



Lehr- und Forschungszentrum  
**Landwirtschaft**  
[www.raumberg-gumpenstein.at](http://www.raumberg-gumpenstein.at)



Schweizerische Eidgenossenschaft  
Confédération suisse  
Confederazione Svizzera  
Confederaziun svizra

Eidgenössisches  
Volkswirtschaftsdepartement EVD  
**Forschungsanstalt**  
Agroscope Reckenholz-Tänikon ART

## **2. Zwischenbericht zum Forschungsprojekt 100800 Einzelbetriebliche Ökobilanzierung landwirtschaftlicher Betriebe in Österreich (FarmLife)**

### **Betriebsdatenerfassung und Vorprüfung für die Ökobilanzierungsschnittstelle SALCA mit FarmLife-Capture (CAP) und FarmLife-Control (CON)**

**Version 1.0**

Mag. Thomas Guggenberger MSc.  
Abteilung für Ökonomie & Ressourcenmanagement  
LFZ Raumberg-Gumpenstein  
Email: [thomas.guggenberger@raumberg-gumpenstein.at](mailto:thomas.guggenberger@raumberg-gumpenstein.at)

30. September 2013

# Inhaltsverzeichnis

<b>1</b>	<b>ORIENTIERUNG.....</b>	<b>3</b>
<b>2</b>	<b>V-MODELL SOFTWAREENGINEERING CAP.....</b>	<b>3</b>
2.1	RAHMEN .....	4
2.1.1	<i>Modularität.....</i>	4
2.1.2	<i>Kompetenzen .....</i>	5
2.2	SYSTEMANFORDERUNG .....	6
2.2.1	<i>Funktionale Anforderungen.....</i>	6
2.2.1.1	Stammdaten.....	6
2.2.1.2	Geodaten der Feldstücke und Schläge.....	6
2.2.1.3	Erfassung von Ein- und Verkauf.....	7
2.2.1.4	Erfassung der Feldarbeit.....	8
2.2.1.5	Erfassung von pflanzenbaulichen Erträgen .....	9
2.2.1.6	Erfassung des Inventarbestandes.....	9
2.2.1.7	Variable Mengenerfassung.....	9
2.2.1.8	Listenelemente .....	10
2.2.1.9	Status der Eingabe/Mitteilungen .....	10
2.2.1.10	Prüfung .....	10
2.2.2	<i>Nicht-funktionelle Anforderungen.....</i>	10
2.2.2.1	Design .....	10
2.2.2.2	Zugang .....	10
2.2.2.3	Datensicherheit/Datenschutz.....	10
2.2.2.4	Schnittstellen.....	11
2.3	SYSTEMARCHITEKTUR .....	11
2.3.1	<i>Technischer Aufbau des Systems .....</i>	11
2.3.1.1	Datenbank-CAP .....	11
2.3.1.2	Server-CAP.....	11
2.3.1.3	Master-Control-Station .....	11
2.3.1.4	Entwicklungsumgebung-CAP.....	11
2.3.1.5	Gesamtsystem .....	11
2.3.2	<i>Identifikation von Schnittstellen .....</i>	12
2.3.3	<i>Anforderungszuordnung.....</i>	12
2.4	SOFTWARE-/HARDWAREANALYSE.....	12
2.4.1	<i>Entwicklungsumgebung.....</i>	12
2.4.1.1	Programmiersprachen .....	14
2.4.1.2	Bibliotheken.....	15

2.5	IMPLEMENTIERUNG.....	15
2.5.1	<i>Datenbankentwurf (Datenschicht)</i> .....	16
2.5.1.1	Datenbankanalyse .....	16
2.5.1.2	Werkzeuge .....	17
2.5.1.3	ER-Modell/Tabellen.....	18
2.5.2	<i>Anwendungslogik (Geschäftsmodell)</i> .....	26
2.5.2.1	Einen einzelnen Prozess erfassen: Diesel einkaufen .....	26
2.5.2.2	Verwaltung von Feldstücken, Schlägen und Bewirtschaftungseinheiten .....	30
2.5.3	<i>Präsentationslogik</i> .....	31
2.5.3.1	Struktur der WEB-Site .....	31
2.5.3.2	Struktur eines Prozesses .....	34
2.5.3.3	Struktur der Erfassung von Feldstücken, Schlägen und Bewirtschaftungseinheiten .....	36
2.5.4	<i>Strukturen des Softwareprojektes</i> .....	42
2.5.4.1	Übersicht .....	42
2.5.4.2	Kernfunktionen in den JS-Dateien .....	43
2.5.4.3	Kernfunktionen in den php-Dateien .....	45
2.5.4.4	Fallstudie eines Datenaufrufes .....	48
2.6	TESTS .....	49
2.7	INTEGRATION UND EINFÜHRUNG .....	49
2.8	FEEDBACK/AUSBLICK.....	49
<b>3</b>	<b>ANHANG .....</b>	<b>50</b>
3.1	ABBILDUNGSVERZEICHNIS .....	50

# 1 Orientierung

Die Erstellung betrieblicher Ökobilanzen kann aus Sicht der Datenverarbeitung ganz grob in die drei Bereiche Datenerfassung/Kontrolle, wissenschaftliche Bewertung und Erstellung von Ergebnissen unterteilt werden. Jeder Bereich benötigt geeignetes Wissen/Werkzeuge aus dem naturwissenschaftlichen und technischen Wissensfeld und wendet sich in der Anwendung an unterschiedliche Adressaten.

- Die Datenerfassung bzw. Kontrolle bildet dabei den Einstieg für den landwirtschaftlichen Betrieb. Dieser soll im Laufe des Jahres seinen Betrieb umfassend dokumentieren. Das dazu notwendige System benötigt hinsichtlich verschiedener Betriebssystem und Arbeitsschritte ein hohes Maß an Variabilität. Zugleich müssen die einzelnen Erfassungsstrukturen aber systematisch angelegt werden um eine spätere Kontrolle überhaupt möglich zu machen. Dieser Zwischenbericht zeigt das Vorgehendmodell und die bisher entwickelten Strukturen in diesem Bereich auf.
- Die wissenschaftliche Bewertung der Daten findet in Modulen des Basissystems SALCA statt. Die notwendigen nationalen Anpassungen werden im Rahmen der wissenschaftlichen Begleitgruppen besprochen und in einem späteren Bericht dargestellt.
- Die Darstellung der Ergebnisse für die landwirtschaftlichen Betriebe wird im Zwischenbericht nur konzeptionell angesprochen. Das Endprodukt wird erst bei einem Workshop mit Beratern im kommenden November definiert. Ziel ist jedenfalls eine umfassende Betriebsdokumentation, die auf die Stärken und Schwächen der Betriebe hinweist und allfällige Beratungsnotwendigkeiten unterstützt bzw. einleitet.

## 2 V-Modell Softwareengineering CAP

Das „Vorgehens-Modell“ ist als Werkzeug des Software-Engineering ein Standard-Ablaufmodell das einen geordneten Projektablauf gewährleistet. Üblicherweise wird das V-Modell abstrakt entwickelt und von der Projektgruppe mit Werkzeugen des Projektmanagements umgesetzt. Der vorliegende Zwischenbericht weicht von dieser Technik aus praktischen Gründen etwas ab. Die ersten Stufen des V-Modells wurden abstrakt gehalten ab 2.4 wird aber die tatsächliche Implementierung dargestellt. Diese Technik vereint das theoretische Konzept mit der praktischen Umsetzung.

## 2.1 Rahmen

Der Rahmen befasst sich einleitend mit der Komplexität und den Ansätzen der Gesamtaufgabe. Diese Aspekte stehen über der eigentlichen Aufgabe der Softwareentwicklung und beleuchten das Projektumfeld.

### 2.1.1 Modularität

Der gelegentlich auftretende Wunsch nach einer kompakten Softwarelösung die alle drei Bereiche integriert ist aus der Sicht von Anwendergruppen verständlich, führt aber aus der Gesamtsicht auf den Prozess zu gravierenden Nachteilen. Das Erfassungsmodul und die Präsentation der Ergebnisse stellen eine Mensch-Maschinen Schnittstelle dar, die möglichst gut an die nationalen/gesellschaftlichen Gepflogenheiten angepasst werden müssen. Die Ökobilanzierung betritt im Kontext der nationalen Beratungswerkzeuge kein Neuland, sondern wirft ein neues Licht auf die Aspekte der landwirtschaftlichen Produktion. Sicherlich werden hierbei neue Parameter eingeführt, aber das Grundkonzept der Betriebserfassung muss konstant bleiben.

Ein Beispiel dafür ist der Umgang mit den landwirtschaftlichen Flächen. In der österreichischen Gewohnheit des Mehrfachantrages wird dabei von Feldstücken und Schlägen gesprochen. Das Feldstück wird dabei als eine unabhängige räumliche Definition verstanden, die sich vom rechtlichen Grundbegriff des Grundstückes abgehoben hat. Feldstücke sind nicht an Grundstücksgrenzen gebunden. Die eigentliche pflanzenbauliche Bewirtschaftung wird als Schlag bezeichnet und findet räumlich in einer 1..n Beziehung innerhalb des Feldstückes statt. Diese Struktur kann nicht in allen Ländern vorausgesetzt werden und würde zur Einführung von Pseudo-Strukturen führen, die schwer zu kommunizieren wären.

Dieses Beispiel ist sinngemäß auch auf die Präsentation von Ergebnissen auszudehnen. Wenn es etwa schon eingeführte Symboliken zur Darstellung von Größen gibt, sollte diese auch Anwendung finden. Man denke an die Darstellung von Energieeffizienzklassen die inzwischen europaweit vereinheitlicht wurde.

Ein modulares Konzept ist selbstverständlich auch Grundvoraussetzung für die Anpassung der Bewertungsmodelle an die nationalen Besonderheiten. Ziel der länderübergreifenden Stakeholder im Themenfeld der Ökobilanzierung ist ein konstanter, transnational akzeptierter Kern von aufeinander abgestimmten Modellen. Die Ausweitung des Kernes auf andere Länder bringt neue Facetten in dessen Entwicklung. Ein Beispiel dafür sind die klimatischen Gegebenheiten der teilnehmenden Länder. Die Niederschläge in der Schweiz bleiben an ihrer unteren Grenze deutlich über jenen des pannonischen Klimagebietes im Osten von

Österreich. Dies hat beispielsweise starken Einfluss auf die N-Auswaschung und muss im Modell deshalb entsprechend erweitert werden. Im Laufe der Zeit wird sich in der Kernkompetenz eine umfassende Bibliothek an Methoden und Zugängen anreichern. Neben diesen natürlichen Notwendigkeiten wird der methodische Kern auch noch durch die nationale Verfügbarkeit von Basisdaten beeinflusst. Die Variabilität in diesem Bereich ist noch deutlich größer als die möglichen Unterschiede in den Modellen.

Die Frage der Modularität führt zu folgender Erkenntnis

1. Zieht ein Ökobilanzierungskonzept bewusst eine weitere oder engere Linie um eine Produkt-/ Kunden-/Anwendergruppe und verzichtet das Konzept wissentlich auf eine Vergleichbarkeit bzw. international anerkannte Akzeptanz, kann eine Entwicklung kompakt in einer Software abgebildet werden. Diese Strategie kann fallweise hinter dem methodischen Vorgehen von Marketingkonzepten vermutet werden. Einige ältere Produkte wurden ebenfalls mit dem Blick auf eine nationale Anwendergruppe als Softwarepaket entworfen.
2. Alle anderen Ansätze führen in die Modularität und die Notwendigkeit einer interdisziplinären Anwendergruppe.

### **2.1.2 Kompetenzen**

Im Allgemeinen, vor allem aber bei dem breiter gefassten Ansatz eines modularen Entwurfes wie in FarmLife, sind einige Schlüsselkompetenzen bereitzustellen und aus der Sicht der Softwareentwicklung mit folgenden Aufgaben zu versehen:

- Projektmanagement: Sicherung der notwendigen Ressourcen, im speziellen der notwendigen Fachkompetenz in der Ökobilanzierung und Informatik
- Fachkompetenzen: Naturwissenschaftliche (Landwirtschaftliche) Kompetenz zur Bewertung von Stoffströmen und LCA-Experten
- Kommunikation: Entwicklung der Mensch/Maschinen-Schnittstelle bei der Datenerfassung und bei der Ergebnispräsentation
- Programmierung: Plattformunabhängige Erstellung und Test von Programmteilen und Schnittstellen in Kooperation mit der Datenmodellierung und den Fachkompetenzen.
- Datenmodellierung: Aufbau und Verwaltung relationaler und objektorientierter Datenstrukturen

Werden die Kompetenzen nicht vollständig besetzt, fallen die Aufgaben automatisch in die ranghöhere Ordnung und werden - je weiter entfernt vom Ursprung - immer

weniger bzw. schlechter bewältigt! Ein klassisches Beispiel ist die Fachkompetenz, die ohne Informatikbackground an der Entwicklung von Gesamtsystemen arbeitet.

## 2.2 Systemanforderung

Die Anforderungsanalyse im Sinne der Erstellung eines Lasten- und Pflichtenheft wird nicht umgesetzt, da die Konzeption und Entwicklung der Erfassungssoftware im Projekt FarmLife in Personalunion mit der Softwareentwicklung umgesetzt wird und der formelle Aufwand gering gehalten wurde.

### 2.2.1 Funktionale Anforderungen

#### 2.2.1.1 Stammdaten

**Inhalte:**

- Als primäre Stammdaten eines Betriebes werden die interne ID des Betriebes und seine Betriebsnummer aus dem INVEKOS mit zugeordneten Adressstamm bezeichnet. Der Stamm wird durch ein Login in Form einer Email-Adresse und durch ein Passwort erweitert.
- Der primäre Datenstamm wird durch die Kontaktdaten eines zugeordneten Fachberaters ergänzt.
- Als sekundäre Stammdaten werden die Angaben zur Erfassungsstrategie des Betriebes bezeichnet. Betriebe haben aus der Sicht des Projektbetreibers mehrere Möglichkeiten für die Erfassung der Daten. Diese Möglichkeiten sind
  - die manuelle Erfassung im Handbuch
  - die vollständige Erfassung in CAP
  - Mischformen aus manueller Erfassung und CAP
  - die Übergabe von Daten aus anderen (lesbaren) Quellen
- Die Stammdaten beinhalten auch eine Erklärung über die Zustimmung zur elektronischen Verarbeitung der Betriebsdaten

**Verarbeitung:**

Die Inhalte müssen zur Erfassung strukturiert dargestellt, verändert und gespeichert werden.

#### 2.2.1.2 Geodaten der Feldstücke und Schläge

**Inhalte:**

- Kleinräumige Bewirtschaftungseinheiten müssen mit geeigneten Werkzeugen auf räumlichen Grunddaten mit geeigneter Auflösung erfasst und dargestellt werden. Die Erfassung ist in der Form sich schließender Polygone

umzusetzen, wobei die Genauigkeitsanforderung geringer ist als die Forderung der nationalen Rechtslage für Grenzpolygone

- Polygone müssen mit ihrer Kontenstruktur als WGS-84 Koordinaten gespeichert werden
- Aus den Polygonen muss die Fläche berechnet werden.

#### **Verarbeitung:**

Für die räumliche Orientierung sind – wenn vorhanden – bestehende Feldstückpolygone darzustellen. Diese sollen für die Digitalisierung der Schläge bei Bedarf übernommen werden. Freie Schläge sind mit einfachen Werkzeugen Eckpunkt für Eckpunkt zu digitalisieren und zu speichern. Die Überarbeitung gespeicherter Polygone muss möglich sein. Polygone müssen gelöscht werden können. Jede Veränderung der Polygone muss eine Neuberechnung der Flächengröße auslösen.

### **2.2.1.3 Erfassung von Ein- und Verkauf**

#### **Inhalte:**

- Alle Betriebsmittel die zugekauft werden sind in Art, Menge und Kosten zu erfassen. Für die Erfassung ist folgende Grobstruktur in einzelne Produktgruppen vorzusehen:
  - Fertigfutter, Einzelfuttermittel, Futtermittelmischungen
  - Energie/Fett/Öle
  - Wirtschaftsdünger
  - Handelsdünger
  - Pflanzenschutz
  - Saatgut/Pflanzgut
  - Tiere
  - Futterkonservierung
  - Sonstiges Kleinmaterial Tierhaltung
- Alle Betriebsmittel, die verkauft werden sind in Art, Menge und Kosten zu erfassen. Für die Erfassung ist eine klare Grobstruktur in einzelne Produktgruppen vorzusehen
  - Feldfrüchte
  - Tiere
  - Milch
  - Wirtschaftsdünger

#### **Verarbeitung:**

Für die einzelnen Erfassungsaktionen ist ein klar strukturiertes Werkzeug mit den Möglichkeiten Speichern, Ändern und Löschen zu entwickeln



#### 2.2.1.4 Erfassung der Feldarbeit

##### Inhalte:

- Alle Feldarbeiten, die durchgeführt werden, sind in granularer Form zu erfassen. Die Erfassung soll je Arbeitsprozess individuell gestaltet werden und muss auf jeden Fall die Bereiche
  - Zuordnung der Arbeit zu Fläche
  - Spezifikation der Arbeit in ihrer
    - Geräteausstattung
    - Hilfsmiteileinsatzbeinhalten.
- Für die Erfassung ist folgende Grobstruktur in einzelne Produktgruppen vorzusehen:
  - Allgemeine Arbeiten
    - Handelsdünger ausbringen
    - Pflanzenschutz ausbringen
    - Beregnen von Kulturen
    - Transport von Gütern
    - Forstarbeiten
    - Winterdienst
  - Arbeiten im Grünland
    - Pflege
    - Wirtschaftsdünger ausbringen
    - Ernten
    - Beweiden
    - Weidepflege
  - Arbeiten im Ackerland
    - Bodenbearbeitung
    - Wirtschaftsdünger ausbringen
    - Saat/Pflanzung
    - Pflege
    - Ernte
    - Verarbeitung
  - Arbeiten im Weinbau
    - Schneiden
    - Pflege

##### Verarbeitung:

Für die einzelnen Erfassungsaktionen ist ein klar strukturiertes Werkzeug mit den Möglichkeiten Speichern, Ändern und Löschen zu entwickeln

### **2.2.1.5 Erfassung von pflanzenbaulichen Erträgen**

#### **Inhalte:**

- Alle pflanzenbaulichen Erträge des Erfassungszeitraums sind in Art und Menge zu erfassen
- Für die Erfassung ist folgende Grobstruktur in einzelne Produktgruppen vorzusehen:
  - Grundfuttermittel
  - Marktfrüchte

#### **Verarbeitung:**

Für die einzelnen Erfassungsaktionen ist ein klar strukturiertes Werkzeug mit den Möglichkeiten Speichern, Ändern und Löschen zu entwickeln

### **2.2.1.6 Erfassung des Inventarbestandes**

#### **Inhalte:**

- Zum Betriebsinventar gehören die Maschinen und Gebäude eines landwirtschaftlichen Betriebes. Das Inventar ist in einer Form zu erheben, die zur späteren Bewertung geeignet ist.
- Bei Maschinen ist dies eine Zuteilung der Maschinen zu einem vorgegebenen Maschinentyp der eine Verbindung zum Gewicht der Maschine zulässt. Dies wird bei Zugmaschinen die Leistung, bei Transportgeräten die Ladekapazität und bei Bearbeitungsgeräten die Breite sein.
- Für Gebäude ist eine Struktur vorzusehen, die über die Konfiguration und Größe des Gebäudes Auskunft gibt. Dazu sind die funktionellen Einheiten des Gebäudes mit ihrer Fläche bzw. mit ihrem Volumen zu beschreiben.

#### **Verarbeitung:**

Für die einzelnen Erfassungsaktionen ist ein klar strukturiertes Werkzeug mit den Möglichkeiten Speichern, Ändern und Löschen zu entwickeln

### **2.2.1.7 Variable Mengenerfassung**

Sind in den einzelnen inhaltlichen Bereichen Mengen zu erfassen, sind eine geeignete Anzahl an Einheiten bzw. Bezugsgrößen vorzusehen. Die Landwirte sollen ihre Erfassungsgröße nicht erst in vorgegebene Zwangsgrößen umrechnen. Die Düngung von Schlägen mit Handelsdünger wird in der Regel mit einer Konzentrationsgröße erfolgen, da das Düngergerät auf die Ausbringmengen pro ha eingestellt werden muss. Bei der Ausbringung von Wirtschaftsdünger wird eher die

Mengengröße erfasst. So verwendet Bauer X für seine Schläge 1/2/3 in Summe 10 Güllefässer mit je 10 m<sup>3</sup>.

#### **2.2.1.8 Listenelemente**

Mit Ausnahme von Datumsangaben und Mengengrößen sind alle anderen Eingaben in der Form von Auswahllisten zu definieren. Eine freie Definition von Geräten/Produkten/Hilfsstoffen/... ist nicht möglich. Der dadurch entstehende Eingabeengpass ist durch ein Kommunikationsmedium zu kompensieren.

#### **2.2.1.9 Status der Eingabe/Mitteilungen**

Die Einengung durch Listenelemente, aber auch alle andern Systemdefizite sollen pro Erfassungselement mitteilbar sein. Gekennzeichnete Elemente sind in den Erfassungslisten als solche darzustellen. Eine Nicht-Erfassung wird so unterbunden, da zumindest eine Nachricht hinterlassen werden kann.

#### **2.2.1.10 Prüfung**

Eine Prüfung auf Vollständigkeit und Inhalte wird nur im Bereich der Stammdaten, bei Datumsangaben und bei der Feldwirtschaft auf eine Mindestauswahl an Schlägen umgesetzt. Die sachliche Prüfung ist in eine eigene Spezifikation auszulagern (CON), da diese erst am Jahresende sinnvoll durchgeführt werden kann.

### **2.2.2 Nicht-funktionelle Anforderungen**

#### **2.2.2.1 Design**

Die Oberflächengestaltung ist graphisch einfach und mit geringer Schachtelungstiefe zu gestalten.

#### **2.2.2.2 Zugang**

Die Erfassung der Daten ist zentral zu organisieren. Daten werden ausnahmslos in einer Datenbankinstanz gespeichert. Die bisher angesprochenen Spezifikationen sind somit als Web-Anwendung oder als Small-Client zu implementieren.

#### **2.2.2.3 Datensicherheit/Datenschutz**

Die zentral erfassten Daten sind regelmäßig zu sichern und gegen Fremdzugriffe zu schützen. Im Rahmen der Weiterverarbeitung/Auswertung ist eine Anonymisierung zu verwenden. Dies gilt nicht nur für die numerische/grafische Gestaltung von Ergebnissen sondern auch für die kartografische Darstellung.

#### **2.2.2.4 Schnittstellen**

Die nationalen Datenerfassungssysteme sind vorab auf eine Einsatzmöglichkeit automatisierter Schnittstellen zu prüfen.

### **2.3 Systemarchitektur**

#### **2.3.1 Technischer Aufbau des Systems**

##### **2.3.1.1 Datenbank-CAP**

Die Daten sind in einer relationalen Datenbank zu speichern. Für den Datenbankentwurf sind die einzelnen Entitäten bis in ihre dritte Normalform zu zerlegen. Die Datentabellen sind mit geeigneten Schlüsseln zu versehen. Definierte Beziehungen zwischen den Tabellen stellen die referenzielle Integrität der Daten sicher.

##### **2.3.1.2 Server-CAP**

Die CAP-Datenbank und die Software – wenn sie als Serverlösung entwickelt wird – sind auf einem gesicherten Web-Server zu installieren. Diese ist mit aktuellen Schutzmaßnahmen zu versehen und über die Firewall entsprechend gegen Fremdzugriffe zu sichern.

##### **2.3.1.3 Master-Control-Station**

Lokaler Arbeitsplatz innerhalb der Ökobilanzierungsgruppe der mit geeigneten Werkzeugen ausgestattet ist. Zur Werkzeugausstattung gehört neben allgemeinen Programmen die Installation einer GIS-Software und speziellen LCA-Programmen.

##### **2.3.1.4 Entwicklungsumgebung-CAP**

Die Wahl der Entwicklungsumgebung und Programmiersprache kann grundsätzlich frei gestaltet werden, wird aber durch die Verwendung von externen Komponenten maßgeblich beeinflusst. Während einfache Erfassungseinheiten beliebig entwickelt werden können, ist bei der Implementierung der räumlichen Erfassung auf bestehende Systeme zurückzugreifen. Diese Auswahl wird die Entwicklungsumgebung stark beeinflussen.

##### **2.3.1.5 Gesamtsystem**

Auf CAP folgt im logischen Systemablauf das Kontrollmodul CON. Während CAP zentral implementiert wird, kann CON besser lokal auf einer Master-Controll-Station umgesetzt werden. CON kann im Gesamtsystem dadurch besser auf andere

Werkzeuge, z.B. spezielle GIS-Software zurückgreifen. CAP und CON sind über eine leistungsfähige ODBC-Schnittstelle zu verbinden.

### 2.3.2 Identifikation von Schnittstellen

Schnittstelle	Zweck/Umsetzung
CAP/CON	Füllen von CAP, Kontrolle durch die Master-Control-Station ODBC
CLIENT/SERVER	HTTP

### 2.3.3 Anforderungszuordnung

Anforderung	Technische Komponente
Inhaltliche Struktur der Stammdaten, Geodaten und Fachdaten der/des Ein/Verkaufs, Feldarbeit, Erträge, Inventar	Datenbank-CAP
Design	Entwicklungsumgebung, Softwaretechnik
Zugang, Datensicherheit	Datenbank-CAP, Server
Schnittstellen	Server, Master-Control-Station

## 2.4 Software-/Hardwareanalyse

### 2.4.1 Entwicklungsumgebung

Die Projektgruppe FarmLife hat sich bei der Entwicklung von CAP für eine Web-Lösung entschieden. Nach der Erstellung der Anforderungsanalyse wurde klar, dass mit Ausnahme des GIS-Teiles keine außergewöhnlichen Anforderungen zu erfüllen sind und der Großteil der Mensch-Maschinen-Kommunikation im Auswählen von Elementen bzw. im Ausfüllen von Wertzahlen besteht. Die Entwicklungsumgebung wird im Projekt CAP letztlich durch den GIS-Teil bestimmt. Die Marktanalyse zu möglichen Komponenten ergibt folgendes Bild:

- Arc Gis Web Server

- Kommerzielles Produkt der oberen Preisklasse
- Maximale Werkzeugunterstützung bei hoher Performance
- Ident zum Anbieter des GIS-Werkzeuges auf der Master-Control-Station
- Eigene Serverstruktur notwendig
- Komplex
- Programmierinterface: .NET
- OpenLayer
  - Open-Source Produkt
  - Breite Werkzeugunterstützung
  - Beliebig in Web-Frameworks zu integrieren
  - Bewährt
  - Komplex
  - Programmierinterface: Java Script/XML
- Leaflet
  - Open-Source Produkt
  - Minimale Werkzeugunterstützung
  - Beliebig in Web-Frameworks zu integrieren
  - Neu
  - Einfach
  - Programmierinterface: Java Script/XML

Für CAP wurde vom Entwickler letztlich Leaflet ausgewählt, weil dieses Produkt die Anforderungen adäquat abbildet und in der Entwickler-Community von Leaflet derzeit eine hohe Entwicklungsfrequenz spürbar ist. Mit dieser Entscheidung wurde die Auswahl der Programmiersprache und Bibliotheken weitgehend mitentschieden.

Für die Codierung des Quellcodes im Rahmen der Implementierung wurde Komodo Edit, eine Open-Source-Editor gewählt.

Durch die Wahl sehr leichtgewichtiger, allgemeiner Komponenten kann CAP auf jeden Linux-Server mit Apache oder auf Windows-Server mit IIS betrieben werden.

### 2.4.1.1 Programmiersprachen

CAP wird als leichtgewichtiger Client entwickelt, der von allen Web-Browsern mit Java-Script unterstützt werden kann. Für den Austausch von Daten zwischen dem Anwender und dem Server wird php (Hypertext Preprocessor) in der Form von AJAX verwendet.

- Java Script: Einfache Skriptsprache die auf dem Computer des Anwenders – in der Regel in den Web-Browser – integriert ist. JS wird innerhalb des Browsers in einer Sandbox ausgeführt. Das heißt die Anwendung hat Zugriff auf die Objekte des Browsers, kann aber nicht auf die Festplatte des Computers zugreifen. Die Anwendung bleibt als im Browser-Fenster gefangen. Diese Maßnahme schützt die Anwender von Angriffsversuchen aus dem Internet. Java-Script übernimmt in CAP die Aufgabe die Bildschirmseiten aufzubauen und auf die Aktionen der Anwender zu reagieren. Beispiel: Häufig sind die Auswahllisten von Elementen hierarchisch geschachtelt (mehr dazu in der Datenbankanalyse) und es können auf der ersten Ebene verschiedene Arten von Zug-/Antriebsmaschinen (Allrad, Einfach, Bergtraktor, Mährescher, ...) ausgewählt werden. Die zweite Ebene sieht die Leistung der Maschinen als beschreibenden Parameter vor. Wenn nun der Anwender einen Typ auswählt, wird diese Aktion via Java-Script als „Event“ erkannt. Dazu wurden sogenannte Event-Listener implementiert. Dieser ruft in Folge eine php-Prozedur am Server auf, die in der Lage ist, die notwendigen Daten aus der Datenbank zu holen und an den Client zurück zu senden. Der aufrufende Java-Script-Listener fängt diese Daten auf und füllt damit das Element der zweiten Ebene. Nun kann dort ein geeigneter Eintrag gewählt werden. JS liegt aktuell in der Version 1.8.5 vor.
  - Quelle: <http://www.w3schools.com/js/>
- php: Übernimmt in CAP die Kommunikation zwischen aufrufenden JS und der am Server installierten Datenbank. php wird dadurch zum Sprachrohr der Datenbank. Verwendete Version 5.4.20.
  - Quelle: <http://php.net/>
- SQL: Alle Anweisungen an die Datenbank werden in SQL definiert und von php ausgeführt.
  - Quelle: <http://www.w3schools.com/sql/>

### 2.4.1.2 Bibliotheken

- Bootstrap: Für den Aufbau der Web-Seite ist neben der logischen Konzeption auch eine Visualisierung zu planen. Diese besteht nicht nur im Design sondern vor allem in der notwendigen Anpassungsdynamik an verschiedene Endgeräte. Web-Anwendungen laufen heute am Standgerät ebenso wie auf mobilen Lösungen bis zum Smartphone. Diese Tatsache kann durch die Anwendung von CSS in der HTML-Webseite dynamisch unterstützt werden. Idealerweise steht mit Bootstrap ein umfangreiches Framework für diese Aufgabe zur Verfügung. Bootstrap wurde für Facebook entwickelt und kann kostenlos verwendet werden. Die Anwendung erhält so ein modernes, dynamisches Design.
  - Quelle: <http://getbootstrap.com/>
- Leaflet: Als Geodatenplattform wurde Leaflet ausgewählt. Leaflet wurde als JS-Framework entwickelt und wird beim Aufruf von CAP vom Server auf den Client geladen. Die Größe des Gesamtpaketes ist dabei extrem klein und beträgt nur 31 KB. Dies garantiert, dass die Performenc der Anwendung auch bei schlechten Datenleitungen hoch bleibt. Leaflet wird von allen Browsern unterstützt und bietet seinerseits Unterstützung für alle kostenlosen Luftbilddatenquellen an. Es entsteht somit eine für CAP günstige Situation. Einerseits verfügen wir mit Leaflet über ein kleines Werkzeug, dass unsere Minimalanforderung – das Erfassen von Schlägen – erfüllt und andererseits liefert Leaflet selber mit BING ein weltweit verfügbares Luftbild. Leaflet kann gerade so viel wie wir benötigen.
  - Quelle: <http://leafletjs.com/>
- Geotools: Leaflet erfasst die Knoten von Punkten, Linien und Polygonen auf der Basis von WGS84. Dieses geodätische Referenzsystem ist weltweit gültig, weicht aber häufig von den lokal verwendeten Referenzsystemen ab. Damit aus den Gradangaben des WGS84 entsprechende Meterangaben im nationalen System entstehen, kann die Bibliothek Geotools verwendet werden. Diese steht ebenfalls kostenlos zur Verfügung.
  - Quelle: <http://sbin.dk/2013/02/09/geotools-a-geo-related-php-53-library/>

## 2.5 Implementierung

Aufbauend auf den bereits erläuterten Anforderungen der Anwender, des Gesamtsystems sowie der Soft-/Hardware kann die Implementierung nach dem 3-Tier-Modell (Datenschicht / Geschäftsmodell / Präsentationsschicht) entwickelt werden. Diese Maßnahme garantiert die Unabhängigkeit der einzelnen



Softwarekomponenten im Rahmen der Weiterentwicklung von CAP in anderen Institutionen. Bei geeigneter Architektur kann etwa die Präsentation verändert werden, ohne dass davon das Datenmodell betroffen wäre.

## **2.5.1 Datenbankentwurf (Datenschicht)**

### **2.5.1.1 Datenbankanalyse**

Für die Erstellung der Datenbank ist das aus dem Anforderungsprofil hervorgehende „Univers of Discourse“ – also die Modellwelt der Anwendung – analytisch zu zerlegen. Ziel der Analyse ist die Identifizierung der einzelnen Objekte, Teilnehmer, ... und deren Beziehung zueinander, um ein geeignetes Entity-Relationship-Modell (ER-Modell) zu erstellen. Folgende Elementbereiche werden für CAP definiert.

#### **2.5.1.1.1 Steuerung und Kommunikation der beteiligten Stakeholder (*org*)**

Im Focus der Erhebung von Betriebsdaten steht der landwirtschaftliche Betrieb. Der Betrieb kann mit einer Adresse und Betriebsnummer beschrieben werden. Der einzelne Betrieb wird von einem ihm zugeordneten Berater betreut, wobei der Berater mehrere Betriebe betreuen kann. Berater und Betrieb einigen sich bei mehrsprachigen Ländern auf eine Sprache und vereinbaren eine Erhebung für ein Erfassungsjahr. Mehrere können nebeneinander bestehen, wobei ein Jahr als aktuelles Jahr zu definieren ist. Für die Kommunikation zwischen der Projektgruppe, dem Berater und dem Landwirt ist eine Möglichkeit für allgemeine und betriebsspezifische Anweisungen vorzusehen.

#### **2.5.1.1.2 Erfassung und Bewirtschaftung der pflanzenbaulichen Kulturen (*gis*)**

Landwirtschaftliche Betriebe bewirtschaften auf ihren Grundstücken pflanzenbauliche Kulturen, die als Schlag bezeichnet werden. Innerhalb eines Schlages und Jahres besteht die Möglichkeit durchgehend eine Kultur oder eine Kulturreihenfolge anzubauen. Jede Kultur ist namentlich zu definieren und das Grenzpolygon/Flächenausmaß ist festzustellen. Jeder Schlag ist Teil einer langjährigen Fruchtfolge. Für den Schlag sind eine relative Ertragserwartung und einige zusätzliche Informationen über die Wassersituation bzw. Form zu erheben. Schläge sind Teile einer größeren räumlichen Einheit, die als Feldstück bezeichnet werden. Feldstücke tragen einen Namen. Ungeachtet der 1..n Beziehung zwischen Feldstücken und Schlägen sollen aus den Schlägen Bewirtschaftungseinheiten aggregiert werden. Die räumlichen Elemente definieren das WO bzw. WOHIN der Ökobilanzierung.

### 2.5.1.1.3 Hierarchische Strukturen zu freien Verwaltung von Sachbeständen

#### **(classes)**

Landwirtschaftliche Betriebe verfügen über eine breite Ausstattung an Gütern, Betriebsmitteln und Produkten. Diese werden bei Notwendigkeit zugeführt und verbrauchen sich mit der Zeit. Die Länge des Zeitraumes führt zu einer Unterscheidung in Güter des Anlagenbestandes und in Verbrauchsgüter. Die zeitliche Dimension beeinflusst nicht nur die Definition (die nicht wirklich maßgeblich ist) sondern klärt vor allem die Frage, ob Güter vollständig oder teilweise einem Bewirtschaftungsjahr zuzuordnen sind. Aus diesem Grund ist für Güter mit hoher Umsatzrate ein Inventurbestand zu definierten Fixterminen vorzusehen. Langfristige Güter wie Maschinen und Gebäude sind als klassische Inventur abzubilden. Für alle Arten von Sachgütern (kurzfristige oder langfristige) ist eine möglichst umfassende Vorschlagsliste zu entwerfen. Damit diese Liste in den Anwendungsfällen übersichtlich bleibt ist eine Hierarchie über 4 Ebenen zu definieren (Super, Main, Sub, Entity). Jede Ebene soll sprachenspezifisch gestaltbar sein. Die Liste der Sachelemente definiert strukturiert und verbindlich das WAS der Ökobilanzierung.

### 2.5.1.1.4 Strukturen zur freien Verwaltung von Parametern (**parameter**)

Die Sachbestände der Ökobilanzierung sind noch durch den Aspekt des WIE oder WIEVIEL zu beschreiben. Diese Beschreibung erfordert freie Strukturen zur Spezifikation von qualitativen oder quantitativen Größen. Auch hier soll teilweise ein hierarchischer Ansatz entworfen werden, weil manche Parameter besser als typisierte Vorgabe angeboten werden sollten. Ein Beispiel ist die Ausbringungszeit von Dünger. Wahlweise könnte die Uhrzeit als Messwert oder eine relative Zeitspanne (morgens, mittags, abends, ganztags) vorgegeben werden.

### 2.5.1.1.5 Strukturen zur freien Definition von Einzelprozessen (**action**)

Die Bereiche *classes* und *parameter* sollen in einer geeigneten Struktur logisch so gereiht werden, dass daraus Einzelprozesse entstehen. Diese Prozesse bilden die Grundlage für die Business-Logik und die Präsentation der Daten in CAP. Diese Einzelprozesse bilden die Grundlage von CON, da die Erfassungsprüfung aktionsorientiert anzulegen ist. Einzelaktionen werden im Rahmen der Erfassung ausgefüllt und als Entität in der Datenbank abgelegt, wobei die letzte Konzeption des Einzelprozesses als Vorlage bestehen bleibt.

## **2.5.1.2 Werkzeuge**

Die im Web-Development klassische Entwicklungskonfiguration mit den Sprachen Java-Script und php wird um die Open Source Datenbank MySql erweitert. Für die

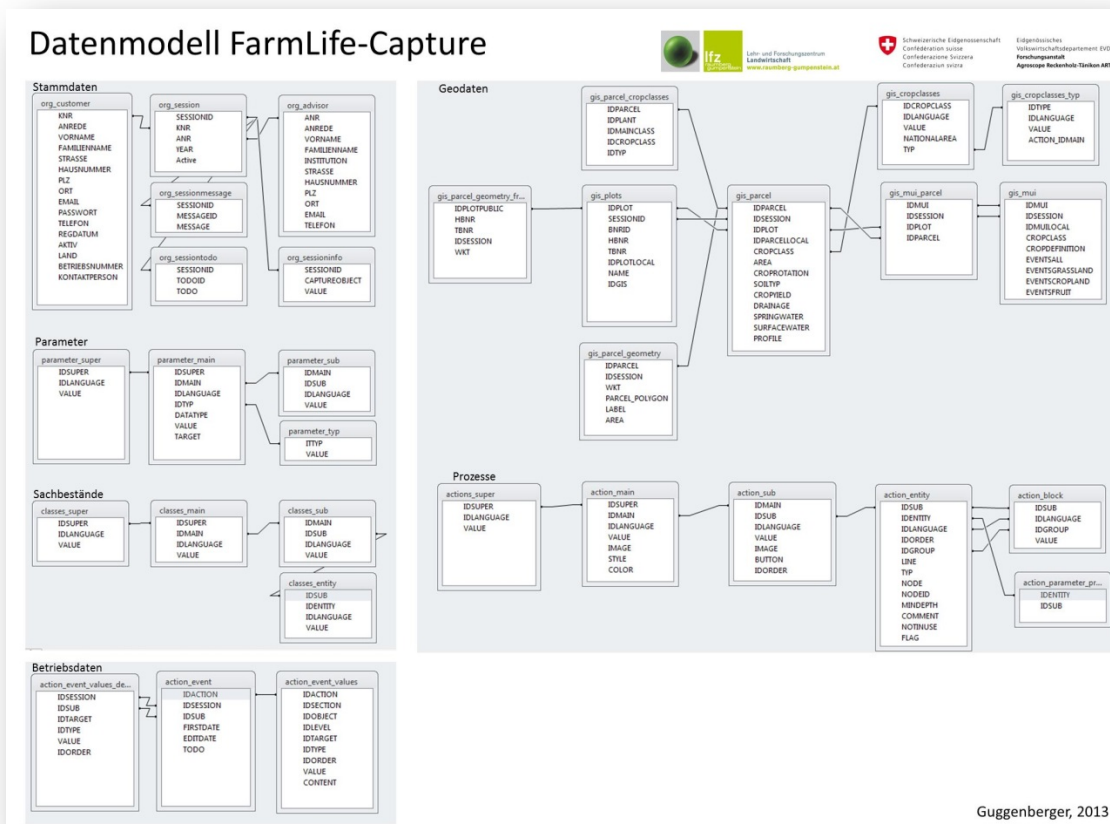
Entwicklung und Implementierung der eigentlichen Tabellen können die Werkzeuge MySQL Workbench bzw. das Tool FlySpeed SQL Query verwendet werden.

- Quelle: <http://www.mysql.de/>

### 2.5.1.3 ER-Modell/Tabellen

Die Übersicht teilt den physischen Datenbankentwurf in drei grobe Bereiche.

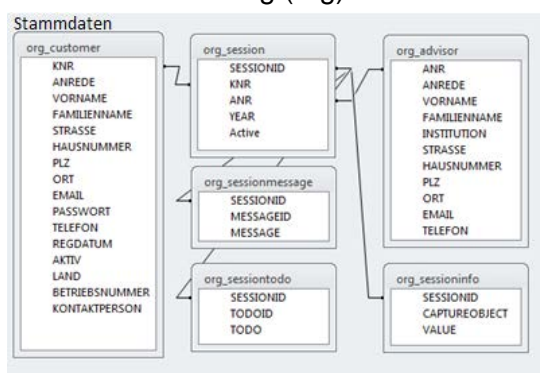
- Stammdaten, Parameter und Sachbestände bilden das Gerüst um einerseits die Erfassung zu steuern und andererseits eine hohe Variabilität mit vielen Sachbeständen zu gewährleisten. Die Beziehungen in den Stammdaten werden durch die SessionID definiert, die eine Sitzung pro Betrieb pro Jahr darstellt. Diese ID zieht sich durch den gesamten Entwurf.
- Geodaten und Prozesse sind individuelle gestaltete Modelle, wobei in Geodaten aktiven Daten einzelner Sitzungen gehalten werden. Prozesse besteht nur aus einer Grunddefinition verschiedener Abläufe und wird nie durch den Landwirt beeinflusst sondern nur durch die Projektgruppe
- Betriebsdaten speichert die eigentlichen Daten ab.



Guggenberger, 2013

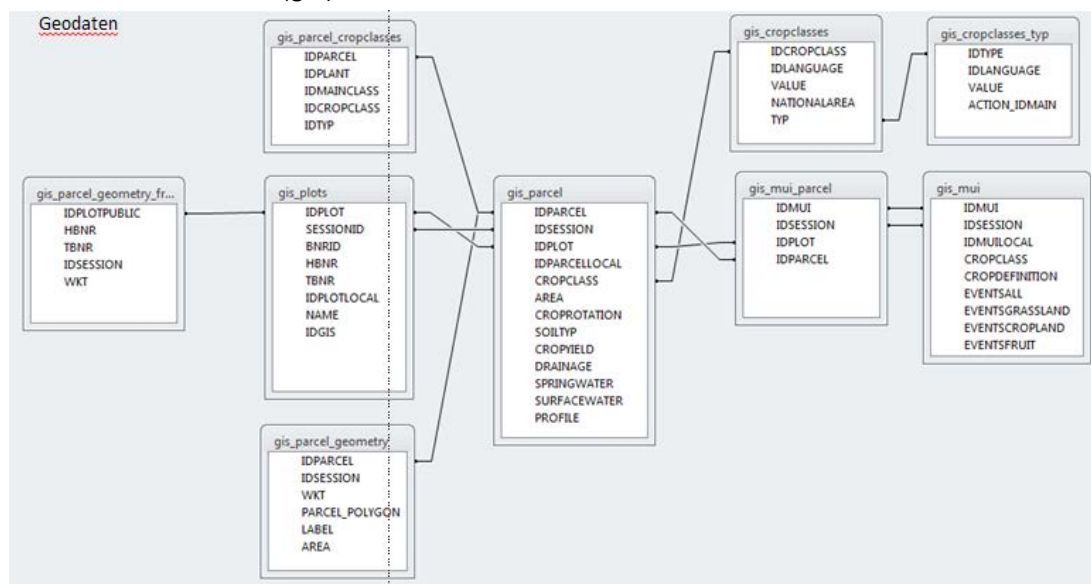
Abbildung 1: ER-Modell Übersicht

## 2.5.1.3.1 Steuerung (org)



Bezeichnung	Inhalt	Beziehungen	Bemerkung
<b>org_customer</b>	Kundendaten als klassischen Adresse	Schlüssel über die Kundennummer zur org_session	<b>Knr</b> ist eine einfache aufsteigende Nummer, <b>Aktiv</b> kann als Flag verwendet werden um den Kunden zu deaktivieren
<b>org_advisor</b>	Adresse des Berater	Schlüssel über die Beraternummer zur org_session	
<b>org_session</b>	Definiert eine Sitzung, die für den Betrieb ein einzelnes Jahr definiert	Die Tabelle entsteht erst durch die Verbindung von Betriebs- und Beraterdaten	Flag markiert das derzeit in der Anwendung aktive Erfassungsjahr
<b>org_sessioninfo</b>	Grundinformationen über die Erfassungstrategie einer Session	SessionID zur Tabelle org_session	
<b>org_sessioninfo</b>	Allgemeine Meldungen an alle FarmLife-Benutzer	SessionID zur Tabelle org_session	Informationen über Veranstaltungen oder Systemänderungen
<b>org_sessionmessage</b>	Betriebspezifische Meldungen an einen FarmLife-Benutzer	SessionID zur Tabelle org_session	

## 2.5.1.3.2 Geodaten (gis)



Bezeichnung	Inhalt	Beziehungen	Bemerkung
<b>gis_plot</b>	Definition der Feldstücke der Betriebe mit eindeutiger Betriebsnummer und Feldstücknamen	Schlüssel zum Feldstückpolygon das als Referenzrahmen für das Digitalisieren der Schläge zur Verfügung gestellt werden kann.	<b>IDPLOT</b> ist ein laufend ansteigender Schlüssel, <b>IDPLOTLOCAL</b> ist jene Feldstücknummer, die der Betrieb beim Mehrfachantrag verwendet.
<b>gis_parcel</b>	Definition eines Schleges mit den in der Anforderungs-liste definierten Parametern	Multiple Beziehungen zu fast allen Tabellen als Parent oder Child-Objekt	
<b>gis_parcel_geometry</b>	Geometrie eines Schleges ausgedrückt als Well-Known-Text (WKT)	1:1 Beziehung zu gis_parcel	WKT wird unterhalb der Tabelle noch genauer dargestellt.
<b>gis_parcel_geometry_frame</b>	Geometrie der Feldstücke zur Orientierung bei der Erfassung der Schläge	[0,1]:1 Beziehung zu gis_plot	Die Geometrie wird aus dem nationalen Bestand an Geodaten übernommen
<b>gis_parcel_croplasses</b>	Auflisten der Kulturen die je Erfassungsjahr	n:1 Beziehung mit gis_parcel	Die Definition der Schlagnutzungen wurde

	am Schlag angebaut werden		von der nationalen Liste für Schlagnutzungsarten aus dem INVEKOS entnommen. IDTYP definiert Haupttypen wie Grünland, Ackerland, ...
<b>gis_cropclass</b>	Struktur der nationalen Schlagnutzungsarten	1:1 bzw. n:1	
<b>es und gis_cropclass es_typ</b>		Beziehung zu gis_parcel	
<b>gis_mui</b>	Abstrakte Formulierung von Bewirtschaftungseinheiten	1:n Beziehung mit gis_mui_parcel	<b>Cropclass</b> kann offen bleiben damit beliebige Schläge eingefügt werden. Wird eine Nutzung definiert, dürfen nur mehr Schläge einer Art verbunden werden.
<b>gis_mui_parcel</b>	Einzelne Schläge einer Bewirtschaftungseinheit	1:1 Beziehung zu gis_parcel, n:n Beziehung zur gis_mui	

#### 2.5.1.3.2.1 Well Known Text (siehe Open Geospatial Consortium bzw. GIS-WIKI)

WKT bildet Geometrien einfacher Geodatentypen in Form einer strikt definierten Zeichenkette ab. Diese Technik unterstützt Punkte, Linien und Polygone. Die Abbildungsform folgt dem abgebildeten Muster:





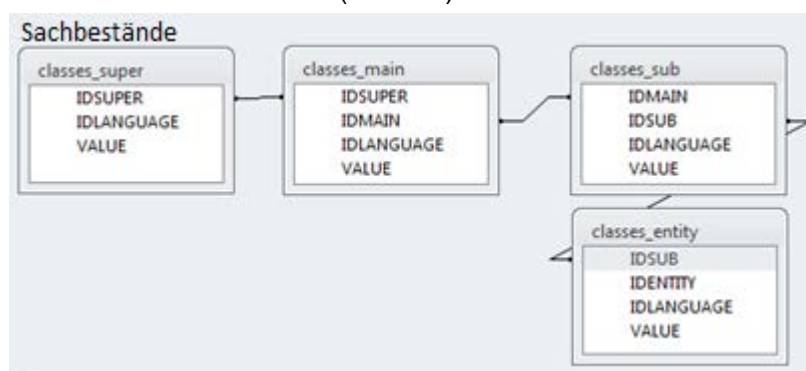
Geometry primitives (2D)		
Type	Examples	
Point	POINT (30 10)	
LineString	LINESTRING (30 10, 10 30, 40 40)	
Polygon	POLYGON ((30 10, 10 20, 20 40, 40 40, 30 10))	
	POLYGON ((35 10, 10 20, 15 40, 45 45, 35 10), (20 30, 35 35, 30 20, 20 30))	

Abbildung 2: WKT aus en.wikipedia.org

## 2.5.1.3.3 Sachbestände (classes)



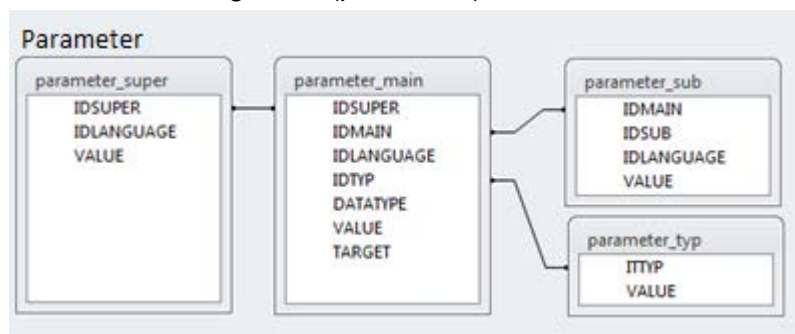
Bezeichnung	Inhalt	Beziehungen	Bemerkung
<b>classes_super</b>	Wurzelknoten des hierarchischen Baumes an Sachbeständen	1:n Beziehung zur hierarchisch tieferen Ebene	Logischer Ausgangspunkt der praktisch kaum Anwendung findet z.B. Hilfsstoffe
<b>classes_main</b>	Hauptknoten im logischen Sprachgebrauch	1:n Beziehung zur hierarchisch tieferen Ebene	z.B. Handelsdünger
<b>classes_sub</b>	Hauptknoten im Gebrauch einzelner Sachbestandsgruppen	1:n Beziehung zur hierarchisch tieferen Ebene	z.B. NPK-Dünger
<b>classes_entity</b>	Einzelnes Betriebsmittel / Sachbestand		z.B. Linzer Pro

## 2.5.1.3.3.1 Info für CON

Für die Überstellung der Sachbestandgruppen in eigentliche Vorleistungen sind 6 Funktionsbereiche zu schaffen. Diese prüfen in der Main die einzelnen Gruppen und verbinden über die Sub alle eigentlichen Sachbestände.

Beispiel: Maschinen spielen in der Ökobilanz auch deshalb eine Rolle, weil für ihre Produktion Vorleistungen erbracht wurden. Diese können über die kg einer definierten Maschinenstruktur angelastet werden. Die Transformation startet also in der Super-Klasse mit dem Auftrag für alle Maschinearten in Main (derzeit 41 Klassen) ein Gewicht zu ermitteln. Die Funktion steigt über Sub zu den eigentlichen Geräten ab und entnimmt ein Gewicht aus einer Funktionssammlung oder aus einer begleitenden Datenbank.

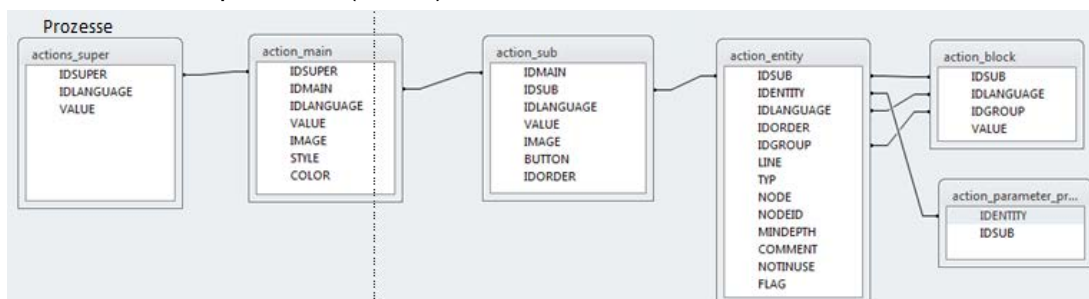
## 2.5.1.3.4 Inhaltsgrößen (parameter)



Bezeichnung	Inhalt	Beziehungen	Bemerkung
<b>parameter_super</b>	Wurzelknoten des hierarchischen Baumes an Parameter	1:n Beziehung zur hierarchisch tieferen Ebene	Logischer Ausgangspunkt der praktisch kaum Anwendung findet z.B. Einheiten
<b>parameter_main</b>	Hauptknoten im logischen Sprachgebrauch	1:n Beziehung zur hierarchisch tieferen Ebene	Hauptklasse die Informationen zur Struktur der Parameter liefert: IDTYP: 1 = Element es darf eine Sub-Liste erwartet werden. 2=Datentype eines konkreten Wertes z.B. Mengeneinheiten
<b>parameter_sub</b>	Hauptknoten im Gebrauch einzelner Sachbestandsgruppen	1:1 Beziehung zu den Typen	z.B. Kg, Liter, m <sup>2</sup> , ...
<b>Parameter_typ</b>	Erklärende Randklasse mit geringer Bedeutung		

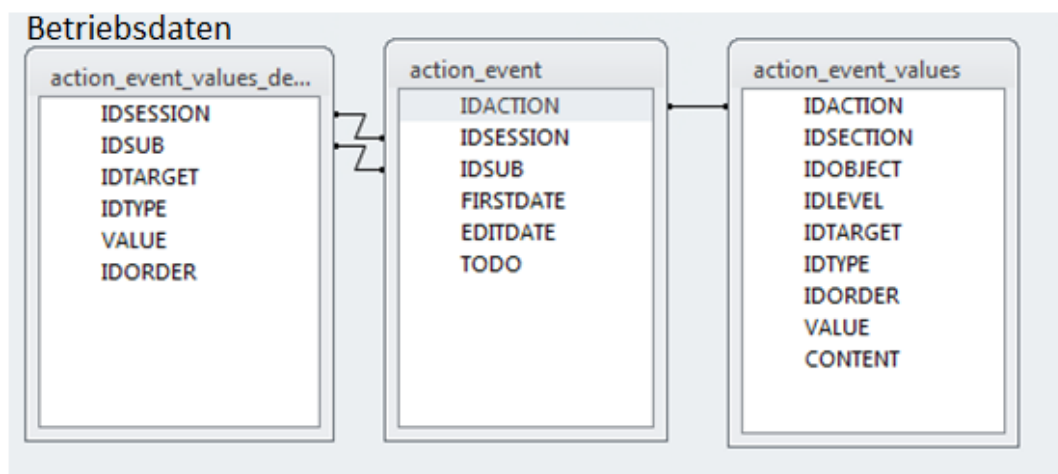


## 2.5.1.3.5 Einzelprozesse (action)



Bezeichnung	Inhalt	Beziehungen	Bemerkung
<b>action_super</b>	Wurzelknoten des hierarchischen Baumes an Parameter	1:n Beziehung zur hierarchisch tieferen Ebene	Logischer Ausgangspunkt der praktisch kaum Anwendung findet z.B. Feldarbeit
<b>action_main</b>	Hauptknoten im logischen Sprachgebrauch	1:n Beziehung zur hierarchisch tieferen Ebene	Die Eintragungen bilden kompakte Einheiten ab die sich bei der Präsentation als gemeinsame Gruppe darstellen sollte. z.B. Arbeiten im Grünland
<b>action_sub</b>	Hauptknoten im Gebrauch einzelner Sachbestandsgruppen	1:n Beziehung zur hierarchisch tieferen Ebene	Einzelne Aktivität im Sinne der landwirtschaftlichen Arbeit. z.B. Grundfutter ernten
<b>action_entity</b>	Detaildarstellung einer Aktivität mit vordefinierter Schnittstelle zu den Sachbestandsdaten und den Parametern		siehe 0
<b>action_block</b>	Bezeichnungstext für einzelne Prozessblöcke		Überschriften für die Darstellung von Einzelschritten z.B. Mähen
<b>action_parameter_preselection</b>	Eingrenzungsklasse für einzelne Parametergruppen		z.B Verkürzung der Mengeneinheiten auf sinnvolle Größen

## 2.5.1.3.6 Betriebsdaten



Bezeichnung	Inhalt	Beziehungen	Bemerkung
<b>action_event</b>	Eintragung eines am Betrieb ausgelösten Prozesses	1:n Beziehung zur hierarchisch tieferen Ebene	Eintragung im Sinne von: Heute habe ich die Schläge 1,2,3 gedüngt
<b>action_event_value</b>	Tatsächliche Eintragungen im ausgelösten Prozess		Normalisierte, mit den Originalschnittstellen verbindbare Datenstruktur
<b>action_event_value_default</b>	Datenbestand um wiederkehrende Prozesse mit der letzten Spezifikation zu befüllen.		

## 2.5.2 Anwendungslogik (Geschäftsmodell)

### 2.5.2.1 Einen einzelnen Prozess erfassen: Diesel einkaufen

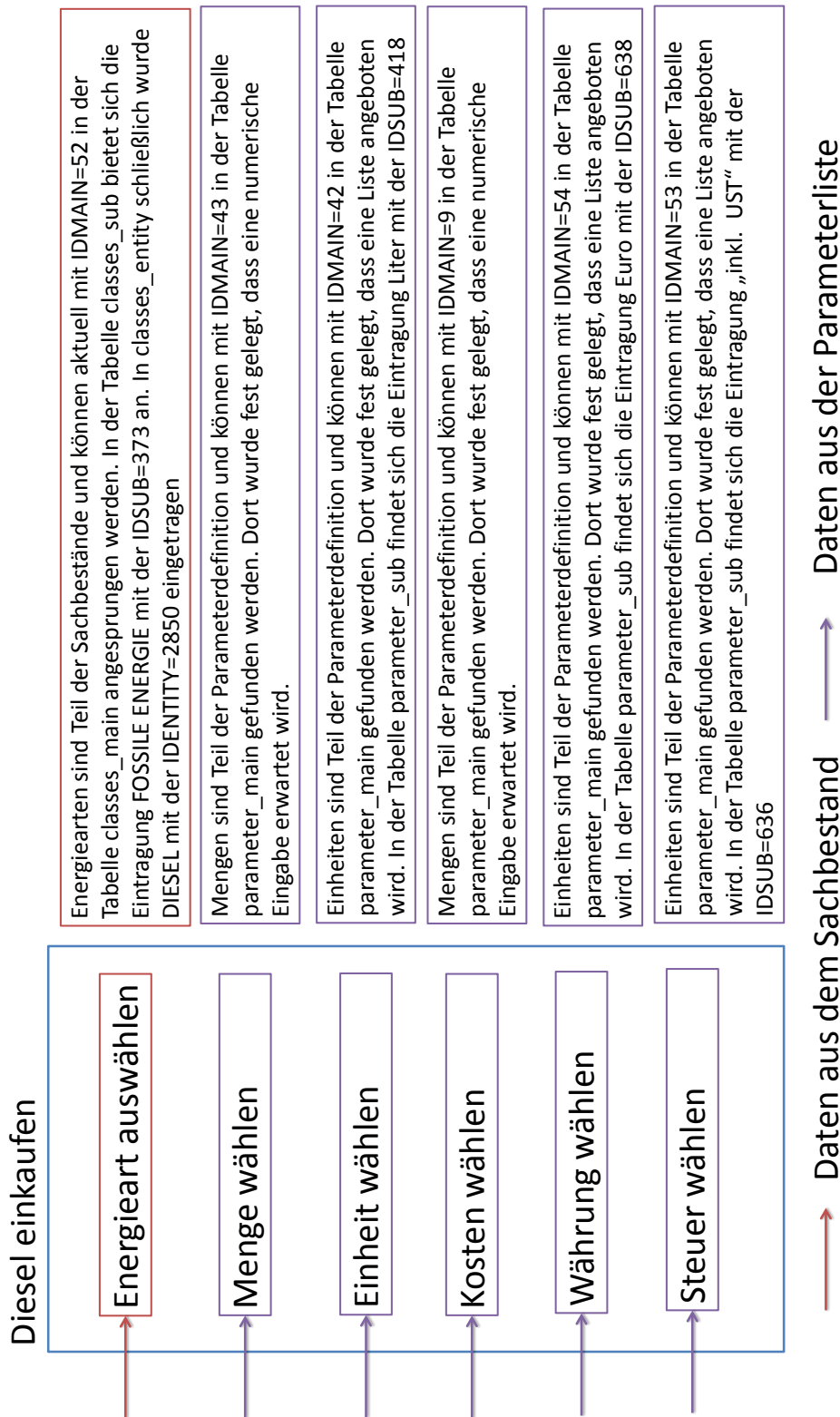


Abbildung 3: Ablauf eines Einzelprozesses

Die Erfassung einzelner Prozesse wird im Datenmodell – genauer in der Tabelle *action-entity* – vollständig abgebildet. Noch vor der Erklärung des Beispiels „Diesel einkaufen“ werden kurz die einzelnen Felder der Kerntabelle (Tabelle 1) besprochen:

**Tabelle 1: Ablauftabelle für einen Erfassungsprozess**

IDSUB	IDENTITY	IDLANGUAGE	IDORDER	IDGROUP	LINE	TYP	NODE	NODEID	MINDEPTH	COMMENT	NOTINUSE	FLAG
19	180	1	1	0	0	Parameter	Main	8	none	Datum	1	NOFLAG
19	181	1	2	1	1	Komponente	Main	52	Enties	Energiearten	0	NOFLAG
19	182	1	3	2	2	Parameter	Main	43	none	Energiemenge	0	NOFLAG
19	183	1	4	2	2	Parameter	Main	42	none	Mengeneinheit	0	NOFLAG
19	184	1	5	2	3	Parameter	Main	9	none	Kosten	0	MOUT
19	185	1	6	2	3	Parameter	Main	54	none	Währung	0	MOUT
19	186	1	7	2	3	Parameter	Main	53	none	BrutteNetto	0	MOUT

- **IDSUB, IDENTITY, IDLANGUAGE:** Bilden gemeinsam den Primärschlüssel, wobei IDSUB ein Fremdschlüssel ist, der die Tabelle *action-sub* referenziert.
- **IDORDER:** Eine einfache Ordnungsnummer innerhalb des Prozesses. Das Geschäftsmodell wird das Datenmodell hinsichtlich der Präsentationslogik immer nach dieser inneren Ordnung abarbeiten.
- **IDGROUP:** Thematisch gekoppelte Eintragungen die auf die Tabelle *action\_block* referenzieren. Jede Gruppe kann eine eigene Überschrift in der Präsentationslogik erhalten.
- **LINE:** Koppelung von Elementen innerhalb einer Gruppe. Elemente mit gleicher Zeilennummer werden in der Präsentation auch so bearbeitet.
- **TYP:** Elementartyp der Daten. Hier verzweigt die Logik die Daten in zwei Hauptdatenbäume. Die Bezeichnung „Komponente“ steht für den Sachdatenbestand (classes). Action-Elemente dieses Typs referenzieren die aktiven und passiven Inventare der Ökobilanzierung. Mit der Bezeichnung „Parameter“ wird die Menge / der Gehalt der Ökobilanzinventare gespeichert. Elemente dieses Action-Typs referenzieren den Bereich der Parameterliste (parameter).
- **NODE:** Definiert die Ebene an der in den jeweiligen Baum eingesprungen werden soll. Zur Auswahl stehen Super, Main, Sub, Entity.
- **NODEID:** ID des jeweiligen Knoten im betroffenen Baum in der definierten Ebene

- **MINDEPT:** Zeigt an wie weit der Restbaum der Daten geöffnet werden soll. Parameter haben hier keine Angabe, Sachbestände können wie die **NODE** verwendet werden.
- **COMMENT:** Beschreibender Text zu den einzelnen Zeilen eines Prozesses. Verwendet wird in der Präsentationslogik nur die Angabe zu den Parametern. Sachbestände werden immer nach der führenden Hierarchie benannt.
- **NOTINUSE:** FLAG um Elemente ein-/auszuschalten
- **FLAG:** Verarbeitungshinweis für die Präsentationslogik. Als Standardwert führt **NOFLAG** zu keiner speziellen Behandlung; **MOU** und **MIN** steuert die ökonomische Bilanzierung inklusive Farbgebung.

#### **So werden Prozesse in der Geschäftslogik abgearbeitet:**

Wenn Kunden z.B. Diesel kaufen wollen, wird Action 19 (Energie einkaufen) über das Präsentationsinterface aufgerufen. Dieser Aufruf entnimmt aus dem Datenmodell die Struktur der oben stehenden Tabelle und arbeitet diese Zeile für Zeile ab. Hier werden jene Verbindungen durch Programmierung geschaffen, die nicht direkt im Datenmodell durch Relationen erstellt werden konnten:

- Für Zeile **IDORDER 1** besteht das Geschäftsmodell in der Forderung eines Präsentationsmodells das den Datentyp Datum aufnehmen kann.
- **IDORDER 2** schafft eine Verbindung zu den Sachdaten und bietet in der Folge gestaffelte Listenelemente an.
- **IDORDER 3 bis 7** schafft Verbindungen zu den Parameterelementen. Diese stellen entweder Auswahllisten (**parameter\_main Typ=1**) oder Werte dar.

Aus diesem begrenzten Regel-Set entsteht eine Eingabeliste, die in **CAP** gespeichert und direkt an **CON** weitergeleitet werden kann. Für 5 namentliche Elemente besteht eine Ausnahme von dieser datenmodellorientierten Geschäftslogik. Diese Elemente tragen den Namen *First\_Date*, *Seconde\_Date*, *Hour*, *ToDo\_Flag*, *Message\_Box*. Diese Elemente werden in der Präsentationslogik noch weiter dargestellt.

Der Zukauf von 1.900 Liter Diesel am 01.01.2013 zu einem Preis von 2.470 Euro inkl. Mehrwertsteuer bildet sich auf eine Betrieb in Folge wie in Tabelle 2 dargestellt ab.

Tabelle 2: Ergebnistabelle für einen Erfassungsprozess

IDACTION	IDSECTION	IDOBJECT	IDLEVEL	IDTARGET	IDTYPE	IDORDER	VALUE	CONTENT
252	4	1	1	First_Date	Text	5	Startdatum	01-01-2013
252	5	181	3	Action_181_Komp_Sub_Second	Select	6	373	Fossile Energie
252	5	181	4	Action_181_Komp_Entity_Third	Select	7	2850	Diesel
252	6	182	1	Action_182_Para_Sub_First	Text	8	Wert	1900
252	6	183	3	Action_183_Para_Sub_First	Select	9	418	Liter
252	6	184	1	Action_184_Para_Sub_First	Text	10	Wert	2470
252	6	185	3	Action_185_Para_Sub_First	Select	11	638	Euro
252	6	186	3	Action_186_Para_Sub_First	Select	12	636	inkl. Mwst.

Folgende Inhalte werden gespeichert:

- IDACTION: Referenz auf das Wurzelement der aufgezeichneten Prozesse. Hier verweist die Nummer 252 auf eine Zeile in der Tabelle action\_event. Die referenzierte Zeile ist einer IDSESSION zugeordnet, womit der betroffene Betrieb und alle seine Attribute bekannt sind.
- IDSECTION: Ist eine interne Zuordnung von gespeicherten Elementen zu einer späteren Verarbeitungsstrategie für CON.

Tabelle 3: Verarbeitungstypen CON

IDSECTION	IDOBJECT
1=Radio Button Feldarbeit	1=Schläge
	2=Bewirtschaftungseinheiten
	3=Für Dritte
	4=Alm
2=Checkboxes der Bezugsflächen--> Schläge	Globale SchlagID des Schlages
3=Checkboxes der Bezugsflächen--> Bewirtschaftungseinheiten	Globale MUIID des Schlages
4=Datum	1=Startdatum
	2=Enddatum
5=Auswahlliste	Objekt aus Eventliste, 99999 = Stunden
6=Werte	
7=Checkboxes TODO/Message	1=TODO
	2=Message

- IDOBJECT: Referenziert IDENTITY in der Tabelle action\_entity. 181 steht etwa für einen Aufruf der Energiearten aus den Sachbeständen.
- IDLEVEL: Aufruftiefe des jeweiligen Objektes innerhalb seines Datenbaumes. In der Zeile 2 wird das erste freie Element der Verarbeitungsanweisung 181 dargestellt. Das erste Element, Main mit einem IDLEVEL=2, ist die allgemeine Definition von Energie, wird nicht gespeichert. In der Sub-Ausprägung, mit einem IDLEVEL=3, stehen mehrere Energiearten zur Verfügung. *Fossile Energie* mit einer IDSUB von 373 und einem IDLEVEL=4 wurde ausgewählt. In Zeile 3 steht mit der Eintragung *Diesel* und einer IDENTITY von 2850 das Element der tiefsten

Ebene. Diese Strategie der Speicherung der Geschäftslogik mit den Daten wurde für Sachbestände und Parameter gleichermaßen umgesetzt.

- IDTARGET: Kombinierte Zeicheninformation für die Präsentationsebene
- IDTYE: Zielelement in der Präsentationsebene
- IDORDER: Interne Reihung im HTML-DOM-TREE
- VALUE: Art des Wertes
- CONTENT: Eigentlicher Wert

Zusammenfassende Aspekte zu den Prozessen:

1. Die Grundstrukturen der Prozesse werden im Datenmodell von CAP – genauer in der Tabelle `action_entity` - definiert. Inhalte können sowohl in der Struktur der Sachbestände als auch im Bereich der Parameter beliebig erweitert werden.
2. Das Geschäftsmodell entnimmt die Anweisungen für Prozesse aus dem Datenmodell und erzeugt daraus die Präsentationslogik
3. CAP kapselt die Erfassung. Die Datenprüfung/-logik wird in CON ebenso gekapselt entwickelt.

### **2.5.2.2 Verwaltung von Feldstücken, Schlägen und Bewirtschaftungseinheiten**

Die Offenheit des Entwurfes wurde im Bereich der Erfassung von Feldstücken, Schlägen und der Bildung von Bewirtschaftungseinheiten deutlich eingeschränkt. Diese Maßnahme ist der Preis für die Auswahl eines bestimmten Werkzeuges (Leaflet) für die Erfassung von GIS-Polygonen und die Folge der nationalen Struktur bei der Verarbeitung der landwirtschaftlichen Flächen. Eine Anpassung in diesem Bereich kann nicht im Datenmodell erfolgen, sondern erfordert einen direkten Eingriff in den Quellcode der Verarbeitungslogik.

## 2.5.3 Präsentationslogik

Daten- und Geschäftsmodell wurden in einem Web-Client abgebildet der unter [www.farmlife.at/cap](http://www.farmlife.at/cap)

zur Verfügung steht.

### 2.5.3.1 Struktur der WEB-Site

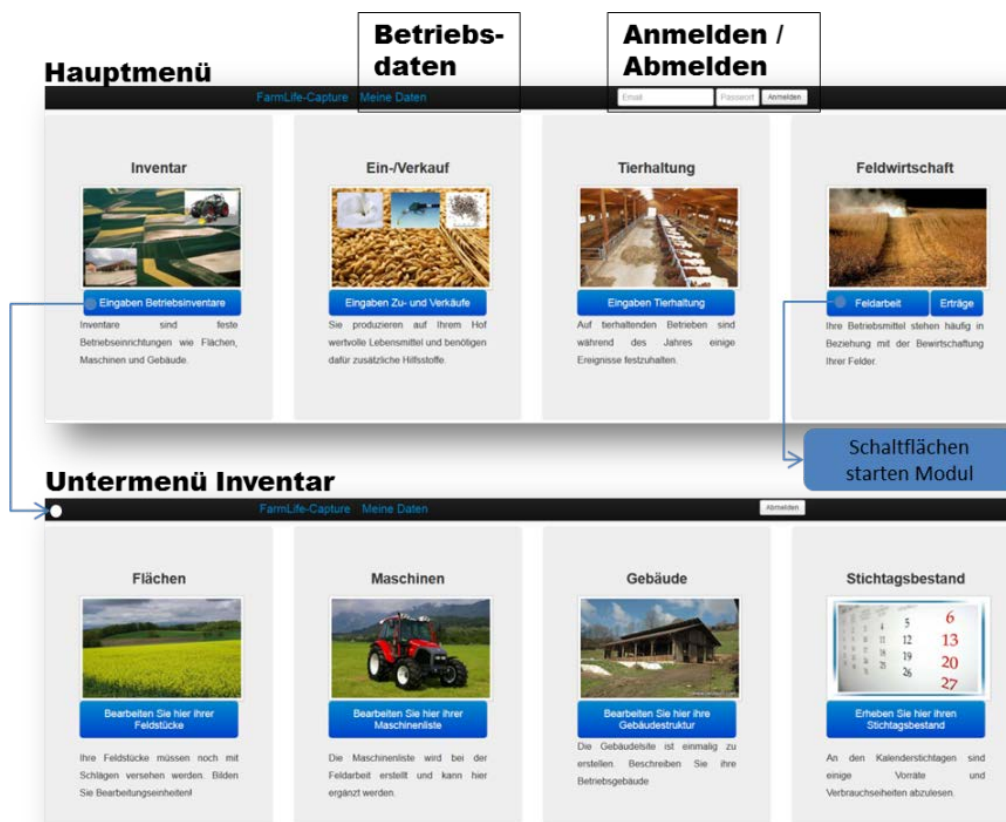


Abbildung 4: Übersicht Farmlife

#### 2.5.3.1.1 Kopfzeile

In rechten Kopfzeilenbereich kann sich jeder Betrieb mit entsprechendem Login anmelden. Logins werden nur nach schriftlicher Vereinbarung mit dem Betrieb erstellt. Diese Vereinbarung ist mit einer Zustimmung zur Nutzung bestehender Datenquell verbunden. Im Bereich der Betriebsdaten werden alle Kontaktdaten vom Betrieb verwaltet. Zusätzlich finden sich die Betriebsdaten des Beraters und eine Angabe über den strategischen Erfassungsplan des Betriebes.

Betriebe haben die Freiheit ihre Daten unterschiedlich zu erfassen. Es liegt in der Verantwortung des Projektteams letztlich alle Daten in Farmlife-Capture auch



ordnungsgemäß zu integrieren. Als Erfassungsoptionen werden folgende Möglichkeiten angeboten:

- Erfassung in FarmLife-Capture: Wunsch des Projektteams - und manchmal strategisches Geschick des Beraters - dieses Ziel zu erreichen.
- Papierform: Das zu Jahresbeginn ausgegebene Handbuch mit seiner papierernen Erfassungsform kann weiter gültig bleiben. Mitarbeiter/innen des LFZ übernehmen die Digitalisierung in FarmLife-Capture.
- Andere digitale Systeme: Manche Betriebe verfügen über Systeme die vom Gedanken her schon recht nahe an FarmLife-Capture heranreichen. Wenn diese Systeme entweder digital oder in papierform die entsprechenden Grundinformationen liefern, wird eine Digitalisierung vom LFZ angeboten.

### Meine Adresse:

**Name**

Anrede:

Nachname:  Vorname:

**Anmeldedaten**

E-Mailadresse:

Passwort:

Telefon:

**Postanschrift**

Straße:  Nummer:

Postleitzahl Ort:   (Postleitzahl ohne Länderkürzel)

### Mein Beraterkontakt:

**Name**

Anrede:

Nachname:  Vorname:

**Kontakt**

E-Mailadresse:

Telefon:

**So erfasse ich:**

Erfassungsbereich	Möglichkeiten	Info
Feldstücke und Schläge	Müssen im Menüpunkt INVENTAR -> FLÄCHEN im LINKEN Bildschirmbereich erfasst werden!	
Schlagsgrenzen	Müssen im Menüpunkt INVENTAR -> FLÄCHEN im RECHTEN Bildschirmbereich erfasst werden!	
Stichtagsinventur	Müssen im Menüpunkt INVENTAR -> STICHTAGSBESTAND erfasst werden!	
Zu-/Verkauf von Betriebsmitteln und Produkten	<input checked="" type="checkbox"/> FarmLife-Web-Client <input type="checkbox"/> Anderes digitales System <input checked="" type="checkbox"/> Papierform (Handbuch oder Sammlung)	Ich übermittel die Daten der alternativen Quellen auf Anfrage!
Feldarbeit	<input checked="" type="checkbox"/> FarmLife-Web-Client <input type="checkbox"/> Anderes digitales System <input checked="" type="checkbox"/> Papierform (Handbuch oder Sammlung)	Ich übermittel die Daten der alternativen Quellen auf Anfrage!
Ertrag Pflanzenbau	<input checked="" type="checkbox"/> FarmLife-Web-Client <input type="checkbox"/> Anderes digitales System <input checked="" type="checkbox"/> Papierform (Handbuch oder Sammlung)	Ich übermittel die Daten der alternativen Quellen auf Anfrage!
Maschinen	<input checked="" type="checkbox"/> FarmLife-Web-Client <input type="checkbox"/> Anderes digitales System <input checked="" type="checkbox"/> Papierform (Handbuch oder Sammlung)	Ich übermittel die Daten der alternativen Quellen auf Anfrage!
Gebäude	<input checked="" type="checkbox"/> FarmLife-Web-Client <input type="checkbox"/> Anderes digitales System <input checked="" type="checkbox"/> Papierform (Handbuch oder Sammlung)	Ich übermittel die Daten der alternativen Quellen auf Anfrage!

**Daten können durchaus im gemischter Form vorliegen. Denkbar sind einzelne Betriebszweige im Rahmen der Arbeitskreisberatung. Beachten Sie bitte, dass wir den gesamten Betrieb erfassen und deshalb die Daten die über die Erfassungssystem der Arbeitskreise hinausgehen zusätzlich im Farmlife-Web-Client erfasst werden müssen!**

Ich bin damit einverstanden, daß meine Daten im Rahmen des Forschungsprojektes FarmLife elektronisch verarbeitet werden! Ich bestehe darauf, dass meine Daten ausschließlich in anonymisierter Form weiterverarbeitet werden!

**Abbildung 5: Formular der Betriebsdaten**

### 2.5.3.1.2 Menü

CAP ist intern sehr dynamisch und präsentiert sich dem Benutzer im schlichten Kleid. In der Hauptebene verweisen lediglich 5 Befehlsschaltflächen auf die zu erfassenden Themengebiete. Im Gegensatz zum Bereich *Inventur* führen die *Ein-/Verkäufe*, *Tierhaltung*, *Feldarbeit* sowie *Erträge* direkt zur Erfassung. Inventurdaten können in einem eigenen Untermenü getrennt als *Fläche*, *Maschinen*, *Gebäude* bzw. *Stichtagsinventur* erfasst werden.

- **Ein-/Verkäufe:** Betriebe kaufen häufig variable Betriebsmittel unterschiedlichster Natur zu. Diese Betriebsmittel werden hier sowohl sachlich (siehe Beispiel Dieseleinkauf), als auch nach ihrer Menge bzw. ihren Kosten erfasst.
- **Tierhaltung:** Derzeit findet sich in diesem Bereich nur ein spezieller Prozess, der die Zuteilung von Kraftfutter aus eigenen Mahl- und Mischanlagen betrifft.
- **Feldarbeit:** Schlagbezogene Erfassung aller Arbeitsgänge auf einem Betrieb mit den dazu notwendigen Maschinen und Betriebsmitteln.
- **Erträge:** Erfassung der jährlichen Ertragslage aus den pflanzenbaulichen Kulturen über alle Schläge hinweg.
- **Flächen:** Jährliche Definition der Betriebsflächen in der bereits beschriebenen Struktur von Feldstücken, Schlägen und Bewirtschaftungseinheiten inklusive einer Digitalisierung auf der Bilddatengrundlage von BING.
- **Maschinen:** Inventarliste von Maschinen.

- Gebäude: Strukturierte Inventarliste von Gebäuden.
- Stichtagsbestand: Inventur von ausgewählten variablen Produkten oder Betriebsmitteln an einem Stichtag. Der Zweck ist die Berücksichtigung von Stoffmengen, die über die Jahresgrenze hinweg Zu- oder Abschläge erzwingen.

### 2.5.3.2 Struktur eines Prozesses

Alle Erfassungsprozesse werden in ihrer Bildschirmpräsentation in drei Bereiche zerteilt. Diese sind:

1. Auslösen von handlungsorientierte Prozessen im Sinne von „ich habe ...“ finden sich immer im oberen Bereich der Bildschirmdarstellung. Im Beispiel von Ein-/Vekäufen steht eine Palette von orangen Schaltflächen für den Einkauf und einige grüne Schaltflächen für den Verkauf bereit.
2. Filtern von Eingaben aus der Liste: Zwischen der Eingabenliste und den Schaltflächen findet sich eine Zeile, die als Filter für die (möglicherweise sehr umfangreiche) Erhebungsliste im unteren Bildschirmbereich verwendet wird.
3. Die Erhebungsliste: Mit den Icons am linken Bildschirmrand können sie einzelne Eintragungen bearbeiten oder löschen. Eintragungen mit einem aktiven TO-DO-Status erhalten hier noch ein Warnsymbol.



Abbildung 6: Präsentation von Eingabeprozessen

Wenn ein neuer Prozess – etwa der Einkauf von Energie – gestartet wird, legt sich über den Grundbildschirm ein eigenes Fenster mit der unter Punkt 2.5.2.1 beschriebenen Geschäftslogik. Das Fenster besteht aus zwei bzw. bei der Erfassung der Feldarbeit, drei Teilen. Der erste Teil beschreibt immer den Prozess, wobei mit einem führenden Datum begonnen wird. Bei einem Handelsprozess besteht das

Datum nur aus einem Zeitpunkt bei der Feldbearbeitung und wird der Zeitpunkt mit einem zweiten Datum auf einen Zeitraum ausgeweitet. Innerhalb des Prozesses bilden sich die einzelnen Elemente nach den Vorgaben des Datenmodells und unter Verwendung der Geschäftslogik ab.

### Arbeitsschritt: Energie

#### 1. Beschreiben Sie den Handelsvorgang

Datum

Energieart auswählen

Energie

Energiemenge/-kosten definieren

Energiemenge

Kosten

#### 2. Nachrichten und Status

Eingabe noch nicht fertig

Tragen Sie hier ihre Nachricht/Memo /Kommentar ein!

Tragen Sie hier ihre Nachricht ein!

**Abbildung 7: Erfassungsfenster des Prozesses "Energie einkaufen"**

Jedes Erfassungsfenster bietet die Option eines Bereiches in dem ein TO-DO-Status bzw. eine Nachricht hinterlassen werden kann. Eine Nicht-Erfassung ist so unmöglich, denn es kann zumindest eine Nachricht hinterlassen werden.

#### 1. Wählen Sie die Flächen aus

Auswahl nach Schlägen    
  Auswahl nach Bewirtschaftungseinheiten    
  Arbeiten für Dritte    
  Alm

Feldstück	Schläge
BODEN	<input type="checkbox"/> S 1 / DAUERWE
LÄRCHENACKER	<input type="checkbox"/> S 1 / WINTERR

**Abbildung 8: Auswahl von betroffenen Flächen**

Bei der Erfassung von Feldarbeitsprozessen wird die beschriebene Struktur durch die Angabe der betroffenen Schläge ergänzt. Zur Auswahl stehen variable Kombinationen einzelner Schläge oder Bewirtschaftungseinheiten sowie die Möglichkeit von Aktivitäten auf der eigenen Alm bzw. von Aktivitäten für Dritte. Bei der Auswahl einer Aktivität für Dritte ergänzt sich das Eingabefeld um eine Angabe in Arbeitsstunden.

Die letzte Erfassungsstruktur eines Prozesses wird immer für die nächste neue Anlage verwendet. Diese Maßnahme reduziert den Erfassungsaufwand auf ein mögliches Minimum, lässt aber immer noch eine maximale Veränderung zu.

### 2.5.3.3 Struktur der Erfassung von Feldstücken, Schlägen und Bewirtschaftungseinheiten

Fz-Nr	Name	Event	S-Nr	Hauptnutzung	Fläche	Events
Betriebsnummer: 106121456						
6	STUTTERING		1	KÖRNERMAIS	0.246	
31	ENNS ACKER 3		1	MÄHWEISE+WEIDE ZWEI NUTZUNGEN	1.1577	
32	Ennsacker aussen		1	HUTWEIDE	0.4273	
33	HÖFERWEISE kontrolle		Noch keine Schläge			
34	ENNSWEISE kontrolle		Noch keine Schläge			
35	GLASERER WEISE kontrolle		Noch keine Schläge			
36	TIERGARTEN kontrolle		Noch keine Schläge			

Abbildung 9: Erfassung von Feldstücken

Im Untermenü *Inventar* startet die Schaltfläche *Flächen* die Erfassung der räumlichen Nutzungseinheiten. Noch einmal kurz zur Orientierung:

In Österreich werden Flächen sehr genau im Rahmen des Förderungsprogrammes der Landwirtschaft erfasst. Im Rahmen dieser Erfassung wurde der Begriff des *Feldstückes* als ein Synonym für ein räumlich abgegrenztes Flurstück eingeführt. Diese räumliche Einheit ist unabhängig von den Grundstücksgrenzen der Katastralmappe. Innerhalb von Feldstücken werden homogene Pflanzengemeinschaften als *Schlag* bezeichnet. Ein Feldstück kann einen oder mehrere Schläge haben. Da betriebliche Informationen über die eher statischen Feldstücke (Namen, Nummer) und deren Umrandpolygon vorliegen, werden diese Informationen aus dem INVEKOS bei der Erstellung der Betriebe eingespielt. Das ist aber nicht zwingend notwendig.

### 2.5.3.3.1 Feldstücke und Schläge erfassen

Feldstücke müssen in CAP erst mit ihrem Namen und einer Feldstücknummer bekannt gegeben werden. Dies kann auf der linken Teilseite des Bildschirms durch das +Icon unterhalb der Feldstücküberschrift ausgelöst werden. Bestehende Feldstücke können wie alle Elemente mit dem Editier-Icon oder dem Lösch-Icon weiterbearbeitet werden. Schläge werden in CAP durch das +Icon innerhalb eines Feldstückes (rechte Seite, linker Teil) neu angelegt. Feldstücke beinhalten folgende Attribute:

#### 1. Grunddaten:

- **Schlagnummer:** Die Grunddaten werden durch eine laufende Schlagnummer angeführt. Diese Schlagnummer wird automatisch vom System nach oben gezählt.
- **Schlagfläche:** Die Schlagfläche kann hier notiert werden. In der Regel wird die Schlagfläche aber erst nach dem Zeichnen des Polygons feststehen und dann automatisch hier eingetragen.
- **Fruchtfolge:** Der Begriff der Fruchtfolge beschreibt das langjährige Verfahren zur Bewirtschaftung des Feldstückes und betrifft dort nur die Hauptfrüchte.

#### 2. Kulturen:

- **Anzahl:** Innerhalb eines Jahres können bis zu 5 Kulturen definiert werden.
- **Hauptklasse:** Ist eine grobe Einteilung der möglichen Schlagnutzungsarten in Dauergrünland, Ackerfrüchte, Dauerkulturen, Gemüseanbau, Almen
- **Schlagnutzung:** Ist die eigentliche Kultur
- **Nutzungstyp:** Zeigt den Status (Hauptfrucht/Zwischenfrucht) der Schlagnutzung innerhalb der Fruchtfolge des Erfassungsjahres.

#### Neuer Schlag für Feldstück: 3087

##### Grunddaten

Schlagnummer:

Fläche (ha):

Fruchtfolge:

##### Kulturen im Erfassungsjahr

Parameter	1. Kultur	2. Kultur	3. Kultur	4. Kultur	5. Kultur
Hauptklasse	<input type="text" value="Wähle Hauptkulturart"/>	<input type="text" value="Wähle Hauptkulturart"/>	<input type="text" value="Wähle Hauptkulturart"/>	<input type="text" value="Wähle Hauptkulturart"/>	<input type="text" value="Wähle Hauptkulturart"/>
Schlagnutzung					
Nutzungstyp					

Abbildung 10: Grunddefinition Schlag

### 3. Zusätzliche Informationen:

- **Ertragslage:** Definiert die Fruchtbarkeit des Schlages innerhalb der vergleichbaren Schlagnutzungen eines Betriebes. Beispiel: Ein Betrieb nutzt 10 Schläge als Grünland mit 3 Aufwüchsen. Ein Schlag liegt in nordexponierter Hanglage und hat immer etwas weniger an Ertrag. Dieses „weniger“ kann hier verzeichnet werden.
- **Drainagen am Schlag:** Zeigt mit einem Flag eventuelle Drainagen an.
- **Quelle am Schlag:** Zeigt einen Hangwasseraustritt an.
- **Fließt Oberflächenwasser zu:** Zeigt mögliche partielle oder permanente Wasserzufuhr an.

#### Informationen zum Schlag


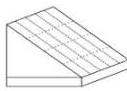
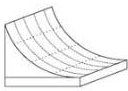

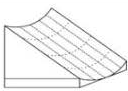
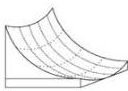
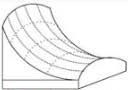
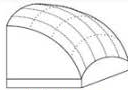
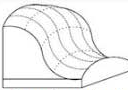
Ertragslage	Drainagen am Schlag?	Quelle am Schlag?	Fließt Oberflächenwasser zu?
<input type="radio"/> 30 % geringer <input type="radio"/> 20 % geringer <input type="radio"/> 10 % geringer <input checked="" type="radio"/> Durchschnittlich <input type="radio"/> 10 % mehr <input type="radio"/> 20 % mehr <input type="radio"/> 30 % mehr	<input checked="" type="radio"/> Nein <input type="radio"/> Ja	<input checked="" type="radio"/> Nein <input type="radio"/> Ja	<input checked="" type="radio"/> Nein <input type="radio"/> Ja

**Abbildung 11: Zusätzliche Informationen zum Schlag**

### 4. Hangneigung

Definiert die räumliche Ausformung eines Schlages. Die tatsächliche Neigung (und andere Lageparameter) wird im Post-Prozessing der Daten an der Master-Control-Station aus dem digitalen Geländemodell entnommen.

#### Informationen zur Hangneigung

Quelle: PRASUHN & GRÜNIG, 2001					
 <input type="radio"/> Keine Hangneigung					
 <input type="radio"/> allseits gestreckt	 <input type="radio"/> längs konkav - quer gestreckt	 <input type="radio"/> längs konvex - quer gestreckt	<input checked="" type="radio"/> längs konvex/konkav - quer gestreckt	 <input type="radio"/> längs gestreckt - quer konkav	 <input type="radio"/> allseits konkav
 <input type="radio"/> längs konvex - quer konkav	 <input type="radio"/> längs konvex/konkav - quer konkav	 <input type="radio"/> längs gestreckt - quer konvex	 <input type="radio"/> längs konkav - quer konvex	 <input type="radio"/> allseits konvex	 <input type="radio"/> längs konvex/konkav - quer konvex

**Abbildung 12: Räumliche Strukturen nach Prasuhn & Grünig**

### 2.5.3.3.2 Bewirtschaftungseinheiten definieren

Im rechten, oberen Bildschirmbereich bietet CAP die Option „Bewirtschaftungseinheiten bilden“ an. Bewirtschaftungseinheiten werden wie die Grundstrukturen verwaltet. Zuerst wird ein Rahmen für die Bewirtschaftungseinheit gebildet. Dieser Rahmen kann in der rechten Bildschirmhälfte durch das +Icon unter der Überschrift „Bewirtschaftungseinheit“ aktiviert werden. Eine neue Bewirtschaftungseinheit trägt eine aufsteigende Nummer und bietet als einzige Option die Vorauswahl eines Schlagnutzungstyps an. Dieser steuert bei der Zuordnung der Schläge die vorgeschlagene Liste. Alternativ kann auch ohne diese Zuordnung gearbeitet werden.

#### Neue Bewirtschaftungseinheit

##### Grunddaten

Einheit:

##### Pflanzenbauliche Nutzung

- Keine bestimmte Nutzung  
 Auswahl aus der Liste

Parameter	Nutzung
Hauptklasse	Wähle Hauptkulturart <input type="text"/>
Schlagnutzung	

**Abbildung 13: Bewirtschaftungseinheit anlegen**

Nach der Definition des Rahmens können im Bereich Bewirtschaftungseinheiten / Schläge entsprechende Eintragungen getätigt werden.

#### Auswahl der Schläge

##### Geeignete Schläge

Verfügbare Schläge			
Feldstück	Schlag	Nutzung	Auswahl
STALLFELD kontrolle	1	MÄHWIESE/-WEIDE DREI UND MEHR NUTZUNGEN	<input type="checkbox"/>
FASANFELD kontrolle	1	MÄHWIESE/-WEIDE DREI UND MEHR NUTZUNGEN	<input type="checkbox"/>

**Abbildung 14: Schläge zur Bewirtschaftungseinheit hinzufügen**



### 2.5.3.3.3 Schläge im GIS zeichnen

Da zur Orientierung der Erfassung von Schlägen bereits die Feldstückpolygone eingetragen wurden, bietet das System zwei alternative Wege zur Erfassung an.

1. Wenn am Feldstück ohnehin nur ein Schlag existiert, kann das Feldstückpolygon als Schlag übernommen werden. Dazu wählt der Anwender zuerst aus der Schlagliste – mit 1 gekennzeichnet – einen Schlag aus und ordnet ein Feldstück aus der Liste 2 zu. In Folge wird das Feldstück vollständig als Schlag übernommen. Dies ist an der blauen Farbe und dem Tag ersichtlich.

2. Wenn am Feldstück mehrere Schläge vorhanden sind, müssen diese auch gezeichnet werden. Dazu wählt der Anwender wieder zuerst einen Schlag aus der mit 1 gekennzeichneten Liste aus. Dann ist allerdings mit dem 5-Eck-Symbol ein manueller Digitalisierungsprozess auf der Grundlage des Luftbildes zu starten. Dazu wird mit einzelnen Klicks eine Linie um das Feldstück gezeichnet, die mit einem Klick auf den ersten Punkt abgeschlossen werden kann. Inselfpolygone können nicht erfasst werden und sind an der Master-Control-Station nachzuführen.

1. Wählen Sie einen Schlag aus! ⓘ

Noch zu zeichnender Schlag:

2. Entweder übernehmen Sie die Feldstückgrenzen oder ... ⓘ

Ganzes Feldstück übernehmen:

3. ... zeichnen Sie frei in der Karte (Start=5-Eck) ⓘ

**Abbildung 15: Optionen zum Digitalisieren von Schlägen**



Während des Digitalisierens

Nach dem Digitalisieren



Veränderungen durchführen

**Abbildung 16: Status eines Schlags während der Bearbeitung**

Bestehende Schlagpolygone können mit einem Doppel-Klick zur Bearbeitung aktiviert werden. Die einzelnen Knoten des Polygons werden sichtbar und können verschoben werden. Der Bearbeitungsvorgang wird wieder mit einem Doppel-Klick abgeschlossen. Zusätzlich erscheint im Textfeld neben dem Button „Schlag löschen“ die Schlagnummer. Jetzt kann der Schlag mit dem roten Button auch gelöscht werden.

## 2.5.4 Strukturen des Softwareprojektes

### 2.5.4.1 Übersicht

Im Rahmen der Implementierung von CAP wurden folgende Strukturen angelegt, die auch in die Präsentationslogik eingreift:

- **Index.php** und **php-Dateien** im **Root-Verzeichnis**: Das Projekt CAP startet über die Datei `index.php` als primäres Dokument des Datenablaufes. Mit Ausnahme der Dateien `register.php`, `confirm.php` und `getPWD.php` verweist jede Datei auf ein thematisches Erfassungsfeld von CAP.
- **Backend**: Verzeichnis aller Funktionen für den Datenaustausch zwischen Datenbank und Client via AJAX.
- **Components**: Templates für die Oberflächengestaltung von CAP. Jeder `php-Datei` im **Root-Verzeichnis** ist hier ein grafischer Entwurf für die Oberfläche hinterlegt.
- **CSS**: Darstellungsvorschriften für die grafische Oberfläche in der Form von Cascading Stylesheets. Die meisten Dateien gehören zum Framework Bootstrap. Die Datei `farmlife.css` ist die tatsächliche Master-Datei für lokale Anpassungen.
- **Doc**: Ziel für alle Begleitdokumente von CAP
- **Img**: Bilderverzeichnis für die Darstellung
- **JS**: Verzeichnis der Java-Script Bibliotheken die beim Client zur Ausführung gebracht werden können. Es existiert wieder eine Reihe von Bootstrap-Dateien die zum Framework Bootstrap gehören. Für den Datenaustausch via AJAX stehen mehrere JQuery-Dateien zur Verfügung. `Canvas-to-blob.min.js`, `load-image.min.js`, `pagin.js`, `pinOpanoStart.js`, `proj4sjs-combine.js` und `templ.in.js` sind verschiedene Arbeitsdateien für einfache, kleine Aufgaben. Die lokale Funktionalität der Geschäftslogik von CAP steckt in folgenden Dateien:
  - `farmLifeStart.js`: Funktionen zum Starten via `php`
  - `farmLifeVariablen.js`: Objekte für die lokale Datenhaltung verschiedener lokaler und globaler Variablen am Client.
  - `Feld.js`: Alle Funktionen für die dynamische Datenerfassung
  - `Flachen.js`: Alle Funktionen für die Verwaltung der räumlichen Daten
  - `Init.js`: Graphikfunktionen
  - `Login.js`: Prüfung der Anmeldung an CAP
  - `myFarmLife.js`: Betriebsdatenerfassung

backend	23.09.2013 16:47	Dateiordner	
components	23.09.2013 16:47	Dateiordner	
css	23.09.2013 16:47	Dateiordner	
doc	13.05.2013 15:46	Dateiordner	
img	23.09.2013 16:47	Dateiordner	
js	23.09.2013 16:47	Dateiordner	
Leaflet.draw-master	23.09.2013 16:47	Dateiordner	
Leaflet.label-master	23.09.2013 16:47	Dateiordner	
leaflet-0.6.2	23.09.2013 16:47	Dateiordner	
leaflet-plugins-master	23.09.2013 16:47	Dateiordner	
proj4js-master	23.09.2013 16:47	Dateiordner	
tinymce	28.08.2013 12:58	Dateiordner	
confirm.php	16.05.2013 16:26	PHP-Datei	3 KB
erfolg.php	13.05.2013 16:21	PHP-Datei	2 KB
ertrag.php	23.07.2013 16:02	PHP-Datei	3 KB
feld.php	23.07.2013 16:02	PHP-Datei	3 KB
flachen.php	23.07.2013 16:01	PHP-Datei	3 KB
getPWD.php	13.05.2013 16:21	PHP-Datei	3 KB
index.php	30.07.2013 12:00	PHP-Datei	4 KB
inventur.php	23.07.2013 16:02	PHP-Datei	3 KB
maschinen.php	23.07.2013 16:02	PHP-Datei	3 KB
myData.php	30.07.2013 14:23	PHP-Datei	2 KB
office.php	23.07.2013 16:02	PHP-Datei	3 KB
register.php	13.05.2013 16:20	PHP-Datei	6 KB
restorePWD.php	26.07.2013 11:54	PHP-Datei	5 KB
stall.php	23.07.2013 16:03	PHP-Datei	3 KB

Abbildung 17: Dateistruktur CAP

### 2.5.4.2 Kernfunktionen in den JS-Dateien

```

// Statusprüfung Login
function noLogin(){
// Startsequenz
function loadFeld(target){
// Aufbau der Listen der Eintragungen in Abhängigkeit vom Filter
function drawAllActionLine(){
// Grundbelegung der Variablen
function initFeld(){
// Aufbau der Funktionen für die Erfassung eines Arbeitsschrittes
function new_action(actionid){
// Aufbau der Funktionen zum Bearbeiten von Arbeitsschritten
function edit_Action(localActionid,obj){
// Funktion zum Laden der einzelnen Feldinhalte in der Kopfzeile
function load_action_edit_feld(localActionid){
// Funktion zum Laden von Feldinhalten
function load_action_edit(localActionid){
// Grund-Funktion zum Speichern der Felder
function save_fieldAction(){
// Grund-Funktion zum Löschen der Felder
function delete_Action(idAction,r){
// Funktion zum Löschen einzelner Werte
function deleteActionValue(idAction){
// Funktion zum Löschen der Default Belegung von Werten
function deleteActionValueDefault(idAction){
// Funktion zum Speicher der Inhalte im Detail
function saveFieldAction Values(){
// Funktion zum Anzeigen der letzten Erfassung am Bildschirm
function addNewFieldActionToGrid(){
// Funktion zur Auswahl des Flächenbezuges
function changeActionUnit(unit){
// Teilsequenz zum Aufbau neuer Aktionen
function new_action_start(){
// Teilsequenz zum Aufbau neuer Aktionen
function new_action_construct(actionid){
// Teilsequenz zur Auswahl neuer Aktionen
function new_action_select(actionbox){
// Laden der zweiten Ebene der Sachdaten
function loadActionSecond(control){
// Laden der dritten Ebene der Sachdaten
function loadActionThird(control){
// Laden der Standardwerte der Sachdaten
function new_action_default(idAction){
// Gewähltes Elemente in Liste aktivieren
function selectOption(obj, index){

```

```

function rescaleFeld(v1) {
// Felder zurücksetzen
function clearBasicItems(v1) {
// Komma internationalisieren
function komma(inText){
// Zeitanzeige im Filter
function set_timefilter(filter, inButton){
// Aktionanzeige im Filter
function set_actionfilter(control){
// Allgemeine Zeilen für Query
$(window).bind("resize", rescaleFeld);
// Hilfe anzeigen
function Info(source){
// Browser prüfen
function checkBrowserName(name) {

```

Abbildung 18: JS-Funktionskern der Datenerfassungsteile (feld.js)

```

function loadFlachen(){
// Initialisieren der graphischen Darstellung
function initFlachen(){
// Neues Feldstück anlegen
function mui new(caller){
// ManagementUnit löschen
function mui delete(mui,r){
// ManagementUnit speichern
function save mui(){
// ManagementUnit editieren
function mui edit(mui,r){
// Neuen Schlag anlegen
function mui parcel new(mui,cropclass,r){
// Speichern der Schläge einer Bewirtschaftungseinheit
function save mui parcel(){
// Löschen der Schläge einer Bewirtschaftungseinheit
function mui parcel delete(mui,parc,r){
// Neues Feldstück anlegen
function plot new(caller){
// Anzeigen der Feldstücke
function plot show(myImg,r){
// Bearbeiten der Feldstücke
function plot edit(plot,r){
// Funktion zum Speicher neuer Schläge
function save plot(){
// Felstück löschen
function plot delete(plot,r){
// Neuen Schlag erstellen
function parcel new(plot,caller){
// Schlag bearbeiten
function parcel edit(plot,parcel,r){
// Funktion zum Speicher neuer Schläge
function save parcel(){
// Vorbereiten der Liste von Schlägen die noch mit Leaflet gezeichnet werden können
function parcel leaflet new list(){
// Vorbereiten der Liste die noch für Schläge in Leaflet übernommen werden können
function plot leaflet new list(){
// Schlag löschnen
function parcel delete(parcel,r){
// Schlagnummer global speichern
function setParcelnumberToVariable(control){
// Verbindung zwischen Feldstück und Schlag herstellen
function setPlotToParcel(control){
// Graphische Objekte neu zeichnen

```

```

// Graphische Objekte neu zeichnen
function rescaleFlachen(v1){
$(window).bind("resize", rescaleFlachen);
// Schlagnutzungsstruktur laden
function loadSchlagnutzung(control,n){
// Schlagnutzung für Bewirtschaftungseinheit laden
function loadSchlagnutzungMUI(control){
// Tabellenstruktur für Kulturarten aufbauen
function createTableSchlaege(sId, sIdLocal,sPlot, sNutzung, sFlache) {
// Leaflet initialisieren
function leaflet init(){
// Schläge in Leaflet zeichnen
function leaflet parcel(){
// Labels für die Schläge zeichnen
function leaflet label(){
// Feldstückpolygone laden
function leaflet background(){
// Auf die Betriebsschläge zoomen
function zoomToFeature(){
// Schlag in Leaflet löschen
function leaflet delete(){
// Koordinaten in WKT WGS84 transformieren
function toWKT(layer){
// Koordinaten in das Landeskoordinatensystem transformieren
function toWKT DEST(layer) {
// Koordinaten in WKT WGS84 transformieren
function toWKT I(layer) {
// Koordinaten in das Landeskoordinatensystem transformieren
function toWKT I DEST(layer) {
// Koordinaten in das Landeskoordinatensystem transformieren
function toWKT II DEST(lng,lat) {
function change fs manage(type){
// Komma internationalisieren
function komma(inText){
// Hilfe aufbauen
function Info(source){

```

Abbildung 19: JS-Funktionskern der räumlichen Erfassung (flachen.js)

### 2.5.4.3 Kernfunktionen in den php-Dateien

Die php-Dateien bilden die Brücke zwischen der am Client laufenden Java-Script-Funktion und den tatsächlichen Daten in der MYSQL-Datenbank. Diese Art der Interaktion ist sicher, da der Client keinen direkten Zugriff auf die Datenbank erhält, sondern immer auf die programmierte Schnittstelle angewiesen ist. Die entwickelten Schnittstellen können in vier verschiedene Arten unterteilt werden:

1. Abfrageschnittstellen: Tragen führend im Dateinamen den Präfix **get**. Die Abfrage, die mit einer dieser Schnittstellen aktiviert wird, geht zumeist sinngemäß aus dem Namen der Schnittstelle hervor. Diese Schnittstellen liefern in der Regel ein Array an die aufrufende Funktion zurück, welches den Status der Abfrage und den erzeugten Inhalte als strukturierten Datentyp enthält. Bei komplexen Abfragen über mehrere Datenschichten kann hier auch bereits der fertige HTML-Text zurückgegeben werden. Nebenstehendes Beispiel liefert aus der Datenbank die Inhalte zurück, die für die Darstellung der Hilfe nach einem Klick auf einen I-Button ausgelöst wurde. Die aufrufende Funktion liefert für die Abfrage einen Parameter in der Variable *INFONAME*. Mit diesem Parameter wird ein Query auf die Datentabelle *INFO* gefeuert. Das Ergebnis, ein strukturierter Datensatz wird in das Array *\$response* geschrieben und an die aufrufende Funktion mit *echo* zurückgegeben.

```

// Rückgabearray definieren
$response = array();
// Anfrageparameter übernehmen
$data = $_POST;
if (isset($data['INFONAME'])) {
    // Query definieren
    $query = "SELECT * FROM INFO WHERE INFONAME='". $data['INFONAME']. "'";

    // Query auf die Datenbank absetzen und Ergebnis auffangen
    try{
        $DBHandle = new DB_Connect('db');
        $result = $DBHandle->simpleQuery($query);
        $DBHandle->close();
    }
    catch(DB_Exception $e){
        // Datenbankfehler aufnehmen
        $result = $e;
    }
    catch(Exception $e){
        // Allgemeine Fehler aufnehmen
        $result = $e;
    }
}
// Ergebnisarray zusammenstellen
$response['data'] = $result[0];
$response['status'] = 'Erfolg!';

// Ergebnis an den Aufrufer zurückliefern
echo json_encode($response);
session_write_close();

```

2. Schnittstellen zum Löschen von Datensätzen tragen den Präfix **delete**. Sie arbeiten sinngemäß wie „get“ Schnittstellen, allerdings werden als Query keine *Select* sondern *Delete* Statements abgesetzt. Als Rückgabewert wird nur ein Status geliefert.

3. Schnittstellen zum Speichern von Daten tragen den Präfix **save**. Diese Schnittstellen werden mit allen Felddaten aus der CAP-Oberfläche aufgerufen und schreiben diese Daten mit einem Insert Statement in die Datenbank. Das gezeigte Beispiel fügt die Daten - die in der Oberfläche der Bewirtschaftungseinheiten gewonnen werden - in die Datentabelle GIS\_MUI ein. Vor dem eigentlichen Insert Statement wird noch eine Löschroutine ausgeführt.

```

if(isset($data['IDMUI']) == TRUE){
    // First step: Delete parcel
    $insert = "DELETE FROM GIS_MUI WHERE IDMUI = " . $data['IDMUI'];
    try{
        $DBHandle = new DB_Connect('db');
        $result = $DBHandle->insert($insert);
        $DBHandle->close();
        $response['result'] = $result;
    }
    catch(DB_Exception $e){
        $error['msg1'] = $insert;
        $error['status1'] = "DB_Exception";
    }
    catch(Exception $e){
        $error['msg1'] = $insert;
        $error['status1'] = "Exception";
    }
}

// Second step: Insert content
$insert2 = "INSERT INTO GIS_MUI (IDMUI, IDSESSION, IDMUILocal, CROPClass, CROPDDEFINITION, EVENTStall, EVENTSGRASSLAND, EVENTSCROPLAND, EVENTSFOUIT) VALUES ("
    . $data['IDMUI'].", "
    . $_SESSION['pansettings']['sessionid'].", "
    . $data['IDMUILocal'].", "
    . $data['CROPClass'].", "
    . $data['CROPDDEFINITION'].", "
    . $data['EVENTStall'].", "
    . $data['EVENTSGRASSLAND'].", "
    . $data['EVENTSCROPLAND'].", "
    . $data['EVENTSFOUT'].")";
try{
    $DBHandle = new DB_Connect('db');
    $result2 = $DBHandle->insert($insert2);
    $DBHandle->close();
    $response['result2'] = $result2;
}
catch(DB_Exception $e){
    $error['msg2'] = $insert;
    $error['status2'] = "DB_Exception";
}
catch(Exception $e){
    $error['msg2'] = $insert;
    $error['status2'] = "Exception";
}
$response['error'] = $error;
}

if(isset($_SESSION['pansettings']['formError']) == "KRIME ID GEBTITZE")
    $_SESSION['pansettings']['form'] = "Keine ID";
$response['error'] = "KRIME ID GEBTITZE";
session_write_close();
}
echo json_encode($error);

```

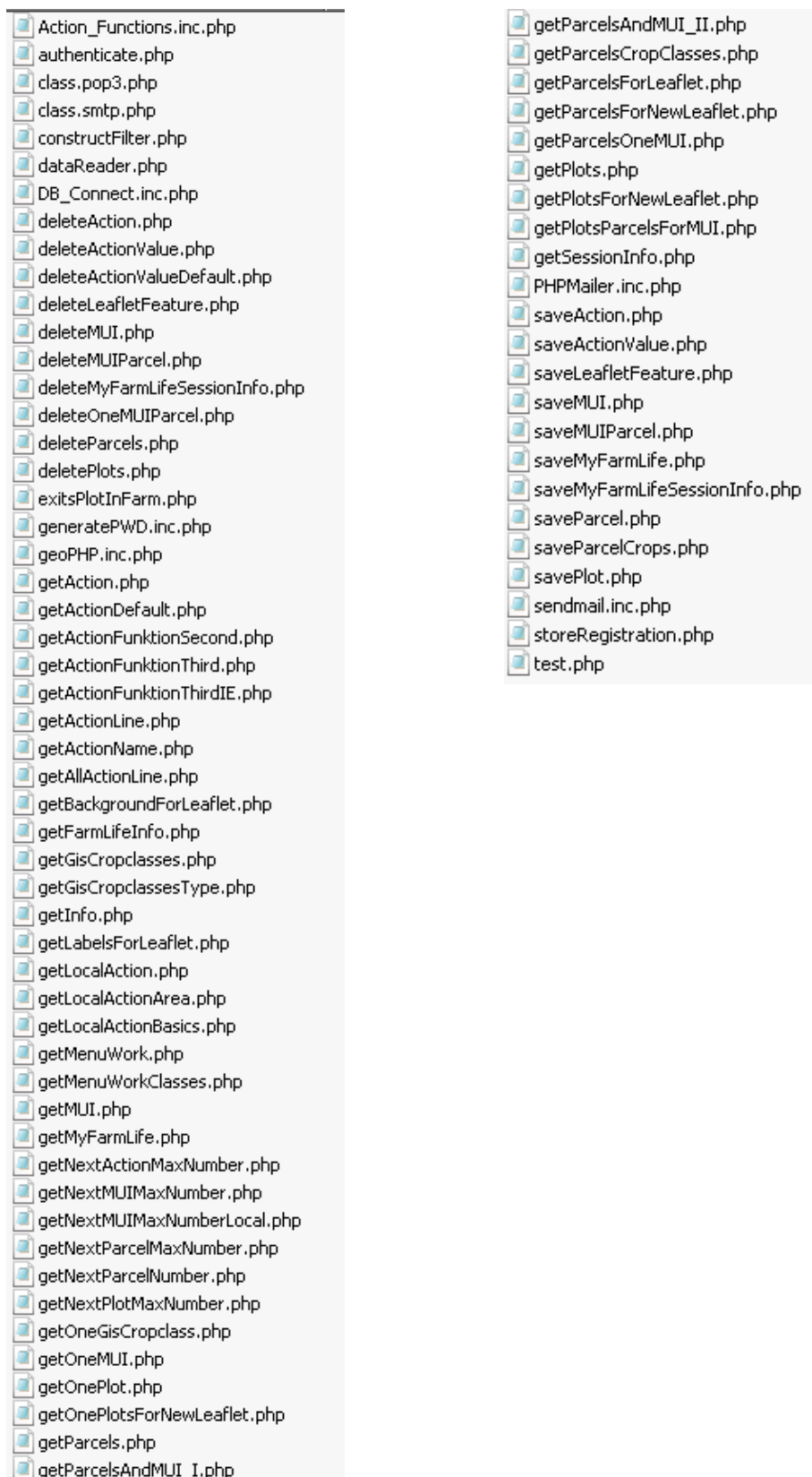


Abbildung 20: Übersicht über alle php-Dateien



### 2.5.4.4 Fallstudie eines Datenaufwurfes

Noch klarer wird die Zusammenarbeit der Komponenten am Beispiel des Aufrufes einer lokalen Hilfe: Ein Klick auf den Button mit dem I-Icon aktiviert die Funktion INFO und gibt dieser Funktion den Wert „Office“ mit. Die Funktion nimmt diesen Wert an und bereitet einen Datenbankaufruf mit AJAX vor. Der Aufruf richtet sich an die php-Datei getInfo.php die oben schon beschrieben wurde. Diese Datei gibt einen Datensatz an den Request des Aufrufes zurück. Wird als Status der Text „Erfolg“ übergeben, findet die Verarbeitung am Client statt. Aus dem Rückgabewert des php-Scripts wird das Datenfeld „Infotext“ entnommen und mit einem Modal von Bootstrap dargestellt.

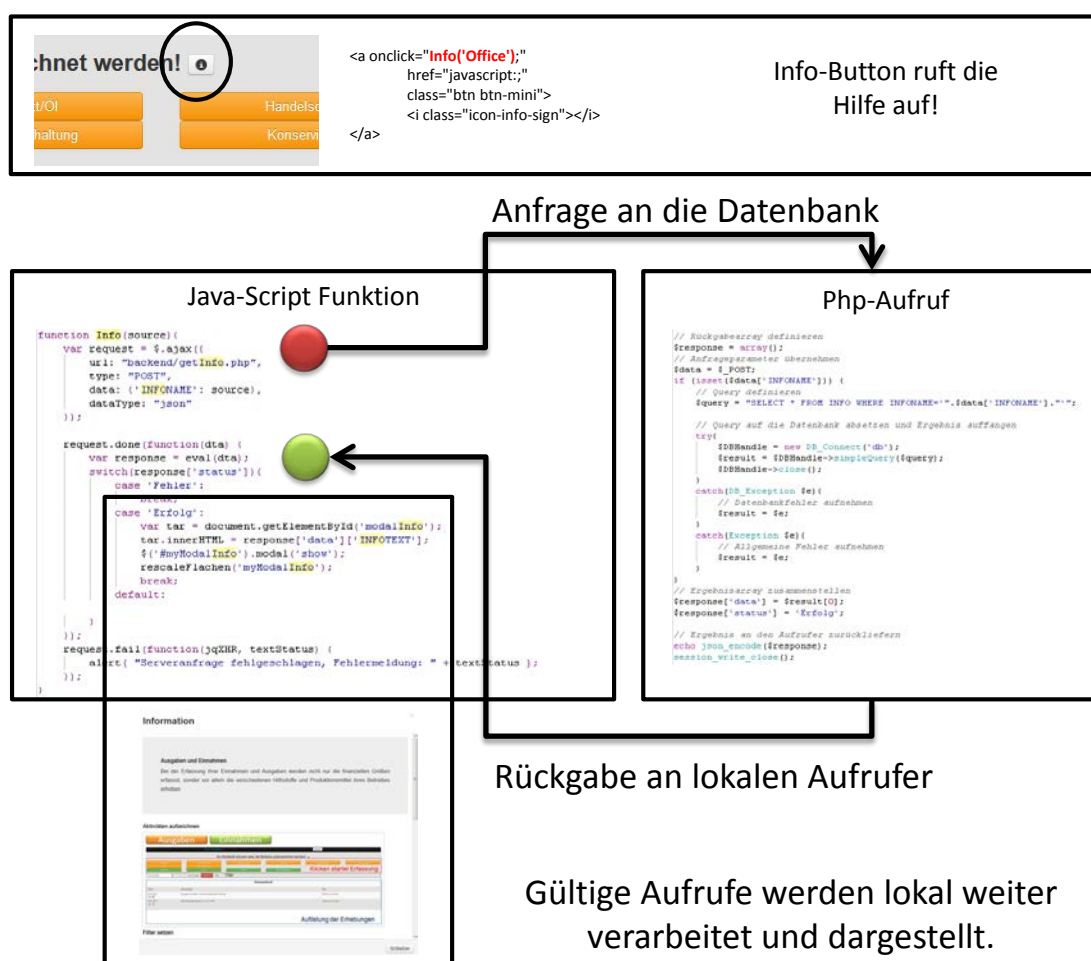


Abbildung 21: Fallstudie der Integration von HMTM, JS und php

## **2.6 Tests**

Für die Entwicklung von CAP wurde die Entwicklungsstrategie des Rapid-Prototyping angewandt. Mit einigen ausgewählten landwirtschaftlichen Betrieben wurde ein Best Practice Konzept entwickelt und zuerst grob implementiert. Die Betriebe haben die neuen Module jeweils getestet und Veränderungsvorschläge gemacht. So wurde iterativ eine Verbesserung des Grundmodells erreicht die noch nicht abgeschlossen ist. Das erste Erhebungsjahr von FarmLife ist durchaus noch zur Testphase zu zählen. Die Berater die derzeit mit den Betrieben an der Erfassung arbeiten kennen die Entwicklungsstrategien und melden laufend Wünsche zur Verbesserung an. Auf eigenständige Unit-Tests wurde verzichtet.

## **2.7 Integration und Einführung**

Die aktuelle Nutzung von CAP läuft derzeit in einem engen Beratungsgespräch mit den Landwirten ab. Jeder Landwirt wird von seinem Berater zu einem Beratungsgespräch besucht. Am Betrieb wird eine kurze, grundsätzliche Einführung in den Sinn und Möglichkeiten der Ökobilanzierung gehalten. Die Berater beginnen dann mit den Landwirten die Daten in den einzelnen Modulen zu erfassen. Bei der Festlegung der Grenzen hat sich eine etwas umfassendere Unterstützung bewährt. Alle 14 derzeit aktiven Berater wurden vorab am LFZ in einer halbtägigen Veranstaltung in CAP eingeführt und melden die Betriebsbesuche vorher bei der Master-Control-Station = Back Office an. Dies ist dann besetzt und steht für telefonische Rückfragen zur Verfügung. Dieses Beziehungsdreieck aus Landwirt-Berater-Back Office hat sich gut bewährt und wird vor Ort auch geschätzt.

## **2.8 Feedback/Ausblick**

Die Anwender – sowohl Berater als auch Landwirte – sind mit CAP durchaus zufrieden und empfinden das Design als praxisnahe und modern. Gelegentlich bestehen noch Probleme mit den verschiedenen Browsern auf den unterschiedlichen Rechnern der Landwirte. Diese werden am einfachsten durch die Vor-Ort Installation der neuesten Version von Mozilla Firefox gelöst.

Farmlife-Capture wird zur Erfassung von Betriebsdaten – das steht jetzt schon im Kollegenkreise des LFZ fest – nicht nur für das aktuelle Projekt, sondern für alle weiteren wissenschaftlichen Projekt auf Betriebsebene verwendet werden!

## 3 Anhang

### 3.1 Abbildungsverzeichnis

Abbildung 1: ER-Modell Übersicht .....	18
Abbildung 2: WKT aus en.wikipedia.org .....	21
Abbildung 3: Ablauf eines Einzelprozesses.....	26
Abbildung 4: Übersicht Farmlife .....	31
Abbildung 5: Formular der Betriebsdaten.....	33
Abbildung 6: Präsentation von Eingabeprozessen.....	34
Abbildung 7: Erfassungsfenster des Prozesses "Energie einkaufen" .....	35
Abbildung 8: Auswahl von betroffenen Flächen .....	35
Abbildung 9: Erfassung von Feldstücken .....	36
Abbildung 10: Grunddefinition Schlag.....	37
Abbildung 11: Zusätzliche Informationen zum Schlag .....	38
Abbildung 12: Räumliche Strukturen nach Prasuhn & Grünig .....	38
Abbildung 13: Bewirtschaftungseinheit anlegen .....	39
Abbildung 14: Schläge zur Bewirtschaftungseinheit hinzufügen.....	39
Abbildung 15: Optionen zum Digitalisieren von Schlägen .....	40
Abbildung 16: Status eines Schlages während der Bearbeitung.....	41
Abbildung 17: Dateistruktur CAP.....	43
Abbildung 18: JS-Funktionskern der Datenerfassungsteile (feld.js) .....	44
Abbildung 19: JS-Funktionskern der räumlichen Erfassung (flachen.js) .....	45
Abbildung 20: Übersicht über alle php-Dateien .....	47
Abbildung 21: Fallstudie der Integration von HMTM, JS und php.....	48