***Source Code Structure,***

***Installation,***

***User Management***



*Confidential*

# Objective

Objective of this document is to review basic structure of Adrack Lead Distribution system source code, installation process and user management. The document includes description of the following components

* Layers
* Dependency schemes
* Database project
* Unit Tests
* Installation files
* Users Management

**SECTION 1**

*Confidential*

***Structure of code***

# Layers

**Target Platform**

Adrack designed under .NET Framework 4.7.2

* Patterns: Model-View-Controller
* Layered
* Visual Studio 2017 required

**Database**

* Microsoft SQL Server Web Edition

**Database project**

Adrack. Database

**Application project (set as startup project)**

Adrack. Web

**Other layers:**

Adrack. Buffering

Helper layer, which solves JSON/XML validation and basic buffering tasks

Adrack. Core

Contains based of component models, accounting, lead structures.

Adrack. Data

Core layer for data processing.

Adrack. Service

Contains services and interfaces for

* buyer channel,
* affiliate channel,
* campaign,
* reporting.
* messaging,

Implements security services.

Implements Per Request Cache Manager (caching database requests mechanism)

Runs tasks for invoice and accounting services, performed by TaskManager.cs

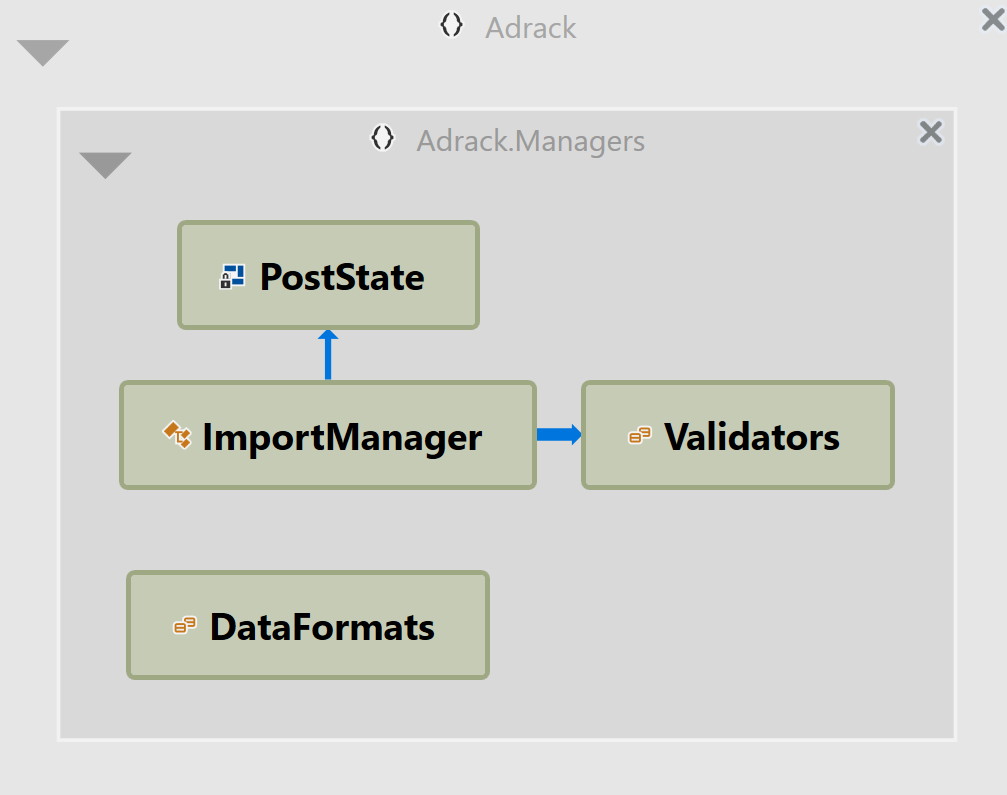
Implements caching services.

Adrack. Encryption

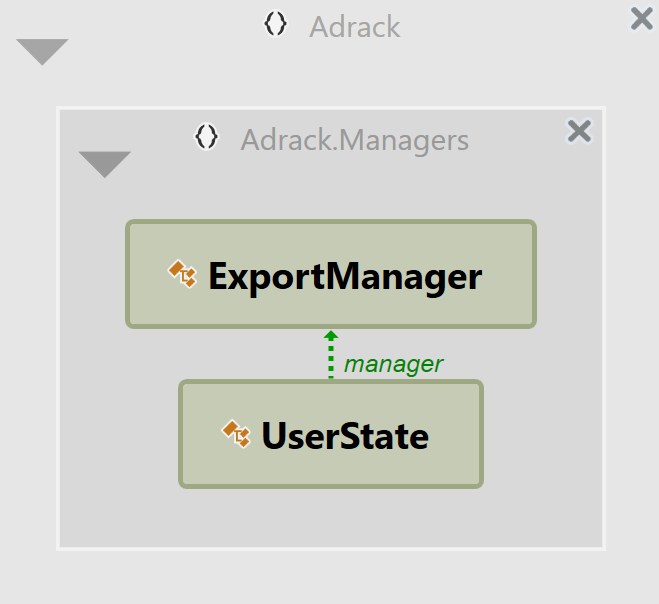
Implements encryption utilities used by project

# Processing Modules

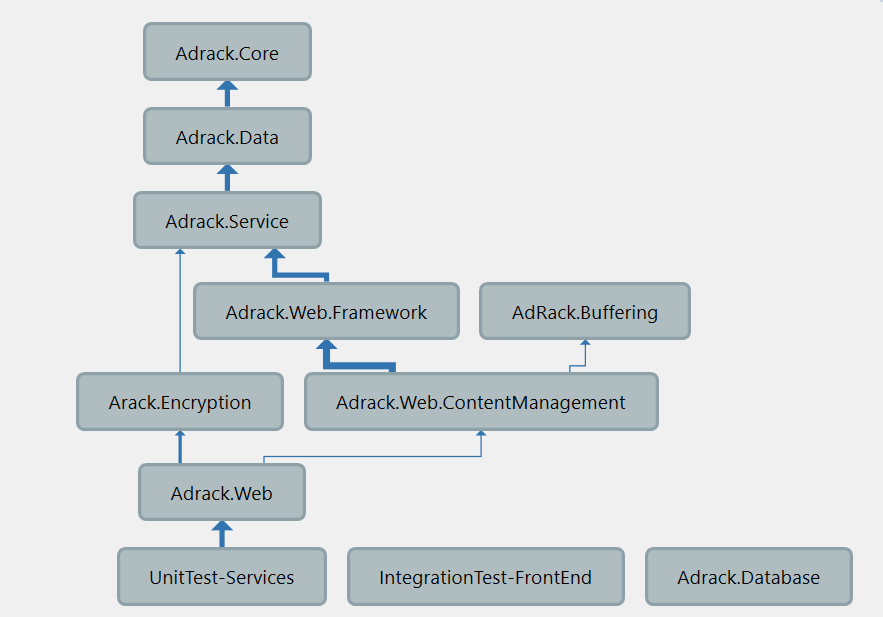
**Import Manager** – main processing module for import data from affiliates.



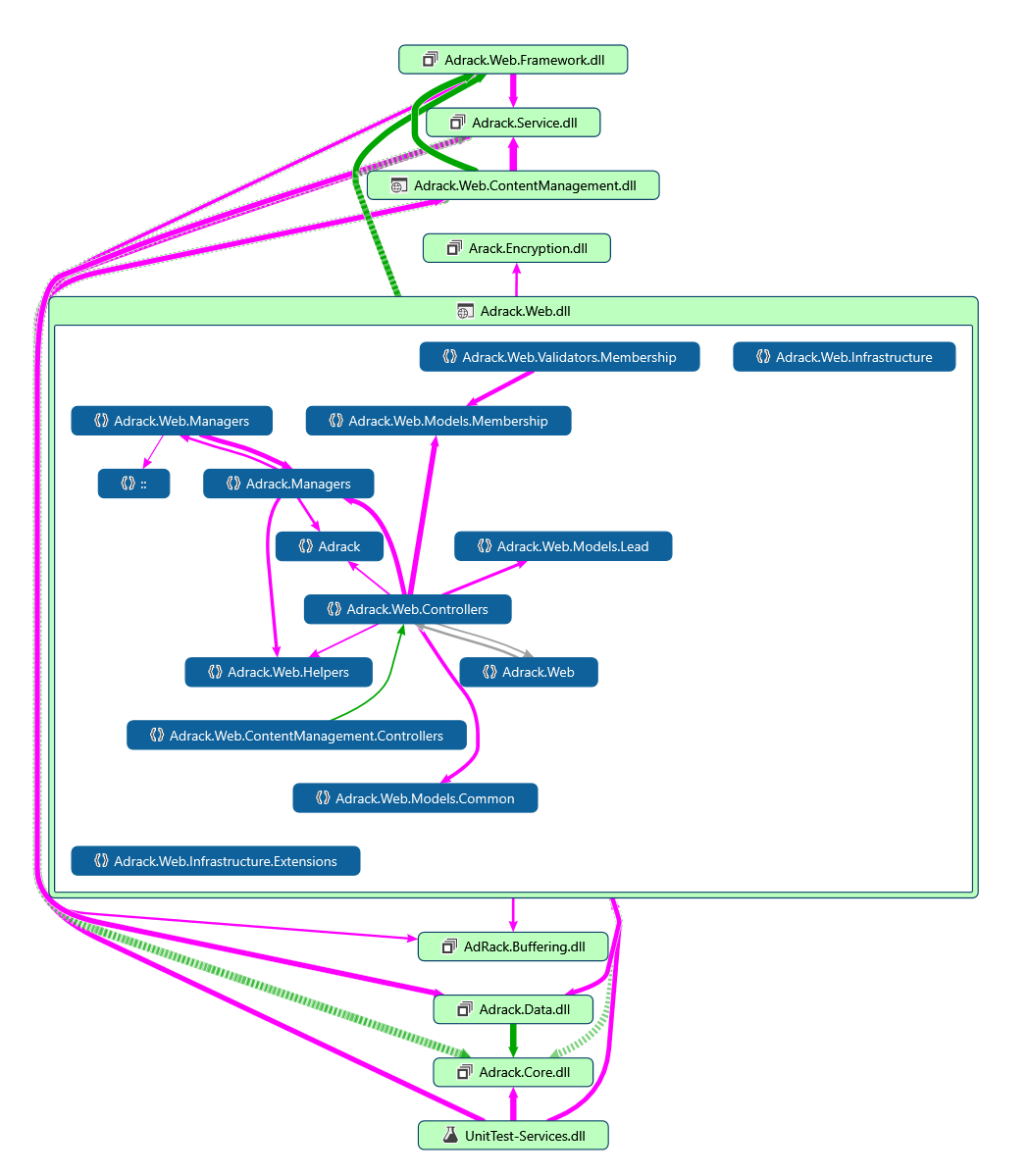
**Export Manager** – main processing module for sending data to buyers



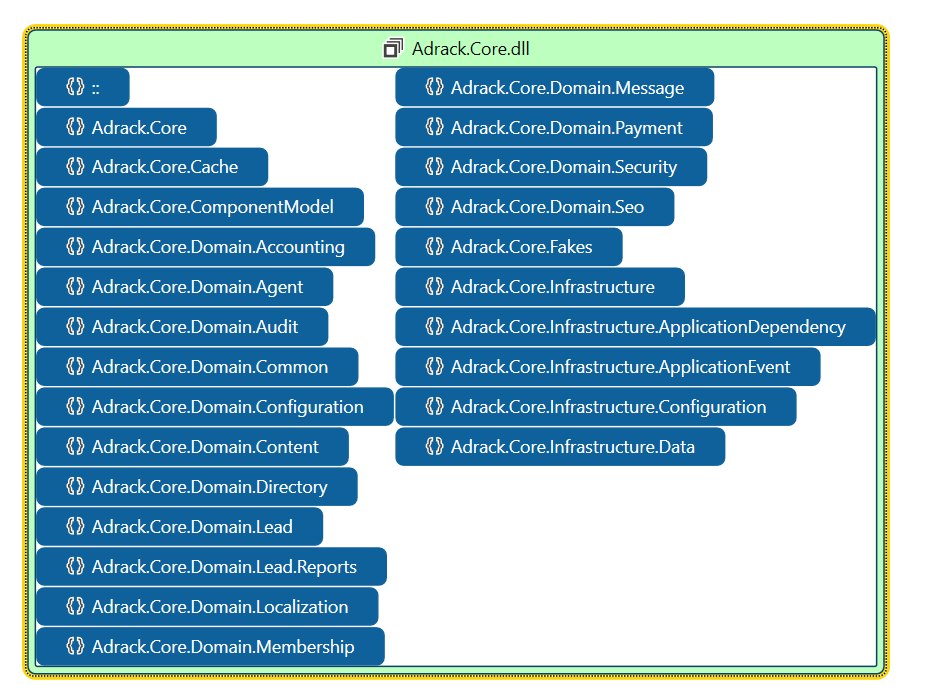
# Code Dependency Graph



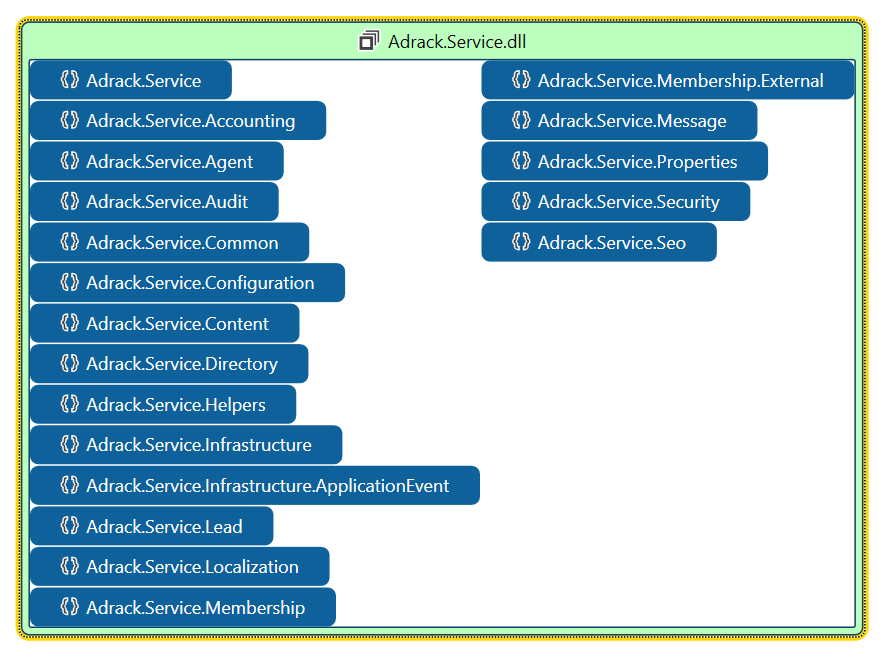
# Startup Project Code Map (Adrack. Web)



Core Code Map



# Services Code Map



# Load Processing Flow

1. Request manager (**Adrack.Managers.RequestManager**)

* Loads processing initial data (ex. MaxProcessingLeads which shows the maximum allowed processing leads at the same time)
* Checks the waiting queue for max processing leads.

1. Import manager (**Adrack.Managers.ImportManager**)

* Identifies affiliate channel from XML.
* Validates received XML data and the values.
* Validate the filters for each field if such exist.
* Writes data into the database.

1. Export manager (**Adrack.Managers.ExportManager**)
   * (executes if import manager passed the processing)

* Checks the filters for each field if exists.
* Sends the lead data to buyer channels.
* Receives and handles the response.

1. Response manager (**Adrack.Managers.ResponseManager**)

* Prepares the response and send it to affiliate.

**Lead Status**

Test – lead in test mode.

Success – lead was succeeded.

Error – error status.

Reject – the lead was rejected

Processing – the lead is in processing.

FilterError – the processing failed due to filter error

MinPriceError - the processing failed due to min price error

ScheduleError - the processing failed due to schedule error

**SECTION 2**

*Confidential*

***Installation procedure***

# Database Installation

**Database is located under the following project**

Project - Adrack. Database

To install database please generate dacpack file by building the project.

The file will be named as Adrack.Database. dacpac

Deploy this file to your SQL server and get SQL connection string.

# Database Connection Strings

Under the **Adrack. Web/App\_Data** folder you can find two files

ConnectionStringDebug.txt – put connection string for debug mode run here.

ConnectionStringRelease.txt – put connection string for release run here.

# Database Location Requirements

Database must be located under the same resource group or same farm of Adrack application.

# SSL configuration

By default SSL is not required, but you want to set required SSL please follow the instruction or use another method (Web.Config redirect, domain redirect or javascript based redirect). Please note, if you turn on SSL flag then all processing should be done via only https:// protocol.

Open AppHttpsRequirementAttribute.cs

* Find OnAuthorization method
* Uncomment the following lines

/\* //In case if SSL is required please uncomment this line

else

appSetting.SslEnabled = true;\*/

# Application Installation and deployment

Project - Adrack. Web

Build main project and deploy to server with IIS and .NET platforms installed. Latest IIS version and .NET 6.2 platforms are required to run the application.

Publishing of application should be done from Visual Studio 2017 Publish option

# Post-Deployment Compiler (Slow loading after publishing)

Adrack project uses post-deployment per-request run-time compiler and memory cache to generate binary data of all constructed pages. It means that after publishing each page of user interface will load slowly just first time. After generating compiled and cached version of your page, it will run faster it will cost low resources on server. However, this option means that after each publishing, the system will cause some delay during first opening of each page. To resolve this problem the following solution is included in project.

1. Open your project, published on VM or Azure service, to be used for user interface access (see configuring multiple VMs section below)
2. Login to Dashboard
3. Run the following file <your server url>/ CalculateCache.html
4. Wait until caching and compiling is finished.
5. Start using

# Configuring Multiple Servers

Adrack project can be configured on single server or on multiple machines for high traffic usage.

For multiple machines and high traffic usage the following configuration is suggested:

* Machine #1 SQL Server service or VM do be used with release connection string in project.
* Machine #2 – the Adrack system, deployed as user interface and user experience control machine
* Machine #3 – Adrack system for lead processing, must be deployed to separate machine and used as separate URL for importing leads.

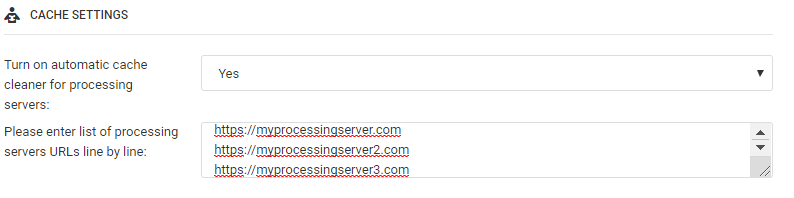
For all cases database connection string must be the same. For case of multiple machines deployment, Adrack cache mechanism must be updated. See next section.

# Adrack Memory Cache

Adrack uses improved cache system to avoid database overloading while accessing to data. Once data is accessed it stored in memory cache, until update or delete operation is performed with given data. Caching also works on lead processing. In case if data is modified then cache is recalculated. Cache always stored in memory, but in case of multiple machines usage cache will cause incorrect data reading from different machines.

To solve this problem, user interface Machine (#2) must clear cache of lead processing machines after each update. For this case please use Settings/General Settings/Cache Settings menu set “Turn ON automatic cache cleaner” flag to YES. Turn this flag to off, if you do not want to see you changes effect immediately after updating filers, buyers or campaigns. In the BOX below the flag please enter list of processing server URLs.

For example:



Processing servers can be included in load balancer, but in the list you need to put exact URL addresses of each server.

**Manual cache clearing**

The following API can be called directly from browser to clear memory cache on given server. This API can be used in case if automatic cache cleaner is turned off.

Please note: this function will NOT clear post-deployment compiled data

<server url>/Home/ClearCacheManager

# Triggered Controller Calls

Project - Adrack. Web

Contains controllers that should be called within some period, as triggers.

Generates report for first page dashboard. Without calling this function dashboard will be empty. Set this function must be called each 2 minutes.

<server url>/Home/FillMainReport?password=

This function generates invoices, should be called one time per day. For each buyer and affiliate this function will generate invoice based on invoice generation period setting.

<server url>/Accounting/GenerateInvoices?password=

# Unit Tests

Project – Unit Test-Services

**Test Database**

Install test database from Documents – Test Database Folder

Set connection strings from Unit Test-Services/App-Data folder

Set connection string to test database only. Never connect these strings to real database, in other case you will loss data.

# Integration Tests

Project - IntegrationTest-FrontEnd

Install test server on any platform

Set in UnitTestWeb Test Server property.

Run tests

**SECTION 4**

*Confidential*

***Users***

# Users Management

* **Network users, Affiliates and Buyers can be setup in the system.**
* **Users management-sign in via one interface**
* **Users permissions management**

There are 3 types of users - Network users, Affiliate users, Buyer users.

The client (Arrowshade) becomes the system administrator who has full access to everything. He can add users (Managers, Accountant etc.) from his company. For adding users, he should fill out their info (First name, Last Name, Email Address, Phone number, Department, Role/Group, Notes) and the user gets a link by email, which brings him to the system log in page but requires to choose a password (and security question probably) before the first log in.

**Administrator (full access)**

Login: Admin

Default password:

**Network (our system users)**

Similar to admin users

**User Access with Permissions**

Manage user's access by defining which permissions the users have.

When a user is being added it should be assigned to one of the Roles. We will have default Roles built in in the system. It will be Administrator, Account Manager, Affiliate Manager, Buyer Manager and Accounting, Compliance. Additional Roles can be created.

The roles and its permission’s document will be created additionally.

**More admins** - Admin can create additional Admins and all other types of users.

**Supervisors**

* Has access to both buyers and affiliates but with less access then admin. Can create campaigns, b-channels, a-channels, see lead and lead 2 (XML responses), be able to credit leads, be able to create buyer managers, affiliate managers (not able to create accounting login, not able to adjust invoices).
* Able to create affiliate manager and buyer manager (account managers)

**Buyer Manager**

* Create New Buyer and Buyer Channels
* See leads (not able to credit leads)
* See lead sale price
* See all buyer reports for buyers assigned to that buyer manager
* Be able to pause Buyer Channels

**Affiliate Manager**

* Create/register new affiliate and Affiliate Channel
* see leads (not able to credit leads)
* See affiliate earning for the lead (not the lead sale price), not able to see buyer name or Buyer Channel name
* See all affiliates assigned to that affiliate manager, and related reports

**Note**: The same user can be Buyer Manager and Affiliate Manager at the same time

**Accounting**

* Accounting Role is created by Admin only
* Access to invoicing and payments
* Have access to both affiliate and buyer reports
* Does NOT see leads, does not have the ability to credit leads

**Compliance**

Access to affiliate and buyer accounts (not Affiliate Channels or Buyer Channels) to view company information only.

**Customization and clone**

Any access level can be customized and additional network user custom access role can be created in the system with ability to clone and customize

**Automated Verification by email**

During login enable users to have to verify their account and it will be automatically approved. Verification scenario can be configured in system settings.

* Verification is required when user is trying to login from different IP (there can be whitelisting for specific users IP range), device and browser.

**Users Monitoring**

* Analyze User login by the IP, device and browser – Track and analyze user’s logins by their IP, device and browser.
* Block and Disable Affiliates and Buyers at any time
* Instantly Disable User Accounts – Network admins/managers can instantly disable any network, buyer or affiliate user account.
* Campaign Monitoring Thresholds and Alerts – Get alerted if Affiliates or Buyers update their account information.

**Notifications and email**

To have a tool for sending a notification to all users or to group of users.

To have a tool (email and/or notification) to send to network users our alerts to groups of users or specific users about offer updates, company news, etc.

**Username and password**

* When user types the username and/or password wrong it should show something as the example below.

**Password requirements**

Here are the requirements:

* It must be 8 to 16 characters.
* A combination of upper- and lower-case letters.
* At least one number or symbol, no spaces.
* While changing the password it shouldn’t allow to use the old password as a new password. It must show a message that the password can't be changed because no changes have been made.
* It should never allow the use of last 5 passwords.
* When there is something wrong in Password Change process, it must show a message with exact reason, for example - the current password doesn't match, or, the new password is invalid (and show the requirements next to that filed or bellow that field for example). If any filed is left empty it should make that field red color and show a message in red below.
* When the password is changed the user should get an email with a message that tells the password has been changed, and in case if the password change hasn’t been done by the user, he/she should take an action and contact to the administrator.