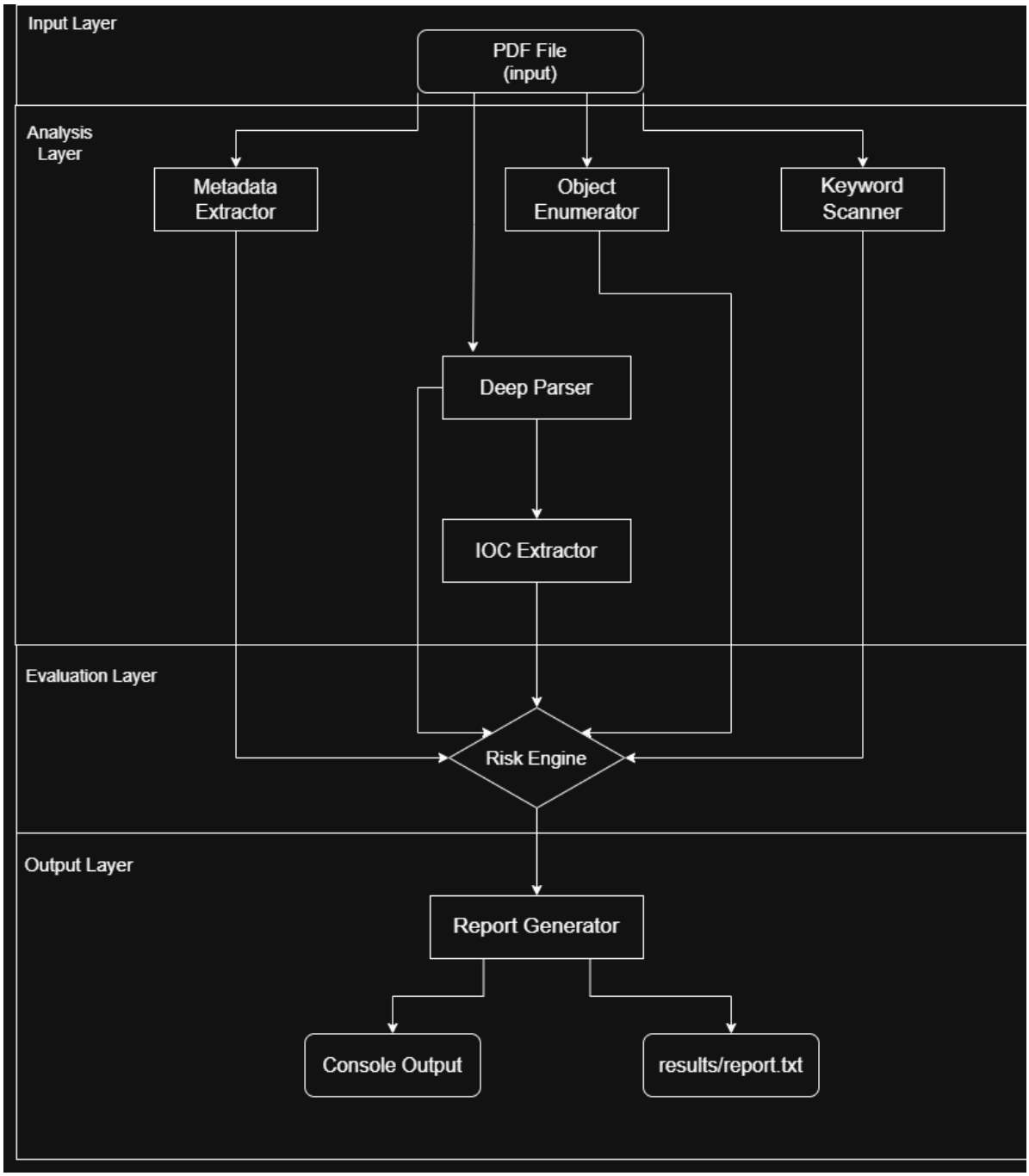
## 1. Workflow Diagram :



## Explaination :

This **workflow diagram** represents the architectural design and data movement inside the Needle toolkit. It is divided into logical **layers**, where each layer performs a specialized function in analyzing a PDF file for malicious behavior.

**Input Layer**

- The process begins with a **PDF file** provided by the user.
- This layer acts as the entry point and forwards the file to multiple analysis components simultaneously.
- No analysis is done here; it only handles file intake and validation.

**Analysis Layer**

This is the core processing layer where different modules inspect the PDF from multiple perspectives:

- **Metadata Extractor**
  Retrieves document properties such as author, creation date, producer software, and modification history. Suspicious or inconsistent metadata can indicate tampering or automated malware generation tools.
- **Object Enumerator**
  Scans and lists all internal PDF objects (streams, fonts, JavaScript objects, embedded files, etc.). This helps in identifying hidden or abnormal structures often used by malicious PDFs.
- **Keyword Scanner**
  Searches for high-risk keywords such as /JavaScript, /OpenAction, /Launch, or encoded strings that commonly appear in exploit-based PDFs.
- **Deep Parser**
  Performs low-level structural parsing of the PDF file, decoding compressed streams and resolving object references. This step uncovers obfuscated or encrypted payloads that simple scanning might miss.
- **IOC (Indicator of Compromise) Extractor**
  Collects potential threat indicators such as URLs, IP addresses, suspicious hashes, embedded executables, or shell commands. These indicators are crucial for threat intelligence correlation.

**Evaluation Layer**

- The **Risk Engine** aggregates outputs from all analysis modules.
- It applies scoring logic or rule-based evaluation to determine the **malicious probability or threat level** of the PDF.
- This layer acts as the decision-making unit of the toolkit.

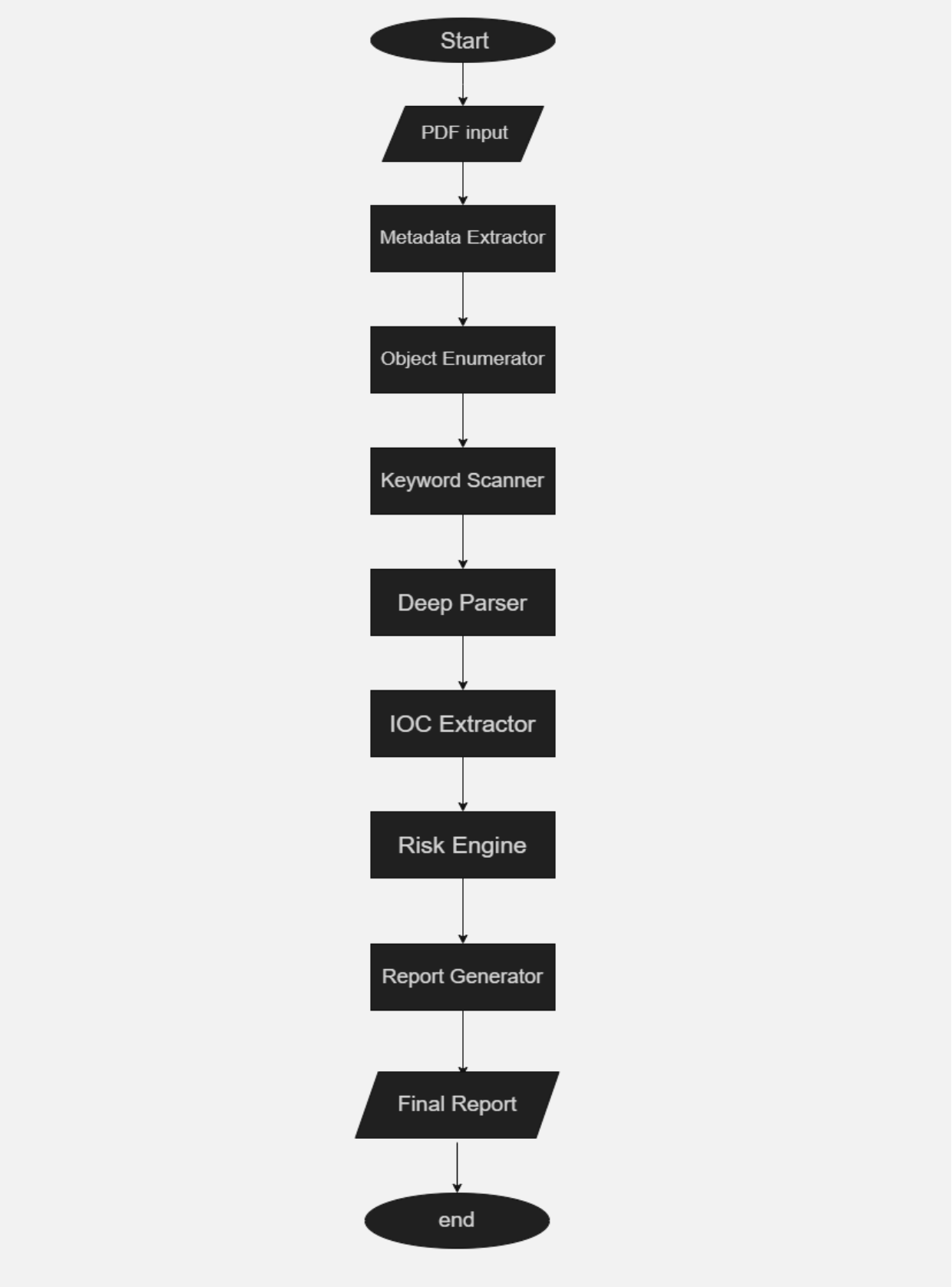
**Output Layer**

- The **Report Generator** converts analysis findings into a readable format.
- Results are delivered through:
    - **Console Output** for quick review.
    - **Text Report File** (e.g., results/report.txt) for documentation and future reference.

**Overall Purpose:**
The workflow diagram highlights how Needle performs **parallel, multi-module inspection** followed by centralized risk evaluation to ensure comprehensive PDF malware detection.

---

## 2. Flowchart – Needle Execution Process :



The **flowchart** illustrates the **sequential execution steps** followed by the Needle toolkit from start to finish. Unlike the workflow diagram, which focuses on architecture, the flowchart focuses on runtime order.

**Step-by-Step Execution**

1. **Start**
   The toolkit execution is initiated by the user or command line trigger.
2. **PDF Input**
   The target PDF file is loaded into the system memory for inspection.
3. **Metadata Extraction**
   Basic document information is analyzed first to quickly detect anomalies or forged attributes.
4. **Object Enumeration**
   All internal objects and streams are listed to understand the file's structural composition.
5. **Keyword Scanning**
   The file content is searched for predefined suspicious or exploit-related keywords.
6. **Deep Parsing**
   Advanced parsing is performed to decode compressed or encoded sections and reveal hidden scripts or payloads.
7. **IOC Extraction**
   Any discovered URLs, IP addresses, embedded files, or suspicious commands are extracted as indicators of compromise.
8. **Risk Engine Evaluation**
   All gathered evidence is assessed collectively to assign a **risk score or classification** (e.g., Safe, Suspicious, Malicious).
9. **Report Generation**
   A structured report is produced summarizing findings, detected indicators, and the final risk assessment.
10. **Final Report / End**
    The report is displayed or saved, and the analysis process terminates.

**Overall Purpose:**
The flowchart demonstrates a **linear and controlled execution pipeline**, ensuring that each analytical stage builds upon the previous one, leading to a systematic and reliable malware assessment of the PDF file.