

HP IUM Fundamentals

FileService
SessionServers
Correlation & Aggregation

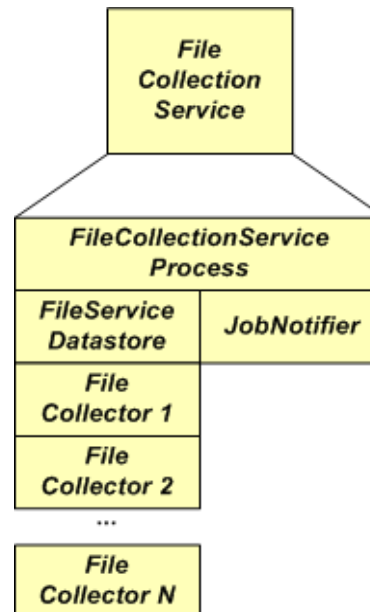


IUM Components - FCS

- What is a File Collection Service?
- Large voice/data network typically has many voice/data switches. Each switch generates files of usage data in the form of call detail records or CDRs
- The HP Internet Usage Manager (IUM) File Service reads multiple CDR files from multiple devices, typically voice switches but it can be other file sources as well. Thus averting the need to have the same number of collectors as switches
- The File Service can also read files from any other device or type of network that generates files containing usage data.

IUM Components - FCS

- Components in a File Collection Service

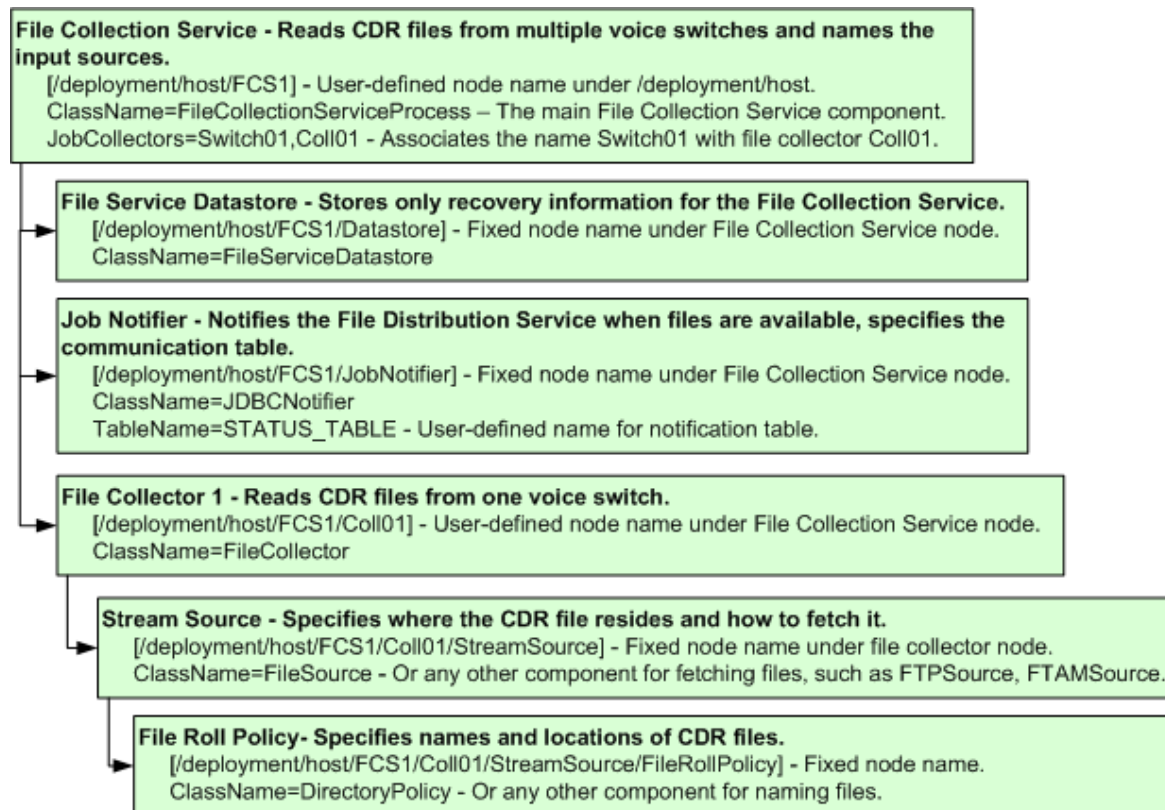


IUM Components - FCS

- Components in a File Collection Service
 - A **FileCollectionServiceProcess** is the top-level container component.
 - A **FileServiceDatastore** stores only recovery information for the file collection service. This is only used for recovery information when the file collection service restarts
 - A **JobNotifier** notifies the file distribution service when CDR files have arrived and are ready to be distributed to a collector. The JobNotifier names a database table where it places information about each CDR file. The file distribution service uses this table to obtain information about the input CDR files to pass each file to the appropriate collector
 - One **FileCollector** per input data source reads CDR files. The FileCollector is similar in some ways to an **encapsulator**. Each FileCollector reads CDR files from one voice switch

IUM Components - FCS

- Components in a File Collection Service: Hierarchy

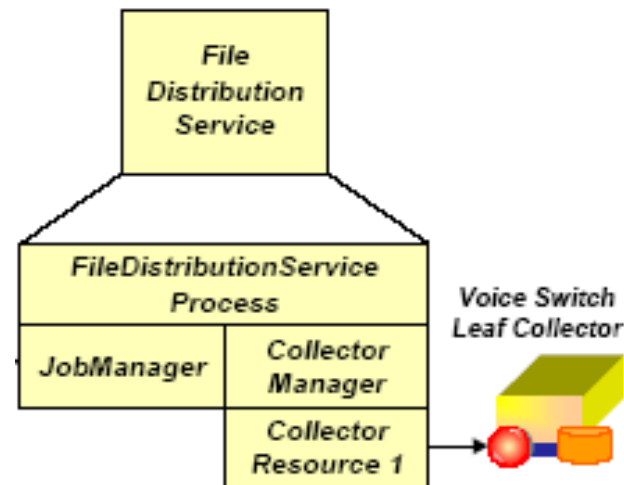


IUM Components - FDS

- What is a File Distribution Service?
- The FDS sends the data collected by a FCS to the 1st level collectors
- It retrieves data from the FCS, so both must be running at the same time
- It can transfer file from multiple data sources based on a scheduler
- The file distribution is based on a source-collector mapping supporting m:n mapping

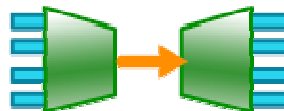
IUM Components - FDS

- Components in a File Distribution Service



IUM Components - FDS

- Components in a File Distribution Service
- The **JobManager** manages the information regarding the input files. It retrieves and updates file data from DB tables
- The **CollectorManager** manages information about individual resources (collectors). It also manages file allocation to collectors based on operator defined mapping
- The **CollectorResource** is a proxy to the actual collectors for file assignment and status monitoring



IUM Components - SessionServer

- The session server is the main component that implements a Real-Time Charging Manager.
- The session server is primarily a container that holds one or more **connectors**, one or more **rule chains**, and one or more **session stores**
- You can configure as many session servers as you need, typically at least one for each protocol. In a highly available environment, you might have a second standby session server for each primary session server.

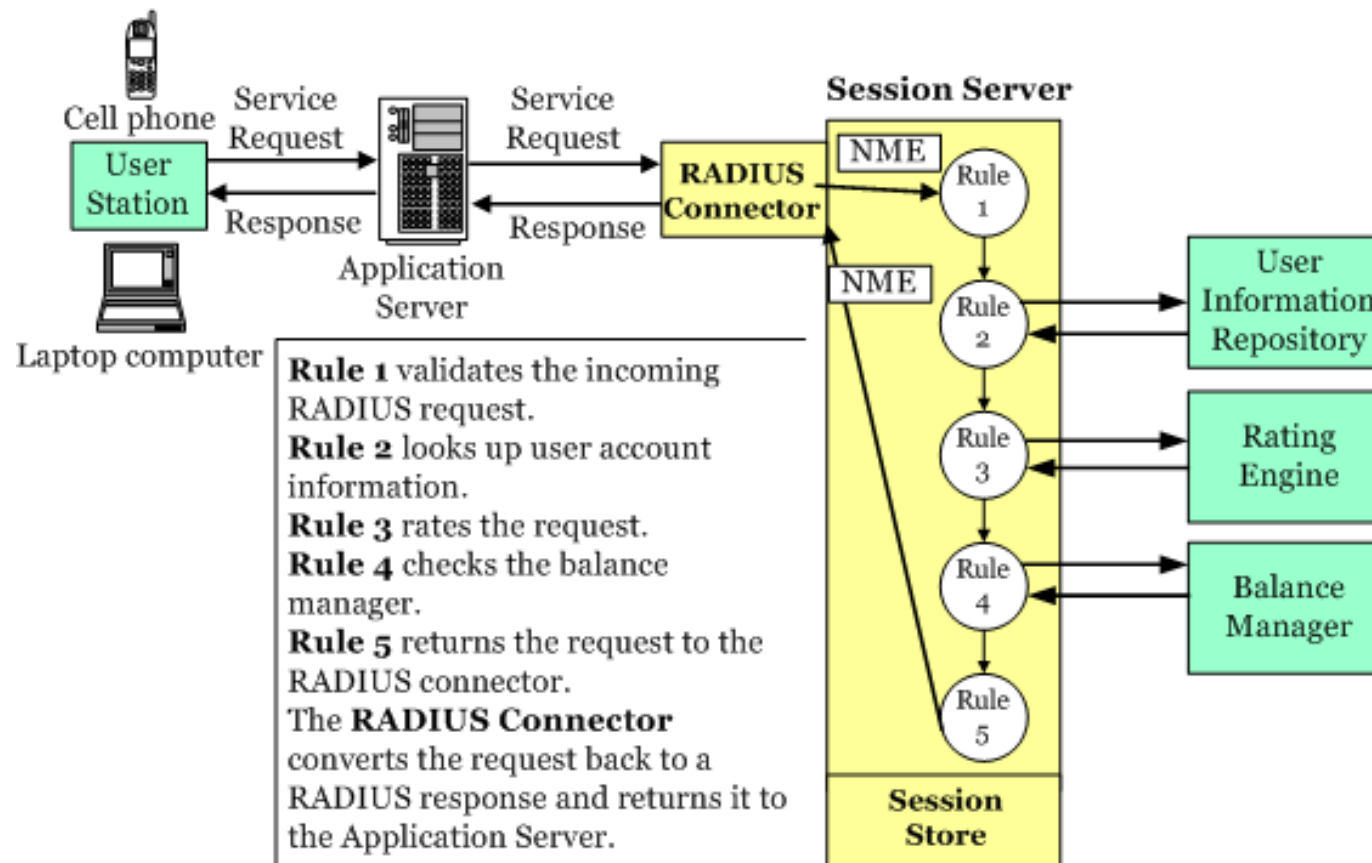
IUM Components - SessionServer

- Session Servers are built with the following components
 - **Connectors** and **Adapters** receive requests for authorization from clients and provide responses back to those clients. They are two way: data flows into the connector and back out to the requesting client
 - **Rule chains** process the incoming requests and implement your business logic, possibly querying other applications. They are multi-threaded for concurrency and real-time performance
 - **Session Stores** hold information about your subscribers to facilitate service authorization, account management and billing.
 - **NME Stores** can hold NMEs for processing by an IUM collector or another third party tool

IUM Components - SessionServer

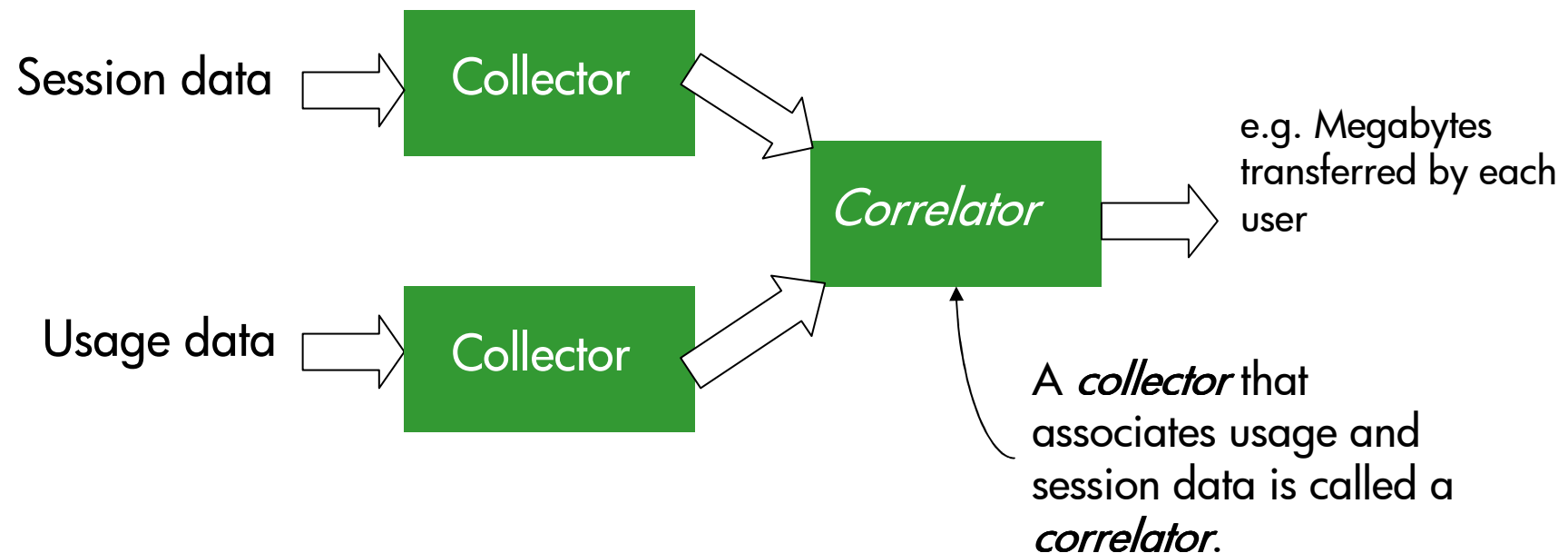
- Session Servers do not contain Encapsulators → Connectors
- Session Servers do not contain Datastores → Session and NME Stores
- Both Collectors and Session Servers have Rule Chains
 - In Collectors, Rule Chains are invoked with each input NME
 - In Session Servers, Rule Chains are invoked by the Connectors depending on the incoming request
- Collectors store NMEs in the Aggregation tree and eventually flush data to the datastore
- Session Servers rules add and query information to the session store and typically returns a response to the connector
- NMEStoreRule can store NMEs in the NMEStore for later processing
- Match/Aggregation Rules are not available for Session Servers

IUM Components - SessionServer



Correlation & Aggregation

- Collectors can be chained together to perform multiple levels of data reduction



Correlation & Aggregation

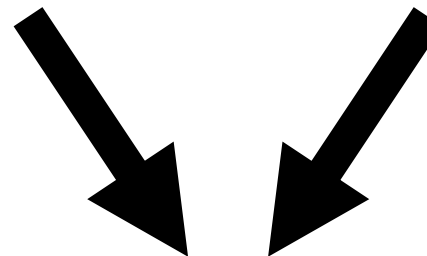
- Example of correlated data

Usage Data NME

SrcIP	DstIP	Num bytes	Start Time	End Time	Src Port	Dst Port
-------	-------	--------------	---------------	-------------	-------------	-------------

Session Data NME

Start Time	End Time	SrcIP	Login Id	Acct Num	Acct Status
---------------	-------------	-------	-------------	-------------	----------------



Correlated NME

SrcIP	DstIP	Numbytes	Start Time	End Time	Login	AcctNum
-------	-------	----------	---------------	-------------	-------	---------

Correlation & Aggregation

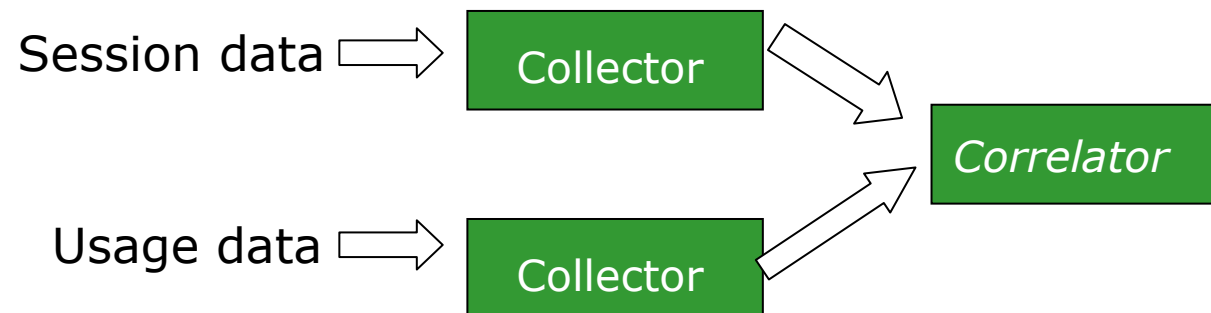
- Sessions are processed by **Session Collectors**
- The Datastore must be Database Type
 - It is tracking both active & inactive sessions and queries are not only over EndTime
- It must include StartTime as well as EndTime
- You should use the `–session` option with `siuquery` to retrieve information regarding active (vs. closed) sessions

Correlation & Aggregation

- The purpose of session Aggregation is to identify session starts and stops and associated IP addresses
- The single output NME would combine matching raw events such that:
 - SrcIP matches
 - LoginId matches
 - A StartTime is created from the EndTime
- Rules Hierarchy
 - Categorize on SrcIP and create a “bucket” (NMEGroup)
 - Categorize on LoginId and create a “leaf” (NME)
 - Adorn, the session must have both Start and End Time
 - Aggregate

Correlation & Aggregation

- The component that associates usage and session data from different collectors is called Correlation Collector
- It is a specific type of collector that involves a specific type of aggregation
- The first Rule Chain sorts session NMEs, the second Rule Chain locates the appropriate session in the Tree for the Inbound usage NME
- The Encapsulator is a PollingMuxEncapsulator



Correlation & Aggregation

Session Events

Src: 1.2.3.4
Time: 12:15-1:45
User: Joe

Src: 1.2.3.4
Time: 2:20-3:00
User: Bob

Src: 2.3.4.5
Time: 11:19-5:20
User: Sue

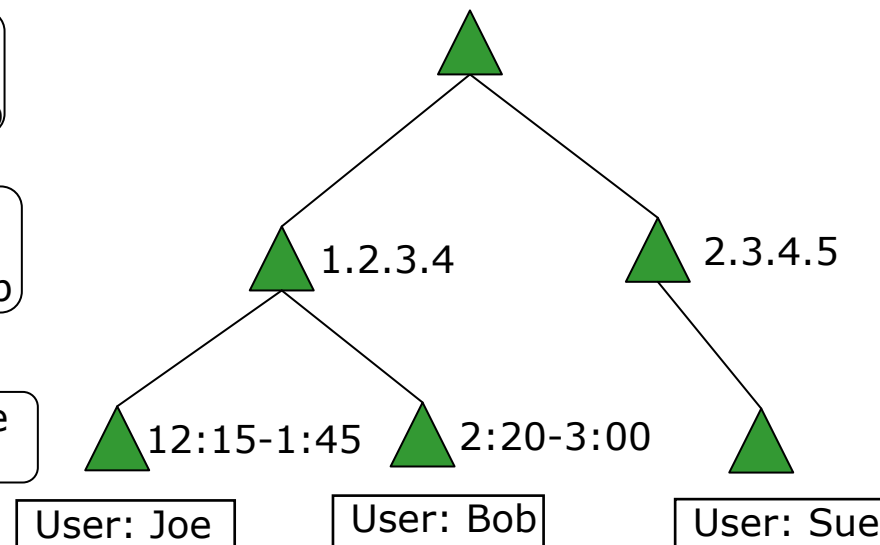
Scheme RuleChain

MatchRule
Match: Src IP
Next: NMEGroup

MatchRule
Match: Time
Next: NMEGroup

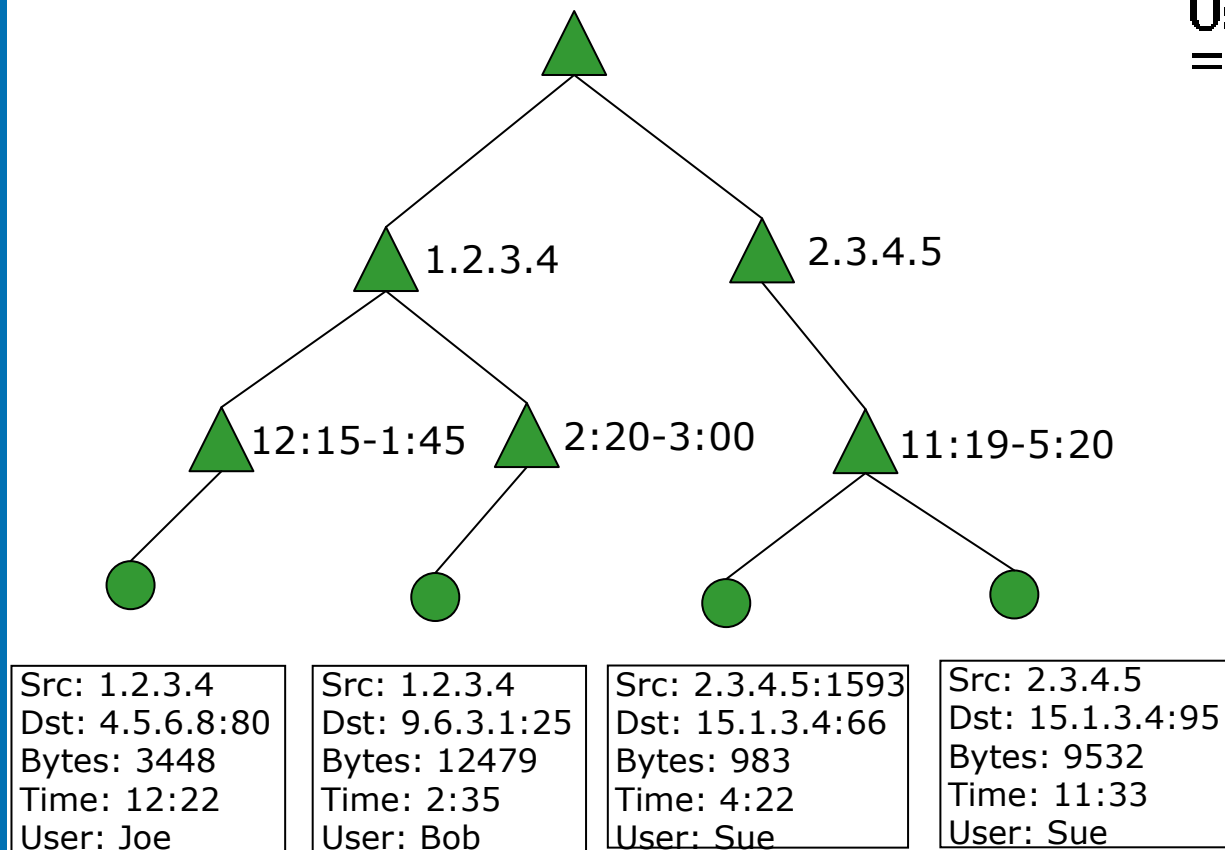
AggregationRule
Copy: User

Aggregation Tree

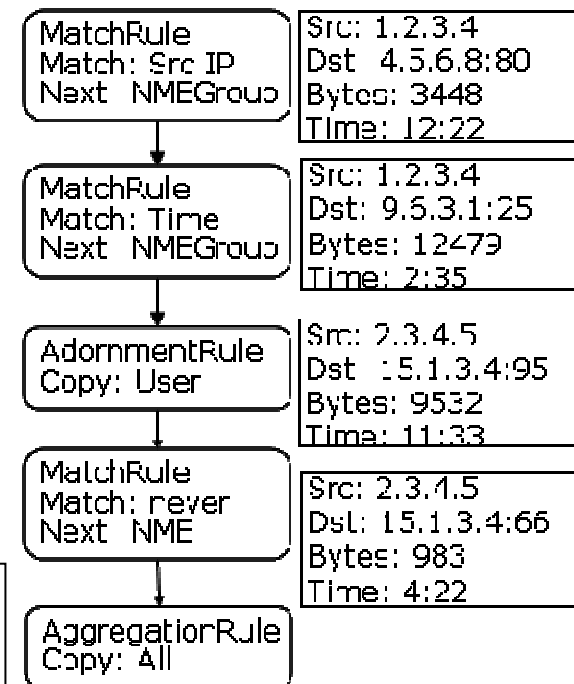


Correlation & Aggregation

Aggregation Tree

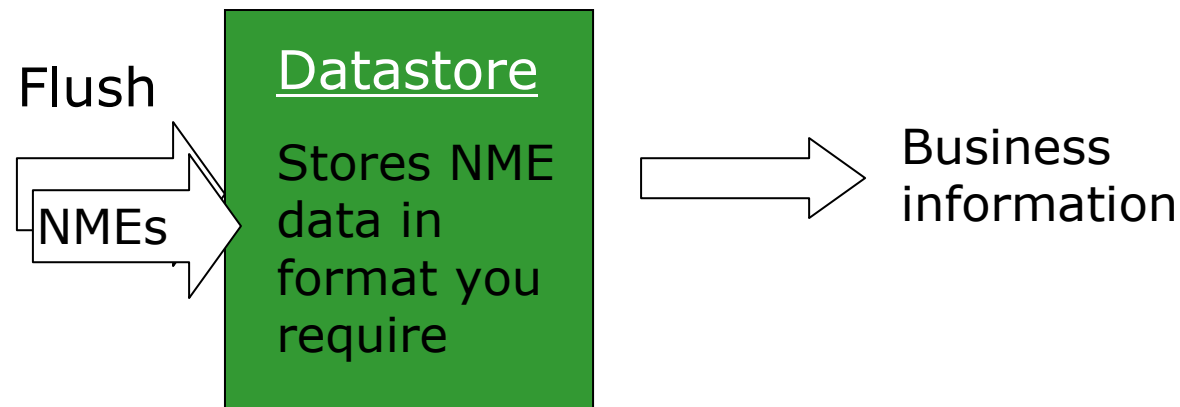


Usage Scheme = RuleChain Usage Events



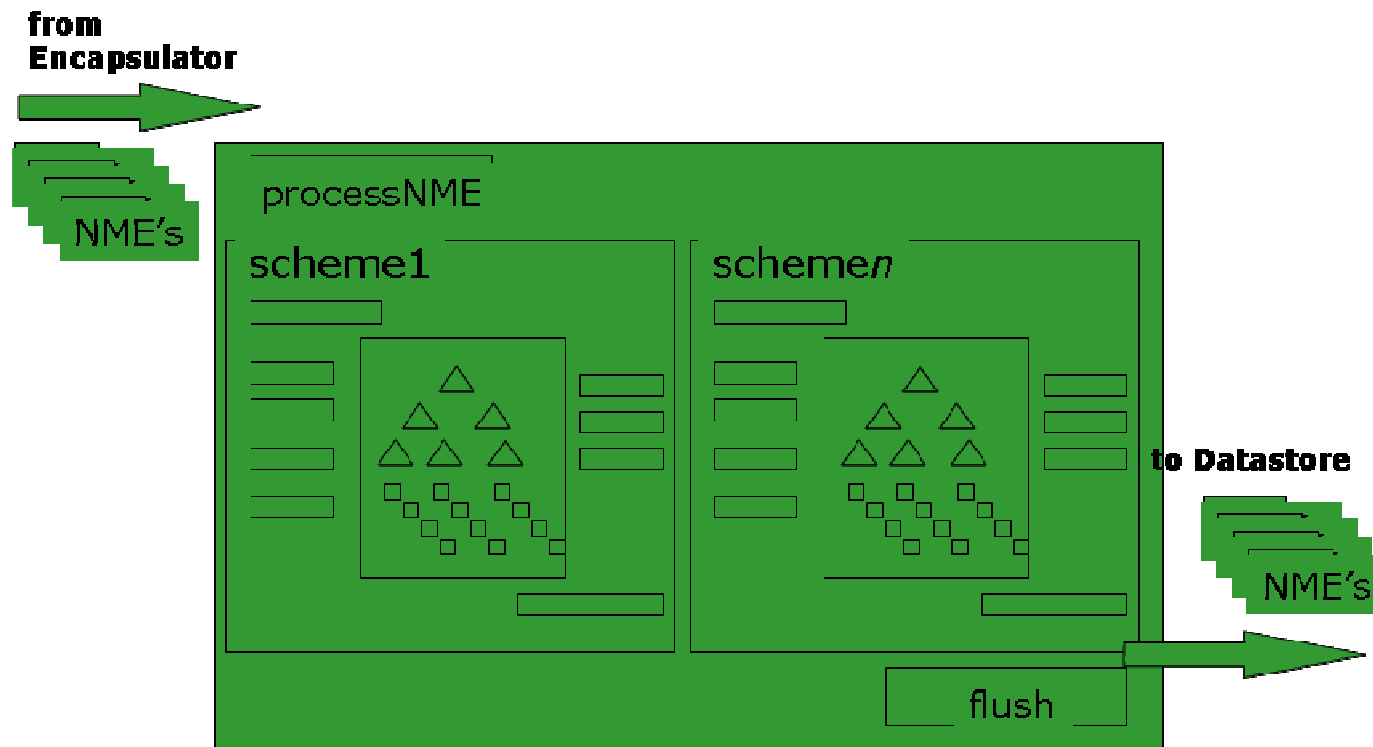
Correlation & Aggregation

- The type of Datastore that is commonly used with Correlation is the Database Type to allow queries from upstream applications or reporting tools
- Initially, it is recommended to use an IDR+ format for ease in debugging



Correlation & Aggregation

- Multi-scheme aggregation. An aggregator can have multiple different aggregation schemes
- NMEs are processed serially for each scheme



Correlation & Aggregation

- To configure multi-scheme aggregation, you should add an aggregation scheme for each desired report
- Example:
 - TopUsers
 - MatchRule on loginID
 - Aggreate
 - TopDestinations
 - MatchRule on dstIP
 - Aggregate
- Different schemes can be mapped to different DataStores