

117-266-D Contrasting AI Teamwork with Systems of Systems

From Architectural Integration to Runtime Stability

A foundational chapter for the Stability Engineering Institute

The ‘AI version of teamwork’ belongs in the intellectual neighborhood of ‘System of Systems engineering’. Both study collectives whose overall behavior cannot be understood by inspecting one member at a time. Both recognize distributed authority, operational independence, evolutionary change, and emergent capability. The resemblance is substantial, especially in defense environments where crewed aircraft, autonomous vehicles, sensors, weapons, communications networks, and command organizations must work together without becoming one centrally owned machine.

The concepts are nevertheless not identical. A System of Systems is a specific architectural condition: its constituent systems remain useful and often independently managed outside the larger arrangement. AI teamwork is a broader runtime condition. Its members may be full systems, small agents, software services, humans, sensors, databases, or components. They become a team when their behavior becomes consequentially coupled, even when no one formally designed, declared, or manages the collective.

System of Systems engineering explains how independent systems combine to deliver mission capability. GUDIYA extends the question to how changing collectives remain stable while they sense, decide, learn, and act together.

Why the Comparison Matters

The comparison prevents two errors.

- The first is to present AI teamwork as though systems engineering had never encountered distributed autonomy, incomplete managerial control, or emergence. The Department of War has worked with those realities for decades.
- The second error is to assume that existing System of Systems methods completely contain the machine-speed stability problem created by adaptive actants. They do not yet make runtime team discovery, cognition velocity, accumulated cognitive state, recursive interaction, or dynamic stability margin their central unit of analysis.

GUDIYA should therefore inherit the mature architectural lessons of System of Systems engineering while naming the additional phenomena introduced by cognition at machine speed. This is an extension of the tradition, not a competing vocabulary exercise.

The System of Systems Foundation

A System of Systems emerges when constituent systems retain meaningful independence yet participate in a larger arrangement that provides capabilities unavailable from any constituent alone. Defense guidance emphasizes practical complications that follow from this structure: constituent systems have their own purposes and lifecycles, multiple managers control decisions, the larger arrangement evolves rather than reaching a permanently finished state, and behavior appears through interaction among the parts.

Five characteristics commonly anchor the concept:

1. Operational independence. A constituent system can perform a useful function apart from the larger arrangement.
2. Managerial independence. A constituent may have its own owner, funding line, priorities, upgrade schedule, and operational authority.
3. Evolutionary development. The larger arrangement changes as members, missions, technologies, and interfaces change.
4. Emergent behavior. Collective capability and collective failure can arise from interaction rather than reside in one constituent.
5. Distribution. Constituent systems may be separated geographically, organizationally, technically, and temporally.

These characteristics make System of Systems engineering unusually compatible with the GUDIYA worldview. They reject the assumption that one designer owns every relevant state and can optimize the whole by decomposing it into obedient parts.

Four Governance Types and Their Grid Implications

DoW's 'System of Systems' practice distinguishes four governance types. They are especially useful for GUDIYA because the Grid cannot assume the same authority relationship in every collective. Stability controls must match the way the participating systems are actually governed.

SoS type	Governance condition	Likely GUDIYA posture
Directed	The larger capability has central purpose and management; constituent systems are directed to participate while retaining independent utility.	A mission Grid can enforce common envelopes, telemetry, rendezvous rules, and recovery actions under explicit authority.
Acknowledged	A recognized manager and objectives exist, but constituent owners retain authority over their systems, funding, priorities, and evolution.	Grid authority must be federated through negotiated participation rules, shared evidence, and preauthorized interventions.
Collaborative	Independent systems cooperate voluntarily and collective decisions arise through agreement rather than central command.	The Grid should emphasize common protocols, mutual visibility, consent-based envelopes, and distributed damping.
Virtual	No central manager or durable common purpose controls the collective; large-scale behavior emerges from otherwise independent participation.	The Grid must first detect the collective from coupling and may rely on ecosystem rules, infrastructure controls, or regulatory authority.

The virtual type is particularly important to the AI version of teamwork. It describes the neighborhood in which an undeclared collective can become operationally real through shared infrastructure, environmental feedback, or repeated indirect interaction. GUDIYA adds the runtime instrumentation needed to recognize that condition before instability becomes the first evidence that a team existed.

The AI Version of Teamwork

The AI version of teamwork begins from coupling rather than organizational membership. A team exists when the behavior of one actant changes the conditions under which other actants perceive, decide, or act toward a shared or consequential outcome. The members may share an explicit mission, pursue partially aligned objectives, or simply react to the same environment. Direct messages are not required. A database update, price signal, sensor observation, route choice, resource reservation, or physical movement can create the coupling.

This definition is intentionally broader than the conventional idea of a team. It includes declared agent groups, but it also includes implicit teams that form through shared infrastructure and environmental effects. It applies to human-machine formations, vehicle fleets, aircraft in common airspace, intelligent products, supply networks, and enterprises connected through markets or data.

The defining feature of AI teamwork is not declared membership. It is consequential coupling among adaptive actants over time.

The Central Contrast

Dimension	System of Systems	AI version of teamwork
Basic unit	Independently useful constituent system	Any consequential actant, including an agent, human, service, vehicle, sensor, component, or composite team
Basis of membership	Participation in a recognized capability arrangement	Active coupling that changes another member's operating conditions
Formation	Usually architected, acknowledged, or operationally recognized	May be declared, inferred, temporary, nested, or emergent during execution
Independent usefulness	Normally central to the definition	Not required; a member may have meaning only inside the collective
Management	Often distributed across programs and authorities	May be distributed, delegated, algorithmic, ambiguous, or absent
Primary concern	Mission capability, architecture, interoperability, governance, and evolution	Runtime behavior, relative motion, cognitive coupling, stability margin, and recovery
Time scale	Often planned across acquisition and mission lifecycles	Can form, morph, and dissolve at machine speed
State	Constituent configuration and mission state	Also includes memory, unfinished work, inferred intent, recursive depth, and accumulated cognitive load
Failure lens	Loss or degradation of SoS capability	Oscillation, amplification, drift, overload, unsafe emergence, authority conflict, or ecosystem cascade

Supervision	SoS engineering and mission governance	Continuous Grid observation, rendezvous management, envelopes, damping, and stabilization
-------------	--	---

The Relationship Is Set Inclusion

The cleanest conceptual relationship is set inclusion. AI teamwork is the broader category because it begins with actants and coupling. A System of Systems is a special case in which the participating actants are themselves systems that remain operationally useful and managerially distinct.

This distinction prevents the term System of Systems from expanding until it means every complicated assembly. Several tightly integrated agents inside one application may constitute a multi-agent system without forming a System of Systems. Components inside a product usually form one system. By contrast, independently useful aircraft, command networks, sensor platforms, and logistics systems assembled for a theater mission present a strong System of Systems case.

Every cognitive System of Systems can be examined as a team of actants. Not every team of actants qualifies as a System of Systems.

Classification Across GUDIYA Examples

Collective	Likely classification	Reason
Specialized agents inside one application	Multi-agent system	The agents may not remain independently useful or independently managed outside the application.
Agents supplied by several firms	Potential cognitive System of Systems	Each service may retain its own purpose, owner, lifecycle, and operating authority.
CCA, MMA, crewed aircraft, sensors, and command networks	Hybrid cognitive System of Systems	Independently useful systems combine temporarily to deliver theater capability.
Autonomous vehicles from different fleets sharing roads and chargers	Collaborative or virtual System of Systems	No single manager controls the whole, but shared infrastructure produces consequential collective behavior.
Software and mechanical components inside one product	Integrated system	The components normally serve the product and are governed by one product architecture.
Supply chains spanning independent firms	Socio-technical System of Systems	Firms remain operationally and managerially independent while producing network-level outcomes.

Architecture and Dynamics

System of Systems engineering begins largely with architecture. It asks which constituent systems contribute to a capability, how their interfaces work, what authority the SoS manager possesses, how changes propagate across programs, and how the arrangement can evolve. These questions remain essential. A Grid cannot stabilize a collective whose members, interfaces, ownership, and mission dependencies are unknown.

GUDIYA adds a dynamic layer. Even a sound architecture can enter an unstable operating regime. Messages can arrive too slowly for one actant and too quickly for another. Several correct agents can create a recursive loop. Independent route optimizers can generate synchronized congestion. Shared memory can preserve stale commitments. Local correction can amplify rather than damp a disturbance. None of these outcomes necessarily indicates a defective component or broken interface.

Architecture describes the possible relationships. Stability Engineering measures what those relationships are doing now, how their effects accumulate, and whether the collective can still recover.

Capability Emergence and Instability Emergence

System of Systems literature rightly treats emergence as a source of mission capability and risk. GUDIYA makes a narrower distinction inside that idea.

- Capability emergence asks what the collective can accomplish that no member can accomplish alone.
- Instability emergence asks whether the interaction creates oscillation, amplification, drift, congestion, deadlock, or collapse that no member intended.

The two forms of emergence can coexist. Increasing connectivity may improve target recognition while creating a faster path for corrupted state. Tighter synchronization may improve coordinated attack while making the formation sensitive to delay. Greater autonomy may preserve mission continuity under lost communications while allowing intent to diverge. The engineering task is to preserve useful emergence without allowing interaction energy to escape recoverable bounds.

Declared Teams and Undeclared Teams

System of Systems arrangements often have a named mission, an acknowledged manager, an architecture, or at least an operational community that recognizes the participating systems. AI collectives can become consequential before any of those structures exist. Several agents may begin reading and writing the same store. Independent defensive tools may respond to the same anomaly and repeatedly trigger one another. Aircraft may re-form around a surviving sensor after communications degrade. Markets can couple autonomous trading systems that were never designed to cooperate.

GUDIYA therefore treats team formation as an observable runtime event. The Grid must infer membership from causal influence, shared state, resource contention, repeated rendezvous, and environmental effects. This creates the requirement for ‘Know Your Team’ alongside ‘Know Your Actant’. Identity remains necessary, but the Grid must also know which collective has formed, why it exists, what binds it, how authority is distributed, and what residue may remain after it dissolves.

Time Becomes a First Class Variable

System of Systems engineering recognizes evolution over time. GUDIYA moves time into the operational state of the collective. Delay, sampling rate, cognition velocity, persistence, repetition, and phase alignment can determine whether the same architecture behaves safely or becomes unstable.

Consider a loop in which Actant A influences B, B influences C, and C later changes the conditions observed by A. Every actant may produce a locally valid response. The collective outcome depends on when each response occurs, how strongly it affects the next actant, which state persists between rendezvous, and whether the actants operate from compatible versions of reality. Static interface correctness cannot answer those questions.

In AI teamwork, time is not merely the interval during which the mission unfolds. Time participates in the behavior of the team.

The Hybrid Fleet as the Bridge Case

The CCA and MMA hybrid fleet provides a useful bridge between the two traditions. It is unmistakably a System of Systems when crewed aircraft, collaborative combat aircraft, massed modular aircraft, sensors, command networks, weapons, tankers, and logistics systems retain independent utility and management while combining for mission capability.

It is also an AI team in the GUDIYA sense. Mission groups form and dissolve during execution. Members have different cognition and motion velocities. Communications degrade. Payload roles change. Human commanders and autonomous systems operate under different observation rates. Losses alter topology, authority, and workload. Indirect coupling through airspace, targeting state, electromagnetic effects, and shared support resources may be as important as explicit messages.

System of Systems engineering provides the architecture and governance vocabulary. GUDIYA provides the runtime stability vocabulary needed to observe and regulate the hybrid fleet as a moving cognitive collective.

How GUDIYA Extends the Engineering Stack

GUDIYA does not replace constituent engineering, systems engineering, or System of Systems engineering. It adds a supervisory stability layer across them.

1. Know Your Actant establishes identity, authority, capability, lineage, and the Actant Cognitive Signature for each participant.
2. Rendezvous Management governs explicit and implicit interactions, including rate compatibility, authorization, timing, shared state, and environmental coupling.
3. Know Your Team detects consequential collectives, including temporary and undeclared teams, and reconstructs their boundaries and dependencies.
4. Cognitive Envelopes define acceptable operating regions for individual actants, rendezvous, teams, and nested Systems of Systems.
5. The GUDIYA Grid supplies an external reference for relative motion, cognition velocity, load, density, recursion, and stability margin.
6. Grid intervention introduces damping, separation, rate limits, quarantine, graceful degradation, or Cognitive SCRAM when recovery margin becomes inadequate.

The resulting architecture respects distributed authority. Constituent managers continue to operate their systems. Mission authorities retain command. The Grid observes cross-boundary dynamics that no constituent can see completely and acts within preauthorized stability rules.

What GUDIYA Should Inherit from DoW Practice

DoW experience offers several disciplines that GUDIYA should adopt rather than rediscover.

- Treat managerial independence as a design constraint, not an inconvenience that centralization will eventually remove.
- Engineer capability through mission threads that cross constituent boundaries.
- Plan for continuous evolution, asynchronous modernization, and uneven technical maturity.
- Separate the authority of a constituent manager from the authority needed to protect the larger mission.

- Use architecture, interface agreements, configuration knowledge, test environments, and operational evidence together.
- Expect emergent effects and organize learning across programs after anomalies and mission changes.

These practices give GUDIYA a credible institutional bridge into defense acquisition and mission engineering. The new contribution is to connect them to continuous telemetry and control of collective stability.

What System of Systems Practice Can Gain from GUDIYA

GUDIYA contributes concepts that become increasingly necessary as constituent systems acquire adaptive cognition.

- Runtime discovery of teams created by actual influence rather than only documented interfaces.
- Cognition-aware telemetry that records intent, state persistence, recursion, delegation, and relative cognition velocity.
- Stability margins that show proximity to oscillation, overload, authority drift, and cascading emergence.
- Persistent lineage across model replacement, host migration, member substitution, and apparent termination.
- Dynamic envelopes that can tighten or relax as mission consequence, density, communications, and recovery capacity change.
- Cross-team controls for instability transferred between nested or neighboring Systems of Systems.

This reframes assurance. A cognitive System of Systems cannot be judged only by whether its constituents pass qualification tests and its mission threads succeed in planned scenarios. It must also demonstrate stable behavior under timing mismatch, rapid reconfiguration, accumulated state, partial observability, indirect coupling, and repeated disturbance.

A Typical Enterprise

Here is a particularly strong enterprise example. It demonstrates how a relatively persistent enterprise System-of-Systems architecture can contain many transient AI teams.

The enduring SoS may include:

- Order Processing;
- Inventory Management;
- Manufacturing;
- Sales and Pricing;
- Logistics and Finance;
- home-grown applications, SaaS platforms, databases and integration services.

These domain systems have their own owners, purposes, vendors, data models, release cycles and operating rules. If they remain meaningfully independent while contributing to enterprise capability, they can reasonably be governed as an acknowledged System of Systems.

The Friday evening AI team

Near Friday's close, several business events may become coupled:

1. A sales agent introduces a discount to meet a weekly target.
2. Order agents accept the resulting demand surge.
3. Inventory agents reallocate limited stock.
4. Manufacturing agents revise production schedules.

5. Procurement agents order additional materials.
6. Logistics agents reserve transportation capacity.
7. Finance agents evaluate revenue, credit and working-capital effects.

These agents may never be declared as one permanent team. Nevertheless, they become a ‘transient team’ because each agent’s decision changes the operating conditions experienced by the others. The team may also include applications, APIs, queues, databases, SaaS systems and human approvers. Some coupling is explicit through agent communication; other coupling is implicit through shared inventory records, prices, forecasts, production capacity and delivery commitments.

The Friday time boundary can increase the stability risk because several optimization cycles converge simultaneously. Every agent may behave correctly according to its local objective, yet their interaction can produce:

- demand amplification;
- inventory oscillation;
- repeated production rescheduling;
- conflicting resource reservations;
- cascading retries;
- unstable pricing;
- excessive coordination burden;
- commitments that persist into the following week.

The precise division of responsibility

The SoS architecture identifies the persistent domain systems and their possible relationships. The AI-Teamwork framework identifies the temporary collective created by the active business cycle. The GUDIYA Grid observes and stabilizes that collective.

GUDIYA would determine which actants have become coupled, reconstruct the shared intent and feedback paths, monitor delay, gain, cognition velocity, resource contention and accumulated state, and ensure that the team dissolves without leaving contradictory commitments or unfinished objectives.

One small terminology refinement: the AI-Teamwork framework recognizes and characterizes these dynamic teams; the GUDIYA Grid governs and stabilizes them.

The enterprise principle is therefore:

A modern enterprise may possess a relatively persistent System-of-Systems architecture, while its daily work is increasingly performed by transient AI teams that form around orders, customers, deadlines, disturbances and operational cycles. Stability must be managed at both levels.

A Layered Assurance Model

Assurance level	Core question	Representative evidence
Constituent	Is each system or actant fit for its assigned role?	Qualification, KYA record, ACS, authority, capability, lineage, and operating limits
Interface	Can intended exchanges occur correctly and securely?	Interface tests, schemas, protocols, cybersecurity controls, and error behavior

Rendezvous	Can the interaction occur safely under actual timing and state conditions?	Rate compatibility, delay, phase, shared state, authorization, and fallback behavior
Team	Does the collective remain coherent and recoverable as membership and conditions change?	Team envelope, coordination burden, intent consistency, topology, density, and recovery margin
System of Systems	Can independent systems sustain the required mission capability?	Mission threads, architecture, constituent agreements, evolution plans, and SoS test evidence
Ecosystem	Can several teams and Systems of Systems coexist without transferring instability?	Cross-boundary telemetry, capacity margins, cascade analysis, exercises, and Grid history

Implications for Acquisition and Operations

The distinction changes what should be specified, purchased, tested, and operated. Acquisition programs traditionally procure constituent systems and require interoperability. Cognitive missions also need observable team formation, machine-readable authority, state lineage, stability telemetry, Grid attachment, and safe degradation. These are not optional enterprise integrations added after fielding. They are properties of mission participation.

Operational exercises should therefore test more than scripted mission success. They should introduce mismatched cognition rates, lost or delayed communications, stale shared state, sudden member substitution, competing local optimizers, ambiguous command transitions, recursive delegation, and partial Grid visibility. The objective is to reveal whether the collective can remain bounded and recoverable when every constituent is forced away from nominal assumptions.

Certification should also become conditional. An actant may be suitable for one density, mission consequence, communications environment, or team composition but unsafe in another. Instrument ratings for actants and rendezvous can express these limits more usefully than a single universal approval.

A Research Agenda for the Stability Engineering Institute

The comparison produces a focused research program:

- Define the minimum observable evidence that a consequential team has formed.
- Formalize when an actant collective qualifies as a System of Systems and when it remains a multi-agent or integrated system.
- Map the four established SoS governance types to appropriate Grid authority and intervention models.
- Develop measures for team topology, cognition velocity, interaction gain, phase, density, coordination burden, and recovery margin.
- Design mission tests that expose dynamic instability without requiring component failure or adversarial behavior.
- Specify how team and SoS history persists through reconfiguration, migration, substitution, and dissolution.
- Determine how local Grid Operators federate when constituent systems cross organizational, service, national, and commercial boundaries.

- Create evidence standards for stable worthiness at constituent, rendezvous, team, SoS, and ecosystem levels.

The Canonical Relationship

The System of Systems tradition gives GUDIYA a strong predecessor. It already understands that independently owned systems can create a larger capability without surrendering their identity, management, or lifecycle. It already rejects the comfort of assuming that one program controls the whole. It already treats emergence and evolution as engineering realities.

GUDIYA carries that logic into a domain where the members can interpret, learn, delegate, remember, re-plan, and change their coupling at machine speed. It broadens the constituent from system to actant, broadens membership from declared participation to consequential coupling, and broadens assurance from capability delivery to continuous collective stability.

System of Systems is the architectural ancestor. AI teamwork is the runtime cognitive condition. GUDIYA is the stability layer that connects them.

This relationship should be preserved precisely. Calling every AI team a System of Systems would weaken a useful defense concept. Treating AI teamwork as unrelated to System of Systems engineering would discard decades of relevant knowledge. The productive path is inheritance with extension: retain the mature architecture of independent systems, then add the science required to stabilize the collectives they create in motion.

References

1. Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering. Systems Engineering Guide for Systems of Systems, Version 1.0, August 2008. <https://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf>
2. Mark W. Maier. Architecting Principles for Systems of Systems. Systems Engineering, Volume 1, Issue 4, 1998, pages 267 to 284. [https://doi.org/10.1002/\(SICI\)1520-6858\(1998\)1:4%3C267::AID-SYS3%3E3.0.CO;2-D](https://doi.org/10.1002/(SICI)1520-6858(1998)1:4%3C267::AID-SYS3%3E3.0.CO;2-D)
3. Office of the Under Secretary of Defense for Research and Engineering. Mission Engineering Guide, November 2020. https://www.cto.mil/wp-content/uploads/2023/06/MEG-v40_20201130_shm.pdf

Persistent Architecture and Transient Team Membership

A System of Systems describes a relatively persistent architectural arrangement. Its constituent systems—such as airlines, airports, aircraft fleets and air-traffic infrastructure—retain independent purposes, ownership and lifecycles while contributing to a larger capability. The architecture may endure for decades even as individual systems are replaced or modernized.

Team membership describes a different phenomenon: the actants that are consequentially coupled at a particular moment. During one flight, an aircraft may enter a departure-sequencing team, a runway-flow team, several sector-level traffic teams, a temporary conflict-resolution team and an arrival-sequencing team. These teams form and dissolve as position, trajectory, weather, congestion and mission phase change. Their members may belong to different organizations and may never recognize one another as teammates.

An actant can therefore remain a persistent constituent of one System of Systems while repeatedly entering and leaving many transient operational teams. System-of-Systems engineering identifies the enduring participants, interfaces and authorities that make the larger capability possible. GUDIYA must additionally determine which actants are dynamically coupled now, what state and intent they carry, how their relative motion is changing and whether the temporary collective remains stable.

This distinction establishes a new requirement for Stability Engineering: architectural membership cannot substitute for runtime team discovery. The architecture tells the Grid what relationships are possible. The evolving Cognitive Field reveals which relationships have become active, consequential and stability-relevant.

As an example, The System-of-Systems framework describes the relatively persistent aviation architecture—airports, airlines, aircraft identities, ATC sectors, navigation systems and their enduring authority relationships—while the AI-Teamwork framework describes the transient operational collectives continually forming inside it. Seventeen aircraft departing LAX between 6:00 and 6:30 become a team only when their trajectories and decisions are consequentially coupled through runway sequencing, separation requirements, shared routes, weather and sector capacity. As each aircraft proceeds, it leaves one team and joins others across departure, TRACON, en-route and arrival environments, sometimes belonging briefly to overlapping teams during handoffs. Its identity remains persistent, but its team membership, neighboring actants, authority context and stability envelope change with time. SoS engineering identifies the enduring architecture and possible relationships; GUDIYA must continuously discover which relationships are active now and stabilize the living topology they create.

The Value of AI Teamwork for CCA and MMA Hybrid Drone Architectures

System-of-Systems engineering describes the relatively persistent architecture of a hybrid combat fleet: crewed aircraft, CCA, MMA, sensors, weapons, satellites, command networks, bases and logistics systems. The AI-Teamwork framework addresses the transient operational collectives that continuously form inside that architecture. A CCA may leave an escort formation, join an interception team, act as a sensor node and later enter a recovery group. Its identity persists, but its role, authority, neighboring actants, active couplings and stability envelope change throughout the mission.

These teams are created by consequential coupling rather than common ownership or permanent assignment. Aircraft become teammates when their trajectories, sensing, targeting, communications, weapons or decisions alter one another's operating conditions. Some teams will be deliberately commanded; others may emerge through shared targets, airspace, sensor observations, resource constraints or environmental effects. Attrition, degraded communications, deception and asymmetric cognition velocities can cause these collectives to re-form at machine speed.

The AI-Teamwork framework therefore gives DoW a runtime capability that complements System-of-Systems and mission engineering. The persistent architecture reveals which relationships are possible; GUDIYA identifies which actants are functioning as a consequential team now and determines whether that evolving collective remains bounded and recoverable. System-of-Systems engineering identifies the force that exists. AI-Teamwork Stability Engineering identifies and stabilizes the force forming in motion.

