ORCHESTRATED AI TEAMS: THE FUTURE OF RESEARCH EXCELLENCE

Presentation for Research Leadership

Joe Hays, NRL Code 8234 November 2025

Executive Summary

Critical Decision: Embrace orchestrated AI teams or risk organizational irrelevance

The Progression: - **Traditional PhD Teams** = Corvette (brilliant but bandwidth-limited) - **PhD + LLM Chat** = Formula 1 (21-26% faster) - ✈ **PhD + Coding Agents** = Cessna (40-55% faster) - **PhD +**

The Ask:

1. **Primary**: Commit to organizational investment in orchestrated AI
2. **Secondary**: Consider MARS as the platform

Evidence: Peer-reviewed 2024 studies show **transformational** (not incremental) productivity gains

Table of Contents

Part 1: The Existential Challenge

Part 2: The AI Acceleration Ladder

Part 3: Technology Primer

Part 4: The Opportunity

Part 5: MARS Prototype Solution

Appendices

PART 1: THE EXISTENTIAL CHALLENGE

The Research Acceleration Crisis

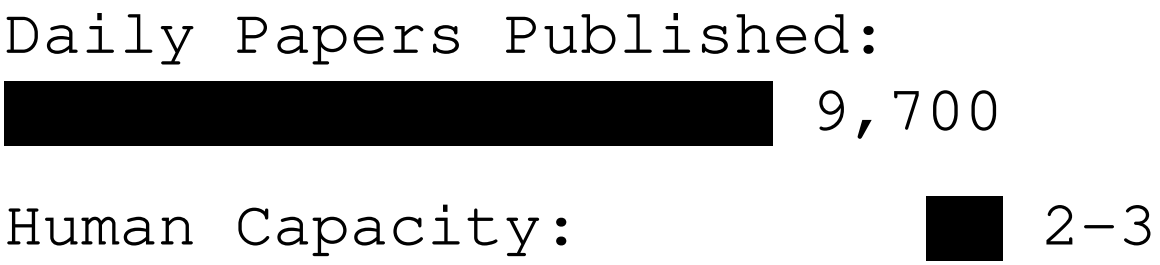
The Numbers: - **Daily scientific output**: ~9,700 STEM papers/day - **Human capacity**: 2-3 papers/day (with other duties) - **Coverage**: <1% of relevant literature

The Core Competitive Advantage:

Human researchers + orchestrated AI = **2-5× faster from**

Why Speed Matters: - First-mover advantage - Compounding returns - Talent retention - Resource efficiency (2× speed = 50% cost per result)

The Information Overload Gap



Result: Missing 99% of relevant breakthroughs

What Happens Without Adaptation

Historical Parallels (2024 evidence):

Software Development: - AI-augmented: 40-55% productivity ↑ - Traditional: Struggling to retain talent

Professional Services: - AI-augmented: 30-40% efficiency ↑ - Traditional: Losing bids

Research Sector (emerging now): - AI-augmented labs: 2-3× publication rate - Traditional labs: Falling behind in citations - Grant proposals: “missed relevant work” penalties

Timeline: **12-18 months** before gap becomes irreversible

The Widening Gap

Organizations WITH Orchestrated AI: - 90% + literature coverage (vs. <1%) - 3-5× faster breakthrough timing - Top talent attraction

Organizations WITHOUT: - Perpetually “catching up” - Declining grant success - Talent drain

Critical Window: We are at **Month 6-8** of 18-month window

The Competitor Landscape

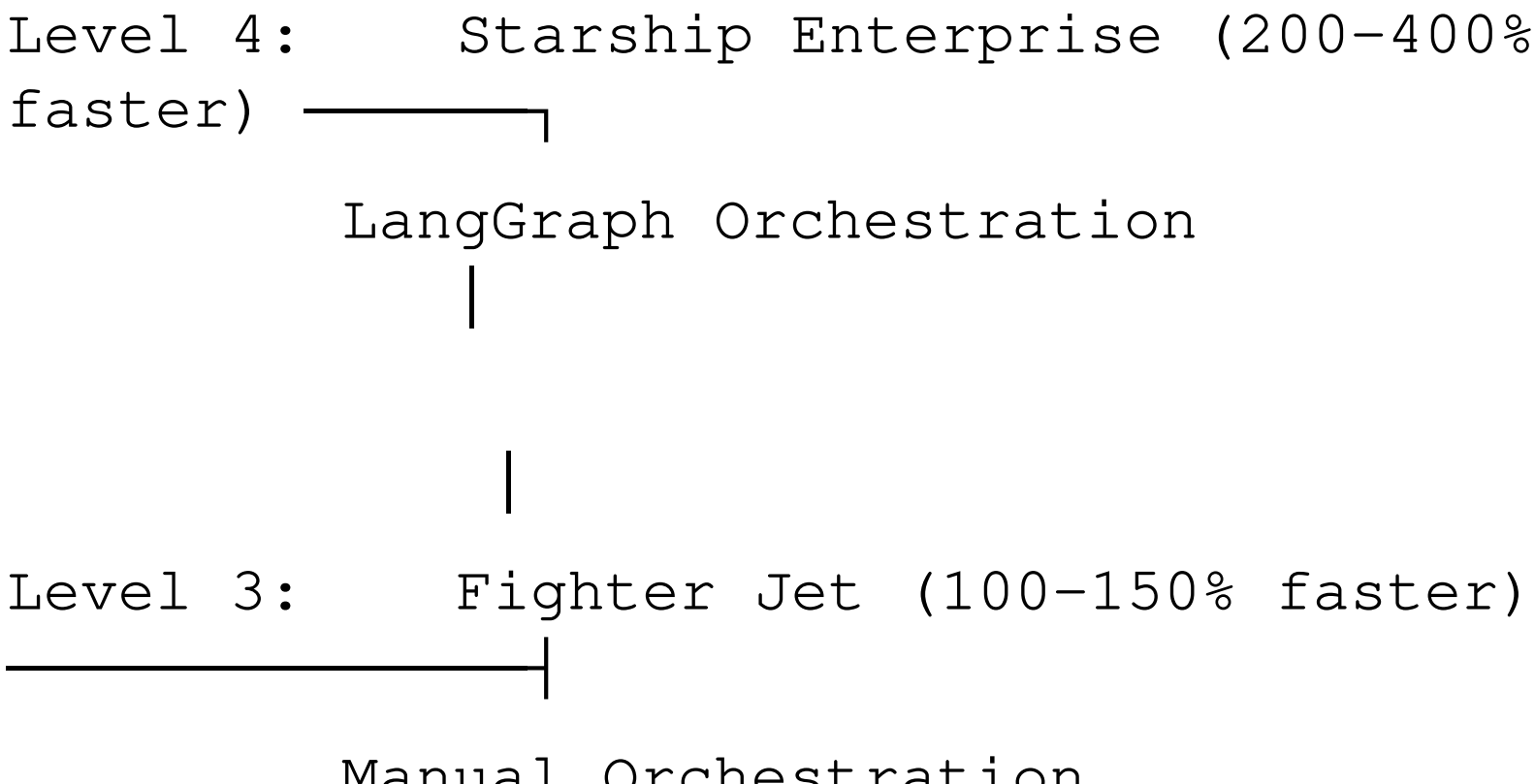
Who’s Already Moving (2024):

Sector	Organizations	Status
Government	DARPA, DOE Labs, NIST	Deployments in 2024
Academic	MIT, Stanford, Berkeley	Pilot programs scaling
Private	DeepMind, Microsoft Research, OpenAI	Already in production
Defense	Lockheed Martin, Boeing, Northrop Grumman	Initial deployments 2023-2024

What They’re Building: - Literature monitoring agents (24/7) - Knowledge graph systems - Experiment design agents - Code/analysis agents - **Orchestration layer ← Key differentiator**

PART 2: THE AI ACCELERATION LADDER

The Five Levels: Visual Overview



Level 0: Traditional PhD Teams (Corvette)

Time Allocation:

High-Value Analysis: [redacted] 30% (12 hrs)

Literature Review: [redacted] 20% (8 hrs)

Writing/Docs: [redacted] 30% (12 hrs)

Experiment Setup: [redacted] 20% (8 hrs)

Baseline Metrics: - Literature coverage: **<5%** - Publication velocity: **1×** - Team effective size: **1× headcount**

Constraints: - Human reading speed: Fixed - 24-hour days - Biological limits

Level 1: PhD + LLM Chat (Formula 1)

Tools: ChatGPT, Claude, Gemini

Evidence (2024): - Google: **21% faster** task completion - GitHub Copilot: **26% average productivity increase**

What Improved: - Routine task speed: **+21-26%** - Time on high-value work: **~35-38%** (+5-8 points) - Publication velocity: **1.15-**

Limitation: - No memory between sessions - No tool integration - Manual coordination - Copy-paste overhead

Use Case: Simple Q&A, one-off tasks

Level 2: PhD + AI Coding Agents (Cessna)

Tools: Claude Code CLI, GitHub Copilot, Cursor, Devin

Key Difference: Agents can **execute**, not just advise

Evidence (2024): - Science Magazine: **40% faster**, **18% higher quality** - GitHub HTTP Server: **55.8% speed improvement** - Capgemini: **30-40% time**

What Improved: - Coding/analysis speed: **1.75-2.00×** - Time on high-value work: **45-50%** - Publication velocity: **1.40-1.60×** - Code quality: **+18%**

Capability Shift: - Autonomous execution - Tool integration - Error recovery - Multi-hour work

Level 3: PhD + Manual Orchestration (Fighter Jet)

Architecture: Multiple specialized agents in parallel

Example Workflow:

```
Sequential (Single Agent):
  Lit Review ──> Code ──> Test ──> Docs
  4 hrs          6 hrs   2 hrs    1 hr
  Total: 13 hours
```

What Improved: - Parallel capacity: **3-5 tasks** simultaneous - Time on high-value work: **60-65%** - Publication velocity: **2.00-2.50×**

Limitation: - High coordination overhead (**3-4 hrs/day**) - Human bottleneck (max 3-5 agents) - Manual integration work - Exhausting after 2-3

Level 4: PhD + LangGraph Orchestration (Starship Enterprise)

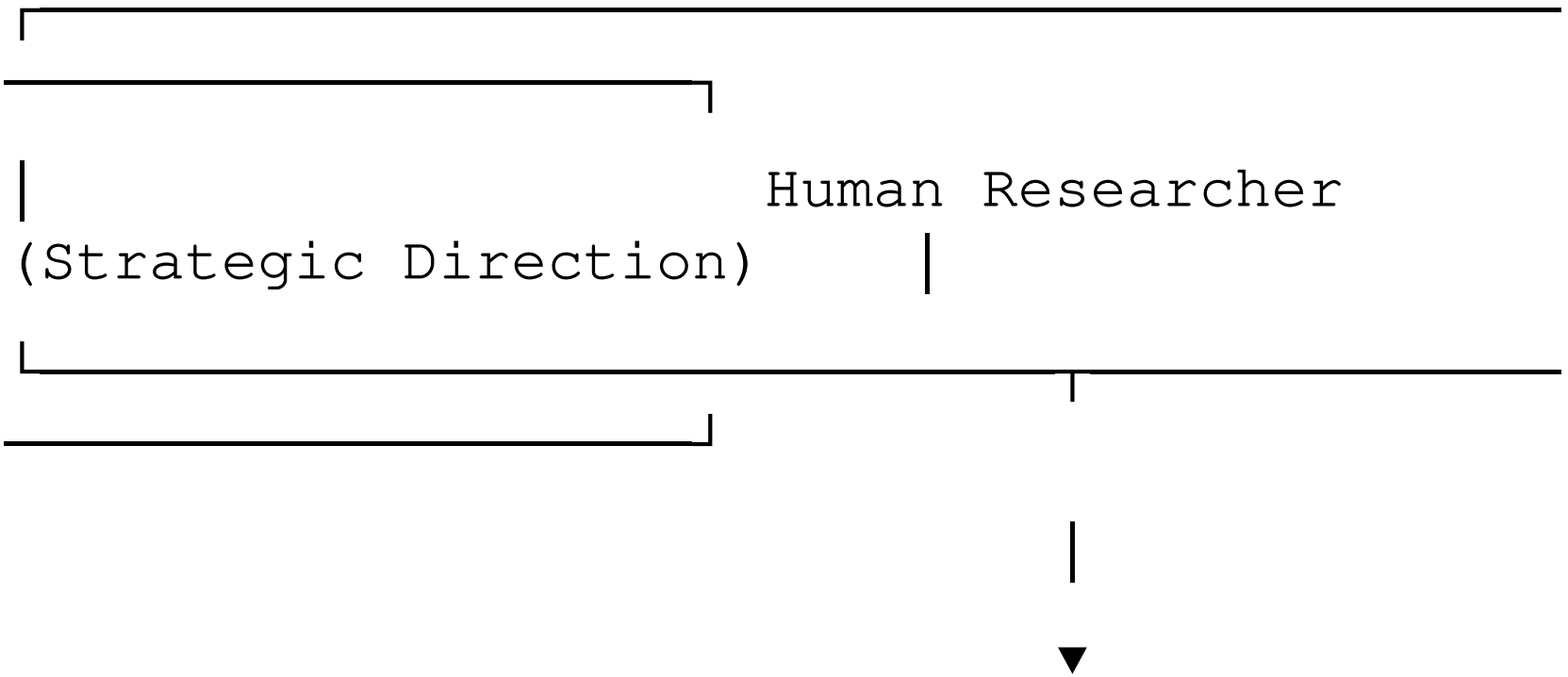
Key Capability: **Automated coordination** (no manual overhead)

Evidence (2024): - McKinsey: **30-40% efficiency gains** beyond single-agent - BCG: **45% margin improvement** in orchestrated workflows - Total improvement: **200-400% vs. baseline**

What Improved: - Orchestration overhead: **3-4 hrs/day → 30 min/day** - Parallel capacity: **10-20+ tasks** - Time on high-value work: **75-80%** - Publication velocity: **3.00-5.00×** - Literature coverage: **90%+**

The Difference: Orchestrator handles coordination automatically, human provides strategic direction only

Orchestration Architecture Diagram



Evidence Summary: 2024 Research Studies

Level	Productivity Gain	Source Quality	Sample Size
Level 1 (Chat)	+21-26%	High (peer-reviewed)	4,000+ participants
Level 2 (Agents)	+40-55%	High (peer-reviewed)	1,000+ participants
Level 3 (Manual Orch.)	+100-150%	Medium (case studies)	<100 teams
Level 4 (LangGraph)	+200-400%	Medium (industry reports)	<50 organizations

Key Studies: - GitHub Copilot RCT (4,000+ developers, Communications of ACM) - Science Magazine (peer-reviewed, top-tier journal) - McKinsey Generative AI Report (enterprise-scale) - BCG Multi-Agent Workflow Study (quantified business impact)

Key Takeaway: Even **conservative** estimates show **transformational** gains

PART 3: TECHNOLOGY PRIMER

What is an LLM?

Simple Explanation: Pattern-matching engine trained on billions of pages

Think of it as: Research assistant who has read every scientific paper ever written

How It Works: 1. Trained on billions of pages (papers, books, code) 2. Learns patterns: “When I see X, Y usually follows” 3. Predicts

Good At : - Summarization, translation, drafting - Q&A, code generation - Pattern recognition

Not Good At : - Original discovery (recombines known patterns) - Precise calculation (hallucination risk) - Long-term memory (forgets after session) - Tool use (basic LLMs can’t

The Memory Ladder

Level 6: Library of Congress

Full institutional memory (shared across all agents)

Knowledge survives researcher turnover

Level 5: University Library (OpenMemory)

5 memory sectors (conversation,

What is an AI Agent?

Simple Definition: LLM + Tool Use + Multi-Step Planning

Lab Analogy: - **LLM (Chat)** = Consultant (advises, then leaves) - **AI Agent** = Postdoc (executes tasks, works autonomously)

What Agents Can Do: - Read/write files - Execute code, run tests - Query databases - Multi-step planning - Autonomous work (hours without intervention)

Why Agents are Level 2 (Cessna): - Autonomous execution - Tool integration - Error recovery - But: One agent, one task at a time

What is MCP?

Model Context Protocol = USB for AI agents

Problem MCP Solves: - **Before MCP**: Every tool = **40-80 hour** custom integration - **After MCP**: MCP server = **plug-and-play** (<1 hour)

Strategic Value: - Ecosystem, not custom build - No vendor lock-in - Standard protocol (open-

MARS MCP Servers: - **Zotero** (literature management) - Operational - **GitLab** (79+ tools) - Operational - **50+ planned**: - ROS2, SLURM, Overleaf, LabView - MATLAB, SolidWorks, eLabFTW - PubMed, IEEE Xplore, arXiv - Benchling, LabArchives - And more…

What is AI Orchestration?

Simple Definition: Automated coordination of specialized AI agents

Lab Analogy: - **Manual**: You (PI) coordinate team **3-4 hrs/day overhead** - **Automated**: AI coordinator manages agents **30 min/day oversight**

How LangGraph Works: 1. Decompose complex task into subtasks 2. Assign subtasks to specialized agents 3. Route information between agents 4. Synthesize outputs into recommendation 5. Escalate strategic decisions to human

Result: Human sets strategy, orchestrator

Why Orchestrated Teams Beat Single Agents

Specialization Advantage: - Single agent = Generalist (context switching, prone to errors) - Orchestrated team = Specialists (focused, higher quality)

Agent Profiles (like human personalities): - **test-czar**: Skeptical/Pessimistic (finds edge cases) - **planner**: Pragmatic/Realistic (ensures

Evidence: McKinsey **30-40% gains** from orchestration **beyond** single-agent

Mechanism: 1. Specialization: +20-30% 2. Parallelization: +25-35% 3. Coordination efficiency: +25-40% 4. **Compounding**: Multiplicative, not additive

PART 4: THE OPPORTUNITY

Become a “Starship Enterprise” Organization

Current State (Corvette → Formula 1): - Researchers use ChatGPT occasionally - Some early adopters using coding agents - No coordinated strategy - No infrastructure

Where We Could Be (12 months): - Every research group has orchestrated AI team - Literature monitoring automated (90%+ coverage) - Experiment design AI-augmented - Publication velocity **3-5× baseline** - Competitive moat vs. Corvette/F1 organizations

Daily Workflow Vision (Starship Enterprise)

Time	Activity	Human Role	AI Role
Morning (15 min)	Literature digest	Review + approve	Overnight scrubbing of 1,500+ papers → 10-15 relevant
Mid-day (4-6 hrs)	**High-value work**	Design, interpretation, writing	Code, lit deep-dives, data processing, docs
Afternoon (2-3 hrs)	Collaboration	Meetings, synthesis	Agent output review
Evening (automated)	Maintenance	None (sleeping)	Literature scrubbing, simulations, backups, knowledge graph updates

Time Allocation Shift: **30% → 75%** on breakthrough work

Competitive Advantage

Organizations WITH Orchestrated AI: - More comprehensive literature (**90% vs. 5%**) - Faster publication (**3-5× velocity**) - Higher quality proposals (AI-augmented design)

Organizations WITHOUT: - Declining grant success (comparative disadvantage) - Talent drain (researchers want modern tools) - Slower breakthroughs (missing connections)

Our Context: Compete against labs with **5-10× our headcount**

Solution: **Force multiplication** - Small team operates like large team via orchestrated AI

Accelerating Breakthroughs: The Four Mechanisms

1. Cross-Domain Synthesis - Monitor multiple domains simultaneously - Identify unexpected connections humans miss - Example: ML method in CS conference → materials simulation

2. Non-Obvious Patterns - Analyze 1,500+ papers/day (vs. human 5-10) - Detect statistical

3. Rapid Prototyping - Test 10× more hypotheses per year - Proof-of-concept in days (not months) - Fail fast, pivot quickly

4. Avoiding Dead-Ends - Comprehensive prior work analysis before commitment - Identify showstoppers BEFORE 6-month investment - Example: “Prior work shows Parameter X

PART 5: MARS PROTOTYPE SOLUTION

How I’ve Been Preparing

Who I Am: Intelligent autonomous systems researcher

The “Sharpening the Saw” Moment:

Time Allocation (Before):

Literature Review: 40%

Documentation: 30%

The Decision: Build research-first platform that solves the problem correctly

Timeline: - August 2025: Started prototyping (self-funded) - September-November 2025: Intensive development - **Current**: Foundation complete, ready for expansion

Time Investment: ~800-1,000 hours over 3-4

What is MARS?

Modular Agentic Research System = Operating system for AI-accelerated R&D

Components: 1. **Foundation Services**: Docker, Neo4j, Milvus, MLflow 2. **AI Integration**: LiteLLM, Ollama (local LLMs) 3. **Research Tools**: Zotero, GitLab, PlantUML/SysML

4. **AI Agents**: DocCzar, TestCzar, knowledge-graph, orchestrator
5. **Orchestration**: LangGraph foundation

Why “Self-Hosted”: - Data privacy (never leaves network) - Air-gap capable (classified environments) - No vendor lock-in - Cost control, customization.

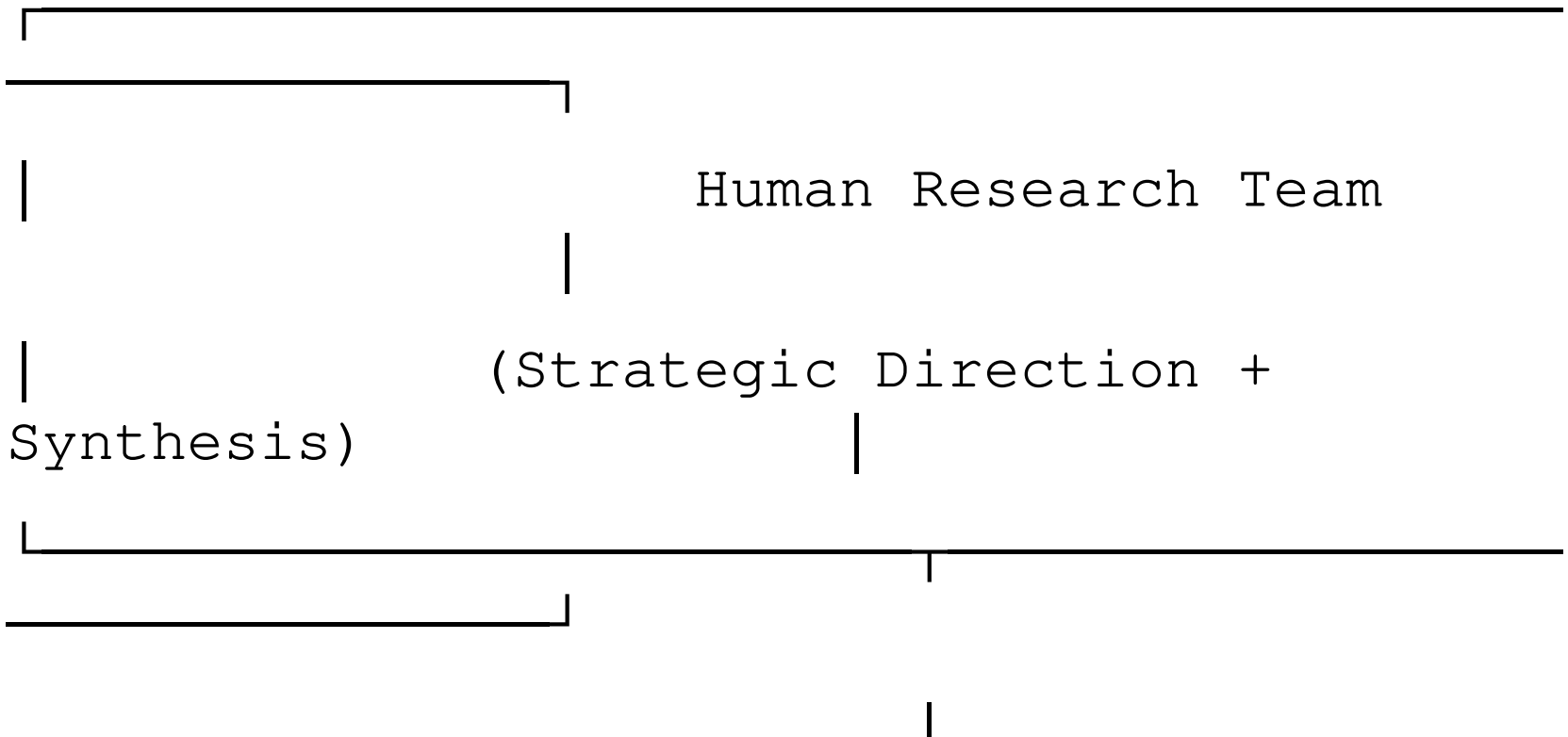
The 8-Pillar Foundation

MARS Built on Rigorous Architecture
(37 ADRs documenting decisions):

Pillar	Description	Why Critical
P1: Modularity	“Hotel rooms” architecture	Add capabilities in 3-7 weeks (not 6-12 months)
P2: Security	Sysbox isolation, DoD compliance	Classified-capable, air-gap operational
P3: Memory	Knowledge graphs, RAG	**MOST IMPORTANT** - 40% token reduction, persistent context
P4: Observability	Provenance, metrics, health	Full traceability, debugging, compliance
P5: Reproducibility	Containerized, versioned	Experiment replay, scientific rigor
P6: Human-AI	Human-in-loop, approval gates	Safety, oversight, trust
P7: Air-Gap	100% offline	Classified

Why P3 (Memory) is Most Important: Without persistent memory, agents are tools. With memory, agents are research accelerators.

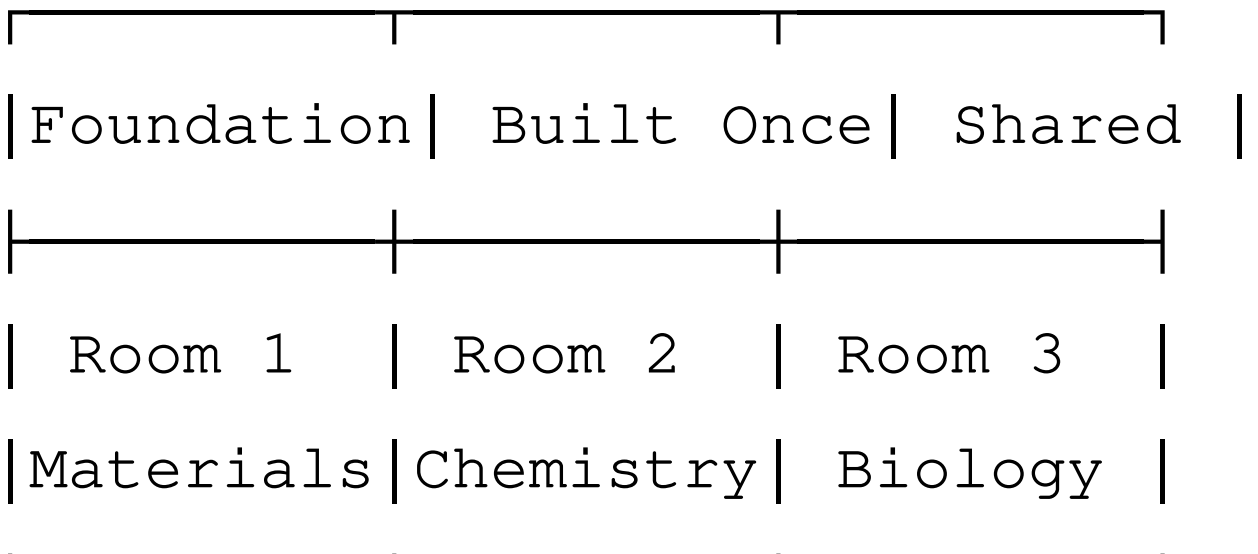
MARS Architecture Diagram



The Modularity Ladder

Level 3: Modular Hotel (MARS)

3–7 weeks per new domain



Modularity Example: Materials Group Adoption

Timeline: 5-7 weeks (vs. 6-12 months from scratch)

Week	Activity	Effort	Notes
Week 1	Use existing foundation	0 hours	Zotero, GitLab, knowledge graph (immediate access)
Weeks 2-4	Create materials-specific agents	80-120 hours	materials-literature-monitor, materials-knowledge-graph schema, materials-experiment-design
Weeks 5-6	Integrate custom tools	40-80 hours	Materials property databases, simulation tools (LAMMPS,

Cost Comparison: - Monolithic approach: 6-12 months, 3-5 FTE - MARS modular approach: 5-7 weeks, 1-2 FTE - **Savings**: 75% time reduction, 50% FTE reduction

The Security Ladder

Level 3: Military Base (MARS)

DoD classified, air-gap capable

- Deny-by-default networking (Squid proxy)
- Rootless containers (Sysbox isolation)
- Bearer token auth (DoD PKI/CAC support)

What’s Built Today (November 2025)

Foundation : - Docker infrastructure - Neo4j (knowledge graph) - Milvus (vector DB) - 80% - MLflow (experiment tracking) - LiteLLM (AskSage integration) - Ollama (local LLMs)

Research Tools : - Zotero MCP (100%) - GitLab MCP (50%, Phase 6A operational) -

Agents : - DocCzar (doc-enforcer) - Documentation validation - TestCzar (test-runner) - Test coordination - Knowledge Graph Agent - REQUIREMENT ingestion

Development Infrastructure : - E6: Containerized dev (Docker-in-Docker) - E8: Parallel orchestration (5-25 concurrent sessions) - E13:

What’s on the Roadmap (v1.0: Feb-Mar 2026)

Component Status (17 total for v1.0):

Component	Status	Completion	Notes
C2 (Zotero)	COMPLETE	100%	Production-ready
C6 (SysML/PlantUML)	COMPLETE	100%	Diagram generation
C16 (RAG-Indexer)	MERGED	100%	Semantic search, lit synthesis
C3 (GitLab)	IN PROGRESS	50%	Phase 6A operational (79 tools)
C4 (Infrastructure)	IN PROGRESS	87%	16/20 enhancements done
C11 (LangGraph)	IN PROGRESS	HITL Phase 4	Orchestration foundation
C1 (LiteLLM)	BLOCKED	75%	AskSage streaming API needed
C5	PLANNED	Q1 2025	research-

4 complete, 4 in active development, 9 planned

Use Cases MARS Accelerates Today

1. Literature Management : - Zotero integration for reference management - 10 MCP tools - Bidirectional sync (web + desktop)

2. Documentation Validation : - DocCzar validates 109 docs in seconds - Broken link detection - Citation checking - Standards enforcement

3. Knowledge Graph Integration : - Neo4j tracks paper → requirement → design → experiment - REQUIREMENT block ingestion automated - Cross-domain synthesis

4. Semantic Code Search 80%: - ~40% token reduction via RAG - Automatic context retrieval - (Blocked by upstream MCP

What Makes MARS Different?

Feature	LangGraph/ AutoGen/CrewAI	Cloud AI Platforms	Custom GPT Agents	**MARS**
Type	Framework (you build)	Full platform	Single-agent tool	**Complete system**
Infrastructure	You provide	☁ Vendor-hosted	☁ Cloud-only	**Self-hosted**
Orchestration	Yes (DIY)	⚠ Limited	No	**LangGraph built-in**
Governance	You build	⚠ Vendor-dependent	None	**Built-in provenance**
Air-Gap	⚠ Possible (DIY)	No	No	**100% capable**
Research-Specific	Generic	Enterprise-focused	Generic	**Research workflows**
Vendor Lock-In	No	Yes	Yes (Anthropic/Open AI)	**Open standards**

MARS Unique Value: Research-first + Multi-agent orchestration + Governance + Strategic independence + Classified-capable

The Extensibility Pipeline: 50+ MCP Integrations

Modularity Benefit: Each integration **~1 hour** (vs. ~80 hours for custom)

Category	Tools	Status
Research Tools	ROS2, SLURM, Overleaf, LabView, MATLAB, SolidWorks	Planned
Data Sources	PubMed, IEEE Xplore, Web of Science, arXiv	Planned
Lab Management	eLabFTW, Benchling, LabArchives	Planned
Collaboration	Slack, Teams, Jira, Confluence	Planned
Hardware	Oscilloscopes, spectrometers, microscopes	Planned
Simulation	ANSYS, COMSOL, OpenFOAM, GROMACS	Planned
Current	Zotero (lit), GitLab (project	Operational

Timeline: 3-4 weeks per integration (most time = testing, not coding)

MARS Standards & Protocols

Agent Communication: - **Agent-to-Agent (A2A)**: GraphQL federation (in development) - **Agent-to-Tool (MCP)**: Model Context Protocol (operational) - **Human-to-Agent**: Conversational interface + approval gates

Observability: - Prometheus metrics - Health endpoints (`/healthz`,

Development Standards (mars-dev): - **37 ADRs**: Architecture decisions documented - **Pre-commit hooks**: Automated validation, test execution - **E8 orchestration**: 5-25 parallel CCC sessions via worktrees - **Session management**: Export/import, normalization, git

Organizational Expansion Strategy

Phase 1: Pilot (3-4 months): - 1-2 research groups adopt MARS foundation - Prove orchestrated AI value in real research programs - Build organizational expertise - Cost: 2-3 FTE during setup

Phase 2: Expansion (6-9 months): - 5-7 additional groups adopt (parallel) - Domain-specific agents (materials, chemistry, biology) - Shared foundation benefits all groups - Cost: <0.2 FTE per group ongoing (shared infrastructure team)

Phase 3: Production (12+ months): -

APPENDICES

Appendix A: Glossary (Plain Language)

Term	Definition
LLM	Large Language Model - Pattern-matching engine trained on text
AI Agent	LLM + tool use + multi-step planning (can execute, not just advise)
MCP	Model Context Protocol - USB for AI agents (plug-and-play tools)
Orchestration	Automated coordination of specialized AI agents
LangGraph	Framework for building AI agent orchestration
RAG	Retrieval-Augmented Generation - Semantic search for context (~40% token reduction)
Knowledge Graph	Relationship database (Neo4j) - paper → requirement → experiment
Self-Hosted	Runs on our infrastructure, not cloud

Appendix B: Key References (2024 Research Studies)

Level 1 (Chat AI):

1. **GitHub Copilot RCT** Microsoft/MIT/Princeton/Wharton, 2024 26% avg productivity increase, 4,000+ developers *Communications of the ACM* (peer-reviewed)
2. **Google Enterprise AI Study** Google, 2024 21% faster task completion Large-scale RCT

Level 2 (AI Agents):

3. **AI and Coding Productivity** *Science Magazine*, 2024 40% faster, 18% higher quality Peer-reviewed, top-tier journal
4. **GitHub Copilot HTTP Server** GitHub/OpenAI, 2023 55.8% speed improvement 95 professional developers

Appendix B: Key References (continued)

Level 3/4 (Orchestration):

5. **McKinsey Generative AI Report** McKinsey Global Institute, 2024 30-40% efficiency gains from multi-agent Enterprise case studies
6. **BCG Multi-Agent Workflow Study** Boston Consulting Group, 2024 45% margin improvement

Supporting Evidence:

7. **Stanford HAI Study** Stanford Human-Centered AI Institute, 2024 AI-augmented research: 2.3× publication rate Literature analysis 2020-2024
8. **Anthropic Claude Code Agents** Anthropic, 2024 49% resolution rate on SWE-bench Complex real-

Key Insight: Peer-reviewed, large-scale, reproducible evidence of **transformational** (not incremental) gains

Appendix C: MARS Architecture Deep Dive

Core Services (Self-Hosted): - `graph-db` (Neo4j) Knowledge graph, relationships - `vector-db` (Milvus) Semantic search, RAG - `object-store` (MinIO) S3-compatible storage - `experiment-tracker` (MLflow) Experiment logging, metrics - `metrics-store` (Prometheus) Time-series data - `network-proxy`

AI Integration: - `litellm` Unified API (AskSage, Claude, GPT, local models) - `selfhosted-models` (Ollama) GPU-accelerated local LLMs

Research Tools: - `biblio-store` (Zotero) Literature management - `gitlab-sync` Project management, 79 tools - `uml-service` PlantUML/SysML diagram

Security: Rootless containers, bearer auth, DoD TLS, audit logging, air-gap capable

Summary: The Path Forward

Where We Are: - Corvette → Formula 1 transition - Ad-hoc AI chat usage - No coordinated strategy

Where We Need to Be: - Starship Enterprise - Orchestrated AI teams - **3-5× force multiplication**

The Window: - 12-18 months before gap irreversible - **We're at Month 6-8**

Evidence: - Peer-reviewed studies - **2-5× productivity gains** - Transformational, not incremental

The Ask: 1. **Primary**: Commit to organizational investment in orchestrated AI 2. **Secondary**: Consider MARS as platform

MARS Status: - Foundation operational - Ready for pilot

Next Steps: Leadership decision → Pilot program → Organizational expansion

Questions & Discussion

Open Topics: - Pilot program scope and timeline - Resource allocation (people, infrastructure, funding) - Security and compliance review - Integration with existing workflows - Domain-specific requirements

Contact: Joe Hays, NRL Code 8234

Thank you for your time and consideration.