Integrity ★ Service ★ Excellence

Unmanned Systems Autonomy Services: Overview

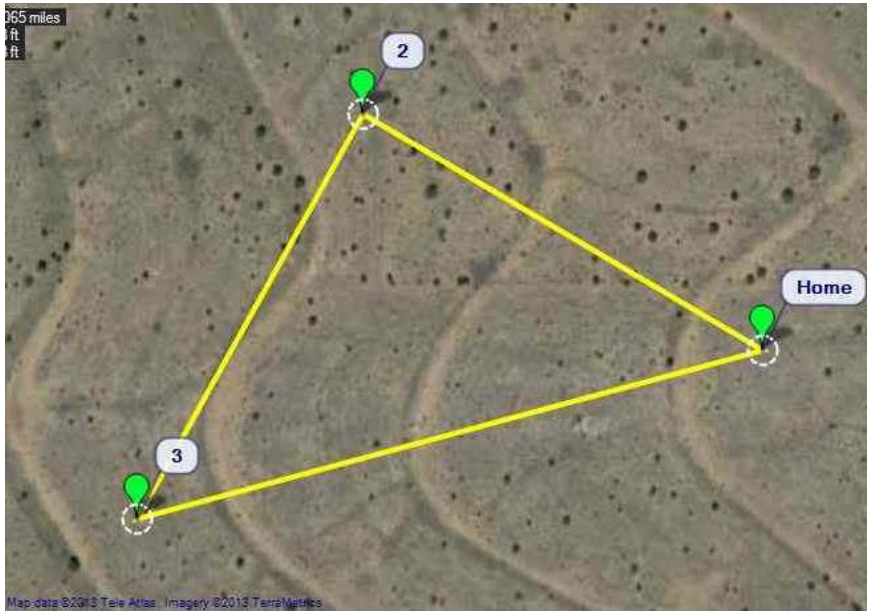
Dr. Derek Kingston
Control Science Center of Excellence
Aerospace Systems Directorate
Air Force Research Laboratory

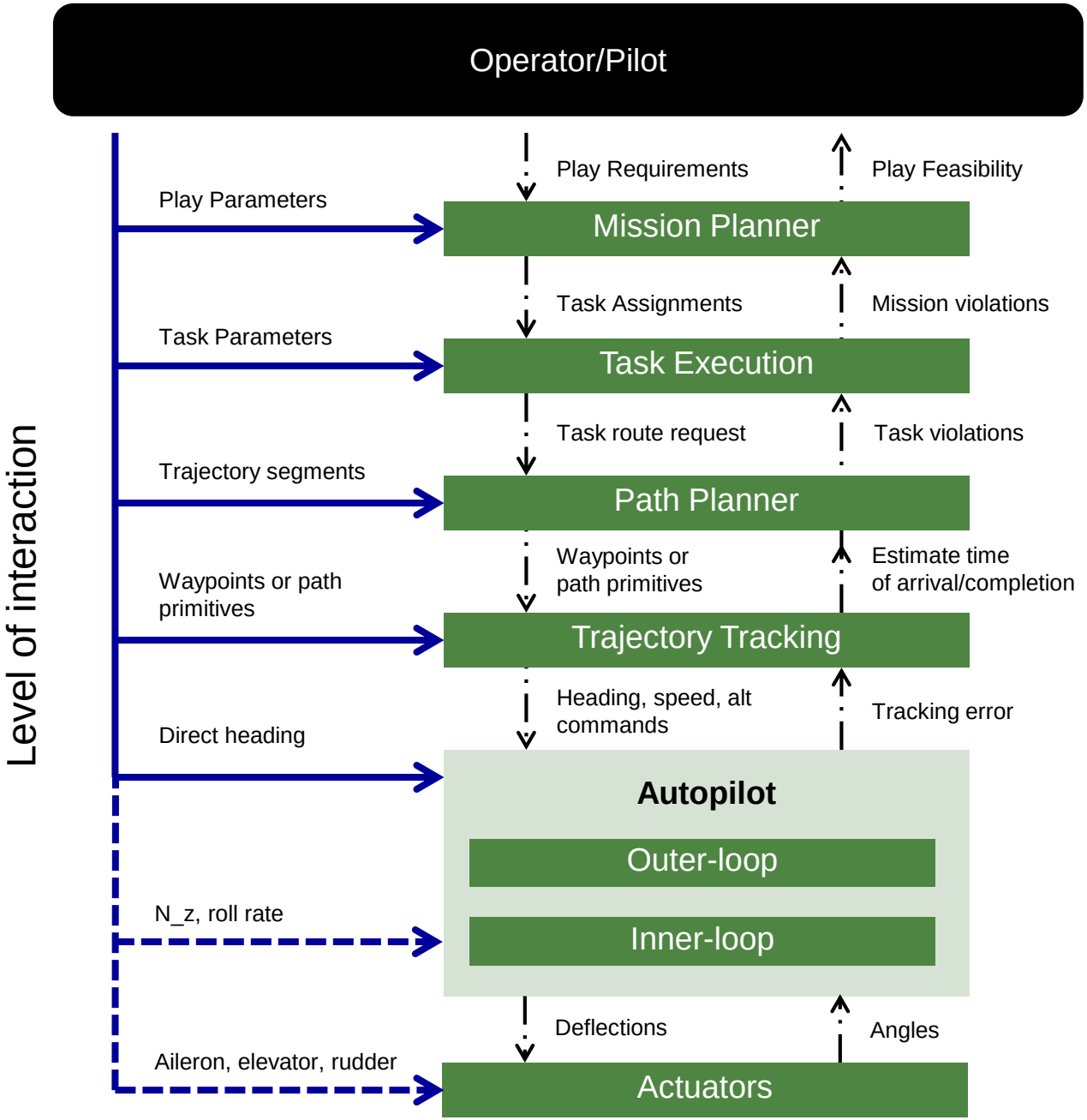
What is UxAS?

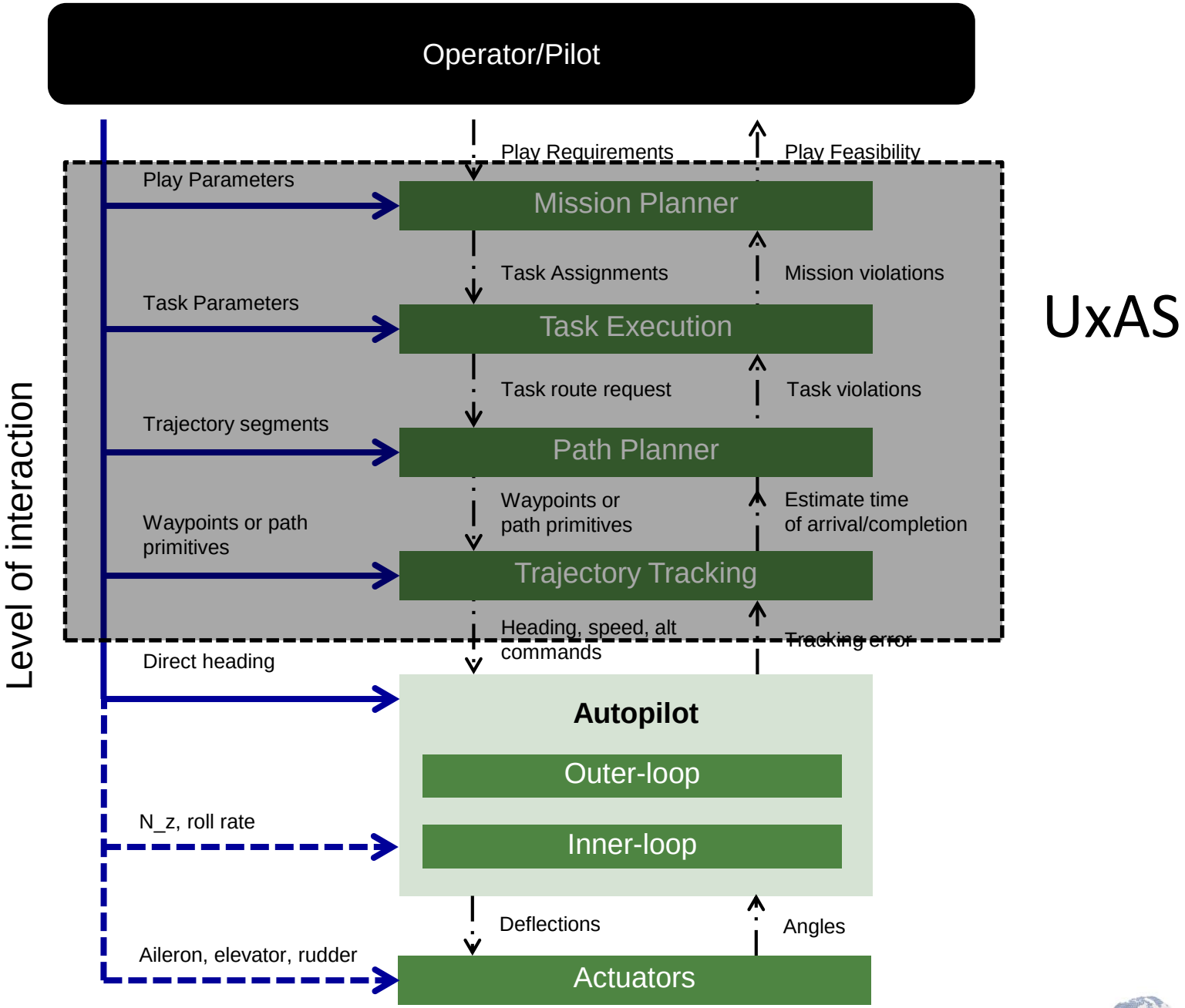
- **UxAS: Unmanned Systems Autonomy Services**
- **Net-centric collection of software modules that interconnect to automate mission-level decision making**
  - **Task assignment**
  - **Cooperative control**
  - **Sensor steering**
- **Used to conduct experiments and demonstrations of cooperative control and human-machine teaming**

Motivation

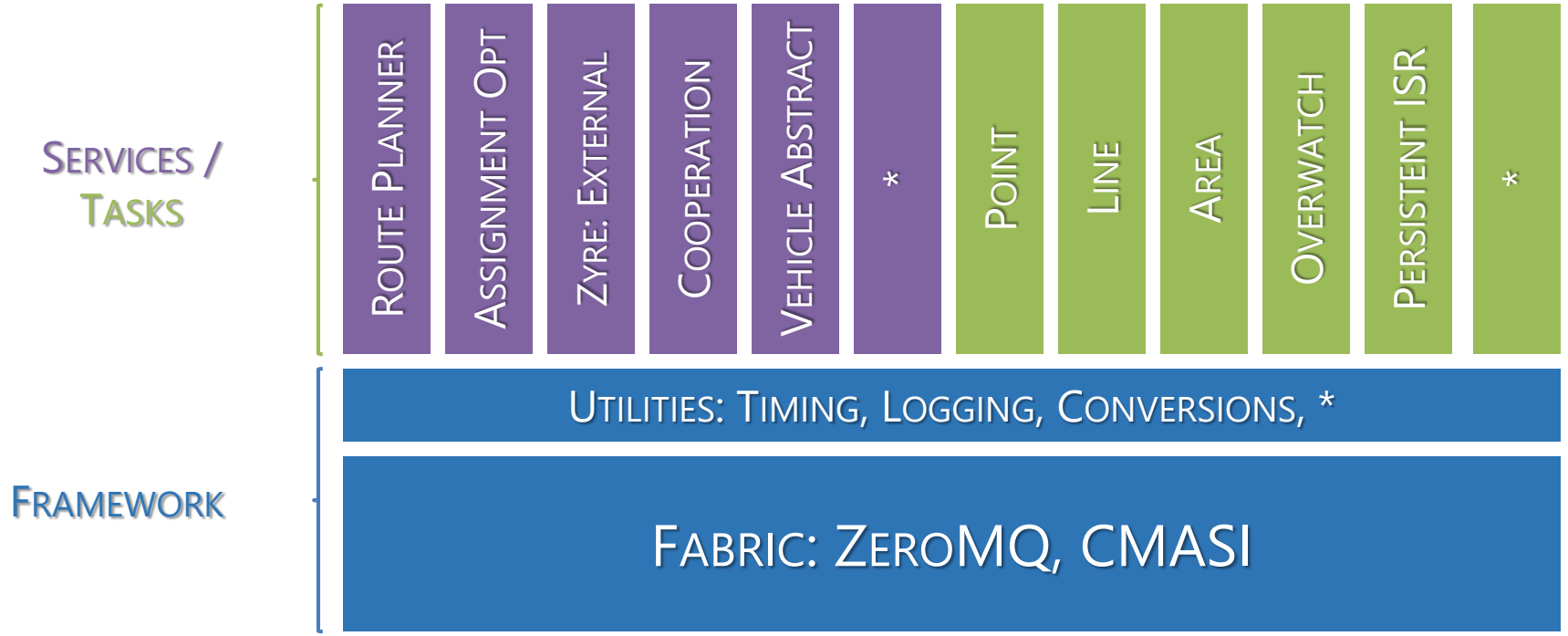
- **Current UAV operations have very low levels of automation: waypoint control often highest level**
- **For multi-vehicle operations, micro-managing each asset will be impractical**
- **Very limited and inflexible contingency routines**
- **Software development costs for inclusion and testing of new capability growing exponentially**



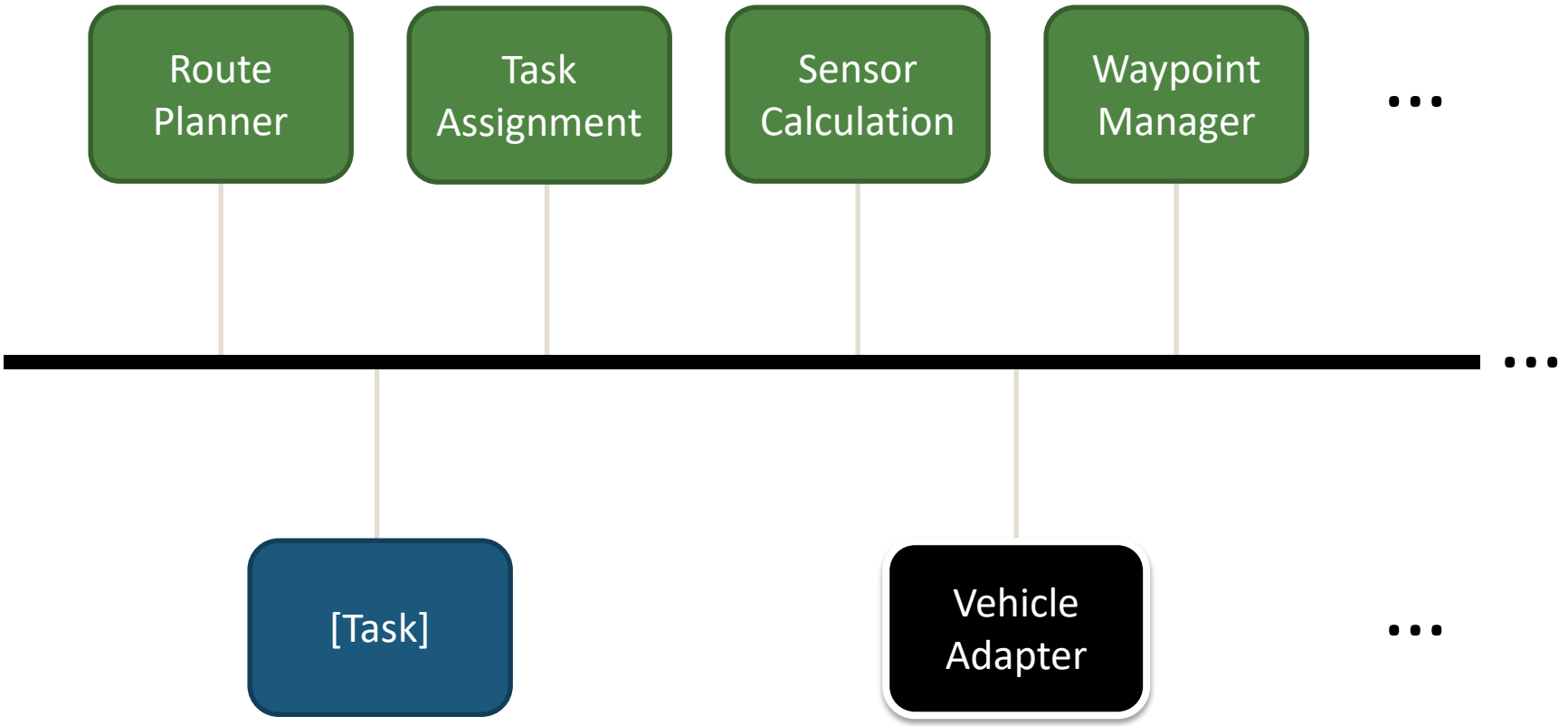

Operator/Pilot
Level of interaction
Play Requirements
Play Feasibility
Play Parameters
Mission Planner
Task Assignments
Mission violations
Task Parameters
Task Execution
Task route request
Task violations
Trajectory segments
Path Planner
Waypoints or path primitives
Waypoints or path primitives
Estimate time of arrival/completion
Trajectory Tracking
Heading, speed, alt commands
Tracking error
Direct heading
Autopilot
Outer-loop
N_z, roll rate
Inner-loop
Deflections
Angles
Aileron, elevator, rudder
Actuators


Operator/Pilot
Play Requirements
Play Feasibility
Play Parameters
Mission Planner
Task Assignments
Mission violations
Task Parameters
Task Execution
Task route request
Task violations
Trajectory segments
Path Planner
Waypoints or path primitives
Estimate time of arrival/completion
Waypoints or path primitives
Trajectory Tracking
Heading, speed, alt commands
Tracking error
Direct heading
Autopilot
Outer-loop
Inner-loop
N_z, roll rate
Deflections
Angles
Aileron, elevator, rudder
Actuators
Level of interaction
UxAS

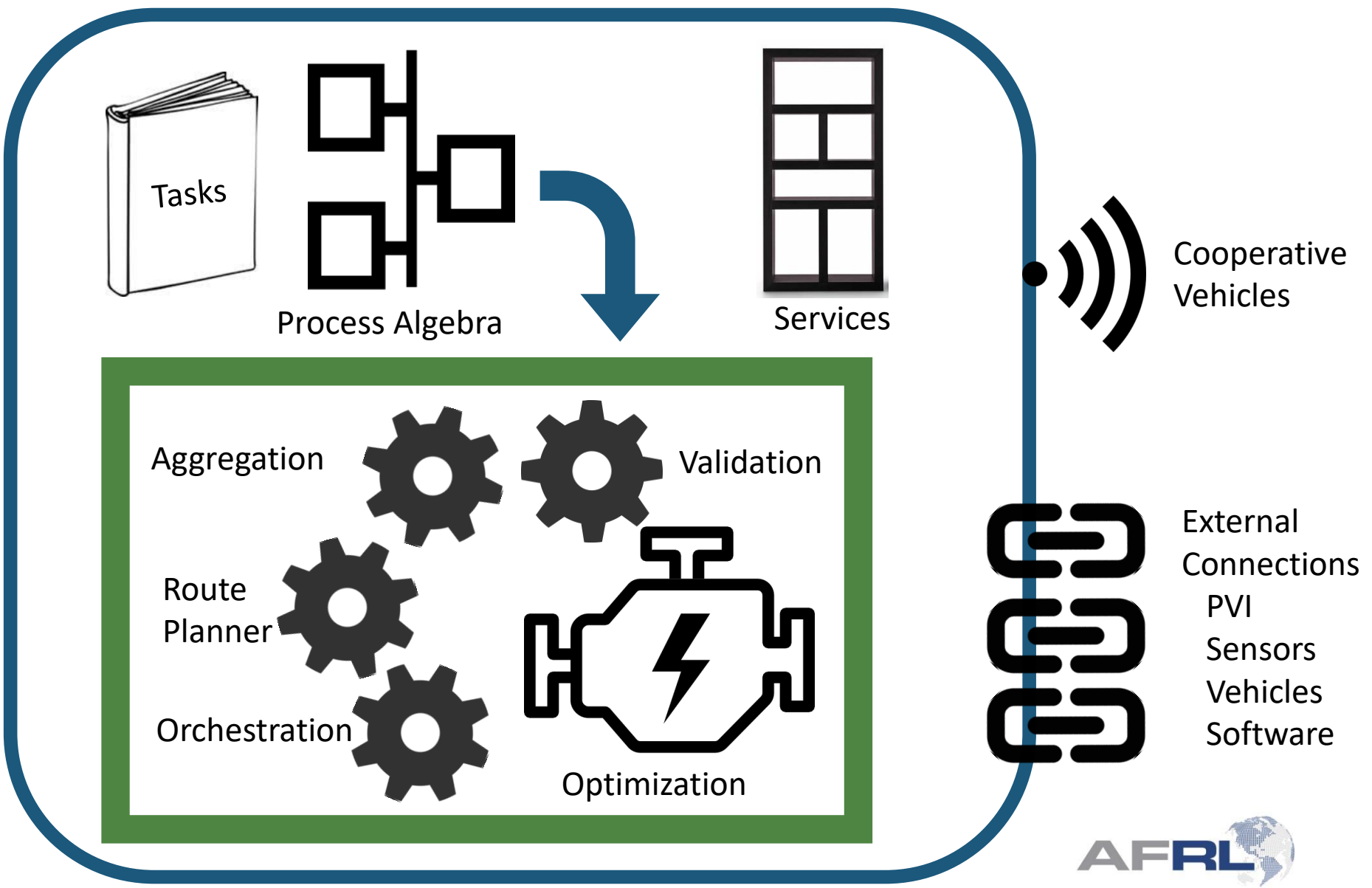
UxAS Architecture



Service-Oriented Architecture

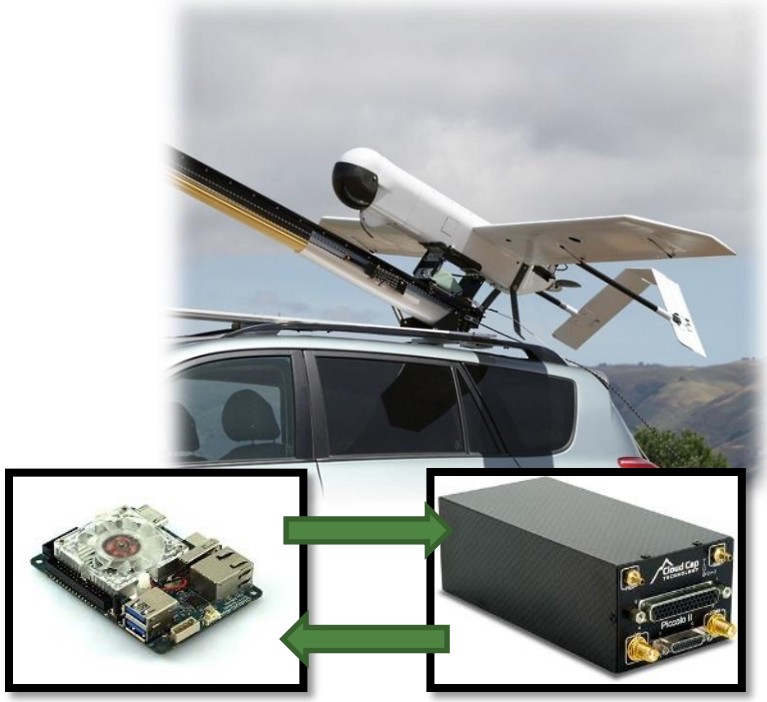


Notional Diagram

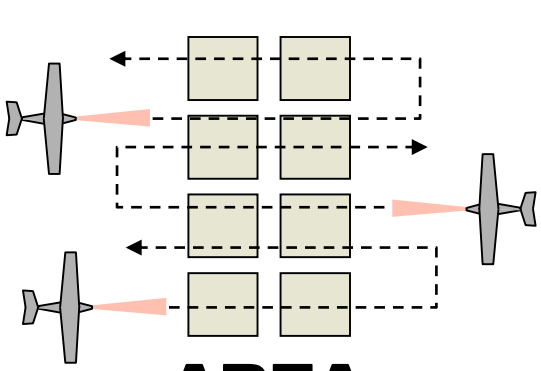


Integration with Vehicles

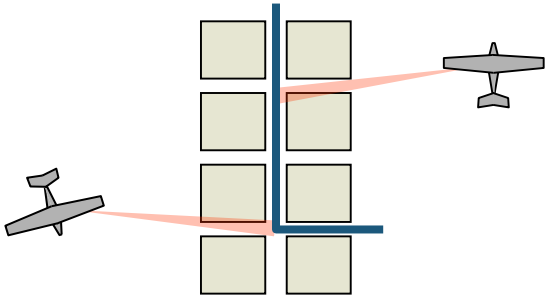
- Capabilities of vehicles are captured by ‘EntityConfiguration’
  - Nominal speed, alt
  - Sensor payloads/capabilities
- Entities are controlled by sending waypoint commands
- Vehicles report current state, including active tasks
- All services are completely independent from vehicle adapter
  - Current hardware adapters: Kestrel, Piccolo, AMASE sim
  - New vehicles simply require vehicle-specific adapter



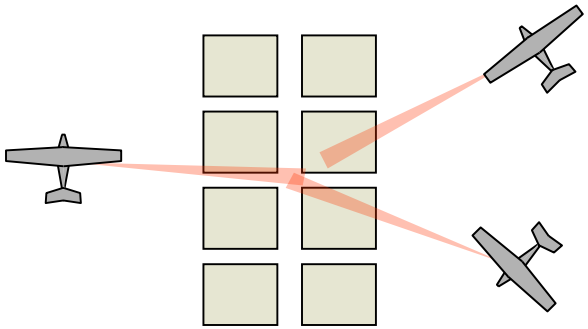
Task Services



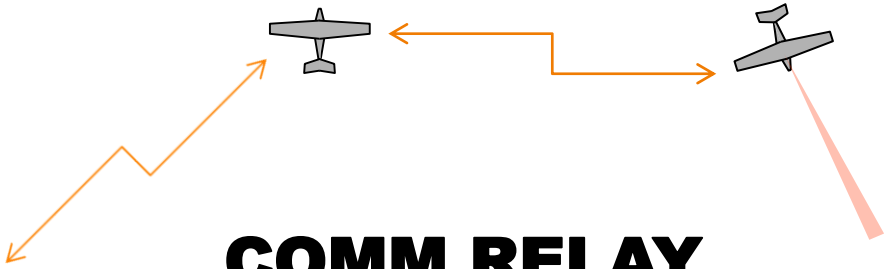
AREA



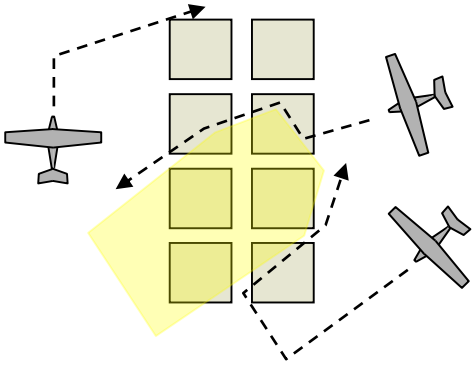
LINE



POINT



COMM RELAY



+ Zone Avoidance

Task Service List

- **Area Search: uniform coverage, selectable angle sweep, selectable coverage patterns**
- **Line Search: match desired camera angle, automatically steer camera during execution**
- **Point Search: obtain imagery of a point from a specified distance and direction**
- **Blockade: intercept approaching entities in cooperative formation to impede progress**
- **Cordon: seal-off intersection by strategically placing vehicles in blocking positions**
- **Escort/Overwatch/Scout: maintain “eyes-on” VIP or enemy**
- **Communication Relay: position vehicle to provide best reach-back communications to others**
- **Communication Transfer: rendezvous with other vehicles and establish positive communication**

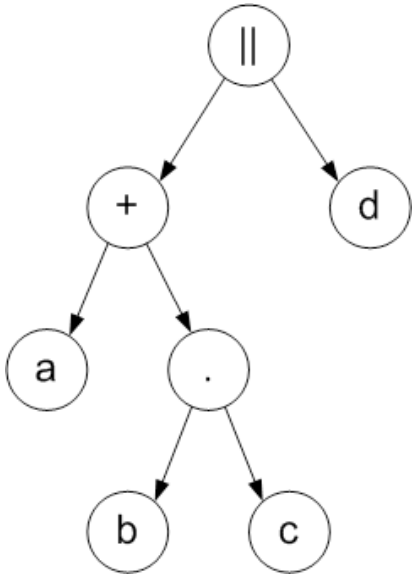
Flexible Task Assignment

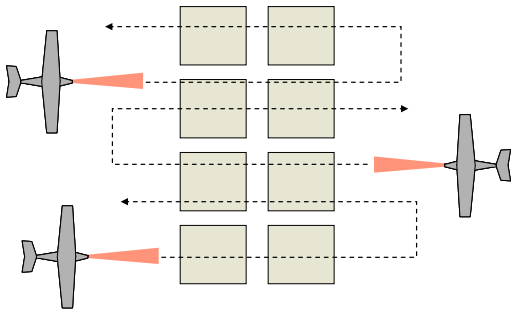
- **Generic Tasks**
  - **Details of tasks are not necessary**
  - **Optimization searches for sequence of tasks that minimize time/fuel**

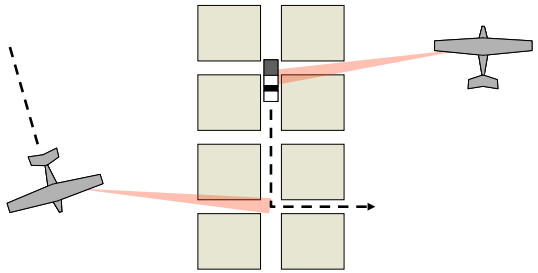
$$\left(\begin{bmatrix} x_b \\ y_b \end{bmatrix}, \begin{bmatrix} x_e \\ y_e \end{bmatrix}, T\right)$$

- ***Process Algebra:*** **language for describing task relationships**
  - **Support for *AND*, *OR*, and *precedence* operators**
  - **Guides optimization to choose feasible possibilities at next step**

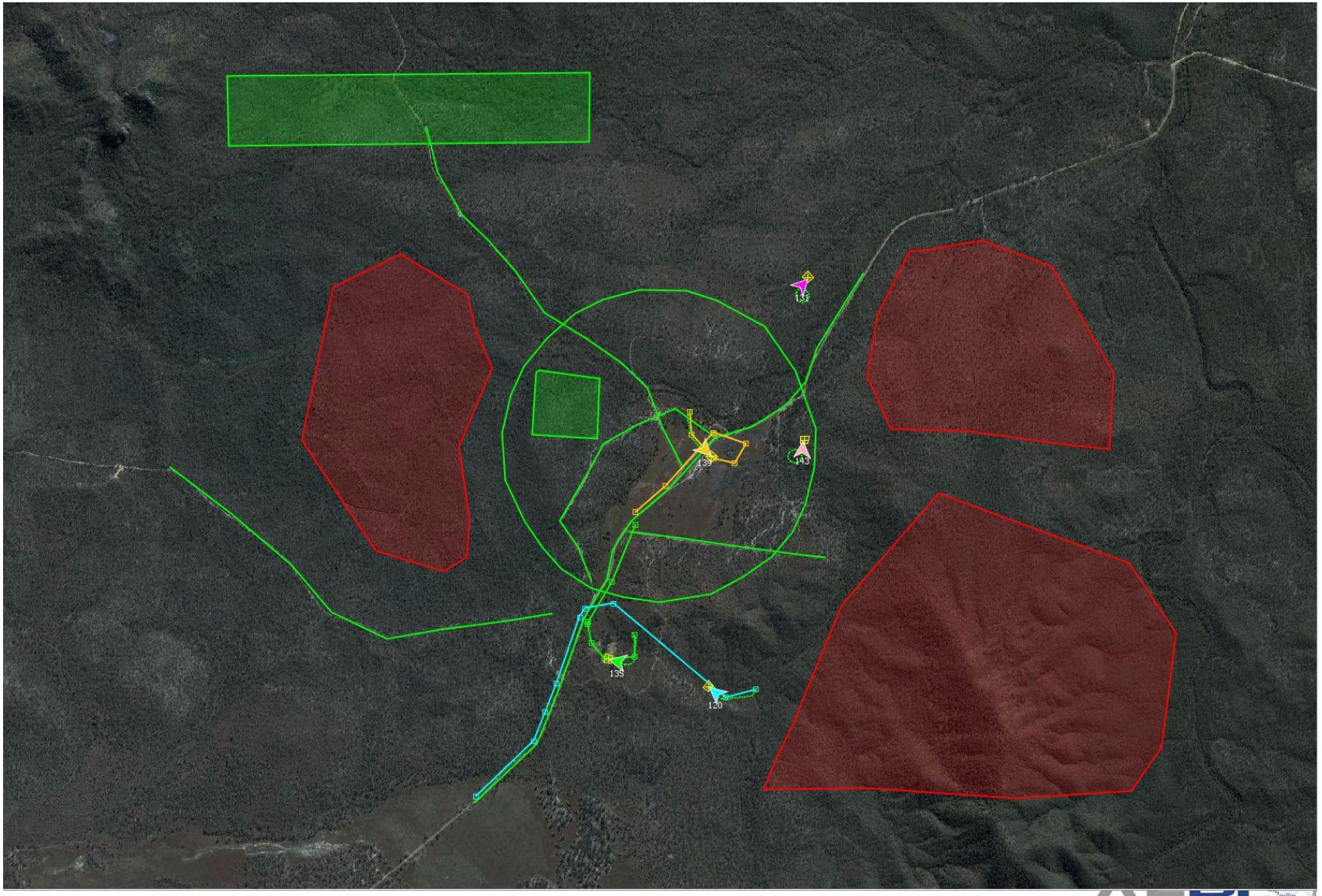
$$(a + b \cdot c) \parallel d$$



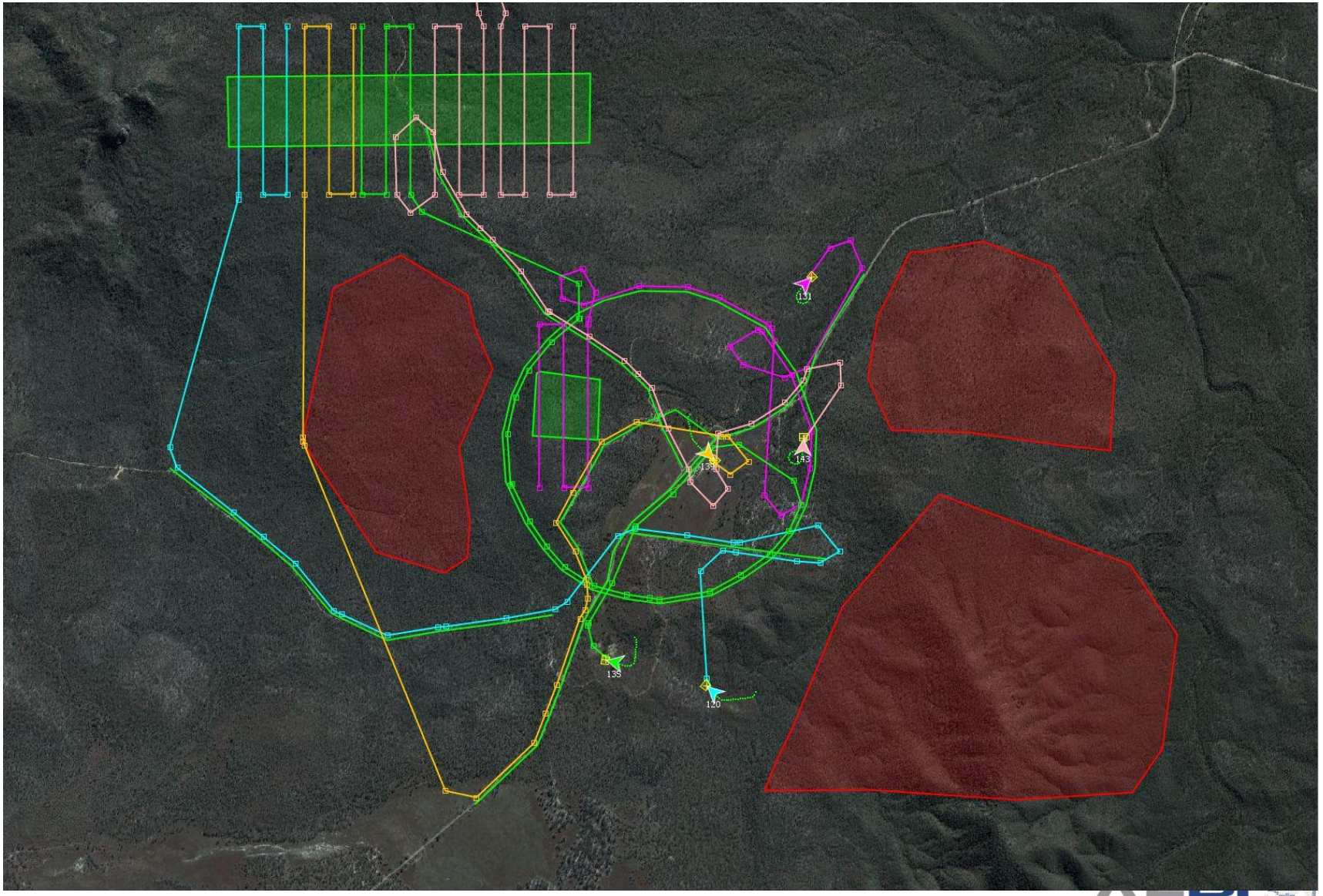




Flexible Task Assignment



Flexible Task Assignment



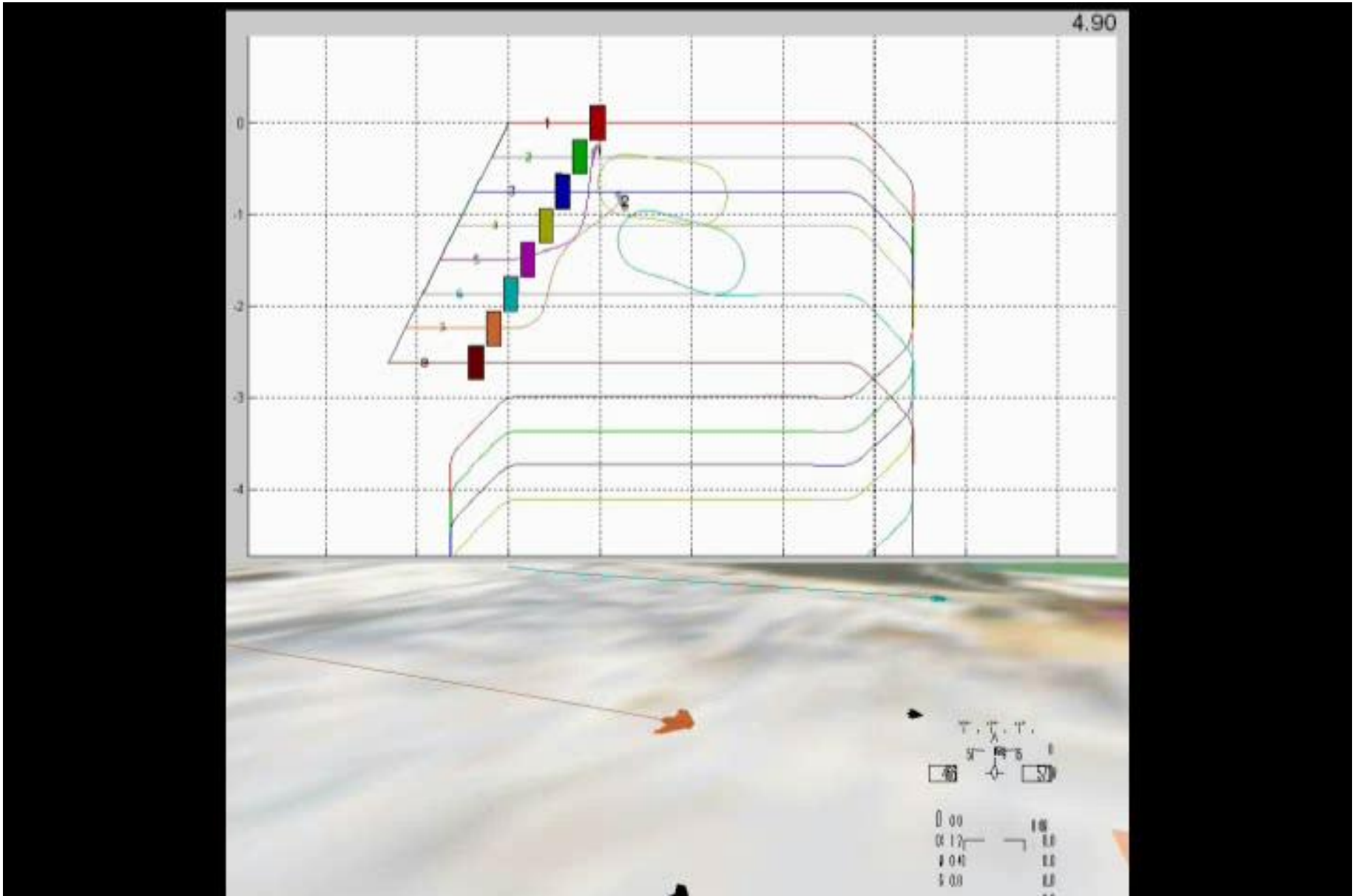
Capability Summary

- **Time-optimal task assignment and scheduling of ISR tasks to multiple heterogeneous vehicles**
- **Decentralized persistent ISR**
- **Fast 2D route planning in complex polygon regions**
- **Automated surveillance patterns: lawnmower, spiral, sector, perimeter/road sweep, fixed heading run-in, stand-off loiter**
- **Cooperative behaviors: comm relay, blockade, cordon, escort**

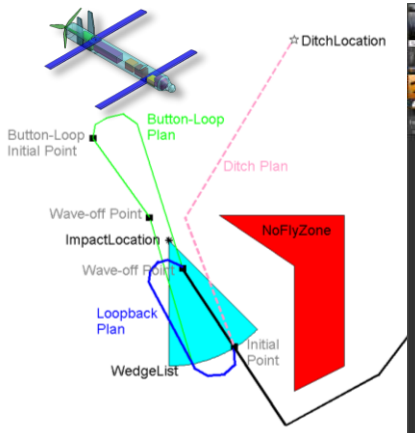
Capability Gaps

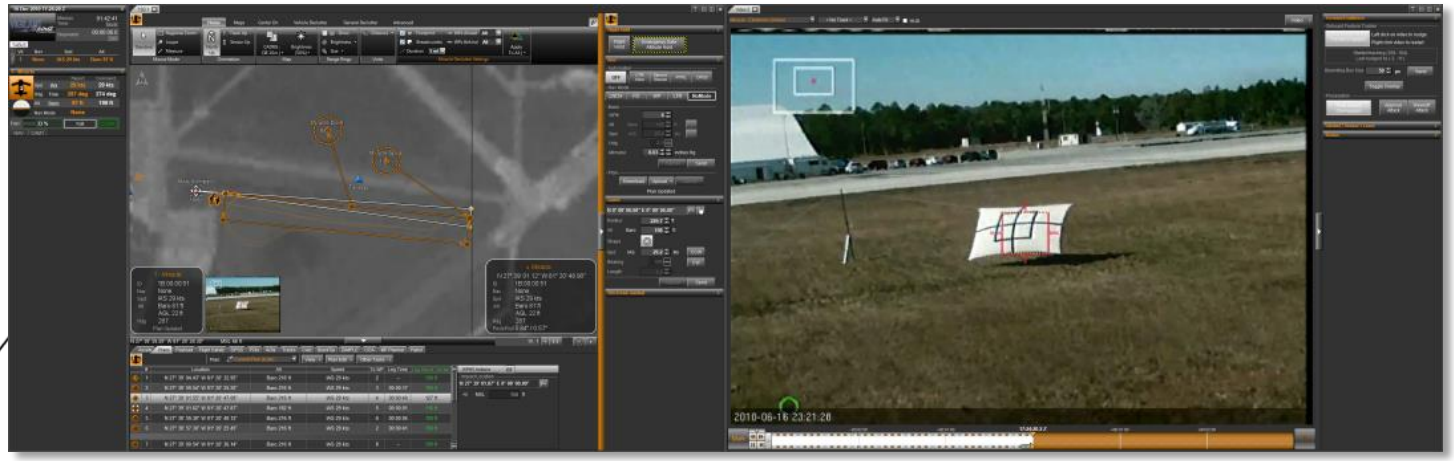
- Exponentially more computationally expensive as number of vehicles and tasks increases → not a long-term solution for swarms
- No sensor processing
  - Lacks any notion of target probability, proper geometries for best sensor looks, track correlation, latency, storage/retrieval
  - How does this impact mission-level decisions?
- Limited environmental awareness
  - Changing winds, communication SNR, vehicle health
- Motion planning is simplified, no aggressive maneuvers, no 3D plans
- Collision avoidance is not considered, assumes altitude separation
- Strict mission timing constraints not supported
- Poor support for additional optimization: threats/fuel/sensor/comms
- Not all Tasks respect airspace constraints
- Messaging is custom, no adherence to standards (e.g. OMS)
- Not designed for cyber security
  - Timing, separation, crypto, deadlock, memory safety

Example
Wide Area Search and Destroy



BlackTip UAV Weapon





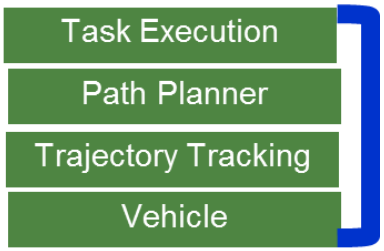
- Close Air Support mission vignette
- Single operator control of mothership and all payloads including RW BlackTip
- Target engagement algorithms for both GBU-12 and an air launched UAV weapon
- Flight tested using surrogate aircraft
- RB contributions: situation-specific routing algorithms and flexible options to support user (e.g. wave-off maneuver)

High-Level Tasking & Detailed Adapting of Plays

High-level plays for most interactions: Operator makes 2-3 voice and/or manual inputs

target size, visibility, optimization, & priority

Autonomy specifies any missing parameters

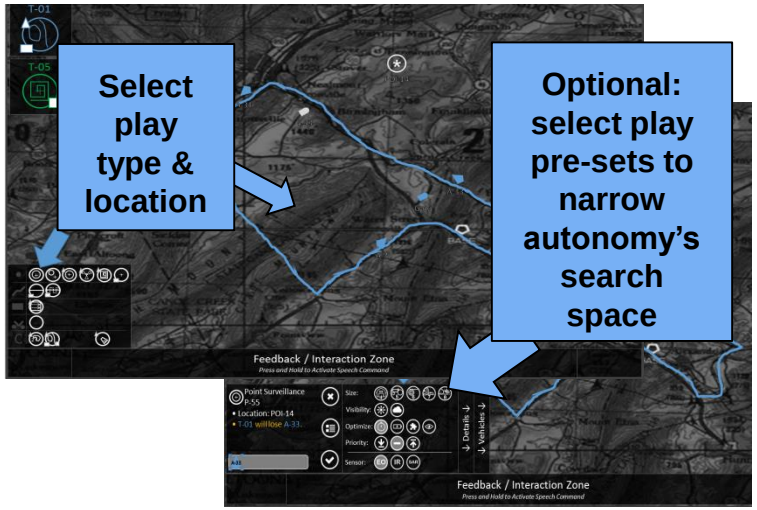


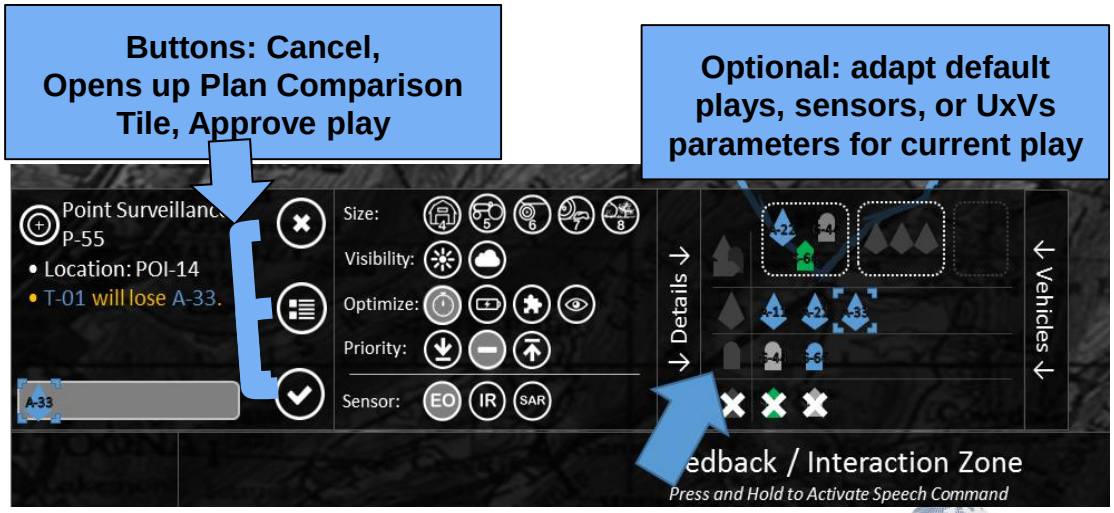
Detailed specification/task tailoring

For example: asset type, specific vehicle, vehicle specs (altitude, speed, approach, standoff), sensor details, orbit type, search pattern, etc.

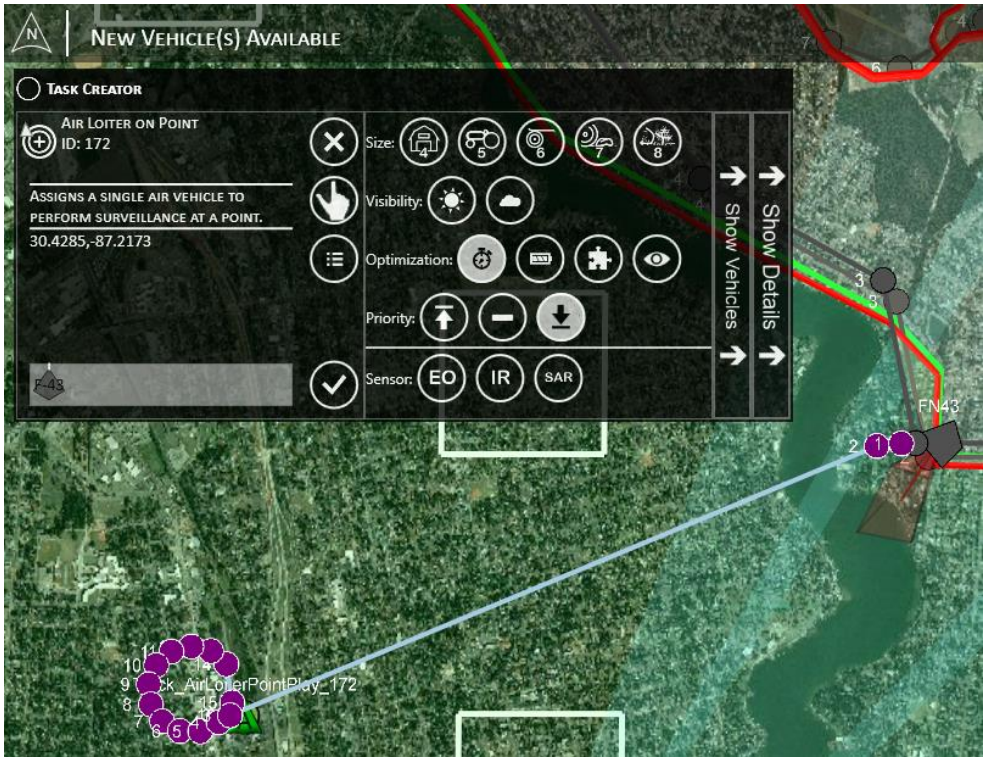
Plays parameters can be adapted both:

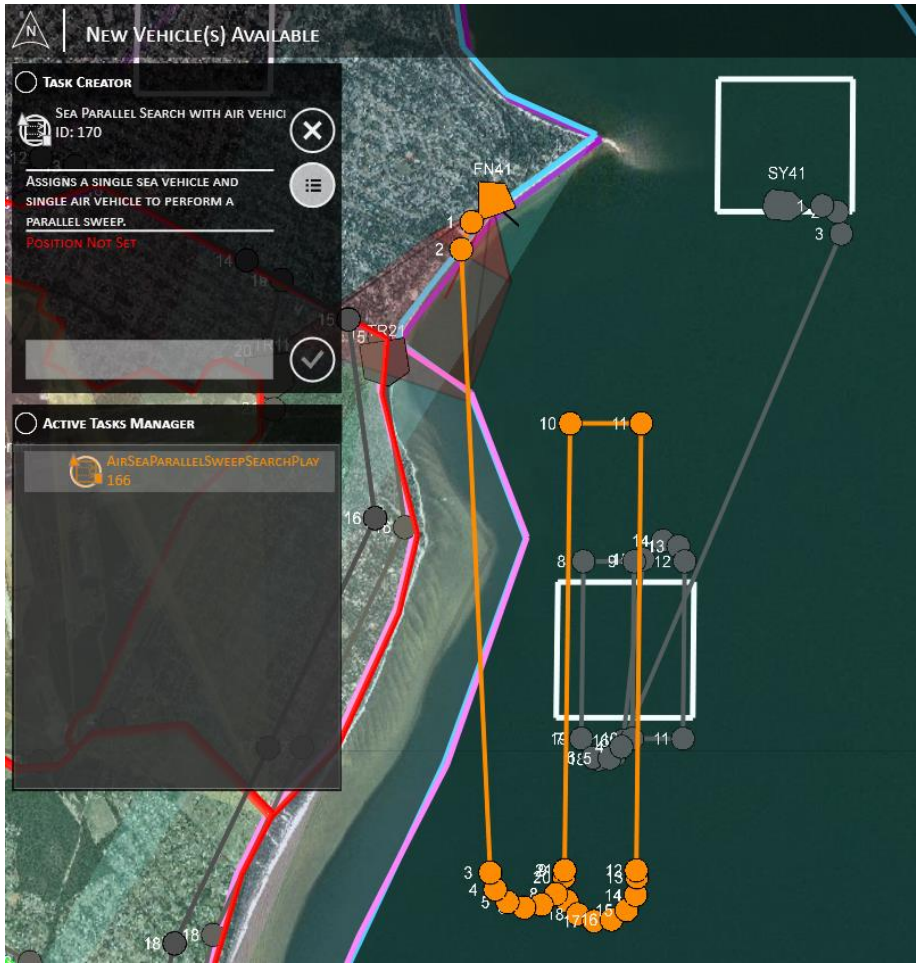
- during play calling
- during play execution





Play Calling

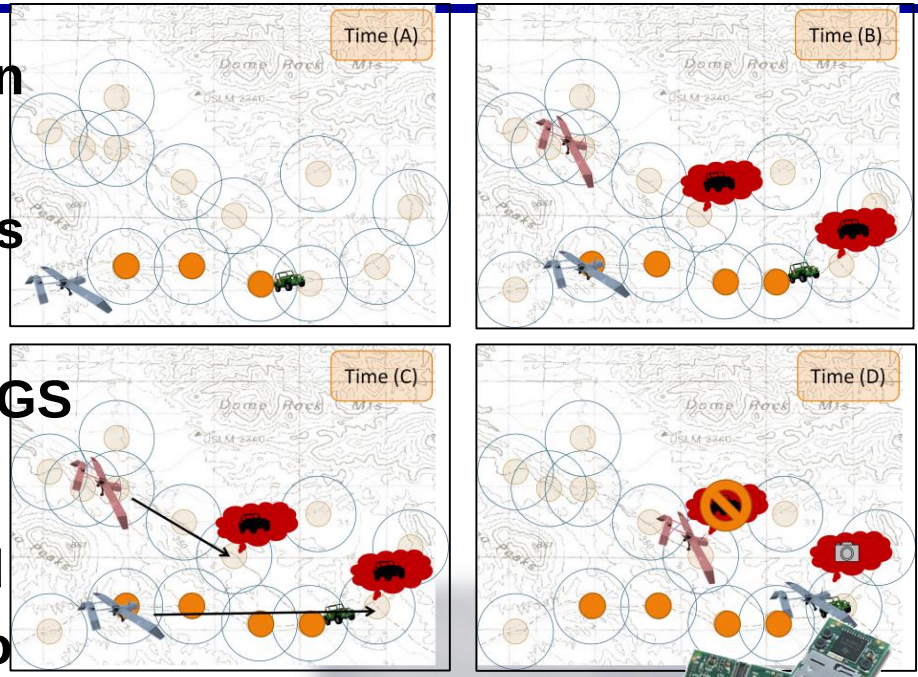


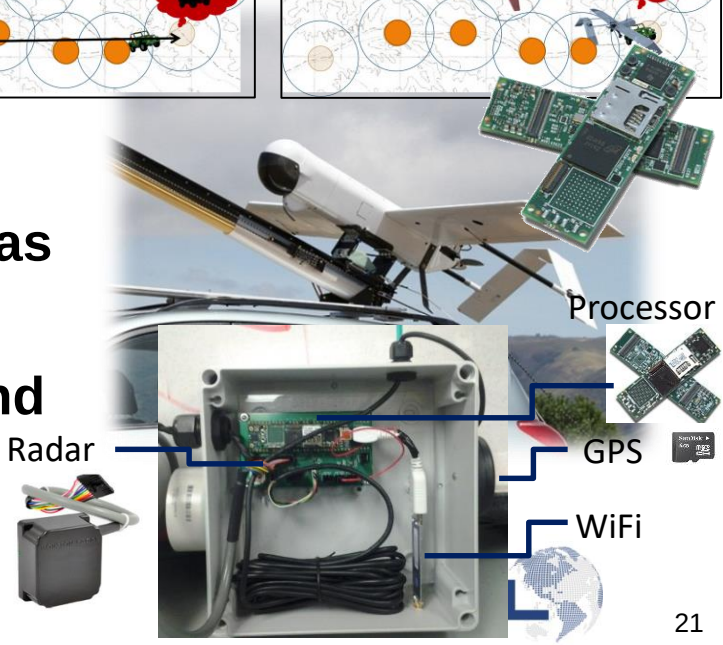


- **Agility: enable rapid ID & management of complex situations and uncertainties that can disrupt or degrade a team's ability to successfully complete their missions**

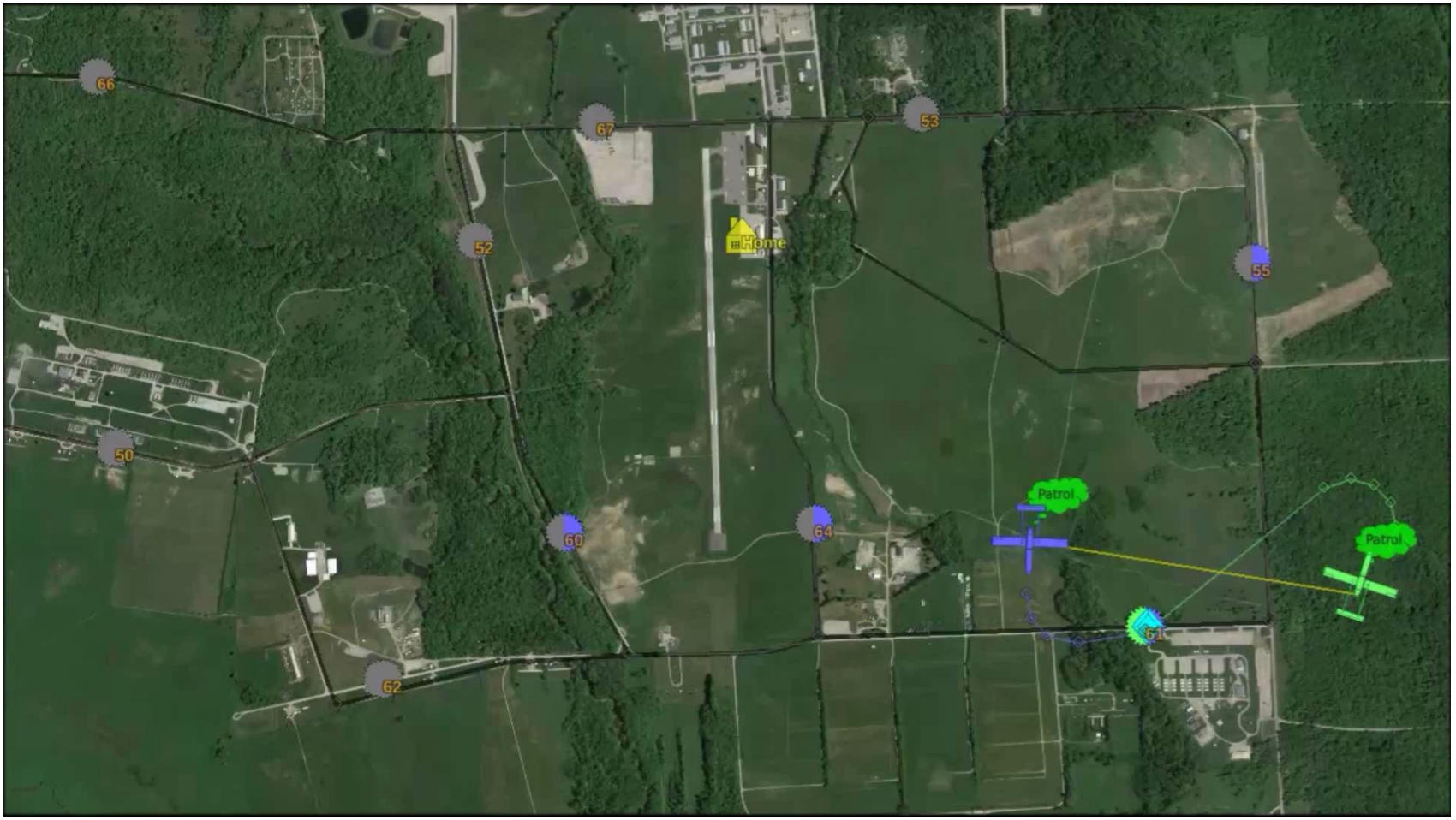
UAV Autonomous Control Demo at Military Training Exercise

- Objective: UAV **autonomous** decision making in **communication degraded** environments fusing UAVs and UGSs
- OSD Unmanned Systems Roadmap: **support predictive movements** via UGS laydown and UAV oversight
- From concept development, detailed analysis, and journal publications, to simulation, engineering, test, and demo
- Demonstrated **onboard, task-driven control** as tech element of military exercise (TS13). Guided by Intel officers, deployed system and successfully interrogated UGS from UAV for **troop movement prediction and imaging**





UxAS Example Capability: Field Test Decentralized Control, Nov 2015



UxAS Example Capability: Field Test Decentralized Control, Nov 2015

Summer of Innovation

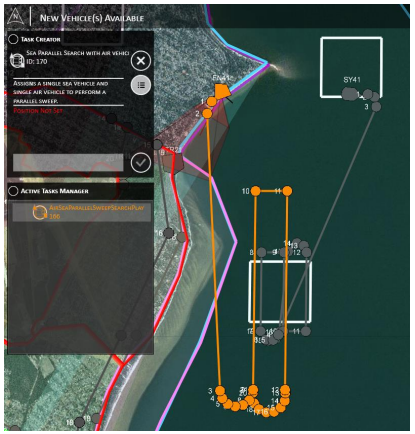
- Industry, academia, government participants (~50 people across 16 organizations)
- Using open-source version of UxAS and AMASE
- Collaborative effort to apply Formal Methods techniques
  - Formalized requirements
  - Formal architecture description (AADL)
  - Methods for proving correct behavior (safety properties)
  - Cyber-security considerations
    - Real-time scheduling, provable separation, crypto
  - Automated test generation and continuous integration
  - Argumentation for modularity
  - Run-time assurance
  - Hybrid systems analysis
- Results and future directions presented at special session of S5 conference

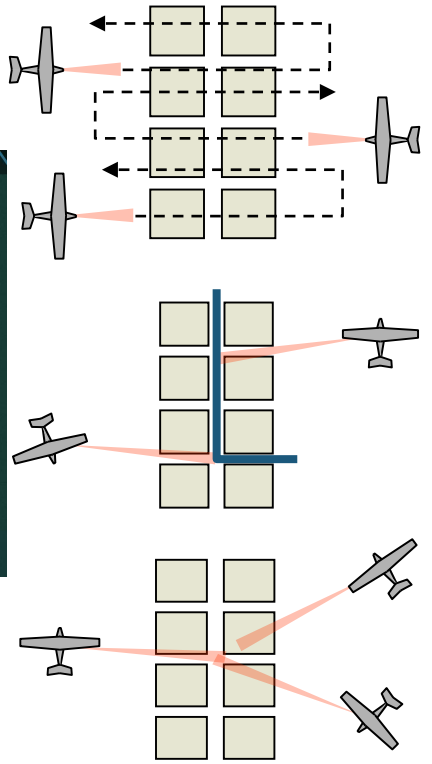
Summer of Innovation: Results

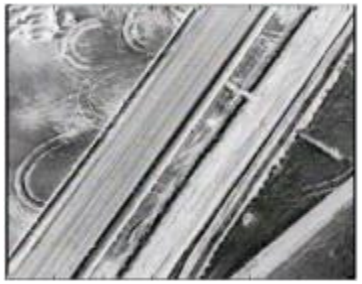
- Formal architecture description of core UxAS services
  - Captured in AADL and ASSERT
- Enforcement of real-time scheduling limits
- Provably safe collision avoidance algorithm (DART)
- Run-time assurance prototype
- Statistical and specification driven testing
  - DEMETER, and S-TaLiRo
- Steps toward cyber-resiliency
  - Critical services proven implementation
  - “Trusted Build” on memory-safe operating system (seL4)
  - Support for Rust
- Synthesized behaviors from formal specifications

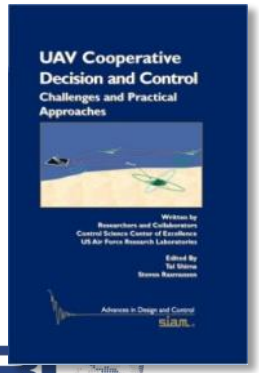
UxAS: Unmanned Systems Autonomy Services

- **Collection of software modules that interconnect to automate mission-level decision making**
  - **Task assignment**
  - **Route planning**
  - **Cooperative control**
  - **Sensor steering**
- **Used to conduct experiments and demonstrations of cooperative control and human-machine teaming (live and simulated)**
- **21 test events, 88 sorties, 150 flight hours**
- **Draws upon nearly 20 years of basic research in UAV cooperative control**
- **Designed for flexibility, rapid extensibility**
- **Open-source**
  https://github.com/afrl-rq/OpenUxAS









Questions?