

Mehrstufige und verteilte landwirtschaftliche Geschäftsprozesse - Vorstellung eines offenen prozessunterstützenden Architekturkonzepts -

Thilo Steckel¹, Volker Brandt¹, A. Kluge², K. Oetzel²

¹Entwicklung Systeme und Dienstleistungen
CLAAS Selbstfahrende Erntemaschinen GmbH
Münsterstr. 33

33428 Harsewinkel
² AGROCOM GmbH & Co. Agrarsystem KG
Potsdamer Str. 211
33719 Bielefeld
thilo.steckel@claas.com
volker.brandt@claas.com
kluge@agrocom.com
koetzel@agrocom.com

Abstract: Neue Internet-basierte Standards und Methoden erleichtern die bedarfsgerechte Anpassung landwirtschaftlicher Geschäftsprozesse an sich dynamisch ändernde Anforderungen. Der Beitrag beschreibt grundlegende Bestandteile einer unterstützenden Systemarchitektur und eine beispielhafte Realisierung.

Einleitung

Durch eine zunehmende Erweiterung der informationstechnischen Sicht auf gesamte landwirtschaftliche Wertschöpfungsketten (Beschaffung, Prozessoptimierung (technisch und betriebswirtschaftlich), Konzentration auf Kernkompetenzen, Vermarktung, Produktsicherheit und Administration) ergibt sich eine Vielzahl von Schnittstellen und Abhängigkeiten. Die dadurch entstehende Komplexität innerhalb der Geschäftsprozesse ist mit den bislang eingesetzten Methoden der Systementwicklung nur schwer zu beherrschen.

Um den dabei entstehenden Anforderungen der Prozessteilnehmer z.B. nach Verfügungsgewalt über eigene Daten, Hoheit dezentraler DV-Systeme und Offenheit von Kommunikation gerecht zu werden, sind für die Strukturierung der Inhalte und die Steuerung der Kommunikation geeignete Verfahrenweisen zu entwickeln.

Die systematische Strukturierung der Inhalte erfolgte im Rahmen der beim KTBL angesiedelten Arbeits- und Expertengruppe agroXML. Dazu wurden zunächst elementare Bereiche zur Abbildung eines landwirtschaftlichen Betriebes und der pflanzenbaulichen

Erzeugung in Form eines XML-Schemas abgebildet. Das agroXML-Schema wurde im Juni 2005 in seiner ersten Form verfügbar gemacht [Ag05]. Damit ist es schon zum jetzigen Zeitpunkt möglich, Daten in einer standardisierten Form zu beschreiben und zwischen Geschäftspartnern auszutauschen. Zur Förderung der Mehrfachverwendbarkeit in diversen Geschäftsprozessen erfolgte eine Trennung von Inhalten (agroXML) und Geschäftsprozessen in denen z.B. agroXML-konforme Daten ausgetauscht werden.

Architektur

Zur Steuerung der Geschäftsprozesse wurde eine Architektur entwickelt, die die Vernetzung der Geschäftspartner realisiert. Die Architektur besteht prinzipiell aus einem zwischen den Partnern vermittelnden Intermediär sowie anwendungsunabhängigen Clients, die auf den dezentralen Systemen zum Einsatz kommen.

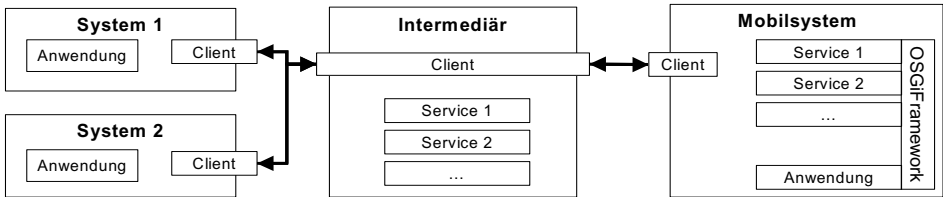


Abbildung 1: Gesamtarchitektur

Im Mittelpunkt der Architektur steht ein Intermediär, der mit seinen Kernfunktionen im Falle mehrstufiger Prozesse die zustandsbehaftete oder zustandslose Kommunikation zwischen den Geschäftspartnern steuert. Zusätzlich soll er ergänzende Dienste wie z.B. eine agroXML-Transformation oder Loggingfunktionen bereit stellen. Die Kommunikation zwischen Clients und Intermediär erfolgt auf Basis ebMS-konformer Web Services [Oa05]. Der Intermediär ist als Web-Applikationsserver (AXIS-Framework) realisiert.

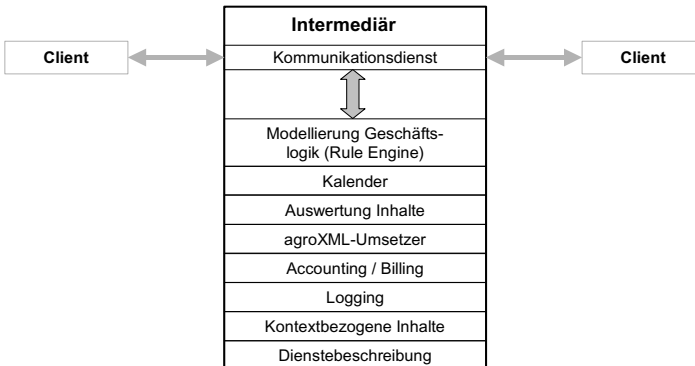


Abbildung 2: Intermediär

Für die Vernetzung der dezentralen DV-Systeme, z.B. Hof-PC, mobile Bordrechner, kommen anwendungsunabhängige Clients zum Einsatz. Vor der erstmaligen Teilnahme eines dezentralen Systems lädt der Client die Beschreibung des benötigten Geschäftsprozesses (Workflow, Teilnehmer) herunter. Anschließend werden die Daten entsprechend den Vorgaben zwischen den Partnern vermittelt. Der Client besteht aus einer Kommunikationskomponente (auch hier mit Web Services) und einer Applikationskomponente, die als Bindeglied zwischen der dezentralen Anwendung und der Kommunikation zu

sehen ist. Zugriffe erfolgen zunächst auf Verzeichnis- und Dateiebene. Weitere Verfahren auf Datenbank- und Applikationsebene werden realisiert. Zur Einbindung unterschiedlicher Systeme wurde der Client in Java und .NET entwickelt.

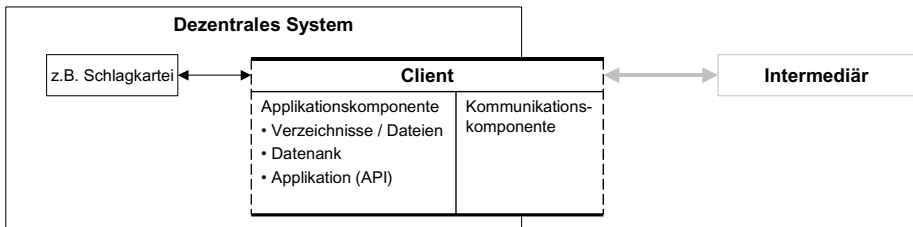


Abbildung 3: Client

Die Architektur ermöglicht neben der Vermittlung von Dokumenten und Daten auch die Übertragung von Anwendungen sowie die Einbindung weiterer Dienste. Insbesondere im Bereich der mobilen Datenverarbeitung gestaltet sich das Nachladen von Anwendungen problematisch. Aus diesem Grund ist für Bordrechner die Einbindung eines OSGi-Frameworks [Os05] vorgesehen. Dieses Framework erlaubt die nachträgliche Installation von Diensten unter Berücksichtigung der besonderen Anforderungen im mobilen Umfeld (Footprint, Sicherheit).

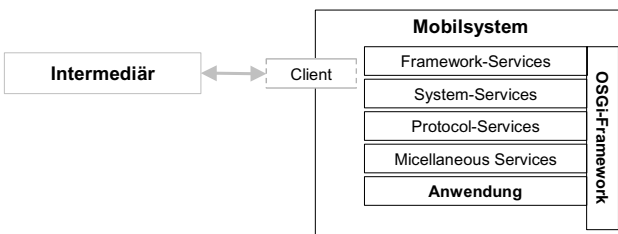


Abbildung 4: OSGi-Framework

Beispielanwendung

Aufbauend auf der beschriebenen Architektur wurde eine Beispielanwendung mit verteilten und voneinander abhängigen Prozessen entwickelt. Die Anwendung stellt Auftrags- und Kontextinformationen zur ordnungsgemäßen Durchführung einer Pflanzenschutzmaßnahme, insbesondere die Einhaltung von Abständen zu Nicht-Zielflächen bereit. Mit Hilfe der entwickelten Architektur wird der abgebildete Prozess unterstützt.

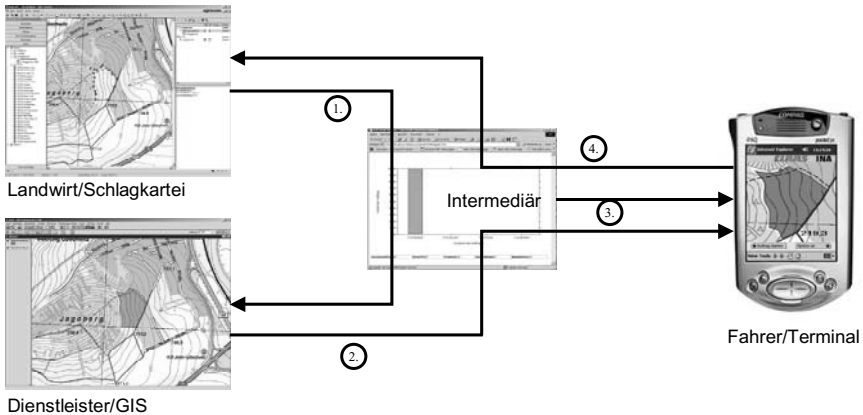


Abbildung 5: Beispielanwendung

- (1) Neuer Auftrag wird von Schlagkartei an GIS geschickt.
- (2) GIS ermittelt exakte Bearbeitungsfläche und leitet Auftrag an Fahrer weiter.
- (3) Terminal lädt die notwendigen Services für den Anwendungstyp nach.
- (4) Terminal schickt nach Auftragsende protokollierte Daten an Schlagkartei.

Die gesamte Kommunikation wird vom Intermediär gesteuert. Schlagkartei und GIS wurden ohne programmiertechnische Änderungen in den Geschäftsprozess eingebunden.

Literaturverzeichnis

- [Ag05] agroXML: URL: <http://www.agroxml.de/> [Stand: 01.08.2005].
[Oa05] OASIS: Message Service Specification 2.0: URL: http://www.oasis-open.org/committees/ebxml-msg/documents/ebMS_v2_0.pdf [Stand: 01.08.2005].
[Os05] OSGi-Alliance: URL: <http://www.osgi.org/> [Stand: 1.8.2005].

Das F&E-Projekt INA (Integrierte und multimedial gestützte Agrardienstleistungen in virtuellen Strukturen) wurde vom Bundesministerium für Wirtschaft und Arbeit (BMWA) unterstützt und am 31.12.2005 abgeschlossen (Förderkennzeichen 01MD201).