**System Architecture**

1. **Input Layer**
   * Microphone captures audio continuously, focusing only on key phrases or high-decibel sounds.
   * AI-powered voice recognition software analyzes the audio locally to detect emergency triggers.
2. **Processing Layer**
   * **AI Model**:
     + Pre-trained to detect emergency keywords (e.g., "HELP!", "FIRE!", "STOP!") and analyze audio for urgency.
     + Continuous learning from real-world data to improve accuracy.
   * **Verification System**:
     + Cross-checks additional factors like tone, location, device motion, or user input.
     + If a match is confirmed or the user doesn’t respond, escalates to the alert phase.
3. **Output Layer**
   * Sends alerts to:
     + Emergency services (911 or local equivalent).
     + Pre-set emergency contacts.
     + User's device notifications for acknowledgment.
   * Optionally, integrates with public safety networks or IoT systems.
4. **Data Security and Privacy**
   * All processing happens locally where possible; sensitive data is encrypted.
   * Users must opt-in and set preferences for data retention and sharing

**Core Features**

1. **Emergency Detection**
   * Keywords like "HELP!", "FIRE!", or "STOP!"
   * Sudden loud noises or distinct sounds like glass breaking.
   * Context-sensitive triggers (e.g., repetitive phrases, distress tones).
2. **Automatic Actions**
   * Immediate call to emergency services with location and audio snippet.
   * Notification to emergency contacts.
3. **False Alarm Mitigation**
   * User confirmation when possible.
   * AI context analysis reduces accidental triggers (e.g., detecting urgency vs. casual shouting).
4. **Customizability**
   * User-defined keywords.
   * Ability to pause the system temporarily (e.g., during events with loud background noise).