ZAL’GATE — One-Pager (Pilot A)

Controlled deviation release for production & quality teams — audit-ready, explainable, no autonomy.

Policy → Identity → Safeguard → Audit → Execute	No Autonomy	Process evidence (not employee tracking)	Target-scoped (Machine/Line IDs)

The problem we solve

In many factories, deviations are handled via informal decisions (phone calls, Excel, paper). That creates risk: unclear responsibility, missing evidence, inconsistent rules, and weak audit trails.

Pilot A: Deviation release (Abweichungsfreigabe) & special release (Sonderfreigabe)

- Use case: A part/process deviates from spec; a human decides whether it is released under controlled conditions.
- Actors: Operator (request), Team Lead (release), optional QA (review).
- Scope controls: Each request is bound to a targetId (e.g., 3A) and a timebox / scope unit (e.g., 30 min or 20 parts).
- Outputs: decision record, reasons, evidence attachments, exportable audit trail.

Important: not employee surveillance

ZAL’GATE monitors the process state, not the employee. No performance tracking, no behavior monitoring, no continuous worker surveillance. We store only what is needed to prove a governance decision (WHO/WHAT/WHY/WHEN, scoped to a targetId).

How it works (governance chain)

Policy	Identity	Safeguard	Audit	Execute (Release)

- Policy: rules define what is allowed/blocked (e.g., release types, required evidence, time limits).
- Identity: action is bound to a real role/person (PIN / biometric / login).
- Safeguard: targetId-scoped timebox/override logic (no global overrides).
- Audit: immutable event trail: request, approve/deny, reasons, attachments, export.
- Execute: records the release decision; optional integrations later (outbound-only, gated).

Deliverables (what you get)

- Pilot Spec v0 (1–2 pages): scope, roles, targetId scheme, timebox rules, evidence requirements.
- Demo flow: request → review → decision → audit view → export (PDF/JSON).
- Explainability view: clear reason for Allow/Block and who approved.
- Audit schema: event types + fields (timestamp, persona/role, command, targetId, decision, reasons).

Typical timeline

Week 1	Define Pilot Spec v0, data fields, roles, targetId naming; confirm “not surveillance”
Week 2	Implement gated flow + audit + export; validate with 2–3 real scenarios.
Week 3–4	Hardening: edge cases, UI polish, reporting package for stakeholders / compliance.

Contact

Patrick Walker — ZAL’THERA
contact@zalthera.de · zalthera.de

Version: v1.0 · Generated: 2026-02-19