Data Aggregation:

DataLake:

Ingestion:

- Since we creating datalake we want to move all the data to S3 (cheapest storage for datalake).
- **Data is saved in S3 as flat json parquet file that is compact and support schema evolution.**

Advantages:

- Datalake is cheap, compact, evolving schema based datapoints on S3.
- Once data is stored in S3 we can get insights using different reporting by extending writing diagnostic code like what we did for leak smart.
- We have written code that can be scaled horizontally using AWS cluster.
- Spark structured streaming and Spark MLLib can be used to provide different insights to customers.
- We can have HBase and Phoenix in front of S3 datalake to let customer run SQL like queries to get insights.

DataBase Schema

mysql> describe datapoints;

```
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| dsn           | varchar(127)  | NO   | PRI | NULL    |       |
| property_name | varchar(191)  | NO   | PRI | NULL    |       |
| oem_id        | varchar(127)  | YES  | MUL | NULL    |       |
| oem_model     | varchar(255)  | YES  |     | NULL    |       |
| uuid          | varchar(36)   | YES  |     | NULL    |       |
| base_type     | varchar(36)   | YES  |     | NULL    |       |
| display_name  | varchar(191)  | YES  |     | NULL    |       |
| value_string  | varchar(255)  | YES  |     | NULL    |       |
| value_boolean | tinyint(4)    | YES  |     | NULL    |       |
| value_integer | int(11)       | YES  |     | NULL    |       |
| value_float   | float         | YES  |     | NULL    |       |
| value_decimal | decimal(10,0) | YES  |     | NULL    |       |
| value_text    | text          | YES  |     | NULL    |       |
| scope         | varchar(36)   | YES  |     | NULL    |       |
| direction     | varchar(36)   | YES  |     | NULL    |       |
| echo          | tinyint(4)    | YES  |     | NULL    |       |
```

```
| closed        | tinyint(4)    | YES  |     | NULL    |       |
| user_uuid     | varchar(127)  | YES  |     | NULL    |       |
| created_at    | datetime(6)   | YES  |     | NULL    |       |
| updated_at    | datetime(6)   | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
20 rows in set (0.01 sec)
```

Data Retention:

- Data retention refers to how long data is accessible to users within the Ayla platform.
  - Default is 90 days
  - Max for Real Time access is 90 days
  - QUESTION INTERNAL:
    - How long will data be held in S3 buckets?

Data Category	Data Type	Data Retention Period	Real Time Retrieval (Via API, S3 Link or Dashboards)	Delayed Retrieval (via Customer Success Request)
Device Data	String, Boolean, Integer, Decimal, Message properties	3 years	3 months	2 years 9 months
	File properties	3 years	3 months	2 years 9 months
	Data Export Files	3 years	3 months	2 years 9 months
	Log Files	3 months	3 months	N/A
Mobile Data	Log Files	3 months	3 months	N/A
System Data	Device Metadata	20 years	20 years	N/A
	User Metadata	20 years	20 years	N/A
	Service Metadata	20 years	20 years	N/A

Reports	Insights Reports	1 year	1 year	N/A
	Transaction Reports	20 years	20 years	N/A

Data Export Schema

- When data is exported via S3 csv files are pulled.
- Here is the format of the csv headers:

NOTE: display_name, val_string,metadata) are base64 encoded values. These values are highlighted in Table 1.

Datapoint	**Datapoint Ack**	**Connection**	**Location**	**Registration**
oem_id	oem_model	oem_id	oem_id	oem_id
oem_model	dsn	oem_model	oem_mod el	oem_model
dsn	property_nam e	dsn	dsn	dsn
property_nam e	display_name	event_time	ip	user_uuid
display_name	base_type	user_uuid	lat	registered
base_type	time_uuid	status	long	registration_ty p e
time_uuid	created_at_fro m_d evice		provider	registered_at
created_at_fro m_d evice	updated_at		user_uuid	unregistered_ at
updated_at	created_at		created_at	
created_at	user_uuid			
user_uuid	echo			
echo	closed			
closed	discarded			
discarded	scope			
scope	val_int			
val_int	val_decimal			
val_decimal	val_float			
Datapoint	**Datapoint Ack**	**Connection**	**Location**	**Registration**
val_float	val_boolean			
val_boolean	val_string			

val_string	metadata			
metadata	ack_message			
direction (input/ouput)	ack_status			
	ack_id			
	acked_at			

Datapoint Schema via API Access:

- When ‘paginated = TRUE’ Format-A is used:

```
{
  "meta": {
    "previous_page": "1234abcd-1234-1234-1234-ab0000000001",
    "next_page": "1234abcd-1234-1234-1234-ab0000000005",
    "current_page_number": 1
  },
  "datapoints": [
    {
      "datapoint": {}
    },
    {
      "datapoint": {}
    }
  ]
}
```

- When ‘paginated = FALSE’ Format-B is used:

```
[
  {
    "datapoint": {}
  },
  {
    "datapoint": {}
  }
]
```

Questions from Thermacell:

Could you provide the current overall database schema used by Ayla to store data?

- Can you confirm how long you’re storing the data?
  -
- Do you have a summary of what actual fields are currently being sent between hub and Ayla Cloud?
- Is Ayla currently doing any data aggregation?