Projekthintergrund eStep

Das Projekt eStep Mittelstand (www.estep-mittelstand.de) hat die Schaffung von modularen Lösungen für den Mittelstand zur Stärkung der eigenständigen Integration von eBusiness-Standards in komplexen Lieferketten-Prozessen zur Aufgabe. Diesem Ziel wurde in den vergangenen drei Jahren nachgegangen und es entstand eine entsprechende Toolkette, nebst zugehöriger Handlungsempfehlung und DIN-Spec. Im Hinblick auf die Werkzeuge bildet ein Self-Assessment Tool für KMU zur Standortbestimmung und Identifikation erster Handlungsfelder den Beginn der Kette. Daran schließt sich der Entscheidungsbaum an, welcher unter Berücksichtigung einer Vielzahl von Kriterien eine strukturierte Empfehlung für den einzusetzenden Standard ermittelt und damit das Unternehmen bei der Wahl des richtigen Standards intensiv unterstützt. Die dritte Komponente, die sich logisch an diese Toolkette anschließt, ist eMiMi - die eStep Mittelstand Middleware - die eine digitale, standardisierte Brücke zwischen dem KMU und seinen Partnern schlägt. Dazu wurde ein Konzept entwickelt sowie eine erste, prototypische Implementierung vorgenommen.

Das Konzept

Die Middleware besteht konzeptionell aus den drei klassischen Clustern, der Transformation, der Verarbeitung und der Kommunikation. Die Zusammenstellung der verschiedenen Services der Cluster erfolgt durch einen konfigurierbaren und parametrisierbaren Workflow. Basis dieses Workflows ist eine Nachricht, die aus einem Header, dem aktuellen Workflow-Schritt mit seinen Parametern, dem definierten Workflow sowie einem Inhaltscontainer mit einem Datenelement. Der Datencontainer ist dabei generisch, die inhaltliche Richtigkeit des darin enthaltenen Inhalts muss durch den empfangenen Service geprüft und interpretiert werden. Der empfangende Service kennzeichnet im Worfklow-Element den Status seiner Bearbeitung und verändert ggf. den Inhalt des Datencontainers. Anschließend gibt er die Nachricht an den Workflow-Service zurück, der den weiteren Ablauf anhand des Bearbeitungsstatus und dem in der Nachricht enthaltenen Gesamtworkflow steuert. Im Erfolgsfall wird der nächste Workflow-Schritt ausgeführt, im Fehlerfall die Nachricht an den Fehlerservice weitergereicht.

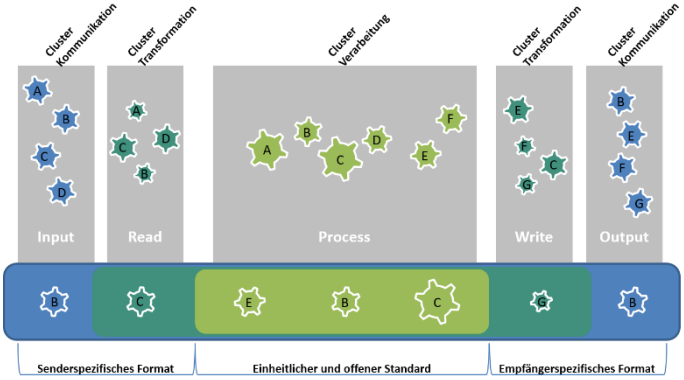

Abbildung 1: Konzept der eStep Mittelstand Middleware

Die drei Cluster enthalten beliebig viele Services, die jeweils spezielle Aufgaben möglichst kleinteilig und abgeschlossen erfüllen. Der Transformations-Cluster bietet Services, die ein beliebiges Format in das Intermediär-Format transformieren (Read) oder aus dem Intermediär verschiedene Ergebnisformate erzeugen (Write). Sie dienen damit als logischer Übergang in die Verarbeitung auf Basis des generischen Intermediär-Formats und aus dieser heraus. Jeder Transformations-Service muss mindestens eine Formattransformation beherrschen, er kann jedoch auch beliebig viele umsetzen.

Der Cluster der Verarbeitungs-Services (Process) führt eine beliebige Aktion auf einem Intermediär Element aus. Dabei kann er dieses Intermediär-Element verändern oder ergänzen. Für den Intermediär wurde der offene und umfassende Standard UN/CEFACT gewählt, so dass alle Services auf den für ihren Anwendungsfall spezifischen Teilbaum aus diesem sehr umfangreichen Datenmodell referenzieren können. Mögliche Services sind beispielsweise die Anreicherung von textuellen Informationen auf Basis eindeutiger Identifikationsnummern (GTIN, GLN), die Durchführung einer Bonitätsabfrage, die Anreicherung mit zentral gespeicherten Informationen wie bspw.

Artikeldarstellungen oder die Registrierung einer Artikelbewegung in einer zentralen Tracking-Datenbank.

Der Kommunikations-Cluster wiederum bietet Services zur Übermittlung der Eingangs- (Input) und Ausgangsdatei (Output). Hier sind generische Ansätze wie Webservices für den Workflow-Start über individualisierte Wege wie dem E-Mail Versand mit Anhang bis hin zu proprietären Verbindungen zu anderen IT- und Software-Systemen denkbar.

Die Technik

Zur Erfüllung der vorgenannten Anforderung sowie zur einfachen Anbindung von Legacy-Systemen und neuer Systeme wurde eine Service-orientierte Architektur gewählt. Als zentrale Komponente der Architektur kommt hier eine Queue (ActiveMQ) zum Einsatz. Durch offene Protokolle wie STOMP ist die eigentliche Implementierung jedoch losgelöst von der Interaktion mit der Queue. Diese Trennung von Kommunikation und Implementierung gibt alle Freiheiten bei der Umsetzung und Betrieb. So sind sowohl klassische Organisationsformen als auch moderne DevOps-Ansätze möglich.

Zur Vermeidung von monolithischen Strukturen wurde jede Nachricht auf der Queue so gestaltet, dass diese alle notwendigen Informationen enthält und es keine zentrale, wissenstragende Komponente geben muss. Damit wird zudem die Skalierbarkeit des Systems sichergestellt, da ebenfalls keine der Komponenten Wissen über den Gesamt-Verarbeitungsstatus der Nachricht haben muss, was auch die parallele Verarbeitung von Nachrichten einfach ermöglicht.

Die Auflösung und Interpretation des Workflows erfolgt durch einen eigenen Service, der den Workflow an Hand des Nachrichtenkopfes rekonstruieren kann. Für jede Art des Workflows existiert eine eigene Queue und für jeden Service existiert eine Queue, damit sind nicht nur die Services voneinander unabhängig, sondern auch unabhängig vom Workflow an sich.
Eine solche getrennte Architektur bringt viele Vorteile mich sich, ergibt aber auch eine hohe Komplexität.

Mit dem Wissen, dass der Einsatzzweck sich auf die oben beschriebenen Schritte READ, PROCESS und WRITE fokussiert, ist es möglich dem Nutzer eine API zur Verfügung zu stellen, die einen Teil der Komplexität kapselt und eine einfache Integration erlaubt. Die im Rahmen der eStep Mittelstand Middleware entwickelte Java API erwartet vom Nutzer eine einzige überschriebene Methode

```
String doProcess(WorkflowElement element, String messageBody) throws Exception,
```

die eine Nachricht empfängt, dekodiert und wieder zurückgibt. Die Fehlerbehandlung erfolgt durch eine Exception, die nicht von der Implementierung selbst behandelt werden muss. Im einfachsten Fall hat die verarbeitende Komponente keinen Zugriff auf das Workflow Element und ist so noch freier in den Workflow intergrierbar. Damit wird der gesamte Overhead der SOA Architektur vom Nutzer weggekapselt und dieser kann sich auf die Business-Logik des Service konzentrieren.

Die (De-)Serialisierug des Workflows und dessen Elemente als JSON Objekt erfolgt über Strategien in einer Factory. Diese wird zur Laufzeit konfiguriert und ermöglicht so ebenfalls maximale Flexibilität.

Die beschriebene Architektur bietet damit also einen modularen Ansatz auf zwei Ebenen. Auf der einen Seite bietet die Queue die Möglichkeit eigene Services zu verwenden ohne dass diese an einer zentralen Stelle registriert werden müssen. Auf der anderen Seite gibt die technisch tiefer liegende API für den Workflow und seine Elemente alle Hilfsmittel an die Hand, um neue Implementierungen zu erstellen.