ChatGPT

Privacy for Persistent Agents: Model, Playbook, Quests, Metrics, Redaction

Executive Summary: A local-first AI wellness agent must enforce *privacy by design*: minimize data, segregate memory, and default to "safe mode" (no file I/O or external calls) [1] [2]. Data is classified into **Secrets** (passwords, keys), **Sensitive Personal** (health, financial PII), **Operational Telemetry** (logs/metrics), and **Benign** (non-sensitive state). Each class has strict handling rules (see Table). Retention is purpose-limited: e.g. session chat logs retained ~30d, telemetry ~30–90d, audit logs ~1y [3] [4]. Data not needed must be summarized, hashed, or deleted. Users must control deletion/export of their data [1]. Eighteen **Privacy Quests** (daily/weekly/monthly/on-event) reinforce hygiene; each has a *Safe-Mode* variant (reflection-only, no file I/O) and an *Authorized* variant (with explicit user permission). Proof artifacts report only aggregate or hashed results (e.g. counts of redacted items, not raw PII). We define 10 leading indicators (e.g. sensitive-memory fraction, redaction events) and 5 lagging indicators (e.g. confirmed leaks, audit failures) to monitor compliance. Finally, we give redaction patterns (regex, field-wiping, whitelisting) for logs and memory, drawn from recent best practices [5] [6]. All recommendations align with privacy principles (GDPR's minimization/storage limitation [7]) and current AI security guidance [8] [9].

1. Data Classes & Handling Rules

Define four data classes with handling rules (local-first context):

- **Secrets**: Credentials (passwords, tokens, API keys), cryptographic keys, private keys. *Rules: Never* persist raw; only handle in-memory during a session and immediately erase. If stored briefly (e.g. for tool auth), encrypt and delete once used. Do not log or display secrets [10] [11].
- **Sensitive Personal Data**: Any PII/PHI (names, addresses, SSNs, health info, financial data). *Rules:* Collect only if essential (data minimization) [2] [7]. Prefer storing as pseudonyms/hashes. Limit retention to the minimum needed (e.g. chat transcripts 30d). Secure at rest (encrypted) and in transit. Logs and telemetry must scrub or aggregate this data before export [12] [11]. Enforce purpose limitation: do not reuse data beyond its original intent without consent [13] [10].
- **Operational Telemetry**: Agent logs, performance metrics, usage events. *Rules:* Collect minimal telemetry. Strip or mask any embedded PII or secrets [9] [11]. Retain only for troubleshooting and compliance: e.g. error logs 30–90 days, access/audit logs ~1 year [3]. Export only aggregate metrics (counts, latencies) and sanitized summaries, never raw logs with sensitive values.
- **Benign Data**: Non-sensitive context (e.g. generic preferences, static knowledge, public facts). *Rules:* May be cached for functionality, but still apply retention limits and access controls. Even innocuous data should not leak external identifiers.

Handling Rules Table:

Data Class	Examples	Handling
Secrets	Passwords, API keys, private tokens [10]	Never log; use only in-memory. Encrypt if stored momentarily, then delete. No export or chat with agent.
Sensitive Personal	Name, DOB, SSN, medical, financial info	Collect only if required. Store hashed/pseudonymized. Purge promptly (e.g. 30d) [2] [7]. Mask or drop in logs. Aggregate telemetry. Consent/user control required [1].
Operational Telemetry	Debug logs, API calls, metrics	Log only metadata (timestamps, status codes). Redact any PII/secrets [5] [11]. Short retention (typically 30–90d) with longer for security logs [3]. Export only sanitized aggregates.
Benign Data	Public knowledge, non-sensitive context	May persist for functionality. Still apply encryption/ACLs and scheduled cleanup. Regularly review for inadvertently sensitive content.

These classes align with privacy-by-design principles: **data minimization**, **purpose limitation**, and **storage limitation** [7] [2]. For example, GDPR requires data be "adequate, relevant and limited to what is necessary" and "kept no longer than necessary" [7], a principle we enforce by strict retention scheduling. Memory governance frameworks echo this: "bounded continuity" (close tasks and delete excess), "selective retention" (keep only important summaries) and "policy-bound memory" (metadata with retention rules) [14] [15]. The agent should log all read/write/delete operations for audit, and tie each memory item to a retention policy [15] [16]. Critically, users retain full control (opt-in consent, view/delete rights) [1] [12].

2. Retention & Deletion Playbook

Default Retention Windows: Following industry norms, set conservative retention by data type. For a local agent:

- **Ephemeral Memory (Working Context):** Cache of immediate conversation context (chat buffer) — delete at session end (e.g. after 1 hr or when task complete).
- **Session Transcripts:** Raw conversation logs — retain ~30 days (adjustable by user). After expiration, delete or replace with a sanitized summary.
- **Vector/Embedding Store:** Long-term memory embeddings — prune entries older than 90 days by default, or convert them to abstracted summaries. Store only nondisclosive embeddings (no raw text) [8].
- **Structured Notes/Plans:** User-created notes or plans — retain ~180 days or per user setting; older items are either archived (locally) or summarized.
- **Security/Audit Logs:** Access events (authentication, permission changes) — keep at least 1 year for audit trails (as per SOC 2/ISO 27001) [3].
- **Telemetry (Metrics):** Performance/error stats — short-term (30–90 days) then roll up. Compliance-related aggregates (e.g. "redactions performed this month") retain longer (audit cycle +1y) [3].

These windows respect both operational need and privacy laws (e.g. GDPR "storage limitation" [7]). A *timeline diagram* (below) illustrates typical retention periods:

```
gantt
    dateFormat  YYYY-MM-DD
    title Retention Schedule (Example)
    section Session Data
    Chat Transcripts :active, ct, 2026-02-01, 30d
    Working Memory    :active, wm, 2026-02-01, 1d
    section Long-Term Memory
    Embeddings        :ltm, after ct, 90d
    Summaries         :sums, after ltm, 180d
    section Logs & Telemetry
    Debug Logs        :logs, 2026-02-01, 90d
    Audit Records     :audit, after logs, 365d
    Metrics Rollups   :met, after audit, 365d
```

Storage vs. Hash: Whenever possible, replace raw data with hashes or summaries. For example, store hashed identifiers (e.g. user IDs) instead of names in logs, or only keep vector similarities rather than full transcripts [8]. Use salted hashes for any identifiers and drop the salt after use. For textual memory (e.g. notes), consider storing only concise abstracts or key facts (with contextual links) and discarding verbatim text [8].

Deletion Workflows: The agent should implement automated and manual deletion:

- *Automated:* At defined intervals, purge expired items per policy. Securely delete (scrub file, remove from vector DB). On memory refresh (new version), archive old content or delete it promptly.
- *Manual:* Provide user-facing controls to *review or delete memory*. E.g. “Clear Last Session”, “Forget All Health Data”, etc. These actions should remove data entirely (or replace with anonymized digest) [1].

User Export: Under “data portability” ethos, the agent can export user data (for backup or GDPR requests). Exports should be limited to user’s personal contributions, and given only in sanitized form (e.g. export of conversation transcripts with all PII removed or replaced by hashes).

Data Flow Diagram: A high-level flow of data handling:

```
flowchart LR
    A[User] -- input --> B(Agent)
    B --> C[Working Memory (volatile)]
    B --> D[Vector DB (persistent)]
    C -- write/read --> D
    B --> E[Tools/API]
    subgraph Logs
        F[Telemetry (raw)]
    end
    B --> F
    style C fill:#ffa
    style D fill:#faa
    style F fill:#fdd
```

```
    %% Sanitization points
    C -.-> G[Sanitization]
    F -.-> G
    G -.-> C
    G -.-> F
```

Legend: Orange = session state, red = persistent store. All data passing between components is filtered by sanitization (e.g. regex redaction).

Cited Guidelines: These controls reflect GDPR/NIST principles (collect minimum info [2] [7]) and observed industry practice (e.g. OpenAI default 30-day logs [4], opt-in zero-retention modes [4] [17]). The OneUptime guide recommends tagging telemetry with categories ("security", "operational", "personal") and applying tailored retention [3], which we adapt here.

3. Quest Pack: Privacy Hygiene Tasks

We propose a *Quest Pack* of 18 routine tasks (quests) to enforce hygiene. Each quest has:
- **Name & Cadence** (daily/weekly/monthly/on_event),
- **Safe-Mode Steps** (agent prompts/questions only, no file access),
- **Authorized-Mode Steps** (requires explicit user permission for scanning or modification),
- **Proof Artifact Schema** (what the agent reports, containing no raw secrets/PII).

Example quests are summarized below:

Name	Cadence	Safe-Mode (Reflective)	Authorized-Mode (Inspected)	Proof Artifact (Redacted)
Secret Self-Check	Daily	Ask user: "Did you input any password or secret today?" (reminder)	Scan recent user input for patterns (e.g. regex for keys/ credentials).	`{\"flags\":0}` or `{\"flags\":1, \"type\":\"[SECRET]\"}`
Memory Summary	Daily	Agent lists count of topics remembered (no details). E.g. "I recall 3 items."	Generate summary of yesterday's memory entries, replacing names with `[NAME_REDACTED]`, etc.	`{\"topics\": 3,\"sensitive_items\":0}`
Telemetry Review	Daily	Report aggregate stats (e.g. "0 errors, 5 queries today").	Show histogram of error counts; ensure no raw logs are displayed.	`{\"errors\":0,\"queries\":5}`

Name	Cadence	Safe-Mode (Reflective)	Authorized-Mode (Inspected)	Proof Artifact (Redacted)
Privacy Tip	Daily	Remind user of best practice (e.g. “Never share passwords”).	N/A (informational only)	`{}` (no artifact)
Consent Check	Weekly	Ask user to review agent permissions (e.g. safe vs. authorized).	If changed, log timestamp; update status.	`{\"consent_updated\":true}`
Memory Cleanup	Weekly	Instruct user to confirm old memories to forget.	Delete memory older than 30d (with user approval).	`{\"deleted_entries\":X}`
PII Scan	Weekly	Ask user: “Did I store any personal info you’d like to remove?”	Run PII-detection on memory DB; mask or flag findings.	`{\"pii_found\":0}` or `{\"pii_found\":2,\"types\":[\"SSN\",\"EMAIL\"]}`
Tool Audit	Weekly	List enabled tools/ libraries in conversation (no file check).	Verify tool API keys are not stored; expire unused tokens.	`{\"tools_reviewed\":true}`
Update Policy	Weekly	Read aloud any privacy-policy changes (if connected).	Compare new vs old policy; flag differences needing acceptance.	`{\"policy_changes\":0}`
Data Export Prep	Monthly	Remind user they can export data.	Bundle user data (sans PII) into an export file (hashed values).	`{\"export_ready\":true}`
Full Data Audit	Monthly	Ask user to confirm “no sensitive data leaked”.	Summarize all memory content types and ages (with [REDACTED]).	`{\"PII_records\":0,\"other_records\":10}`
Sanitization Test	Monthly	Pose a dummy question to ensure no secrets returned.	Insert a test token into memory and verify redaction occurs.	`{\"redaction_ok\":true}`

Name	Cadence	Safe-Mode (Reflective)	Authorized-Mode (Inspected)	Proof Artifact (Redacted)
Backup Keys	Monthly	Remind user to rotate their keys regularly.	Generate new encryption keys; securely delete old ones.	`{\"key_rotated\":true}`
Review Logs	Monthly	Summarize log volumes in conversation (no content).	Browse log files for anomalies (without showing them).	`{\"anomalies\":0}`
Health Disclaimer	On_event	After any health-related query, remind user: "I'm not a doctor."	N/A (compliance reminder).	`{}`
Tool Gating	On_event	Before any file/ tool use, remind "I need your permission."	Enforce permission step before file reads/writes.	`{\"permission_granted\":true}`
Incident Drill	On_event	Simulate a data leak scenario and walk through user's response steps.	Check that "delete" and "report" commands work on dummy data.	`{\"drill_passed\":true}`
Transparency Check	On_event	Ask user: "Do you know what data I've saved about you?" and explain.	Show list of data categories stored (counts only).	`{\"transparency_reported\":true}`

Notes: In **Safe-Mode**, the agent uses conversation only (asks questions, confirms policy) and never reads or writes files or databases. In **Authorized-Mode**, actions like scanning logs or deleting memory are gated by an explicit user confirmation step. The **Proof Artifact** should be a simple record (e.g. JSON) containing only sanitized, aggregate information. For instance, if a quest scans for secrets, it might output `{"flags":0}` or `{"flags":2,"patterns":["[EMAIL]","[PASSWORD]"]}` without showing any actual email or password. This ensures auditability without exposing raw PII.

4. Privacy Metrics (Indicators)

Leading Indicators (predictive, proactive metrics):

- **Sensitive-Data Rate**: Fraction of memory entries flagged as PII/sensitive (lower is better).
- **Quest Compliance**: % of scheduled privacy quests completed on time.
- **Safe-Mode Usage**: Ratio of sessions in safe-mode vs. authorized-mode (higher safe-mode is safer).
- **Redaction Events**: Count of redactions performed per day (used to track trends).

- **Data Exposure Attempts**: Number of times user pastes potential secrets (monitored by regex).
- **Memory Age**: Average age of items in long-term memory (tracked against TTL).
- **Consent Status**: Time since last user consent review (should be refreshed regularly).
- **Sanitization Coverage**: % of logs/outputs passed through sanitizers (aim for 100%).
- **Audit Log Completeness**: % of memory operations that have audit records.
- **Configuration Drift**: Changes to privacy settings (should be minimal and logged).

Lagging Indicators (reflect past incidents):

- **Privacy Incidents**: Count of confirmed data leaks or policy violations.
- **Sensitive Data Stored**: Volume of sensitive data remaining past retention (target=0).
- **Audit Failures**: Number of audit queries that found noncompliance.
- **User Complaints**: Reports from user about unexpected data usage.
- **Regulatory Actions**: Any legal notices or fines received (should be zero).

These indicators help track the health of the agent's privacy posture. For example, an uptick in *Redaction Events* or in *Sensitive-Data Rate* would *lead* to investigating a privacy risk. Meanwhile, incidents and audit failures are *lagging* outcomes to minimize through all above controls.

5. Redaction & Sanitization Patterns

To ensure no secrets or PII leak from logs, transcripts, or memory, implement robust text-sanitization recipes. Recommended patterns and tools include:

- **Regex Replacement (unstructured text):** Use regular expressions to mask common PII. Examples (based on recent guidance [5] [6]):
- **Social Security Numbers:** `\b\d{3}-\d{2}-\d{4}\b` → replace with `[SSN_REDACTED]` [5].
- **Emails:** `\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\b` → `[EMAIL_REDACTED]`.
- **Credit Cards:** `\b4[0-9]{12}(?:[0-9]{3})?\b` → `[CARD_REDACTED]` (example for Visa cards) [6].
- **IP Addresses:** `\b(?:\d{1,3}\.){3}\d{1,3}\b` → `[IP_REDACTED]` (optional) [18].
- **Secrets/Tokens:** Patterns like `Bearer\s+[A-Za-z0-9\.\-_]+\b` or known key prefixes (e.g. `AKIA[0-9A-Z]{16}` for AWS keys) → `[TOKEN_REDACTED]` [19].

- **Field Stripping (structured data):** For JSON logs or key-value attributes, remove entire fields that typically contain sensitive data [11]. For example:

```
delete_key(attributes, "user.password")
delete_key(attributes, "http.request.header.authorization")
delete_key(attributes, "http.cookie")
```

This drops any value that shouldn't appear in logs.

- **Whitelist Filtering:** Instead of blacklisting values, use a whitelist of allowed fields and drop all others [6]. For example, only permit `service.name`, `severity`, `http.status_code` in

telemetry exports; omit everything else. This approach prevents unknown keys (like user data) from slipping through.

- **Redaction Processor:** Use log processing tools (e.g. OpenTelemetry's Redaction processor) that can automatically mask values matching certain regex and report summary counts [6] [20]. Configuring a summary level (e.g. `summary: debug`) gives you the number of redacted values for auditing.

- **Structured JSON Sanitization:** If logs contain nested JSON, target specific fields. For example:

```
# Pseudocode for structured JSON redaction
replace_pattern(body["user"]["email"], ".*", "[EMAIL_REDACTED]") if
body.user.email exists
replace_pattern(body["payment"]["card_number"], ".*", "[REDACTED]") if
body.payment.card_number exists
```

  This ensures only the field values are sanitized, not the whole log.

- **Testing & Validation:** Always test your redaction rules with sample data [21]. E.g. feed a log containing "User paid with card 4111111111111111" into the pipeline and verify the output reads "User [EMAIL_REDACTED] paid with card [CARD_REDACTED]" [21]. Confirm performance impact is acceptable (regex on large logs adds ~5–10% CPU) and adjust complexity as needed.

- **On-the-Fly Sanitization:** In interactive chat, the agent should apply redaction in real time. For instance, if a user accidentally mentions a phone number, the agent's transcript processing should replace the number with `[REDACTED]` before storing. Alert the user (e.g. "I removed what looked like a phone number for your privacy").

Citations: The above patterns and strategies follow industry practice [5] [6]. For example, AWS CloudWatch's data protection policies automatically mask credit card and PII fields in logs, demonstrating that native masking is feasible [22]. Likewise, privacy regulations forbid storing raw identifiers; robust redaction (as shown) is essential to compliance.

Sources: Authoritative guidelines and recent industry sources were used. Key references include NIST privacy guidelines [2], GDPR principles [7], an AI memory governance analysis [14] [15], and technical best-practices from cloud/AI vendors and privacy experts [9] [4] [3] [5] [10]. These inform the data classes, retention policies, quests, metrics, and redaction recipes above.

---

[1] [9] How to prevent privacy leakage risks in AI Agent? - Tencent Cloud
https://www.tencentcloud.com/techpedia/126522

[2] [12] Privacy
https://pages.nist.gov/800-63-4/sp800-63a/privacy/

3 How to Implement Telemetry Data Retention Policies That Satisfy SOC 2 and ISO 27001 Requirements
https://oneuptime.com/blog/post/2026-02-06-telemetry-retention-soc2-iso27001/view

4 17 Data controls in the OpenAI platform
https://developers.openai.com/api/docs/guides/your-data/

5 6 11 18 19 20 21 How to Redact Sensitive Data (PII, Tokens, Credentials) from Logs in the OpenTelemetry Pipeline
https://oneuptime.com/blog/post/2026-02-06-redact-sensitive-data-pii-opentelemetry-pipeline/view

7 13 Art. 5 GDPR – Principles relating to processing of personal data - General Data Protection Regulation (GDPR)
https://gdpr-info.eu/art-5-gdpr/

8 14 15 16 What Is Memory Governance (and Why Is It Important for AI Security)? - Acuvity
https://acuvity.ai/what-is-memory-governance-why-important-for-ai-security/

10 Ranked: 8 Things You Should Never Paste Into an AI Chatbot
https://www.eweek.com/news/8-things-never-paste-into-ai-chatbot/

22 Handling sensitive log data using Amazon CloudWatch | AWS Cloud Operations Blog
https://aws.amazon.com/blogs/mt/handling-sensitive-log-data-using-amazon-cloudwatch/