Ziel

Die nachfolgend beschriebene Applikation, Auto-Update-Center für icom OS, dient zur Pflege, Erstellung und Auslieferung von so genannten Update-Paketen für icom OS basierende Geräte. Zweck dieses Systems ist die kontrollierte Verteilung von Firmware- und Software-Updates, als auch von Geräte-Konfigurationen, Zertifikaten und Lizenzen. Die Anwendung unterstützt bei der Definition von Update-Paketen, ist verantwortlich für die kontrollierte Ablage der Dateien auf Basis eines S3-kompatiblem Speichersystems und kann sowohl über eine Web-basierende Benutzeroberfläche, als auch über eine API bedient werden.

Produkteinsatz

Das Auto-Update-Center für icom OS ist wesentlicher Bestandteil der neuen Operation-, Administration- & Management-Plattform für INSYS icom Smart Devices. Aus dem Device-Management heraus, werden mit Hilfe des Auto-Update-Centers, Firmware- und Software Updates auf Geräte ausgerollt, wie auch Veränderungen der Konfiguration auf die betroffenen Geräte angewendet, bis hin zur Verteilung von Zertifikaten und Lizenzen.

Ein Einsatz in der Produktion der INSYS Geräte ist ebenso denkbar, um Kunden-individualisierte Werkskonfigurationen automatisiert auf Geräte aufzuspielen.

Produktfunktionen

Einleitung

Die Applikation Auto-Update-Center für icom OS basiert auf der bestehenden Technologie namens „Automatische Updates“ (nachfolgend Auto-Update Verfahren genannt). Ein icom OS-basierender Router verbindet sich dafür mit dem Update-Server und lädt dort – falls vorhanden – ein Update-Paket herunter. Im Anschluss wird dieses Update-Paket auf dem Gerät entpackt und die Inhalte entsprechen angewendet bzw. installiert.

Auto-Update Verfahren

icom OS-basierenden Geräte können über das so genannte Auto-Update Verfahren Firmware-Updates, Geräte-Konfigurationen, etc. laden und installieren. Der Zugriff erfolgt wahlweise zyklisch oder der Auto-Update wird manuell, durch den Anwender, ausgelöst. Auf dem Gerät werden dafür ein oder mehrere Update-Server hinterlegt. Sobald das Auto-Update Verfahren startet, werden alle aktiven Update-Server der Reihe nach abgefragt. Steht ein Update-Paket zur Verfügung, wird dieses geladen und ausgeführt. Danach wird der nächste verfügbare Update-Server aus der Liste kontaktiert und verarbeitet. Wurden alle Update-Server abgearbeitet und die Update-Pakete erfolgreich angewendet, ist das Auto-Update Verfahren abgeschlossen.

Achtung: Ein Update-Paket kann einen Neustart initiieren. Erfolgt der Zugriff auf einen weiteren Update-Server, wird der Neustart für eine bestimmte Zeit unterdrückt. Läuft jedoch die Zeit ab, während noch ein weitere Update-Paket geladen oder installiert wird, erfolgt trotzdem der Neustart des Geräts!

Update-Paket

Ein Update-Paket enthält wahlweise Firmware-Updates, Geräte-Konfigurationen, Zertifikate, Lizenzen oder Container-Abbilder (icom SmartBox Software). Diese (nachfolgend auch Artefakt genannt) werden zusammen mit einer MANIFEST-Datei zu einem TAR-Archiv gebündelt. Die Manifest-Datei beschreibt die Merkmale der einzelnen Artefakte und legt zudem auch die Reihenfolge der Abarbeitung fest.

Eine detaillierte Beschreibung des Update-Pakets ist der icom OS Online-Hilfe zu entnehmen: Hilfe → Dokumentation → Online-Hilfe → Update-Pakete

Update-Server

Die Auslieferung von Update-Paketen erfolgt über HTTP (alternativ auch FTP, aber im Rahmen dieses Projektes nicht gewünscht), weswegen es sich beim Update-Server grundsätzlich um einen HTTP-basierenden Server-Prozess handelt. Damit ein Gerät das für ihn relevante Update-Paket identifizieren kann, wird im ersten Schritt ein Update-Index geladen. Der Update-Index enthält eine Liste mit Seriennummern und den dazugehörigen Update-Paketen, in Form eines absoluten Pfads für den Download-Prozess. Wahlweise kann auch ein so genannter Wildcard-Eintrag vorgenommen werden, um alle Geräte zu adressieren. Grundsätzlich gilt, der erste zutreffende Verweis in der Liste wird verarbeitet. Das hinterlegte Update-Paket wird vom Update-Server geladen und im Anschluss ausgeführt.

Information: Die Angabe einer vollständigen URL im Update-Index ist nicht zulässig. Das Update-Paket muss vom Update-Server selbst ausgeliefert werden, weswegen der Verweis einem absoluten Download-Pfad entspricht.

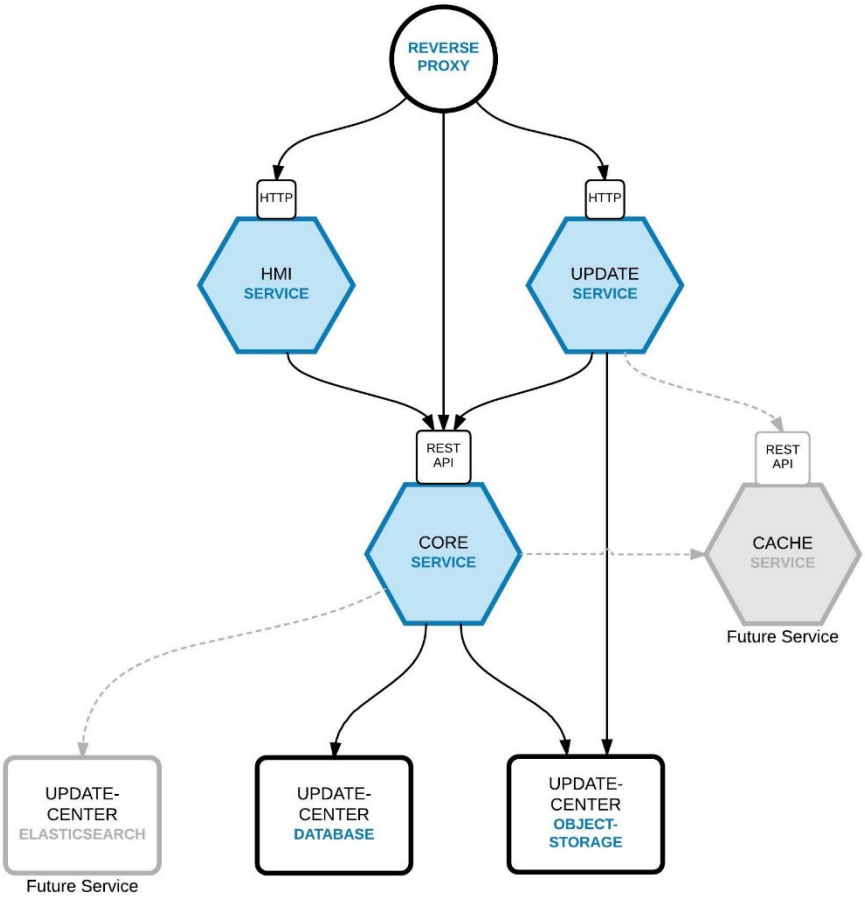
Information: Nachdem ein Update-Paket heruntergeladen und installiert wurde, erfolgt **keine** Rückmeldung an den Server. Der Update-Server liefert lediglich die Update-Pakete aus. Im icom OS Auto-Update Verfahren ist derzeit kein Mechanismus vorgesehen, um den Status und damit den Erfolg des Updates zu dokumentieren.

Information: Bevor das Gerät den Download eines Update-Pakets einleitet, werden die HTTP-Felder ETag, Last-Modified und Content-MD5 ausgewertet. Wurde das Update-Paket bereits erfolgreich geladen und angewendet, und das Update-Paket auf dem Update-Server ist seitdem unverändert, erfolgt keine erneute Verarbeitung. Dieses Cache-Konfiguration gilt es im Auto-Update-Center unbedingt beizubehalten!

Weitere detaillierte Informationen zum Auto-Update Verfahren ist der icom OS Online-Hilfe zu entnehmen: Hilfe → Dokumentation → Online-Hilfe → Automatisches Update

Service Architektur

Die Auto-Update-Center Applikation besteht aus drei Haupt-Komponenten, wie dem nachfolgenden Diagramm zu entnehmen ist.



Core Service

Das Herzstück der Anwendung ist der Core-Service. Innerhalb dieser Komponente sind alle Operationen und Funktionen des Auto-Update-Centers implementiert.

Update Service

Diese Komponente enthält alle Operationen, die für den Zugriff durch das Gerät, und somit für das Auto-Update Verfahren erforderlich sind. Mit steigender Anzahl an Geräten, soll dieser Service unabhängig von den anderen Komponenten skalieren können.

HMI Service

Die UI des Auto-Update-Centers dient zur Konfiguration & Parametrisierung der Applikation.

Information: Im Rahmen des OAM Projektes eignet sich die Benutzeroberfläche lediglich für administrative Tätigkeiten. Das Device Management übernimmt hingegen die Parametrisierung des Auto-Update-Centers über die REST-Schnittstelle.

Cache Service (Future Service)

Um die Last der Datenbank zu reduzieren, werden im Cache Service alle Daten zur Auslieferungen von Update-Paketen vorgehalten. Der Update-Service kontaktiert die Datenbank nur noch, wenn im Cache keine passenden Daten vorliegen. Der Cache Service wird vom Core entsprechend mit Daten angereichert.

Elasticsearch (Future Service)

Die Einbindung von Elasticsearch dient ebenfalls zur Entlastung der Datenbank, als auch um die so genannte User-Experience zu verbessern.

Anwendung des Object-Storage-Systems

Sämtliche Update-Pakete bestehen aus einzelnen Artefakten; angefangen bei Binär-Dateien (Firmware, Software) bis hin zu Konfigurationsdateien, Zertifikaten und Lizenzen. Diese Dateien sollen in einem S3-kompatiblen Objekt-Speicher abgelegt werden, statt lokal auf einem Dateisystem verwaltet zu werden. Dem Kunden steht es frei, seinen eigenen Objekt-Speicher – zum Beispiel Amazon AWS S3 – einzusetzen oder den internen Speicher vom Auto-Update-Center zu nutzen.

Der Einsatz eines Objekt-Speichers bietet folgende Vorteile im Vergleich zur Ablage in einem Dateisystem:

- Zugriff auf den Speicher erfolgt über eine definierte API
- Dateisystem- und Zugriffskonflikte sind ausgeschlossen
- Hohe Verfügbarkeit der Daten ist einfacher (und kostengünstiger) zu implementieren
- Bessere Skalierbarkeit, bei hohem Aufkommen an Zugriffen

Die Ablage der Daten erfolgt hierarchisch mit Hilfe der Objekt-Präfixe (vergleich mit einem Verzeichnis). Grundsätzlich wird zwischen einem Artefakt Speicher (Artificat-Repository) und einem Update-Package Speicher (Package-Repository) unterschieden. In Ersteres kann der Anwender beliebige Dateien zur Wiederverwendung ablegen. Im Package-Repository befinden sich dagegen ausschließlich die Inhalte eines Update-Pakets und das automatische erzeugte Update-Paket selbst.

Achtung: Bei Nutzung des internen Speichers wird pro Mandant ein eigener Bucket erzeugt. Die Organisation der Repositorys erfolgt ausschließlich über die Objekt-Präfixe

Soll ein neuer Speicher erstellt werden, darf der Speicherort (Objekt-Präfix) noch nicht existieren. Bei Erzeugung der Ablage oder des Update-Pakets wird immer eine leere Meta-Datei zur Kennzeichnung hochgeladen:

- `__ARTIFACT_REPOSITORY__`
- `__PACKAGE_REPOSITORY__`
- `__PACKAGE__`

Information: In Zukunft sollen diese Meta-Dateien tatsächlich mit verschiedenen Informationen angereichert werden.

In der Artefakte-Ablage organisiert der Anwender seine Dateien selbst. Ein Update-Paket hat hingegen immer nachfolgende, beispielhafte Struktur:

Objekt-Präfix	Objekt-Name	Bemerkungen
MeineAblage / icomOS28FullActivate	__PACKAGE__	Automatisch hinzugefügt
MeineAblage / icomOS28FullActivate	update-2.0-full.bin	

MeineAblage / icomOS28FullActivate	update-2.0-to-2.8.bin	
MeineAblage / icomOS28FullActivate	ascii.txt	ASCII Konf. zum Aktivieren der F/W
MeineAblage / icomOS28FullActivate	icom_os28_full_activate.tar	Generiertes Update-Paket

"MeineAblage" im obigen Beispiel entspricht dem Package-Repository. Wird der interne Speicher angewendet ergibt sich folgende Hierarchie (Tenant-ID = 5f9ac918):

Bucket	Objekte
5f9ac918	- MeineAblage \
	__PACKAGE_REPOSITORY__
	- icomOS2.8FullActivate \
	__PACKAGE__
	update-2.0-full.bin
	…

Beim Erzeugen eines Update-Pakets können die Dateien direkt über das HMI oder die API hochgeladen werden. Alternativ kann ein bestehendes Objekt aus einem Repository (Artefakt- oder Update-Packet Ablage) in den Update-Paket Ordner **kopiert** werden. Vorausgesetzt das Quell-Repository wird im gleichen Objekt-Speicher gepflegt.

Mehr Informationen zum Objekt-Speicher und dessen Organisation folgen im Kapitel: Datenbank-gestützte Definition der Update-Pakete

Multi-Tenancy (Mandantenfähigkeit)

Die Applikation wird von mehreren Kunden genutzt, weswegen eine Mandantenfähigkeit unumgänglich ist. Je Kunde eine separate Auto-Update-Center Instanz zu starten, birgt aber viele Nachteile, darunter:

- Pflege von zentralen Eigenschaften (z.B. MANIFEST File-Type)
- Eindeutige Service-Zuordnung zu Kunde
- Programmatischer Eingriff in den Frontend-Webserver (oder Load-Balancer)
- Komplexität beim Zugriff aus der UI heraus
- Update der Instanzen

Im Rahmen des Auto-Update-Center Projektes soll der Ansatz verfolgt werden, je Kunde lediglich ein eigenes Datenbank-Schema zu generieren. Zentrale Tabellen können dagegen im Standard-Schema geführt werden und bei Bedarf sogar von Tabellen aus den einzelnen Schemas referenziert werden.

Beim Zugriff über die REST-Schnittstelle wird per OAuth-Token der Mandant ermittelt und das entsprechende Kundenschema adressiert. Für den Geräte-Zugriff (Auto-Update Verfahren) können je Mandat eigene Basic-Auth Zugangsdaten hinterlegt werden. Im Falle des Managed-Service Auto-Update-Center ist die Absicherung durch eine Basic-Auth immer erforderlich. Handelt es sich dagegen um eine On-Premise Installation oder um den Bootstrap-Service kann auf die Authentifizierung verzichtet werden. Grundsätzlich erfolgt Unterscheidung des

Mandanten, beim Zugriff durch das Gerät, immer über den URL Pfad, welche die Mandanten-ID enthält.

Information: In Phase-1 – d.h. der Dienst steht nur ausgewählten Kunden zur Verfügung und wird noch nicht öffentlich betrieben – wird die REST-Schnittstelle ohne OAuth-Absicherung betrieben. Die sichere Unterscheidung des Mandanten kann demnach nicht über das OAuth-Token erfolgen! Der Zugriff auf die REST-Schnittstelle erfolgt deswegen mit dem zusätzliche HTTP-Header Attribute `X-INSYS-icom-Tenant-ID`, um den Mandanten eindeutig zu referenzieren.

Das Datenbankmodell für die Pflege der Mandanten sieht wie folgt aus:

Tenant	
PK	ID
UQ	Title
UQ, IX	Slug
UQ	Schema-Name
	Admin-Email-Address
	Update-Server-Username
	Update-Server-Password-Hash
	Is-Active

Die Spalte Slug entspricht einer URL-freundlichen Notation der Mandanten Bezeichnung. Dieser Wert ist für die Update-Server URLs erforderlich.

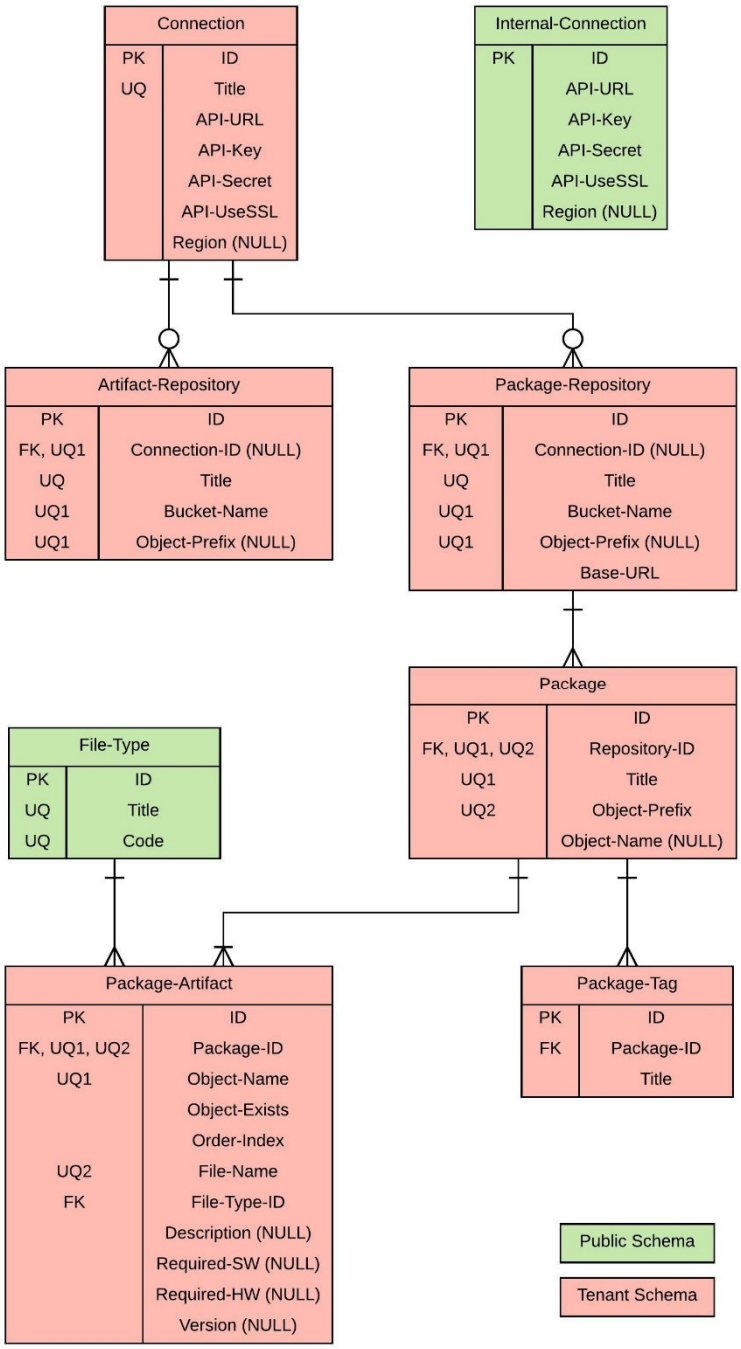
Achtung: Bei Installation der Anwendung sollte ein „default tenant“ angegeben und das entsprechende Datenbank-Schema erstellt werden. Viele Applikation-Frameworks arbeiten mit so genannte Object-Relational-Mappern. Diese erfordern meistens das Vorhandensein eines Datenbank-Schemas.

Datenbank-gestützte Definition der Update-Pakete

Die primäre Eigenschaft des Auto-Update-Centers ist die Datenbank-gestützte Definition von Update-Paketen. Im Wesentlichen wird dabei der Inhalt der MANIFEST-Datei eines Update-Pakets in einer Datenbank-Struktur festgehalten. Der Inhalt des Update-Pakets kann somit jederzeit einfach verändert und vervielfacht werden.

Die Artefakte eines Update-Pakets werden auf einem S3-kompatiblen Objekt-Speicher abgelegt. Der Verweis auf das entsprechende S3-Objekt wird ebenfalls in der Datenbank hinterlegt. Sobald ein Update-Paket angefordert wird, wird auf Basis der Datenbank-Inhalte die MANIFEST-Datei erstellt, die entsprechenden S3-Objekte vom S3-Objekt-Speicher geladen und das TAR-Archiv generiert und zum Download bereitgestellt.

Nachfolgendes Datenbank-Modell beschreibt die Struktur zur Definition von Update-Paketen:



Connection & Internal-Connection

Im Connection Objekt sind die Verbindungsdaten zu einem S3-kompatible Object-Storage-System hinterlegt. Pro Mandant können beliebig viele Verbindungen gepflegt werden.

Das Internal-Connection Objekt enthält die Verbindungsdaten zum so genannten „Internal-Storage“. Die Auto-Update-Center Applikation stellt einen eigenen Objekt-Storage-Speicher bereit, falls der Kunde keine externen Speichersysteme einsetzen will.

Information: Für den „Internal-Storage“ wird der Minio Object-Storage-Server eingesetzt. Dabei handelt es sich um eine Open-Source und Amazon S3-kompatible Applikation. Minio wird in Form von so genannten Docker Container betrieben. Darüber hinaus kann das System

bedarfsgerecht skaliert und die Daten hochverfügbar bereitgestellt werden. Mehr Informationen zum Produkt sind nachfolgendem Link zu entnehmen: www.minio.io

Information: Für den Zugriff auf die S3-kompatiblen Objekt-Speicher-Systeme bietet sich zu dem an, das Minio Client SDK einzusetzen. Mehr Infos gibt es hier: https://docs.minio.io/

Artificat- & Package-Repository

Zur besseren Organisation der Update-Pakete und Artefakte, kann der Kunden beliebig viele Ablagen (Repository) erstellen. Eine Ablage entspricht entweder einem eigenständigen Objekt-Speicher Bucket oder wird in einem bestehenden Bucket erzeugt, dann aber mit Angabe eines eigenen und eindeutigen Verzeichnisses (Objekt-Präfix). Ein bereits vorhandener Speicherort, kann nicht als neue Ablage genutzt werden!

Die Angabe einer Verbindung ist dagegen optional. Wird ein Repository ohne Verweis auf eine Connection-ID erstellt, werden die Daten immer im so genannten „Internal-Storage" abgelegt. Der interne Speicher entspricht ebenfalls einem S3-kompatiblen Object-Storage-System, welches das Auto-Update-Center selbst bereitstellt. Um zwischen den Mandanten unterscheiden zu können, wird je Kunde ein eigener Bucket mit der Mandantennummer (Tenant-ID) als eindeutige Bucket Bezeichnung erstellt.

Achtung: Bei Nutzung des internen Speichers ist die automatische Generierung oder direkte Angabe des eindeutigen „Object-Prefix" für das Repository immer Pflicht. Die Angabe der Bucket Bezeichnung ist dagegen unterbunden, da dieser der Tenant-ID entspricht.

Package & Package-Tag

Die Definition eines neuen Update-Pakets erfolgt über die Tabelle Package. Der Ablage-Ort der Artefakte wird durch die Angabe des Repositorys festgelegt. Sämtliche Inhalte eines Update-Pakets müssen immer innerhalb eines eigenen Verzeichnisses abgelegt werden. Der Objekt-Präfix (entspricht dem Verzeichnis) wird demnach automatisch generiert, kann aber auch durch den Anwender selbst bestimmt werden.

Der Dateiname des automatisch erstellten Update-Pakets, welches schlussendlich dem Gerät bereitgestellt wird, ist im Feld Object-Name gespeichert. Diese Datei wird allerdings erst beim erstmaligen Zugriff durch das Gerät vom Auto-Update-Center generiert und das Objekt in der gleichen Ablage, wo die Inhalte der Update Pakets hinterlegt sind, gespeichert. Im Anschluss wird dann erstmalig der Objekt-Name in der Tabelle gesetzt und erfolgt ein weiterer Zugriff auf das Update-Paket, ist die erneute Erstellung des Update-Pakets nicht mehr erforderlich. Das Auto-Update-Center kann das Objekt mit dem hinterlegten Namen direkt vom Objekt-Speicher-System laden und ausliefern. Werden Veränderungen an der Definition des Update-Paket vorgenommen, wird das Feld Object-Name wieder auf Null zurückgesetzt. Somit wird dem Auto-Update-Center signalisiert, das Update-Paket vor der Auslieferung neu zu erstellen.

Darüber hinaus kann die Update-Paket Definition mit so genannten Tags gekennzeichnet und damit organisiert werden. Die Tags zu einem Update-Paket werden in der Package-Tag Tabelle gespeichert.

Tipp: Um die Suche nach Paketen, insbesondere nach Tags oder Inhalten zu vereinfachen und die Datenbank zu entlasten, wäre der Einsatz von Elasticsearch lohnenswert. In einem Elasticsearch Index könnte man ein Dokument „Package" definieren und alle relevanten Daten darin hinterlegen.

Package-Artifact & File-Type

Die einzelnen Artefakte bzw. Bestandteile eines Update-Pakets werden in der Package-Tabelle hinterlegt. Der Object-Name entspricht dem tatsächlichen Namen des Datei-Objekts im Object-Storage-System. Alle folgenden Informationen entsprechen dagegen den Merkmalen die in er MANIFEST-Datei hinterlegt werden. Mittels Order-Index wird die Reihenfolge der Artefakte innerhalb der MANIFEST-Datei festgelegt.

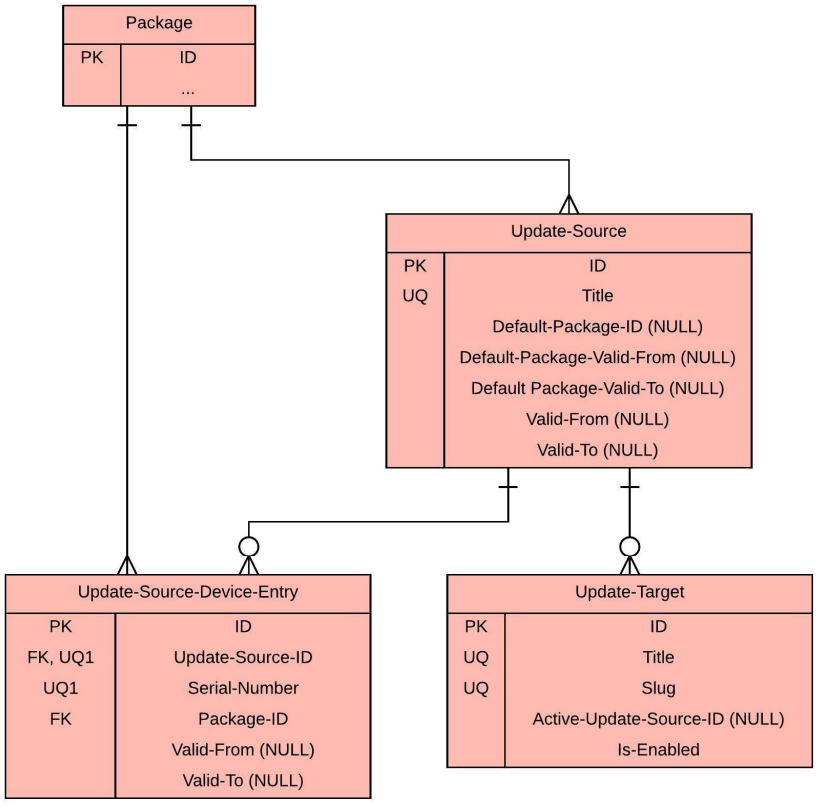
Die File-Types – siehe icom OS Online-Hilfe zum Thema Update-Pakete, Merkmal: FILETYPE – werden global gepflegt, da diese von uns festgelegt werden müssen.

Achtung: Sobald neue Artefakte der Paket-Definition hinzugefügt werden, oder bestehende geändert oder gelöscht werden, muss in der Tabelle Package die Spalte Object-Name auf Null gesetzt werden. Dieser Vorgang löst die Neuerstellung des Update-Pakets beim Gerätezugriff aus.

Datenbank-gestützte Auslieferung der Update-Pakete

Gemäß Auto-Update Verfahren, lädt ein Gerät im ersten Schritt eine so genannte Download-Liste. Diese Liste enthält Zeilen bestehend aus jeweils zwei Spalten (komma-separiert). Die erste Spalte enthält die Serien-Nummer eines Gerätes oder das Wildcard Symbol (*). Die zweite Spalte den absoluten Pfad zu einem Update-Paket.

Das dazugehörige Datenbankmodell sieht wie folgt aus:



Die Definition einer Download-Liste erfolgt in der Tabelle Update-Source. Die einzelnen Zeilen werden in der Tabelle Update-Source-Device-Entry hinterlegt, mit Ausnahme dem Wildcard-

Eintrag. Der Wildcard-Eintrag adressiert alle Geräte und wird deshalb als optionales Default-Package in der Tabelle Update-Source geführt.

Wichtig: Bei Erstellung der Download-Liste wird das Default-Package – falls vorhanden – immer am Ende der Liste angefügt. Andernfalls könnte ein Gerät zuerst das Default-Package laden, statt das mit Seriennummer adressierte Update-Paket.

Neben der Referenz auf die Update-Packet Definition (Package-ID) kann optional ein Gültigkeitszeitraum angegeben werden; Valid-From und Valid-To. Beim Erzeugen der Download-Liste werden nur die Einträge übernommen, die zum Zeitpunkt der Erstellung gültig sind. Andernfalls entfallen die Einträge. Mit dem Gültigkeitszeitraum kann eine zeitgesteuerte Update-Planung vorgenommen werden.

Damit nun ein Gerät tatsächlich eine Download-Liste von einer definierten URL laden kann, muss ein entsprechender Update-Target Eintrag erfolgen. Das Update-Target referenziert eine Update-Source Definition (entspricht der Download-Liste) und die Spalte Slug definiert den Inhalt der Pfad-Variable {target} der Update-Server URL. Ein Update-Target kann zudem per Is-Enabled aktiviert oder deaktiviert werden. Diese Option hilft bei der Planung von Software/Firmware Rollouts.

Information: Das Auto-Update-Center hält zwei URLs für den Zugriff durch das Gerät bereit:
GET - /autoupdate/v1/{tenant}/{target}/index.txt *(Content-Type: text/plain)*
GET - /autoupdate/v1/{tenant}/{target}/package/{package-object-name} *(Content-Type: application/tar)*

Information: Die Spalte Titel der Tabelle Update-Target kann beim automatischen Konfigurieren der Update-Server Liste im icom OS durch das Device-Management in das Feld „Beschreibung“ übernommen werden. Dieser Vorgang könnte zum besseren Verständnis beitragen, als auch bei der Fehlersuche behilflich sein.

Caching

Um der Last von vielen Geräte-Zugriffen auf Update-Pakete entgegen zu wirken, sollen alle Daten zur Erstellung des Indexes und der Update-Pakete „gecached“ werden.