Advanced IUM Understanding – Session 1.2

- Presenter:
- Date:
- Duration:

- Ujjwal Barman
- June,2009
- 1 days

Agenda

- Basic Code Understanding
- Encapsulator,Aggregator,Datastore
- Commands Discussion.

- Working with the collectors
- Creating Collectors
- How to Customize Collectors
- How to Create a Collector Using Config Files
- The Deployment Editor in LaunchPad, Linked Collectors, Collector Creation Model, Editing the NME Schema
- How to Validate a Collector
- Types of Encapsulator, LogFileEncapsulator Overview
- Aggregator Architecture, Types of Rules, Simple Rule Scheme

Basic Code Understanding

Important Coding Tips:

- We should always follow the naming conventions. All collector configurations and NMESchemas should be named as .config file.
- Once the collector is developed use launchpad/command prompt to load the configuration and its associated NMESchema file into config server.
- Also place the associated map files and scripts in the correct location. Create the input directory as per the configuration.
- Make sure config server, adminagent , Solid DB is running.
- Start the collector using launchpad/Command prompt.

Note: Before starting the collectors all are requested to go through the command reference document to get acquainted with the commands.

Basic Code Understanding Continued…

- Please go through the following demo collector code and demo NMESchema.
- 1) Collector code.

demo.config

2) NME Schema

demo_NMESchema.config

Commands Discussion

Command Reference

Internet Usage Manager 4.5

Working with Collectors

- You can select multiple collectors at the same time. For example, if you want to stop all the collectors at once. In the right-hand pane, use your mouse to draw a box around all the collectors you want to select.

- When making lots of changes to collector configurations, cleanup the collector before re-starting. Cleanup deletes all the data and log files for a collector, as if the collector had never been run. Beware though of using cleanup in a production environment!

- When you have a chain of collectors, do not start all the collectors at once. Start the Level 1 (Leaf) collectors first, make sure they are running and producing data.Next start the Level 2 collectors and so forth.

Working with Collectors continued..

If you don't use Launchpad, you can do everything via the command line.
For example, here are the commands for managing collectors. They can be found in C:\Siu\bin or /opt/SIU/bin:

1. To start a Collector:
```
siucontrol –n <collector_name> -c startProc
```

2. To stop a Collector:
```
siucontrol –n <collector_name> -c stopProc
```

3. To cleanup a Collector:
```
siucontrol –n <collector_name> -c cleanProc
```

4. To save a Collector to a file:
```
saveconfig –n <collector_name> -f <filename>
```

5. To load a Collector from a file:
```
loadconfig –merge –f <file_name>
```

Note: Commands to start and stop IUM on UNIX:
```
/sbin/init.d/siu/start
/sbin/init.d/siu/stop
```
All details can be found in the Tools manual.

Working with Collectors continued..

- We can have more then one instance of IUM running under one deployment. If we take the earlier code as reference, deployment is the name of the Deployment and ium1 was the name of one instance. We can also install one more instance of ium named ium2,3…etc under the same deployment.
- Each instance can have multiple collectors under then deployed. Even one whole production system can be defined under one instance. So in scenarios where we need to have duplicate copies of production for load balancing, we can do it under one deployment having multiple instances for multiple production copies.

Working with Collectors Continued …

- How to invoke a launchpad of a remote system (can be of a production system)

Each ium deployment will have its specified IOR path. This is a path or address which is understood by IUM and it helps us to invoke that IUM instance. So we can just put that path in the command prompt and can remotely open any IUM instance.

Example:

```
c:\SIU\bin\SIUJavaw -Xmx456m
    com.hp.siu.gui.launchpad.LaunchPad -iorURL http://<ip
    address>:<port number>
```

Note:

To do this we should have IUM installed in the local system because using the local instance , the remote IOR path will be invoked.

For local running of launchpad, please use the launchpad Icon.

Creating Collectors

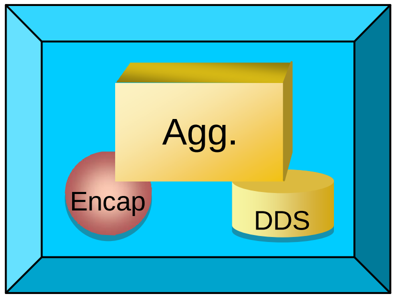
• The 3 components of a collector – Encapsulator, Aggregator, Datastore are independent objects.

• There are MANY different types of Encapsulators, Aggregator Rules and Datastores.

• One type of Encapsulator can be exchanged for a different type within a collector and so can an Aggregator and a Datastore.

• To understand what type of Encapsulator, Aggregator and Datastore is inside a collector, look at the CLASSNAME attribute inside it. This is the java class or object name.
For example, ClassName=LogFileEncapsulator is a type of Encapsulator
that is programmed to read log files.

• ALL the many different types of Encapsulators, Aggregators and   Datastores are defined in the Component Reference manual.

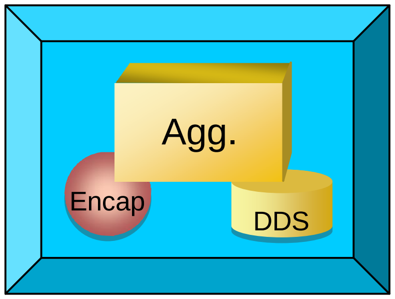


Creating Collectors (continued)

- Inside each Encapsulator, Aggregator, and Datastore are sub-objects with their own Classnames.

  For example, a Record Encapsulator has the sub-objects Record Factory, Stream Source and File Roll Policy. Each can be of different types and have their own classnames to identify them.
  In the Aggregator, each rule is its own sub-object with its own classname associated with it. Rules can be added, modified and deleted and moved around as independent objects.

- As before ALL the many different types of sub-objects in the Encapsulators, Aggregators and Datastores are defined in the Component Reference manual.



How to Create Collectors

There are three methods to create Collectors:

1. Create collectors from Templates and modify them in LaunchPad. This method is best for beginners.

2. Edit Config Files and import them into LaunchPad. This method is more advanced and alot more flexible. The sooner you learn this method, the better.

3. Use the Deployment Editor in LaunchPad. This method is also advanced and very useful.

Collector Templates

- A Collector Template is a set of objects (Encapsulator,
- Aggregator, Datastore and their sub-objects) that HP has decided
- to group together as a collector, to serve a specific purpose.

- Each template has a name to reflect its purpose.

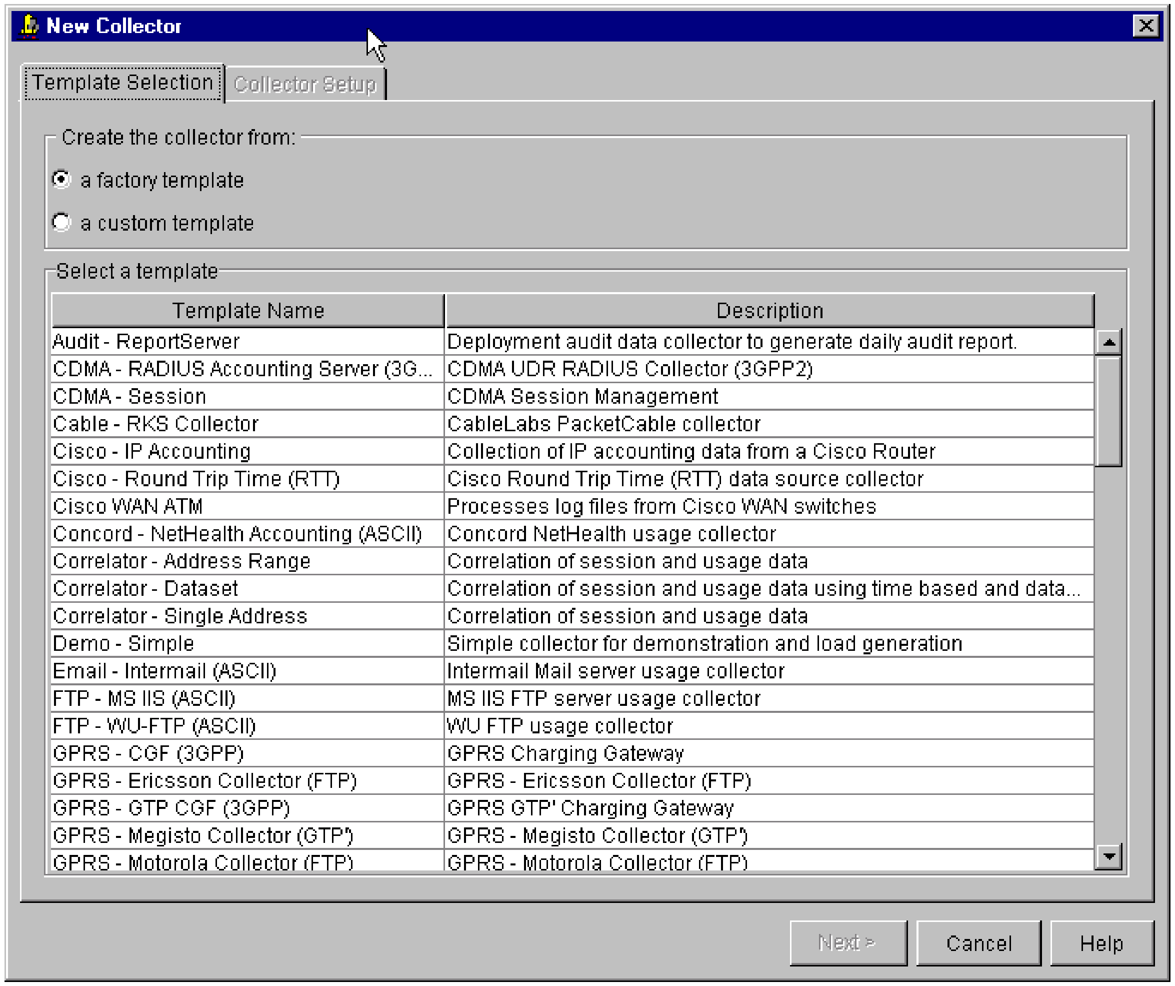
- Templates can be easily re-used to create similar collectors.

- In launchpad when you “create a collector” you are instantiating
- a copy of a template. Typically then you need to make some
- modifications to it to exactly fit your needs.

- You can also create your own templates.

Collector Templates (continued)

- Choose a Template that most closely reflects your objective. It automatically generates an Encapsulator, Aggregator and Datastore.
- Customize the Collector created from the Template to reflect your specific requirements.



Create the collector from:

(● a factory template) ○ a custom template

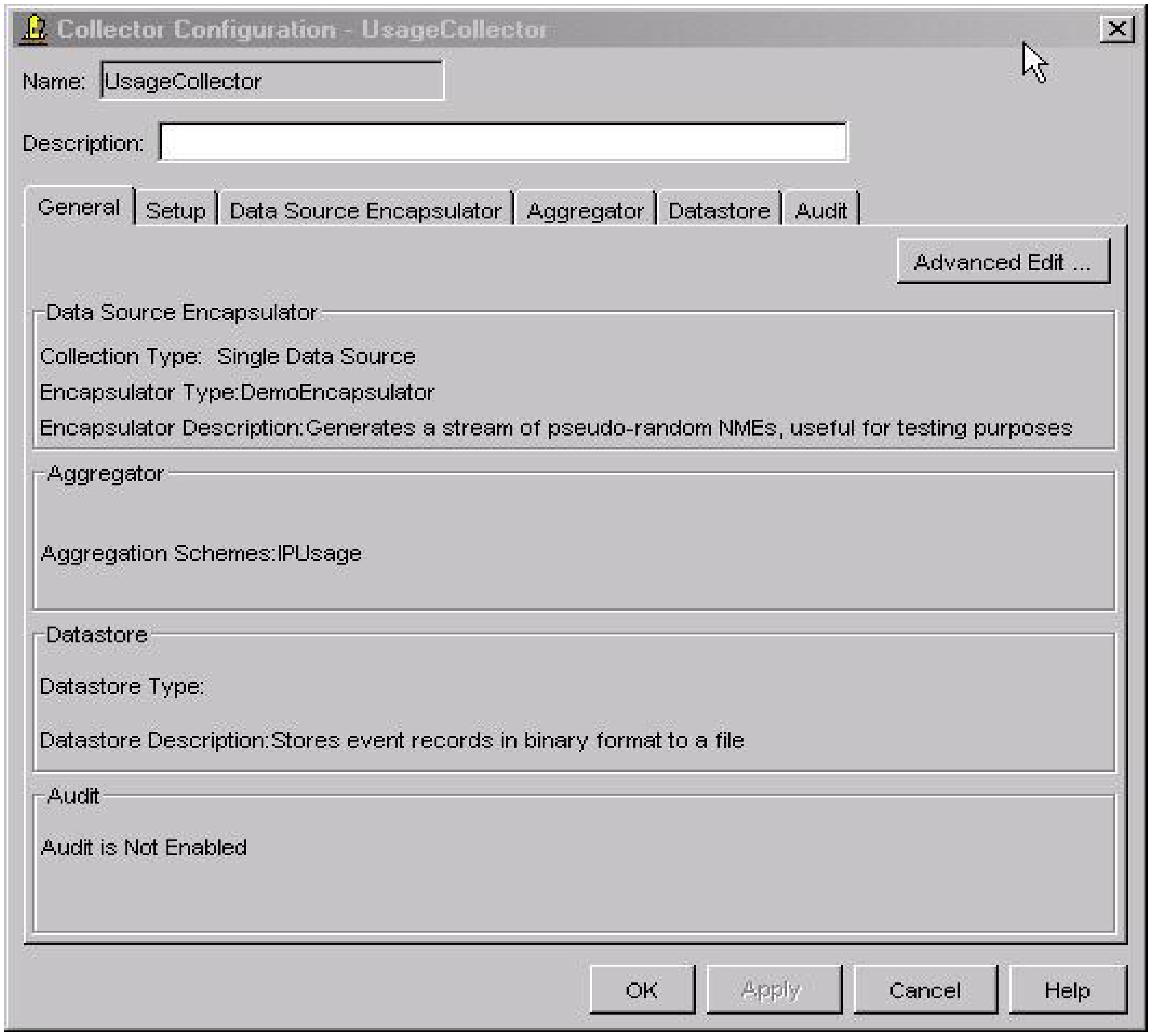
Select a template

Template Name	Description
Audit - ReportServer	Deployment audit data collector to generate daily audit report.
CDMA - RADIUS Accounting Server (3G...	CDMA UDR RADIUS Collector (3GPP2)
CDMA - Session	CDMA Session Management
Cable - RKS Collector	CableLabs PacketCable collector
Cisco - IP Accounting	Collection of IP accounting data from a Cisco Router
Cisco - Round Trip Time (RTT)	Cisco Round Trip Time (RTT) data source collector
Cisco WAN ATM	Processes log files from Cisco WAN switches
Concord - NetHealth Accounting (ASCII)	Concord NetHealth usage collector
Correlator - Address Range	Correlation of session and usage data
Correlator - Dataset	Correlation of session and usage data using time based and data...
Correlator - Single Address	Correlation of session and usage data
Demo - Simple	Simple collector for demonstration and load generation
Email - Intermail (ASCII)	Intermail Mail server usage collector
FTP - MS IIS (ASCII)	MS IIS FTP server usage collector
FTP - WU-FTP (ASCII)	WU FTP usage collector
GPRS - CGF (3GPP)	GPRS Charging Gateway
GPRS - Ericsson Collector (FTP)	GPRS - Ericsson Collector (FTP)
GPRS - GTP CGF (3GPP)	GPRS GTP' Charging Gateway
GPRS - Megisto Collector (GTP')	GPRS - Megisto Collector (GTP')
GPRS - Motorola Collector (FTP)	GPRS - Motorola Collector (FTP)

How to Customize Collectors

- In Launchpad, highlight the collector, right-click and select “Edit”. This opens the Collector Configuration window.

- You can select the Tab to customize the Encapsulator, Aggregator and Datastore.



Name: UsageCollector

Description:

Data Source Encapsulator

Collection Type: Single Data Source

Encapsulator Type:DemoEncapsulator

Encapsulator Description:Generates a stream of pseudo-random NMEs, useful for testing purposes

Aggregator

Aggregation Schemes:IPUsage

Datastore

Datastore Type:

Datastore Description:Stores event records in binary format to a file

Audit

Audit is Not Enabled

How to Create a Collector Using Config Files

- Config Files are ASCII files that define a collectors configuration.
- They contain sections for the Encapsulator, Aggregator and Datastore.
- The whole structure is hierarchical, within each section are subsections that contain configuration values.
- They are very flexible but beware of typing mistakes. You can cut-and-paste sections from one file to another, email your file to your support team, etc.
- You can also export a currently running collector to a file. So they can be used to back-up a collectors configuration.

How to Create a Collector Using Config Files (continued)

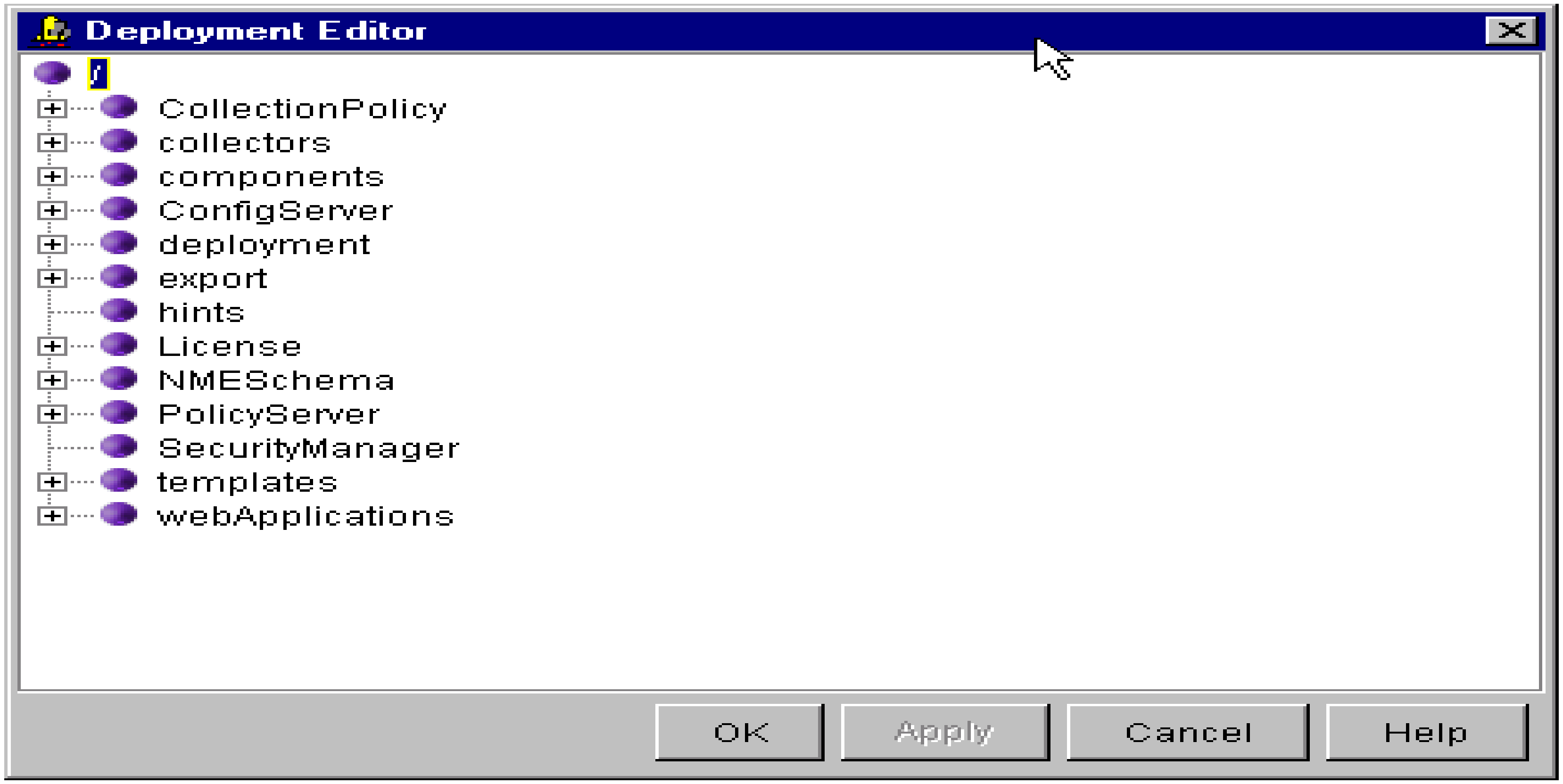
1. In Launchpad use the File → Export Configuration option or the saveconfig command to save a collector's configuration to a config file. Either in /opt/SIU/bin or C:\Siu\bin:
   saveconfig –n <collector name> -f <config file name>

2. Edit the Config File using any text editor. Save the file.

3. In Launchpad use the File → Import Configuration option or the loadconfig command to load a collector's configuration from a config file. Either in /opt/SIU/bin or C:\Siu\bin:
   loadconfig –f <config file name> [-merge]

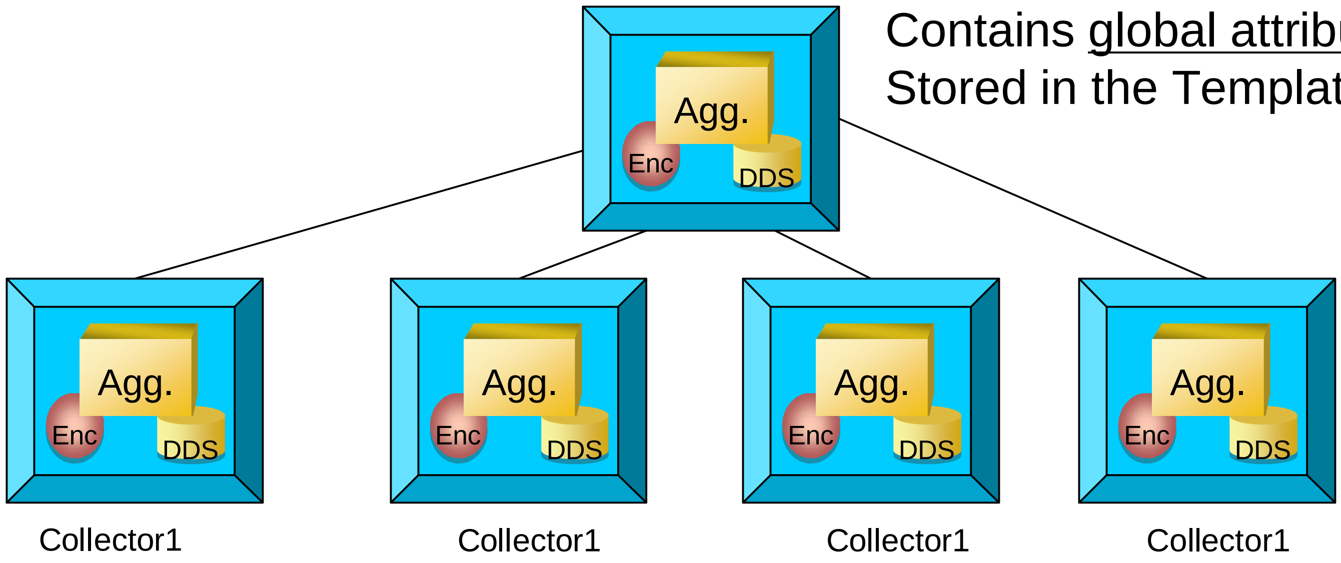
The Deployment Editor in LaunchPad

- The Deployment Editor gives a global static view of an IUM installation.
- Templates section - list of all the available Templates.
- Deployment section - all the servers and collectors you have configured.
- You can **copy** a collector from one machine to another, or **cut-and-paste collectors** or **components of collectors**.



Linked Collectors

- What if you had to create a large number of collectors that were nearly
- all the same? For example, you have 40 switches on your network and
- need one collector for each switch. The configuration for the collector's
- Encapsulators, Aggregators and Datastores would be nearly identical,
- the only difference is the switch id you are connecting to using ftp to read
- the data.
- Instead of having to create 40 collectors from scratch and then have to
- edit them all individually when you have to make changes, you can use
- Linked Collectors.



Master Collector
Contains global attributes
Stored in the Templates Section

Linked Collectors
Contain only over-ridden attributes specific to that collector. Managed in launchpad or via command line like any other collectors

Linked Collectors (continued)

How to create Master and Linked Collectors:

1. Create the Master collector first. Configure the Encapsulator, Aggregator and Datastore with all the global attributes. Save this collector as a Custom Template in Launchpad or as a config file.

2. Create a new config file for each Linked Collector. In the Config File create only the components and/or subcomponents that are different from the default attributes in the Master.

3. At the top level in the file add the line Link to link to the Master Collector in the Templates Section, for example:

```
[/deployment/hostname/collectorname]
AdminInterface=
ClassName=com.hp.siu.adminagent.procmgr.CollectorProcess
Description=
Link=/templates/custom/mastercollectorname
QueryInterface=
```

NOTE: Do not save your Custom Template under the Factory section as they may be over written!

Linked Collectors (continued 2)

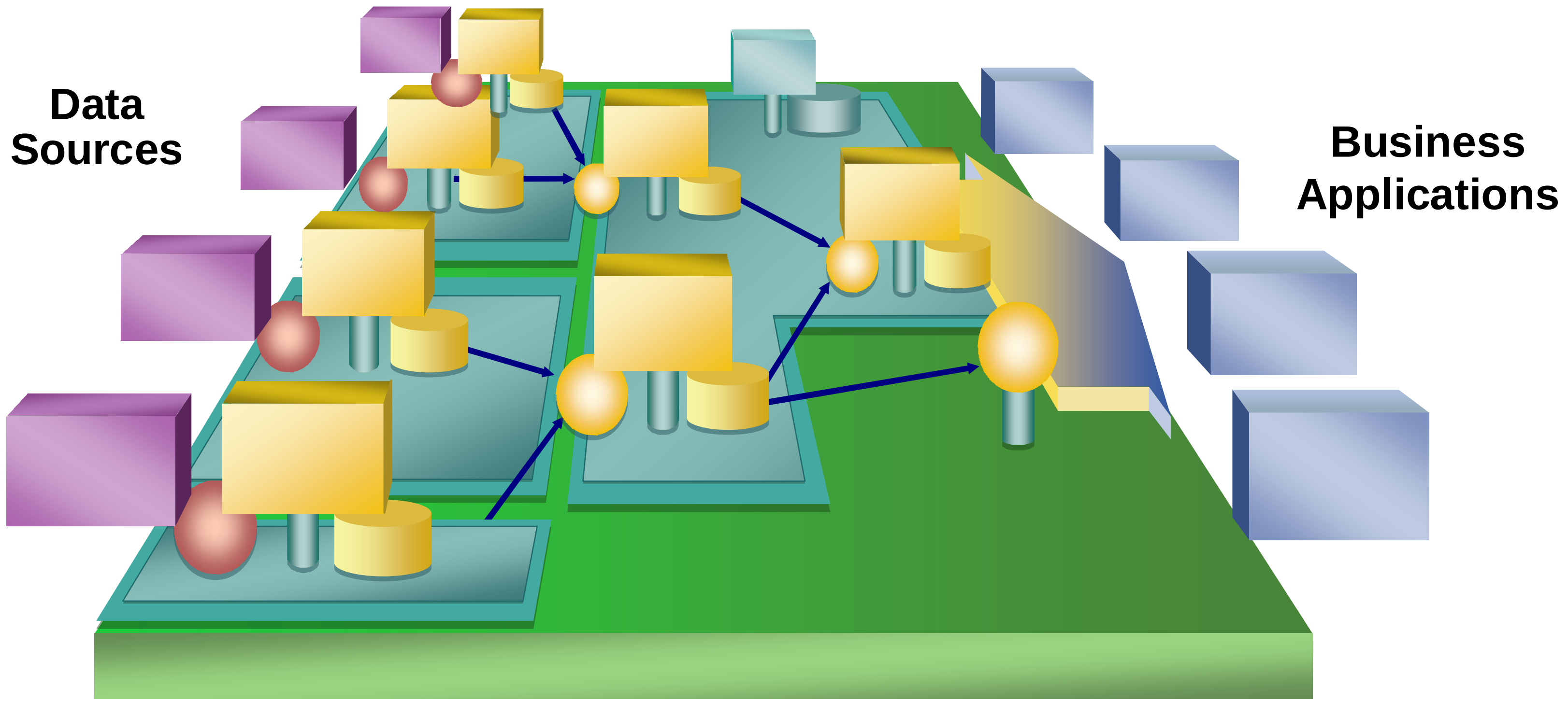
How to create Master and Linked Collectors, continued:

4. Save the file and load / import the collector. This will load /import all the default or global attributes from the Master into this collector. You can now start, stop and manage that collector like any collector.

When you want to make changes across all the collectors, edit the Master collector. Stop and restart the Linked Collectors and they will automatically be updated.

If you want to make changes to any individual collector, just edit it as normal using Launchpad or config file.

Collector Creation Model



For whatever type of Collectors you are creating, remember Collectors are building blocks:

1. build the Encapsulator, test and modify.
2. build the Aggregator, test and modify.
3. build the Datastore, test and modify.

Do this for each collector one-by-one before moving onto the next.

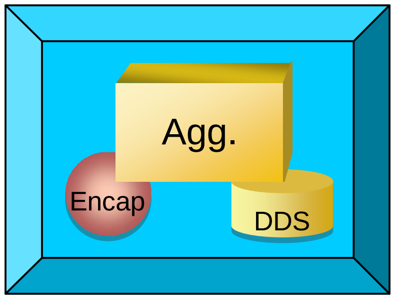
Collector Creation Model (continued)

Tips to help you build and test collector components:

1. Encapsulator – you will need some real input data or simulated input data. Configure the encapsulator and its subcomponents. To test, add just a Store Rule in the Aggregator and a simple Datastore, for example, the IDR+ datastore with text or html output. Or if lots of data, use the JDBCDatastore and query it.

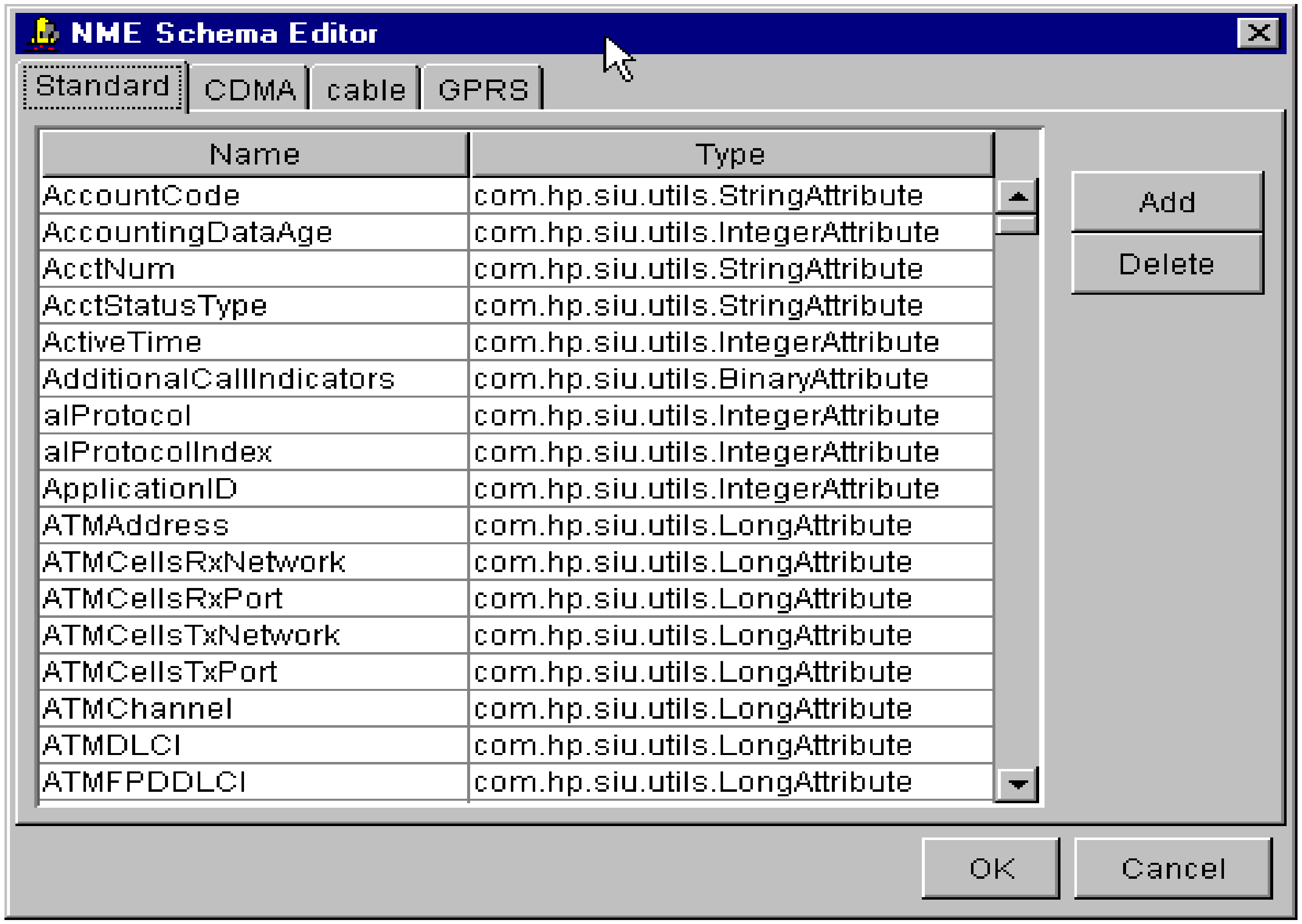
2. Aggregator – use your new Encapsulator. Add one rule at a time. Use the WriteToFile rule to test the output of each rule. Use a simple datastore as for the Encapsulator.

3. Datastore – use your new working Encapsulator. Use just a StoreRule or your new Aggregation Scheme.



Editing the NME Schema

You can use the NME Schema Editor Tool in Launchpad to add any new NME Attributes.
Go to Launchpad Menu bar → Tools → NME Schema Editor



Name	Type
AccountCode	com.hp.siu.utils.StringAttribute
AccountingDataAge	com.hp.siu.utils.IntegerAttribute
AcctNum	com.hp.siu.utils.StringAttribute
AcctStatusType	com.hp.siu.utils.StringAttribute
ActiveTime	com.hp.siu.utils.IntegerAttribute
AdditionalCallIndicators	com.hp.siu.utils.BinaryAttribute
alProtocol	com.hp.siu.utils.IntegerAttribute
alProtocolIndex	com.hp.siu.utils.IntegerAttribute
ApplicationID	com.hp.siu.utils.IntegerAttribute
ATMAddress	com.hp.siu.utils.LongAttribute
ATMCellsRxNetwork	com.hp.siu.utils.LongAttribute
ATMCellsRxPort	com.hp.siu.utils.LongAttribute
ATMCellsTxNetwork	com.hp.siu.utils.LongAttribute
ATMCellsTxPort	com.hp.siu.utils.LongAttribute
ATMChannel	com.hp.siu.utils.LongAttribute
ATMDLCI	com.hp.siu.utils.LongAttribute
ATMFPDDLCI	com.hp.siu.utils.LongAttribute

Editing the NME Schema (continued)

You can edit the NME Schema using configuration files.
This method is easier if you have a large number of new NME Attributes to add.
For example, create a text file called new NMESchema.config and add the new attributes like this:

```
[/NMESchema]
Attributes=NASPortType,com.hp.siu.utils.IntegerAttribute
Attributes=AcctAuthentic,com.hp.siu.utils.StringAttribute
Attributes=TermCause,com.hp.siu.utils.IntegerAttribute
Attributes=ServiceType,com.hp.siu.utils.StringAttribute
Attributes=AcctSessionTime,com.hp.siu.utils.IntegerAttribute
Attributes=AcctDelayTime,com.hp.siu.utils.IntegerAttribute
Attributes=PseudoReqSrc,com.hp.siu.utils.IPAddrAttribute
Attributes=PseudoReqType,com.hp.siu.utils.StringAttribute
```

Use either the Import option in Launchpad or the command loadconfig to update the NME Schema using this file.

How to Validate a Collector

Don’t just assume that if a collector is running then its OK. You need to validate the collector.
Whichever method you use to create a Collector, you validate them in the same way:

1. Check the Collector Stats Tab in Launchpad or **siucontrol** command. Either in /opt/SIU/bin or C:\Siu\bin:

   siucontrol –n <collectorname> -c showStats
2. Check the Collector Log file Tab in Launchpad or <collectorname>.log in either /var/opt/SIU/log or C:\Siu\varlog.
3. Query the collector using the **Query Data** option in Launchpad or **siuquery** command.

   siuquery –n <collectorname> -scheme <schemename> (-session) –f <filename>.htm –html
4. Validate the output data. Is the data from your query approximately what you expect ?

Types of Encapsulators

Read data from network data sources (Level 1 Collectors):

- Record
- BERFile
- CiscoWAN
- IPDRv3
- IPAcct
- JDBC
- LogFile
- P-Cube
- SingleFile

- File
- HybridMux
- NortelUMTS
- Protocol
- SNMP

Stream/Packet

- DynamicSNMP
- Netflow
- RMON
- RttMIB
- SFlow
- UDP

NME

- Demo

Read data from collectors (Level 2+ Collectors):

- NME
- BatchCollector
- Collector
- Dataset
- DatasetMux
- PollingMux
- SimpleMux
- Report
- Dataset Report
- NMEFile

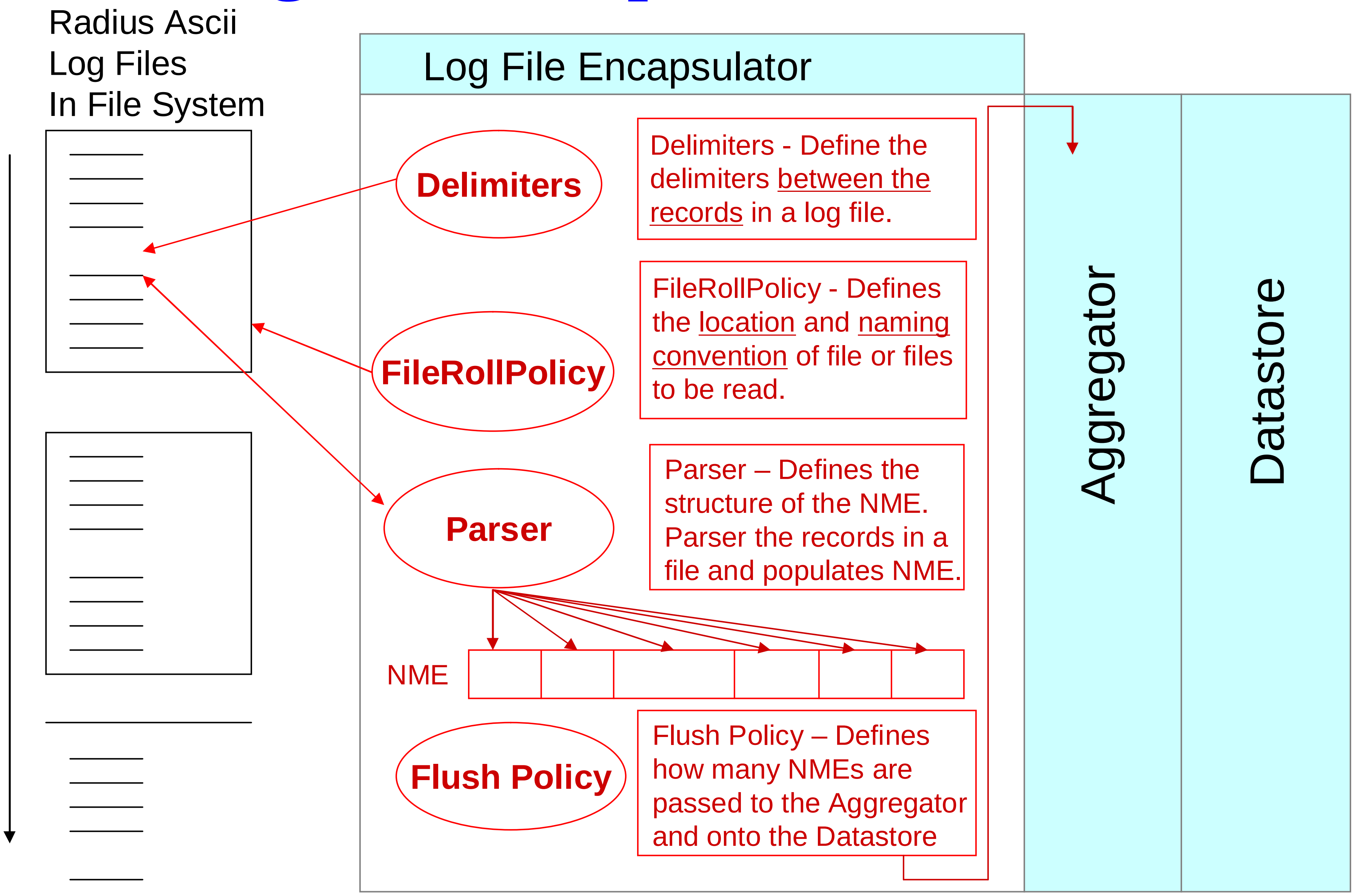
All Encapsulators and their attributes are documented in full detail in the Component Reference manual.

LogFileEncapsulator

Objective – Learn about the LogFileEncapsulator and some of its features:

- Its an encapsulator specifically designed to read log files = LogFileEncapsulator
- It reads delimiters between records in the log files = Delimiters
- It knows the location and name of files to read = FileRollPolicy
- It controls how much data is passed to the Aggregator = FlushPolicy
- It reads data from each record in the log file and creates NMEs = Parser

LogFileEncapsulator – Overview



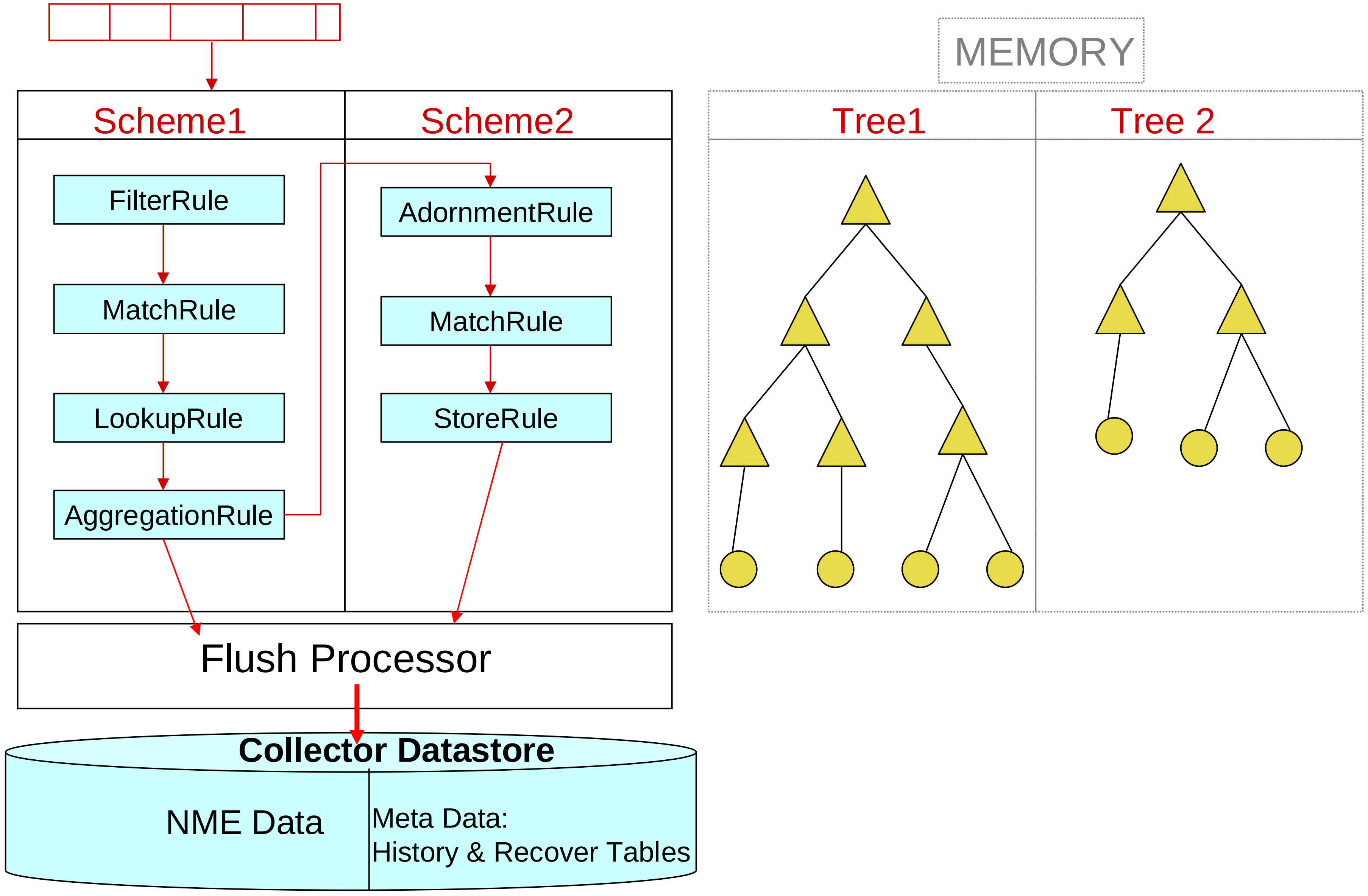
Delimiters - Define the delimiters between the records in a log file.

FileRollPolicy - Defines the location and naming convention of file or files to be read.

Parser – Defines the structure of the NME. Parser the records in a file and populates NME.

Flush Policy – Defines how many NMEs are passed to the Aggregator and onto the Datastore

Aggregator Architecture



Types of Rules

Matching:
HashMatchRule
SessionMatchRule
CorrelatorMatchRule
CorrelatorSessionMatchRule
CountUniqueMatchRule
HistogramMatchRule
RangeCorrelatorMatchRule
VectorMatchRule

All Rules are documented in full detail in the Component Reference manual.

Important and commonly-used Rules are highlighted in bold.

Modifying:
AdornmentRule
BinaryToIPRule
ConvertCaseRule
DNSLookupRule
DoubleToIntegerRule
DuplicateCDRDetectorRule
DuplicateNMEDetectorRule
IndexSplitRule
IPRangeRule
JDBCLookupRule
JDBCUpdateRule
JNDILookupRule
MultiLookupRule
PerlRegExStringRule
PercentileRule
ProxyURLRule
RMONIPProtocolMapRule
RttTypeMapRule
SequenceRule
SplitListAttributeRule
SplitNMERule

Modifying(continued):
SplitURLRule
SubstrRule
TimeAdornmentRule
TimeGapDetectionRule
TAP3Rule
UsageActivityLookupRule

Conditional:
FilterRule
ConditionalRule
ConditionRangeRule
SwitchConditionRule

Event related
PushRule
DuplicateNMEDetectorRule
CloseAllSessionsRule
LogNMERule
VOIPClossAllSessionsRule
WriteToFileRule

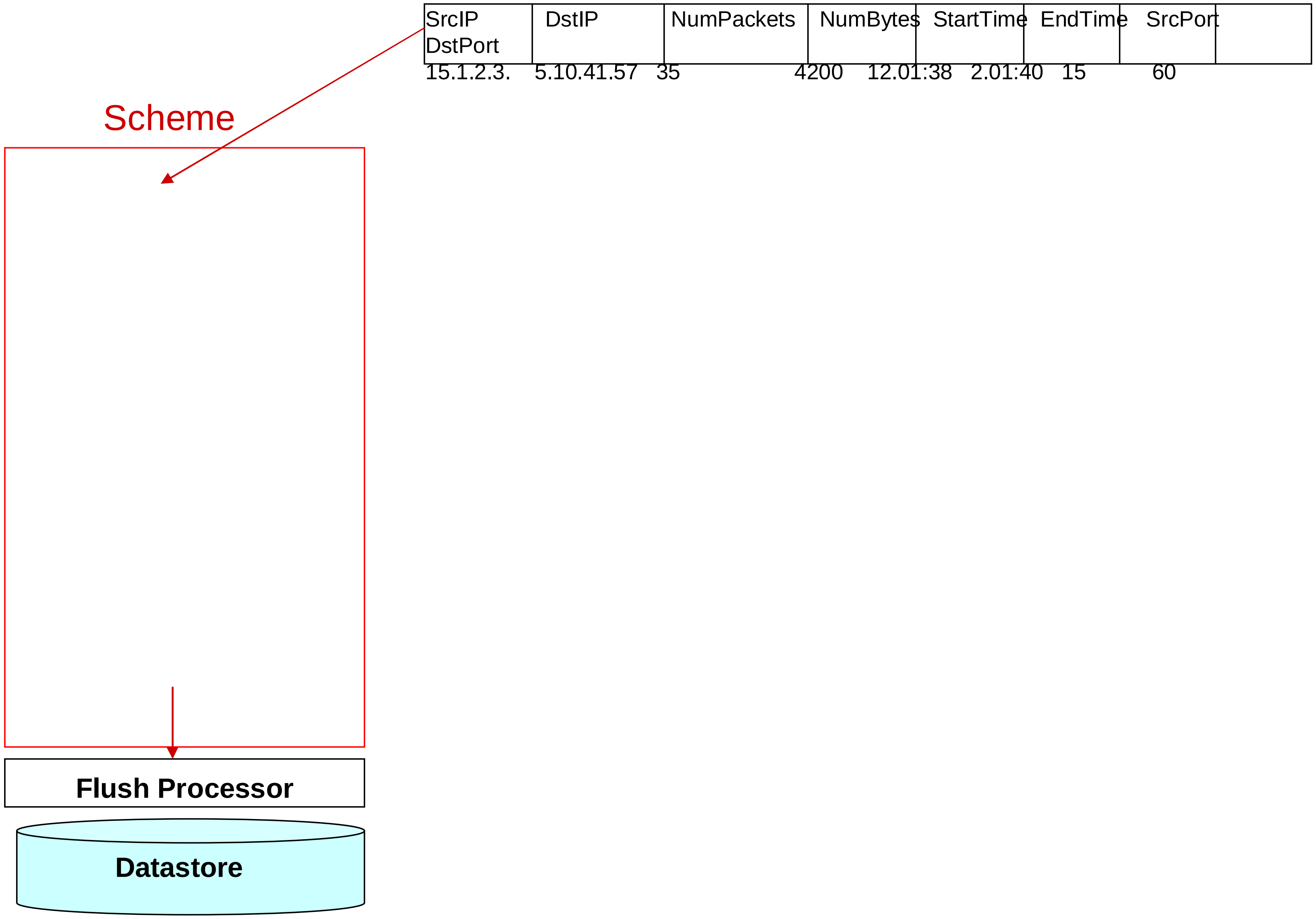
Storing
AggregationRule
StoreRule
SessionAggregationRule

Simple Rule Scheme

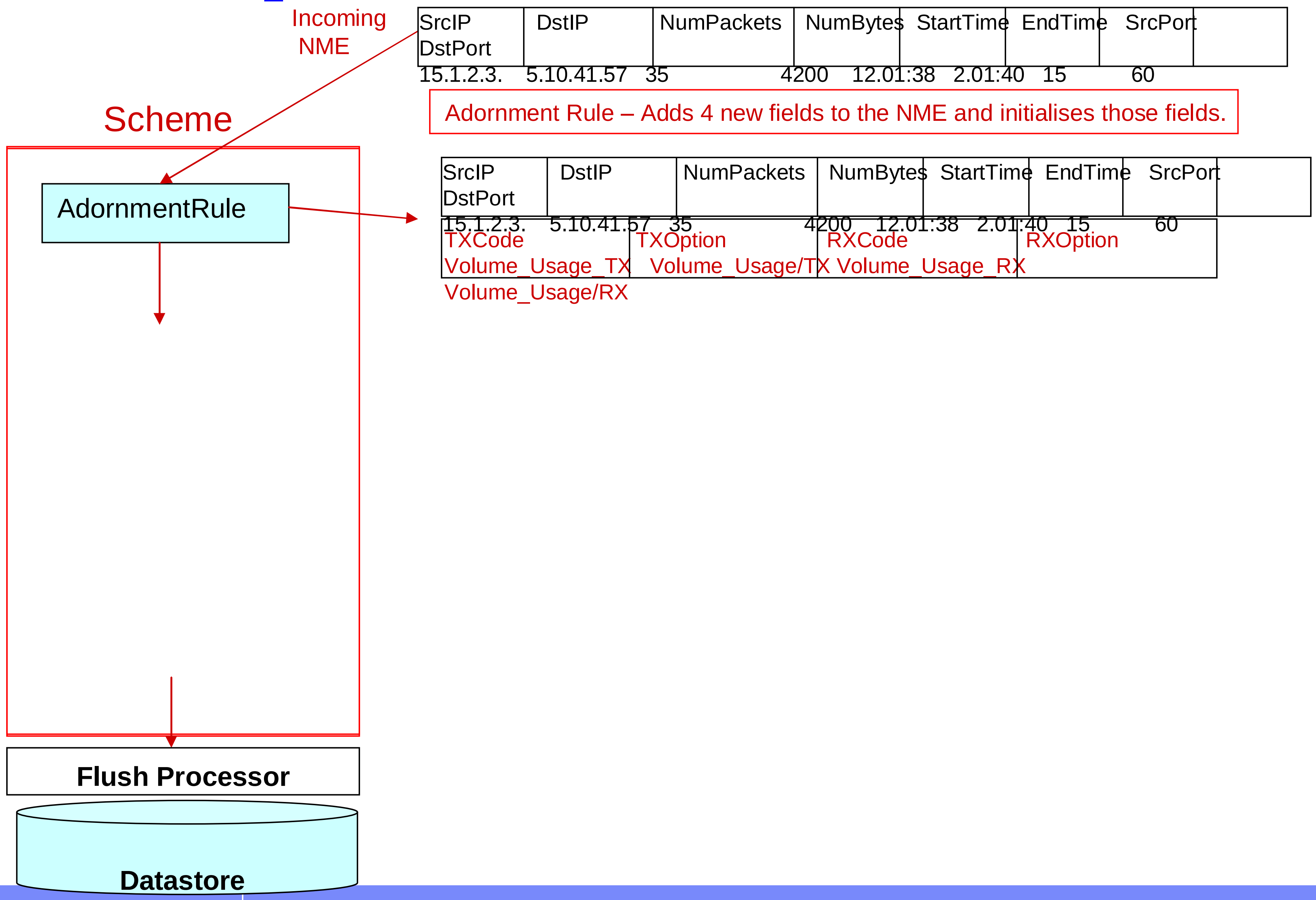
Objective – Learn about a simple Rule Scheme and some
Aggregator
features:

- How to add new rules to an existing Scheme
- How to match values based on a key = Hash Match Rule
- How to access external information = UsageActivityLookupRule
- How to add new data to the NMEs = Adornment Rule
- How to summarise and writes out results = Aggregation Rule

Simple Rule Scheme – Overview (1)



Simple Rule Scheme – Overview (2)



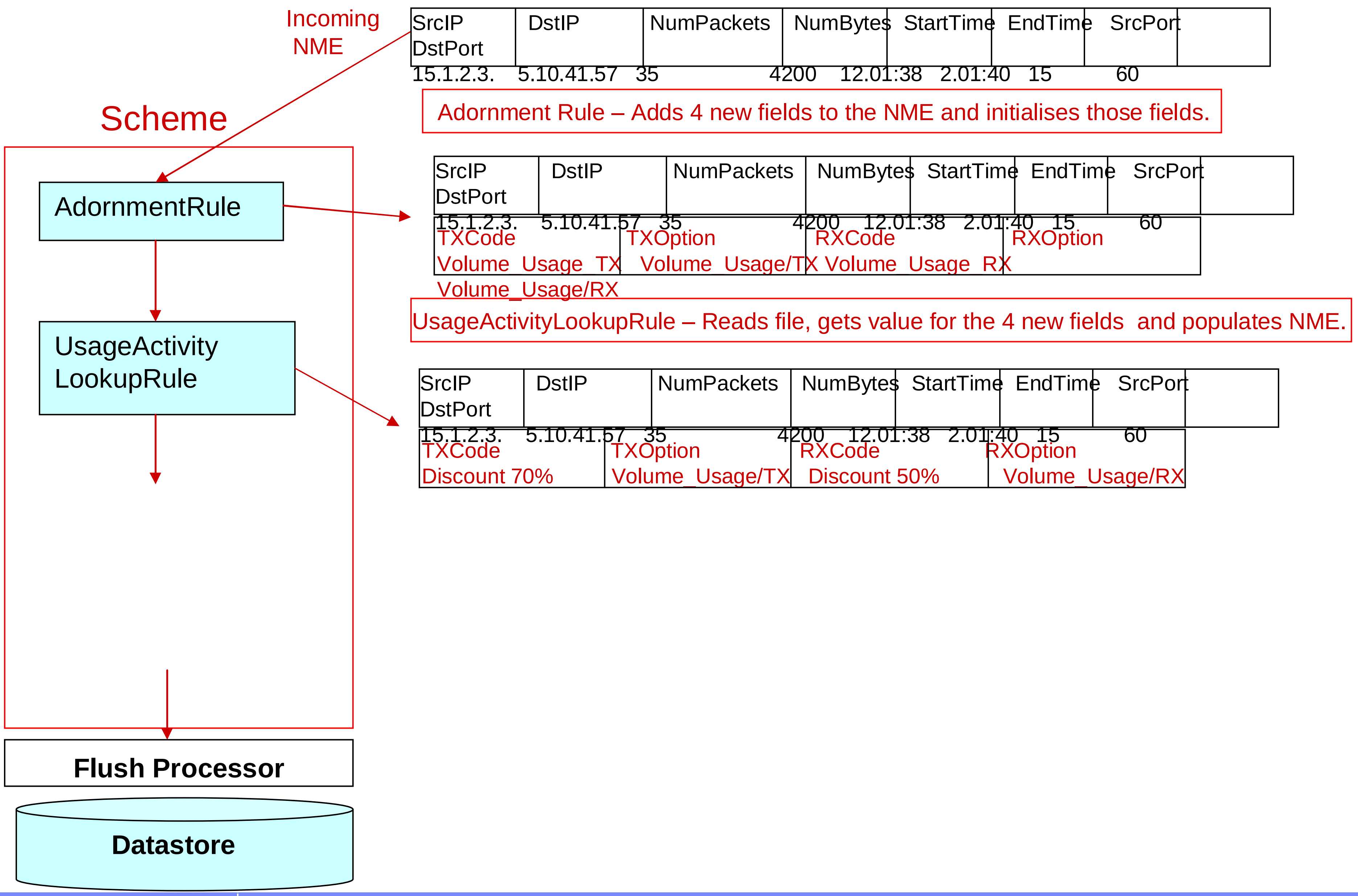
SrcIP DstPort	DstIP	NumPackets	NumBytes	StartTime	EndTime	SrcPort	
15.1.2.3.	5.10.41.57	35	4200	12.01:38	2.01:40	15	60

Adornment Rule – Adds 4 new fields to the NME and initialises those fields.

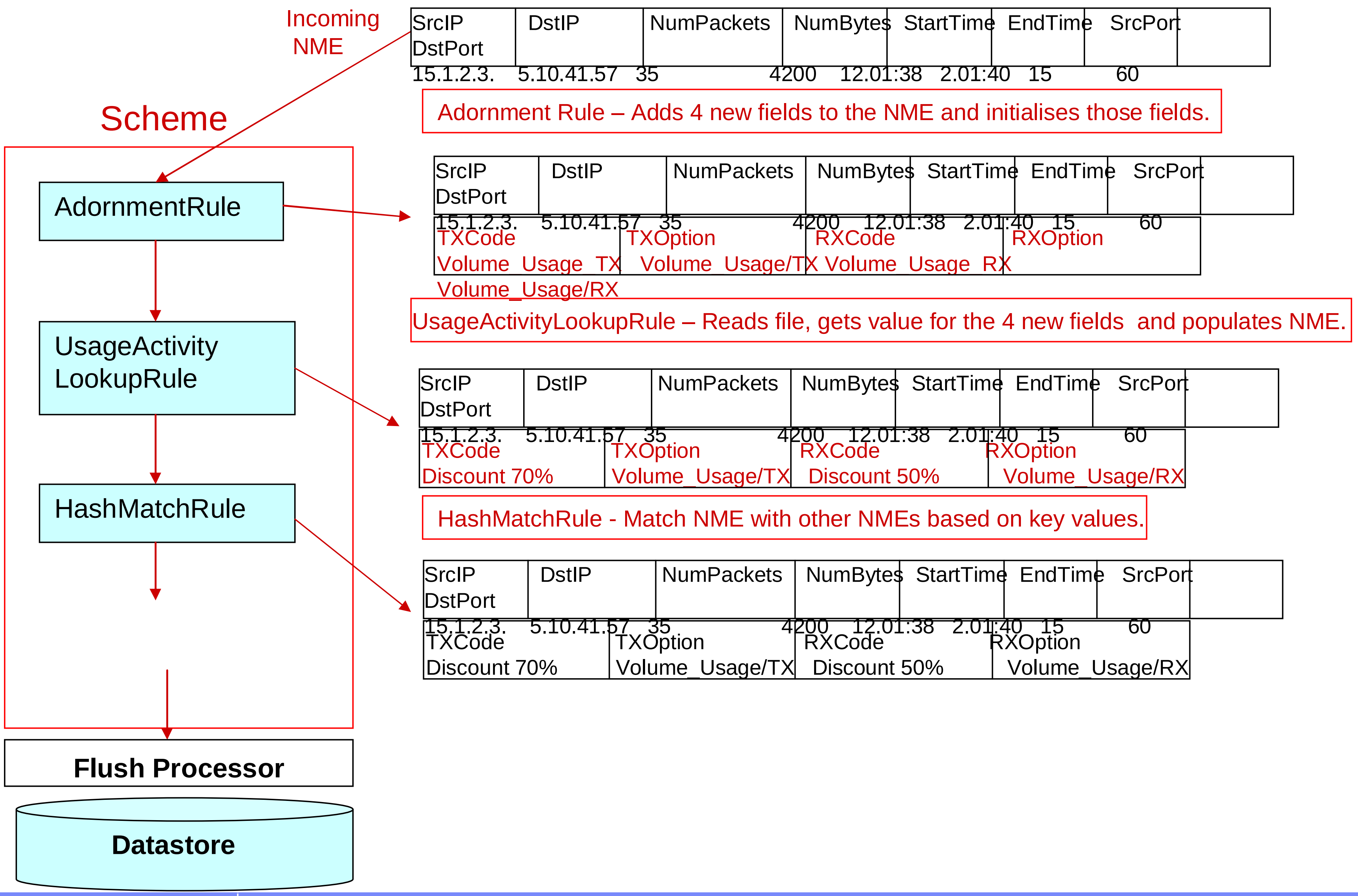
SrcIP DstPort	DstIP	NumPackets	NumBytes	StartTime	EndTime	SrcPort	
15.1.2.3.	5.10.41.57	35	4200	12.01:38	2.01:40	15	60

TXCode Volume_Usage_TX Volume_Usage/RX	TXOption Volume_Usage/TX	RXCode Volume_Usage_RX	RXOption

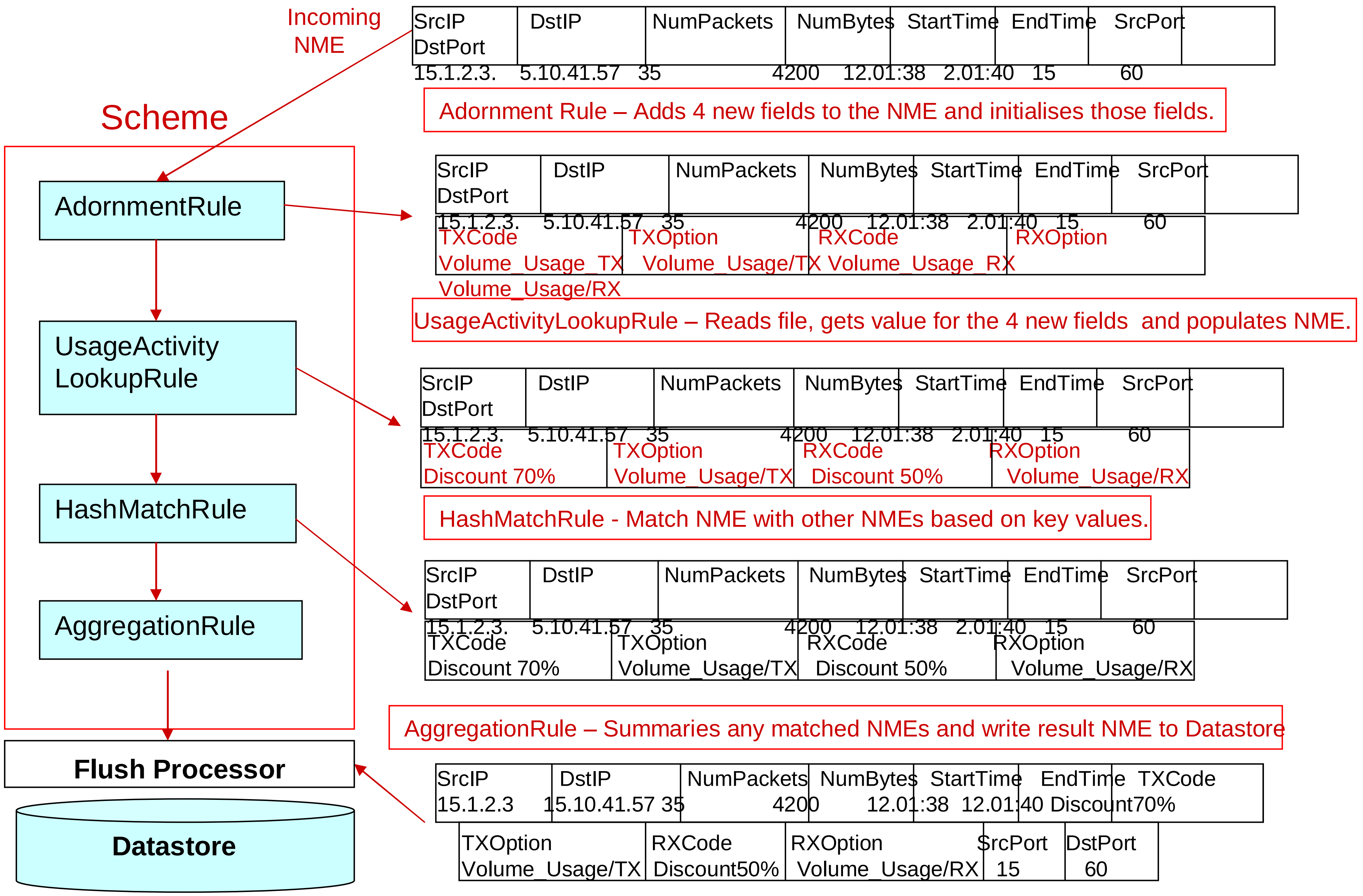
Simple Rule Scheme – Overview (3)



Simple Rule Scheme – Overview (4)


Incoming NME
Scheme
AdornmentRule
UsageActivity LookupRule
HashMatchRule
Flush Processor
Datastore
SrcIP DstPort | DstIP | NumPackets | NumBytes | StartTime | EndTime | SrcPort
15.1.2.3. 5.10.41.57 35 4200 12.01:38 2.01:40 15 60
Adornment Rule – Adds 4 new fields to the NME and initialises those fields.
TXCode Volume_Usage_TX | TXOption Volume_Usage/TX | RXCode Volume_Usage_RX | RXOption Volume_Usage/RX
UsageActivityLookupRule – Reads file, gets value for the 4 new fields and populates NME.
TXCode Discount 70% | TXOption Volume_Usage/TX | RXCode Discount 50% | RXOption Volume_Usage/RX
HashMatchRule - Match NME with other NMEs based on key values.

Simple Rule Scheme – Overview (5)



SrcIP DstPort	DstIP	NumPackets	NumBytes	StartTime	EndTime	SrcPort
15.1.2.3.	5.10.41.57	35	4200	12.01:38	2.01:40	15 60

Adornment Rule – Adds 4 new fields to the NME and initialises those fields.

SrcIP DstPort	DstIP	NumPackets	NumBytes	StartTime	EndTime	SrcPort
15.1.2.3.	5.10.41.57	35	4200	12.01:38	2.01:40	15 60

TXCode	TXOption	RXCode	RXOption
Volume_Usage_TX	Volume_Usage/TX	Volume_Usage_RX	Volume_Usage/RX

UsageActivityLookupRule – Reads file, gets value for the 4 new fields and populates NME.

SrcIP DstPort	DstIP	NumPackets	NumBytes	StartTime	EndTime	SrcPort
15.1.2.3.	5.10.41.57	35	4200	12.01:38	2.01:40	15 60

TXCode	TXOption	RXCode	RXOption
Discount 70%	Volume_Usage/TX	Discount 50%	Volume_Usage/RX

HashMatchRule - Match NME with other NMEs based on key values.

SrcIP DstPort	DstIP	NumPackets	NumBytes	StartTime	EndTime	SrcPort
15.1.2.3.	5.10.41.57	35	4200	12.01:38	2.01:40	15 60

TXCode	TXOption	RXCode	RXOption
Discount 70%	Volume_Usage/TX	Discount 50%	Volume_Usage/RX

AggregationRule – Summaries any matched NMEs and write result NME to Datastore

SrcIP	DstIP	NumPackets	NumBytes	StartTime	EndTime	TXCode
15.1.2.3	15.10.41.57	35	4200	12.01:38	12.01:40	Discount70%

TXOption	RXCode	RXOption	SrcPort	DstPort
Volume_Usage/TX	Discount50%	Volume_Usage/RX	15	60

Types of Datastores

Driver Name	NME Data	Meta Data
FileJDBCDatastore	Binary	Solid or Oracle
IDRJDBCDatastore	Ascii HTML XML IPDR	Solid or Oracle
ExternalJDBCDatastore	Solid or Oracle	Solid or Oracle
MuxDatastore		Solid or Oracle
JDBCDatastore	Solid or Oracle	Solid or Oracle
Accumulation FileJDBCDatastore	Binary	Solid or Oracle
Application JDBCDatastore	SDK	Solid or Oracle

This table shows the relationship between the Datastore Driver configuration attribute and the physical storage they represent.

For full details of all Datastores, see the Collector Component manual.

Recommended Datastores

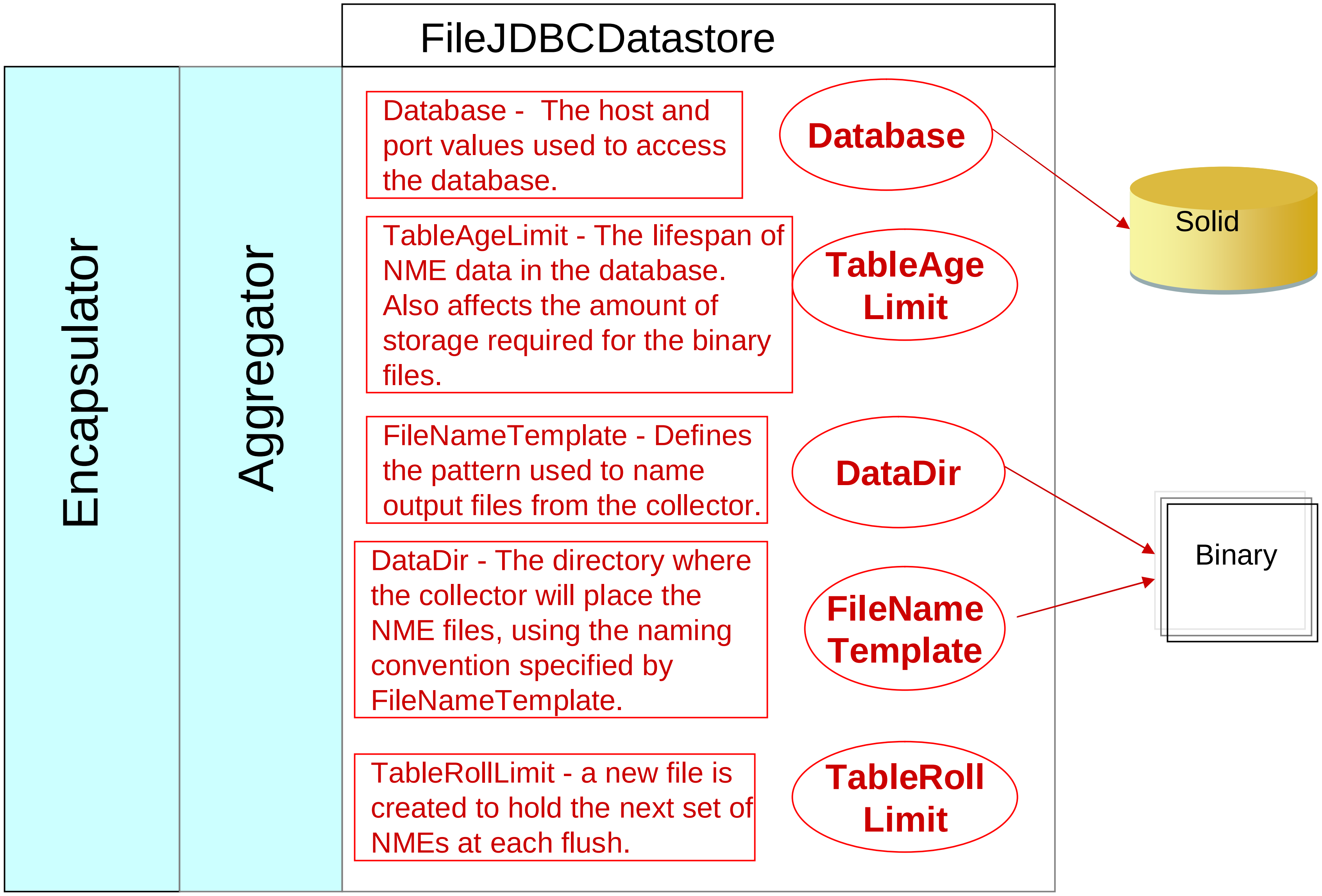
Driver Name	NME Data	Meta Data	Usage
FileJDBCDatastore	Binary	Solid or Oracle	Typically used for Level1 collectors reading large quantities of files, for example any CDR-reading collector. The performance of this datastore is the best of all datastores.
IDRJDBCDatastore	Ascii HTML XML IPDR	Solid or Oracle	Collectors which need to write out to a specific format, typically for output collectors to consuming applications, for example Billing systems. Can use the Data Delivery Agent to guarantee data transfer centralized DB or any external system or application.
External **JDBCDatastore**	Solid or Oracle	Solid or Oracle	External DB storage or centralized DB for geographically dispersed collectors. Used for Session Collectors and any other collectors where you need to do long queries. Gives control by specifying table and column configuration in the DB. Recommend Data Delivery Agent with FileJDBCDatastore instead if possible.

FileJDBC Datastore

Objective – Learn about the JDBCDatastore and some of its features:

- It outputs NMEs as BinaryFiles in the file system = FileDBCDatastore Driver.
- It controls how long NMEs are stored = Table Aging and Table Rolling.
- The filenames and locations are configurable = FileNameTemplate and DataDir.
- It stores MetaData in a Solid Database = Location of Solid DB.
- The Datastore can be queried to see the output NME Data = Query Data Tool and siuquery command.

FileJDBC Datastore - Overview



Database - The host and port values used to access the database.

TableAgeLimit - The lifespan of NME data in the database. Also affects the amount of storage required for the binary files.

FileNameTemplate - Defines the pattern used to name output files from the collector.

DataDir - The directory where the collector will place the NME files, using the naming convention specified by FileNameTemplate.

TableRollLimit - a new file is created to hold the next set of NMEs at each flush.

- End of session 2 on Advanced HP IUM

- Thanks a lot for your attention……