Advanced IUM Understanding – Session 4

- Presenter:
- Date:
- Duration:

- Ujjwal Barman
- June,2009
- 5 days

Agenda

- Charging Manager/Session server Features & Architecture
- Load Balancing in IUM
- Online/Real time Mediation
- Example of Real time mediation system
- Offline Mediation system
- Example of Offline Mediation System
- End to End Flow
- Description of one Production system Architecture
- Installing IUM
- Deploying the Patch for IUM
- Basic Code Understanding
  - Encapsulator,Aggregator,Datastore
- Commands Discussion.

Agenda

- 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
- Types of Datastores, Recommended Datastores, FileJDBC Datastore - Overview
- Data Backup and Recovery

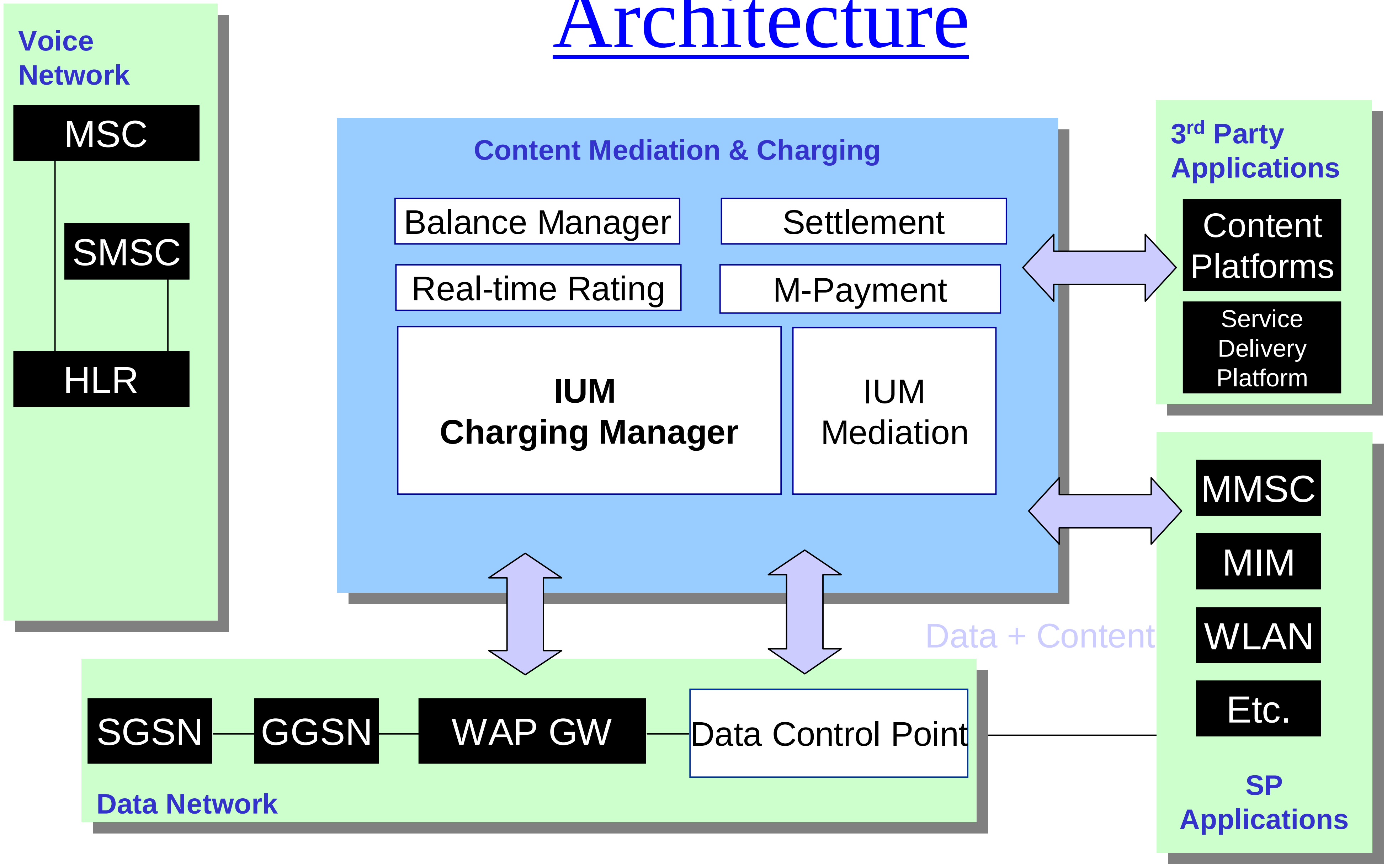
Agenda

- Solid Database Recovery, IUM Logging, Basic Logging
- Basic Log File Locations, Logging Tools: LogLooker, Database Admin: DBLooker
- Troubleshooting and Debugging

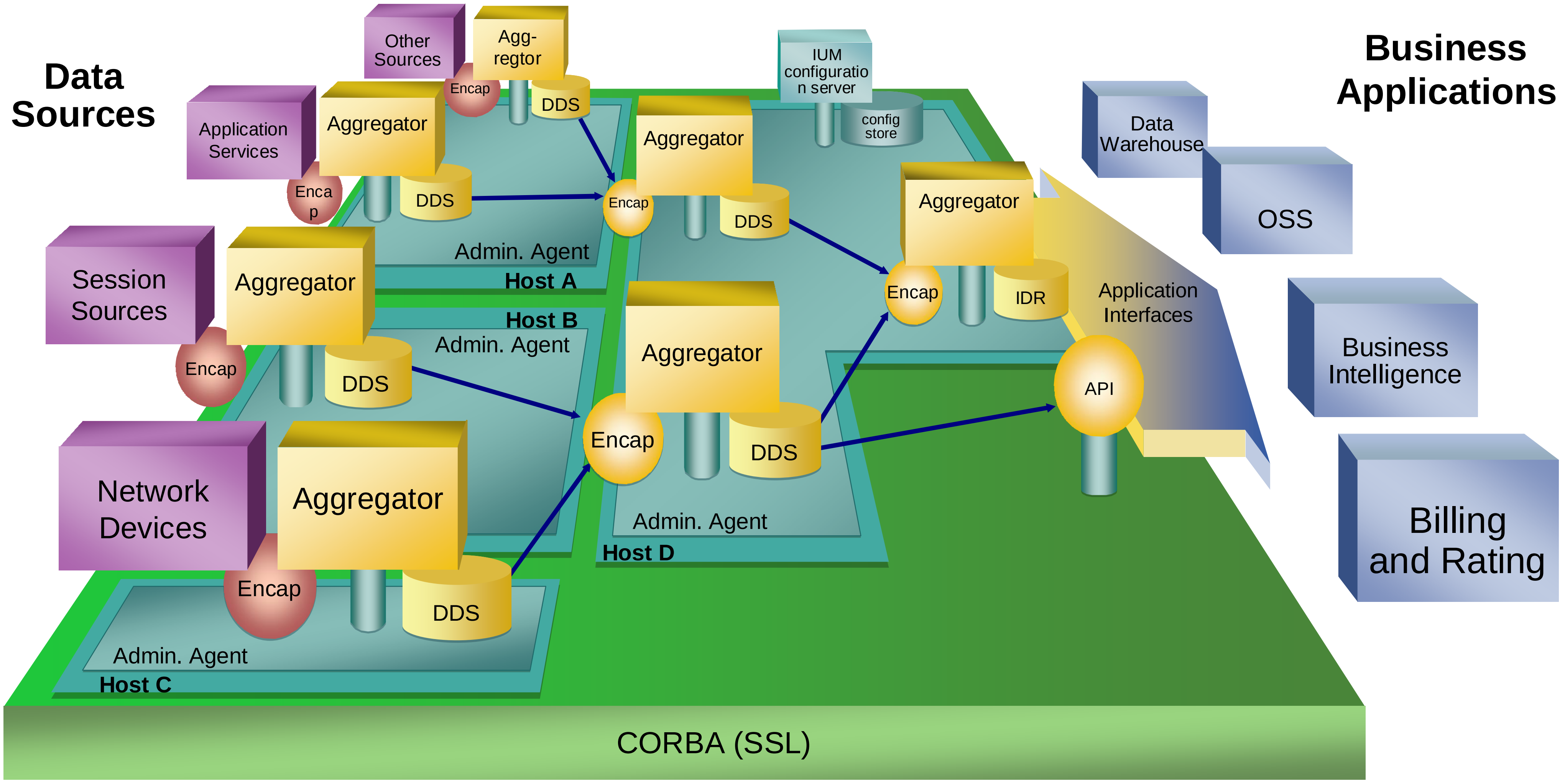
Charging Manager / Session Server Features

- Connects to Service Control Points and mediation
- Integrates with business applications
- Processes Real-Time authorization and accounting requests
- Rich rule set for flexible business logic
- Resilient and high-performance architecture
- Enables rapid deployment of new services

Charging Manager / Session Server Architecture



Designing a Deployment



What is IUM launchpad ?

•Tool to run all the collectors

•shows the complete deployment with all the collectors

•Gives option to start, stop any collector,admin agent, config server

•Gives option to see the logs

•Gives option to see the data getting processed.

•Gives option to query the solid DB

•Gives option to set log level of each collector.

IUM Processes

Config Server Process:

- The ConfigServer writes all configuration information for an IUM deployment to a database file: C:\Siu\var\config.db or /var/opt/SIU/config.db

- Every time a configuration change is made to a deployment, for example, changing a Collector attribute, the whole configuration is written to config.db, not just the changes.

- The last known copy of config.db is always written to config.db.last. This can be used to recover from if necessary.

- The log file C:\Siu\var\log\ConfigServer.log or /var/opt/SIU/log/ConfigServer.log records all changes to the Config Server.

IUM Processes (continued)

Admin Agent Process:

- The AdminAgent process is the IUM bootstrap process or parent process on an IUM host/server.

- If the process is not running, then no other processes or collectors can start-up or run.

- The AdminAgent is always associated with hostname of the host where it is running.

- The Admin Agent log file which records all processes starting and stopping is at: C:\Siu\var\log\hostname.log or /var/opt/SIU/log/hostname.log

IUM Processes (continued 2)

Process ids for every IUM process - AdminAgent, Config Server and all collectors are written to the log file of that process when that process starts up.

The PID is also written to a text file in the var directory for that process.

File Validation in IUM

- The handler validates files received from network elements like MSC/SMSC/MMSC. This component is used as a handler for the files validation in the InputFilePreprocessor component of the encapsulators for these sources

- The file validation handler validates the following
  - ✓ Header type and trailer values
  - ✓ Header and trailer lengths
  - ✓ Header format version
  - ✓ Charging block size
  - ✓ Actual data length in block as against the value put in header
  - ✓ Record count and match with record count in trailer

- In case the file validation fails in IUM, IUM parks the whole file in the configured error directory and logs a message in the collector log file. This file will not be processed in IUM until the file is corrected

Long Call Assembly in IUM

- For long call assembly we need to find the partial records belonging to the same session
- Hold on to partials for configurable amount of time when one or more are missing
- Create single output record for all available partials after predefined period of time. The consolidated CDR has the start time of the first available CDR, the end time of the last available CDR and an elapsed time, which is the sum of the individual durations
- In case of 1st partial CDR is missed, aggregate all the remaining records and send the aggregated CDR to downstream application. The consolidated CDR has the start time of the first available CDR, the end time of the last available CDR and an elapsed time, which is the sum of the individual durations

Commands- Please refer the Commands reference to know all the IUM related commands regarding the collector and system

Load Balancing in IUM

- Most of Production system of IUM runs parallal instances for load balancing and failover enablement. Load balancing between the two independent systems is done by the networks team.
- Each collector based on the amount of traffic and amount of logic being applied will have a maximum capacity of file to process. Based on that figure we need to decide the load balancing and how to design the architecture.
- Failover enablement should be configured and tested by IUM team. And load sharing should be tested by networks team.

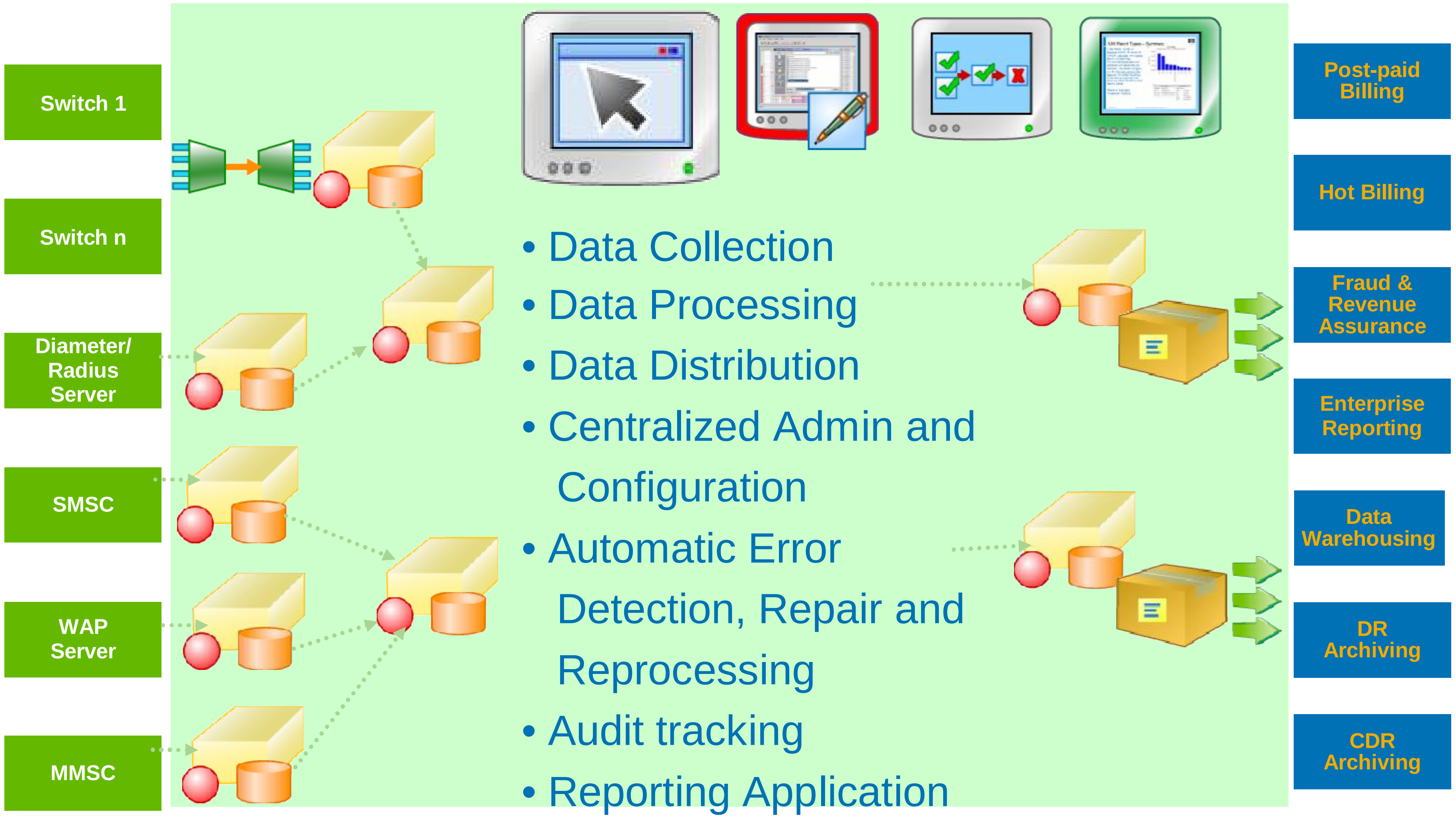
Online/Real time Mediation

- Real time / Online mediation is termed as High Alert system because they deal with the real time files. And in worst scenario if IUM is down its possible that no one in the network will be able to make or receive any calls.
- It deals with upstream applications like Cisco CSG(Real time system),etc and doen stream applications like rtBilling, LDAP server,Ranger,etc
- In real time it handles functionalities like authenticating each user information from LDAP servers before making or receiving any calls, Making checks with the rtBilling, checks with Ranger system for any fraud registered,etc.
- Real time system sits in the middle tier and do have the responsibility of doing the mediation and also doing the check on a real time basis and that makes the deployment complex and interesting too.

Offline Mediation system

- As per the name it deals with the offline switch level data
- The Usage data captured at the switch level is collected by IUM
- IUM processes the data and delivers the mediated O/P data to down stream applications like BSCS, DWH, FMS,etc
- It pulls the data from switches using file collection service or any of the fetch mechanism.
- It does' not deals with real time functions like authenticating each user and real time checks

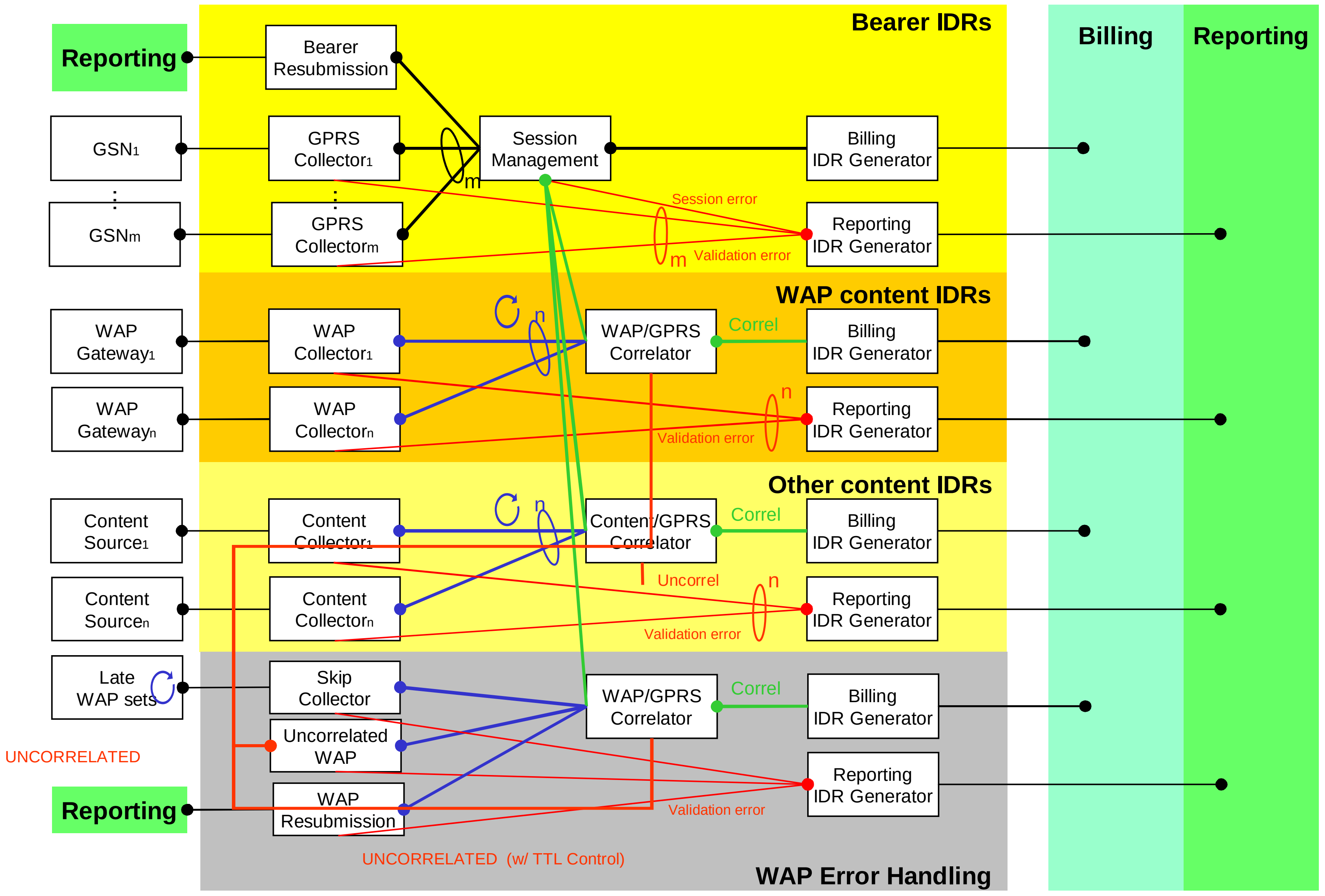
Example of a Offline mediation



- Data Collection
- Data Processing
- Data Distribution
- Centralized Admin and Configuration
- Automatic Error Detection, Repair and Reprocessing
- Audit tracking
- Reporting Application
- Guaranteed data delivery

End to End Flow – Offline Mediation scenario

HP Internet Usage Manager Overview



Installing IUM

The IUM License

- IUM Licenses are associated with the IP address of the management host machine (ConfigServer).
- Licenses can be requested from:
- http://www.hp.com/support/usage
- (Login and Password Required)
- The license gets emailed back to you as a an ASCII file – “license.config”.
- Copy the file to C:\Siu or /etc/opt/SIU
-
- Note: Demo/Evaluation Licenses can have “open IP addresses”
- (0.0.0.0) but will expire after 3 months.

The IUM License

- If the License expires, License.config can be over-written with a new License without having to reinstall IUM.
- Adding new solutions to IUM (for example, Session Server) means adding to the existing license.
- To update an existing license while IUM is running, overwrite existing license.config and the type: C:\Siu\bin updatelicense –f <license.config> or /opt/SIU/bin/updatelicense –f <license.config>.
- If IUM is not running, overwrite existing license.config and start IUM application (or restart Admin Agent process).

Install the Java Development Kit

• IUM 4.5 requires the full J2SDK (not the JRE) to be installed before you install IUM.

• To download, go to **http://www.java.sun.com** to download Solaris, Windows and Linux version. Go to **http://www.hp.com/go/java** for the HP-UX version

Time Synchronization

- All datasources and IUM Collectors within a deployment must be time synchronised. This allows correct data rule processing and correlation.
- IUM manages components based on UTC (GMT) Time.
- It is recommended that IUM host machines and data source that IUM receive data from use a NTP Service. Check with your network administrator.

The Installation Process

- Run the IUM installation wizard and following the instructions.
- The installation process copies all IUM related files into default locations - C:\SIU on Windows and

  /opt/SIU, /var/opt/SIU and /etc/opt/SIU on UX.

Notes:

1. You can install multiple instances of IUM on the

same host.

2. There is also an unattended install option.

3. On Unix platforms there is a non-root installation option.

Activating IUM 4.5

Once the earlier steps are done it will start installing and the later on it will ask if we want to activate the product. Click Yes and then automatically IUM product will be activated. The activation process is a separate process from the installation process. The installation process copies the IUM files onto the host.

The activation process does the following:
- Verifies the license.
- Configures access to the Config Server process on all hosts
- Starts the hosts Admin Agent process.
- Installs and starts the Config Server process on the management host.
- Loads all initial configuration information into the Config server.
- Starts the IUM Report Server if selected.

Activating IUM 4.5 (continued)

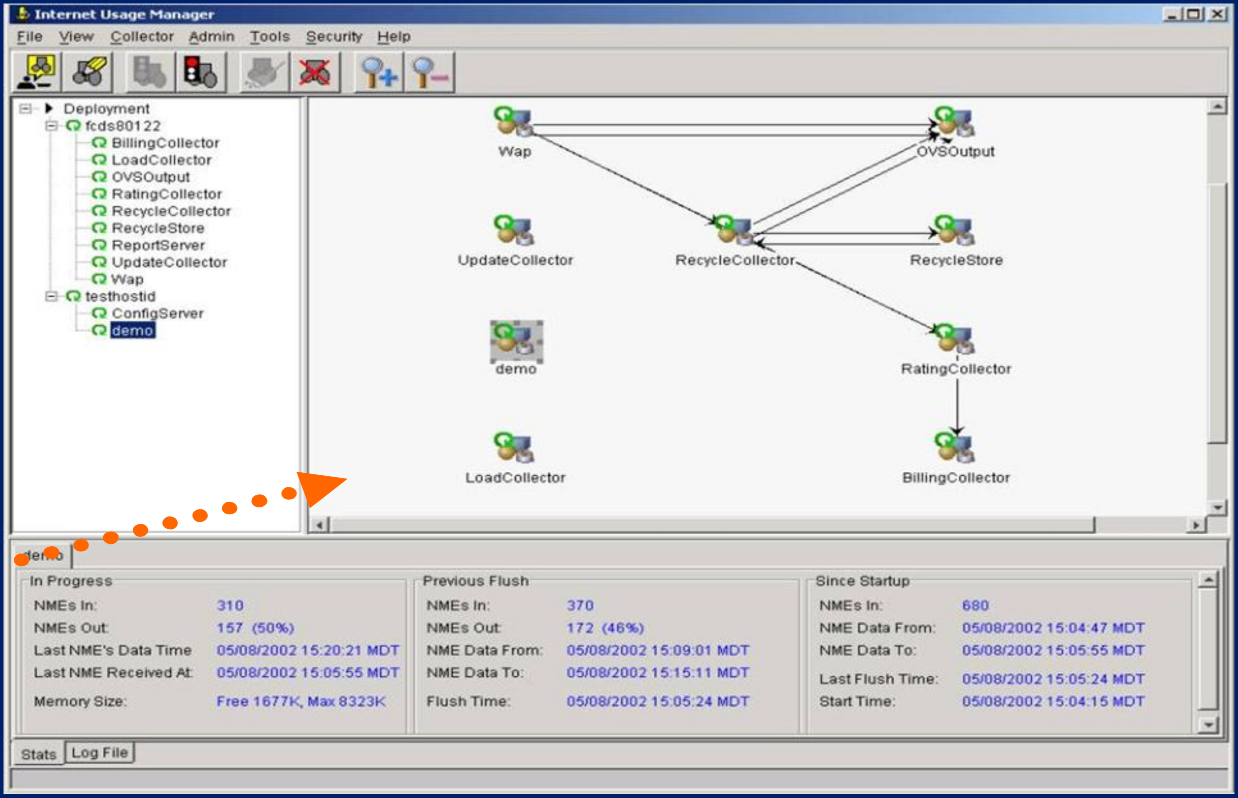
The activation process typically is run immediately after the installation process.

However it can be run separately, for example to add new functionality, as:
/opt/SIU/bin/activate –product ium on Unix platforms
C:\Siu\bin\activate –product ium on Windows platforms

IUM 4.5 GUI- Launchpad

Internet Usage Manager Launchpad

easy configuration, administration and monitoring of the entire IUM deployment



In Progress		Previous Flush		Since Startup	
NMEs In:	310	NMEs In:	370	NMEs In:	680
NMEs Out:	157 (50%)	NMEs Out:	172 (46%)	NME Data From:	05/08/2002 15:04:47 MDT
Last NME's Data Time	05/08/2002 15:20:21 MDT	NME Data From:	05/08/2002 15:09:01 MDT	NME Data To:	05/08/2002 15:05:55 MDT
Last NME Received At:	05/08/2002 15:05:55 MDT	NME Data To:	05/08/2002 15:15:11 MDT	Last Flush Time:	05/08/2002 15:05:24 MDT
Memory Size:	Free 1677K, Max 8323K	Flush Time:	05/08/2002 15:05:24 MDT	Start Time:	05/08/2002 15:04:15 MDT

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

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

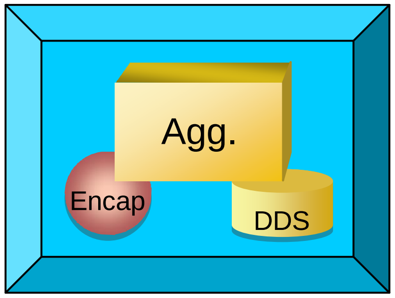
- The 3 components of a collector – Encapsulator, Aggregator, Datastore are independent objects.

- 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.

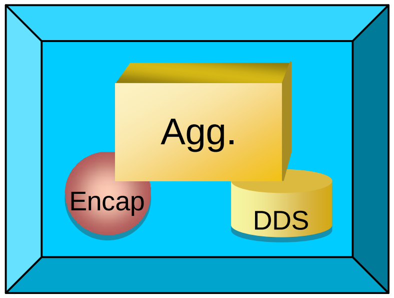


Creating Collectors (continued)

• 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.

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)

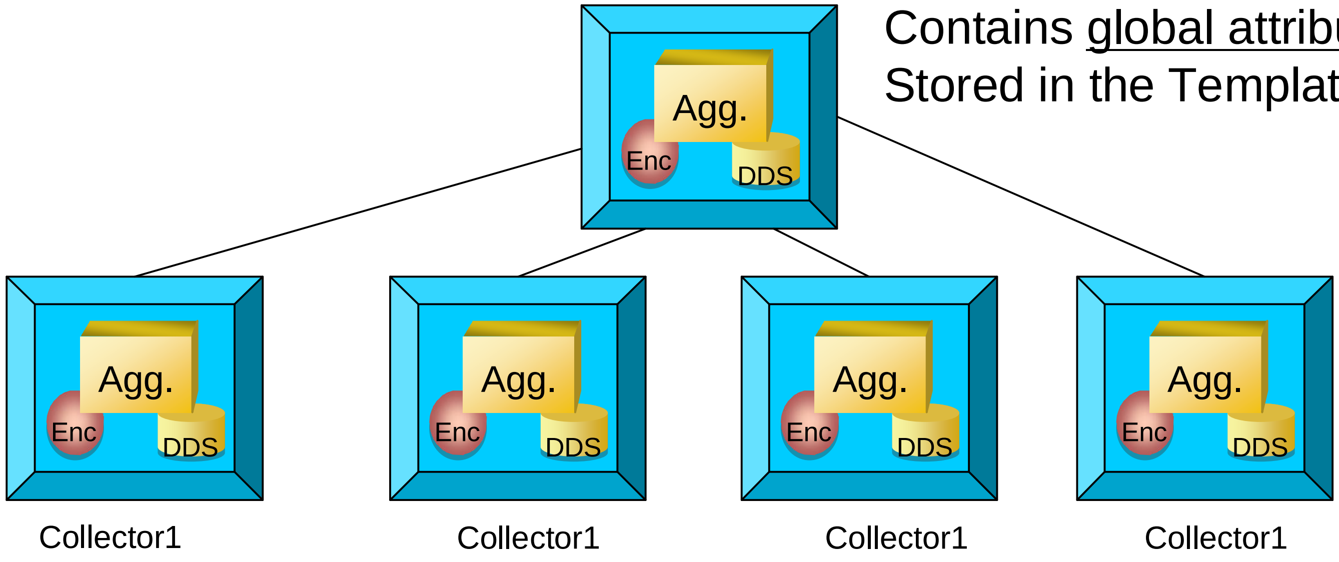
1. In Launchpad use the File → Export Configuration option or the saveconfig command to save a collector's configuration to a config file. Either in /opt/SIU/bin or C:\Siu\bin:
   ```
   saveconfig –n <collector name> -f <config file name>
   ```

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]
   ```

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)

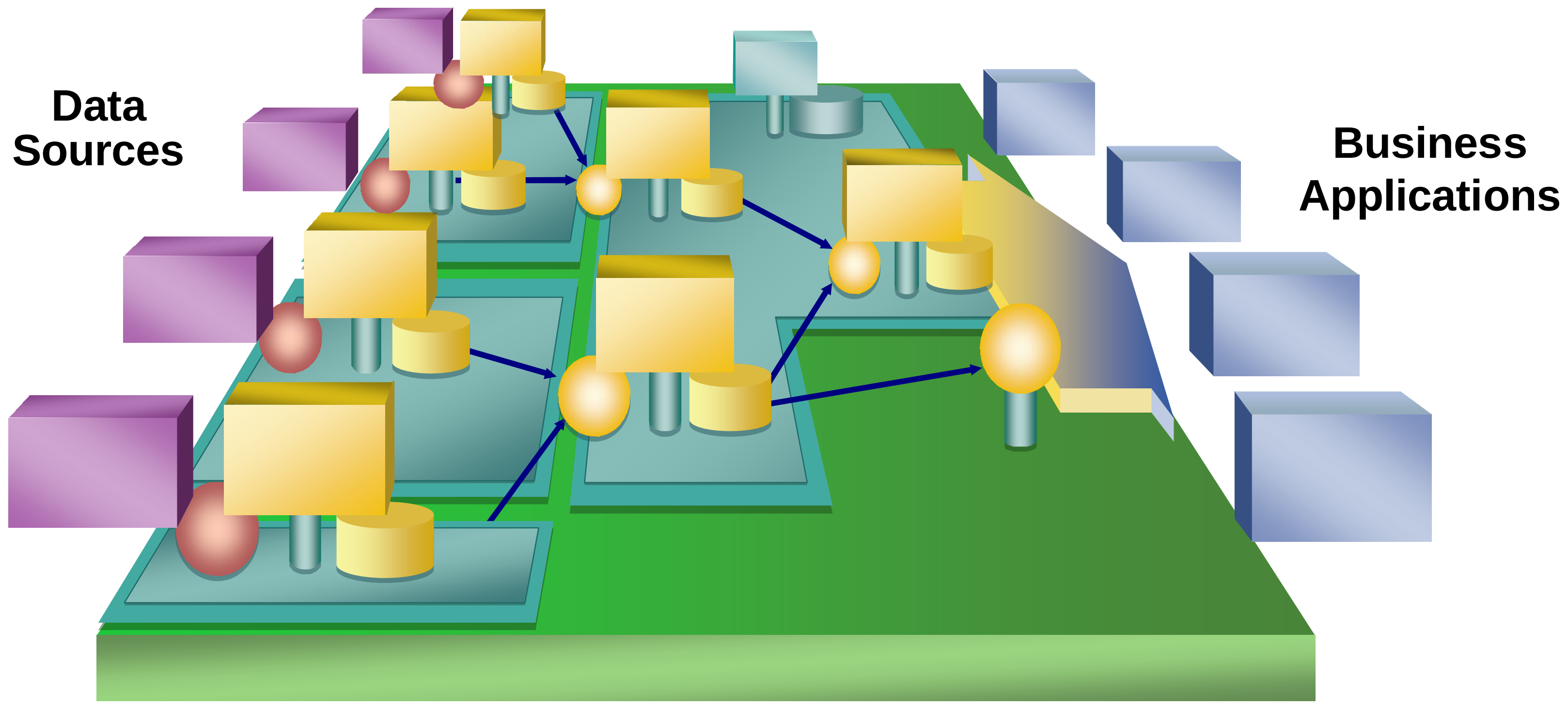
How to create Master and Linked Collectors, continued:

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.

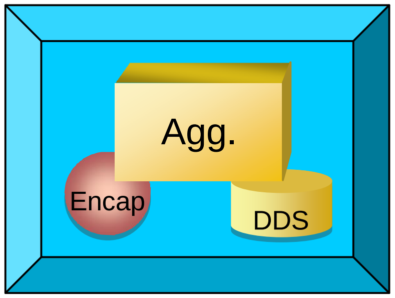
Collector Creation Model (continued)

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 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
```

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.

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

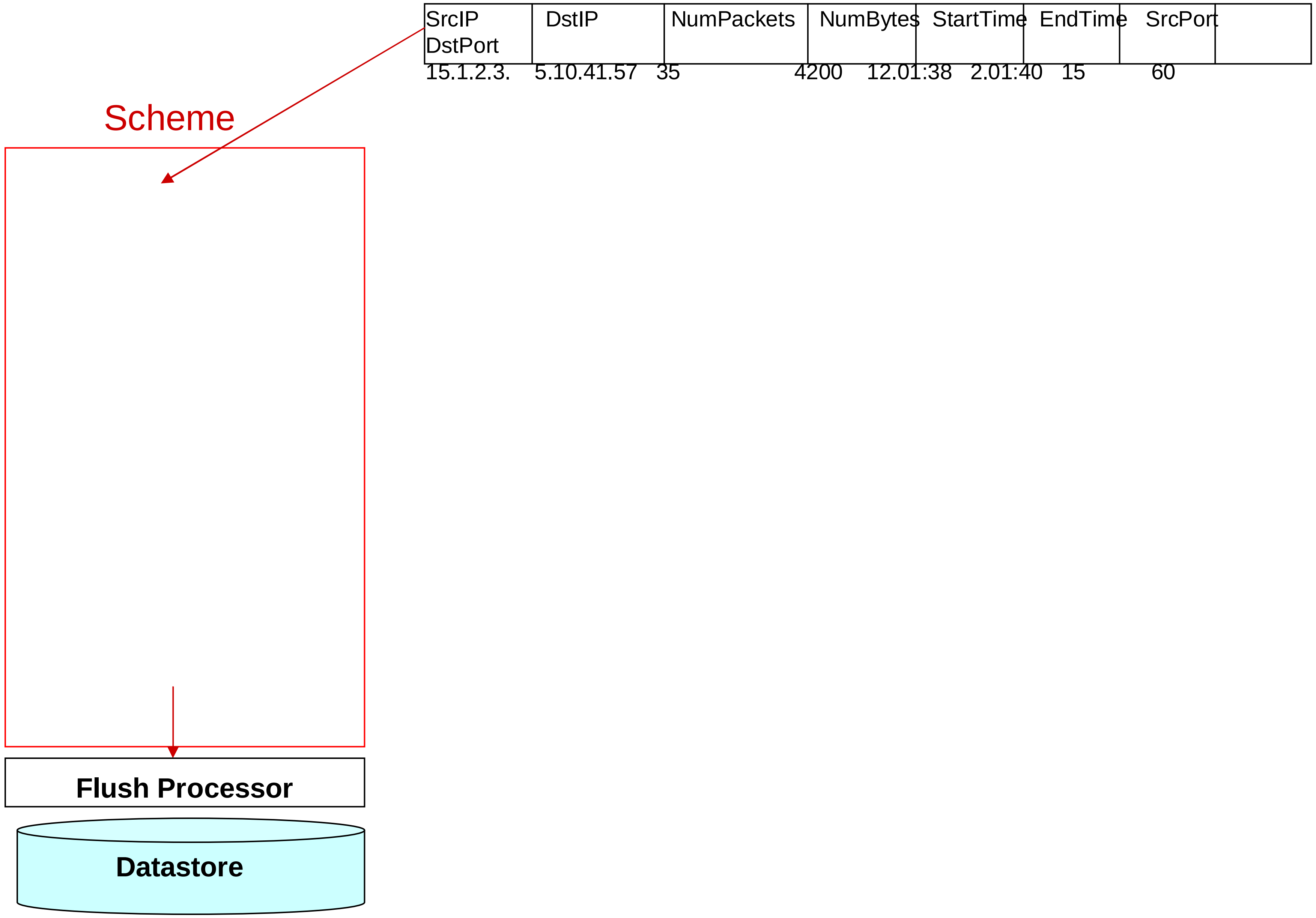
Storing
AggregationRule
StoreRule
SessionAggregationRule

Simple Rule Scheme

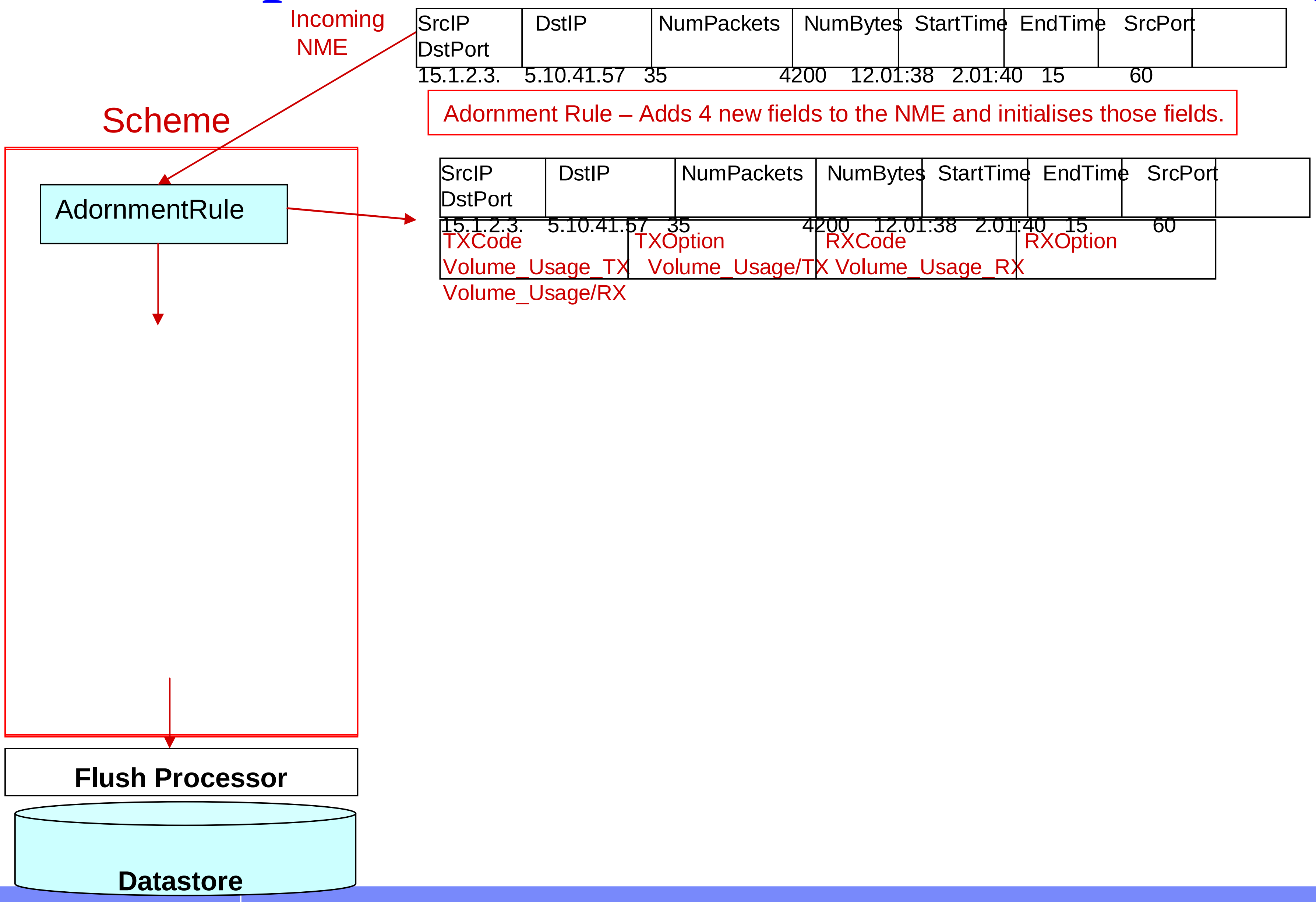
Objective – Learn about a simple Rule Scheme and some Aggregator
features:

- 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)



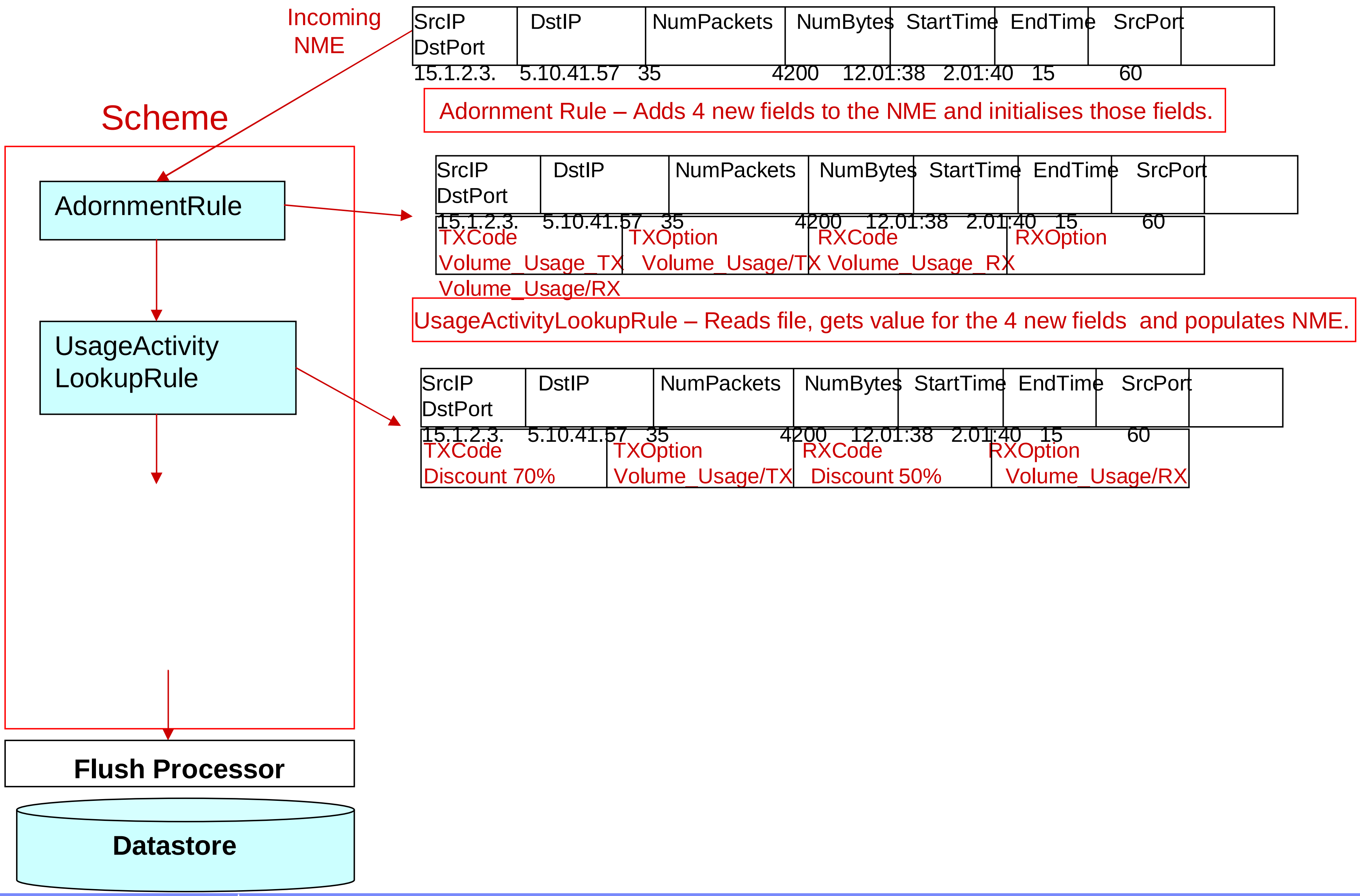
SrcIP DstPort	DstIP	NumPackets	NumBytes	StartTime	EndTime	SrcPort	
15.1.2.3.	5.10.41.57	35	4200	12.01:38	2.01:40	15	60

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	TXOption Volume_Usage/TX	RXCode Volume_Usage_RX	RXOption Volume_Usage/RX

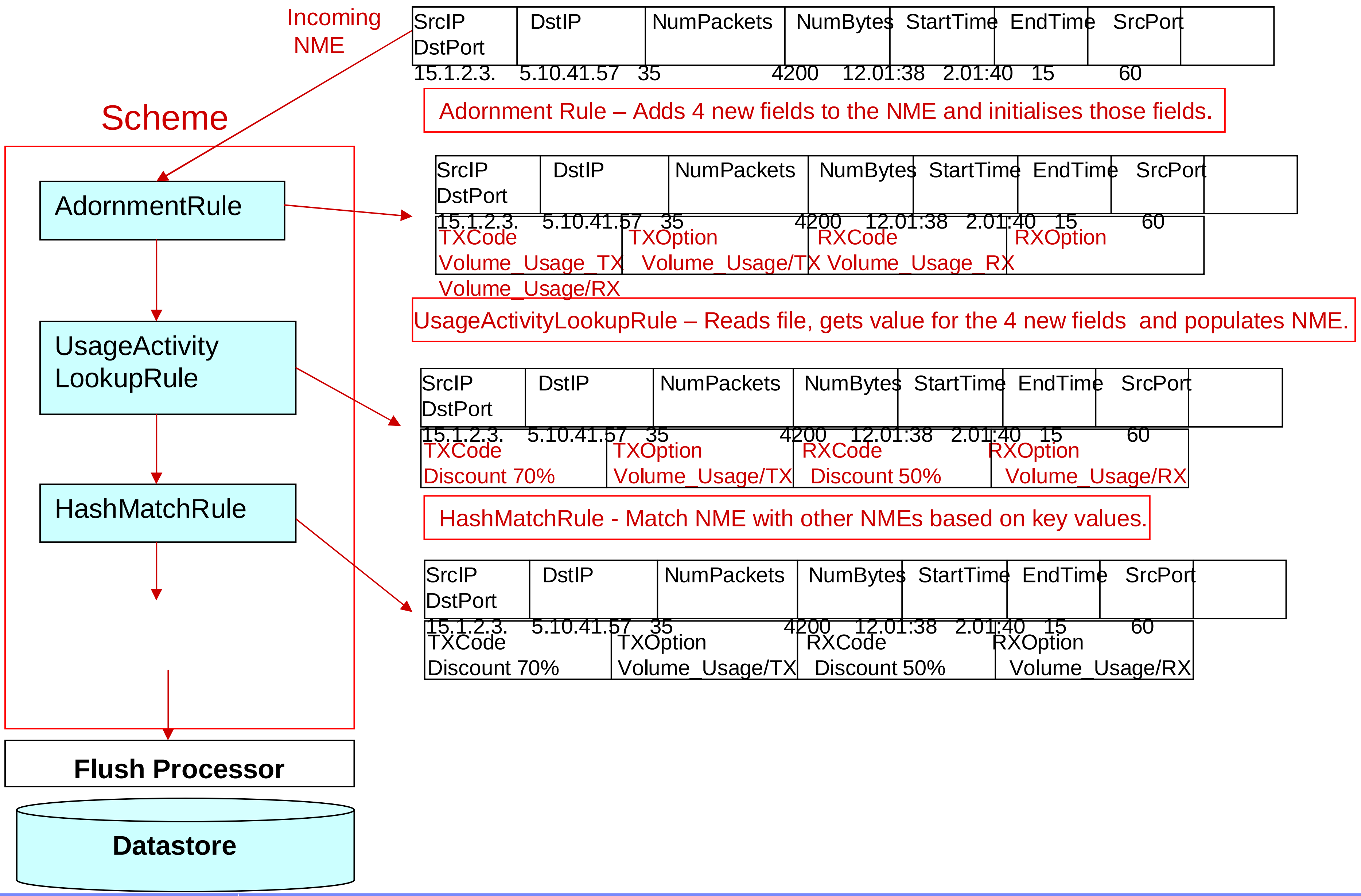
Simple Rule Scheme – Overview (3)


Incoming NME
Scheme
AdornmentRule
UsageActivity LookupRule
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
TXCode Volume_Usage_TX | TXOption Volume_Usage/TX | RXCode Volume_Usage_RX | RXOption Volume_Usage/RX
TXCode Discount 70% | TXOption Volume_Usage/TX | RXCode Discount 50% | RXOption Volume_Usage/RX

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

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

Simple Rule Scheme – Overview (4)

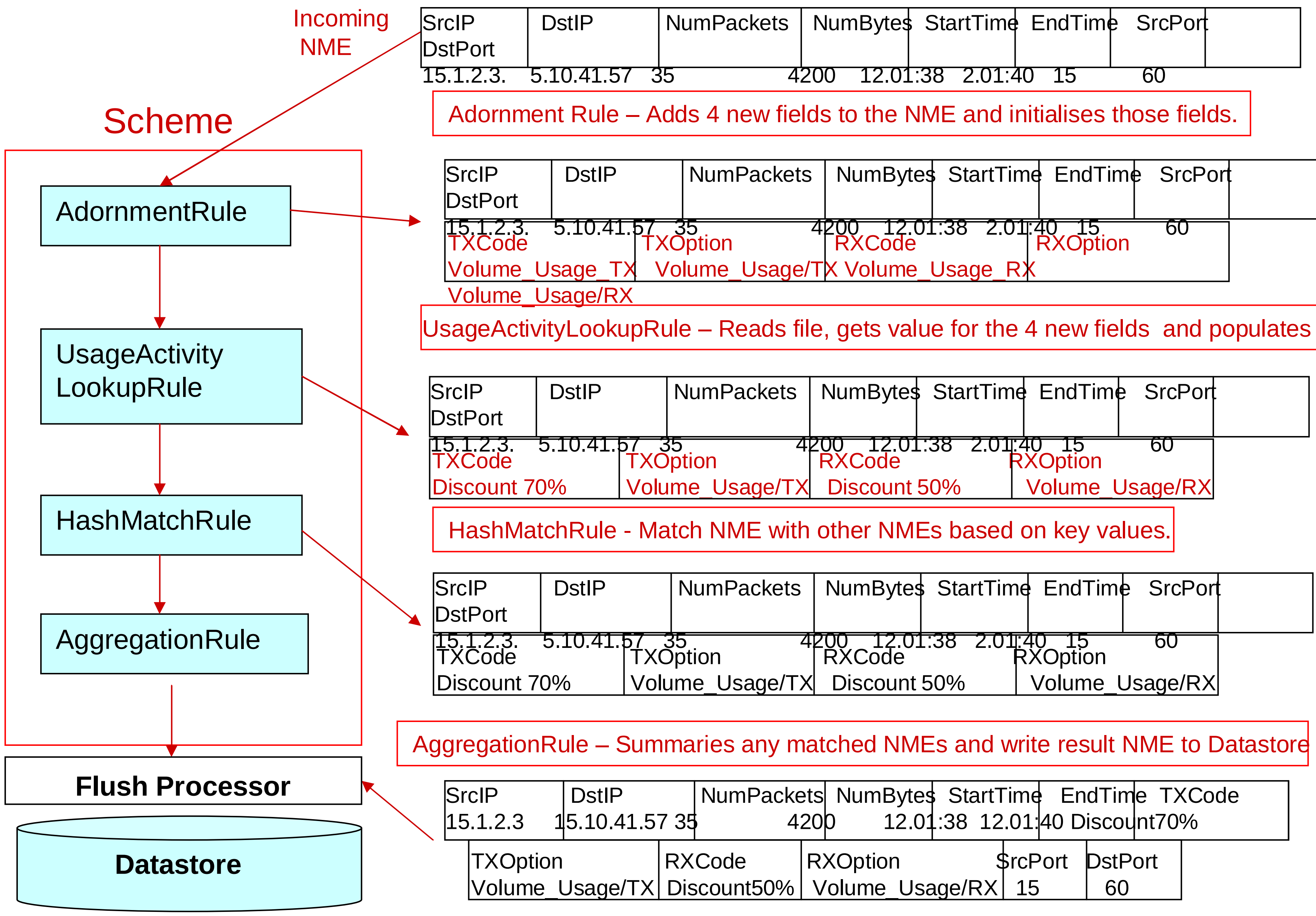


Adornment Rule – Adds 4 new fields to the NME and initialises those fields.

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

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

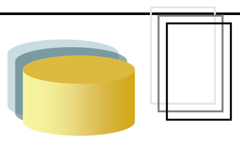
TXCode	TXOption	RXCode	RXOption
Discount 70%	Volume_Usage/TX	Discount 50%	Volume_Usage/RX

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.

IUM Back-up and Recovery

- These are recommendations of WHAT to back-up within IUM, NOT HOW to back-up.

- Everyone has a different back-up method.

IUM Back-up Recommendations

Data	Description
Collector Data and Configuration	Dynamic data. Use the Collector Online Backup utility. The big advantage is not stopping collectors and also one utility can backup different datastore types – db's or files.
Input Data	Dynamic data. Files or data stored in a local file system before being read into L1 collectors. Can use file system back-up.
Config Server store	Dynamic data during development phase. Config file containing configuration for all collectors, NME schema and all templates. Its always worth backing this file up regularly, even if using Collector online backup.
IUM Log Files	Dynamic data. Use a file system backup to backup these files if you want to keep them longer than their configure rollovers.
IUM file system	Static data. Not required to be backed up frequently but as part of a file system backup to backup the folders and files of C:/Siu or /var/opt/siu /opt/SIU and /etc/opt/SIU.
Databases for Metadata and/or NMEData	Dynamic Data. Solid or Oracle DBs can be used for MetaData and/or NMEData. If you use the Collector Online Backup then database backup is not also required.
File-based NMEData	Dynamic Data. Need to back up NME data files created by collectors with FileJDBC or IDRJDBCDatastore. If you use the Collector Online Backup then this backup is not also required.

How does Online Backup work ?

Sequence:

1. Flushes are **disabled**
2. Aging of Data is **disabled**
3. Recovery and History Tables are **saved**
4. Data is **saved**
5. Flushes are **enabled**
6. Aging of Data **enabled**

How does Online Backup work ? (continued)

- A Backup directory for the collector is created (if it doesn't exist).

- For a Full Backup, the directory contains:
  - BackupHistory.xml
  - BackupHistory<timeStamp>.xml
  - <collectorName><timeStamp>FULL.jar

- For an Incremental Backup, the directory contains:
  - BackupHistory.xml (updated)
  - <collectorName><timeStamp><id>.jar

How does Collector Restore work ?

Sequence:

1. Files are Restored
2. Database Tables are Restored
3. Collector is Configured (unless –noConfig was specified)

Config Server store Backup

- Configuration Server Config File Backup
  - The **saveconfig** command creates a backup:
    In C:\Siu\bin or /opt/SIU/bin:

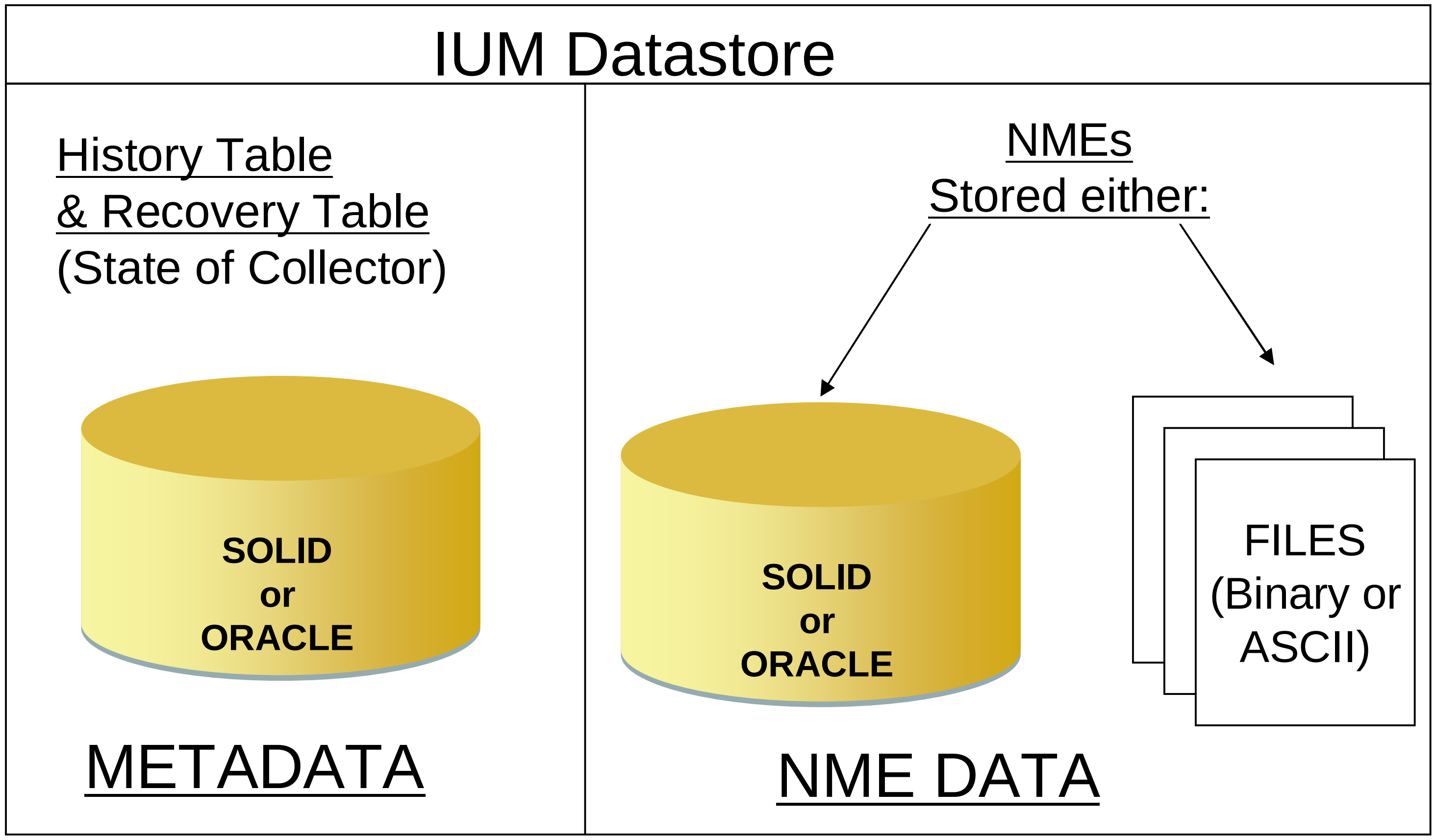
    saveconfig -p / -f <filename>.config

This will backup all collectors whether locally or remotely, plus the NME schema and all collector templates to one ascii file.

To backup the configuration into separate files for collectors, NMESchema, templates, etc = saveconfig –p / -f -dir <directoryname>

Note: Collectors do not need to be stopped during the backup of the Configuration Store

Datastore Database Backup



Database backup depends on which database is used - Solid or Oracle.
For either database type, it also depends how each collectors datastores are configured – to write MetaData to the database and optionally to write NME Data to the database.
Some deployments use multiple databases (geographically dispersed) or one centralized database for all collectors datastores.

Solid Database Recovery

There are Two options for recovery:

1. Recover to current state of failure
   - copy the backup db files to the database directory C:\Siu\var\Solid\db or /var/opt/SIU/Solid/db
   - copy the logfiles from the backup directory to the log dir (Do NOT overwrite the existing logfiles)
   - start the SOLID server
   - recovery is automatic

2. Return to state at backup
   - delete the current logfiles from the log directory
   - copy the db files to the database directory C:\Siu\var\opt\Solid\db or /var/opt/SIU/Solid/db
   - start the SOLID server process

IUM Logging

There are different logging capabilities available:

- Basic Logging
- Centralized Logging:
  - Using SYSLOG
  - Logging integration with OpenView Operations

These different logging methods are all fully documented in the Foundation Guide manual.

Basic Logging

Log Files – Levels:

- Set in LaunchPad – or suing the siucontrol command:

  0 = Critical

  1 = Accounting

  2 = Error

  3 = Warning

  4 = Informative (default setting)

  5 = Debug

  6 = Debug2

  7 = Debug 3

  8 = Debug 4
- View in LaunchPad or in C:\Siu\var\log or /var/opt/SIU/log

Basic Log File Locations

- Location =

  C:\Siu\var\log – NT
  /var/opt/siu/log – UX
- Naming Convention:
  <collector_name>.log
  <collector_name>.logOLD – rolled-over Logs
- LOGLINELIMIT – controls log size before roll-over
  Set to 0 = no rollover
  Default is 5120 lines.

Database Admin: DBLooker

- Graphical Database Browser customized to view IUM database contents.
- Displays IUM datastore tables in an organized way.
- Provides the ability to sort and filter table columns.
- Can run SQL commands directly.
- Can administer Solid and Oracle databases

- Available on request.

Troubleshooting

- Check all the IUM related process (in unix/HPUX/Linux) and service(windows) are running.

In unix run the following command to see the IUM related processes:

Ps –ef | grep SIU

It should show you that Configserver, Adminagent and Solid DB is running.

In case of windows go to services and check the services for the same.

- Check if all the related NME files are uploaded with the correct type description
- If the associated directories are created.
- Check the log files under the specified location. In UNI/HPUX all collector related logs will be under /var/opt/SIU/logs . And in windows under C:\SIU\var\logs
- Use unix level commands to grep each of the log files to see the error and check the corresponding collector code having error.

- *Now Please start practicing with IUM , try playing around with it to know more…*

- *Thank You for Attending….*