Project Blueprint

Forecast-then-Act Quality Control with Gen AI Reports

1. Why this project?

- Statin-tablet plants still lose ≈ $1.2 M / year from batches rejected after lab testing.
- Traditional 10 % sampling ignores real-time process drift, leaving 22 % of high-risk periods under-tested[1].
- RL has been shown to cut false-positive alarms by 37 % and improve throughput 12 % in pharma lines.
- Generative-AI RAG chatbots already speed FDA-style documentation by 50-65 %.

It fuses three proven levers—time-series forecasting, safe RL control and RAG-based Gen AI reporting—on the same Nature 2022 dataset of 1 005 batches that we used earlier.

2. End-to-end functional flow

1. Sensors stream 1 Hz data on raw-material moisture, compression force, speed, humidity, temperature and in-line NIR quality signals. (Azure Data Factory, Azure Event Hubs)
2. Forecasting micro-service (LSTM) predicts 1 h quality trajectory; if defect probability > 0.7 it raises a “yellow” flag 30 min before drift.
3. Safe RL controller (Proximal Policy Optimization with action-masking) receives latest state + forecast.
    - If forecast is healthy, it exploits for throughput; if risky, it explores tighter settings.
    - Hard safety layer clips actions that breach pharmacopeia limits (content ±5 %, weight ±3 mg).
4. Actions (e.g., –10 % speed, +2 kN compression, 2× sampling rate) are dispatched through OPC-UA to the tablet press.
5. All events are logged (Cosmos DB); a LangChain-powered RAG service pulls batch data + FDA/EMA guidelines and drafts a 21 CFR 11-compliant “Real-Time Release” report with root-cause narrative and citations.
6. Power BI dashboard shows live OEE, forecast curves, RL actions and autogenerated reports.

3. Modular design & implementation options

Module	Minimal MVP (skip RL)	Full version (with RL)
Forecasting	Prophet or XGBoost predicts defect probability every 15 min[1].	LSTM-Encoder-Decoder (60 min window, 128 cells) F1 ≈ 0.89 for early-defect flag.
Control	Rule-based SPC: if p(defect) > 0.7 apply preset "safe recipe"; else nominal.	PPO agent in custom Gym `PharmaQCEnv` (78-D state, multi-discrete action). Offline pre-train on SMPL simulator (1 M episodes), then on-line fine-tune with conservative Q-learning to respect safety-filter.
Gen AI	Batch-summary template filled with Jinja + Pandas.	GPT-4 Turbo RAG: vector-store (FAISS) of SOPs, batch e-logs, regulatory PDFs; hybrid question-answer retrieval delivers grounded report (QA-RAG scored F1 59 % vs 42 % for vanilla RAG).

4. Key technical details

Dataset – unchanged: 1 005 batches, 60 M sensor rows, defect labels (drug-release < 90 %, CU outside 6 %), plus cost tags.

Feature engineering

- STL imputation for 12 % missing compression-force series[1].
- Phase flag: startup / steady / wind-down.
- Rolling variance of API water over 30 min (strong 0.68 correlation with defects).

RL specifics

```
action_space = gym.spaces.MultiDiscrete([3, 7, 2]) # test freq, comp.force delta, alarm band
reward = 100*(1-defect_rate) - 5*test_cost + 50*reg_compliance - 10*downtime
safety_layer = OptLayer(constraints=['compression_force in (10,20)',
                                     'speed <= 180k'])
```

Offline RL via `d3rlpy ConservativeQLearning (CQL)`; switch to PPO once KL-divergence < 0.05 against expert policy.

RAG pipeline

```
docs = ingest_pdf_folder('FDA_Guidelines/')
index = FAISS.from_documents(docs, embeddings=OpenAIEmbeddings())
chain = RetrievalQA.from_chain_type(llm=GPT4Turbo,
                                    retriever=index.as_retriever(search_type='mmr'),
                                    chain_type='map_reduce')
```

5. Expected business impact (targets vs baseline)

KPI	Lab-centred QC	Smart Pharma Copilot	Δ
Defect detection	72 %	94 %	+22 pp[1]
Sampling cost	$18 k / batch	$12 k	–33 %[1]
Batch-release time	48 h	6 h	–87 %[1]
Report prep time	8 h	45 min	–90 %[2]

6. Demonstration plan (8-week sprint)

Week 1-2 Data cleaning & feature store

Week 3 Forecasting MVP, live dashboard

Week 4-5 Build and validate `PharmaQCEnv`, offline RL training

Week 6 Safety-gated on-line RL in simulation; A/B test vs SPC

Week 7 RAG fine-tune with 200 regulatory PDFs, integrate chat UI

Week 8 Dry-run “virtual plant day”: stream one historical batch, show RL interventions, see autogenerated FDA report.

7. If decide to skip RL

- Keep the forecasting “yellow-flag” model.
- Add Bayesian-Optimization tuner that optimizes compression force & speed subject to constraints every 30 min (surrogate model = Gaussian Process).

- Operators accept/reject suggestions; acceptance feedback becomes new labelled data for a future RL upgrade.

8. Tech stack

Python 3.11, PyTorch 2, d3rlpy/Ray RLlib, Prophet, LangChain, FAISS, Azure IoT Hub, Stream Analytics, Cosmos DB, Power BI, GitHub Actions CI/CD.

9. Why it stands out

- Combines three hot skills—time-series ML, safe RL and Gen AI RAG—rare in typical student projects yet directly aligned with pharma manufacturing pain-points.
- Dataset is public and already pre-labelled; no IP hurdles.
- Each module is decoupled, so we can deliver value even if RL is deferred.