AI and DeepLearning Architectures

IBM Watson IoT™ Romeo Kienzler

In this module we will cover AI and DeepLearning Architectures

Source: buggi - openclipart.org

Watson IoT™ Romeo Kienzler

When it comes to Machine Learning, there is nothing else you should try to increase than the amount of good data available for training. Please note that I'm talking about good data. So this especially holds for DeepLearning. So it's always a good idea to collect and store as much data as possible. And one important cloud based solution is called object store.

- **Unlimited**

Source: buggi - openclipart.org

So in Object Store you have Unlimited storage capacity

- **Unlimited**
- **Redundancy**

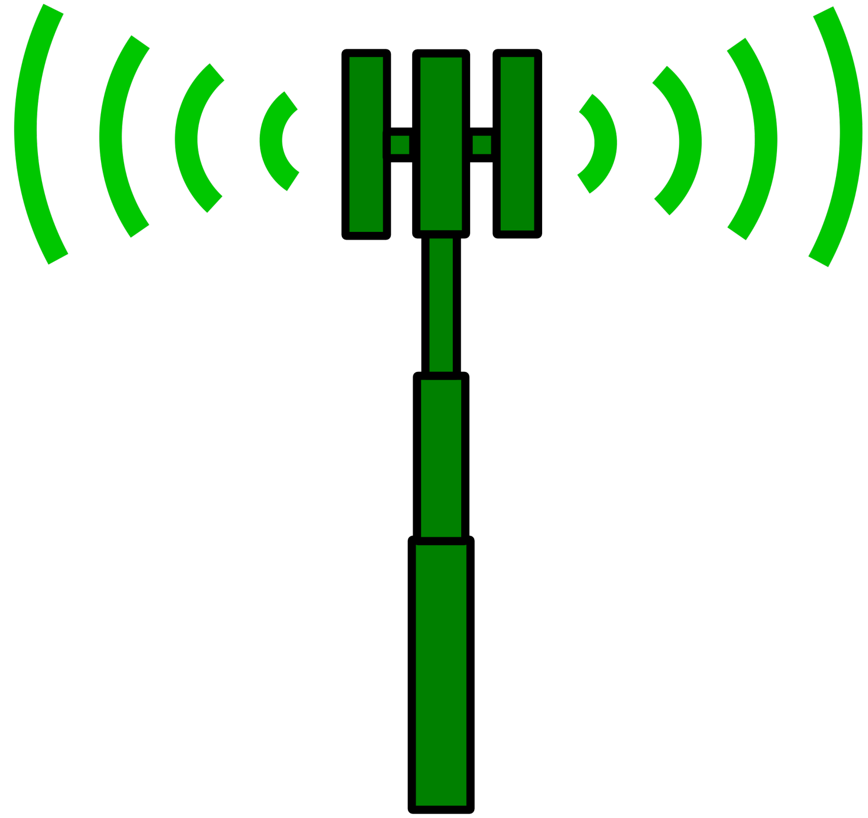
Source: buggi - openclipart.org

redudancy, fail over and automated backup

- **Unlimited**
- **Redundancy**
- **Performance**

Source: buggi - openclipart.org

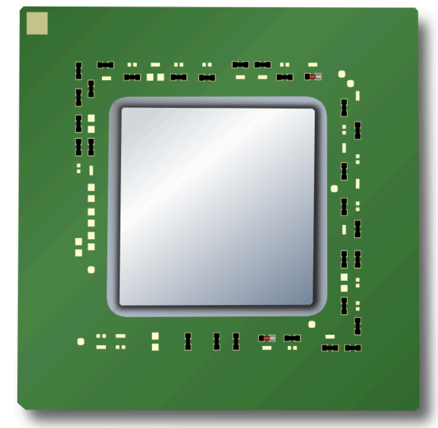
and the best I/O performance. Therefore we make sure, that during this course you will be able to train a neural network directly from data residing in an object store



Source: GR8DAN - openclipart.org

IBM Watson IoT™ Romeo Kienzler

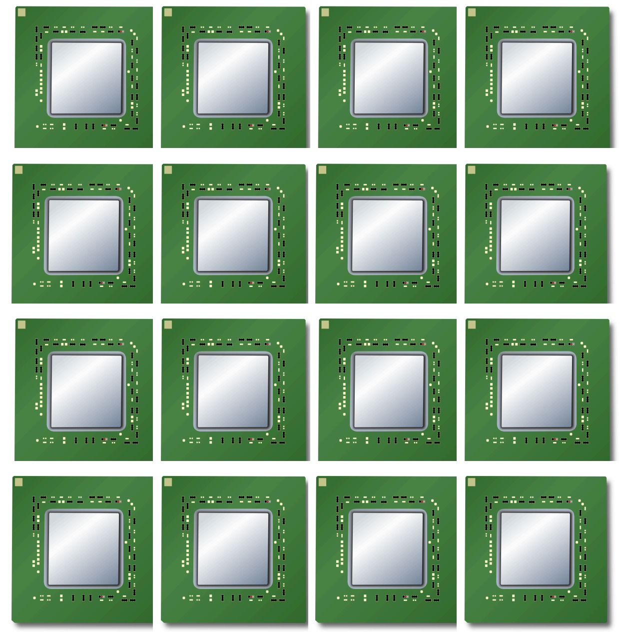
Another aspect is real-time data processing. Often data looses value just after a couple of seconds. Think of stock market data for example. Therefore we also make sure you understand how to apply and deploy deep learning models on a real-time data stream



Source: ivak - openclipart.org

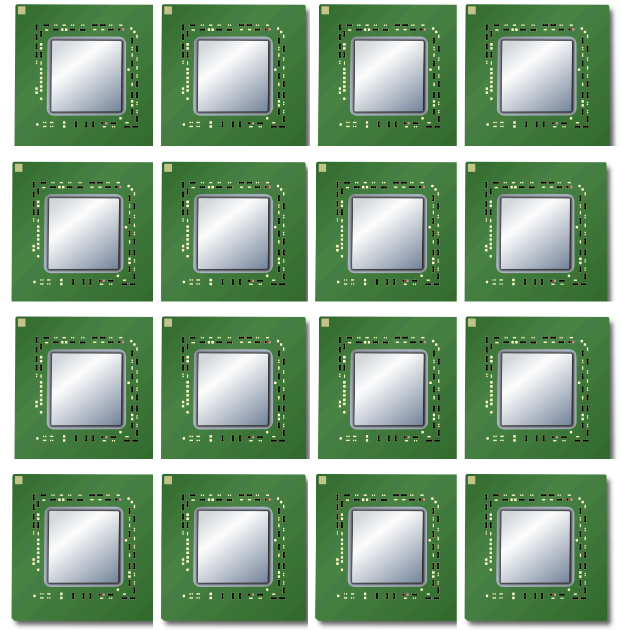
IBM Watson IoT™ Romeo Kienzler

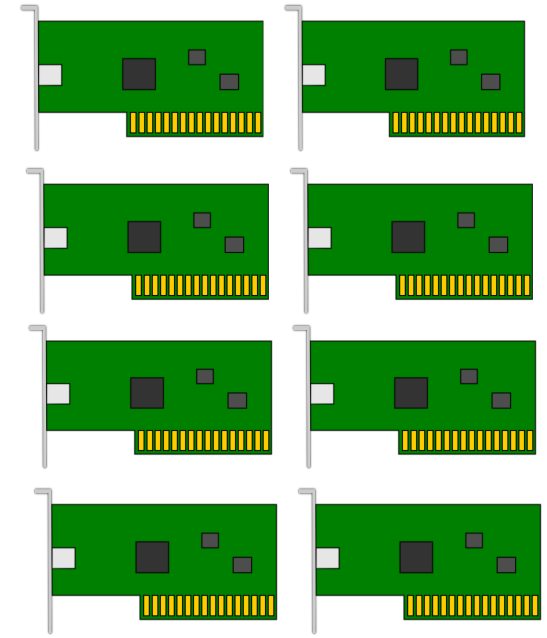
Finally, those models are making heavy use of CPU and GPU hardware.



Source: ivak - openclipart.org

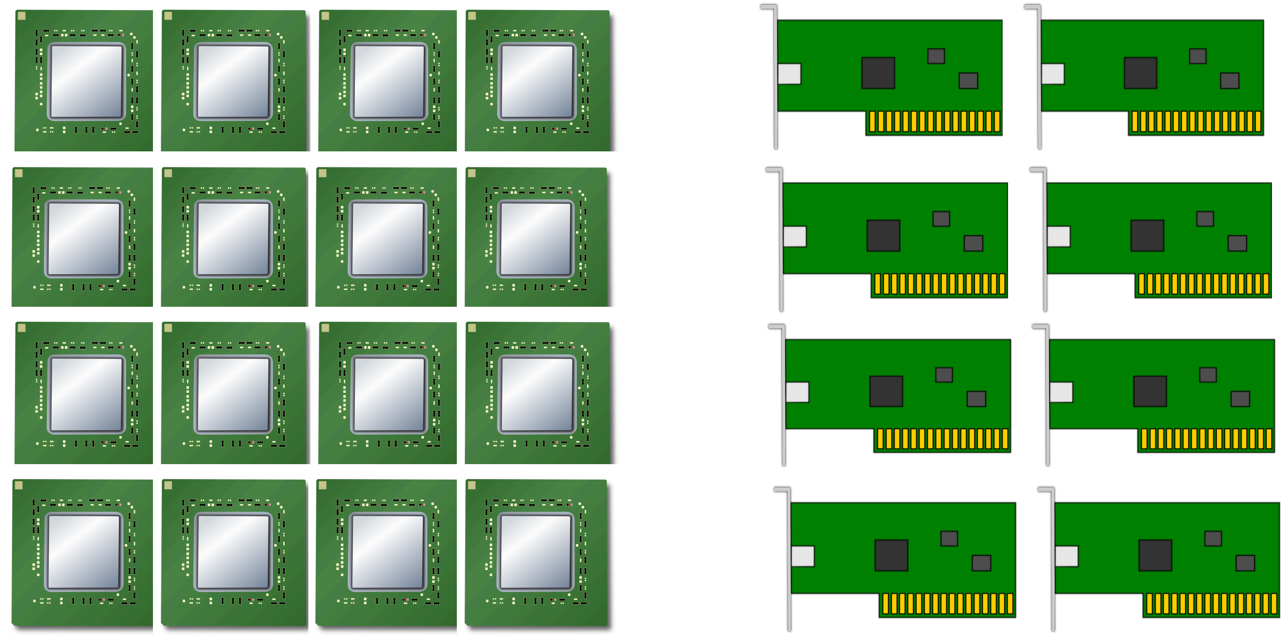
Therefore we'll also explain how to scale deep learning models on CPU ….





Source: ivak - openclipart.org

….and GPU clusters.



Source: ivak - openclipart.org

….and GPU clusters.

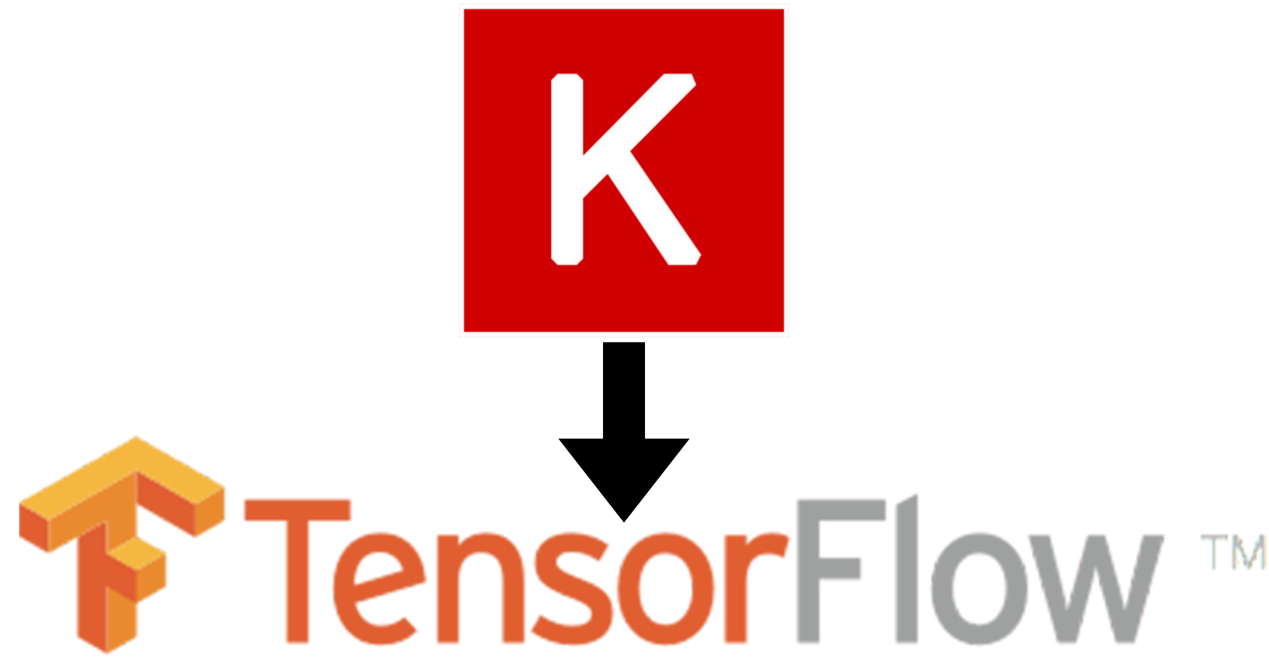
Source: ivak - openclipart.org

But that's only part of the story. Before we can deploy a DeepLearning model we actually have to create it. In this course we'll use jupyter notebooks for all the coding and we'll use a cloud based version of it, …

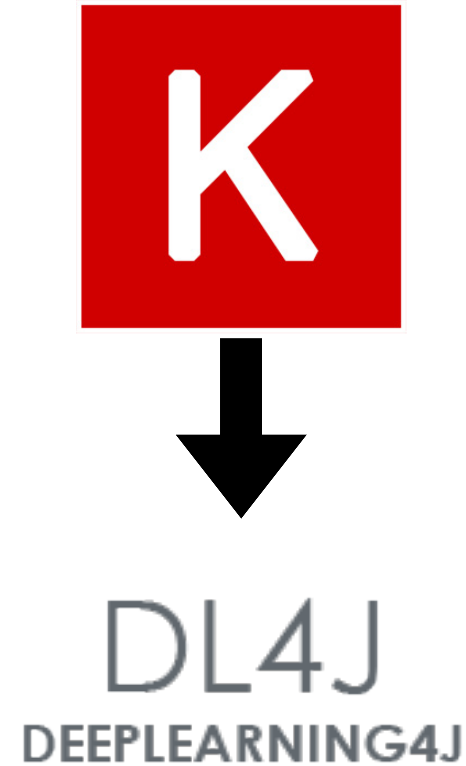
Source: ivak - openclipart.org

… called Data Science Experience - so no need for installing any software on your machine.

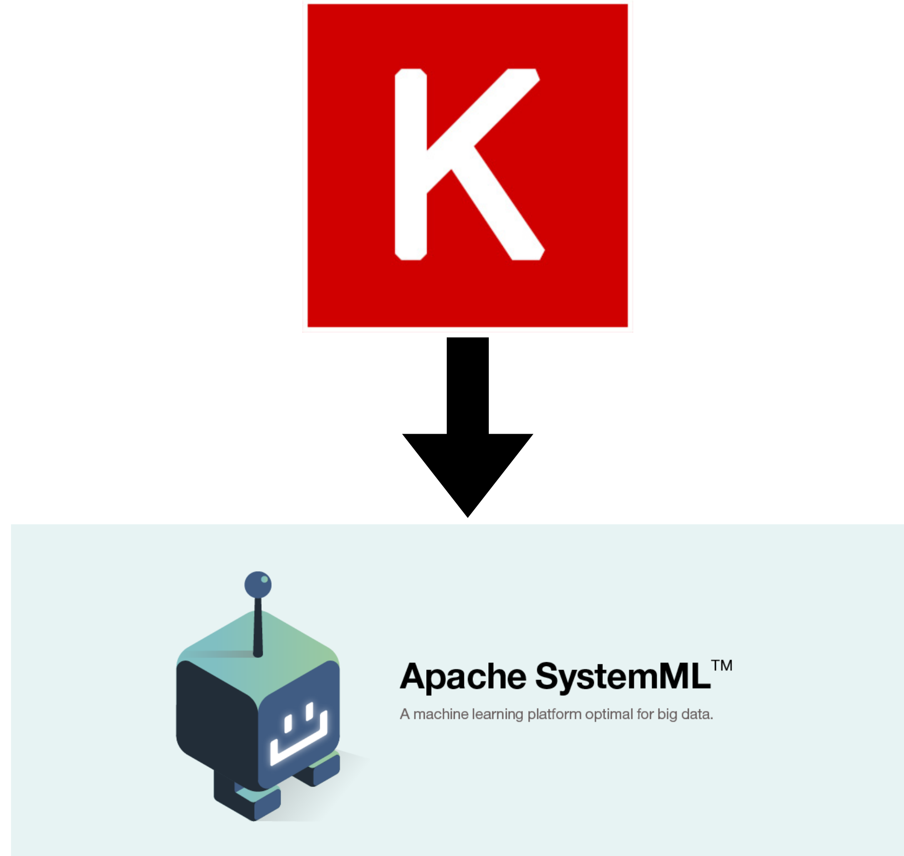
No DeepLearning without a DeepLearning framework. And currently it’s pretty hard to choose the optimal framework for a given task. There exist high level frameworks where you simple define neural network layers but also low level framework where you have to implement everything on a linear algebra level. Unless explicitly needed we don’t recommend using low level frameworks at all. Therefore we will use Keras as high level framework whenever possible. You can implement a deep learning neural network from scratch within minutes using Keras. The other good thing is that Keras is using low level deep learning frameworks for execution.



So per default Keras uses TensorFlow as execution engine. In addition CNTK and Theano are also supported. By the way, theano has been discontinued. So there is only TensorFlow and CNTK left.



In addition, Keras models can be exported. And those models can be read by other frameworks as well. In this course we'll use DeepLearning4J



And Apache SystemML as possible runtimes in addition to TensorFlow. We'll introduce all this frameworks later in the course. Keras exports are no open standard, but nevertheless a lot of frameworks support is as an input format.

ONNX

 Romeo Kienzler

When talking about open neural network interchange formats we have to briefly mention ONNX which tries to establish an open standard. Currently Caffe2, The Cognitive Toolkit, MXNET and pytorch are supported. But non of those frameworks can be considered high level, therefore we stick to Keras.

 Romeo Kienzler

Finally, as execution environment for parallel execution we'll use Apache Spark. Apache Spark is the de-facto standard for big data processing in academic and enterprise contexts and also available as a service in the cloud from different providers. Data pre-processing is much more straightforward in apache spark then for example in plain TensorFlow.

IBM Watson Machine Learning

 Romeo Kienzler

Finally, we'll use Watson Machine Learning for Model Deployment on GPU clusters. This is a commercial offering from IBM since there doesn't exist an open source alternative and it nicely illustrates how automatic and scalable deep learning model deployment has to look like. Again, as always, there exists a free tier - so anyone can use all these services without paying.

So this leads us to the final architecture.

We will use jupyter notebooks

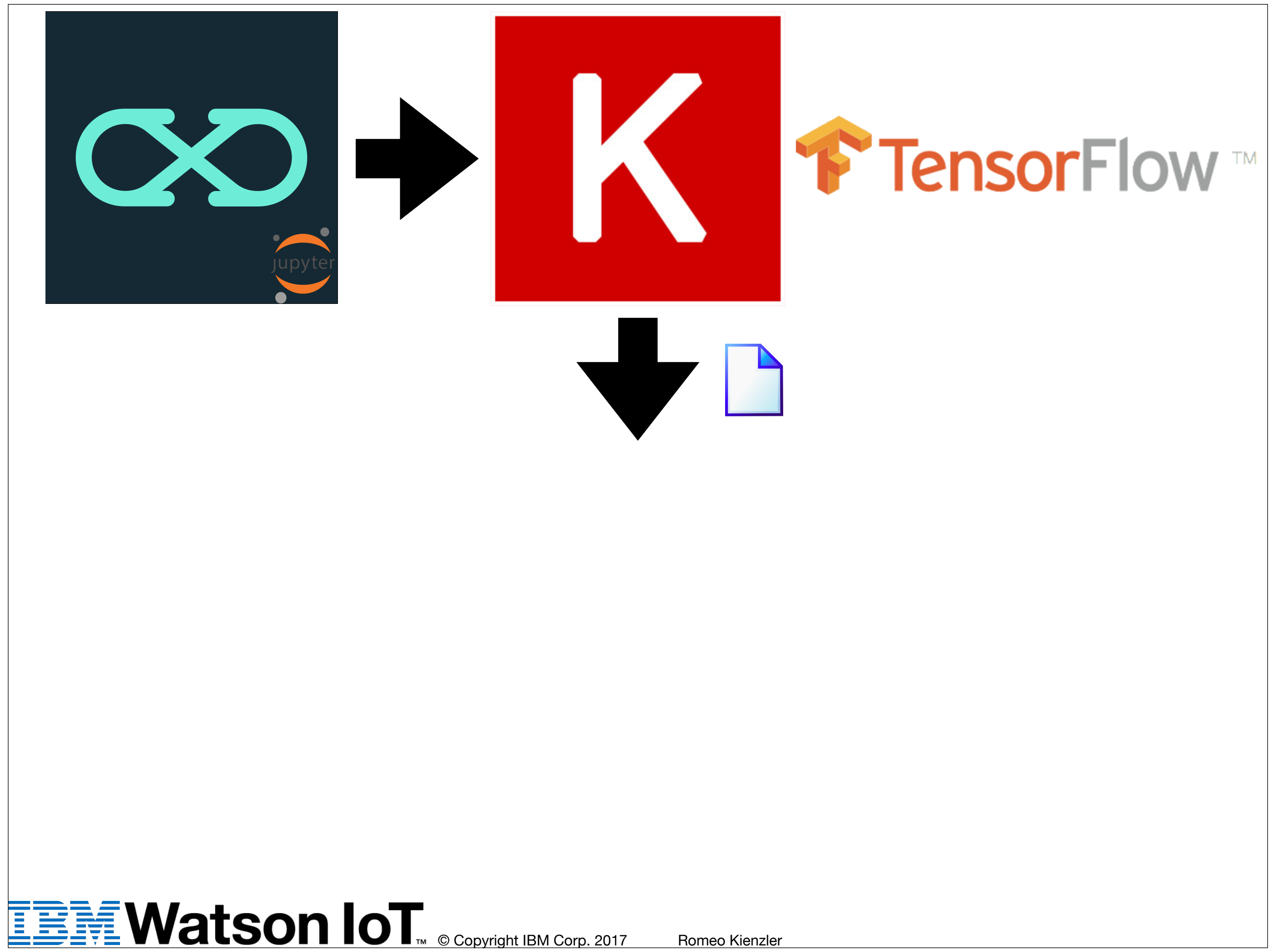
on top of data science experience for coding.

  Romeo Kienzler

We’ll use the Keras framework..

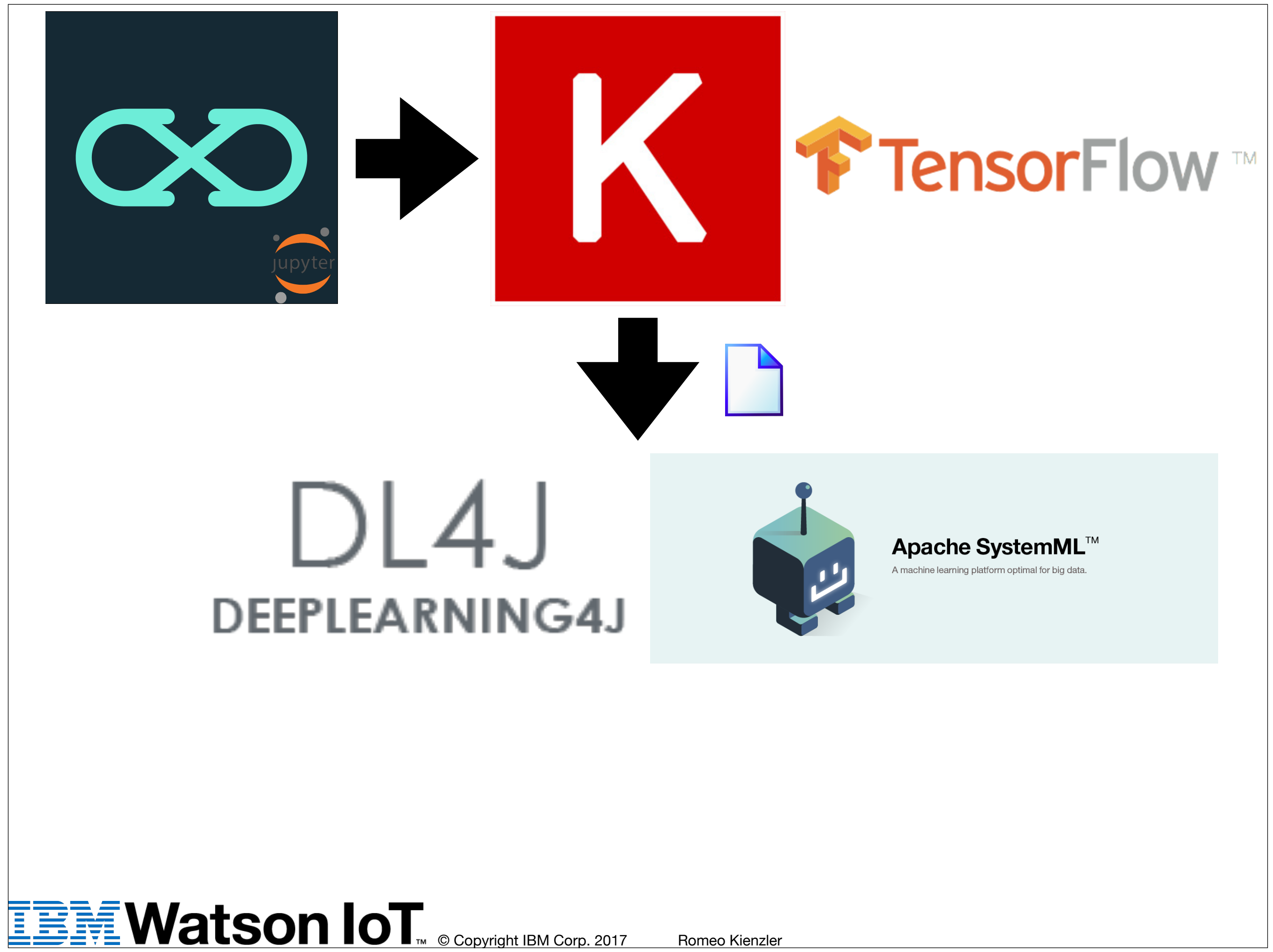
 Romeo Kienzler

in conjunctions with TensorFlow for development and testing of the models.

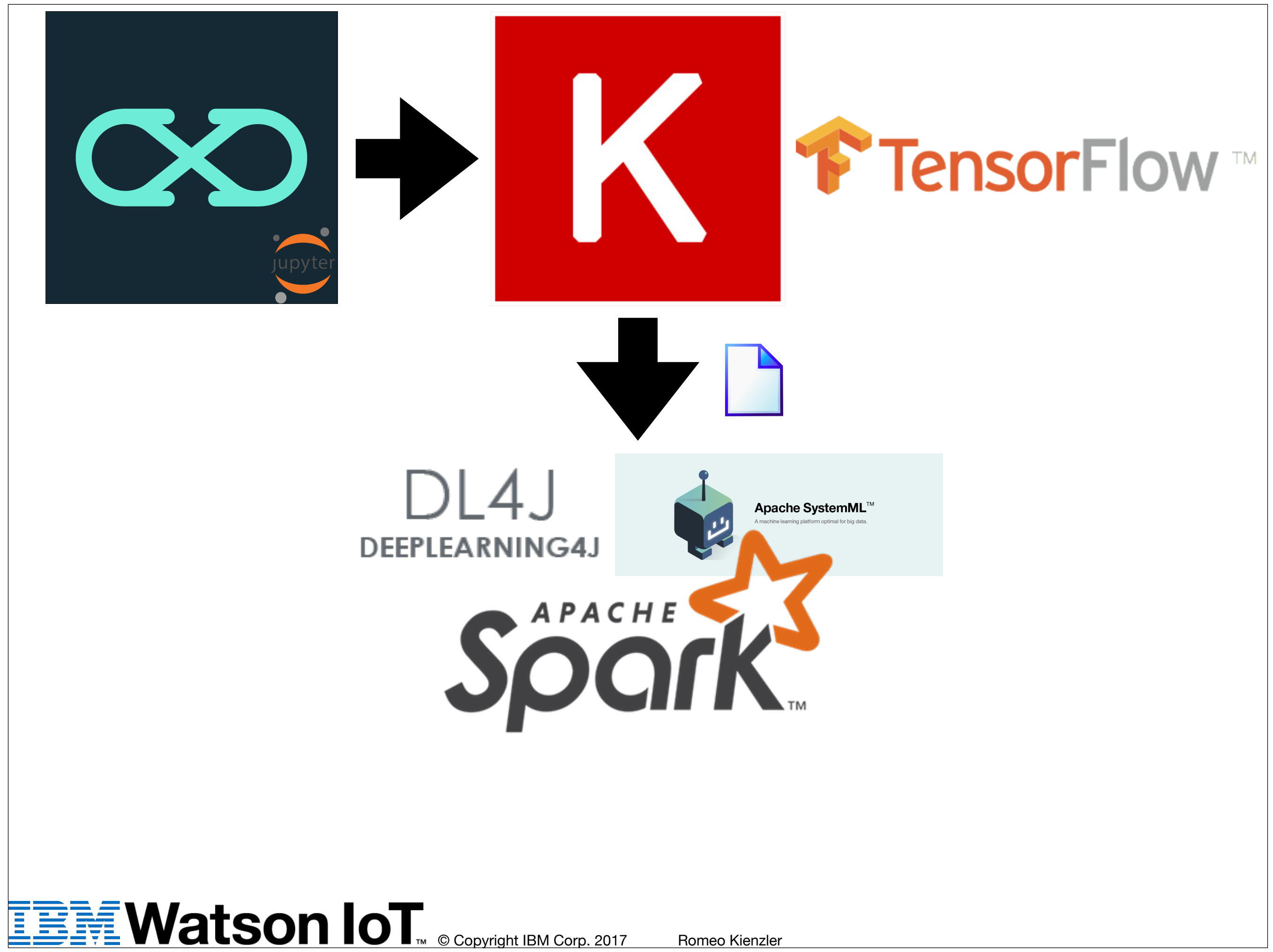


If we are happy with the model we export it and run it on either

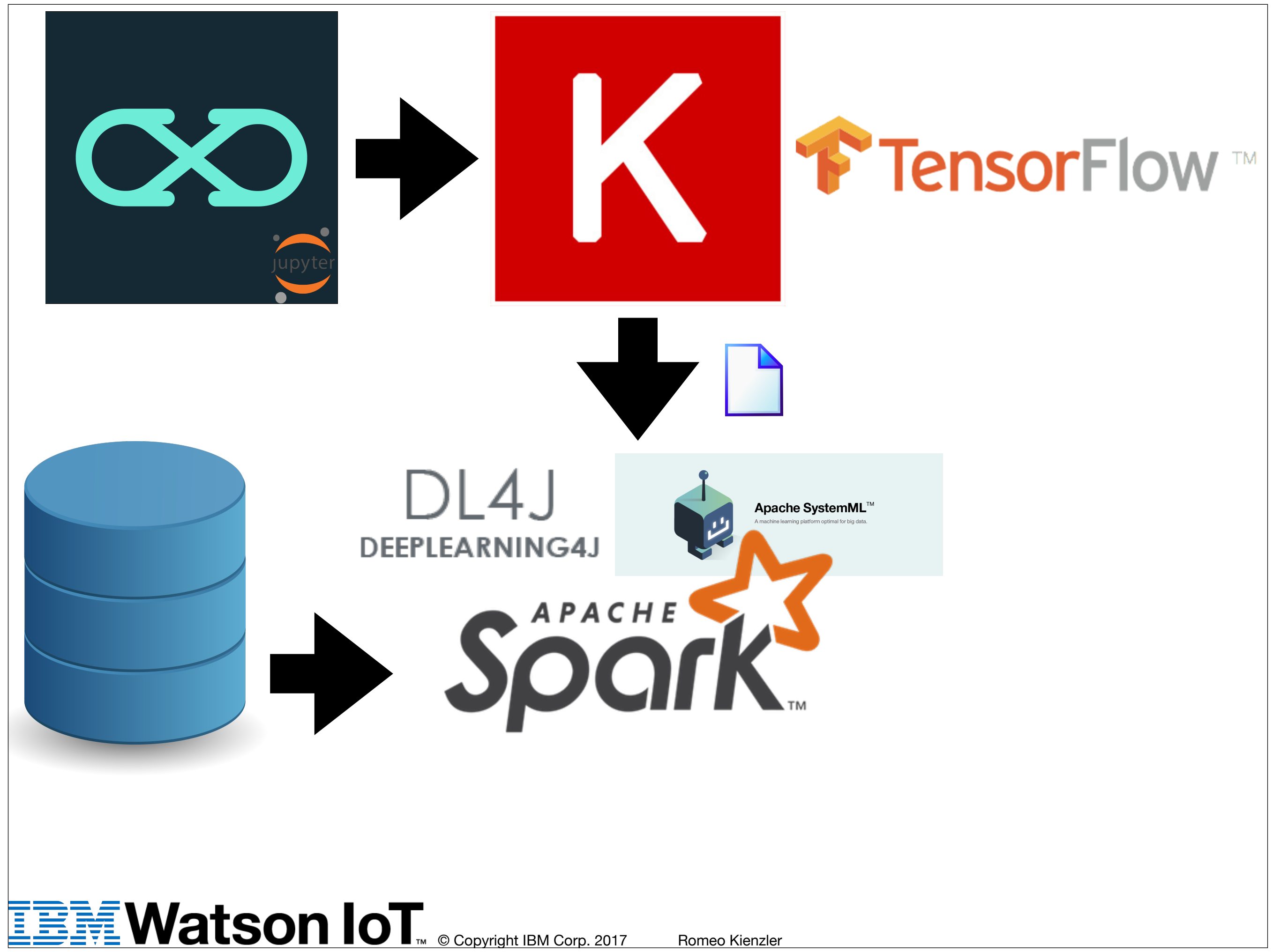
DeepLearning4J or



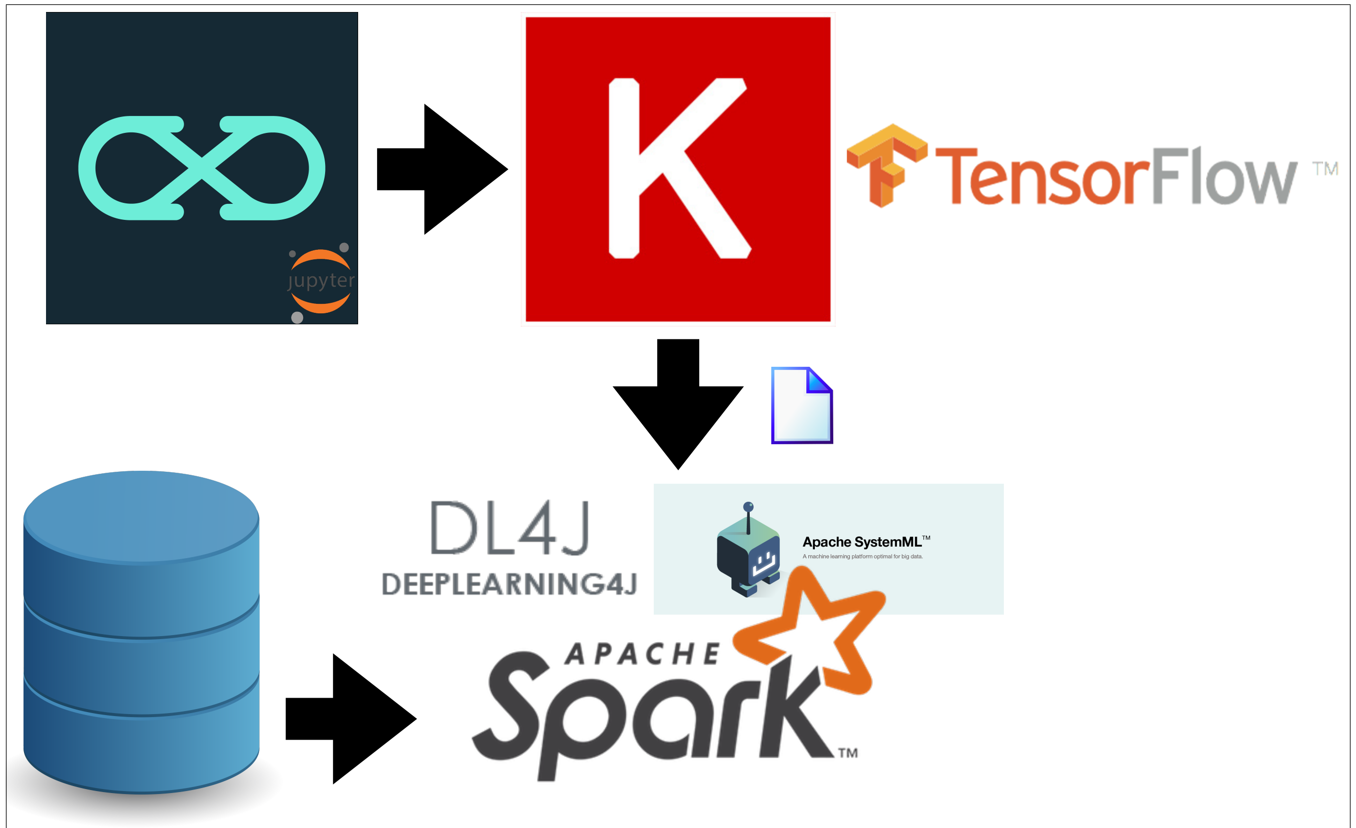
Apache SystemML



Both frameworks are capable of running neural networks in parallel on top of Apache Spark

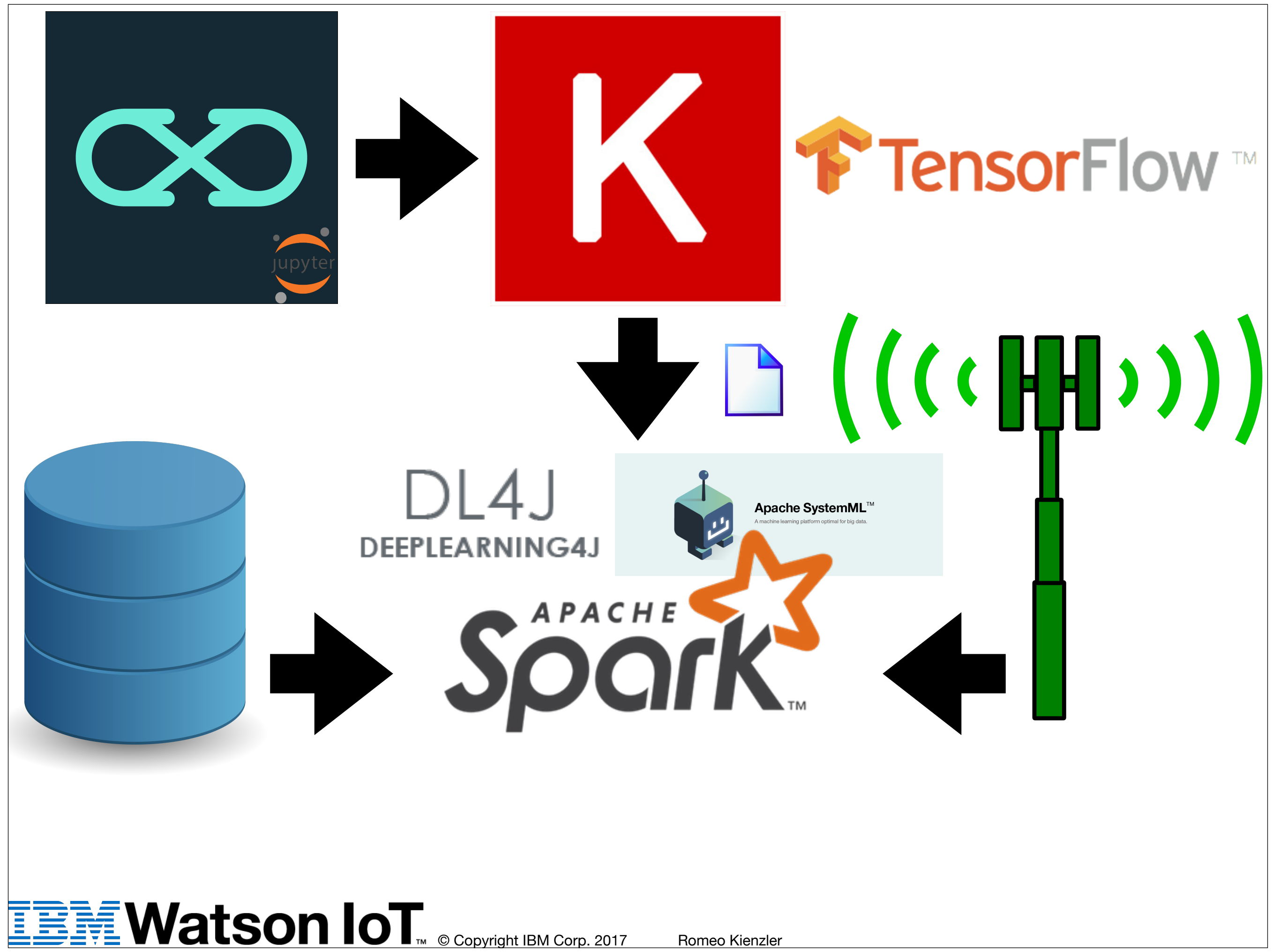


And in both frameworks we can work on petabytes of data residing ObjectStore

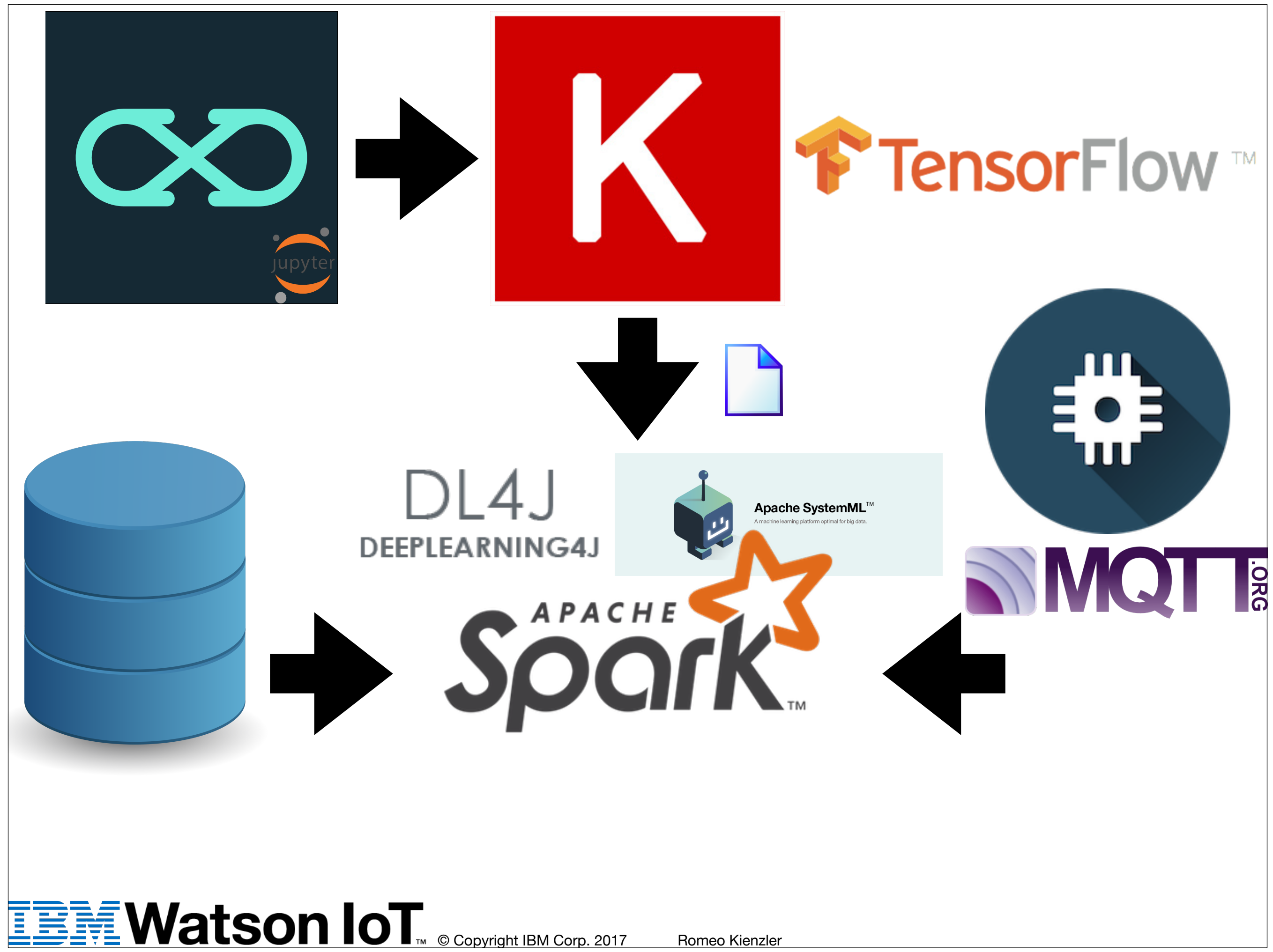


  Romeo Kienzler

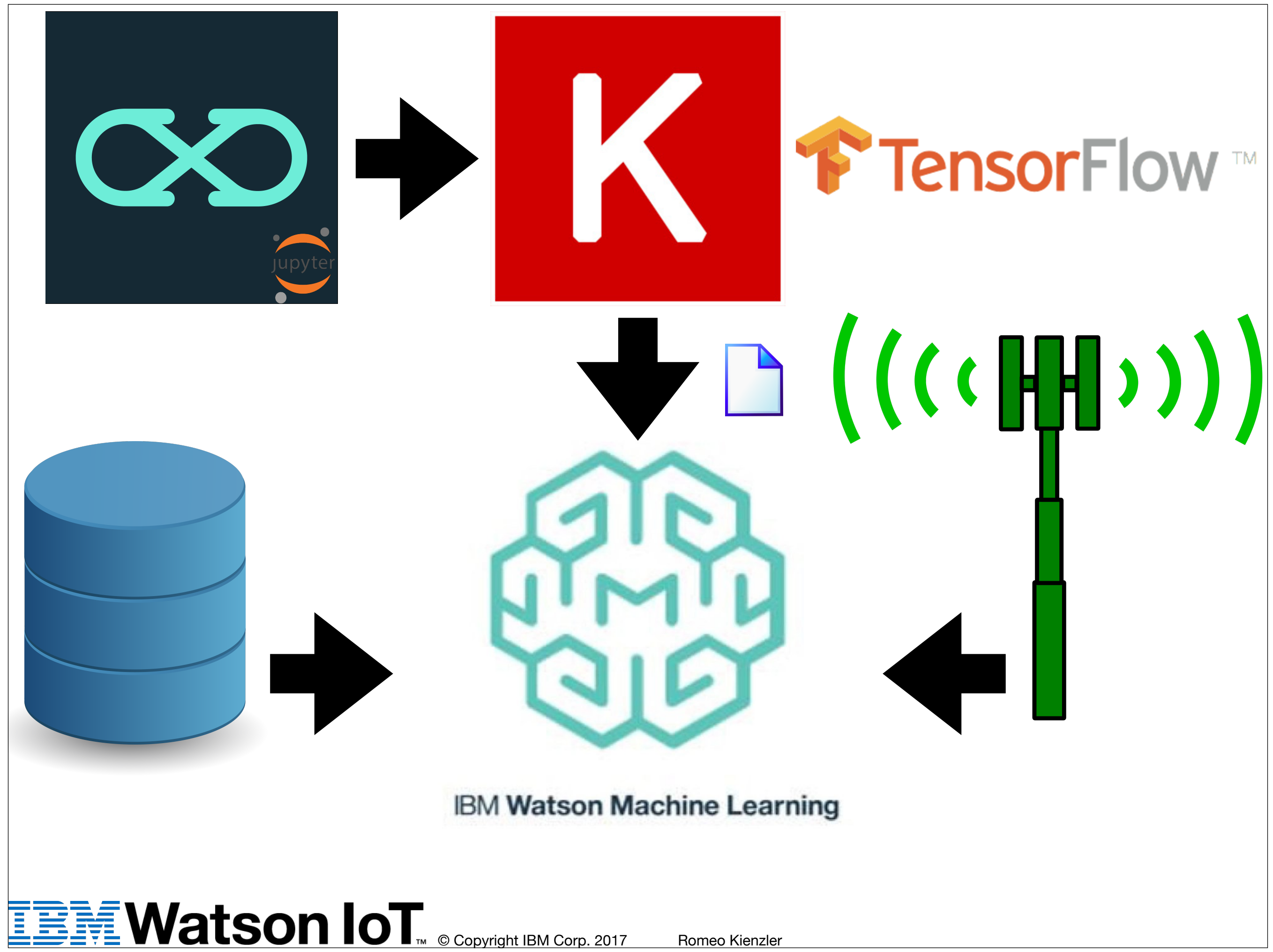
And in both frameworks we can work on petabytes of data residing ObjectStore



..or work on live streams of data in the gigabit range



We'll exemplify this by subscribing to topics on the MQTT message broker provided by the Watson IoT Platform



Instead of Apache Spark we'll also see how to use Watson Machine Learning to deploy such models for nearly unlimited throughput and scalability.

Summary

This way TensorFlow is capable of automatically compute the derivative of ANY function defined as a TensorFlow graph. So we only need to come up with a creative idea of a model and TensorFlow does the rest. Isn't that great?