Autonomous Drone Engineer

B1 – Architecture Introduction

Paul.Guermonprez@intel.com
Autonomous Drone Solutions Architect

Define your project

Before we talk about the drone hardware and software architecture, we need to define your project:

- Is it a Unmanned Aerial Vehicle (UAV) or System (UAS)?
- What is the level of autonomy you need?
- What is part of the flying intelligence, and part of your payload?
- What is your business model: are you selling hardware or a service?

UAV or UAS?

With RC drones, you have a drone and a pilot.

With autonomous drone fleets, you're removing the pilot but drone is not left alone. IT needs several **server services** to work:
mapping, flight authorizations, data storage and analysis.

In fact, such drones are part of a larger IT system.
They are the part of this IT system interacting with the physical world, by capturing data or moving objects.

If a drone is autonomous and professional,
it's likely a UAS.

Autonomy – Levels

Autonomy means integrating two opposing constraints:

- Being able to **follow** a mission, planned or defined on-the-fly (harder)
- While **avoiding** problems like collision with random objects

Having the two **interact** finely shows more autonomy.

Examples:

- Low autonomy: Following simple instructions, like a given GPS path
- Higher autonomy: Follow a GPS path, and stop to avoid collisions
- Highest autonomy: Following a GPS path, avoiding obstacles and be able to adapt and redefine the mission depending on obstacles

Autonomy – Flight phases

Your drone may be autonomous for certain parts of the flight and not others:

- Perhaps the drone can follow GPS paths at high altitude but requires a pilot to take-off and land?
- Or it is able to avoid collisions at low speed, close to the ground, but requires supervision when flying at high altitude/high speed?

A fully autonomous drone requires different solutions for different flight phases. Specify your needs.

Drone Flight Stack or Payload?

Drones have a flight stack to fly, but they also carry software and hardware workloads.

They have different characteristics,
solving different problems.
It is not always useful/easy to have them interact.

Ex: a drone deployed over a mine may have full 3D short range sensors to avoid collisions, plus a 3D long range laser sensor to scan the mine. But the LIDAR may not be useful to avoid collisions, and the short range 3D Sensor of the drone may not help with the mine scanning.

Photos: Flylab.io drone presentation video

https://www.youtube.com/watch?v=XRTTNbXkrO8

Business model: solution or hardware?

Professional clients can buy a RC drone to perform some tests and perhaps 1 autonomous drone to test their software.

But they will likely outsource their drone operation to specialists, just like they do for their helicopter/jet fleets for the following reasons:

- Flying carries a legal responsibility
- Maintenance and operation will probably be cheaper large fleets are shared between several customers
- They'll want to focus on their differentiator: their sensor payload? their data processing algorithm? their API?

Be specific: are you providing a core technology?
A drone as hardware? A drone-based service?

Conclusion

Each drone solution is **highly specialized**, **dedicated for a specific task**

Be clear about your **goals** and **expectations**

The architecture will be a lot **easier to define**

And your drone solution easier to **position on the market**

Thanks

Paul.Guermonprez@intel.com