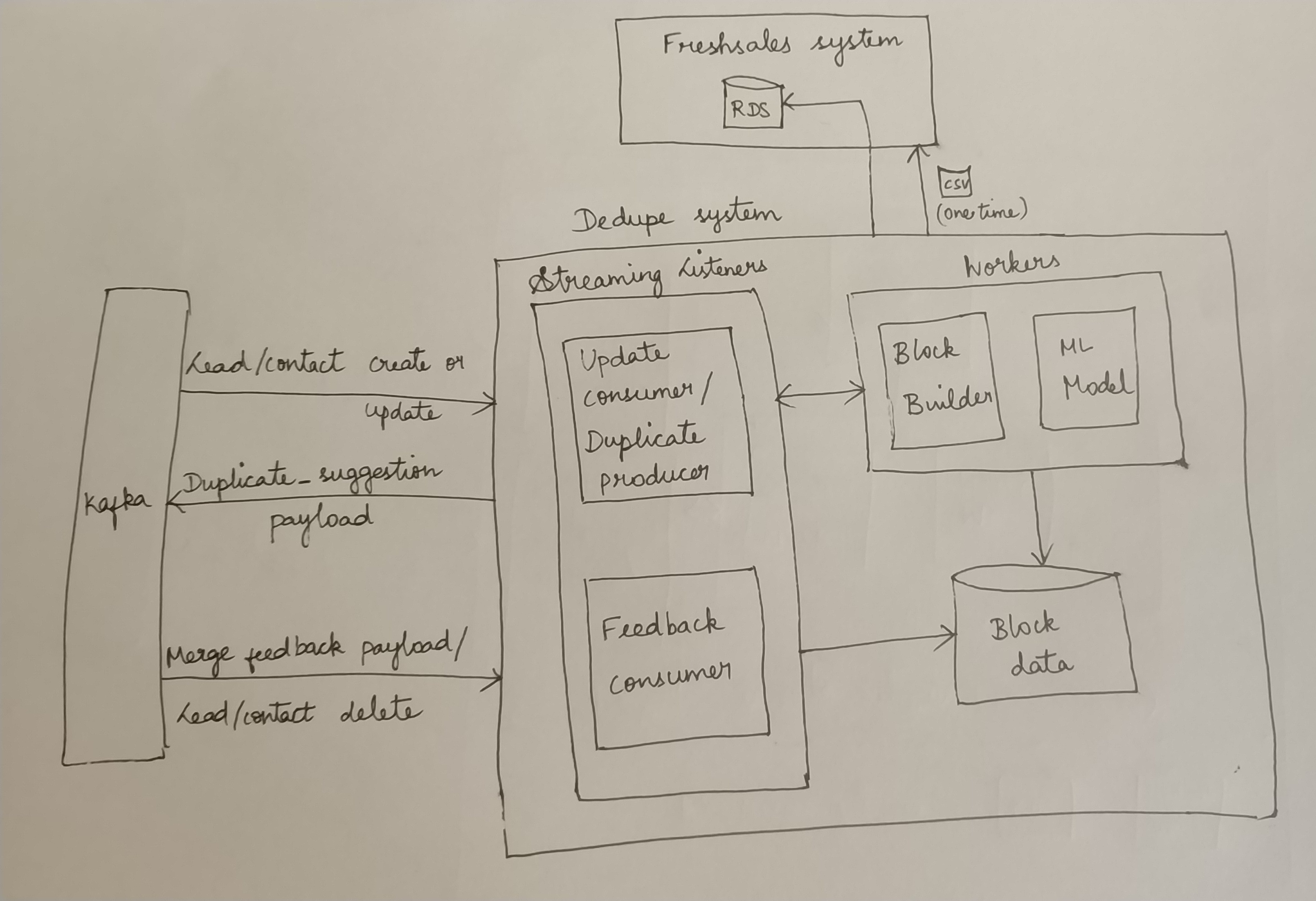
# Title: Design doc



Contracts:

From Freshsales:

* No changes needed in lead/contact create/update/delete
* A new payload type to be added for merge\_feedback.

merge\_feedback payload:

This can be same as lead\_create payload but payload\_type and action will have different values. The primary record's fields are passed through the "model\_properties"

It should additionally have "duplicates" which is the list of duplicates suggested. Each duplicate should contain their position in the list, whether it was merged by the user or not, whether it was a lead/contact and all the model properties of that particular duplicate.

{

        ...

"payload\_type":"merge\_feedback",

"payload":{

        ...

"action":"merge",

"model\_properties":{},

"duplicates":[

{

"rank":1,

"merged":true,

"model":"Lead",

"properties":{}

},

{},

...

]

}

}

To Freshsales:

* For every lead/contact create or update payload in kafka, we compute the duplicates for the incoming lead/contact and push the following payload to kafka .
* This payload consists of a list of duplicate\_pairs and their confidence score. Atleast one of each duplicate\_pair will be the incoming lead/contact.
* This duplicate pair could be a lead-lead or contact-contact or lead-contact pair.
* The duplicate\_pair is also symmetrical in nature i.e. if lead\_a and lead\_b are in a duplicate\_pair, lead\_a is duplicate of lead\_b and lead\_b is duplicate of lead\_a with the same confidence.   
  If (A,B) exists in our payload, we will not include (B,A) again.
* We do not take care of giving contacts higher priority than leads in duplicates. Leads and contacts are treated equally in the dedupe system to find all possible duplicates.
* We also do not restrict the number of duplicates for a particular lead/contact to a maximum limit(25 in the current system)

{

"payload\_type" : "duplicate\_suggestion",  
"payload":[

{  
"duplicate\_pair": [

{

"model": "lead",

"id" : 12345

},

{

"model": "contact",

"id" : 123

}

],

"confidence":0.8  
},

...

]

}

One-time CSV:

For all the existing leads/contacts, we will compute duplicates(lead\_1, lead\_2) and send a csv with the following fields.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [s.no](http://s.no) | lead\_1 | Model of lead\_1 | lead\_2 | Model of lead\_2 | Confidence |
| 1 | 123 | lead | 345 | lead | 0.99 |
| 2 | 456 | contact | 578 | lead | 0.89 |
| 3 | 098 | contact | 234 | contact | 0.79 |

Queries to Freshsales team:

1. lead\_create/update payload traffic expected per min
2. Is there a spark deployment in the current freshsales system?
3. Procedure to get access to FS production system
4. Procedure to get RDS connectors in Freshsales system
5. Would they be okay with an additional x queries/min posted on their RDS from dedupe system?  
   Our assessment is x would be ~40\* no. of lead\_create/update queries per minute  
   Further analysis needed to arrive at a more precise estimate
6. Agreement between teams on all the new payloads listed and the csv format.
7. Difference between lead/contact delete and destroy
8. Number of accounts for which dedupe feature will be enabled. This is needed to design our datastore.
9. We will need the actual email\_ids(unhashed) for all leads and contacts so that the ML model could be trained.

Features not included in the beta version:

1. The streaming jobs are not distributed(to decided based on the number of accounts in beta version and kafka traffic expected).
2. Not deployed for all accounts, maybe a few paid accounts only.
3. There is no retraining of model
4. No bulk upload dedupe feature
5. No realtime deduplication before a lead is created.

To be decided by dedupe team:

* Efficient datastore for storing block data

It will be key value store.

Key → account\_id:block value

value → list of lead\_ids in that block

The amount of storage required will depend on the avg number blocks in an account, avg.size of blocks, number of accounts,etc.