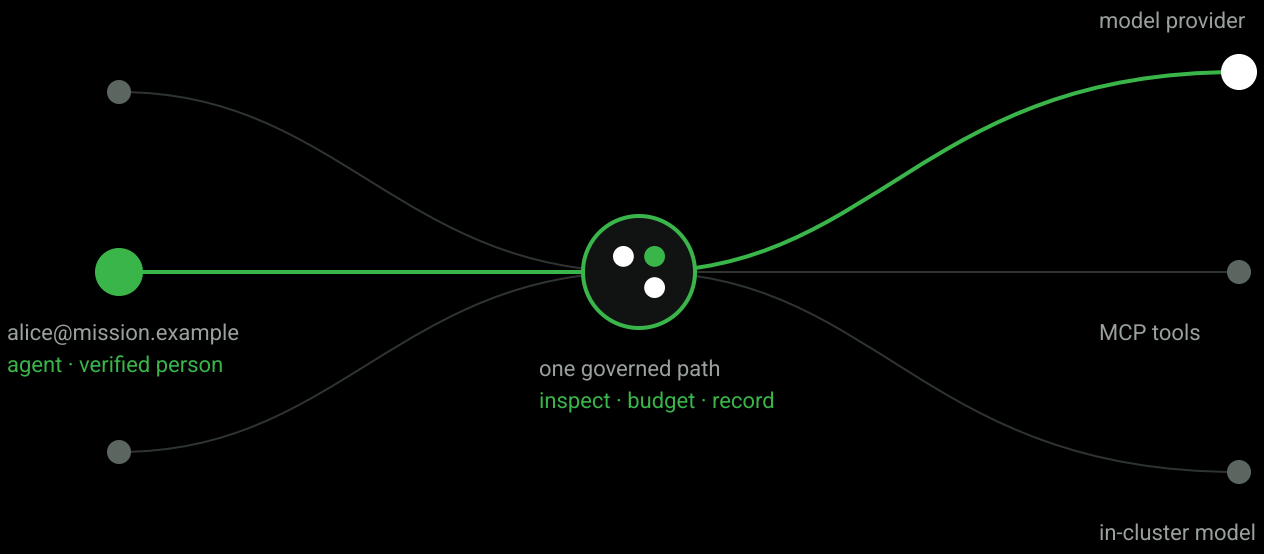
Greymatter.io

WHITE PAPER
GREYMATTER 2.5.0 · NOVEMBER 2026



Governed AI: one path from people and agents to every model and tool

How Greymatter puts every model call and every MCP tool call on one governed path. The path carries who asked, inspects what goes out and comes back, enforces each person's budget, and records it all.

© 2026 Greymatter.io, Inc.

Contents

	SECTION		PAGE
THE PROBLEM	1	Executive summary	1
	2	AI traffic escapes the controls you already have	3
	3	The architecture in one picture	4
MODELS	4	LLM egress: one governed path out	5
	5	Guardrails and DLP	6
	6	Token budgets and rate limits	8
TOOLS	7	MCP Router: many tool servers, one endpoint	10
	8	Ask Greymatter: the platform governs its own AI	13
EVIDENCE	9	The record of every model and tool call	14
	10	Mapping to the OWASP LLM Top 10	15
	11	Benefits and key takeaways	16
	12	Evidence for AI governance	17
	13	Ask Greymatter	18

How to read the status marks

Every capability in this paper carries one mark. Nothing marked Fielded is aspirational.

FIELDED

Ships in Greymatter 2.5.0 today.

PREVIEW

Ships in 2.5.0 as a technology preview.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

1

Executive summary

Every model call carries a prompt, a key and often mission data. Every tool call lets software act. A program must know who asked, what left, what came back, and what it cost.

IN BRIEF

Greymatter governs AI traffic in the data plane that already carries every call. There is no separate AI gateway, no new seam, no SDK and no application change.

WHO THIS IS FOR

Program offices, AI and data leads, security architects and assessors who must approve AI use on mission systems.

Mission teams adopt language models and agents faster than security teams can review them or budgets can track them. Many programs run several frontier and sovereign models at once, each with its own account and its own bill. Each application brings its own provider key, calls the model directly, and logs whatever it chooses. Agents add tools through the Model Context Protocol (MCP). Each tool server becomes one more endpoint with its own sign-in and its own idea of who the caller is.

The result is a set of blind spots. Nobody can say which person a model call served. Prompts can carry personal data or classification markings out of the enclave. Responses can carry injected instructions back in. A single runaway agent can spend a month's provider budget in a day.

Greymatter closes these gaps in the data plane, where every call already passes. The data plane holds the verified person behind each call, because Greymatter carries that person from hop to hop. It inspects prompts, responses and tool calls. It counts each person's tokens against one budget across every model, so the enterprise manages the accounting for all of them in one place. And it records every model call and every tool call, without keeping a credential that someone could replay.

Greymatter does not add a separate AI gateway. A bolt-on gateway is one more product with its own identity model. It opens a new seam outside the platform. An outside endpoint that internal traffic must reach before anyone can govern it does the same. Each of these breaks the zero trust a program is trying to build. Greymatter governs AI calls with the same verified identity, mutual TLS and policy as every other call in the platform.

THE CENTRAL POINT

Greymatter gives every model call and every tool call one governed path: a verified person, inspection both ways, a budget, and a complete record.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

Six properties a program office can test

One path out

FIELDED

Programs route model calls through an LLM egress they declare. Greymatter shows the other outbound calls of the workloads it manages by name.

The person on every call

FIELDED

Greymatter carries the verified person to the egress and the tool server. No token travels with the call, and the person's identity never reaches the provider.

Inspection both ways

FIELDED

Guardrails covers the OWASP LLM Top 10. DLP finds personal, health, logistics and sensitive data in prompts and responses.

A budget per person

FIELDED

Each person gets one daily token budget, optionally by group. One total can span every model the program offers, and every replica.

Tools by role

PREVIEW

The MCP Router offers many tool servers as one endpoint, and each role lists and calls only the tools it holds.

A record without secrets

FIELDED

Every model and tool call lands in the audit record with its person, its findings and its tokens. No record holds a usable credential.

STATED PRECISELY, SO IT IS NOT OVER-READ

Greymatter governs the traffic to and from models and tools. It does not host models, judge the quality of an answer, or replace a program's own review of an AI use case.

Guardrails and DLP detect known patterns of attack and sensitive data. They reduce risk and produce evidence. No inspection layer can promise that a model never misbehaves.

AGENTS AND BOTS STILL NAME A PERSON

Agents and bots increasingly act on their own, long after a person asked them anything. Greymatter keeps a person on the record anyway.

An agent that keeps working after a request ends still names the person who asked. A workload that runs for one person names that person on every call. Each audit record and finding shows whom the work served.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

2

AI traffic escapes the controls you already have

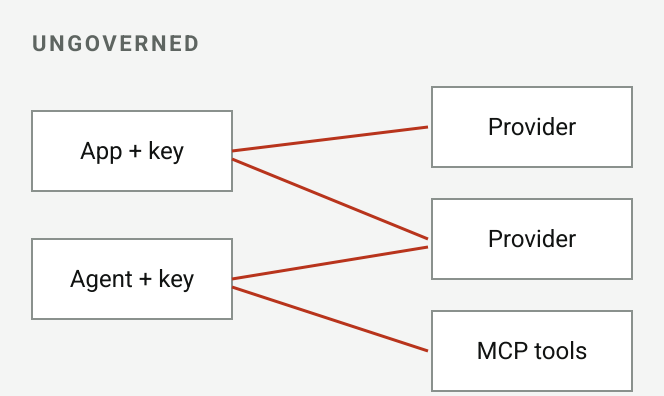
Firewalls see an HTTPS connection to a provider. They do not see the person, the prompt, the tool or the cost.

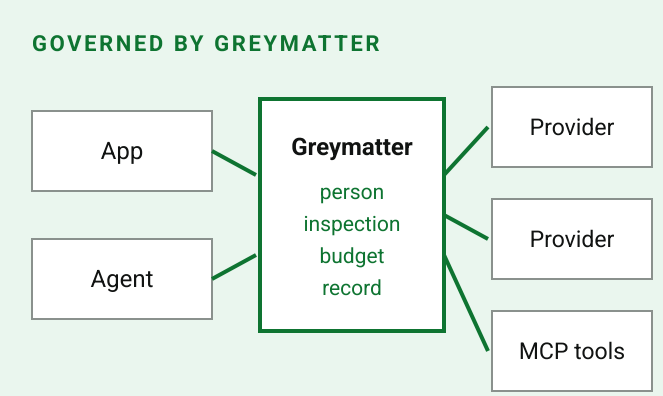
WHY NOW

Agents turn one question into many model and tool calls. Most of them run after the person's request has already returned.

1. **Keys live in applications.** Each team holds its own provider key, so the program cannot tell which person or workload spent it.
2. **Data leaves in prompts.** A prompt can carry personal data, health records, logistics details or classification markings to a provider outside the enclave.
3. **Instructions come back in responses.** A response or a retrieved document can carry injected instructions that turn an agent against its own user.
4. **Tools multiply the reach.** Every MCP server is a new endpoint that can act, each with its own sign-in and its own view of the caller.
5. **Cost has no owner.** A runaway loop can spend the provider quota for everyone, and nothing ties the spend to a person.

FIGURE 1 · UNGOVERNED AND GOVERNED AI TRAFFIC



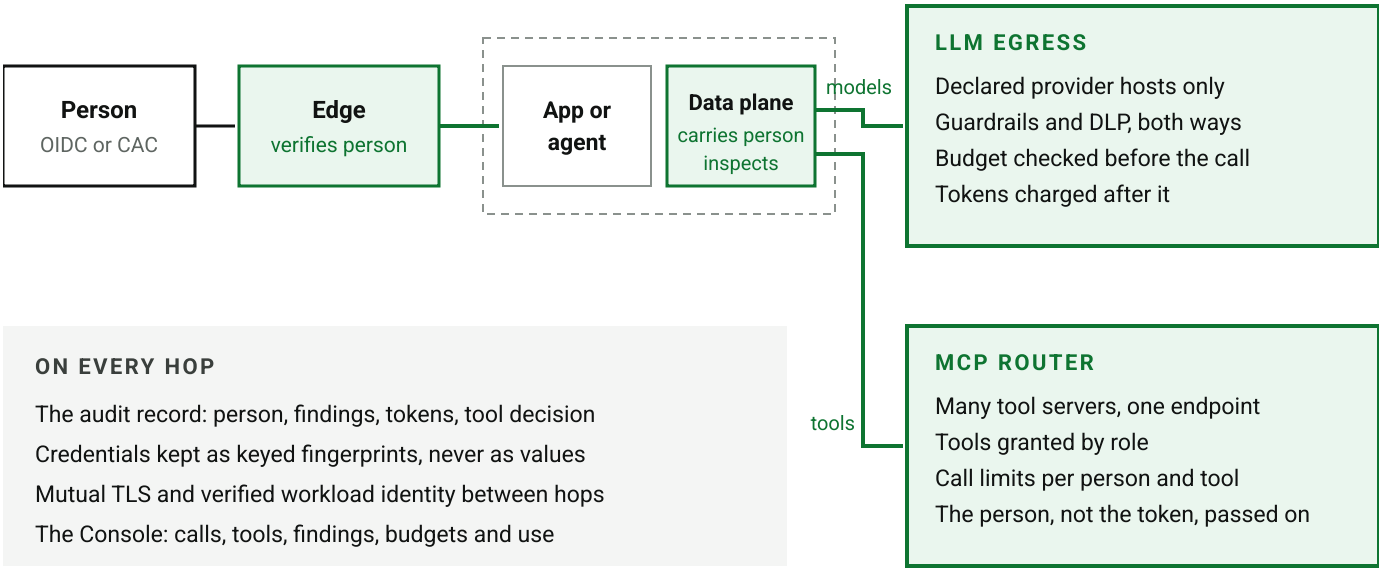


© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

3 The architecture in one picture

Four capabilities, all in the data plane that already carries every call. Nothing new sits in the path. FIELDED

FIGURE 2 · THE GOVERNED PATH FOR MODELS AND TOOLS



Solid green marks the governed path. The application and agent change nothing to use it.

CAPABILITY	THE QUESTION IT SETTLES
LLM egress	Which providers may this program reach, and through what path?
Guardrails and DLP	What did the prompt, the response or the tool call contain?
Budgets and limits	How much may each person, group or tenant use?
MCP Router	Which tools may this person's role see and call?

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

4

LLM egress: one governed path out

An LLM egress is an edge or service whose route sends calls to a model provider. Applications call it instead of the provider. FIELDED

HOW IT WORKS

A tenant declares the egress and its routes like any other Greymatter edge, in GSL or as custom resources.

PROVIDERS

Hosted APIs such as OpenAI, Anthropic and Azure OpenAI, and OpenAI-compatible models that run in the program's own cluster.

The application keeps its OpenAI-compatible client and points it at the egress. Its call travels over mutual TLS through its own data plane, which carries the verified person beside the call. The egress then checks the person's budget, inspects the prompt, forwards the call to the provider, inspects the response, and charges the tokens.

What the egress guarantees

- **Only the hosts you declare.** The egress reaches the provider hosts that its routes name, over TLS, and no others.
- **The person's identity never reaches the provider.** The egress records the person and the person's groups, then removes the headers that carry them before the call leaves.
- **One count across replicas.** Every replica of the egress shares the same totals, so scaling out never multiplies a budget.
- **Standard clients keep working.** A person over budget gets the error that OpenAI-compatible clients already show.

Calls that skip the egress stand out

Greymatter observes the outbound connections of the workloads it manages and names each destination it can. The Console network graph names a direct call to a model provider or a package registry, apart from the governed path. The program can then find the call and close it.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

5

Guardrails and DLP

Two inspection products in one pass over each body: Guardrails for attacks on AI systems, and DLP for sensitive data. FIELDED

WHERE IT RUNS

In the data plane of any listener: an edge, a service, an LLM egress, an MCP Router, or a workload's own outbound calls.

NOT THE WAF

Greymatter WAF runs the OWASP Core Rule Set as a separate filter. A listener can carry all three.

Guardrails and DLP read each request and response once and run both products over it. They inspect HTTP bodies, event streams, WebSocket and Socket.IO messages, and each MCP message on its own. They run after authentication, so a request that fails sign-in never reaches them, and before authorization, so policy can act on what they found.

FIGURE 3 · DETECT FIRST, THEN ENFORCE

MODE	MODE · DEFAULT	MODE · GUARDRAILS ONLY
Off	**Detect**	**Enforce**
The product skips the scan.	The product records every finding and blocks nothing, so a program sees what enforcement would stop.	Guardrails also blocks, for each module the program sets to enforce.

DLP records and alerts, and never blocks. A program turns on enforcement one Guardrails module at a time, after detect mode shows what each module would stop.

A PROGRAM'S CHOICE

Enforcement can hold traffic until a scan finishes, or let it flow and block as soon as the scan ends.

Controls a caller cannot switch off

- **No header changes the mode.** A request cannot turn a scan off, change a score or exempt itself. Greymatter records such an attempt as a finding.
- **Findings name a verified person.** Rules about a user read the person Greymatter resolved, never a header the application set.
- **Records hold no secrets.** Findings keep fingerprints of tokens and session values, never the values.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

What each module looks for

OWASP	GUARDRAILS MODULE	WHAT IT LOOKS FOR	AUDIT
LLM01	Prompt injection	Instructions that try to override the system or the user	Yes
LLM03	Supply chain	Risky models, weights and software artifacts	Yes
LLM04	Data and model poisoning	Attempts to poison training or retrieval data	Yes
LLM05	Improper output handling	Model output that a caller could execute or render unsafely	Yes
LLM06	Excessive agency	Tool misuse and actions beyond an agent's remit	Yes
LLM07	System prompt leakage	Responses that reveal hidden instructions or credentials	Yes
LLM08	Vector and embedding weaknesses	Retrieval that crosses tenant boundaries	Yes
LLM09	Misinformation	Signs of fabricated or unsupported claims	Yes
LLM10	Unbounded consumption	Requests built to exhaust tokens or compute	Yes

OWASP	DLP MODULE	WHAT IT FINDS	AUDIT
LLM02	Disclosure	PII: national ID, card and bank numbers, contact details and credentials	Yes
LLM02	Health data	Patient records, diagnosis and drug codes, genomic IDs	Yes
LLM02	Logistics	Hazardous material IDs, manifests, fleet locations, infrastructure plans	Yes
LLM02	Classification markings	CAPCO banners and portion marks, such as SECRET//NOFORN	Yes

ALLOW AND DENY LISTS

Egress hosts, packages, tool names and MCP methods. Each list stays off until a program turns it on.

Agents and MCP

With MCP inspection on, Guardrails checks each MCP request against a method allowlist and the program's tool lists. It returns a proper JSON-RPC error for a blocked MCP call. It scans each MCP message in an event stream on its own. Image inspection is a program's choice, and it can block in enforce mode.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

6

Token budgets and rate limits

Each person gets a daily number of model tokens. Each tenant, service or person can also get a request rate. Greymatter counts both in one place. FIELDED

THE PERSON

The person Greymatter verified for the request the call serves, carried by the data plane. No token reaches the egress, and no token exchange takes place.

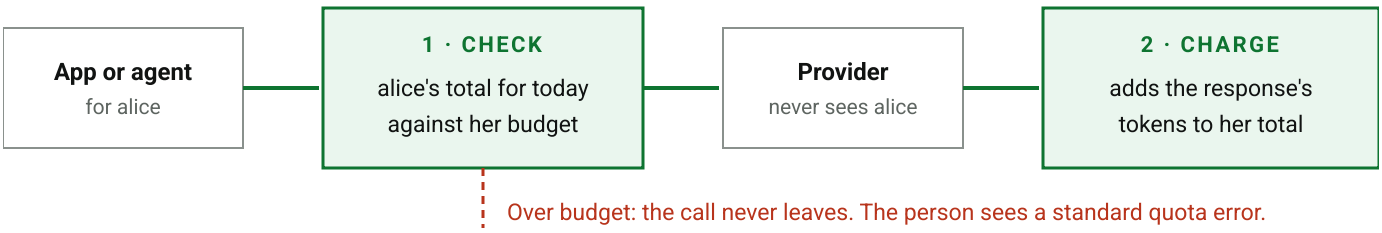
A CALL WITH NO PERSON

A call that serves no verified person counts against its workload's own budget. A program can refuse such calls instead.

A budget sits on the route to the provider. The program sets a daily figure per person, and it can set a different figure for each group, such as analysts or interns. A person in several groups gets the largest of their budgets. Groups come from the person's verified sign-in, and the data plane carries them beside the person.

Each person's total starts again every day. The egress reads each response's token use from the provider's own figures. When a provider reports none, the egress estimates it, and the record says which it used.

FIGURE 4 · CHECK BEFORE THE CALL, CHARGE AFTER IT



A person can pass the budget by one call, never by more. The egress learns the token use only when a response ends. So the call that crosses the line completes, and the next one meets a refusal.

One budget across every model

BUDGET DOMAIN

Routes in one namespace that name the same domain share each person's total.

Suppose a program offers two frontier models and a sovereign model, and means to allow each person 200,000 tokens a day. A separate budget on each model would allow 600,000. When all three routes name one budget domain, each person draws on one daily total, whichever model they choose. Ask Greymatter works this way: all its providers share one budget.

The budget counts tokens, not money. Providers price tokens differently, so a program sets the figure with its costliest model in mind.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

One count, everywhere, for each tenant alone

HOW IT WORKS

The platform runs one rate limit service for every namespace. Its totals live on the platform's message bus, which every replica reads and adds to.

- **Every replica shares one total.** A person's budget means the same thing whether one data plane or fifty serve the traffic.
- **Each tenant counts alone.** Greymatter takes the caller's namespace from the calling workload, not from the request, so one tenant cannot read or change another tenant's totals.
- **Figures come from policy.** Budgets and limits come only from the program's declared configuration, never from a request.
- **The program chooses how failure behaves.** When the count is unavailable, the egress allows the call by default. A program can refuse the call instead.

Global rate limits

WINDOWS

Second, minute, hour or day, set by the program.

The same service enforces request limits on any listener: per tenant, per person, or per any value a request carries. A limit can give each value its own total, such as one per person, or apply to one value alone. Every replica in the namespace shares each limit.

Who sees the use

PRIVACY

Rows that name a person show only to administrators of that person's namespace or the mesh.

The Console shows each budget and limit on the service that enforces it: the current window, the use against it and the top users. A platform view lists every namespace with budgets or limits, with totals and no names. Each view holds only the namespaces the viewer's roles cover.

SETTING	WHAT THE PROGRAM DECIDES
Daily budget per person	The token figure for everyone without a group budget
Budgets by group	A figure for each named group from the person's verified sign-in
Budget domain	Which routes share one total for each person
Require a person	Whether to refuse calls that serve no verified person
Failure mode	Whether to allow or refuse calls when the count is unavailable

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

7

MCP Router: many tool servers, one endpoint

An MCP client connects once and lists the tools, prompts and resources of many MCP servers as one server. PREVIEW

WHERE IT RUNS

In the data plane of a tenant service. The data plane responds to the client and calls each MCP server itself, so no other component sits in the path.

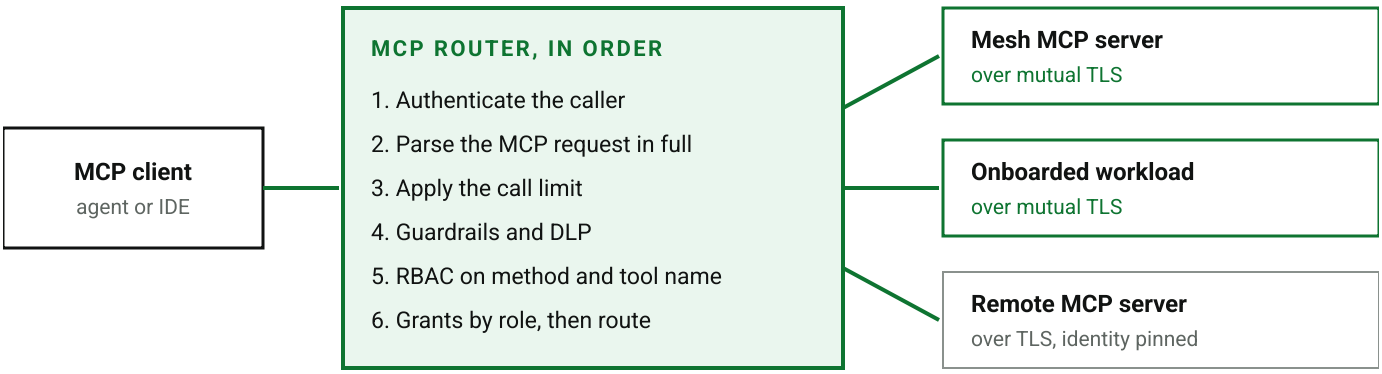
BOTH PROTOCOL ERAS

The router serves the 2025 session-based revisions and the 2026 sessionless revision on one endpoint, and reaches MCP servers of either era.

Agents gain their reach through tools. Without a router, every MCP server is a separate endpoint with its own sign-in and its own list of tools. Each one also has its own idea of who the caller is. The MCP Router replaces that with one governed endpoint per service.

The router's MCP servers can be services in the mesh, workloads that Greymatter onboarded automatically, or MCP servers outside the mesh reached over TLS. When two servers offer a tool of the same name, the router keeps them apart by prefixing each name with its server's name.

FIGURE 5 · ONE ENDPOINT, EVERY TOOL SERVER



Guardrails is a fast first look. The router's own checks are final. Grants, tool selections and method lists work on the full parse and apply in every Guardrails mode. A refused tool stays refused even while Guardrails only detects.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

Tools, prompts and resources by role

FAILS CLOSED

A caller's token might carry no roles, or roles that grant nothing of a kind. That caller receives an empty list and meets a refusal on every call of that kind.

The router reads the caller's roles from the verified token on every request. Each role holds grants: single tools, every tool of one server, prompts, and resources by exact address or prefix. The router removes from each list what the caller's roles do not grant. It refuses a call to anything outside them with a clear error.

A tool a role cannot use never appears to that role, so an agent never learns that it exists. That matters, because a model plans with the tools it can see.

GRANT	WHAT THE ROLE MAY SEE AND CALL
One tool	A single tool of one MCP server
One server	Every tool that one MCP server offers
Prompts	One prompt, or every prompt of a server
Resources	One resource, every resource under a prefix, or every resource of a server

Call limits per person and tool

A PROGRAM'S CHOICE

How many calls each caller may make to each tool, and over what window.

A call limit bounds how often one caller may call one tool. Every replica of the router shares the count. A call over the limit receives a standard rate-limited response that states when the client can try again. Lists, prompts and resources stay available, so an agent can still plan while it waits.

Sign-in that clients find on their own

The router publishes standard OAuth discovery metadata. An MCP client given only the router's address finds where to sign in, with no hand configuration. Clients that follow the MCP specification do this already.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

The person reaches the tool, the token does not

NEVER FORWARDED

The client's authorization header, cookies and session ID never reach any MCP server.

A tool server often needs the identity of the person it serves. Passing the client's token on would hand every tool server a credential it could replay elsewhere. The router passes the verified person instead, in a header that the router itself writes for each MCP server that requests it. The router never forwards the client's own copy of that header.

A tool server trusts that header only when its own data plane admits the router alone. Greymatter's impersonation check provides exactly that: the router's identity on the allowlist, and every other caller that sends the header refused.

Sessions that cannot move between callers

KEY ROTATION

One platform setting rotates every namespace's key. A larger step revokes the key in force everywhere.

For clients that use sessions, the router signs each session with its namespace's own key. The session records the caller and the listener, and the router checks both on every request. A session cannot move to another caller or another router. Sessions keep no state in the data plane, so the router scales with its service's replicas. Audit records keep a fingerprint of each session ID, never the ID.

Tool servers outside the mesh

A remote MCP server needs TLS, a trust bundle and the identity its certificate must present. When it needs a credential, Greymatter stores that credential as a secret and never places it in configuration. The data plane adds it to each request for that server only.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

8

Ask Greymatter: the platform governs its own AI

The Console's assistant runs on the same governed path this paper describes. It is a working example of every control. FIELDED

YOUR MODEL

OpenAI, Anthropic, Azure OpenAI, or any OpenAI-compatible model, including one that runs in the program's own cluster.

Ask Greymatter answers questions about the mesh in plain language on every Console page. For example, it explains what is unhealthy, why a call failed, and which controls are in place. It reads live health, security, traffic, change and documentation data through Greymatter's own tools.

CONTROL	HOW ASK GREYMATTER APPLIES IT
The person on every call	The tools read the Greymatter API as the person who asked, through the MCP Router. An answer shows only what that person's roles let them see in the Console.
One path out	The assistant calls its model through the platform's own LLM egress, which reaches only the provider hosts the platform lists. With no host listed, it reaches none.
Sealed provider keys	The Greymatter API seals each provider key to its provider and stores only the sealed form. Only the assistant can open a key, and a key moved to another provider does not open.
Inspection	Guardrails and DLP read each model call and each response.
A budget per person	Each person has a daily token budget, optionally by platform role. A person over budget gets a plain message that says when it resets.
The record	Each model call and each tool call lands in the audit record with the person who asked.

WHY THIS MATTERS TO A PROGRAM

An assessor can watch the controls work on a real assistant before the program approves its own AI applications. The same LLM egress, budgets, inspection and MCP Router serve a tenant's applications the same way.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

9

The record of every model and tool call

Each hop writes its own audit record where the call happens, with credentials already scrubbed. FIELDED

CALL	WHAT THE RECORD HOLDS
A model call	The verified person and how Greymatter learned it, the model, and the input, output and total tokens. The record also says whether the provider reported the tokens or Greymatter estimated them.
A tool call	The MCP method and the tool, prompt or resource the client named. The router's decision and its reason, the MCP server, and the protocol revision on each side.
An inspection	Each Guardrails and DLP finding, which product raised it, the scores, the mode, and the person the call served.
A credential	A keyed fingerprint for each token, session ID or API key, so records link without holding anything a reader could replay.

PLAIN LANGUAGE

The Console explains each finding in words, with its scores, and says whether a filter verified the person behind the token.

Where to read it

- **App Guard.** Guardrails, DLP and WAF findings on one page, with what detect mode would have blocked.
- **MCP Router tab.** The router's MCP servers and whether it reaches each one, every call explained in words, and each tool with how callers used it.
- **Rate limits tab.** Each budget and limit, today's use and the top users.
- **Forensic Audit.** Sensitive-data findings person by person, and every hop of a request in order.
- **The Greymatter API.** The same data for a program's own SIEM and reporting.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

10

Mapping to the OWASP LLM Top 10

Each risk, and the Greymatter controls that address it. Inspection is one layer. The path, the person and the budget add others.

RISK	GREYMATTER CONTROLS
LLM01 Prompt injection	Guardrails prompt injection module, in detect or enforce mode
LLM02 Sensitive disclosure	DLP on prompts and responses, and credentials fingerprinted in every record
LLM03 Supply chain	Guardrails supply chain module, with package and host allowlists
LLM04 Data and model poisoning	Guardrails poisoning module
LLM05 Improper output handling	Guardrails output handling module on responses
LLM06 Excessive agency	Guardrails agency module, with MCP Router grants by role, tool selections and call limits
LLM07 System prompt leakage	Guardrails prompt leakage module on model responses
LLM08 Vector and embedding weaknesses	Guardrails embedding module, with tenant isolation in the mesh
LLM09 Misinformation	Guardrails misinformation module
LLM10 Unbounded consumption	Guardrails consumption module, token budgets per person and global rate limits

FRAMEWORKS

The Console's Stoplight view also measures controls against DoD Zero Trust, CISA ZTMM, federal mandates, responsible AI and GDPR.

Beyond inspection

Every model and tool call also crosses mutual TLS between verified workloads, names a verified person, and leaves a record. Those properties come from the mesh itself, and they hold whether Guardrails runs in detect or enforce mode.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

11 Operational and mission benefits

Each outcome, what produces it, and its status.

OUTCOME	WHAT PRODUCES IT	STATUS
Approved providers only	An LLM egress that reaches only the provider hosts the program declares	FIELDED
Shadow AI made visible	Outbound calls to providers and registries named in the network graph	FIELDED
Attribution to a person	The verified person carried on every model and tool call	FIELDED
Privacy toward providers	The person removed before the call leaves	FIELDED
Prompt injection defense	Guardrails in detect mode, then enforce per module	FIELDED
Data loss visibility	DLP for personal, health, logistics and classified data, both ways	FIELDED
Cost control per person	Daily token budgets by person and group, one count across replicas	FIELDED
Fair use across tenants	Global rate limits, with each tenant counted alone	FIELDED
Tools by least privilege	MCP Router grants by role that fail closed	PREVIEW
No tokens at tool servers	The router passes the person, never the client's credentials	PREVIEW
Bounded agent activity	Call limits per person and tool	PREVIEW
Evidence without secrets	Every call recorded, with credentials kept as keyed fingerprints	FIELDED
Governed assistant in the Console	Ask Greymatter on the same path, as the person who asked	FIELDED

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

12

Evidence for AI governance

AI governance frameworks require controls on how AI is used and records that prove they ran. Greymatter supplies both on every model and tool call. FIELDED

EXTEND, DO NOT BUILD

A team governs AI in the mesh it already runs. It builds no logging pipeline and buys no second gateway.

International AI frameworks share a pattern. Examples are the OECD AI Principles, ISO/IEC 42001, the NIST AI Risk Management Framework (AI RMF) and the EU AI Act. Each requires an organization to find who uses AI and how, set limits, and keep a trail. Defense programs add NATO's principles, and national rows come last.

FRAMEWORK	WHAT IT REQUIRES	WHAT GREYMATTER SUPPLIES
OECD AI Principles	Accountability, and transparency and explainability	Every model and tool call on record, with the person, the findings and the tokens
NATO Principles of Responsible Use of AI in Defence	Governability, and responsibility and accountability	Budgets and tool grants per person and group, and a person on every call
ISO/IEC 42001	Operation and monitoring	Runtime controls on one governed path, and the records that show they ran
NIST AI RMF	Map, Measure and Manage	Which people and agents use which models and tools, at what cost, under which limits
EU AI Act	Record-keeping, Article 12	Automatic logs of each use, kept by the platform rather than the application
United States Intelligence Community AI ethics	Transparent and accountable. Secure and resilient	The same record, sealed provider keys, and inspection on every call
United States DoD AI Ethical Principles	Traceable and governable	A record for every call, and policy that can cut off a model or tool

Using Greymatter does not make a system compliant with any framework. It supplies controls and evidence an assessor can test. National rows come last.

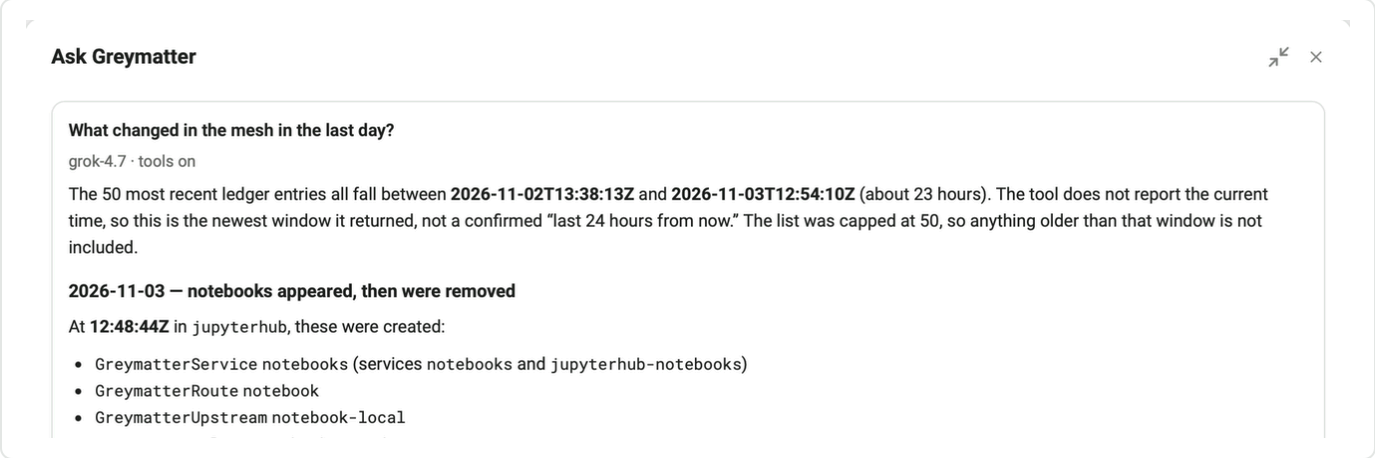
© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

13

Ask Greymatter

Ask Greymatter answers questions about who uses which models and tools, and at what cost. It runs on the same governed path. FIELDED

FIGURE 6 · ASK GREYMATTER IN THE CONSOLE



READS AS YOU

Each tool reads the platform as the person who asked, and only reads. An answer holds only what that role allows.

GOVERNED LIKE ANY AI USE

Guardrails and DLP check each model call. The audit trail names the person and the agent.

Ask Greymatter is on every page of the Console. A person types a question in plain language. A model the organization chooses answers it from the platform's own data, through read-only tools.

For AI governance, it reads token budgets and use, each MCP Router and its calls, and the Guardrails findings. Only an administrator sees who used the most tokens.

Tools come in groups: Health, Security, Traffic, Changes and Docs. Health and Docs are on by default. A person turns on the others in the Tools menu.

Questions a team can ask

"Who used the most tokens in the analytics namespace today?"
`Traffic` · token budgets and today's use

"Which MCP tool calls did the agents router refuse today, and why?"
`Security` · one MCP Router's calls and decisions

"Who called the ticketing server's tools on the agents router in the last hour?"
`Security` · each call's caller, tool and granting role

"Is the MCP Router on the agents service healthy, and can it reach its servers?"
`Health` · the router's servers, status and tools

"Which identities did Guardrails flag today?"
`Security` · Guardrails findings by identity

"How do I set a token budget on an LLM egress?"
`Docs` · the Greymatter documentation

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

Key takeaways

1

One path out

Model calls go through an egress the program declares, and other outbound calls show up by name.

2

The person, not the token

Greymatter carries the verified person to every model and tool. Credentials stay where they belong.

3

Inspection both ways

Guardrails covers the OWASP LLM Top 10, and DLP finds sensitive data in prompts and responses.

4

A budget per person

Daily token budgets by person and group, counted once across every replica.

5

Tools by role

One MCP endpoint, with tools granted by role and limited per person.

6

Evidence without secrets

Every model and tool call names its person in a record that holds no replayable credential.

IN ONE SENTENCE

Greymatter gives every model call and every tool call one governed path. That path carries a verified person, inspection both ways, a budget, and a record that holds no secret.

© 2026 Greymatter.io · Greymatter.io is a trademark of Greymatter.io, Inc.

Greymatter.io

See every call. Inspect every prompt. Prove every control.

LEARN MORE

greymatter.io

Greymatter.io is a trademark of Greymatter.io, Inc.

© 2026 Greymatter.io, Inc.