



# Welche App-Technologien gibt es? Was ist bei ihrem Einsatz zu beachten?

Webinar der

**DVZ**

# Einsatz von logistischen Apps

---

## auf Hardware

Industrie-Handheld



Smartphone / Tablett



## bei Nutzern

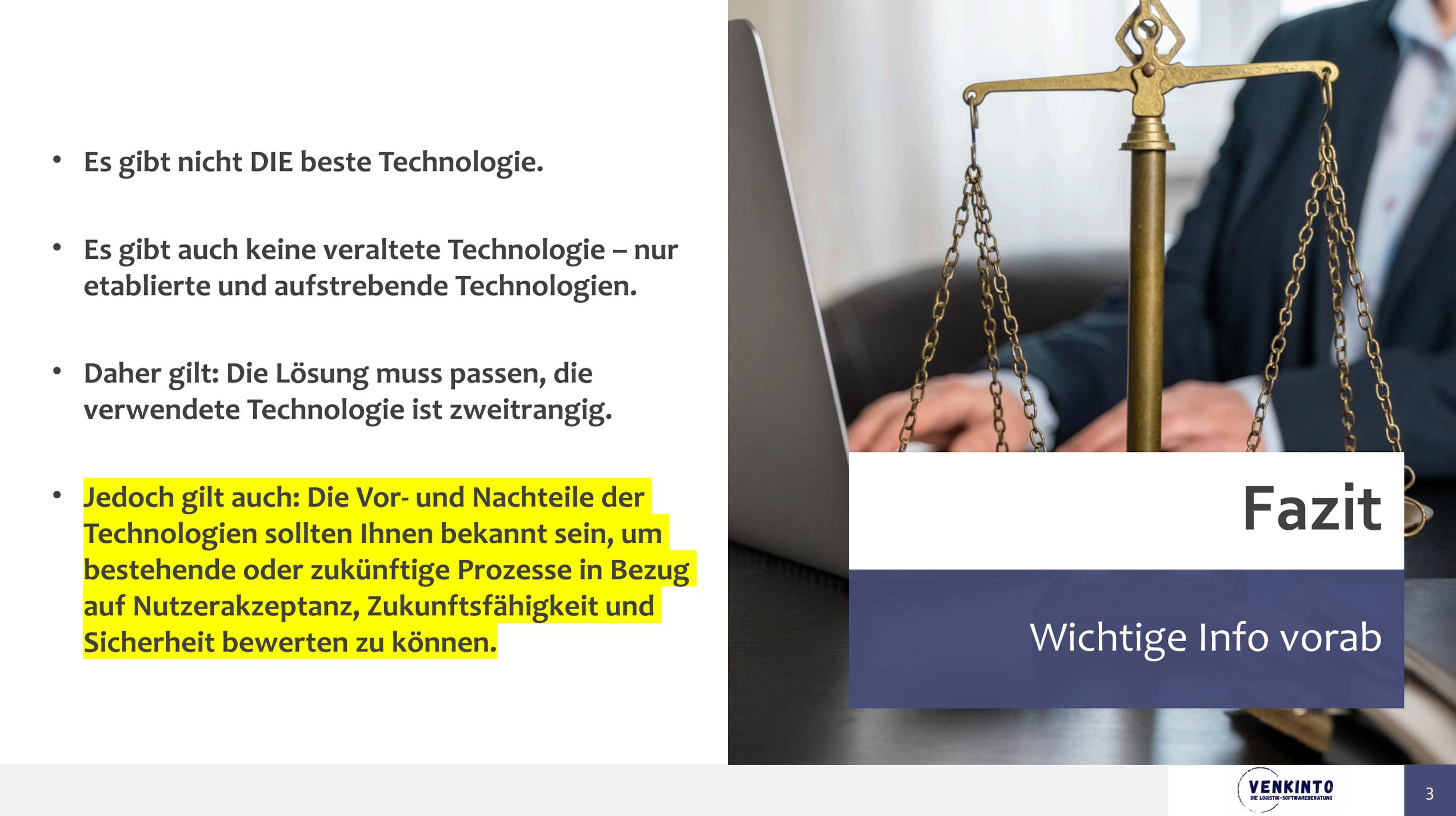
„Bekannte“ Fahrer



„Unbekannte“ Fahrer



- Es gibt nicht DIE beste Technologie.
- Es gibt auch keine veraltete Technologie – nur etablierte und aufstrebende Technologien.
- Daher gilt: Die Lösung muss passen, die verwendete Technologie ist zweitrangig.
- Jedoch gilt auch: Die Vor- und Nachteile der Technologien sollten Ihnen bekannt sein, um bestehende oder zukünftige Prozesse in Bezug auf Nutzerakzeptanz, Zukunftsfähigkeit und Sicherheit bewerten zu können.



Fazit

Wichtige Info vorab

# Die Klassiker

## Native Apps

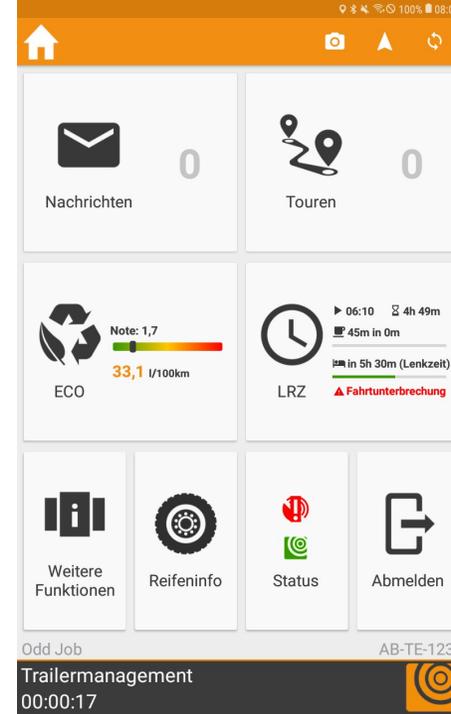
- „Klassische“ Apps, die speziell für ein Betriebssystem geschrieben und auf dem Endgerät, meist über einen App-Store, installiert werden.

## Web-Apps

- Web-Seiten, die von einem Browser angezeigt werden. Der Aufruf kann über einen Link, QR-Code, Lesezeichen oder auch ein App-Icon auf dem mobilen Endgerät erfolgen. Es erfolgt keine Installation.

## Hybride Apps

- Web-Seiten, die in einem nativen Rahmen eingebunden werden und die über den App-Store o.ä. Wege installiert werden können.



# Technologien

## Grobe Abgrenzung

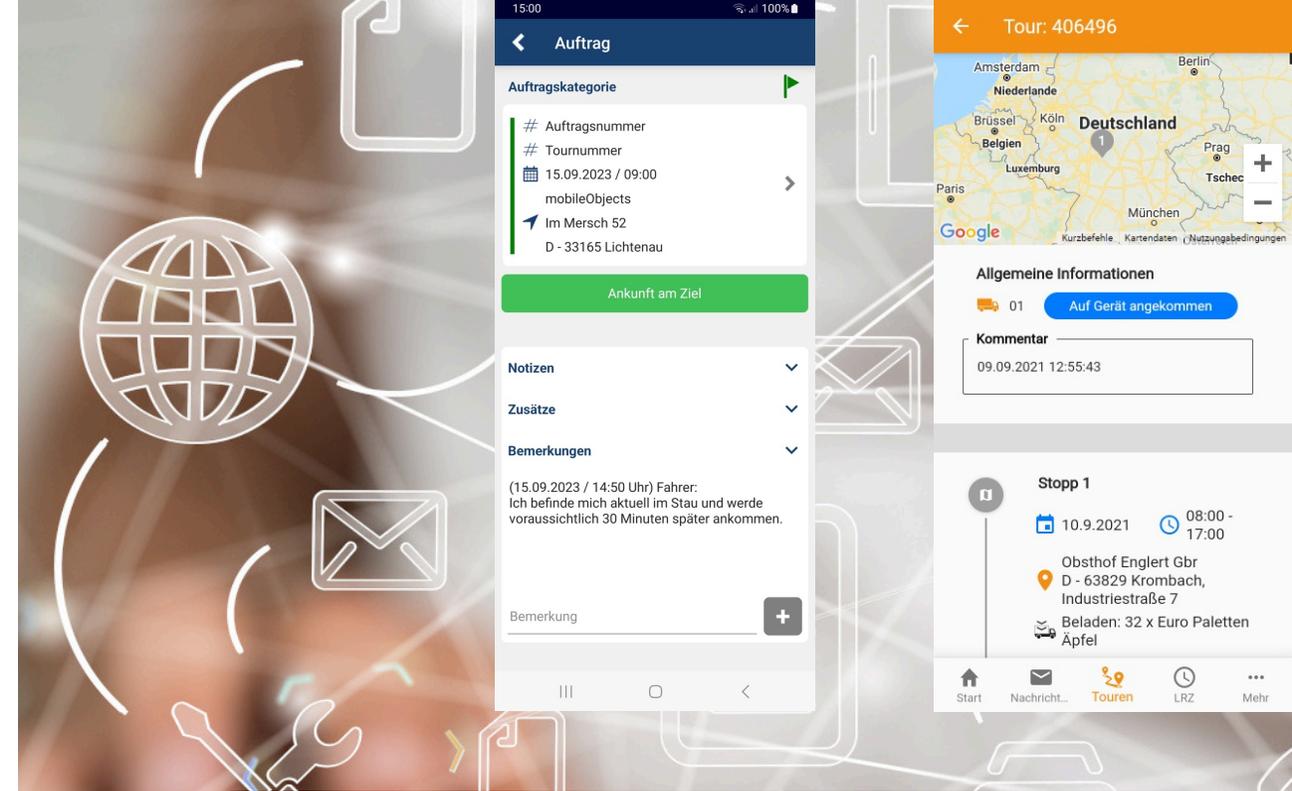
# Die Aufstrebenden

## Progressive Web-Apps

- Progressive Web-Apps sind quasi eine Weiterentwicklung der Web-Apps sind.
- Websites, die aus der Website heraus oder über einen App-Store installiert werden können, Zugriff auf Gerätefunktionen haben und die auch offline funktionieren.
- Ca. 2015 erschienen, seit 2016 aktiv von Google gefördert, erst 2023 von iOS „vollständig“ unterstützt.

## Cross Plattform Apps

- Cross Plattform Apps basieren auf einem Framework, dass entwickelten Code automatisiert in native Apps für für verschiedene Plattformen und Websites umwandelt.
- Cross Plattform Ansätze gibt es schon lange (Java, Flash, .NET, ... ), sie haben sich aber erst mit Frameworks wie Flutter (Google, 2017) und React Native (Meta, 2015) durchgesetzt.



# Technologien

## Grobe Abgrenzung

## Vorteile

- Voller Zugriff auf gerätespezifische Hard- und Software, wie bspw. Kamera, Scanner, Kontakte, ...
- Vollständige Offline-Nutzung möglich, sofern die Inhalte dies zulassen.
- Höchste Stufe bei Performance und Security\* möglich.

## Nachteile

- Nur auf freigebenden Betriebssystemen und mit freigegebener Hardware nutzbar.
- Es muss eine Installation der App erfolgen.
- Hoher Entwicklungsaufwand bei breiter Geräteunterstützung.

## Einsatzszenarium in der Logistik

- eigene Flotte oder feste/bekannte Fahrer
- Bedarf an der Nutzung umfangreicher Hardwarefunktionen (bspw. Industrie Handhelds)
- inhaltlich komplexe/umfangreiche Lösung mit einem hohen Standardisierungsgrad



# Vor- und Nachteile

Native Apps

## Vorteile

- Lauffähig ohne Installation.
- Geräteunabhängig lauffähig auf allen Geräten mit einem Browser.
- User Experience durch die Ähnlichkeit mit bekannten Consumer-Anwendungen gefühlt „besser“.
- Geringer Entwicklungsaufwand.

## Nachteile

- Nur online nutzbar.
- Fehleranfällig und/oder nur eingeschränkt nutzbar, bei der Verwendung von Hardware (bspw. GPS).

## Einsatzszenarium in der Logistik

- Es werden wechselnde/unbekannte Fahrer oder Dienstleister eingesetzt.
- Die eingesetzte Lösung/App wächst stetig und wird häufig aktualisiert/erweitert oder hat nur einen „geringen“ Funktionsumfang.

# Vor- und Nachteile

Web-Apps

## Vorteile

- Zugriff auf gerätespezifische Hard- und Software.
- Offline-Nutzung möglich, sofern die Inhalte dies zulassen.
- Bestehende Web-Lösungen integrierbar.
- Ist leichter/schneller plattformübergreifend zu entwickelt als native Apps, da der native Code-Anteil in der Regel deutlich kleiner als 50% ist.

## Nachteile

- Es muss eine Installation der App erfolgen.
- Komplexere Systemarchitektur.

## Einsatzszenarium in der Logistik

- Eigene Flotte oder feste/bekannte Fahrer.
- Bestehende Web-Lösung vorhanden, die als mobile Lösung eingebunden werden soll.

# Vor- und Nachteile

## Hybride Apps

## Cross Plattform

- Die Entwicklung basiert auf einem separaten Entwicklungsframework, das am Ende den Code für die verschiedenen Betriebssysteme ausgibt. Native Apps sind somit schneller und kostengünstiger erstellt.
- Vermischt mobile und native Elemente, ähnlich wie hybride Apps, verwendet aber eine eigene Oberfläche.
- Einziger Nachteil ist, dass die Weiterentwicklung von einem aktuellem Entwicklungsframework anhängt.

## Progressive Web-App

- Nutzt eine neue Architektur, die von Google/Android forciert wird und inzwischen auch schon auf iOS nutzbar ist. Im Grunde eine Web-App, die alle Nachteile verliert. Offline nutzbar, ohne Installation aufrufbar und dennoch installierbar. Aus Sicht vieler Experten die Zukunft – dies allerdings bereits seit vielen Jahren.
- Nachteil ist, dass Apple aktuell noch in einigen Punkten vom Standard abweicht und erst seit 2023 „vollständig“ Progressive Web-Apps unterstützt.



# Die Aufstrebenden

Cross Plattform und  
Progressive Web-App

# Vor- und Nachteile der Technologien

	Nativ	Web	Hybrid	Cross	PWA
App Store / Mail-/SMS-Link	✓ / -	- / ✓	✓ / ✓	✓ / ✓	✓ / ✓
Plattformabhängig	-	✓	✓	✓	✓
Gerätefunktionen	X X X	X	X X	X X	X X
Kosten & Entwicklungszeit	X	X X X	X X	X X	X X
Performance	X X X	X	X X	X X	X X
Sicherheit *	X X X	X	X X	X X X	X X
Offline Verfügbar	X X X	-	X X	X X X	X X
Etabliert	X X X	X X X	X	X X	X
Individualisierbarkeit *	X	X X X	X X	X X	X X X

\* Ergänzende Informationen auf den Folgeseiten

## Unterschiedliche Prozesse von unterschiedlichen Unternehmen in einer App

- Moderne Lösungen beinhalten Workflow-Manager oder breite Konfigurationsmöglichkeiten. Hierdurch lassen sich standardisierte Apps für unterschiedliche Bedarfe frei konfigurieren – unabhängig von der verwendeten Technologie.
- Umgesetzte Änderungswünsche sind bei einer Standard-App von Release-Zyklen abhängig und müssen unter Umständen bei installationsbasierten Apps „over the air“ oder in definierten Standorten (Depot, Heimatadresse, ... ) heruntergeladen und installiert werden, was die schnelle und gleichzeitige Verbreitung erschweren kann.
- Änderungen an webbasierte Apps stehen unbemerkt und umgehend für alle Nutzer bereit. Vom Standard abweichende Lösungen können über eine „Link-Wechsel“ einfach bereit gestellt werden.

```
ajaxResponse: function() {  
  {  
    params = params || {};  
    params.force_exec = true;  
    params.callback = true;  
    params.pre_processing = true;  
  
    var regex_all = new RegExp('');  
    var matches = [];  
    var match = '';
```

# Individualisierung

Gleich ist nicht gleich gleich

## Maximale Sicherheit

- Alle App-Technologien sind sicher, wenn sie die aktuellen Sicherheitsstandards berücksichtigen.
  - Verschlüsselte Datenübertragung, sichere Anmeldung, ...
- Alle web-basierten Technologien sind durch die die Nutzung eines Web-Viewers bzw. Browsers abhängig von der Sicherheit von Drittkomponenten.
- In der Theorie bieten native Apps den höchsten Sicherheitsstandard. In der Praxis reicht aber auch hier eine kompromittierter Programmteil (Code, Treiber, ...), um Schwachstellen zu öffnen.
  - Aktuelles Beispiel: Die bonify App. Die Tochter der Schufa hat ihre native App aktuell aus dem Store entfernt und bietet nur noch eine Web-App an.
- Es gibt keine 100%ige Sicherheit. Daher ist das Sicherheitskonzept des Anbieters sehr hoch zu gewichten, dies gilt auch für das zugesicherte Verhalten in Bezug auf Reaktion und Transparenz im Falle einer Sicherheitslücke.



# Sicherheit

Sicher ist alles, bis jemand einen Fehler macht.



# Vielen Dank

Volker Möller



+49 176 45671996



vmoeller@venkinto.com



<https://venkinto.com>

