From Volume to Voice: AI in the Senator's Office

A nine-panel briefing for chiefs of staff, Senate IT leadership, and federal contractors. What's now possible — built, demonstrated, and ready to replicate across all 100 Senate offices.

AWARENESS BRIEFING | SENATE AI INFRASTRUCTURE | REFERENCE ARCHITECTURE

PANEL 1

The Volume Problem

"A senator's office is drowning in signal."

A large-state Senate office is not a communications operation — it is a signal-processing organization that also does communications. Every week, constituent letters arrive by the tens of thousands. Every legislative day, dozens of bills enter the hopper. Every hour, calls, press inquiries, casework submissions, and social media mentions accumulate faster than any staff rotation can absorb. The math is unforgiving: a congressionally-mandated budget cap holds most offices to roughly 60 staff, while the volume of inbound data has grown continuously for two decades.

Senator John Doe of Maryland — a fictitious large-state office used throughout this briefing — offers a clear illustration. Across 24 months, his office processed 60,000 casework cases, 150,000 phone calls, 36,000 meeting requests, 12,000 press inquiries, 750,000 constituent emails across 100 advocacy campaigns, and 120,000 social media mentions. Approximately 85 staff handle all of this across one DC office and four state offices.

25K

Letters per Week

Pieces of constituent correspondence per week in big-state offices — Congressional Management Foundation

30+

Bills per Day

Bills introduced every legislative day on average — Congress.gov

~60

Staff Cap

Maximum staff positions under SOPOEA budget rules for most Senate offices

750K

Constituent Emails

Emails processed across 100 advocacy campaigns in Senator Doe's office over 24 months

The problem is not that staff are not working hard enough. The problem is that the data streams — constituent letters, phone logs, social mentions, press inquiries, legislative filings — all flow in through separate systems, are handled by separate teams, and are never integrated into a unified institutional record. Every piece of signal is touched once and then lost.

PANEL 2

The Hidden Cost

"Most of this data is read once, then disappears."

Institutional memory is the invisible asset that makes a Senate office effective — and it is being destroyed quietly, every day, by fragmented systems and staff turnover. A caseworker handling a complex Social Security appeal has no easy way to know whether a nearly identical case was resolved three months ago by a colleague in the Baltimore district office. A press secretary drafting a response to a breaking infrastructure story cannot quickly surface the senator's prior floor statements on highway funding. A legislative staffer briefing a floor vote on agricultural appropriations starts from scratch, unaware that the office has already developed detailed position documents on five of the seven amendments in play.

Casework — Lost Precedent

Caseworkers cannot find similar prior cases. Every resolution is reinvented. Agency procedures are re-researched from scratch. Constituents wait longer than necessary because the institutional answer already exists — buried in a closed ticket from 18 months ago.

Press — Ungrounded Drafts

Press secretaries draft reactive statements without surfacing the senator's prior public positions, floor speeches, or committee testimony. The risk of inconsistency — and the hours spent manually cross-referencing — is constant and largely invisible to leadership.

Legislative — Cold Briefings

Legislative staff brief floor votes from scratch, even when the office has previously analyzed identical or closely related legislation. Prior positions, constituent pressure data, and cosponsor analysis are not systematically preserved or retrievable.

The institutional memory exists. It lives in inboxes, in PDFs on shared drives, in email threads between staffers who have since left the office, and in case management systems that were never designed to be queried analytically. The problem is not a lack of data — it is a lack of architecture to make that data useful across time, across staff, and across offices.

PANEL 3

The Shift

"AI on the senator's own hardware, inside the Senate enclave."

The architectural answer is not a cloud subscription, a vendor-managed SaaS platform, or a shared federal AI service. Each of those options introduces external data egress for constituent personally identifiable information — an unacceptable risk profile under Senate data governance expectations and constituent trust obligations. The answer is local inference: AI compute that runs entirely inside the Senate network enclave, on hardware the office controls, with no data leaving the building.

Linux File Server

Commodity server-grade hardware running PostgreSQL with pgvector, ingest pipelines for correspondence and casework, Streamlit applications for staff-facing interfaces, and Ollama for model serving. All open-source. All AOC-approved stack components.

- PostgreSQL + pgvector database
- Ingest pipelines (correspondence, casework, press, social)
- Streamlit staff applications
- Ollama model serving layer

NVIDIA GB10 AI Compute

A single NVIDIA GB10 Superchip providing 128GB of unified memory — enough to run the full model stack simultaneously. No GPU cluster. No rack of servers. One purpose-built compute unit priced within the bounds of a standard MRA hardware procurement.

- 128GB unified memory
- Full model stack: ~50GB in use
- 78GB headroom for expansion
- Single unit, MRA-budget priced

ⓘ No cloud. No external data egress for constituent PII. The entire system runs on MRA budget hardware, inside the Senate enclave, under the office's direct administrative control.

This is not an experimental configuration. The Linux server plus GB10 combination is a production-capable architecture. Both units sit inside the Senate's existing physical and network enclave. Constituent data never traverses a public network. The office retains full custody of every record it processes.

PANEL 4

The Model Stack

"Three models, one box, no cloud."

Running large language models locally requires careful memory management. The GB10's 128GB unified memory is the enabling constraint: it is large enough to hold multiple models simultaneously, but finite enough that model selection and quantization matter. The stack below occupies approximately 50GB of that memory, leaving 78GB of headroom for additional models, vector indexes, and database operations.

1

BGE-M3 — Embeddings & Semantic Search

The foundation layer. BGE-M3 converts all ingested text — correspondence, casework records, press transcripts, bill text — into dense vector representations stored in pgvector. Powers all semantic search: "find cases similar to this one," "surface prior statements on this topic." Lightweight, fast, always running.

2

Qwen 2.5 14B Instruct (FP8) — Triage & Classification

The workhorse for high-volume, lower-complexity tasks. Classifies incoming correspondence by topic, urgency, and routing destination. Runs batch jobs overnight. Handles the 25,000 letters per week triage pipeline without saturating the compute budget. Cheap calls go to the small model — this is the small model.

3

Llama 3.3 70B Instruct (FP4) — Drafting, Briefings & Formal Correspondence

The senior model. Reserved for tasks requiring judgment, nuance, and length: full draft constituent responses, bill one-pagers, hearing question rounds, press response frameworks. FP4 quantization fits the 70B model within the GB10's memory budget without material quality loss at these task types. Expensive calls go to the big model — this is the big model.

Total stack memory footprint: ~50GB on the GB10's 128GB unified memory. Routing logic is automatic: triage tasks call Qwen 2.5 14B; drafting and briefing tasks call Llama 3.3 70B. Both run locally, simultaneously, with no cloud dependency.

PANEL 5

Six Use Cases at a Glance

The model stack and database architecture are not ends in themselves. They are infrastructure for a specific set of staff-facing capabilities — six use cases, each mapped to a real workflow pain point in a large-state Senate office. None of these use cases require staff to learn a new system from scratch. Each is designed to surface inside the tools and workflows staff already use.

Correspondence Triage

25,000 letters per week classified by topic, urgency, and routing destination. Draft responses generated for standard issue types. Staff review and approve — they do not start from blank.

Bill Briefing Generator

Every floor vote triggers a one-pager: state impact, the senator's prior votes on similar legislation, and a cosponsor map. Legislative staff receive structured context, not a blank page.

Hearing Prep

Witness statements paired with the senator's prior committee questions and floor remarks. Draft question rounds generated for staff review. Preparation time cut from days to hours.

Casework Copilot

Semantic search across all prior cases. Surface similar resolutions in seconds. Relevant agency procedures retrieved automatically. Constituents get faster, more consistent responses.

Press Response

Inquiries automatically paired with the senator's voting record and prior public statements. Press staff draft from a grounded context package, not from memory or a manual archive search.

District Intelligence

Emerging issue detection from constituent contacts and local press monitoring. Issues surface in the office's awareness before they reach national media — enabling proactive rather than reactive response.

PANEL 6

Senate Office Analytics

"Ask anything. Get real answers."

The analytic interface is the showpiece capability — the one that makes the scale of the underlying data infrastructure immediately legible to any chief of staff or senior staffer who sits down with it. Built on Vanna (vanna-ai/vanna, MIT license) over the same PostgreSQL database that powers all other use cases, the interface accepts plain English questions and returns SQL queries, charts, and grounded prose summaries. Staff do not need to know that SQL is happening. They type a question. They get an answer.

"Top 5 issues constituents wrote about in Q1 2025?"

Returns a ranked list with volume counts, trend direction versus Q4 2024, and a geographic breakdown by Maryland county. Generated in seconds from the correspondence database.

"Which MD counties generate the most casework?"

Returns a county-level bar chart with case volume, average resolution time, and the top three issue categories per county. Actionable for state office resource allocation decisions.

"Show unresponded letters from Baltimore mentioning VA benefits"

Returns a filtered record set with constituent names, submission dates, and letter summaries. Staff can immediately queue these for response drafting. Constituent services becomes proactive.

"Draft a weekly constituent services briefing"

Returns a structured prose briefing — top issues, open cases by age, resolved cases this week, emerging signals — ready for chief of staff review. A weekly task that previously took hours is complete in under two minutes.

Plain English in. SQL, charts, and grounded summaries out. The Vanna layer learns the office's specific schema and terminology over time, improving accuracy with use. No data scientist required on staff.

The analytics interface is not a reporting dashboard with pre-built charts that someone decided in advance were the right questions. It is a queryable institutional record that staff can interrogate in real time, with their own questions, in their own words. This is the difference between a data warehouse and a working knowledge system.

PANEL 7

Built, Not Slideware

"The Senator John Doe Demonstration"

Everything described in this briefing has been built and is queryable today. The Senator John Doe demonstration environment is a fully operational instance of the reference architecture, populated with synthetic Maryland office data and real Congressional bill data, running on the GB10 hardware configuration described above. This is not a prototype in the sense of a proof-of-concept that explores whether something might work. It is a production-candidate deployment that demonstrates exactly what a live office environment would look like.

488K

Synthetic Records

Synthetic Maryland office data records loaded and queryable — correspondence, casework, press inquiries, phone logs, social mentions, meeting requests. No real constituent PII.

68,716

Real Bills Loaded

Real Congressional bills already ingested from the Congress.gov public data pipeline, fully indexed and semantically searchable across all six use cases.

Live

Congress.gov Pipeline

The Congressional data pipeline is active — new bills ingested and indexed automatically as they are introduced. The demonstration environment stays current with the legislative calendar.

What "Synthetic" Means

The constituent data in the demonstration environment is entirely synthetic — generated to reflect realistic volume, issue distribution, geographic spread, and casework complexity for a large Maryland Senate office, but containing no real names, addresses, case numbers, or personally identifiable information of any kind. The Congressional bill data is real, public, and drawn directly from Congress.gov's official data API under its standard terms of use.

The distinction matters for two reasons. First, it means the demonstration can be shown to any audience — including contractors and external briefing participants — without any data governance concern. Second, it means the demonstration accurately represents the performance characteristics of a real deployment: the synthetic data was designed to stress-test the architecture at realistic scale, not to make it look easy.

Database Tables Active

- constituent_correspondence
- casework_records
- press_inquiries
- phone_call_logs
- social_media_mentions
- meeting_requests
- congressional_bills
- bill_embeddings (pgvector)

PANEL 8

Built for the Senate Environment

The reference architecture was designed from the outset against the actual constraints of the Senate operating environment — not retrofitted to meet them after the fact. Three design principles govern every component selection, data flow decision, and infrastructure choice in the stack.

Three-Zone Data Segmentation

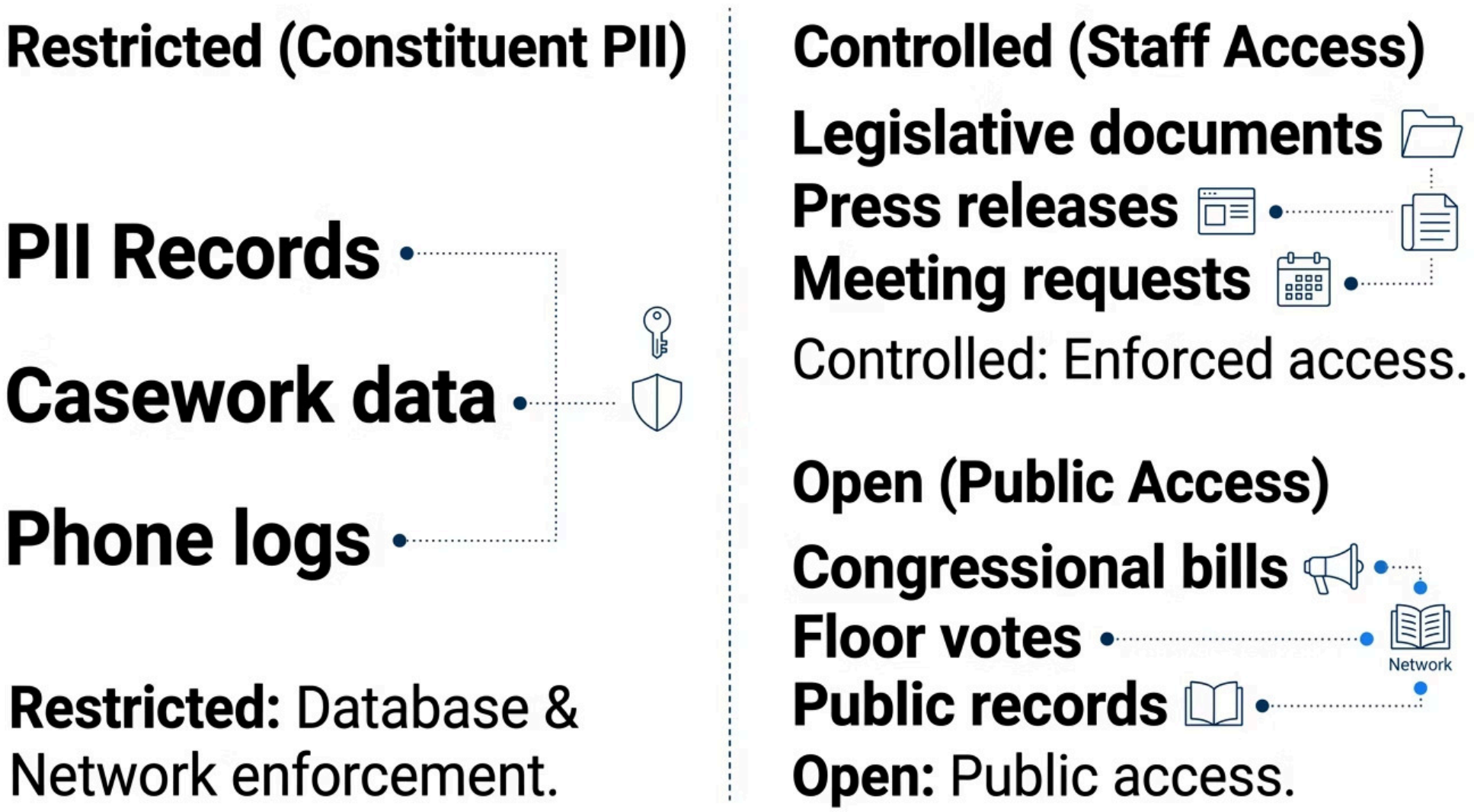
Constituent PII is classified and stored in a restricted zone that never touches public-facing or contractor-accessible services. Legislative and press data occupy a second zone with role-based access controls. Public Congressional data — bill text, floor votes, committee records — occupies an open zone accessible to all staff applications. Zone boundaries are enforced at the database and network layer, not by policy alone.

100% Open-Source Stack

Every component in the stack — PostgreSQL, pgvector, Ollama, Streamlit, Vanna, the Llama and Qwen model weights — is open-source software under permissive licenses (MIT, Apache 2.0, or equivalent). The full stack is compatible with the AOC-approved software list. There are no vendor lock-in dependencies, no per-seat licensing costs, and no subscription renewals. The office owns the deployment entirely.

On-Premise Inference

All AI inference — every model call, every embedding generation, every SQL query synthesized from natural language — runs on hardware physically located inside the Senate enclave. No constituent data, no draft correspondence, no casework detail, and no staff query ever traverses a public network or reaches an external API. The system functions completely offline if required. Cloud dependency is zero.



PANEL 9

What Comes Next

"Reference architecture, ready to replicate."

The Senator John Doe demonstration is not the destination — it is the blueprint. The reference architecture was designed explicitly for replication across all 100 Senate offices, with a deployment timeline and capital cost that are achievable within existing MRA budget frameworks and without requiring specialized IT staff in every office. The architecture is not bespoke. It does not require customization for each office from scratch. It is parameterized: office name, state, district configuration, and data source connections are configuration variables, not architectural variables.

1 Weeks 1–2: Infrastructure

Hardware procurement, network placement inside the Senate enclave, base OS and software stack installation, initial database schema deployment.

2 Weeks 3–4: Data Ingest

Connection to the office's existing correspondence, casework, and press systems. Initial data load and embedding generation. Congressional bill pipeline activation.

3 Weeks 5–6: Staff Onboarding

Working pilot with real staff on real workflows. Correspondence triage, bill briefings, and the analytics interface are the Day 1 capabilities. Additional use cases activated on the office's schedule.

4–6

Weeks to Pilot

From hardware delivery to a working staff pilot with real office data and all six use cases active.

$6–9K

Capital per Office

Commodity Linux server plus NVIDIA GB10. Achievable within standard MRA hardware procurement budgets. No ongoing subscription costs.

100

Senate Offices

The same stack, without modification, deploys into all 100 Senate offices. One reference architecture serves the entire chamber.

> The mission is not to build AI for one senator's office. The mission is to build AI infrastructure for the United States Senate — and to demonstrate that it can be done securely, affordably, and without waiting for a federal program office to lead the way.

Mission. People. Outcomes.

Prepared by DMI / ITSC-V for Senate IT leadership, chiefs of staff, and federal contractors.

All constituent data in the demonstration environment is synthetic. Congressional bill data is real and public.