Deduplicating Cloud Functions - Sprint 5

Beliz Kaleli | Vikash Sahu | Paritosh Shirodkar | Asutosh Patra

The Purpose

The purpose of this project is to design and implement a deduplication framework for serverless platform in order to improve overall throughput of the platform.

Recap

Sprint - 1

- Familiarizing with Serverless Technology

Sprint - 2

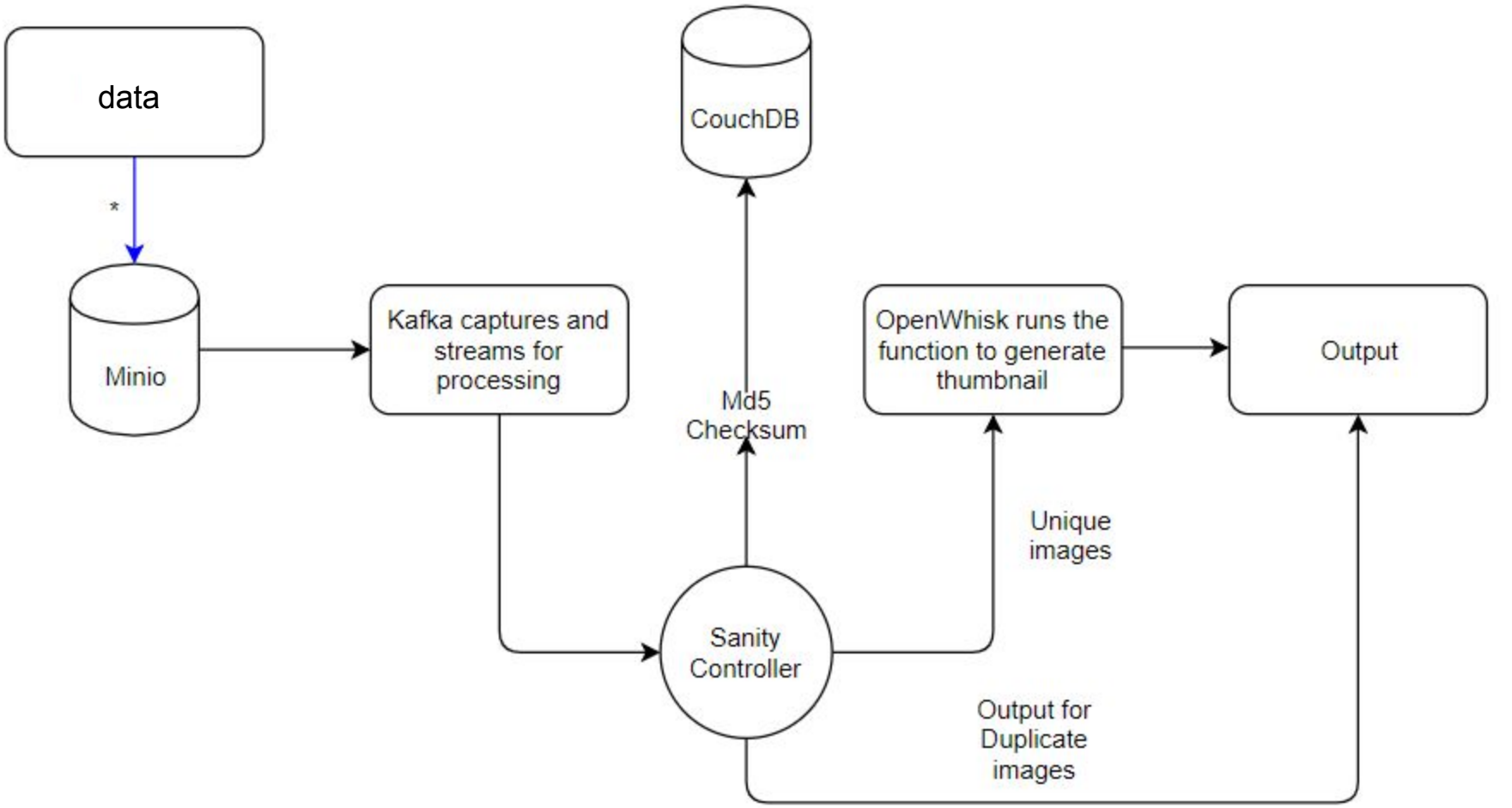
- Setting up all the components (Kafka, Minio, CouchDB and OpenWhisk)

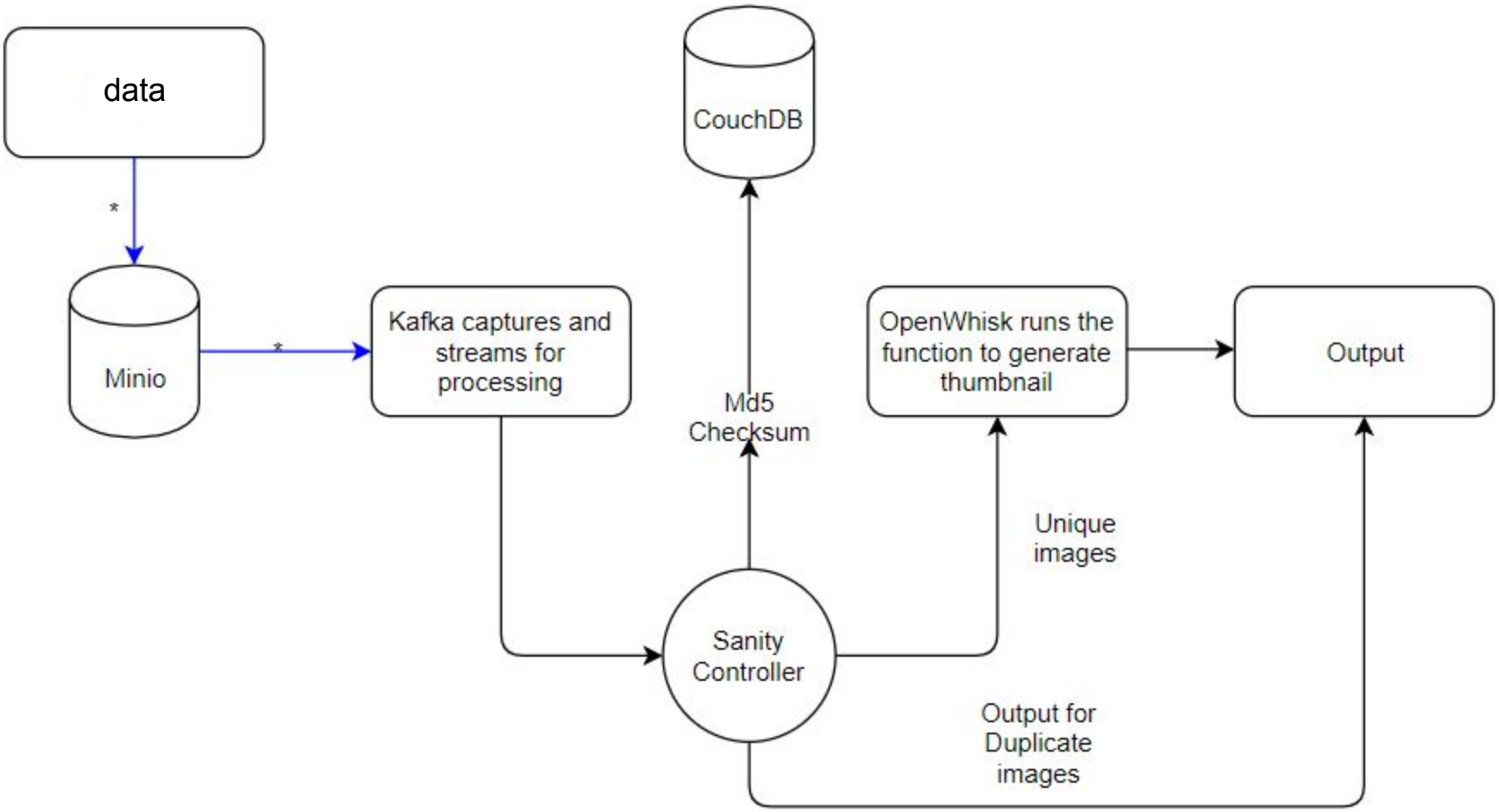
Sprint - 3

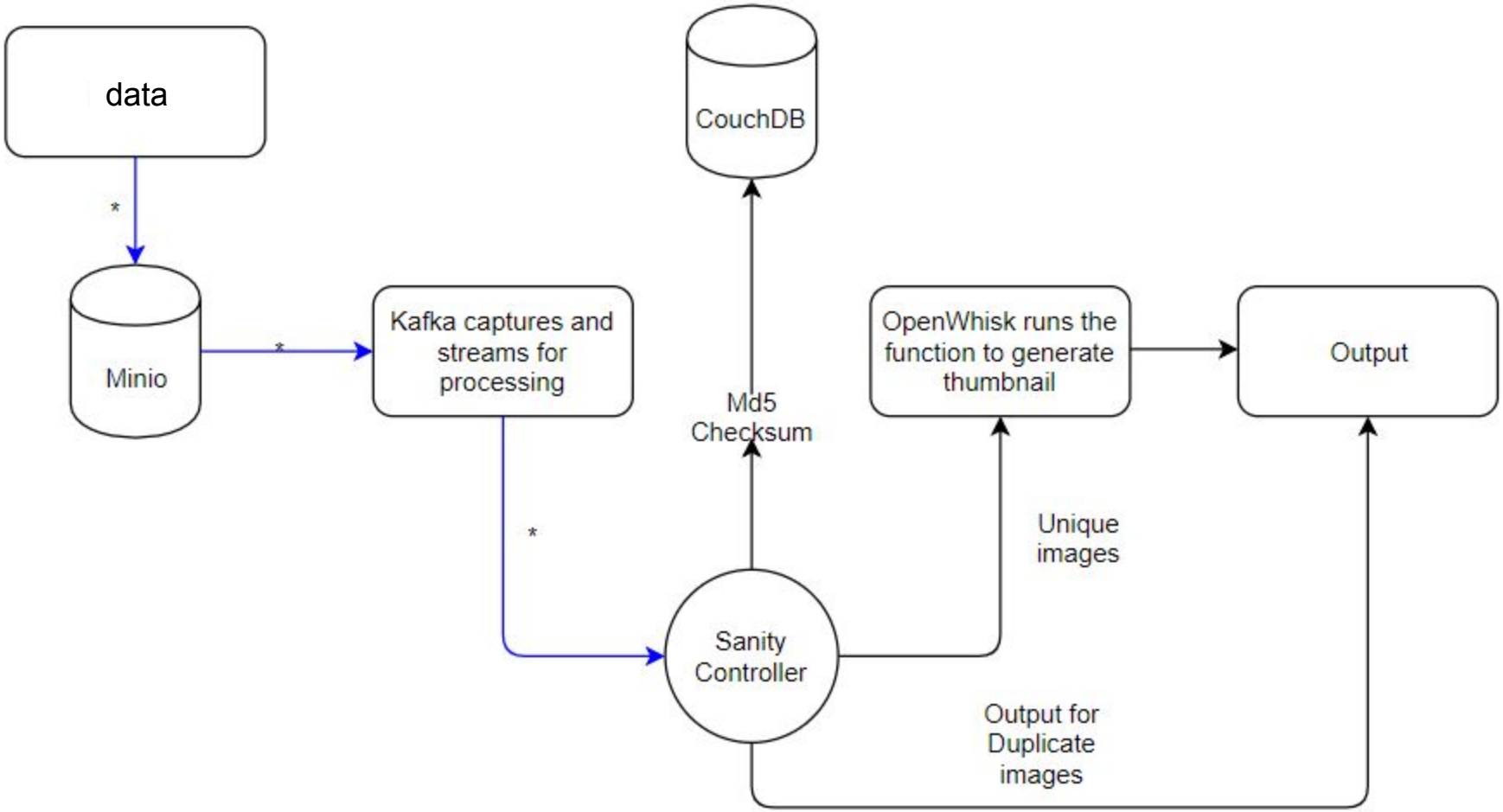
- Developed deduplication framework for Image Thumbnail Use Case in IBM Cloud

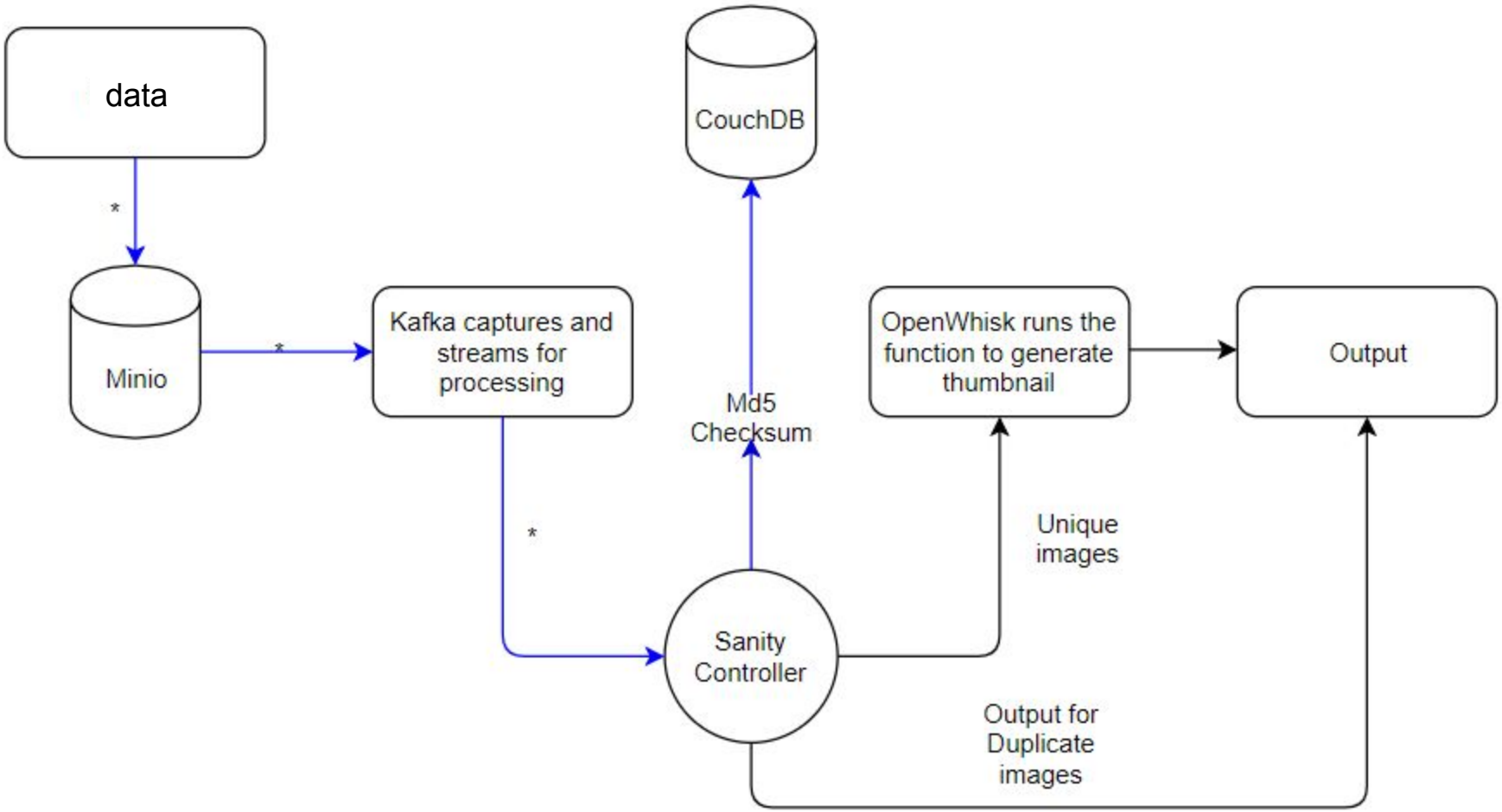
Sprint - 4

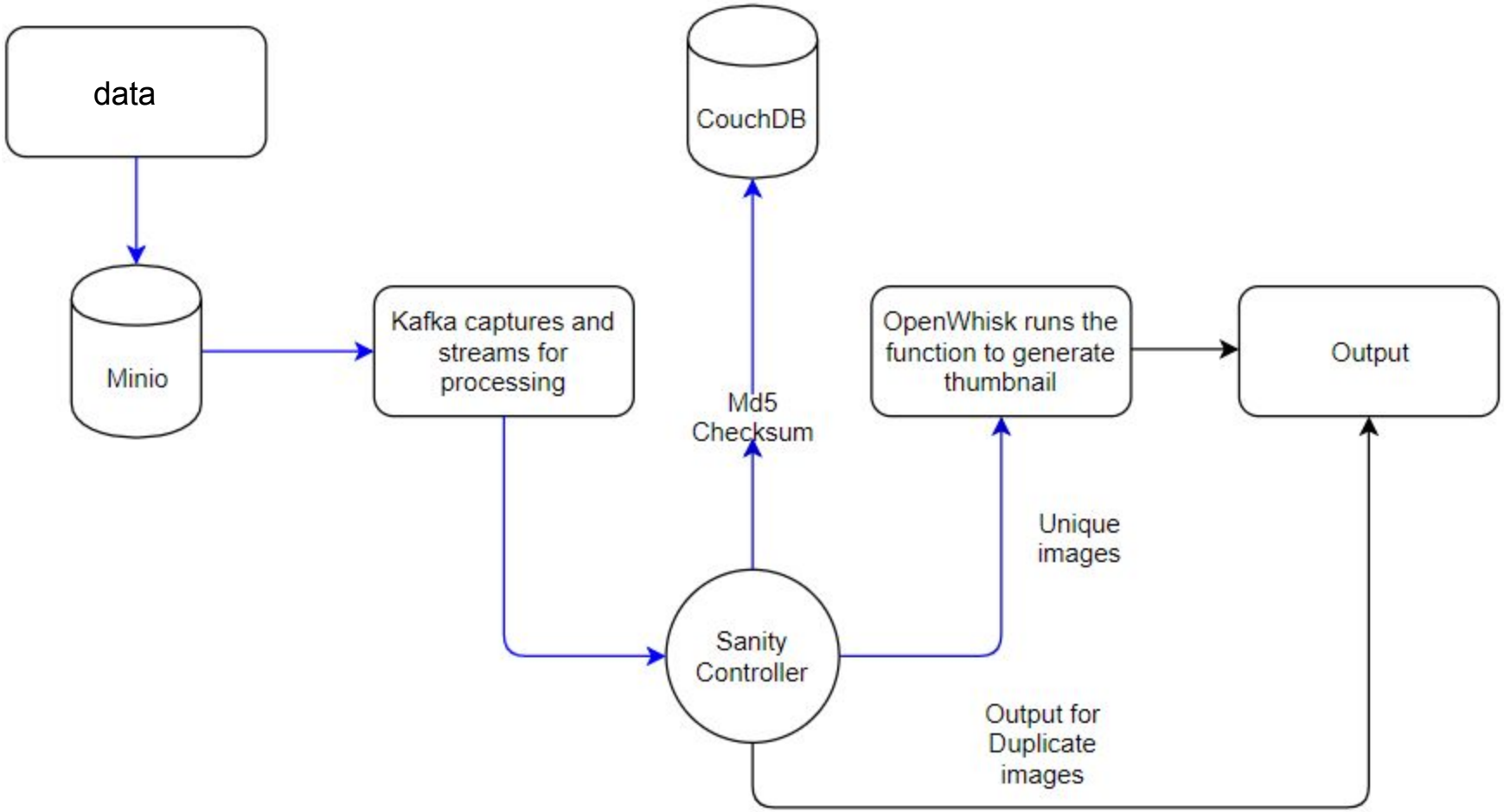
- Generalized our framework by implementing different use cases
- Designed our CLI

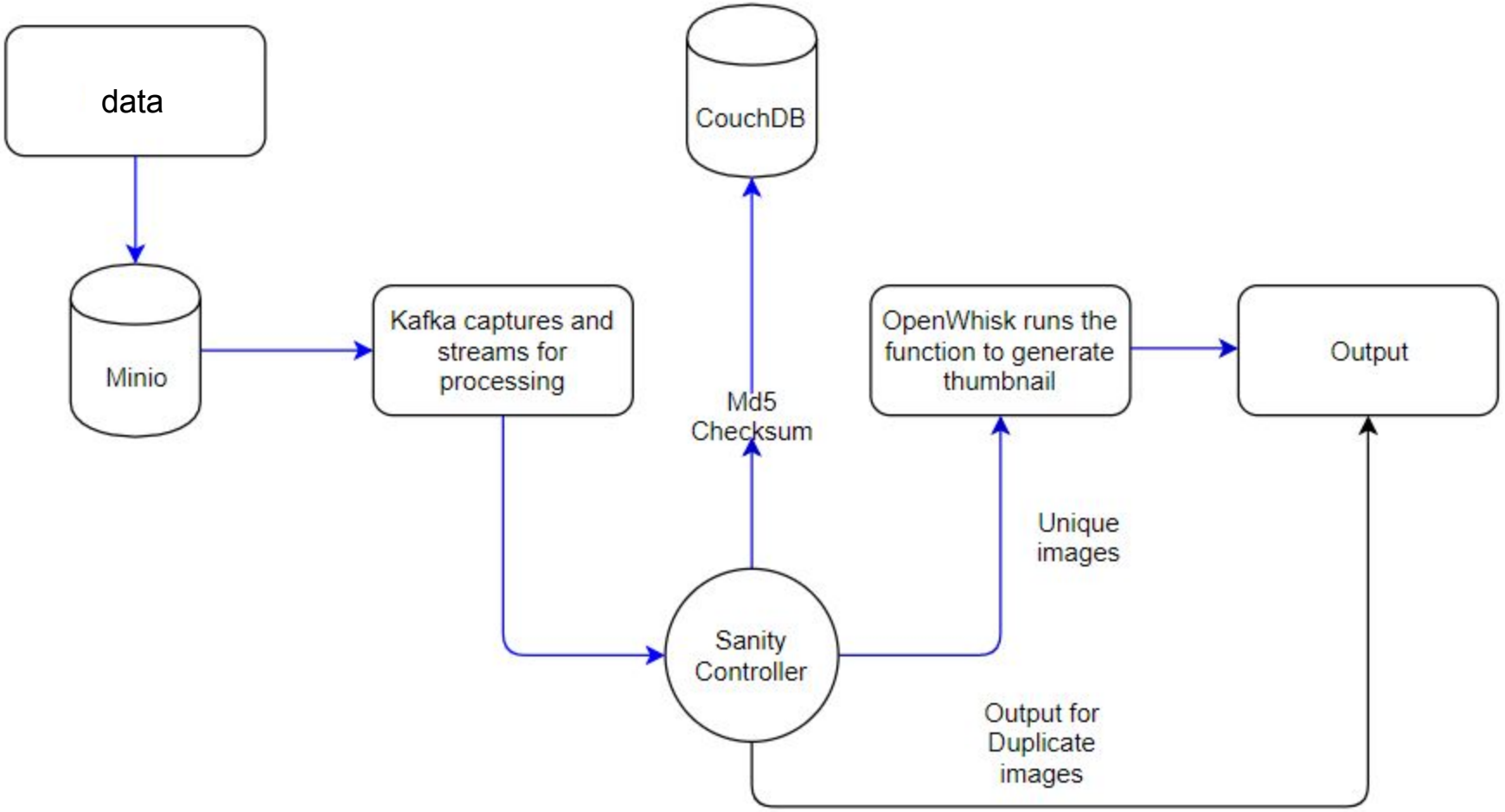

data
*
Minio
Kafka captures and streams for processing
CouchDB
Md5 Checksum
Sanity Controller
Unique images
OpenWhisk runs the function to generate thumbnail
Output
Output for Duplicate images

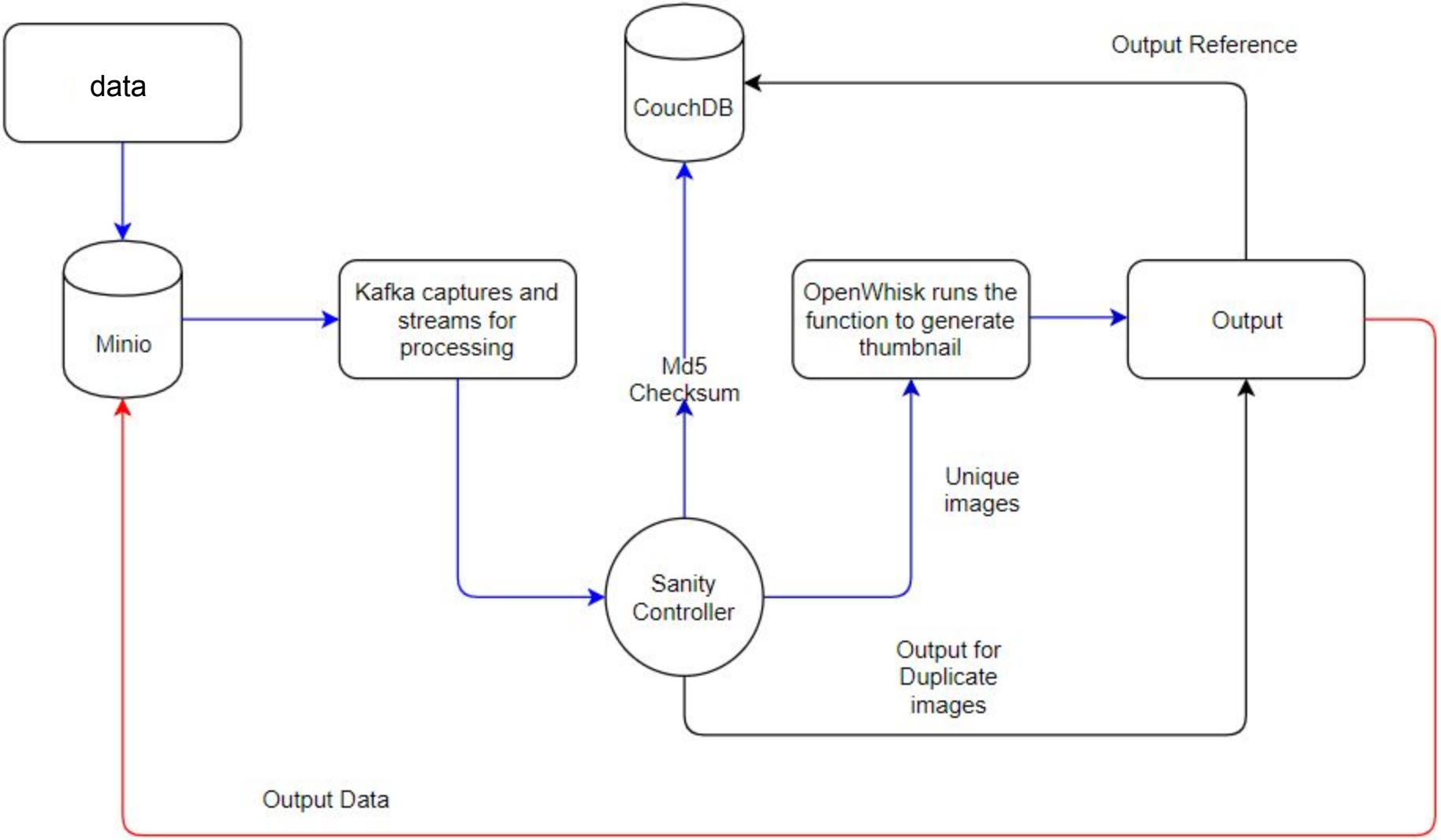

data
*
Minio
*
Kafka captures and streams for processing
CouchDB
Md5 Checksum
Sanity Controller
Unique images
OpenWhisk runs the function to generate thumbnail
Output
Output for Duplicate images

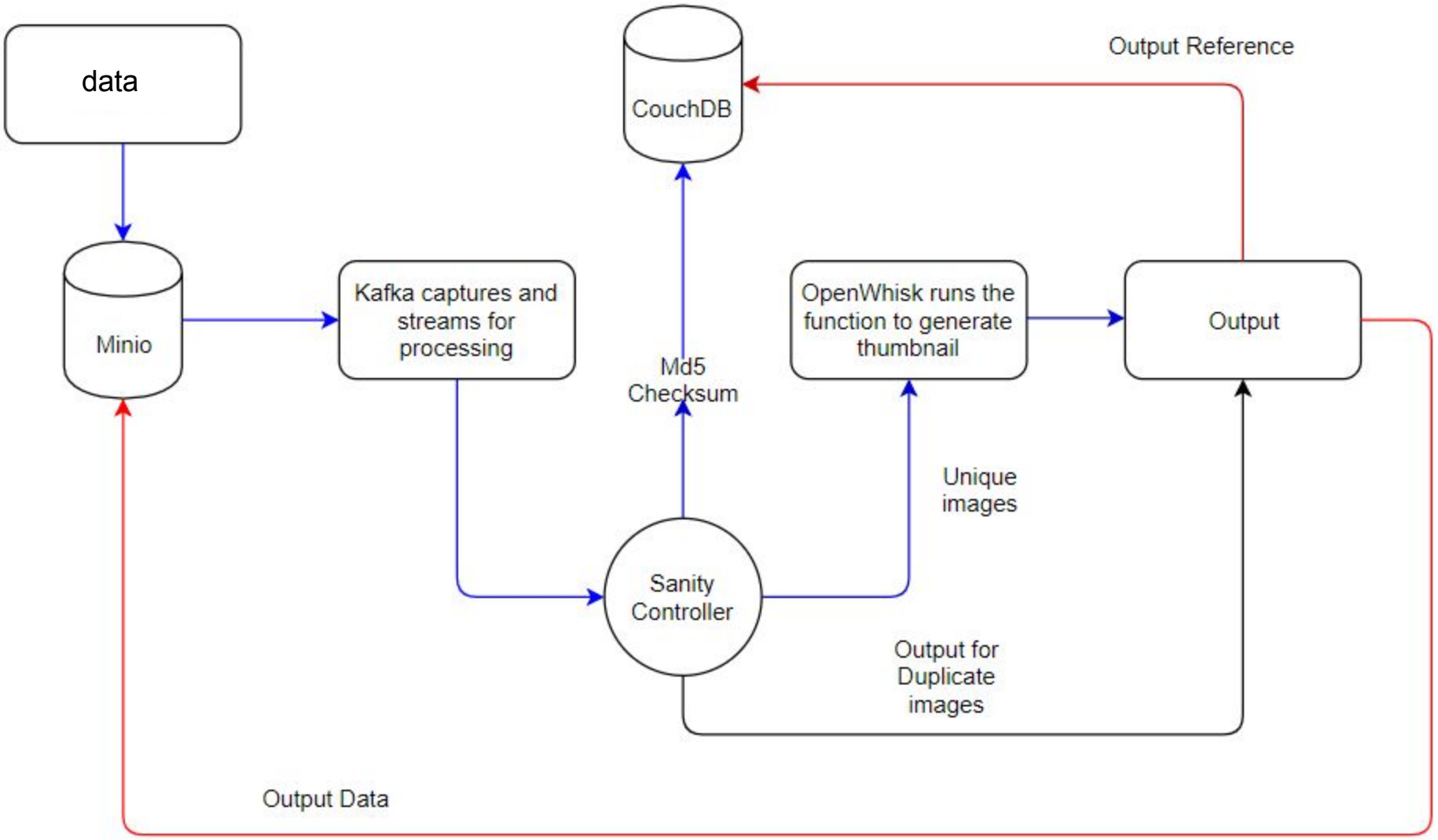

data
*
Minio
*
Kafka captures and streams for processing
*
CouchDB
Md5 Checksum
Sanity Controller
Unique images
OpenWhisk runs the function to generate thumbnail
Output
Output for Duplicate images

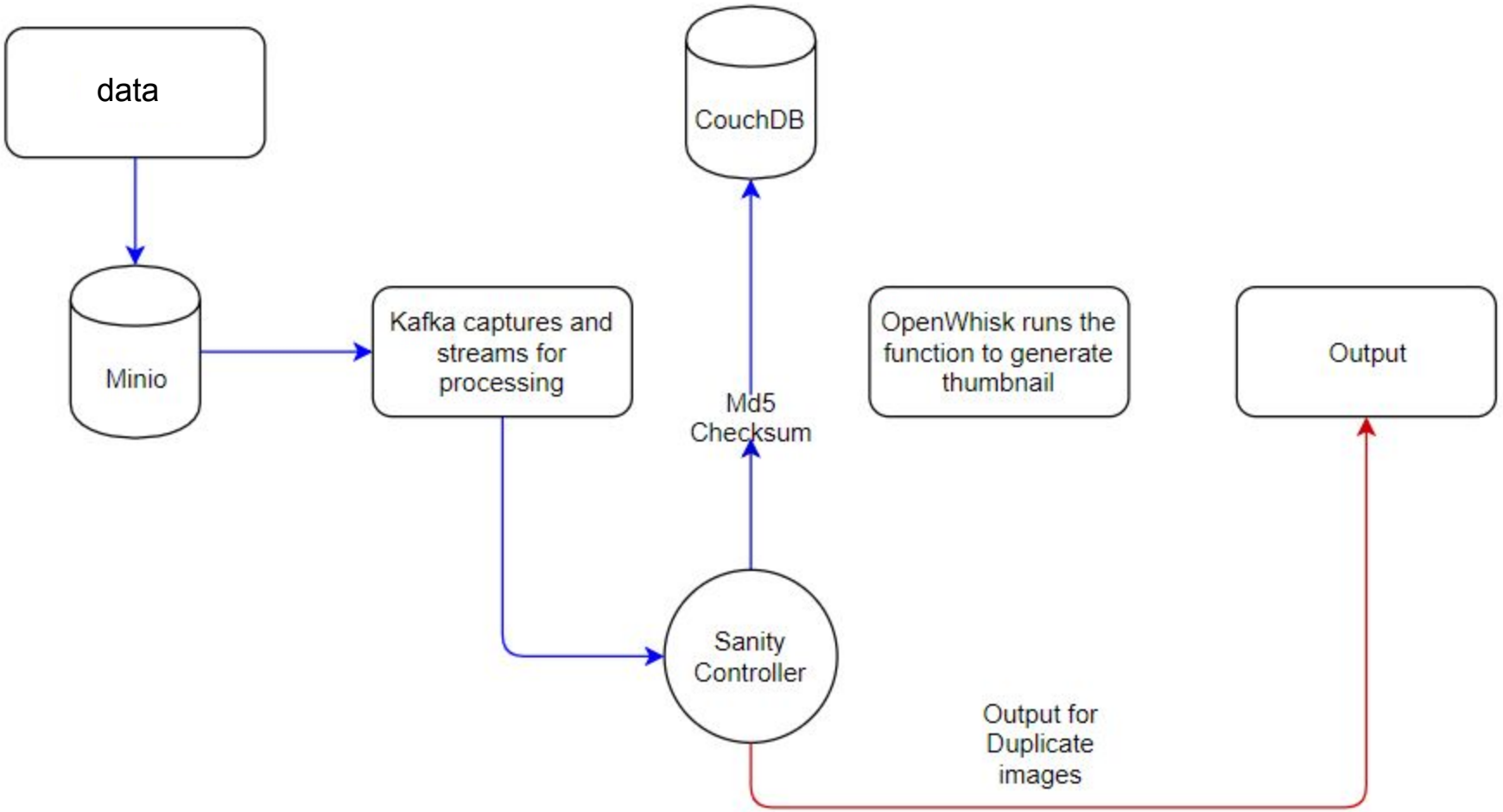

data
*
Minio
*
Kafka captures and streams for processing
*
CouchDB
Md5 Checksum
Sanity Controller
Unique images
OpenWhisk runs the function to generate thumbnail
Output
Output for Duplicate images


data
Minio
Kafka captures and streams for processing
CouchDB
Md5 Checksum
Sanity Controller
OpenWhisk runs the function to generate thumbnail
Output
Unique images
Output for Duplicate images


data
Minio
Kafka captures and streams for processing
CouchDB
Md5 Checksum
Sanity Controller
OpenWhisk runs the function to generate thumbnail
Output
Unique images
Output for Duplicate images


data
Minio
Kafka captures and streams for processing
CouchDB
Output Reference
Md5 Checksum
Sanity Controller
Unique images
OpenWhisk runs the function to generate thumbnail
Output
Output for Duplicate images
Output Data


data
Minio
Kafka captures and streams for processing
CouchDB
Output Reference
Md5 Checksum
OpenWhisk runs the function to generate thumbnail
Output
Unique images
Sanity Controller
Output for Duplicate images
Output Data


data
Minio
Kafka captures and streams for processing
CouchDB
Md5 Checksum
OpenWhisk runs the function to generate thumbnail
Output
Sanity Controller
Output for Duplicate images

What has changed from the Sprint - 4

- Then,
  - Support for **Single User - Multi Use Cases**

    sanity --i <input_bucket> --o <output_bucket> --f <function_name>

- Now,
  - Support for **Multi User - Multi Use Cases**

sanity --i <input_bucket> --o <output_bucket> --f <function_name> --u <user_name>

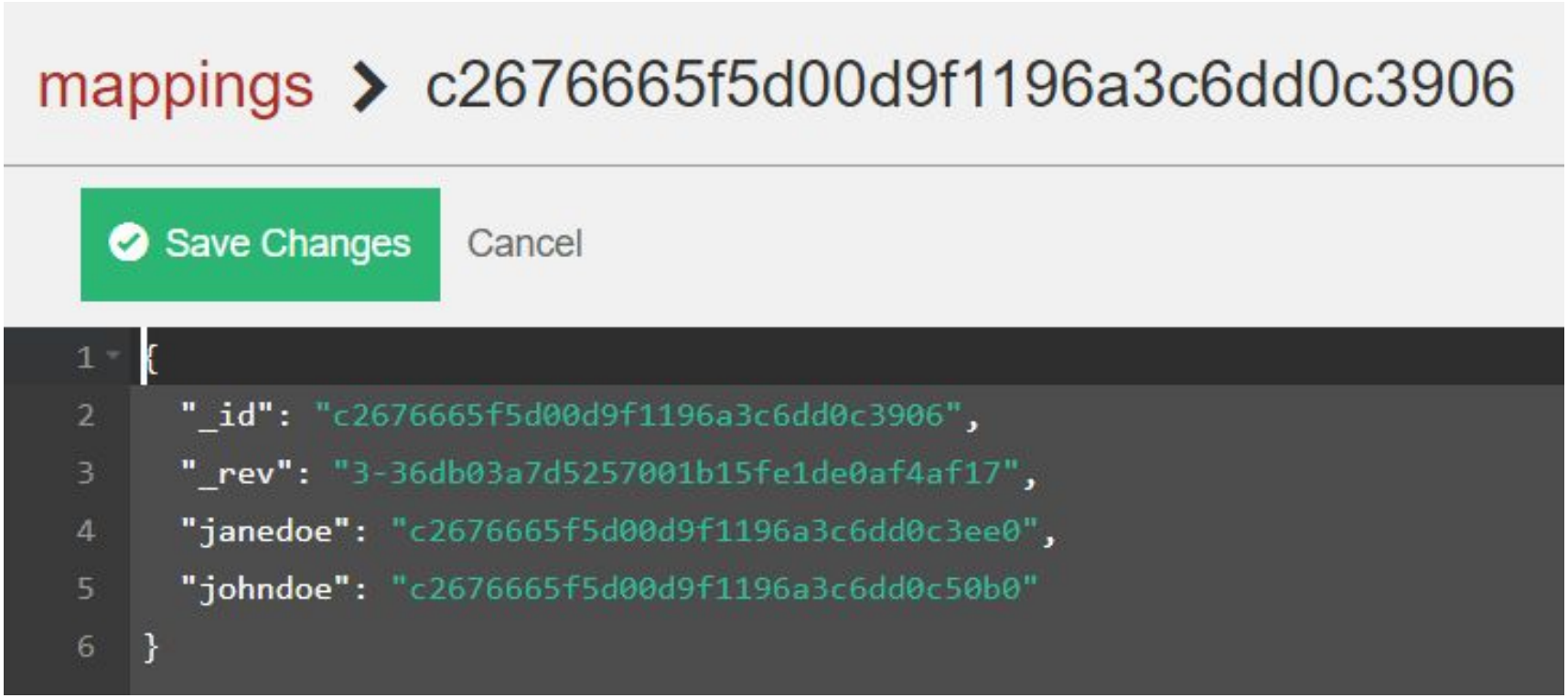
New Schema for CouchDB

2 DBs:

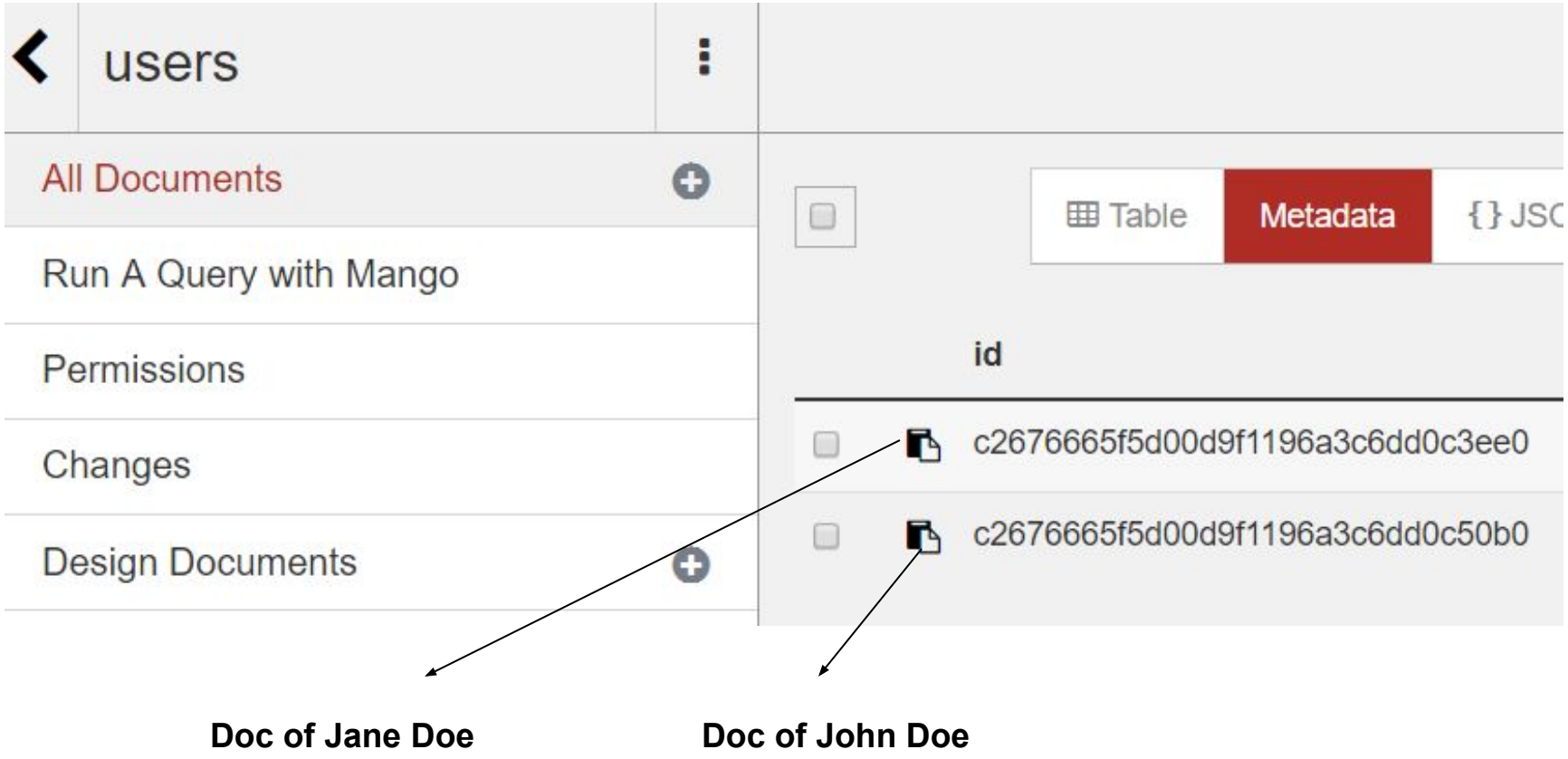
- **Mappings**: Username and document id mappings
- **Users**: User documents

Name	Size	# of Docs
mappings	0.6 KB	1
users	1.9 KB	2

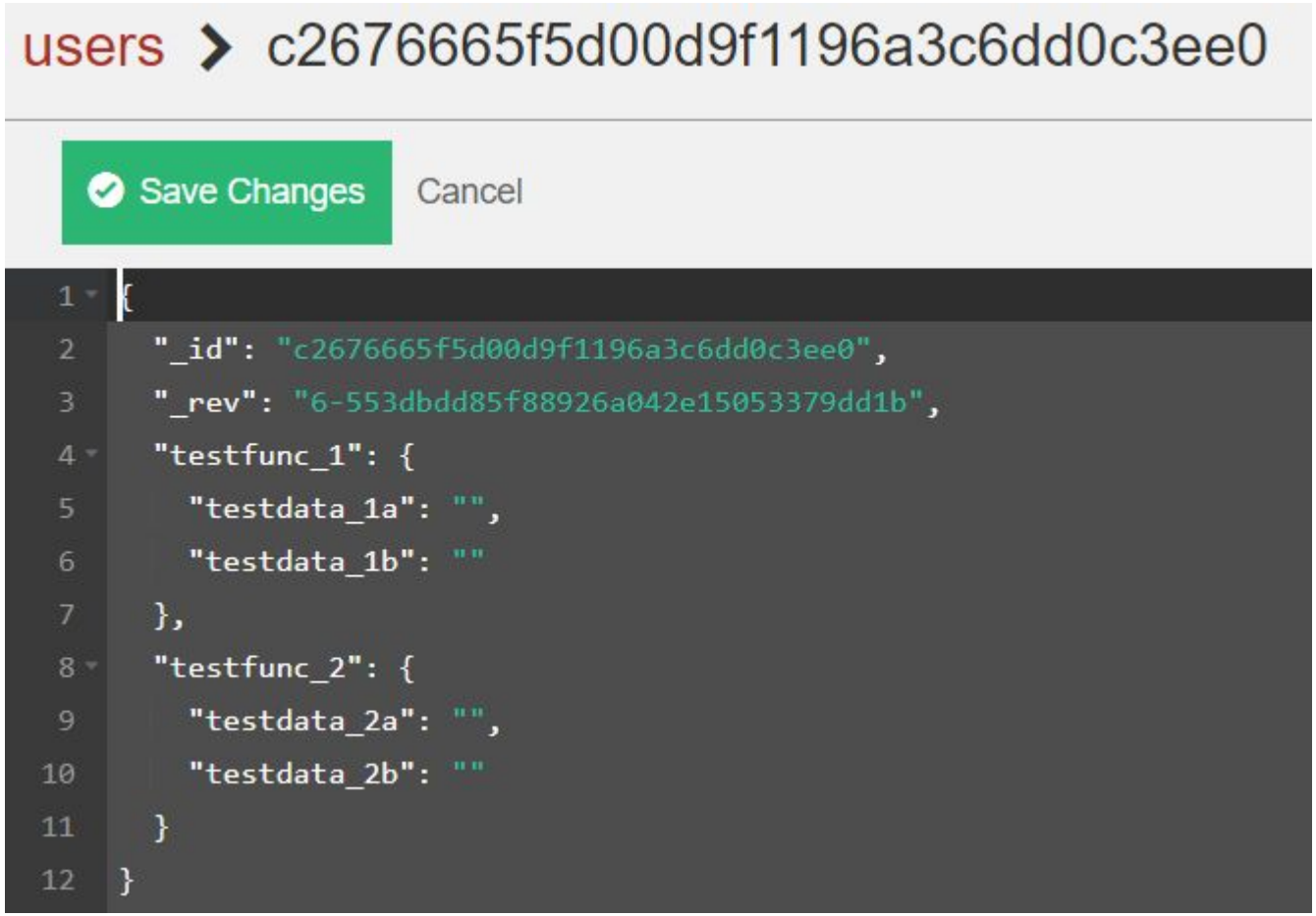
Mappings DB


mappings > c2676665f5d00d9f1196a3c6dd0c3906
Save Changes
Cancel
{
"_id": "c2676665f5d00d9f1196a3c6dd0c3906",
"_rev": "3-36db03a7d5257001b15fe1de0af4af17",
"janedoe": "c2676665f5d00d9f1196a3c6dd0c3ee0",
"johndoe": "c2676665f5d00d9f1196a3c6dd0c50b0"
}

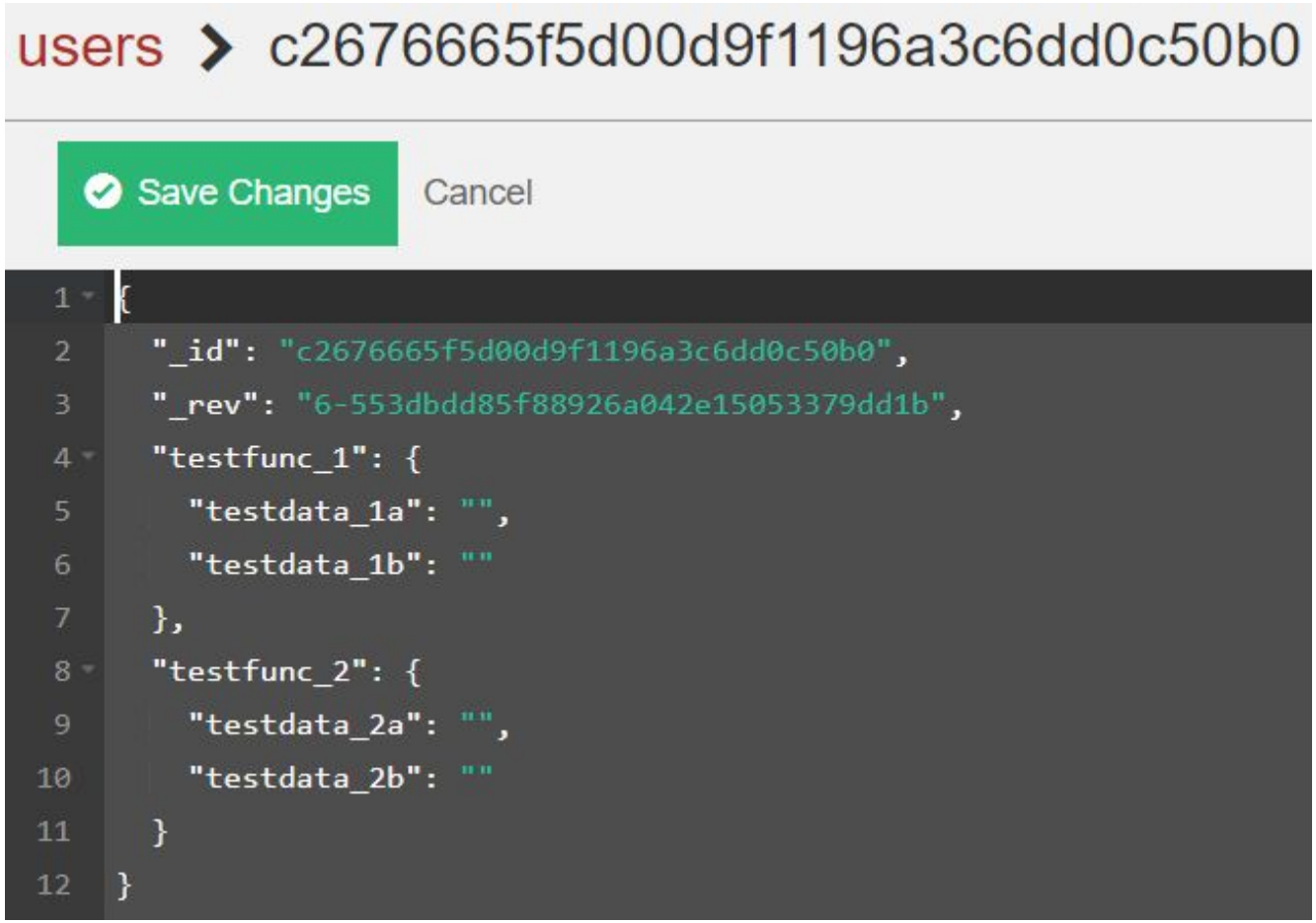
Users DB


users
All Documents
Run A Query with Mango
Permissions
Changes
Design Documents
Table
Metadata
{} JSC
id
c2676665f5d00d9f1196a3c6dd0c3ee0
c2676665f5d00d9f1196a3c6dd0c50b0
Doc of Jane Doe
Doc of John Doe

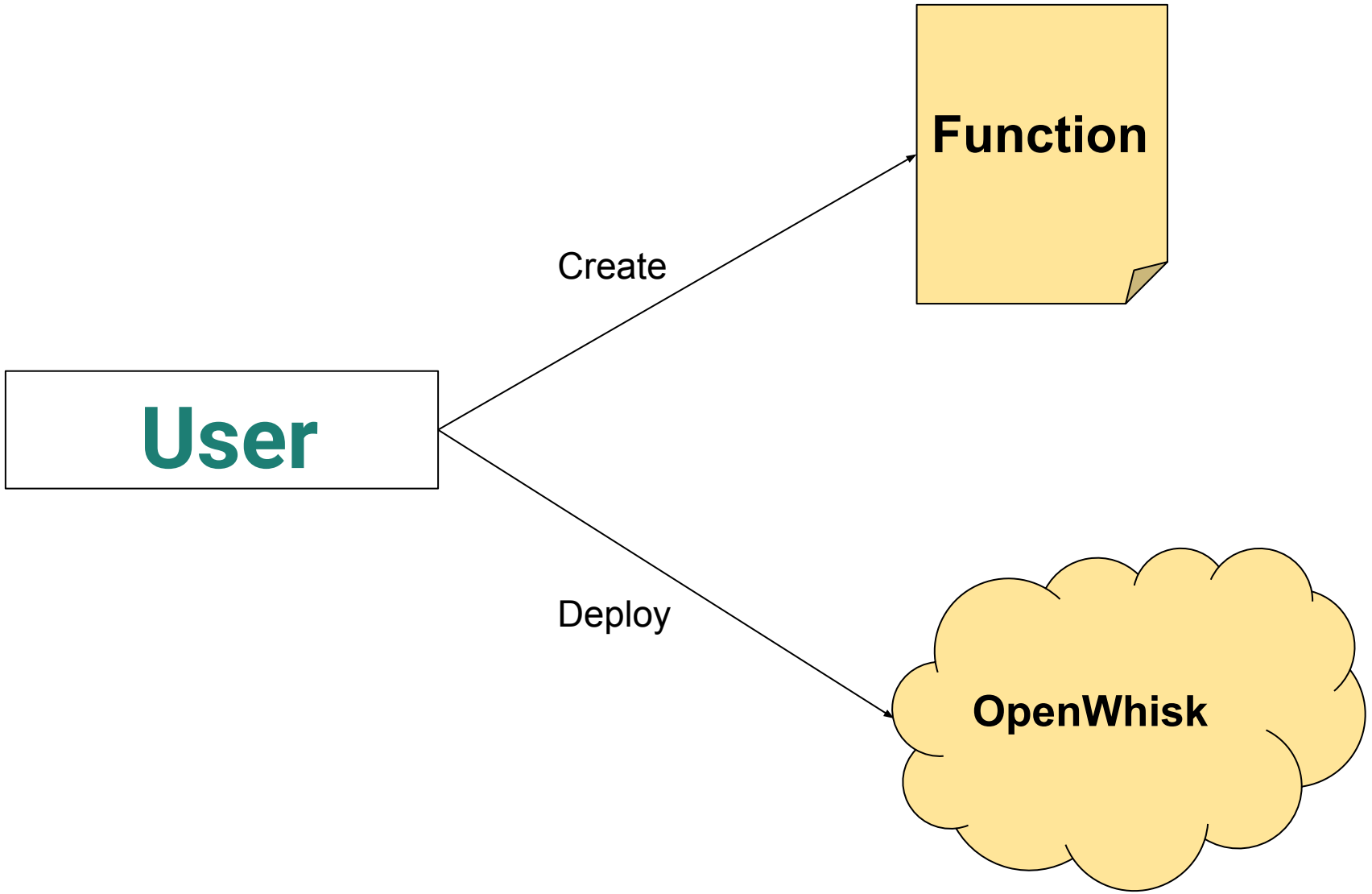
Document of Jane Doe



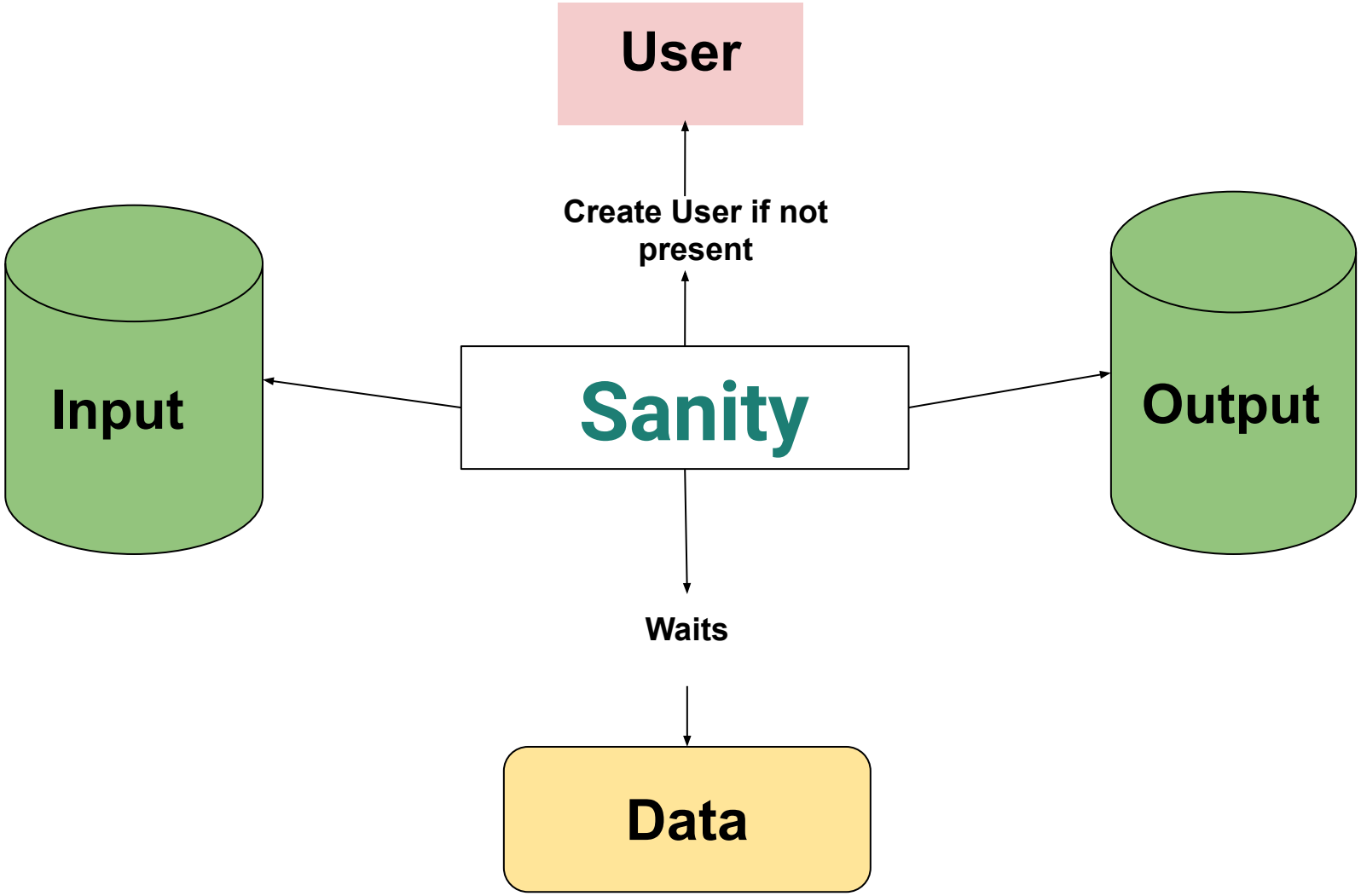
Document of John Doe

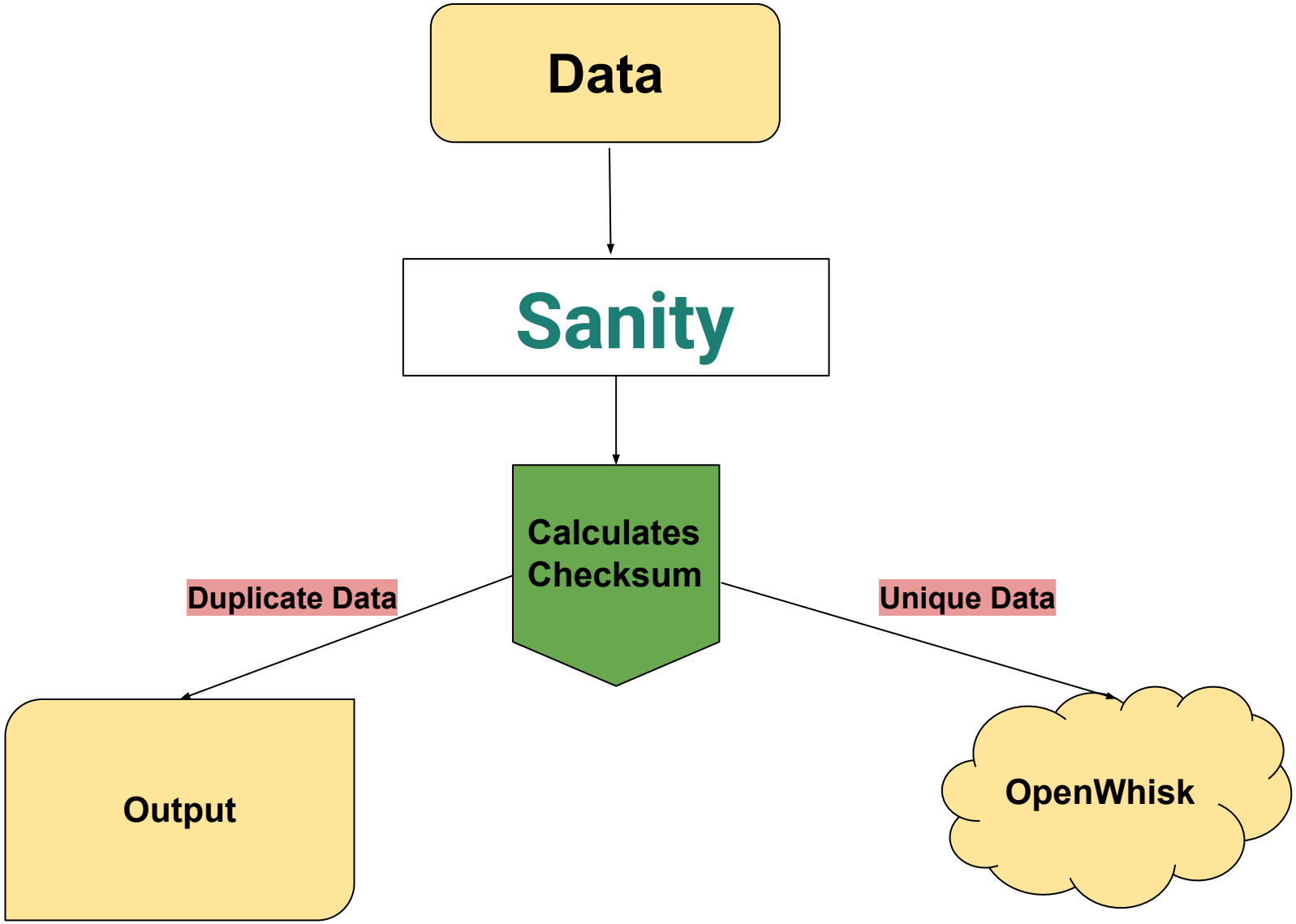


A Quick Recap


Function
Create
User
Deploy
OpenWhisk

sanity --i input --o output --f function --u user




Data
Sanity
Calculates Checksum
Duplicate Data
Unique Data
Output
OpenWhisk

DEMO

What did we achieve in this sprint?

- **Designed and developed** our database to handle multiple users
- **Developed** CLI
- **Demonstrate** how fast de duplication would be(covered in Demo)

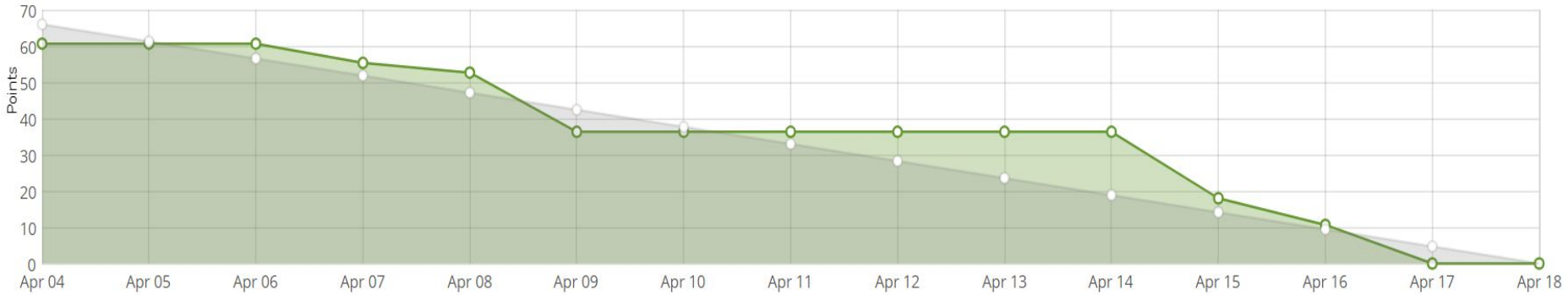
Challenges in Current Sprint

- Integrating CLI with the existing pipeline
- Debugging actions inside the openwhisk
- Building a multi user interface with CLI

Future Scope

- User authentication
- Support for multi threading
- Generalizing sanity to support multiple serverless platforms

Burndown Chart



THANK YOU