

THE DISTRIBUTED STUDIO · OPERATIONAL PLAYBOOK

STEALTH RESEARCH · JUNE 2026 · VOL. I

# The Distributed Studio **Playbook.**

*A 30-page operational manual extending Working Paper 01. The four pillars in deep dive. Implementation roadmap. Failure modes. Cost structure. First 100 days. Hiring profile. Governance. Year-by-year milestones. Templates.*

For studio operators · founders · LP allocators · academic  
researchers · CC BY 4.0

---

# Contents

00	Preface — Who this is for	03
01	Pillar Deep-Dive · The Workspace	04–07
02	Pillar Deep-Dive · The Codified OS	08–11
03	Pillar Deep-Dive · The Credential Layer	12–15
04	Pillar Deep-Dive · The Writing Culture	16–19
05	Implementation Roadmap (18–36 months)	20–22
06	Common Failure Modes	23–24
07	Cost Structure	25
08	The First 100 Days	26
09	Hiring Profile	27
10	Governance & Legal Structure	28
11	Year 1 / Year 2 / Year 3 Milestones	29
12	Appendix — Templates & Checklists	30

---

---

# Preface — who this is for.

This playbook is for operators planning to build a distributed venture studio. It is also useful for founders considering joining one, LPs evaluating a distributed-studio fund commitment, and academics studying the model.

It extends Working Paper 01 (*Distributed Studio Operating Architecture*) with operational depth. Working Paper 01 introduced the four pillars; this Playbook documents how to actually build each one.

Three things this playbook is not:

- **Not a startup guide.** It assumes the reader already understands venture capital, studio operations, and basic company-building. If this is your first studio, read Working Paper 01 first.
- **Not a finished framework.** The distributed studio model has only existed at scale for five years. This playbook captures the state of the practice as of June 2026. Expect v2 in 2028 to look different.
- **Not free of bias.** This is written from STEALTH's perspective and reflects our specific implementation choices. Other distributed studios will build differently and may make different trade-offs. We've documented our reasoning where it differs from likely alternatives.

**How to read this playbook:** sequentially if you're building from scratch; sections 01–04 plus section 05 if you're partway through and want to evaluate where you are; the appendix templates if you want to copy something quickly.

SECTION 01

# The Workspace.

*A task-resolution system that produces institutional-grade output from distributed labor. The unit of execution moves from the meeting to the task.*

01.1	What it is
01.2	Build vs buy
01.3	Task template structure
01.4	Supervisor evaluation rubric
01.5	Common workspace mistakes

---

## — 01.1 What the Workspace is

The Workspace is the operational substrate of a distributed studio. It is not a project management tool, not a chat platform, not a knowledge base. It is a *task resolution system*: a structured environment where:

- Institutional knowledge is encoded as task templates
- Distributed operators pick up tasks asynchronously
- Evaluation rubrics enforce quality without supervisor-by-attention
- Finished tasks compound into reusable artifacts

The defining test: can a new operator who has never worked at the studio before complete a non-trivial task to institutional-grade quality, supervised only by the rubric, within their first week?

## — 01.2 Build vs buy

Most off-the-shelf project management tools (Asana, Linear, Notion, ClickUp) can serve as the foundation of a workspace if augmented appropriately. The core augmentations are:

- **Task templates with embedded rubrics** — the off-the-shelf tools support templates but don't enforce rubric-driven evaluation. Build this layer.
- **Contributor ranking + Score integration** — off-the-shelf tools don't expose contributor performance data. Build a parallel database.
- **Async-first review flow** — most tools assume synchronous review. Configure for async with explicit SLAs.

STEALTH built our workspace on Notion as the primary canvas, with a custom service handling contributor ranking and task routing. This worked at our scale; at 10× our scale it will probably need to be rebuilt as a dedicated product. Other distributed studios have built on Linear, Airtable, or custom stacks.

## 01.3 Task template structure

Every task in the workspace follows a 7-field template. The template enforces the discipline that makes distributed execution work.

FIELD	PURPOSE
<b>Goal</b>	The one-sentence "this task succeeds when X is true." Forces clarity at intake.
<b>Context</b>	Background the contributor needs. Links to prior work, customer interviews, related decisions.
<b>Acceptance criteria</b>	3–7 specific tests the output must pass. Replaces "I'll know it when I see it."
<b>Estimated effort</b>	Hours, set by the assigner. Calibrates expectations and tracks throughput.
<b>Reference materials</b>	Previous similar tasks, playbook entries, templates. Reduces repeated discovery.
<b>Reviewer</b>	Named operator who will evaluate the finished work. Not the assigner by default.
<b>Deadline</b>	Hard. Slipping a deadline requires a renegotiation, not a silent extension.

Every task template is open-source-equivalent inside the studio — contributors can read all task templates, including those they haven't been assigned to. This is intentional. The workspace is a learning environment as well as a production system.

### WORKSPACE TASK TEMPLATE — QUICK CHECK

- Goal is one sentence and outcome-focused, not activity-focused
- Acceptance criteria are testable, not subjective
- Reviewer is named, not "team"
- Deadline is specific, not "ASAP"
- Reference materials include at least one prior similar task

## — 01.4 Supervisor evaluation rubric

When a task is delivered, the named reviewer evaluates it against a 5-dimension rubric. This is the same rubric across every task type at the studio — only the acceptance criteria vary by task. Standardizing the rubric is critical because it lets the Score aggregate meaningfully across heterogeneous work.

DIMENSION	SCORING (1–5)
<b>Completion</b>	Did the work meet all acceptance criteria? 5 = all met; 1 = none met.
<b>Quality</b>	How well-crafted is the output, beyond meeting minimum criteria? 5 = exceptional; 1 = poor.
<b>Independence</b>	How much supervision was required? 5 = self-directed; 1 = needed step-by-step guidance.
<b>Communication</b>	Quality of progress updates, questions asked, handoff. 5 = exemplary; 1 = silent or noisy.
<b>Compounding</b>	Did this work produce reusable artifacts (templates, playbooks, lessons)? 5 = significant compounding; 1 = single-use only.

The 5-dimension score aggregates into the contributor's Forkaia® Score over time. The methodology for that aggregation is documented in the Forkaia × STEALTH Joint Paper (separate document).

## — 01.5 Common workspace mistakes

- **Underspecified acceptance criteria.** The biggest failure mode. Without clear acceptance criteria, every review becomes a subjective renegotiation.
- **Treating the workspace as a chat tool.** Conversations belong in chat. Tasks belong in the workspace. Mixing the two destroys both.
- **Skipping the rubric scoring step.** Reviewers under time pressure skip scoring. The fix is making scoring required before the task closes, not a separate step.
- **Allowing deadline slippage without renegotiation.** Silent slippage destroys throughput discipline. Every slip should require an explicit conversation.
- **Hiding underperformance.** A contributor with consistently low scores should be told so within their first 5 tasks, not buried.

---

## — Section 01 summary

The Workspace is the operating substrate of a distributed studio. Built right, it produces institutional-grade output from distributed labor. Built wrong, it produces a glorified Slack channel that adds friction without leverage.

*"Build the workspace first if you must, but build the rubric earlier. A workspace without an evaluation discipline is worse than no workspace at all."*

### Key takeaways for operators building this pillar:

- Pick an off-the-shelf foundation (Notion, Linear, Airtable). Resist the urge to build custom for the first 18 months.
- Implement the 7-field task template before opening the workspace to distributed contributors.
- Implement the 5-dimension rubric scoring before the first 50 tasks complete. After that, retroactive scoring becomes politically impossible.
- Designate one operating-team member as workspace owner. They own template hygiene and rubric enforcement.

**What's next:** Section 02 covers the Codified OS — the playbooks and decision rubrics that the workspace operationalizes.

# The Codified Operating System.

*Playbooks, decision rubrics, evaluation frameworks. The institutional memory written down so it survives the founders' attention.*

02.1	What gets codified
02.2	The playbook taxonomy
02.3	Versioning + maintenance
02.4	The Decision Memo template

## — 02.1 What gets codified

Every recurring decision, every operational pattern that occurs more than 3 times in 6 months, every quality threshold the studio enforces — all get written down. The bar is not "complete documentation"; the bar is "enough documentation that a new operator can execute without consulting a founder."

## — 02.2 The playbook taxonomy

STEALTH organizes playbooks across 8 categories. Other studios will have different taxonomies; what matters is that the taxonomy is explicit and stable.

- **Formation playbooks** — how to spin up a new company from idea to LLC formation
- **Hiring playbooks** — sourcing, interviewing, offering, onboarding
- **Customer development playbooks** — discovery interview templates, ICP definition, pricing tests
- **Capital playbooks** — cap table modeling, term-sheet response patterns, follow-on decision frameworks
- **Legal playbooks** — template documents, IP licensing, common contract patterns
- **Operational playbooks** — meeting cadences, reporting formats, board governance
- **Exit playbooks** — M&A engagement patterns, IPO readiness frameworks
- **Crisis playbooks** — what to do when a portfolio company is failing, how to retire credentials, how to handle a contributor dispute

## — 02.3 Versioning and maintenance

Every playbook has an owner, a version number, and a "last reviewed" date. Playbooks are reviewed quarterly. If a playbook hasn't been reviewed in 6 months, the workspace automatically flags it for an owner check. If a playbook hasn't been used in 12 months, it gets archived.

## — 02.4 The Decision Memo template

Beyond standing playbooks, the Codified OS includes a steady-state production of Decision Memos. Every significant decision — formation approval, capital deployment, exit decision — produces a written memo with: context, options considered, decision, reasoning, dissents (if any), date.

The Decision Memos accumulate into the institution's reasoning history. Five years of memos is the closest a studio gets to institutional wisdom. Most workshop studios never produce them.

### CODIFIED OS — QUICK CHECK

- You can identify the owner of every active playbook
- Quarterly review cadence is enforced via workspace flags
- Decision Memos are produced for every significant decision (~5-15 per month)
- New operators can find the relevant playbook within 60 seconds

- Playbook archive contains retired versions, not deleted ones

SECTION 03

# The Credential Layer.

*A system to recognize and rank distributed contributors over time.  
Documented in detail in the Forkaia × STEALTH Joint Paper.*

03.1	Why credentials, not interviews
03.2	The Score architecture
03.3	Calibration + drift
03.4	Common pitfalls

### — 03.1 Why credentials, not interviews

Traditional studios run on interviews. Founders interview every operator. This works at 15 people and breaks at 50. The distributed studio replaces the interview as the primary trust mechanism with the credential — a portable, time-evolved signal of operator quality.

Interviews aren't eliminated; they become the final filter rather than the primary one. A contributor with a Forkaia® Score of 720 doesn't need 5 conversations to get assigned a complex task. The Score has already done most of the trust work.

### — 03.2 The Score architecture

The Forkaia® Score is a 300–850 work-readiness index. Five components: Project completion (25%), Supervisor evaluation (25%), Peer collaboration (15%), Self-direction (20%), Output quality (15%). Full methodology in the Joint Paper.

### — 03.3 Calibration + drift

Scores must be calibrated against a fixed baseline cohort to prevent drift. STEALTH calibrates quarterly against ~12,000 individuals across the combined Forkaia + STEALTH ecosystem. Drift detection: if the median Score across the population shifts by more than 15 points in a quarter, something is wrong with the calibration.

### — 03.4 Common credential pitfalls

- **Building the credential too early.** Need at least 100 evaluated outputs per active contributor before formalizing ranking. Premature credentialing creates legibility without signal.
- **Founder-preference encoding.** The first generation of any scoring rubric encodes the founders' preferences rather than universal quality signal. Plan for a 14-month recalibration cycle.
- **Score as performance review proxy.** The Score is not a performance review. It is a trust signal for task assignment. Conflating the two creates gaming.
- **No appeal process.** Contributors disputing their Score need a structured appeal path. Without one, Score becomes politically toxic.

SECTION 04

# The Writing Culture.

*The substrate that makes the other three pillars actually work. The deepest cultural prerequisite — and the one most workshop studios cannot adopt.*

04.1	Why writing matters more than meetings
04.2	The async default
04.3	Post-mortems
04.4	Disagree in writing

## — 04.1 Why writing matters more than meetings

A studio that does not write things down — decisions, lessons learned, post-mortems, hiring rationale, pricing experiments — will revert to a workshop within 12 months, regardless of how good the other three pillars are. Writing culture is the substrate.

Writing is not the same as documentation. Documentation is reactive (we document after the fact). Writing culture is proactive (we make decisions in writing in the first place).

## — 04.2 The async default

In a workshop studio, the default mode of operation is the meeting. In a distributed studio, the default is async writing. Meetings happen, but they become the exception rather than the rule.

STEALTH's operating-team norm: anything that requires more than 10 minutes of discussion gets a written brief first. Anything that involves 3+ people gets a Decision Memo. Anyone can request a meeting; meetings cap at 30 minutes and produce written outputs.

## — 04.3 Post-mortems

Every meaningful failure — a venture that didn't pan out, a hire that didn't work, a customer that churned — produces a post-mortem. The post-mortem is written by the person closest to the failure, reviewed by the operating team, and added to the playbook archive. Five years of post-mortems is a substantial body of institutional wisdom.

The post-mortem template: What happened. What we expected. Where the expectation diverged from reality. What we'll do differently. What we won't do differently. Lessons applicable to other situations.

## — 04.4 Disagree in writing

The hardest writing-culture norm to establish: disagreement happens in writing, not in meetings. When a contributor disagrees with a decision, the expectation is they write up the disagreement. The original decision-maker responds in writing.

This is uncomfortable at first. It is the single most important norm to establish. A studio that cannot disagree in writing will surface disagreements only verbally, and only some of the time, and only with some of the team. That is how studios silently collapse on the wrong decisions.

### WRITING CULTURE — QUICK CHECK

- Significant decisions happen in writing first, then verbally if needed
- Every meeting produces a written output, not just a verbal consensus
- Every meaningful failure produces a post-mortem within 14 days
- Disagreements are written, not just spoken

- New operators can audit the studio's reasoning trail without interviewing anyone

---

# Implementation roadmap.

*From workshop to distributed in 18–36 months.*

## — Months 1–6 · Writing culture first

The hardest pillar. Cannot be retrofitted. Start by making one specific norm change: **every significant decision gets a written brief**. The first 90 days are about establishing that norm. If you can't get the operating team to write decisions down in 90 days, the distributed model is not for you yet.

## — Months 6–12 · Codified OS

Once writing is normal, start codifying the existing playbooks. You will discover that 60% of what you thought was institutional knowledge was actually one person's preference. Document anyway. The act of writing forces clarity.

## — Months 12–18 · The Workspace

With writing culture and OS in place, the workspace becomes productive. Without them, the workspace is just an expensive task tracker. This is the year where you actually start onboarding distributed contributors at scale.

## — Months 18–36 · The Credential Layer

By month 18 you should have ~100 distributed contributors with at least 10 completed tasks each. That's the minimum data set to start calibrating a credential. Don't formalize earlier; the credential will encode noise instead of signal.

# Failure modes.

*Five ways distributed studios collapse back into workshops.*

- 1. The Workspace First mistake.** Building the workspace before the writing culture and codified OS. The workspace will produce low-quality output for ~12 months before the lack of OS becomes visible. Most failed distributed studio attempts fail this way.
- 2. Founder bandwidth re-emergence.** The founders, finding that distributed execution feels uncomfortably hands-off, start re-injecting themselves into individual tasks. Within 6 months the attention bottleneck is back. The fix: explicit founder no-touch policies on routine tasks.
- 3. Premature credentialing.** Trying to build the Score before you have 100 evaluations per active contributor. The resulting credential encodes noise, becomes politically toxic, and has to be unwound.
- 4. The Star contributor problem.** One distributed contributor becomes critical to multiple portfolio companies. When they leave, several companies stumble. The fix: deliberate redundancy. No contributor should be irreplaceable to more than one company.
- 5. The Writing decay.** Writing culture starts strong, decays over 18 months as the team gets busier. By month 24, decisions are happening verbally again and post-mortems are skipped. The fix: explicit writing-culture audits each quarter.

# Cost structure.

Approximate annual operating cost for a distributed studio operating at 100+ portfolio companies:

LINE ITEM	ANNUAL	NOTES
Core operating team (8 FTE)	\$2.4M	Founders + 6 senior operators at market comp
Workspace infrastructure	\$60K	SaaS tools (Notion, Linear), custom services, hosting
Distributed contributor compensation	\$1.8M	1,200 active contributors paid per-task; varies widely
Legal + compliance	\$240K	Heavy at studio scale — entity formation, IP, contracts
Research function	\$180K	Annual Report, DSI, quarterly briefs — small team
Travel + offsites	\$120K	Annual operating-team offsite + portfolio events
Studio overhead	\$200K	Office (minimal), insurance, professional services
<b>TOTAL</b>	<b>~\$5.0M / yr</b>	For a 700-company portfolio

Per-company operating cost: ~\$7K/yr at this scale. For comparison, workshop studios typically run \$50–200K/yr per active portfolio company. The distributed model is roughly an order of magnitude more cost-efficient at scale.

---

# The first 100 days.

*If you have committed to building a distributed studio, here is what to do in the first 100 days.*

## — Days 1–14 · Establish writing culture

- Write a one-page "How We Make Decisions" document. Distribute to the team.
- Establish the Decision Memo template. Use it for every significant decision starting Day 1.
- Hold one all-hands meeting per week — but require every meeting to produce a written output.

## — Days 14–45 · Pick the workspace foundation

- Evaluate 3 candidate tools (Notion, Linear, Airtable). Pick one.
- Configure the 7-field task template. Train the operating team.
- Start running existing operational work through the new workspace.

## — Days 45–75 · Begin codifying

- Identify the 3 most-repeated operational patterns. Write playbooks for them.
- Establish playbook ownership and quarterly review cadence.
- Begin assigning workspace tasks to 2–3 external contributors as a test.

## — Days 75–100 · Evaluate readiness for distribution

- Review the first 30 distributed-contributor task outputs. What's working? What's failing?
- Refine the rubric based on what you've learned.
- Decide whether to scale to 20+ distributed contributors in the next 100 days. If not, identify what to fix first.

# Hiring profile.

*The operating team of a distributed studio has different requirements than a traditional studio.*

## — The 4 must-haves for operating-team hires

- **Writing fluency.** If they cannot write a clear decision memo in 90 minutes, they will damage the writing culture. Test this in the interview.
- **Async tolerance.** Some people need same-day verbal validation to function. They are great employees in workshop studios and terrible employees in distributed ones. Screen for it.
- **Process orientation.** The distributed studio runs on process. Operators who resent process or feel constrained by it will work against the system.
- **Founder-grade judgment.** Each operator will be making decisions without supervision. They need to think like founders even if they are not.

## — What you do NOT need to optimize for

- Pedigree. The distributed model selects on output, not credentials.
- Specific sector experience. The Codified OS reduces sector dependency.
- Geographic proximity. Truly distributed teams work; choose the best operator regardless of location.
- Synchronous availability. If they are awake and producing within 24 hours, they are available enough.

# Governance & legal.

*The legal structure of a distributed studio looks more like a software company than a fund.*

## — Entity structure

STEALTH operates as a Delaware LLC with a holding-company structure. Each portfolio company is a separate LLC or C-corp held by the parent. Fund I is a parallel LP vehicle that holds equity positions in portfolio companies; the parent LLC operates the studio itself and is compensated through operating fees + carry from the fund.

## — Contributor relationships

Distributed contributors are independent contractors, not employees. They work on a per-task or per-engagement basis. Tax treatment is 1099 (US) or equivalent. Contributors retain IP rights to background work; foreground work assigned through workspace tasks belongs to the studio.

## — IP architecture

Playbooks, templates, and operating-system artifacts are IP of the parent studio. Each portfolio company holds its own commercial IP. The studio licenses operational IP to portfolio companies on standardized terms.

## — Conflict-of-interest disclosures

STEALTH operates Fund I + research function + CVS engagements. Three lines of business that can create conflicts: (1) the fund could over-allocate to companies the studio operates, (2) the research function could favorably treat the studio's own portfolio, (3) CVS engagements could compete with Fund I deals. All three are mitigated through governance separation + public disclosure.

# Year 1, 2, 3 milestones.

*What "on track" looks like for a distributed studio built from scratch.*

## — Year 1 — Foundation

- Writing culture established and audited monthly
- Codified OS contains at least 25 playbooks across the 8 categories
- Workspace processes ~50 tasks per week through 10–20 distributed contributors
- First 8–12 portfolio companies in formation or operating
- Operating-team size: 4–6 FTE

## — Year 2 — Scaling

- Codified OS contains ~75 playbooks; quarterly review cadence is reliable
- Workspace processes ~300 tasks per week through 100+ distributed contributors
- Portfolio at 50–100 active companies; first exits in pipeline
- Operating-team size: 6–10 FTE
- First credential calibration completed

## — Year 3 — Compounding

- Codified OS at 150+ playbooks; some now retired and archived
- Workspace at 1,000+ weekly tasks; 500+ distributed contributors active
- Portfolio at 200–400 active companies; multiple Series A clearances
- Operating-team size: 10–15 FTE
- Credential layer producing portable Score with cross-organization recognition
- First Annual Report or equivalent research publication

---

# Appendix — templates & checklists.

*Quick-reference templates the playbook references. Available in editable form at [stealth1000.com/playbook](https://stealth1000.com/playbook).*

## — Available templates

- **T-01 · Workspace task template (7 fields)** — page 5
- **T-02 · 5-dimension supervisor rubric** — page 6
- **T-03 · Decision Memo template** — for any significant decision
- **T-04 · Post-mortem template** — for any meaningful failure
- **T-05 · Playbook header standards** — owner, version, last reviewed
- **T-06 · Contributor onboarding checklist** — first-week ramp
- **T-07 · Quarterly writing-culture audit** — 5-question self-assessment
- **T-08 · Credential calibration data template** — for the Score

## — Closing note

This playbook is a working document. STEALTH Research publishes a revised version annually; v2 expected June 2027 incorporating peer feedback received between now and then.

If you build a distributed studio using anything in this playbook, write us. We want to learn from your implementation. [support@stealth1000.com](mailto:support@stealth1000.com).

---

The Distributed Studio Playbook · Volume I · June 2026

STEALTH Research · CC BY 4.0 · [support@stealth1000.com](mailto:support@stealth1000.com)