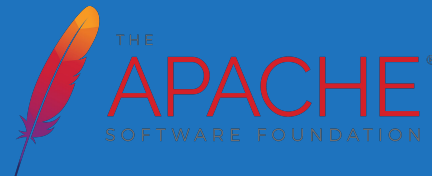


# Moven

Machine/Deep Learning Models Distribution  
Relying on the Maven Infrastructure

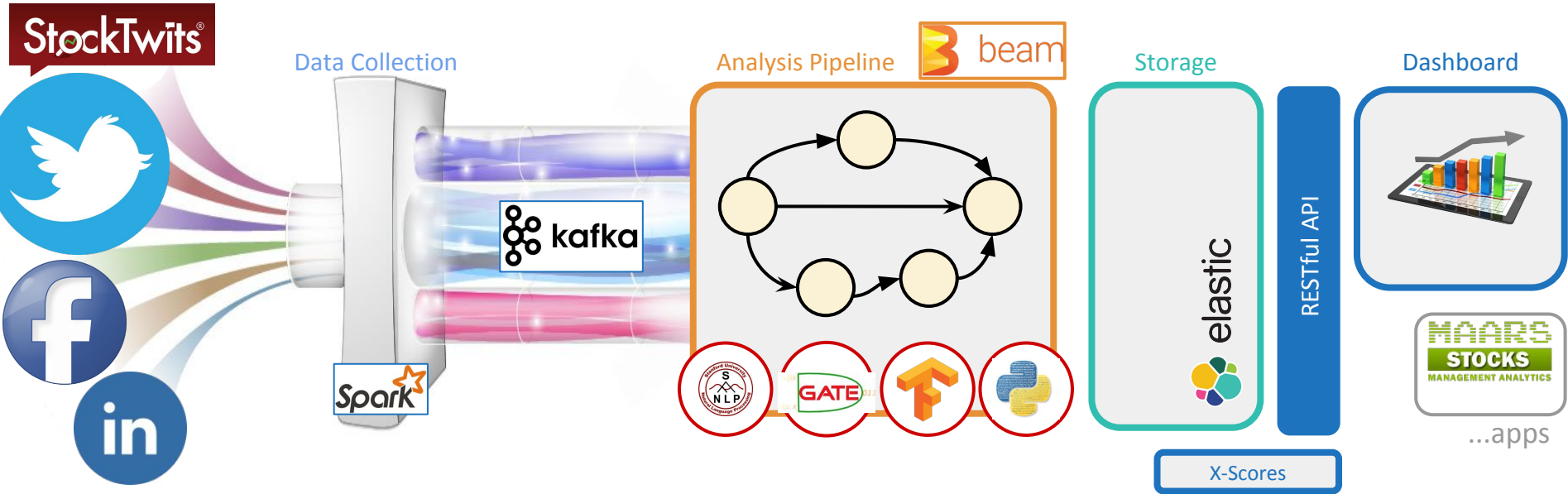
**Sergio Fernández (Redlink GmbH)**

November 14th, 2016 - Sevilla





**SSIX** aims  
to exploit the  
**predictive power**  
of Social Media  
on **Financial**  
**Markets**



Further details at <http://ssix-project.eu/>

# Models in SSIX and Redlink

---

In **Redlink**, particularly in the **SSIX project**, we deal with quite deep neural networks that produce very large models (several Gigabytes).

Therefore we thought how to address two problems:

1. How to properly manage its **distribution** and versioning?
2. How to automatize its **testing**?

moven = models + maven

The thesis why we started to work on Moven was the lack of proper state-of-the-art technology for addressing the two needs described before (distributing and testability).

Some examples:

- TensorFlow public models use a regular git repository
- In Spark ML most of the people use a shared storage (e.g., HDFS)
- OpenNLP also bundle them as JARs
- Freeling uses a share folder from the native installation packages
- Some other proprietary methods...

As Maven does a great work for software artifact, we decided to reuse that infrastructure for models too.

# Moven key features

---

- Model agnostic
- Publication based on a regular Maven plugin
- Distribution relying on the existing Maven infrastructure
  - benefiting of all the features provided by existing tooling (access control, mirroring, etc)
- Retrieval current supported in:
  - Java (Maven of course)
  - Python (relying on [jip](#))
- Built-in gzip-based compression

# Related work

---

There are some interesting work related with our goals:

- StandardNLP has recently (>3.5.2) changed to bundle the modules as Maven artifacts
- [TensorFlow Serving](#) helps to deploy new algorithms for TensorFlow models
- [PipelineIO](#) combines several technologies (Spark, NetflixOSS, etc; they call it the *PANCAKE STACK*) to provide models distribution, including incremental training, among many other features ([more details](#)).

# Publish Moven models

Create a regular Maven artifact, placing the models at `src/main/models`, just including a plugin configuration:

```
<plugin>
  <groupId>io.redlink.ssix.moven</groupId>
  <artifactId>moven-maven-plugin</artifactId>
  <version>0.1.0-SNAPSHOT</version>
  <executions>
    <execution>
      <phase>process-resources</phase>
      <goals>
        <goal>copy-models</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

Then normally deploy your artifacts with `mvn deploy`



# Using your Moven models

## From Java:

- Declare the dependency to your models at your `pom.xml`
- Then models will be available in your classpath:

```
this.getClass().getClassLoader()  
    .getResourceAsStream("META-INF/resources/models/foo.ex")
```

- Also exposed via HTTP as static resources when the JAR is deployed in any Servlet  $\geq 3.0$  container (inspired by James Ward and the [WebJars](#) project).

## From Python:

- Install it: `pip install moven`
- Declare at `models.txt` your models in your project (as we do with `requirements.txt`) with a syntax similar to Groovy's Grape:

```
io.redlink.ssix.moven:moven-syntaxnet-example:1.0-SNAPSHOT
```

- Execute `moven models.txt` to retrieve all models to `./moven` organized by `artifactId`.
- Thought-out specifically for container deployments

# let's play



# Current status and future

---

Moven is still in a very early stage, but already being used in production in SSIX and other Redlink projects.

We will keep exploring such approaches to find a way to better manage the lifecycle of the models that drive our information extraction (Natural Language Processing, Machine Learning, Deep Learning, etc) stack.

For example, we want to target more specific needs in some concrete environments, such as **Apache Spark** and/or **Apache Beam** Runners API.



**Gracias!**



# Sergio Fernández

Software Engineer

[sergio.fernandez@redlink.co](mailto:sergio.fernandez@redlink.co)

<https://www.wikier.org/>

## Redlink GmbH

<http://redlink.co>

Coworking Salzburg

Jakob Haringer Straße 3

5020 Salzburg (Austria)



SOCIAL SENTIMENT INDEX

POWERED BY X-SCORES

project partially funded by the European Union's Horizon 2020 research and innovation programme, under grant agreement no. 645425

