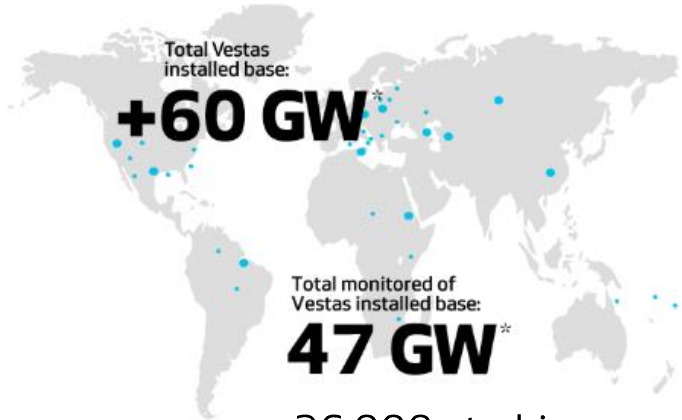
Industrial Big Data Analytics for Wind Turbines

UseR! 2015 Ålborg

Sven Jesper Knudsen - sjknu@vestas.com

Kim-Emil Andersen & Martin Qvist

Vestas Performance & Diagnostic Centre (VPDC)



Vestas Installed base:
+60 GW*

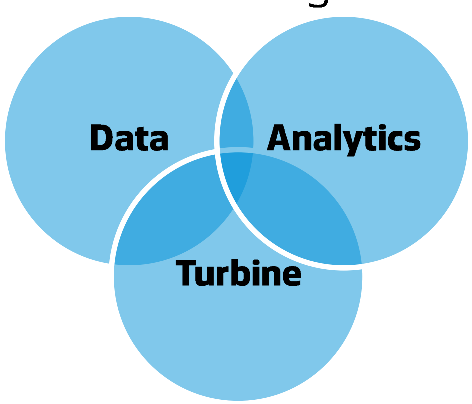
Worldwide installed wind capacity:
+318 GW**

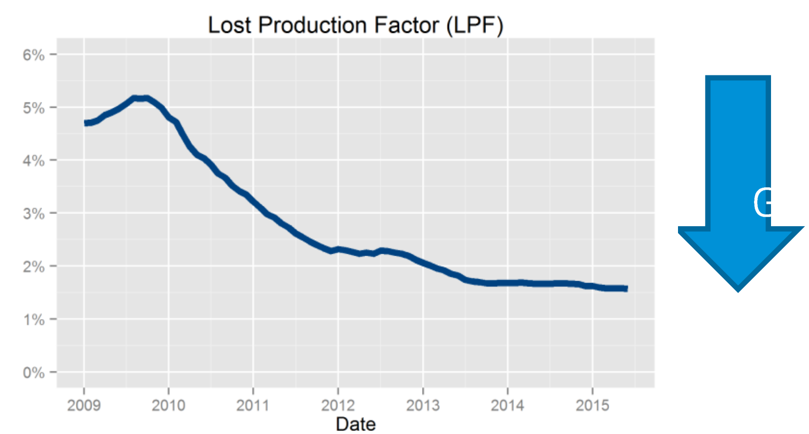
*Vestas Performance & Diagnostics Centre, 31/12/13
**Global Wind Statistics 2013 (Year-end), GWEC

26.000+ turbines are monitored
55.000 date

Vestas Wind Turbine Data Science

- A Decade of data driven Asset Management
- Predictive Maintenance
- Condition Based Monitoring





Vestas UseR!

Desktop

RStudio is on our Vestas Software portal

Markdown reports to clients

TIBCO Spotfire + R Apps users

SQL, Py, .NET

Vestas' HPC, FireStorm

15000 cores, 2.5+ petabyte, 40Gb infiniband interconnected

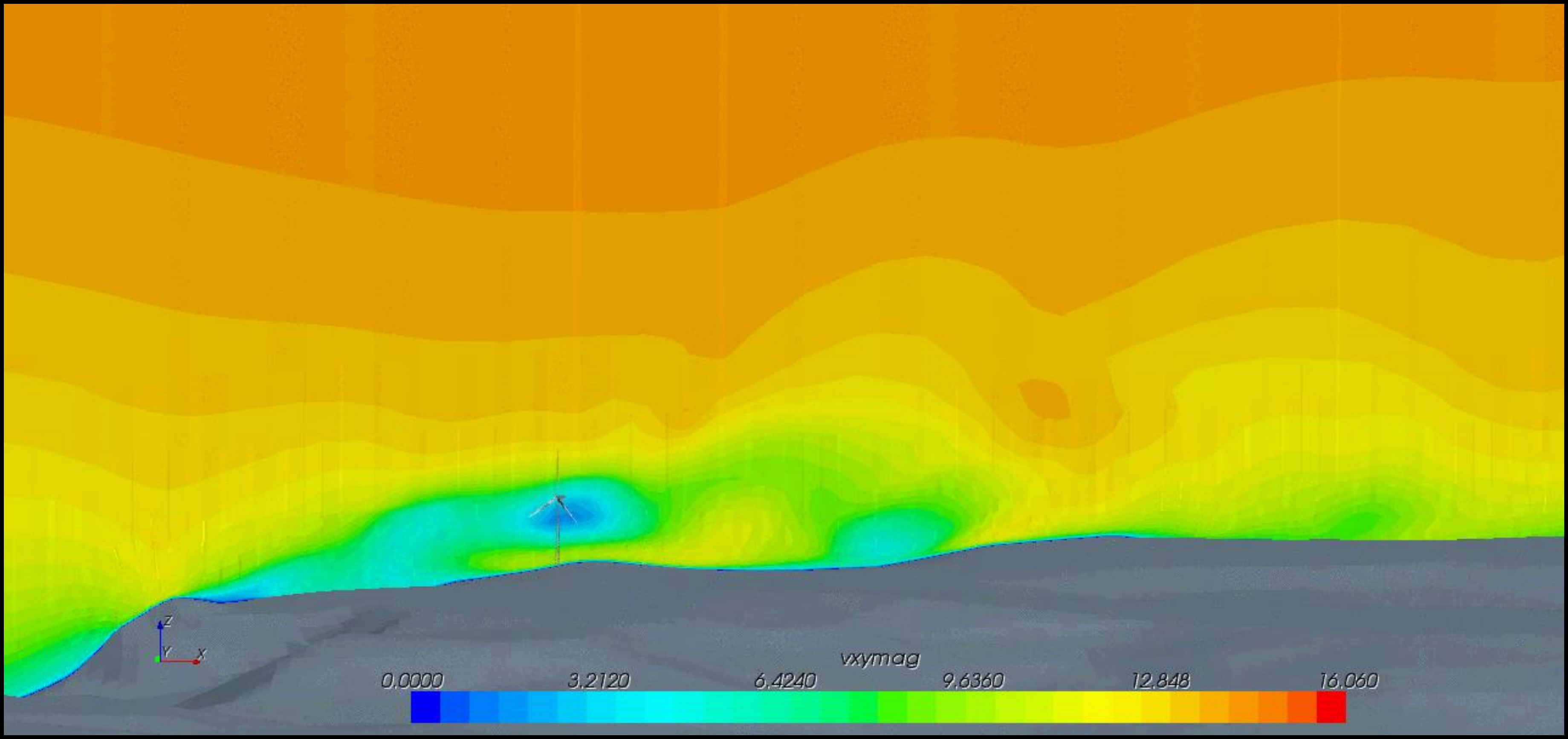
mesoscale climate model, petabyte storage

Computational Fluid Dynamics, high performance

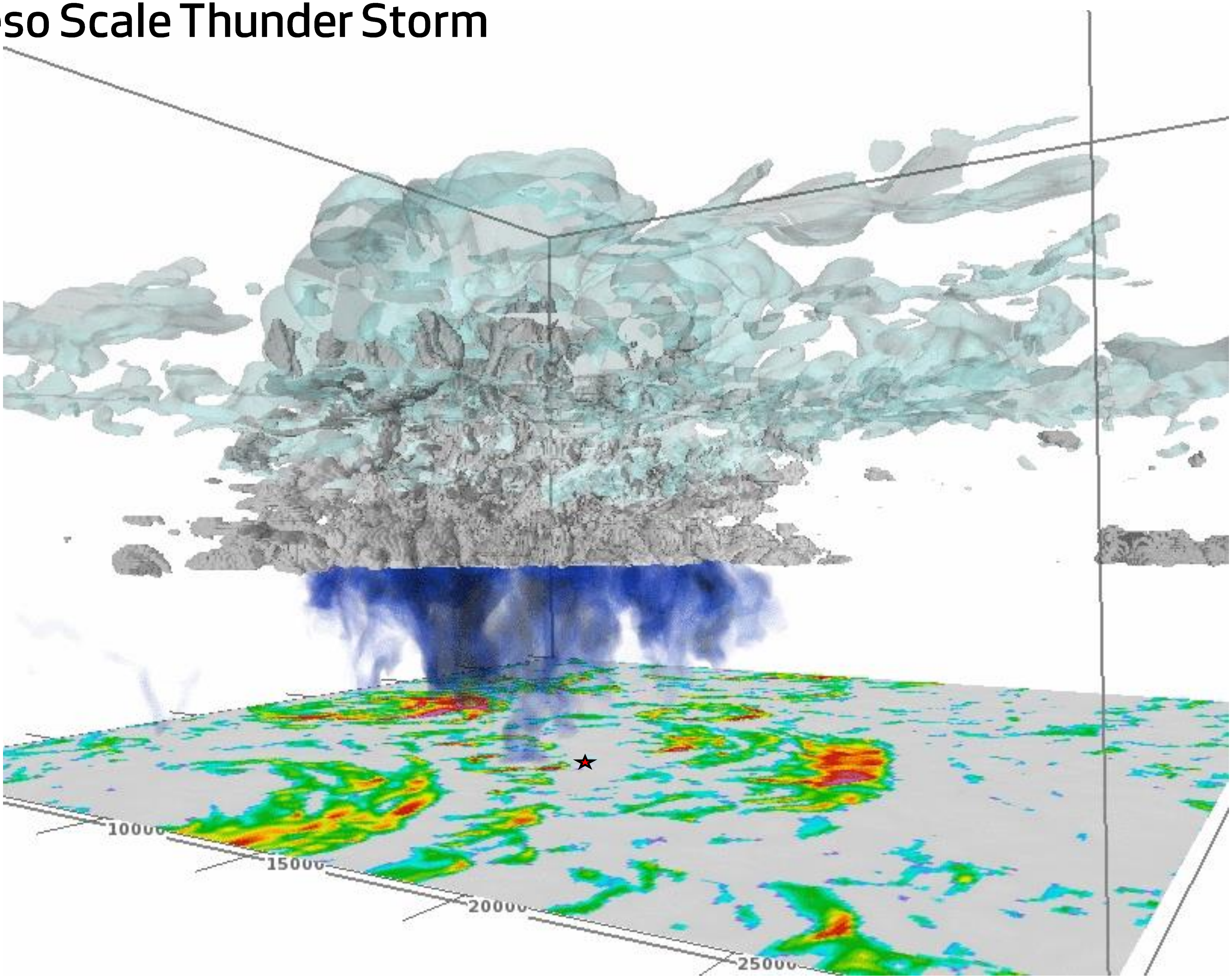
Today focus on HADOOP+HIVE+R

Open Source and Open Standards

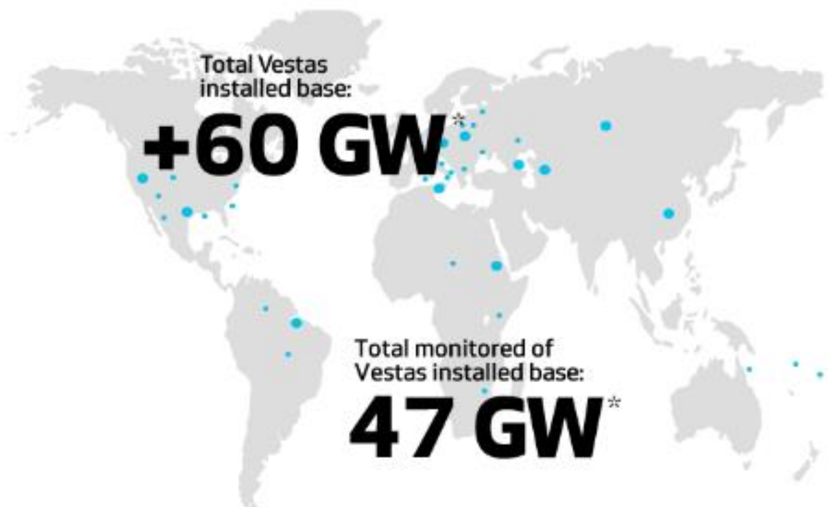
CFD Wind Flow

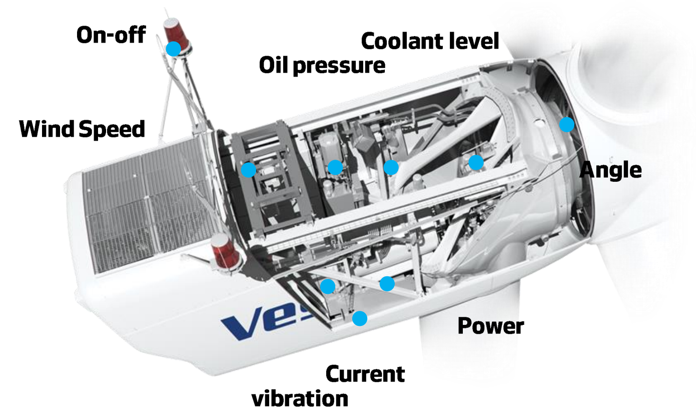


Meso Scale Thunder Storm



Industrial Big Data at Vestas





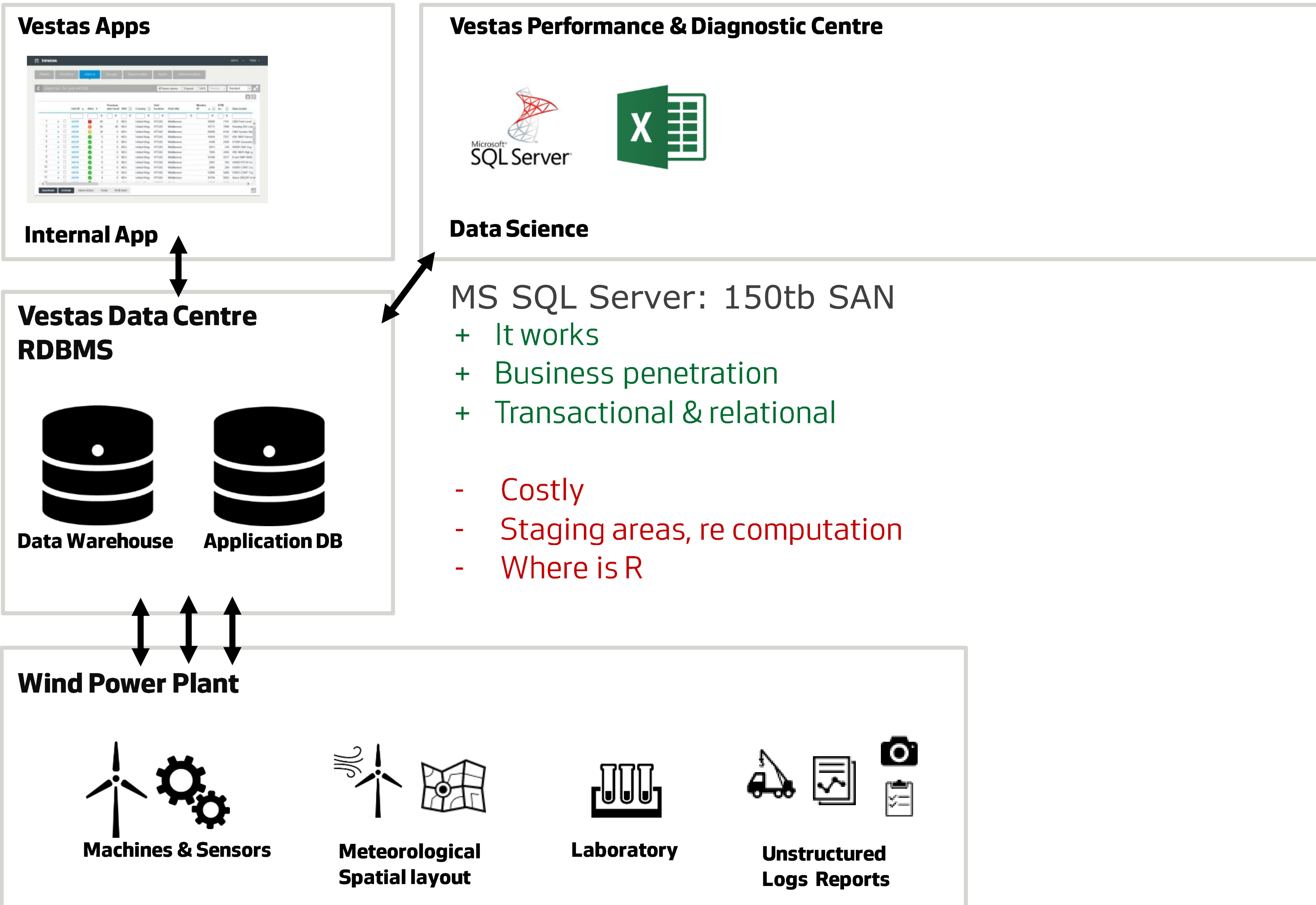
Volume 50.000 turbines, ca 150terabyte
Vestas Climate Lib, ca petabyte

Variety 500+ sensors, vibration 'audio', event logs
image, lab test, service text

Velocity Real time, near time, batch

Varacity Sensitive and diverse sensors, parameters, humans …

Data Platform

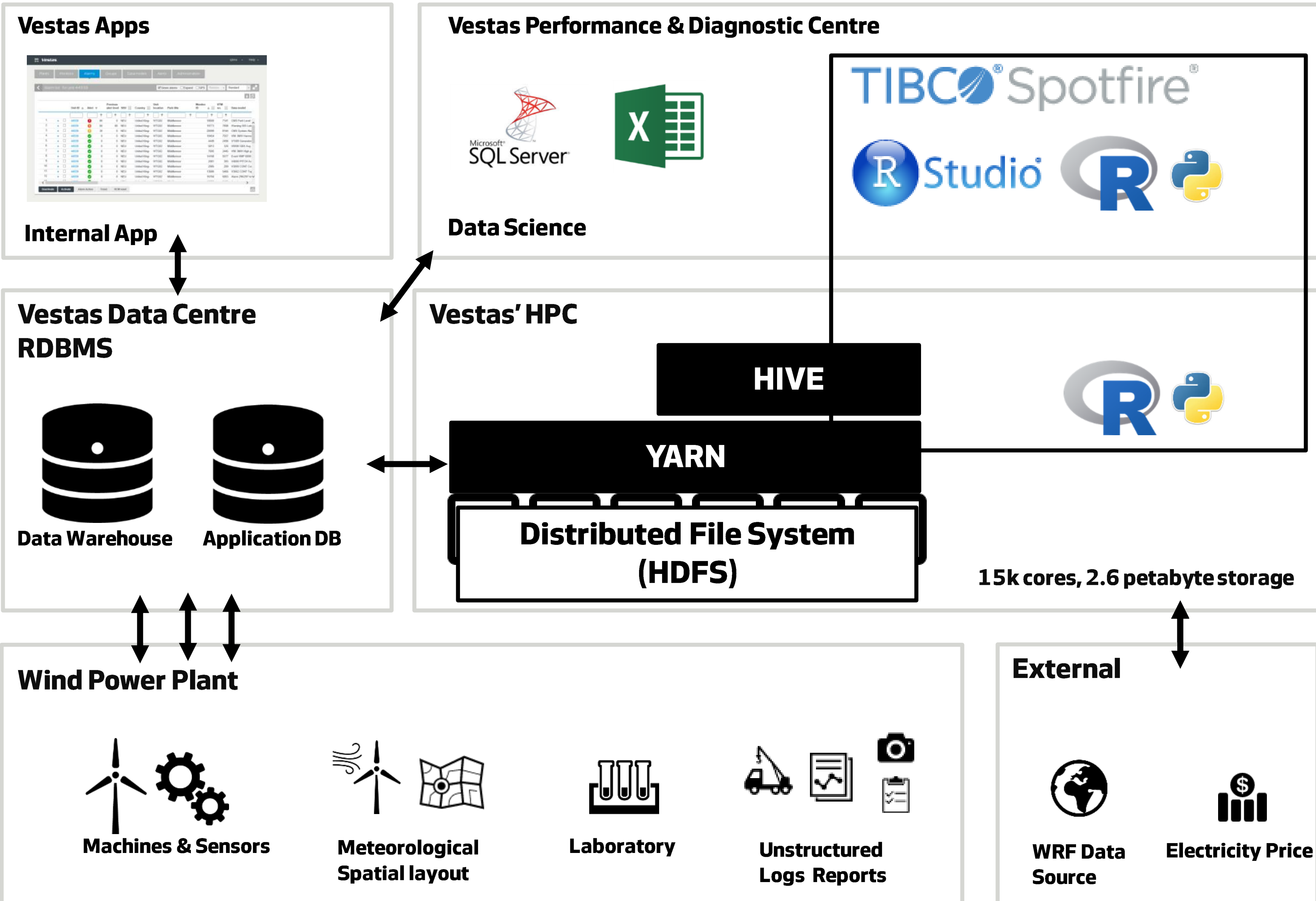


MS SQL Server: 150tb SAN

+ It works
+ Business penetration
+ Transactional & relational

- Costly
- Staging areas, re computation
- Where is R

Modern Data Platform



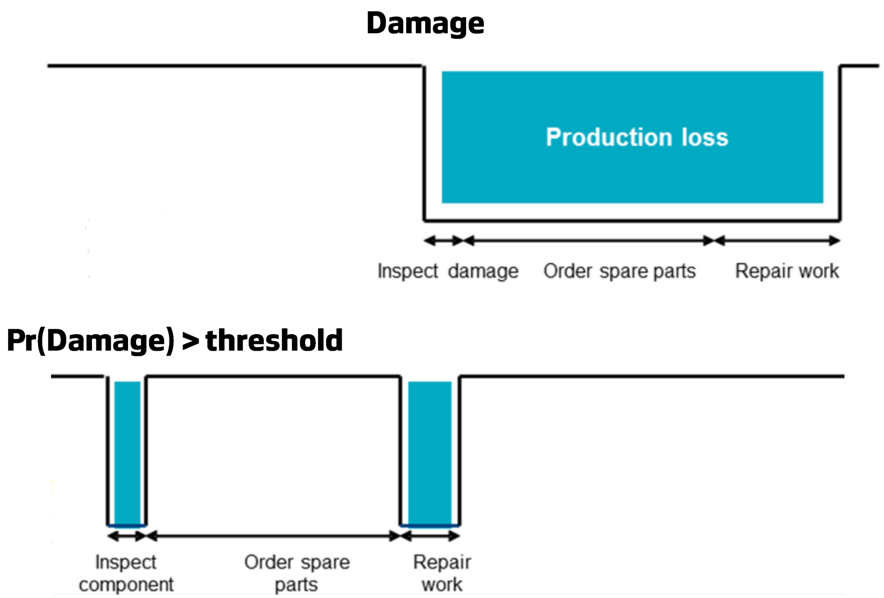
Value Driver

Time is Money – Downtime is Expensive

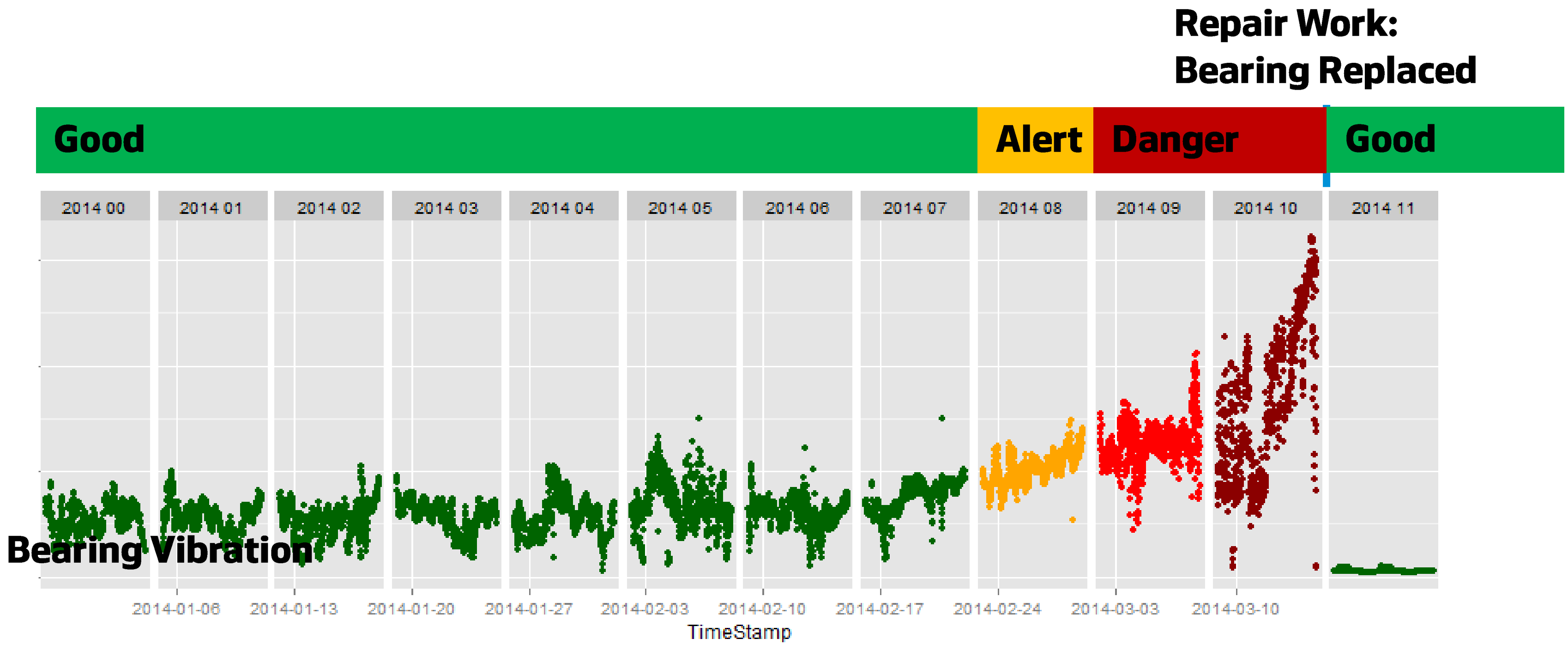
Downtime Lost Wind Energy Production

Corrective

Predicitve



Vestas Turbine Monitor



Predict generator bearing failure

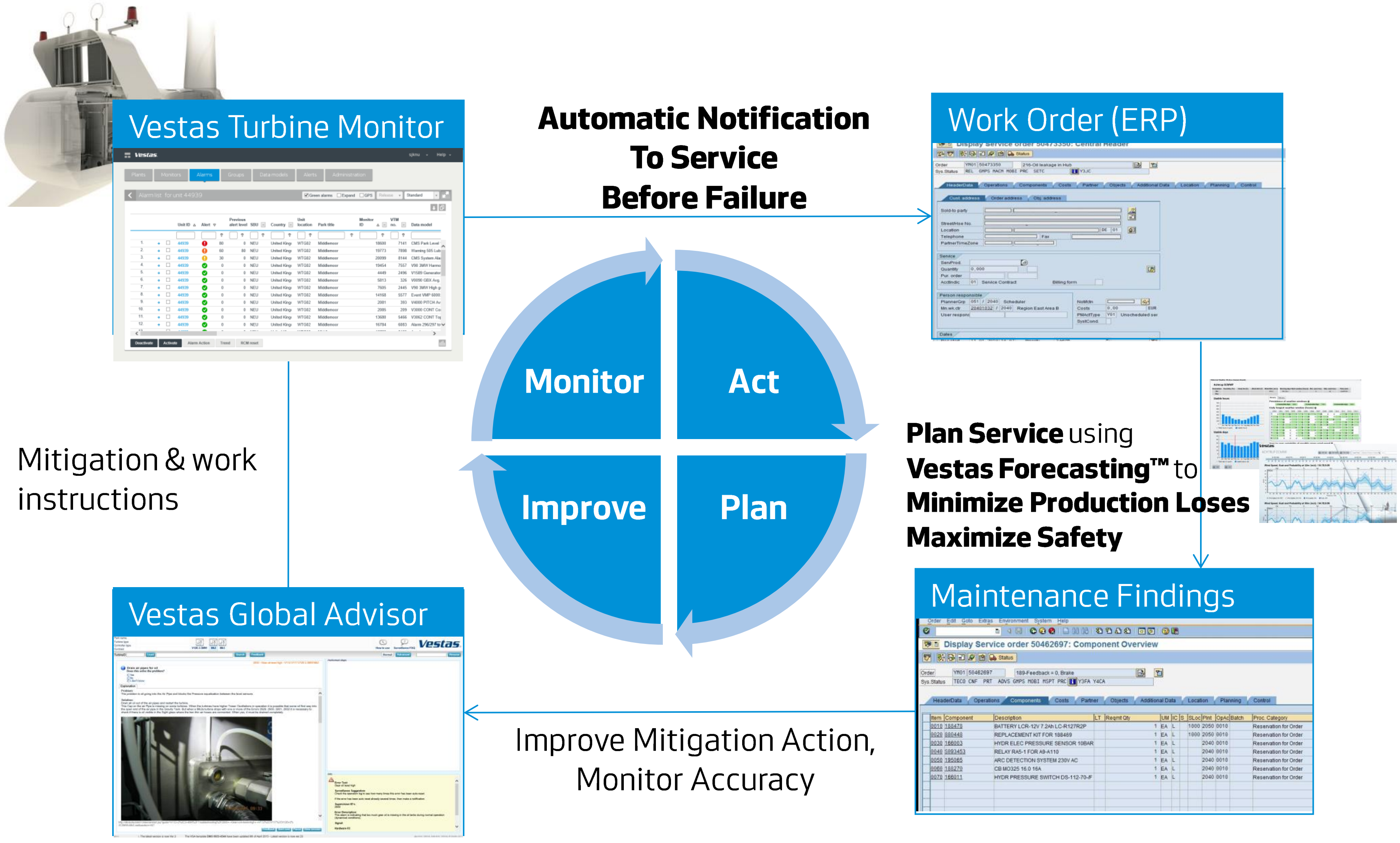
- Training data :
  - Vibration 10min values, several parks
  - filter down to turbine in full operation

- Changepoint algorithm on raw or residuals of a machine learned model
- Tune using confusion matrix : best precession, accuracy, sensitivity
  - False Alert vs Wasted Service Visit

package: changepoint , caret

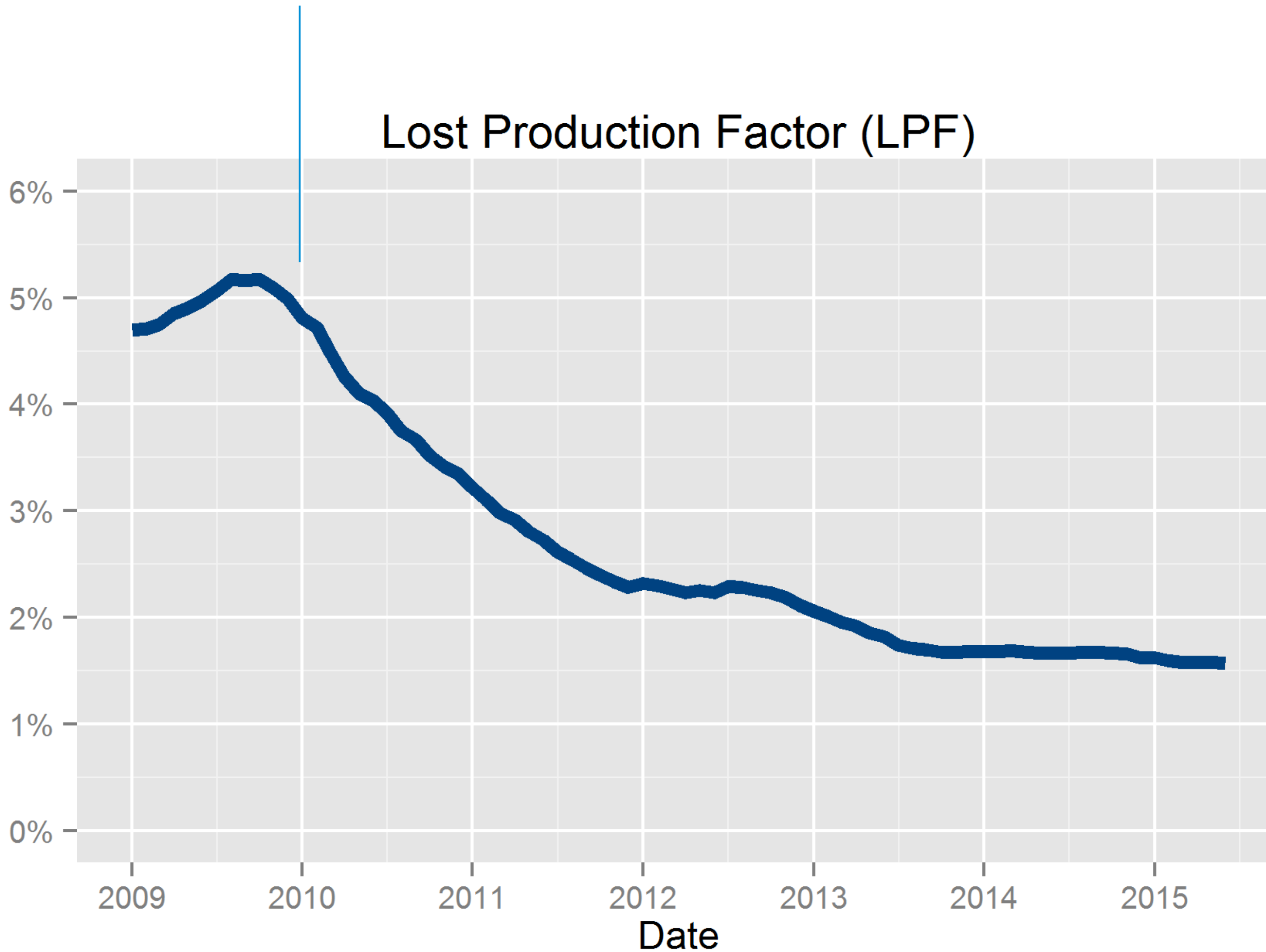
Vestas Turbine Monitor

Full Business Support for Process & Information Flow



Wind. It means the world to us.™

Measurable Business



Wind. It means the world to us.™

HIVE – Hadoop **I**nteracti**VE**

Hadoop allows distributed processing of **large data sets across clusters**

Called Google's data model (Map Reduce)

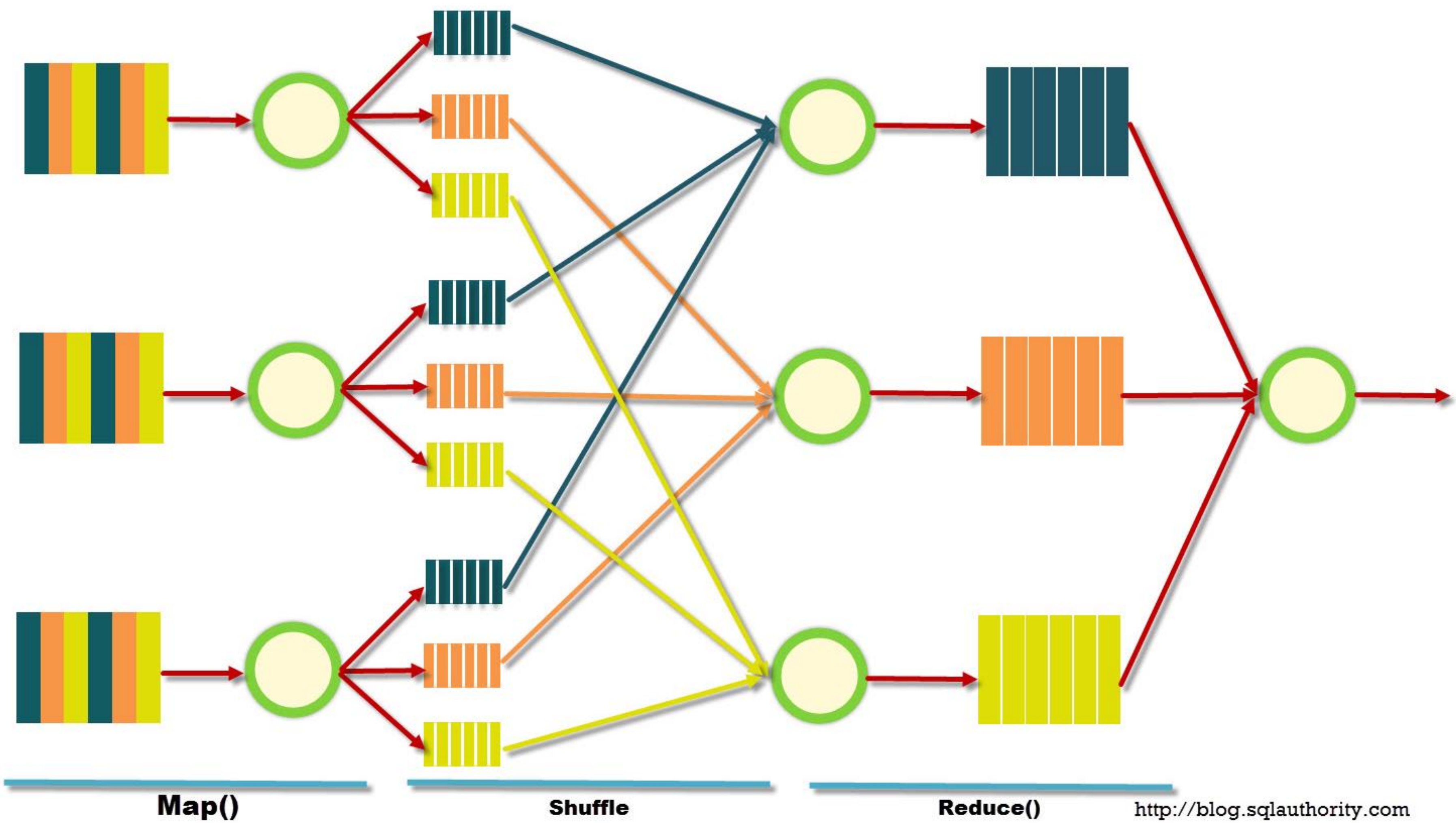
HIVE provides SQL like queries over **Hadoop**

```
Select
        turbineNumber,
        avg(WindSpeed)
from Turbine10minData
group by turbineNumber
```

Faster than indexed MS SQL Server

Much faster if grouped by non indexed column, like temperature bin

HADOOP Map Reduce (MR)

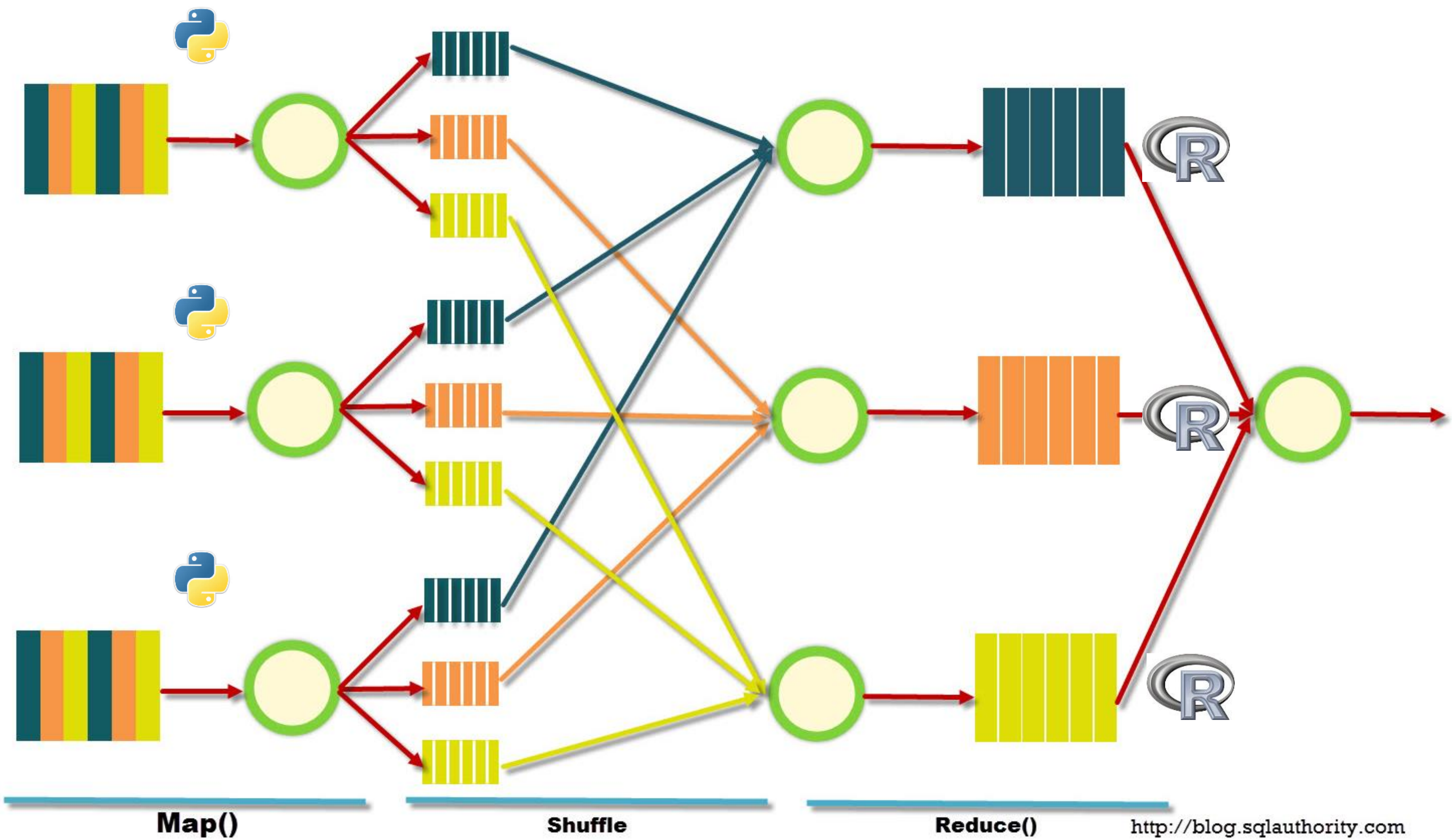


Distributed Data

Are collected and sorted

Distributed Compute

HIVE Custom Map Reduce - inject R , Py, Octave . . .



Distributed Data

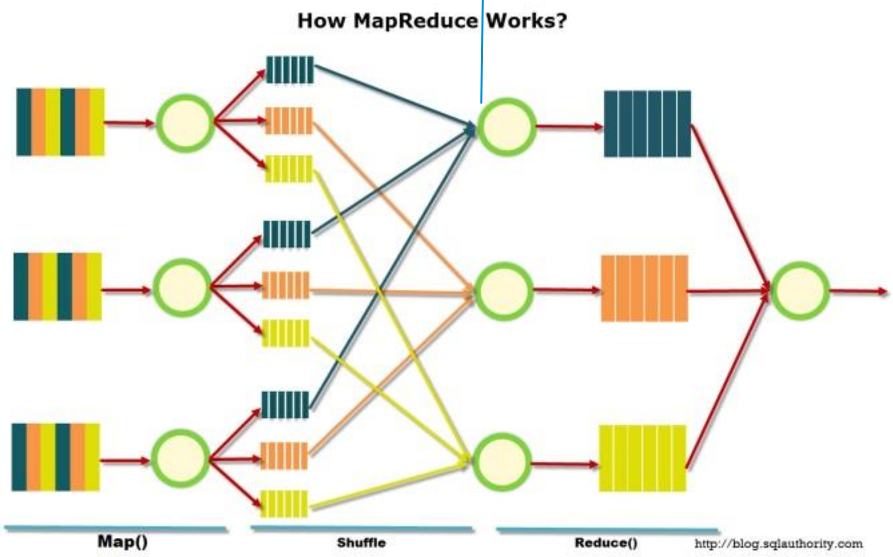
Are collected and sorted

Distributed Compute

Inject script s using HIVE Transform

```
FROM (
  SELECT * FROM TurbineData
  DISTRIBUTE BY turbineId
) as map
SELECT
    TRANSFORM ( * ) USING 'wrapper.sh someAlgorithm.R'
  as (turbineId,ttimeStamp, propbilityOfFailure);
```

Map step: DISTRIBUTE data to compute nodes BY tubine



http://hortonworks.com/blog/using-r-and-other-non-java-languages-in-mapreduce-and-hive/

HIVE: Inject script based computations

```
FROM (
  SELECT * FROM TurbineData
  DISTRIBUTE BY turbineId
) as map
SELECT
    TRANSFORM ( * ) USING 'wrapper.sh someAlgorithm.R'
  as (turbineId,ttimeStamp, propbilityOfFailure);
```

wrappe.sh loads R and #!Rscript
Data is piped to R using stdin
Data piped back using stdout

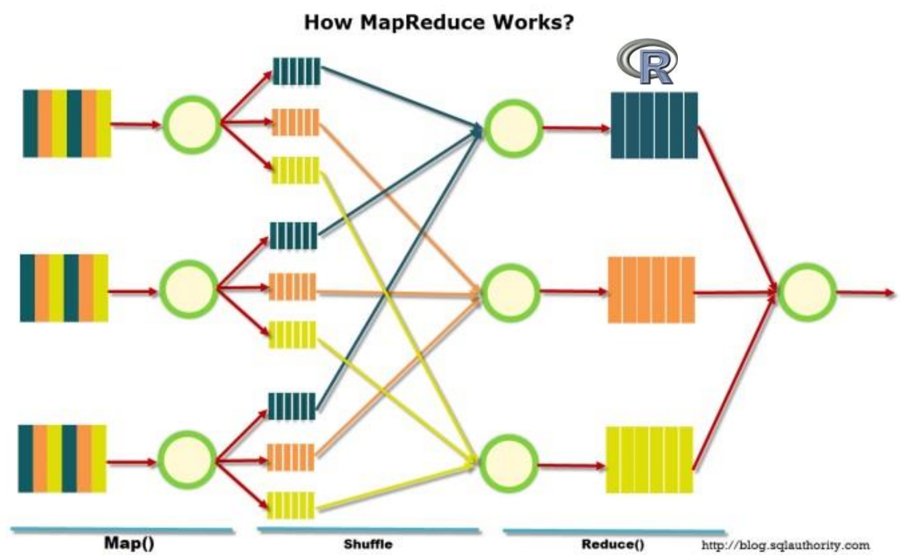
Hive gives total control to R – please mind

- Compute speed
- Memory consumption
- Error handling

Useful HIVE resource settings:

```
set mapred.reduce.tasks=200;
set mapreduce.reduce.memory.mb=10240;
```

Executes on mapped data



someAlgorithm.R

```r
#!/usr/bin/env Rscript

.libPaths('/gpfs01/R/packages/bmk/v1.0.0')
suppressMessages( library(data.table) )
suppressMessages( library(fasttime) )
suppressMessages( library(vbmk) )

outDt <- tryCatch( {

  #fast read of piped stdin
  inDt <- fread('file:///dev/stdin', sep ="\t", showProgress=FALSE)

  #fast convert to POSIXct
  inDt <- inDt[,timestamp:=fastPOSIXct(ttimestamp, tz = "UTC")]

  #do your thing
  bmkProcess(inDt)

},
error =  bmkErrorHandler )

if (!is.null(odata)) {
  write.table(odata, quote=FALSE, sep="\t",
              col.names=FALSE, na =" ", row.names=FALSE)

}
```

Summary

HIVE for large batch / interactive jobs

- Proven more reliable with version 0.14
- Spark is lurking for interactive analysis

Familiar to users and cost efficient – no brainer

- Need to understand how to MapReduce the problem

Need to manage your **R**esources

- Fast readers (fread) make a difference
- tweek HIVE resources settings to get job completion

ValueBrings more data to table

References

Vestas and Big Data

http://www.ibmbigdatahub.com/video/ibm-helps-vestas-turn-climate-big-data-capital

R:

http://www.r-project.org/
http://www.revolutionanalytics.com/what-r

Python:

https://www.python.org/

Apache HADOOP

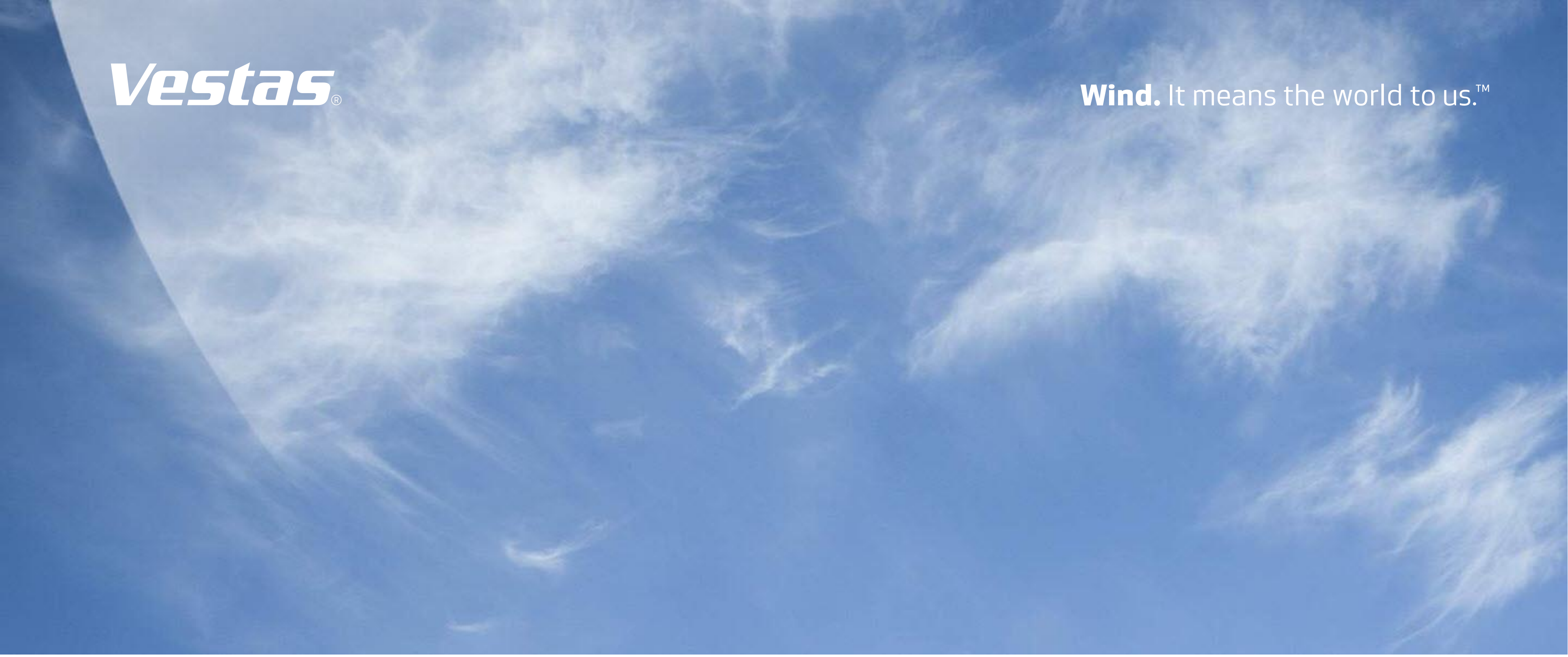
http://hadoop.apache.org/

Apache HIVE

https://hive.apache.org/

Apache Spark

https://spark.apache.org/

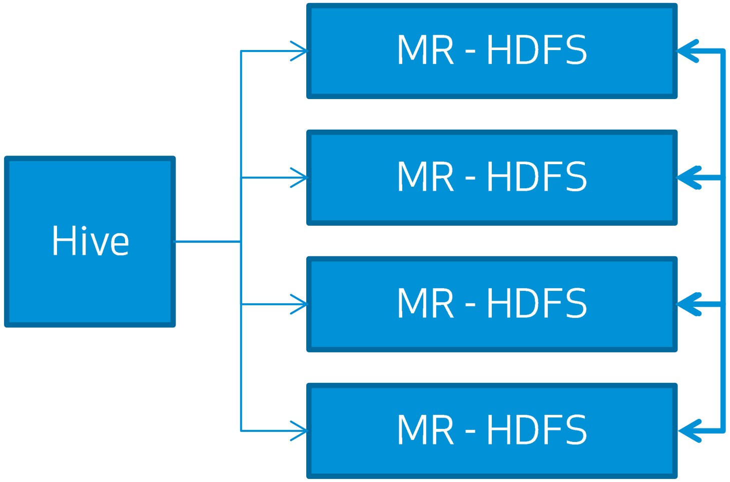


Thank you for your attention

Hive setup

- Distribution
  - Hortonworks
  - HDP 2.2
  - Hive 0.14
  - 20 nodes

- Traditional setup runs 1/10GbE
- Disks on each node, HDFS running on top



- Our setup runs on Vestas' HPC
- Has its own parallel filesystem (GPFS)
- Fast (40Gb) Infiniband interconnect
- Very high performance with few nodes
- Very high utilization per node