memphis.dev

Streaming Systems for Real-Time Analytics

RJ Nowling

Memphis.dev

MadPy Meetup

August 10, 2023

memphis.dev

Introducing Myself

- Developer Advocate at Memphis.dev
- Associate Professor of Computer Science at the Milwaukee School of Engineering (5 years)
- ML engineer in industry (4 years)

memphis.dev

BigPetStore

memphis.dev

Simulated Transactions

```
{
  "transaction_id": 1107316,
  "customer_id": 69770,
  "timestamp": "2023-01-01T05:59:40.799777",
  "line_items": [
    { "item_id": 10, "quantity": 1 },
    { "item_id": 5, "quantity": 2 }
  ]
}
```

1.5 million transactions

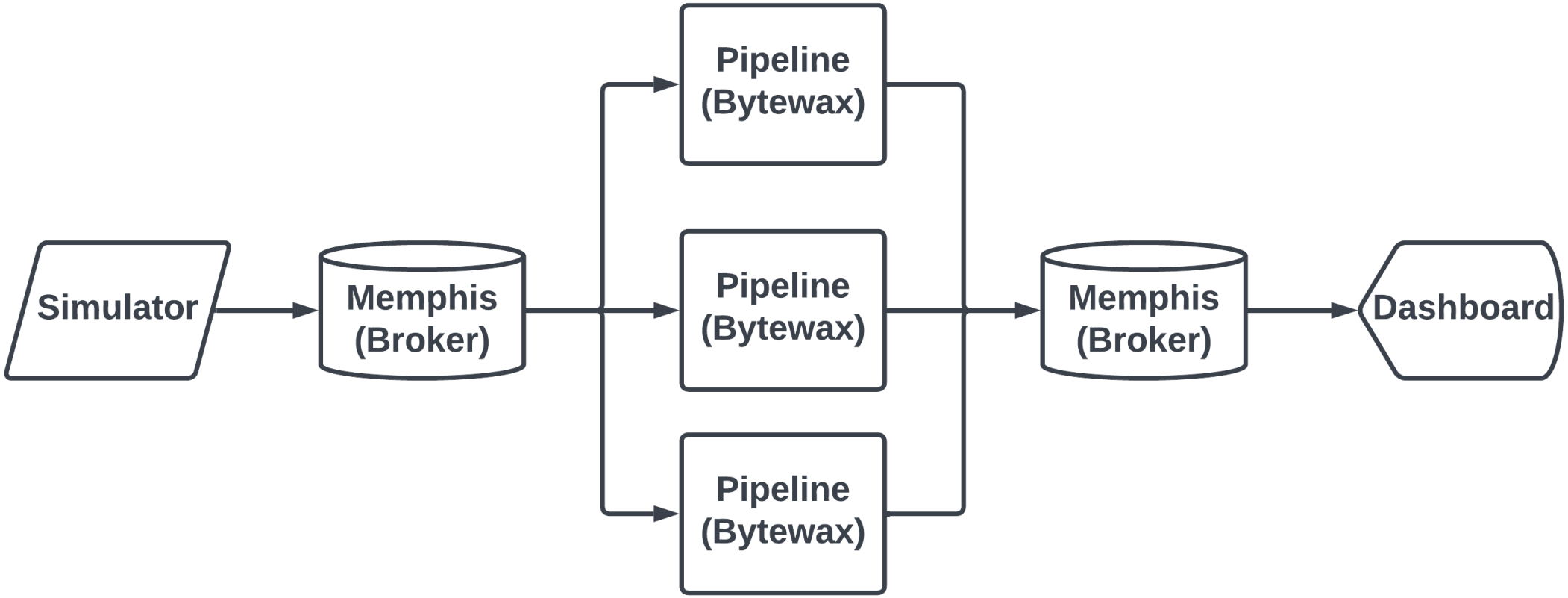
1 year of time

Analyses

- Weekly transaction volume (tumbling window)
- Week-over-week change in transaction volume (sliding window)
- Week-over-week change in product sales volume (sliding window, group by key)

memphis.dev

Architecture

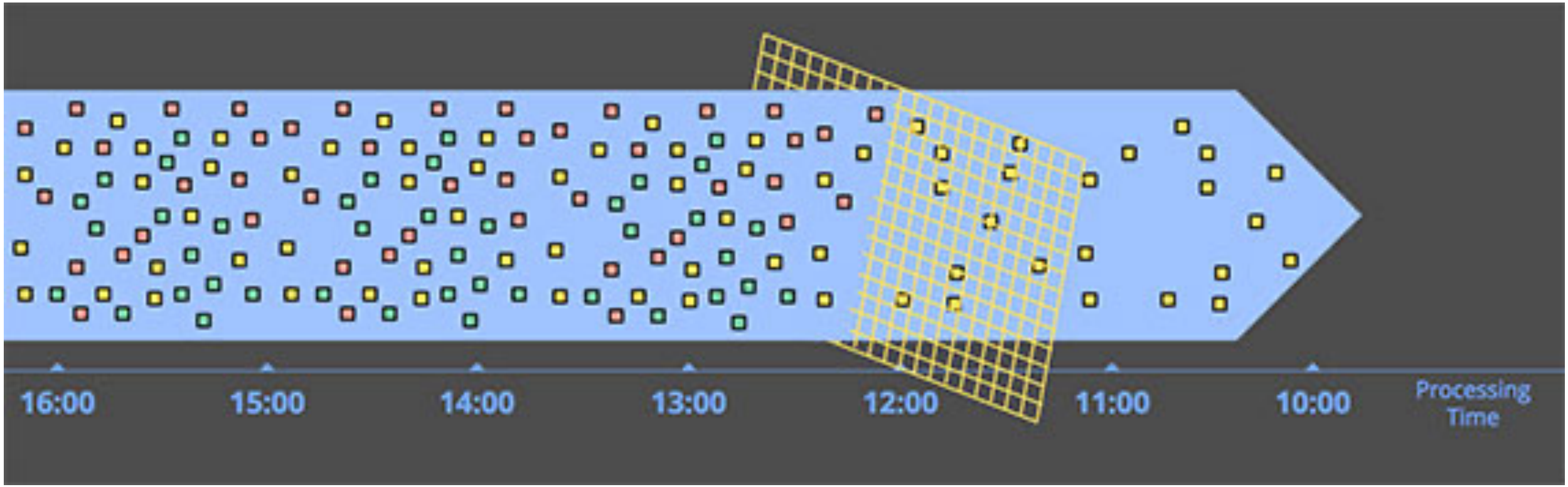


Agenda

- Introduce message brokers
- Introduce stream processing engines
- Demo our transaction analysis system

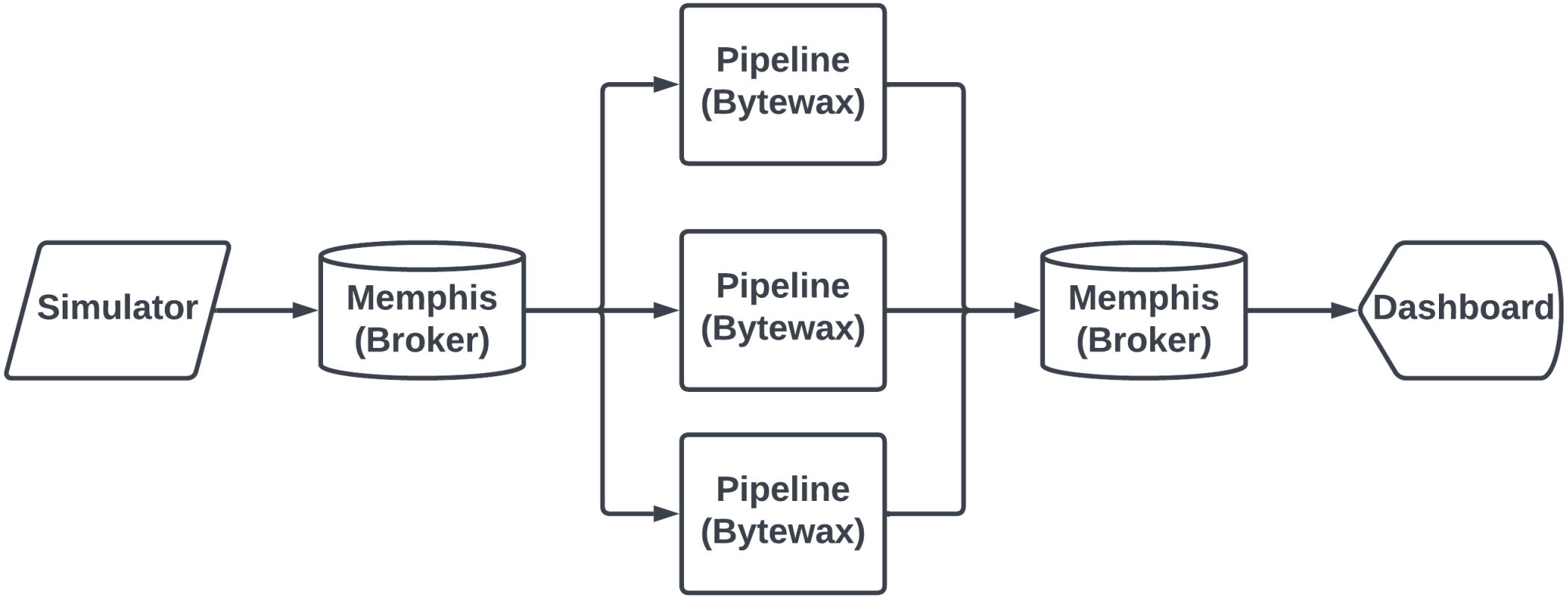
Message Brokers

Streaming Data



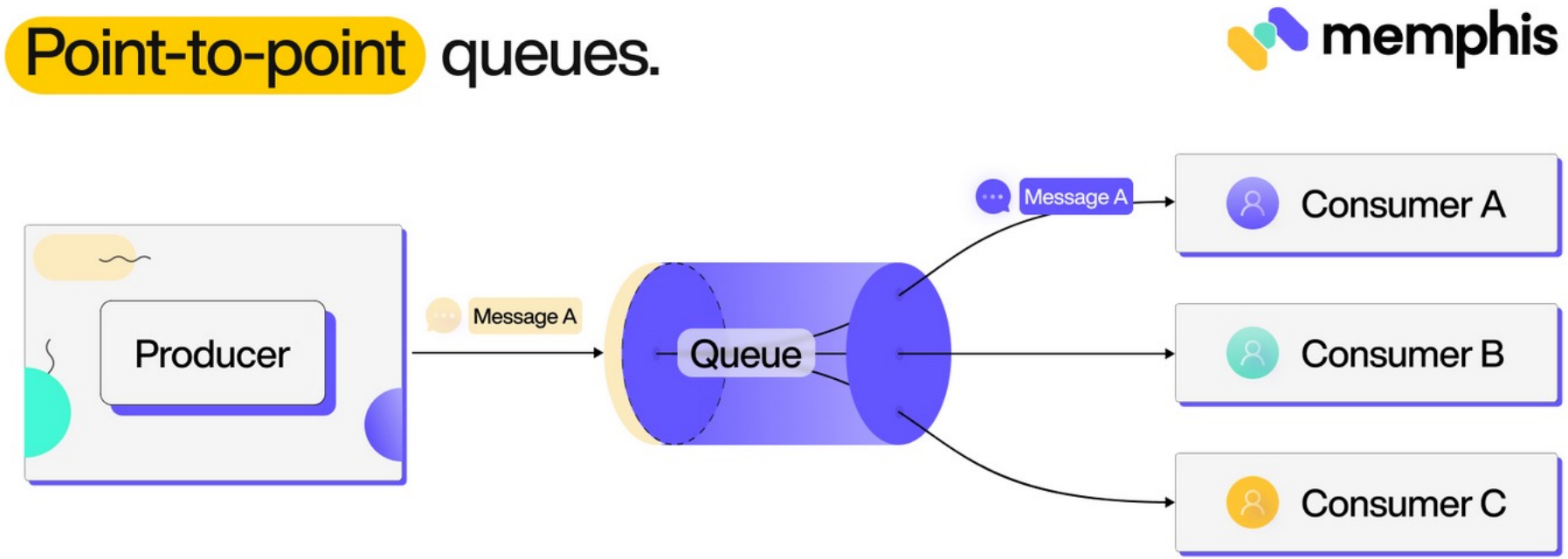
memphis.dev

Architecture



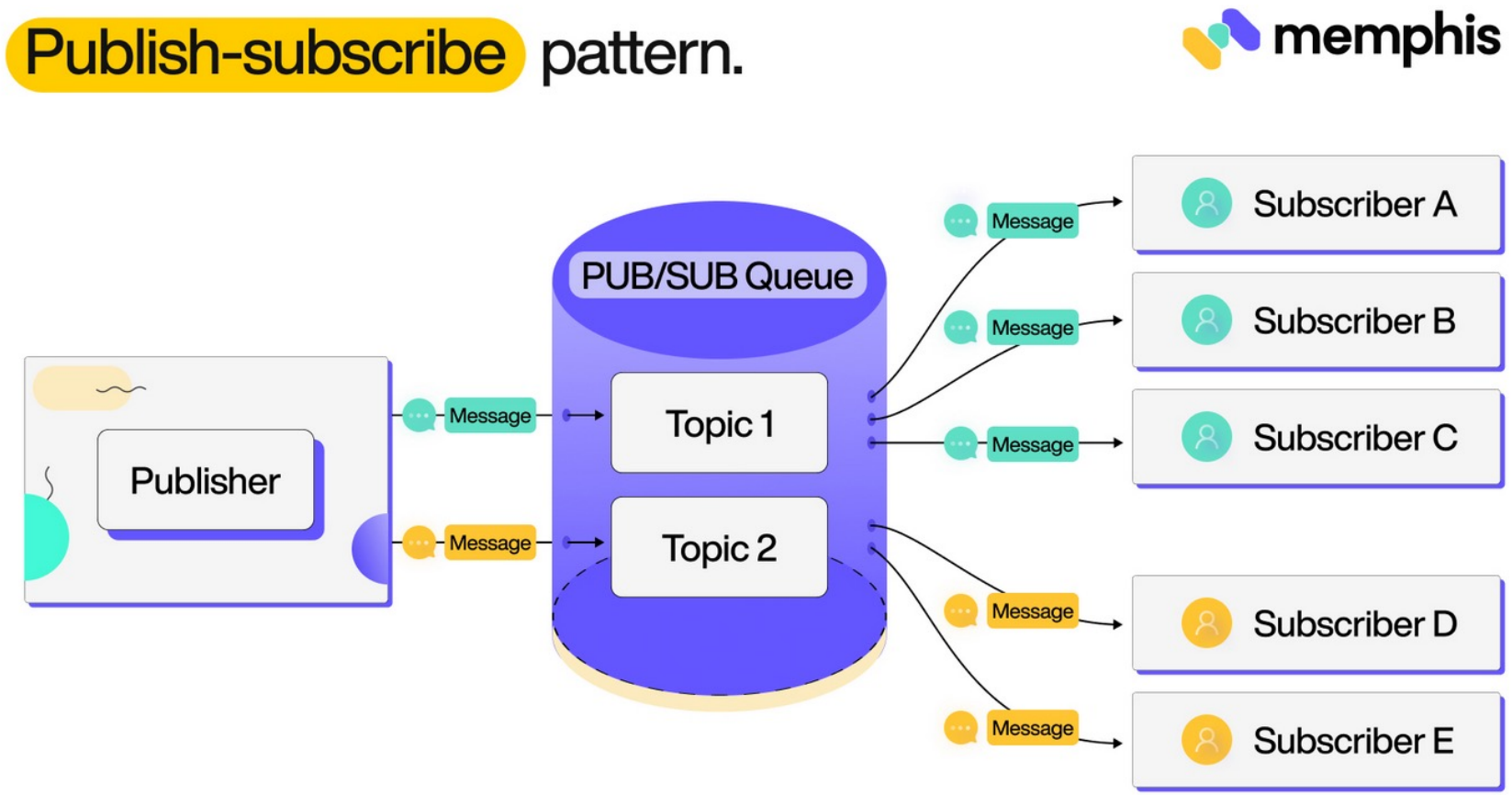
memphis.dev

Message Queue Pattern



memphis.dev

Publish / Subscribe Pattern



Architectural Advantages

- Decouples routing
- Decouples availability
- Enables asynchronous communication
- Rate limiting / buffering

Producer Code

```
memphis = Memphis()
await memphis.connect(host=HOST, username=USERNAME,
                      password=PASSWORD)
producer = await memphis.producer(
                station_name=STATION,
                producer_name="test-producer")
msg_id = 0
while True:
    msg = f"This is test message {msg_id}."
    await producer.produce(bytearray(msg, "utf-8"))
```

Consumer Code

```
consumer = await memphis.consumer(
                station_name=STATION,
                consumer_name="test-consumer")
while True:
    batch = await consumer.fetch()
    for msg in batch:
        s = msg.get_data().decode("utf-8")
        print(s)
        msg.ack()
```

memphis.dev

todo-cdc-events

+ Add new tag

Purge

Created by root at Apr 16, 2023, 5:55 PM

Retention: 7 days
Replicas: 1
Local Storage: disk
Remote Storage: —

Schema
+ Attach schema

Total messages
8

Av. message size
187 Bytes

Code examples
View details >

Audit
View details >

Producers (1)

Name	User	Status
test-producer	todocdcse...	

Details

Name	test-producer
User	todocdcservice
IP	192.168.64.1:42380

Station

Showing last 8 out of 8 messages

Drop

Messages | Dead-letter | Details

Messages	Information
This is a test me...	
This is a test me...	
This is a test me...	
This is a test me...	
This is a test me...	
This is a test me...	
This is a test me...	
This is a test me...	

No message selected

Consumer groups (1)

Name	Unacked	Unprocessed	Status
test-consu...	0	0	

Details

Unacked messages	0
Unprocessed messages	0
In process message	0
Max ack time	30,000ms
Max message deliveries	10

Consumers 1

"Classic" Message Brokers

- Apache ActiveMQ
- Apache RocketMQ
- BlazingMQ
- IBM MQ
- NATS
- Oracle Advanced Queuing
- RabbitMQ

memphis.dev

"Modern" Message Brokers

- Apache Kafka
- Apache Pulsar
- Memphis
- NATS Jetstream
- Redpanda

Architectural Features

- Partitioning: enables parallelism
- Message retention: turns message broker into a database
- Consumer can rewind: message replay (time travel)

Stream Processing Engines

Stream Processing Engines

- Provide a programming model
- "Automatic" parallelization
- State management
- Recovery

- Apache Flink
- Apache Samza
- Apache Spark
- Apache Storm
- Bytewax
- Faust

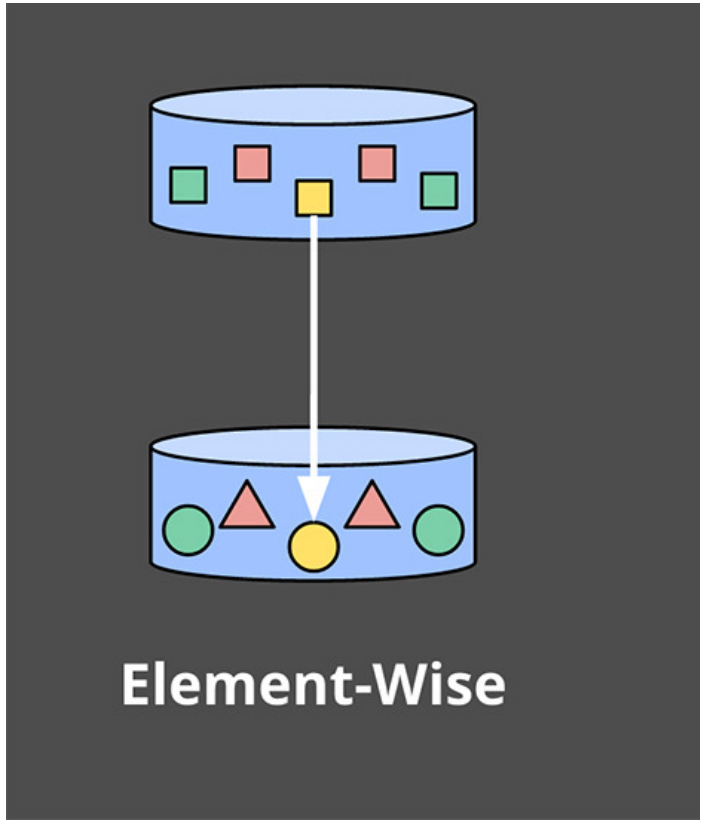
memphis.dev

bytewax

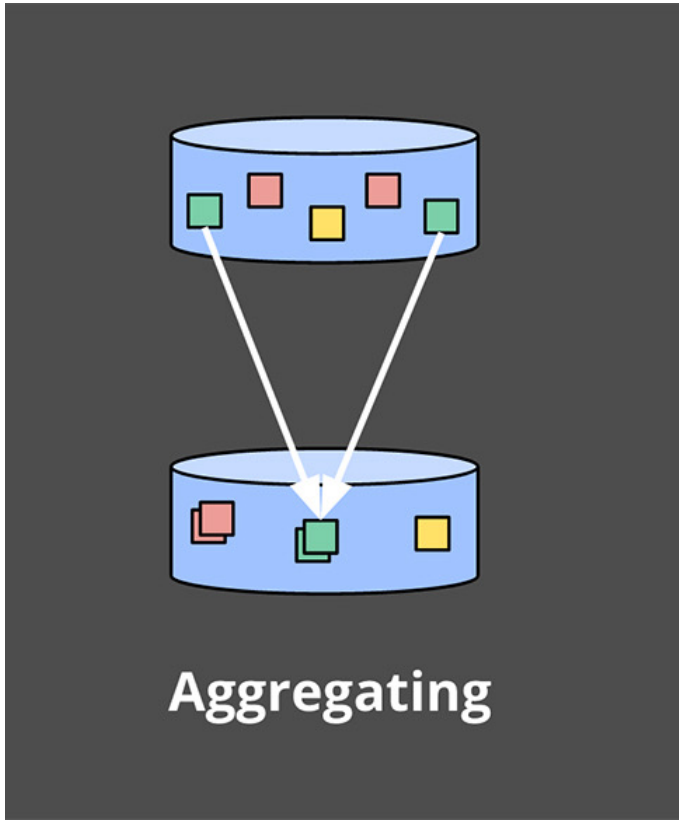
```
flow = Dataflow()
flow.input("memphis-consumer", memphis_src)

# bytearray to UTF-8 string
flow.map(lambda m: m.decode("utf-8"))

# deserialize JSON document
flow.map(json.loads)

# grab customer id
flow.map(lambda r: r["customer_id"])

# convert to bytearray
flow.map(lambda s: bytearray(s, "utf-8"))

# print customer id
flow.output("console-output", StdOutput())
```

Transformations

Map

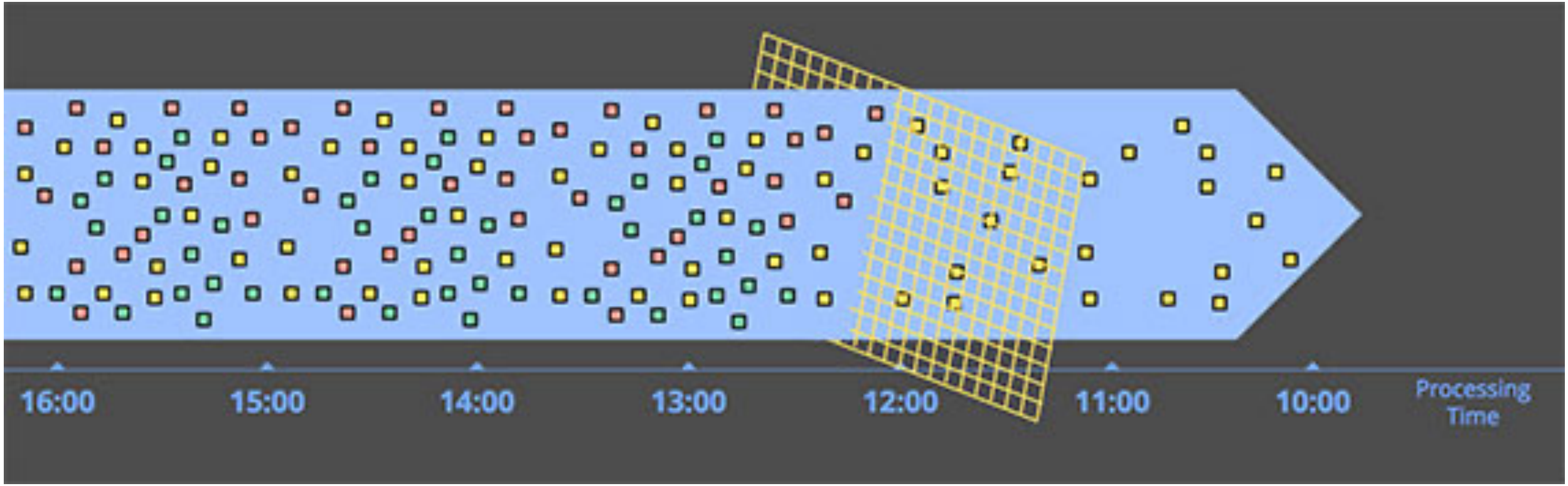


Reduce / Fold

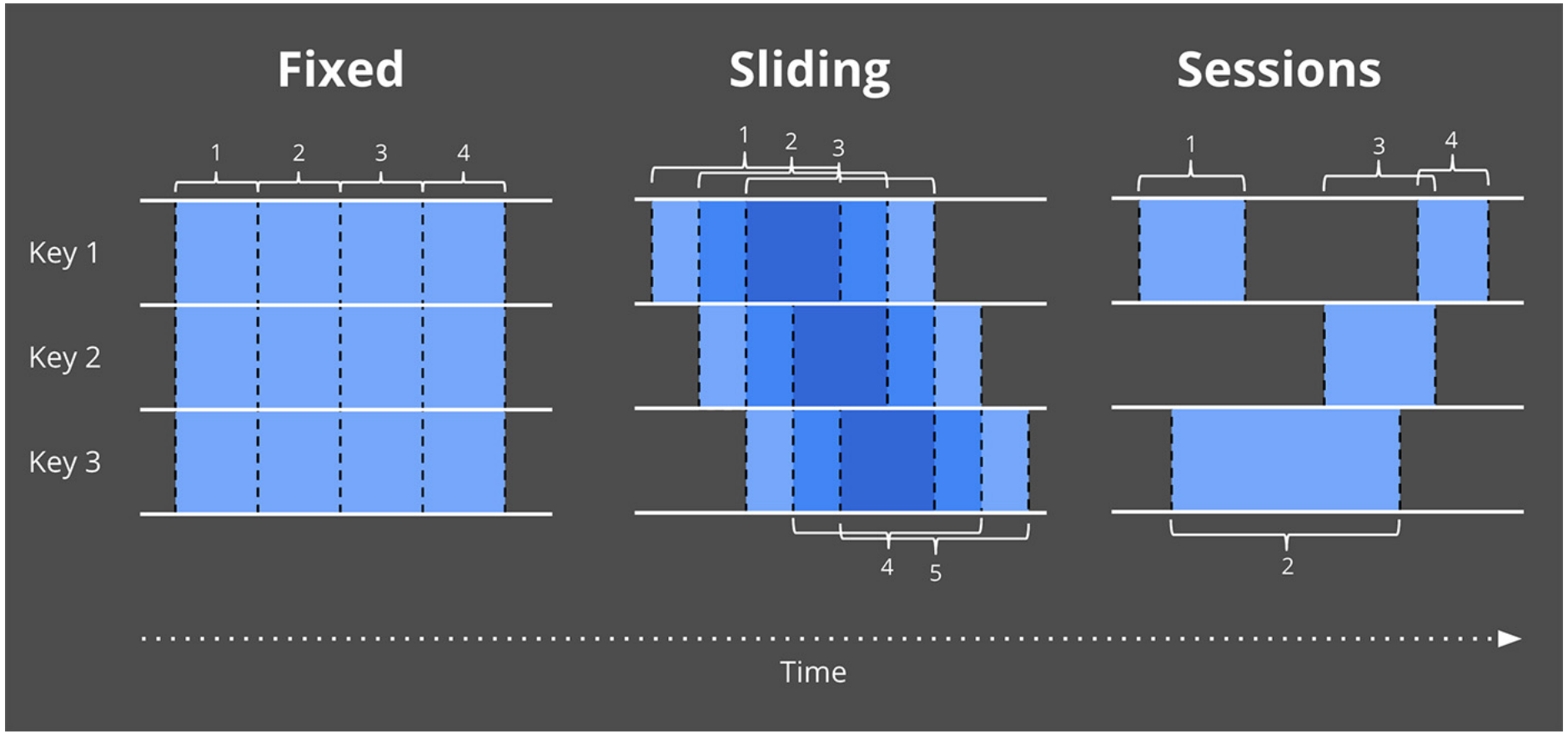


memphis.dev

Filter



Windowing



Demo

- Let's switch to our live demo…

memphis.dev

memphis-example-solutions Public

Edit Pins | Watch 1 | Fork 0 | Star 1

master | 3 branches | 0 tags | Go to file | Add file | Code

rnowling-memphis Remove curl (#18) ... 5d5a39d 2 weeks ago 28 commits

mongodb-debezium-cdc-example	Remove curl (#18)	2 weeks ago
postgres-debezium-cdc-example	Use Python Debian images instead of vanilla Debian images (#15)	last month
CODE_OF_CONDUCT.md	Add license, code of conduct, and README (#1)	3 months ago
LICENSE	Add license, code of conduct, and README (#1)	3 months ago
README.md	Add license, code of conduct, and README (#1)	3 months ago

README.md

Memphis is a next-generation alternative to traditional message brokers.

CLOUD NATIVE COMPUTING FOUNDATION
SILVER MEMBER

Docs - Twitter - YouTube

Introduction to this Repository

This repository contains examples of integrating Memphis, the next-generation message broker, with other open-source technologies to solve technical use cases.

About

Example solutions using Memphis

Readme
View license
Code of conduct
Activity
1 star
1 watching
0 forks
Report repository

Releases

No releases published

Packages

No packages published

Contributors 3

rnowling-memphis RJ Nowling
rnowling RJ Nowling
idanasulinmemphis Idan Asulin

Languages

Python 94.6% Dockerfile 5.4%

https://github.com/memphisdev/memphis-example-solutions

Thanks!