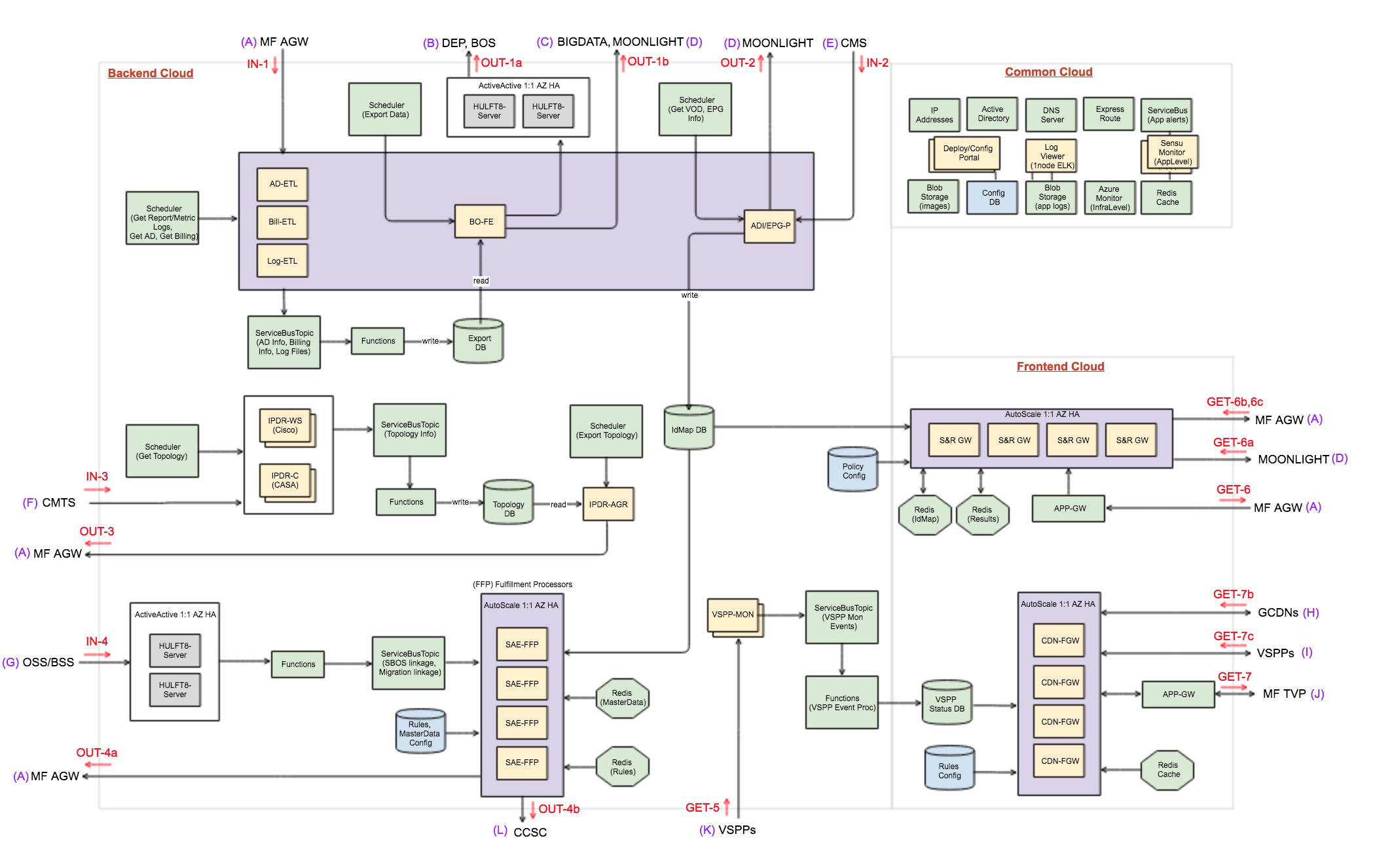
## Network plan

Services and modules within scope of system integration shall be hosted within back-end cloud and frontend cloud instances. Common services hosting portals for deployment, configuration and monitoring shall be deployed in common-cloud instance. The connectivity protocols between various services and systems is shown in the diagram.

## Infrastructure design

System integration components belonging to backend and frontend cloud instances are shown in the infrastructure design diagram below. It also shows the dependency on various Azure managed services such as database, message bus, scheduler, functions and fast access cache. The summary of interactions of components with external systems and Azure managed services is captured in the table below.

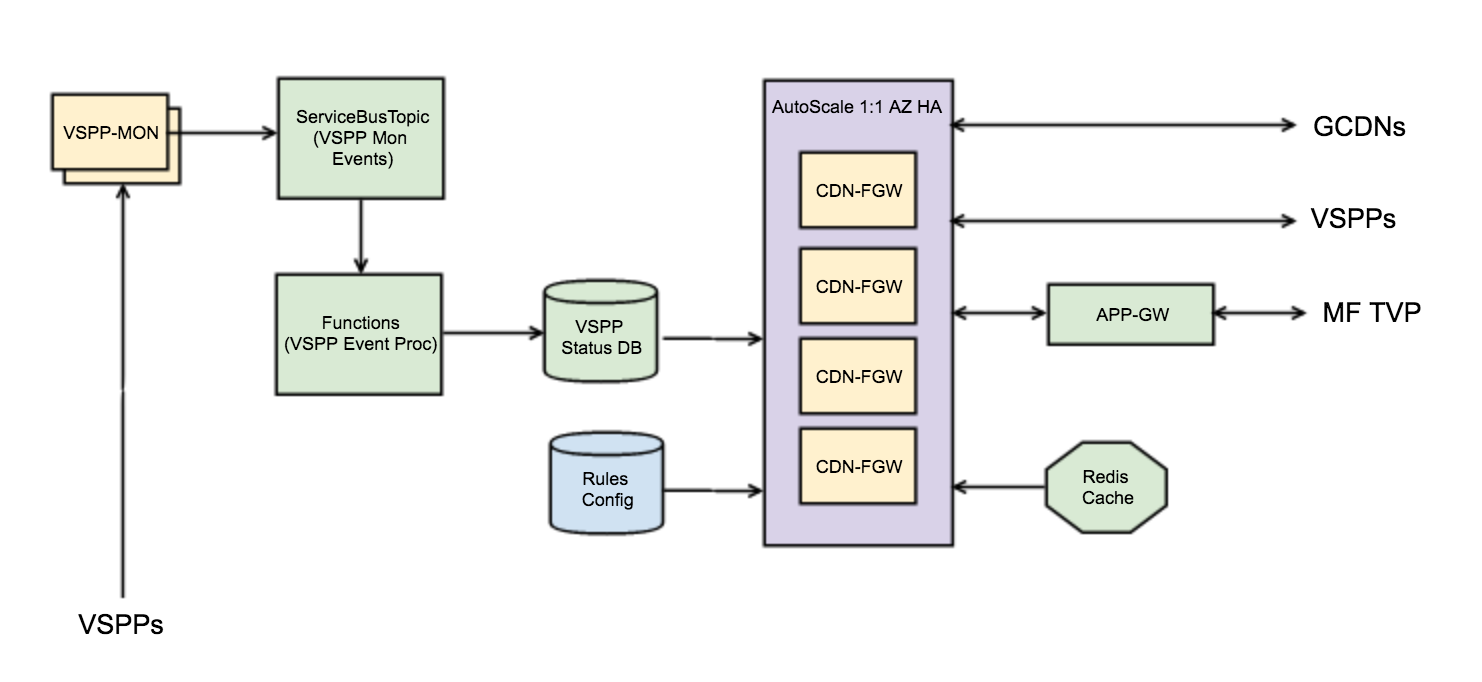
|  |  |  |
| --- | --- | --- |
| Cloud | Component | Description |
| Backend | AD-ETL | Extracts/transforms AD info (clicks and views) from MediaFirst. |
| Backend | Bill-ETL | Extracts/transforms billing records for TVOD/GOD/EST from MediaFirst. |
| Backend | Log-ETL | Extracts metrics/reports on customer viewing from MediaFirst. |
| Backend | BO-FE | Loads AD, Billing, Log info into back-office, big-data and moonlight. |
| Backend | ADI/EPG-P | Reads linear, vod metadata extracts ids and loads metadata into moonlight. |
| Backend | IPDR-WS | Extracts/transforms topology information from Cisco CMTS devices. |
| Backend | IPDR-C | Extracts/transforms topology information from CASA CMTS devices. |
| Backend | IPDR-AGR | Aggregates Cisco, CASA CMTS topology and loads into MediaFirst. |
| Backend | SAE-FFP | Process file based (de)activation orders calling MediaFirst, Concurrent APIs |
| Backend | VSPP-MON | Periodically check CDN utilization levels and store it locally. |
| Frontend | CDN-FGW | Resolves device content play request to CDN url based on selection policies. |
| Frontend | S&R GW | Bridge gaps between MediaFirst search/reco’s UX and Moonlight APIs. |
| Common | Deploy Portal | Web portal for deployment, upgrade of backend, frontend services. |
| Common | Config Portal | Web portal for all central configuration, rules and policy definitions. |
| Common | Sensu Monitor | Web portal for central health monitoring of services and infrastructure. |
| Common | Log Viewer | Web portal for application specific KPIs and dashboards. |



## CDN Federation Gateway (CDN-FGW)

This gateway is responsible for resolving client device requested playback content identifier to dynamic URL from one of the CDNs available to fulfill the delivery of this content. The CDN selection policy is configurable and based on a number of static parameters such as device type, in-home status, subnet or region/distribution node, service type (Linear, VOD, Catchup, Time-Shift) and time sensitive parameters such as day of week, CDN load distribution ratio and CDN utilization threshold.

Currently available global CDNs (GCDN) at Jcom are AWS cloud-front & CD network, while cable data edge distribution networks are VSPP Edge1 and VSPP Edge2. New CDNs can be configured.



### Components

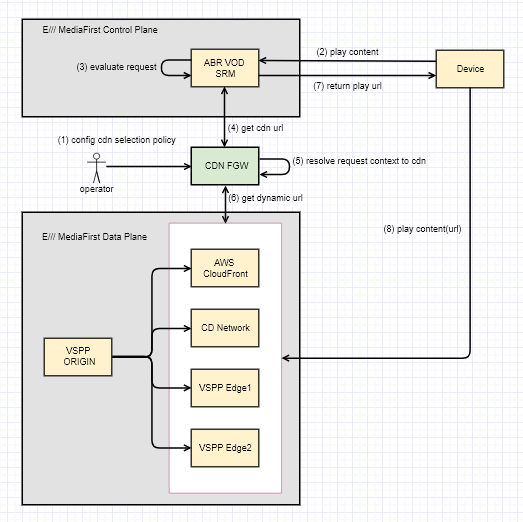
CDN-FGW: Implements client device facing http API that resolves request to specific CDN based on configured selection policies (rules configuration blob-storage) and returns dynamic URL from that CDN.

VSPP-MON: Periodically polls (based on configuration) VSPP data edge devices to retrieve latest percentage utilization level and forwards to a data store (VSPP-status-db).

Azure Managed Services: Service bus and functions are used to relay and write VSPP monitoring status to a data store. Redis cache is used to hold session state that influence the CDN selection criteria. Application gateway configured with web-app firewall policies that secure and load balance client device-initiated requests.

CDN selection policies are managed via common configuration portal and stored in blob-storage service. CDN-FGW http API is invoked by MediaFirst ABR session management service by passing necessary client device context information via application gateway.

CDN-FGW reads policies from blob-storage, resolves input context to specific CDN by updating and reading session state maintained in Redis cache. CDN-FGW invokes CDN service API to get dynamic URL for a given content identifier and return it to MediaFirst session manger.

****

### Configuration

Connection details of all the external (VSPP, GCDN) and internal (DB, Service Bus, Blob Storage, Redis) dependencies shall be made configurable. Sampling time duration for load distribution shall be configurable and defaulted to 1hour.

Sample CDN selection policy is shown below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Device  Type | IH/OOH  Status | Service Type | Region/  Subnet | Day of Week | Load Distribution | Utilization Threshold | Failover CDN | CDN  Resolution |
| STB | In-Home | Linear | Region1 | - | - | 90% | CD-N/W | VSPP-E1 |
| STB | In-Home | Linear | Region2 | - | - | 95% | AWS-CF | VSPP-E2 |
| CDA | Out-of-Home | Vod | Subnet1 | Mon-Fri | 10% | - | - | CD-N/W |
| CDA | Out-of-Home | Vod | Subnet2 | Sat-Sun | 30% | - | - | CD-N/W |
| CDA | Out-of-Home | Vod | All | | | - | - | AWS-CF |

### Non-Functional Requirements

A single instance of gateway must support a peak load of processing 5 requests per second with average latency of 200 milliseconds.

All aspects of processing requests in the gateway must be logged for auditory purpose. In order to join external flow context with the logs generated by the gateway, a suitable correlation id must be added to log entry where possible.

Additional processing information and failure details must be logged separately for troubleshooting purposes with different levels of logging i.e. info, warning, error etc. The reason for failure or warnings, such as bad configuration, bad input data, infrastructure service faults, CDN error must be logged.

It shall be possible to update CDN selection policy and reflect the changes in a best effort basis without having to restart the gateway instances.

It shall be possible to remotely monitor the functional health and the utilization rate of the underlying infrastructure resources.

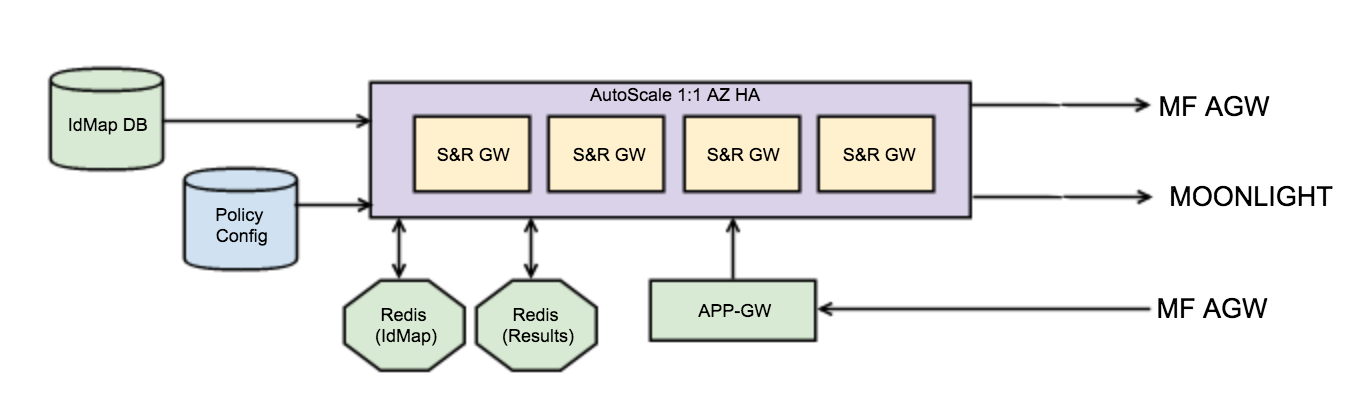
It shall be possible to automatically scale out/in compute resources. The gateway service shall be made available across two available sets.

# Search & Recommendations Gateway (S&R GW)

This gateway is responsible for performing all the adaptations required to bridge the gap between MediaFirst search & recommendations user experience needs and Moonlight search & recommendations engine capabilities. It also provides web-cache service to maintain (configurable time-to-live per customer context) and return results from fast access cache.

It involves content identifier remapping between MediaFirst and Moonlight domains when they appear in both requests/responses, assigning session index to request and aggregating Moonlight content results with content metadata, user-data from MediaFirst, and applying api specific presentation rules such as content sorting, filtering and collapsing.

The gateway assumes availability of content identifiers (title, episode, series, season) required for id remapping and to apply api specific presentation rules. These identifiers are extracted from CMS provided linear and vod metadata files described under OSS/BSS/CMS gateway section.



### Components

S&R GW: Implements client-device facing http APIs and orchestrates invocation of Moonlight search & recommendations engine APIs, MediaFirst metadata/user-data aggregation APIs, web-cache lookup, Moonlight MediaFirst content identifier remapping and implementing api specific presentation rules.

Azure managed services: Application gateway configured with web-app firewall policies that secure and load balance client device-initiated requests, Redis cache instances for maintaining web-cache and content identifier map, and Blob-storage service that hold configuration data and api specific presentation policies.

### Configuration

Connection details of all the external (MediaFirst URL endpoints, Moonlight S&R URL endpoints) and internal (Redis cache, id-map-db, policy-config-store) dependencies shall be configurable.

The S&R api specific presentation policies such as filter criteria, sort criteria and collapsing criteria shall be configurable. Time-to-live of results in the web-cache which are session specific shall be configurable.

### Non-Functional Requirements

All aspects of processing requests in the gateway must be logged for auditory purpose. In order to join external flow context with the logs generated by the gateway, a suitable correlation id must be added to log entry where possible.

Additional processing information and failure details must be logged separately for troubleshooting purposes with different levels of logging i.e. info, warning, error etc. The reason for failure or warnings, such as bad configuration, bad input data, infrastructure service faults, Moonlight search & recommendations engine errors must be logged.

It shall be possible to update presentation policy and reflect the changes in a best effort basis without having to restart the gateway instances.

It shall be possible to remotely monitor the functional health and the utilization rate of the underlying infrastructure resources.

It shall be possible to automatically scale out/in compute resources. The gateway service shall be made available across two available sets.

# Sunrise Activation Engine (SAE)

SCSK/Amdocs in the primary provider of business and operations support systems at JCOM. SmartPack is the customer order handling system that drives all the site, user account, device and service provisioning requests via batches of flat files to MediaFirst.

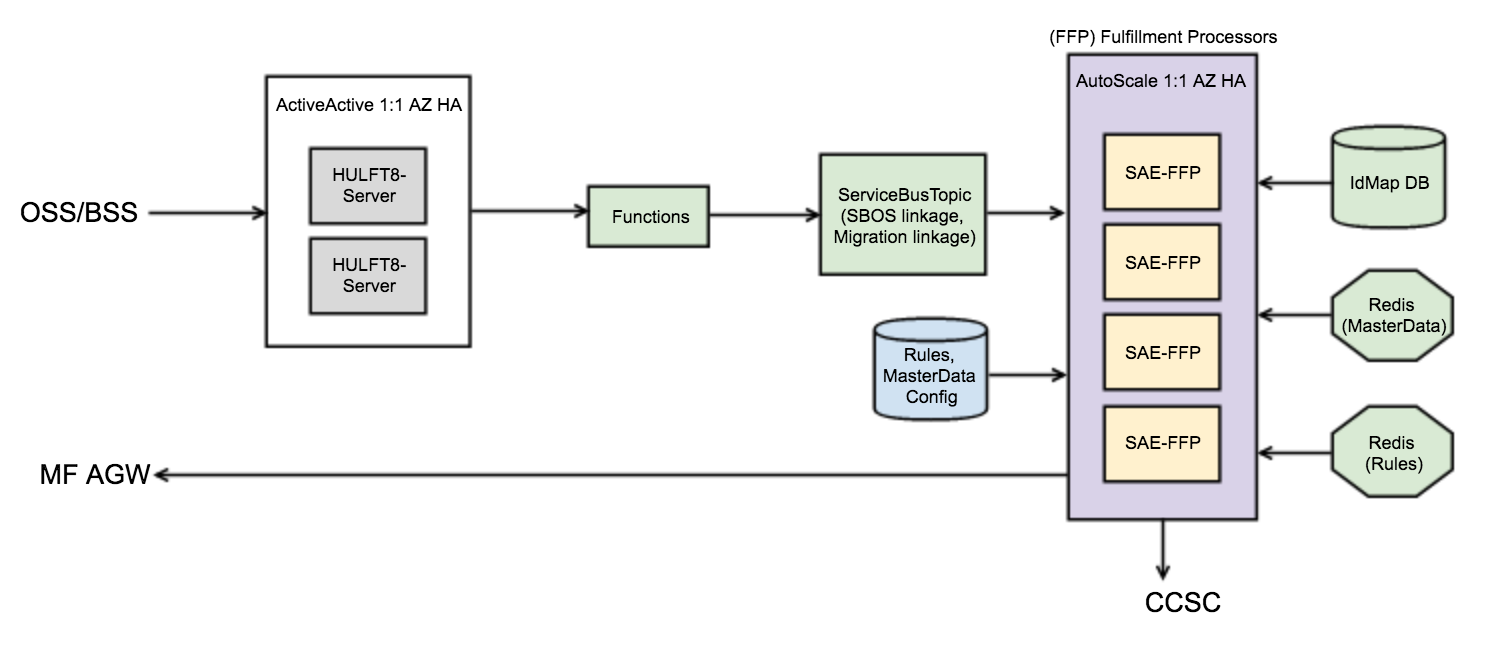
The SmartPack DEP (Data Extraction Platform) transfers files via HULFT8 managed data transfer protocol and the frequency is configured in HULFT8. Data in these files follow table/column/record structure, contain UTF-8 encoded characters and follow fixed length format.

Support following activation flows each invoking one or more MediaFirst APIs:

1. customer account creation and update,
2. customer account update (J:COM ID)
3. new service activation/deactivation,
4. customer account update (device add/remove)
5. customer service temporary deactivation
6. customer service permanent deactivation
7. customer relocation processing
8. customer account deletion
9. customer access rights for vod handover list

MediaFirst also supports segmenting users based on the services they purchase and apply promotional/discounted offers to certain users. After writing vod handover list to MediaFirst, it is also required to confirm this action by calling concurrent backend API.

The service activation and order fulfillment rules are expected to churn frequently and the activation engine is in the critical path of processing customer service requests. It is therefore required to externalize these business rules via configuration, provide rich set of audit/debugging logs and generate alerts that help monitor functional health of the engine and its dependent services.



### Components

HULFT8-Server: This managed file transfer server is configured to periodically pull files from OSS/BSS and write to shared folder. Functions are configured to forward these fixed length tag value files to service bus topics, based on the file type.

SAE-FFP: Sunrise activation engine fulfilment processor accepts data records in flat files and based on the business rules invokes one or more MediaFirst & Concurrent backend (CCSC) APIs. The activation engine uses operator configured rules to parsing files, validate file content and executing order fulfilment workflow.

### Configuration

Connection details of all the external (MediaFirst URL endpoints, Concurrent backend URL endpoints) and internal (Redis cache, id-map-db, policy-config-store) dependencies shall be configurable.

Any pre-requisite data (aka master data) like validation schema, postal codes, device types, sites, corps etc. which is required for the activation service to function shall be pre-configured.

The activation engine shall backup processed files and keep it for a configurable number of days (60 days default) for audit purposes.

The order fulfilment workflow has multiple steps in the flow and the workflow actions are configured via business rules. The different steps in the workflow can include:

(a) Pre-processing stage to determine the kind of action to be taken based on record type and its values, filtering of duplicate and extraneous records, dependency on reading the MediaFirst state required to fulfil the flow

(b) Main processing stage to provision/de-provision customers, devices, services, customer-relocation and set customer vod-handover-list

(c) Post processing stage to optionally assign customers to promotional offers in MediaFirst based on certain conditions.

Customers moving from one corp to another should be supported and this is signalled via special file type. This file has source customer-corp identifier and destination customer-corp identifier. The activation engine will have to invoke a move customer API profile which requires calling around 10 MediaFirst APIs.

Customers moving from concurrent to the new sunrise platform will require handover of purchased vod assets so that access rights are established when they start using sunrise powered devices. The asset list per customer will be signalled in a special file type and this will require calling MediaFirst and Concurrent backend APIs.

### Non-Functional Requirements

All aspects of processing requests in the gateway must be logged for auditory purpose. In order to join external flow context with the logs generated by the gateway, a suitable correlation id must be added to log entry where possible.

Additional processing information and failure details must be logged separately for troubleshooting purposes with different levels of logging i.e. info, warning, error etc. The reason for failure or warnings, such as bad configuration, bad input data, infrastructure service faults, Moonlight search & recommendations engine errors must be logged.

It shall be possible to update activation workflow business rules and reflect the changes in a best effort basis without having to restart the instances.

It shall be possible to remotely monitor the functional health and the utilization rate of the underlying infrastructure resources.

It shall be possible to automatically scale out/in compute resources. The activation service shall be made available across two available sets.

A peak load of processing 15,000 records in 5 mins must be supported. It shall be possible to quickly filter and view processed records by rate code, account, time window, order status etc.

Some relevant static data entities -

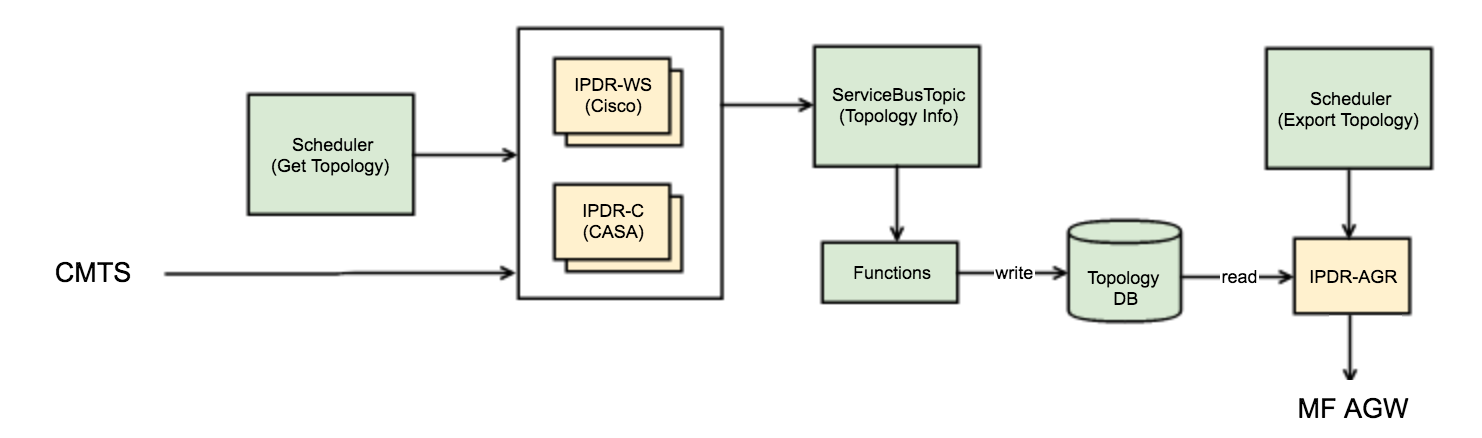
* Rate code master = 30,000
* Corp = 73
* MGT = 100
* Sites (linking site to corps and site to nodes) = 4000
* Equipment types = 50
* Node, Outlet
* Map of corp, rate code to MediaFirst offer id
* Customer, Device & Service defaults

It must be possible to reload static data without requiring service restarts.

Static data should also include promotional/discounted segments that are used in the post processing stage of certain orders. The customer segmentation rules are configurable.

# IPDR Collector (IPDR-C, WS, AGR)

This service is responsible for periodically collecting fibre-node topology information using IPDR (IP Data Records) from 350+ CASA and Cisco CMTS data edge routers, aggregate collected data and synchronize it with MediaFirst platform.



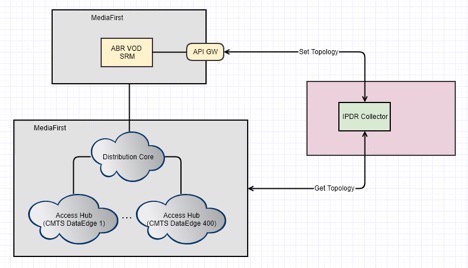
### Components

IPDR-WS: This component when triggered is responsible for collecting topology information from Cisco CMTS either using IPDR streaming protocol from IPDR exporter running in every CMTS instance or using CMTS SNMP MIBs that provide the same topology information. The collected records are filtered based on MAC Domain service group & downstream channel bonding information, relayed via service bus and stored in topology database.

IPDR-C: This component when triggered is responsible for pulling topology information from CASA CMTS using IPDR format over FTP. The collected records are filtered based on MAC Domain service group & downstream channel bonding information, relayed via service bus and stored in topology database.

IPDR-AGR: This component when triggered is responsible for reading latest topology information from CASA and Cisco CMTS devices, aggregating them in single output format based on MediaFirst SRM definition and loaded using MediaFirst APIs.

Azure Managed Services: Scheduler is used to configure topology collection and export triggers based on operator needs. Service bus and functions used to relay collected data to data store.



### Configuration

Connection details of all the external (MediaFirst URL endpoints) and internal (service bus, topology store) dependencies shall be configurable.

The collector shall keep local copy of exported topology for a configurable number of days (60 days default). If it is required to re-transfer files due to any error in MediaFirst, the local copy can be used but manual intervention is required and export triggered via scheduler.

The collector shall not perform any validation, consistency checks, uniqueness checks are referential integrity checks of the data it exports.

### Non-Functional Requirements

All aspects of processing requests in the gateway must be logged for auditory purpose. In order to join external flow context with the logs generated by the gateway, a suitable correlation id must be added to log entry where possible.

Additional processing information and failure details must be logged separately for troubleshooting purposes with different levels of logging i.e. info, warning, error etc. The reason for failure or warnings, such as bad configuration, bad input data, infrastructure service faults, CMTS API/data errors must be logged.

It shall be possible to remotely monitor the functional health and the utilization rate of the underlying infrastructure resources.

# Back-office Gateway (AD-ETL, Bill-ETL, Log-ETL, BO-FE, ADI/EPG-P)

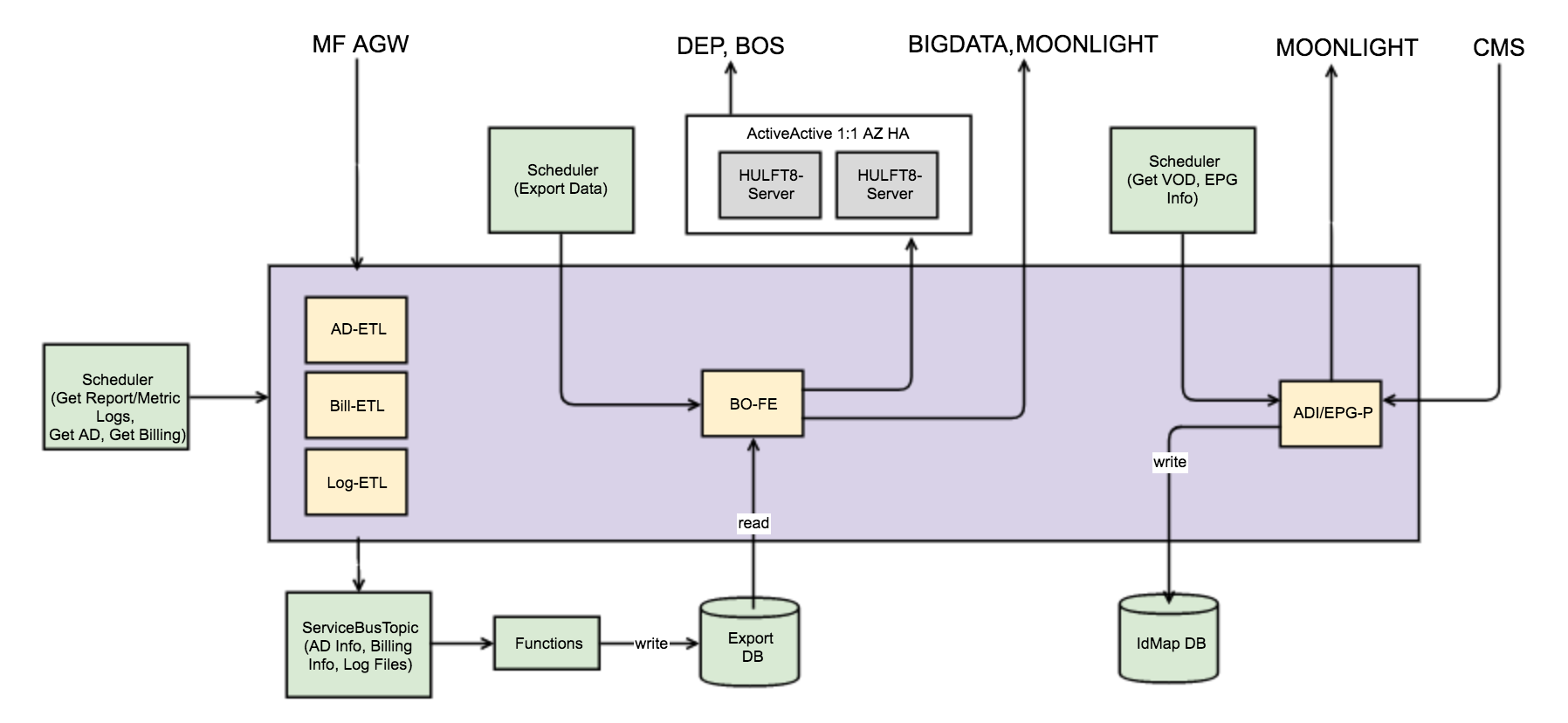
The AD ETL module is responsible for extracting AD information (clicks and views) from MediaFirst via xml/http-based API, transforming from UTF-8-character code to shift-JIS in a fixed length file format with file headers/trailer info and loading converted files to db.

The Bill ETL module is responsible for extracting billing records generated for TVOD/GOD rental/EST purchase from MediaFirst via xml/http-based API, transforming from UTF-8-character code to shift-JIS in a fixed length file format with file headers/trailer info and loading converted files to db.

The Log ETL module is responsible for extracting ~10 different types of metrics/reports related to customer viewing activity from MediaFirst via xml/http-based API, and loading files to db.

The BO-FE module is responsible for reading different types of files (AD, Bill, Logs) from db previously extracted from MediaFirst, and triggered by scheduler transfer different set of files to multiple destinations OSS/BSS (via HULFT8) and BIG-DATA, Moonlight via FTP.

The ADI/EPG-P module is responsible for extracting linear XMLTV and vod ADI xml files from CMS via FTP, extracting certain identifiers and types from the file writing to db, and loading ingested xml files as is to Moonlight system via FTP.



### Components

AD-ETL: This component reads AD viewing records from MediaFirst, filtering records based on Jcom needs and converting to flat file format with header/footer/consistency checks and writes to internal data store via service bus relay and functions.

Bill-ETL: This component reads Billing records from MediaFirst, filtering TVOD/GOD/EST purchase records, extracting required info from records and converting to flat file format with header/footer/consistency checks and writes to internal data store via service bus relay and functions.

Log-ETL: This component reads different kinds of metrics/reports related to customer content consumption and writes to internal data store via service bus relay and functions.

BO-FE: This component is triggered by scheduler to read different types of data files available in data store and export them to configured external systems.

ADI/EPG-P: This component reads linear and vod metadata files, extracts certain identifiers to data store and writes files to Moonlight FTP server.

Azure Managed Services: Scheduler is used to configure data collection and export triggers based on operator needs. Service bus and functions used to relay collected data to data store.

### Configuration

Connection details of all the external (MediaFirst URL endpoints, CMS FTP server, Moonlight FTP server, BIG-DATA FTP server) and internal (service bus, export store) dependencies shall be configurable.

Different kinds of data files collected and stored in db shall be kept for a configurable number of days (60 days default). If it is required to re-transfer files, the local copy can be used but manual intervention is required and export triggered via scheduler.

The xml paths of elements in the XMLTV and ADI xml files that signal identifiers to be extracted and stored shall be made configurable.

The export modules shall not perform any validation, consistency checks, uniqueness checks are referential integrity checks of the data it exports.

### Non-Functional Requirements

All aspects of processing requests in the gateway must be logged for auditory purpose. In order to join external flow context with the logs generated by the gateway, a suitable correlation id must be added to log entry where possible.

Additional processing information and failure details must be logged separately for troubleshooting purposes with different levels of logging i.e. info, warning, error etc. The reason for failure or warnings, such as bad configuration, bad input data, infrastructure service faults, MediaFirst API/data errors must be logged.

It shall be possible to remotely monitor the functional health and the utilization rate of the underlying infrastructure resources.

### File Structure

Billing Export File Format:

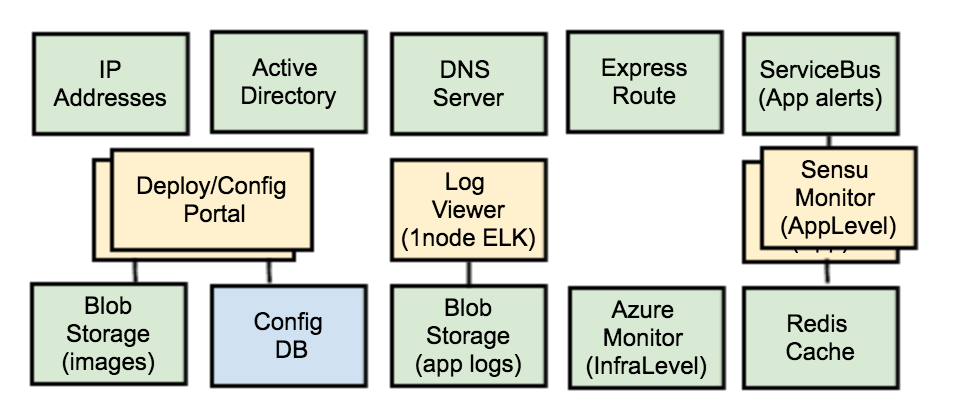
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Item  Name | Level | Item  ID | Attribute | Number of Digits | Digits in Use | Description |
| 1 | <Header> |  | | | | | |
| 1-1 | Record identifier | 01 |  | Single-byte | 1 | 1 | "H" fixed |
| 1-2 | Date of creation | 01 | CreationDate | Single-byte number | 8 | 8 | Date of file creation (in the format of YYYYMMDD) |
| 1-3 | Time of creation | 01 | CreationTime | Single-byte number | 6 | 6 | Time of file creation (in the format of HHMI24SS) |
| 1-4 | Reserved | 01 |  | － | 375 | 0 | Field for future extension |
| 2 | <Data> |  | | | | | |
| 2-1 | Record identifier | 01 |  | Single-byte | 1 | 1 | "D" fixed |
| 2-2 | Event ID | 01 | EventId | Single-byte | 10 | 10 | Unique ID assigned for charging  Used for checking redundant acquisition of the same record  Left-align, single-byte space padding to the right. |
| 2-3 | Customer ID | 01 | CustomerId | Single-byte | 20 | 8 | (house + cust). This is treated as reference.  Left-align, single-byte space padding to the right. |
| 2-5 | Site ID | 01 | SiteId | Single-byte | 10 | 2 | (Corp number) This is treated as reference.  Left-align, single-byte space padding to the right. |
| 2-6 | Date of event occurrence | 01 | EventDate | Single-byte number | 8 | 8 | Date of event order (in the format of YYYYMMDD) |
| 2-7 | Time of event occurrence | 01 | EventTime | Single-byte number | 6 | 6 | Time of event order (in the format of HHMI24SS) |
| 2-8 | Item code | 01 | ItemCode | Single-byte | 10 | 10 | Code to identify each event  Left-align, single-byte space padding to the right.  (Not used in BOS) |
| 2-9 | Item description | 01 | ItemDescription | － | 128 | 128 | Description of each event  (Not used in BOS) |
| 2-10 | Taxable amount | 01 |  | Single-byte number | 19 | 5 | Taxable amount (in the format of 99999, currency: JPY)  Right-aligned, zero padding to the left  (To be used in the case of future expansion. Not used at present.) |
| 2-11 | Consumption tax | 01 |  | Single-byte number | 19 | 5 | Consumption tax (in the format of 99999, currency: JPY)  Right-aligned, zero padding to the left  (To be used in the case of future expansion. Not used at present.) |
| 2-12 | Amount including consumption tax | 01 | Price | Single-byte number | 19 | 5 | Taxable amount + Consumption tax (in the format of 99999, currency: JPY)  Right-aligned, zero padding to the left |
| 2-13 | Provider ID | 01 | ProviderId | Single-byte | 10 | 10 | ID to identify a provider. This is allocated to a BOS broadcast operator code.  Left-align, single-byte space padding to the right. |
| 2-14 | Provider name | 01 | ProviderName | － | 50 | 50 | Setting of the provider name.  Left-align, single-byte space padding to the right. |
| 2-15 | Asset ID | 01 | ProviderTitleCode | Single-byte | 20 | 13 | Event unique ID allocated by a provider. A pair of Provider ID and Asset ID is used to uniquely identify contents.  Left-align, single-byte space padding to the right. |
| 2-16 | DSTBMAC address | 01 | SetTopId | Single-byte | 20 | 12 | DSTB Mac address. 12-digit format without delimiters.  Key item to identify customers.  Left-align, single-byte space padding to the right. |
| 2-17 | DSTBID | 01 | SerialNumber | Single-byte | 20 | 20 | DSTB ID  Left-align, single-byte space padding to the right. |
| 2-18 | Rating | 01 | Rating | － | 20 | 20 | Rating. ("PG-12", "R-15", "R-18", "R-20", etc.)  Left-align, single-byte space padding to the right. (Not used in BOS) |
| 3 | <Trailer> |  | | | | | |
| 3-1 | Record identifier | 01 |  | Single-byte | 1 | 1 | "T" (fixed) |
| 3-2 | Number of records | 01 |  | Single-byte number | 10 | 10 | Total of records in the data part  Right-aligned, zero padding to the left |
| 3-3 | Reserved | 01 |  | － | 379 | 0 | Field for future expansion  Left-align, single-byte space padding to the right. |

AD-Info Export File Format:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Item  Name | Level | Item  ID | Attribute | Number of Digits | Number of Bytes | Description |
| <Header> | |  | | | | | |
| 1 | Record section |  |  | CHAR |  | 1 | The fixed value "H" is set. |
| 2 | Record SEQ |  |  | CHAR |  | 7 | Filled by single spaces. |
| 3 | Date of creation |  |  | CHAR |  | 8 | YYYYMMDD |
| 4 | BMS code |  |  | CHAR |  | 10 | Possible to determine BMS. (Left-align, space padding to the right) |
| 5 | Linefeed Code |  |  | CHAR |  | 1 | LF |
| <Data> | |  | | | | | |
| 1 | Record section |  |  | CHAR |  | 1 | The fixed value "D" is set. |
| 2 | Record SEQ |  |  | CHAR |  | 7 | Sequential number from '0000001' |
| 3 | Banner ID |  |  | CHAR |  | 10 | (Left-aligned, space padding to the right) |
| 4 | Banner name |  |  | CHAR |  | 60 |  |
| 5 | MAC address |  |  | CHAR |  | 20 | (Left-aligned, space padding to the right, without a MAC address colon) |
| 6 | Clicked year month date hour minute second |  |  | CHAR |  | 14 | In the format of YYYYMMDDHH24MISS |
| 7 | Banner section |  |  | CHAR |  | 1 | 0: Click only, 1: With questionnaire, etc. |
| 8 | Answer 1 |  |  | CHAR |  | 20 |  |
| 9 | Answer 2 |  |  | CHAR |  | 20 |  |
| 10 | Answer 3 |  |  | CHAR |  | 20 |  |
| 11 | Answer 4 |  |  | CHAR |  | 20 |  |
| 12 | Answer 5 |  |  | CHAR |  | 20 |  |
| 13 | Answer 6 |  |  | CHAR |  | 20 |  |
| 14 | Answer 7 |  |  | CHAR |  | 20 |  |
| 15 | Answer 8 |  |  | CHAR |  | 20 |  |
| 16 | Answer 9 |  |  | CHAR |  | 20 |  |
| 17 | Answer 10 |  |  | CHAR |  | 20 |  |
| 18 | Answer 11 |  |  | CHAR |  | 20 |  |
| 19 | Answer 12 |  |  | CHAR |  | 20 |  |
| 20 | Answer 13 |  |  | CHAR |  | 20 |  |
| 21 | Answer 14 |  |  | CHAR |  | 20 |  |
| 22 | Answer 15 |  |  | CHAR |  | 20 |  |
| 23 | Answer 16 |  |  | CHAR |  | 20 |  |
| 24 | Answer 17 |  |  | CHAR |  | 20 |  |
| 25 | Answer 18 |  |  | CHAR |  | 20 |  |
| 26 | Answer 19 |  |  | CHAR |  | 20 |  |
| 27 | Answer 20 |  |  | CHAR |  | 20 |  |
| 28 | Linefeed Code |  |  | CHAR |  | 1 | LF |
| <Trailer> | |  | | | | | |
| 1 | Record section |  |  | CHAR |  | 1 | The fixed value "H" is set. |
| 2 | Record SEQ |  |  | CHAR |  | 7 | Filled by single spaces. |
| 3 | Number of data records |  |  | CHAR |  | 9 | 000000000 to 999999999  \* Zero padding at the head when the number of digits is in short. |
| 4 | Linefeed Code |  |  | CHAR |  | 1 | LF |

# Common Services

Deployment, Configuration and Monitoring capabilities applicable to all the system integration modules are captured in this section.



### Deployment

Central portal for automated orchestration of infrastructure resources and application services shall be provided.

Setup and automation of continuous integration environments shall be carried out which includes development environment, iteration planner, source control, build automation (build jobs, report generation, release promotion), release artefactory, change request, defect tracking and documentation.

Setup and automation of continuous delivery environment shall be carried out which involves continuous delivery server (electric commander or Go framework) for different cloud instances (development, integration, e2e testing, staging, migration and production).

Cloud infrastructure orchestration with controlled deployment to regulate deployment variations such as install/upgrade, steering from one version to another, allow selection of different modules/versions to install/upgrade, and variations in deployment scripts to meet the needs of different cloud instances.

Application services deployment automation for each module along with configuration of load balancing, auto scaling and wiring dependency on infrastructure services like service bus, cache, functions and storage.

### Configuration

Central portal for configuring external connectivity details and module specific policies, business rules, scheduled triggers shall be provided. This includes wireframes for all the relevant screens.

### Monitoring

Central portal for monitoring health of application services shall be provided. Default dashboard with 5 data points per module and 4 MIBs each with 6-8 data points for sunrise activation service shall be provided.

Azure monitor service will be used for infrastructure resource health and utilization monitoring with module specific dashboards.