40° South Guard

Guard alongside AWS Bedrock

Why this comes up

Bedrock Guardrails and 40 South Guard sound similar on the surface. Both inspect AI inputs and outputs. Both can act on PII. They aren't the same kind of tool, and the differences are most visible when an Australian regulator asks for evidence.

This document covers the four points that come up in technical evaluations: PII coverage, logging, cryptographic attestation, and the APRA supplier-risk angle.

1. PII detection coverage

Bedrock Guardrails ships built-in detectors for a fixed list of PII types. As of May 2026 the list includes generics (NAME, EMAIL, ADDRESS, PHONE, CREDIT_DEBIT_CARD_NUMBER, IP_ADDRESS) and country-specific identifiers for the UK, US, and Canada (UK_NATIONAL_INSURANCE_NUMBER, US_SOCIAL_SECURITY_NUMBER, CA_HEALTH_NUMBER, CA_SOCIAL_INSURANCE_NUMBER, plus a handful of US bank, tax, and passport entities).

Australian identifiers (Tax File Number, Medicare number, ABN, and BSB plus account number combinations) are **not** in the built-in list.

Bedrock supports custom regular expressions, so a customer can build their own AU detection patterns. Two practical issues with that path:

1. Regex alone doesn't validate checksums. A nine-digit string isn't necessarily a valid TFN. TFNs require a weighted-sum mod 11 check. Medicare numbers need a Luhn check on the first eight digits. ABNs need a weighted-sum mod 89 check (after subtracting 1 from the first digit). Without checksum validation, false-positive rates climb and operators stop trusting the alerts.
2. Maintenance and assurance sit with the customer. Each pattern, each profile, each regulatory change is the customer's problem, with no audit artefact showing the patterns were active at the time a call was made.

Guard ships checksum-validated detection for TFN, Medicare, ABN, BSB and account combinations, AU phone formats, dates of birth (with context-keyword scoping), and AU address patterns. Detection runs on every message in the request (every role, every position) so PII inside uploaded document content is caught, not just the user prompt. Each detection is configurable per API key (flag, redact, or block) and tied to a regulatory mapping in the resulting attestation.

2. Logging vs evidence

Bedrock writes invocation logs to CloudWatch and S3. With effort, customers can enable S3 Object Lock for WORM-style immutability and configure retention out to seven years. That's achievable, just not the default, and it isn't the same thing as cryptographic evidence.

Three differences come up in technical evaluations:

- **Cryptographic attestation per call.** Guard generates a per-call attestation signed with a Cloud KMS ECDSA P-256 key, with HMAC-SHA256 as fallback. The attestation ties the model response to the specific compliance controls that ran, including PII findings, injection findings, and the regulatory mapping. A Bedrock log records what was sent and received. A Guard attestation cryptographically proves which controls ran on that call.
- **Tamper evidence by default.** Guard's evidence vault writes to a GCS bucket with an immutability policy applied at provisioning. There's no opt-in step on the customer side. With Bedrock, immutability is a customer configuration choice and a customer audit responsibility.
- **Regulatory mapping.** Guard's attestation carries mappings to CPS 234 (sections S14, S15, S16, S17), ADM transparency controls (T1 to T4), and Privacy Act APP 8 where cross-border disclosure is detected. Bedrock logs don't carry regulatory metadata; that mapping has to be reconstructed downstream by the customer.

---

3. The APRA supplier-risk

APRA's letter to industry on artificial intelligence, dated 30 April 2026, sets explicit expectations for governance over the AI supply chain:

> *"APRA expects entities to manage supplier risks by mapping and maintaining visibility over the full AI supply chain, including material, third-party and fourth-party dependencies, with contractual and governance arrangements which provide sufficient transparency, auditability and assurance over AI services."*

For a regulated entity using Bedrock, this is a real obligation. AWS's Shared Responsibility Model puts AI-service governance on the customer side of the line. Bedrock doesn't produce the artefacts a CRO needs to demonstrate ongoing oversight to an APRA reviewer, particularly the cross-border (APP 8) and concentration-risk dimensions APRA called out.

Guard sits between the entity and Bedrock. Every call to a Bedrock model is intercepted, scanned, attested, and logged. The evidence vault is the artefact the CRO points to when APRA asks how supplier oversight is enforced in practice.

---

4. Where Guard fits in an AWS estate

Guard isn't a Bedrock replacement. Guard sits in front of it. The customer keeps using Bedrock as their model layer and adds Guard as their evidence and control layer.

The integration is a one-line change. The application points at Guard's endpoint instead of Bedrock's, with the same OpenAI-compatible payload. Guard authenticates the call, runs the scans, forwards to the upstream provider, scans the response, signs the attestation, and returns the response with an X-Guard-Attestation-ID header. Latency overhead is single-digit milliseconds for scanning; signing is async via Pub/Sub.

At-a-glance comparison

Capability	AWS Bedrock Guardrails	40 South Guard
Built-in AU PII (TFN, Medicare, ABN, BSB)	No	Yes, with checksum validation
Custom regex support	Yes	Yes
Per-call cryptographic attestation	No	Yes (Cloud KMS ECDSA P-256)
Tamper-evident audit by default	No (S3 Object Lock is opt-in)	Yes (immutable GCS evidence vault)
Regulatory mapping in	No	Yes (CPS 234, ADM, APP 8)
Cross-border disclosure flagging (APP 8)	No	Yes
AU data residency	Configurable per region	Enforced (australia-southeast1)
Maintenance burden for AU rules	Customer	40 South