APACHE FLUME

JPA Solutions
Center Of Excellence

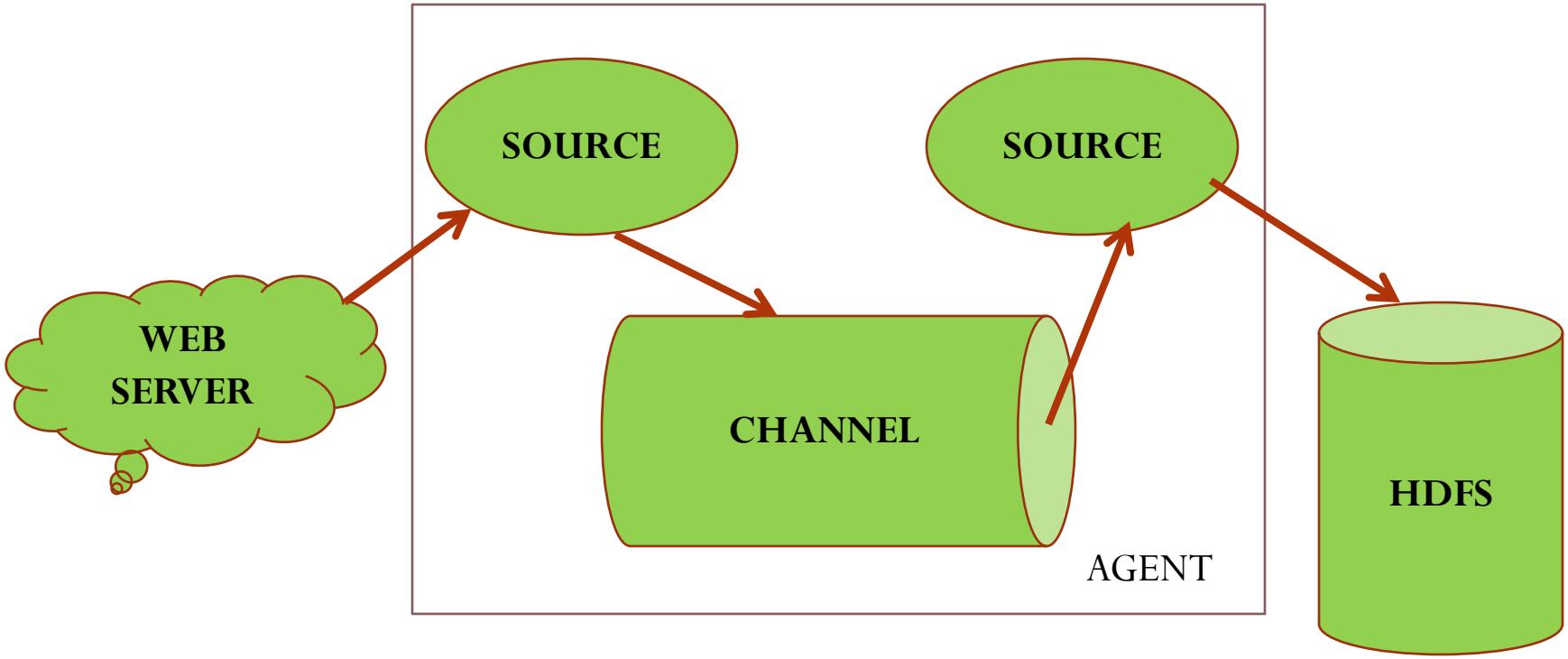
What is Apache Flume ??

- Reliable Service for Collection and aggregation of large amount of data
  - Especially streaming Data (eg. Log data)
- Distributed

Fundamental Unit - Event

- An unit of data flow having a byte payload and an optional set of string attributes
- Flume transfers from the event from its Origin to Destination
- Optionally Headers also can be added as part of an event

Agent



A JVM Process hosting the components (Sources , Sinks, channels) which transports the events

Components

- Source
  - An Active Component which receives the event and places it in the Channel
- Channel
  - A passive component which buffers the event and send it to the Sink
- Sink
  - Writes the data into next hop or final destination

An Example Configuration

```
# Name the components on this agent
a1.sources = r1
a1.sinks = k1
a1.channels = c1

# Describe/configure the source
a1.sources.r1.type = netcat
a1.sources.r1.bind = localhost
a1.sources.r1.port = 44444

# Describe the sink
a1.sinks.k1.type = logger

# Use a channel which buffers events in memory
a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100

# Bind the source and sink to the channel
a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

Agent is a1
Source is r1
Channel is c1
Sink is k1

Configuration

JPA Solutions
Center Of Excellence

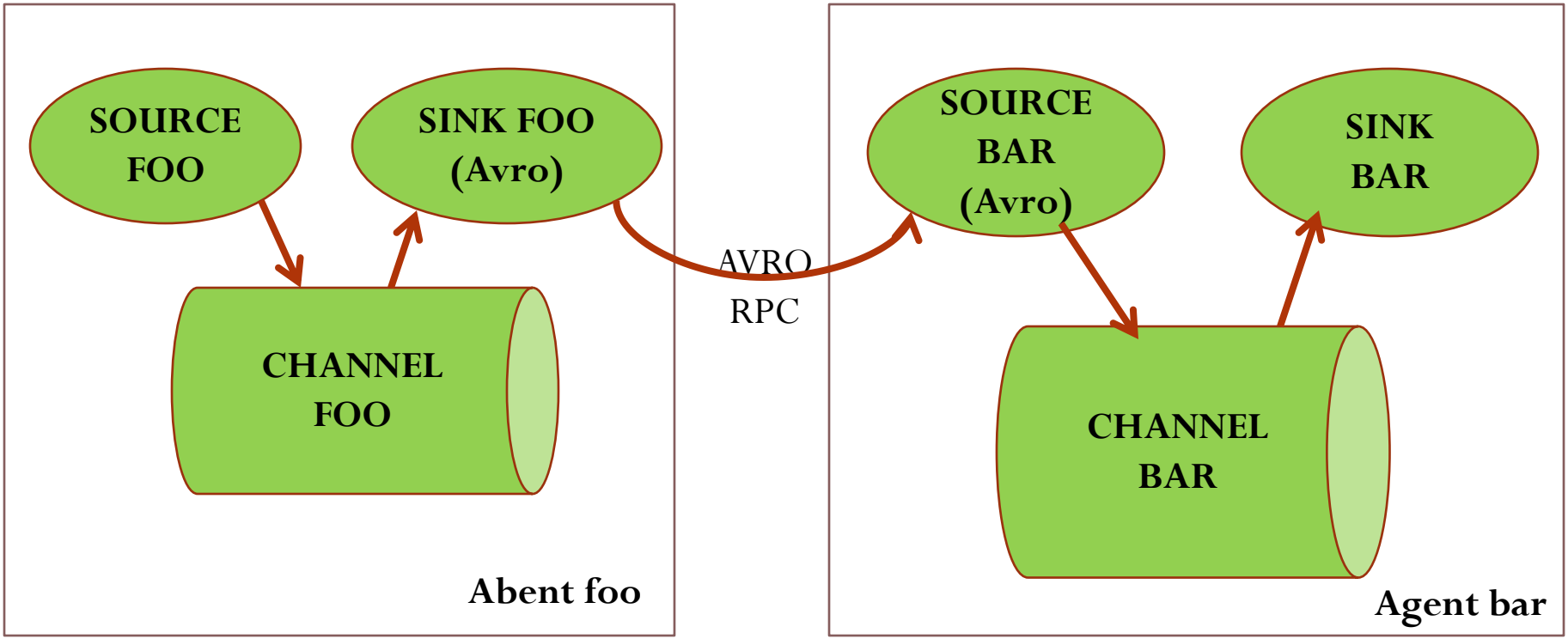
- Follows hierarchical Configuration which follows key Value pair
- A conf file can have many agents but only named agents are loaded

- Basic rules
  - Every Agent must have at least one channel
  - Every Source must have at least one channel
  - Every Sink must have exactly one channel
  - Every Component Must have a type

JPA Solutions
Center Of Excellence

A Sample Demo available

Multi Agent



Courtesy : flume.apache.org

Sources

- Event Driven
- Read data from clients or Sinks
- Stores events in channels
- Supports Batching
- Fan Out Flow
- Interceptors

NetCat Source

- Listens on a given port
- Converts each line of data into an event

```
a1.sources = r1
a1.channels = c1
a1.sources.r1.type = netcat
a1.sources.r1.bind = 0.0.0.0
a1.sources.r1.bind = 6666
a1.sources.r1.channels = c1
```

Exec Source

- Reads data from the output of a unix command or script
  - Eg: tail -F <filePath>

```
a1.sources = r1
a1.channels = c1
a1.sources.r1.type = exec
a1.sources.r1.command = tail -F /var/log/secure
a1.sources.r1.channels = c1
```

JMS Source

- Reads messages from JMS destination (queue/topic)
- Has pluggable converters to convert the Bytes, Text, and Object messages to Flume Events

```
a1.sources = r1
a1.channels = c1
a1.sources.r1.type = jms
a1.sources.r1.channels = c1
a1.sources.r1.initialContextFactory = org.apache.activemq.jndi.ActiveMQInitia
a1.sources.r1.connectionFactory = GenericConnectionFactory
a1.sources.r1.providerURL = tcp://mqserver:61616
a1.sources.r1.destinationName = BUSINESS_DATA
a1.sources.r1.destinationType = QUEUE
```

Spooling Directory Source

- Checks for new files for the configured directory and generates new events out of them

```
agent-1.channels = ch-1
agent-1.sources = src-1
agent-1.sources.src-1.type = spooldir
agent-1.sources.src-1.channels = ch-1
agent-1.sources.src-1.spoolDir = /var/log/apache/flumeSpool
agent-1.sources.src-1. deletePolicy = imediate
```

Other Sources

- Sequence generator source
  - Generates counter starting 0 and incrementing 1
- Syslog Source
  - Has both UDP ass well as TCP sources
  - Has multiport Syslog source also
- Legacy Sources
  - Avro Source
  - Thrift Source
- Custom Source - can be wriiten by implementing Source Interface

Contextual Routing

JPA Solutions
Center Of Excellence

- Provided by
  - Interceptors
  - Channel selectors

Interceptors

JPA Solutions
Center Of Excellence

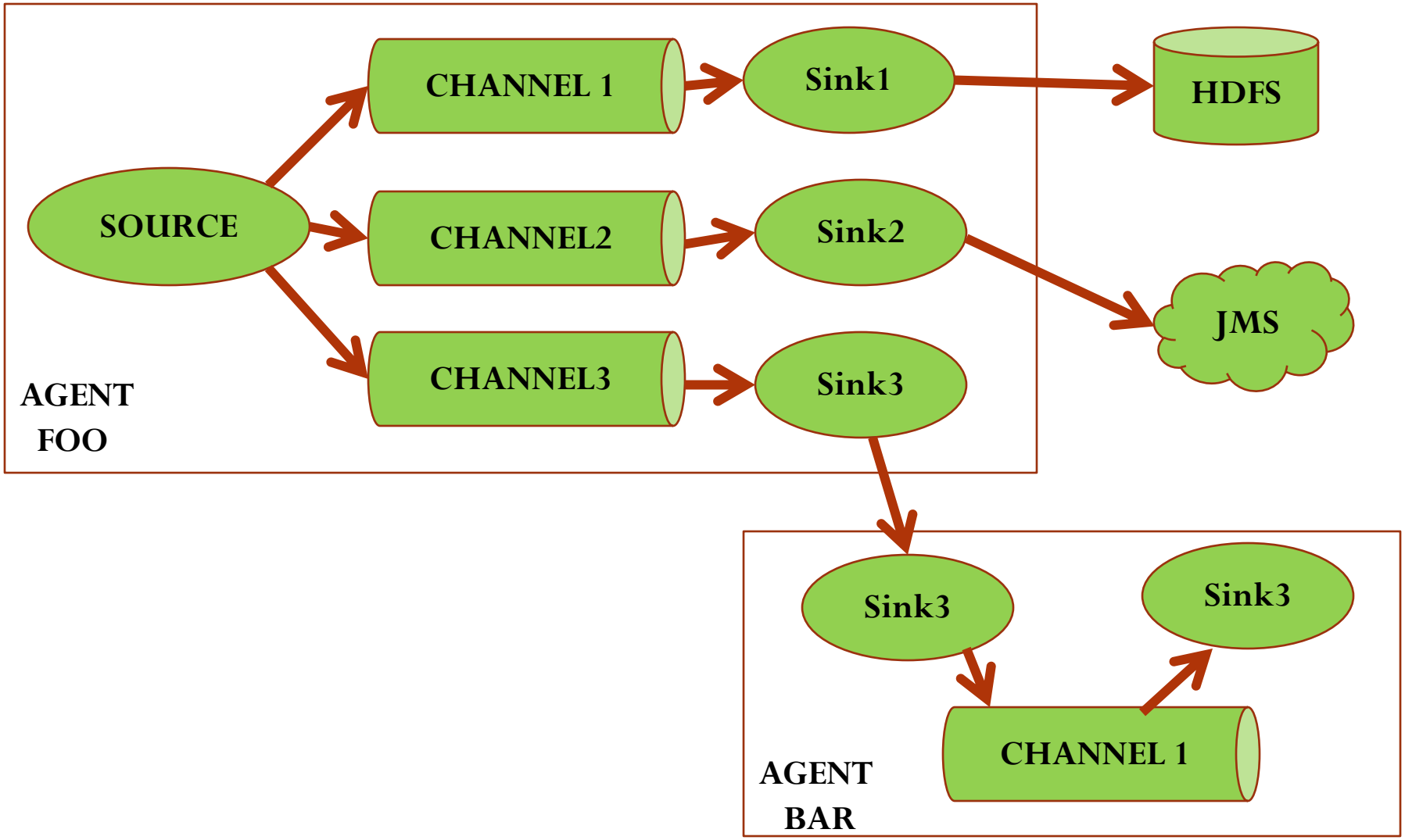
- To Modify or drop events on the flight
- Examples
  - Timestamp Interceptors
  - Host Interceptors
  - Static Interceptors
  - Regex Filtering/Extractor interceptors
- supports chaining of Interceptors
- To Write Custom Interceptors classes should implement `org.apache.flume.interceptor.Interceptor` interface
- Example

Fan out Flow - Channel Selectors

- Allows a source to select one or more channels based on criteria
- Available Channels
  - Replicating - duplicating the events
  - Multiplexing - routing based on headers

JPA Solutions
Center Of Excellence

Fan Out


SOURCE
CHANNEL 1
CHANNEL2
CHANNEL3
Sink1
Sink2
Sink3
HDFS
JMS
AGENT
FOO
Sink3
Sink3
CHANNEL 1
AGENT
BAR

Channels

JPA Solutions
Center Of Excellence

- Passive Component Buffers the events onto it.
- Bundled(built in) channels as part of Flume
  - Memory Channel - lower latencies, not durabilities
  - File Channel - used often, uses disk based queue
  - JDBC - uses a DB as backstore

Sink

- Writes the data to next Hop or destination
- Supports Batching
- Always attached to a single channel

HDFS Sink

- Writes the event in HDFS in the form of text or sequence files.
- Supports compression
- Can make use of headers for dynamic path(file/directory)

JPA Solutions
Center Of Excellence

HDFS Configuration

Name	Default	Description
channel	–	
type	–	The component type name, needs to be `hdfs`
hdfs.path	–	HDFS directory path (eg hdfs://namenode/flume/webdata/)
hdfs.filePrefix	FlumeData	Name prefixed to files created by Flume in hdfs directory
hdfs.fileSuffix	–	Suffix to append to file (eg `.avro` - *NOTE: period is not automatically added*)
hdfs.inUsePrefix	–	Prefix that is used for temporal files that flume actively writes into
hdfs.inUseSuffix	`.tmp`	Suffix that is used for temporal files that flume actively writes into
hdfs.rollInterval	30	Number of seconds to wait before rolling current file (0 = never roll based on time interval)
hdfs.rollSize	1024	File size to trigger roll, in bytes (0: never roll based on file size)
hdfs.rollCount	10	Number of events written to file before it rolled (0 = never roll based on number of events)
hdfs.idleTimeout	0	Timeout after which inactive files get closed (0 = disable automatic closing of idle files)
hdfs.batchSize	100	number of events written to file before it is flushed to HDFS
hdfs.codeC	–	Compression codec. one of following : gzip, bzip2, lzo, lzop, snappy
hdfs.fileType	SequenceFile	File format: currently `SequenceFile`, `DataStream` or `CompressedStream` (1)DataStream will (2)CompressedStream requires set hdfs.codeC with an available codeC
hdfs.maxOpenFiles	5000	Allow only this number of open files. If this number is exceeded, the oldest file is closed.
hdfs.minBlockReplicas	–	Specify minimum number of replicas per HDFS block. If not specified, it comes from the def
hdfs.writeFormat	–	Format for sequence file records. One of "Text" or "Writable" (the default).
hdfs.callTimeout	10000	Number of milliseconds allowed for HDFS operations, such as open, write, flush, close. This operations are occurring.
hdfs.threadsPoolSize	10	Number of threads per HDFS sink for HDFS IO ops (open, write, etc.)
hdfs.rollTimerPoolSize	1	Number of threads per HDFS sink for scheduling timed file rolling
hdfs.kerberosPrincipal	–	Kerberos user principal for accessing secure HDFS
hdfs.kerberosKeytab	–	Kerberos keytab for accessing secure HDFS
hdfs.proxyUser		
hdfs.round	false	Should the timestamp be rounded down (if true, affects all time based es

File Roll Sink

- Stores Events on the local file System

```
a1.channels = c1
a1.sinks = k1
a1.sinks.k1.type = file_roll
a1.sinks.k1.channel = c1
a1.sinks.k1.sink.directory = /var/log/flume
```

Hbase Sinks

- Has two types - hbase & asynchbase
- Writes events to Hbase incrementally
- Hbase conf is picked up from hbase-site.xml

```
a1.channels = c1
a1.sinks = k1
a1.sinks.k1.type = hbase
a1.sinks.k1.table = foo_table
a1.sinks.k1.columnFamily = bar_cf
a1.sinks.k1.serializer = org.apache.flume.sink.hbase.RegexHbaseEventSerial
a1.sinks.k1.channel = c1
```

Other Sinks

- Avro Sink
- Thrift Sink
- Logger Sink
- ElasticSearch Sink
- MorphlineSolr Sink

- We can write our custom sink

Sink Processor

- Groups sinks into a single entitiy
- To provide Load balancing or fail over capabilities.

```
a1.sinkgroups = g1
a1.sinkgroups.g1.sinks = k1 k2
a1.sinkgroups.g1.processor.type = load_balance  (or failover)
```

Serializers

- Convert the event into user defined format
- EventSerializer is a file-oriented serialization interface
  - Supported in the HDFS sink and File Rolling sink
- HBaseSerializer (AsyncHBaseSerializer) used in Hbase

JPA Solutions
Center Of Excellence

ThAnK yOu