d

TOne Architecture

Contents

[Document Purpose 2](#_Toc418614260)

[High-Level Architecture 3](#_Toc418614261)

[System Layers 3](#_Toc418614262)

[Technical Modules 4](#_Toc418614263)

[Projects Structure 4](#_Toc418614264)

[Module-Level Projects 4](#_Toc418614265)

[Common Projects 5](#_Toc418614266)

[Routing Module 7](#_Toc418614267)

# Document Purpose

The purpose of this document is to discuss the high-level architecture of the new TOne. This consists of:

* High Level Architecture
* TOne Technical Modules and their main components

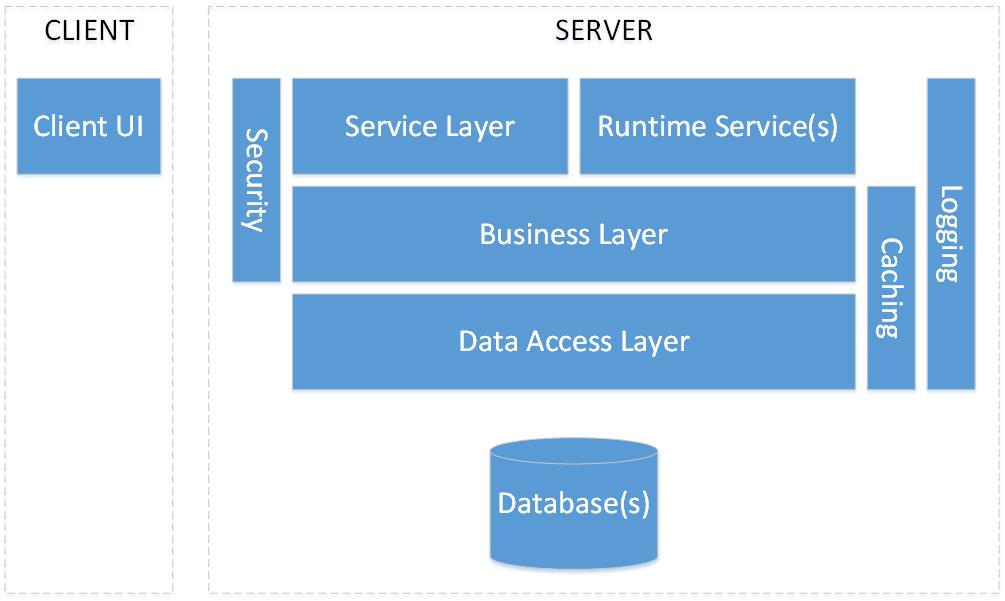
# High-Level Architecture

## System Layers

We use the following methodologies in our development process:

* Two Tiers design: Client-Server
* 3 layers architecture: Data, Business, Service

The following diagram shows the different components that would be involved in the development process of the system



This table gives a brief description about each component:

|  |  |
| --- | --- |
| Component | Description |
| Database(s) | The data store of the system data (e.g. SQL, Oracle…) |
| Data Access Layer | This component is responsible for all kind of data access. It interacts with the Database(s) and provide needed operations to upper layer |
| Business Layer | All business logic in the system should be implemented in this layer |
| Service Layer | This layer provides a set of services (e.g. web services) that allows other components (e.g. Client UI) and other systems to interact with our system |
| Runtime Service(s) | This component consists of hosting the system services that need to be always running or run based on schedules. Those services would perform system jobs and process queued requests |
| Caching | This is a crosscutting component gives caching services to the Data and Business layers |
| Logging | This is a crosscutting component gives logging/tracing/exception handling services to all other layers |
| Security | The security layer applies required security mechanism to the Service and Business layers. This includes Authentication, Authorization, Encryption/Decryption |
| Client UI | This consist of the User Interface component(s). this should interact with the system through the service layer only (after applying the security mechanism) |

Later sections in this document describe the responsibilities and base implementations of the above components in further details

## Technical Modules

The system is composed of the following technical modules:

|  |  |  |
| --- | --- | --- |
| Module | Description | Dependent On |
| Main Module | The base module of the system. It includes common entities, system security, system configuration |  |
| Business Entity | It includes functionalities of: Carrier Account, Code, Zone, Rate, Switches | Main Module |
| LCR | Routing and LCR module | Main Module, Business Entity |
| CDR | Traffic and Billing module | Main Module, Business Entity |
| Analytics | Generating reports/dashboard from the Statistic tables | Main Module, Business Entity |
| BI | Generating reports/dashboard from the BI database | Main Module, Business Entity |
| Web Config | It includes functionalities to manage the Web User Interface (e.g. Pages, Menu...) | Main Module |

## Projects Structure

### Module-Level Projects

Each module in the system should have at least the following visual studio projects:

|  |  |  |
| --- | --- | --- |
| Project | Example Name | Description |
| Entities | TOne.LCR.Entities | * It should include C# classes of all entities of the module * The classes should only have properties (no methods, no business logic) |
| Data Definition Project | TOne.LCR.Data | * It should include C# interfaces that define all needed operations to the database |
| Data Implementation SQL Project | TOne.LCR.Data.SQL | * It should include data managers that retrieve data from the data store and convert them to business entities; and that manipulate data in the data store. Each data manager is an implementation of an interface in the “Data Definition Project” against the SQL database |
| Business Project | TOne.LCR.Business | * It should include business managers that manipulate the business entities of the module and apply business logic |

If the module includes workflows, the following projects would be also available:

|  |  |  |
| --- | --- | --- |
| Project | Example Name | Description |
| Workflow | TOne.LCRProcess | * It should include all the workflows of the module |
| Activities | TOne.LCRProcess.Activities | * It should include the WF activities used for the workflows of the module |
| Arguments | TOne.LCRProcess.Arguments | * It should include the classes that define the input and output arguments of each workflow |

If the module includes user interfaces on the web, the following projects would be available:

|  |  |  |
| --- | --- | --- |
| Project | Example Name | Description |
| Web | TOne.LCR.Web | * It should include all the WebAPI controllers of the module (e.g. RoutingController) * It should include all the web user interface for the module:   + HTML Views   + AngularJS Controllers   + AngularJS Services, Directives (if any)… |

### Common Projects

The following table shows the common visual studio projects:

|  |  |
| --- | --- |
| Project | Description |
| TOne.Web | The main web project that should be configured as a web site on IIS.   * This project has the base Client files   + External library files: AngularJS, Bootstrap, highChart..   + AngularJS main modules   + Common directives: DataGrid, Select, Button..   + Styles.css   + … * It has Web configuration file * It includes Routing configuration and WebAPI Security implementation * It includes references to all system dlls needed for the web. This includes Entities, Data, Data.SQL, Business, and Web * It includes references to Vanrise Framework DLLs * It includes references to needed external libraries * Each web project of the technical modules has a folder named “Client”. This folder is registered as a virtual directory under Client\Modules |
| TOne.RuntimeService | This project is of type Windows Service. Its main job is to host all system runtime services (Workflow, Queuing…).   * It includes references to all system dlls needed for the services. This includes Entities, Data, Data.SQL, Business, and Workflow (BP, Activities, Arguments) * It includes references to Vanrise Framework DLLs * It includes references to needed external libraries |
| TOne.Runtime | This project includes the implementation of the MainService that is hosted in the TOne.RuntimeService |
| TestRuntime | This is a console application used in the development environment instead of the TOne.RuntimeService. It is temporary |
| TOne.Caching | It includes an implementation of the CacheManager used for caching |

# Routing Module

The Routing Module is available in a separate technical document “TOne Routing”