Deduplicating Cloud Functions - Sprint 4

Members:

Beliz Kaleli

Vikash Sahu

Paritosh Shirodkar

Asutosh Patra

Mentor:

Shripad Nadgowda

BOSTON UNIVERSITY

Recap

Sprint - 1

- Familiarizing with Serverless Technology

Sprint - 2

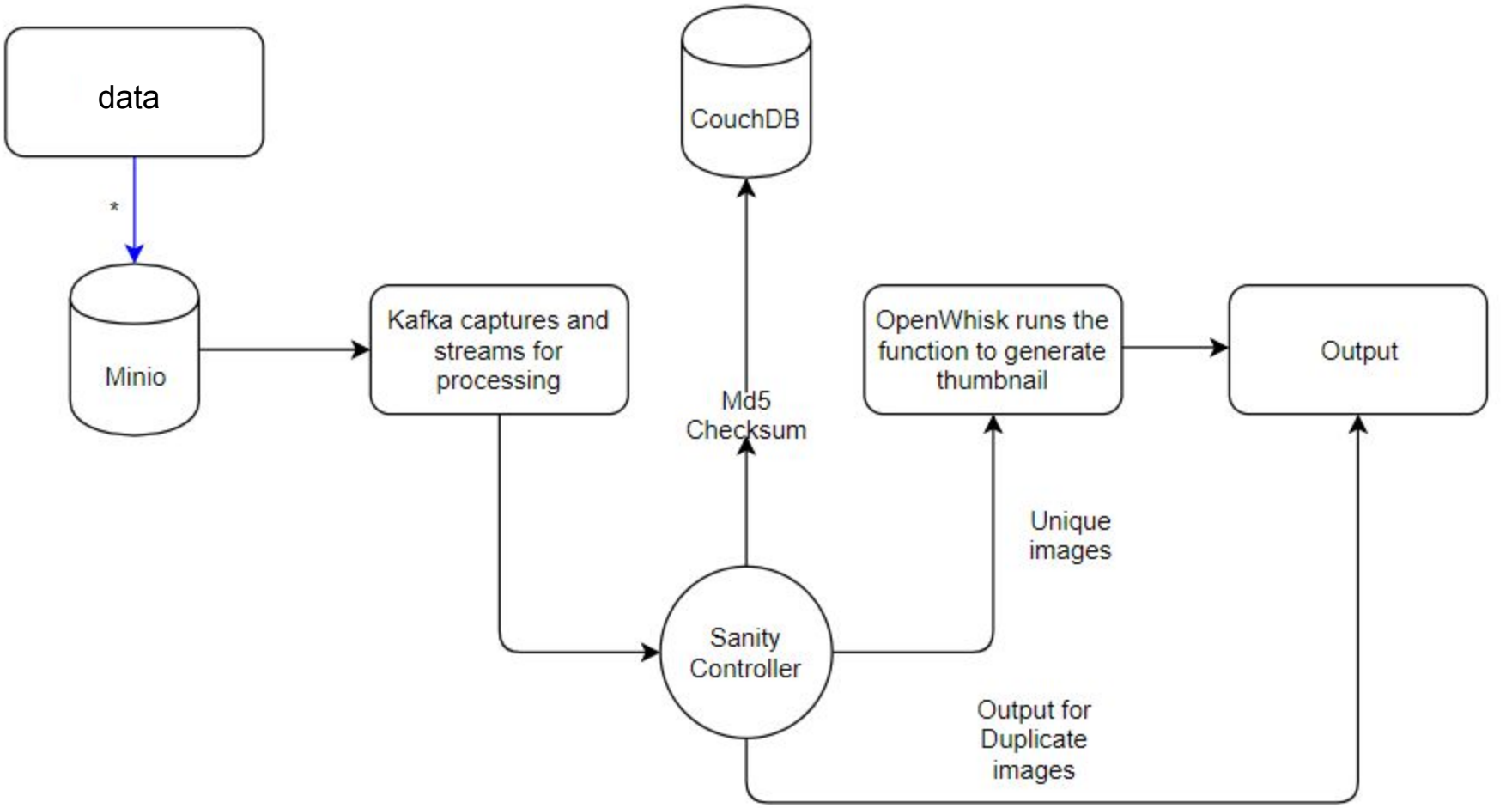
- Setting up all the components (Kafka, Minio, CouchDB and OpenWhisk)

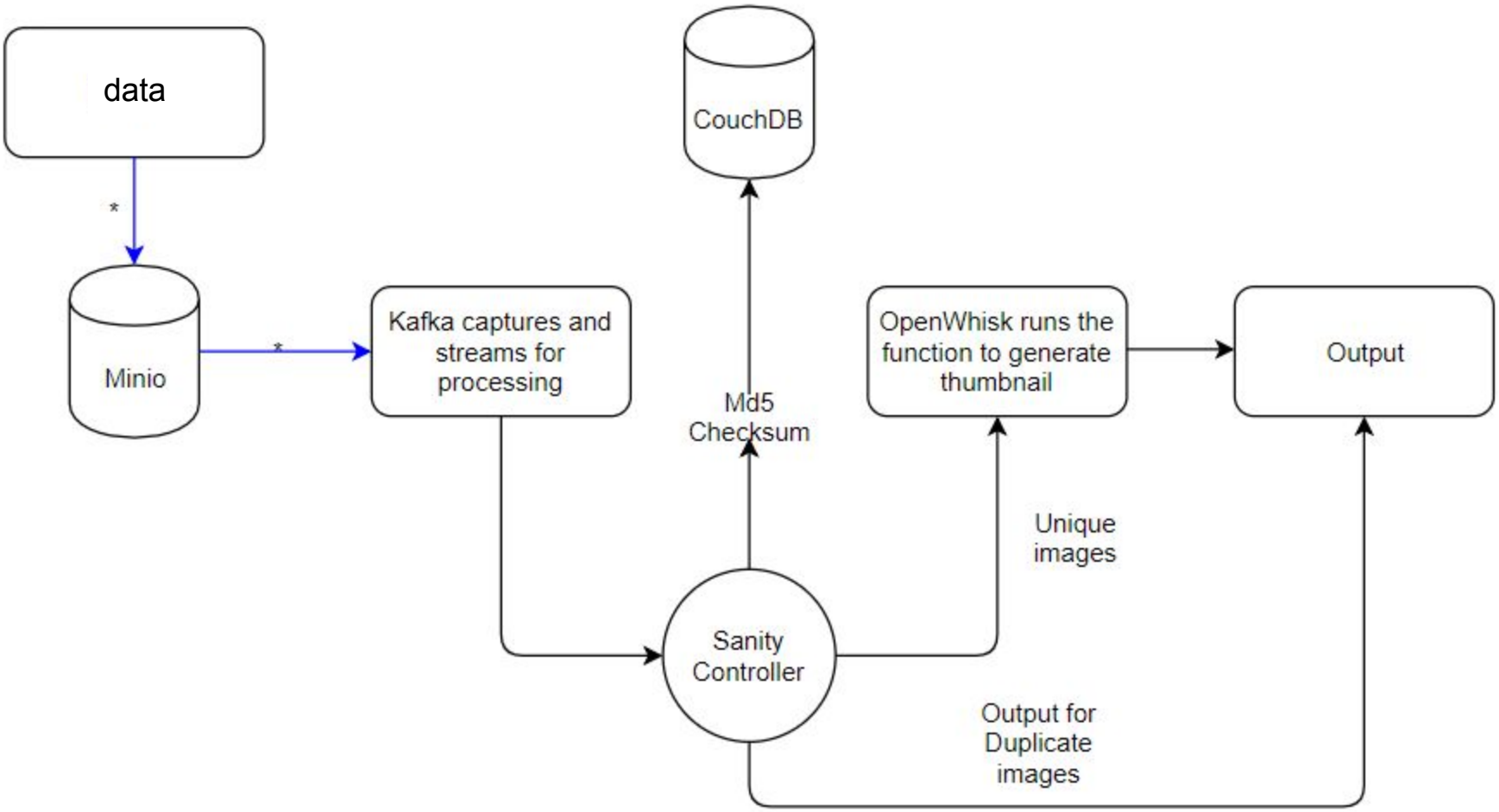
Sprint - 3

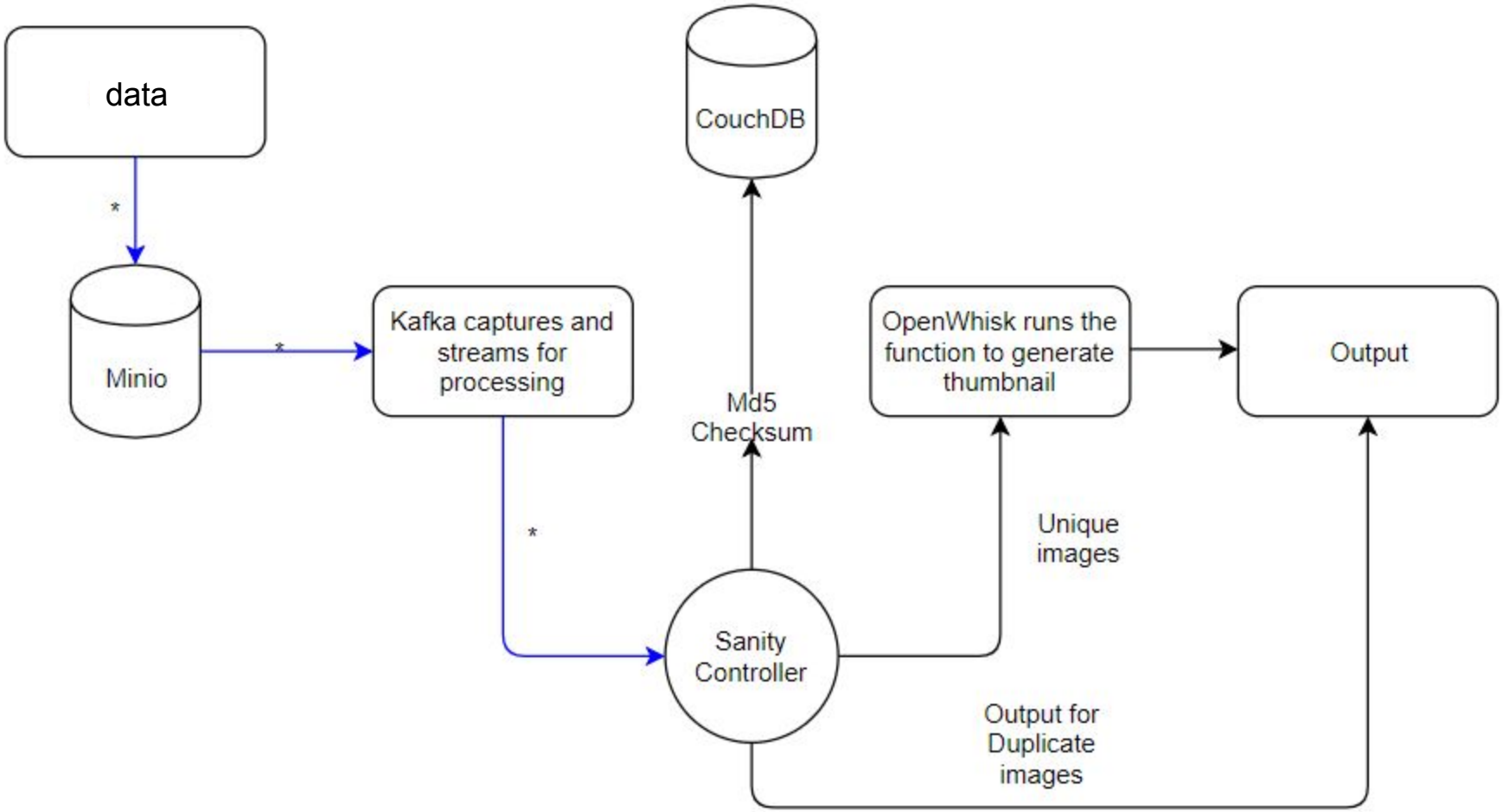
- Developed deduplication framework for Image Thumbnail Use Case in IBM Cloud

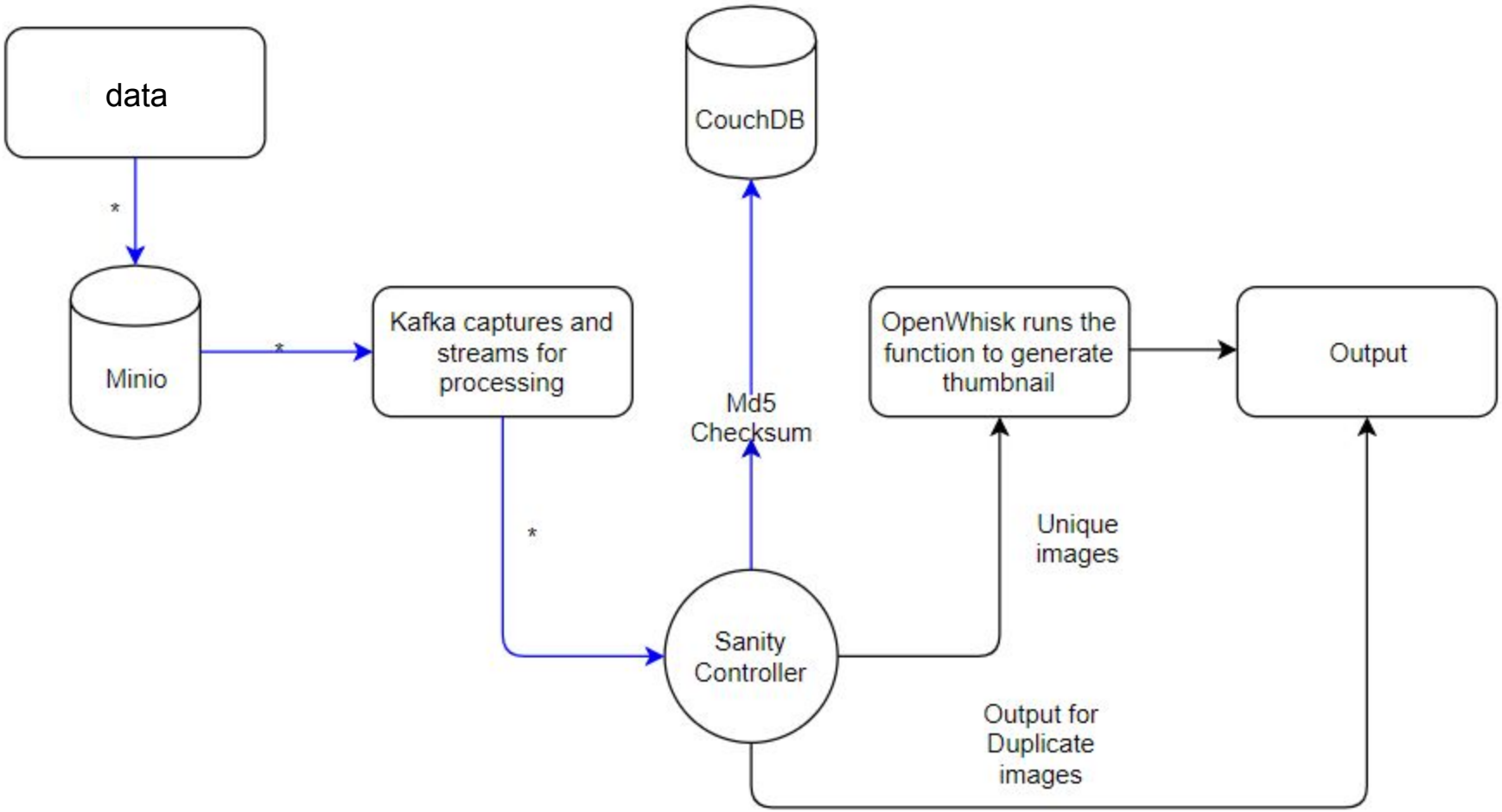
Feedback from the Previous Sprint

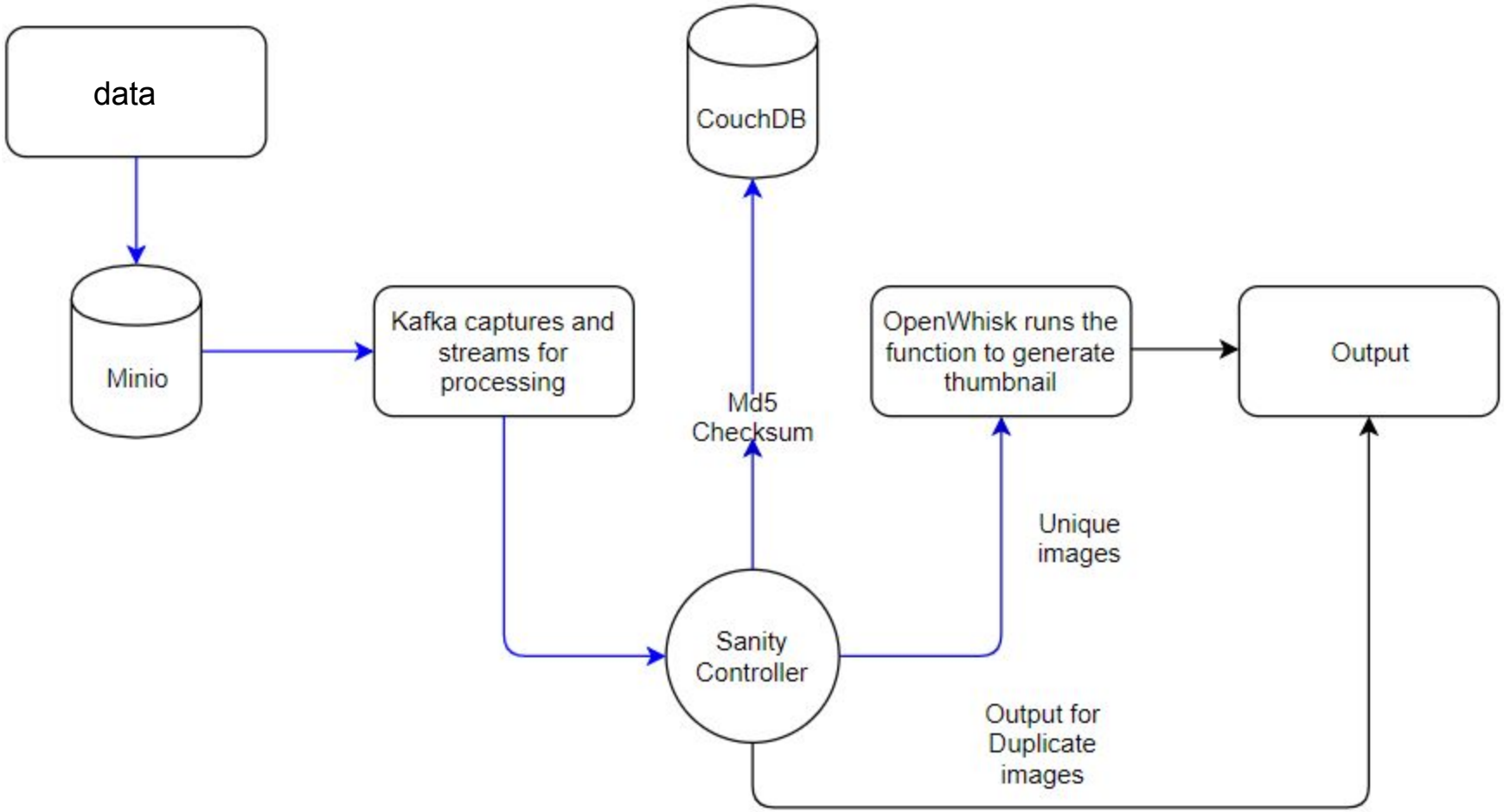
- Generalizing the current architecture
- Benchmarking the performance of Sanity Framework

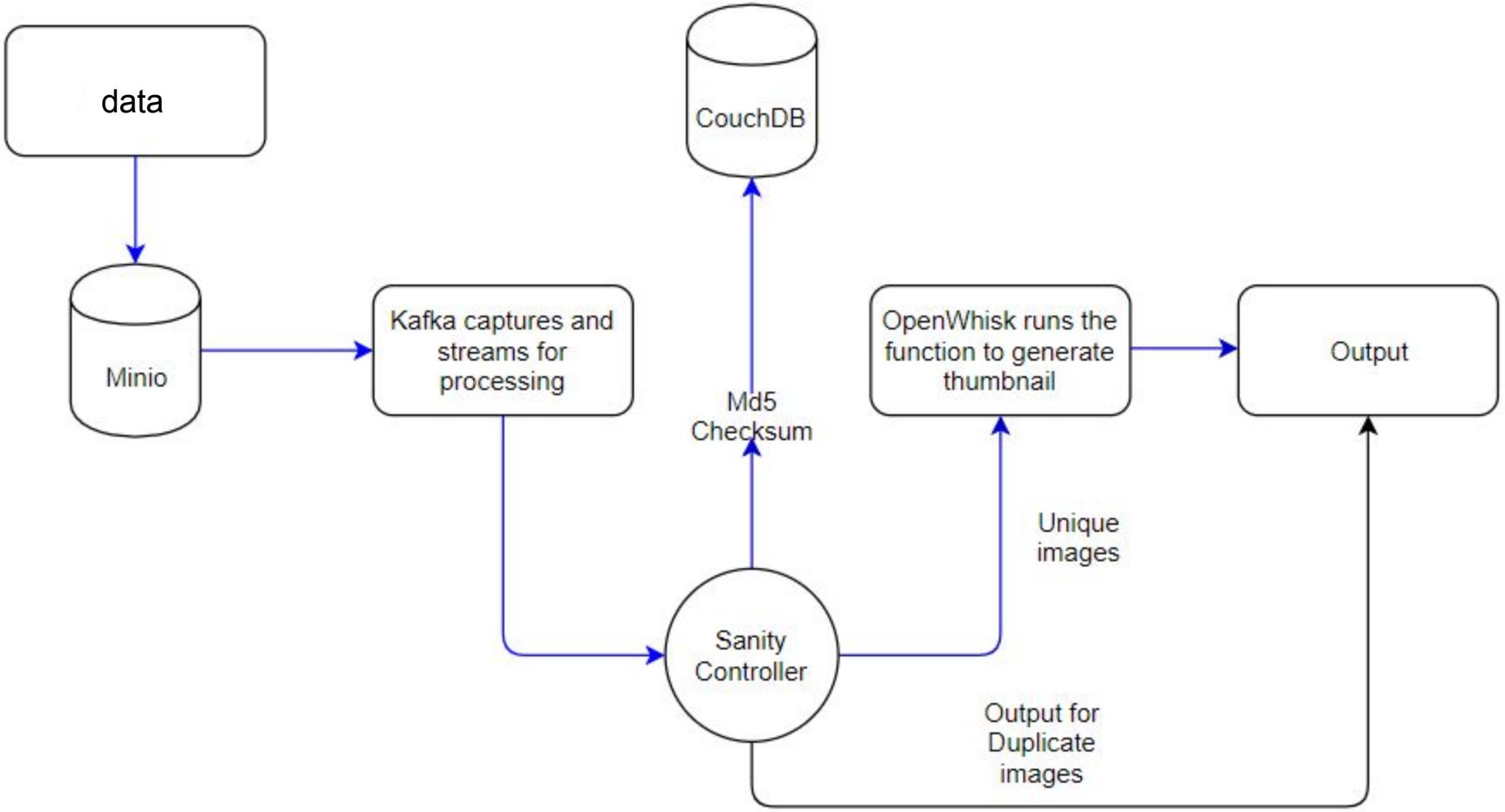

data
*
Minio
Kafka captures and streams for processing
CouchDB
Md5 Checksum
Sanity Controller
Unique images
OpenWhisk runs the function to generate thumbnail
Output
Output for Duplicate images

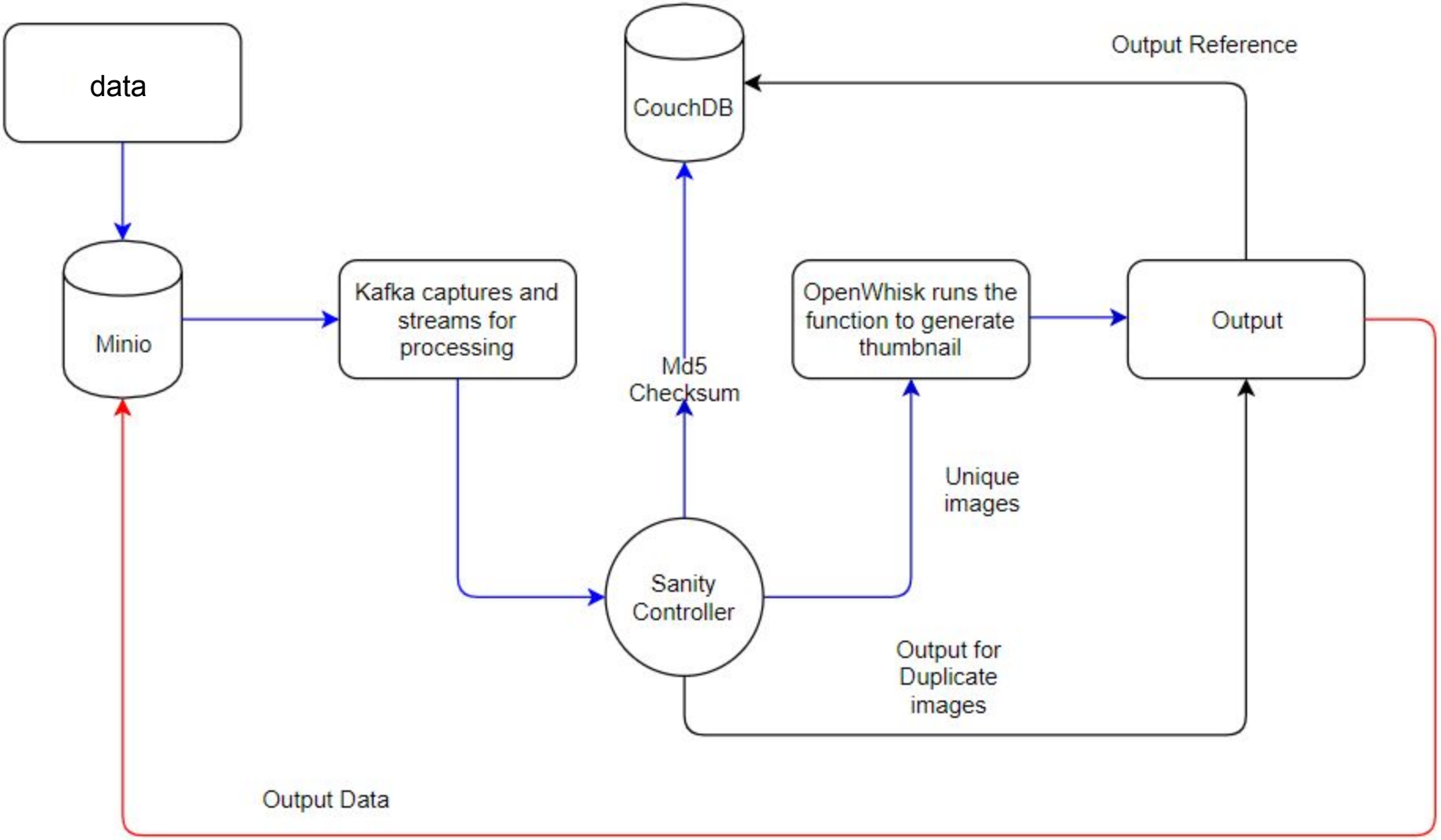

data
*
Minio
*
Kafka captures and streams for processing
CouchDB
Md5 Checksum
Sanity Controller
Unique images
OpenWhisk runs the function to generate thumbnail
Output
Output for Duplicate images

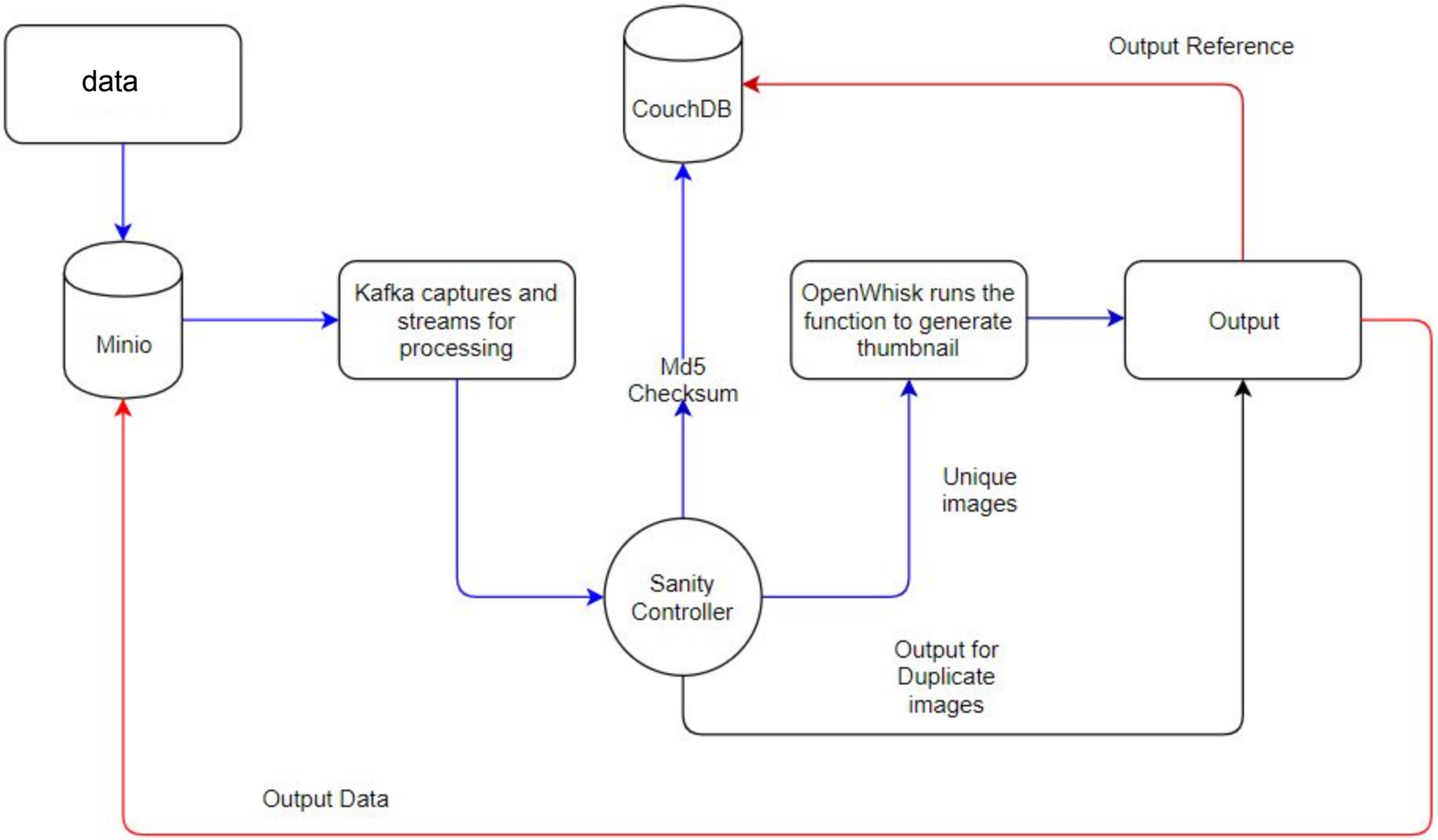

data
*
Minio
*
Kafka captures and streams for processing
*
CouchDB
Md5 Checksum
Sanity Controller
Unique images
OpenWhisk runs the function to generate thumbnail
Output
Output for Duplicate images


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

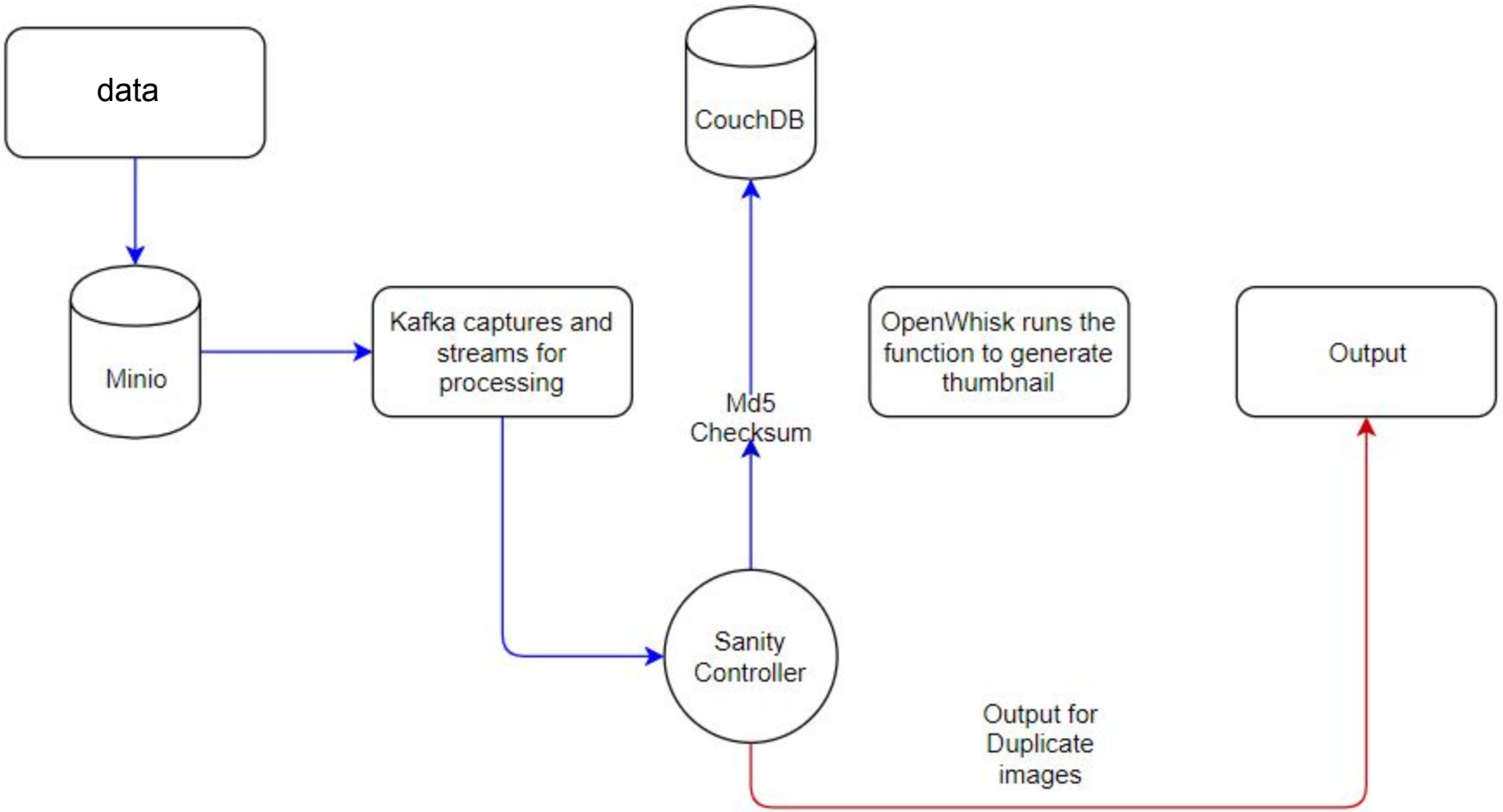

data
Minio
Kafka captures and streams for processing
CouchDB
Output Reference
Md5 Checksum
OpenWhisk runs the function to generate thumbnail
Output
Unique images
Sanity Controller
Output for Duplicate images
Output Data

Couch DB


```
{
  "_id": "dae770ad388c898fa85dc140a0014a24",
  "_rev": "5-2b4a80a884faf4eab778ab5902f5e3ae",
  "function1hash": {
    input1forfunction1hash: "outputbucketfor_input1forfunction1hash/outputfilenamefor_input1forfunction1hash",
    input2forfunction1hash: "outputbucketfor_input2forfunction1hash/outputfilenamefor_input2forfunction1hash"
  },
  "function2hash": {
   input1forfunction2hash: "outputbucketfor_input1forfunction2hash/outputfilenamefor_input1forfunction2hash",
   input2forfunction2hash: "outputbucketfor_input2forfunction2hash/outputfilenamefor_input2forfunction2hash"
  }
}
```


input_checksum

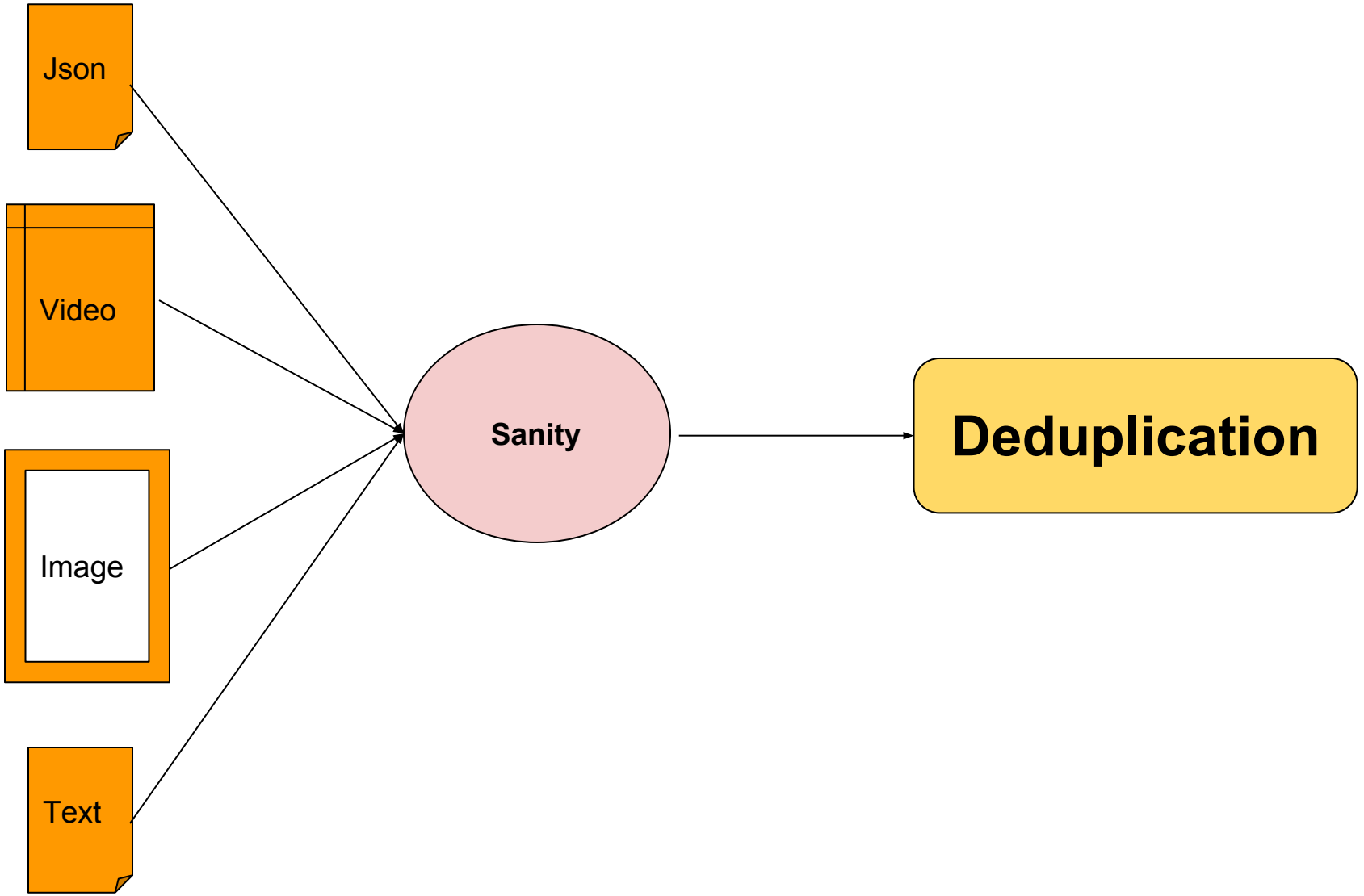
output_bucket/output_file


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 did we achieve in this sprint?

- **Generalizing** our **architecture** for handling any kind of incoming data for deduplication
- **Benchmarking** our **architecture** on performing different use cases with and without deduplication functionality
- **Designing** the CLI

Generalizing Sanity Framework (Using Checksum)



FINDINGS

Benchmarking

- Currently, IBM Cloud charges **0.00002** dollars for single execution of a function in cloud (per sec)
- Imagine, if we try to invoke the same functions across with same data hitting **100,000** times, it would cost around 2 dollars for a single use case and if the cluster which has multiple functions scales up, cost would go too much.
- Sanity would help save this execution multiple times and also potentially a lot of money.

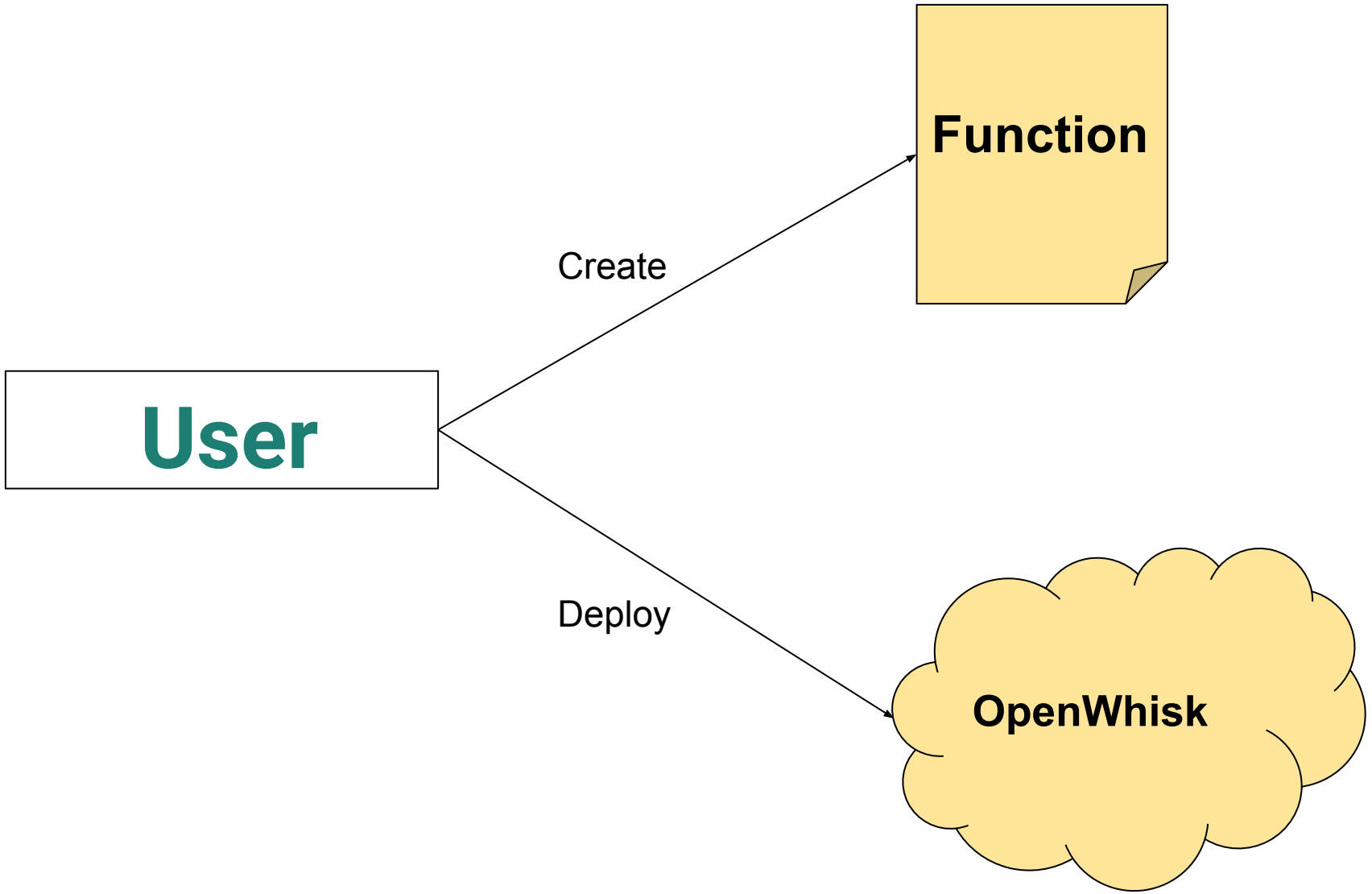
Benchmarking *Contd.*

- In **Word count example**, the unique function takes around 0.69 sec
- In **Video compression**, the unique function takes around 2 sec.
- In **Weather api example** , the unique function takes around 0.5 sec.

Next Steps (Sprint - 5)

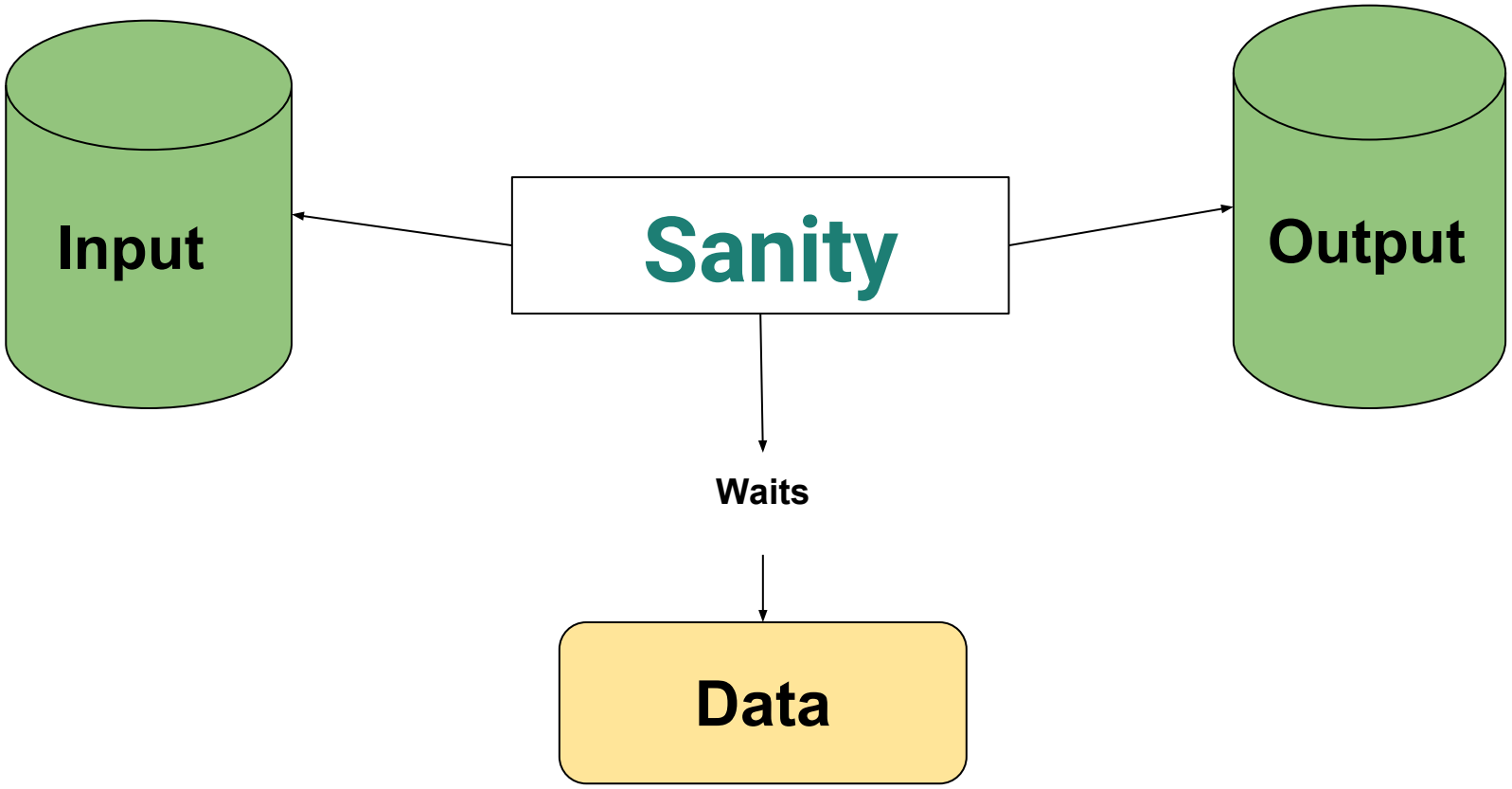
- Integrating the CLI with Sanity Framework

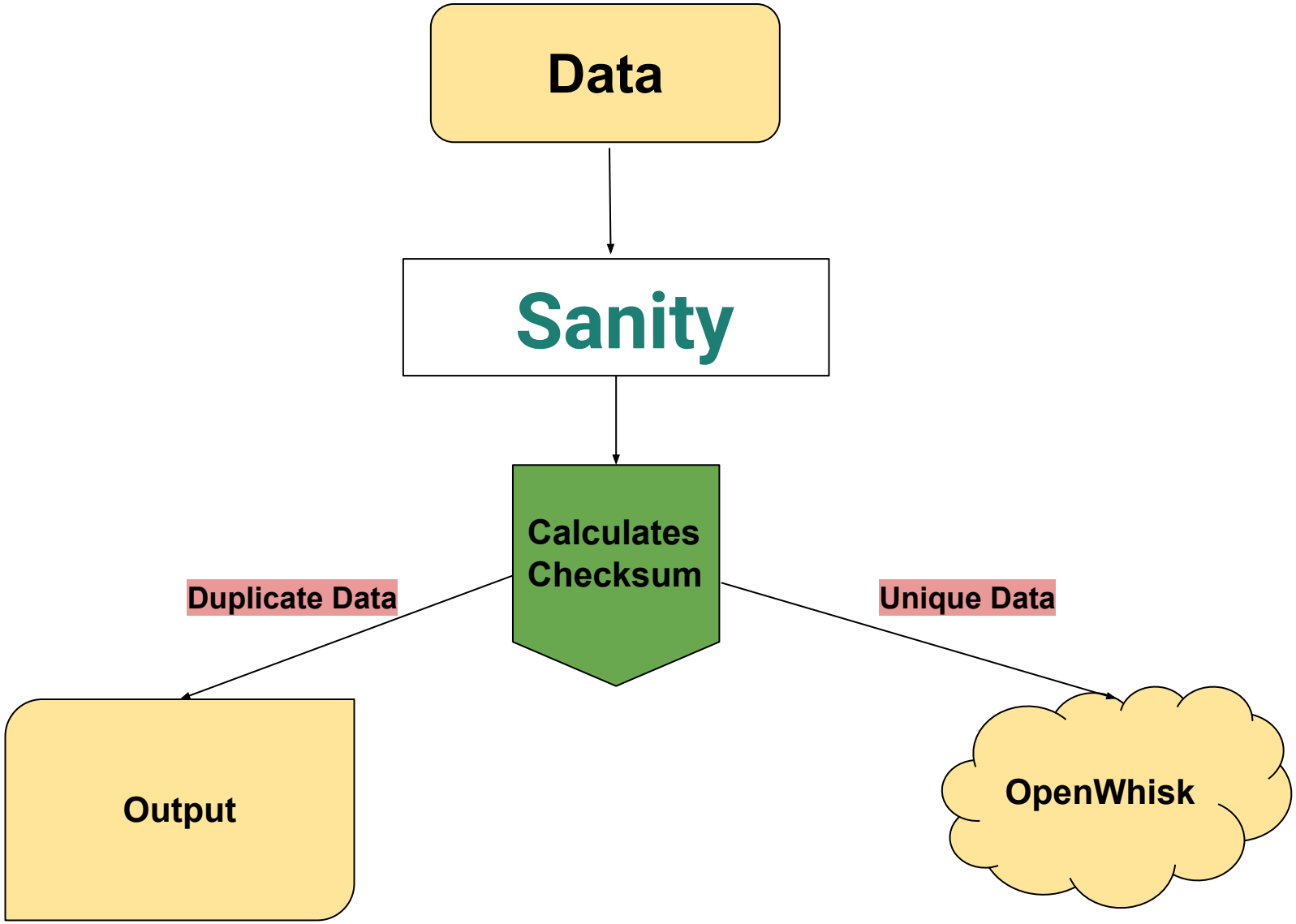
sanity --i <input_bucket> --o <output_bucket> --f <function_name>


Function
Create
User
Deploy
OpenWhisk


User
sanity --i input --o output --f function

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




Data
Sanity
Calculates Checksum
Duplicate Data
Unique Data
Output
OpenWhisk

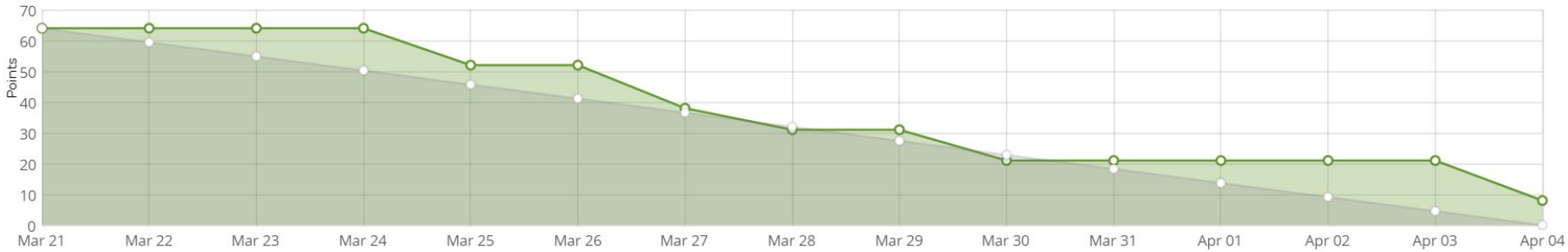
Next Steps (Sprint - 5) *Contd.*

- Explore other benchmarking methods and compare with the existing framework
- Brainstorm with mentor regarding the stretch goals
  - Detect the outputs for an arbitrary function

Challenges in Current Sprint

- Configuring custom docker components for executing functions in them
- How can we invoke custom docker scripts in openwhisk using parameterized input?
- Debugging actions inside the openwhisk

Burndown Chart



THANK YOU