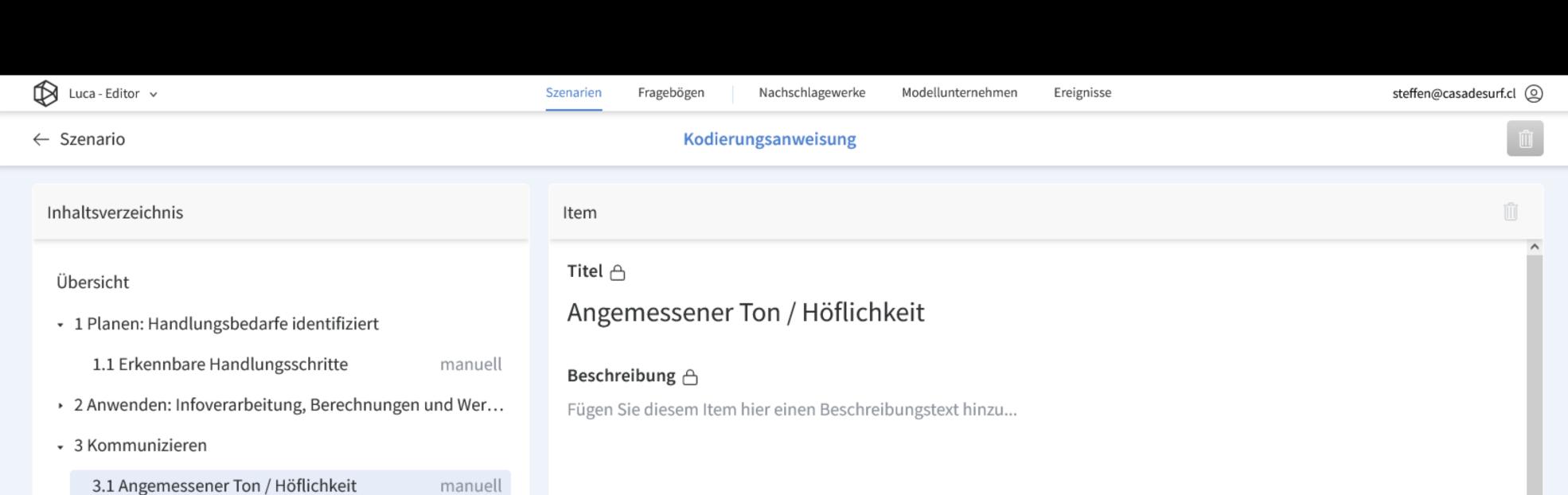
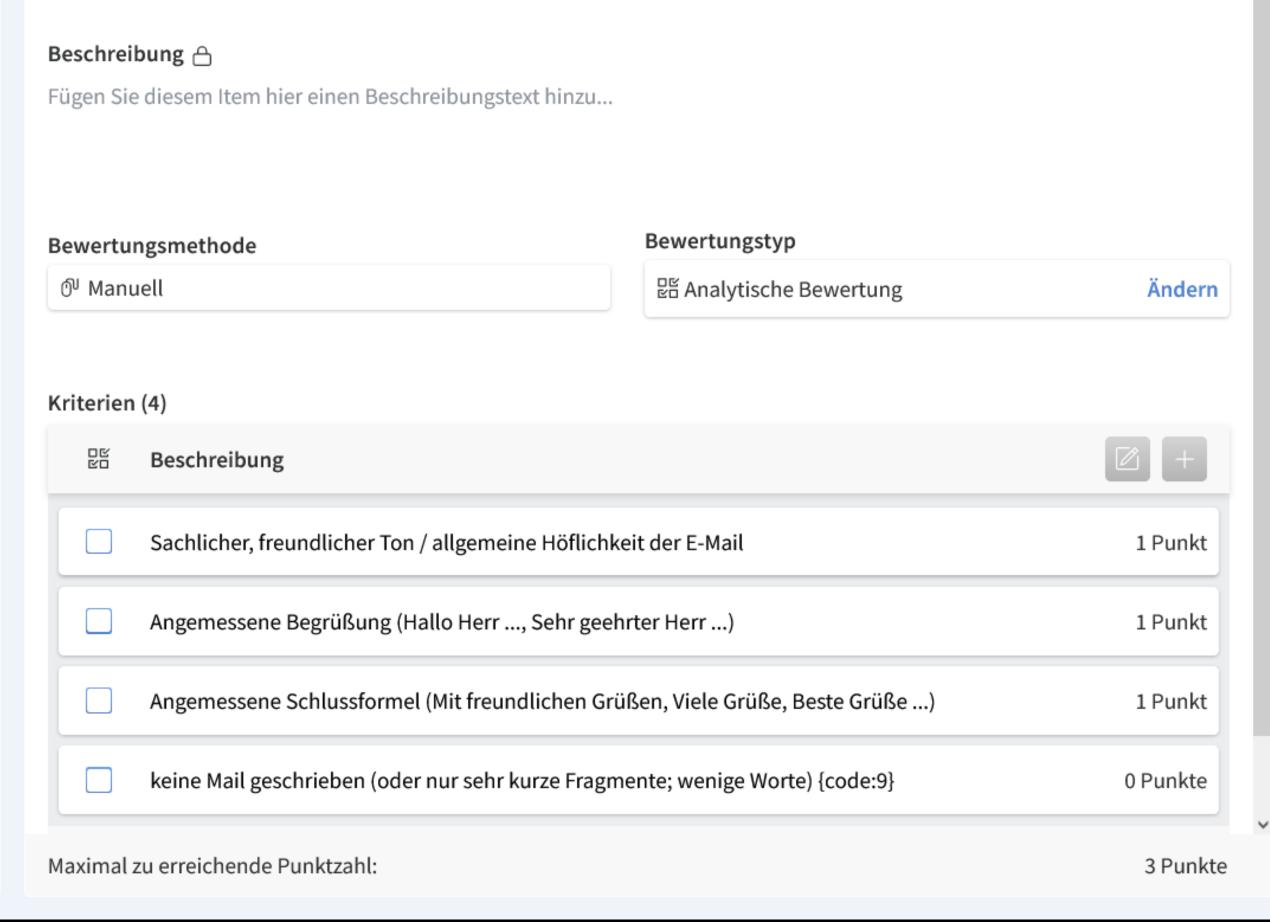


GRUNDLEGENDEIDEE

- Unterstützung des Kodierens von analytischen Bewertungen durch eine maschinelle Kodierung.
- Verknüpfung jeweils eines Bewertungskriterium mit einer Funktion zur automatisierten Auswertung.



3.2 Rechtschreib-, Grammatik- und/oder Ze... manuell





INHALTLICHE UMSETZUNG

Eingabe:

Daten einer Teilnahme zu einem Szenario
 (beinhalten die Antwort-E-Mail sowie alle aufgezeichneten Aktivitäten)

Ausgabe:

Kriterium ist erfüllt /
Kriterium ist nicht erfüllt /
Einordnung ist unklar bzw. zu unsicher



TECHNISCHE BEDINGUNGEN ZUR IMPLEMENTIERUNG DER FUNKTION

Programmiersprache:

Implementierung in R oder Python

Trennung von der eigentlichen Anwendung:

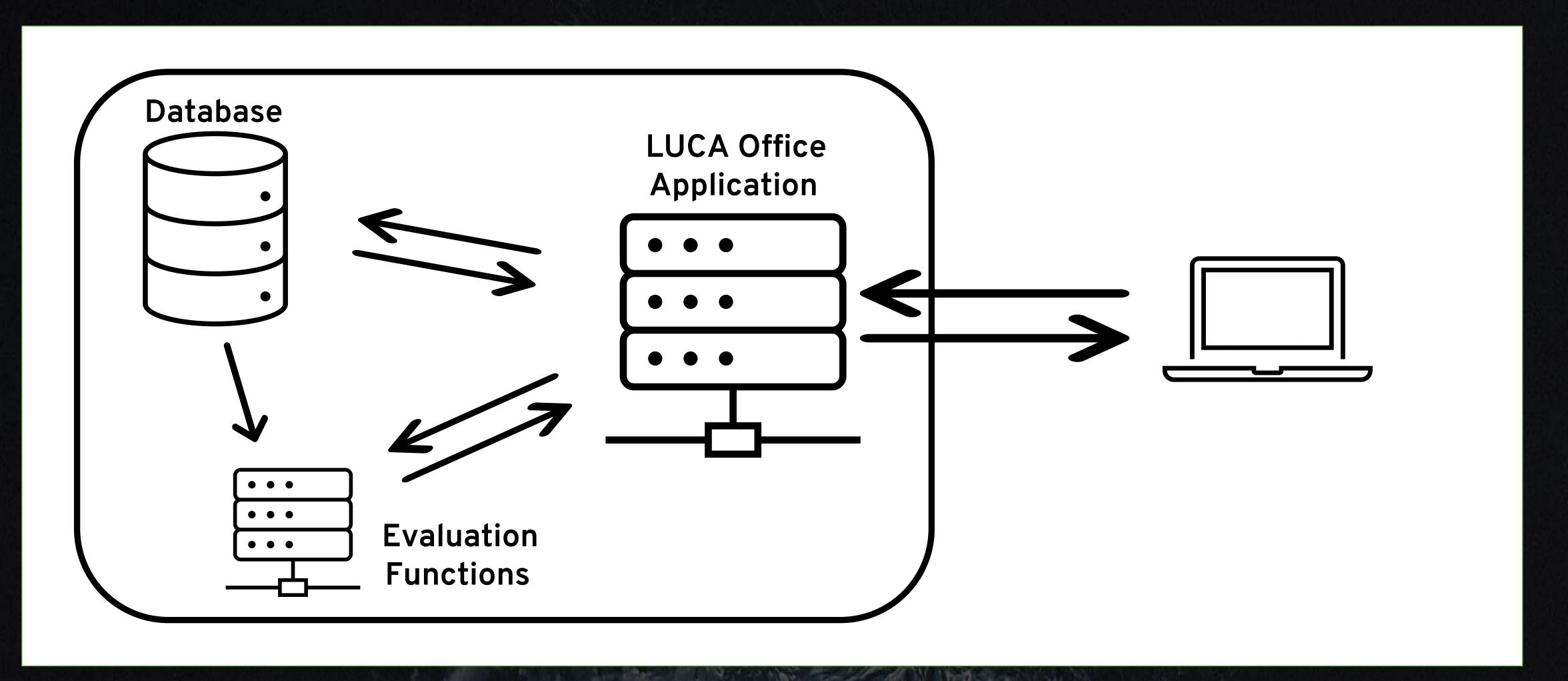
- Vereinfachung der Verwendung anderer Programmiersprachen
- Minimierung der Gefahr, die grundlegende Anwendung in ihrer Funktion zu beeinflussen
- -> Hosting der Funktionen auf einem eigenen, internen Server

Datenzugang:

- Direkter Lese-Zugriff auf die Datenbank von LUCA-Office
- Auslesen nur der jeweils benötigten Daten

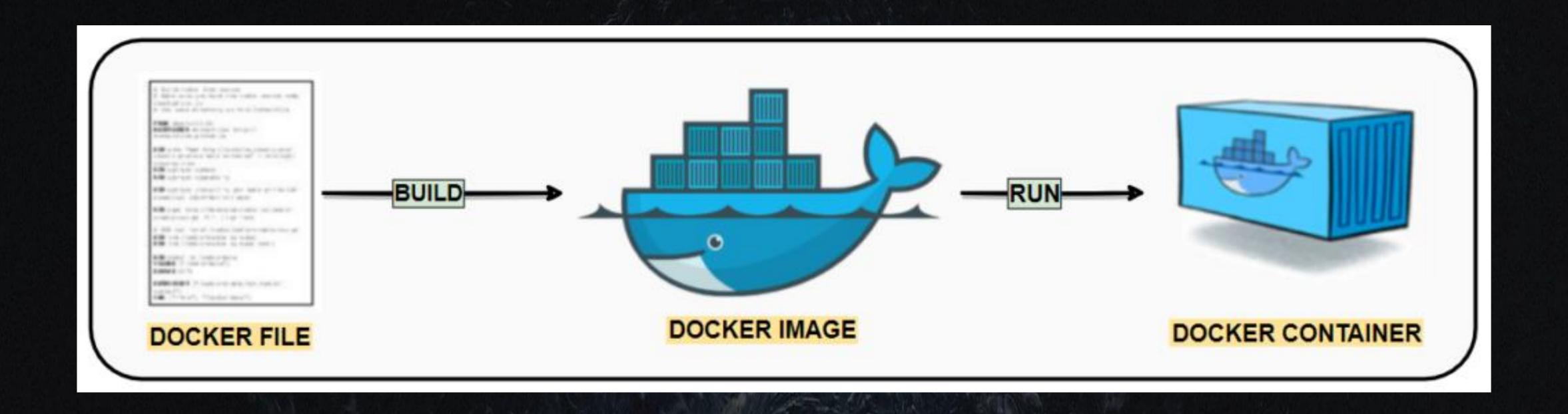


UMSETZUNGEN DER TECHNISCHEN BEDINGUNGEN



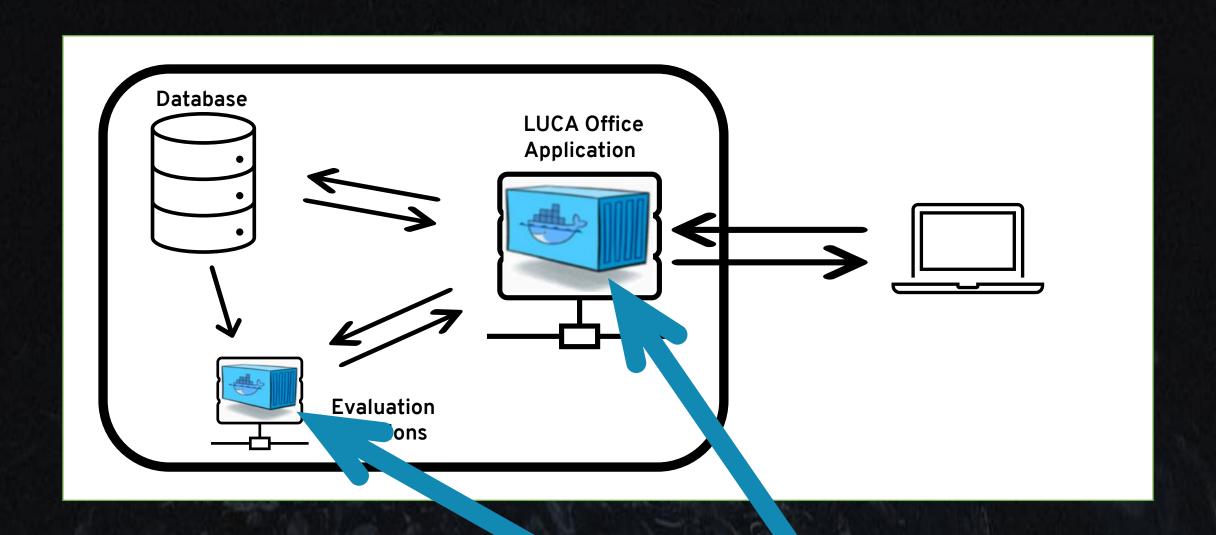


INSTALLATION VON ANWENDUNGEN MIT DOCKER





VERWALTUNG DER ANWENDUNG



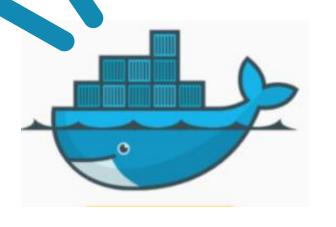
Possible Code Repositories

- GitHub
- GitLab
- Private Repository

•



- Docker file
- Application Code



- LUCA Office Image
- Evaluation API Image

Possible Image Repositories

- GitHub
- Docker Hub
- Private Repository
- •



UMSETZUNG DER INHALTLICHEN BEDINGUNGEN

```
evaluate <- function(functionId="p_001_politeness", invitationId, scenarioId, surveyId) {
    # evaluation code calling the function with the given id and
    # returning a response object in the following format:
    response = list(function name = functionId,
                    criterion_no = "0",
                    criterion_prediction = "fulfilled",
                    criterion_probability = .9879123,
                    criterion_threshold = .95,
                    data = "")
    return (response)
```



UMSETZUNG DER INHALTLICHEN BEDINGUNGEN

LUCA Office ruft die Funktion evaluate via API auf

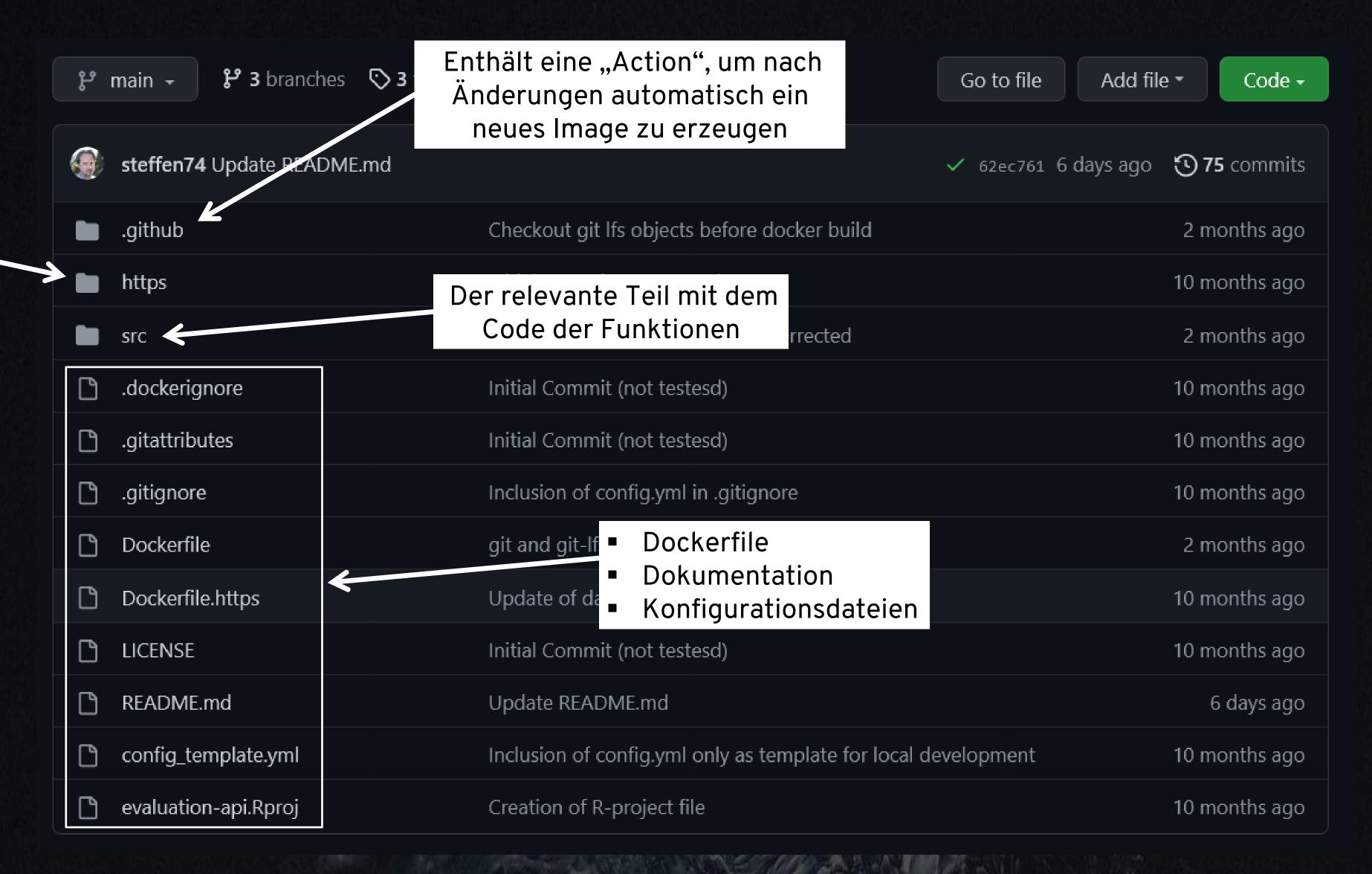
Funktion evaluate:

- Ruft die Funktion mit dem von LUCA Office übergebenen Namen auf
- In Abhängigkeit des ersten Buchstabens im Funktionsnamen wird die Funktion in Python oder R aufgerufen
- Enthält der Funktionsname _answer _ wird die Antwort-E-Mail als Argument beim Funktionsaufruf mitgegeben



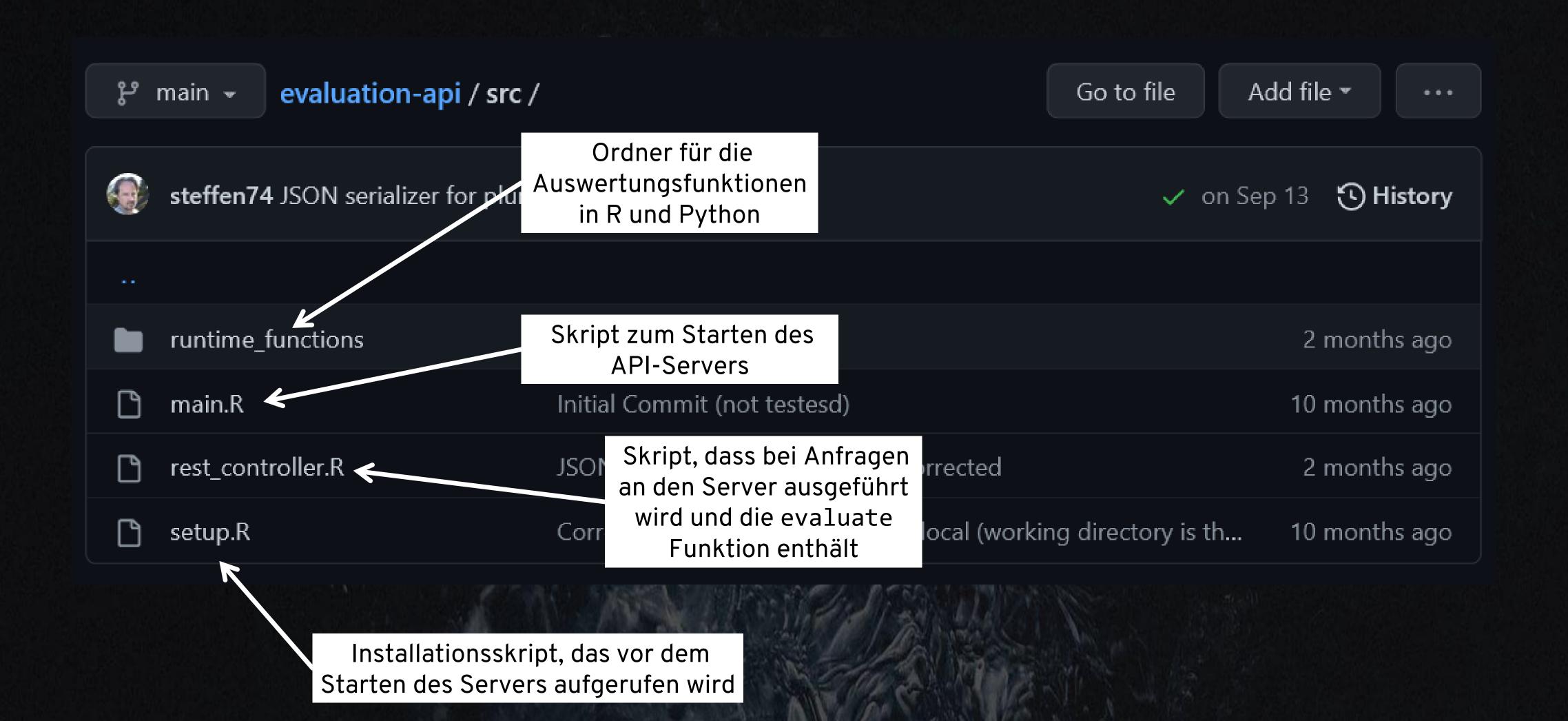
CODE REPOSITORY

Nur relevant, wenn der Server offen im Netz steht





CODE REPOSITORY



ENTWICKLUNG EINER EIGENEN FUNKTION

- 1) Anlegen eines GitHub Account
- Fork des evaluation-api Repositories unter dem eigenen GitHub Account erzeugen
- 3) Anlegen eines neuen Projektes in RStudio auf Basis des "geforkten" Repositories
- 4) Schreiben der eigenen Funktion im entsprechenden Projekt-Verzeichnis
- 5) Testen der Funktion in RStudio über Ausführen von rest_controller.r als API
- 6) Commit der Änderungen in das geforkte Repository in GitHub
- 7) Pull Request vom eigenen Repository in das Repository der evaluation API von LUCA