

```
ss TIANE:
```

```
def __init__(self):
    self.Modules = Modules
    self.Analyzer = Analyzer

    self.active_modules = {}
    self.continuous_modules = {}
    self.rooms = Rooms
    self.rooms_connecting = Rooms_connecting
```

```
self.start
```

```
self.local
self.user
self.room
self.serv
```

```
def start_room
```

```
if user =
    user
    return se
```

```
def route_say
```

```
if raum =
    # we
    if user == None:
```

```
    print( [WARNING] Der text {} könnte nicht gesagt werden, weil weder ein raum noch ein Nutzer
```

```
    return
    # Der Text soll zu einem bestimmten user gesagt werden
```

```
current_waiting_room = ('',None)
```

```
while True:
```

```
    for name
```

```
        if
```

```
            if not cancel_response == 'ongoing':
```

```
                break
```

```
            time.sleep(0.03)
```

```
        if cancel_response == False:
```

```
            # Konnte nicht abgebrochen werden, wurde bereits gesagt
```

```
            # Und Ja, das heißt wirklich "wurde bereits gesagt" und nicht "wird gerade ges
```

```
            # weil in dem Fall im Raum die Requests gar nicht erst bearbeitet werden können
```

```
            return
```

```
            # Alles okay, wir fragen bei einem anderen Raum nach
```

```
            current_waiting_room[1].request_end_Conversation(original_command)
```

```
            current_waiting_room = (name,room)
```

```
            room.request_say(original_command,text,raum,user,send=True)
```

```
            if room.request_say(original_command, text, raum, user) == True:
```

```
                return
```

```
            time.sleep(0.03)
```

```
        else:
```

```
            # Der Text soll in einem bestimmten Raum gesagt werden
```

```
            for name, room in self.rooms.items():
```

```
                if name.lower() == raum.lower():
```

```
                    # Dem Raum den Auftrag erteilen, es zu sagen
```

```
                    room.request_say(original_command, text, raum, user, send=True)
```

# TIANE

## Installationsanleitung

**FERDINAND und KLARA KRÄMER**

**Bundeswettbewerb Jugend Forscht 2019**

**Mathematik/Informatik**

## **Hardware und Vorbereitungen**

Um TIANE bei dir zu Hause einrichten zu können, benötigst du natürlich zunächst mal passende Geräte.

Dafür ist es wichtig zu wissen, dass TIANE im Einsatz als Server-Client-System aufgebaut ist: Ein zentraler Home-Server – zum Beispiel ein ausrangierter Arbeitscomputer – dient als Zentrale für die Verwaltung des Netzwerks sowie für rechenintensive Aufgaben wie die Bilderkennung, in den mit TIANE ausgestatteten Räumen genügt dann als Hardware zum Beispiel jeweils ein Raspberry Pi (RasPi), an den Mikrofon, Kamera, Lautsprecher sowie jegliche Smart-Home-Hardware für den jeweiligen Raum angeschlossen werden.

### **Die Räume**

Als Hardware in den Räumen, die mit TIANE ausgestattet werden sollen, empfehlen wir dringend den günstigen Kleincomputer Raspberry Pi: Dieser bietet genügend Leistung für TIANE und hat viele Schnittstellen wie WLAN, Bluetooth und die nützlichen GPIO-Pins, über die du später eigene Smart-Home-Geräte oder sonstige Projekte mit TIANE verbinden und steuern kannst. Vor allem aber hat der RasPi als bekanntestes Gerät seiner Klasse stets eine sehr gute Software-Unterstützung: Die Einrichtung geht mittels zahlloser Anleitungen im Internet selbst absoluten Einsteigern leicht von der Hand, und sollte doch mal Hilfe benötigt werden, ist diese in Foren schnell zu finden. In unseren eigenen TIANE-Aufbauten verwenden wir das derzeit leistungsstärkste Gerät der RasPi-Familie („3B+“) mit dem Standard-Betriebssystem „Raspbian (Stretch)“, mit jedem anderen Modell der 2er- oder 3er-Serie sollte unser Programm aber ebenfalls ohne Probleme kompatibel sein. Nur etwas erfahreneren Nutzern empfehlen wir dagegen die Verwendung eines anderen Betriebssystems oder gar einer anderen Hardwareplattform für die Raumclients, da für diese zumindest diese Installationsanleitung nur recht eingeschränkt gültig sein kann.

Da diese Raum-RasPis vor allem später mit der vollen Verkabelung doch recht unordentlich aussehen können, empfiehlt es sich, sie versteckt unterzubringen: Im Medianschrank, im Regal hinter Büchern, in Schränken mit Kabeldurchführung in der Rückwand oder, gerade in Aufbauten mit Kameras, gerne auch oben auf dem Schrank. In Schlafräumen sollte man zudem erwägen, die kleinen LEDs auf der Platine abzukleben oder ein geschlossenes Gehäuse zu verwenden.

Neben dem RasPi-„Gehirn“ benötigt TIANE in den Räumen aber natürlich auch noch mindestens „Mund“ und „Ohren“, um ihrer Aufgabe als Sprachassistentin gerecht werden und mit dir sprechen zu können, optional unterstützt sie zudem noch Kamera-„Augen“, mit denen sie dich von ihren anderen Nutzern im Haushalt unterscheiden und dir so personalisierte Informationen zukommen lassen kann.

Für die Tonausgabe empfiehlt sich natürlich besonders eine schon vorhandene Stereo- oder gar Surroundanlage, die du an die 3,5mm-Klinkenbuchse des RasPis anschließen und so TIANE mit volltönender Stimme erleben kannst, ein kleiner, billiger Lautsprecher im Regal tut es allerdings auch. Du solltest vielleicht darauf achten, dass dieser eine aktive Stromversorgung besitzt, da der Audioausgang des RasPis nur recht begrenzt Energie liefert.

Als Mikrofone für die Tonaufnahme benutzen wir in unseren Aufbauten kleine USB-Konferenzmikrofone, die im Internet für unter 20€ zu finden sind. Diese bieten zwar eine miserable, aber eben ausreichende Tonqualität und den großen Vorteil, dass sie gut einen gesamten Raum abdecken und ein Sprachkommando aus 3m Entfernung kaum anders aufnehmen als eines aus nächster Nähe. Als gute Faustregel zur Platzierung des Mikrofons hat sich bei uns „etwa in der Mitte der langen Seite des Raumes möglichst weit vorne auf einem Regalbrett unter Hüfthöhe“ erwiesen, aufgrund der begrenzten Länge der Anschlusskabel kann das auch als zusätzliches Kriterium zur Platzierung des Raum-RasPis angesehen werden. In einigen Fällen, gerade in kleinen oder wenig schallschluckenden Räumen, genügen unserer Erfahrung nach aber auch schon die eingebauten Mikrofone der USB-Webcams.

Diese Kameras, sofern du denn welche einrichten möchtest, werden von TIANE vor allem für Gesichtserkennung genutzt und sollten daher so angebracht werden, dass sie den Raum möglichst auf Kopfhöhe gut überblicken. TIANE unterstützt theoretisch beliebig viele Kameras pro Raum, sollen Auflösung und Bildrate allerdings vernünftig bleiben, ist aufgrund der geringen Bandbreite der USB-Ports des Raspberry Pi in der Praxis meist bei zwei Kameras Schluss. Vorzugsweise in gegenüberliegenden Raumecken oben auf Schränken liegend angebracht, erfüllen diese aber schon den Zweck. Für die meisten Räume dürfte sogar eine einzige, auf den Eingangsbereich gerichtete Kamera ausreichen, da TIANE ihre Nutzer nicht permanent sehen können muss, sondern sich ihren letzten Aufenthaltsort merkt. Das genaue Modell der Webcams ist hier völlig egal; neben der riesigen Auswahl an billigen USB-Webcams unterstützt TIANE auch die RasPi-eigene „PiCam“, die an einen eigens dafür vorgesehenen Slot angeschlossen wird und auch als „NoIR“-Variante mit (gewissen) Nachtsicht-Fähigkeiten erhältlich ist, was in manchen Fällen von Vorteil sein kann.

## Der Server

Zur Koordinierung von Aufgaben über Räume hinweg sowie für manche aufwändige Berechnungen wie z.B. Gesichtserkennung benötigt TIANE einen kleinen, zentralen Heimserver. In unseren eigenen Aufbauten verwenden wir dafür einen alten Bürocomputer (~10 Jahre), aufgestellt in einem Raum, wo die Lüftergeräusche nicht stören, und haben damit kaum Probleme. Das entscheidende Element ist hier die zusätzlich gekaufte Nvidia-CUDA-fähige Grafikkarte aus der 50€-Kategorie: Für die machine-learning-basierte Gesichtserkennung in Echtzeit ist diese selbst bei Verwendung eines wesentlich besseren Prozessors unerlässlich, andererseits macht bereits eine solche Grafikkarte einen recht alten PC absolut fit für den Einsatz als TIANE-Server. Achtung: Für das erstmalige *Training* der Gesichtserkennung können die Hardware-Anforderungen je nach Anzahl und Größe der verwendeten Fotos durchaus höher sein (ca. 8GB RAM + 4GB Grafikspeicher), falls du beim Training also auf Probleme stößt, solltest du es auf einem anderen Computer probieren (nur das Training, zur späteren Ausführung der Gesichtserkennung sollte ein Computer mit den oben genannten Spezifikationen für einen TIANE-Server genügen).

Solltest du in deinem TIANE-System aber *gar keine* Gesichtserkennung verwenden wollen, kannst du den zusätzlichen Server-Computer sogar ganz weglassen: Das TIANE-Server-Programm braucht zur reinen Verwaltung des TIANE-Netzwerks kaum Leistung und kann parallel zum Client-Programm auf einem beliebigen der Raum-RasPis ausgeführt werden.

Als Betriebssystem für den TIANE-Server empfehlen wir dringend eine aktuelle Ubuntu-LTS-Version (16.04 oder 18.04); mit anderen Linux-Distributionen sollte unser TIANE-Programm zwar kompatibel sein, die Installation könnte aber zusätzliche Schritte erfordern, Windows und MacOS werden derzeit ausdrücklich nicht unterstützt, da hierfür Modifikationen am Programm nötig wären.

## **Das Netzwerk**

Damit die einzelnen TIANE-Komponenten im ganzen Haus flüssig zusammenarbeiten können, müssen sie alle mit dem selben Heimnetzwerk verbunden werden. In unserer Testanlage verwenden wir dafür die optimale Lösung, bei der alle Räume und der Server mit hinter Fußleisten verlegten, flachen Gigabit-LAN-Kabeln und den dazugehörigen Switches in Schränken verbunden sind, bei gutem Empfang oder wenn keine Kamerabilder oder sonstige größere Datenmengen gesendet werden sollen, kann aber auch eine WLAN-Verbindung genügen. Zu beachten ist, dass alle Geräte mit dem selben Router verbunden sein müssen und dieser für einige TIANE-Features wie Wettervorhersagen oder generell bei Verwendung der Google-Spracherkennung zusätzlich eine Internetverbindung benötigt. Gut zu wissen: Obwohl wir das selbstverständlich absolut nicht empfehlen würden, kann es grundsätzlich als sicher betrachtet werden, TIANE an einem unverschlüsselten, öffentlichen Netzwerk oder zumindest einem Netzwerk, zu dem auch Gäste uneingeschränkter Zugang haben, zu betreiben: TIANE überträgt standardmäßig sämtliche Daten von Steuersignalen und Sprachkommandos bis hin zu Kamerabildern nur sicher AES-256-verschlüsselt über das Netzwerk, wobei der Schlüssel zufällig generiert und nie selbst übertragen wird.

## **Installation**

Wenn du deine zukünftigen TIANE-Geräte und die Zusatzhardware (Mikrofone etc.) angeschlossen und eingerichtet hast, folge einfach dieser Anleitung, um die von TIANE benötigten Programme zu installieren. Bitte beachte: TIANE funktioniert nur mit Python 3.x, achte daher darauf, sämtliche Pakete in einer Python 3-Umgebung zu installieren.

## Installationsanleitung für TIANE-Raumclients (Raspberry Pi unter Raspbian Stretch):

1. Nur, wenn in diesem Raum Kameras verwendet werden sollen:
  - I. Folge der Anleitung unter <https://www.pyimagesearch.com/2018/09/26/install-opencv-4-on-your-raspberry-pi/>, um OpenCV zu installieren.
  - II. `pip install imutils`
2. Installiere das von TIANE benötigte Audio-Backend:  
`sudo apt-get install portaudio19-dev sox flac`  
`pip install pyaudio`
3. Stecke sämtliche Kameras, Mikrofone und Lautsprecher ein und folge den Schritten im Abschnitt „Set up Audio“ unter <https://snowboy.kitt.ai/docs>, um die Audio-Einrichtung abzuschließen.
4. Installiere TIANEs Standard-Spracherkennung:  
`pip install SpeechRecognition`
5. Installiere TIANEs Standard-Sprachausgabe:  
`sudo apt-get install libttspic0 libtts pico-utils`
6. Gehe auf <https://snowboy.kitt.ai/dashboard>, um das Hotword (Aktivierungswort) für deinen neuen Sprachassistenten zu trainieren, auf das er hören soll (z.B. „Hey TIANE“). Trainiere am besten direkt mit dem richtigen Mikrofon an seinem finalen Platz, um gute Erkennungsquoten zu erreichen. Nach dreimaligem Sprechen deines gewünschten Aktivierungswortes erhältst du von der Seite eine Datei mit der Endung „.pmdl“ oder „.umdl“, die du nach dem Nutzer umbenennen musst, der gerade trainiert hat. Beachte, dass du für jeden Nutzer, auf den TIANE hören soll, das Aktivierungswort eigens trainieren musst, auch wenn es sich um das selbe Wort handelt, da die trainierten Dateien stark auf die Stimme des jeweiligen Trainers geprägt sind, was allerdings den Vorteil hat, dass TIANE bei korrekter Benennung der Dateien später in der Lage sein wird, ihre Nutzer anhand deren Stimme zu unterscheiden.

## Installationsanleitung für den TIANE-Server (Ubuntu 18.04):

Für den TIANE-Server gibt es zwei verschiedene Ausbaustufen. Traditionell installierst du wie in „Variante A“ beschrieben alle nötigen Pakete und legst verschiedene Konfigurationsdateien an.

Um dir bei Installation und Betrieb des TIANE-Servers die Arbeit zu erleichtern, gibt es optional ein Webserver-Modul. So kannst du den TIANE-Server komfortabel von jedem internetfähigen Gerät in deinem Heimnetzwerk aus bedienen und einrichten. Beachte dafür einfach die Einzelnen Schritte im Abschnitt „Variante B“!

Und für die etwas Erfahrenen ist es auch möglich den TIANE-Server in einem Docker-Container zu betreiben. Wie genau dies funktioniert ist in der ReadMe-Datei im server-Verzeichnis beschrieben.

### ***Variante A - konventionell***

1. Nur, wenn in dieser TIANE-Installation Kameras verwendet werden sollen:
  - I. Folge der Anleitung unter <https://www.pyimagesearch.com/2018/05/28/ubuntu-18-04-how-to-install-opencv/>, um OpenCV zu installieren.
  - II. 

```
pip install imutils
```
2. Nur, wenn zusätzlich Gesichtserkennung verwendet werden soll:
  - I. Wenn dein TIANE-Server über eine CUDA-fähige Nvidia-Grafikkarte verfügt: Folge der Anleitung unter <https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html>, um CUDA zu installieren.
  - II. Folge der Anleitung unter <https://www.pyimagesearch.com/2018/01/22/install-dlib-easy-complete-guide/>, um dlib zu installieren.
  - III. 

```
pip install face_recognition scikit-learn
```
3. Nur, wenn Telegram verwendet werden soll:
  - I. 

```
pip install telepot
```
  - II. Installiere das von TIANE benötigte Audio-Backend (Damit TIANE Sprachnachrichten senden und empfangen kann):

```
sudo apt-get install portaudio19-dev sox flac ffmpeg  
pip install pyaudio
```
  - III. Installiere TIANEs Standard-Spracherkennung (Mikrofon/Lautsprecher-Setup diesmal nicht benötigt):

```
pip install SpeechRecognition
```
  - IV. Installiere TIANEs Standard-Sprachausgabe:

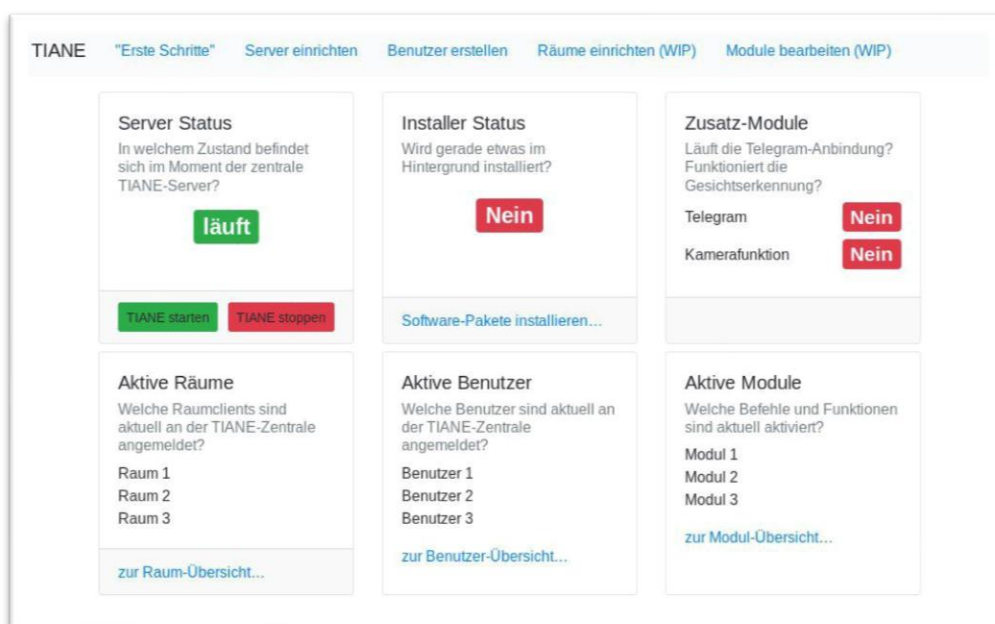
```
sudo apt-get install libttspic0 libttspic0-utils
```
4. Von Modulen benötigte Bibliotheken installieren:

```
pip install wikipedia goooglemaps spacex_py
```

Wenn weder Kameras noch Telegram verwendet werden sollen, sind für den TIANE-Server keine weiteren Schritte erforderlich.

### **Variante B – mit Webserver**

1. Alle nötigen Module vorinstallieren:
  1. `pip install flask gevent werkzeug`
2. In das Verzeichnis „webserver“ wechseln und dort „sudo python3 server.py“ ausführen. Diese erweiterten Rechte benötigt die Anwendung nur in der Installationsphase, später kannst du TIANE auch ohne „sudo“ starten.
3. Nun öffnest du im Webbrowser des Servers die Webseite <http://localhost:50500> oder von einem anderen Computer im Heimnetzwerk aus <http://<hostnameDesServers>:50500/>.
4. Es öffnet sich eine Willkommenseite. Die Texte auf den darauf folgenden Seiten werden dich durch den Prozess leiten.
5. Bitte achte darauf, dass zu zuerst den Server, und dann erst Benutzer oder Räume einrichtest.
6. Wenn alles geklappt hat, begrüßt dich TIANE nun mit folgendem Startbildschirm. Glückwunsch!



## **TIANE-Setup**

Wenn du alle Geräte aufgesetzt und die benötigte Software installiert hast, musst du noch TIANE selbst herunterladen und konfigurieren. Dafür lädst du zunächst sämtliche Dateien und Ordner aus unserem TIANE-GetHub-Repository <https://github.com/FerdiKr/TIANE> herunter, optimalerweise auf deinen späteren TIANE-Server. Achtung: Speichere dabei alle Dateien im selben Ordner, so, wie sie auch auf GitHub stehen, da die folgenden Schritte sonst fehlschlagen werden.

Jetzt kannst du deinen neuen Sprachassistenten einrichten: Dafür startest du einfach der Reihe nach die Setup-Assistenten, die dich durch die Konfiguration führen. Zunächst legst du mit `TIANE_setup_server.py` einen Namen für deinen Sprachassistenten fest und richtest den Server ein. Dann führst du für jeden Raum, den du einrichten möchtest, einmal `TIANE_setup_room.py` aus und kopierst die von diesem Assistenten generierten Ordner auf die entsprechenden Geräte. Durch wiederholtes Ausführen von `TIANE_setup_user.py` kannst du zudem beliebig viele Nutzerkonten zu TIANE hinzufügen.

Wenn diese Einrichtung abgeschlossen ist, ist dein Sprachassistent bereit zum Start: Starte einfach zuerst auf dem Server `TIANE_server.py`, dann in beliebiger Reihenfolge auf den Raum-RasPis `TIANE_room.py`. Herzlichen Glückwunsch, du verwendest ab jetzt einen quelloffenen Sprachassistenten!

## **Wie geht es weiter?**

TIANE ist noch ein sehr junges System, dessen große Stärke aber unter anderem in seiner umfangreichen Modulschnittstelle liegt. Wenn du dir also ein paar einfache, eigene Zeilen in Python zutraust, kannst du damit selbst dazu beitragen, TIANEs Funktionsumfang für alle zu erweitern. Wirf dazu am besten einfach mal einen Blick in unseren Guide zur Modulentwicklung (<https://github.com/FerdiKr/TIANE/blob/master/TIANE%20-%20Guide%20zur%20Modulentwicklung.pdf>) und trau dich ruhig, deine Ergebnisse auch über GitHub mit uns und anderen zu teilen! Und wenn du noch tiefer in die Funktionsweise von TIANE einsteigen möchtest, können wir dir unsere schriftliche Ausarbeitung zum Projekt für Jugend forscht empfehlen (<https://github.com/FerdiKr/TIANE/blob/master/TIANE%20-%20Schriftliche%20Arbeit.pdf>).

Wenn du Probleme bei der Installation oder sonstige Fragen zum Projekt hast, wende dich außerdem gerne an uns unter [jufo.teamkraemer@gmail.com](mailto:jufo.teamkraemer@gmail.com)!