LeakWatch – High Level Design (HLD)

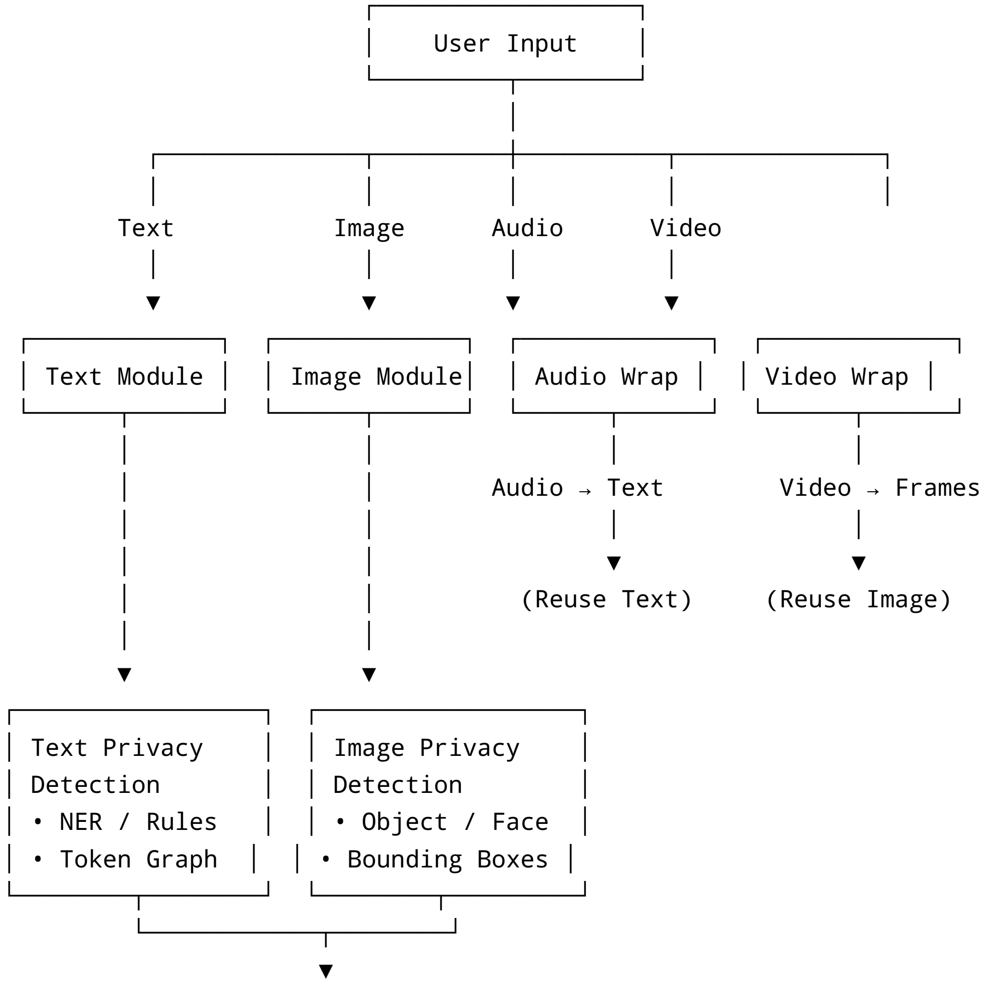
1. System Overview

LeakWatch is a privacy-preserving middleware designed to detect and mitigate privacy leaks in multimodal user content before it is processed by AI systems or shared on social platforms. The system operates as an intermediate layer that analyzes incoming data, identifies sensitive information, applies appropriate mitigation strategies, and produces a safe output along with an audit trail.

The framework supports multiple modalities including **text, images, audio, and video**, with full implementation focus on text and image modalities in Phase 2. Audio and video are handled through lightweight wrappers and reuse of existing modules.

---

2. High-Level Architecture Diagram (Conceptual)



```
                         ┌──────────────────┐
                         │    User Input    │
                         └─────────┬────────┘
                                   │
                                   │
          ┌──────────────┬─────────┼──────────┬──────────────┐
          │              │         │          │              │
         Text          Image     Audio      Video
          │              │         │          │
          ▼              ▼         ▼          ▼
   ┌──────────────┐ ┌──────────────┐ ┌──────────────┐ ┌──────────────┐
   │ Text Module  │ │ Image Module│ │ Audio Wrap   │ │ Video Wrap   │
   └──────┬───────┘ └──────┬───────┘ └──────┬───────┘ └──────┬───────┘
          │                │                │                │
          │                │          Audio → Text     Video → Frames
          │                │                │                │
          │                │                ▼                ▼
          │                │           (Reuse Text)     (Reuse Image)
          │                │
          ▼                ▼
  ┌──────────────────┐ ┌──────────────────┐
  │ Text Privacy     │ │ Image Privacy    │
  │ Detection        │ │ Detection        │
  │ • NER / Rules    │ │ • Object / Face  │
  │ • Token Graph  │ │ • Bounding Boxes │
  └────────┬─────────┘ └────────┬─────────┘
           └──────────┬──────────┘
                      ▼
```

```
┌─────────────────────────────┐
│ Unified Detection Output    │
│ (JSON: entity, location,    │
│  confidence)                │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│ Orchestration Layer         │
│ • Route results             │
│ • Decide mitigation flow    │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│ Mitigation Module           │
│ • Text masking              │
│ • Image blur / inpaint      │
│ • Audio mute / beep         │
│ • Video frame mitigation    │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│ Explainability & Audit      │
│ • Highlights / heatmaps     │
│ • Logs (what, why, where)   │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│ Safe Output + Audit Trail   │
└─────────────────────────────┘
              │
              ▼
┌─────────────────────────────┐
│ Target AI / Social Platform │
└─────────────────────────────┘
```

3. Module Descriptions

3.1 Input & Pre-processing Module

- Accepts raw user input in different modalities
- Performs basic validation, resizing, and format normalization
- Routes input to the appropriate modality-specific module

3.2 Text Privacy Detection Module

- Tokenizes input text
- Applies NER and rule-based detection for sensitive entities
- Constructs a token-level graph
- Outputs detected sensitive tokens with confidence scores

3.3 Image Privacy Detection Module

- Detects objects and faces using object detection models
- Extracts bounding boxes for sensitive regions
- Constructs region-level representations
- Outputs sensitive regions with confidence scores

3.4 Audio & Video Wrappers

- **Audio**: Converts speech to text using ASR and reuses text detection
- **Video**: Extracts frames and reuses image detection
- These modules act as adapters rather than independent detection engines

---

3.5 Orchestration Layer

- Aggregates detection outputs from all modalities
- Maintains a unified JSON-based data format
- Decides mitigation actions and execution flow

3.6 Mitigation Module

- Applies modality-specific privacy protection:
- Text: masking or replacement
- Image: blurring or inpainting
- Audio: muting or beeping
- Video: frame-wise mitigation

3.7 Explainability & Audit Module

- Generates visual and textual explanations
- Maintains logs describing detected entities and applied mitigations
- Supports transparency and compliance requirements

---

4. Phase 2 Scope Clarification

- **Fully Implemented**: Text and Image detection + mitigation
- **Partially Implemented**: Orchestration and Explainability
- **Design / PoC Only**: Audio and Video

This scoped design ensures feasibility while maintaining extensibility for future phases.