

Aerospace, Programming Languages, and Information Technology Co-Development: Contemporary Snapshot and Extrapolation

Brendan Sechter

2026-07-24

This twelfth and final article of the series closes the twelve-part treatment of aerospace and computing co-development with a contemporary snapshot as of 2026 and a forward extrapolation across the 2026 to 2050 window. The series framing established in [A237](#) treats aerospace demand and computing capability as coupled through a positive feedback loop across the entire twentieth and early twenty-first centuries. This closing article takes the current state of the coupling as its starting point, identifies the contemporary forcing pressures that are shaping the current phase of co-development, extrapolates plausible trajectories under different assumption sets, and identifies the load-bearing open questions whose resolution will determine which trajectory in fact obtains.

The closer treatment follows the same pattern the earlier series articles established of grounding claims in evidence and marking uncertainty explicitly. Extrapolation across a twenty-four-year forward window is intrinsically uncertain, and the extrapolation strategies treated below are offered as illustrative alternatives rather than as competing forecasts to be adjudicated among. The article's epistemic commitment is that the aerospace-computing coupling as a mechanism continues to operate but that the technological and geopolitical context in which it operates has shifted in ways that produce meaningfully different trajectories from the trajectories that the twentieth-century experience projected.

Current State of the Coupling

The aerospace-computing coupling as of 2026 differs from its twentieth-century form in three respects. First, the direction of dual-use technology transfer has largely reversed. Where twentieth-century aerospace defense procurement subsidized the semiconductor manufacturing base that later commercial computing consumed, contemporary commercial computing supplies the base capability that aerospace-specific suppliers adapt at premium prices. This reversal, treated in [A246](#) and [A247](#), continues to accelerate as commercial computing capability outpaces what specialized aerospace suppliers can independently produce.

Second, the fraction of aerospace platform capability implemented in software rather than in hardware has approached but not reached saturation. The software-mediated aerospace platform treated in [A247](#) represents the mature form of the twentieth-century trajectory, with contemporary combat aircraft, commercial transport aircraft, unmanned aerial vehicles, and spacecraft all considerably software-defined. The saturation asymptote reflects the physical functions including primary structure, engine thermodynamics, and aerodynamic surfaces that necessarily remain hardware artifacts.

Third, machine learning integration into aerospace systems has emerged as the new frontier that was not part of the twentieth-century trajectory. The technical, engineering, and regulatory questions that machine learning raises for safety-critical aerospace applications

are distinct from the questions that the safety-critical software engineering discipline treated in A245 was developed to answer, and the engineering framework for reasoning about machine learning components remains under active development in the institutional bodies that produce industry consensus process standards. Contemporary aerospace machine learning applications are largely confined to non-safety-critical functions including image analysis for surveillance imagery, mission-support decision aids, and predictive maintenance. Extension to safety-critical functions remains a considerable open question.

Machine Learning in Aerospace Systems

Machine learning integration into aerospace systems has grown significantly across the 2020s driven by the commercial capability of contemporary deep learning models and by the operational demand for capabilities that traditional software engineering approaches cannot economically provide. The application domains where machine learning has achieved production deployment in aerospace include perception including radar signal processing and computer vision for target recognition and terrain following, mission-support decision aids including flight-path recommendation and threat-assessment assistance for human operators, and predictive maintenance including component-failure prediction from sensor time-series data.

The engineering integration of machine learning components into aerospace systems raises distinct questions that traditional software verification and validation was not developed to answer. Machine learning components are typically trained from data rather than programmed from specifications, which means that the traditional specification-implementation-verification chain treated in A245 does not apply directly. The data distribution under which a machine learning component was trained may not match the distribution under which it operates, producing failure modes that the training and validation processes did not exercise. The decision boundary of a trained machine learning component is typically not humanly interpretable at the level of detail required to argue that failure modes cannot occur.

The compute cost of training contemporary machine learning models has grown at a substantially faster rate than the general Moore’s Law trajectory. Per the empirical analysis in Sevilla et al. 2022, the compute used to train the largest machine learning models has doubled approximately every 6 months in the deep learning era from 2010 onward, a rate roughly four times the semiconductor-density doubling rate treated in A237. The compute demand for training a model of parameter count N under approximate scaling-law assumptions is

$$C_{\text{train}}(N) \approx 6ND$$

with D the number of training tokens and the factor 6 accounting for forward and backward passes and the gradient computation cost. For contemporary frontier models with parameter counts of order 10^{12} and training-token counts of order 10^{13} , the training compute reaches order 10^{25} floating-point operations, or approximately 10^{16} times the total operation count of an Apollo mission treated in A242. The relationship between achieved test loss and the resources deployed follows an approximate scaling law of the form

$$L(N, D) \approx \frac{A}{N^\alpha} + \frac{B}{D^\beta} + L_{\text{irreducible}}$$

with exponents α and β empirically in the range 0.2 to 0.4 across the deep-learning era, per the analyses in Kaplan et al. 2020 and the subsequent compute-optimal refinement in Hoffmann et al. 2022 introducing the balance between parameter count and training data that produces optimal loss for a fixed compute budget. This scaling law markedly shapes

the economic and infrastructural planning that machine learning integration into aerospace applications requires.

The engineering framework for reasoning about machine learning correctness in safety-critical aerospace applications remains under active development. Approaches include restricting machine learning components to non-safety-critical functions with human oversight for any safety-relevant outputs, developing formal verification techniques adapted to machine learning components including neural-network verification tools, and constructing runtime monitors that check machine learning outputs against traditional-software correctness constraints. Each approach has limitations and none has yet reached the maturity that the industry consensus process standards for traditional safety-critical software achieved by the 1990s.

Autonomous Swarming

Autonomous swarming is the operational capability of coordinating significant numbers of unmanned aerial vehicles to accomplish missions collectively that single vehicles could not accomplish individually. The technical prerequisites for autonomous swarming include low-cost expendable airframes, low-cost onboard computing sufficient for local decision-making, low-cost communication for inter-vehicle coordination, and mission-planning software that can specify swarm behavior at a level of abstraction above individual vehicle assignments.

The operational deployment of autonomous swarming reached substantial scale during the 2022 to 2026 period through the experience of the war in Ukraine, where inexpensive commercial-derivative drones were used at scale for both surveillance and strike missions by both parties to the conflict. The operational lessons include the value of expendability over sophistication for many mission profiles, the vulnerability of high-value platforms to inexpensive threats, and the procurement-agility advantage of commercial-derivative systems over traditional defense-procurement systems.

The mathematical scaling of swarm capability with swarm size follows several competing regimes. For simple coordination tasks the swarm capability grows approximately linearly with size,

$$Y_{\text{swarm, linear}}(N) \approx N \cdot y_1$$

for individual vehicle capability y_1 . For coordination tasks that scale with pairwise interaction between swarm members, the capability grows quadratically as

$$Y_{\text{swarm, pairwise}}(N) \approx \binom{N}{2} \cdot y_{\text{interact}} \approx \frac{N^2}{2} \cdot y_{\text{interact}}$$

for per-pair interaction value y_{interact} . For adversarial engagements the capability may grow super-linearly if the swarm can concentrate mass against an opponent's defenses faster than the defenses can adapt. The scaling regime for a mission determines the size beyond which additional vehicles produce diminishing marginal capability.

The defense-industry implications of autonomous swarming remain subject to marked ongoing engineering, doctrinal, and geopolitical debate. Some analysts including [Scharre 2023](#) treat autonomous swarming as a fundamental shift in the character of aerospace warfare that requires extensive changes in force structure, procurement practice, and operational doctrine. Other analysts treat autonomous swarming as an incremental capability that complements rather than replaces existing high-value aerospace platforms. Both positions have advocates among senior defense officials in 2026, and the resolution of the debate will shape defense procurement for the following decade.

Contemporary Forcing Pressures

Three forcing pressures are shaping the contemporary aerospace-computing coupling beyond the ongoing machine-learning-integration and autonomous-swarming trajectories treated above.

The first pressure is semiconductor supply concentration. The geographic concentration of leading-edge semiconductor manufacturing in Taiwan through the dominance of Taiwan Semiconductor Manufacturing Company hereafter TSMC in advanced-node foundry services creates a supply-chain risk for both commercial and defense computing that neither the twentieth-century Silicon Valley pattern nor the earlier defense-procurement pattern faced. The concentration can be quantified using the Herfindahl-Hirschman index

$$H = \sum_{i=1}^N s_i^2$$

for market shares s_i of N suppliers with $\sum s_i = 1$. For leading-edge foundry services in 2026 with TSMC at approximately 90 percent market share, Samsung at approximately 8 percent, and Intel Foundry at approximately 2 percent, $H \approx 0.81$, appreciably above the 0.25 threshold typically identified with highly concentrated markets. The historical treatment in [Miller 2022](#) documents the commercial and geopolitical dynamics that produced this concentration. The reshoring and diversification programs that the United States, European Union, Japan, and other governments launched in the early 2020s aim to reduce this concentration but require decade-scale investment to achieve sizable results.

The second pressure is quantum computing. Practical quantum computers capable of executing quantum algorithms including the integer-factorization algorithm of [Shor 1994](#) and Grover’s algorithm for unstructured search would have appreciable implications for aerospace-relevant cryptography and computation. Contemporary quantum computing capability as of 2026 remains greatly below what these algorithms would require for practical operation, though the state of the noisy intermediate-scale quantum era treated in [Preskill 2018](#) represents a transitional phase in which quantum devices can perform specialized computations but not the general-purpose quantum computation that would break contemporary cryptography. The development trajectory has been faster than most contemporary analysts expected in the mid-2010s and the transition timeline remains uncertain. Contemporary aerospace software programs are beginning to adopt post-quantum cryptography for long-lived communications systems in anticipation of eventual quantum-computer availability, per the engineering guidance in [Bernstein and Lange 2017](#) published in Nature.

The third pressure is cyber-physical security. The integration of aerospace platforms into networked infrastructure and the increase in software-mediation treated in [A247](#) together produce an attack surface that traditional aerospace security engineering was not developed to address. The general character of contemporary state-level cyber conflict, including its implications for critical infrastructure, is treated in [Sanger 2018](#). Contemporary aerospace security engineering must integrate the traditional physical-security approach with cyber-security approaches drawn from commercial information technology, and the institutional arrangements for doing so remain under active development.

Extrapolation Framework and Forward Projection

Forward extrapolation of the aerospace-computing coupling across the 2026 to 2050 window can proceed under several plausible framework applications. Any forward projection is subject to compounding uncertainty that grows with the projection horizon. For an initial uncertainty σ_0 at the projection origin, the extrapolation uncertainty at time t typically

satisfies

$$\sigma(t) \gtrsim \sigma_0 \cdot (1 + \gamma t)$$

with growth-rate γ of order 0.03 to 0.10 per year for technology and geopolitical forecasts, giving a factor 2 to 4 increase in uncertainty across the twenty-four-year projection window. The six-axis framework from [A237](#) applied forward suggests that each axis will continue to develop under pressures that the twentieth-century trajectory did not anticipate.

Numerical computation demand will continue to grow, driven by machine learning training and inference workloads that exceed traditional aerospace computation by many orders of magnitude. The fraction of aerospace platform lifecycle cost consumed by machine learning training compute may exceed the fraction consumed by traditional software development by the 2030s under contemporary trajectory assumptions.

Real-time control will continue to shift toward more sophisticated closed-loop autonomous operation, with machine learning components entering the control loop for applications where the engineering verification framework can be developed to accommodate them. Full autonomous operation of civil-aviation transport aircraft remains considerably beyond the engineering, regulatory, and public-acceptance thresholds that would permit deployment even under favorable trajectory assumptions.

Reliability and verification will continue to develop as an engineering discipline. The expansion required to accommodate machine learning components represents the largest expansion in the discipline since the initial formalization treated in [A245](#), and the engineering framework that emerges will significantly shape the character of contemporary aerospace software through the 2030s and 2040s.

Networking and distribution will continue to expand, with pressure toward higher-bandwidth space-based communications, on-vehicle edge computing, and distributed autonomous coordination for swarming applications. The engineering integration of these capabilities with traditional aerospace networking treated in [A243](#) will produce distinct network architectures that neither pure aerospace nor pure commercial networking traditions individually anticipated.

Software engineering as a discipline will continue to evolve. The integration of machine learning development practices with traditional safety-critical software engineering practices treated in [A245](#) represents an active research and standards-development area whose outcome will substantially shape the character of aerospace software over the extrapolation window.

Semiconductor economics and dual-use will continue to shift. The commercial-to-aerospace dual-use pattern treated in [A247](#) will continue to dominate as commercial computing capability continues to outpace what dedicated aerospace suppliers can independently produce. The geographic-diversification programs launched in the early 2020s will reshape the semiconductor supply landscape but not the general dual-use pattern.

Competing Extrapolation Strategies

The extrapolation strategy that the preceding section applies is one among several plausible alternatives. Alternative strategies emphasize different structural drivers that would produce meaningfully different trajectories under the extrapolation window.

The demographic-projection strategy, exemplified in the general context by [Kotkin 2020](#), emphasizes the effects of population aging in developed economies, migration patterns, and generational succession on the institutional and workforce composition of aerospace and computing industries. Under demographic-projection assumptions, the workforce constraints on

aerospace and computing capability may become binding through the 2030s and 2040s in ways that the technology-projection strategy underestimates.

The resource-limits strategy, associated with the general approach exemplified in [Sachs 2020](#), emphasizes the physical, energy, and material constraints on the rate of technological development. Under resource-limits assumptions, the compute-demand growth that machine learning integration implies may be constrained by electrical power availability, cooling capacity, and semiconductor manufacturing throughput in ways that the current trajectory does not accommodate.

The techno-economic-cycle strategy, exemplified by [Perez 2002](#), treats major technological transitions as approximately fifty-year Kondratiev cycles with characteristic period

$$T_{\text{Kondratiev}} \approx 50 \text{ years}$$

comprising an installation phase of roughly 25 years, a deployment phase of roughly 25 years, and eventual crisis and transition to the next cycle. Under techno-economic-cycle assumptions, the machine learning integration wave that dominates contemporary aerospace-computing discussion is the beginning of a new cycle whose institutional and productive implications will unfold across the following decades.

The energy-transition strategy, exemplified by [Smil 2022](#), emphasizes the physical and energy foundations of technological civilization and treats aerospace-computing developments as markedly constrained by underlying energy availability. Under energy-transition assumptions, the ability of contemporary computing to continue its historical trajectory depends appreciably on the outcome of the electrical-power supply and cooling infrastructure development required to support it.

The geopolitical-positioning strategy, exemplified by [Zeihan 2022](#), emphasizes the effects of demographic, geographic, and institutional factors on the competitive position of major powers and blocs. Under geopolitical-positioning assumptions, the direction of aerospace and computing development in the extrapolation window depends greatly on the configuration of great-power competition, alliance structures, and industrial-policy choices that different major powers pursue.

None of these strategies is exhaustive or definitive. Each captures a structural feature of the extrapolation problem that other strategies underweight. Reasonable forward projection combines elements of multiple strategies with explicit acknowledgment of the uncertainty each contributes.

Load-Bearing Open Questions

Several open questions will considerably shape the aerospace-computing coupling across the extrapolation window. The list is not exhaustive.

Whether machine learning components can be integrated into safety-critical aerospace functions under some engineering framework that produces acceptable reliability targets remains open. The answer will significantly determine whether contemporary aerospace platforms transition toward considerable machine-learning mediation or remain constrained to non-safety-critical machine-learning applications.

Whether autonomous swarming reshapes aerospace warfare in the ways that some analysts predict remains open. The answer will determine major defense procurement decisions across the extrapolation window and will shape the character of the twenty-first-century aerospace industry.

Whether semiconductor supply concentration in Taiwan is stable across the extrapolation window remains open. The answer depends on both geopolitical developments and reshoring and diversification progress, and the outcome will shape both aerospace and commercial computing.

Whether quantum computing crosses practical-utility thresholds within the extrapolation window remains open. The answer will determine whether aerospace-relevant cryptography and computation must transition substantially or whether contemporary cryptographic and computational infrastructure remains usable across the window.

Whether the commercial-computing capability trajectory continues at its contemporary rate or slows markedly remains open. The answer depends on both technical questions including semiconductor scaling limits and economic questions including capital availability for continued fabrication capacity expansion.

Framework Application to the Contemporary Era

The six-axis framework introduced in [A237](#) applies to the contemporary aerospace-computing coupling with axis weightings reflecting the character of the 2020s.

The first axis is numerical computation demand. Aerospace numerical demand is significant but small compared to contemporary machine learning training and inference demand, with the implication that aerospace-computing coupling now includes commercial machine learning infrastructure as a significant new participant.

The second axis is real-time control. Traditional real-time aerospace control continues to dominate contemporary aerospace platforms, with machine learning integration beginning to enter the loop for well-bounded applications.

The third axis is reliability and verification. The engineering discipline of safety-critical software continues to develop, with machine learning integration representing the largest expansion since the discipline's initial formalization.

The fourth axis is networking and distribution. Contemporary aerospace platforms are appreciably networked, with ongoing development in space-based communications, on-vehicle edge computing, and distributed autonomous coordination.

The fifth axis is software engineering as a discipline. Contemporary aerospace software engineering integrates traditional safety-critical practices with emerging machine learning development practices, with the integration remaining under active development.

The sixth axis is semiconductor economics and dual-use. Contemporary aerospace consumes commercial computing capability adapted for aerospace-specific requirements, with the earlier defense-to-commercial dual-use pattern greatly reversed.

Conclusion

The aerospace-computing coupling operates in 2026 through mechanisms considerably similar to those that operated across the twentieth century but under technological, geopolitical, and institutional conditions that differ significantly from the twentieth-century context. The direction of dual-use technology transfer has largely reversed. The software fraction of aerospace platform capability has approached but not reached saturation. Machine learning integration has emerged as a new frontier with distinct engineering and regulatory questions that the twentieth-century safety-critical software discipline was not developed to answer. Autonomous swarming has reached substantial operational deployment. Contemporary forcing pressures including semiconductor supply concentration, quantum computing development, and cyber-physical security requirements are shaping the current phase of co-development in ways that the twentieth-century trajectory did not anticipate.

Forward extrapolation across the 2026 to 2050 window depends substantially on the outcome of load-bearing open questions including machine learning integration into safety-critical functions, autonomous swarming's effect on aerospace warfare, semiconductor supply concentration stability, quantum computing threshold-crossing, and commercial-computing capability trajectory sustainability. The extrapolation strategies presented above capture structural features of the problem that different analytical traditions have emphasized, but no single strategy is exhaustive or definitive. Reasonable forward projection combines elements of multiple strategies with explicit acknowledgment of the uncertainty each contributes.

The engineering, institutional, and geopolitical context in which the aerospace-computing coupling continues to operate is more complex than the context in which it was originally established in the mid-twentieth century, but the coupling itself as a mechanism remains recognizable. The historical arc traced across this twelve-article series from pre-war ballistic-table computation through wartime code-breaking, SAGE, aerospace simulation and real-time systems, Apollo, ARPANET, Space Shuttle software, safety-critical software as a discipline, Silicon Valley's origins in defense contracting, and software-defined aerospace to the contemporary snapshot demonstrates the durability of the coupling as an engineering pattern across marked changes in technology, institutions, and geopolitics. The trajectory of the coupling across the next quarter century will markedly depend on the choices that engineers, institutional leaders, and policy makers make in response to the forcing pressures the contemporary context presents.

This concludes the twelve-article series on aerospace, programming languages, and information technology co-development.

References

Books

- [Kotkin 2020](#)
- [Miller 2022](#)
- [Perez 2002](#)
- [Sachs 2020](#)
- [Sanger 2018](#)
- [Scharre 2023](#)
- [Smil 2022](#)
- [Zeihan 2022](#)

Reference

- [NIST Post-Quantum Cryptography Project](#)
- [TSMC](#)

Related Posts

- [A237 Framing and the Co-Development Mechanism](#)
- [A242 The Apollo Guidance Computer](#)
- [A243 ARPANET and Networking Origins](#)
- [A245 Safety-Critical Software](#)
- [A246 Silicon Valley from Defense Contracting](#)
- [A247 Software-Defined Aerospace and Autonomy](#)

Research

- [Bernstein and Lange 2017](#)

- [Hoffmann et al. 2022](#)
- [Kaplan et al. 2020](#)
- [Preskill 2018](#)
- [Sevilla et al. 2022](#)
- [Shor 1994](#)