Reducing the Mission Time of Drone Applications through Location-Aware Edge Computing

Theodoros Kasidakis, Giorgos Polychronis, Manos Koutsoubelias and Spyros Lalis

Computer Systems Lab
ECE Dept, University of Thessaly
Volos, Greece

Auto-Scout

European Union
European Regional Development Fund

HELLENIC REPUBLIC
MINISTRY OF ECONOMY & DEVELOPMENT
SPECIAL SECRETARY FOR ERDF & CF
MANAGING AUTHORITY OF EPAnEK

EPAnEK 2014-2020
OPERATIONAL PROGRAMME
COMPETITIVENESS
ENTREPRENEURSHIP
INNOVATION

ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Partnership Agreement 2014 - 2020

Co-financed by Greece and the European Union

Drones as key system components

Applications

Restrictions

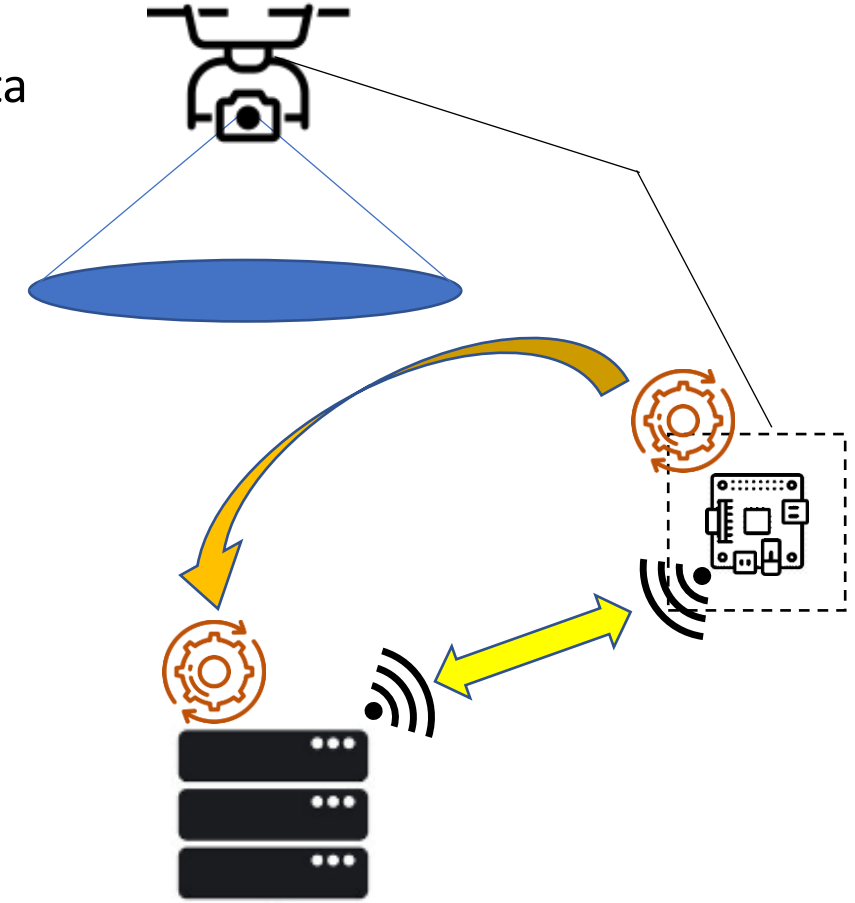
System model

- Drone scans an area by going through a list of waypoints
- In each waypoint: takes sensor measurements & processes this data
  - to notify the user or perform some actuation

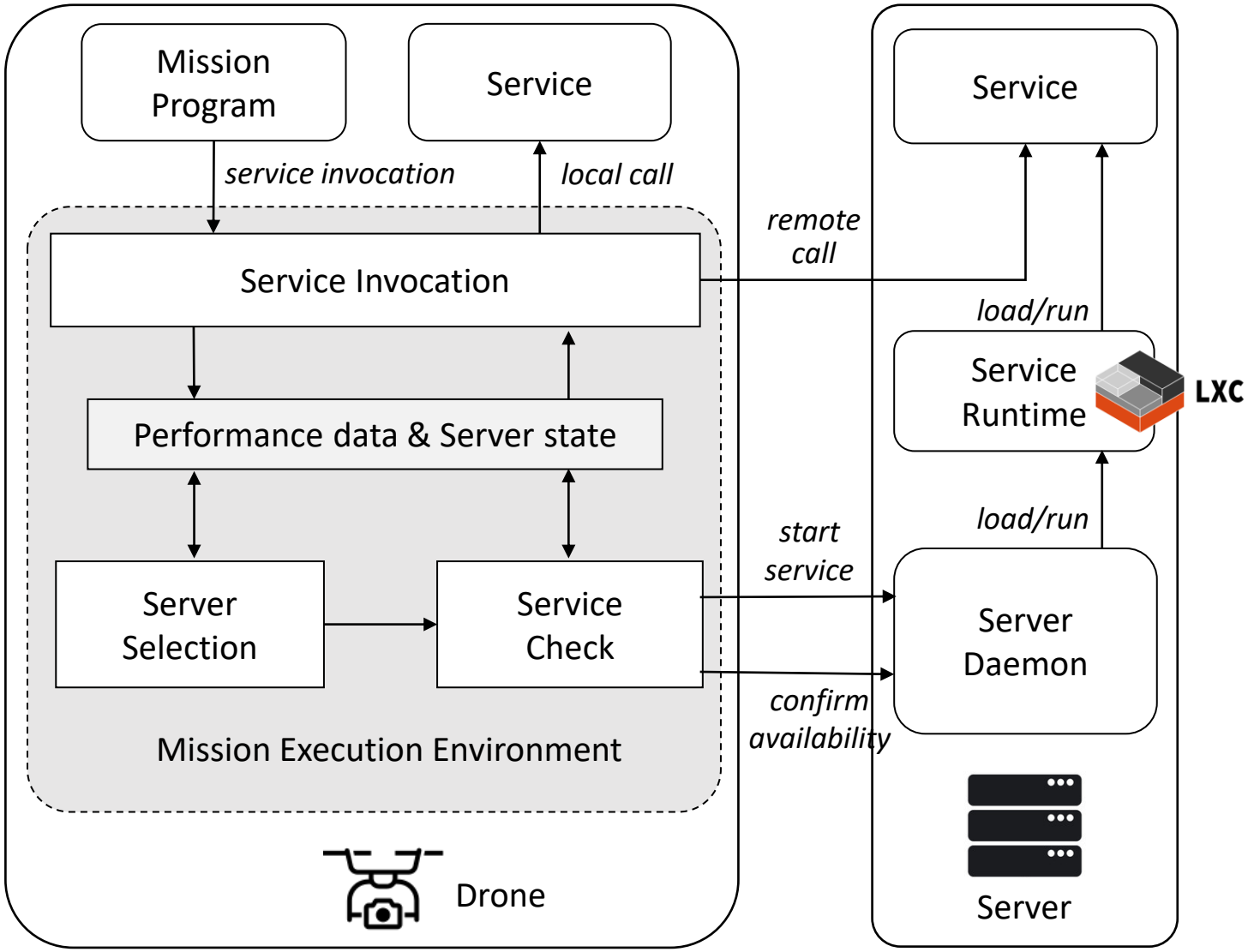
- Drone has its own onboard computer
  - used to run the application software/logic
  - used to perform the data processing / computations

- Servers near the mission area
  - location of servers is known
  - each server is accessed through a separate WLAN
  - server availability is not guaranteed

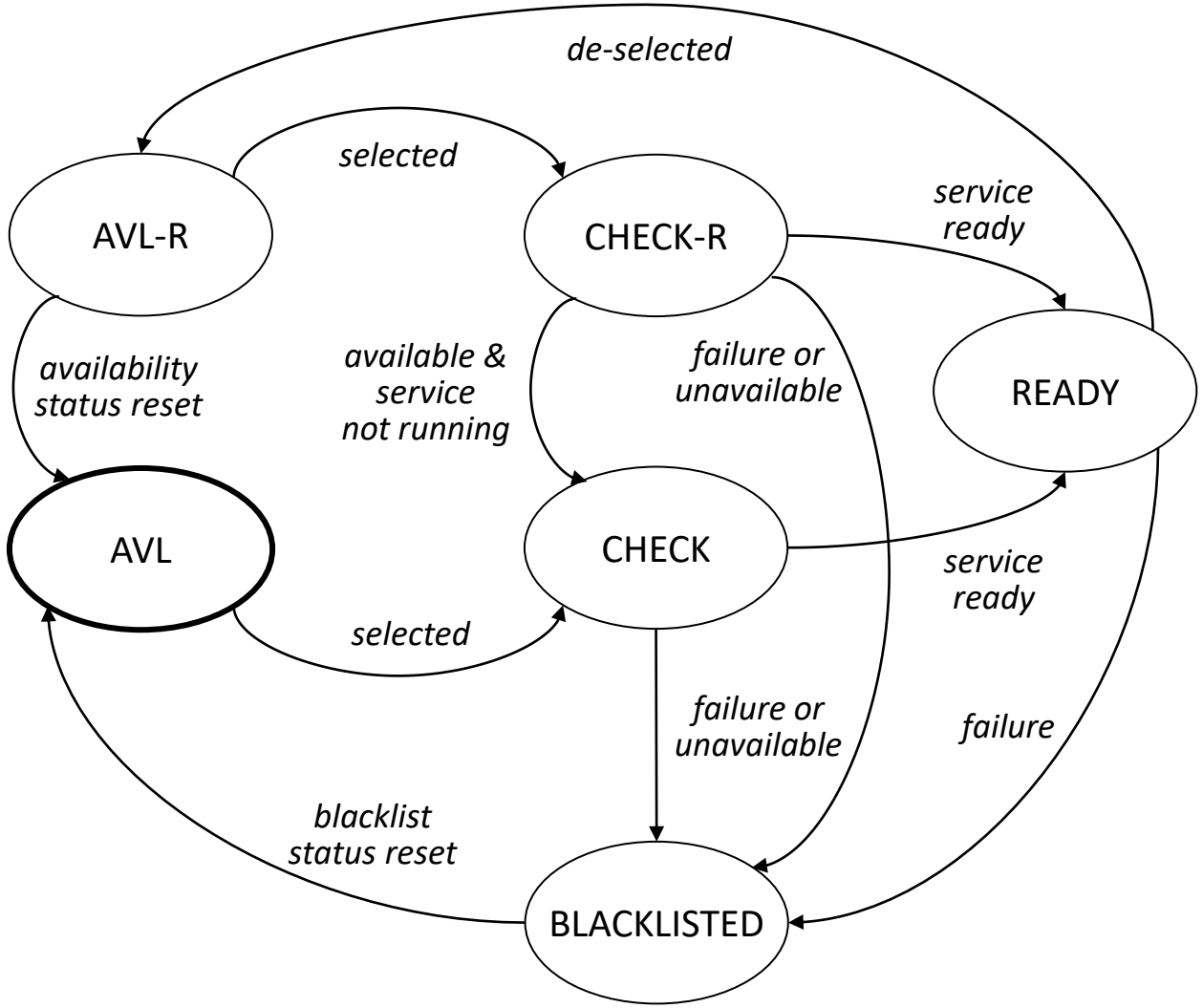
- Drone can offload its computation on such servers
  - in an opportunistic, ad-hoc way



Software architecture



State management



Key parameters of server selection & invocation

Symbol	Description
$Call$	Local end-to-end service call delay.
$Data$	Size of service request and reply data that need to be sent over the wireless networks.
$Proc_s$	Remote service processing time on server s.
$Call_s$	Remote end-to-end service call delay on server s, including data transfer over the wireless network.
$Check_s$	Connection & availability confirmation delay when the service needs to be started on s.
$CheckR_s$	Connection & availability confirmation delay when the service is already running on s.

Server selection & service invocation

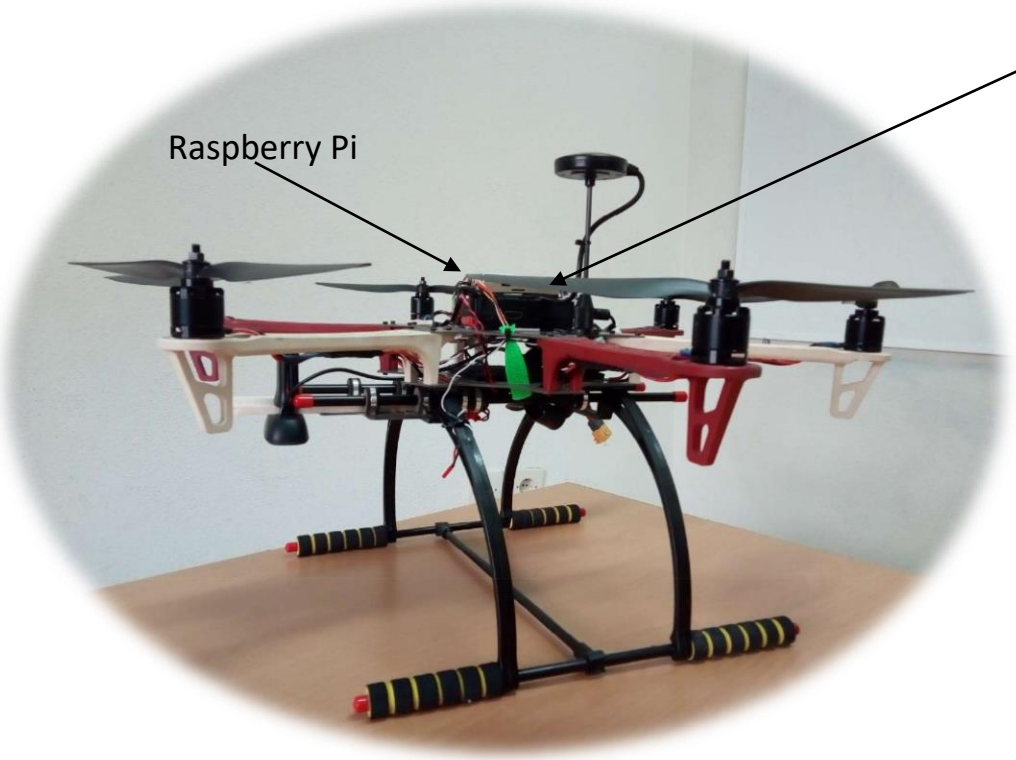
Server selection

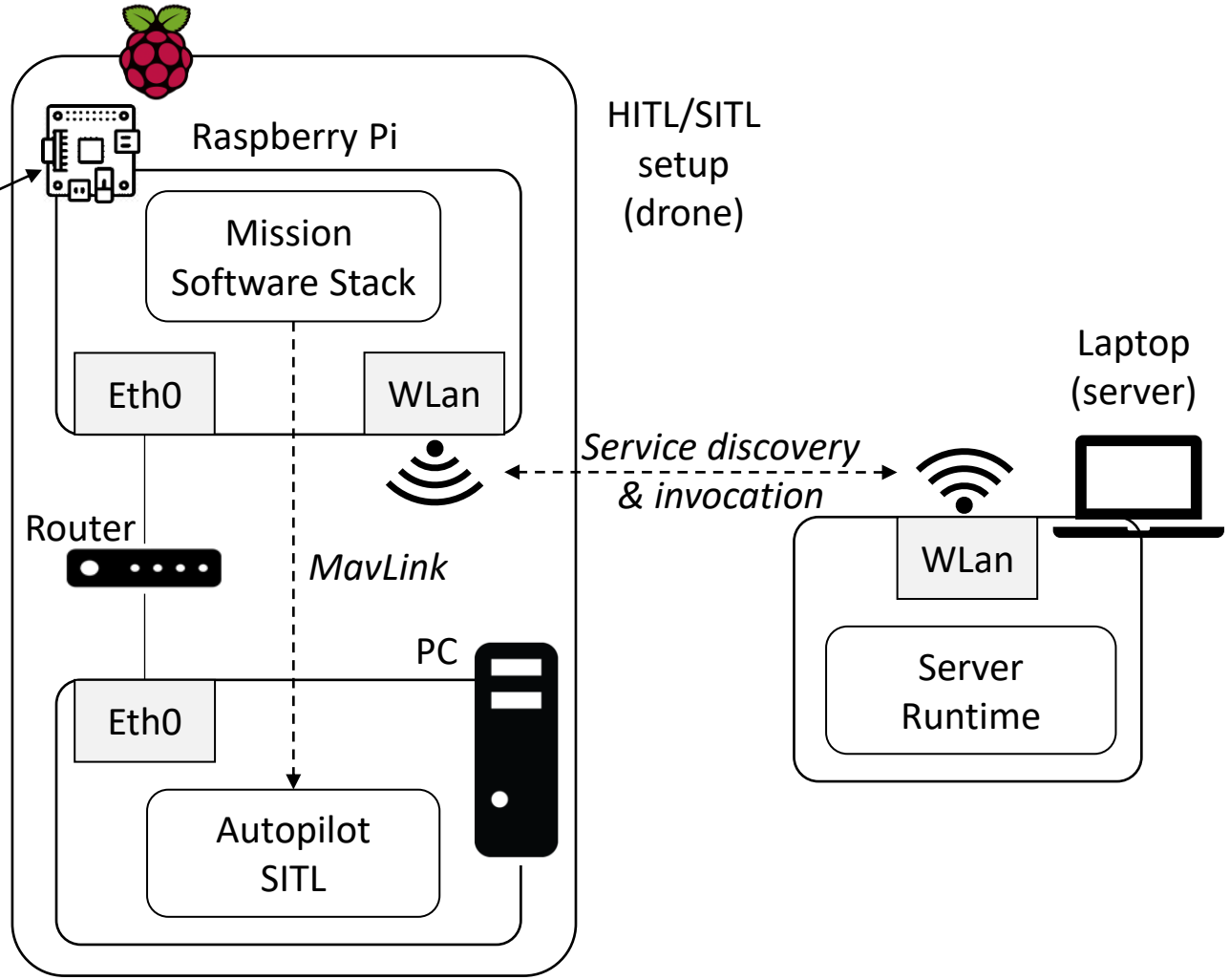
- $candidate(s) = inrange(s) \wedge Call_s < Call \wedge s.state \neq BLACKLISTED$
- $best(s) = candidate(s) \wedge \nexists\, s' : candidate(s') \wedge Call_{s'} < Call_s$
- $select(s) = best(s) \wedge \frac{Call_{cur} - Call_s}{Call} > Gain_{switch}$

Service invocation

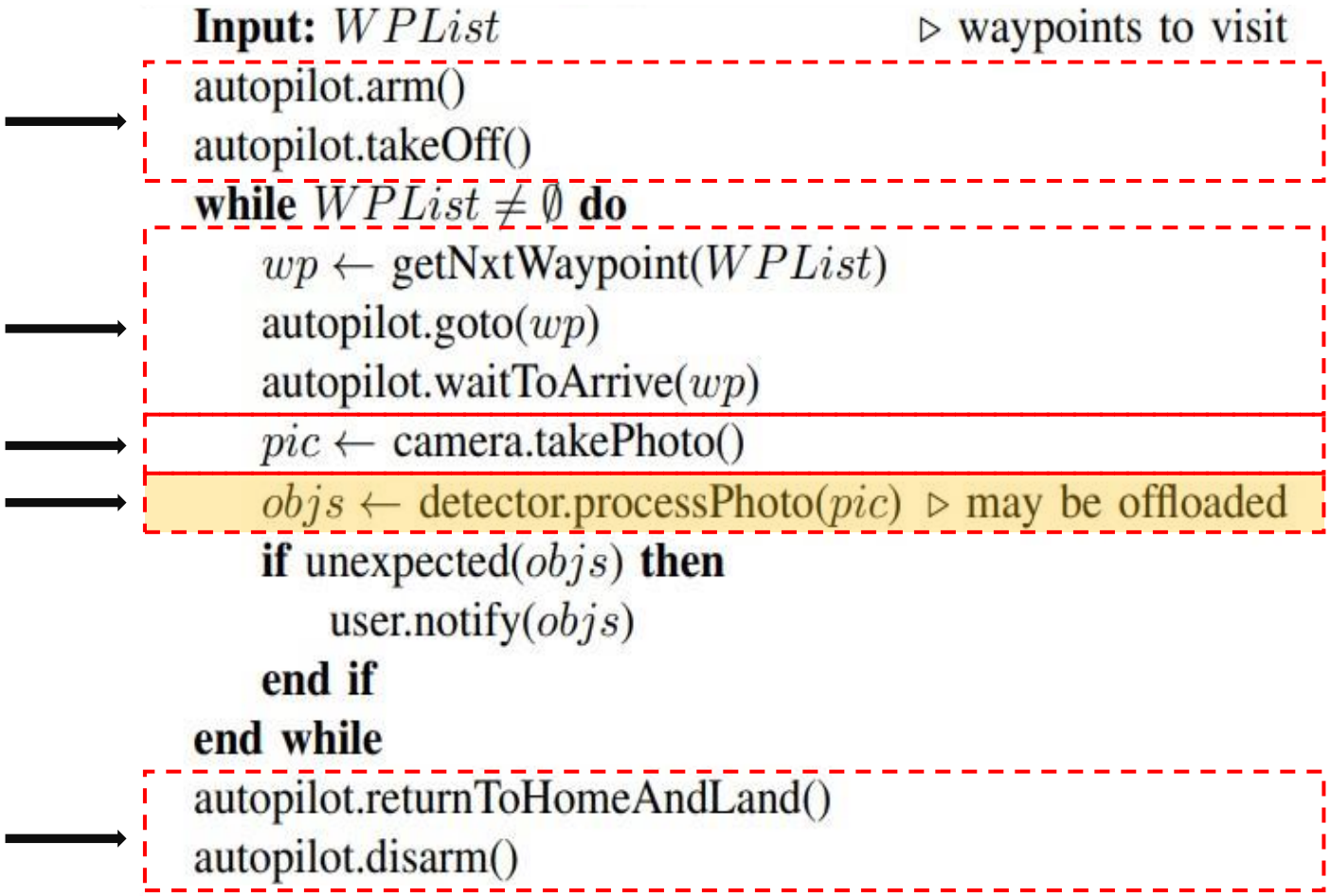
- $\frac{Call - (Call_{cur} + waitT(Check_{cur}))}{Call} > Gain_{wait}$
- $waitT(X) = X - (getTime() - t_{check})$
  - X value can be $Check_{cur}$ or $CheckR_{cur}$
  - t_{check} : time of server selection

Experimental setup





Mission program template



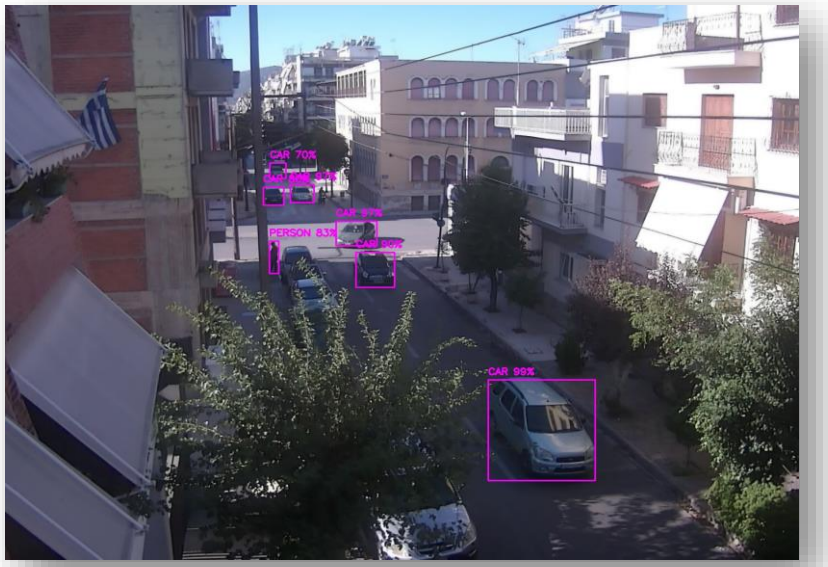
```
Input: WPList                                   ▷ waypoints to visit
autopilot.arm()
autopilot.takeOff()
while WPList ≠ ∅ do
    wp ← getNxtWaypoint(WPList)
    autopilot.goto(wp)
    autopilot.waitToArrive(wp)
    pic ← camera.takePhoto()
    objs ← detector.processPhoto(pic)  ▷ may be offloaded
    if unexpected(objs) then
        user.notify(objs)
    end if
end while
autopilot.returnToHomeAndLand()
autopilot.disarm()
```

Service characteristics

- Photos are processed by an object detection service
  - light or heavy mode

light (secs)	heavy (secs)
3	10

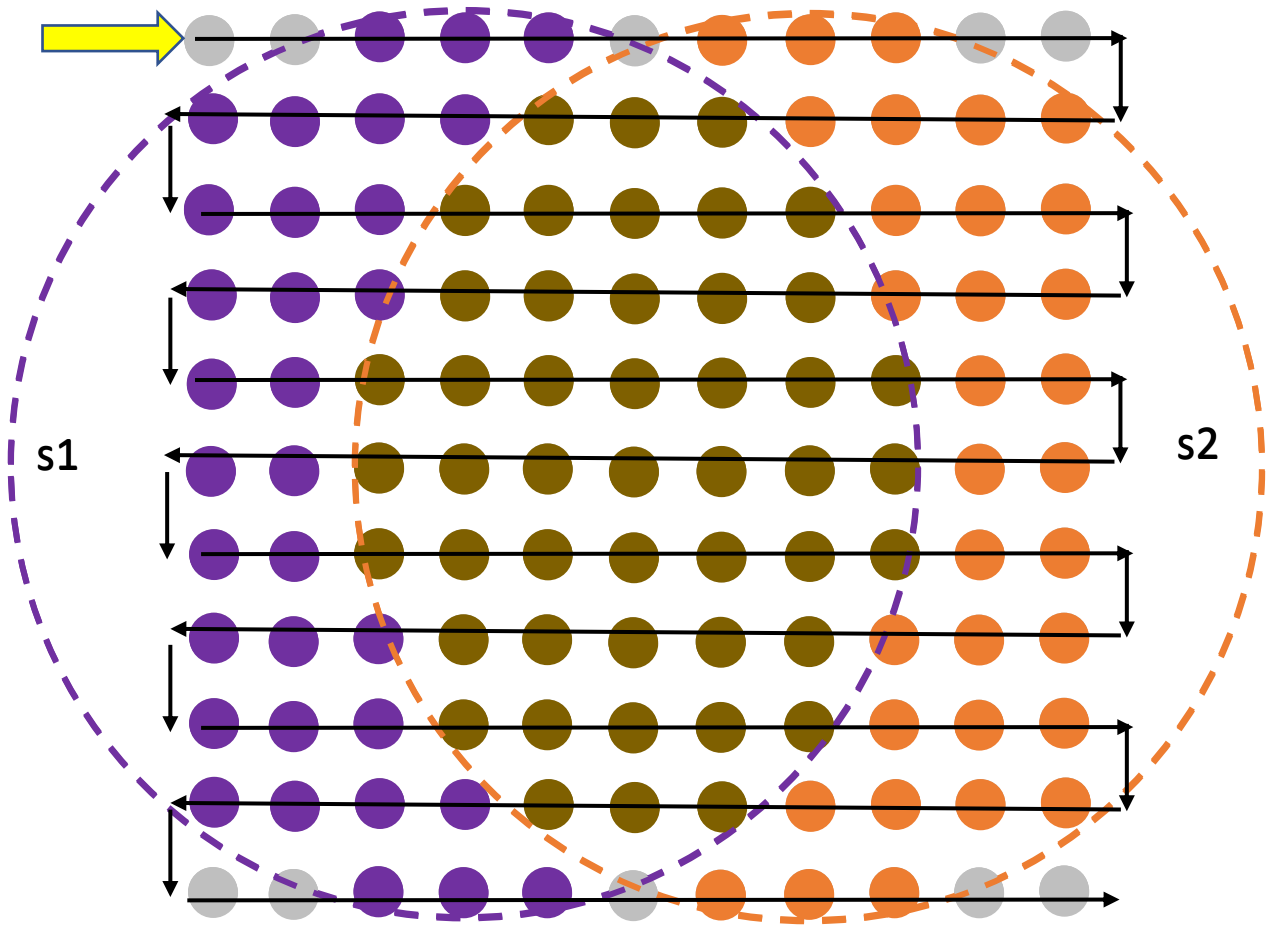
light (secs)	heavy (secs)
0.3	0.7



- Startup time on the server (load and service start) is about 8 seconds
- Gain thresholds are set both to 25%

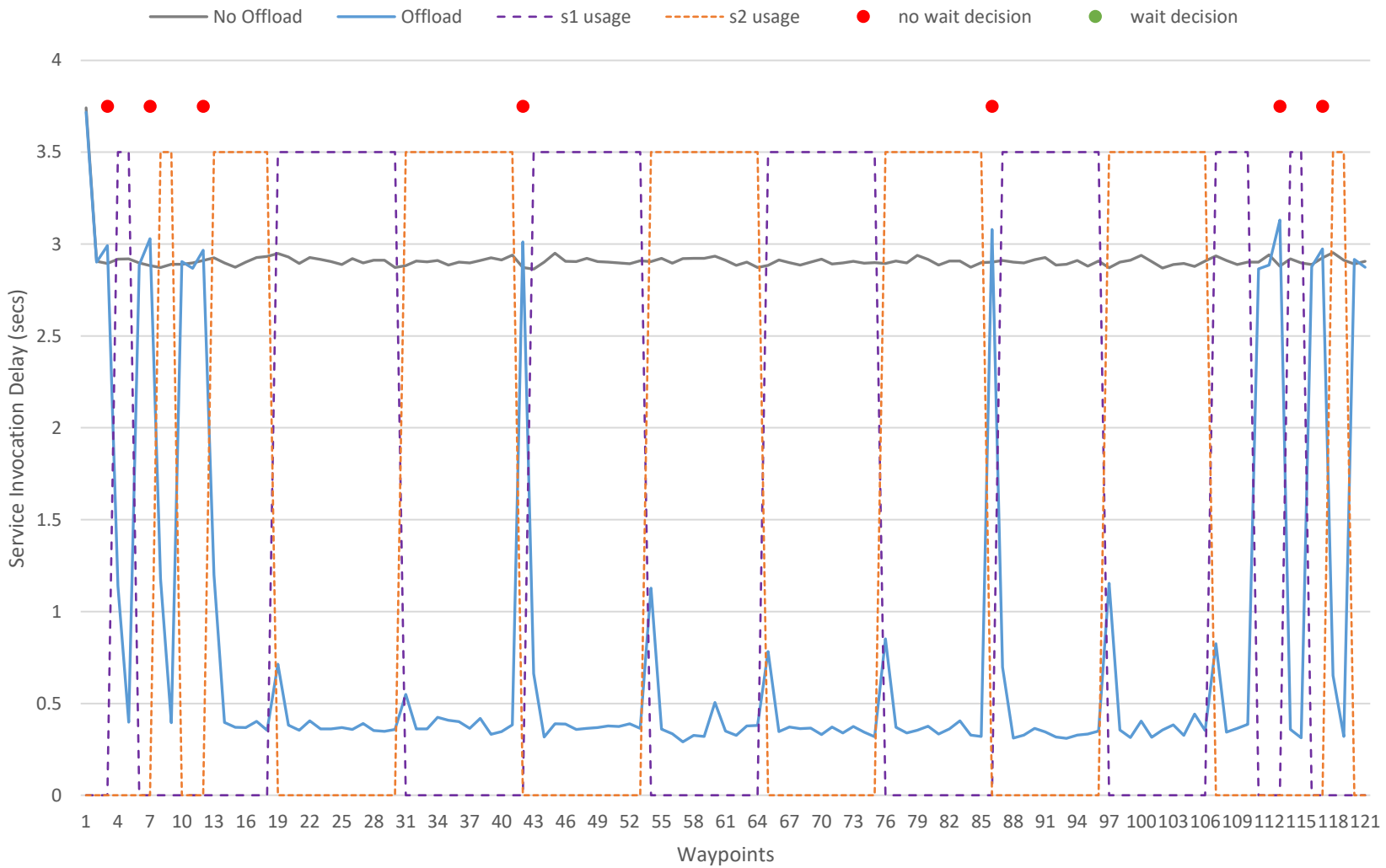
Experiments

- Area of 200×200 square meters
- 121 waypoints set every 20 meters
- s1 and s2 with a communication range of about 100 meters
- Drone moves row-wise from the top to the bottom of the area
- Flying speed 4 m/s



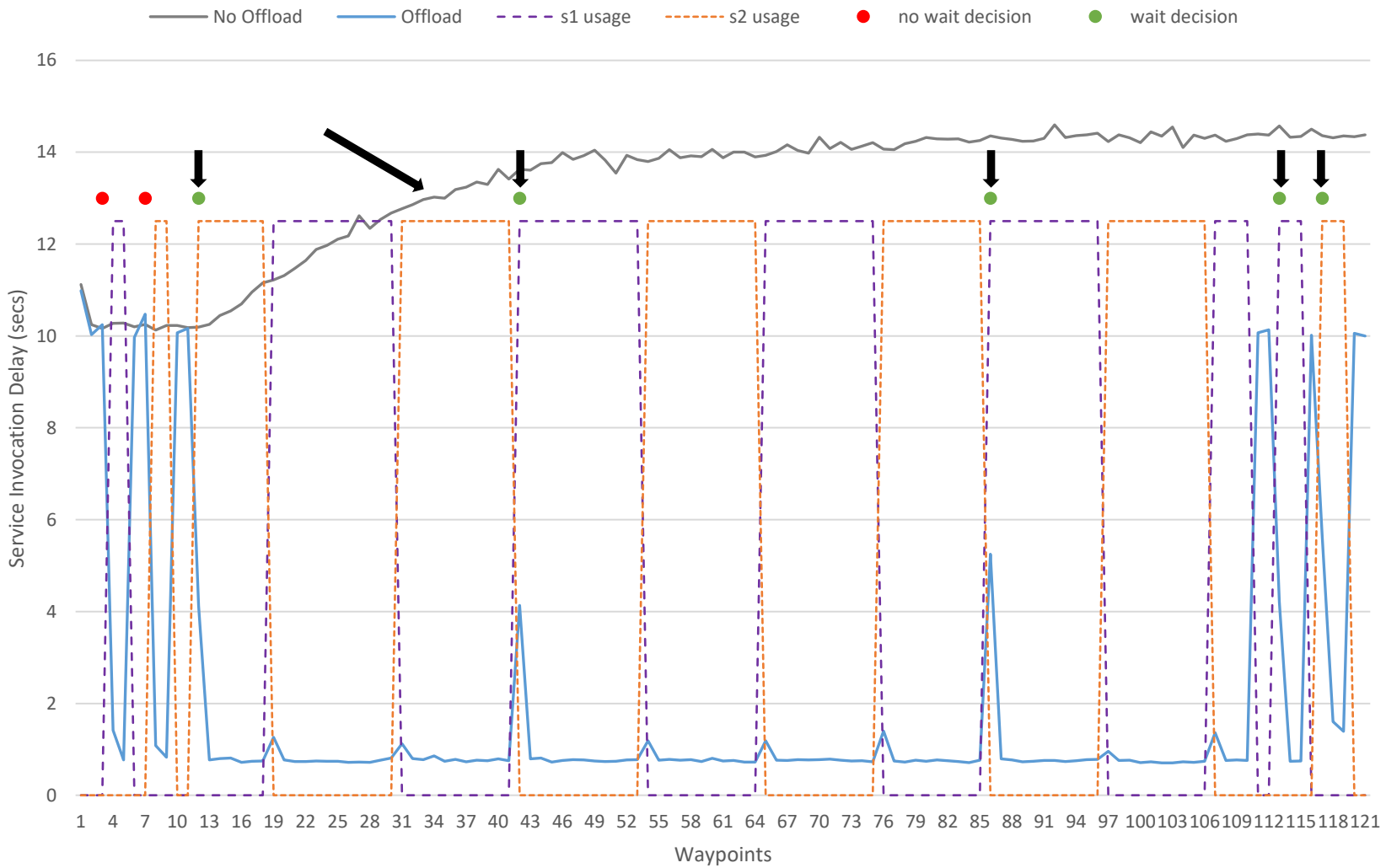
Both servers full capacity – light mode

- Service invocation delay reduced by 73%
- Mission time reduced by 16%
  - 22.2 vs. 26.5 minutes



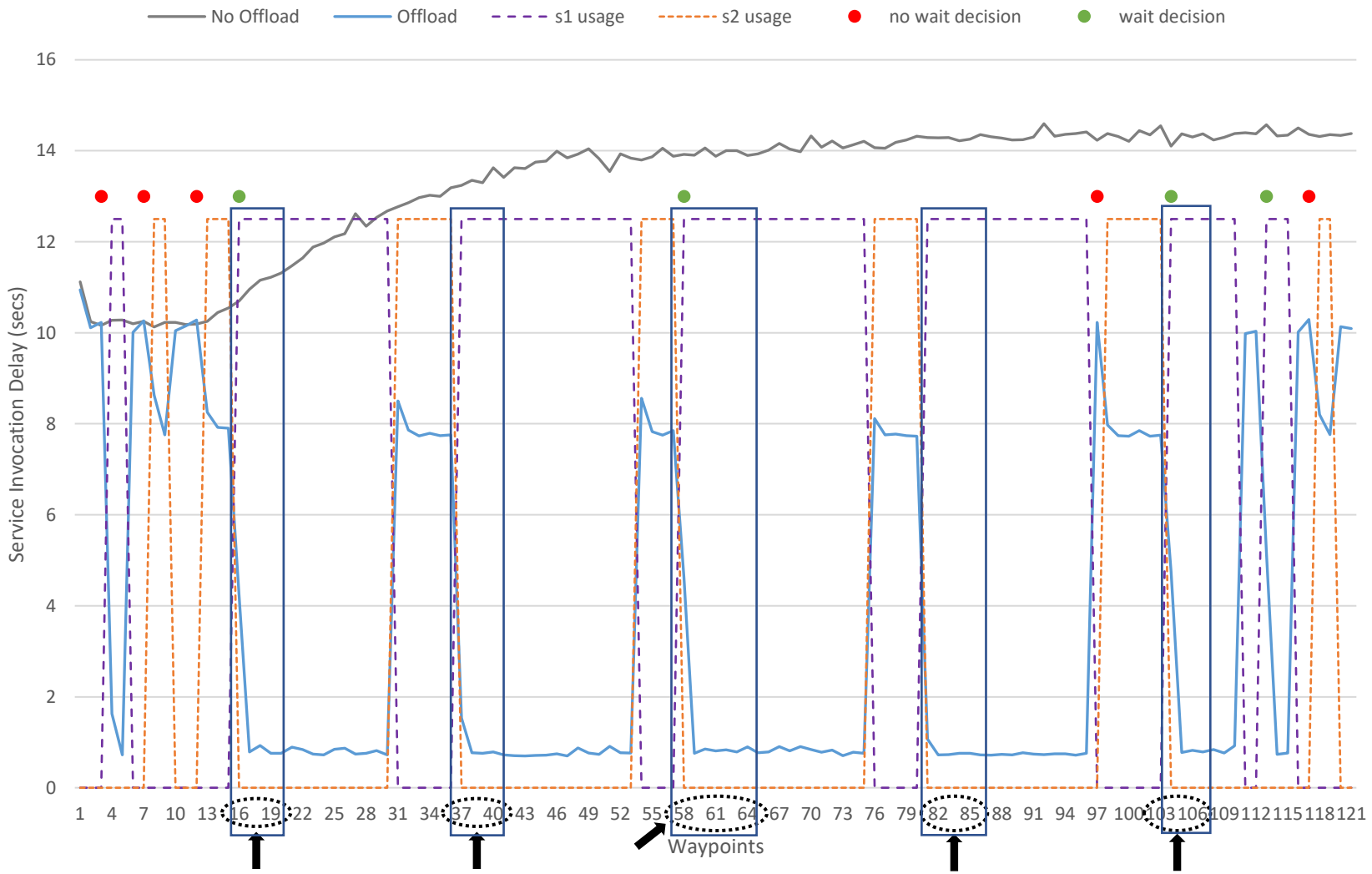
Both servers full capacity – heavy mode

- Service invocation delay reduced by 86%
- Mission time reduced by 48%
  - 24.4 vs. 47.5 minutes



One server loaded – heavy mode

- s2 processes incoming requests with an additional delay of 7 seconds
- Service invocation delay reduced by 72%
- Mission time reduced by 42%
  - 28 vs. 48.5 minutes
- Naïve (equal) server selection would have increased the mission time by +2.5 minutes



Conclusions

- Service-oriented approach for task offloading
- Part of a full software stack for autonomous drones
- Fully transparent for the application program

- Evaluation shows potential for very significant reduction of the mission time
  - increased operational autonomy/range

- Future work
  - more advanced offloading policies, taking into account the service call rate
  - combination with higher-level server allocation and path-planning algorithms

tkasidakis@uth.gr

https://csl.e-ce.uth.gr/projects/pv-auto-scout

Thanks for watching

Feel free to contact us

Auto-Scout

European Union
European Regional Development Fund

HELLENIC REPUBLIC
MINISTRY OF ECONOMY & DEVELOPMENT
SPECIAL SECRETARY FOR ERDF & CF
MANAGING AUTHORITY OF EPAnEK

EPAnEK 2014-2020
OPERATIONAL PROGRAMME
COMPETITIVENESS ENTREPRENEURSHIP INNOVATION

ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Partnership Agreement 2014 - 2020

Co-financed by Greece and the European Union