ATB
Leibniz-Institut für
Agrartechnik und Bioökonomie

Konzeptplan

SQL basierende Datenlogger-Testumgebung zur vollständigen Automatisierung des Datenmanagements am Beispiel der Gaskonzentrationsmessungen am Standort Dummerstorf.

Alexander Prinz

Dieser Konzeptplan ist eine erste grobe Präsentation eines alternativen voll automatisierten Datenmanagementsystems zur Verwaltung der anfallenden Messdaten am Beispiel der Gaskonzentrationsdaten am Standort Dummerstorf. In der ersten Testphase, wird das bestehende System über 2 Einplatinencomputer konkret dem Raspberry Pi 3 emuliert/simuliert.
Dazu wird eine Testumgebung via beider Raspberry Pi entwickelt, wobei eines den aktuellen Messgeräte-Server emuliert und das 2. Raspberry Pi eine SQL-Datenbank in Form eines Servers simuliert. In der zweiten Testphase -so sie denn passieren soll- wird das Testsystem in das reale System in der Form integriert, dass der herkömmliche Ablauf unangetastet weiter läuft. Die Realdaten werden jedoch zusätzlich in eine SQL-Datenbank transferiert. Ein Datenverlust durch Systemversagen aufgrund von anfänglichen Bugs etc. soll somit vermieden werden.
Die benötigte Hardware stelle ich. Auf etwaige Attribute wissenschaftlicher Arbeiten wie die Zitation, Quellenangabe etc. wird in diesem Konzeptplan verzichtet.

Problemstellung

Am Standort Dummerstorf werden seit Jahren Gaskonzentrationsmessungen in einem Milchviehstall zu Monitoring-Zwecken getätigt. Dabei wird an 12 Messstellen innerhalb sowie außerhalb des Stalls, über ein Schlauchsystem das Luft-Gas-Gemisch angesaugt und zu einem Spektrometer geleitet. In dem Spektrometer werden die Gasspektren spezifischer Stoffe ermittelt und als Datei geloggt. Über ein Programm, können die Spektraldaten in Gaskonzentrationsdaten transformiert und als ASCII-Textdateien gespeichert werden. Je nach definierten Sample-Intervall fallen integriert über den Messzeitraum entsprechend große Datenmengen an.
Die Menge der anfallenden Daten, sowie die Art der Formatierung zwingt zu unnötigen Umgang mit den Daten, sowie hohen Speicherbedarf. Dieses Konzept soll ein erster alternativer Ansatz zum konventionellen System sein und aktuelle Datenbankenmanagementsysteme als Optimierung hinsichtlich verschiedener Aspekte wie Datenspeichergröße, Datenzugriffsvereinfachung etc. etablieren.

Motivation

Die Digitalisierung macht vor keiner Branche halt und scheint Fluch und Segen zugleich zu sein. Fluch, weil sie uns zwingt, oft neue, ungewohnte Pfade zu betreten, welche nicht selten neue Kompetenzen, Risiken und Kosten erfordern. Segen, weil sie -vorausgesetzt, jene neue Kompetenzen sind vorhanden und die Risiken und Kosten stehen im guten Verhältnis zum Profit- unser Dasein wesentlich vereinfachen kann. Um als Unternehmen, gleich welcher Art, also auch und vor allem als modernes wissenschaftliches Forschungsinstitut, wettbewerbsfähig zu bleiben, ist es vielmehr also schon Zwang und Not der Digitalisierung beizuwohnen.
Die bei der Gaskonzentrationsmessung am Standort Dummerstorf anfallenden Datenmengen sind aufgrund ihrer Gestallt als ASCII-Textdateien unnötig groß, als auch wenig gut zu handhaben. Der ASCII-Standard ist eine 7 Bit Dateiformatierung für Textdateien, welche bereits ca. 50 Jahre alt ist. Es zeichnet sich aus moderner informatischer Sicht negativ durch ein unnötig hohen Speicherverbrauch aus. Zudem, sind die Ausgabetextdateien des Messgerätes erheblich schlecht arrangiert/formatiert (siehe dazu Anhang; Abb.: I) Daraus ergibt sich, dass über den gesamten Messzeitraum enorm große Datenmengen anfallen, welche in einem speziellen, nicht universellen Format vorliegen und zur weiteren Verarbeitung aufbereitet werden müssen. Die Aufbereitung kostet Zeit, vor allem auch deshalb, weil an entsprechenden Stellen nicht selten durch unvorteilhaftes, erschwertes Arbeiten mit Tabellenkalkulationsprogrammen agiert wird. Des Weiteren erfordern die hohen Datenmengen unnötig viel Speicherplatz. Speicherplatz kostet Geld, und die Tatsache, dass vor allem moderne Speichermedien immer geringere Lebenserwartungen haben und entsprechende Wartungsmaßnahmen bzw. Ersetzung ihrer verlangen, sollte eine signifikante wirtschaftliche Relevanz dieses Konzeptes (als Anstoß für weitere) rechtfertigen. Es sollte also schon aus ökonomischer Sicht interessant sein, dass die anfallenden Datenmengen so klein wie nur möglich gehalten werden sollten, und durch universelle Formate und Datenbankenmanagementsysteme unkompliziert und schnell jedem Nutzer zugänglich gemacht werden.

Methodik

Das Konzept ist ein Komplex aus verschiedenen Instanzen, welche insgesamt ein Datenmangementsystem bilden. Diese Instanzen sind: ein open Source Datenbankenmangementsystem, welches auf SQL basiert, ein Server, auf dem das SQL-System läuft (im Folgenden beide zusammen SQL-Server genannt), eine individuell programmierte Software welche die Ausgabedateien des Messgeräts in einem festgelegten Intervall verarbeitet und in die Datenbank überführt, eine Software, welche die zeitliche Ausführung der eben genannten Software steuert, sowie der Messgeräte-Server. An diesem wird der SQL-Server eingebunden.
Um möglichen Datenverlust durch anfängliche Systemfehler auszuschließen, wird das System in der zweiten Testphase gekapselt parallel zum bestehenden System als untergeordneter Server laufen. Der SQL-Server erhält nur Leserechte zum Messgeräte-Server, kann also nur auf die Daten zugreifen um sie auf sich zu kopieren ohne sie jedoch auf dem Messgeräte-Server zu manipulieren. Somit ist gesichert, dass der herkömmliche Ablauf des Datenzugriffs bzw. der Datenaufzeichnung weiterhin besteht und in keiner Weise verändert wird.
Im folgenden Verlauf werden die einzelnen Instanzen erläutert und die einzusetzende Peripherie hinsichtlich auch ihrer möglichen Vor- und Nachteile vorgestellt.

Datenbakenmangementsystem

Dem Datenmangement soll ein modernes SQL basierendes System zu Grunde liegen.
SQL-basierende Datenbankenmangementsysteme (im folgenden nur noch DBMS) bilden als relationale DBMS neben den nichtrelationalen DBMS vermutlich aktuell die Basis nahezu aller digitaler Datenbanken egal ob web-basiert oder nicht. Der Vorteil ist, dass sie unkompliziert in Serverumgebungen integriert werden können und ein hohes Maß an flexibler Funktionalität bieten. Sie lassen sich in diverse Anwendungen durch einfache Skriptsprachen implementieren und skalieren. Somit ist es möglich, bestehende Datenbanken zu jeder Zeit durch neue Daten zu ergänzen und die Daten auszulesen um sie entsprechend verarbeiten zu können.
SQL-basierende DBMS existieren in verschiedenen Formen und von verschieden Anbietern/Entwicklern. Es gibt sowohl kommerziell erhältliche DBMS wie z.B. MS Access, ORACLE, IBM DB2 usw. als auch nichtkommerzielle oft open-source verfügbare DBMS z.B. MYSQL, POSTGRE SQL usw. Die kommerziellen Systeme bieten den Vorteil, dass die Systemstabilität und Integration durch die Entwicklerunternehmen gewährleitet bzw. abgesichert werden und oft eine grafische Nutzeroberfläche existiert. Die Nachteile sind vor allem hohe Kosten der Lizenzen sowie weniger flexible Einbindungsmöglichkeiten in entsprechend individuelle Software-Entwicklungsprojekte.
Die Vorteile der open-source Systeme sind natürlich die Nachteile der kommerziellen Systeme. Sie lassen sich flexibler in spezielle Softwareprojekte einbinden und sind kostenfrei. Zudem steht, wie hinter den meisten anderen open Source Projekten auch, ein großer Nutzer/Entwicklerzusammenschluss, welcher in einschlägigen Foren Hilfe bei Problemen bietet (Kostenfrei). Die Nachteile beschränken sich meist auf die (relativ unwahrscheinliche) Möglichkeit, dass die Entwickler irgendwann den kostenfreien Support bzw. die kostenlose Nutzung einschränken bzw. einstellen, wodurch bestehende Systeme entsprechend angepasst werden müssten.
Das gängigste open Source SQL System ist derzeit MYSQL, welches aufgrund der hohen Anwenderzahl und dem dadurch guten Support für diesen Zweck als optimal gelten kann. Deshalb sieht dieses Konzept auch die Nutzung von MYSQL vor. Des Weiteren gibt es viele etablierte Bibliotheken bzw. Einbindungsmöglichkeiten in Python- oder PHP-Programmcode.

SQL-Server- und Messgeräteserveremulator-Hardware

Als Hardware zur Realisierung der Server soll jeweils ein RASPBERRY PI 3 dienen. Es handelt sich dabei um einen sogenannten Einplatinen-Computer, welcher die nötigen Hardware-Parameter (Arbeitsspeicher und leistungsstarke CPU, nötige Anschlüsse) zu genüge für die Tests integriert. Die Vorteile sind die geringen Beschaffungskosten (vor allem gegenüber einem „großen“ Rechner), die niedrigen Energiekosten und Abmaße, sowie das flexible Konfigurieren an den geplanten Nutzen. Das Raspberry Pi hat sich über seine nun schon vielen Jahre des Bestehens einen großen Namen gemacht, welcher über seinen anfänglichen Status als „Bastelercomputer“ nun weit hinaus geht. Nachteile können aus meiner Sicht keine genannt werden.
Als Speichermedium dient ein handelsüblicher USB-Flashspeicher in guter Qualität, da die SD-Karten keine ausreichend gute Stabilität hinsichtlich hoher schreib-/Leseaufgaben bieten bzw. durch diese sehr schnell kaputt geht.

Zusatz: Zwar wird in der ersten Phase das System durch beide Raspberry emuliert. Jedoch kann zumindest der Messgeräte-Server durch ein Raspberry Pi ersetzt werden, sofern die entsprechende Software auch auf Linux Systemen läuft. Der SQL-Server muss im Realzweck natürlich auf einer hochgradig leistungsstarken Hardware aufgebaut werden, sofern er dann generell für alle weiteren Messgeräteaufzeichnungen anderer Standorte dienen soll.

SQL-Server-Betriebssystem

Das RASPBERRY PI 3 legt das Nutzen eines Linux basierenden Betriebssystem fest und hat durch RASPBIAN als eine modifizierte Variante von DEBIAN ein eigenes, auf seine Hardware optimiertes Betriebssystem erhalten, welches dem Zweck absolut hinreichend dienen sollte. Die Tatsache, dass es sich um ein Linux Betriebssystem handelt sollte als reiner Vorteil gesehen werden. Es ist -entgegen üblicher MS-Server-Betriebssysteme- kostenlos und sehr einfach zu handhaben, sowie sehr stabil. DEBIAN resp. RASPBIAN ist eine Linux-Distribution, welche eigens für den Betrieb von Server-Umgebungen entwickelt und optimiert wurde.

Konvertierungs- und Überführungssoftware

Die im Anhang gezeigte Abbildung I zeigt einen Ausschnitt aus einer der ASCII-Textdateien zu den Gaskonzentrationsmessungen. Zu sehen ist, dass nach der Kopfzeile alle Messparameter und ihre Einheiten aufgeführt sind. So zum Beispiel für den Fall H_2O die Konzentration in Spalte 6 und die Einheit in Spalte 7. Auch zu allen weiteren 14 Messparametern werden in den entsprechenden Spalten die Einheiten aufgeführt. Da sich diese aber über den gesamten Messzeitraum nicht ändern, sind diese Angaben sowohl redundant als auch sehr speicherintensiv. Auch die Pfadangaben in Spalte 4 und 5 über alle Zeilen sind -da konsistent gleich- überflüssig und bei ihrer Zeichenlänge sogar erheblich speicherintensiv. Eine Optimierung ist an erster Stelle die vorhandenen Metadaten wie Einheiten und Pfadangaben nur einmal zu speichern. Die Reduktion um diese Metadateien beträgt dann bereits 43 %. Die Software ist daher so angelegt, dass sie die Metadateien aufgreift und separat in einer Log-Datei speichert.
Die Software agiert sowohl als Extraktor hinsichtlich der Daten, als auch als Schnittstelle zwischen der Datenausgabe des Messgerätes und der SQL-Datenbank. Sie ist in der Lage auf die Datenbank zuzugreifen und neu anfallende Daten in die Datenbank zu speichern. Da die Sample-Frequenz des Messgerätes je nach dem wie sie eingestellt wurde, im sekündlichen

bzw. menütlichen Bereich liegt, macht es aus informatischer Sicht hinsichtlich der Hardware-Resourcen (Arbeitsspeicher-Prozessor-Festplattenaktivität) keinen Sinn, das Programm für jedes neu entstandene Sample auszuführen. Deshalb wird ein Service-Intervall von z.B. 24 h festgelegt, in dem es die angefallenen ASCII-Daten aus dem Speicher des Messgeräte-Servers ausließt, konvertiert und in die Datenbank überführt. Die Software wird in der Programmiersprachen Python geschrieben und soll einem guten wartungsbaren Code entsprächen.
Das Programm wird mit diversen Sicherheits-Features ausgestattet, etwa einer Log-Datei-Ausgabe mit Fehlerprotokollierung bei entsprechenden möglichen Fehlern wie Laufzeitfehler und anschließende Warnung via Email an den Administrator etc. Weitere nützliche Features werden noch mit den technischen Mitarbeitern entwickelt.

Cronjob-Software

Wie bereits im vorhergehenden Punkt beschrieben, ist es unnötig und aus informatischer Sicht heikel, anfallende Daten durch das Messgerät pro Sample-Zeitpunkt in die Datenbank zu überführen. Deshalb soll ein Service-Intervall von 24 h geplant werden, indem die Konvertierungs- und Überführungssoftware angestoßen wird. Die Software „Vivie Cron“ ist eine für Linux-Betriebssysteme entwickelte, frei erhältliche Software, welche für diese Zwecke entwickelt wurde. Sie dient dazu, ausführbare Dateien zu einem geplanten Zeitpunkt (auch Periodisch) auszuführen. Zwar wäre es auch möglich, Die Konvertierungs- und Überführungssoftware derart zu entwickeln, dass sie permanent läuft und im täglichen Intervall die Datenkonvertierung und Überführung realisiert, jedoch halte ich es zum gegenwärtigen Zeitpunkt für optimaler, eine Cronjob Software als Sicherheitslösung zu nutzen, falls das Datenkonvertierungs- und Überführungsprogramm versagt. In dem Fall wird der Administrator gewarnt und die Anfallenden Daten weiterhin gespeichert. Datenverlust ist also auszuschließen.

Schema der SQL-Datenbank

Relationale Datenbanken funktionieren nach einem Zuordnungsprinzip. Einem Indexobjekt etwa die Messgeräte ID’s werden verschiedene Daten zugeordnet. Also etwa dem Prinzip einer Tabelle, welche in der Ersten Spalte die ID’s stehen haben könnte und in den weiteren Spalten stehen die zugehörigen Daten. Somit sind über die ID’s als Schlüssel sowie die Schlüsselwörter Parametername relevante Datenbankenbereiche auswählbar, um ihnen Daten zuzuführen, wie eben auch zur Nutzung zu entnehmen.
Die 12 Messplätze sind von Grund auf einem Messplatzindex zugeordnet, welcher als Schlüssel ID fungieren wird. Die relationalen Parameter sind nun jeweils: Messdatum, Samplezeitpunkt, die jeweiligen Gaskonzentrationswerte zu den einzelnen Stoffen als jeweils eigenständige Objekte, sowie der Messgerätestatus zum Samplezeitpunkt. Einige der relationalen Parameter lassen sich nun wieder aufschlüsseln, um so Speicher zu sparen. So ist es etwa bzgl. des Datums nicht nötig, für jedes Sample gleichen Datums pro ID das Datum in eine Zeile zu schreiben, sondern stattdessen eine untergeordnete Tabelle zu erzeugen, welche nur jedes vergangene Datum und einem Zuordnungsindex enthält. Dieser Zuordnungsindex wird dann anstatt des ganzen Datums in die Haupttabelle in das jeweilige Fach eingetragen.

Wenn ein(e) Wissenschaftler(in) nun mit den Daten arbeiten möchte, und sich dabei eventuell nur für bestimmte Parameter interessiert, vielleicht auch nur für einen bestimmten Messzeitraum, so kann dies individuell durch eine entsprechende Datenbankabfrage durch einen SQL-Code geschehen. Sofern mit passenden Programmiersprachen wie Python oder anderen gearbeitet wird, kann durch den Code direkt auf die Datenbank zugegriffen werden. D.h. die Daten müssen nicht auf eine Festplatte gespeichert werden, sondern werden lediglich in den Arbeitsspeicher des Systems, von welchem aus gearbeitet wird, geladen, wo sie dann wie gewohnt verarbeitet werden. Dies setzt jedoch den Umgang mit SQL-Datenbankenabfragen voraus. Bei SQL handelt es sich jedoch um einfache, wenig umfangreiche Befehle, welche schnell durch jeden erlernbar sein sollten.
Die Abbildung II (siehe Anhang) veranschaulicht eine mögliche Realisierung des gesammten Workflows. Die Kernkomponenten sind der Messgeräteserver und der SQL-Server, sowie ein Administrator, welcher einen umfangreichen Zugriff sowie die nötigen Kompetenzen im Umgang mit den beiden Servern hat. Der Administrator wartet die Server und pflegt die nötige Software. Er steht bereit, wenn Wissenschaftler Daten benötigen und kann gegebenen Falls gewünschte Datenanalysen direkt auf dem SQL-Server durch entsprechende Programme tätigen. Andernfalls gewährt man den Wissenschaftlern (Netzwerk-Clients) eingeschränkte Zugriffrechte auf den SQL-Server (lediglich Leserechte) um eigenständig gewünschten Datenzugriff/Verarbeitung ausüben zu können.
Im Idealfall, ist also der SQL-Server mit entsprechend hohen Hardwareparametern ausgestatten, sodass nicht nur die SQL-Datenbank als solches, sondern auch die Datenverarbeitung auf ihn passieren kann. Dies würde entsprechenden Datentransfer verhindern, birgt jedoch das Risiko, durch unsachgemäßen Gebrauch, etwa schlechter Programm-Code der Datenanalyse etc. Systemabstürze zu iduzieren. Abhilfe könnten hier virtuelle Systemumgebungen bieten. D.h. Auf dem Host-System des SQL-Servers läuft die SQL-Datenbank. Auf virtuellen Umgebungen können dann über FTP-Verbindungen die entsprechenden Datenverarbeitungen geschehen. Sollten diese zu Systemabstürzen führen, würde lediglich die virtuelle Systemumgebung ausfallen. Das Host-System bleibt davon unberührt.

Anhang

	Messstelle	Datum	Zeit	Spektrum	Anwendung	H2O Einheit	Kompensation	Rest	CO2 Einheit	Kompensation	Rest	N2O Einheit	Kompensation	Rest	NH
2	10	2014-05-01	00:00:21	C:\CalcmetSamples\20140501\GAS_48192.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.77	vol-%	feucht	0.0114	582	ppm	feucht			
3	1	2014-05-01	00:00:51	C:\CalcmetSamples\20140501\GAS_48193.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.75	vol-%	feucht	0.0115	500	ppm	feucht			
4	4	2014-05-01	00:01:21	C:\CalcmetSamples\20140501\GAS_48194.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.72	vol-%	feucht	0.0110	465	ppm	feucht			
5	7	2014-05-01	00:01:51	C:\CalcmetSamples\20140501\GAS_48195.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.77	vol-%	feucht	0.0114	500	ppm	feucht			
6	12	2014-05-01	00:02:21	C:\CalcmetSamples\20140501\GAS_48196.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.78	vol-%	feucht	0.0118	563	ppm	feucht			
7	2	2014-05-01	00:02:51	C:\CalcmetSamples\20140501\GAS_48197.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.78	vol-%	feucht	0.0116	590	ppm	feucht			
8	3	2014-05-01	00:03:21	C:\CalcmetSamples\20140501\GAS_48198.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.81	vol-%	feucht	0.0145	733	ppm	feucht			
9	5	2014-05-01	00:03:51	C:\CalcmetSamples\20140501\GAS_48199.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.81	vol-%	feucht	0.0116	672	ppm	feucht			
10	6	2014-05-01	00:04:21	C:\CalcmetSamples\20140501\GAS_48200.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.82	vol-%	feucht	0.0116	665	ppm	feucht			
11	9	2014-05-01	00:04:51	C:\CalcmetSamples\20140501\GAS_48201.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.81	vol-%	feucht	0.0148	729	ppm	feucht			
12	11	2014-05-01	00:05:21	C:\CalcmetSamples\20140501\GAS_48202.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.81	vol-%	feucht	0.0117	652	ppm	feucht			
13	8	2014-05-01	00:05:51	C:\CalcmetSamples\20140501\GAS_48203.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.80	vol-%	feucht	0.0109	670	ppm	feucht			
14	10	2014-05-01	00:06:21	C:\CalcmetSamples\20140501\GAS_48204.SPE	Luft_50C_Stall_20151210_SN122359.LIB	0.79	vol-%	feucht	0.0114	645	ppm	feucht			

Abbildung I:
Ausschnitt aus einer Messwertedatei der Gaskonzentrationsmessungen. Sie zeigt, dass ein erheblich hoher Anteil an Daten, z.B. die Einheiten des jeweiligen Parameters aufgeführt in jeder Zeile, Pfadangaben etc., welche konstant über den gesamten Zeitraum sind, unnötiger Weise in jeder Zeile vorkommen.

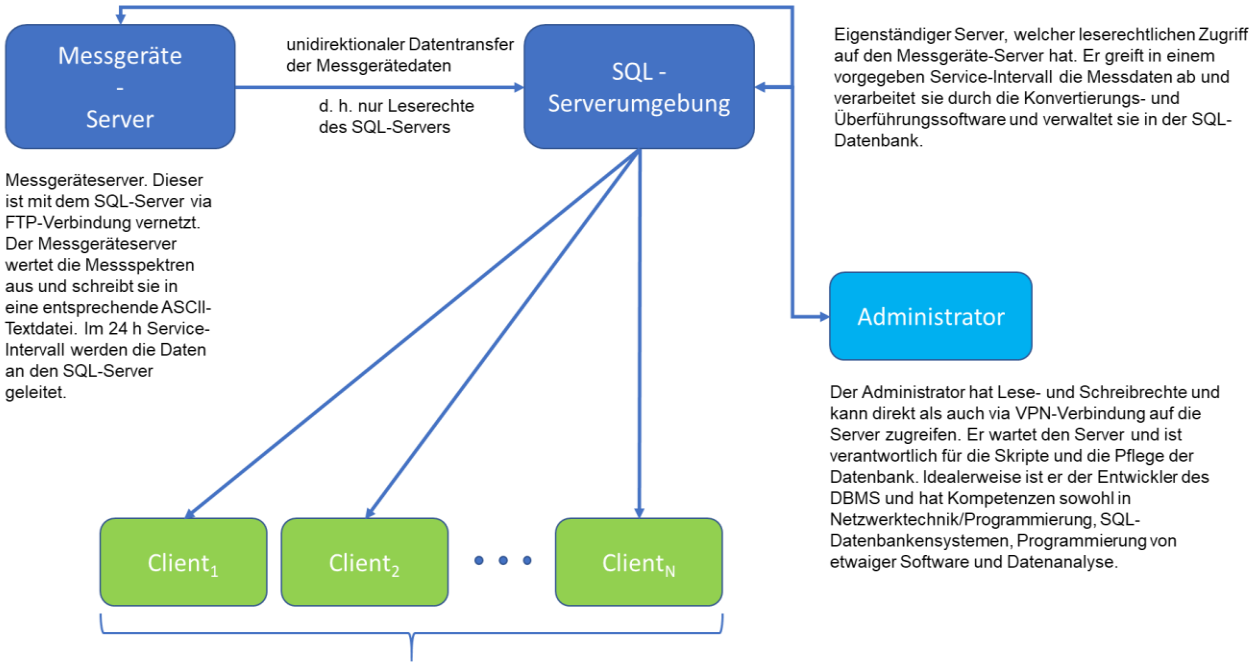

Abbildung II:
Schema des Workflows bzw. des Konzepts.
Die Hauptkomponenten bilden der Messgeräteserver sowie der SQL-Server. Der Administrator hat volle Zugriffsrechte auf beide Server und wartet diese. Die Clients sind die Datenbankenanwender, welche eingeschränkten Zugriff auf den SQL-Server haben. Sie können lediglich die Daten aus der Datenbank abgreifen.
Eine weitere Möglichkeit sieht eine erweiterte Zugriffsmöglichkeit der Clients auf den SQL-Server in der Art vor, dass auf einem virtuellen Betriebssystem auf dem SQL-Server direkt bzgl. der Datenverarbeitung von jedem Client gearbeitet werden kann. Somit sind Systemabstürze unkritisch (siehe dazu „Schema der SQL-Datenbank“ auf Seite 5-6).