

**MedApps High Level Design**

**Architecture, Data Center**

**15 May 2015**

**Author: Jim Emerson**

**Project: MedApps High Level Design**

**Version: 1.0**

CONTENTS

[Document Control 4](#_Toc501362194)

[Document History 4](#_Toc501362195)

[Signoff Authority and Approval 4](#_Toc501362196)

[Glossary of Terms 5](#_Toc501362197)

[Purpose and Scope 5](#_Toc501362198)

[Document Purpose and Scope 5](#_Toc501362199)

[Audience 5](#_Toc501362200)

[References 5](#_Toc501362201)

[Executive Summary 5](#_Toc501362202)

[MedApps System 5](#_Toc501362203)

[MedApps Architecture 5](#_Toc501362204)

[MedApps2 Architecture 5](#_Toc501362205)

[Structural Layers 6](#_Toc501362206)

[Application Service Models 7](#_Toc501362207)

[MedApps1 Architecture 10](#_Toc501362208)

[MedApps2 Development Infrastructure 11](#_Toc501362209)

[MedApps System Components and Operations 11](#_Toc501362210)

[MedApps Servers 11](#_Toc501362211)

[Alere HQ Servers 12](#_Toc501362212)

[IO Leased Co-Located Servers 13](#_Toc501362213)

[MedApps IIS Websites 13](#_Toc501362214)

[MedApps Database Servers 13](#_Toc501362215)

[MedApps1 DB Servers 13](#_Toc501362216)

[MedApps2 DB Servers 14](#_Toc501362217)

[MedApps Databases 14](#_Toc501362218)

[MedApps1 Databases 14](#_Toc501362219)

[MedApps2 Databases 14](#_Toc501362220)

[MedApps Data Handling 14](#_Toc501362221)

[MedApps Data Flow 15](#_Toc501362222)

[MedApps Services Data Flow 15](#_Toc501362223)

[Firmware Over the Air (FOTA) Updates to MedApps Service Devices 16](#_Toc501362224)

[MedApps Connections and Networking 17](#_Toc501362225)

[Connection and Network Elements 17](#_Toc501362226)

[MedApps Service Operations 18](#_Toc501362227)

[APPENDIX: MedApps Project Shutdown 18](#_Toc501362228)

[Turn Off MedApps IIS Websites 18](#_Toc501362229)

[Push Brick FOTA to MedApps Services 19](#_Toc501362230)

[Turn Off MedApps Listeners 19](#_Toc501362231)

[INDEX 20](#_Toc501362232)

# Document Control

This document has been produced using Microsoft ®Word 2010 for Windows. Printed versions of this document or versions stored elsewhere are subject to change without notice and should be considered ‘uncontrolled’.

Changes to the ‘controlled’ version will not be performed without prior consent from the author. Upon completion of such changes, the revised document will be delivered to all persons listed below.

## Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Modified by** | **Comments** |
| 0.1 | 4/20/2015 | Jim Emerson | Started document |
| 0.1 | 5/5/2015 | Jim Emerson | Submitted for technical review. |
|  |  |  |  |

## Signoff Authority and Approval

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Version** | **Date** |
| tbd |  | 1.0 |  |
|  |  | 1.0 |  |

# 

# Glossary of Terms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Alere Connect | See MedApps2. |
| Alere San Diego (ASD) | An Alere campus located in San Diego, CA, ASD includes five buildings A, B, C, D, and the outside grounds. |
| MedApps2 Architecture | MedApps2 architecture can be represented from structural and application service model perspectives. Structurally, MedApps2 has four layers: Presentation, Application, Domain, and Infrastructure. From an application service perspective, MedApps2 SaaS (Software as a Service) architecture is designed to support CRM, UserData, Device, PatientHealth, and/or device RawReadings services. |
| MedApps2.0 Remote Patient Monitoring System | <<<remainder deleted>>> |

# Purpose and Scope

## Document Purpose and Scope

The purpose and scope of this document is to present a high-level description of the currently implemented Alere MedApps2 (also known as “HealthCOM2”). Alere is shutting this system down.

## Audience

The primary audience for this document is Alere GIS Personnel.

## References

The following documents contain additional project information.

* *Alere Connect Data Flows.ppt*
* *Alere Connect-Servers\_List\_ with\_ Dependencies-022715 NO IP.xlsx*

# Executive Summary

This is a high-level description of Alere’s current MedApps2 (also known as “Alere Connect” and the “MedApps2.0 Remote Patient Monitoring System”). Details in this document (server types, locations, connections, etc.) can be used as supplemental information when MedApps is shutdown.

**Note**: See the APPENDIX: MedApps Project Shutdown for information about shutting down IIS Websites, sending brick FOTA message(s) to MedApps services, and turning off MedApps Listeners.

# MedApps System

This section describes the current MedApps2 system. The following sections describe MedApps architecture, system components and operations, and applications.

## MedApps Architecture

MedApps 2 architecture is built on the MedApps 1 foundation. The following sections introduce both designs.

### MedApps2 Architecture

MedApps2 architecture can be represented from structural and application service model perspectives.

#### Structural Layers

Structurally, Medapps2 architecture is organized into four layers: Presentation, Application, Domain, and Infrastructure. The following figure shows the key elements in each layer.

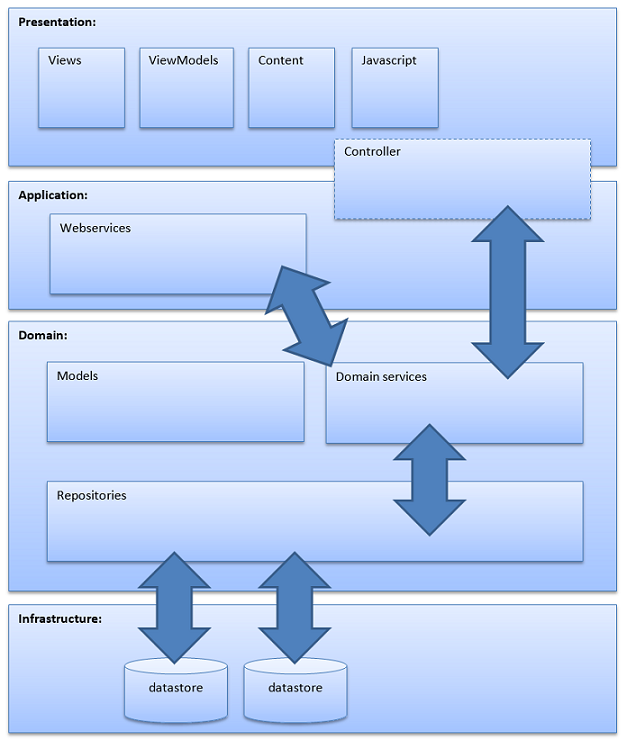


Figure 1. MedApps2 Architecture Layers

#### Application Service Models

MedApps 2 architecture is designed to support CRM, UserData, Device, PatientHealth, and/or device RawReadings services. These models are represented in the next figure.

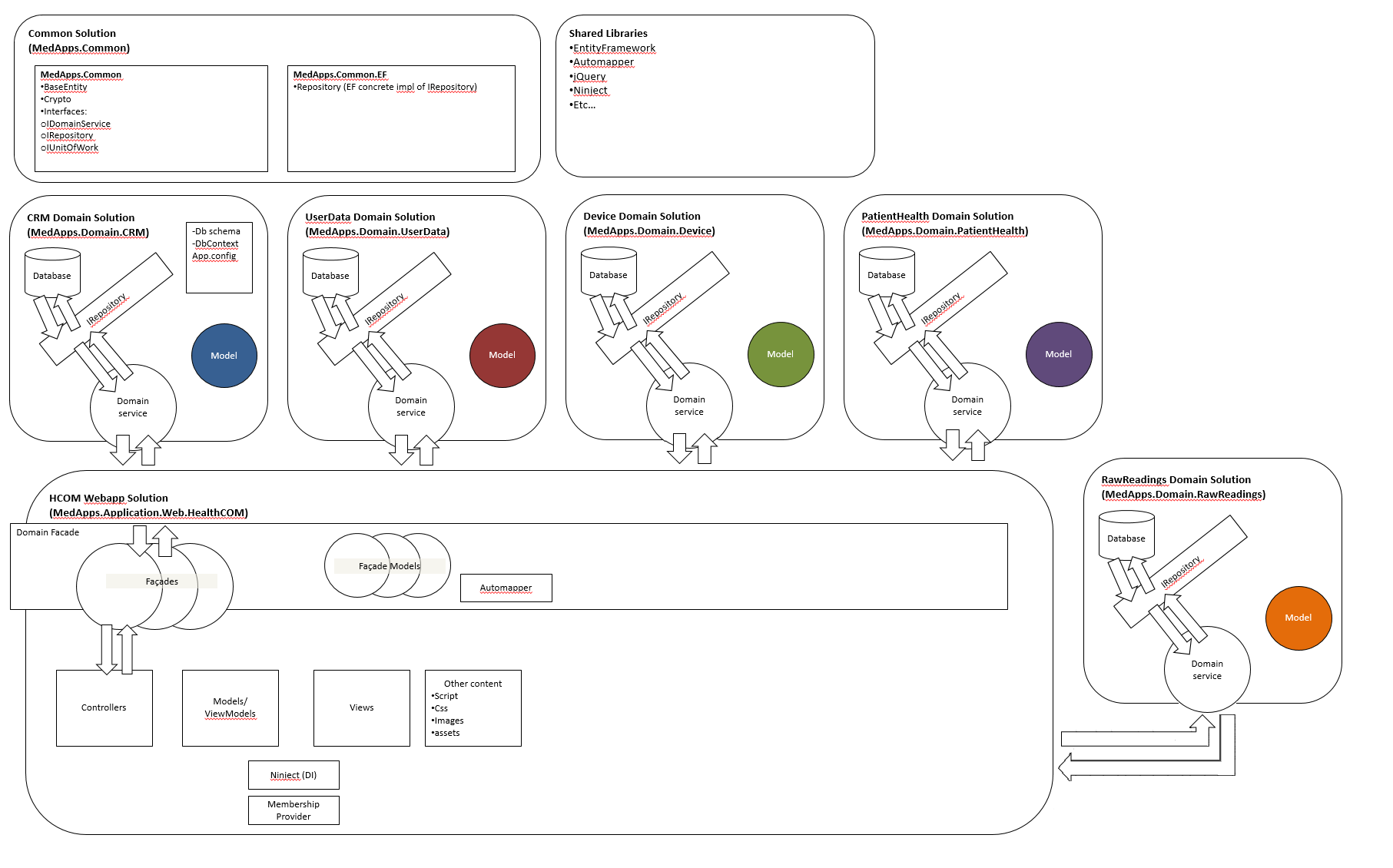


Figure 2. MedApps2 Service Solution Architecture

CRM and UserData models shown in the figure above were designed. The Device, Patient Health, and RawReadings models shown in the figure were proposed but not designed. The following sections show entity relationship diagrams (ERD) for the proposed class CRM and UserData designs.

**CRM Model ERD**

The CRM design model incorporates the entities and entity-to-entity relationships shown below. The abstract class BaseEntity has a one-to many relationship to two main entity classes (CRMEntity and CRMEntityChildAssociation) and three contact information classes (Address, Phone, Email).



Figure 3. MedApps CRM Model ERD

**UserData Model ERD**

The UserData design model incorporates the entities and entity-to-entity relationships shown below. The abstract class BaseEntity has a one-to many relationship to five user data type entity classes (User, Role, SecurityQuestion, SecurityQuestionAnswer, SecurityImageKey).

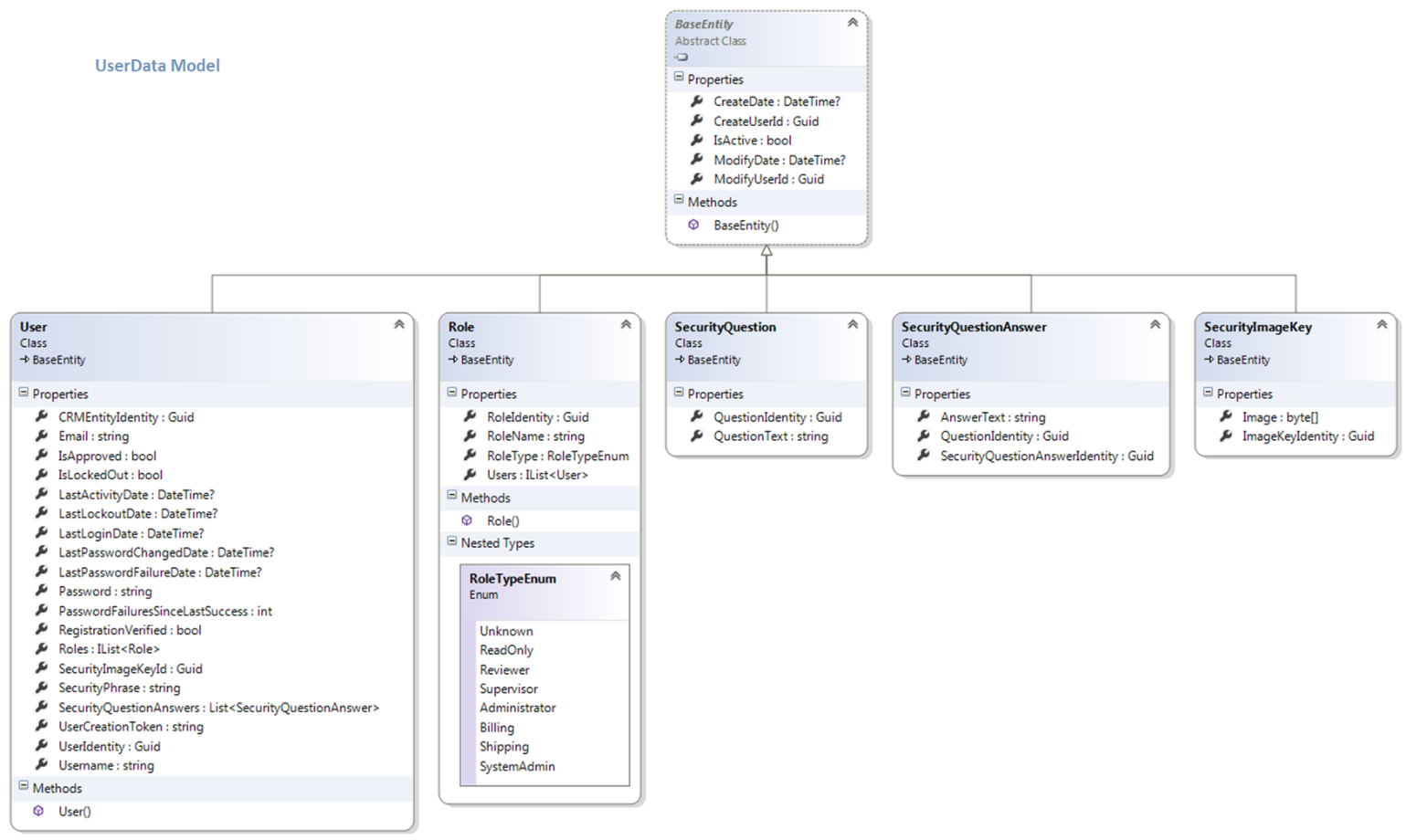
****

Figure 4. MedApps UserData Model ERD

### MedApps1 Architecture

The legacy MedApps1 SaaS architecture is represented in the next figure. The entities and communications represented in this legacy system diagram were included in designs for MedApps2.

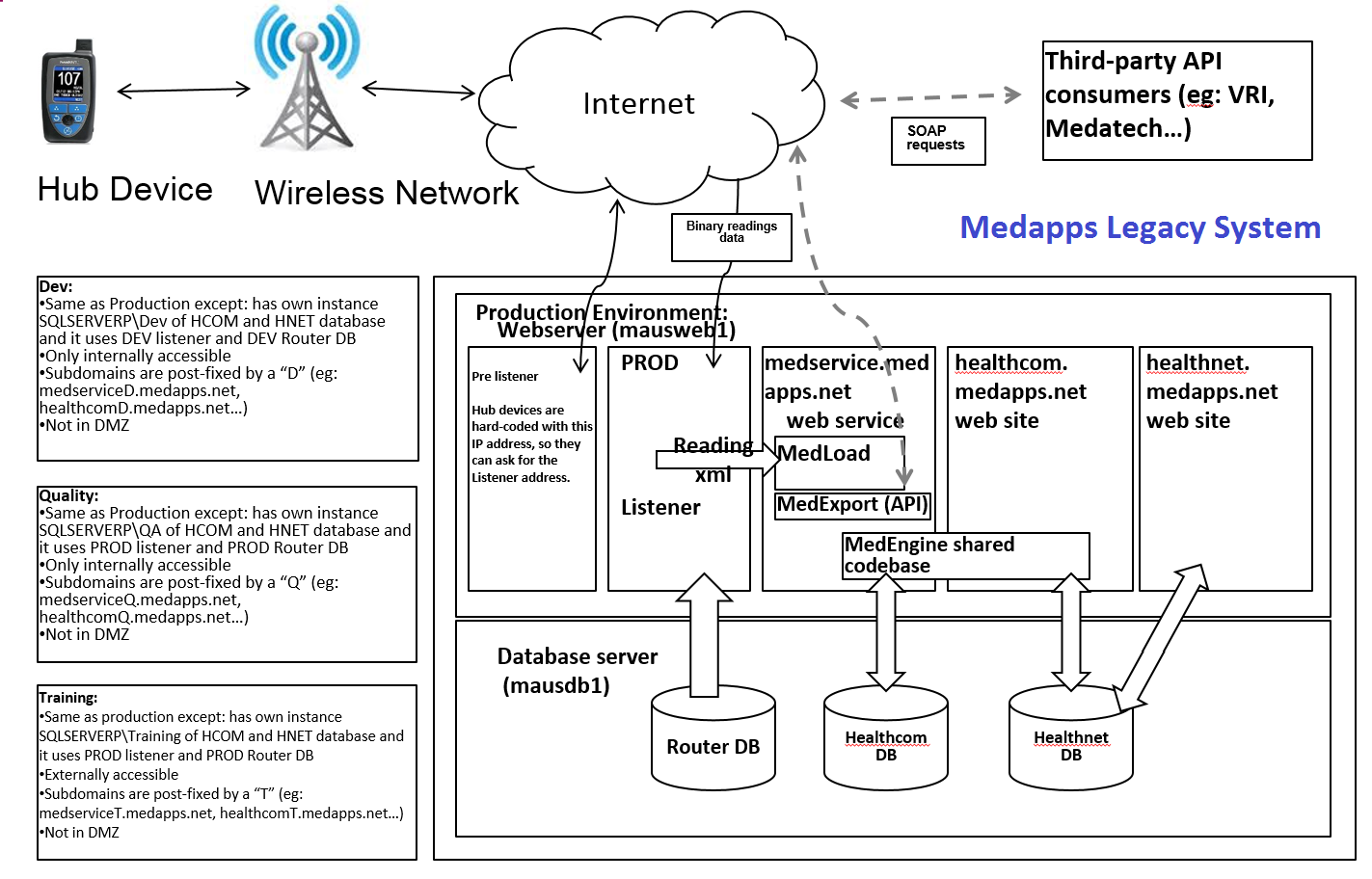


Figure 5. Alere MedApps1 Architecture

## MedApps2 Development Infrastructure

The infrastructure used in developing MedApps2 is represented below. Developer sent requests and data updates to Prod, Training, QA, and Dev databases (passed to custom Webserver/Listeners) via a utility server.

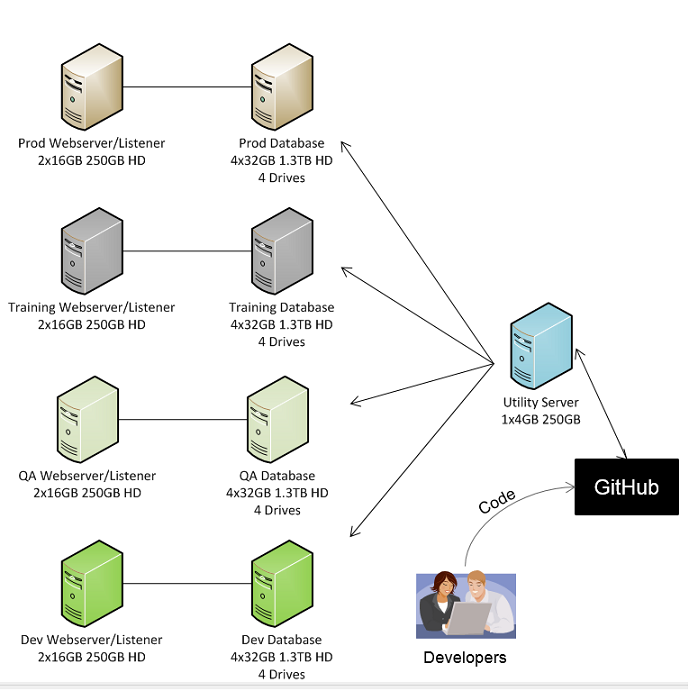


Figure 6. Alere Medapps2 Development Infrastructure

## MedApps System Components and Operations

This section introduces key MedApps system components and operations: servers, service websites, databases, data handling and flow, connections, and networking; and application log files.

### MedApps Servers

MedApps2 system and service applications run on servers located at two data centers in Scottsdale, AZ.

* Alere HQ Servers

These non-critical Alere-owned servers are located in a server room at the Alere Corporate HQ in Scottsdale, AZ.

* Leased IO Servers

These critical leased servers are co-located at the Scottsdale Colocation data center Scottsdale, AZ.

H2COM web and db servers (DEV, QA, TEST, PROD)

HCOM1-Prod webserver (mausweb1)

HCOM1-Prod db-server (mausdb1)

#### Alere HQ Servers

The Alere MedApps servers located at the company HQ in Scottsdale AZ are listed below. This table identifies the server type, host name, server, group, and IP address; and applications running on the server.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type** | **Host Name** | **Server** | **Group** | **IP Address** | **Apps** |
| NIC | webserverqa |  | DEV | 172.20.100.27 | IIS |
| NIC | webservertrain |  | DEV | 172.20.100.31 | IIS |
| NIC | webserverp | webcluster.medapps-phx-hq.local | BUS | 172.20.100.22 | IIS / MOSS frontend |
| NIC | webserverp | webserverp-nic2.medapps-phx-hq.local | BUS | 172.20.100.26 | IIS / MOSS frontend |
| Physical |  | maexch02.medapps-phx-hq.local | BUS | 172.20.100.50 | Exchange 2010 |
| Physical |  | masharepointp1.medapps-phx-hq.local | BUS | 172.20.100.18 | None |
| Physical |  | masql2008dev1.medapps-phx-hq.local | DEV | 172.20.100.70 | None |
| Physical |  | medappssql.medapps-phx-hq.local | BUS | 172.20.100.17 | SQL back end for sharepoint |
| Physical |  | medappsvm1.medapps-phx-hq.local | BUS | 172.20.100.13 | VM Host |
| Physical |  | medappsvm2.medapps-phx-hq.local | BU | 172.20.100.14 | VM Host |
| VM | medappsvm1 | appserverp.medapps-phx-hq.local | DEV | 172.20.100.25 | internal apps for DEV |
| VM | medappsvm1 | masqldev1.medapps-phx-hq.local | DEV | 172.20.100.43 | None |
| VM | medappsvm1 | masqldev2.medapps-phx-hq.local | DEV | 172.20.100.44 | None |
| VM | medappsvm1 | mawebdev1.medapps-phx-hq.local | DEV | 172.20.100.41 | IIS |
| VM | medappsvm1 | mawebdev2.medapps-phx-hq.local | DEV | 172.20.100.42 | IIS |
| VM | medappsvm2 | mawug01.medapps-phx-hq.local | BUS | 172.20.100.34 | What’s Up Gold |
| VM | medappsvm2 | medappsbkup.medapps-phx-hq.local | BUS | 172.20.100.15 | Trend Micro Worry Free Business Edition AV console |
| VM | medappsvm2 | medappsdc2.medapps-phx-hq.local | BUS | 172.20.100.20 | Domain Controller/DNS |
| VM | medappsvm1 | medappsmon01.medapps-phx-hq.local | BUS | 172.20.100.45 | WSUS |
| VM | medappsvm2 | sharepointp.medapps-phx-hq.local | BUS | 172.20.100.23 | Sharepoint 2007 |
| VM | medappsvm1 | sqlserverp.medapps-phx-hq.local | BUS | 172.20.100.24 | SQL for DEV |
| VM | medappsvm2 | webserverdev healthcomd.medapps.net | DEV | 172.20.100.28 | IIS |
| VM | medappsvm2 | webserverqa.medapps-phx-hq.local | DEV | 172.20.100.27 | IIS |
| VM | medappsvm2 | webservertrain.medapps-phx-hq.local | DEV | 172.20.100.31 | IIS |

NIC = NIC card for clusters, etc.

Physical = Physical server

VM = Virtual Machine (VM) server

#### IO Leased Co-Located Servers

The Alere MedApps servers co-located at the IO company in Scottsdale AZ are listed below. This table identifies the server type, host name, server, group, and IP address; and applications running on the server.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type** | **Host Name** | **Server** | **Group** | **IP Address** | **Apps** |
| VM | IO vCenter | mausdb1 | PROD | 172.20.110.12 | SQL 2005 |
| VM | IO vCenter | mausweb1 | PROD | 172.20.110.11 | IIS / Medapps Listener Service |
| VM | IO vCenter | MAUSWEBD1 | DEV | 172.20.110.14 | AT&T Listener |
| VM | IO vCenter | S-ACUS-DEVQA1 | DEV | 172.20.110.41 | IIS 8/Medapps Listener Service |
| VM | IO vCenter | S-ACUS-DEVQADB1 | DEV | 172.20.110.51 | SQL 2012 |
| VM | IO vCenter | S-ACUS-DEVTEST1 | DEV | 172.20.110.40 | IIS 8/Medapps Listener Service |
| VM | IO vCenter | S-ACUS-DEVTESTDB1 | DEV | 172.20.110.50 | SQL 2012 |
| VM | IO vCenter | S-ACUS-DEVTRN1 | DEV | 172.20.110.42 | IIS 8/Medapps Listener Service |
| VM | IO vCenter | S-ACUS-DEVTRNDB1 | DEV | 172.20.110.52 | SQL 2012 |
| VM | IO vCenter | S-ACUS-H2DB1 | PROD | 172.20.110.30 | SQL 2012 |
| VM | IO vCenter | S-ACUS-H2WEB1 | PROD | 172.20.110.20 | IIS 8/Medapps Listener Service |
| VM | IO vCenter | S-ACUS-UTIL1 | DEV | 172.20.110.15 | Development Tools |
| VM | IO vCenter | S-ACUS-DEVSGDB1 | STAGE | 172.20.110.110 | SQL 2012 |
| VM | IO vCenter | S-ACUS-DEVSTG1 | STAGE | 172.20.110.100 | IIS 8/Medapps Listener Service |

For more details, see *Alere MedApps Servers and Dependencies.xlsx*.

### MedApps IIS Websites

The MedApps system incorporates the IIS websites listed below. All these servers except s-acus-util1 generate application log files.

* mausweb1 – running batch job hcom1-prod
* webservertrain – running batch job hcom1-training
* webserverqa – running batch job hcom1-qa
* webserverdev – running batch job hcom1-dev
* s-acus-h2web1 – running batch job hcom2-prod
* s-acus-devstg1 – running batch job hcom2-stage
* s-acus-devtrn1 – running batch job hcom2-training
* s-acus-devqa1 – running batch job hcom2-qa
* s-acus-devtest1 – running batch job hcom2-dev
* s-acus-util1 – continuous-integration server

### MedApps Database Servers

MedApps incorporated the database servers listed below.

#### MedApps1 DB Servers

MedApps1 (HealthCOM1) included the hcom1 db-servers listed below.

* mausdb1 (prod)
* sqlserverp/dev (dev)
* sqlserverp/qa (qa)
* sqlserverp/train (train)

#### MedApps2 DB Servers

The following MedApps2 (HealthCOM2) incorporates hcom2 db-servers listed below.

* s-acus-h2db1 (prod)
* s-acus-devsgdb1 (stage)
* s-acus-devtrndb1 (training)
* s-acus-devqadb1 (qa)
* s-acus-devtestdb1 (dev)

### MedApps Databases

Alere MedApps stored data in the databases listed below.

#### MedApps1 Databases

MedApps1 (HealthCOM1) stored data in the following databases:

* Healthcom—Stores MedApps1 patient data including diagnostic readings and user data.
* Healthnet—Stores MedApps1 device and kitting data.
* Medrouter—Store MedApps1 device routing information.

#### MedApps2 Databases

MedApps2 (HealthCOM2) stores MedApps1 service records along with MedApps2 information. This information is stored in the following databases.

* Healthcom—Stores MedApps1 patient data including diagnostic readings and user data.
* Healthnet—Stores Me
* device routing information.
* Medapps.CRM—Stores MedApps2 person and group name, contact, and association information.
* Medapps.CrossDomain—Stores MedApps stored procedures and views that interact with other medApps2 databases.
* Medapps.Devices—Stores MedApps2 device and kitting information for hubs and peripherals.
* Medapps.HabitII—Stores data related to the Habitll study. This information is no longer used.
* Medapps.PatientHealth—Stores MedApps2 patient records including diagnostic readings, thresholds, and alert data.
* Medapps.RawReadings—Stores MedApps2 patient diagnostic reading data in the raw XML format. The MedApps Listener sends this XML data to MedApps2 system.
* Medapps.Users—Stores medApps2 user authentication details.

### MedApps Data Handling

MedApps2 handles the following types of data (also referred to as “contexts”):

* Device data

Devices, cell carriers, provisioning information, hub devices (HealthPal, HomeLink, MobileLink) and associated peripheral devices (BP meter, Pulse Ox meter, Glucose meter, etc.), drivers, device firmware, etc.

* Patient health data

Patient information, diagnostic readings, thresholds, alerts, etc.

* CRM data

Health care providers, health care professional, patient name, caregiver-contract, etc.

* Raw diagnostic readings data

Raw input data from devices.

* User data

User identifications, roles, credentials, etc.

A MedApps2 class library corresponds to each type of data. These libraries contain:

* interfaces and classes

These system components define the data model.

* Repositories

These data stores abstract the models interfaces to the underlying data store.

* Domain services

These services expose the context interface to the outside world.

The Alere team built a custom Microsoft Windows Communication Foundation (WCF) Web HTTP Service for MedApps2. The service provides web functions including exposing a Web endpoint that sends data as XML or JSON.

MedApps2 implementation also includes a web application (HealthCOM2) complete with views, view-models, controllers, static content, Javascript, etc.

The context data and WCF implementations are developed in the infrastructure represented in Figure 7. Alere Medapps2 Development Infrastructure. This project incorporates MedApps2 Entity Framework references and database creation/change scripts.

### MedApps Data Flow

Two types of data flow are central to MedApps: MedApp services operations and FOTA provisioning. These processes are described in the following sections.

#### MedApps Services Data Flow

Data requests and information flow into MedApps2 from remote patients and medical service providers as Web service calls via HTTP. In the opposite direction, responses flow from MedApps2 back end servers and storage out to remote patients and medical service providers. The overall flow of this data is represented in below.

**Note**: MedApps1 and 2 features were used by medical service providers (physicians, nurses, administrators, etc.). However, MedApps1 and 2 were not directly available to patients.

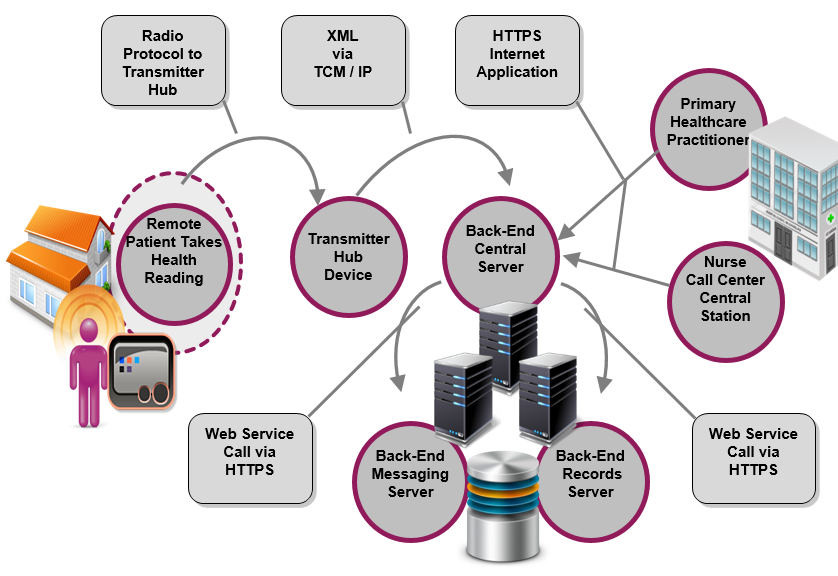


Figure 8. Alere Medapps2 Data Flow

#### Firmware Over the Air (FOTA) Updates to MedApps Service Devices

Over the Air (OTA) programming systems distribute new software updates, reconfiguration settings, etc. to distant devices via the Web. Alere uses a custom Firmware Over the Air (FOTA) to provision MedApps devices.

The FOTA system is used to provision MedApps service hubs (HealthPal, HomeLink, MobileLink) remotely, re-route the hubs, update hub firmware, and download new drivers for new types of peripheral devices.

At the end of the MedApps project, Alere will send a final FOTA data transfer to “brick” any of our devices that are still active (e.g., HomeLinks). This action will disable our devices, preventing them from transmitting patient diagnostic readings data.

### MedApps Connections and Networking

MedApps is accessible through Web-accessed Alere Connect. This system connects home point-of-care (POC) service recipients with services supported by backend servers. These operations are represented in the figure below.

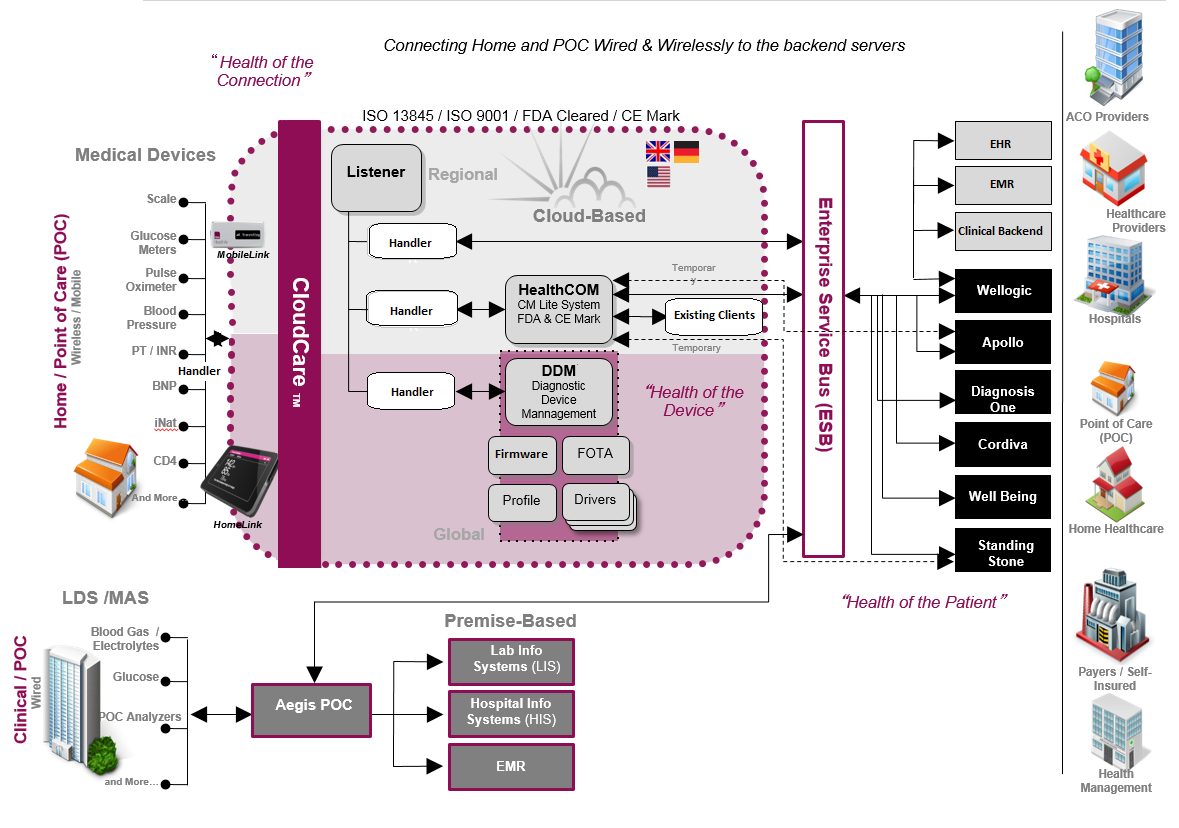


Figure 9. Alere MedApps2 Connections and Networking

#### Connection and Network Elements

During MedApps operations, an external device communicates with a MedApps hub device (HealthPal, MobileLink, HomeLink) via USB cable or Bluetooth connections. Each hub has a SIM card and communicates (via a cellular APN) over a 2G network to the MedApps Listener service.

##### MedApps Hub Devices for Patient Monitoring

MedApps uses hub device systems for medical monitoring connectivity: HealthPal, MobileLink, and HomeLink.

HealthPal—Alere’s HealthPal hub connects devices at patient sites with MedApps. HealthPal-connected devices uses wireless (Bluetooth), wire, or cellular technology to collect and send health readings from retail medical monitors to a patient’s electronic health record (EHR). HealthPal is FDA-cleared for use with blood pressure and glucose meters, pulse oximeters, and weight scales.

MobileLink—Alere’s MobileLink is a communications hub that links compatible, remotely located medical monitors at patient homes or other POC sites to MedApps backend systems.

HomeLink—Alere’s Homelink is a wireless home hub that patients use to received, store, and transmit patient-generated health care information. The tablet-sized touch screen captures data from the patient’s health monitoring and recording devices.

### MedApps Service Operations

MedApps2 was designed to connect patients at remote locations with service provides via the Web. The general design for service delivery is represented in the following figure.

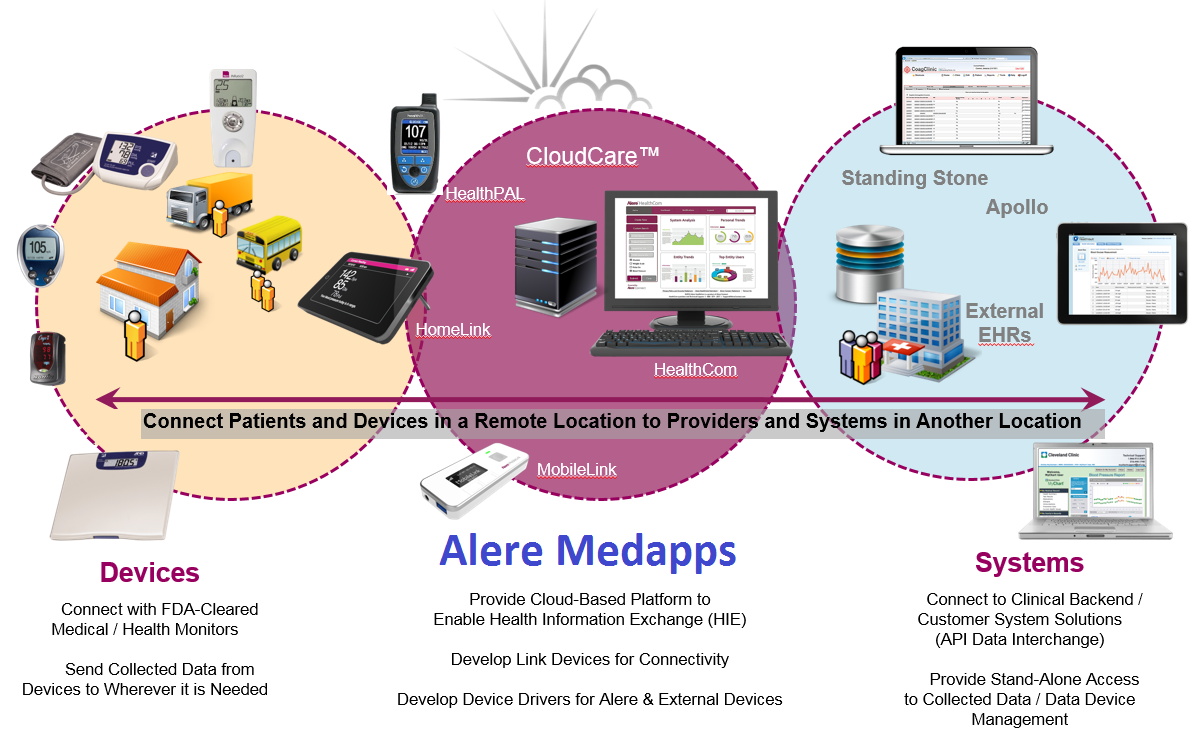


Figure 10. Alere Medapps2 Service Delivery

MedApps HomeLink and MobileLink systems are accessed by devices recording patient health data to send data to distant service provider systems.

# APPENDIX: MedApps Project Shutdown

The MedApps project shutdown will involve turning off IIS websites including hcom batch jobs, pushing a brick FOTA to MedApps service sites, and turning off MedApps Listeners.

**Note**: In the lists below, the character string before the parenthesis is the machine name; e.g., mausweb1. The string in the parenthesis is the system or instance name; e.g., hcom1-prod. In this case, hcom1-prod is the HealthCOM1-Production system.

### Turn Off MedApps IIS Websites

The Alere team will turn off all MedApps IIS Websites and hcom batch jobs running on the following servers:

* mausweb1—running batch job hcom1-prod
* webservertrain—running batch job hcom1-training
* webserverqa—running batch job hcom1-qa
* webserverdev—running batch job hcom1-dev
* s-acus-h2web1—running batch job hcom2-prod
* s-acus-devstg1—running batch job hcom2-stage
* s-acus-devtrn1—running batch job hcom2-training
* s-acus-devqa1—running batch job hcom2-qa
* s-acus-devtest1—running batch job hcom2-dev
* s-acus-util1— running batch job continuous-integration server

### Push Brick FOTA to MedApps Services

The Alere team will push a FOTA brick message to disable MedApps services HealthPals, HomeLinks, and MobileLinks from MedApps Listeners instances running on the following servers:

* mausweb1—running batch job hcom1-prod)
* webservertrain—running batch job hcom1-training
* webserverqa—running batch job hcom1-qa
* webserverdev—running batch job hcom1-dev
* s-acus-h2web1—running batch hcom2-prod
* s-acus-devstg1—running batch hcom2-stage
* s-acus-devtrn1—running batch hcom2-training
* s-acus-devqa1—running batch hcom2-qa
* s-acus-devtest1—running batch hcom2-dev

### Turn Off MedApps Listeners

The Alere team will turn off the MedApps Listeners instances running on the following servers:

* mausweb1—running batch hcom1-prod
* webservertrain—running batch hcom1-training
* webserverqa—running batch hcom1-qa
* webserverdev—running batch hcom1-dev
* s-acus-h2web1—running batch hcom2-prod
* s-acus-devstg1—running batch hcom2-stage
* s-acus-devtrn1—running batch hcom2-training
* s-acus-devqa1—running batch hcom2-qa
* s-acus-devtest1—running batch hcom2-dev

# INDEX

<<<deleted>>>