Look Before You Leap: Preventing In-Motion K8s Updates in Fleet Robotics

KubeCon North America 2026:
Salt Lake CIty



Tom Ingleby
<tom.ingleby@agilityrobotics.com>

Rian Bogle
<rian.bogle@agilityrobotics.com>

Agility

Who we are
What we do

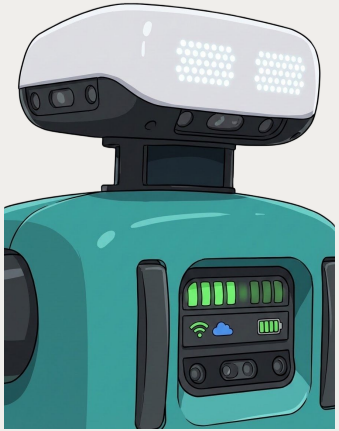
- 10 year old humanoid robotics startup
- Redefining labor for a future where humans and humanoids work together with ease
- Robots deployed in warehouses and factories alongside humans.
- Robots as a Service



Managing Fleets of Remote Devices

Agility

Robot Software Systems

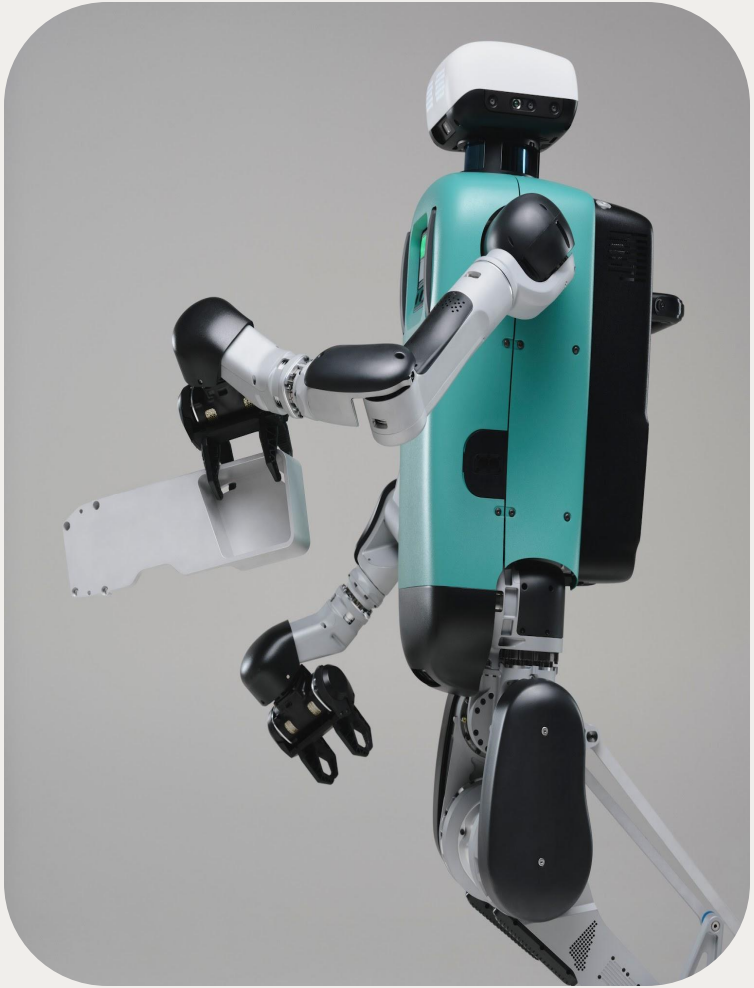


- State
- Control
- Perception
- Inference
- Skills
- Protocol Clients
- Observability
- System Management

Managing lifecycle of heterogeneous devices

Devices

- Robots
- Safety Systems
- Charger Systems
- Integration Devices



Management Platform Core Requirements

Connectivity

- Egress initiated
- Encrypted and Tunnelled
- Break Glass access
- JIT RBAC ingress
- Auditable

Device IAM

- Unique Identity
- Extensible
- Short Lived / Revocable
- Authentication and Authorization

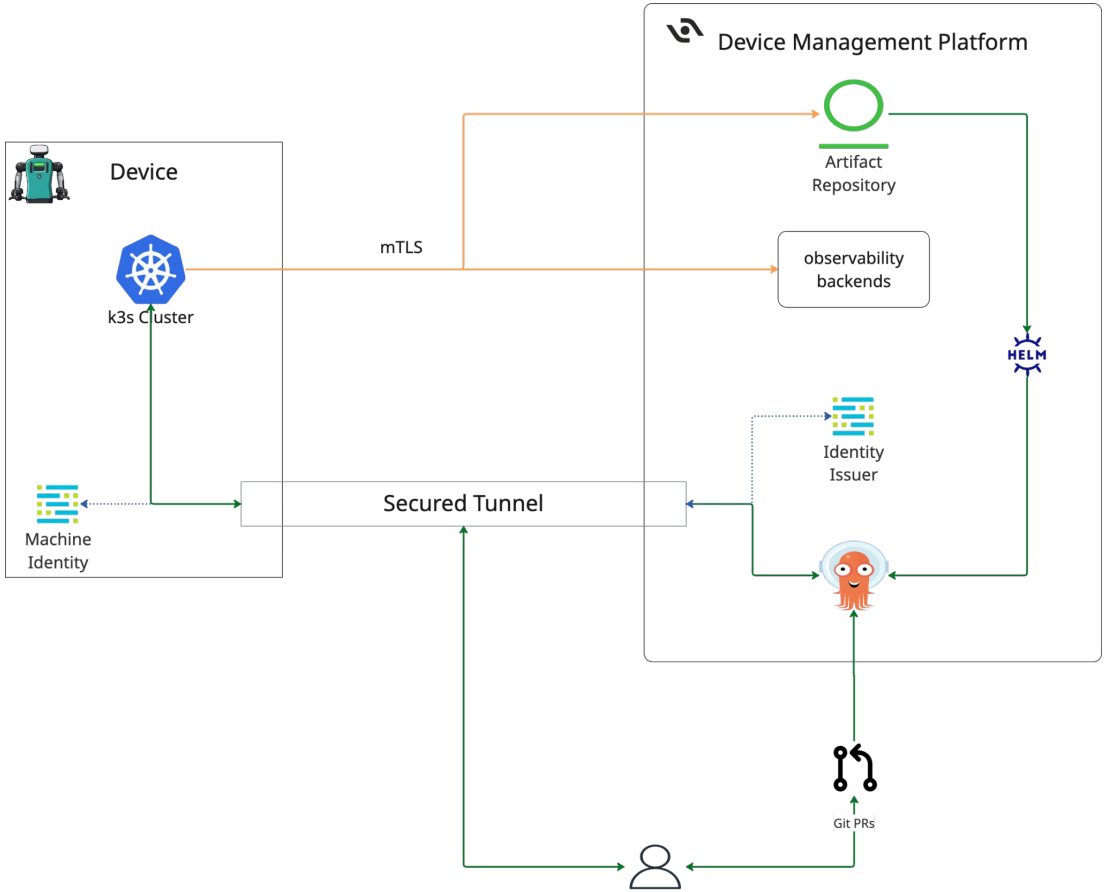
Configuration Management

- Over the Air
- Declarative
- Deployments
  - Software
  - Configurations
  - Models
  - Calibrations
  - Secrets and Credentials

Observability

- Full stack observability
- Metrics
- Logging
- Tracing
- Close to realtime

- K3S + EKS
- ArgoCD
- Helm Charts
- OCI artifacts
- Spiffe / Spire
- Teleport
- Open Telemetry
- FluentBit



Preventing in motion updates



K8s on a walking platform

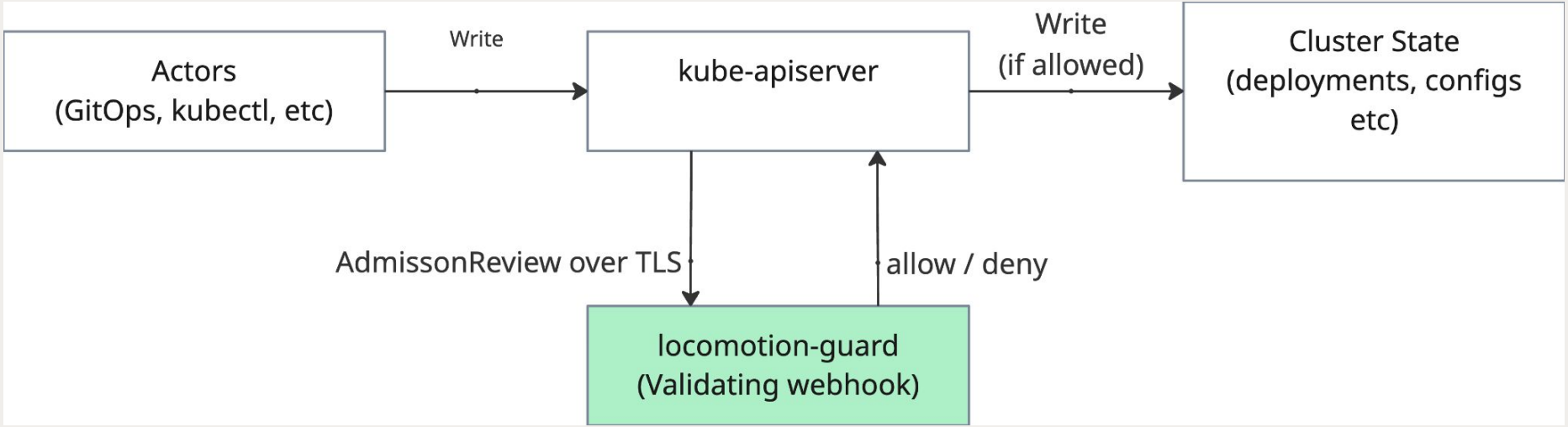
- Restart = physical risk: an uncontrolled stop of the control software mid-walk could contribute to a fall
- Control software handles termination itself, but even a clean stop abandons the task, and a forced kill skips that handling entirely
- Still worth it: declarative deploys, health checks, GitOps and a uniform fleet. Another layer of fall protection
- The challenge: teach the cluster when not to change things

Why sync windows are not enough

- While carefully defining and implementing a sync window helps ensure safe updates, certain scenarios require additional protection and flexibility.
- While primarily for development purposes, operator usage of `kubectl` also needs proper handling.
- Defense in depth: Maintain the schedule for planning while adding an infrastructure-level gate that functions independently of the schedule.

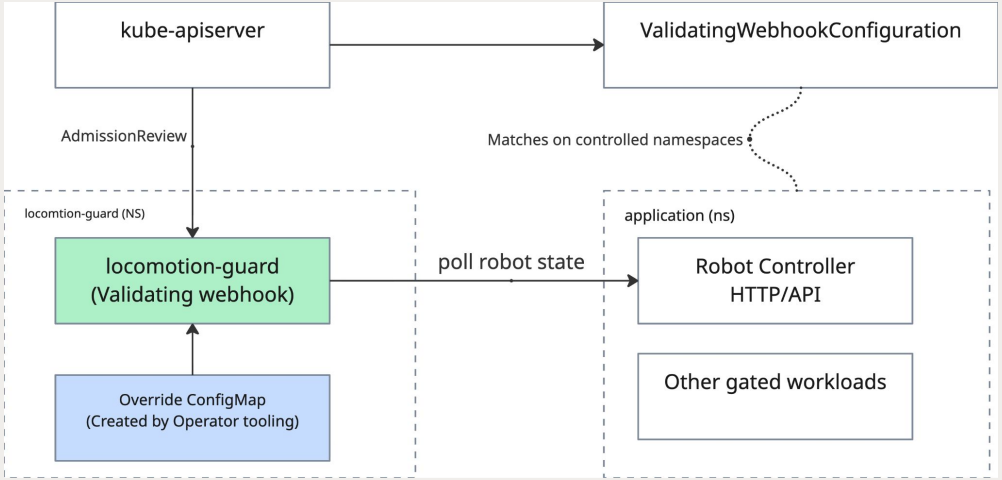


Adding on robot admission controller



Anything that writes to the API is gated, not just one tool.

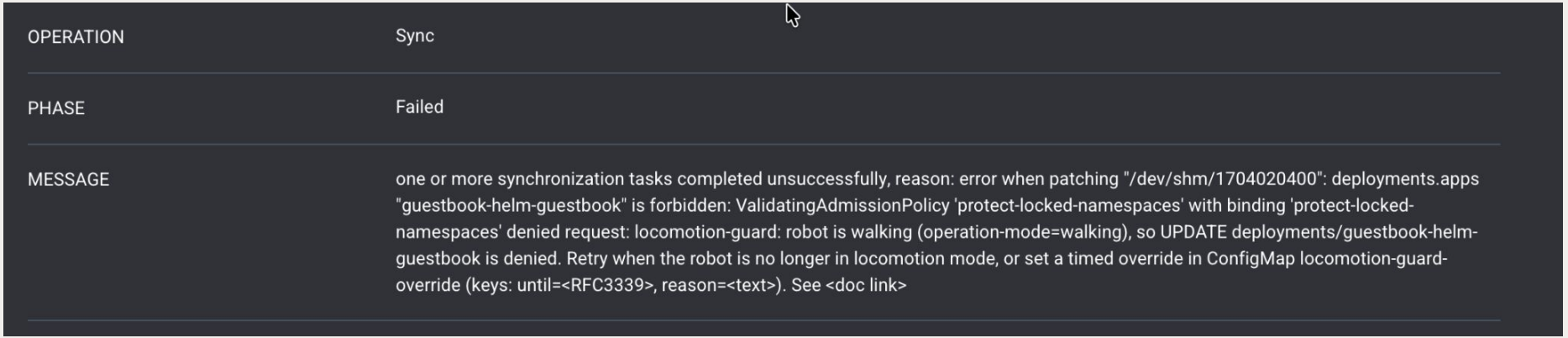
How the guard knows what the robot is doing



- **Poll and cache:** a background loop polls at 1/s and the admission path reads memory, so answers are near-instant even if the robot is busy
- **State expires:** an observation older than 5 s counts as unknown
- **Own namespace:** never gated, so the guard and its override stay reachable
- **Tiny footprint:** one Go binary with typed `admission/v1`
- **TLS from cert-manager**

What an operator sees.

```
$ kubectl -n locked-demo create configmap demo-cm --from-literal=a=b
error: failed to create configmap: configmaps "demo-cm" is forbidden:
ValidatingAdmissionPolicy 'protect-locked-namespaces' with binding
'protect-locked-namespaces' denied request: locomotion-guard: robot is walking
(operation-mode=walking), so CREATE configmaps/demo-cm is denied. Retry when the robo
is no longer in locomotion mode, or set a timed override in ConfigMap
locomotion-guard-override (keys: until=<RFC3339>, reason=<text>). See <doc link>.
```



OPERATION	Sync
PHASE	Failed
MESSAGE	one or more synchronization tasks completed unsuccessfully, reason: error when patching "/dev/shm/1704020400": deployments.apps "guestbook-helm-guestbook" is forbidden: ValidatingAdmissionPolicy 'protect-locked-namespaces' with binding 'protect-locked-namespaces' denied request: locomotion-guard: robot is walking (operation-mode=walking), so UPDATE deployments/guestbook-helm-guestbook is denied. Retry when the robot is no longer in locomotion mode, or set a timed override in ConfigMap locomotion-guard-override (keys: until=<RFC3339>, reason=<text>). See <doc link>

Break-glass

- Ability to override using a config map
  - Ability to override by creating/updating a ConfigMap.
    - Time bound expiration and text field with the rationale of the override config.

```
$ kubectl -n locomotion-guard create configmap locomotion-guard-override \
  --from-literal=until="$(date -u -d '+15 min' +%Y-%m-%dT%H:%M:%SZ)" \
  --from-literal=reason="hotfix TICKET-123"
```

Typical override flow

```
operator@workstation:~$
```

Careers - We Are Hiring!

OPEN POSITIONS - https://www.agilityrobotics.com/careers	
Senior Staff Hardware Reliability Engineer	Hybrid - Any Office (Fremont, CA, Salem, OR, or Pittsburgh, PA)
Senior Data Engineer	Hybrid / Remote
Senior Software Engineer - Mobile & Web Apps Developer	Hybrid / Remote
Senior/Staff Build & Release Engineer(s)	Hybrid / Remote
Senior Robotics Software Engineer(s)	Hybrid / Remote
Director of AI	Hybrid / Remote
Senior Manager, Hardware Abstraction	Hybrid - Any Office (Fremont, CA, Salem, OR, or Pittsburgh, PA)





Session Feedback