# Mobility Assistance for People with Cognitive Disabilities (MAPCD) Interface Control Document

for the Smart Columbus Demonstration Program

**FINAL REPORT | July 26, 2018**

Produced by City of Columbus

## Notice



---

## Acknowledgement of Support

This material is based upon work supported by the U.S. Department of Transportation under Agreement No. DTFH6116H00013.

---

## Disclaimer



---

# Table of Contents



## List of Tables



## List of Figures

# Chapter 1. Introduction

## 1.1. PURPOSE

The purpose of this Interface Control Document (ICD) is to document the information required to effectively define the systems interfaces for the Mobility Assistance for People with Cognitive Disabilities (MAPCD) project. The intended audience for this document is the City of Columbus project team, the vendor development team, United States Department of Transportation (USDOT), and project stakeholders interested in understanding the system interfaces for the MAPCD project.

## 1.2. BACKGROUND

The Mobility Assistance for People with Cognitive Disabilities (MAPCD) project consists of an innovative smartphone application (WayFinder) which will be piloted for between 15 to 30 individuals in the Columbus region in partnership with the Central Ohio Transit Authority (COTA) and Ohio State University (OSU). WayFinder will enable persons with cognitive disabilities to transition off costly paratransit services and travel independently on the fixed-route bus system. Phone-based GPS tracking allows WayFinder to safely guide users with step-by-step visual and audio instructions. The WayFinder system was developed through research funding from the United States Department of Education (USDOE) and the National Institutes of Health (NIH). Data generated by the application will be written to the Smart Columbus Operating System (Operating System) to support data analysis and performance measures. For more information on the WayFinder system refer to https://www.ablelinktech.com/.

## 1.3. INTRODUCTION

This ICD describes the interchange of data between the Mobility Management Server (MMS) (source system) and the Operating System (target system), including relevant message structure and protocols that govern the interchange of data between these two systems. An estimate of the size and frequency of data exchange is provided as appropriate.

### 1.3.1. Assumptions

AbleLink Smart Living Technologies, the WayFinder vendor, will provide the MMS. The MMS will be a cloud-based application server that will provide real-time data to the Operating System. The data connection will be secured through an encrypted point-to-point virtual private network (VPN) tunnel between the MMS and the Operating System. AbleLink Smart Living Technologies will provide a RESTful interface to allow the data to be extracted in real-time.

### 1.3.2. Constraints

N/A

### 1.3.3. Risks

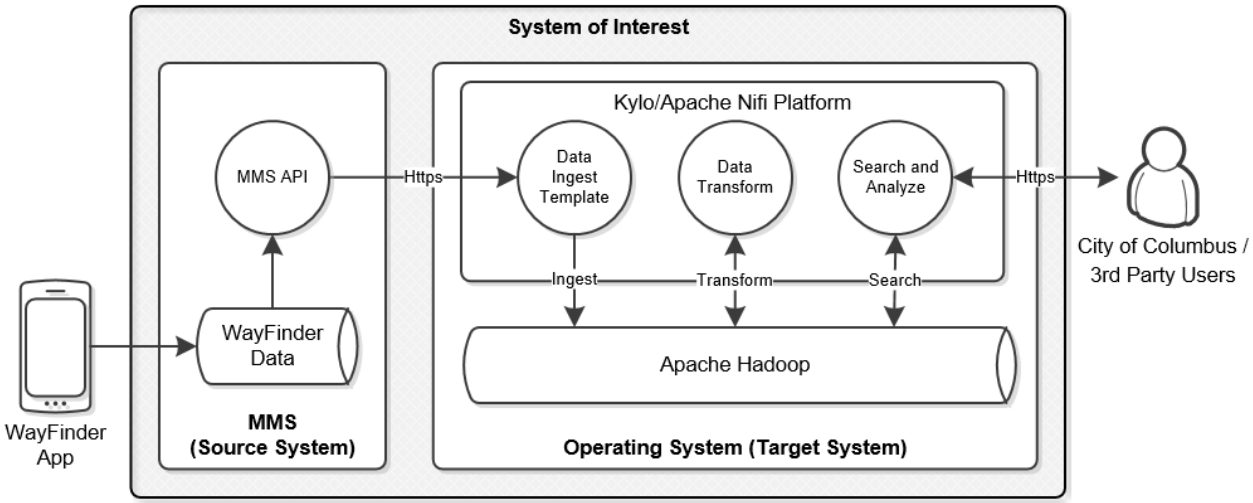
The MMS is currently in development by AbleLink Smart Living Technologies. As a result, the message structure for the various data elements of a trip (**Trip Data**) may continue to change as AbleLink Smart Living Technologies continues to develop and implement new features within the MMS. Agile development of the source system and target system will help to mitigate risks.

# Chapter 2. Interface Requirements

## 2.1. OPERATING SYSTEM OVERVIEW

The Operating System will utilize Apache NiFi and Kylo™ to ingest data from the MMS interface. The data will be transformed and stored in a Hadoop data lake, where it will be searchable utilizing Apache Hive to end-users of the system (City of Columbus and third-party users) through MMS Application Programming Interfaces (APIs).

**Figure 1: High-Level System Overview** describes the ingestion process at a high-level.




*Source: City of Columbus*

**Figure 1: High-Level System Overview**

### 2.1.1. Data Ingest Template

The Operating System development team will be responsible for creating the template (data structure/schema) to import data (**Trip Data**) from the MMS into the Hadoop Distributed File System (HDFS) tables. Apache Hive is a data warehousing application that provides logical access to data being stored in the Apache Hadoop infrastructure.

### 2.1.2. Data Transform

Once imported, trip data will be prepared for analytics using transformation features provided by Kylo. The Operating System development team will be responsible for data transformation that will be achieved in the Kylo front-end.

### 2.1.3. Search and Analyze

End-users of the system will be able to search the trip data in Hadoop and build Hive queries to support analysis. The Operating System development team will be responsible for establishing a search and query interface to allow end-users of the system to analyze the data.

## 2.2. MOBILITY MANAGEMENT SERVER (MMS) OVERVIEW

The MMS provides cloud-based storage for trip data generated by the mobile app. The mobile app provides geo-location-based multimedia instructions to individuals with cognitive disabilities to facilitate independent travel. Trip data generated by the mobile app is transmitted to the MMS in near-real time. This data is accessible to developers through a secure API.

### 2.2.1. Trip Data

Trip data is the information that is created by the WayFinder app while a user is in route from an origin to a destination. Trip data is transmitted from the WayFinder app to the MMS. The following data elements may be requested from the MMS via the MMS API. Detailed documentation on the following data elements is provided in **Table 1: Data Dictionary**.

Data elements:

- Starting a particular route
- Completing a route
- Canceling a route before completion
- Pausing a route
- Resuming a route
- Loss of GPS connection
- Reacquisition of GPS connection
- Leaving the travel corridor for the route
- Reentering the travel corridor for the route
- Requests for assistance by the traveler
- Downloading a route with mode of travel used
- Caregiver updates and communications

### 2.2.2. Communication Protocol

HTTPS will be the mandated protocol to protect the data in transit.

### 2.2.3. Authentication

The vendor will provide authentication credentials to the Operating System team to be used in all API calls. The MMS will not accept any API commands from a client without a proper authentication token, unless a particular API endpoint allows anonymous access.

### 2.2.4. Format

JSON will be the mandated format for the payload in the requests and responses from the MMS.

### 2.2.5. Personally Identifiable Information (PII)

PII will not be requested.

### 2.2.6. JSON Tags

**Table 1: Data Dictionary** describes the JSON tags available from the MMS API.

### Table 1: Data Dictionary

| Tag | Description | Type | Specific Values | Unique | Required | Example |
|---|---|---|---|---|---|---|
| Trip Data | | | | | | |
| #mmsapi_version# | Field identifying the version of MMS API that the current trip was completed with. | String | n/a | No | Yes | "#mmsapi_version#": "1.0" |
| trip_id | Unique identifier for the trip. | String | n/a | Yes | Yes | "trip_id": "acda1fc0130b4f99f1e577bf822" |
| user_id | Unique identifier for the traveler. | String | n/a | Yes | Yes | "user_id": "acda1fc0130b4f99f1e577bf822" |
| title | Descriptive title of the route. | String | n/a | No | Yes | "title": "Bus 25 - Center Mall to North Transfer Station" |
| routedownloaded | Date and time route was downloaded. | Unix epoch date value | Valid epoch date value, which represents the number of seconds that have elapsed since January 1, 1970. | No | Yes | "routedownloaded": 2522562400 |
| starttime | Start date and time for the current trip. | Unix epoch date value | Valid epoch date value, which represents the number of seconds that have elapsed since January 1, 1970. | No | Yes | "starttime": 4522562400 |
| completetime | Date/ time trip was completed. | Unix epoch date value | Valid epoch date value, which represents the number of seconds that have elapsed since January 1, 1970. | No | No | "completetime": 8522562400 |

| Tag | Description | Type | Specific Values | Unique | Required | Example |
|---|---|---|---|---|---|---|
| canceltime | Date/time the trip was cancelled. | Unix epoch date value | Valid epoch date value, which represents the number of seconds that have elapsed since January 1, 1970. | No | No | "canceltime": 1522562400 |
| assistancerequested | Value indicating the number of times assistance was requested during the trip. | Integer | n/a | No | No | "assistancerequested": 0 |
| caregiverrequest | Value indicating the number of times caregiver was contacted during the trip. | Integer | n/a | No | No | "caregiveralertrequest": 0 |
| Diagnostics Data | | | | | | |
| batterylevel | Value indicating the percentage value of battery power on the device at the time the data was posted to the MMS during the trip. | Integer | 0 - 100 | No | No | "batterylevel": 85 |

| Tag | Description | Type | Specific Values | Unique | Required | Example |
|---|---|---|---|---|---|---|
| gpsaccuracy | Relative value indicating the accuracy level of the GPS signal at the time the data was posted to the MMS during the trip. | Integer | 0 – 9 (9 represents the highest value) | No | No | "gpsaccuracy": 9 |
| routetype | Text description of the type(s) of travel included in the route, used in the Route Library to identify the route type. This value corresponds with the routetype value in the MMS Standard. | Comma-delimited string | autonomous vehicle, bus, bike, cable car, ferry, funicular, gondola, light rail, metro, rail, ride share, streetcar, subway, tram, walking, other | No | No | "routetype": "walking, bus" |
| cellnetworkcoverage | String value indicating the cell network coverage at the time the data was posted to the MMS during the trip. | String | excellent, very good, good, fair, poor, no signal | No | No | "cellnetworkcoverage": “excellent” |
| Performance Data | | | | | | |
| trippaused | Specific date/time(s) route was paused during trip. | Unix epoch date value in comma-delimited string | Valid epoch date value, which represents the number of seconds that have elapsed since January 1, 1970. | No | No | "trippaused": "1507827232, 1507837819, 1507849123" |

| Tag | Description | Type | Specific Values | Unique | Required | Example |
|---|---|---|---|---|---|---|
| tripresumed | Specific date/time(s) route was resumed during trip. | Unix epoch date value in comma-delimited string. | Valid epoch date value, which represents the number of seconds that have elapsed since January 1, 1970. | No | No | "tripresumed": "1507827232, 1507837819, 1507849123" |
| gpssignalloss | Specific date/time(s) gps signal was lost during trip. | Unix epoch date value in comma-delimited string | Valid epoch date value, which represents the number of seconds that have elapsed since January 1, 1970. | No | No | "gpssignalloss": "1507827232, 1507837819, 1507849123" |
| gpssignalreaquired | Specific date/time(s) gps signal was re-acquired during trip. | Unix epoch date value in comma-delimited string | Valid epoch date value, which represents the number of seconds that have elapsed since January 1, 1970. | No | No | "gpssignalreaquired": "1507827232, 1507837819, 1507849123" |
| Off-Route/Return to Route Data | | | | | | |
| offroutetime | Specific date/time(s) representing the time the traveler exited the travel corridor. | Unix epoch date value in comma-delimited string | Valid epoch date value, which represents the number of seconds that have elapsed since January 1, 1970. | No | No | "offroutetime": "1507827232, 1507837819, 1507849123" |

| Tag | Description | Type | Specific Values | Unique | Required | Example |
|---|---|---|---|---|---|---|
| offroutelat | Numeric value representing the latitude of the current specific location when the traveler left the travel corridor. | Real number in comma delimited string | Valid latitude in signed degrees format | No | Yes | "offroutelat": -104.845149625467 |
| offroutelong | Numeric value representing the longitude of the current specific location when the traveler left the travel corridor. | Real number in comma delimited string | Valid longitude in signed degrees format | No | Yes | "offroutelong": -104.845149625467 |
| returntoroutetime | Date/time(s) representing when the traveler returned to the route after being off route. | Real number in comma-delimited string | Unix epoch date value. | No | Yes | "returntoroutetime": 1507837819 |
| returntoroutelat | Numeric value representing the latitude of the current specific location the traveler returned to the route after being off route. | Real number | Valid latitude in signed degrees format | No | Yes | "returntoroutelat": -104.845149625467 |
| returntoroutelong | Numeric value representing the longitude of the current specific location the traveler returned to the route after being off route. | Real number | Valid longitude in signed degrees format | No | Yes | "returntoroutelong": -104.845149625467 |

*Source: City of Columbus*

# Appendix A. Acronyms and Definitions

**Table 2: Acronym List** contains project specific acronyms used throughout this document.

**Table 2: Acronym List**

| Acronym / Abbreviation | Definition |
|---|---|
| API | Application Programming Interface |
| HDFS | Hadoop Distributed File System |
| HTTPS | Hypertext Transfer Protocol Secure |
| ICD | Interface Control Document |
| JSON | JavaScript Object Notation |
| MAPCD | Mobility Assistance for People with Cognitive Disabilities |
| MMS | Mobility Management Server |
| N/A | Not Applicable |
| NIH | National Institutes of Health |
| OSU | Ohio State University |
| PII | Personally Identifiable Information |
| USDOE | United States Department of Education |
| USDOT | United States Department of Transportation |
| VPN | Virtual Private Network |

*Source: City of Columbus*

# Appendix B. Glossary

**Table 3: Glossary** contains project specific terms used throughout this document.

**Table 3: Glossary**

| Term | Definition |
|---|---|
| Authentication | The process or action of verifying the identity of a user or process. |
| Communication Protocol | A system of rules that allow two or more entities of a communications system to transmit information. |
| End-Users | City of Columbus and third-party users who will connect to the Operating System for trip data. |
| Operating System | The Smart Columbus Operating System is a cloud-based platform designed to ingest and disseminate data from external systems for processing via a microservices architecture. |

*Source: City of Columbus*

SMART COLUMBUS

THE CITY OF COLUMBUS

ANDREW J. GINTHER, MAYOR