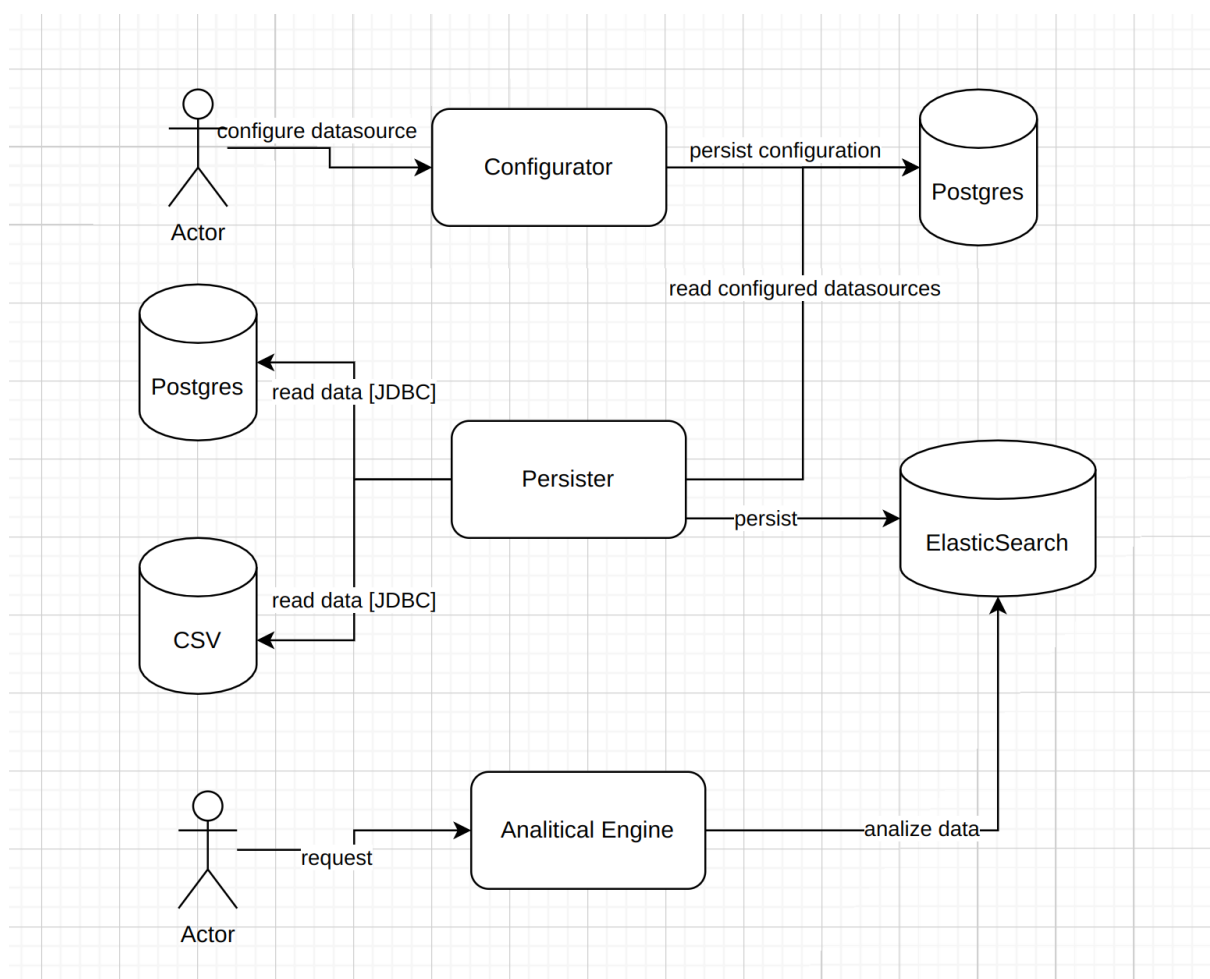


## Идея решения

В качестве решения предлагается использовать Kotlin + Springboot + ElasticSearch. Kotlin и Springboot позволяют быстро создавать прототипы, а ElasticSearch - это документная база данных, что необходимо в нашем случае, так как формат данных не описан. Также у ES есть много встроенных функций по агрегации, интерполяции и работе с данными. А для ML в новых версиях есть соответствующий модуль (<https://www.elastic.co/what-is/elasticsearch-machine-learning>)



## Поставщик данных

В качестве абстракция поставщика данных используется следующий интерфейс

```
data class DataProviderConnect(  
    val jdbcUrl: String,
```

```

    val query: String
)

interface DataProvider : Supplier<Collection<Any>>

interface DataProviderFactory {
    fun get(connect: DataProviderConnect): DataProvider
}

```

Общаться с поставщиком данных будем через jdbc driver. Поэтому была выбрана следующая схема базы для хранения информации про поставщика данных.

DATA_PROVIDER_INFO	
ID	bigint
NAME	varchar(255)
JDBC_URL	varchar(2048)
QUERY_TO_RETRIEVE	varchar(65536)
SYNC_INTERVAL_SECONDS	bigint

Исходный код расположен здесь:

<https://github.com/scrat98/yaprofi-software-engineering/tree/master/semi-final/task3>