dCache.org

Kafka with dCache

Marina Sahakyan

Hamburg, 28 May

Nordic e-Infrastructure Collaboration

eXtreme DataCloud

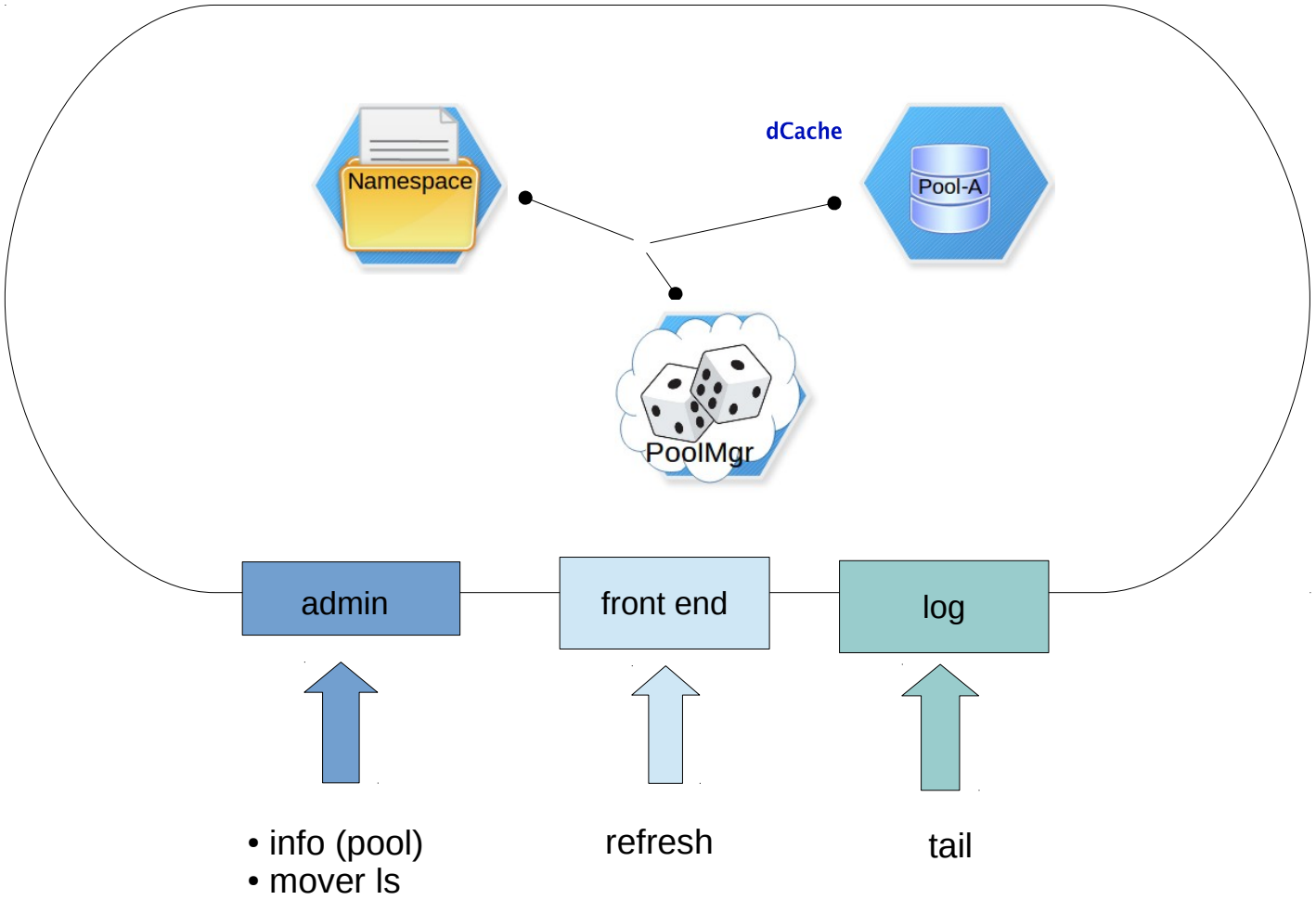
LSDMA

Fermilab

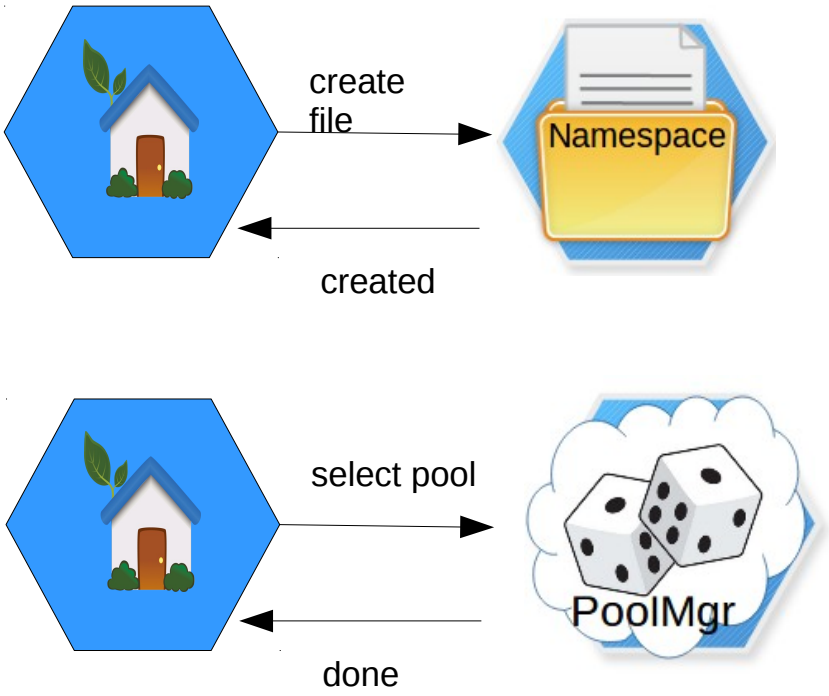
DESY.

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

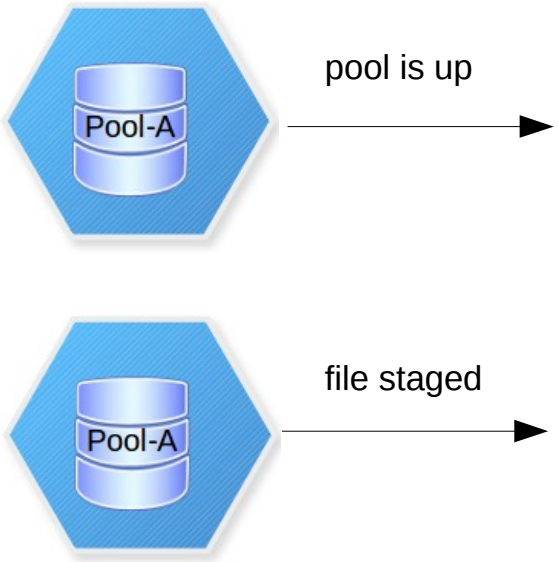
Getting dCache events



Types of messages in dCache

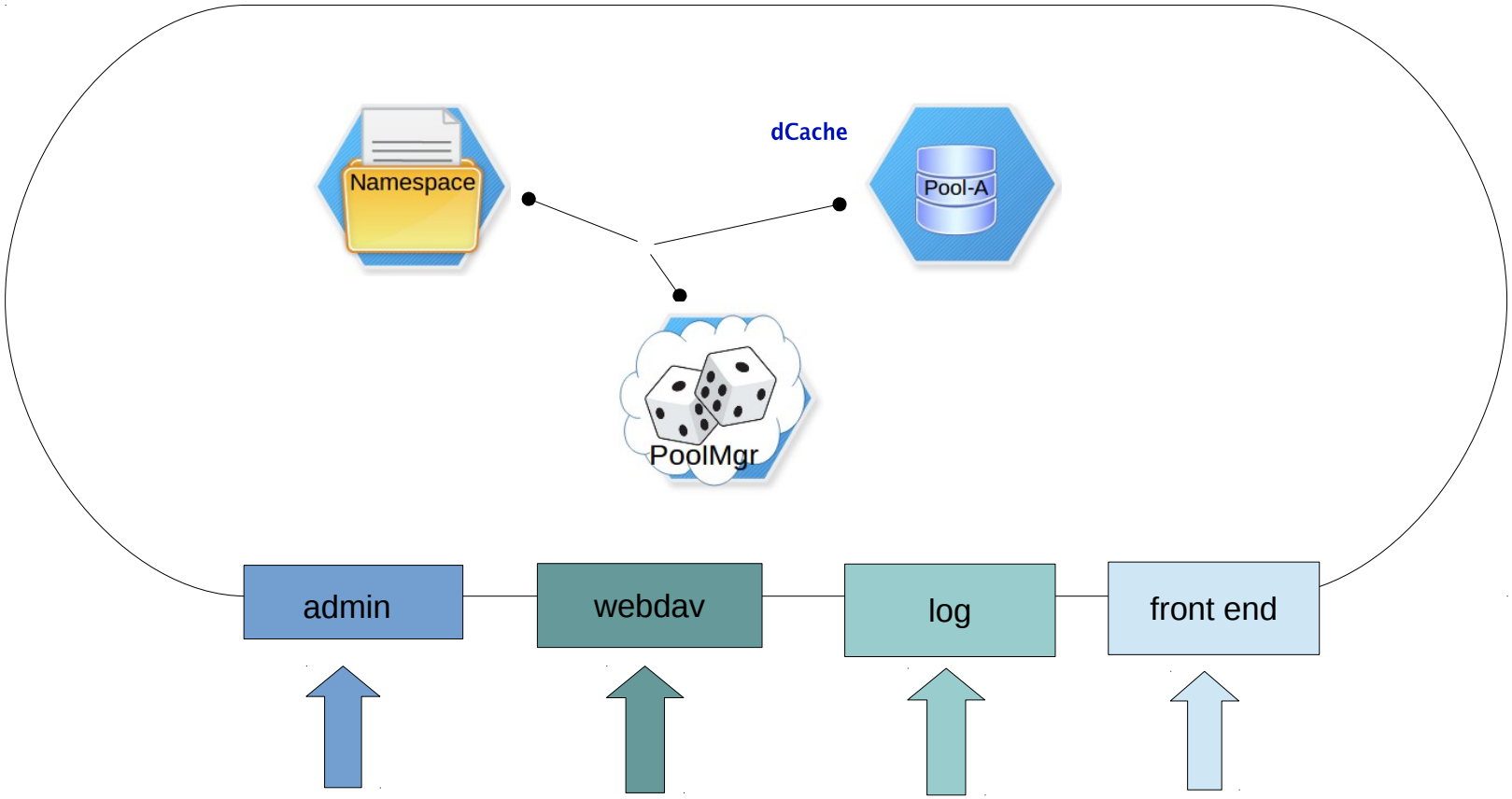


request/response
(rpc like)



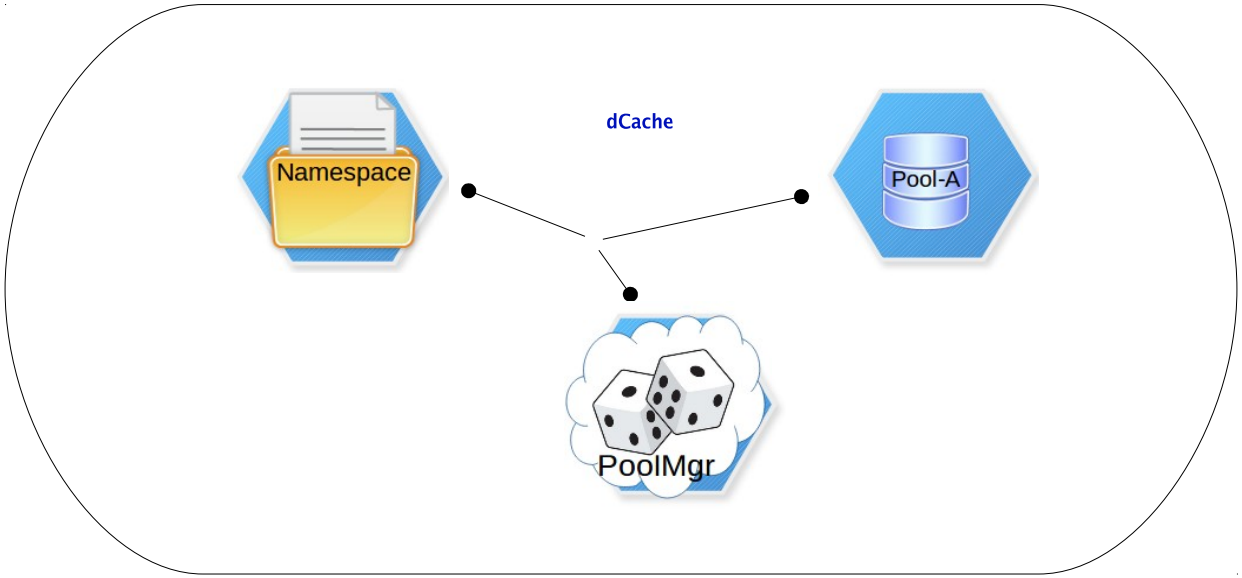
Event notifications

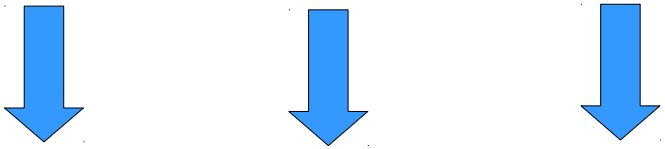
Getting dCache events



poll

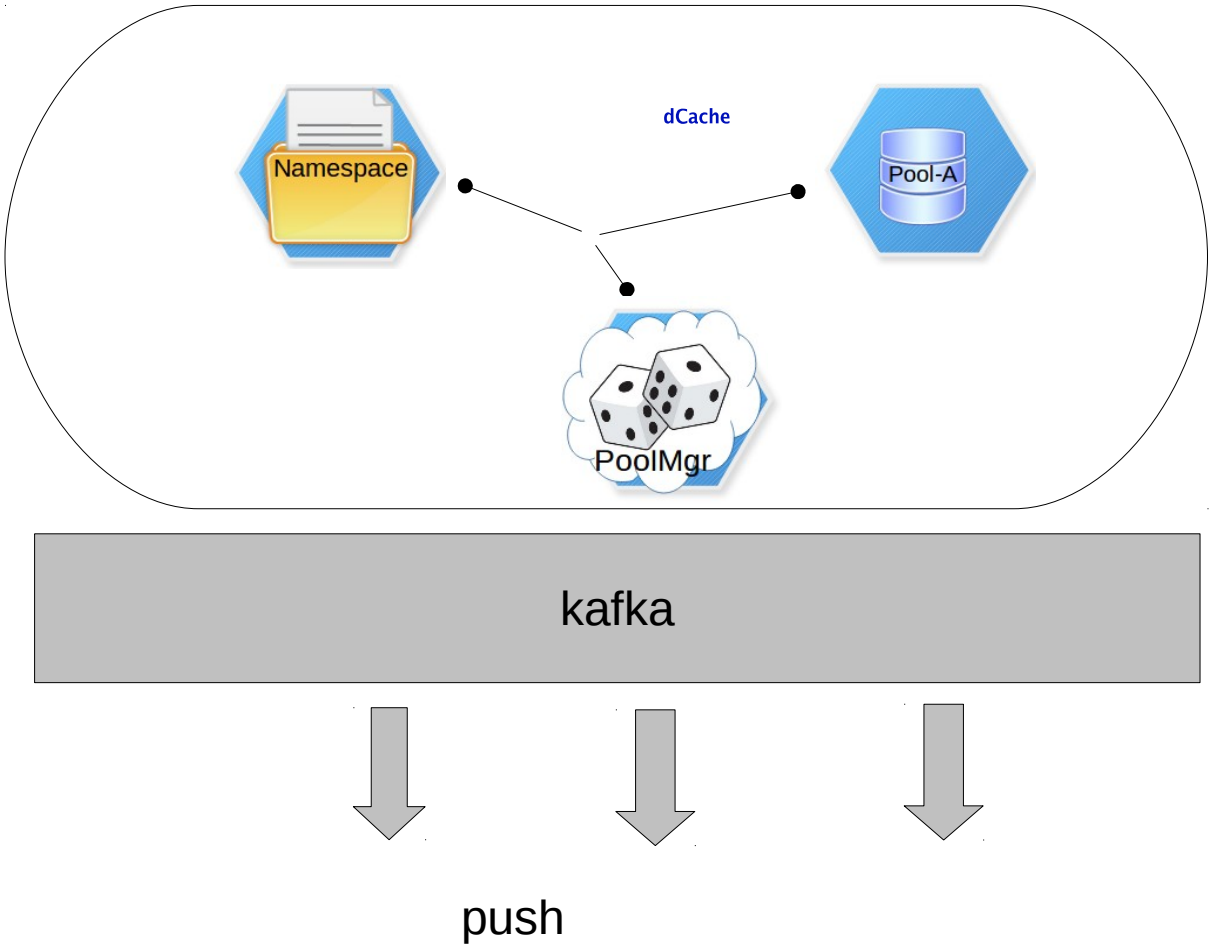
Inversion of event flow



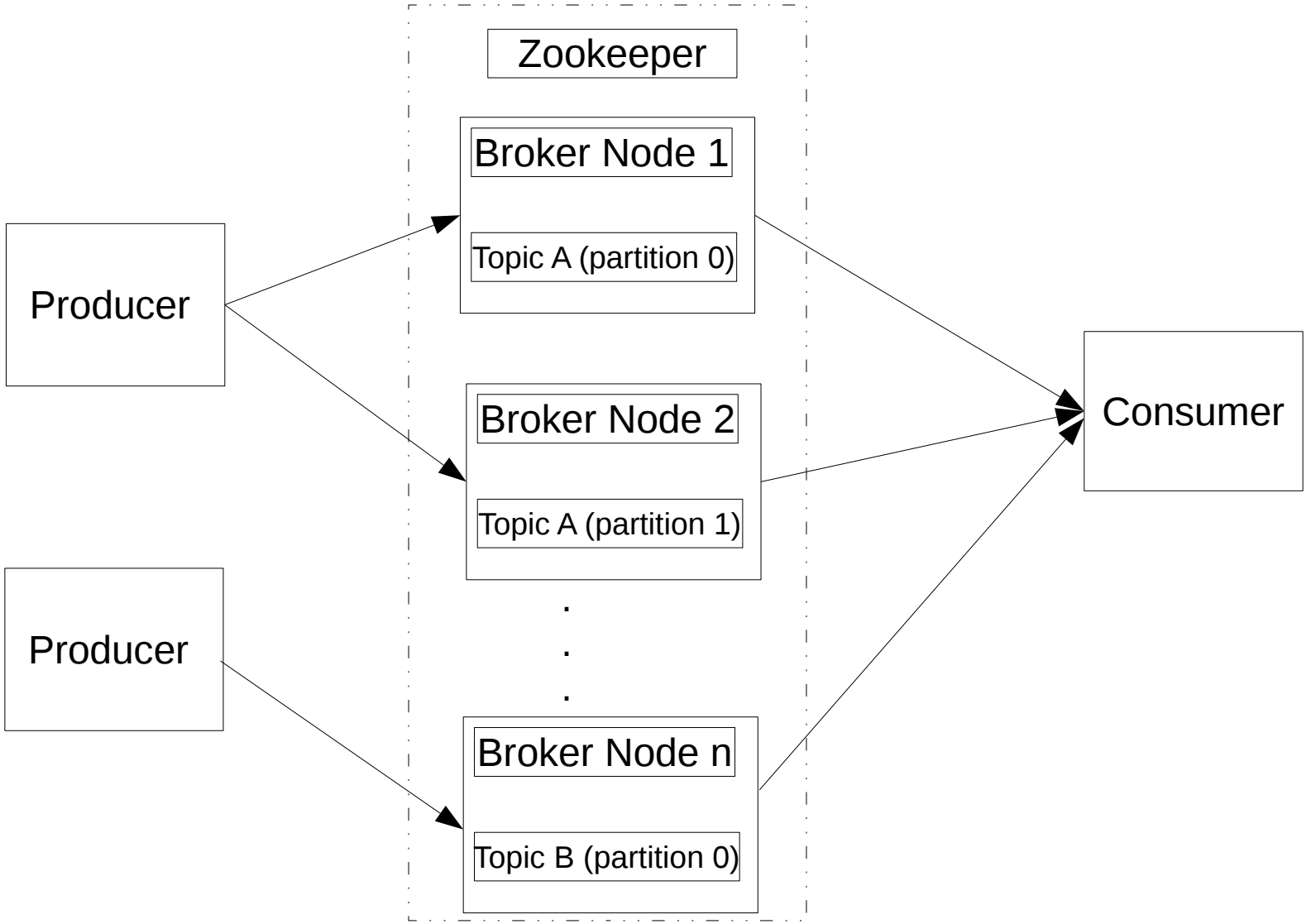


push

How to push dCache events



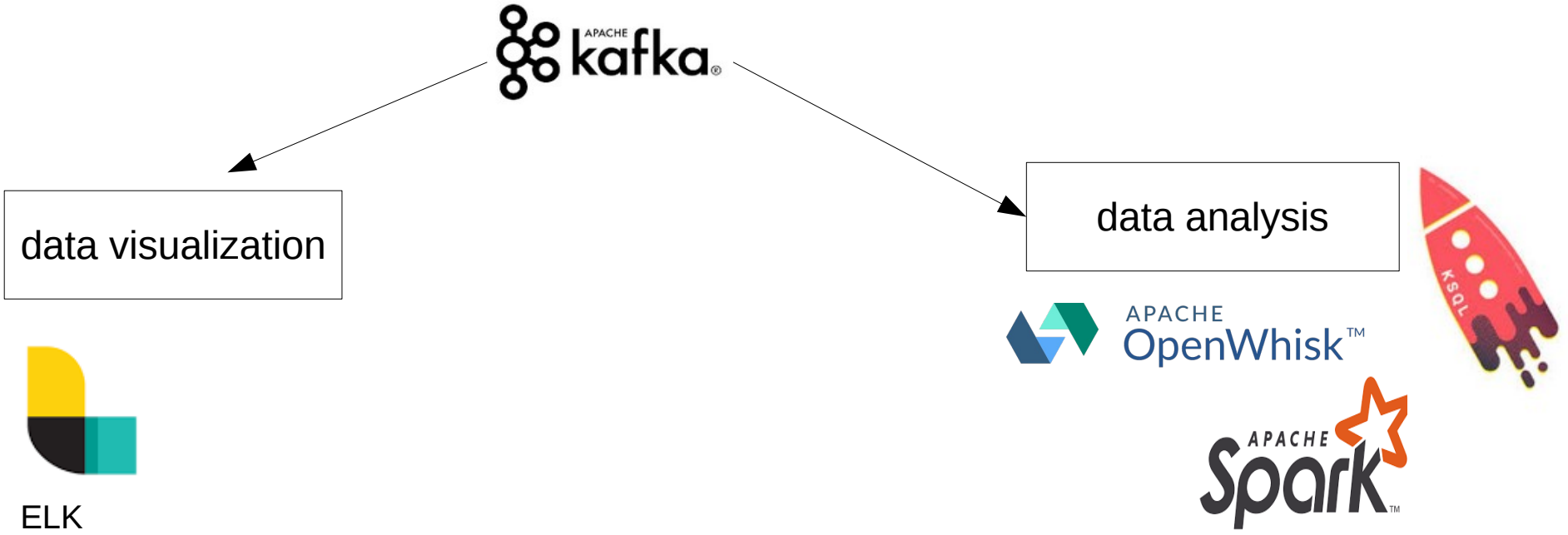
Kafka architecture



Why Kafka

- Industry standard
- Scalable
- Disk-Based Retention
- High Performance

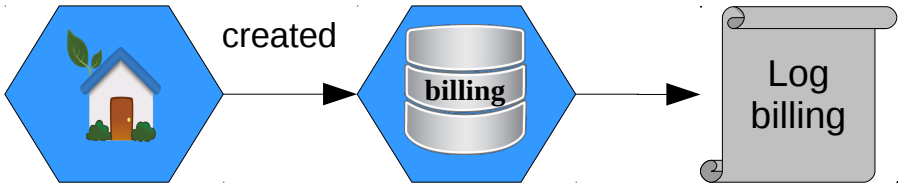
Kafka ecosystem



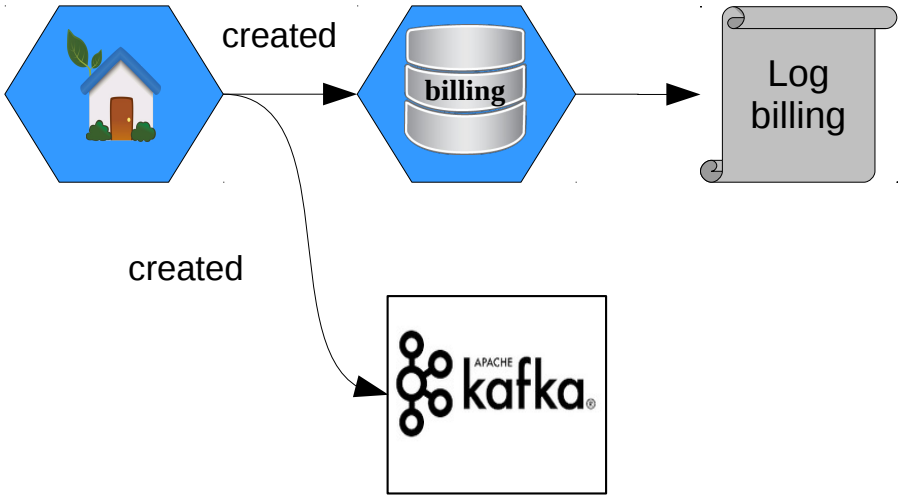
First steps with Kafka

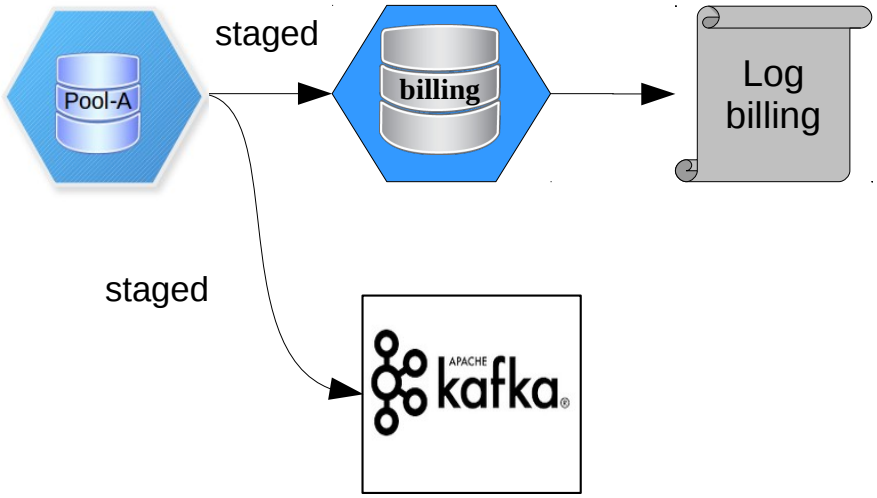
- This is an experiment
  - We may drop it, if doesn't fly
  - If you do nothing dCache carries on as it is
- We start with billing

Billing events

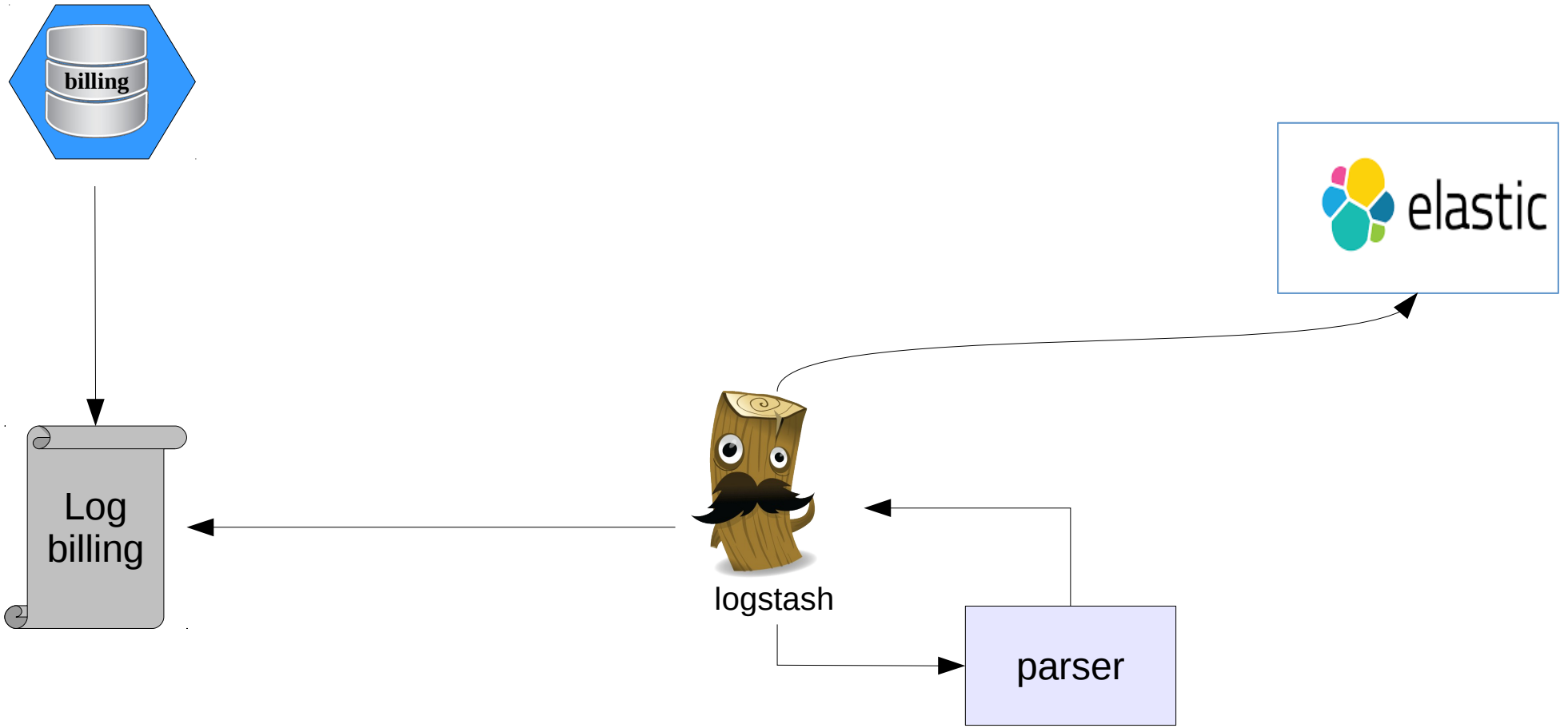


Billing events

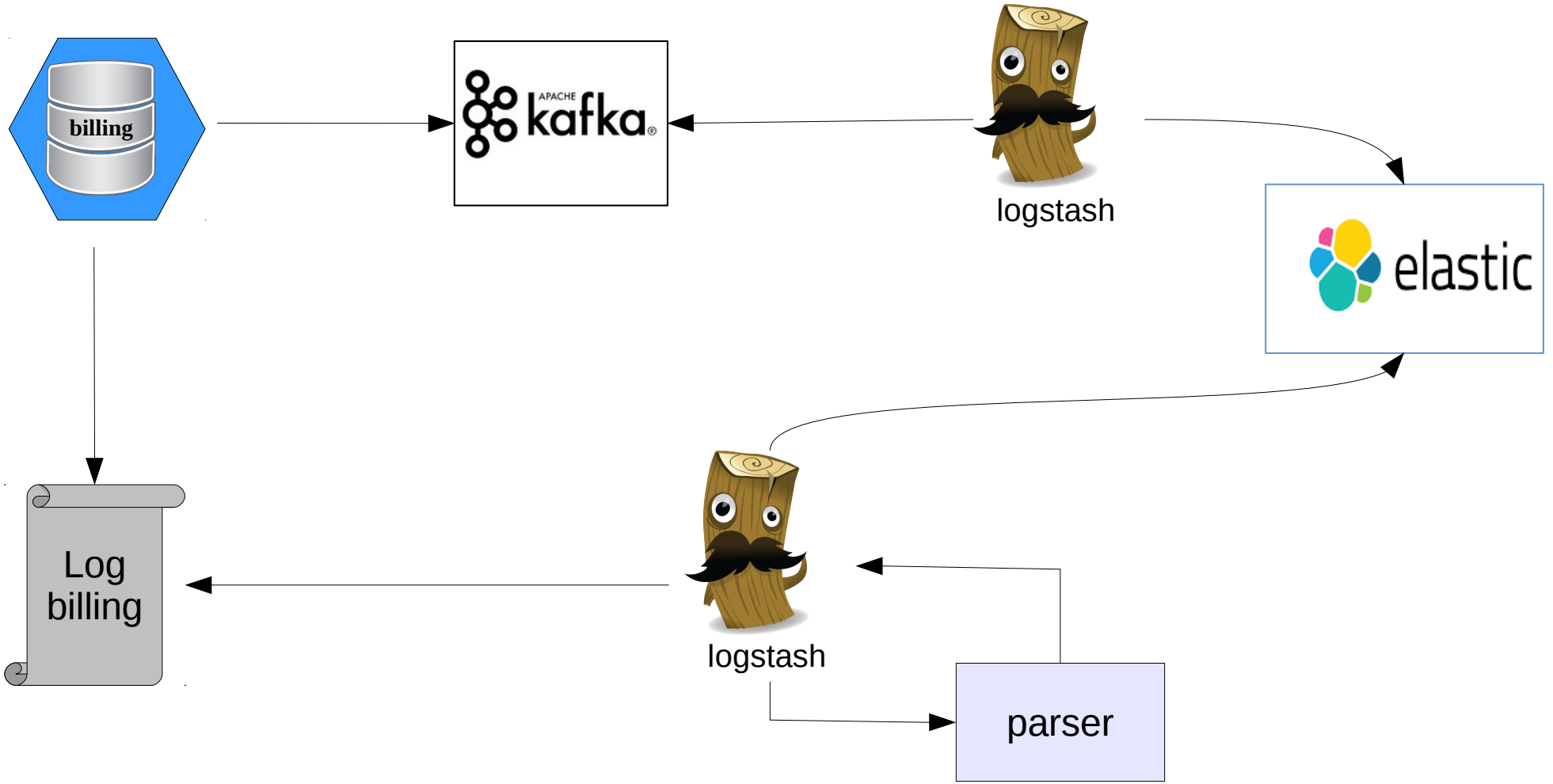




Kafka with dCache



Kafka with dCache



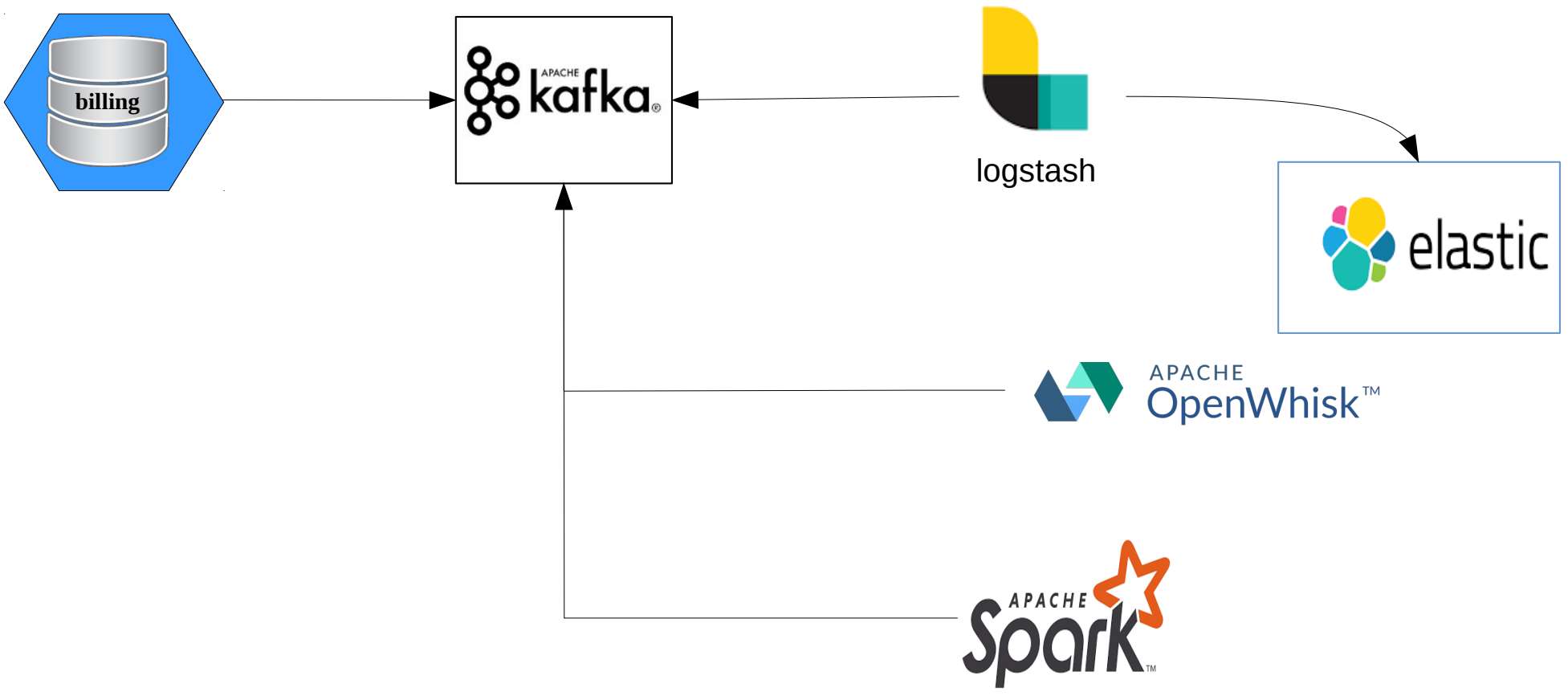
Logstash: Billing file vs Kafka topic

```
filter {
  if ":transfer)" in (message) {
    grok {
      match => ( "message", "%{TRANSFER_CLASSIC}" )
      remove_field => ( "message" )
    }
  } else if ":store)" in (message) {
    grok {
      match => ( "message", "%{STORE_CLASSIC}" )
      remove_field => ( "message" )
    }
  } else if ":restore)" in (message) {
    grok {
      match => ( "message", "%{RESTORE_CLASSIC}" )
      remove_field => ( "message" )
    }
  } else if ":request)" in (message) {
    grok {
      match => ( "message", "%{REQUEST_CLASSIC}" )
      remove_field => ( "message" )
```

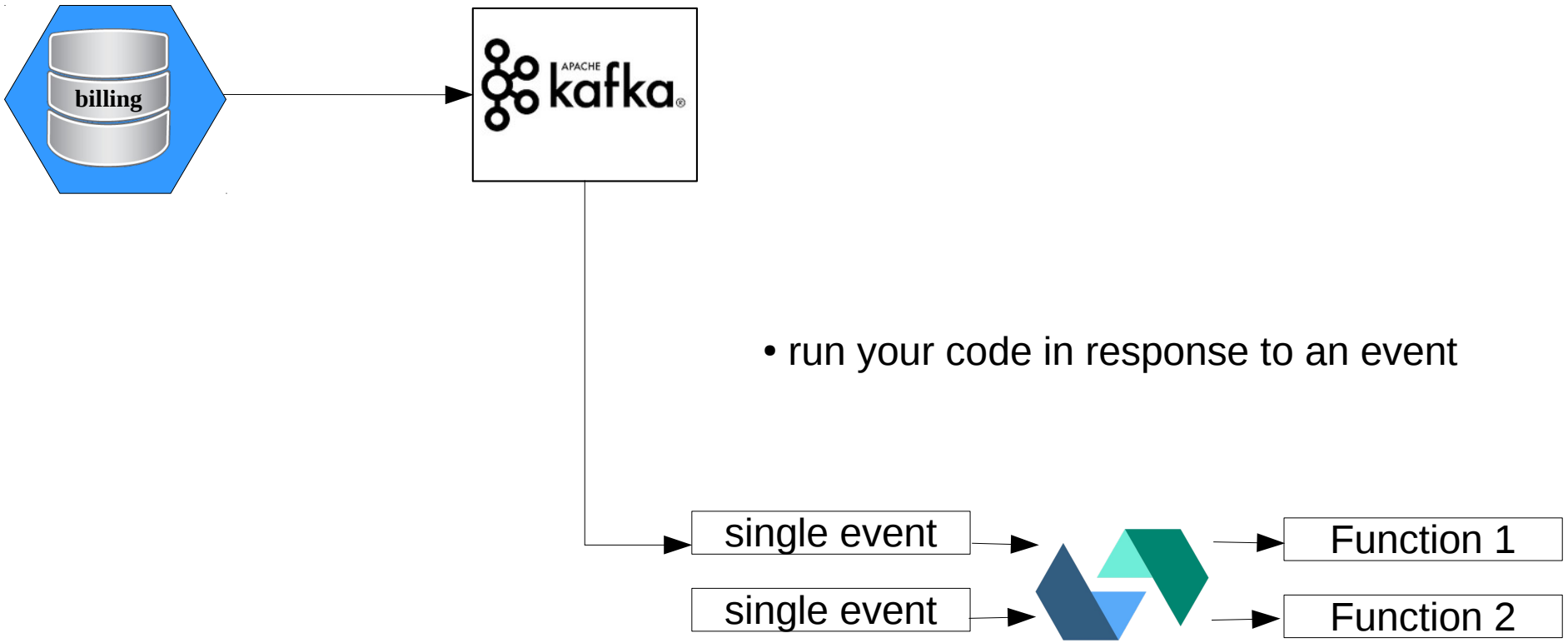
```
filter {

}
```

Kafka with dCache



Kafka with dCache



- run your code in response to an event

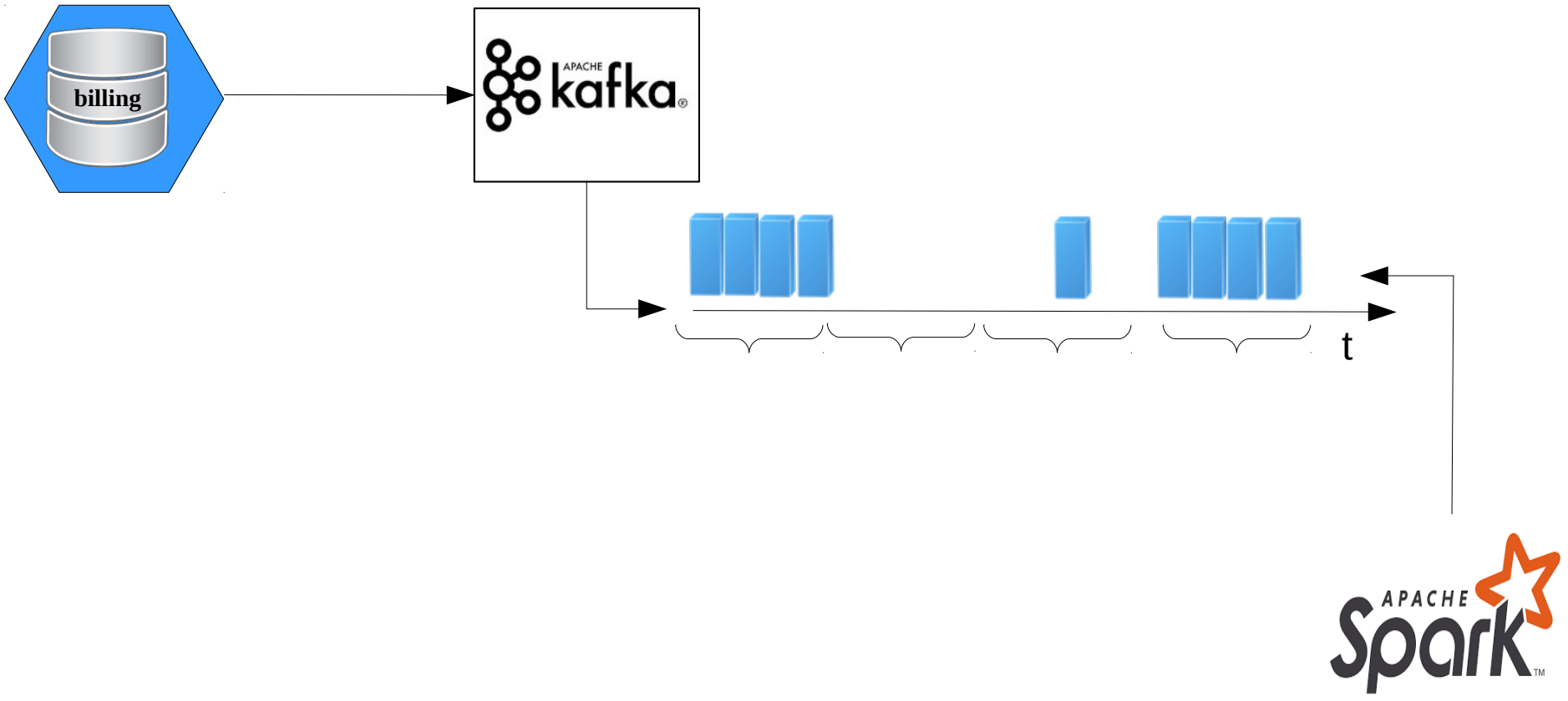
Kafka with dCache

For example

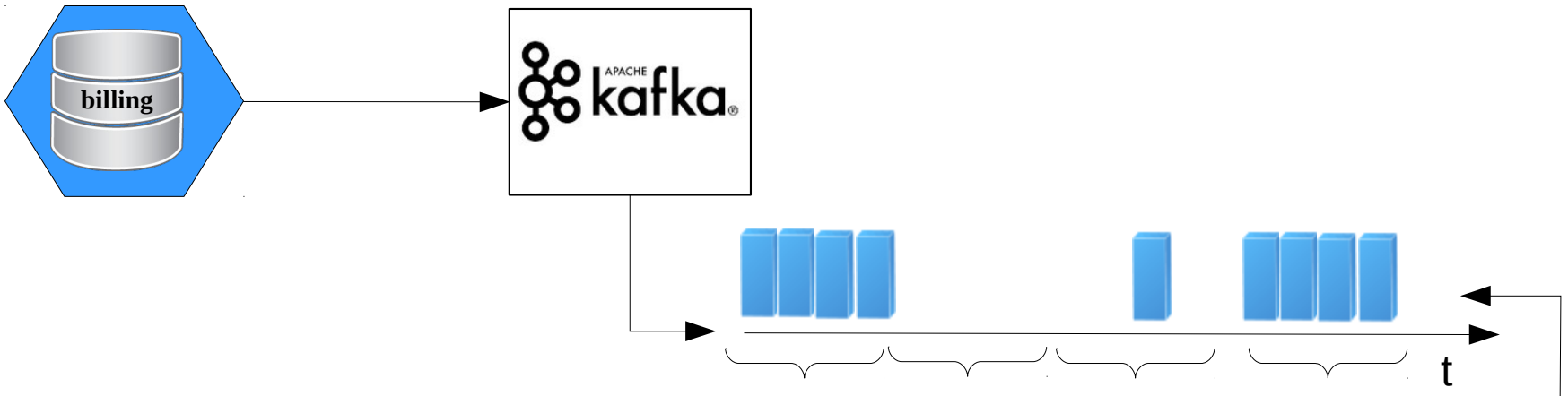
- generate Thumbnails for each file
- extract metadata
- file compression

• run your code in response to an event

Kafka with dCache



Kafka with dCache



Some ideas

- total size of file transfers in last 24 hours
- average transferTime
- pool usage statistics
- (or apply deep learning :-D), if you want to be buzz word compliant

dCache configs for Kafka

1. install Kafka

- you can share zookeeper with dCache

Enabling Kafka in dCache

1. install Kafka

- you can share zookeeper with dCache

2. enable Kafka in dCache

- dcache.enable.kafka = true
- webdav.enable.kafka
- nfs.enable.kafka
- xrootd.enable.kafka
- pool.enable.kafka

Enabling Kafka in dCache

1. install Kafka

- you can share zookeeper with dCache

2. enable Kafka in dCache

- dcache.enable.kafka = true
- webdav.enable.kafka
- nfs.enable.kafka
- xrootd.enable.kafka
- pool.enable.kafka

3. set servers

- dcache.kafka.bootstrap-servers = broker-host:9092

Enabling Kafka in dCache

1. install Kafka

- you can share zookeeper with dCache

2. enable Kafka in dCache

- dcache.enable.kafka = true
- webdav.enable.kafka
- nfs.enable.kafka
- xrootd.enable.kafka
- pool.enable.kafka

3. set servers

- dcache.kafka.bootstrap-servers = broker-host:9092
  - dcache.kafka.bootstrap-servers = broker-host1:9092,broker-host2:9092

Enabling Kafka in dCache

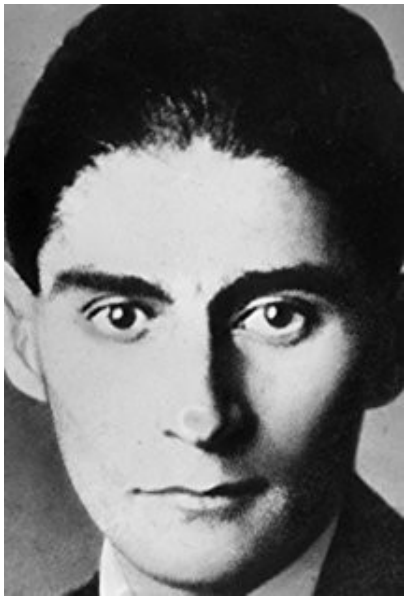
Example/ write to storage


```
{
        "date":"Fri May 11 12:14:45 CEST 2018“,
        "msgType":"store",
        "transferTime":1062,
        "cellName":"pool_write",
        "session":"pool:pool_write@dCacheDomain:1526033685223-22",
        "subject": ("UidPrincipal(0)",
        "GidPrincipal(0,primary)"),
        "version":"1.0",
        "storageInfo":"test:tape@osm",
        "cellType":"pool",
        "fileSize":378,
        "queuingTime":1148,
        "cellDomain":"dCacheDomain",
        "pnfsid":"000003EBFAC026BB4521B8B68E7FE7734D9A",
        "transaction":"pool:pool_write@dCacheDomain:1526033685223-22",
        "billingPath":"/",
        "status":{
                "msg":"",
                "code":0}
}
```

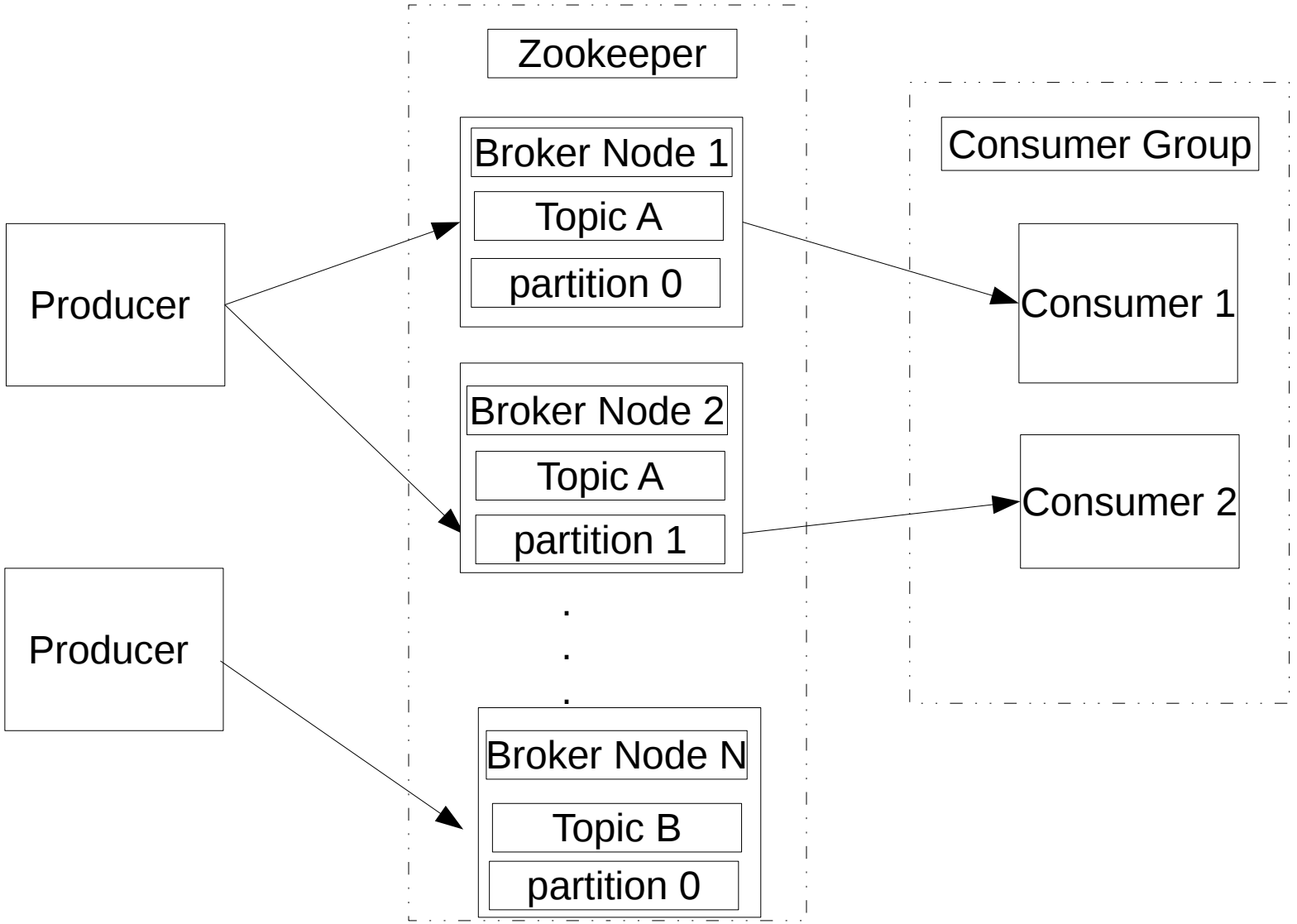
Summary

- We are working on exposing internal events to non-dCache components
- Apache-Kafka
  - You can connect your analytics or event driven processing to dCache
- Give it a try, share with us your experience!

Thank you!



Kafka architecture/simple Kafka cluster



Why Kafka

- Industry standard / accessible documentation
  - Widely used
- Scalable
  - Easy to handle any amount of data
    - ???
- Disk-Based Retention
  - Consumer do not always need to work in real time
- High Performance
  - Scaled to handle very large message streams with ease

Why Kafka

- Scalable (consumer throughput)
  - Single Consumer
    - 940,521 records/sec
    - (89.7 MB/sec)
  - Three Consumers
    - 2,615,968 records/sec
    - (249.5 MB/sec)
  - Producer and Consumer
    - 795,064 records/sec
    - (75.8 MB/sec)