

**plain concepts**



# ABOUT US

**X** XAMARIN  
**PREMIUM**  
CONSULTING PARTNER

**13** MICROSOFT  
\*\*\*\*\*  
**MOST VALUABLE**  
PROFESSIONAL

 **PREMIER**  
DEVELOPER  
PARTNER  
**LIVE Apps**

**ALM**  
PARTNER OF THE YEAR  
FOR 7 CONSECUTIVE YEARS

**AGILE**  
ALLIANCE  
CORPORATE MEMEBER

MICROSOFT  
**GOLD CLOUD**  
PLATFORM

MICROSOFT  
**GOLD APP**  
DEVELOPMENT

MICROSOFT  
**GOLD** LIFECYCLE  
MANAGEMENT  
APPLICAITON

MICROSOFT  
**SILVER APP**  
INTEGRATION

MICROSOFT **SILVER**  
**COLLABORATION**  
AND CONTENT

**WINDOWS 8**  
APPLICATIONS  
PARTNER OF THE YEAR

MICROSOFT  
**BEST CLOUD**  
APPLICATIONS **2016**

# OUR SERVICES

UI/UX Design



Web & App  
development



Demos &  
Whitepapers



Marketing  
Campaigns



Custom CMS



# BIG DATA

## IMPLEMENTACION DE HADOOP EN AZURE

Alex Casquete

Software Developer at Plain Concepts

[acasquete@plainconcepts.com](mailto:acasquete@plainconcepts.com)

@acasquete

Francisco Martínez

Data Engineer at Plain Concepts

[fmartinez@plainconcepts.com](mailto:fmartinez@plainconcepts.com)

@pacommiranda

# LOGISTICA

- Horario
  - Martes 19: 9:30 – 14:00
  - Miércoles 20: 9:30 – 14:00
- Otros temas
  - WiFi:
  - Servicios, Máquina de Café, etc.

# LEARNING PATH

- Learning Path
  - Dos jornadas presenciales
  - Tres sesiones on-line



## Sesiones Presenciales

19 Abril – Implementación de Hadoop en Azure. Despliegue y administración

20 Abril – Implementación de Hadoop en Azure. Desarrollo

## Sesiones On-Line

28 Abril - Procesado de Streams sobre Hadoop y Azure Stream Analytics.

<https://goo.gl/mjBHRp>

5 Mayo - Machine Learning sobre Hadoop y Azure ML.

<https://goo.gl/IOQtIC>

12 Mayo - Visualización en Hadoop IaaS y Power BI.

<https://goo.gl/dOOUAi>



plain concepts

SOBRE VOSOTROS



**plain concepts**  
PREPARACION DEL  
ENTORNO



# BIG DATA

¿Qué es Big  
Data?

¿Qué NO es  
Big Data?

¿Para qué  
sirve Big  
Data?

# LAS TRES UVES

Volumen

Variabilidad

Velocidad

# VOLUMEN

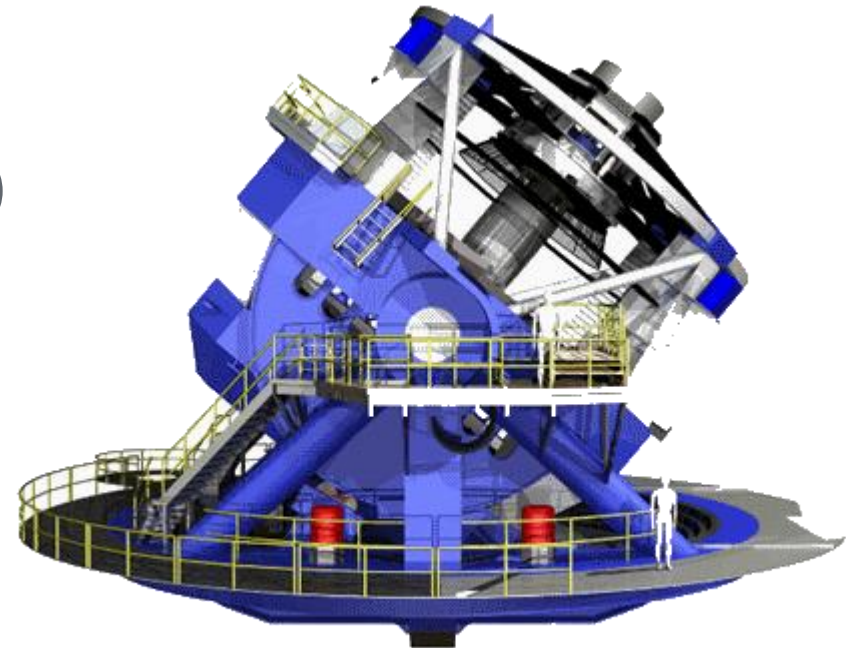


YAHOO!

Google

# VOLUMEN

- Large Synoptic Survey Telescope (LSST)
- Unos 40TB/día
- +100PB durante su expectativa de vida



# VOLUMEN

- Illumina HiSeq 2000
- 1Tb por día aproximadamente
- Un solo laboratorio puede tener de 25 a 100 de estos



# VOLUMEN





¿VOLUMEN?



# VELOCIDAD

## Ingeniería

- Mantenimiento Predictivo
- Gestión de Alarmas

## Detección de Fraudes

- Análisis de Actividad
- Análisis de Logs

## Publicidad Online

- Asignación de Anuncios
- Calculo de Rutas de Exposición

# VARIABILIDAD

```
{"delete":{"status":{"id":55628890374275072,"user_id":91674696,"id_str":"55628890374275072","user_id_str":"91674696"}}}
{"delete":{"status":{"id":240507246616915968,"user_id":382406164,"id_str":"240507246616915968","user_id_str":"382406164"}}}
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192957165568,"id_str":"337831192957165568","text":"RT @rokaya_mohareb: \u00e4\u00e4 \u00e1\u00e4\u00e5\u00e2
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192957161472,"id_str":"337831192957161472","text":"RT @bilio_muydunuz: 22:00-06:00 aras\u0031 i\u00e7ki yas
{"delete":{"status":{"id":671814677976832,"user_id":47682688,"id_str":"671814677976832","user_id_str":"47682688"}}}
{"delete":{"status":{"id":260724223939588096,"user_id":592170650,"id_str":"260724223939588096","user_id_str":"592170650"}}}
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192961380352,"id_str":"337831192961380352","text":"benget lah opi :)) @OvieS12: Sehun is Mine \u2665","sou
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192986533888,"id_str":"337831192986533888","text":"RT @cherryaam1: \u307f\u3063\u3061\u3083\u3093\u306e\u5f
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192978145281,"id_str":"337831192978145281","text":"RT @ImamShafiee: When Imam Ahmad heard that any of his r
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192965570560,"id_str":"337831192965570560","text":"\u0627\u0644\u0644\u06be\u064f\u0645 \u062c\u0645\u0644
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192969768960,"id_str":"337831192969768960","text":"@_okamickey \u3048\u3001\u306a\u3093\u3067\u00ff1f","sourc
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192978141184,"id_str":"337831192978141184","text":"\u306b\u3057\u3053\u304f\u3093\u3084\u3070\u308a\u3043\u
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192982327296,"id_str":"337831192982327296","text":"To be able to play with the birds and to see the stars u
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192978137088,"id_str":"337831192978137088","text":"http://t.co/LSMBPD0YD6","source":"\u003ca href=\"http
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192952971264,"id_str":"337831192952971264","text":"RT @Hind34: \u0623\u0646\u0627 \u0644\u0627 \u0623\u062e\u062e
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192973946881,"id_str":"337831192973946881","text":"RT @yakko_talk: 4\u0642\u306b\u9a12\u3052\u00ff01\u3044\u3
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192961351681,"id_str":"337831192961351681","text":"\u5c11\u3057\u722a\u306e\u4f38\u3073\u305f\u0624b\u3067\u
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192952967168,"id_str":"337831192952967168","text":"\u201c \u0627\u0644\u0644\u0647 \u0645\u0627 \u064a\u0627
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192973946882,"id_str":"337831192973946882","text":"@CmasPal yeah just do that and what time will you be at
{"created_at":"Fri May 24 07:23:27 +0000 2013","id":337831192986525696,"id_str":"337831192986525696","text":"RT @sahmk_: @busine_\n\u0627\u0633\u0647\u0645 \u064a\u0627\u0627
```

# VARIABILIDAD

- Schema-on-write
  - Trabajamos con un schema estático
  - Transformamos los datos a este schema (utilizando ETL)
  - Antes de admitir nuevos datos, debemos modificar el esquema
- Schema-on-read
  - Copiamos los datos con su schema original
  - Creamos un schema
  - Consultamos los datos usando el formato original (ETL on the fly)
  - Los datos siempre están ahí, solo necesitamos crear el schema que los describe

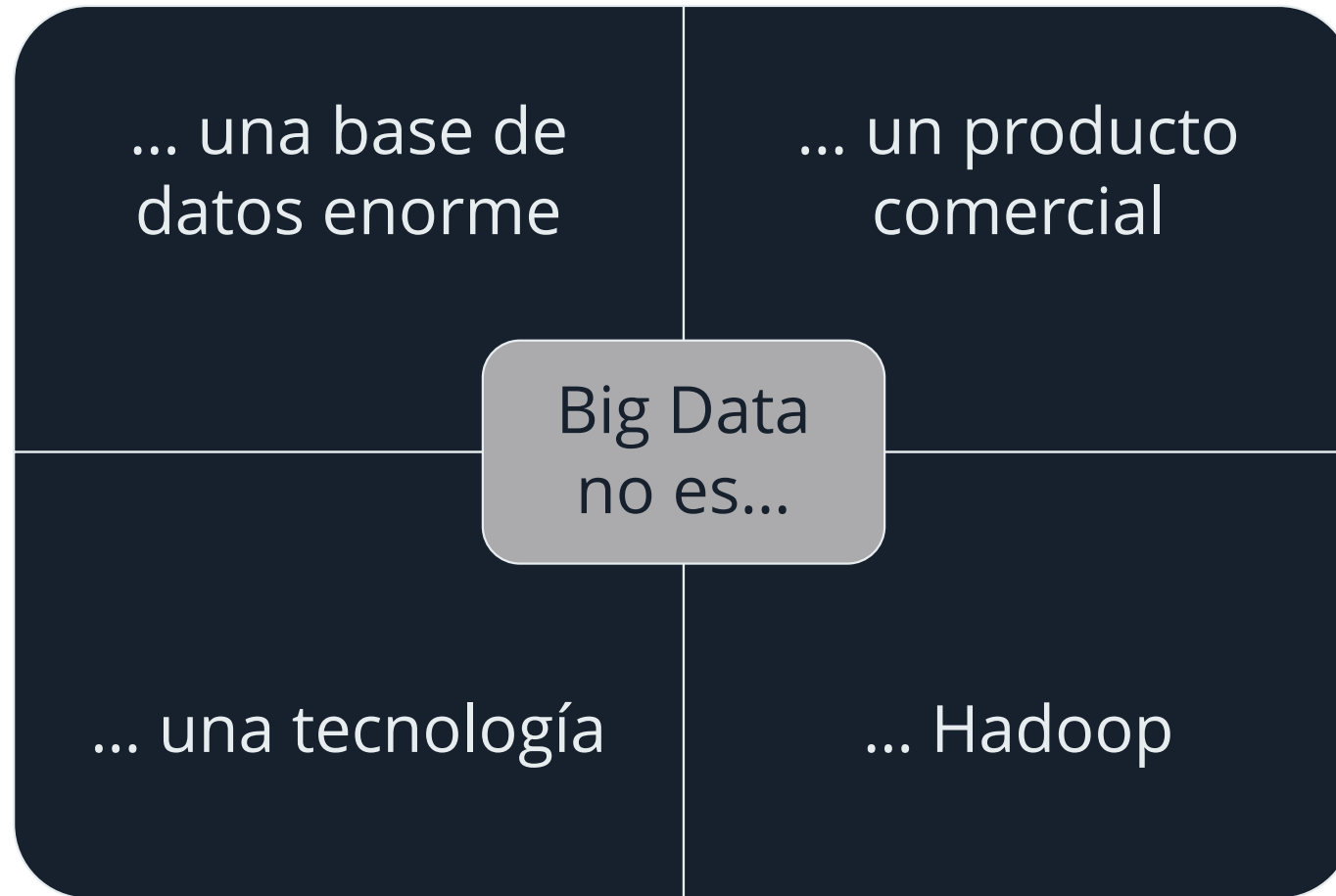
# SISTEMA TRADICIONAL



Y LA UTILIDAD?



# BIG DATA NO ES...

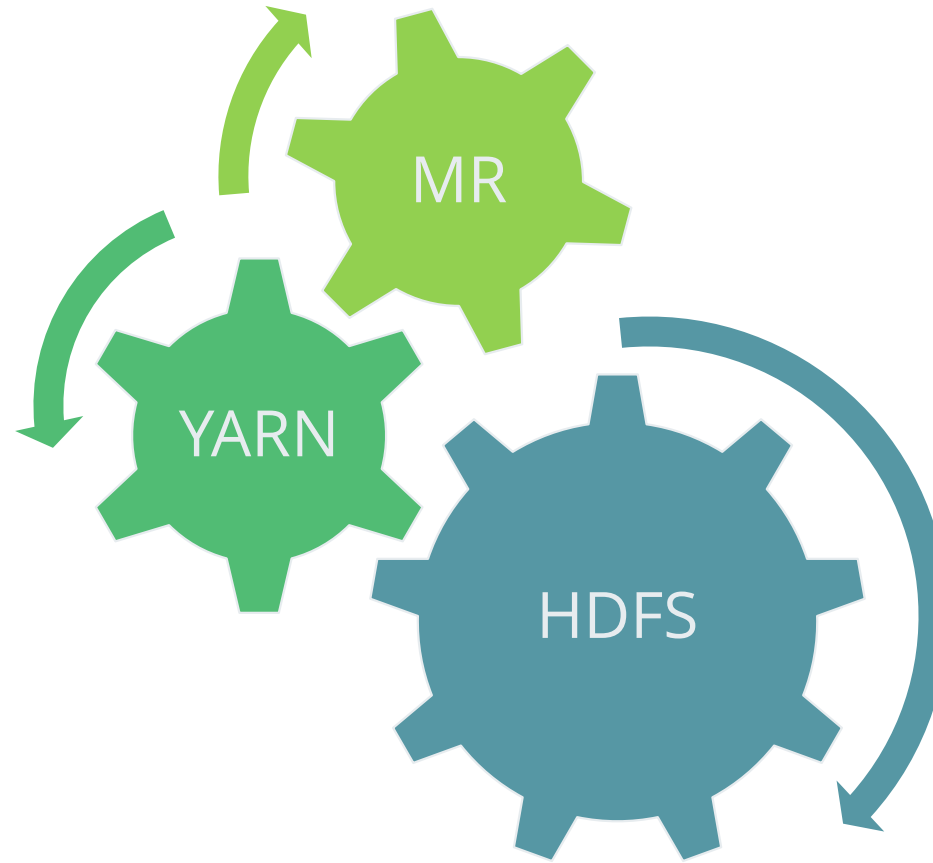


# RESUMEN

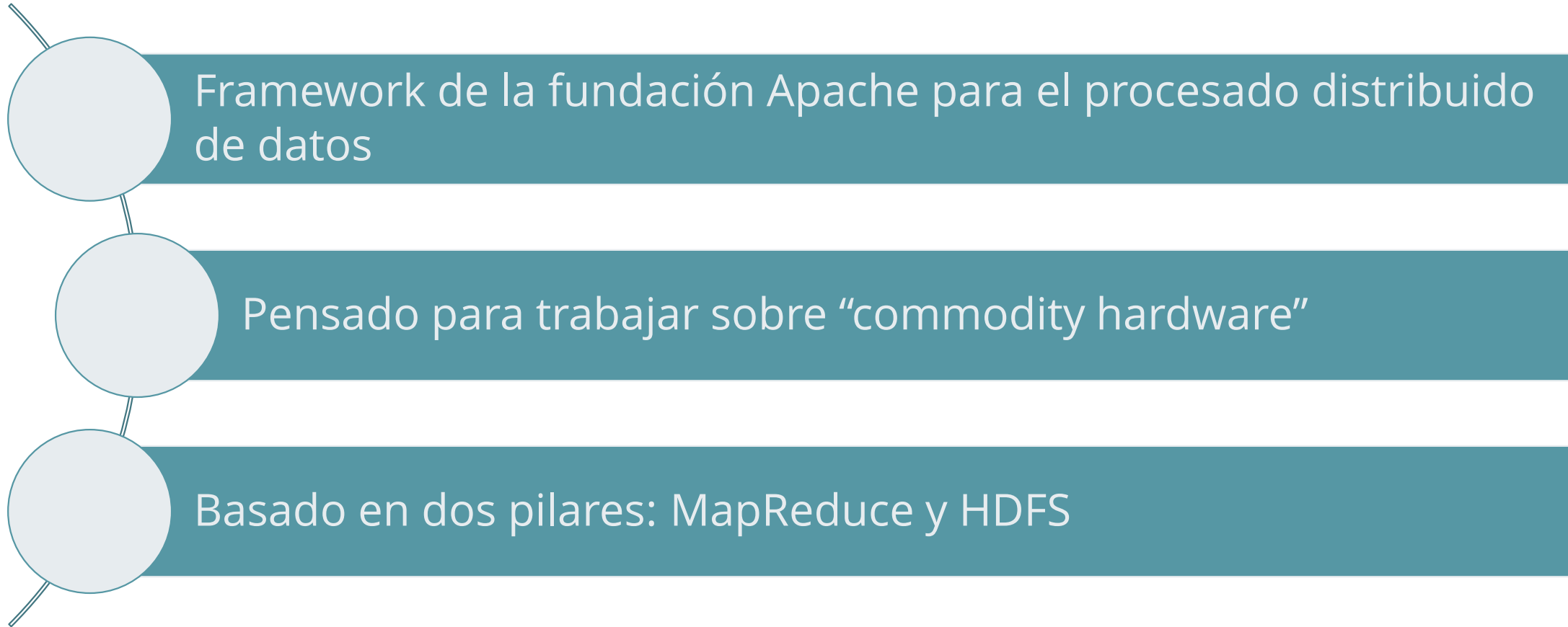
Big Data es el cambio de paradigma que representa la búsqueda de soluciones para almacenar y procesar datos NO estructurados Y datos estructurados conjuntamente de un modo económico y escalable



# HADOOP CORE



# ¿QUE ES APACHE HADOOP?



# MI INFRAESTRUCTURA PUEDE SER ASI...



O ASI...



# APACHE HADOOP

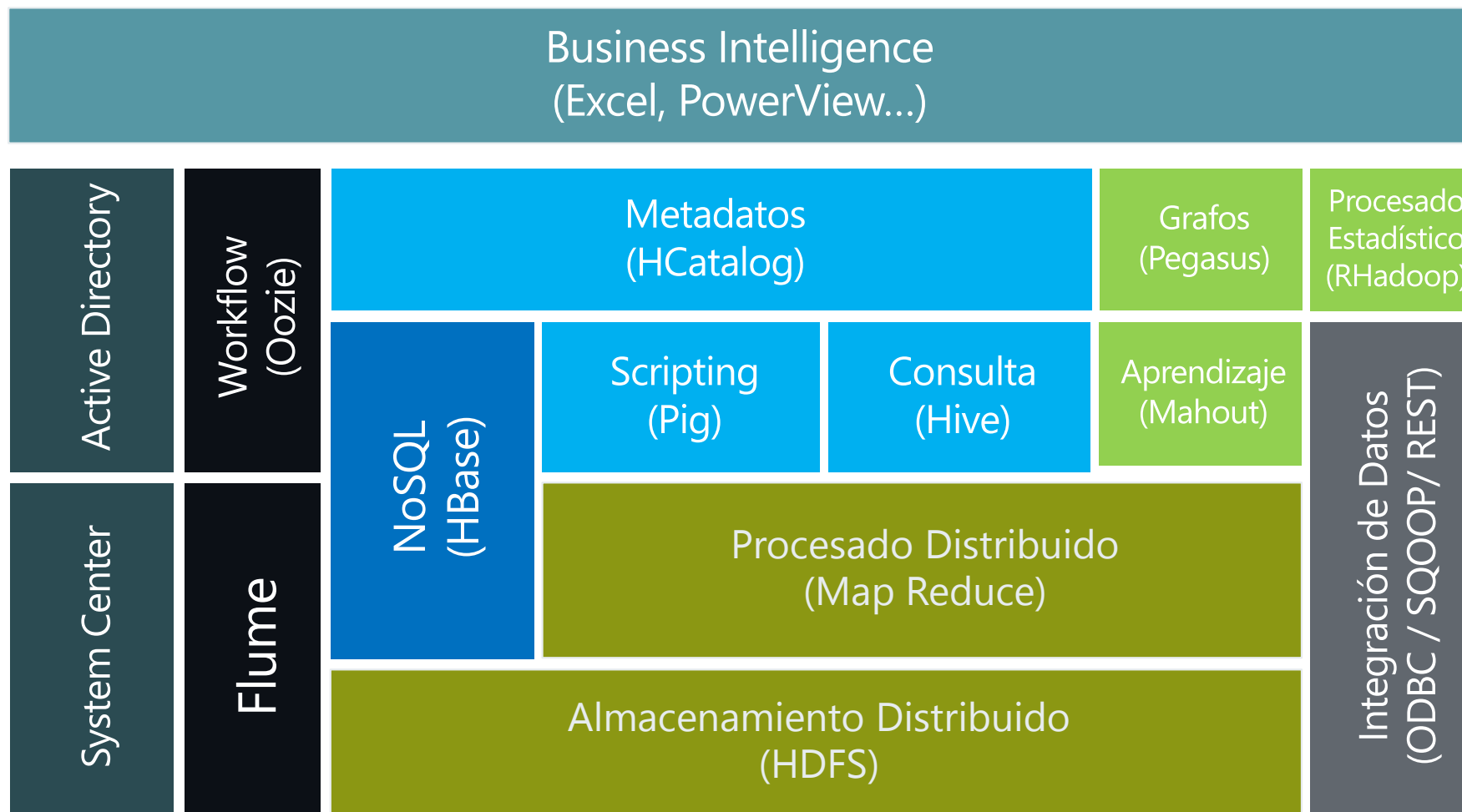
## MapReduce

Divide las tareas entre procesadores “cercanos” a los datos  
Compone los resultados

## HDFS

Almacenamiento distribuido  
Auto-reparable  
Redundante  
Nodo maestro (NameNode)

# ECOSISTEMA DE HADOOP V1





# HDFS

- Sistema de Ficheros distribuido
- Construido sobre Hardware no específico
- Alta Resistencia a fallos
  - Replicación de ficheros
  - Deteccion y Recuperación automática
- Optimizado para procesos por lotes (batch)
  - Lista de ubicaciones expuesta para minimizar trafico
  - Proporciona un ancho de banda agregado muy elevado

# HDFS CLUSTER

## Name Node

- Gestiona los Data Nodes
- Guarda metadatos para todos los ficheros y bloques

## Data Nodes

- Almacenan los bloques de datos
- Se distribuyen por la topología de racks

## Cientes

- Hablan directamente con el Name Node, y después con los Data Nodes necesarios



# HDFS - ORGANIZACION

- Sistema de Ficheros Lógico
  - Soporta creación, borrado, renombrado, etc...
- Gestionado por el Name Node
- Metadatos
  - Organización en ficheros y directorios
  - POSIX compatible (permisos, estructura...)

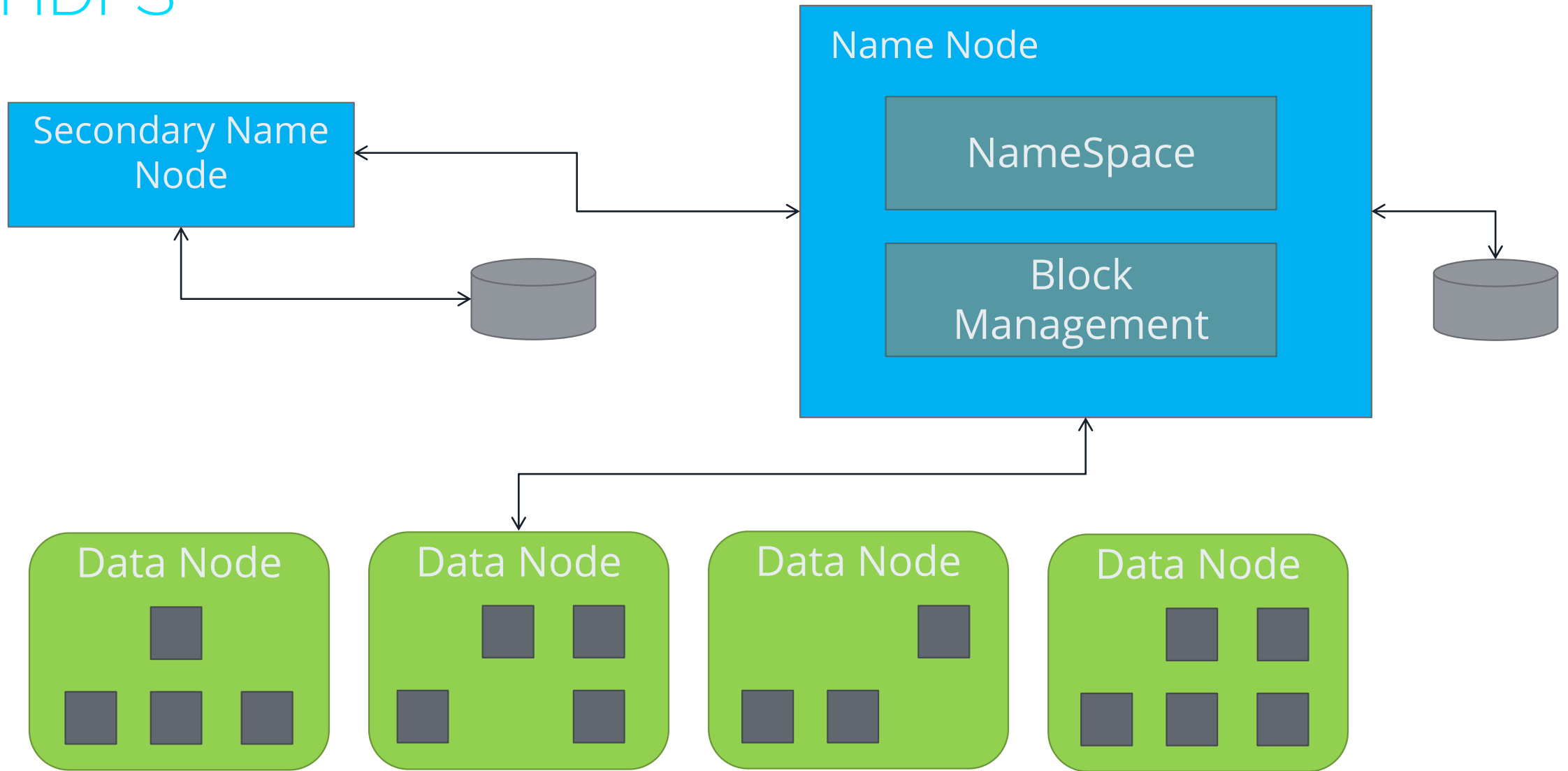
# HDFS – NAME NODE

- Mantiene una imagen del sistema de ficheros en memoria
  - 4GB de memoria son suficientes
- Utiliza un log de transacciones (EditLog) para almacenar cambios en el Sistema de ficheros (nuevos ficheros, cambios en el numero de replicas, etc...)
  - Se almacena en el Sistema de ficheros local del Name Node
- El Sistema de ficheros complete, incluyendo el mapeo de bloques y demás metadatos, se almacena en un fichero FsImage
  - También se almacena en el Sistema de ficheros local del Name Node
- Utiliza un Sistema de checkpoints para poder recuperar el sistema en caso de fallo
  - En cada arranque, recupera FsImage, lo actualiza con la información de EditLog y almacena una copia de FsImage como checkpoint

# HDFS – DATA NODE

- Un Data Node almacena bloques de ficheros en su Sistema de ficheros local
- No conoce la existencia de HDFS
- Almacena cada bloque en un fichero diferente
- No crea todos los ficheros en el mismo directorio
  - Utiliza un algoritmo para calcular en numero optimo de ficheros por directorio, creando directorios nuevos a medida que los necesita
- Cuando arranca, genera una lista de todos los bloques y se los envía al Name Node como BlockReport

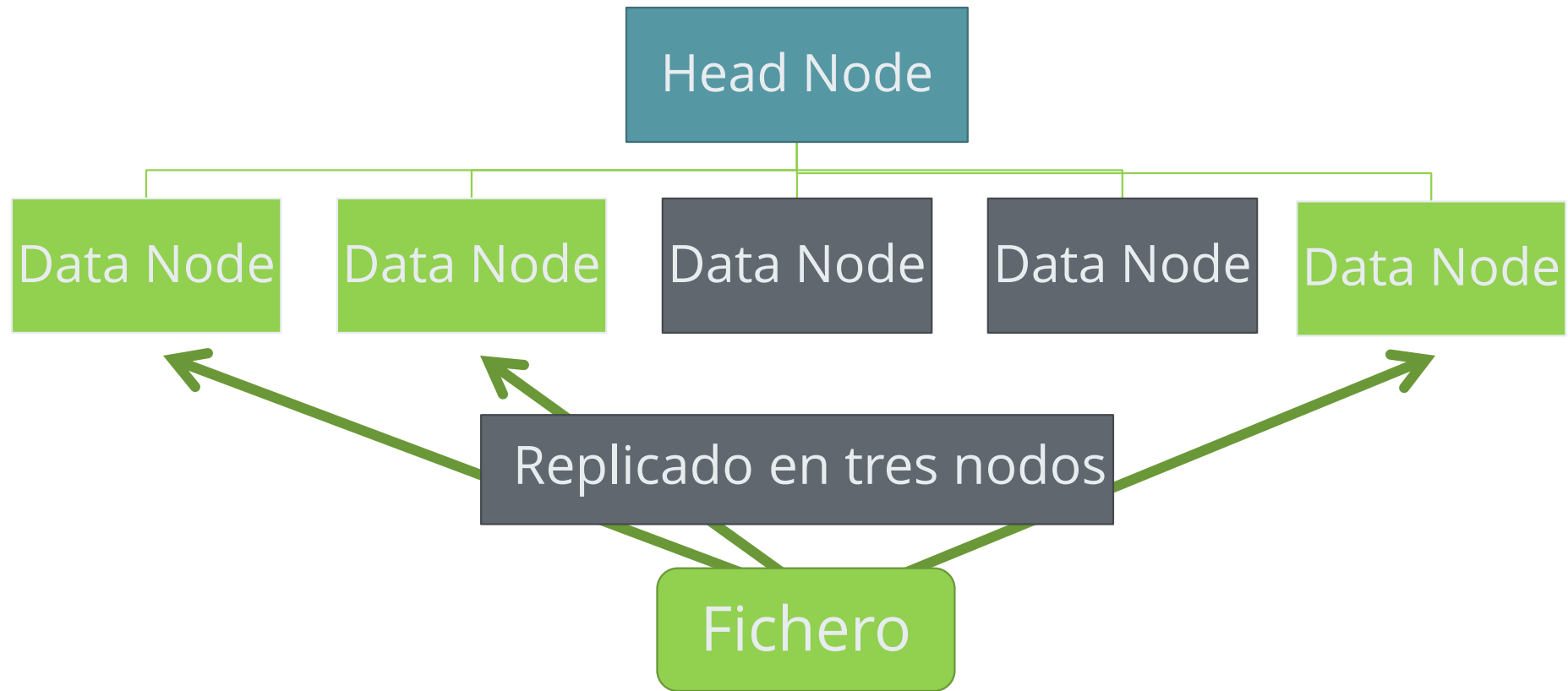
# HDFS



# HDFS – REPLICACION

- Diseñado para almacenar ficheros muy grandes en varias máquinas
  - Cada fichero se divide en bloques, del mismo tamaño salvo el último
  - Los bloques se replican automáticamente
  - El tamaño del bloque y el numero de réplicas es configurable por fichero
  - Por defecto, bloques de 64MB y 3 réplicas

# HDFS

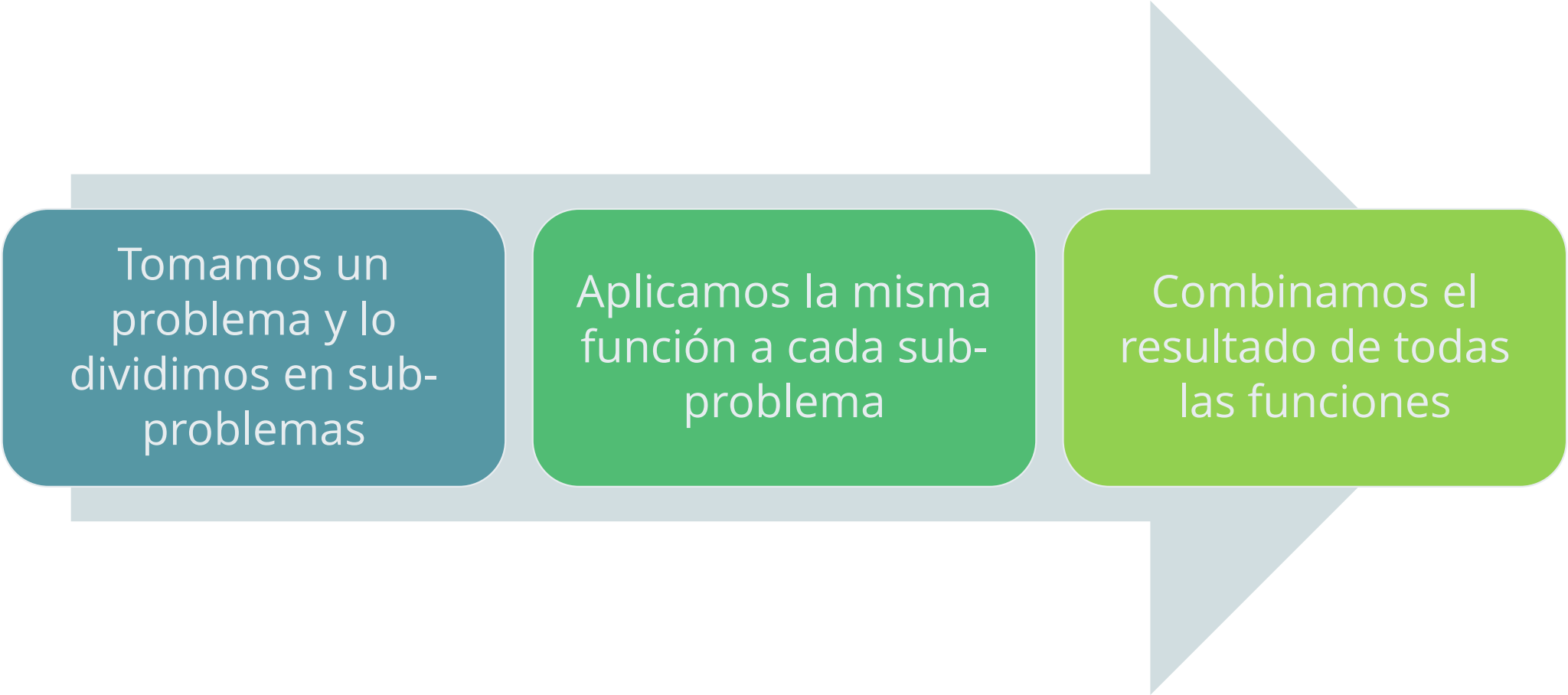


Optimizado para la lectura y tolerante a fallos

# MAP REDUCE - ¿QUE ES?

- Framework de computación distribuida para el análisis de datos
  - Grandes conjuntos de datos
- Computación de datos locales
  - Llevar la computación a los datos y no al revés
- Procesado en paralelo
- Función Map, procesa un par clave-valor para generar un valor intermedio
- Función Reduce, procesa los valores intermedios con la misma clave intermedia

# MAP REDUCE - ¿COMO FUNCIONA?



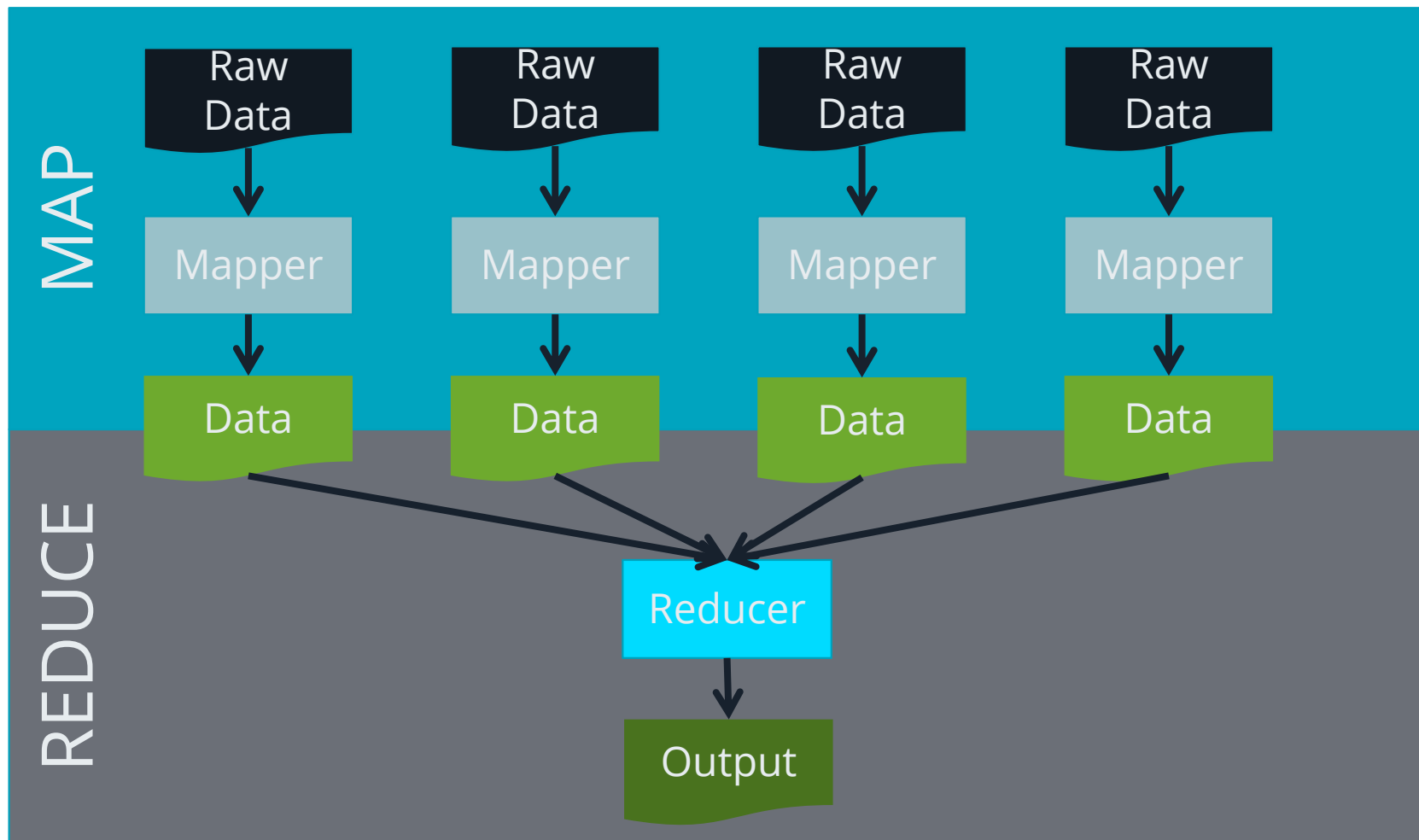
Tomamos un problema y lo dividimos en sub-problemas

Aplicamos la misma función a cada sub-problema

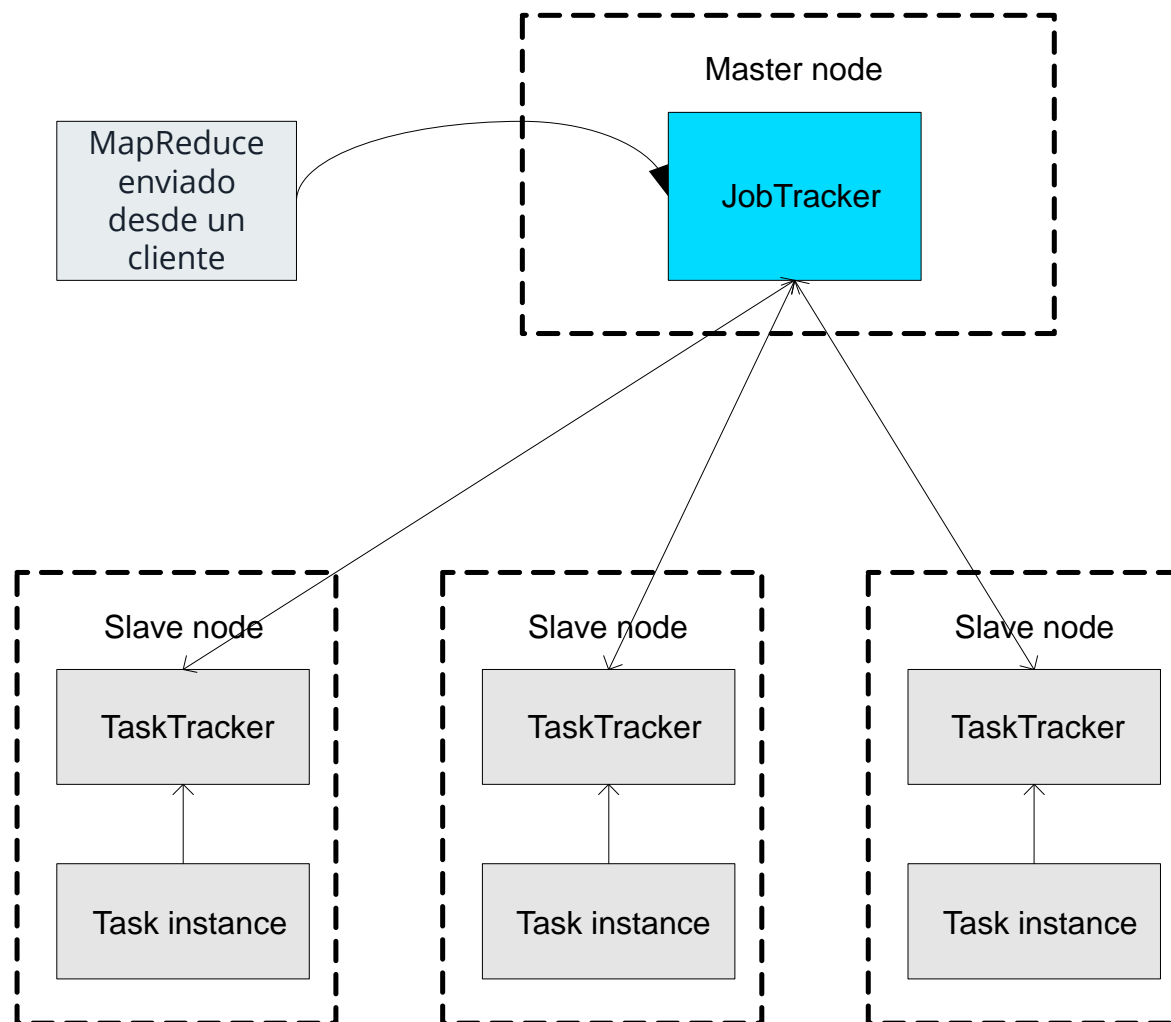
Combinamos el resultado de todas las funciones



# MAP REDUCE



# MAPREDUCE - ARQUITECTURA



# MAP REDUCE

## Funcionalmente

Map  $f(k1, v1) \rightarrow list(k2, v2)$

Reduce  $f(k2, list(v2)) \rightarrow (k2, v3)$

## Código

```
var map = function (key, value, context) {  
  var words = value.split(/^[a-zA-Z]/);  
  for (var i = 0; i < words.length; i++) {  
    if (words[i] !== "") {  
      context.write(words[i].toLowerCase(), 1);  
    }  
  }  
};  
  
var reduce = function (key, values, context) {  
  var sum = 0;  
  while (values.hasNext()) {  
    sum += parseInt(values.next());  
  }  
  context.write(key, sum);  
};
```

## En la práctica, WordCount

chalaneru chalaneru que lleves en la chalana

## Map

(chalaneru,1) (chalaneru,1), (que, 1), (lleves,1), (en,1), (la,1),(chalana,1)

## Shuffle

(chalaneru,(1,1)) (que,1), (lleves,1), (en,1), (la,1),(chalana,1)

## Reduce

(chalaneru,2) (que,1), (lleves,1), (en,1), (la,1), (chalana,1)

plain concepts

MAP REDUCE



# MAP REDUCE - PROBLEMAS

- Presencia de “single-point of failure”
  - Si el JobTracker falla, todo se viene abajo
- Pobre escalado
- Amplio consumo de memoria
- Rendimiento pobre
- Acoplamiento entre la gestión de recursos y el proceso de datos

# EVOLUCION A HADOOP 2.0

## **Single Use System**

*Batch Apps*

### **HADOOP 1.0**

#### **MapReduce**

(cluster resource management  
& data processing)

#### **HDFS**

(redundant, reliable storage)

## **Multi Use Data Platform**

*Batch, Interactive, Online, Streaming, ...*

### **HADOOP 2.0**

#### **MapReduce**

(batch)

#### **Tez**

(interactive)

#### **Others**

(varied)

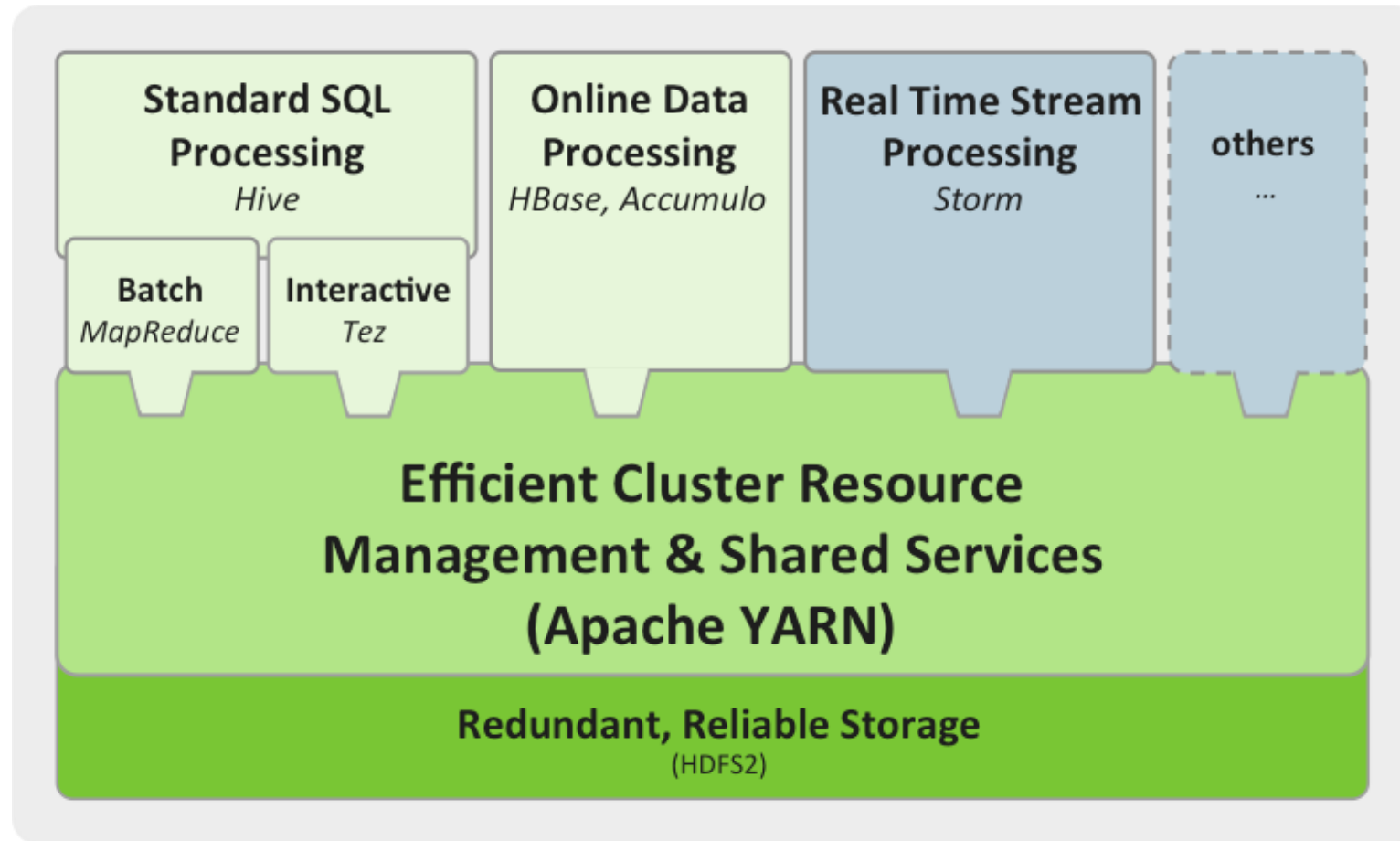
#### **YARN**

(operating system: cluster resource management)

#### **HDFS2**

(redundant, reliable storage)

# HADOOP 2.0

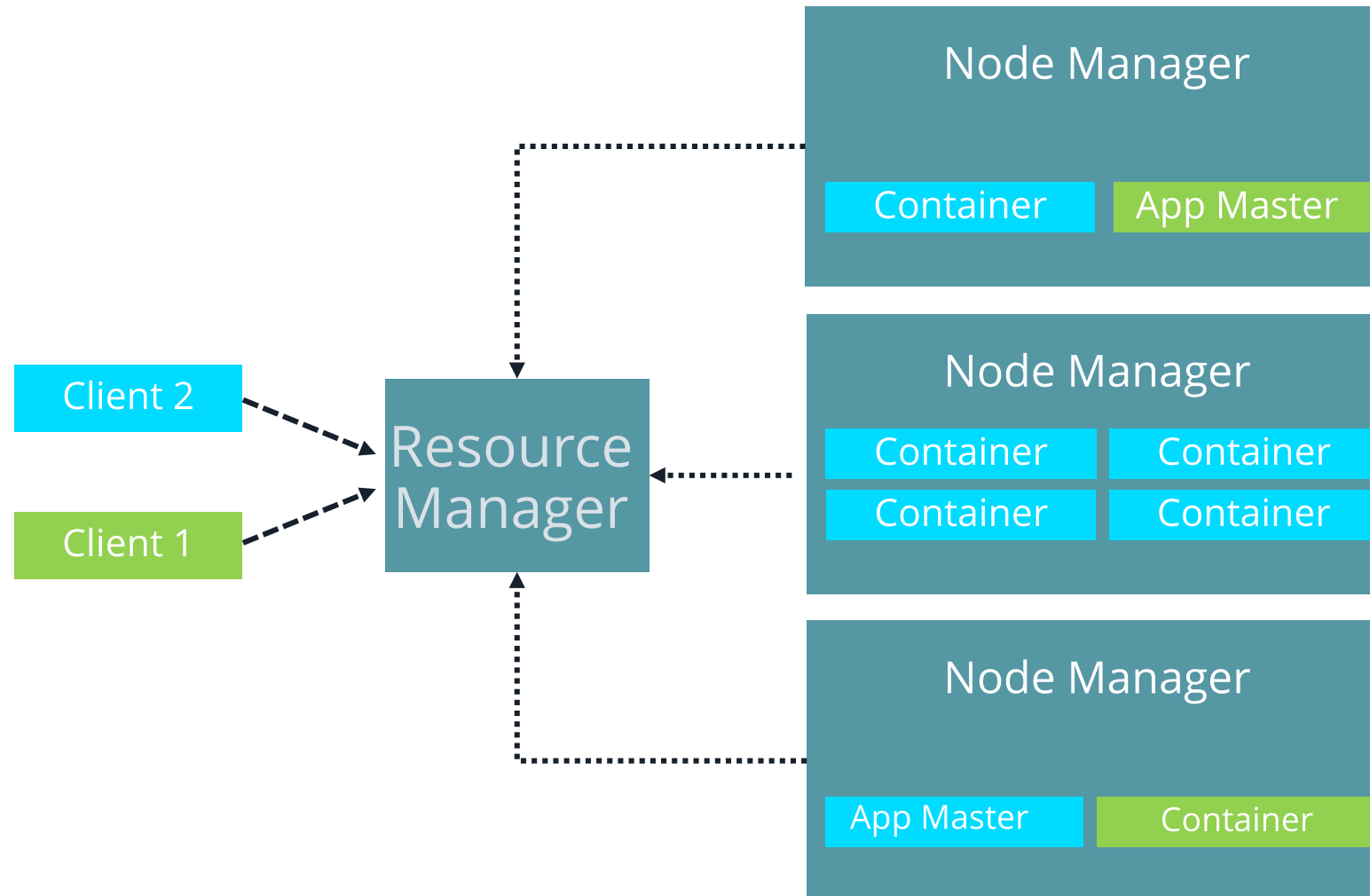


# YARN - ¿QUE ES?

- Yet Another Resource Negotiator
- YARN Application Resource Negotiator
- Framework de proposito general del que MapReduce es una aplicación más
  - Nos permite trabajar con muchas otras aplicaciones, como Tez
- Nace para separar las funcionalidades del JobTracker
  - El ResourceManager y sus NodeManager se encargan de gestionar los recursos de las aplicaciones de forma distribuida
  - El ApplicationMaster se encarga de gestionar una aplicación
- Reemplaza la parte de gestión de recursos de MapReduce



# YARN



plain concepts

YARN



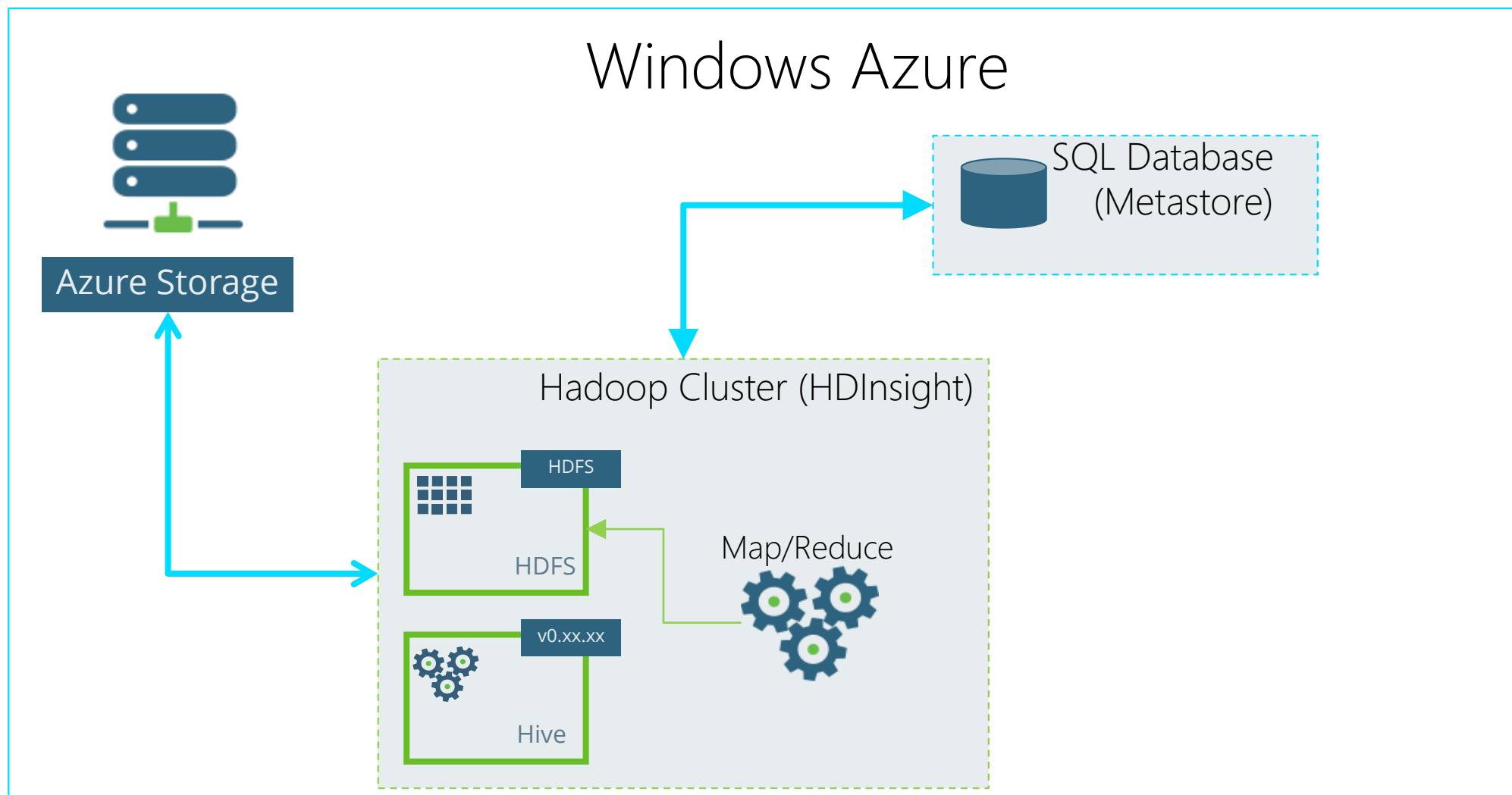
# DESPLIEGUE DE HDINSIGHT

- Azure SQL Database como Metastore
- Azure Storage como HDFS
- Azure Data Lake como HDFS
- Administración
- Clusters bajo demanda con PowerShell

# HD INSIGHT - ¿QUE ES?

- Distribución de Apache Hadoop en Azure
  - Basado en la distribución de HortonWorks
- Nos permite levantar clusters en minutos
- Utilizando Azure Blob Storage como almacenamiento

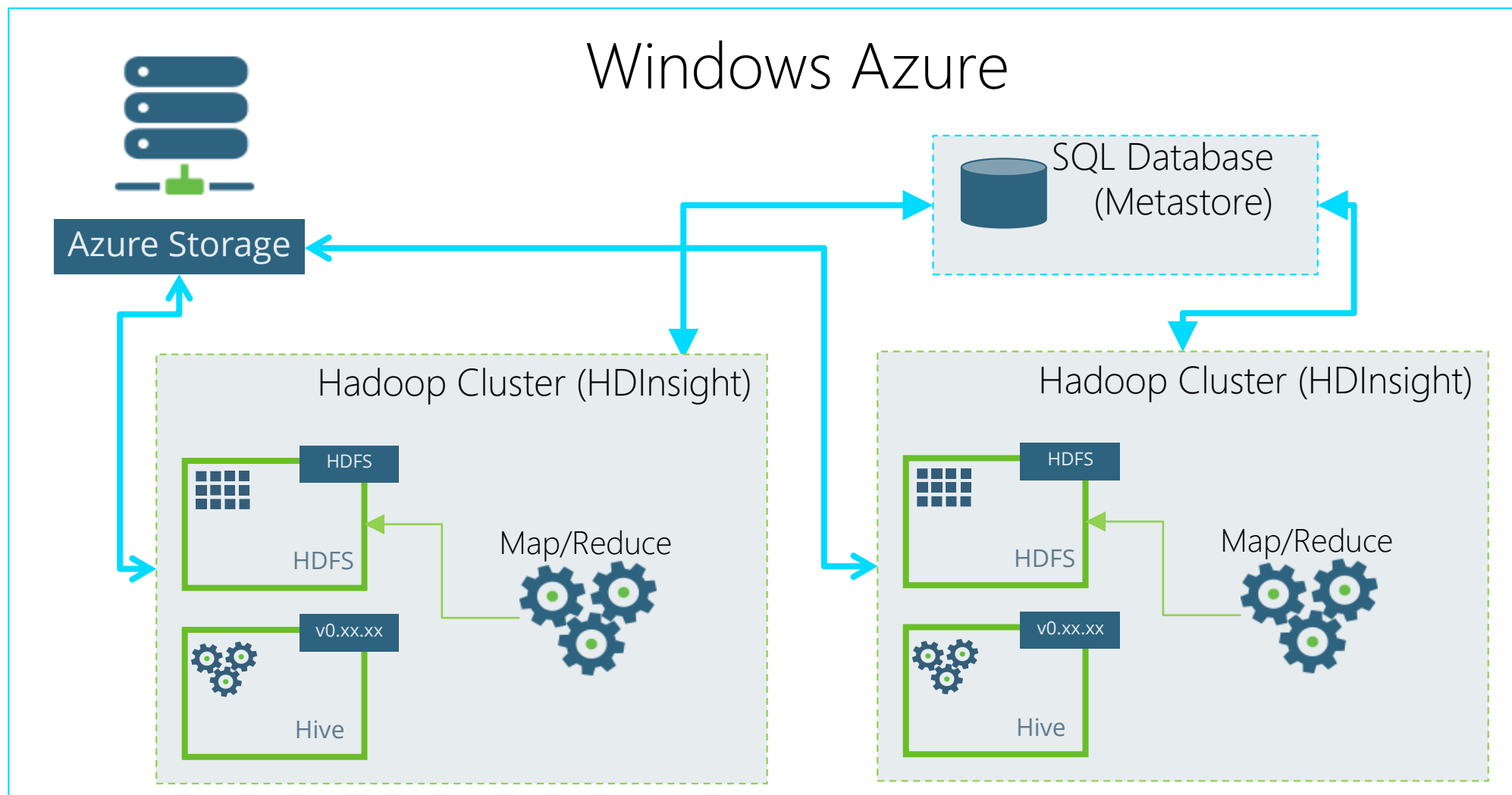
# HDINSIGHT - ARQUITECTURA



# AZURE SQL DATABASE COMO METASTORE

- HDInsight requiere una base de datos para almacenar metadatos
  - Veremos mas detalle de esto cuando veamos HIVE
- Por defecto se despliega con una base de datos no permanente
- Si queremos poder crear y destruir el cluster a nuestro antojo, pero mantener esos metadatos necesitamos una Azure DB como Metastore
- Configurable durante la creación del cluster
  - Desde PowerShell, SDK o Web UI

# MULTIPLES CLUSTERS



**plain concepts**  
CONFIGURANDO EL  
METASTORE





# AZURE STORAGE COMO HDFS

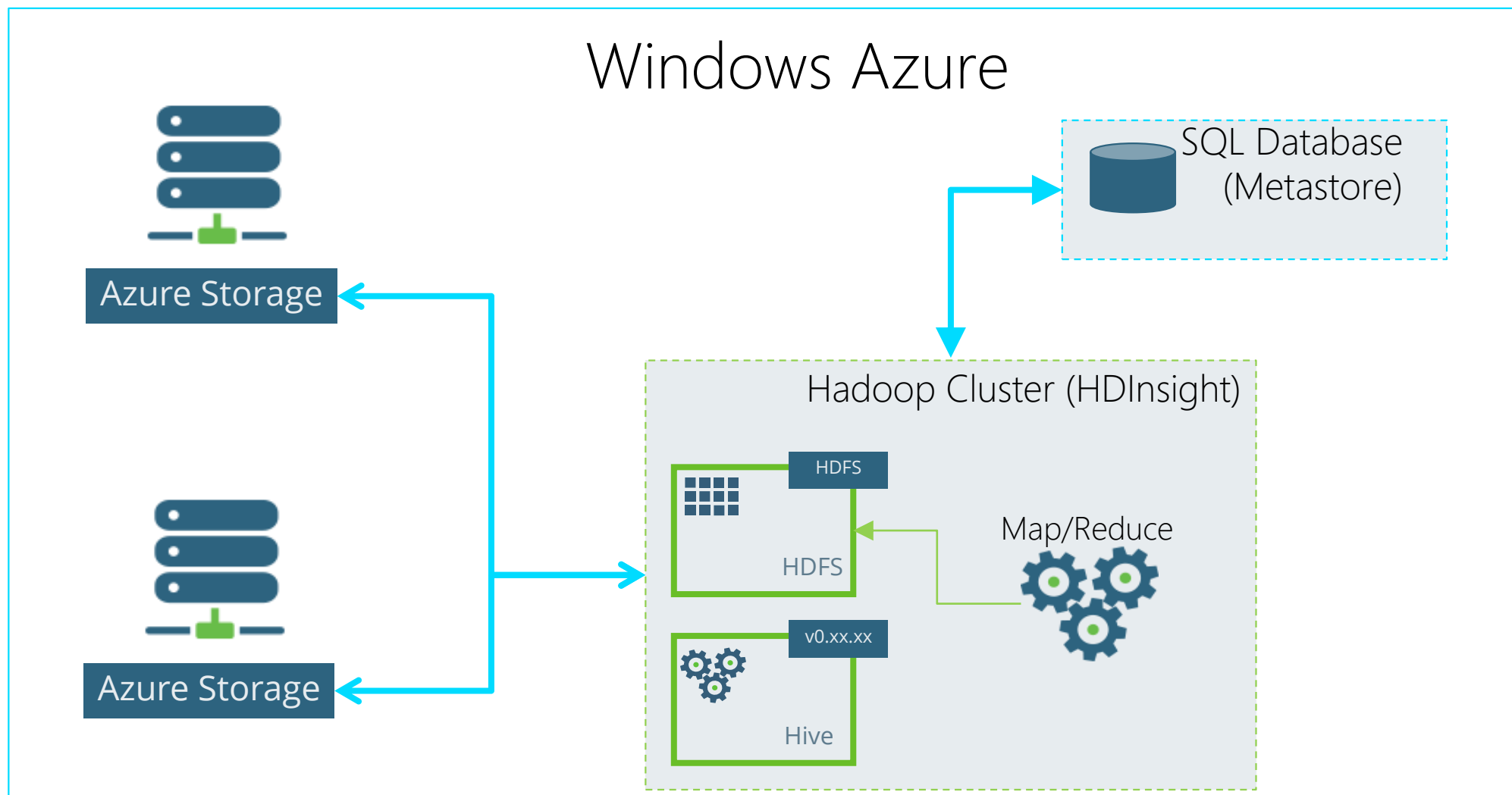
- HDInsight trabaja con Windows Azure Blob Storage
  - Azure Blob Storage proporciona almacenamiento persistente, escalable, geo-replicado y compatible
- Al desacoplar el almacenamiento de los datos del cluster que se encarga de procesarlos, habilitamos nuevos escenarios
- ¿Y el rendimiento?
  - Azure Flat Network Storage
  - Casi el mismo rendimiento que el HDFS local en lecturas
  - Mucho mejor rendimiento en escrituras
    - Replicación en nodos
  - En muchos casos, el cuello de botella es la velocidad de proceso y no la de transferencia de datos

**plain concepts**

HDINSIGHT CON BLOB  
STORAGE



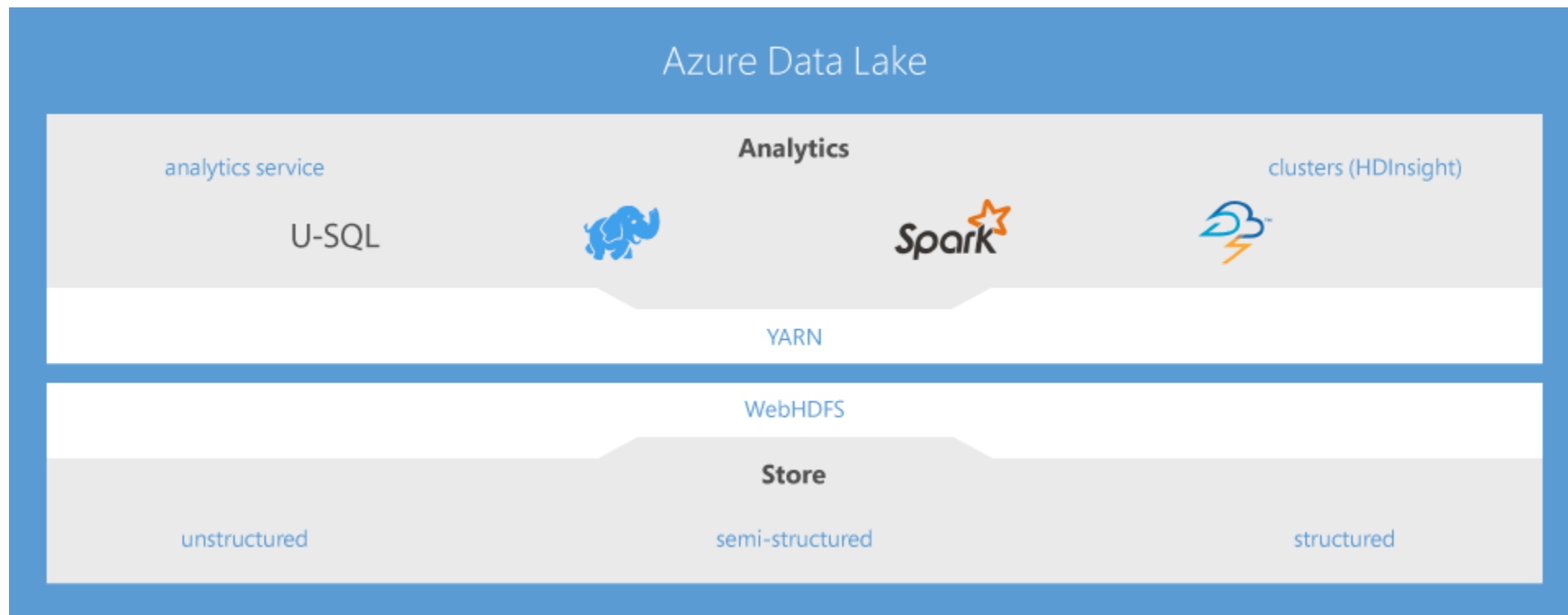
# MULTIPLES STORAGE ACCOUNTS



# AZURE DATA LAKE COMO HDFS

- Data Lake Store es un repositorio de datos en su formato original
  - Diseñado para volúmenes infinitos y alto rendimiento en el procesamiento y el análisis
  - Gran cantidad y variedad, rapidez, estructurados y no estructurados...lo que viene siendo Big Data
- Azure Data Lake Analytics es un servicio analítico distribuido construido sobre YARN
  - Permite ejecutar jobs en U-SQL directamente contra los datos almacenados en el Data Lake
- Azure Data Lake HDI es HDInsight sobre Azure Data Lake
  - Las mismas características de HDInsight, pero trabajando con datos almacenados en Data Lake

# AZURE DATA LAKE COMO HDFS



# ADMINISTRACION DE HDINSIGHT

- Los clusters de HDInsight pueden gestionarse desde el portal de Azure
  - Creación y destrucción de clusters
  - Escalado
  - Conexión por RDP
  - Ejecución de queries desde el portal
  - Consulta de la UI de YARN
- También podemos utilizar Azure PowerShell
- Si los clusters son Linux, podemos administrarlos usando Ambari
  - Si son Windows también, pero necesitaremos usar la API (no hay Web UI)

**plain concepts**

ADMINISTRACIÓN DE CLUSTERS



**plain concepts**

HDINSIGHT Y POWERSHELL





# PROS Y CONTRAS DE HDINSIGHT



## Pros

- Facilidad de despliegue, clúster desplegado en unos 25 minutos
- Soporte de Hadoop por parte de Microsoft (y Hortonworks)
- Separación entre los datos y la computación
- SLA en disponibilidad del clúster
- Gestión por parte de MS de actualizaciones de SO, VM, Hadoop (PaaS)

## Contras

- Restricciones inherentes a un servicio PaaS
- Ecosistema menos maduro

# INSTALACION DE HWX EN AZURE

- Sandbox
- Multinodo
- Automatización con PowerShell

# HORTONWORKS SANDBOX

- Maquina virtual de VMWare o Virtualbox
- Todo lo necesario para desplegar un cluster de Hadoop
  - En local
  - Con un único nodo
  - Sobre Linux
- Nos permite hacer pruebas en una MV local sin necesidad de desplegar Hadoop desde cero
- Tambien podemos desplegarla en Azure desde el Marketplace

**plain concepts**

HORTONWORKS SANDBOX



# HORTONWORKS DATA PLATFORM

- Nos permite desplegar toda la infraestructura de Hadoop empresarial en Azure
  - Sobre maquinas virtuales que podemos gestionar a nuestro antojo
- No es necesario que configuremos nada
  - La configuración por defecto nos permite comenzar a trabajar
- Versión de evaluación con un número reducido de nodos

# HORTONWORKS DATA PLATFORM

- Data Management
  - YARN para gestionar los recursos, HDFS para almacenar los datos
- Data Access
  - MR, Pig, Hive, TEZ...
  - Pero tambien HBase (NoSQL), Storm (data streaming), Spark (in memory)...
- Data Governance & Integration
  - Sqoop, Oozie, Flume...

**plain concepts**

DESPLEGANDO HDP IN  
AZURE



# INSTALACION DE CLOUDERA EN AZURE

- Cloudera Enterprise Data Hub
- Automatización con PowerShell



# CLOUDERA ENTERPRISE DATA HUB

- Maquinas virtuales en Azure Marketplace
- Despliega el Enterprise Data Hub de Cloudera
  - En maquinas DS13 o DS14
  - Con un número de nodos variable entre 3 y 90
  - Sobre Linux
- Nos permite configurar entornos para hacer pruebas de concepto, o en producción

**plain concepts**

DESPLEGANDO CEDH EN  
AZURE



# CLOUDERA CON POWERSHELL

<https://github.com/Azure/azure-quickstart-templates/tree/master/cloudera-on-centos>

**plain concepts**  
CLOUDERA CON  
POWERSHELL

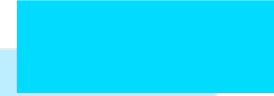


# PROS Y CONTRAS DE HDP Y CLOUDERA



## Pros

- Customización completa de la experiencia
- Soporte a todo el stack de Hadoop
- Similitudes con solución on-premises
- Servicios “propios” de cada distribución, como Impala



## Contras

- Mayores costes de mantenimiento
- Mayor coste operacional (riesgos de seguridad...)

# AMBARI

- Administración de clusters Hadoop
- API programática

# AMBARI

- Ambari es una plataforma Open Source para provisionar, gestionar, monitorizar y securizar clusters de Hadoop
- Dispone de una interfaz web y una API REST
- Extensible y customizable

# USER VIEWS

- Tez
  - Permite visualizar y optimizar el uso de recursos del cluster
- Hive
  - Permite ejecutar consultas HiveQL (ANSI SQL)
- Pig
  - Permite ejecutar scripts Pig
- Capacity Scheduler
  - Permite gestionar workloads y queues en YARN
- Files
  - Permite gestionar ficheros en HDFS



plain concepts

AMBARI WEB UI



**plain concepts**

CLOUDERA MANAGER



# LA API DE AMBARI

- Ambari ofrece una API REST para acceder a sus características
- En los clusters Windows de HDInsight es la única opción
  - Los clusters Linux si tienen Web UI

plain concepts

AMBARI API





# ¿PREGUNTAS?



# GRACIAS

Barcelona



Bilbao



Madrid



Sevilla



Dubai



London



Seattle