Outcome Owl

See Clearly. Act Decisively.

Customer Privacy Guidance

Hygiene patterns to keep regulated content out of Outcome Owl.

Introduction

Outcome Owl provides operational visibility into your business processes. Most of that visibility comes from a small set of attributes — identifiers, statuses, timestamps, business categories — none of which need to contain personal, regulated, or otherwise sensitive content to be useful.

This document is for the privacy officer, compliance officer, security architect, or operations leader who is deciding what data should flow into Outcome Owl. The recommendations here are not requirements imposed by the platform — they are hygiene patterns that let you keep regulated content out of an external observability tool while still getting full value from the analytics, deviation detection, and process visibility the platform delivers.

If you are evaluating Outcome Owl on behalf of a regulated organization (healthcare payer, financial services, insurance, government, EU-resident-data handler), start with *Regulated-Vertical Considerations* and *Worked Examples*.

1. Architectural Privacy Posture

These are the privacy-relevant facts about how Outcome Owl is deployed. They are deployment defaults, not negotiated per-customer terms.

Property	What this means
Single-tenant deployment	Each customer runs on dedicated AWS infrastructure such as a dedicated database, application servers, S3 buckets, and VPC. No customer's data is co-mingled with any other customer's data at any layer.
Customer-controlled data flow	Outcome Owl receives data your systems push to it. The platform does not pull data from your environment. You decide what attributes are populated and what values they contain.
Optional IP allowlisting	The UI and API entry points can be restricted to specific IP ranges (e.g. corporate offices, VPN, or defined network ranges) so the deployment is reachable only from your approved networks.
Standard PostgreSQL storage	Data is stored in standard relational tables, not a proprietary format. Export at any time, in any volume, by standard SQL or CSV. There is no vendor format converting your data into something only Outcome Owl can read.

Property	What this means
Encryption	TLS in transit. Encryption at rest on both RDS and S3 per AWS security posture.
Configurable retention	Restatement history and intake staging records each have independent retention policies (1 – 3,650 days). Data is automatically purged on the schedule you configure. Audit logs and session records are manually purged to ensure SOC2 compliance.
AWS Secrets Manager	Database credentials, JWT signing keys, SSO secrets, and API keys are stored in AWS Secrets Manager and retrieved at runtime. Not in code, not in environment files.
Audit trail	Every authentication event, configuration change, user account modification, and administrative action is recorded with actor identity, timestamp, IP, and what changed.
No proprietary lock-in	If you leave the platform, you take your data. There is no proprietary export format. PostgreSQL is open and standard.

2. The Customer / Platform privacy split

A useful frame for the privacy conversation:

Concern	Who owns it
What the platform does with whatever it receives (encryption, access control, audit trail, retention, isolation).	**Platform.** Covered in 1.
What you choose to send into the platform in the first place.	**Customer.** Covered in 3, 4, and 5.

The architectural side (platform) is fixed by deployment. The data side (customer) is fully under your control. Most privacy and regulatory questions reduce to a single design choice: *what attributes should we populate, and with what values?*

The default answer for regulated organizations is **surrogate identifiers and process metadata only**. The rest of this document explains how to apply that default.

3. Attribute Categories and Hygiene

Every Observation has system-defined fields (UUIDs, timestamps, statuses) and customer-defined Attributes (name/value pairs you populate). This table classifies the most common attribute categories and recommends a hygiene pattern for each.

Category	Examples	Recommended Hygiene
Process identifiers	ObservationID, ParentObservationID, OriginatingObservationID.	Use **surrogate IDs** — hash or tokenize the natural business key (claim number, loan number, member ID). The tree-assembly logic depends on the IDs being stable and unique within your deployment, not on them being human-readable.
Process statuses	"Pending Adjudication," "Awaiting Documentation," "Paid," "Denied"	Send as-is. Status values are operational, not regulated. They are the primary grouping dimension for analytics.
Timestamps	ProcessStepStartUTCTimestamp, ProcessStepEndUTCTimestamp	Send as-is. Timestamps are not regulated content.
Business categories	Claim type, region, product line, network tier, channel, geography	Send as-is. These are the primary attributes for Attribute Insights, Attribute Drift, and lift analysis. They are not personal data.
Quantitative values	Allowed amount, paid amount, deductible, copay, cycle time	Send as-is for amounts and durations; **tokenize** if the value is itself an identifier (e.g., account number masquerading as a number).

Category	Examples	Recommended Hygiene
Personal identifiers	Member name, claimant name, borrower name, email, phone, SSN, DOB, address	**Do not send.** Use a tokenized member/claimant key instead. The platform never needs the cleartext name to assemble the process tree or compute analytics.
Clinical or financial detail	Diagnosis codes, procedure codes, prescription content, account balance, card data	**Do not send.** These are not needed for process visibility. If you need to slice analytics by, say, "high-cost claim category," create a derived non-regulated category attribute (e.g., `cost_tier = "high" | "medium" | "low"`) and send the category, not the underlying value.
Free text	ObservationComments, custom notes, narrative descriptions	**Be explicit about what is permitted.** Free-text fields can hold any content; they are the most common path for regulated content to enter the platform unintentionally. Either (a) restrict free-text fields to operational metadata (e.g., "Approved by SR adjuster after second review"), or (b) disable free-text attributes entirely for sensitive process types.
System of origin tags	"claims_platform_v3", "eligibility_legacy", "manual_intake"	Send as-is. Useful for diagnosing data quality issues by source system.
External case references	A pointer back to the source system's record	Use a **surrogate** that maps to the source record only inside your environment. The platform does not need a lookup-able external ID; it needs a stable identifier.

Hygiene by design, not by review.

The most reliable privacy posture is "the platform never receives the regulated value in the first place." Tokenization, hashing, and category derivation should happen in your ingestion pipeline (the layer that publishes to Outcome Owl), not as a cleanup pass after data has already been sent. Review-after-ingestion is a remediation strategy; tokenization-before-ingestion is a privacy strategy.

4. Surrogate Identifier pattern

Surrogate identifiers are the central hygiene pattern for regulated-data customers. The pattern works as follows:

1. **Inside your environment**, maintain a mapping table: (natural_business_key) → (stable_surrogate_id). The mapping never leaves your environment.
2. **In the data you publish to Outcome Owl**, use only the surrogate. Every ObservationID, ParentObservationID, OriginatingObservationID, member reference, and claimant reference is the surrogate value.
3. **In the Outcome Owl UI**, your operations team sees the surrogate. When they need to look up a specific case in the source system, they cross-reference the surrogate against your internal mapping.

The key property: the surrogate is **stable** (the same business record always maps to the same surrogate) and **unique** (no two business records map to the same surrogate). Tree assembly, analytics, and deviation detection all work identically on surrogates as on natural keys.

Choosing a surrogate generation strategy:

Strategy	When to use	Notes
Hash of natural key (e.g., SHA-256)	Default for most customers.	Deterministic, repeatable, no mapping table needed if you accept that the hash is the surrogate. Caveat: if the natural key is low-entropy (e.g., 9-digit SSN), the hash is brute-forceable; salt with a per-deployment secret if this matters.

Strategy	When to use	Notes
UUID with mapping table	When you need to dissociate the surrogate from the natural key entirely.	Generate a fresh UUID per business record; store the mapping internally. Highest privacy posture. Requires the mapping infrastructure.
Format-preserving encryption (FPE)	When the surrogate must look like the natural key for legacy compatibility.	Specialized; usually overkill for observability.

For most customers, the SHA-256 hash with a per-deployment salt is the right default. The salt makes the hash useless to anyone outside your deployment.

5. Free-text Attribute discipline

Free-text fields (`ObservationComments`, custom narrative attributes) are the most common privacy hazard. They are unbounded, and operations staff naturally type whatever context they need — including PHI, PII, and account-identifying detail — without realizing it is leaving the source system.

Recommended posture:

1. **Make a deliberate decision** about whether free-text fields are populated at all for sensitive process types. The default answer for healthcare payer claims, regulated financial transactions, and processes touching identifiable individuals should be "no."
2. **If free-text is populated**, define an explicit allowed-content rule and train the team on it. Examples of allowed content: process action descriptions ("Routed to specialist for second review"), system-of-record pointers ("See case ref 4892 in claims platform"), operational tags ("Vendor SLA exception"). Examples of disallowed content: claimant names, member identifiers, diagnosis codes, account numbers, dollar amounts tied to identifiable records.
3. **Periodically sample** the free-text content for drift. Operations staff under time pressure will revert to whatever is fastest; a quarterly review catches and corrects this.

4. **For automated free-text** (system-generated comments populated from the source system), validate the upstream system does not include regulated content in the comment template. The privacy posture is only as good as the most generous source.

The platform itself does not parse, search, or analyze free-text content for analytics; free-text fields are display-only metadata. The cost of populating them with regulated content is therefore high (regulatory exposure) and the analytical benefit is low (the platform's analytics work on structured attributes, not free text). The cost-benefit strongly favors keeping free-text minimal.

6. Regulated-vertical considerations

Outcome Owl's regulatory posture is framework-by-framework and is discussed directly with each prospective customer. The following defaults apply *until that conversation is closed* — they describe what to send if you are operating without a framework-specific commitment from Outcome Owl.

HIPAA (Healthcare Payers, Providers, Business Associates)

- **Default posture: no PHI in any attribute.** Use surrogate member identifiers, surrogate claim identifiers, surrogate provider identifiers. Send the process metadata (status, claim category, processing queue, region, network tier) — not the underlying clinical or member-identifying values.
- **BAA status**: a Business Associate Agreement between your organization and Outcome Owl is a roadmap conversation, not a current default. Until a BAA is in place, treat Outcome Owl as a non-BA recipient and design your ingestion accordingly (no PHI).
- **Diagnosis / procedure codes**: derive non-PHI categories upstream (e.g., `claim_type = "Inpatient" \| "Outpatient" \| "Pharmacy" \| "DME"`) and send the category. Never send raw ICD-10, CPT, HCPCS, or NDC codes attached to identifiable individuals.
- **Free-text fields**: disable for healthcare payer claims processes. The risk-to-benefit ratio is unfavorable.

GLBA (Financial Services — Banks, Credit Unions, Lenders, Insurers)

- **Default posture: no NPI (nonpublic personal information) in any attribute.** Surrogate borrower / customer identifiers. Surrogate account numbers (or send only a derived category — "personal_loan," "auto_loan," "mortgage" — instead).

- **Account balances, transaction amounts, credit scores**: derive non-NPI categories upstream when possible (e.g., `balance_tier`, `score_band`). If raw amounts are truly necessary for threshold rules, ensure the account identifier on the same record is a surrogate so the amount is not linkable to an identifiable customer.
- **Free-text fields**: same posture as HIPAA — disable for processes touching identifiable customer accounts unless content discipline is enforced and audited.

PCI DSS (Payment Card Data)

- **Default posture: zero card data in the platform.** No PAN, no CVV, no expiration date, no cardholder name, even tokenized. The cleanest posture is to never include payment card data in any attribute, free-text or structured.
- **If you need payment-process visibility**: send the process state, the merchant category, the channel, the outcome — not the card. The platform's purpose is process observability, not transaction processing.
- **PCI scope**: keeping card data out of the platform keeps Outcome Owl out of your PCI scope. Do not introduce PCI scope by sending card data when surrogate or category-level data would suffice.

GDPR (EU Personal Data)

- **Default posture: pseudonymization at minimum.** EU personal data sent to the platform should be pseudonymized (surrogate identifier substitution) before it leaves your environment. The mapping back to the natural identity stays inside your data controller environment.
- **Data Processing Agreement (DPA)**: a DPA covering Outcome Owl as a data processor is a roadmap conversation, not a current default. Until a DPA is in place, treat the platform as a non-DPA recipient.
- **Data residency**: Outcome Owl is currently AWS-only and the deployment region is configurable. If your DPO requires EU-resident data, confirm the region of the deployment before any data is sent.
- **Right to erasure**: pseudonymized data is still subject to GDPR if linkable. Plan for erasure as: (a) you delete the natural-to-surrogate mapping in your environment (removing the link), and (b) the surrogate is purged from Outcome Owl per your retention policy. The combination delivers effective erasure for most use cases.

Other Frameworks

For SOC 2 (the *customer's* SOC 2, not Outcome Owl's), CCPA, NYDFS Cybersecurity Regulation, FFIEC, and other regulatory frameworks: the same first principle applies.

Send process metadata, not regulated content. Where the framework requires a specific contractual posture (BAA, DPA, attestation, audit cooperation), discuss directly with Outcome Owl before sending data subject to that framework.

7. Worked examples

Worked Example A — Healthcare Payer / Medical Claim Adjudication

A regional health plan wants observability into its claim adjudication workflow. The natural data model has member names, claim numbers, diagnosis codes, dollar amounts, and free-text adjuster notes. None of those need to enter Outcome Owl.

What to send:

Attribute	Value pattern
ObservationID	`SHA256(salt + claim_number)` — surrogate, stable, unique
ParentObservationID	Surrogate of the parent process step (intake → eligibility, eligibility → adjudication, etc.)
OriginatingObservationID	Surrogate of the claim intake event
claim_type	`"Inpatient"`, `"Outpatient"`, `"Pharmacy"`, `"DME"`, `"Behavioral Health"` — categorized upstream
network_tier	"In-Network", "Out-of-Network", "Tier 2"
state	Member's resident state (no street address)
processing_queue	"Auto-Adjudicated", "Pended-Documentation", "Pended-Authorization", "Manual Review", "FWA Review"
prompt_pay_clock_start	Timestamp
current_status	Status from the lifecycle
cost_tier	`"<$500"`, `"$500–5K"`, `"$5K–50K"`, `"$50K+"` — categorized upstream from the actual dollar amount
assigned_team	"Adjudication Team A", "Specialty Pricing", "Provider Relations"

What NOT to send:

- Member name, member ID, DOB, SSN, address, phone, email
- Claim number (use surrogate)
- Diagnosis codes (use derived `claim_type`)
- Procedure codes (use derived `procedure_category` if needed)
- Raw dollar amounts on identifiable records (use `cost_tier`)
- Adjuster free-text notes — disabled for this process type
- Provider name, NPI (use surrogate provider ID)

Result: the operations team sees adjudication SLA performance, pended-claim aging, network-tier patterns, FWA review throughput, prompt-pay risk by state — all the analytics they need — without any PHI ever leaving the claims platform.

Worked Example B — Loan Origination (GLBA)

A regional lender wants observability into the loan origination pipeline. The natural data model has borrower SSNs, FICO scores, account numbers, and dollar amounts.

What to send:

Attribute	Value pattern
ObservationID	SHA256(salt + application_id)
loan_type	"Personal", "Auto", "Mortgage", "HELOC", "Small Business"
channel	"Branch", "Online", "Broker", "Direct Mail"
State	`"Inpatient"`, `"Outpatient"`, `"Pharmacy"`, `"DME"`, `"Behavioral Health"` — categorized upstream
processing_step	"Application Intake", "Document Verification", "Credit Pull", "Underwriting", "Appraisal Order", "Approval", "Funding"
score_band	`"720+"`, `"680–719"`, `"640–679"`, `"<640"` — derived from FICO
loan_amount_band	"<$25K", "$25K-100K", "$100K-500K", "$500K+"
submission_day	Day-of-week (useful for staffing analytics)

Attribute	Value pattern
assigned_underwriter	Surrogate ID, not name

What NOT to send:

- Applicant name, SSN, DOB, address
- Account numbers (loan, deposit, anything)
- Raw FICO score
- Raw loan amount
- Underwriter name (surrogate only)

Worked Example C — Order Fulfillment (Lower Sensitivity)

For a non-regulated process like B2B order fulfillment, the hygiene posture is lighter. Customer names may be acceptable as-is (B2B customer names are typically not regulated personal data in most jurisdictions). The defaults still apply: send process metadata, prefer categories over raw values, and be deliberate about free-text. But the regulated-vertical concerns above are not load-bearing for non-regulated B2B processes.

8. Implementation Checklist

Before sending the first observation to Outcome Owl from a regulated-data process:

- **Identify all regulatory frameworks** that apply to the data (HIPAA, GLBA, PCI, GDPR, state-level frameworks, contractual frameworks).
- **Confirm Outcome Owl's posture** for each applicable framework (BAA, DPA, attestation status). Discuss directly if the framework requires a contractual posture and one is not yet in place.
- **Inventory the proposed attributes**. List every attribute name and example value.
- **Classify each attribute** against Attribute Categories and Hygiene. Identifier, status, timestamp, category, quantitative, personal identifier, clinical/financial detail, or free text.
- **For every personal identifier or clinical/financial detail attribute**: design a surrogate or category-derivation path upstream of Outcome Owl.

- **For every free-text attribute**: decide *enabled* or *disabled.* If enabled, document the allowed-content rule and the periodic sampling cadence.
- **Implement surrogate generation** in the ingestion pipeline (hash, UUID + mapping, FPE — see *Surrogate Identifier pattern*).
- **Sample the first batch** of data being prepared for Outcome Owl and review attribute-by-attribute against the hygiene plan.
- **Run a privacy review** with the privacy/compliance officer using this document as the working agreement.
- **Document the hygiene plan** as part of the deployment configuration. Keep it current as new processes are onboarded.
- **Schedule a privacy-posture refresh** annually (or whenever a new process is added) to confirm the hygiene plan still matches reality.

Outcome Owl is a product of Seattle Software Works, Inc.