DisplayNote Launcher

# Technical documentation

21 June 2019

# Overview

Launcher is a software solution that turns your display into a more personal device, through the use of Launcher receivers (Windows) and Launcher client apps (iOS and Android).

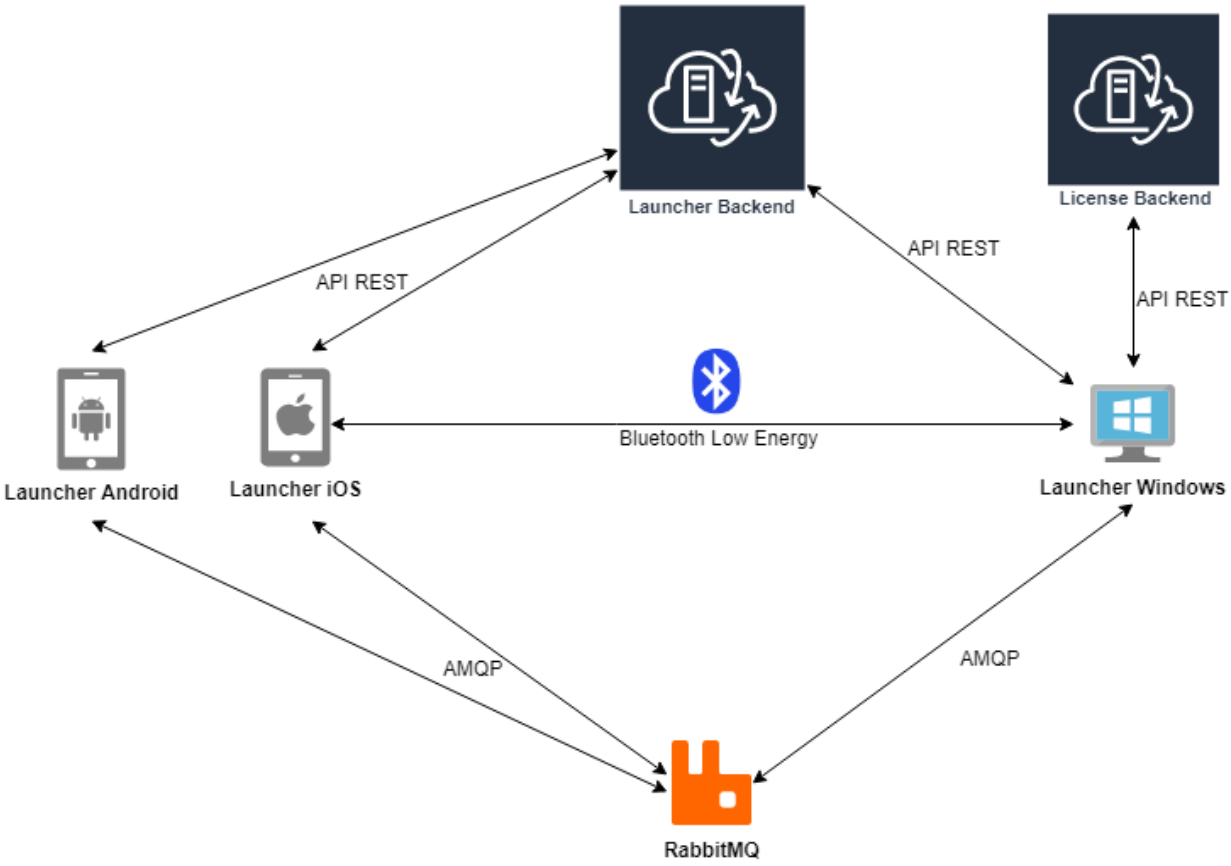
Launcher allows smartphone users to create a secure and more personalised touchscreen experience. Using an Office 365 account Launcher users can get instant access to One Drive content, calendars, and they can even initiate calls on the large interactive touchscreen.

Users can organize and decide what applications to use consistently across their meeting room touch screens, making it easier for teams to find and launch apps they need for their meetings.

For more information on DisplayNote Launcher, please visit our website at: displaynote.com/solutions/launcher

# Architecture

The following diagram illustrates the Launcher architecture between Launcher receivers, clients, the message broker and backends. Please see additional information in subsequent sections.




## Launcher receiver

Launcher receiver for Windows is the central piece of software for the Launcher solution. It uses Bluetooth Low Energy (BLE) to announce its presence to Launcher client apps.

Once a Launcher client app (see information in next section) is detected, all communication between them is achieved over RabbitMQ (www.rabbitmq.com), secured by TLS.

## Launcher client app

The Launcher client app (iOS/Android) is a mobile app that acts as user identifier and allows users to interact with Launcher receivers. The iOS version of the mobile app uses Bluetooth Low Energy to discover nearby devices that are running the DisplayNote Launcher application.

Once a Launcher receiver is detected, all communication between them is done over RabbitMQ (www.rabbitmq.com), secured by TLS.

## Backends

### Hosting, certifications and licensing

Our backends are hosted on Microsoft Azure. Azure's data centres are geographically dispersed and comply to ISO/IEC 27001:2005, SOC 1 and SOC 2 and have a CSA STAR certification. The backend manages licensing and updates API REST, and it is only accessed by the Launcher receiver application.

### Backend communication with Launcher

Launcher backend is accessed by Launcher receivers and clients, it is used for verifying that devices discovered through BLE are Launcher instances, and returns all information needed to establish a connection via RabbitMQ.

### Encryption and certificates

Communication between our backends and the Launcher receivers or clients, is encrypted and transmitted over HTTPS with 2048-bit asymmetric encryption and 256-bit symmetric encryption, using certificates from third party credited authorities.

# Launching calls

Launcher receiver is able to start any meeting/call that has been set up through Microsoft Calendar. A calendar event for a Teams/Skype for Business contains a public link to the meeting, that is used by Launcher to start the call.

## Microsoft Teams

If MS Teams is not installed on the Windows device, the Launcher receiver will open the meeting link via web browser. Otherwise, the Launcher receiver will call MS Teams process passing the meeting link as parameter. Launcher receiver does not sign in or out MS Teams accounts. If MS Teams is previously signed in with an account, user will join to the meeting/call under that account.

## Skype for Business

If Skype for Business (SfB) is not installed on the Windows device, the Launcher receiver will open the meeting link via web browser. Otherwise, the Launcher receiver will interact with Skype for Business using its SDK (SDK documentation).

The SDK needs the username and password in order to sign in into Skype for Business. This information is requested from the user via the Launcher client app and it is sent to the Launcher receiver securely through RabbitMQ using TLS. The password is never stored, and it is completely forgotten once users disconnect or the application is closed.

# Tokens and authentication

Users can sign in within their Office 365 account to get Calendar and OneDrive information.

Authentication is done using the Microsoft identity platform (MS Identity Client documentation) to retrieve a valid token that is used for accessing Microsoft Graph API (MS Graph API documentation).

As authentication is done in Launcher client apps, the Microsoft token is sent to Launcher Windows securely through RabbitMQ using TLS. This token is never stored, and it is completely forgotten once users disconnect or the application is closed.

# Data caching

## No user connected

If there are no users connected, no data or cache will be cleared. Every launched application will behave normally.

## Connected user

When a user signs in (through Launcher clients or directly in Launcher receiver), every application except Windows store apps will be executed under a private and temporary Windows user. This means that any cache or local data will be destroyed once the user disconnects. Local data does not include any file that a user could save in hard disk out of Windows user folders.

Launching Windows store apps will show an alert to communicate that any personal data will remain if it is not manually removed.

# Proximity detection and location services

## Bluetooth Low Energy

Proximity detection is carried out by Bluetooth Low Energy (BLE).

### Launcher receiver

Launcher receiver publishes BLE advertisement packets with a unique Launcher identifier, which is refreshed from our backend in each execution of the Launcher receiver. Note that Bluetooth must be enabled on the Windows device in order for the BLE function of Launcher receiver to operate.

### Launcher client

The iOS Launcher client apps listen for BLE advertisement packets, and when a Launcher packet is found the client app retrieves all information associated with that Launcher identifier from the backend. If a valid Launcher packet is found, it will appear as a nearby device on the Launcher client app.

### Proximity detection

If the client app exceeds a certain distance from the Launcher Windows application, it will stop receiving packets and therefore it will not detect any nearby Launcher receivers. Additionally, Launcher client apps will automatically disconnect from Launcher receivers upon exceeding the proximity limit.

## Location services

For the Launcher iOS client app, users need to allow location services requests in order to discover nearby Launcher receivers.

### Location services option: Always allow

The recommended option is ‘Always allow’, which will discover devices even when app is not running, or when the iOS device is locked. When ‘Always allow’ is enabled, a notification will appear for each device discovered.

### Location services option: Only while using the app

The option to allow location services ‘Only while using the app’ will only discover devices while the app is running in foreground sending to background the app will be managed as a moved away device.

# Microsoft Exchange integration

Launcher receiver can be configured to connect with an MS Exchange account in order to retrieve its calendar events, including rooms accounts if they are properly set up.

MS Exchange integration has been achieved using MS Exchange Web Services API (MS Exchange server development documentation). MS Exchange passwords are encrypted and stored in Launcher receivers’ application data.