

SUBMISSION

Australian Parliament Joint Select Committee on Artificial Intelligence

Governance requirements for AI systems that make and execute operational decisions

Submission by Urbyte Pty Ltd on the Keelo Executive Operating System

12 September 2026

Core proposition

As artificial intelligence moves from assistance into operational decision making and execution, governance must extend beyond the model. It must govern the evidence relied upon, persistent organisational state, authority to act, accountable ownership, execution records and verified outcomes.

Purpose of this submission

This submission asks the Committee to recognise an operational governance gap. Model evaluation and policy controls remain necessary, but they do not establish whether an AI enabled action used valid evidence, respected current authority, executed once, left a durable record, produced the intended outcome or created an unresolved obligation.

Keelo is an organisation agnostic Executive Operating System delivered as Waste Recovery Software as a Service. It governs the operational chain from verified evidence through state, authority, execution and outcome verification. Urbyte is a separate organisation package and initial proving application that verifies allocation waste. The technical annex records a completed controlled execution across 1,000 synthetic enterprise mobile services.

Submission detail	Position
Submitting entity	Urbyte Pty Ltd
Author	Jarryd Thamer, Founder and Chief Executive Officer
System addressed	Keelo Executive Operating System
Evidence annex	Controlled 1,000 service Urbyte and Keelo execution
Public contact	urbyte.com.au
Confidentiality	Public submission. Further non-public technical material will be provided only under an executed confidentiality agreement.

Executive submission

Requested findings

The Committee should find that governance for AI enabled operational systems must cover the full decision and execution chain. Assurance limited to model behaviour cannot establish operational legitimacy after a model output enters an organisation's systems.

- Evidence must retain source identity, integrity information, time, scope and the rule used to derive a decision relevant fact.
- Operational state must persist across process restarts and support an attributable history.
- Authority must be current, scoped to the organisation and action, and checked when execution occurs.
- Every consequential action must have an accountable owner and an identified actor.
- Concurrent or repeated instructions must not create duplicate external effects.
- Execution must produce a durable receipt linked to the governing decision and source evidence.
- Outcome verification and reconciliation must distinguish completed results from unresolved obligations.

Requested recommendations

Recommendation	Required policy effect
1 Extend assurance beyond models	Require assurance frameworks for evidence, state, authority, execution and outcomes where AI influences or initiates consequential operational action.
2 Require attributable evidence	Require provenance, integrity references, observation time, organisational scope and applicable method or policy version for material decisions.
3 Require execution controls	Require current authority checks, accountable ownership, optimistic concurrency or an equivalent state control, idempotency and duplicate safety.
4 Require outcome closure	Require durable execution receipts, independent outcome evidence where proportionate, reconciliation and explicit unresolved obligation handling.
5 Use controlled proving environments	Permit staged technical proofs and pilots that test failure modes before production deployment, without treating a synthetic proof as proof of production readiness.
6 Preserve technology neutrality	Define required operational properties and evidence rather than mandating a specific model, vendor or software category.

Why the distinction matters

An organisation may have a capable model and a sound policy yet still execute an invalid action. The source evidence may be incomplete. Authority may have been revoked. State may have changed after approval. A retry may create a second external effect. A provider response may not prove the intended outcome. These are operational control failures. They require system controls around the model, not additional claims about the model itself.

The operational governance gap

The governed object is the transaction

For consequential use, the relevant unit of assurance is the complete governed transaction. A transaction begins with attributable evidence and ends only when the outcome is verified or an unresolved obligation is retained. Model output may inform part of the transaction, but it does not replace the controls that establish whether an action was valid and completed.

Transaction stage	Control question	Required record
Evidence	What facts were relied upon and where did they come from	Source reference, digest, time, scope and confidence
Verification	What method converted evidence into a decision relevant condition	Method and policy version, calculation and disposition
State	What was true when the decision and action occurred	Persistent versioned state and append only events
Authority	Who could authorise and execute this action at that time	Actor, owner, grant or approval, scope and disposition
Execution	Was the external effect performed once under the authorised condition	Idempotency identity, expected version and durable receipt
Outcome	Did the action produce the intended result	Outcome evidence, verifier, time and measured result
Reconciliation	Do all inputs, decisions, actions and outcomes account for each other	Counts, exceptions and unresolved obligations

Persistent state is a governance requirement

Operational governance cannot depend on transient conversational context. State must survive restarts, retain prior and resulting versions, and support deterministic reconstruction from durable events. This allows an organisation to determine which evidence, authority and policy applied when an action occurred.

Authority must be checked at execution

Approval captured earlier in a process is insufficient if authority changes before execution. A governed system should bind the action to the organisation, actor, accountable owner, permitted operation and current authorisation. It should fail closed when authority or version requirements are not satisfied.

Outcome verification is distinct from execution

A successful request or provider receipt proves that an execution attempt was accepted or processed. It does not necessarily prove the intended operational result. Systems should retain the receipt and obtain outcome evidence appropriate to the action. Where verification fails or authority changes leave residual work, the obligation should remain visible until resolved.

Application to public policy

These controls are relevant where AI affects financial exposure, operational continuity, safety, compliance, infrastructure performance or customer outcomes. The degree of control should scale with consequence, reversibility, uncertainty and the quality of available evidence.

Keelo architecture and maturity

System definition

Keelo is an organisation agnostic Executive Operating System delivered as Waste Recovery Software as a Service. It maintains governed organisational state and applies evidence, authority and execution controls to operational conditions. Organisation specific definitions and rules remain in separate packages. Urbyte supplies one such package for allocation waste verification.

Maturity class	Meaning in this submission
Evidenced Current	Implemented and supported by committed code, automated tests or the completed controlled proof described in Annex A.
Next Proving Work	Defined engineering work required to strengthen scale, failure recovery, security assurance or real customer evidence before production claims.
Destination Case	The intended operating capability after further engineering, assurance, deployment and customer validation. It is not represented as live.

Evidenced Current

Capability	Current evidence	Boundary
Evidence provenance	Each qualifying billing row receives an ingestion identity and digest. Verified records retain evidence references.	Controlled synthetic billing export
Persistent state	Firestore backed transaction state and events were used in the completed controlled execution.	Emulator environment, not production infrastructure
Authority and ownership	Transactions bind an actor, accountable owner, policy evaluation, compliance review and principal approval.	Configured controlled proof identities
Expected version	Execute and verify outcome operations require the expected transaction version.	Optimistic concurrency within the implemented transaction service
Idempotency and duplicate safety	Transaction and external request identities constrain replay. The fleet proof reported zero duplicate external executions.	Controlled adapter and proof conditions
Controlled external execution	Authorised actions reached a separate loopback HTTP receiver through the controlled adapter.	Reversible proof receiver only
Durable receipts	External provider request and object identities were persisted against the governed transaction.	Receipt durability demonstrated through the proof store
Outcome verification	The system observed controlled external object state and recorded verified outcomes.	Binary controlled hold observation
Reconciliation	All 1,000 services were classified once and all completed actions reconciled.	Single deterministic synthetic dataset

Next Proving Work

- Implement and test atomic cross process leases for concurrent workers and recovery after worker termination.

- Propagate authority revocation through queued, executing and unresolved transactions, with explicit compensating or containment behaviour.
- Complete production Runtime composition beyond the evidenced durable scheduler composition, including secrets, observability, backup, restore and disaster recovery evidence.
- Extend persisted external action retry timing into full crash recovery, exhaustion, operator intervention and reconciliation proof.
- Run the same controlled transaction against customer supplied, permissioned evidence in a segregated pilot environment.
- Obtain independent security, privacy and control review before production reliance.

Destination Case

The destination case is a production governed execution layer that continuously maintains evidence linked organisational state, identifies attributable waste conditions, applies current authority, executes approved interventions through controlled adapters, verifies outcomes and retains unresolved obligations. This requires further engineering and external assurance. The completed fleet proof does not establish that destination case.

Governance and assurance implications

Minimum control properties

Property	Minimum expectation	Assurance evidence
Provenance	Trace material facts to source records and integrity references	Source identity, digest, capture time and scope
Persistence	Retain state and history outside a single process	State records, events, restart and restore tests
Authority	Verify current permission for the specific action	Grant or approval identity, actor, action scope and time
Accountability	Assign an owner for the condition and residual obligation	Owner identity and status history
Concurrency	Reject stale updates	Expected version failures and transaction tests
Idempotency	Return or reconcile the original result for exact replay	Stable request identity and replay tests
Duplicate safety	Prevent multiple external effects for one authorised action	External mutation counts and concurrent delivery tests
Receipt durability	Retain evidence of the external execution	Provider request, object and response digest
Outcome closure	Verify intended effect or retain an unresolved obligation	Outcome evidence and reconciliation report

Risk proportionality

The controls should scale with the consequence and reversibility of the action. A low consequence recommendation may need provenance and human review. A consequential or externally executed action should also require current authority, state version checks, duplicate safety, durable receipts and outcome closure. Higher consequence uses may require independent verification, separation of duties and stronger recovery controls.

Privacy and data minimisation

Governed execution does not remove privacy obligations. Evidence ingestion should be permissioned, purpose limited and minimised. Organisations should separate identity from analytical fields where possible, apply retention rules, restrict cross organisation access and record who accessed or acted on evidence. The controlled annex used synthetic service records and does not demonstrate compliance for customer personal information.

Security and operational resilience

A production system would require threat modelling, least privilege access, secret management, monitoring, incident response, tested backup and restore, recovery objectives, dependency controls and independent assurance. Current automated tests and emulator persistence support engineering confidence but do not constitute certification or production security accreditation.

Environmental and ESG reporting

Waste verification may support environmental management and ESG reporting when the organisation can establish a causal and evidentiary link to a measured environmental outcome. Keelo should retain the organisational boundary, baseline, reporting period, method and evidence references needed for such claims. The telecommunications fleet proof measured allocation waste and controlled execution. It did not establish energy, emissions or ISO 14001 outcomes and must not be used to claim them.

Conclusion

Australia's AI assurance settings should address the operational system in which AI acts. Model evaluation remains necessary, but it cannot by itself establish evidence integrity, current authority, accountable execution, duplicate safety or verified outcomes. Those properties determine whether an AI influenced operational action is governable after it leaves the model boundary.

The controlled proof in Annex A demonstrates that this wider control chain can be implemented and tested. It also shows the importance of explicit maturity claims. The proof establishes bounded mechanics in a synthetic environment. It does not establish production deployment, customer savings, environmental benefit or general readiness for high consequence use.

The Committee should recommend outcome based operational control requirements for consequential AI systems and support controlled proving environments that allow organisations to demonstrate those requirements before production deployment.

Annex A Technical Proof

Keelo and Urbyte 1,000 Service Controlled Execution Record

Annex purpose

This annex records a completed controlled execution that tested the chain from raw allocation evidence to Urbyte verification, Keelo governed state and authority, controlled external action, durable receipt, outcome verification and final reconciliation. It separates observed results from future work and intended capability.

Evidence identity

Field	Controlled proof record
Repository commit	0bab2ec01bd3751a89f04e90787af048c79bf570
Current verification head	878b7ead6a125c6c9415f19dad320c96799a1ca3
Branch at verification	work/current-203
Commit subject	proof: add 1000-service enterprise fleet execution
Commit timestamp	11 September 2026 14:00:38 minus 04:00
Dataset	controlled-enterprise-fleet-billing-1000
Dataset specification	fixtures/urbyte-enterprise-fleet-billing.json
Runner	scripts/run-urbyte-enterprise-fleet.mjs
Package command	npm run run:urbyte-enterprise-fleet
Environment lock	GCLOUD_PROJECT demo-keelo-emulator with FIRESTORE_EMULATOR_HOST required

Committed proof files

File	Purpose
fixtures/urbyte-enterprise-fleet-billing.json	Defines 1,000 services, allocation options and three contiguous 28 day billing periods.
scripts/run-urbyte-enterprise-fleet.mjs	Materialises deterministic evidence, ingests rows, verifies waste, executes governed transactions and reconciles results.
package.json	Registers the controlled fleet execution command.

Subsequent evidenced engineering

Commit	Evidenced Current change
d401c744f5d6e2986006e9fbd03087662f536961	Composed durable scheduler persistence into the Runtime and added persistence integration coverage.
5ef416c	Hardened the controlled execution boundary.
878b7ead6a125c6c9415f19dad320c96799a1ca3	Persisted external action retry timing.

These commits strengthen the current architecture after the fleet proof commit. They do not change the recorded 1,000-service metrics or convert the controlled proof into a production deployment.

Evidence provenance

The source is a deterministic synthetic enterprise billing export specification. The runner derives service identities and variation from SHA 256 entropy using the dataset identity and service identity. It does not ingest carrier or customer production data. Each generated billing row includes a service identity, period, allocation, usage when present and confidence. Each row is assigned an ingestion identity and a SHA 256 digest before verification.

Raw input measure	Value
Distinct services	1,000
Billing periods specified	3 contiguous periods of 28 days
Materialised billing rows	2,938
Declared allocation across rows	211,580 GB
Declared usage where present	104,182.162 GB
Rows missing usage	58
Derived dataset digest	sha256:aa63580ec841fdcea51cadbf89e3bbcdb21c589b23b6dbd5a5a7552f9838e345

Raw totals above were independently regenerated from the committed dataset specification and runner algorithm on 12 September 2026. Floating point display was normalised to three decimal places for the usage total.

Verification method

Stage	Implemented method
Baseline qualification	Require three rows, contiguous billing periods and at least 84 days from first start to last end.
Evidence validation	Require positive allocation, present non negative usage, usage no greater than allocation and sufficient confidence.
Allocation record	Aggregate the three periods into one service allocation with evidence references and minimum observed confidence.
Waste verification	Apply the Urbyte telecommunications verification method with a 10 GB forecast demand allowance, minimum confidence 0.9 and policy version 1.0.0.
DBU record	Establish 10 Data Backed Units per verified unused GB. DBUs are non transferable internal records. They are not currency or tokens.
Governed condition	Bind the verified record to a transaction, accountable owner, actor, policy evaluation, compliance review and approval.
Execution	Require the expected version and current authorisation before applying a controlled verification hold through a separate HTTP receiver.
Outcome	Observe whether the controlled external object exists, then record an outcome verified state with an evidence identity.
Reconciliation	Account for every service once and compare completed actions with verified or unresolved outcomes.

Evidenced Current results

Measure	Result	Interpretation
Services discovered	1,000	All specified service identities materialised
Verified services	805	Met evidence and baseline requirements
Actionable verified services	770	Verified unused allocation remained after the forecast allowance
Verified no action services	35	Valid evidence but no positive recoverable remainder
Unverifiable services	195	Excluded from action
Verified unused allocation	79,087.919 GB	System derived across actionable verified services
Data Backed Units	790,879.19	Internal records at 10 DBUs per verified unused GB
Governed conditions	770	One for each actionable verified service
Authorised external actions	770	All controlled actions passed configured authority checks
External executions	770	Controlled receiver mutations
Durable external receipts	770	One receipt retained per execution
Verified outcomes	770	All controlled external states observed and recorded
Duplicate executions	0	No receiver mutation count exceeded one

Measure	Result	Interpretation
Unresolved obligations	0	No completed transaction remained unresolved
Final reconciliation	1,000 of 1,000	770 actionable plus 35 no action plus 195 excluded

Derived rates

Rate	Calculation	Result
Verification rate	805 divided by 1,000	80.5 percent
Exclusion rate	195 divided by 1,000	19.5 percent
Actionable rate	770 divided by 1,000	77.0 percent
No action rate	35 divided by 1,000	3.5 percent
Outcome closure rate	770 divided by 770 executed actions	100.0 percent in the controlled run

Control observations

Required control	Evidence in the completed proof	Status
Evidence provenance	Ingestion identities and digests linked qualifying rows to DBUs, conditions, actions and receipts.	Evidenced Current
Persistent state	Firestore backed transaction records and audit events held state during the controlled run.	Evidenced Current
Authority controls	Configured policy, compliance and principal decisions produced an AUTHORISED disposition before execution.	Evidenced Current
Ownership and accountability	Each governed transaction identified the enterprise fleet operations owner and authorised principal.	Evidenced Current
Expected version	Execution and outcome verification used the transaction version returned by the preceding state transition.	Evidenced Current
Idempotency	Stable transaction and external request identities were applied throughout the run.	Evidenced Current
Duplicate safety	The receiver reported 770 executions and zero duplicates.	Evidenced Current
Controlled external execution	A separate loopback HTTP receiver accepted reversible APPLY VERIFICATION HOLD actions.	Evidenced Current
Durable receipts	Each completed action retained provider request, external object and evidence linkage fields.	Evidenced Current
Outcome verification	External object presence was observed and recorded for each executed action.	Evidenced Current
Reconciliation	All services and actions reconciled, with no unresolved obligation.	Evidenced Current

Automated verification

Gate	Result	Date and qualification
Architecture boundary check	Passed	12 September 2026
TypeScript typecheck	Passed	12 September 2026
Unit tests	506 of 506 passed across 44 files	12 September 2026 using Vitest 5.0.0
Firestore integration tests	19 of 19 passed	Completed proof record reported before this submission build
Fleet proof execution	PASS with metrics stated above	Completed controlled run represented by the committed proof
Fleet rerun during submission build	Passed	12 September 2026. Controlled fleet proof rerun completed successfully with metrics consistent with the committed proof record.

Limitations

- The billing evidence is synthetic and deterministic. It is not customer, carrier or production evidence.
- The proof used one organisation, one dataset, one policy version and one controlled environment.

- The controlled dataset used individually attributable service records. Shared pools, consolidated accounts, multiple carriers and mixed allocation structures remain customer evidence proving cases. Each requires a defined allocation boundary and reconciliation method.
- The external receiver was a reversible loopback service. It was not a carrier, billing, financial or customer production system.
- The run demonstrates controlled throughput for 1,000 services. It does not establish sustained performance, multi tenant scale or service level objectives.
- Configured proof approvals demonstrate enforcement mechanics. They do not establish a customer's legal authority, consent or regulatory compliance.
- A recorded external receipt proves the controlled execution record. The outcome check proves only the defined controlled hold state.
- No realised financial saving, revenue, consumer benefit, emissions reduction or ISO 14001 outcome was measured.
- No independent penetration test, security certification, privacy impact assessment or production accreditation is claimed.
- The proof does not close cross process leases, authority revocation propagation, full external action recovery or production deployment assurance. Durable scheduler composition and persisted retry timing are now evidenced in later commits but remain short of the destination case.

Next Proving Work

Work item	Evidence required to close it
Cross process execution safety	Concurrent worker, lease expiry, crash recovery and exactly once external mutation tests
External action recovery	Crash recovery, retry exhaustion, operator intervention and reconciliation tests building on persisted retry timing
Authority revocation	Tests covering revocation before dispatch, during execution and before outcome closure
Runtime composition	Production configured deployment with secrets, isolation, monitoring, backup and restore
Customer evidence pilot	Permissioned multi source customer evidence, documented data controls and reproduced results
Independent assurance	Security, privacy and control assessment with recorded findings and remediation

Destination Case

The destination case extends the proven transaction properties into a production service across organisations, evidence sources and consequential adapters. It would maintain current authority, persistent retries, cross process execution safety, verified outcomes, unresolved obligation handling and independent assurance. That capability remains conditional on the next proving work and must not be inferred from this annex.

Annex B Definitions

Term	Definition used in this submission
AI enabled operational system	A system in which AI output influences, proposes, authorises, schedules or initiates an operational decision or action.
Keelo	An organisation agnostic Executive Operating System delivered as Waste Recovery Software as a Service.
Urbyte	An industry and organisational agnostic allocation waste verification system. Its initial application verifies unused mobile data allowance across personal and fleet services.
Data Backed Unit	A non transferable internal platform record established from verified systemic waste. A DBU is not currency, a token or a transferable asset.
Expected version	The state version a command expects. A mismatch identifies stale state and prevents the transition.
Idempotency	The property that an exact repeated request returns or reconciles the original result without a second effect.
Durable receipt	A persistent execution record that links the governed action to the external request, object and response evidence.
Outcome verification	A separate determination that the intended external state or result exists, based on identified evidence.
Unresolved obligation	A retained duty to investigate, complete, contain or reverse work when a transaction cannot be fully closed.

Evidence status statement

All statements labelled Evidenced Current are bounded by the commit, tests, controlled environment and limitations recorded in Annex A. Next Proving Work is not complete. Destination Case statements describe intended capability only.