D-E-ENG

(An Data Extraction Engine)

Sakthi Kumaran

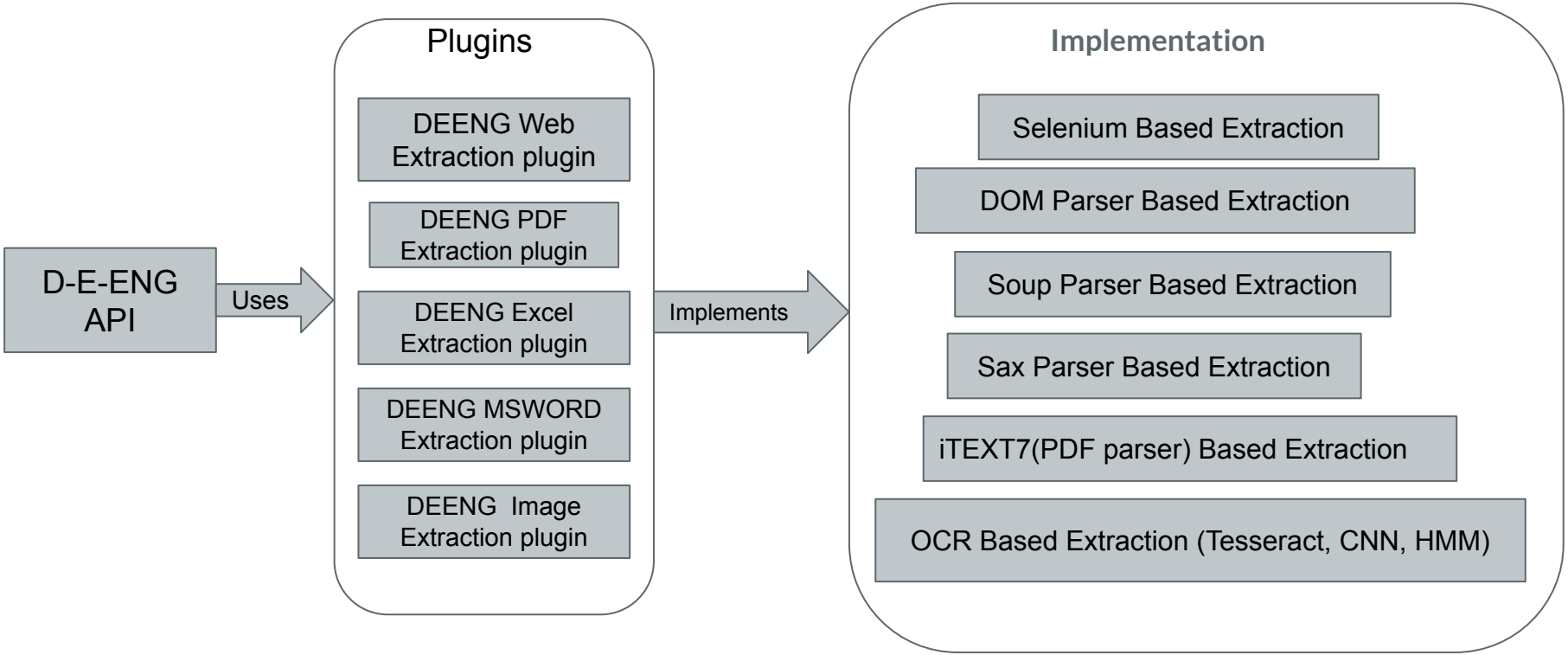
Introduction

DEENG is a store-and-execute, event-driven, multi-layered data-extraction web application that powers with the ability to extract data from various flat data sources (web sites, files etc.) with the provisioning for user to choose the data extraction techniques suitable for each type of extraction and ability to provide exhaustive extraction processing report at field level.

Advantages

- A generic user friendly process for data extraction .
- Complete asynchronous and scalable extraction.
- Supports re-processing of extraction requests.
- Exhaustive extraction execution reporting at field level.
- Maintains history of extraction executions.
- Supports different sources to extract data from:
  - Web Site - (Current focus)
  - PDF
  - Excel
  - Word
- Ability to include new techniques for data extraction as needed.

Extraction Layers



Extraction Layers

DEENG API Layer: Provides APIs to create, view, initiate, and report data extraction.

Plugin Layer: Provides extraction plugins for each type of data source.

Technique/Implementation Layer:

- Pool of various implementation techniques that can be applied for the extraction process.
- A single extraction process can use combination of multiple techniques. For example, web extraction can use selenium for all the click and value set events in combination with soup parser for get-events.

Extraction steps: User Interactions

- User will add the extraction request that contains fields and its location on the data source.
- User initiates the extraction request for execution.
- Each extraction request will have set of events to be performed.
- Events are grouped as per user configuration and each group is executed as separate process. Go Routine of Golang is handy here.
- Extraction technique is chosen based on the approach-key or default approach key.
- Each extraction plugin will have its own extraction algorithm that uses the selected extraction technique to extract the fields.
- The status, value or failure errors on each extraction process is captured on the execution report.

Extraction Execution Reports

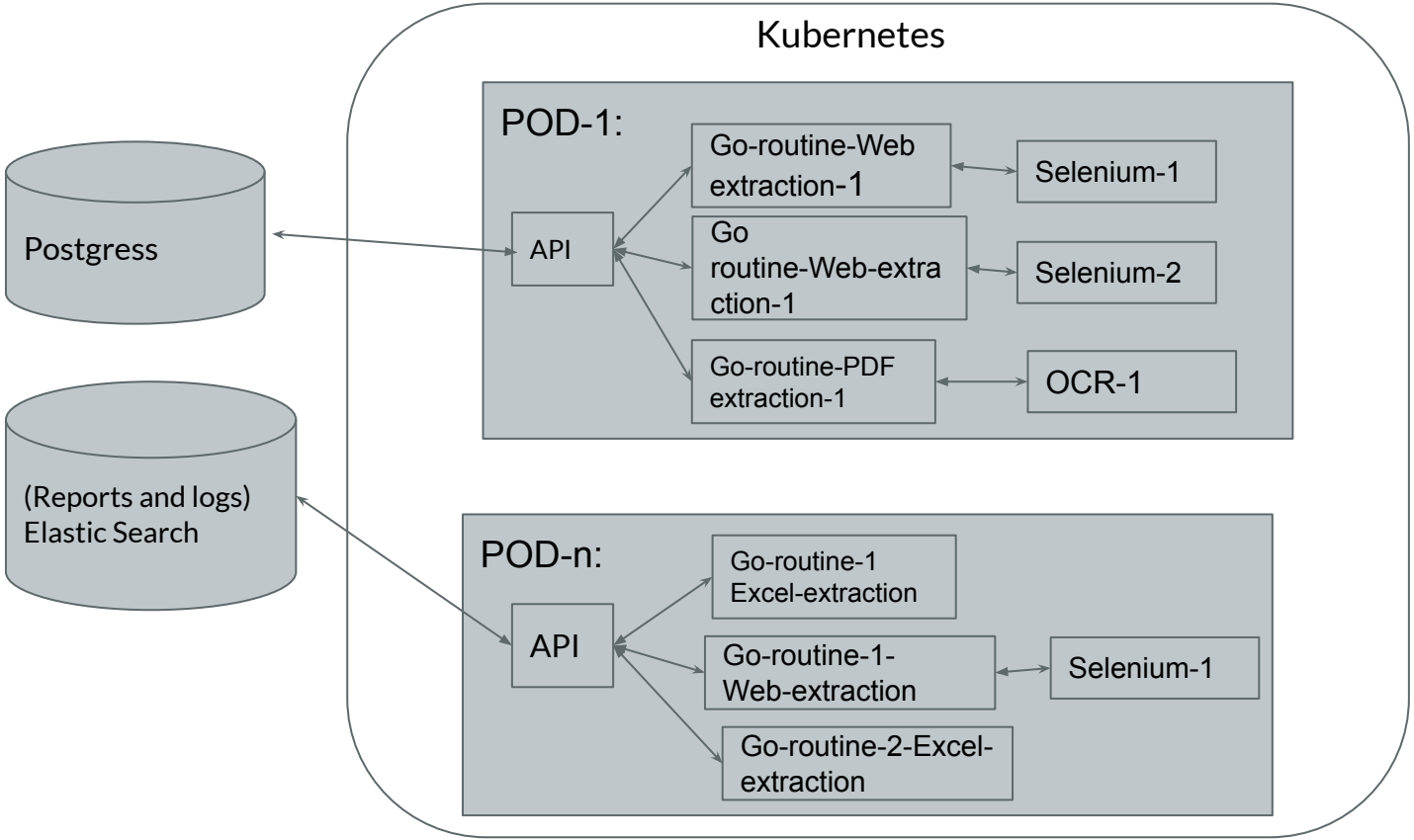
Below are the fields of the Execution Report and their descriptions:

- **Execution-Id :** Uniquely represents the execution of extraction process.
- **Name :** Execution Name
- **URL/DOC :** Initial URL that was used or the s3 document
- **Start-Time :** Start time of the execution.
- **End-Time :** End time of execution.
- **Status :** Completed/Pending
- **Total Number of Fields :** Number of fields for the given group
- **Accuracy :** Confidence level of the extracted value. Mostly applicable for OCR
- **Fields :**
    - **Name :** name of the field used
    - **Location** : XPath or type of value identifier or
    - **EventType** : type of the event corresponding to the field
    - **GroupId** : Identifier of the field group- represents set of events
    - **Sequence Number** : Execution priority of event execution
    - **Status:** Failed/Success
    - **Time Taken** : for the extraction
    - **Error** : error message on failure

Technology used

- Golang
- Docker/K8
- AWS API gateway
- Postgres DB
- Elastic Search

Technical Architecture



THANK YOU!