



TITLE ALN-RUST-BCI-NEUROMORPHIC-WORKFLOW-QPU-DATASHARD

VERSION 1.0.0

FORMAT ALN-NANOBIN-V3

SCOPE biomech_devices, bci_eeg_mci, neuromorphic_organic, smart_city_nodes

=====

=====

SECTION 1: REPO AND PIPELINE ROOT

=====

=====

REPO,PATH,TYPE,DESCRIPTION

[infra-bio-systems](#),git@github.com:Org/infra-bio-systems.git,monorepo,"Rust+ALN infra for biomechanical, BCI/EEG, neuromorphic, smart-city nodes"[file:14]

infra-bio-systems/crates/devices,rust_workspace,"Device abstractions, safety guards, hardware mappings"[file:14]

infra-bio-systems/crates/protocols,rust_workspace,"Low-level transport, websocket/dev-tunnel abstractions (sanitized)"[file:14]

infra-bio-systems/crates/metrics,rust_workspace,"Prometheus metrics and logging adapters" [web:6]

infra-bio-systems/crates/research_manifest,rust_workspace,"Machine-readable research manifest (JSON-LD/ALN)"[file:13]

infra-bio-systems/aln_specs,aln_root,"ALN invariant design specs for all device classes"[file:12]

infra-bio-systems/.github/workflows,ci_root,"CI/CD pipelines for codegen, tests, doc, security scan"[file:12]

=====

=====

SECTION 2: DAILY WORKFLOW STAGES (HIGH-LEVEL)

=====

=====

DAILY_STAGE,ORDER,OWNER,INPUT,OUTPUT,DESCRIPTION

model_in_ALN,1,research_eng,"aln_specs/.aln","validated ALN invariants","Update device/biophysics & signal invariants in ALN; run static validation"[file:12][file:13]

codegen_rust_guards,2,platform_eng,"aln_specs + templates","crates/devices/_guard.rs","Generate Rust guard modules from ALN invariants (type-safe, no unsafe)"[file:12]

codegen_metrics,3,platform_eng,"aln_specs + metrics templates","crates/metrics/_metrics.rs","Generate Prometheus metric definitions per device/signal family"[web:6]

update_research_manifest,4,research_ops,"new papers, regs, lab results","research_manifest/.json","Append machine-readable research evidence and jurisdiction metadata"[file:13][file:14]

run_ci_pipeline,5,ci_bot,"git push + PR","CI status, artifacts","Build, test, lint, security scan, docs, manifest checks"[file:12]

deploy_dev_nodes,6,devops,"green CI + feature flags","k8s helm releases","Rollout to dev smart-city clusters and lab rigs"[file:14]

observability_review,7,site_reliability,"Prometheus, logs, traces","tuning PRs","Daily review of metrics/alerts for biomech & BCI infra"[web:6]

compliance_sync,8,compliance_officer,"regs updates","aln_specs/compliance.aln","Update legal/reg constraints, export for audits"[file:13][file:14]

=====

=====

SECTION 3: ALN INVARIANT DESIGN FOR DEVICES/SIGNALS

=====

=====

ALN_SPEC,PATH,KIND,DESCRIPTION

aln_specs/dev_biomech_joint_v1.aln,device,"Torque/position-sensing exoskeleton joint with safety envelopes"[file:13]

aln_specs/dev_bci_16ch_eeg_v1.aln,device,"16-channel EEG/BCI front-end (e.g. Galea/OpenBCI-class), 250–1kHz"[file:1]

aln_specs/dev_neuromorphic_spikearray_v1.aln,device,"Event-based spike array interfacing to CIM / in-memory neuromorphic cores"[file:13][file:14]

aln_specs/dev_organic_interface_v1.aln,device,"Organic/biomaterial interface abstraction (no

implant specifics)"[file:13]
aln_specs/sig_eeg_band_power_v1.aln,signal,"Band-power/phase features derived from BCI/EEG
for control/neuromod"[file:13]
aln_specs/sig_mech_load_profile_v1.aln,signal,"Biomech joint load trajectory, jerk/impulse
constraints"[file:13]
aln_specs/sig_spike_rate_map_v1.aln,signal,"Neuromorphic firing-rate map and sparsity
invariants"[file:14]

ALN_DEVICE_INVARIANT,ID,FIELD,TYPE,CONSTRAINT,EXPR,COMMENT
dev_bci_16ch_eeg_v1,sampling_rate_hz,u32,range,"250 <= x <= 4000","Supported sample rates
for lab+clinical systems"[file:1]
dev_bci_16ch_eeg_v1,channel_count,u8,fixed,"x == 16","Fixed 16-channel design"[file:1]
dev_bci_16ch_eeg_v1,input_range_v,f32,range,"-0.5 <= x <= 0.5","Safety envelope for front-end
differential pairs"[file:1]
dev_bci_16ch_eeg_v1,dc_offset_mv,f32,abs_max,"|x| <= 50.0","Reject out-of-spec drifts at
input"[file:1]
dev_biomech_joint_v1,torque_nm,f32,range,"0.0 <= x <= 120.0","Per-joint safe-assist torque cap
(depends on joint type)"[file:13]
dev_biomech_joint_v1,position_deg,f32,range,"-45.0 <= x <= 135.0","Joint ROM guard vs
biomech model"[file:13]
dev_neuromorphic_spikearray_v1,spike_rate_hz,f32,range,"0.0 <= x <= 500.0","Expected firing
rate envelope for neuromorphic edge nodes"[file:14]
dev_neuromorphic_spikearray_v1,firing_sparsity,f32,range,"0.0 <= x <= 0.2","Encodes target
sparsity for energy-efficient coding"[file:14]

=====

=====

SECTION 4: RUST GUARD CODE GENERATION CONTRACT

=====

=====

RUST_GUARD_TEMPLATE,ID,DEST_CRATE,DEST_PATH,DESCRIPTION
template_numeric_guard_v1,dev_bci_16ch_eeg_v1,crates/devices,"src/bci/eeg16/guards.rs","Gen
erate typed guards for EEG16 parameters"[file:12]
template_numeric_guard_v1,dev_biomech_joint_v1,crates/devices,"src/biomech/joint/guards.rs","
Joint torque/position guard API"[file:13]
template_numeric_guard_v1,dev_neuromorphic_spikearray_v1,crates/devices,"src/neuromorphic/
spikearray/guards.rs","Spike rate/sparsity guards"[file:14]

RUST_GUARD_API,DEVICE_ID,FN_NAME,INPUT_TYPE,ERROR_TYPE,SEMANTIC
dev_bci_16ch_eeg_v1,guard_sampling_rate,"u32","EegConfigError","Reject non-supported EEG
sampling rates"

```
dev_bci_16ch_eeg_v1.guard_dc_offset,"f32","EegSignalError","Fail fast on DC offset drift"
dev_biomech_joint_v1.guard_torque,"f32","JointSafetyError","Prevent unsafe assist torque"
dev_biomech_joint_v1.guard_position,"f32","JointSafetyError","Prevent ROM violations"
dev_neuromorphic_spikearray_v1.guard_spike_rate,"f32","NeuromorphicRangeError","Catch
runaway firing regimes"
dev_neuromorphic_spikearray_v1.guard_sparsity,"f32","NeuromorphicRangeError","Maintain
encoding sparsity bounds"
```

```
RUST GUARD CODEGEN_TASK, ID, TOOL, COMMAND, CI_STAGE
cg_guard_bci16, aln_codegen, "aln codegen rust-guards --spec
aln_specs/dev_bci_16ch_eeg_v1.aln --out
crates/devices/src/bci/eeg16/guards.rs", "generate_rust_guards"
cg_guard_biomech_joint, aln_codegen, "aln codegen rust-guards --spec
aln_specs/dev_biomech_joint_v1.aln --out
crates/devices/src/biomech/joint/guards.rs", "generate_rust_guards"
cg_guard_neuromorphic_spike, aln_codegen, "aln codegen rust-guards --spec
aln_specs/dev_neuromorphic_spikearray_v1.aln --out
crates/devices/src/neuromorphic/spikearray/guards.rs", "generate_rust_guards"
```

=====

=====

SECTION 5: PROMETHEUS METRICS DEFINITION (MACHINE-READABLE)

=====

=====

```
PROM_METRIC, ID, NAME, TYPE, LABELS, HELP
bci_eeg_ingest_rate, bci_eeg_ingest_rate, "bci_eeg_ingest_rate_hz", "gauge", "device_id,site_id", "Ef
fective EEG samples/sec at ingest" [file:1][web:6]
bci_eeg_guard_failures, bci_eeg_guard_failures, "bci_eeg_guard_failures_total", "counter", "device_i
d,reason", "Total EEG guard violations" [file:1]
biomech_joint_torque, bj_torque, "biomech_joint_torque_nm", "gauge", "joint_id,side", "Current
assist torque" [file:13]
biomech_guard_failures, bj_guard, "biomech_joint_guard_failures_total", "counter", "joint_id,reason",
"Joint guard violations (ROM/torque)" [file:13]
neuromorphic_spike_rate, neu_spike, "neuromorphic_spike_rate_hz", "histogram", "node_id,layer",
"Spike rate distribution per neuromorphic node" [file:14]
neuromorphic_energy_per_op, neu_energy, "neuromorphic_energy_per_op_j", "gauge", "node_id,lay
er", "Estimated Joules/op from CIM back-end" [web:6][file:14]
organic_interface_link_quality, org_lq, "organic_interface_link_quality", "gauge", "node_id,medium",
"Quality index for organic/biological interface channel" [file:13]
pipeline_ci_status, ci_status, "infra_ci_status", "gauge", "pipeline,branch", "Latest CI status per
branch (0=fail,1=pass)" [file:12][web:6]
```

PROM_RUST_EXPORTER,CRATE,PATH,ENDPOINT,DESCRIPTION
crates/metrics,"src(exports/prometheus.rs","/metrics","HTTP exporter integrated with existing
infra metrics endpoint"[web:6]
crates/metrics,"src(exports/grpc_bridge.rs","grpc://metrics-bus:9090","Optional bridge into
central metrics bus"[web:6]

=====

=====

SECTION 6: GITHUB-READY CRATES + TEST STRATEGY

=====

=====

RUST_CRATE,NAME,PATH,EDITION,FEATURES,DESCRIPTION
devices,infra_bio_devices,"crates/devices","2021","bci,biomech,neuromorphic,organic","Unified
device+signal guard and type layer"[file:12][file:13]
protocols,infra_bio_protocols,"crates/protocols","2021","websocket,tcp,dev_tunnel_sanitized","S
anitized transport abstractions for virtual nodes"[file:12]
metrics,infra_bio_metrics,"crates/metrics","2021","prometheus,logging","Shared observability
primitives"[web:6]
research_manifest,infra_bio_research_manifest,"crates/research_manifest","2021","jsonld,aln_parse","Machine-readable research/regs manifest"[file:13][file:14]

RUST_TEST_SUITE,CRATE,TYPE,COMMAND,FOCUS
devices,unit,"cargo test -p infra_bio_devices","Guard invariants, conversions, error paths"
devices,property,"cargo test -p infra_bio_devices --features proptest","Randomized numeric
boundary tests"
metrics,unit,"cargo test -p infra_bio_metrics","Metric registration and label sets"
research_manifest,unit,"cargo test -p infra_bio_research_manifest","Schema evolution and
backward compatibility"
workspace,ci,"cargo test --workspace --all-features","Daily full suite in CI"

=====

=====

SECTION 7: MACHINE-READABLE RESEARCH MANIFEST (CI-FRIENDLY)

=====

=====

RESEARCH_MANIFEST_DOC, ID, PATH, FORMAT, DESCRIPTION

rm_bci_energy_v1, crates/research_manifest/data/bci_energy_v1.json, jsonId, "Energy savings and safety bounds for BCI/EEG front-ends" [file:13]
rm_neuromorphic_energy_v1, crates/research_manifest/data/neuromorphic_energy_v1.json, jsonId, "Energy-accuracy trade-offs for neuromorphic CIM systems" [file:14]
rm_organic_interfaces_v1, crates/research_manifest/data/organic_interfaces_v1.json, jsonId, "Organic/biomaterial interface constraints and safety" [file:13]
rm_neural_data_reg_v1, crates/research_manifest/data/neural_data_reg_v1.json, jsonId, "Neural data privacy and jurisdiction map (US/EU/etc.)" [file:14]

RESEARCH_ENTRY, MANIFEST_ID, ENTRY_ID, DOMAIN, KEY, CALC, SCIENTIFIC_FACT, LEGAL_TERMS, GEO

rm_bci_energy_v1, bci_energy_001, "bci_eeg", "E_saving", "E_silicon - E_bio", "Energy saving model: E_saving = E_silicon - E_bio; compute both from lab measurements; same calculation uses device power logs over task duration and subtracts per condition to get net savings.", "Natural biomaterials and optimized spike-based encoding reduce BCI front-end power use by up to one order of magnitude while maintaining signal quality.", "All deployments must comply with applicable medical-device and data-protection law, including ISO 13485 device safety management, HIPAA-equivalent health data safeguards, and GDPR-class rules for biometric identifiers in cross-border research scenarios.", "Sweden; USA; Germany; China; South Korea" [file:13]

rm_neuromorphic_energy_v1, neuromorphic_001, "neuromorphic", "P_symbol", "P_symbol = N_spikes * E_spike", "Using neuromorphic receiver models, dynamic power per symbol is P_symbol = N_spikes × E_spike; by capping N_spikes and engineering low E_spike devices, energy per symbol is demonstrably reduced vs conventional digital links.", "Biologically plausible spiking models and in-memory computing reduce energy per operation while preserving accuracy for perception and control workloads.", "Deployments must integrate energy-efficiency standards, transparent AI documentation, and sector-specific regulations for critical infrastructure where neuromorphic edge nodes operate.", "Manchester (UK); Zurich (Switzerland); Singapore; Helsinki (Finland); Toronto (Canada)" [file:13] [file:14]

rm_organic_interfaces_v1, organic_001, "organic_interfaces", "H_security", "H = - $\sum p_i \log_2(p_i)$ ", "Security entropy H = - $\sum p_i \log_2(p_i)$ models uncertainty over attack states; by adding independent organic-layer security channels the entropy of successful compromise increases, hardening tamper resistance.", "Organic and bio-compatible layers provide physical and chemical tamper resistance beyond rigid silicon-only stacks, improving privacy and device integrity.", "Interfaces must uphold medical ethics, environmental regulations on biomaterials, and cross-border liability rules for bio-hybrid systems.", "San Francisco (USA); London (UK); Stockholm (Sweden); Sydney (Australia); Munich (Germany)" [file:13]

rm_neural_data_reg_v1, neural_reg_001, "neural_data", "I_Adopt", "I = T_Adopted / T_total", "Technology adoption impact metric I = T_Adopted / T_total expresses fraction of deployments using neurorights-compliant pipelines; compute using registry totals per jurisdiction.", "Emerging neuro-data laws and neurorights proposals explicitly treat neural signals as sensitive data requiring heightened protection and transparency, especially for public-

infrastructure uses." , "Neural data handling must align with neural-specific privacy acts, broader AI governance rules, and existing health-data laws in every participating jurisdiction with clear documentation and audit trails." , "Colorado (USA); California (USA); Barcelona (Spain); Zurich (Switzerland); Kyoto (Japan)" [file:14]

=====

=====

SECTION 8: TEN DAILY RESEARCH ACTIONS (FUTURE-TECH/SYSTEMS)

=====

=====

RESEARCH_ACTION, ID, DOMAIN, DESCRIPTION, MATH_MODEL, SCIENTIFIC_GROUNDING, LEGAL_TERMS, GEO_EVIDENCE

1,ra_bci_power_profile, "BCI/EEG", "Update BCI/EEG device power profiles from latest lab runs and regenerate rm_bci_energy_v1 entries.", "P_avg = (1/T) ∫_0^T P(t) dt; compute P_avg per mode (idle, capture, stream) from logged power traces, then compare across firmware builds for regression detection.", "Recent intracortical and surface BCI work shows that power-aware front-end design and duty-cycled streaming can cut consumption without degrading decoding accuracy.", "Raw and derived power logs must be treated as regulated technical data; sharing outside the lab is conditioned on contract terms, confidentiality provisions, and export-control screening for dual-use circuits.", "Cambridge (UK); Zurich (Switzerland); Boston (USA); Seoul (South Korea); Tokyo (Japan)" [file:13]

2,ra_neuro_energy, "Neuromorphic", "Refresh neuromorphic CIM energy-per-op curves using current chip-run data and feed rm_neuromorphic_energy_v1.", "E_op = E_total / N_ops; compute E_total from on-board energy counters and N_ops from workload logs; repeat per benchmark and store in manifest.", "Recent CIM and resistive-memory research reports 20–30% latency gains with around 30% energy reduction vs traditional architectures for pattern-recognition tasks.", "Energy benchmarking must respect vendor NDAs, critical-infrastructure security policies, and non-disclosure of sensitive layout or process information in public manifests.", "Dresden (Germany); Austin (USA); Hsinchu (Taiwan); Grenoble (France); Neuromorphic Sweden sites (Sweden)" [file:14]

3,ra_biomech_safety, "Biomechanical devices", "Calibrate biomechanical safety envelopes (torque/ROM) against the latest gait and rehab studies and sync into dev_biomech_joint_v1 invariants.", "τ_safe(t) <= τ_max(joint,type); compute worst-case torque from motion-capture and EMG-derived loads and set τ_max with safety margin factor m>1.", "Clinical biomechanics literature provides joint-specific safe torque and ROM ranges for assistive devices under varied activities (walking, sit-to-stand, stairs).", "Safety constraints must comply with rehabilitation standards, worker-safety rules, and medical-device risk classifications when systems approach therapeutic domains.", "Boston (USA); Munich (Germany); Rotterdam (Netherlands); Kyoto (Japan); Singapore" [file:13]

4,ra_organic_channels, "Organic/BCI", "Characterize organic/biomaterial interface link quality and map into organic_interface_link_quality metric and invariants.", "LQ = SNR / (1 + BER); estimate

signal-to-noise ratio SNR and bit-error rate BER from experiments and compute LQ as a compressed channel-quality index." "Neuromorphic and organic-computing research shows stable charge-transport and signal transduction through tailored biomaterials suitable for interfacing sensors to computation units." "Experiments must document material provenance, biosafety classifications, and environmental compliance for production deployment beyond laboratory settings." "Stockholm (Sweden); Zurich (Switzerland); San Diego (USA); Shenzhen (China); Melbourne (Australia)" [file:13][file:14]

5,ra_neural_privacy,"Neural data","Ingest new neurorights/legal updates into rm_neural_data_reg_v1 and verify pipelines against them." "R_risk = $\sum r_i p_i$ over neural-data hazards; maintain R_risk below jurisdiction-specific thresholds by enforcing stronger controls as new laws pass." "Recent analyses confirm that existing privacy laws only partially cover neural data, prompting specific neural-data privacy statutes in several regions." "All CI and runtime systems must tag neural payloads as sensitive, apply least-privilege access controls, and provide traceable consent and revocation mechanisms." "Denver (USA); Sacramento (USA); Brussels (Belgium); Santiago (Chile); Barcelona (Spain)" [file:14]

6,ra_smart_city_nodes,"Smart-city nodes","Update smart-city node ALN specs with current neuromorphic/BCI sensor layouts and NIST-style control mappings (sanitized)." "A_node = f(sensors, links, policies); define mapping from sensor set and communication links into allowed data flows respecting access-control matrices." "Smart-city digital twin deployments already integrate biosensing and traffic/environmental sensors, demonstrating the feasibility of large-scale observability meshes." "Asset and data flows must align with urban data-governance frameworks, cybersecurity standards, and public-health regulations for AI-enhanced infrastructure." "NEOM region (Saudi Arabia); Singapore; Barcelona (Spain); Atlanta (USA); Oslo (Norway)" [web:6][file:14]

7,ra_metrics_valid,"Metrics/observability","Verify that all Prometheus metrics match current device IDs, labels, and retention policies and update metrics schema if needed." "For each metric, validate dimensional consistency e.g. samples/sec \times seconds = count and ensure histograms cover observed ranges via quantile checks." "Production monitoring for neuromorphic and BCI systems shows that deployment stability depends on accurate, well-bounded metrics with clear units." "Telemetry must respect log-retention rules, data-minimization principles, and jurisdiction-dependent consent when metrics embed user or patient identifiers." "Frankfurt (Germany); Phoenix (USA); Dublin (Ireland); Tokyo (Japan); Sydney (Australia)" [web:6]

8,ra_ai_routing,"Agentic routing","Refine ALN specs for AI routing of experiments and CI workflows to Mistral and Qwen backends over sanitized dev tunnels." "Latency_model: L_total = L_net + L_queue + L_model; measure each term separately to ensure SLOs for CI and research-assistant tasks." "Modern AI backends can orchestrate code generation, documentation, and manifest updates when integrated through secure APIs and audited pipelines." "All AI orchestrations must record intent, inputs, and outputs for auditability, avoid uncontrolled data exfiltration, and follow AI-governance guidelines in each region." "Paris (France); Beijing (China); London (UK); Montreal (Canada); Singapore" [file:14]

9,ra_manifest_ci,"CI integration","Ensure research manifest checks run in CI, blocking merges when evidence or jurisdiction mappings are stale." "Let age_days = (now - last_update)/1d; enforce age_days <= threshold for critical entries and fail pipeline when exceeded." "Continuous manifest validation keeps code aligned with up-to-date science, safety evidence, and legal constraints rather than freezing at initial assumptions." "CI policies must be documented,

transparent to collaborators, and consistent with partner contractual obligations and regulatory expectations for traceability." "Seattle (USA); Berlin (Germany); Toronto (Canada); Tel Aviv (Israel); Zurich (Switzerland)" [file:12][file:14]
10,ra_bci_gaming,"Gaming & XR","Update BCI/neuromorphic profiles specific to next-gen gaming and XR smart-city experiences (latency budgets, comfort, fairness).","T_budget = T_sensor + T_decode + T_render + T_network; ensure T_budget <= T_max for comfortable real-time interaction (e.g. tens of ms)." "Work in gaming and XR shows that end-to-end latency and jitter bounds determine usability of real-time control via biosignals and neuromorphic processing." "Gaming/XR uses of biosignals must still uphold neural-privacy and anti-discrimination standards, especially in public or shared smart-city spaces." "Los Angeles (USA); Helsinki (Finland); Tokyo (Japan); Barcelona (Spain); Vancouver (Canada)" [file:13][file:14]

=====

=====

SECTION 9: RUST-CENTRIC CI PIPELINE (MACHINE-READABLE)

=====

=====

CI_PIPELINE,NAME,PATH,TRIGGER,STAGES
github_actions_main,.github/workflows/ci.yml,"push,
pull_request","aln_validate,codegen_rust,build,test,lint,manifest_check,security_scan,package"

CI_STAGE_DEF,NAME,STEP_ORDER,COMMAND,DESCRIPTION
aln_validate,1,"aln validate aln_specs","Validate all ALN invariants for devices/signals/compliance" [file:12]
codegen_rust,2,"aln codegen rust-guards --all-specs","Generate/update Rust guard code from ALN" [file:12]
build,3,"cargo build --workspace --all-features","Build all crates" [file:12]
test,4,"cargo test --workspace --all-features","Run unit/integration/property tests"
lint,5,"cargo fmt --all && cargo clippy --all-targets --all-features -D warnings","Enforce style and linting"
manifest_check,6,"cargo test -p infra_bio_research_manifest && aln research check --manifest-path crates/research_manifest/data","Verify manifest schemas and staleness thresholds" [file:13]
[file:14]
security_scan,7,"cargo audit && cargo deny check","Scan for vulnerable dependencies"
package,8,"cargo publish --dry-run","Verify crates are publishable to registry"

SECTION 10: HEX OUTPUT CONQUERING ALL QUESTIONS

```
HEX_SUMMARY, ID, VALUE
workflow_hex, core, "0x414c4e2d4252494447452d5245553452d4e4555524f4d4f525048494
32d4253492d43495f574f524b464c4f573a20525553542b414c4e204749545f524f4f54204d4
f44454c5320464f522042494f4d4543482c204243492c204545472c204d43492c204e4555
524f4d4f52504849432c204f5247414e49432053594e5448454d532c2050524f4d455448455
553204d4554524943532c20474f562d5245414459204352415445532c2043492d534146452
04d414e4946455354532c20414e4420534d4152542d43495459204e4f4445532e"
```

TITLE infra-bio-systems-qpu-dashard.aln
VERSION 1.0.0
FORMAT ALN-NANOBIN-V3
SCOPE biomech_devices, bci_eeg_mci, neuromorphic_organic, smart_city_nodes[1]

```
# =====
# SECTION 0: QPU DATASHARD HEADER (RUST + ALN + NEUROMORPHIC CORE)
# =====
QPU_DATASHARD, ID, ARCH, DEST_PATH, PROFILE, DESCRIPTION
qpu_infranet_core, bio_neuro_rust_v1, "infra-bio-systems/qpu/dashards/qpu_infranet_core\_"

# =====
# SECTION 1: REPO AND PIPELINE ROOT (RUST+ALN)
# =====
REPO, PATH, TYPE, DESCRIPTION
infra-bio-systems, git@github.com:Org/infra-bio-systems.git, monorepo, "Rust+ALN infra for t
infra-bio-systems/crates/devices, rust_workspace, "Device abstractions, safety guards, harc
infra-bio-systems/crates/protocols, rust_workspace, "Sanitized transport (WS/TCP/dev-tunne]
infra-bio-systems/crates/metrics, rust_workspace, "Prometheus/OpenMetrics exporters for BC]
infra-bio-systems/crates/research_manifest, rust_workspace, "JSON-LD/ALN manifests for ene
infra-bio-systems/aln_specs, aln_root, "ALN device + signal + compliance invariants, CI-val
infra-bio-systems/.github/workflows, ci_root, "Rust+ALN CI (validation, codegen, tests, sec

# =====
# SECTION 2: DAILY WORKFLOW STAGES (EXECUTABLE PIPELINE)
# =====
DAILY_STAGE, ORDER, OWNER, INPUT, OUTPUT, DESCRIPTION
model_in_ALN, 1, research_eng, "aln_specs/*.aln", "validated_aln/", "Update device/biophysics
codegen_rust_guards, 2, platform_eng, "validated_aln + templates", "crates/devices/*_guards.i
codegen_metrics, 3, platform_eng, "validated_aln + metrics templates", "crates/metrics/*_met
update_research_manifest, 4, research_ops, "lab logs, new papers, legal bulletins", "crates/i
run_ci_pipeline, 5, ci_bot, "git push + PR", "CI artifacts, status badges", "Build, test, lint
deploy_dev_nodes, 6, devops, "green CI, signed artifacts", "k8s/helm releases", "Roll out to /
```

```

observability_review,7,site_reliability,"Prometheus, traces, logs","tuning PRs","Adjust compliance_sync,8,compliance_officer,"regs updates, neurorights tracker","aln_specs/comp

# =====
# SECTION 3: ALN INVARIANT DESIGN (DEVICES + SIGNALS)
# =====

ALN_SPEC,PATH,KIND,DESCRIPTION
aln_specs/dev_biomech_joint_v1.aln,device,"Torque/position-sensing exoskeleton joint with aln_specs/dev_bci_16ch_eeg_v1.aln,device,"16-channel EEG/BCI front-end (Galea/OpenBCI-cla aln_specs/dev_neuromorphic_spikearray_v1.aln,device,"Event-based spike array for neuromor aln_specs/dev_organic_interface_v1.aln,device,"Abstract organic/biomaterial interface (nc aln_specs/sig_eeg_band_power_v1.aln,signal,"Band-power/phase features for control and ne aln_specs/sig_mech_load_profile_v1.aln,signal,"Biomech joint load trajectories with jerk/ aln_specs/sig_spike_rate_map_v1.aln,signal,"Neuromorphic rate/sparsity maps for low-energ

ALN_DEVICE_INVARIANT,ID,FIELD,TYPE,CONSTRAINT,EXPR,COMMENT
dev_bci_16ch_eeg_v1,sampling_rate_hz,u32,range,"250 <= x <= 4000","Supported EEG sample r dev_bci_16ch_eeg_v1,channel_count,u8,fixed,"x == 16","Fixed channel count for simplified dev_bci_16ch_eeg_v1,input_range_v,f32,range,"-0.5 <= x <= 0.5","Differential input range dev_bci_16ch_eeg_v1,dc_offset_mv,f32,abs_max,"|x| <= 50.0","DC drift guard to protect use dev.biomech_joint_v1,torque_nm,f32,range,"0.0 <= x <= 120.0","Assistive torque cap per jo dev.biomech_joint_v1,position_deg,f32,range,"-45.0 <= x <= 135.0","Range-of-motion guard dev.neuromorphic_spikearray_v1,spike_rate_hz,f32,range,"0.0 <= x <= 500.0","Expected fir dev.neuromorphic_spikearray_v1,firing_sparsity,f32,range,"0.0 <= x <= 0.2","Target sparsi

# =====
# SECTION 4: RUST GUARD CODEGEN CONTRACT (ALN → RUST)
# =====

RUST_GUARD_TEMPLATE,ID,DEST_CRATE,DEST_PATH,DESCRIPTION
template_numeric_guard_v1,dev_bci_16ch_eeg_v1,crates/devices,"src/bci/eeg16/guards.rs",1 template_numeric_guard_v1,dev_biomech_joint_v1,crates/devices,"src/biomech/joint/guards.rs",1 template_numeric_guard_v1,dev_neuromorphic_spikearray_v1,crates/devices,"src/neuromorphic

RUST_GUARD_API,DEVICE_ID,FN_NAME,INPUT_TYPE,ERROR_TYPE,SEMANTIC
dev_bci_16ch_eeg_v1,guard_sampling_rate,"u32","EegConfigError","Reject unsupported EEG sa dev_bci_16ch_eeg_v1,guard_dc_offset,"f32","EegSignalError","Fail fast on DC drift at inge dev.biomech_joint_v1,guard_torque,"f32","JointSafetyError","Clamp unsafe assist torque co dev.biomech_joint_v1,guard_position,"f32","JointSafetyError","Reject ROM violations" dev.neuromorphic_spikearray_v1,guard_spike_rate,"f32","NeuromorphicRangeError","Detect ri dev.neuromorphic_spikearray_v1,guard_sparsity,"f32","NeuromorphicRangeError","Keep sparsi

RUST_GUARD_CODEGEN_TASK,ID,TOOL,COMMAND,CI_STAGE
cg_guard_bci16,aln_codegen,"aln codegen rust-guards --spec aln_specs/dev_bci_16ch_eeg_v1. cg_guard_biomech_joint,aln_codegen,"aln codegen rust-guards --spec aln_specs/dev_biomech_ cg_guard_neuromorphic_spike,aln_codegen,"aln codegen rust-guards --spec aln_specs/dev_neu

# =====
# SECTION 5: PROMETHEUS METRICS (RUST EXPORTER BINDING)
# =====

PROM_METRIC,ID,NAME,TYPE,LABELS,HELP
bci_eeg_ingest_rate,bci_eeg_ingest_rate,"bci_eeg_ingest_rate_hz","gauge","device_id,site_ bci_eeg_guard_failures,bci_eeg_guard_failures,"bci_eeg_guard_failures_total","counter","c biomech_joint_torque,bj_torque,"biomech_joint_torque_nm","gauge","joint_id,side","Current biomech_guard_failures,bj_guard,"biomech_joint_guard_failures_total","counter","joint_id,neuromorphic_spike_rate,neu_spike,"neuromorphic_spike_rate_hz","histogram","node_id,layer_ neuromorphic_energy_per_op,neu_energy,"neuromorphic_energy_per_op_j","gauge","node_id,lay
```

```

organic_interface_link_quality,org_lq,"organic_interface_link_quality","gauge","node_id,",
pipeline_ci_status,ci_status,"infra_ci_status","gauge","pipeline,branch","Latest CI statu

PROM_RUST_EXPORTER,CRATE,PATH,ENDPOINT,DESCRIPTION
crates/metrics,"src(exports/prometheus.rs","/metrics","HTTP exporter merged into existing
crates/metrics,"src(exports/grpc_bridge.rs","grpc://metrics-bus:9090","Optional gRPC bric

# =====
# SECTION 6: GITHUB-READY CRATES + TESTS
# =====

RUST_CRATE,NAME,PATH,EDITION,FEATURES,DESCRIPTION
devices,infra_bio_devices,"crates/devices","2021","bci,biomech,neuromorphic,organic","Uni
protocols,infra_bio_protocols,"crates/protocols","2021","websocket,tcp,dev_tunnel_sanitiz
metrics,infra_bio_metrics,"crates/metrics","2021","prometheus,logging","Shared observabil
research_manifest,infra_bio_research_manifest,"crates/research_manifest","2021","jsonld,&

RUST_TEST_SUITE,CRATE,TYPE,COMMAND,FOCUS
devices,unit,"cargo test -p infra_bio_devices","Invariants, conversions, guard error path
devices,property,"cargo test -p infra_bio_devices --features proptest","Boundary/property
metrics,unit,"cargo test -p infra_bio_metrics","Metric registration, label cardinality, e
research_manifest,unit,"cargo test -p infra_bio_research_manifest","Schema evolution and
workspace,ci,"cargo test --workspace --all-features","Daily full-suite CI regression"

# =====
# SECTION 7: RESEARCH MANIFEST (MATH + SCIENCE + LAW + GEO)
# =====

RESEARCH_MANIFEST_DOC,ID,PATH,FORMAT,DESCRIPTION
rm_bci_energy_v1,crates/research_manifest/data/bci_energy_v1.json,jsonld,"Energy and safe
rm_neuromorphic_energy_v1,crates/research_manifest/data/neuromorphic_energy_v1.json,jsonl
rm_organic_interfaces_v1,crates/research_manifest/data/organic_interfaces_v1.json,jsonld,
rm_neural_data_reg_v1,crates/research_manifest/data/neural_data_reg_v1.json,jsonld,"Neur

RESEARCH_ENTRY,MANIFEST_ID,ENTRY_ID,DOMAIN,KEY,CALC,SCIENTIFIC_FACT,LEGAL_TERMS,GEO
rm_bci_energy_v1,bci_energy_001,"bci_eeg","E_saving","E_saving = E_silicon - E_bio","Comp
rm_neuromorphic_energy_v1,neuromorphic_001,"neuromorphic","P_symbol","P_symbol = N_spikes
rm_organic_interfaces_v1,organic_001,"organic_interfaces","H_security","H = -Σ p_i log2(|
rm_neural_data_reg_v1,neural_reg_001,"neural_data","I_adopt","I = T_adopted / T_total","/"

# =====
# SECTION 8: TEN DAILY RESEARCH ACTIONS (NON-FICTIVE FUTURE-TECH)
# =====

RESEARCH_ACTION,ID,DOMAIN,DESCRIPTION,MATH_MODEL,SCIENTIFIC_GROUNDING,LEGAL_TERMS,GEO_EV
1,ra_bci_power_profile,"BCI/EEG","Refresh BCI/EEG power profiles from latest lab runs and
2,ra_neuro_energy,"Neuromorphic","Update neuromorphic CIM energy-per-op curves from curre
3,ra_biomech_safety,"Biomechanical devices","Calibrate torque and ROM envelopes against i
4,ra_organic_channels,"Organic/BCI","Characterize organic interface link quality and driv
5,ra_neural_privacy,"Neural data","Monitor neurorights/legal changes and update rm_neural
6,ra_smart_city_nodes,"Smart-city nodes","Extend smart-city ALN specs with neuromorphic/E
7,ra_metrics_valid,"Metrics/observability","Validate all metrics against device IDs, labe
8,ra_ai_routing,"Agentic routing","Refine ALN specs for routing CI and experiment jobs to
9,ra_manifest_ci,"CI integration","Treat manifest checks as first-class CI gates that blo
10,ra_bci_gaming,"Gaming & XR","Tune BCI/neuromorphic profiles for next-gen gaming and XF

# =====
# SECTION 9: RUST-CENTRIC CI PIPELINE (MACHINE READABLE)
# =====

```

```
CI_PIPELINE,NAME,PATH,TRIGGER,STAGES
github_actions_main,.github/workflows/ci.yml,"push,pull_request","aln_validate,codegen_rust

CI_STAGE_DEF,NAME,STEP_ORDER,COMMAND,DESCRIPTION
aln_validate,1,"aln validate aln_specs","Validate all device/signal/compliance ALN invariants"
codegen_rust,2,"aln codegen rust-guards --all-specs","Generate/update Rust guard code for build"
build,3,"cargo build --workspace --all-features","Build all crates with all features."
test,4,"cargo test --workspace --all-features","Run unit/integration/property tests over lint"
lint,5,"cargo fmt --all && cargo clippy --all-targets --all-features -D warnings","Enforce manifest_check,6,"cargo test -p infra_bio_research_manifest && aln research check --manifest"
security_scan,7,"cargo audit && cargo deny check","Scan dependencies for known vulnerabilities"
package,8,"cargo publish --dry-run","Verify crates are publishable to registry.

# =====
# SECTION 10: HEX OUTPUT CONQUERING ALL QUESTIONS
# =====

HEX_SUMMARY,ID,VALUE
workflow_hex,core,"0x414c4e2d4252494447452d4253492d4e4555524f4d4f52504849432d4243492d4249
```

Trivia-mode 10-answer bundle

1. Mathematical solution: $P_{avg} = \frac{1}{T} \int_0^T P(t) dt$; numerically, sample power at times t_i , sum $P(t_i)\Delta t$, then divide by total time T to reuse the same procedure across firmware profiles.
[1]
 2. Scientific grounding: Neuromorphic and event-driven BCI front-ends reduce energy per operation while maintaining decoding accuracy on perception and control tasks.[1]
 3. Legal terms (≥ 100 chars): All deployments must treat neural and biomechanical telemetry as sensitive regulated data, enforce strong authentication and encryption on all links, document lawful bases for processing, and provide audit-ready evidence of consent, minimization, and cross-border transfer safeguards.[1]
 4. Geographical evidence: Barcelona (Spain); Singapore; Zurich (Switzerland); Phoenix (USA); NEOM region (Saudi Arabia).[1]
 5. Rust+ALN guard stack ensures biomech torque, EEG ranges, and neuromorphic spike envelopes remain within lab-validated, regulation-aware safety bands.[1]
 6. Smart-city virtual nodes combine sanitized tunneling, neuromorphic mesh compute, and metrics-driven observability to support AU gaming and rehab experiences.[1]
 7. Research manifests encode energy, safety, and neurorights constraints as JSON-LD so CI can block merges when evidence or jurisdiction mappings become stale.[1]
 8. Agentic routing to Mistral and Qwen uses signed, policy-guarded tunnels where each job records intent, payload class, and outputs for downstream audits.[1]
 9. Combined BCI, biomech, organic, and neuromorphic metrics flow into Prometheus and dashboards to keep latency, comfort, and safety in-spec for XR and smart-city UX.[1]
 10. Hex answer (≥ 50 chars) conquering all questions:
0x5150555f4441544153484152445f494e4652414e45542d52455345415243482d4e45555

24f4d4f52504849432d4243492d42494f4d4543482d534d415254434954592d4e4f444
5532d525553545f414c4e5f434f5245