SPARK+AI
SUMMIT 2018

Enterprise Data Governance and Compliance at Scale

Sri Esha Subbiah, ssubbiah@twilio.com, DataPlatform, Twilio
Sunil Patil, spatil@twilio.com, Data Platform, Twilio

twilio

Who are We?

Presenters / Q&A

- Sri Esha Subbiah
- Senior Engineering Manager, Data Platform
- https://www.linkedin.com/in/sri-esha-subbiah/
- @srieshas

- Sunil Patil
- Senior Software Engineer, Data Platform
- https://www.linkedin.com/in/wpcertification/
- @pppsunil

- Jeechee Chen
- Senior Software Engineer, Data Platform
- https://www.linkedin.com/in/jeechee/

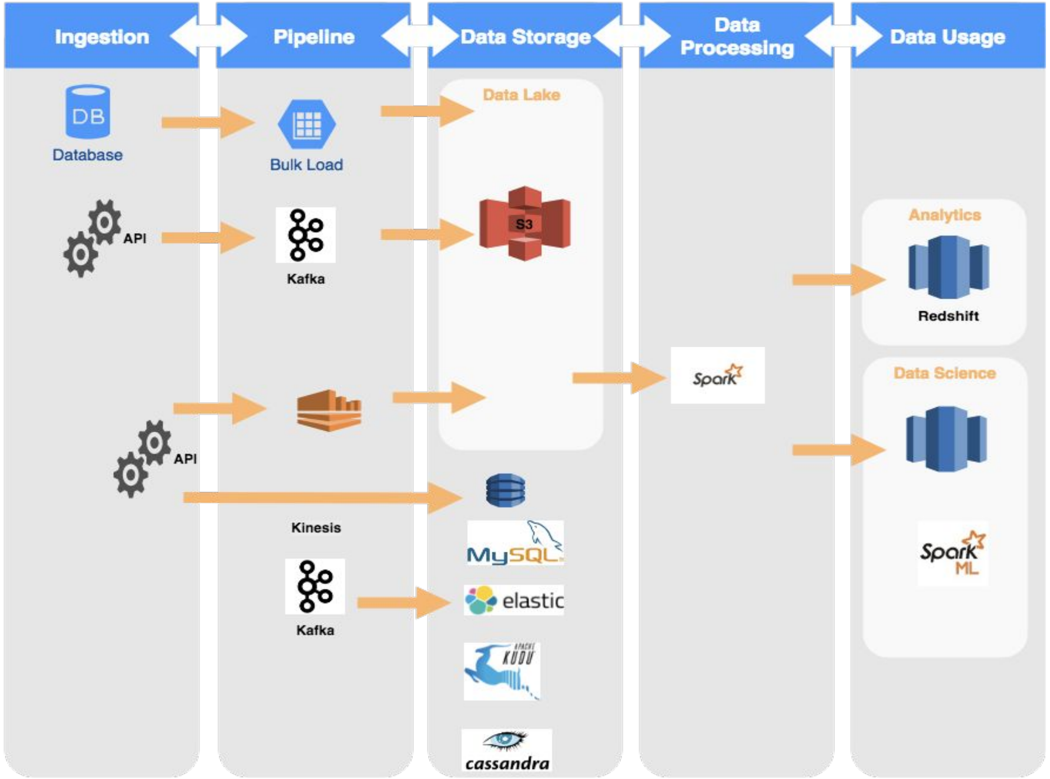
Communication Cloud

- Twilio Cloud Communication Platform provides programmable API for SMS, Voice, Video, IM Chat plus lots more.
- **Twilio's mission** is to fuel the future of communications
- 46000+ Customers, https://customers.twilio.com/
- 1 Billion Voice & Message data points per day
- 1.9 Million Developers
- 100+ Countries with varying compliance requirements

Twilio’s Data Platform

Scale

- 25+ teams
- 150K Messages/sec
- 30+ Brokers/ Nodes
- 210+ Kafka Topics
- 150+ Bulk Load
- Petabytes of data
- 350+ Cores Spark
- Multiple Sources
- Multiple Destinations



Factors to consider for Governance & Compliance

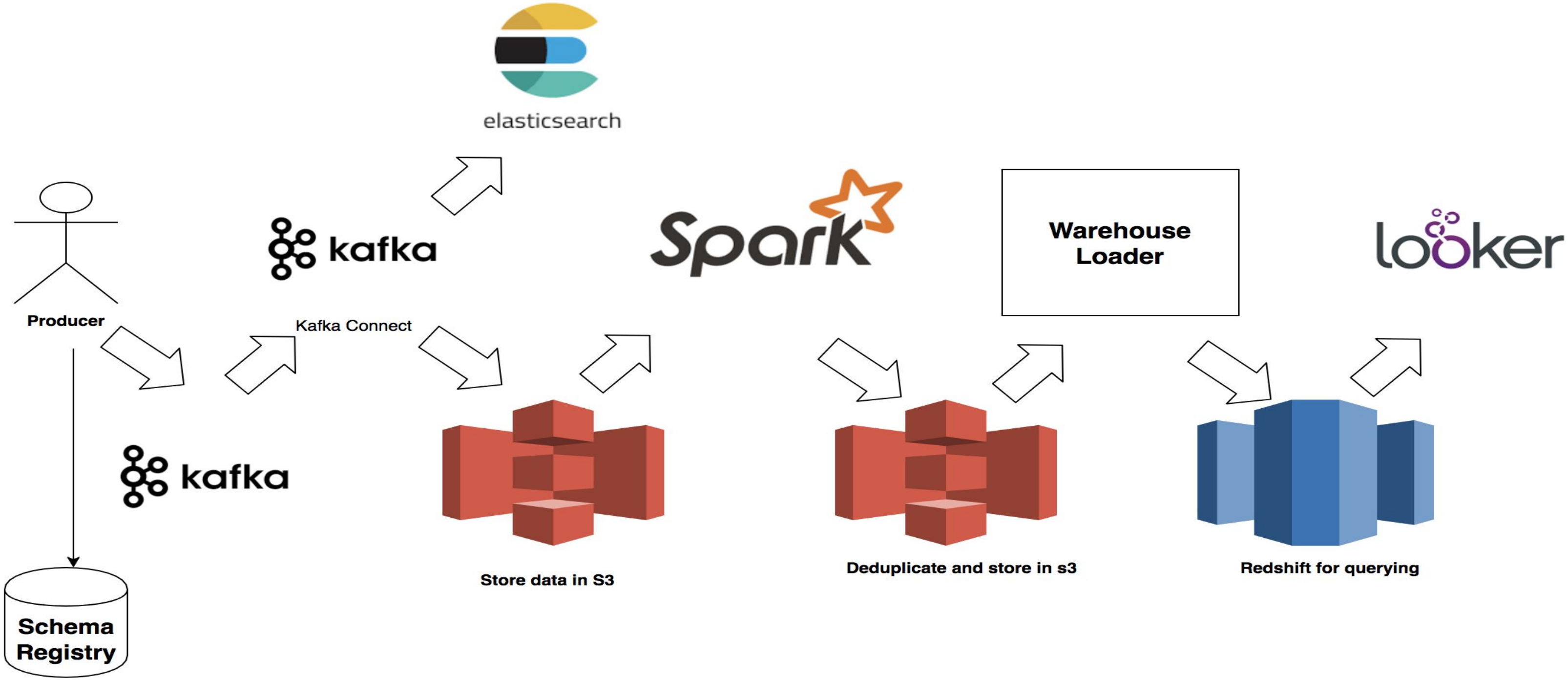
- Collect what is needed
- MetaData management
- Identify kinds of data and Classify
- Data cleansing and wrangling
- Enable easy onboarding
- Collaboration and accessibility
- Visualization of Data
- Data Lineage
- Security
- Auditing
- Data Retention and Cleanup
- Compliance: SOX , GDPR, HIPAA, PCI

GDPR

- Personal Data
- Data Processing and Data Subjects
- Processor and Controller

Obligations	Measures
● Lawfulness, fairness, and transparency ● Purpose limitation ● Data minimization ● Accuracy ● Storage limitation ● Integrity and confidentiality ● Accountability	❖ Secure Storage ❖ Anonymization ❖ Encryption ❖ Retention Policies ❖ Deletion of Data ❖ Auditing ❖ Access Control

Kafka Pipeline



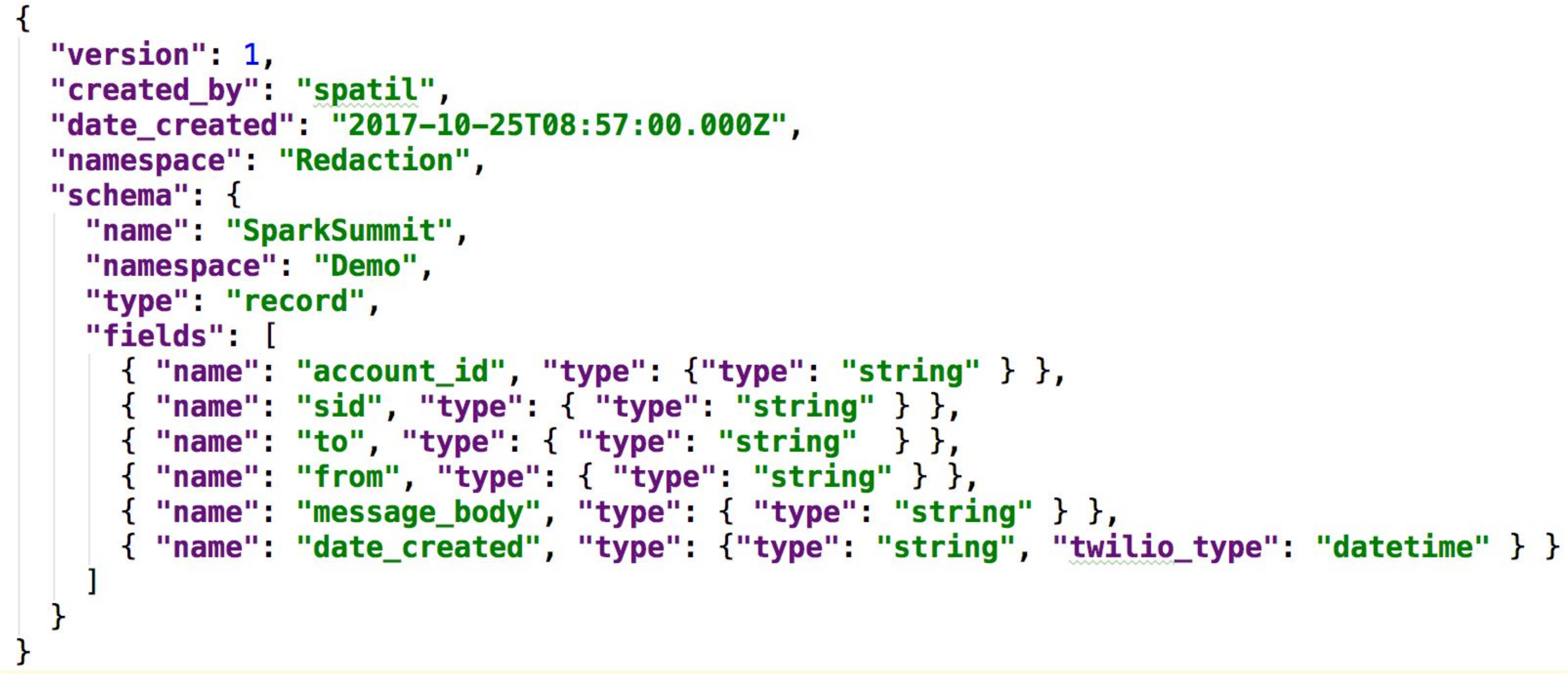
Schema Registry

Schema registry is a dynamo backed REST service that is used by different team in Twilio

- JVM client for producing and consuming compliant data
- HTTP API for producing JSON compliant with Schema
- API for managing schema entities
- API for storing schema entity to topic mapping
- Each Kafka topic has schema entity associated with it

Entity -> Topic -> Redshift Table/ ElasticSearch Index

Sample Schema



```
{
  "version": 1,
  "created_by": "spatil",
  "date_created": "2017-10-25T08:57:00.000Z",
  "namespace": "Redaction",
  "schema": {
    "name": "SparkSummit",
    "namespace": "Demo",
    "type": "record",
    "fields": [
      { "name": "account_id", "type": {"type": "string" } },
      { "name": "sid", "type": { "type": "string" } },
      { "name": "to", "type": { "type": "string"  } },
      { "name": "from", "type": { "type": "string" } },
      { "name": "message_body", "type": { "type": "string" } },
      { "name": "date_created", "type": {"type": "string", "twilio_type": "datetime" } }
    ]
  }
}
```

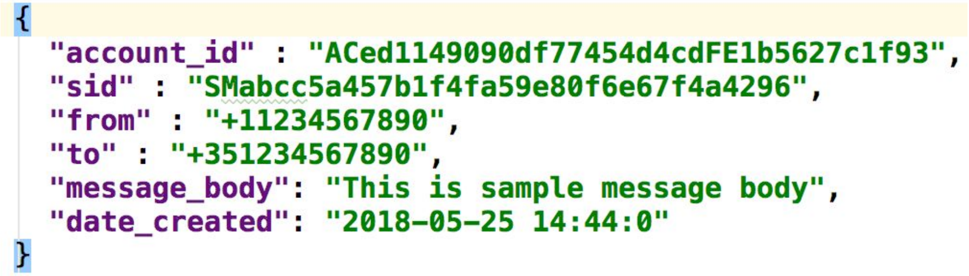
Anonymization - Redaction

Redaction

Redaction is removing PII information in type specific manner

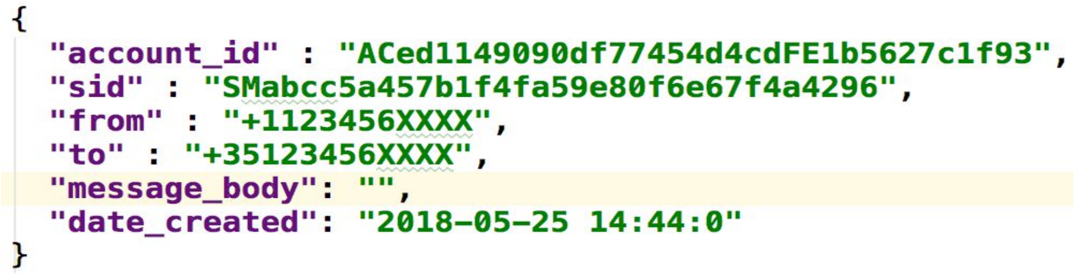
1. Phone Number:- Remove last 4 digits
2. Email :- Remove everything but first letter and domain
3. Customer Text: - Remove completely

Input



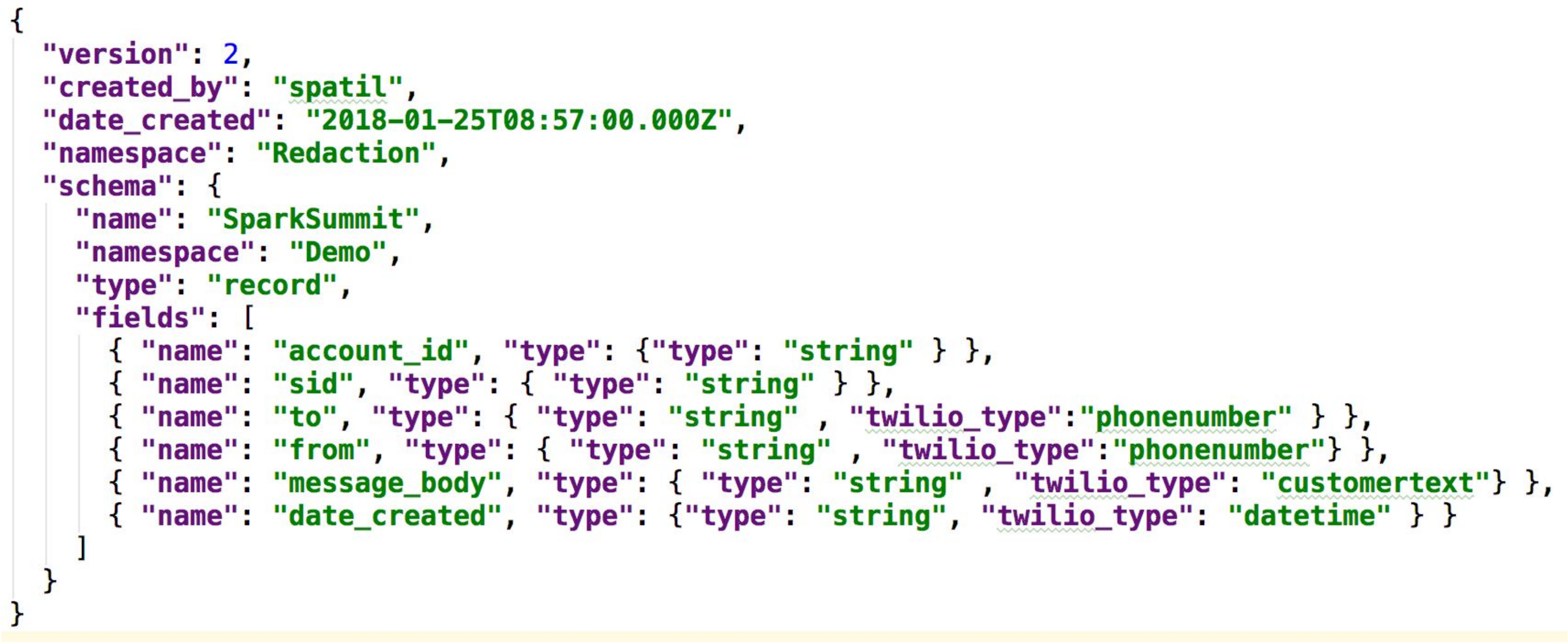
```
{
    "account_id" : "ACed1149090df77454d4cdFE1b5627c1f93",
    "sid" : "SMabcc5a457b1f4fa59e80f6e67f4a4296",
    "from" : "+11234567890",
    "to" : "+351234567890",
    "message_body": "This is sample message body",
    "date_created": "2018-05-25 14:44:0"
}
```

Redacted Output



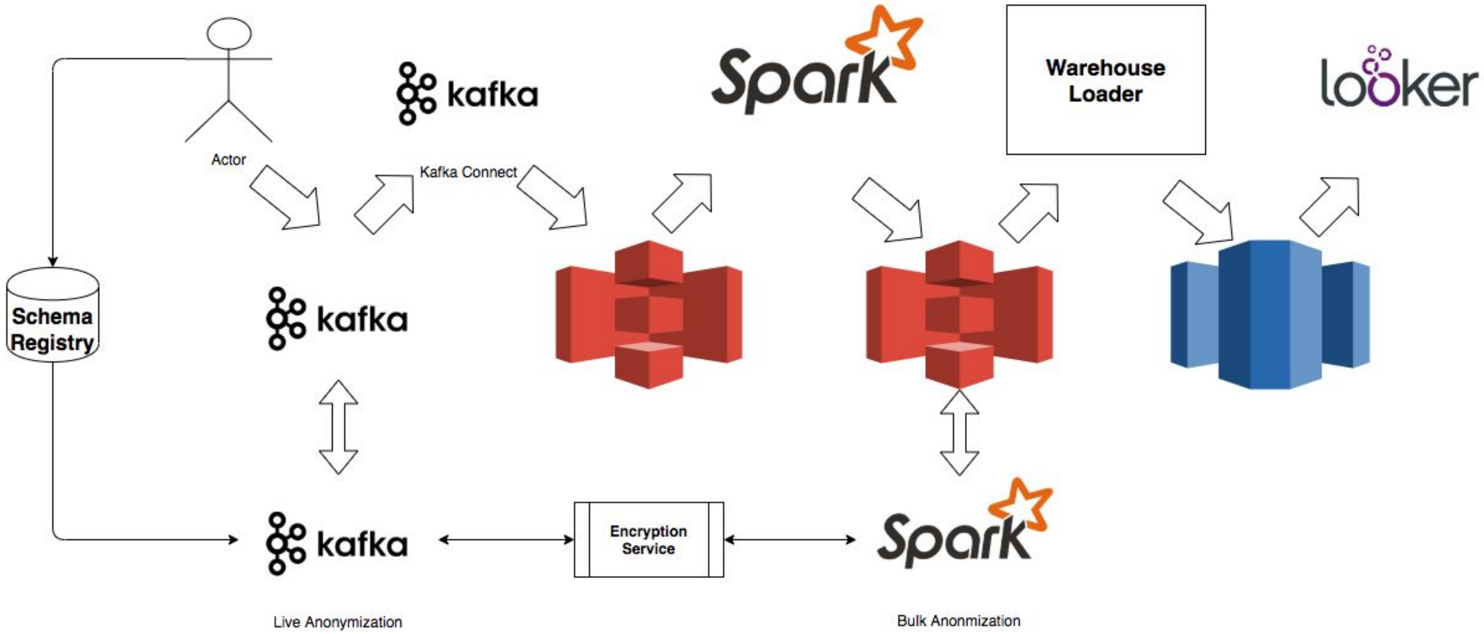
```
{
    "account_id" : "ACed1149090df77454d4cdFE1b5627c1f93",
    "sid" : "SMabcc5a457b1f4fa59e80f6e67f4a4296",
    "from" : "+1123456XXXX",
    "to" : "+35123456XXXX",
    "message_body": "",
    "date_created": "2018-05-25 14:44:0"
}
```

Sample Schema with Twilio Type



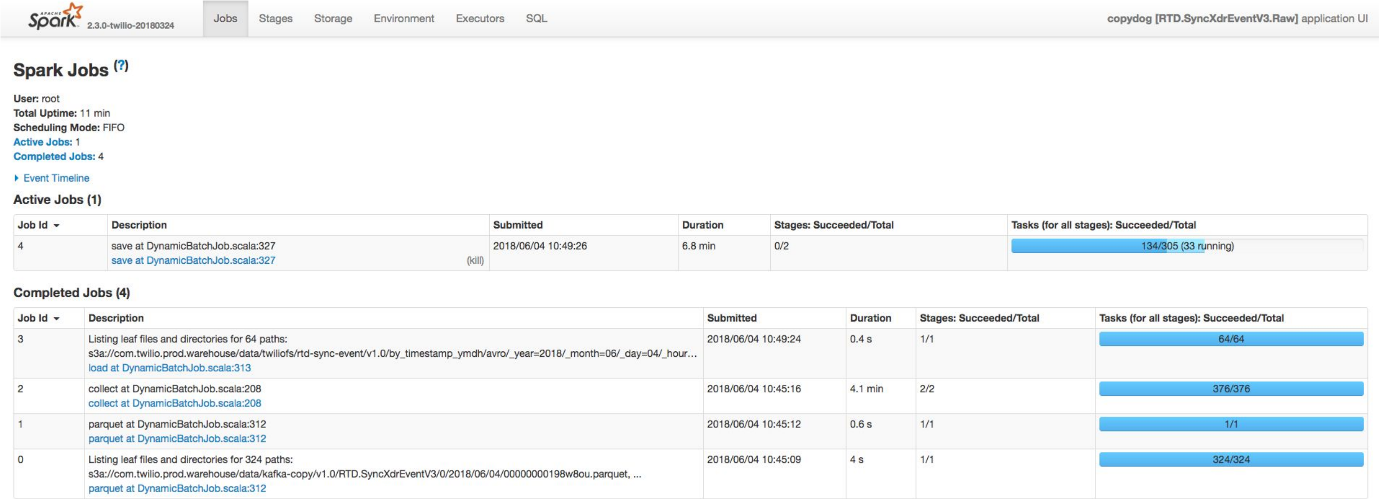
```
{
  "version": 2,
  "created_by": "spatil",
  "date_created": "2018-01-25T08:57:00.000Z",
  "namespace": "Redaction",
  "schema": {
    "name": "SparkSummit",
    "namespace": "Demo",
    "type": "record",
    "fields": [
      { "name": "account_id", "type": {"type": "string" } },
      { "name": "sid", "type": { "type": "string" } },
      { "name": "to", "type": { "type": "string" , "twilio_type":"phonenumber" } },
      { "name": "from", "type": { "type": "string" , "twilio_type":"phonenumber"} },
      { "name": "message_body", "type": { "type": "string" , "twilio_type": "customertext"} },
      { "name": "date_created", "type": {"type": "string", "twilio_type": "datetime" } }
    ]
  }
}
```

Compliance - Anonymization Architecture



Historical Anonymization - Spark

```
def recordRedacter(records: DataFrame): DataFrame = {
  records
    .withColumn("from_number", anonymizeRecordToStringUdf(lit(TwilioTypeFactory.PHONENUMBER: String), $"from_number"))
    .withColumn("to_number", anonymizeRecordToStringUdf(lit(TwilioTypeFactory.PHONENUMBER: String), $"to_number"))
    .withColumn("message_body", encryptStringUdf($"account_sid", $"message_body"))
    .withColumn("account_name", lit("": String))
}
```



copydog [RTD.SyncXdrEventV3.Raw] application UI

Spark Jobs (?)

User: root
Total Uptime: 11 min
Scheduling Mode: FIFO
Active Jobs: 1
Completed Jobs: 4

Active Jobs (1)

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
4	save at DynamicBatchJob.scala:327 save at DynamicBatchJob.scala:327 (kill)	2018/06/04 10:49:26	6.8 min	0/2	134/305 (33 running)

Completed Jobs (4)

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
3	Listing leaf files and directories for 64 paths: s3a://com.twilio.prod.warehouse/data/twiliofs/rtd-sync-event/v1.0/by_timestamp_ymdh/avro/_year=2018/_month=06/_day=04/_hour... load at DynamicBatchJob.scala:313	2018/06/04 10:49:24	0.4 s	1/1	64/64
2	collect at DynamicBatchJob.scala:208 collect at DynamicBatchJob.scala:208	2018/06/04 10:45:16	4.1 min	2/2	376/376
1	parquet at DynamicBatchJob.scala:312 parquet at DynamicBatchJob.scala:312	2018/06/04 10:45:12	0.6 s	1/1	1/1
0	Listing leaf files and directories for 324 paths: s3a://com.twilio.prod.warehouse/data/kafka-copy/v1.0/RTD.SyncXdrEventV3/0/2018/06/04/00000000198w8ou.parquet, ... parquet at DynamicBatchJob.scala:312	2018/06/04 10:45:09	4 s	1/1	324/324

Anonymization - Encryption

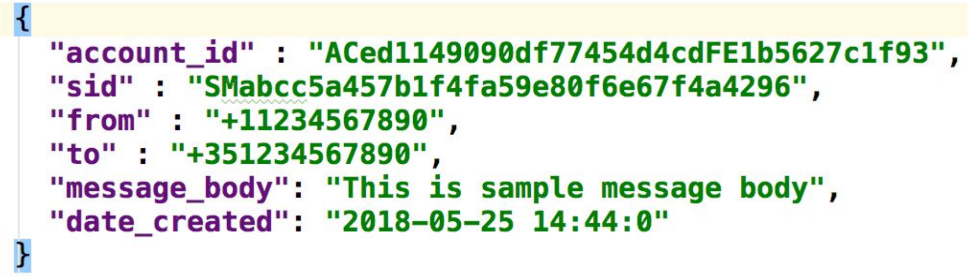
Encryption

Twilio has field level encryption in addition to volume level encryption.

Encryption & Decryption API

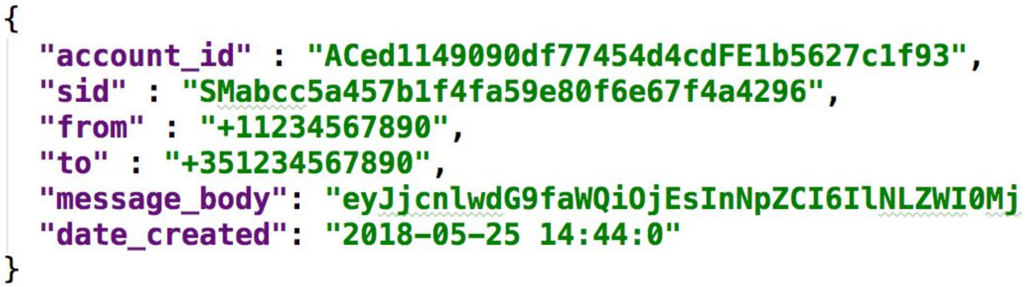
1. Input: AccountId, Value that needs encrypting
2. Output: Encrypted value in base64
3. Uses account specific encryption key
4. Provides point and bulk API
5. Symmetric key encryption

Input



```
{
  "account_id" : "ACed1149090df77454d4cdFE1b5627c1f93",
  "sid" : "SMabcc5a457b1f4fa59e80f6e67f4a4296",
  "from" : "+11234567890",
  "to" : "+351234567890",
  "message_body": "This is sample message body",
  "date_created": "2018-05-25 14:44:0"
}
```

Encrypted Output



```
{
  "account_id" : "ACed1149090df77454d4cdFE1b5627c1f93",
  "sid" : "SMabcc5a457b1f4fa59e80f6e67f4a4296",
  "from" : "+11234567890",
  "to" : "+351234567890",
  "message_body": "eyJjcnlwdG9faWQiOjEsInNpZCI6IlNLZWI0Mj
  "date_created": "2018-05-25 14:44:0"
}
```

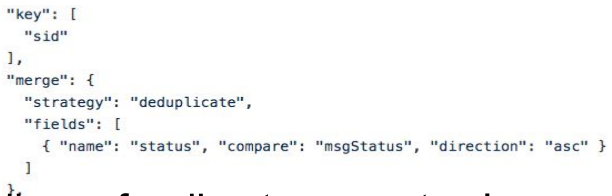
Data Lake(TwilioFS) and not Swamp

Challenges:

- Teams across Twilio store data in different places and different forms, difficult for internal teams to access
- Will Turn into Swamp if not managed

Solution:

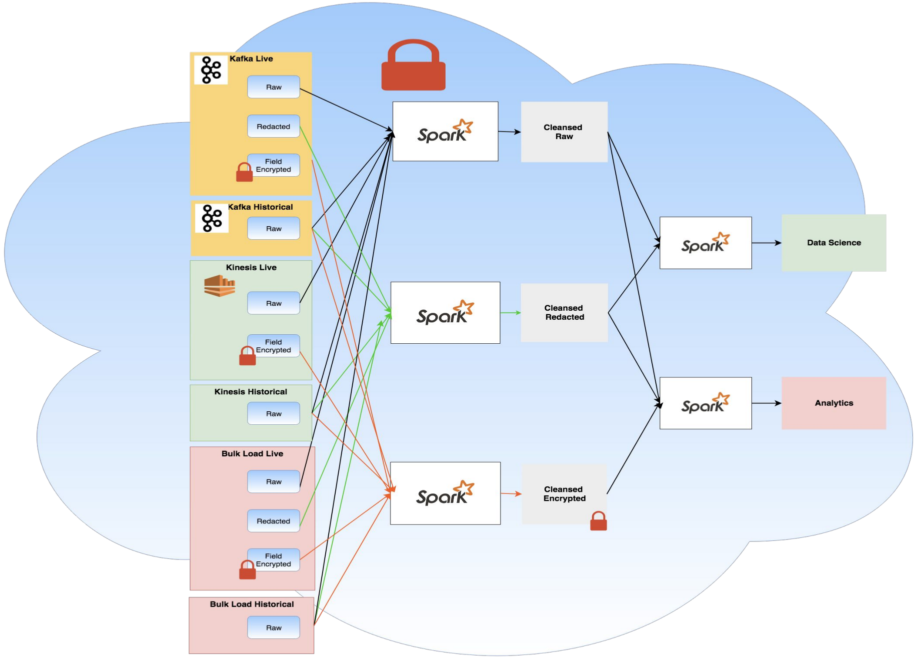
- Metadata Management: Descriptive, Structural, Administrative
- Deduplicated
- Standardized timestamps, indexing, directories, tags etc.
- Versioning, Encryption



```
"key": [
  "sid"
],
"merge": {
  "strategy": "deduplicate",
  "fields": [
    { "name": "status", "compare": "msgStatus", "direction": "asc" }
  ]
},
```

- Library for direct access to cleansed data in S3
- Access control based on Roles, IAM Rules and Type of Data
- Auditing using CloudTrail

Data cleansing and wrangling - Spark



Data Processing - Spark

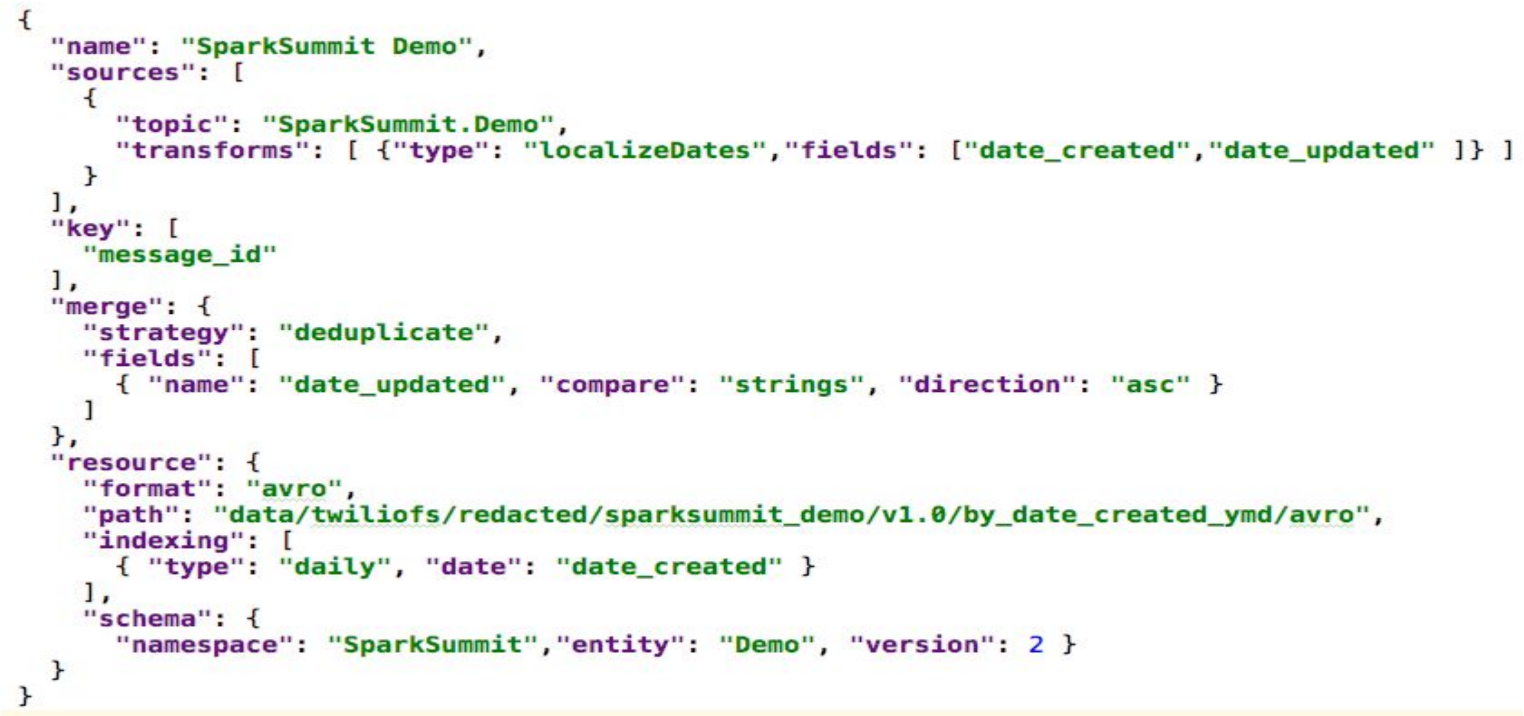
Challenges:

- Data lake in Petabytes Scale
- Various Data Processing requirements across different teams
- Compliance on a huge volume and Variety of data
- Migrating from One System to another
- Standing up a new System with all the historical data

Solutions:

1. Dynamic Transformers for entities
2. Transforming the data formats from sources.
3. Compliance: Bulk Redaction and Encryption processors using Spark
4. Transformation Library on standard Time zones, Indexing
5. Parquet format suitable for crunching
6. SparkSQL, Spark DataFrames, RDD, Spark Streaming, Spark MLib

Dynamic Transformers - Spark



```
{
  "name": "SparkSummit Demo",
  "sources": [
    {
      "topic": "SparkSummit.Demo",
      "transforms": [ {"type": "localizeDates","fields": ["date_created","date_updated" ]} ]
    }
  ],
  "key": [
    "message_id"
  ],
  "merge": {
    "strategy": "deduplicate",
    "fields": [
      { "name": "date_updated", "compare": "strings", "direction": "asc" }
    ]
  },
  "resource": {
    "format": "avro",
    "path": "data/twiliofs/redacted/sparksummit_demo/v1.0/by_date_created_ymd/avro",
    "indexing": [
      { "type": "daily", "date": "date_created" }
    ],
    "schema": {
      "namespace": "SparkSummit","entity": "Demo", "version": 2 }
  }
}
```

Data Deletion and Retention - Spark

Requirements:

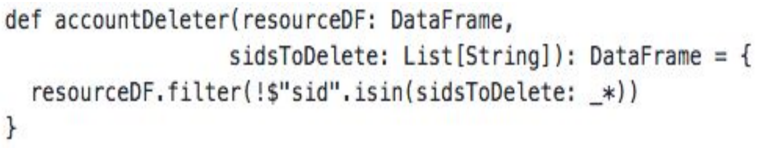
- deleting our customers' data
- deleting their customers' data
- customer data legal holds
- Customer Initiated ~200k deletions/day

Challenges:

- Deleting data in Data Lake is not as simple as DB
- Number of days that we have to delete can go back to 3285 days
- Indexing and Deletion Strategy

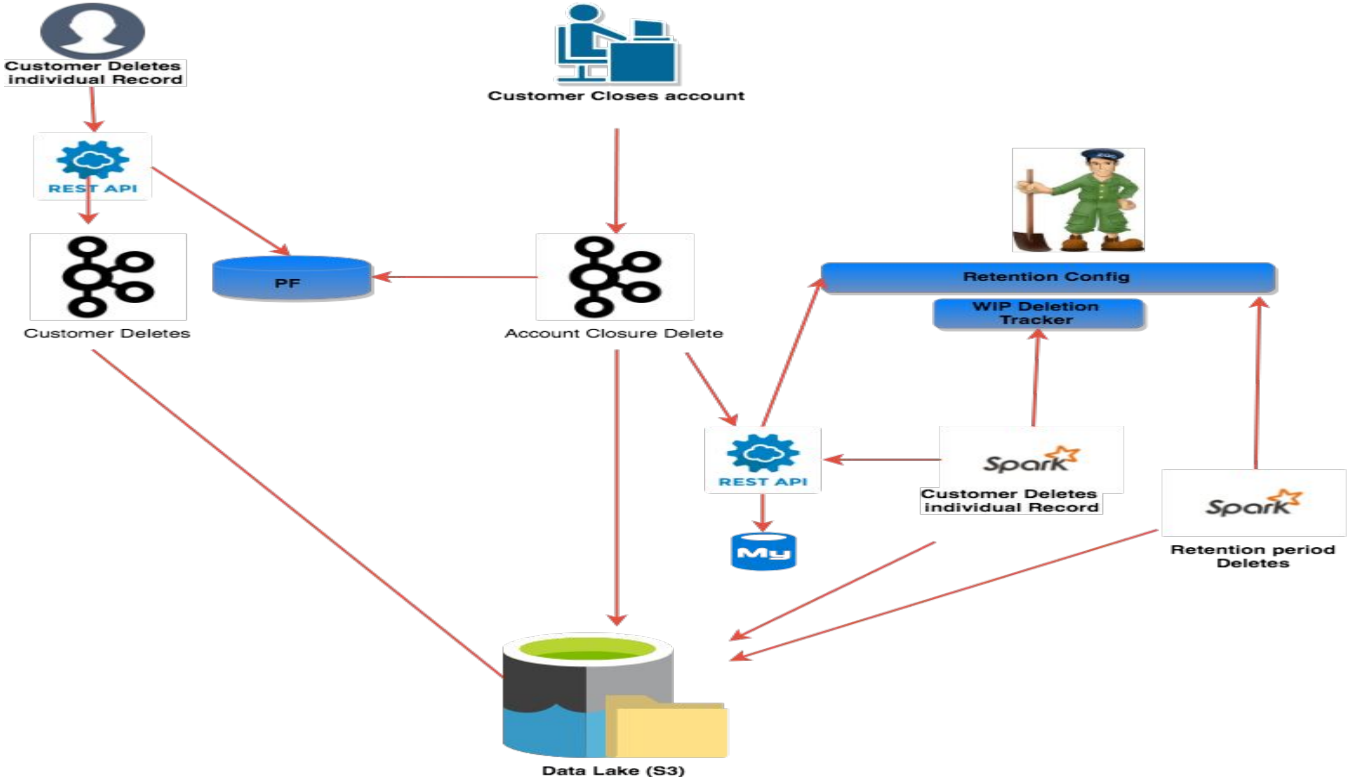
Solution:

- Spark for deleting and migrating data in bulk
  - Distributed
  - Simplicity - few lines of code to achieve variety of deletions, SparkSQL
  - Scalability
- Load Testing & Tuning Executors



```
def accountDeleter(resourceDF: DataFrame,
                   sidsToDelete: List[String]): DataFrame = {
  resourceDF.filter(!$"sid".isin(sidsToDelete: _*))
}
```

Compliance - Retention and Deletion



Spark Deletion - Performance & Capacity

- 3 Approaches have been analyzed: Account Index, Group Index, Day Index

Account Index

Executor Core	Driver Core	Memory	Files Processed	Duration in Hours
100	50	50g	12	Failed
128	110	110g	24	42

Group Index

Executor Core	Driver Core	Memory	Files Processed	Duration in Hours
64	32	100	288(24 hour)	72
32	16	100	288(24 hour)	168

Spark Deletion - Learnings

- Make sure Data lake is indexed, partitioned appropriately
- Consider IO: Too many small files and too many indexes
- Need for tracking & Locking if job runs longer

Day Index

Worker Core	Date Range	Indexes Affected	Duration in Hours
56	2018-01-08 through 2018-01-15	115	1.5
56	2018-01-15 through 2018-01-22	260	2.6
56	2018-01-22 through 2018-01-	452	4.8

What Next?

Self Service at all Layers

Related Links

Twilio’s GDPR White Paper:
https://s3.amazonaws.com/ahoy-assets.twilio.com/Whitepapers/Twilio_Whitepaper_GDPR.pdf

PII Description:

https://www.twilio.com/blog/2018/05/personally-identifiable-information-pii-fields-twilio-docs-gdpr-compliance.html

Twilio’s Support:

https://www.twilio.com/blog/2017/09/twilios-gdpr-commitment-support-for-customer-compliance-objectives.html

We are Hiring

Twilio Job Board: https://www.twilio.com/company/jobs

Sr. Engineering Manager: https://boards.greenhouse.io/twilio/jobs/961366

Sr. Software Engineer: https://boards.greenhouse.io/twilio/jobs/1101370

Thank You, Q & A

Twilio’s Compliance Officer

Sheila Jambekar

https://www.linkedin.com/in/sheilajambekar/

@sheilajambekar