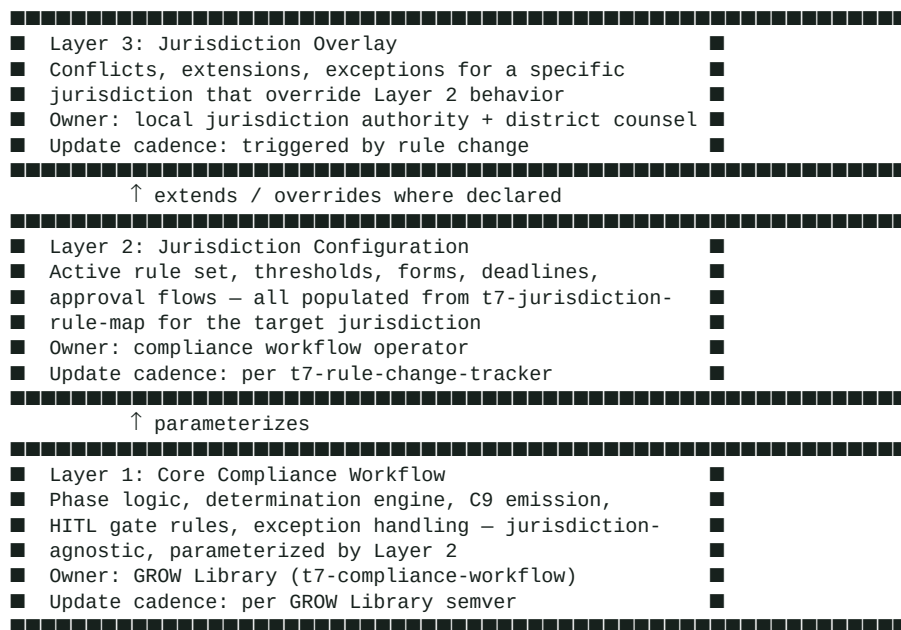


1. Three-Layer Architecture



Layer 1 invariants (must hold across all jurisdictions):

1. expert_review_required: true always raises an S1 HITL event and always blocks applicant output.
2. A determination with decision_origin: compliance-determination is only emitted for rule rows that are within the agent's deterministic classification authority.
3. Any rule_id not found in the active jurisdiction config produces determination: needs-review, not a default not-applicable.
4. retention_class: civic-7yr is applied to all compliance provenance records regardless of jurisdiction.
5. The "not legal advice" disclaimer is appended to all applicant-facing outputs regardless of jurisdiction.

Layer 2 — Jurisdiction Configuration

The jurisdiction configuration is a structured object loaded at agent initialization. It is the runtime materialization of the relevant rows from t7-jurisdiction-rule-map for a specific jurisdiction_id plus any cross-cutting universal rules.

Jurisdiction configuration schema:

```
jurisdiction_config:
  jurisdiction_id: <kebab-case>
  display_name: <human-readable name>
  effective_date: <YYYY-MM-DD>
  config_version: <semver>
  reviewer_role_map:
    engineering: <role name for engineer escalations>
    legal: <role name for legal escalations>
    administrative: <role name for administrative escalations>
  rules:
    - rule_id: <resolves to t7-jurisdiction-rule-map>
      active: <true | false>
      threshold_overrides: {} # if blank, use rule-map defaults
      form_ids: [<list>]
      deadline_days: <integer>
  intake_schema: <pointer to the applicable intake form schema>
  applicant_notice_template: <pointer to jurisdiction-specific notice language>
  appeal_path: <text describing the local appeal procedure> # [ATTORNEY REVIEW]
  retention_schedule_ref: <pointer to applicable state/local retention schedule>
```

Population rule. The jurisdiction configuration is populated by extracting rows from t7-jurisdiction-rule-map where jurisdiction_id matches AND scope is universal, combined with local rows for the same jurisdiction. A configuration may not reference a rule_id that does not exist in the map; adding a rule to a config requires adding it to the map first.

Layer 3 — Jurisdiction Overlay

An overlay is a structured exception declared by the local jurisdiction authority. It may extend Layer 2 (add a rule unique to this municipality) or override a Layer 2 threshold, form, or deadline. Overlays are the controlled mechanism for local variation; they replace ad hoc modifications to the core workflow.

Overlay schema:

```
jurisdiction_overlay:
  jurisdiction_id: <must match Layer 2 config>
  overlay_id: <kebab-case, unique>
  overlay_version: <semver>
```

```
effective_date: <YYYY-MM-DD>
authorized_by: <role; not personal identity>
attorney_review: true # all overlays require attorney review [ATTORNEY REVIEW]
changes:
  - type: <threshold-override | deadline-override | form-override | rule-extension | rule-
suppression>
    rule_id: <rule_id from Layer 2 config>
    field: <the specific field being changed>
    prior_value: <what Layer 2 says>
    new_value: <what this overlay requires>
    rationale: <why local law requires this deviation>
    citation: <local ordinance or ruling authorizing the deviation>
    verify_flag: true # overlays always carry [VERIFY]
```

Overlay safety rules:

- An overlay may not suppress a rule that has attorney_review: true in the jurisdiction map without a documented counsel sign-off.
- An overlay may not remove the expert_review_required flag from a rule that already carries it.
- Overlays are version-controlled; the active overlay for a jurisdiction is tracked in t7-rule-change-tracker as a distinct change_type: rule-extension or rule-suppression item.

2. Multi-Jurisdiction Deployment Pattern

Step 1 — Baseline audit. Before deploying to a new jurisdiction, run a jurisdiction audit: pull all universal-scope rule rows from t7-jurisdiction-rule-map, confirm last_confirmed dates, and flag any with verify_flag: true. The audit produces a gap list.

Step 2 — Local rule population. For the new jurisdiction, add scope: local rows to t7-jurisdiction-rule-map covering the jurisdiction's local ordinances, forms, and approval flows. Each new row requires a last_confirmed date and a reviewer-role attestation.

Step 3 — Build Layer 2 config. Populate the jurisdiction_config object, referencing only rule_id values now present in the map.

Step 4 — Overlay review. If the jurisdiction has local variations that modify universal rules, document them as Layer 3 overlays. Each overlay requires [ATTORNEY REVIEW] before activation.

Step 5 — Test. Run the operational test harness (defined in t7-govtech-package) against the new config using jurisdiction-specific test cases. Pass/fail criteria are the same as the baseline; the config, not the core logic, absorbs jurisdiction variation.

Step 6 — Go-live. Activate the config. The core workflow executes unchanged.

3. Preventing Hard-Code Drift

Hard-coding jurisdiction values into the core workflow — a statute citation, a form ID, a threshold number — is the most common cause of multi-municipality breakage. The architecture prevents it through structural rules, not code

review alone.

Lint rules (enforced at CI):

1. The t7-compliance-workflow artifact text must not contain any specific statute citation, form ID, or numerical threshold. Statutes and thresholds live in the jurisdiction map; the workflow references rule_id values only.
2. The jurisdiction config schema must validate that every rule_id in the config resolves to a row in t7-jurisdiction-rule-map at deployment time.
3. Any change to the core workflow that adds a jurisdiction-specific string is a gate-failing CI violation.

Review check. At each t7-rule-change-tracker quarterly review, a multi-jurisdiction reviewer role confirms that no new hard-codes have crept into the core workflow or the shared codebase.

4. Worked Example — Expanding from South Walton County to Okaloosa County

Scenario. GoodSam AI deploys the stormwater pre-screener in South Walton County (Walton County jurisdiction_id: fl-walton-county). A neighboring Okaloosa County special district requests the same capability. The expansion requires no changes to the core workflow.

Expansion checklist:

1. Jurisdiction audit for Okaloosa County: confirm universal rules are current. Both fdep-stormwater-general and fl-pub-records-access carry over unchanged (universal scope).
2. Local rules for Okaloosa County identified: Okaloosa County Land Development Code, Okaloosa County Stormwater Ordinance, and OWCD (Okaloosa Water & Sewer District) intake process. Each gets a new rule_id and scope: local rows added to t7-jurisdiction-rule-map with jurisdiction_id: fl-okaloosa-county. [VERIFY] on all new rows.
3. jurisdiction_config built for fl-okaloosa-county: universal rules included, local rules included.
4. Overlay review: Okaloosa County has a shorter notice period for certain permit classes. Overlay oc-overlay-001 is created: type: deadline-override, rule_id: okaloosa-co-ldco, prior_value: 90, new_value: 60. [ATTORNEY REVIEW] obtained from district counsel.
5. Test cases run against fl-okaloosa-county config. Walton County config remains unchanged and continues serving its existing applications.
6. Go-live: two active configs, one core workflow, one rule map with both jurisdiction sets.

What did NOT change: the C9 emission logic, the HITL gate trigger conditions, the aggregation logic, the "not legal advice" disclaimer, the retention class. These are Layer 1 invariants.

What changed: two config objects, six new rule-map rows, one overlay. Total new artifacts: contained and jurisdiction-scoped.

5. Config Versioning and Compatibility

Each jurisdiction_config carries a config_version semver. A config version bump is triggered by:

- A new rule being added to the active set (MINOR).
- A threshold, form, or deadline changing within an existing rule (MINOR).
- A rule being removed from the active set (MAJOR — requires [ATTORNEY REVIEW]).
- The intake_schema pointer changing (MINOR if backward-compatible, MAJOR if not).

Config version mismatches between a running workflow and the loaded config are a schema-drift-input failure mode per s1-failure-mode-register and route to HITL.