INDEPENDENT EVALUATION

GSA CALC+ API Testing Record

Federal acquisition skill validated across eight runs on two Claude models; Round 2 patches shipped and validated

Skill
gsa-calc-ceilingrates (post-Round 2 patches; 112/112 assertions passed)
github.com/1102tools/federal-contracting-skills

Date
April 2026
1102tools.com

Part 1: For Federal Acquisition Users

The bottom line

Independent testing in April 2026 (8 end-to-end runs, 112 binary assertions graded, 2 Claude models) shows the GSA CALC+ skill reliably pulls MAS ceiling rate distributions across four real-world federal acquisition workflows. Wave 1 testing surfaced a critical silent-wrong-answer bug in the skill's response schema documentation that was patched before publication. Post-patch validation on a second model confirmed the fix held.

Wave 1 aggregate: 112/112 (100%).

Scenarios tested and how reliably they work

Workflow	Models	Result
Baseline rate distribution (single LCAT keyword, full percentile pull)	Opus, Sonnet	Reliable
Keyword refinement via suggest-contains for noisy terms (Program Manager)	Opus, Sonnet	Reliable
SIN-filtered query with clearance layer (HACS, 54151HACS)	Opus, Sonnet	Reliable
Rate validation workflow ($185/hr Senior Software Developer, IGCE narrative)	Opus, Sonnet	Reliable

Manual-verification checklist

Scan every CALC+ output for these items before using in a contract file:

1. Confirm the aggregation path was used correctly. The skill's canonical statistics all live under `response["aggregations"]`, NOT at the top level. Pre-patch the skill documented them as top-level which caused silent `None` returns. If an output contains `None` or zero values where you expect real statistics, the model likely read from the wrong path.

2. `keyword=` is for scoping only, not rate stats. The `keyword=` parameter searches three fields simultaneously (labor_category, vendor_name, idv_piid). For rate analysis, use `suggest-contains=labor_category:<term>` to find real LCAT bucket names, then `search=labor_category:<exact value>` for stats. Outputs citing keyword-query rates in a price reasonableness memo should be flagged.

3. `security_clearance:yes` is a floor, not a true count. The filter does fuzzy partial-matching on variant bucket keys (`true`, `Yes`, `SECRET or TS/SCI eligible`). Counts are floors. For cleared requirements, the all-clearance HACS or cyber pool is often the more defensible benchmark than the cleared-filtered subset.

4. Record counts above 10,000 require the aggregation count. `hits.total.value` caps at 10,000 with `relation: "gte"`. Use `response["aggregations"]["wage_stats"]["count"]` for the true population size. A memo citing "10,000 records" when the true population is 45,000 misrepresents sample size.

5. SIN filter is case-sensitive. `54151HACS` works; `54151hacs` returns zero. Check SIN casing in the output.

6. Apply `price_range:30,500` by default for professional services IGCE work. CALC+ contains sentinel values ($9,999 placeholders, clerical floor rates) from messy vendor PPT submissions. The default bound excludes these. Adjust for specialty work (SCIF `50,600`; administrative `15,200`). Document the bound used in the memo.

7. `median_price` vs `histogram_percentiles["50.0"]`. Both present a "median" but use different interpolation. `histogram_percentiles["50.0"]` is the canonical median (linear interpolation between ranked values). If an output cites $1-$2 different from what the web UI shows, this is likely the cause; cite the histogram_percentiles value.

8. Dual-pool analysis for senior LCATs. For a "Senior X" requirement, both `suggest-contains=labor_category:Senior X` (title-match) and `X + min_years_experience:8` (experience-match) are legitimate benchmark pools. They often differ by 10-15% at median. The CO decides which best matches the vendor's LCAT description. An output citing only one pool should be flagged.

Choosing between Opus 4.7 and Sonnet 4.6

Short answer: both work. Use either. Post-patch behavior is effectively identical.

Opus 4.7 tends toward comprehensive workflow execution. In testing, Opus provided 4-pool comparisons for rate validation, dual SIN cross-checks for HACS work, and more detailed narrative con-

text. Opus's comparisons were slightly more thorough but no more defensibly accurate than Sonnet's.

Sonnet 4.6 is tighter in output and frequently uses exact-match searches over keyword queries. Sonnet S2 and S4 in testing produced visualization-rich output (ASCII percentile positioning, comparison tables) that reads well in a CO memo. Sonnet's exact-match discipline is methodologically slightly tighter than Opus's keyword-based approach.

Neither model hits tool-use limits on CALC+ queries. Both skills complete in a single response.

What the skill does not do

- **It does not reflect task-order-level prices paid.** CALC+ contains awarded NTE (Not-To-Exceed) ceiling rates from master MAS contracts. Order-level actuals are typically 10-20% below ceiling per FAR 8.405-2(d).
- **It does not apply geographic cost adjustment.** Rates are worldwide. A cleared DC-area vendor and an uncleared Midwest vendor may appear in the same bucket.
- **It does not substitute for a price reasonableness determination.** The CO still makes the determination per FAR 15.4. CALC+ data informs it; it does not replace it.
- **It has not been tested on** multi-vendor competitive range analysis, vendor-specific rate card queries for IGCE comparison, historical rate trending across contract years, or LCAT-to-LCAT comparison across multiple SINs.

Environmental gotchas on claude.ai web chat

Gotcha	What happens	Workaround
`aggregations` silently returns None if read from wrong path	Pre-patch skill documented paths as top-level	Fixed in Round 2 patch; current skill documents nested path explicitly
`suggest-contains` returns exactly 100 buckets	Default cap; silent truncation on noisy terms	Narrow the discovery term if you hit exactly 100
Keyword contamination understatement	Skill's anti-pattern language may overstate cross-field contamination on specific terms	For terms like "Program Manager" the 100+ long tail of LCAT variants dominates; always verify with `sum_other_doc_count` check

---

Part 2: For Developers and Technical Reviewers

Testing methodology

Scenarios

Four scenarios selected before any testing began, chosen to exercise distinct CALC+ workflows:

- **S1 — Baseline reachability + aggregation path:** Full rate distribution for Information Security Analyst. Tests endpoint, parameter name (`keyword` vs `q`), aggregation path (`histogram_percentiles.values["50.0"]` vs top-level), distribution presentation.
- **S2 — High-ambiguity keyword refinement:** Program Manager via `suggest-contains` then `search=labor_category:<exact>`. Tests search refinement, large-result-set handling, discovery discipline.

- **S3 — SIN-filtered cleared cyber:** Rates under SIN 54151HACS with clearance layer. Tests SIN filter syntax, cleared LCAT data availability, filter+keyword combo, `security_clearance` filter behavior.
- **S4 — Rate validation workflow:** $185/hr Senior Software Developer reasonableness check. Tests real IGCE decision-support, percentile positioning, divergence analysis, narrative production.

Each scenario had a 14-point binary assertion matrix: 8 general (G1-G8, same across all scenarios) and 6 scenario-specific. Assertions written before any worker output was seen.

Environment

- claude.ai web chat, fresh conversation per run
- Skills installed: `gsa-calc-ceilingrates` with reference skill merged into main SKILL.md (Round 2)
- Models: one run on Opus 4.7, one run on Sonnet 4.6, per scenario
- Total: 4 scenarios × 2 models = 8 runs

Grading

Grader (Claude Code session separate from any worker run) read only the worker's final response text. Workers not coached during runs. Each assertion graded binary pass/fail. Suspicious details noted even when the assertion passed.

Wave 1 results

Scenario	Opus 4.7	Sonnet 4.6
S1 Baseline distribution	14/14	14/14
S2 Keyword refinement	14/14	14/14
S3 SIN filter + cleared	14/14	14/14
S4 Rate validation	14/14	14/14
Total	**56/56 (100%)**	**56/56 (100%)**

Aggregate: 112/112 (100%).

Cross-model data cross-validation

Both models pulled identical raw data from the API, confirming the skill's endpoint and parameters work consistently:

- **S1 Information Security Analyst (unfiltered):** Both models N=402, median $128.01-$128.02 (rounding), mean $133.21.
- **S2 Program Manager (broad keyword):** Both models N=7,763, P50 ≈ $180.70.
- **S3 HACS SIN cleared cyber:** Both models N=1,148, median $150.07.
- **S4 Senior Software Developer:** Opus (keyword-based) N=145 median $151.66; Sonnet (exact-match) N=54 median $157.44. Different sample sizes reflect different methodological approaches (keyword vs exact match), both defensible. Documented as a "dual-pool" patch candidate.

Round 1 findings: 9 skill bugs surfaced

From Wave 1 Opus worker self-assessments (ordered by severity):

1. **Aggregations schema documentation wrong (CRITICAL).** Skill documented `wage_stats`, `histogram_percentiles`, etc. as top-level response fields. Actual API nests all aggregations under `response.aggregations.*`. Caused silent `None` returns. 3 of 4 Opus workers hit this and corrected in-flight.
2. **`keyword=` contamination across 3 fields.** The parameter searches `labor_category`, `vendor_name`, and `idv_piid` simultaneously. Skill did not flag this as an anti-pattern for rate statistics.
3. **"Discover first" was scoped to vendor names in Rule #2.** Should be labor_category first (noisier and more relevant to IGCE use case).
4. **`security_clearance:yes` is fuzzy, not normalized.** Skill claimed "API normalizes"; actually returns multiple variant bucket keys (`true`, `Yes`, `SECRET or TS/SCI eligible`). Counts with this filter are floors.
5. **`hits.total` caps at 10,000 buried under Rate Limiting.** Should be prominent in "reading results" context. Sanity checks comparing `hits.total.value` vs `wage_stats.count` will false-alarm on every query over 10K.
6. **`suggest-contains` default 100-bucket cap undocumented.** Silent truncation on noisy terms.

7. **`page_size=1` aggregation-only idiom undocumented.** Efficient pattern for distribution pulls was not called out.
8. **SIN filter case sensitivity undocumented.** `54151HACS` works; `54151hacs` returns empty.
9. **Filter encoding guidance missing.** Pipe-delimited OR and comma-ranges work but when to URL-encode vs not was unclear.

Patches shipped before Sonnet wave

All patches applied to `~/.claude/skills/gsa-calc-ceilingrates/SKILL.md`. The `gsa-calc-ceilingrates-reference` split-skill architecture was merged into the main SKILL.md in the same pass (mirrors the BLS OEWS merge).

1. **New "Response Shape" section at top** with full envelope tree showing nested `aggregations.*` paths. Canonical JSON paths for the 7 IGCE-relevant statistics.
2. **New `distribution_snapshot()` recipe** as the Quick Start. Returns count/min/max/mean/P25/P50/P75 in one call with the correct paths baked in.
3. **Critical Rule #2 rewritten: "Never use `keyword=` for rate statistics."** Explicit anti-pattern with the Program Manager contamination example.
4. **Critical Rule #3 rewritten: "Discover Labor Categories First."** Reframed from vendor-name scope to LCAT scope (higher IGCE value).
5. **Critical Rule #4 added: "`security_clearance` filter is fuzzy, not normalized."** Documented variant bucket keys, counts as floors, recommendation to use all-clearance pool for IGCE.
6. **Critical Rule #5 added: "`hits.total` caps at 10,000."** Moved from Rate Limiting to prominent context; explicit guidance on `wage_stats.count` as source of truth.
7. **Critical Rule #7 added: "Use `page_size=1` for aggregation-only queries."**
8. **Critical Rule #6 added: "Messy Field Values"** including SIN case sensitivity, LCAT discovery, vendor name discovery.
9. **Filter encoding guidance added** (pipe-OR, comma-ranges, when to quote vs not).
10. **`price_range:30,500` outlier-strip convention** documented as IGCE default with adjustment ranges for specialty work.
11. **New `price_reasonableness()` dual-pool recipe** for senior LCATs (title-match vs experience-match pools).
12. **New `tiered_rate_card()` recipe** for seniority stratification.

13. **`median_price` vs `histogram_percentiles["50.0"]` interpolation difference documented.**
14. **`suggest-contains` 100-bucket cap documented** with truncation warning.
15. **Quick Reference workflow section at the end** (discover → stratify → position → narrate).
16. **Expanded Troubleshooting table** with all silent-wrong-answer bugs.

Wave 1 Sonnet validated the patches

Sonnet wave executed after Round 2 patches shipped. All 4 Sonnet runs (56/56) used the patched response paths correctly from the first attempt: - No NullType crashes (pre-patch, Opus hit the bug on 3 of 4 runs) - All Sonnet workers used `aggregations.wage_stats.*` from the start - All applied `price_range:30,500` convention - All followed `suggest-contains` → `search=labor_category:<exact>` discipline - Sonnet S3 empirically tested and confirmed the SIN case sensitivity patch

Truncation investigation

A targeted audit asked an Opus Wave 2 session to quote verbatim from three specific locations in the 719-line merged file: 1. Critical Rule #4 heading and first sentence (line 177) 2. `tiered_rate_card()` function signature (line 399) 3. Troubleshooting table entry for `wage_stats` returns None (line 678)

Worker returned all three verbatim with correct line numbers. **Truncation is not real web-chat behavior.** Workers who described mid-file sections as "truncated" were imprecise about "I didn't re-read the full file to check." The 719-line merged file loads fully. Same finding as BLS.

What was not tested

- Multi-vendor competitive range analysis
- Vendor-specific rate card queries (`vendor_rate_card()` recipe in skill)
- Historical rate trending across contract years
- LCAT-to-LCAT comparison across multiple SINs in one query
- The `exclude` parameter for outlier removal
- CSV export (`&export=y` parameter)

Round 3 patches queued

These emerged from Wave 1 Sonnet self-assessments and one Opus Round 2 finding. None block the current ship state.

1. **Keyword contamination claim is overstated on common LCAT terms.** Opus Round 2 worker empirically proved the 7,763 "Program Manager" keyword hits reconcile entirely to `labor_category` buckets (top 100 + `sum_other_doc_count` long tail). The real contamination driver on common terms is tier conflation, not cross-field pollution. Refine the patch language: contamination is technically possible per the API spec but often negligible in practice.
2. **Document methodology choice between keyword and exact match for same LCAT.** Sonnet S4 exact-match `search=labor_category:"Senior Software Developer"` got N=54; Opus S4 keyword search got N=145. Both defensible, different approaches. Dual-pool workflow should explicitly address this choice.
3. **Verify whether `suggest-contains` bucket cap is configurable via `size` parameter.** Currently unknown; skill notes the 100-bucket observation but has not tested expansion.
4. **Add `median_price` vs `histogram_percentiles["50.0"]` worked example.** Skill explains the interpolation difference in text; a numeric example on a known dataset would help COs defend the chosen value.
5. **Add CO decision tree for pool selection.** When SOW language is title-weighted vs experience-weighted vs SIN-specific, which pool is the right anchor? Explicit flowchart would reduce methodology inconsistency across analysts.

Independent grading methodology

Wave 1 testing record produced under consistent methodology:

- Scenarios and assertion matrices committed in writing before any worker output was read
- Grader did not coach workers during runs
- Assertions graded strict on literal wording; ambiguous assertions noted and refined for the next wave (not retroactively reinterpreted)
- Methodology source: `calc-plus-wave1-runbook.md`
- All findings come from direct observation of worker output, not inference from memory of prior sessions

> *Testing record prepared April 2026 by James Jenrette / 1102tools. Independent grading methodology. MIT licensed. Source: github.com/1102tools/federal-contracting-skills.*