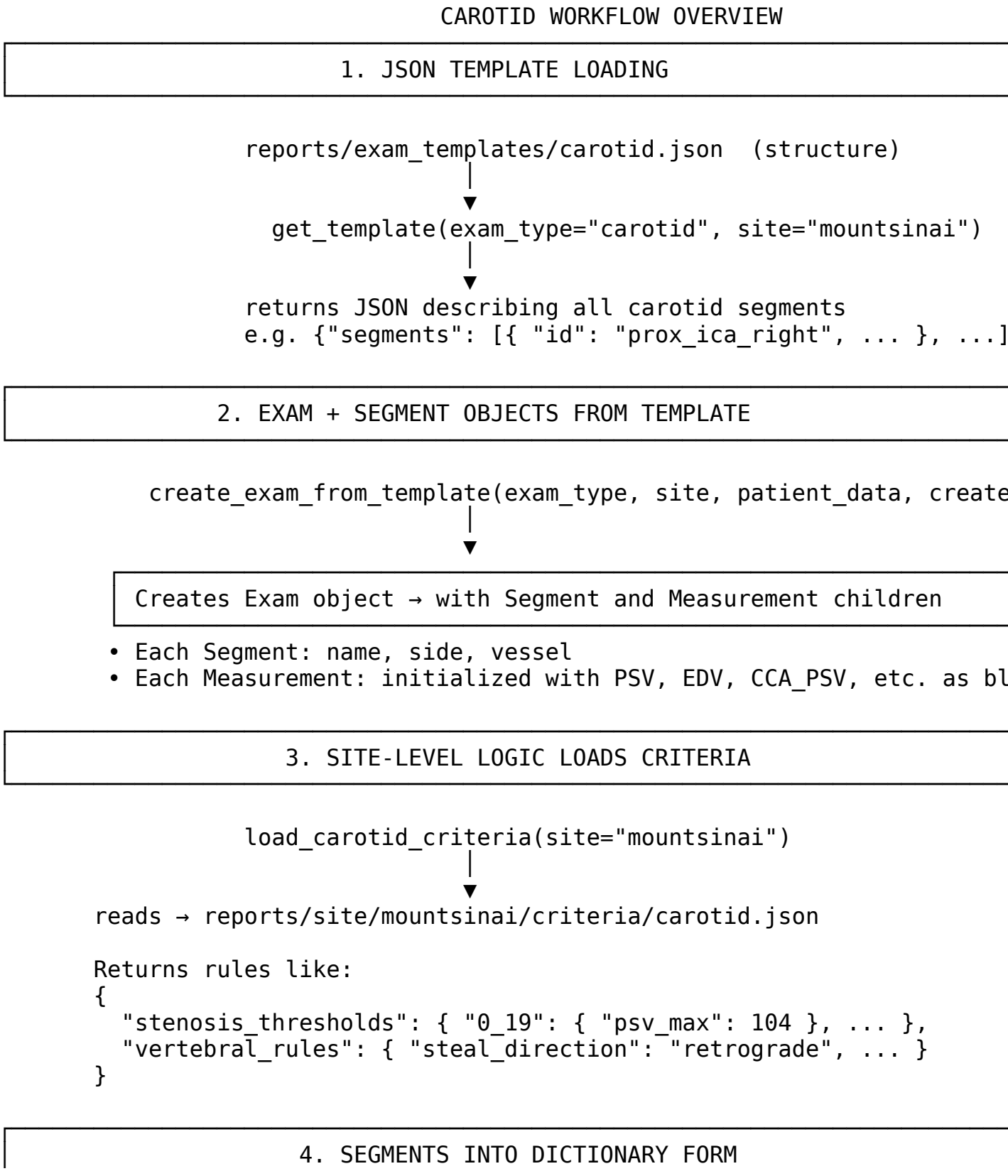
LUMEN: Carotid Exam Engine — Full Lifecycle Overview (Phase 1A + 1B)

From Template to Final Report



```
└──────────────────────────────────────────────────────────────────────────────

                    for segment in exam.segments:
                        segment_dict[segment.name] = {
                            "psv": segment.measurement.psv,
                            "edv": segment.measurement.edv,
                            ...
                        }

    These are typed using:
    → ArterialSegmentBase
    → CarotidSegmentDict (adds ica_cca_ratio, stenosis_category, etc.)

┌──────────────────────────────────────────────────────────────────────────────
│                          5. CALCULATOR LOGIC APPLIED
└──────────────────────────────────────────────────────────────────────────────

            calculator = CarotidCalculator(segment_dict, carotid_criteri
            calculator.run_all()

  ▶ For each segment:
      • compute_ica_cca_ratio() → adds "ica_cca_ratio"
      • apply_stenosis_logic()  → adds "stenosis_category" + "stenosis_no
      • interpret_vertebral_waveform() → adds "vertebral_comment"

┌──────────────────────────────────────────────────────────────────────────────
│                          6. OUTPUT + INTEGRATION
└──────────────────────────────────────────────────────────────────────────────

    calculator.get_segment_data()
        → gives structured dict[str, CarotidSegmentDict] with all fields

    calculator.export_json()
        → gives pretty-printed JSON (debug/log use)

    ⬇

    Optional:
    generate_conclusion(exam) → builds editable findings summary like:
    "Prox ICA R: Findings consistent with 60–79% stenosis."
    "Left Vertebral: Retrograde vertebral flow consistent with subclavian

    ⬇

    Final Output:
    • Frontend Form (Formik + MeasurementTable)
    • PDF (WeasyPrint via PDF template)
    • HL7 Payload (NextGen via Mirth)



##     Quick Reference – Key Modules and Their Role
```

File / Component	Purpose
`carotid.json`	Defines the structure of all carotid segments
`get_template()`	Loads template based on exam_type + site
`create_exam_from_template()`	Creates DB models from that template
`load_carotid_criteria()`	Loads threshold rules per site
`ArterialSegmentBase`	Shared segment fields (psv, edv, waveform, etc.)
`CarotidSegmentDict`	Extended carotid-specific fields
`CarotidCalculator`	Executes stenosis, ICA/CCA, vertebral logic