Syncon Commloss – Zwischenspeicherung von Sensordaten bei fehlender Internetverbindung zur SensorCloud [DBAP7]

09.12.2013, Andreas Lockermann

Thomas Partsch

Alexander Stec

Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages

1. Problemstellung

Zur Wahrung eines konsistenten Datenbestandes in der SensorCloud werden Sensordaten bei Verlust der Konnektivität mit der Cloud-Datenbank in der Datenbank des Gateways (LocationMaster) gepuffert. Die Pufferung der Sensordaten in der Gateway-Datenbank stellt sicher, dass keine Sensordaten verloren gehen. (vgl. [1]) Bei wiederhergestellter Konnektivität werden die gepufferten Sensordaten in die Datenbank der SensorCloud übertragen.

Ist die Verbindung wiederhergestellt, soll das Paket S der in dem Zeitintervall $T=[t_1, t_2]$ entstandenen Sensordaten in die Cloud nachträglich übertragen werden.

Woher *weiß* die SensorCloud-Datenbank, welche von den Sensordaten in der Gateway-Datenbank bereits in der Cloud gespeichert sind und somit nicht noch einmal gespeichert werden müssen? Welche Voraussetzungen müssen die Sensordaten besitzen, damit keine Duplikate in die Cloud-Datenbank eingefügt werden? Synchronisation bedeutet in diesem Zusammenhang, die *einmalige* Speicherung der Sensordaten in der SensorCloud.

2. Ziele

Das Ziel des Arbeitspaketes *Syncon Commloss*, ist die Spezifikation und Implementierung eines Konzepts, dass die Zwischenspeicherung der Sensordaten auf der Gateway-Datenbank sicherstellt, bevor die Sensordaten in die Cloud-Datenbank übertragen werden. Die Zwischenspeicherung der Sensordaten in der Gateway-Datenbank stellt sicher, dass keine Sensordaten verloren gehen, wenn die Internetverbindung zur SensorCloud unterbrochen ist. (vgl. [1]).

3. Vorüberlegungen

Der Lösungsansatz wird am Beispiel der für Sensordaten relevanten Entität *Messwert* diskutiert. Dabei soll sich das Lösungskonzept auch für weitere Entitäten wie *Events* oder auch im Umgekehrten Fall, wie bei der *Aktoranforderungen*, die Steuerungskommandos von der Cloud Richtung Gateway erhält, umsetzen lassen

Die Entität *Messwert* auf dem LocationMaster hat aktuell den folgenden Aufbau:

`MesWerID` + `MesWerTimSta` + `MesWerSenID` + `MesWerNam` + `MesWerWer`.

`MesWerID` := Messwert-ID (UUID)

`MesWerTimSta` := Zeitstempel

`MesWerSenID` := Sensor-ID

`MesWerNam` := Name der gemessenen Entität

`MesWerWer` := Wert der gemessenen Entität

Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages

Der Datensatz für die Messwert Tabelle vom Multi-Raum-Sensor würde wie folgt aussehen:

MesWerID	MesWerTimSta	MesWerSenID	MesWerNam	MesWerWer
0002d966-ac87-46a9-9b3c-6d95751b22a8	1358941375252	1676566	Luftfeuchte	b7c

Es wird davon ausgegangen, dass der Zeitstempel für jeden Datensatz vorhanden und korrekt ist. Zudem wird angenommen, dass die Cloud-Datenbank in einem konsistenten Zustand ist.

4. Synchronisationsstrategien mit lokaler Replikation

Nachfolgend werden mehrere Synchronisierungsstrategien für die Übertragung der Sensordaten vom Gateway zur SensorCloud vorgestellt.

Die in diesem Kapitel vorgestellten Strategien gehen von einer gleichzeitigen Daten-Replikation in der lokalen Datenbank des LocationMasters und Daten-Pufferung vor der Übertragung in die Cloud-DB aus. Dieses Verfahren benötigt einen gewissen Prozess- und Ressourcen-Overhead, kann aber u.U. sinnvoll erscheinen.

Eine temporäre Replikation von Sensordaten, d.h. die redundante Speicherung der Sensordaten in mehreren Datenbanken (z.B. in der Gateway- und Cloud-Datenbank), würde nach Conrad in [5] den Zugriff auf lokale Sensordaten verbessern. Dies bedeutet für das Gateway, dass mögliche Analyseprogramme weiter auf der Messwert-Tabelle durchgeführt werden könnten.

Strategie A1 (ohne Berücksichtigung der MesWerID)

Theoretisch können mehrere Datensätze des gleichen Sensors (gleiche `MesWerSenID`) identische Werte im `MesWerWer` besitzen. Somit kann bei der Synchronisation das Kriterium `MesWerWer` für die Eindeutigkeit eines Datensatzes nicht berücksichtigt werden.

Mit der Annahme, dass der Zeitstempel (`MesWerTimSta`) bei jedem Datensatz eines bestimmten Sensors (`MesWerSenID`) verschieden ist, kann der Synchronisationsmechanismus mit Hilfe des Zeitstempels und der eindeutigen `MesWerSenID` eines Sensors entscheiden, ob ein Datensatz, von diesem Gateway, bereits in der Cloud-DB vorhanden ist. Die `MesWerSenID` des Sensors muss dabei berücksichtigt werden, da es theoretisch möglich ist, dass Datensätze von unterschiedlichen Sensoren mit gleichem Zeitstempel in der Datenbank des Gateways erfasst werden.

Ansatz für die Synchronisation:

Der Datensatz 1 mit Sensor-ID = 1 und Zeitstempel t_1 = 123456789 soll in die Cloud-Datenbank eingefügt werden:

a) Existiert bereits ein Datensatz mit identischem Zeitstempel t_1 mit gleicher Sensor-ID, wird der Eintrag als Duplikat interpretiert und nicht in die SensorCloud übertragen.

b) Existiert kein Datensatz mit diesem Zeitstempel, wird der Eintrag in die SensorCloud übertragen.

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Um eine größere Menge m_1 von Datensätzen auf Duplikate zu prüfen, kann als Kriterium der größte Zeitstempel t_x (mit Sensor-ID) aus der Menge m_1 gewählt und geprüft werden, ob in der SensorCloud-Datenbank Datensätze in der Menge m_2 zu diesem Sensor existieren, deren Zeitstempel t_y größer oder gleich als t_x ist:

a) Falls ein solcher Datensatz existiert, gilt m_1 als Duplikat und wird verworfen.

b) Wird kein Datensatz mit Zeitstempel t_y gefunden, der größer oder gleich einem Datensatz mit t_x ist, wird der Datensatz mit Zeitstempel t_x als noch nicht in der SensorCloud eingefügt markiert und der nächst kleinere Datensatz aus der Menge m_1 betrachtet und geprüft.

   Nach Prüfung aller Datensätze, werden die neuen Datensätze aus der Menge m_1 in die SensorCloud-Datenbank eingefügt.

Beispiel:

1) Zur Vereinfachung des Beispiels sind als Datensätze nur Zahlenwerte (:= Zeitstempel) in den Mengen enthalten und gehören zum gleichen Sensor.

   Menge m_1 = {1 , 2, 3, 4, 5} und Menge m_2 = {1, 2, 3, 4, 5, 6}

   → Es tritt Fall (a) ein und die Menge m_1 wird verworfen.

2) Menge m_1 = {4 , 5, 6, 7, 8} und Menge m_2 = {1, 2, 3, 4}

   → Es tritt so lange Fall (b) ein, bis „4“ erreicht wird. Nach der Synchronisation ergibt sich die Menge m_2 = {1, 2, 3, 4, 5, 6, 7, 8}

Der Lösungsansatz stellt sicher, dass keine Datensätze bei der Synchronisation verloren gehen, da jeder zu synchronisierende Wert aus dem Gateway im Fall (b) geprüft wird.

Als Erweiterung kann eine Hilfstabelle (`SyncHilfe`) mit folgendem Aufbau verwendet werden, so dass nicht alle zu synchronisierenden Daten durchlaufen werden:

SynHID	SynHEnt	SynHObj	SynHTimestp	SynHSum	SynHZ
f634d20e-51d7-4a8c-bd68-60089cae9432	Messwert	1151267	1342628583071	350	2

Erläuterung der Spalten:

- `SynHID` ist eine UUID und ist PRIK
- `SynHEnt` ist eine zu synchronisierende Entität, z.B. Messwert
- In `SynHObj` steht ein konkretes Objekt (z.B. `MesWerSenID`, FKEY) dessen Werte in Messwerte bzw. in der sonst zu synchronisierenden Entität gespeichert sind
- `SynHTimestp` ist der Zeitstempel bis zu dem die letzte Synchronisation dieses Objekts (`SynHObj`) in dieser Entität (`SynHEnt`) erfolgreich durchgeführt worden ist
- `SynHSum` enthält die Summe der Messwerte dieses Objekts in dieser Entität
- `SynHZ` gibt den Zustand der Synchronisation aus:
  - 0 := noch nie synchronisiert worden
  - 1 := wird gerade synchronisiert

Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages

- 2 := Synchronisation abgeschlossen (fester Zustand)

Beispiel:

Gegeben sind zwei Sensoren (`MesWerSenID`s sind 4710 und 4711) mit jeweils 3 Messwerten:

Menge m_1 = {{12345, 4710} , {12346, 4710}, {12347, 4710}, {12345, 4711}, {12346, 4711}, {12347, 4711}}

Die Tabelle `SyncHilfe` sieht dazu wie folgt aus:

SynHID	SynHEnt	SynHObj	SynHTimestp	SynHSum	SynHZ
f634d20e-51d7-4a8c-bd68-60089cae9432	Messwert	4710	12345	250	2
f634d20e-51d7-4a8c-bd68-60089cae9433	Messwert	4711	12346	300	2

Mit Hilfe der Tabelle `SyncHilfe` „weiß“ das Synchronisationsprogramm, dass alle neuen Messwerte aus der Menge m_1, die einen größeren Zeitstempel als 12345 besitzen und vom Sensor (`SynHObj`) 4710 sind, übertragen werden sollen: {12346, 4710}, {12347, 4710}.

Analog nach diesem Prinzip werden auch die Messwerte mit dem Zeitstempel größer 12346 und `MWSenID` 4711 übertragen: {12347, 4711}.

Bei Verlust der Tabelle `SyncHilfe` kann die Synchronisation nach diesem Verfahren nicht mehr fortgeführt werden, bis die Entität `SyncHilfe` wieder initialisiert wird.

Die vorgestellte Strategie zeigt, wie man die Synchronisation durchführen kann, wenn die Verbindung zur SensorCloud verloren gegangen oder unterbrochen wurde und man nicht weiß, welche Daten schon synchronisiert worden sind. In diesem Fall können die Messwerte als Mengen gruppiert nach der Sensor-ID aus der lokalen Datenbank heraussucht und dann gemäß des oben beschriebenen Verfahrens nach Zeitstempel sortiert verarbeitet werden.

Strategie A2 (mit Berücksichtigung der MesWerID)

Bei der folgenden Strategie wird die `MesWerID` als Kriterium für die Synchronisation aufgenommen. Die `MesWerID` wird auf dem LocationMaster erzeugt und mit übertragen. Die `MesWerID` ist eindeutig und besteht aus einer UUID (Universally Unique Identifier) [4], die zwar eindeutig ist, jedoch nicht fortlaufend wie eine gewöhnliche ID ist (1, 2, 3, …).

Um eine Sortierung und einen Vergleich auf Cloud-Seite durchführen zu können, benötigt man einen zweiten Faktor, den Zeitstempel. Somit kann als Lösungsansatz auch `MesWerID` und Zeitstempel angesehen werden.

Strategie A3

Als dritte Strategie kann mit Hilfe von Flags (Boolean-Werte: 0 oder 1) bestimmt werden, ob ein Wert schon synchronisiert wurde oder noch nicht. Bei dieser Strategie ist es notwendig, in jeder Tabelle dieses Flag zu setzten und zu pflegen.

Strategie B1

Die Strategie B1 nach [2] verzichtet auf das Synchronisieren. Stattdessen soll diese Strategie, die

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Übertragung der Sensordaten vom Gateway zur SensorCloud sicherstellen, und zwar so, dass die erfolgreich übertragenen Sensordaten in die SensorCloud-Datenbank auch wieder von der Gateway-Datenbank gelöscht werden. Diese Strategie setzt voraus, dass sowohl das Gateway als auch Cloud-Datenbank über Mechanismen der Transaktionssicherheit (Commit, Rollback, Abort) verfügen.

Werden z.B. 10 Datensätze vom Gateway in die SensorCloud übertragen, so kann man die 10 erfolgreich übertragenen Datensätze auch wieder vom Gateway entfernen, da sie ja sicher in der Cloud gespeichert sind. Das bedeutet auch, dass alle Sensordaten die sich in der lokalen Datenbank des Gateways befinden immer *„NEU“* sind, und somit in die Cloud übertragen werden müssen. Es muss demnach nicht mehr geprüft werden, welche Sensordaten vom Gateway noch nicht in die SensorCloud übertragen wurden.

Das nachfolgende Programmablaufdiagramm, soll die beschriebene Idee verdeutlichen:

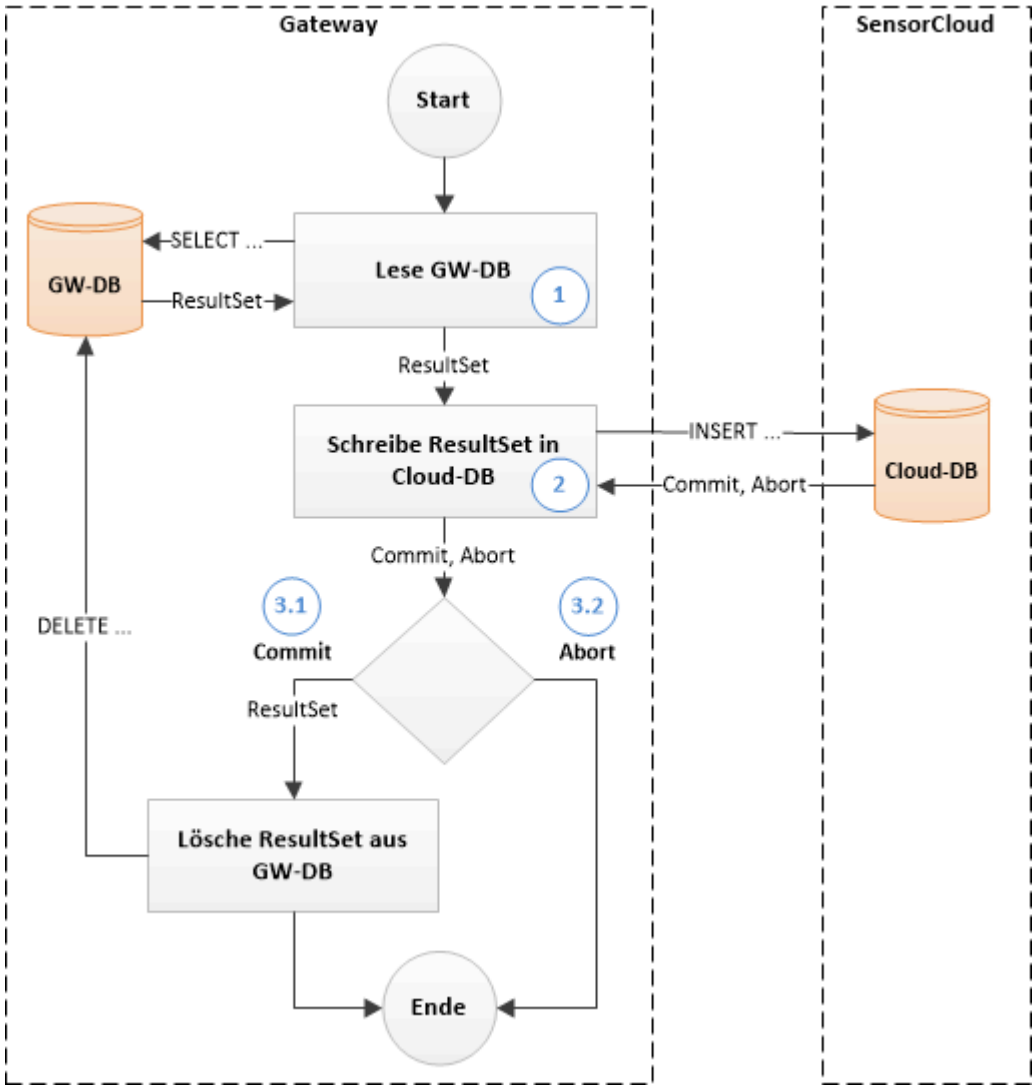

Abbildung 1: Programmablaufdiagramm - Strategie B1

Ablaufbeschreibung

1. Ein lesender Zugriff `SELECT * FROM Messwert` wird auf die Gateway-Datenbank durchgeführt. Als Ergebnis erhält man einen `ResultSet`, welches die in diesem Augenblick

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

gespeicherten Datensätze der Messwert-Tabelle der Gateway-Datenbank beinhaltet.

2. Die Datensätze des ResultSets werden in die Cloud-Datenbank geschrieben `INSERT INTO Messwert (...) VALUES (...)`.

3. Verzweigung, unterteilt in zwei mögliche Pfade:

   3.1. Hat das Schreiben in die Cloud-Datenbank fehlerfrei funktioniert, so können die übertragenen Datensätze wieder aus der Messwert-Tabelle der Gateway-Datenbank gelöscht werden. Da sich in der Zwischenzeit neue Datensätze in der Gateway-Datenbank befinden könnten, werden nur die Datensätze anhand der `MWID` des ResultSets aus der Gateway-Datenbank gelöscht `DELETE FROM Messwert WHERE MWID = …` . Die MWIDs, der transferierten Datensätze in die Cloud-Datenbank, werden in einer ArrayList festgehalten. Dadurch ist gewährleistet, dass nur die Datensätze aus der Gateway-Datenbank gelöscht werden, die man im 1. Schritt auch gelesen hat.

   Es ist durchaus vorstellbar, dass ein Fehler mitten beim Schreibvorgang in die Cloud-Datenbank auftritt. Beispielsweise, wurden 4 von 10 Datensätzen erfolgreich in die Cloud-Datenbank übertragen und ab dem 5ten Datensatz wurde ein Abort von der Cloud-Datenbank gemeldet. Ist dies der Fall, dann beinhaltet die ArrayList mindestens 4 MWID Einträge, der 4 erfolgreich übertragenen Datensätze in die Cloud-Datenbank. Somit kann die ArrayList abgearbeitet werden und dieselben 4 Datensätze wieder von der Gateway-Datenbank entfernen.

   3.2. Ist ein Fehler beim anfänglichen Schreiben der Datensätze in die Cloud-Datenbank aufgetreten (z.B. keine Verbindung zur Cloud-Datenbank vorhanden), so wird die Prozedur einfach beendet. Die Commit-Verzweigung (3.1) im Programmablaufdiagramm entspricht einem `try`-Block, wohingegen die Abort-Verzweigung (3.2) einem `catch`-Block entspricht.

Diese *„Anti“*- Synchronisationsstrategie wurde erfolgreich für die Dienste *Push2Cloud* und *Pull2Cloud* aus [3] implementiert und getestet. Diese Strategie sorgt dafür, dass die Sensordaten bei einem Internetausfall erfolgreich vom der Datenbank des Gateway, in die die Cloud-Datenbank der SensorCloud übertragen werden und anschließend wieder von der Gateway-Datenbank entfernt werden, ohne zusätzlichen Synchronisationsbedarf.

Diskussion der Strategien A1 und B1 gegeneinander

Die Strategien A1 und B1 sollen nach Vor- und Nachteile gegenübergestellt werden.

Strategie A1:

- **Vorteile**

  (1) Garantiert, dass theoretisch keine Datensätze bei der Übertragung verloren gehen.

- **Nachteile**

  (1) Hoher Synchronisationsbedarf – jeder neuer Datensatz auf der Gateway DB muss vor der Übertragung auf bereits vorhandene Existenz in der Cloud DB geprüft werden.

  (2) Hoher Kommunikations- und Datenverkehr zwischen Gateway und Cloud – bedingt durch (1).

  (3) Zusätzlicher Rechenleistungsbedarf – übernimmt das Gateway die

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Synchronisationskontrolle, so kann die Rechenleistung eines Einplatinenrechners an ihre Grenzen stoßen.

Strategie B1:

- **Vorteile**

  (1) Kein Synchronisationsbedarf - Kein Synchronisierungsaufwand nötig, da nichts synchronisiert wird.

  (2) Geringerer Kommunikations- und Datenverkehr - beschränkt sich nur auf die zu übertragenden Sensordaten.

  (3) Kein zusätzlicher Rechenleistungsbedarf – das Gateway muss keine Synchronisationskontrolle betreiben.

- **Nachteile**

  (1) Benötigt Transaktion-/Übertragungssicherheit – es muss sichergestellt werden, dass die übertragenen Sensordaten auch zuverlässig in der Cloud DB gespeichert werden. Erst dann können die übertragenen Sensordaten wieder von der Gateway DB entfernt werden.

5. Synchronisationsstrategien ohne lokale Replikation

Bei einer kontinuierlichen Übertragung von Daten in die SensorCloud ohne gleichzeitige Replikation ist es denkbar einen ressourcensparsameren Weg zu gehen, indem man die Daten direkt überträgt und nur bei einem Kommunikationsausfall in der lokalen Datenbank puffert.

Abbildung 2 beschreibt eine angedachte Kommunikation zwischen den Prozessen auf dem Gateway und der Kommunikation mit der Cloud-DB. Messwerte werden von den Treibern an die Gateway-Software weitergereicht. Die Gateway-Software erkennt, ob eine Internetverbindung verfügbar ist. Um den Datenbestand auf dem Gateway gering zu halten, versendet die Gateway-Software bei bestehender Internetverbindung die Messwerte direkt über den Nachrichtendienst und der anschließenden Programmkette in die Cloud-DB.

Die Cloud versendet Rückmeldungen über den Erfolg oder Nichterfolg der Verarbeitung der Messwerte innerhalb der Cloud. Die Rückmeldung erfolgt anhand der `MesWerID`. Die `MesWerID` besteht aus einer UUID (Universally Unique Identifier) [2] und sichert eine Eindeutigkeit über das gesamte Föderierte Datenbanksystem. Die `MesWerID` ist somit eine eindeutige Referenz im gesamten System. Die sendenden Programme (Gateway- und Syncon Comloss-Software) erhalten die Rückmeldungen ihrer versandten Messwerte und reagieren auf diese.

Auftretende Fehler können hierbei Fehlermeldungen aus der Cloud sowie das Nichterhalten einer Rückmeldung aus der Cloud sein. Bei Fehlern, die einen erneuten Versand nötig machen, werden die Messwerte in der lokalen Datenbank zwischengespeichert. Der erneute Versand erfolgt über die Syncon Comloss-Software, die neben der eventgesteuerten Ausführung noch periodisch ausgeführt wird.

Messwerte mit einer erfolgreichen Rückmeldung werden von dem ursprünglich sendenden Programm aus dem flüchtigen Puffer der Gateway-Software, bzw. der lokalen Datenbank gelöscht.

Bei einem Konnektivitätsverlustes puffert die Gateway-Software die Messwerte direkt in der lokalen Datenbank persistent. Das Syncon Commloss-Programm erkennt die Wiederaufnahme der

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Internetverbindung und sendet die zwischenzeitlich in der Datenbank gepufferten Messwerte über den Nachrichtendienst und der anschließenden Programmkette in die Cloud.

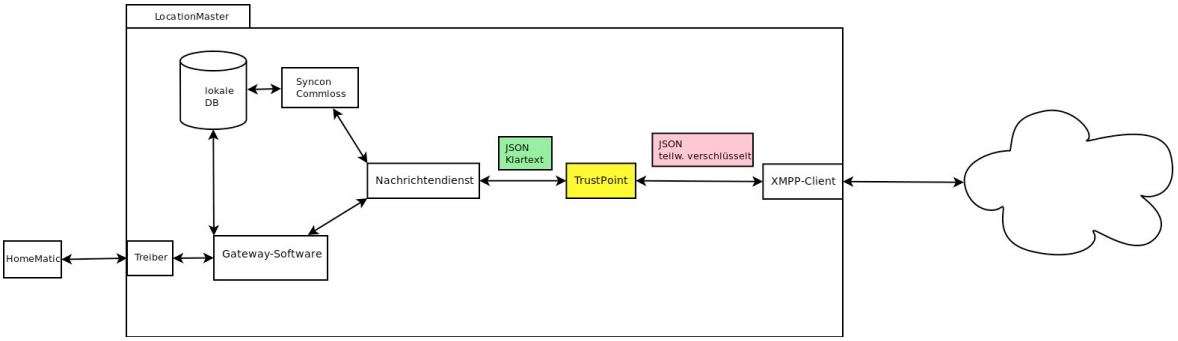

Abbildung 2: Übersicht Prozesskommunikation auf Gateway (Daniel Scholz, 2013)

Aktuell werden zur Synchronisation die in [3] getesteten und implementierten Dienste Push2Cloud und Pull2Cloud verwendet. Diese können bei den folgenden Überlegungen nach Umstellung auf XMPP und SensorCloud-Protokoll weiterverwendet werden. Gegenwärtig verwenden Push2Cloud und Pull2Cloud klassische Datenbank-Transaktionen die gegen eine Kommunikation über das SensorCloud-Protokoll ausgetauscht werden müssen.

Die aktuell eingesetzten Programme Push2Cloud und Pull2Cloud lassen sich an das beschriebene Szenario anpassen, sofern aus der Cloud eine Rückmeldung über den Erfolg der Verarbeitung der Messwerte gegeben wird. Dazu steht noch eine Entscheidung der Protokoll-Arbeitsgruppe bzgl. Return-Codes aus der Cloud aus.

Zudem kann das oben beschriebene Szenario um eine Verdichtung der in der lokalen Datenbank gepufferten Messwerte erweitert werden, um ein Volllaufen des Gateways bei längerem Kommunikationsausfalles zu vermeiden. Hierzu muss das konzeptionelle Schema um die Verdichtungseigenschaft von Messwerttypen erweitert werden.

Eine Verdichtung von Messwerten kann bei hochfrequenter Messwerterzeugung Übertragungskapazität sparen. Die Verdichtung kann in der Datenbank durch ein weiteres periodisch laufendes Programm vorgenommen werden. Durch die periodische Ausführung der Syncon Comloss-Software können die zwischenzeitlich in der lokalen Datenbank gespeicherten und verdichteten Messwerte auch bei einer bestehenden Internetverbindung versendet werden.

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Quellen

[1] FH Köln, *SensorCloud Teilvorhabenbeschreibung*, 2011

[2] A. Stec: *Transaktionen in einem föderierten Datenbanksystem der SensorCloud*, Master-Thesis, FH-Köln, 2013

[3] A. Stec: *Übertragung von Sensordaten unter Berücksichtigung von Transaktionskonzepten in der SensorCloud, Ergebnisse [DBAP6]*, Bericht für Meilenstein 3, FH-Köln, Gruppe FDBS, 2013

[4] Universal Unique Identifier (UUID), http://de.wikipedia.org/wiki/Universally_Unique_Identifier, (Aufruf: 16.03.2013)

[5] S. Conrad: Föderierte Datenbanksysteme; Konzepte der Datenintegration, Springer-Verlag, 1997

Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages