

## Inkompatible Browser und Monokulturen

**Egg, 6. März 2019:** Das Release 2019/II bringt neben vielen Fortschritten bei ArchivistaERP vor allem eines: Kompatibilität zu den aktualisierten Web-Browsersn. Dabei geht es um ziemlich unspannende Details wie Funktionstasten, aber auch um die Frage, wie kompatibel aktuelle Browser sind bzw. wie die Zukunft webbasierter Lösungen aussehen kann bzw. wird.



## Browser-Updates werden automatisch aufgespielt

Irgendwann im Januar, ein Kunde meldet, dass mit dem aktuellen Update (Firefox 65) die Funktionstasten nicht mehr funktionieren würden, alle übrigen Browser würden einwandfrei arbeiten (selbst Firefox 64 laufe). Wohl ein Bug von Firefox? Der Kunde meldet, dass er die automatisierten Updates abschalte, alles weiter nicht so tragisch.

Einige Tage später meldet der Kunde, die automatisierten Updates liessen sich leider nicht abschalten, ein Wechsel auf ein anderer Browser werde geprüft. Es melden sich weitere Kunden, bei denen allesamt die Funktionstasten mit Firefox ab Version 65 nicht mehr zur Verfügung stehen.

Ein erstes Update auf Firefox 65 unter Windows10 zeigt inhouse, die Funktionstasten arbeiten problemlos. Nach einem erneuten Update (noch immer wird Firefox 65 angezeigt) lässt es sich jedoch zweifelsfrei reproduzieren, die Funktionstasten laufen nicht mehr. Die Suche nach der Ursache beginnt. Nach einigen Stunden Recherche wird folgende Seite gefunden:

<https://developer.mozilla.org/en-US/docs/Web/Events/keydown>

Auf besagter Seite steht, dass ab Firefox 65 ein sogenannter IME-Bug behoben worden sei, dazu der folgende Link:

<https://www.fxsitecompat.com/en-CA/docs/2018/keydown-and-keyup-events-are-now-fired-during-ime-composition>

Auf der obigen Seite wiederum steht (ziemlich verklausuliert), dass es ein neues Verhalten gebe und dass, falls dieses nicht gewünscht werde, der untenstehende Code zur Anwendung gelangen könne/sollte:

```
$input.addEventListener('keydown', event => {
  if (event.isComposing) {
    return;
  }
  // Do something
});
```

Und siehe da, mit obigem (modifizierten Code) laufen die Funktionstasten auch

alle wieder mit Firefox 65. Kurz und gut, Firefox 65 hat sich selber abgesägt. Nicht für Firefox 65 geänderte Lösungen laufen ganz einfach deshalb nicht mehr, weil Mozilla die Kompatibilität «gebrochen» hat. Welchen Sinn derartige Änderungen im Sinne einer Abwärtskompatibilität ergeben, bleibe dahingestellt. Fakt ist, die ArchivistaBox erhält dafür ein Update.

### **Kompatibilität des Patches zu anderen Browsern**

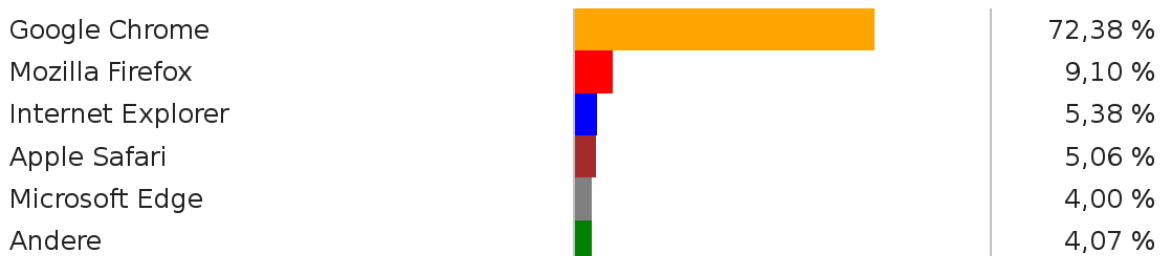
Alle verfügbaren Browser werden mit den Anpassungen, die für Firefox 65 nötig sind, getestet. Läuft alles einwandfrei bis zu jenem Moment, wo in einer Terminal-Server-Umgebung Internet Explorer 11 nicht läuft. Es stellt sich heraus, dass der obige Patch nur deshalb nicht läuft, weil der Vorschlag der Änderung selber Code enthält, der nicht abwärtskompatibel zu älteren Lösungen ist (Stichwort anonyme Funktionen).

Selbstverständlich konnte der Code so umgeschrieben werden, dass er abwärtskompatibel läuft. Fragt sich einfach: Erleben wir da gerade einen Browser-«Krieg» mit umgekehrten Vorzeichen? Das sei hier ausdrücklich nicht behauptet, es reicht allemal, dass Firefox 65 die eigene Abwärtskompatibilität über den Haufen wirft.

### **Browser-Alternativen**

Seit der ersten Version der ArchivistaBox wird Firefox als Standard-Browser auf der ArchivistaBox mitgepackt. Ebenso werden seit dieser Zeit möglichst alle gängigen Browser unterstützt: Safari, Internet Explorer, Edge, Chrome und selbst Opera konnten in der «freien» Wildbahn mit der ArchivistaBox gesichtet werden.

Aus prinzipiellen Überlegungen (eine webbasierte Lösung sollte plattformunabhängig laufen) gibt es bei der ArchivistaBox keine explizite Empfehlung zur Wahl eines Browsers. Trotzdem sei an dieser Stelle noch kurz die Frage aufgegriffen, welche Optionen bei der Wahl des Browsers aktuell bestehen.



Sowohl Safari, Firefox, Opera als auch Chrome basieren auf dem gleichen Fundament (WebKit mit Abspaltung von Chrome). Der Internet Explorer von Windows wurde vor einigen Jahren entwicklungstechnisch für den Nachfolger Edge in Rente geschickt, sodass mit Edge die einzige technische «Alternative» bleibt, bzw. bliebe, denn Edge wird ab diesem Jahr ebenfalls auf der gleichen Technologie wie die übrigen Browser beruhen, siehe dazu:

<https://blogs.windows.com/windowsexperience/2018/12/06/microsoft-edge-making-the-web-better-through-more-open-source-collaboration>

Microsoft verspricht darin: «*Making the web better through more open source collaboration*». Natürlich spricht nichts dagegen, dass Microsoft auch auf Open Source Technologie zurückgreift. Vorliegend führt der Wechsel aber dazu, dass anstelle einer offenen Spezifikation (HTML/CSS/JavaScript) noch genau eine Technologie übrigbleibt.

### **«Offene» Monokultur**

Wer die aktuellen Zahlen zur Browser-Vielfalt begutachtet, stellt fest (siehe dazu <https://de.wikipedia.org/wiki/WebKit> bzw. <http://gs.statcounter.com>), es gibt da vor allem eines: Monokultur. In den obigen Zahlen sind nur die Desktop-Rechner enthalten, auf den Smartphones dürfte es noch viel krasser zugunsten des Platzhirsches aussehen.

Vor zehn Jahren hätte hier wohl gestanden, [siehe dazu den Portokässeli-](#)

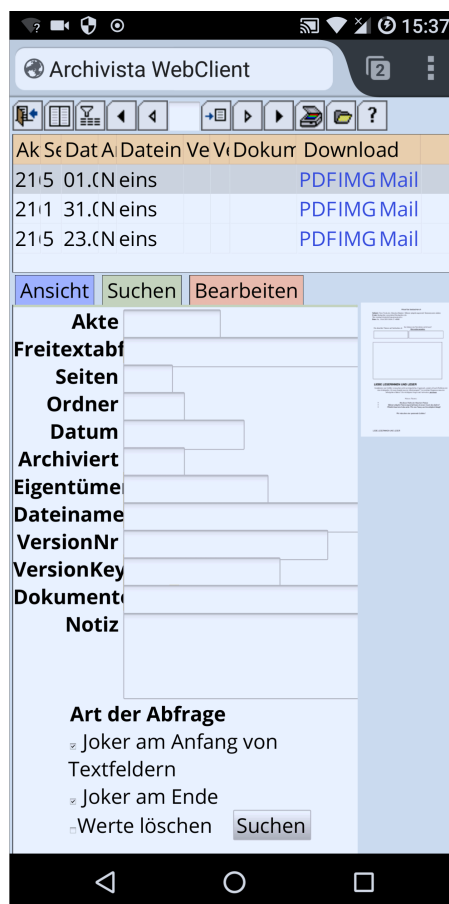
**Artikel**, wie bösartig das Monopol zuschlägt (damals ging es um Microsoft). Im Prinzip müsste dies auch heute noch hier so stehen, dies nur noch viel pointierter. Doch das Jahrzehnt hat ja auch gezeigt, Märkte können sich schnell(er) ändern.

Natürlich ist es bedenklich, wenn es faktisch ein Monopol bei der Suche im Web gibt, und auch die massive Verbreitung von Android bei den Smartphones ist nicht ohne. Der Unterschied zu damals besteht aber doch darin, dass sowohl die heutige Browser-Engine als auch Android als Betriebssystem (siehe dazu die **Artikel zu LineageOS**) im Quellcode offengelegt sind.

### Browser-Alternativen zum Monopol

Die Browser-Unabhängigkeit ist für die ArchivistaBox entscheidend. Nur so kann sichergestellt werden, dass die ArchivistaBox auf allen Betriebssystemen einwandfrei läuft. Aus diesem Grunde wird der Markt seit Jahren immer wieder beobachtet.

Wie oben ausgeführt, der Alternativen sind nunmehr nicht mehr viele. Und doch sei an dieser Stelle **PaleMoon** kurz vorgestellt, weil dieser Browser mit relativ bescheidenen Ressourcen klarkommt. Dies ist insbesondere dort wichtig, wo ältere Rechner vorhanden sind.

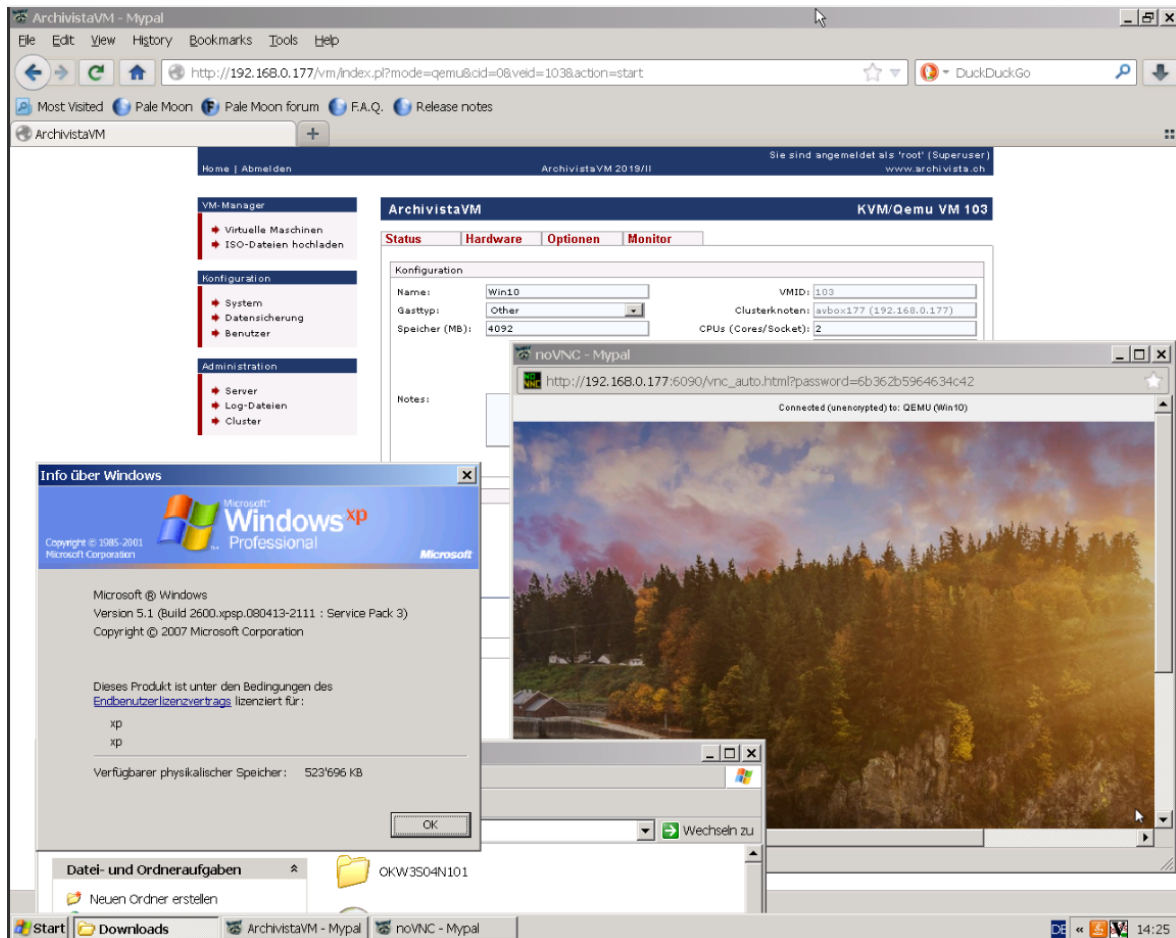


**PaleMoon** ist eine Abspaltung von Firefox und läuft unter Windows und Linux; selbst eine Version für Android (getestet mit LineageOS) gibt es. Wer also lieber ohne Tracking durch die «Giganten» surft, darf sich **PaleMoon** gerne ansehen. Ausserdem gibt es von **PaleMoon** mit **MyPal** eine Version, die problemlos noch unter WindowsXP läuft. So konnte denn unter WindowsXP über ArchivistaVM problemlos eine Windows10-Instanz bedient werden.









## ArchivistaBox 2019/II

Genug der «Spielereien», was bringt die ArchivistaBox 2019/II sonst mit. Neben der Browser-Kompatibilität gibt es insbesondere viele Neuerungen zu ArchivistaERP. Zur Erinnerung: Mit ArchivistaERP steht eine leichtgewichtige ERP-Lösung zur Verfügung, welche auf jeder ArchivistaBox zur Verfügung steht. Sie kann (zusammen mit ArchivistaDMS) in der ArchivistaCloud komplett kostenfrei betrieben werden. ArchivistaERP ist perfekt auf Schweizer Verhältnisse abgestimmt, sie lässt sich aber auch in anderen Ländern problemlos einsetzen.

Mit der Version 2019/II werden alle Menüs mit Funktionstasten unterstützt. Konnten bisher die Tastenkombinationen primär beim Buchen von Aufträgen verwendet werden, so stehen neu über sämtliche Menü Funktionstasten zur Verfügung.

Neu können Zahlungen auch mit dem neuen ISO20022-Standard automatisch verbucht werden, daneben gibt es Batches zum Buchen über Referenznummern, womit Zahlungseingänge deutlich einfacher erfasst werden können. Daneben wurden die Reporte (z.B. Bilanz/Erfolgsrechnung) «aufpoliert», weitere Neuerungen werden 2019 noch folgen.