

# Progressive Web Apps Regelwerk

## Inhaltsverzeichnis

1.	Einleitung .....	3
1.1.	Zweck des Dokuments.....	3
1.2.	Was sind Progressive Web Apps? .....	3
1.3.	Wie nutze ich das Regelwerk?.....	3
2.	Progressive Web App Regeln .....	4
2.1.	Hardwarebasierte Funktionen .....	4
2.1.1.	Aufnahme von Medien.....	4
2.1.1.1.	Aufnahme von Bildern.....	4
2.1.1.2.	Aufnahme von Audio.....	4
2.1.1.3.	Aufnahme von Videos .....	5
2.1.2.	Ausgabe von Medien und Informationen.....	6
2.1.2.1.	Anzeigen von Bildern .....	6
2.1.2.2.	Wiedergabe von Audio .....	7
2.1.2.3.	Wiedergabe von Videos .....	7
2.1.2.4.	Erzeugung von Vibrationen .....	8
2.1.3.	Persistenz.....	8
2.1.3.1.	Speichern von Schlüssel-Wert-Paaren .....	8
2.1.3.2.	Speichern von komplexen Daten .....	8
2.1.3.3.	Zugriff auf das Dateisystem.....	9
2.1.4.	Kommunikation.....	9
2.1.4.1.	Mobilfunkbasierte Kommunikation.....	9
2.1.4.1.1.	Telefonie .....	9
2.1.4.1.2.	Messaging (SMS) .....	10
2.1.4.2.	Internetbasierte Kommunikation .....	10
2.1.4.2.1.	Abfrage des Netzwerkstatus .....	10
2.1.4.2.2.	Kommunikation über das Internet .....	11
2.1.4.3.	Sonstige Kommunikation.....	11
2.1.4.3.1.	Positionsbestimmung.....	11
2.1.4.3.2.	Bluetooth.....	11
2.1.4.3.3.	NFC .....	12
2.1.4.3.4.	USB .....	12
2.1.5.	Auslesen von Sensoren.....	13

2.1.5.1.	Bewegungssensoren .....	13
2.1.5.1.1.	Beschleunigungssensor .....	13
2.1.5.1.2.	Rotationssensor .....	14
2.1.5.2.	Umgebungssensoren .....	15
2.1.5.2.1.	Lichtsensoren .....	15
2.1.5.2.2.	Temperatursensor .....	15
2.1.5.2.3.	Magnetfeldsensor .....	16
2.1.5.2.4.	Näherungssensor .....	16
2.1.5.2.5.	Luftdrucksensor .....	17
2.1.5.2.6.	Luftfeuchtigkeitssensor .....	17
2.2.	Softwarebasierte Funktionen .....	17
2.2.1.	Softwarebasierte Kommunikation .....	18
2.2.1.1.	Versenden von E-Mails .....	18
2.2.1.2.	Push Notifications .....	18
2.2.1.3.	Kommunikation mit nativen Apps .....	19
2.2.2.	Installation und Offlinebetrieb .....	19
2.2.3.	Anwendungen zur persönlichen Organisation .....	20
2.2.3.1.	Wecker .....	20
2.2.3.2.	Kalender .....	21
2.2.3.3.	Kontaktliste .....	21
2.2.3.4.	Notizen .....	22
2.2.4.	Landkarten .....	22
2.2.5.	Vertrieb .....	23
2.2.6.	Text- und Sprachverarbeitung .....	24
2.2.6.1.	Sprachsynthese (Text to Speech) .....	24
2.2.6.2.	Spracherkennung (Speech to Text) .....	24
3.	Gutachten und Korrektur .....	25

# **1. Einleitung**

## **1.1. Zweck des Dokuments**

Dieses Dokument beschreibt durch eine Reihe von Regeln die Funktionalitäten und möglichen Einsatzszenarien von Progressive Web Apps. Das Regelwerk soll Softwareentwickler bei der Entscheidungsfindung unterstützen, ob eine Progressive Web App anstelle einer nativen App entwickelt werden kann. Dabei werden lediglich die technischen Möglichkeiten betrachtet. Aspekte, wie z.B. Usability, User Experience oder das Vorwissen der Entwickler, werden in diesem Regelwerk nicht aufgeführt.

## **1.2. Was sind Progressive Web Apps?**

Progressive Web Apps sind responsive und per HTTPS übertragene Webanwendungen, die nach dem Grundsatz des Progressive Enhancement die Fähigkeiten der Browser für eine fortschreitende Verbesserung nutzen, wodurch mittels Offlinefunktionalität über Service Worker, eine Installation anhand eines Web App Manifests und Push Notifications eine zuverlässige, motivierende und native Nutzererfahrung gewährleistet wird.

Progressive Enhancement ist ein Prinzip, nach dem Webanwendung so entwickelt werden sollen, dass deren grundlegende Inhalte selbst auf alten Geräten und in alten Browsern genutzt werden können, wohingegen die Funktionen neuer Browser nur bei Vorhandensein für eine fortschreitende Verbesserung der Nutzererfahrung eingesetzt werden.

## **1.3. Wie nutze ich das Regelwerk?**

Das Regelwerk sollte genutzt werden, nachdem in einem Requirements Engineering die Anforderungen zur Anwendung ermittelt wurden. Im ersten Schritt muss festgelegt werden, welche Browser in welchen minimalen Versionen unterstützt werden sollen. Im zweiten Schritt müssen die Anforderungen mit den im folgenden Kapitel „2 Progressive Web App Regeln“ aufgeführten Funktionalitäten verglichen werden. Dazu wurden neben einer Beschreibung die Möglichkeiten zur Umsetzung einer Funktion zusammen mit den Browserkompatibilitäten aufgeführt. Wird eine Funktion nicht von einem gewünschten Browser unterstützt, liegt es im eigenen Ermessen, ob die Funktion so wichtig ist, dass eine Progressive Web App nicht in Frage kommt. Generell sollte eine fehlende Funktion nicht zur Resignation führen, sondern eher als Grund dazu gesehen werden, einen kreativen Weg um dieses Problem herum zu finden. So lassen sich z.B. Funktionen, die durch keine Web API unterstützt werden, in bestimmten Situationen durch Auslagerung von Arbeit an einen Webservers oder durch Nutzung von Browserplugin doch umsetzen. So handelt es sich bei den im Abschnitt „2.2 Softwarebasierte Funktionen“ aufgeführten Punkten hauptsächlich um Komfortfunktionen, die zwar nicht direkt vom Browser bereitgestellt werden, jedoch durch eigene Implementierungen oder einem Web Service umgesetzt werden können.

## 2. Progressive Web App Regeln

### 2.1. Hardwarebasierte Funktionen

Mit hardwarebasierten Funktionen sind solche gemeint, die in direktem Zusammenhang mit der Hardware des Endgerätes stehen. Da sich die vielen Gerätetypen in ihrer Ausstattung unterscheiden, ist das Vorhandensein der benötigten Hardware-Komponente immer Voraussetzung für die Nutzung der hardwarebasierten Funktion.

#### 2.1.1. Aufnahme von Medien

##### 2.1.1.1. Aufnahme von Bildern

Die Aufnahme von Bildern erfolgt über eine integrierte Kamera oder einer externen Webcam. Das Web bietet 2 Möglichkeiten zur Aufnahme von Bildern:

Einerseits kann ein Datei-Input-Element so konfiguriert werden, dass es nur Bilder als Werte akzeptiert. Dieses könnte wie folgt aussehen:

```
<input type="file" accept="image/*">
```

Abhängig vom Betriebssystem und Endgerät wird ein Dateiauswahldialog angezeigt oder eine App zur Aufnahme von Fotos geöffnet. Diese Variante wird von allen Browsern unterstützt.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	+	+	+

Andererseits kann auch direkt über die Media Capture API auf die Kamera zugegriffen werden. Dieser Zugriff liefert zwar nur Videodaten, es kann jedoch ein Frame des Videos ausgelesen und in ein Bild umgewandelt werden. Über die Media Capture API kann lediglich die Auflösung des Bildes eingestellt werden.

Spezifikation:

Titel	Link	Status
Media Capture API	<a href="https://www.w3.org/TR/mediacapture-streams/">https://www.w3.org/TR/mediacapture-streams/</a>	Candidate Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
47	36	11	-	+

##### 2.1.1.2. Aufnahme von Audio

Die Aufnahme von Audio erfolgt über ein Mikrofon. Das Web bietet 2 Möglichkeiten zur Aufnahme von Audio:

Einerseits kann ein Datei-Input-Element so konfiguriert werden, dass es nur Audiodaten als Werte akzeptiert. Dieses könnte wie folgt aussehen:

```
<input type="file" accept="audio/*">
```

Abhängig vom Betriebssystem und Endgerät wird ein Dateiauswahldialog angezeigt oder eine App zur Aufnahme von Audio geöffnet. Diese Variante wird von allen Browsern unterstützt.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	+	+	+

Andererseits kann auch direkt über die Media Capture API auf das Mikrofon zugegriffen werden. Es können verschiedene Einstellungen zu z.B. Lautstärke, Rauschunterdrücken, Sampling usw. vorgenommen werden.

Spezifikation:

Titel	Link	Status
Media Capture API	<a href="https://www.w3.org/TR/mediacapture-streams/">https://www.w3.org/TR/mediacapture-streams/</a>	Candidate Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
47	36	11	-	+

### 2.1.1.3. Aufnahme von Videos

Die Aufnahme von Videos erfolgt über eine Kombination von Kamera und Mikrofon. Das Web bietet 2 Möglichkeiten zur Aufnahme von Videos:

Einerseits kann ein Datei-Input-Element so konfiguriert werden, dass es nur Videos als Werte akzeptiert. Dieses könnte wie folgt aussehen:

```
<input type="file" accept="video/*">
```

Abhängig vom Betriebssystem und Endgerät wird ein Dateiauswahldialog angezeigt oder eine App zur Aufnahme von Videos geöffnet. Diese Variante wird von allen Browsern unterstützt.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	+	+	+

Andererseits kann auch direkt über die Media Capture API auf Kamera und Mikrofon zugegriffen werden. Neben den in Punkt „2.1.1.2 Aufnahme von Audio“ genannten Einstellungen zum Audio können die Auflösung und die Bildrate konfiguriert werden.

Spezifikation:

Titel	Link	Status
Media Capture API	<a href="https://www.w3.org/TR/mediacapture-streams/">https://www.w3.org/TR/mediacapture-streams/</a>	Candidate Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
47	36	11	-	+

## 2.1.2. Ausgabe von Medien und Informationen

### 2.1.2.1. Anzeigen von Bildern

Die Darstellung von Bildern erfolgt auf einem Bildschirm oder Touchscreen. Das Web bietet 2 Möglichkeiten zur Anzeige von Bildern:

Einerseits können Bildern über das `<img>` Element dargestellt werden. Darstellbar sind sowohl Raster- als auch Vektorgrafiken. Es können die Formate JPEG, GIF, PNG, APNG, SVG, BMP und ICO verwendet werden. Das `<img>` Element wird von allen Browsern unterstützt.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	+	+	+

Andererseits kann ein fortgeschrittenes Rendering über das `<canvas>` Element umgesetzt werden. Dabei können zweidimensionale Grafiken über den 2D Kontext des Elementes und dreidimensionale Grafiken mittels OpenGL ES über die WebGL API erzeugt werden. Sollte ein Bildformat nicht durch das `<img>` Element unterstützt werden, kann dieses über ein Workaround durch den 2D Context des `<canvas>` Elements angezeigt werden.

Spezifikation:

Titel	Link	Status
Canvas 2D Context	<a href="https://www.w3.org/TR/2dcontext/">https://www.w3.org/TR/2dcontext/</a>	Recommendation
WebGL	<a href="https://www.khronos.org/registry/webgl/specs/latest/1.0/">https://www.khronos.org/registry/webgl/specs/latest/1.0/</a>	unbestimmt
WebGL 2.0	<a href="https://www.khronos.org/registry/webgl/specs/latest/2.0/">https://www.khronos.org/registry/webgl/specs/latest/2.0/</a>	unbestimmt

Browser-Support:

Version	Chrome	Firefox	Safari	Internet Explorer	Edge
2D Kontext	+	1.5	2	9	+
WebGL 1	9	4	5.1	11	+
WebGL 2	56	51	-	-	-

## 2.1.2.2. Wiedergabe von Audio

Die Wiedergabe von Audio erfolgt über Lautsprecher. Das Web bietet 2 Möglichkeiten zur Wiedergabe von Audio:

Einerseits können Audiodaten über das `<audio>` Element abgespielt werden. Hierbei können die Lautstärke, das wiederholte Abspielen oder das Vorladen der Audiodaten konfiguriert werden.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
3	3.5	3.1	9	+

Eine feinere Kontrolle über die Wiedergabe des Audios kann über die Web Audio API erreicht werden. Neben den obigen Einstellungen können noch die Tonhöhe (Pitch) und der Panoramaeffekt (Pan) konfiguriert werden. Zudem kann der Ton über einen Oszillator erzeugt, verschiedene Filter angewandt und die Audiokurve visualisiert werden.

Spezifikation:

Titel	Link	Status
Web Audio API	<a href="https://www.w3.org/TR/webaudio/">https://www.w3.org/TR/webaudio/</a>	Candidate Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
14	23	6	-	+

## 2.1.2.3. Wiedergabe von Videos

Die Wiedergabe von Videos war eine lange Zeit im Web nur über ein Plugin mit dem Adobe Flash Player möglich. Der Flash Player wird zwar bei vorhandenem Plugin in jedem Browser unterstützt, er wird jedoch aus verschiedenen Gründen mit dem Ende des Jahres 2020 nicht mehr von Adobe aktualisiert und verteilt.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
~	~	~	~	~

Alternativ sollte das mit HTML5 eingeführte und native `<video>` Element genutzt werden. Bei diesem können neben der Höhe und Breite des Darstellungsbereiches auch ein Vorschaubild, das wiederholte Abspielen des Videos und sein Vorladen konfiguriert werden.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
3	3.5	3.1	9	+

## 2.1.2.4. Erzeugung von Vibrationen

Die Erzeugung von Vibrationen erfolgt über einen Vibrationsaktor. Im Web können Vibrationen durch das Endgerät über die Vibration API erzeugt werden. Bei dieser kann die Länge der Vibration, jedoch nicht die Stärke, eingestellt werden. Zudem sind wiederholte Vibrationen und die Erzeugung von Vibrationsmuster möglich.

Spezifikation:

Titel	Link	Status
Vibration API	<a href="https://www.w3.org/TR/vibration/">https://www.w3.org/TR/vibration/</a>	Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
32	16	-	-	-

## 2.1.3. Persistenz

### 2.1.3.1. Speichern von Schlüssel-Wert-Paaren

Das Speichern von Schlüssel-Wert-Paaren (Auch „Key-Value“ genannt) dient dazu, recht simpel strukturierte Daten zu persistieren. Dazu zählen u.a. Einstellungen zu Nutzerpräferenzen. Das Speichern von Schlüssel-Wert-Paaren kann durch die Web Storage API umgesetzt werden. Dabei besteht die Möglichkeit, Daten nur bis Ende der Sitzung (also bis der Browser geschlossen wurde), oder darüber hinaus zu sichern.

Spezifikation:

Titel	Link	Status
Web Storage	<a href="https://www.w3.org/TR/webstorage/">https://www.w3.org/TR/webstorage/</a>	Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
4	3.5	4	8	+

### 2.1.3.2. Speichern von komplexen Daten

Sollen Daten über Schlüssel-Wert-Paare hinaus gesichert werden, kann die IndexedDB API genutzt werden. Diese erlaubt neben dem Speichern von stark strukturierten Daten auch die Ablage von binären Daten, wie z.B. Bilder, Videos usw.



Spezifikation:

Titel	Link	Status
IndexedDB	<a href="https://www.w3.org/TR/2015/REC-IndexedDB-20150108/">https://www.w3.org/TR/2015/REC-IndexedDB-20150108/</a>	Recommendation
IndexedDB 2.0	<a href="https://www.w3.org/TR/IndexedDB-2/">https://www.w3.org/TR/IndexedDB-2/</a>	Recommendation

Browser-Support:

Version	Chrome	Firefox	Safari	Internet Explorer	Edge
IndexedDB	23	10	7.1	10	+
IndexedDB 2.0	48	44	10.1	-	-

### 2.1.3.3. Zugriff auf das Dateisystem

Ein programmatischer Zugriff aus dem Web auf das Dateisystem des Gerätes ist aus Sicherheitsgründen nicht möglich. Es besteht lediglich die Möglichkeit, dass der Nutzer über ein `<input type="file">` Element eine Datei aus dem Dateisystem aktiv hochlädt.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
-	-	-	-	-

### 2.1.4. Kommunikation

#### 2.1.4.1. Mobilfunkbasierte Kommunikation

Für eine mobilfunkbasierte Kommunikation wird eine entsprechende Antenne für die Verbindung zum Mobilfunknetz sowie eine SIM-Karte zur Nutzung der Dienste des Mobilfunkanbieters benötigt.

##### 2.1.4.1.1. Telefonie

Einer dieser Dienste besteht darin, ein Telefonat über das Mobilfunknetz herzustellen. Zwar kann ein Telefonat aus dem Web heraus nicht direkt gestartet werden, eine Telefonnummer kann jedoch über das `tel`-Schema so verlinkt werden, dass sie direkt in einer für Telefonie vorgesehenen Anwendung geöffnet wird. Ein Telefonlink könnte z.B. wie folgt aussehen:

```
<a href="tel:123456789">123456789 anrufen!</a>
```

Spezifikation:

Titel	Link	Status
RFC 3966	<a href="https://tools.ietf.org/html/rfc3966">https://tools.ietf.org/html/rfc3966</a>	Proposed Standard

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	+	+	+

## 2.1.4.1.2. Messaging (SMS)

Ein weiterer Dienst besteht darin, Textnachrichten (SMS) über das Mobilfunknetz zu versenden. Dies ist ebenfalls nur indirekt möglich, indem eine Telefonnummer über das sms-Schema verlinkt wird. Neben der Telefonnummer kann ebenfalls der Nachrichteninhalt spezifiziert werden. Ein SMS-Link könnte z.B. wie folgt aussehen:

```
<a href="sms:123456789?body=hello">SMS senden!</a>
```

Spezifikation:

Titel	Link	Status
RFC 5724	<a href="https://tools.ietf.org/html/rfc5724">https://tools.ietf.org/html/rfc5724</a>	Proposed Standard

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	+	+	+

## 2.1.4.2. Internetbasierte Kommunikation

Eine Verbindung zum Internet kann entweder über einen DSL-Anschluss (kabelgebunden über Ethernet, kabellos über WLAN) oder über das Mobilfunknetz hergestellt werden. Die Internetverbindung bildet dabei die Grundlage einer jeden Webanwendung, weswegen in diesem Kapitel nur fortgeschrittene, internetbasierte Funktionalitäten betrachtet werden.

### 2.1.4.2.1. Abfrage des Netzwerkstatus

Eine Abfrage des Netzwerkstatus kann dazu genutzt werden, um in einer Webanwendung abhängig von der Konnektivität des Gerätes unterschiedlich zu agieren. So kann bspw. bei einer langsamen, mobilen Verbindung anstatt ein großes, hochauflösendes Bild ein komprimiertes Bild geladen werden, dass sich schneller übertragen lässt. Informationen zum Netzwerkstatus können über die Network Information API ermittelt werden. Über diese können die maximale und effektive Downloadrate sowie der Verbindungstyp (Mobilfunk, Ethernet oder WLAN) ermittelt werden. Die Spezifikation zu dieser API wird durch das W3C nicht weitergeführt, wodurch sie nur partiell unterstützt wird.

Spezifikation:

Titel	Link	Status
Network Information API	<a href="http://wicg.github.io/netinfo/">http://wicg.github.io/netinfo/</a>	Work in Progress

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
61	31*	-	-	-

\* nur auf mobilen Endgeräten verfügbar

## 2.1.4.2.2. Kommunikation über das Internet

Ein initiales Laden aller für eine Webseite benötigten Ressourcen bildet den Kern einer jeden Webanwendung und wurde somit schon immer unterstützt, eine Funktion, die jedoch nicht immer vorhanden war, besteht darin, Ressourcen nachträglich zu laden. Diese Funktion wird AJAX (Asynchronous JavaScript and XML) genannt und kann über XHR (XML HTTP Request) umgesetzt werden.

Spezifikation:

Titel	Link	Status
XMLHttpRequest	<a href="https://xhr.spec.whatwg.org/">https://xhr.spec.whatwg.org/</a>	Living Standard

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	1.2	7	+

## 2.1.4.3. Sonstige Kommunikation

### 2.1.4.3.1. Positionsbestimmung

Die Position eines Gerätes kann über das Global Positioning System (GPS) oder anderen, WLAN-basierten Ortungsdiensten bestimmt werden. Die Position auf der Erde wird durch Längen- und Breitengrade definiert. Im Web kann eine Positionsbestimmung über die Geolocation API erfolgen. Über diese API kann eine einmalige oder kontinuierliche Ortung durchgeführt werden.

Spezifikation:

Titel	Link	Status
Geolocation API	<a href="https://www.w3.org/TR/geolocation-API/">https://www.w3.org/TR/geolocation-API/</a>	Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
5	3.5	5	9	+

### 2.1.4.3.2. Bluetooth

Bluetooth ist ein Industriestandard zur Datenübertragung zwischen 2 Geräten über eine kurze Distanz. Zur Nutzung von Bluetooth im Web existiert eine Web Bluetooth API, die auf einer

Spezifikation der Web Bluetooth Community Group basiert. Bei dieser Spezifikation handelt es sich um keinen W3C Standard, weswegen die API lediglich durch Chrome unterstützt wird. Dieser Browser erlaubt es, über die GATT-Protokoll (Generic Attribute) eine Verbindung zu einem anderen Bluetooth-Gerät aufzubauen.

Spezifikation:

Titel	Link	Status
Web Bluetooth	<a href="https://webbluetoothcg.github.io/web-bluetooth/">https://webbluetoothcg.github.io/web-bluetooth/</a>	Work in Progress

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
53*	-	-	-	-

\* benötigt eine HTTPS Verbindung, muss durch den Nutzer initiiert werden

### 2.1.4.3.3. NFC

NFC ist ein weiterer Übertragungsstandard zur kontaktlosen Datenübertragung über wenige Zentimeter per elektromagnetische Induktion. Das W3C hat zwar eine Spezifikation für eine Web NFC API erstellt, diese gilt jedoch als veraltet und wird nicht mehr weiterentwickelt, wodurch diese API von keinem Browser unterstützt wird.

Spezifikation:

Titel	Link	Status
Web NFC API	<a href="https://w3c.github.io/web-nfc/">https://w3c.github.io/web-nfc/</a>	Work in Progress

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
-	-	-	-	-

### 2.1.4.3.4. USB

USB (Universal Serial Bus) ist ein Standard zur physischen Verbindung zweier Geräte. Über USB kann bspw. externe Hardware wie Tastaturen und Mäuse mit einem Computer verbunden werden. Zur Nutzung von USB im Web existiert eine WebUSB API, die auf einer Spezifikation der Web Platform Incubator Community Group basiert. Bei dieser Spezifikation handelt es sich um keinen W3C Standard, weswegen die API lediglich durch Chrome unterstützt wird. Dieser Browser erlaubt es, eine Verbindung zu angeschlossenen USB-Geräten aufzubauen, Metadaten auszulesen und Daten über spezielle USB-Schnittstellen auszutauschen.

Spezifikation:

Titel	Link	Status
Web USB API	<a href="https://wicg.github.io/webusb/">https://wicg.github.io/webusb/</a>	Work in Progress

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
61*	-	-	-	-

\* benötigt eine HTTPS Verbindung

## 2.1.5. Auslesen von Sensoren

### 2.1.5.1. Bewegungssensoren

#### 2.1.5.1.1. Beschleunigungssensor

Ein Beschleunigungssensor erfasst Daten zur Bewegung eines Endgerätes im dreidimensionalen Raum. Diese Bewegungsdaten können über 2 Wege ausgelesen werden:

Einerseits kann auf ein globales `devicemotion` Ereignis reagiert werden, das bei Bewegung des Gerätes ausgelöst wird und die Bewegungsdaten enthält. Dieses Ereignis wird durch die DeviceOrientation Event Spezifikation beschrieben. Die Weiterentwicklung dieser Spezifikation wurde jedoch zugunsten der zweiten, folgenden Möglichkeit eingestellt, wodurch das Ereignis nur sehr partiell unterstützt wird.

Spezifikation:

Titel	Link	Status
DeviceOrientation Event	<a href="https://www.w3.org/TR/orientation-event/">https://www.w3.org/TR/orientation-event/</a>	Discontinued

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	6	4.2*	11**	+

\* nur auf mobilen Endgeräten verfügbar

\*\* nur auf Windows 8.1 und höher

Andererseits bietet die neue Generic Sensor API einen direkten Zugriff auf Sensoren an. Die Generic Sensor API definiert dabei Sensoren im Allgemeinen und wird durch die Accelerometer API für Beschleunigungssensoren spezifiziert.

Spezifikation:

Titel	Link	Status
Generic Sensor API	<a href="https://www.w3.org/TR/generic-sensor/">https://www.w3.org/TR/generic-sensor/</a>	Candidate Recommendation
Accelerometer API	<a href="https://www.w3.org/TR/accelerometer/">https://www.w3.org/TR/accelerometer/</a>	Candidate Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
67	-	-	-	-

## 2.1.5.1.2. Rotationssensor

Ein Rotationssensor erfasst die Daten zur Rotation eines Endgerätes um alle drei Achsen über ein Gyroskop. Diese Rotationsdaten können über 2 Wege ausgelesen werden:

Einerseits kann auf ein globales `deviceorientation` Ereignis reagiert werden, das bei Drehung des Gerätes ausgelöst wird und die Rotationsdaten enthält. Dieses Ereignis wird durch die DeviceOrientation Event Spezifikation beschrieben. Die Weiterentwicklung dieser Spezifikation wurde jedoch zugunsten der zweiten, folgenden Möglichkeit eingestellt, wodurch das Ereignis nur sehr partiell unterstützt wird.

Spezifikation:

Titel	Link	Status
DeviceOrientation Event	<a href="https://www.w3.org/TR/orientation-event/">https://www.w3.org/TR/orientation-event/</a>	Discontinued

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
7	6	4.2*	11**	+

\* nur auf mobilen Endgeräten verfügbar

\*\* nur auf Windows 8.1 und höher

Andererseits bietet die neue Generic Sensor API einen direkten Zugriff auf Sensoren an. Die Generic Sensor API definiert dabei Sensoren im Allgemeinen und wird durch die Gyroscope API für Rotationssensoren spezifiziert.

Spezifikation:

Titel	Link	Status
Generic Sensor API	<a href="https://www.w3.org/TR/generic-sensor/">https://www.w3.org/TR/generic-sensor/</a>	Candidate Recommendation
Gyroscope API	<a href="https://www.w3.org/TR/gyroscope/">https://www.w3.org/TR/gyroscope/</a>	Candidate Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
67	-	-	-	-

## 2.1.5.2. Umgebungssensoren

### 2.1.5.2.1. Lichtsensor

Ein Lichtsensor erfasst die Stärke des Lichtes in der Umgebung des Endgerätes. Diese Umgebungslichtdaten können über 2 Wege ausgelesen werden:

Einerseits kann auf ein globales `devicelight` Ereignis reagiert werden, das ausgelöst wird, wenn sich die Stärke des Umgebungslichtes ändert. Dieses Ereignis wurde durch die Ambient Light Event Spezifikation beschrieben. Die Weiterentwicklung dieser Spezifikation wurde jedoch zugunsten der zweiten, folgenden Möglichkeit eingestellt, wodurch das Ereignis nur sehr partiell unterstützt wird.

Spezifikation:

Titel	Link	Status
Ambient Light Event	<a href="https://www.w3.org/TR/2015/WD-ambient-light-20150903/">https://www.w3.org/TR/2015/WD-ambient-light-20150903/</a>	Discontinued

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
-	62*	-	-	+

\* es muss die Konfiguration „device.sensors.ambientLight.enabled“ auf „true“ gesetzt werden

Andererseits bietet die neue Generic Sensor API einen direkten Zugriff auf Sensoren an. Die Generic Sensor API definiert dabei Sensoren im Allgemeinen und wird durch die Ambient Light Sensor API für Lichtsensoren spezifiziert.

Spezifikation:

Titel	Link	Status
Generic Sensor API	<a href="https://www.w3.org/TR/generic-sensor/">https://www.w3.org/TR/generic-sensor/</a>	Candidate Recommendation
Ambient Light Sensor API	<a href="https://www.w3.org/TR/ambient-light/">https://www.w3.org/TR/ambient-light/</a>	Candidate Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
54	-	-	-	-

### 2.1.5.2.2. Temperatursensor

Ein Temperatursensor erfasst die Temperatur in der Umgebung des Endgerätes. Zum aktuellen Zeitpunkt existiert weder ein Ereignis, über das die Umgebungstemperaturdaten ausgelesen werden können, noch wurde eine Spezifikation für eine Temperatursensor API erstellt.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
-	-	-	-	-

#### 2.1.5.2.3. Magnetfeldsensor

Ein Magnetfeldsensor erfasst das geomagnetische Feld um das Endgerät im dreidimensionalen Raum. Auf den Magnetfeldsensor kann über die Generic Sensor API zugegriffen werden. Die Generic Sensor API definiert dabei Sensoren im Allgemeinen und wird durch die Magnetometer API für Magnetfeldsensoren spezifiziert.

Spezifikation:

Titel	Link	Status
Generic Sensor API	<a href="https://www.w3.org/TR/generic-sensor/">https://www.w3.org/TR/generic-sensor/</a>	Candidate Recommendation
Magnetometer API	<a href="https://www.w3.org/TR/magnetometer/">https://www.w3.org/TR/magnetometer/</a>	Candidate Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
67	-	-	-	-

#### 2.1.5.2.4. Näherungssensor

Ein Näherungssensor erfasst, ob sich das Endgerät in der Nähe von einem Objekt befindet. Auf diese Weise kann beispielsweise ein Touchscreen deaktiviert werden, wenn das Smartphone zum Telefonieren an das Ohr gehalten wird. Auf den Näherungssensor kann über 2 Wege zugegriffen werden:

Einerseits kann auf ein globales `userproximity` Ereignis reagiert werden, das ausgelöst wird, wenn sich das Gerät in die Nähe eines Objektes oder davon wegbewegt wird. Dieses Ereignis wird durch die Proximity Event Spezifikation beschrieben. Die Weiterentwicklung dieser Spezifikation wurde jedoch zugunsten der zweiten, folgenden Möglichkeit eingestellt, wodurch das Ereignis nur sehr partiell unterstützt wird.

Spezifikation:

Titel	Link	Status
Proximity Event	<a href="https://www.w3.org/TR/2015/WD-proximity-20150903/">https://www.w3.org/TR/2015/WD-proximity-20150903/</a>	Discontinued

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
-	62*	-	-	-

\* es muss die Konfiguration „device.sensors.proximity.enabled“ auf „true“ gesetzt werden



Andererseits bietet die neue Generic Sensor API einen direkten Zugriff auf Sensoren an. Die Generic Sensor API definiert dabei Sensoren im Allgemeinen und wird durch die Proximity Sensor API für Näherungssensoren spezifiziert. Obwohl eine solche Spezifikation existiert, wurde sie zum aktuellen Zeitpunkt von keinem Browser umgesetzt.

Spezifikation:

Titel	Link	Status
Generic Sensor API	<a href="https://www.w3.org/TR/generic-sensor/">https://www.w3.org/TR/generic-sensor/</a>	Candidate Recommendation
Proximity Sensor API	<a href="https://www.w3.org/TR/proximity/">https://www.w3.org/TR/proximity/</a>	Candidate Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
-	-	-	-	-

### 2.1.5.2.5. Luftdrucksensor

Ein Luftdrucksensor erfasst den Luftdruck in der Umgebung des Endgerätes. Aus diesen Daten kann dann z.B. errechnet werden, in welcher Höhe sich das Gerät befinde. Zum aktuellen Zeitpunkt existiert weder ein Ereignis, über das die Luftdruckdaten ausgelesen werden können, noch wurde eine Spezifikation für eine Luftdrucksensor API erstellt.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
-	-	-	-	-

### 2.1.5.2.6. Luftfeuchtigkeitssensor

Ein Luftfeuchtigkeitssensor erfasst die Luftfeuchtigkeit in der Umgebung des Endgerätes. Diese Daten können dann z.B. für eine Wetter-App genutzt werden. Zum aktuellen Zeitpunkt existiert weder ein Ereignis, über das die Luftfeuchtigkeitsdaten ausgelesen werden können, noch wurde eine Spezifikation für eine Luftfeuchtigkeitssensor API erstellt.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
-	-	-	-	-

## 2.2. Softwarebasierte Funktionen

Mit softwarebasierten Funktionen sind solche gemeint, die in keinem direkten Zusammenhang mit der Hardware, sondern eher indirekt über die hardwarebasierten Funktionen auf die einzelnen Komponenten zugreifen. Da somit bei entsprechend vorhandener Hardware in den meisten Fällen eine eigene Implementierung möglich ist, werden die unten aufgeführten, softwarebasierten Funktionen als Komfortfunktionen gesehen.

## 2.2.1. Softwarebasierte Kommunikation

### 2.2.1.1. Versenden von E-Mails

Eine E-Mail ist das digitale Gegenstück zu einem herkömmlichen Brief. Das Versenden einer E-Mail erfolgt über einen Mailserver und wird von einer Vielzahl von Dienstleistern als Service angeboten. Die E-Mail selbst kann vor dem Versand in einem Mail-Client erfasst werden. Ein solcher Mail-Client ist auf den meisten Endgeräten standardmäßig installiert und kann über ein Link mit dem `mailto`-Schema angesprochen werden. Über dieses Schema können Empfänger, CC, BCC, Betreff und Nachrichteninhalte spezifiziert werden. Bei Klick auf den Link wird ein vorhandener Mail-Client mit einer vorausgefüllten E-Mail geöffnet, die der Nutzer dann verschicken kann. Ein solcher Mail-Link könnte wie folgt aussehen:

```
<a href="mailto:e@mail.de?subject=Betreff">E-Mail versenden!</a>
```

Spezifikation:

Titel	Link	Status
RFC 6068	<a href="https://tools.ietf.org/html/rfc6068">https://tools.ietf.org/html/rfc6068</a>	Proposed Standard

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	+	+	+

### 2.2.1.2. Push Notifications

Der Großteil der Kommunikation im Internet verläuft so, dass eine Webanwendung als Client eine Anfrage an einen Webserver stellt, welche diese Anfrage dann beantwortet. Die Kommunikation wird also vom Client initiiert. Bei einer Push Notification handelt es sich um eine vom Webserver initiierte Nachricht, die an den Client versendet („gepusht“) werden soll. Solche Push Notifications sind bei herkömmlichen Webanwendungen daran gescheitert, dass die Nachricht nicht zu gestellt werden konnte, wenn die Webanwendung geschlossen wurde. Progressive Web Apps nutzen zur Umsetzung von Push Notifications die Push API in Kombination mit Service Worker. Bei Letzteren handelt es sich um Hintergrundprozesse, die selbst nach Beendigung der Webanwendung weiterlaufen und somit eingehende Nachrichten verarbeiten können. Service Worker sind eine vergleichsweise neuartige Technologie, welche somit nur in modernen Browsern zur Verfügung steht.

Spezifikation:

Titel	Link	Status
Push API	<a href="https://www.w3.org/TR/push-api/">https://www.w3.org/TR/push-api/</a>	Work in Progress
Service Workers	<a href="https://www.w3.org/TR/service-workers-1/">https://www.w3.org/TR/service-workers-1/</a>	Work in Progress

## Browser-Support:

API	Chrome	Firefox	Safari	Internet Explorer	Edge
Push API	40	44	_*	-	17
Service Worker	40	44	11.1**	-	17

\* Safari bietet eine eigene API an. Siehe <https://developer.apple.com/notifications/safari-push-notifications/>.

\*\* auf mobilen Endgeräten ab Version 11.3

### 2.2.1.3. Kommunikation mit nativen Apps

Eine Kommunikation einer Progressive Web App mit nativen Apps kann auf mobilen Plattformen über die Web Share API umgesetzt werden. Über `navigator.share()` können im JavaScript eine URL oder ein beliebiger Text zusammen mit einem Titel mit anderen nativen Apps geteilt werden. Durch Nutzung der API wird der native Dialog zum Teilen von Informationen geöffnet.

#### Spezifikation:

Titel	Link	Status
Web Share API	<a href="https://wicg.github.io/web-share/">https://wicg.github.io/web-share/</a>	Work in Progress

## Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
61*	-	-	-	-

\* nur auf mobilen Endgeräten verfügbar, benötigt eine HTTPS-Verbindung

### 2.2.2. Installation und Offlinebetrieb

Die Installation von Webanwendungen war bis vor kurzer Zeit noch nicht möglich, ein Offlinebetrieb mittels Caching der für die Webseite benötigten Ressourcen konnte jedoch bereits über den sogenannte Application Cache umgesetzt werden. Die Spezifikation zum Application Cache wurde von der WHATWG (Web Hypertext Application Technology Group) erstellt, die Technology ist jedoch veraltet und wird in Zukunft zugunsten der folgenden Möglichkeit zunehmend vom Web entfernt.

#### Spezifikation:

Titel	Link	Status
Application Cache	<a href="https://html.spec.whatwg.org/multipage/offline.html">https://html.spec.whatwg.org/multipage/offline.html</a>	Discontinued

## Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
4	3.5	4	10	+

Progressive Web Apps hingegen können einen Offlinebetrieb über die Nutzung von Service Worker in Kombination mit Caching gewährleisten. Bei Service Worker handelt es sich um Hintergrundprozesse, die als clientseitige Proxy agieren kann. Somit können Service Worker Anfragen der Webanwendung an den Webserver unterbrechen und selbstständig aus dem Cache beantworten.

Mit den Ressourcen der Webanwendung im Cache ist sie quasi bereits auf dem Endgerät gespeichert. Die Integration in das Betriebssystem geschieht durch das Web App Manifest. Dieses Manifest enthält Metainformationen zur Anwendung, wie z.B. Name, Beschreibung oder Icons. Mit diesen Informationen kann die Progressive Web App dann auf den Startbildschirm des Gerätes installiert werden. Zudem kann die Progressive Web App durch das Manifest so konfiguriert werden, dass sie bei Start über das Icon auf dem Startbildschirm immer im Vollbildmodus gestartet wird.

Spezifikation:

Titel	Link	Status
Service Workers	<a href="https://www.w3.org/TR/service-workers-1/">https://www.w3.org/TR/service-workers-1/</a>	Work in Progress
Web App Manifest	<a href="https://www.w3.org/TR/appmanifest/">https://www.w3.org/TR/appmanifest/</a>	Work in Progress

Browser-Support:

API	Chrome	Firefox	Safari	Internet Explorer	Edge
<b>Service Worker</b>	40	44	11.1*	-	17
<b>Web App Manifest</b>	39	47**	11.3***	-	17

\* auf mobilen Endgeräten ab Version 11.3, die Spezifikation ist nur unvollständig umgesetzt

\*\* nur auf mobilen Endgeräten

\*\*\* nur auf mobilen Endgeräten, die Spezifikation ist nur unvollständig umgesetzt

### 2.2.3. Anwendungen zur persönlichen Organisation

Viele Hersteller bieten auf ihren Endgeräten standardmäßig eine Reihe von Anwendungen zur persönlichen Organisation an. In den meisten Fällen kann von einer nativen App aus ohne Probleme auf solche Anwendungen zugegriffen werden, der Zugriff von Webanwendungen ist jedoch stark von den Schnittstellen und Services abhängig, die der jeweilige Hersteller anbietet. Da die Anzahl an Hersteller, die solche Anwendungen bereitstellen, sehr groß ist, beschränken sich die Betrachtungen hier auf die Services von Google und Apple, deren mobile Plattformen Android und iOS aktuell über 95% des Marktanteils ausmachen.

#### 2.2.3.1. Wecker

Über eine Wecker-App können zu bestimmten Zeitpunkten Weckrufe eingestellt werden, die den Nutzer an einer Aufgabe erinnern oder am Morgen aus dem Schlaf wecken sollen. Sowohl Android als auch iOS bieten standardmäßig eine Wecker-App an, in beiden Fällen existiert jedoch keine Schnittstelle, über die auf die jeweilige Wecker-App zugegriffen werden kann.

Eine eigene Entwicklung einer Wecker-Anwendung gestaltet sich ebenfalls als kompliziert. Zwar können über ein Timeout Tätigkeiten zu bestimmten Zeitpunkten ausgeführt werden, wird die Webanwendung jedoch geschlossen, dann wird auch die Tätigkeit nicht mehr ausgeführt. Für eben diesen Fall wurde vom W3C in einer Spezifikation die Web Alarms API beschrieben, die später in die Task Scheduler API umbenannt wurde. Die Arbeit an dieser Spezifikation wurde im Jahr 2015 eingestellt.

Spezifikation:

Titel	Link	Status
Task Scheduler API	<a href="https://www.w3.org/TR/task-scheduler/">https://www.w3.org/TR/task-scheduler/</a>	Discontinued

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
-	-	-	-	-

### 2.2.3.2. Kalender

Eine Kalender-App dient der zeitlichen Organisation von Ereignissen. Google bietet mit Google Calendar eine Anwendung für digitale Kalender an, die sich problemlos in die Android Plattform integrieren lässt. Zudem bietet Google für seine Kalenderanwendung eine Calendar API an, worüber diese problemlos in Webanwendungen integriert werden können.

Google Calendar API: <https://developers.google.com/calendar/>

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	+	+	+

Apple bietet mit iCloud Calendar ebenfalls eine Kalenderanwendung an, die sich problemlos in die iOS Plattform integrieren lässt. Im Gegensatz so Google stellt Apple jedoch keine öffentliche API für den Zugriff auf iCloud Calendar bereit, wodurch ein Zugriff aus dem Web nicht oder nur anhand von internen Informationen möglich ist.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
-	-	-	-	-

### 2.2.3.3. Kontaktliste

Eine Kontaktliste dient der Organisation von Kontakt- und weiteren Informationen zu bekannten Personen. Auf mobilen Endgeräten dient die Kontaktliste in den meisten Fällen auch als Ausgangspunkt für das Durchführen von Telefonaten und Schreiben von Nachrichten.

Die in Android integrierte Kontaktliste nennt sich Google Contacts und lässt sich aus dem Web heraus problemlos über die sogenannte People API ansprechen.

People API: <https://developers.google.com/people/>

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	+	+	+

Die in iOS integrierte Kontaktliste nennt sich iCloud Contacts. Apple selbst stellt keine öffentliche API zum Zugriff auf iCloud Contacts bereit, wodurch ein Zugriff aus dem Web nicht oder nur anhand von internen Informationen möglich ist.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
-	-	-	-	-

### 2.2.3.4. Notizen

Eine Notizenanwendung dient der Erfassung von kurzen Texten, wodurch sich der Nutzer ggfs. später an einen Sachverhalt erinnern oder den Inhalt nicht vergessen will. Da sich die Anwendungsfälle von Notizenanwendungen teilweise mit denen von anderen Anwendungen (z.B. Kalender) überschneiden, sind sie vermutlich nicht allzu populär. Google bietet auf der Android Plattform standardmäßig eine Notizen-App mit dem Namen Google Keep an, eine Web API dazu existiert jedoch nicht. Apple bietet standardmäßig keine Notizen-App an.

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
-	-	-	-	-

### 2.2.4. Landkarten

Kartenanwendungen sind das digitale Gegenstück zu herkömmlichen Straßenkarten und spielen in der heutigen Zeit eine große Rolle in der Navigation.

Recht weit verbreitete Kartenanwendung sind z.B. Google Maps, Apple Maps oder OpenStreetMap. Diese Kartendienste bieten eine Vielzahl an Funktionen an, die aus dem Web heraus über verschiedene APIs angesprochen werden können. So kann beispielsweise eine Karte, die eine bestimmte Position anzeigt, in die eigene Webseite integriert werden. Es kann bspw. auch die eigene Position, die über die Geolocation API (Vgl. Kapitel „**Fehler! Verweisquelle konnte nicht gefunden werden.**“) ermittelt wurde, angezeigt werden. Ein weiterer Service ist das Geocoding, bei welchem eine Positionsangabe in Längen- und Breitengraden in eine für den Menschen lesbare Ortbezeichnung (z.B. Straße und Hausnummer) umgewandelt wird. Das

Geofencing, bei dem potenziell interessante Orte in der Nähe des Nutzers ermittelt werden, ist ebenfalls möglich.

Google Maps und OpenStreetMaps können über eine eigene JavaScript API und Apple Maps über das sogenannte MapKit JS Framework in eine Webanwendung integriert werden.

Google Maps Platform: <https://cloud.google.com/maps-platform/>

MapKit JS Framework: <https://developer.apple.com/documentation/mapkitjs>

OpenStreetMap API: <https://wiki.openstreetmap.org/wiki/API>

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	+	+	+

### 2.2.5. Vertrieb

Für Android- und iOS-Apps stehen der Google Play Store oder der Apple App Store für einen Vertrieb der Anwendungen zur Verfügung. Dieser direkte Verkauf über App-Stores ist bei Webanwendungen nicht möglich, da diese frei über das Internet zugänglich sind. Stattdessen kann der Verkauf der Progressive Web App über In-App Käufe oder über eine derartige Gestaltung umgesetzt werden, dass die relevanten Funktionen hinter einem Login in einen geschützten Bereich versteckt werden, wobei dann der Zugang zu diesem Bereich verkauft wird.

Für die Umsetzung beider Varianten kann die vom W3C spezifizierte Payment Request API genutzt werden, über die durch den Browser gesteuert ein Zahlungsprozess ausgelöst werden kann. Die Payment Request API verspricht dabei eine schnelle Durchführung, eine einfache Bedienbarkeit durch konsistentes Verhalten, Barrierefreiheit und eine sichere Handhabung der Kontodaten.

Da die Payment Request API ein recht junger Webstandard ist, wird dieser noch nicht von allen Browsern unterstützt. Kann die API in einem Browser nicht genutzt werden, können alternativ die üblichen, in den Online-Shops genutzten Zahlungswege umgesetzt werden.

Spezifikation:

Titel	Link	Status
Payment Request API	<a href="https://www.w3.org/TR/payment-request/">https://www.w3.org/TR/payment-request/</a>	Candidate Recommendation

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
61	55*	11.1	-	15

\* es muss die Konfiguration „dom.payments.request.enabled“ auf „true“ gesetzt werden

## 2.2.6. Text- und Sprachverarbeitung

### 2.2.6.1. Sprachsynthese (Text to Speech)

Bei der Sprachsynthese wird geschriebener Text in eine Audiodatei umgewandelt, die den gesprochenen Inhalt des Textes enthält. Die Sprachsynthese kann beispielsweise für Screen Reader zum Zwecke der Barrierefreiheit genutzt werden.

Eine Möglichkeit der Sprachsynthese bietet die Web Speech API. Diese API kann direkt über den JavaScript Code der Webanwendung angesprochen werden, sie wird jedoch nicht von allen Browsern unterstützt.

Spezifikation:

Titel	Link	Status
Web Speech API	<a href="https://w3c.github.io/speech-api/">https://w3c.github.io/speech-api/</a>	Work in Progress

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
33	49	7	-	+

Eine andere Möglichkeit besteht in der Nutzung eines Webservice, dem der zu synthetisierende Text übergeben wird und welcher die Audiodaten zurückgibt. So bietet Google z.B. eine Sprachsynthese über seinen Cloud Text-to-Speech Service an.

Google Cloud Text-to-Speech: <https://cloud.google.com/text-to-speech/>

Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	+	+	+

### 2.2.6.2. Spracherkennung (Speech to Text)

Bei der Spracherkennung wird die Sprache in einer Audiodatei analysiert und in Text umgewandelt. Diese Funktion kann bspw. für eine effiziente Transkription auf aufgenommenen Interviews genutzt werden.

Eine Möglichkeit der Spracherkennung bietet die Web Speech API. Diese API kann direkt über den JavaScript Code der Webanwendung angesprochen werden, sie wird jedoch nicht von allen Browsern unterstützt.

Spezifikation:

Titel	Link	Status
Web Speech API	<a href="https://w3c.github.io/speech-api/">https://w3c.github.io/speech-api/</a>	Work in Progress



## Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
25*	44**	-	-	-

\* die Spracherkennung wird über einen Webserver durchgeführt

\*\* es muss die Konfiguration „media.webspeech.recognition.enable“ auf „true“ gesetzt werden

Eine andere Möglichkeit besteht in der Nutzung eines Webservice, dem der zu erkennende Text übergeben wird und welcher transkribierten Text zurückgibt. So bietet Google z.B. eine Spracherkennung über seinen Cloud Speech-to-Text Service an.

Google Cloud Speech-to-Text: <https://cloud.google.com/speech-to-text/>

## Browser-Support:

Chrome	Firefox	Safari	Internet Explorer	Edge
+	+	+	+	+

## 3. Gutachten und Korrektur

Die folgenden Gutachter haben durch ein Review zur Korrektur des Regelwerks beigetragen:

- Achim Gosse  
digitalnoise GmbH  
[achim.gosse@digitalnoise.de](mailto:achim.gosse@digitalnoise.de)
- Kevin Taron  
Gutwerker Keppler/Taron GBR  
[k.taron@gutwerker.de](mailto:k.taron@gutwerker.de)
- Peter Kröner  
Trainer und Spezialist für Webtechnologien  
<https://www.peterkroener.de/>