TECHNISCHE UNIVERSITÄT
CHEMNITZ

Fakultät für Informatik
Professur Technische Informatik

TUC DriveCloud

A Big Data Storage for Automotive Test Drive Data

René Bergelt

TUC DriveCloud

TUC DriveCloud is a combination of 3 research aspects at the professorship

Automotive Software Engineering, embedded systems

Data aggregation in sensor networks, wireless communication

Cloud computing, web technologies

Research Field

Development of ADAS and autonomous driving functions

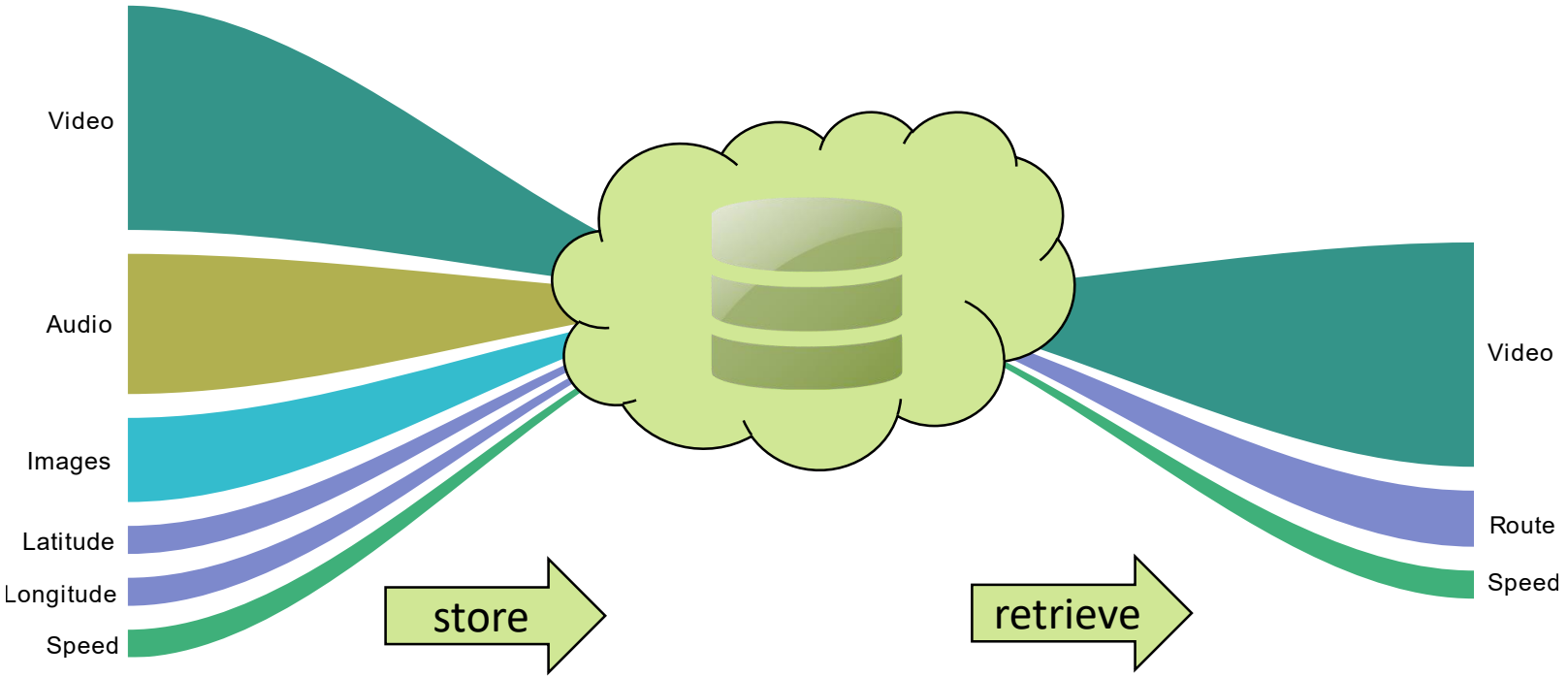
Development of complex automotive functions

- Large number of test drives to be conducted
- High volume of sensor data to be collected and stored
- Heterogenous types of data
  - Single values (e.g. GPS coordinates)
  - Images
  - Video streams
  - Audio

Challenges

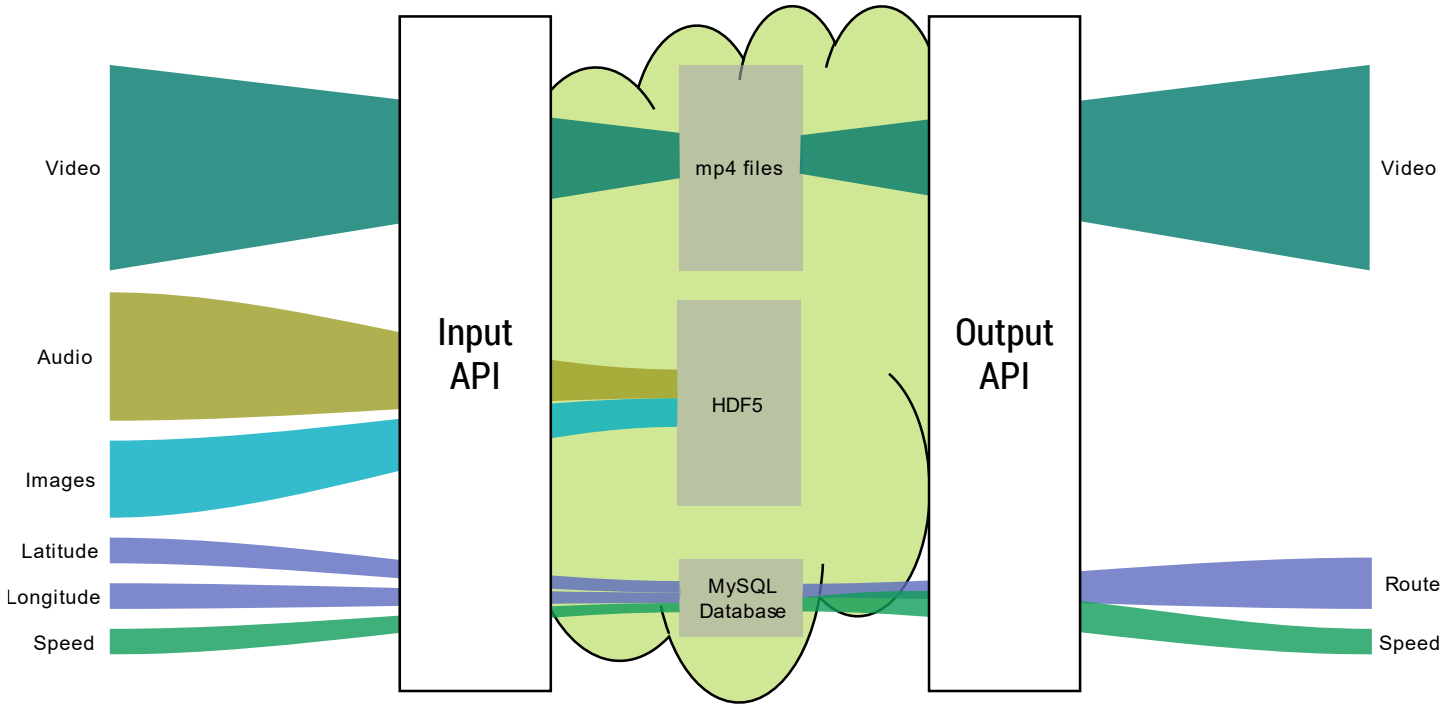
Aim of the system:

- Have a standardized way to store and retrieve test drive data through a cloud system independently of the types of sensors and vehicles
- Abstract the „heavy-lifting“ of data storage and retrieval
- Test drive data should be automatically uploaded



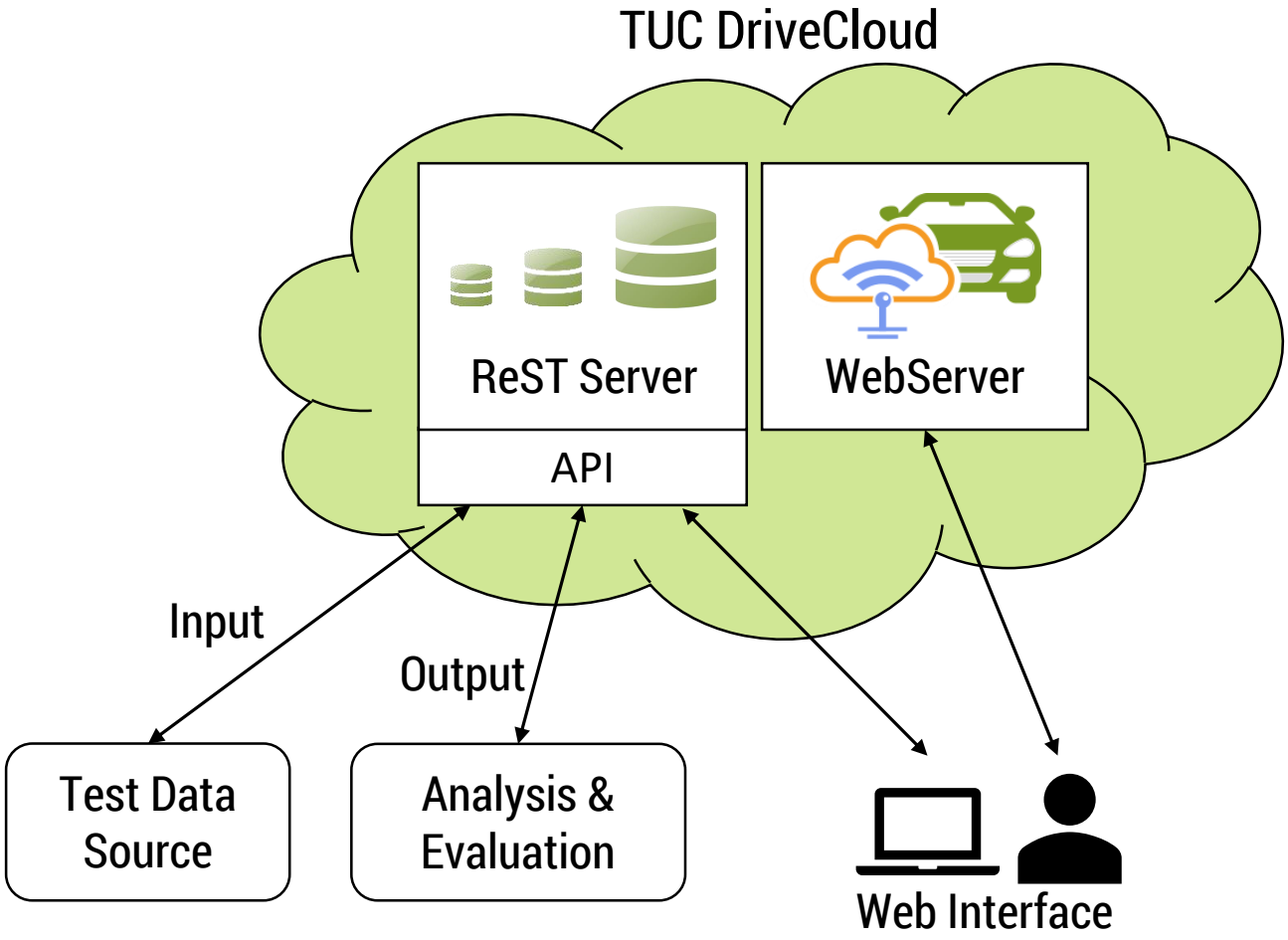
TUC DriveCloud - Storage abstraction

How data is stored is completely hidden from the user
Data can **only** be accessed through standardized APIs
Makes it easy to change/extend storage strategies since user applications cannot rely on specific structures



TUC DriveCloud – Architecture

Separation between frontend and backend

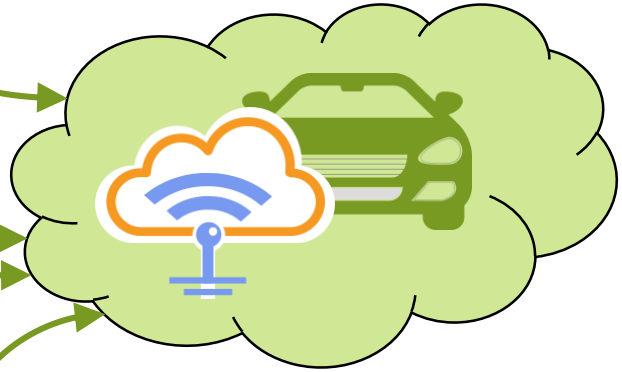


TUC DriveCloud – Test Data Sources

Test data sources access the cloud through the unified API
Arbitrary, timestamped sensor data information can be stored
TUC DriveCloud knows the concept of vehicles, sensors and test drives for management

Research Car

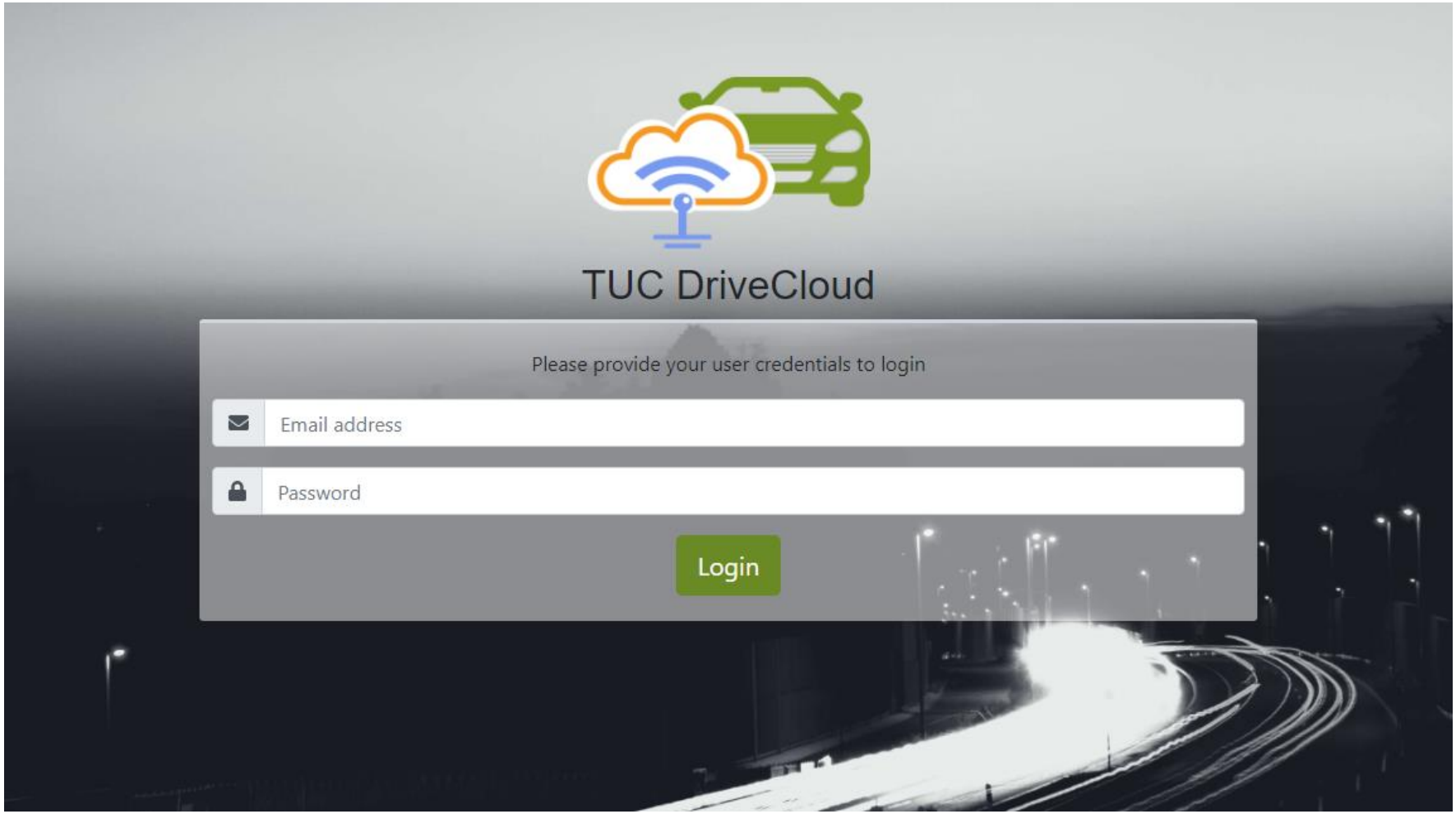
Driving Simulator



Yellow Car
Demonstrator

Test Drive Data Generator

TUC DriveCloud – WebInterface Demo



TUC DriveCloud – Documentation

Challenges:

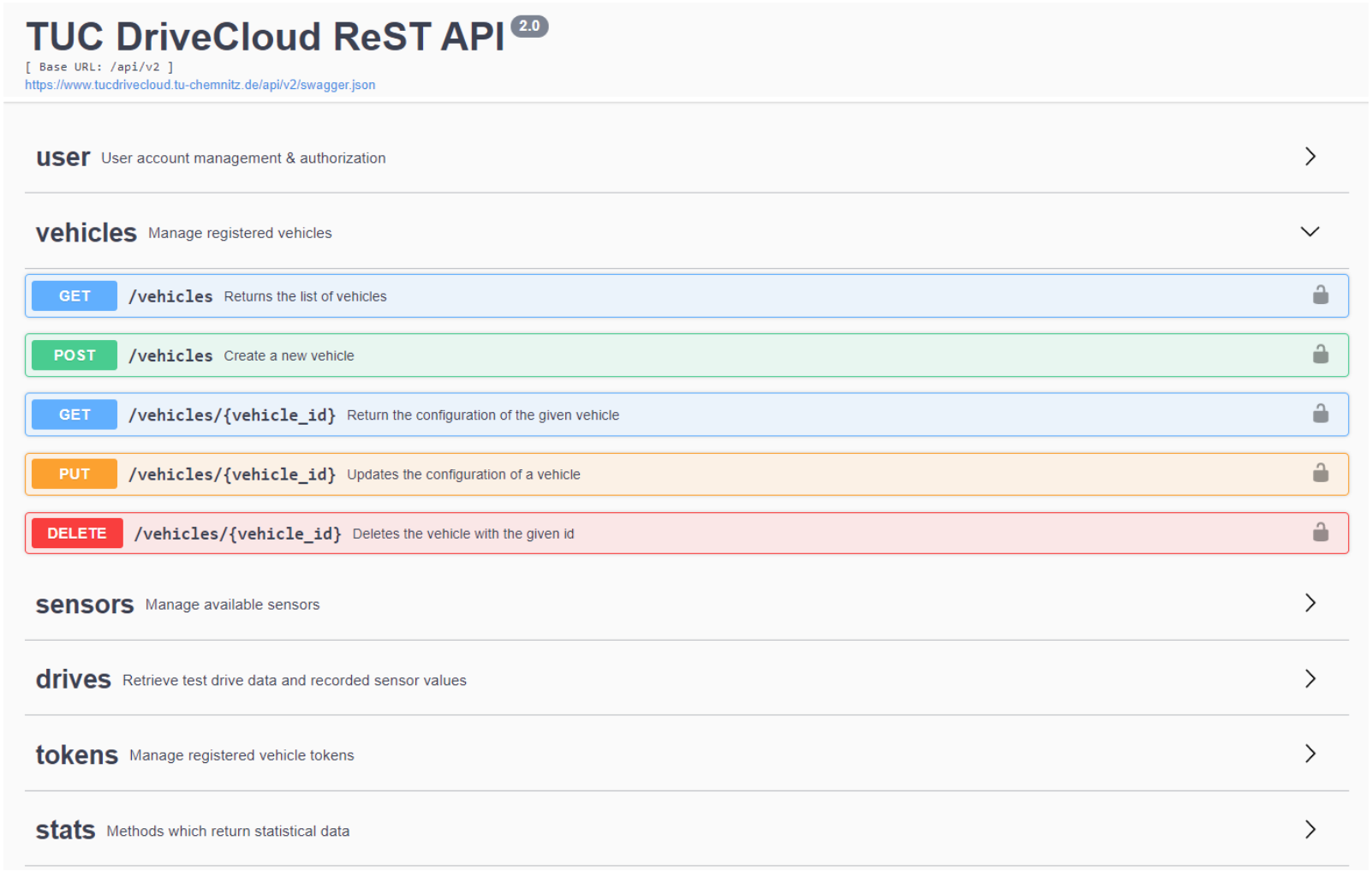
- APIs grow, documentation is build for a specific Version X, server is running Y
- Usually, API documentations are not very vivid
- Where should the documentation be stored?

Our Solution: OpenAPI and Swagger

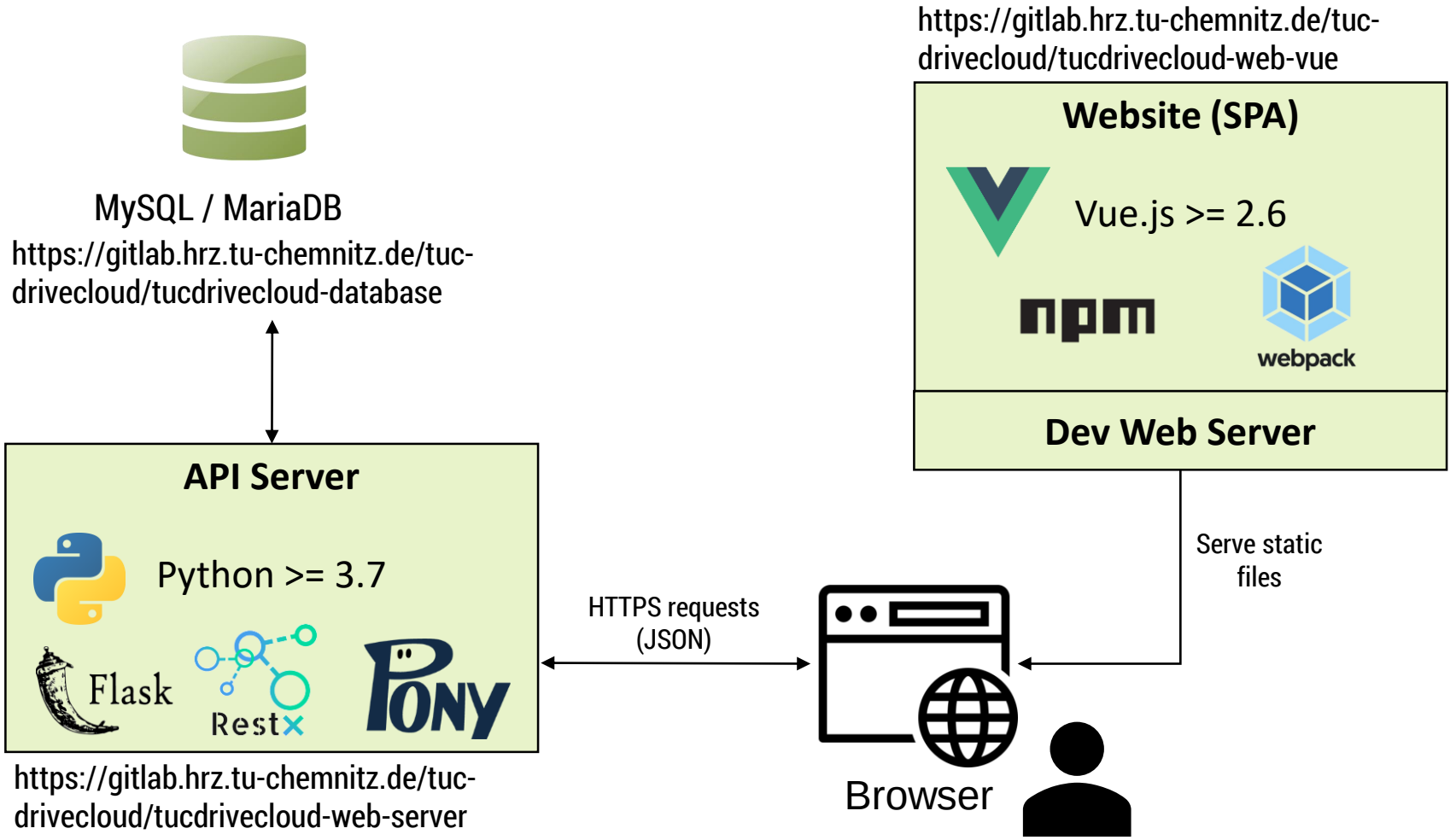
- Server which provides the API also serves the documentation
- Documentation is generated from the code which is actually running on the server
- Documentation web site is interactive, i.e. all requests can be tested in place

TUC DriveCloud Wiki:
https://gitlab.hrz.tu-chemnitz.de/tuc-drivecloud/tucdrivecloud-documentation/-/wikis/home

TUC DriveCloud – Interactive API Documentation Demo

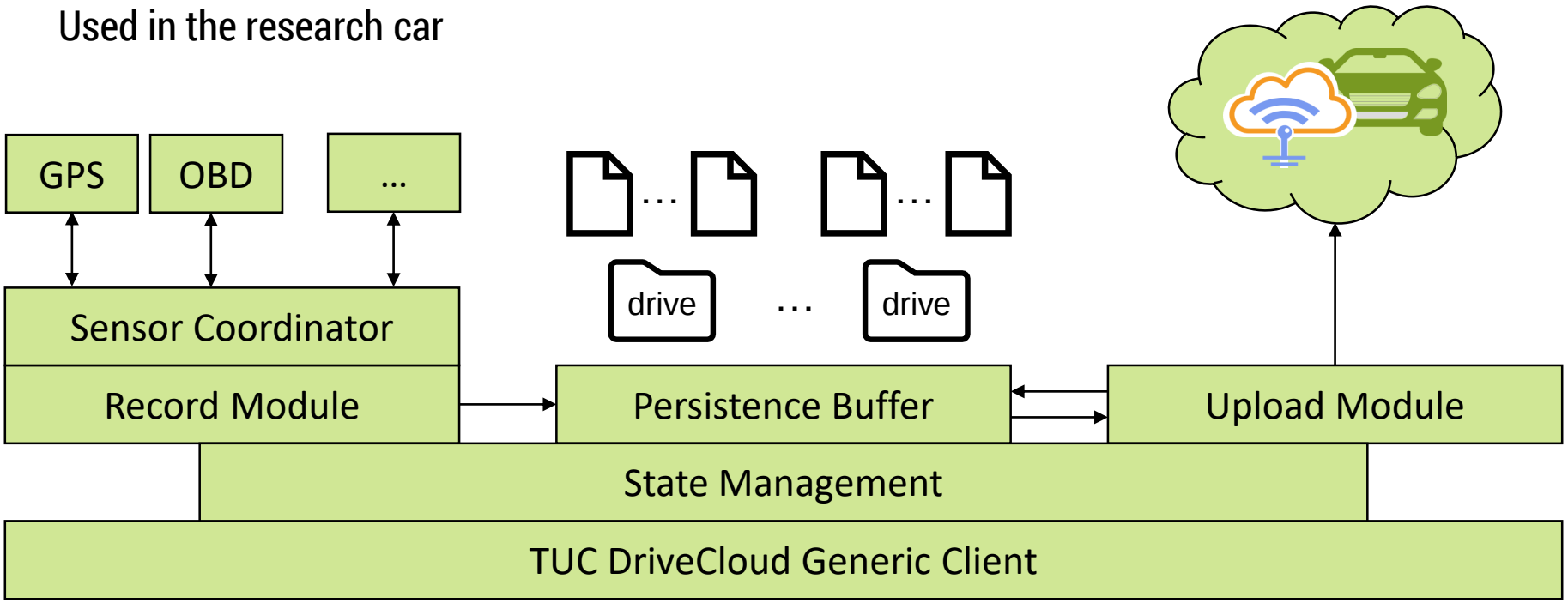


TUC DriveCloud – Local Development Instance



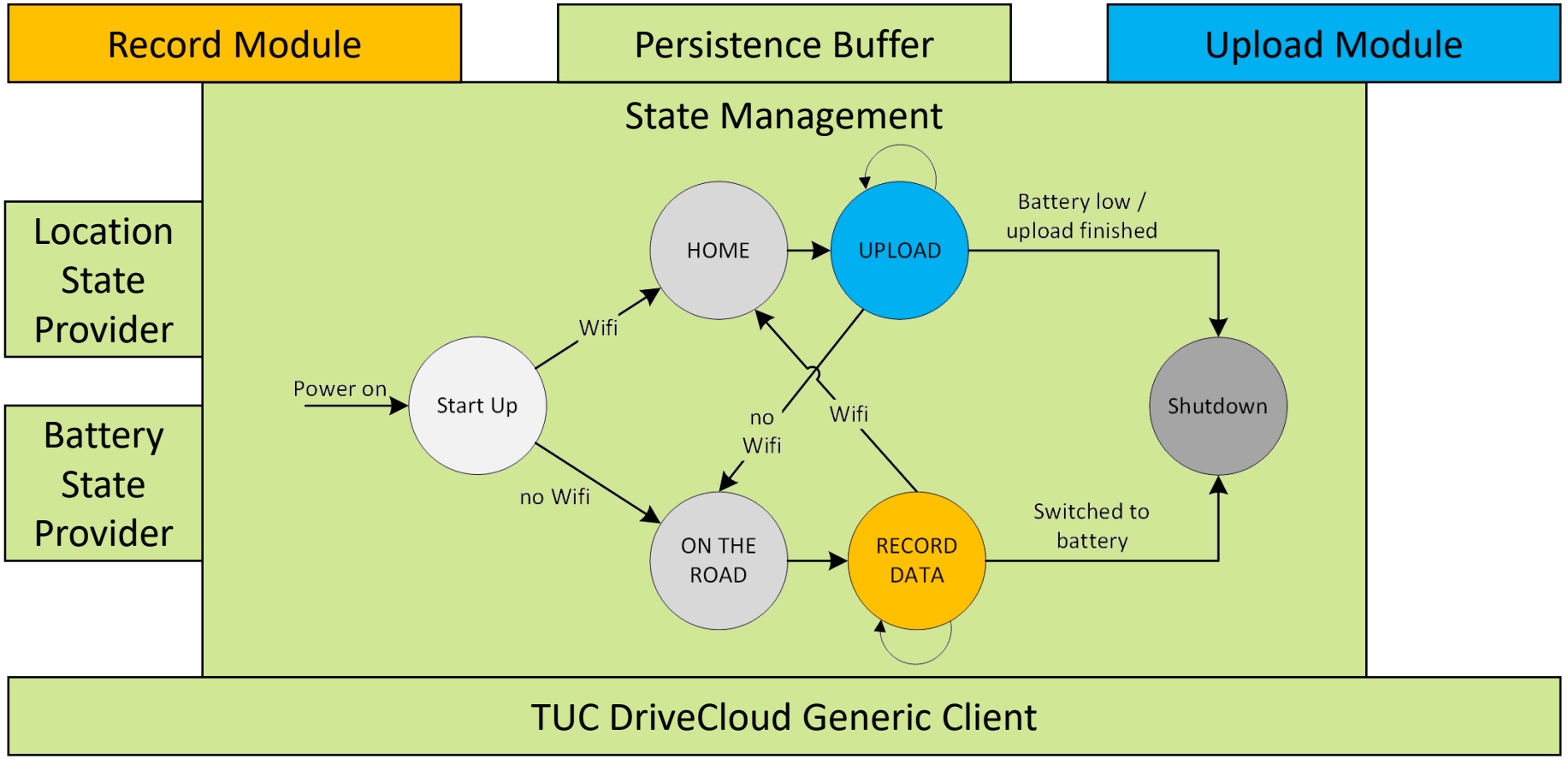
TUC DriveCloud – Generic Client

Written in Python
Base for custom client implementations
Supports local buffering and automatic upload
State management for upload and battery
Used in the research car



https://gitlab.hrz.tu-chemnitz.de/tuc-drivecloud/tucdrivecloud-generic-client

TUC DriveCloud – Generic Client



TUC DriveCloud - Summary

TUC DriveCloud is an enabling platform for:

- Research topics
- ASE developments at the professorship
- Student works and Master theses

Developers can setup a fully local instance which facilitates understanding and debugging

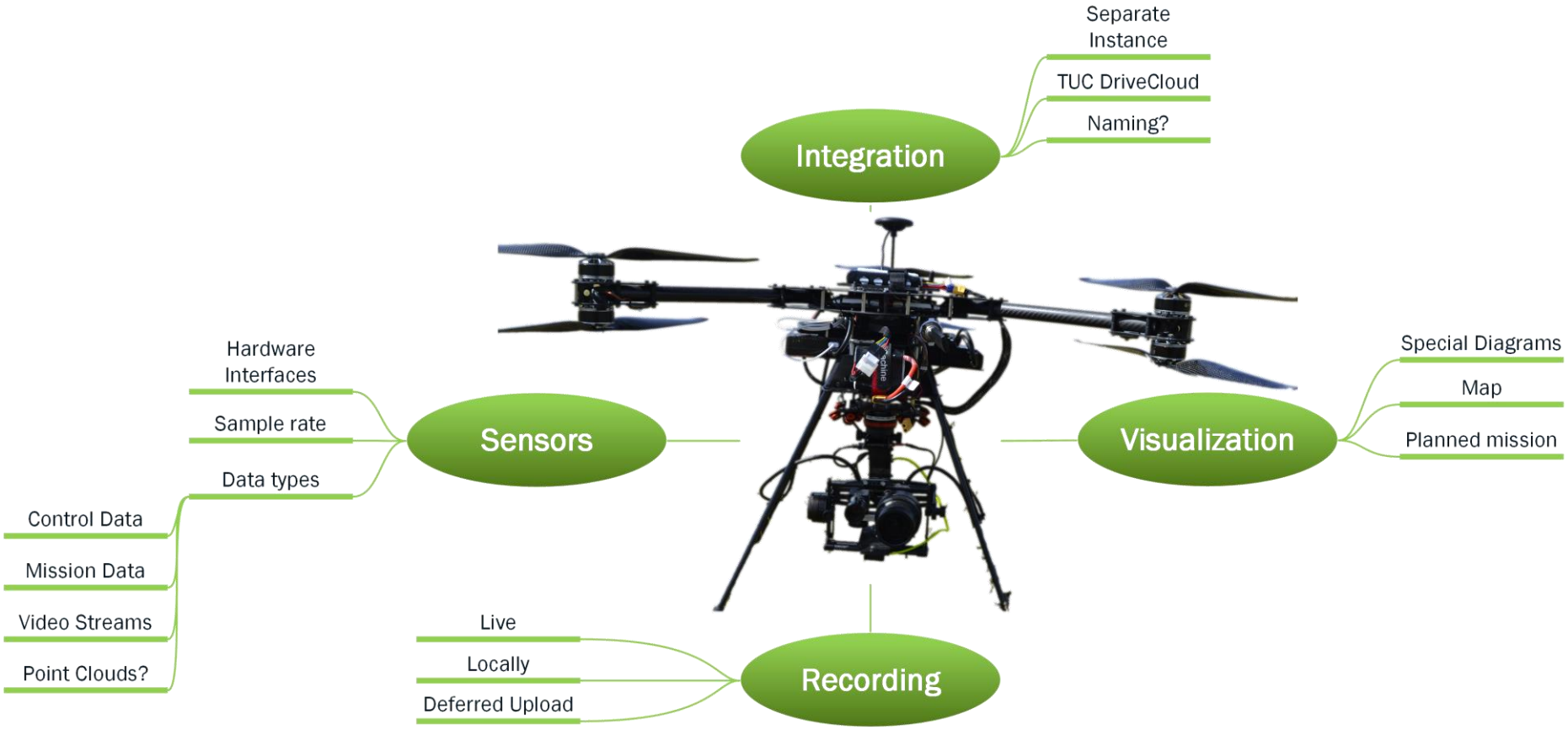
Main goal:

Building a pool of test drive data to train, test and assess automotive software developments

Not limited to automotive data

TUC DriveCloud - Discussion

Not limited to automotive data → integrate CE copter

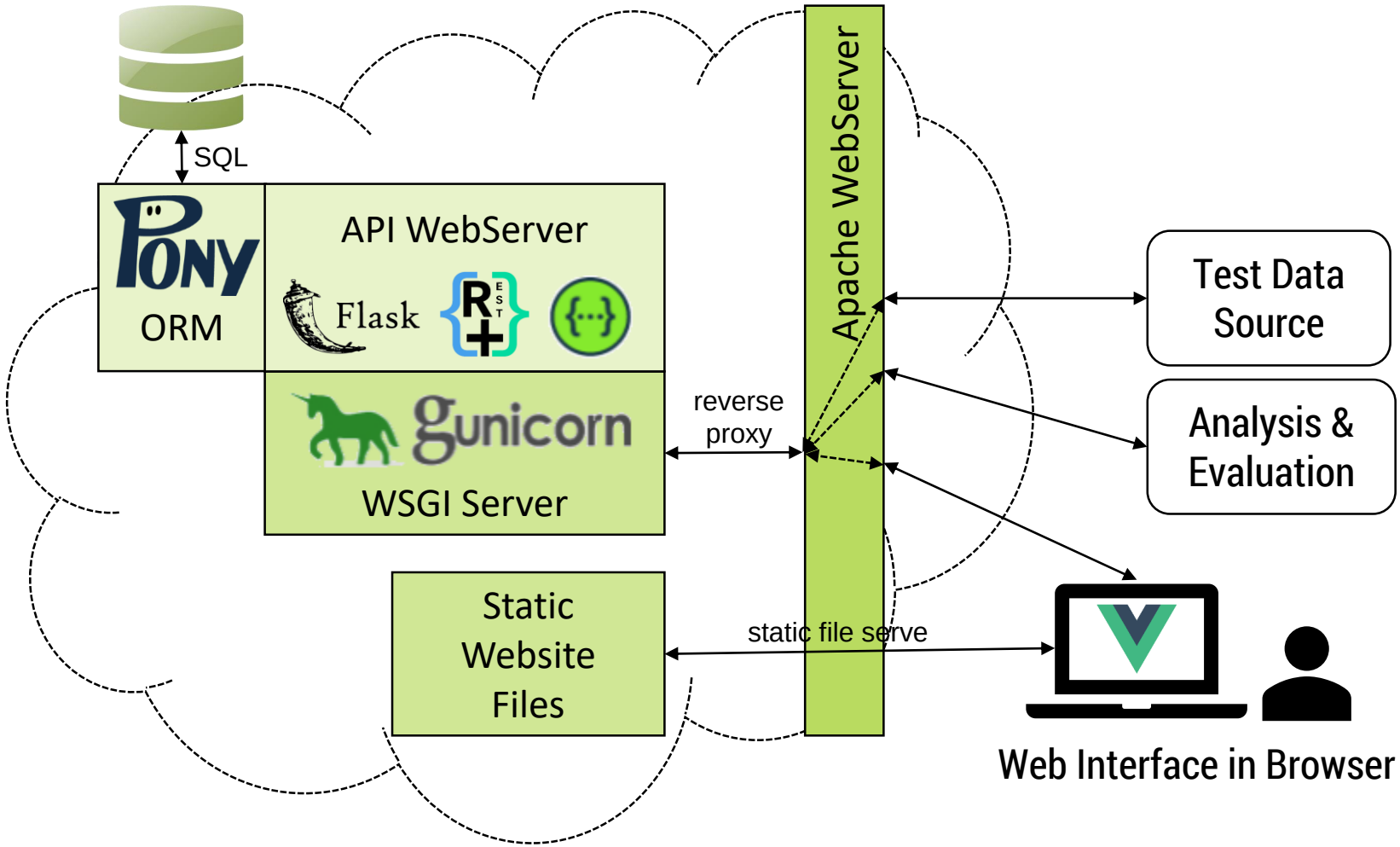


TUC DriveCloud

A Big Data Storage for Automotive Test Drive Data

René Bergelt

TUC DriveCloud –Technology Stack



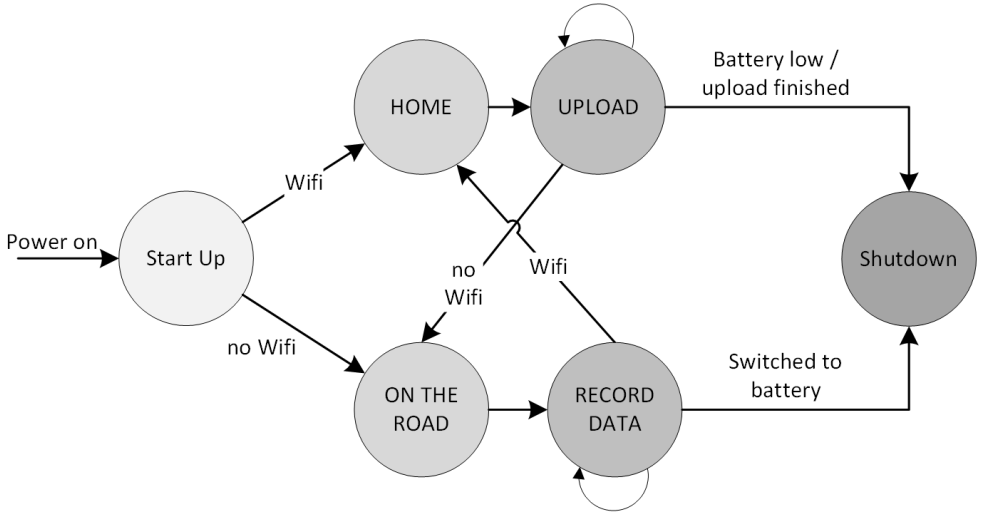
TUC DriveCloud – Research Car

Collect vehicle sensor data

- OBD, GPS sensor, Video Camera
- Driver data (e.g. heart rate)
- Lidar (in development)

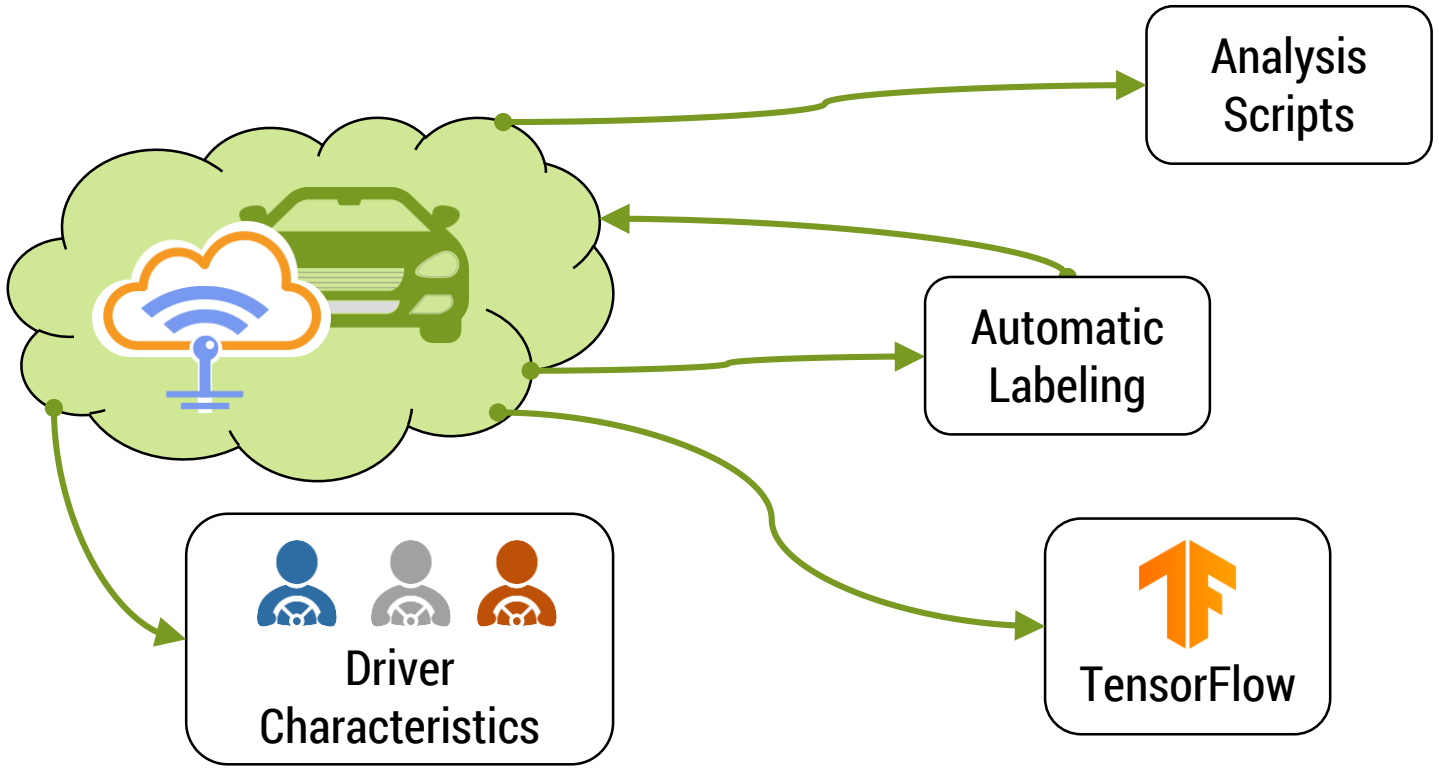
Vehicle is equipped with a „logger box“:

- Based on Raspberry Pi
- Connected to car power
- Equipped with a rechargable battery



TUC DriveCloud – Data Analyses

Test evaluation tools access TUC DriveCloud through the unified API
Arbitrary sensor data streams can be retrieved (random access is possible)



TUC DriveCloud – Analysis sample

The ReST interface allows an easy, standardized way to access data streams of stored test drives
Example: Retrieving number of total test drive kilometers

```python
base_url = 'https://www.tucdrivecloud.tu-chemnitz.de/api/v2/'
# login
r = requests.post(base_url + 'user/login', json={'username': username,'password': pwd})
token = r.json()['token']

# iterate over all test drives
r = requests.get(base_url + 'drives', headers={'Authorization': f'Bearer {token}'})
drives = r.json()
total_dist = 0

for d in drives:
    # get route points
    r = requests.get(base_url + f'drives/{d["id"]}/streams/route',
                     headers={'Authorization': f'Bearer {token}'})
    pts = r.json()['points']
    if len(pts) >= 1:
      route_dist = calc_route_len(pts)
      total_dist += route_dist

print(f'total test drive kilometers in database: {total_dist}')
```

TUC DriveCloud – Modularized SPA