RADICALBIT
radicalbit.io

Blending Event Stream Processing with Machine Learning using the Kafka Ecosystem

Data Council, Barcelona, Oct 2nd, 2019

DISCLAIMER

A bit of me.

Andrea Spina

Head of R&D @Radicalbit

andrea.spina@radicalbit.io

@Spina89

andrea-spina

RADICALBIT PRODUCTS

Radicalbit is a highly specialized software firm, founded in Milan, in 2015, focused on the design and development of products dedicated to Event Stream Processing solutions, daily working to combine streaming technologies, Machine Learning and AI with a self-service approach.

Radicalbit Natural Analytics

RNA is a platform offering the most advanced self-service capabilities for Data Integration, Data Governance, Data Preparation and Data Visualization over streaming based architectures. It offers a complete set of features aimed to manage every step of the Data Lifecycle: from ingestion to visualization.

Natural Series Database

NSDb is a storage solution conceived having streaming real-time analytics in mind. It fits perfectly the read side of Kappa Architectures (or for systems based on Command Query Responsibility Segregation pattern). The idea is to store metrics and to bind directly the incoming indexed data to the final users, thanks to pushing technologies like WebSocket.

radicalbit.io

DISCLAIMER (AGAIN)

During this talk, you're going to listen about some **buzzwords**

- Event Stream Processing
- Machine Learning

You might also hear about topics you already know, and a few you might not ;)

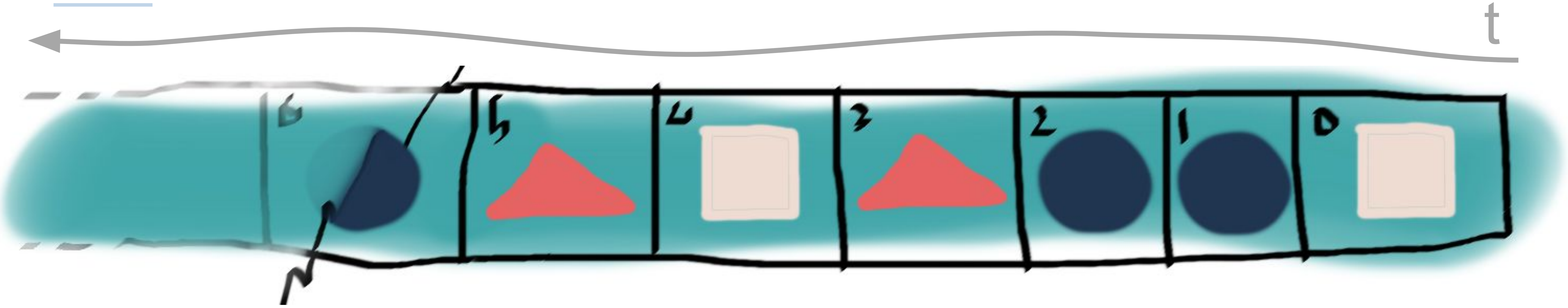
- Lambda v.s. Kappa architectures
- Machine Learning Logistics
- Online Machine Learning

AGENDA

1. Events Stream Processing
2. Machine Learning
3. Model Serving on Kafka
4. Online Learning on Kafka
5. Conclusion

Events Stream Processing

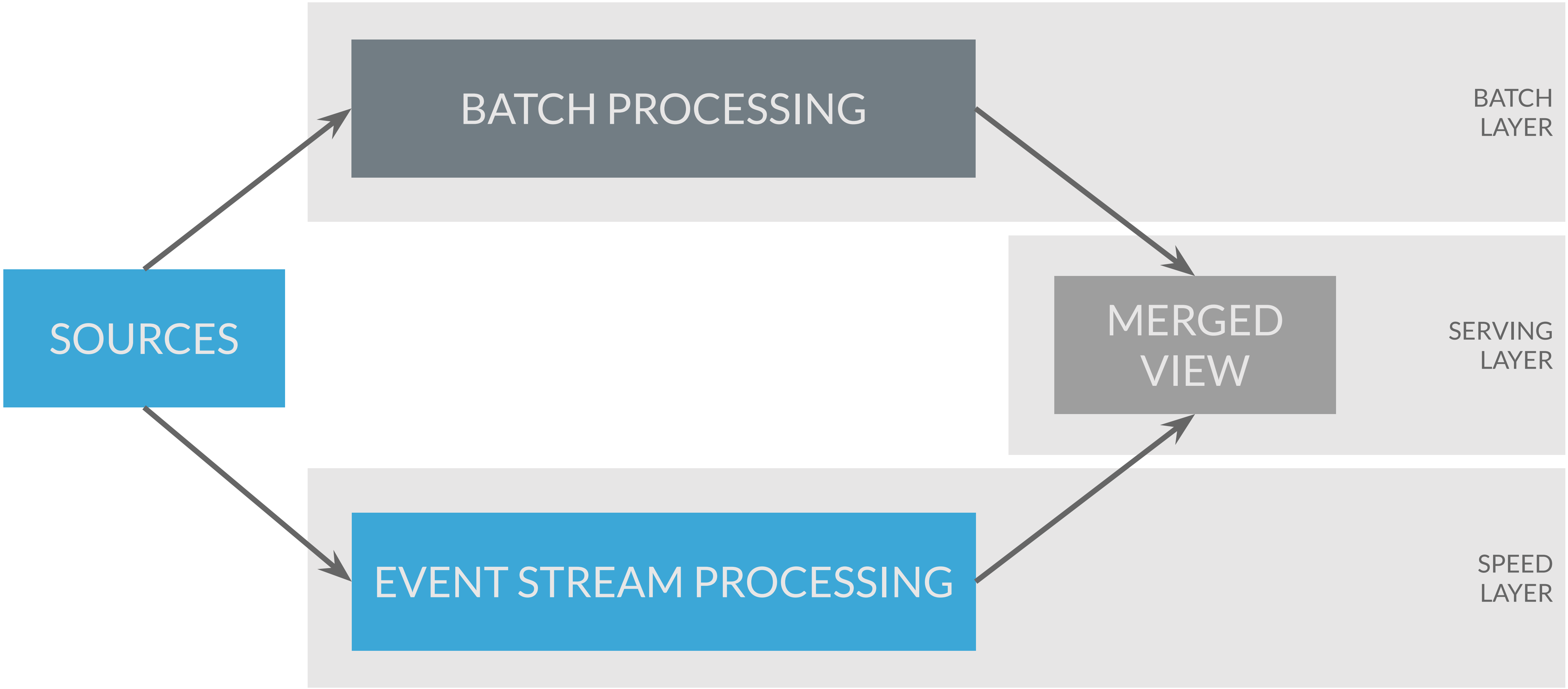
Data Streams



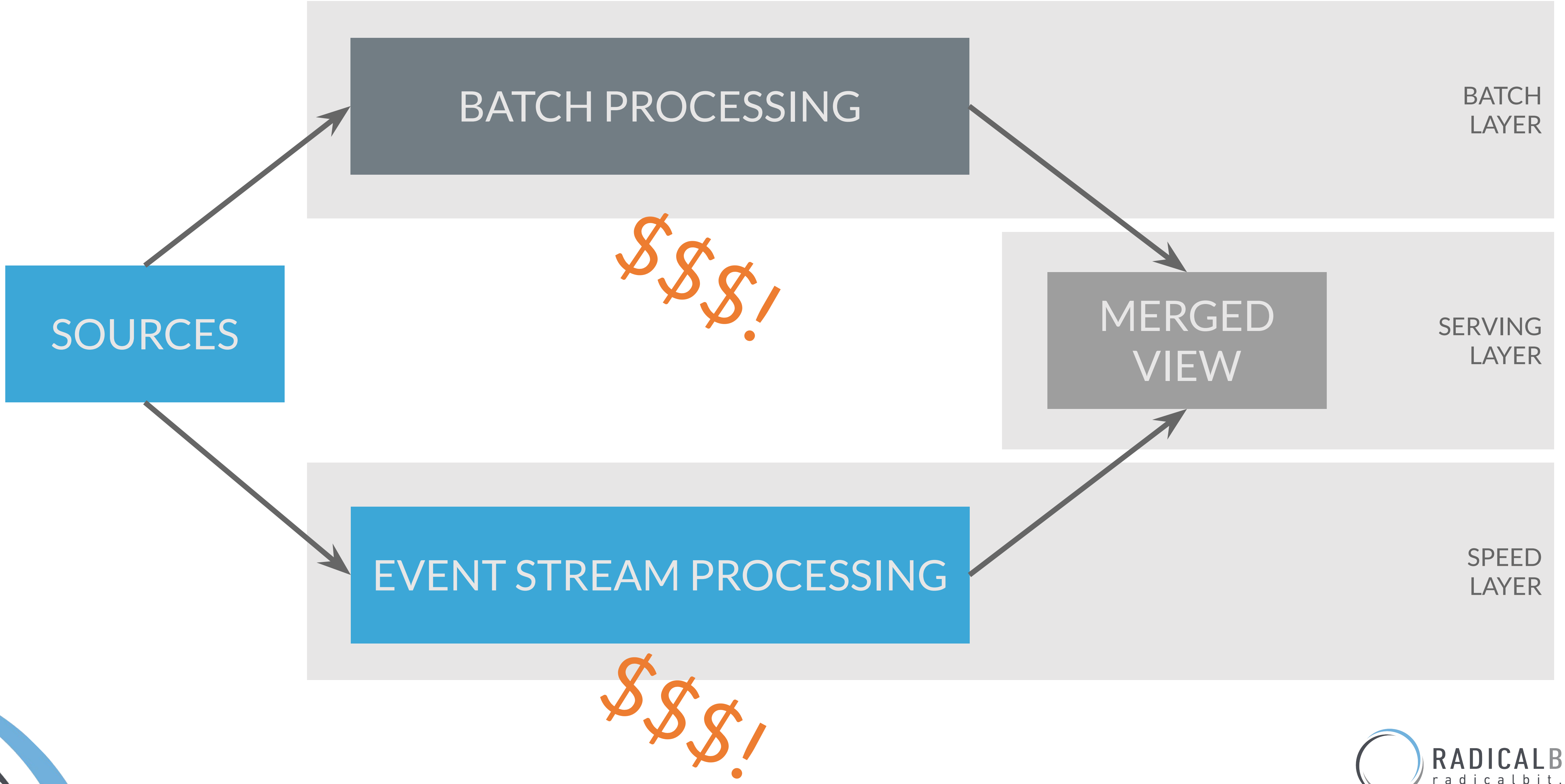
Unbounded
Immutable
Unknown

Batch is stream-*able*
Storable
Transformable

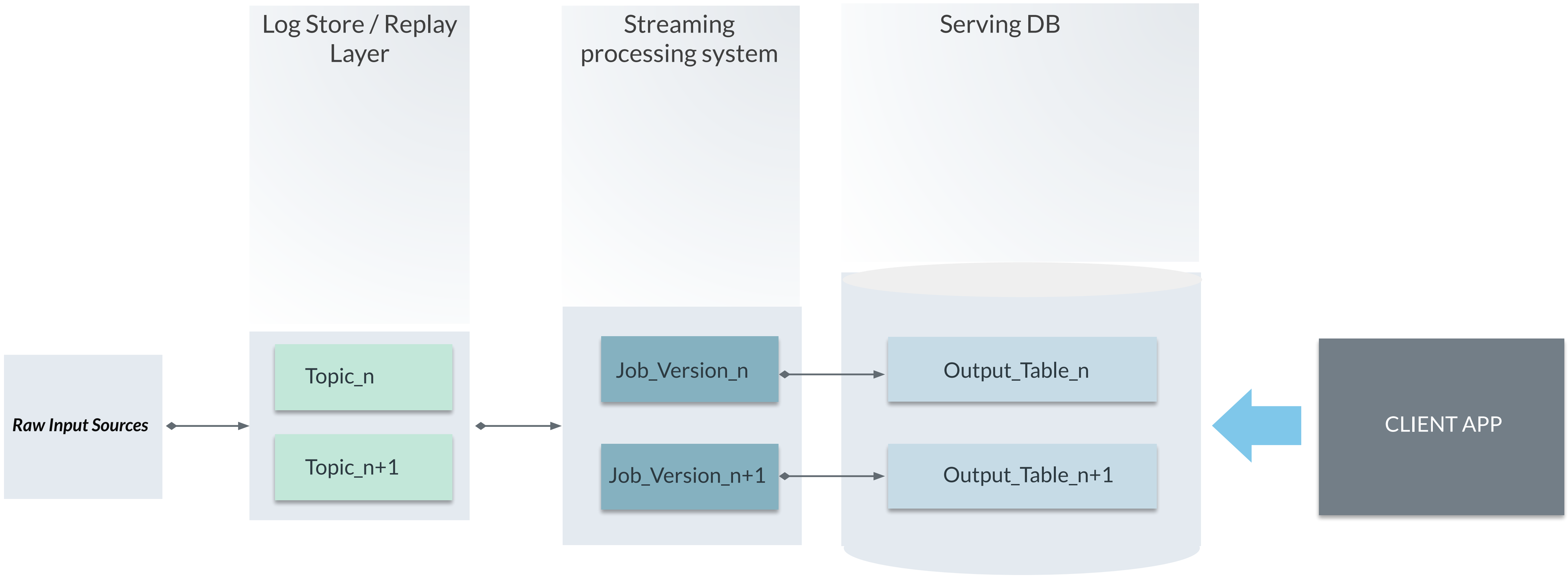
LAMBDA ARCHITECTURE



LAMBDA ARCHITECTURE ISSUE

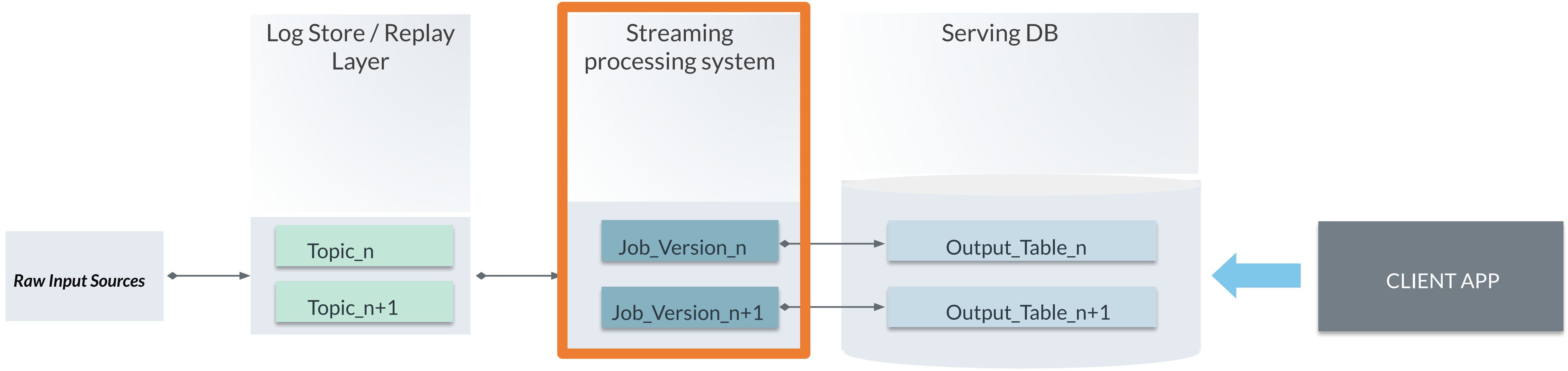


KAPPA ARCHITECTURE



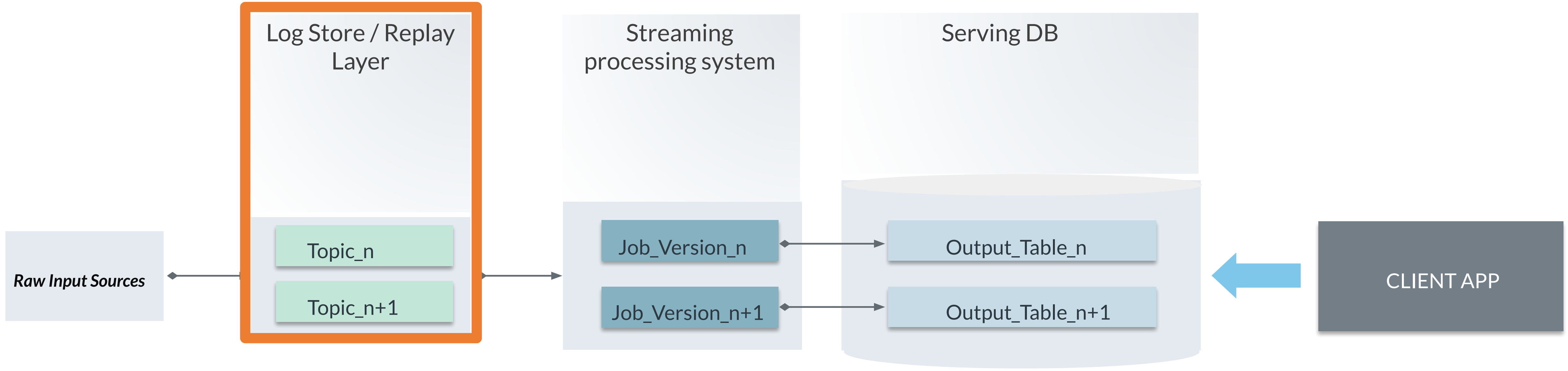
KAPPA ARCHITECTURE: REQUIREMENTS

1. Low Latency / High throughput



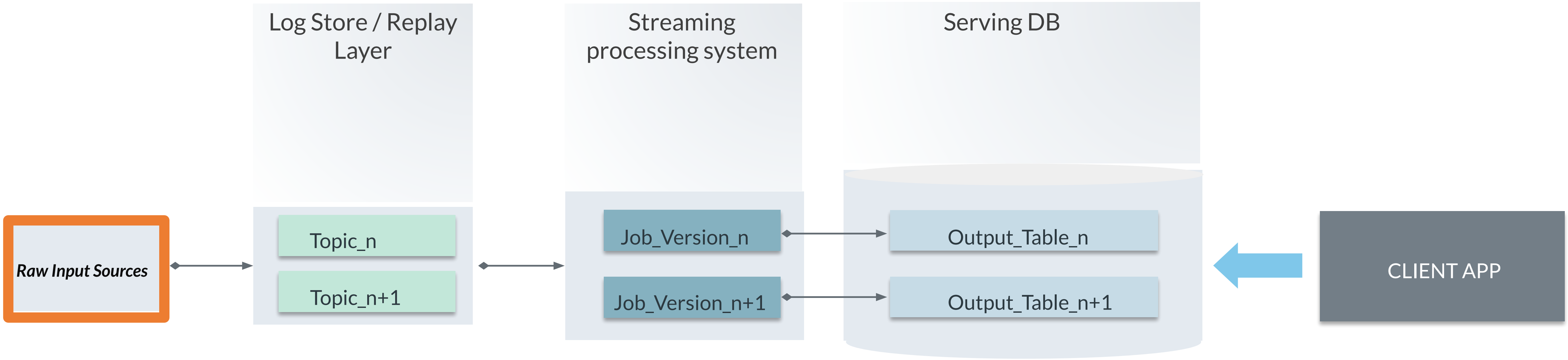
KAPPA ARCHITECTURE: REQUIREMENTS

1. Low Latency / High throughput
2. Agile *data-reprocessing* method



KAPPA ARCHITECTURE: REQUIREMENTS

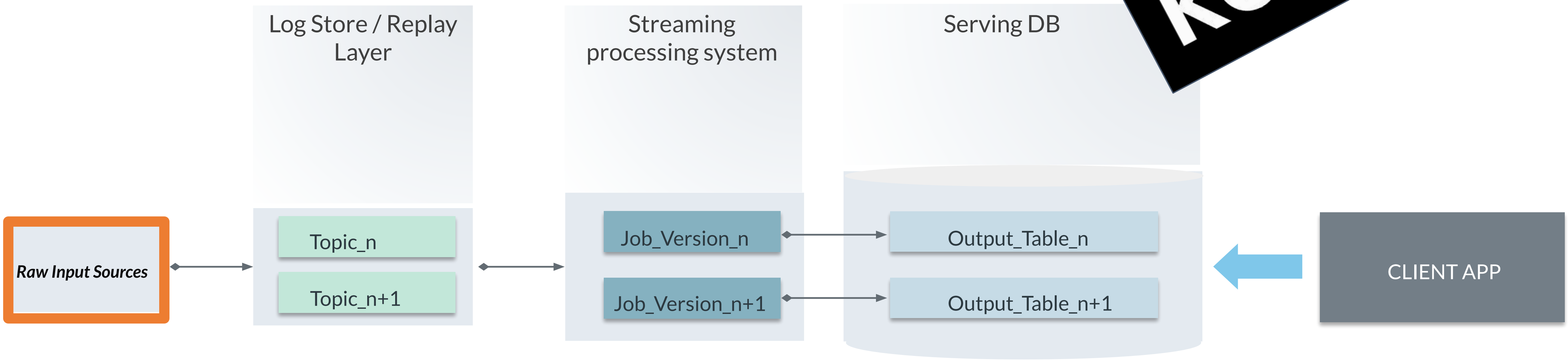
1. Low Latency / High throughput
2. Agile *data-reprocessing* method
3. Long-time retention message system



RADICALBIT
radicalbit.io

KAPPA ARCHITECTURE: REQUIREMENTS

1. Low Latency / High throughput
2. Agile *data-reprocessing* method
3. Long-time retention message system



KAFKA

" Apache Kafka is a distributed streaming platform.

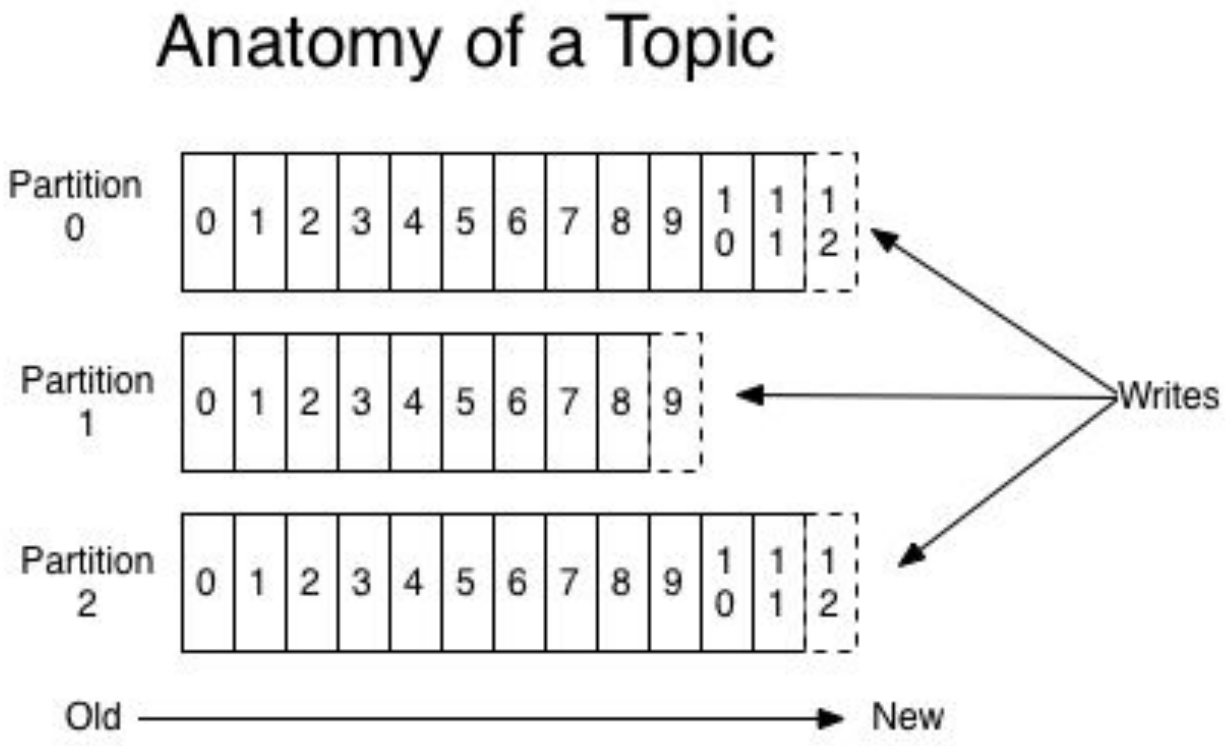
- *Publish and subscribe to streams of records, similar to a message queue or enterprise messaging system*

- *Store streams of records in a fault-tolerant durable way*

- *Process streams of records as they occur "*

KAFKA OFFSETS

*Data reprocessing means "**resetting offsets**"*



RADICALBIT
radicalbit.io

KAFKA MODULES

1. Consumer API
2. Producer API
3. Connect API
4. **Streams API**

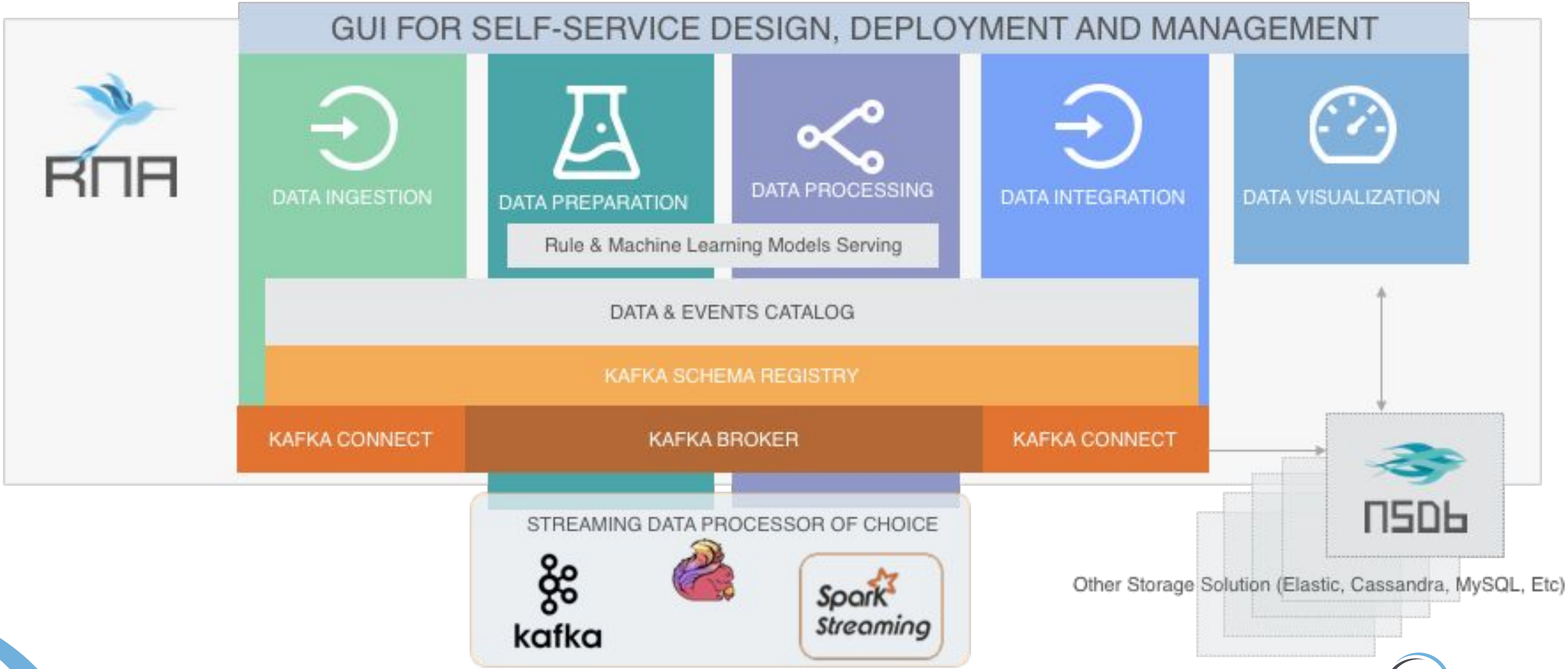
KAFKA AS STANDARD STREAMING ENABLER

"Kafka at the core of tens of thousands production use-cases"

Jay Kreps, Kafka Summit - New York, 2019

RNA AND THE KAPPA ARCHITECTURE

Radicalbit platform has been optimized to take full advantage of Kafka core features such as Kafka Connect, the Schema registry, and Kafka Streams but can be used to manage data pipelines also over Apache Flink or Spark Streaming with code portability



RADICALBIT
radicalbit.io

Blending Machine Learning with Streaming

TARGETED ML TASKS

- Models Serving
- Online Machine Learning

Streaming Models Serving

and the magic of machine learning logistics

RADICALBIT
radicalbit.io

STREAMING MODELS SERVING

- Serve models in a event stream processing architecture
- It's a **Machine learning logistics** issue (1)
    - Organisations need Data Scientists and Data Engineers
    - New Tools make it harder (2)
- Fragmented solution space
    - Framework based: Tensorflow Serving, Spark, Openscoring
    - Cloud based: Google, IBM, MS Azure, Amazon

(1) Ted Dunning & Ellen Friedman - Machine Learning Logistics - OREILLY

(2) Boris Lublinsky - Serving Machine Learning Models - OREILLY

RADICALBIT
radicalbit.io

SOLUTIONS TO ML ENTROPY

1. **STANDARD BASED**

   Define a **youNameIt**-independent format to represent a wide range of ML models

- PMML (PFA) - traditional learning
- ONNX - deep learning
- MLEAP - not a STD

```xml
<LocalTransformations>
    <DerivedField name="GroupedInputColor" dataType="integer"
        optype="categorical">
        <MapValues outputColumn="group" defaultValue="3" mapMissingTo="4">
            <FieldColumnPair column="color" field="VarInputColor" />
            <TableLocator>
                <Extension extender="ADAPA" name="TABLE_NAME"
                    value="GroupedInputColor" />
            </TableLocator>
        </MapValues>
    </DerivedField>
    <DerivedField name="VarColorGroup_1" dataType="integer" optype="categorical">
        <NormDiscrete value="1" field="GroupedInputColor" method="indicator" />
    </DerivedField>
    <DerivedField name="VarColorGroup_2" dataType="integer" optype="categorical">
        <NormDiscrete value="2" field="GroupedInputColor" method="indicator" />
    </DerivedField>
    <DerivedField name="VarColorGroup_3" dataType="integer" optype="categorical">
        <NormDiscrete value="3" field="GroupedInputColor" method="indicator" />
    </DerivedField>
    <DerivedField name="VarColorGroup_4" dataType="integer" optype="categorical">
        <NormDiscrete value="4" field="GroupedInputColor" method="indicator" />
    </DerivedField>
</LocalTransformations>
```

SOLUTIONS TO ML ENTROPY

2. **CONTAINER BASED**

   Creating containers wrapping environments natively aimed at models deployment

   Exposing a communication protocol for serving (usually a REST endpoint)

- Seldon core[1]
- Clipper[2]
- MLFlow[3]

(1) https://www.seldon.io/open-source/
(2) http://clipper.ai/
(3) https://mlflow.org/

PROS and CONS[1]

STANDARD BASED

PROS Performance, flexibility, many people are happy

CONS Adoption, algorithms

CONTAINER BASED

PROS Repeatability, adoption is not a problem, everybody is happy

CONS Performance depends on systems, devops competence

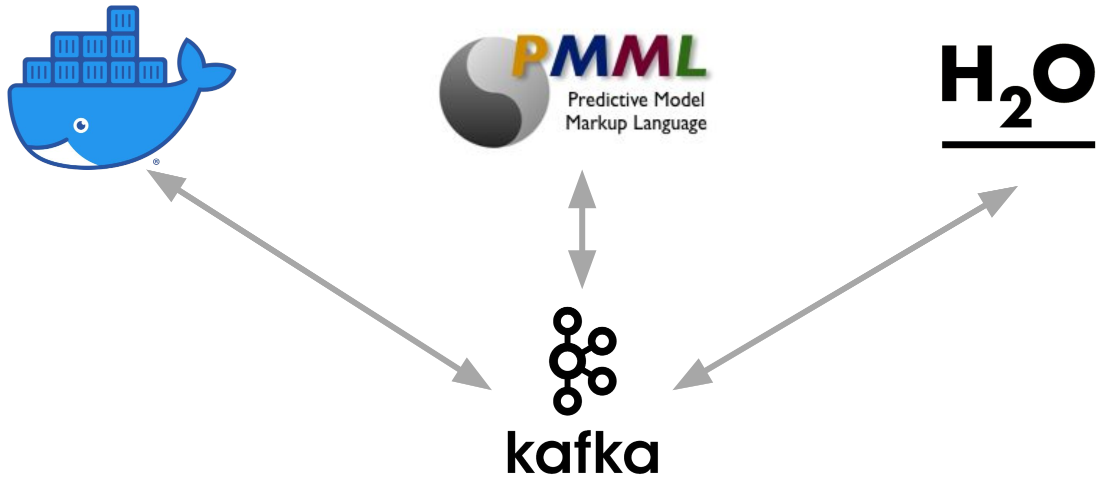
(1) https://qconsp.com/sp2018/system/files/presentation-slides/qconsp18-deployingml-may18-npentreath.pdf

SERVING AS A SERVICE WITH KAFKA

The goal

Attempting to serve seamless **Standards**, **Containers**, and **Tools** using Kafka

- No constraints about models deployment (it has not to be even a ML model!)
- It potentially has not to be even a ML model!

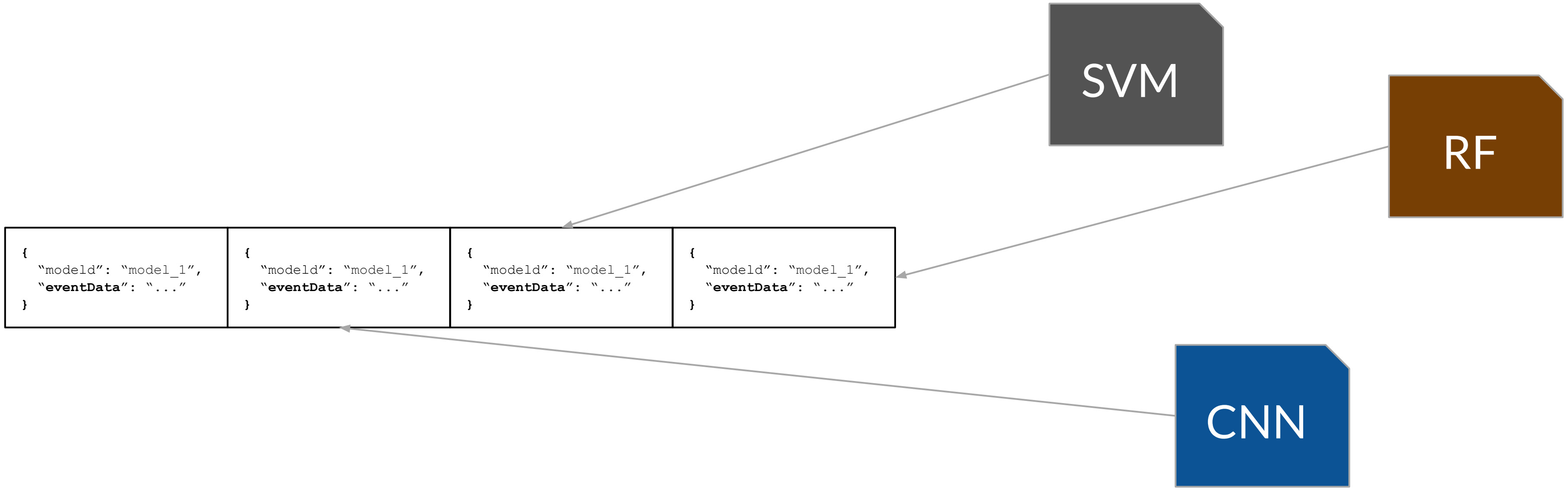


SERVING AS A SERVICE WITH KAFKA

Our predictive **k**-pipelines shall:

- **dynamically serve the evolution of trained models**
    - models often change in behavior during their long-lasting lifetime
    - updates

- apply simultaneously multiple models against the same stream, the same model to many streams



KAFKA STREAMS APIs

- Kafka Streams is not a DSPE, is a library[1]
- By Kafka Streams APIs, users define a processor topology
- Two API levels
  - `Kafka Streams DSL`
  - `Processor API`

(1) https://kafka.apache.org/23/documentation/streams/

SERVING AS A SERVICE: KS-H_2O Example

- Gartner 2019 magic quadrant for Machine Learning
- Most of the code is *open source*
- High support for algorithms
- H_2O flow

Figure 1. Magic Quadrant for Data Science and Machine Learning Platforms

Source: Gartner (January 2019)

Main features

- well-built Rest API layer
- **POJO** and **MOJO** formats + client library

RADICALBIT
radicalbit.io

KSH$_2$O - THE CONTROL STREAM

```
{
   …
   "id": "unsupervised_cusomers_1",
   "algorithm": "kmeans",
   "format": "mojo",
   "exp_date": null,
   "more_info": " ... "
   …
}
```


RADICALBIT
radicalbit.io

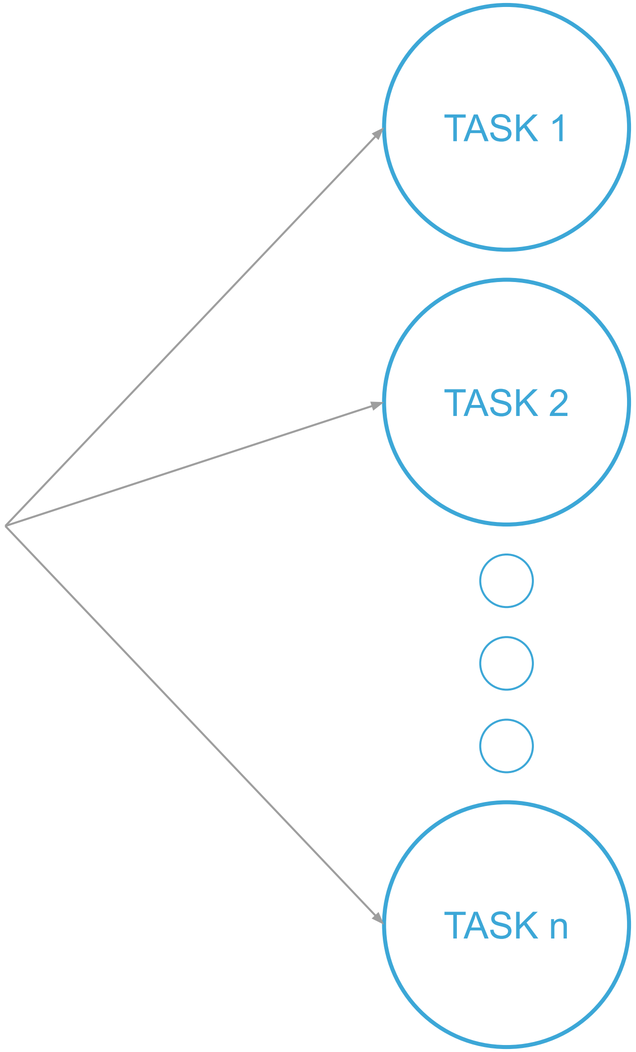
KSH$_2$O - THE CONTROL STREAM (2)

1 - to - 1 Bind

Model Repository Server → Control Stream

KSH$_2$O - FEEDING A METADATA TABLE

```
{
    …
    "id": "unsupervised_cusomers_1",
    "algorithm": "kmeans",
    "format": "mojo"
    "exp_date": null
    "more_info": " ... "
    …
}
```



GlobalKTable

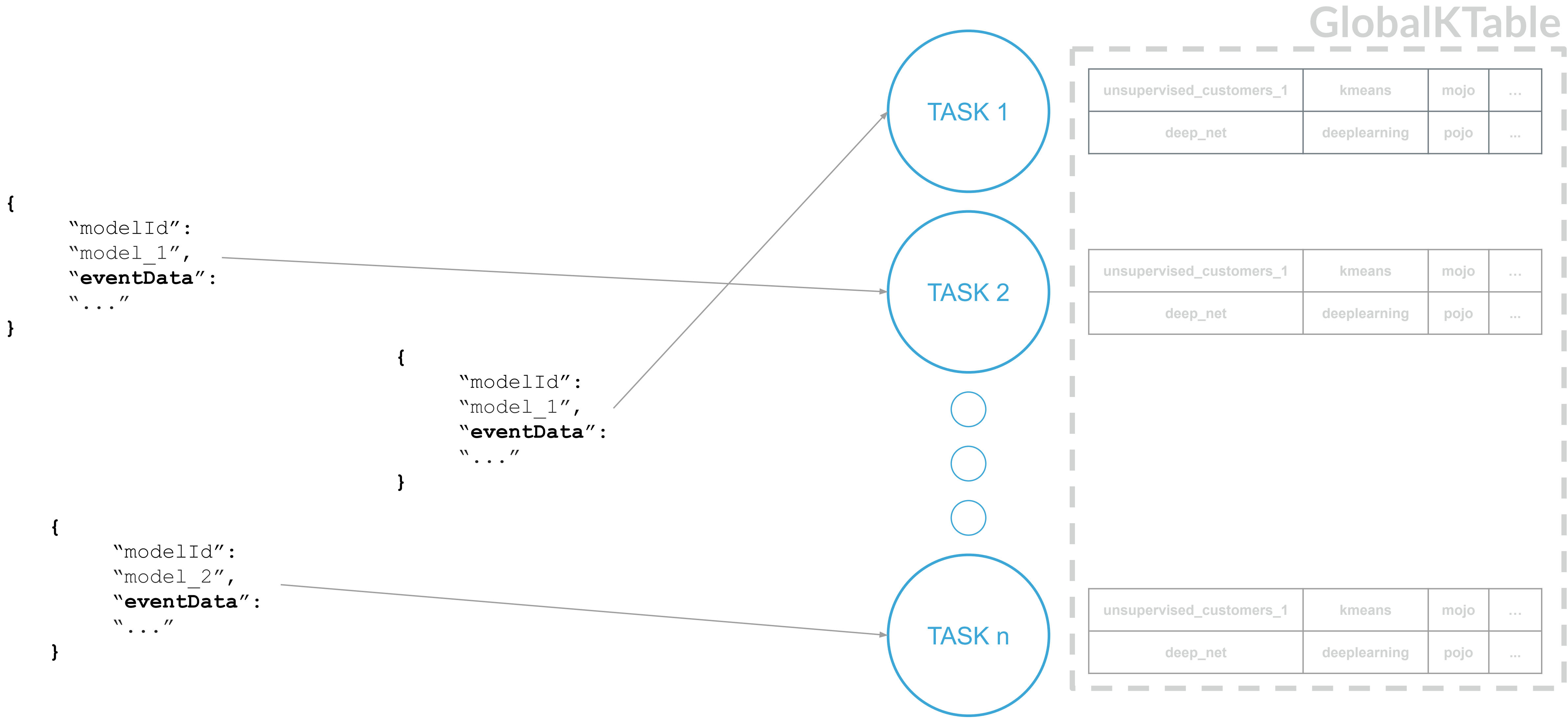
unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

RADICALBIT
radicalbit.io

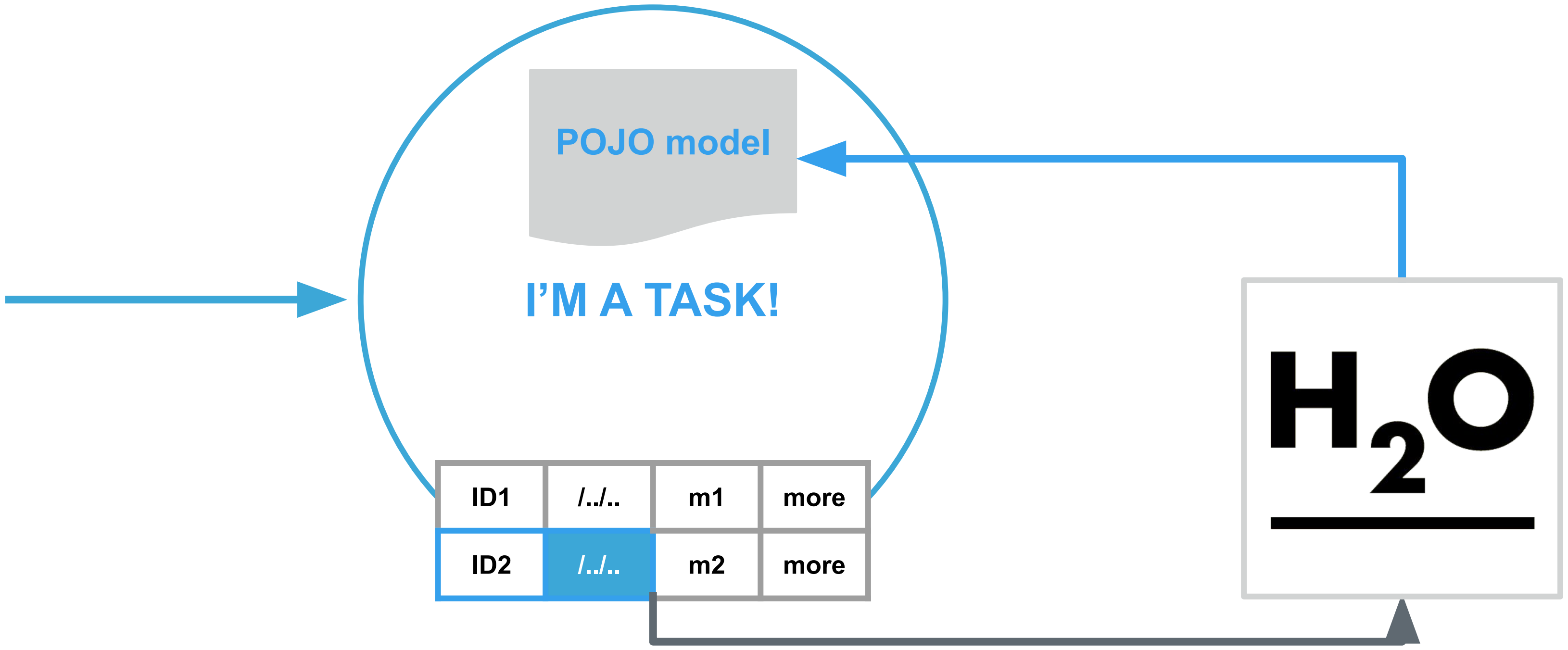
KSH_2O - THE DATA STREAM



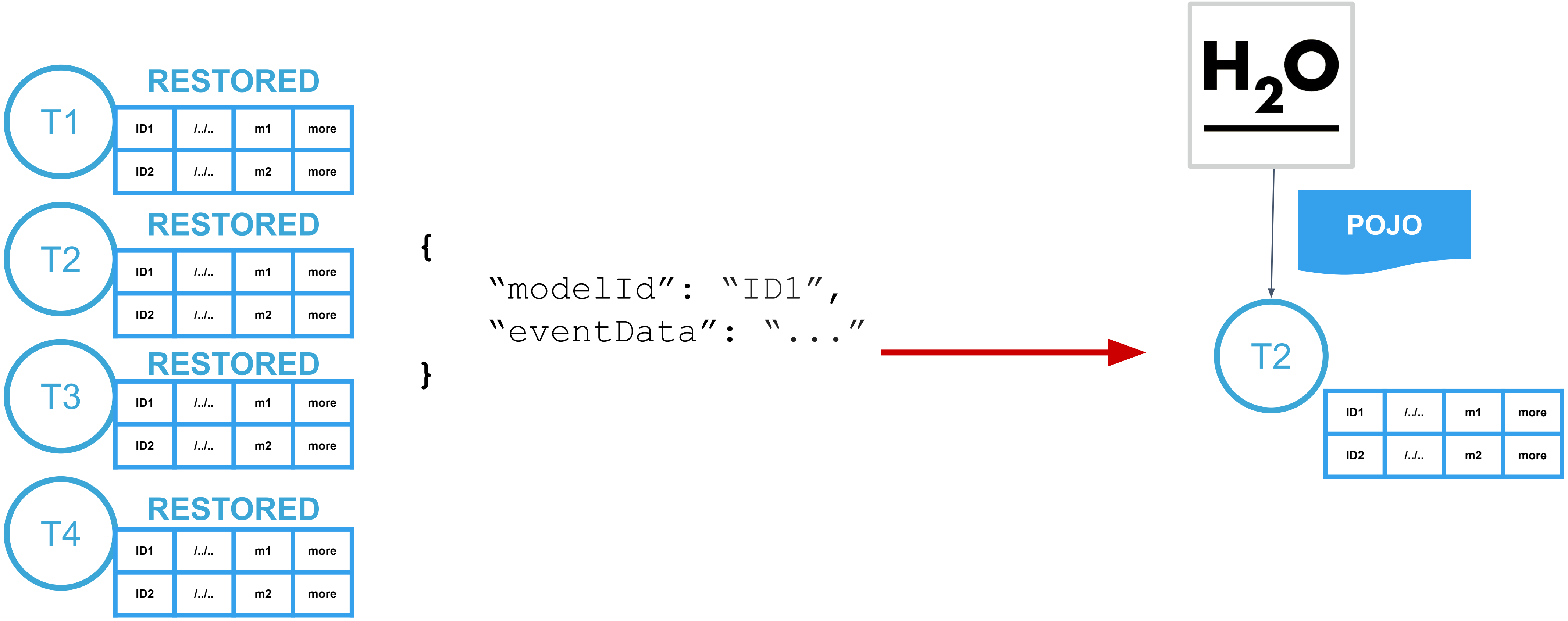
RADICALBIT
radicalbit.io

KSH_2O - LAZY MODELING LOADING

```
{
    "modelId": "ID1",
    "eventData":
    "..."
}
```



KSH$_2$O - MODEL STORAGE AND FAULT TOLERANCE



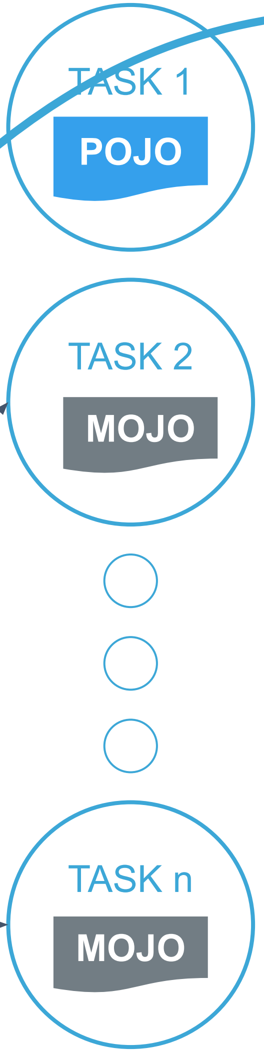
On *restore*, lazy uploading applies models' recovering

THE KS-H$_2$O ARCHITECTURE

Configuration

- Empty predictions strategy
- NaN management
- UDF
- …


```
{
        …
        "id": "unsupervised_cusomers_1",
        "algorithm": "kmeans",
        "format": "mojo"
        "exp_date": null
        "more_info": " ... "
        …
}
```

unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...


```
{
        "modeld": "model_1",
        "eventData": "..."
}
```

```
{
        "modeld": "model_2",
        "eventData": "..."
}
```

```
{
        "UserDefinedOutput": { … },
        "prediction": { /*PREDICTION OBJECT*/ }
}
```


ks-h2o - https://github.com/radicalbit

RADICALBIT
radicalbit.io

TOWARDS A GENERIC ARCHITECTURE

```
{
        …
        "id":
        "unsupervised_cusomers_1",
        "algorithm": "kmeans",
        "format": "mojo"
        "exp_date": null
        "more_info": " ... "
        …
}
```

unsupervised_customers_1	kmeans	mojo	…
deep_net	deeplearning	pojo	…

First generalisation

Given a control message, how to build the shared state

TOWARDS A GENERIC ARCHITECTURE

Second generalisation

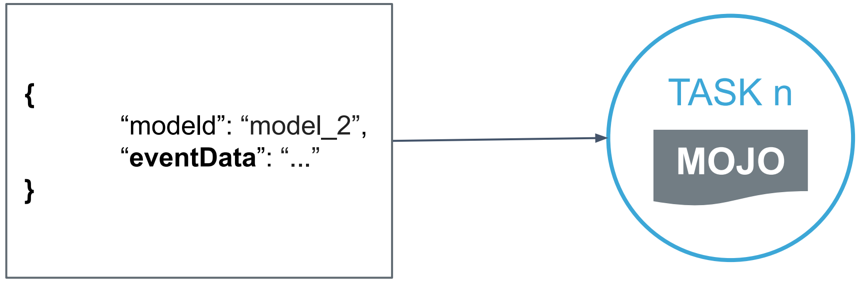
Given the record to score, how to build the model

{
"modelId": "model_2",
"**eventData**": "..."
}

unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

Third generalisation

Given the record to score, implement the scoring method



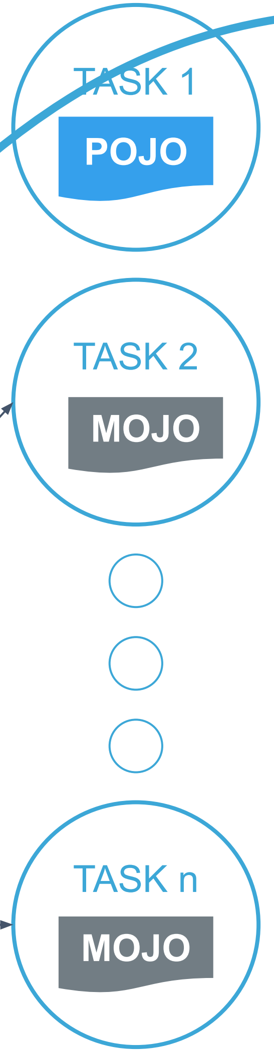
unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

THE KS-H_2O ARCHITECTURE

```
{
        …
        "id": "unsupervised_cusomers_1",
        "algorithm": "kmeans",
        "format": "mojo"
        "exp_date": null
        "more_info": " ... "
        …
}
```

```
{
        "modelId": "model_1",
        "eventData": "…"
}
```

```
{
        "modelId": "model_2",
        "eventData": "…"
}
```



unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

Configuration

- Empty predictions strategy
- NaN management
- UDF
- …

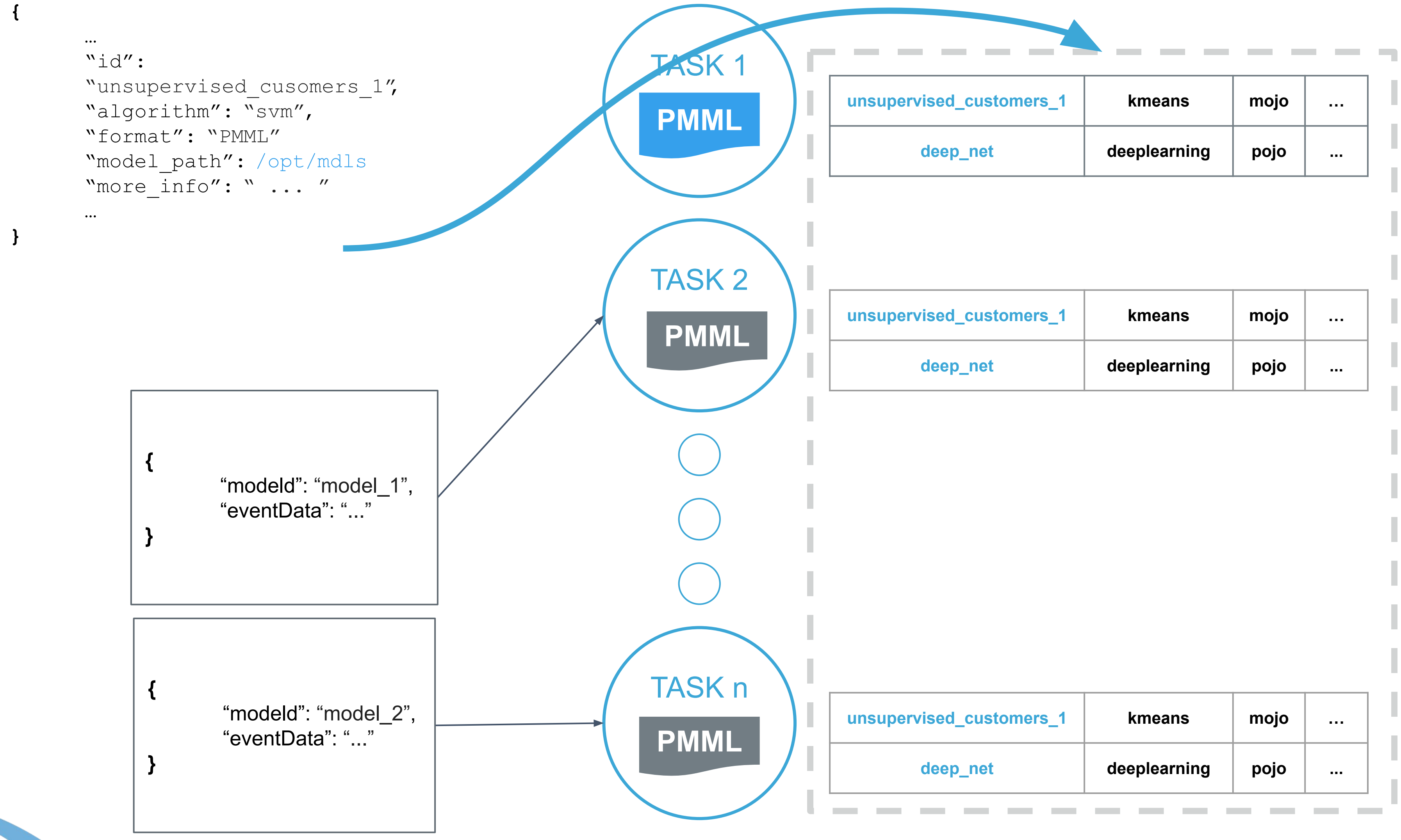
```
{
        "UserDefinedOutput": { … },
        "prediction": {
        /*PREDICTION OBJECT*/
        }
}
```

https://github.com/radicalbit

ks-h2o - https://github.com/FlinkML/flink-jpmml

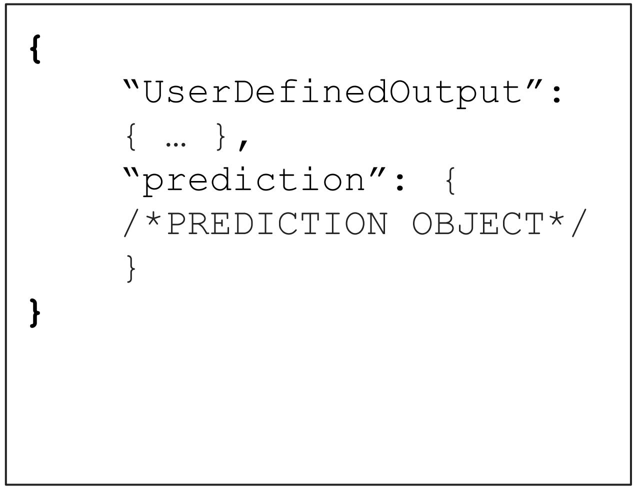
RADICALBIT
radicalbit.io

SERVING STANDARD MODELS DEPLOYMENTS



Configuration

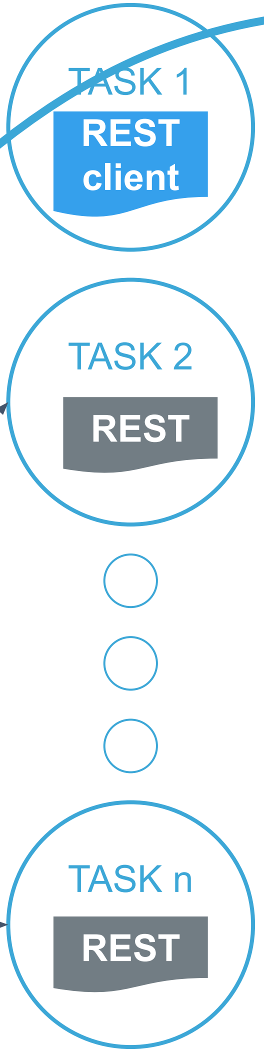
- Empty predictions strategy
- NaN management
- UDF
- …



flink-jpmml - https://github.com/FlinkML/flink-jpmml

SERVING CONTAINERIZED MODELS DEPLOYMENTS

```
{
    …
    "id": "rnn-1",
    "algorithm": "net",
    "format": "Tensorflow"
    "serve_url":
    https://serve.com/opt/mdls
    …
}
```



unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

unsupervised_customers_1	kmeans	mojo	...
deep_net	deeplearning	pojo	...

```
{
    "modeId": "model_1",
    "eventData": "..."
}
```

```
{
    "modeId": "model_2",
    "eventData": "..."
}
```

Configuration

- Empty predictions strategy
- NaN management
- UDF
- ...

```
{
    "UserDefinedOutput":
    { … },
    "prediction": {
    /*PREDICTION OBJECT*/
    }
}
```

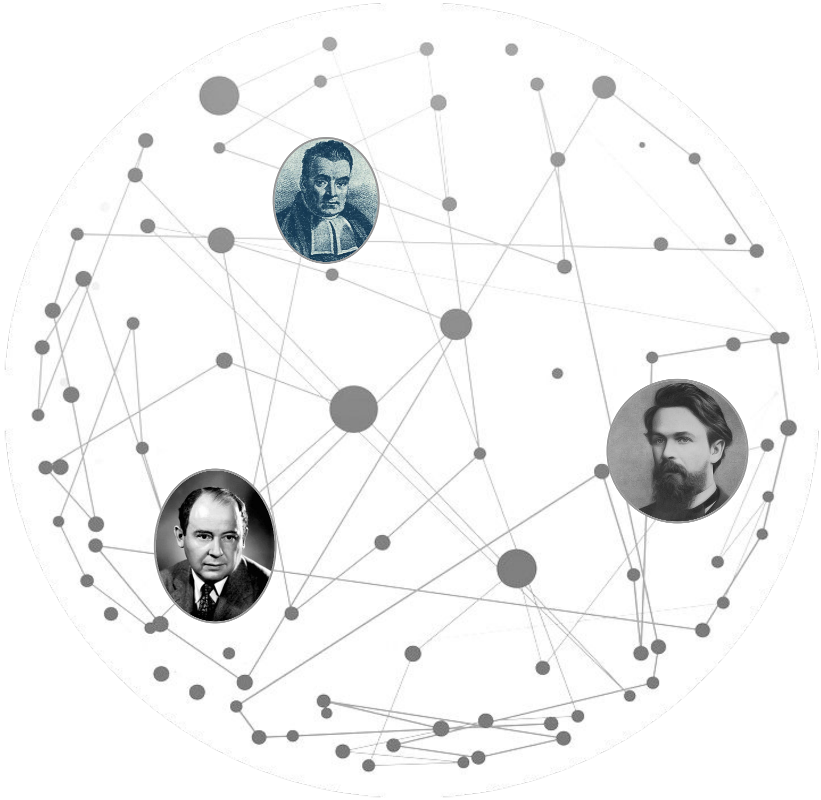
KS-OML

Online Machine Learning using Kafka Streams

RADICALBIT
radicalbit.io

MACHINE LEARNING, TRAINING

DIM	WEIGHT	COLOR
3	0.8	R
1	0.2	b
1	1.2	y
4	2.1	g
3	0.9	r
2	1.0	r
12	0.2	b
1	0.3	g
1	0.4	y
3	0.1	b
3	0.2	g
4	2.0	r
4	3.1	c
3	0.8	R
1	0.2	b
1	1.2	y

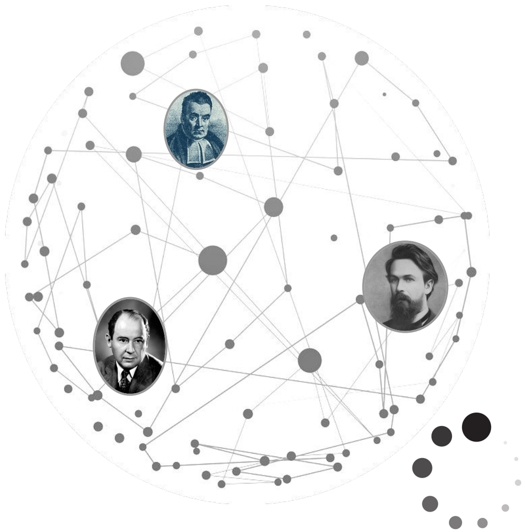


RADICALBIT
radicalbit.io

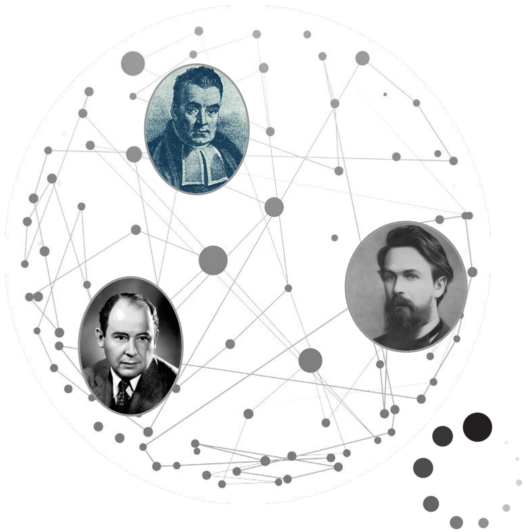
ONLINE LEARNING IS ABOUT A PORTION OF DATA

t

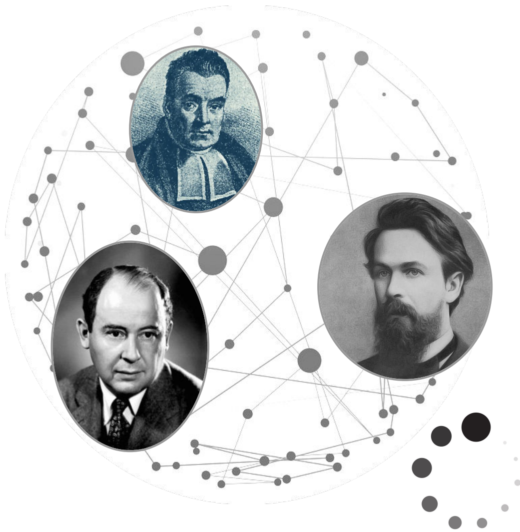
DIM	3
WEIGHT	0.8
COLOR	R



DIM	3
WEIGHT	1.6
COLOR	O



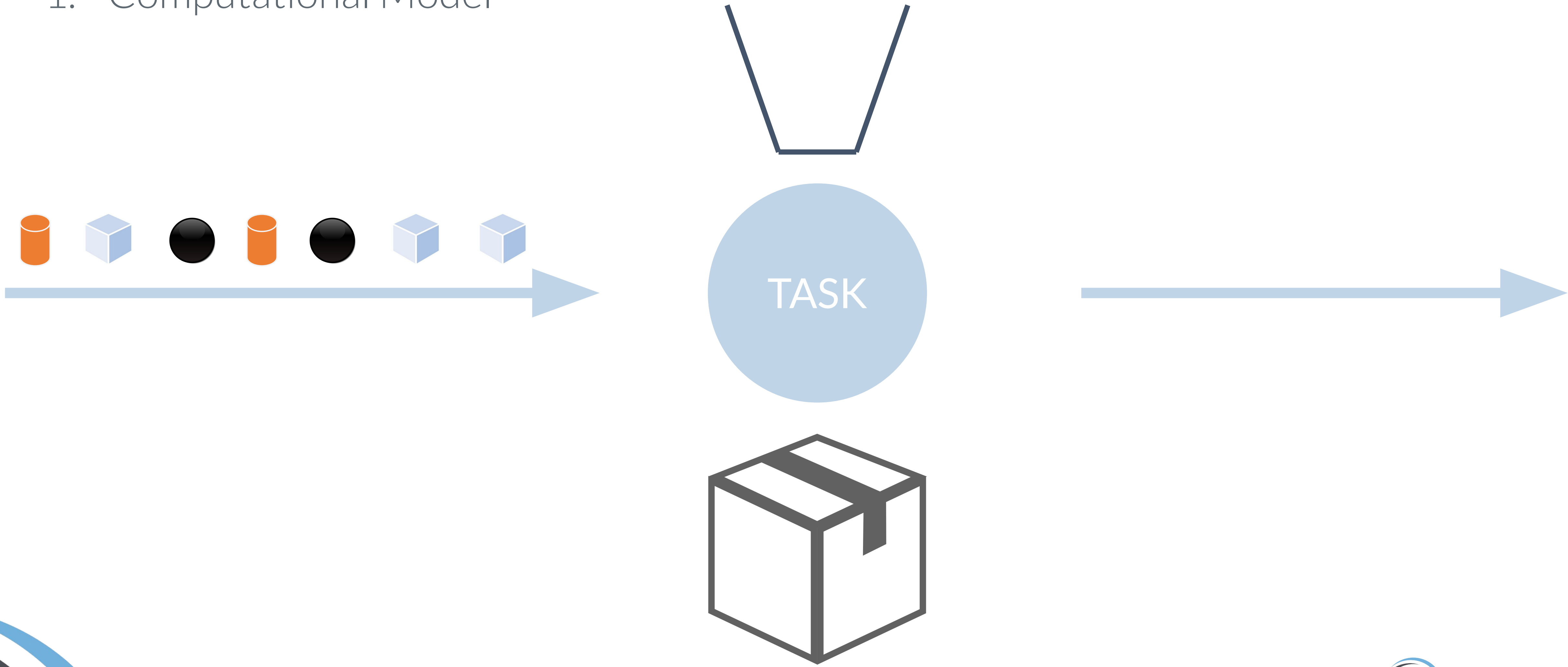
DIM	3
WEIGHT	0.3
COLOR	G



RADICALBIT
radicalbit.io

KS-OML - ONLINE LEARNING CHALLENGES

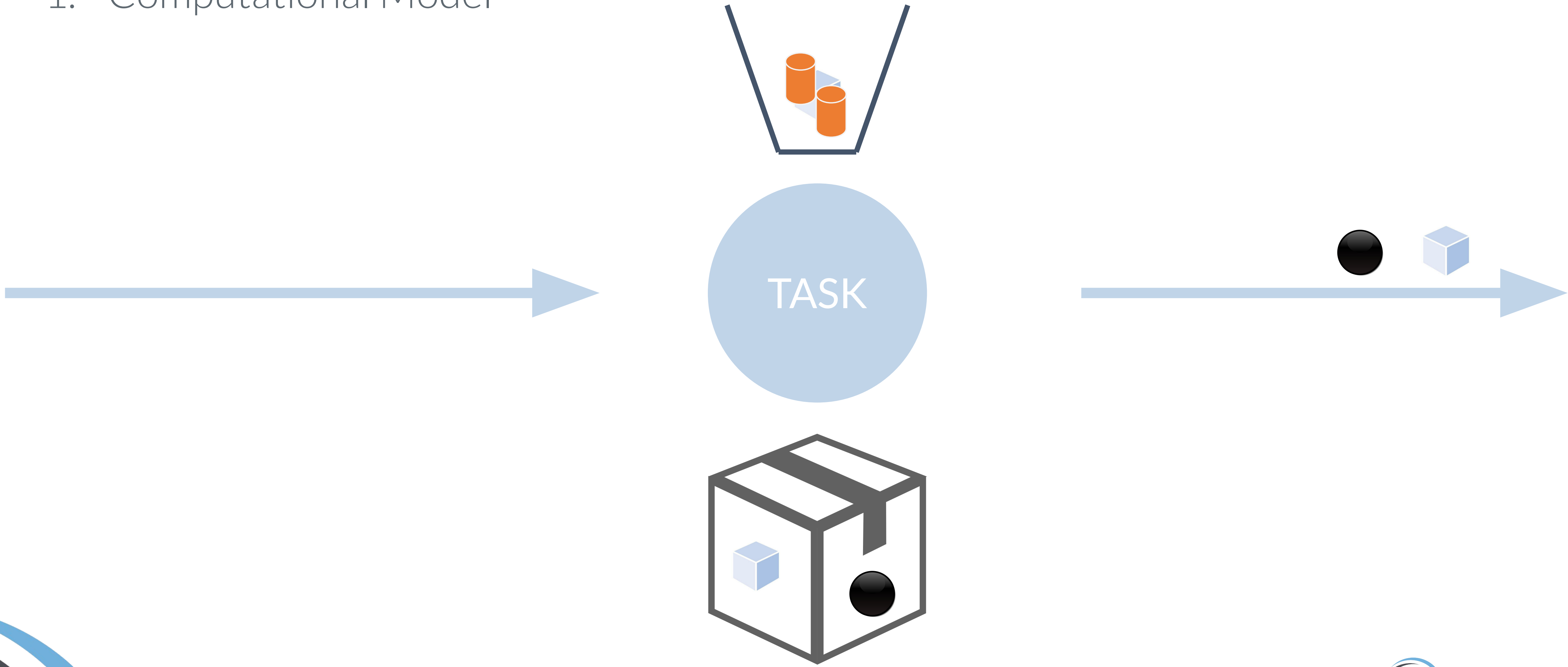
1. Computational Model



RADICALBIT
radicalbit.io

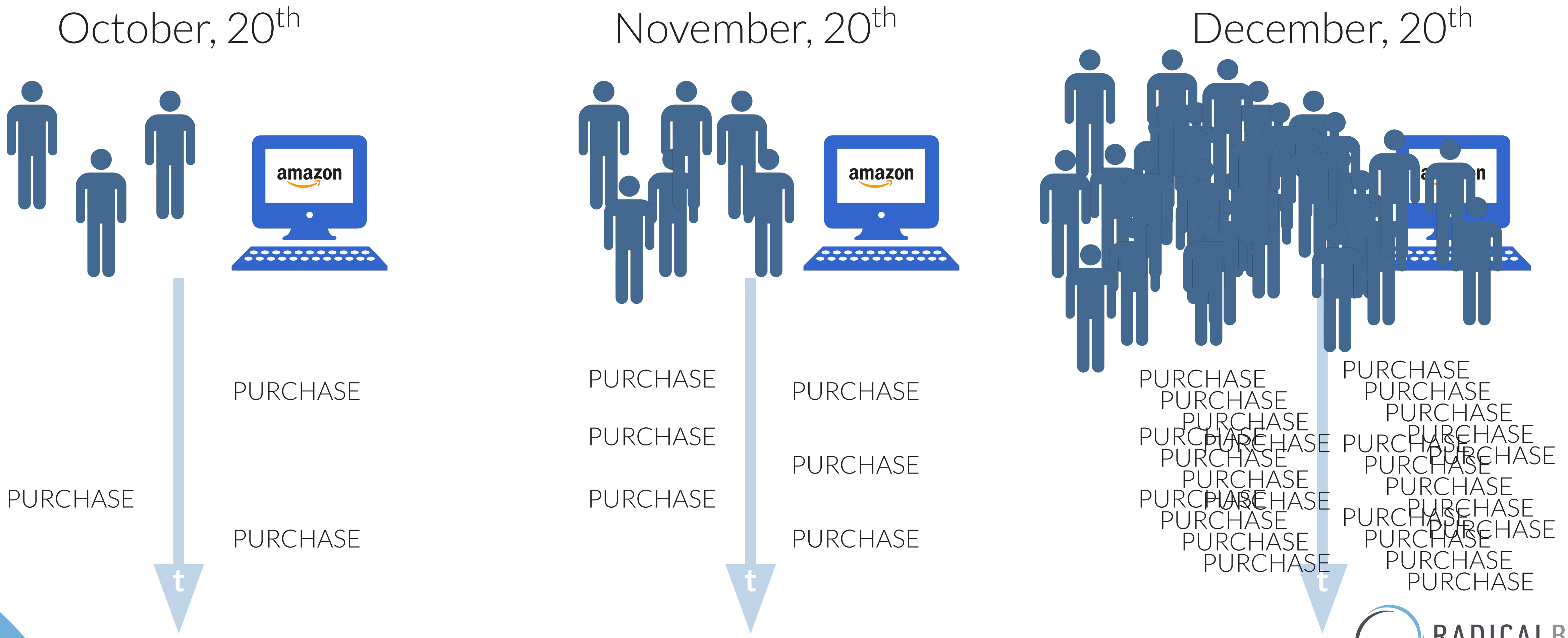
KS-OML - ONLINE LEARNING CHALLENGES

1. Computational Model



KS-OML - ONLINE LEARNING CHALLENGES

2. Evolving Data - or *the long-standing issue of the Concept Drift*



RADICALBIT
radicalbit.io

ONLINE MACHINE LEARNING STATE

Growing **academic** interest

`Online Machine Learning in Big Data Streams, 2018, András Benczúr, Levente Kocsis, Róbert Pálovics`

Still *rare* productionized OML implementations

The value

1. When the ability **to fast adapt** is more important than the best performance
   Newspaper domestic affairs drift example

2. When keeping data offline is not possible
   healthcare data, not reachable data

3. Resource savings

KS-OML

Online Machine Learning on Kafka architecture

Main objective

all-contained operator with configurable **algorithms suite**

First Implementation: *passive-aggressive algorithm*, Daniele Tria

Second Implementation: *soft confidence-weighted algo*, Seyedmasih Hosseinimotlagh

(1) Online Passive-Aggressive Algorithms - Crammer, Dekel, Keshet, Shalev-Shwartz, Singer
http://jmlr.csail.mit.edu/papers/volume7/crammer06a/crammer06a.pdf

(2) Soft confidence-weighted Algorithms - Wang, Zaho, Hoi
https://arxiv.org/ftp/arxiv/papers/1206/1206.4612.pdf

RADICALBIT
radicalbit.io

KS-OML - PASSIVE AGGRESSIVE ALGORITHM

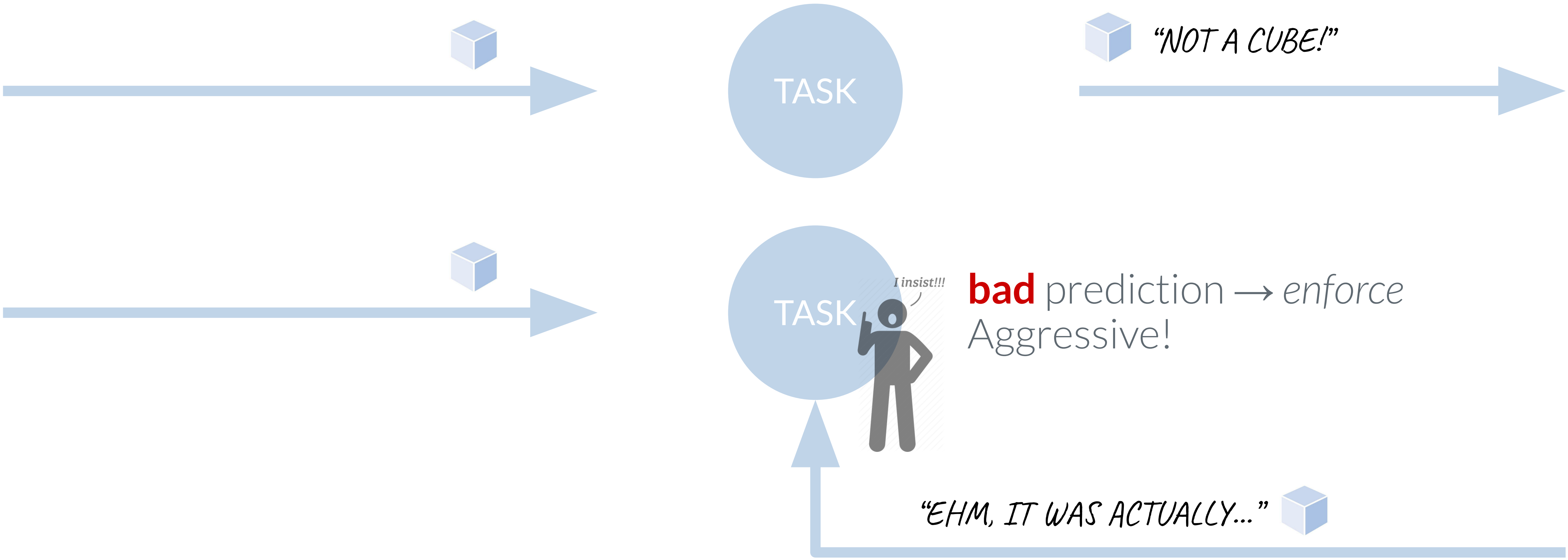
- *margin* based algorithm
- able to solve **binary** class, **multiclass** and **regression** problems
- *feedback* concept

- for binary class, given
  - y_t the true label
  - $\mathbf{x}_t$ the feature vector
  - $\mathbf{w}_t$ the weighted vector of the model

- $y_t(\mathbf{x}_t \cdot \mathbf{w}_t) = \mathit{margin}$ $\quad$ $\mathit{margin} > 0 \longrightarrow$ *correct prediction* $\quad$ $\mathit{sign}(\mathbf{x}_t \cdot \mathbf{w}_t) = y_t$

- why *passive-aggressive* ?
  - if *margin* $\geq +1$ $\longrightarrow$ *do nothing*
  - *else* $\longrightarrow$ *enforce the margin*

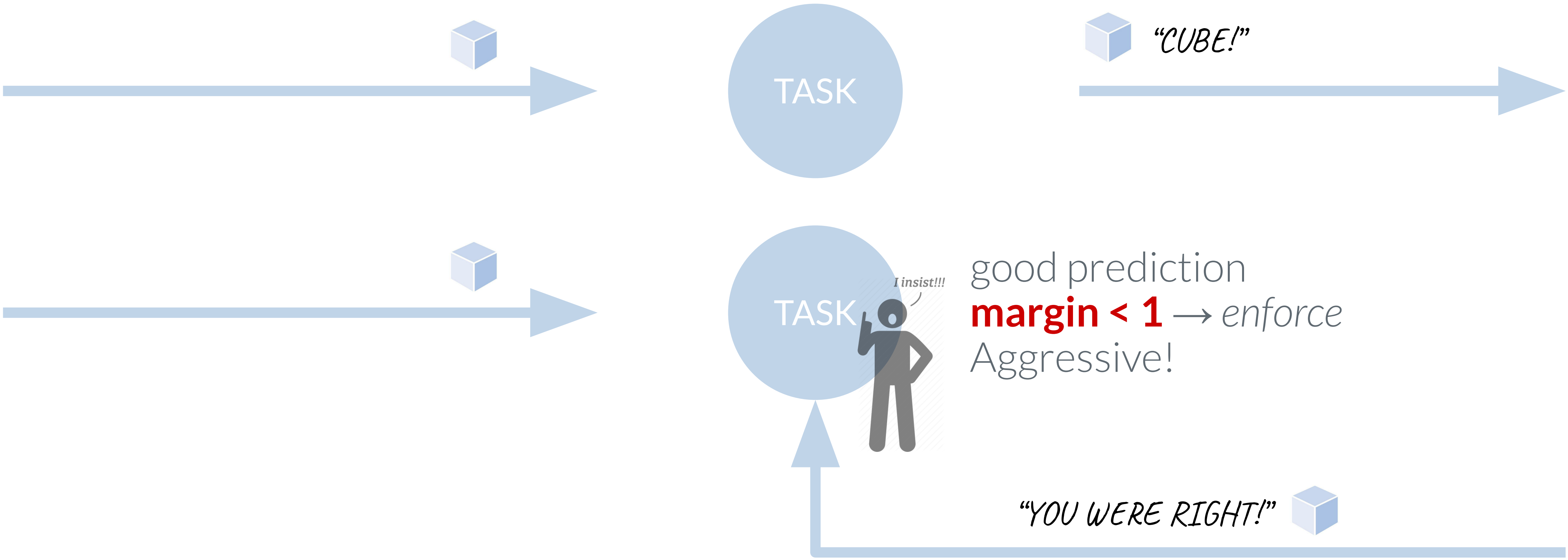
KS-OML - HIGH-LEVEL WORKFLOW



bad prediction → *enforce*
Aggressive!

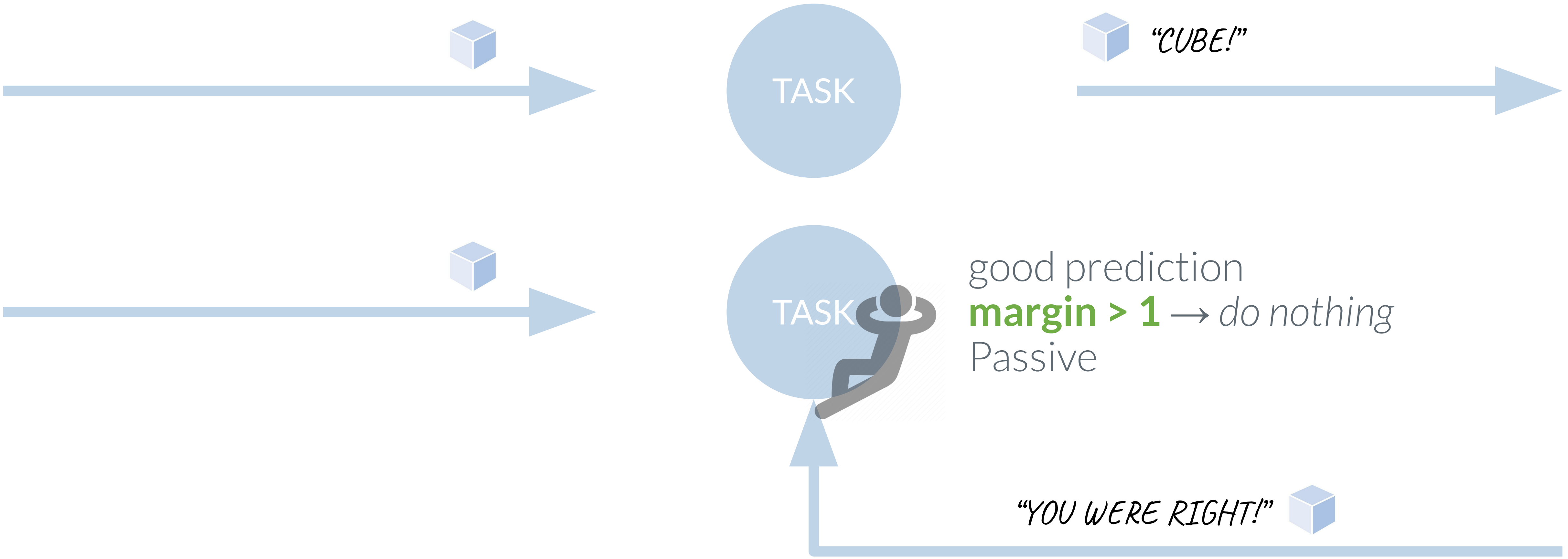
RADICALBIT
radicalbit.io

KS-OML - HIGH-LEVEL WORKFLOW



good prediction
margin < 1 → *enforce*
Aggressive!

KS-OML - HIGH-LEVEL WORKFLOW

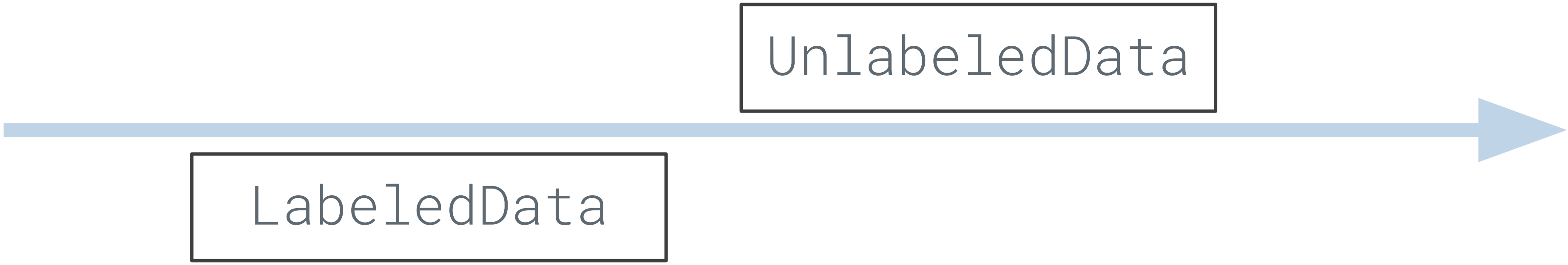


KS-OML - INPUT STREAMS

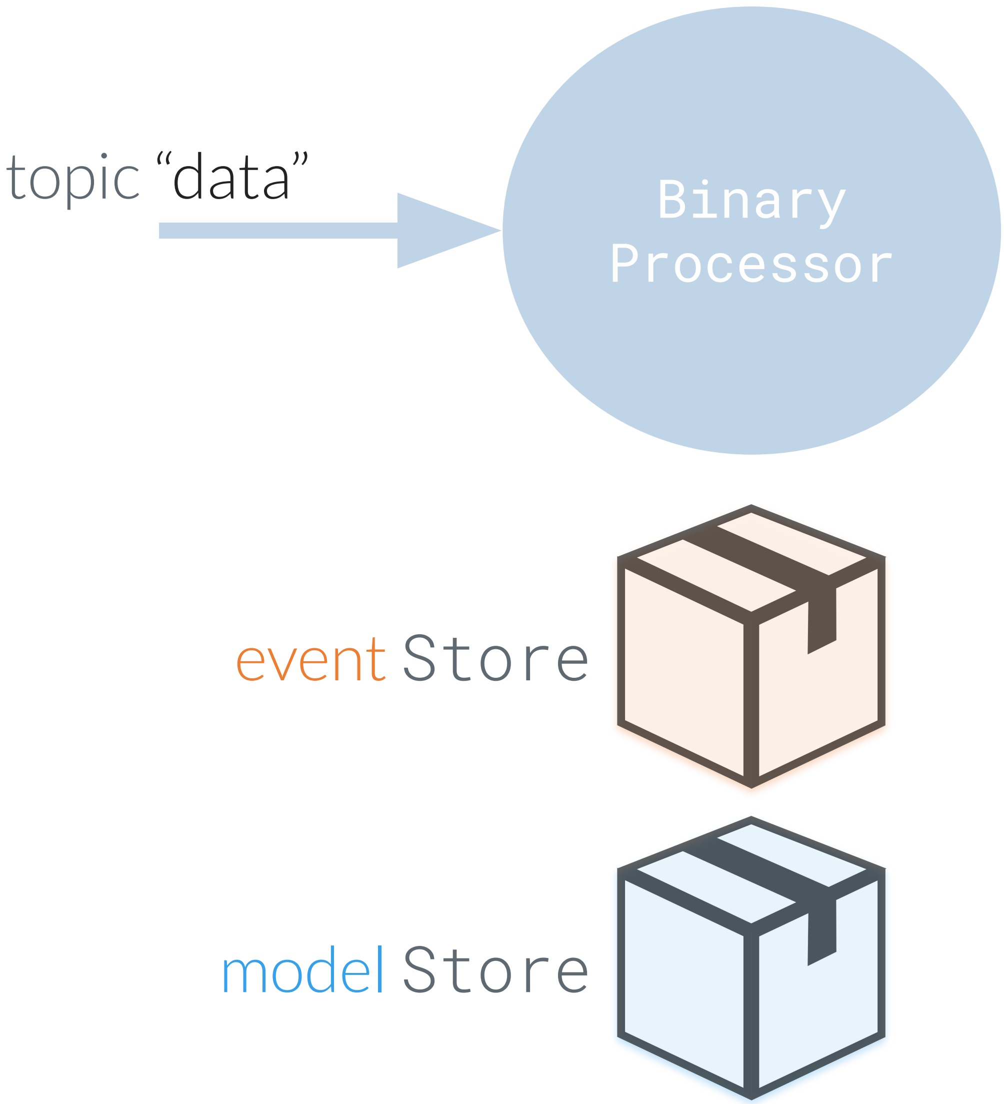
1. Main Event Stream : `UnlabeledData`
2. Feedback Event Stream : `LabeledData`

- *Connected* by the same Scala case class
- written in the same topic “data”

topic “data”



KS-OML - THE OPERATOR



When `UnlabeledData`

- get model from store
- compute prediction
- store event by hashing with prediction
- emit prediction

When `LabeledData`

- get event and model from stores
- check loss function *margin*
- eventually update the model
- delete the event from store
- emit again if required

KS-OML - RESULTS

	BANKNOTES DATASET (binary)	IRIS DATASET (multiclass)
First RUN	-	-
Second RUN	accuracy: 0.949671 precision: 0.97872	accuracy: 1.0 precision: 1.0
Third RUN	accuracy: 0.950747 precision: 0.98391	accuracy: 1.0 precision: 1.0

RADICALBIT
radicalbit.io

KS-OML - TAKEAWAYS

1. Feedback algos are good if you get the feedback a.s.a.p
2. Passive Aggressive is good when you can suffer of *cold start*
3. Passive Aggressive is *adaptive*
4. It works. Cool!

RADICALBIT
radicalbit.io

Pouring the blend

RADICALBIT
radicalbit.io

CONCLUSION

Model Serving

Streams are the perfect fit

Kafka is a natural solution for distribution and performance - but you need to tackle Kappa challenges!

Growing desire of an unique abstraction (both low and high level)

Online Learning

Increasing interest, industry still immature

Global shared streams, or states, even stores are fundamental to Machine Learning

THANKS!

Office hours 12.45 - 13.30

Your questions are welcome!

<radicalbit.team/>

info@radicalbit.io

RADICALBIT
radicalbit.io

REFERENCES

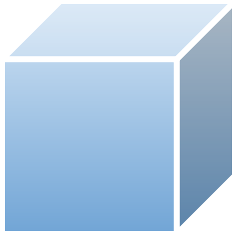
https://medium.com/value-stream-design/online-machine-learning-515556ff72c5
https://medium.com/analytics-vidhya/data-streams-and-online-machine-learning-in-python-a382e9e8d06a

RADICALBIT
radicalbit.io

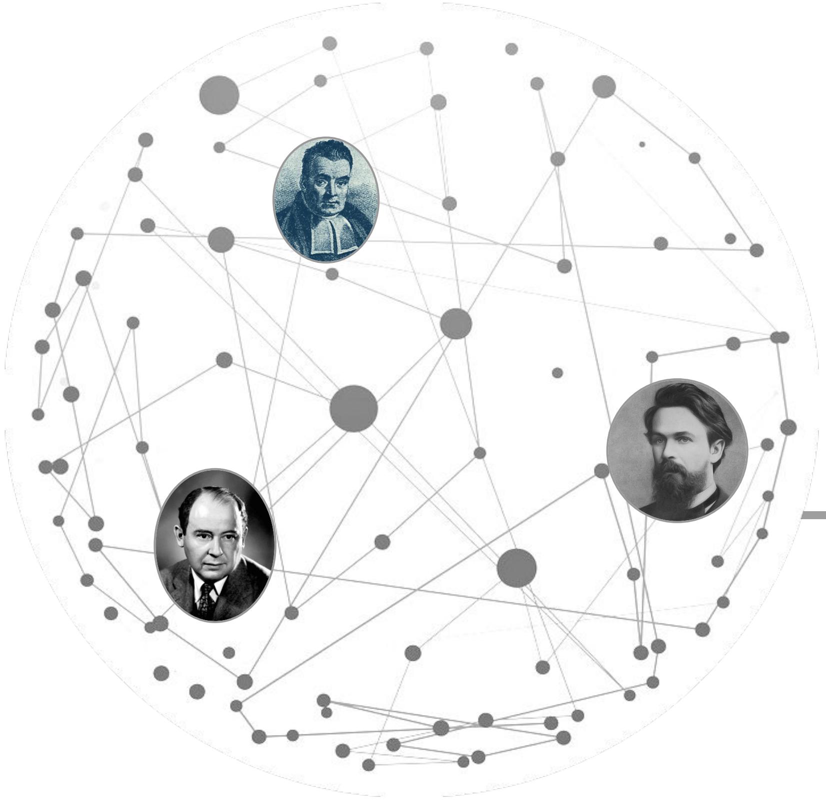
Bonus Slides

MACHINE LEARNING

MACHINE LEARNING, THEN



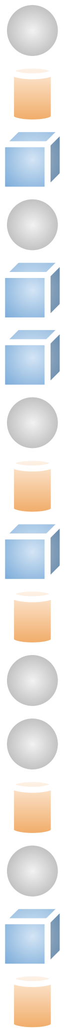
DIM	WEIGHT	COLOR
3	0.8	“R”

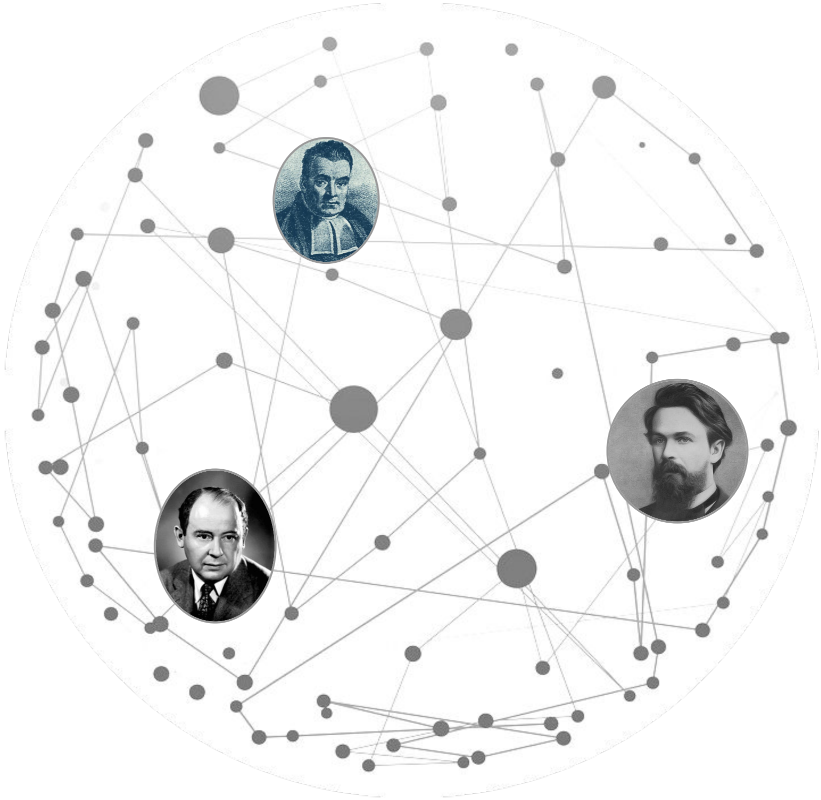


IT'S A "CUBE" !

MACHINE LEARNING, TRAINING

DIM	WEIGHT	COLOR
3	0.8	R
1	0.2	b
1	1.2	y
4	2.1	g
3	0.9	r
2	1.0	r
12	0.2	b
1	0.3	g
1	0.4	y
3	0.1	b
3	0.2	g
4	2.0	r
4	3.1	c
3	0.8	R
1	0.2	b
1	1.2	y

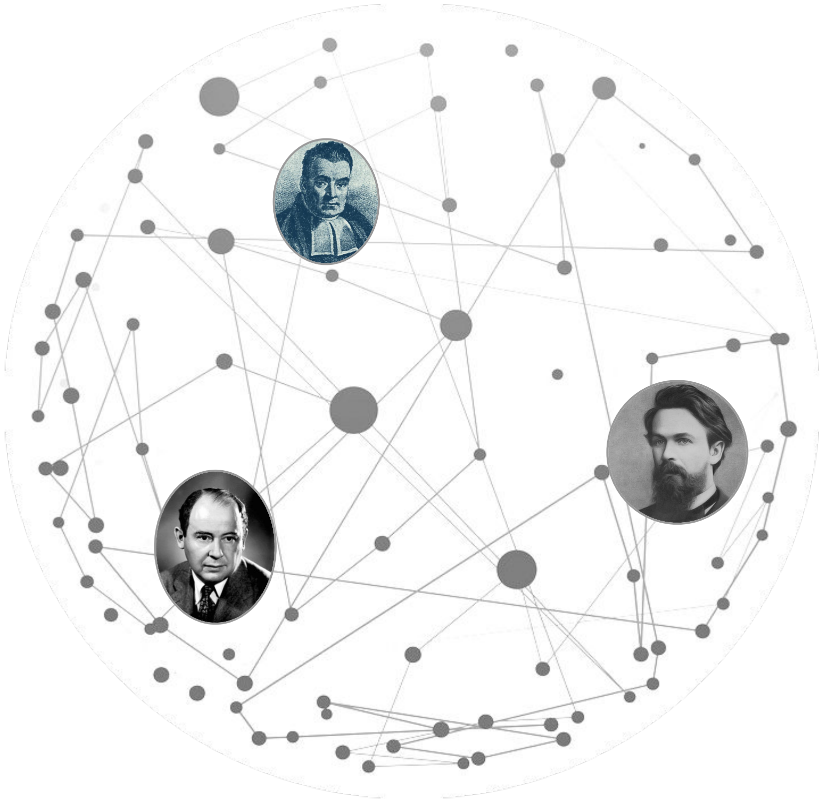




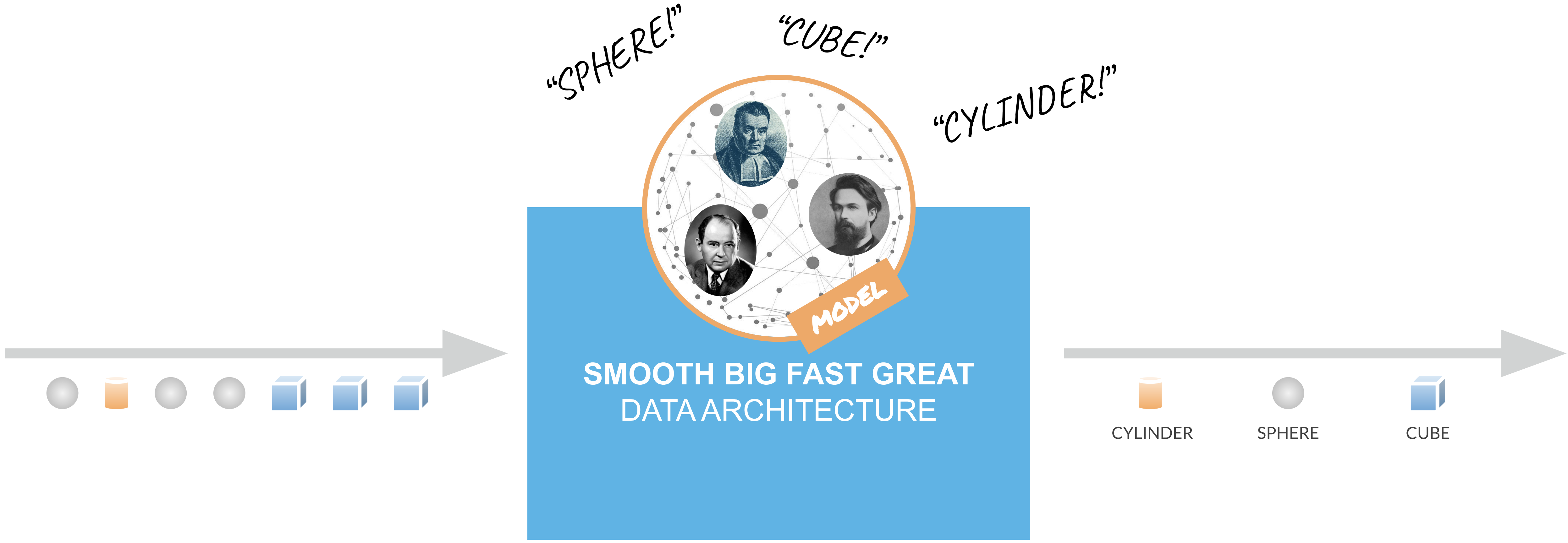
RADICALBIT
radicalbit.io

MACHINE LEARNING, TRAINING

DIM	WEIGHT	COLOR
3	0.8	R
1	0.2	b
1	1.2	y
4	2.1	g
3	0.9	r
2	1.0	r
12	0.2	b
1	0.3	g
1	0.4	y
3	0.1	b
3	0.2	g
4	2.0	r
4	3.1	c
3	0.8	R
1	0.2	b
1	1.2	y



MACHINE LEARNING, SCORING



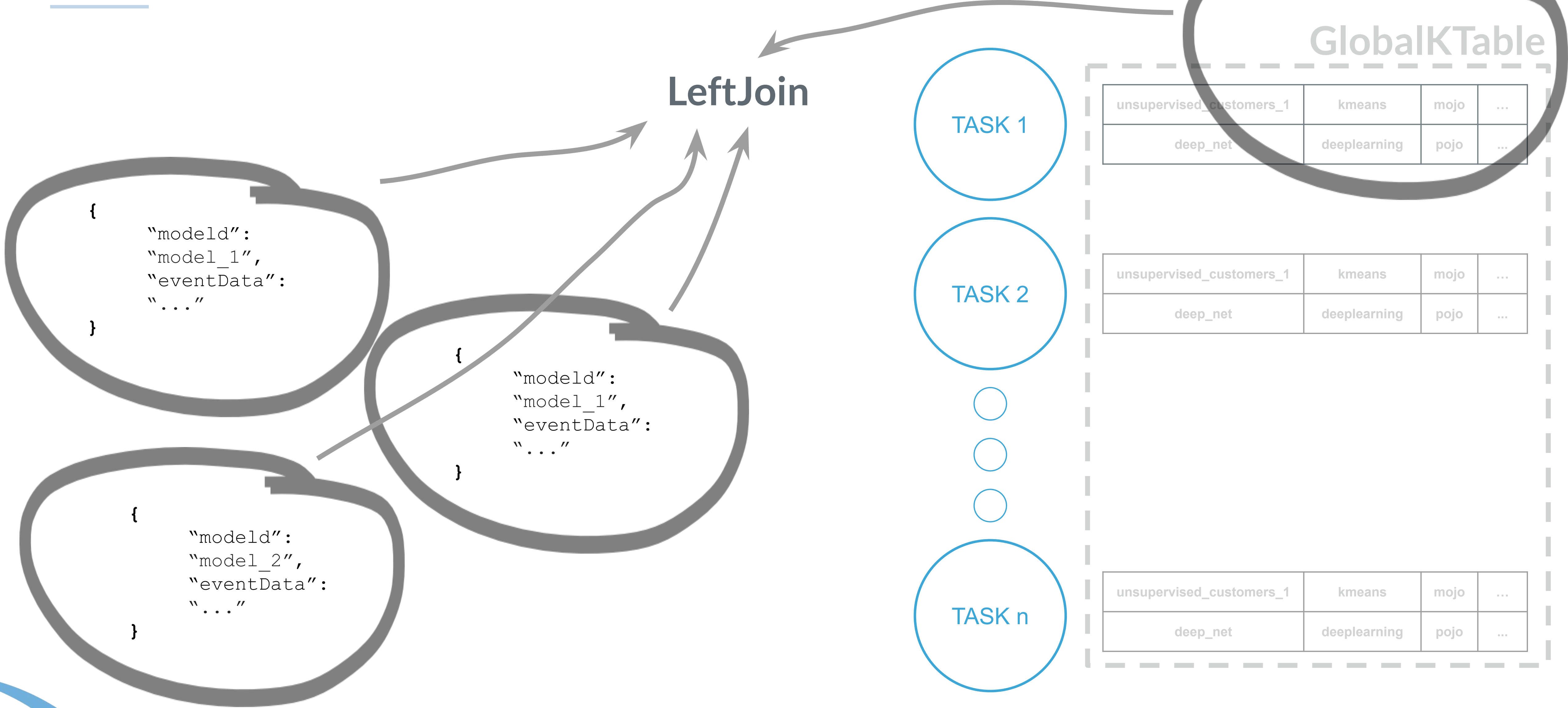
BLENDING ISSUES - MAIN GOALS

Main goal is introducing the above mentioned features in a native event stream platform, whereby:

- data is not finite and is *unknown*
- domain semantic changes over time
- processing logic might change over time
- applications evolve dynamically
- …

KSH_2O

KSH$_2$O - THE DATA STREAM



RADICALBIT
radicalbit.io

KSH$_2$O - KTABLE

- KTable is referring to a stream as of a **changelog**

1

1	kafka

```
{
 "k": 1,
 "value": "kafka"
}
```

2

1	kafka
2	mommy

```
{
 "k": 2,
 "value": "mommy"
}
```

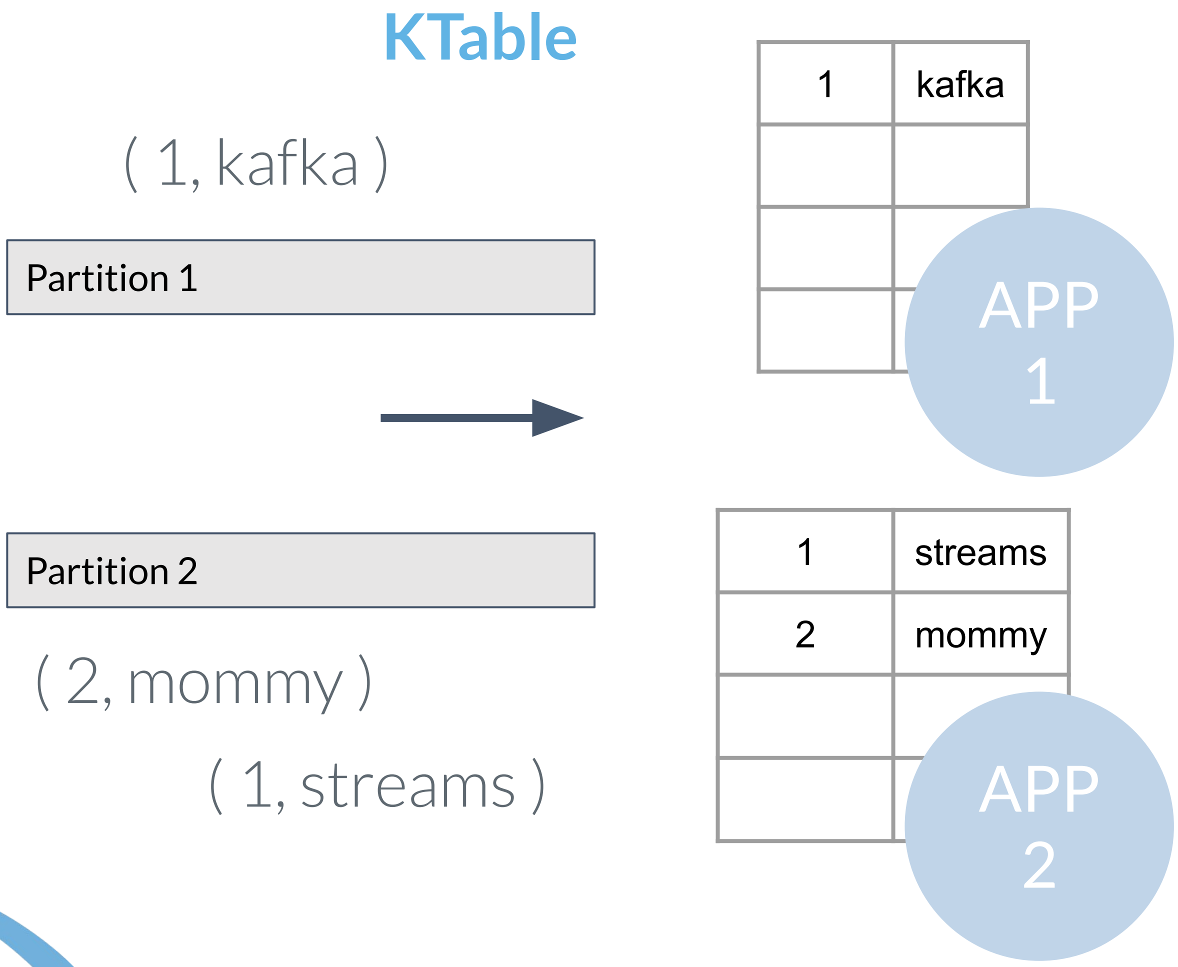
3

1	streams
2	mommy

```
{
 "k": 1,
 "value": "streams"
}
```

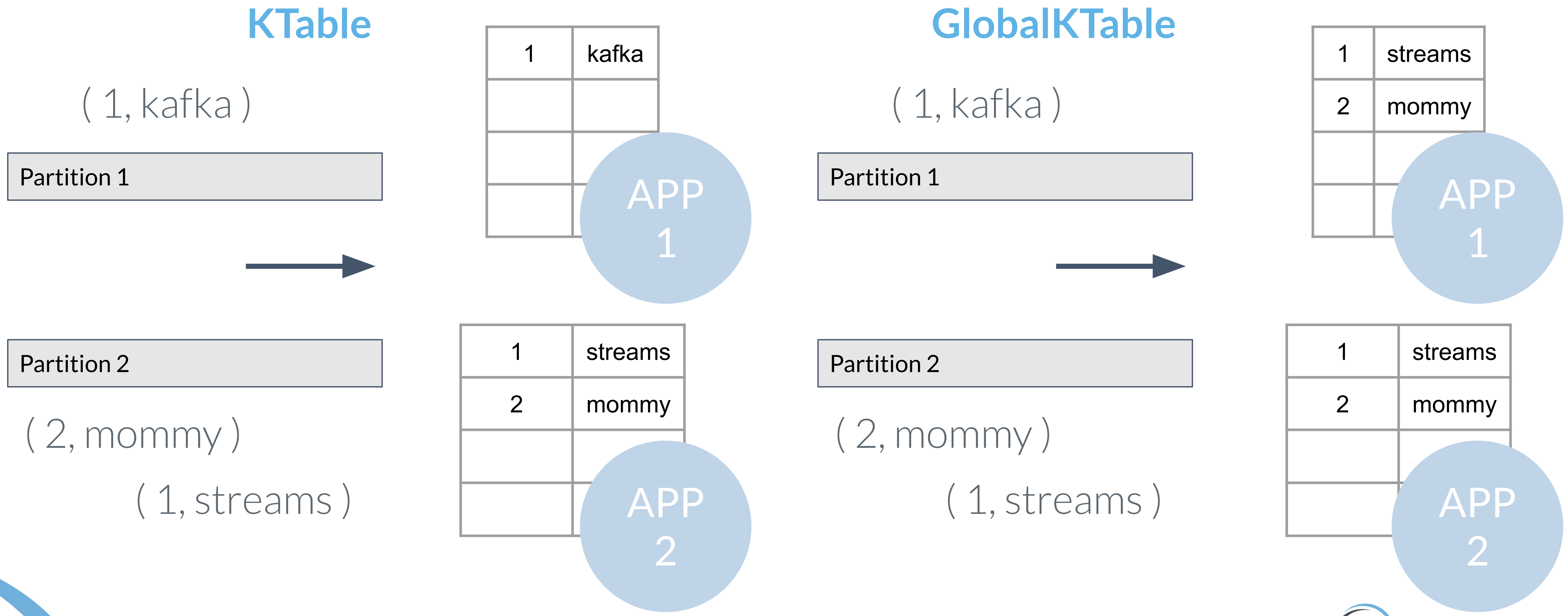
KSH_2O - KTABLE

- GlobalKTable is a KTable that is global in terms of **topic supervision**



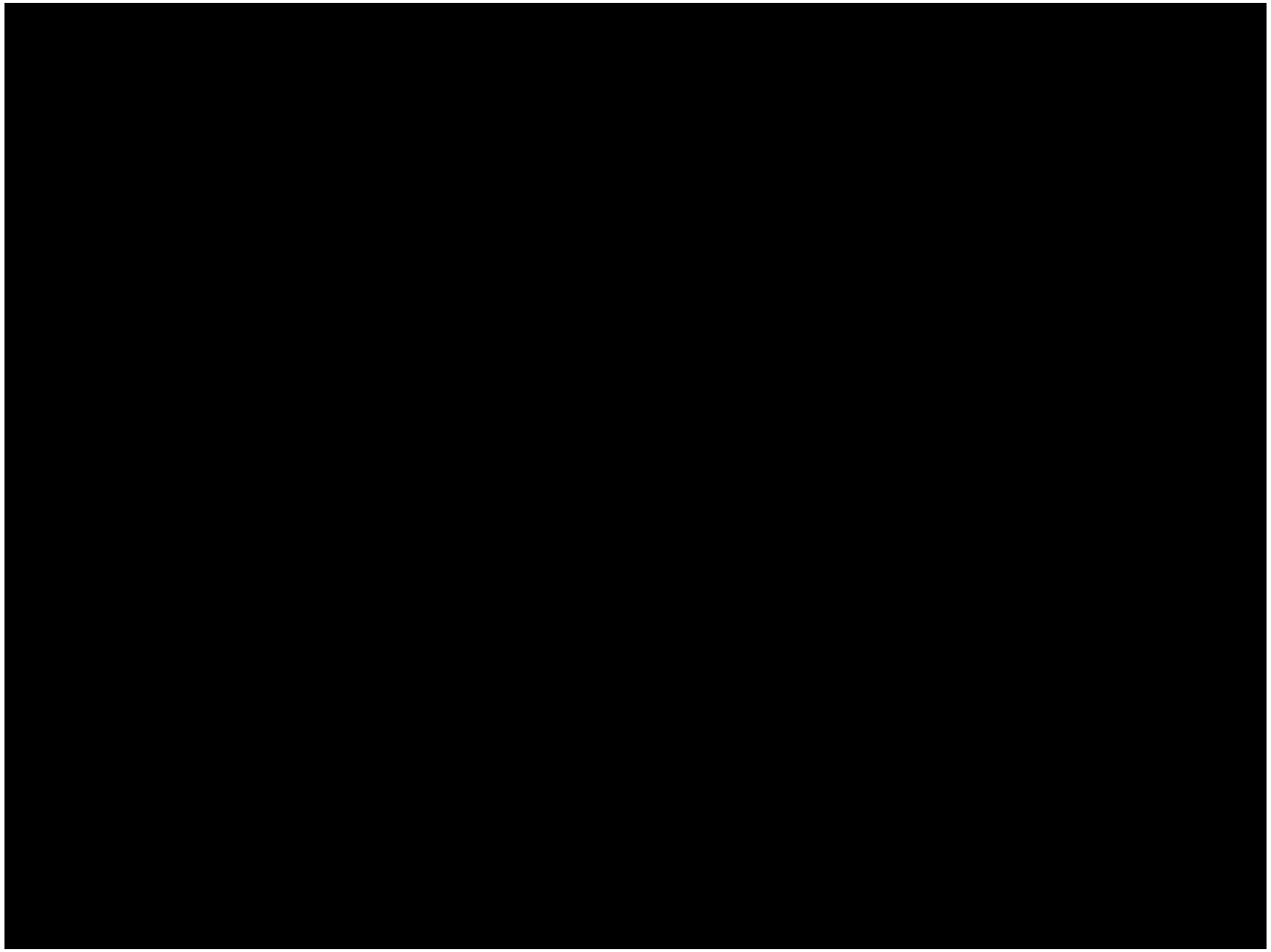
KSH$_2$O - KTABLE

- GlobalKTable is a KTable that is global in terms of **topic supervision**



Why don’t partitioning accordingly model stream and data stream?

KSH$_2$O - DEMO

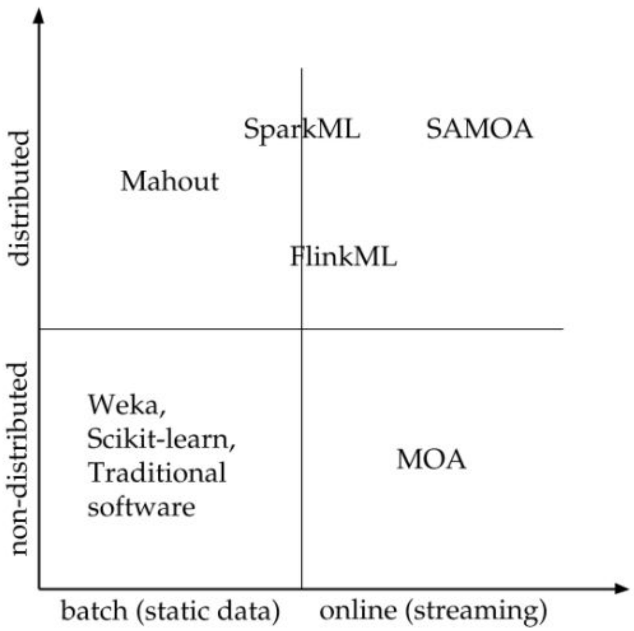


KSH_2O

KS-OML - ONLINE MACHINE LEARNING STATE

OML tools

- Apache SAMOA
  `Large-Scale Learning from Data Streams with Apache SAMOA, 2018`
  `Nicolas Kourtellis, Gianmarco De Francisci Morales, and Albert Bifet`
- side ML libraries on Apache Flink, Apache Spark, Apache Storm



KS-OML - RESULTS

Multi-class problem GENERATED DATASET	KS - OML Accuracy	Python “batch” implementation Accuracy
First RUN	-	0.905
Second RUN	0.99749374	0.99
Third RUN	0.9987469	0.995

RADICALBIT
radicalbit.io