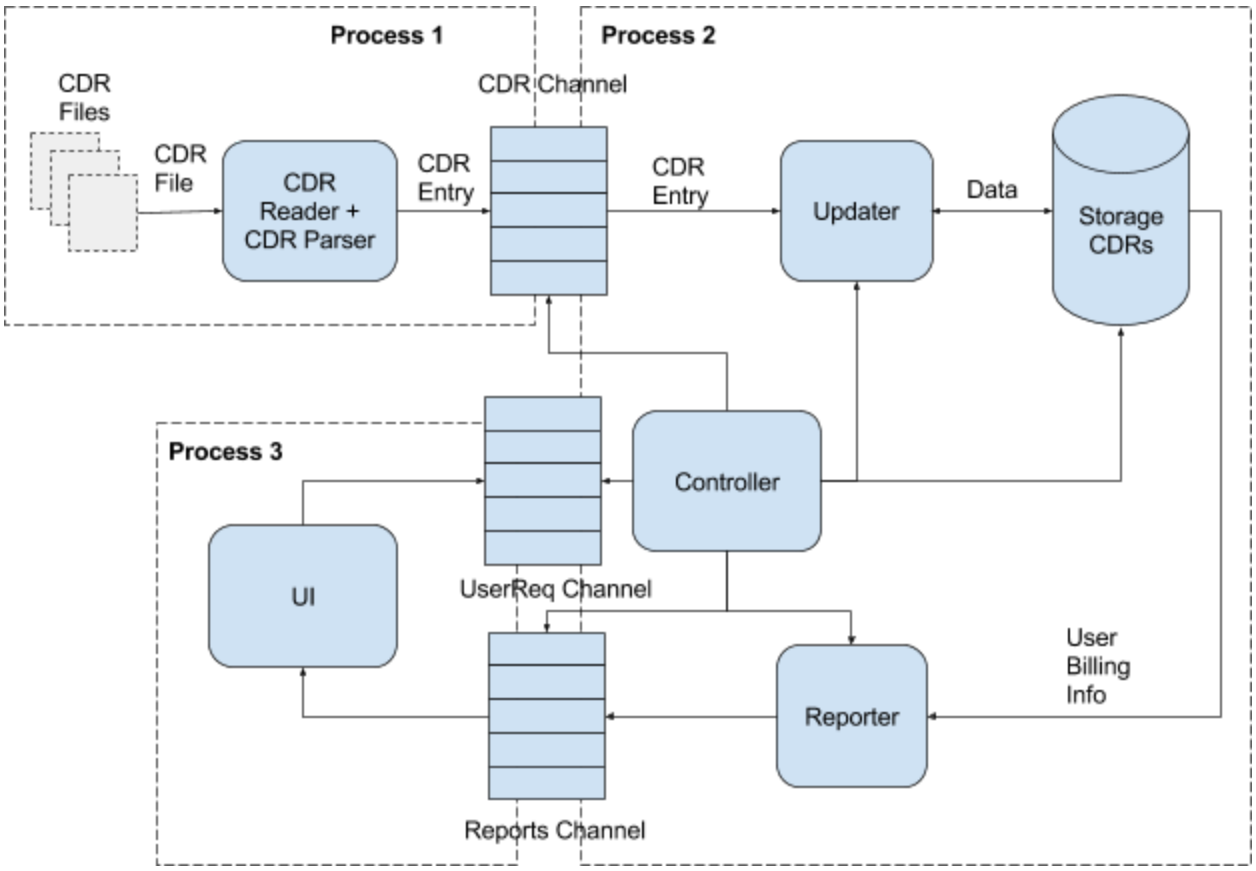
Actual High Level Design:

CDR Processing Flow



Process 1:

CDR Reader:

- Open channel in process 1:
    - Reader → Updater (CDR Channel)
- Open files in specified location (config file)
- Send Start message to CDR channel
- Read one line into buffer and move it to parser [MAX_BUFFER_SIZE 512]
- When EOF: send end message to CDR channel
- Move finished file to Done folder
- Multithreading - one thread per file
- Start msg: includes num of threads running ⇒ num of End msgs to receive from the channel on updater side

CDR Parser:

- Parse one line and turn it into CDR Record
- Return the result to reader

Process 2:

Controller:

- Open channels in process 2:
  - Reader → Updater (CDR Channel)
  - UI → Controller (UserReq Channel)
  - Reporter → UI (Reports Channel)
- Start Storage: Aggregator
- Start Updater threads
- Start Reporter threads
- Wait for UI request
- Signal reporter to generate reports
- Shutdown every open entity upon request

Updater:

- Multithreaded
- Extract CDR record from the channel
- Store or update and store the record in aggregator
- Automatically wait for records in CDR channel

CDR Storage: (v.1) - future version features two aggregators and data stored by type

- Store unaltered CDR records
- Each extracted record is removed from the storage completely
- After update the record has to be re-inserted into aggregator
- Aggregator is created and destroyed by Controller

Reporter:

- Extract the record from the Aggregator
- Accumulate the relevant info by report type for global reports
- Place the record in the Report channel

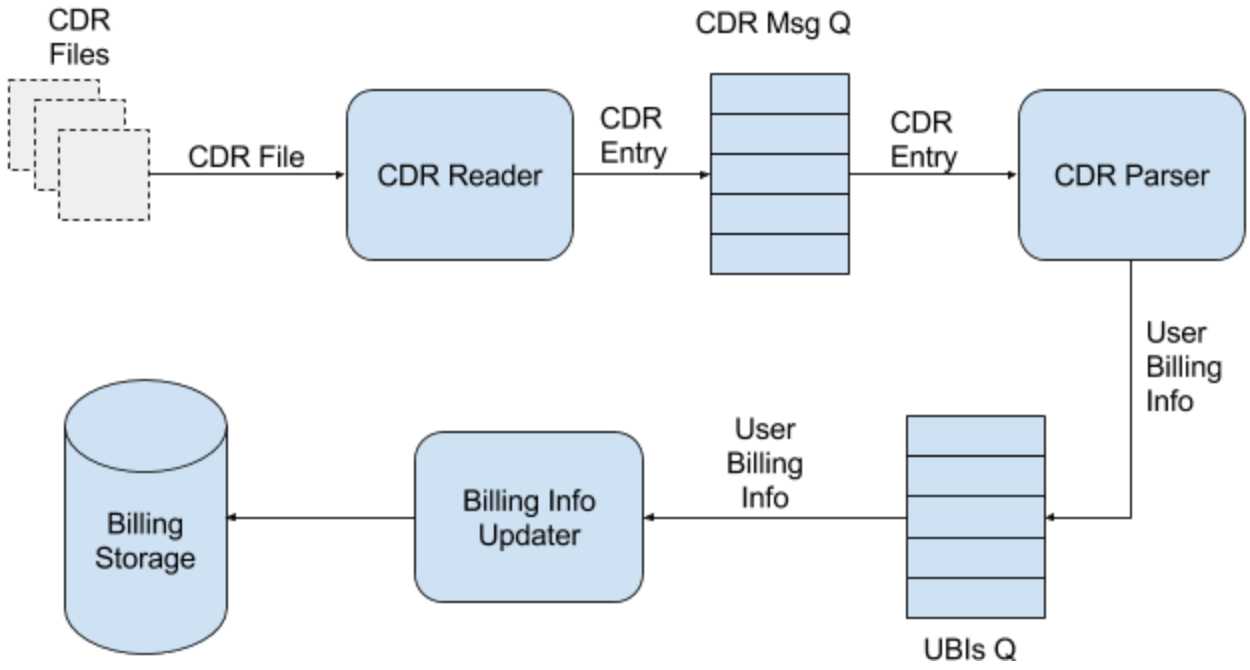
Process 3:

User Interface:

- Open UserReq channel UI side
- Open Reports channel UI side
- Print user options
- Get user options
- Send user request into UserReq channel
- Receive Report via Report Channel

Early Stages High Level Design:

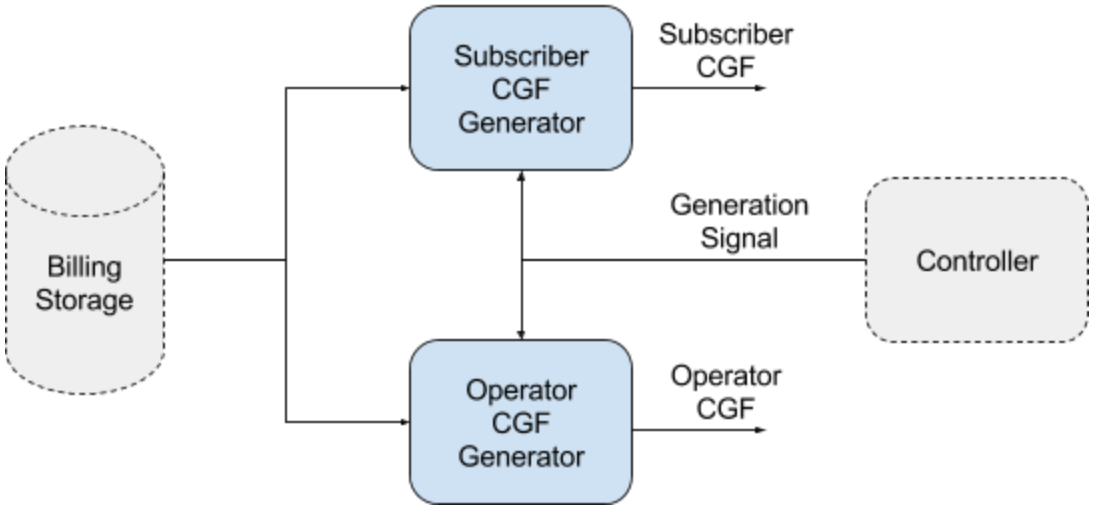
A. CDR Processor



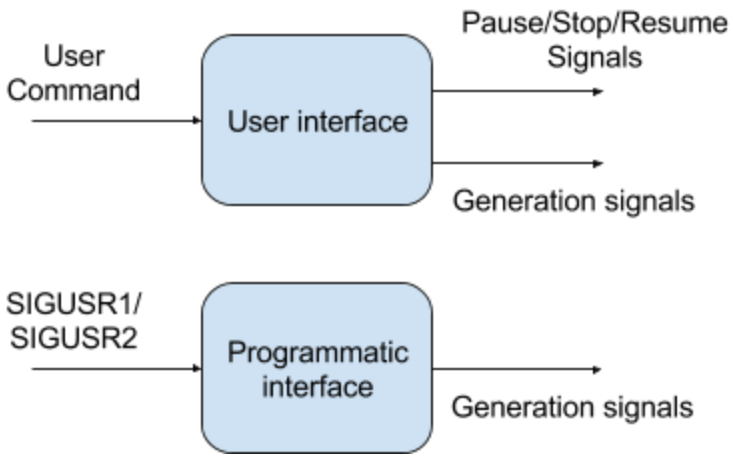
Billing Info Updater:

- Create Updater
- Destroy Updater
- Create Protected Input Channel /* not part of API ?? */
- Create multiple threads /* ?????*/
- Get UBI instance from Q
- Put in storage

B. CGF Generation



C. Controller



Entities:

CDR Entry:

Field	Type	Max len.
CDR	char*	255 (512)

Info:

- IMSI - 15 chars (max)
- MSISDN - 15 chars (max)
- Subscriber IMEI - 15 chars (exactly)
- Operator Brand Name - 64 chars (max)
- Operator MCC + MNC tuple - 5 or 6 chars
- Call Type - enum { MOC, MTC, SMS-MO, SMS-MT, GPRS }
- Call Date - DD/MM/YYYY
- Call Time - HH:MM:SS
- Duration - n seconds, int
- Download - MB, int
- Upload - MB, int
- Party MSISDN - 15 chars (max)
- Party Operator MCC + MNC tuple - 5 or 6 chars

OperatorBillingInfo (OBI):

Field	Type	Max len.
m_MCCMNC	char*	6
m_brandName	char*	64
m_MTCallDuration	int	
m_MOCallDuration	int	
m_MTSMSCount	int	
m_MOSMSCount	int	

UserBillingInfo (UBI):

Field	Type	Max len.
m_IMSI	char*	15
m_insMO	int	
m_insMT	int	
m_outMO	int	
m_outMT	int	
m_insSMS-MO	int	
m_insSMS-MT	int	
m_outSMS-MO	int	
m_outSMS-MT	int	
m_upMB	int	
m_downMB	Int	

int m_internMO;
int m_internMT;
int m_outsideMO;
int m_outsideMT;
int m_internSMS_MO;
int m_internSMS_MT;
int m_outsideSMS_MO;
int m_outsideSMS_MT;