AENA Application functionality:-s

1.Send real-time rental notifications to AENA for car park exit on rental checkouts, exchanges and enquiries that happens at Spanish airport locations. Send nearly real-time notifications on any other rental transactions like modifications on road, check-ins, post rental modifications and adjustments but excluding Canary Islands data.

1. Send monthly XML report with the summation of financial breakups of commissionable and non-commissionable charges (subfamilies and canons) as per AENA requested format. The monthly report should reconcile and include all the transactions that were sent for the month.
2. Yearly XML report with the representation of each month sum.

## Architecture Requirements

To follow the guidelines and architectural principles of AGB. Follow the high level architecture design depicted in [BRD](file:///K:\IT\Projects\Aella%20-%20Scandinavian%20Integration\AELLA%20PROJECTS\Commercial%20&%20Ops_PRJ0019662\05.%20Workstream\30_Workstream%203%20Azure%20Apps).

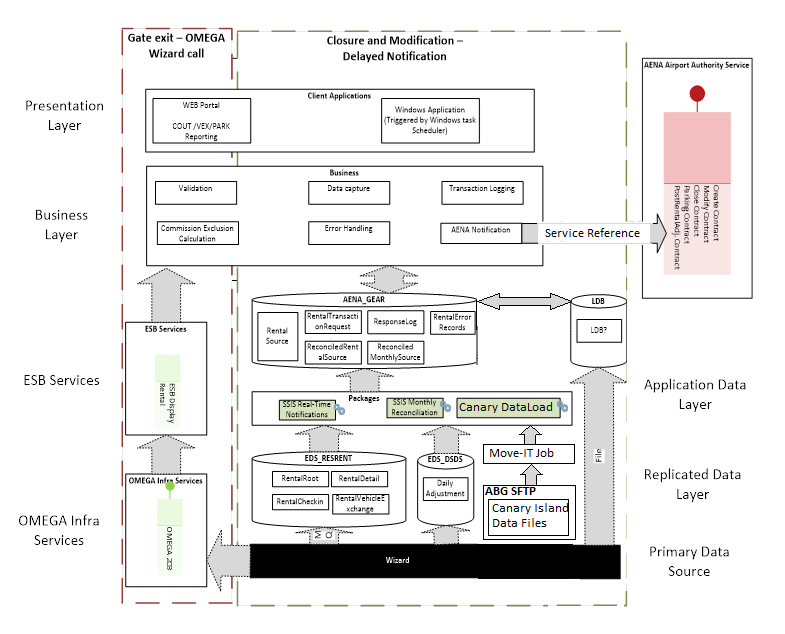
# Technical Design

## Key design considerations

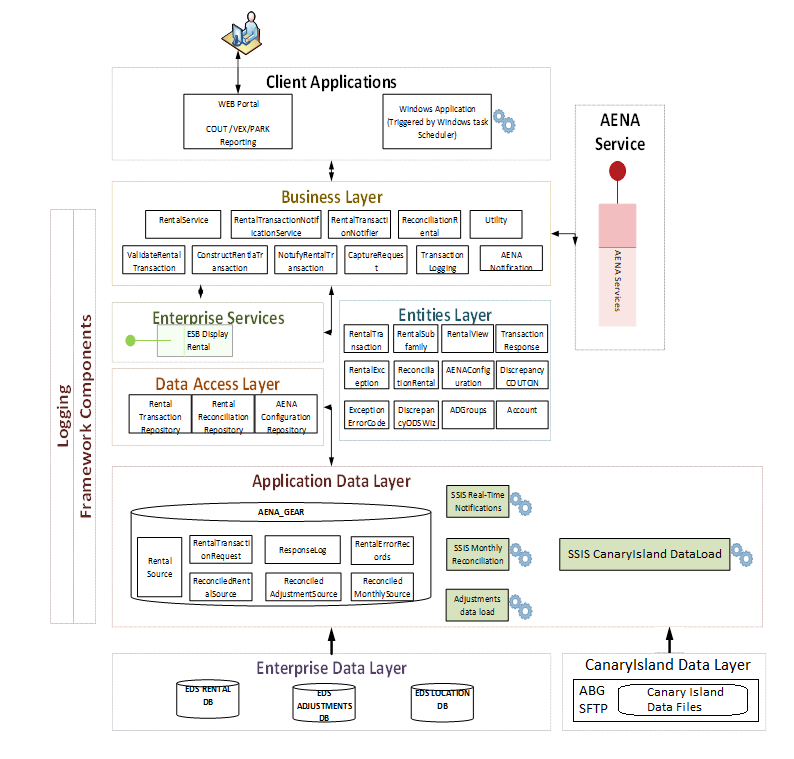
1. Interface and collaborate with existing enterprise systems and service for data
2. Clear segregation of layers for business, data and client to have minimalized change impacts
3. Future proof business layer to be hosted as independent AENA services for strategic solution
4. Clear segregation of daily notifications and monthly report for processing
5. Reduce redundant data storage

## Solution Design

#### Solution Referential View



#### Logical View



**Enterprise Data Layer:**

The application consumes nearly real-time rental data from EDS RentalRoot and RentalCheckin table. ONLY AENA required columns are retrieved from the EDS to AENA database for further processing. ONLY the data that is sent to AENA and required for monthly report are being stored in AENA database. Locations data will be loaded from EDS Locations data once that is being populated.

**Canary Island Data Layer:**

The application consumes daily rental check-ins, post rental adjustments and fleet data of Canary Islands. Data files from ABG’s SFTP will be transferred to application database server via Move-IT jobs. SSIS packages are used to load these Canary Islands data into application data layer via scheduled SQL Agent jobs. Canary Islands team is responsible for placing the files onto ABG’s SFTP, Move-IT team is responsible for copying/moving these Canary Islands data files from SFTP location to application database server.

**Application Data Layer:**

The data layer was designed broadly on three categories.

* Daily notifications (nearly real-time), sub families, response logs and exceptions: All the transactions that is being sent to AENA are being stored in the daily notification schema. The data get accumulated through Manual checkout, modification and hourly automated notifications.
* Monthly reconciled data and sub families: Holds all the Check-in, post rental modifications and adjustments data. The data is being loaded and processed once per day.
* Master data configurations to store the AENA specific configurations: All of them have audit tables to capture the changes.

**Data Access Layer:**

The data access layer consists of Entity Data Model (EDMX) bounded to it. The Data access layer is classified similar to Application Data segregation.

1. Rental Transaction Repository - Daily notifications
2. Reconciliation Repository – Monthly report
3. Aena Config. Repository – Master data configuration

All the CRUD operations were performed to the AENA database through any one of the above implementations.

**Entities Layer:**

The layer consists of common objects that could be used across different layers. Most of the objects are created in the data access layer and passed across to business layer for its processing.

**Business Layer:**

* The core business logic for AENA system exists in the layer. The layer has been designed to facilitate and future proof the AENA strategic implementation.
* The Rental Transaction Notification service handles the different types of notification. (Checkout, Modification, Check-in, Parking Stay and Post rental modification, and Adjustments).
* The Renal Transaction Notify handles the following process for all type of transactions. Each individual transaction implementations will go through the following process on iterative fashion. Notification to AENA was performed through the notify object

1. Validate the transaction
2. Construct the payload
3. Capture the request
4. Notify AENA
5. Capture the response

* Rental Service is used to fetch the rental details real-time from the ESB-OMEGA-Wizard through Framework RESTful call.
* The Reconciliation rentals and Utility is used to load the monthly and exception reports respectively in the web application. The business layer could be hosted as independent service with minimal changes

**Framework Layer:**

Common framework components for JSON serialization and RESTful service invocation through HTTPGET and HTTPPOST asynchronous implementation is handled with the layer.

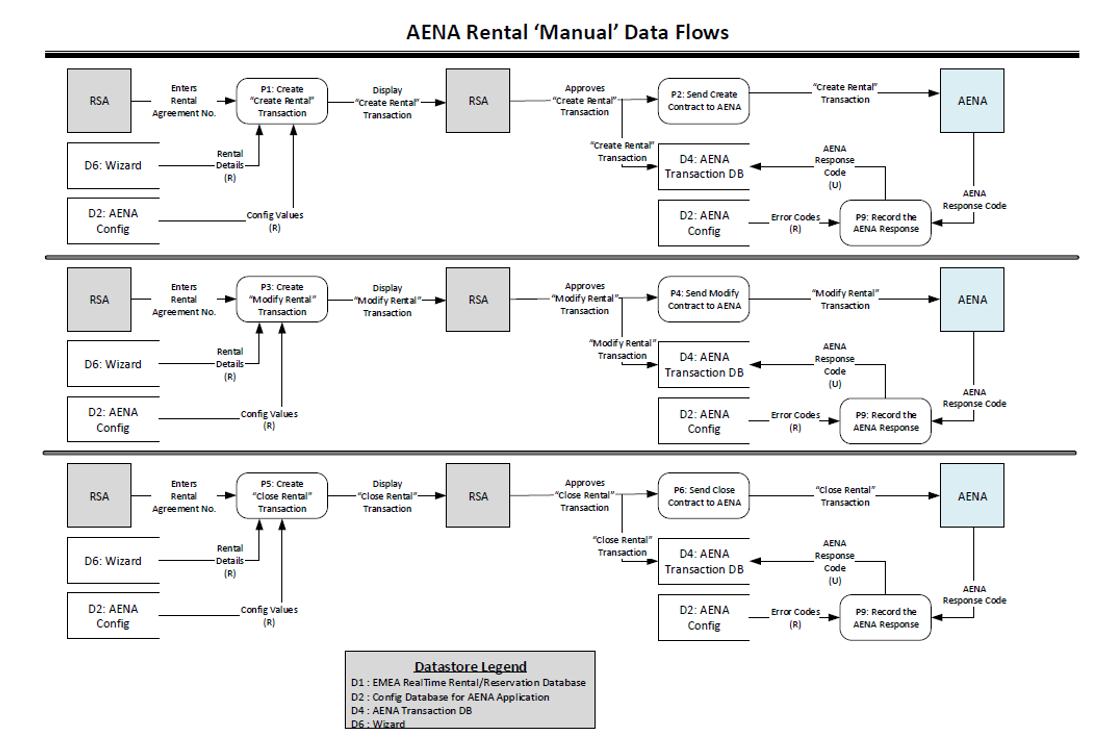
**Client Layer:**

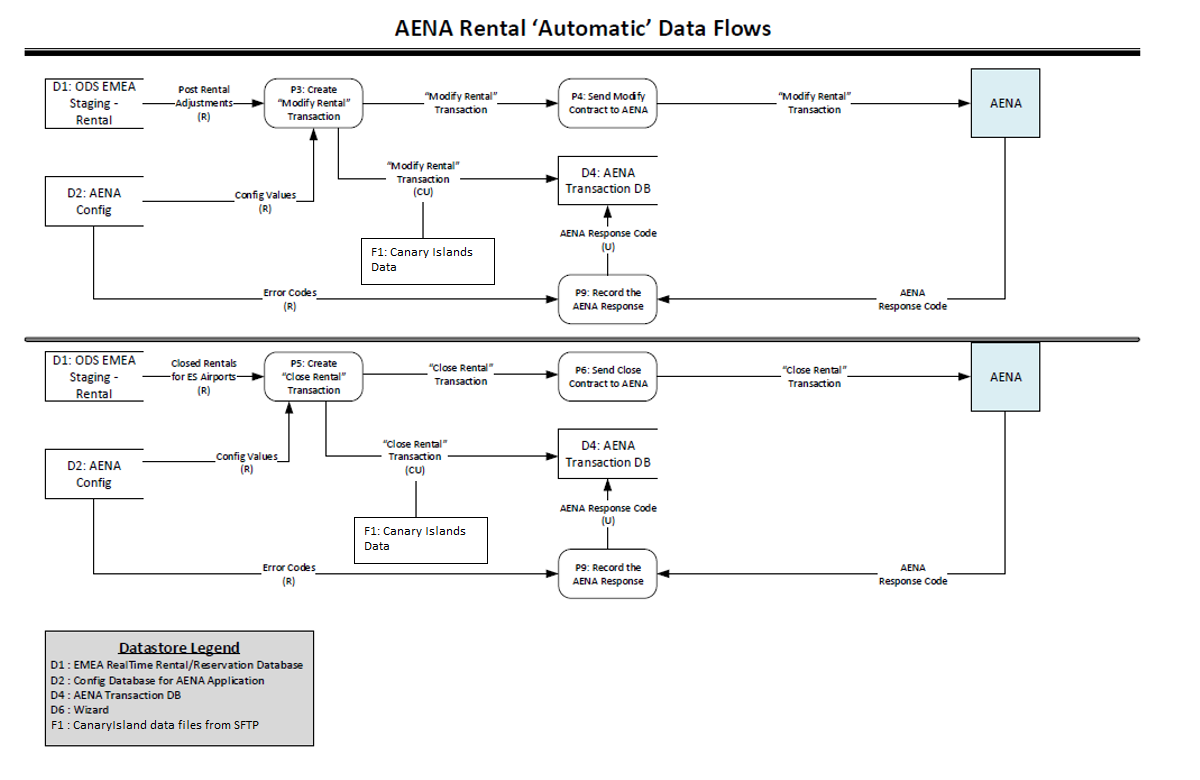
1. **Web Component**
   1. Access controls are managed through the Windows AD security groups and defined in detail on [security section](#_Security_and_Access).
   2. The web application is used to notify Checkouts, Modifications and Parking Stay. The web client will invoke the business layer for any processing and the data is fetched real-time from Wizard for notification. Except the basic validations from client layer, all the core logics will be driven through business layer. The feature will be used by Rental Station Agent.
   3. The finance and Operations department use the application to view the friendly version of Reconciled Monthly report, Yearly Report, Exceptions and discrepancies.
2. **Windows Scheduled Component**
   1. The application is scheduled every hour to notify checkout, modifications, check-in, post-rental modification and adjustment notifications to AENA. The windows client will invoke the business layer for all processing and notifications. The data is fetched from local AENA database which in turn was loaded from real-time EDS. The database has the temporary data source from Wizard for notification
   2. All the error transaction reprocessing is handled through the client. The error codes that could be reprocessed on need basis could be configured through the application config file

## Data design

#### Data flow

The data flow of different process is explained here.





|  |  |
| --- | --- |
| **Database** | **[EESDB01].[AENA]** |