Slice 1

Data Lake Architecture Overview

Le Nhu Chu Hiep

1. Big Data
2. Data warehouse
3. Data Lake
4. Methodology

Section 1

Big Data

What is big data

Data is too large for traditional storage to handle and retrieve.

Section 2

Data warehouse

What ?

- Centralize relation information for specific purpose.
- A RDBMS that has a schema to support BI tools and OLAP.

Section 3

Data Lake

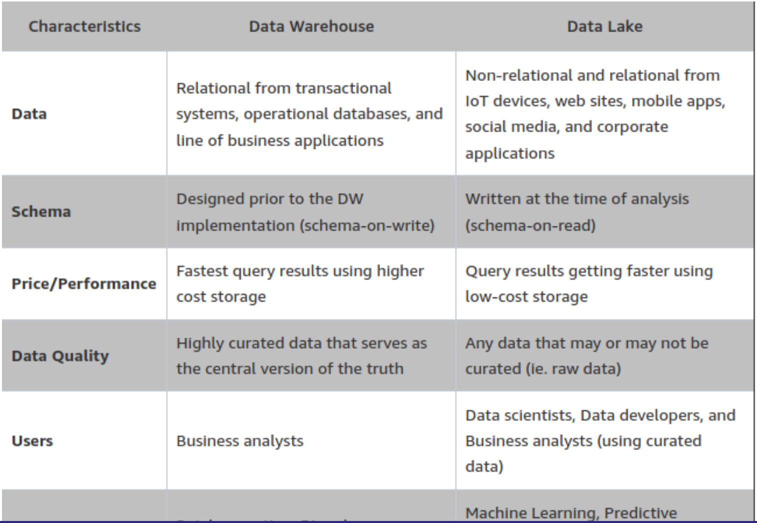
What ?

- Centralized repository that allows you to store all your structured and unstructured data at any scale.
- Update version of data warehouse to handle large data with dirverse type.
- Big Data version of data warehouse

Why ?

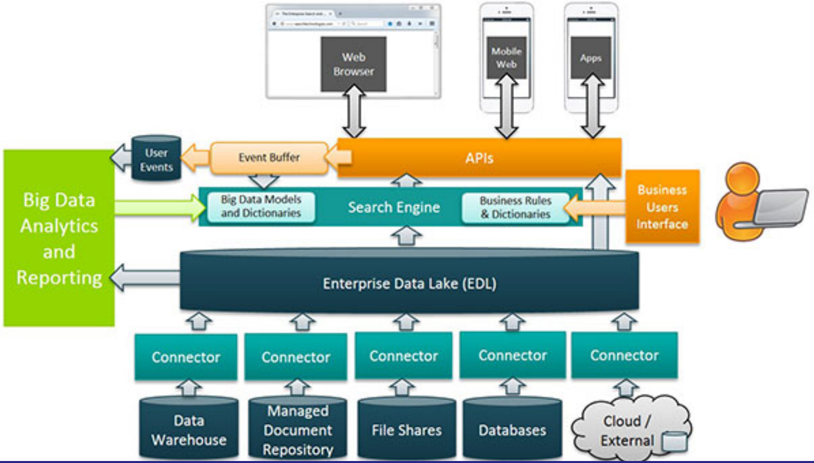
- Data warehouse and its analytic tool can not handle big data.
- Requirement of storing and trieving unstructured data.

Data Lake and Data Warehouse



Characteristics	Data Warehouse	Data Lake
Data	Relational from transactional systems, operational databases, and line of business applications	Non-relational and relational from IoT devices, web sites, mobile apps, social media, and corporate applications
Schema	Designed prior to the DW implementation (schema-on-write)	Written at the time of analysis (schema-on-read)
Price/Performance	Fastest query results using higher cost storage	Query results getting faster using low-cost storage
Data Quality	Highly curated data that serves as the central version of the truth	Any data that may or may not be curated (ie. raw data)
Users	Business analysts	Data scientists, Data developers, and Business analysts (using curated data)
		Machine Learning, Predictive

Data Lake prototype



Data Lake Storage

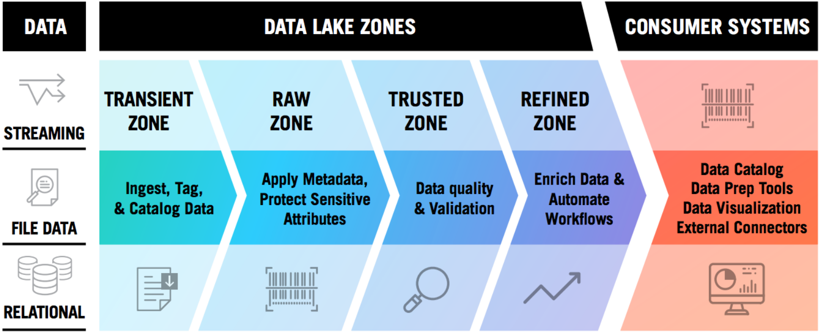


Figure 3: Data Lake Stg

Stage of the Art

- Teradata
- Think Big

Stage of the Art

Stage of the Art

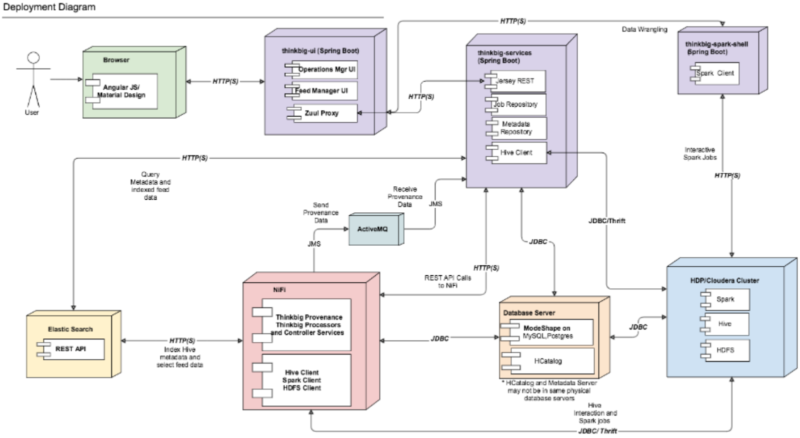

Deployment Diagram
User
Browser
Angular JS/ Material Design
HTTP(S)
thinkbig-ui (Spring Boot)
Operations Mgr UI
Feed Manager UI
Zuul Proxy
HTTP(S)
HTTP(S)
Data Wrangling
thinkbig-services (Spring Boot)
Jersey REST
Job Repository
Metadata Repository
Hive Client
thinkbig-spark-shell (Spring Boot)
Spark Client
Interactive Spark Jobs
HTTP(S)
HTTP(S)
Query Metadata and indexed feed data
Receive Provenance Data
JMS
Send Provenance Data
ActiveMQ
JMS
JDBC/Thrift
JDBC
HTTP(S)
REST API Calls to NiFi
NiFi
Thinkbig Provenance Thinkbig Processors and Controller Services
Hive Client Spark Client HDFS Client
Elastic Search
REST API
HTTP(S)
Index Hive metadata and select feed data
JDBC
Database Server
ModeShape on MySQL,Postgres
HCatalog
* HCatalog and Metadata Server may not be in same physical database servers
JDBC
HDP/Cloudera Cluster
Spark
Hive
HDFS
Hive Interaction and Spark jobs
JDBC/ Thrift

Figure 5: kylo_arch

Section 4

Methodology

General idea

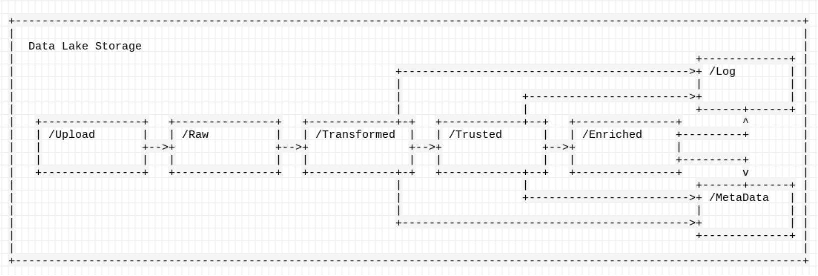
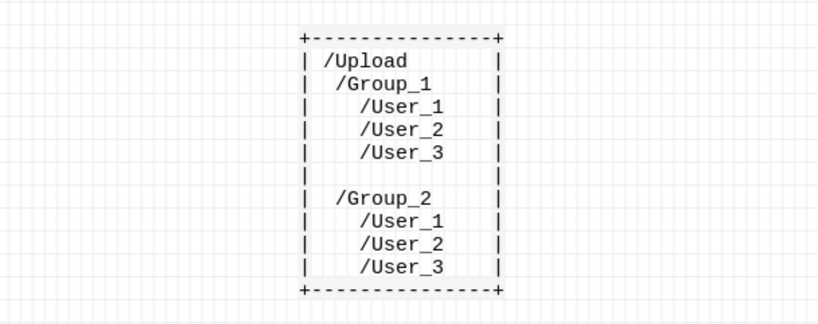


Figure 6: DL_arch

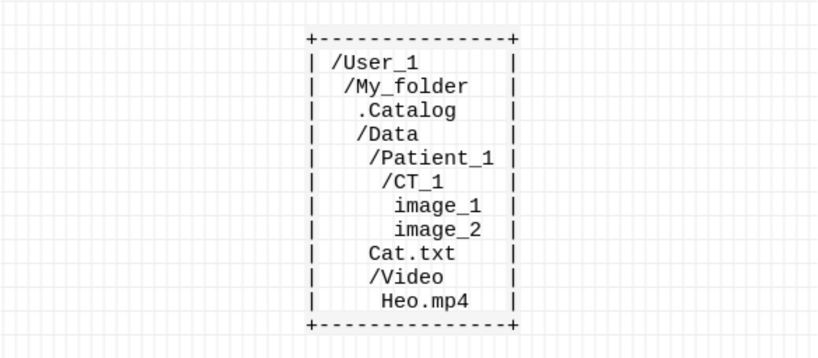
Upload



```
+---------------+
| /Upload       |
|  /Group_1     |
|    /User_1    |
|    /User_2    |
|    /User_3    |
|               |
|  /Group_2     |
|    /User_1    |
|    /User_2    |
|    /User_3    |
+---------------+
```

Figure 7: Upload

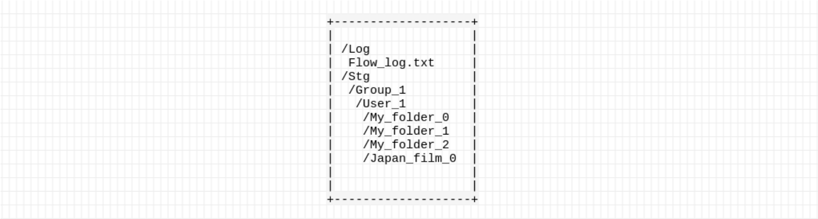
Upload



```
+---------------+
| /User_1       |
|  /My_folder   |
|   .Catalog    |
|   /Data       |
|    /Patient_1 |
|     /CT_1     |
|      image_1  |
|      image_2  |
|    Cat.txt    |
|    /Video     |
|     Heo.mp4   |
+---------------+
```

Figure 8: Upload

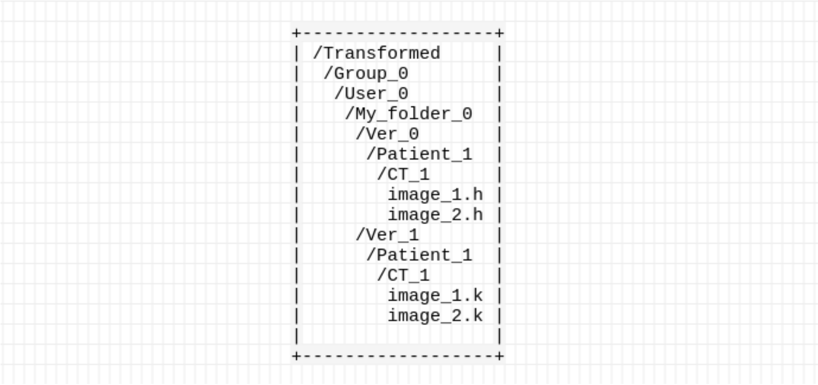
Raw



```
+-------------------+
|                   |
| /Log              |
|  Flow_log.txt     |
| /Stg              |
|  /Group_1         |
|   /User_1         |
|    /My_folder_0   |
|    /My_folder_1   |
|    /My_folder_2   |
|    /Japan_film_0  |
|                   |
|                   |
+-------------------+
```

Figure 9: Raw

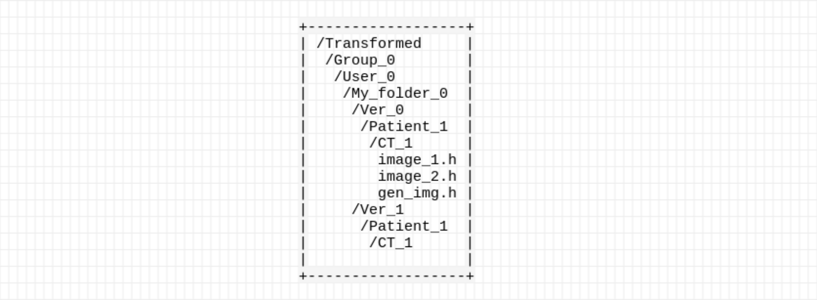
Transformed



```
+-------------------+
| /Transformed      |
|  /Group_0         |
|   /User_0         |
|    /My_folder_0   |
|     /Ver_0        |
|      /Patient_1   |
|       /CT_1       |
|        image_1.h  |
|        image_2.h  |
|     /Ver_1        |
|      /Patient_1   |
|       /CT_1       |
|        image_1.k  |
|        image_2.k  |
|                   |
+-------------------+
```

Figure 10: Transformed

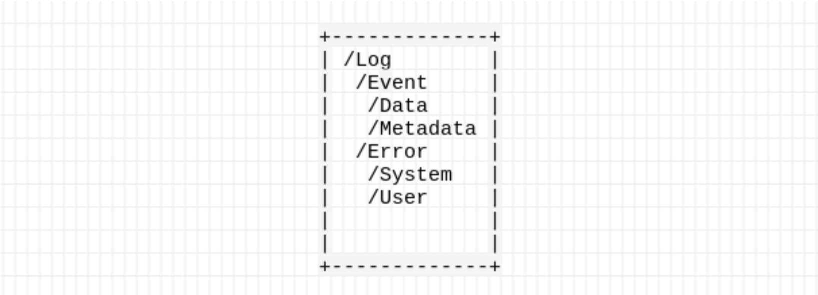
Trusted



```
+------------------+
| /Transformed     |
|  /Group_0        |
|   /User_0        |
|    /My_folder_0  |
|     /Ver_0       |
|      /Patient_1  |
|       /CT_1      |
|        image_1.h |
|        image_2.h |
|        gen_img.h |
|     /Ver_1       |
|      /Patient_1  |
|       /CT_1      |
|                  |
+------------------+
```

Figure 11: Trusted

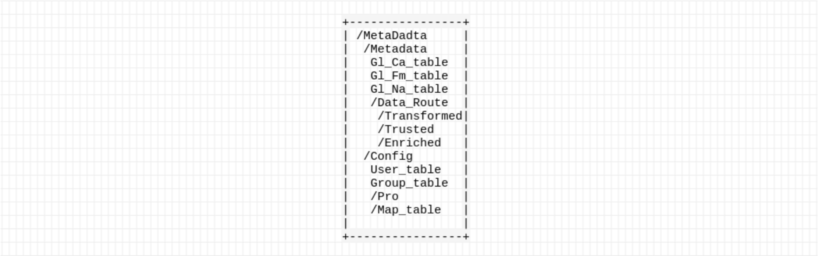
Log_struct



```
+--------------+
| /Log         |
|  /Event      |
|   /Data      |
|   /Metadata  |
|  /Error      |
|   /System    |
|   /User      |
|              |
|              |
+--------------+
```

Figure 12: Log

Metadata



```
+----------------+
| /MetaDadta     |
|  /Metadata     |
|   Gl_Ca_table  |
|   Gl_Fm_table  |
|   Gl_Na_table  |
|   /Data_Route  |
|    /Transformed|
|    /Trusted    |
|    /Enriched   |
|  /Config       |
|   User_table   |
|   Group_table  |
|   /Pro         |
|   /Map_table   |
|                |
+----------------+
```

Figure 13: Meta Data

Q/A ?