

Thinkie for Thinking-Aloud

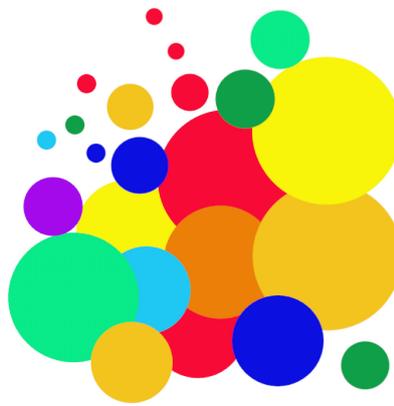
Brigitte Endres-Niggemeyer

18.09.2021

06:49 Sonntag 14. März

📶 97% 🔋

Thinking-aloud for field studies on mobiles



Some background knowledge on Thinking Aloud

Moran , Kate (2019): Usability Testing 101.

Nielsen Norman Group <https://www.nngroup.com/articles/usability-testing-101/>

Netto, Ivan (2019): Usability tests in a nutshell.

UXCollective <https://uxdesign.cc/usability-tests-in-a-nutshell-1baa6744be10>

Wikipedia: Think aloud protocol. https://en.wikipedia.org/wiki/Think_aloud_protocol

Thinking aloud

The Thinking-aloud method lets you observe what a person is thinking when dealing with some issue. The key instruction is to have her talk while performing some task, and to record her talk. The resulting concurrent protocol contains thoughts that are lost later because forgotten or overlain by newer content. Thinking-aloud gets you a record of the person's short term memory content.

Using the verbal protocols of thinking aloud may show for instance which problems a user has with a new technical device or user interface. If these problems are caused by technical shortcomings or poor design, you get to know them and can improve your product. This is a main use of thinking aloud in usability engineering.

It is easy to imagine other applications of thinking aloud:

What goes on in the head of a person interpreting a sculpture?

How does a tourist go about when making her way through a foreign town, using a map, a smartphone, asking people and looking at other information sources around?

How does a plumber find out from where comes the water oozing out of a wall?

For obtaining valid data of thinking aloud you often have to be on the spot. The subject you observe needs the normal surroundings and affordances, otherwise you get biased results. This is a core condition of field research. For being on the spot you need equipment that is easy to transport, to handle and works anywhere. Both the test subject and the researcher should get along with it. For this situation Thinkie offers an iPhone and an iPad that store their data on iCloud.

The test subject will often refer to things nearby in the environment. If you make a movie of the situation this will ease the interpretation of the verbal protocol. So a thinking-aloud tool should handle videos.

The verbal protocol normally needs to be written down (partially) in order to exploit it. You will often want to discuss it with the test subject. This is why the mobile research outfit should support in particular the transcription of the verbal protocol.

All data files resulting from the study may be needed elsewhere. So the thinking-aloud tool should be able to export them.

Thinkie for data capturing

Let us assume that one has agreed on some data capture event for thinking aloud. The aim of investigation is clear and the test subject knows her stuff. The investigation environment is set up with all required items such as instruments and documents. The researcher is to bring the rest — the iPhone and the iPad for the data capture.

The researcher has to be fully prepared. Thinkie is implemented on her iPhone and iPad. The Internet access works ok, the iCloud access is on. If so, the data can go to the private iCloud container of the researcher, audio files to the sector „phone“, videos to the sector „pad“, every file getting a record there.

The test subject records her thinking-aloud talking on the iPhone. In parallel the researcher may film the scene on the iPad. The audio file on the iPhone and the video file on the iPad are kept together by giving them suitable names. When the record is finished, one may listen to the audio fully or in part before uploading it to the cloud. All the same with the video: check and upload. Just in case something went wrong you would try again if possible.

After all that the researcher can begin with data reworking. She may consider the movie, but her main point is the speech data. Transcribing it to a written verbal protocol is her first task. Without that, interpretation is difficult. For doing that one has to pick time intervals from the audio file, which can either be submitted to speech recognition or be transcribed manually — most of the time by the researcher herself. Both ways lead to the same Thinkie transcript.

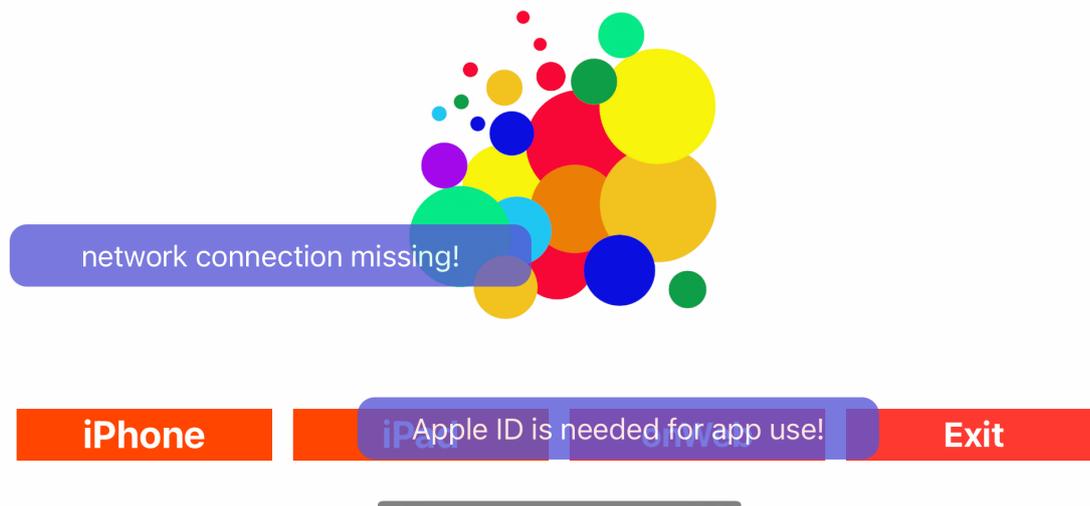
For manual transcription you can produce a list of audio intervals of suitable length. Thinkie offers them on a table so that you can go through. The size of the intervals should conform to the takes you can absorb at a time.

The speech recognition expects the language that is set as the basic language of your device. You change it as required, for instance from English to German or Italian. Then you go one by one. You grip an item that makes sense and is short enough for recognition (currently < 60 seconds). You send it to the recognition. After some wait the result arrives. It is attached to the existing transcript. You correct it as needed and store it locally and in the iCloud data record.

Frequently the test person and the researcher will discuss their shared recording. So it is helpful to have suitably reworked data for this purpose as fast as possible.

Thinkie: Common iPhone and iPad start

Thinking-aloud for field studies on mobiles



Start on iPhone or iPad: WiFi / WLAN iCloud and AppleID are required

The first Thinkie screen checks whether Thinkie's working conditions are fulfilled. Then it directs to the right device: iPhone or iPad.

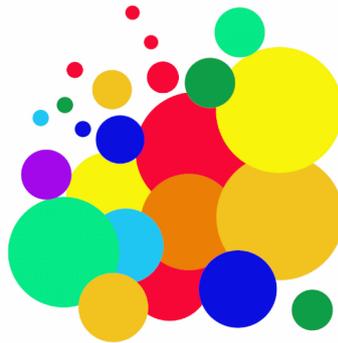
If there is no internet connection an error message states the problem, often caused by the WiFi put off or the flight mode set on. iCloud access is required, too. It can be set on in the settings inside your AppleID. Next an explicit AppleID authorization for the Thinkie app is asked for. The page is localized in your system environment, for instance using face recognition or asking for the passcode of the device. The image below shows the passcode version on an iPad in German. If the authorization goes wrong you return to IOS.

„onWeb“ works all the time. It leads to the Thinkie page of my web site. There you find background information and a Thinkie user instruction. The „Exit“ button returns you to the IOS.

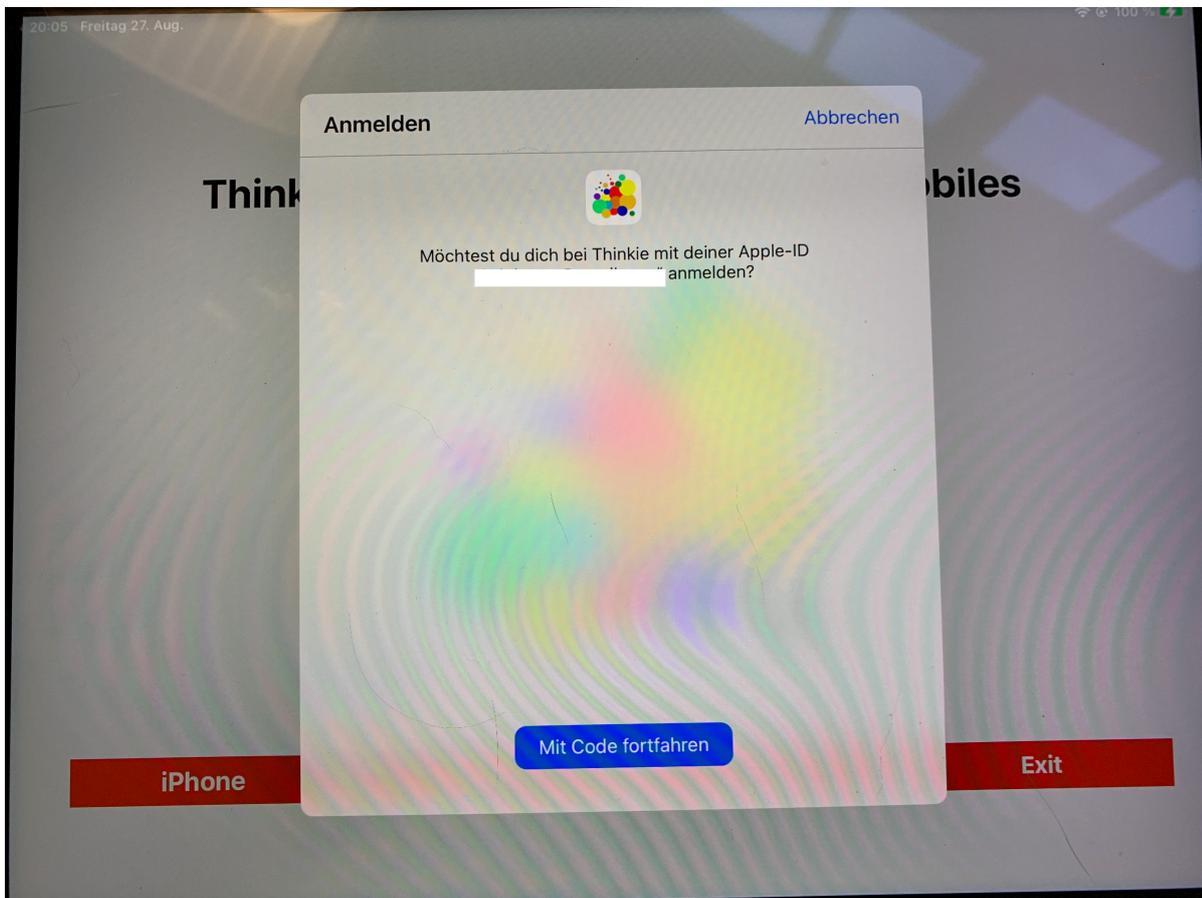
If the right device button does not work in spite of authorization it helps to click „onWeb“ and return from there.

iPad and iPhone run independently. They deliver their recording to the private iCloud container of the owner. The iPhone writes into the sector „phone“, the iPad into the sector „pad“. The data record format is all the same for both of them.

Thinking-aloud for field studies on mobiles

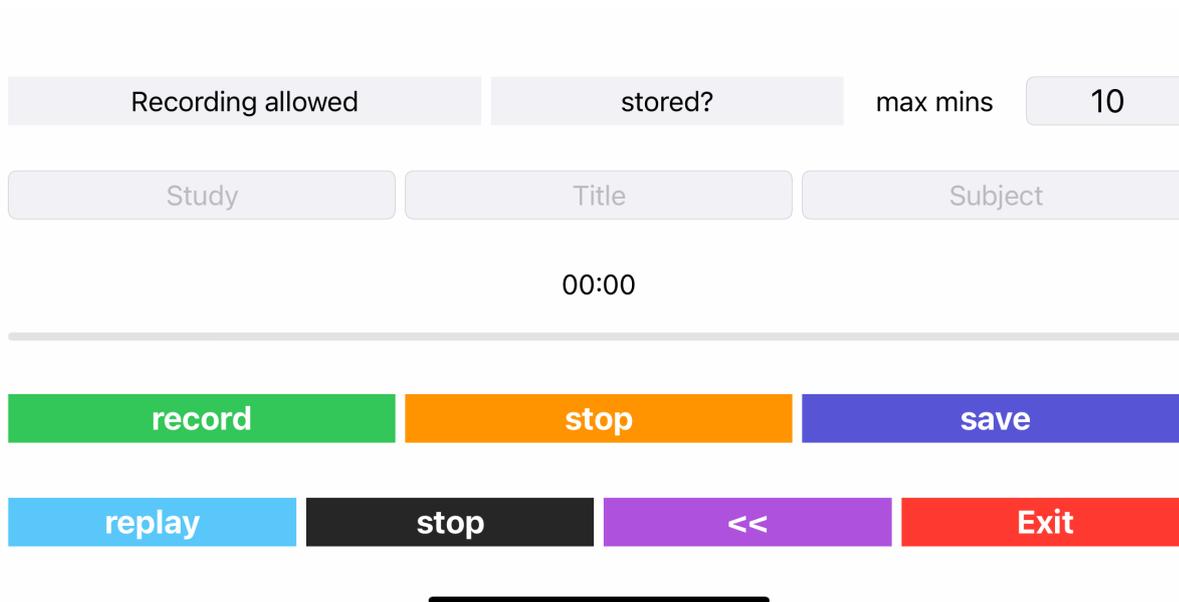


Standard entry situation: The AppleID will be asked for



Localized AppleID authorization with device code on iPad

On the iPhone



Screen for audio recording in initial state

On the phone the sound is the point. The test subject is reporting on her current mental activity. This is recorded in any language with accompanying noises. The speech recognition for Speech-To-Text will later expect the basic system language. In the example below you choose between German, English, French and Italian. Other languages may be preset as well.

The audio file needs a three-item name. They are categorized as study, title, and subject. But you put in the labels you want.

You can adapt the expected duration of the audio. This trims the slider. The slider stops when reaching its end, but the recording goes on.

When the recording is done, one can listen to it, wholly or in part. After that the file should be saved in the cloud container. If this does not succeed, an error message appears. Otherwise the upload is confirmed : „stored!“.

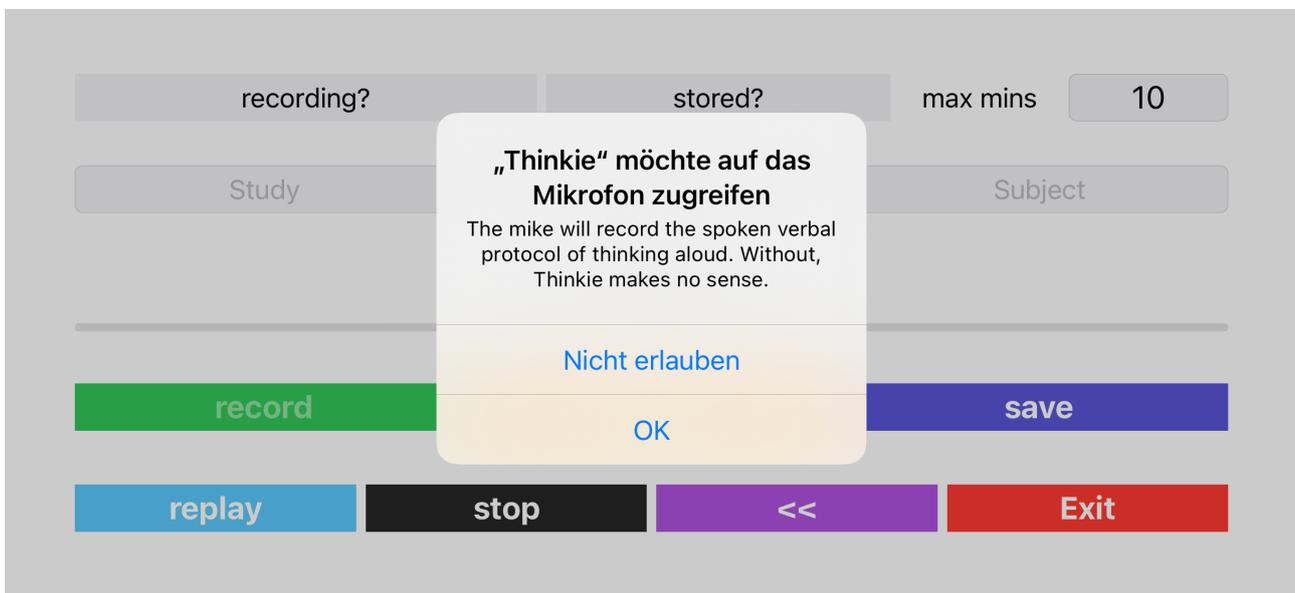
The audio file will show up on the audio table of the iPad.



Keyboard for text input with choice of language blended in

The keyboard appears when a text field is clicked at. A click into the background moves it away.

The slider time is preset to 10 minutes. The value can be changed. The keyboard adapts to numeric values.

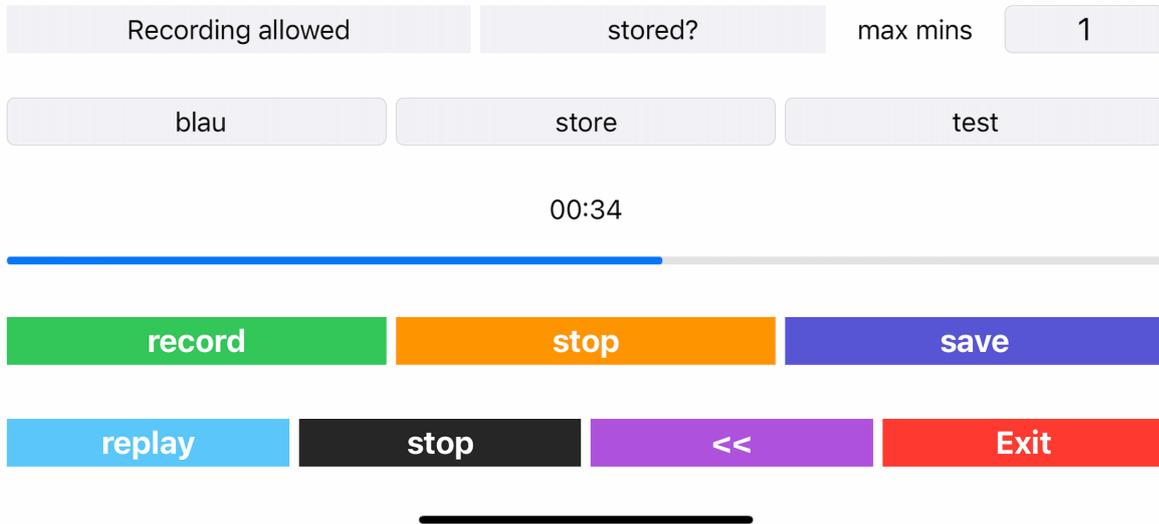


Localized microphone use authorization needed

The owner of the phone is entitled to record audio data. The „allowed“ statement appears when pressing the record button. Clicking „stop“ finishes the recording. Only the very first time the iPhone owner has to authorize microphone use.

With „save“ you send your audio file to the cloud container.

The local version is temporary. It will be overwritten with the next file.



Recording is on, nothing is stored



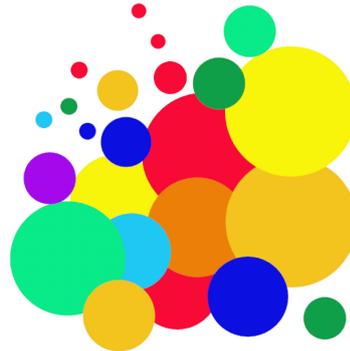
Audio saved in cloud record

On the iPad

07:37 Montag 13. Sept.

100% 🔋

Thinking-aloud for field studies on mobiles



Use audio and video

Record video

toStart

Exit

Distribution interface of the iPad

On the iPad users first see a distribution screen.

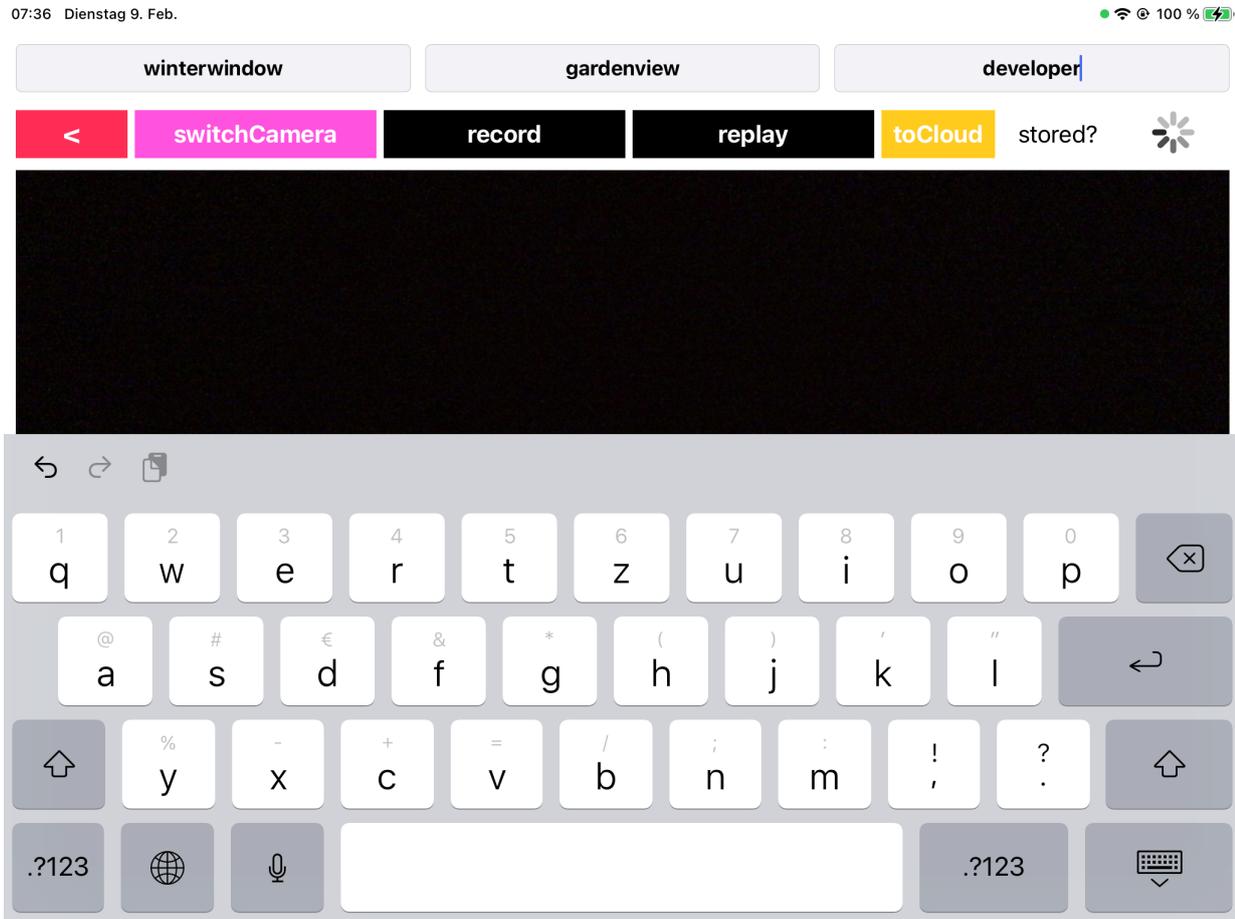
It helps to get out (“Exit“) and to return to the initial interface of the app in order to go to the web, for instance to look into the user instruction („toStart“).

On the iPad you are invited to either use an existing audio or video file („Use audio and Video“) or to record a new video („Record video“).

In the first case the button leads to an overview table of the audio and video files stored in your private iCloud container. There you choose the file of your interest.

The „Record video“ button directs straight to the page for video recording.

Video recording



Video recording page with keyboard overlay

As audio files video files need a three-label name. For entering the text, the keyboard comes up and covers the video preview. Tapping the button down on the right makes the keyboard disappear.

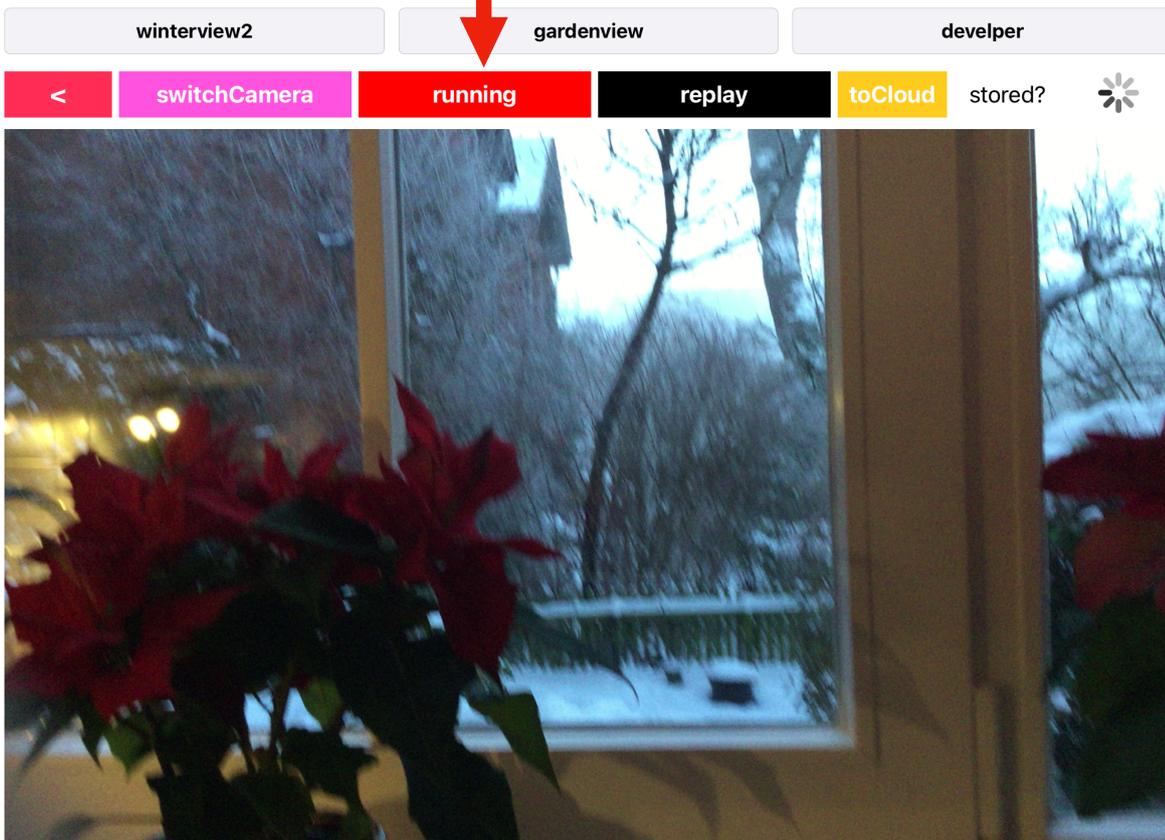
Before recording the camera can be switched to the front or rear side. The „record“ button starts the video recording, changing its color to red and displaying its current state „running“. You stop recording by clicking the button again. It will return to black and display the „record“ state.

You can check your movie using the „replay“ button.

„toCloud“ sends the video file into the cloud record. Upload and getting the arrival confirmation may take some time. You may have to ask more than once.

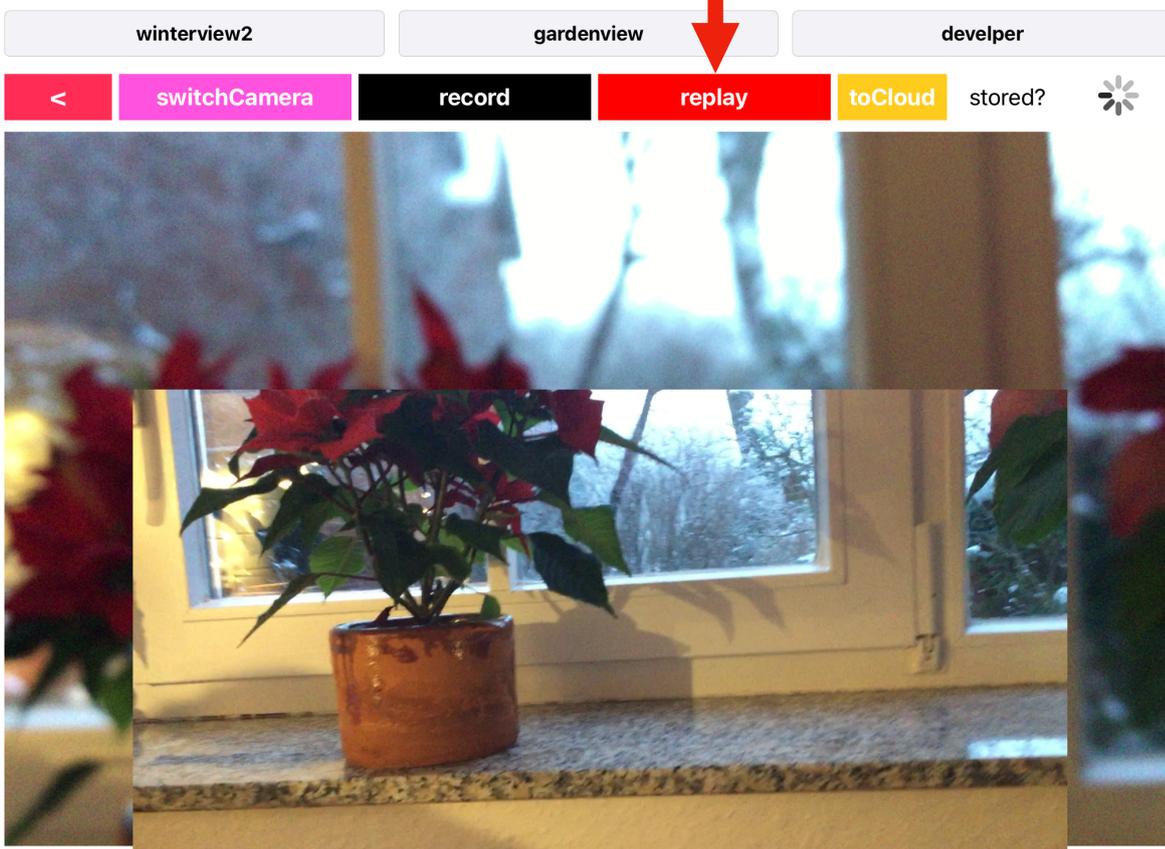
07:39 Samstag 13. Feb.

100 %



07:40 Samstag 13. Feb.

100 %



Video recording (above) and replay (below)

08:28 Mittwoch 17. Feb.

100%

uxmake error upload

< switchCamera record replay toCloud failure!

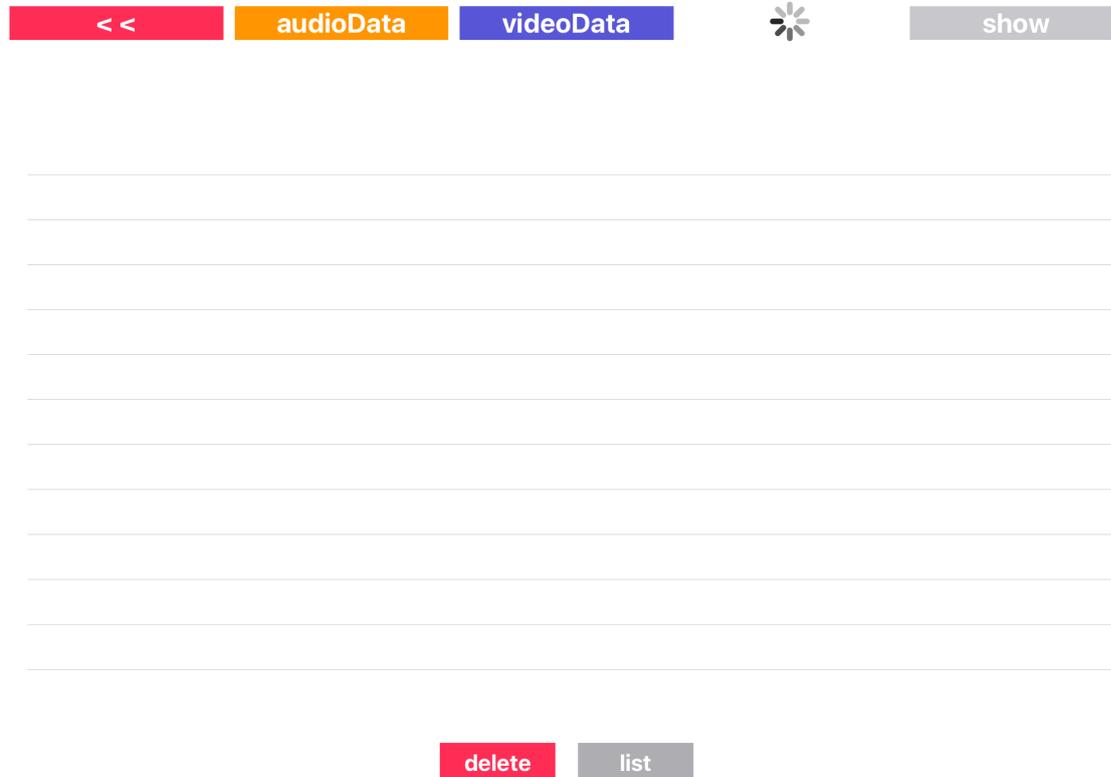


A test video is still loading up to the cloud record

Overview of cloud files

07:41 Dienstag 9. Feb.

📶 @ 100% 🔋



Empty table of audio and video files

The table above is used for metadata of files: it displays up to 24 names of audio or video files from the cloud container.

To populate it you choose „audioData“ or „videoData“ and press „show“.

The newest files are presented on the top. The button at the right links to the file itself.

The button „delete“ puts the table into delete mode, the button „list“ returns to the standard.

You cannot delete the first file.

<<
audioData
videoData
show

winterwindow + gardenview + developer	video
- dabei + test + audio	video
- schnee + desktop + mollie	video
- weiss + gruen + rot	video
- ichwaer + einhuhn + ichwollt	video
b + rot + blau	video Löschen
- Szene + Pauli + Mollie	video
- myhome + xmasstate + kitchen	video
- kitchen + International + Cat	video
- Video + montalbano + me	video
- Video + lang + Garten	video
- Video + Test + molliifenster	video

delete
list

Video table in delete mode (above), regular audio list (below)

<<
audioData
videoData
show

neuer + Ton + fragen	audio
irrtuemer + wert + kaestner	audio
cucina + mia + developer	audio
catprofile + sculptureview + Mollie	audio
schnee + draussen + jetzt	audio
schnee + mollie + desktop	audio
kinder + brauchen + maerchen	audio
kitchen + english + me	audio
cucina + internationale + io	audio
kitchen + myhome + xmasstate	audio
kitchen + International + Cat	audio
Video + montalbano + me	audio

Video playback, export and deletion

Videos must be downloaded from the cloud container for handling them. The „getVideo“ button fetches the video of your choice. Video files may be big, so that the download can take time. Messages may appear on the empty interface, telling you to wait and to tap „getVideo“ again.

As soon as the video file has arrived, the video handling buttons are enabled. Now the file can be played and exported.

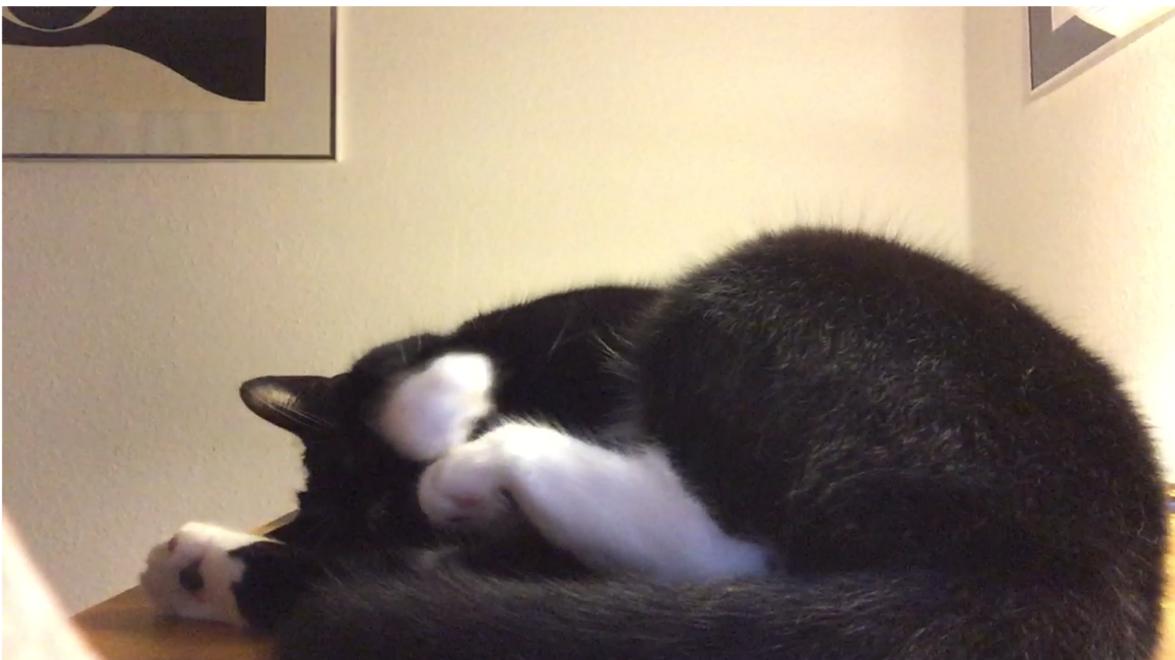
On the export overlay below you see a range of external addresses in the owner's system environment. You click on what fits to export the video there.

The exported video keeps the nobody name „videoname“ of the local copy from where it is fetched. You better rename the exported file.

Using the „deleteRecord“ button removes the video record from the cloud container. A short „done“ message confirms that the deletion was performed.

23:34 Donnerstag 16. Sept.

📶 100 % 🔋



getVideo

deleteRecord

play

exportVideo

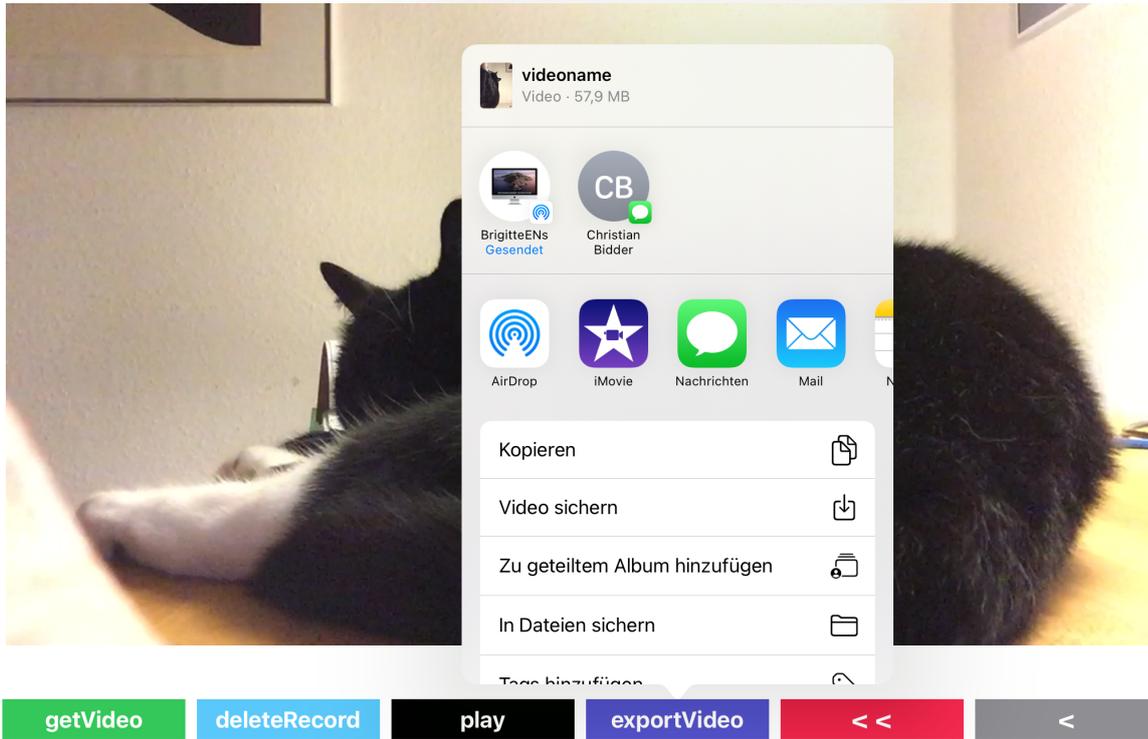
<<

<

Video ready for playing, deletion and export

07:02 Freitag 17. Sept.

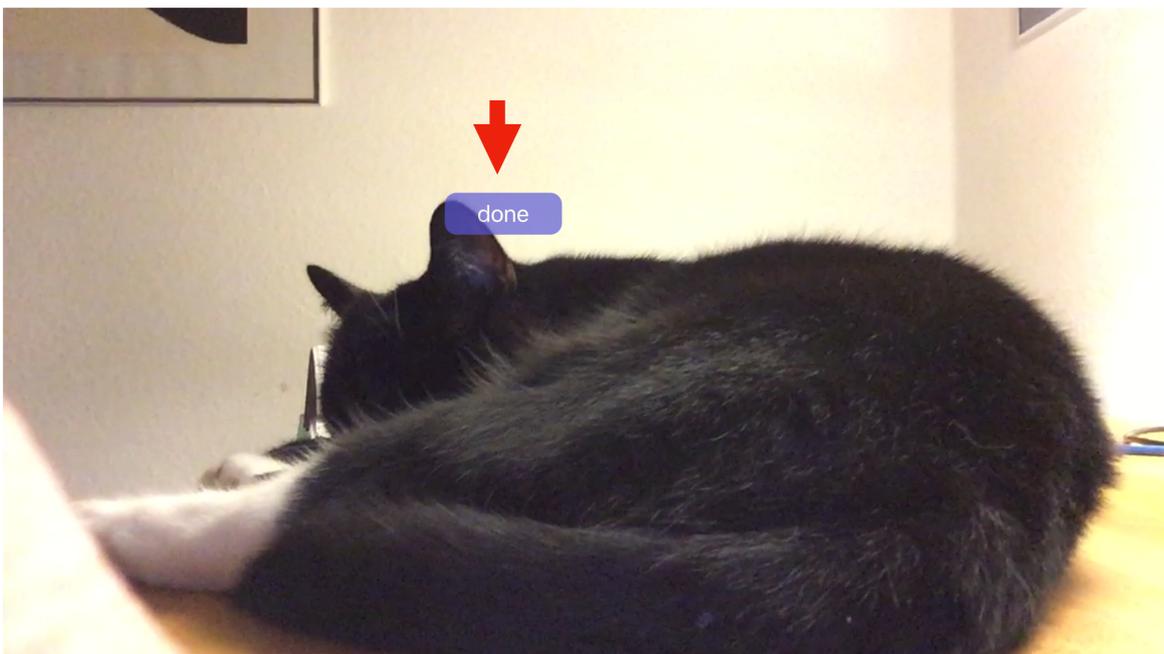
100 %



Video with export overlay, export is already done

21:06 Montag 20. Sept.

100 %

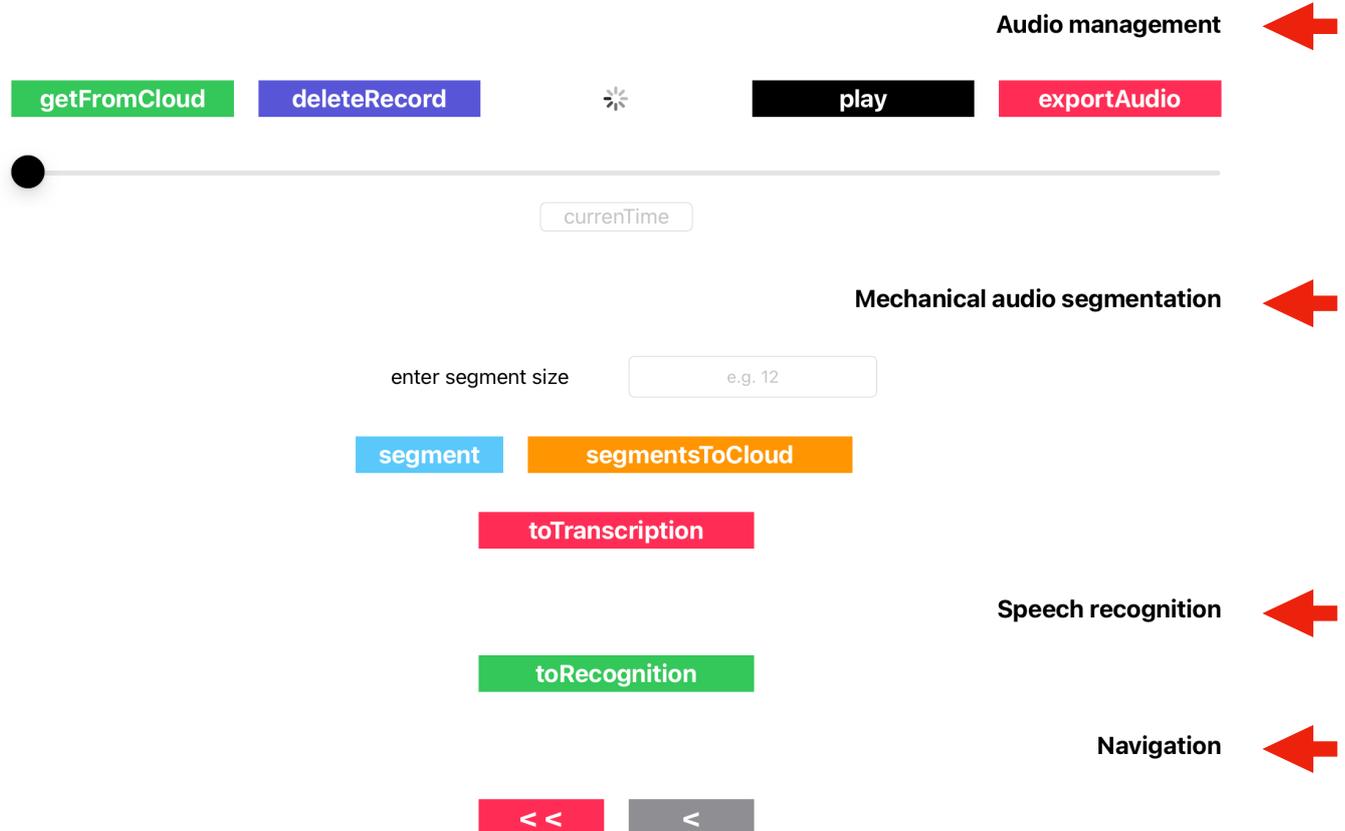


Video removed

Language processing start page

11:53 Donnerstag 8. Apr.

📶 100 % 🔋



Sound processing: playback, segment, transcribe and recognize (Speech-to-Text)

The first audio handling task is to fetch the audio file from the cloud container. As soon as it arrives, the disabled buttons are activated. „play“ will play the audio file.

„exportAudio“ will export it. The export display works like the one shown above for videos. „deleteRecord“ will remove the audio file from the cloud container. As shown above for videos, it confirms with a „done“ message that it succeeded.

In the „Mechanical Audio Segmentation“ area you define intervals (clips) of the audio file that you want to transcribe at a time. You set their duration and tap the „segment“ button. During segmentation the audio file is replayed. The finished clips go to the cloud record with the „SegmentsToCloud“ button.

With „toTranscription“ you switch to the transcription page.

All speech recognition happens on the recognition page. You reach it tapping the „toRecognition“ button. A demo below shows how to go about there.

The navigation links return you to the data file overview and to the app start screen.

Audio management

getFromCloud deleteRecord * pause exportAudio

00:45

Mechanical audio segmentation

enter segment size

segment segmentsToCloud stored!

toTranscription

Speech recognition

toRecognition

Navigation

<< <

Later state: audio file segmented, audio clips stored in the cloud record

Transcription

09:18 Freitag 9. Apr.

📶 100 % 🔋

Aha

clip 00:	00:00	00:11	play	getFromCloud
clip 01:	00:12	00:23	play	saveText
clip 02:	00:24	00:35	play	TextToCloud
clip 03:	00:36	00:45	play	

exportText

playAudio

<

<<



Transcription page: Data downloaded, four audio clips available, transcript empty

On the transcription page, the „getFromCloud“ button fetches the audio file, the audio clips, and the transcript. In the present case four clips are listed.

The transcript is empty. The „play“ and „playAudio“ buttons are enabled. „playAudio“ plays the whole file, unless you halt it setting it to „pause“ state.

The slider is in its start position. It follows „playAudio“ of the whole audio file. „play“ enters its position in the still empty field under the red arrow. The slider does not react to it.

You enter the text into the transcript field.

The buttons below return you to the file overview and to the iPad start screen from where you can leave the app.

09:20 Freitag 9. Apr.

📶 100 % 🔋

Èccoci in cucina mia. Si va dalla sinistra alla destra. Si vede il gatto, poi il mio posto di lavoro, un tavolo nero d'Austria

clip 00:	00:00	00:11	play	getFromCloud
clip 01:	00:12	00:23	play	saveText
clip 02:	00:24	00:35	play	TextToCloud
clip 03:	00:36	00:45	play	

exportText
playAudio
<
<<



00:24

Transcript entered till clip 2, nothing has been saved so far

On the screen above, the data from the cloud record has been fetched. The slider is in its zero position.

The clips 0 and 1 have been played and typed into the transcript. Clip 2 + 3 are on. As displayed (see red arrow), clip 2 begins at second 24 of the audio file.

So far, the transcript was neither stored locally nor saved in the cloud record: nothing in the white field in the button column. „saveText“ will save the transcript locally, with „TextToCloud“ you write it into the cloud record. The state of saving is displayed in the white field.

On the screenshot below, all clips have been played. Their content has been written into the transcript. The slider is in its end position, what shows that the whole file has been played, possibly for checking the breaks between the clips. The storage field confirms that the transcript has been saved, but not yet in the cloud record.

The transcript can be exported in any state.

Èccoci in cucina mia. Si va dalla sinistra alla destra. Si vede il gatto, poi il mio posto di lavoro, un tavolo nero d'Austria

clip 00:	00:00	00:11	play	getFromCloud
clip 01:	00:12	00:23	play	saveText
clip 02:	00:24	00:35	play	TextToCloud
clip 03:	00:36	00:45	play	not stored to cloud
				exportText
				playAudio
				<
				<<

00:24

Transcript stored on the device (above), in the cloud record (below)

Èccoci in cucina mia. Si va dalla sinistra alla destra. Si vede il gatto, poi il mio posto di lavoro, un tavolo nero d'Austria ed il frigorifero rosa italiano. E adesso torniamo per la mia cucina internazionale da Mollie, il gatto.

clip 00:	00:00	00:11	play	getFromCloud
clip 01:	00:12	00:23	play	saveText
clip 02:	00:24	00:35	play	TextToCloud
clip 03:	00:36	00:45	play	stored!
				exportText
				playAudio
				<
				<<

00:45

Speech-To-Text: Speech recognition

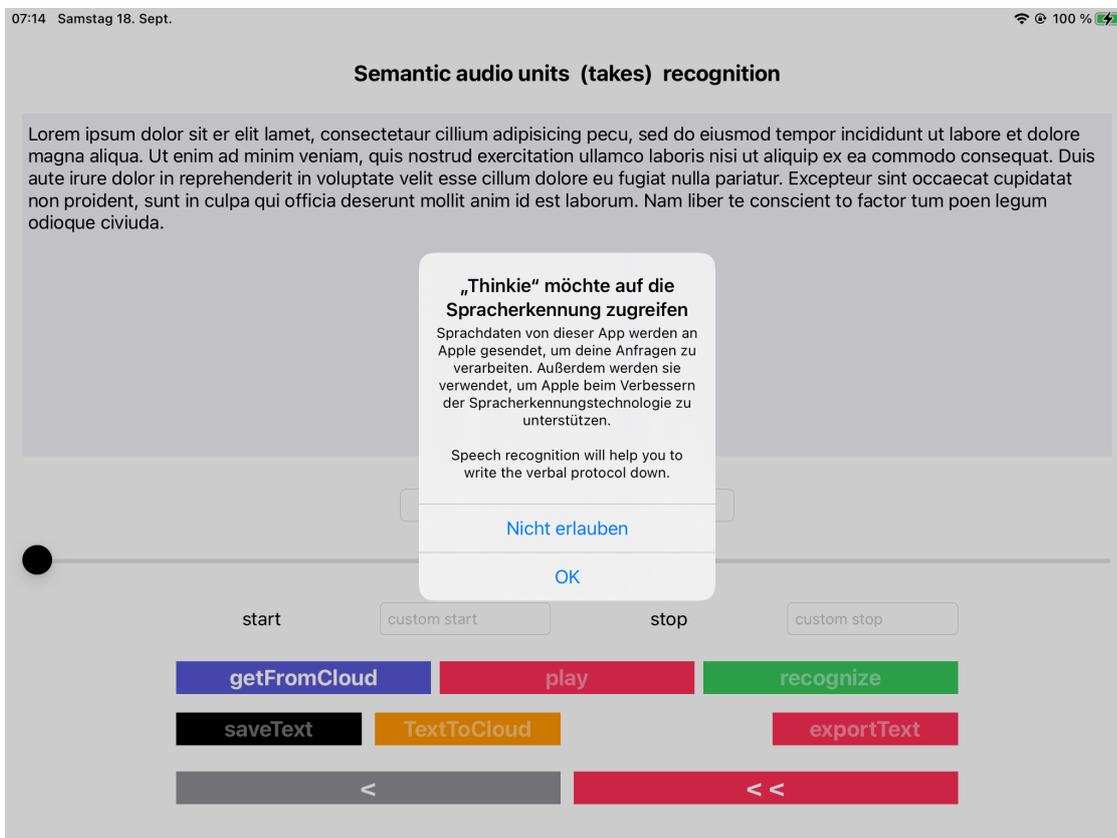
Speech-To-Text - speech recognition - needs spoken sequences which are short enough to be transcribed at a time (current max around 60 seconds), and that correspond to sense units. Sequences that make sense are recognized better.

The speech recognition offered by Thinkie is preset to the system language of your device. You adapt your system language to the language you want to be recognized. You have to agree to the use of speech recognition.

Thinkie copies a sequence from the audio file and sends it to the recognizer. When the result arrives, it is appended to the transcript. It can be corrected there.

You grip the sequences to be submitted one by one from the audio file. You start the playback with „play“ and finish the segment with „pause“. The values appear in the start and stop field. Tapping „recognize“ you send the selected sequence to the recognizer. Thinkie uses the server-based recognizer.

As soon as the recognition result appeared on the transcript, you can go on. You restart the audio. The old stop value moves to the start field. When you tap „pause“, you set the new end value. Your new sequence is ready to be forwarded to the recognizer.



Agree to speech recognition (localized)

Practically speaking, for a recognition action:

1. you press „play“
2. you press „stop“ - the stop value appears in its field

Then for every recognition unit:

3. you press „play“ again - the old stop value moves to the start field
4. you press „stop“ - a new stop value appears in the stop field
5. you click „recognize“ for the sequence from start to stop
6. you wait for the recognition result to be appended to the transcript, you may correct it if necessary
7. back to step 3. - handle the next sequence

The following demo slides shows this, using a short poem of Erich Kästner:

Irrtümer haben ihren Wert.

Jedoch nur hier und da.

Nicht jeder, der nach Indien fährt,

Entdeckt Amerika.

(Errors are of value, but only from time to time: not everybody who is going to India will discover America.)

07:02 Samstag 10. Apr.

📶 @ 100 % 🔋

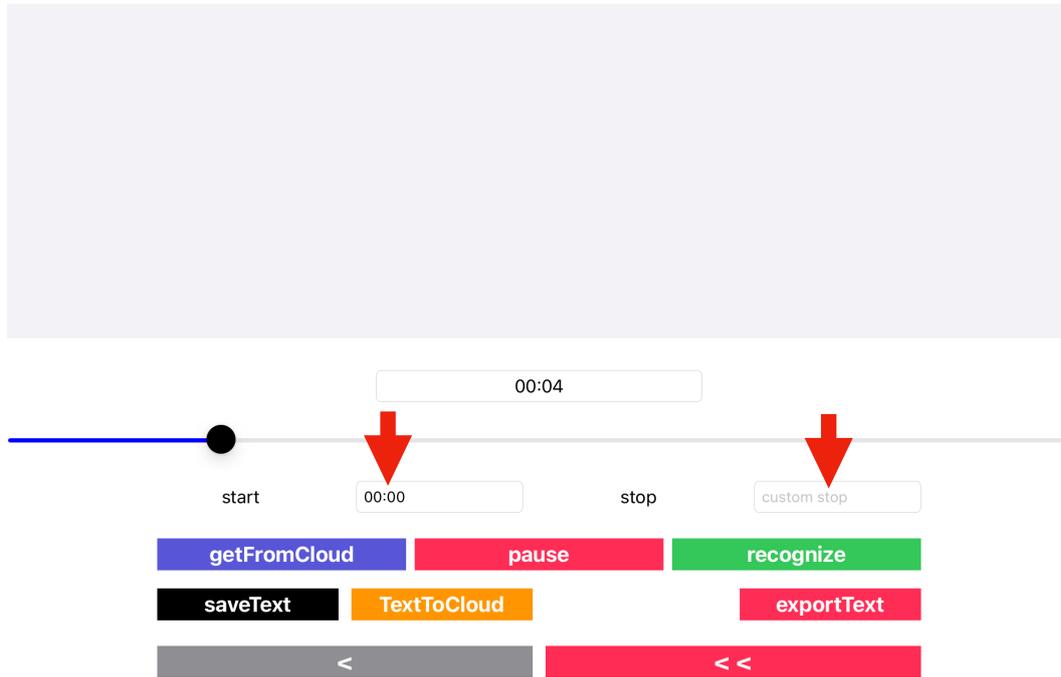
Semantic audio units (takes) recognition

Lorem ipsum dolor sit er elit lamet, consectetur cillum adipiscing pecu, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Nam liber te conscient to factor tum poen legum odioque civiuda.

The interface includes a text area with placeholder text, a 'current time' input field, a progress bar with a black dot, and several control buttons: 'start', 'custom start', 'stop', 'custom stop', 'getFromCloud', 'play', 'recognize', 'saveText', 'TextToCloud', 'exportText', and navigation arrows.

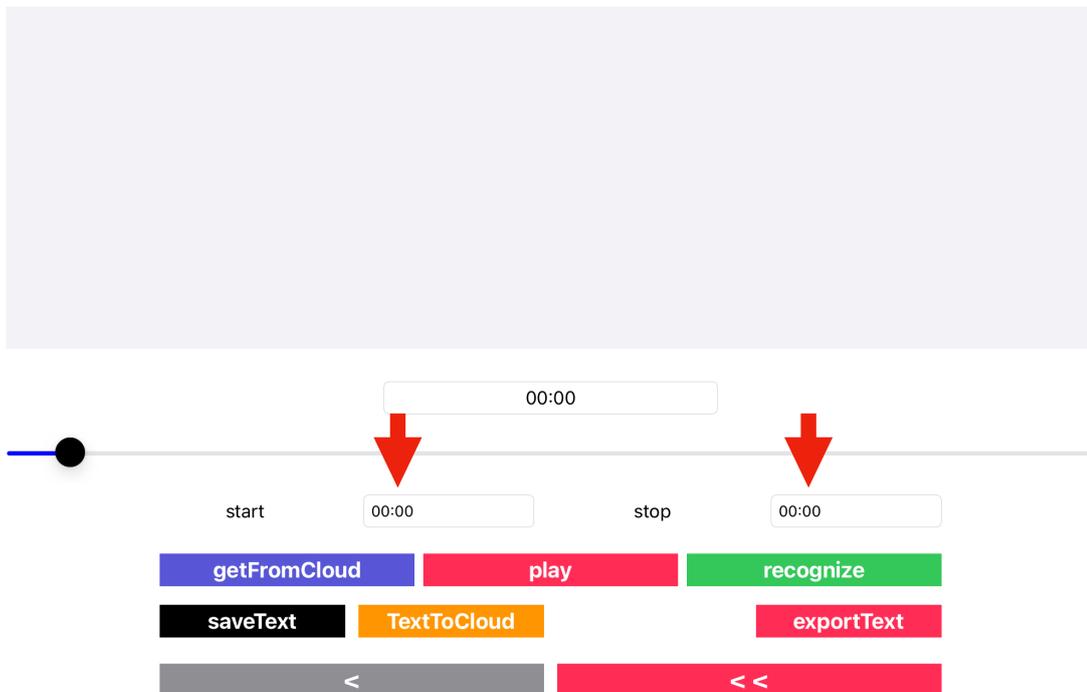
Tapped „toRecognition“ on the distribution page, arrived on the recognition screen

Semantic audio units (takes) recognition



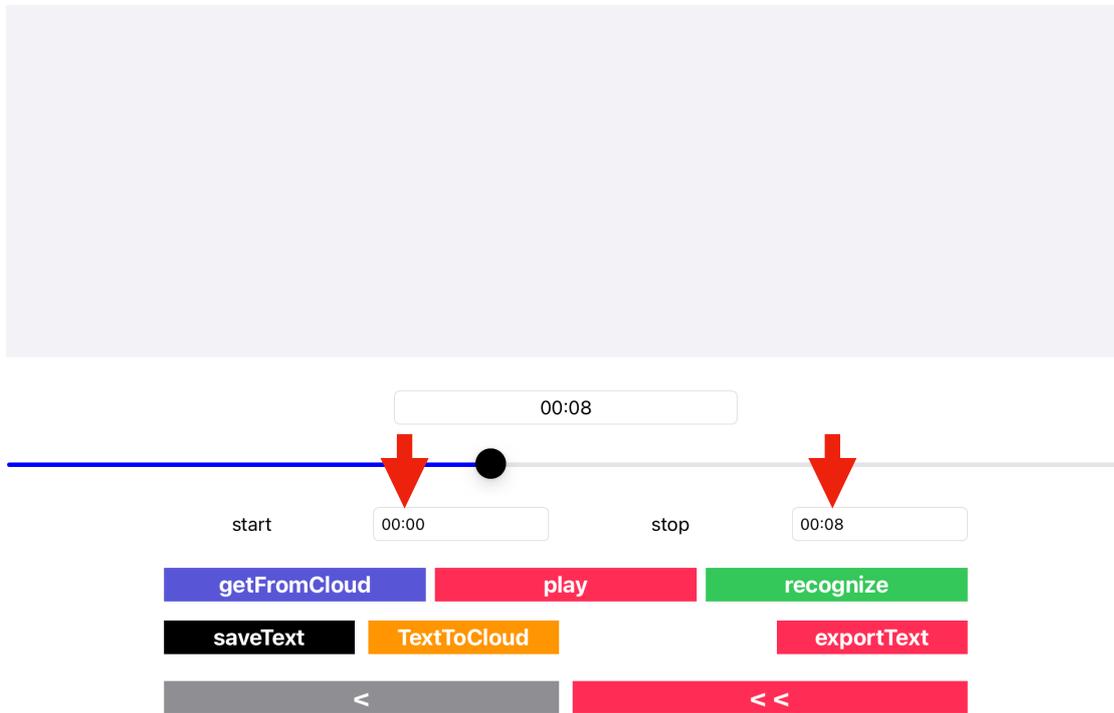
Fetches data using „getFromCloud“, tapped „play“ : we are playing

Semantic audio units (takes) recognition



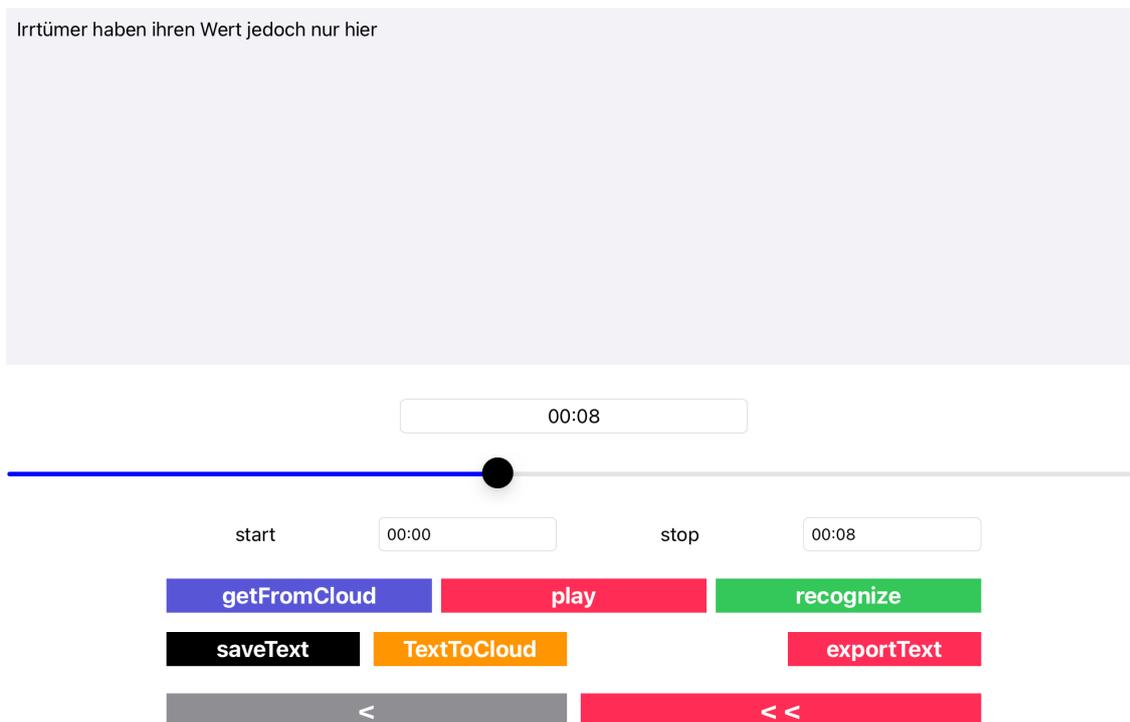
Starting recognition: audio started, stopped quickly, values in „start“ and „stop“

Semantic audio units (takes) recognition



Stopped: Sequence from 00:00 to 00:08 set

Semantic audio units (takes) recognition



Tapped „recognize“, waited till text item appeared on the transcript

Semantic audio units (takes) recognition

Irrtümer haben ihren Wert jedoch nur hier

Next: „play“ for playing, stopped, currently „start“ : 00:08, „stop“: 00:18

Semantic audio units (takes) recognition

Irrtümer haben ihren Wert jedoch nur hier Und da nicht jeder der nach Indien fährt entdeckt

Additional text item from 00:08 - 00:18 recognized and added to the transcript

20:11 Freitag 26. Feb.

100%

Semantic audio units (takes) recognition

Irrtümer haben ihren Wert jedoch nur hier Und da nicht jeder der nach Indien fährt entdeckt

00:19

start 00:18 stop 00:19

getFromCloud play recognize

saveText TextToCloud exportText

< <<

„play“ and „pause“ again: the next audio sequence 00:18 - 00:19 is picked

20:12 Freitag 26. Feb.

100%

Semantic audio units (takes) recognition

Irrtümer haben ihren Wert jedoch nur hier Und da nicht jeder der nach Indien fährt entdeckt Ja

00:19

start 00:18 stop 00:19

getFromCloud play recognize

saveText TextToCloud exportText

< <<

Last piece of text received and attached to the transcript

Semantic audio units (takes) recognition

Irrtümer haben ihren Wert jedoch nur hier Und da nicht jeder der nach Indien fährt entdeckt Ja

00:19

start 00:18 stop 00:19

getFromCloud play recognize

saveText TextToCloud not stored to cloud exportText

< <<

Transcript stored locally

Semantic audio units (takes) recognition

Irrtümer haben ihren Wert jedoch nur hier Und da nicht jeder der nach Indien fährt entdeckt Ja

00:19

start 00:18 stop 00:19

getFromCloud play recognize

saveText TextToCloud stored! exportText

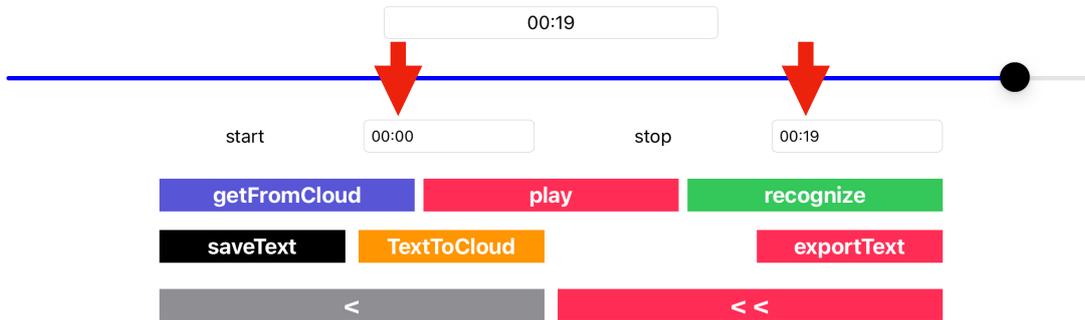
< <<

Transcript saved in the cloud record

Semantic audio units (takes) recognition



Irrtümer haben ihren Wert jedoch nur hier Und da nicht jeder der nach Indien fährt entdeckt Ja Irrtümer haben ihren Wert jedoch nur hier und da nicht jeder der nach Indien fährt entdeckt Amerika



Sent meaning unit (= the whole poem) to recognition -> better result!