An Integrated Data Engineering and Data Science platform for [redacted]

Created by Vasista Polali.

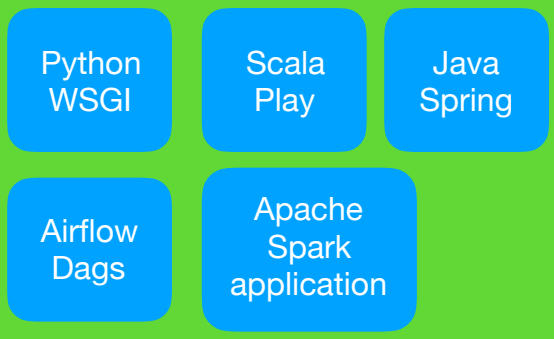
Date: [redacted]

This document outlines a draft architecture to built a scalable data platform with managed services to store, process and expose data for consumption.Each of the coloured boxes depict a different layer in the platform.

The platform design and the development processes thereby established are domain and cloud services provider agnostic and is suitable for adaptation by any organisation looking towards building a Data driven decision making approach.

Data Consumption Layer

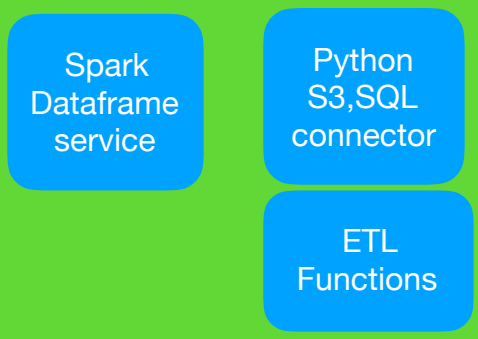
Seed Templates to quickly bootstrap API's and ETL pipelines in a DevOps approach.



- Seed Templates will be built complete with CICD ,Dockerfile and Kubernetes deployment templates.
- Selecting a seed template initiates a process to create a git hub repo with the user as the owner along with the necessary framework to bootstrap the service.
- Developers have to simply fill-in the necessary code and required parameters and

Data Consumption Layer

Libraries for Data engineering and data science tasks in common programming languages.



- Libraries primarily in Python and Scala will be developed for Data Scientists and Engineers to import and use the functions to easily perform ETL on data sets, manipulate and persist them for future use and train their models over them.
- Apart from Libraries, managed services will also be built to let the users easily interact with Apache Spark and AWS services to create , manipulate ,persist and share large datasets as data structures like data frames and tables and use them for ETL and analytical processing.
- A data lake with S3 as the storage engine will be developed and a Catalog service like Hive will be used to define schema and manage the metadata of the data sets being persisted by the above services and functions.

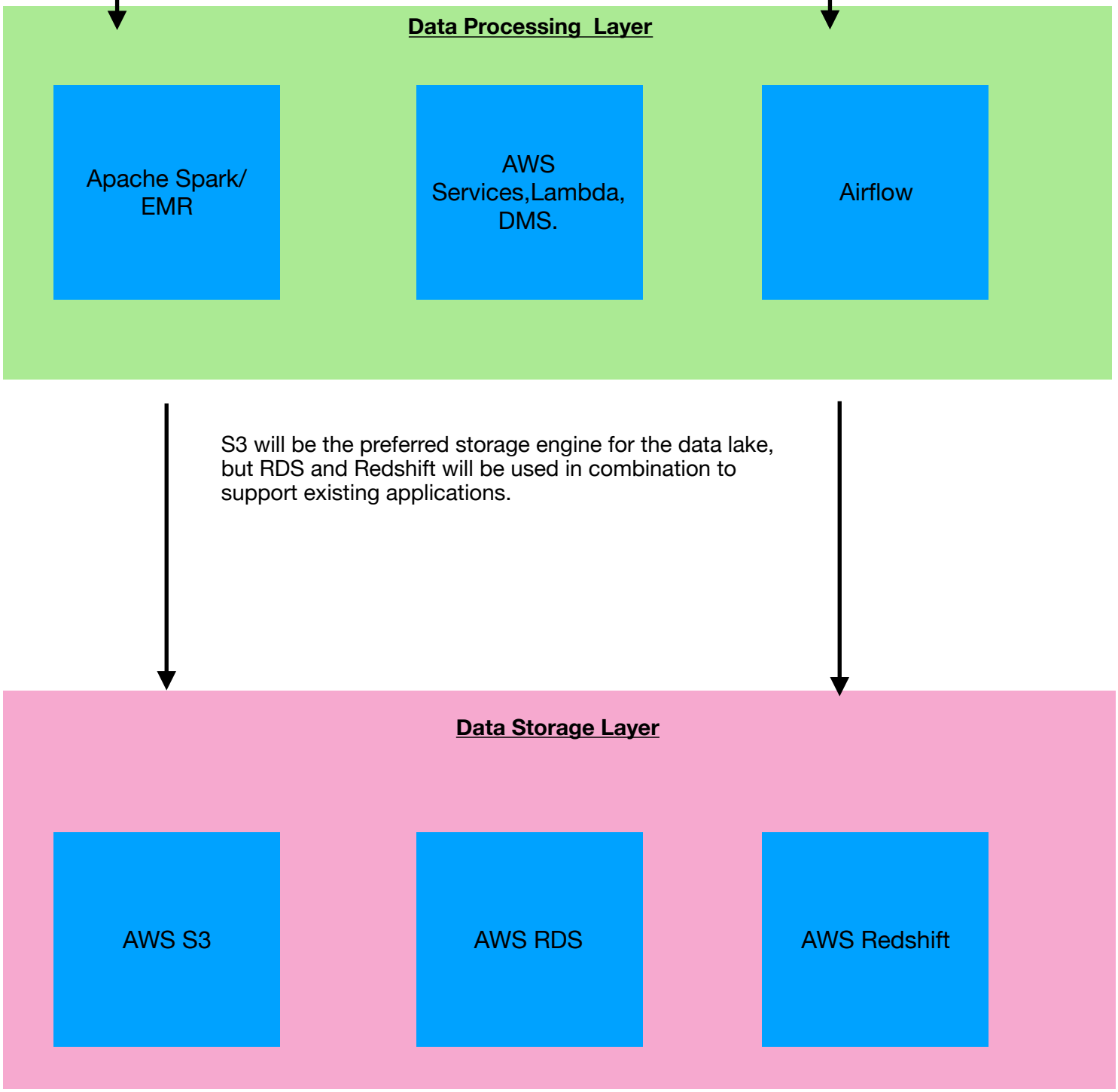
Data Consumption Layer

DOCKER/KUBERNETES

- Services deployed as containerised applications with Docker on Kubernetes.
- Existing [redacted] API standards on security and authentication will be followed and new ones introduced wherever necessary in consultation with the stakeholders

- Functional Libraries are maintained in either an internal artifact repo or as Git repos.
- Users can import them by downloading them into the class path or installing them with the appropriate package manager.
- An example of service or package that will be built is a set of methods/functions to load data in a managed services fashion from various sources like S3, Redshift and RDS into a Spark Data frame.Combine , clean and process them into a table saved on S3, with the schema and other metadata being persisted in Hive for future use by the same user or other members of the organisation.
- User access management will be enabled on the data.
- The saved data will be processed as Schema on Read, allowing the same copy of the dataset to be used with different schemas in a multi-dimensional way by different applications, reducing redundancy and improving churn out speeds and data integrity

The Services are either stand alone micro services or may only contain the methods for routing with the actual processing happening on the components in the processing layer.



S3 will be the preferred storage engine for the data lake, but RDS and Redshift will be used in combination to support existing applications.

Current issues.

- Currently the development approach is very ad-hoc and siloed in nature causing redundancy in data and duplication in effort.
- Lack of proper technical planning or design approach to build new product features is resulting in half baked products and improper effort estimation which have become time centric as in more focussed on meeting deadlines rather than quality centric to build a product in a holistic manner with comprehensive functional coverage.

- Lack of a comprehensive approach in development and deployment is resulting in a lot of failures in the current system and pipelines, resulting in continuous tweaking of existing pipelines and fire fighting, which in turn is eating into the overall development time in the sprints.
- Lack of comprehensive approach is also resulting in a very isolated development approach by individual team members and the design and development of the features/pipelines being a reflection of individual skill levels and preferences ,thereby sometimes affecting the robustness and effectiveness of the end product.

What this platform can do?

- A centralised and scalable platform defines a more comprehensive approach to the development of Data Engineering/Science related features in particular and back end services in general.
- Take cares of the Boiler plate code , which helps developers focus more on developing the actual functionality rather than getting bogged down with bootstrapping the API's and pipelines increasing velocity.
- The libraries and services built as part of the platform will help automate a lot a recurring tasks improving reusability of functionality and code.
- Creates a Data lake which will be a central store for all the data sets and a single source of truth, thereby improving data consistency, data availability ,data integrity and data security.
- Sets certain standards and expectations and brings uniformity in approach to design and development of features across the organisation, making it more robust and easy to plan and manage.
- Helps data scientists focus more on training the machine learning models and flattens their learning curve if any ,on tasks related to processing and persisting training data sets on different environments.
- Being inherently scalable and flexible it can easily be adapted to evolving use cases and traffic.
- Creates a set of standards and processes that will be uniform across the board resulting in higher quality products with reduced development effort.