|  |  |  |
| --- | --- | --- |
|  | |  |
|  | | Engineering Design Document |
|  | |  |
|  | Integration of the Security Framework into MetraNet  **Version: 1.0**  **Last Revised:**  **Author: Anatoliy Lokshin** | |

Table of Contents

[1 System Overview 4](#_Toc284420766)

[2 Assumptions and Dependencies 4](#_Toc284420767)

[3 Functional Design 4](#_Toc284420768)

[4 Error list 6](#_Toc284420769)

[5 Outstanding Issues 6](#_Toc284420770)

Document Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Revision No.** | **Date** | **Description of Change** |
| **Anatoliy Lokshin** | 1.0 | 02/02/2010 | First revision |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Functional Design Review

| **Name** | **Title** | **Date Reviewed** | **Date Approved** |
| --- | --- | --- | --- |
| **Kyle Quest** | Architect |  |  |
| **Anatoliy Lokshin** | Development lead |  |  |
| **Yulia Kuchmai** | QA Representative |  |  |
|  |  |  |  |

Technical Design Review

| **Name** | **Title** | **Date Reviewed** | **Date Approved** |
| --- | --- | --- | --- |
| **Kyle Quest** | Architect |  |  |
|  |  |  |  |
|  |  |  |  |

Engineering Task Worklist Review

| **Name** | **Title** | **Date Reviewed** | **Date Approved** |
| --- | --- | --- | --- |
| **Anatoliy Lokshin** | Development lead |  |  |
|  |  |  |  |
|  |  |  |  |

References

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Author** | **Location** |
| SecurityFrameworkConfigurationIntegrationDesign.docx | Viktor Grytsay |  |
|  |  |  |

Version Configuration

|  |  |  |
| --- | --- | --- |
| **Name** | **Version** | **Additional Comments** |
| MetraNet | 6.4.0 |  |
|  |  |  |

# System Overview

The integration goal is to have all security related code in a single place. This place will be the Security Framework module. To facilitate this, some code in the current security modules will be moved to the Security Framework. This document describes which code will be moved and where.

# Assumptions and Dependencies

All cryptography code will be placed into the Security Framework and implemented as a set of Security Framework engines. The engines can be used from within MetraNet as directly as thru a wrapper. The wrapper will be made from current implementation represented by MetraTech.Security and MetraTech.Security.Crypto modules.

# Functional Design

At the high level the planned changes are represented on the Figure 1.

Also all of the cryptographic code will be placed directly to the Encryptor subsystem of the Security Framework. It will be implemented as a set of engines that is not described in details by this document and will be defined during the development.



Figure 1. New modules and migration of significant classes.

There are types (classes and enumerations) in the current implementation used from within the MetraNet code and will be used by the Security Framework too. Those types will be moved into a separate assembly – MetraTech.Security.Common. It will be referenced by both the MetraNet and the Security Framework.

At the same time the configuration loader will be separated from the Security Framework into its own module MetraTech.SecurityFramework.Configuration. This module will be used by both the Security Framework and the module with common classes.

Configuration loader will be improved to support existing MetraNet security module configuration. This task is described in the design document on the configuration loader.

Unit tests will be redesigned to be able to run from the same location where the common MetraNet compilation output is placed. Unit tests configuration will be copied into subfolder during compilation to avoid interferention with a normal Security Framework configuration files.

The MetraTech.Security assembly will only contain the Auth and PasswordManager classes and the LoginStatus enum. The MetraTech.Security.Crypto assembly will only contain the CryptoManager and CryptoInstall classes and the enums shown in the figure below.



The SecurityFramework is initialized in the Initialize(..) method of the Auth and PasswordManager class. The initialization code shown in the figure assumes that the SecurityFramework is a singleton. The appropriate initialization code can be substituted with the assumption that multiple calls to the initialization code does not repeat the process. One part of the SecurityFramework initialization process, is the initialization of the crypto system based on the current initialization code in the Initialize() method in MetraTech.Security.CryptoManager.

The Auth, PasswordManager, CryptoManager and CryptoInstall initialize the SecurityFramework and store a reference to the SecurityFramework and forward calls to it from their public API’s as shown in the figure.

# Error list

The following table shows a listing of errors that can occur.

|  |  |  |  |
| --- | --- | --- | --- |
| **Error Code** | **Error Message** | **Description** | **Area** |
|  |  |  |  |
|  |  |  |  |

# Outstanding Issues

List all open issues regarding this document.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Date raised** | **Description and Resolution** | **Page/ Section** | **Raised by** | **Allocated to** | **Status** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |