



Martin Heilmann

Trends in der App-Entwicklung

Apps ziehen zunehmend in die Industrie ein. Was jedoch häufig unterschätzt wird: Der Entwicklungsaufwand steigt proportional zur Anzahl der verwendeten Plattformen und Geräte. Die Lösung ist eine plattformübergreifende App-Entwicklung.

Durch den Entwicklungsprozess mobiler Plattformen seit der Einführung des ersten iPhones im Jahre 2007 stellt sich auch im industriellen Umfeld zunehmend die Frage: Wie kann aus dieser Technologie ein Mehrwert entstehen? Die Kosten für ein handelsübliches mobiles Gerät sind wesentlich geringer als die Kosten für die Entwicklung eigener Hardware in kleiner Auflage. Doch sowohl Android als auch iOS und Windows Phone sind keine echtzeitfähigen Betriebssysteme. Daher kann beispielsweise die Steuerung einer Maschine nicht ohne Weiteres als App realisiert werden. Wann also sind Apps sinnvoll?

Die Stärke von mobilen Endgeräten und ihren Apps liegt in der Darstellung von Informationen, der intuitiven Nutzerinteraktion und der einfachen Vernetzung. Ein mögliches Anwendungsszenario: Die Display-Anzeige einer Maschine über eine App realisieren. Jeder Maschinenbediener hat sein eigenes Device, das gleichzeitig der Authentifizierung und Autorisierung dient; er klinkt dieses per USB-Halterung in die Maschine ein. Anschließend startet die App und der Nutzer kann die Maschine darüber bedienen. Ein weiteres denkba-

res Szenario: Der Service-Mitarbeiter, der mit seiner Service-App und einer Bluetooth- oder WLAN-Verbindung eine Diagnose der Maschine vornimmt, Parameter ausliest, verschickt und ändert. In allen Fällen kommuniziert die jeweilige App mit der Steuerungssoftware über eine geeignete Verbindung (Bluetooth, USB, WLAN, NFC) mittels eines Standard- oder eines proprietären Protokolls.

Für die Auswahl der genutzten Verbindung sind die Faktoren Übertragungssicherheit, Verbindungszuverlässigkeit, Reichweite und Exklusivität der Verbindung entscheidend – also ob die Verbindungsart für andere Funktionen verwendet wird. Ist der Nutzer zum Beispiel per WLAN mit der Maschine exklusiv verbunden, kann er nicht gleichzeitig mit dem Firmen-WLAN verbunden sein.

Die Hürden der App-Entwicklung

Eine der größten Herausforderungen bei der Realisierung von Apps ist die Auswahl der passenden Technologie. Während in den Anfängen der App-Entwicklung separate (native) Apps für jede einzelne Plattform (iOS, Android

und Windows Phone) entwickelt und gepflegt wurden, haben sich mittlerweile unterschiedliche Möglichkeiten zur Verwendung von plattformübergreifenden Lösungen etabliert. Der Grund hierfür hängt mit der Änderbarkeit und Wartbarkeit der App zusammen. Wenn für jede einzelne Plattform eine App existiert, muss jede Änderung, jede Neuerung und jede Fehlerbehebung für jede einzelne Plattform durchgeführt werden.

Das bringt eine ganze Reihe von Problemen mit sich. Üblicherweise werden bei komplexen Apps pro Plattform unterschiedliche Entwicklungsteams eingesetzt. Auch bei der Entwicklung gleicher Features auf den diversen Plattformen kommt es zu erheblichen Unterschieden. Zudem liegen der Architektur der Apps teilweise diverse Design-Patterns zugrunde. Während Android- und iOS-Apps am besten mit dem MVC-Pattern zu realisieren sind, kann eine Windows-Phone-App ebenso mit dem MVVM-Pattern erstellt werden.

All diese Einflussfaktoren können dazu führen, dass sich die App-Versionen für jede Plattform im Laufe der Zeit immer mehr voneinander entfernen. Die

(Bild: Sergey Nivens/Fotolia)



Pushing Performance



(Bild: scanrail/123RF)

Bei der App-Entwicklung gilt es nicht nur die unterschiedlichen Plattformen zu berücksichtigen. Ein weiterer wichtiger Aspekt ist die Kompatibilität mit den diversen Displaygrößen und Auflösungen der Endgeräte.

Folge: Drei Apps für jeweils eine Plattform anstatt einer App für drei Plattformen – und in Folge der dreifachen Änderungsaufwand. Und Änderungen kommen definitiv, denn Apps gilt es immer auf einem aktuellen Stand zu halten, da die Hersteller mobiler Plattformen einer Überalterung ihrer Plattform mit einer rigorosen Versionspolitik der Apps in ihren App-Stores entgegentreten. Ein Endlosbetrieb eines veralteten Systems, wie man ihn aktuell bei Windows XP sieht, wird es auf den aktuellen mobilen Plattformen nicht geben.

Der Einsatz plattformunabhängiger Technologien hat immer auch das Ziel, den Entwicklungs- und Pflege-Aufwand zu reduzieren und damit einen schnelleren ROI zu gewährleisten. Mobile Anwendungsprogramme im industriellen Umfeld sind vor allem auf die drei am meisten verbreiteten Plattformen iOS, Android und Windows Phone auszurichten. Nachfolgend werden drei gängige Lösungsmöglichkeiten zur plattformunabhängigen Implementierung von Apps vorgestellt und die jeweiligen Vor- und Nachteile diskutiert.

formunabhängigen Implementierung von Apps vorgestellt und die jeweiligen Vor- und Nachteile diskutiert.

Die Web-App

Eine Web-App ist ein plattformunabhängiges Anwendungsprogramm, das in einem Internet-Browser des Zielgeräts ausgeführt wird. Als Technologie für die Realisierung kommen hier in der Regel die aktuellen Web-Technologien HTML5, CSS und JavaScript zum Einsatz. Folglich kann eine Web-App auf allen Geräten laufen, die einen Internet-Browser mit HTML5-Unterstützung mitbringen. Die Darstellung der Web-App kann sich je nach Display-Format des Endgeräts anpassen. Die eigentliche Datenverarbeitung findet bei dieser Lösung jedoch nicht auf dem Endgerät, sondern auf einem Webserver statt, mit dem permanent kommuniziert wird. Daher ist auch der Programmieraufwand entsprechend gering, da es im Grunde nur um die optische Darstel-

HARTING Ha-VIS eCon Switches mit PoE+ Data und Power – perfekt kombiniert.



Wir stellen aus:
Husum
 15.-18. Sept. 2015
 Halle 4
 Stand C13

People | Power | Partnership

Die App-Versionen im Überblick	WEB APP	HYBRIDE APP	XAMARIN APP
Plattformübergreifende Benutzeroberfläche	X	X	-
Einbindung vieler Online-Inhalte	X	X	-
Plattformspezifische Designvorgaben	-	-	X
Überwiegend gemeinsame Programmiersprache	-	X	X
Anwendungslogik aufwändiger als Benutzeroberfläche	-	-	X
Extrem aufwändige Benutzeroberfläche ohne viel Anwendungslogik	X	X	-
Zugriff auf Funktionen des mobilen Geräts	-	X	X

Bei den Einsatzmöglichkeiten der vorgestellten Lösungen zeigt sich: Jede plattformübergreifende App hat ihre Vor- und Nachteile.

(Bild: Lachmann & Rink)

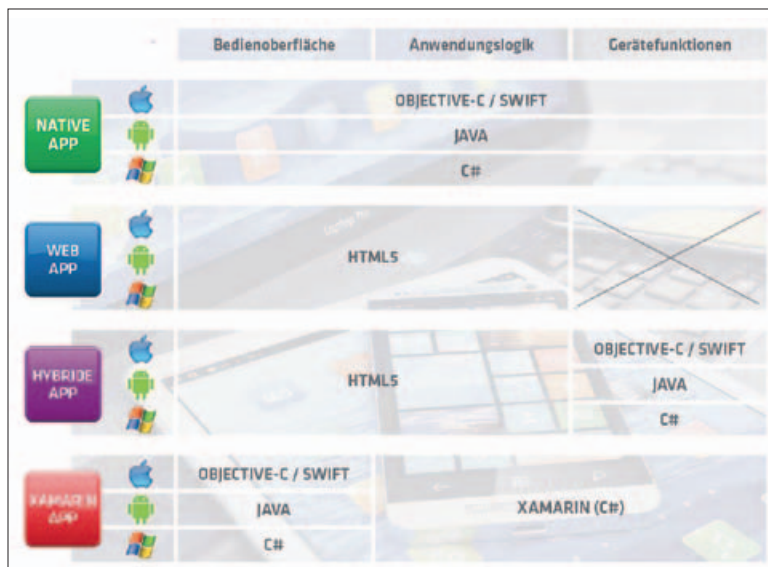
Flexible Energieversorgung direkt über Ethernet-Verbindungen.

Ha-VIS eCon Switches sind jetzt noch leistungsstärker: der IEEE PoE+ Standard ermöglicht neben der Übermittlung von Daten, die gleichzeitige Übertragung von Energie über ein Ethernet-Kabel. Bis zu vier Endgeräte lassen sich mit einer Leistung von jeweils 34,2 Watt versorgen – vom IP-Telefon über Kamerasysteme bis hin zu WLAN-Access-Points oder RFID-Readern. Switchvarianten mit integriertem Spannungswandler können direkt mit 24V statt der üblichen 54V versorgt werden. Und dank des intelligenten Energiemanagements senken eCon Switches mit PoE+ nicht nur den Verkabelungsaufwand, sondern auch Ihre Energiekosten.

Mehr erfahren Sie unter 0571 8896-0 oder mailen Sie an de@HARTING.com

Ein Überblick über die eingesetzten Technologien bei den unterschiedlichen Plattformen beziehungsweise App-Arten.

(Bild: Lachmann & Rink)



lung geht.

Web-Apps eignen sich gut für Anwendungen, bei denen kein Zugriff auf die Sensorik des Mobilgeräts nötig ist, die Oberfläche nicht den Designvorgaben der Plattform entsprechen muss oder wenn Inhalte gegebenenfalls auch außerhalb der mobilen Plattformen geteilt werden. Bekannte Beispiele sind Online-Shops oder soziale Netzwerke. Die Darstellung des Oberflächen-Designs wird bei Web-Applikationen durch CSS 3.0 erzeugt. Mit dieser Technologie kann die App selbst ihr Design entsprechend der Auflösung des Ziel-Displays verändern.

Hybride Apps

Die so genannten hybriden Apps kombinieren die Vorteile der Web-App mit denen der nativen. Die App bringt damit ihren eigenen kleinen Web-Server mit, der die Seiten bereitstellt. Zur Darstellung wird, wie bei der Web-App, die Webview-Funktion der jeweiligen mobilen Plattform verwendet. Für den Benutzer ist dieses Vorgehen transparent, da diese Funktion in die App eingebettet ist.

Einige der bekanntesten Frameworks für diese Variante sind Apache Cordova und Adobe PhoneGap. Beide Frameworks warten mit der Möglichkeit auf, HTML5-Seiten auf unterschiedlichsten Plattformen anzeigen zu lassen. Zusätzlich ermöglicht eine Reihe von zur Verfügung stehenden Plug-Ins den Zugriff auf die Sensoren des mobilen Gerätes (beispielsweise GPS oder Kamera) sowie auf Plattform-Funktionen (etwa der Zugriff auf das Dateisystem und die

Kontakte). Nicht vorhandene Funktionen können eigene Plug-Ins ergänzen.

Diese Variante der plattformübergreifenden App-Entwicklung eignet sich besonders für Anwendungen, die eigenen Design-Richtlinien folgen und auch Online-Inhalte darstellen sollen. Die Verteilung dieser Apps erfolgt wie bei den nativen Apps über die App Stores der unterschiedlichen Plattformen.

Die Xamarin-App

Eine andere Philosophie verfolgt die Xamarin-Technologie mit ihrem speziellen Programmiergerüst. Hier werden alle Bedienoberflächen jeweils für die Zielplattform entworfen und alle Zugriffe auf die Sensorik und die Plattformdienste sind ebenfalls für jede einzelne Plattform getrennt zu entwickeln.

Die eigentliche Logik der Anwendung und die Ansteuerung der Bedienoberfläche jedoch werden nur ein einziges Mal programmiert – durch geeignete Designpatterns scharf von der Bedienoberfläche getrennt. Xamarin eignet sich besonders für Apps, deren Anwendungslogik viele Aufgaben übernehmen muss, während die Oberfläche den Designvorgaben der jeweiligen Plattform entsprechen soll. Da die gesamte App in der Programmiersprache C# entwickelt wird, kann die Anwendungslogik auch für ein Windows-Programm verwendet werden. Nach dem Compilieren des C#-Codes mittels des Xamarin-Frameworks ist das Ergebnis ein echtes Java-Kompilat für Android, echter ARM-Binary-Code für iOS und echter MSIL-Code für Windows Phone. Anders als bei den vorgenannten Lö-

sungen sieht die App nicht nur aus wie eine native App, sie ist tatsächlich eine native App.

Ein mögliches Anwendungsszenario für Xamarin ist eine App für den Service-Techniker, der sich mit der Maschine verbindet und Parameter sowie Fehler auslesen, setzen und weiterverarbeiten möchte. Dies soll sowohl mittels mobilem Endgerät (iOS, Android oder Windows Phone) via Bluetooth oder USB als auch mit einem Laptop via WLAN oder RS 232 möglich sein. Die Implementierung des Protokolls und der Anwendungslogik erfolgt dann nur ein einziges Mal. Lediglich das eigentliche Senden und Empfangen von Daten muss pro Verbindung und Plattform separat implementiert werden. Selbiges gilt für alle plattformabhängigen Funktionen wie zum Beispiel E-Mails senden oder Dateien speichern. Diese Komponenten sind dann aber App-unabhängig, da sie keine Logik beinhalten, und können immer wieder in anderen Apps verwendet werden. Außerdem muss der Entwickler die Oberfläche für jede Plattform gestalten. Die Methoden zum Erstellen der Bedienoberflächen sind dieselben wie beim Erstellen einer nativen App.

Reduzierung des Aufwands

Der Entwicklungs- und Pflegeaufwand von Apps für mobile Lösungen wird durch plattformübergreifende App-Entwicklung im Vergleich zu der jeweils nativen Entwicklung pro Zielplattform deutlich reduziert. Je nach Rahmenbedingungen sollte vor der Entwicklung die am besten passende Lösung für die plattformunabhängige Realisierung gewählt werden. Wie so oft gibt es hierfür kein Patentrezept, denn die am besten geeignete Technologie ist abhängig von verschiedenen Randbedingungen. Wichtig für eine fundierte Entscheidung sind der Zweck und das Umfeld, in dem die App eingesetzt werden soll. *ld*



Martin Heilmann

ist im Bereich App- und .NET-Entwicklung bei Lachmann & Rink tätig.