FLINK FORWARD # ASIA 2022 ONLINE | 实时即未来

FeatHub: 流批一体的实时特征工程平台

林东

阿里巴巴高级技术专家
Apache Flink/Kafka committer

01 为什么需要FeatHub

02 FeatHub架构和概念

03 FeatHub API展示

01 为什么需要FeatHub

目标场景

需要Python环境的
数据科学家

生成实时特征

需要开源方案
支持多云部署

实时特征工程的痛点

- 开发难度高
  - 特征穿越
- 部署难度高
  - 需要手动翻译
  - 性能压力大
- 监控难度高
  - 特征分布变化
- 分享难度高
  - 开发工作重复

point-in-time correct语义

多版本特征

时间	用户ID	最近2分钟点击数
5:00	1	5
6:00	1	10
7:00	1	6

样本数据

时间	用户ID	Label
5:02	1	1
6:03	1	0
6:10	1	1
7:05	1	0

错误的join

训练数据

时间	用户ID	最近2分钟点击数	Label
5:02	1	6	1
6:03	1	6	0
6:10	1	6	1
7:05	1	6	0

point-in-time correct语义

多版本特征

时间	用户ID	最近2分钟点击数
5:00	1	5
6:00	1	10
7:00	1	6

样本数据

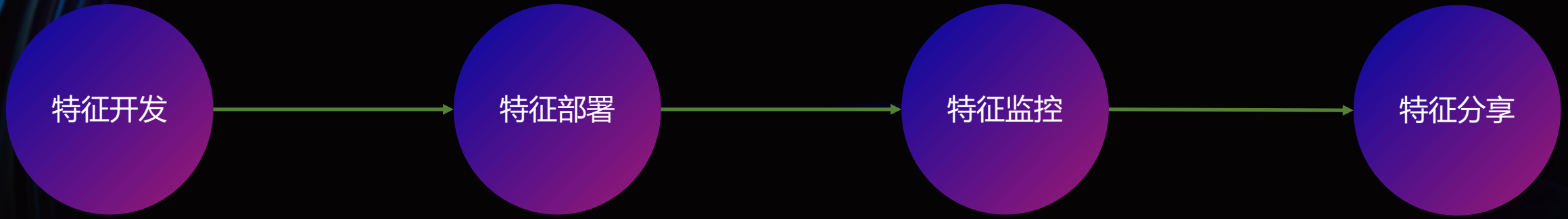
时间	用户ID	Label
5:02	1	1
6:03	1	0
6:10	1	1
7:05	1	0

正确的join

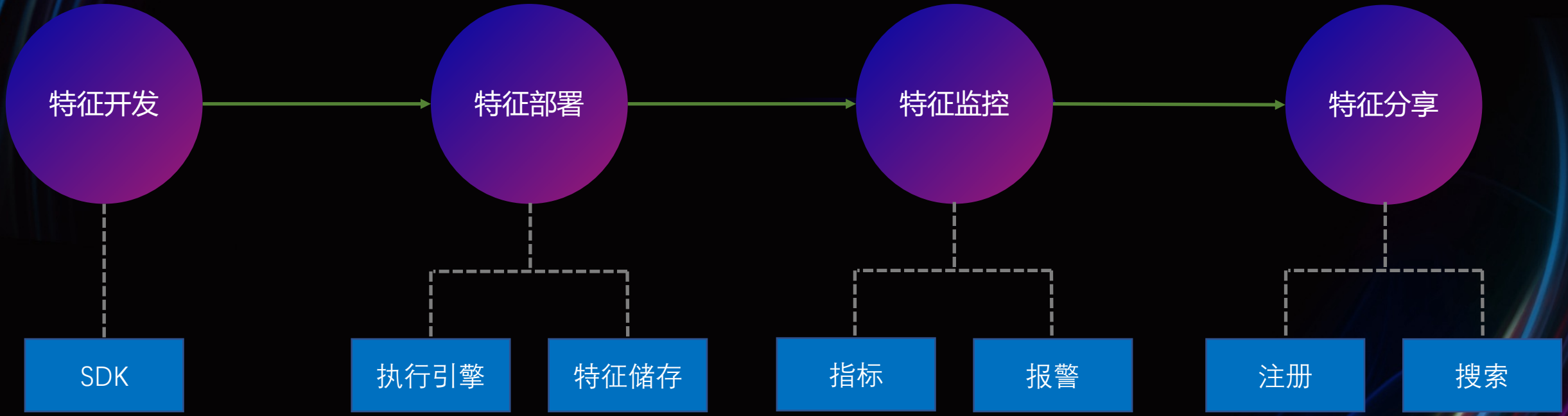
训练数据

时间	用户ID	最近2分钟点击数	Label
5:02	1	**5**	1
6:03	1	**10**	0
6:10	1	**10**	1
7:05	1	6	0

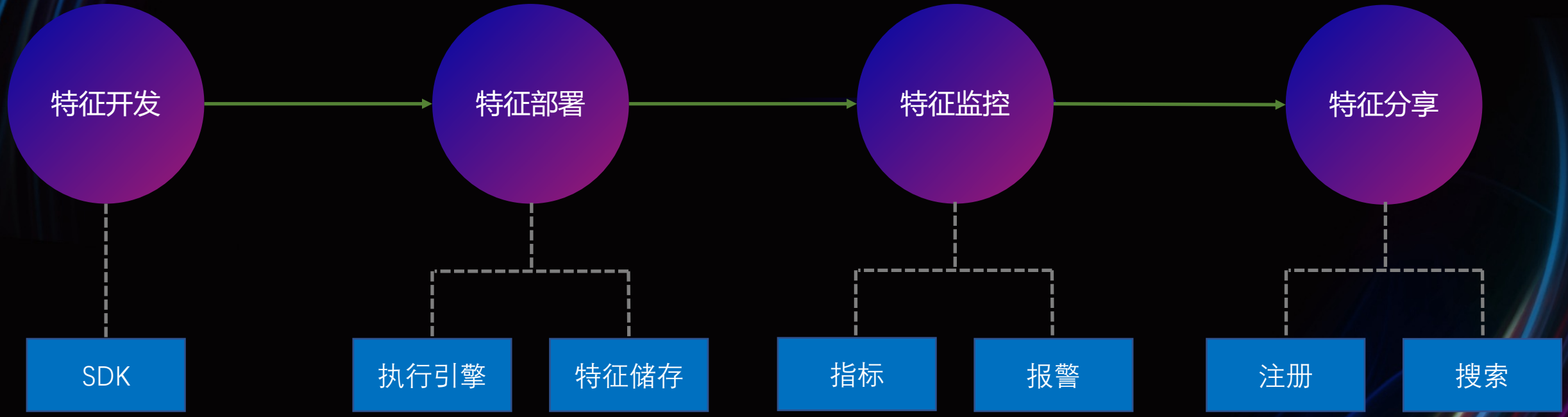
Feature Store的核心场景



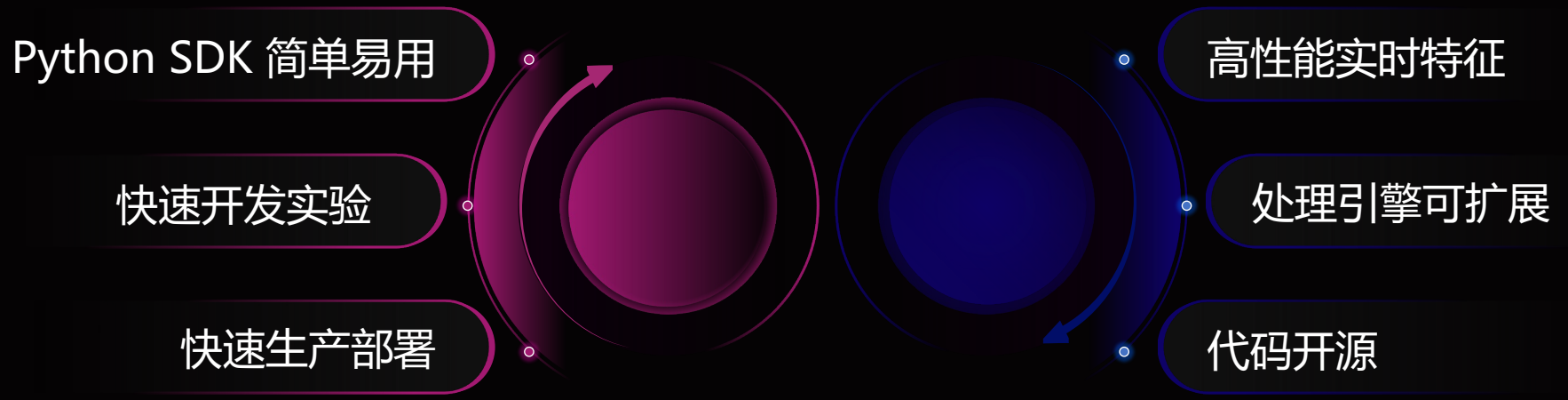
Feature Store的核心场景



Feature Store的核心场景

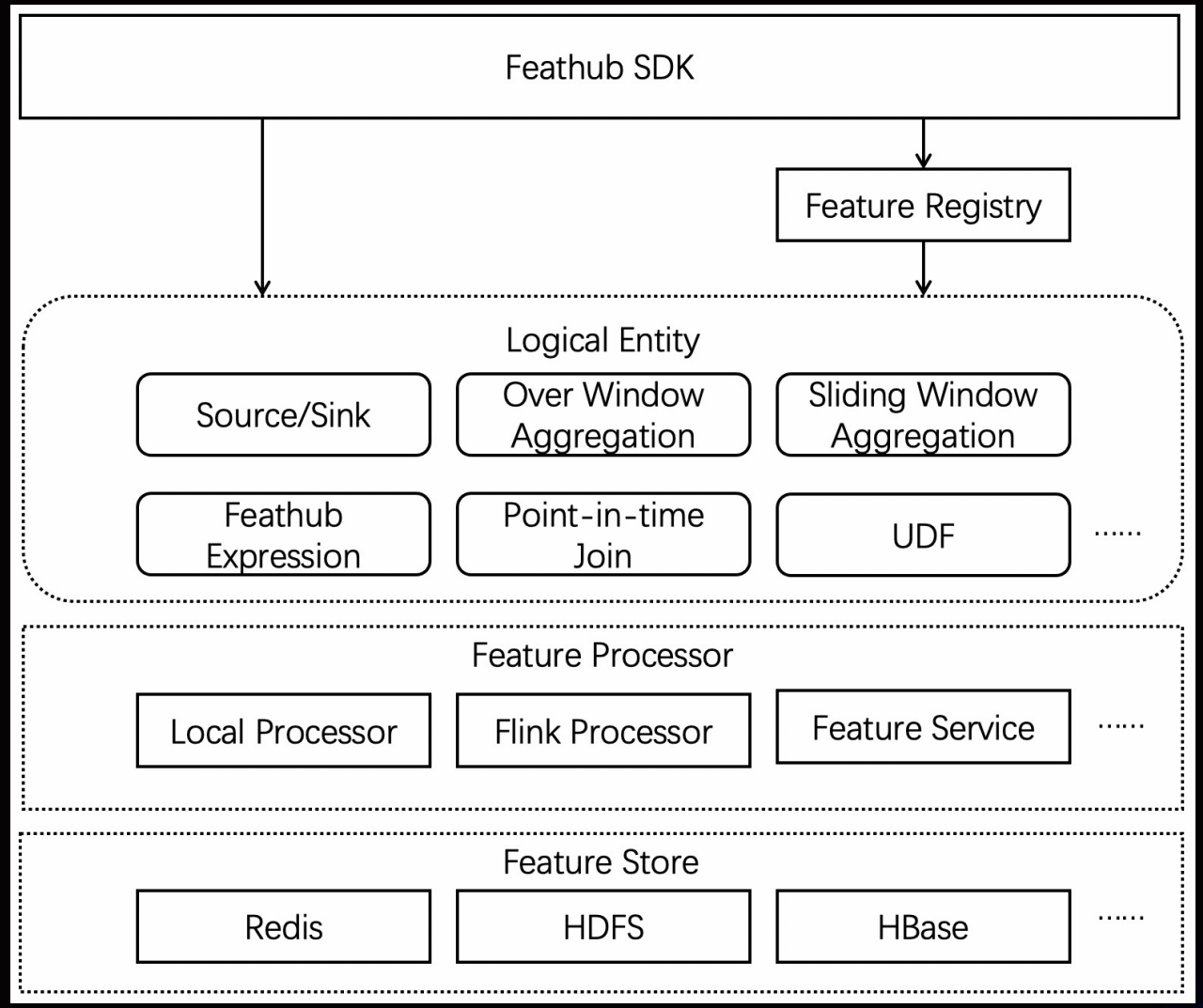


为什么需要FeatHub

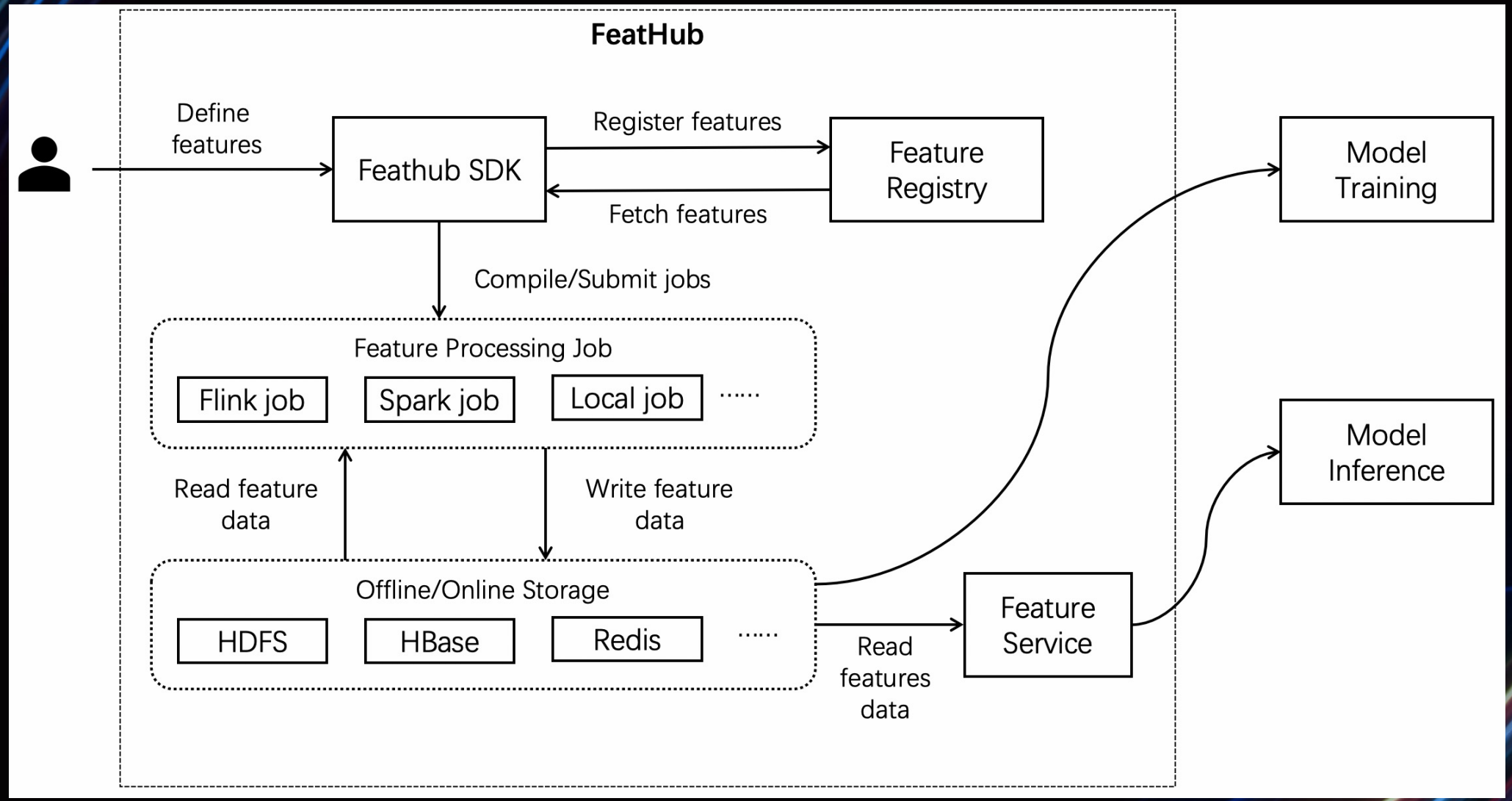


02 FeatHub架构和概念

架构

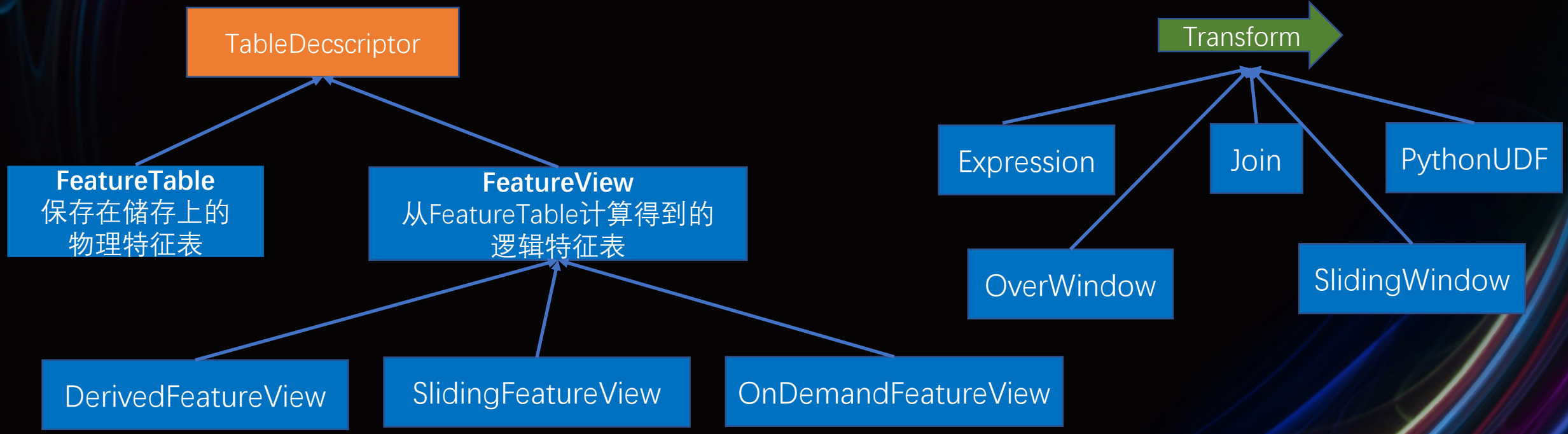

Feathub SDK
Feature Registry
Logical Entity
Source/Sink
Over Window Aggregation
Sliding Window Aggregation
Feathub Expression
Point-in-time Join
UDF
.....
Feature Processor
Local Processor
Flink Processor
Feature Service
.....
Feature Store
Redis
HDFS
HBase
.....

架构 (续)


FeatHub
Define features
Feathub SDK
Register features
Fetch features
Feature Registry
Compile/Submit jobs
Feature Processing Job
Flink job
Spark job
Local job
.....
Read feature data
Write feature data
Offline/Online Storage
HDFS
HBase
Redis
.....
Read features data
Feature Service
Model Training
Model Inference

核心概念


TableDecscriptor
Transform
TableDecscriptor
Transform
TableDecscriptor


TableDecscriptor
FeatureTable
保存在储存上的
物理特征表
FeatureView
从FeatureTable计算得到的
逻辑特征表
DerivedFeatureView
SlidingFeatureView
OnDemandFeatureView
Transform
Expression
Join
PythonUDF
OverWindow
SlidingWindow

核心概念 (续)

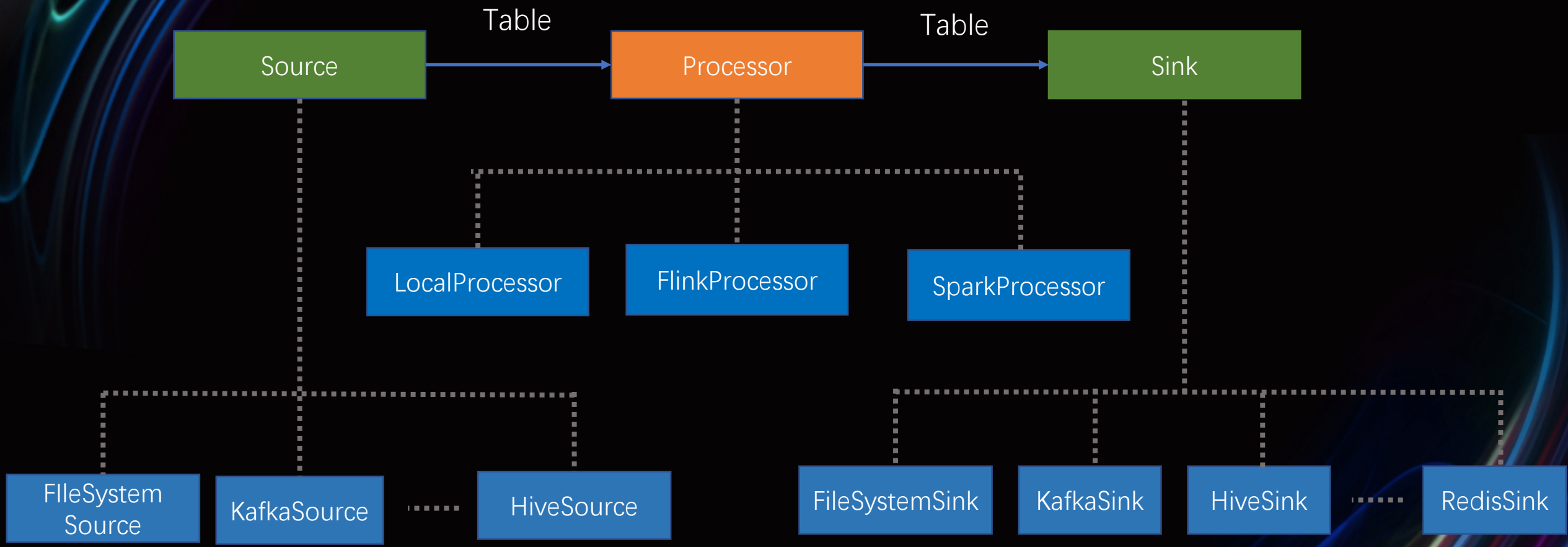


Table
Table
Source
Processor
Sink
LocalProcessor
FlinkProcessor
SparkProcessor
FlleSystem Source
KafkaSource
HiveSource
FileSystemSink
KafkaSink
HiveSink
RedisSink

03 FeatHub API展示

特征计算功能

特征拼接

```
f_price = Feature(
    name="price",
    dtype=types.Float32,
    transform=JoinTransform(
        table_name="price_updates",
        feature_name="price"
    ),
    keys=["item_id"],
)
```

Over窗口聚合

```
f_total_payment_last_two_minutes = Feature(
    name="total_payment_last_two_minutes",
    dtype=types.Float32,
    transform=OverWindowTransform(
        expr="item_count * price",
        agg_func="SUM",
        window_size=timedelta(minutes=2),
        group_by_keys=["user_id"]
    )
)
```

滑动窗口聚合

```
f_total_payment_last_two_minutes = Feature(
    name="total_payment_last_two_minutes",
    dtype=types.Float32,
    transform=SlidingWindowTransform(
        expr="item_count * price",
        agg_func="SUM",
        window_size=timedelta(minutes=2),
        step_size=timedelta(minutes=1),
        group_by_keys=["user_id"]
    )
)
```

内置函数调用

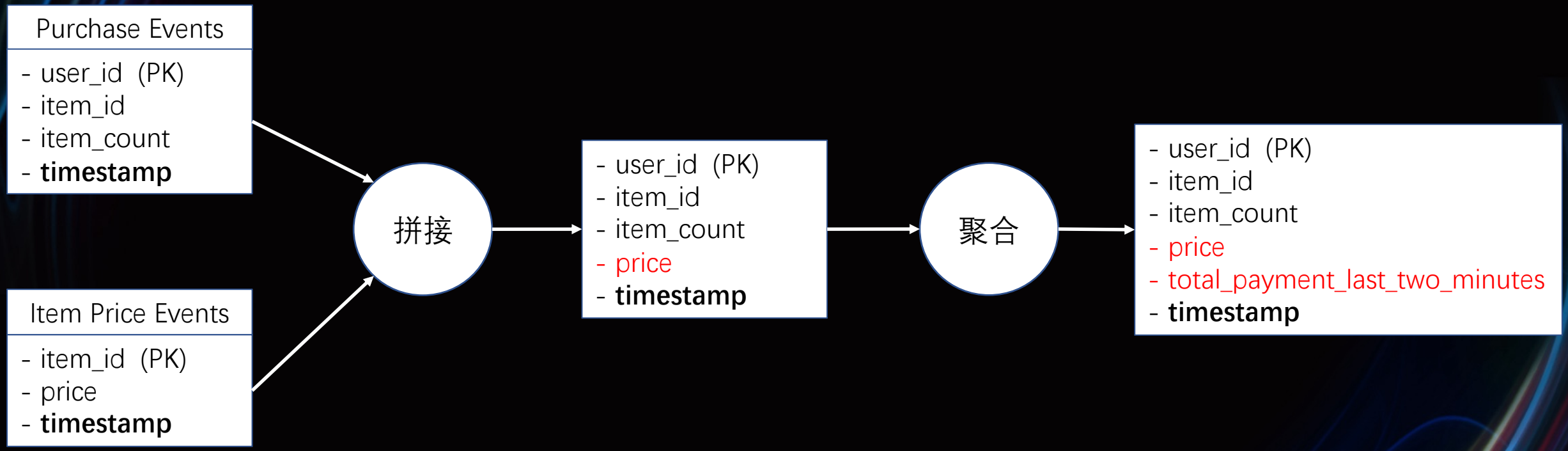
```
f_trip_time_duration = Feature(
    name="f_trip_time_duration",
    dtype=types.Int32,

    transform=
      "UNIX_TIMESTAMP(taxi_dropoff_datetime) -
       UNIX_TIMESTAMP(taxi_pickup_datetime)",
)
```

Python UDF调用

```
f_lower_case_name = Feature(
    name="lower_case_name",
    dtype=types.String,
    transform=PythonUdfTransform(lambda row: row["name"].lower()),
)
```

样例场景

生成机器学习训练数据集



样例代码

创建FeatHub Client

```
client = FeathubClient(
    config={
        "processor": {
            "processor_type": "flink",
            "flink": {
                "rest.address": "localhost",
                "rest.port":" 8081"
            },
        },
        ...
    }
)
```

创建Source

```
purchase_events_source = FileSystemSource(
    name="purchase_events",
    path="/tmp/data/purchase_events.json",
    data_format="json",
    schema=purchase_events_schema,
    timestamp_field="timestamp",
    timestamp_format="%Y-%m-%d %H:%M:%S",
)

item_price_events_source = FileSystemSource(
    name="item_price_events",
    path="/tmp/data/item_price_events.json",
    data_format="json",
    schema=item_price_events_schema,
    keys=["item_id"],
    timestamp_field="timestamp",
    timestamp_format="%Y-%m-%d %H:%M:%S",
)
```

创建FeatureView

```
f_total_payment_last_two_minutes = Feature(
    name="total_payment_last_two_minutes",
    dtype=types.Float32,
    transform=OverWindowTransform(
        expr="item_count * price",
        agg_func="SUM",
        window_size=timedelta(minutes=2),
        group_by_keys=["user_id"],
    ),
)

purchase_events_with_features =
DerivedFeatureView(
    name="purchase_events_with_features",
    source=purchase_events_source,
    features=[
        "item_price_events.price",
        f_total_payment_last_two_minutes,
    ],
    keep_source_fields=True,
)
```

样例代码（续）

获取特征到本地Pandas DF

```
result_table = client.get_features(
    features=purchase_events_with_features
)

result_table_df = result_table.to_pandas()

print(result_table_df)
```

创建Sink

```
hdfs_sink = FileSystemSink(
    path="/tmp/data/output.json",
    data_format="CSV"
)
```

输出特征到离线储存HDFS

```
result_table.execute_insert(
    sink=hdfs_sink,
    allow_overwrite=True
)
```

FeatHub性能优化

- 滑动窗口聚合的特征仅在特征数值变化时输出数据，减少网络带宽使用
- 滑动窗口聚合的特征在窗口为空的时候输出数据，无需下游周期性扫描数据来移除过期数据
- 提供高性能内置UDAF (e.g. VALUE_COUNTS)，减少稀疏特征的网络带宽使用量
  - E.g. 最近一分钟每个用户点击每种产品分类的数量
- 滑动窗口聚合的特征共用状态，减少计算量和内存使用 (尚未完成)
  - E.g. 同时计算最近1分钟/5分钟/10分钟/30分钟内的用户点击数
- ...

FeatHub未来工作

- 完善LocalProcessor和FlinkProcessor的功能和性能
- 支持常用的离线/在线储存 (e.g. HDFS, Redis)
- 支持对接Notebook
- 提供可视化UI来访问特征元数据 (e.g. 特征定义，特征血缘)
- 支持特征质量监控和报警
- 支持Spark作为处理引擎
- 支持特征授权，鉴权，审计等企业级功能
- ...

欢迎加入FeatHub社区

FeatHub代码库：https://github.com/alibaba/feathub

FeatHub代码样例：https://github.com/flink-extended/feathub-examples

FLINK FORWARD # ASIA 2022 ONLINE | 实时即未来

THANK YOU

谢 谢 观 看