,6	-1,0	0,4	2,0	-0,5	1,7	-2,6
1,9	1,2	-2,8	0,3	-0,5	0,8	-0,7	1,5	-0,3	-0,7	0,2	1,8	-1,4	0,8	-1,1	1,4	0,3	-0,4	-1,5	0,3
-0,6	-0,9	-1,2	-0,5	-0,5	-0,2	0,1	0,0	-0,7	0,3	-2,0	-0,4	2,4	-0,9	1,2	0,2	0,5	-0,7	-1,5	0,4
2,4	0,0	-1,4	1,4	0,1	0,5	-0,9	-1,9	0,8	1,0	-0,4	0,9	-0,6	1,1	-1,3	-1,8	0,3	0,5	0,0	0,3
-0,5	-1,2	0,9	-1,0	0,3	0,1	-0,9	-1,8	-0,4	0,8	-0,5	-0,6	-1,9	-0,9	1,4	0,4	1,6	-0,7	-1,9	1,5
0,1	0,0	0,8	-0,6	0,5	0,6	0,6	-0,7	0,4	0,6	0,8	-2,4	0,2	0,3	-1,0	-0,3	0,6	0,0	0,3	-0,4
-0,1	-0,8	-0,6	-0,8	-1,1	0,2	0,0	-0,5	-0,4	-0,2	-0,5	0,8	0,4	-1,4	-0,4	1,4	-0,3	0,0	0,2	0,9
0,3	-0,2	0,0	-0,6	0,1	-0,6	-0,4	0,3	0,9	-0,4	0,6	-0,4	-0,2	2,1	-1,2	0,0	-0,3	-0,1	0,3	-0,9
0,5	-0,4	-1,0	-2,2	1,5	-0,4	0,3	-1,3	-0,2	-0,2	0,1	-0,2	0,3	-0,1	-0,3	0,4	-0,8	2,1	0,7	0,6
-0,1	0,6	1,4	-0,1	0,6	-1,1	-0,2	1,1	0,2	0,3	0,0	0,6	0,0	0,3	0,2	-0,9	0,3	-0,5	0,0	1,3
-0,2	0,0	0,7	-0,3	-0,8	1,7	-0,1	-0,1	0,9	-0,1	-1,0	-0,3	0,3	1,6	1,0	-0,1	0,7	-0,1	-0,3	-0,2
0,0	0,8	-0,8	-0,3	-0,1	0,6	1,4	0,8	-1,0	-1,1	-0,5	0,0	1,4	0,3	-0,6	0,5	0,4	-0,1	-0,1	-0,5

Features – iVectors

- iVectors abstract the speech signal by a transformation into a feature space.
- An assignment from those features back to specific language features (e.g. pitch level, ..) is not possible.
- The dimensionality of the iVector is dependent on the implementation of the iVector extraction algorithm. We are currently using a dimensionality of 400.
- iVectors have the same amount of dimensions, even for different lengths of segments. Therefore the iVectors enable us to do a comparison of speakers of different segments.

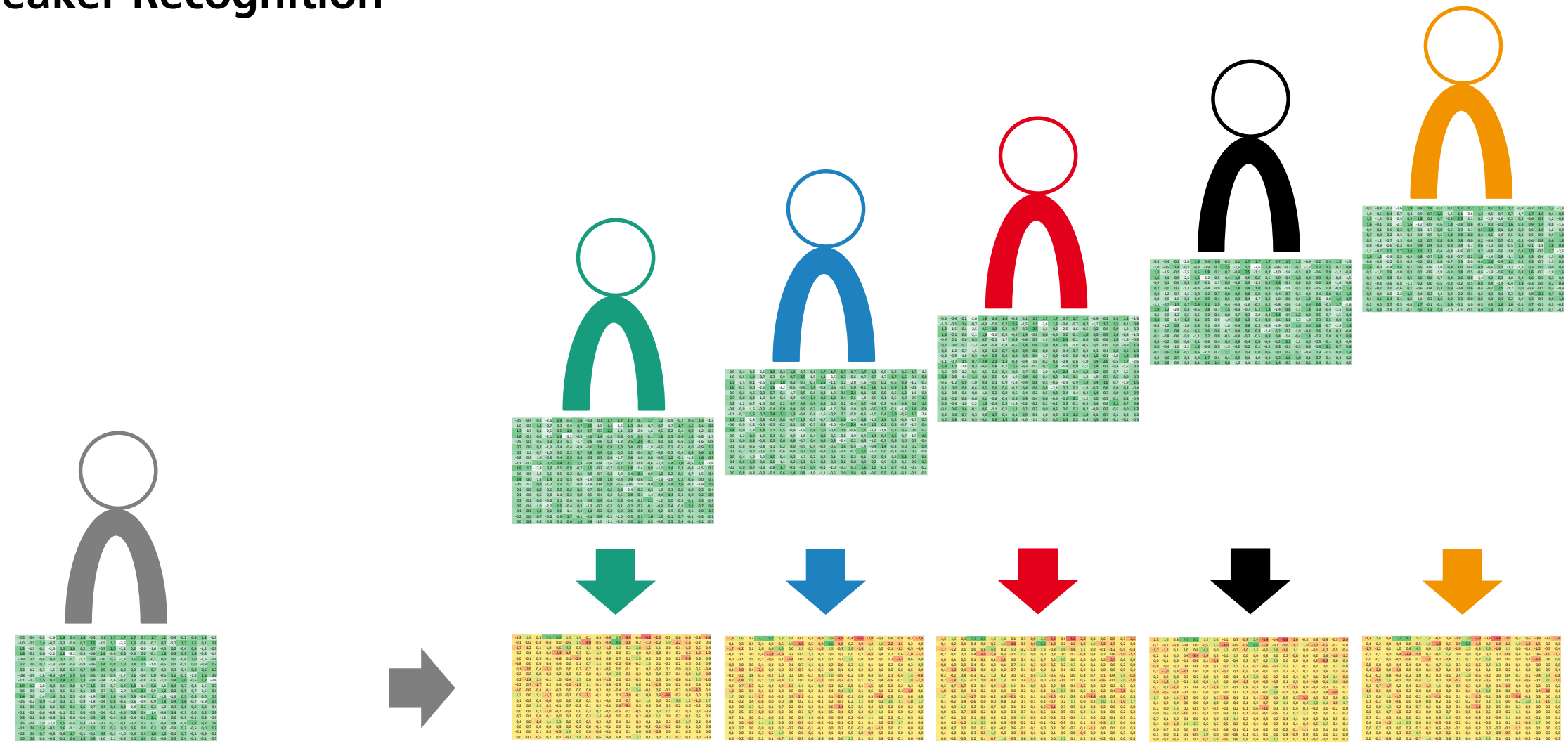
Features – Generation/Extraction



Features – Generation/Extraction

- Based on the segmentation and speech/non-speech detection results an iVector for all speech segments can be created.
- The dimensionality of the iVectors for varying length of segments are constant. This makes an easy comparison possible.
- The creation of the iVector is not yet a speaker recognition, but can be seen as the fingerprint creation step. The creation of the iVector is quite computation intensive; however this creation is required only once, even if the database of known speakers is updated. The comparison of already created iVectors with reference iVectors – which is done during the actual recognition – is quite fast.

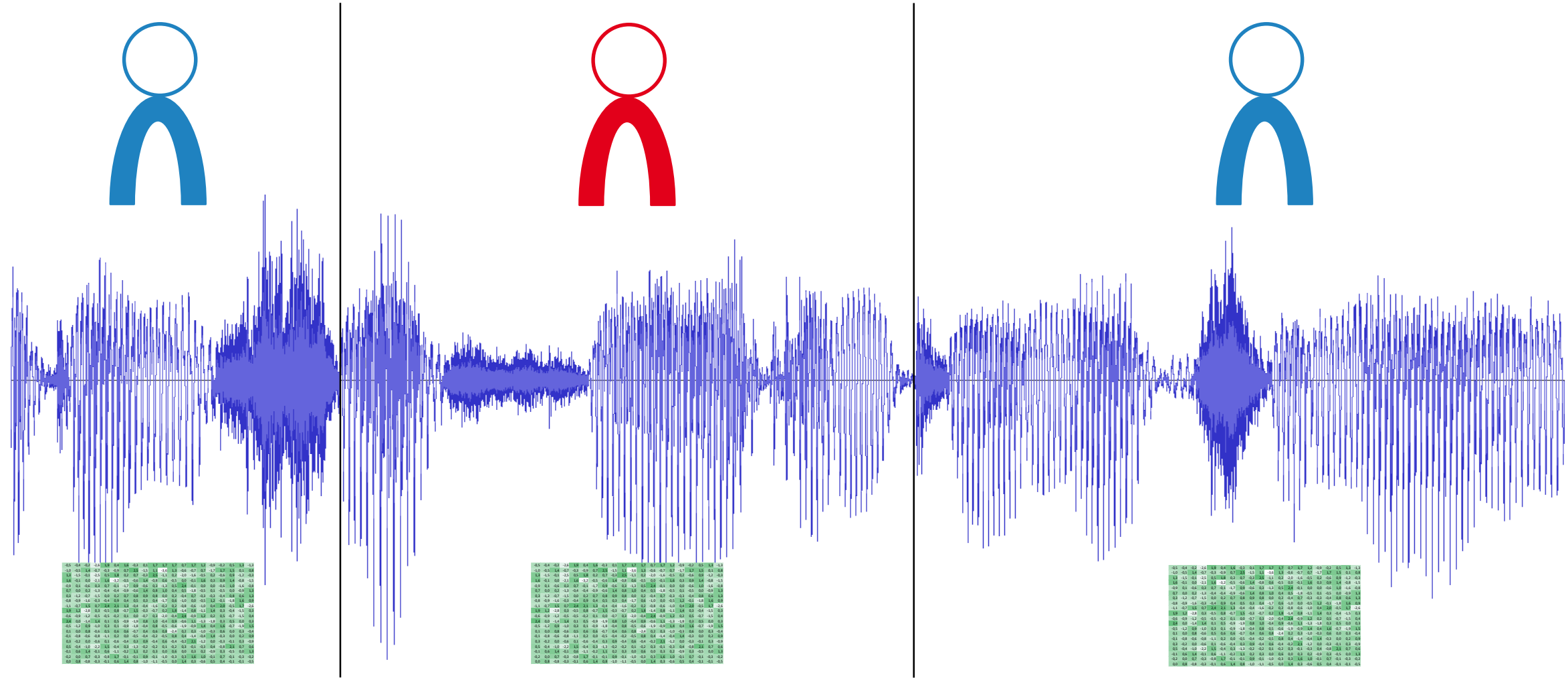
Speaker Recognition



Speaker Recognition

- The iVector of a segment is compared with all known speaker in a database. The comparison is a scalar multiplication (cosine distance) and thus a very fast operation for the computer.
- This comparison is performed for all segments to assign a speaker to each of those segments.

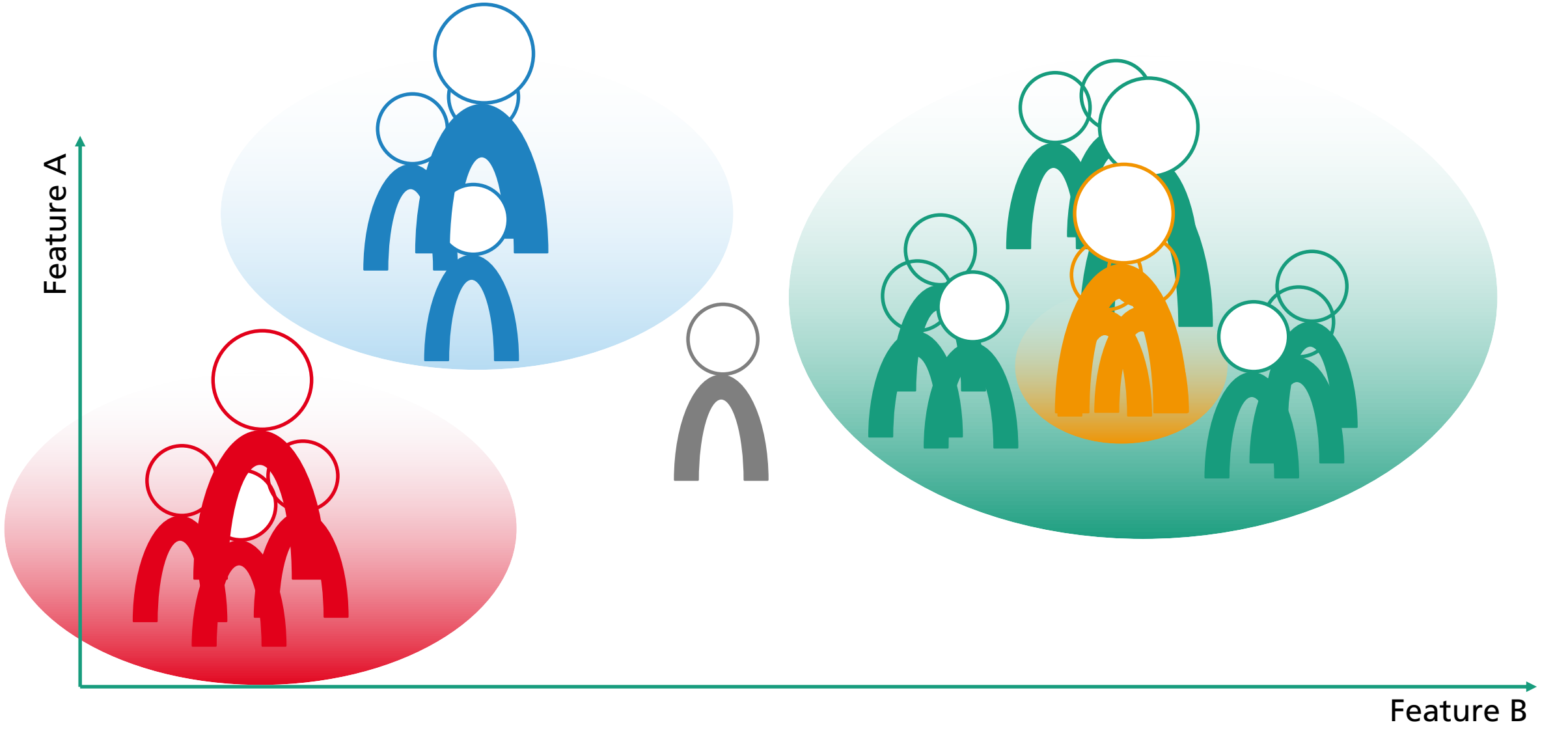
Speaker Recognition



Speaker Recognition

- In our example the 3 segments get assigned different speakers. Segment 1 and 3 had – see the example from the structural analysis and concept detection session – the same speaker John Doe. The 2nd segments speaker is Jane Doe.
- In the example you see a colored representation of the speakers. Instead of colors names, identifier or links to database entries could have been used. The algorithm does not know either of them; the mapping between iVector and the person is part of the software system around the algorithm.

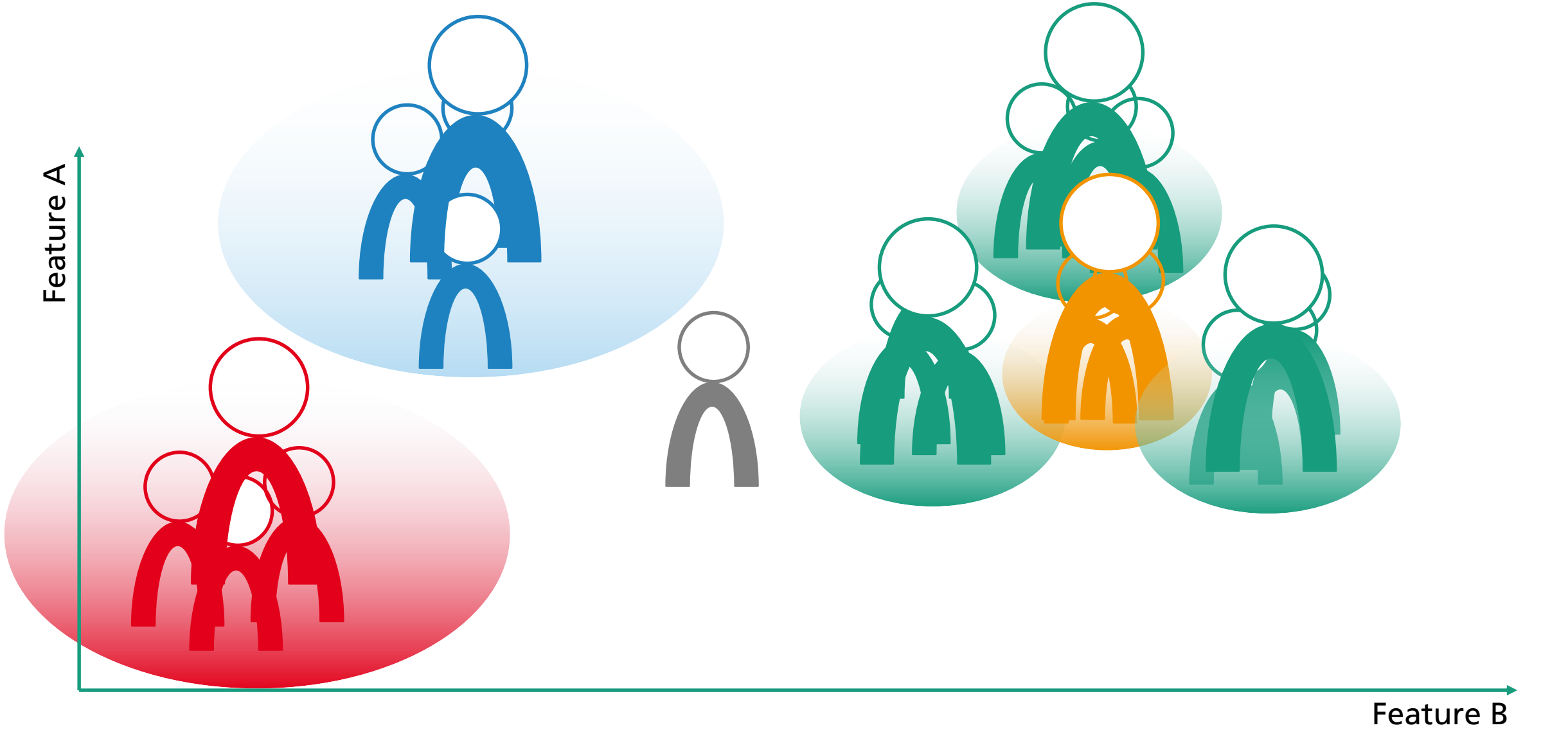
Training of Speaker Models



Training of Speaker Models

- The training of speaker models requires annotated data. The training creates reference iVectors, which are representatitve for the speakers. Each speaker will usually be represented by a single iVector. In the graphic it is displayed as a large figure. The small figures denote the training data which was used in the creation.
- The training itself is a weighted mean of the annotated iVectors of a speaker and thus can be computed very fast. Therefore it is possible to train a speaker iteratively and gathering more and more training data over time for speakers.

Training of Speaker Models



Training of Speaker Models

- This example shows a possible speaker training in which speaker green is represented by multiple reference iVectors. This can be advantageous if the training data of a speaker varies to much and would create conflicts with other speakers. This can happen due to changes of a speakers voice over time, due to different recording situations (studio, telephone) ..

Who trains new speaker models?

Who trains new speaker models? You!

- The creation of new speaker models can be done independently of already existing models and each speaker can be represented by multiple reference fingerprints.
- The creation should use approximately 5 minutes of speech of the trained speaker.
  - Its required to use at least a single iVector to label a speaker, though it will not be robust to naturally occuring variations of the voice.
  - Ideally audio material from different audio situations is used to create robust models.
- The creation of new speakers can be performed by (expert) users. It is not required to adopt the fingerprint creation.
- The training is done by calculating the weighted mean of the training iVectors of a speaker.

Open Source Software

Open Source Software

- Alice
  - Developed and maintained by the university Avignon (since 2014)
  - Source code in C++, available under LGPL Licence
- Bob.spear
  - Developed by IDIAP, Switzerland (2014)
  - Source code in Python, available under LGPL Licence
- Kaldi
  - Included iVector support for speech recognition
  - Source code in C++, available under Apache Licence

Who am I?

- **David Laqua**
- Research Engineer
- Telefon: +49 2241 / 14 2725
- Email: david.laqua@iais.fraunhofer.de

Disclaimer

Responsible contact: David Laqua
E-mail: david.laqua@iais.fraunhofer.de