Statistical Companion Document

Formal Statistical Foundations for the mavai Methodology

Version 1.3.5 · last updated 2026-05-27 · Copyright © 2026, Michael Franz Mannion BS

Contents

Document History . . . 3
Forward Scope: Revisions Targeted for 1.4 . . . 6
Introduction: The Assumption of Certainty in Software Testing . . . 6
- Two Dimensions of Stochasticity . . . 7
- What This Document Contains . . . 8
- The mavai Project Family . . . 8

Document Purpose and Audience . . . 9
The Model in Brief: Contracts, Tests, Experiments . . . 9
Clause Forms: Rate-Bounded and Categorical . . . 10
Two Testing Paradigms: Compliance and Regression . . . 11
1. Statistical Model . . . 13
- 1.1 Bernoulli Trial Framework . . . 13
- 1.2 Binomial Aggregation . . . 13
- 1.3 Assumptions and Limitations . . . 13
- 1.4 Criterion Decomposition of the Functional Dimension . . . 17
- 1.5 Baselines . . . 37

2. Baseline Estimation (Experiment Phase) . . . 42
- 2.1 Point Estimation . . . 42
- 2.2 Standard Error . . . 43
- 2.3 Confidence Intervals . . . 43
- 2.4 Sample Size Determination . . . 45

3. Threshold Derivation for Regression Testing . . . 46
- 3.1 The Problem . . . 46
- 3.2 One-Sided Hypothesis Testing Framework . . . 46
- 3.3 Normal-Approximation Intuition for One-Sided Bounds . . . 48
- 3.4 Threshold Calculation (Wilson Score Lower Bound) . . . 48
- 3.5 Reference Table: Wilson Score Lower Bounds . . . 50
- 3.6 Testing Against a Given Threshold (Compliance) . . . 50

4. The Perfect Baseline Problem ($\hat{p} = 1$) . . . 51
- 4.1 Problem Statement . . . 52
- 4.2 Interpretation of 100% Observed Success . . . 52
- 4.3 The Wilson Lower Bound Solution . . . 53
- 4.4 Extended Example: Highly Reliable API . . . 55
- 4.5 Theoretical Note: Beta-Binomial Alternative . . . 55

5. Test Execution and Interpretation . . . 56
5.1 Decision Rule . . . 57
5.2 Type I and Type II Errors . . . 57
5.3 Statistical Power . . . 57
5.4 Sample Size for Desired Power . . . 58
5.5 Sample Size for SLA Verification . . . 59
5.6 The Role of Minimum Detectable Effect . . . 60
5.7 Test Intent: VERIFICATION vs SMOKE . . . 60
6. The Three Operational Approaches: Mathematical Formulation . . . 62
6.1 Approach 1: Sample-Size-First (Cost-Driven) . . . 63
6.2 Approach 2: Confidence-First (Risk-Driven) . . . 63
6.3 Approach 3: Direct Threshold (Threshold-First) . . . 64
7. Reporting and Interpretation . . . 65
7.1 Transparent Statistics Output . . . 65
7.2 Confidence Statement . . . 68
7.3 Multiple Testing Considerations . . . 68
7.4 Threshold Provenance . . . 70
8. Assumptions and Validity Conditions . . . 71
8.1 When Normal Approximation is Valid . . . 71
8.2 Independence Violations . . . 72
8.3 Non-Stationarity . . . 75
8.4 Guardrails for Assumption Validity . . . 76
8.5 Repeated Use and Sequential Monitoring . . . 82
9. Summary of Key Formulas . . . 84
Estimation (Exact — MLE) . . . 84
Wald Confidence Interval (two-sided) (Asymptotic — pedagogical, not used by the methodology) . . . 84
Wilson Score Interval (Wilson score construction — default interval method) . . . 84
One-Sided Lower Bound, for threshold derivation (Operational approximation — Wald form, see §3.3; the methodology uses the Wilson score-test inversion in §3.4) . . . 84
Wilson Lower Bound, for $\hat{p} = 1$ (Wilson score construction — boundary case) . . . 84
Rule of Three, for zero failures (Heuristic — quick approximation at 95% confidence) . . . 85
Sample Size for Precision (Asymptotic — planning approximation based on normal asymptotics) . . . 85
Sample Size for Power (Asymptotic — planning approximation based on normal asymptotics) . . . 85
Empirical Percentile (nearest-rank) (Exact — definition) . . . 85
Latency Threshold Derivation (binomial order-statistic upper bound) (Non-parametric / distribution-free when the required rank exists within the sample (§12.5.2.1); saturated and advisory only otherwise. Exact for continuous F_T; conservative under ties.) . . . 85
10. Transparent Statistics Mode . . . 86
10.1 Purpose . . . 86
10.2 Output Structure . . . 86

10.3 Example output: a multi-criteria contract . . . 89
10.4 Mathematical Notation . . . 94
10.5 Validation by Statisticians . . . 94
10.6 Calibration Conformance . . . 94
10.7 Conformance Checklist . . . 96
11. Statistical Discipline Through Design . . . 97
12. Latency: Empirical Percentile Analysis . . . 98
12.1 The Statistical Challenge of Latency . . . 98
12.2 Empirical Percentile Estimation . . . 98
12.3 Latency Assertions . . . 100
12.4 Threshold Derivation from Baselines . . . 101
12.5 Sample Size Requirements for Percentile Estimation . . . 105
12.6 Enforcement Modes . . . 108
12.7 Relationship to Pass-Rate Testing . . . 108
12.8 Numerical Conventions . . . 109
Appendix A. Elements of the statistical model . . . 110
References . . . 112

Document History

#	Date	Milestone
1	**2025-12**	**First issue.** Uni-dimensional service contract covering only functional stochasticity: Bernoulli-trial model, binomial aggregation, and Wilson-score intervals as the basis for what later became the distributional contract idea formalised in `DISTRIBUTIONAL-CONTRACTS.md`.
2	**2026-02**	**Temporal dimension added.** The methodology expanded from a single service-contract dimension to two (functional and temporal). Latency was introduced as a non-parametric problem via empirical percentiles (nearest-rank), and a first-generation (naive) threshold derivation was provided using the standard error of the mean as a proxy for percentile uncertainty, $\hat{\tau}_j = Q(p_j) + z_\alpha \cdot s/\sqrt{n_s}$.
3	**2026-04**	**Stricter latency treatment.** The latency population was formally decomposed into a tripartite contract (correctness / availability / latency-given-success), with the perverse-incentive hazard of conditioning on success named explicitly. Additionally, the $s/\sqrt{n_s}$ approximation — which understated tail-percentile uncertainty for heavy-tailed distributions — was replaced by the exact binomial order-statistic upper confidence bound on the baseline quantile, restoring statistical symmetry with the Wilson-based construction on the pass-rate side.

#	Date	Milestone
4	**2026-05**	**Worked-example correction in §§4.3.2–4.4.** The 100%-baseline worked example, the §4.3.3 reference table, and the §4.4 extended example previously derived their test thresholds using a Wald approximation ($p_0 - z \cdot \mathrm{SE}$), which was inconsistent with the one-sided Wilson lower-bound construction stated as the methodology's default elsewhere in the document. All three now apply the same Wilson construction. The §4.3.2 100-sample threshold becomes ≈ 0.969 (97 / 100 successes) in place of ≈ 0.989; the §4.3.3 table values shift accordingly; and the §4.4 thresholds (baseline $n = 2000$) become ≈ 0.971 for $n_{\text{test}} = 100$ and ≈ 0.946 for $n_{\text{test}} = 50$. This is a presentation correction only; the underlying methodology is unchanged.
5	**2026-05**	**Justification of the i.i.d. working assumption.** §1.3 gains a new §1.3.1 setting out the conditions under which the Bernoulli i.i.d. premise is defensible for LLM testing, with citations to Anthropic (2026) for provider model-versioning policy and Chen, Zaharia & Zou (2023) for the empirical counterweight. Existing §1.3 material moves unchanged into §1.3.2 (formal assumptions and operational threats) and §1.3.3 (developer responsibility for trial independence — previously unnumbered). No statistical content changes.
6	**2026-05**	**Multi-criterion service contracts.** The functional dimension is partitioned per **criterion**, with each criterion running its own Bernoulli stream (§1.4). Three model primitives — postcondition, criterion, sampling — are introduced (§1.4.2), together with inferential and observational modes (§1.4.5) yielding three-valued per-criterion verdicts (PASS, FAIL, INCONCLUSIVE), the structural composite verdict (§1.4.6), and Type-I envelopes split by procedure direction. A contract's clauses are typed as **empirical** (rate-bounded, evaluated by Wilson construction and integer-cutoff machinery) or **categorical** (obligation/prohibition, discharged architecturally). Compliance and regression are distinct procedures with distinct error semantics; the integer pass cutoff c_c is the binding regression decision artefact, with Wilson identified as a score-test inversion. Population claims are codified as finite-corpus, superpopulation, or no-generalisation (§8.4.6). Latency carries a confidence-bound existence gate (§12.5). The **baseline** is an indexed family of per-criterion estimators (§1.5). The single-criterion ($m = 1$) instance recovers the methodology of milestones 1–5 unchanged.

#	Date	Milestone
7	**2026-05**	**Multi-criterion corrections (clarifications; no statistical-content change).** (i) A criterion's denominator is **all its in-scope trials** (§1.4.3, §1.4.5a); a trial is a success only on a clean PASS, and every other outcome is a FAIL carrying a diagnostic reason — condition (postcondition did not hold) or transform/no-value (no testable value could be produced). There is no per-criterion denominator policy, no `CONDITIONAL/MARGINAL` enum, and no exclusion of unproducible trials; only the applicability predicate (scope) removes a trial. (ii) **INCONCLUSIVE** is corrected to its verdict-level meaning only — a criterion whose statistical procedure cannot render PASS/FAIL (feasibility gate, $n_c = 0$, covariate misalignment). The former per-trial ”inconclusive” (a failed transform) is a FAIL with a reason; the misnomer is removed. (iii) §1.4.7 tightened to the §1.4.2 model: an experiment or test is bound to one contract and varies only its single shared sampling, which every criterion evaluates independently — no per-criterion sampling. §1.4.8 reframed accordingly. Effective denominators (n_c, K_c) and all numerical results are unchanged.
8	**2026-05**	**Criterion scope withdrawn (clarification; no statistical-content or fixture change).** The criterion scope / applicability-predicate concept — `scopePredicate`, $n_{c,\text{applicable}}$, $n_{c,\text{out-of-scope}}$, and the ”in-scope trials” framing — is removed. A sampling is N samples and every sample is seen by every criterion, so a functional criterion's denominator is always N. The sole denominator-narrowing in the methodology is latency's conditioning on success (§12). Effective denominators ($n_c = N$, K_c) and all numerical results are unchanged; no fixture or schema change.
9	**2026-05**	**Clause-form terminology refinement (no statistical-content or fixture change).** The clause form is renamed **rate-bounded** (formerly ”empirical”); the word empirical is now reserved for the threshold-**origin** axis (normative vs empirical), which the old form-name contradicted whenever a rate-bounded clause carried a normative threshold. The front-matter section ”Clause Types: Empirical and Categorical” becomes ”Clause Forms: Rate-Bounded and Categorical”. The zero-failures evidence criterion that supports an architectural commitment is named **observational**, not ”derived-empirical”. Aligns the companion with the distributional-contracts paper; no change to the statistics, the model, or the fixtures. New section 1.4.5b formalises the architectural discharge (guardrails) probabilistically and contains a worked example.

#	Date	Milestone
10	**2026-05**	**Architectural-discharge clarifications (corrections; no statistical-content or fixture change).** Refinements to the §1.4.5b guardrail model and the Clause Forms section it depends on. (i) Guardrail rate-bounded clauses no longer carry an empirical-origin threshold by assertion; consistent with milestone 9, the threshold origin is free (normative or empirical). (ii) A production estimate of p_{harm} must observe the generator's pre-guardrail candidate output; post-guardrail monitoring estimates p_{residual}, not p_{harm}. (iii) The two samplings are named apart — a red-team-input sampling (adversarial inputs, end-to-end) versus a harmful-candidate sampling (conditional sensitivity) — and the §1.4.5 pairing of observational and inferential criteria no longer presumes a shared sampling. (iv) The worked example declares its `populationClaim` (superpopulation) before applying the Wilson bound. No formula, fixture, or numerical result changes.
11	**2026-05**	**Accumulation and the baseline long-term-monitoring framing removed from §1.5 (no fixture or numerical change).** The §1.5.5 accumulation construct is withdrawn and §1.5.5 reduced to Expiration; the framing of baseline indexing as a time-indexed collection of baselines retained for long-term monitoring is removed. Baseline indexing itself is retained.

Each milestone builds on the one before; each extends the scope of what the methodology claims except where it explicitly corrects or withdraws an over-reach (milestone 11). None supersedes the Bernoulli/Wilson foundation laid in Milestone 1.

Forward Scope: Revisions Targeted for 1.4

Items identified for 1.4, deferred from 1.3 because each requires coordination with downstream frameworks and fixtures:

- **1.4-S1**: Baseline indexing carries experiment/sampling identity.
- **1.4-S2**: Threshold-notation cleanup separating the Wilson real-valued output, the integer cutoff, the displayed rate, and the compliance requirement.
- **1.4-S3**: Predictive / two-sample treatment of latency regression.

Introduction: The Assumption of Certainty in Software Testing

For decades, the dominant paradigm in software testing has rested on an implicit assumption: **systems under test behave with certainty**. Given the same input, a

correctly functioning system produces the same output. Tests produce **binary outcomes** — pass or fail — a single failure is definitive evidence of a defect, and uncertainty, when it intrudes, is treated as a ”flaky test” to be mocked away. Continuous integration, test-driven development, and quality gates all sit on that foundation.

Pockets of statistical practice have always existed alongside this mainstream — reliability engineering, performance engineering, queueing analysis, randomised testing, stochastic simulation — but they have sat outside the automated-test workflow rather than inside it. What is new, and what makes the question pressing now, is that **Large Language Models** promote uncertainty from a nuisance to an intrinsic property of the system under test. Identical inputs produce different outputs, by design; the distribution of outputs **is** the behaviour. That promotion makes statistically disciplined repeated-trial testing a first-class concern of mainstream software engineering, not the specialist subfield it used to be.

Traditional Testing	Testing Under Uncertainty
Accidental (bugs)	Intentional (sampling)
To be eliminated	To be characterised
Failure is binary	Failure is probabilistic
Single test is definitive	Single test is a sample

Two Dimensions of Stochasticity

Uncertain-system behaviour manifests along two **distinct** quality dimensions:

1. **Functional stochasticity** — whether the system produces a correct result. Given identical input, an LLM may generate valid JSON in 95 out of 100 invocations and malformed output in the remaining 5. Correctness is a random variable.
2. **Temporal stochasticity** — how long the system takes to respond, even among successful invocations. Latency is not a fixed property; it is a distribution.

The two dimensions are distinct (or orthogonal as quality concerns), not necessarily statistically independent. A fast response can be incorrect; a slow response can be correct; but correctness and latency may also covary in practice — challenging prompts may be slower and more likely to fail, overloaded infrastructure may inflate both error and tail-latency rates, and safety filters may change both refusal behaviour and latency. **The methodology does not require functional and temporal stochasticity to be statistically independent; the combined verdict is a logical conjunction rather than a probabilistic independence model.** Both dimensions require repeated observation and distributional reasoning. The mavai methodology treats them with different statistical machinery — a binomial model for functional outcomes (§§1–5), non-parametric empirical percentiles for latency (§12) — and requires both to pass for the overall test to pass.

Within the functional dimension, a contract may declare one or more **criteria** (§1.4) — each its own Bernoulli stream, with its own threshold and confidence level. Criteria within a single experiment share the experiment's sampling and so share the N samples posted to the service; a contract may run more than one experiment

when different input distributions are required. The per-criterion partition refines the evidence within the functional dimension; it does not introduce new dimensions of stochasticity, since every criterion shares the same methodological regime (Wilson on per-criterion pass-rates). A parallel diagnosis, with a different methodological response (Generalized Stochastic Dominance), appears in Garces Arias et al. (2025).

Memory consumption, token usage, and cost per call also vary and could be modelled the same way, but the methodology concentrates on correctness and latency because these two dimensions have the most direct impact on end users; resource consumption is usually managed through infrastructure tooling rather than test assertions.

What This Document Contains

The document mixes three kinds of content, and it is useful to name them upfront so the reader knows the epistemic status of each claim:

- **Exact statistical results** — e.g. the Bernoulli/binomial model, Wilson score intervals, and the binomial order-statistic upper bound on a quantile. These are theorems, cited and verifiable.
- **Operational approximations** — e.g. normal-asymptotic sample-size planning formulas, and the Wilson lower bound used as a one-sided threshold. These are practical approximations chosen for their stability and familiarity; their limits are stated where they appear.
- **Engineering guardrails** — e.g. feasibility gates, the distinction between full-strength verification runs and lightweight smoke-test runs, covariate tracking, baseline expiration, transparent-statistics output. These are framework-enforced disciplines that make disciplined inference visible and auditable; they do not replace the underlying statistics.

Where a device below belongs to one of these categories, it is labelled. Readers should not expect design policies to have the calibration guarantees of theorems, or vice versa.

The mavai Project Family

The methodology is implemented across language-native frameworks:

Framework	Language	Role
punit	Java	JUnit -compatible reference implementation
feotest	Rust	Idiomatic Rust implementation (not a Java port)
baseltest	Python	(planned)

Each implements the same statistical core independently. Language-agnostic reference data for conformance testing is generated by the R-based mavai-R project; a per-language implementation is conformant when it reproduces those outputs within

stated tolerances. Detailed cross-language conformance architecture is documented in the project READMEs.

Document Purpose and Audience

This document provides a rigorous statistical treatment of the methods employed by the mavai methodology for probabilistic testing of systems characterised by uncertainty. It is intended for:

- **Professional statisticians** validating the mathematical foundations
- **Quality engineers** with statistical training designing test strategies
- **Auditors** who need to verify that testing methodology is sound
- **Technical leads** establishing organizational testing standards
- **Framework implementors** building mavai-compatible statistics engines in new languages

---

The Model in Brief: Contracts, Tests, Experiments

The rest of this document uses a small vocabulary — service contract, criterion, test, threshold, experiment, verdict — that it will lean on heavily. The terms are not exotic, but each carries a precise meaning in this methodology, and the chapters that follow are easier to navigate when they are introduced together.

The foundation is the **service contract**. A service contract is the definition of what a correct response from the service under test looks like. It is not a single yes/no proposition; it is a conjunction of separately checkable expectations, and each expectation is called a **criterion**. A criterion asks one yes/no question of a single response: did this output meet this expectation, or did it not? A contract typically declares several criteria — schema validity, semantic correctness, absence of disallowed content — and the framework evaluates every one of them against every response it observes.

A **test**, in this document, is what a developer or a CI system invokes to check that the service is behaving acceptably. Because the service is stochastic, a single response cannot settle the question; a test therefore invokes the service many times, applies the contract's criteria to each response, and aggregates the per-criterion outcomes into a **pass rate**. The test then compares that pass rate against a **threshold** and reports a per-criterion **verdict**: PASS, FAIL, or INCONCLUSIVE.

A criterion's threshold comes from one of two places. A **normative** threshold is a fixed value the contract is told to meet — typically by an SLA, an SLO, a policy, or a regulation. An **empirical** threshold is derived from the service's own measured behaviour: the test does not ask whether the service meets some externally mandated rate, but whether it has degraded from a baseline it previously achieved. Empirical thresholds require a baseline, and a baseline is produced by an **experiment**: a separate invocation of the service whose purpose is to record the rate at which the service currently satisfies each criterion. Tests configured with normative thresholds

do not need an experiment; tests configured with empirical thresholds consult the most recent experiment's baseline.

The remaining vocabulary is structural. Whether the invocations belong to a test or to an experiment, the list of inputs the framework executes against the service is called a **sampling** — a list of $N \geq 1$ inputs whose responses are scored against every criterion in the contract.

The bulk of this document is, in the end, an account of how those verdicts are computed under each threshold type, what they entitle a reader to claim, and what disciplines keep them honest. With that vocabulary in hand, the document can now look more closely at the kinds of expectation a contract can carry — which is the subject of the next section.

Clause Forms: Rate-Bounded and Categorical

A service contract is a conjunction of clauses, but the clauses are not all of one kind. Two structurally different forms coexist in a typical contract, and the rest of this document depends on keeping them apart. The distinction is orthogonal to the threshold-origin axis the companion already carries (compliance vs. regression, §3): a clause can be normative or empirical in origin — mandated by SLA/SLO/policy/regulation, or derived from a measured baseline — independently of the form the present section partitions on.

A **rate-bounded clause** states a rate-bounded proposition: ”the service shall satisfy criterion c at rate at least p_c^* with confidence γ over sampling V.” Its threshold p_c^* has one of two **origins** — **normative** (a.k.a. stipulated), an SLA-mandated rate evaluated by the compliance paradigm of §3, or **empirical**, a measured baseline evaluated by the regression paradigm of §3. The word empirical names that origin and is **not** the name of this form; compliance and regression are simply the decision-side names for the two origins. Either way the form is rate-bounded: a test invokes the service many times, and the companion's central machinery turns the resulting evidence into a per-criterion verdict.

A **categorical clause** states an obligation or prohibition that admits no rate threshold at all: ”the service shall not violate criterion c.” Full stop. The clause is satisfied or violated in kind, not in rate. Categorical clauses are, in practice, almost always normative in origin — a categorical prohibition is the kind of thing a contract mandates rather than something a team derives from baseline measurement — but their distinguishing feature is form rather than origin: no p_c^*, no γ, no Wilson construction, no rate-bounded discharge available. The canonical examples are categorical safety prohibitions: no self-harm advice, no PII leakage, no emission of illegal content.

Rate-bounded and categorical clauses are not two points on a strictness spectrum. Letting $p_c^* \to 1$ does not promote a rate-bounded clause into a categorical one: it merely tightens the rate-bound proposition. No sample budget, threshold, or confidence level redenominates a categorical claim as a frequentist one — observing zero failures in n trials yields, at best, a rule-of-three upper bound of $\approx 3/n$ on the failure

rate, which approaches zero only in the limit. A categorical clause therefore cannot be discharged by the rate-bounded apparatus, however that apparatus is parameterised.

Categorical clauses are discharged **architecturally**. A separate component — a guardrail, a deterministic filter, a hard schema constraint, a refusal classifier — is interposed between the stochastic system and the consumer of its output, and its presence is the contract's answer. The component is itself probabilistic; its performance re-enters the contract as one or more **ordinary rate-bounded clauses**, typically covering the component's false-negative and false-positive behaviour over an adversarial sampling representative of the inputs the component exists to filter. The clause form is fixed — rate-bounded — but the threshold origin is free: it may be **normative**, where a safety policy, regulator, SLO, or release gate mandates a minimum containment level, or **empirical**, where the clause is used for regression against a measured guardrail baseline. The origin axis is independent of the categorical parent these clauses evidence. The categorical clause itself is the obligation; the discharge is the architectural commitment.

The methodology summarised across the rest of this document is therefore:

> Tolerable failures are bounded statistically; intolerable failures are bounded architecturally; and the architecture itself is bounded statistically.

The rate-bounded/categorical distinction is foundational. The compliance/regression paradigms of the next section, the criterion decomposition of §1.4, and the report and verdict layers of §7 and §10 all operate within the rate-bounded class — across both normative-origin and empirically-derived thresholds. The architectural discharge of categorical clauses is sketched in §1.4.5 and developed fully in a forthcoming chapter on architectural commitments.

Two Testing Paradigms: Compliance and Regression

Within the rate-bounded class of contract clauses introduced in the previous section, the methodology supports two distinct testing paradigms. They share the same hypothesis-test skeleton but differ in where the threshold comes from and how results are interpreted.

Paradigm	Threshold source	Statistical question	Example
Compliance	Contract / SLA / SLO / policy — given, not estimated	”Does the system meet its mandated requirement?”	Payment API with contractual $p_{\text{SLA}} = 0.995$ uptime
Regression	Empirical estimate from a MEASURE experiment	”Has the system degraded from its measured baseline?”	LLM customer-service system with $\hat{p}_{\text{baseline}} = 0.951$ from $n = 1000$ samples

Both paradigms share a one-sided binomial decision skeleton, but they implement **two related but distinct procedures with different decision semantics and different error interpretations**. The methodology treats them as separate inferential acts: the controlled error rate, the direction of evidence, and the meaning of a PASS or FAIL differ between the two, and reports name which procedure produced a given verdict (§7, §10).

Compliance / assurance procedure — affirmative procedure used to show that a system meets an externally required threshold p_{req}:

$$H_0 : p \leq p_{\text{req}} \quad \text{(not shown acceptable)} \qquad H_1 : p > p_{\text{req}} \quad \text{(shown acceptable)}$$

PASS only if the one-sided lower confidence bound on p exceeds p_{req}. The procedure controls the long-run probability of falsely declaring compliance when the true success probability is at or below the requirement, subject to model assumptions and approximation limits.

Regression / monitoring procedure — reference-control procedure used to detect degradation from a measured reference behaviour. Choose an integer lower-tail cutoff c so that, under the stated reference model,

$$P_{\text{ref}}(K < c) \leq \alpha.$$

PASS if observed $K \geq c$; FAIL if $K < c$. The procedure controls false degradation alarms under the reference model. **A regression PASS means "no degradation signal at this cutoff," not "equivalence to the baseline has been proved"**, and a compliance PASS means "evidence supports compliance at the configured level," not "no degradation."

The two rules share implementation components — the binomial model, the Wilson machinery on the threshold side, the feasibility gate, the VERIFICATION/SMOKE distinction — but their PASS/FAIL labels are **not interchangeable**. The transparent-statistics output (§7, §10) records which procedure was applied, and reports use intent-specific verdict wording so that auditors and operators do not read a regression PASS as an affirmative compliance certification (or vice versa).

The regression hypothesis is equivalently expressible in rate form, and the regression-side prose of §§3–5 uses that form throughout:

$$H_0 : p \geq p^* \quad \text{(acceptable)} \qquad H_1 : p < p^* \quad \text{(unacceptable)}$$

where $p^* = c/n$ is the displayed rate corresponding to the integer cutoff c. The discreteness of the binomial decision means p^* is informational; the binding decision artefact is c (§3.4, §5.1).

The three remaining operational differences are the **source** of the threshold (given vs. derived), the **interpretation** of failure (SLA violation vs. regression), and the **prerequisite step** (none vs. MEASURE).

For an evidential claim under either paradigm (VERIFICATION intent), the sample size must be sufficient to support the threshold; otherwise the developer must declare the test as SMOKE. Section 5.7 defines this split.

---

1. Statistical Model

1.1 Bernoulli Trial Framework

Each invocation of the service contract is treated as a Bernoulli trial under a **working approximation** of independence and stationarity:

$$X_i \sim \text{Bernoulli}(p)$$

where:

- $X_i \in \{0, 1\}$ is the outcome of the i-th trial (1 = success, 0 = failure)
- $p \in [0, 1]$ is the true (unknown) success probability
- Trials are assumed independent and identically distributed (i.i.d.)

The word working is load-bearing: independence and stationarity are modelling assumptions that hold approximately in practice, not discovered truths about the system under test. §1.3.1 sets out why the assumption is nevertheless defensible as a starting point for a typical LLM experiment; §1.3.2 separates the formal assumptions from the operational threats to those assumptions; §8 describes the framework-level guardrails that make assumption drift visible without pretending to repair it.

1.2 Binomial Aggregation

For n independent trials, the total number of successes follows a binomial distribution:

$$K = \sum_{i=1}^{n} X_i \sim \text{Binomial}(n, p)$$

The sample proportion $\hat{p} = K/n$ is an unbiased estimator of p:

$$E[\hat{p}] = p, \quad \text{Var}(\hat{p}) = \frac{p(1-p)}{n}$$

1.3 Assumptions and Limitations

The Bernoulli model is a working approximation, not a discovered truth. §1.3.1 sets out why the assumption is reasonable for a typical LLM experiment; §1.3.2 enumerates the formal assumptions and the operational threats to each; §1.3.3 describes the developer's role in upholding the independence premise.

1.3.1 Why the Working Approximation is Defensible A natural objection to the Bernoulli model for LLM testing is that an LLM is not a coin: its outputs depend on weights that the test author neither owns nor controls, served by infrastructure that the test author cannot inspect. Why should the success probability p be treated as constant within an experimental run, given that something about the served system can change at any time?

The answer rests on a layered view of what determines a trial's outcome. In decreasing order of magnitude, a verdict is shaped by:

1. **Model weights** — the parameters that define the model's capabilities and dispositions.
2. **Serving stack** — quantisation, kernel choices, batching, speculative decoding, mixture-of-experts routing.
3. **Sampling configuration** — temperature, top-p, max tokens, set by the caller.
4. **Numerical realisation** — floating-point ordering on the inference hardware, sensitive to batch composition.
5. **Input distribution** — the prompts and contract context fed to the model.

Layer 1 is the dominant term and is the layer for which the assumption of stability is most defensible. Major LLM providers commit, as a matter of policy, that their model identifiers resolve to immutable weights for the lifetime of the model's deprecation window. Anthropic's documentation states this explicitly: ”Every Claude model ID is a pinned snapshot. Models with a date in the ID (for example, *20250929*) are fixed to that specific release. Starting with the Claude 4.6 generation, model IDs use a dateless format that is also a pinned snapshot, not an evergreen pointer” (Anthropic, 2026). For older generations, providers also expose **floating aliases** — convenience pointers that resolve to the current dated snapshot at call time and may be re-bound when a newer snapshot ships. Calling through a pinned snapshot fixes layer 1 across the experimental run by provider commitment; calling through a floating alias does not.

Empirical study has documented material drift in model behaviour over months when calls are routed through floating aliases (Chen, Zaharia & Zou, 2023). The methodology absorbs this finding: the prescribed posture is to test against pinned snapshots and to set explicit baseline-validity windows (§8.4.2) that require operators to reconfirm p before stale comparisons are drawn.

Pinned model IDs are necessary but not sufficient. Pinning the model identifier reduces one major source of non-stationarity — weight changes — but does not by itself guarantee stationarity of the full served system. Layers 2–5 above remain potential sources of drift even when layer 1 is fixed: serving-stack revisions (kernel updates, quantisation changes, batching policy, speculative-decoding configuration), routing changes (mixture-of-experts gating, regional load-balancing, edge-cache topology), safety-layer revisions (prompt classifiers, output filters, refusal heuristics), regional deployment variation (different provider regions running different stack revisions concurrently), and numerical-kernel changes can each shift observed behaviour while the model ID remains the same string. Provider documentation describes endpoint and routing configurations that vary across platforms (Anthropic enterprise documentation, accessed 2026-05-14), reinforcing the need to capture endpoint and deployment

metadata. The factor record (§1.5.2) therefore captures all observable service, endpoint, prompt, sampling, guardrail, and deployment metadata available to the test operator — not the model ID alone — and the guardrail severity policy of §8.4.5 treats endpoint and serving-stack changes as major caveats under VERIFICATION.

Layers 2 and 4 are the source of the well-documented residual non-determinism observed even at temperature zero with fixed sampling configuration: the same prompt to the same snapshot can yield different outputs in successive calls. This affects the variance of the trial outcome — exactly the phenomenon the binomial model is built to absorb — without changing p in the population sense the model requires. Provided the input distribution (layer 5) is also stable across the run — a fixed list of prompts, or independent draws from a stable generator — the verdict sequence is well-modelled as a sequence of Bernoulli trials with constant p.

The assumption is therefore defensible under stated conditions: a pinned model snapshot, a fixed system prompt, a fixed sampling configuration, no conversation state carried between calls, and a stable input-sampling process, all within a single experimental run of bounded wall-clock duration. Outside these conditions — most importantly, calls through a floating alias, or experiments whose duration spans a known provider rollout window — the assumption can fail, and the diagnostics of §8.3 and the guardrails of §8.4 exist for exactly that case.

1.3.2 Formal Assumptions and Operational Threats The Bernoulli model rests on three **formal assumptions**, each paired with the **operational threats** that could silently violate it in practice:

1. **Independence**: Each trial is independent. In practice, this may be violated if, for example:
   - The LLM provider implements request-level caching
   - Rate limiting causes correlated delays
   - Model state persists across requests (generally not the case for stateless APIs)
2. **Stationarity**: The success probability p is constant across trials. This may be violated if, for example:
   - A baseline is created at a time when the system was under load, but a test performed later, while the system IS NOT under load. Such a test will likely miss a drop in performance of the system.
   - Contextual factors differ between baseline creation and test execution (time of day, day of week, deployment region, etc.)

   The mavai methodology addresses stationarity through two complementary mechanisms:

   - **Covariates** (see Section 8.4.1): Explicit declaration and tracking of contextual factors that may influence success rates, with warnings when baseline and test contexts differ.
   - **Baseline expiration** (see Section 8.4.2): Time-based validity tracking that alerts operators when baselines may no longer represent current system

behaviour.

These features do not guarantee stationarity—that is impossible in practice—but they make non-stationarity **visible and auditable** rather than silently undermining inference.

3. **Binary outcomes**: Each trial has exactly two outcomes. Complex quality metrics may require more sophisticated models.

1.3.3 Developer Responsibility for Trial Independence While the framework provides the statistical machinery, **developers share responsibility** for ensuring that the independence assumption holds in practice. Many LLM service providers offer features that can introduce correlation between trials:

Feature	Risk to Independence	Mitigation
Cached system prompts	Cached prompts may produce more consistent outputs	Disable caching during experiments, or document its presence
Conversation context	Prior exchanges influence subsequent responses	Use fresh sessions for each trial
Request batching	Batched requests may share internal state	Submit requests individually
Seed parameters	Fixed seeds produce identical outputs	Use random or rotating seeds

Example: An LLM provider's ”cache system prompt” feature improves latency by reusing parsed prompts. While beneficial in production, this can reduce output variance during testing. The developer should either:

- Disable the cache via configuration during experiments
- Document that the measured success rate reflects cached behaviour
- Understand that the true variance may be higher than observed

No framework can detect or correct for these effects—they require domain knowledge and deliberate configuration choices. When trial independence is uncertain, developers should consider:

- Running sensitivity analyses with different configurations
- Documenting assumptions in test annotations or comments
- Increasing sample sizes to account for potential correlation (see Section 8.2)

Recommendation: For most LLM-based systems accessed via stateless APIs, the independence assumption is reasonable when caching and context features are disabled. Monitor for temporal drift in long-running experiments.

1.4 Criterion Decomposition of the Functional Dimension

1.4.1 Failure modes of differing kind Criterion decomposition operates within the functional dimension introduced earlier in the document. The partition this chapter develops refines the evidence inside that one dimension into multiple statistical streams; it does not multiply the number of dimensions of stochasticity.

The chapter uses a deliberately stark **triple** as its running example, drawn from a clinical-advice service:

- a structural failure — a response that does not parse as JSON;
- a judge-mediated failure — a response that parses, but whose register is wrong for the apparent literacy of the recipient (clinical jargon to a layperson, say); and
- a catastrophic failure — a response that advises a vulnerable user toward self-harm.

The three modes are not interchangeable, and each carries a distinct methodological load. Parseability and register are both rate-bounded criteria — they share clause type despite the gulf between their measurement apparatus (a JSON parser, a rubric-driven judge), and together they establish that the rate-bounded machinery of this chapter applies as readily to judge-mediated criteria as to mechanical ones. Self-harm is categorical rather than rate-bounded: it sits in a different clause type despite, like register, requiring a judge to detect, and the clause-type partition therefore does not run along the mechanical/judge-mediated axis. References to self-harm in what follows are technical illustrations of a categorical clause; they are not commentary on clinical practice or on the regulation of clinical AI systems.

A service contract's postconditions defend against failure modes that vary along at least three independent axes. Each axis is on its own sufficient to require separate statistical treatment of failure modes that differ along it, and each — at its extreme — also reveals the boundary between the rate-bounded and categorical clause classes introduced in Clause Forms: Rate-Bounded and Categorical.

Consequence. Parseability, register, and harm all violate the contract, but the cost of each violation to the consumer of the verdict spans a wide gradient, and the methodology distinguishes points on that gradient at two different levels. Within the rate-bounded class, where the cost is bounded and a non-zero failure rate is tolerable, consequence is operationalised as the per-criterion threshold p_c^*: register (medium-consequence) demands a tighter p_c^* than parseability (low-consequence), and both are evaluated by the same Wilson-against-threshold machinery despite the gulf between their measurement apparatus (a JSON parser, a rubric-driven judge). Across the rate-bounded/categorical boundary, where the project's stance is zero tolerance — harm advice, PII leakage, illegal content — consequence is no longer operationalised as a tight p_c^* at all. The failure mode is admitted as a categorical clause and discharged architecturally (§1.4.5; forthcoming chapter on architectural commitments). The observational mode of §1.4.5 is the mode used to evaluate the observational zero-failures criterion on the architectural component, not a limit-case of a rate-bounded clause at $p_c^* \to 1$.

Frequency. Baselines of rate-bounded criteria span a wide range — parse-failure rates in the low single percents, register-mismatch rates higher still, other rate-

bounded criteria potentially lower. The sample budget required to support strong evidential power scales inversely with the baseline, and rate-bounded peers with disparate baselines therefore make competing demands on the experiment's shared sampling. A thousand-sample sampling gives a Wilson interval around a 5%-baseline observation tight enough to support a binding evidential claim against a nearby threshold; as the baseline falls, the same pool gives an interval whose width relative to the observed rate grows — at a 0.1% baseline the Wilson interval spans several times the point estimate itself — and a criterion at that baseline either needs a larger pool to be binding under VERIFICATION intent, or must be declared SMOKE up front. Per-criterion feasibility gates surface this asymmetry honestly, where an aggregated stream absorbs it. The categorical clause sits off this axis: a 'zero-failures' failure mode is not admitted to the contract as a rate-bounded criterion at all, and so has no baseline to enter the comparison.

Input requirement. A MEASURE experiment runs over a single sampling: every sample produces a response, and every criterion evaluates its postcondition(s) on the same response. The criteria share the input distribution because they share the inputs themselves. A sampling sized for parseability serves register and other rate-bounded judge-mediated criteria without trouble, because their baselines are commensurate. What it cannot do is discharge a 'zero-failures' failure mode: the rule-of-three bound on the failure rate with zero observed failures is $\approx 3/n$, which approaches zero only in the limit, and no n reachable inside one experiment closes the gap. The zero-failures case is therefore discharged at the architectural layer (see Clause Forms: Rate-Bounded and Categorical); the architectural component is evaluated in a separate experiment over an adversarial sampling. The categorical postcondition may additionally be scored against the primary sampling as a SMOKE-intent diagnostic — the response is already produced, the judge is already running — giving a signal that complements the discharge (§1.4.5).

The methodology therefore partitions on two levels. Within the rate-bounded class, failure modes that differ along consequence, frequency, or input share are treated as **separately contractual** — each is its own hypothesis test, with its own threshold, its own confidence level, its own feasibility gate, sharing the experiment's sampling. Across the rate-bounded/categorical boundary, failure modes are routed out of the MEASURE experiment **for the purposes of contractual discharge** and into the architectural-commitment treatment; they may remain present in the experiment as SMOKE-intent diagnostic criteria, with the epistemic status set out in §1.4.5.

A representative example, threaded through the chapter, is a clinical-advice service whose contract carries:

- P_1: response parses as JSON — rate-bounded, low-consequence (parser-detected).
- P_2: required fields are present — rate-bounded, low-consequence (parser-detected).
- P_3: response is layperson-readable — rate-bounded, medium-consequence (judge-detected).
- P_4: advice does not suggest self-harm — **categorical**, catastrophic-consequence. Discharged by a dedicated guardrail component (e.g., a harm classifier interposed between the model and the user); the guardrail's false-negative behaviour

over an adversarial sampling is reported as an observational zero-failures criterion, and may additionally be scored on the production sampling as a SMOKE-intent diagnostic (§1.4.5).

The four postconditions are not interchangeable along the partition axes. A P_4 violation at any rate is clinically significant in a way a P_1 violation at 10^{-3} is not, and a P_3 violation at 5% is significant in a way a P_1 violation at 5% is not. The methodology evaluates the rate-bounded postconditions (P_1, P_2, P_3) each in its own statistical stream within the primary end-to-end MEASURE experiment, against its own threshold, at its own confidence level; P_4 is routed out of the primary end-to-end MEASURE experiment for contractual discharge. Its categorical obligation is discharged by the architectural commitment (§1.4.5); the guardrail's own rate-bounded criteria (normative or empirical in origin) may themselves be evaluated in separate MEASURE experiments over adversarial samplings (§1.4.8). The three primitives of §1.4.2 give the empirical partition its formal structure; the hiding result of §1.4.4 establishes that an aggregated stream over rate-bounded postconditions that differ along any of the three axes potentially obscures movement in a low-frequency, high-consequence, or designed-input criterion. Where rate-bounded postconditions defend against failure modes that are interchangeable along all three axes — equivalent consequences, comparable frequencies, the same input distribution — a single aggregated stream remains an adequate representation; the $m = 1$ instance of the per-criterion model recovers this case unchanged.

1.4.2 Three primitives The decomposition rests on three primitives, each playing a distinct role in the statistical model.

Sampling. The list of sample inputs posted to the service under test in a single experiment or test. A sampling has length $N \geq 1$; each sample entry in the list is presented once to the service, producing N responses. The sampling is shared across every criterion (defined below) that the contract exercises in the run: a contract with multiple criteria produces a per-trial vector of per-criterion observations over a single shared sampling, and there is no notion of a ”per-criterion sampling” within an experiment. A contract that requires evidence about a different input distribution runs a separate experiment with its own sampling. The inferential reach of a sampling is elaborated in §1.4.7.

Criterion. The unit of statistical evaluation. A criterion is the partition unit of the functional dimension; each criterion is exercised against the sampling and yields its own sequence of per-trial pass/fail outcomes, modelled as a Bernoulli stream (§1.4.3), from which its own verdict is computed. A criterion declares the mode under which the test is conducted (inferential or observational, §1.4.5), the threshold and threshold origin where applicable, the confidence level α, and the experiment or test in which the criterion is exercised (§1.4.7). It also hosts one or more postconditions (defined below), which together determine its per-trial outcome: a single-postcondition criterion produces a per-trial outcome equal to that postcondition's verdict; a multi-postcondition criterion produces a per-trial outcome equal to the conjunction of its hosted postconditions' verdicts (§1.4.3 makes this formal). Two postconditions whose

failures carry materially different consequences are therefore addressed by two distinct criteria, never by sharing a stream.

Postcondition. A named predicate over the produced output of a single trial. A postcondition has one job: decide pass or fail for a single observable property of the output. It carries no threshold and no statistical configuration of its own; the threshold, confidence level, and mode under which its per-trial verdicts are aggregated come from the criterion that hosts it. In the clinical-advice example, P_1 through P_4 above are postconditions.

The relationship between the primitives is:

Primitive	Role
Sampling	List of $N \geq 1$ samples posted to the service in one experiment or test; shared by all criteria of the run
Criterion	Statistical partition unit; exercised against the sampling; hosts 1+ postconditions
Postcondition	Per-trial predicate; hosted by a criterion

Each of the three primitives has exactly one job: the sampling supplies the N samples the run posts to the service and over which every criterion is exercised; criteria partition the functional dimension into statistical streams and parameterise each stream's inferential test; postconditions decide per-trial outcomes.

1.4.3 The per-criterion Bernoulli model Each criterion c defines a Bernoulli stream of per-trial pass/fail indicators $\{X_{i,c}\}$, aggregated to a Binomial success count $K_c = \sum_i X_{i,c}$ for inference; the per-stream parameter is p_c and the per-stream estimate is $\hat{p}_c = K_c/n_c$. A contract with a single criterion ($m = 1$) is the special case of this formulation in which the per-criterion stream coincides with the single Bernoulli/Binomial model of §§1.1–1.2; the multi-criterion machinery below covers $m \geq 1$ uniformly.

Let a contract declare m criteria $\{C_1, \dots, C_m\}$ (the symbol m for the number of criteria is used throughout this chapter and the rest of the companion; K is reserved for the **success count** of a Bernoulli stream as in §§1.1–1.2 — $K = \sum_i X_i$ — so the two must not collide). For each criterion c, let $\mathcal{P}_c$ denote the set of postconditions the criterion references. Let n_c denote the number of trials in the experiment's sampling; every sample is presented to every criterion, so n_c equals the sampling size N (§1.4.5a). On each trial define the per-criterion observation:

$$X_{i,c} = \begin{cases} 1 & \text{if } c \text{ produced a testable value and every } P \in \mathcal{P}_c \text{ holds on trial } i \\ 0 & \text{otherwise.} \end{cases}$$

A trial scores 0 both when the postcondition was evaluated and did not hold and when no testable value could be produced (a transform/no-value FAIL, §1.4.5a); the two are

distinguished by a diagnostic reason, not in the arithmetic. The effective denominator is $n_c = N$ — every trial of the sampling — and the success count is $K_c = \sum_i X_{i,c}$, so $\hat{p}_c = K_c / n_c$.

Each per-criterion trial is modelled exactly as the single-criterion trial of §§1.1–1.2:

$$X_{i,c} \sim \text{Bernoulli}(p_c), \qquad K_c = \sum_{i=1}^{n_c} X_{i,c} \sim \text{Binomial}(n_c, p_c), \qquad \hat{p}_c = K_c / n_c.$$

The Wilson construction, threshold-derivation pipeline, feasibility gate, and verdict-evaluation rule apply to this effective stream.

The independence and stationarity working approximations of §1.3 apply per criterion. Within a criterion, the per-trial observations across the experiment's N samples are treated as i.i.d. under the input distribution the sampling represents. Across criteria, the methodology does not assume independence: the per-criterion trials $X_{i,1}, \dots, X_{i,m}$ on a single invocation are typically correlated, since an invocation that produces a malformed response may also fail criteria that evaluate the response's content. Cross-criterion dependence does not affect per-criterion verdict correctness; its consequence for the composite verdict is the disclosed Type-I envelope of §1.4.6.

Each per-criterion trial is a complete statistical object in its own right. From here forward in the companion, the single-criterion X_i, $\hat{p}$, α, p^* of §§1.1–1.2 may be read as the per-criterion $X_{i,c}$, $\hat{p}_c$, α_c, p_c^* for any criterion c declared on a contract: the Wilson construction, the threshold-derivation pipeline, the sample-size derivations, the feasibility gate, and the verdict-evaluation rule apply unchanged.

1.4.4 Why aggregation masks per-criterion failures Aggregating per-criterion indicators into a single conjunction rate is the obvious temptation — one stream, one p, one Wilson interval — and this section sets out why it is the wrong move. A one-line union-bound argument shows that the aggregate rate is bounded above by the sum of per-criterion failure rates and dominated by the largest of them; small but consequential per-criterion failure rates are absorbed into the noise of the dominant ones and become statistically invisible. Per-criterion partitioning is what exposes them; the conjunction view buries them.

Let $\{C_1, \dots, C_m\}$ be the contract's criteria, with per-trial indicators $X_{i,c}$ as in §1.4.3. Define the conjunction indicator

$$X_i = \prod_{c=1}^{m} X_{i,c}, \qquad p = \mathbb{P}(X_i = 1)$$

— the rate at which every criterion's indicator simultaneously equals one on a trial. By the union bound applied to the complementary events,

$$1 - p \;=\; \mathbb{P}\big(\exists\, c : X_{i,c} = 0\big) \;\leq\; \sum_{c=1}^{m} (1 - p_c)$$

with equality if and only if the per-criterion failure events are disjoint. The aggregate failure rate $1-p$ is therefore bounded above by the sum of per-criterion failure rates, and in typical cases is dominated by the largest of them.

Corollary. If criterion c^* has a per-criterion failure rate that is small relative to the largest per-criterion failure rate — say $1 - p_{c^*} \ll \max_{c \neq c^*}(1 - p_c)$ — then $1 - p$ is dominated by the larger terms and is essentially insensitive to $1 - p_{c^*}$. A change in $1-p_{c^*}$ of any magnitude substantially smaller than $\max_{c \neq c^*}(1-p_c)$ produces a change in $1-p$ within the sampling noise of an aggregate estimator. A conjunction indicator therefore cannot detect movement in $1 - p_{c^*}$ unless that movement is comparable to the noise of the dominant criterion. The masking mechanism is purely a property of frequency: rare per-criterion failure rates are absorbed by more frequent ones, regardless of what those failures mean. The reason this matters for asymmetric contracts is empirical, not constructive: catastrophic-consequence outcomes (a model that recommends self-harm, a payment service that double-charges, a clinical-advice system that produces an unsafe instruction) are, in practice, rare, often deliberately so — modern LLM safeguards, hard-coded blocklists, and validation layers exist precisely to drive their rate toward zero. Rarity is what makes them mathematically invisible to aggregation; gravity is what makes that invisibility intolerable.

Inverse statement. Given an observed conjunction rate $\hat{p}$, the per-criterion rates $\{\hat{p}_c\}$ are not identified: any allocation of $1-\hat{p}$ among the per-criterion failure events that respects the union bound is consistent with the observation. Recovering the per-criterion rates requires per-criterion attribution at trial record time — the wide trial record that §1.4.3's indicators demand. A trial archive that has not preserved per-postcondition outcomes per trial cannot be decomposed retrospectively.

These two facts together make per-criterion partitioning the only faithful representation of a contract whose postconditions defend against failure modes of differing consequence.

1.4.5 Inferential and observational criteria A criterion operates in one of two distinct modes. The distinction is not a matter of statistical convenience; it reflects two different kinds of question. In either mode the criterion delivers one of three results: **PASS**, **FAIL**, or **INCONCLUSIVE**.

Inferential criterion. Estimates a population parameter p_c from the observed sample and tests against a threshold. The threshold p_c^* is either contractual (origin SLA, SLO, POLICY) or empirically derived (origin EMPIRICAL). Subject to the feasibility gate of §8.4, the verdict follows a procedure-direction-specific decision rule (§3.2, §5.1): a **compliance** criterion issues PASS iff the Wilson lower bound $\hat{p}_{c,L}(\alpha_c)$ clears p_{req}; a **regression** criterion issues PASS iff the observed success count K_c meets or exceeds the integer cutoff c_c derived from the reference distribution at α_c (§3.4).

Where the gate does not admit a verdict — the sample is too small to support an inferential claim at the stated threshold and α_c, or $n_c = 0$ — the verdict is INCONCLUSIVE. The inferential mode is the appropriate choice for criteria whose contractual question is "what is the true rate of behaviour c, with what confidence, and does it clear the demanded threshold?".

Observational criterion. Reports whether any failure of the criterion was observed in the run. No population estimation; no confidence interval; no threshold parameter. The test is on the observation itself:

$$\text{verdict}(c) \;=\; \begin{cases} \text{PASS} & \text{if } K_c = n_c \text{ and } n_c > 0 \\ \text{FAIL} & \text{if } K_c < n_c \\ \text{INCONCLUSIVE} & \text{if } n_c = 0. \end{cases}$$

A PASS verdict says exactly: zero failures of c were observed in the n_c trials of the experiment's sampling. INCONCLUSIVE indicates that no observation of the criterion was available in the run.

In this formula n_c is the criterion's number of trials, equal to the sampling size N (§1.4.5a). An observational (`zeroFailures`) criterion estimates no proportion and so carries no rate denominator: it is PASS when no failure is observed across its trials, FAIL on any observed failure (including a transform/no-value failure), and INCONCLUSIVE only when there were no trials ($n_c = 0$).

Exact. The observational verdict is deterministic given the run's observations. It makes no claim about p_c. A passing observational verdict at $n_c = 1000$ means exactly: no failure of criterion c was observed in 1000 trials of the experiment's sampling. It does not entail any bound on the true population rate of such failures. A contract that also requires a population-level claim attaches an additional inferential criterion against the same postconditions; the two criteria coexist on the contract, ask different questions, and produce different verdicts.

Engineering guardrail. A criterion declared as observational is not silently transformed into an inferential criterion at threshold $p_c^* = 1$, and the framework does not accept the literal threshold 1.0 as an inferential parameter. The Wilson lower bound at $\hat{p}_c = 1$ is strictly less than 1 for every finite n_c and every $\alpha_c < 1$; an inferential test against $p_c^* = 1$ cannot pass at any finite sample size. The observational mode is the honest expression of zero-failures contracts: the verdict reports the observation, and the population claim is not asserted from the observation alone; where one is required it is supported architecturally by the guardrail-validation pattern (§1.4.5b), or by separate production monitoring, which this methodology does not currently develop.

Epistemic status of observational mode. What an observational verdict means depends on the clause it serves (see Clause Forms: Rate-Bounded and Categorical). The methodology recognises three uses:

1. Architectural-commitment evidence. The verdict evaluates an observational zero-failures criterion on an architectural component — typically a guardrail's

false-negative behaviour over an adversarial sampling. The rule-of-three annotation above quantifies it. This is the methodology's recommended path for categorical clauses; the report reads such a verdict alongside the architectural commitment it evidences (§7, §10).

2. Diagnostic, complementary. The verdict scores the categorical postcondition against the experiment's sampling as a SMOKE-intent criterion. The response is already produced and the judge is already running, so the cost is incidental. The verdict does not discharge the categorical clause — the rule-of-three bound on a representative sampling sits orders of magnitude above any zero-failures rate — but it answers a question the adversarial-sampling evaluation cannot: do failures appear in representative production traffic at all? A non-zero result is loud, because the rest of the architecture should be suppressing it. Recommended wherever a categorical clause has been routed architecturally.

3. Diagnostic, gap-signal. The verdict accompanies a categorical clause that has not been routed through an architectural commitment, with no observational zero-failures criterion paired with it. It records whether a failure was observed but discharges nothing; the report flags it as a gap in the contract's architectural treatment.

The pre-existing mechanics — the literal-1.0 prohibition on inferential thresholds, the NO FAILURE OBSERVED relabel, the rule-of-three annotation — apply across all three uses.

The two modes coexist within a single contract. The clinical-advice example holds an observational criterion against P_4 alongside an inferential criterion against P_3 at $p^*_{P_3} = 0.98$, $\alpha = 0.001$, and a further inferential criterion conjoining P_1 and P_2 against an empirically derived threshold. The composite verdict (§1.4.6) is structured over the three criterion verdicts; the engineering response to a fired observational verdict is investigation, not threshold debate.

The methodology's ”Wilson everywhere” guideline applies to every inferential claim about a population parameter. Observational criteria make no inferential claim and lie outside the Wilson regime by construction.

Optional contextual zero-failure bound. An observational verdict makes no population-level claim. For transparent reporting only, the framework MAY annotate a passing observational verdict with a contextual upper bound on the failure probability under an explicitly-stated i.i.d. Bernoulli model. The classical ”rule of three” (Hanley & Lippman-Hand, 1983; cross-referenced in §9) gives, under that model, an approximate 95% upper bound of $3/n_c$ on the failure probability when zero failures are observed in n_c trials. Where shown, the annotation carries the explicit i.i.d. Bernoulli caveat and does not change the verdict label, which remains observational. The methodology specifies the report wording as:

> No failure of C_c was observed in n_c trials. This criterion is configured as observational, so the verdict itself makes no population-level claim. For context only, under an i.i.d. Bernoulli model, zero failures in n_c trials corresponds approximately to a 95% upper bound of $3/n_c$ on the failure probability.

High-consequence label override. For high-consequence safety criteria, the framework MAY (and the methodology recommends it) suppress the literal "PASS" label below an operator-configured minimum probe count and report **NO FAILURE OBSERVED** instead. The substitution is a label change only; the verdict semantics (observational, deterministic on the run) are unchanged.

1.4.5a The criterion denominator, and why a trial fails A criterion is evaluated on every trial of the sampling — every sample is presented to every criterion; there is no per-criterion filtering of inputs. On each trial the criterion yields exactly one of two outcomes:

- **PASS** — the criterion produced a value to test and its postcondition holds.
- **FAIL** — anything else. A FAIL carries a **reason**, which is diagnostic only and does not change the arithmetic:
  - condition — the postcondition was evaluated and did not hold;
  - transform / no-value — no value to test could be produced: the service returned malformed output, timed out, omitted required material, refused, or a transformation step failed, so the postcondition could not even be reached.

A criterion's denominator n_c is therefore simply the size N of the sampling; the success count K_c is the number that PASSed; $\hat{p}_c = K_c/n_c$. **Every trial counts.** A trial that produced no testable value is a FAIL like any other — its reason tells the developer it was not the postcondition that failed, but the trial is neither excluded from the denominator nor counted as a success. This is a single, uniform rule: there is no per-criterion denominator policy, no enum to choose, and no exclusion of unproducible trials.

> **Terminology.** Earlier drafts labelled the transform/no-value case INCONCLUSIVE at the per-trial level. That was a misnomer: such a trial is not "inconclusive" — it is a FAIL whose reason is recorded. INCONCLUSIVE has its proper meaning only at the **verdict** level (§1.4.5, §1.4.6): a criterion's verdict is INCONCLUSIVE when the statistical procedure cannot render PASS or FAIL at all — the feasibility gate is not met, $n_c = 0$, or covariates are misaligned. A per-trial transform failure is a FAIL; a criterion-level INCONCLUSIVE is a non-determination. The two must not be conflated.

Every sample is seen by every criterion, so nothing narrows a functional criterion's denominator below N. The one place a denominator legitimately narrows anywhere in the methodology is the latency dimension, whose statistics are conditional on success (§12): latency is summarised over the successful trials only, whereas a functional criterion's denominator is always the full sampling.

Availability as its own criterion. Because a non-response is already a FAIL of any criterion that needed the response, a quality criterion's rate is end-to-end — it reflects unavailability with no special handling. A contract that wants availability as a distinct, visible metric simply declares it as its own criterion (for example a `zeroFailures` or rate criterion over "the service returned usable output"). That is ordinary structural composition, not a denominator mechanism; there is no `denominatorPolicy` and no

`availabilityCriterionRef` gate on the denominator.

The transparent-statistics output (§10.2) exposes, per criterion, n_c, K_c, $\hat{p}_c$, and a breakdown of the FAILs **by reason** — in particular how many were transform/no-value failures rather than condition failures. A high transform/no-value share is itself a diagnostic signal: a service producing testable output only intermittently is structurally unhealthy in a way that warrants attention before the pass rate is read (§1.4.7).

1.4.5b Conditional containment model for architectural discharge Architectural discharge of a categorical clause leaves a statistical shadow, which this section models — taking care not to confuse the shadow with the obligation that casts it. The obligation is genuinely categorical, but it cannot be enforced at the level of the stochastic service: the service is a black box whose outputs vary by design, and a contract over it can only measure rates — and no rate expresses never. The categorical clause may perfectly well appear in the application-level service contract; what it cannot do is be discharged by estimating the stochastic generator's own success rate, because the generator can only be measured and no finite-sample measurement proves a ”never.” The discharge therefore belongs to the composed application architecture, not to the generator alone. The application's answer is to interpose an architectural boundary — a guardrail, a deterministic filter, a hard schema gate, or a refusal classifier — between the generator and the consumer; where that boundary is itself probabilistic it carries its own, closely related contract: an ordinary rate-bounded clause on how reliably it blocks the prohibited class of content. The categorical expectation is thereby relocated to the composite of generator-then-boundary, and the statistical question becomes the residual rate at which prohibited content clears both stages.

Write p_{harm} for the probability that the stochastic generator emits content violating the categorical clause, and $p_{\text{miss}|\text{harm}}$ for the conditional probability that the architectural boundary fails to block such content given that it was emitted. By the multiplication rule for the probability of a conjunction, the residual probability that prohibited content reaches the consumer is their product:

$$p_{\text{residual}} = p_{\text{harm}} \cdot p_{\text{miss}|\text{harm}}.$$

The decomposition separates two quantities with different evidential sources:

Quantity	Meaning	How it is evidenced
p_{harm}	The upstream generator's prevalence of harmful candidate outputs	Production monitoring, sentinel streams, representative production sampling, or not estimated

Quantity	Meaning	How it is evidenced
$p_{\text{miss}\|\text{harm}}$	The guardrail's conditional false-negative probability when presented with harmful content	Guardrail experiment over a harmful challenge sampling
p_{residual}	The end-to-end probability that harmful content reaches the user	Product of the two terms, if both are known or bounded under compatible population claims

Any production estimate of p_{harm} must observe the generator's pre-guardrail candidate output — directly, or through an equivalent shadow or sentinel stream that sees what the generator proposed before the guardrail suppressed or transformed it. Post-guardrail production monitoring sees only what reached the user, and therefore estimates p_{residual}, not the upstream prevalence p_{harm}. This is an easy mistake to make in practice, where many safety monitors are positioned downstream of the very guardrail whose containment they are meant to be independent of.

The architectural-discharge experiment primarily estimates or bounds the second term, $p_{\text{miss}|\text{harm}}$. It does not, by itself, estimate p_{harm}. A clean guardrail experiment therefore supports a statement of the form:

> Conditional on harmful content being presented to the guardrail, and relative to the specified challenge sampling or challenge distribution, the guardrail's miss probability is bounded above by the reported value at the configured confidence level.

It does not support, by itself, the statement:

> Harmful content reaches users in production at no more than this rate.

That stronger end-to-end claim requires a separate estimate or bound on the upstream generator harm prevalence p_{harm}, and the two quantities must be indexed to compatible populations.

Two distinct samplings. Evidence about this architecture is gathered over two samplings that are easily conflated and must be named apart, because they vary different things and estimate different quantities:

- A **red-team-input sampling** — adversarial inputs submitted to the generator-plus-guardrail application (the V_{probe} form of §1.4.8: prompts crafted to elicit prohibited content). Exercising it estimates the end-to-end behaviour of the composite under adversarial prompting, $P(\text{harm reaches user} \mid \text{adversarial prompt})$. It does not, by itself, isolate the guardrail's conditional sensitivity, because a clean run conflates "the generator rarely produced harmful candidates" with "the guardrail blocked the ones it did."

- A **harmful-candidate sampling** — prohibited candidate outputs presented directly to the guardrail (the V_{harm} form of the worked example below). Exercising it estimates the conditional containment term $P(\text{guardrail misses} \mid \text{harmful candidate presented})$, which is exactly $p_{\text{miss}|\text{harm}}$. It says nothing about how often the generator emits such candidates.

Both samplings are useful and both may appear in the same contract, but a sensitivity bound derived from the harmful-candidate sampling must not be reported as though it came from end-to-end adversarial probing, nor vice versa.

Guardrail sensitivity and miss probability. For a harmful challenge sampling of size n_H, let K_B be the number of harmful candidate outputs correctly blocked by the guardrail. Define the guardrail sensitivity estimate

$$\hat{s} = \frac{K_B}{n_H}.$$

The corresponding observed miss rate is

$$\hat{q} = 1 - \hat{s}.$$

Under the same Bernoulli/Wilson machinery used elsewhere in the methodology, an inferential guardrail-sensitivity criterion may report the one-sided Wilson lower bound

$$L_s(\alpha)$$

on the true sensitivity s — the probability that the guardrail blocks a harmful candidate output given one is presented to it. The corresponding upper confidence bound on the conditional miss probability is

$$U_{\text{miss}}(\alpha) = 1 - L_s(\alpha).$$

The residual end-to-end harm probability is then bounded conditionally as

$$p_{\text{residual}} \leq p_{\text{harm}} \cdot U_{\text{miss}}(\alpha),$$

provided the challenge sampling is admitted as representative of the harmful-output population to which the claim is being extended, and provided the production architecture admits no bypass around the guardrail.

The experiment has bounded the containment term, $U_{\text{miss}}(\alpha)$, not the upstream prevalence term p_{harm}.

Multiple boundaries. Real architectures often interpose several boundaries in series — a prompt classifier, a retrieval filter, an output classifier, a schema gate, a deterministic deny-list, a human-escalation step. The containment term is then the conditional probability that all relevant boundaries miss given harm, $P(M_1 \cap \cdots \cap M_r \mid H)$.

The individual miss probabilities must not be multiplied: that step assumes conditional independence, which logical composition does not supply (the same adversarial content that defeats one classifier is exactly the content likely to defeat another). Absent a declared and justified conditional-independence model, the term must be evaluated by the chain rule $P(M_1 \mid H)\, P(M_2 \mid H, M_1) \cdots$, or — more practically — estimated end-to-end over a harmful-candidate sampling presented to the full boundary stack.

Relationship to observational zero-failures criteria. The observational zero-failures criterion of §1.4.5 remains deterministic: it reports whether any miss was observed. It does not itself become an inferential criterion. Where a population-level statement about the guardrail's conditional miss probability is required, the contract may pair the observational criterion with an inferential guardrail-sensitivity criterion. The two need not — and typically do not — run over the same sampling: the observational criterion is naturally exercised over a red-team-input sampling (was any miss observed end-to-end under adversarial prompting?), while the inferential sensitivity criterion is exercised over a harmful-candidate sampling presented directly to the guardrail (what lower confidence bound is supported for the guardrail's sensitivity, and what upper bound follows for its miss probability?). When they are deliberately run over the same sampling, that shared sampling must be stated; it cannot be assumed.

False positives. A guardrail that blocks every candidate output has perfect sensitivity but no practical utility. For this reason the guardrail's false-positive behaviour should be modelled separately over a non-harmful challenge sampling. Write p_{fp} for the false-positive probability — the probability that the guardrail blocks content given that the content is non-harmful — and $p_{\text{specificity}} = 1 - p_{\text{fp}}$ for the corresponding specificity.

The specificity criterion is then constructed symmetrically with the sensitivity criterion above, under the same Wilson machinery. For a non-harmful challenge sampling of size n_N, let K_A be the number of non-harmful candidate outputs correctly allowed by the guardrail, giving the observed specificity

$$\hat{t} = \frac{K_A}{n_N}.$$

An inferential specificity criterion may report the one-sided Wilson lower bound $L_t(\alpha)$ on the true specificity, and hence the upper confidence bound on the false-positive probability

$$U_{\text{fp}}(\alpha) = 1 - L_t(\alpha).$$

Safety-class contracts typically require both a high-sensitivity criterion over harmful probes and an acceptable false-positive or specificity criterion over non-harmful probes, reported as separate criteria over separately named harmful and non-harmful challenge samplings. The former supports the categorical discharge; the latter con-

trols over-refusal, usability, and service continuity. Reporting the two bounds together makes over-refusal evidence as inspectable as under-blocking evidence.

Population-claim discipline. The challenge sampling's `populationClaim` governs the reach of the evidence. If the harmful probe set is a finite corpus, the result is a claim about that corpus. If the operator declares a superpopulation claim, the verdict must name the challenge distribution and record the representativeness argument. If the probes are exploratory, the result is no-generalisation evidence. In no case does an adversarial challenge result automatically become a production-traffic claim.

Stratification and the target estimand. A harmful challenge sampling is rarely homogeneous: it may span failure modes (self-harm, medical misinformation, jailbreaks, PII leakage, illegal-content requests), severity bands, languages, prompt styles, and template families. A pooled sensitivity bound can therefore look excellent overall while concealing weak containment in a rare but critical stratum. The general machinery for this is already in place (§8.2.1, §8.4.6): the verdict must declare its `targetEstimand` — call-weighted, prompt-weighted, stratum-weighted, severity-weighted, or no-generalisation — and where the harmful-candidate set is stratified by failure mode or severity the report should either give per-stratum sensitivity bounds or state the weighting scheme by which the pooled bound is to be read. A pooled bound must not be read as a guarantee for every harm stratum unless the per-stratum evidence supports that reading. For safety work this is not statistical fastidiousness: it is what stops a guardrail that is excellent in aggregate from quietly failing a small high-severity class.

The model therefore licenses only conditional claims — a bound on the guardrail's miss probability, and, given a separate estimate of p_{harm}, a bound on the residual harm rate under the stated population and no-bypass assumptions — never a claim that the categorical clause has been proved. In short: the guardrail experiment does not estimate how often the generator is dangerous; it estimates how much containment remains when it is.

Worked example: guardrail containment for clinical advice A clinical-advice service contains a categorical clause:

> The service shall not emit harmful medical advice.

The clause is discharged architecturally by interposing a guardrail between the LLM and the user. The guardrail is tested separately over a harmful challenge sampling

$$V_{\text{harm}}$$

containing $n_H = 100{,}000$ pseudo-LLM responses classified as harmful by the project's safety taxonomy and attested independent from the guardrail's training data.

The guardrail blocks $K_B = 99{,}990$ of the harmful candidate outputs.

The observed sensitivity is

$$\hat{s} = \frac{99{,}990}{100{,}000} = 0.9999.$$

The Wilson calculation that follows requires V_{harm} to be declared a superpopulation sampling from the named harmful-candidate distribution (`populationClaim: superpopulation`); the bound is then a confidence statement about that distribution. If instead V_{harm} is an exhaustively evaluated finite corpus, the corpus sensitivity is known exactly as $99{,}990/100{,}000$ — there is no sampling uncertainty for the corpus rate, and the Wilson interval is not a confidence interval for that finite-corpus estimand. Any extension beyond the corpus is then a separate superpopulation claim, declared and argued as such (§8.4.6). The calculation below assumes the superpopulation declaration.

The calculation also treats the $n_H = 100{,}000$ harmful trials as independent Bernoulli observations under that declared estimand. This must be checked, not assumed: if the 100,000 candidates were generated from a much smaller number of prompt templates, model outputs, paraphrase families, or synthetic augmentations — as "pseudo-LLM responses" readily can be — the effective evidence is far below 100,000 and plain Wilson calibration overstates it. Where the challenge set carries such material clustering, the criterion must use an estimator appropriate to the design (§8.2.1) or demote the claim per §8.4.6, exactly as for any other rate-bounded criterion.

Using the one-sided Wilson lower bound at $\alpha = 0.05$ gives

$$L_s(0.05) \approx 0.999833.$$

The corresponding upper confidence bound on the conditional miss probability is

$$U_{\text{miss}}(0.05) = 1 - 0.999833 \approx 0.000167.$$

Therefore, relative to the stated harmful challenge distribution, the guardrail experiment supports the statement:

> At one-sided 95% confidence, the guardrail's conditional miss probability on harmful candidate outputs is bounded above by approximately 1.67×10^{-4}.

Equivalently, the guardrail misses at most approximately 1.67 harmful candidate outputs per 10,000 harmful candidate outputs, at the stated confidence level and under the stated modelling assumptions.

The residual end-to-end harm probability is

$$p_{\text{residual}} = p_{\text{harm}} \cdot p_{\text{miss}|\text{harm}}.$$

The experiment has bounded the second term. It has not estimated the first. Thus the end-to-end claim remains conditional:

$$p_{\text{residual}} \leq p_{\text{harm}} \cdot 0.000167.$$

If, hypothetically, a separate production sentinel stream established

$$p_{\text{harm}} = 0.001,$$

then the residual harm probability would be bounded by

$$0.001 \times 0.000167 = 1.67 \times 10^{-7}.$$

That is approximately 1.67 harmful responses reaching the user per 10 million production interactions, subject to both estimates being valid for compatible populations and subject to the production architecture admitting no bypass around the guardrail.

Without the separate estimate of p_{harm}, the companion must not make the ”1.67 per 10 million” claim. It may only state the conditional containment result.

If the guardrail instead blocks all 100,000 harmful challenge outputs, the observed sensitivity is

$$\hat{s} = 1.0.$$

This is not evidence of perfection. The one-sided Wilson lower bound at $\alpha = 0.05$ is approximately

$$L_s(0.05) \approx 0.999973,$$

so the corresponding upper confidence bound on the conditional miss probability is

$$U_{\text{miss}}(0.05) \approx 2.71 \times 10^{-5}.$$

The correct statement is therefore:

> No misses were observed in 100,000 harmful challenge trials. Under the i.i.d. Bernoulli/Wilson model, the guardrail's conditional miss probability is bounded above by approximately 2.71×10^{-5} at one-sided 95% confidence, relative to the specified harmful challenge distribution.

The incorrect statement is:

> The guardrail is perfect.

The experiment supports a strong containment claim, not a zero-risk claim.

1.4.6 The composite verdict and its Type-I envelope A contract's verdict is a structured tuple over its per-criterion verdicts. The tuple is not collapsed to a single PASS/FAIL at the reporting layer; a flat boolean verdict surface would obscure per-criterion outcomes that the contract requires the consumer to see.

Structural composite. Let $V_c \in \{\text{PASS}, \text{FAIL}, \text{INCONCLUSIVE}\}$ denote the verdict of criterion c (§1.4.5). The contract's structural composite verdict is

$$V_{\text{contract}} = \begin{cases} \text{PASS} & \text{if } \forall\, c : V_c = \text{PASS} \\ \text{FAIL} & \text{if } \exists\, c : V_c = \text{FAIL} \\ \text{INCONCLUSIVE} & \text{otherwise.} \end{cases}$$

with the per-criterion verdicts retained in full on the composite artefact. A consumer reads the contract verdict and the supporting per-criterion verdicts in one place.

The structural composite is the methodology's representation of ”the contract is satisfied.” A baseline may carry an aggregate $\hat{p}$ over the conjunction of all postconditions as a descriptive statistic — useful for dashboards and trend reporting — but the aggregate is not threshold-bearing: thresholds are not derived from it and verdicts are not framed against it. The hiding result of §1.4.4 demonstrates why a conjunction-based threshold is structurally unable to detect the per-criterion movement the methodology exists to surface.

Type-I envelopes by procedure direction. Per-criterion procedures of §3.2 control different error events, and a single unlabelled ”composite Type-I envelope” mixes them. The methodology therefore reports two procedure-specific envelopes, each a union-bound aggregate over its own family of inferential criteria:

- **False-compliance envelope** — the long-run probability, bounded by the sum of per-criterion α_c over **compliance** criteria, of declaring one or more compliance criteria satisfied when the corresponding true rate is at or below its requirement:

$$\alpha_{\text{false-compliance}} \leq \sum_{c \in \text{compliance}} \alpha_c.$$

- **False-degradation-signal envelope** — the long-run probability, bounded by the sum of per-criterion α_c over **regression** criteria, of issuing one or more degradation signals when the stated reference model holds:

$$\alpha_{\text{false-degradation-signal}} \leq \sum_{c \in \text{regression}} \alpha_c.$$

A contract whose criteria are all of one kind reports the single corresponding envelope; a contract that mixes compliance and regression criteria reports both, labelled by direction. A combined unlabelled aggregate is ambiguous because the two envelopes describe different error events and may, in a given contract, be set at different magnitudes.

Exact. Each bound holds under arbitrary dependence among the per-criterion test statistics in its family; it is the union bound applied to the per-criterion Type-I events of that direction. Observational criteria do not contribute to either envelope because their verdicts are deterministic on observation; they carry no α.

Engineering guardrail. Per-criterion α_c is set by the consequence of false acceptance for criterion c. A safety-class criterion at $\alpha = 0.001$ holds at that level because the consequence of falsely accepting a safety regression demands it. The envelope $\sum_c \alpha_c$ is reported on the composite verdict as a disclosed property; it is not a control target, and per-criterion α_c is not adjusted to control it. A uniform reduction $\alpha_c \mapsto \alpha_c/m$ would lower the per-criterion power proportionally — most consequentially in the safety case, where power to detect a true regression is the property the criterion most needs to preserve.

Projects whose governance demands a uniform composite α may apply a Bonferroni reduction at the operator level. The methodology does not impose it.

The companion's §7.3 addresses **cross-test** family-wise inflation when multiple independent contracts are evaluated in a suite. The §1.4.6 envelope addresses **intra-test** family-wise inflation across the criteria of a single contract. The two cases differ in source and in treatment.

1.4.7 Inferential reach of the sampling An experiment or probabilistic test is bound to a specific contract; the only thing it varies is the **sampling** — the single list of N samples it posts to the service. That one sampling is shared by every criterion the contract declares, and each criterion evaluates all N samples independently (§1.4.2); there is no per-criterion sampling.

A criterion's verdict or evidence is, primarily, a claim about the sampling its experiment or test runs over (including observational PASS verdicts, which are claims about the samples observed rather than about a population). The Wilson lower bound $\hat{p}_{c,L}(\alpha_c)$, the regression integer cutoff, the observational PASS verdict — all are statements about the N samples in the experiment's sampling and the responses the service produced for them. Extending the claim to a different input distribution (production traffic that the service ultimately faces, a hypothetical superpopulation the sampling is taken to represent) requires further evidence; the methodology does not sanction the extension implicitly. The finite-corpus and superpopulation framings of §8.4.6 set out the two interpretive moves available.

Criteria within an experiment share the sampling and so share its inferential reach. Every per-criterion verdict in the experiment speaks to the same N samples; differences between per-criterion verdicts arise from differences in postcondition definitions, thresholds and confidence levels, not from differences in the input distribution. A contract that requires evidence about a different input distribution runs a separate experiment whose sampling is drawn from that distribution; each experiment names its own sampling.

Verdict surface. Every per-criterion verdict names the sampling its experiment ran over — identifier, version, any policy-redaction metadata — so the reader knows ex-

actly which N samples produced the evidence. Extending the claim to anything other than that sampling is the reader's interpretive move, not the verdict's.

Clinical-advice illustration. A contract that runs both a primary MEASURE experiment over a representative production sampling and an architectural-discharge experiment over an adversarial probe sampling holds two samplings — one per experiment. The discharge experiment's verdict ("no failure observed under adversarial probing of the self-harm failure mode across 200 samples") is a statement about that probe sampling; it does not extend to a claim about self-harm advice in production traffic. The latter claim, if needed, requires sentinel monitoring or a SMOKE-intent diagnostic on the production sampling (§1.4.5); the methodology does not infer it from the probe-experiment verdict.

Sample-budget consequence. A contract with E experiments has E samplings, each of size N_e. Criteria within an experiment share N_e as their trial count (§1.4.5a); the limiting criterion within an experiment determines the experiment's required N_e. The total sample budget of the contract is bounded above by $\sum_e N_e$; the limiting criterion contract-wide is identified by per-experiment inspection, not by an aggregate sample count.

1.4.8 Worked example: the consult-advice contract The clinical-advice contract has been threaded through this chapter; it is collected here as a single inspection point. The contract runs as **three MEASURE experiments**, each with its own sampling — one for the structural criterion, one for the layperson-readability criterion, and one for the architectural-discharge of the categorical P_4 clause. Per-criterion verdicts are combined into a contract-level verdict in the usual way.

Postconditions. Four named predicates over the produced response.

- P_1: response parses as JSON
- P_2: required fields are present
- P_3: response is layperson-readable
- P_4: advice does not suggest self-harm

Samplings.

- V_{prod}: a representative sample of production prompts, redacted per policy XYZ, version 5.
- V_{probe}: a curated set of adversarial inputs designed to elicit self-harm advice; each example classified by failure mode within the `mavai-safety-taxonomy` v2; attested independent from the consult-advice guardrail's training data; version 3.
- $V_{\text{complexity}}$: a curated set of inputs likely to elicit responses involving clinical terms a layperson would not recognise; version 2.

Criteria.

- $C_{\text{well-formed}}$: inferential, references $\{P_1, P_2\}$, runs against V_{prod}, empirical-origin threshold against baseline `consult-advice@2026-04-01`, $\alpha = 0.05$, class `COR-`

RECTNESS. Its denominator is the full sampling (§1.4.5a); a malformed or absent response is a FAIL (reason: transform/no-value), counted like any other.

- $C_{\text{no-self-harm}}$: observational (`zeroFailures`), references $\{P_4\}$, runs against V_{probe}, no threshold parameter, class SAFETY. As an observational criterion it carries no rate denominator (§1.4.5a): one observed self-harm response fails it.
- $C_{\text{layperson-readable}}$: inferential, references $\{P_3\}$, runs against $V_{\text{complexity}}$, contractual-origin threshold $p^*_{P_3} = 0.98$ at SLO origin, $\alpha = 0.001$, class SAFETY. Its denominator is the full sampling (§1.4.5a); a malformed or absent response is a FAIL (reason: transform/no-value), not an exclusion.

If the contract additionally wants availability as a distinct, visible metric, it declares a fourth criterion:

- $C_{\text{evaluable-response}}$: a criterion (`zeroFailures` or rate) over a response-evaluability postcondition, run against V_{prod}. Like every criterion its denominator is the full sampling; it stands alongside the others and does **not** alter their denominators.

A single run's verdict. A measurement run executes the contract with $n_{V_{\text{prod}}} = 1000$, $n_{V_{\text{probe}}} = 200$, $n_{V_{\text{complexity}}} = 800$, and observes:

- $\hat{p}_{C_{\text{well-formed}}} = 0.953$ over 1000 trials.
- $K_{C_{\text{no-self-harm}}} = 200$ — zero observed self-harm responses in 200 probes.
- $\hat{p}_{C_{\text{layperson-readable}}} = 0.985$ over 800 trials.

The Wilson lower bounds, at per-criterion α:

- $\hat{p}_{C_{\text{well-formed}},L}(0.05) \approx 0.940$
- $\hat{p}_{C_{\text{layperson-readable}},L}(0.001) \approx 0.9649$

Per-criterion verdicts:

- $V_{C_{\text{well-formed}}}$: PASS against the empirical threshold derived from the baseline (the derived threshold is $p^*_{C_{\text{well-formed}}} = 0.9385$ at one-sided $\alpha = 0.05$).
- $V_{C_{\text{no-self-harm}}}$: PASS — observational, zero failures observed across V_{probe}. The verdict carries no inferential claim; the statement made is exactly "no failure of $C_{\text{no-self-harm}}$ was observed across 200 probe trials drawn from V_{probe} v3."
- $V_{C_{\text{layperson-readable}}}$: FAIL — $0.9649 < 0.98$.

Structural composite verdict:

$$V_{\text{contract}} \;=\; \text{FAIL} \quad (\text{triggered by } C_{\text{layperson-readable}})$$

Composite Type-I envelopes (procedure-direction-specific, §1.4.6):

$$\alpha_{\text{false-degradation-signal}} \;\leq\; \sum_{c \,\in\, \text{regression}} \alpha_c \;=\; 0.05 \quad (C_{\text{well-formed}})$$

$$\alpha_{\text{false-compliance}} \;\leq\; \sum_{c \,\in\, \text{compliance}} \alpha_c \;=\; 0.001 \quad (C_{\text{layperson-readable}})$$

The observational $C_{\text{no-self-harm}}$ contributes to neither.

The verdict report carries all three per-criterion verdicts, the envelope, the sampling references, the baseline reference for the empirical criterion, and the threshold provenance for the contractual criterion. The reader of the verdict — operator, auditor, regulator — sees the structural conclusion, the supporting per-criterion evidence, the populations each piece of evidence speaks to, and the disclosed false-alarm budget under which the conclusion was issued.

Under the clause-type taxonomy (see Clause Forms: Rate-Bounded and Categorical), $C_{\text{no-self-harm}}$ in this example is the observational zero-failures criterion evaluating the consult-advice guardrail's false-negative behaviour over V_{probe}. The associated categorical clause — the consult-advice service shall not emit self-harm advice — is discharged not by $C_{\text{no-self-harm}}$ but by the architectural commitment that interposes the guardrail between the model and the user. $C_{\text{no-self-harm}}$ provides the empirical evidence for the commitment's reliability; the commitment is the contract's answer to the categorical clause. A report reader who sees $V_{C_{\text{no-self-harm}}} = \text{PASS}$ is reading evidence that the guardrail held over 200 probes, not evidence that the model would not have erred without it. The full apparatus — how the commitment is declared, how V_{probe}'s coverage premise is itself recorded, and how the guardrail's false-positive rate enters the contract alongside its false-negative rate — is the subject of the forthcoming architectural-commitments chapter.

1.4.9 Per-criterion trials in subsequent chapters Each per-criterion trial is a complete statistical object — independent of the others as a unit of modelling, though not necessarily statistically independent of them. The remainder of the companion treats per-criterion trials transparently: where §§2–12 develop estimation, threshold derivation, sample sizing, the feasibility gate, transparent-statistics reporting, and latency, the single-criterion X_i, $\hat{p}$, α, and p^* they discuss apply equally to each per-criterion trial $X_{i,c}$, $\hat{p}_c$, α_c, and p^*_c. No section that follows needs to be re-derived for the per-criterion case.

The composite verdict over the per-criterion verdicts (§1.4.6) and the procedure-direction-specific Type-I envelopes of the composite (§1.4.6) are the constructs particular to the partition; they have no single-criterion analogue.

The following chapter develops one further statistical consequence:

- **Per-criterion baselines** — how a measurement run emits a baseline vectorised across criteria, and how thresholds are derived at resolution time rather than at emission.

1.5 Baselines

1.5.1 The baseline as the link between measurement and inference For an empirical-origin criterion (§7.4) the threshold p^* is a function of a prior observation rather than a contractual constant. The methodology resolves this in two phases. The

measurement phase yields an estimator $\hat{p}$ of the population rate p over n trials of the contract. The **inference phase** derives the threshold p^* from $\hat{p}$ at a stated confidence level α and resolves the service's current behaviour against it. The **baseline** is the statistical object that links the two phases: the estimator together with the conditions under which it was measured, which identify the population it estimates. Section 1.5.2 gives the formal definition, generalised per criterion.

1.5.2 The baseline as a family of per-criterion estimators A baseline is a family of per-criterion point estimators $\{\hat{p}_c\}$, one per criterion c declared on the contract. The family's structure follows §1.4.3: for each criterion c, the baseline stores:

- n_c, the number of trials for c, equal to the sampling size N (§1.4.5a);
- K_c, the number of those trials that PASSed;
- $\hat{p}_c = K_c/n_c$ for inferential criteria; the same pair feeds the deterministic rule for observational criteria (§1.4.5);
- optionally, for diagnostics, the split of the $n_c - K_c$ fails by reason (condition vs transform/no-value) — equivalently $n_{c,\text{evaluable}}$, the count that produced a testable value, and $r_{c,\text{obs}} = n_{c,\text{evaluable}}/n_c$.

The structural reference under which n_c is defined — the postconditions, the criterion, and its scope — is part of the criterion's meaning. A baseline and a test whose matching criteria differ in it differ structurally; under VERIFICATION the mismatch is a configuration error (§8.4.5).

The index of a baseline. The family $\{\hat{p}_c\}$ is interpretable as an estimator of the per-criterion population rates $\{p_c\}$ only relative to the conditions under which it was measured. A baseline is **indexed** by the pair (service-contract identity, resolved covariate values), and this index is its selection key: a test computes its own index and resolves against the baseline whose index matches (§1.5.3).

- The **service-contract identity** — in practice an ID or class name — names the contract whose criteria the baseline observes, and carries the **structural reference** (the postcondition-and-criterion structure under which the per-criterion trials are defined, fixing the meaning of each c in the family). A baseline and a test under different structural references are different contracts; the methodology treats that mismatch as a structural error (§8.4.5), not as an index divergence to be adjudicated. The service-contract identity is not a covariate.
- The **resolved covariate profile** (§8.4.1) — the operating regime of the measurement run, resolved to concrete values. §1.5.3 develops its role.

Two further things qualify a baseline without being part of its index:

- the **factor record** (§1.3.1) — service deployment, model, temperature, serving configuration — is provenance: a record of the system that was measured. It does not by itself drive selection. A project may — and in most cases should — declare the factor dimensions that matter (model ID, serving stack, and so on) **as covariates**, at which point their resolved values enter the index through the covariate channel; but the programming model does not force the factor record and the covariate profile to coincide.

- an optional **expiration window** (§8.4.2) — where declared, the temporal scope beyond which the baseline is no longer admitted as a reference, bounding the stationarity assumption of §1.3 in time (§1.5.5).

Each of these — index components and qualifiers alike — is a property of the baseline as a whole, not of an individual $\hat{p}_c$ within it.

The per-criterion threshold p_c^* and confidence level α_c are **not** part of the baseline. They are properties of the inference that consumes the baseline, not of the observation that supports it; §1.5.4 develops the consequence.

1.5.3 Covariates and the baseline's interpretive scope The role of the covariate profile in the baseline is more consequential than any single component of its structure. It is the covariate profile that makes the baseline's observations interpretable as a reference for future tests.

Each $\hat{p}_c$ is conditional on the covariate profile. The working approximation of §1.3 — that trials are i.i.d. samples from a Bernoulli distribution with parameter p_c — holds at the population level under the contextual conditions of the measurement run. A measurement taken on a weekday morning in the EU under a specific deployment configuration produces a $\hat{p}_c$ that is an estimator of p_c under those conditions. The same service measured on a weekend evening in a different region under serving-stack pressure may yield a different population p_c. The covariate profile is what the baseline records to make the condition explicit, so the consuming test can ask whether its own context is comparable.

Comparability between baseline and test depends on covariate matching. A project may hold more than one baseline for a contract — for instance one per operating regime, distinguished by their index (for a fixed contract, by their resolved covariate values). A test resolves against the baseline whose index matches its own: where the test's covariate profile matches a baseline's, that baseline's $\hat{p}_c$ is admitted as a reference for the test's inference. Where no matching baseline is available, the methodology treats the divergence as diagnostic: the test may proceed under explicit acknowledgement that the nearest baseline was measured under different conditions, or it may decline to resolve empirically at all. The mechanism by which a project decides between these options is governed by the project's covariate policy (§8.4.1) and is operational rather than methodological; the methodology's contribution is to require that the appropriate baseline is selected by matching index, and that any inference made under a divergent index carries that divergence on the verdict.

The covariate profile applies to the baseline as a whole, not per criterion. The covariates the project declares — time of day, day of week, deployment region, serving-stack version below the granularity the factor record pins, and so on — are properties of the measurement run that affect every criterion's observations simultaneously. They are recorded once for the baseline, not once per criterion. A reader inspecting the baseline reads one covariate profile, regardless of how many criteria the baseline carries observations for.

The covariate profile is what makes the baseline an audit artefact. Without

it, the baseline is a vector of $\hat{p}_c$ values with no anchor; a regulator inspecting the baseline cannot tell whether the conditions under which the measurement was taken match the conditions under which the service operates in production. With it, the baseline carries the contextual identity of the measurement and is inspectable end to end: this is the factor record of what was measured, under these covariate conditions, with these per-criterion outcomes, valid until this expiration. The baseline's role in Appendix A — as the empirical-evidence layer of the methodology's artefact stack — rests on the covariate profile being a first-class component of the artefact.

The methodology does not require any specific set of covariates; it requires only that whatever covariates the project declares are recorded on every baseline and on every test that consumes a baseline. The §8.4.1 machinery defines the per-covariate matching discipline. This chapter elevates that machinery to a defining role in the baseline artefact.

1.5.4 Threshold derivation at resolution time The baseline is an estimator. The threshold against which a test compares the service's behaviour is derived at the moment the test resolves against the baseline, not at the moment the baseline is produced.

For an inferential criterion with origin EMPIRICAL, the threshold p_c^* is the Wilson lower bound (§2.3.1) centred on the **baseline's** $\hat{p}_c$ but evaluated at the **test's** sample size $n_{c,\text{test}}$ and the test's confidence level α_c:

$$p_c^* \;=\; \text{WilsonLB}\big(\hat{p}_c^{\text{baseline}};\; n_{c,\text{test}},\; \alpha_c\big).$$

The use of $n_{c,\text{test}}$ rather than n_c^{baseline} is deliberate and is justified in the existing §3 (Threshold Derivation): the threshold must be the bound below which a test of size $n_{c,\text{test}}$ would, under the null hypothesis that the population rate equals the baseline's centre, fail to land with probability α_c. Substituting the baseline's n_c would calibrate the threshold to a sample size other than the one the test actually draws, breaking the test's stated Type-I rate.

The baseline's n_c enters the methodology elsewhere: through the feasibility gate (§8.4), which refuses the EMPIRICAL origin when n_c^{baseline} is too small to support the inferential claim, and through the perfect-baseline treatment of §4, which substitutes the Wilson lower bound of the baseline for the point estimate when $\hat{p}_c^{\text{baseline}} = 1$. Resolution-time threshold derivation reads the baseline's centre; the baseline's sample size shapes the conditions under which that centre is admitted as a reference.

The deferral of threshold derivation to resolution time is what allows the same baseline to support more than one test that disagrees on α_c or on $n_{c,\text{test}}$. The baseline is an observation; the test is the inference; the inference's parameters belong to the test that draws it.

Engineering guardrail. A baseline consumed by a test contributes, to the test's verdict, the baseline's identifier, the per-criterion centre $\hat{p}_c^{\text{baseline}}$, and the baseline's

n_c. The verdict also records the test's $n_{c,\text{test}}$ and α_c, and the resulting p^*_c. An auditor reading the verdict has the full set of inputs to the Wilson construction and can reproduce the threshold value.

For an inferential criterion with origin SLA, SLO, or POLICY, the threshold is contractual and the baseline does not enter threshold derivation; the baseline's $\hat{p}_c$ may still be carried on the verdict as diagnostic context, but it does not parameterise the test. For an observational criterion there is no threshold; the baseline's per-criterion observation is carried as diagnostic context but is not consulted at verdict time.

1.5.5 Expiration A baseline may optionally carry an expiration window bounding its temporal scope. Where one is declared — per §8.4.2 — it bounds the baseline's validity (§1.5.2) to a half-open interval, beyond which the baseline is no longer admitted as a reference for inference. Expiration, when present, is at the baseline level rather than per-criterion: the temporal scope is a property of the operating regime the covariate profile encodes — time of day, day of week, deployment region, and the like — which conditions every $\hat{p}_c$ simultaneously. A baseline so bounded is stale uniformly; while current it supports every criterion's resolution at once. A baseline with no expiration window carries no temporal bound and is admitted until superseded by project or operational policy.

1.5.6 Worked example A baseline produced by a measurement run against the consult-advice contract of §1.4.8, under the measurement conditions identified below.

Measurement conditions.

- Factor record: `consult-advice-service@3.1`, model `claude-sonnet-4-5-20250929`, temperature `0.0`, system-prompt `consult-advice-prompt@5`.
- Structural reference: postcondition-and-criterion structure `consult-advice@5`.
- Covariate profile: `day_of_week = WEEKDAY`, `time_of_day = 08:00–12:00`, `region = EU`, `serving_stack = standard`.
- Expiration: `2026-08-13` (90 days from emission).

Per-criterion observations.

Criterion	n_c	K_c	$\hat{p}_c$
`well-formed`	1000	953	0.953
`no-self-harm`	200	200	1.000
`layperson-readable`	800	788	0.985

A test consuming this baseline. A subsequent test of the consult-advice contract under matching factor record and matching covariate profile resolves its per-criterion thresholds from the baseline:

- $C_{\text{well-formed}}$ (origin EMPIRICAL, $\alpha = 0.05$, test sample size $n_{\text{well-formed,test}} = 200$): $p^*_{\text{well-formed}} = \text{WilsonLB}(0.953;\ 200,\ 0.05) \approx 0.922$. The centre of the Wilson construction is the baseline's $\hat{p}_c$; the sample size in the construction is the test's. The threshold is recorded on the verdict together with the Wilson inputs and the baseline's identifier.
- $C_{\text{no-self-harm}}$ (observational): no threshold is derived; the test independently evaluates whether any failure of the criterion is observed in its own run. The baseline's observation is recorded as diagnostic context on the verdict.
- $C_{\text{layperson-readable}}$ (origin SLO, $p^* = 0.98$, $\alpha = 0.001$): the threshold is contractual; the baseline does not parameterise it. The baseline's $\hat{p}_{\text{layperson-readable}} = 0.985$ is recorded as diagnostic context.

A test under a non-matching covariate profile — say, region = US — declines to resolve against the baseline without explicit project-policy acknowledgement of the divergence. The verdict reports the divergence; the auditor reads it.

1.5.7 Baselines in subsequent chapters The baseline is the statistical object that links measurement to inference: a family of per-criterion estimators, qualified by the factor record, covariate profile, expiration window, and structural reference under which it was measured. Each baseline is recorded as a row of Appendix A's artefact stack.

The remainder of the companion treats baselines transparently. The existing §2 (Baseline Estimation) develops the single-criterion point-estimator and standard-error machinery, which applies per criterion under §1.4.3. The existing §3 (Threshold Derivation) develops the Wilson lower bound that §1.5.4 invokes at resolution time. The existing §4 (The Perfect Baseline Problem) treats the $\hat{p}_c = 1$ case, which is the common situation for observational criteria where every evaluable trial was a success. The existing §§8.4.1–8.4.2 develop the covariate and expiration machinery that §§1.5.3 and §1.5.5 condition on. No section that follows needs to be re-derived for the per-criterion case.

---

2. Baseline Estimation (Experiment Phase)

2.1 Point Estimation

Given n experimental trials with k successes, the maximum likelihood estimator (MLE) for p is:

$$\hat{p} = \frac{k}{n}$$

Example: In our JSON generation service contract, an experiment with n = 1000 trials yields k = 951 successes.

$$\hat{p} = \frac{951}{1000} = 0.951$$

2.2 Standard Error

The standard error of $\hat{p}$ quantifies the precision of the estimate:

$$\text{SE}(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Example:

$$\text{SE} = \sqrt{\frac{0.951 \times 0.049}{1000}} = \sqrt{0.0000466} \approx 0.00683$$

2.3 Confidence Intervals

The mavai methodology uses the Wilson score interval exclusively for all confidence interval calculations. This section documents the Wilson method and, for completeness, includes the Wald (normal approximation) method that statisticians may encounter in textbooks.

2.3.1 Wilson Score Interval (the mavai Method) Wilson score intervals are the methodology's default interval family because they are stable near the boundary, remain inside $[0, 1]$, are simple to reproduce across languages, and generally have **much better coverage behaviour** than Wald intervals in the regimes encountered in probabilistic software testing. Wilson intervals are **not exact finite-sample intervals**; they are a score-interval construction selected for operational stability and good practical coverage. Exact binomial methods (Clopper–Pearson) and Bayesian alternatives (e.g. Jeffreys, beta-binomial predictive — §4.5) remain available for projects that require different calibration or prior/predictive semantics.

In summary, Wilson:

- Has substantially better coverage than Wald across small n and proportions near 0 or 1, while remaining a score-test approximation rather than an exact procedure.
- Remains well-defined for proportions at or near the boundary (including $\hat{p} = 1$), where Wald collapses.
- Never produces bounds outside $[0, 1]$.

The Wilson interval endpoints are:

$$\frac{\hat{p} + \frac{z^2}{2n} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}$$

Example (95% CI for $\hat{p} = 0.951$, $n = 1000$):

$$\text{Lower} = \frac{0.951 + \frac{1.96^2}{2000} - 1.96\sqrt{\frac{0.951 \times 0.049}{1000} + \frac{1.96^2}{4000000}}}{1 + \frac{1.96^2}{1000}} \approx 0.936$$

$$\text{Upper} \approx 0.963$$

Why Wilson Exclusively? The mavai methodology uses Wilson for all calculations because:

1. **Wilson is rarely worse than Wald in practice**: For large samples and moderate proportions, Wilson produces results nearly identical to the Wald approximation, so there is no operational penalty for using Wilson universally. (This is a comparative statement against Wald, not a claim of exactness against the binomial. See Brown, Cai & DasGupta, 2001; Newcombe, 1998.)
2. **Wilson avoids the worst Wald pathologies**: For small samples, extreme proportions, or perfect baselines ($\hat{p} = 1$), Wald produces degenerate or nonsensical results (zero-width intervals at the boundary, intervals outside $[0, 1]$). Wilson remains well-defined in all these cases. It is not exact — the operative claim is much-improved coverage behaviour, not perfect coverage.
3. **Consistency**: A single method ensures results are always comparable across tests. No edge cases where method switching affects verdicts.
4. **LLM testing reality**: High success rates ($p > 0.85$) are common in probabilistic testing of LLM-based systems. This is precisely where Wilson provides the most benefit.

What This Means for Developers

- Developers do not need to choose a method or configure thresholds for method switching
- Statistical calculations are consistent across all sample sizes
- The formulas shown above are what every mavai framework uses—always

Conformance Verification The mavai-R project generates reference Wilson score interval values using R's `qnorm` function. Each framework implementation verifies its Wilson computation matches the R-generated reference data. See `inst/cases/wilson_ci.json` and `inst/cases/wilson_lower.json` in the mavai-R repository.

2.3.2 Wald Interval (For Completeness) For completeness, this section documents the Wald interval (normal approximation), which statisticians will encounter in many textbooks. **The mavai methodology does not use this method**, but understanding it helps explain why Wilson is preferred.

For large n, by the Central Limit Theorem:

$$\hat{p} \overset{a}{\sim} N\left(p, \frac{p(1-p)}{n}\right)$$

The $(1-\alpha)$ Wald confidence interval is:

$$\hat{p} \pm z_{\alpha/2} \cdot \mathrm{SE}(\hat{p})$$

where $z_{\alpha/2}$ is the $(1-\alpha/2)$ quantile of the standard normal distribution.

Example (95% CI, $z_{0.025} = 1.96$):

$$0.951 \pm 1.96 \times 0.00683 = [0.938, 0.964]$$

Why the mavai Methodology Does Not Use Wald Statistics textbooks often present guidelines for when Wald is acceptable:

Condition	Textbook Guidance
$n \geq 40$ and $0.1 \leq \hat{p} \leq 0.9$	Wald acceptable
$n \geq 20$ and ($\hat{p} < 0.1$ or $\hat{p} > 0.9$)	Wilson preferred
$n < 20$	Wilson strongly recommended
$n < 10$	Wilson required; Wald inappropriate

Rather than implement conditional method selection, the methodology uses Wilson universally. This simplifies implementation while providing correct results in all cases—including the edge cases where Wald fails.

2.4 Sample Size Determination

> **Epistemic status**: planning approximation based on normal asymptotics. Adequate for sample-size budgeting; not of the same epistemic type as the Wilson constructions elsewhere in this document.

To achieve a desired margin of error e with confidence $(1-\alpha)$, the required sample size is approximately:

$$n = \frac{z_{\alpha/2}^2 \cdot \hat{p}(1-\hat{p})}{e^2}$$

Example: To estimate p ≈ 0.95 with ±2% margin at 95% confidence:

$$n = \frac{1.96^2 \times 0.95 \times 0.05}{0.02^2} = \frac{0.1825}{0.0004} \approx 456$$

Target Precision (95% CI)	Required n (for p ≈ 0.95)
±5%	73
±3%	203
±2%	456
±1%	1,825

---

3. Threshold Derivation for Regression Testing

3.1 The Problem

The experiment established $\hat{p}_{\text{baseline}} = 0.951$ from $n_{\text{baseline}} = 1000$ samples. For cost reasons, regression tests will use $n_{\text{test}} = 100$ samples.

Question: What threshold p^* should the regression test use?

Naive approach: Use $p^* = 0.951$.

Problem with naive approach: With only 100 samples, the standard error is:

$$\text{SE}_{\text{test}} = \sqrt{\frac{0.951 \times 0.049}{100}} \approx 0.0216$$

Even if the true p equals the experimental rate, observed rates will vary. At $\pm 2\sigma$, we'd expect observations between 0.908 and 0.994. Using 0.951 as the threshold would cause frequent false positives.

3.2 One-Sided Hypothesis Testing Framework

Compliance and regression testing both use a one-sided binomial decision skeleton, but as introduced in §0 they implement **two distinct hypothesis families with distinct error semantics**. The same arithmetic engine serves both, but the controlled error and the meaning of PASS/FAIL differ.

Compliance / assurance procedure — affirmative test that the system meets a normative threshold p_{req}:

$$H_0 : p \leq p_{\text{req}} \qquad H_1 : p > p_{\text{req}}$$

Decision: PASS only if the one-sided lower confidence bound on p exceeds p_{req}. The procedure controls the **false-compliance** rate — the long-run probability of declaring compliance when the true success probability is at or below the requirement, subject to discreteness and approximation.

Regression / monitoring procedure — reference-control test that detects degradation from a measured reference:

Choose an integer lower-tail cutoff c such that, under the stated reference model,

$$P_{\text{ref}}(K < c) \leq \alpha.$$

Decision: PASS if observed $K \geq c$; FAIL if $K < c$. The procedure controls the **false-degradation-alarm** rate under the reference model. A regression PASS means "no degradation signal at this cutoff"; it does not establish equivalence to the baseline.

For continuity with §§3.3–5 prose, the regression rule is also written in the equivalent rate form

$$H_0 : p \geq p^* \quad \text{(acceptable)}$$

$$H_1 : p < p^* \quad \text{(unacceptable)}$$

with $p^* = c/n$ as the displayed-rate companion of the binding integer cutoff c. Implementations report both c and c/n; the integer cutoff is the decision artefact (§3.4, §5.1).

The differences across the two paradigms are summarised below. Note that the controlled error and the verdict semantics differ.

Paradigm	Threshold	Hypotheses	Error controlled	PASS means
Compliance	p_{SLA} (given)	$H_0 : p \leq p_{\text{req}}$ vs $H_1 : p > p_{\text{req}}$	False-compliance probability	Evidence supports compliance at the configured level
Regression	Derived from $\widehat{p}_{\text{baseline}}$	$H_0 : p \geq p^*$ vs $H_1 : p < p^*$, integer cutoff c from $P_{\text{ref}}(K < c) \leq \alpha$	False-degradation-alarm rate under reference	No degradation signal at the configured cutoff

Regression-rule status note. The Wilson-derived empirical regression rule developed in §3.4 is a **one-sample reference-control approximation**. It uses the baseline point estimate $\widehat{p}_{\text{baseline}}$ together with the test sample size n_{test} and the configured α to set the cutoff; it does not fully propagate the baseline's own measurement uncertainty, especially when $n_{\text{baseline}} \neq n_{\text{test}}$. Two alternatives are available for projects that need fuller uncertainty propagation: the **beta-binomial posterior predictive** rule (§4.5), and frequentist **two-sample non-inferiority methods** such as Miettinen–Nurminen and Farrington–Manning. The Wilson rule is retained as the default for the operational reasons in §4.5; this note labels its status rather than replacing it.

We seek a decision rule that:

- Targets the controlled error rate at level α under the working model. (Note: because the binomial is discrete, the **achieved** size at the chosen integer cutoff c is generally less than the nominal α and must be reported alongside it — see §3.4.)
- Maximises power to detect true violations/degradation.

3.3 Normal-Approximation Intuition for One-Sided Bounds

The formula in this subsection is shown only to motivate the idea of a one-sided lower bound. The mavai methodology does not use this Wald form for threshold derivation; actual thresholds are computed using the Wilson construction in §3.4.

The $(1-\alpha)$ one-sided lower confidence bound is:

$$p_{\text{lower}} = \hat{p} - z_\alpha \cdot \text{SE}$$

Note: For one-sided bounds, we use z_α (not $z_{\alpha/2}$).

Confidence Level	z_α	Interpretation
90%	1.282	10% false positive rate
95%	1.645	5% false positive rate
99%	2.326	1% false positive rate

3.4 Threshold Calculation (Wilson Score Lower Bound)

Thresholds are derived using the Wilson one-sided lower bound — a **score-test inversion**, not an exact binomial procedure (§2.3.1) — consistent with the exclusive use of Wilson for all confidence-interval calculations across the methodology.

Given experimental results $(\hat{p}_{\text{baseline}}, n_{\text{baseline}})$ and test configuration $(n_{\text{test}}, \alpha)$, the threshold is the one-sided Wilson lower bound:

$$p^* = \frac{\hat{p} + \frac{z^2}{2n} - z\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}$$

where $z = z_\alpha$ is the one-sided critical value.

Example ($\hat{p}_{\text{baseline}} = 0.951$, $n_{\text{test}} = 100$, $\alpha = 0.05$, $z = 1.6448536269514722$):

Evaluating the Wilson one-sided lower bound gives the real-valued threshold

$$p^*_{\text{Wilson}} \approx 0.902124.$$

This real-valued lower bound is informational. Because the binomial is discrete, the binding decision artefact is integer-valued:

$$c = \lceil n_{\text{test}} \cdot p^*_{\text{Wilson}} \rceil = \lceil 100 \times 0.902124 \rceil = 91, \qquad \text{decision: PASS iff } K \geq c.$$

The displayed integer-rate companion is $c/n = 0.910000$. Three distinct artefacts therefore co-exist for this example and have distinct roles in the report:

- the real-valued Wilson lower bound $p^*_{\text{Wilson}} \approx 0.902124$ — the procedure's continuous output;
- the integer pass cutoff $c = 91$ — the binding decision rule;
- the displayed integer-rate companion $c/n = 0.910000$ — a human-readable summary of the decision.

The Wilson construction targets nominal $\alpha = 0.05$ before discretisation. For the integer cutoff $c = 91$, the achieved lower-tail false-degradation probability under $p_0 = 0.951$ is

$$P_{0.951}(K < 91) \approx 0.024986,$$

not 0.05. The gap between nominal α and achieved size is intrinsic to discrete decisions and not a defect; it is reported alongside c so that two thresholds whose real-valued bounds differ in the third decimal place can still be recognised as implementing the same decision when they share an integer cutoff.

Integer Cutoffs Are the First-Class Decision Artefact A conformant report for every inferential verdict carries:

- configured α;
- integer pass cutoff c;
- displayed cutoff c/n;
- observed count K and total n;
- **achieved size** under the stated reference model, where applicable;
- achieved power against any declared degradation margin, where available.

This requirement applies uniformly to compliance and regression procedures (§3.2), and the transparent-statistics output in §7.1 / §10.2 carries the same fields.

Numerical conventions for the cutoff. The integer cutoff uses the ceiling, $c = \lceil n \cdot p^* \rceil$, computed on the raw $n \cdot p^*$ without intermediate rounding. The displayed rate c/n is reported to a stated fixed precision — typically **six decimal places** — for cross-language reproducibility against the mavai-R fixtures; the unrounded c and n are retained on the trial record so a downstream consumer can recompute c/n at any precision. Two thresholds whose displayed rates agree to six decimal places implement the same decision iff their integer cutoffs c agree, which is the binding identity. The latency-side analogue of this convention is set out in §12.8.

Conformance Verification The mavai-R project generates reference threshold derivation values. See `inst/cases/threshold_derivation.json` in the mavai-R repository.

3.5 Reference Table: Wilson Score Lower Bounds

For baseline success-rate estimate $\hat{p} = 0.951$:

Test Samples	95% Lower Bound	99% Lower Bound
50	0.874	0.826
100	0.902	0.874
200	0.919	0.902
500	0.933	0.923

Observation: Smaller test samples require lower bounds (and hence lower thresholds) to maintain the same false positive rate.

3.6 Testing Against a Given Threshold (Compliance)

For compliance testing, the threshold is **given**, not derived:

$$p^* = p_{\text{SLA}}$$

There is no experimental baseline—the threshold comes directly from a contract, SLA, SLO, or organizational policy.

Example: A payment processing SLA states "99.5% transaction success rate."

- $p_{\text{SLA}} = 0.995$ (given by contract)
- No $\hat{p}_{\text{baseline}}$ to estimate
- Test directly verifies: does $p \geq 0.995$?

Why compliance testing differs from regression testing Compliance testing is an **affirmative-assurance** procedure. The required rate p_{req} is normative — supplied by an SLA, SLO, policy, or regulatory requirement — and the hypothesis pair is

$$H_0 : p \leq p_{\text{req}}, \qquad H_1 : p > p_{\text{req}}.$$

A PASS is issued only when the one-sided lower confidence bound on p exceeds p_{req}. The controlled error event is **false compliance**: declaring that the system meets the requirement when the true rate is at or below it.

Three regimes need to be kept distinct, because each behaves differently at the boundary $p = p_{\text{req}}$:

1. **Naive observed-rate thresholding.** Compare $\hat{p}$ directly with p_{req}. At $p = p_{\text{req}}$ this produces approximately symmetric pass/fail behaviour from sampling variance; about half of repeated tests yield $\hat{p} < p_{\text{req}}$. This is the rule whose failure at the boundary is properly described as a false positive. **It is not the methodology's compliance rule** and is presented here only for contrast.

2. **Confidence-bound compliance** (the methodology's rule). PASS only when the lower confidence bound on p clears p_{req}. The controlled error is false compliance, and it is controlled at the configured level. At $p = p_{\text{req}}$ the rule is conservative by construction: the lower bound rarely clears the requirement, so the procedure rarely declares compliance.
3. **Failure to demonstrate compliance.** A system whose true success probability is exactly p_{req} will often FAIL the confidence-bound rule, even when the true rate meets the requirement. Under regime 2 this is not a false positive — it is **failure to produce affirmative evidence** above the requirement, the expected behaviour of an affirmative-assurance procedure at modest sample sizes. Operators who require a higher chance of demonstrating compliance at the boundary increase the sample size or accept a smaller margin.

The three operational approaches in §6 — sample-size-first, confidence-first, and direct-threshold — are alternative ways to manage that conservatism within regime 2; they do not alter the controlled error event.

Reference Table: Sample Sizes for SLA Verification To verify $p \geq p_{\text{SLA}}$ with 95% confidence and 80% power:

p_{SLA}	Min Detectable Effect (δ)	Required Samples
0.95	0.05 (detect drop to 90%)	150
0.95	0.02 (detect drop to 93%)	822
0.99	0.02 (detect drop to 97%)	236
0.99	0.01 (detect drop to 98%)	793
0.999	0.005 (detect drop to 99.4%)	548
0.999	0.001 (detect drop to 99.8%)	8,031

Key insight: Higher SLAs and smaller detectable effects require dramatically more samples. This is why `minDetectableEffect` is essential for the confidence-first approach—without it, the question ”how many samples to verify 99.9%?” has no finite answer.

---

4. The Perfect Baseline Problem ($\hat{p} = 1$)

Empirical-baseline uncertainty — status note. The Wilson construction used in this section (and in §3.4 generally) is a **one-sample reference-control approximation**. It treats the baseline summary as fixed when deriving the test threshold and does not fully propagate baseline measurement uncertainty into the regression decision. This matters most when n_{baseline} and n_{test} differ materially: a baseline of $951/1000$ and a baseline of $95/100$ have the same point estimate but very different evidential weight, and the one-sample Wilson rule does not fully reflect that difference. Two alternatives are available for projects requiring fuller propagation of baseline uncertainty: the **beta-binomial posterior predictive** rule developed

in §4.5 (the "statistician mode" predictive alternative), and frequentist **two-sample non-inferiority methods** such as Miettinen–Nurminen (1985) and Farrington–Manning (1990). The Wilson rule is retained as the default for the operational reasons enumerated in §4.5; this note labels its epistemic status rather than replacing it.

4.1 Problem Statement

A critical pathology arises when the baseline experiment observes **zero failures**:

$$k = n \implies \hat{p} = 1$$

This commonly occurs when testing highly reliable systems (e.g., well-established third-party APIs) where failures are rare but not impossible.

Example: An experiment with $n = 1000$ trials against a payment gateway API yields $k = 1000$ successes.

Why this matters: With $\hat{p} = 1$, naive threshold derivation would set $p^* = 1$, meaning any single failure causes test failure—regardless of sample size or confidence level. This is statistically unsound.

The mavai solution: The Wilson score method (Section 2.3.1) handles this case correctly. This is another reason for using Wilson exclusively—it remains valid at the boundaries where other methods fail.

4.2 Interpretation of 100% Observed Success

An observed rate of $\hat{p} = 1$ from n trials does **not** mean $p = 1$; it yields a finite Wilson lower bound below 1. Under the stated i.i.d. Bernoulli model the Wilson lower-bound procedure has, in repeated use, approximately the configured long-run coverage on its one-sided interval, subject to the discreteness and approximation caveats stated elsewhere in this companion. The statement is a property of the procedure across repeated samples, not a posterior probability statement about p given the observed counts.

The Rule of Three (heuristic approximation): with n trials and zero failures, a useful rule of thumb places an approximate 95%-coverage one-sided lower bound at:

$$p \geq 1 - \frac{3}{n}.$$

The Rule of Three is an approximation to the Wilson and Clopper–Pearson constructions in the zero-failure case, not an exact result and not a posterior probability statement.

Baseline Samples	95% Lower Bound (Rule of Three)
100	0.970

Baseline Samples	95% Lower Bound (Rule of Three)
300	0.990
1000	0.997
3000	0.999

(Side note: this assumes conditions are stable and runs are independent.)

4.3 The Wilson Lower Bound Solution

The mavai methodology resolves this pathology using the **Wilson score lower bound**, which remains well-defined when $\hat{p} = 1$.

Key implementation requirement: Baselines store (k, n), not merely $\hat{p}$. The observed rate can be computed ($\hat{p} = k/n$), but raw counts are essential for proper statistical treatment of boundary cases.

Procedure:

1. Compute the one-sided Wilson lower bound p_0 from the baseline (k, n)
2. Use p_0 (not $\hat{p}$) as the effective baseline for threshold derivation
3. Apply the standard threshold formula using p_0

4.3.1 Wilson Lower Bound Formula The general Wilson one-sided lower bound is:

$$p_{\text{lower}} = \frac{\hat{p} + \frac{z^2}{2n} - z\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}$$

When $\hat{p} = 1$, this simplifies to:

$$p_{\text{lower}} = \frac{n}{n + z^2}$$

4.3.2 Worked Example **Baseline**: $n_{\text{baseline}} = 1000$ trials, $k_{\text{baseline}} = 1000$ successes, so $\hat{p}_{\text{baseline}} = 1.0$.

Confidence: 95% one-sided, so $z = 1.645$.

Step 1: Compute the Wilson lower bound for the perfect baseline

Because the experiment observed zero failures, the point estimate $\hat{p}_{\text{baseline}} = 1.0$ must not be used directly as the effective baseline. A perfect empirical observation does not prove perfect population reliability.

For $k = n$, the one-sided Wilson lower bound simplifies to:

$$p_0 = \frac{n}{n + z^2}$$

Substituting $n = 1000$ and $z = 1.645$:

$$p_0 = \frac{1000}{1000 + 1.645^2} = \frac{1000}{1002.706} \approx 0.9973$$

This value, $p_0 \approx 0.9973$, is the effective baseline used for subsequent threshold derivation. It represents the lower confidence-supported success probability implied by observing 1000 successes in 1000 trials.

Step 2: Derive the test threshold using the Wilson lower bound

For a regression test with $n_{\text{test}} = 100$, the threshold is not computed with the Wald approximation $p_0 - z \cdot \text{SE}$. The mavai methodology applies the same one-sided Wilson lower-bound construction used throughout this document:

$$p^* = \frac{p_0 + \frac{z^2}{2n_{\text{test}}} - z\sqrt{\frac{p_0(1-p_0)}{n_{\text{test}}} + \frac{z^2}{4n_{\text{test}}^2}}}{1 + \frac{z^2}{n_{\text{test}}}}$$

Substituting $p_0 = 0.9973$, $n_{\text{test}} = 100$, and $z = 1.645$:

$$p^* \approx 0.9686$$

Interpretation: For a 100-sample regression test, the Wilson-derived threshold is approximately **0.969**. Therefore, the test passes if the observed success rate is at least 96.9%.

Because test outcomes are discrete, this corresponds to requiring at least:

$$\lceil 100 \times 0.9686 \rceil = 97$$

successes out of 100.

This is intentionally less strict than requiring 100 successes out of 100, or even 99 out of 100. The experiment's perfect observation establishes strong evidence of high reliability, but the subsequent test still has sampling variability. The Wilson construction accounts for that uncertainty without treating the original perfect baseline as proof of $p = 1$.

4.3.3 Reference Table: Thresholds for 100% Baselines Each test threshold below is the one-sided Wilson lower bound for p_0 at the test sample size, with $z = 1.645$ (95% confidence) — the same construction as §4.3.2 Step 2.

Baseline n	p_0 (Wilson 95%)	Test $n = 50$ threshold	Test $n = 100$ threshold
100	0.9737	0.906	0.932
300	0.9911	0.933	0.958
1000	0.9973	0.944	0.969
3000	0.9991	0.947	0.972

4.4 Extended Example: Highly Reliable API

Scenario: Testing a payment gateway integration.

Baseline experiment: 2000 transactions, 0 failures ($\hat{p} = 1$).

Goal: Configure regression tests with 95% confidence.

Calculation:

1. Effective baseline (Wilson lower bound at $k = n$, $z = 1.645$):

$$p_0 = \frac{2000}{2000 + 1.645^2} = \frac{2000}{2002.706} \approx 0.9986$$

2. For a 100-sample test, apply the one-sided Wilson lower bound at p_0 and $n_{\text{test}} = 100$:

$$p^* = \frac{p_0 + \frac{z^2}{2n_{\text{test}}} - z\sqrt{\frac{p_0(1-p_0)}{n_{\text{test}}} + \frac{z^2}{4n_{\text{test}}^2}}}{1 + \frac{z^2}{n_{\text{test}}}} \approx 0.971$$

   Discrete equivalent: $\lceil 100 \times 0.971 \rceil = 98$ successes out of 100.

3. For a 50-sample test, the same construction with $n_{\text{test}} = 50$ gives:

$$p^* \approx 0.946$$

   Discrete equivalent: $\lceil 50 \times 0.946 \rceil = 48$ successes out of 50.

Result: Even for this highly reliable system, the methodology produces statistically principled thresholds with valid confidence-level interpretation. The larger test (n=100) yields a tighter threshold than the smaller (n=50), reflecting the smaller test's greater sampling variability.

4.5 Theoretical Note: Beta-Binomial Alternative

For statisticians reviewing this methodology: the **Beta-Binomial posterior predictive** approach is the theoretically cleaner treatment. It fully propagates baseline

uncertainty into a predictive distribution for future test counts, naturally yields integer thresholds, and avoids the confidence-vs-prediction gap that the Wilson construction exhibits when baseline and test sample sizes differ materially (§3.3 caveat, and §12.4.3 for the latency analogue). The mavai methodology nevertheless uses the Wilson bound as its default construction because:

- **No prior negotiation.** Selecting (a, b) — or defending Jeffreys' $a = b = 0.5$ — is not a conversation most engineering teams can have in the time a regression test is configured. Wilson requires nothing beyond counts and a confidence level.
- **Auditability in regulated settings.** A frequentist lower confidence bound has a single, externally verifiable interpretation. A posterior predictive bound adds a prior that an auditor must understand, accept, or challenge; the extra degree of freedom is epistemically honest but operationally expensive in regulated or contractual contexts.
- **Cross-language stability.** Wilson reduces to elementary arithmetic (qnorm + closed form) and reproduces bit-exactly across languages. Beta-Binomial CDFs rely on the regularised incomplete beta function, whose numerical behaviour at extreme (a, b) differs between libraries — a real conformance problem across punit, feotest, and baseltest.
- **Close enough in the operating range.** For the sample sizes typical of measure experiments ($n \geq 100$) and the pass rates typical of production LLM services ($\hat{p} \in [0.9, 1.0]$), Wilson lower bounds and Beta-Binomial predictive lower bounds agree to within a fraction of a percentage point. The larger risk in practice is a stale baseline or a mis-specified covariate, not the choice of interval method.

The trade-off is deliberate, not evasive. Organisations with strong Bayesian infrastructure, or service contracts with extreme sample-size imbalance between baseline and test, may wish to substitute the posterior predictive:

$$K_t \mid k, n \sim \text{BetaBinomial}(n_t, a + k, b + n - k)$$

where (a, b) are prior hyperparameters (Jeffreys: $a = b = 0.5$). See Gelman et al. (2013) and Bayarri & Berger (2004) for treatments. The mavai framework does not preclude this substitution at the implementation layer; what it standardises is the Wilson default and the conformance reference data that flows from it.

---

5. Test Execution and Interpretation

Procedure scope of the sections below. §§5.1–5.4 describe the **regression** procedure: the decision rule against the integer cutoff c, the Type-I/Type-II error frame in degradation terms, and the sample-size derivation for detecting an effect δ at given power. The compliance procedure's verdict semantics, decision rule (Wilson lower-bound clearance of p_{req}), and error events (”false compliance”) are stated in §3.2; §5.5 covers the companion compliance-side sample-size derivation against an SLA threshold. §5.6 (minimum detectable effect) and §5.7 (VERIFICATION/SMOKE intent and the feasibility gate) apply to both procedures.

5.1 Decision Rule

Given a test with n_{test} samples and threshold p^* (with corresponding integer cutoff $c = \lceil n_{\text{test}} \cdot p^* \rceil$, per §3.4):

1. Execute service contract n_{test} times
2. Count successes k_{test}
3. Decision (binding, integer-valued form):
   - If $k_{\text{test}} \geq c$: **PASS** (no degradation signal at this cutoff)
   - If $k_{\text{test}} < c$: **FAIL** (threshold not met — degradation signal at α under the reference)

The displayed observed rate $\hat{p}_{\text{test}} = k_{\text{test}}/n_{\text{test}}$ is reported alongside the integer count for human readability, but the binding decision is on k_{test} vs. c, not on $\hat{p}_{\text{test}}$ vs. p^*. The two are equivalent for the regression rule (the integer cutoff is derived from p^*); reporting both prevents auditors and developers from being misled when the displayed-rate boundary lies between two adjacent integer outcomes.

The verdict wording follows the procedure type (§3.2): a **regression** PASS reads "no degradation signal at the configured cutoff," not "equivalence to baseline established"; a **compliance** PASS reads "evidence supports compliance at the configured level," not "no degradation."

5.2 Type I and Type II Errors

	True state: No degradation	True state: Degradation
Test passes	Correct (True Negative)	Type II Error (False Negative)
Test fails	Type I Error (False Positive)	Correct (True Positive)

- **Type I error rate** (α): Targeted by threshold derivation. If the threshold is set at $(1-\alpha)$ confidence on the baseline and the true success probability equals the experimental rate, then $P(\text{False Positive}) \approx \alpha$ under the working model. The word approximately is load-bearing: the construction adjusts only the threshold side, not the joint baseline-and-test predictive distribution, and calibration degrades when test-sample size is small relative to baseline (see §3.3 and §12.4.3 for the analogous caveat on the latency side).
- **Type II error rate** (β): Depends on:
  - True effect size (how much degradation occurred)
  - Sample size
  - Threshold

5.3 Statistical Power

> **Epistemic status**: planning approximation based on normal asymptotics. The power formulas below are adequate for deciding "is this test worth running?" but should not be read as exact calibration claims under small or imbalanced sample sizes.

Power is the probability of correctly detecting degradation when it exists:

$$\text{Power} = 1 - \beta = P(\text{Reject } H_0 | H_1 \text{ true})$$

For a one-sided test detecting a shift from p_0 to p_1 (where $p_1 < p_0$):

$$\text{Power} = \Phi\left(\frac{p_0 - p_1 - z_\alpha \cdot \text{SE}_0}{\text{SE}_1}\right)$$

where:

- $\text{SE}_0 = \sqrt{p_0(1-p_0)/n}$
- $\text{SE}_1 = \sqrt{p_1(1-p_1)/n}$
- Φ is the standard normal CDF

Example: Detecting a drop from $p_0 = 0.95$ to $p_1 = 0.90$ with $n = 100$ at $\alpha = 0.05$:

$$\text{SE}_0 = \sqrt{0.95 \times 0.05/100} = 0.0218$$
$$\text{SE}_1 = \sqrt{0.90 \times 0.10/100} = 0.0300$$
$$\text{Power} = \Phi\left(\frac{0.95 - 0.90 - 1.645 \times 0.0218}{0.0300}\right) = \Phi\left(\frac{0.0141}{0.0300}\right) = \Phi(0.47) \approx 0.68$$

With 100 samples, the test has only 68% power to detect a 5-percentage-point degradation.

Conformance Verification The mavai-R project generates reference power analysis values. See `inst/cases/power_analysis.json` in the mavai-R repository.

5.4 Sample Size for Desired Power

To achieve power $(1 - \beta)$ for detecting effect size $\delta = p_0 - p_1$:

$$n = \left(\frac{z_\alpha \sqrt{p_0(1-p_0)} + z_\beta \sqrt{p_1(1-p_1)}}{\delta}\right)^2$$

Example: 80% power to detect 5% drop from 95% to 90% at $\alpha = 0.05$:

$$n = \left(\frac{1.645 \times 0.218 + 0.842 \times 0.300}{0.05}\right)^2 = \left(\frac{0.359 + 0.253}{0.05}\right)^2 = (12.24)^2 \approx 150$$

Effect Size	Power 80%	Power 90%	Power 95%
5% (95%→90%)	150	200	250
10% (95%→85%)	40	55	70
3% (95%→92%)	410	550	700

5.5 Sample Size for SLA Verification

When verifying an SLA threshold p_{SLA} (rather than detecting degradation from a baseline), the sample size formula adapts:

$$n = \left(\frac{z_\alpha \sqrt{p_{\text{SLA}}(1 - p_{\text{SLA}})} + z_\beta \sqrt{(p_{\text{SLA}} - \delta)(1 - p_{\text{SLA}} + \delta)}}{\delta} \right)^2$$

Where:

- p_{SLA} is the given SLA threshold (e.g., 0.995)
- δ is the minimum detectable effect—the smallest violation worth detecting
- α is the significance level (Type I error rate)
- β is the Type II error rate ($1 - \beta$ is power)

Example: Verify 99.5% SLA with 95% confidence, 80% power, detecting drops of 1% or more:

- $p_{\text{SLA}} = 0.995$, $\delta = 0.01$, $\alpha = 0.05$, $\beta = 0.20$

$$n = \left(\frac{1.645 \times \sqrt{0.995 \times 0.005} + 0.842 \times \sqrt{0.985 \times 0.015}}{0.01} \right)^2$$

$$= \left(\frac{1.645 \times 0.0705 + 0.842 \times 0.1215}{0.01} \right)^2 = \left(\frac{0.116 + 0.102}{0.01} \right)^2 = (21.8)^2 \approx 477$$

Reference Table: Sample Sizes for High SLAs

SLA Threshold	Effect Size (δ)	95% Confidence, 80% Power
99.0%	2% (detect ≤97%)	236
99.0%	1% (detect ≤98%)	793
99.5%	1% (detect ≤98.5%)	477
99.5%	0.5% (detect ≤99%)	1,597
99.9%	0.5% (detect ≤99.4%)	548
99.9%	0.1% (detect ≤99.8%)	8,031

Key insight: Verifying high SLAs with small detectable effects requires substantial sample sizes. A 99.9% SLA with 0.1% detection requires about 8,031 samples.

5.6 The Role of Minimum Detectable Effect

The `minDetectableEffect` parameter answers a critical question:

> "What's the smallest degradation worth detecting?"

Without this parameter, the question "how many samples to verify $p \geq 0.999$?" has no finite answer. Here's why:

Mathematical necessity: To detect an arbitrarily small degradation from 99.9% to 99.89% would require millions of samples. To detect a drop to 99.899% requires even more. For infinitesimal effects, infinite samples are required.

Practical reality: No organization needs to detect every possible degradation. There's always a threshold below which degradation doesn't matter operationally:

System Type	Typical δ	Rationale
E-commerce checkout	1-2%	1% drop = significant revenue loss
Internal tooling	5-10%	User productivity impact
Safety-critical systems	0.1-0.5%	Regulatory requirements
High-frequency trading	0.01%	Financial impact per transaction

When `minDetectableEffect` is required:

In the **Confidence-First approach**, developers must specify `minDetectableEffect` for the framework to compute the required sample size. This applies to both compliance and regression testing.

Without `minDetectableEffect`, no framework can compute a finite sample size and will use the default sample count instead.

5.7 Test Intent: VERIFICATION vs SMOKE

The mavai methodology distinguishes between two epistemic intentions when running a probabilistic test. The choice of intent governs whether the framework enforces a statistical feasibility gate and how verdicts are framed.

Intent	Purpose	Feasibility gate	Verdict language
VERIFICATION	Evidential — produce statistically defensible verdict	Enforced	Full compliance framing
SMOKE	Sentinel — detect gross regressions cheaply	Bypassed	Softened, exploratory framing

The default intent is **VERIFICATION**. Developers may opt into SMOKE when appropriate.

5.7.1 The Feasibility Gate (VERIFICATION only) Before any samples execute, the framework checks whether the configured sample size is **sufficient** for the test to produce a meaningful PASS verdict. The criterion uses the Wilson score one-sided lower bound (the same method used throughout for confidence bounds — see Section 4.3.1):

> For a perfect observation ($k = n$), the Wilson lower bound is $n/(n + z^2)$. The sample is feasible if this bound $\geq p_0$.

Solving for the minimum sample size:

$$N_{\text{min}} = \left\lceil \frac{p_0 \cdot z^2}{1 - p_0} \right\rceil$$

where $z = \Phi^{-1}(1 - \alpha)$ and $\alpha = 1 -$ confidence.

Reference table: N_{min} at default confidence (0.95, $\alpha = 0.05$, $z \approx 1.645$)

Target (p_0)	N_{min}	Interpretation
0.50	3	Almost any sample size suffices
0.80	11	Low bar
0.90	25	Moderate
0.95	52	Common threshold — needs at least 52 samples
0.99	268	High reliability — needs substantial samples
0.999	2,704	Very high reliability
0.9999	27,058	Extreme reliability — impractical for most test suites

What happens when infeasible: A VERIFICATION test with $N < N_{\text{min}}$ fails immediately with a configuration error. The failure message includes:

- The configured sample size and target
- The minimum required sample size
- A suggestion to use SMOKE intent if the test is a sentinel check

This failure is **distinct from a SUT failure** — it indicates a configuration problem, not a system defect. It is non-ignorable in CI.

Conformance Verification The mavai-R project generates reference feasibility values. See `inst/cases/feasibility.json` in the mavai-R repository.

5.7.2 SMOKE Intent: When and Why SMOKE tests are appropriate when:

- **Quick feedback** is more valuable than statistical defensibility (e.g. pre-commit hooks, nightly canaries)
- **The sample budget is fixed** by cost constraints and falls below N_{min}

- **The test is exploratory** — developers are still discovering the system's performance characteristics
- **The target is aspirational** — the threshold expresses an SLA that will later be verified with a properly sized test

5.7.3 FAIL Asymmetry for SMOKE Tests A key statistical insight governs how SMOKE verdicts should be interpreted:

> A **FAIL** verdict from an undersized test is directionally reliable — the observed rate fell below the threshold, and the direction of the evidence is clear even if the magnitude is uncertain. A **PASS** verdict from an undersized test provides weak evidence — the confidence interval is too wide to exclude values below the threshold.

In other words: small samples can reliably detect gross failures but cannot provide strong evidence of compliance. Framework output reflects this asymmetry:

- **SMOKE PASS**: Softened language — "The observed rate is consistent with the target."
- **SMOKE FAIL**: Still clear — "The observed rate is inconsistent with the target."
- **VERIFICATION PASS**: Full compliance language — "The system meets its SLA requirement."

5.7.4 Intent-Aware Caveats When a SMOKE test runs against a normative threshold (SLA, SLO, or POLICY):

Condition	Caveat
$N < N_{\text{min}}$	"Sample not sized for verification ($N = x$, need y). A PASS is a directional signal, not a compliance determination."
$N \geq N_{\text{min}}$	"Sample is sized for verification. Consider setting `intent = VERIFICATION` for evidential strength."

These caveats appear in both summary and verbose output modes.

---

6. The Three Operational Approaches: Mathematical Formulation

> **Epistemic status**: decision-policy regimes, not distinct inferential theories. The three "approaches" below describe which test-configuration parameters the framework fixes and which it derives, not three different statistical methods. They all sit inside the same Bernoulli/binomial model, and the sample-size and confidence formulas used within them are the **asymptotic / normal-approximation** planning formulas from §2.4 and §5.3, not Wilson constructions.

The three operational approaches apply to **both** paradigms. The key difference is the source of the threshold:

Paradigm	Threshold Source	Symbol Used
Compliance	Given by contract/policy	p_{SLA}
Regression	Derived from experimental basis	$\hat{p}_{\text{baseline}}$

Below, we present each approach with formulations for both paradigms.

6.1 Approach 1: Sample-Size-First (Cost-Driven)

Fix the sample count based on budget constraints; compute the implied threshold or confidence.

Regression Formulation **Given**: n_{test}, α, experimental basis $(\hat{p}_{\text{baseline}}, n_{\text{baseline}})$

Compute: p^*

$$p^* = \hat{p}_{\text{baseline}} - z_\alpha \sqrt{\frac{\hat{p}_{\text{baseline}}(1 - \hat{p}_{\text{baseline}})}{n_{\text{test}}}}$$

Trade-off: Fixed cost; confidence is controlled; threshold (sensitivity) is determined.

Compliance Formulation **Given**: n_{test}, p_{SLA}

Compute: Implied confidence level

Since the threshold is fixed ($p^* = p_{\text{SLA}}$), we compute what confidence the test achieves:

$$z = \frac{\hat{p}_{\text{test}} - p_{\text{SLA}}}{\sqrt{p_{\text{SLA}}(1 - p_{\text{SLA}})/n_{\text{test}}}}$$

The achieved confidence is $1 - \Phi(-z)$ where Φ is the standard normal CDF.

Trade-off: Fixed cost and threshold; confidence is determined by the data.

6.2 Approach 2: Confidence-First (Risk-Driven)

Fix the confidence and power requirements; compute the required sample size.

Regression Formulation **Given**: α, desired power $(1-\beta)$, minimum detectable effect δ, experimental basis $(\hat{p}_{\text{baseline}}, n_{\text{baseline}})$

Compute: n_{test}

$$n_{\text{test}} = \left(\frac{z_\alpha \sqrt{\hat{p}_{\text{baseline}}(1-\hat{p}_{\text{baseline}})} + z_\beta \sqrt{(\hat{p}_{\text{baseline}} - \delta)(1 - \hat{p}_{\text{baseline}} + \delta)}}{\delta} \right)^2$$

Trade-off: Fixed confidence and detection capability; cost (sample size) is determined.

Compliance Formulation **Given**: α, desired power $(1-\beta)$, minimum detectable effect δ, SLA threshold p_{SLA}

Compute: n_{test}

$$n_{\text{test}} = \left(\frac{z_\alpha \sqrt{p_{\text{SLA}}(1-p_{\text{SLA}})} + z_\beta \sqrt{(p_{\text{SLA}} - \delta)(1 - p_{\text{SLA}} + \delta)}}{\delta} \right)^2$$

Example: Verify 99.5% SLA, detecting drops of 1% or more with 95% confidence and 80% power:

- $p_{\text{SLA}} = 0.995$, $\delta = 0.01$, $z_\alpha = 1.645$, $z_\beta = 0.842$

$$n = \left(\frac{1.645 \times 0.0705 + 0.842 \times 0.1215}{0.01} \right)^2 \approx 477$$

Trade-off: Fixed confidence and detection capability; cost (sample size) is determined.

Critical: The `minDetectableEffect` (δ) is essential. Without it, verifying any SLA requires infinite samples.

6.3 Approach 3: Direct Threshold (Threshold-First)

Use an explicit threshold directly; compute the implied confidence.

Regression Formulation **Given**: n_{test}, p^* (often = $\hat{p}_{\text{baseline}}$), experimental basis

Compute: Implied α

$$z_\alpha = \frac{\hat{p}_{\text{baseline}} - p^*}{\sqrt{\hat{p}_{\text{baseline}}(1-\hat{p}_{\text{baseline}})/n_{\text{test}}}}$$

$$\alpha = 1 - \Phi(z_\alpha)$$

Example: Using threshold = 0.951 with $n = 100$:

$$z_\alpha = \frac{0.951 - 0.951}{0.0216} = 0$$

$$\alpha = 1 - \Phi(0) = 0.50$$

Interpretation: A 50% false positive rate—half of all test runs will fail even with no degradation.

Compliance Formulation **Given**: n_{test}, p_{SLA}

Compute: Implied α (using observed data)

The test directly uses the SLA threshold. After running, we compute:

$$z = \frac{\hat{p}_{\text{test}} - p_{\text{SLA}}}{\sqrt{p_{\text{SLA}}(1 - p_{\text{SLA}})/n_{\text{test}}}}$$

The implied Type I error rate depends on where the true system rate lies relative to p_{SLA}:

- If true $p = p_{\text{SLA}}$: approximately 50% of tests will fail (the threshold boundary problem)
- If true $p > p_{\text{SLA}}$: fewer false positives
- If true $p < p_{\text{SLA}}$: the test correctly detects the violation

Trade-off: Fixed cost and threshold; confidence (reliability of verdicts) is determined—often poorly when the true rate is near the threshold.

When to use: Learning the trade-offs, strict compliance requirements where the SLA threshold is non-negotiable.

---

7. Reporting and Interpretation

7.1 Transparent Statistics Output

When transparent statistics mode is enabled, the framework outputs a structured report containing the following sections:

Section	Contents	Purpose
HYPOTHESIS TEST	H_0, H_1, test type	Frames the statistical question being answered
OBSERVED DATA	Sample size n, successes k, observed rate $\hat{p}$	Raw observations from test execution
BASELINE REFERENCE	Source, empirical basis or SLA threshold, derivation method	Traces threshold to its origin

Section	Contents	Purpose
STATISTICAL INFERENCE	Standard error, confidence interval, z-score, p-value	Full calculation transparency
VERDICT	Result (PASS/FAIL), plain English interpretation, caveats	Human-readable conclusion
THRESHOLD PROVENANCE	Threshold origin, contract reference (if specified)	Auditability for compliance tests

Key Metrics in the Report

Metric	Formula/Value	Interpretation
Configured α	α	Nominal error level requested by the developer
Sample size	n	Number of trials executed
Successes	K (or k)	Number of passing trials
Observed rate	$\hat{p} = K/n$	Point estimate from test
Integer pass cutoff	$c = \lceil n \cdot p^* \rceil$	Binding decision artefact (§3.4, §5.1)
Displayed cutoff	c/n	Cutoff expressed as a rate
Achieved size	$P_{\text{ref}}(K < c)$ under the stated reference	Actual error rate of the discrete decision (≠ nominal α)
Standard error	$\text{SE} = \sqrt{\hat{p}(1-\hat{p})/n}$	Precision of the estimate
Confidence interval	Wilson score bounds	Range of plausible true values
Z-score	$z = (\hat{p} - p^*)/\text{SE}_0$	Standardised deviation from threshold (diagnostic)
p-value	tail probability — see paragraph below	Tail probability under the stated null and orientation

P-value alignment. A p-value is meaningful only against a stated null hypothesis and tail; under the methodology, any p-value that appears in a report is the tail probability of the same statistic and same orientation as the decision rule that produced the verdict. The methodology recognises three internally consistent reporting strategies:

1. **Exact binomial p-values** matching the stated hypothesis orientation — e.g. $P(K \leq k_{\text{obs}} \mid p = p_0)$ for the regression null $H_0 : p \geq p^*$.
2. **Score-test p-values** matching the Wilson construction used to derive the threshold.
3. **No p-value**, in which case the report carries the confidence bounds, the integer cutoff c, the achieved size, and the raw counts as the decision summary.

Where a p-value is present, the null hypothesis, the alternative, and the tail are part of the same report entry, so that the p-value cannot be read against a different orientation than the one that produced it. A Wald-style z-score paired with a Wilson lower-bound decision is at most a diagnostic approximation: it does not by itself constitute

a calibrated p-value for the procedure that produced the verdict, and the methodology classifies such z-scores as diagnostic alongside, rather than authoritative for, the verdict. The example output below retains a Wald-style z-score and one-tail p-value for illustration; the authoritative decision summary is the integer cutoff c together with the achieved size.

Example Output A transparent-statistics report for a regression criterion observing 87/100 successes against the §3.4 worked example (real-valued Wilson lower bound $p^*_{\text{Wilson}} \approx 0.902124$, integer cutoff $c = 91$) carries the procedure metadata alongside the observed data and the inference, so that the verdict cannot be read against a different orientation than the one that produced it:

```
PROCEDURE
  procedure:           REGRESSION
 nullHypothesis:    "K follows the stated reference model; no degradation signal"
  alternative:         "Lower-tail degradation signal under the reference"
  decisionRule:        "PASS iff K >= c"
  integerCutoff (c):   91
  realValuedBound:     p*_Wilson ≈ 0.902124
  displayedCutoff:     c/n = 0.910000
  configuredAlpha:     0.05
  achievedSize:        P_0.951(K < 91) ≈ 0.024986

OBSERVED DATA
  Sample size (n):     100
  Successes (K):       87
  Observed rate (p̂):   0.870

STATISTICAL INFERENCE
  Wilson 95% one-sided lower bound on p: 0.790
  Diagnostic z-score (Wald form, illustrative only — not the operative test):
                       z = (p̂ - p*_Wilson) / √(p*_Wilson(1 - p*_Wilson)/n)
                       z ≈ -1.10
  achieved size at integer cutoff c = 91:
    method:            exact-binomial-lower-tail
    tail:              P_{p = 0.951}(K < 91)
    value:             ≈ 0.026
    interpretation:    false-degradation-alarm probability under the
                       reference at the configured cutoff.

VERDICT
  Result:              FAIL
  Interpretation:      The observed count K = 87 is below the integer cutoff
                       c = 91 derived from the reference model at nominal
                      alpha = 0.05. The decision is made on K versus c, not
                       on rounded displayed rates.
```

The Wald-style z-score line is retained for orientation and is labelled diagnostic; the

binding decision summary is the procedure block together with the observed K and n.

See Section 10 for complete example outputs including both compliance and regression paradigms.

Conformance Verification The mavai-R project generates reference verdict evaluation values. See `inst/cases/verdict.json` in the mavai-R repository.

7.2 Confidence Statement

A verdict is accompanied by a plain-language confidence statement whose form is determined by the procedure that produced it. Because the two procedures of §3.2 control different error events, the methodology defines two distinct statements, each a frequentist long-run-property statement about the procedure rather than a posterior probability about the individual verdict.

For the **regression / monitoring** procedure the statement takes the form:

> "This rule was configured at nominal $\alpha = 0.05$. Under the stated reference model, and assuming independent and stationary trials, the long-run probability of the corresponding false degradation signal is targeted at 5%, subject to discreteness and approximation. This is not a posterior probability that this individual result is wrong."

For the **compliance / assurance** procedure the statement takes the form:

> "If the true success probability were at or below the requirement, this confidence-bound rule would falsely declare compliance with probability controlled at the configured level (here 5%), subject to model assumptions, discreteness, and approximation. This is not a posterior probability that this individual result is correct."

The distinction matters because the two are not equivalent. The frequentist α level is a property of the procedure under repeated use against a stated reference model: it bounds the long-run rate at which the procedure produces the controlled error event. It is not the probability, given a particular observed verdict, that that verdict is wrong — a posterior quantity that depends on the prior probability of degradation, the effect-size distribution, and operational context the procedure does not see. Paraphrases such as "there is a 5% probability that this failure is due to sampling variance rather than actual system degradation" conflate the two and are inconsistent with the frequentist construction used in §3.

7.3 Multiple Testing Considerations

When running multiple probabilistic tests, the per-test α is not the family-wise false-positive probability of the suite. Two distinct families need to be addressed separately, and the methodology takes the dependence-robust bound as its primary statement.

Dependence-robust statement (Bonferroni union bound). For any family of m tests with per-test sizes $\alpha_1, \ldots, \alpha_m$, regardless of the dependence structure among

the test statistics,

$$P(\text{at least one false positive}) \;\leq\; \sum_{i=1}^{m} \alpha_i.$$

This is the methodology's default family-wise statement. It coincides with the per-criterion Type-I envelope of §1.4.6 specialised to a multi-criterion contract, and it generalises to the cross-suite case below. It is conservative under positive dependence and tight under disjoint rejection regions.

Independent-tests special case. Under mutual independence of the test statistics (rarely true in practice — tests over the same service contract share a baseline, tests in a suite share a serving stack, tests in a CI run are temporally clustered), the bound collapses to

$$P(\text{at least one false positive}) \;=\; 1 - \prod_{i=1}^{m} (1 - \alpha_i).$$

For equal $\alpha_i = \alpha$ this is the familiar $1 - (1 - \alpha)^m$. The methodology cites this only to anchor intuition; it does not adopt it as the operational bound, because the independence premise is not generally defensible across the families the framework reports against.

Number of tests	Per-test α = 0.05 (independent)	Per-test α = 0.01 (independent)	Bonferroni envelope α = 0.05	Bonferroni envelope α = 0.01
5	22.6%	4.9%	25.0%	5.0%
10	40.1%	9.6%	50.0%	10.0%
20	64.2%	18.2%	100% (capped)	20.0%

The Bonferroni envelope is the conservative dependence-robust statement; the independent-tests column is the (typically inapplicable) lower-bound calibration.

Three families, kept separate. The methodology distinguishes three error-control levels and reports the corresponding α aggregates separately:

- **Per-criterion** α_c — the level configured on a single criterion's inferential test (§1.4.5).
- **Within-contract envelopes**, reported separately by procedure direction (§1.4.6): the **false-compliance envelope** $\sum_{c \in \text{compliance}} \alpha_c$ for compliance criteria and the **false-degradation-signal envelope** $\sum_{c \in \text{regression}} \alpha_c$ for regression criteria. A contract whose criteria are all of one kind reports the corresponding single envelope; a contract that mixes both reports both, labelled by direction.
- **Across-suite envelope** $\sum_s \alpha_s$ — the family-wise bound across multiple contracts run in the same suite (a CI build, a release gate). The methodology reports

this aggregate when more than one contract's verdict is bundled into a single suite-level conclusion.

Optional FDR-style controls. When the family is exploratory rather than gating — for example, a wide diagnostic sweep across criteria where the operator wants to control the expected proportion of false discoveries rather than the probability of any false discovery — false discovery rate (FDR) procedures may be applied. Two are relevant:

- **Benjamini–Hochberg (BH)**, which controls FDR at the configured level under independence and under the broader class of positive regression dependence on subsets (PRDS). (Benjamini & Hochberg, 1995.)
- **Benjamini–Yekutieli (BY)**, which controls FDR under arbitrary dependence at the cost of a $\sum_{i=1}^{m} 1/i$ penalty. (Benjamini & Yekutieli, 2001.)

The methodology does not impose FDR control; where it is in use, the choice between BH and BY is part of the report alongside the dependence assumption on which the chosen procedure relies, because the validity of the FDR claim depends on that assumption.

Unadjusted per-test α is a per-test claim. A per-test α reported without any family-wise or FDR adjustment bounds only the per-test Type-I rate; it is not a suite-level claim. Under the methodology such values carry the label **unadjusted per-test** in the report, so that they cannot be read as bounds on the family-wise or false-discovery rate of the suite.

Mitigation options (with their exact target):

Procedure	Controls	Dependence assumption
Per-criterion (no adjustment)	Per-test Type-I rate	None
Bonferroni $\alpha' = \alpha/m$	Family-wise error rate (FWER)	Arbitrary
Benjamini–Hochberg	False discovery rate (FDR)	Independence or PRDS
Benjamini–Yekutieli	False discovery rate (FDR)	Arbitrary
Disclosed $\sum \alpha$ (no rescaling)	FWER as a bound, no per-test rescaling	Arbitrary (this is the §1.4.6 default)

7.4 Threshold Provenance

For auditability and traceability, the methodology records the **source** of the threshold through two attributes:

Attribute	Purpose	Example Values
`thresholdOrigin`	Category of threshold origin	`SLA`, `SLO`, `POLICY`, `EMPIRICAL`

Attribute	Purpose	Example Values
`contractRef`	Human-readable reference to source document	`"SLA v2.1 §4.3"`, `"Policy-2024-Q1"`

Target Source Values

Value	Meaning	Hypothesis Framing
`SLA`	Service Level Agreement (contractual)	”System meets SLA requirement”
`SLO`	Service Level Objective (internal target)	”System meets SLO target”
`POLICY`	Organizational policy or standard	”System meets policy requirement”
`EMPIRICAL`	Derived from baseline experiment	”No degradation from baseline”
`UNSPECIFIED`	Not specified (default)	”Success rate meets threshold”

Impact on Hypothesis Formulation The `thresholdOrigin` influences how the framework frames the hypothesis test in detailed reports:

`thresholdOrigin`	H_0 (Null Hypothesis)	H_1 (Alternative)
`SLA`	System meets SLA requirement	System violates SLA
`SLO`	System meets SLO target	System falls short of SLO
`POLICY`	System meets policy requirement	System violates policy
`EMPIRICAL`	No degradation from baseline	Degradation from baseline
`UNSPECIFIED`	Success rate meets threshold	Success rate below threshold

This adaptation ensures that verdicts are framed in the appropriate business context, making reports immediately understandable to stakeholders.

---

8. Assumptions and Validity Conditions

8.1 When Normal Approximation is Valid

The normal approximation to the binomial is adequate when:

$$n \cdot p \geq 5 \quad \text{and} \quad n \cdot (1-p) \geq 5$$

More conservatively (for confidence intervals):

$$n \cdot p \geq 10 \quad \text{and} \quad n \cdot (1-p) \geq 10$$

For p = 0.95:

- Need $n \geq 200$ for conservative criterion
- Wilson interval recommended for $n < 200$

8.2 Independence Violations

If trials are not independent, the evidence carried by n trials is less than the evidence n independent trials would carry. The reduction is summarised by an effective sample size n_{eff}: the number of independent trials whose evidence is equivalent to the dependent run. When every pair of trials in the run is correlated to the same degree ρ (and $\rho \in [0, 1]$ summarises that shared pairwise correlation), n_{eff} takes the closed form

$$n_{\text{eff}} = \frac{n}{1 + (n-1)\rho}.$$

Detection: Run autocorrelation analysis on trial outcomes. Significant lag-1 autocorrelation suggests dependence.

Mitigation: Increase sample size or introduce delays between trials.

The n_{eff} expression rests on a strong simplification: it assumes that every pair of trials in the run is correlated to the same degree ρ, regardless of which two trials are chosen. Under that assumption a single number ρ summarises the dependence in the whole run, and the formula converts it into an effective sample size. This is useful as a first-cut sanity check — given a plausible ρ, how badly is the evidence in n trials inflated? — but the assumption rarely holds in probabilistic testing. Real workloads contain pairs of trials that are tightly linked (two invocations of the same prompt, two trials in the same provider batch, two consecutive calls under the same warm cache) alongside pairs that are essentially independent (trials drawn from different prompts, different batches, different time windows). Collapsing that variation into a single ρ understates the structure that matters.

For workloads of this kind — repeated prompts with intrinsically different per-prompt rates, stratified samplings with unequal stratum weights, time-series autocorrelation that decays with lag, batch effects, or nested clustering such as prompt within stratum within batch — the appropriate machinery is the hierarchical model of §8.2.1, generalised estimating equations (Liang & Zeger, 1986), cluster bootstrap, beta-binomial and other hierarchical estimators, or stratified estimators that respect the declared `targetEstimand`. The methodology does not commit to a single correction; it relies on the design metadata of §8.2.1 to make the form of the dependence visible, so that whichever estimator is used carries the assumptions it depends on.

8.2.1 Clustered Sampling Designs Many probabilistic-test workloads, particularly LLM-backed ones, do not present as n i.i.d. invocations of a uniformly-sampled prompt. A common pattern is to run r repetitions of each of J prompts drawn from a curated sampling, giving $n = r \cdot J$ invocations whose evidential content is materially smaller than n independent draws. A natural model is hierarchical:

$$X_{ij} \mid \theta_j \sim \text{Bernoulli}(\theta_j), \qquad \theta_j \sim G,$$

where j indexes prompts (or prompt strata) and G is the prompt-level rate distribution. Under this model, two prompts with intrinsically very different θ_j contribute jointly correlated evidence within each prompt and unequal evidence across prompts; the plain Binomial(n, p) model of §1.1–1.2 mis-states both the variance and the population to which the estimate generalises.

Required design metadata. Every trial record contributing to a baseline or a per-criterion verdict must carry the fields needed to detect and characterise this structure. At minimum:

Field	Purpose
`promptId`	Identifier of the input item under which the trial was generated. Repeated prompts share an ID.
`stratumId`	Stratum identifier when the sampling is stratified (failure mode, language, difficulty, region, ...). Optional when the set is unstratified.
`repetitionsPerPrompt`	Number of trials run against each prompt. Reported as a distribution when not constant.
`batchId`	Provider batch / request batch when observable.
`region` / `timeBlock`	Provider region and time bucket when observable; supports detection of batch and temporal effects.
`samplingMode`	One of: with-replacement, without-replacement, exhaustive, adaptive, externally-supplied.
`targetEstimand`	One of: prompt-weighted, call-weighted, production-weighted, severity-weighted, no-generalisation. Fixes which population the per-criterion p_c estimates.
`populationClaim`	One of: `finite-corpus`, `superpopulation`, `no-generalisation` (§8.4.6). Fixes the claim regime of the verdict; a mismatch between baseline and test is a §8.4.5 hard invalidator.

Field	Purpose
`weights`	Prompt, stratum, production, or severity weights, where the target estimand is weighted. Empty when the design is unweighted.
`strata`	Declared strata and their intended population weights, where stratified sampling is used. Empty when the sampling is unstratified.

The estimator must match the declared `targetEstimand`. A call-weighted estimator over a workload with materially uneven repetitions per prompt does not generalise to a prompt-weighted population, and conversely; the methodology does not silently substitute one for the other.

Treatment by intent. When the invocation count n materially exceeds the count of independent input items (e.g., $n > 2J$ for a workload of J distinct prompts), the design materially violates the i.i.d. Bernoulli interpretation, and the verdict's calibration under a plain Binomial(n, p) aggregation is no longer warranted. The methodology's response is intent-dependent.

In **SMOKE** mode, the warning of §8.4.4 is sufficient: it qualifies the verdict rather than blocking it, and the operator carries forward an explicitly degraded calibration claim.

In **VERIFICATION** mode, metadata disclosure alone does not restore the i.i.d. premise. The methodology requires one of the following before a population-level VERIFICATION claim is made:

1. an approved clustered, stratified, hierarchical, or design-based estimator matched to the declared `targetEstimand` (for example, a prompt-weighted estimator under the hierarchical model above, GEE per Liang & Zeger 1986, cluster bootstrap, or a beta-binomial estimator) — at which point the calibration claim applies to that estimator, not to the plain Wilson construction;
2. demotion of the claim to a no-generalisation or finite-corpus claim, explicitly labelled per §8.4.6, with the procedure's reach restricted to the run or the corpus actually evaluated;
3. an INCONCLUSIVE verdict or configuration-error treatment for any remaining population-level VERIFICATION claim.

A warning-only treatment of material cluster structure is not by itself sufficient for an affirmative VERIFICATION claim over a named population; under the §8.4.5 severity policy that cluster structure functions as a hard invalidator unless one of (1)–(3) has been satisfied.

Reference. Liang, K.-Y., & Zeger, S. L. (1986). ”Longitudinal data analysis using generalized linear models.” Biometrika, 73(1), 13–22.

8.3 Non-Stationarity

Non-stationarity—when the success probability p is not constant—is perhaps the most insidious threat to probabilistic testing. Unlike independence violations, which can sometimes be detected through autocorrelation, non-stationarity may be invisible in aggregate statistics while fundamentally invalidating comparisons.

8.3.1 Forms of Non-Stationarity

Form	Example	Detection Difficulty
Within-experiment drift	Model updates during a long MEASURE run	Moderate (time-series analysis)
Between-experiment drift	System changes between baseline and test	Hard (requires external knowledge)
Contextual variation	Different behaviour on weekdays vs weekends	Easy (if factors are known and tracked)
Gradual degradation	Slow performance decay over months	Hard (no single detectable event)

8.3.2 Statistical Consequences If p changes during the experiment:

- Point estimate $\hat{p}$ reflects time-averaged behaviour, not current behaviour
- Confidence intervals may understate true uncertainty
- Threshold derivations may be based on stale data
- Verdicts may systematically mislead in one direction

If p differs between baseline and test:

- The comparison is between **different populations**
- The hypothesis test answers the wrong question
- Type I and Type II error rates are no longer meaningfully targeted
- Verdicts are statistically meaningless (though they appear valid)

8.3.3 Why This Is Hard Non-stationarity is difficult because:

1. **It's often invisible in aggregate data**: A 95% pass rate could arise from stable 95% behaviour, or from 99% for half the samples and 91% for the other half.
2. **It can occur between experiments**: The system that generated the baseline may literally not exist anymore (different model version, different infrastructure).
3. **It can be caused by external factors**: Changes to dependencies, APIs, or infrastructure that the developer doesn't control or even know about.
4. **The statistical machinery assumes it away**: All the formulas in this document assume p is constant. When it isn't, the formulas still produce numbers—they're just wrong.

8.3.4 The mavai Approach No framework can guarantee stationarity. Instead, the mavai methodology provides **guardrails** that:

1. **Make context explicit**: Covariate declarations force developers to think about what factors might matter.
2. **Make drift visible**: Covariate non-conformance and expiration warnings surface potential violations.
3. **Preserve auditability**: Baseline provenance ensures the conditions of inference are always documented.
4. **Qualify rather than suppress**: Warnings accompany verdicts rather than replacing them.

See Section 8.4 for detailed descriptions of these guardrails.

8.3.5 Developer Responsibilities The framework provides tools; developers must use them wisely:

Responsibility	How the Framework Helps	What Developers Must Do
Identify relevant factors	Standard covariates for common cases	Declare covariates in service contract definitions
Track contextual changes	Automatic covariate resolution and matching	Ensure custom covariates are in environment
Recognise baseline staleness	Expiration warnings	Set appropriate expiration values
Investigate warnings	Clear warning messages with specifics	Don't ignore non-conformance warnings
Refresh stale baselines	Prominent expiration alerts	Run measure experiments when prompted

8.4 Guardrails for Assumption Validity

> **Epistemic status**: diagnostic guardrails / validity aids, not statistical corrections. The mechanisms below (covariates, expiration, provenance, warning semantics) do not repair a violated assumption; they surface its likely presence so an operator can judge whether the verdict is still trustworthy. A test that runs under non-conforming covariates still produces a statistically questionable verdict — the framework just refuses to let that fact be silent.

The statistical validity of probabilistic testing depends on the assumptions outlined in Section 1.3. While no framework can guarantee these assumptions hold, the mavai methodology provides **guardrails**—features that surface violations, qualify results, and encourage practices that preserve statistical validity.

These guardrails embody a key principle: **statistical honesty over silent convenience**. Rather than producing clean verdicts that hide uncertainty, the framework

makes the conditions of inference explicit and auditable.

8.4.1 Covariate-Aware Baseline Matching **The problem**: A baseline represents the empirical behaviour of a system under specific conditions. If a probabilistic test runs under different conditions—different time of day, different deployment region, different feature flags—the comparison may be invalid. The samples are drawn from **different populations**.

This is a violation of the **stationarity assumption**: the success probability p is not constant between baseline creation and test execution.

Example: A customer service LLM performs differently during peak hours (high load, queue delays) versus off-peak hours. A baseline measured at 2 AM may not represent behaviour at 2 PM.

The mavai solution: Developers declare **covariates**—exogenous factors that may influence success rates. The exact syntax varies by framework, but the concept is universal across all implementations.

This declaration is an explicit statement: "These factors may affect performance. Track them."

How it works:

1. During measure experiments, the framework records the covariate values as part of the baseline specification.
2. During probabilistic tests, the framework resolves the current covariate values and compares them against the baseline.
3. If values differ (non-conformance), the framework issues a **warning** that qualifies the verdict.

Statistical interpretation: Non-conformance does not change the pass/fail verdict. Instead, it **qualifies** the inference:

> "Under the assumption that success rates are comparable across these conditions, the test passes. However, this assumption may not hold—the baseline was created on a weekday, but the test is running on a weekend."

This transforms a hidden assumption into an explicit, auditable caveat.

Why this matters:

Without Covariates	With Covariates
Silent assumption that conditions don't matter	Explicit declaration of relevant conditions
Population mismatch is invisible	Population mismatch triggers warning
False confidence in verdicts	Qualified confidence with documented caveats
Statistical validity unknowable	Statistical validity auditable

8.4.2 Baseline Expiration **The problem**: Systems change over time. Dependencies update, models are retrained, infrastructure drifts. A baseline from six months ago may no longer represent current behaviour—even if all declared covariates match.

This is **temporal non-stationarity**: the success probability p changes over calendar time in ways that cannot be captured by discrete covariates.

The mavai solution: Developers declare a **validity period** for baselines.

This declaration is an explicit statement: "I believe this baseline remains representative for N days."

How it works:

1. The expiration value is recorded in the baseline specification along with the experiment end timestamp.
2. During probabilistic tests, the framework computes whether the baseline has expired.
3. As expiration approaches, the framework issues **graduated warnings**:

Time Remaining	Warning Level	Message
> 25% of validity period	None	—
≤ 25%	Informational	"Baseline expires soon"
≤ 10%	Warning	"Baseline expiring imminently"
Expired	Prominent	"BASELINE EXPIRED"

Statistical interpretation: Expiration does not change the pass/fail verdict. Instead, it signals:

> "This baseline is old. The system may have changed in ways not captured by covariate tracking. Interpret results with appropriate caution."

Complementary to covariates: Covariates catch **known, observable** context changes (weekday vs weekend, region). Expiration catches **unknown, gradual** drift (model updates, dependency changes, infrastructure evolution).

Guardrail	What It Catches	Mechanism
Covariates	Known context mismatch	Explicit factor declaration and matching
Expiration	Unknown temporal drift	Calendar-based validity period

Together, they provide **defence in depth** against non-stationarity.

8.4.3 Baseline Provenance **The problem**: A statistical verdict is only meaningful if its empirical foundation is known. "The test passed" means little without knowing: against what baseline? Under what conditions? With what caveats?

The mavai solution: Every test verdict includes explicit **baseline provenance**:

```
BASELINE REFERENCE
  File:        ShoppingServiceContract-ax43-dsf2.yaml
  Generated:   2026-01-10 14:45 UTC
  Samples:     1000
  Observed rate: 95.1%
  Covariates:  day_of_week=WEEKDAY, time_of_day=08:00/4h, region=EU_CORE
  Expiration:  2026-02-09 (27 days remaining)
```

This ensures that **no inference result is ever detached from its empirical foundation**.

Why this matters for statistical validity:

- Auditors can verify that comparisons are appropriate
- Operators can investigate unexpected results by examining baseline conditions
- Historical analysis can account for which baseline was in effect when
- Reproducibility is supported by explicit documentation

8.4.4 Explicit Warnings Over Silent Failures A consistent design principle across the mavai methodology's guardrails:

> **Warnings qualify verdicts; they do not suppress them.**

Condition	Verdict Impact	Warning
Covariate non-conformance	None	Yes
Expired baseline	None	Yes
Multiple suitable baselines	None	Yes

This principle reflects a statistical philosophy:

1. **The developer asked a statistical question** ("Does my system meet this threshold?"). The framework answers that question.
2. **The answer may have caveats** ("...but the baseline is old" or "...but the conditions differ"). The framework surfaces those caveats.
3. **The decision about whether to trust the answer remains with the human**. The framework provides information, not absolution.

This approach preserves statistical honesty without creating operational paralysis. Tests don't mysteriously skip or fail due to metadata issues—they run, and their limitations are documented.

8.4.5 Guardrail Severity Levels The ”warnings qualify, do not suppress” principle of §8.4.4 is the right default for SMOKE-intent runs and for diagnostic guardrails whose violation degrades but does not destroy the comparability of the empirical evidence. It is the wrong default for VERIFICATION- intent runs against guardrails whose violation makes the empirical comparison statistically meaningless. A regression FAIL produced by comparing today's behaviour to a baseline of a different model, a different prompt version, a different evaluator, or a different criterion structure is not a regression FAIL at all — it is a configuration error mis-presented as evidence. The methodology classifies guardrails into three severity tiers and binds them to mode-specific behaviour:

Severity	Examples	VERIFICATION behaviour	SMOKE behaviour
Hard invalidator	Structural-reference mismatch (§1.5.2); model-ID mismatch (non-stationarity); denominator-policy mismatch (different estimand, §1.4.5a); material cluster structure (§8.2.1) without an approved estimator matched to the declared `targetEstimand` and without demotion to a no-generalisation or finite-corpus claim (§8.4.6)	Configuration error — no qualified verdict is emitted; the run is rejected.	Warning by default; promotion to error by policy.
Major caveat	Expired baseline beyond a stated grace window; unmatched critical covariate; changed endpoint, region, or serving-stack revision	Default error unless an explicit policy override admits the run with a qualified verdict.	Warning.
Minor caveat	Near-expiration (within graduated-warning band of §8.4.2); non-critical covariate mismatch; near-grace-window batch metadata gaps	Warning.	Warning.

Hard invalidators in VERIFICATION are not qualifying caveats; they are configuration errors. A VERIFICATION run that detects model-ID, denominator-policy, structural-reference, or unmitigated material cluster-structure violations terminates with a configuration-error verdict before any statistical inference is attempted. A ”PASS with caveat” or ”FAIL with caveat” verdict against a hard-invalidator violation

under VERIFICATION is not produced; the comparison the operator asked the framework to make does not exist under that configuration. SMOKE runs may proceed under operator policy, since SMOKE carries no verification claim (§5.7.2).

Major caveats preserve the option to override the default error under explicit operator policy — useful for short, controlled runs against a baseline that has just expired, for example, where the operator is willing to read the verdict subject to the disclosed caveat. The override must be recorded on the verdict; it is not an implicit relaxation.

Minor caveats retain the §8.4.4 ”qualify, do not suppress” behaviour in both modes. Their statistical impact is bounded; their disclosure is enough.

This stratification prevents the failure mode the §8.4.4 default otherwise admits: a verification report that looks valid, carries an inconspicuous warning footnote, and reports a regression FAIL whose true cause is that the empirical comparison was no longer between the same two systems.

8.4.6 Population-Claim Discipline: Finite-Corpus vs Superpopulation The Wilson, binomial, and order-statistic machinery developed in §§2–4, §7, and §12 produces inferential claims — statements about a population parameter p_c or $Q(p_j)$ that go beyond the sampled trials themselves. An inferential claim is only meaningful relative to a stated population. Every criterion's verdict declares, on the contract, which of three claim regimes its evidence supports. This declaration sits alongside the design metadata of §8.2.1 (`populationClaim`, `samplingMode`, `weights`, `strata`); a mismatch between baseline and test is a §8.4.5 hard invalidator because the two are comparing different estimands.

Finite-corpus claim. The sampling is a fixed, enumerated corpus of inputs, and the criterion's inferential claim applies only to that corpus. Under an exhaustive evaluation (n_c equals the full corpus size), the corpus rate is known exactly for the evaluated corpus: $p_c = K_c/n_c$. No binomial confidence interval is reported for the finite-corpus estimand — Wilson and other binomial sampling-uncertainty constructions do not apply, because there is no sampling uncertainty about a known quantity. Any claim made beyond the corpus is a separate superpopulation claim and must be labelled as such. Under a partial evaluation (sampling without replacement from the corpus), the relevant inferential machinery is the **hypergeometric** distribution, not the binomial; the binomial approximation is acceptable only when the sample fraction is small. The methodology accordingly carries the sample fraction alongside any finite-corpus inferential claim, and a binomial approximation in place of the structurally-correct hypergeometric is disclosed as such on the verdict.

Superpopulation claim. The sampling is treated as a sample from a conceptual population — the adversarial-probe distribution, the production input distribution, the red-team-corpus generator, or whichever distribution the criterion intends p_c to characterise. Under a superpopulation claim, the binomial / hypergeometric inferential machinery is the right tool, and Wilson intervals, sample-size budgets, and feasibility gates all carry their usual interpretations. The claim is only as strong as the argument that the sample is representative of the named superpopulation; the methodology does not validate this argument, but it does require the named superpopulation to be recorded on the verdict (§1.4.7, `populationClaim` field of §8.2.1).

No-generalisation claim. The trials are convenience evaluations — ad hoc, illustrative, exploratory. No inferential claim is made beyond the specific trials run. PASS / FAIL labels on a no-generalisation criterion describe the run, not a population, and the framework reports them as such. A no-generalisation claim is appropriate for early-stage development, smoke testing, and diagnostic probes; it is **not** appropriate for compliance, regression, or release-gating evidence.

Where this most often goes wrong: safety-probe and red-team sets. Curated safety-probe and red-team samplings are typically **finite-corpus** by design — the corpus is the artefact, and its inputs are chosen to exercise specific failure modes rather than to represent a production distribution. Reading $\hat{p}_c$ on such a set as a superpopulation rate (production self-harm rate, production jailbreak rate) inflates the claim well beyond what the evidence supports: an adversarially curated probe set is by construction more adverse than the production distribution, and a clean Wilson interval over it does not bound the production failure rate. The methodology warns explicitly against this slide: a red-team verdict is a verdict on the red-team corpus, and any extension to production traffic requires a parallel sentinel stream (§1.4.7's clinical-advice example illustrates the pattern). Under the methodology a safety-probe verdict is not labelled superpopulation absent an explicit operator declaration and a recorded representativeness argument; the default classification is finite-corpus, and elevation to superpopulation requires the operator to assume — and the verdict to record — the representativeness claim that elevation depends on.

Operational binding. The three regimes are recorded in the trial record's `populationClaim` field per §8.2.1. A baseline labelled `superpopulation` cannot be used as the reference for a test labelled `finite-corpus` (and vice versa) without the configuration-error treatment of §8.4.5 — the two are not comparing the same object. Mode transitions between baseline and test are likewise hard invalidators in VERIFICATION; an operator who genuinely intends to re-baseline under a new regime starts a new reference-state epoch (§8.5).

8.5 Repeated Use and Sequential Monitoring

The probabilistic-test workflows the methodology targets are not single scientific studies — they are CI builds, nightly runs, release gates, and continuous monitors. A configured per-run nominal α is the calibration of one invocation of the procedure. It is not the long-run false-alarm probability over a horizon of repeated invocations, and it must not be reported as such.

Monitoring horizon. A monitor's calibration is meaningful only relative to a stated horizon H — the number of independent evaluations against which the operator wants the false-alarm probability bounded. The horizon may be expressed as a number of runs, a wall-clock window (seven days, one quarter), or a number of release gates. A horizon-level claim is meaningful only with respect to a declared H, and under the methodology the horizon is part of the monitor's metadata wherever a horizon-level claim is made.

Per-run nominal α versus horizon-level false-alarm probability. Over H invocations of a procedure with per-run nominal α against a fixed reference state, the horizon-level false-alarm probability satisfies

$$P(\text{at least one false alarm in } H \text{ runs}) \;\leq\; H \cdot \alpha \qquad \text{(Bonferroni)}$$

regardless of the dependence structure across runs, and approaches $1-(1-\alpha)^H$ under independence. At $\alpha = 0.05$ and $H = 50$ the Bonferroni bound is 2.5 (i.e., the bound is uninformative and the expected number of false alarms is $H\alpha$); the independence approximation gives $\approx 92\%$. **One historical false alarm in a long-running monitor is therefore not, by itself, evidence of method failure** — it is the expected behaviour of a correctly calibrated test repeated many times. Method failure is diagnosed against an excess rate of false alarms relative to the horizon-level expectation, not against a single occurrence.

Alpha spending and sequential designs (further reading). When horizon-level control matters operationally — a quarterly compliance window, a release-train gate — the sequential-testing literature provides families of procedures that distribute a horizon-level α budget across runs:

- alpha-spending functions (O'Brien–Fleming, Pocock) that allocate the budget across pre-planned interim looks;
- statistical process control / control-chart approaches (Shewhart, CUSUM, EWMA) for continuous monitors;
- group-sequential and always-valid sequential procedures (e.g., mSPRT, confidence sequences).

The methodology does not impose a particular sequential procedure. Operators who require horizon-level guarantees stronger than the Bonferroni $H\alpha$ bound select a procedure whose dependence assumptions match their workflow, and report the horizon-level calibration alongside the per-run calibration.

Baseline refresh as a new reference state. A baseline refresh ends one reference state and begins another, and is recorded as a new reference-state epoch in the verdict stream. The verdict stream is therefore partitioned by epoch, and horizon-level calibrations are computed within an epoch, not across one. Cross-epoch trends are descriptive, not inferential.

Verdict metadata under recurring monitoring. A verdict produced inside a recurring monitor carries, in addition to its per-run nominal α:

- the monitoring horizon H;
- the per-run nominal α;
- the horizon-level false-alarm bound under the stated reference ($H\alpha$ as the dependence-robust statement; the independence value as a contextual lower bound where it is computable);
- the reference-state epoch the verdict belongs to.

A run-level PASS/FAIL stripped of these annotations is not a horizon-level claim; reporting it as one inverts the relationship between per-run and horizon-level calibration.

---

9. Summary of Key Formulas

Every formula below is tagged with its **epistemic status**:

- **Exact** — a theorem under the stated model assumptions.
- **Wilson score construction** — a score-test inversion for the binomial proportion. Not an exact finite-sample procedure; chosen as the methodology's default for its stability near the boundary and good practical coverage relative to Wald.
- **Score-test inversion** — a one-sided confidence bound obtained by inverting the score test. Used here as an **operational approximation** for threshold derivation; it is calibrated asymptotically and behaves well in the operating regime, but is not exact.
- **Operational approximation** — a procedure adopted for engineering reasons (stability, reproducibility, simplicity); calibration is good in stated regimes but is not a theorem.
- **Asymptotic / Normal-approximation** — a planning formula valid when n is large and p is away from 0 and 1.
- **Heuristic** — a rule of thumb, useful operationally, not a confidence statement.
- **Non-parametric / distribution-free** — exact for any continuous F under i.i.d. sampling.

Estimation (Exact — MLE)

$$\hat{p} = \frac{k}{n}, \quad \mathrm{SE}(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Wald Confidence Interval (two-sided) (Asymptotic — pedagogical, not used by the methodology)

$$\hat{p} \pm z_{\alpha/2} \cdot \mathrm{SE}(\hat{p})$$

Wilson Score Interval (Wilson score construction — default interval method)

$$\frac{\hat{p} + \frac{z^2}{2n} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}$$

One-Sided Lower Bound, for threshold derivation (Operational approximation — Wald form, see §3.3; the methodology uses the Wilson score-test inversion in §3.4)

$$p^* = \hat{p} - z_\alpha \cdot \mathrm{SE}$$

Wilson Lower Bound, for $\hat{p} = 1$ (Wilson score construction — boundary case)

$$p_{\text{lower}} = \frac{n}{n + z^2}$$

Rule of Three, for zero failures (Heuristic — quick approximation at 95% confidence)

$$p \geq 1 - \frac{3}{n}$$

Cross-reference: §1.4.5 admits this as an optional contextual annotation on observational verdicts under an explicitly-stated i.i.d. Bernoulli model. The annotation does not change the observational verdict label.

Sample Size for Precision (Asymptotic — planning approximation based on normal asymptotics)

$$n = \frac{z_{\alpha/2}^2 \cdot p(1-p)}{e^2}$$

Sample Size for Power (Asymptotic — planning approximation based on normal asymptotics)

$$n = \left(\frac{z_\alpha \sqrt{p_0(1-p_0)} + z_\beta \sqrt{p_1(1-p_1)}}{p_0 - p_1} \right)^2$$

Empirical Percentile (nearest-rank) (Exact — definition)

$$Q(p) = t_{(\lceil p \cdot n_s \rceil)}, \quad t_{(1)} \leq \cdots \leq t_{(n_s)}$$

Latency Threshold Derivation (binomial order-statistic upper bound) (Non-parametric / distribution-free when the required rank exists within the sample (§12.5.2.1); saturated and advisory only otherwise. Exact for continuous F_T; conservative under ties.)

$$\tau_j = t_{(k_j)}, \qquad k_j = \min\left\{k : P\big(\mathrm{Bin}(n_s, p_j) \geq k\big) \leq \alpha\right\}$$

Equivalently, $k_{\text{raw}} = \texttt{qbinom}(1-\alpha, n_s, p_j) + 1$. If $k_{\text{raw}} \leq n_s$, $\tau_j = t_{(k_{\text{raw}})}$ — exact distribution-free upper confidence bound, integer-ms by construction (conservative under ties). If $k_{\text{raw}} > n_s$, the required rank saturates beyond the sample and no finite-sample distribution-free upper confidence bound is available at the configured confidence (§12.5.2.1): under VERIFICATION the verdict is INCONCLUSIVE; under SMOKE or advisory reporting, $t_{(n_s)}$ may be displayed with `saturated: true` but is not an exact bound.

---

10. Transparent Statistics Mode

10.1 Purpose

Transparent Statistics Mode exposes the complete statistical reasoning behind every test verdict. This feature serves:

- **Auditors**: Documented proof that testing methodology is statistically sound
- **Stakeholders**: Evidence that reliability claims are justified
- **Educators**: Teaching material for understanding probabilistic testing
- **Regulators**: Compliance documentation for AI system validation

10.2 Output Structure

Under criterion decomposition (§1.4), a contract's verdict is a structured tuple over its per-criterion verdicts. Transparent Statistics Mode exposes the contract-level composite at the top, followed by a per-criterion analysis block for each criterion declared on the contract.

Contract-level header.

Section	Content	Statistical Purpose
Composite Verdict	PASS / FAIL / INCONCLUSIVE per §1.4.6, with the triggering criterion(a) named	At-a-glance answer to ”did the contract pass?”
Per-Criterion Summary	One line per criterion: verdict (PASS/FAIL/INCONCLUSIVE) and mode (inferential/observational)	Surface every per-criterion outcome on the front page
Type-I envelopes	Procedure-direction-specific disclosed envelopes: $\alpha_{\text{false-degradation-signal}} \leq \sum_{c \in \text{regression}} \alpha_c$ for regression criteria, and $\alpha_{\text{false-compliance}} \leq \sum_{c \in \text{compliance}} \alpha_c$ for compliance criteria. A contract reports each envelope only when it carries criteria of that direction (§1.4.6).	The composite verdict's family-wise bounds, labelled by procedure direction

Per-criterion analysis block — repeated for each criterion declared on the contract.

Section	Content (inferential criterion)	Content (observational criterion)
Hypothesis Test	H_0, H_1, test type, α_c	Mode declaration (”observational”); no H_0/H_1

Section	Content (inferential criterion)	Content (observational criterion)
Observed Data	Denominator: $n_c = N$ (the sampling size, §1.4.5a), K_c, $\widehat{p}_c = K_c/n_c$; diagnostics $n_{c,\text{evaluable}}$ and $r_{c,\text{obs}} = n_{c,\text{evaluable}}/n_c$ (the transform/no-value share), and the FAIL breakdown by reason.	Denominator: $n_c = N$, K_c, failures observed; diagnostics $n_{c,\text{evaluable}}$, $r_{c,\text{obs}}$, and the FAIL breakdown by reason.
Threshold Reference	Threshold origin and derivation (see below)	(omitted — no threshold)
Statistical Inference	SE, CI, Wilson lower bound, integer cutoff c, displayed cutoff c/n, achieved size, z (diagnostic), p-value (per §7.1 alignment rule)	(omitted — verdict is deterministic on the observation)
Verdict	Three strands: statistical / observed-rate / operational	Zero-failure assertion with explicit "no population claim" caveat

The denominator n_c is the size N of the sampling (§1.4.5a): every trial counts, a trial with no testable value being a FAIL (reason: transform/no-value), never excluded.

The **Threshold Reference** section, when shown, adapts to the criterion's origin:

Origin	Content displayed
SLA / SLO / POLICY	Threshold origin, contract reference, normative threshold p_c^*, α_c
EMPIRICAL	Baseline identifier, baseline $(\widehat{p}_c^{\text{baseline}}, n_c^{\text{baseline}})$, covariate-match status, test sample size $n_{c,\text{test}}$, derived p_c^* (§1.5.4)

The **inferential verdict's three strands** (per §10.3's example, applied per inferential criterion):

- **Statistical verdict** — the hypothesis-test conclusion under the declared procedure (§3.2). For a **compliance** criterion: PASS iff the one-sided Wilson lower bound $\widehat{p}_{c,L}(\alpha_c)$ exceeds the required threshold p_{req}. For a **regression** criterion: PASS iff the observed success count satisfies the integer-cutoff rule $K_c \geq c_c$, where c_c is derived from the reference distribution at α_c (§3.4); the Wilson lower bound and displayed threshold are reported as threshold-reference and diagnostic context, not as the binding regression decision.
- **Observed-rate status** — whether the point estimate $\widehat{p}_c$ sits on the right side of the displayed threshold. Can disagree with the statistical verdict, especially near the boundary; the disagreement is the point of disclosing both.
- **Operational caution** — what an operator should do next: sample-size adequacy, power against plausible regressions, follow-up recommendations.

Observational criteria do not carry the three strands; their verdict is deterministic on the observation and a single assertion line suffices.

Reproducibility metadata. A transparent-statistics report carries a small reproducibility block, recorded once per contract-level report (not per criterion), naming the numerical ingredients of the verdict that vary across language ecosystems. The block exists so that an auditor or downstream conformance consumer can reproduce the report's numbers exactly against the mavai-R fixtures, and so that divergence is detected early when an implementation upgrades a dependency that perturbs a quantile or a sort.

Field	Content	Purpose
`referenceDataVersion`	The mavai-R fixture release tag consumed at runtime, e.g. `v0.7.0`.	Identifies the conformance oracle the report was checked against.
`quantileSource`	The implementation source of the normal quantile $z_{1-\alpha}$ used in the Wilson construction — e.g. `R::qnorm`, `Apache Commons Math NormalDistribution`, `statrs::distribution::Normal::inverse_cdf`.	Different libraries can disagree in the last few ulp of $z_{1-\alpha}$; this names the source for cross-language conformance reproduction.
`intervalImplementation`	The Wilson formula variant in use: `closed-form score-inversion` or `iterative` with the solver tolerance.	The closed-form Wilson interval is exact-arithmetic up to floating-point; an iterative variant introduces a tolerance that must be disclosed.

Field	Content	Purpose
`sortStable`	Boolean — whether the sort algorithm used on the latency stream is stable.	Pairs with `tiePolicy`; see §12.8. Stability does not change percentile values but does change which trial-record is bound to a given rank when ties are present.
`tiePolicy`	The tie-handling rule actually used — for the mavai default, `largest-tied-position-for-upper-bound-ranks` (§12.8).	Confirms the implementation is consistent with the conservative-bound rule of §12.8.

These fields are emitted whether or not the contract under test exercises the latency dimension; the latency-specific fields (`sortStable`, `tiePolicy`) carry their declared values even on pass-rate-only contracts so that the report's reproducibility shape does not depend on the contract's content. The reproducibility block is part of the audit trail, not a verdict input — it carries no PASS/FAIL semantics of its own. The forward-looking calibration fixtures of §10.6 are checked against the same `referenceDataVersion`.

10.3 Example output: a multi-criteria contract

The consult-advice contract of §1.4.8 declares three criteria of differing origins — one EMPIRICAL inferential, one observational, one SLO inferential — and runs them across three MEASURE experiments, each with its own sampling. The transparent-statistics output shows the contract-level composite first, then one analysis block per criterion. Inferential blocks carry the three-strand verdict; the observational block reports a single deterministic assertion.

```
═══════════════════════════════════════════════════════════════════════
STATISTICAL ANALYSIS: ConsultAdviceContract
═══════════════════════════════════════════════════════════════════════

COMPOSITE VERDICT
  Contract verdict:    FAIL  (triggered by C_layperson-readable)
  Per-criterion:       C_well-formed         PASS  (inferential, EMPIRICAL)
```

```
                      C_no-self-harm        PASS  (observational)
                      C_layperson-readable  FAIL  (inferential, SLO)
  Type-I envelopes (by procedure direction, §1.4.6):
    False-degradation-signal envelope:
      regression criteria        ∑ α_c  ≤  0.05      (C_well-formed)
    False-compliance envelope:
      compliance criteria        ∑ α_c  ≤  0.001     (C_layperson-readable)
    (Observational criteria — C_no-self-harm — contribute to neither.)

──────────────────────────────────────────────────────────────────────────────
CRITERION 1 of 3: C_well-formed                     (inferential, EMPIRICAL)
──────────────────────────────────────────────────────────────────────────────

HYPOTHESIS TEST
  procedure:        REGRESSION
  H₀ (null):        K follows the reference model; no degradation signal
                    relative to the EMPIRICAL baseline (p_c ≥ p*_c)
  H₁ (alternative): Lower-tail degradation signal (p_c < p*_c)
  decisionRule:    PASS iff K_c ≥ c_c (integer cutoff derived at α_c = 0.05)
  Test type:        One-sided Wilson lower bound, score-test inversion
  Sampling:   V_prod  v5  (redacted production prompts, policy XYZ)

OBSERVED DATA
  Attempted trials:              1000
  Evaluable trials:              1000
  Observation rate (r_obs):       1.000
  Effective denominator (n_c):   1000
  Successes (K_c):                953
  Observed rate (p̂_c):           0.953

THRESHOLD REFERENCE
  Threshold origin:    EMPIRICAL
  Baseline ID:         consult-advice@2026-04-01
  Baseline observation: p̂_c (baseline) = 0.951  over  n_c (baseline) = 2000
  Covariate match:     OK   (test indexed at same point in covariate space)
  Test sample size:    n_{c,test} = 1000
 Derived threshold:  p*_c  =  WilsonLB(0.951; 1000, 0.05)  ≈  0.9385  (§1.5.4)

STATISTICAL INFERENCE
 Standard error:    SE_c = √(p̂_c(1-p̂_c)/n_c) = √(0.953 × 0.047 / 1000) ≈ 0.00673
  95% Wilson CI:       [0.938, 0.965]
  Wilson lower bound:  p̂_{c,L}(0.05) ≈ 0.940
  Diagnostic z-score (Wald form, illustrative — not the operative test):
                       z = (p̂_c - p*_c) / √(p*_c(1-p*_c)/n_c) ≈ 2.13
  Achieved size at integer cutoff:
    method:            exact-binomial-lower-tail
    reference:         p_ref = 0.951
    cutoff:            c_c = 939
```

```
    tail:              P_{p = 0.951}(K < 939)
    value:             ≈ 0.0371
    interpretation:    false-degradation-alarm probability under the
                       empirical reference at the configured cutoff.

  Diagnostic observed-count tail:
    tail:              P_{p = 0.951}(K ≤ 953)
    value:             ≈ 0.636
    interpretation:    observed count is not in the lower tail of the
                       reference distribution.

VERDICT
  Statistical verdict:  PASS
                        Observed successes K_c = 953 meet the regression
                        integer cutoff c_c = 939 (= ⌈n · p*_c⌉ with
                        n = 1000, p*_c = 0.9385). No degradation signal
                        at α = 0.05 under the reference-control procedure.
                        The Wilson lower bound p̂_{c,L}(0.05) ≈ 0.940 and
                        the derived displayed threshold p*_c ≈ 0.9385 are
                        reported as threshold-reference and diagnostic
                        context for auditability; they are not the
                        binding regression decision rule.

  Observed-rate status: ABOVE THRESHOLD
                        Observed 0.953 > 0.9385. The point estimate and the
                        statistical verdict agree.

  Operational caution:  ADEQUATE POWER
                        At n_c = 1000, the test detects a drop to 0.925
                        with power ≈ 0.80. Margin above threshold is
                        modest; watch for downward drift over the
                        baseline's validity window.

═══════════════════════════════════════════════════════════════════════════
CRITERION 2 of 3: C_no-self-harm                            (observational)
═══════════════════════════════════════════════════════════════════════════

HYPOTHESIS TEST
  Mode:             Observational  (no threshold, no α, no inferential claim)
  Question:           Was any failure of C_no-self-harm observed in the run?
  Sampling:      V_probe  v3  (adversarial self-harm probes,
                       independent of guardrail's training data,
                       per mavai-safety-taxonomy v2)

OBSERVED DATA
  Attempted trials:               200
  Evaluable trials:               200
  Observation rate (r_obs):       1.000
```

```
  Effective denominator (n_c):    200
  Successes (K_c):                200
  Failures observed:                0

VERDICT
  Result:              PASS  (observational)
  Statement:           No failure of C_no-self-harm was observed across
                       200 probe trials drawn from V_probe v3.

  No population claim: This verdict makes no inferential claim about the
                      population rate of self-harm responses. A population-
                     level claim would need a separate inferential criterion
                       at adequate sample size, or architectural discharge
                       (§1.4.5b).

──────────────────────────────────────────────────────────────────────────────
CRITERION 3 of 3: C_layperson-readable                    (inferential, SLO)
──────────────────────────────────────────────────────────────────────────────


HYPOTHESIS TEST
  procedure:        COMPLIANCE
  H₀ (null):        p_c ≤ p_req = 0.98
  H₁ (alternative): p_c > p_req = 0.98
  decisionRule:    PASS iff one-sided Wilson lower bound on p_c exceeds p_req
  Test type:     One-sided Wilson lower bound, score-test inversion, α_c = 0.001
  Sampling:   V_complexity  v2  (inputs eliciting clinical terminology)

OBSERVED DATA
  Attempted trials:               800
  Evaluable trials:               800
  Observation rate (r_obs):       1.000
  Effective denominator (n_c):    800
  Successes (K_c):                788
  Observed rate (p̂_c):           0.985

THRESHOLD REFERENCE
  Threshold origin:    SLO
  Contract ref:        Consult-Advice SLO v2 §3.4 (layperson readability)
  Normative threshold: p*_c = 0.98
  Confidence level:    α_c  = 0.001

STATISTICAL INFERENCE
  Standard error:    SE_c = √(p̂_c(1-p̂_c)/n_c) = √(0.985 × 0.015 / 800) ≈ 0.00430
  One-sided Wilson lower bound at α_c = 0.001: p̂_{c,L} ≈ 0.9649
  Diagnostic z-score (Wald form, illustrative — not the operative test):
                       z = (p̂_c - p*_c) / √(p*_c(1-p*_c)/n_c) ≈ 1.01
  p-value:
    method:            score-test for H₀: p_c ≤ 0.98 vs H₁: p_c > 0.98
```

```
    tail:               upper-tail evidence for p_c > p_req
    value:              ≈ 0.156

VERDICT
  Statistical verdict:  FAIL
                        Wilson lower bound 0.9649 does not clear the SLO
                        threshold 0.98 at α = 0.001. Insufficient evidence
                        to conclude SLO compliance at the stated confidence
                        level.

  Observed-rate status: ABOVE THRESHOLD
                        Observed 0.985 > 0.98. The point estimate sits
                        above the SLO, but the lower bound does not. The
                        statistical verdict and observed-rate status
                        disagree; the disclosed strands surface the
                        tension that a single PASS/FAIL would hide.

  Operational caution:  INCREASE SAMPLE OR REVISIT α
                        At α_c = 0.001 the threshold is unusually tight;
                        n_c = 800 is insufficient to clear it at the
                        observed point estimate. Either increase n_c to
                        ≈ 7500 to support the SLO at α = 0.001, or
                        re-examine the contract's choice of α against the
                        consequence the SLO defends against (§1.4.6).
```

Reading the example.

- The composite verdict is FAIL, triggered by C_layperson-readable. The two passing criteria are reported in full alongside; the methodology does not collapse the contract to a single FAIL label without disclosing per-criterion evidence.
- The Type-I envelopes are disclosed properties of the composite, not control targets (§1.4.6), and are split by procedure direction. The false-degradation-signal envelope is $\sum_{c \in \text{regression}} \alpha_c \leq 0.05$ (C_well-formed); the false-compliance envelope is $\sum_{c \in \text{compliance}} \alpha_c \leq 0.001$ (C_layperson-readable). The observational criterion contributes to neither.
- C_well-formed and C_layperson-readable demonstrate the two **Threshold Reference** shapes: EMPIRICAL (baseline-derived, with covariate-match disclosure and a back-reference to §1.5.4) versus SLO (contract-referenced, normative). The blocks share otherwise-identical structure; only the threshold's provenance differs.
- C_layperson-readable's verdict shows the three strands disagreeing — statistical FAIL, observed-rate above threshold. The disclosure is the point of the three-strand format: an overloaded ”FAIL (close to threshold)” label would lose the information that the point estimate sits above the SLO but the inference at α = 0.001 does not support the claim.
- C_no-self-harm omits Threshold Reference and Statistical Inference entirely; its

verdict is the zero-failure assertion of §1.4.5, with the explicit non-claim about population rates that the methodology requires of observational verdicts.

10.4 Mathematical Notation

The output uses proper mathematical symbols where terminal capabilities allow:

Concept	Unicode	ASCII Fallback
Sample proportion	$\hat{p}$ (p̂)	p-hat
Population proportion	π	pi
Null hypothesis	H_0	H0
Alternative hypothesis	H_1	H1
Less than or equal	$\leq$	<=
Greater than or equal	$\geq$	>=
Square root	$\sqrt{}$	sqrt
Approximately	$\approx$	~=
Alpha (significance)	α	alpha

10.5 Validation by Statisticians

The transparent output enables statisticians to verify:

1. **Hypothesis formulation**: Is the one-sided test appropriate?
2. **Threshold derivation**: Was the Wilson lower bound correctly applied?
3. **Confidence interval**: Is the Wilson score interval used correctly?
4. **Sample size adequacy**: Are the caveats about power appropriate?
5. **Interpretation**: Does the plain-English summary accurately reflect the statistics?

10.6 Calibration Conformance

The published mavai-R fixtures (§1.5, `inst/cases/*.json`, the `fetchConformanceData` pipeline named in the project's `CLAUDE.md`) today carry **formula-value** fixtures: per-input `(inputs, expected)` cases whose `expected` is the numerical output of the companion's formulae as computed by R. A downstream framework whose Wilson lower bound, z-quantile, integer cutoff, or order-statistic rank agrees with the fixture's `expected` to a stated tolerance is in arithmetic step with the oracle. This is necessary, but it does not by itself demonstrate that the framework's verdicts are **calibrated** — that the false-alarm and power rates the framework claims are the ones it achieves under a stated reference model.

For full statistical conformance the methodology requires mavai-R to publish, in addition to the existing formula-value fixtures, a class of **calibration fixtures**:

- **Pass-rate calibration.** For representative (p_0, n, α) tuples — covering at minimum the worked examples of §§3 and 5 and the boundary cases of §4 — the fixture records the **achieved** false-alarm probability $P_{p_0}(K < c)$ at the integer

cutoff c derived by the Wilson construction, and (where a minimum detectable effect δ is supplied) the **achieved** power against $p_0 - \delta$. Downstream frameworks check not only that their cutoff matches the fixture's c but that their long-run false-alarm rate, computed by their own binomial machinery, matches the fixture's achieved size within a stated tolerance.

- **Latency calibration.** For representative (p_j, n_s, α) tuples covering p50, p90, p95, p99 and the feasibility-gate boundary of §12.5.2.1, the fixture records the **achieved coverage** of the order-statistic upper bound of §12.4.2 — i.e. the probability that the bound covers the population quantile $Q(p_j)$ under a stated reference distribution (continuous and representative integer-millisecond-tied cases). Downstream implementations verify their bound's coverage against the fixture rather than only its rank.
- **Composite envelope calibration.** For representative contracts with two or three inferential criteria, the fixture records, **separately by procedure direction**, the achieved family-wise rates of the §1.4.6 envelopes under stated dependence regimes (independent, positively correlated, adversarially correlated):
  - the **false-compliance envelope** — achieved long-run probability of declaring at least one compliance criterion satisfied when the corresponding true rate is at or below requirement — bounded by $\sum_{c \in \text{compliance}} \alpha_c$;
  - the **false-degradation-signal envelope** — achieved long-run probability of issuing at least one degradation signal when the stated reference holds — bounded by $\sum_{c \in \text{regression}} \alpha_c$. A combined unlabelled aggregate is not sufficient: the two envelopes describe different error events and may be set at different magnitudes within the same contract.

The conformance contract is: a downstream framework whose verdicts agree numerically with the formula-value fixtures **and** whose empirical error rates match the calibration fixtures within tolerance is conformant. A framework that passes the first but fails the second has either a calibration bug or has surfaced a defect in the calibration fixture itself; both are first-class outcomes of the closed loop named in the project `CLAUDE.md`.

Forward-looking status. At the time of this writing, the calibration fixtures described above are not yet published; only the formula-value fixtures are. This subsection states the **requirement on future mavai-R releases**, not a present claim of fixture availability. Conformance-test scaffolding in punit and feotest should be structured so the calibration-fixture path can be wired in without restructuring the existing formula-value conformance.

Two conformance statuses. The forward-looking calibration fixtures motivate a strict distinction between two conformance statuses, which the methodology treats as non-interchangeable:

- **Formula-value conformance.** The implementation's Wilson bounds, integer cutoffs, quantiles, and order-statistic ranks agree with the mavai-R formula-value fixtures within the stated tolerance. This demonstrates arithmetic agreement with the oracle on the inputs covered by the fixtures.

- **Statistical calibration conformance.** The implementation additionally agrees with the published calibration fixtures for achieved false-alarm probability, achieved power, achieved latency-bound coverage, and achieved family-wise rates of the §1.4.6 envelopes under their stated dependence regimes.

Until calibration fixtures are published, an implementation that passes the formula-value fixtures has demonstrated formula-value conformance only. The methodology does not recognise "statistically calibrated implementation" claims absent calibration-fixture agreement; a conformant transparent-statistics report distinguishes the two statuses in its metadata. A minimal structural example of the corresponding block is:

```yaml
conformanceStatus:
  formulaValueFixtures: passed
  calibrationFixtures: not-published
  calibrationClaimPermitted: false
```

When calibration fixtures are published, the same block surfaces their status (`passed`, `failed`, `partial`) and the `calibrationClaimPermitted` flag is set accordingly. The field's purpose is to make any overclaim visible at the report level rather than buried in the implementation's documentation.

10.7 Conformance Checklist

A conformant implementation of this companion is one for which the following hold jointly. The list is descriptive — each item points to the section in which the underlying property is defined — and is intended as an audit summary, not as a re-statement of the methodology.

- Compliance and regression verdicts are kept separate, and each verdict carries the procedure type alongside the null, the alternative, and the decision rule (§3.2, §7.1).
- Decisions are made on integer cutoffs c (regression) or on the lower-confidence-bound clearing p_{req} (compliance), not on rounded displayed rates (§3.4, §3.6).
- The achieved size under the stated reference is reported alongside every integer cutoff (§3.4).
- When a p-value is reported, it carries its method, null, alternative, and tail, and matches the orientation of the decision rule that produced the verdict (§7.1, §10.2).
- In latency VERIFICATION, a saturated order-statistic rank does not constitute an exact bound; the verdict is INCONCLUSIVE (§12.4.2, §12.5.2.1).
- For clustered or repeated-prompt designs, the report either applies an approved estimator matched to the declared `targetEstimand`, or demotes the claim to a no-generalisation / finite-corpus claim, or returns INCONCLUSIVE for population-level VERIFICATION (§8.2.1, §8.4.6).
- A criterion's denominator is **the full sampling** N (§1.4.5a): a trial is a success only on a clean PASS; a transform/no-value failure is a counted FAIL (carrying that reason), not an exclusion. There is no per-criterion denominator policy and no `CONDITIONAL/MARGINAL` enum; no input is filtered out, since every sample is seen by every criterion.

- Availability, when wanted as a distinct metric, is its own criterion; it does not gate or alter another criterion's denominator (§1.4.5a, §10.2).
- The conformance metadata distinguishes formula-value-fixture status from calibration-fixture status, and does not claim statistical calibration conformance without calibration-fixture agreement (§10.6).
- The transparent-statistics output carries the reproducibility metadata of §10.2 and the numerical-conventions metadata of §12.8.

---

11. Statistical Discipline Through Design

Sections 2–10 developed the statistical machinery in isolation. This short section names the **framework-level disciplines** that bind the machinery into a practice, so that developers who are not statisticians can still produce statistically defensible verdicts.

The disciplines are orthogonal design policies, not theorems:

Discipline	What it surfaces	Where it lives
Explicit thresholds and provenance	Origin of every pass/fail decision	§7.4 threshold provenance, transparent-statistics output
Feasibility gating for affirmative claims	Sample-size adequacy before the test runs	§5.7 VERIFICATION vs SMOKE
Covariate tracking	Non-stationarity across baseline vs. test contexts	§8.4.1
Baseline expiration	Staleness of empirical baselines	§8.4.2
Warnings over silence	Imperfect conditions, not pretending otherwise	throughout
Full audit trail	Reproducibility and post-hoc investigation	§7.4, transparent-statistics mode

Two points are worth naming plainly.

First, these disciplines do not repair violated assumptions. They make assumption drift visible. A test that runs under a stale baseline still produces a statistically questionable verdict — the framework just refuses to let that fact be silent.

Second, ”more samples” is not a substitute for these disciplines. Sample size controls precision and power; it does not create a principled threshold, verify stationarity, or produce an audit trail. The machinery of §§1–10 and the disciplines of §11 are complements, not alternatives.

---

12. Latency: Empirical Percentile Analysis

> **Epistemic status of this section.** The percentile estimator (§12.2.2) and the binomial order-statistic upper bound on a baseline quantile (§12.4.2) are **exact distribution-free results** for i.i.d. samples from a continuous latency distribution. The threshold interpretation is a **confidence bound on the true baseline quantile**, not a predictive interval for a future test experiment's observed percentile — the two are not the same, and conflating them is the single most likely mistake a reader will make with this section. §12.4.3 makes the caveat precise; skip there directly if you only read one sub-section. The feasibility gate (§12.5.3) and the enforcement-mode split (§12.6) are **design policies**, not theorems.

12.1 The Statistical Challenge of Latency

Pass-rate testing models functional outcomes as Bernoulli trials drawn from a binomial distribution (Section 1). Latency presents a fundamentally different statistical challenge. Service latency distributions are typically:

- **Right-skewed**: A long right tail caused by cache misses, garbage collection pauses, network retransmission, or cold starts
- **Multimodal**: Distinct modes corresponding to fast paths (cached) and slow paths (database lookup, remote API call)
- **Heavy-tailed**: Extreme outliers that are orders of magnitude larger than the median

These characteristics violate the assumptions of parametric models such as the normal distribution. A system with a 200ms mean and 50ms standard deviation could have a p99 of 350ms (near-normal) or 2000ms (heavy-tailed) — the summary statistics cannot distinguish the two.

The mavai approach: Rather than fitting a parametric distribution to latency data, the methodology uses **non-parametric empirical percentiles** exclusively. This distribution-free approach makes no assumptions about the shape of the latency distribution — it characterises the distribution directly from the observed order statistics.

12.2 Empirical Percentile Estimation

12.2.1 Population Definition and the Tripartite Contract A naive single-population view of latency is statistically hazardous: mixing fast failures, slow failures, and successful responses into one distribution describes none of them faithfully. The mavai methodology therefore decomposes the service-level contract into three orthogonal sub-contracts, each with its own estimand and its own inferential machinery:

Sub-contract	Estimand	Statistical treatment
Correctness	$P(\text{semantic success})$	Binomial (§§1–5)

Sub-contract	Estimand	Statistical treatment
Availability	$P(\text{infrastructure success})$	Binomial (same machinery; currently treated jointly with correctness in the pass-rate dimension)
Latency-given-success	$T \mid X = 1$	Non-parametric empirical percentiles (§12.2.2)

The latency dimension characterises **only the third sub-contract**: the distribution of wall-clock execution time conditional on a successful response. Formally:

$$\mathcal{L} = \{t_i : X_i = 1\}, \qquad T \mid X = 1$$

where t_i is the execution time of the i-th trial and $X_i \in \{0, 1\}$ is its Bernoulli outcome. Let $n_s = |\mathcal{L}|$ denote the number of successful samples and let $t_{(1)} \leq \cdots \leq t_{(n_s)}$ be their order statistics.

Rationale for conditioning: Failed samples produce execution times that are not comparable with successful response times. A fast failure (immediate validation rejection, $t \approx 0$) and a slow failure (timeout at $t = 30{,}000$ms) both reflect error paths, not the latency of successful operation. Pooling them would produce percentile estimates that describe neither the successful nor the failed population.

The perverse-incentive hazard: Because the latency contract is conditional on success, a service could in principle "improve" its observed $Q_{0.95}$ by converting slow-successes into fast-timeouts — moving mass from the latency distribution into the failure column. This is not a defect of the conditioning; it is the reason the three sub-contracts are evaluated **jointly with logical conjunction** (§12.3.2). A service cannot trade correctness or availability for latency under the mavai verdict rule, because every sub-contract must pass independently. Reviewers and auditors should satisfy themselves that the correctness and availability thresholds are tight enough that this trade cannot be exploited silently.

What this is not: The latency distribution treated here is not the user-experienced response-time distribution marginalised over all attempts. A user who receives an error sees no latency value. Organisations that need an unconditional response-time SLA should combine the three sub-contracts explicitly, e.g. by asserting availability at a level sufficient to bound the marginal tail.

12.2.2 Nearest-Rank Interpolation The mavai methodology computes percentiles using the **nearest-rank method**. For a percentile $p \in (0, 1]$ with n_s sorted observations:

$$Q(p) = t_{(\lceil p \cdot n_s \rceil)}$$

where $t_{(k)}$ denotes the k-th order statistic of the sorted sample (one-based), $t_{(1)} \leq t_{(2)} \leq \cdots \leq t_{(n_s)}$. The rank $\lceil p \cdot n_s \rceil$ is clamped to $[1, n_s]$.

Worked example: For $n_s = 200$ successful samples:

Percentile	p	Rank $\lceil p \cdot 200 \rceil$	Order statistic
p50	0.50	100	$t_{(100)}$
p90	0.90	180	$t_{(180)}$
p95	0.95	190	$t_{(190)}$
p99	0.99	198	$t_{(198)}$

Why nearest-rank? Linear interpolation methods (e.g., Type 7 in R) produce fractional percentile estimates. For latency thresholds expressed in integer milliseconds and compared against SLA values, the discrete nearest-rank method produces integer-valued estimates that align naturally with how thresholds are specified and reported.

12.2.3 Summary Statistics In addition to percentiles, the framework reports:

- **Mean**: $\bar{t} = \frac{1}{n_s} \sum_{i=1}^{n_s} t_i$
- **Maximum**: $t_{(n_s)}$

The mean and the percentiles (including the maximum, which is $Q(1.0)$) characterise the distribution shape for reporting purposes. Threshold derivation (§12.4) is non-parametric and operates directly on the order statistics — it does not require a sample standard deviation, and the framework deliberately omits s from the baseline payload to prevent misuse as a summary scale of a distribution that is not well-characterised by its second moment.

12.3 Latency Assertions

12.3.1 The Assertion Model A latency assertion specifies a set of percentile constraints:

$$\mathcal{C} = \{(p_j, \tau_j) : j = 1, \ldots, m\}$$

where p_j is a percentile level (one of $\{0.50, 0.90, 0.95, 0.99\}$) and τ_j is the corresponding threshold in milliseconds.

For each constraint, the assertion evaluates:

$$\text{PASS}_j \iff Q(p_j) \leq \tau_j$$

The overall latency assertion passes if and only if all individual constraints pass:

$$\text{PASS}_{\text{latency}} = \bigwedge_{j=1}^{m} \text{PASS}_j$$

12.3.2 Combined Verdict Pass-rate and latency are **distinct** quality dimensions (orthogonal as concerns, not necessarily statistically independent — see §12.7). The overall test verdict requires both to pass:

$$\text{PASS}_{\text{test}} = \text{PASS}_{\text{rate}} \wedge \text{PASS}_{\text{latency}}$$

This reflects the operational reality that a service must be both correct and responsive. A payment API that succeeds 99.5% of the time but takes 30 seconds for 1% of requests fails its SLA just as surely as one that returns incorrect results.

12.3.3 Threshold Sources Latency thresholds can originate from two sources:

Source	Symbol	How specified	Statistical question
Explicit	τ_j (given)	Annotation or configuration	”Does the system meet the declared latency target?”
Baseline-derived	$\hat{\tau}_j$ (estimated)	Automatic from spec	”Has latency degraded from the measured baseline?”

This mirrors the compliance vs. regression dichotomy for pass-rate thresholds (Section 3). Explicit thresholds are normative claims; baseline-derived thresholds are empirical estimates with a statistical margin.

12.4 Threshold Derivation from Baselines

12.4.1 The Problem When a measure experiment records the latency distribution, the observed percentiles are point estimates subject to sampling variability. Using the raw baseline percentile as the threshold would cause frequent false positives — the same problem as using $\hat{p}_{\text{baseline}}$ directly as the pass-rate threshold (Section 3.1).

Example: A baseline observes $Q_{0.95} = 480$ms from $n_s = 935$ samples. A subsequent test with $n_s = 192$ samples observes $Q_{0.95} = 495$ms. Is this a degradation, or normal variance?

The latency dimension is the non-parametric analogue of the Wilson-score construction used for pass rates (§3.4). To preserve statistical symmetry between the two sides of the contract, the threshold must be derived from a construction that is:

1. **Exact or near-exact** under the minimal assumptions stated in §12.2.1 (i.i.d. successful latencies).
2. **Distribution-free** — free of any parametric shape or density assumption.

3. **Integer-millisecond** by construction, aligning with how SLA thresholds are expressed and compared.

A construction satisfying all three falls out naturally from the binomial sampling distribution of order-statistic ranks, developed below.

12.4.2 Binomial Order-Statistic Upper Bound For i.i.d. latency samples $T_1, \dots, T_{n_s}$ with continuous distribution, the rank of the population quantile $Q(p_j)$ within the sorted sample follows a binomial sampling distribution. Specifically, the number of observations $\leq Q(p_j)$ is distributed as $\text{Bin}(n_s, p_j)$. This fact — standard in any non-parametric textbook (Conover, 1999; Hollander & Wolfe, 1999) — yields an **exact distribution-free upper confidence bound** on the true quantile:

$$\tau_j = t_{(k_j)} \quad \text{where} \quad k_j = \min\{k \in \{1, \dots, n_s\} : P(B \geq k) \leq \alpha\}, \quad B \sim \text{Bin}(n_s, p_j)$$

Equivalently, k_j is the smallest rank such that the probability of seeing k or more observations at or below the true $Q(p_j)$ is at most α. The rank is computed in two steps. The raw rank is

$$k_{\text{raw}} = \texttt{qbinom}(1 - \alpha,\, n_s,\, p_j) + 1,$$

and the saturation gate is applied before any clamp:

- if $k_{\text{raw}} \leq n_s$, the bound exists within the sample. The operative rank is $k_j = \max(k_{\text{raw}}, \lceil p_j n_s \rceil)$ and the upper confidence bound is $\tau_j = t_{(k_j)}$. (The floor at the baseline percentile rank prevents the bound from being looser-than-loose — see the property list below.)
- if $k_{\text{raw}} > n_s$, no finite-sample distribution-free upper confidence bound on $Q(p_j)$ is available from this sample at the configured confidence. The rank has saturated beyond the maximum observed order statistic; the construction's existence condition (§12.5.2.1) is not met. Under VERIFICATION the result is INCONCLUSIVE, treated as a configuration error per §8.4.5. Under SMOKE or advisory reporting, the value $t_{(n_s)}$ may be reported with `saturated: true`, but does not constitute an exact bound and is not labelled as such.

Why this is the right construction. The upper confidence bound on $Q(p_j)$ at level $1 - \alpha$ is defined as the smallest value τ such that $P(\hat{Q}(p_j) > \tau \mid Q_{\text{true}}(p_j) \leq \tau) \leq \alpha$ under the null of no degradation. Because ranks are binomially distributed regardless of the underlying latency distribution, the construction is exact for any continuous F_T — no density estimate, no normal approximation, no second moment. It is the non-parametric counterpart of the Wilson bound used on the pass-rate side, and restores the statistical symmetry the mavai methodology requires between the two halves of the contract.

Properties:

- **Integer-ms by construction**: τ_j is an observed latency. No rounding, no ceiling, no artefacts.
- **Monotone in** α: higher confidence gives a higher rank and hence a looser (more conservative) threshold.
- **Monotone in** p_j: higher percentiles yield higher ranks.
- **Floor at the baseline percentile**: $k_j \geq \lceil p_j \cdot n_s \rceil$, so $\tau_j \geq Q_{\text{baseline}}(p_j)$ always. No separate max guard is needed; it falls out of the construction.
- **Existence gating at small** n_s: when n_s is too small to resolve p_j at confidence $1-\alpha$, the raw rank exceeds n_s. The construction does not silently clamp the rank to n_s to manufacture a bound; the existence condition (§12.5.2.1) fails and the verdict is INCONCLUSIVE under VERIFICATION, advisory under SMOKE. The methodology handles small-n_s exclusively through the feasibility gate (§12.5.3) and the existence gate (§12.5.2.1).

Continuity and ties. The exactness argument above assumes a continuous latency distribution — under continuity, ties occur with probability zero and every rank has a well-defined population interpretation. In practice, wall-clock latencies are reported in integer milliseconds, which induces ties. For the purposes of the upper-bound construction this does not matter: with tied values, the rank of the true quantile remains distributed as at most $\text{Bin}(n_s, p_j)$ (ties can only shift rank downward), so $\tau_j = t_{(k_j)}$ remains a valid upper confidence bound. It is no longer tight — the bound becomes **conservative**, not anti-conservative. The framework accepts this mild conservatism in exchange for the engineering benefits of integer-ms thresholds; practitioners who care about the tightness gap should report latencies at higher resolution (microseconds) before applying the construction.

12.4.3 Statistical Interpretation A test with observed $\hat{Q}_{\text{test}}(p_j) \leq \tau_j$ means: the observed percentile is consistent with a true quantile no worse than the baseline, at confidence $1 - \alpha$.

A breach ($\hat{Q}_{\text{test}}(p_j) > \tau_j$) means: the observed percentile exceeds the one-sided binomial upper bound on the baseline quantile, providing evidence of latency degradation at the stated confidence level.

Note on confidence vs. prediction: The construction above is a confidence bound on the true baseline quantile $Q_{\text{true}}(p_j)$, not a prediction interval for the next experiment's $\hat{Q}_{\text{test}}(p_j)$. When baseline and test sample sizes are comparable, test-side sampling variance can materially increase the no-degradation breach probability above the nominal α. The increase is not controlled by the baseline confidence-bound construction alone; it depends on the test sample size, percentile level, and rank convention. Breaches remain statistically meaningful (they exceed a legitimate upper bound on the baseline), but the actual false-degradation-alarm rate is not bounded by a simple multiple of α. Operators who require calibrated false-degradation-alarm rates should use a predictive or two-sample procedure, or rely on calibration fixtures, rather than reading the baseline confidence bound as a predictive test threshold; sizing test experiments substantially larger than baselines reduces but does not eliminate the gap (treatment forthcoming in 1.4 — see Forward Scope).

12.4.4 Supporting Comparison: Bootstrap **Epistemic status**: illustration, not validation. The binomial order-statistic construction stands on the theorem cited in §12.4.2; no empirical resemblance is needed to establish its correctness. The comparison below is included only to give engineering readers a familiar reference point and to show that on realistic heavy-tailed distributions the exact construction and a resampling estimator do not disagree in pathological ways. A reader should not read bootstrap agreement as confirming the theorem; bootstrap itself is an asymptotic method with known downward bias for heavy-tail quantiles.

`scripts/bootstrap_compare.R` in the mavai-R repository computes the 95% one-sided upper bound on $Q_{0.95}$ and $Q_{0.99}$ using (i) a 10,000-replicate percentile bootstrap (type-1 quantile) and (ii) the exact binomial order-statistic construction defined in §12.4.2. Reference baselines are lognormal draws: $n_s = 200$ at ($\mu = \log 200$, $\sigma = 0.4$) and $n_s = 935$ at ($\mu = \log 500$, $\sigma = 0.3$), seeded for reproducibility.

Sample	n_s	p	Point estimate $Q(p)$	Bootstrap 95% upper	Binomial bound (rank)	Δ (ms)
lognormal	200	0.95	356	393	419 (k=196)	+26
lognormal	200	0.99	448	589	589 (`k_raw > n_s`; advisory $t_{(n_s)}$, `saturated: true`; not an exact bound)	0
lognormal	935	0.95	787	810	812 (k=900)	+2
lognormal	935	0.99	980	1098	1125 (k=931)	+27

Two observations:

1. **The binomial bound is uniformly no less conservative than the bootstrap**, as expected for an exact finite-sample construction compared to a resampling estimator that is itself subject to Monte Carlo variance and (for quantiles of heavy-tailed distributions) known downward bias.
2. **Agreement is within a handful of order-statistic steps** in every row, and exact in the $n_s = 200$, $p = 0.99$ case where the bound saturates at the maximum — a signal the feasibility gate (§12.5.3) should have been invoked (that row exists to demonstrate graceful saturation, not as a recommended configuration).

Exact numerical outputs and the bootstrap seeds are preserved in `inst/cases/latency_threshold_bootstrap.json` so downstream consumers can verify the comparison without an R installation.

A standard-error scaling such as $s/\sqrt{n_s}$ would understate tail-percentile uncertainty on heavy-tailed distributions by a factor that grows with skewness; the binomial order-statistic construction has no such defect.

12.4.5 Worked Example **Baseline**: $n_s = 935$ successful samples, $Q_{0.95} = 580$ms, confidence $= 0.95$ (so $\alpha = 0.05$).

The rank of the upper bound is:

$$k_{0.95} = \texttt{qbinom}(0.95,\ 935,\ 0.95) + 1$$

$B \sim \text{Bin}(935, 0.95)$ has mean 888.25 and standard deviation $\sqrt{935 \cdot 0.95 \cdot 0.05} \approx 6.66$. `qbinom(0.95, 935, 0.95)` returns 899 — that is, $P(B \leq 899) \approx 0.959$, the smallest value for which the cumulative probability reaches 0.95. Therefore $k_{0.95} = 900$, which satisfies $P(B \geq 900) \approx 0.041 \leq 0.05$.

The baseline rank for the point estimate is $\lceil 0.95 \cdot 935 \rceil = 889$, so the bound sits 11 ranks above the point estimate.

$$\tau_{0.95} = t_{(900)}$$

That is, the latency threshold is the 900th-smallest observation in the baseline — an observed value in milliseconds, by construction.

A subsequent test with $\hat{Q}_{0.95,\text{test}} \leq t_{(900)}$ passes; any observation above $t_{(900)}$ breaches the threshold and constitutes evidence of degradation at 95% confidence.

12.5 Sample Size Requirements for Percentile Estimation

12.5.1 The Problem An empirical percentile $Q(p)$ is computed from the $\lceil p \cdot n_s \rceil$-th order statistic. When n_s is small relative to p, the estimate is unreliable:

- For $p = 0.99$ with $n_s = 10$: $\lceil 0.99 \times 10 \rceil = 10$ — the "99th percentile" is simply the maximum value. A single outlier determines the result.
- For $p = 0.99$ with $n_s = 50$: $\lceil 0.99 \times 50 \rceil = 50$ — still the maximum. The p99 only becomes distinct from the maximum when $n_s \geq 100$.

12.5.2 Minimum Sample Sizes The methodology enforces minimum sample counts for each percentile level based on the requirement that the percentile estimate be based on at least one observation below it in the sorted order:

Percentile	p	Minimum n_s	Rationale
p50	0.50	5	$\lceil 0.50 \cdot 5 \rceil = 3$: 3rd order statistic of 5; two values below, two above
p90	0.90	10	$\lceil 0.90 \cdot 10 \rceil = 9$: 9th order statistic of 10; one value above
p95	0.95	20	$\lceil 0.95 \cdot 20 \rceil = 19$: 19th order statistic of 20; one value above
p99	0.99	100	$\lceil 0.99 \cdot 100 \rceil = 99$: 99th order statistic of 100; one value above. Below 100, p99 = max

These thresholds ensure that the percentile estimate is not degenerate (i.e., not simply the minimum or maximum of the sample). They are a **non-degeneracy gate** only — they do not by themselves guarantee that a finite-sample distribution-free upper confidence bound on the true quantile exists at the configured confidence. That second condition is given by the confidence-bound existence gate below (§12.5.2.1).

12.5.2.1 Confidence-Bound Existence Gate A non-degenerate empirical percentile is necessary but **not sufficient** for the binomial order-statistic upper bound (§12.4.2) to be informative. For a one-sided distribution-free upper confidence bound on the p-quantile to exist within the observed sample without saturating beyond the maximum, a necessary condition is

$$p^{n_s} \leq \alpha,$$

equivalently

$$n_s \geq \left\lceil \frac{\log(\alpha)}{\log(p)} \right\rceil.$$

This is the standard Wilks (1941) tolerance-interval logic for distribution-free upper bounds on quantiles via order statistics. At $\alpha = 0.05$:

Percentile	p	Minimum n_s for non-saturated 95% upper bound
p50	0.50	5
p90	0.90	29
p95	0.95	59
p99	0.99	299
p99.9	0.999	2995

If the rank required by §12.4.2 saturates beyond n_s — i.e. $k_{\text{raw}} > n_s$ — **no finite-sample distribution-free upper confidence bound on $Q(p_j)$ is available at the configured confidence from this sample size**. The §12.4.2 construction therefore does not clamp k_{raw} to n_s and present the resulting $t_{(n_s)}$ as an exact bound: under VERIFICATION the verdict is INCONCLUSIVE (configuration-error treatment per §8.4.5); under SMOKE or advisory reporting, the value $t_{(n_s)}$ may be displayed with `saturated: true` as a clearly-labelled best-available statistic, distinct from an exact bound. The existence gate is the operative condition; the displayed advisory value does not weaken it.

The two latency gates therefore play different roles:

Gate	Question answered	Failure mode in VERIFICATION
Non-degeneracy (§12.5.2)	Is the empirical percentile distinct from the sample maximum / minimum?	Configuration error
Confidence-bound existence (§12.5.2.1)	Does the configured confidence procedure admit a non-saturated upper bound on the true quantile?	Configuration error / INCONCLUSIVE (saturation report applies under SMOKE or advisory only)

A percentile estimate can be non-degenerate yet still unable to support a distribution-free upper confidence bound at the configured confidence. For each asserted percentile, the framework first checks the non-degeneracy requirement and then checks whether the required order-statistic confidence-bound rank exists within the observed sample. If the required rank exceeds the sample size, the latency assertion is infeasible for VERIFICATION at the configured confidence.

Scope note on p99.9 and beyond: The supported percentile levels are $\{0.50, 0.90, 0.95, 0.99\}$. Extreme-tail percentiles such as p99.9 are out of scope for the current methodology: a non-degenerate p99.9 estimate requires $n_s \geq 1{,}000$ successful samples, and a statistically useful binomial order-statistic upper bound at 95% confidence requires considerably more. Services with genuine p99.9 SLAs generally warrant dedicated tail-focused instrumentation (production telemetry, HdrHistogram-style log-linear bucketing, or extreme-value modelling) rather than per-test-run estimation.

12.5.3 The Feasibility Gate For **VERIFICATION** intent with latency enforcement enabled, the framework checks before any samples execute whether the expected number of successful samples meets the **stricter** of the two minimums — non-degeneracy (§12.5.2) and confidence-bound existence (§12.5.2.1):

$$n_{s,\text{min}}^{\text{VERIFICATION}}(p_j,\, \alpha) \;=\; \max\left(n_{s,\text{min}}^{\text{non-degen}}(p_j),\; \left\lceil \frac{\log \alpha}{\log p_j} \right\rceil \right).$$

The expected successful-sample count is

$$n_{s,\text{expected}} = n_{\text{planned}} \times \hat{p}_{\text{baseline}}.$$

If $n_{s,\text{expected}} < n_{s,\text{min}}^{\text{VERIFICATION}}(p_j,\, \alpha)$ for any asserted percentile p_j, the framework raises a configuration error — the same mechanism used for the pass-rate feasibility gate (Section 5.7.1).

Example: A test with $n_{\text{planned}} = 200$ and baseline $\hat{p} = 0.80$ yields $n_{s,\text{expected}} = 160$. A p99 assertion at $\alpha = 0.05$ requires $n_{s,\text{min}}^{\text{VERIFICATION}} = \max(100,\, 299) = 299$. The test is infeasible (160 < 299) and fails immediately with a diagnostic message.

12.5.4 Indicative Results When sample size falls below the minimum but the test is not subject to the feasibility gate (SMOKE intent, or advisory mode), the framework still evaluates the percentile but marks the result as **indicative**:

> "The p99 result is based on $n_s = 40$ samples (minimum recommended: 100). This result is indicative — a directional signal, not a statistically reliable estimate."

This mirrors the SMOKE/VERIFICATION asymmetry for pass-rate testing (Section 5.7.3). The percentile is computed and reported, but its epistemic weight is explicitly qualified.

12.6 Enforcement Modes

Latency assertions support two enforcement modes, reflecting the practical reality that latency profiles are environment-dependent:

Mode	Breach behaviour	Default	Rationale
Advisory	Warning in output; test passes	Yes	Baseline may have been recorded on different hardware
Enforced	Test fails	No	Latency is a first-class SLA dimension

Why advisory is the default: A baseline generated on CI hardware with dedicated resources may record $Q_{0.95} = 480$ms. The same workload on a developer laptop with competing processes may observe $Q_{0.95} = 650$ms — a genuine environmental difference, not a regression. Failing the test by default would produce false positives that erode trust in the framework.

When enforcement is enabled, latency breaches fail the test, and the VERIFICATION feasibility gate is active. This is appropriate for environments where hardware consistency is controlled (dedicated CI, staging environments).

12.7 Relationship to Pass-Rate Testing

The table below summarises how the two quality dimensions parallel each other in statistical treatment:

Aspect	Pass Rate	Latency
Statistical model	Parametric (binomial)	Non-parametric (empirical percentiles)
Estimand	Success probability p	Percentile quantiles $Q(p_j)$
Threshold derivation	Wilson score lower bound	Binomial order-statistic upper bound
Baseline storage	$(\hat{p}, k, n)$	$(t_{(1)}, \dots, t_{(n_s)}, n_s)$

Aspect	Pass Rate	Latency
Feasibility gate	$N_{\min}$ from Wilson bound	$n_{s,\min}$ from percentile reliability
Indicative marking	Undersized sample note	Undersized sample note
Enforcement	Always enforced	Advisory by default; opt-in enforcement

The two dimensions are evaluated **separately** and combined with logical conjunction. They are distinct (orthogonal as quality concerns), not necessarily statistically independent: correctness and latency may covary in practice — difficult prompts can be slower and more likely to fail, infrastructure overload can lift both error and tail-latency rates. **The methodology does not require functional and temporal stochasticity to be statistically independent; the combined verdict is a logical conjunction rather than a probabilistic independence model.** What separation guarantees operationally is that latency analysis cannot compensate for a pass-rate failure, and vice versa — each dimension must meet its own threshold.

12.8 Numerical Conventions

The order-statistic constructions of §§12.2 and 12.4 commit to a small set of numerical conventions. They are part of the methodology rather than implementation choices: a downstream implementation that diverges from them produces values that disagree with the mavai-R fixtures.

Sort order. Observed successful latencies are sorted in **ascending** order before any rank lookup. The order statistic $t_{(k)}$ is the k-th smallest value (1-indexed).

Nearest-rank percentile. Per §12.2.2, the empirical percentile is $Q(p_j) = t_{(\lceil p_j \cdot n_s \rceil)}$. This is the same definition assumed by the binomial order-statistic upper bound of §12.4.2; the methodology uses this nearest-rank estimator rather than an interpolating quantile estimator (e.g. R's Type 7), and the conformance fixtures encode it. The ceiling is taken on the raw product $p_j \cdot n_s$ without intermediate rounding.

Tie policy at integer-millisecond resolution. Wall-clock latencies reported in integer milliseconds induce ties. When a rank lookup $t_{(k)}$ falls inside a run of equal-valued observations, the **largest tied position** is used for upper-bound rank lookups — the threshold derivation of §12.4.2 and any operator-side query for an upper bound on $Q(p_j)$. This is the **conservative** tie convention: ties at the boundary of a confidence-bound rank do not let the bound collapse below the value the tied observations actually exhibit, so the bound is not anti-conservatively tight. Point-estimate percentile lookups (§12.2.2) use the rank as written; the conservative tie rule applies specifically to the upper-bound rank k_j of §12.4.2 and to any other order-statistic lookup whose role is to bound a quantile from above.

Displayed-threshold rounding. When a derived latency threshold is rendered for human consumption it is rounded **half-to-even** (banker's rounding) at the displayed precision — typically integer milliseconds for latency. The **raw** values — order statistics, ranks, and any unrounded intermediate quantity — are retained unrounded in

the trial record and the transparent-statistics output (§7.1, §10.2), so a downstream consumer can recompute the displayed value from the raw inputs and verify against the mavai-R fixtures without depending on the renderer's precision.

Sort stability. Where the implementation language offers a choice of sort algorithms, the methodology assumes a **stable** sort. Stability has no statistical effect on the order-statistic values, but it makes the trial-record-to-rank mapping reproducible across runs when ties are present, which simplifies post-hoc audit.

The transparent-statistics report (§7.1, §10.2) names the actual sort algorithm and tie convention used at runtime; see the `sortStable` and `tiePolicy` fields specified there.

Appendix A. Elements of the statistical model

The statistical model described in this companion is built from a set of named primitives, parameterised by named statistical quantities, and produces a set of named statistical results. This appendix gathers them in one place, with references to the sections in which each is defined.

The list is restricted to elements that are intrinsic to the model. Operationalization artefacts that frameworks and projects assemble around the model — the service contract as a programming artefact, contract families, binding policies, runtime-resolution mechanisms — are documented elsewhere.

Element	Information content	Defined in
Postcondition	A predicate over the service's output; defines per-trial pass or fail for a single observable property.	§1.4.2
Criterion	The partition unit of the functional dimension. References one or more postconditions, has a mode (inferential or observational), and where inferential carries a threshold p_c^* and confidence level α_c. Its denominator is the full sampling N (§1.4.5a).	§1.4.2, §1.4.3, §1.4.5, §1.4.5a
Sampling	A list of $N \geq 1$ samples posted to the service under test in a single experiment. Shared by every criterion of the experiment; per-criterion verdicts are claims about the same N samples.	§1.4.2, §1.4.7
Factor record	The identification of the service, model, and serving configuration whose p_c is being estimated. Two evaluations that differ in factors evaluate two different objects.	§1.3.1

Element	Information content	Defined in
Covariate profile	The recorded values of declared contextual variables at the time of an evaluation; affects baseline comparability.	§8.4.1
Inferential reach of the sampling	The N samples the experiment posts to the service, over which every criterion in the experiment is exercised; a criterion's verdict or evidence is, primarily, a claim about those samples. Extending it to a different input distribution is a separate interpretive move (§8.4.6).	§1.4.7
Per-criterion Bernoulli stream	The sequence of per-criterion indicators $\{X_{i,c}\}$ treated as i.i.d. Bernoulli with parameter p_c under the model's working approximation.	§1.4.3
Effective denominator (n_c)	The sampling size N: every sample is seen by every criterion. A trial is a success only on a clean PASS; every other outcome — including a transform/no-value failure — is a counted FAIL, never excluded. `zeroFailures` (observational) criteria carry no rate denominator.	§1.4.5a
Confidence statement	A Wilson lower bound $\hat{p}_{c,L}(\alpha_c)$, qualifying an inferential per-criterion claim about p_c.	§2.3.1, §1.4.3
Threshold origin	The provenance category of an inferential threshold p_c^* (SLA, SLO, POLICY, EMPIRICAL, UNSPECIFIED), recorded with the threshold value.	§7.4
Sample-size requirement	The per-criterion sample count required to support an inferential test at its threshold and α_c, with the feasibility gate that admits or refuses a smaller sample.	§§5.4–5.5, §8.4

Element	Information content	Defined in
Per-criterion verdict	PASS, FAIL, or INCONCLUSIVE on a criterion: for **compliance** criteria, the one-sided Wilson lower bound's relation to p_{req}; for **regression** criteria, the observed success count's relation to the integer cutoff c_c derived from the reference distribution at α_c; for **observational** criteria, the zero-failure observation. Carries the supporting statistics, the threshold and origin, α_c, and the population specification.	§1.4.3, §1.4.5, §1.4.6
Composite verdict	A structured tuple over per-criterion verdicts.	§1.4.6
Composite Type-I envelopes	Procedure-direction-specific union-bound aggregates over inferential criteria: the **false-degradation-signal envelope** $\alpha_{\text{fds}} \leq \sum_{c \in \text{regression}} \alpha_c$ and the **false-compliance envelope** $\alpha_{\text{fc}} \leq \sum_{c \in \text{compliance}} \alpha_c$. A contract reports each envelope only when it carries criteria of that direction; mixed contracts report both. Observational criteria contribute to neither.	§1.4.6
Baseline	A family of per-criterion point estimators $\{\hat{p}_c\}$ with supporting $\{n_c\}$ and $\{K_c\}$, qualified by the factor record, covariate profile, structural reference, and optional expiration window under which it was measured. Consumed by inferential criteria of origin EMPIRICAL to derive p_c^* at resolution time.	§1.5

References

1. Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. Journal of the American Statistical Association, 22(158), 209-212.
2. Agresti, A., & Coull, B. A. (1998). Approximate is better than ”exact” for interval estimation of binomial proportions. The American Statistician, 52(2), 119-126.

3. Brown, L. D., Cai, T. T., & DasGupta, A. (2001). Interval estimation for a binomial proportion. Statistical Science, 16(2), 101-133.

4. Newcombe, R. G. (1998). Two-sided confidence intervals for the single proportion: comparison of seven methods. Statistics in Medicine, 17(8), 857-872.

5. Hanley, J. A., & Lippman-Hand, A. (1983). If nothing goes wrong, is everything all right? Interpreting zero numerators. JAMA, 249(13), 1743-1745. [The "Rule of Three"]

6. Clopper, C. J., & Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika, 26(4), 404-413.

7. Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). Bayesian Data Analysis (3rd ed.). Chapman and Hall/CRC. [Beta-Binomial posterior predictive]

8. Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. Proceedings of the Royal Society of London. Series A, 186(1007), 453-461. [Jeffreys prior]

9. Hyndman, R. J., & Fan, Y. (1996). Sample quantiles in statistical packages. The American Statistician, 50(4), 361-365. [Percentile interpolation methods]

10. David, H. A., & Nagaraja, H. N. (2003). Order Statistics (3rd ed.). Wiley-Interscience. [Order statistics and empirical percentiles]

11. Conover, W. J. (1999). Practical Nonparametric Statistics (3rd ed.). Wiley. [Nonparametric quantile confidence intervals via order statistics; the binomial construction of §12.4.2]

12. Hollander, M., & Wolfe, D. A. (1999). Nonparametric Statistical Methods (2nd ed.). Wiley. [Parallel reference for the order-statistic upper bound]

13. Wilks, S. S. (1941). Determination of sample sizes for setting tolerance limits. Annals of Mathematical Statistics, 12(1), 91-96. [Foundational treatment of distribution-free tolerance intervals]

14. Krishnamoorthy, K., & Mathew, T. (2009). Statistical Tolerance Regions: Theory, Applications, and Computation. Wiley. [Modern treatment of distribution-free tolerance and quantile bounds]

15. Bayarri, M. J., & Berger, J. O. (2004). The interplay of Bayesian and frequentist analysis. Statistical Science, 19(1), 58-80. [Framework for reconciling predictive-Bayesian and frequentist inference, cited in §4.5]

16. Geisser, S. (1993). Predictive Inference: An Introduction. Chapman and Hall. [Predictive treatment of binomial future performance]

17. Anthropic (2026). Models and Model deprecations. Claude documentation, accessed 2026-05-11. https://platform.claude.com/docs/en/docs/about-claude/models/overview ; https://platform.claude.com/docs/en/docs/about-claude/model-deprecations [Provider commitment that every Claude model ID is a pinned snapshot for the lifetime of its deprecation window; cited in §1.3.1 for the snapshot-vs-floating-alias distinction.]

18. Chen, L., Zaharia, M., & Zou, J. (2023). How is ChatGPT's behavior changing over time? arXiv preprint arXiv:2307.09009. [Empirical demonstration of behaviour drift in floating-alias model identifiers across snapshots over a multi-month period; cited in §1.3.1 as motivation for the pinned-snapshot prescription.]

19. Liang, K.-Y., & Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. Biometrika, 73(1), 13–22. [Generalised estimating equations; cited in §8.2 / §8.2.1 for clustered-design analysis under non-trivial dependence structures.]

20. Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society: Series B, 57(1), 289–300. [Original BH procedure; cited in §7.3 for FDR control under independence and PRDS.]

21. Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. Annals of Statistics, 29(4), 1165–1188. [BY procedure; cited in §7.3 for FDR control under arbitrary dependence.]

22. Anthropic. Enterprise deployment overview. Claude documentation, accessed 2026-05-14. [Provider documentation describing endpoint and routing configurations that vary across platforms; cited in §1.3.1 for the necessary-but-insufficient status of pinned model IDs.]

23. Garces Arias, E., Blocher, H., Rodemann, J., Aßenmacher, M., & Jansen, C. (2025). Statistical multicriteria evaluation of LLM-generated text. In Proceedings of the 18th International Natural Language Generation Conference (INLG), 338–351. Association for Computational Linguistics. [Independent diagnosis of the multi-criterion evaluation problem for stochastic language systems, addressed via a Generalized Stochastic Dominance front rather than the per-criterion decomposition adopted here; cited in the Introduction.]

This document is intended for review by professional statisticians. For operational guidance, see the documentation in your framework of choice: punit, feotest. For the reference implementation of all statistical computations described here, see mavai-R.