|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Description** | **Reviser** | **Date** |
| A | Initial Draft | Axel Meyer | 24.02.2011 |
|  | Content increments | Jan Horstmann | 26.03.2011 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

System Specification

For

iGreen MachineConnector

Contents

[1.1. Scope 3](#_Toc286314207)

[1.2. Reference Documents 3](#_Toc286314208)

[1.3. Definitions 3](#_Toc286314209)

[1.4. Acronyms 3](#_Toc286314210)

[1.5. History 3](#_Toc286314211)

[2.1. General 4](#_Toc286314212)

[2.2. Data types 4](#_Toc286314213)

[2.2.1. Real time data 4](#_Toc286314214)

[2.2.2. Documentation data / Task data 4](#_Toc286314215)

[2.2.3. Configuration data 4](#_Toc286314216)

Tables

[Table 1.1 - Definitions 3](#_Toc286314217)

[Table 1.2 – Acronyms 3](#_Toc286314218)

1. General

# Scope

This specification defines how the communication between a machine terminal (Embedded Display, Mobile Device, Automotive Computer) and the iGreen OnlineBox is working. Also the iGreen Machine Connector should be able to allow a Machine to Machine (M2M) communication. This specification must be in a detailed level which allows independent implementations of the machine connector communicating with each other. One implementation of this specification is done by DFKI as a part of the iGreen project.

# Reference Documents

For details on J1939 communication see the common available specification for J1939.

The ISOBUS and ISOXML standard defined in ISO11783 is as well referenced in this specification.

# Definitions

These are the definitions that are common to all documents in the Document. Despite other definitions that may exist for these terms, they are always to be used in this document as defined below to avoid confusion.

| **Term** | **Definition** |
| --- | --- |
| **Terminal** | Embedded Machine Terminal, Cell phone device, Automotive Computer |

Table 1.1 - Definitions

# Acronyms

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| REST | Representational State Transfer |
| CouchDB | Apache CouchDB |

Table 1.2 – Acronyms

# History

* 24.02.2011 – Initial release
1. Specification

# General

The iGreen MachineConnector team has decided in the telephone conference at the 24.02.2011 to use a REST interface based communication interface. This technology allows to communicated with an OnlineBox directly as well as with an on the terminal installed Apache CouchDB. The CouchDB allows having an offline cache which is synchronized bidirectional with other terminals and the OnlineBox.

# Data types

## Real time data

Bei den Echtzeitdaten handelt es sich um Statusinformationen von der Maschine welche schnellst möglich der iGreen Wolke zur Verfügung stehen sollen. Unter Echtzeit wird jedoch nur bezeichnet dass die Daten für die Anwendung sinnvollen Updatezyklen gesendet werden um das Datenvolumen nicht in die Höhe zu treiben. Außerdem soll von diesen Daten keine Historie in der Datenbank gespeichert werden, sondern immer nur die letzte Information. Daten mit Historie werden ausschließlich in Form von ISOXML Daten in der Dokumentation abgespeichert.

## Documentation data / Task data

Diese Art von Daten stellt einen kompletten Dokumentationsdatensatz dar, bzw. so wie es der ISOXML Standard vorsieht eine zwischengespeicherte ISOXML Datei, damit man z.B. einen Task der mehrere Tage dauert auch schon zwischendurch beurteilen kann. Die ISOXML Tasks können vor Beginn angelegt werden (z.B. von der Desktop Software) oder im Nachhinein nach Abschluss (Zwischenspeichern) eines Tasks. Eine Datenübermittlung zwischendurch ist nicht vorgesehen.

## Configuration data

Die Konfigurationsdaten beschreiben mit welchen iGreen Nodes (anderen Maschinen Agenten) synchronisiert wird und welche Verbindungen bevorzugt verwendet werden. Außerdem werden die ACL’s dort gespeichert. Außerdem sind dort Basis Informationen zur Maschine hinterlegt (z.B. Seriennummer). Da die Konfigurationsdaten mit synchronisiert werden stehen die Datensätze auf allen anderen Maschinenagenten auch zur Verfügung und die Fern-Konfigurierbarkeit ist gegeben.

# Functions

## Setup

Die Funktion Setup übernimmt das Setzen der Konfigurationsdaten. Zu den Konfigurationsdaten zählen

* OnlineBox [Pflichteingabe, mandantory]
* Benutzername [Pflichteingabe, mandantory]
* Kennwort [Pflichteingabe, mandantory]
* Domain [Pflichteingabe, mandantory]
* Device-ID (eineindeutige Gerätekennung gemäß ISOBUS) [optional]
* Zwischenspeicher der Daten [Cookies]

## Login

Die Funktion Login sorgt für die Authentifizierung des Endgeräts an der OnlineBox. Dieser Funktion müssen folgende Parameter zur Verfügung gestellt werden

* OnlineBox
* Benutzername
* Kennwort
* Domain

Alternativ können diese Informationen aus dem Zwischenspeicher [Cookie] herangezogen werden.

Ein erfolgreicher Anmeldevorgang kennzeichnet sich dadurch, dass ein definierter Text zurückgeliefert wird.

## Logout

Die Funktion Logout sorgt für die Abmeldung des Endgeräts von der OnlineBox. Dieser Funktion müssen folgende Parameter zur Verfügung gestellt werden

* OnlineBox
* Domain

Alternativ können diese Informationen aus dem Zwischenspeicher [Cookie] herangezogen werden.

## Create\_Bucket / Speicherbereich anlegen

Die Funktion Create\_Bucket legt einen eindeutig definierten Speicherbereich innerhalb der OnlineBox an. Bei der Anlage des Speicherbereichs durch das Endgerät werden automatisch die zur Verfügung stehenden Metadaten

* Datum
* Uhrzeit
* Benutzer

zu dem Speicherbereich referenziert und lassen sich später für eine Suche nutzen.
Um die Function Create\_Bucket aufzurufen, muss eine gültige Session mit der Function login aufgebaut sein. Das angemeldete Endgerät muss Schreiberechte auf der OnlineBox besitzen. Die Function Create\_Bucket wird mit folgenden Parametern ausgeführt:

* Aktuelle Session-ID aus dem Zwischenspeicher [Cookie]

Das Ergebnis dieser Function ist eine eineindeutige URI, die auf den angelegten Speicherbereich verweist. Im Fehlerfall muss ein definierter Text ausgegeben werden.

## Close\_Bucket / Speicherbereich schließen

Die Funktion Close\_Bucket sperrt den Speicherbereich ab. Nach Ausführung dieser Funktion sind keine inhaltlichen Änderungen und Ergänzungen möglich. Dieser Vorgang wird ebenfalls dokumentiert in den Metadaten mit folgenden Attributen

* Datum
* Uhrzeit
* Benutzer

zu dem Speicherbereich referenziert und lassen sich später für eine Suche nutzen.
Um die Function Close\_Bucket aufzurufen, muss eine gültige Session mit der Function login aufgebaut sein. Das angemeldete Endgerät muss Schreiberechte auf der OnlineBox besitzen. Die Function Close\_Bucket wird mit folgenden Parametern ausgeführt:

* Aktuelle Session-ID aus dem Zwischenspeicher [Cookie]
* URI des zu schließenden Buckets [Pflichteingabe, mandantory]

Das Ergebnis dieser Function ist eine definierte Textmeldung. Im Fehlerfall muss ein definierter Text ausgegeben werden.

## Upload\_ISOXML / Taskdata in ein Bucket hochladen

Die Funktion Upload\_ISOXML speichert ein ZIP-Archiv mit einer Taskdata.xml sowie den zugehörigen \*.xml und \*.bin-Dateien in einem zuvor defninierten Bucket. Dieser Vorgang wird ebenfalls dokumentiert in den Metadaten mit folgenden Attributen

* Datum
* Uhrzeit
* Benutzer

zu dem Speicherbereich referenziert und lassen sich später für eine Suche nutzen.

Um die Function Upload\_ISOXML aufzurufen, muss eine gültige Session mit der Function login aufgebaut sein. Das angemeldete Endgerät muss Schreiberechte auf der OnlineBox besitzen. Die Function wird mit folgenden Parametern ausgeführt:

* Aktuelle Session-ID aus dem Zwischenspeicher [Cookie]
* URI des Ziel-Buckets [Pflichteingabe, mandantory]
* Name und Speicherort der ZIP-Archivs mit den ISO-Inhalten [Pflichteingabe, mandantory]
* Beschreibung der Datei [optional]
* Status [optional]

Das Ergebnis dieser Function ist eine definierte Textmeldung. Im Fehlerfall muss ein definierter Text ausgegeben werden. Nach dem Upload wird die Datei im Hintergrund in der OnlineBox verarbeitet und entsprechend ausgewertet.

## ISOXML\_Search / Suche nach ISOXML-Dateien

Um die Function ISOXML\_Search aufzurufen, muss eine gültige Session mit der Function login aufgebaut sein. Das angemeldete Endgerät muss Leserechte auf der OnlineBox besitzen. Die Function wird mit folgenden Parametern ausgeführt:

* Startzeit des Tasks [nur nach Tasks gehen ist nicht zielführend, da n-Tasks pro Auftragsset]
* Endzeit des Tasks
* Description des Tasks
* Status des Tasks
* DVC [Maschine]

Das Ergebnis dieser Function ist eine Bucket-URI und ein Dateiname.

## Download\_Data / Herunterladen von Dateien

Um die Function Download\_Data aufzurufen, muss eine gültige Session mit der Function login aufgebaut sein. Das angemeldete Endgerät muss Leserechte auf der OnlineBox besitzen. Die Function wird mit folgenden Parametern ausgeführt:

* Bucket-URI [z.B. durch ISOXML\_Search ermittelt]
* Dateiname [z.B. durch ISOXML\_Search ermittelt]

Das Ergebnis dieser Function ist die Datei in dem Standardverzeichnis gemäß Setup.

# Flow Diagramms / Ablaufdiagramme

## Upload Taskdata to Onlinebox



Der erste Schritt ist das Hochladen eines Tasks auf die Onlinebox. Mit Hilfe einer Task Management Software wird eine Taskdatei erzeugt und in als zip-File komprimiert. Diese hat den Status intial. Um eine bessere Lesbarkeit zu ermöglichen sollte das Filenaming wie folgt vorgenommen werden. Das Filenaming hat keinen Einfluss auf die Suchfunktionen.

Bucket-ID-Terminologie:

* Land\_iGreenUser\_Taskstatus\_Terminal-ID\_Date
* Land\_iGreenUser\_RealTimeInfo\_Terminal-ID

Bucket-ID-Terminologie Beispiele:

* DE\_LU1\_INITIAL\_KR0815\_2011\_02\_10
* DE\_LW2\_PAUSED\_GR0916\_2011\_02\_11
* DE\_XY7\_FINISHED\_JD0917\_2011\_02\_12
* DE\_ZZ9\_RealTimeInfo\_CL0918

Das weitere Vorgehen ist im Ablaufdiagramm beschrieben und ist in den Beispielbatchdateien ersichtlich.

## Search and download Taskdata from Onlinebox



Das weitere Szenario beinhaltet das Suchen und Herunterladen eines Tasks auf den Machine-Connector (MC) und das Übertragen auf den Task-Controller. Das Übertragen des Tasks vom MC auf den Task-Controller ist herstellerspezifisch zu lösen, da hier unterschiedliche Terminals verwendet werden.

Die Search-Parameter enthalten den Auftragsstatus (initial, paused), die MaschinenId (DVC) und die Uploadzeit des Buckets.

Alle Maschinen überprüfen in zu definierenden Zeitabständen (ca. eine Minute) ob ein neuer Auftrag für sie vorliegt. Ist ein entsprechender Auftrag vorhanden wird dieser heruntergeladen und in das Terminal übertragen.

## Taskdata transfer between two machines



Eine Anforderung des Bundesverbands Lohnunternehmen ist ein herstellerübergreifender Austausch von Taskdaten.

Der Disponent oder Lohnunternehmer erzeugt mit einer Taskmanagement Software ein Auftragsset (Task) und überträgt diese wie im Ablaufdiagramm 2.4.1. beschreiben auf die Onlinebox. Der MC der Maschine 1 lädt diese wie im Search and download Ablaufdiagramm beschreiben herunter und beginnt den Auftrag.

Aufgrund nicht näher beschriebenen Ursachen wird die Maschine 1 abgezogen und muss den Auftrag pausieren (Status: paused) und auf die Onlinebox übertragen.

Der Disponent muss mit Hilfe des Dispotools (FH Bingen) diesen Auftrag um planen und wieder auf Onlinebox bereitstellen. Der Disponent muss festlegen welche Maschine den Auftrag fortsetzen und beenden soll.

Die Maschine 2 lädt sich den pausierten Auftrag herunter und setzt ihn fort. Nachdem der Auftrag vollständig abgearbeitet ist wird der Task beendet und auf die Onlinebox transfriert.

Beide Maschinen müssen von unterschiedlichen Herstellern stammen.