TEST / CONTROL / AI / MULTIPLIED

SUMMARY REPORT · LLM GATEWAY · GOVCLOUD

An LLM Gateway in AWS GovCloud

One controlled door to every model. What it satisfies in NIST SP 800-171, what it saves, and what it does not cover.

7 BUDGET LEVELS / 13 CONTROLS TOUCHED / 4 TEAM TIERS

LITELLM · AWS GOVCLOUD · DoD CC SRG IL4/IL5 · 5 SEPTEMBER 2026

Alexander's Test & Control Solutions

Huntsville, Alabama · (256) 341-7917 · Billy@AlexanderTestControl.com

alexandertestcontrol.com

SECTION 1

The problem a gateway solves

Without a gateway, every application team that wants a model gets its own credential to Bedrock, writes its own retry logic, picks its own model, and produces its own logs in its own format. Six months later nobody can answer three questions that an assessor will certainly ask: who called the model, what did they send it, and how do you know.

An LLM gateway puts one controlled door in front of every model. Every request carries an identity, hits a budget, obeys a rate limit, is checked against a model allow list, and lands in one audit log with a consistent schema. LiteLLM is the reference implementation used throughout this report because it is open source, it speaks the OpenAI API shape that most tooling already expects, and it fronts Bedrock natively, which is what matters in GovCloud.

> **On the terminology, because it gets used loosely**
> LLM gateway and LLM governance are frequently spoken about as two programmes. In practice the gateway is where the governance is enforced. A governance policy that lives only in a document is a statement of intention. The same policy expressed as a model allow list on a virtual key, a hard budget that refuses the request, and an immutable audit record is a control an assessor can test. Build the document, then build the gateway that makes the document true.

SECTION 2

Architecture in GovCloud

Nothing exotic. The deliberate choices are that no component reaches the public internet and that the audit trail is written somewhere the application team cannot alter.

Component	Service	Why this one
Gateway runtime	ECS Fargate behind an internal ALB	No instances to patch, scales to zero traffic overnight, and Fargate keeps the compute inside the assessment boundary without a host hardening argument
Spend and key store	RDS PostgreSQL, Multi-AZ, encrypted with a customer managed key	LiteLLM requires PostgreSQL for spend tracking. This database holds your entire attribution record, so treat it as evidence, not as a cache
Model access	Bedrock via VPC interface endpoint	Traffic never leaves the AWS network. Confirms the boundary claim in your SSP rather than asserting it
Identity	IAM Identity Center federated to your IdP	Named humans, not shared keys. This is the control that makes 3.3.2 satisfiable
Secrets	Secrets Manager with customer managed keys	Provider credentials never sit in a task definition or a config file
Audit sink	CloudWatch Logs to S3 with Object Lock, separate account	Write once, and outside the account the application team administers. An audit log a developer can delete is not an audit log
Region	us-gov-west-1	The West region carries the model catalogue. See the companion GovCloud report for why East is not a failover

Everything above sits in private subnets with no internet gateway. The only ingress is the internal ALB, reachable from your corporate network over Direct Connect or a VPN. If a component in your design needs egress to the public internet, that is the component to question.

SECTION 3

The control surface

This is what you are actually buying by putting a gateway in the path. Budgets can be enforced at seven levels simultaneously, and the most restrictive one wins.

Level	Endpoint or config	Parameter
Global proxy	`config.yaml`	`max_budget, budget_duration`
Team	`/team/new, /team/update`	`max_budget`
Team member	`/team/member_update`	`max_budget_in_team`
Internal user	`/user/new`	`max_budget`
Virtual key	`/key/generate`	`max_budget`
Per model	`enterprise feature`	`model_max_budget`
End customer	`/budget/new, /customer/new`	`user on the request`

Rate limits use tpm_limit, rpm_limit and max_parallel_requests on the same hierarchy, with model_rpm_limit and model_tpm_limit for per model control. Budget periods are set with budget_duration, which accepts values in the form 30s, 30m, 30h and 30d. Spend is written to the LiteLLM_VerificationToken table and read back through /key/info and /user/info.

> **The one that catches people**
> If budget_duration is left unset, the budget never resets. A team hits its monthly ceiling in week two, and then stays blocked in week five, in week nine, and until somebody manually intervenes. Set the duration on every budget you create, and put a test in your deployment pipeline that fails if a budget exists without one.

SECTION 4

What you offer the teams

A gateway lets you publish a small menu instead of answering the question individually every time somebody asks for model access. Four tiers cover almost every internal request.

Tier	Models	Budget and limits	Data	Who gets it
Explore	Small and mid tier only	$50 per user per month, 30d reset, low rpm	Non-CUI only	Anyone, self service, approved same day
Build	Full authorized catalogue	$500 per team per month, 30d reset	Non-CUI only	A named team with a named owner and a stated use case
Production	Pinned model versions only	Set from measured load, alerting at 80 percent	CUI permitted, boundary determination on file	A workload that has been through review and has an owner on call

Tier	Models	Budget and limits	Data	Who gets it
Restricted	One model, one version	Hard ceiling, low parallelism	CUI and export controlled	Programme specific work under an explicit written approval

Two rules make the menu work. First, the tier is enforced on the virtual key, not written in a wiki, so nobody can quietly use a model their tier does not include. Second, moving between tiers is a request with a named approver and a record, which means the SSP can describe the process and the log can evidence it.

The Explore tier matters more than it looks. A self service path with a small budget and a non-CUI restriction is the single most effective shadow AI control available to you. People use consumer chatbots because asking for access is slow. Make asking fast and the reason evaporates.

SECTION 5

What this satisfies in NIST SP 800-171

A gateway is not a compliance product and nobody should sell it as one. It does, however, materially contribute to thirteen requirements, and being able to name them is the difference between an assessment conversation that takes twenty minutes and one that takes a day.

Requirement	Contribution
3.1.1 Limit access to authorized users	Virtual keys are issued to named users and teams. No key, no model
3.1.2 Limit access to authorized transactions and functions	Per key model allow lists mean a key can reach some models and not others
3.1.5 Least privilege	The tier structure in section 4 is least privilege expressed as configuration
3.1.12 Monitor and control remote access	Internal ALB only, reachable from the corporate network, logged per request
3.3.1 Create and retain audit logs	Every request logged with identity, model, token counts and cost, shipped to write once storage
3.3.2 Trace actions to individual users	The reason keys are per person rather than per application. This is the requirement most homegrown setups fail
3.3.5 Correlate audit review and reporting	One log schema across every provider, which is what makes correlation possible at all
3.4.2 Enforce security configuration settings	Gateway config held in version control and deployed through a pipeline, not edited in a console
3.5.1 Identify users and processes	Federated identity through IAM Identity Center rather than shared secrets
3.5.2 Authenticate identities	Authentication happens at the IdP with your existing MFA policy
3.13.1 Monitor and control communications at boundaries	The gateway is the boundary. Private subnets, VPC endpoint to Bedrock, no public egress
3.13.8 Cryptographic protection of CUI in transit	TLS to the gateway, TLS to Bedrock over the VPC endpoint, and the traffic never traverses the internet

Requirement	Contribution
3.14.6 Monitor the system and its traffic	Prompt and response volume, error rates and spend anomalies are all visible in one place

SECTION 6

What it does not cover

It does not	So
Authorize a model for CUI	Model availability and model authorization are separate facts. Keep the per model authorization register described in the companion GovCloud report, and enforce it through the allow list
Stop a user pasting CUI into a non-CUI tier	The tier is a control on the model, not on the human. You still need the written policy, the training, and ideally a guardrail that inspects content
Satisfy the other 97 requirements	800-171 has 110. Thirteen is a real contribution and it is still twelve percent
Give your application an ATO	AWS holds the IL5 provisional authorization for the infrastructure. Your system still needs its own authorization or its place inside your customer's boundary
Replace human review	Anything customer facing, contractual or safety related still needs a person. Put that in the policy rather than leaving it to judgement
Survive being bypassed	If application teams can still reach Bedrock directly with their own IAM role, the gateway is decorative. Deny bedrock:InvokeModel everywhere except the gateway's task role, with an SCP, and test it

The last row is the one that actually fails in practice
Every gateway deployment that quietly stopped working failed the same way: somebody needed something the gateway did not do yet, got a direct Bedrock permission as a temporary exception, and the exception outlived the reason. Six months later half the traffic is invisible again. Write the service control policy on day one, before the first team onboards, because retrofitting it means taking capability away from people who already have it.

SECTION 7

What it saves

A gateway pays for itself in four distinct ways, and only the first is the one people expect.

Saving	Mechanism	Typical scale
Model right sizing	Route extraction, classification and routing traffic to a small model instead of a frontier model. The gateway is where that routing lives, so it can be changed without touching application code	80 to 95 percent on the traffic that moves, and it is usually most of the traffic
Prompt caching applied consistently	Cached input bills at roughly a tenth of the standard input rate. Doing it once at the gateway beats hoping six teams each remember	Up to 90 percent of the cached portion

Saving	Mechanism	Typical scale
Runaway prevention	A hard budget refuses the request. Without one, a looping agent spends until somebody notices, which is typically Monday	Uncapped. This is insurance, not efficiency
Shadow subscription consolidation	Individual tool subscriptions bought on expense cards disappear once there is a fast internal path	$20 to $60 per person per month, on however many people were doing it

A worked figure

Forty engineers, ten model interactions each per working day, roughly 8,000 tokens of retrieved context and 700 tokens of output per interaction. That is about 640,000 input and 56,000 output tokens a day. Run entirely on a frontier model at GovCloud rates, inference lands near $150 a month. Route the seventy percent of that traffic which is extraction and summarisation to a small model, cache the fixed preamble, and the same workload runs under $40. The gateway infrastructure, meaning Fargate, a Multi-AZ RDS instance, the load balancer and the logging, costs roughly $300 to $500 a month in GovCloud.

Read that honestly. At forty engineers the gateway costs more than the inference it optimises. It is not justified by token savings at that scale and nobody should pretend otherwise. It is justified by the audit trail, the budget ceiling and the ability to answer an assessor. The token savings become the dominant argument somewhere north of about two hundred users or when a single production workload starts running continuously.

SECTION 8

Time saved for engineering and software teams

The evidence here is genuinely mixed and any document that shows you only the favourable half is selling something.

Finding	Source	What it measured
55.8 percent faster task completion. 2h41m down to 1h11m, success rate 70 to 78 percent	GitHub Copilot controlled experiment, arXiv 2302.06590	One scoped task, developers new to the problem
19 percent slower, while believing they were 20 percent faster	METR randomized controlled trial, arXiv 2507.09089	16 experienced developers, 246 tasks, repositories they already knew well. Design revised February 2026
66 percent report more time fixing almost right code	Stack Overflow Developer Survey 2025	Self reported across the 51 percent who use AI daily
Review time up as much as 91 percent under high adoption	Published DORA-aligned analyses	Pull request volume rises and the bottleneck moves to review

The pattern underneath the contradiction

The two randomized trials do not actually disagree. Copilot measured developers working on something unfamiliar, where the model supplies knowledge they lack. METR measured experts working in codebases they know intimately, where the model supplies plausible output they then have to verify, and verification costs more than recall. Gains concentrate where familiarity is low. Losses concentrate where familiarity is high. That is one coherent finding, not two conflicting ones, and it tells you exactly where to point the tool.

Where a test and control organisation actually gains

Workflow	Why it is a good target
Test report and ATP first drafts	Structured, repetitive, heavily templated, and the engineer's time goes into review rather than composition. Low risk because a human signs it either way
Requirements to test traceability	Mechanical cross referencing that people do badly because it is tedious. The model is good at it and the output is checkable
Legacy code comprehension	Explaining twenty year old LabVIEW or TestStand to the engineer who inherited it. Maximum unfamiliarity, which is where the gains are
Instrument driver and script scaffolding	Boilerplate against a documented API. Verifiable by running it
Retrieval over procedures and program documentation	Removes the interruption cost of one engineer asking another a question the document already answers. Rarely measured, consistently real
Deep changes in a codebase the engineer wrote	The METR case. Expect no gain and possibly a loss. Do not mandate it here

How to measure it so the answer means something

Pick two or three workflows. Baseline the hours in writing before the tool arrives, using the team's own time records rather than an estimate. Re-measure at ninety days on the same basis. Report the result whichever way it comes out. A programme that only reports favourable findings cannot be trusted on the favourable ones either, and the gateway gives you the usage data to tie the measurement to actual model use rather than to a survey.

SECTION 9

Standing it up

#	Phase	Elapsed	What happens
0	**Boundary and policy**	Wk 1-3	Data classification, acceptable use policy, tier definitions from section 4, and the model authorization register. Paper before infrastructure
1	**Landing zone**	Wk 2-5	VPC, private subnets, Bedrock VPC endpoint, RDS, Secrets Manager, the separate logging account, and the SCP that denies direct Bedrock access
2	**Gateway deployment**	Wk 4-6	Fargate service, internal ALB, config in version control, model allow lists, guardrails, budgets with durations set
3	**Identity and onboarding**	Wk 5-7	IdP federation, first two teams onboarded onto Explore and Build tiers, key issuance process documented
4	**Evidence**	Wk 6-9	Architecture diagram, data flow, the control mapping in section 5, SSP language, POA&M; entries for anything open

#	Phase	Elapsed	What happens
5	**Production workload**	Wk 8-14	The first real application on the Production tier, with pinned model versions and measured load

Six to seven weeks to a gateway two teams are using, fourteen to a production workload behind it. Add three to four weeks if the GovCloud account itself does not exist yet, because the US persons screening in that process is outside your control.

SSP language you can lift

All large language model inference is brokered through a centrally managed gateway deployed within the [system name] boundary in AWS GovCloud (us-gov-west-1). Direct invocation of Amazon Bedrock is denied by service control policy for all principals except the gateway execution role. Access is granted through per user virtual keys issued under federated identity, scoped to an approved model allow list and a spend budget with a defined reset period. Every request is recorded with the authenticated user identity, model identifier, token counts and computed cost, and forwarded to write once storage in a separate logging account. Only models with documented FedRAMP and DoD Impact Level authorization are included in allow lists permitting Controlled Unclassified Information.

Adjust the system name and the region, confirm each sentence is true of what you actually built, and have your ISSO review it. A paragraph that describes something you did not build is worse than no paragraph.

SECTION 10

Sources

What	Where
LiteLLM budget hierarchy, rate limit parameters, budget_duration formats, PostgreSQL spend tracking	docs.litellm.ai/docs/proxy/users
AWS GovCloud regions, FedRAMP High, DoD IL2/IL4/IL5, ITAR, US persons screening	aws.amazon.com/govcloud-us
Bedrock model availability and authorization in GovCloud	See the companion report, Running AI in AWS GovCloud
Bedrock cost allocation by IAM user and role, April 2026	aws.amazon.com/about-aws/whats-new/2026/04/bedrock-iam-cost-allocation
NIST SP 800-171 Rev 2 requirement text and numbering	csrc.nist.gov, NIST SP 800-171 Rev 2 and 800-171A
DFARS 252.204-7012 and the FedRAMP equivalency requirement	acquisition.gov
DoD Cloud Computing SRG impact level definitions	public.cyber.mil/dccs/dccs-documents
55.8 percent faster task completion	Peng et al., The Impact of AI on Developer Productivity: Evidence from GitHub Copilot, arXiv 2302.06590
19 percent slowdown for experienced developers, February 2026 design revision	METR, arXiv 2507.09089, and metr.org
51 percent daily use, 66 percent fixing almost right code	Stack Overflow Developer Survey 2025
Prompt caching and batch discount rates	Anthropic and OpenAI published pricing documentation, 2026

What is measured here and what is judgement

Measured: the LiteLLM parameter names and budget hierarchy, the GovCloud authorizations, the 800-171 requirement text, and every study result in section 8. Judgement: the architecture choices in section 2, the four tier structure in section 4, the mapping of gateway features to specific 800-171 requirements, the cost figures in section 7, and the phase durations in section 9. The control mapping in particular is a defensible reading rather than an official one, and your assessor's opinion outranks mine.