



# **Business Intelligence in Microservice Architecture**

Debarshi Basak @ bol.com

# What can you expect?

- Introduction
- Monolithic days
- Mapreduce Era
- Flink Era
- Operational Aspect

# Who am I?

Debarshi Basak

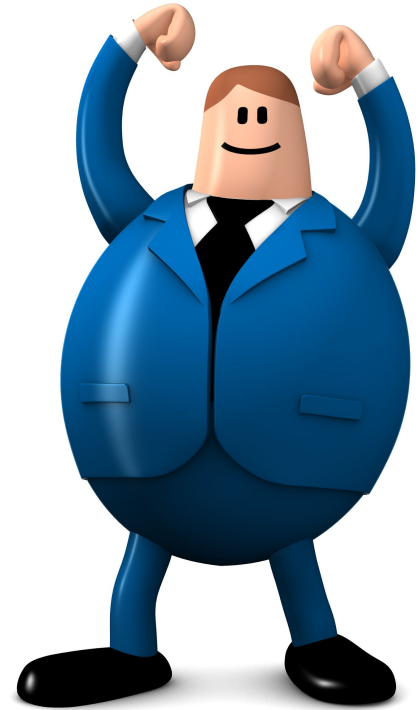
Software engineer at bol.com

Part of Bigdata platform team and online marketing

# About bol.com

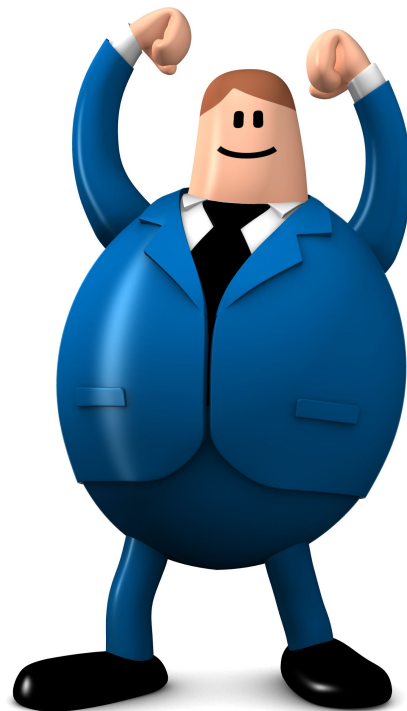
- Leader in Dutch eCommerce
- Scrum
- 1000+ employees
- 40+ scrum teams
- Young and relaxed

You build it. You run it. You love it.



# How big is big data at bol.com?

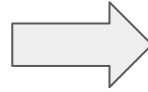
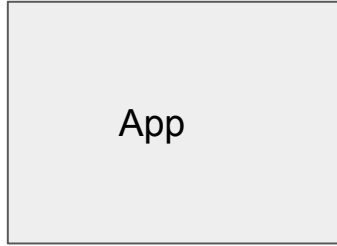
- > 11.000.000 products for sale
  - Catalog > 38.000.000
  - 400.000.000 newsletter responses
  - 15.000.000 new clicks every day
- 
- 26 node cluster
  - More than 300 jobs a month in production



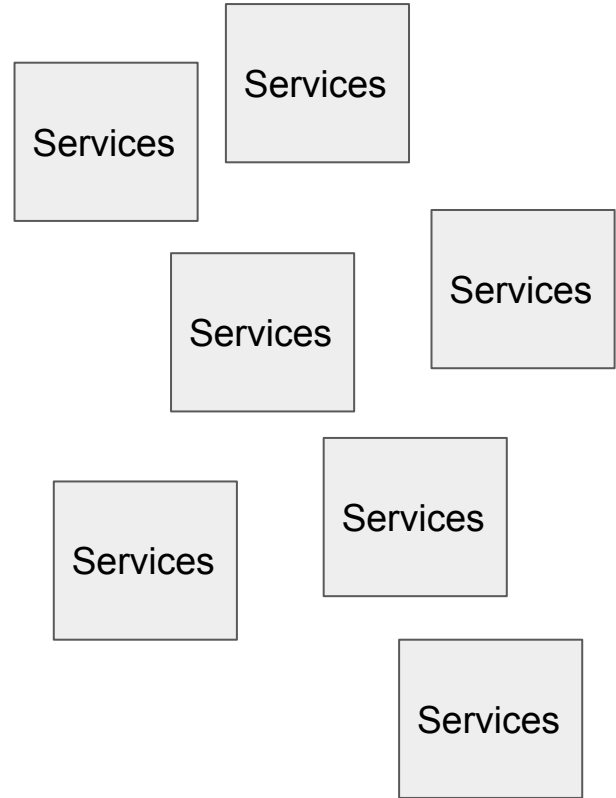
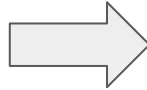
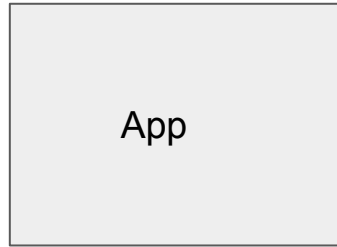
**What is Microservice?**



# What is Microservice?



# What is Microservice?





# Business Intelligence 101





# Business Intelligence 101

- Analyzing data and presenting actionable items



# Business Intelligence 101

- Analyzing data and presenting actionable items
- Automated and Continuous Integration with internal and external data sources



# Business Intelligence 101

- Analyzing data and presenting actionable items
- Automated and Continuous Integration with internal and external data sources
- Flexible Analytics



# Business Intelligence 101

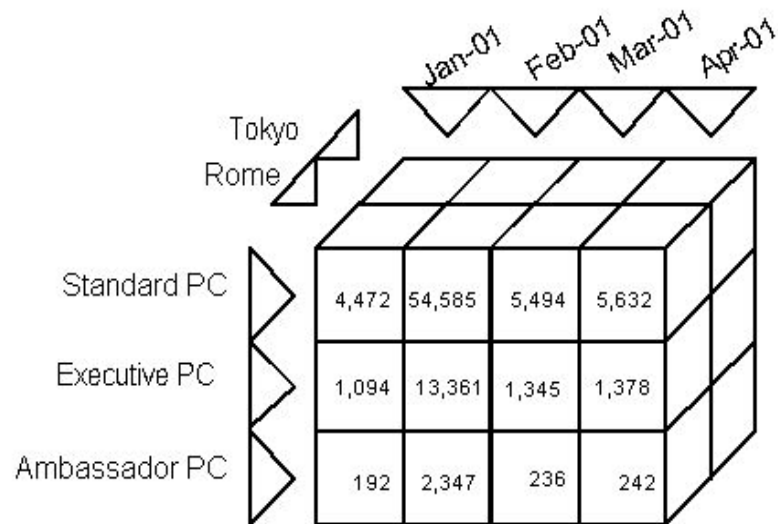
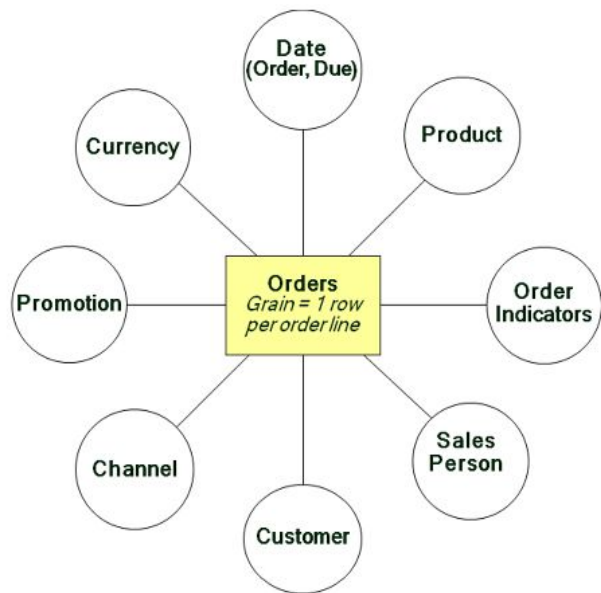
- ETL
  - Extract from source
  - Transform the data
  - Load into target data models



# Business Intelligence 101

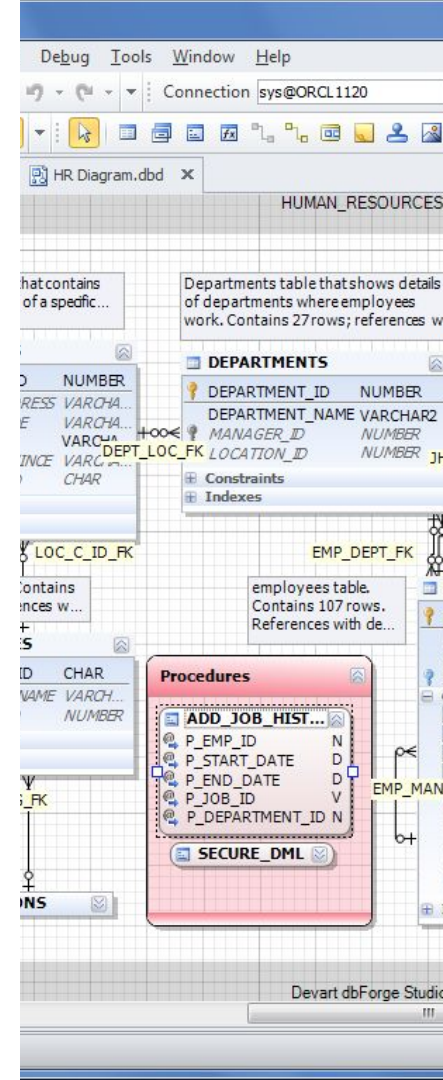
- ETL
  - Extract from source
  - Transform the data
  - Load into target data models
  
- Business data modelling
  - Kimball's dimensional modeling technique
  - OLAP cubes



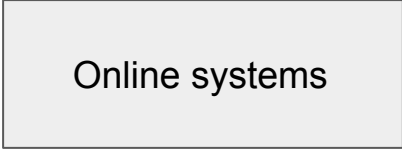




# Monolithic days

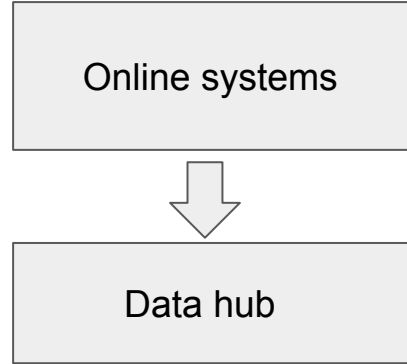


# Evolution of BI Systems (at bol.com)

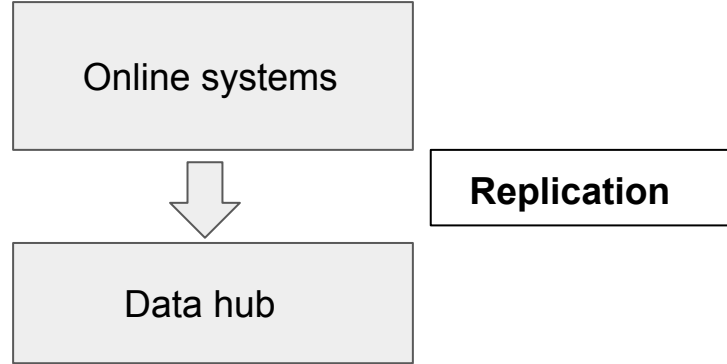


Online systems

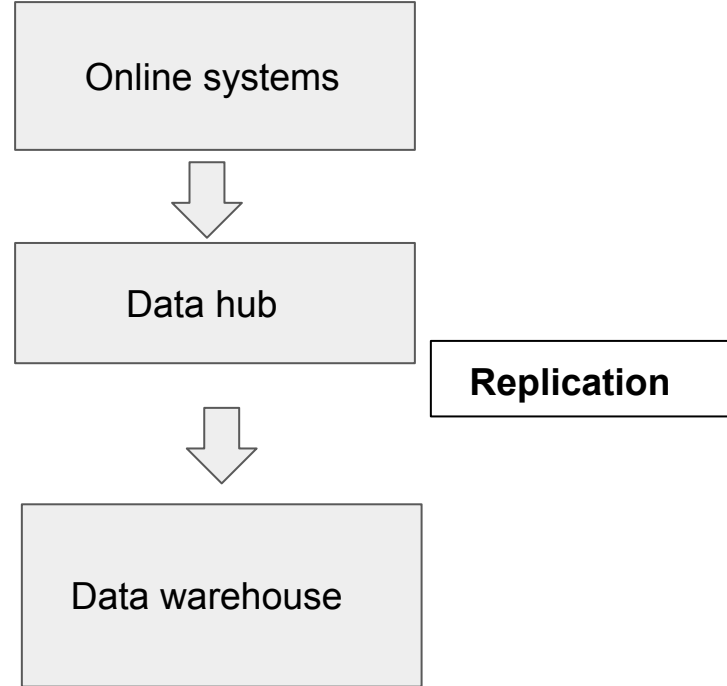
# Evolution of BI Systems (at bol.com)



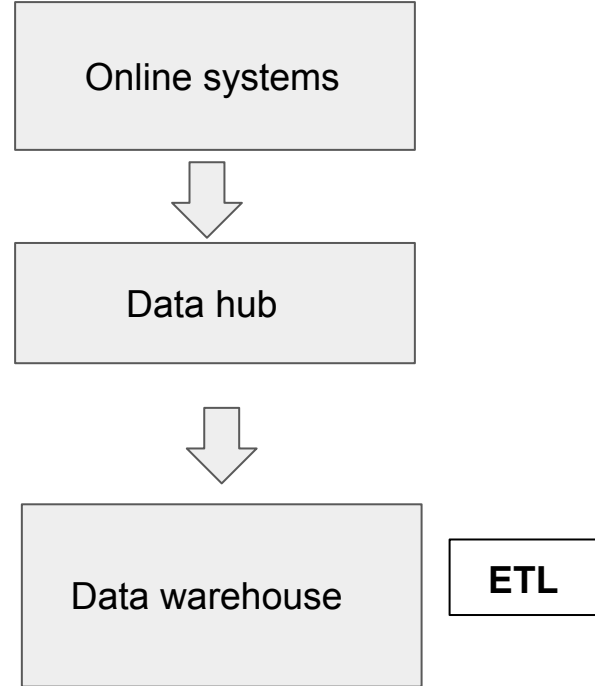
# Evolution of BI Systems (at bol.com)



# Evolution of BI Systems (at bol.com)



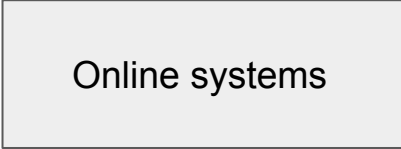
# Evolution of BI Systems (at bol.com)



# Evolution of BI Systems (at bol.com)

- Easy to implement
- Complexities are abstracted
- Data Overheads
- Latency

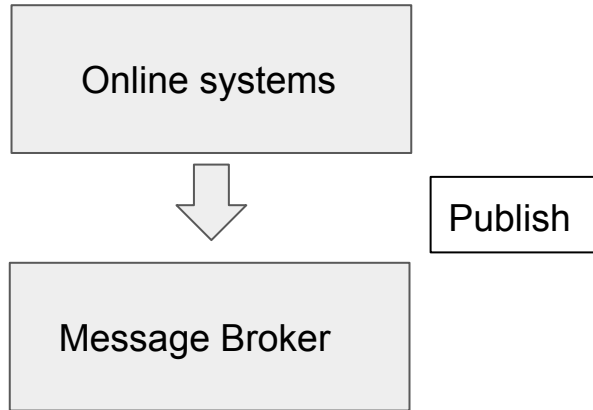
# Evolution of BI Systems (at bol.com)



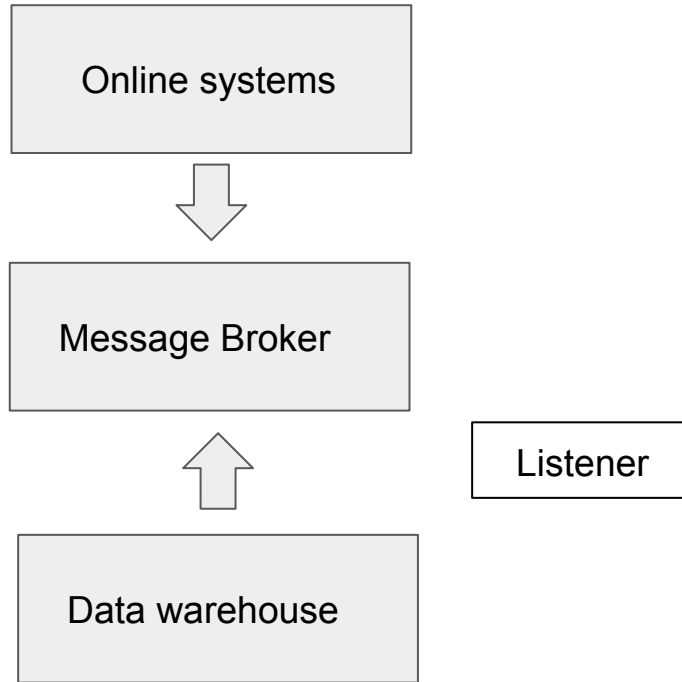
Online systems



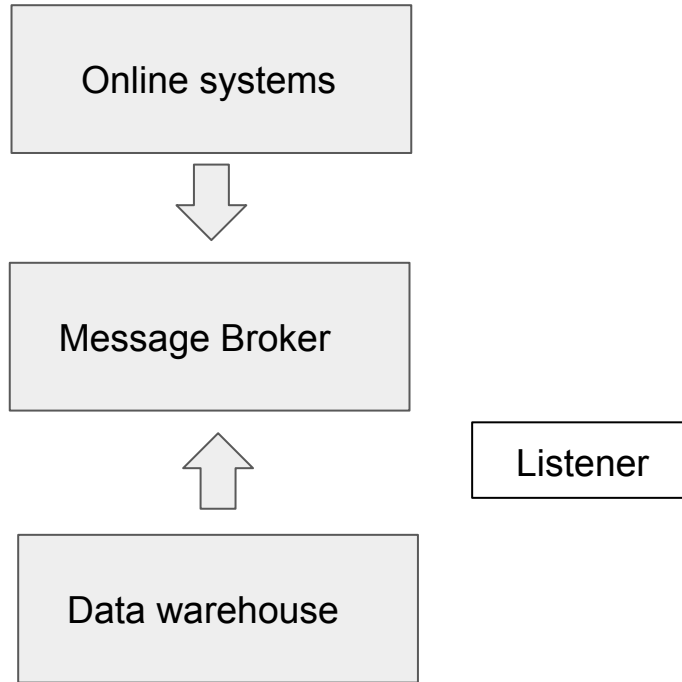
# Evolution of BI Systems (at bol.com)



# Evolution of BI Systems (at bol.com)



# Evolution of BI Systems (at bol.com)



# Evolution of BI Systems (at bol.com)

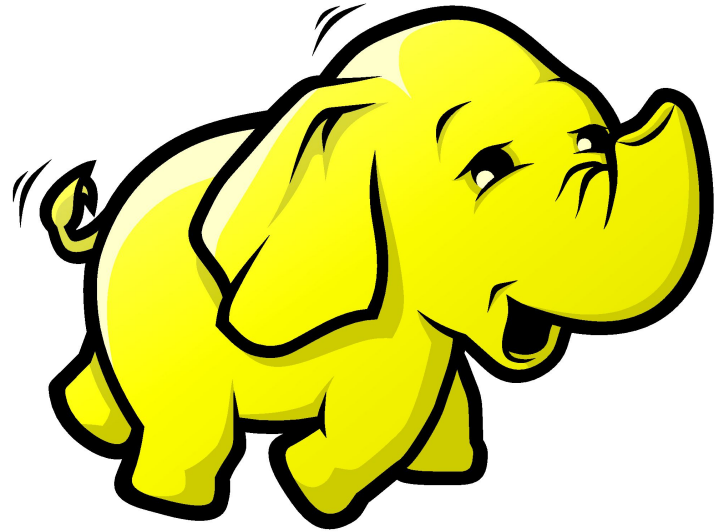
- Loss of Messages and Consistency guarantees
- Database are kind of not made for this
- Complex implementation
- Nightmare for operations

# Challenges in Microservice Architecture

# Challenges in Microservice Architecture

- Too many sources
- Can affect scalability and stability of reports
- BI cannot scale
- Extraction logic, transformation operations for each Service
- Joins.

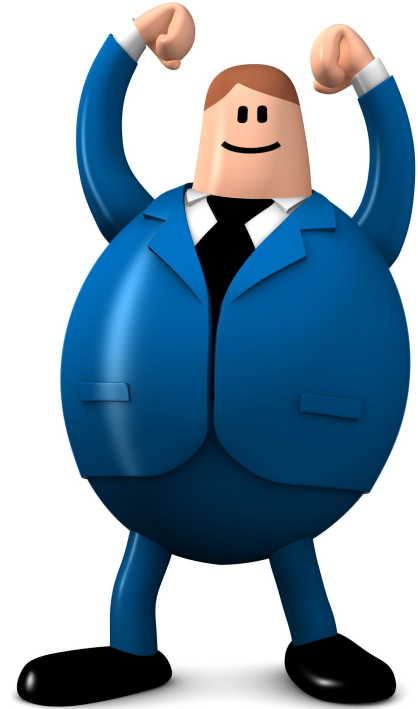
**Hadoop era**



# History of Hadoop at bol.com

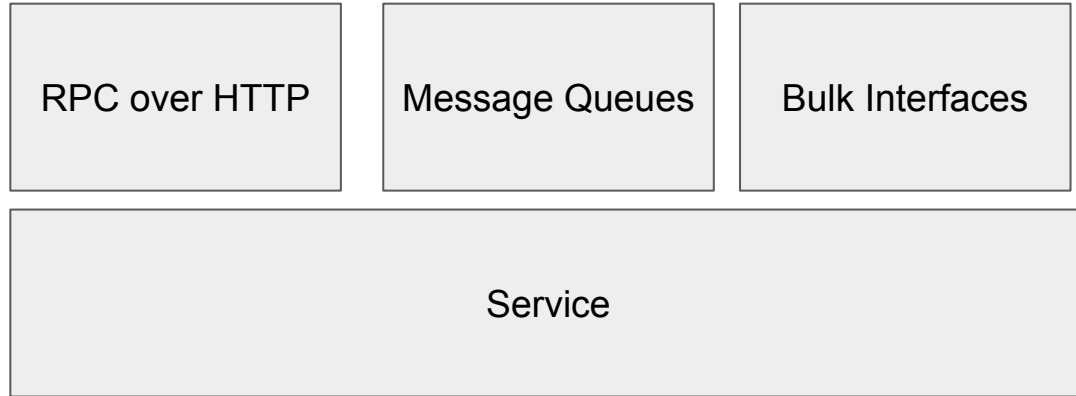
Operational experience in hbase, hadoop based tooling

- Supplier connector
- Recommendation system





# Service Definition



# Bulk Interfaces

t1-productId1	f:price::15,50
t2-productId2	f:price::15,35
t3-productId1	f:price::15,25
t4-productId1	f:price::15,75
t5-productId3	f:price::15,50

# Bulk Interfaces

t1-productId1	f:price::15,50
t2-productId2	f:price::15,35
t3-productId1	f:price::15,25
t4-productId1	f:price::15,75
t5-productId3	f:price::15,50

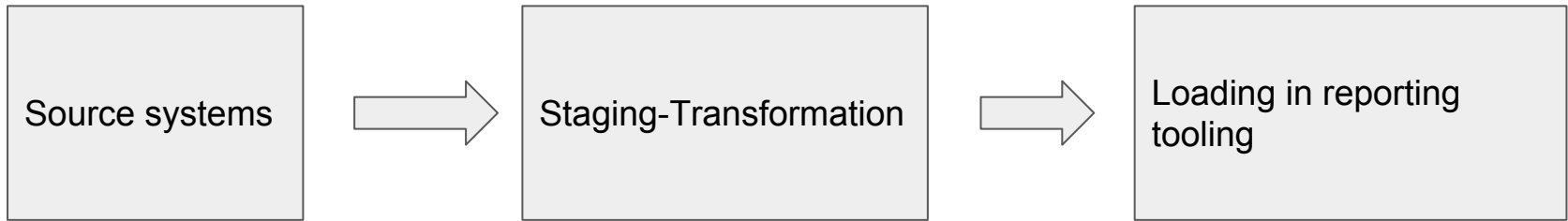
We can replay event to get the latest state of the event.  
This is also known as Event Sourcing Pattern.

Similar key design can be found in OpenTSDB

# Re-imagine traditional BI on hadoop



# Re-imagine traditional BI on hadoop



Supplier  
Service

Offers  
Services

Pricing  
Services

Data warehouse

From Queues



$\delta$   
Supplier  
Service

$\delta$   
Offers  
Services

$\delta$   
Pricing  
Services

Data warehouse



Supplier  
Service



Offers  
Services



Pricing  
Services



Data warehouse

Supplier  
Service



Offers  
Services



Pricing  
Services



Data warehouse

 RUNDECK

Cronacle

Supplier  
Service



Offers  
Services

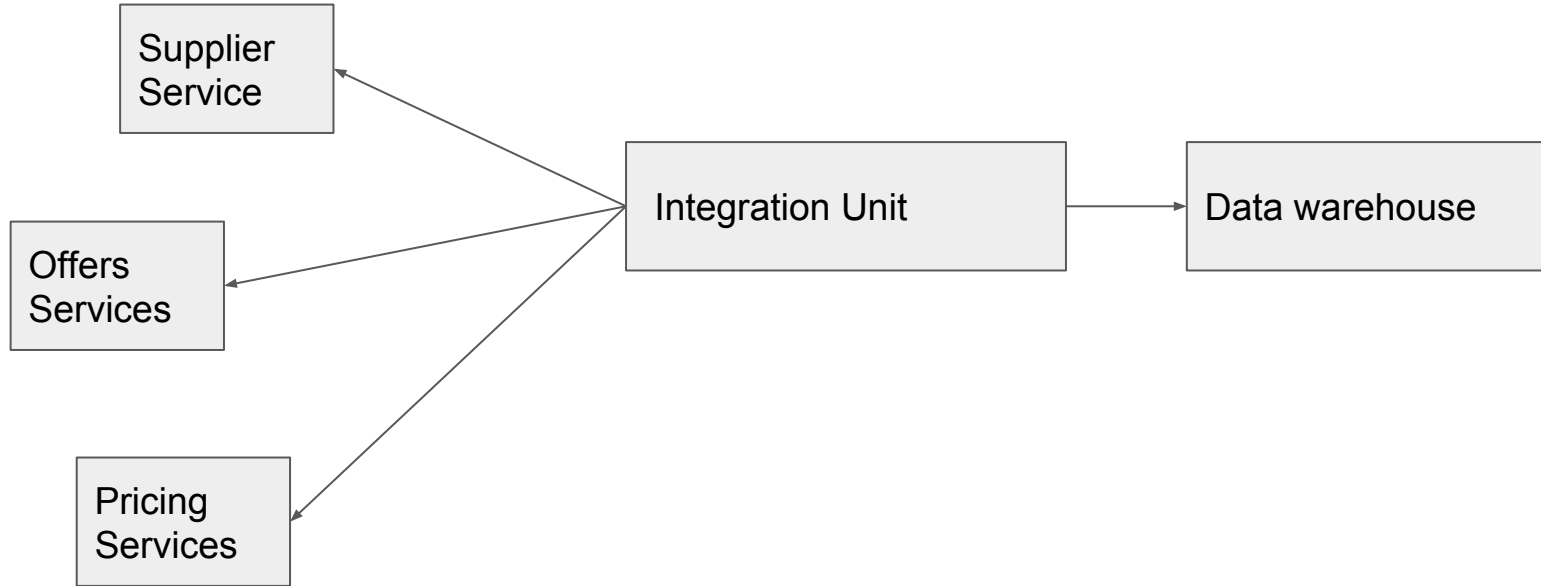


Data warehouse

Pricing  
Services



# Automation



# What kind of jobs we build

- Aggregation jobs
  - Single service, aggregate on a function key
- Interface concatenation
  - Multiple services combined on one/many functional keys.

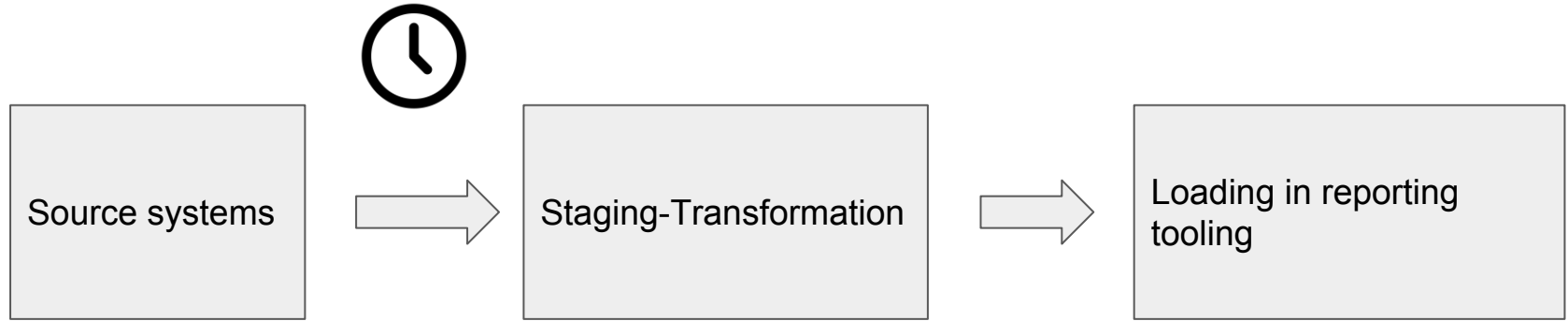
# Automation

```
{
  "bulk_interface": "transport_acc_public_v1_Transporter_versions",
  "table_name": "transporter",
  "primary_key": "f:Transporter.TransporterId",
  "ctl_name": "transporter",
  "service_version": "v2",
  "col_map":[
    {
      "hbase_col_name": "f:Transporter.TransporterId",
      "ora_col_name":"Id",
      "function_name":"Transporter id",
      "data_type" : "NUMBER"
    },
    {
      "hbase_col_name": "f:Transporter.TransporterCode",
      "ora_col_name":"Code",
      "function_name":"Transporter code",
      "data_type" : "VARCHAR2(40)"
    },
    {
      "hbase_col_name": "f:Transporter.TransporterName",
      "ora_col_name":"TransporterName",
      "function_name":"Transporter Name",
      "data_type" : "VARCHAR2(40)"
    }
  ]
}
```

# Problem

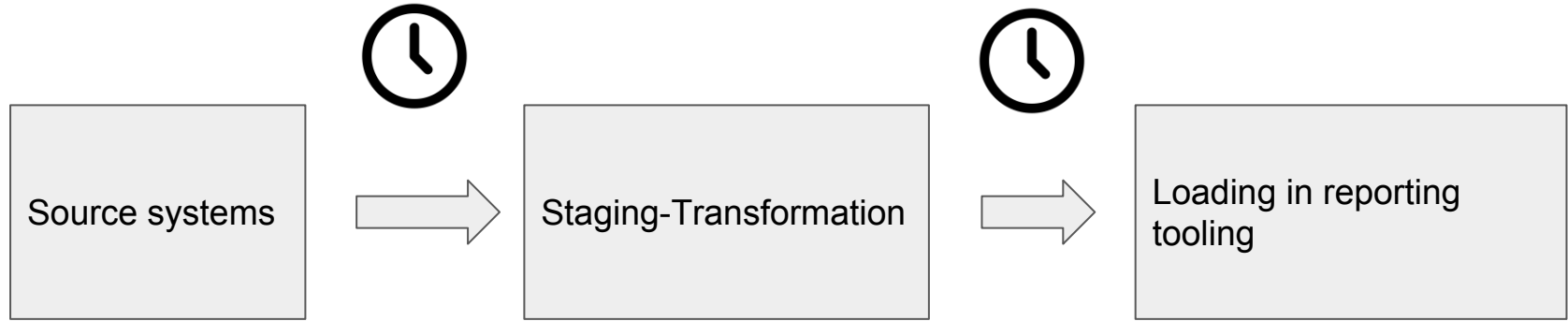


# Problem





# Problem



# **But everything is stream.**

Nature of data in most of use cases is asynchronous.

Clicks are asynchronous

Orders are asynchronous

Updates are asynchronous

In fact, Batch is a bounded stream.

**Streaming era**



**Flink**

# Enter Flink

Low entry barrier

Java/Scala functional apis.

Operational expertise.

# Emulating Stream

You don't always need queues for stream

Streaming HBase tables.

Give a starting point in stream



t1-productId1	f:price::15,50
t2-productId2	f:price::15,35
t3-productId1	f:price::15,25
t4-productId1	f:price::15,75
t5-productId3	f:price::15,50

Give next 'x' records



t1-productId1	f:price::15,50
t2-productId2	f:price::15,35
t3-productId1	f:price::15,25
t4-productId1	f:price::15,75
t5-productId3	f:price::15,50

Give next 'x' records



t1-productId1	f:price::15,50
t2-productId2	f:price::15,35
t3-productId1	f:price::15,25
t4-productId1	f:price::15,75
t5-productId3	f:price::15,50



```
while(true){  
  Give next 'x' records  
}
```

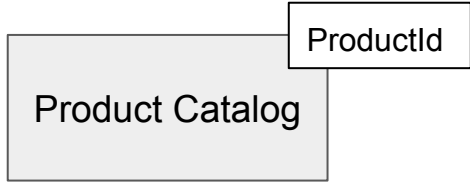
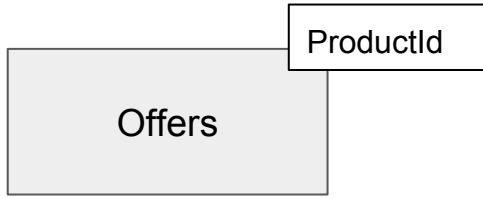


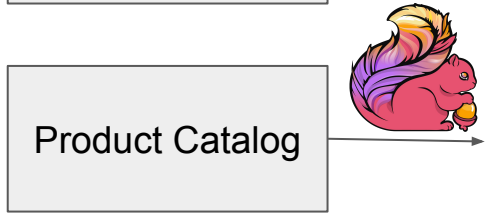
t1-productId1	f:price::15,50
t2-productId2	f:price::15,35
t3-productId1	f:price::15,25
t4-productId1	f:price::15,75
t5-productId3	f:price::15,50



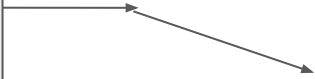
Offers

Product Catalog





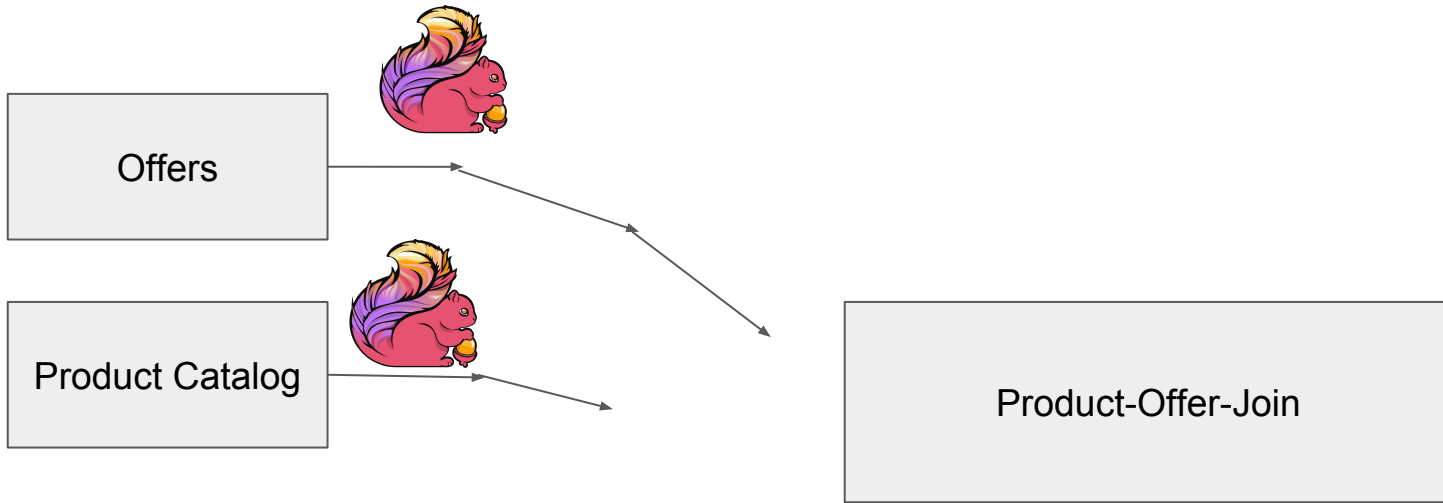
Offers

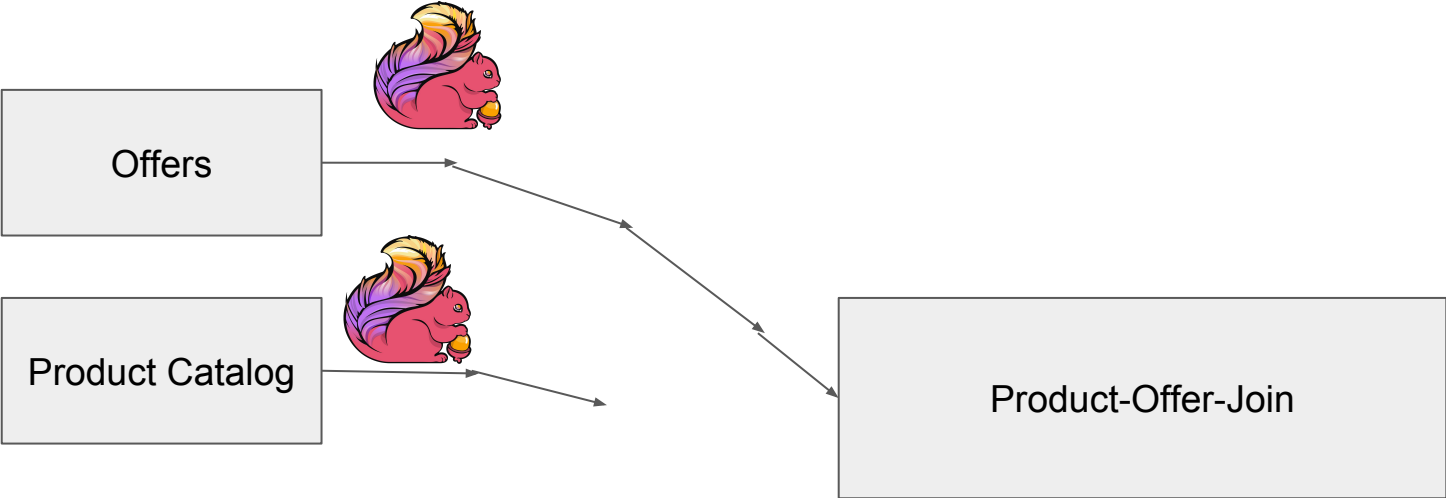


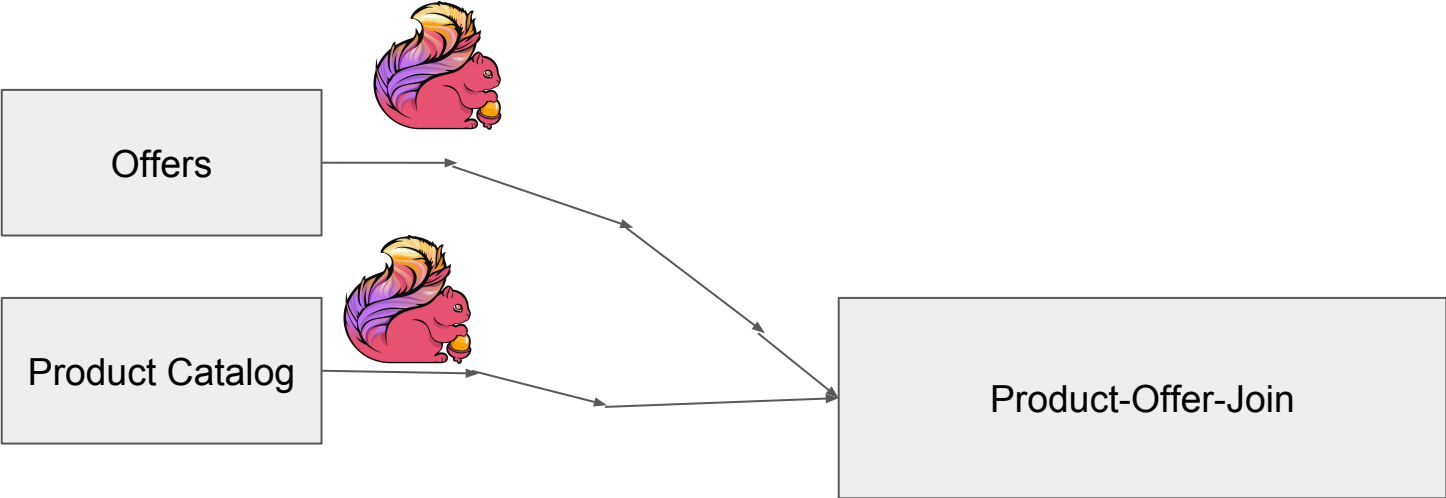
Product Catalog



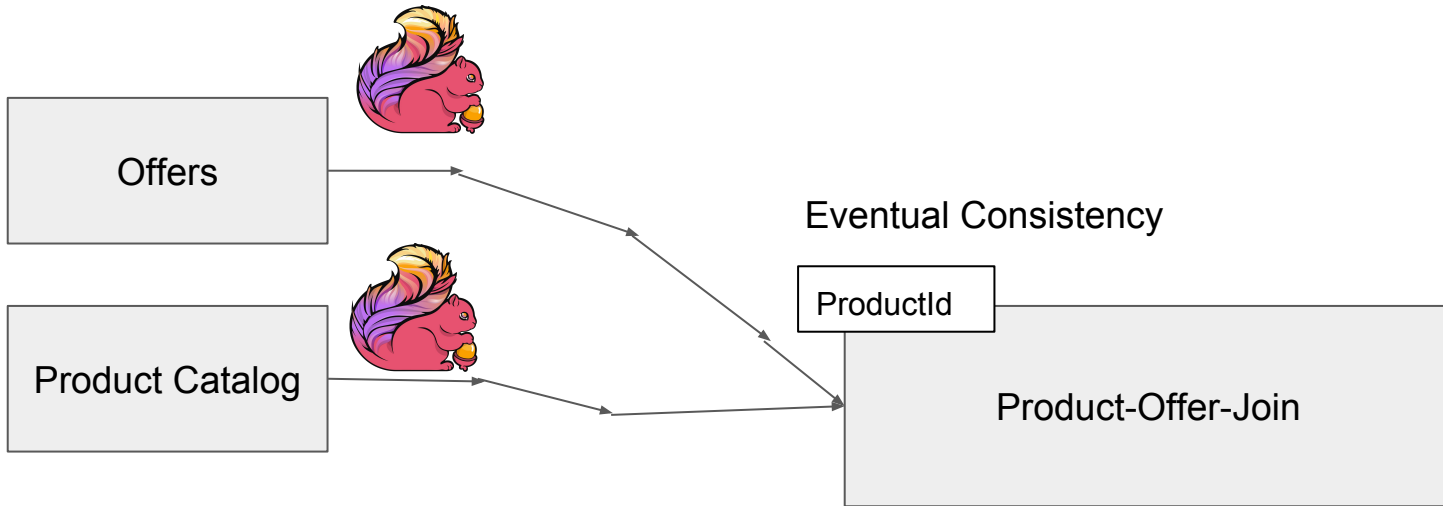
Product-Offer-Join

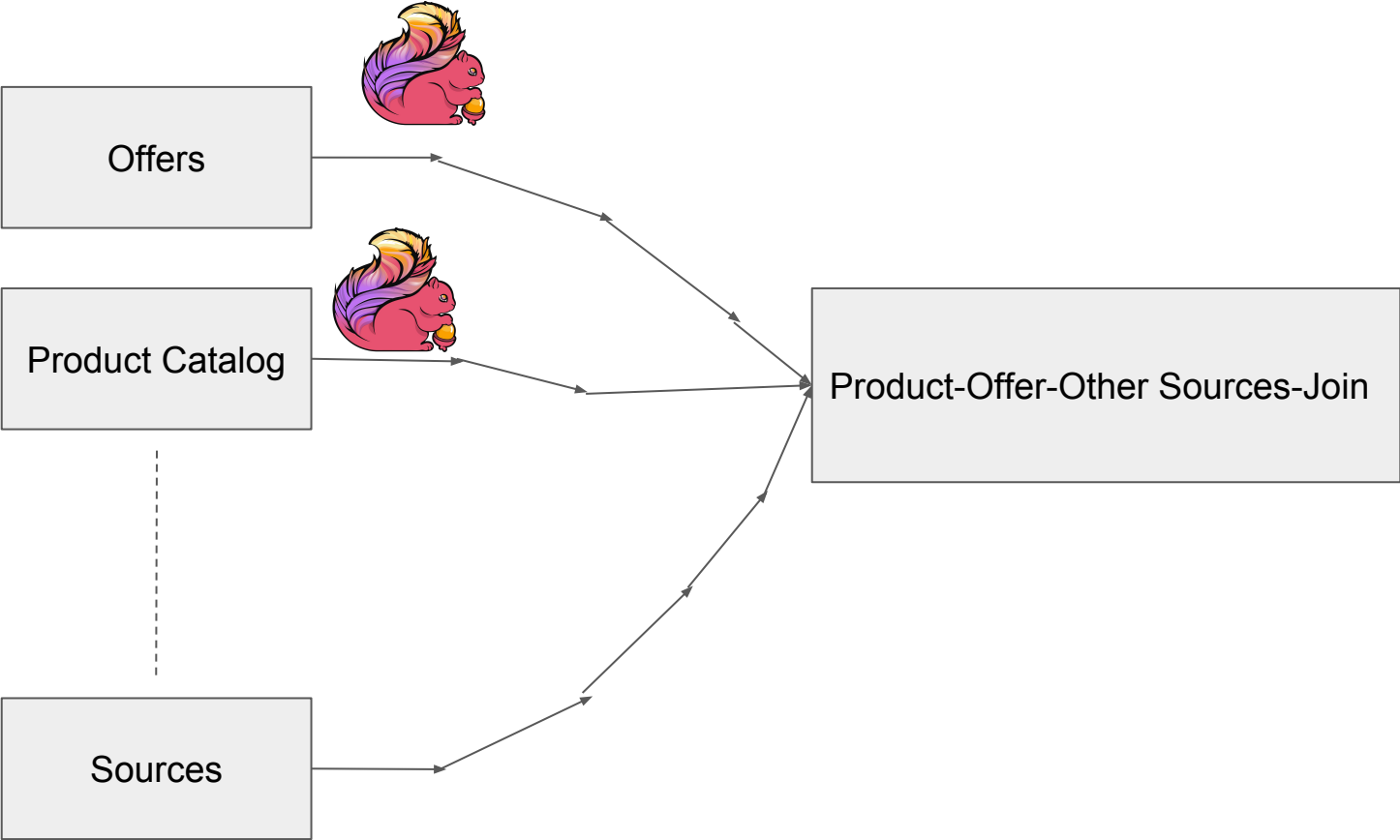


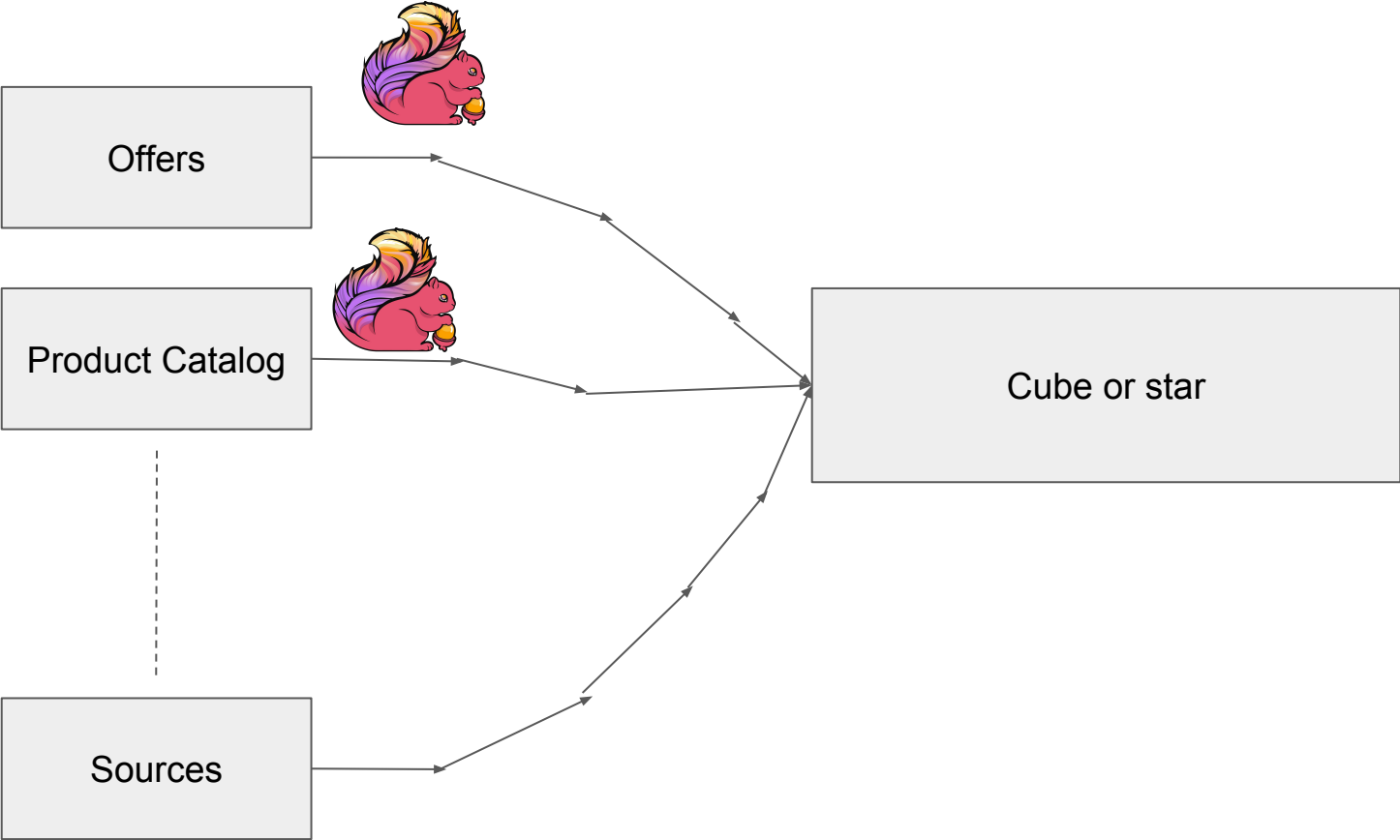


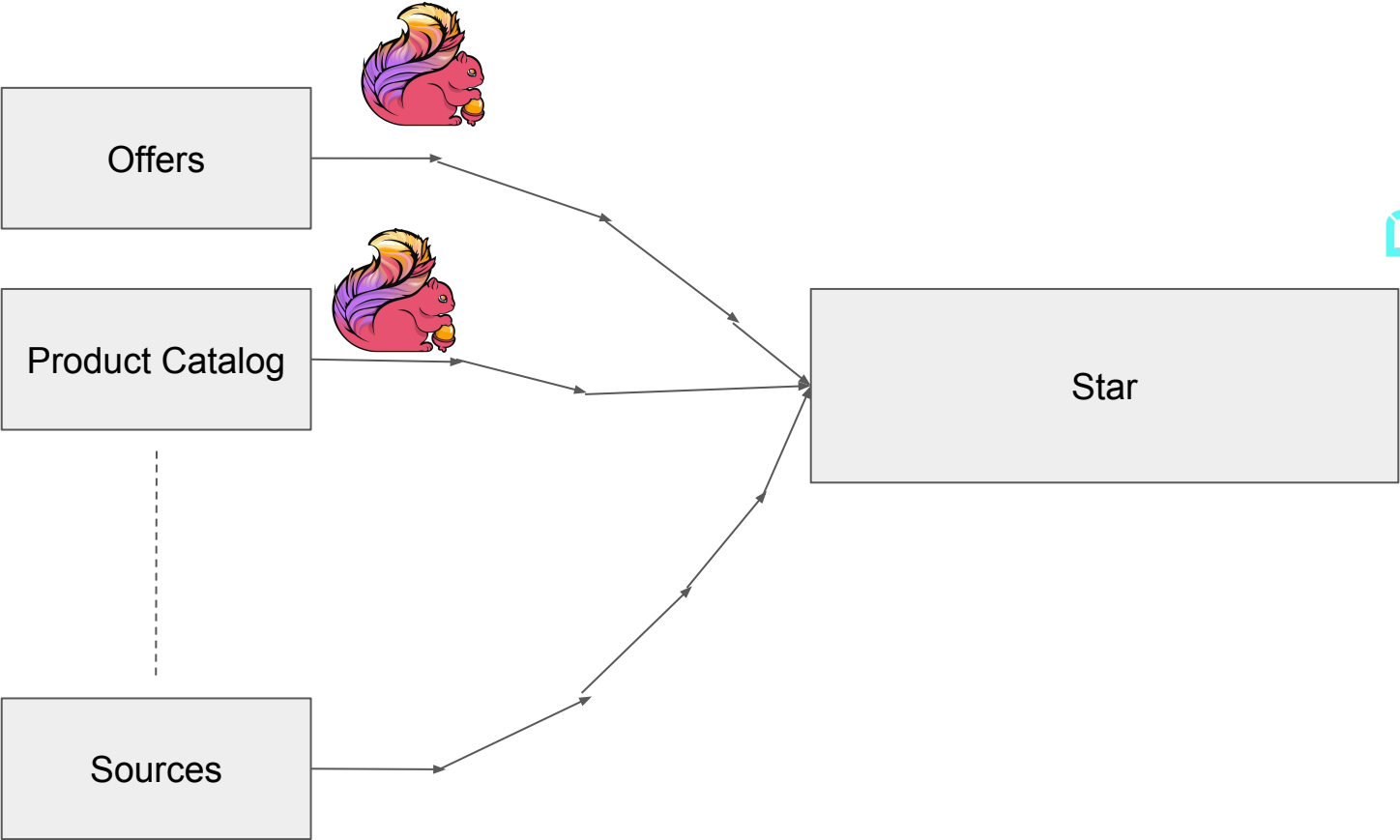












Can we automate this?

# Can we automate this?

Yes, We can.

# Can we automate this?

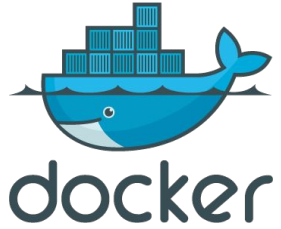
```
cube_builder
  .from(
    table("productoffer_tst_public_v1.0_SellingOffers_versions")
  )
  .on(
    key("f:GlobalId", key -> new StringBuilder(key).reverse().toString())
  )
  .lookUp(
    key("f:GlobalId"),
    table("financecategory_tst_public_v1_ProductFinanceCategoryCurrents")
  )
  .to(
    table("final_join_version"),
    table("reverse_index_lookup", key("f:GlobalId"), columns("f:OfferId")),
    table("final_join_version1", columns("f:SellingOfferData.ListPrice"))
  )
  .build()
  .execute();
```

# Operational Aspect

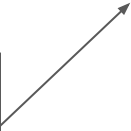
Build



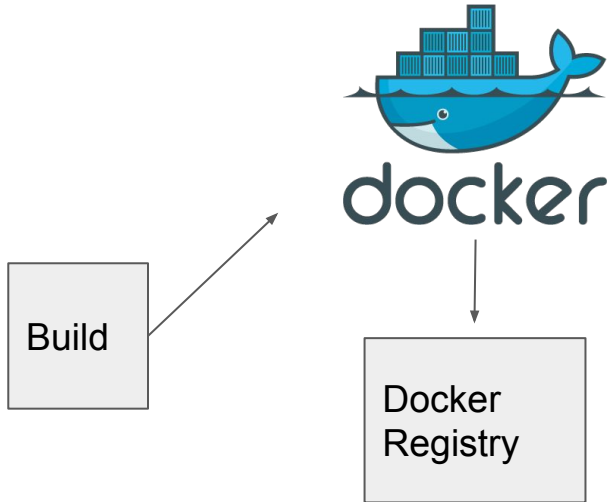
# Operational Aspect



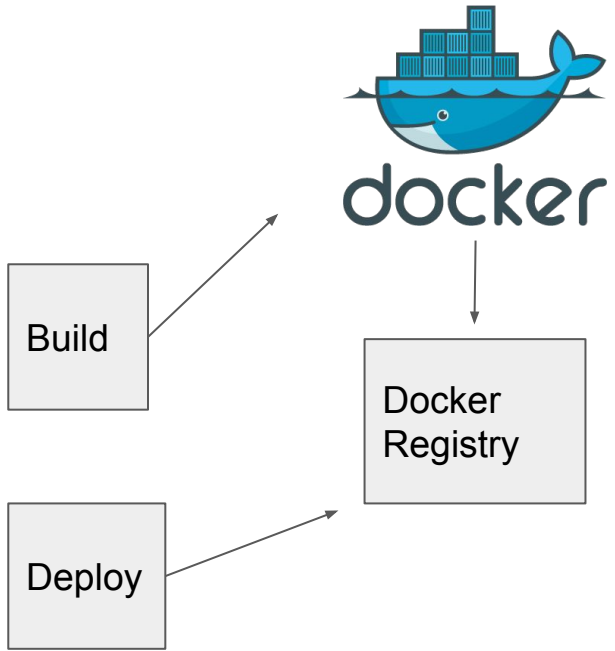
Build



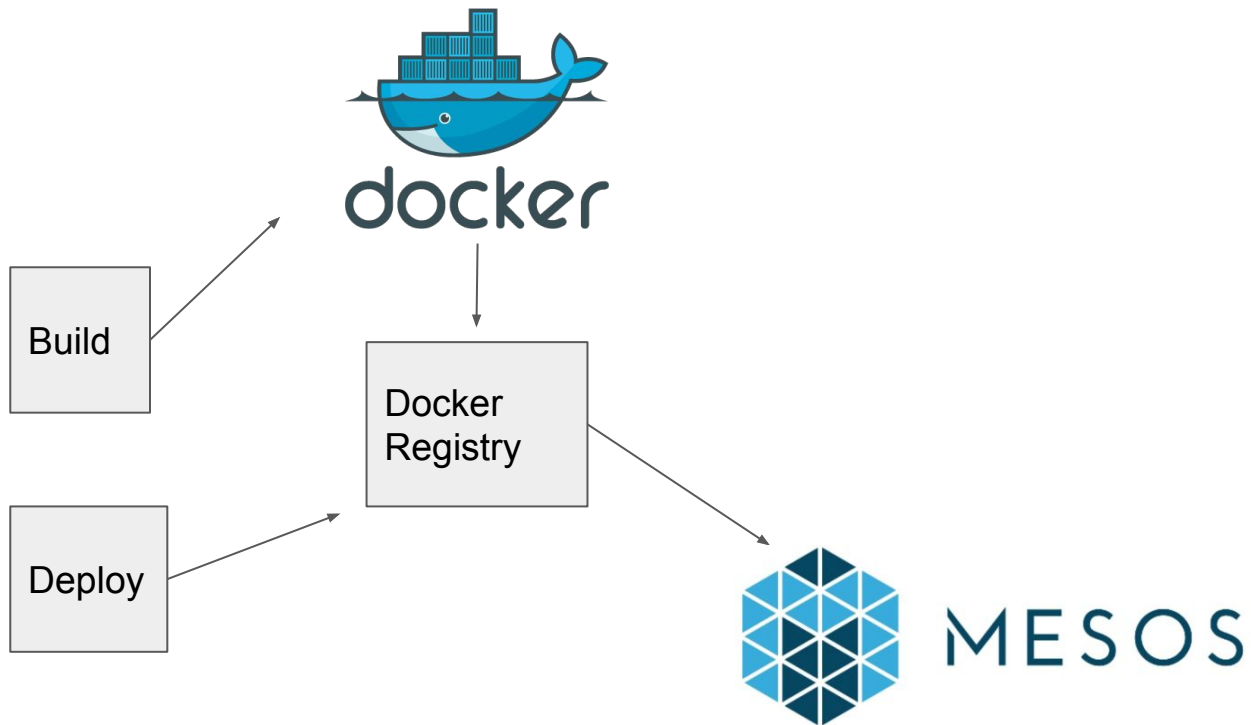
# Operational Aspect



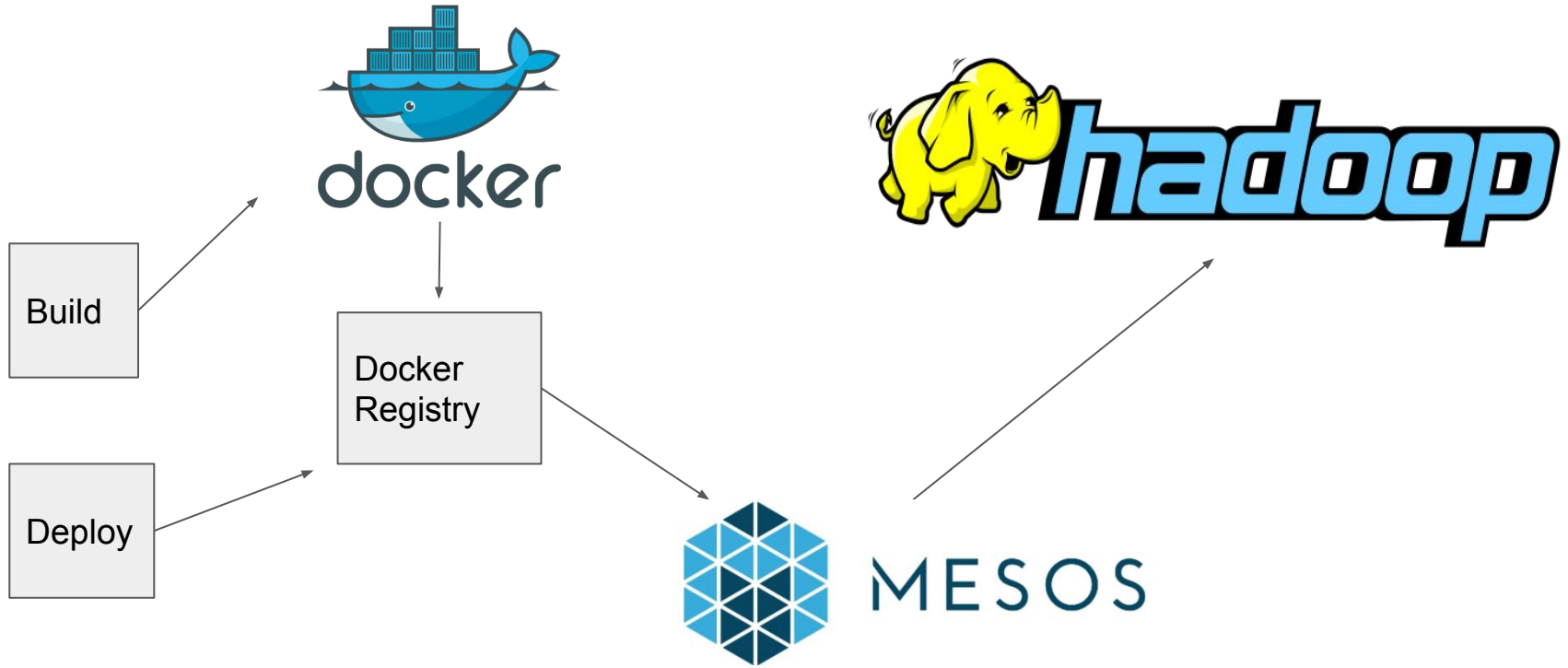
# Operational Aspect



# Operational Aspect



# Operational Aspect



# Lessons learned

- Dedicated team for hadoop
- Think not tools but how to solve problems
- Flink can be flinky
- Frameworks are out there
- Kylin
- Think infrastructure too

