StegoSafe: Secure Image Steganography in Java

Project: StegoSafe — Java | LSB steganography | OOP + JDBC

Presented by :

Shriyanshi Srivastava - 24SCSE1180199

Harshita Singh - 24SCSE1180154

Naveen Kumar - 24SCSE1180217

Foundations: OOP & LSB Steganography

Architectural OOP decisions and concise LSB encoding workflow for StegoSafe Java

1 **OOP concepts applied**

- **Encapsulation**: StegoEngine, Persistence, UI encapsulate state and expose minimal APIs
- **Inheritance & Polymorphism**: Abstract ImageCarrier with PNG/BMP subclasses for polymorphic I/O
- **Single Responsibility**: UI, stego algorithm, encryption, JDBC separated for maintainability

2 **LSB steganography steps (concise)**

- **Read carrier image**: ImageIO → BufferedImage; preserve header/offset for BMP
- **Create user space**: copy to modifiable BufferedImage via createGraphics/drawRenderedImage
- **Prepare payload**: Encrypt message; serialize filename, name length, payload length, then bytes
- **Encode bits into LSBs**: write payload bits into LSB of successive R/G/B color bytes

Design Goals & Non-Functional Requirements for StegoSafe

Architectural principles guiding security, portability, maintainability, performance, and usability

Security (confidentiality & stealth) — Encrypt payload before embedding; use PNG to avoid lossy transforms; preserve header bytes to reduce detection.

Portability — Pure Java (JDK 1.7+), Swing UI, ImageIO APIs for Windows/Linux cross-platform support.

Maintainability — OOP separation: StegoEngine, ImageCarrier, CryptoUtil, Persistence for clear responsibilities.

Performance — Linear O(n) embedding/extraction; accept trade-off for PNG I/O and BufferedImage memory usage.

Usability — Simple home UI with embed/decode flows, import/export, and preference read from DB.

High-Level System Architecture: Modules & Interactions

StegoSafe Component Map — LSB steganography, symmetric crypto, JDBC persistence

UI (Swing)

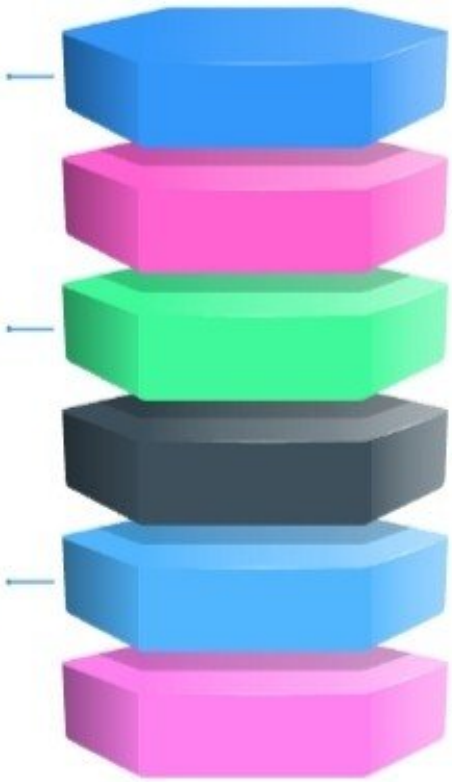
Accepts user input and displays embed/decode flows; calls **StegoEngine** and **Persistence**.

ImageCarrier

Format-specific read/write for **PNG/BMP**; returns/accepts BufferedImage byte arrays.

Persistence (JDBC)

Stores jobs, metadata, user prefs, operation logs via connection pool CRUD.



StegoEngine

Core **LSB** embed/extract logic and payload framing; orchestrates image and crypto operations.

CryptoUtil

Symmetric message encryption/decryption; exposes encrypt(byte), decrypt(byte).

Utils

Validation, capacity checks, and file I/O helpers used across modules.

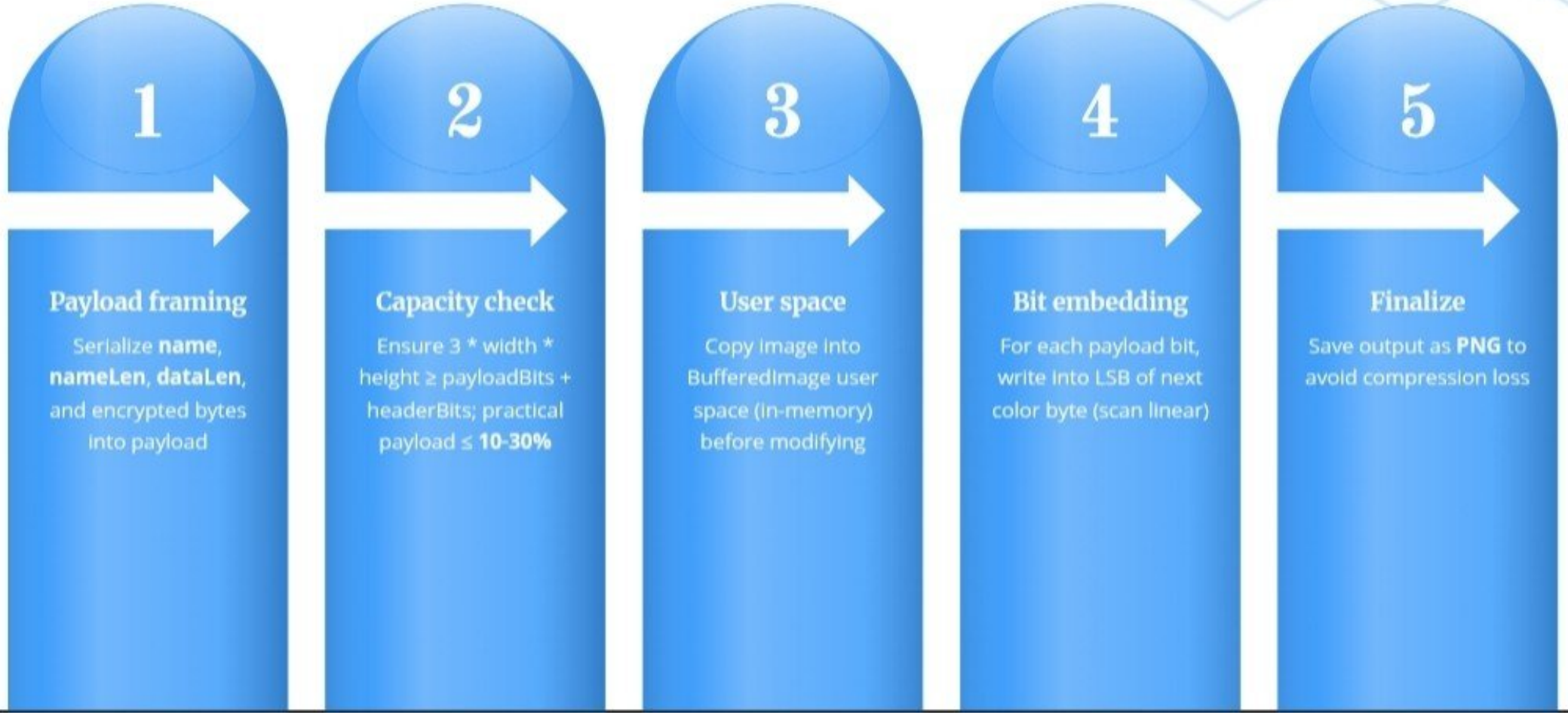
OOP Class Design: Key Classes, Roles, and Method Signatures

Clear mapping of responsibilities for StegoSafe Java (LSB steganography, JDBC, Swing)

Class	Role	Key methods / fields
StegoEngine	Embedding/extraction coordinator	embed(BufferedImage,imgPath,byte payload); extract(BufferedImage)
ImageCarrier (abstract)	Format adapter	read(File):BufferedImage; write(BufferedImage,File)
PngCarrier, BmpCarrier	Format implementations	override read/write; preserve header/offset
CryptoUtil	Encrypt/decrypt payload	encrypt(byte key, byte data); decrypt(byte key, byte data)
PersistenceDao	JDBC metadata and logs	saveJob(JobMeta); findById(id); initPool()
UiController	Connects Swing UI to backend	onEmbedClick(); onDecodeClick()

LSB Embedding & Extraction Flow

Compact step sequence, capacity rule, and complexity notes for StegoSafe Java LSB module



JDBC Persistence: Schema, Transactions, and Queries

Schema overview, transaction strategy (HikariCP), security notes and sample queries

Table	Columns (PK)	Purpose
jobs	job_id (PK), user_id, filename, stego_path, status, created_at	Track embed/decode operations and file paths
prefs	user_id (PK), default_format, last_dir	Store UI preferences (read-only for Home module)
logs	log_id (PK), job_id (FK), level, message, timestamp	Operation diagnostics and errors
Transaction & pooling	HikariCP; single-statement transaction per job	Ensure atomic metadata writes
Sample queries	INSERT INTO jobs(...); SELECT * FROM jobs WHERE status='FAILED' ORDER BY created_at DESC;	Common operations: insert, failure audit
Security	Store only metadata; payload optional	Do not persist payload bytes unless configured and encrypted at rest

Demonstration: Carrier vs Stego Images & Code Mappings

Visual parity at 1x, LSB embedding of a 10 KB encrypted payload into a 1024×768 PNG

Visual comparison

Carrier image (PNG, 1024×768)

Stego image after embedding **10 KB encrypted text** — visually identical at 1x

Detection note: visual check won't find LSB; use histogram or steganalysis for statistical anomalies

Code mapping (key snippets)

```
BufferedImage img = ImageIO.read(file); // ImageCarrier.read()
```

```
byte payload = CryptoUtil.encrypt(key, data); // prepare payload
```

```
stegoEngine.embed(img, payload); // bitwise LSB loop
```

Best practice: use PNG and limit payload to preserve low perturbation

Next Steps & Contribution Guide — StegoSafe

Week-by-week milestones, contributor checklist, and demo artifacts for a secure LSB steganography Java project

Week 1

- JDBC pooling & schema
- Complete JDBC pooling; finalize schema migrations

Week 2

Testing & CI

- Add integration tests and CI; expand unit coverage to 90%

Week 3

- Encryption & keys
- Integrate configurable encryption algorithms; secure key storage

Week 4

Demo package

- Prepare sample PNGs, encrypted payloads, step-by-step runbook