**Medtronic MITQ**

**Research & Development**

**High Level Architecture Document**

**Common Client v1.0**

**RE00026722 Rev B**

|  |  |  |  |
| --- | --- | --- | --- |
| Issue Date: Refer to Agile | | | Page 1 of 20 |
|  | | | |
| Revision History: | | | |
| Revision | SDA | Description | |
| A | RC026977 | Initial Release | |
| B | RC033270 | Updates to include authentication workflow information | |
|  |  |  | |
|  |  | | |
| Author: Refer to Agile for approvals | | | |
| Maverick Zhang – Insigma US [Manual Approval] | | | |
|  | | | |
| **Approvals:** Refer to Agile for approvals | | | |
| Ashwini Pandey – MI Product Owner | | | |
| *Paul Ourada – MI System Architect* | | | |
| *Shawn Li- MI Tech Consultant* | | | |
| *Robert Boyer – VS System Architect* | | | |

Table of Contents

[1. INTRODUCTION 3](#_Toc436842421)

[1.1 Purpose 3](#_Toc436842422)

[1.2 Scope 3](#_Toc436842423)

[2. REFERENCED DOCUMENTS 3](#_Toc436842424)

[3. DEFINATIONS AND ACRONYMS 3](#_Toc436842425)

[4. ASSUMPTION 3](#_Toc436842426)

[5. ARCHITECTURE OVERVIEW 4](#_Toc436842427)

[6. MODULARIZATION 4](#_Toc436842428)

[6.1 Biomed App 5](#_Toc436842429)

[6.1.1 Online Help 6](#_Toc436842430)

[6.2 DMM 6](#_Toc436842431)

[6.3 CC Timer 8](#_Toc436842432)

[7. ATTRIBUTES 9](#_Toc436842433)

[7.1 Interface & communication 9](#_Toc436842434)

[7.2 Deployment 9](#_Toc436842435)

[7.3 Availability 9](#_Toc436842436)

[7.4 Security 9](#_Toc436842437)

[7.5 Performance 10](#_Toc436842438)

[7.6 Scalability 10](#_Toc436842439)

[7.7 Maintainability 10](#_Toc436842440)

[7.8 Supportability 10](#_Toc436842441)

[7.9 PHI, PII and HIPAA 11](#_Toc436842442)

[8. AUTOMATIC TESTING 11](#_Toc436842443)

[8.1 Development Phase 11](#_Toc436842444)

[8.2 Integration Test/Regression Test Phase 11](#_Toc436842445)

[8.3 Exception 11](#_Toc436842446)

[9. REQUIRED THIRD PARTY SOFTWARE/COMPONENTS 11](#_Toc436842447)

[10. PRODUCTION RELEASE PROCEDURE 11](#_Toc436842448)

[11. PRODUCTION SUPPORT & ESCLATION PATH 12](#_Toc436842449)

[12. Appendix A: Benchmark of SignalR 12](#_Toc436842450)

[13. Appendix B: User Authentication Sequence Diagram 12](#_Toc436842451)

[13.1 User Authentication Sequence Diagram – Standalone Mode 12](#_Toc436842452)

[13.2 User Authentication Sequence Diagram – Centralized Deployment Mode 13](#_Toc436842453)

# INTRODUCTION

## Purpose

This document describes the system architecture and high level design of the software. It identifies the architectural components and software items and the interfaces and communication mechanisms between the software items.

## Scope

The scope of this document is describing architecture of the Common Client software from overview of architecture.

This document includes the following components for the Common Client architecture:

Assumption

Requirement coverage Matrix

Architecture Overview

Modularization

Attributes

Required 3rd party software/components

Production release procedure

Production support & escalation path

# REFERENCED DOCUMENTS

[1] RE00026719 System Requirements Specification CC v1.0, Rev A

# DEFINATIONS AND ACRONYMS

| Term or Abbreviation | Description |
| --- | --- |
| CC | Common Client |
| DMM | Data Management Module, A core component of CC |
| GDMP | Gateway Device Management Platform |
| GDMPA | Gateway Device Management Platform Agent |
| VS | Vital Sync. A Medtronic remote monitoring platform. |
| LB | Load Balance |
| DPAPI | Data Protection API. It is .Net framework API, it will use the user specific information to generate a key and protect the data in memory |
| ROLE | A set of user privileges to define what feature/function user can be used. |
| RIGHT | Individual access privilege with attributes, such as read-only, editable, or none |

# ASSUMPTION

CC will be designed under the following assumptions:

1. CC will be running on Windows platform.
2. CC is a pure operation GUI, it will not store any data or file.
3. CC needs to support single and load balance (LB) cluster deployment.

# ARCHITECTURE OVERVIEW

The CC is a software module that integrates the Vital Sync remote monitoring platform (VS) and the Gateway Device Management Platform (GDMP). The diagram below illustrates the high level components and their association.



Figure 1 Common Client Architecture Overview

* **Common Client**: This is browser based application, it is responsible for pulling and pushing data to the Vital Sync/GDMP for its supported devices.
* **GDMP Agent (GDMPA):** This is the software that manages communication between the GDMP server and the Common Client Application.
* **Vital Sync (VS)**: Undertaken the device operation, including apply software package, feature license file to device, collect device log and all other Common client required operation to devices.

# MODULARIZATION

CC consists of three components, they are Biomed App, Device Management Module (DMM) and CC Timer. The following diagram illustrate the overview of modularization.



Figure 2Modularization View

## Biomed App

Biomed App is a session based front-end application, represented by HTML-5 page, handle all operations related to user interactive, including collects user inputs, and displays the results. This application uses the bootstrap to handle look & feel and Angular JS as the Javascript library to process the communication with DMM. This browser-based UI application will be running under IIS runtime environment. It will follow single-page design pattern and rights driven.

The diagram below illustrates the component view of Biomed App.



Figure 3: Biomed Component

**Single-page Application**: Biomed App is a single-page application (SPA) which provides no-refresh experience. It will use AngularJS as the front-end MVC framework. There will be a single index view and serval partial views. The partial views will be loaded on the time it is necessary.

**No-refresh Interactions**: The interaction between browser and server are through Ajax calls. The back-end will use ASP.NET MVC Web API which returns data in JSON format. Controllers of AngularJS will receive the data and updating the view.

**Mobility display ready:** Biomed App will use Bootstrap as the UI layout framework. The views would be responsive to different screen size, including smart phone and tablet screen. The views will be uniformly accessible across devices. It provides the same functionalities for different screen sizes, however the layout may change.

**Server-push message:** Biomed App will use SignalR as the server-push message framework. The server-push message enables server to update views proactively, such as:

* Device status change: when the device is connected to the system, the web views need to be notified.
* Displaying process completion: show the percentage of completion to the long duration processing, such as transfer software package to VS.
* Handle asynchronous calls: the views need to be updated upon a response is received.

### Online Help

Online help contains the usage of CC as well as other useful information related to device operations such as procedure of the software upgrades.

It is also HTML-5 based pages and provides the searchable features as same as CC.

## DMM

DMM will be packaged as a shared dynamic linked library, and it will be loaded by Biomed App within IIS environment. It will provide the RESTFul webservice interface to outside world. DMM will work as request/response approach to align with VS behavior.



Figure 4: DMM component

**REST Handler**: this component handles the physical REST request/response for all inbound and outbound.

**Business Service**: Business related services

* **Rights Generator**: Use the responded entitlements from VS to generate the rights, these rights will be used to generate the pages and to enable/disable certain feature while loading the page.
* **Sign On and Session Management**: this is the first module when every user trying to use Biomad App. This module manages user sign on and session.
* **Device Management**: this module handles the operation related to devices, such as obtain the device list from VS and GDMPA, list/search/filter a device, also, handle checking the device status and enable/disable the operation button on the page.
* **Configuration Management**: this module handles the operation to device configuration, including obtains configuration from GDMP and VS, compare the configuration, sync configuration to GDMPA.
* **Software Management**: this module is responsible for upgrading software package, checking result and showing progress.
* **Feature License Management**: this module handles the operations related to feature license, including push feature license to VS, display results, obtain and display feature license list from VS, and obtain a feature license file from GDMPA.
* **Notification**: all notifications and alerts will managed by this module. Once the notification available, it will publish the notification to all connected sessions.
* **Log File Management**: this module will be used to obtain, view, and upload log files as well as set up schedule to obtain the log file in recurrent fashion.
* **Schedule Job Management**: this module will be used to setup, display the schedule jobs.

**Common Service:** Systematic related service.

* **Sign on & Entitlement Service:** handle sign on to the VS and doing the shadow sign on to GDMPA upon sign on to VS successfully, it will collect all entitlements of the particular user from both system and map them into a set of rights, these rights will return to Biomed App for page generation propose.
* **Data Operation Service**: this service handles all requests related to data transfer, including log files, software package, device configuration and feature license file. When sending data out, this service will chunk the large size of data into a smaller size to fit the receiver capacity. And return the percentage of progress to the **Notification** module.
* **Resource Manager**: Manage all configuration, multi-language support, and resource file management.
* **Exception Handler**: all run time exception will handled by this component and it will work with Resource Manager to support mapping from error code to error message.
* **Metadata Service**: all metadata will be managed by this component and cached in memory, also, this module will proactively sync with GDMPA and VS for metadata updates.

**Utility**: Static type utility classes.

* **REST Message Parser**: This utility handles parse REST messages from/to VS and GDMPA.
* **Business Rule Handler**: It is responsible for handling business rule for the current CC when processing any request and responses.
* **Scheduler Generator**: This utility will be used to generate scheduled tasks. And it will recurrently to check results from VS by a certain time period (e.g., once a day).
* **Data Protector**: All data protection related feature goes to this utility, including a Data Protection API (DPAPI) wrapper, Key/Cert management related functions.

**Configuration File**: the file contains all non-sensitive information related to CC runtime, such as endpoint of VS and GDMPA.

**Audit Trail**: this is a mandatory service. It will record every footprint of operation, and the audit log will eventually submit to GDMPA via the same mechanism which CC communicates with GDMPA.

## CC Timer

CC timer is used to trigger particular command to CC when pre-defined check time occurs. It includes two components.

* **Job Scheduler**: it uses Windows Task Scheduler to execute CC Trigger when pre-defined check time occurs.
* **CC Trigger**: it is an executable application that is able to send request to DMM through web service over the secured communication channel. It is an extensible service and is able to send different requests to DMM based on configuration file.

# ATTRIBUTES

## Interface & communication

RESTFul Webservice will be the interface for communication between CC and VS, Socket will be used between CC and GDMPA.

## Deployment

The CC intended to support three major deployment approaches, they are load balancing based on IIS, load balancing based on hardware (BIG-IP), and single node deployment.

To support load balancing based on IIS, Application Request Routing (ARR) is required. (Refer

http://www.iis.net/learn/extensions/configuring-application-request-routing-(arr)/http-load-balancing-using-application-request-routing for details).

To support load balancing based on hardware, setup the hardware LB, and deploy CC to two separate IIS environments, configure the hardware LB to use session based routing mechanism instead of round-robin.

To support single node deployment, just simply deploy the CC to one IIS.

## Availability

The CC will be available to the same level of service as the greater of GDMP and Vital Sync for centralized deployment scenario.

## Security

When implementing the CC, the following security guide must be enforced:

|  |  |
| --- | --- |
| Category | Approach |
| Data validation | * Data must be validated before use no matter where data comes from * Validate input but not limited to on length, range, format and type, CRC, hash code * Constrain, reject and sanitize the input |
| Authentication | * No credential stored in application, even in memory |
| Authorization | * Use “least privilege” policy * RIGHT based design |
| Configuration | * Never use local security authority (LSA) * file must be stored under protection of Windows EFS |
| Sensitive Data | * Data must be encrypted any time. * If the temp file a must, then the folder must be protected by Windows EFS * If the temp file a must, then the file must be destroyed immediately after use * The Data Protection API (DPAPI) of .Net framework must be enforced when unencrypted data in memory |
| Communication channel | * HTTP/SSL must be enforced when doing RESTFul webservice calls between components |
| Session | * User must sign on before use the CC application * Use short period for session time out (i.e. 5 minutes by default) * The CC will reject sign on request if the same user already signed on * No sensitive data stored in session store * Secure the channel to the session store |
| Cryptography | * When using the certificate, it must be per user bases * Certificate must be stored under current user store * The private key cannot be exported |
| Exception | * Exception must be handled by structured blocks (try/cache) * Catch and wrap the exception only when the operation adds value/information * Do not expose any information related to sensitive and system in exception * Do not log private data |
| Auditing and Logging | * Identify malicious behavior (i.e. user try to access the unauthorized feature/function) * Monitoring the access traffic, and statistic them. Report DoS |
| Certificate | * Different channel must use different certification for SSL |

## Performance

CC is an IIS based application; IIS will manage the balance between the concurrent requests and system resource. CC is also a lightweight process application. The heavy processes are in VS and GDMPA; the performance of CC is performance of VS and GDMPA essentially. However, the KPI for performance must meet the following indicators:

1. Concurrent active users: 25+
2. Connected device list retrieve: 500+

## Scalability

CC supports the same scalability as Vital Sync.

## Maintainability

The CC will embody maintainability through these means:

* Use of the Vital Sync Light
* Data driven design pattern must be enforced.
* No hard-code logic is allowed
* Externalization of business rules to ease the onboarding of new devices

## Supportability

The CC will follow the support model of Vital Sync in terms of surveillance of running instances in the Facility Vital Sync installation. In the Facility installation deployment scenario, CC components will be monitored for correct operation.

The Standalone CC will at the very least log events and issues. Such logs will be identifiable to the instance level.

## PHI, PII and HIPAA

At present there is no requirement for PHI, PII, or HIPAA compliance for the CC.

# AUTOMATIC TESTING

Automated test will be used during the entire development as much as possible.

## Development Phase

Unit test will be used during the phase. Every public function must have a set of unit test case associated with to cover positive, negative, edge and comer cases.

## Integration Test/Regression Test Phase

In terms of reduce the effort of human mistake and effort for regression testing, an automatic testing framework will be introduced. The automatic testing will leverage the Robot Framework (open source).

## Exception

UI layout and look and feel testing still requires human interactive.

# REQUIRED THIRD PARTY SOFTWARE/COMPONENTS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Version | License | Description |
| 1 | Windows Server | 2012 | Commercial software | OS. |
| 2 | .NET Framework | 4.5 | Commercial software | Foundation for DMM. |
| 3 | Internet Information Services (IIS) | 7.5 | Commercial software | Run time environment for CC. |
| 4 | InstallAnywhere | 8.0 | Commercial software | Installer package. |
| 5 | ASP.NET MVC | 5.2.3 | Apache License 2.0 | MVC framework for Biomed App. |
| 6 | ASP.NET SignalR | 2.2.0 | Apache License 2.0 | Asynchronized notification without refresh web page. |
| 7 | JQuery | 2.1.4 | MIT License | Display content on web page. |
| 8 | AngularJS | 1.4.5 | MIT License | fundamental framework form Biomed App. |
| 9 | Bootstrap | 3.3.5 | MIT License | Look & feel, theme library for Biomed App. |
| 10 | Log4net | 1.2.11 | Apache License 2.0 | Logging framework. |

# PRODUCTION RELEASE PROCEDURE

CC will be packaged by installAnywhere, and it will be published on GDMP Server. Once a new version of CC is available on the GDMP server, and CC connected to GDMPA in online mode, then a notification will be showing on CC page. User decides the best time to download and install the CC if he/she has the Administrator privilege. Otherwise, IS will be assisted to push the installation to the designated laptop.

# PRODUCTION SUPPORT & ESCLATION PATH

Align with Medtronic standard, follow existing support and escalation path.

# Appendix A: Benchmark of SignalR

4797 connections can be connected concurrently in the following test environment:

Configuration

Connect Interval: 10ms

Connect Timeout: 300ms

Send Interval: 500ms

Send Timeout: 300ms

Hardware (Physical Machine)

CPU: Intel Core i7-2670QM

RAM: 16GB

Hardware (Virtual Machine)

CPU: 4 cores

RAM: 8GB

Software

OS: Windows Server 2008 R2

Host: IIS 7.5

# Appendix B: User Authentication Sequence Diagram

Common Client user authentication process is different in standalone mode vs. centralized deployment mode.

## User Authentication Sequence Diagram – Standalone Mode

The following sequence diagram shows the user authentication process in standalone mode.



Figure 5 User Authentication Sequence Diagram – Standalone Mode

## User Authentication Sequence Diagram – Centralized Deployment Mode

The following sequence diagram shows the user authentication process in centralized deployment mode.



Figure 6 User Authentication Sequence Diagram – Centralized Deployment Mode