Risk Assessment: Handling PII in Housing Application System

Purpose

To evaluate the privacy, security, and compliance risks of two architectural approaches for processing personally identifiable information (PII) when assessing housing risk scores.

Use Case Overview

- A housing applicant submits personal data via a NestJS app.
- This data must be anonymized before risk scoring.
- Two options are being compared:

Option 1: Anonymize in NestJS (Before Sending)

- Raw applicant data is anonymized inside the API layer.
- Only anonymized data is sent to the microservice for risk scoring.

Option 2: Send Raw PII to a Cleaning Microservice

- Raw applicant data is sent to a separate microservice.
- That microservice performs anonymization and then passes anonymized data to the risk scoring engine.

Risk Comparison Table

Category	Option 1: Anonymize in NestJS	Option 2: Clean in Microservice
Data Exposure Risk	🔒 Low – PII stays within the API boundary	⚠️ Medium–High – PII leaves the original context
Compliance (GDPR/CPRA)	✅ Strong – aligns with data minimization principles	⚠️ Riskier – needs stronger justification and controls
Attack Surface	🔐 Smaller – fewer components handle PII	🚨 Larger – multiple services handle raw data
Auditability	✅ Easier – clear PII access path	⚠️ More complex – multiple services must be audited
Implementation Complexity	✅ Lower – centralized logic	⚠️ Higher – requires secure service communication
Modularity/Reusability	⚠️ Slightly less reusable	✅ Better – shared anonymizer logic possible
Responsibility Clarity	✅ Clear – NestJS owns full responsibility	⚠️ Blurred – shared ownership across services
Breach Impact	✅ Contained – limited exposure scope	⚠️ Higher – microservice breach exposes PII
Encryption Needs	✅ Basic HTTPS sufficient	✅ Requires transport + internal encryption

---

Summary & Recommendation

Category	Preferred Option
Privacy Compliance	✅ Option 1
Security Risk Mitigation	✅ Option 1
Simplicity / Maintainability	✅ Option 1
Modularity	✅ Option 2 (only if needed)

Final Recommendation:

Implement **Option 1: Anonymize inside the NestJS API** before sending any data to microservices. This minimizes exposure, complies with data protection laws, and simplifies auditing.

If Option 2 is required for reuse or architectural flexibility, ensure the following:

- Full encryption of inter-service data
- Access control and logging on the microservice
- Justification and documentation for compliance (e.g., GDPR Article 5, CPRA 1798.100(c))
- Data processing agreements between internal services (if applicable)

References

- GDPR Article 5 – Principles of Data Processing
- CPRA Section 1798.100(c) – Data Minimization
- Troutman Pepper: Data Minimization Under the CCPA
