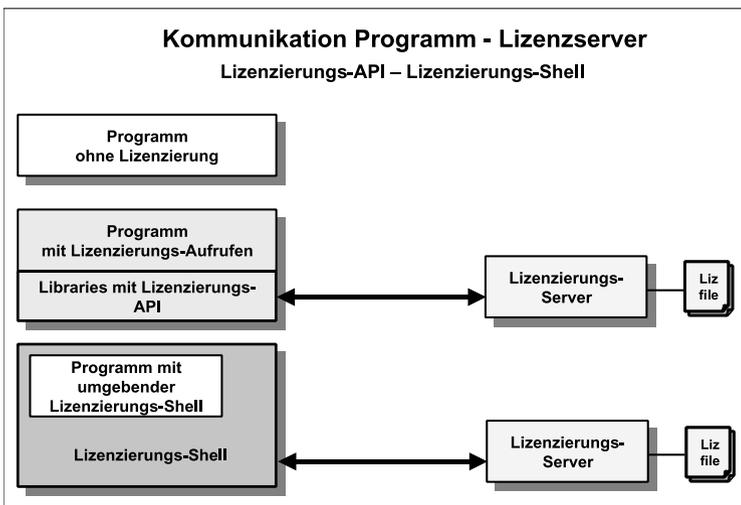


7 Entwicklungsaufwände: Änderungen im Sourcecode

7.1 Libraries und APIs

Um eine effiziente elektronische Lizenzierung zu realisieren, müssen Ihre Programme in Zukunft mit einem Lizenzserver kommunizieren. Dieser wiederum prüft ab, ob eine gültige Lizenz vorliegt oder nicht. Wegen dieser Kommunikation zwischen Ihrem Programm und dem Lizenzserver sind Programmänderungen unvermeidlich. Nur die Verwendung von Lizenzierungs-Shells enthebt Sie dieser Aufgabe, allerdings sinken auch der Komfort und die Anzahl verfügbarer Leistungsmerkmale der Lizenzierung (siehe Kap. 5.3.1: „Lizenzierungsarten“ sowie Kap. 7.3: „Einsatz von Shells“).

Lizenzserver bedingt Programmänderungen



*Abb 5:
Kommunikation
Programm –
Lizenzserver*



Auswahl des geeigneten Tools

Die Lizenztool-Hersteller liefern Ihnen Libraries und APIs, um die notwendigen Lizenzserver-Aufrufe einbauen zu können. Für jede Entwicklungsabteilung sind natürlich die dadurch entstehenden Aufwände von großer Wichtigkeit. Schon bei der Auswahl des Lizenzierungstools ist es wichtig, auf die verfügbaren Schnittstellen zu achten. Einige Hersteller stellen nur C-Libraries zur Verfügung, andere auch solche für Java und Visual Basic¹. Auch die Plattform ist natürlich wichtig: einige Tools unterstützen nur Windows²-Varianten, andere hingegen bis zu 20 verschiedene Plattformen. Hier kommt es nun sehr auf Ihre Produktvielfalt an, um zu entscheiden, welche Bandbreite Sie jetzt oder eventuell in Zukunft benötigen. Ein späterer Wechsel des Lizenzierungstools ist nicht unbedingt anzuraten, so dass Sie sich also zukünftige Optionen durch Auswahl des geeigneten Tools offenhalten sollten.

Aufwände abhängig von der Anzahl Aufrufe

Wenn Sie die Aufwände für eine Lizenzierungs-Einführung abschätzen wollen und Sie konsultieren dazu die Hersteller des Lizenzierungstools, können Sie eventuell Antworten folgender Art hören: „In 15 Minuten ist alles eingebaut und lauffähig“. Es leuchtet ein, dass diese Aussagen natürlich keinerlei Wert haben. Sicherlich hängt der entstehende Aufwand davon ab, welche Anforderungen Sie an die Lizenzierung stellen. Der einfachste Fall wäre eine einzige pauschale Abfrage, ob eine Lizenz pro Applikation vorliegt oder nicht. Hier kann es sein, dass wenige Aufrufe genügen. Aber auch dies will beherrscht sein.

Weitere Gründe für Entwicklungsaufwände

Wodurch entstehen aber mehr Aufwände? Beginnen wir mit der Einarbeitung. Sie werden etwa 2 Wochen brauchen, um die Grundlagen des Lizenzierungstools zu verstehen. Es ist sicher angebracht, vom Hersteller angebotene Schulungen für das Entwicklungsteam wahrzunehmen. In der Regel wollen Sie mehrere Dinge pro Applikation lizenzieren; dies können einzelne Programmfunktionen sein oder zum Beispiel die Anzahl Clients, die die Anwendung im Netzwerk nutzen dürfen. Sie werden sich Gedanken machen müssen, wie Sie auf Fehlerfälle reagieren, und Sie wollen vielleicht alle Sicherheitsmechanismen verwenden, die das Tool bietet. Sie wollen Demo-Lizenzen anders abhandeln als reguläre Lizenzen, und Sie treffen Vorkehrungen für einen zwischenzeitlichen Lizenzausfall zum Beispiel durch Hardwarewechsel.

Folgende **Checkliste** kann Ihnen helfen, alle Aufwände zu kalkulieren:

¹ Visual Basic = Eingetragenes Warenzeichen der Firma Microsoft

² Windows = Eingetragenes Warenzeichen der Firma Microsoft

- Einarbeitung und Schulung
- Abfrage aller verwendeten Lizenztypen
- Erste Tests
- Erzeugung zusätzlicher DLLs (auf Windows) zwecks sicherer Kommunikation
- Zusätzliche Verschlüsselungen
- Veränderte Installationen
- Integrationstests mit dem Lizenzgenerator
- Verhalten nach Installation
- Verhalten bei Lizenzausfall durch Hardwarewechsel
- Verhalten bei Lizenzüberschreitungen
- Vermehrter Testaufwand durch Abprüfen aller Lizenzfehlerfälle

*Checkliste
für alle
Entwicklungs-
aufwände*

Der Einbau der Lizenzierungsaufrufe geschieht zweckmäßigerweise an zentraler Stelle, am besten in einem separaten Modul. Auf diese Weise stören Sie in einem größeren Team nicht die normale Weiterentwicklung des Produktes, da der bisherige Code im Idealfall – nämlich dann, wenn Lizenzierungsabfragen nur bei Programmstart stattfinden – nur um einen Lizenzierungsaufruf an das neue Lizenzmodul erweitert wird. Alle weitere Lizenzierungslogik wird in diesem zusätzlichen Modul realisiert. Dies macht es auch leicht, auf Bedarf die Lizenzierung wieder abzuschalten, sei es zu Testzwecken, sei es für einen beabsichtigten Wechsel des Lizenztools oder weil sich Ihre Firmenpolitik geändert hat.

Zentrales Lizenzierungsmodul

7.2 Lizenzierung auf Funktionsebene

Lizenzierung kann unter Umständen sehr einfach aussehen. Wollen Sie nur überprüfen, ob eine Programmdatei aufgerufen werden darf oder nicht, kommen Sie mit einem simplen Aufruf im Programmcode und einem sehr kleinen Lizenzfile aus.

*Einfachste
Lizenz*

Anders sieht es aus, wenn Ihr Produkt in vielen Varianten und wenn eine Reihe von Leistungsmerkmalen zusätzlich verkauft werden. Sie sind eventuell mit einem Gemisch qualitativ unterschiedlicher Positionen konfrontiert, die sich zwar leicht in dieser Form verkaufen lassen, die aber durch unterschiedliche Lizenztypen abgebildet werden müssen. Dies allein lässt Ihr Lizenzmodul, um das Sie Ihre Quellen erweitern, anwachsen.

*Unterschiedliche
Lizenzabfragen*



| | |
|----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Verschiedene Lizenztypen</i> | <p>Die bereits vorhandenen Verkaufspositionen können unter anderem von folgendem Typ sein:</p> <ul style="list-style-type: none"> ■ Zusätzliche Funktionen, durch Menü aufrufbar ■ Anzahl Clients im Netz ■ Anzahl bestimmter Datenbank-Einträge ■ Upgrades oder Erstversionen ■ Bündel mehrerer Leistungsmerkmale in unterschiedlichen Zusammenstellungen |
| <i>Sind die Programme vorbereitet?</i> | <p>Wichtig aber ist vor allem: Ist Ihre Software bereits in der Lage, alle verkauften Leistungsmerkmale und Varianten freizuschalten oder zu sperren? Denn was nützt Ihnen im laufenden Betrieb das Abprüfen eines lizenzierten Leistungsmerkmals, wenn Sie im Programm auf die positive oder negative Antwort des Lizenzservers gar nicht reagieren können?</p> |
| <i>Wie klar ist die Software strukturiert</i> | <p>Ein Beispiel: Die Berechtigung dafür, dass bestimmte Sätze in der Datenbank Ihres Produktes geändert werden dürfen, verkaufen Sie als kostenpflichtiges Leistungsmerkmal. Diesen Teil Ihres Sourcecodes müssen Sie zum Beispiel insgesamt durch Abfrage eines Schalters sperren können, zusätzlich vielleicht auch einige zugehörige Menüpunkte. Vielleicht ist die Software gar nicht so klar strukturiert, dass Ihnen dies ohne großen Aufwand gelingt?</p> |
| <i>Datenbank-Abfragen</i> | <p>Ein anderes Beispiel: Sie lizenzieren eine komplexe Datenbank-Anwendung. Sie lassen in der Grundversion maximal 500 Basisätze zu. Die nächsten 500 Basissätze verkaufen Sie als kostenpflichtiges Leistungsmerkmal. Diese Informationen haben Sie zwar in der Datenbank verfügbar, aber die Abfrage dafür ist nicht vorhanden. Auch müssten Sie sich gut überlegen, bei welchen Vorgängen die Abfrage aktiviert werden soll: nur bei Programmstart oder bei jedem Einfügevorgang, der die Anzahl Teilnehmer in der Datenbank vermehrt. Und geben Sie als Resultat nur Warnungen aus oder beenden Sie die Software mit einer entsprechenden Meldung?</p> |
| <i>Unterschiedliche Bündelung von Produkt-Funktionen</i> | <p>Und ein drittes Beispiel: Ihre Software besitzt die Funktionen A bis G. Die Grundversion beinhaltet die Funktionen A bis C, die Medium-Version die Funktionen A, B und E (aber nicht C und D), die Professional Version alles. Diese unterschiedliche Bündelung müsste eigentlich schon in der Software realisiert sein. In solch einem Falle wird es Ihnen auch nicht schwerfallen, die schon vorhandenen Staffellungen zu Lizenzaufrufen in Beziehung zu setzen.</p> |

Erst wenn Sie sich genauestens darüber klar sind, ob Sie das derzeitige Verkaufsmodell bei der Lizenzierungseinführung übernehmen wollen, können Sie verlässlich abschätzen, welche Entwicklungsaufwände auf Sie zukommen.

Verkaufsmodell vor Abschätzung der Aufwände

Die abgeschätzten Entwicklungsaufwände stellen auch eine gute Basis für die nächste wichtige Überlegung dar: Bietet nicht die Lizenzierung die Möglichkeit, das Produkt vollkommen anders zu vermarkten? Hier ist eine gute Zusammenarbeit zwischen Produktmarketing und Entwicklung vonnöten. An dieser Stelle sei aber noch einmal folgender dringender Hinweis ausgesprochen:

Rückwirkungen auf Produktmarketing

Warnung:

Packen Sie in das erste Lizenzierungsprojekt nicht zu viele Schwierigkeiten hinein. Sie wollen vor allem erfolgreich lizenzieren und dieses an Ihrem ersten Projekt auch demonstrieren. Das Ausreizen aller Lizenzierungsmöglichkeiten sollten Sie sich für Nachfolgeprojekte oder zukünftige Softwareversionen vorbehalten.

Nicht alle Möglichkeiten ausreizen!

7.3 Einsatz von Shells

Eine Änderung der Programmquellen ist unerlässlich, wenn Sie alle Möglichkeiten eines Lizenzierungstools ausschöpfen wollen. Was aber können Sie tun, wenn Sie ältere Produkte lizenzieren wollen, deren Quellen Sie nicht mehr verändern wollen? Oder wie verfahren Sie mit Fremdprodukten, die ursprünglich nicht lizenziert sind, die Sie aber in Ihre Lizenzierungsanstrengungen mit einbeziehen möchten?

Lizenzierung ohne Zugriff auf Quellcode

Hier kommt das Prinzip der Lizenzierungs-Shell zum Tragen. Eine solche Shell ist eine Software-„Hülle“, die Sie um die zu lizenzierende Software, zum Beispiel ein ausführbares Programm, herumlegen. Alle Aufrufe an den Lizenzserver werden von dieser Shell übernommen.

Shell übernimmt Lizenzierung

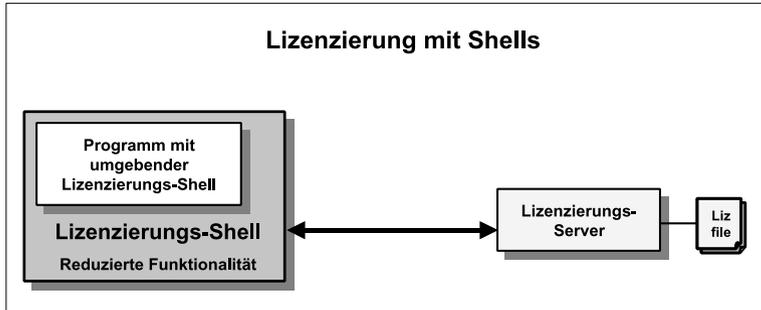
Solche Shells werden von einigen Lizenzierungstool-Herstellern angeboten, bieten aber in der Regel bei weitem nicht den Funktionsumfang wie ein API bzw. Libraries. Wenn Ihre Anforderungen an die Lizenzierungsmechanismen aber nicht allzu hoch sind, ist eine Shell nicht nur für ältere Applikationen, die Sie nicht mehr verändern möchten, die Lösung der Wahl. Auch für Ihre aktuelle Software, mit der Sie Ihren Lizenzierungsprozess starten möchten, ist es vielleicht ausreichend, auf eine einzige Lizenz hin abzufra-

Geringer Funktionsumfang der Shells



gen, und Sie benötigen vielleicht auch nicht die Vielzahl von Modi für das Ausstellen einer Lizenz (siehe Kap. 5.3.1: „Lizenzierungsarten“). Ein weiteres Plus: Sie sind mit einer Shell unabhängig von der Programmiersprache, in der Ihr Programm oder die Fremdsoftware geschrieben wurde.

Abb 6:
Lizenzierung mit
Shells



Abstriche bei der
Sicherheit

Beim Thema Sicherheit müssten Sie eventuell Abstriche machen, wenn Sie eine Shell verwenden. Hier haben Sie es nicht mehr in der Hand, zum Beispiel eigene DLLs in die Datenkommunikation zwischen Applikation und Lizenzserver einzuschalten oder diesen Verkehr noch einmal extra zu verschlüsseln. Hier gilt es, sich genauestens beim Hersteller zu erkundigen, welche Sicherheitsmechanismen in den Datenverkehr mit der Shell eingebaut sind.



<http://www.springer.com/978-3-540-00966-5>

Erfolgreiche Software-Lizenzierung
Electronic License Management - Von der Auswahl bis zur
Installation

Bürkner, R.M.

2003, X, 134 S., Hardcover

ISBN: 978-3-540-00966-5