

Estilos arquitecturales (Patrones arquitecturales)

Estilo arquitectural

- Expresa la organización estructural de un software
- Provee tipos de elementos y sus responsabilidades
- Da guías en cuanto a cómo se relacionan dichos elementos
- Se escoge en función de los atributos de calidad

¿Cuál es la diferencia entre estilo y vista arquitectural?

- Estilo
 - lenguaje para hablar de elementos de un sistema y relaciones
- Vista
 - representación/notación gráfica del lenguaje
 - entendida por ciertos stakeholder

CERN

<https://www.youtube.com/watch?v=3wtUr3iVVlw> (CERN)

Cern

- Responder la pregunta: cómo el universo funciona y de qué está hecho?
- 40 millones de fotos por segundo de la colisión de protones
 - 1 petabyte de datos por segundo que necesita análisis
- Guardar los datos producidos en 20 años, por año 35 petabytes (10^{15} bytes)
- Disminuir costos y ser eficientes
 - **Hadoop Platform as a Service in the Cloud**

Hadoop

- Hadoop Distributed File System (HDFS) es un sistema de archivos distribuídos

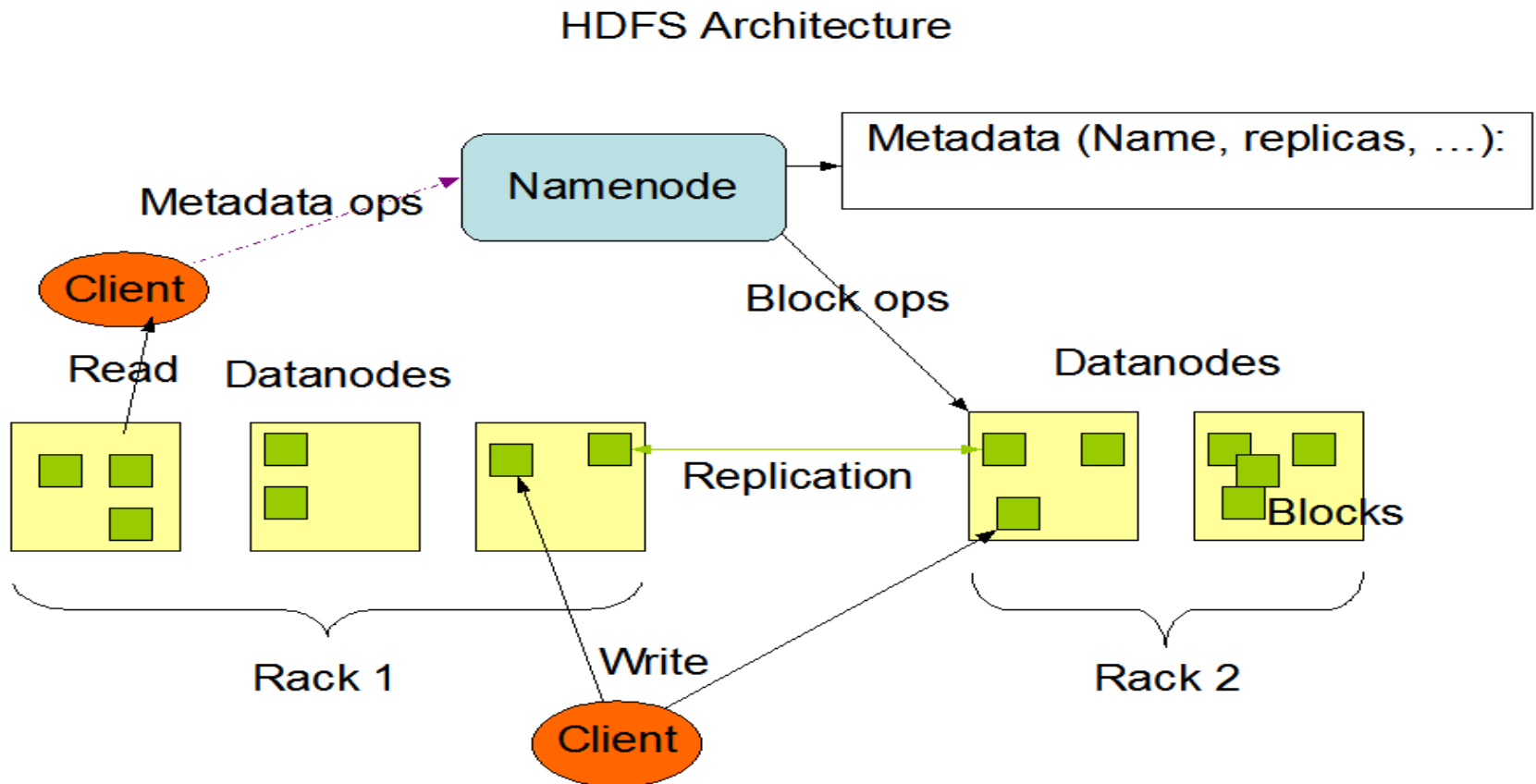
¿A qué atributos de calidad apunta?

- Desempeño
 - Rápido acceso a datos
 - Grandes cantidades de datos, e.g., petabytes
- Integridad de los datos
- Disponibilidad y tolerancia a fallos
 - Detección de fallos, recuperación rápida y automática

Decisiones arquitecturales

- Los datos se colocan cerca al cliente que los necesita
- Diseño para procesamiento por lotes (batch) en lugar de interactivo
- Arquitectura maestro/esclavo*

Arquitectura maestro/esclavo*



Name node: mapping, regula acceso

Data nodes: crean, suprimen y replican bloques

Client

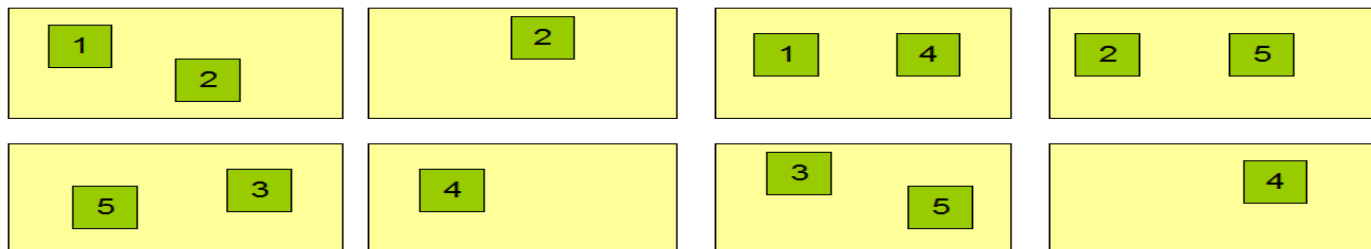
Desempeño

- Múltiples copias de datos
- Namenode tiene metadata para saber dónde las copias están
 - Actualizaciones sincrónicas

Block Replication

Namenode (Filename, numReplicas, block-ids, ...)
/users/sameerp/data/part-0, r:2, {1,3}, ...
/users/sameerp/data/part-1, r:3, {2,4,5}, ...

Datanodes



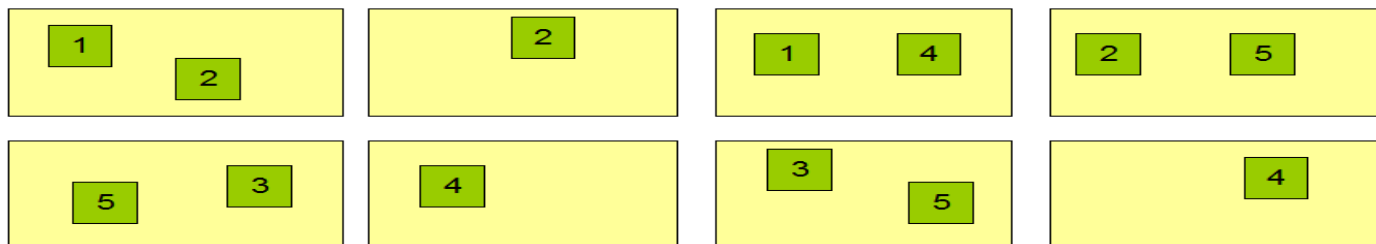
Integridad de los datos

- Bloques de un DataNode pueden estar corruptos
 - fallas en disco duros/red o bugs del software
- Cliente calcula un checksum del contenido de cada bloque y lo compara con los checksum iniciales (archivos ocultos)
- Si el checksum de un bloque es diferente, el cliente busca el bloque en otro DataNode

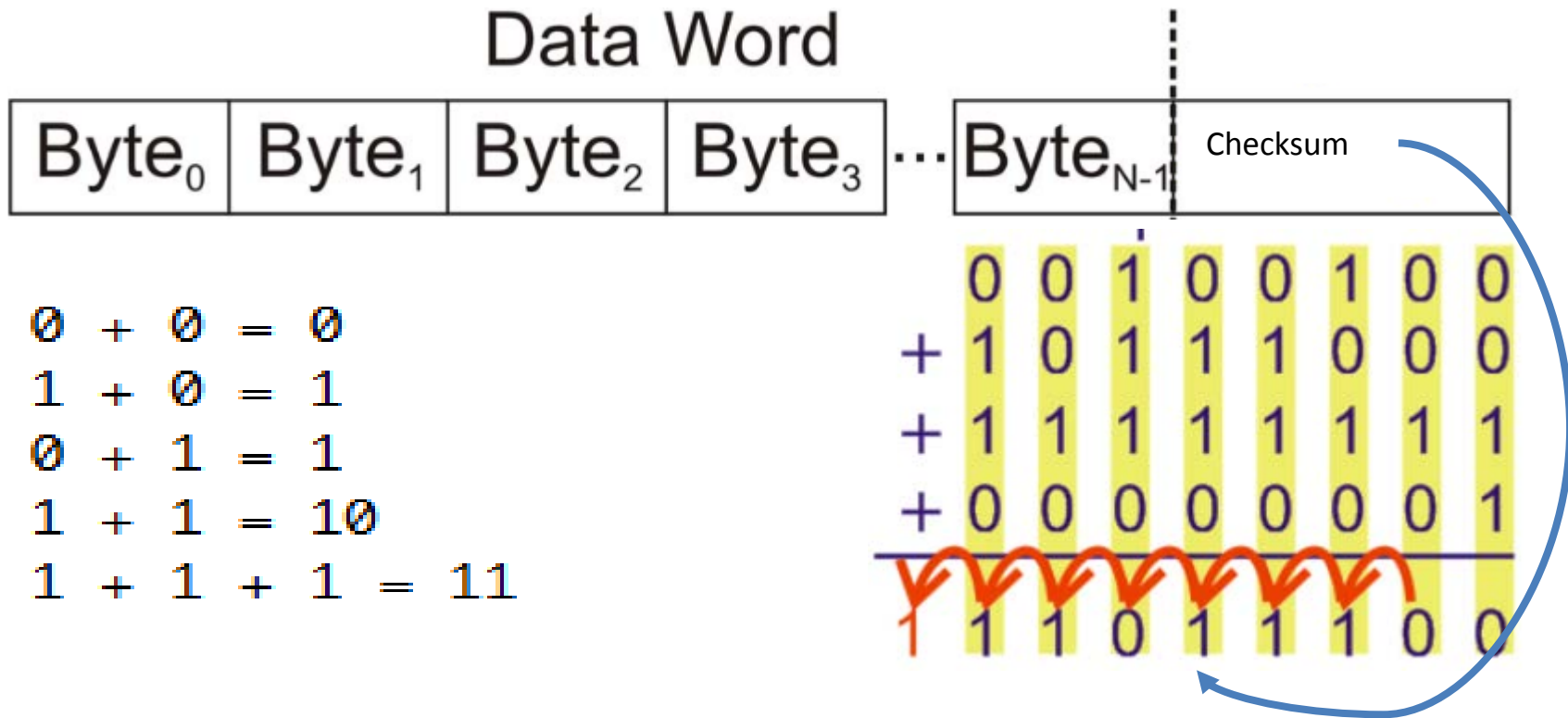
Block Replication

```
Namenode (Filename, numReplicas, block-ids, ...)  
/users/sameerp/data/part-0, r:2, {1,3}, ...  
/users/sameerp/data/part-1, r:3, {2,4,5}, ...
```

Datanodes

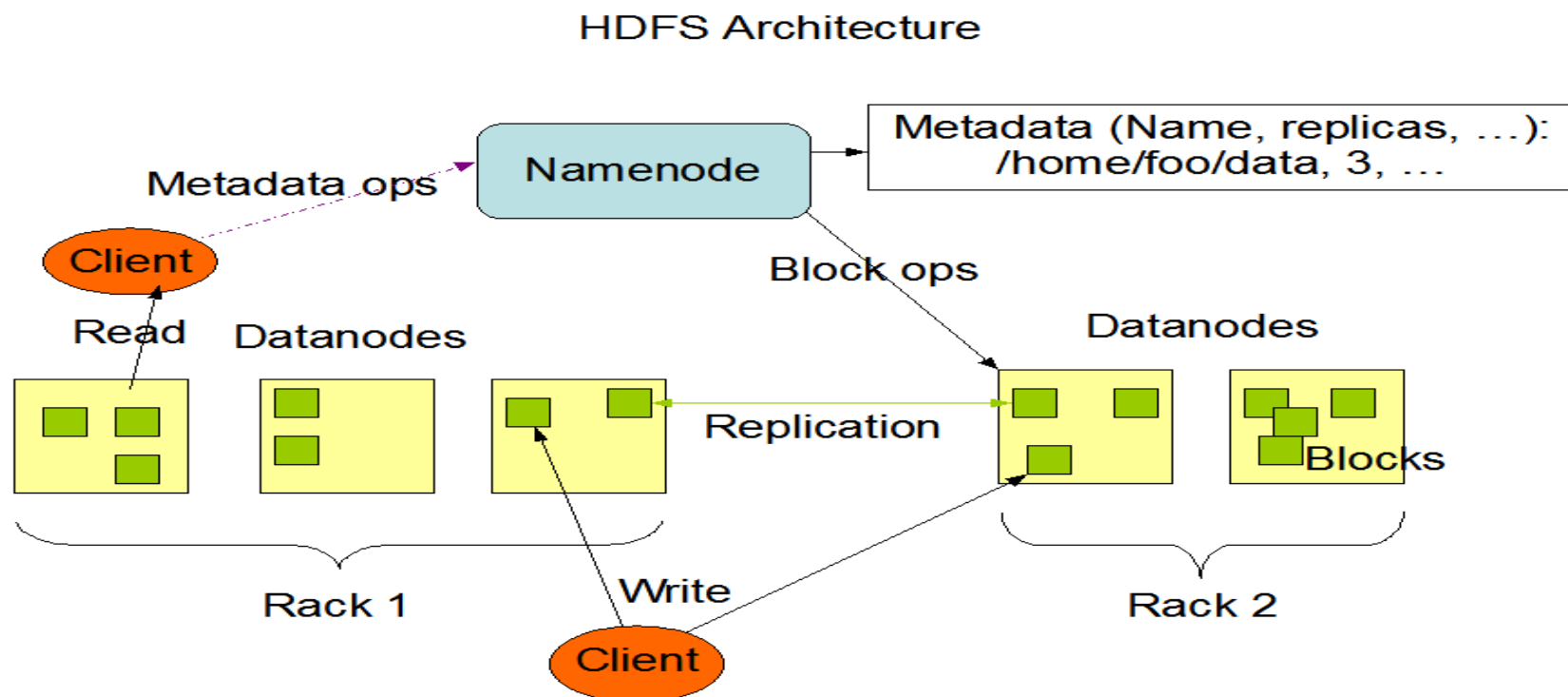


Checksum



Disponibilidad y tolerancia a fallas

- DataNode envía un heartbeat al NameNode periódicamente
- NameNode detecta problemas de conectividad si heartbeat ausente
- NameNode marca el DataNode como muerto y no le envía más peticiones
- NameNode hace más réplicas de bloques que se encontraban en nodo muerto



Netflix

<https://www.youtube.com/watch?v=W3mIZMwnX6k>

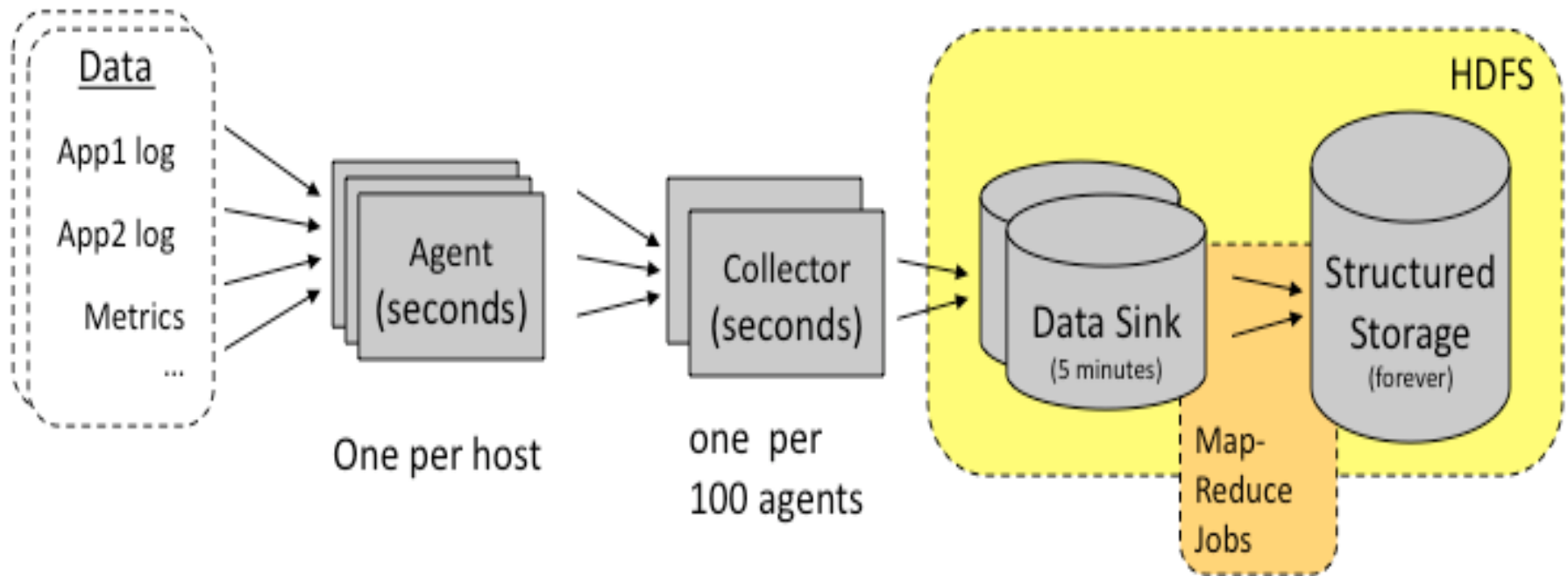
Netflix

- Streaming (difusión en flujo) de videos en diferentes dispositivos sobre demanda
- Cliente puede parar y reiniciar un video cuando desee
- En 2011, Netflix consumió el 22% de TODO el tráfico de internet
- Alquiler de DVDs en USA

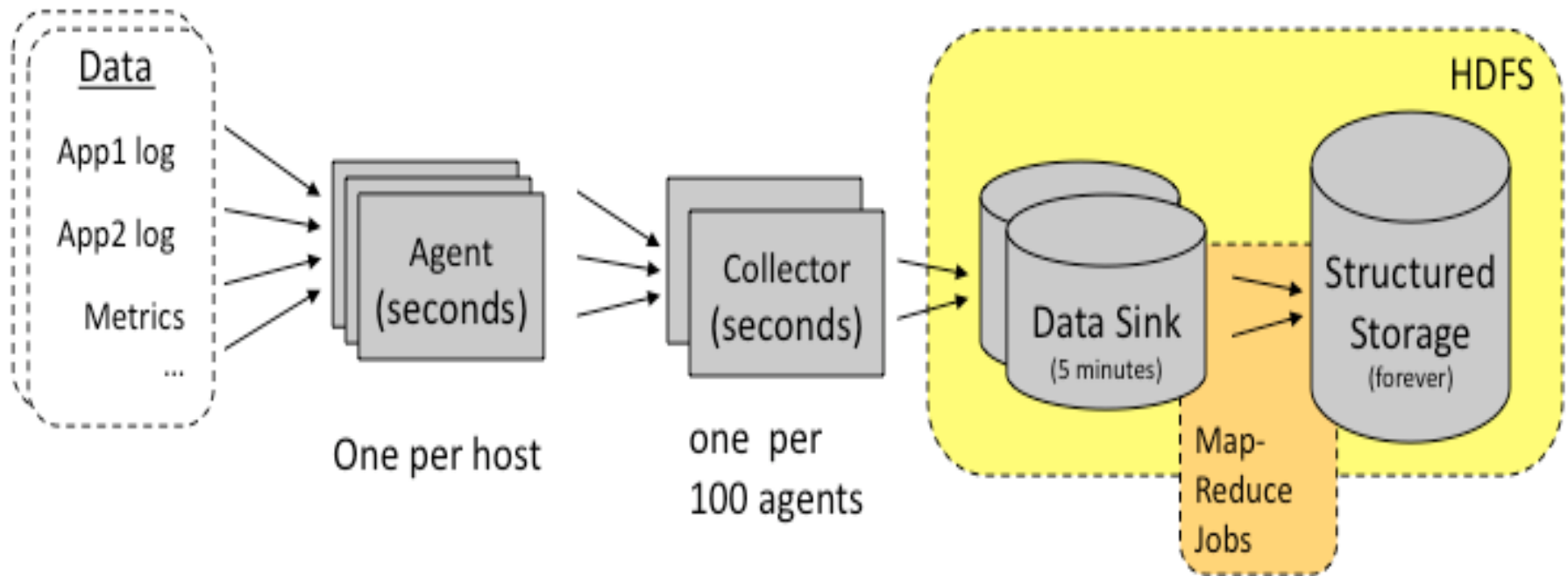
Suro

- Para toma de decisiones es crítico coleccionar datos de lo que ocurre y esto en tiempo real
 - 1.5 millones de eventos por segundo durante los picos
 - 80 millones de eventos por día
- Los datos pueden ser:
 - Logs
 - Registros de actividad de usuario
 - Datos operacionales
 - Etc.

Suro architecture



Suro architecture

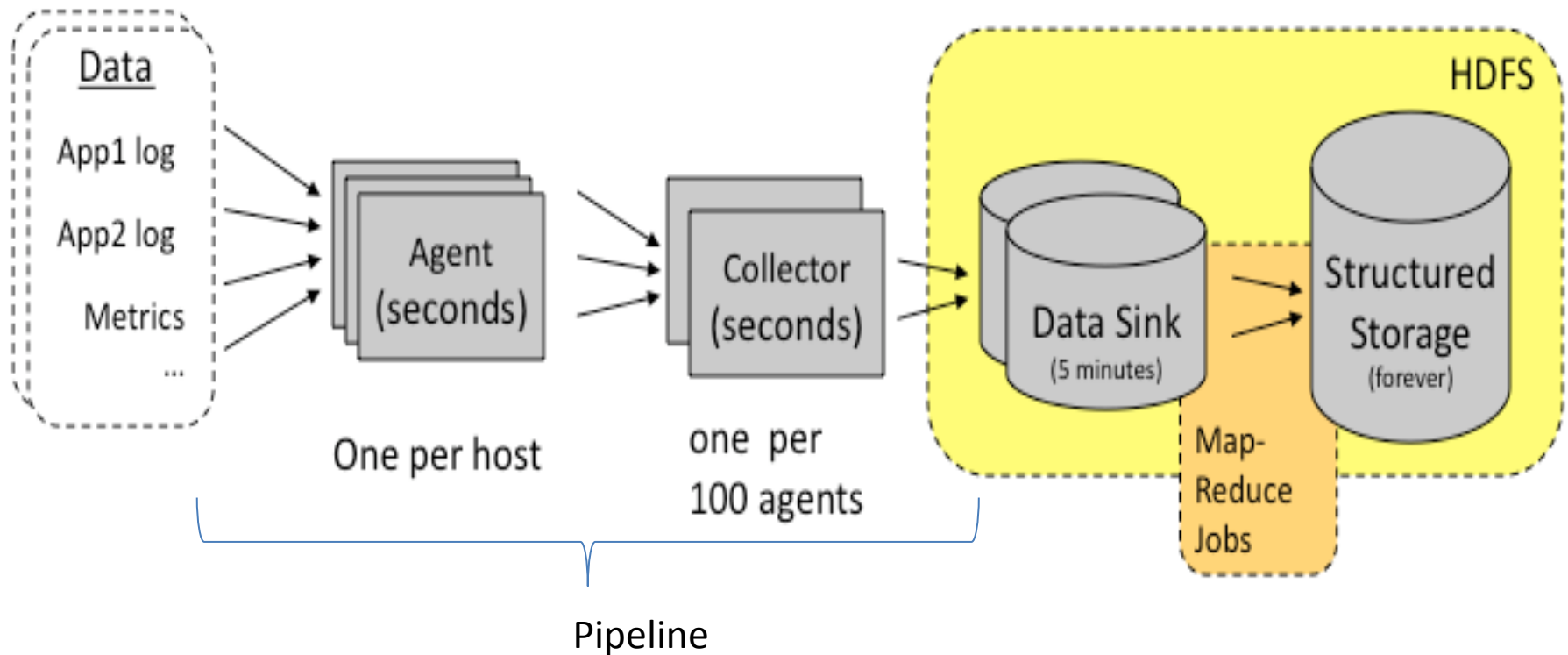


Sequence file

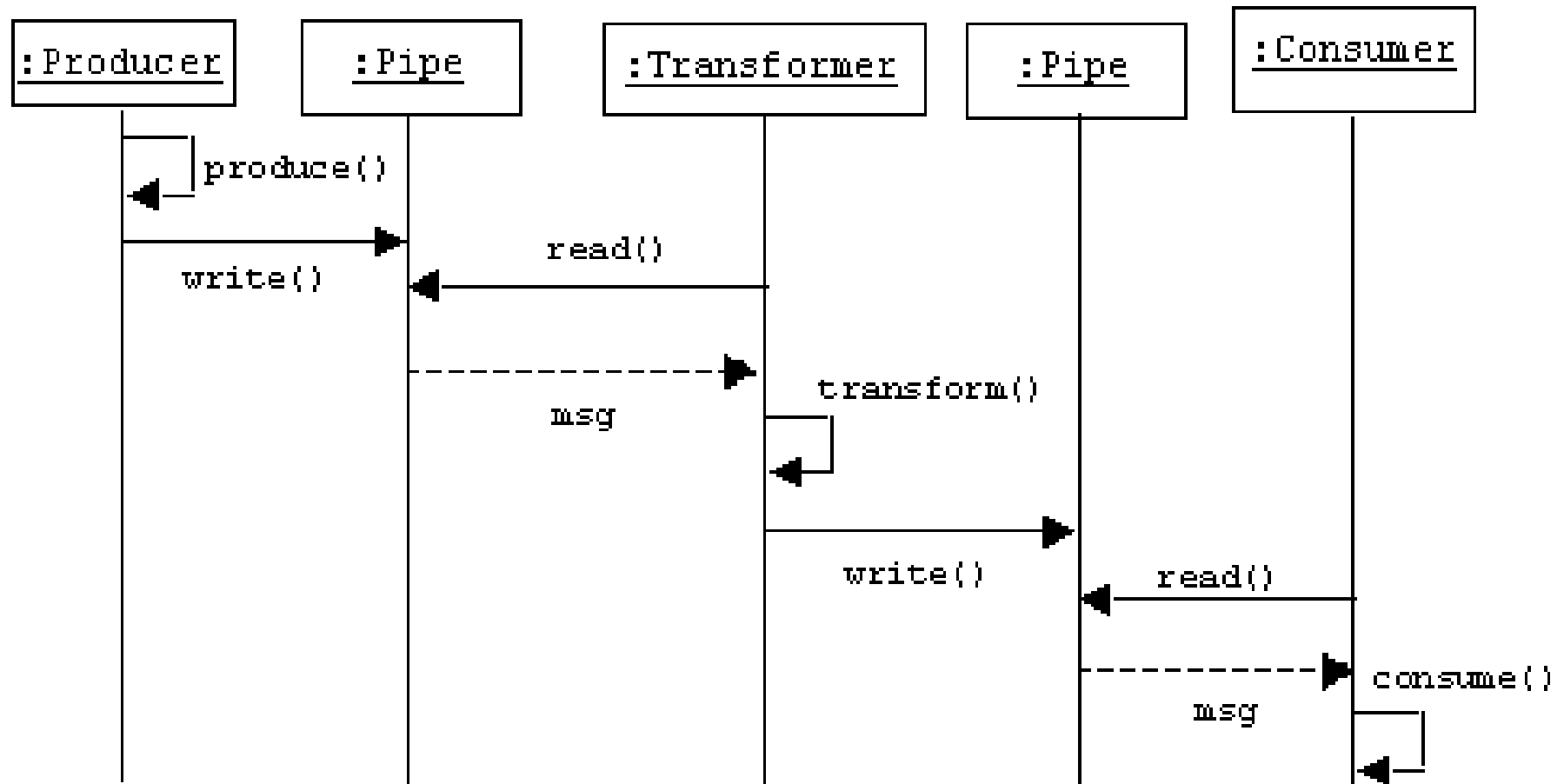
Los
primeros
1000 bytes

→	1000	App1Log-part0
	2000	App1Log-part1
	3000	App1Log-part2

Suro architecture



El productor escribe un msg en pipe1 (memoria temporal), el transformador lo lee, transforma y copia en pipe2. El consumidor lee y consume el mensaje



Map reduce

- Sequence file no es muy conveniente para hacer análisis, ya que cada tupla es vista como algo aislado

Sequence file

1000	App1Log-part0
2000	App1Log-part1
3000	App1Log-part2

- Suro usa Map reduce para organizar y procesar los datos de estos archivos **GRANDES**

Fundamentos del map reduce

- Map
 - Entrada <id documento, contenido>
 - Salida un map <palabra, lista de valores>
- Reduce
 - Entrada un map <palabra, lista de valores>
 - Salida un map <palabra, lista de valores reducida>

Ejemplos de aplicación de map reduce

- Buscar la película más vista en 2014
- Contar el número de veces que una URL de película se encuentra en los logs de películas vistas

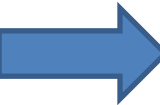
Sequence

file

key	value
1000	App1Log-part0
2000	App1Log-part1
3000	App1Log-part2



```
map (key, value)
for each url u in value:
if partitioni.containsKey(u)
    list = partitioni.get(u);
else
    list = new List();
list.add("1");
partitioni.put(u, list);
```



Partition i

Key	value
https://www.net flix.com/Hercule s	{"1", "1", "1"}

Ejemplo de aplicación (cont.)


Sequence
file

key	value
1000	App1Log-part0
2000	App1Log-part1
3000	App1Log-part2

Ejemplo de aplicación (cont.)

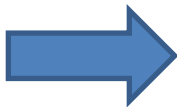
Partitioni

```
reduce (Map partitioni)
for each u in partitioni.getKeys()
value = partitioni.get(u);
if (output.containsKey(u))
result = output.get(u);
else
result = 0;
result = result + value.size();
output.put(u, AsString(result))
```



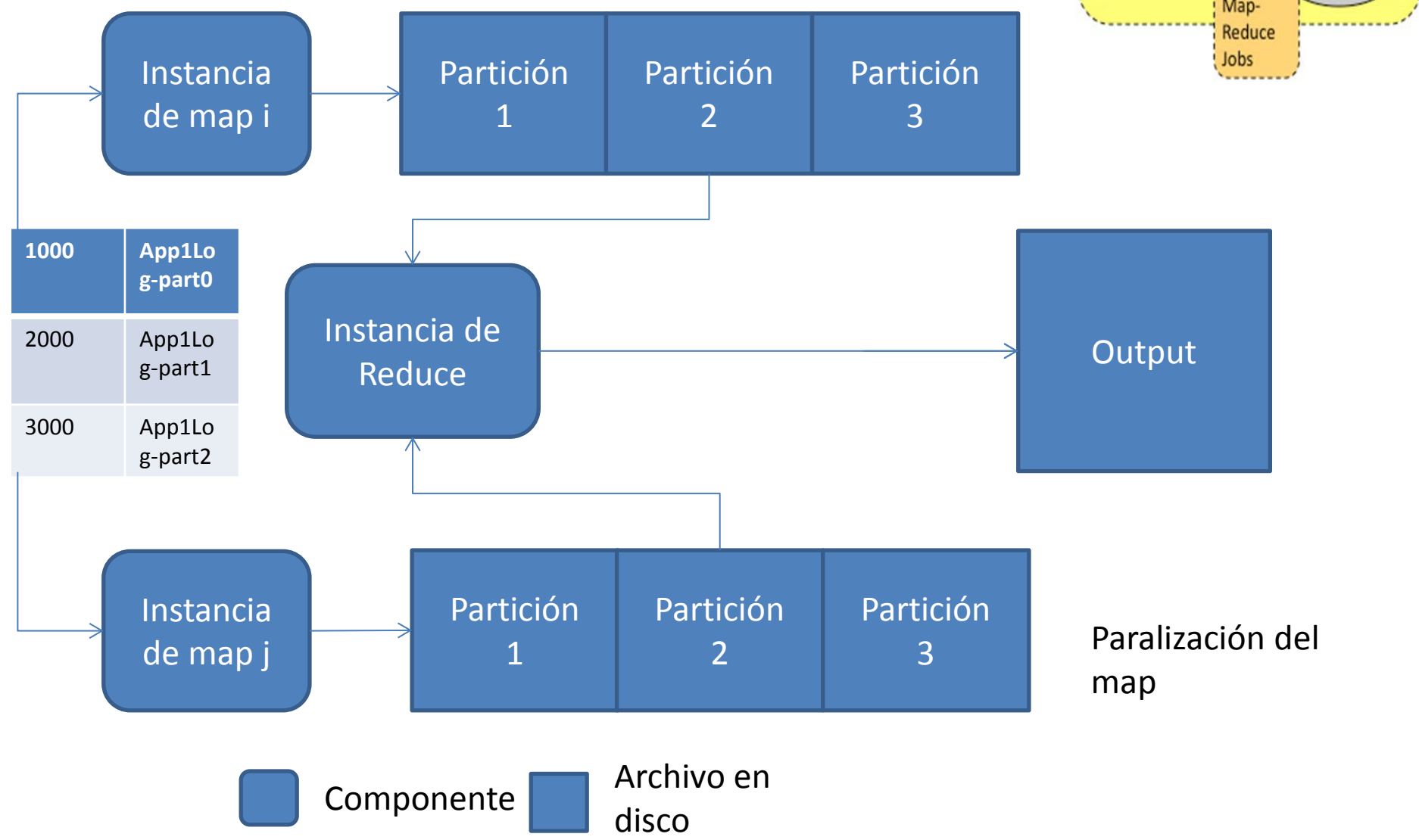
Key	value
https://www.net flix.com/Hercule s	{"1", "1", "1"}

Output



Key	value
https://www.net flix.com/Hercule s	3

Arquitectura de map reduce

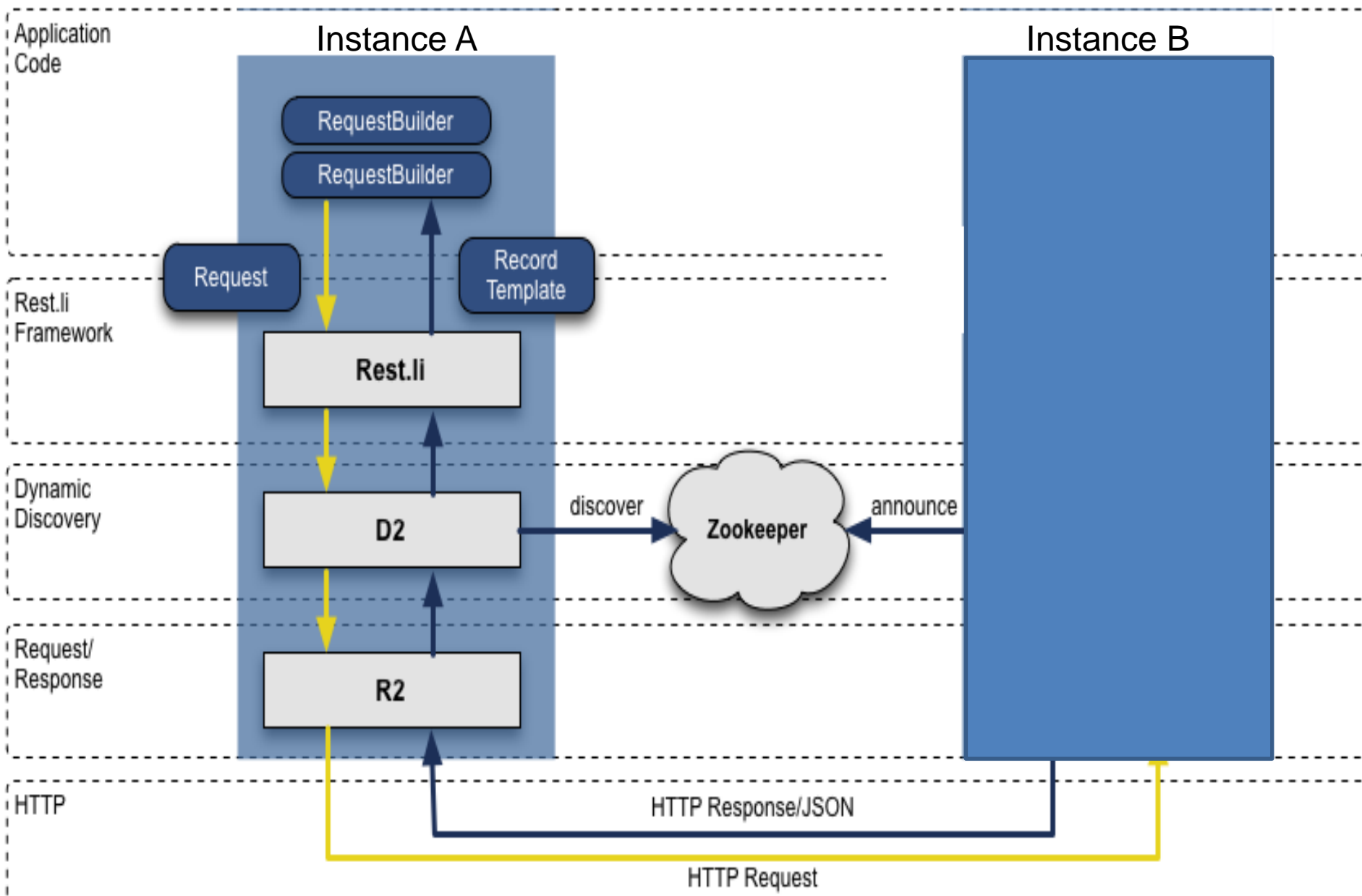


Otros ejemplos de aplicación

- Encontrar el no. de ocurrencias de un string particular en un documento
 - Ejemplo tweets
- Sacar la lista de las palabras más importantes que se encuentran en un documento

Linkedin

- Antes Linkedin era un sistema monolítico
- Desde hace algunos años empezó a usar una arquitectura orientada a servicios
- Motivaciones detrás de la decisión
 - Escalar en función del tráfico del sitio
 - Desarrollo fácil y rápido para los programadores
- Solución
 - Arquitectura Rest.li (diapositiva a continuación)

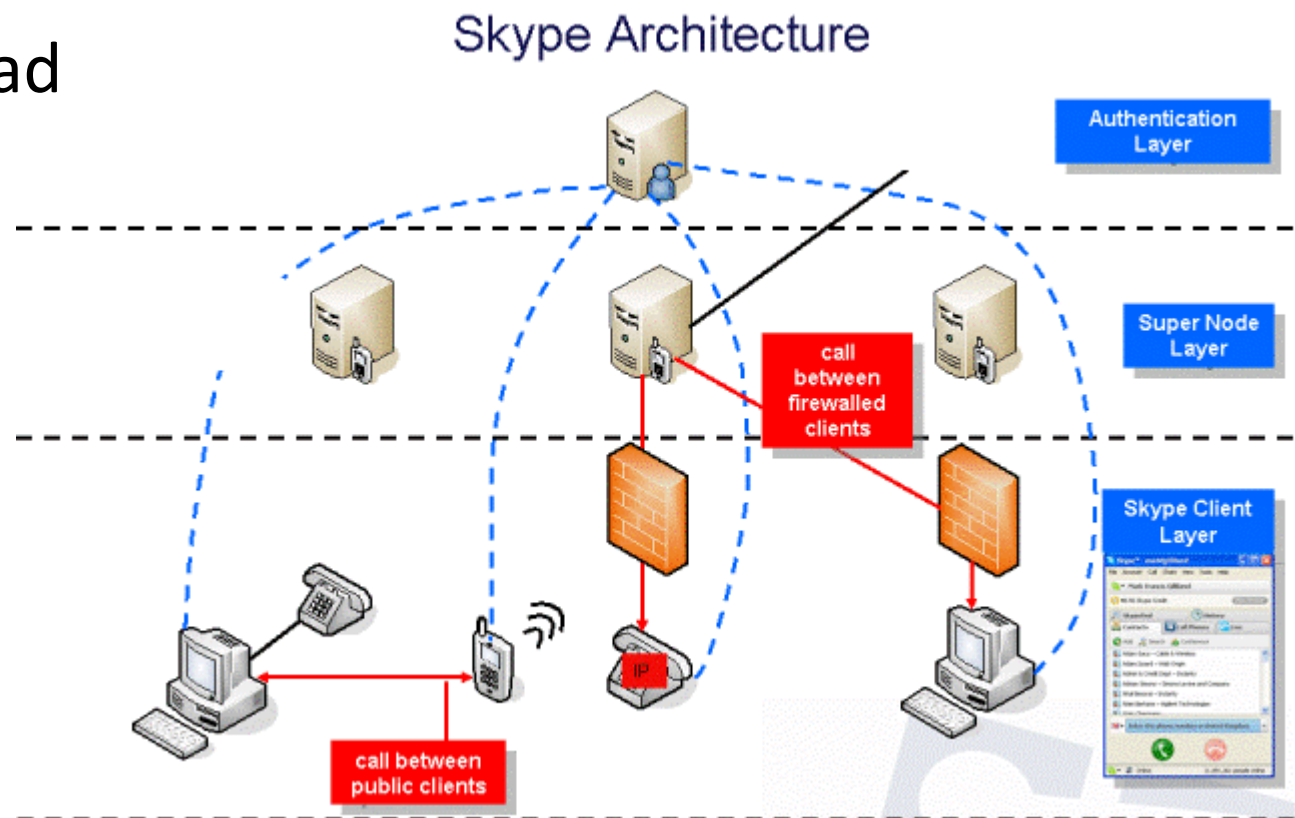


Skype (Arquitectura peer-to-peer)

- Negocio
 - Llamada gratis entre usuarios skype
 - Llamada paga a fijo/celular de múltiples países
 - En 2007, 5.4 billones de minutos Skypeout
 - Alquiler de un número que le permitirá recibir/hacer desde su fijo/celular llamadas de/a otros usuarios
 - En 2008, 276 millones de cuentas de usuario

Skype (Arquitectura peer-to-peer)

- Requerimientos
 - Escalabilidad
 - Desempeño
 - Disponibilidad



Listado de estilos arquitecturales vistos en clase

- Layer
- Cliente/servidor
- Maestro/esclavo
- Map reduce
- MVC
- Push
- Pull
- Pipeline
- Multi-tier
- SOA
- Peer-to-peer