SRH DAO

Executive Brief

Can SRH optimize the client LLM in production-relevant ways?

Yes — provided SRH has direct control of the inference wrapper (custom containers or self-hosted serving inside the client VPC), SRH can deliver measurable optimization as defined in the SAI Protocol Gate materials.

Highlights

What "optimize" means in SRH's model (per our technical briefs):

- Target outcome: Improve reliability and honesty without degrading task success, evidenced by:
  - Lower ASR and CUP
  - Improved HUCL and UCE
  - Maintained or improved task success under latency budgets
- Mechanism: Detect high-uncertainty "distribution gaps" with a multi-signal detector and apply minimal, policy-gated interventions that guide the model toward coherent, honest behavior.

What that requires technically (and is feasible with direct wrapper control):

- Signals and scoring
  - Run a deterministic scorer model in-parallel (e.g., Llama-3.1-8B T=0) to compute per-token NLL and entropy.
  - Capture auxiliary signals: entropy deltas, committee disagreement (two small diverse models), retrieval conflict index, tool error density.
  - Maintain EWMA baselines, drift monitors, and thresholding logic; support a fallback z-score mode if training data is sparse.
- Policy-gated interventions
  - Acknowledgment, Orthogonal Nudge (with tuned similarity bands and safety screening), and Bounded Synthesis—within strict latency budgets and dual-consent gates.
  - Kill switches, audit logs, and cooldowns to prevent oversteer.
- Markers-only console and KPI binding
  - Real-time operational markers and weekly reports.

  - Client computes KPIs; SRH only provides event markers and analysis windows for causal attribution.

What SRH cannot do under black-box managed endpoints:

- You won’t get true “logit→token” interception on managed services that don’t expose internals (Bedrock/Vertex default). You can still approximate signals via a parallel scorer on outputs, but you lose some precision/control and can’t enforce low-level sampling policies server-side
- Deeper server-side optimization (e.g., middleware hooks around decoding, deterministic logit capture, structured intervention routing) requires custom containers/self-hosting.

Expected impact (from the KPI handout and Technical Annex):

- Typical targets and examples shown include:
  - ASR reduction into ≤10%
  - HUCL ≥90%
  - UCE improvement ≥15% vs. baseline
  - Task success maintained or improved, with P95 latency ≤ 1.35–1.5× baseline depending on stack
- Validation via shadow/canary runs, pre-registered analysis, and safety non-regression checks.

Bottom line

- With direct wrapper/inference control in the client VPC, SRH can optimize the client LLM in production-relevant ways (honesty, calibration, reliability) using a markers-only console plus minimal, guardrailed interventions—without requiring raw data egress or ownership of KPIs.

- With only black-box managed endpoints: Partial optimization is possible via output-based scoring and policy-layer prompting, but you should expect smaller gains and less precise control compared to custom/self-hosted serving.