2026 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SANTA CLARA, CA, USA
MARCH 2 - 5, 2026

Etabot: Multi -Agent Verification Management with Chat-Accessible Midpoint Metrics and Finish-Date Forecasts

Zili Fang

SILICON LABS

Agenda

- Motivation
- What Etabot does
- Architecture
- Midpoint metrics
- Finish-date
- Extensibility

Motivation

Why this hurts

- “What’s the project status now?”
- “When will we be done?”

When using AI

- Data Security

What is Etabot?

- One intent → status

Run

Submit/render regressions (farm + simulator)

Analyze

Merge coverage
Classify failures
Compute midpoint metrics

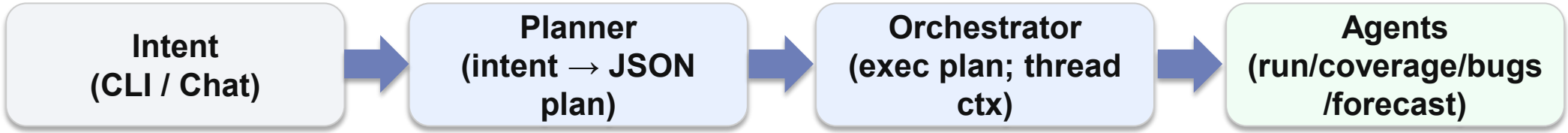
Forecast

Reset-aware Monte Carlo
→ P50 / P80 dates

Report

One-page Markdown

Architecture



Registered tasks (typed I/O):

- run_regression → dispatch to CI / scheduler / internal simulation tool (dry-run by default)
- merge_coverage / report_coverage → unified coverage DB + text report
- generate_results → create RESULTS.CSV from regression artifacts
- collect_metrics → append coverage + pass rate into METRICS_TS.CSV
- jira_fetch → open bugs + avg fix time
- forecast_dual → P50/P80 to code & functional targets
- report → manager-ready REPORT.MD

Intent → strict JSON plan

- Deterministic + safe
- Planner converts a short intent into an ordered task list.
- Side-effectful tasks default to dry_run=true.
- Plan is loggable, diffable, replayable.
- Optional: local / on-prem LLM; rule fallback works offline.

```
intent: "nightly regression, coverage,
     metrics, jira, forecast 95%"

                    {
                  "tasks": [
              {"task":"run_regression",
          "params":{"dry_run": true}},
              {"task":"merge_coverage",
        "params":{"dry_run": true, ...}},
             {"task":"report_coverage",
        "params":{"dry_run": true, ...}},

{"task":"generate_results","params":{...}},
             {"task":"collect_metrics",
                "params":{...}},
               {"task":"jira_fetch",
                "params":{...}},
              {"task":"forecast_dual",
                "params":{...}},
     {"task":"report",          "params":{}}
                    ]
                   }
```

Agents → Outputs

- **Planner** → plan.json (tasks + params)
- **Runner** → job trigger info + reg_root
- **Coverage (merge)** → merged coverage DB
- **Coverage (report)** → cov_report.txt
- **Results** → results.csv (PASS/FAIL + infra/design)
- **Metrics** → metrics_ts.csv (date, ccov, fcov, passrate)
- **Bug** → {open_bugs, avg_fix_days}
- **Forecaster** → {P50, P80} dates (code / functional / overall)
- **Reporter** → report.md (1-page status)

(optional) **MCP Server** → exposes the same tools/resources to chat clients

Midpoint metrics

Same primitives everywhere (CLI, chat, MCP):

- Design-only pass rate (infra excluded)
- Code / functional coverage status
- Open bugs + average fix time
- Forecast to targets (P50/P80)

Manager-ready report

- One-page summary (Markdown)
- Midpoint metrics: pass rate, coverage, open bugs
- Forecast: P50 / P80 dates to targets
- Rendered commands included when dry-run (auditability)

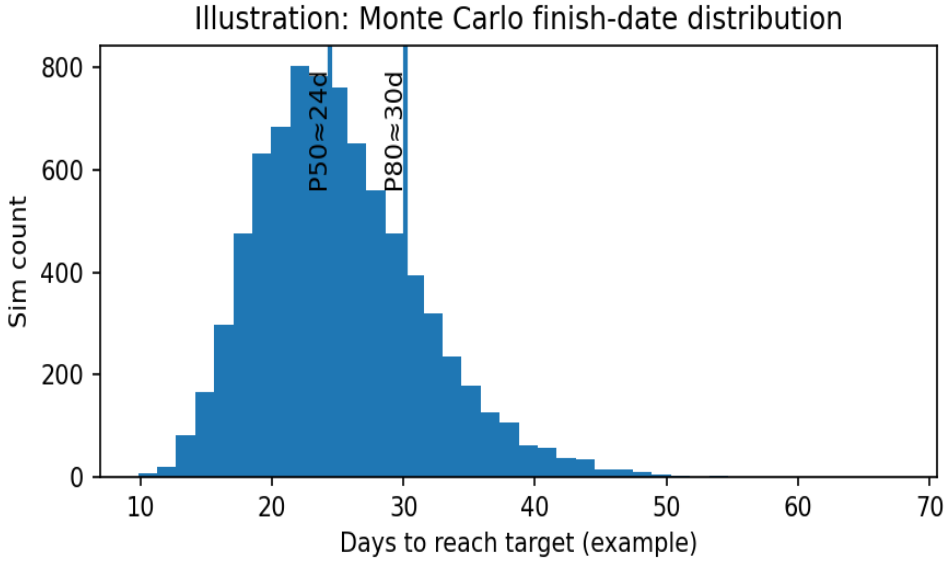
```
# Etabot Nightly Status
Date: 2026-02-04

## Midpoint Metrics
- Design-only pass rate (7-day): 93.0%
- Code coverage: 80.0%
- Functional coverage: 70.0%
- Open bugs: 12 | Avg fix time: 3.8 days

## Forecast (targets: 95% / 95%)
- Code:  P50 2026-02-08 | P80 2026-02-12
- Func:  P50 2026-03-01 | P80 2026-03-10
- Overall: P50 2026-03-01 | P80
2026-03-10
```

What are P50 / P80 dates?

How to read the forecast
- We run many simulations of “days to reach target coverage”.
- Each simulation produces one finish date.
- Sort finish dates → compute percentiles.
- P50 = median date (50% finish on/before).
- P80 = conservative date (80% finish on/before).



Finish-date forecast

- Reset-aware Monte Carlo over completion time
  1) Fit recent coverage slope (code + functional)
  2) Scale for farm capacity (slots)
  3) Apply pass-rate penalty (Beta posterior over design-only pass rate)
  4) Apply bug drag (open bugs × average fix time)
  5) Sample uncertainty → distribution → P50 / P80 dates
- Output is explainable and model-swappable via Forecast API.

```
Example output
code:  P50 2026-02-08 | P80 2026-02-12
func:  P50 2026-03-01 | P80 2026-03-10
overall: P80 2026-03-10
```

Plan vs Agent vs Chat (+ LLM on/off)

PLAN

- intent → **JSON plan**
- **no execution**
- audit / diff

AGENT

- plan → **run agents**
- **dry-run default**
- artifacts → out/…

CHAT

- Q&A on artifacts
- pass-rate / bugs / P50-P80
- “what’s true now?”

LLM optional:

- **OFF:** rule/regex → deterministic tool calls (offline)
- **ON:** LLM → same tool calls (better NL), still structured outputs

Local-first safety

- Default: render-only
  - All side-effectful commands default to dry_run=true.
- Execution requires BOTH:
  - 1) dry_run=False
  - 2) ETABOT_ALLOW_EXEC=1 (explicit gate)
- No-egress mode
  - Local artifacts only; chat tools call explicit functions.
  - LLM planning is optional and can be local / on-prem.
  - Bind MCP to localhost to keep resources inside firewall

Extensibility

- Where to extend
  - Add a new runner backend (new adapter; no orchestration changes)
  - Replace coverage backend (vendor tool or UCIS reader)
  - Swap issue tracker adapter
  - Replace forecast model behind stable output contract
- Why it stays maintainable
  - Single task registry + typed params/results
  - Small agents; easy to mock; unit tests for classifier/forecast

Limitations & roadmap

- What Etabot is (and isn’t)
  - Not a coverage-closure optimizer (it doesn’t generate stimulus).
  - Orchestrates existing tools and improves visibility & forecasting.
  - Depends on backend fidelity (text parsing vs DB-native where possible).
- Roadmap
  - More adapters: simulators, schedulers, issue trackers
  - Database-native readers to reduce parsing
  - Dashboards backed by the same MCP resources
  - Forecast calibration (real data)

Wrap-up

- Three takeaways:
  - One intent → typed plan → safe on-prem execution → manager-ready report
  - Midpoint metrics are self-serve (CLI + chat)
  - Reset-aware Monte Carlo forecasts (P50/P80) with a swappable model
- Questions?