K

KGP-001 / WHITEPAPER

Permission infrastructure for physical AI.

A narrow, open authorization and accountability layer for machines that act in the real world.

STATUS Experimental open draft 0.1
DATE 10 August 2026
LICENSE Apache-2.0 reference implementation
WEB https://kinegrant.com

01 / ABSTRACT

Machines can act. They still need a common way to prove they were allowed.

Physical-AI systems increasingly cross boundaries between people, spaces, devices, and organizations. Existing standards describe robot communication, device access, privacy terms, and data-use policy, but they do not provide one common object that an actuator can verify immediately before a physical action and bind to a signed outcome afterward.

KineGrant introduces a narrow execution layer: normalize an action request, evaluate all applicable policy with default-deny semantics, issue a short-lived signed capability, verify it locally at the actuator gate, then emit a privacy-minimized signed receipt.

KineGrant is not a robot operating system, motion planner, functional-safety controller, identity registry, blockchain, token, or legal authority. It is an experimental interoperability proposal designed to integrate with existing systems without widening their permissions.

Core path

REQUEST	POLICY	CAPABILITY	GATE	RECEIPT
intent	decision	permission	local verify	attestation

Design goals

DEFAULT DENY	Unknown or incomplete authorization never becomes permission.
LOCAL VERIFICATION	The actuator does not wait for a cloud service or public ledger.
REQUEST BINDING	A capability cannot be repurposed for another agent, target, action, or purpose.
MINIMAL RECEIPTS	Auditability does not require raw household video, audio, or location logs.

02 / CORE OBJECTS

Five objects, one enforceable authorization path.

2.1 ACTION REQUEST

A globally unique request binds an authenticated agent to a physical or cyber-physical target, normalized action, declared purpose, issue time, and policy-visible context. Its canonical SHA-256 digest anchors every later object.

2.2 POLICY RULE

A rule carries issuer, target pattern, allow or deny effect, actions, subjects, purposes, constraints, and obligations. Engines default to deny. Draft 0.1 uses deny-overrides: one applicable prohibition defeats every allow rule.

2.3 DECISION

A decision records the request digest, outcome, reason, matched policies, and obligations. It is evidence of evaluation; it is not permission by itself.

2.4 PHYSICAL ACTION CAPABILITY

Only an allowed decision can produce a capability. It is signed by a trusted authority, bound to request and policy digests, includes a random nonce, expires within five minutes, and is one-time by default.

2.5 ACTION GATE AND RECEIPT

The gate sits as close to the actuator as possible and fails closed on invalid signature, issuer, time, binding, or replay state. After execution, the executor signs a receipt with result, interval, optional evidence hash, and previous-receipt hash.

Illustrative capability envelope

```
{
  "type": "kinegrant:PhysicalActionCapability",
  "agent": "urn:robot:delivery-1",
  "target": "urn:space:demo:door-7",
  "action": "open", "purpose": "delivery",
  "expires_at": "...", "nonce": "...",
  "signature": { "alg": "Ed25519", "kid": "..." }
}
```

03 / DECISION SEMANTICS

Safe, deterministic, and deliberately conservative.

The v0.1 policy engine evaluates applicable rules against the same normalized request. It rejects requests when no applicable allow exists, when any applicable deny exists, or when a required constraint cannot be interpreted.

Unknown restrictions must never silently become permission.

Normative evaluation order

01	Normalize and validate the ActionRequest.
02	Resolve authenticated policy sources and trusted issuers.
03	Collect applicable allow and deny rules.
04	Apply deny-overrides and default-deny semantics.
05	Record obligations and compute request and policy digests.
06	Issue a short-lived capability only for an allowed decision.
07	Verify and consume the capability once at the local gate.
08	Emit a signed, privacy-minimized receipt.

Conflict example

A delivery policy allows robot A to open door 7 for delivery. A space-level safety policy denies all opening during a fire-alarm lockout. Both rules apply; the prohibition wins and no capability is issued.

04 / SECURITY MODEL

A signature is necessary. It is not physical truth.

PROPERTY	V0.1 MECHANISM	LIMIT
Authenticity	Ed25519 signed envelopes	Stolen keys remain dangerous
Scope	Agent/target/action/purpose/request binding	Identity enrollment is external
Freshness	1-300 second lifetime	Secure time is deployment-specific
Replay	One-time nonce cache	Reference cache is in-memory
Audit	Signed hash-chained receipts	Executor may still lie
Privacy	Evidence hashes, no raw media required	Traffic analysis remains

Required deployment hardening

Use hardware-backed authority and executor keys; place the gate inside the trusted actuator path; persist replay state across restarts; distribute revocation locally; pin adapter profiles; use trustworthy time; checkpoint receipts outside the executor; and retain an independent functional-safety controller.

Do not deploy draft 0.1 as the sole control for hazardous machinery. Independent cryptographic, robotics-safety, and privacy review is required.

05 / INTEROPERABILITY

Standards remain authoritative at their own boundaries.

SOURCE	REUSED CONCEPT	KINEGRANT BOUNDARY
W3C ODRL 2.2	permission, prohibition, duty	conservative PolicyRule profile
IEEE 7012-2025	machine-readable privacy terms	experimental privacy bridge
W3C WoT TD	thing identity and actions	target/action discovery
ROS 2 / SROS2	robot identity and middleware ACL	physical request adapter
OPC UA	nodes, methods, roles	method-to-target adapter
Matter	fabric, endpoint, cluster, command	command request adapter

Cumulative authorization

Native platform authorization AND a valid KineGrant capability AND the local safety controller must all allow. Any layer may veto; no layer may force another to permit.

The adapters are reference profiles, not certification claims or endorsements by the named standards organizations.

06 / DEPLOYMENT AND ROADMAP

Local safety path. Optional asynchronous anchoring.

KineGrant does not mandate a discovery transport. A target may expose authenticated identity and policy pointers through WoT, QR or fiducial markers, NFC, BLE, Matter, OPC UA, or a trusted local registry. Unauthenticated discovery may add restrictions but must not grant capability.

Implementations may anchor hashes of identities, policy versions, revocations, or receipt batches to a public or consortium ledger. Ledger confirmation must never be required for emergency stop or immediate actuator safety.

Milestones

0.1	**Executable kernel**	Python reference, tests, adapters, threat model
0.2	**Precise profiles**	strict schemas, canonical encoding, revocation, delegation
0.3	**Robot bridge**	ROS 2 gate, persisted replay cache, fault-injected simulation
0.4	**Hardware trust**	secure elements, measured gate build, trustworthy time
0.5	**Privacy + formal methods**	selective disclosure, model checking, adapter fuzzing
1.0	**Open standard candidate**	independent maintainers and interoperable implementations

07 / OPEN QUESTIONS

The work starts where the draft is weakest.

01	Policy conflict among legitimate co-owners.
02	Revocation during an already-running action.
03	Delegation, attenuation, and multi-agent actions.
04	Privacy-preserving subject and target discovery.
05	Hardware-backed actuator attestation.
06	Emergency and accessibility override semantics.
07	Formal verification of adapter behavior.

Success criteria

KineGrant succeeds when unrelated robot stacks can request the same capability, people and spaces can publish one rule understood across implementations, safety auditors can reproduce decisions, actuators cannot execute without verified capabilities, and receipts expose less personal data than raw operational logs.

The first meaningful signals are an external implementer, a valid vulnerability report, and a cross-device demonstration - not page views, token price, or unsupported claims of standard status.

08 / REFERENCES AND PARTICIPATION

Review the threat model before the marketing.

W3C ODRL Information Model 2.2 - https://www.w3.org/TR/odrl-model/

W3C Web of Things Thing Description 1.1 - https://www.w3.org/TR/wot-thing-description11/

IEEE 7012-2025 overview - https://standards.ieee.org/ieee/7012/7192/

ROS 2 access controls - https://docs.ros.org/en/humble/Tutorials/Advanced/Security/Access-Controls.html

OPC UA role-based security - https://reference.opcfoundation.org/specs/OPC-10000-18/4/

Matter specifications - https://csa-iot.org/developer-resource/specifications-download-request/

Participate

Download the v0.1 reference implementation, run the tests, challenge the semantics, and report bypasses or ambiguities. Security criticism is a primary contribution, not an obstacle.

```
Website: https://kinegrant.com
Protocol status: Experimental open draft 0.1
Reference implementation license: Apache-2.0
```

This document does not claim legal effect, safety certification, or endorsement by any standards organization.