

DIE TOP 5-MARKTTRENDS UND IHRE AUSWIRKUNGEN AUF SOFTWAREUNTERNEHMEN IN 2020/2021

BY STEPHEN BALL - OCTOBER 2020

TABLE OF CONTENTS

Einführung	2
Trend 1: Das Ende der 32-Bit-Versionen.....	2
Technische Umsetzung	3
Trend 2: Bildschirme mit hoher Auflösung	4
Unterstützung mehrerer Monitore	5
Technische Umsetzung	6
Trend 3: Sandbox-Technik, App Stores und Notarisierung.....	7
Warum ist dies für die Zukunft wichtig?.....	8
Technische Umsetzung.....	8
Trend 4: „Priorität für“ und Unterstützung durch mobile Apps	10
Probleme bei der Auswahl der Plattform?	10
Technische Umsetzung.....	11
Trend 5: Architekturen für Interkonnektivität.....	12
Konnektivität bei der mobilen Nutzung	12
Direkte lokale Verbindungen.....	13
Technische Umsetzung	14
Zusammenfassung	15
Weitere Anmerkungen	15

EINFÜHRUNG

Die größte Herausforderung für jedes Softwarehaus ist es, seine Kunden möglichst schnell mit einer Lösung zu bedienen, die allen Erwartungen in Bezug auf Bedienbarkeit, Flexibilität und Leistung entspricht und somit eine optimierte Benutzererfahrung liefert. Dafür gilt es zuerst zu ermitteln, auf welchen Kundengeräten die Software zum Einsatz kommen soll, um dann die Anwendungen und Architekturen für diesen Geschäftsbereich zu entwickeln.

Um eventuellen Missverständnissen vorzubeugen: In diesem Papier geht es nicht um interne Praktiken der Softwareentwicklung. Unabhängig von Ihrem jeweiligen Entwicklungsansatz ist das Ziel nämlich immer das gleiche: Sicherzustellen, dass die Anwendungen kompiliert und ausgeführt werden können, dass sie sicher sind und dass sie auf den Kundensystemen auch optisch gut zur Geltung kommen. Diese Papier wird Ihnen Markttrends aufzeigen, die für die internen technischen Fähigkeiten zur Bereitstellung der Software für Endbenutzer wichtig sind.

Neue Risiken für vorhandene Produkte

Die neuesten Entwicklungen in der technologischen Landschaft haben zur Folge, dass auch funktionierende Anwendungen dringend Updates zur Aufrechterhaltung ihrer Funktionalität und der Benutzererfahrung benötigen.

Wichtige Änderungen in Bezug auf Plattformen, Hardware- und Softwarearchitekturen machen es für Entwickler unumgänglich, zentrale Anwendungen anzupassen, um auch nur den aktuellen Stand der Benutzererfahrung sicherzustellen. Das bedeutet gleichzeitig, dass die internen technischen Entwicklungsfähigkeiten aktualisiert werden müssen, um die gewohnte Benutzererfahrung gewährleisten zu können.

Wir analysieren dafür in diesem Papier nicht nur die zentralen Softwaretrends, sondern möchten auch die Frage beantworten, wie diese Herausforderungen schnell und mit geringem Risiko für vorhandene Anwendungen bewältigt werden können.

TREND 1 - DAS ENDE DER 32-BIT-VERSIONEN

Viele Dinge, an die sich Entwickler als scheinbar gesichert gewöhnt haben, haben sich in den vergangenen Jahren geändert. Seit der Einführung von Windows NT im Jahr 1993 hat es immer eine 32-Bit-Edition von Windows gegeben. Im Zuge der weiteren Entwicklung wurde die 32-Bit-Version vor über 20 Jahren mit Windows 95 und 98 schließlich populär. Die neueren Versionen von Windows unterstützen 32-Bit-Plattformen zwar weiterhin, der primäre Fokus liegt aber auf den nun dominierenden 64-Bit-Prozessoren.

Während Microsoft bemerkenswerterweise weiterhin zwei Betriebssysteme in einem (32 Bit und 64 Bit) ausliefert, beginnen die übrigen Marktteilnehmer sich langsam von 32 Bit zu verabschieden und nur noch für 64-Bit-Plattformen zu entwickeln.

In den beiden vergangenen Jahren sind 64-Bit-Anwendungen – auch wenn sie letztlich auf Geräten mit weniger als 4 GB Arbeitsspeicher nicht unbedingt erforderlich sind – zum Quasi-Standard für Geräte geworden. Apple hat mit iOS den Übergang zu 64 Bit beschleunigt und es als neue technische Errungenschaft vermarktet. Android folgte auf dem Fuß. Dies hat auch zu neuen Anforderungen für die App Store-Bereitstellung geführt, für die nun 64-Bit-Versionen erforderlich sind.

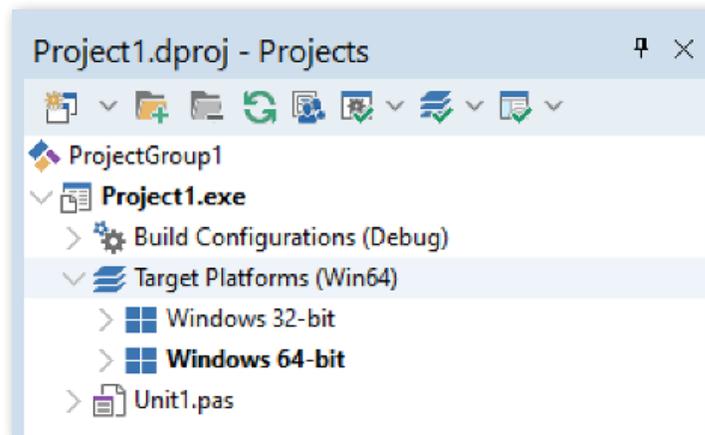
Die logische Konsequenz, die sich daraus langfristig ergibt, ist die Beschränkung der Betriebssysteme und die Einstellung des 32-Bit-Codes, um die Plattformverwaltung zu vereinfachen und die Zielsysteme zu verringern, für die Entwickler programmieren müssen. Zyniker könnten argumentieren, dass es wohl kurzfristig das Ziel gewesen ist, damit den Hardwareumsatz anzukurbeln, da die Kunden quasi gezwungen waren, ihre Mobilgeräte zu aktualisieren.

Im Desktop-Bereich können nun mit den neuesten Apple-Betriebssystemen NUR noch 64-Bit-Anwendungen ausgeführt werden. Der letzte 32-Bit-Laptop wurde ja bereits 2011 ausgeliefert. Dies hat den Übergang der Anwendungsumgebungen zu 64 Bit im Lauf der Zeit beschleunigt. Die Zeit wurde so insgesamt reif für diesen grundlegenden Wandel. Auch wenn Windows immer noch vielfältigere Hardwareumgebungen berücksichtigen muss, tickt die Uhr unaufhaltsam für die letzte 32-Bit-Version von Windows.



TECHNISCHE UMSETZUNG

Sollten Sie noch nicht umgestellt haben, dann ist es nun an der Zeit, Ihre Anwendungen auf 64 Bit zu erweitern. RAD Studio bietet die Möglichkeit, eine einzige Codebasis für 32 Bit und 64 Bit zu nutzen, wobei für die meisten 32 Bit-Anwendungen nur als eine neue Zielplattform mit anschließender Kompilierung hinzugefügt werden muss.



Unabhängig von der Entwicklungssprache müssen dabei auch externe Packages aktualisiert werden. Prüfen Sie deshalb die Verfügbarkeit von 64-Bit-Versionen für alle externen Bibliotheken, die Ihre Software derzeit nutzt, um für den Übergang vorbereitet zu sein.

Sehr hilfreiche Materialien für diesen Übergang finden Sie im Upgrade- und Migrations-Center von Embarcadero unter <https://www.embarcadero.com/rad-in-action/migration-upgrade-center>



TREND 2 - BILDSCHIRME MIT HOHER AUFLÖSUNG

Die Prozessorleistung ist nicht der einzige Bereich, in dem sich in den vergangenen Jahren viel geändert hat. Die Hardware zur Visualisierung, die das Computing auf Desktop-Geräten, Laptops und Smartphones unterstützt, hat sich in kürzester Zeit in Richtung hochauflösende Monitore und Verwendung mehrerer Bildschirme gleichzeitig weiterentwickelt.

Da die Monitore größer und hochauflösender geworden sind, ist nun ein Punkt erreicht, wo die Skalierung von jeder einzelnen Anwendung selbst verwaltet werden muss.

Was ist ein Retina-Display?

„Retina-Display ist ein von Apple eingeführter Marketingbegriff, der sich auf Geräte und Monitore bezieht, die über eine so hohe Auflösung und Pixeldichte – ca. 300 oder mehr Pixel pro Zoll auf mobilen Geräten– verfügen, dass ein Benutzer die einzelnen Pixel bei einem normalen Abstand nicht mehr erkennen kann.“

Apple hat seine ersten Retina-Displays bereits im Jahr 2011 auf den Markt gebracht. Die ersten entsprechenden Android-Geräte folgten nur kurze Zeit später. Retina-Displays haben bewirkt, dass Monitore für klarere und schärfere Grafiken auf dem Markt populär wurden. So wurden z. B. 4K-Monitore mit Ultrahochauflösung entwickelt, mit denen jeweils mehr auf dem Bildschirm angezeigt werden kann bzw. bei gleicher Schriftgröße die Schrift schärfer angezeigt werden kann.

Für Entwickler bedeutet dies, dass die Grafiken und Icons in den Anwendungen klarer und schärfer sein müssen. Traditionell werden Icons im Format 16x16 Pixel angezeigt. Wenn nun die Auflösung der Bildschirme höher und weitgehend dynamisch zwischen den Monitoren wird, erscheinen die Symbole in vielen Anwendungen immer kleiner oder unscharf. Das hat zur Folge, dass der traditionelle Standard von 96 DPI nicht mehr der allgemein gültige ist.

Leider lassen sich aber die alten Symbole nicht einfach durch größere Grafiken ersetzen. Teilweise werden etwa größere Grafiken nicht korrekt herunterskaliert. Deshalb muss für eine optimale visuelle Qualität ein dynamischer Wechsel zur Laufzeit implementiert werden.

UNTERSTÜTZUNG MEHRERER MONITORE

In dem Maße, in dem die Monitore immer größer und klarer in der Darstellung werden, geht die Entwicklung auch in Richtung auf die Nutzung mehrerer Monitore gleichzeitig. Zwei Monitore für eine Workstation sind nicht mehr ungewöhnlich, oft sind es sogar drei, wenn noch ein Laptop in einer Docking-Station angeschlossen ist. Durch diese Innovation im Bereich der Monitore und deren Nutzung werden mittlerweile oft verschiedene Monitore mit unterschiedlichen Auflösungen verwendet, die für die Ausgabe der Inhalte eines PCs kombiniert werden.

Die Herausforderung für Entwickler besteht nun darin, dass durch Nutzung einer Anwendung auf einem Gerät mit mehreren Monitoren und jetzt auch mit unterschiedlichen DPI-Werten die Anwendungsformulare von Monitor zu Monitor unterschiedlich aussehen können. Hinzu kommt, dass die endgültige Handhabung dieser Abweichung im Moment noch offen ist. Microsoft hat z. B. bereits die zweite Haupt-API zur Verarbeitung mehrerer Monitore herausgebracht. Unterschiedliche Versionen von Windows 10 unterstützen dieses Feature deshalb in jeweils anderer Weise. Dies muss ebenfalls berücksichtigt werden.

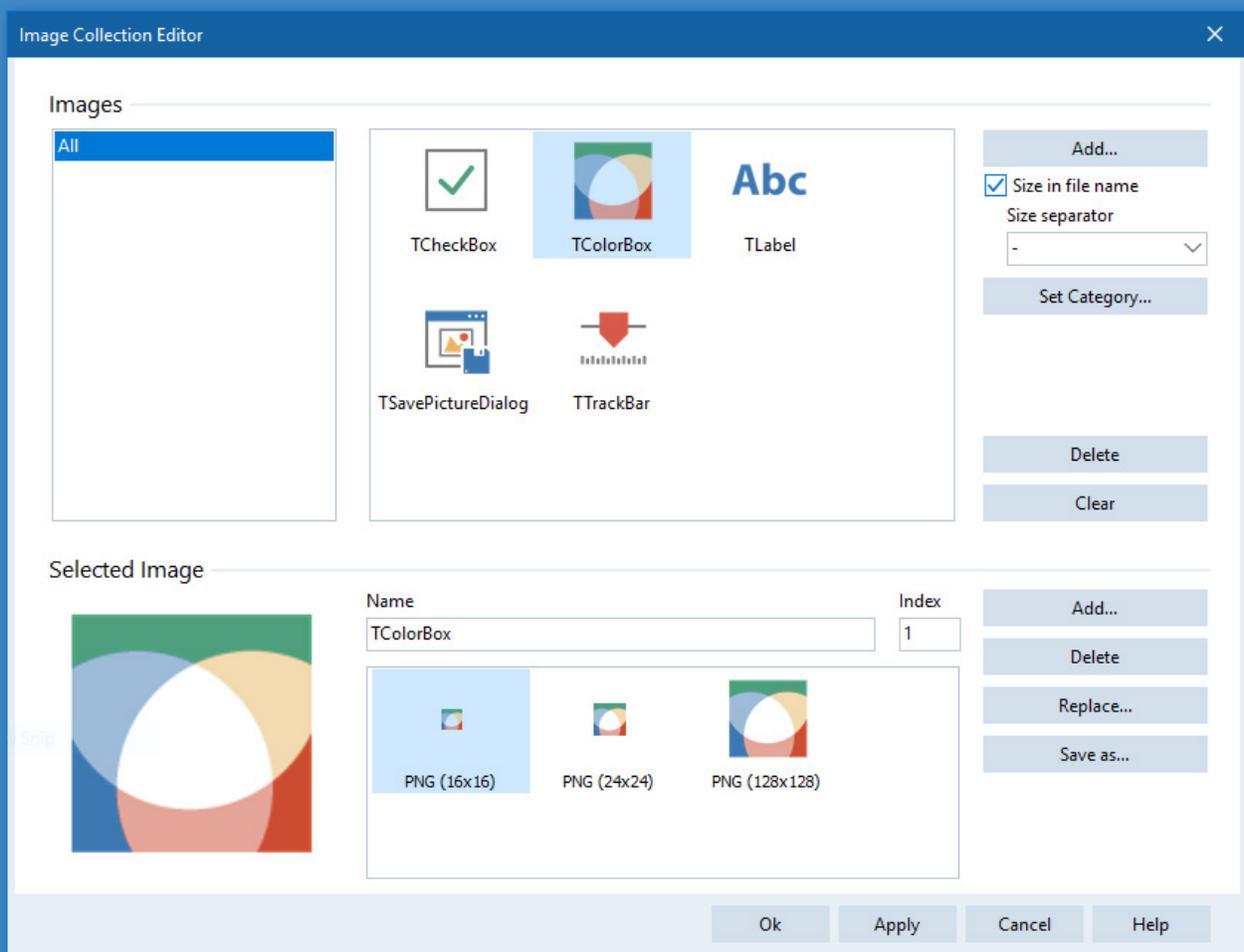
Als Konsequenz unterstützen viele Anwendungen die Skalierung nicht korrekt, was aus Benutzersicht in vielen Anwendungen eine suboptimale Darstellung und Benutzererfahrung nach sich zieht.

Eine ausführliche Darstellung dieses Problems finden Sie im interessanten Blog-Beitrag [The Perils of using a 4k monitor](#). (Risiken bei der Nutzung von 4K-Monitoren).

TECHNISCHE UMSETZUNG

Die traditionellen Windows API-Aufrufe wurden bereits vor vielen Jahren entwickelt, bevor die Monitornutzung von heute auch nur denkbar war. Das hat zur Folge, dass die Anwendung von APIs für Windows-Steuererelemente in der Welt der Mehr-Monitor-Nutzung und Mehrfachauflösung zu einer kniffligen Aufgabe wird. Für viele Entwickler ist dies eine enorme Herausforderung.

RAD Studio-Kunden haben nun mit der VCL das perfekte Instrument, dieses Problem reibungslos zu bewältigen. Die neue Komponente TVirtualImageList bietet eine einfache Alternative zur bisherigen Komponente TImageList. Sie stellt die Kompatibilität zu traditionellen APIs sicher und ermittelt gleichzeitig die korrekte Skalierung für die optimale Grafikdarstellung je nach für den Bildschirm erforderlicher Auflösung. Dies garantiert durchgehend maximale Qualität, Geschwindigkeit und Leistung in Ihrer Anwendung. Gleichzeitig wird damit ein einfacher Migrationspfad zur Unterstützung von Windows 10 für ältere Quellcodes bereitgestellt und zwar auch für solche mit überwiegend Drittanbieter-Komponenten.



TREND 3 - SANDBOX-TECHNIK, APP STORES UND NOTARISIERUNG

Jeder kennt heutzutage App Stores, ob mit oder ohne Mobiltelefon. Über App Stores wie den Apple AppStore, Google Play Store und Amazon App Store werden die Apps für die Geräte bereitgestellt. Einerseits sind sie als vertrauenswürdige Quelle von Inhalten für das Erschließen des Massenmarkts durch Softwarehäuser unverzichtbar. Andererseits beschränken sie die Möglichkeiten der App-Bereitstellung für Zielplattformen. Beispielsweise können Sie eine mobile App nicht einfach kopieren und einfügen.

Als Marktplatz sind App Stores einem gewissen Risiko ausgesetzt, mit dem sie umgehen müssen. Zur Minimierung dieses Risikos wird die Sandbox-Technik für Apps zum Schutz von Systemen und Benutzern genutzt. Dabei werden die Rechte einer App auf die vorgesehene Funktionalität begrenzt. Dies erhöht die Hürden für Schadsoftware im Hinblick auf die Gefährdung eines Benutzersystems.

Die Sandbox-Technik ist eine zentrale Voraussetzung für die Bereitstellung in App Stores und in vielen Betriebssystemen bereits standardmäßig enthalten.

Neben der Sandbox-Technik ist die Notarisierung ein weiterer Trend, der von Apple zur Förderung von macOS forciert wird. Bei der Notarisierung werden Anwendungen von Apple geprüft, aber dann vom Anwendungsentwickler vertrieben. Damit können Unternehmensanwendungen ohne einen öffentlichen App Store schneller und einfacher auf den Markt gebracht werden. Mit der letzten macOS-Version (10.15, Catalina) ist die Bereitstellung von Apps, die nicht notarisiert sind, außerhalb eines App Stores schwieriger geworden, da dafür mehrere manuelle Eingriffe erforderlich sind. Das ist für die Benutzererfahrung nicht gerade förderlich.

WARUM IST DIES FÜR DIE ZUKUNFT WICHTIG?

Der Marktplatz für mobile Apps hat zu einer Vielzahl von Innovationen in Bezug auf die mögliche Interaktion der Benutzer mit Anwendungen geführt, aber auch im Hinblick auf die Durchführung von Aufgaben der Anwendungsverwaltung. Eine dieser Aufgaben ist das Scannen auf Viren. Ein zentrales Ziel von Microsoft zur Verringerung der Anforderungen an den CPU-Zyklus ist die Reduzierung des Stromverbrauchs und die Unterstützung von grünen IT-Initiativen.

Virenprüfungen sind Bestandteil des Bereitstellungsverfahrens in einem App Store. Wenn ein App-Paket entsprechend gescannt ist, kann überwacht werden, ob die Dateien außerhalb des Anwendungspakets geändert werden. Dies verringert die Notwendigkeit einer regelmäßigen Prüfung auf Viren. Die Durchführung einer Aufgabe vor der Bereitstellung und in einer Cloud reduziert die Anforderungen an die CPU eines Geräts erheblich.

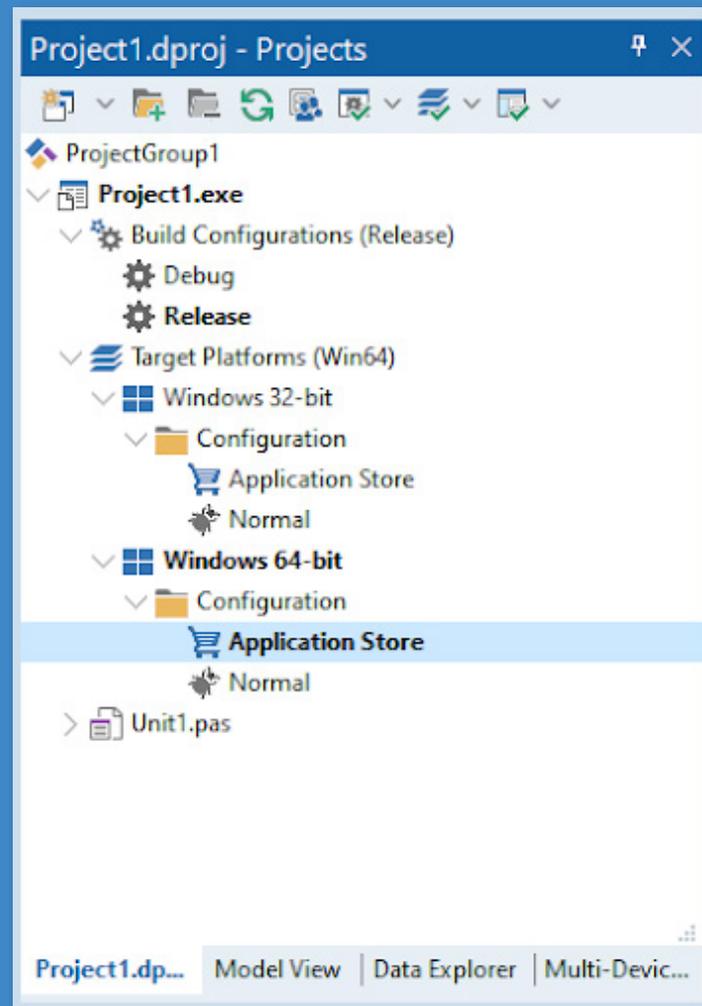
In diesem Zusammenhang haben einige Studien gezeigt, dass bei der Virenprüfung bis zu einem Viertel des CPU-Zyklus auf einem einzelnen Computer in Anspruch genommen wird. Man stelle sich nur vor, was möglich ist, wenn für Ihre Computer-CPU plötzlich zusätzlich 20 % bis 25 % der Kapazität zur Verfügung steht. Diese Entwicklung wird auch durch Cloudanbieter forciert, die ihren PUE-Wert für die Effektivität des Stromverbrauchs (PUE, Power Usage Effectiveness) ihres Rechenzentrums zur Kosteneinsparung verbessern möchten.

Es ist deshalb nicht überraschend, dass Microsoft sein Windows 10-APPX-Paket für Anwendungen zur Verbesserung der Benutzererfahrung und zur Erhöhung des Nutzens für die einzelnen Windows-Plattformen vorantreibt.

TECHNISCHE UMSETZUNG

Benutzer von RAD Studio sind mit der neuesten IDE bereits in der Lage, auf diesen Wandel zu reagieren. RAD Studio war die erste IDE mit integrierter APPX-Paketerstellung. Sie ermöglicht den Zugang zum Windows 10 Store und zur Notarisierung unter Windows. Anwendungen werden in einer Sandbox-Umgebung installiert und ausgeführt.

Entwickler benötigen damit zur Paketerstellung und zur Vorbereitung von Anwendungen für die Bereitstellung in Windows-, macOS-, iOS- und Android-Stores nur die IDE als einziges Tool. Diese beinhaltet auch die Notarisierung von Anwendungen für macOS. Zusammenfassend lässt sich feststellen: Durch Vereinfachung dieser vielfältigen, komplexen Prozesse wird Entwicklungszeit eingespart und die Komplexität im Entwicklungszyklus reduziert, um eine herausragende Benutzererfahrung in schnellst möglicher Zeit gewährleisten zu können.



TREND 4 - „PRIORITÄT FÜR“ UND UNTERSTÜTZUNG DURCH MOBILE APPS

Es gibt mittlerweile mehr Mobilgeräte, die mit dem Internet verbunden sind, als traditionelle Computergeräte. Deshalb ist es kaum überraschend, dass in dieser hochgradig vernetzten Welt mehr Anwendungen zuerst als mobile Apps auf dem Markt kommen bzw. durch mobile Apps unterstützt werden.

Auch wenn sich in diesem Kontext die Benutzererfahrung immer mehr in Richtung der Funktionalität für hochgradig vernetzte Anwendungen weiterentwickelt, heißt das nicht, dass die gesamte Funktionalität auf mobile Nutzung abgestimmt sein muss. Wir werden auf den Aspekt der Vernetzbarkeit für diese Problemstellung in Trend 5 noch ausführlich eingehen, da es je nach Anforderung verschiedene Optionen gibt. Zunächst soll es aber weiter um das zentrale Thema dieses Papiers gehen, um die Benutzererfahrung.



PROBLEME BEI DER AUSWAHL DER PLATTFORM?

In Kürze: Welche mobilen Plattformen nutzen Kunden? Weltweit gibt es hier Unterschiede, aber als Faustregel gilt: 40 % aller Geräte laufen unter Android, 40 % unter Windows und der Rest unter iOS, macOS sowie Linux. Android ist zwar die populärste mobile Plattform, es gibt aber eine signifikante Anzahl von iOS-Benutzern, vor allem bei den hochwertigen Geräten.

Letztlich werden Sie wahrscheinlich Ihre Anwendung sowohl für Android wie für iOS entwickeln. Wenn die Plattformen geklärt sind, für die entwickelt werden soll, müssen für die Benutzererfahrung auf Mobilgeräten u. a. folgende zentrale Aspekte berücksichtigt werden:

- Es muss gewährleistet sein, dass Funktionalitäts-Updates auf allen Plattformen synchron erfolgen.
- Benutzer von Mobilgeräten verwenden aufgrund der einfachen Nutzung von App Stores meist eine Vielzahl von Apps. Um hier erfolgreich sein, ist eine optisch ansprechende App unverzichtbar. App-Designer können dafür eine wichtige Hilfe sein.
- Native Apps sind in App Stores erfolgreicher, da sie eine bessere Leistung und eine umfangreichere Benutzererfahrung als Webanwendungen bieten.

FÜR ENTWICKLER

- Sicherheit muss ein zentraler Aspekt für alle personenbezogenen Daten sein, wenn Sie sich nicht dem Risiko von DSGVO-Strafzahlungen anteilig zum weltweiten Umsatz Ihres Unternehmens aussetzen möchten.
- Bei der Entwicklung für mehrere Plattformen müssen Sie potenziell viele verschiedene APIs, Komponenten und Visualisierungs-Frameworks verwenden, damit Ihre Apps auch wirklich eine herausragende Qualität haben.
- Mehrere Plattformen erfordern mehrfache Tests und ziehen enge Zeitpläne für den Marktauftritt nach sich.

Die naheliegende Lösung besteht hier darin, die Entwicklung nativ aus der Perspektive der Benutzererfahrung und der Sicherheit anzugehen. Das bedeutet aber in der Regel, dass Entwickler sich weiterbilden und sich Fertigkeiten für mehrere Plattformen aneignen müssen. Dies erhöht potenziell die Kosten, bedeutet mehr Zeitaufwand und steigert das Risiko.

TECHNISCHE UMSETZUNG

Delphi bietet Entwicklern die Möglichkeit, mit einer einzigen Codebasis für iOS wie für Android zu entwickeln und dafür mithilfe des FMX-Frameworks auch native Steuerelemente zu nutzen. FMX (Abkürzung für FireMonkey) ist eine plattformübergreifende Version der VCL, dem besten Framework für die native Windows-Entwicklung. Entwickler von Windows-VCL-Anwendungen benötigen deshalb dafür in der Regel keine Einarbeitung.

FMX wurde auf Basis der objektorientierten Programmierung komplett neu entwickelt. Es basiert weitgehend auf Interfaces, sodass Entwickler für die Komponenten und Schnittstellen programmieren können, mit denen die plattformspezifischen Objekte zur Laufzeit bereitgestellt werden.

Das Ergebnis sind leistungsstarke native Anwendungen, die von Grund auf plattformübergreifend sind. Sie können schneller entwickelt werden und verfügen über eine höhere Leistung, da sie sozusagen „hinter den Kulissen“ direkt mit den plattformspezifischen APIs kommunizieren.

Durch Reduzierung des erforderlichen Testaufwands und Verwendung nur einer einzigen Codebasis werden auch die Gesamtkosten verringert und die Kundenreichweite pro Codezeile wird erhöht. Darüber hinaus können Prototypen erstellt und sofort im Produktionscode genutzt werden – mit plattformspezifischen Details, die automatisch verwaltet werden, z. B. die Position von Registerkarten auf der jeweiligen Plattform (oben oder unten). Anwendungsdesigner haben dabei auch die Möglichkeit, das Erscheinungsbild ansprechend zu gestalten. Sie können diese Gestaltung dann sofort für die einzelnen Plattformen prüfen, sowohl innerhalb der IDE als auch mithilfe von FireUI auf den physischen Geräten.

Darüber hinaus lässt sich nicht visueller Code von VCL-Anwendungen meist gemeinsam mit FMX-Anwendungen nutzen. Diese Möglichkeit der unmittelbaren Übertragung der Geschäftslogik beschleunigt zusätzlich den Marktauftritt und verringert die Risiken von doppeltem Code. Dabei sind auch Komponenten der FireDAC-Datenbankkonnektivität enthalten. Mit der kostenfreien Einbindung von InterBase ToGo in die Enterprise-Version von RAD Studio (für iOS und Android) ist eine starke 256-Bit-Verschlüsselung für eine plattformübergreifende Datenspeicherung zur Einhaltung der DSGVO verfügbar.

TREND 5 - ARCHITEKTUREN FÜR INTERKONNEKTIVITÄT

Da komplementäre mobile Apps immer mehr zum Standard werden, ist die Architektur für die Datenkonnektivität zwischen vorhandenen Desktop-/Unternehmenslösungen und neuen mobilen Apps ein wichtiger Aspekt.

Beim Thema Interkonnektivität stellt sich als Erstes die Frage nach dem Konnektivitätstyp, der für die Unterstützung der gewünschten Benutzererfahrung erforderlich ist. Konnektivität betrifft zwei zentrale Bereiche:

- 1. Konnektivität bei der mobilen Nutzung**, also wie werden Daten über einen zentralen Server weitergegeben und abgerufen.
- 2. Direkte lokale Konnektivität**, also wie können Desktop-Anwendungen und mobile Apps Daten auf schnelle Weise austauschen.

KONNEKTIVITÄT BEI DER MOBILEN NUTZUNG

Anders als bei der traditionellen Client-/Server-Konnektivität, bei der die Geräte sich in einem lokalen Netzwerk befinden, tauschen mobile Lösungen Daten außerhalb der Firewall aus. Dies rückt die Logikschicht als mögliche Brücke zwischen den vertrauenswürdigen und den nicht vertrauenswürdigen Domänen in den Mittelpunkt.

Der wichtigste Trend bei der Logikschicht ist die weit verbreitete Nutzung von RESTful-Servern, die eine zentrale API bieten. REST ist deshalb so populär, weil damit jede Anfrage nach Anforderung verarbeitet werden kann. Dabei wird ein Standardansatz verfolgt, der von jeder Programmiersprache mit minimalem Aufwand und wenig Ressourcen genutzt werden kann.

Für die Verarbeitung großer Datenmengen bieten REST APIs nicht nur eine ressourcenintensive Unterstützung für Sitzung und Status (der RESTful-Aufruf ist nur temporär). Alles, was für die Verarbeitung der Anfrage erforderlich ist, also auch der Status, wird außerdem mit dem API-Aufruf als Parameter übergeben. Aufrufe an REST-Dienste sind unkompliziert und schnell. Sie bieten die Möglichkeit, Server für eine große Anzahl an Benutzern zu skalieren, da nicht alle angeforderten Daten zum selben Zeitpunkt verarbeitet werden. Dies reduziert den Ressourcenbedarf serverseitig auf ein Minimum.

Bei der Bereitstellung gibt es schließlich einen weiteren Trend für RESTful-Dienste: die Nutzung von Containern wie z. B. Docker. Dabei ist die erforderliche serverseitige Software vorkonfiguriert und kann innerhalb eines Container-Images ausgeführt werden. Dies vereinfacht Bereitstellung, Tests und Veröffentlichung der Software, da alles innerhalb eines Stacks ausgeführt wird.

Container bieten auch die Möglichkeit, plötzlich auftretende Spitzen der Datenanforderung auf einfache Weise zu verarbeiten. Dafür wird dann eine weitere Container-Instanz erstellt und die Arbeitslast darauf verteilt. IaaS-Anbieter (Infrastructure as a Service, Infrastruktur als Dienst) wie Amazon und Azure ermöglichen ein flexibles automatisches Herauf- und Herunterskalieren, das sich durch komplette Automatisierung einfach durchführen lässt. Damit können die Kosten erheblich reduziert werden, da nur Hardware für den aktuellen Bedarf bereitgestellt werden muss und Gebühren lediglich für den zusätzlichen Bedarf, wenn erforderlich, anfallen.

REST (in der Regel JSON-basiert) ist das zentrale Element des heutzutage populären Konzepts der Mikrodienste, bei dem jede API eine bestimmte Aufgabe ausführt. Zusammen bieten sie dann auf schnelle Weise eine umfangreiche Funktionalität für die Endbenutzer. JSON ist die am weitesten

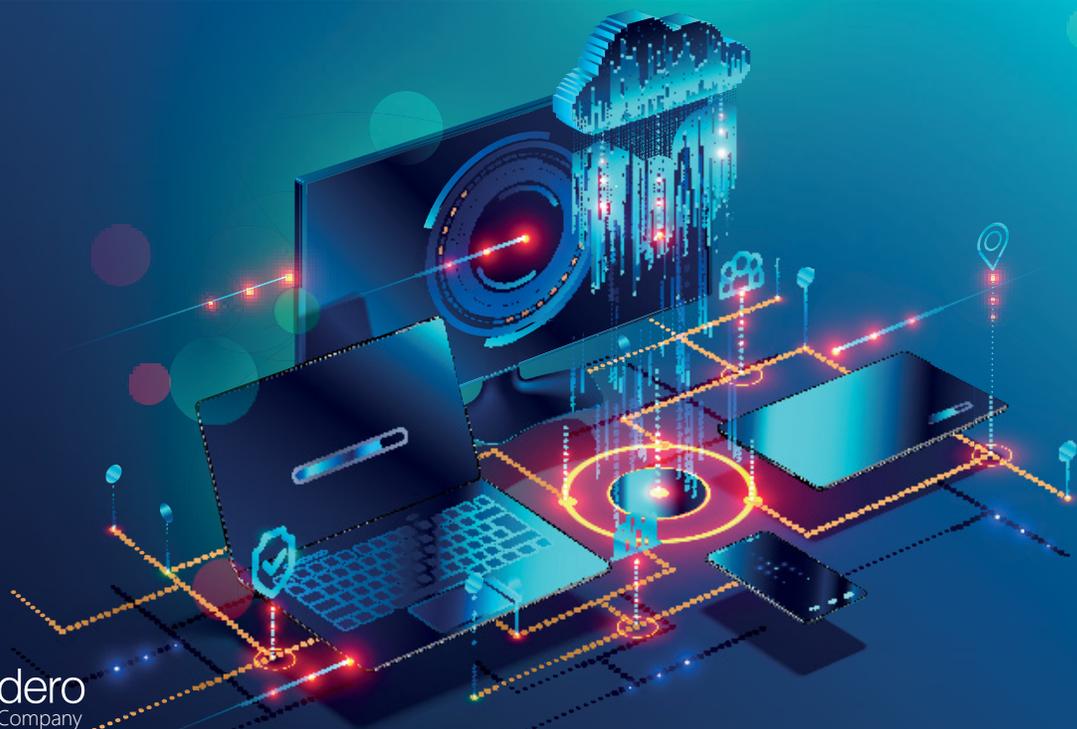
verbreitete Struktur zur Datenübertragung, da sie einerseits einfach, ressourcenarm (es werden wenige Daten genutzt) und für jede Programmiersprache problemlos zu lesen ist und andererseits ausreichend Flexibilität für Änderungen bietet.

Eine zentrale Aufgabe bei der Nutzung von REST-Diensten ist das Erfassen von vorgenommenen Änderungen. Diese können einerseits mit Push-Benachrichtigungen vom Server übermittelt werden, wobei dies nicht bei jeder Änderung erfolgen sollte. Dazu fügen Sie zusätzliche Änderungsprotokolle hinzu, die der Client dann abfragen kann. Oder Sie rufen den gesamten Datenbestand ab, was aber bei umfangreichen Datenbeständen nicht optimal ist.

DIREKTE LOKALE VERBINDUNGEN

Eine weitere Möglichkeit des Datenaustauschs ist die direkte gemeinsame Nutzung von Inhalten auf einem Mobilgerät und in einer geöffneten Desktop-Anwendung. Dies eignet sich optimal für die Nutzung der Mobilgerät-Features wie GPS-Ortung, Kamera usw., um Daten direkt für die geöffnete Desktop-Anwendung bereitzustellen. Es ist ideal, wenn sofortige Ergebnisse benötigt werden, da die Daten lokal verfügbar sind und keinen zentralen Server erfordern.

Ein gutes Beispiel ist dafür ein Arzt, der mit seinem Smartphone ein Foto von einem Patienten bei einer Sprechstunde macht. Das Foto wird auf dem Smartphone angezeigt und kann dort geprüft werden, wird aber aus Datenschutzgründen nicht in der Galerie des Geräts gespeichert. Stattdessen wird das Foto direkt in eine geöffnete Anwendung auf dem Computer übernommen, um die Krankenakte des Patienten zu ergänzen.



TECHNISCHE UMSETZUNG

Für REST bietet der RAD Server als Teil der Enterprise-Edition und der Architect-Edition eine attraktive MEAP-Plattform (Mobile Enterprise Application Platform) mit einer Reihe von Produkten und Diensten, die für die Entwicklung mobiler Apps und für einen schnellen Marktauftritt genutzt werden können. Dazu gehören u. a. Benutzersicherheit, Push-Benachrichtigungen und Beacon Fencing. Durch die Docker-Unterstützung mit genehmigten Basis-Images wird die Bereitstellung beschleunigt und vereinfacht. Sie erfolgt aufgrund der leistungsstarken Komponenten, mit denen die umfangreiche Anpassung von JSON verwaltet wird, in Sekundenschnelle. Mit leistungsstarken Integrationsoptionen für das Hosten von Websites in RAD Server lässt sich die Bereitstellung außerdem schnell und sicher durchführen. Dies gilt vor allem in Verbindung mit Sencha Architect, einer Komponente der Architect-Edition.

Mit dem für IoT preisgekrönten InterBase-Feature der Change Views lassen sich vorhandene Systeme und Datenbanken mit einer protokollfreien, hochskalierbaren Änderungsnachverfolgung erweitern. Diese ist sehr hilfreich für die Datensynchronisierung bei einer großen Anzahl an Remote-Geräten.

Für direkte Verbindungen bieten die App-Tethering-Komponenten von RAD Studio leistungsstarke Funktionen, die vorhandenen Anwendungen hinzugefügt werden können, um Daten von unterstützenden Anwendungen (Desktop oder mobil) zu teilen oder zu erfassen. Dies lässt sich mit automatischer Ermittlung und Passwortschutz für eine integrierte Sicherheit sehr einfach implementieren. Auch ist die Aktualisierung und gemeinsame Nutzung von Inhalten so unkompliziert wie die Festlegung einer Eigenschaft. Die Komponenten stellen das entsprechende Ereignis in der unterstützenden Anwendung automatisch zur Verfügung und lösen es aus.



ZUSAMMENFASSUNG

In diesem Papier haben wir die 5 Top-Trends erörtert sowie das, was Sie heute schon für eine Modernisierung und Vorbereitung auf die Zukunft tun können. Für ältere Quellcodes sind einige Änderungen notwendig, damit Sie die neuesten zeitsparenden Features nutzen können. Aufgrund der herausragenden Rückwärtskompatibilität von RAD Studio sind Zeitaufwand, Kosten und Risiken, die damit verbunden sind, gering.

Der beste Weg, um aus diesen branchenweiten Änderungen für sich Kapital zu schlagen, ist, den Wandel für Innovation zu nutzen. RAD Studio, die VCL, FireMonkey und RAD Server bieten die Möglichkeit, auf schnelle Weise den vorhandenen Code ertragsfördernd zu optimieren und mit den bestehenden Anwendungen neue Einnahmequellen zu erschließen. Machen Sie Ihre Anwendungen fit für Windows 10 und für 64 Bit, mit einer profitablen Mobilstrategie, die durch die neuesten nativen Architekturen unterstützt wird. Mehr denn je ist „Powered by Delphi/C++Builder“ der smarteste Weg für alle, die Rentabilität, Sicherheit, Reputation und die Geschwindigkeit des Marktauftritts optimieren möchten.

WEITERE ANMERKUNGEN

Auch wenn es nicht das zentrale Thema dieses Papiers ist, wollen wir noch einige Anmerkungen zur Verbesserung der Entwicklerproduktivität in den vergangenen Jahren durch RAD Studio hinzufügen.

RAD Studio enthält nun Compiler zur nativen Anwendungsentwicklung unter Windows (32 Bit und 64 Bit), Linux (64 Bit), Android (32 Bit und 64 Bit), iOS (64 Bit) und macOS (64 Bit) mit Zugang zu 32 Bit-Compilern für iOS und macOS in früheren Editionen. Die Kompilierung erfolgt mit einer einzigen Codebasis, sodass sich Test- und Verwaltungskosten für alle Plattformen erheblich reduzieren lassen.

Entwickler, die die neueste Version der IDE nutzen, profitieren von vielen Updates zur Entwicklerproduktivität auch in der IDE. Dies betrifft u. a. folgende Bereiche:

- Integrierte Unterstützung eines Quellcode-Repositorys für Git, SVN und Mercurial
- Modernisiertes IDE-Layout und optionales dunkles Design
- Unterstützung eines größeren Arbeitsspeichers zur Kompilierung noch größerer Projekte
- Unterstützung von LSP (Language Server Protocol) für eine schnellere Programmierung mit Hintergrundverarbeitung der Codevervollständigung und für Aufgaben der Fehlererkennung
- Und vieles mehr ...

Weitere Informationen erhalten [Sie von Ihrem Embarcadero-Vertriebsbeauftragten](#).

DER ANSATZ VON EMBARCADERO

Native, plattformübergreifende Softwareentwicklung mit einer einzigen Codebasis.

Die IDE und die Frameworks von RAD Studio bieten Ihnen die Möglichkeit, in den modernen Sprachen Delphi oder C++ zu programmieren und mit einer einzigen Codebasis nativ für Windows, Linux, macOS, iOS sowie Android zu kompilieren.

Design it, Build it, Run it - today!

Jetzt testen

