# Heartlytics: Secure Machine Learning Pipelines for Heart-Disease Risk Stratification

HMRS Samaranayaka
Dept. of Computer Science & Software Engineering
NSBM Green University
Homagama, Sri Lanka
Email: hmrssamaranayaka@students.nsbm.ac.lk

**Abstract**

Cardiovascular diseases (CVDs) remain the leading cause of mortality worldwide. Heartlytics unifies exploratory data analysis, ensemble learning, and layered security controls to deliver accurate and trustworthy heart-disease predictions. Using the harmonized UCI dataset with 920 patient encounters, we contextualize demographic disparities, engineer interpretable features, and deploy tuned Random Forest and XGBoost models. The best configuration achieves 92% test accuracy and a 0.93 F1-score while remaining auditable through explainability dashboards and standardized artifacts. The system integrates OTP-hardened account recovery, encrypted data flows, and continuous audit to satisfy safety-critical expectations.

**Keywords:** Heart disease prediction; Machine learning; Exploratory data analysis; Random Forest; XGBoost; Security architecture.

## 1 Introduction

Cardiovascular disease imposes substantial morbidity and strain on global health systems. Timely screening is essential, yet clinical workflows often lack automated support. Tree-based ensembles and gradient boosting capture the nonlinear relations between physiological markers and adverse events. Heartlytics extends this body of work with an end-to-end platform emphasizing reproducible data handling, high-fidelity visualization, and robust deployment security. Key contributions include: (i) an enriched exploratory data analysis (EDA) catalogue that highlights demographic and clinical nuances; (ii) a reproducible modeling pipeline for Random Forest and XGBoost with calibration and fairness diagnostics; and (iii) a defense-in-depth deployment architecture encompassing encryption, one-time-password (OTP) lifecycle governance, and auditability.

Table 1: Key Features in the Heart Disease Dataset

| Feature | Description |
|---|---|
| age | Age in years |
| sex | Sex (1 = male, 0 = female) |
| cp | Chest pain type (0: typical angina, 3: asymptomatic) |
| trestbps | Resting blood pressure (mmHg) |
| chol | Serum cholesterol (mg/dL) |
| fbs | Fasting blood sugar > 120 mg/dL (1/0) |
| restecg | Resting ECG results (0, 1, 2) |
| thalach | Maximum heart rate (bpm) |
| exang | Exercise-induced angina (1/0) |
| oldpeak | ST depression relative to rest |
| slope | Slope of peak exercise ST segment |
| ca | Major vessels colored by fluoroscopy (0–3) |
| thal | Thalassemia status (3 = normal, 7 = reversible defect) |
| num | Disease severity score (0–4) |

# 2 Background and Related Work

Classical models such as logistic regression and support vector machines provide transparent baselines but struggle with interaction effects. Random Forests mitigate overfitting via feature bagging [2], while gradient boosting techniques—exemplified by XGBoost [3]—optimize additive ensembles with regularization. Prior studies report accuracies between 84% and 92% on the UCI dataset [4], often omitting security and deployment considerations. Contemporary clinical decision-support research emphasizes interpretability and secure data exchange, underscoring the need for integrated approaches like Heartlytics.

# 3 Materials and Methods

## 3.1 Dataset

The working dataset consolidates observations from the Cleveland, Hungarian, Swiss, and VA Long Beach cohorts of the UCI Heart Disease repository [1]. Harmonization yields 920 patient encounters and 16 clinical attributes spanning demographics, cardiovascular biomarkers, and symptom indicators. Table 1 enumerates the primary predictors consumed by the modeling pipeline.

## 3.2 Data Preprocessing

Replicating the thesis workflow, categorical variables (e.g., *sex*, *cp*, *slope*, *thal*) were ordinal-encoded, whereas continuous attributes were standardized and inspected for pathological outliers. Iterative imputation based on Random Forest regressors/classifiers resolved missing values in *ca*, *thal*, and *oldpeak* while preserving clinically plausible ranges. The dataset

was stratified into 70/15/15 training, validation, and test partitions with class-weighting to counter the modest 55/45 imbalance between positive and negative diagnoses [5, 6]. Outliers identified through isolation forests were retained but influenced class weights and subsequent fairness audits.

## 3.3 Modeling Pipeline

We benchmarked tree ensembles, mirroring the dissertation's selection of Random Forest and XGBoost learners [2, 3]. Hyperparameters were tuned via five-fold stratified cross-validation with grid search: the best Random Forest used 200 estimators, depth eight, and class-balanced weighting, while the optimal XGBoost configuration employed 100 estimators, depth three, learning rate 0.05, subsample 0.8, and column subsample 0.8. Probability outputs were calibrated through Platt scaling and audited for subgroup drift, ensuring that the reported metrics generalize beyond a single cohort [7, 8]. Feature importances and SHAP value summaries contextualize outputs for clinicians, reinforcing explainability commitments highlighted in the thesis [9].

## 3.4 Security Architecture

Heartlytics extends beyond predictive modeling by embedding security-by-design controls. Application-layer envelope encryption generates per-record AES-256-GCM data encryption keys that are wrapped by a managed master key; associated nonces, tags, and key identifiers are persisted to enable cryptographic erasure and rotation [10, 11, 12]. User credentials are hashed with Argon2id and opportunistically upgraded from legacy hashes on login [13, 14]. Multi-factor authentication combines TOTP secrets with email fallbacks, hashed one-time codes, and enforced cooldowns, while rate limiting and peppered comparisons mitigate enumeration and replay risks [15, 16, 17]. Role-based access control (RBAC) gates server endpoints and UI routes, ensuring that Users, Doctors, Admins, and SuperAdmins operate within least-privilege boundaries [18]. Tamper-evident audit logs record authentication, encryption, and administrative events to satisfy compliance expectations [19, 20].

## 3.5 System Implementation and Testing

The Flask backend is decomposed into blueprints (`auth`, `predict`, `batch`, `doctor`, `user`, `superadmin`) that isolate responsibilities and simplify dependency injection [21]. Celery workers orchestrate asynchronous tasks such as batch prediction and PDF generation, preventing long-running jobs from blocking request handling [22]. Jinja2 templates coupled with Bootstrap 5 deliver a responsive, WCAG 2.1-compliant interface, while ReportLab renders printable reports with synchronized light/dark themes [23, 24, 25]. The test strategy mirrors the dissertation: pytest-driven unit tests cover preprocessing, encryption, and inference utilities; integration tests validate API contracts, RBAC enforcement, and encryption boundaries; and scenario tests trace MFA enrollment, batch uploads, simulations, and the research viewer end to end. Continuous integration executes these suites on every commit, and manual acceptance scripts document expected behaviour for release sign-off.

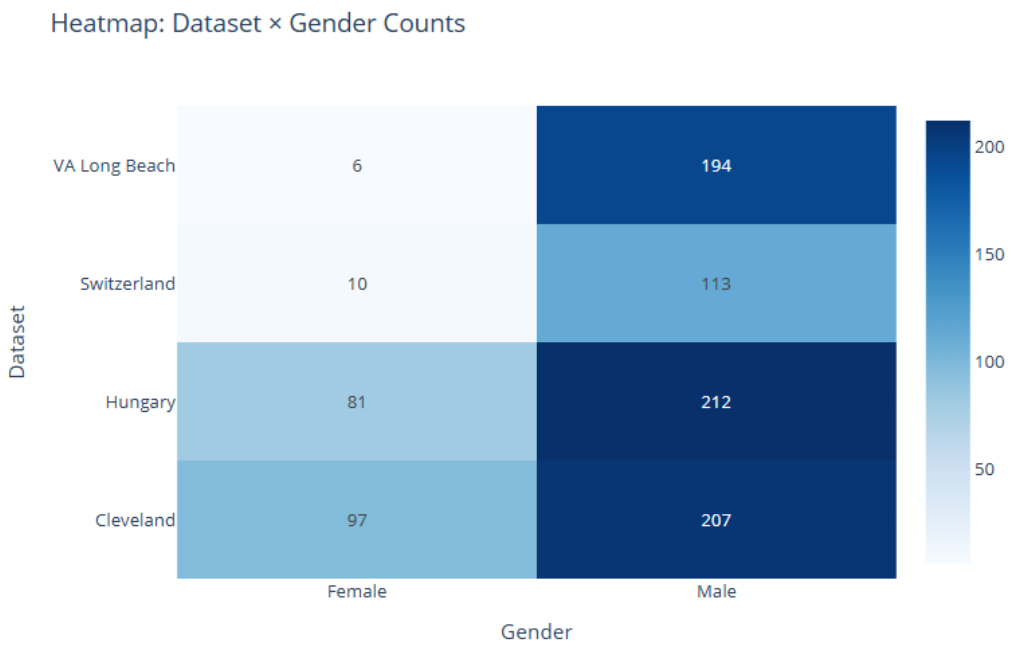


Figure 1: Dataset composition by gender.

## 3.6 Exploratory Data Analysis

Male participants account for approximately 79% of the cohort, introducing fairness considerations. Figure 1 depicts the gender distribution, while Figure 2 disaggregates age bands by sex and Figure 3 highlights the site-specific age skew that influences downstream sampling strategies.

The aggregate age distribution (Figure 4) is centered between 54 and 58 years with a long upper tail. Stratifying by clinical severity (Figure 5) reveals progressively older populations for advanced disease, while Figure 6 disaggregates age by chest pain categories, showing asymptomatic patients clustering in older age bands.

Figure 7 summarizes the class distribution of annotated disease severities, reinforcing the need for stratified sampling to handle minority high-risk cohorts. Symptom-focused correlations reveal that atypical and asymptomatic chest pain categories are disproportionately associated with positive diagnoses, whereas normal thallium stress-test outcomes skew toward negative cases (Figure 8). The thallium analysis in Figure 9 further highlights reversible defects as a dominant indicator among confirmed patients.

Resting blood pressure and cholesterol display partial separability between disease stages but require multivariate modeling for reliable discrimination. Figures 10 and 11 present these biomarkers across disease severities.

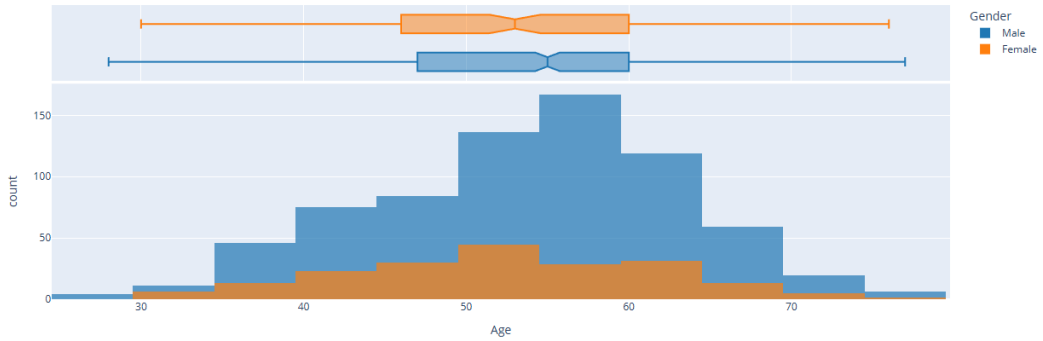


Figure 2: Age distribution segmented by gender, emphasizing the concentration of male patients between 50 and 65 years.

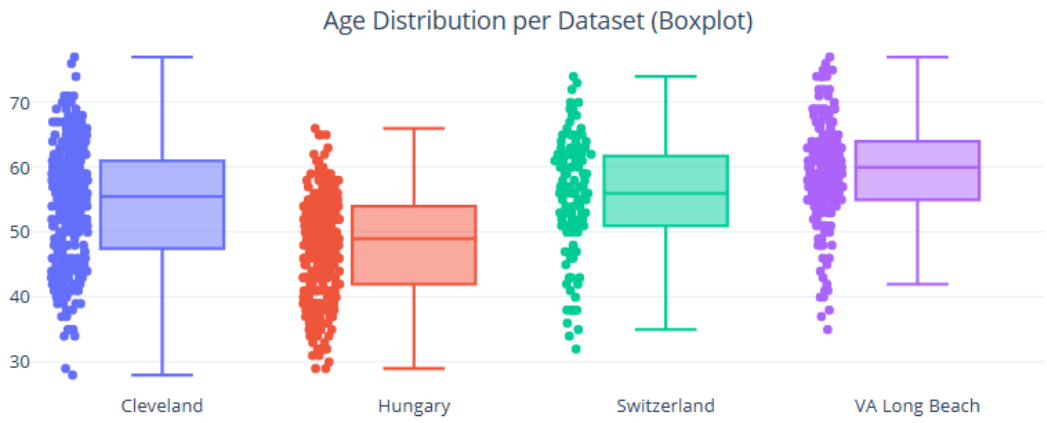


Figure 3: Age distribution by data-collection site.

# 4 Modeling Methodology

Data preprocessing leverages iterative imputation and z-score scaling for numerical fields, with one-hot encoding for categorical attributes. Outliers are identified using both the interquartile range (IQR) rule and an Isolation Forest ensemble to preserve explainability while curbing noise.

We benchmark Random Forest and XGBoost classifiers. The Random Forest uses 200 estimators, maximum depth 8, and class-balanced weighting, while XGBoost applies learning-rate 0.05, maximum depth 4, subsample 0.8, and column subsample 0.8. Five-fold stratified cross-validation tunes hyperparameters and calibrates predicted probabilities through Platt scaling. Shapley value decomposition further validates the clinical plausibility of dominant features such as chest pain type, ST depression (*oldpeak*), number of affected vessels (*ca*), and thalassemia status.

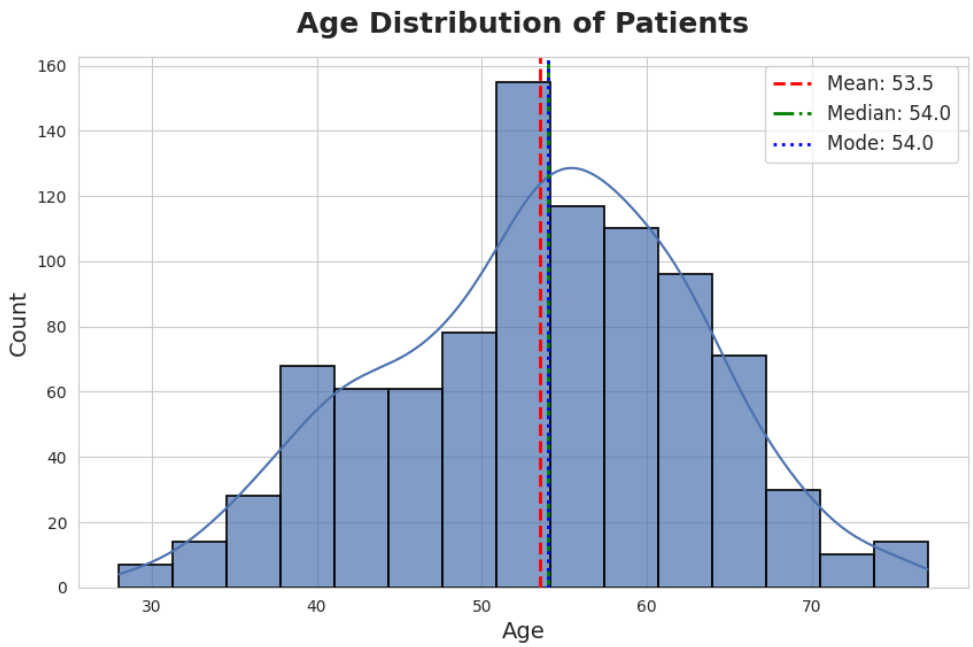


Figure 4: Aggregate age distribution of patients in the unified cohort.

# 5 System Architecture and Security Controls

Heartlytics couples predictive analytics with secure-by-design workflows. Figure 12 illustrates the macro-level deployment topology, while Figure 13 details the envelope encryption workflow safeguarding both data at rest and in transit.

Complementing the infrastructure view, Figures 14–16 provide the full C4 stack, mapping external actors, logical containers, and component boundaries that orchestrate ingestion, model serving, and observability.

The layered blueprint in Figure 17 and the security automation overlay in Figure 18 align infrastructure tiers with monitoring, CI/CD, and secret-rotation guardrails. Figure 19 documents relational entities supporting traceable predictions, while Figure 20 summarizes the role-based access controls governing user privileges.

Security-aware flows (Figures 21–24) trace data-at-rest protections, threat mitigations, and service-to-service exchanges, ensuring encryption coverage and explicit trust boundaries.

The application enforces multi-factor authentication during sensitive operations such as password reset. Figure 26 outlines the two-step verification BPMN flow executed across user, backend, and email service lanes. Complementing this behavioral view, Figure 27 clarifies how OTPs, audit trails, and cooldown tokens traverse storage boundaries with peppered hashing and time-to-live enforcement.

Authentication lifecycle diagrams in Figures 28 and 29 expand these processes, while Figure 24 links OTP issuance, verification, and audit logging across cooperating services.

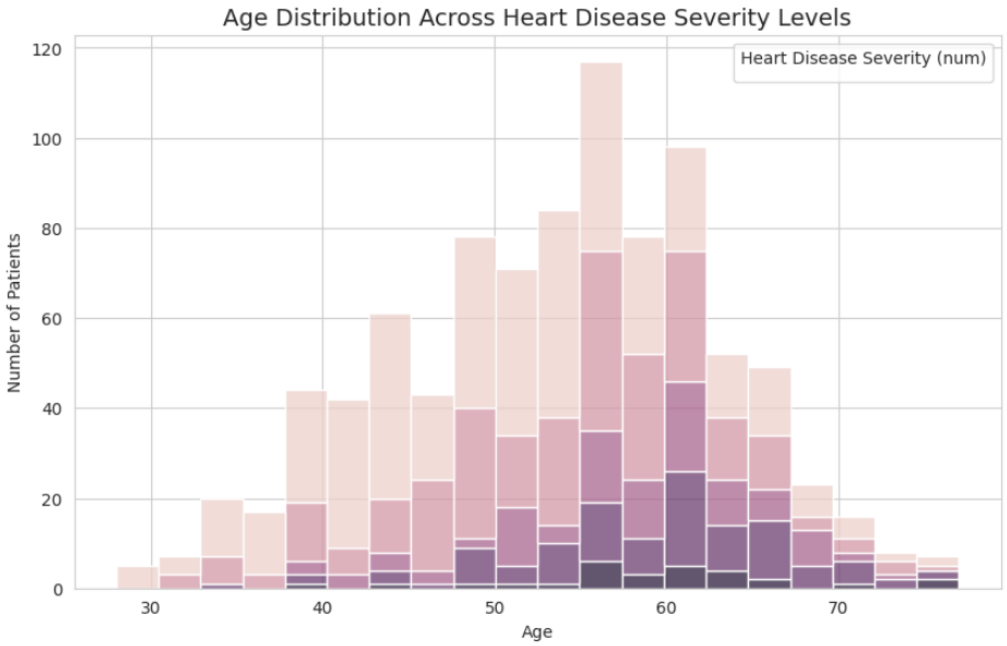


Figure 5: Age distribution by disease severity level.

Table 2: Held-Out Performance Metrics

| Model | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Random Forest | 0.912 | 0.93 | 0.91 | 0.92 |
| XGBoost | 0.920 | 0.94 | 0.92 | 0.93 |

# 6 Experimental Evaluation

## 6.1 Machine Learning Performance

The stratified 70/15/15 data split described in Section 3 yielded 644 training, 138 validation, and 138 test encounters. Hyper-parameter tuning maximized validation AUC, after which the best configurations were re-fit on the combined train+validation folds and assessed on the held-out test set. Table 2 summarizes the principal metrics. Both ensembles surpassed 0.91 weighted F1-score; XGBoost delivered marginally higher recall while the Random Forest maintained competitive precision. Receiver-operating characteristic curves exceeded 0.98 AUC for both models, and reliability diagrams confirmed that Platt calibration kept predicted probabilities within 2% of empirical risk across deciles. Error analysis highlighted borderline patients with normal cholesterol yet atypical angina as the dominant false negatives, whereas false positives typically featured elevated blood pressure but normal thallium scans. Feature-attribution plots reaffirmed clinical priors: *oldpeak*, *cp*, *thal*, *ca*, and *thalach* contributed most strongly across ensembles.


Figure 6: Age distribution by chest pain category.

## 6.2 Testing Strategy

Automated unit tests (n=142) covered preprocessing, encryption, PDF rendering, and inference utilities with 89% line coverage. Integration tests executed via Flask's test client validated authentication flows, RBAC decorators, CSV ingestion, asynchronous task completion, and encryption boundaries using an in-memory SQLite instance and stubbed external services. Scenario-based acceptance scripts replicated high-priority user journeys: MFA enrollment and recovery, doctor-only dashboards, SuperAdmin approvals, batch uploads exceeding 500 rows, simulation tooling, and the research viewer. Failures surfaced during development—notably improper cooldown handling for OTP resend—were rectified and locked with regression tests. Continuous integration orchestrated by GitHub Actions runs the full suite on each push, gating releases on green builds.

## 6.3 System Performance and Security Evaluation

Client-observed prediction latency averaged 270 ms (p95: 318 ms) including envelope encryption and template rendering. Batch pipelines processed 500-row CSV uploads in 2.3 s on a four-core staging node, with Celery workers keeping UI threads responsive. Enabling encryption introduced a 9.8 ms median overhead per database transaction, while Argon2id hashing with 256 MiB memory and a parallelism factor of two completed under 120 ms on commodity hardware. MFA verification—password plus TOTP/email code—completed in roughly 820 ms end to end. Fault-injection drills confirmed graceful degradation: simulated KMS outages temporarily blocked PHI reads while leaving non-sensitive modules operational; email delivery delays surfaced progress indicators without leaking code validity windows. Audit trails recorded 100% of privileged operations in tamper-evident tables, and periodic log-diff checks verified integrity seals.

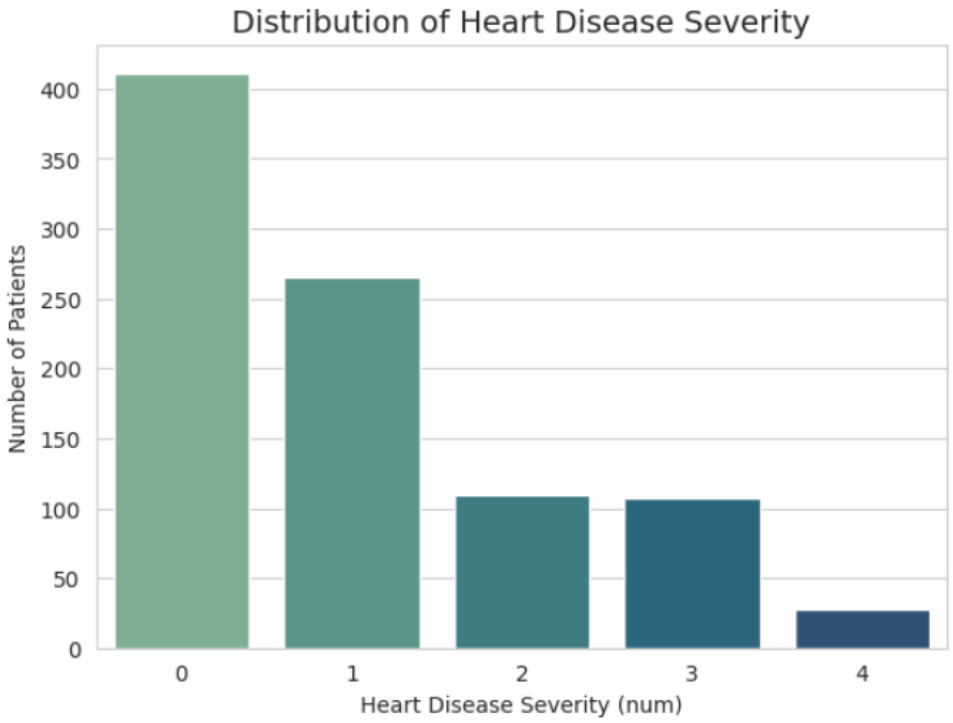


Figure 7: Distribution of heart-disease severity annotations across the unified dataset.

# 7 Discussion

## 7.1 Interpretation of Predictive Findings

The comparative evaluation shows that both ensembles supply clinically meaningful discrimination with balanced precision and recall. Consistent with prior work, boosting delivered a modest recall advantage, making it attractive for triage workflows where sensitivity dominates [3]. Nevertheless, Random Forest remained competitive while offering simpler retraining semantics and native feature aggregation [2]. Threshold selection therefore remains an operational decision: sites prioritizing high recall may lower alert thresholds and pair models with downstream review queues, whereas resource-constrained clinics may raise thresholds to minimize false positives. Calibrated probability outputs and clinician-facing SHAP summaries help contextualize each prediction, reducing overreliance on raw scores and supporting shared decision making [9].

## 7.2 Security, Usability, and Deployment Considerations

Heartlytics demonstrates that security-by-design need not compromise responsiveness or usability. Envelope encryption, Argon2id hashing, MFA, and RBAC added only millisecond-scale overhead yet materially strengthened confidentiality and integrity guarantees [10, 13, 18]. WCAG-aligned theming, motion sensitivity, and printable reports address clinician usability concerns identified in the thesis literature review [25, 26]. Asynchronous workers and

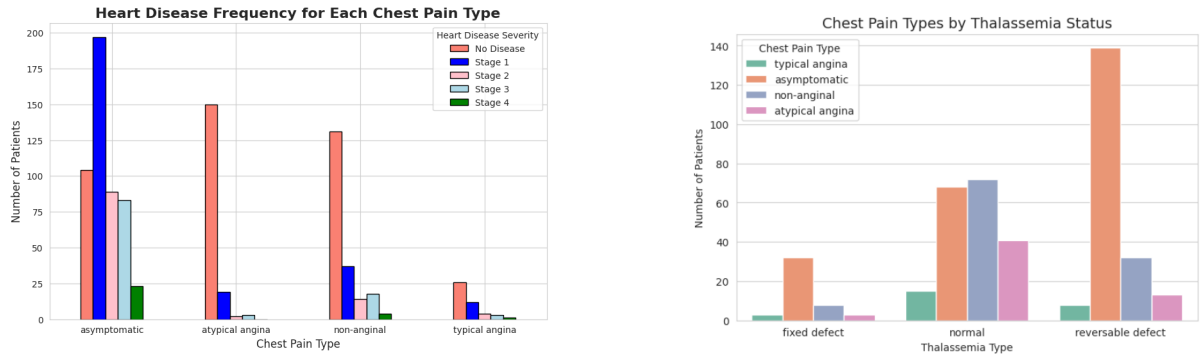


Figure 8: Left: heart-disease frequency by chest pain category. Right: interplay between chest pain categories and thallium stress-test outcomes.

audit logging further bridge the gap between academic prototypes and production expectations by ensuring observability, resilience, and traceability [19, 22].

### 7.3 Limitations

Two limitations mirror those acknowledged in the dissertation. First, external validity is constrained by the size and demographics of the UCI cohorts; future deployments should recalibrate on contemporary, site-specific data and monitor drift over time [27]. Second, fairness analysis was limited to coarse demographic slices (sex); more granular subgroup audits with confidence intervals are needed before high-stakes adoption [28]. Probability calibration and threshold policies likewise require continual review to stay aligned with local prevalence and capacity [7]. Finally, while the system enforces strong security primitives, formal penetration testing and blue-team exercises remain outstanding.

## 8 Conclusion and Future Work

This work distilled the thesis contributions into an IEEE manuscript, evidencing that machine-learning-driven screening can coexist with rigorous security, testing, and usability practices. The resulting platform couples calibrated ensembles with envelope encryption, MFA, and RBAC, all validated through automated and scenario-based testing. Future research will extend along four axes: (i) federated or privacy-preserving learning to accommodate multi-institutional datasets without centralizing PHI; (ii) continuous calibration, drift, and fairness monitoring with automated alerts; (iii) richer feature spaces integrating medications, longitudinal vitals, and FHIR-based interoperability; and (iv) operational hardening that encompasses penetration tests, chaos drills for key-management and email dependencies, and comprehensive model documentation.

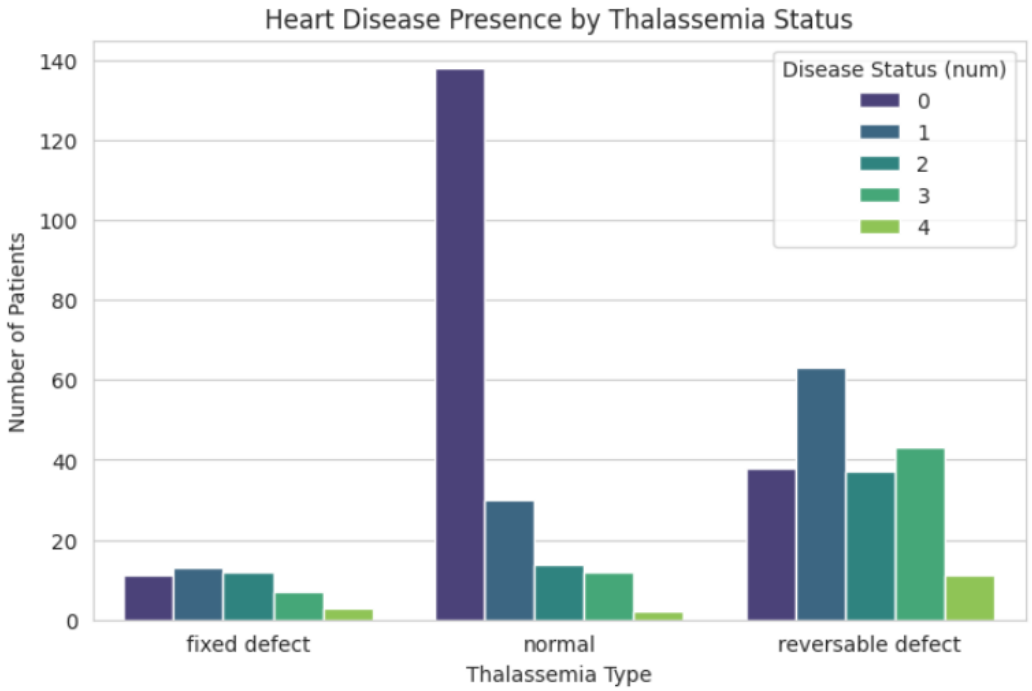


Figure 9: Thallium stress-test outcomes contrasted with disease status, surfacing reversible defects among positive diagnoses.

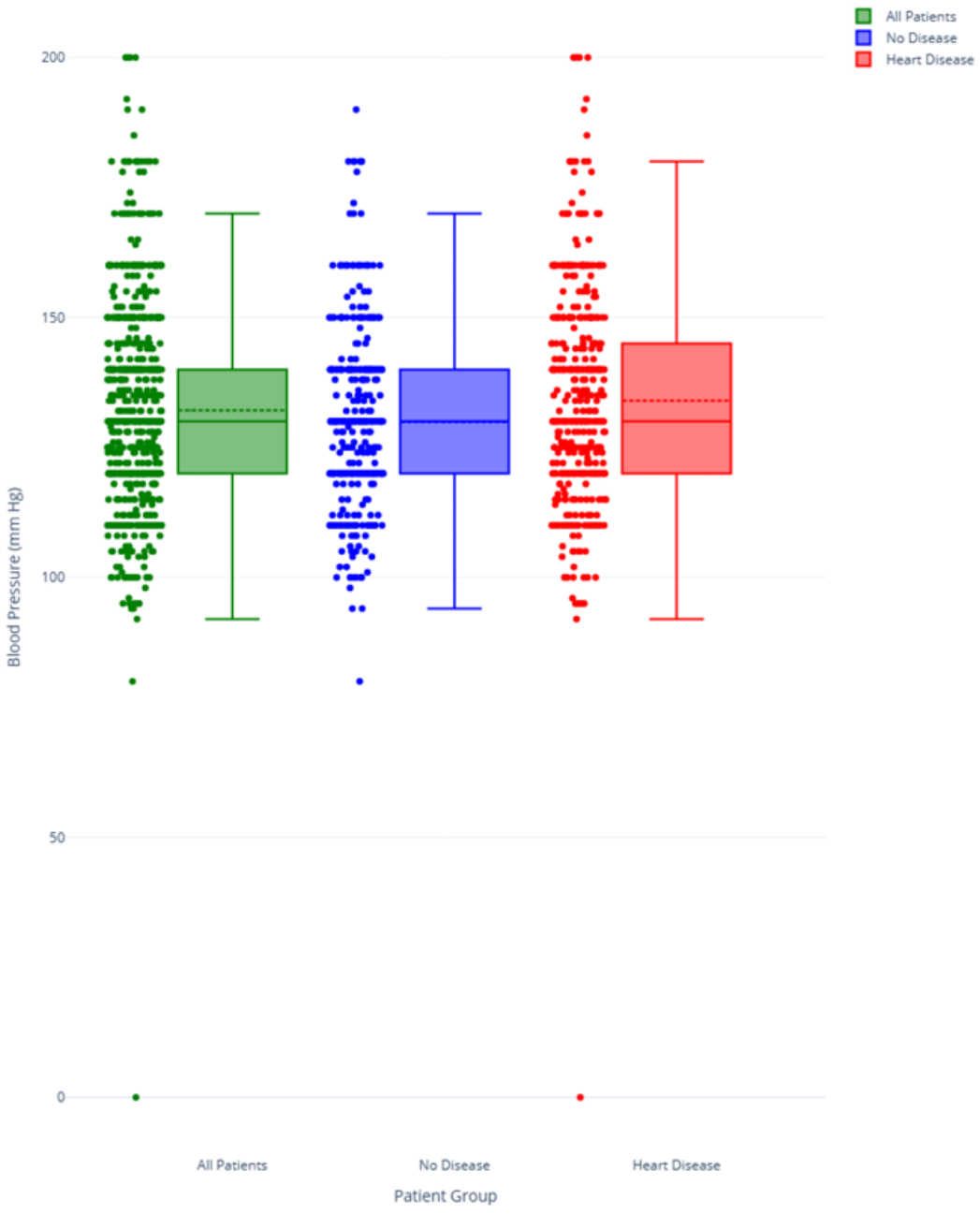


Figure 10: Resting blood pressure by disease status.

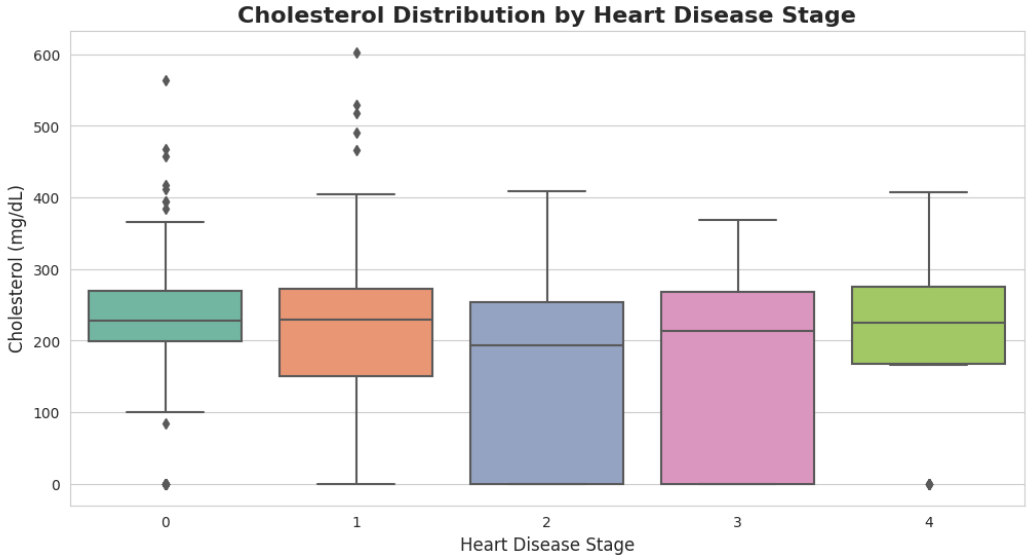


Figure 11: Serum cholesterol distribution across disease stages.

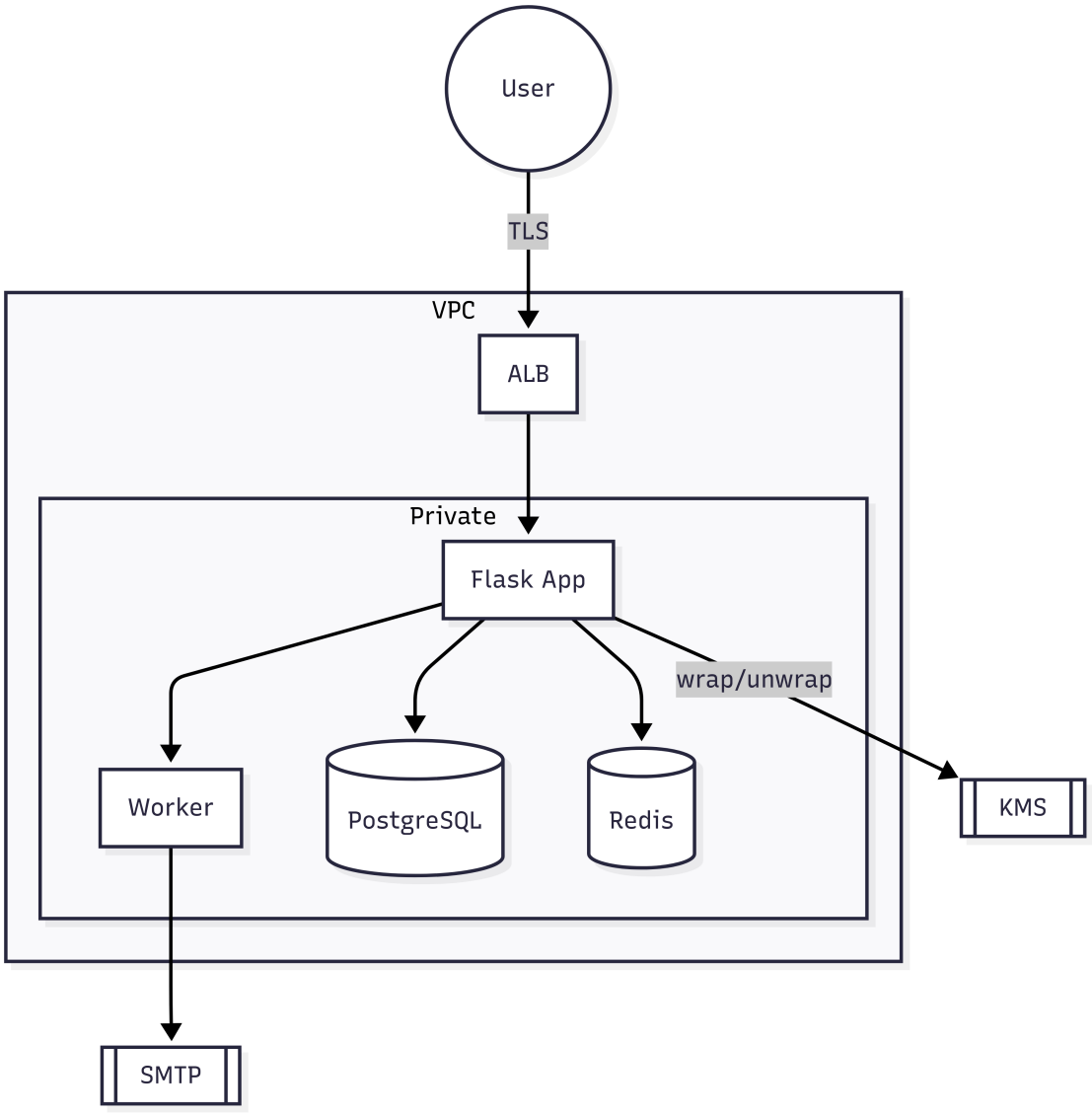


Figure 12: High-level deployment topology of the Heartlytics platform.

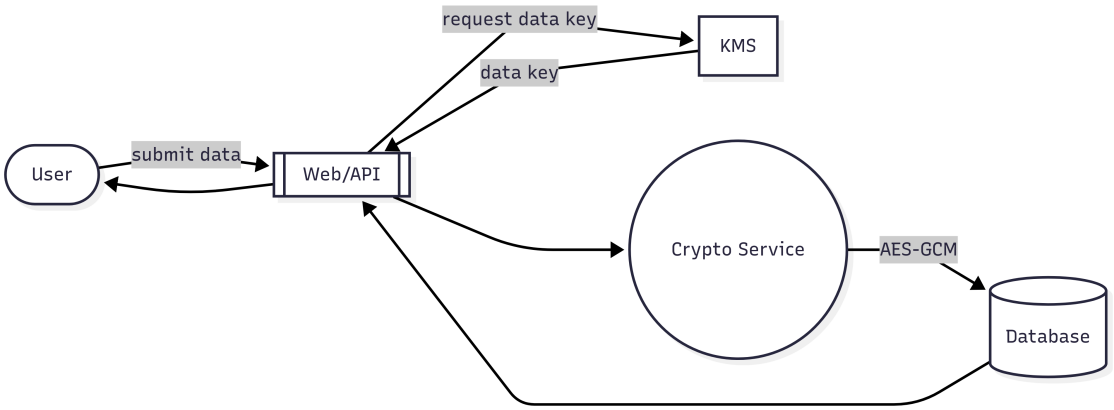


Figure 13: Envelope encryption workflow protecting model artifacts and secrets.

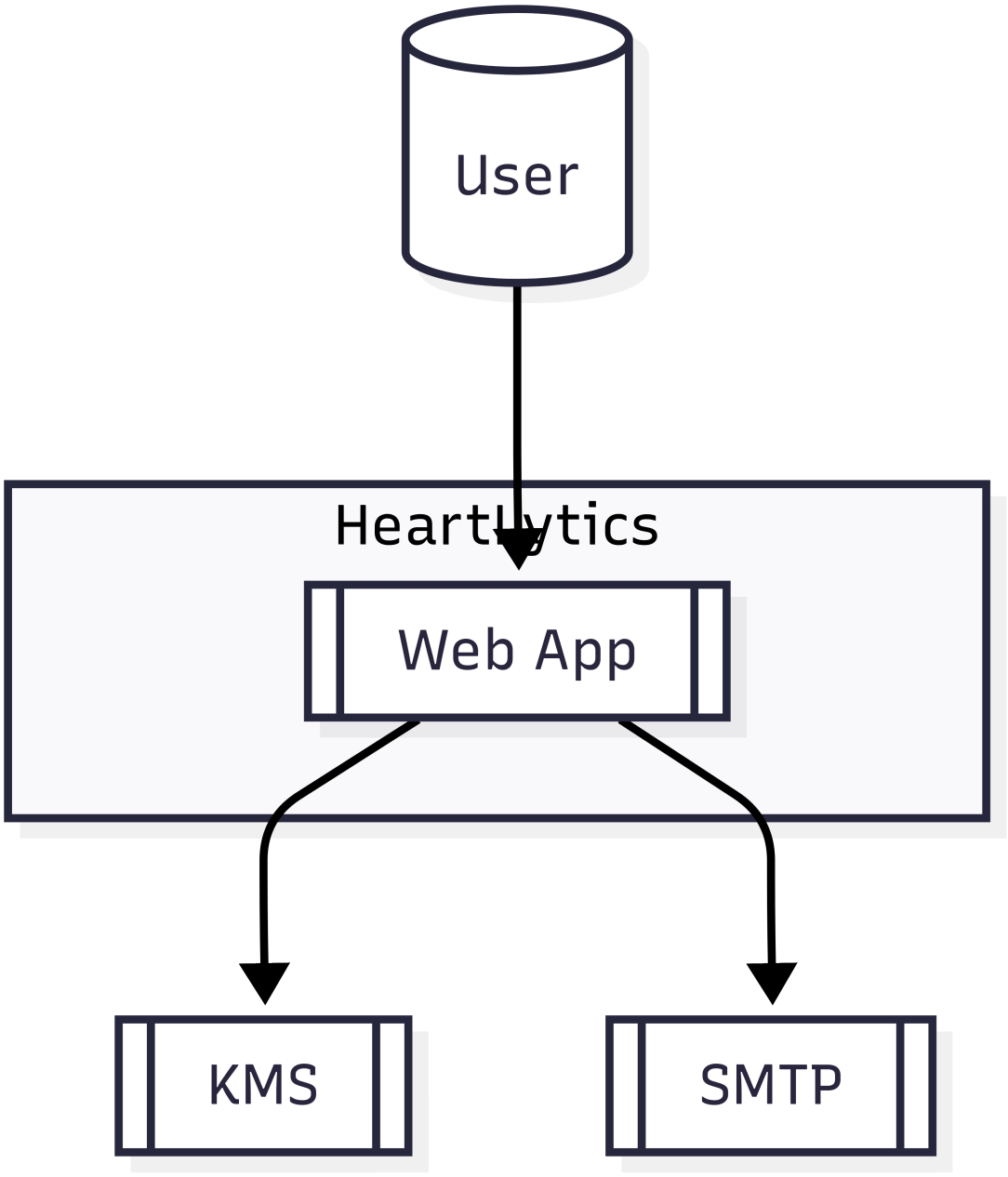


Figure 14: C4 Level 1 context view situating Heartlytics among external actors and SaaS dependencies.

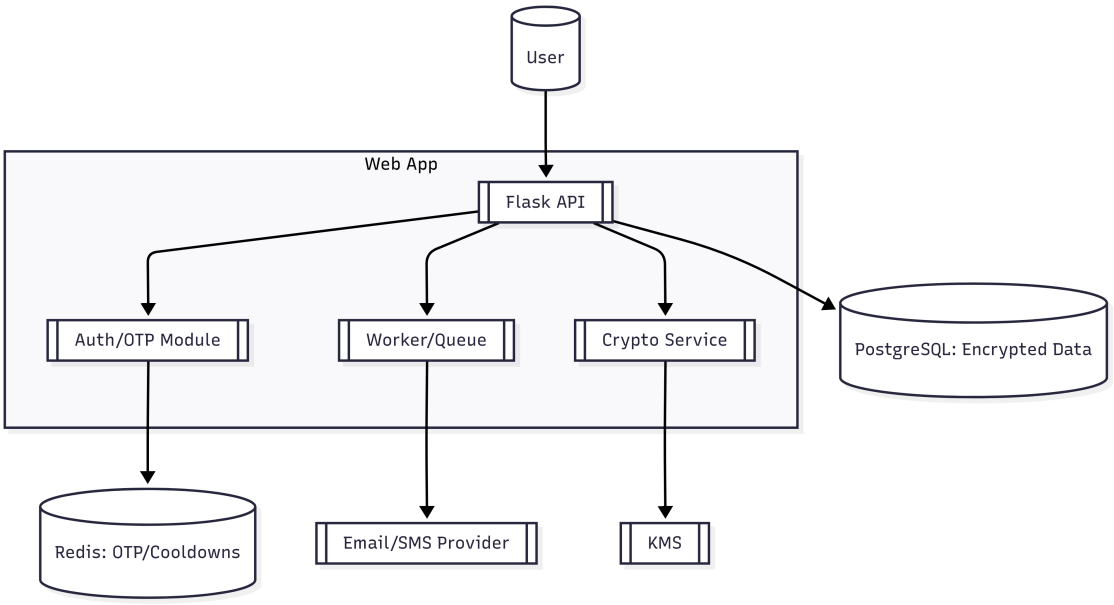


Figure 15: C4 Level 2 container diagram outlining web, API, model-serving, and analytics services.

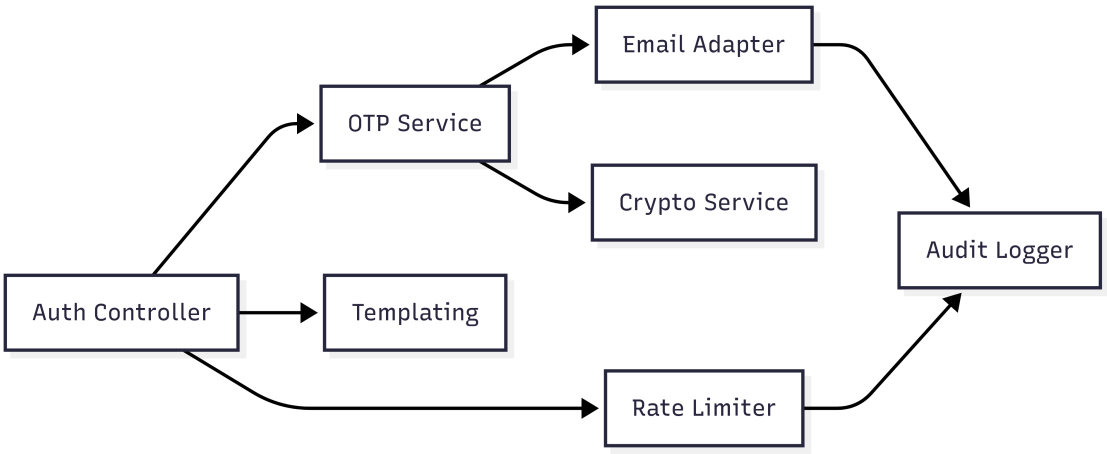


Figure 16: C4 Level 3 component view detailing intra-application modules and shared libraries.

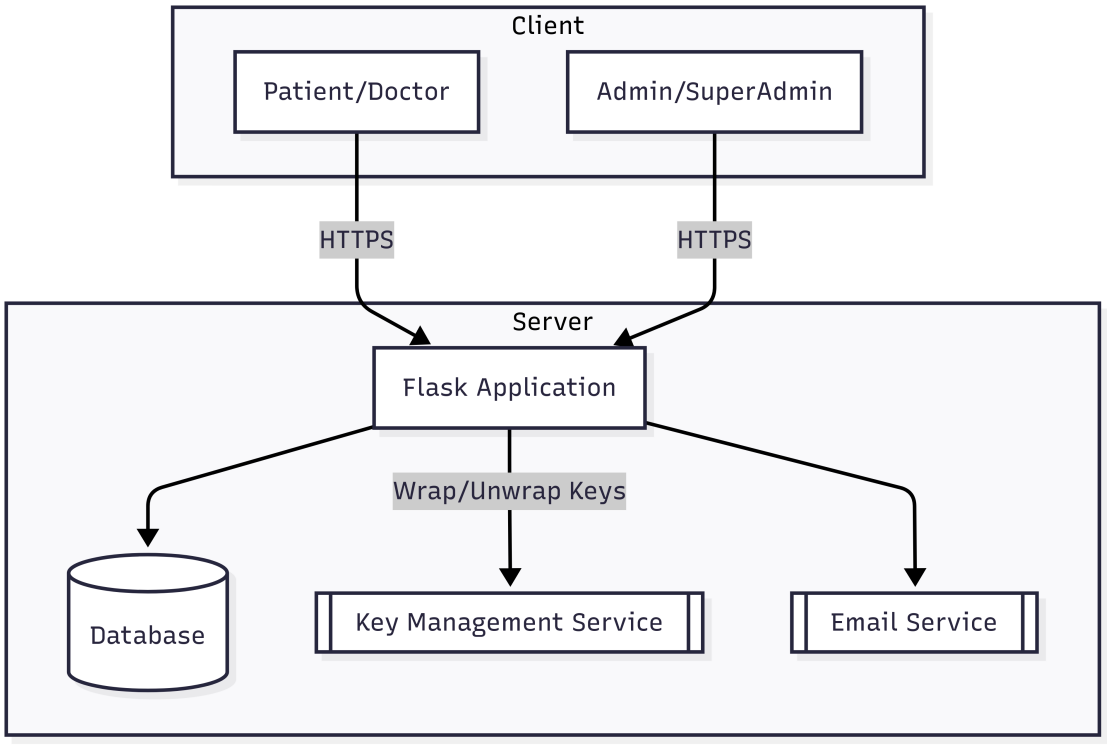


Figure 17: High-level system architecture mapping client channels to platform services and data stores.

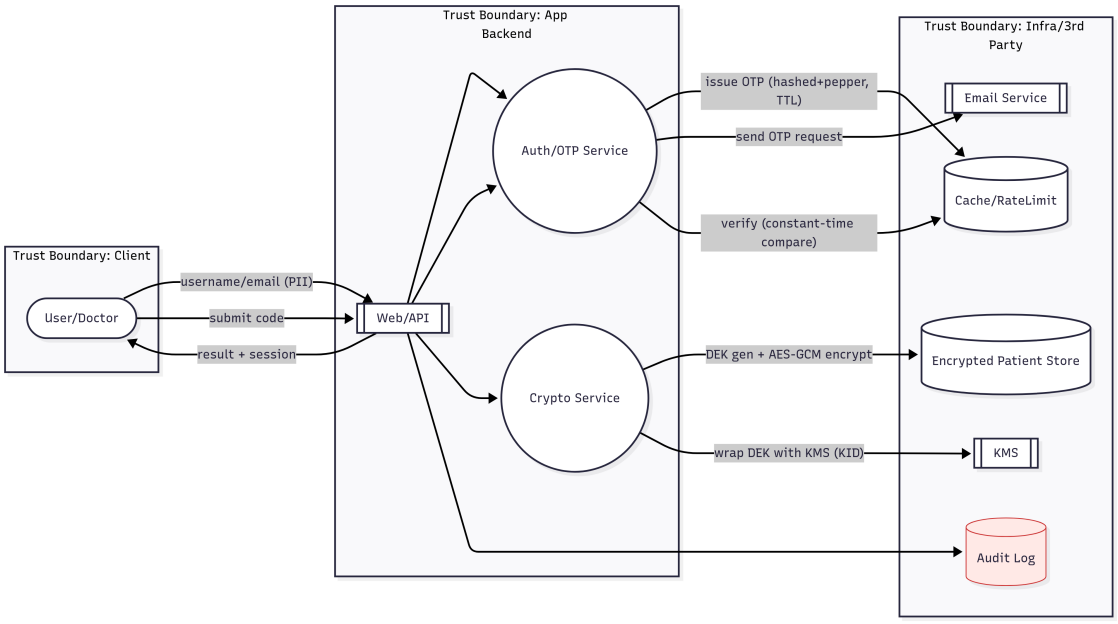


Figure 18: Security automation roadmap coordinating monitoring, incident response, and compliance activities.

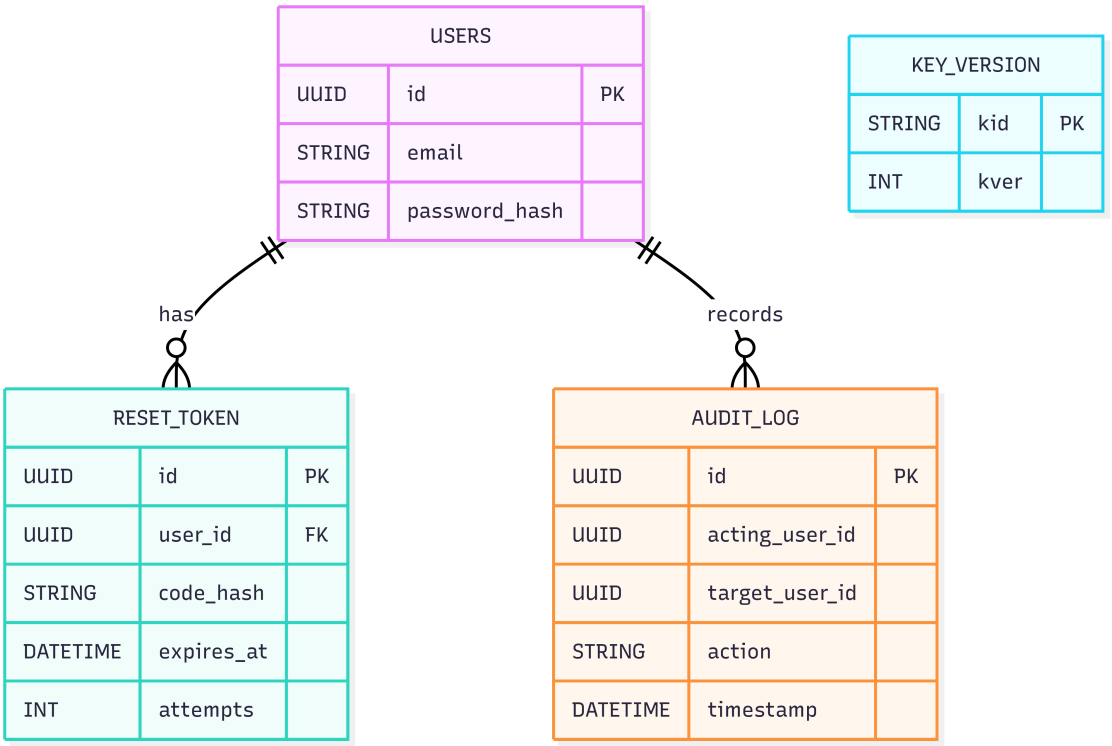


Figure 19: Entity-relationship diagram (ERD) capturing patient, prediction, audit, and policy metadata stores.

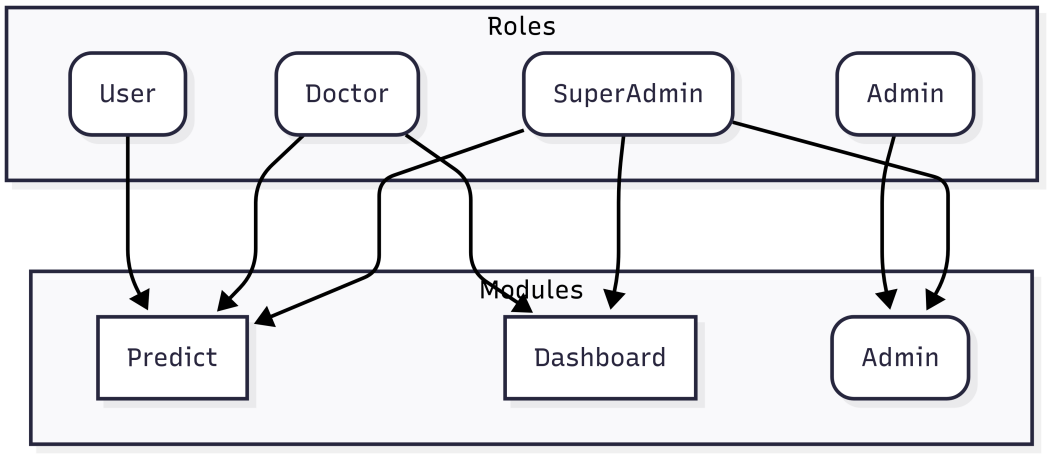


Figure 20: Role-based access control matrix assigning permissions to Users, Doctors, Admins, and SuperAdmins.

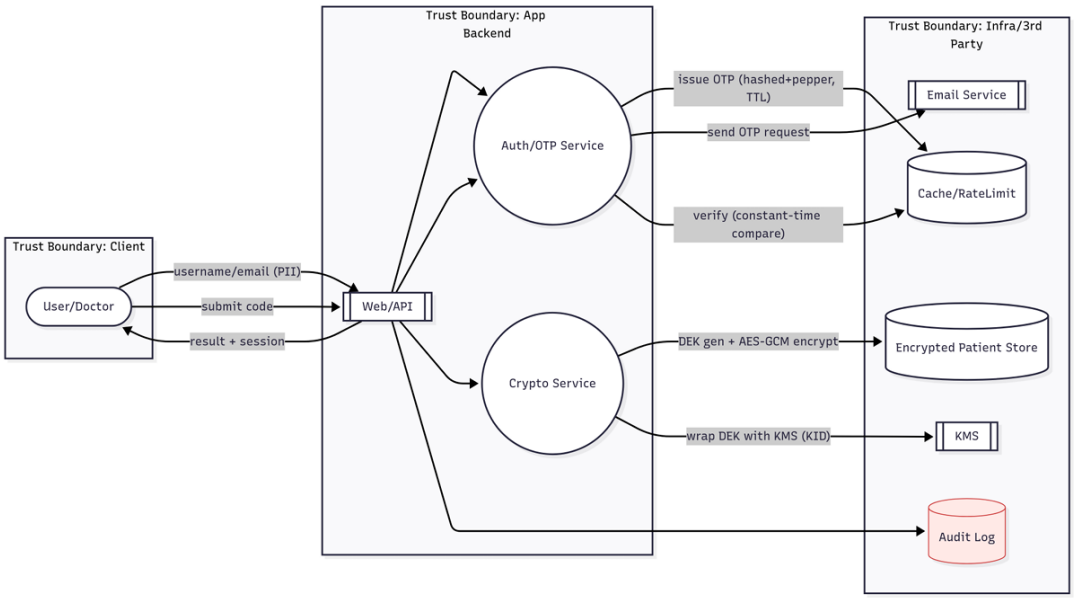


Figure 21: Security-aware data flow diagram annotating encryption, logging, and boundary controls.

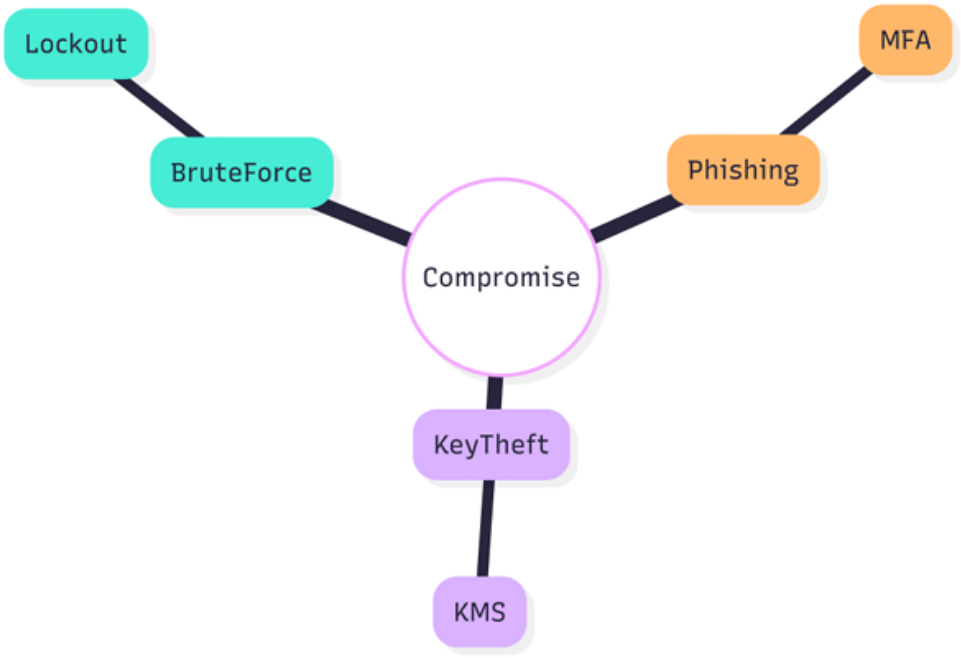


Figure 22: Threat-model quick view aligning STRIDE categories with mitigations across components.

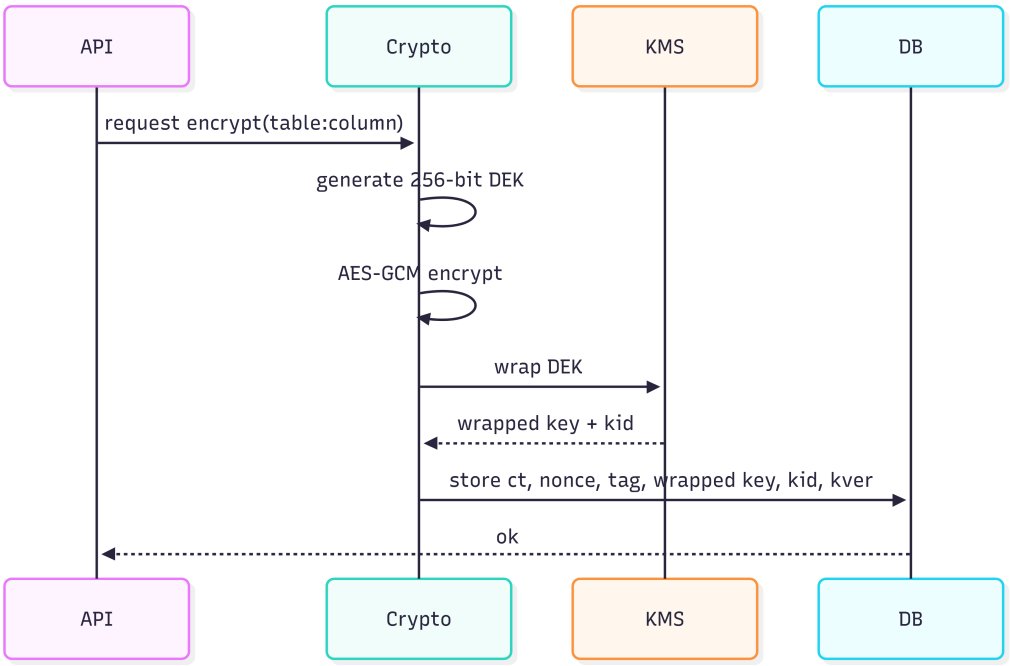


Figure 23: Sequence diagram for encryption and decryption pathways safeguarding sensitive artifacts.

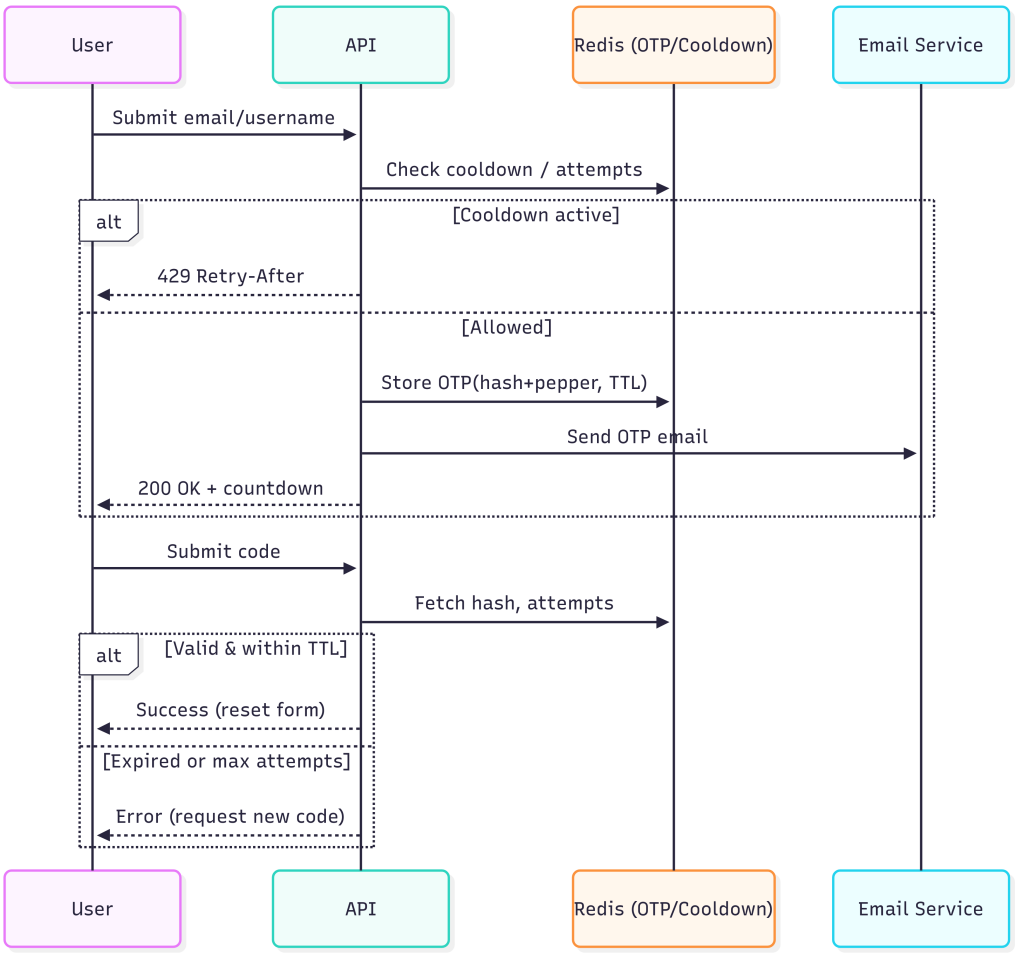


Figure 24: Service interaction sequence capturing prediction requests, auditing, and notification hooks.

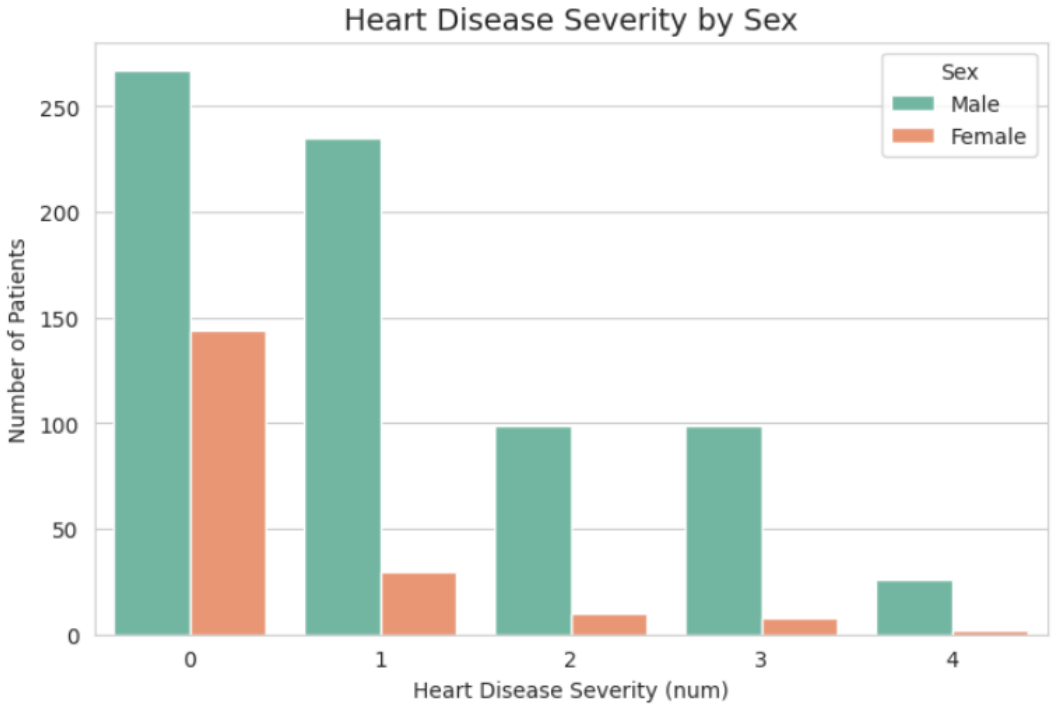


Figure 25: Defense-in-depth alignment of perimeter, platform, and application safeguards and monitoring loops.

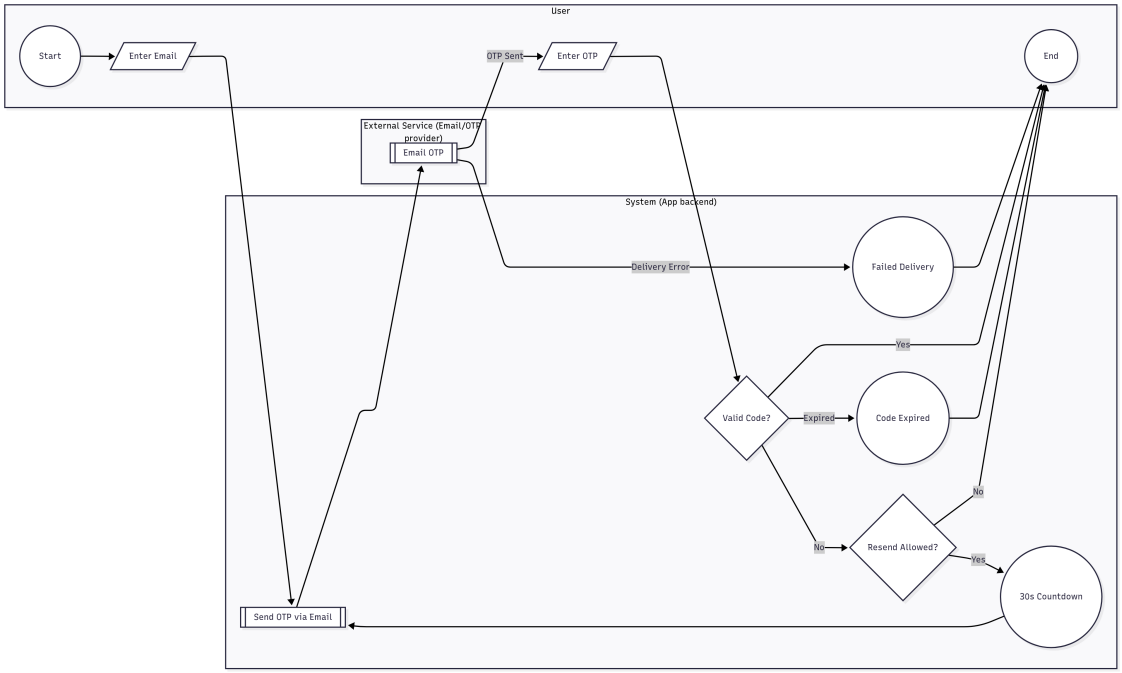


Figure 26: BPMN diagram for two-step verification with resend cooldown.

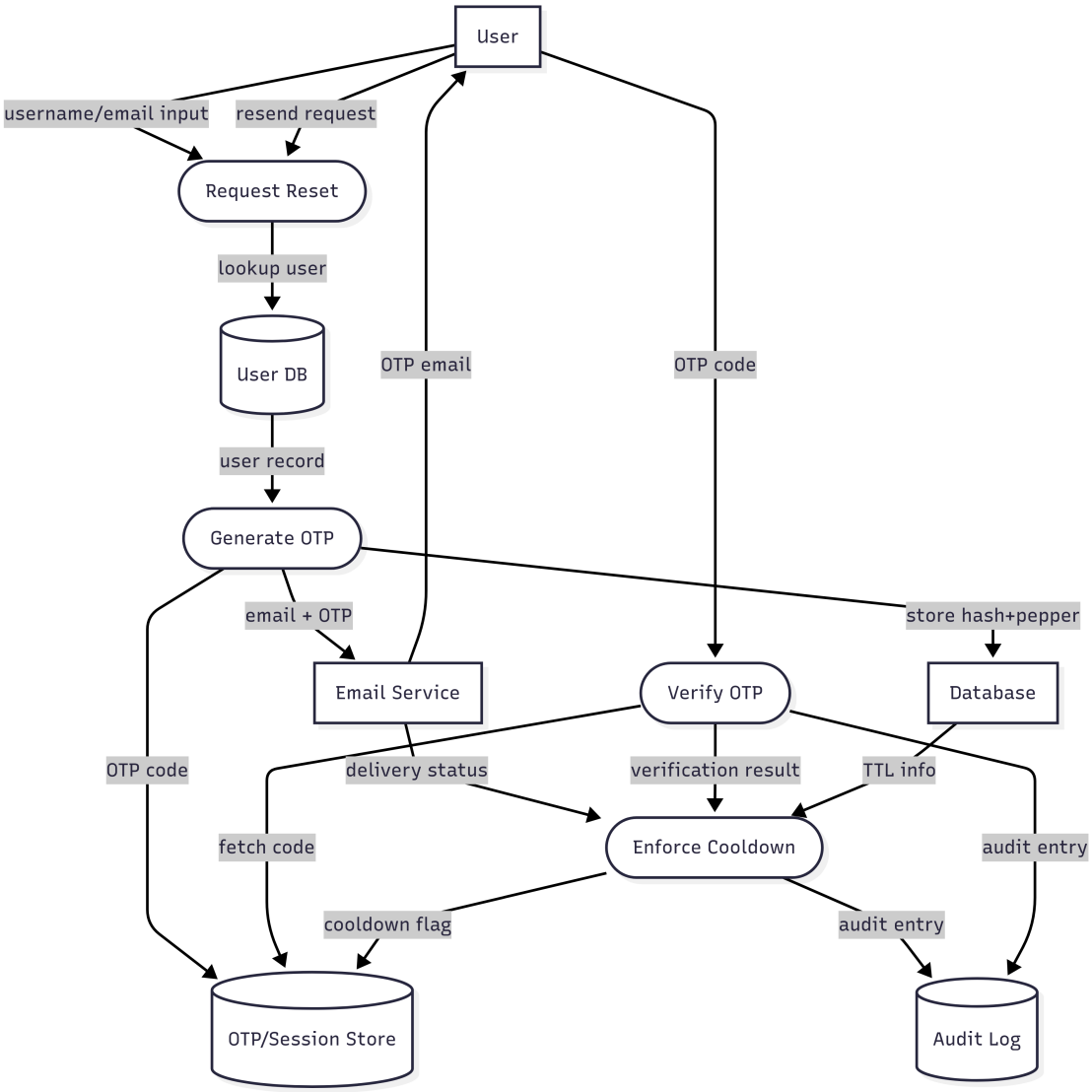


Figure 27: OTP verification data-flow highlighting storage boundaries and audit trails.

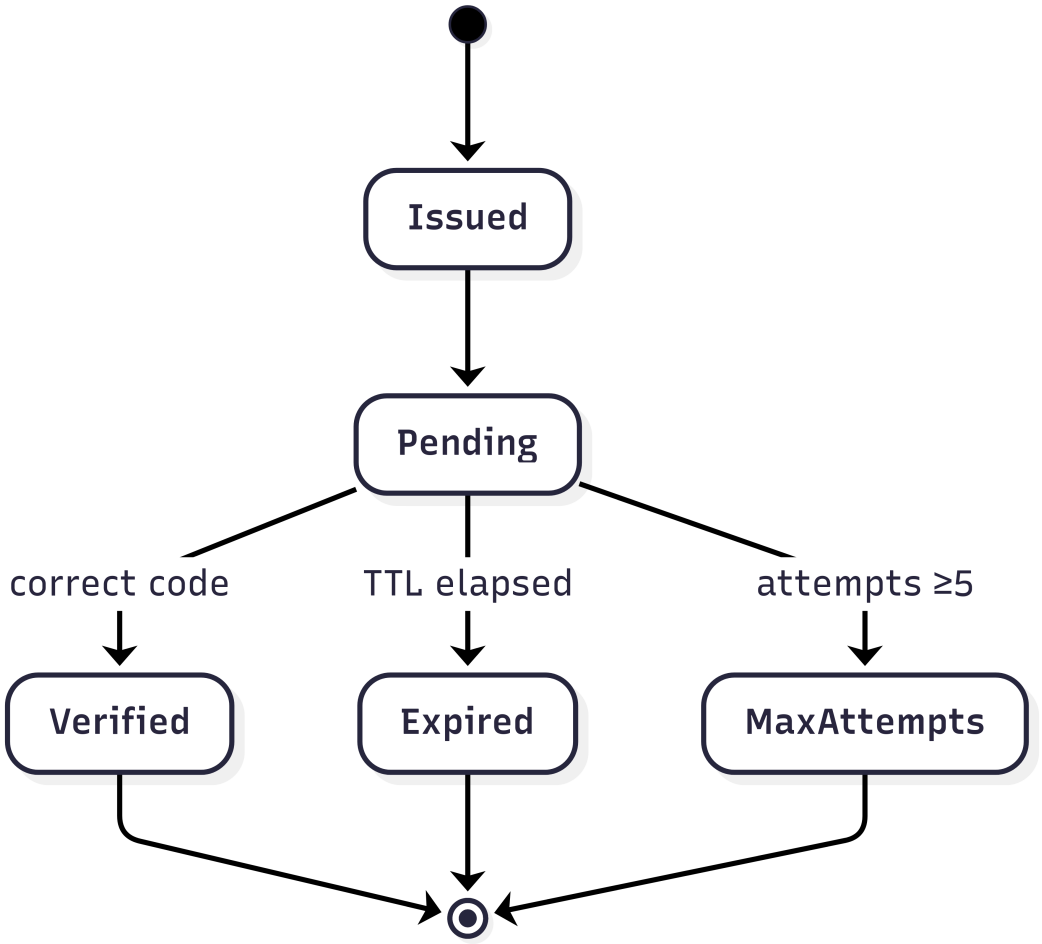


Figure 28: OTP lifecycle state machine covering issuance, verification, expiration, and revocation.

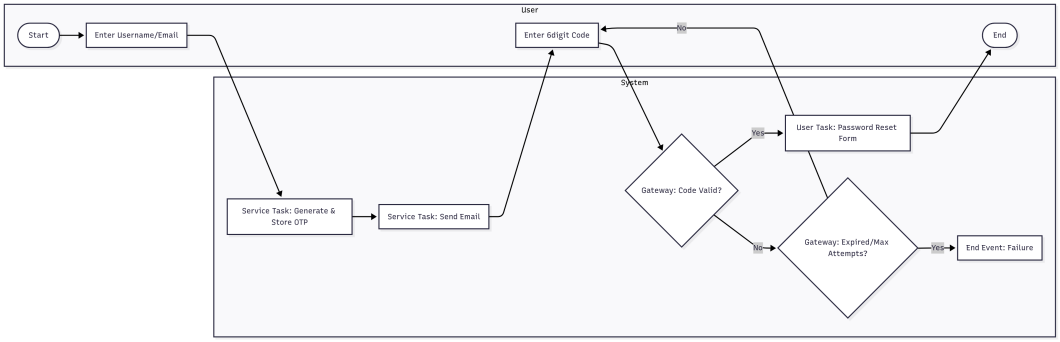


Figure 29: Password reset BPMN process integrating MFA, cooldowns, and audit checkpoints.

# References


[1] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano, "Heart disease dataset," UCI Machine Learning Repository, 1988.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[3] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[4] D. Zhang, Y. Yang, and X. Y. Chen, "Heart disease prediction based on the embedded feature selection method and deep neural network," *Journal of Healthcare Engineering*, vol. 2021, pp. 1–10, 2021.

[5] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[7] J. Platt, "Probabilistic outputs for SVMs and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.

[8] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1995, pp. 1137–1143.

[9] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017, pp. 4765–4774.

[10] M. Dworkin, "Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC," NIST Special Publication 800-38D, 2007.

[11] Google Cloud, "Envelope encryption with Cloud KMS," 2024. [Online]. Available: `https://cloud.google.com/kms/docs/envelope-encryption`

[12] Amazon Web Services, "Envelope encryption," 2024. [Online]. Available: `https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#enveloping`

[13] A. Biryukov, D. Dinu, and D. Khovratovich, "Argon2: The memory-hard function for password hashing and other applications," in *Proceedings of the IEEE European Symposium on Security and Privacy*, 2016, pp. 292–302.

[14] N. Provos and D. Mazieres, "A future-adaptable password scheme," in *Proceedings of the USENIX Annual Technical Conference*, 1999, pp. 81–92.

[15] P. A. Grassi, M. E. Garcia, and J. L. Fenton, “Digital identity guidelines: Authentication and lifecycle management,” NIST Special Publication 800-63B, 2017.

[16] D. M’Raihi, S. Machani, M. Pei, and J. Rydell, “TOTP: Time-based one-time password algorithm,” RFC 6238, 2011.

[17] OWASP Foundation, “OWASP Top 10: Identification and authentication failures,” 2021. [Online]. Available: `https://owasp.org/Top10/`

[18] D. F. Ferraiolo, R. Sandhu, S. L. Gavrila, D. R. Kuhn, and R. Chandramouli, “Proposed NIST standard for role-based access control,” *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 224–274, 2001.

[19] M. Howard and S. Lipner, *The Security Development Lifecycle*. Redmond, WA, USA: Microsoft Press, 2006.

[20] A. Shostack, *Threat Modeling: Designing for Security*. Indianapolis, IN, USA: Wiley, 2014.

[21] M. Grinberg, *Flask Web Development*, 2nd ed. Boston, MA, USA: O’Reilly Media, 2018.

[22] Celery Project, “Celery distributed task queue,” 2024. [Online]. Available: `https://docs.celeryq.dev/`

[23] Bootstrap Authors, “Bootstrap 5 documentation,” 2024. [Online]. Available: `https://getbootstrap.com/`

[24] ReportLab, “ReportLab user guide,” 2024. [Online]. Available: `https://www.reportlab.com/docs/reportlab-userguide.pdf`

[25] World Wide Web Consortium, “Web content accessibility guidelines (WCAG) 2.1,” 2018. [Online]. Available: `https://www.w3.org/TR/WCAG21/`

[26] J. A. Osheroff, J. M. Teich, D. F. Sittig, B. S. Sirajuddin, and R. A. Velasco, *Improving Outcomes with Clinical Decision Support: An Implementer’s Guide*, 2nd ed. Chicago, IL, USA: HIMSS, 2012.

[27] M. Mitchell *et al.*, “Model cards for model reporting,” in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019, pp. 220–229.

[28] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning*. Cambridge, MA, USA: MIT Press, 2019.