



Create tomorrow

Mint Vision Application Architecture

10 December 2020

Created by Peter Reid



1 Introduction

This document outlines the Architecture of the Mint II Vision Product

The Mint II Vision Product combines a Universal Windows application with Azure Cognitive computing, storage and processing to leverage the best of both worlds. The full architecture of the system is described below.

All the components built below are Mint's Intellectual Property. For components that reside in Azure, such as the CosmosDB and Cognitive Services, these are Microsoft's Services but the manner in which our code consumes these services is our Intellectual Property.

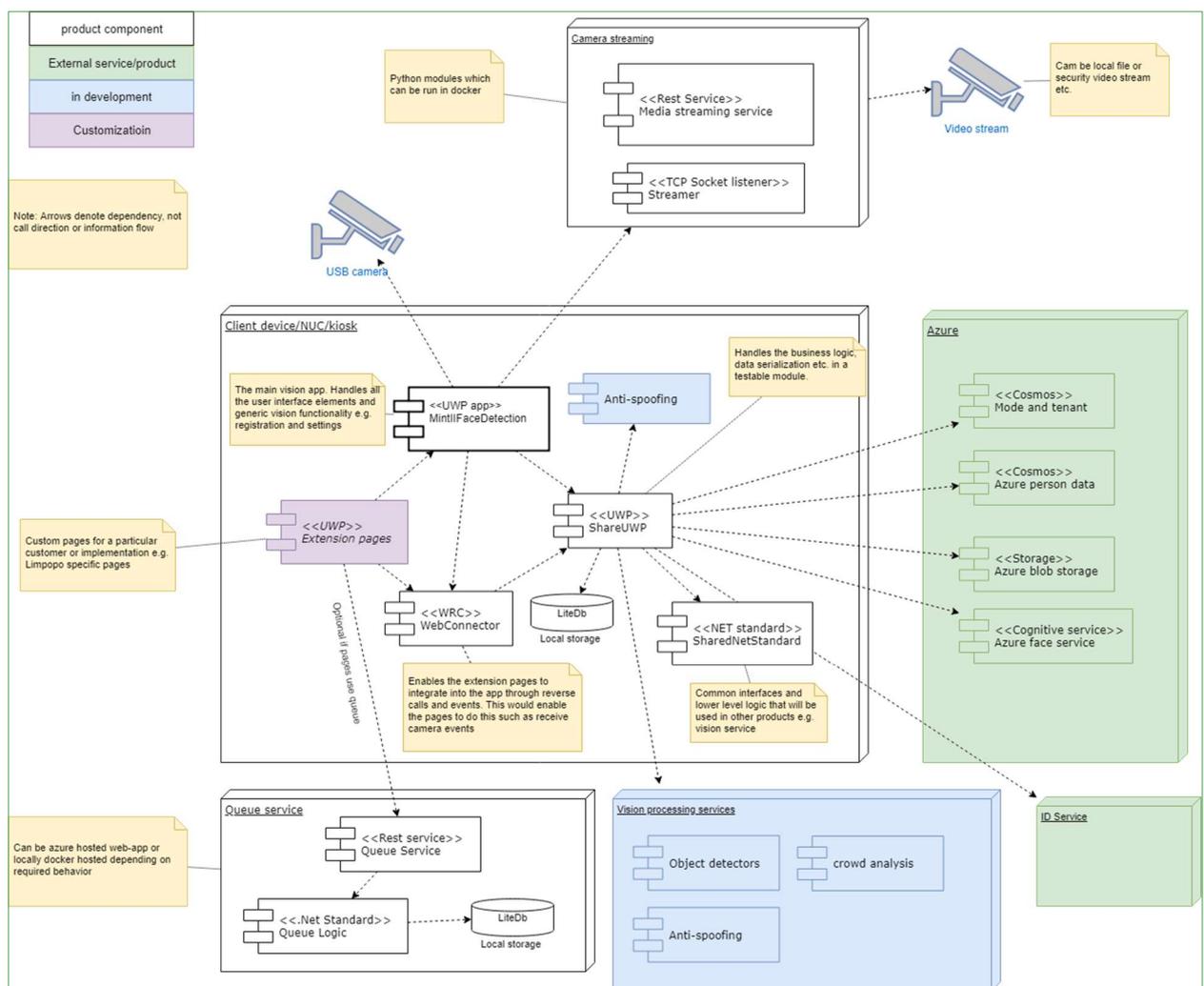


Figure 1: Mint II Vision Application Architecture

The individual components are described below.

1.1 Client Device

The Client Device is a Universal Windows Application. This application runs in Kiosk Mode on any Windows 10 Device, including embedded devices such as Raspberry Pi's.

The application has several components that are noteworthy:

- The application can run in online or offline mode
- The application can bind to streaming or usb cameras
- The application has a local storage solution that is then synchronized to the cloud
- The application can embed custom pages such as web pages, which are used to extend the application into several different use-cases

1.2 Azure

The application relies on several technology layers in Azure. Each layer is indicated below.

1.2.1 Storage

1.2.1.1 Images

Images that are captured of each face event are stored as images in Azure. This means that minutes after a capture event, the images and audit entries are stored in the cloud and retrievable from anywhere in the world.

1.2.1.2 Data

Data for the application, such as users, audit trails and settings, are stored in Azure CosmosDB. They are stored as documents and synchronized regularly to the devices themselves.

Cognitive Services

The application uses Microsoft Cognitive Services for

- Face Detection
- Face Recognition
- Similar Face Matching

1.3 Camera Streaming Service

In order to consume the stream from network cameras, a separate streaming server is deployed with the application. This is a service that is deployed using containers either on-premises, independently in Azure or within a Kubernetes cluster. The service consumes streaming camera in a multitude of formats, applies any necessary image correction, and then publishes the stream to the application in order to be consumed by the system.

1.4 Vision Processing Service

The Vision Processing Service is a REST-based, container-distributed service that performs Image Processing services such as Object Detection, Object Recognition, and similar machine-learning-based services. This processing engine is external to the application so that powerful GPU-based servers can process the images.

1.5 Queue Service

Hosted external to the application (often as a service in Azure) is the Queue Management service. The Queue Management Service allows for people queueing, such as at a branch at a bank or other such use cases.

1.6 Extension Services

The application is designed to be extended to several different use-cases. The extension part of the application combines two pluggable items:

- Custom Pages. For example, pages that serve content for touchless purchasing in a canteen differ vastly to pages that dispense medication
- Custom integration. For example in a university setting the application connects to the Student Database to synchronize student details
- Custom Triggers. When using the application for example as an Access Control mechanism, the triggers open or close doors. When in a canteen, the triggers can cause vending machines to dispense, etc.

2 User Flow

The following flows describe how the application is used in an example application. Note that this is a sample use-case; the application is highly extensible and so can be used in many different ways.

2.1.1 User Enrolment



When a new user arrives at the system and is not recognized, the system prompts the patient to register.

2.1.1.1 Image Capture



The system takes several pictures of the user for enrolment

2.1.1.2 ID Card



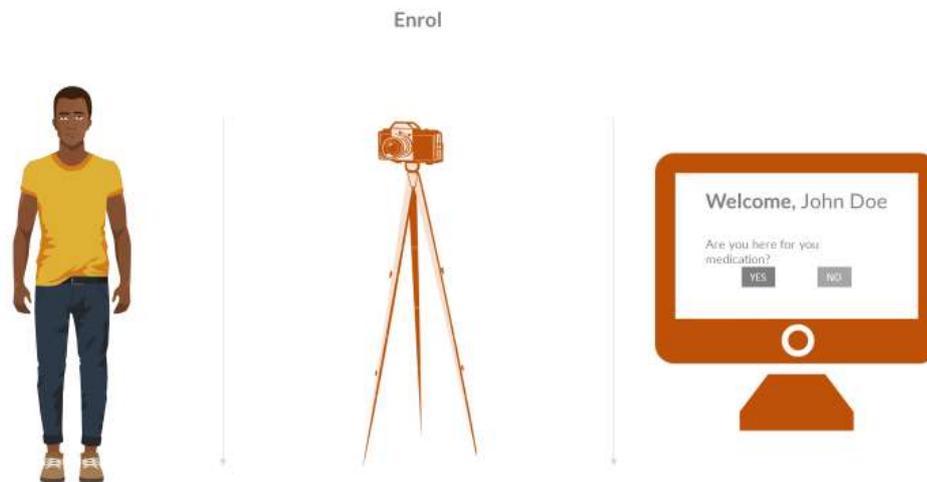
The user holds up their ID Card, and the system extracts their ID Number, first name and last name. They are also optionally prompted to enter their cell-phone number, so that they can receive SMS notifications of their progress.

2.1.1.3 Prompts for the Queue (if Queue Management is enabled)



When the user first registers, they are asked what the purpose of their visit is. This is so that they can be directed to the correct queue.

2.1.2 User Greeting



If the user is recognized on arrival, they are asked to confirm their presence and the reason they are at the facility.

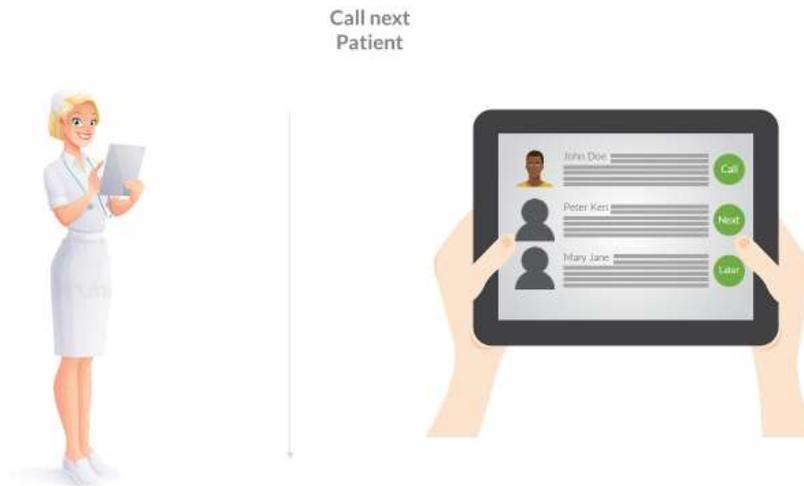
2.1.2.1 Queue Prompts



The system prompts them regarding the queue time, and optionally notifies them that they will be sent an SMS when they are close to the front of the queue

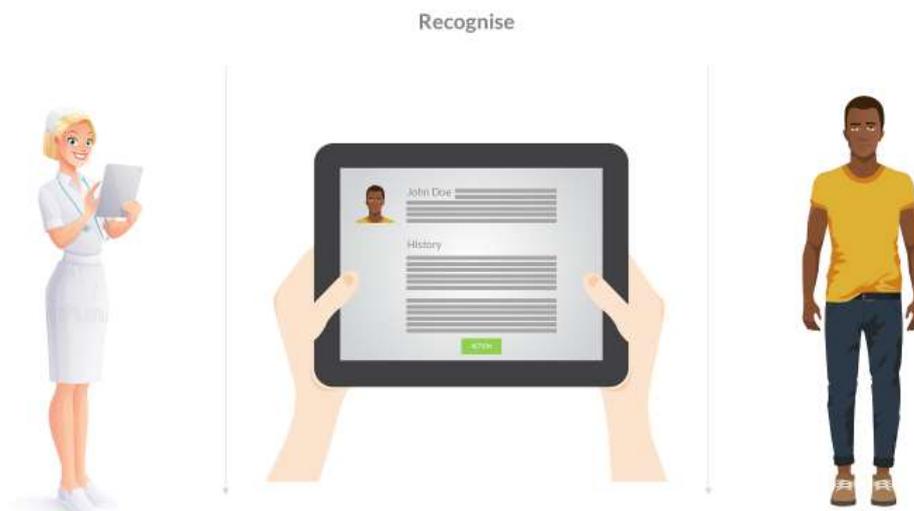
2.1.3 At Station Greeting

2.1.3.1 At station



At each station, the station manager is shown a list of the current queue, with pictures of each user, and can call the next user in the queue. In this example a clinician is seeing a list of patients and is calling the next patient.

2.1.3.2 Recognition



As the user walks up to the station, they are recognized by the cameras and their history is displayed to the station manager. This will include the history of a user that has visited multiple locations (where this system is installed) over the last few days or weeks.

2.1.3.3 Capture of actions



The station manager performs whichever action is necessary for the use and captures a free-text description of what they did in the patient history.

2.1.4 Reports and Dashboards (Included in Phase 2)

2.1.4.1 Reports and Dashboards

Reports will be displayed to admin staff.

(Note: Some elements on the *example* dashboard that relate to the full user journey, as illustrated below, are only applicable in some use-cases.

