

Workspace Boundary Map v1

Evidence form for repository containment, workspace selection, package and dependency declarations, configuration ownership, mutation response, and a bounded human decision

Outcome. Identify the smallest failing architecture boundary, reject folder-proximity shortcuts, and issue ACCEPT, REJECT, or DEFER with residual risk.

Boundary. This map does not infer deployment topology or reproducible build closure from a shared repository, workspace membership, or a successful command.

Four architecture boundaries

1 REPOSITORY GRAPH Authority: contained paths and repository state Boundary: containment does not prove membership	2 WORKSPACE SELECTION Authority: workspace.json members and exclude Boundary: membership does not prove dependencies or deployment
3 PACKAGE AND DEPENDENCY GRAPH Authority: manifest identity and dependency fields Boundary: adjacency is not an edge	4 CONFIGURATION OWNERSHIP Authority: config content, declared owner, path, and scope Boundary: location may contradict ownership

Record identity

Learner	<div></div>	Reviewer	<div></div>
Case and attempt	<div></div>	Capture date	<div></div>
Workspace declaration	<div></div>	Final decision	ACCEPT / REJECT / DEFER

Safety and capture order

Baseline	Run python setup_fixture.py --case baseline, then python inspect_repository.py. Copy environment_identity and observation_boundary before changing the fixture.
Mutations	Reset before each mutation. Change membership, configuration location, and dependency declaration in separate runs; record changed and stable claims.
Safety boundary	Use only the supplied local fixture. No network, credentials, privileges, broad environment dump, deployment action, or host-destructive operation is required.

Path, node, membership, and identity field guide

Close workspace selection from its declared authority. A folder and a manifest are observations, not membership votes.

Field	Question to close	Reviewable evidence	Invalid promotion
repo_path	Which material repository path is under review?	Exact path from the bounded inventory	Path proximity proves membership
node_kind	What kind of node is actually observed?	Directory, manifest, config, or package-like classification	Package-like means workspace member
declared_member	Is the path selected by the workspace authority?	Resolved member and exclude rules	A valid manifest is sufficient
membership_evidence	Which declaration and precedence rule support the state?	workspace.json locator, member match, and exclusion result	The tree or a command replaces declaration evidence
package_identity	Which identity does the manifest declare?	Manifest path and exact package name	Identity proves workspace membership
Layer check	Are repository and workspace claims still separate?	One containment locator and one selection locator	Same repository means same workspace
Non-member check	Why is experiments/demo outside the baseline workspace?	Exact exclusion wins despite its package.json	Excluded means absent from the repository
Environment identity	Which runtime and declaration were inspected?	Python version, resolved executable, declaration hash	Tool identity proves map correctness

Dependencies, configuration, scope, and bounded closure

Read package and configuration authorities directly. Keep deployment and reproducible build claims out of scope unless independently evidenced.

Field	Question to close	Reviewable evidence	Invalid promotion
declared_dependencies	Which package edges are declared?	Manifest dependency field and workspace specification	Adjacent folders are connected
config_owner	Who does the configuration declare as owner?	Config path plus declared_owner content	File location alone decides ownership
scope	Is the node a member, repository-only, or otherwise bounded?	Workspace resolution plus explicit scope label	Scope proves deployment topology
evidence_state	Is the material statement observed, inferred, or unknown?	OBSERVED, INFERRED, or UNKNOWN	Inference can be relabeled as observation
notes	Which contradiction, limitation, or non-claim matters?	Evidence-linked explanation	Narrative can repair missing authority
Build state	Was a build or install actually executed?	Exact command evidence or UNKNOWN_NOT_OBSERVED	A green inspection command proves reproducibility
Decision	Is the full map acceptable, rejected, or deferred?	ACCEPT, REJECT, or DEFER plus rationale	A formula or AI selects sufficiency
Residual uncertainty	What remains unavailable or out of scope?	Named unknown, consequence, and next probe	Unknown means harmless

Workspace Boundary Map capture

Use one row per material node. Cite the exact declaration, manifest, or configuration observation. If a value is unavailable, write UNKNOWN and narrow the claim.

repo_path	node_kind	declared_m ember	membership_evidenc e	package_identity	declared_dependenci es	config_owner	scope	evidence_state	notes
<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	true / false	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	OBSERVED / INFERRED / UNKNOWN	<div><div></div><div></div></div>
<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	true / false	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	OBSERVED / INFERRED / UNKNOWN	<div><div></div><div></div></div>
<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	true / false	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	OBSERVED / INFERRED / UNKNOWN	<div><div></div><div></div></div>
<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	true / false	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	OBSERVED / INFERRED / UNKNOWN	<div><div></div><div></div></div>
<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	true / false	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	OBSERVED / INFERRED / UNKNOWN	<div><div></div><div></div></div>
<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	true / false	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div>	OBSERVED / INFERRED / UNKNOWN	<div><div></div><div></div></div>

Map-level evidence boundary

Control field	Learner entry	Evidence locator	Review boundary
Environment/tool identity	<div><div></div></div>	<div><div></div><div></div></div>	Do not dump environment variables or credentials
Observed execution/build state	UNKNOWN_NOT_OBSERVED / exact observed state	<div><div></div><div></div></div>	Static evidence is not reproducibility proof
Independent observation	<div><div></div></div>	<div><div></div><div></div></div>	Name a file or command oracle that could falsify the map

Mutation response ledger

Reset to baseline before each row. Record one changed authority, the dependent claims that change, and the boundaries that remain stable.

Case	Changed authority	Observed mutation	Changed claims	Stable claims	Independent oracle
Baseline	workspace.json + manifests + policies	Initial inspection	Three members; one excluded node; owner mismatch	N/A	_____
add-demo-member	Workspace selection	_____	_____	Containment, existing manifests, config placement	_____
move-ui-config	Configuration path	_____	_____	Membership, package identities, dependency edges	_____
swap-web-dependency	apps/web manifest edge	_____	_____	Containment and workspace selection	_____

Evidence classification and human closure

Closure field	Learner entry	Evidence refs	Review question
Observed / inferred / unknown	_____	_____	Are material statements classified honestly?
Architecture tradeoff	_____	_____	Is one cost/benefit tradeoff explicit?
Deployment boundary	OUT_OF_SCOPE unless independently evidenced	_____	Was workspace membership kept separate?
Reproducibility boundary	UNKNOWN unless closure evidence exists	_____	Was a green command kept bounded?
Residual uncertainty	_____	_____	Does every material unknown remain visible?
Support disclosure	_____	_____	Was material AI or tool support independently validated?
Final human decision	ACCEPT / REJECT / DEFER	_____	Does the decision follow the evidence?

Changed context transfer

Create the transfer fixture and reconstruct the method without copying baseline paths, package names, probe order, or conclusion. The evidence standard stays fixed while the authorities change.

Transfer item	Second-case entry	Evidence locator	Review check
Fixture, environment, and declaration identity	_____	_____	All identity comes from the transfer case
Repository inventory	_____	_____	services/api, libs/shared, libs/policy, scratch/probe are observed
Workspace selection and exclusion	_____	_____	scratch/probe exclusion wins
Package identities and dependency edges	_____	_____	api → shared → policy is manifest-derived
Configuration path and declared owner	_____	_____	libs/shared/api.policy.json mismatch remains visible
Observed execution/build state	UNKNOWN_NOT_OBSERVED / exact observed state	_____	No reproducibility claim is promoted
Evidence classes and independent oracle	_____	_____	Unknown evidence remains unknown
New decision and residual uncertainty	ACCEPT / REJECT / DEFER	_____	The baseline conclusion is not copied

Critical dimension review

Dimension or closure field	Result or entry	Evidence locator	Repair or rationale
Boundary correctness — critical	PASS / REPAIR	_____	_____
Evidence traceability — critical	PASS / REPAIR	_____	_____
Mutation robustness — critical	PASS / REPAIR	_____	_____
Transfer and residual risk	PASS / REPAIR	_____	_____
Final human decision	ACCEPT / REJECT / DEFER	_____	_____