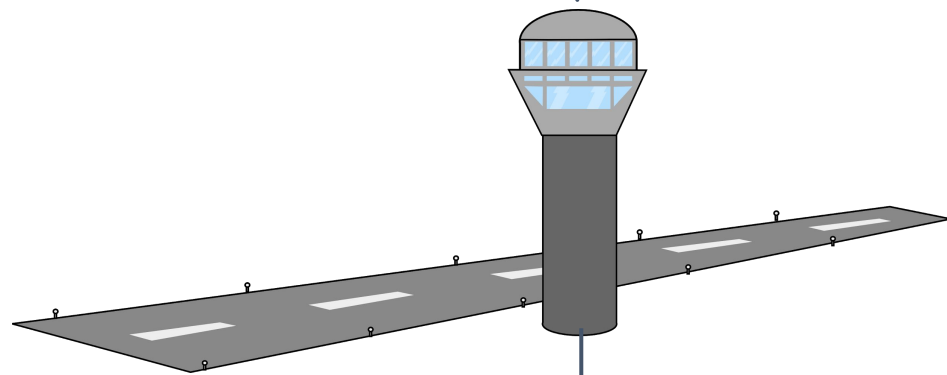
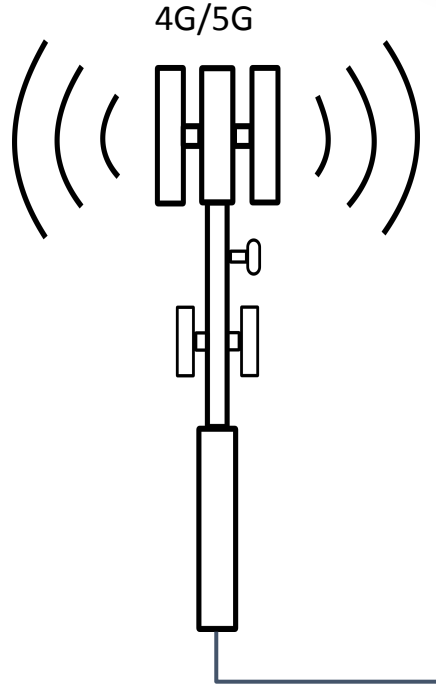


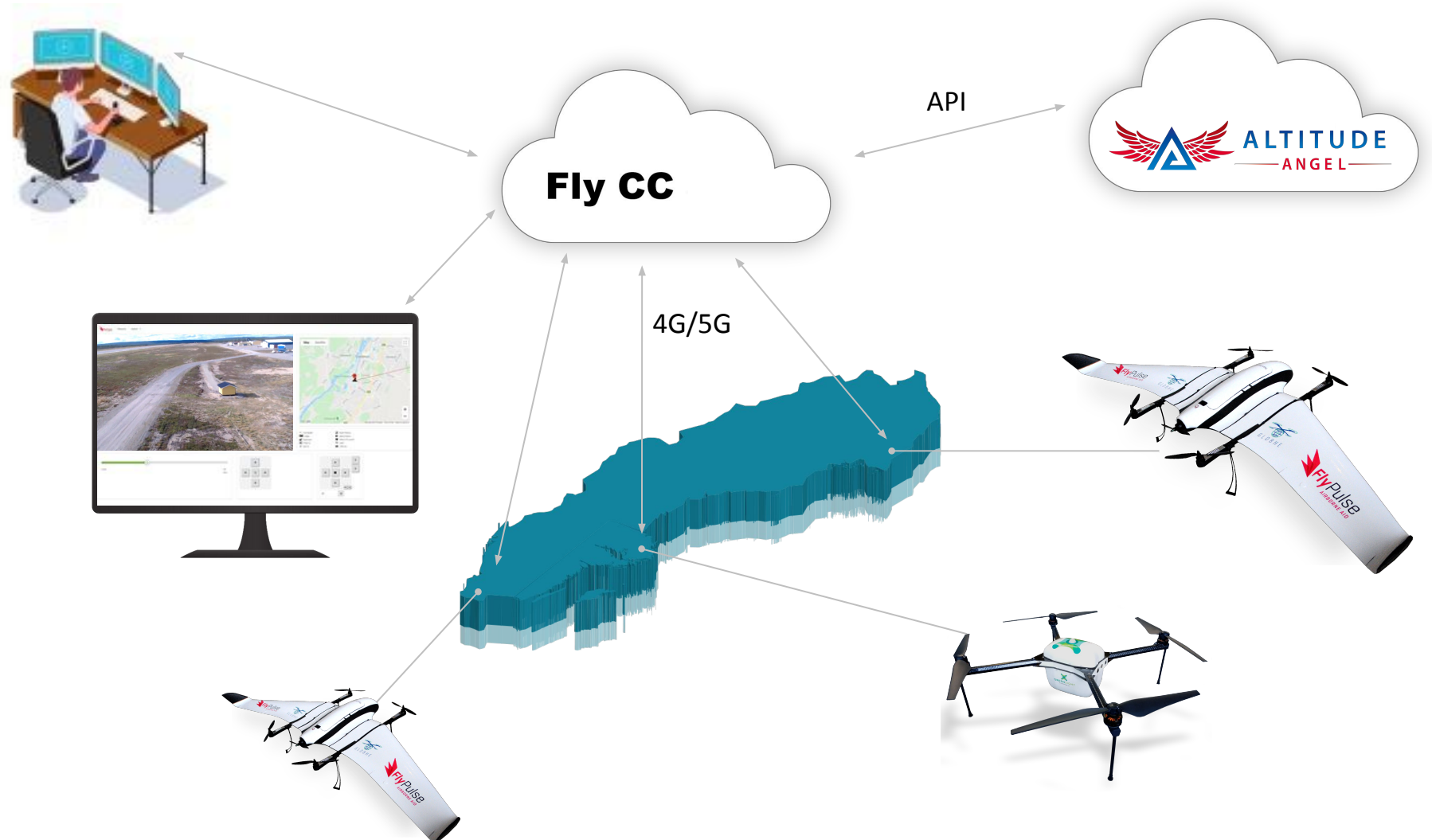


# Fly CC 3.0

Cloud servers

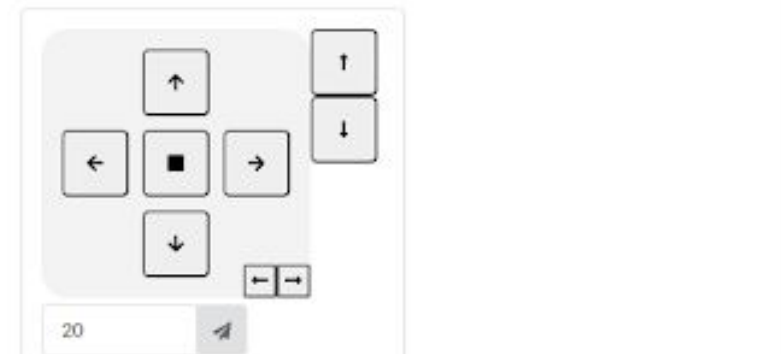


# Flypulse systemskiss





- Connected
- 99%
- Armed
- Waypoint
- 85 m
- Push Mission
- Start Mission
- Return To Launch
- Land
- 15.57 m/s



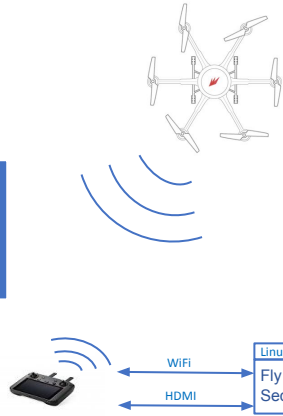
# THE FUTURE OF DRONE MANAGEMENT

Autonomous and Beyond Visual Line of Sight (BVLOS)!

Use-cases for our cloud platform

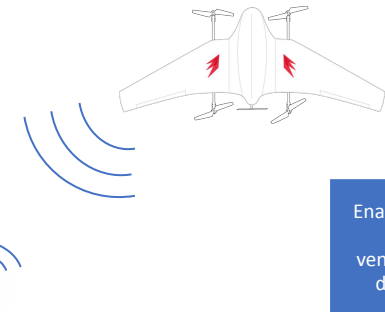
## Fly Stream Secure

Enables encrypted video stream from vendor-independent drones to Fly CC



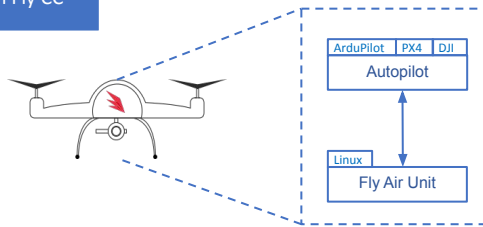
## Fly Stream Lite

Enables video stream from vendor-independent drones to Fly CC



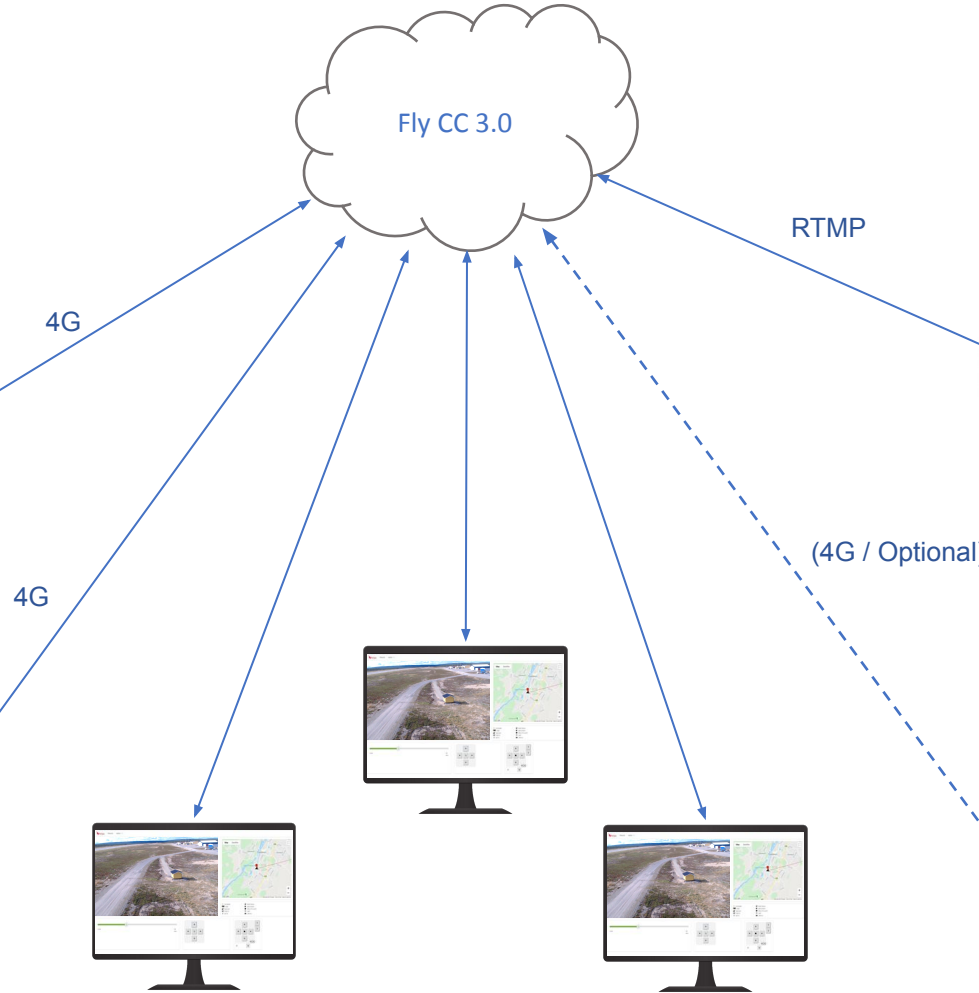
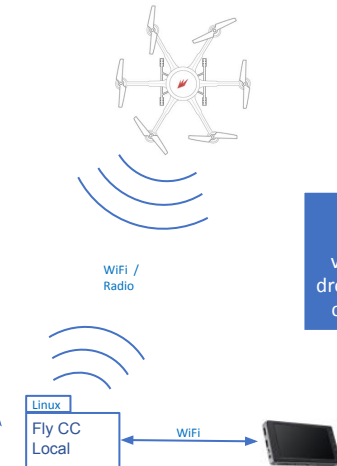
## Fly Air Unit

Enables drone management for vendor-independent drones with Fly CC



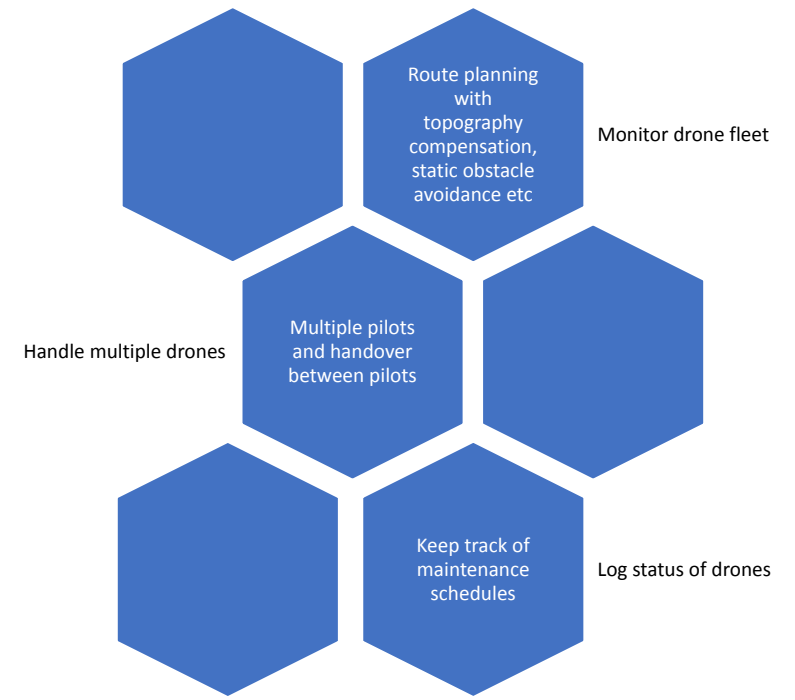
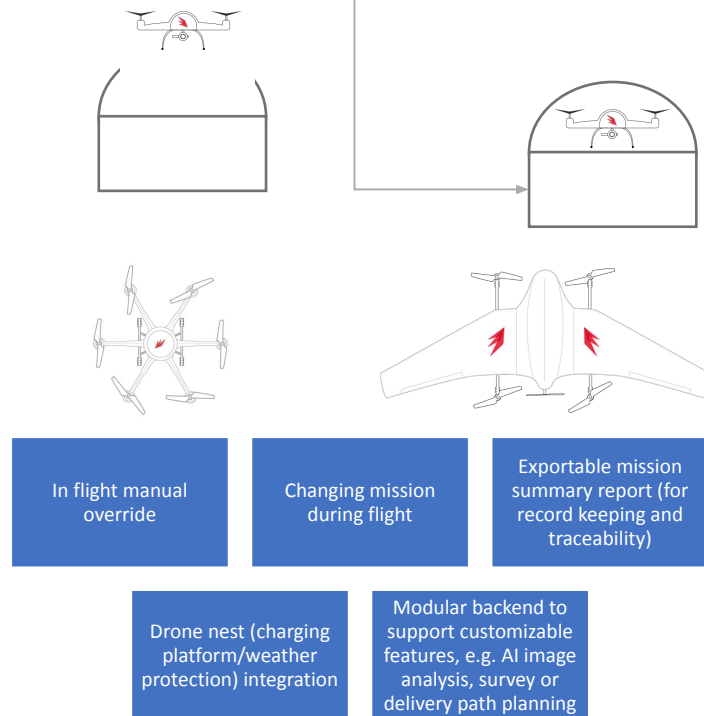
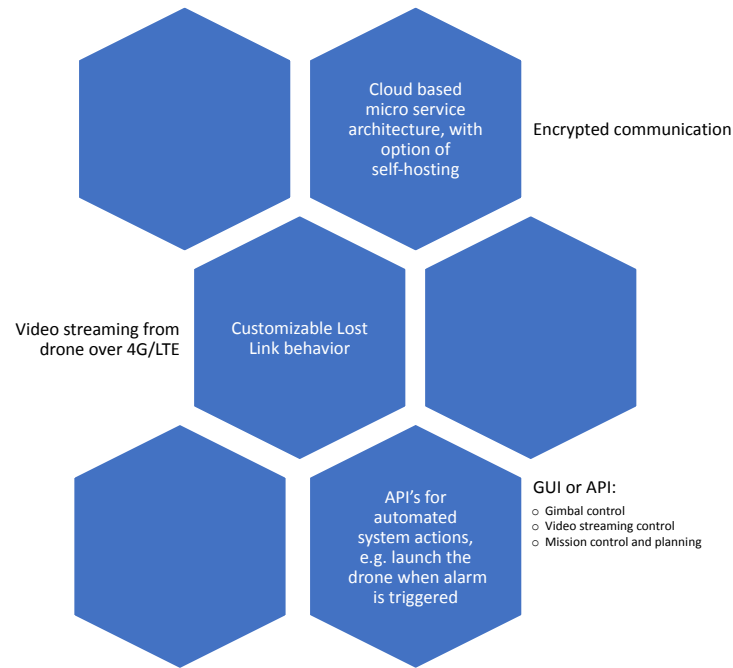
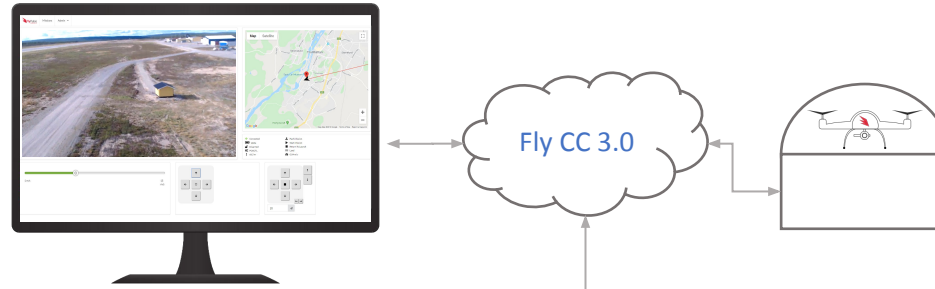
## Fly CC Local

Enables drone management for vendor-independent drones without internet connection to Fly CC

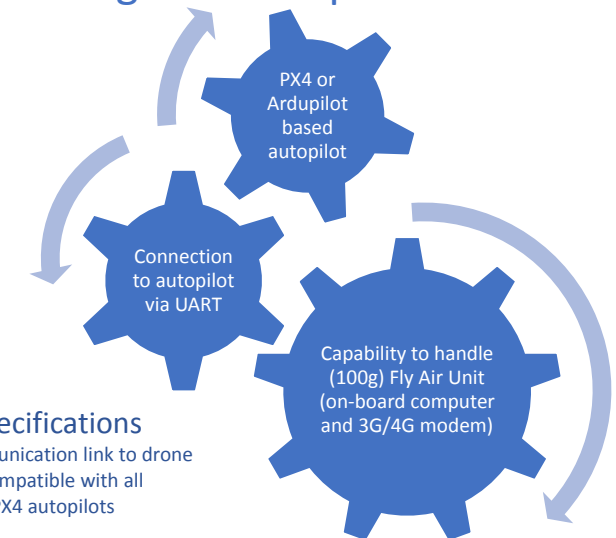


## Supports Automated and Beyond Visual Line of Sight (BVLOS) operations

Fly CC is a cloud-based drone fleet management system for planning and executing BVLOS missions.

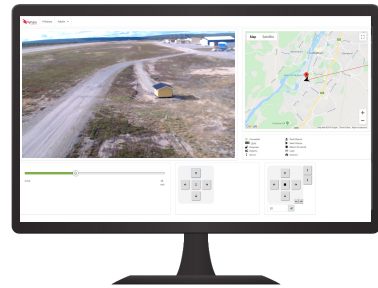


## Integration Requirements

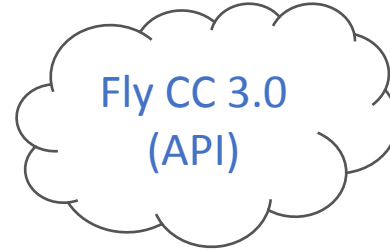


### System specifications

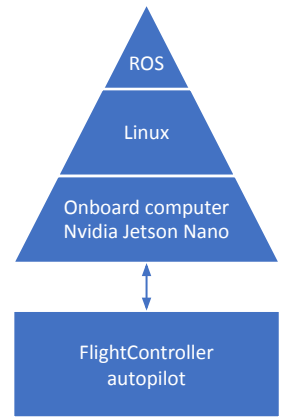
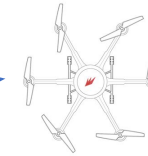
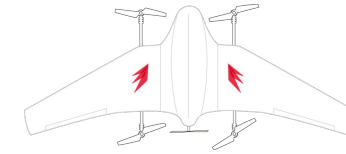
- 4G/LTE communication link to drone
- Fly Air Unit compatible with all Ardupilot and PX4 autopilots



Internet



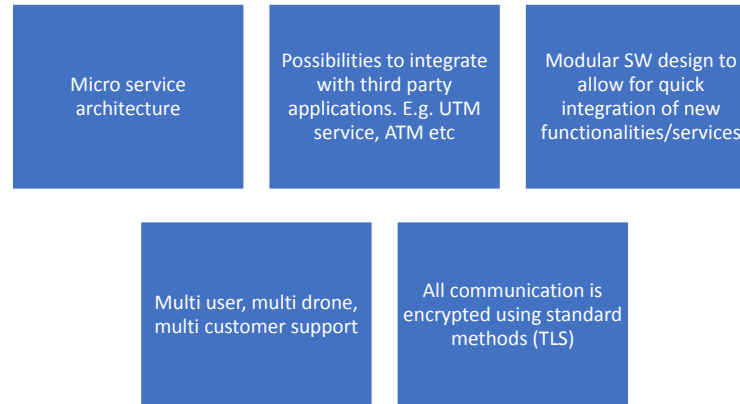
4G / LTE



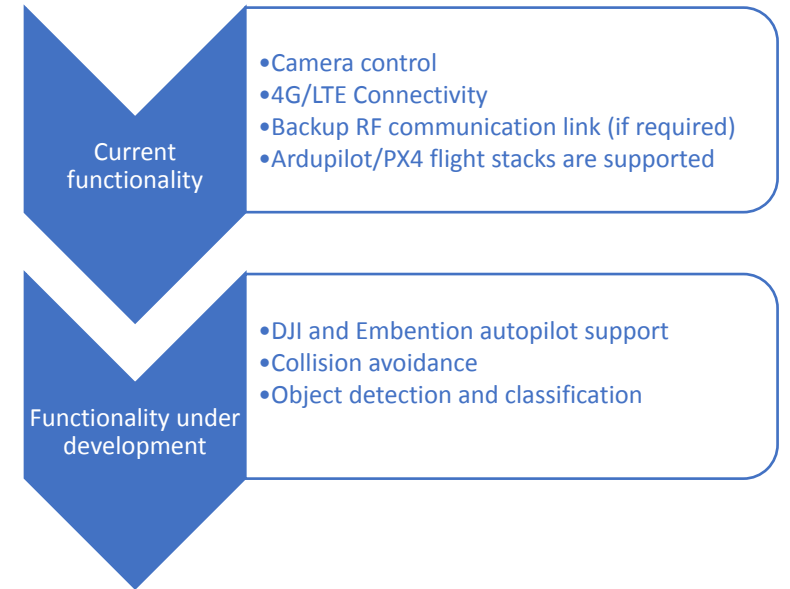
### WEB-UI functionality

- Create/edit missions
- Camera control
- Vehicle telemetry presentation
- Live stream video
- Manual control and override
- Data presentation

### Cloud functionality



### Drone functionality



After Fly CC 1.0, a proof of concept, came Fly CC 2.0. Now after more than 3 years development and over 10.000 development hours Flypulse launch Fly CC 3.0.

# Software contents

- Companion computer software (for the drone)
  - Camera control and streaming
  - Autopilot communication link
  - Cloud communication link
- Backend components (mostly python code)
  - Gatekeeper
  - Authentication
  - HTTP Gateway
  - Drone communication
  - Video Streaming
  - Databases for storing user and mission data
- Frontend components (React)
  - Login
  - Mission control

# Software contents, continued

- Software contents contained in git repositories for each component for version control and development history
- Readme-documentation for setting up and deploying the software stack
- Docker registry containing the docker images for components
- Code contained in 740Mb tar.gz compressed file
- Docker registry contained in 10Gb tar.gz compressed file



