Multikopter

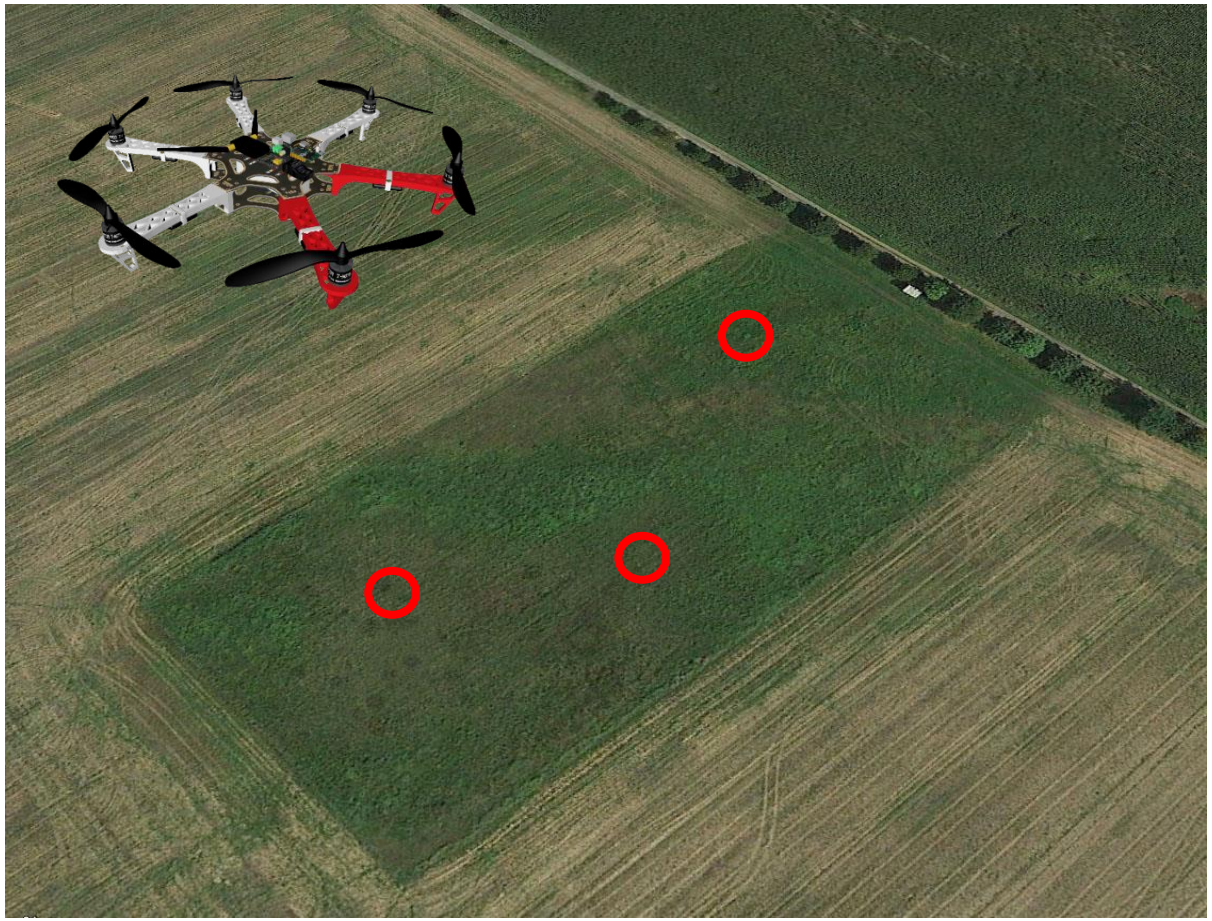
SDK & REST-API

Über mich

- Marcel Kühne, marcel.kuehne@hszg.de
- ehem. Mitglied des Quantocopter Projektes

  → Verortung von Kiebitz-Gelegen mithilfe von Multikoptern und bodengestützter Bilderkennung

Über mich



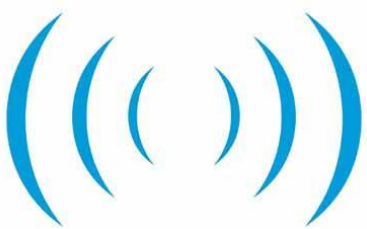
QUANTOCOPTER ● QUANTEO® Software Engineering GmbH ● Schlaurother Straße 13 ● 02827 Görlitz ● kontakt@quantocopter.de

Inhalt

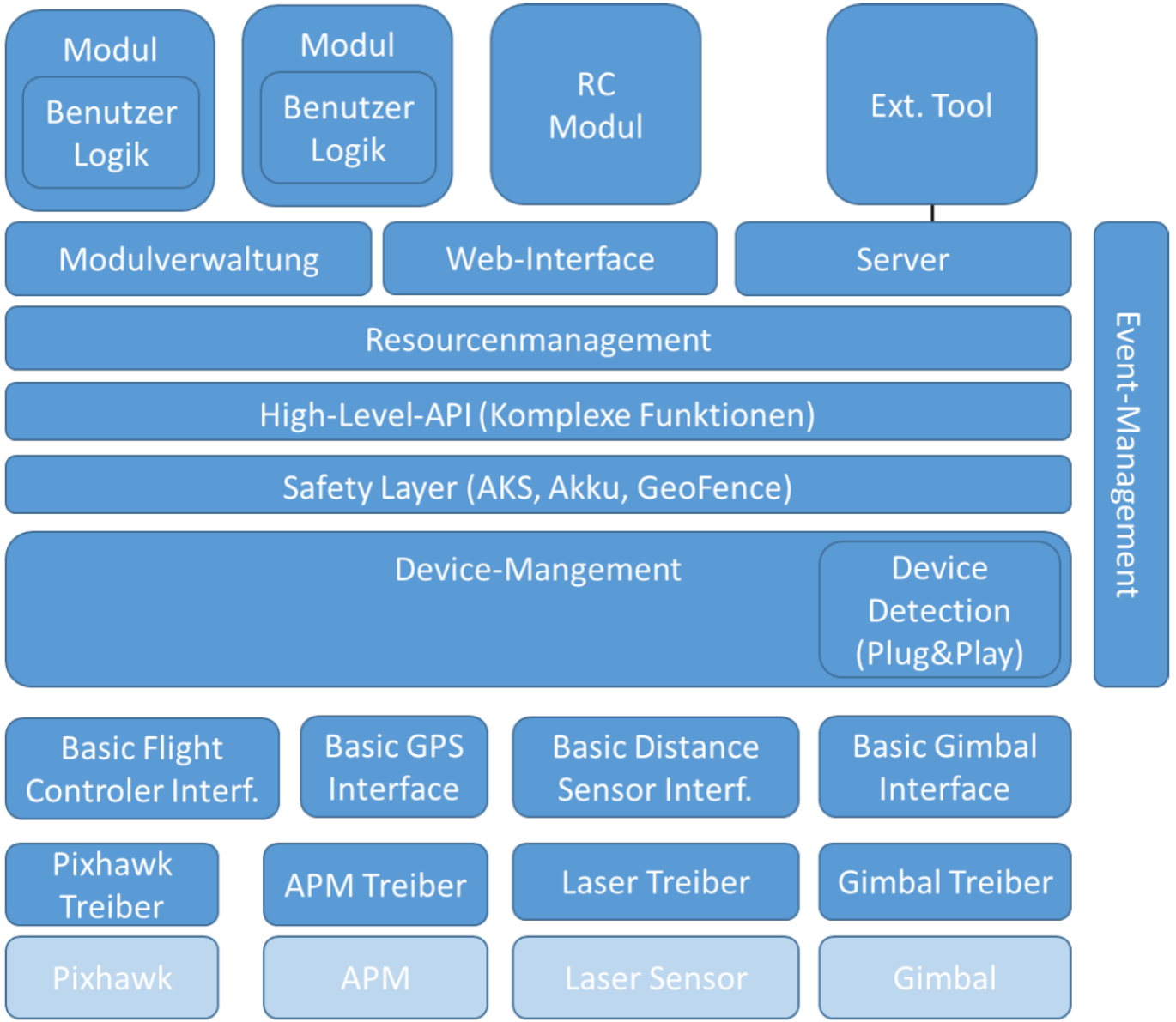
1. Anwendungsfall
2. SDK-Einführung
3. REST-API
4. Zielstellung und Teambildung
5. Übung: praktischer Einstieg in das SDK

QUANTOCOPTER ● QUANTEO® Software Engineering GmbH ● Schlaurother Straße 13 ● 02827 Görlitz ● kontakt@quantocopter.de

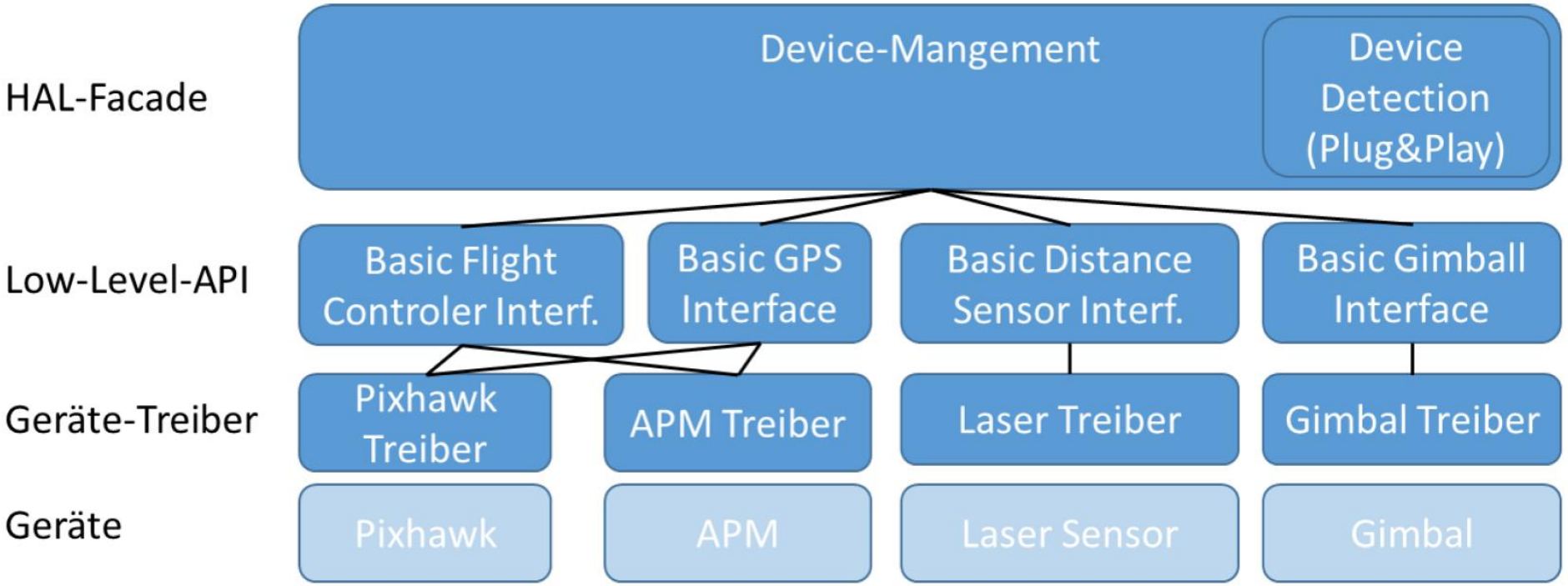
Anwendungsfall



SDK

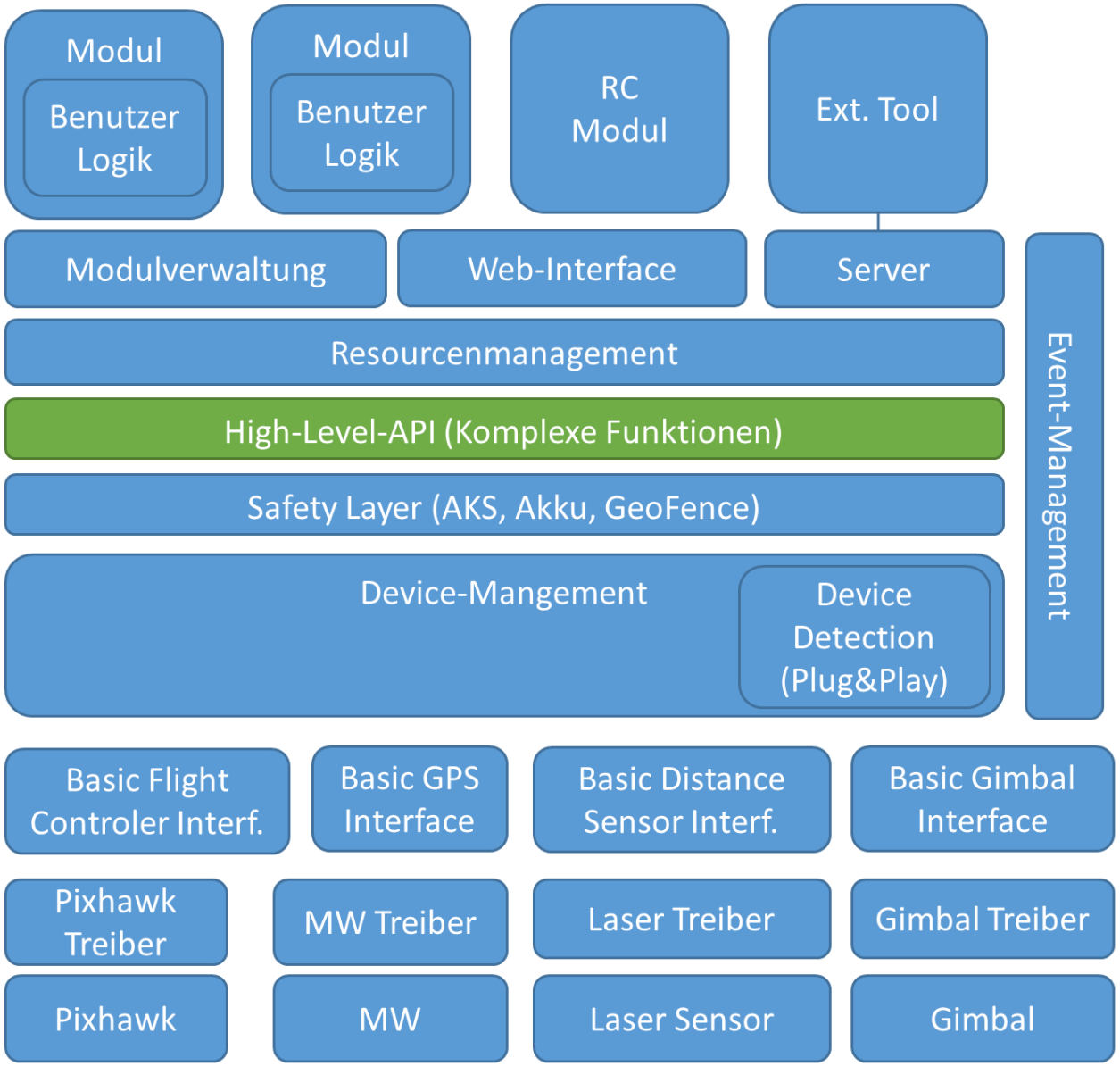

Modul
Benutzer Logik
Modul
Benutzer Logik
RC Modul
Ext. Tool
Modulverwaltung
Web-Interface
Server
Resourcenmanagement
High-Level-API (Komplexe Funktionen)
Safety Layer (AKS, Akku, GeoFence)
Device-Mangement
Device Detection (Plug&Play)
Event-Management
Basic Flight Controler Interf.
Basic GPS Interface
Basic Distance Sensor Interf.
Basic Gimbal Interface
Pixhawk Treiber
APM Treiber
Laser Treiber
Gimbal Treiber
Pixhawk
APM
Laser Sensor
Gimbal

SDK: Hardware-Abstraction-Layer



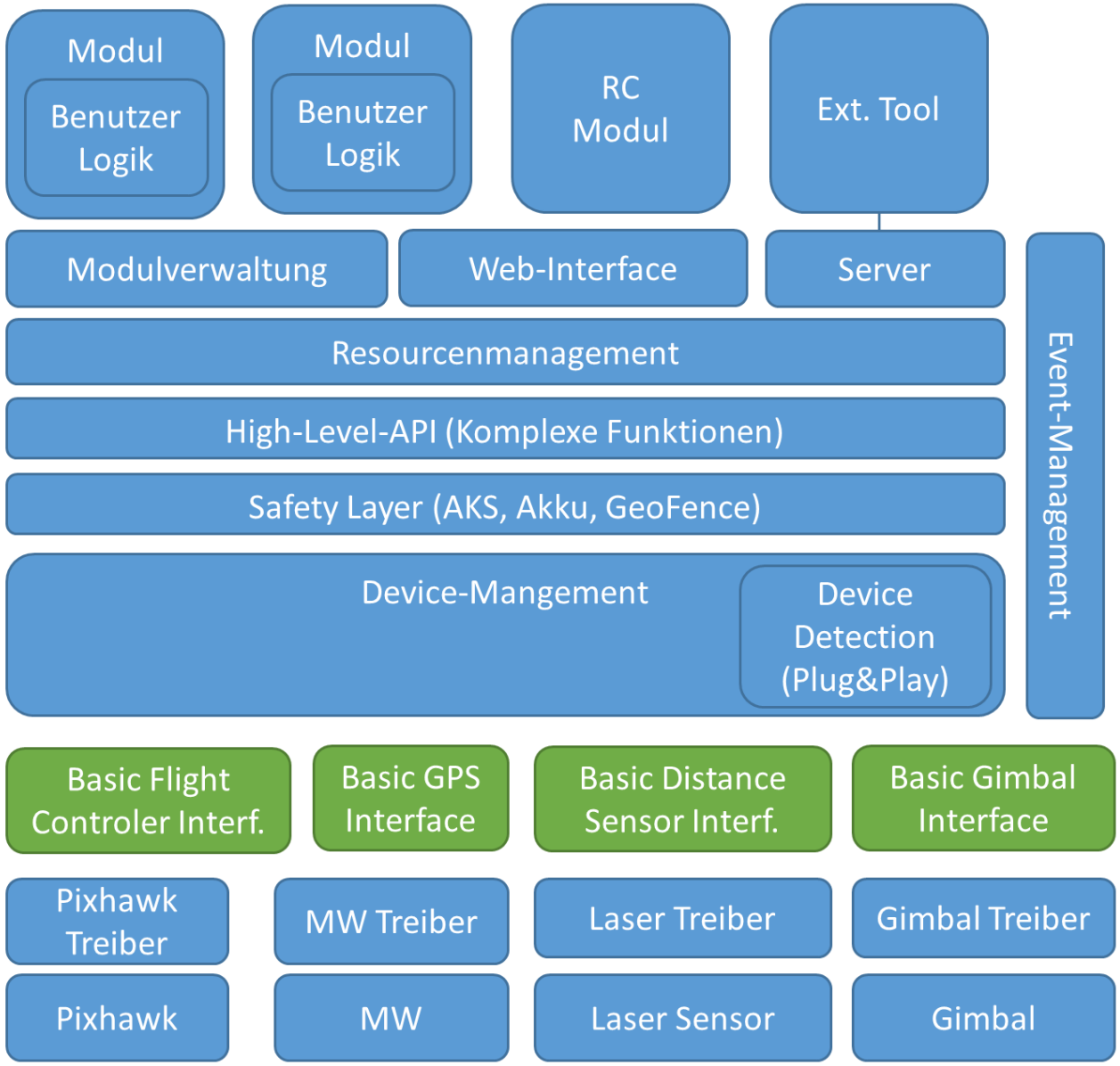
SDK

CISC


Modul
Benutzer Logik
Modul
Benutzer Logik
RC Modul
Ext. Tool
Modulverwaltung
Web-Interface
Server
Resourcenmanagement
High-Level-API (Komplexe Funktionen)
Safety Layer (AKS, Akku, GeoFence)
Device-Mangement
Device Detection (Plug&Play)
Event-Management
Basic Flight Controler Interf.
Basic GPS Interface
Basic Distance Sensor Interf.
Basic Gimbal Interface
Pixhawk Treiber
MW Treiber
Laser Treiber
Gimbal Treiber
Pixhawk
MW
Laser Sensor
Gimbal

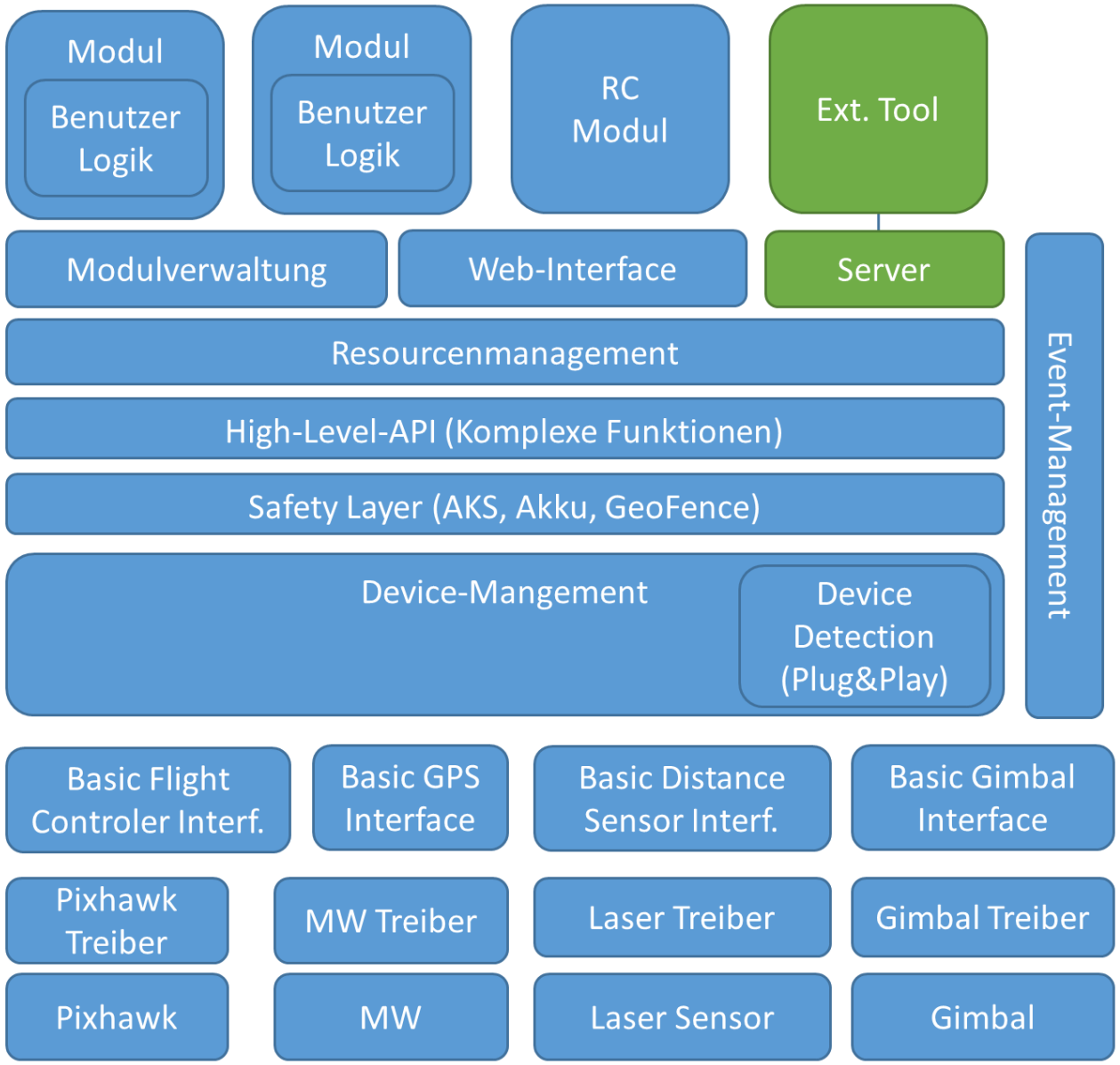
SDK

Unabhängige Programmierung



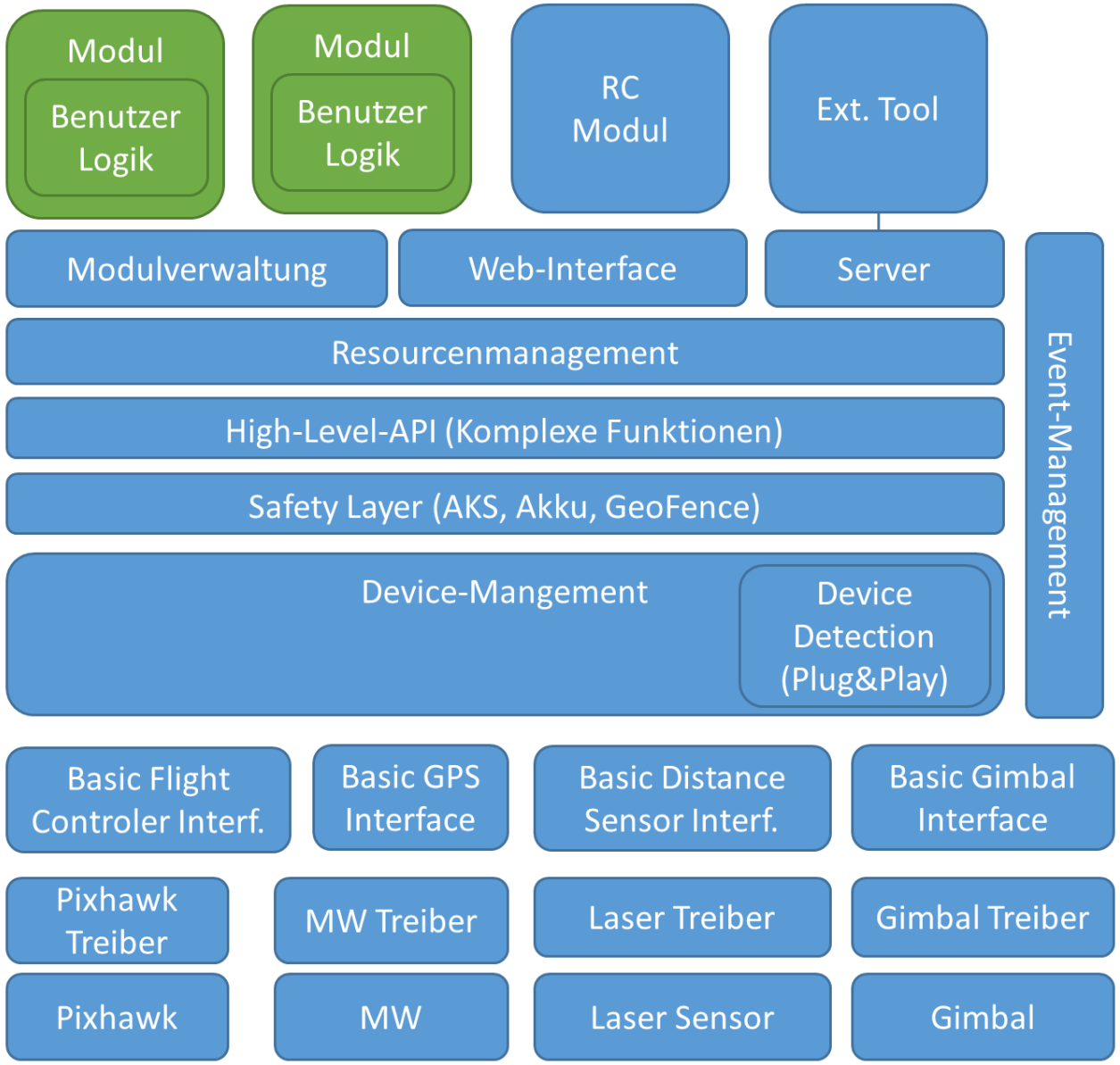
SDK

Offenes System



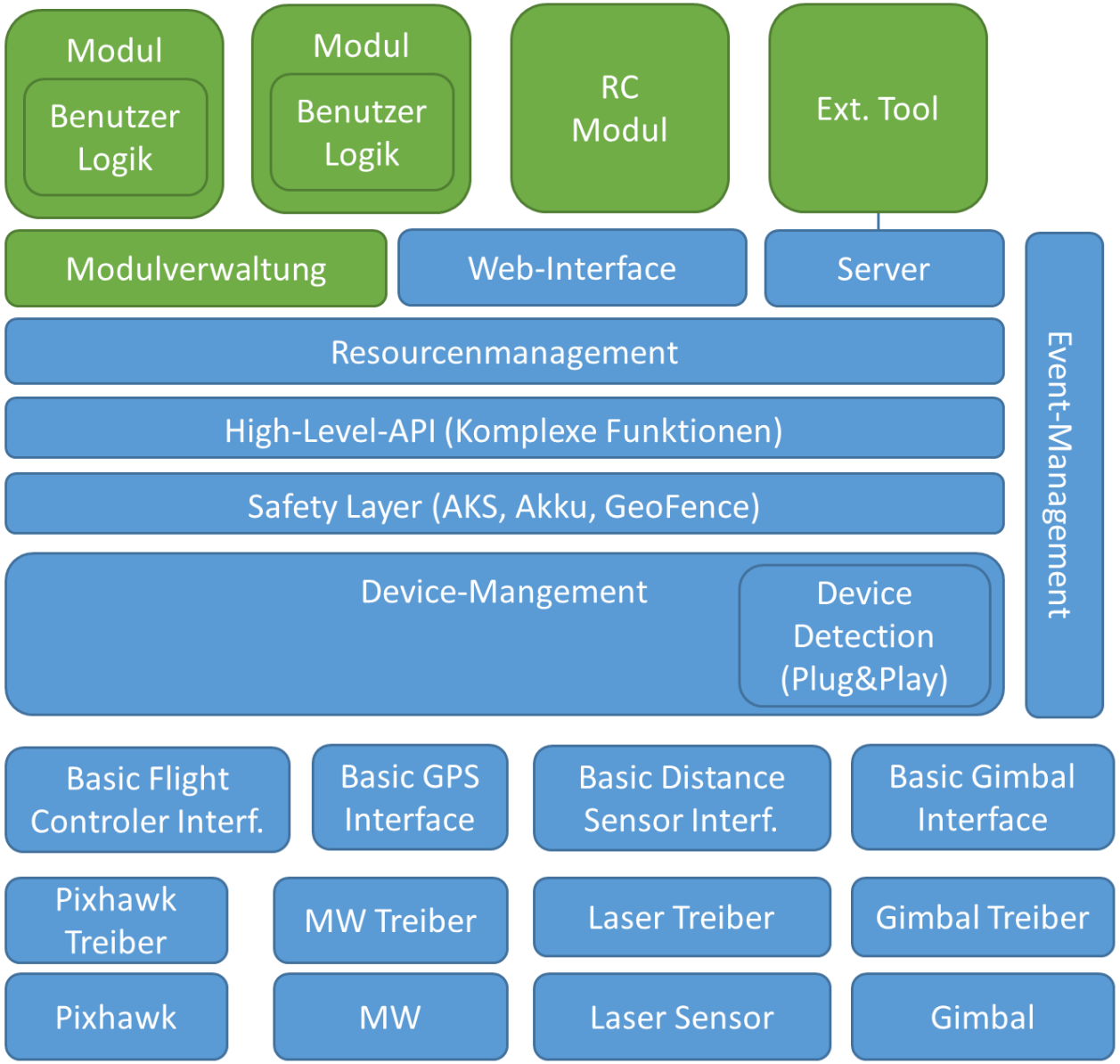
SDK

Gekapseltes System



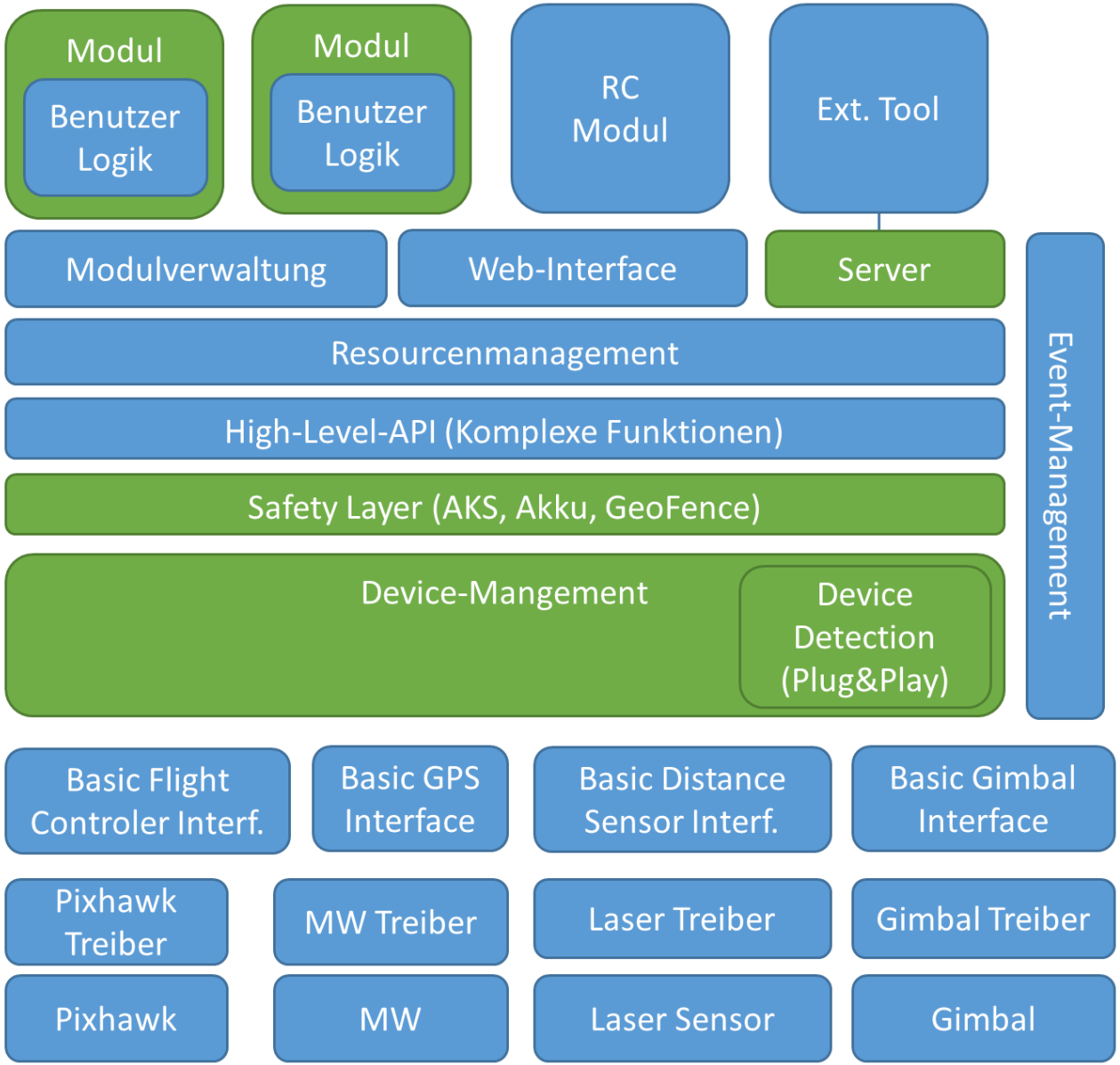
SDK

Multipoint of Control



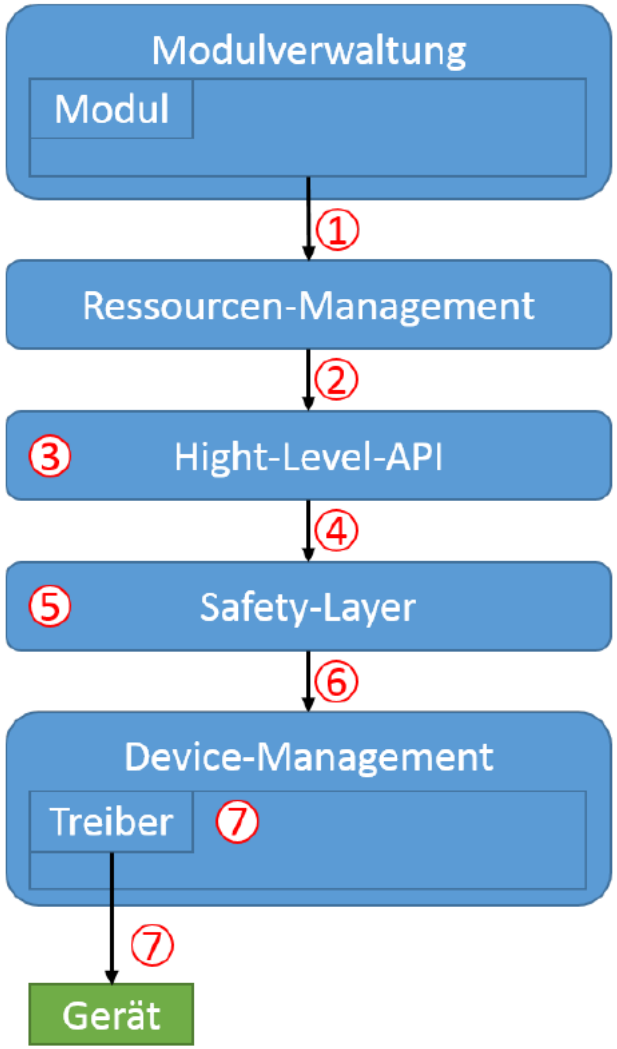
SDK

Sicherheitsaspekt

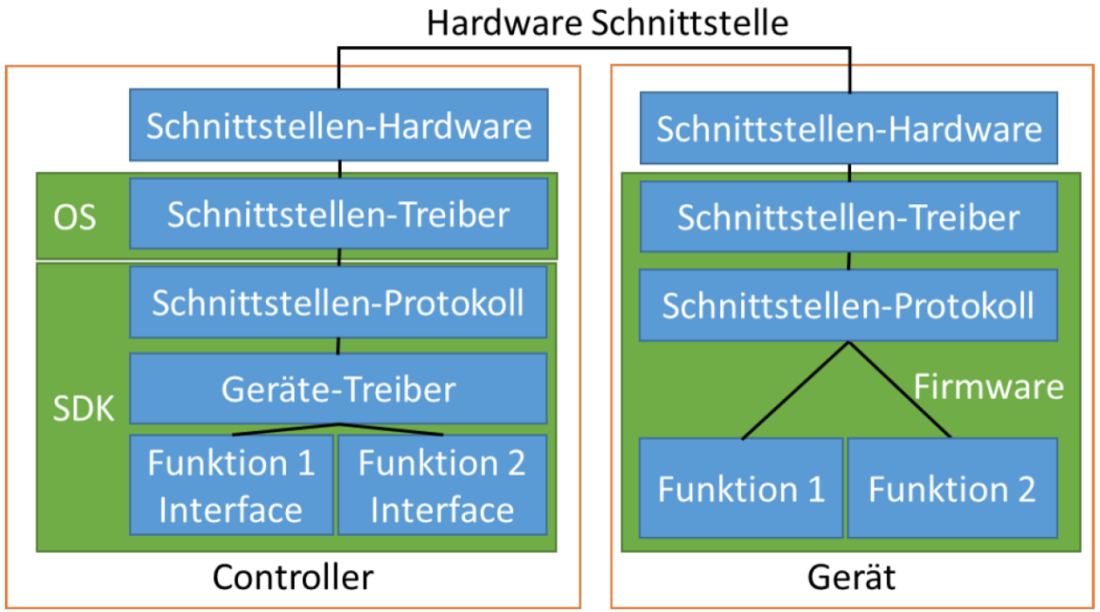


SDK: Befehlsstruktur

1. Das Benutzermodul beantragt bei der Ressourcenverwaltung Schreibrechte auf die Flugsteuerung.
2. Mit Hilfe der High-Level-API wird ein Flugbefehl erstellt.
3. In der High-Level-API wird der Flugbefehl in viele Low-Level-Flugbefehle zerlegt.
4. Jeder erzeugte Low-Level-Flugbefehl wird an den Safety-Layer weitergegeben.
5. Der Low-Level-Befehl wird durch alle Assistenzsysteme im Safety-Layer geprüft.
6. Der Low-Level-Befehl wird an das Device-Management und von dort an den Treiber weitergegeben.
7. Der Treiber übersetzt den Befehl in das gerätespezifische Protokoll.
8. Der codierte Befehl wird an die Flugsteuerung übertragen.

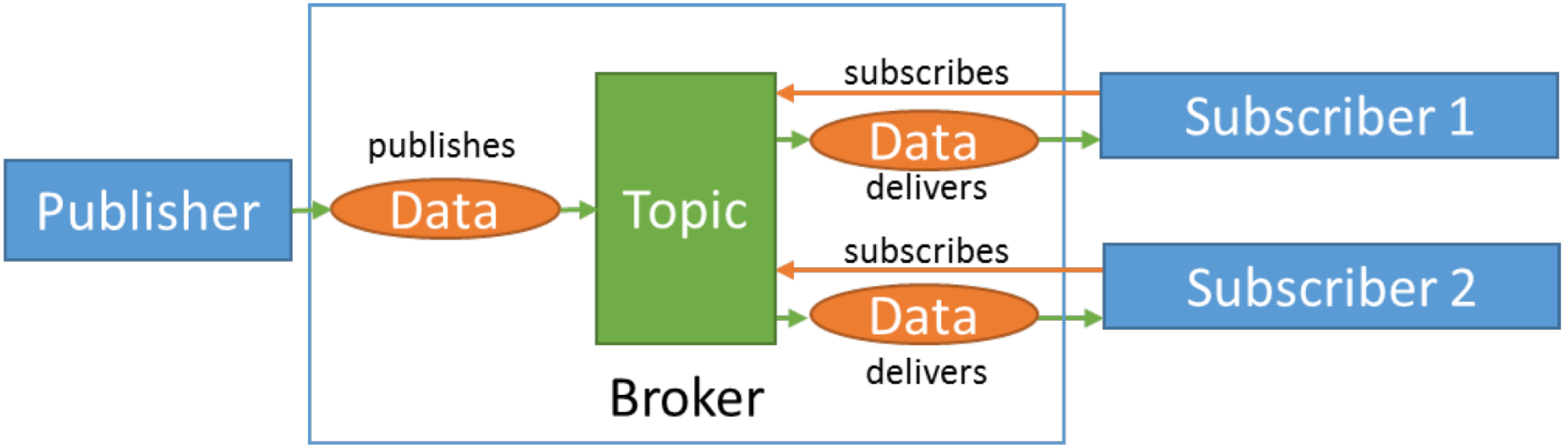


SDK: Ansteuerung der Hardware

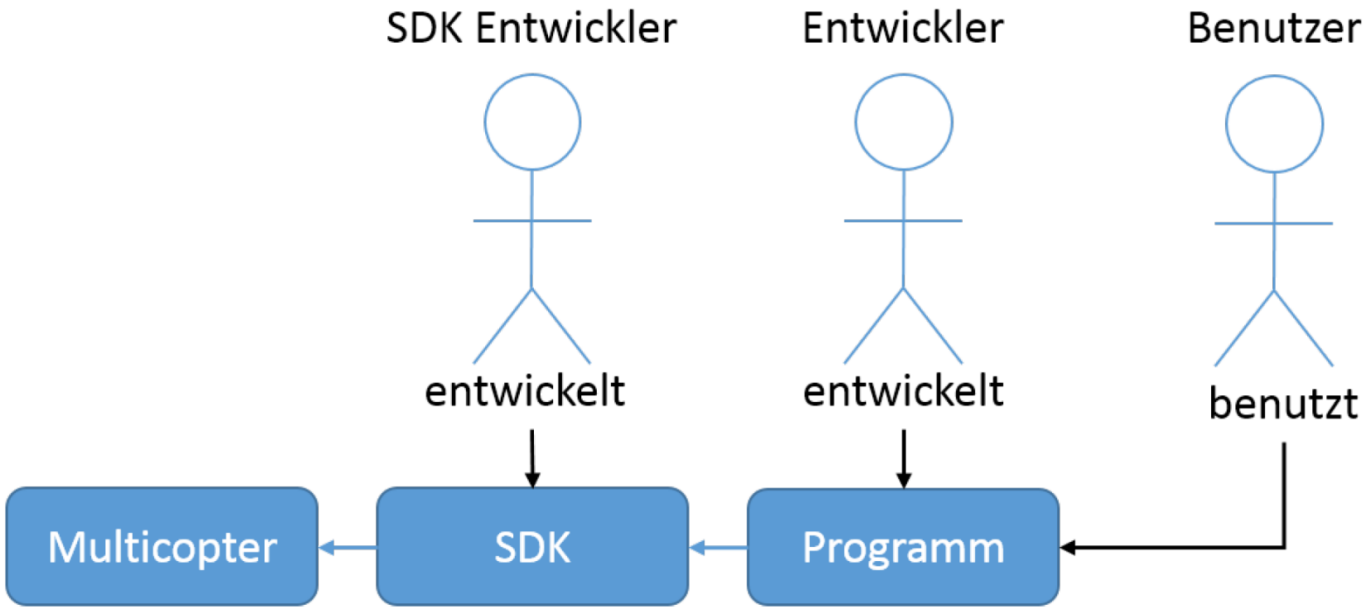


Abstrakter Name aus Abbildung 4.2	**Pixhawk & GPS**	**LED**
Controller	Linux Board	Linux Board
Gerät	Pixhawk	LED
Firmware	PX4	-
Hardware Schnittstelle	4 pol. Kabel	2 pol. Kabel
Schnittstellen-Hardware	UART	GPIO Pin
Schnittstellen-Treiber	Linux UART Treiber	Linux GPIO Treiber
Schnittstellen-Protokoll	Mavlink	-
Geräte-Treiber	Pixhawk Treiber	LED Treiber
Funktion 1	Flugsteuerung	LED
Funktion 2	GPS	-

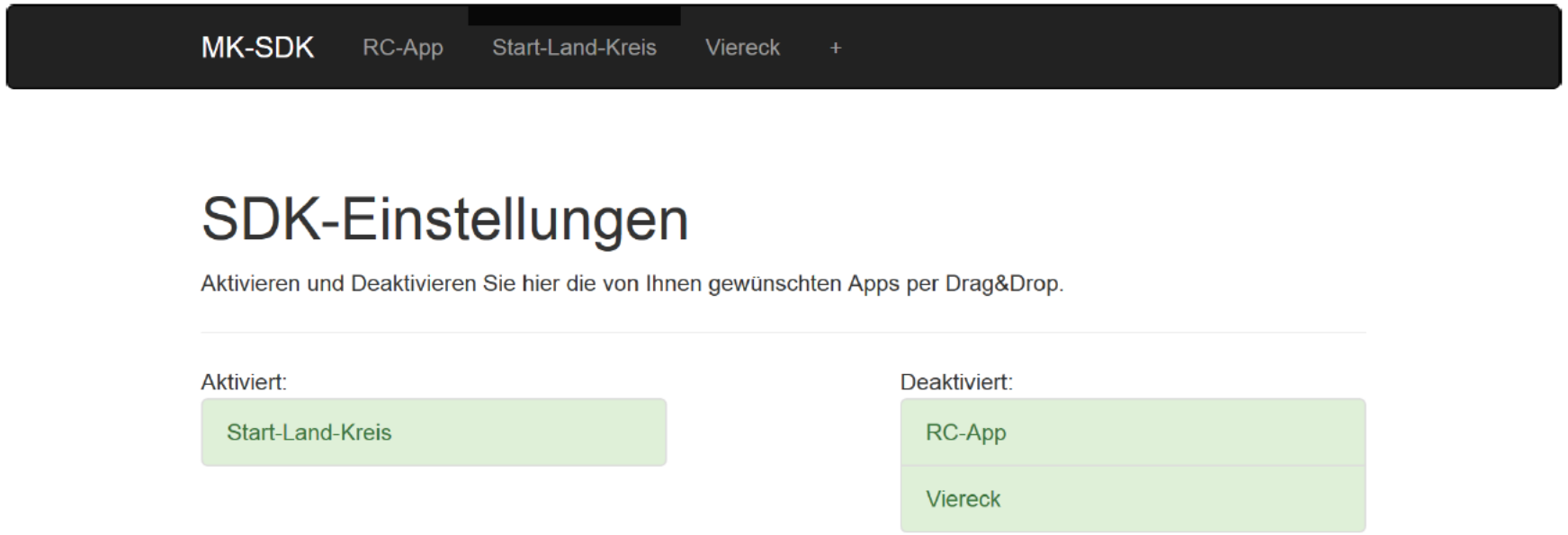
SDK: Daten/Event-Management



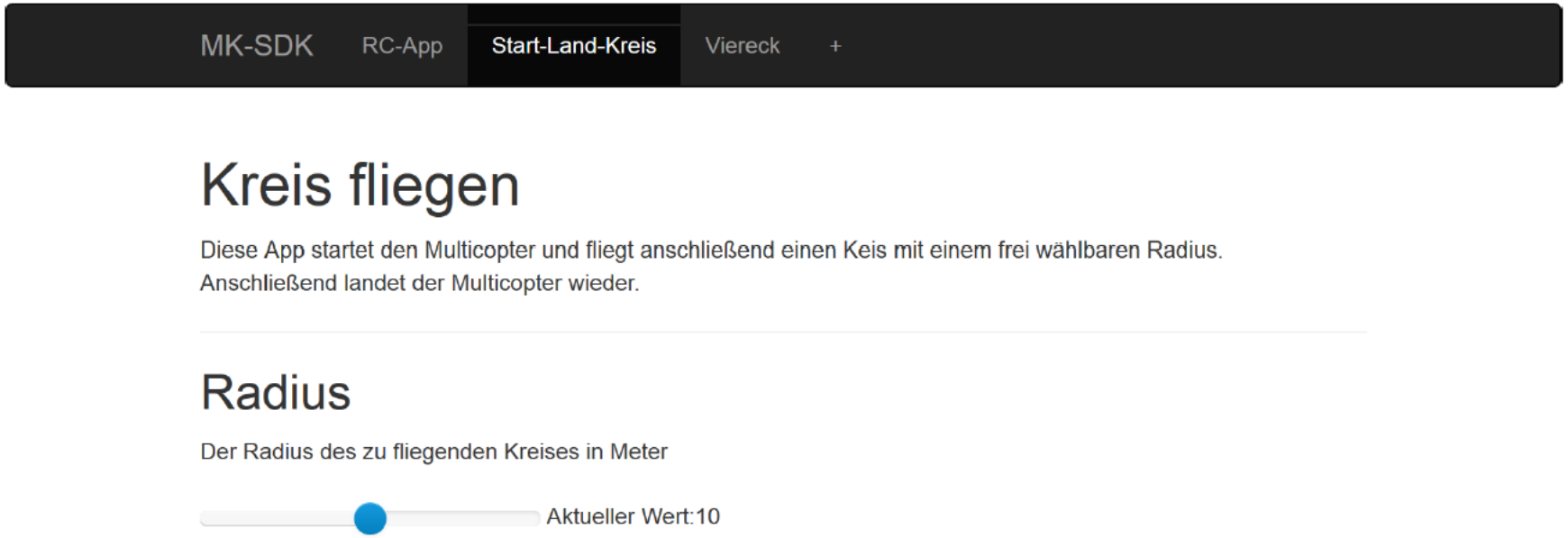
SDK: Benutzergruppen



SDK: App-Verwaltung (Modul)



SDK: App-Verwaltung (Modul)



REST-API

http://80.153.90.104/QuantocopterRESTful/

REST-API

- GET
  - Liste mit Apps
  - Position
  - Attitude (Lage)
  - Geschwindigkeit
  - Beschleunigung
  - Altitude (Höhe)
  - Batteriezustand
  - Empfangsqualität
  - Kamera: Modus, Snapshot, Stream
  - Mission: Start-/Zielpunkt, aktuelle Wegpunkte, History
  - Checkliste
- POST
  - Kameramodus: Video/Bild
  - neue Mission anlegen
  - Start/Stopp/Pause/RTH/Landen
  - Batterie: Kapazität festlegen
- DELETE
  - Mission löschen

Quellcode: http://www.quantocopter.de/files/QuantocoperRESTful.zip

REST-API: Code-Beispiel

```
@Path("/position")
public class PositionResource
{
    @GET
    @Produces({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON, MediaType.TEXT_XML, MediaType.TEXT_HTML })
    @Path("/gps")
    public String getGPS(@Context HttpServletResponse response)
    {
        //set header information
        response.setHeader("Access-Control-Allow-Origin", "*");
        response.setHeader("Access-Control-Allow-Credentials", "true");

        return "N51.148969, E14.979607";
    }
}
```

REST-API: Code-Beispiel

```
@Path("/mission")
public class MissionResource
{
    @POST
    @Path("/newMission")
    @Consumes(MediaType.APPLICATION_JSON)
    public Response newMission(@Context UriInfo uriInfo, @Context HttpServletResponse response, Mission mission)
    {
        //set header information
        response.setHeader("Access-Control-Allow-Origin", "*");
        response.setHeader("Access-Control-Allow-Credentials", "true");

        //check for data object
        if(mission == null)
        {
            throw new RestException(Response.Status.BAD_REQUEST, "Missing data for mission.");
        }

        return this.addMission(mission, uriInfo);
    }
}
```

Zielstellung

- REST-Services für Multikopter-Steuerung möglich?
- Wie gestaltet man die Missionsplanung für den Benutzer so einfach wie möglich?
- Wie können Flugrouten automatisch aufgezeichnet werden (Dokumentation, Wiederholung, gesetzliche Vorgaben)?

Teams

1. Team SDK
   → REST-API mit SDK verknüpfen
   → notwendige Funktionen bereitstellen

2. Team Bodenstation
   → Missionsplanung, automatische App für Upload in SDK

3. Team Flugbuch
   → zentrale Dokumentation der Flugrouten

REST-API

- REST-API URL: http://80.153.90.104/QuantocopterRESTful/
- REST-API Quellcode: http://www.quantocopter.de/files/QuantocoperRESTful.zip

Installation/Einrichtung

1. Download + Installation von Netbeans, SE-Version ausreichend
2. Gradle Plugin in Netbeans installieren (Tools → Plugins, Plugin 'Gradle Support')
3. Daten-Download: http://www.quantocopter.de/files/SDK-DATA.zip
4. Plugin in Netbeans installieren: *de-quanteo-mksdk.nbm*
5. neues Projekt in Netbeans anlegen: Typ 'Multicopter App'
6. eigenes Modul mit Beispielcode steht jetzt zur Verfügung
7. Beispielcode "App.java" zeigt das Abgreifen von Daten des Multikopters

Pixhawk ansteuern

1. Pixhawk an PC via USB anschließen
2. PC sollte Pixhawk korrekt erkennen
3. das SDK ist in der Lage, Livedaten vom Pixhawk abzugreifen
4. falls der Pixhawk nicht angesprochen wird: im Netbeans-Projektordner im Modul die Datei dev.conf löschen

Simulation starten

1. Download + Installation des Simulators: http://www.aerosimrc.com/en/download.htm
2. kompletten SIL-Ordner in den Plugin-Ordner kopieren: <AeroSIM-RC-Ordner>/Plugin
3. Dongle anstecken, sonst nur 2 Minuten Demomodus
4. Simulator starten
5. *Sim -> Select Multirotor* und 'DJI-F405' auswählen
6. *Plugin -> SIL Plugin* auswählen
7. Modul in Netbeans starten
8. Meldung mit "OK" bestätigen