

# Thinkie für Lautes Denken

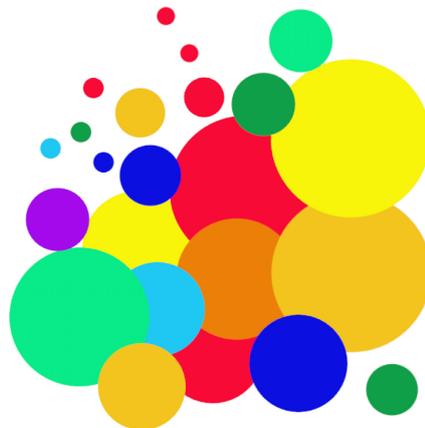
Brigitte Endres-Niggemeyer

18.09.2021

◀ App Store 18:49 Mittwoch 15. Sept.

📶 100 % 🔋

## Thinking-aloud for field studies on mobiles



iPhone

iPad

onWeb

Exit

## Hintergrundwissen Thinking Aloud / Lautes Denken

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: Lautes Denken. [https://de.wikipedia.org/wiki/Lautes\\_Denken](https://de.wikipedia.org/wiki/Lautes_Denken)

Konrad K. (2020) Lautes Denken. In: Mey G., Mruck K. (eds) Handbuch Qualitative Forschung in der Psychologie. Springer, Wiesbaden. [https://doi.org/10.1007/978-3-658-26887-9\\_41](https://doi.org/10.1007/978-3-658-26887-9_41)

Hofmann, Maria (2017). Die Erhebungsmethode des Lauten Denkens. QUASUS. Qualitatives Methodenportal zur Qualitativen Sozial-, Unterrichts- und Schulforschung. URL (<https://quasus.ph-freiburg.de/2217-2/>)

## Lautes Denken

Mit Lautem Denken („Thinking Aloud“) kann man erfahren, was Menschen durch den Kopf geht, wenn sie sich mit irgendetwas auseinandersetzen. Die Grundtechnik ist, sie zu bitten, immer zu sagen, was sie gerade denken. Dieses Protokoll zeichnet man auf. Es hält fest, was gerade jetzt gedacht wird. Im Nachhinein ist dieses Wissen größtenteils nicht mehr verfügbar, weil es vergessen oder von neueren Denkinhalten überlagert wurde. Lautes Denken liefert einen Auszug aus dem Kurzzeitgedächtnis der Person.

Mit Protokollen des Lauten Denkens kann man beispielsweise herausfinden, welche Schwierigkeiten eine Person hat, wenn sie ein neues technisches Gerät oder eine Benutzeroberfläche erforscht. Wenn Probleme durch technische Mängel oder ungünstiges Design verursacht werden, wird man sie während der Produktentwicklung beheben. Usability engineering ist dementsprechend ein wichtiges Anwendungsgebiet für Lautes Denken.

Andere Anwendungen sind leicht vorstellbar:

Was denkt jemand, der sich mit einer Skulptur auseinandersetzt?

Wie arbeitet sich ein Tourist mit Karte, Smartphone und sonstigen Informationsquellen durch eine fremde Stadt?

Wie stellt ein Installateur fest, woher das Wasser kommt, das irgendwo aus der Wand tritt?

Um Daten des Lauten Denkens zu bekommen, muss man am Ort der Handlung sein. Die Testperson braucht die normale Umgebung, sonst stimmen die Ergebnisse nicht. Die Ausrüstung für die Datenaufnahme muss deshalb transportabel, leicht handhabbar und überall einsetzbar sein. Die Testperson und die Forscherin müssen gut damit zurechtkommen. Thinkie bietet dazu ein iPhone und ein iPad an, die beide ihre Daten auf iCloud ablegen.

Was die Testperson sagt, nimmt oft Bezug auf Dinge in ihrer direkten Umgebung. Man kann die Ergebnisse oft besser interpretieren, wenn man diese Umgebung dabei sehen kann. Eine Videoaufzeichnung ist also günstig.

Das Protokoll des Lauten Denkens wird normalerweise (ganz oder teilweise) transkribiert und dann ausgewertet. Was herauskommt, wird man mit der Testperson diskutieren. Deswegen ist es gut, wenn die mobile Ausrüstung auch die Aufarbeitung der Daten unterstützt. Für später muss man die Ergebnisse exportieren können.

## Datenaufnahme mit Thinkie

Nehmen wir an, dass in einem Projekt eine Datenaufnahme mit Lautem Denken verabredet ist. Der Gegenstand der Untersuchung ist klar, die Testperson kennt sich in der Tätigkeit aus, die untersucht werden soll. Am Ort der Handlung ist alles vorhanden, was dafür benötigt wird, etwa Werkzeug und Unterlagen. Alles andere - iPhone und iPad - bringt die Forscherin mit.

Die Forscherin muss ihrerseits dafür sorgen, dass sie gut vorbereitet ist. Auf ihrem iPhone und iPad läuft Thinkie. Der Internetzugang funktioniert, der Zugang zu iCloud ist eingeschaltet. Die Daten landen im privaten Container der Forscherin, Audiodateien in der Sektion „phone“, Videos in der Sektion „pad“. Jede Datei bekommt dort einen Datensatz.

Die Testperson nimmt ihr mündliches Protokoll des Lauten Denkens mit dem iPhone auf. Gleichzeitig filmt die Forscherin das Geschehen auf dem iPad. Die Tondatei vom iPhone und die Videodatei vom iPad sind durch ihre Namen einander zugeordnet. Wenn die Aufnahme beendet ist, hört man sich den Ton auf dem iPhone eventuell noch einmal teilweise oder ganz an, dann wird er hochgeladen. Mit dem Video geht es auf dem iPad ebenso: anspielen oder abspielen, prüfen, hochladen. Sollte etwas schief gegangen sein, wiederholt man die Aufnahme.

Nun kann die Forscherin mit der Aufarbeitung der Daten beginnen. Wenn es sinnvoll ist, sieht sie sich dabei die Videoaufnahme an. In der Hauptsache befasst sie sich jedoch mit den Tondaten. Zunächst wird der gesprochene Text transkribiert, damit man ihn interpretieren kann. Dazu muss man aus der Audiosequenz Zeitfenster abgreifen, die dann entweder mit automatischer Spracherkennung oder manuell-intellektuell transkribiert werden. Beide Wege führen zu demselben Transkript.

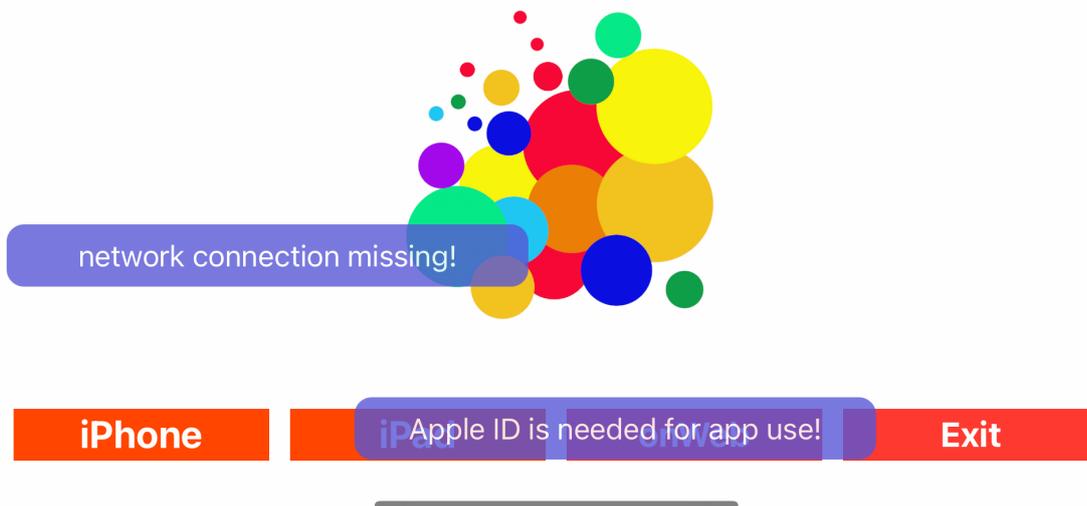
Für die manuelle Transkription werden aus der Audiodatei Abschnitte einer geeigneten Länge abgegriffen und in einer Liste zur weiteren Bearbeitung angeboten. Die Länge stellt man nach dem ein, was auf einmal umgesetzt werden kann.

Die Spracherkennung erwartet die Sprache, die als Grundsprache eingestellt ist. Man passt sie dem Bedarf entsprechend an. Hier sorgt man selbst dafür, dass Zeitfenster aus der Audiodatei segmentiert werden, die möglichst Bedeutungseinheiten entsprechen. Sie werden zur Spracherkennung geschickt. Das Resultat kommt in das Transkript und wird dort bei Bedarf korrigiert. Man lädt das Transkript auf dem iPhone in seinen Datensatz im Cloud-Container.

Sehr oft werden Testperson und Forscherin die gemeinsame Aufnahme miteinander besprechen, sobald die Daten ausreichend aufgearbeitet sind.

## Thinkie: Start von iPhone und iPad

### Thinking-aloud for field studies on mobiles



#### Start auf iPhone oder iPad: WiFi / WLAN, iCloud und AppleID fehlen

Der erste Bildschirm von Thinkie prüft, ob die Voraussetzungen für Thinkie erfüllt sind. Dann lenkt er auf das richtige Gerät.

Fehlt der Internet-Zugang, erscheint eine Fehlermeldung. Der Grund kann insbesondere der Flugmodus oder ein ausgeschaltetes WLAN sein. Außerdem muss Thinkie auf die iCloud zugreifen können. Den Cloudzugang kann man in den Systemeinstellungen einschalten.

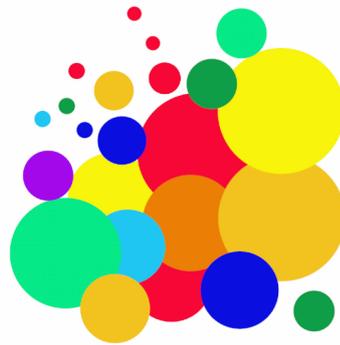
Dann wird die Autorisierung mit der AppleID abgefragt und zwar so, wie es auf dem Gerät eingestellt ist, beispielsweise per Gesichtserkennung. Die Abbildung ganz unten zeigt eine Autorisierung mit der Kennzahl des iPads. Wenn Vorbedingungen nicht erfüllt sind, scheitert die Autorisierung. Ohne geht es nicht weiter.

Die Taste „onWeb“ funktioniert immer. Sie führt zu der Thinkie-Beschreibung auf meiner Webseite. Dort findet sich insbesondere eine Anleitung für die Benutzung von Thinkie. Die „Exit“-Taste schaltet zurück ins iOS.

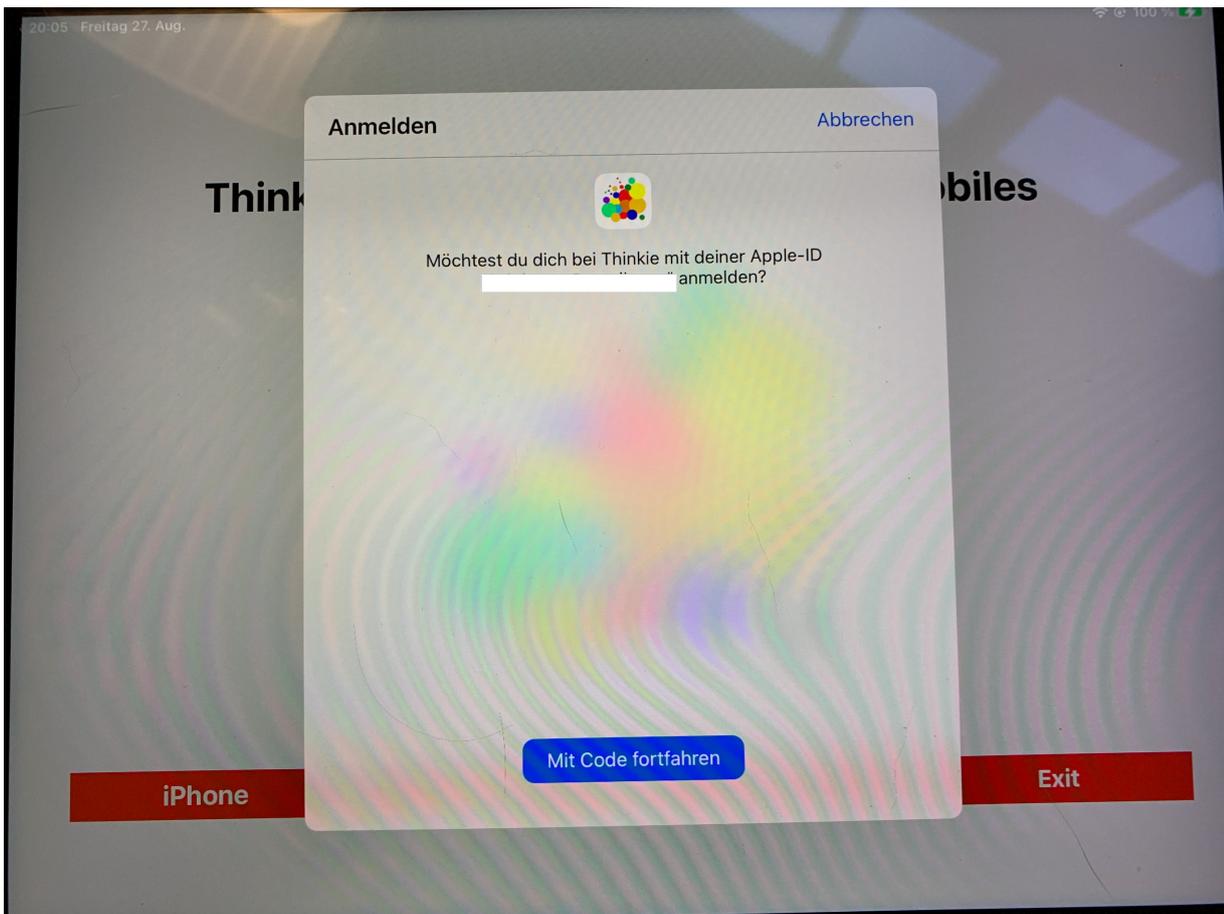
Sollte die Taste für das aktuell benutzte Gerät nicht funktionieren, hilft es, zunächst mit „onWeb“ ins Netz und von dort wieder zurück zu gehen.

iPad und iPhone laufen unabhängig voneinander. Sie liefern ihre Aufnahmen im iCloud-Container der Eigentümerin ab. Das iPhone schreibt dort in den Sektor „phone“, das iPad in den Sektor „pad“. Beide verwenden dasselbe Datensatz-Format.

# Thinking-aloud for field studies on mobiles

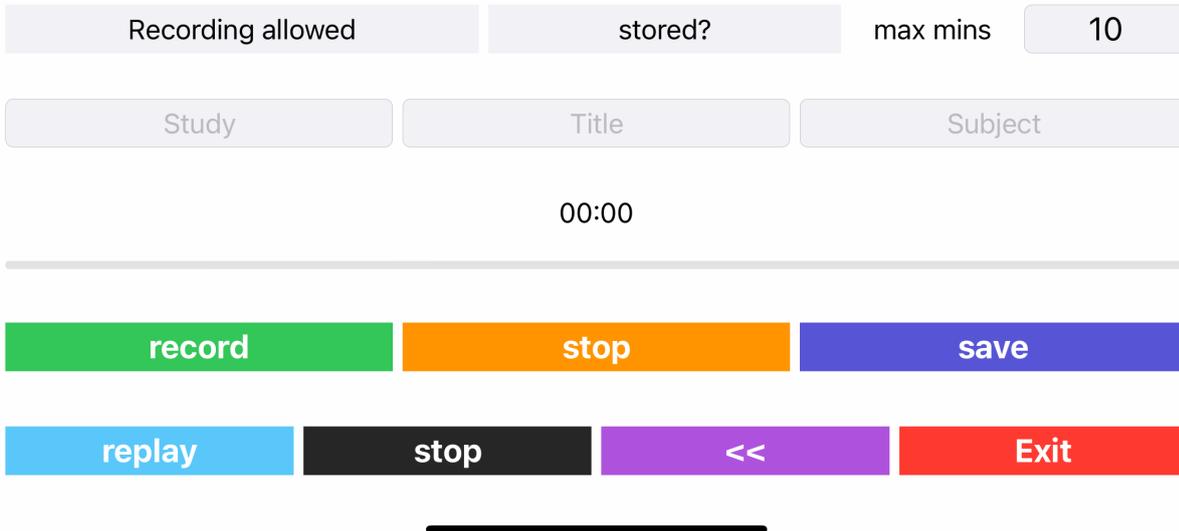


Normalfall: Zu Beginn wird die AppleID abgefragt



Autorisierung mit AppleID: Beispiel mit der Kennung auf dem iPad

## Auf dem iPhone



### Bildschirm für die Tonaufnahme im Startzustand

Auf dem iPhone geht es um den Ton. Die Testperson sagt, was sie denkt. Das wird aufgezeichnet. Die Sprache ist beliebig. Geräusche werden ebenfalls aufgenommen. Man kann die Tastatur entsprechend einstellen. Später wird bei Speech-to-Text die Sprache erwartet, die als Systemsprache verwendet wird. Im Beispiel sind als mögliche Sprachen Deutsch, Englisch, Französisch und Italienische eingestellt. Andere gehen auch.

Die Tondatei wird mit drei Elementen benannt, die standardmäßig als Studie, Titel und Person kategorisiert sind. Ihre Belegung ist beliebig.

Die Länge der Aufzeichnung justiert den mitlaufenden Schieberegler. Dauert die Aufzeichnung länger als eingestellt, bleibt der Regler stehen, die Aufnahme geht jedoch weiter.

Wenn die Aufnahme beendet ist, kann man sie sich ganz oder teilweise anhören. Im Anschluss sichert man die Audiodatei im eigenen iCloud-Container. Funktioniert es nicht, bekommt man eine Fehlermeldung. Sonst wird der Upload mit „stored!“ bestätigt.

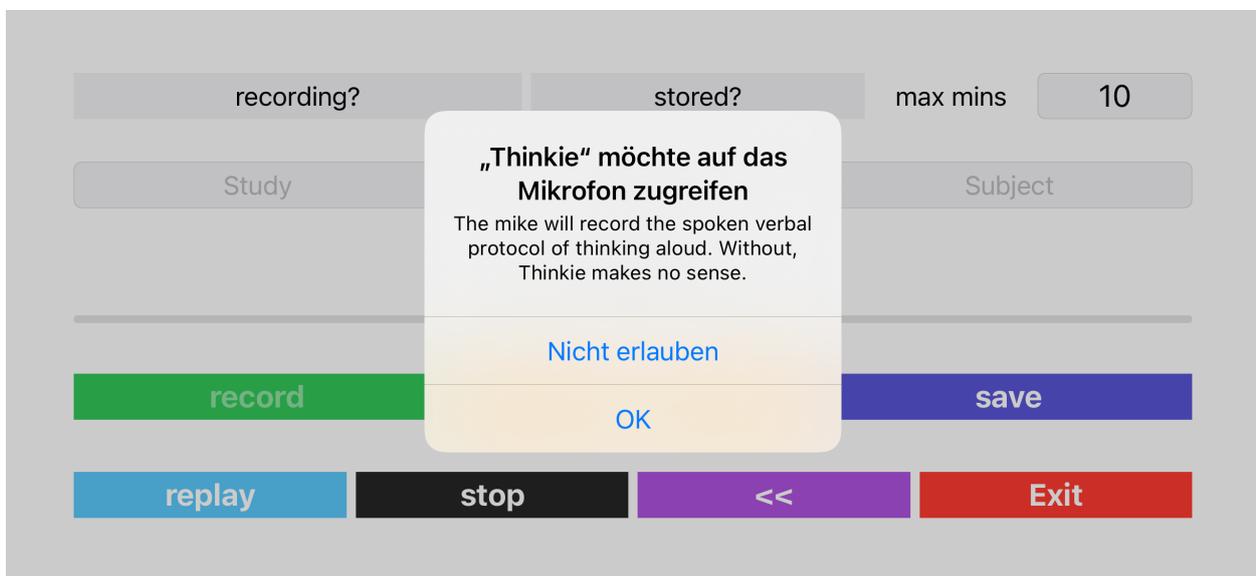
Die Datei erscheint im Erfolgsfall auf der Audio-Liste auf dem iPad.



### Keyboard beim Eintragen von Text, Sprachwahl eingeblendet

Klickt man in ein Feld für Namen, erscheint das Keyboard. Durch einem Klick in den Hintergrund verschwindet es wieder.

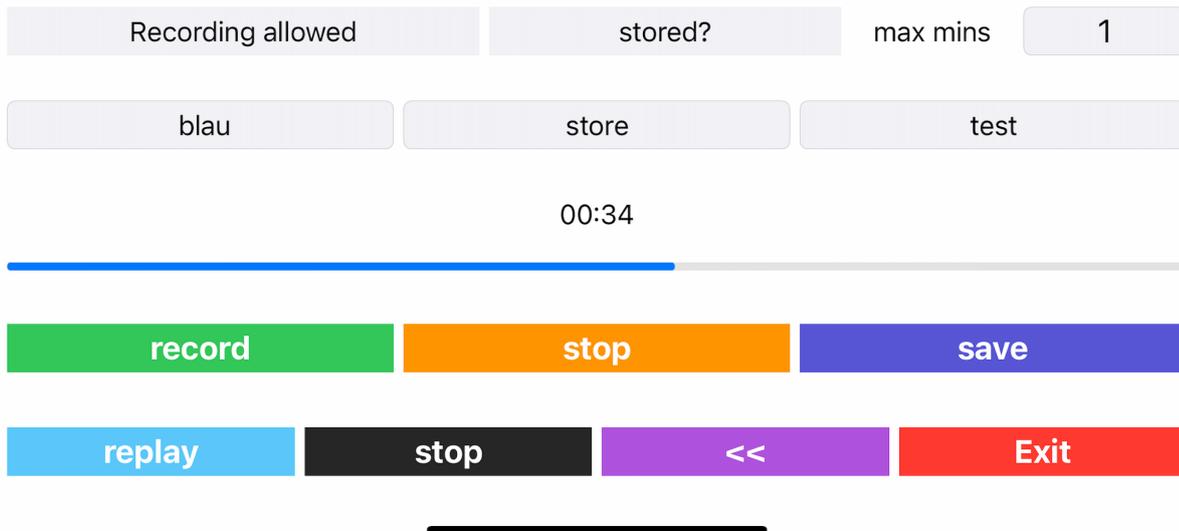
Der Regler ist auf 10 Minuten für eine Aufnahme voreingestellt. Den Wert kann man anpassen. Die Tastatur stellt sich dabei auf numerische Werte um.



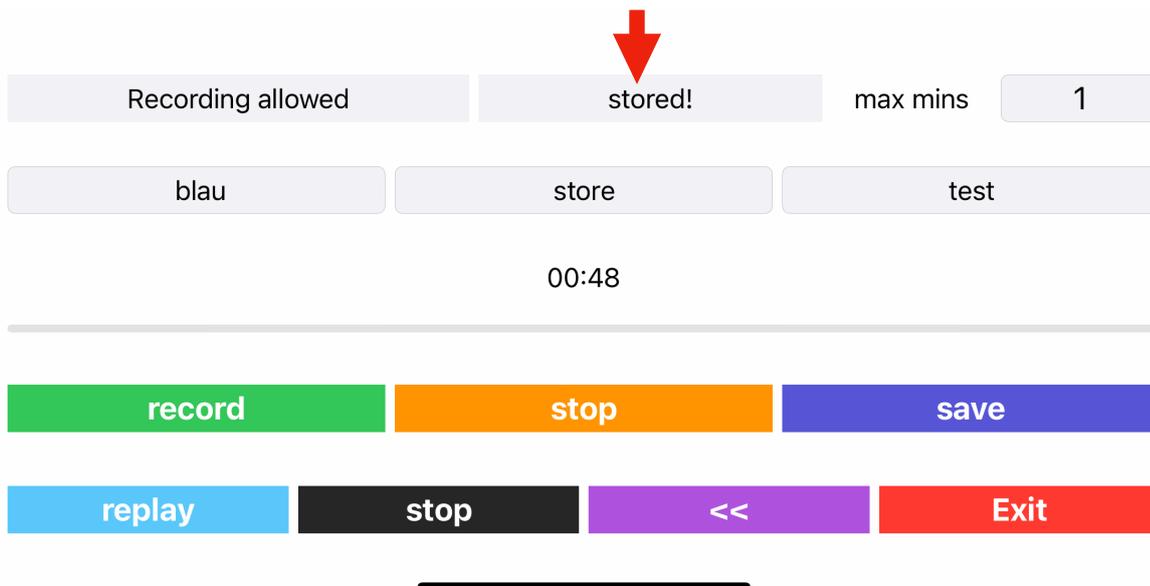
### Bei der ersten Aufnahme: Mikrofon-Zugriff erlauben

Für die Besitzerin des iPhones ist die Erlaubnis zur Aufnahme voreingestellt. „Recording allowed“ erscheint, sobald man den „record“-Button anklickt. Nur beim allerersten Aufruf erscheint die Aufforderung, dem Gebrauch des Mikrofons zuzustimmen.

Mit „stop“ beendet man die Aufnahme. Man kann sie mit „replay“ noch einmal anhören.



### Aufnahme unterwegs, noch nicht gespeichert



### Aufnahme im Cloud-Datensatz gespeichert

Die Tondatei wird mit „save“ in einem Cloud-Datensatz gesichert.

Die lokale Fassung wird nicht gespeichert. Sie wird von der folgenden Aufnahme überschrieben.

## Auf dem iPad

07:37 Montag 13. Sept.

100 % 

### Thinking-aloud for field studies on mobiles

[Use audio and video](#)[Record video](#)[toStart](#)[Exit](#)

#### Verteilerseite auf dem iPad

Auf dem iPad erscheint zunächst eine Verteilerseite.

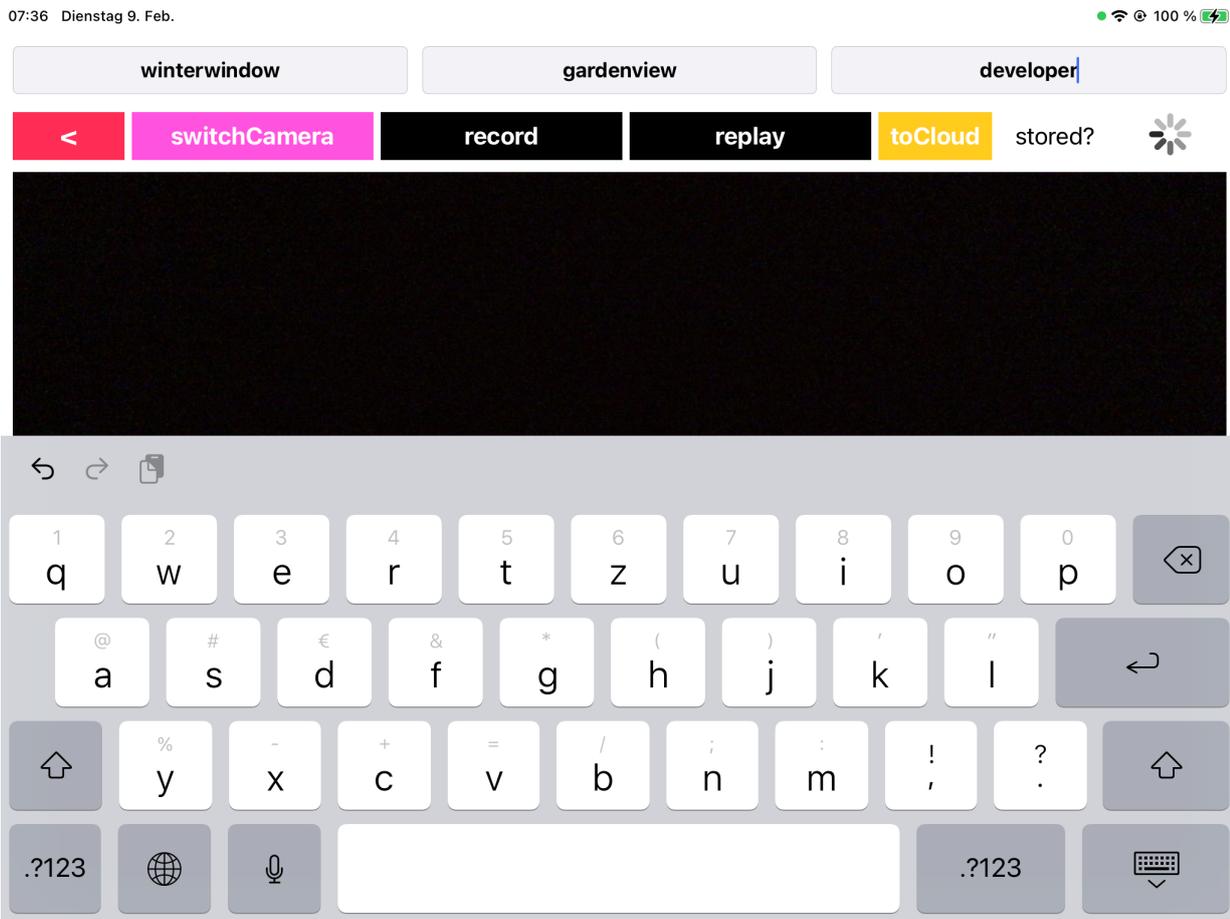
Hier kann man wählen, ob man eine vorhandene Video- oder Audiodatei benutzen oder einen neuen Video aufnehmen will.

Im ersten Fall wird man zu einer Übersicht über die im iCloud-Container vorhandenen Audio- und Videodaten geführt.

Die Taste „Record video“ schaltet direkt zur Seite für die Videoaufnahme.

„toStart“ geht zurück zur ersten Seite. Von dort kann man mit „onWeb“ beispielsweise diese Anleitung benutzen. „Exit“ schaltet zurück ins IOS.

## Video aufnehmen



### Videseite mit Keyboard

Wie Audiodateien brauchen Videodateien zunächst einen dreiteiligen Namen. Die Tastatur legt sich dazu über die Voransicht der Videoaufnahme. Sie wird mit ihrer Retourtaste rechts unten wieder ausgeschaltet.

Vor der Aufnahme kann man die Kamera nach vorne oder nach hinten ausrichten. Der Schalter „record“ startet die Videoaufnahme. Dabei wird er rot und nennt seinen aktuellen Zustand „running“. Gestoppt wird die Aufnahme, indem man den Schalter nochmals anklickt. Er wird wieder schwarz und zeigt den Zustand „record“ an.

Mit „replay“ kann man den Video zur Kontrolle anzeigen.

Mit „toCloud“ wird der Video im Cloud-Datensatz gesichert. Es kann etwas dauern, bis man die Bestätigung erhält. Also nochmals nachfragen.

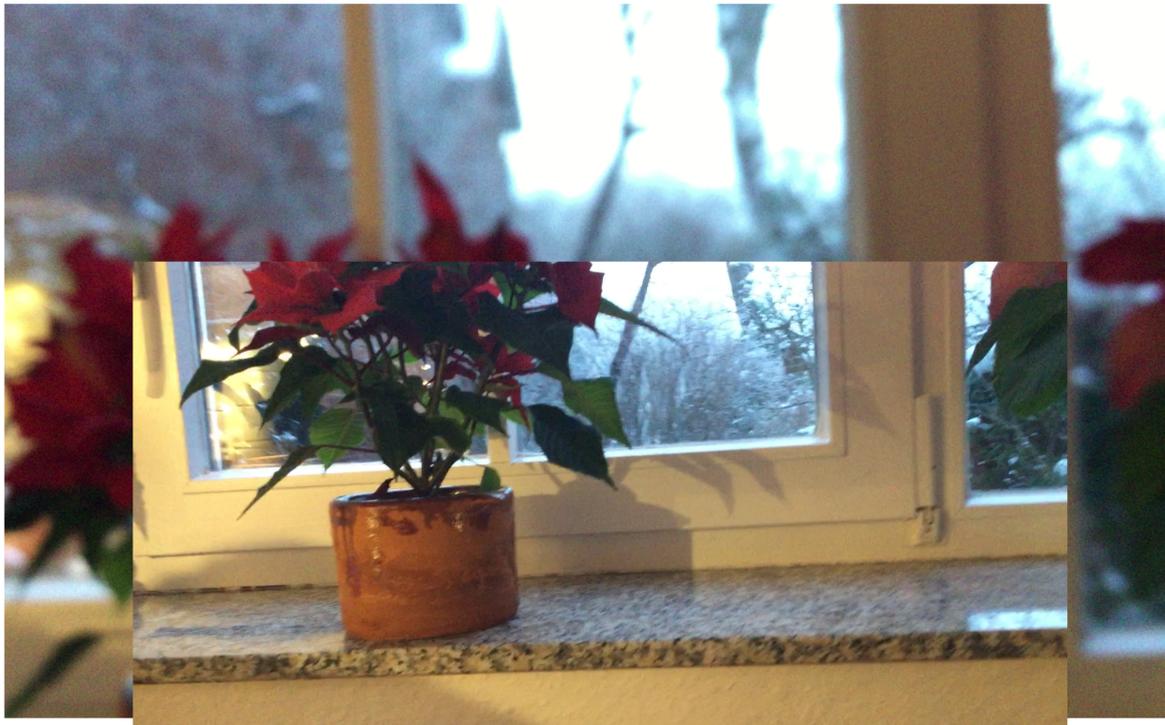
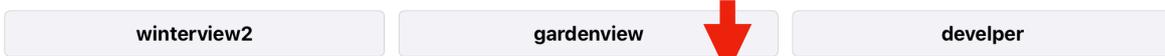
07:39 Samstag 13. Feb.

100 %



07:40 Samstag 13. Feb.

100 %



Videoaufnahme (oben) und Replay (unten)

08:28 Mittwoch 17. Feb.

100 %

uxmake error upload

< switchCamera record replay toCloud failure! 

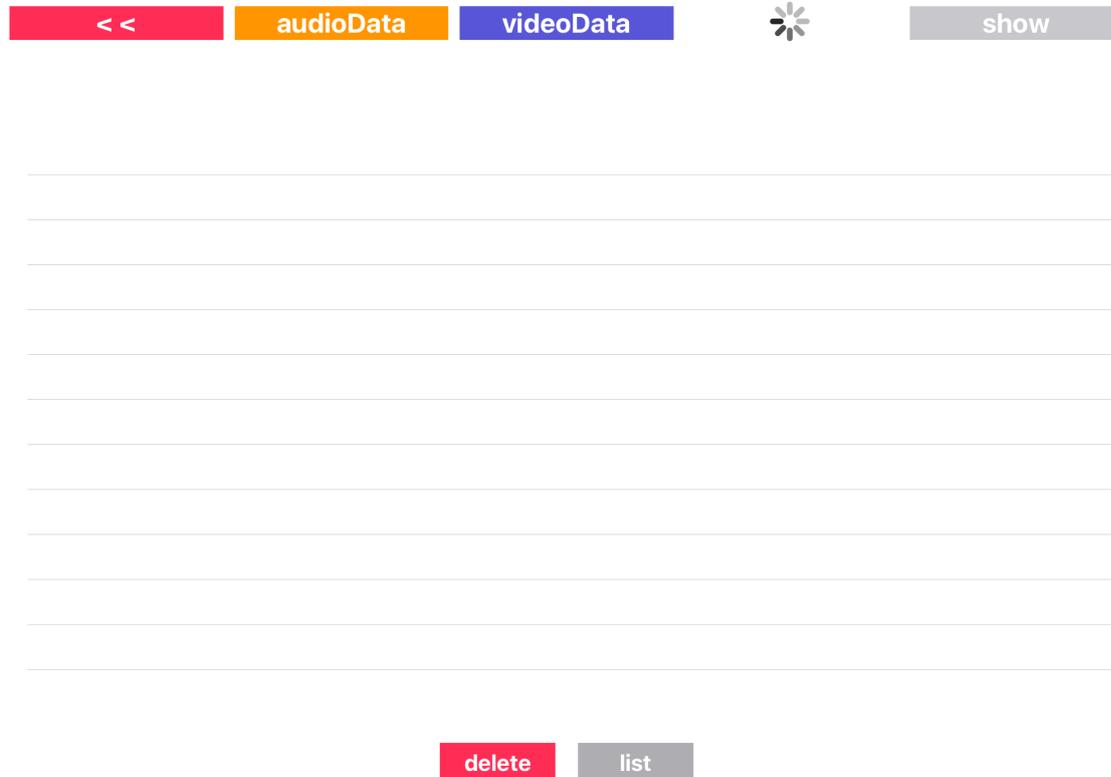


Testaufnahme noch auf dem Weg in den Clouddatensatz

## Übersicht über die Cloud-Dateien

07:41 Dienstag 9. Feb.

📶 100% 🔋



### Leere Übersichtsliste für Audio- und Videodateien

Metadaten der Audio- und Videodateien aus dem iCloud-Container werden in die Übersicht oben gestellt. Man wählt „audioData“ oder „videoData“, dann zeigt man mit „show“ die entsprechende Liste an. Sie enthält bis zu 24 Einträgen. Die neuesten Dateien stehen oben. Jeder Eintrag verweist auf seine Datei. Mit dem Schalter rechts wird sie geholt.

„delete“ beginnt das Löschen einer Datei. „list“ kehrt aus der Löschansicht der Liste in das Normalformat zurück.

Die erste Datei lässt sich nicht löschen.

07:43 Dienstag 9. Feb.

100%

Navigation: << 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

07:44 Dienstag 9. Feb.

100%

Navigation: << audioData videoData show

catprofile + sculptureview + Mollie	audio
schnee + draussen + jetzt	audio
schnee + mollie + desktop	audio
kinder + brauchen + maerchen	audio
kitchen + english + me	audio
cucina + internazionale + io	audio
alles + was + geschicht	audio
postdelete + done? + me	audio
kitchen + myhome + xmasstate	audio
kitchen + Internatinal + Cat	audio
kitchen + International + Cat	audio
cloud Store + problem + ich	audio
Video + Drehung + ich	audio

delete list

**Videodateien im Löschezustand (oben), normale Liste der Tondateien (unten)**

## Videos abspielen und exportieren

Videos, die man abspielen will, müssen zunächst mit „getVideo“ aus dem Cloud-Container geholt werden. Sie können ein großes Datenvolumen haben, der Download kann also etwas dauern. Dann kommen blaue Meldungen zur Sachlage: warten und erneut klicken.

Sobald der Video verfügbar ist, werden die Tasten zum Umgang mit dem Video aktiviert. Man kann den Video abspielen und exportieren.

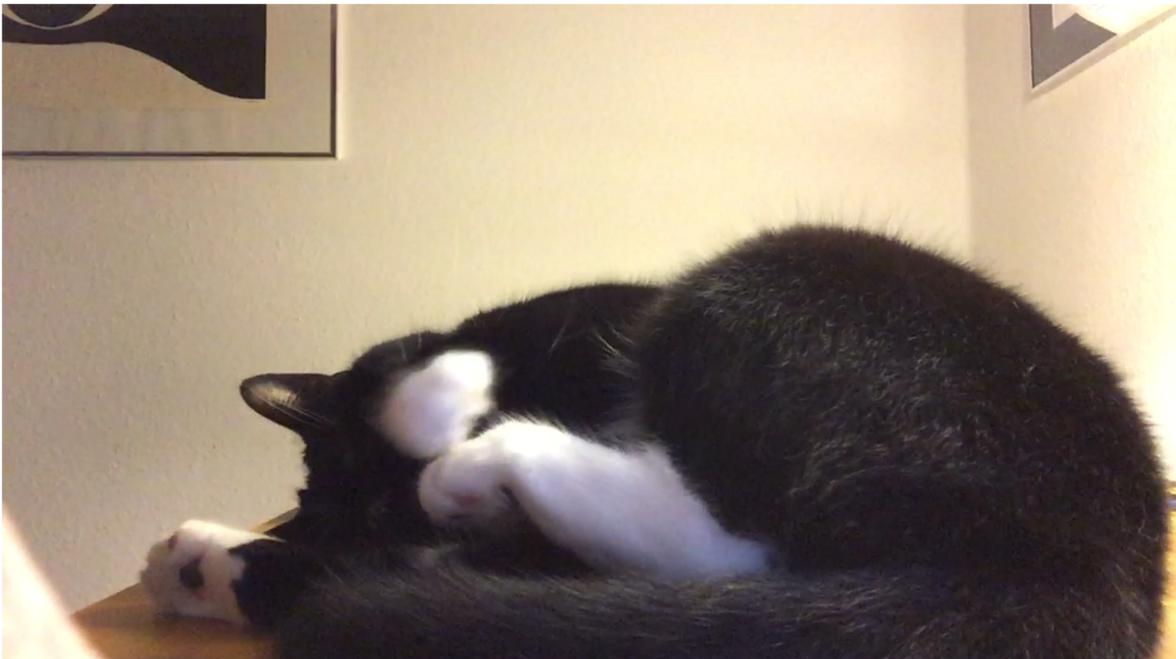
Unten sieht man, wie sich die Exportseite über den Video gelegt hat. Man klickt dort je nach Lage in der eigenen Systemumgebung eine der Optionen an.

Der Video wird aus einer lokalen Kopie namens „videoname“ entnommen. Sobald man die exportierte Fassung hat, benennt man sie besser um.

Mit „deleteRecord“ kann man den Video löschen. Eine kurze Mitteilung „done“ bestätigt, dass der Video aus dem Cloud-Container entfernt wurde.

23:34 Donnerstag 16. Sept.

📶 100% 🔋



getVideo

deleteRecord

play

exportVideo

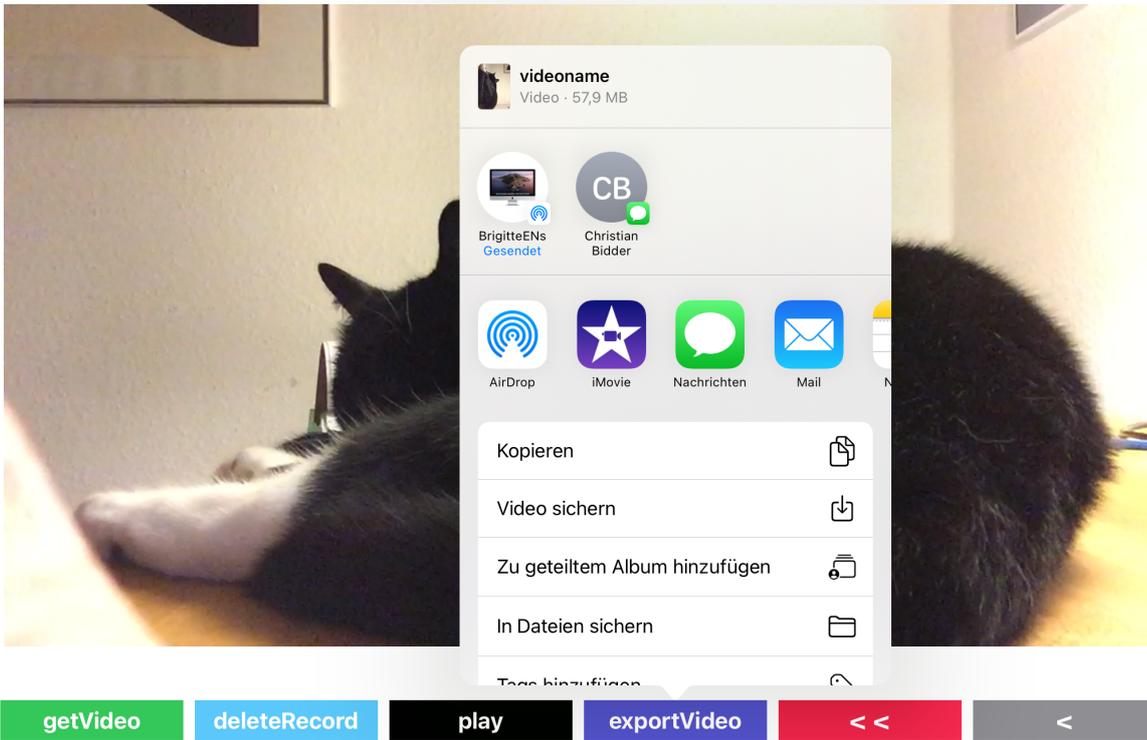
<<

<

**Video bereit zum Abspielen, zum Löschen und zum Exportieren**

07:02 Freitag 17. Sept.

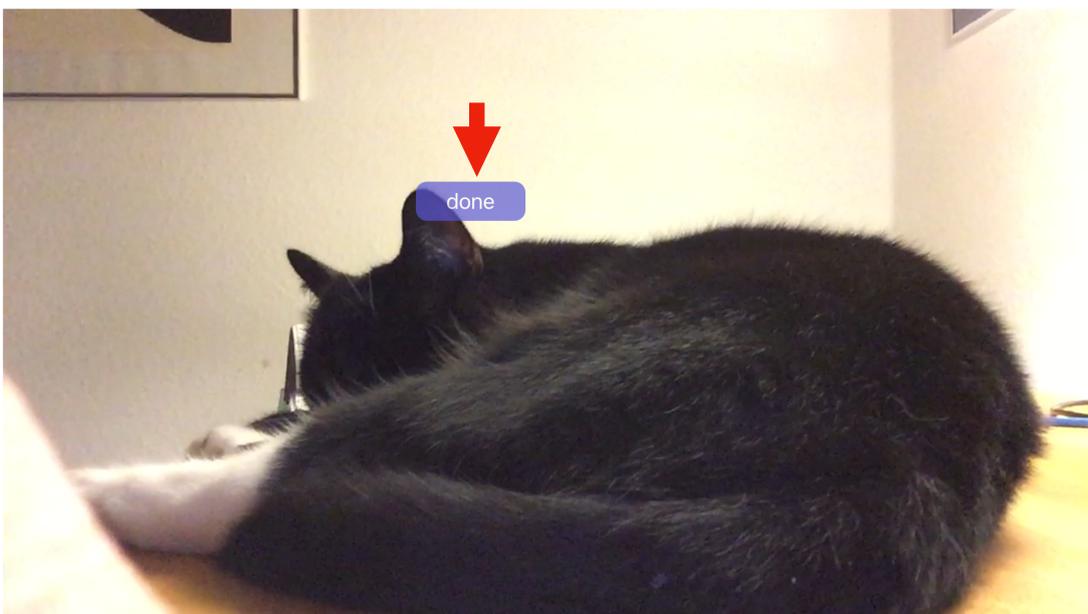
100%



### Video mit Exportseite, Export ist bereits erfolgt

21:06 Montag 20. Sept.

100%

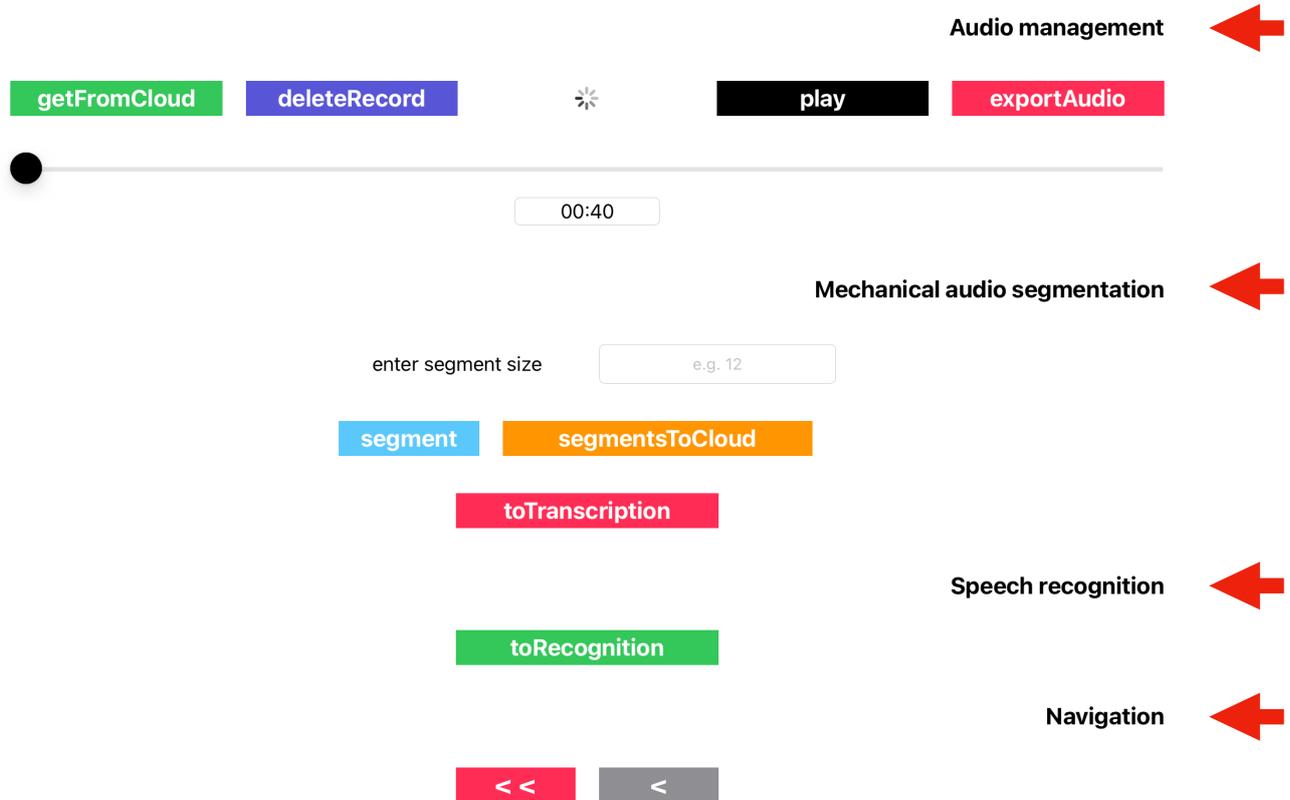


### Video gelöscht

## Sprachverarbeitung: Verteilerseite

07:44 Montag 13. Sept.

100 % 🔋



### Verteiler für Sprachverarbeitung: Ton abspielen, segmentieren, transkribieren und Speech-to-Text

Zuerst holt man die Audiodatei aus dem Cloud-Container ab. Wenn sie da ist, werden die Tasten zur Verarbeitung aktiviert. Man kann den Ton mit „play“ abspielen und mit „exportAudio“ exportieren.

Das Exportfenster entspricht dem, das oben für den Videoexport zu sehen ist. Wenn man die Audiodatei mit „deleteRecord“ löscht, bekommt man eine Bestätigung mit „done“ wie bei einem Video (siehe oben).

Im Abschnitt „Mechanical Audio Segmentation“ bestimmt man Intervalle (clips) aus der Audiodatei. Deren Länge stellt man ein. Mit „segment“ werden sie erzeugt. Der Gleitregler läuft mit, man hört den Ton. Sobald die Segmentierung erledigt ist, schreibt man das Ergebnis mit „segmentsToCloud“ in den Cloud-Datensatz. Weiter geht es mit „toTranscription“ zur Transkriptionsseite.

Unter „Speech Recognition“ wird man nur mit „toRecognition“ auf die Recognition-Seite weitergeleitet. Eine Demo unten zeigt, wie man dort vorgeht.

Die Navigation führt zurück zur Dateienübersicht und zum Start.

20:21 Freitag 17. Sept.

📶 100 % 🔋

**Audio management**

getFromCloud deleteRecord \* pause exportAudio

---

00:32

**Mechanical audio segmentation**

enter segment size

segment segmentsToCloud stored!

toTranscription

**Speech recognition**

toRecognition

**Navigation**

<< <

**Audiodatei segmentiert, Audioclips im Cloud-Datensatz gesichert**

## Transkription

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  
<  
<<

↓

●

### Transkriptseite: Daten geholt, Transkript noch leer, vier Audioclips vorhanden

Auf der Transkription-Seite erhält man mit einem „getFromCloud“ die Tondatei, den aktuellen Stand im Transkript (noch leer) und die vorhandenen Audiosegmente. Im konkreten Fall sind 4 Audioclips eingetragen.

Die Tasten „play“ und „playAudio“ sind aktiviert.

„playAudio“ spielt die ganze Tondatei ab. Mit „pause“ kann man das Abspielen unterbrechen.

Der Gleitregler steht bei 0. Er richtet sich nach der gesamten Tondatei, folgt also „playAudio“.

„play“ spielt einen Clip ab. Es schreibt seine Endposition in das noch leere Zeitfenster. Der Gleitregler reagiert nicht.

Den Text trägt man in das Transkript ein.

Die Tasten unten führen wie anderswo zurück zur Audio-Übersicht und zum Start auf dem iPad. Von dort kann man Thinkie mit dem Exit verlassen.

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
			<
			<<



### Transkript bis Clip 1 geschrieben, noch nicht gesichert

Auf dem Bildschirm oben ist die Tondatei insgesamt noch nicht abgespielt worden. Der Gleitregler steht am Anfang.

Die Clips 0 und 1 wurden abgespielt und in das Transkript geschrieben. Der Clip 2 steht noch aus.

Das Transkript wurde noch nicht lokal gesichert und auch noch nicht in den Cloud-Datensatz geschrieben.

Mit „saveText“ wird das Transkript lokal gesichert, „TextToCloud“ schreibt es in den Cloud-Datensatz. Wie weit gesichert ist, wird in dem weißen Feld angezeigt.

Unten sieht man die Meldung, dass lokal gespeichert wurde und nicht mehr. Zudem wurden alle Clips abgespielt und übertragen. Der Gleitregler steht am Ende. Die Tondatei wurde also ganz abgespielt.

Man kann das Transkript im Endzustand und auch in den Zwischenfassungen exportieren.

09:21 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	not stored to cloud
				exportText
				playAudio
				<
				<<

00:24



### Transkript lokal gespeichert (oben), im Cloud-Container gesichert (unten)

09:45 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 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



## SpeechToText: Recognition - Spracherkennung

Für SpeechToText - die Spracherkennung - braucht man sinnvolle Sequenzen aus der Tondatei, die die Erkennung nicht überfordern. Sie sollen maximal 60 Sekunden lang sein. Gut gewählte Einheiten werden besser erkannt.

Erkannt wird die Sprache, die als Systemsprache auf dem iPad eingestellt ist. Wenn man eine andere Sprache will, stellt man die Systemsprache um. Der Verwendung der Spracherkennung muss man zustimmen.

Thinkie greift aus der Tondatei eine angegebene Sequenz ab und schickt sie an die Spracherkennung. Sobald das Ergebnis eingeht, wird es an das Ende des Transkripts angehängt. Wenn man will, kann man es sofort bearbeiten.

Die Teilsequenzen greift man nacheinander beim Abspielen der Datei ab. Mit „play“ beginnt man eine Sequenz, mit „pause“ stoppt man sie.

Die aktuellen Start- und Stoppwerte erscheinen auf Bildschirm. Ein Startfeld zeigt den Beginn der Sequenz an, im Stoppfeld steht das Ende. Wenn man die nächste Teilsequenz abspielt, wird das Ende von zuvor ins Startfeld geschrieben. Wenn man die Wiedergabe stoppt, wird der Zeitpunkt in das Stoppfeld geschrieben.

Mit „recognize“ schickt man die ausgewählte Sequenz zur Spracherkennung. Sobald sie im Transkript erscheint, kann man die nächste Sequenz abgreifen und zur Erkennung schicken.

Und nun ganz praktisch:

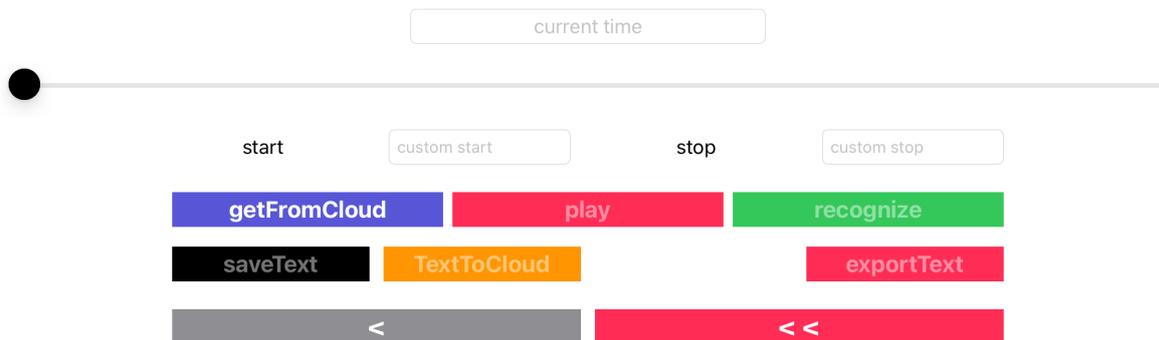
Mit „play“ kann man den Ton abspielen. Sobald man auf „pause“ drückt, wird der Stopp-Zeitpunkt in das Feld „stop“ eingetragen. Die Taste bringt zurück auf „play“. Klickt man dann auf „play“, läuft der Ton weiter und der vorherige Stopp-Wert wird in das Feld „start“ geschrieben. In „stop“ erscheint der aktuelle Wert.

Mit „recognize“ besorgt die Spracherkennung für die Sequenz von „start“ bis „stop“ das entsprechende Textsegment. Es wird an das Transkript angehängt.

Klickt man nun auf „play“, wiederholt sich das Spiel: der Stoppwert von zuvor wandert in den Startwert. Sobald man auf „pause“ klickt, wird der neue Stoppwert eingetragen, „play“ kommt zurück. Man kann wieder die Spracherkennung starten.

### 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.



### Mit „toRecognition“ von der Verteilerseite auf der Recognition-Seite angekommen

Das Vorgehen in Form einer Wiederholschleife:

1. „play“ drücken
2. „stop“ drücken - der Stoppwert erscheint in seinem Feld
- Dann für jede Einheit:
3. „play“ drücken - der alte Stoppwert wandert in das Startfeld
4. „stop“ drücken - ein neuer Stoppwert erscheint im Stoppfeld
5. „recognize“ für die Sequenz zwischen Start und Stopp anklicken
6. abwarten, bis das Ergebnis der Spracherkennung an das Transkript angehängt ist, dann eventuell korrigieren
7. zurück zu Schritt 3. : die nächste Sequenz verarbeiten.

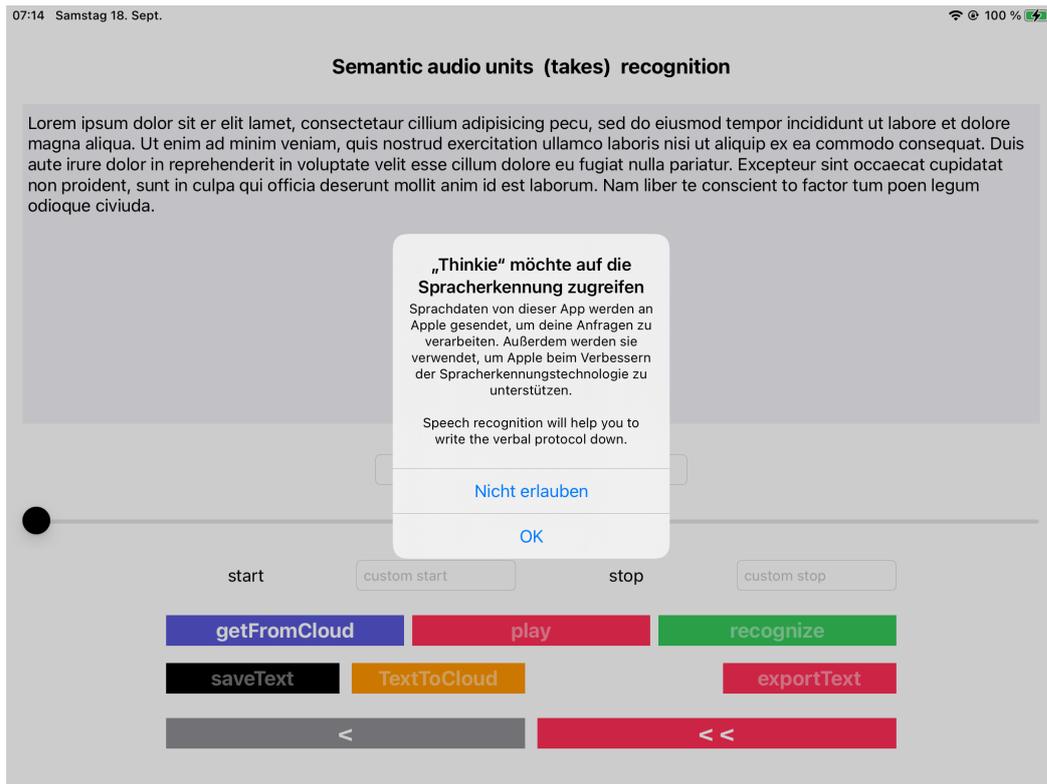
Speech-To-Text wird an einem kurzen Gedicht von Erich Kästner demonstriert:

Irrtümer haben ihren Wert.

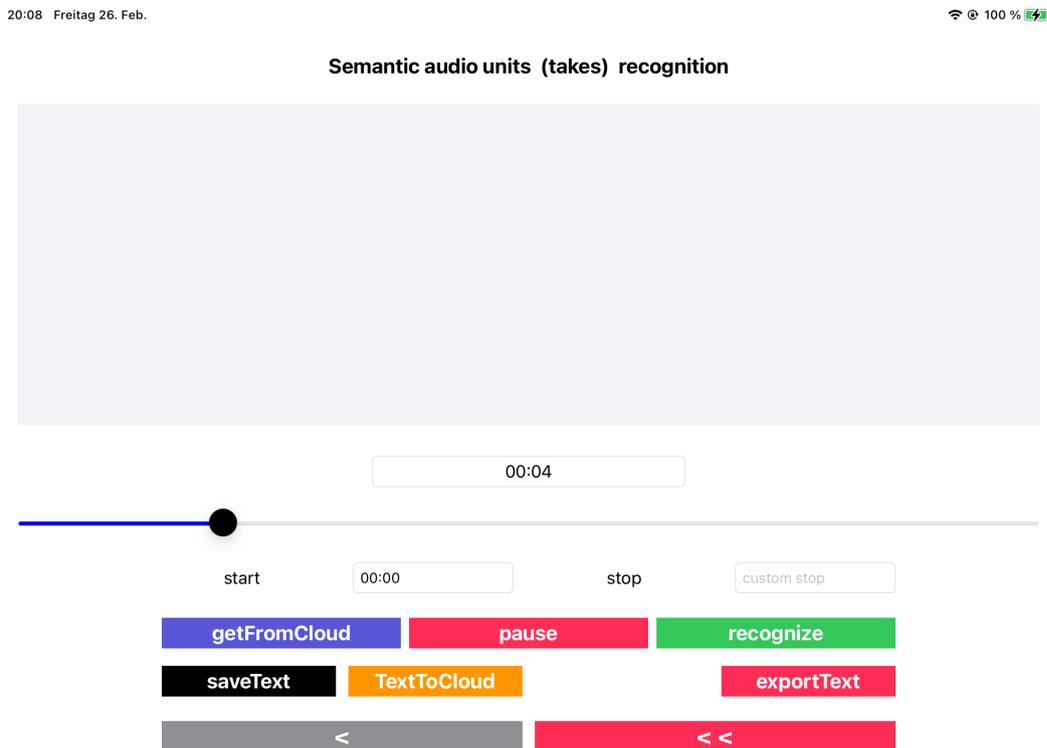
Jedoch nur hier und da.

Nicht jeder, der nach Indien fährt,

Entdeckt Amerika.

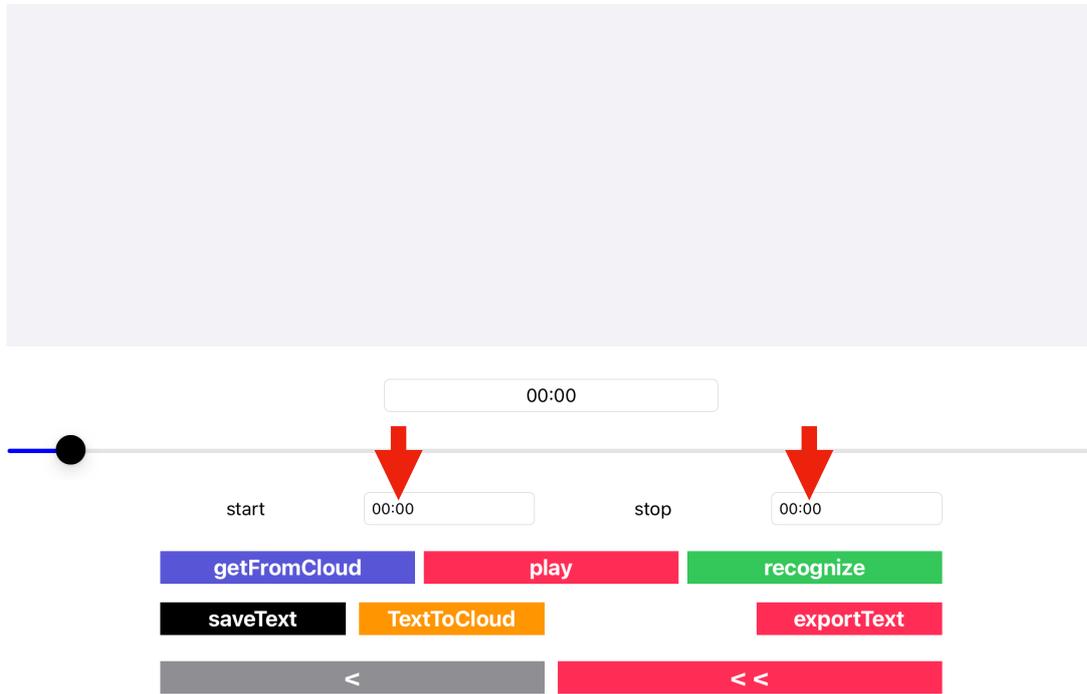


### Die Spracherkennung erlauben



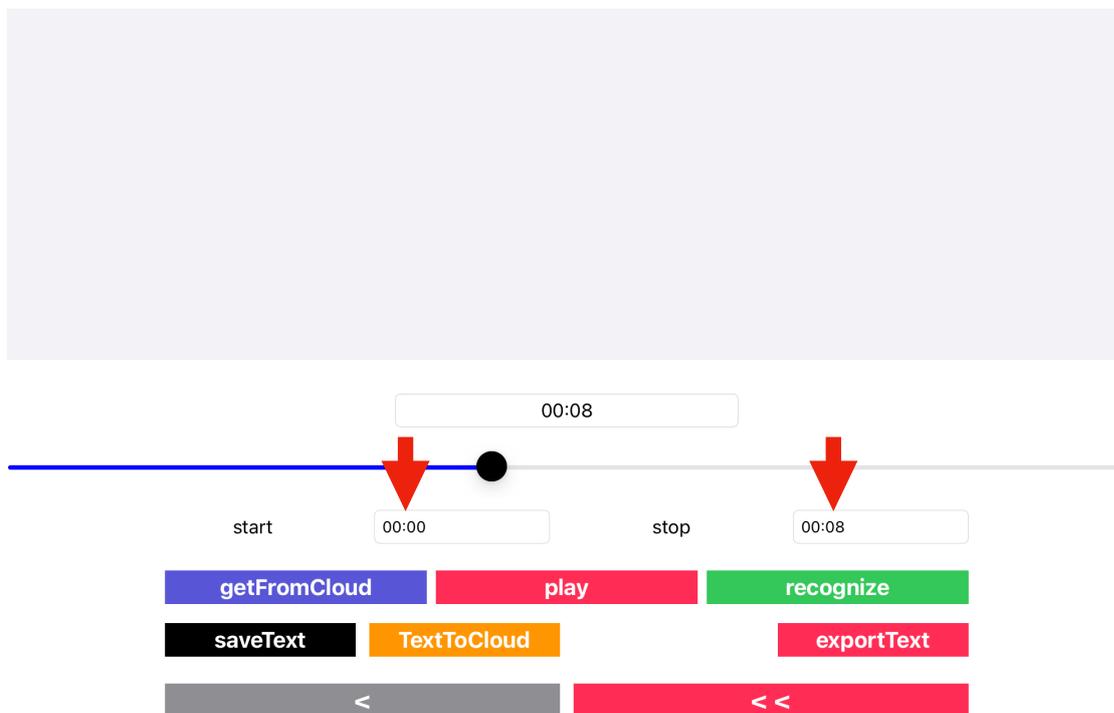
Mit „getFromCloud“ Daten geholt, „play“ angeklickt, es wird abgespielt

### Semantic audio units (takes) recognition



**Beginn Erkennung: Ton gestartet, schnell wieder gestoppt, „start“ und „stop“ gefüllt**

### Semantic audio units (takes) recognition



**Gestoppt: Sequenz von 00:00 bis 00:08 festgelegt**

### Semantic audio units (takes) recognition

Irrtümer haben ihren Wert jedoch nur hier

00:08

---

start 00:00 stop 00:08

getFromCloud play recognize

saveText TextToCloud exportText

< <<

Mit „recognize“ Textstück erhalten, wurde ins Transkript geschrieben

### Semantic audio units (takes) recognition

Irrtümer haben ihren Wert jedoch nur hier

00:18

---

start 00:08 stop 00:18

getFromCloud play recognize

saveText TextToCloud exportText

< <<

Weiter: mit „play“ abgespielt, gestoppt, nun „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

00:18

start 00:08 stop 00:18

getFromCloud play recognize

saveText TextToCloud exportText

< <<

**Zusätzliches Textstück von 00:08 - 00:18 geholt und ins Transkript kopiert**

### 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

< <<

**Wieder „play“ und „pause“: nächste Tonsequenz 00.18 - 00:19 abgegriffen**

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

A screenshot of a media player interface. At the top, the title "Semantic audio units (takes) recognition" is displayed. Below it is a text area containing the sentence: "Irrtümer haben ihren Wert jedoch nur hier Und da nicht jeder der nach Indien fährt entdeckt Ja". A progress bar shows the current time at 00:19. Below the progress bar, there are two input fields for "start" (00:18) and "stop" (00:19). A row of control buttons includes "getFromCloud" (blue), "play" (red), and "recognize" (green). Below this row are "saveText" (black), "TextToCloud" (orange), and "exportText" (red). At the bottom, there are navigation buttons: a grey "<" button and a red "<<" button. A red arrow points from the title to the text area.

### (Rest)-Textstück von Spracherkennung bekommen und angehängt

20:13 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

A screenshot of the same media player interface as above. The text area contains the same sentence. The progress bar is at 00:19. The control buttons are the same. In this screenshot, two red arrows point to the "getFromCloud" button and the "recognize" button. The "TextToCloud" button is now greyed out and labeled "not stored to cloud".

### Transkript lokal gesichert

### Semantic audio units (takes) recognition

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



### Transkript im Cloud-Datensatz gesichert

### 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



Jetzt ganzes Gedicht auf einmal zur Erkennung geschickt, besseres Ergebnis(!) angehängt