



## Browser-basiert versus nativ

In industriellen Anwendungen drängen Browser-basierte Lösungen seit der Einführung von HTML5 verstärkt in den Markt. Doch native Anwendungen sind in einigen Bereichen schlicht sinnvoller. Deshalb müssen sich Entscheider der Stärken und Schwächen beider Ansätze bewusst sein.

Der populäre Browser-Ansatz hat ohne Frage viele Vorteile – vor allem in den Bereichen Usability und in der Nutzbarmachung mobiler Devices. Statt einem Bedienterminal können beispielsweise mehrere mobile Geräte als HMI verwendet werden, die individuell auf den Nutzer angepasst sind. Sie sind immer zur Hand, niemals ‚besetzt‘ – wenn beispielsweise verschiedene Personen in verschiedenen Funktionen gleichzeitig auf ein stationäres Gerät zugreifen wollen – und zeigen nur die Informationen an, die für den jeweiligen Anwender relevant sind. Auch die Vielfalt der nutzbaren Devices, von stationär bis mobil, von aktuellen IPCs, Smartphones und Tablets bis hin zu künftigen Wearables, ist ein klares Pro-Argument für diese Technologie. Browser-Lösungen punkten aber auch in der Praktikabilität – zum Beispiel durch Fernwartungskonzepte und den einfacheren Service durch zentralisierte Updates.

Diese Vorteile gibt es allerdings nicht kostenlos – je mehr Browser, Plattformen und Devices nutzbar sein sollen, desto grö-

ßer ist der damit verbundene Aufwand im Hintergrund und damit die Entwicklungs- und Erhaltungskosten. Denn selbst wenn Safari, Chrome, Edge, Firefox und Co ähnlich funktionieren, so muss Software jeweils für jede Version eines Browsers angepasst werden, um nicht nur die optimale, sondern auch die grundlegende Funktionalität zu erhalten. Dabei darf der Zyklus neuer Versionen bei Browsern nicht unterschätzt werden. Getrieben von B2C-Plattformen, permanenten Angriffen auf Schwachstellen und dem Wettbewerb kommt es zu deutlich häufigeren Updates als bei anderen Software-Angeboten. Mozillas Firefox erhält beispielsweise alle sechs Wochen eine planmäßige Aktualisierung und außerdem kommen noch diverse kleine Updates dazu, um Sicherheitslücken zu schließen. Auch die Basistechnologien zur Implementierung von Webseiten durchlaufen ihre Evolution in immer schnelleren Schritten – hier sei als Stichwort nur Googles Angular genannt – und zwingt Nutzer zu kontinuierlichen Veränderung. Zwar gibt es Anbieter von HMI-Software, die eigene,

festgeschriebene Browser-Versionen voraussetzen, doch damit verspielt man viele Vorteile des Browser-basierten Ansatzes – wie etwa die Flexibilität.

Ebenfalls mit großem Aufwand verbunden ist die notwendige Entwicklung von responsiven Designs beim Einsatz einer Vielzahl von unterschiedlichen Devices. Der Bau einer Oberfläche für ein Endgerät wie einen PC mag zwar schnell und einfach sein, doch will man diverse Devices unterschiedlicher Arten wie Smartphones, Smartwatches, Wearables und Co zusätzlich nutzen, muss man diese Oberfläche an die jeweiligen Displays und Funktionalitäten anpassen. Ein detailliertes Übersichtsbild einer Maschine mag zwar auf dem großen Monitor eines Desktop-PCs gut sichtbar und verständlich sein, auf dem winzigen Display einer Smartwatch würde es hingegen nicht funktionieren, da die ganzen Details viel zu klein dargestellt sind. Hier ist möglicherweise eine Beschränkung auf bestimmte Daten, Alarmer oder einfache Übersichten sinnvoll, womit man aber wieder bei einer

Mehrzahl an Oberflächen wäre, die gestaltet werden müssen.

### Bring your own device?

Ein großer Vorteil von Browser-Anwendungen ist die Installationsfreiheit. Die Software muss nicht installiert werden, sondern kann direkt über den Browser auf dem Server genutzt werden. Das reduziert den Aufwand für den Rollout neuer Versionen, Fehlerbehebung oder Wartung – denn Upgrades finden nur noch serverseitig statt. Auch ermöglicht der Ansatz das Gedankenspiel mit modernen Konzepten wie „Bring your own device“. Die Idee: Jeder Mitarbeiter kann seine privaten und gewohnten Devices wie Smartphone oder Tablet nutzen, um tagsüber über den Browser in der Produktionsanlage die Maschinen zu steuern oder Daten zu überwachen. Abends kann das Gerät wieder für die liebsten Serien, Spiele oder den Einkaufsbummel im Internet genutzt werden. Der Vorteil: Jeder Anwender nutzt das ihm vertraute Gerät und hat bei Bedarf jederzeit Zugriff auf die wichtigsten Informationen – so zumindest die Theorie. In der Praxis schlagen die meisten ITler der produzierenden Unternehmen bei dieser Vorstellung die Hände über dem Kopf zusammen. Die Security-Gefahren, die dem Betrieb durch die Verwendung dieser privat genutzten Devices drohen, sind enorm und

nach heutigem Stand der Technik nicht oder nur mit extremen Aufwand einzudämmen.

Um diese Risiken zu umgehen, führen viele Industrieunternehmen mobile Endgeräte als Arbeitsgeräte ein, die ausschließlich für den Einsatz im Unternehmen gedacht sind. Dieser Ansatz führt allerdings das „Bring your own device“-Konzept ad absurdum und berücksichtigt auch nicht den Faktor Mensch, der nicht immer sicherheitsbewusst und diszipliniert agiert.

Sicherheit ist und bleibt das zentrale Thema im industriellen Kontext – vor allem im Zusammenhang mit der zunehmenden Vernetzung in der Industrie 4.0 beziehungsweise dem IIoT. Auch bei der Wahl der für die jeweilige Applikation sinnvollsten HMI-Lösung spielt die Sicherheit eine gewichtige Rolle – und hier kommt es bei Browser-Lösungen zu teils gravierenden Einschränkungen der Praktikabilität gegenüber nativen Anwendungen.

### Die Sandbox

Browser sind primär für die Bewegung im Internet gedacht. Das Internet ist aus IT-Sicht die feindlichste und gefährlichste vorstellbare Umgebung, weshalb Browser von ihren Anbietern besonders gut vor Viren, Trojanern und anderer Schadsoftware gesichert werden. Ein zentrales Ele-

ment des üblichen Schutzkonzepts ist die Sandbox, in der der Browser agiert. Diese stellt einen isolierten Bereich dar, in dem die Anwendungen ausgeführt werden. So wird verhindert, dass Änderungen in der Sandbox – zum Beispiel ein Angriff auf den Browser – auch außerhalb – also an Hardware, Betriebssystem und Co – Schaden anrichten können.

Auf der einen Seite ist die Sandbox ‚State of the Art‘ bei allen relevanten Browser-Anbietern und für die Sicherheit unerlässlich. Auf der anderen Seite schränkt sie viele HMI-Funktionen bei der Nutzung der Web-UI ein, wie die Browser-basierte Lösung bei Inosoft genannt wird. Das Web-UI hat zum Beispiel keine Möglichkeit, auf die Hardware zuzugreifen, ohne dass die Sandbox entsprechend geöffnet wird. Eine Öffnung birgt aber nicht nur Gefahren durch höhere Verwundbarkeit gegenüber Angriffen aus dem Internet, sondern bedeutet auch erheblichen Mehraufwand bei der Programmierung: Um die Verwundbarkeit so gering wie möglich zu halten und die Öffnung jeweils in jeder Version eines Browsers nutzbar zu machen, müsste theoretisch jede Version jedes Browsers neu angepasst werden. In den handelsüblichen Browsern ist dies aber nicht möglich, da eine entsprechende Option eine skandalöse Sicherheitslücke darstellen würde.



Faktisch sind Öffnungen also nur bei selbstentwickelten Browsern denkbar – die dann allerdings wieder auf alle Plattformen und Geräte angepasst werden müssten. Nur so kann der Vorteil der Plattform- und Gerätefreiheit, ein zentrales Argument für Browser-basierte Lösungen, erhalten werden.

So sinnvoll die Sandbox für die Sicherheit im Internet ist, so sehr schränkt sie die Praktikabilität der Web-UIs ein. Denn sie verhindert zwar den Zugriff von Schadprogrammen auf die Systemhardware, aber sie verhindert in gleichem Maße auch die gewünschte Kommunikation mit externen Zusatzgeräten oder das Schreiben von Dateien auf die lokale Festplatte. Es gibt

zwar die Möglichkeit, diese Funktionen auf Webserver auszulagern, doch dafür muss der Datenaustausch via Client-Server-Kommunikation implementiert werden, um den Server zu veranlassen, die Daten vom Client aus abzulegen. Auch arbeiten viele Unternehmen mit stationären Lösungen wie Handscannern, Electronic-Key-Systemen oder RFID-Chips. Diese externen Lesegeräte sind üblicherweise via USB mit dem freizuschaltenden System verbunden – für Browser-Lösungen nur auf Seiten des Servers zu realisieren, und der steht nicht da, wo der Scanner ist.

Auch im Bereich der Maschinen- und Anlagensteuerung stellt die Undurchlässigkeit der Sandbox ein Problem dar. Doch

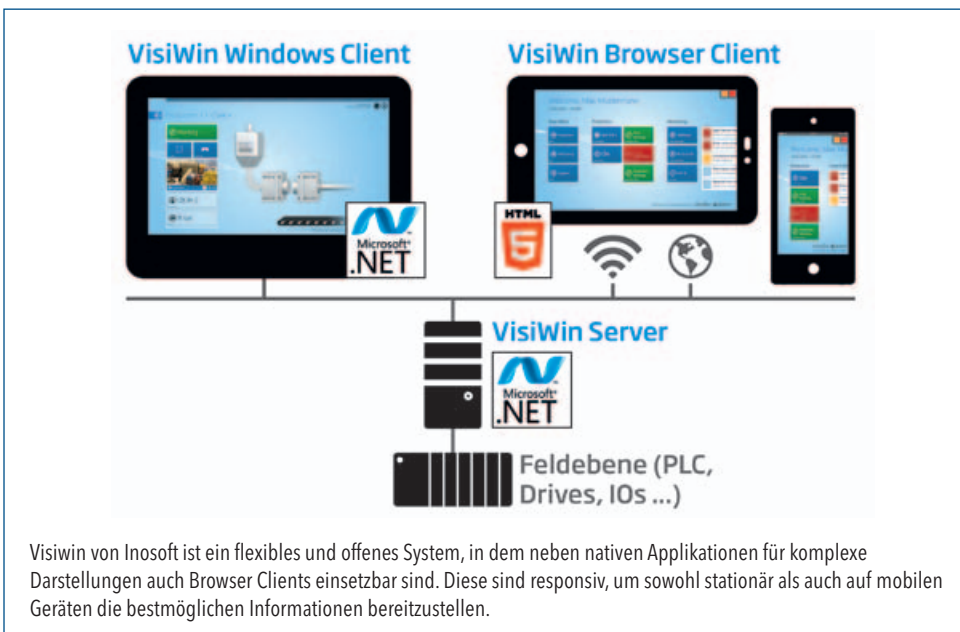
selbst wenn diese Herausforderung gelöst werden könnte, könnte die Browser-Lösung nur stationär betrieben werden. Die Maschinenrichtlinie erfordert bei jeder Art der Maschinensteuerung direkten Sichtkontakt vom Befehlsgeber zur Anlage. Sonst könnte es zu schlimmen Unfällen kommen, wenn beispielsweise ein Servicetechniker gerade den Kopf in die Anlage hält und der Operator von einem anderen Standort aus die Maschine startet. Steuerbefehle müssen also direkt an der Maschine erteilt werden – über ein fest installiertes Panel oder einen IPC. Da der mobile Vorteil hier also wegfallen würde, ist eine native Applikation ohne Sandbox für diese Aufgabe meist die effizientere Lösung.

**Komplexe Oberflächen sind stationär**

Wenn Browser-Lösungen auch auf lokalen IPCs und Panels genutzt werden können, kann hier der native Windows-Client punkten. Eine seiner Stärken ist die Fähigkeit, komplexe Applikationen auszuführen. Daten schreiben und lokal speichern sowie Hardware-Abfragen sind seit jeher PC-Standard und möglich, da die native Applikation auf dem IPC ohne die Sandbox direkten Zugriff auf alle verfügbaren Hardware- und Software-Komponenten erhalten kann. Selbst komplexe Darstellungen gepaart mit logischen Prozessen lassen sich bei überschaubarem Aufwand und geradlinigem Engineering problemlos umsetzen, da sowohl Bildschirmgrößen als auch Rechenleistung die benötigte Infrastruktur bieten.

Apropos Rechenleistung: Egal, ob das letztendliche HMI stationär oder mobil ist, irgendwo im System müssen Server- und Rechenleistung mit den entsprechenden Zugängen und damit verbundenen nativen Anwendungen bereitgestellt werden. In cloudbasierten Konzepten muss der Server zwar nicht mehr an der Maschine stehen, dafür müsste dann aber die Produktionsmaschine direkt mit der Cloud verbunden sein. Es gibt zwar Pläne in die Richtung, doch wird die Umsetzung auf sich warten lassen, da das produzierende Gewerbe noch erhebliche Bedenken hat. Daher müssen die Hersteller von HMI-Software bis dahin auch Maschinen von heute und morgen bedienbar machen.

Was die Usability angeht, ist der stationäre PC vielen Anlagentechnikern mindestens genauso vertraut wie das Smartphone,



unterscheidet sich das Windows-Betriebssystem in der industriellen Anwendung doch kaum von dem am heimischen Computer.

## Die Kombination macht's

Beide Ansätze – die native Windows-Anwendung und der auf Browser-Technologie basierende – haben ihre Vor- und Nachteile. Diese verhalten sich in vielerlei Hinsicht komplementär – die Stärke des einen ist die Schwäche des anderen. Deshalb ist es für Industrieunternehmen unabdingbar, das gesamte Einsatzgebiet holistisch zu betrachten und die Implementierung sorgfältig zu durchdenken. Was soll wann, wo, wie und von wem gemacht werden? Lässt sich vielleicht ein Ausblick in die technologischen Produktionsbedingungen der nahen Zukunft werfen? Können daraus weitere Schlüsse für den Bedarf gezogen werden? Sprich: Kann das System quasi „bereit für die Zukunft“ gemacht werden? Aus allen verfügbaren Informationen lässt sich die richtige Software-Lösung für den HMI-Bereich ableiten.

In einer abgeschiedenen Maschinenanlage, wie einer Pumpenstation oder einer Windkraftanlage, wo im Normalfall kein Techniker vor Ort ist, ist der Fernzugriff über den Browser für den Wartungs- und Überwachungsbetrieb alternativlos – native

Apps auf einem Panel-PC an der Anlage direkt sind hier entsprechend unbrauchbar. An einer Produktionsmaschine hingegen, die rund um die Uhr von einem Operator betrieben wird, reicht der stationäre Zugriff über den IPC oder Panel-PC meist völlig aus und die native App erlaubt den benötigten Zugriff auf die Hardware.

Genau genommen ist die Kombination beider Ansätze heute bereits Basis der meisten mobilen Lösungen. Die Rechenleistung, die für Sammlung und Bereitstellung der Datenbasis sowie Datenaufzeichnung, -upload oder -analyse benötigt wird, kann technologisch noch nicht von mobilen Geräten erbracht werden. Das heißt, sowohl heute als auch auf absehbare Zeit wird der stationäre Rechner mit nativen Applikationen als Server-Grundlage einer jeden Datenverarbeitung im großen Umfang bleiben – egal, ob diese Daten nativ oder Browser-basiert auf mobilen oder stationären Geräten angezeigt werden.

Der Abgesang auf die native Windows-App kommt daher sicher viel zu früh. Als ein Hersteller von Prozessvisualisierungs-Software setzt Inosoft bei seiner Software Visiwin auf ein offenes und flexibles System, das entsprechend der Kundenbedürfnisse sowohl auf native als auch auf web-basierte Lösungen angepasst werden kann. Die native Lösung, die bei Visiwin „Modern

UI“ genannt wird, basiert dabei auf der sehr leistungsfähigen Grafikschnittstelle Windows Presentation Foundation (WPF), welche standardmäßig bei Microsoft-Betriebssystemen seit Vista integriert ist. Diese ermöglicht die Erstellung und Anzeige von animierten High-End-Grafiken, anspruchsvollen Designs oder 3D-Grafiken. Als natives System lassen sich dank der .Net-Programmierung beliebig komplexe Logiken realisieren und sogar Komponenten von Drittanbietern nahtlos einbinden. Darüber hinaus lässt sich Visiwin via Web-UI nutzen, welches Browser-basiert Plattform-unabhängige Nutzeroberflächen auf mobilen Geräten auf Basis von HTML5 und Java-Script ermöglicht. *ld*



**SVEN KRÖGER**  
ist Produktmanager bei Inosoft.



**STEFAN NIERMANN**  
ist in Vertrieb und Marketing bei Inosoft tätig.