**CARESTREAM HEALTH**

Template Part Number: 7H1891, Rev A – 0.5

|  |  |
| --- | --- |
| Part Number: 7K5807 | **Author: Xiaoming Zhao** |
| **Project : Kiosk** | **Product : Mongoose** |
| **Document Title: Mongoose Print Server Architecture Design Document** | |

Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Description of Change** | **Revision date** |
| 0.1 | Initial draft | 2015/7/13 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**TABLE OF CONTENTS**

[1 Overview 3](#_Toc424903141)

[1.1 Purpose 3](#_Toc424903142)

[1.2 Scope 3](#_Toc424903143)

[1.3 Related Document 3](#_Toc424903144)

[1.4 Definitions 3](#_Toc424903145)

[1.4.1 Acronyms 3](#_Toc424903146)

[1.4.2 Meanings 4](#_Toc424903147)

[1.5 Conventions 4](#_Toc424903148)

[*2* Introduction 5](#_Toc424903149)

[2.1 Customer Context 5](#_Toc424903150)

[2.2 Interactions Diagram 5](#_Toc424903151)

[*3* Architecture Design 7](#_Toc424903152)

[3.1 General Description 7](#_Toc424903153)

[3.2 Components of Mongoose Print Server 7](#_Toc424903154)

[*4* Components Design & Description 7](#_Toc424903155)

[4.1 PrintInterface, BulletinInterface and Mongoose Interface Web Service 7](#_Toc424903156)

[4.2 MongooseInterface 7](#_Toc424903157)

[4.3 AutoFormatComposer(AFC) Server 8](#_Toc424903158)

[4.4 Store2Media 9](#_Toc424903159)

[4.5 PrintScheduler Server 10](#_Toc424903160)

[4.6 Payment Server 12](#_Toc424903161)

[4.7 HL7 Server 12](#_Toc424903162)

[4.8 Query/Retrieve Server 12](#_Toc424903163)

[4.9 LRU Server 13](#_Toc424903164)

# Overview

## Purpose

The intent to create this document is to descript the Mongoose Print Server Architecture. The Mongoose Print Server was introduced based on the King Platform Print Server.

## Scope

This document describes the architecture of the Mongoose Print Server.

The Mongoose Print Server was created based on the King Platform Print Server, so this document will focus on the Mongoose individual architecture or features, for the King Platform server architecture, please refer to the King Platform Architecture Design Document.

## Related Document

|  |  |
| --- | --- |
| 9J5699 | Customer Requirements Document (CRD) Self-Service Radiology Kiosk (Project: Mongoose) |
| 9J7285 | Mongoose Kiosk Product Requirements Specification (PRS) |
| AC1803 | Kiosk Mongoose System Architecture Design |
| 7K4751 | Carestream Kiosk King Platform Architecture Design |
| AC1311 | Kiosk Consumable Data (KCD) Tracking File Definition |
| AC1621 | Kiosk Exam Data (KED) Tracking File Definition |
| 9H3411 | Kiosk Gen1 (1.0) Architecture document – legacy info |

## Definitions

### Acronyms

|  |  |
| --- | --- |
| AFP6000 | Model number of the first generation self-print terminal hardware and software |
| AIO | All In One – this describes the new kiosk self-print terminal hardware that uses a single all-in-one chassis to house all components (Embedded PC, LCD display, 5950 imager, report printer, bar code reader, IC and mag card reader, ...). |
| CRD | Customer Requirements Document |
| CSH | Carestream Health, Inc. |
| GUI, UI | Graphical User Interface – the interface used by the patient at the kiosk |
| Kiosk K1, K1 | The project name for the original version of the Self-Service Radiology Kiosk sold only in China. |
| Kiosk K2, K1 | The project name for a new version of the Self-Service Radiology Kiosk - sold only in China. This version is to use the new AIO kiosk hardware using the updated background images on the existing K1 user interface. |
| Kiosk K2, K2 | The project name for a new version of the Self-Service Radiology Kiosk - sold only in China. This version is to use the new AIO kiosk hardware, and to use an updated user interface. |
| Mongoose | The project name for a new version of the Self-Service Radiology Kiosk - sold worldwide outside of China. This version is to use the new AIO kiosk hardware, to include additional optional components, and to use an updated (and localizable) user interface. |
| Platform, King Platform | King Platform is the base platform to serve the Panda and Mongoose product. |
| PRS | Product Requirement Specification – the document that lists all the requirements for the product and which are the inputs that drive the content of this document |
| PS1000 | The Carestream model number of the DICOM Print Server hardware and software used in the first generation self-printing kiosk system. |
| Terminal, Terminal Server | The terminal software |

### Meanings

|  |  |
| --- | --- |
| Term | Meaning |
| Mongoose | Kiosk World Wide Product |
| Panda | The product for the China market (ROW) |
| King Platform | A new platform for the Mongoose and Panda product, which will leverage the GX Platform |
| “PrintImages” Folder | When 3rd party system (RIS, PACS) print the images to PS, the PrintSCP Server will received the raw images and save the files into this folder.  Generally, this folder path is:  “C:\GX Platform\PrintImages” |
| “Inbox” Folder | This folder is used to save the DICOM files, and the DICOM files will be archived by the Acquisition Server.  SSCP received DICOM file will be saved into this folder;  AutoDelivery Server will compose the RAW images to DICOM images and save the DICOM images into this folder;  Generally, this folder path is:  “C:\GX Platform\Inbox” |

## Conventions

N/A

# Introduction

## Customer Context

Below is a diagram to describe the Mongoose Print Server, King Platform Server and PANDA Print Server relationship.

As design, the King Platform will be the base platform to for the Mongoose and PANDA product, the King Platform will support the general functionality for the product. Each product need inherit from the King Platform Server, and append their product individual features, disable, block the unnecessary features in the King Platform if necessary.



## Interactions Diagram

Mongoose as the world wide Kiosk Product, it need a lot of integration effort to communicate with others systems or subsystems in the hospital. Such as the RIS, PACS, Modality, HIS, payment, film printer, report printer, DVD handler, USB handler, Terminal software, etc.

Below is a diagram describing the interactions diagram.



# Architecture Design

## General Description

A lot of components for Mongoose product were already supported by the King Platform Server, such as the SSCP Server, Acquisition Server, Print Scheduler Server, etc.

The Mongoose product architecture document will focus on the new/missing component/server for the Mongoose.

## Components of Mongoose Print Server

Diagram blow was based on the King Platform to describe the server/component for the Mongoose Print Server, from the diagram you can find main component inherit from the King Platform and the required but not exist components for the Mongoose product.



# Components Design & Description

## PrintInterface, BulletinInterface and Mongoose Interface Web Service

In the King Platform, the PrintInterface and BulletinInterface Web Service was hosted in the same web service site, in the Mongoose product, we will split them into 2 individual WebService site to decouple the web service site deployment.

A new web service will be introduced into the Mongoose product: MongooseInterface Web Service. The reason to introduce a new interface is that we still need some of the interface in the previous PrintInterface and need some more interface for the Mongoose, so we decide to create a new Web Service to provide new interface features for the Mongoose.

## MongooseInterface

In the MongooseInterface, it will provide several key functions:

* *QueryStudyInfo interface*: Allow the terminal to retrieve the study list by PatientId;
* *Store2Media interface*: allow the terminal to request the DICOMDIR fileset from the PS;
* *Store2Print interface*: allow the terminal to request the film output;
* *GetTaskStatus*: allow the terminal to query the Store2Print and Store2Media output status;

For the QueryStudyInfo interface, the return value is a XML to show the patient study list, in this xml, it will present some detailed information, the important information is that it will told the Terminal if the study could be printed to film printer or not.

Here is some sequence diagram to describe the interface.



QueryStudyInfo

Below diagram describe the Store2Print and Store2Media task sequence, first, the TerminalUI will create a Store2Media/Store2Print task by calling the MongooseInterface, then the TerminalUI will get a task id from the PS, it could use this task id to check the status interval.



## AutoFormatComposer(AFC) Server

The AFC Server provides the eFilm generation job.

The new task is created by the Terminal, when patient swipe the card, first, the Terminal will ask the study list from the PS MongooseInterface by call the QueryStudyInfo interface. Then the patient could be able to select a study and print to the film printer, the Terminal will create the Store2Print task by calling the MongooseInterface web service (you can refer above sequence diagram). The new task will be created into the Store2PrintTask table in the DB.

The AFC Server is a windows service; it will monitor the Store2PrintTask table to check any new task arrived. If new task created, the AFC Server will prepare the config xml and call the eFilmComoposer module to generate the eFilm. At last, the eFilm will be saved into local disk and the task status will be updated, then the PrintScheduler server will know the eFilm is ready and begin to print.

Below is a sequence diagram for the total sequence of the AFC Server.



## Store2Media

The Mongoose has a new feature to support the DICOMDIR export; user could get a USB device which stores the DICOMDIR fileset.

The workflow is a bit similar with the Store2Print, first, the patient should choose a study from the study list at terminal, then ask the USB flash output. The Terminal will create a Store2Media task by calling the MongooseInterface; the MongooseInterface will create a task record into the Store2MediaTask table in the DB. The Terminal will query to retrieve the task status periodly.

The Store2Media Server will monitor the DB to check any new task was created, then prepare the DICOMDIR in the background, after the DICOMDIR fileset is ready, it will be moved to the ftp site and update the task status in the DB, and then the terminal will start to download the DICOMDIR from the PS to terminal.

Here is a diagram to show the total sequence.



## Payment Server

TODO

## HL7 Server

HL7 is a very popular protocol through the Medical Information System, a lot of system running in the hospital implemented this interface, for example, RIS, HIS, EMR, EPR, PM system, etc.

Previously, if Kiosk need to communicate with the RIS system, our integration team develop a integration software to access the RIS DB directly, this approach has some limitations, for example, if the RIS didn’t allow others system to connect to their DB directly due to the security reason, there is no way to communicate with the RIS system for the Kiosk. In this case, we can use the MWLSCU Server to retrieve the message from the RIS/HIS system, but this is not the only way we can use, HL7 is another useful interface we can leverage.

With the HL7 interface, the King Platform will have the ability to communicate with other systems running in the hospital that support HL7 interface.

In the Mongoose, we will use the HL7 server to communicate with HIS/RIS system to retrieve the patient information or report.

As design, the HL7 Server will receive the message from 3rd party system, after receive, it will check the message type or content to determine the purpose of the message, then forward to different consumer to parse and consume the message, for example, if the message is to notify the report ftp URL, the HL7Server will forward the message to the specified module to parse and consume.



## Query/Retrieve Server

In the Kiosk K1, there is only one way to get the DICOM images in the Print Server, modality print the images to PS PrintSCP Server. This is not enough in the King Platform. We need to add some more method to retrieve DICOM images from the 3rd party system to fit different situations in the customer site.

The Query/Retrieve Server will be very useful to retrieve the DICOM images from 3rd PACS/Workstations. We can create a scheduler in the Query/Retrieve Server to monitor the 3rd party PACS/Workstations system images and another scheduler to retrieve the images from the 3rd party systems. For example, there should be a Query scheduler in the Query/Retrieve Server, it will monitor and check if there are new DICOM images in the remote PACS and not retrieved by the Print Server, create accordingly retrieve tasks into the DB, then Retrieve scheduler will retrieve the DICOM images from the remote PACS according to the retrieve tasks from the DB.

We must design additional rules to allow user/system create some emergency task with the top priority to ensure these images can be retrieved in the shortest time for the emergency patient.

## LRU Server

TODO