Approvers

GSM Role	Name
Project Manager	Evelyne Lemauvais
Technical Reviewer	Sunil Kolli
Technical Reviewer	Ritwick Kumar

NOTE: Approvers listed apply only to this document revision.

Revision History

Revision	Description of Change
A	Initial Release (of Consolidated Design Document)

Table of Contents

1. Introduction

Purpose

In accordance with Enterprise objective EDH would be main hub for storing and shaping up GCMS data which would then be used by downstream systems for their analytics and reporting needs. As per the requirement , data would be ingested from GCMS source on incremental/on-demand basis and stored in EDH. This document presents details of the implementation approach which is subjected to review by appropriate stakeholders.

Overview

The GCMS project in EDH would deliver a Hadoop-based analytics platform providing initial capabilities in use cases like data storage, data discovery and data transformation for downstream systems.

The initial platform will consist of the following two major components:

- Full Data Ingestion – GCMS (Multiple sources) to EDH

The scope of this application design document is to define the overall design and architecture of this process, GCMS to EDH. The document will only cover the solution with in the Hadoop platform along with end to end Architecture, Design and Implementation whereas the data recovery & clustered environment backup and BI data backup/recovery is out of scope for this design document.

References

Document Number	Document Name	Location
xxxxx	xxxxxxx	Global Workflow

Terminology

Term	Description
BSC	Boston Scientific
BDR	Backup and Disaster Recovery
CDH	Cloudera Distribution of Hadoop
EDH	Enterprise Data Hub
HDFS	Hadoop File Distributed System
SVN	Apache Subversion
GCMS	Global Complaint Management System

2. Requirements

In accordance with Enterprise objective EDH would be main hub for storing and shaping up GCMS data which would then be used by downstream systems for analytics and reporting. As per requirement GCMS data would be ingested from RDBMS source on incremental/on-demand basis. This document presents details of the implementation approach which is subjected to review by appropriate stakeholders.

2.1. Data Ingestion

Data Ingestion requirements are listed here.

2.2. Transformation Requirements

N/A

3. Assumption And Prerequisites

This document is intended for detailing out the implementation requirements. For more detailed requirements please refer the project location. Source system GCMS is an Oracle's Database is EST time. EDH is following UTC time. The core of the data load expects time synchronization between the source and the EDH system.

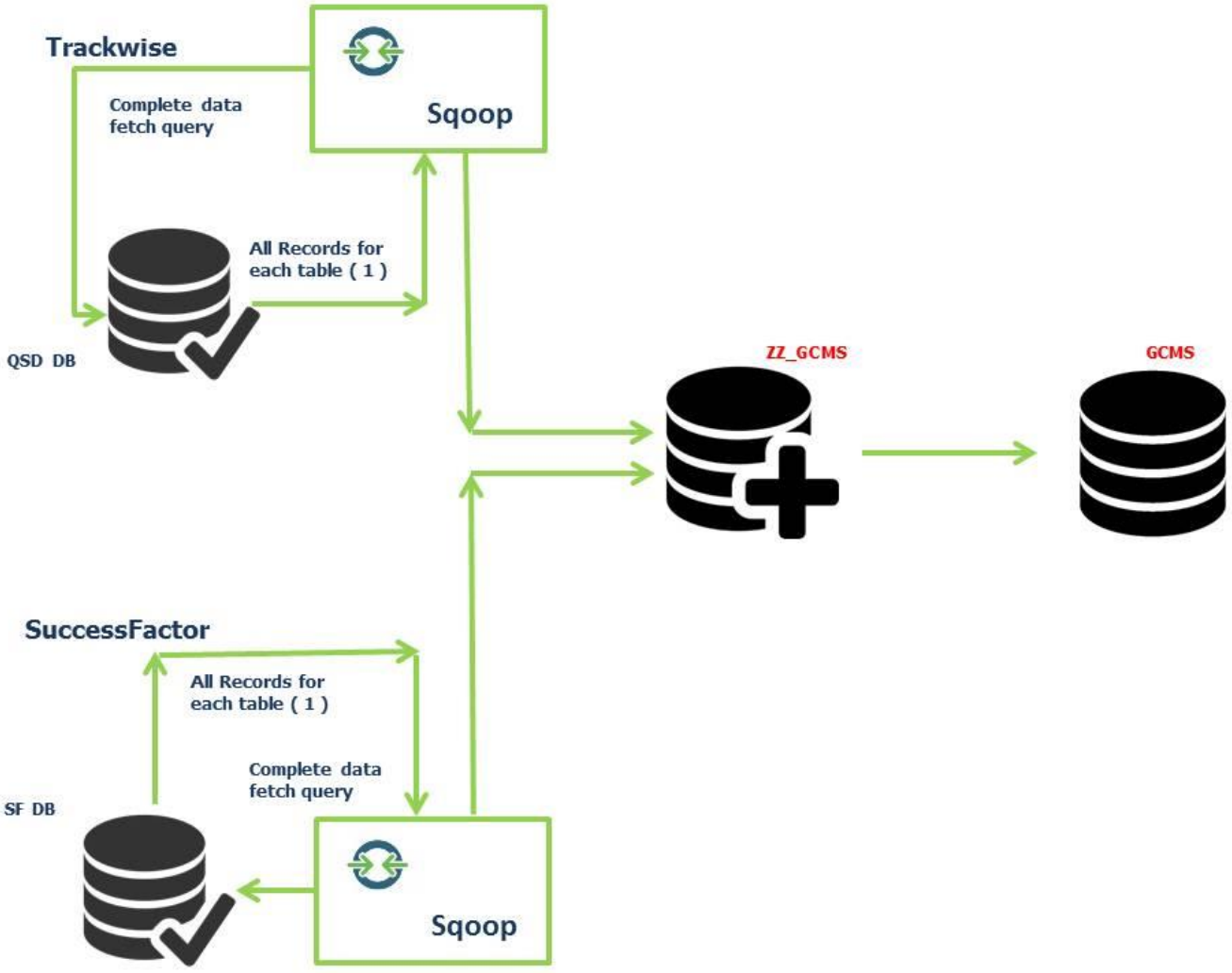
4. Hive Level Design

4.1. Application Architecture

This section would highlight the structural design process and their implementation activities which includes several stages to accomplish the end to end Implementation such as

→GCMS to EDH

Application Architecture Diagram:



These are the three major components.

→ZZ_GCMS: Contains as-in data load from source system in a single hive data store named ZZ_GCMS.

→GCMS : Contains views of the as-in load as per the requirement document in a single hive data store named GCMS.

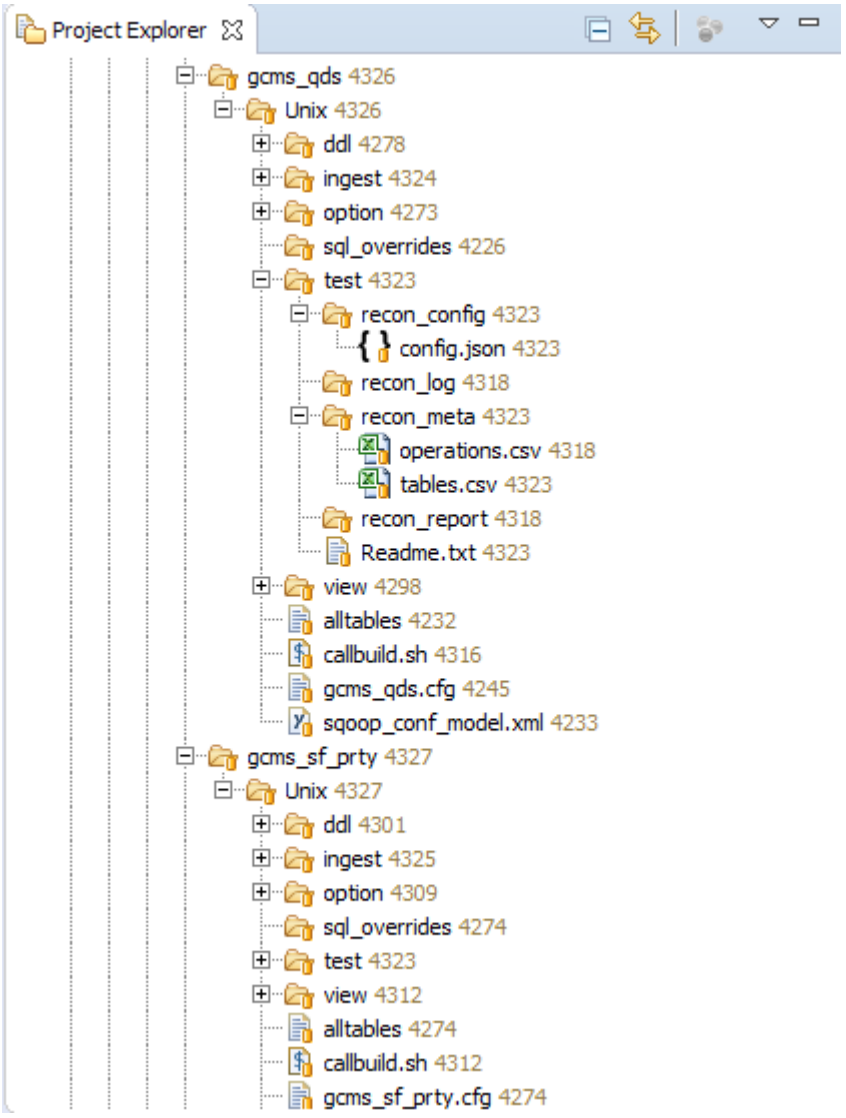
5. Detailed Technical Design

This section highlights the detailed level structural design process and their implementation activities which includes several stages to accomplish the end to end Implementation.

- Code Structure
- Environment Setup
- Execution Process

5.1. Code Structure

Since GCMS ingestion has two interface namely QDS (Trackwise) and SuccessFactor hence the code is maintained in separate repos in SVN at
svn://mapls026.bsci.bossci.com/export/svn/svnroot/edh/tenants/gcms1/nonstreaming/gcms_qds (For QDS Trackwise)
svn://mapls026.bsci.bossci.com/export/svn/svnroot/edh/tenants/gcms1/nonstreaming/gcms_sf_prty(For SuccessFactor)



Root Folder: Unix
Folders under Unix:

- **ddl**: contains folder parq_ddl which has create table ddl for all tables which are to be ingested. This was done optionally as a reference.
- **ingest** : contains files
    - GCMS_createDB.sql: Create GCMS db hive commands for creation of db if not exists already. This db is be used to store the views
    - ZZ_GCMS_createDB.sql: Create ZZ_GCMS db hive commands for creation of db if not exists already. This db is be used to store the ingested base tables.

    - SetDBPrivs.sql: Sql file which contain the command for setting security to the above mentioned dbs.
- **option** : contains individual option files per table be used as a parameter for Sqoop , which is being called from callbuild.sh
- **sql_overrides** : optional (in case any sql overrides are to be run , none in this case)
- **test**: contains the reconciliation folder namely log, config, meta, report which are path where the automated reconciliation files are being hosted.
- **view**: contains view files to be created.
- **alltables** : contains list of tables to be ingested
- **callbuild.sh**: actual code which ingests data from source db to target db using Sqoop with option files as mentioned above. All the logging is being handled through this script.
- **gcms_qds.cfg/ gcms_sf_prty.cfg**: Source db configuration file containing the credentials
- **sqoop_conf_model.xml**: Sqoop configuration file.

5.2. Environment Setup

Following steps are performed as part of initial setup to prepare the EDH environment for GCMS load process:

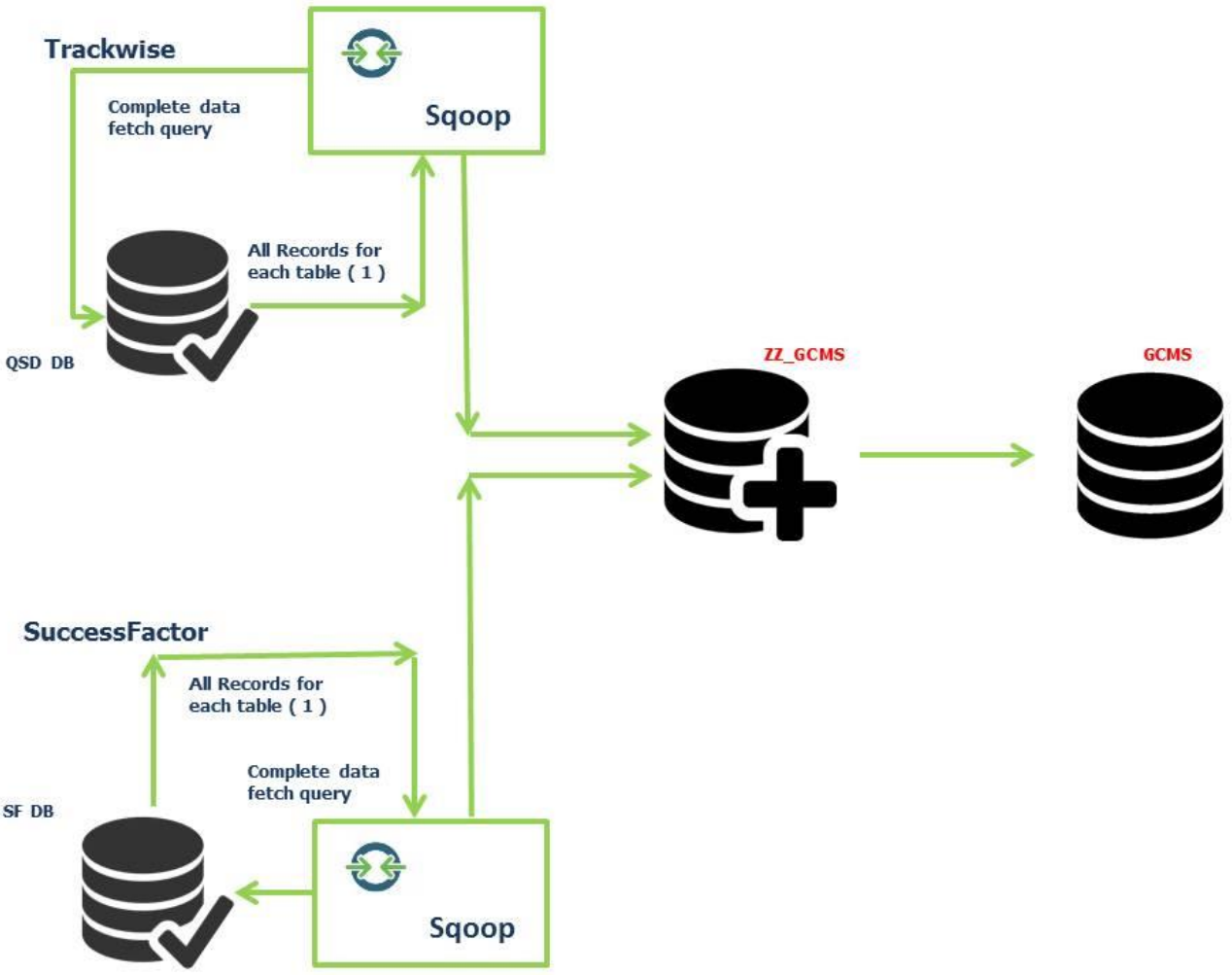
1. Creating landing/staging database in hive (ZZ_GCMS) in EDH if not exists. This database is used for as-is load from source system.
2. Creating views database in hive (GCMS) in EDH. This database is used for storing as-is data views as per requirements.
3. Create Sqoop jobs for all tables.
4. Grant access to the appropriate group members (edh_bsc_internal_users).

This how the current state of GCMS specific databases (ZZ_GCMS, GCMS) with respect to tables and views looks like:

alltables

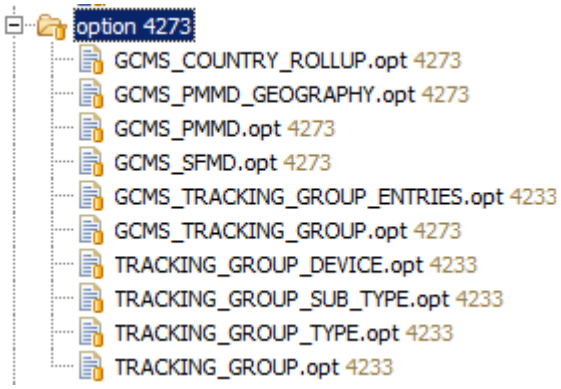
```
1 GCMS_TRACKING_GROUP_ENTRIES
2 GCMS_TRACKING_GROUP
3 GCMS_COUNTRY_ROLLUP
4 TRACKING_GROUP
5 TRACKING_GROUP_SUB_TYPE
6 TRACKING_GROUP_DEVICE
7 GCMS_PMMD_GEOGRAPHY
8 GCMS_SFMD
9 GCMS_PMMD
10 TRACKING_GROUP_TYPE
11
```

5.3. Execution Process



A. Sqoop Jobs:

The first step after the environment setup is to create the Sqoop option file for all the tables. A Sqoop job is a representation of a SQL query in key-value map format from which the query is generated and fired at runtime.



And one Sqoop job looks like :

GCMS_COUNTRY_ROLLUP.opt

```
--as-parquetfile --fetch-size 10000 --compress  --compression-codec snappy -m 1 \
--mapreduce-job-name GCMS_COUNTRY_ROLLUP \
--query 'SELECT
COUNTRY_CODE
, HIERARCHY_TYPE
, LEVEL1_CODE
, LEVEL1_DESCRIPTION
, LEVEL2_CODE
, LEVEL2_DESCRIPTION
, LEVEL3_CODE
, LEVEL3_DESCRIPTION
, cast(cast(LAST_UPDATE AS TIMESTAMP WITH LOCAL TIME ZONE) AS TIMESTAMP) as LAST_UPDATE
 FROM EIAFDAUSER.GCMS_COUNTRY_ROLLUP
 WHERE $CONDITIONS'
```

6. Application Security

1. GCMS data for downstream systems would be available via views in GCMS database and user's part of edh_GCMS group would have select access to this database.

7. Key Roles

Role	Name
Digital Health Project Manager	Evelyne Lemauvais
EDH Technical Architect	Sunil Kolli, Ritwick Kumar
EDH Developer	Anurag Kumar

8. Sign-Offs

The completed application design document will be circulated to the relevant teams for feedback. The review process will be followed by iterations of review and circulation of document. The review comments will be addressed and incorporated in the document before final circulation for sign-off. The sign-off will indicate that it is complete and fit for purpose. Following sign-off it will be baselined and become subject to formal change control procedures

Role and Area	Name	Acceptance Signature / Date
EDH Group		
EDH Group		