From: Anna Hix <hix.anna@gmail.com>
Sent: Monday, September 14, 2026 5:04 PM
To: Comments
Subject: [EXT]: PCAOB Release No. 2026-006, Request for Public Comment: Draft 2026-2030 Strategic Plan Goals and Objectives

Dear Board,
I am an independent practitioner, and I operate a daily behavioral monitor over five production large language model APIs from four vendors. I write in support of Objective 1.2 of the draft plan, which states: "We also intend to assess whether additional guidance is needed to address the use of AI and other emerging technologies used in financial reporting and the audit process." This letter supplies one concrete gap for that assessment, in four steps: the auditing doctrine at issue, why its conditions cannot be verified for a vendor-hosted model, what my measurements show, and three requests.

The doctrine at issue
AS 2201, Appendix B, "Benchmarking of Automated Controls," paragraphs .B28 through .B33, lets the auditor conclude that an entirely automated application control remains effective without repeating the prior year's tests. The conditions are specific. General controls over program changes, access, and operations must be effective and continue to be tested, and the auditor must verify that the control has not changed since the baseline (.B29). Paragraph .B31 lists as a risk factor the availability and reliability of a report of the compilation dates of the programs placed in production. Paragraph .B32 calls benchmarking especially effective for purchased software when the possibility of program changes is remote.
Why these conditions cannot be verified for a vendor-hosted model Issuers are beginning to embed vendor-hosted AI models in controls this framework was written to cover. For such a control, the problem is not that the benchmarking conditions are hard to satisfy. The records needed to satisfy them do not exist on the customer side of the API. The vendor can change model weights, quantization, serving configuration, or safety layers at any time, without notice, under a constant model identifier. No compilation-date report or any analog of one is exposed. And the .B32 premise inverts: purchased software earns benchmarking because the vendor rarely changes the program, while a served model is changed continuously and unilaterally by the vendor, invisibly to the customer relying on it.

What I measured
Whether this gap matters in operation is an empirical question, so I measured it. I built a bank of 68 multiple-choice questions about enterprise security decisions. 45 are clear-cut: one answer any practitioner would give. 23 are judgment questions: all four choices are defensible, and good practitioners would split on them. Every day at 13:00 UTC, a script asks each of the five models every question ten times and records how the ten answers split. On August 2 I froze each question's answer split as its baseline. Each day's split is scored for distance against that baseline, and every question carries a pre-set alarm line calibrated so that chance alone trips it about one day in a hundred. A tripped alarm triggers an immediate re-ask of that question. The alarm rules and the verdict log were committed to a public repository on August 2, before the data.

What the record shows
The figures below are from the August 20 snapshot, distributed to the CSA AARM Working Group on August 21 and committed to the same public repository. On the clear-cut questions, 4,003 daily checks produced zero alarms. All 43 alarms landed on the judgment questions, and the behavior behind them flipped and reverted on a day timescale. Across all 67,070 recorded calls, the echoed model identifier took exactly one value per model: zero bits of information, through every flip and every reversion. The conclusion I draw for the Board is narrow.
Behavioral monitoring from the customer side can detect that an automated control changed. It cannot say what changed, when it deployed, or whom to hold to it. Those answers exist only as infrastructure-side records on the

vendor's side of the API. The question bank, alarm rules, verdict log, and self-checking analysis scripts are public at github.com/annawhooo/self-context-calibration.

The Board's own comment file shows the question is live and unclaimed I read 70 of the 71 letters posted on the strategic priorities request (PCAOB No. 2026-001; letter 33 would not retrieve). No letter addresses Appendix B benchmarking as applied to AI-based controls. The premises appear separately: Comment Letter No. 45 observes that issuers are moving from deterministic controls to AI-based probabilistic controls, and Comment Letter No. 13 documents non-deterministic outputs and model drift as audit risks. No letter connects those premises to the reliance doctrine that assumes an unchanged control.

Three requests:
- Include vendor-hosted AI and LLM-based automated controls in the Objective 1.2 guidance assessment, and address specifically whether the conditions of the Appendix B benchmarking strategy can be met for such controls.
- Identify the vendor-side records that would serve as the analog of the .B31 compilation-date report: change disclosures or attestations sufficient for an auditor to verify that a served model has not changed since a baseline was established.
- Under Goal 5, Modernize Oversight Through Technology and Data, recognize customer-side behavioral monitoring for what it is: a detection instrument deployable today, which identifies that behavior changed but cannot attribute the change to a cause, a deployment, or a responsible party.

I recognize this letter arrives after the September 4 comment date, and I appreciate the Board's consideration. The monitor runs daily and its record is public. I am glad to walk staff through the instrument or the record.

Respectfully submitted,

Anna Hix
hix.anna@gmail.com