

PRODUCTDASHBOARD – LIZENZVERWALTUNG & UPDATEPROZESS

Matthes Nagel, Robert Janetzko

n³ data analysis – software development – consulting GmbH & Co. KG,
Dr.-Friedrichs-Str. 42, D-08606 Oelsnitz

Als Software Hersteller mit eigenen Produkten wird man täglich vor die Herausforderung gestellt einen schlanken und reibungsarmen Support-Prozess zu bieten. Der Support-Vorgang bzw. -Prozess beginnt mit der Auslieferung und Installation der Software, sodass der erste Support Teilprozess die Software-Lizenzierung bzw. Software-Aktivierung ist. Neben der Aktivierung sind Software-Upgrades und Software-Updates häufige Use Cases, welche im Rahmen eines Supportvertrages benötigt werden. Ziel des Projektes ProductDashboard war die Schaffung einer eigenen Plattform zur Abbildung der oben genannten Prozesse inkl. der Erweiterungsmöglichkeit zur Abbildung von Lösungsupdates und zur Visualisierung von Lösungsstatus. Dazu wurde eine eigene Webapplikation auf Basis aktueller Webtechnologien implementiert, welche mittels Webinterface vom Mitarbeiter bedient werden kann und ebenfalls direkt mit den Softwareinstallationen beim Kunden kommuniziert. Selbstverständlich wurden dabei auch mobile Zugänge – wenn der Mitarbeiter beim Kunden vor Ort ist – berücksichtigt und realisiert.

1. Einleitung

Gängiges Businessmodell der meisten Softwarehersteller ist der Umsatz mittels Softwarelizenz- und Softwarepflegekosten. Indem ein Softwarehaus eine definierte Marktlücke entdeckt bzw. ein Problem einer Branche identifiziert und dies mit einer Software behebt kann relativ viel Umsatz mit dieser Software erzielt werden. Der Softwarehersteller hat mit seiner Software ein Produkt geschaffen, welches dem Urheberrecht unterliegt und in Form von Lizenzen, welche Kunden die bestimmungsgemäße Benutzung erlauben, verkauft. [1]

Doch wie verhindert der Hersteller, dass der Kunde X die Software an den Kunden Y unter der Hand weitergibt oder verkauft? Die meisten Hersteller geben hierzu Lizenzschlüssel für ihre Software heraus mit deren Hilfe nachvollziehbar ist wer die Software verwendet bzw. teilweise sind in die Softwarelizenzen die Kunden „hineincodiert“. Ziel eines Softwarehauses ist es die Lizenzschlüsselgenerierung so geheim und sicher wie möglich zu machen um eine böswillige Duplizierung oder Erstellung zu vermeiden.

Abhängig vom Markt und der Lösung gibt es unterschiedliche Lizenzklassen [2] (z.B. Vollversion, Upgrade, etc.) und Lizenzmetriken [3] (z.B. pro Installation, pro Anwender, pro Prozessor, etc.)

Ziel der hier vorgestellten Softwarelösung ist es, das Lizenzmanagement eines Softwareherstellers als Prozess abzubilden. D.h. unterschiedliche Softwareprodukte, Lizenzklassen, Lizenzmetriken, Kunden, etc. zu verwalten und zu verarbeiten. Neben der reinen Lizenzierung ist eine weitere Anforderung das Updatemanagement mit dem Lizenzmanagement zu kombinieren.

2. Anforderungen

Im Rahmen der Konzeption und Realisierung des ProductDashboard galt es die folgenden Anforderungen umzusetzen.

2.1. Akteure

Als Akteure, welche in Form von Use Cases mit dem ProductDashboard interagieren, sind drei zu erläutern:

Kunde: ist Firma bzw. Personen, welche sich Nutzungsrechte an Softwareprodukten des Softwareherstellers erworben hat. Aus Kundensicht muss eine zeitnahe Freigabe und Aktivierung der Software erfolgen, sowie Updates regelmäßig und mit minimalem Aufwand installiert werden.

Mitarbeiter: ist Angestellter des Softwareherstellers und dafür verantwortlich, dass die Kunden, die erworbene Software erhalten, dass keine Drittakeure fremde Lizenzen verwenden und dass Updates und Upgrades zeitnah und ohne größere Kundenaktivitäten ausgeliefert werden.

Produkt: ist Softwareprodukt für das Lizenzen erstellt, vergeben und aktiviert werden sollen. Die vollständige Lizenzverwaltung eines Softwareproduktes soll mittels des ProductDashboards erfolgen.

2.2. Use Cases

Use Cases oder Anwendungsfälle beschreiben alle Szenarien die auftreten können wenn ein Akteur mit Hilfe des beschriebenen Systems (ProductDashboard) versucht ein Ziel zu erreichen. Die hier erläuterten Use Cases sind in fachlich ähnliche Gruppen zusammengefasst. Eine detaillierte Use Case Beschreibung ist in der Projektdokumentation enthalten.

Kundenverwaltung: wird vom Akteur Mitarbeiter ausgeführt und beschreibt alle Funktionen zum Anlegen, Aktualisieren, Darstellen und Verwalten der Endkunden.

Produktverwaltung: wird analog vom Akteur Mitarbeiter benötigt. Hier werden alle Softwareprodukte mit den jeweiligen Lizenzmetriken hinterlegt und administriert. Ziel ist es hier eine Umgebung zu schaffen in die neue Produkte ohne programmier-technische Änderungen hinzugefügt werden können.

Lizenzverwaltung: der Akteur Mitarbeiter teilt den Kunden hier die Produkte zu und stellt parametrisiert die Lizenzmetriken (z.B. Anzahl User, etc.).

Produktaktivierung: im Rahmen einer automatischen Produktaktivierung überprüft das Softwareprodukt selbständig mit dem ProductDashboard inwiefern die Lizenz erneut verwendet werden darf. Ist eine automatische Aktivierung nicht gewünscht oder technisch nicht möglich, muss der Kunde per Telefon/Email seinen Aktivierungsschlüssel dem Mitarbeiter zustellen, sodass dieser dann die Aktivierung manuell vornehmen kann.

Softwareupdates: neben der Aktivierung und Lizenzierung der installierten Software soll das ProductDashboard ebenfalls Softwareupdates verwalten. D.h. der Akteur Mitarbeiter gibt ein neues Software Release frei und das Produkt bekommt diese Information bei der nächsten Interaktion. Liegt ein neues Update vor, bekommt das Produkt die Adresse und lädt sich die Inhalte entsprechend herunter.

Produktmonitoring: neben Produkten, die direkt vom Kunden bedient werden, gibt es auch Produkte die im Hintergrund definierte Tasks ausführen – ohne Benutzerinteraktion. Um diese Tasks überwachen zu können kann das Produkt einzelne Stati zurückschicken und somit den Mitarbeiter indirekt informieren.

2.3. Prozess

Ziel dieses Projektes ist es alle Use Cases in Form eines durchgehenden Business Prozess nach der BPMN 2.0 [4] zu entwerfen und diesen BPMN Prozess mittels Software zu realisieren.

3. Realisierung

Aufbauend auf den Anforderungen wird deren Realisierung im folgenden Kapitel erläutert.

3.1. Systemarchitektur

Für die Realisierung werden verschiedene Server eingesetzt, die für ihre jeweilige Aufgabe optimiert wurden und in unterschiedlichen Netzen an unterschiedlichen Orten stehen. Das zentrale System ist der Anwendungsserver, über welchen die Konfiguration der Kunden, Lizenzen und auch die Verteilung direkt oder über einen Mail-Server erfolgt. Die gespeicherten Kunden- und Lizenz-Informationen werden auf einem separaten Datenbank-Server abgelegt. Die Bereitstellung der Softwaredaten für Updates erfolgt über einen ausgelagerten Datei-Server. Der externe Datei-Server sitzt in einem entsprechenden Rechenzentrum und hat eine für größere Bandbreiten ausgelegte Internetanbindung. Der Anwendungsserver sitzt hingegen beim Softwarehersteller lokal, welche meist geringer konzipierte Internetanbindungen besitzen.

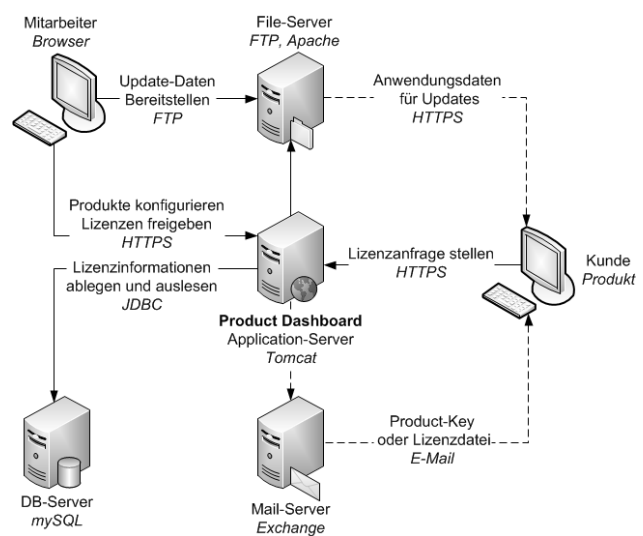


Abbildung 1: Systemarchitektur

3.2. Technologiestack

Um die Verwaltung der Lizenzen von überall zu ermöglichen, wurde das ProductDashboard als Web-Applikation realisiert. Als Applikations-Server wurde ein Tomcat [5] mit MySQL-Backend [6] verwendet. Die Daten für Softwareupdates werden per FTP auf einem Datei-Server abgelegt. Ein Apache-Webserver wird verwendet um auf Anfragen des ProductDashboards zu antworten oder Dateien zum Kunden zu übertragen. Die eigentliche Applikation – das ProductDashboard wurde mittels dem J2EE Framework Spring [7] und den Modulen Spring Roo [8], Spring Security [9] und Spring LDAP [10] implementiert.

3.3. Prozess & Kommunikation

Das gesamte Lizenz- und Updatemanagement erfolgt in drei Phasen: Erfassung, Aktivierung, Update. Nachdem der Kunde ein Softwareprodukt erworben hat, muss ein Mitarbeiter des Softwarehauses den Kunden und seine erworbenen Produkte erfassen. Er meldet sich dazu über seinen Browser am ProductDashboard an. Die Authentifizierung erfolgt hierbei per LDAP über die zentrale Benutzerverwaltung der Firma. Im ProductDashboard selbst müssen keine Benutzer konfiguriert werden. Anschließend wird ein neuer Eintrag für den Kunden erzeugt, sofern dieser noch nicht vorhanden ist. Danach kann ein neues Produkt für diesen Kunden angelegt werden. Dazu gehört die Auswahl der Softwareprodukte die der Kunde nutzen darf zusammen mit verschiedenen Rahmenbedingungen (Aktivierungsart, Lizenzklasse, Lizenzmetriken, Updates). Für jedes Produkt wird ein eindeutiger Produktschlüssel generiert. Dieser Schlüssel wird dem Kunden automatisch per E-Mail mitgeteilt.

Nachdem der Kunde seine Software installiert hat, muss er diese aktivieren bevor er sie verwenden kann. Dieser Schritt ist notwendig, um sicher zu stellen, dass der die Software rechtmäßig erworben wurde. Jede Installation errechnet dabei einen

Hardwareschlüssel über die sie eindeutig bestimmt werden kann. Zur Aktivierung können zwei Verfahren verwendet werden. Bei der automatischen Aktivierung kann der Kunde seinen Produktschlüssel verwenden, um über eine Internetverbindung eine Freischaltung zu erbitten. Die beiden Schlüssel (Produkt- und Hardware Schlüssel) werden dabei zum ProductDashboard gesendet. Dort wird dann über den Produktschlüssel ermittelt, ob und wie der Kunde die Software nutzen darf. Anschließend wird eine Lizenzinformation mit einem Aktivierungsschlüssel für den Hardware Schlüssel erzeugt und in der Datenbank abgelegt. Im Erfolgsfall wird eine Lizenzdatei zum Kunden übertragen, die den Aktivierungsschlüssel und die Nutzungsinformationen enthält. Kann die Software keine direkte Internetverbindung herstellen, kann der Kunde den Hardware Schlüssel per Telefon oder E-Mail mitteilen. Ein Mitarbeiter trägt ihn dann manuell in das ProductDashboard und kann von dort direkt eine Antwort-E-Mail mit der Lizenzdatei zurückschicken.

Die Software prüft beim Start die Lizenzdatei und nutzt die enthaltenen Informationen, um zu entscheiden ob und in welchem Umfang der Anwender die Software nutzen darf. Dies kann unter anderem den Zugriff auf eine automatische Software-Aktualisierung umfassen.

Der Mitarbeiter des Softwarehauses hat die Möglichkeit über das ProductDashboard neue Versionen eines Produktes freizugeben. Dazu muss er zunächst die zu aktualisierenden Daten auf dem Datei-Server per FTP bereitstellen. Das ProductDashboard fragt dann beim Datei-Server an, ob neue bzw. welche Versionen verfügbar sind. Ist dies der Fall, kann der Mitarbeiter das Update im ProductDashboard freigeben. Startet nun der Kunde seine Software das nächste Mal, sendet dieses eine Anfrage an das ProductDashboard, ob neue Versionen freigegeben wurden und erhält als Antwort die Liste der Dateien die vom Datei-Server abgeholt werden müssen, um das Update durchzuführen. Diese Dateien werden dann per HTTP-Download übertragen und installiert.

Führt das Softwareprodukt wiederkehrende Aufgaben ohne Benutzerinteraktion aus, muss eine Rückmeldung über Erfolg oder Nichterfolg geschehen. Hierzu kann das Produkt Statusmeldungen an das ProductDashboard versenden. Diese enthalten neben der eigentlichen Meldung auch den Produkt und Hardware Schlüssel, so dass dem Mitarbeiter genaue Informationen zu Verfügung stehen welche Installation wann ausgeführt wurde und wo Fehler aufgetreten sind.

4. Fazit

Das abschließende Fazit ist mehr als positiv zu betrachten. So kann durch das ProductDashboard, welches als verteiltes Softwaresystem konzipiert und realisiert wurde enorme Vorteile erzielt werden.

- Die bestehende, verteilte und nichtsynchrone Excel-Lösung wurde durch eine webbasierte Multi-User Lösung ersetzt.

- Durch ein zentrales Kunden- und Produktmanagement erfolgt eine klare und einheitliche Zuordnung sowie eine zentrale Übersicht.
- Mit Hilfe des Mobile-Template kann direkt beim Kunden vor Ort über ein Smartphone auf die Lizenzdaten zugegriffen werden.
- Der Aufwand für die Produktaktivierung durch den Telefonsupport konnte durch die automatische Lizenzaktivierung erheblich gesenkt werden.
- Dank des Produktmonitoring und den damit verbundenen Statusinformationen kann erstmals ein proaktiver Support geliefert werden, da der Mitarbeiter Fehlfunktionen vor den Kunden bemerken kann.

Danksagung

Die Autoren bedanken sich beim Team von n³ für die fachliche Unterstützung und das Entgegenkommen bei der Anfertigung dieses Artikels.

Literaturverzeichnis

- [1] S. Donovan: Patent, copyright and trade secret protection for software, Potentials, IEEE, 2002, doi:10.1109/45.310923
- [2] Torsten Groll: 1x1 des Lizenzmanagements. Praxisleitfaden für Lizenzmanager. Hanser Verlag, München 2009, ISBN 978-3-446-41828-8 – Seite 44
- [3] Torsten Groll: 1x1 des Lizenzmanagements. Praxisleitfaden für Lizenzmanager. Hanser Verlag, München 2009, ISBN 978-3-446-41828-8 – ab Seite 46
- [4] Jakob Freund, Bernd Rücker, Thomas Henninger: Praxishandbuch BPMN. Hanser Verlag, München 2010, ISBN 978-3-446-41768-7
- [5] Apache Tomcat J2EE Server, URL: <http://http://tomcat.apache.org/> Stand: 01.09.2011
- [6] MySQL – Die populärste Open-Source-Datenbank der Welt, URL: <http://www.mysql.de> Stand: 01.09.2011
- [7] Spring Framework (Core), URL: <http://www.springsource.org/about> - Stand: 01.09.2011
- [8] Spring Roo, URL: <http://www.springsource.org/roo> - Stand: 01.09.2011
- [9] Spring Security, URL: <http://static.springsource.org/spring-security/site/index.html> - Stand: 01.09.2011
- [10] Spring LDAP, URL: <http://www.springsource.org/ldap> - Stand: 01.09.2011