

The Interoperable Digital Twin Proving Ground

A Standards-Based Foundation for Safe, Scalable Physical AI and Autonomous Systems

Executive Summary

Deploying physical AI and autonomous systems at scale demands a capability the defense community urgently needs to establish: a shared, trusted digital proving ground (DPG) environment for training, testing, and validating embodied AI and autonomy software before real-world deployment. Operational testing is expensive, time-consuming, dangerous, and structurally incomplete—it cannot cover the edge cases that define operational safety. Effective evaluation also requires the ability to team with human-operated and autonomous systems, making that a critical requirement for any digital proving ground.

The Interoperable Digital Twin Proving Ground (IDTPG) addresses this gap through a layered, standards-based architecture that delivers unified proving ground capability across the full product life cycle. This spans six primary use cases: synthetic data generation for AI training; physics-grounded environments for test and evaluation; closed-loop simulation for autonomous verification; side-by-side evaluation of composable defense marketplace offerings; warfighter training for human-machine teaming intuition; and decision support for concept of operations (ConOps) development. Its long-term value rests on interoperability defined at the interface level—not the implementation level—so the proving ground can absorb new world model types as the AI landscape evolves without requiring infrastructure reinvestment.

1. Digital Proving Ground Requirements

Advanced AI represents a step change in machine intelligence, yet training, testing, procurement and deployment planning of physical AI and autonomous robots in the real world is expensive, time-consuming, dangerous, and incomplete. These four constraints compound: field exercises consume scarce resources; dangerous failure modes are the least safe to produce deliberately; and even extensive testing cannot cover the combinatorial space of conditions that define robustness at scale.

The Data Gap

When it comes to cyberphysical systems, the problem is not a shortage of raw data—it is a shortage of the right data: scenarios that are rare, hazardous, or operationally sensitive; edge cases adversaries will exploit; conditions requiring precise programmatic control to generate meaningful training signals. For autonomous systems, incompleteness is a structural problem. System behavior is a function of scenario conditions, operating environment, and mission context, making the test space combinatorially unbounded. No live test program can cover it by definition, regardless of budget. Real-world collection is structurally incapable of filling this gap at the pace modern AI development demands.

The IDTPG addresses a gap in the simulation landscape, not just a gap relative to live testing. Existing simulation environments are purpose-built for a single program, tool, or domain: assets authored for one simulator rarely transfer to another, there is no shared infrastructure for comparative evaluation across programs. Federating across vendors or classification levels requires custom integration work that often costs more than the simulation itself saves. The result is duplicated investment, incompatible outputs, and no common evidentiary basis for procurement decisions or certification claims.

In order to deliver the key use cases discussed earlier, the following key capabilities are required in a DPG solution:

- **Synthetic data for AI model training:** High-fidelity simulation generates annotated sensor data for perception, reward-based environments for reinforcement learning, and scenario diversity that grounds world model development.
- **Multi-domain simulation environments:** Physics-accurate environments spanning surface, sub, air, and ground domains, backed by execution-environment-agnostic digital twins of platforms, sensors, and payloads.
- **Rapid prototyping and iterative development:** Virtual environments and agent-assisted MBSE allow autonomy stacks to be built, tested, and revised at the speed of AI-assisted development..
- **Closed-loop test and evaluation:** End-to-end testing across software-, AI-, and hardware-in-the-loop (SIL/AIIL/HIL) configurations and live-virtual-constructive (LVC) integration, including contested scenarios such as electronic warfare (EW), GPS-denied, and multi-agent and swarm environments.
- **Autonomy and AI/ML model V&V:** Systematic behavioral, edge-case, and drift testing—integrated into secure continuous integration/delivery (CI/CD) and automated evaluation pipelines—provides a continuously updated technical evidence base for safety and performance claims.
- **Fleet telemetry, mission replay, and debug analytics:** Recorded run data enables reproducible failure diagnosis and continuous improvement of both models and scenarios.

- **Mission effectiveness scoring and operational certification:** Standardized metrics translate simulation results into actionable insights that program offices and oversight bodies required to implement fast-turn processes such as continuous authority to operate (ATO).
- **Procurement and comparative analysis:** Common scenarios and objective metrics support side-by-side evaluation of candidate systems before acquisition decisions are made.
- **Warfighter training for human-autonomy teaming:** Simulation provides realistic practice for managing and collaborating with autonomous systems, building operator intuition about when to trust, override, and direct AI teammates.
- **Decision support for ConOps:** Mission planners can explore, stress-test, and refine concepts of operation in simulation before committing to doctrine, force structure, or hardware investment.
- **Zero-friction adoption:** A proving ground that requires IT work to produce output will not be used. The DPG must meet users where they already work — embedded in C2 tools, mission dashboards, and engineering IDEs — with no integration burden. The benchmark is the AI subscription tools that non-technical users adopt without IT involvement.

These requirements informed the architecture that follows. No single program needs all of these capabilities equally—but the IDTPG infrastructure must support all of them intrinsically and without requiring custom integration for each. Doing so demands specific structural properties that run across the capability stack: modular, composable assets that can be reused across programs without custom integration; an accessible interface layer that doesn't require simulation expertise to operate; federated simulation that closes the sim2real gap across domains; and containerized access controls that protect IP while enabling multi-vendor participation.

2. IDTPG Architecture

The IDTPG is a specific architecture for realizing a DPG — distinguished by interoperability defined at the interface level and digital twin standards as the shared asset foundation. It consists of three functional layers connected by two interoperability interfaces—each designed for composability, openness, and accessibility.

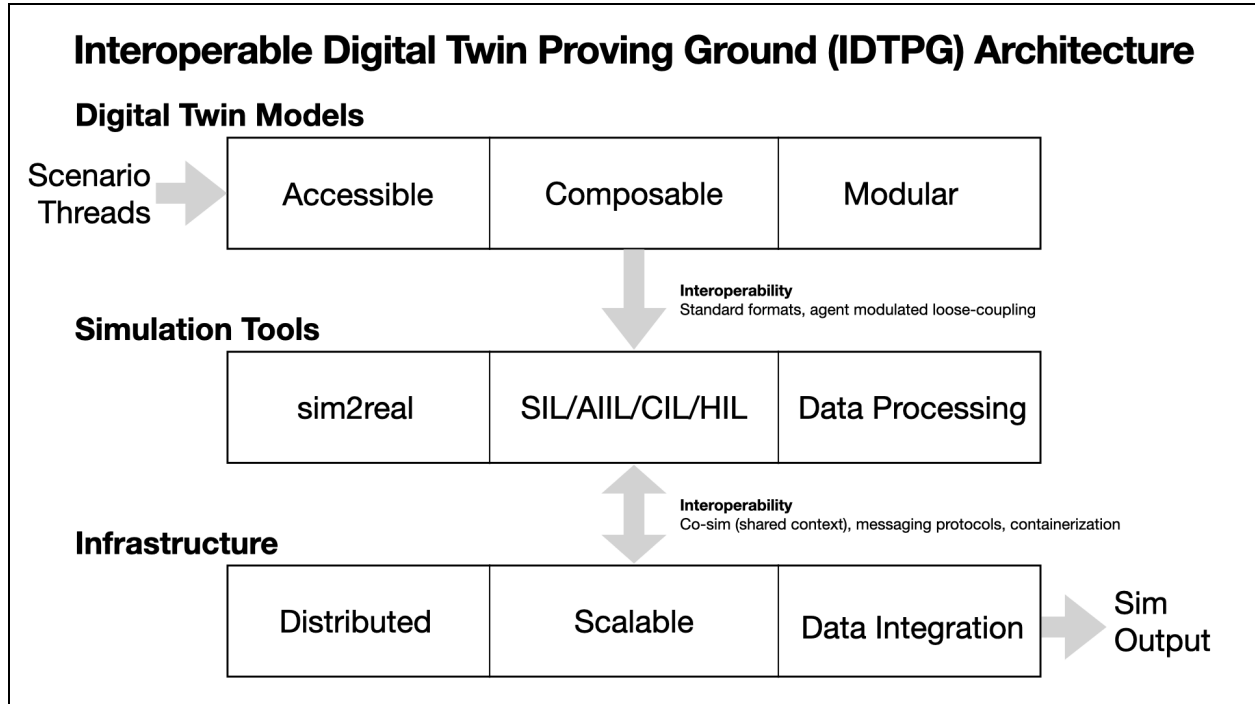


Figure 1. IDTPG Three-Layer Architecture. Scenario threads composed using reusable digital twins (Models layer) are simulated using cooperative tools (Simulation layer) and the resulting data is scalably and efficiently harvested (Infrastructure layer). Two interoperability interfaces govern cross-layer communication.

Modularity and composability in the **Digital Twin Models** layer convert one-time asset development into durable digital infrastructure. A modular twin—geometry, semantics, behavior, and sensor subsystems encapsulated independently—can be recombined with any conformant environment or payload without custom integration. A program that builds a high-fidelity platform twin for its own test and evaluation (T&E) simultaneously contributes an asset every future program using that platform can inherit. This reuse dynamic only compounds if assets are discoverable and maintained, which is the function of a digital twin catalog: a shared, versioned inventory organized by domain and fidelity class that transforms twin development from a per-program cost into an amortized institutional investment.

A proving ground that requires 3D simulation expertise to operate will never achieve the adoption it needs. **Accessibility** is therefore an architectural requirement: modularity, composability, machine-readable formats, and agent-accessible model context protocols (MCPs) across every building block of the IDTPG stack allow LLMs and AI agent workflows to drive simulation on behalf of users who lack deep sim experience, dramatically lowering the barrier to entry for program offices, analysts, and operators alike.

The same MCP layer that enables LLM-driven simulation also enables flexible coupling to existing workflows promoting **adoption**. Mission planning interfaces, C2 systems, and engineering IDEs can connect to the IDTPG as a service capability rather than requiring users to enter a separate simulation environment.

A proving ground that does not accurately predict real-world behavior provides false confidence rather than reduced risk. Closing the **sim2real** gap is therefore the simulation layer's central obligation: the tools that power a DPG must be physically grounded, and must remain precise and accurate across the full range of domains, fidelity levels, and operating conditions the proving ground is expected to cover. No single simulation tool meets this bar across all domains—a physics engine that excels at rigid-body dynamics may be inadequate for electromagnetic propagation or underwater acoustics. **Federation** of simulation tools is not an architectural preference; it is a practical necessity. The layered nature of IDTPG's architecture connected by interoperability interfaces is built around this core principle.

Federation takes two distinct forms that serve different needs. **Cosimulation** couples tools tightly, advancing a shared world state at each simulation step—every engine sees the same ground truth at every moment, which is essential when cross-domain interactions drive the outcome (an EW effect degrading a sensor that controls an actuator, for example). Standardized **messaging protocols** and pub/sub architectures such as Robot Operating System (ROS2) or High-level Architecture (HLA) couple tools at a higher level, exchanging state updates asynchronously across nodes—better suited to large-scale distributed scenarios where tight synchronization is impractical. Given the breadth of DPG requirements—from closed-loop HIL integration to large-scale swarm evaluation—both forms must be supported, with the choice governed by specific domain & use cases as well as fidelity & accuracy thresholds.

Simulation at proving ground scale generates continuous, high-volume data streams that must be securely processed, stored, and governed. Access control is not a perimeter concern—it is structural: credentials must govern access at the level of individual digital twin assets, mission threads, simulation tools, and generated datasets, ensuring that sensitive program data remains appropriately compartmented across a shared infrastructure. **Containerization** enforces these boundaries while simultaneously enabling the automated workflows that make scale achievable. Two patterns illustrate the range: parallelized simulation runs executing stochastic variations across scenario parameters—necessary to characterize the non-deterministic behavior of generative AI models—and large-scale distributed simulations coordinating blue and red team entities across many simultaneous agents. Both place substantial demands on the underlying cloud infrastructure. A state-of-the-art DPG must be architected in close coordination with cloud resource management to sustain the GPU throughput, data bandwidth, and network performance these workloads require.

A proving ground that requires participants to expose their proprietary autonomy stacks in order to engage will not achieve the participation it needs. Yet exercising those stacks in SIL/AIIL/HIL—testing how systems perform under diverse conditions and in concert with other autonomous systems—is precisely the proving ground's highest-value function. Containerization resolves this tension: with **secure access controls and entitlements**, participating teams can engage fully in federated simulations while limiting visibility into their software, models, and behavioral logic to only what is necessary for the scenario to execute. This architectural separation is what makes multi-vendor, multi-program engagement viable without requiring competitors to reveal proprietary technology to one another or to the government. It should be noted, however, that containerization addresses the technical dimension of IP protection—it is

equally important to resolve the contractual dimension through clear IP and data rights frameworks negotiated when contributors are onboarded into the DPG.

2.1 IDTPG's Digital Twin Model Layer: DTES

Interoperability in the Digital Twin Models layer requires a standard all participants can adopt without sacrificing tool flexibility. Universal Scene Description (USD) has emerged as the dominant open standard for 3D data interchange precisely because it was designed for interoperability at scale—not for a single industry, but across manufacturing, robotics, autonomous vehicles, industrial simulation, and AI development, with broad and growing toolchain support. The Digital Twin Encapsulation Standard (DTES) builds on this foundation, defining what a twin is, not how it will be used. DTES encapsulates geometry, semantics, behavioral parameters, and sensor subsystems in an execution-environment-agnostic format, organized into three categories that map directly to defense contexts: Space Twins (terrain and environments), System Twins (autonomous platforms), and Item Twins (sensors and payloads). Any combination can be assembled into a scenario without custom integration.

Protecting Digital Investment

Three properties make DTES durable: encapsulation reduces integration friction at every level of the platform hierarchy; extensibility via USD schemas supports defense-specific metadata without forking the standard; and non-destructive composition arcs enable platforms to be assembled from reusable subsystem twins—and scenarios assembled from reusable platforms—without modifying any source asset. A subsystem twin, such as a sensor or payload, developed for one platform can be directly plugged into another compatible platform. DTES-compliant assets are not locked to any simulation vendor—as the tool landscape evolves, the investment survives at every level of the hierarchy, from individual subsystems to complete mission-ready platforms.

2.2 Mission Threads and Scenario Standards

Open standards are required not only for entity models but for mission threads themselves. A scenario thread—the composition of twins, environmental conditions, behavioral scripts, and evaluation parameters that defines what is being simulated—is a shareable, versionable artifact in its own right. The defense community must be able to exchange mission threads with the same provenance guarantees it applies to platform and capability twins. Without this, scenario development becomes duplicated work rather than shared infrastructure, and institutional knowledge of what was tested, under what conditions, and with what outcomes disappears when a program closes.

USD is a natural foundation for scenario definition. Its composition arc mechanism allows all model properties—environmental conditions, behavioral parameters, sensor configurations—to be defined within the scenario hierarchy without requiring underlying models to be duplicated or modified for each use case. A platform twin remains canonical; the scenario overrides only what

is specific to that engagement. As with digital twin encapsulation, the entire scenario thread stays cohesive rather than fragmenting across files or tools, and can be reliably versioned, audited, and secured with the same rigor applied to the assets it references.

2.3 Reference Simulator: AFSim

Within the Simulation Tools layer, a reference simulator—a common, authoritative baseline that programs can test against and results can be compared across—is essential for making evaluation meaningful at enterprise scale. AFSim, the Advanced Framework for Simulation, plays this role across the defense autonomy community: a government-accessible, high-fidelity constructive simulation environment with broad adoption across programs, commands, and domains that makes it a natural simulator of record for the IDTPG.

AFSim's value compounds when federated with dual-use simulation platforms—commercial-grade engines developed for industrial robotics, autonomous vehicles, and AI research that bring physics fidelity, sensor modeling, and generative environment capabilities that purpose-built defense tools were not designed to provide. Where AFSim excels at modeling force-on-force engagements, mission-level behaviors, and joint operational contexts; dual-use commercial-grade simulation engines developed for industrial robotics, autonomous vehicles, and AI research extend the proving ground into high-resolution perception training, hardware-in-the-loop integration, and scenario generation. Together they expand the range of problems a DPG can credibly address—from platform-level autonomy development to theater-level ConOps analysis—within a single shared scenario thread.

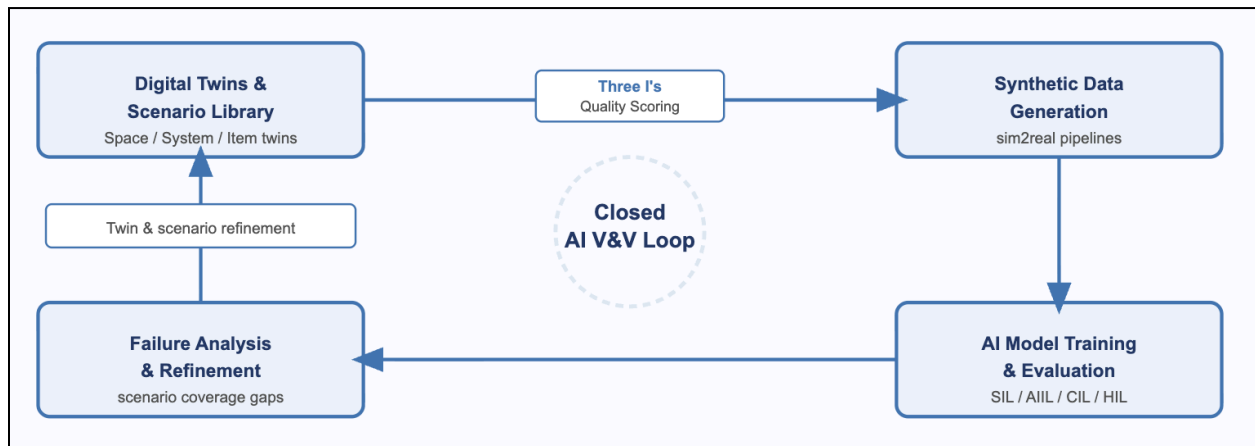
Realizing this federation depends on the same standardized models and protocols that underpin the rest of the IDTPG architecture. Common digital twin formats ensure that assets authored for an AFSim scenario are directly consumable by a physics engine or generative simulator without translation. Standardized messaging protocols ensure state is exchanged consistently across tools regardless of their origin. Without these contracts, combining AFSim with dual-use platforms produces integration work that scales with every new tool added; with them, each new simulator becomes an additive capability rather than a new integration problem.

3. DPG Future Proofing

The AI landscape is evolving faster than any procurement cycle. A DPG built around today's tools and model architectures will be obsolete before it reaches full deployment—making future-proofing not a nice-to-have but a design requirement from day one. Ensuring that synthetic data will remain fit for purpose as sensor technology and processing capabilities advance requires objective quality metrics, such as those defined in the Three I's Framework.

The same agent-accessible infrastructure that makes the DPG usable also makes it extensible. MCP interfaces that allow a non-expert to generate a synthetic dataset today allow automated

pipelines to run scenarios, evaluate outputs, and trigger downstream workflows without human intervention tomorrow. Model-Based Systems Engineering (MBSE) as traditionally practiced means humans working in rigid drawing syntax within vendor-locked tools — heavyweight, slow, and disconnected from simulation. SysML v2 changes this: its textual syntax is purpose-built for agentic workflows and natural human-machine interaction, allowing architecture, requirements, behavior, and logic decisions to be jointly developed by human teams and AI in the same interface that drives the DPG. The resulting model is not separate from the digital twin but rather incorporated within it, alongside USD and DTES, as a formal, machine-readable specification that agent workflows can act on directly to generate scenarios, configure test runs, and feed results back into the architecture. This is *MBSE 2.0*: system design and digital proving ground unified in a single, continuously updated model as a collaborative flywheel for system engineers and AI.



*Figure 2. Closed AI Verification & Validation (V&V) Loop. Digital twins generate scenarios; Three I Framework's objective scoring validates data quality before training; model failures feed back to refine twin fidelity and scenario coverage, enabling continuous improvement with **loop engineering**.*

As AI model architectures diversify, so does the challenge of ensuring simulation remains a valid proxy for the real world. The world model taxonomy spanning explicit models—physics engines, digital twin composition, structured representations—through to implicit models such as vision-language-action (VLA) backbones, reasoners, and generative video predictors, each carries a distinct sim2real or **gen2real** gap with different characteristics and mitigation strategies. A future-proof DPG must support grounding and tuning workflows for this full spectrum: providing the real-world reference data, evaluation benchmarks, and closed-loop feedback mechanisms each model type requires to remain calibrated to operational conditions. Interface-level interoperability is what makes this sustainable—a DPG that defines clean contracts around what a world model exposes, rather than how it works internally, can absorb new model architectures as they emerge without requiring infrastructure reinvestment.

As AI accelerates MBSE, we also need to consider formal program based Verification, Validation, and Accreditation (VV&A) since it could quickly become the bottleneck to rapid

deployment. In this context, IDTPG serves as a complementary process expanding the testable domain beyond what live range time and operational safety constraints permit. The result is an analytical evidence base that formal accreditation can then validate by arriving at live T&E with higher confidence, narrower uncertainty, and more targeted test objectives. For autonomous systems whose behavioral variability across scenario conditions makes exhaustive live testing structurally impossible, this complementary role is not optional — the digital proving ground is the only credible path to the test coverage that safe, scalable deployment requires.

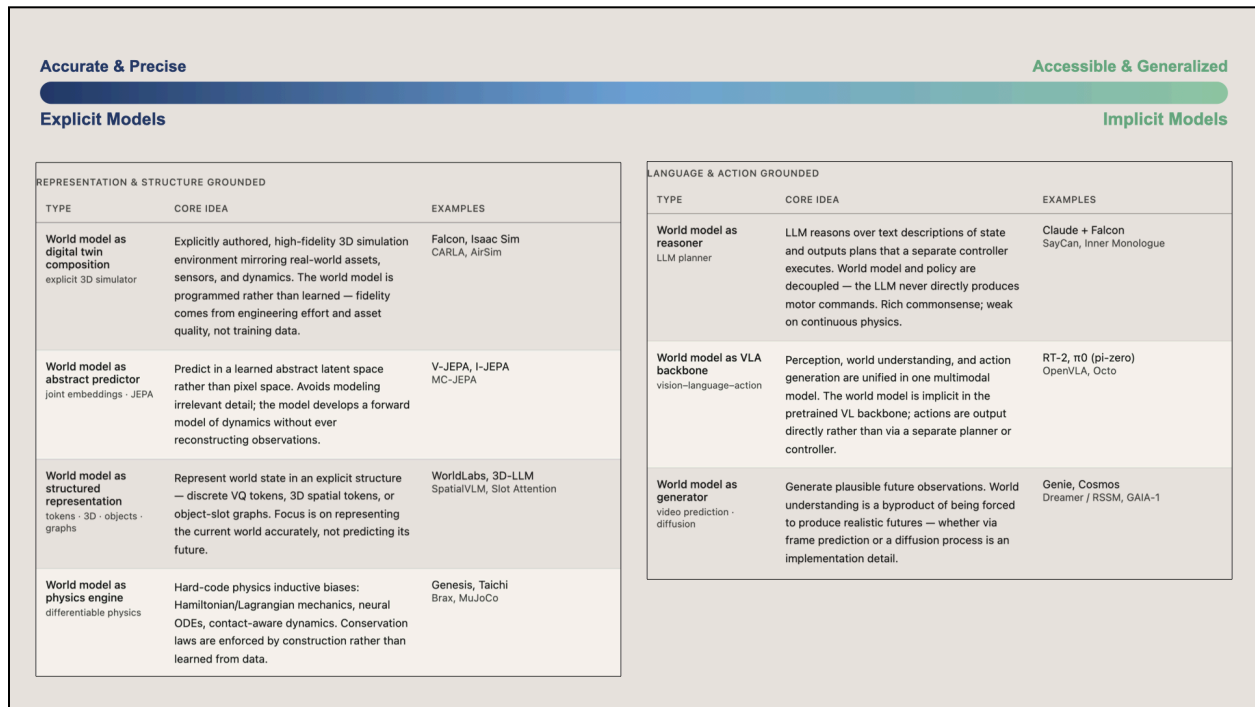


Figure 3. World Model Taxonomy. Seven paradigms span the explicit-to-implicit spectrum. All world models must capture both individual entities (subsystems, platforms, environments) and the interactions between them; explicit models compose these layers deliberately, implicit models learn them jointly. Each paradigm carries distinct sim2real or gen2real gap characteristics, and a future-proof DPG must support the full range. Worldmodel Kitbashing encompasses hybrid pipelines that combine world model types within shared IDTPG scenario threads.

4. Conclusion

Some major themes have emerged consistently across the defense ecosystem's engagement with digital proving grounds and at recent major industry workshops on defense robotics and autonomy digital proving grounds: the need for low barriers to entry that don't burden industry participation; standards defined and published upfront as specific artifacts, not slogans; physics-fidelity environments that replicate real operational conditions rather than clean-room scenarios; portable outputs with open interfaces even where tools remain proprietary; and IP frameworks addressed contractually before industry is asked to participate. That these themes surface independently and repeatedly is itself a signal—the community has reached sufficient

maturity to know what it needs, even where the shared interoperability foundation to deliver it does not yet exist.

Each of those requirements maps directly to what the IDTPG architecture is designed to address.

- The demand for low barrier to entry is answered by agent-accessible model context protocol MCPs; flexible coupling with existing workflows; LLM-driven simulation interfaces, and a digital twin catalog that makes institutional assets discoverable without requiring 3D expertise.
- Standards defined upfront as specific artifacts is the argument for USD and DTES: an open format that industry can build to before the government shows up with a contract.
- Physics-fidelity in real operational conditions drives the federated simulation architecture—multi-domain physics, contested environment modeling, and AFSim as a reference simulator.
- Portable output with open interfaces is the IDTPG's core architectural contract resulting in programs retaining their tool choices, but the interfaces between them remaining open, defined, and enforced — interoperability at the seams, flexibility everywhere else.
- IP protection determines whether industry participates. In addition to clear contractual IP and data rights negotiated before onboarding, IDTPG's architecture addresses this through strict entitlements and limiting visibility to proprietary stacks during federated simulation.

The technical components are available — cloud infrastructure, simulation tools, digital twin platforms, open standards. The convergence is real. What remains is the institutional commitment to bind them together: defining and enforcing cross-model interface standards, establishing a shared digital twin catalog, and resolving IP frameworks before the ecosystem is asked to contribute. Done in that order, the IDTPG becomes what the defense autonomy community actually needs: not another simulation environment, but a shared foundation that makes every program's investment more durable, every test result more comparable, and every autonomous system safer before it ever enters the field.

References

Hudson Department of Navy, Rapid Capabilities Office workshop, 2026.

PAE RAS Industry Day Workshop. Requirements and Convergence Themes for Defense Robotics and Autonomy Systems Digital Proving Grounds. 2026.

Shah, A., Leacche, F., De Ioris, R., and Godse, A. "The Digital Twin Encapsulation Standard (DTES): A USD-Based Open Standard for Simulation-Ready Digital Twins." *IIITSEC*, Paper No. 23352, 2023.

Duality AI. "The Three I's Framework: Objective Metrics for Synthetic Data Quality in AI and Autonomy Applications." Duality AI White Paper, 2024.

Shah, A. "Accelerating Physical AI Deployment by Closing the Gen2Real Gap." Duality AI Blog, 2025. <https://www.duality.ai/blog/gen2real>

Shah, A. "World Modeling and Vibe Simulation with Unreal Engine: The Tinkerer's Guide to Kitbashing World Models." *Unreal Fest 26 Chicago*, June 18, 2026.

Venditti, S. "An Open Standards Foundation for Modeling & Simulation." LinkedIn, May 2026. <https://www.linkedin.com/posts/activity-7458575398541365248-t6Wh>

Air Force Research Laboratory. "Advanced Framework for Simulation, Integration and Modeling (AFSim)." AFRL Aerospace Systems Directorate, Wright-Patterson Air Force Base, OH. <https://www.wpafb.af.mil/afrl/>

Pixar Animation Studios. "OpenUSD: Universal Scene Description." Academy Software Foundation Open Source Project, 2016. <https://openusd.org>

Office of the Under Secretary of War, Research and Engineering. "Verification, Validation, and Accreditation (VV&A) Recommended Practices Guide (RPG)." OUSW(R&E), 2025. https://www.cto.mil/sea/vva_rpg/

Duality AI. "Falcon Digital Twin Simulation Platform." <https://www.duality.ai/product>

NVIDIA Corporation. "NVIDIA Omniverse: A Platform for Developing Physical AI and Industrial Digital Twin Applications." <https://www.nvidia.com/en-us/omniverse/>