

04 System Architecture

Icarus Orbital Systems — System Architecture

The technical architecture · three components · operational model. Source for all pitch content.

Three-component fleet architecture



Syntax error in text
mermaid version 10.9.5

Component A: Primary Tele-Operated Tug (PTOT)

Role: Command-and-control hub.

Attribute	Spec
Operation mode	Human-supervised + AI-assisted
Crew	None (uncrewed; ground-controlled)

Attribute	Spec
Function	Coordinates Dragonfly fleet · stages captured material · interfaces with OPS
Direct debris contact	No — directs, doesn't capture
Loss profile	Centralized · not in contact zone

Component B: Dragonfly DF-1 drone flotilla

Role: Active debris interaction.

Attribute	Spec
Configuration	Modular · distributed fleet · 6-DOF maneuverability
Propulsion	Hall-effect (xenon) primary + hydrogen peroxide RCS thrusters
Capture systems	Articulated end-effector (stable targets) + deployable nets (poly-mylar + carbon-strand for tumbling fragments)
Target range	1U CubeSat → large fragments
Resilience	Distributed — loss of one unit = degraded ops, not failure

Component C: Orbital Processing Stations (OPS) [Phase 2]

Role: Sort · refine · stage recovered material for in-orbit reuse or controlled deorbit.

Attribute	Spec
Location	ELEO (Equatorial Low Earth Orbit) · 200-2000km altitude · $\pm 5^\circ$ latitude
Why ELEO	Geometry minimizes inclination changes between debris-dense zones

Attribute	Spec
Output	Recovered metals · composites · fed back to in-orbit supply chain
Phase	2 (after demonstration mission validates capture economics)

Operational model — Orbital Dredging



Syntax error in text
mermaid version 10.9.5

Each step is independently revenue-generating. Phase-1 economics (debris removal) close before OPS comes online.

Why distributed beats monolithic

Architecture choice	Pro	Con
Distributed Dragonfly fleet (chosen)	Loss of 1 unit = degraded ops · scalable · parallel work	More units to manufacture · coordination overhead
Single large tug (alternative)	Simpler ops · one mission · centralized control	Single point of failure · serial work · risk concentration

Distributed wins for resilience + parallelism. Coordination overhead solved via PTOT + AI-assist.

Why tele-op + AI assist beats fully autonomous

Architecture choice	Pro	Con
Tele-op + AI assist (chosen)	Insurance approval · regulatory posture · human-in-loop trust	Slower than fully autonomous · ground crew cost
Fully autonomous	Faster cycles · no comms latency	Insurance · regulatory · fail-mode liability open question

Tele-op + AI gets us to operations sooner. Fully autonomous is a Phase-3 transition, not a Day-1 decision.

Cross-references

- [\[\[01_strategy\]\]](#)
- [\[\[02_action_plan\]\]](#)
- [\[\[../IOS_03_System_Architecture\]\]](#) — full technical spec source
- [\[\[../IOS_05_Technology\]\]](#) — propulsion + capture details
- [\[\[../Dragonfly Design Project/Dragonfly_Drone_Concept_Design\]\]](#)