

A PHYSICS-CONSTRAINED DECISION FRAMEWORK FOR ORBITAL-DEBRIS INTERVENTION SEARCH

The Orbital Intervention Compiler (OCIE): Concept Architecture, Mathematical Formulation, Verification Plan, and Prior-Art Boundaries

Mahib Sadman

Technology Enthusiast and Business Professional | Independent concept proposal

27 October 2026 | Version 0.1 | Research proposal, not validated experimental research

Document status and integrity statement

This revised document presents the Orbital Intervention Compiler (OCIE) as a testable research programme, not a proven invention. The proposed candidate contribution is a traceable, uncertainty-aware intervention certificate: every candidate portfolio is accompanied by a replayable evidence record containing its counterfactual baseline, physical-feasibility checks, uncertainty envelope, side-effect accounting, constraint outcomes, and explicit abstention conditions. This specific integration is a novelty hypothesis requiring prior-art review; it is not claimed to be unprecedented.

The proposal draws on publicly documented work in orbital-debris mitigation, active removal, automated design, trajectory optimization, uncertainty quantification, and multi-objective decision support. The acronym and title do not imply novelty. Any claim of inventorship, exclusivity, or patentability requires a documented prior-art search and qualified legal review.

Authorship note: Mahib Sadman is listed as the proposing author of this document. This does not establish that every underlying method is new or that no other person has conceived similar work. The author should verify all citations and technical assumptions before external submission.

Abstract

Orbital debris threatens spacecraft through impacts, collision cascades, mission disruption, and growing avoidance demands. Existing mitigation includes post-mission disposal, drag devices, active removal, tracking, collision avoidance, and regulation. This paper proposes the Orbital Intervention Compiler (OCIE), a research framework for generating and evaluating candidate interventions under orbital-mechanics, safety, uncertainty, cost, and policy constraints. Rather than asserting a new physical removal device, OCIE frames intervention design as a constrained search and verification problem. Candidate actions may include evaluating a disposal trajectory, prioritizing tracking, scheduling an avoidance analysis, assessing a servicing option, or recommending no action when evidence is insufficient. A formal optimization model is presented for expected risk reduction, lifecycle cost, intervention side effects, uncertainty, and robustness. The paper distinguishes established physical laws and documented technologies from proposed architecture and untested hypotheses. It provides a reproducible validation plan, baseline comparisons, illustrative diagrams, cost-estimation methods, a staged roadmap, and an intellectual-property discussion. No performance result is claimed because the framework has not yet been implemented or experimentally evaluated. The central research question is whether a transparent,

physics-constrained search process can identify intervention portfolios that improve a defined debris-risk metric over current planning baselines. Novelty remains unverified and must be assessed through a systematic prior-art search.

Index Terms—orbital debris, active debris removal, constrained optimization, automated engineering design, uncertainty quantification, collision risk, decision support, reproducibility.

Executive summary

OCIE is not a proposed magic cleaner. It is a decision-support workflow intended to help engineers compare what to do, where, when, and why. Existing debris-removal systems focus on particular physical mechanisms; OCIE would compare candidate interventions with shared objectives and explicit uncertainty accounting. The first prototype can be software-only, using public orbital data and simplified propagators. Any operational recommendation would require validated data, expert review, and compatibility with established flight-safety processes.

The defensible contribution is a research hypothesis: a transparent compiler-like pipeline may make intervention search, feasibility screening, and evidence tracking more reproducible. Optimization, simulation, orbital propagation, active removal, and automated design are established areas and are not claimed as new. The project succeeds only if an implementation outperforms credible baselines on held-out scenarios without increasing unacceptable risk.

Claims and non-claims

Category	Statement
Claim	OCIE is a proposed architecture and testable research plan.
Claim	The objective and verification protocol can be implemented and evaluated.
Non-claim	OCIE is not proven unprecedented or patentable.
Non-claim	No NASA endorsement, invitation, funding, or approval is implied.
Non-claim	Illustrations are schematic, not experimental findings.
Non-claim	No cost savings or removal performance has been measured.

Contents

- 1. Problem definition and orbital environment
- 2. Existing mitigation and removal approaches
- 3. Automated design and prior-art boundary
- 4. Research questions and hypotheses
- 5. System architecture
- 6. Orbital mechanics foundations
- 7. Risk and consequence modelling
- 8. Optimization formulation
- 9. Candidate generation
- 10. Feasibility compiler
- 11. Uncertainty and robust decisions
- 12. Safety and governance
- 13. Data architecture
- 14. Prototype implementation

- 15. Evaluation design
- 16. Illustrative scenarios
- 17. Cost model
- 18. Verification and validation
- 19. Risk register
- 20. Roadmap and NASA engagement
- 21. IP, attribution and royalties
- Appendices A–J and references

1. Problem definition and orbital environment

Orbital debris is a systems problem, not a single-object cleanup problem. Objects differ in mass, orbit, size, trackability, material, status, and collision consequence. A removal action may be technically possible but economically irrational, unsafe, or inferior to tracking and avoidance. OCIE compares alternatives consistently rather than assuming physical capture is always best.

LEO is a useful initial domain, but not all debris is in LEO. GEO, MEO, and highly elliptical orbits have different constraints. The prototype should restrict its initial scope to a documented orbit subset and state exclusions. Catalogues are incomplete and state estimates have uncertainty; object counts should not be presented as exact.

Formulate the decision as choosing intervention portfolio x from feasible set X to reduce expected future harm while controlling lifecycle cost, intervention-induced risk, uncertainty, and operational complexity. The system must be allowed to return “no action” or “collect more data” if evidence is insufficient.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

1. Problem definition and orbital environment — validation plan

Validation objective: determine whether the claims associated with “problem definition and orbital environment” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

2. Existing mitigation approaches

Documented practice includes design-for-demise, post-mission disposal, propulsive deorbiting, drag sails, tethers, robotic capture, nets, harpoons, magnetic interfaces, servicing vehicles, collision avoidance, tracking improvements, and regulatory requirements. NASA's Small Spacecraft Systems Virtual Institute summarizes passive and active deorbit technologies [1].

Active removal is difficult because targets may tumble, lack capture interfaces, or be expensive to reach. Removal vehicles require navigation, rendezvous, capture, force management, and disposal. A failed intervention can itself create debris. The model must include intervention-induced risk, not just the target's current risk.

Software can help select targets, compare options, and expose assumptions, but it cannot replace physical capabilities or authoritative tracking data. OCIE is complementary to physical systems, not a substitute for them.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such

as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

2. Existing mitigation approaches — validation plan

Validation objective: determine whether the claims associated with “existing mitigation approaches” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

Review questions

- What is the closest documented predecessor?
- Which assumption can reverse the result?
- What observation would falsify the benefit?
- What evidence is needed before the next stage?

3. Automated design and optimization background

Automated engineering design searches candidate configurations subject to constraints. NASA's Advanced Control and Evolvable Systems group documents automated design using evolutionary algorithms and research into planning and control [4]. AI-assisted scientific discovery has also been studied as prioritized search over large design spaces [5]. Therefore, using AI to search for designs is not itself a novelty claim.

OCIE uses a compiler metaphor: parse requirements; normalize scenario data; generate candidates; apply feasibility and safety gates; simulate survivors; compare objectives; and produce an auditable report. Invalid plans should fail early with explicit reasons rather than pass through an opaque score.

A potential contribution would need to be a precise algorithmic detail: for example, a new candidate representation, uncertainty-aware search strategy, verifiable pruning rule, or improved combination of cascade consequences and intervention portfolios. These are hypotheses, not established novelty.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

3. Automated design and optimization background — validation plan

Validation objective: determine whether the claims associated with “automated design and optimization background” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

4. Research question and hypotheses

Primary question: does a physics-constrained intervention-search pipeline identify safer or more cost-effective portfolios than conventional target-ranking baselines on held-out scenarios? A secondary question is whether explanations and uncertainty bounds help a qualified reviewer reproduce or reject recommendations.

H1: under a fixed computational budget, OCIE may improve a preregistered utility metric relative to at least one baseline. H2: explicit uncertainty penalties may reduce overconfident recommendations when orbital knowledge is poor. H3: staged feasibility filters may reduce wasted high-fidelity simulation calls.

Null outcomes are valuable. If OCIE does no better than simpler methods, report the finding and identify which components add complexity without benefit. Publish negative results and model limitations where disclosure is safe.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

4. Research question and hypotheses — validation plan

Validation objective: determine whether the claims associated with “research question and hypotheses” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

5. Scope and exclusions

The initial prototype is decision-support software, not flight software. It will not command spacecraft, access restricted systems, or autonomously authorize manoeuvres. It will not claim to track every millimetre-scale fragment individually or use interventions outside validated physical domains.

Start with one documented orbit regime and a bounded candidate class. Later phases may include uncertainty envelopes, perturbations, and higher-fidelity tools. Each expansion should follow a stage gate and expert review.

Hardware manufacturing, launch procurement, autonomous proximity operations, laser operation, legal determinations, and claims of complete debris-environment prediction are out of scope for the first prototype.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

5. Scope and exclusions — validation plan

Validation objective: determine whether the claims associated with “scope and exclusions” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

6. System architecture

OCIE contains evidence ingestion, scenario normalization, candidate generation, a feasibility compiler, propagation and simulation, multi-objective ranking, and human-reviewed reporting. Separation of data, physics, optimization, and governance allows independent testing and prevents an AI model from overriding hard safety constraints.

Inputs record source, timestamp, frame, units, covariance availability, object identity confidence, and licence. Candidate records state target, time window, required capability, expected effect, cost range,

uncertainty, and safety conditions. The feasibility compiler checks dimensions, reachability, operational constraints, and intervention-induced hazards.

The evaluator estimates outcomes under uncertainty. The ranking layer returns a Pareto set rather than hiding trade-offs in one unexplained scalar. Reports record why each candidate passed or failed, model versions, and assumptions that dominate the result.

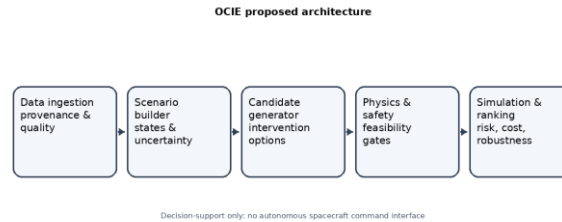


Fig. 1. Proposed OCIE architecture; all recommendations require human review.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

6. System architecture — validation plan

Validation objective: determine whether the claims associated with “system architecture” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

Review questions

- What is the closest documented predecessor?
- Which assumption can reverse the result?
- What observation would falsify the benefit?
- What evidence is needed before the next stage?

7. Orbital-mechanics foundations

In a two-body approximation, the position vector r follows $\ddot{r} = -\mu r / \|r\|^3$, where μ is Earth's gravitational parameter. Higher-fidelity propagation may require Earth oblateness, atmospheric drag, solar radiation pressure, third-body effects, and other perturbations according to orbit and time horizon.

A candidate intervention changes state or orbital elements through an impulse, continuous thrust, drag change, or another physically justified effect. Models must distinguish position, velocity, ballistic coefficient, area-to-mass ratio, and covariance. A small instantaneous change does not guarantee a universally safe outcome.

Relative state is $\Delta r = r_1 - r_2$ and $\Delta v = v_1 - v_2$. Close-approach assessment depends on uncertainty as well as nominal separation. A full method must account for covariance, geometry, hard-body radius, and applicable assumptions.

$$\ddot{r} = -\mu r / \|r\|^3$$

$$\Delta r = r_1 - r_2; \Delta v = v_1 - v_2$$

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

7. Orbital-mechanics foundations — validation plan

Validation objective: determine whether the claims associated with “orbital-mechanics foundations” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

8. Risk and consequence modelling

A framing equation for expected loss is $E[L] = \sum_i P(C_i) \cdot S_i$, where C_i is a collision or hazardous event and S_i is a consequence measure. Real risk assessment requires appropriate probability modelling and avoidance of double-counted correlated events.

For cascade analysis, an event can create fragments and alter the future orbital environment. A branching approximation can support sensitivity analysis, but a real model must represent fragment-size distributions, orbital dispersion, survival, and time horizon. A branching factor is not a universal threshold for all orbits.

Report multiple consequences: collision probability, expected fragments under stated assumptions, mission criticality, time in hazardous orbit, and uncertainty. If evidence is weak, report ranges or qualitative flags rather than false precision.

$$E[L] = \sum_i P(C_i) \cdot S_i$$

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

8. Risk and consequence modelling — validation plan

Validation objective: determine whether the claims associated with “risk and consequence modelling” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

9. Formal optimization

Let x denote an intervention portfolio, θ uncertain parameters, C lifecycle cost, R residual risk, and I intervention-induced risk. One formulation is to minimize the vector $(E[R+\lambda I], E[C])$ subject to physical, operational, legal, and safety constraints. λ is a documented policy parameter, not a physical constant.

A Pareto frontier exposes trade-offs. A weighted sum can be used as secondary analysis, but weights must be visible and sensitivity-tested. Constraints may include propellant, thrust, reachability, manoeuvre windows, collision limits, no-go zones, uncertainty thresholds, and rules against uncontrolled fragmentation.

Exact constraints require review by orbital-dynamics and flight-safety specialists. A candidate that appears cheap because a side effect was omitted is not a successful optimization result.

$$\begin{aligned} \min_x \quad & (E[R(x,\theta)] + \lambda E[I(x,\theta)], E[C(x,\theta)]) \\ \text{subject to } & g_i(x,\theta) \leq 0; x \in X_{\text{feasible}} \end{aligned}$$

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

9. Formal optimization — validation plan

Validation objective: determine whether the claims associated with “formal optimization” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes. The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

Review questions

- What is the closest documented predecessor?
- Which assumption can reverse the result?
- What observation would falsify the benefit?
- What evidence is needed before the next stage?

10. Candidate generation

Candidates can be typed records containing target, action class, timing, capability, state assumptions, expected effect, cost, uncertainty, and safety conditions. Candidate actions may include tracking, avoidance analysis, disposal trajectory assessment, servicing feasibility, or no intervention.

A grammar or graph representation can prevent incompatible action sequences. Search operators may include enumeration, mutation, recombination, local search, constraint programming, or model-based proposals. Their performance must be tested rather than assumed.

Maintain candidate diversity, but do not equate diversity with scientific novelty. A structurally different candidate can still duplicate a known concept or be physically useless.

Candidate evaluation workflow



Decision-support only: no autonomous spacecraft command interface

Fig. 2. Candidate evaluation pipeline and safety gates.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

10. Candidate generation — validation plan

Validation objective: determine whether the claims associated with “candidate generation” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

11. Feasibility compiler

The compiler applies gates: data completeness, unit/type checking, physics, reachability, safety, policy, and evidence sufficiency. Each rejection should have a machine-readable reason such as missing covariance, unreachable target, excessive delta-v, incompatible interface, or unsupported mechanism.

Early filters can save computation, but an aggressive filter may discard the best candidate. Test false-rejection rates against a high-fidelity reference set. Safety-sensitive filters should be conservative and fully documented.

The compiler should never silently discard a candidate. Rejection logs make search outcomes reproducible and reveal whether a filter is doing useful work.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

11. Feasibility compiler — validation plan

Validation objective: determine whether the claims associated with “feasibility compiler” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance

Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

12. Uncertainty and robust decisions

Represent measurement uncertainty, model uncertainty, and future environmental variability separately. Methods may include covariance, parameter ranges, ensembles, or probabilistic models. A robust objective may be $J_r(x) = E[J(x, \theta)] + \kappa \cdot SD[J(x, \theta)]$, where κ controls aversion to variability.

Alternative formulations include worst-case optimization and conditional value at risk. They are established tools; using them is not itself novel. Selection depends on the risk context and must be justified.

Run sensitivity analyses and report which inputs can reverse rankings. If a candidate is preferred only under a narrow, weakly supported assumption, do not present it as robust. Better tracking may be more valuable than immediate action.

$$J_{\text{robust}}(x) = E[J(x, \theta)] + \kappa \cdot SD[J(x, \theta)]$$

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

12. Uncertainty and robust decisions — validation plan

Validation objective: determine whether the claims associated with “uncertainty and robust decisions” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes. The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

Review questions

- What is the closest documented predecessor?
- Which assumption can reverse the result?
- What observation would falsify the benefit?
- What evidence is needed before the next stage?

13. Safety and human oversight

OCIE must remain decision-support software. No score should authorize a manoeuvre, rendezvous, capture, or intervention. A qualified operator must review data freshness, uncertainty, operational context, and the consequences of acting or not acting.

The interface should display assumptions, source timestamps, model versions, confidence intervals, constraint failures, and ranking rationale. Reviewers must be able to challenge inputs, disable models, and reject candidates.

Safety assurance should include hazard identification, failure-mode analysis, independent review, cybersecurity, logging, and change control. A high cost score cannot compensate for an unsafe recommendation.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

13. Safety and human oversight — validation plan

Validation objective: determine whether the claims associated with “safety and human oversight” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

14. Data architecture and provenance

Each data item should include source, acquisition time, epoch, coordinate frame, units, quality flag, transformation history, and permitted use. Preserve original records alongside normalized values. Cleaning and transformation must be deterministic.

Public orbital elements are useful for exploratory research but have accuracy and validity limits. Operational work may require validated state vectors, covariance, and authorized access. Licensing and export-control issues should be reviewed before redistribution.

Reproducibility requires frozen snapshots, version identifiers, configurations, random seeds, dependency manifests, and test fixtures. Where data cannot be shared, provide synthetic alternatives and document access requirements.

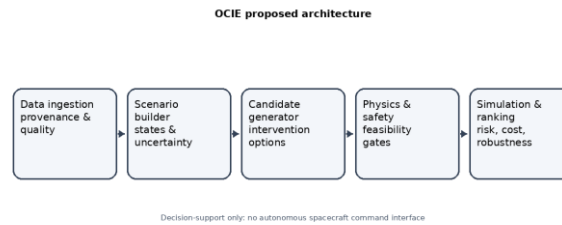


Fig. 1. Proposed OCIE architecture; all recommendations require human review.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

14. Data architecture and provenance — validation plan

Validation objective: determine whether the claims associated with “data architecture and provenance” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

15. Prototype implementation

A software-only prototype can use Python, a validated orbital mechanics library or propagator, a provenance database, and a web interface. Begin with rule-based enumeration before adding evolutionary search or machine learning. This makes the initial results easier to interpret.

Separate physics from optimization. Unit tests should check units, reference frames, conservation behaviour in simplified cases, and responses to known perturbations. Integration tests should ensure every candidate is traceable to its inputs.

A minimal prototype should run locally for a small synthetic scenario. Cloud compute is unnecessary until profiling justifies it. Cache repeated simulations and stop when the compute budget or evidence threshold is reached.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

15. Prototype implementation — validation plan

Validation objective: determine whether the claims associated with “prototype implementation” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

Review questions

- What is the closest documented predecessor?
- Which assumption can reverse the result?
- What observation would falsify the benefit?
- What evidence is needed before the next stage?

16. Evaluation and baselines

Compare OCIE with credible baselines: random feasible search, rule-based ranking, cost-only ranking, risk-only ranking, and a conventional optimizer using the same simulator. Beating a weak baseline is not evidence of superiority.

Separate development and held-out scenarios. Include nominal and stress cases: uncertain states, missing data, conflicting constraints, limited intervention capacity, and model mismatch. Avoid tuning thresholds on final evaluation cases.

Metrics include residual risk estimate, cost, runtime, feasible-candidate yield, false rejection, robustness, trace completeness, and abstention frequency. Pre-register the main metric and report uncertainty intervals where justified.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

16. Evaluation and baselines — validation plan

Validation objective: determine whether the claims associated with “evaluation and baselines” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

17. Illustrative scenario: target priority

Consider a hypothetical orbital region with three inactive objects. One has high estimated consequence but poor state certainty; another has moderate consequence and high certainty; a third has lower consequence but is inexpensive to reach. This example illustrates why a single score can mislead.

OCIE estimates risk ranges, intervention costs, and feasibility, then examines whether improved tracking changes the decision. A low-cost information action may dominate removal if uncertainty drives the ranking. Removal may be preferred if consequence remains high across plausible states.

The correct output is not a predetermined winner. It is a transparent comparison with confidence ranges, assumptions, and the value of additional information.

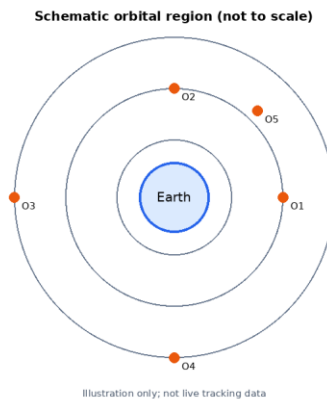


Fig. 3. Schematic orbital region; not to scale and not live tracking data.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

17. Illustrative scenario: target priority — validation plan

Validation objective: determine whether the claims associated with “illustrative scenario: target priority” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement,

and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

18. Illustrative scenario: portfolios

A portfolio may combine tracking for one object, avoidance analysis for another, and removal planning for a third. Benefits may be non-additive because actions share resources, change future conjunctions, or compete for time windows.

Portfolio evaluation must model dependencies and sequencing. If action A changes an object's orbit, rerun downstream risk analysis before accepting action B. Shared costs must not be counted twice.

Compare portfolio search with greedy selection. More complex search is justified only if it yields consistent gains on held-out cases.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.

- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

18. Illustrative scenario: portfolios — validation plan

Validation objective: determine whether the claims associated with “illustrative scenario: portfolios” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

Review questions

- What is the closest documented predecessor?
- Which assumption can reverse the result?
- What observation would falsify the benefit?
- What evidence is needed before the next stage?

19. Abstention and value of information

When candidate outcome intervals overlap and rankings change with covariance, the system should flag an unstable decision. It may recommend additional tracking or expert review instead of selecting the nominally highest score.

Value of information approximates expected decision improvement after an observation minus its cost. It requires modelling how new data change the state estimate and decision; more data are not always worth the delay or expense.

This scenario tests whether the system is appropriately cautious. Always selecting a winner can look decisive but be less safe than acknowledging insufficient evidence.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

19. Abstention and value of information — validation plan

Validation objective: determine whether the claims associated with “abstention and value of information” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

20. Cost and economic feasibility

Separate development, data acquisition, compute, expert review, integration, operations, verification, and maintenance. Hardware removal costs belong only to interventions that use hardware. Software prototype cost must not be confused with flight-mission cost.

Lifecycle cost can be written

$C_{LCC} = C_{dev} + C_{data} + C_{compute} + C_{ops} + C_{review} + C_{maintenance} + C_{failure}$. Each term should be a range with labour, compute, data, and contingency assumptions.

Low initial cost comes from public data, open-source tooling where licences permit, small synthetic scenarios, and staged validation. Cost-effectiveness must be measured against a realistic baseline; savings cannot be claimed before comparison.

$$C_{LCC} = C_{dev} + C_{data} + C_{compute} + C_{ops} + C_{review} + C_{maintenance} + C_{failure}$$

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

20. Cost and economic feasibility — validation plan

Validation objective: determine whether the claims associated with “cost and economic feasibility” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

21. Verification and validation

Verification asks whether software implements its specification; validation asks whether the model represents the intended problem. Passing unit tests does not establish physical accuracy.

Verification should include dimensional checks, idealized two-body tests, regression against trusted propagators, randomized property tests, and traceability checks. Validation should compare with independent implementations or published benchmark cases where possible.

Record discrepancies and model uncertainty. Operational deployment would require formal safety assurance, independent testing, cybersecurity review, validated data, and approval by the responsible operator.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

21. Verification and validation — validation plan

Validation objective: determine whether the claims associated with “verification and validation” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement,

and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

Review questions

- What is the closest documented predecessor?
- Which assumption can reverse the result?
- What observation would falsify the benefit?
- What evidence is needed before the next stage?

22. Risk register

Risks include stale data, inaccurate covariance, unit errors, propagation bugs, optimizer overfitting, false rejection, underestimated induced debris, manipulated weights, cybersecurity compromise, and misinterpretation of a score as a command.

Mitigations include quality gates, typed units, independent propagator comparison, held-out testing, conservative uncertainty margins, human review, immutable logs, access control, version pinning, and a no-action option.

A critical failure is a cheap-looking candidate whose side effects are omitted. Require an explicit side-effect assessment for every candidate; missing information must trigger rejection or abstention.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

22. Risk register — validation plan

Validation objective: determine whether the claims associated with “risk register” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

23. Prior-art and novelty boundary

The broad idea of automated search and simulation for interventions is established. OCIE is not presented as a wholly unprecedented discovery. Potential contributions may lie in a specific typed grammar, auditable feasibility compiler, uncertainty-aware portfolio evaluator, or a pruning method that reduces compute while preserving high-value candidates.

A defensible novelty claim requires a precise description, nearest-prior-art comparison, and demonstrated technical effect. The word “compiler” is an architectural analogy, not proof of novelty.

Maintain a novelty ledger with feature, closest references, search terms, search date, differences, and unresolved questions. Say “no exact match identified in the documented search,” never “nobody has ever thought of this.”

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

23. Prior-art and novelty boundary — validation plan

Validation objective: determine whether the claims associated with “prior-art and novelty boundary” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

24. Research ethics and responsible AI

AI-assisted systems can reproduce hidden assumptions, favour well-observed objects, or optimize a proxy that fails to represent safety. Disclose model roles, data, limitations, human decisions, and conflicts of interest.

Keep the system interpretable enough for technical review. Compare machine learning with rule-based and classical optimization baselines. Retain candidate generation and filtering logs; do not fabricate validation results or describe synthetic data as observations.

The author remains responsible for correctness, originality claims, permissions, and final submissions. Disclose assistance where required by the chosen venue or institution.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

24. Research ethics and responsible AI — validation plan

Validation objective: determine whether the claims associated with “research ethics and responsible ai” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

Review questions

- What is the closest documented predecessor?
- Which assumption can reverse the result?
- What observation would falsify the benefit?
- What evidence is needed before the next stage?

25. Roadmap and stage gates

Stage 0: literature and patent landscape. Stage 1: requirements and synthetic benchmark. Stage 2: baseline implementation. Stage 3: constrained candidate generation. Stage 4: uncertainty and portfolio search. Stage 5: independent validation. Stage 6: external review and integration study.

Advance only if results are reproducible, baselines credible, metrics meaningful, and the next stage resolves an important uncertainty. A small negative result can prevent a much larger waste of resources.

A software-only first phase is low cost relative to flight hardware, but later phases may require paid data, expert time, and formal collaboration. A NASA proposal must fit an appropriate programme and its current rules.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

25. Roadmap and stage gates — validation plan

Validation objective: determine whether the claims associated with “roadmap and stage gates” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

26. NASA engagement strategy

NASA does not automatically evaluate, fund, or adopt unsolicited ideas. Check current programme rules and submission channels. A credible package includes a concise abstract, technical gap, method, test plan, milestones, budget, risks, and measurable outcomes.

Prepare a one-page concept note, short technical summary, reproducible prototype or benchmark, and prior-art map. This long-form report can support that package, but length alone does not improve the chance of acceptance.

Do not promise that NASA will be amazed, call the author, approve the proposal, or award a contract. The goal is to make the research easy to evaluate and falsify.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

26. NASA engagement strategy — validation plan

Validation objective: determine whether the claims associated with “nasa engagement strategy” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

27. IP, attribution and royalties

Listing Mahib Sadman as proposing author identifies authorship of this document; it does not establish sole inventorship of every technical element. Copyright in the text differs from patent rights in a potentially patentable invention. Applicable legal standards vary by jurisdiction.

There is no automatic royalty entitlement from sending a proposal to NASA. Compensation depends on a contract, grant terms, licence, procurement award, or other written agreement. Government procurement and funding clauses can affect rights.

Record dated design notes, contributors, and prototype versions. Seek qualified IP advice before public disclosure if a patent may be appropriate. Do not claim exclusive ownership until inventorship and prior art are assessed.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

27. IP, attribution and royalties — validation plan

Validation objective: determine whether the claims associated with “ip, attribution and royalties” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

Review questions

- What is the closest documented predecessor?
- Which assumption can reverse the result?
- What observation would falsify the benefit?
- What evidence is needed before the next stage?

28. Limitations and conclusions

This proposal has no implementation results, measured savings, verified orbital simulation, or flight demonstration. The candidate generator, risk model, and objective function remain to be built and tested. Figures are conceptual, and the literature search is not exhaustive.

Open questions include benchmark choice, required model fidelity, public data quality, portfolio-search cost, and whether the compiler architecture adds value over existing tools. Answers may differ by orbit and intervention class.

The next step is to select one benchmark and implement the simplest baseline. Narrow or abandon claims if the method shows no measurable improvement or prior art fully covers the proposed mechanism.

Operational implications

This section must be converted into a measurable requirement before implementation. A requirement should specify its pass/fail condition, data source, model version, and reviewer. Replace vague terms such as “significantly safer,” “low cost,” or “highly novel” with a metric, uncertainty interval, baseline, and acceptance threshold.

Set thresholds before final evaluation. If no authoritative threshold exists, report a sensitivity curve and avoid presenting an arbitrary cutoff as a standard. Maintain traceability from raw data through assumptions to the final recommendation.

Minimum evidence package

- Versioned inputs, provenance, units, timestamps, and quality flags.
- Baseline implementation and independently reproducible test case.
- Failure cases, uncertainty intervals, and rejected-candidate reasons.
- Reviewer sign-off on assumptions affecting safety or feasibility.

28. Limitations and conclusions — validation plan

Validation objective: determine whether the claims associated with “limitations and conclusions” are supported by reproducible evidence. The team should specify a test fixture, a baseline, a measurement, and a stopping rule. The evaluation should include normal, boundary, and failure cases, not only scenarios in which the method performs well.

A robust test design separates training or development scenarios from held-out evaluation scenarios. The held-out set should include state uncertainty, missing data, conflicting constraints, and at least one case where the correct output is abstention. The evaluator should report both performance and failure modes.

The result should be considered inconclusive if the baseline is weak, the simulator is unvalidated, or the outcome metric is selected after inspecting final results. A negative result is still useful because it identifies where the architecture adds no value.

Test protocol

Test	Method	Evidence to retain
Input quality	Inject stale or incomplete data	Gate result and reason code
Physics validity	Compare with reference case	State error and tolerance
Safety constraints	Create infeasible candidate	Rejection log
Robustness	Vary uncertain parameters	Outcome intervals
Reproducibility	Repeat fixed run	Versioned output comparison

Decision gate

Proceed only if the test is reproducible, the baseline is credible, the result meets a predeclared threshold, and an independent reviewer agrees that the model's scope is appropriate. Otherwise, revise, narrow, or stop the research effort.

Appendix A — Mathematical notation

μ is Earth's gravitational parameter; r and v are inertial position and velocity; x is an intervention portfolio; θ is uncertain parameters; C is lifecycle cost; R is residual risk; I is intervention-induced risk; κ and λ are decision parameters. All objectives depend on scenario, time horizon, and model fidelity. A change in horizon may change the preferred intervention. Real collision-risk assessment must use appropriate conjunction analysis and account for covariance, geometry, correlations, and model validity.

Implementation checklist

- Define scope and success metrics.
- Record sources and assumptions.
- Build baseline and synthetic fixtures.
- Run independent review.
- Document failures and limitations.
- Update prior-art ledger before novelty claims.

Appendix B — Pseudocode

INPUT scenario S , data snapshot D , constraints G , candidate grammar A , simulation budget B . 1) Validate D ; if critical fields are missing or stale, return `DATA_INSUFFICIENT`. 2) Normalize units, frames, epochs, object identifiers, and uncertainty. 3) Generate candidate portfolios including no-action and information-gathering options. 4) Apply type, reachability, safety, and policy filters; store every rejection reason. 5) Allocate simulation fidelity based on uncertainty and decision impact. 6) Estimate cost, residual risk, induced risk, robustness, and compute cost. 7) Construct a Pareto set and run sensitivity/ablation tests. 8) Emit evidence and require human review.

Implementation checklist

- Define scope and success metrics.
- Record sources and assumptions.
- Build baseline and synthetic fixtures.
- Run independent review.
- Document failures and limitations.
- Update prior-art ledger before novelty claims.

Appendix C — Example data schema

Object record: object_id, source, source_timestamp, epoch, frame, state_vector, covariance_or_quality_flag, object_class, status, mass_estimate, area_to_mass_estimate, provenance, licence, limitations. Intervention record: action_class, target_id, time window, capability, expected state change, actuation limit, cost range, failure modes, induced-debris assessment, assumptions, confidence, validation status. Evaluation record: candidate_id, simulator_version, input_snapshot_id, random_seed, objective values, uncertainty intervals, constraint results, rejection reasons, reviewer, review timestamp.

Implementation checklist

- Define scope and success metrics.
- Record sources and assumptions.
- Build baseline and synthetic fixtures.
- Run independent review.
- Document failures and limitations.
- Update prior-art ledger before novelty claims.

Appendix D — Requirements and acceptance tests

REQ-001 provenance retained: remove metadata and verify ingestion fails. REQ-002 unit mismatch rejected: inject km/m mismatch. REQ-003 no-action always considered: make all active actions infeasible. REQ-004 reproducibility: repeat fixed snapshot, configuration, software version, and seed. REQ-005 abstention: increase uncertainty beyond threshold. REQ-006 rejection reason logged: force each gate to fail. REQ-007 no command interface: inspect architecture and network permissions.

Implementation checklist

- Define scope and success metrics.
- Record sources and assumptions.
- Build baseline and synthetic fixtures.
- Run independent review.
- Document failures and limitations.
- Update prior-art ledger before novelty claims.

Appendix E — Baseline and ablation plan

Baselines: risk-only ranking, cost-only ranking, transparent weighted score, random feasible search under equal compute budget, and standard optimizer using the same simulator. Ablations: remove uncertainty penalties; remove portfolio dependencies; replace staged filtering with one final check; remove no-action; remove provenance generation. Report paired scenario results, intervals where justified, runtime, feasible yield, failures, and excluded cases with reasons.

Implementation checklist

- Define scope and success metrics.
- Record sources and assumptions.
- Build baseline and synthetic fixtures.
- Run independent review.
- Document failures and limitations.
- Update prior-art ledger before novelty claims.

Appendix F — Preliminary cost and resource plan

Phase 0 literature and patent landscape: researcher time, expert consultation, and possibly specialist search. Phase 1 software prototype: laptop compute, version control, Python environment, test data, and dashboard; no launch or flight hardware. Phase 2 validation: independent orbital-mechanics review, reference propagator comparison, benchmark curation, and uncertainty tests. Phase 3 external study: only after demonstrated value, seek a university or research partner. Actual budget depends on labour, data access, and scope; no fixed cost is asserted.

Implementation checklist

- Define scope and success metrics.
- Record sources and assumptions.
- Build baseline and synthetic fixtures.
- Run independent review.
- Document failures and limitations.
- Update prior-art ledger before novelty claims.

Appendix G — Risk register

R1 data quality failure: timestamps, gates, abstention. R2 model mismatch: independent propagators and benchmarks. R3 false novelty claim: prior-art search and expert review. R4 optimizer overfitting: held-out scenarios and preregistered metrics. R5 unsafe interpretation: decision-support-only scope and human approval. R6 IP misunderstanding: legal advice and no royalty guarantee. R7 scope creep: fixed orbit regime and stage gates. R8 reproducibility failure: versioned inputs, dependencies, and seeds.

Implementation checklist

- Define scope and success metrics.

- Record sources and assumptions.
- Build baseline and synthetic fixtures.
- Run independent review.
- Document failures and limitations.
- Update prior-art ledger before novelty claims.

Appendix H — Prior-art search protocol

Search NASA and ESA official material, NASA Technical Reports Server, NASA TechPort, IADC, peer-reviewed journals, conference proceedings, Google Patents, WIPO PATENTSCOPE, and Espacenet. Search synonyms: automated design, evolutionary engineering, debris remediation, collision-cascade mitigation, risk-based target selection, intervention portfolio, uncertainty-aware optimization, decision support, space traffic management, and active debris removal. Record each closest reference, date, overlap, difference, and unresolved question. A search can report no exact match in a bounded corpus, not prove nobody ever conceived an idea.

Implementation checklist

- Define scope and success metrics.
- Record sources and assumptions.
- Build baseline and synthetic fixtures.
- Run independent review.
- Document failures and limitations.
- Update prior-art ledger before novelty claims.

Appendix I — NASA proposal checklist

One-page abstract; specific gap; technical contribution; benchmark; baseline; acceptance criteria; uncertainty model; safety controls; reproducible prototype; prior-art table; budget; milestones; risks; and correct current submission channel. Avoid claims of guaranteed novelty or agency adoption. A concise concept note plus a short technical summary is usually more useful for initial outreach than a 70-page report alone.

Implementation checklist

- Define scope and success metrics.
- Record sources and assumptions.
- Build baseline and synthetic fixtures.
- Run independent review.
- Document failures and limitations.
- Update prior-art ledger before novelty claims.

Appendix J — Authorship, attribution and IP

This document lists Mahib Sadman as author of the proposal. The author should verify all technical content, references, and venue-specific disclosure requirements. The name OCIE is a working title, not a registered mark. Copyright in prose does not establish patent rights in a method. No royalty percentage is proposed because royalties require an agreement defining rights, payment, milestones, and termination. Obtain qualified IP advice before public disclosure if patent protection may be appropriate.

Implementation checklist

- Define scope and success metrics.
- Record sources and assumptions.
- Build baseline and synthetic fixtures.
- Run independent review.
- Document failures and limitations.
- Update prior-art ledger before novelty claims.

22. Refined candidate contribution: the intervention certificate

The central candidate contribution in this revision is not the broad idea of optimizing debris removal. That broad area is established. The narrower hypothesis is that each recommended intervention portfolio should carry a machine-checkable, replayable certificate that makes the decision auditable across counterfactual, physical, uncertainty, side-effect, and governance layers. The certificate is intended to be a standard output object, not merely a paragraph generated after optimization.

A certificate would include: (1) an immutable identifier for the scenario snapshot; (2) the baseline policy and baseline result; (3) the proposed action sequence; (4) model versions and parameter provenance; (5) physical and operational constraints checked; (6) estimated residual risk and intervention-induced risk; (7) uncertainty intervals and sensitivity rankings; (8) known omitted effects; (9) a list of rejected alternatives with reason codes; (10) the conditions under which the recommendation must be withdrawn; and (11) a reproducibility manifest. The certificate must be generated from structured data, not hand-edited prose.

The novelty hypothesis is that this certificate, when integrated with candidate generation and uncertainty-aware portfolio comparison, may improve independent review time, reproducibility, and rejection of overconfident recommendations. Each of these outcomes can be measured. The certificate is not itself a new law of physics and is not claimed as a new type of orbital intervention. Formal verification, provenance tracking, and safety cases are established disciplines. A prior-art review must determine whether an equivalent end-to-end certificate has already been proposed in space traffic management or other safety-critical optimization domains.

The main falsification condition is straightforward: if existing systems already produce an equivalent certificate, or if adding the certificate does not improve reviewer performance or error detection, the proposed contribution is not supported. The project must then narrow the claim or abandon it. This document therefore labels the certificate as a candidate contribution rather than a confirmed invention.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

23. Counterfactual baseline design

An intervention can appear beneficial only because its comparison baseline is weak. OCIE should therefore freeze a baseline policy before candidate evaluation. The baseline could represent no new action, existing operator practice, a simple risk-ranked target list, or a standard optimization method. The correct baseline depends on the use case and must be selected before final evaluation.

For each scenario s , let $J_0(s, \theta)$ be the outcome under the baseline and $J_x(s, \theta)$ the outcome under candidate x , with θ representing uncertain parameters. Define the estimated improvement $\Delta J(x, s, \theta) = J_0(s, \theta) - J_x(s, \theta)$ when lower J is better. This difference must be computed using matched assumptions and compatible model fidelity. If baseline and candidate use different propagation models, the comparison may be biased by the model rather than the intervention.

The certificate should include absolute outcomes as well as differences. A candidate with a large relative improvement can still be unacceptable if its absolute risk remains too high. Conversely, a small relative improvement may be valuable if it affects a high-consequence hazard. Report distributions or intervals for ΔJ , not only its mean, and show the probability that the candidate is worse than the baseline under the selected uncertainty model.

A key methodological risk is baseline contamination: developers may tune the baseline to be weak or select it after seeing the candidate's performance. Prevent this by versioning the baseline, publishing its parameters, using at least two credible baselines, and reserving a held-out scenario set. When baseline policies differ by operator or orbit regime, report results separately rather than blending incompatible cases into one average.

The counterfactual is a modelled comparison, not an observed alternate universe. The paper must state that causal claims are conditional on model validity. The correct language is “the model estimates a reduction under assumptions X ,” not “the intervention will prevent Y collisions.”

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

24. Certificate schema and machine-checkable invariants

A practical certificate schema should be versioned and typed. A minimal JSON-like record would contain `certificate_id`, `scenario_snapshot_hash`, `data_sources`, `epoch`, `frame`, `units`, `candidate_actions`, `baseline_id`, `simulator_id`, `constraints`, `objective_vector`, `uncertainty_summary`, `sensitivity_summary`, `omitted_effects`, `rejected_alternatives`, `abstention_conditions`, `reviewer_status`, and `reproducibility_manifest`. Each field needs a defined type, unit, permissible null state, and validation rule.

The schema should enforce invariants. Every orbital state must identify a reference frame and epoch. Every numerical field must include units or be dimensionless by definition. Every action must point to a defined capability and a set of assumptions. Every candidate ranking must cite the objective function and parameter values. Every claimed improvement must link to a baseline result produced from a compatible scenario snapshot. Every rejected candidate must have at least one reason code. Missing critical evidence must cause an explicit incomplete or abstain status.

A hash can detect changes to a stored input or certificate, but a hash does not prove that the source data are correct. Similarly, a digital signature can identify who signed a record but does not prove the physical

model is valid. The paper should not confuse data integrity with truth. Independent model validation and expert review remain necessary.

Machine-checkable invariants make the certificate testable. Unit tests can intentionally remove a timestamp, alter a coordinate frame, mismatch baseline versions, or omit an intervention side-effect assessment. The expected output is a failed validation with a diagnostic. A candidate that passes schema checks is not automatically safe; schema validation is only one gate in the complete verification chain.

The schema should be designed to support future standards rather than impose a proprietary data silo. Export to open formats, explicit version migration, and a public synthetic benchmark would make the research more reproducible and more likely to be evaluated independently.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

25. Intervention portfolio model and interaction terms

A portfolio is a sequence or set of actions rather than a single action. Let $x=(x_1,\dots,x_n)$ represent candidate actions. The total benefit generally cannot be assumed to equal the sum of independent benefits because actions may share a service vehicle, alter later orbital geometry, compete for time windows, or change the value of additional observations. A model that ignores these interactions can overstate portfolio value.

One conceptual objective is $J(x,\theta)=R(x,\theta)+\lambda I(x,\theta)+\mu C(x,\theta)$, where R is residual risk, I is intervention-induced risk, C is lifecycle cost, and λ and μ are explicit decision weights. This scalar form is useful for examples but hides trade-offs; the preferred implementation should retain a vector objective and compute a Pareto set. The weights are policy assumptions, not physical constants, and should be tested over a range.

Pairwise interaction can be represented as $J(x)=\sum_i J_i(x_i)+\sum_{i<j} K_{ij}(x_i,x_j)+\dots$ where K_{ij} represents interaction between actions. For realistic orbital planning, higher-order interactions may matter and exact enumeration may become expensive. The prototype should therefore start with small portfolios, compare exact enumeration against approximations, and record the size at which computation becomes impractical.

A portfolio evaluator must account for temporal order. An action may change an object's state, so later feasibility and risk estimates must use the updated state. The sequence cannot be evaluated by independently scoring every action at the initial epoch. Any shared resource must be represented as a constraint, and duplicated costs must be removed.

Evaluation should compare greedy selection, integer or constraint programming where tractable, and a general heuristic. The most complex method should not be assumed best. The research claim is supported only if portfolio modelling materially improves outcomes on held-out cases and does so without an unacceptable increase in computation or model error.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

26. Objective functions, units, and policy transparency

A scientific optimization paper must separate measurable quantities from value judgments. Propellant mass, estimated cost, manoeuvre duration, and model runtime have units. Collision probability is dimensionless but depends on a defined event and time horizon. Expected fragment count depends on a fragment model and size threshold. A weighted score combining these quantities requires normalization and explicit weights.

One transparent approach is to preserve a vector objective $F(x)=[R(x), C(x), T(x), U(x), I(x)]$, where R is residual risk, C lifecycle cost, T operational time, U uncertainty or instability, and I intervention-induced risk. A candidate is Pareto-dominated if another candidate is no worse on all objectives and strictly better on at least one. Pareto analysis reduces hidden weighting but does not eliminate decision-making: a human still chooses among non-dominated options.

If a scalar score is needed, normalization must be defined from physical bounds, policy thresholds, or a training set that is fixed before final evaluation. Min-max normalization based on the candidate set can cause a candidate's score to change merely because a new candidate is added. That behaviour must be understood and documented. Robustness tests should include alternative plausible weights and normalization methods.

No single universal objective should be used for all orbital regimes and mission classes. An operational satellite's immediate conjunction risk, a derelict rocket body's long-term fragmentation consequence, and a tiny untrackable fragment require different models. A common framework can host these cases only if each use case specifies its assumptions and outcome metrics.

The certificate should display objective values in original units wherever possible, show the normalized score separately, and include a sensitivity plot. It should not hide an unsafe trade-off behind a single green badge or an unexplained “optimal” label.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

27. Uncertainty taxonomy and propagation

Uncertainty should be separated into at least three classes. Aleatoric uncertainty describes variability represented probabilistically in the model. Epistemic uncertainty describes incomplete knowledge, uncertain model structure, or poorly characterized parameters. Operational uncertainty includes future actions, manoeuvre execution, data latency, and unobserved events. These categories may overlap in implementation, but the paper should state how they are represented.

For a state estimate, a covariance matrix may characterize local estimation uncertainty under stated assumptions. Propagation can transform that covariance, but nonlinear dynamics and non-Gaussian distributions can make a Gaussian approximation inaccurate. A robust implementation should compare linearized covariance propagation with Monte Carlo or other appropriate methods on selected benchmark cases. If the distributions diverge materially, the certificate should state that the simplified method is not sufficient for the decision.

The uncertainty penalty $J_{\text{robust}}=E[J]+\kappa\cdot SD[J]$ is a useful illustrative form, but κ is a preference parameter. It is not a universal safety factor. Alternative approaches include worst-case optimization,

chance constraints, and conditional value at risk. The selected method should follow the decision context and be validated against synthetic cases with known distributions.

The system must distinguish “uncertain because the measurement is poor” from “uncertain because the physical model is incomplete.” Additional tracking may reduce the first category but not necessarily the second. A high-quality state estimate cannot compensate for an invalid fragment-generation model. The certificate should record uncertainty sources separately and identify which are reducible by new data.

A critical test is ranking stability. Perturb uncertain inputs across their justified ranges and measure how often the preferred candidate changes. If rankings are unstable, OCIE should abstain, recommend information gathering, or present a set of conditional recommendations instead of one definitive answer.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

28. Value of information and abstention

A system that can recommend only an intervention is structurally biased toward action. OCIE should allow three broad outputs: act or investigate a feasible intervention; acquire information; or abstain because the evidence is inadequate. Abstention is a legitimate result, not a system failure.

The expected value of information (EVI) can be framed as the expected improvement in decision quality after an observation minus the cost of acquiring it. In a simplified formulation, $EVI(y) = E[\max_a U(a|D,y)] - \max_a E[U(a|D)] - \text{Cost}(y)$, where D is current evidence and y is a possible observation. The exact expression depends on whether utility or loss is used and how uncertainty is represented. The formula should be implemented carefully and checked against small examples where the optimal decision is analytically known.

Information has value only if it can plausibly change a consequential decision. If every plausible observation leads to the same action, further data may not justify their cost. Conversely, if the decision flips across plausible states and the consequences are high, improved tracking or a specialist assessment may be worthwhile. The system should report the assumptions behind any EVI estimate.

Abstention thresholds must be calibrated and predeclared. A threshold tuned after observing test results can make the model appear safer than it is. Evaluation should report both error rate and coverage: how often the system makes a recommendation and how often those recommendations satisfy the acceptance criteria. A system can reduce errors by abstaining on every case, so coverage and usefulness must be measured together.

The research hypothesis is that explicit abstention and value-of-information analysis may reduce overconfident choices in uncertain scenarios. It must be compared with baselines that do not include these features. If no improvement is observed, the feature should be simplified or removed.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

29. Collision-cascade models and limits

Long-horizon debris growth is a population-level problem. A collision can generate fragments, change the distribution of orbital objects, and alter future collision rates. A reduced-order model can be useful for

scenario screening, but it must not be mistaken for a high-fidelity physical forecast. Fragment size, velocity distribution, mass, material, orbital dispersion, and object survival affect outcomes.

A simple stock-flow representation may use $N_k(t)$ for the population in size or orbit bin k . Its rate of change can be written $dN_k/dt = \text{Sources}_k - \text{Sinks}_k + \sum_j T_{j \rightarrow k} - \sum_j T_{k \rightarrow j}$, where source and sink terms include launches, fragmentation, natural decay, and controlled disposal, while T represents transfers between bins. This is a modelling template, not a complete debris-environment model. Every term requires a documented empirical or physical basis.

A key hazard is double-counting. If the collision term already includes fragments produced by a simulated event, adding a separate expected-fragment penalty may count the same consequence twice. Similarly, an intervention that lowers one object's collision probability may shift risk to another orbit or create a different close-approach geometry. The model should track the event definitions and causal dependencies explicitly.

Validation should compare the reduced-order model against published benchmark scenarios or a trusted population model where access permits. Parameters should be varied over justified ranges. If the model cannot reproduce known qualitative behaviour, it should not be used to rank long-term interventions. A low-cost software project can still be useful as a screening tool if it clearly labels its domain of validity and defers high-stakes decisions to validated models.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

30. Physical feasibility and intervention side effects

An intervention candidate must be evaluated against the physical mechanism that makes it work. A deorbit manoeuvre requires a reachable trajectory and sufficient propulsion or another justified source of momentum change. A drag device requires a suitable atmosphere, area-to-mass ratio, deployment reliability, and a remaining orbital lifetime that makes the effect meaningful. A capture operation requires target-state knowledge, relative-velocity management, compatible geometry, and a disposal plan.

The framework should not generate unconstrained science-fiction actions and then assign them low cost. Candidate grammars must specify required actuators and their limits. A candidate without a physical model should be labelled speculative and excluded from performance claims. If an unconventional mechanism is proposed later, it needs a separate physics analysis, energy and momentum accounting, and a safety review before inclusion in the optimizer.

Intervention-induced risk includes accidental fragmentation, unintended orbit changes, new debris, collision with other objects, loss of the servicing vehicle, and mission interference. The total risk must include the probability of intervention failure and its consequences. A low-probability catastrophic failure can dominate the expected-loss calculation; risk-sensitive methods may be appropriate.

The first software prototype should use existing, documented action classes. Novelty research should focus on the decision and certificate architecture until the team has enough evidence to propose new physical hardware. This sequencing prevents the paper from confusing a potentially useful planning tool with an unvalidated physical invention.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

31. Baseline algorithms and fair comparison

A fair comparison requires baselines that solve the same problem under the same constraints and compute budget. Recommended baselines include: no-action; simple risk ranking; cost-effectiveness ranking; greedy portfolio selection; a standard mixed-integer or constraint-programming formulation where tractable; and a widely used heuristic or evolutionary optimizer. The selected baseline set should be justified in the protocol before final results are examined.

Every method must receive the same scenario snapshot, action catalogue, physical feasibility model, and uncertainty samples. Otherwise, apparent gains may result from different assumptions rather than the algorithm. Report hyperparameter tuning budget and compute resources. If OCIE uses more simulation calls, compare both at equal compute and at equal wall-clock time where feasible.

The main metric should match the stated use case. For example, if the purpose is risk reduction under a fixed budget, compare residual-risk estimates and intervention-induced risk at the same budget. If the purpose is reviewer auditability, measure error detection, time to review, and agreement between independent reviewers. A single composite score is unlikely to capture all dimensions.

Use held-out scenarios and multiple random seeds for stochastic algorithms. Report distributions, not only the best run. Include failure cases and the proportion of scenarios in which the method abstains. A claim of superiority should include uncertainty intervals and an explanation of practical significance, not just a nominal percentage difference.

If the certificate feature is the candidate contribution, use an ablation that removes the certificate while holding the optimizer constant. This isolates whether the certificate improves reproducibility or safety review rather than attributing all results to a complex pipeline.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

32. Human factors and review experiment

The certificate's intended benefit includes helping reviewers identify missing evidence and reject unsafe or unsupported recommendations. That claim requires a human-factors experiment rather than only an algorithmic benchmark. Participants could be qualified graduate students or domain specialists, subject to appropriate ethical review and a clearly stated limitation on representativeness.

Prepare a set of fictional but realistic decision packets with known injected defects: stale orbital epoch, inconsistent units, omitted intervention side effect, unstable ranking, missing baseline, or unsupported certainty. Randomly assign participants to a conventional report condition and a certificate-assisted condition. Measure detection rate, false alarms, time to decision, confidence calibration, and inter-reviewer agreement.

The study should predefine the primary outcome and sample-size rationale. A pilot can estimate task difficulty and variance, but pilot outcomes should not be presented as confirmatory evidence. The

experiment should avoid exposing participants to live operational commands or implying that the test outputs are safe for flight.

A certificate may increase reading burden. Therefore, measure both error detection and time. If it improves detection only by making reviews dramatically slower, the design may need progressive disclosure, better summaries, or a smaller required certificate. If it increases confidence without improving accuracy, that is a negative safety result.

The protocol should include consent, anonymization, data-retention rules, and appropriate institutional review if human-subjects requirements apply. No claim about NASA operator performance should be made from a small convenience sample.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

33. Benchmark design and synthetic data

A reproducible benchmark should contain multiple scenario families: well-observed and poorly observed targets; single-action and portfolio cases; feasible and infeasible actions; low- and high-consequence hazards; and cases in which the correct answer is to gather information or abstain. Synthetic scenarios allow controlled ground truth, but they must be clearly distinguished from real orbital data.

Each benchmark record should specify the orbit regime, state uncertainty, action capabilities, objective weights or Pareto criteria, model fidelity, and known expected behaviour. At least some cases should be constructed so that the optimal choice can be verified analytically or by exhaustive enumeration. These small cases test algorithm correctness before larger scenarios are attempted.

The benchmark should include adversarial cases: missing covariance, inconsistent coordinate frames, unrealistic action capabilities, and model parameters outside the validated domain. It should also include benign cases where a simple baseline is expected to perform well. A system that only succeeds on cases designed around its own architecture is not persuasive.

For real-data evaluation, freeze a documented snapshot, record source and epoch, and comply with the data provider's terms. Public element sets may not be sufficient for operational conjunction probability; the paper must not imply otherwise. If exact truth is unavailable, evaluate calibration and sensitivity rather than pretending that a modelled collision probability can be compared directly with an unobserved true probability.

Publish the generator, scenario definitions, test seeds, expected outputs for synthetic cases, and known limitations where licensing permits. A public benchmark can become one of the most useful contributions even if the proposed optimizer does not outperform existing methods.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

34. Statistical analysis plan

The statistical plan should be specified before final evaluation. For each scenario, compute paired differences between OCIE and each baseline using identical inputs and uncertainty samples. Summarize the distribution of paired differences, including median, interquartile range, and confidence intervals

where the assumptions justify them. Avoid relying only on a mean that can be dominated by a few extreme cases.

For stochastic algorithms, use multiple seeds and report seed-to-seed variability. For scenario families, stratify results by uncertainty level, action count, and feasibility complexity. Aggregate performance can hide failure in a particular class, so subgroup results are necessary. If many metrics are tested, identify a primary metric and treat secondary analyses as exploratory or apply an appropriate multiple-comparison strategy.

Calibration matters if the system outputs probabilities. Reliability diagrams, proper scoring rules, and calibration error can be considered when ground-truth labels or well-defined synthetic outcomes exist. A model that ranks cases correctly but gives badly calibrated probabilities should not be used to support probabilistic safety claims.

Report practical significance alongside statistical uncertainty. A tiny improvement that requires ten times the compute may not be valuable. Conversely, a modest average gain may matter if it reduces rare high-consequence failures, but that claim requires enough relevant scenarios to estimate tail behaviour. Do not infer rare-event reliability from a small sample.

All exclusions, missing runs, failed simulations, and deviations from the protocol should be logged. The final report must include negative and inconclusive results. If the sample is too small to support a claim, say so rather than converting uncertainty into a definitive conclusion.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

35. Reproducibility package and software assurance

The research package should include source code, environment files, pinned dependencies, test data, synthetic scenario generator, configuration files, random seeds, and a command that reproduces the published tables. Where operational data cannot be redistributed, include schema-compatible synthetic fixtures and instructions for authorized users to supply their own data.

Software assurance should include unit tests for equations and transformations, integration tests across the pipeline, regression tests against a known benchmark, property-based tests for invariants, and fault injection for missing or corrupted data. Every release should record a commit identifier and version number. A model or parameter change that alters rankings should create a new certificate version rather than silently replacing the previous result.

Reproducibility also requires documenting the computing environment and numerical tolerances. Floating-point differences can affect optimization near a constraint boundary. The system should state whether a result is deterministic and how nondeterminism is managed. If a model depends on an external service or a changing dataset, the exact response or snapshot must be preserved where legally possible.

An independent reviewer should be able to reproduce at least a subset of results from a clean environment without relying on undocumented personal settings. This test is more meaningful than simply confirming that the original developer can rerun the code.

A reproducibility manifest can improve scientific trust, but it does not guarantee that the scientific model is correct. Verification, validation, and independent review remain separate requirements.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

36. Security, governance, and operational boundary

Although the initial prototype is not flight software, it may process sensitive operational information in future versions. Security design should include least-privilege access, dependency scanning, audit logging, input validation, and separation between research computation and any operational command network. The prototype should not include command credentials or direct spacecraft-control interfaces.

Model governance should identify who may approve a scenario, change objective weights, add an action class, release a version, or override a recommendation. Changes to thresholds and weights should be logged with rationale. A system that permits a user to change risk tolerance without leaving a trace can make results impossible to audit.

The decision certificate should state its intended use and invalidation conditions. A certificate becomes stale if the underlying orbital state changes, a model is updated, an action capability changes, or the scenario leaves its validated domain. Staleness should trigger a warning or invalidate the recommendation rather than leave an old ranking looking current.

Operational adoption would require review by the responsible agency or operator, and potentially safety, cybersecurity, legal, and export-control review. This paper does not claim that OCIE is approved for operational use. A successful research prototype is only an early stage in a long assurance process.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

37. Lifecycle economics and cost-effectiveness

The software cost model should include development labour, data acquisition, compute, expert review, integration, maintenance, security, verification, and failure remediation. A later hardware mission adds spacecraft design, manufacturing, testing, launch, operations, insurance or risk reserves where applicable, and disposal. These categories should not be mixed into one vague “low-cost” claim.

For a candidate x , cost-effectiveness may be expressed as $CE(x) = \Delta R(x) / C_LCC(x)$, where ΔR is the estimated reduction in a defined risk metric relative to baseline. This ratio is meaningful only if ΔR and cost are comparable and uncertainty is shown. If the estimated risk reduction crosses zero, a ratio can become unstable; report the joint distribution or Pareto frontier instead.

A fair cost comparison uses the same lifecycle boundary and includes the cost of failure. A reusable service vehicle may have a higher initial cost but lower marginal cost across multiple targets; a single-use device may be simpler but more expensive per intervention. OCIE can represent both only if the model includes target compatibility, servicing constraints, and realistic mission duration.

The proposed certificate can help prevent cost claims from omitting risk or lifecycle expenses, but it does not make an intervention cheap by itself. A software-only prototype may be inexpensive relative to space hardware, yet labour and independent review still have real costs. All budget numbers should be presented as estimates with assumptions and contingency, not as guaranteed procurement costs.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

38. IP strategy and authorship boundaries

The manuscript may list Mahib Sadman as its author and originator of this proposal document. However, a document's authorship is not equivalent to sole inventorship of every underlying method. Many elements discussed here—orbital propagation, optimization, robust decision-making, provenance records, and safety review—are established methods. A potential patent claim would have to focus on a specific eligible technical contribution that meets applicable legal standards.

Before public disclosure of a potentially patentable implementation, obtain qualified legal advice in the relevant jurisdictions. A prior-art search should include patent claims, technical papers, conference presentations, source repositories, standards, and analogous safety-critical systems. Search terms alone are not enough; claims and technical features must be compared element by element.

There is no automatic royalty percentage for submitting an idea to NASA or any other agency. Royalty rights arise from contracts, licences, grants, procurement terms, or applicable law. The author should not assume that a proposal submission creates an obligation to pay. Any collaboration agreement should define background IP, new IP, ownership, licensing, confidentiality, publication rights, milestones, and termination.

Keep dated records of the concept's development and the contributions of collaborators. If other people materially contribute to a potentially patentable claim, inventorship must be assessed under applicable law rather than decided by branding or the order of names on a report. The appropriate objective is accurate attribution and a legally defensible rights position.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

39. Literature landscape update through October 2026

The literature landscape changes quickly. A 2026 Scientific Reports article, “AI-driven collision risk prediction and debris optimization in low earth orbit,” reports a pipeline combining a machine-learning surrogate, interpretability, and multi-objective evolutionary optimization. This is a direct warning against claiming that the broad integration of AI, collision-risk prediction, and optimization is novel [24]. The proposed OCIE contribution must be narrower than that broad pipeline.

NASA TechPort lists Space Debris Elimination (SpaDE), which studies focused pulses of atmospheric gas to increase drag on debris [25]. NASA also lists a project on reconfigurable space-based laser networks for rapid-response debris removal [26], and a project on targeted dust-cloud deployments for just-in-time collision avoidance [27]. These projects show that unusual physical intervention concepts are actively explored; OCIE should not imply that its software can replace or out-invent them without evidence.

Research on optimal active-debris-removal mission planning has formulated target selection and multi-target mission planning as combinatorial optimization problems [28], [29]. A 2026 paper describes a bi-

level evolutionary framework for multi-debris mission design [30]. Therefore, multi-objective search, target sequencing, and uncertainty-aware optimization are established neighbouring fields.

The revised candidate contribution is consequently framed as a certificate-based audit and counterfactual workflow, with explicit falsification criteria. Even that may have precedents in safety-critical software, formal methods, model-based systems engineering, and space operations. A full novelty search must cross domain boundaries; searching only for the acronym OCIE would be inadequate.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

40. Scientific status and evidence ladder

The project should distinguish five evidence levels. Level 0 is a conceptual hypothesis. Level 1 is a formal specification with unit tests. Level 2 is a reproducible synthetic benchmark. Level 3 is validation against an independent reference model and realistic public data. Level 4 is independent peer review and a human-factors or operator study. Level 5 is integration into an authorized operational workflow. Each level answers different questions; passing one does not imply the next.

The current document is at Level 0 with a proposed path toward Levels 1–3. The equations are standard formulations and the diagrams describe a proposed architecture, but the implementation has not been run and no performance results have been measured. The paper must not be described as scientifically validated. It is scientifically structured and validation-ready, which is a different claim.

The minimum first milestone is a small executable prototype that ingests a fixed synthetic scenario, produces a candidate list, checks constraints, calculates a transparent objective vector, and emits a reproducible certificate. The second milestone is comparison with a simple baseline on analytically checkable cases. The third is independent propagation comparison and uncertainty stress testing.

A scientific claim should only be elevated when the evidence supports it. If a test fails, update the manuscript and report the failure. If the prior-art search finds a close match, narrow the novelty claim. If the system provides no measurable benefit, the correct result may be that the concept is not worth further development. Such outcomes are part of rigorous research rather than setbacks to be hidden.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

41. Go/no-go decision matrix

A staged decision matrix protects the project from continuing on enthusiasm alone. At Gate A, literature review must identify a specific gap not already solved by a close predecessor. If no defensible gap remains, stop or redefine the claim. At Gate B, the formal specification must be internally consistent and reviewed by someone with orbital-mechanics expertise. If assumptions are undefined, do not proceed to performance claims.

At Gate C, the prototype must pass deterministic tests on synthetic scenarios with known outcomes. Any unit, frame, or baseline mismatch blocks further evaluation. At Gate D, the method must be compared with credible baselines on held-out cases. If the primary metric does not improve, simplify the architecture or change the research question only with transparent versioning.

At Gate E, uncertainty and failure-mode testing must show that the system does not systematically become overconfident under missing data or model mismatch. At Gate F, independent reviewers must be able to reproduce the results and understand why a candidate was selected or rejected. Only after these gates should the team consider a NASA-facing technical submission or partnership request.

The go/no-go thresholds should be defined before the results are known. A decision matrix is not a way to guarantee success; it is a way to identify early when the project is not meeting its own stated purpose. Funding requests should be tied to the next uncertainty being resolved, not to a promise of a breakthrough.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

42. Proposal language suitable for external review

A defensible abstract should say: “This proposal investigates whether a certificate-based, counterfactual workflow can improve auditability and uncertainty handling in orbital-debris intervention planning. The system is a decision-support prototype. Its novelty and performance remain to be established through prior-art review and comparative evaluation.” This wording is clear about the research question and does not imply a completed discovery.

Avoid phrases such as “nobody has ever thought of this,” “NASA will be amazed,” “guaranteed to clean space,” or “scientifically proven” until evidence exists. These claims are not only difficult to support; they can reduce the credibility of a technically sound proposal. A reviewer should be able to identify exactly what is proposed, what is known, what remains uncertain, and how success will be measured.

The submission package should include a one-page concept note, a short technical summary, a reproducible prototype or benchmark if available, a prior-art matrix, a risk register, and a budget for the next stage. A long report is supporting material rather than a substitute for a crisp research question. Use the agency's current submission process and respect any rules on unsolicited proposals or programme eligibility.

The paper should invite criticism: ask reviewers to identify the closest prior art, assess the benchmark, and suggest the right reference model. This increases the chance of receiving useful technical feedback without implying that any agency is obligated to accept or fund the idea.

Acceptance criterion: this section is considered supported only when its assumptions, data lineage, test method, and failure conditions are documented and independently reviewable. Until then, it remains a proposed design element rather than a validated result.

43. Requirements traceability matrix

Requirements traceability links each stated goal to a design element, test, result, and reviewer. Without traceability, a long technical report can sound comprehensive while leaving its central claims untested. OCIE should maintain a matrix with requirement ID, rationale, source, implementation location, test case, acceptance threshold, result, and reviewer. The matrix should distinguish mandatory safety constraints from optimization preferences and research hypotheses.

A requirement such as “the system must account for uncertainty” is too vague. A testable version might state that every candidate report must include the defined state uncertainty representation and that missing

required uncertainty metadata causes an explicit abstention. A performance requirement such as “improve cost effectiveness” must specify the baseline, cost boundary, risk metric, scenario set, and minimum effect considered practically meaningful. If no threshold can be justified before evaluation, the study should report the response curve rather than invent a pass mark after seeing results.

Traceability should work in both directions. Every major claim in the paper should point to a method or evidence source, and every major test should map to a claim it can support. A test that is not connected to a requirement may still be useful exploratory work, but it should not be presented as validation of the central hypothesis. A requirement without a test is an open obligation, not a completed capability.

For an independent review, export the matrix to a machine-readable format and provide a short human-readable summary. Reviewers should be able to select a claim and trace it to the equation, input data, code version, test, and result. This is a key mechanism for reducing overclaiming: it becomes difficult to label a capability “validated” when the evidence link is missing.

44. Model fidelity ladder

Not every candidate needs the highest-fidelity simulation. A model-fidelity ladder can allocate effort progressively: level A uses analytical screening; level B uses simplified numerical propagation; level C adds relevant perturbations and uncertainty propagation; level D uses an independently validated high-fidelity tool. The exact levels should be defined by the chosen orbit regime and problem class rather than treated as universal standards.

Low-fidelity models are useful for eliminating clearly impossible options, but they can be misleading near constraint boundaries. For example, a candidate that appears reachable under a two-body approximation may fail when perturbations, finite thrust, attitude constraints, eclipses, or operational windows are included. The system must estimate the risk of false rejection and false acceptance when moving between levels.

Adaptive fidelity should depend on decision relevance. If two candidates are far apart in estimated performance, additional simulation may not change the ranking. If their outcome intervals overlap or a safety constraint is nearly active, high-fidelity analysis is more valuable. The certificate should record why a particular fidelity was selected and what uncertainty remains because higher fidelity was not used.

The validation study should compare rankings produced at each level with a reference model across a designed set of cases. Metrics should include rank correlation, false-feasible rate, false-infeasible rate, and runtime. A faster low-fidelity model is useful only if its errors are acceptable for the screening role. Any result near a hard safety boundary should be escalated rather than decided by a low-fidelity approximation alone.

45. Design of experiments for orbital scenarios

The benchmark should be designed as an experiment, not assembled only from convenient examples. Define factors such as orbit regime, object-state uncertainty, action count, candidate reachability, intervention cost, target consequence, model mismatch, and data completeness. Select levels for each factor based on documented ranges or synthetic stress conditions. A full factorial design may be too large, so fractional factorial or space-filling designs can be used for exploration, followed by focused tests around influential regions.

The purpose of design of experiments is to understand which conditions drive the system's performance. If OCIE performs well only when state uncertainty is low, that is a meaningful limitation. If the certificate improves reviewer error detection mainly when a specific field is missing, that can guide interface design. Random sampling alone may underrepresent rare but important failure cases.

The design should include edge cases intentionally: a candidate exactly at a reachability boundary, a ranking that changes under a small perturbation, a case where no action is feasible, and a case where information gathering is more valuable than intervention. Such cases are important because a model can achieve good average performance while failing in precisely the scenarios where caution matters most.

Pre-register the primary factors and outcomes. Exploratory analyses can be reported separately, clearly marked as exploratory. Publish the scenario-generation method and seed values so another researcher can recreate the benchmark. Where actual orbital data are used, explain how the synthetic design relates to real-world ranges without claiming that synthetic frequencies match the true population distribution.

46. Calibration and probability semantics

The word “probability” should be reserved for a quantity whose event definition, conditioning information, model, and time horizon are explicit. A conjunction probability, probability of successful capture, probability of disposal failure, and probability that a ranking is wrong are different quantities. They should not be merged into a single confidence number without a mathematically justified relationship.

If OCIE outputs probabilistic estimates, calibration should be evaluated where outcomes or trustworthy reference distributions exist. A reliability diagram compares predicted probabilities with observed frequencies across bins, while proper scoring rules assess probabilistic forecasts. Synthetic cases with known distributions are useful for software tests but do not prove calibration on the real debris environment. For rare events, sample sizes may be insufficient to estimate calibration precisely; uncertainty intervals must be reported.

A confidence score generated by a machine-learning model is not automatically a calibrated probability. Likewise, an uncertainty interval derived from one model does not necessarily cover model misspecification. The certificate should distinguish estimated probability, epistemic confidence, quality flags, and expert judgment. A reviewer should be able to see whether a number comes from a validated estimator, a simplified analytical model, or a heuristic.

The abstention policy should depend on the decision consequences and calibration evidence. A threshold that is suitable for a low-consequence research prioritization may be inappropriate for a safety-critical operational decision. The paper should avoid universal thresholds and instead describe the process for selecting them with the responsible operator and independent safety review.

47. Counterfactual validity and causal language

Counterfactual evaluation estimates what would happen under a different action, but only through a model. It is not direct observation of an unchosen future. Causal language is justified only when assumptions about dynamics, intervention effects, environmental variables, and competing actions are sufficiently specified. In an orbital environment, those assumptions can be demanding because the state evolves and actions can change future interactions.

The report should use conditional language: “under the specified propagation and collision model, candidate X has estimated outcome Y relative to baseline Z.” It should avoid “X will prevent a collision” unless the statement is a deterministic consequence under clearly stated conditions, which will be uncommon in uncertain orbital contexts. The certificate should identify which effects are directly simulated, approximated, or omitted.

Counterfactual consistency tests can be run on small synthetic cases. If the candidate is “no action,” the counterfactual outcome should match the baseline exactly within numerical tolerance. If an intervention is defined to have zero effect in a controlled test, the model should not produce a spurious improvement. If two independent actions are combined in a scenario designed without interaction, the portfolio result should match the additive expectation within the known model tolerance.

These tests do not establish causal validity in the real world, but they detect implementation errors. More realistic validation requires independent dynamics models and expert review. The paper should treat counterfactual estimates as decision aids with stated uncertainty, not as proof of a future outcome.

48. Fairness, responsibility, and allocation of limited resources

Intervention resources may be limited, and risk is not distributed uniformly across operators, nations, orbital regimes, or mission types. A decision framework can encode priorities implicitly through its objective function, even when developers do not intend to make a policy choice. The model should therefore disclose whether it prioritizes total expected loss, public-safety consequence, critical infrastructure, cost per risk reduction, or another criterion.

A technically efficient solution is not automatically a socially or institutionally acceptable solution. A ranking that maximizes aggregate benefit might leave some operators bearing disproportionate residual risk. A governance process may need constraints or reporting categories that show how risk reduction is distributed. These considerations should not be hidden in an arbitrary fairness coefficient; they require explicit policy decisions by accountable stakeholders.

For the first prototype, the goal is not to settle international policy. It is to make policy assumptions visible and allow reviewers to compare outcomes under different documented priorities. Sensitivity analysis can show whether the selected portfolio changes when priorities change. If it does, the report should identify the choice as policy-sensitive rather than purely technical.

The system should not infer national or commercial priorities from incomplete data. It should not treat data availability as evidence that an object is less important. Missing information should be recorded as uncertainty, and the model should be tested for systematic preference toward well-observed objects or operators with greater data resources.

49. Failure Mode and Effects Analysis protocol

A structured Failure Mode and Effects Analysis (FMEA) should be completed before any realistic scenario is used. For each failure mode, record cause, local effect, system effect, detectability, severity, likelihood basis, mitigation, residual risk, and accountable reviewer. A simple risk-priority number can help organize discussion but should not be mistaken for a precise probability or a substitute for safety engineering.

Candidate failure modes include stale data, incorrect frame conversion, a missing covariance, an invalid action capability, a propagation error, a flawed cost estimate, an unmodelled side effect, a false baseline

comparison, a misconfigured objective weight, a software dependency change, and a reviewer misunderstanding. The analysis should also include organizational failures, such as a recommendation being reused after its scenario becomes stale.

Test cases should inject these failures deliberately. The system should detect the issue, refuse to issue a confident recommendation, and preserve a diagnostic record. Some failures may not be detectable by the software alone; in those cases, the certificate should identify the required external review or data source. An undetected failure should be treated as a design defect, not merely as a user error.

FMEA is a living record. A new algorithm, data source, action class, or interface can create new failure modes. Update the analysis at each stage gate and before changing the system's intended use. No FMEA score should be used to claim that the system is safe without a broader assurance argument and independent review.

50. Threat model for software integrity

The threat model should include accidental and malicious alteration of inputs, candidate definitions, model parameters, objective weights, and output certificates. A research prototype can begin with local files, but a shared or deployed system may receive data from external sources and multiple collaborators. Provenance records and access controls reduce the risk that a result is changed without detection.

Threats include compromised dependencies, manipulated scenario data, unauthorized changes to thresholds, replay of stale certificates, and misleading reports generated from an altered model. Mitigations include dependency pinning, signed releases where appropriate, access control, immutable logs, schema validation, and clear expiry rules. A hash can detect alteration but does not establish data authenticity unless the source and signing process are trusted.

Security requirements should not be over-engineered for a local research prototype, but the architecture should avoid unsafe coupling to operational systems. The prototype should have no ability to send commands to spacecraft. Any future integration must be treated as a separate safety and cybersecurity project with authorization from the relevant operator.

The certificate should include software version, configuration hash, data snapshot identity, and generation time. If any critical element changes, the old certificate should be marked superseded or invalid. This helps prevent an otherwise correct historical result from being treated as a current recommendation.

51. Interface design and reviewer workload

The user interface should be designed around the questions a reviewer needs to answer: What is being recommended? Compared with what baseline? What is the estimated benefit? What can go wrong? Which assumptions dominate? What information is missing? Why were other candidates rejected? What changes would invalidate this recommendation? The interface should not force the reviewer to read every raw field before understanding the decision, but it must make the raw evidence available.

A layered interface can show a one-screen summary, followed by objective trade-offs, uncertainty and sensitivity, physical feasibility checks, provenance, and a complete audit trail. Each summary number should link to its definition and source. Reducing complexity must not remove caveats that materially affect safety. Colour or ranking badges should never imply certainty when evidence is weak.

A human-factors experiment should test whether the interface helps reviewers identify deliberately injected defects. Include both false-positive and false-negative opportunities. A reviewer who rejects

every candidate may appear safe but is not useful; therefore, measure acceptance accuracy, rejection accuracy, review time, and confidence calibration.

The interface should support disagreement. A reviewer must be able to mark a candidate as unsupported, request new data, or document a competing interpretation. The system should preserve dissent rather than forcing consensus. This is important for scientific transparency and for future institutional review.

52. Governance of model updates

A model update can change a ranking even if the scenario inputs remain the same. Every model version should have a release note, validation status, known limitations, and regression results. The certificate must identify the exact model version used. Historical results should remain reproducible under the previous version where dependencies and data permit.

The update process should distinguish bug fixes, improved physical fidelity, new data, new objective weights, and changes in candidate grammar. Each type of change has different implications. A bug fix may require rerunning affected results; a new force model may alter reachability and ranking; a policy-weight change may change the preferred candidate without any physical model change.

Regression tests should include known scenarios where outputs should remain stable and scenarios where change is expected. If a model update changes a safety-relevant ranking, the release process should require explanation and independent review. A version that performs better on an aggregate benchmark can still regress on a critical subgroup.

Governance should assign ownership for approving releases and responding to incidents. A research prototype may use lightweight review, but any operational system needs formal change control. Model updates should never silently overwrite the provenance needed to reproduce a previous recommendation.

53. Reporting uncertainty and avoiding false precision

Numerical precision in a report should match the accuracy of the underlying model and data. Reporting a risk value to six decimal places does not make the estimate accurate to that level. The document should state significant figures, uncertainty intervals, and the model assumptions that support them. When inputs are poorly characterized, qualitative ranges may be more honest than precise point estimates.

Separate numerical error, parameter uncertainty, and model uncertainty. Numerical error arises from computation and approximation. Parameter uncertainty reflects unknown values within a model. Model uncertainty reflects missing or inaccurate structure. These should not be collapsed into one standard deviation unless the mathematical justification is clear.

Every figure should state whether its data are measured, simulated, synthetic, or illustrative. A Pareto plot made from synthetic points should be labelled as a demonstration of visualization, not as a result. A cost estimate based on labour assumptions should be labelled as preliminary and accompanied by its assumptions. A scenario model that predicts future debris should not be presented as a forecast without validation.

Reviewers should be able to identify which claims are established by sources, which are derived mathematically, which are model outputs, and which are hypotheses. The manuscript should use distinct wording for each class. This separation is a central part of scientific authenticity.

54. Independent validation partners and roles

Independent review should be planned as a set of bounded tasks. An orbital-dynamics reviewer checks frames, epochs, propagation, and perturbation assumptions. An optimization reviewer checks objective functions, constraints, baselines, and algorithm fairness. A software-assurance reviewer checks reproducibility, schema invariants, and test coverage. A space-debris specialist checks the use case and whether the outcome metric is meaningful. A patent professional reviews novelty claims and the nearest prior art.

No single reviewer should be asked to certify the entire project without expertise across all relevant areas. Record each reviewer's scope, conflicts of interest, findings, and unresolved issues. An independent reviewer should have enough access to reproduce the relevant results, subject to data licensing and security constraints.

If an institution cannot provide a full review, use a staged approach: first validate the mathematical specification, then the prototype, then the benchmark, then the interpretation. Do not describe informal feedback as peer review. Likewise, a positive comment from a NASA employee or researcher does not constitute NASA endorsement unless the agency has formally stated it.

The project should budget time for revisions and negative feedback. A credible research proposal makes it easy for reviewers to identify weaknesses, because addressing those weaknesses is how the work improves.

55. Publication and open-science plan

Publication should follow the evidence stage. A concept note can describe the hypothesis and method. A methods paper can document the formal specification and synthetic benchmark. A results paper should be written only after the implementation and evaluation are complete. The paper should not be styled as a completed experimental study while it remains a proposal.

Where licences permit, release source code, synthetic data, test cases, and the reproducibility manifest. Use a persistent version identifier and provide a clear licence. If patent protection is being considered, obtain legal advice before releasing implementation details publicly; disclosure can affect rights depending on jurisdiction. The decision to publish or file should be deliberate rather than accidental.

A suitable publication venue must be selected based on scope, review standards, and audience. IEEE formatting alone does not make a manuscript an IEEE paper. The author should follow the target venue's official template and requirements, verify references, and disclose assistance as required. The current document is a long-form research proposal, not an accepted or peer-reviewed publication.

Open science includes negative results. If the proposed certificate adds no value, reporting that result can still help the field avoid redundant complexity. If a narrower contribution works only under certain conditions, state those boundaries rather than presenting universal claims.

56. Reproducible experiment record template

For every experiment, create a record containing experiment ID, hypothesis, primary metric, scenario family, train/test split or development/held-out designation, data snapshot, model version, configuration, random seeds, compute environment, baseline versions, run status, exclusions, and output artifact locations. The record should also contain a plain-language interpretation and a list of caveats.

The hypothesis should be falsifiable. For example: “In the defined synthetic benchmark, certificate-assisted review will increase detection of injected evidence defects compared with a conventional report without increasing median review time beyond a predeclared margin.” This can be tested. “OCIE will revolutionize space cleanup” cannot be tested as stated and should not be used as a research hypothesis.

Each result should be traceable to raw outputs. Summary tables must be generated from the recorded data rather than manually typed. Figures should include units, sample sizes, uncertainty bars where appropriate, and labels that distinguish synthetic from real data. If a run fails, preserve the failure log and document whether it was excluded from the analysis.

A reproducibility record is a practical way to convert a promising idea into a scientific programme. It allows an independent researcher to rerun the analysis, discover errors, and determine whether the stated conclusion follows from the evidence.

57. Prototype milestones and deliverables

Milestone 1 is a literature and prior-art map. Deliverables include a search protocol, closest-reference table, a list of unresolved technical gaps, and a decision about whether the certificate concept remains sufficiently distinct to test. Milestone 2 is a formal schema and baseline workflow. Deliverables include typed records, invariants, and tests for missing or inconsistent data.

Milestone 3 is a synthetic benchmark with analytically checkable cases. Deliverables include scenario generator, expected outcomes, test seeds, and a reproducibility guide. Milestone 4 is a baseline optimizer and certificate generator. Deliverables include source code, logs, candidate reports, and unit/integration test results. Milestone 5 is comparative evaluation with held-out cases, uncertainty stress tests, and ablation studies.

Milestone 6 is independent review of the orbital model and software assurance. Milestone 7 is a decision about whether the measured results justify an external proposal or publication. Each milestone should have a cost range, owner, acceptance criterion, and a no-go option. A milestone is not complete because code exists; it is complete when the acceptance evidence has been reviewed.

The project should avoid a hardware demonstration until a specific physical intervention need is identified and the software evidence shows that it can support a meaningful decision. Hardware would increase cost and risk and is not necessary to test the initial auditability hypothesis.

58. Research budget structure and cost uncertainty

A budget should distinguish cash expenses from labour and in-kind contributions. Cost categories include research time, software development, specialist consultation, data access, compute, testing, independent review, legal prior-art review, publication, and contingency. Each line should identify a quantity, unit cost, source, confidence range, and dependency. A single total without assumptions is not decision-grade.

For a software-only proof of concept, hardware and launch costs should be zero by design. The budget should not imply that a low-cost prototype can physically remove debris. It tests the decision framework, not a flight mission. If the prototype demonstrates value, a later phase can estimate integration and operational costs with an appropriate partner.

Sensitivity analysis should identify which cost assumptions dominate. If specialist review accounts for most of the budget, reducing compute costs may have little effect. If a data licence is required, the project

may need a different data strategy. Cost estimates should include failure and rework, because first implementations rarely meet all requirements without revision.

Funding requests should be tied to specific uncertainties: for example, funds for independent model validation, a benchmark study, or a human-factors experiment. Avoid requesting money on the basis of guaranteed novelty or promised NASA adoption. A modest, well-scoped request with measurable deliverables is more credible than a large claim with no prototype.

59. Legal and policy interface

Orbital-debris mitigation operates within national licensing, international space law, technical standards, and operator-specific requirements. OCIE should not encode a generic statement of “legal compliance” without a current, jurisdiction-specific requirements source. Policy constraints should be versioned and reviewed by qualified counsel or responsible compliance specialists.

The software should distinguish a legal requirement, an agency guideline, an operator rule, and a research preference. These have different authority and enforcement. If a requirement changes, the system must record the effective date and rerun affected cases. A policy rule should not be hidden inside an optimization weight because a hard obligation is not interchangeable with a preference.

The prototype should not provide legal advice. It can flag that a candidate requires compliance review and preserve the relevant rule reference. A real operational deployment would need a maintained requirements baseline and accountable approval process. The document should not imply that a software certificate can substitute for authorization or licensing.

This section is deliberately conservative. The research contribution is decision traceability, not a claim to solve international space governance. Legal and policy constraints should be treated as explicit inputs whose authority and applicability are documented.

60. Final integrated evaluation protocol

The final evaluation should be conducted in four phases. Phase A validates implementation correctness on synthetic unit cases. Phase B compares candidate-generation and ranking methods against baselines on held-out synthetic scenarios. Phase C tests robustness under uncertain inputs, missing data, and model mismatch. Phase D evaluates whether the certificate improves independent reviewer performance. Each phase answers a different question and should have its own acceptance criteria.

The primary algorithmic outcome might be risk reduction relative to a fixed baseline at a fixed lifecycle budget, but only if the risk model is valid enough for that comparison. The primary auditability outcome might be defect-detection rate in a controlled review experiment. Do not combine these into a single headline result unless the relationship is justified. Report the results separately and discuss trade-offs.

Pre-register the hypothesis, scenarios, baselines, metrics, and exclusions. Use multiple seeds for stochastic methods. Retain all failures. Independent reviewers should inspect the analysis plan before final results are unblinded. If the study is exploratory, label it as such and avoid confirmatory language.

The final conclusion should be proportional to evidence. A positive result supports only the tested conditions. It does not establish that OCIE can safely operate spacecraft, reduce all orbital debris, or outperform all NASA systems. A null result means the current implementation did not show the expected benefit under the tested conditions. Both outcomes are scientifically informative.

61. Ablation study for the certificate contribution

An ablation study is necessary because the proposed certificate combines several features that may independently affect the result. The study should compare at least four conditions: baseline report with conventional ranking; certificate with provenance only; certificate with counterfactual comparison and uncertainty; and full certificate with abstention rules and rejected-alternative explanations. The optimizer, scenario set, and objective function should remain constant across conditions. This isolates the contribution of each reporting and decision component.

For algorithmic outcomes, measure whether the certificate changes the selected candidate, how often it reveals an infeasible or unsupported candidate, and whether it changes the estimated benefit. For human review, measure defect-detection rate, false alarms, time, and confidence calibration. A certificate may improve traceability without changing the optimizer's numerical result; that is a valid but different contribution from improved risk reduction.

Predefine how results will be interpreted. If provenance alone improves reproducibility, the more complex uncertainty layer may not be necessary for the target use case. If reviewers detect more defects but take substantially longer, the interface may need redesign. If the full certificate does not outperform the baseline report, the proposed novelty hypothesis is not supported.

The ablation should include negative controls. For example, adding a field that is irrelevant to the decision should not materially improve performance. If it does, the evaluation may be measuring presentation or participant expectations rather than the certificate's technical value. Publish the ablation matrix and results, including inconclusive findings.

62. Benchmark leakage and data contamination

Benchmark leakage occurs when information from final evaluation cases influences model design, thresholds, feature selection, or manual rules. In an optimization system, leakage can happen even without machine learning: a developer may tune a heuristic after inspecting the scenarios that later appear in the test set. The resulting score overstates generalization.

Separate scenario creation, development, and final evaluation. Ideally, a person who did not develop the optimizer holds the final cases until the evaluation plan is frozen. If the benchmark is synthetic, use a generator with hidden random seeds and varied parameter combinations. If the benchmark uses public orbital data, freeze a dated snapshot and document how cases were selected.

The system should not learn target-specific rules from the final test set and then claim general performance. Any case used for debugging should be moved to the development set. If the final set is inspected after a failed run and the algorithm is changed, a new held-out set or a new evaluation round is needed.

Report all tuning budgets and manual interventions. A complex heuristic that was adjusted repeatedly by the author may perform well on known cases but fail on unseen scenarios. Reproducibility requires not only code and data but also a record of the decision process used to develop the algorithm.

63. Computational complexity and scalability

The search space can grow combinatorially as the number of objects, action types, time windows, and action sequences increases. If there are n candidate actions and arbitrary subsets are permitted, the number

of portfolios can grow on the order of 2^n before sequence constraints are considered. Adding multiple target visits and timing choices can increase the space further. The exact complexity depends on the formulation and constraints.

The first prototype should measure candidate count, feasible-candidate count, simulation calls, memory, wall-clock time, and solution quality as scenario size increases. Do not claim scalability based on a small scenario. Use controlled scaling experiments with increasing numbers of objects and action options. Report where exact enumeration becomes impractical and which approximation is introduced.

Possible strategies include pruning by hard constraints, bounding methods, decomposition, caching, surrogate models, and adaptive simulation fidelity. Each can introduce bias. A surrogate model may accelerate search but should be validated in the regions where it guides decisions, particularly near safety constraints. A pruning rule must be evaluated for false rejection of good candidates.

Scalability is not only computational. A system may generate more candidates than a human reviewer can meaningfully inspect. Candidate diversity, explanations, and certificate summaries must remain usable as the search grows. The output should present a manageable Pareto set and retain access to the full audit trail.

64. Surrogate models and safe use boundaries

A surrogate model approximates a computationally expensive simulator. It may make large searches practical, but its prediction errors can be consequential if used near a feasibility or safety boundary. OCIE should treat a surrogate as an acceleration tool, not as an authority. Each surrogate must state its training domain, validation domain, error distribution, and known extrapolation risks.

Training data should cover the relevant candidate and state space. Randomly sampled cases may underrepresent boundary conditions, so active learning or targeted sampling may be needed. The final evaluation should include out-of-distribution cases and report how the system detects or responds to them. If the model cannot identify that a query is outside its domain, it should not be trusted to rank it confidently.

A safe architecture can use the surrogate for preliminary ranking and call a validated higher-fidelity model for finalists. The certificate should indicate which results are surrogate predictions and which are independently propagated. A candidate that only appears superior under the surrogate should not be presented as a validated improvement.

The use of a surrogate is not itself novel. The research question is whether a specific, transparent allocation of fidelity and certificate requirements improves compute efficiency while preserving decision quality. This must be measured against a baseline that uses the same compute budget and has access to the same high-fidelity reference model.

65. Sensitivity to model assumptions

A sensitivity analysis should identify which assumptions influence the final recommendation. Vary orbital-state uncertainty, drag parameters, mass or area estimates, action costs, intervention success probabilities, objective weights, and the model horizon within defensible ranges. The purpose is not to show every conceivable parameter combination, but to identify those that can change the decision.

Global sensitivity methods can be considered when the model is sufficiently well-defined, while one-at-a-time variation can provide an initial diagnostic. The paper should state the limits of each method.

Correlated parameters should not be varied independently if doing so creates physically impossible combinations. Where parameter distributions are unknown, use scenario ranges and avoid assigning arbitrary probabilities.

The output should show ranking stability, the share of plausible cases in which each candidate is preferred, and the conditions under which the recommendation changes. A candidate that is optimal only in a narrow region should be described as conditional. If multiple candidates remain non-dominated across plausible assumptions, the system should preserve the trade-off rather than force a single winner.

Sensitivity analysis does not prove that the selected parameter range is correct. The range itself needs evidence or expert justification. A robust-looking result can still be wrong if the analysis omits the parameter that matters most. The certificate should therefore include a list of known omitted variables and a review status for the sensitivity model.

66. Model discrepancy and external validation

Model discrepancy is the difference between a mathematical model and the real process it represents. It is not always captured by parameter uncertainty. For orbital debris, discrepancy may arise from simplified atmospheric density, uncertain object properties, incomplete fragment distributions, unmodelled manoeuvres, or approximations in conjunction geometry. A model can be numerically solved correctly and still be a poor representation of the environment.

External validation should use a trusted reference model, published benchmark, or independently curated dataset appropriate to the target problem. Agreement between two models that share the same assumptions is weaker evidence than agreement between independently developed methods. The validation report should describe the independence of the reference and identify shared assumptions.

Where observational truth is unavailable, validate components separately: orbital propagation against known test cases, schema and units through property tests, optimization against exact solutions on small cases, and certificate review through controlled human-factors tasks. Do not claim full-system validation merely because each component passes a unit test.

The manuscript should distinguish verification, validation, calibration, and operational acceptance. These terms answer different questions. The project should not use “scientifically validated” until the relevant tests have been completed and reported, and even then the claim should be bounded to the validated use case and evidence level.

67. Decision certificate example: illustrative only

A certificate might summarize a hypothetical scenario as follows: Scenario S-014 uses a fixed synthetic snapshot and a documented state uncertainty. Baseline B-02 is a rule-based risk ranking. Candidate C-07 proposes an information-gathering action before committing to removal. The certificate records the model versions, objective vector, uncertainty interval, constraints, side effects, and the reason the candidate is preferred under a specified policy. These labels are illustrative and do not represent real satellites or measured performance.

The certificate should show the result in plain language and in machine-readable form. A human-readable section states the recommendation and caveats. A technical section lists equations, inputs, model versions, and constraints. A reproducibility section contains the scenario hash, software version, seed, and

configuration. An audit section records rejected candidates and reviewer comments. A change-control section defines when the certificate becomes stale.

The certificate must not imply that a recommendation is safe merely because every field is filled. Completeness is necessary but not sufficient. A false model can produce a complete and internally consistent certificate. Independent validation of the physics and assumptions remains necessary.

This example should not be converted into a performance table with invented percentages. Until the system is run, the report must not say that C-07 reduced risk by a given amount, saved a given budget, or outperformed a NASA system. The example exists to specify a format and test requirements.

68. Data governance and source quality scoring

A data-quality score may help prioritize review, but it should not replace the underlying metadata. Each input should retain its source, epoch, frame, uncertainty representation, known limitations, transformation history, and permitted use. If a score is used, its components and weights must be visible. A single quality score can hide a critical defect, such as missing covariance, behind several good metadata fields.

Quality checks should include freshness, completeness, consistency, provenance, unit validity, and compatibility with the model. Some checks are binary gates; others can be graded. The distinction matters because a high average score should not compensate for a missing field required by the physics model. Critical omissions must trigger a hard failure or explicit abstention.

When multiple sources disagree, the system should preserve the disagreement rather than silently choosing one. A data-fusion method can be applied only if its assumptions and correlation structure are understood. Otherwise, the certificate should record the conflict and request expert review. Data quality is a scientific input, not merely a database housekeeping task.

For public data, licensing and access terms must be recorded. If a dataset changes over time, archive a snapshot where permitted or provide a checksum and acquisition procedure. This helps other researchers reproduce the same analysis and understand why their current data produce a different result.

69. Research communication and claim taxonomy

Every statement in the manuscript should belong to a claim class: established background supported by literature; mathematical consequence under stated assumptions; model result; experimental observation; proposed design; or open hypothesis. These classes should be linguistically distinct. For example, “Newtonian orbital motion is modelled by...” is background; “the optimizer returned...” is a result only after an actual run; “we hypothesize that...” is future work.

The current paper contains a proposed architecture and research plan, not completed experiments. Figures generated for this report are conceptual. Equations are standard formulations and are included to define the problem, not to imply that the full model has been solved. The document should not be submitted as an experimental paper unless an implementation and results are added.

A claim-evidence table can list the sentence, claim type, supporting source or test, confidence, and remaining caveat. This is especially useful when a proposal includes both public facts and the author's proposed method. Reviewers should be able to distinguish an idea from evidence that the idea works.

Clear claim taxonomy strengthens originality rather than weakening it. It allows the author to identify exactly which part is a new hypothesis and which part is inherited from established engineering practice. That is the foundation for a defensible research contribution.

70. Prior-art search reproducibility log

A prior-art search should be repeatable. Record search date, databases, query strings, language, date range, patent classifications where applicable, inclusion criteria, exclusion criteria, and screening decisions. Save references to each candidate document and explain why it was considered close or not close. Search engines change, so a dated log is more useful than saying “the internet was searched.”

Search by function and technical features, not only by the project name. For the certificate hypothesis, relevant terms include proof-carrying decisions, safety cases, model provenance, counterfactual decision support, uncertainty-aware portfolio optimization, orbital debris risk, space traffic management, audit certificates, and safety-critical optimization. Search analogous fields because a technique used in aviation, medicine, energy, or autonomous systems may be relevant prior art even if it was not originally described for orbital debris.

A patent professional should review claims and technical disclosures, not just abstracts. A paper that uses different terminology may still disclose the same core mechanism. Conversely, superficial keyword overlap does not automatically mean that two inventions are identical. The analysis should compare elements and technical effect carefully.

The final conclusion should be bounded: “No exact match was identified in the searched sources and date range.” It should not say “no human being has ever thought of this.” Private disclosures, unpublished work, inaccessible databases, and different terminology prevent that absolute conclusion.

71. Reviewer checklist for technical credibility

A reviewer should be able to answer the following questions from the paper: What precise decision does OCIE support? What is the baseline? What data and model are required? Which constraints are hard constraints? How are intervention-induced risks represented? What happens when data are missing? What evidence supports each physical assumption? What test would falsify the claimed benefit? Which components are existing techniques and which are proposed? What would count as success?

The reviewer should also examine whether the objective uses quantities with incompatible units, whether normalization hides important trade-offs, whether uncertainty is treated consistently, and whether the system can return no action. The model should not assume that the highest risk object is always the best removal target; accessibility, cost, interaction effects, and the possibility of new hazards can change the decision.

The paper should include a concise list of known limitations near the beginning, not only in an appendix. It should identify the absence of experimental results, the incomplete prior-art search, and the need for independent orbital-mechanics review. A credible proposal makes these limitations visible and provides a plan to resolve them.

If the reviewer cannot reproduce a key result or identify its supporting data, the claim should be downgraded until evidence is available. The purpose of the checklist is to improve the work, not to persuade a reviewer to approve it regardless of weaknesses.

72. Final research statement and bounded contribution

The bounded contribution proposed here is an auditable decision workflow for orbital-debris intervention planning in which each candidate carries a structured certificate connecting its counterfactual baseline,

physical feasibility, uncertainty, side effects, rejected alternatives, and invalidation conditions. The system is designed to be tested against conventional reports and optimization baselines. It is not a claim of a new physical force, capture device, or guaranteed debris-removal technology.

The core hypothesis is that such certificates may improve reproducibility and independent review, particularly when candidate rankings are sensitive to uncertainty or when information is incomplete. The evidence required is a working implementation, synthetic tests with known outcomes, comparison with credible baselines, uncertainty stress tests, and a controlled review experiment. The hypothesis is falsified if existing tools already provide equivalent functionality or if the certificate does not measurably improve the intended outcomes.

The proposal is scientifically structured but not yet scientifically validated. It contains no experimental performance claim. The next step is to build the smallest executable prototype, conduct a documented prior-art review, and obtain independent review of the mathematical assumptions. Only then can the contribution be narrowed and described with confidence.

This conclusion is intentionally modest. A technical contribution earns importance through evidence, reproducibility, and demonstrated usefulness, not through claims of being unprecedented or guaranteed to attract NASA. The proposed work should be judged by the same standards as other research: clarity, correctness, comparison, falsifiability, and transparency.

73. Formal decision-certificate acceptance logic

A certificate should have explicit states rather than a single pass/fail flag. Suggested states are INCOMPLETE, FEASIBILITY-FAILED, EVIDENCE-INSUFFICIENT, MODEL-OUT-OF-DOMAIN, CANDIDATE-ELIGIBLE, REVIEW-REQUIRED, APPROVED-FOR-ANALYSIS, and EXPIRED. These states describe the research workflow; none is an authorization to execute a spacecraft command. Transitions between states should be triggered by logged events and validated fields.

The state machine should prohibit invalid transitions. For example, a candidate cannot become eligible if a required state uncertainty is missing, and it cannot be approved for analysis if a hard physical constraint fails. A candidate that was previously eligible should become expired if its scenario snapshot or critical model version changes. These rules can be tested through state-transition coverage.

A reviewer should be able to reconstruct why a certificate reached its state. Store the event sequence, input identifiers, validation results, and the user or process that made each change. If a human overrides a recommendation for research purposes, the override should be documented as a separate decision, not silently alter the original result.

The novelty hypothesis is not that state machines are new. It is that a defined state machine, combined with the certificate fields and counterfactual risk accounting, could make orbital intervention decisions more auditable. A literature review should compare this workflow with safety-case systems, mission-planning tools, and decision-record standards before claiming a distinct contribution.

74. Uncertainty-aware constraint margins

Hard constraints should not be evaluated only at the nominal state when uncertainty can move a candidate across a boundary. A constraint $g(x, \theta) \leq 0$ may be satisfied for the mean estimate while violated for a plausible state or parameter realization. The model should therefore distinguish nominal feasibility from robust feasibility and chance-constrained feasibility.

A robust condition may require $g(x, \theta) \leq 0$ for all θ in a justified uncertainty set. This can be conservative and computationally difficult. A chance constraint may require $P(g(x, \theta) \leq 0) \geq 1 - \alpha$, where α is a chosen risk tolerance. The value of α is a policy or safety requirement and must not be selected merely to make a candidate pass. The probability model and uncertainty set must be supported by evidence.

For the prototype, implement the simplest form that matches the synthetic benchmark and validate it against known cases. Test candidates near the constraint boundary, because these are where nominal and robust feasibility are most likely to disagree. Report the rate of false-feasible classifications against the reference model.

The certificate should include the margin to each critical constraint, the uncertainty model, the chosen tolerance, and whether the margin is nominal or robust. A green status without this information could create false confidence. No single generic margin should be assumed suitable for every orbit or action type.

75. Pareto-front interpretation and selection protocol

A Pareto frontier is a set of non-dominated candidates, not a single answer. The system should help a reviewer understand why one candidate is cheaper, another reduces more estimated risk, and another is more robust to uncertainty. A reviewer should be able to inspect the objective vector rather than infer trade-offs from an opaque score.

The selection protocol should identify who sets the policy preferences, when those preferences are set, and how changes are logged. For a research benchmark, the preferences can be fixed before evaluation. For an eventual operational use, the responsible operator or agency must define priorities within applicable safety and legal requirements. The software author should not unilaterally choose a universal trade-off between cost and safety.

A candidate may be dominated under one model but non-dominated under another plausible uncertainty model. The report should therefore show whether Pareto membership is stable. If the frontier changes substantially under modest perturbations, the system should present the decision as sensitive rather than stable.

The evaluation should measure whether reviewers can correctly interpret the frontier. In a controlled study, ask participants to select a candidate under specified constraints and explain the trade-off. Compare their selections and explanations with a documented decision rubric. This can test the usability of the certificate without pretending that the underlying physical intervention has already been validated.

76. Decision latency and operational freshness

A decision-support recommendation can become stale while it is being reviewed. The system should define freshness requirements for every input class and explain how elapsed time affects the validity of the result. A state estimate used in a hypothetical long-horizon study may remain useful for research after it is old, while a conjunction-specific analysis may become unusable much sooner. Freshness cannot be set as one universal timeout.

The certificate should record the scenario epoch, data acquisition time, analysis time, and an expiry rule tied to the intended use. If the system cannot calculate a defensible expiry time, it should mark the result as not suitable for time-critical use. A new observation, manoeuvre, or model update may invalidate a previous certificate before the nominal expiry.

Measure end-to-end latency separately from computation time. Latency includes data acquisition, preprocessing, simulation, review, and decision communication. A faster optimizer does not necessarily produce a faster operational decision if its output is difficult to review or relies on delayed data.

This proposal does not define real-time service-level requirements. Those depend on a specific operational context and authorized data source. The initial prototype should report latency in reproducible tests and avoid claims about real-time readiness.

77. Negative-result handling and research stop conditions

A rigorous project must define how it will handle negative or inconclusive results. If the certificate does not improve review accuracy, if the optimizer fails to beat a simple baseline, or if results are highly sensitive to unvalidated assumptions, the correct response is to report the outcome and narrow the project. A predetermined desire for novelty must not override the evidence.

Stop conditions should include: no distinct technical gap after prior-art review; failure to reproduce benchmark results; high false-feasible rate; no measurable benefit over baseline; unacceptable compute cost; and inability to validate a critical physical assumption. A pivot may be justified if the evidence identifies a narrower use case where the method works, but the change must be documented and tested on a fresh evaluation set.

A negative result can still yield a useful publication if the benchmark, failure analysis, or reproducibility tooling is valuable. The author should not reinterpret a failed primary hypothesis as success by selecting a secondary metric after the fact. Secondary findings should be labelled exploratory and tested in a subsequent study.

The manuscript should contain a results template that remains blank until experiments are run. This discourages accidental fabrication and makes the distinction between planned work and completed work explicit.

78. Evidence ledger for each factual statement

In addition to the novelty ledger, maintain an evidence ledger for factual claims. Each record should include the claim, claim type, source or test, date checked, exact supporting passage or data, limitations, and the manuscript locations where the claim is used. For numerical claims, record units and whether the number is an estimate, observation, or output of a model.

The ledger should distinguish a source that establishes the existence of a project from a source that establishes its performance. A NASA TechPort entry may document a research objective without demonstrating that the proposed technology has succeeded. Similarly, a paper's abstract may describe reported results, but the method and limitations should be examined before transferring its conclusion to OCIE.

Every reference should be checked for correct title, author, date, DOI, and version. When a source is not accessible or bibliographic information is uncertain, mark it for verification rather than inventing details. A reference list is not evidence by itself; each important claim needs an appropriate citation in the text.

This practice is particularly important for agency proposals. A reviewer may check the cited material quickly, and a broken link or overstated interpretation can undermine otherwise careful work. The ledger should be maintained as the manuscript evolves and included in the reproducibility package where practical.

79. Research integrity and authorship statement

The author statement should accurately describe who created the manuscript and who contributed to any future implementation, experiments, or patentable technical features. If collaborators later contribute substantially to the conception or execution, update acknowledgements and authorship according to the applicable venue's rules. The author should not claim sole inventorship of established methods merely because they appear in a proposal written by one person.

The current document is a proposal authored for Mahib Sadman. It includes public background concepts and a candidate integration workflow. It does not prove that the workflow is new, nor that the author is legally the sole inventor of any future implementation. Authorship, copyright, patent inventorship, and ownership are distinct questions.

The author is responsible for reviewing the manuscript, checking citations, and ensuring that no synthetic result is presented as observed data. If generative tools or outside assistance are used, disclose them where required by the target venue or institution. The final submitted version should be personally reviewed and technically validated.

An honest authorship statement increases credibility. It is better to state that the paper proposes a candidate novelty hypothesis and has not yet been experimentally validated than to imply that results or intellectual-property rights exist when they do not.

80. Requirements for a future experimental-results section

A future results section should only be added after the prototype has been run according to a documented protocol. It should state the test environment, dataset or synthetic generator, number of scenarios, baseline versions, compute budget, random seeds, exclusions, and primary metrics. Include a table of raw or aggregated results and enough detail for independent reproduction.

The section should report both benefits and costs: estimated risk outcomes, lifecycle cost, runtime, feasible-candidate yield, false-feasible and false-infeasible rates, ranking stability, and abstention coverage. For a certificate review experiment, report reviewer count, task design, outcome measures, and limitations of the participant sample. Do not combine software benchmark results with physical debris-removal performance.

Any claim of improvement should include uncertainty and practical significance. If the confidence interval includes no improvement, state that the result is inconclusive. If the method performs well only on a particular scenario family, restrict the conclusion to that family. If a baseline was not implemented, do not imply it was beaten.

The current paper intentionally does not include fabricated result values. A results section template can be prepared, but it must remain unfilled until the evidence exists. This is a central requirement for scientific authenticity.

81. Conclusion on originality and validation status

The project now has a narrower candidate contribution than the original broad OCIE concept: a structured intervention certificate that binds counterfactual comparison, physical-feasibility checks, uncertainty, side effects, rejected alternatives, and invalidation conditions into a replayable decision record. The integration

is a research hypothesis, not a proven invention. Similar mechanisms may exist in space operations or other safety-critical engineering fields, so prior-art review remains necessary.

Scientific validation is also pending. The document supplies a mathematical framing, software architecture, test protocol, benchmark plan, risk analysis, and reproducibility requirements. It does not contain experimental proof that the certificate improves decisions, that OCIE reduces orbital debris, or that it is cheaper than existing methods. Those claims require implementation and measured comparisons.

The recommended next action is a small, software-only prototype and a focused prior-art search. The first experiment should test a narrow claim—such as whether certificates help independent reviewers detect missing evidence—before the project attempts to optimize physical interventions. This sequence is lower cost, easier to falsify, and more likely to produce useful evidence.

The project should be presented to NASA or another agency as a request for technical evaluation only after the core claim is sufficiently precise. No agency can be expected to accept the idea simply because the report is long or ambitious. Credibility comes from a clearly bounded contribution, transparent limitations, reproducible results, and a willingness to revise the idea when evidence requires it.

References

- [1] NASA Small Spacecraft Systems Virtual Institute, “Deorbit Systems,” State-of-the-Art of Small Spacecraft Technology, 2026. [Online]. <https://www.nasa.gov/smallsat-institute/sst-soa/deorbit-systems/>
- [2] NASA Orbital Debris Program Office, “Orbital Debris Mitigation Standard Practices.” [Online]. <https://orbitaldebris.jsc.nasa.gov/mitigation/>
- [3] Inter-Agency Space Debris Coordination Committee, “IADC Space Debris Mitigation Guidelines.” [Online]. <https://www.iadc-home.org/>
- [4] NASA, “Advanced Control and Evolvable Systems Group.” [Online]. <https://www.nasa.gov/intelligent-systems-division/autonomous-systems-and-robotics/aces-group/>
- [5] A. K. Agrawal, J. McHale, and A. Oettl, “Artificial Intelligence and Scientific Discovery: A Model of Prioritized Search,” NBER Working Paper 31558, Aug. 2023. doi:10.3386/w31558.
- [6] D. J. Kessler and B. G. Cour-Palais, “Collision Frequency of Artificial Satellites: The Creation of a Debris Belt,” *Journal of Geophysical Research*, vol. 83, no. A6, pp. 2637–2646, 1978. doi:10.1029/JA083iA06p02637.
- [7] NASA Office of Technology, Policy, and Strategy, “Cost and Benefit Analysis of Mitigating, Tracking, and Remediating Orbital Debris,” 2024. [Online]. <https://www.nasa.gov/wp-content/uploads/2024/05/2024-otps-cba-of-orbital-debris-phase-2-plus-svgs-v3-tjc-tagged.pdf>
- [8] NASA TechPort, “Adaptive Just-in-time Collision Avoidance via Targeted Dust Cloud Deployments from Orbital Platforms,” Project 156380, 2026. <https://techport.nasa.gov/projects/156380>
- [9] NASA TechPort, “Rapid Response Debris Removal Using Reconfigurable Space-Based Laser Networks,” Project 156377, 2026. <https://techport.nasa.gov/projects/156377>
- [10] NASA Office of Inspector General, “NASA’s Efforts to Mitigate the Risks Posed by Orbital Debris,” Report IG-21-011, 2021. <https://oig.nasa.gov/wp-content/uploads/2024/02/ig-21-011.pdf>
- [11] European Space Agency, “Space Debris.” https://www.esa.int/Space_Safety/Space_Debris
- [12] European Space Agency, “ClearSpace-1.” https://www.esa.int/Space_Safety/ClearSpace-1
- [13] IEEE, “Resources, Forms, and Templates for IEEE Conferences.” <https://events.ieee.org/forms-and-templates/>
- [14] NASA, “Micrometeoroids and Orbital Debris.” <https://www.nasa.gov/centers-and-facilities/white-sands/micrometeoroids-and-orbital-debris-mmod/>
- [15] NASA, “Space Debris Elimination (SpaDE),” NASA TechPort Project 4347. <https://techport.nasa.gov/projects/4347>
- [16] NASA Technical Reports Server. <https://ntrs.nasa.gov/>
- [17] NASA TechPort. <https://techport.nasa.gov/>
- [18] World Intellectual Property Organization, “PATENTSCOPE.” <https://patentscope.wipo.int/>
- [19] European Patent Office, “Espacenet Patent Search.” <https://worldwide.espacenet.com/>
- [20] Google, “Google Patents.” <https://patents.google.com/>
- [21] United Nations Office for Outer Space Affairs, “Space Debris Mitigation Guidelines.” <https://www.unoosa.org/>

- [22] NASA, “Systems Engineering Handbook,” NASA/SP-2016-6105 Rev. 2, 2016. <https://www.nasa.gov/reference/systems-engineering-handbook/>
- [23] IEEE, “Manuscript Templates for Conference Proceedings.” <https://www.ieee.org/conferences/publishing/templates.html>
- [24] E. Alkhazraji, M. E. Yahia, R. E. Tolba, and N. S. Alharthi, “AI-driven collision risk prediction and debris optimization in low earth orbit,” *Scientific Reports*, vol. 16, art. 27675, 2026, doi: 10.1038/s41598-026-55288-0. <https://www.nature.com/articles/s41598-026-55288-0>
- [25] NASA TechPort, “Space Debris Elimination (SpaDE),” project 4347, updated 2026. <https://techport.nasa.gov/projects/4347>
- [26] NASA TechPort, “Rapid Response Debris Removal Using Reconfigurable Space-Based Laser Networks,” project 156377, updated 2026. <https://techport.nasa.gov/projects/156377>
- [27] NASA TechPort, “Adaptive Just-in-time Collision Avoidance via Targeted Dust Cloud Deployments from Orbital Platforms,” project 156380, updated 2026. <https://techport.nasa.gov/projects/156380>
- [28] “Optimal Active Debris Removal mission planning to inform policy decisions,” *Acta Astronautica*, vol. 228, pp. 224–236, 2025, doi: 10.1016/j.actaastro.2024.11.050.
- [29] “Large-scale object selection and trajectory planning for multi-target space debris removal missions,” *Acta Astronautica*, vol. 170, pp. 289–301, 2020, doi: 10.1016/j.actaastro.2020.01.032.
- [30] “Bi-level evolutionary framework for designing multi-debris active removal missions,” *Acta Astronautica*, vol. 246, pp. 247–257, 2026, doi: 10.1016/j.actaastro.2026.04.002.
- [31] J.-C. Dolado-Perez et al., “Improvement of orbital data precision and accuracy,” *Acta Astronautica*, vol. 229, pp. 218–224, 2025, doi: 10.1016/j.actaastro.2025.01.003.

Reference verification note

Before external submission, verify each title, author, date, version, DOI, and access date against the source. This document is an IEEE-style long-form proposal, not a peer-reviewed IEEE publication. Some cited sources describe active research programmes; their existence does not establish that OCIE is novel or effective.

Final disclaimer

This report does not guarantee novelty, patentability, NASA interest, funding, royalty income, or recognition. No experimental results are fabricated or claimed. All diagrams are conceptual. Obtain technical peer review and legal advice before claiming exclusive ownership or filing a patent.