Technical Overview — 2 and 1/2 Hackers

> **Oral contraceptive recommendation engine** powered by a medical-safety-first agentic AI loop, two purpose-trained ML models, and a deterministic clinical safety gate. No LLM ever makes a medical safety decision.

Table of Contents

---

1. System Architecture

```
┌──────────────────────────────────────────────────────────────
┐
│                         FastAPI Backend
│
│  Patient JSON ──► LangGraph Agent Graph ──► Ranked Recommendation
│                             │
│                 ┌───────────┼───────────┐
│                 ▼           ▼           ▼
│            [Safe Gate]  [Cluster API] [Simulator API]
│   Deterministic        Red Hat          Red Hat
│   Hard Rules           OpenShift        OpenShift
└──────────────────────────────────────────────────────────────
┘
```

The system is built around three pillars:

Pillar	Technology	Purpose
Agent Brain	LangGraph + LangChain + LLM	Orchestrates reasoning, weight selection, convergence
Safety Gate	Pure Python rule engine	Enforces WHO Medical Eligibility Criteria — no LLM
ML Models	Scikit-learn, served via FastAPI on OpenShift	Patient clustering + pill trajectory simulation

2. The Agentic Loop

The agent is implemented as a **LangGraph** `StateGraph` — a directed graph where typed state flows between nodes. The loop is:

```
validate
   │
assign_cluster  ◄───  calls Cluster Model API (Tool 1)
   │
generate_candidates  ◄───  Safe Gate filters contraindicated pills
   │
assess_risk  ◄───────────  AGENT DECISION: which pills to simulate?
   │                                            ▲
simulate  ◄─────────────────────────────────────  │
   │        calls Simulator API for each pill     │
score_utility  ◄──  pure math utility formula     │
   │                                              │
check_convergence  ◄──  AGENT DECISION: stop or iterate?
   │                          │
   └── converged? ──YES──► END │
                    NO─────────┘  (new weights, new pills)
```

Loop State

Every node reads from and writes to a shared `SystemState` TypedDict:

```python
class SystemState(TypedDict):
    patient_data: dict              # Normalized patient record
    cluster_profile: str | dict     # GMM cluster assignment
    cluster_confidence: float       # 0.0 – 1.0
    candidate_pool: list            # Safe pills ONLY (post-gate)
    risk_scores: dict               # {pill_id: risk_factors}
    pills_to_simulate: list         # Agent-selected subset
    simulated_results: dict         # Accumulated across iterations
    utility_scores: dict            # {pill_id: float}
    utility_weights: dict           # Agent-chosen {α, β, γ, δ}
    best_candidate: str             # Current top pill
    iteration: int                  # Loop counter (max 10)
    converged: bool                 # Termination signal
    reason_codes: list              # Human-readable explanation
```

> **Key design**: `hard_constraints` are **never stored in state**. The agent receives a pre-filtered `candidate_pool` and has no visibility into what was excluded. Medical safety is structurally enforced, not prompted.

Maximum Iterations

The loop is bounded at **10 iterations**. If the agent has not converged by then, the best candidate at that point is returned. This guarantees bounded latency while still allowing the agent multiple passes to refine its weights and compare pills it may have missed.

---

3. Agent Decision Points

The LLM is the agent at exactly **three nodes** in the graph. Everywhere else, decisions are deterministic.

3.1 `assign_cluster` — Low-Confidence Weight Adjustment

When the Cluster Model confidence is **< 0.70**, the LLM is invoked to produce a **risk weight multiplier**:

```
Triggered when: cluster_confidence < 0.70
Output: {"weight_adjustment": float, "rationale": str}
Effect: α and β (risk penalty weights) are scaled UP by this multiplier
        → more conservative scoring when the patient profile is ambiguous
```

The LLM does **not** choose the cluster — that is the GMM's exclusive job. It only adjusts how cautiously to treat the uncertainty.

3.2 `assess_risk` — Which Pills to Simulate?

This is the agent's primary reasoning step. The LLM receives:

- Full patient profile (conditions, vitals, age)
- Cluster assignment and confidence
- Detailed pill data (type, substance, dosage, boxed warnings, contraindication excerpts from FDA labels)
- Previous simulation results (on re-iterations)
- Any specific pills the convergence agent requested to re-examine

The LLM outputs a **structured JSON**:

```
{
  "risk_assessments": {
    "EE30_DRSP3": {
      "risk_score": 0.15,
      "risk_factors": ["high VTE class", "patient has hypertension flag"],
      "simulation_priority": "high"
    }
  },
  "pills_to_simulate": ["EE30_LNG150", "NET_PO_350", "EE20_DRSP3"],
  "simulation_rationale": "Prioritising lower VTE-risk options given hypertension..."
}
```

This prevents brute-forcing all 9 pills every iteration. The agent acts like a clinician who decides *which* options are worth investigating further.

3.3 `check_convergence` — Stop or Iterate?

After scoring, the LLM gets the top-3 pills with their full simulation trajectories and utility scores, then decides:

If converging:

```
{
  "converged": true,
  "reason_codes": ["Lowest discontinuation probability in cohort", ...],
  "top3_reason_codes": {"EE30_LNG150": [...], "NET_PO_350": [...]},
  "medical_rationale": "Best safety/tolerability profile for this patient"
}
```

If continuing:

```
{
  "converged": false,
  "new_weights": {"alpha": 2.5, "beta": 2.0, "gamma": 0.3, "delta": 0.8},
  "pills_to_reconsider": ["EE20_DRSP3"],
  "medical_rationale": "Severity weight too low — re-examining anti-androgenic options"
}
```

The agent can **adjust the utility weight vector** each iteration to shift emphasis (e.g., prioritise safety over effectiveness for a high-risk patient) and request specific pills to be re-simulated with the updated weights.

---

4. Tools — ML Models as APIs on Red Hat OpenShift

Both ML models are **deployed on Red Hat OpenShift** and exposed as REST APIs. The agent calls them as external tools — clean HTTP boundaries that decouple the model lifecycle from the agent.

```
Agent Node              HTTP Tool Client           OpenShift Route
──────────              ──────────────────         ──────────────────
```

```
assign_cluster    ───▶   cluster_api.py       ───▶   /cluster/predict
simulator         ───▶   simulator_api.py     ───▶   /simulator/simulate
```

Tool 1 — Cluster API

```
POST {CLUSTER_API_URL}
{
  "patient": {
    "age": 28,
    "cond_dvt": 0, "cond_hypertension": 1, "cond_migraine_with_aura": 0,
    "obs_bmi": 24.5, "obs_systolic_bp": 138, "obs_smoker": 0,
    ...  (37 features total)
  }
}
```

Response:

```
{
  "cluster_profile": "cluster_3",
  "cluster_confidence": 0.94
}
```

Error handling: 400/422 raises immediately (schema bug); 503 retries once after 2 s; 500 propagates with logged context.

Tool 2 — Simulator API

Called **once per pill, concurrently** using `asyncio.gather`, to maximize throughput:

```
POST {SIMULATOR_API_URL}
{
  "candidate_pill": {
    "combo_id": "EE30_DRSP3",
    "pill_type": "combined_monophasic",
    "estrogen_dose_mcg": 30,
    "progestin_dose_mg": 3.0,
    "vte_risk_class": "high",
    ...
  },
  "patient": { "age": 28, "cond_*": ..., "obs_*": ... },
  "n_months": 12
}
```

Response — full monthly trajectory:

```
{
  "combo_id": "EE30_DRSP3",
  "months": [1, 2, 3, ...., 12],
  "symptom_probs": {
    "sym_nausea":        [0.021, 0.018, ...],
    "sym_mood_worsened": [0.031, 0.028, ...],
    "sym_acne_improved": [0.042, 0.071, ...],
    "still_taking":      [0.91,  0.88,  ...],
    "evt_dvt":           [0.000, 0.000, ...]
  },
  "satisfaction": [6.1, 6.3, 6.5, ...],
  "discontinuation_probability": 0.09,
  "severe_event_probability": 0.0002,
  "mild_side_effect_score": 0.18,
  "contraceptive_effectiveness": 0.63
}
```

The four summary metrics at the bottom are what feeds the utility formula. The full monthly trajectory is logged and available for the frontend to visualize.

---

5. Medical Safety Gate — Zero LLM Involvement

This is the most important architectural decision in the system: **medical safety is enforced by deterministic code, not by the LLM**.

The `SafeGateEngine` runs inside the backend before the agent ever sees a list of pill options. It operates in two stages.

Stage 1 — Hard Constraint Rules (Patient-Specific)

Seven rule functions, each encoding a **WHO Medical Eligibility Criteria Category 3/4** contraindication:

Rule	Clinical Basis
`_contraindicated_combined_vte`	Combined OC + DVT/stroke/PE history → excluded
`_contraindicated_combined_smoking_over_35`	Combined OC + smoker + age > 35 (WHO MEC Cat 3/4)
`_contraindicated_combined_migraines_with_aura`	Combined OC + migraine with aura (WHO MEC Cat 4)
`_contraindicated_combined_breast_cancer`	All hormonal OC + breast cancer history (WHO MEC Cat 4)
`_contraindicated_combined_liver_disease`	Combined OC + hepatitis/cirrhosis (WHO MEC Cat 3/4)

Rule	Clinical Basis
`_contraindicated_combined_lupus`	Combined OC + SLE/lupus (WHO MEC Cat 3/4)
`_contraindicated_high_vte_hypertension`	High VTE-risk pill + hypertension (3rd/4th-gen progestin classes)

These are pure boolean functions: `(patient_data, pill_record) → bool`. No probability, no weighting — a pill is either contraindicated or it isn't.

Stage 2 — Cluster-Level Exclusions (Population-Level)

After patient-specific rules, the GMM cluster assignment is used to apply **population-level exclusions**. If the patient's cluster is associated with a high prevalence of a contraindicated condition, the corresponding pill families are excluded even if the patient's individual flags don't trigger Stage 1.

This second layer catches statistically elevated risk in patient subgroups even when individual binary flags are absent.

What the Agent Sees

```
All 9 pills ──► [Hard Rules] ──► [Cluster Exclusions] ──► candidate_pool
                  excluded           excluded                  (1–9 pills)
                     │                   │
                     └───────────────────┘
                       NEVER returned to agent
```

The agent receives only `candidate_pool` — a list of pill IDs that are safe to consider. It has no access to what was excluded or why. The exclusion logic is **structurally unreachable** from any LLM prompt path.

Relative Risk Rules (Soft Layer)

After the hard filters, five soft signals are generated and passed to the risk assessor as context. These do not exclude — they inform:

Rule	Signal
`hypertension_estrogen_penalty`	Higher estrogen doses carry elevated risk
`obesity_effectiveness_concern`	BMI ≥ 30 may reduce effectiveness of low-dose pills
`smoking_combined_penalty`	Smoking increases VTE risk (age ≤ 35)
`depression_progestin_preference`	Some progestins may worsen mood
`epilepsy_interaction_concern`	Enzyme-inducing AEDs reduce pill effectiveness

---

6. ML Model 1 — Clustering (Patient Profiling)

Architecture

Gaussian Mixture Model (GMM) with diagonal covariance (`covariance_type=diag`), **k = 12 components** selected by Bayesian Information Criterion (BIC) over k ∈ {3, 4, … 12}.

- Train/test split: **80/20**, stratified on `has_absolute_contraindication_combined_ocp`
- Confidence threshold for hard assignment: **θ = 0.40** per component
- Confidence threshold for triggering LLM weight adjustment: **0.70**

Why GMM?

GMM gives **soft probabilistic membership** — each patient gets a probability vector over all 12 clusters, not a single hard label. This is critical for borderline patients: a patient with confidence 0.55 on "Hypertension + Diabetes" and 0.35 on "Baseline" should be treated more conservatively than one with confidence 0.94. The soft output is used directly in the utility weight scaling.

Input Features (37 total)

Group	Features
Continuous vitals	`age`, `obs_bmi`, `obs_systolic_bp`, `obs_diastolic_bp`, `obs_phq9_score`, `obs_testosterone`
WHO MEC Cat 4	`cond_migraine_with_aura`, `cond_stroke`, `cond_mi`, `cond_dvt`, `cond_breast_cancer`, `cond_lupus`, `cond_thrombophilia`, `cond_atrial_fibrillation`, `cond_liver_disease`
WHO MEC Cat 3	`cond_hypertension`, `cond_migraine`, `cond_gallstones`, `cond_diabetes`, `cond_prediabetes`, `cond_epilepsy`, `cond_chronic_kidney_disease`, `cond_sleep_apnea`
Indications	`cond_pcos`, `cond_endometriosis`
Comorbidities	`cond_depression`, `cond_hypothyroidism`, `cond_rheumatoid_arthritis`, `cond_fibromyalgia`, `cond_osteoporosis`, `cond_asthma`, `obs_smoker`

Missing continuous values → median imputation (training medians). Missing binary values → 0 (conservative absence assumption).

The 12 Discovered Profiles

#	Profile Name	Train n (%)	Key Conditions	Blocked Pills
0	Rheum. Arthritis + Sleep Apnea	14 (0.3%)	RA, Sleep Apnea, Hypertension	All 9
1	Migraine + Depression	661 (16.1%)	Migraine, Depression, Endometriosis	**None**
2	Diabetes + Breast Cancer	66 (1.6%)	Diabetes, Breast Cancer, Depression	All 9

#	Profile Name	Train n (%)	Key Conditions	Blocked Pills
3	Hypertension + Diabetes	57 (1.4%)	Hypertension, Diabetes, Endometriosis	All 8 combined OCPs
4	Thrombophilia + Endometriosis	138 (3.4%)	Thrombophilia, Endometriosis, Migraine+Aura	All 8 combined OCPs
5	**Baseline / Low-Risk**	2403 (58.4%)	—	**None**
6	PCOS + Thrombophilia	13 (0.3%)	PCOS, Thrombophilia, Epilepsy	All 9
7	Epilepsy	65 (1.6%)	Epilepsy	All 8 combined OCPs
8	PCOS + Hypertension	72 (1.7%)	PCOS, Hypertension, Migraine	All 8 combined OCPs
9	PCOS	239 (5.8%)	PCOS	**None**
10	Hypertension + Migraine+Aura	360 (8.7%)	Hypertension, Migraine+Aura	All 8 combined OCPs
11	Hypertension + Thrombophilia	29 (0.7%)	Hypertension, Thrombophilia, Diabetes	All 8 combined OCPs

Test-Set Performance

Metric	Value	Notes
Safety Recall	**96.5%**	3 false negatives out of 85 absolutely contraindicated patients
Safety Precision	41.4%	Conservative over-blocking is acceptable in a medical safety context
Per-condition block rate	94–100%	All 6 audited WHO MEC Cat 4 conditions ≥ 94%
Mean assignment confidence	**99.9%**	Near-zero entropy — patients are cleanly assignable
Silhouette score	−0.02	Expected on sparse binary feature space; GMM BIC, not silhouette, is the validity criterion here

> **96.5% safety recall** is the headline metric: of every 100 patients who should have combined pills blocked, 96 or 97 are correctly protected by the model. The 3.5% who slip through are still caught by Stage 1 hard rules (the two layers are complementary).

7. ML Model 2 — Simulator (Trajectory Forecasting)

Architecture

Two **HistGradientBoosting** models trained on **444,636 rows** (4,117 training patients × 9 pills × 12 months):

Model	Type	Target
`model_symptoms.pkl`	`MultiOutputClassifier(HistGBM)`	18 binary symptom / event flags per month
`model_satisfaction.pkl`	`HistGBMRegressor`	`satisfaction_score` (1–10 continuous) per month

`month` (integer 1… N) is passed as a plain numeric feature. This means **any horizon up to 12 months** can be requested at inference without retraining — you query month 1 through 12 independently and the model sees the temporal position as a feature.

Why HistGradientBoosting?

- Native NaN handling: PHQ-9 score is missing in 64% of records, testosterone in 92% — HistGBM handles this without any imputation pipeline
- Scales to 444k rows without the memory overhead of exact gradient boosting
- `MultiOutputClassifier` wrapper allows joint learning for all 18 binary targets in one sklearn-compatible interface

Input Features

37 patient features (identical to the clustering model) plus **6 pill features** derived from `pill_reference_db.csv`:

Pill Feature	Encoding
`pill_type_binary`	1 = combined, 0 = progestin-only
`estrogen_dose_mcg`	0, 20, 25, 30, 35 mcg
`progestin_dose_mg`	numeric (mg)
`progestin_generation`	1–4 ordinal
`androgenic_score`	anti-androgenic = −1, low = 1, moderate = 2, high = 3
`vte_risk_numeric`	very_low = 1 … high = 5

Plus the temporal feature `month` (1–12).

The 18 Binary Targets (symptom_probs)

Symptom / Event	Clinical Meaning
`sym_nausea`	Nausea side effect
`sym_mood_worsened`	Mood deterioration
`sym_acne_improved`	Acne improvement (positive)

Symptom / Event	Clinical Meaning
`sym_cramps_relieved`	Dysmenorrhoea relief (positive)
`sym_spotting`	Breakthrough bleeding
`sym_pcos_improvement`	PCOS symptom reduction (positive)
`still_taking`	Patient continues taking the pill (NOT discontinued)
`evt_dvt`	Deep vein thrombosis event
`evt_pe`	Pulmonary embolism event
`evt_stroke`	Stroke event
… (8 more)	Additional symptom and safety targets

Test-Set Performance (1,030 held-out patients)

Binary Targets — AUROC

Target	AUROC	Note
`sym_pcos_improvement`	**0.992**	PCOS binary flag is the dominant driver
`sym_cramps_relieved`	**0.987**	Endometriosis flag + progestin generation
`sym_acne_improved`	**0.804**	Anti-androgenic score (drospirenone)
`sym_spotting`	**0.786**	Progestin-only flag + dose
`still_taking`	**0.778**	Satisfaction + profile interaction
Mood / MH signals	0.62–0.67	PHQ-9 64% missing, multi-condition interactions
`evt_dvt / evt_pe / evt_stroke`	n/a	~0 prevalence in test set — primary gate is the WHO MEC blocking layer
Mean AUROC	**0.695**	Across all meaningful binary targets

Satisfaction Regression

RMSE	MAE	R^2
0.873	0.645	0.35

The 65% unexplained variance reflects inherent stochasticity: the model predicts the **expected probability** of a symptom occurring, not a random draw. Individual patients will vary around these expectations — the model gives the best estimate given clinical features.

---

8. Utility Scoring — Pure Mathematics

Once simulation results are available, utility is computed by a **closed-form formula** with no LLM involvement:

$$U(pill) = -\alpha \cdot P(\text{severe_event}) - \beta \cdot P(\text{discontinuation}) - \gamma \cdot \text{mild_side_effect_score} + \delta \cdot \text{contraceptive_effectiveness}$$

Weight	Default	Meaning
α	2.0	Penalty for severe clinical events (DVT, PE, stroke)
β	1.5	Penalty for discontinuation (patient stops the pill)
γ	0.5	Penalty for mild side effect burden
δ	1.0	Reward for contraceptive effectiveness

The agent sets these weights at the `check_convergence` step — but the **computation itself is always deterministic arithmetic**. The agent reasons about what emphasis is medically appropriate for this patient; the math executes that emphasis.

When cluster confidence is low, `α` and `β` are automatically scaled up by a `weight_adjustment` multiplier generated at the `assign_cluster` step — being more conservative when patient classification is uncertain.

---

9. Data Pipeline

Synthetic Patient Data

- **Synthea** used to generate realistic synthetic patient records
- Custom Synthea modules model OCP-relevant conditions (PCOS, endometriosis, thrombophilia, migraine with aura)
- 4,117 training patients, split 80/20 stratified on contraindication status

Pill Reference Database (`pill_reference_db.csv`)

- 9 oral contraceptive formulations covering the full clinical spectrum
- Columns include: `combo_id`, `pill_type`, `estrogen`, `estrogen_dose_mcg`, `progestin`, `progestin_dose_mg`, `progestin_generation`, `androgenic_score`, `vte_risk_class`, `known_brand_examples`
- Enriched with FDA drug label data (warnings, contraindications, boxed warnings) via `pull_fda_drug_labels.py`
- FAERS adverse event rates pulled via `pull_fda_faers_events.py`

Symptom Diary Generation

- For each (patient, pill, month) triplet, a synthetic monthly symptom diary is generated using clinically-grounded probability rules
- Probabilities are conditioned on patient conditions, pill pharmacological profile, and temporal patterns (e.g., nausea peaks at month 1, declines)
- 444,636 training rows total

10. End-to-End Request Flow

1. **Patient JSON** arrives at the FastAPI endpoint
2. `validate` normalizes and type-checks all fields; initializes loop state
3. `assign_cluster` calls the Cluster Model API (Tool 1 on OpenShift) → gets `cluster_profile` + `cluster_confidence`
    - If confidence < 0.70: LLM generates a `weight_adjustment` scalar (more conservative scoring)
4. `generate_candidates` runs the Safe Gate Engine:
    - Stage 1: 7 hard constraint rules check every pill against this patient
    - Stage 2: cluster-level exclusions applied
    - → `candidate_pool` (1–9 safe pills)
5. `assess_risk` (LLM): receives the pool + patient context + pill details; selects the highest-priority subset to simulate (typically 3–5 pills on first iteration)
6. `simulate` calls the Simulator API (Tool 2 on OpenShift) **concurrently** for each selected pill → gets full 12-month trajectories + summary metrics
7. `score_utility` applies the deterministic utility formula with current weights → ranks all simulated pills
8. `check_convergence` (LLM): analyzes top-3 pills; decides **STOP** (→ reason codes) or **CONTINUE** (→ new weights + pills to reconsider)
9. Loop back to step 5, or exit with the ranked recommendation

Total LLM calls per request: minimum 2 (assess_risk + check_convergence), maximum ~12 (1 weight_adjustment + 10 × assess_risk + 10 × check_convergence).

Total ML API calls per request: 1 cluster call + (N_pills_simulated × N_iterations) simulator calls, all IO-multiplexed via asyncio.

Summary of Design Principles

Principle	Implementation
LLM makes no medical safety decisions	Hard rules and cluster exclusions run before LLM has any visibility
Agent selects strategy, math computes scores	Utility formula is fixed; agent only adjusts weights
Bounded iteration	MAX_ITERATIONS = 10, guaranteed termination
ML models are swappable APIs	Two client files (`cluster_api.py`, `simulator_api.py`) are the only ML boundary
Concurrency	Simulator calls are async-gathered — all pills simulated in parallel per iteration
State accumulation	Simulation results accumulate across iterations — agent sees all pills ever simulated, not just the current batch

Principle	Implementation
Conservative uncertainty handling	Low cluster confidence → higher risk penalties; missing data → 0 (absent condition assumed)