RADICALBIT

Deep, Different, Disruptive.

FLINK | JPMML

An open source Flink library for streaming machine learning model serving

FRANCESCO FRONTERA
francesco.frontera@radicalbit.io
@ffrontera

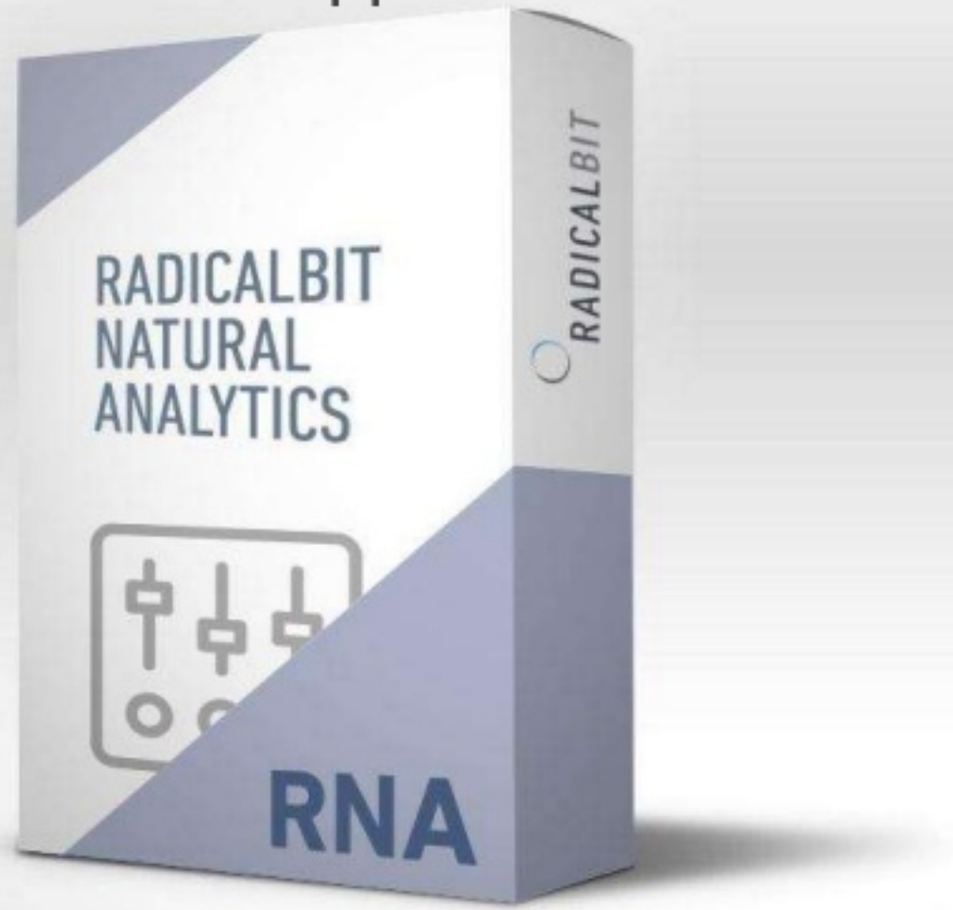
ANDREA SPINA
andrea.spina@radicalbit.io
@Spina89

RNA

Modern high performance real time analytics engine developed by **Radicalbit** (Italy)

Employing most modern microservice technologies and Scala

It's way awesome! Let's check it out at FF



Its core is Flink!

RADICALBIT
Deep. Different. Disruptive.

RNA

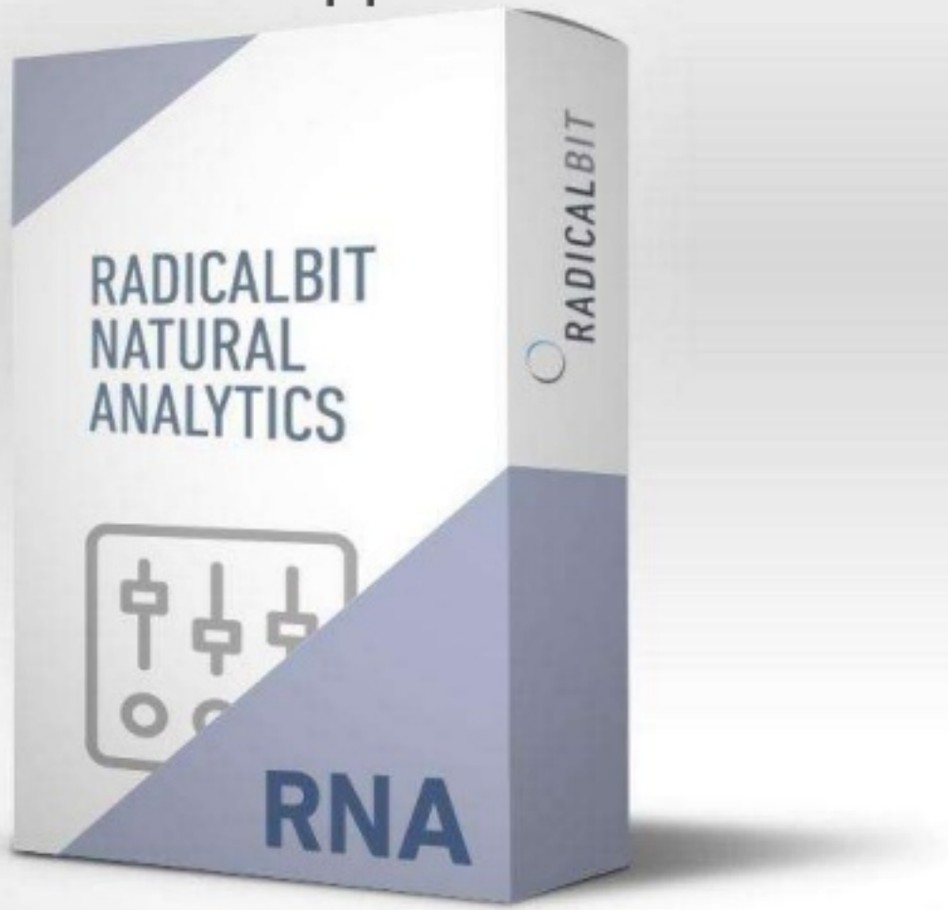
Modern high performance real time analytics engine developed by **Radicalbit** (Italy)

Employing most modern microservice technologies and Scala

What is missing out there

Need to investigate a method to exploit **evolving** Artificial Intelligence models in real time.

It's way awesome! Let's check it out at FF



Its core is Flink!

RADICALBIT
Deep. Different. Disruptive.

SWIFTLY DEMYSTIFYING MODEL SERVING

- Provide *continuous* services on intelligent apps ML integrated
- Exploiting AI models efficiently

It brings constraints:

- Low latency and high throughput
- Lightweight

And it fits in:

- indeed, Streaming and Scalable Processing
- exploit ML models efficiently

RADICALBIT
Deep. Different. Disruptive.

MAKE IT WITH THE SQUIRREL!

*… and become an **Apache Flink** Model Serving System*

RADICALBIT
Deep. Different. Disruptive.

THE JOURNEY TO THE FIRST RELEASE

Predictive Model Markup Language based

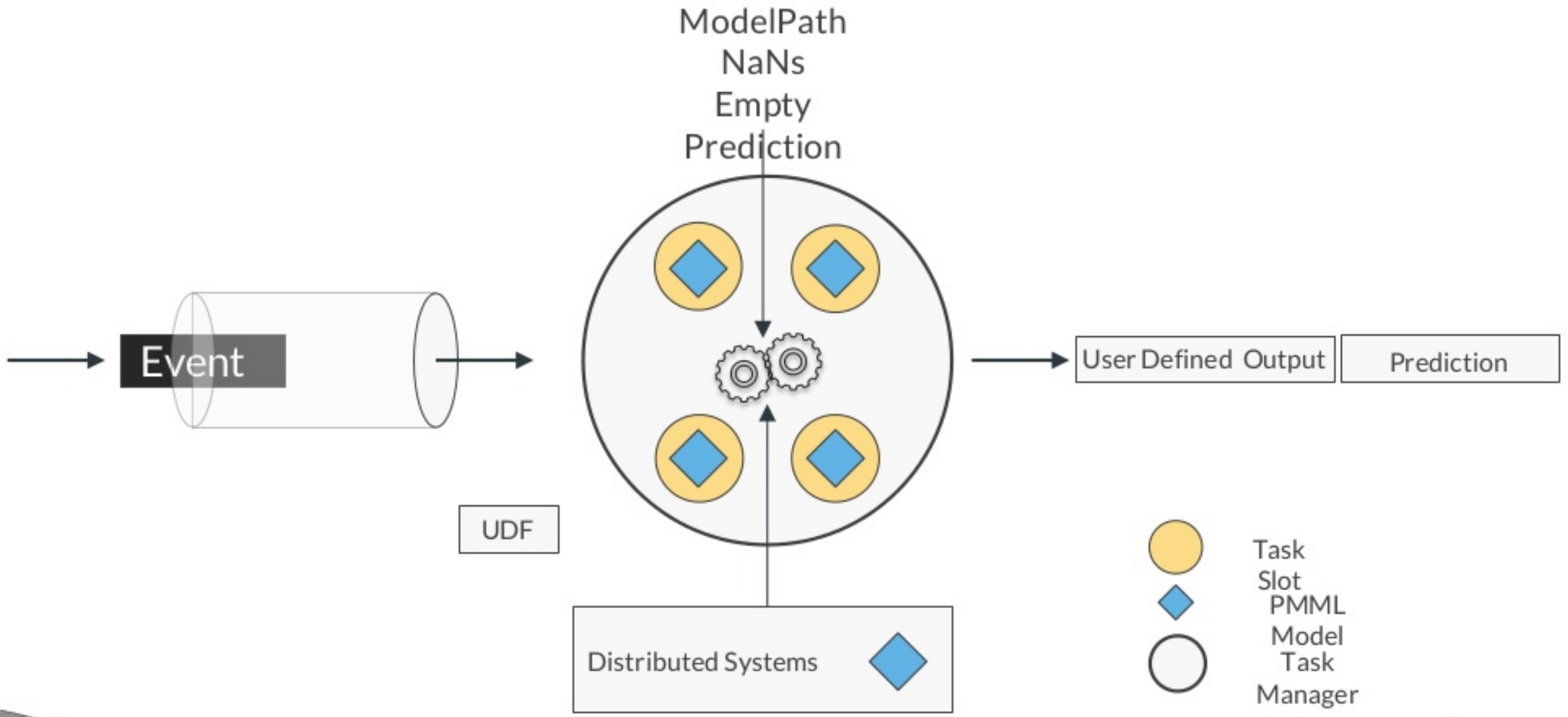
It should be a lightweight operator, KISS compliant

It should be fully-configurable

It should leverage distributed storage flink backend

Generally, we wanted it **FAST**

THE FIRST FLINK-JPMML ARCHITECTURE



RADICALBIT
Deep. Different. Disruptive.

THE FLINKML PROJECT LAUNCH

Rebooting old Flink Library
redirected to something bigger
recently strengthened project

Main objective: enhancing Flink on supporting
Streaming Model Serving
Incremental Learning
Online Machine Learning

This is awesome as well! Hope you enjoy Flink community!

- But there is a long road ahead
-

LACK OF THE FIRST FLINK-JPMML

Static behavior

Load a model and that's it

Single Model

Load the model exclusively

Fault intolerant

No integration with Flink state backend

And more …

Let's solve them!

RADICALBIT
Deep. Different. Disruptive.

DYNAMIC MODEL SERVING

The meaning

The operator should serve different models within the same operator at the same time

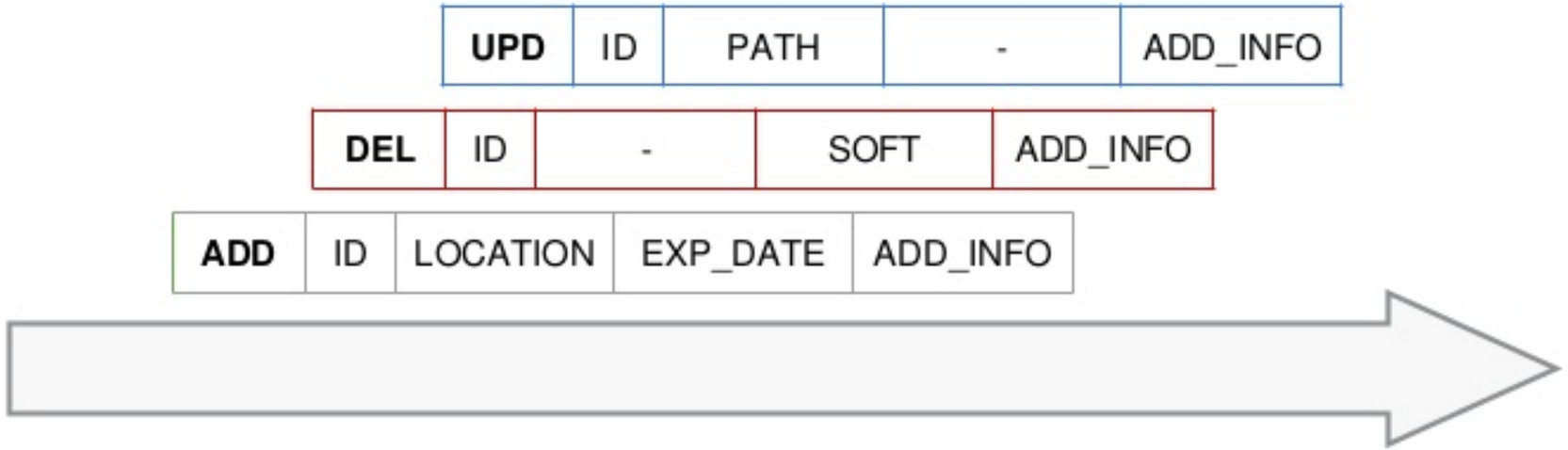
update models with newer versions if it's required

The constraints

The events have to declare somehow which model they want in order to be evaluated against

We need to take trace of the models active in the system

THE CONTROL STREAM



- we held the concept of node parametrization by control stream, and we based it on an expandable protocol
- messages match actions affecting the operator

ADD, (DEL), UPDATE: 1–to–1 bind with model repository servers

we don't deliver models

RADICALBIT
Deep. Different. Disruptive.

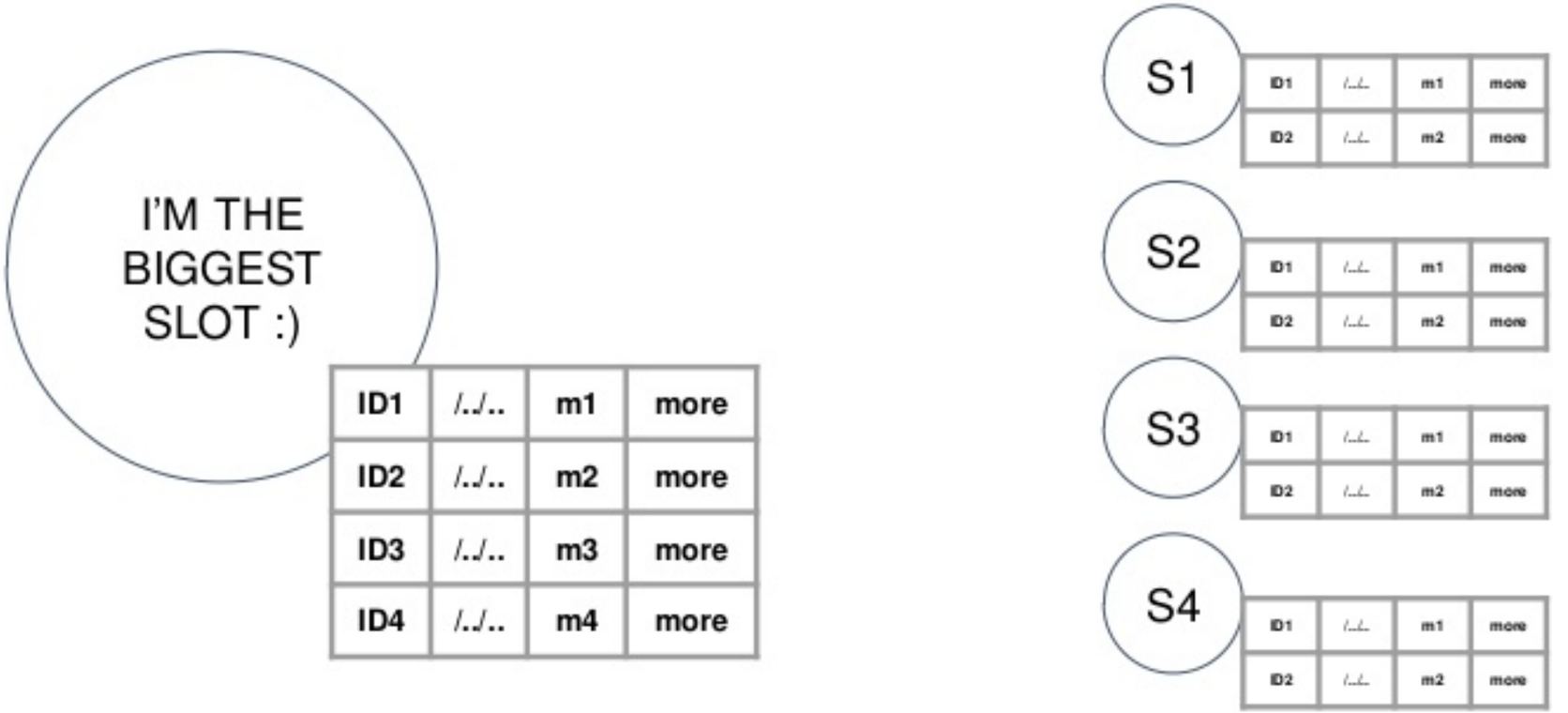
THE CONTROL STREAM

Each slot must be equally configured - **broadcasting**

Metadata Table Concept

RADICALBIT
Deep. Different. Disruptive.

INTEGRATING WITH FLINK STATE BACKEND



we have not *keyed state* - operator state is fine
attempting to keep flink jpmml lightweight on state

We don't checkpoint models but their metadata

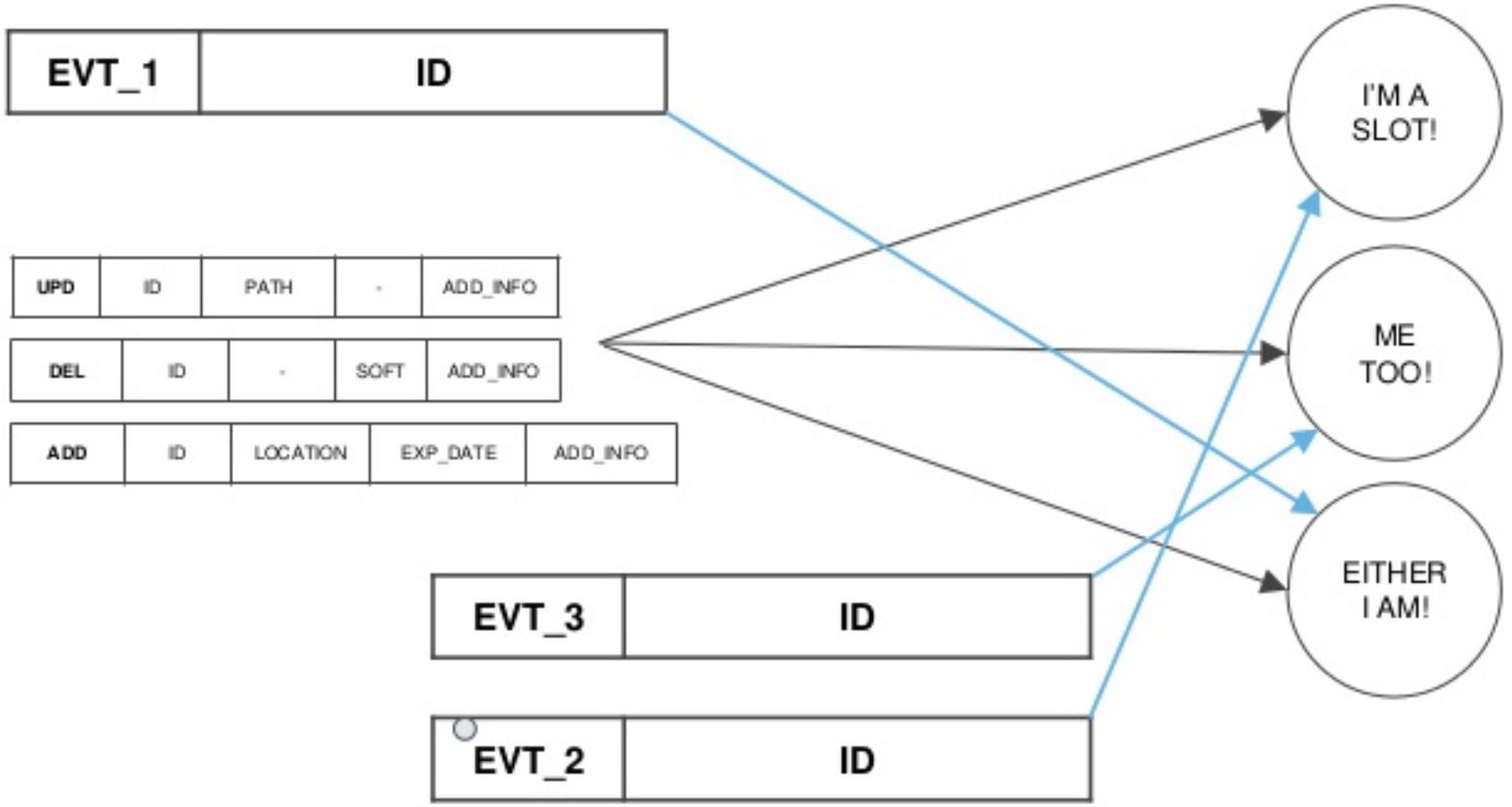
RADICALBIT
Deep. Different. Disruptive.

EVENTS BRING MODEL IDs

No partitioning strategies on main stream

Hashed IDs

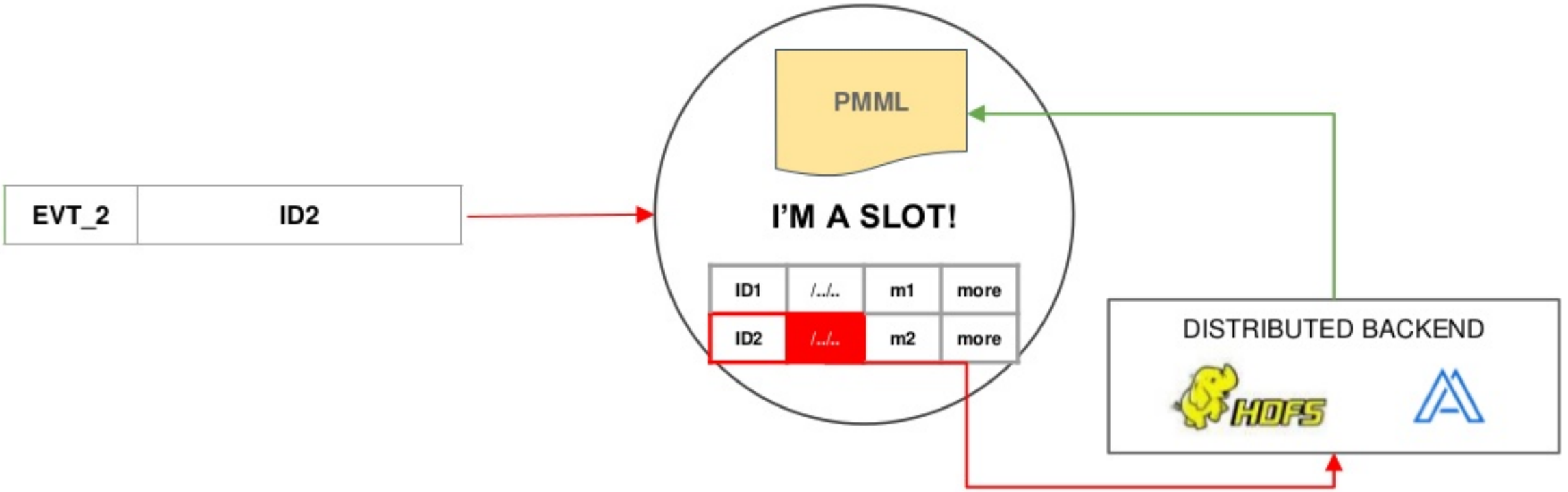
Events declare a conversion function to the internal format



Flink Vectors

RADICALBIT
Deep. Different. Disruptive.

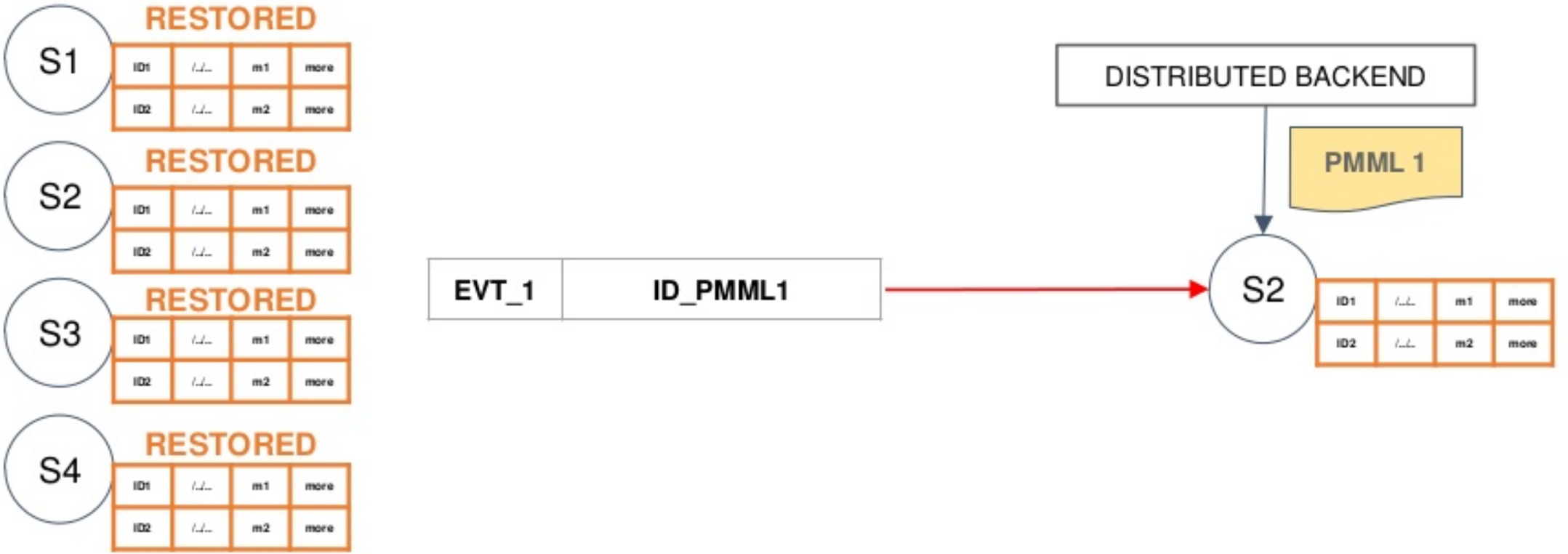
LAZY LOADING FROM DISTRIBUTED BACKEND



Events trigger lazy upload of distributed models by ID

RADICALBIT
Deep. Different. Disruptive.

INTEGRATING WITH LINK STATE BACKEND



On restore, lazy uploading applies models' recovering

RADICALBIT
Deep. Different. Disruptive.

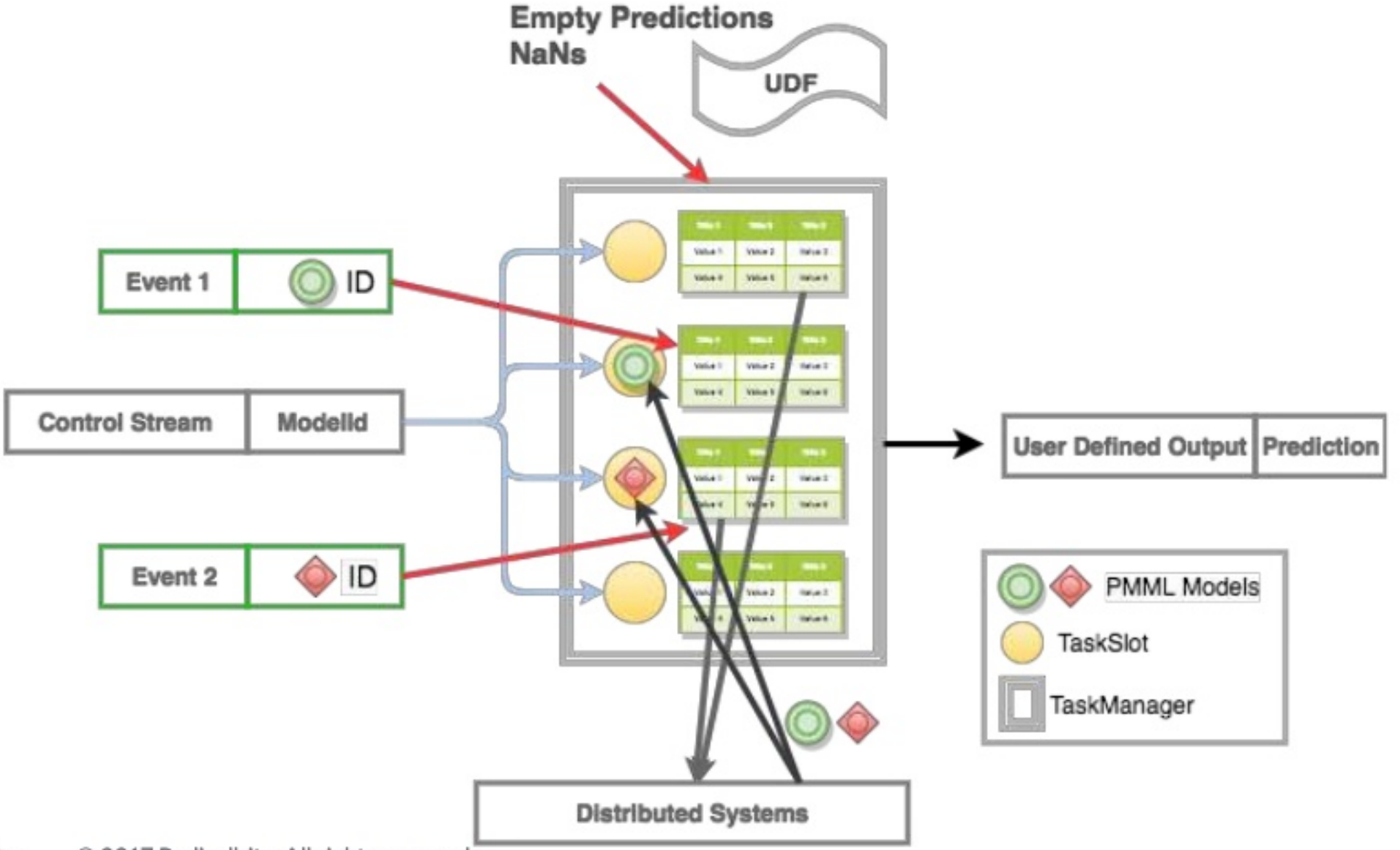
WHY DON'T KEYING THE STREAM?

UPD	ID	PATH	-	ADD_INFO
DEL	ID	-	SOFT	ADD_INFO
ADD	ID	LOCATION	EXP_DATE	ADD_INFO

- the control stream is broadcasted
- events are free to be flink partitioned
- models and dependent events are unambiguously identified - they will meet each other :)

RADICALBIT
Deep. Different. Disruptive.

TO THE NEWER ARCHITECTURE



RADICALBIT
Deep. Different. Disruptive.

FLINK JPMML - WHAT IS GOING NEXT

Contribute to FlinkML

Put the effort together with the community

There are a lot awesome tasks out there

online learning

incremental learning

Keep flink-jpmml better

feature polymorphism

by now flink JPMML is quite chained by numerics

outcome is really task dependent

Keyed Solution

partition the streams if we have a plenty of models: anyway, we should not force user to implement complex distribution kafka partitioning strategies

HANDS-ON

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

Lets quicky approach to the library
https://github.com/spi-x-i/flink-handson

MANY PEOPLE TO THANK

Early Contributors

Stefano Baghino

Simone Robutti

Who joined the flink-jpmml effort

Marco Tagliabue

Gloria Ronzoni

Ali Alerwi

Mauro Cortellazzi

Michele Ridi

Riccardo Diomedi

RADICALBIT
Deep. Different. Disruptive.

ACHTUNG! DON'T MISS NSDB TALK

13/09, 2:30pm - MachinenHaus

NSDB - A time series streaming oriented database optimized for the serving layer

Saverio Veltri, Roberto Bentivoglio @Radicalbit

RADICALBIT
Deep. Different. Disruptive.

AND THANK YOU!

<radicalbit.team/>

info@radicalbit.io

RADICALBIT
Deep. Different. Disruptive.