OFFICE TIMESHEETS

AI Timesheet Assistant

Security & Compliance Overview

Lookout Software, LLC.

Contents

1. Purpose and Scope

This document describes the security architecture, data-handling practices, and compliance posture of the AI Timesheet Assistant in Office Timesheets (OTS). It is intended for customer IT, security, and compliance teams evaluating how artificial intelligence is applied to their organization's time entry and time off workflows.

The AI Timesheet Assistant is a conversational interface through which employees enter time, review leave balances, and submit time off requests. Its AI involvement is deliberately narrow:

- **Sentence parsing** — when a user types a free-text sentence such as “8 hours on Acme yesterday”, a language model classifies the intent and extracts the stated details (description, hours, day words) into a fixed structured form.
- **Everything else is conventional software.** Guided (button-driven) entry, task matching, leave balance display, and the entire time off request flow involve no language model at all. Balance figures come from the OTS database; time off questions are answered by taps or code-parsed text.

A central design principle underpins the assistant: the language model never touches the database, never decides what is saved, and never produces a number that is written anywhere. Its sole role is converting a typed sentence into a structured suggestion. Office Timesheets validates that suggestion in code, the user reviews the result on a confirmation card, and only an explicit user confirmation triggers a write — which then passes through the same validation pipeline as the standard entry forms.

2. Executive Summary

- **Minimal data exposure.** Only the user's typed message, a boolean conversation flag (whether a proposal is pending), and the current date/week are sent to the model. No employee records, no task lists, no balances, no credentials, and no database access of any kind.
- **The model never writes and never calculates what is saved.** It emits a structured suggestion in a closed vocabulary (a fixed intent list; symbolic day words; stated hours echoed, never invented). OTS validates every field in code; anything out of shape or out of bounds becomes an unanswered question to the user, never a value.
- **A human confirms every write.** Nothing is saved without the user explicitly confirming a card that displays exactly what will be created. Confirmed writes run through the standard OTS pipeline — permissions, period locks, required notes, validation, and approvals apply unchanged.
- **Certified configuration.** The exact model, prompts, and settings in production were certified against a versioned evaluation suite that includes adversarial and prompt-injection cases, with zero-tolerance gates on safety behaviors. Prompt changes require re-certification, and the certified prompts ship as a release artifact.
- **Zero-retention inference.** Inference runs on the Ollama harness. Prompt and response data is never logged, never retained, and never used to train models.
- **Graceful degradation, optional AI.** If the model is unavailable, slow, or returns anything unexpected, the assistant silently falls back to its fully functional button-driven mode. Installations may disable AI parsing entirely; every feature remains usable with zero inference.
- **User and tenant isolation.** The assistant is available only to authenticated OTS users; conversations are private to the user's sign-in; all data access is scoped by existing OTS permissions.

3. Architecture Overview

The assistant's architecture is built on four pillars:

- **A dialog engine that owns the conversation.** All questions, state, and decisions live in Office Timesheets application code. The model is consulted only when the user types free text outside a structured question.
- **Structured suggestion via certified prompts.** The model receives the typed sentence and a strict system prompt, and emits a fixed JSON structure — an intent from a closed list, and extracted fields (work description, hours, symbolic day words). It never emits SQL, dates, record identifiers, or commands.
- **Code-side validation and resolution.** OTS validates the structure and every value: intents against the closed list, hours against sane bounds, day words against a fixed vocabulary resolved to calendar dates in code. Any deviation nullifies the field — the assistant then asks the user instead.
- **One write path, behind human confirmation.** All saves — time entries and time off requests alike — go through a single committer that re-validates permissions, locks, and business rules at commit time. The trigger is always the user's explicit confirmation of a card showing the exact values.

Three trust boundaries separate the system: the customer boundary (the user's browser and typed input), the Office Timesheets platform boundary (dialog engine, validation, task matching, committers, and database), and the inference boundary (the Ollama harness, whether OTS-managed or customer-hosted). The only data that crosses into the inference boundary is the typed sentence with minimal context; the only thing that comes back is a structured JSON suggestion.

4. How the AI Feature Works

4.1 What the model is asked

When a user types free text, the assistant makes up to two model calls. The first classifies intent — is this a time entry, a confirmation, a cancellation, an edit, or something else — returning one value from a closed list plus a confidence flag. If (and only if) the intent is time entry, a second call extracts what the sentence states: the user's own words for the work, any stated hours or day fractions, and day words like “yesterday” or “last tuesday” as symbolic tokens. The model is explicitly instructed, and structurally constrained, never to invent items, hours, or days that are not in the message.

4.2 What Office Timesheets does with the answer

- Every field is validated in code: unknown intents, malformed structures, out-of-range hours, or unrecognized day tokens are discarded (the assistant asks the user instead).
- Symbolic day words are resolved to calendar dates by OTS code against the current date — the model never produces a date.
- The work description is matched to the user's assigned tasks by deterministic OTS code (learned aliases, lexical scoring, usage history). Ambiguity produces a question to the user, never a guess that is silently saved.
- The completed suggestion is presented on a confirmation card. The user can edit any detail, cancel entirely, or confirm — and only confirmation writes, through the standard OTS validation pipeline.

The consequence of this design is that the model's output determines only what the assistant asks or proposes — never what is saved. A wrong or malicious model output can, at worst, produce a proposal the user sees, edits, or cancels; it cannot write data, skip validation, or bypass approvals.

4.3 What involves no model at all

Guided entry (buttons and pickers), typed answers to specific questions (hours, notes, times, task names), leave balance display, and the entire time off request flow are conventional application code. Balance figures are read from the OTS database with the same permission checks as the My Accruals page; time off requests are

assembled from tapped choices and code-parsed text and saved through the standard time off pipeline, including its approval process.

5. Data Handling

5.1 What Is and Is Not Sent to the Model

Data category	Sent to the LLM?	Notes
The user's typed free-text message	Yes	Only when free text is typed outside a structured question
Whether a proposal is pending (true/false)	Yes	One boolean of conversation context
Current date and reporting week	Yes	So day words can be interpreted; contains no user data
Employee records, task lists, balances	Never	Task matching and balance display are performed by OTS
Database credentials / connection strings	Never	The model has no data-access capability
Authentication tokens, passwords	Never	Outside the scope of prompt assembly
Raw database tables or tenant datasets	Never	The model receives no records of any kind

Because prompts contain only the typed sentence and calendar context, the personal data crossing the inference boundary is limited to whatever the user chooses to type. Notes, titles, and answers to specific questions (hours, times, task names) are captured directly by Office Timesheets and are not sent to the model.

5.2 Data in Transit and at Rest

All traffic between the user's browser, the Office Timesheets platform, and the inference layer is encrypted in transit. The assistant stores its conversation transcripts, proposals, and learned task shorthand in the customer's own Office Timesheets database — alongside the timesheet data itself, under the customer's existing retention and backup controls. No new data store is introduced at any third party. Assistant-created entries and requests are marked as such for audit purposes, and each proposal records exactly what was confirmed and what was created.

5.3 Data Retention at the Inference Layer

Inference requests are processed transiently: the prompt is used to generate a response and is then discarded. Prompt and response data is never logged and is never used for model training. Where Ollama partners with NVIDIA Cloud Providers to host models, no-logging, no-training, and zero-data-retention policies are contractually required.

6. AI Guardrails: Structure, Validation, and the Human Gate

Where an AI feature could act on data, it needs boundaries. The assistant implements them in three independent layers:

- **Closed vocabulary.** The model can only classify into a fixed intent list and echo details stated in the message. Day words come from a fixed token set; hours are bounded; there is no free-form escape hatch, no record identifiers, and no expressible command of any kind.

- **Code-side enforcement.** Every model response is parsed and checked before it influences anything. Out-of-shape or out-of-bounds output is not “best-effort” interpreted — the affected field is discarded and the assistant asks the user. Model output is treated as data, never as code: no SQL, no expressions, no commands originate from it.
- **The human gate and the single write path.** No write occurs without the user confirming a card that displays the exact values. Confirmed writes pass through one committer that re-validates the user's permission, period locks, required notes and times, and all business rules at commit time — identical to the standard entry forms. Time off requests additionally follow the normal approval process; the assistant creates requests and never approves them.

Defense in depth, from the outside in: transport and provider controls (TLS, zero-retention inference); data minimization (only the typed sentence crosses the boundary); structural constraint (closed-vocabulary suggestion); code-side validation (every field checked, fail-safe to a question); the human confirmation; and finally the OTS validation pipeline itself. Six independent layers stand between the language model and the customer's data.

7. Certification: A Tested Configuration, Not a Hopeful One

The assistant's parsing runs one exact, certified configuration: a specific model, specific prompts, deterministic settings, and defined timeouts. Before production use, that configuration was certified against a versioned evaluation suite — dozens of cases across intent classification and extraction, each run repeatedly — with gates that include zero-tolerance families for safety behaviors:

- Adversarial and prompt-injection cases: instructions embedded in user messages (“ignore your instructions and…”) must be treated as text, not obeyed.
- No-invention cases: the model must never fabricate hours, days, or work items not present in the message.
- Bounds cases: absurd values must flow into code-side rejection, never into a saved entry.

The certified prompts ship as part of the release, so production runs exactly the reviewed configuration. Any prompt change requires a full re-certification; the certification report is retained as the regression baseline, and the configuration is re-verified when the model version changes. Latency and failure behavior were measured in certification and set the assistant's timeouts — beyond which it degrades to buttons rather than waiting.

8. Inference Layer: Ollama Harness and Model Options

8.1 The Ollama Harness

- Trusted, US-based harness; servers operated only in the United States, Europe, and Singapore, on modern NVIDIA hardware.
- Prompt and response data is never logged or trained on.
- Collaborates with NVIDIA Cloud Providers (NCPs) to host open models, and requires no-logging, no-training, and zero-data-retention policies from those partners.

8.2 Default Model: Google Gemma 4 – 31B

When customers use the OTS-managed inference option, parsing is performed by Google's Gemma 4 – 31B cloud model — fast and accurate for the narrow structured task the assistant requires, US-based in its hosting, and backed by Google. This is the model the assistant's certification was performed against.

8.3 Deployment Options

Option	How it works	Best for
OTS-managed inference (default)	OTS routes parsing prompts to Gemma 4 – 31B on Ollama's US-based cloud under zero-retention terms.	Customers who want AI features with no infrastructure to run
Customer-hosted Ollama	You connect Office Timesheets to your own Ollama harness, running a cloud model you control.	Organizations with stricter data control policies
AI disabled	AI parsing is switched off; the assistant runs in its fully functional button-driven mode.	Organizations that want the assistant with zero inference

All options use the identical application-side architecture — validation, the human confirmation gate, and the single write path apply regardless of where (or whether) inference runs.

9. User Isolation and Access Control

- Authenticated access only. The assistant's API requires the same authenticated OTS session as the rest of the product, with token-based authentication on every request.
- Private conversations. Each conversation belongs to the signed-in user; conversation identifiers are non-guessable, and ownership is verified on every request.
- Permission-scoped data. Task lists, balances, leave types, and every write are scoped by the user's existing OTS permissions — the assistant can never show or save anything the user's account could not access through the standard pages.
- Audit trail. Entries and requests created by the assistant are marked as assistant-created and linked to the conversation and confirmed proposal that produced them.

10. Threat Model and Mitigations

Threat	Risk if unmitigated	Mitigation in OTS
Prompt injection via typed message	Malicious text tries to make the model exceed its role	Certified adversarial test cases; output constrained to a closed structure; code-side validation of every field; and the human confirmation gate — injected output can at most produce a visible proposal the user rejects
Hallucinated entries	Invented hours or work saved to the timesheet	The model may only echo stated details; no-invention certification gates; bounds fail safe to a question; nothing saves without user confirmation of the exact values
SQL injection via model output	Model-generated SQL executes commands	The model never emits SQL or identifiers; all data access is OTS application code with parameterized queries; model output is treated as data, not code
Destructive operations	AI modifies or deletes records	Model output cannot express a write; the single write path requires an explicit user confirmation and re-validates every rule at commit time
Data leakage to model provider	Timesheet data retained or trained on externally	Only the typed sentence and calendar context are sent; zero-retention, no-logging, no-training inference; customer-

		hosted option keeps prompts fully in-boundary
Cross-user exposure	One user's conversation or data visible to another	Authenticated, ownership-checked conversations with non-guessable identifiers; all reads and writes permission-scoped
Model outage or misbehavior	Feature failure or degraded output	Timeouts and strict validation route every failure to the button-driven mode; no data loss, no error state, no partial writes

11. Compliance Considerations

- **Data minimization and purpose limitation.** Only the typed sentence needed to interpret the request crosses the inference boundary, aligning with data-minimization principles found in frameworks such as GDPR and CCPA.
- **No new data processor accumulation.** Because inference is zero-retention with no logging and no training, use of the assistant does not create a growing copy of personal data at a third party. The assistant's own records (transcripts, proposals) reside in the customer's OTS database under existing controls.
- **Data residency options.** The Ollama harness operates servers only in the United States, Europe, and Singapore, and the default model is US-based Gemma 4 (Alphabet/Google). Customers with stricter requirements can host their own Ollama harness, or disable AI parsing entirely.
- **Human-verifiable outputs.** Every saved value was displayed to and confirmed by a person, then validated by OTS — supporting internal controls that require traceable, attributable records. The audit trail links each assistant-created record to its confirmation.
- **Tested AI behavior.** The certification suite and its retained reports provide documented, reproducible evidence of the model's behavior on safety-relevant inputs — material customers can reference in their own AI governance reviews.

Customers subject to specific regulatory regimes should evaluate the assistant under their own compliance frameworks; Lookout Software can support security reviews with architecture details from this document. This document describes technical design and provider policies and is not legal advice.

12. Shared Responsibility

Party	Responsibilities
Office Timesheets (Lookout Software)	Maintaining the dialog engine, certified prompts and their re-certification discipline, code-side validation, the confirmation gate and single write path, permission scoping, audit marking, and the integration with the inference layer under zero-retention terms.
Inference provider (Ollama and hosting partners)	Operating inference on modern NVIDIA hardware in the US, Europe, and Singapore with no logging, no training on user data, and zero data retention.
Customer	Managing user accounts, permissions, and OTS configuration (locks, required notes, approval and time off policies — all of which the assistant enforces); choosing the inference deployment option; governing retention of the OTS database, which holds the assistant's transcripts.

13. Frequently Asked Questions

Does the AI have access to our database? No. The model receives only the user's typed sentence and calendar context. All data access is performed by Office Timesheets application code under the user's permissions.

Can the AI create, change, or delete timesheet data? No. The model's output is a structured suggestion. Only an explicit user confirmation triggers a write, which then passes the same validation as the entry forms. Time off requests additionally follow the normal approval process.

Is our data used to train AI models? No. Prompt and response data is never logged or trained on, and Ollama's hosting partners are required to operate under no-logging, no-training, zero-retention policies.

What personal data reaches the model? Only what the user types in a free-text sentence. Structured answers — notes, titles, hours, times, task picks — are captured by Office Timesheets directly and never sent to the model.

What happens if the model is wrong? A wrong suggestion becomes a visible proposal or a question. The user edits or cancels it; nothing wrong can be saved silently, and out-of-bounds values are discarded by validation before the user even sees them.

What happens if the model is down? The assistant silently switches to its button-driven mode. Every feature remains available; no data is lost.

Can we run without any AI? Yes. With AI parsing disabled, the assistant is a fully functional guided interface with zero inference.

Where does inference run? By default, on Ollama's US-based cloud using Google's Gemma 4 – 31B model, on modern NVIDIA hardware — or on your own Ollama harness.