INFORMATION SECURITY LEADERSHIP FIELD GUIDE

The Security Leader's Operating System

A Practical Field Guide to Reclaiming Time, Delegating Decisions, and Protecting Leader-Only Work

L. Brent Huston

MICROSOLVED, INC.

ELIMINATE • DELEGATE • SIMPLIFY • AUTOMATE • MAINTAIN

FIRST EDITION • 2026 |

First edition, 2026

Cover illustration generated with artificial intelligence under the editorial direction of MicroSolved, Inc.

EDSAM was created by L. Brent Huston. This field guide is published by MicroSolved, Inc. and applies the EDSAM mental model to the personal work of information security leadership.

Contents at a Glance

Select any entry to jump directly to that section.

A Personal Note from L. Brent Huston

I have spent a large part of my career helping people make difficult security decisions with incomplete information, limited resources, and a clock that never seems to stop. The technologies change. The regulations change. The attacks change. The pressure does not.

If you lead an information security program, you probably know the feeling. Your day starts with a plan and is quickly claimed by an incident question, a vendor review, an executive request, an audit issue, a project escalation, and a meeting that somehow requires six people to avoid making one decision. You remain accountable for the program, so the work keeps finding you. Eventually, being the person who can handle almost anything becomes the reason you have time to think about almost nothing.

EDSAM grew out of my attempt to solve that problem for myself.

It was never intended to be a productivity trick or a clever acronym. It became a way to interrogate work before giving that work more of my life. Should this exist? Who should own it? What is the smallest useful outcome? Which repeatable parts can run without consuming another slice of attention? What deserves to remain in my hands because the judgment, relationship, or consequence is genuinely mine?

Those questions have followed me through security operations, advisory work, research, product development, writing, and the strange collection of experiments that becomes a career. They have also survived waves of technology that promised to save time and often created new things to administer instead.

Artificial intelligence is the latest and most consequential of those waves. I use it. I build with it. I am excited by what it can do. I am equally interested in the authority we quietly hand it, the context it does not have, the maintenance it creates, and the speed with which a helpful experiment can become invisible infrastructure. LLMs, GPTs, skills, and agents can make EDSAM far more powerful, but only if the mental model remains in charge of the tools.

That is why this guide starts with the leader rather than the technology.

My hope is that you use these pages with the actual work in front of you. Decline one meeting that exists only because it always has. Transfer one outcome to the person who should own it and give them enough authority to succeed. Turn one sprawling deliverable into the decision it is meant to support. Build one bounded assistant that prepares evidence without pretending to own the judgment. Protect one hour for the work only you can do.

Small changes matter because systems compound. A saved hour is useful. A redesigned flow that prevents the same demand from returning every week is leverage. A clear decision boundary makes a team stronger. A documented method can help someone you may never meet avoid rebuilding the lesson from scratch.

I also believe a system is not real if it works only on your best days. It should make a bad day better and keep the worst day at least minimally productive. That is when structure earns its place.

That last point matters deeply to me.

I have long wanted my work to contribute to a legacy of knowledge: practical ideas, systems, questions, tools, and hard-earned lessons that remain useful beyond the engagement, the project, the product, or the person who first wrote them down. Not a monument. A working library. Something other people can test, improve, teach, and carry forward.

This field guide is part of that effort. It is not the final word on security leadership or personal operating systems. It is an invitation to make the work more intentional and then add what you learn to the body of knowledge available to the next leader.

If EDSAM gives you back enough attention to make one better decision, coach one person more thoughtfully, prevent one avoidable failure, or leave behind one system that continues to help after you step away, it has done its job.

Use it. Challenge it. Adapt it. Make it yours. Then pass the useful parts on.

— L. Brent Huston

How to Use This Guide

This is a field guide, not a book to admire and postpone.

Take the capacity diagnostic, then read the first two chapters to understand the system. Go directly to the chapter matching your strongest source of overload. When you are ready to apply the full method, use the completed SaaS walkthrough as a model and complete the weekly review and 30-day reset with work from your calendar, inbox, and program—not hypothetical examples.

The goal is not to do everything faster. It is to ensure that fewer demands require your time and attention, and that those that survive deserve leadership.

You will need:

- Your calendar for the last two weeks
- Your current task list or project tracker
- A sample of recent email and messaging requests
- A list of recurring meetings and reports
- Thirty uninterrupted minutes each week

The Promise

Security leaders are rarely short of work. They are short of protected attention.

Every vulnerability, vendor, audit request, incident, project, policy question, executive concern, and technology change can arrive as an apparent security obligation. Because the leader is accountable for the program, those obligations tend to migrate upward. The result is a leader who is present everywhere but able to think deeply almost nowhere.

EDSAM creates a filter between incoming demand and personal obligation:

1. **Eliminate** what does not need to happen.
2. **Delegate** what does not require your authority or judgment.
3. **Simplify** what is more complicated than its required outcome.
4. **Automate** the predictable parts of the surviving work.
5. **Maintain** only the work that must, for now, remain with you.

Used repeatedly, EDSAM does more than organize a workload. It changes the shape of the work.

Start in Five Minutes

If you cannot read the full guide today, take one demand currently consuming attention and write five answers:

1. **Eliminate:** What credible consequence follows if this does not happen?
2. **Delegate:** Who is closest to the outcome, and what authority would they need?
3. **Simplify:** What is the smallest output that protects the outcome?
4. **Automate:** Which preparation, routing, reminder, or evidence step follows a stable rule?
5. **Maintain:** What remaining decision genuinely requires your authority, judgment, or relationship?

Then take one action. Decline the meeting. Name the owner. Shorten the deliverable. Create the rule. Block time for the decision. EDSAM becomes useful when it changes real work.

Capacity Diagnostic: Find Your Starting Point

An overloaded leader rarely needs every part of the system at once. Use this short diagnostic to identify the pattern creating the most pressure. Score each statement **0 = rarely, 1 = sometimes, or 2 = often** based on the last four weeks.

1. My planned day is routinely replaced by requests and escalations from other people.
2. I attend recurring meetings where my role and the decision required are unclear.
3. Problems reach me without an owner, recommendation, decision deadline, or consequence of delay.
4. Routine work waits for my approval even when it falls inside an established policy or tolerance.
5. I personally assemble, reconcile, or reformat information that other people already possess.
6. Important non-urgent work such as strategy, coaching, architecture, and relationship building repeatedly slips.
7. Work requiring my judgment begins after normal working hours because the day is consumed by coordination.
8. Delegated work returns to me as questions, status updates, or requests to redo the analysis.

9. Commitments are scattered across email, chat, meetings, tickets, notes, and memory.
10. An automation or AI assistant requires enough checking, correction, and maintenance that the promised time savings are uncertain.
11. My team or stakeholders cannot explain which decisions they may make without me.
12. Even after removing low-value work, required security outcomes exceed sustainable capacity.

Add the scores. The total is a signal, not a diagnosis:

- **0-7 — Local friction:** Start with one recurring demand and the five-minute exercise. A few operating corrections may create noticeable relief.
- **8-15 — Systemic overload:** Complete the weekly review and 30-day reset. The problem is probably not the number of tasks alone; it is the way work enters, moves, and escalates.
- **16-24 — Leader dependency or capacity failure:** Treat this as an operating-model issue. Use EDSAM to reduce avoidable work, but also prepare evidence for changes in authority, service levels, staffing, or risk acceptance.

Use the strongest symptom to choose your first route:

- **Calendar saturation:** Chapters 3 and 9
- **Approval or escalation bottleneck:** Chapters 2 and 4
- **Oversized deliverables and too many handoffs:** Chapter 5
- **Repetitive preparation or tool supervision:** Chapters 6 and 13
- **Strategy, coaching, and relationships pushed aside:** Chapter 7
- **Required outcomes still exceed capacity:** Chapter 11

Before changing anything, capture three baseline facts: your active personal commitments, hours spent after normal working time, and percentage of the last two weeks spent on work that genuinely required your authority or judgment. Those numbers will make the improvement—or the remaining capacity gap—visible.

If your score is 8 or higher, begin with the 30-Day EDSAM Reset.

1. The Security Leader's Real Constraint

Demand Is Not the Same as Work

A request is not automatically your work because it reached your inbox. A risk is not automatically owned by security because it has a security dimension. A meeting is not automatically necessary because the subject matters. A document is not automatically valuable because an auditor, customer, or executive asked for it.

Security leadership becomes overloaded when three distinctions disappear:

- **Accountability versus execution:** You may be accountable for a program outcome without personally producing every artifact or approving every action.
- **Security ownership versus risk ownership:** Security advises, enables, monitors, and escalates. Business and technology owners still own the risks created by their decisions and systems.
- **Availability versus leadership:** Being continuously reachable is not the same as providing timely judgment.

When these boundaries blur, the leader becomes the organization's universal security queue.

The Three Currencies: Money, Attention, and Time

EDSAM evaluates work through three currencies, abbreviated MAT:

- **Money:** Cash, budget, contractual expense, opportunity cost, or avoided loss.
- **Attention:** Cognitive effort, emotional load, context switching, political effort, and the ability to think clearly.
- **Time:** Your hours, the team's hours, other people's time, and machine time.

Attention is usually the hidden constraint. A fifteen-minute task can consume far more capacity when it interrupts incident planning, forces a difficult context switch, or remains unresolved for days.

Rate each currency High, Medium, or Low. Precision is less important than consistency.

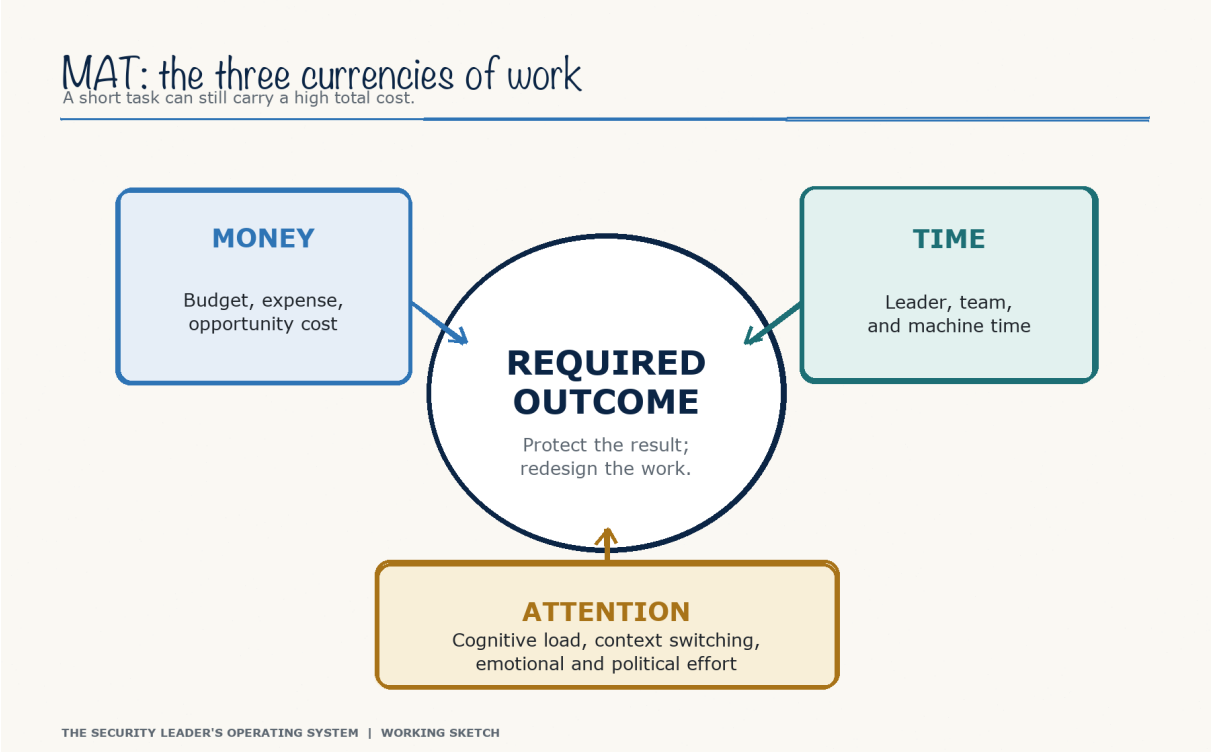

Figure 1. MAT exposes the combined Money, Attention, and Time cost of preserving a required outcome.

MAT workload scan

For each recurring activity, record:

- What outcome does it produce?
- Money cost: High, Medium, or Low
- Attention cost: High, Medium, or Low
- Time cost: High, Medium, or Low
- Does it require leader authority or judgment?
- What would happen if it stopped for thirty days?

Apply EDSAM first to high-attention, low-value work. Protect high-attention, high-value work from interruption.

Leadership Work Versus Management Exhaust

Leadership work changes direction, resolves uncertainty, builds capability, or accepts consequence. Management exhaust is the residue created by the way the organization coordinates work.

Examples of leadership work include:

- Setting risk appetite and program priorities
- Translating technical exposure into enterprise decisions

- Coaching security leaders and developing successors
- Resolving ownership and funding conflicts
- Leading through serious incidents
- Building trust with executives, boards, regulators, customers, and partners
- Challenging systemic assumptions

Examples of management exhaust include:

- Reformatting the same metrics for several audiences
- Manually reminding owners of overdue work
- Attending meetings only to provide status already available elsewhere
- Reconciling duplicate trackers
- Reviewing every item because escalation criteria are unclear
- Repeating standard security explanations
- Approving routine actions that fit established policy

Administration can be valuable. Leadership attention should not be its default production mechanism.

The Outcome Guardrail

Apply EDSAM to the method of achieving a security outcome. Do not use it casually to erase the outcome.

Before changing an activity, identify:

1. The security or business outcome it supports
2. The obligation or risk that makes the outcome necessary
3. The evidence that demonstrates the outcome
4. The owner accountable for it
5. The consequence if the outcome fails

You may eliminate a duplicate access report, but not the ability to detect inappropriate access. You may simplify vendor diligence, but not ignore concentration risk in a critical supplier. You may automate evidence collection, but not acceptance of material risk.

EDSAM is not a substitute for adequate staffing. If required outcomes remain uncovered after unnecessary work is removed and ownership is corrected, you have evidence of a capacity gap. Document the missing outcomes, exposure, options, and consequence. Efficiency should reveal under-resourcing, not conceal it through unsustainable effort.

One System, Different Leadership Contexts

The filters remain the same across roles, but the operating boundary changes:

- **Solo security leader:** Emphasize elimination, business ownership, managed services, standard patterns, and explicit executive risk decisions. Do not become the manual control layer for the entire organization.
- **Manager of a security team:** Emphasize delegation levels, coaching, clear escalation criteria, and the removal of approval dependency.
- **Enterprise CISO:** Emphasize decision rights, portfolio tradeoffs, executive communication, systemic risk, and the capability of subordinate leaders.
- **vCISO or fractional leader:** Emphasize client-specific intake, strict scope boundaries, reusable templates, scheduled decision windows, and separation of advice from client-owned execution.

EDSAM should reduce dependency on the leader while increasing the quality of leadership applied to material issues.

2. The Security Leader’s Operating System

EDSAM means **Eliminate, Delegate, Simplify, Automate, Maintain**. Those five moves are the decision engine inside a larger personal operating system. The system captures demand, distinguishes urgency from importance, changes the work before scheduling it, and protects only the small remainder that genuinely needs the leader.

The Operating System on One Page

The entire method can be held in one flow:

> Demand → Capture → Classify → Apply EDSAM → Place surviving work → Review → Measure

One trusted intake prevents commitments from hiding. MAT and the Eisenhower Matrix reveal cost, importance, and urgency. EDSAM changes the work. Four queues make the remainder visible. Work-in-progress limits protect capacity. A daily, weekly, monthly, and quarterly cadence keeps the system honest. Capacity measures show whether attention was reclaimed; security outcome guardrails show whether it was reclaimed safely.

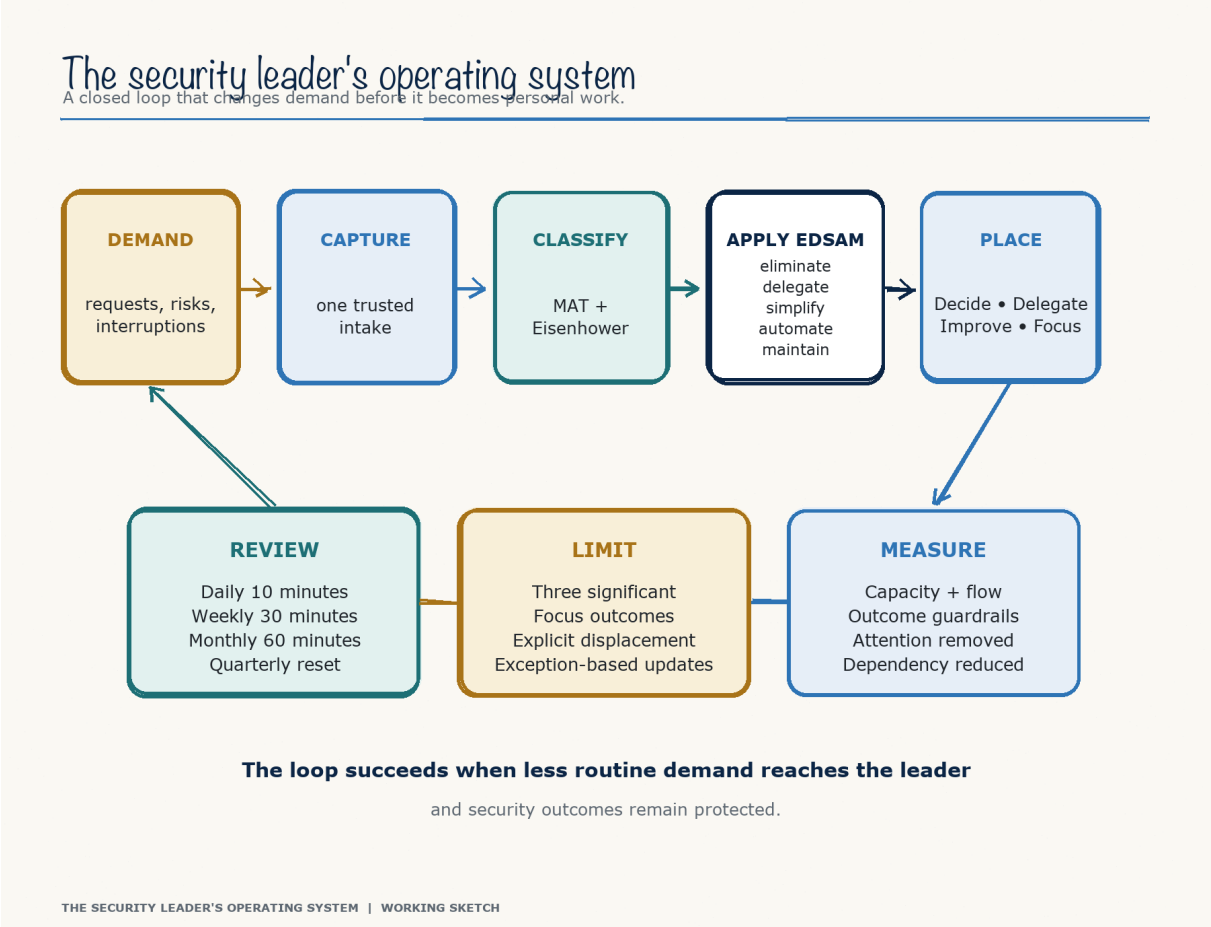

Figure 2. The operating system changes demand before it becomes personal work, then reviews and measures the result as a closed loop.

This is not another layer of administration. It is the minimum control loop required to prevent raw demand from becoming personal obligation. If a tool adds more ceremony than this loop removes, simplify the tool.

One Door for New Demand

Overload is difficult to manage when commitments live in email, chat, meeting notes, tickets, documents, and memory. Choose one trusted intake location. It can be a task manager, a board, a notebook, or a simple list. The technology matters less than the rule:

> No request becomes a commitment until it is captured and processed.

Capture only enough information to decide what happens next:

- Request or trigger
- Required outcome
- Requester or stakeholder

- Real deadline and source of the deadline
- Risk or obligation involved
- Proposed owner
- Decision needed
- MAT rating

The Eisenhower Matrix as a Triage Lens

The Eisenhower Matrix separates work along two dimensions: importance and urgency. It is valuable because leaders often confuse a fast clock with a material outcome. EDSAM solves a different problem. The matrix helps determine when attention may be needed; EDSAM determines whether the work should exist, who should own it, how small it can become, what can run predictably, and what must remain with the leader.

Use the matrix as a routing lens, not a final disposition. An item can be urgent and still deserve elimination. An important item can be delegated. A non-urgent item can require leader-only judgment. The quadrant creates a useful hypothesis; EDSAM tests it.

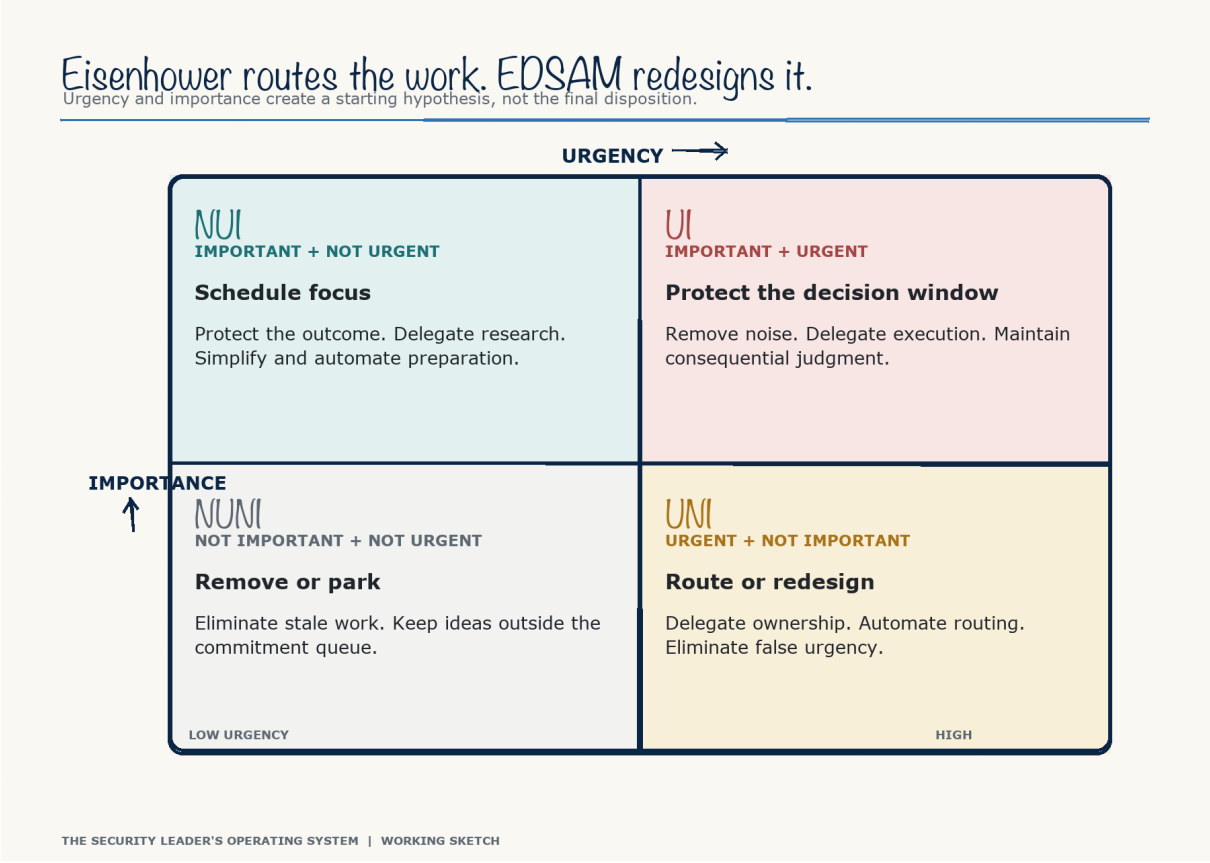

Figure 3. The Eisenhower Matrix routes work by importance and urgency; EDSAM tests what the work should become.

Important and Urgent

These items need timely attention because both the outcome and the clock matter. Examples include a material incident decision, a regulatory deadline, an expiring containment window, or an executive commitment due today.

Apply EDSAM without delaying necessary action:

- **Eliminate** noise, duplicated coordination, and nonessential reporting around the event.
- **Delegate** investigation, evidence gathering, logistics, and execution to defined owners.
- **Simplify** the immediate requirement to the smallest safe decision or outcome.
- **Automate** trusted preparation, enrichment, routing, timestamps, and evidence preservation.
- **Maintain** the consequential decision, relationship, communication, or risk acceptance that requires leadership.

Urgent and important should not mean that the leader performs every task. It means the leader protects the decision window while the operating system supplies prepared, owned work.

Important and Not Urgent

This quadrant contains much of the work leaders say matters but repeatedly postpone: strategy, architecture direction, succession, control redesign, relationship building, exercises, research, and capability development.

This is usually the leader's Focus queue. Schedule it before urgency consumes the calendar. Maintain the outcome and judgment, but delegate research, simplify the deliverable, and automate preparation. EDSAM keeps important work from becoming urgent through neglect.

Urgent and Not Important

This is where other people's deadlines, routine escalations, status chasing, low-risk approvals, and poorly routed requests often appear. Treat this quadrant as a delegation and system-design signal.

Ask whether the urgency is real, who owns the outcome, and what recurring rule would keep the demand from returning. Delegate what belongs elsewhere. Simplify the request to the actual decision. Automate routing and reminders. Eliminate urgency that exists only because the workflow is unclear.

Not Important and Not Urgent

This is the primary elimination quadrant. Remove stale commitments, unused reports, recurring meetings without decisions, abandoned experiments, low-value alerts, and someday items that no longer reflect a genuine intention.

Some ideas are worth retaining without becoming commitments. Move those to a separate idea store with a review trigger. Do not let possibility occupy the same queue as obligation.

TaskGrid: Integrating the Mental Models into a Tool

I built TaskGrid for myself because I wanted these mental models to exist somewhere more concrete than a notebook page or a good intention. I often share it by request with people interested in how I work. It currently runs on Linux, macOS, and Windows.

TaskGrid is one tool in my overall system, not a requirement for using EDSAM. It makes the method repeatable by combining four Eisenhower queues with other useful views:

- **UI - Urgent and Important:** Protect the decision or response window.
- **NUI - Not Urgent and Important:** Schedule strategic, preventive, developmental, and relationship work.
- **UNI - Urgent and Not Important:** Delegate, reroute, simplify, or redesign the recurring rule.
- **NUNI - Not Urgent and Not Important:** Eliminate, archive, or move to a non-committed idea store.
- **Control / Influence / No Control:** Separate action from influence and observation so an outcome outside my control does not masquerade as a task I can complete.
- **Must / Should / Want:** Expose whether a commitment is required, inherited without examination, or deliberately chosen.
- **Week / Date / Project / Decision / Ideas:** Separate active work, real dates, multi-step outcomes, waiting judgments, and possibilities.

The result is a grid rather than another undifferentiated list. A demand can be understood by urgency, importance, agency, commitment, horizon, and required decision before it consumes attention.

How I Run EDSAM Through TaskGrid

My practical flow is:

1. **Capture:** Put a demand into the system before treating it as a commitment.
2. **Classify:** Identify its urgency, importance, control boundary, commitment type, and time horizon.
3. **Apply EDSAM:** Eliminate, delegate, simplify, automate, or consciously maintain it.
4. **Place the surviving outcome:** Move it into the appropriate active view with an owner, decision, next action, or review trigger.
5. **Review the grid:** Clear stale work, protect important non-urgent work, inspect repeated urgent work for system failures, and keep the maintained set small.

The important shift is that UI becomes a prepared decision queue, NUI becomes protected Focus work, UNI becomes a system-improvement signal, and NUNI becomes an elimination test instead of a low-priority warehouse.

TaskGrid, Project Chief, and AI Assistance

In my broader system, TaskGrid can remain the local, portable system of record while Project Chief provides a read-only executive layer across selected task, calendar, and mail context. The assistant removes routine noise, groups commitments, and prepares a small decision queue.

The authority boundary matters. The assistant may retrieve, summarize, compare, and draft; it does not silently create commitments, send messages, or turn a draft into an external action. The tool reduces the cost of seeing and preparing the work. I remain responsible for what enters the system and what leaves it.

TaskGrid shows what happens when a mental model becomes a durable personal tool: noise is filtered, preparation moves to bounded assistance, many sources become a few decisions, stable routing runs automatically, and material judgment remains visible and human-owned.

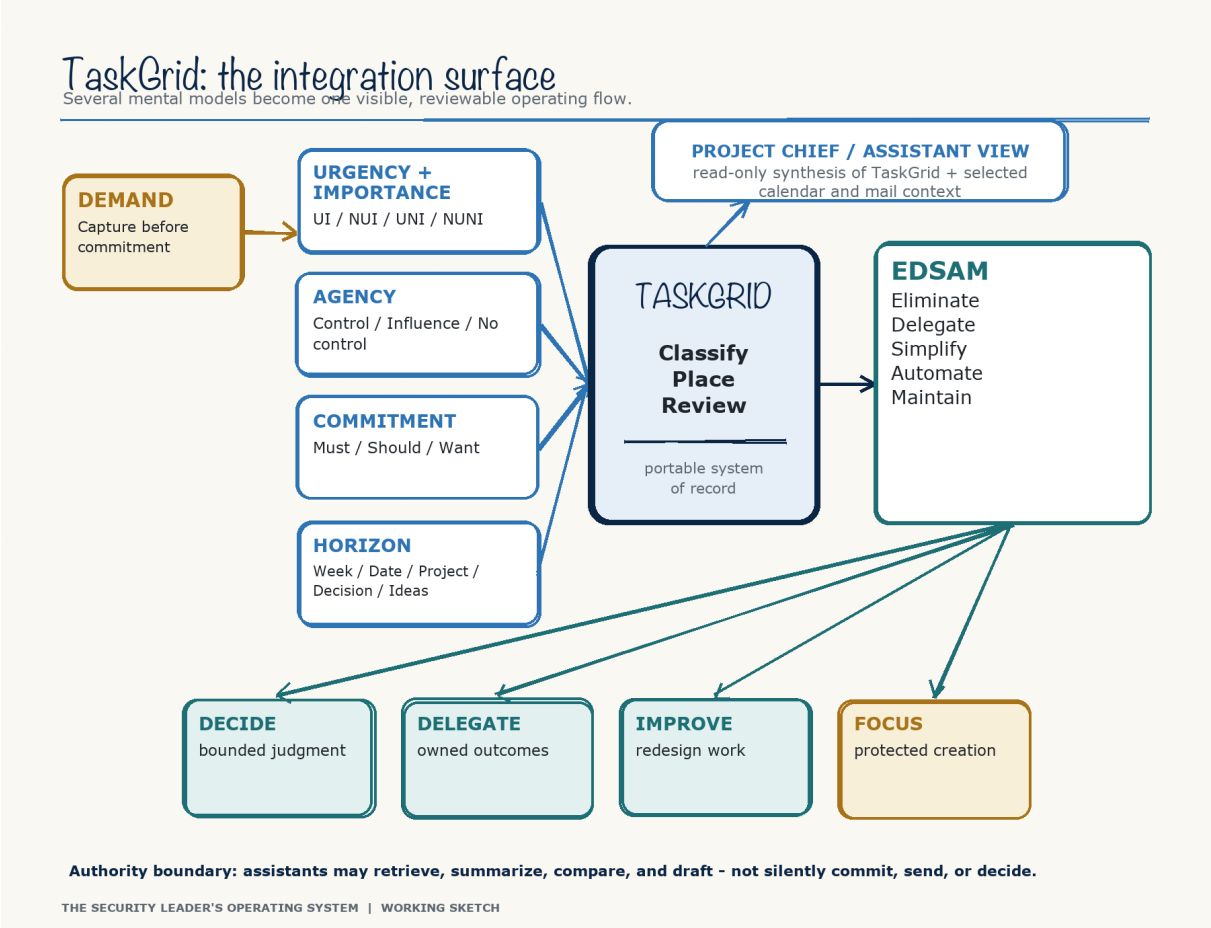

Figure 4. TaskGrid turns several mental models into one visible operating flow, with assistants bounded to read-only synthesis and preparation.

The Five Dispositions

Every captured item should leave intake with one of five dispositions:

Eliminate

Stop, decline, defer to an explicit trigger, reduce frequency, or remove the unnecessary portion.

Delegate

Assign the outcome to someone with sufficient context, authority, capacity, and a clear escalation path.

Simplify

Reduce the work to the minimum output that achieves the outcome. Then reconsider delegation and automation.

Automate

Use a reliable rule, integration, template, or tool to perform predictable steps and surface exceptions.

Maintain

Keep the work personally because it requires your judgment, authority, relationship, or accountability and no safe alternative currently exists.

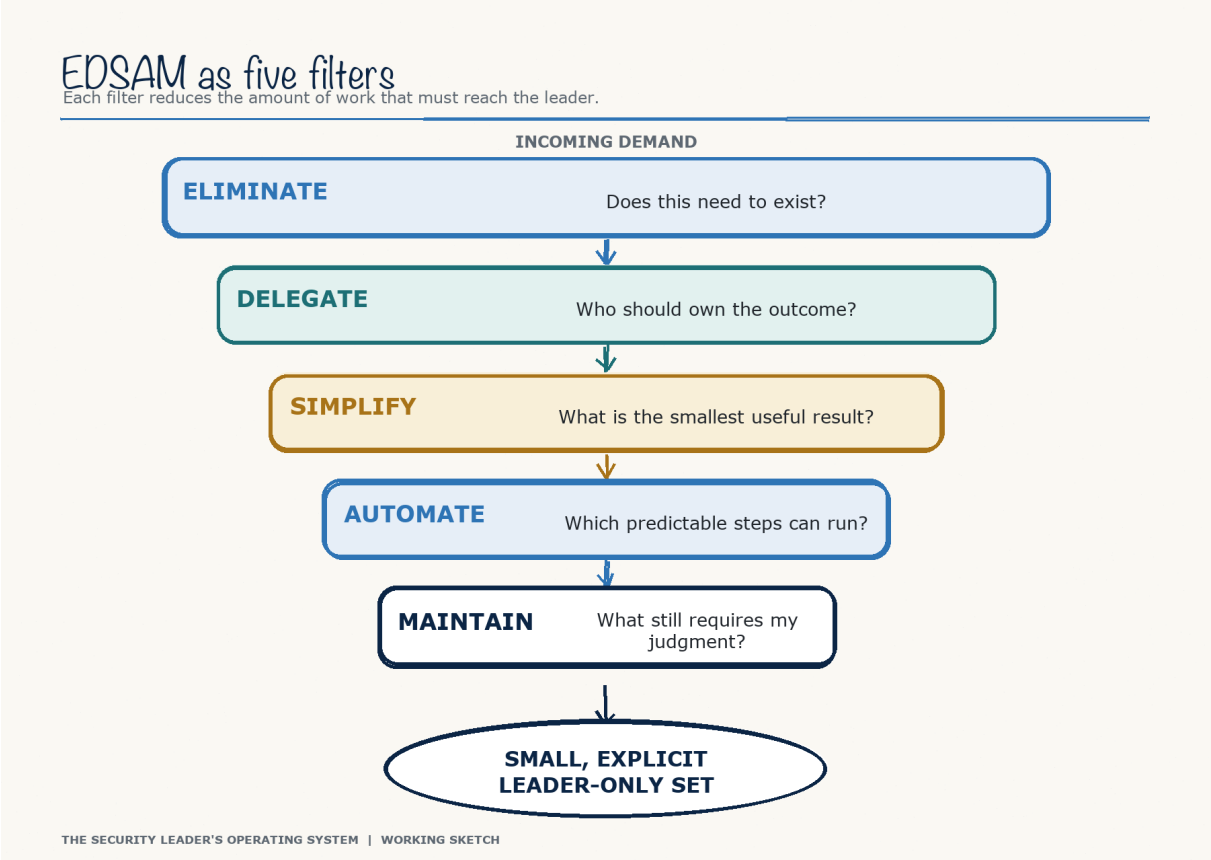

Figure 5. The five EDSAM filters progressively reduce incoming demand to a small, explicit leader-only set.

Four Active Queues

A leader does not need five separate backlogs. Use four active views:

1. **Decide:** Bounded decisions requiring your judgment or authority
2. **Delegate:** Outcomes assigned to others, visible only until completion or escalation
3. **Improve:** Recurring work being eliminated, simplified, or automated
4. **Focus:** The small set of outcomes you are personally producing

Eliminated items leave the active system. Automated work appears only when an exception occurs.

Work-in-Progress Limits

An unlimited priority list is not a priority system. Begin with no more than three significant items in Focus:

- One material risk or governance outcome
- One program improvement outcome
- One people or relationship outcome

This is a starting constraint, not a law. Its purpose is to force an explicit choice when new work appears. If a fourth priority enters, identify what it displaces.

The Personal Ownership Test

Before accepting work, ask:

1. Does this require authority that only I hold?
2. Does it require judgment that cannot yet be delegated safely?
3. Does my relationship with the stakeholder materially change the outcome?
4. Is the action irreversible or above established risk tolerance?
5. Would transferring it create more risk than retaining it?

If the answer to all five is no, it probably should not remain personal work.

Interruption Rules

Define the conditions that justify immediate interruption. Examples include:

- Potentially material business impact
- Threat to life, safety, or critical operations
- A rapidly closing containment or recovery window
- Legal, regulatory, contractual, or notification consequences
- An irreversible action

- Executive, customer, regulator, or public communication
- A decision exceeding delegated authority or risk tolerance

Everything else enters a normal decision window. This protects responsiveness where it matters without allowing urgency to become the default operating condition.

Keep a Decision Ledger

Security leaders repeat analysis when prior decisions are trapped in email, meeting memory, or disconnected tickets. Maintain a lightweight ledger for material choices:

- Decision and date
- Decision owner
- Context and credible risk scenario
- Options considered
- Decision and rationale
- Residual risk
- Evidence and confidence
- Commitments and owners
- Review or expiration trigger

The ledger reduces repeated debate, makes risk acceptance visible, improves executive continuity, and gives automation a trusted source for retrieving precedent. It should record decisions, not become another narrative report.

Before and After: A Security Leader’s Tuesday

Our manager’s Tuesday begins with a customer questionnaire, vulnerability meeting, project reviews, audit request, chat escalations, executive metrics, and a risk memo that would otherwise begin after dinner. EDSAM changes the path:

- Approved responses reduce the questionnaire to two novel assertions.
- An exception report replaces the vulnerability meeting unless exposure exceeds tolerance.
- An approved architecture pattern removes one review; a one-page decision packet prepares the novel case.
- The control owner supplies audit evidence, and chat escalations require a recommendation and deadline.
- Metrics populate automatically so the manager interprets material change instead of copying numbers, returning several hours and an uninterrupted block for the risk decision only the manager can make.

3. Eliminate: Stop Accepting Every Demand

Elimination is the highest-leverage filter because work that does not exist consumes no time, attention, budget, coordination, or reporting.

It is also the filter security leaders resist most. Security culture rewards vigilance, and leaders may fear that stopping an activity will look careless. The answer is disciplined elimination: understand the outcome, dependencies, and consequence before stopping the work.

The Elimination Questions

Ask:

- What happens if this is not done?
- Who uses the output, and what decision does it change?
- Is it required by law, contract, policy, risk treatment, or an explicit commitment?
- Is another process already producing the same outcome or evidence?
- Has the activity outlived the condition that created it?
- Could it happen less frequently?
- Could we eliminate only the leader's involvement?
- Is the activity reducing risk, or only documenting motion?

Meeting Elimination

Meetings are recurring claims on future attention. Apply a simple test to each one:

- Is there a decision, conflict, collaboration need, or sensitive conversation?
- Is synchronous participation necessary?
- Is your role specific and necessary?
- Can the outcome be achieved asynchronously?
- Could you attend only the decision portion?
- Could the meeting occur only when an exception exists?

Eliminate or decline meetings whose primary purpose is reading status aloud. Replace them with an exception update containing changes, blockers, decisions, and overdue commitments.

Meeting response script

"I do not think my attendance is needed for the full meeting. Please send the decision, options, recommendation, and deadline. I will respond during today's decision window. Add me if the discussion reaches the defined escalation condition."

Reporting Elimination

Inventory recurring reports and ask each recipient:

- Which section do you use?
- What decision does it support?
- What would be lost if the report stopped?
- Would an exception alert or quarterly trend be sufficient?

Possible outcomes include stopping the report, reducing its audience, removing unused measures, lowering its frequency, or generating it only when a threshold changes.

Approval Elimination

Approvals accumulate after incidents and audit findings. They often become permanent after their original purpose disappears.

For each approval, determine:

- What failure is it intended to prevent?
- Can policy, technical guardrails, or delegated authority prevent that failure?
- What risk threshold requires leader review?
- Is the approval creating accountability or merely transferring anxiety upward?

Replace routine approval with clear decision rights. Reserve leader approval for exceptions, material risk, irreversible actions, and commitments above delegated authority.

What Not to Eliminate

Do not eliminate activities merely because they are uncomfortable, politically difficult, or currently produce bad news. Preserve:

- Independent assurance
- Required segregation of duties
- Evidence needed for material obligations
- Challenge and dissent
- Incident readiness and recovery testing
- Direct contact with important stakeholders

- Time for coaching, succession, and strategic thinking

Elimination Worksheet

Select one recurring activity and answer:

Activity:

Intended outcome:

Actual consumer:

Decision influenced:

Obligation or dependency:

Consequence if stopped for thirty days:

Proposed change: Full elimination / partial elimination / reduced frequency / no change

How the outcome will remain protected:

Tuesday: Remove the Reflex Meeting

On our leader's redesigned Tuesday, the vulnerability meeting no longer happens by default. The system produces an exception report; the leader joins only when exposure exceeds delegated tolerance. One calendar block disappears, and the security outcome—timely treatment of material exposure—remains intact.

Do This Now

Choose one recurring meeting or report. Ask its consumer what decision it changes. If the answer is unclear, suspend it for thirty days with a named restoration trigger.

Proof It Worked

Record the hours removed, the outcome preserved, and whether any legitimate dependency failed. No complaint is not enough; look for evidence that the decision or control still occurred.

Use the meeting test above with the EDSAM Intake Card in Chapter 14.

4. Delegate: Put Decisions at the Right Level

Delegation is not task movement. It is ownership design that lets the organization act without waiting for the security leader.

Security programs centralize when only the leader can interpret policy, accept uncertainty, communicate with executives, or decide what "good enough" means. Explicit authority, competence, evidence, and escalation rules reverse that dependency.

Delegate Outcomes, Not Motion

Weak delegation sounds like: "Please handle the vendor review."

Strong delegation defines:

- **Outcome:** What must be true when the work is complete
- **Owner:** One accountable person
- **Authority:** Decisions the owner may make without permission
- **Constraints:** Policy, budget, risk, legal, or timing boundaries
- **Evidence:** What demonstrates completion and quality
- **Deadline:** When the outcome is required and why
- **Escalation:** Conditions requiring assistance or higher authority

The Security Ownership Boundary

The security function should not quietly inherit every technology or business risk.

Examples:

- The system owner owns remediation of system vulnerabilities.
- The data owner approves who should access sensitive information.
- The business sponsor owns the decision to use a supplier.
- Procurement and legal own their portions of supplier contracting.
- Human resources owns employment processes that trigger access changes.
- Technology teams own secure operation of the platforms they manage.
- Security defines expectations, advises, tests, monitors, and escalates.

Leadership workload falls when accountability becomes visible and decisions occur closer to the work.

Delegation Levels

Use a clear ladder:

1. **Investigate:** Gather facts and return for direction.
2. **Recommend:** Analyze options and propose a decision.
3. **Decide, then inform:** Act within defined boundaries and report the outcome.
4. **Own:** Manage the outcome and report only exceptions or material changes.

State the level when assigning work. Ambiguous delegation causes either unnecessary escalation or unauthorized decisions.

The Escalation Contract

Require escalations to contain:

- Business context
- Current exposure or consequence
- Actions already taken
- Options considered
- Owner's recommendation
- Decision needed
- Decision deadline
- Consequence of delay

This prevents raw problems from being transferred upward without analysis.

Delegation Failure Modes

Watch for:

- Responsibility without authority
- Work assigned without removing other commitments
- Multiple owners and no accountable person
- Escalation based on discomfort rather than thresholds
- Delegating sensitive judgment before capability exists
- Constant status requests that reclaim the delegated work
- Treating a missed outcome as proof that delegation never works

When delegation fails, inspect the system before taking the work back. The missing element may be context, authority, skill, capacity, clarity, or feedback.

Delegation Brief

Outcome:

Accountable owner:

Why it matters:

Authority granted:

Constraints:

Definition of done:

Evidence required:

Deadline and source:

Escalate when:

Update cadence: Exceptions only / milestone / scheduled

Tuesday: Move the Work, Keep the Boundary

The customer questionnaire is no longer the leader's document to produce. An analyst owns the approved-response library and drafts the routine answers. The leader sees two novel assertions with evidence, a recommendation, and a deadline. Delegation removes production work without delegating accountability for material claims.

Do This Now

Take one item you have completed at least three times. Write a delegation brief that grants a named owner one higher level of authority than they hold today.

Proof It Worked

The owner completes or escalates the outcome inside the stated boundary, and the leader does not recreate the work. Track both decision quality and the number of avoidable escalations.

Use the Delegation Brief here and the Decision Ledger Entry in Chapter 14.

5. Simplify: Reduce Work to the Required Outcome

Security work expands when uncertainty is answered with more documentation, meetings, participants, and approvals. Simplification asks what must be done to produce the necessary result.

The objective is not lower quality. It is less machinery between demand and outcome.

Convert Deliverables into Decisions

Many requests arrive in the form of a deliverable:

- “Prepare a risk assessment.”
- “Write a policy.”
- “Build a dashboard.”
- “Review the architecture.”
- “Create a board presentation.”

Translate the request:

- What decision will this enable?
- Who will make that decision?
- What information is necessary for that decision?
- What is the smallest credible artifact?
- What evidence must remain after the decision?

A twenty-page assessment may become a one-page decision record with supporting evidence linked separately. A weekly dashboard may become an alert when tolerance is exceeded. A broad architecture review may become five risk-trigger questions.

The One-Page Risk Decision

Use this structure:

1. **Decision:** What must be decided, by whom, and by when?
2. **Context:** What changed or is proposed?
3. **Exposure:** What credible scenario matters?
4. **Options:** What choices are available?
5. **Recommendation:** What does the accountable team recommend and why?
6. **Residual risk:** What remains after the recommendation?
7. **Evidence:** What supports the conclusion?
8. **Review trigger:** What change would cause reconsideration?

Standardize the Common; Investigate the Novel

Create reusable patterns for repeated work:

- Approved architecture patterns
- Standard contract security clauses
- Evidence request libraries
- Incident severity criteria
- Vendor risk tiers
- Vulnerability treatment rules
- Common customer questionnaire responses
- Role-based access patterns
- Executive metric definitions

Standardization should remove repeated interpretation while preserving a route for exceptions. The leader should see the novel case, not re-decide the common one.

Reduce Participants and Handoffs

Each handoff adds delay, context loss, and coordination. For a recurring workflow, map:

- Who touches it?
- What information is added?
- What decision is made?
- What waits between steps?
- Which step exists only because the prior step is unreliable?

Remove observers, duplicate reviewers, and serial approvals that can safely occur in parallel or under policy.

Simplification Worksheet

Current deliverable or workflow:

Required outcome:

Decision and decision-maker:

Minimum information required:

Evidence that must be retained:

Steps, participants, or sections to remove:

Standard pattern that could replace custom work:

Exception trigger:

Tuesday: Replace a Deliverable with a Decision

The afternoon project review becomes a one-page decision packet. One project matches an approved pattern and exits the queue. The novel project reaches the leader with a trust-boundary decision, three options, a recommendation, and evidence. The meeting shrinks because the decision has become visible.

Do This Now

Rewrite one current deliverable request as a single decision question: who decides what, by when, using which minimum evidence?

Proof It Worked

The intended decision occurs with fewer pages, participants, handoffs, or elapsed days—and the supporting evidence remains traceable.

Use the One-Page Decision Packet in Chapter 14.

6. Automate: Let Predictable Work Run

Automation is the fourth filter for a reason. Automating unnecessary or overcomplicated work makes waste faster and harder to see.

Automate only after the outcome, owner, inputs, rules, and exceptions are understood.

Good Automation Candidates

Look for work that is:

- Repeated at meaningful volume
- Based on stable rules
- Fed by structured or consistently available data
- Time-sensitive
- Prone to manual error
- Valuable but cognitively uninteresting
- Easy to verify
- Reversible when it acts

Examples include:

- Collecting recurring control evidence
- Routing vulnerabilities to system owners
- Enriching alerts with asset and identity context
- Reminding owners before exceptions expire
- Generating standard report sections
- Detecting missing ownership or overdue actions
- Scheduling access reviews and exercises
- Reusing approved questionnaire responses
- Creating tickets when defined thresholds are crossed

Automate the Preparation, Preserve the Decision

The safest first target is preparation:

- Gather facts
- Normalize inputs
- Remove duplicates

- Apply known classifications
- Retrieve prior decisions
- Draft routine text
- Identify missing information
- Assemble a decision packet

Keep humans responsible for high-consequence interpretation, approval, and exception handling.

The Automation Control Card

Before automating, define:

- **Outcome:** What is the automation intended to achieve?
- **Trigger:** What starts it?
- **Inputs:** Which sources does it trust?
- **Rule:** How is the action determined?
- **Owner:** Who is responsible for correct operation?
- **Evidence:** What record does it produce?
- **Exception:** What causes human review?
- **Failure mode:** How could it fail silently or dangerously?
- **Rollback:** How is the action reversed?
- **Review:** How often is performance checked?

Automation Risks

Do not automate merely because a tool can perform the action. Pay special attention to:

- Incorrect or stale source data
- Hidden bias in scoring or prioritization
- Excessive privileges granted to integrations
- Sensitive data disclosure
- Unreviewed changes to rules or models
- Alert floods that recreate manual work
- Irreversible containment or deletion
- Automatic closure without evidence
- Lack of an accountable owner

For incident response, begin with enrichment, evidence preservation, notification, and reversible containment. Keep high-impact actions within explicit authority and human confirmation unless the operating context clearly supports another design.

Measure Automation by Attention Removed

The success measure is not how many workflows exist. Ask:

- Did the leader stop touching routine cases?
- Are exceptions accurate and meaningful?
- Has cycle time improved?
- Is evidence more reliable?
- Did the automation create new monitoring work?
- Can the process operate when its designer is unavailable?

Tuesday: Automate the Preparation

Executive metrics arrive already calculated and checked against thresholds. The leader spends the reporting block interpreting two material changes and drafting a recommendation, not copying numbers between formats. Automation prepares the judgment; it does not impersonate it.

Do This Now

Choose one repeated preparation step with stable inputs and an obvious error signal. Automate or template only that step, then run it beside the current method long enough to compare results.

Proof It Worked

Measure attention removed after review, not workflow runs. Evidence should show fewer manual touches, acceptable exception quality, reliable rollback, and no new monitoring burden larger than the work removed.

Use the Automation Candidate Card in Chapter 14.

7. Maintain: Protect the Work Only Leadership Can Do

Maintain is the last resort, not a failure. Some work should remain with the security leader because the consequence, authority, ambiguity, or relationship is inseparable from the role.

Work Worth Maintaining

Security leaders should usually protect time for:

- Risk appetite and material risk acceptance
- Program direction and investment choices
- Executive and board communication
- Serious incident leadership
- Regulatory and critical customer relationships
- Organizational conflict involving ownership or priorities
- Talent decisions, coaching, and succession
- Novel risks without established decision patterns
- Decisions that are difficult to reverse
- Independent challenge when incentives favor unsafe behavior

Calendar the Work You Claim Matters

If strategic thinking, coaching, architecture direction, and relationship building matter, they must appear on the calendar before routine demand fills the available space.

Use protected blocks for:

- Focused program work
- Decision windows
- Team coaching
- Stakeholder relationships
- Improvement work

Batch routine approvals and consultations. Use office hours for non-urgent questions. Do not leave your highest-value work dependent on finding leftover time.

Maintain Does Not Mean Forever

For every maintained activity, record why it remains personal and what would allow it to move:

- A team member develops the necessary judgment

- Policy defines the decision boundary
- A standard pattern becomes available
- Better data reduces uncertainty
- An integration removes preparation work
- The risk or organizational context changes

Review these activities monthly. The Maintain category should remain deliberately small.

Leader-Owned Work Card

Outcome:

Why only I can own it now: Authority / judgment / relationship / consequence

Protected time:

Information and people required before the work begins:

Definition of done:

Transfer condition and review date:

Tuesday: Protect the Work Only Leadership Can Do

The risk memo remains with the leader, but it no longer begins after dinner. Noise has been removed, preparation delegated and automated, and the decision packet reduced. A protected afternoon block now holds the consequential judgment.

Do This Now

Block one hour this week for a named leader-only outcome. Name the evidence required and the condition that would let the work move later.

Proof It Worked

The block happens, the decision advances, and routine work does not reclaim the time; the Maintain set stays small.

8. Ten Information Security Workload Playbooks

The following playbooks show how EDSAM changes common sources of leadership overload.

1. Inbox and Messaging

Eliminate: Unsubscribe from low-value alerts and reports. Leave channels that do not require your presence. Stop using the inbox as a task list.

Delegate: Route operational requests to a visible team intake. Give an operations lead authority to resolve routine questions.

Simplify: Require messages to state context, recommendation, decision, and deadline. Use short response templates for common requests.

Automate: Filter system notifications, create tasks from marked messages, and surface only threshold-based alerts.

Maintain: Sensitive conversations, executive requests, material risk decisions, and relationship work.

2. Calendar and Meetings

Eliminate: Status-only meetings, duplicate governance forums, and attendance without a defined role.

Delegate: Send the accountable owner with decision authority. Rotate representation where continuity allows.

Simplify: Shorten meetings, publish pre-reads, attend only the decision segment, and end when the outcome is reached.

Automate: Scheduling, agenda collection, transcription, action capture, and reminders.

Maintain: Decisions, conflict resolution, coaching, crisis coordination, and high-value stakeholder conversations.

3. Vulnerability and Exposure Management

Eliminate: Duplicates, dead assets, informational findings without a plausible scenario, and leader review of individual routine findings.

Delegate: System owners remediate. Risk owners decide treatment within delegated tolerance.

Simplify: Prioritize using exploitability, exposure, business criticality, threat activity, and compensating controls, not severity alone.

Automate: Enrichment, ownership lookup, ticketing, deadlines, retesting, evidence, and escalation.

Maintain: Systemic exposure, chronic ownership failure, material exceptions, and investment tradeoffs.

4. Vendor Security Reviews

Eliminate: Reviews for vendors below a documented risk threshold and repeated requests for evidence already available and current.

Delegate: Business sponsors own justification; procurement and legal own their decisions; analysts perform evidence review.

Simplify: Use tiers based on criticality, data, access, substitutability, concentration, and operational dependency.

Automate: Intake, tiering, evidence expiry, monitoring, reassessment triggers, and offboarding reminders.

Maintain: Critical supplier risk, unresolved exceptions, concentration risk, and risk acceptance.

5. Architecture and Project Reviews

Eliminate: Security attendance in every project meeting and reviews of changes fully covered by approved patterns.

Delegate: Engineers and architects apply security standards; project owners provide required context.

Simplify: Trigger review based on sensitive data, privileged authority, external exposure, novel trust boundaries, critical dependencies, or irreversible design choices.

Automate: Policy checks, code and configuration scanning, evidence collection, and standard recommendations.

Maintain: Novel designs, major trust decisions, material exceptions, and cross-enterprise tradeoffs.

6. Audit and Compliance

Eliminate: Duplicate evidence requests and controls created solely to satisfy different framework wording.

Delegate: Control owners operate controls and produce evidence. Compliance coordinates requests and mappings.

Simplify: Use one common control library and one authoritative evidence package for each control.

Automate: Evidence collection, expiry, sampling, reminders, cross-framework mapping, and readiness reporting.

Maintain: Control design judgments, contradictory evidence, material deficiencies, and executive commitments.

7. Policies, Standards, and Exceptions

Eliminate: Policies that repeat other policies, have no accountable owner, or do not influence behavior.

Delegate: Subject-matter owners maintain standards and procedures within the policy framework.

Simplify: State principles, responsibilities, mandatory outcomes, and exception rules clearly. Move volatile technical details into standards.

Automate: Review schedules, approval routing, publication, attestations, and exception expiry.

Maintain: Risk appetite, major policy choices, and exceptions above delegated tolerance.

8. Incidents and Escalations

Eliminate: Unactionable detections, unnecessary executive notifications, and leadership attendance in routine investigations.

Delegate: Analysts investigate; incident commanders coordinate; designated owners communicate with their stakeholder groups.

Simplify: Define declaration criteria, severity, roles, decision authority, communication triggers, and a common operating record.

Automate: Enrichment, evidence preservation, stakeholder lookup, task creation, timestamps, and reversible containment.

Maintain: Material business decisions, legal and public communication, destructive actions, major containment tradeoffs, and restoration of trust.

9. Metrics and Executive Reporting

Eliminate: Measures without an audience, threshold, trend, owner, or decision.

Delegate: Metric owners validate source data and explain exceptions.

Simplify: Report material change, exposure, action, ownership, and decision, not every available count.

Automate: Data collection, trend calculation, visual generation, and first-draft commentary.

Maintain: Interpretation, candor, recommendation, and the executive conversation.

10. Team Leadership

Eliminate: Routine approval dependency and meetings that substitute for clear ownership.

Delegate: Give managers outcomes and decision rights, not task lists alone.

Simplify: Use a small number of program outcomes, clear role charters, and exception-based updates.

Automate: Administrative reminders, recurring data collection, and one-on-one preparation. **Maintain:** Coaching, feedback, hiring, succession, standards of behavior, and difficult personnel decisions.

A Complete Walkthrough: The Critical SaaS Launch

The short playbooks above show the shape of EDSAM. The following illustrative case shows the full operating system in motion, including the artifacts a leader would actually use.

The demand as it arrived

At 9:10 Monday morning, a business sponsor sends this request:

> We need security approval today for a SaaS platform scheduled to launch Friday. Procurement says the contract is almost done. The vendor sent a questionnaire and several documents. Can you join a call at 11:00, complete the review, and tell us whether we can proceed?

Without an operating system, the security leader is likely to accept the deadline, join the call, read every document, chase missing answers, negotiate contract language, and become the apparent owner of the launch decision. The request is legitimate. Its proposed ownership and form are not.

1. Capture and classify

The leader records the request before accepting the work:

- **Required outcome:** A risk-informed decision on whether and under what conditions the organization may use the service.
- **Real deadline:** Wednesday at 3:00 p.m., when the sponsor must decide whether to preserve the Friday launch. “Today” was a preference, not an external obligation.
- **Material context:** The service will process confidential customer information, use single sign-on, and support a revenue-producing workflow.
- **MAT:** Money Medium; Attention High; Time High if handled as requested.
- **Eisenhower position:** Important and urgent, but not automatically leader-executed.
- **Decision owners:** The business sponsor owns whether to proceed; security owns its recommendation and any security exception within delegated authority.

The item enters **Decide**, while evidence preparation enters **Delegate**. It does not become an all-day personal project.

2. Apply all five moves

Eliminate. Remove the generic introductory call, the duplicate questionnaire sections already answered by current evidence, and review of controls irrelevant to the data and integration in scope. Decline the fiction that security alone approves the business launch.

Delegate. A security analyst validates the evidence and identifies contradictions. The identity owner confirms single sign-on and provisioning. Procurement and legal retain the contract and commercial obligations. The business sponsor confirms data, use, criticality, and acceptable launch alternatives.

Simplify. Reduce the review to six decision questions:

1. What data and business process are exposed?
2. Which identities, privileges, and integrations create access?
3. What credible failure or abuse scenarios could materially affect the organization?
4. Which controls are demonstrated by current evidence, and which are only asserted?
5. Which gaps can be bounded by contract, configuration, monitoring, or launch conditions?
6. Who has authority to accept the residual risk by Wednesday?

Automate. Retrieve approved vendor records, compare received artifacts with the evidence request list, identify expirations and missing fields, and draft a cited gap summary. The automation may prepare work; it may not mark a control verified, send the recommendation, approve the vendor, or accept risk.

Maintain. The leader retains the material security recommendation, any exception above delegated tolerance, the executive conversation if the gap is significant, and the accuracy of security's final representation.

3. Complete the delegation brief

Outcome: Deliver a cited evidence summary and recommended conditions sufficient for the Wednesday decision.

Accountable owner: Security analyst.

Authority granted: Close routine evidence requests when current authoritative evidence satisfies the approved criteria; request clarification directly from the vendor; involve identity and data owners.

Constraints: Do not accept risk, waive required controls, characterize untested controls as operating, or make external commitments.

Definition of done: The six decision questions are answered; assertions and verified evidence are distinguished; open gaps have owners, options, and time sensitivity.

Escalate when: The vendor will handle regulated or unusually sensitive data; evidence conflicts; privileged access cannot be bounded; a required control is absent; or the sponsor asks to proceed outside delegated tolerance.

Update cadence: One decision-ready packet by Tuesday at 3:00 p.m.; immediate escalation only for a listed condition.

4. Produce the one-page decision packet

Decision required: May the organization launch Friday, and under which conditions?

Decision owner: Business sponsor, with security and legal recommendations.

Material exposure: Confidential customer information and a revenue workflow would depend on a supplier whose administrative-access evidence is incomplete.

Options:

1. Launch Friday with full scope.
2. Launch a limited pilot without confidential customer data while evidence and controls are completed.
3. Delay launch until the material evidence gap is resolved.

Recommendation: Use the limited pilot. Enforce single sign-on, restrict administrative access, prohibit confidential customer data, enable available logging, and require the missing evidence before expansion.

Residual risk: The pilot still depends on vendor availability and operational integrity, but the most consequential data exposure is withheld and administrative access is bounded.

Evidence and confidence: Current architecture and identity documentation; contract draft; vendor responses; identity-owner confirmation. Confidence is moderate because production administrative-access evidence remains incomplete.

Consequence of delay: A full launch delay may affect the sponsor's schedule. Proceeding at full scope would accept a material uncertainty without sufficient evidence.

Review trigger: Receipt of the missing evidence, expansion to confidential data, a material integration change, or thirty days after pilot start.

5. Record the result

The business sponsor selects the limited pilot. The analyst records the conditions and review trigger in the decision ledger. Procurement records the contractual commitment. The identity owner implements the approved access pattern. The leader holds one twenty-minute decision conversation rather than becoming the production team for the entire review.

For an illustrative before-and-after comparison:

- Leader effort falls from three to four fragmented hours to approximately seventy focused minutes.
- Two broad meetings become one prepared decision conversation.
- Routine evidence moves to an authorized owner while material uncertainty remains visible.
- The business owns the decision, and no required security outcome silently disappears.

The numbers are illustrative, not promises. The proof is the sequence: test the clock, correct ownership, reduce the work, bound the preparation, and retain consequential judgment. The short templates exist to move that decision with enough authority and evidence—not to create another reporting burden.

Copy the completed pattern with the EDSAM Intake Card, One-Page Decision Packet, and Decision Ledger Entry.

9. The EDSAM Cadence

EDSAM creates value through repetition. Small improvements compound as recurring work changes or disappears.

Daily: Ten Minutes

1. Process new demands through the five filters.
2. Identify the one outcome requiring your best attention.
3. Assign owners and escalation conditions.
4. Move non-urgent decisions into a decision window.
5. Confirm that active material issues have an owner and next action.

At the end of the day, capture commitments, close stale items, and protect tomorrow's first focus block.

Weekly: Thirty Minutes

Review your calendar, task system, and delegated queue:

- What should stop?
- What am I doing that another owner should control?
- What consumed disproportionate attention?
- Which deliverable can be smaller?
- Which repeated action is ready for automation?
- What reached me that should have been resolved below my level?
- Which temporary responsibility has become permanent?
- What important work received no protected time?

Choose at least one concrete improvement. Do not use the review merely to reorganize the same work.

Monthly: Sixty Minutes

- Compare planned and actual time.
- Review maintained work for transfer opportunities.
- Audit recurring meetings and reports.
- Examine overdue delegated outcomes and escalation quality.
- Review automation exceptions and failure modes.
- Identify systemic work entering through the wrong owner.
- Rebalance Focus work against enterprise priorities.

Quarterly: Reset the System

- Revisit role expectations with your executive sponsor.
- Confirm risk and decision authorities.
- Review the program's major recurring obligations.
- Remove obsolete governance mechanisms.
- Identify skills or roles needed to expand delegation.
- Select one workflow for substantial redesign.

Use the Cadence With Your Team

Once the personal review is stable, add a short team-level EDSAM question to an existing operating meeting:

- What work entered the team that should not belong to security?
- What recurring request should become a standard or self-service path?
- Which escalation lacked a decision owner or threshold?
- Where did a team member wait unnecessarily for leader approval?
- Which activity consumed attention without changing risk?

Choose one system correction. Avoid creating a separate "EDSAM meeting"; improve the cadence already in place.

10. The 30-Day EDSAM Reset

The reset is designed to produce visible relief while protecting security outcomes.

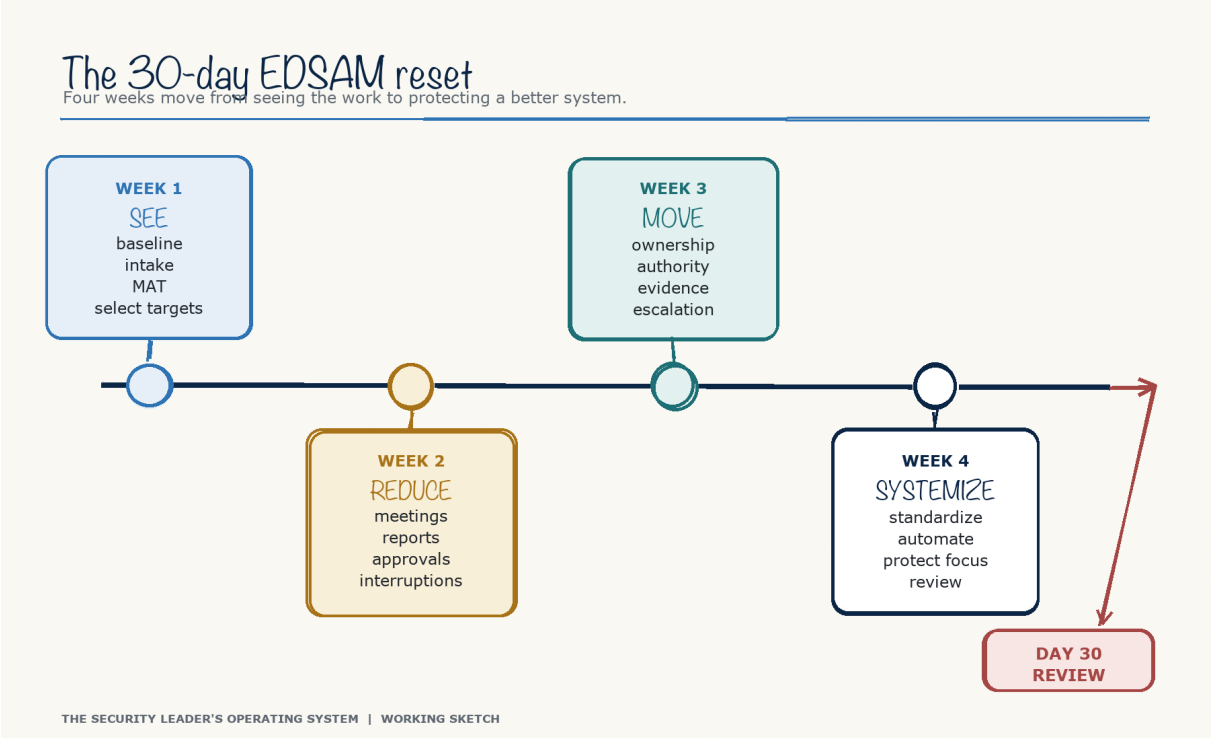

Figure 6. The 30-day reset progresses from seeing the work to reducing demand, moving ownership, and protecting the redesigned system.

Week 1: See the Work

Day 1: Establish a baseline

Estimate the last two weeks:

- Hours in meetings
- Hours producing or reformatting reports
- Hours reviewing routine work
- Hours handling interruptions
- Hours on strategic program outcomes
- Hours coaching or developing people
- Number of active personal commitments

Day 2: Build one intake

Collect open commitments from email, chat, notes, and memory into one trusted location.

Day 3: Rate MAT

Rate the major recurring activities by Money, Attention, and Time. Mark high-attention activities.

Day 4: Identify leader-only work

Apply the personal ownership test. Mark work that genuinely requires authority, judgment, relationship, consequence, or independent challenge.

Day 5: Select targets

Choose:

- One activity to eliminate
- One outcome to delegate
- One workflow to simplify
- One repeated step to automate
- One leader-owned outcome to protect

Week 2: Reduce Demand

- Cancel or redesign one recurring meeting.
- Stop, reduce, or merge one report.
- Remove yourself from one routine approval path.
- Publish interruption and escalation rules.
- Set a work-in-progress limit for personal Focus work.

Communicate the changes as improvements to ownership and decision quality, not as withdrawal.

Week 3: Move Ownership

- Write delegation briefs for selected outcomes.
- Confirm authority and capacity with each owner.
- Define evidence and escalation conditions.
- Replace status chasing with milestone or exception updates.
- Coach the first decisions rather than taking them back.

Week 4: Systemize and Protect

- Standardize one common decision or deliverable.
- Automate one preparation or routing step.
- Pilot one Tier 0 or Tier 1 AI use case with explicit sources, review, evidence, and a shutdown path.
- Create protected focus and decision windows.
- Complete the weekly EDSAM review.
- Compare the new workload with the baseline.

Day 30 Review

Record:

- Hours reclaimed or redirected
- Meetings removed or shortened
- Personal commitments reduced
- Decisions delegated
- Recurring steps automated
- Review time and attention removed after AI-assisted work
- Assistants, permissions, integrations, or scheduled runs retired
- Interruptions reduced
- Security outcomes preserved or improved
- Problems or risks created by the changes
- Next workflow to redesign

The purpose is not an impressive number. It is evidence that the operating system is changing.

11. Measures That Matter

Measure whether EDSAM is improving leadership capacity and program outcomes together.

Begin with a small scorecard you can maintain for four weeks. Do not create a measurement program that consumes the capacity you are trying to recover. Use estimates when precise data would cost more than the decision requires, and keep the definition stable long enough to see direction.

Capacity measures

- Percentage of time spent on leader-only work
- Number of significant personal work items in progress
- Hours of protected focus completed
- Recurring meetings and reports eliminated
- Number of routine decisions moved to delegated owners
- Interruptions that met versus failed escalation criteria
- High-attention activities removed or redesigned

Flow measures

- Time from request to accountable owner
- Time from escalation to decision
- Age of material risks awaiting business action
- Percentage of updates delivered by exception
- Percentage of recurring evidence gathered automatically
- Rework caused by unclear outcomes or ownership

Outcome guardrails

- Material incidents or control failures associated with a removed step
- Overdue high-risk remediation
- Expired risk acceptances
- Missed legal, contractual, or regulatory obligations
- Stakeholder confidence and decision quality
- Team capability and succession depth

Efficiency without outcome guardrails can hide risk. Guardrails without capacity measures can preserve needless work. Use both.

The Four-Week Capacity Scorecard

Record a baseline, then update the same measures once each week:

- **Leader-only work:** Percentage of working time spent on outcomes requiring your authority, judgment, relationship, or accountability
- **Personal WIP:** Number of significant outcomes you are personally carrying
- **After-hours work:** Hours spent outside your intended working boundary
- **Recurring claims:** Hours committed to recurring meetings and reports
- **Unprepared escalations:** Issues arriving without context, options, recommendation, owner, or decision deadline
- **Delegated decisions:** Decisions completed inside delegated authority without avoidable leader intervention
- **Decision latency:** Median or typical time from a decision-ready request to an accountable decision
- **Attention reclaimed:** Hours of leader involvement removed or redirected after review and maintenance costs
- **Outcome guardrails:** Material incidents, missed obligations, control failures, or stakeholder harm plausibly associated with a changed step

Interpret the measures together. A reduction in personal WIP is not success if material risks become ownerless. More delegated decisions are not success if people act outside authority. More automation is not success if after-hours supervision increases. The strongest signal is a sustained increase in leader-only work while outcome guardrails remain stable or improve.

When EDSAM Is Not Enough

EDSAM removes avoidable demand and corrects ownership. It cannot manufacture competent capacity for irreducible work. If required outcomes still exceed sustainable capacity after the five moves have been applied, the operating system has exposed a resource or service-level decision.

Do not answer that condition with hidden overtime, indefinite deferral, or untested controls. Prepare a capacity decision packet:

1. **Required outcomes:** Name the security, business, legal, contractual, and risk outcomes that must be produced.
2. **Sustainable capacity:** Show the available people, skills, services, and leader attention after leave, operations, and normal variability.
3. **Irreducible gap:** Identify outcomes that remain uncovered, delayed, fragile, or dependent on a single person after elimination, delegation, simplification, and appropriate automation.
4. **Consequence:** Describe the credible business effect of the gap, including slower service, accumulated exposure, missed assurance, or reduced resilience.

5. **Options:** Add or develop staff, acquire a service, transfer ownership, reduce scope, change a deadline or service level, accept defined risk, or combine those choices.
6. **Decision:** Name the authorized executive, the date by which a choice is needed, and the consequence of no decision.

Use ranges and observable workload evidence rather than claims of being busy. Show which work was removed, which ownership was corrected, and why the remainder cannot safely disappear. This makes the request harder to dismiss as a productivity problem.

The leader's obligation is not to prove that more resources are always necessary. It is to make the tradeoff visible. An organization may choose less coverage, slower service, transferred risk, or additional capacity. It should not receive an apparently complete security program by consuming unrecorded nights, weekends, and cognitive reserve.

Use the Four-Week Capacity Scorecard and Capacity Decision Packet in Chapter 14.

12. Failure Modes and Corrections

“Everything is important.”

Correction: Require an explicit consequence, owner, and displacement decision. Importance without tradeoff is aspiration, not prioritization.

“It is faster if I do it myself.”

Correction: Compare today’s task time with the lifetime cost of remaining the only capable person. Use coaching and staged delegation.

“The team keeps escalating everything.”

Correction: Clarify decision rights, tolerance, examples, and escalation conditions. Review whether prior reactions punished reasonable independent judgment.

“Automation created more alerts.”

Correction: Measure precision and attention removed. Tune or remove workflows whose exceptions are not actionable.

“The agent is faster, but I still check everything.”

Correction: Count review and trust overhead as work. Narrow the task, improve evidence and validation, sample low-risk outputs, or return to assisted execution if dependable review cannot become smaller.

“The AI answer sounded complete.”

Correction: Ask which sources, dates, systems, stakeholders, and constraints were unavailable. Treat fluency as presentation quality, not evidence of complete context.

“We simplified the evidence and upset the auditor.”

Correction: Separate the control outcome from the evidence format. Confirm requirements and preserve traceability before removing evidence.

“The business will not own the risk.”

Correction: Present the decision in business terms, identify the accountable executive, document the consequence, and escalate through established governance. Security should not silently accept ownership by default.

“My calendar filled again.”

Correction: Recurring demand expands into unprotected capacity. Reserve leader-only work first, publish office and decision windows, and repeat meeting elimination monthly.

“Maintain keeps growing.”

Correction: Require every maintained activity to state why it remains personal, what would enable transfer, and when it will be reviewed.

“EDSAM proves we do not need more people.”

Correction: EDSAM removes avoidable work and exposes the irreducible workload. Compare required outcomes and service expectations with sustainable capacity. If a gap remains, present it as a risk and operating-model decision. Never disguise unpaid hours, chronic interruption, or untested controls as efficiency.

“We eliminated something and a dependency broke.”

Correction: Restore the minimum safe mechanism, identify the missed dependency, and update the elimination test. Favor reversible trials for activities with uncertain consumers or consequences.

Language That Protects Capacity

Boundaries are operating controls, but leaders often hesitate because a technically correct refusal can damage trust. The following scripts keep the outcome visible while changing ownership, timing, or form. Adapt the words to your relationships and authority.

Decline a meeting without withdrawing support

> I do not think my attendance is necessary for the full meeting. Please send the decision, options, recommendation, and deadline. I will review them during today’s decision window and join if the discussion reaches the stated escalation condition.

Return business risk to the business owner

> Security will provide the exposure, options, recommended controls, and residual risk. The decision to operate the service under that residual risk belongs to the accountable business owner. I will record and support the decision; I should not make it on the business’s behalf.

Delegate an outcome with real authority

> I am asking you to own this outcome, not just prepare it for me to redo. You may decide routine cases inside these boundaries. Bring me exceptions that meet these conditions, along with your recommendation and the consequence of delay.

Challenge false urgency

> I understand that this is important. What creates the deadline, who set it, and what consequence occurs at that time? Once we know that, we can decide whether this interrupts current work or enters the normal decision queue.

Require a decision-ready escalation

> Please return with the business context, current exposure, actions already taken, available options, your recommendation, the decision owner, and the latest useful decision time. If the situation is changing faster than that packet can be prepared, tell me which escalation condition has been met.

Protect a focus block

> I am unavailable during this block because I am completing the risk decision only my role can make. Interrupt me for the published material-impact conditions. I will process all other requests in the next decision window.

Preserve a control that appears inefficient

> I support reducing the effort, but we should separate the current mechanism from the outcome it protects. Before we remove it, let us identify the obligation, owner, evidence, dependencies, and restoration trigger. We can simplify the method without silently deleting the control outcome.

Present a capacity gap without asking for sympathy

> We have applied elimination, corrected ownership, simplified deliverables, and automated the stable preparation work. These required outcomes still exceed sustainable capacity. Here is the uncovered work, its consequence, and the available choices: add capacity, acquire a service, reduce scope, change the service level, or accept the defined risk. I need an authorized decision by this date.

These are not magic phrases. Their value comes from consistency between the words and the operating system behind them. A leader who says “exceptions only” but continues answering every routine question teaches the organization to ignore the boundary.

13. AI-Augmented Security Leadership

Modern AI extends the reach of the operating system. An LLM can reduce cognitive setup. A configured assistant or skill can preserve a repeatable method. A bounded agent can gather evidence or prepare work across connected systems. None of those capabilities changes the governing rule:

> Apply EDSAM to the work before applying AI to the work.

Do not preserve a bad request with a better prompt. Do not automate a report no one uses. Do not give an agent authority because a demonstration looked fluent. Eliminate, delegate, and simplify first. Then select the least powerful capability that reliably removes attention while preserving evidence and accountable judgment.

A Working Vocabulary

Product names will change. These operating distinctions are more durable:

- **LLM or copilot:** Helps a person think, compare, summarize, classify, or draft during a live interaction. The person decides what leaves the session.
- **Configured assistant or GPT:** Combines standing instructions, selected knowledge, examples, and perhaps tools for a recurring job.
- **Skill:** A versioned package of instructions, templates, examples, and optional scripts that teaches an agent a defined workflow. A skill is executable institutional memory.
- **Workflow automation:** Deterministic rules move information or trigger known steps.
- **AI agent:** A model interprets a goal, chooses among tools or actions, and can create consequences in connected systems.

These are capability levels, not maturity badges. A careful prompt used by an accountable leader can create more value than an agent that requires another program to supervise.

The Capability Escalation Ladder

Move upward only when a lower level cannot produce the outcome:

1. **Think:** Challenge assumptions or expose missing information.
2. **Draft:** Prepare reviewable text, classifications, or summaries.
3. **Retrieve:** Read approved sources and return citations.
4. **Recommend:** Compare evidence to explicit criteria and propose an action.
5. **Prepare action:** Create a draft ticket, change set, or message that remains pending.

6. **Act with approval:** Execute only after an authorized person reviews the specific action.
7. **Act within a budget:** Perform limited, reversible, observable actions inside explicit thresholds.
8. **Emergency autonomy:** Act at machine speed only during a declared, narrow, logged, and automatically expiring condition.

Most personal security-leadership systems should remain at levels 1 through 5. Levels 6 through 8 are operating-model decisions, not productivity settings.

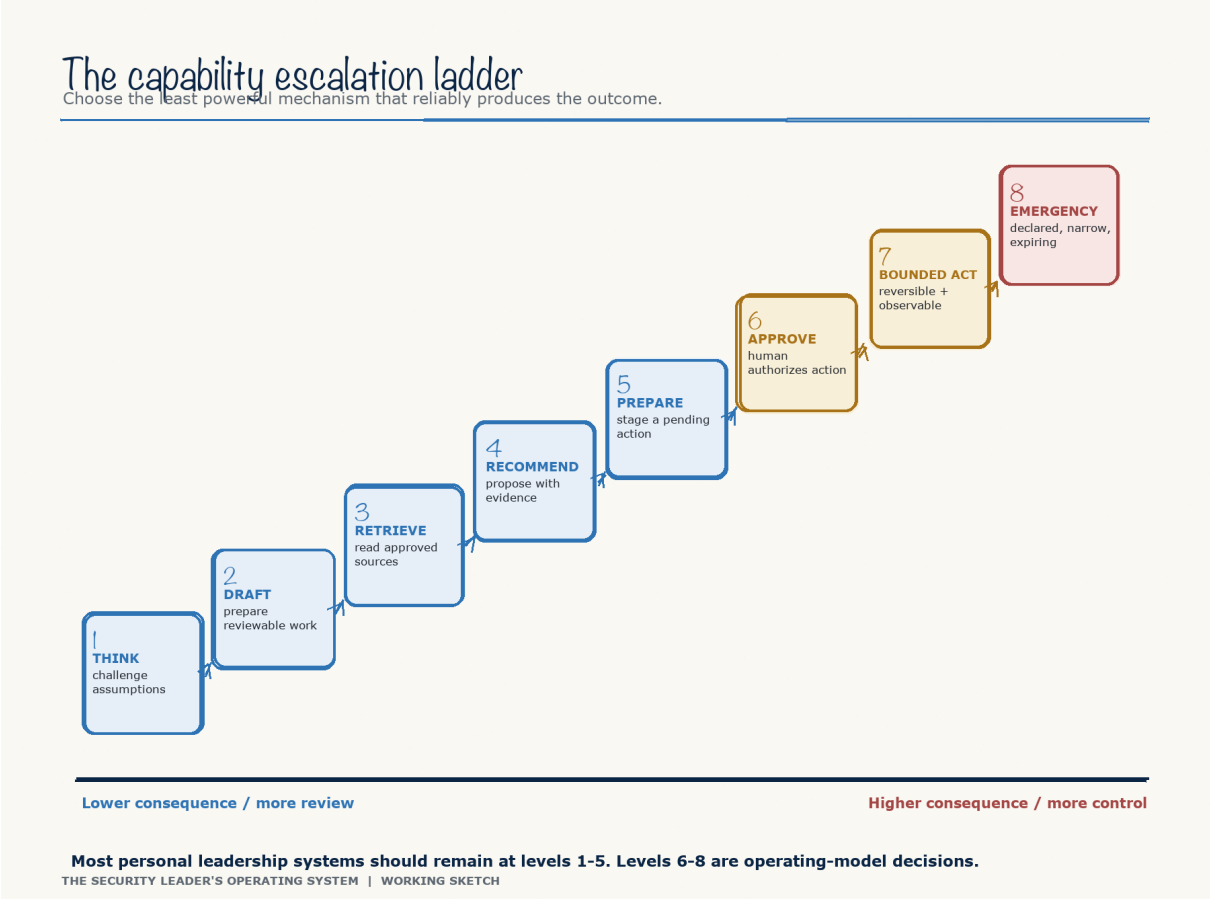

Figure 7. Capability should escalate only when a lower-consequence level cannot produce the required outcome.

Divide the Work Three Ways

Keep accountable authority, values, political judgment, relationship repair, risk acceptance, legal interpretation, executive commitments, and hard-to-reverse decisions in the **human zone**.

Use the **model zone** for language-rich work that can be reviewed: summarize evidence, compare artifacts to a checklist, identify contradictions, translate technical findings, draft questions, and prepare decision packets.

Use the **deterministic zone** where expected behavior can be stated exactly: calculate dates and thresholds, deduplicate records, verify required fields, route approved artifacts, preserve logs, and alert on expiration.

The strongest design is often hybrid. Deterministic systems collect and validate. The model interprets or drafts. The human authorizes consequential action.

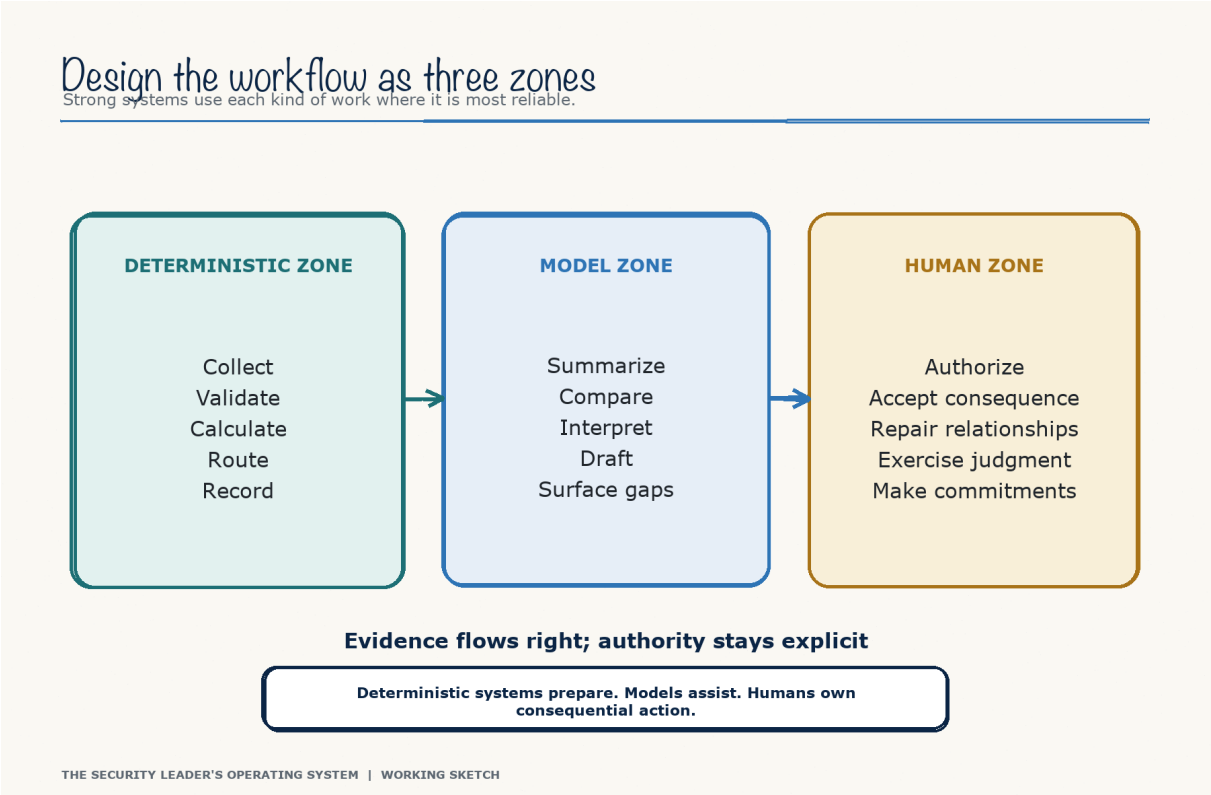

Figure 8. Deterministic systems prepare evidence, models assist with interpretation, and humans retain consequential authority.

Find the Right Work with FRICT

FRICT identifies stronger automation candidates: **Frequent, Rules-based, Information-moving, Checklist-driven, and Templated**. Score time saved, error reduction, repeatability, data feasibility, and risk if wrong:

> FRICT score = Time saved + Error reduction + Repeatability + Data feasibility − Risk if wrong

A strong score is a reason to investigate, not authorization to deploy. EDSAM still asks whether the activity should exist and who should own it. The complete model appears in The FRICT Method.

Price the Automation Tax

Time saved is not value created:

> Expected value = (Time saved × value of time) − error cost − review time − trust overhead − maintenance cost

Review and trust are often transferred work. Maintenance includes broken integrations, model changes, expired credentials, drift, evaluation, and exception handling. Ask: **If this fails quietly, how expensive is it?** The analysis in The Automation Tax We're Not Pricing is a useful companion.

Use SCOPE Before Granting Access

FRICT tests whether work is worth automating. **SCOPE** tests whether a design is safe enough to try:

- **Stakes:** What credible harm follows if the system is wrong, late, incomplete, or manipulated?
- **Context:** What can it see, what can it not see, and how will stale or conflicting information surface?
- **Options:** Can it only draft, or can it read, write, send, trigger, change, delete, spend, or approve?
- **Proof:** What reconstructs sources, actions, approvals, outputs, and final state?
- **Exceptions:** What pauses the workflow, reduces authority, invokes a human, or shuts it down?

If any answer is vague, stay lower on the capability ladder.

Six Systems That Buy Back Attention

- **Read-only security chief of staff:** Read approved work sources, remove noise, group commitments, and draft without sending. Keep source allowlists, freshness, and the boundary between preparation and commitment visible. This is the pattern behind Project Chief.
- **Evidence-gated diligence copilot:** Compare vendor artifacts with an evidence tracker, distinguish assertions from production proof, and prepare cited questions and a one-page decision packet.
- **UAT evidence runner:** Execute an approved browser or API test plan, preserve raw evidence, compare results with acceptance criteria, and leave severity, acceptance, and destructive actions with authorized people.
- **vCISO continuity assistant:** Carry forward open items, remove stale status, identify overdue owners, and prepare the next agenda while preserving client boundaries and positive confirmation before anything is called sent.
- **Research signal filter:** Monitor approved sources, deduplicate and classify by program relevance, disclose uncertainty, and prevent retrieved text from directing privileged tools.
- **Decision-latency packet builder:** Gather the smallest evidence set, show options and time sensitivity, identify the owner, and make assumptions, confidence, missing authority, and consequence of delay explicit.

The Governance Minimum

Every routine AI-enabled workflow needs five controls:

1. A named purpose and accountable owner
2. Explicit sources, permissions, and data boundaries
3. A written capability level and autonomy limit
4. Evidence sufficient to reconstruct outputs and side effects
5. A tested way to pause, revoke, roll back, and retire it

Treat untrusted email, web pages, tickets, documents, and retrieved text as influence, not passive data. Put validation next to any tool that creates a side effect. Test ordinary cases, ambiguous cases, incomplete context, adversarial input, stale data, tool failure, and at least one high-impact case that must stop for approval.

The **AI Agent Governance Checklist** in the back matter expands this minimum. For deeper research on autonomy budgets, control planes, and agent architecture, use the topical links in **Sources and Further Reading**.

The AI-Augmented Weekly Review

Ask:

1. Which repeated task should be eliminated before anyone automates it?
2. Which model output consumed more review time than it saved?
3. Which assistant or skill needs tighter instructions, examples, or source boundaries?
4. Which system gained access, memory, tools, or actionability beyond its original purpose?
5. Which output influenced a decision without sufficient evidence?
6. Which permission, integration, memory store, or scheduled run can now be revoked?

The goal is not a growing fleet of helpers. It is a smaller amount of routine work requiring the leader's mind.

Do This Now

Select one Tier 0 or Tier 1 use case. Write its SCOPE answers, give it approved read-only sources, and test it against five real cases before connecting any write-capable tool.

Proof It Worked

Compare total time before and after review. Record errors, exceptions, evidence quality, permissions, and maintenance effort. Keep the system only if dependable attention removed exceeds the automation tax.

Conclusion: Build Less Dependency

Return to the security leader whose Tuesday opened this guide.

Before EDSAM, every legitimate concern became a claim on the leader's time. The customer questionnaire arrived as a document to produce. The vulnerability program arrived as a meeting to attend. Project risk arrived as two broad reviews. Audit evidence arrived as a hunt. Chat delivered raw problems. Executive reporting consumed the afternoon. The decision that truly required leadership began after dinner.

The redesigned Tuesday is not empty, and it is not easy. Security leadership still carries uncertainty, consequence, conflict, and responsibility. What changed is the path by which work earns access to the leader.

The unused questionnaire sections are eliminated. A capable owner prepares routine answers. Approved architecture patterns remove predictable reviews. Evidence comes from the control owner and authoritative repository. Rules route normal cases without ceremony. Decision packets replace sprawling deliverables. Protected time holds the risk judgment, relationship, coaching conversation, or commitment that cannot safely move elsewhere.

The visible result is reclaimed time. The deeper result is less dependency.

That distinction matters. Activity can be accelerated without making an organization stronger. A leader can answer faster, attend more, approve more, and still remain the constraint through which the entire program must pass. EDSAM does not ask how to increase that throughput. It asks how to stop building the program around a heroic person.

Eliminate work that no longer protects an outcome. Delegate outcomes with authority and escalation rules. Simplify the surviving work to the decision or evidence that matters. Automate predictable preparation and routing with visible controls. Maintain only the judgment, consequence, and relationships that belong with leadership now—and keep asking what would allow that set to shrink.

Use AI the same way. Let models absorb setup, comparison, and drafting where review is meaningful. Let deterministic systems carry rules and records. Let agents act only inside deliberate authority. Keep the human accountable for material decisions. Count review, doubt, maintenance, and failure as real costs. A system that saves keystrokes while creating permanent supervision is not leverage.

Do one thing when you close this guide: choose the recurring demand that consumes the most attention without requiring your authority. Run it through all five moves. Change the system before you reorganize the task. Then record what happened—what stopped, who owns the outcome, what became smaller, what runs predictably, what remains yours, and which security result proves the change was safe.

Return the next week and do it again.

Over time, those small corrections become an operating system. Meetings no longer renew themselves. Decisions move closer to the work. Common cases use known patterns. Exceptions arrive prepared. Important non-urgent work receives daylight. The team grows more capable because the leader is no longer protecting capability by doing everything personally.

That is also how a legacy of knowledge is built. It is not only the advice we publish or the lessons we can recite. It is the methods, decision boundaries, templates, tools, and examples that let another person act well without our constant presence. Good leadership leaves behind more judgment in the system and less dependency on the leader.

Get back to that work.

14. Field Templates

Copy these templates into the tool you already use. Do not introduce a new platform unless the current one cannot support the operating rules.

EDSAM Intake Card

Request or trigger:

Required outcome:

Stakeholder:

Deadline and source:

Risk or obligation:

MAT rating: Money H/M/L; Attention H/M/L; Time H/M/L

Does this require leader authority or judgment?

Eliminate: What happens if we do nothing?

Delegate: Who is closest to the outcome?

Simplify: What is the minimum useful result?

Automate: Which predictable steps can run?

Maintain: Why must the remaining work stay with me?

Disposition and next action:

One-Page Decision Packet

Decision required:

Decision owner:

Deadline:

Context:

Material exposure or opportunity:

Options:

Recommendation:

Residual risk:

Evidence and confidence:

Consequence of delay:

Review trigger:

Weekly EDSAM Review

Eliminate: What stopped, should stop, or should occur less often?

Delegate: What am I still holding that belongs elsewhere?

Simplify: Which deliverable or workflow can become smaller?

Automate: Which repeated preparation or routing step is ready?

Maintain: Which leader-only outcome receives protected time next week?

Attention audit: What consumed the most cognitive or emotional capacity?

System lesson: What reached me that should not reach me next time?

One improvement commitment:

Four-Week Capacity Scorecard

Use one row definition for all four weeks. A direction is useful only when the measurement stays comparable.

Measure	Baseline	Week 1	Week 2	Week 3	Week 4
Leader-only work (%)					
Personal WIP (count)					
After-hours work (hours)					
Recurring claims (hours)					
Unprepared escalations (count)					
Delegated decisions (count)					
Decision latency					
Attention reclaimed (hours)					
Outcome guardrail failures					

Direction or lesson to carry forward: _____

Automation Candidate Card

Recurring activity:

Frequency and volume:

Current Money / Attention / Time:

Simplified outcome:

Stable rule:

Trusted inputs:

Human decision retained:

Exception and escalation:

Evidence produced:

Failure mode and rollback:

Owner and review date:

AI Use-Case Card

Leadership outcome:

Current workflow and burden:

EDSAM disposition: Eliminate / Delegate / Simplify / Automate / Maintain

FRICT evidence: Frequent / Rules-based / Information-moving / Checklist-driven / Templated

Capability level: Think / Draft / Retrieve / Recommend / Prepare / Act with approval / Act within budget

Human zone:

Model zone:

Deterministic zone:

SCOPE — Stakes:

SCOPE — Context available and missing:

SCOPE — Options and side effects:

SCOPE — Proof required:

SCOPE — Exceptions and stop conditions:

Expected value after review and maintenance:

Pilot scope and success measure:

Agent Authority Ledger

Agent, GPT, skill, or automation name:

Purpose and accountable owner:

Model/provider, runtime boundary, instructions, and skill version:

Data and knowledge sources:

May read:

May write:

May trigger:

May share or send:

Memory type, retention, and deletion:

Autonomy tier and numeric limits:

Actions requiring approval:

Evidence and logs retained:

Known failure and abuse paths:

Kill switch, rollback, last test, and last review:

Expiration or reauthorization date:

Decision Ledger Entry

Decision and date:

Decision owner:

Context:

Credible risk scenario:

Options considered:

Decision and rationale:

Residual risk:

Evidence and confidence:

Commitments, owners, and dates:

Review or expiration trigger:

Capacity Decision Packet

Decision required:

Decision owner and date:

Required outcomes:

Sustainable people, skill, service, and leadership capacity:

Work already eliminated, delegated, simplified, or automated:

Irreducible uncovered, delayed, or fragile outcomes:

Credible consequence of the gap:

Option 1 — Add or develop capacity:

Option 2 — Acquire or transfer a service:

Option 3 — Reduce scope or change the service level:

Option 4 — Accept defined delay or risk:

Recommendation and rationale:

Consequence of no decision:

Evidence and confidence:

Review trigger:

Appendix: NIST CSF 2.0 Scope Check

EDSAM changes how leadership work is performed. Use NIST Cybersecurity Framework 2.0 to check that efficiency improvements preserve the program's intended outcomes.

Use the six Functions during quarterly review:

- **Govern:** Are strategy, policy, roles, oversight, context, and supply-chain risk receiving appropriate leadership?
- **Identify:** Are material assets, services, suppliers, risks, and improvement opportunities understood?
- **Protect:** Are identity, awareness, data, platform, and infrastructure safeguards operating as intended?
- **Detect:** Can potentially adverse events be found and analyzed in time to matter?
- **Respond:** Are incidents managed, analyzed, mitigated, reported, and communicated with clear authority?
- **Recover:** Can affected assets and operations be restored, and can recovery communication be coordinated?

The Functions are concurrent outcomes, not a task sequence. Use them to test coverage, not recreate a personal checklist. If an outcome lacks an owner, evidence, or escalation path, correct the operating system instead of silently adding it to the leader's task list.

For AI-enabled personal and program workflows, add this overlay:

- **Govern:** Approve use cases, owners, autonomy budgets, data boundaries, review cadence, and retirement criteria.
- **Identify:** Inventory AI systems, integrations, memory, data stores, credentials, and affected processes.
- **Protect:** Enforce least privilege, source separation, secrets handling, skill review, validation, approvals, and context boundaries.
- **Detect:** Monitor tool calls, retrieval, permission drift, injection signals, anomalies, failures, and out-of-role behavior.
- **Respond:** Suspend access, preserve traces, scope impact, route decisions, and communicate verified facts.
- **Recover:** Roll back, restore trustworthy state, rotate credentials, repair sources, revalidate, and revise authority.

Appendix: AI Agent Governance Checklist

Use this checklist when an AI-enabled workflow moves beyond drafting or occasional analysis. Keep the record short enough to maintain and specific enough to stop unsafe ambiguity.

Context and provenance

- Separate public research, internal policy, client evidence, production telemetry, and personal notes.
- Record source names, timestamps, effective dates, and known blind spots.
- Distinguish what a document states from what production evidence demonstrates.
- Prevent one client, project, or security boundary from becoming another by convenience.

Control plane

- Give every assistant, GPT, skill, automation, plugin, and agent a name, owner, purpose, and expiration condition.
- Record what it may read, write, trigger, send, retain, and share.
- State whether memory is ephemeral, local, provider-held, or shared, plus how it is deleted.
- Preserve enough audit evidence to reconstruct sources, approvals, tool calls, outputs, and final state.
- Test revocation of credentials, scheduled runs, integrations, plugins, and write access.

Autonomy budget

- Set numeric operational limits: systems, environments, records, changes, rate, and duration.
- Set information limits: classifications, exports, retention, and cross-boundary retrieval.
- Separate recommendations from decisions the system may make real.
- Bound recipients, channels, message types, spend, outage impact, and public visibility.
- Require approval for production, regulated data, destructive actions, external commitments, and difficult-to-reverse changes.

Instruction–data boundary

- Treat retrieved text and external content as untrusted influence.
- Extract and validate only the fields required for the next action.
- Allowlist tools, destinations, data types, and side effects.
- Put deterministic checks beside the action-producing tool.

- Test indirect prompt injection, poisoned documents, hidden text, malformed files, and manipulated tool results.

Agent record and evaluation

- Record purpose, owners, users, model/provider, runtime, instructions, skill versions, sources, permissions, autonomy, approvals, evidence, failure modes, cost ceiling, kill switch, and review date.
- Test ordinary, ambiguous, adversarial, incomplete-context, stale-data, tool-failure, and mandatory-stop cases.
- Measure correctness, citations, false confidence, boundary compliance, human approval, recovery, attention removed, and maintenance created.
- Retire systems whose dependable value no longer exceeds their automation tax.

Further reading: AI Agents Need Autonomy Budgets, AI Agents Are Already Working for You, Why My AI Agents Needed CaneCorso, The Personal AI Control Plane, and When the AI Moves Into the House.

Appendix: How This Book Came to Be

This book began as a work problem. The ability to handle more attracted more: questions, exceptions, projects, and unfinished decisions flowed toward the person who had proved he could carry them. From the outside that looked useful. From the inside it felt like a life being consumed by other people's queues.

I did not need a prettier task list. I needed questions that could change the work itself.

From MESAD to EDSAM

The earliest working form was MESAD, a compact set of filters I tested against real projects and workflows. David Allen's distinction between projects and next actions, Pareto's emphasis on leverage, Tim Ferriss's push toward elimination and automation, and years of practical experimentation all contributed pieces. Daily work was the test bench.

For roughly four years, I treated the model as a series of betas. I watched which questions people remembered, where the system failed, and whether it reduced the Money, Attention, and Time consumed while preserving the result. That three-currency view became MAT.

Dan Eckstein helped with the transition from MESAD to EDSAM. The new name gave the model a durable public form: Eliminate, Delegate, Simplify, Automate, Maintain. The 2012 presentation, *EDSAM in 30 Minutes or Less: Getting Back to "DO"*, called the filters "mental Legos." I still like that. The sequence matters because it prevents premature automation, but the filters can be reapplied whenever the work changes.

The continuity is simple: do not give work more of your life until you have asked what it deserves.

Why Security Leadership

Security combines incomplete information, asymmetric consequences, continuous change, and accountability that rarely fits inside an org chart. Again and again I heard leaders say, "I do not have time to do what I need to do."

That became this guide's design constraint. The answer was not faster personal throughput. It was a system in which less work required the leader. NIST CSF 2.0 offered a coverage check, but the lived Tuesday of the security leader remained the center.

Modern AI made the question more urgent. LLMs can summarize and compare; skills can preserve repeatable methods; agents can gather evidence and move bounded work. They can also accelerate bad processes, hide missing context, accumulate authority, and create infrastructure no one intended to operate. EDSAM supplies the question technology enthusiasm often skips: should this work exist in this form at all?

My Knowledge, Codex, and the Work Between Us

I brought the substance: the MESAD and EDSAM history, the original presentation, decades of security and advisory work, TaskGrid, patterns from vendor diligence, vCISO work, incidents, testing, governance, research, and writing, plus the judgment about what would help a working leader.

I used Codex as a research, editorial, production, and verification collaborator. It helped organize and pressure-test the guide, connect relevant writing from *State of Security* and *Not Quite Random*, build diagrams and publication files, test links and accessibility, render pages, and perform repeated editorial and visual checks.

Those capabilities accelerated the work. They did not replace authorship. I chose the direction, accepted or rejected changes, supplied lived context, resolved the model's meaning, and remained responsible for the publication. Codex could retrieve, structure, draft, compare, and verify. The history, intent, taste, relationships, and consequences remained mine.

The collaboration mirrored EDSAM:

- **Eliminate:** Reuse proven source material; remove repetition and abandoned paths.
- **Delegate:** Give research, formatting, checking, rendering, and mechanical consistency to a capable tool.
- **Simplify:** Keep one audience and one capacity problem at the center.
- **Automate:** Use a repeatable build to synchronize DOCX, PDF, links, metadata, and visual checks.
- **Maintain:** Retain voice, history, risk boundaries, editorial judgment, and final approval.

A Legacy of Knowledge

This appendix exists because readers deserve transparency. AI-assisted does not mean experience-free, responsibility-free, or error-free. Sources still need checking. Generated language still needs editing. Pages still need inspection. A human still owns what is published.

MESAD became EDSAM through use and collaboration; EDSAM became this field guide because the environment changed and the model remained useful. My larger goal is a legacy of knowledge: a working library that other people can test, challenge, improve, and pass forward. This book should not freeze the method in place. Use it, keep evidence, change what fails, and share the lesson.

Glossary

Attention: Cognitive and emotional capacity for interpretation, decisions, communication, and accountability.

AI agent: A model-enabled system that pursues a goal through tools or actions that can create consequences.

Autonomy budget: An agent's maximum independent authority before human approval, bounded by operational, information, decision, communication, economic, or reputational limits.

Control plane: Governance for the identity, permissions, memory, audit, and revocation of AI-enabled systems.

Control owner: The person accountable for a security control's design, operation, monitoring, or evidence.

Decision window: A scheduled period for reviewing non-urgent approvals, recommendations, and escalations in a batch.

EDSAM: Eliminate, Delegate, Simplify, Automate, Maintain. A sequence of filters applied to projects, tasks, and recurring workflows.

Exception: A case that falls outside an approved rule, pattern, tolerance, or delegated authority and therefore requires additional review.

FRICT: Frequent, Rules-based, Information-moving, Checklist-driven, and Templated; a filter for identifying stronger automation candidates.

Human-in-the-loop approval: A pause before a side effect until an authorized person or policy approves it.

Leader-only work: Work requiring the leader's unique authority, judgment, relationship, accountability, or independent challenge.

MAT: Money, Attention, and Time, rated High, Medium, or Low to expose the true resource cost of work.

Risk owner: The person authorized and accountable to decide how an organizational risk will be treated.

SCOPE: Stakes, Context, Options, Proof, and Exceptions; a pre-deployment test for an AI-enabled workflow.

Skill: A reusable, versioned package of instructions, templates, examples, and optional scripts that encodes a defined AI workflow.

TaskGrid: L. Brent Huston's cross-platform personal task system for integrating urgency, importance, agency, commitment, planning horizons, decisions, and EDSAM dispositions into one operating workflow.

Work-in-progress limit: A deliberate cap on significant outcomes receiving personal effort.

Important Notice and Disclaimer

This publication is provided solely for general educational and informational purposes. It presents a management framework and practical considerations for information security leaders. It does not constitute legal, regulatory, compliance, audit, accounting, tax, financial, insurance, employment, human-resources, safety, incident-response, or other professional advice. Reading or using it does not create an attorney-client, consultant-client, fiduciary, advisory, or other professional relationship with MicroSolved, Inc., L. Brent Huston, or any contributor.

Information security, privacy, technology, legal, and regulatory conditions change rapidly and vary by organization, industry, location, contract, threat environment, and use case. Readers are responsible for identifying and satisfying all applicable laws, regulations, contractual obligations, policies, standards, professional duties, and risk-management requirements. Verify current requirements and obtain advice from appropriately qualified professionals before making decisions with legal, regulatory, financial, safety, employment, or material business consequences.

Examples, scenarios, checklists, worksheets, metrics, decision criteria, automation concepts, and templates are illustrative and intentionally general. They are not complete controls, operating procedures, incident plans, legal analyses, or compliance determinations. Readers must independently evaluate, tailor, authorize, test, monitor, and document any practice before use. No example should remove required oversight, segregation of duties, evidence preservation, safety protections, legal review, regulatory reporting, or accountable human judgment.

Applying EDSAM does not guarantee that an organization can safely eliminate work, reduce staffing, delegate authority, simplify controls, automate decisions, prevent incidents, detect attacks, satisfy obligations, recover operations, improve productivity, reduce cost, or achieve another result. Outcomes depend on governance, competence, staffing, authority, data quality, implementation, testing, monitoring, culture, and threat conditions. Efficiency gains are not evidence that required capacity, independent assurance, or professional expertise is unnecessary.

Automation and artificial intelligence can introduce errors, bias, privacy concerns, security weaknesses, excessive permissions, unsafe actions, and difficult-to-detect failures. Readers remain responsible for decisions and actions taken using automated systems, generated content, scripts, integrations, models, or tools. High-impact, destructive, irreversible, safety-sensitive, legally significant, or externally communicated actions require controls and human authorization appropriate to their consequences. Do not use this publication as the sole direction during an active incident, emergency, or crisis.

To the fullest extent permitted by law, this publication and associated materials are provided “as is” and “as available,” without express, implied, statutory, or other warranties, including warranties of accuracy, completeness, currency, reliability, merchantability, fitness for a particular purpose, title, noninfringement, security, availability, or results. MicroSolved, Inc., L. Brent Huston, and contributors do not warrant that the publication is error-free, complete, suitable for a particular use, or sufficient to identify or manage every risk.

To the fullest extent permitted by law, MicroSolved, Inc., L. Brent Huston, and contributors will not be liable for any loss, claim, obligation, damage, cost, or expense arising from use of, inability to use, or reliance on this publication or associated materials. This includes direct, indirect, incidental, consequential, special, exemplary, and punitive damages; loss of data, revenue, profits, opportunity, goodwill, or continuity; security incidents; compliance failures; regulatory action; third-party claims; and investigation, response, remediation, restoration, or professional-service costs. Where law does not permit a particular exclusion or limitation, it applies only to the maximum extent permitted.

Names, trademarks, service marks, product names, company names, frameworks, and third-party materials remain their owners’ property and are used for identification, commentary, or reference. Inclusion does not imply sponsorship, certification, affiliation, or endorsement. References to the NIST Cybersecurity Framework do not imply endorsement by NIST or the United States government. MicroSolved, Inc. does not control third-party websites or resources and is not responsible for their content, availability, security, or privacy practices.

By using this publication, readers accept responsibility for their analysis, decisions, implementation, supervision, and results. If these terms are unacceptable, do not use the publication or associated materials.

Sources and Further Reading

External links and product documentation were checked on August 31, 2026. Product names, features, availability, and guidance can change; verify current documentation before implementation.

EDSAM and Personal Operating Systems

- Huston, L. Brent. *EDSAM in 30 Minutes or Less: Getting Back to "DO"*. Digital Lizard, LLC, 2012.
- Huston, L. Brent. "Systems Thinking and Mental Models: My Daily Operating System." *Not Quite Random*, April 23, 2025. Read the article
- Huston, L. Brent. "Build Systems for Your Worst Days, Not Your Best." *Not Quite Random*, November 3, 2025. Read the article
- Allen, David. *Getting Things Done: The Art of Stress-Free Productivity*. Penguin Books.
- Ferriss, Timothy. *The 4-Hour Workweek*. Crown.

Security Program Coverage

- National Institute of Standards and Technology. *The NIST Cybersecurity Framework (CSF) 2.0*. NIST CSWP 29, February 26, 2024. NIST CSF 2.0 publication
- National Institute of Standards and Technology. *NIST CSF 2.0 Implementation Examples*. February 26, 2024. NIST implementation examples

AI Agents, Authority, and Security

- Huston, L. Brent. "AI Agents Need Autonomy Budgets, Not Just Governance Policies." *MSI State of Security*, June 22, 2026. Read the article
- Huston, L. Brent. "AI Agents Are Already Working for You. Who's Managing Them?" *MSI State of Security*, May 11, 2026. Read the article
- Huston, L. Brent. "Why My AI Agents Needed CaneCorso as a Security Control Plane." *MSI State of Security*, May 4, 2026. Read the article
- Huston, L. Brent. "How to Secure Your SOC's AI Agents: A Practical Guide to Orchestration and Trust." *MSI State of Security*, July 9, 2025. Read the article
- Huston, L. Brent. "The Personal AI Control Plane: How to Govern Your Agents Before They Govern Your Workflow." *Not Quite Random*, May 11, 2026. Read the article
- Huston, L. Brent. "When the AI Moves Into the House." *Not Quite Random*, May 18, 2026. Read the article
- Huston, L. Brent. "The Automation Tax We're Not Pricing." *Not Quite Random*, March 25, 2026. Read the article

- Huston, L. Brent. "The FRICT Method: A Not-Quite-Random Way to Spot Automation Gold." *Not Quite Random*, February 23, 2026. Read the article

Platform Guidance

- OpenAI. "Skills." *OpenAI Developers*. OpenAI Skills documentation
- OpenAI. "Safety in Building Agents." *OpenAI Developers*. OpenAI agent-safety guidance
- OpenAI. "Guardrails and Human Review." *OpenAI Developers*. OpenAI guardrails and approvals

About the Author

L. Brent Huston is the creator of EDSAM, CEO and Security Evangelist at MicroSolved, Inc., and Principal Lizard at Digital Lizard, LLC. For more than two decades, he has worked across information security, risk, research, advisory services, and emerging technology. His writing focuses on turning mental models into useful systems—and on building a practical legacy of knowledge that others can test, improve, and carry forward.

Continue the Work

- For practical information security research and guidance, visit MSI State of Security.
- For personal essays on mental models, systems thinking, AI, and the future, visit Not Quite Random.
- To learn more about MicroSolved, Inc., visit MicroSolved.

> Useful knowledge should not end with the person or project that produced it. Test it, improve it, and pass it forward.