

Wie kommt der Elefant ins Internet ?

oder:

Responsive Images, Bildformate und Optimierung

von Felix Kirmaier

Über mich

Felix Kirmaier

Frontendentwickler aus
München

TYPO3 Integrator

RWD Hardliner

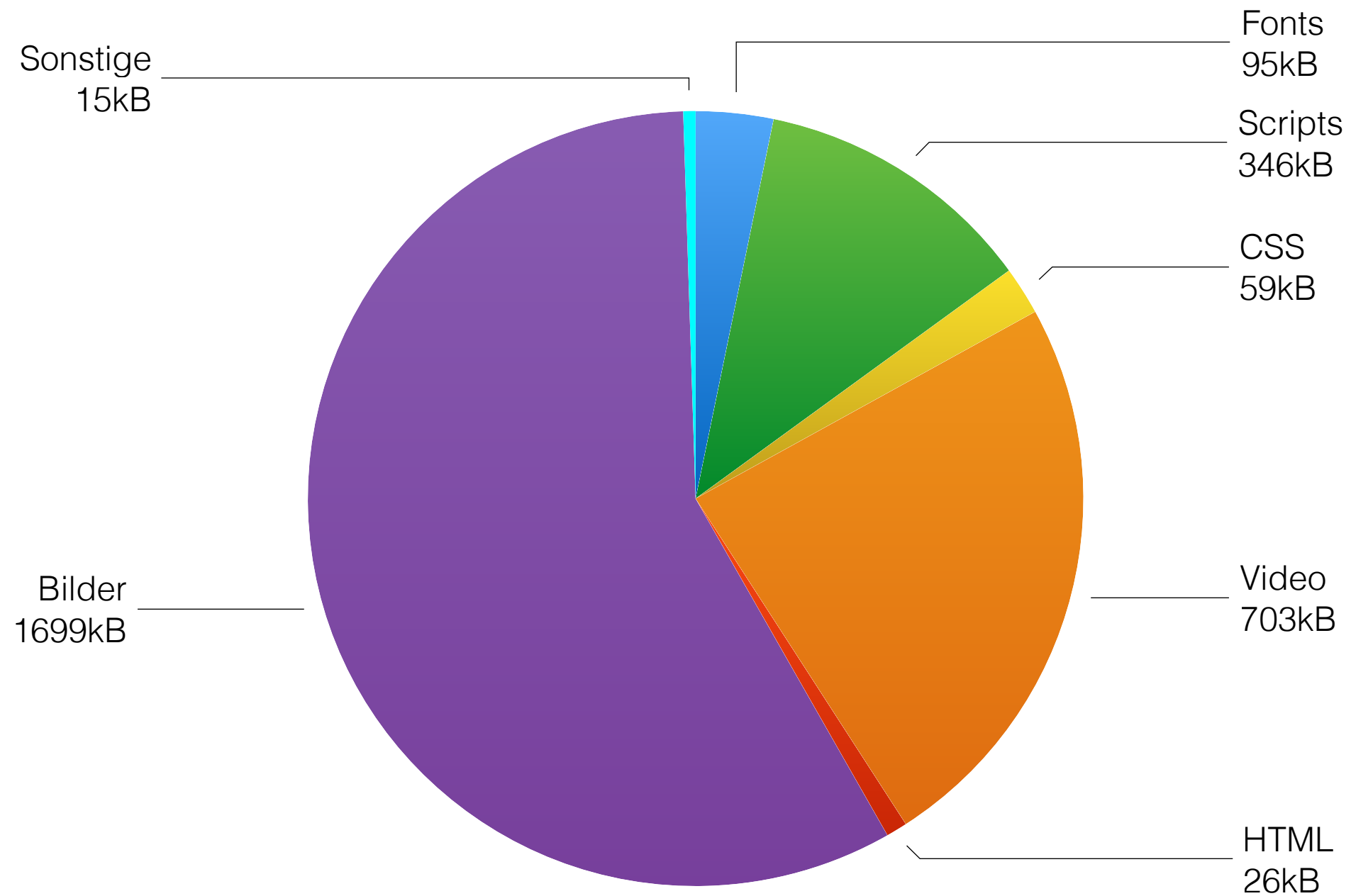
@felixk76

[https://www.xing.com/
profile/Felix_Kirmaier](https://www.xing.com/profile/Felix_Kirmaier)



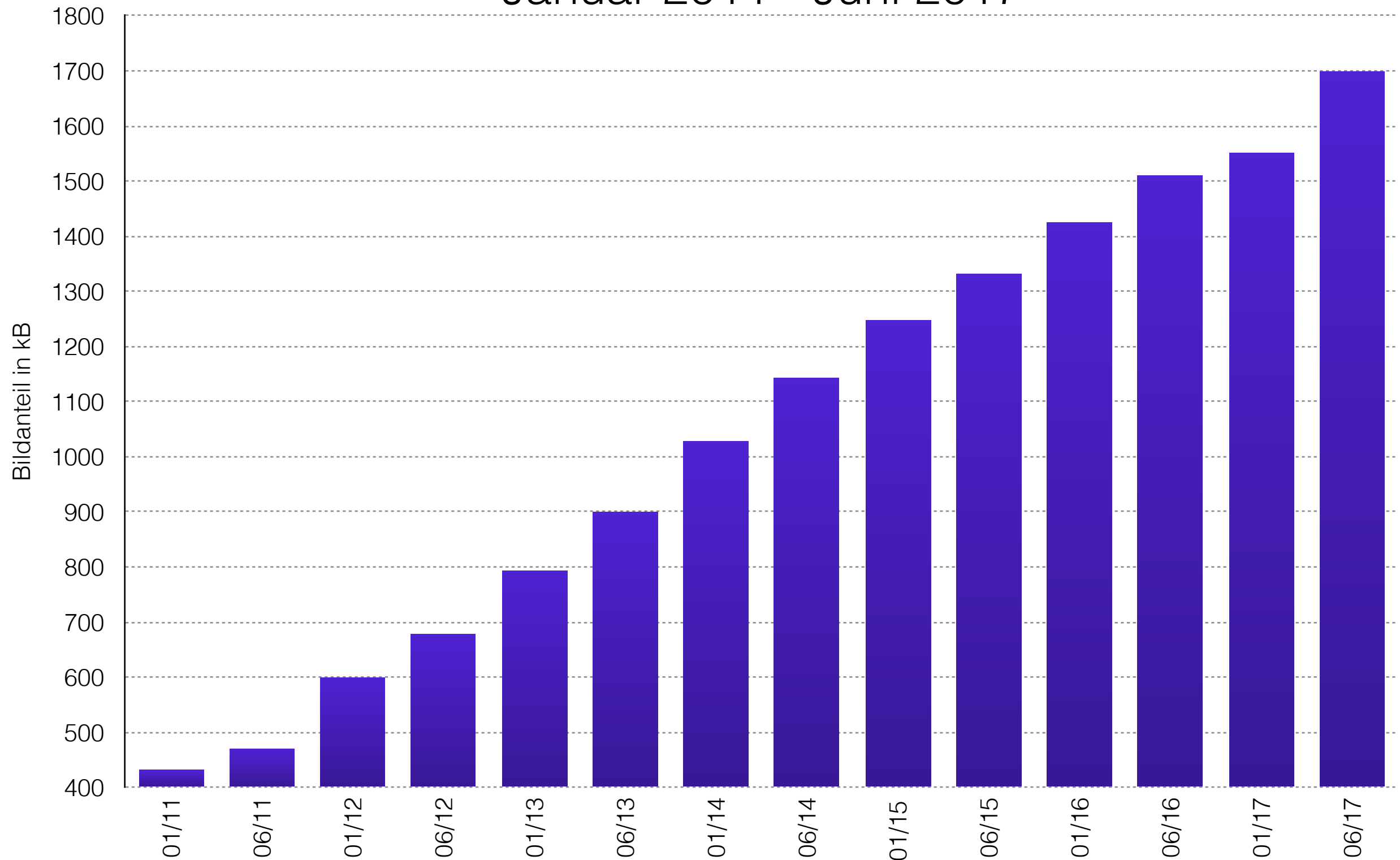
Warum muss der Elefant ins
Internet ?

Warum ?



Entwicklung Bildgröße

Januar 2011 - Juni 2017



Typische Fehler

- falsches Bildmaterial
- falscher Grund
- falsche Bildgröße
- falsches Dateiformat
- falsche Komprimierung

Falsches Bildmaterial

- Bildmaße zu klein
- Qualität zu schlecht
- Bildrechte unklar
- Dateiformat / Motiv zur Weiterverarbeitung ungeeignet

Falscher Grund

- „Ein Bild sagt mehr als 1000 Worte“
- Headerbilder ohne Bezug
- Seitenfüllende Hintergrundbilder oder -videos
- Unnötige Transparenzen, Schatten und sonstige Effekte
- Animationen, „Daumenkinos“
- Überladene Seiten (zig Slides oder Klickstrecken)
- Überladene Navigation

Falsche Bildgröße



PNG 1280 x 960 1890kB



PNG 400 x 300 227kB

Falsches Dateiformat



PNG 400 x 300 227kB



JPG 400 x 300 47kB

Falsche Komprimierung



JPG 400 x 300 47kB



JPG 400 x 300 17kB

Wie macht man es besser ?

Materialfragen zu Projektstart

- Haben die Bilder die benötigte Qualität und Größe ?
- Reicht das Budget für ein Fotoshooting und oder Stockphotos ?
- Welche Bildquelle liegen vor ?
- Wieviele Bildmaße werden maximal benötigt ?
- Müssen Bilder für Retinadisplays ausgegeben werden ?
- Welche Bilder müssen pflegbar sein ?

Bildauswahl

- Weniger ist oftmals mehr
- „Deko“ vermeiden
- Performance Budget beachten
- Die passende Technologie wählen
- Userverhalten tracken

Bildgröße

Anwendungsfälle für Responsive Images

- feste Bildgröße
 - Variable Bildgröße
- } **Resolution Switching Usecase**
- Unterschiedliche Bilder
 - Kein Bild
- } **Art Direction Usecase**

... und verschiedenste Mischformen.

Resolution Switching Usecase

Das bekannte - Tag

```

```

Responsive Image mit srcset

```

```

Responsive Image mit srcset und sizes

```

```

Warum das sizes-Attribut benötigt wird

Ohne srcset / sizes:

?	Entwickler bei Erstellung	Browser beim Laden
Viewportgröße	✗	✓
Bildgröße relativ zum Viewport	✓	✗
Pixeldichte des Devices	✗	✓
Bildgröße (px)	✓	✗

Warum das sizes-Attribut benötigt wird

Mit srcset / sizes:

?	Entwickler bei Erstellung	Browser beim Laden
Viewportgröße	✗	✓
Bildgröße relativ zum Viewport	✓	✓ (sizes)
Pixeldichte des Devices	✗	✓
Bildgröße (px)	✓	✓ (srcset)

Art Direction Usecase



<picture> - Tag

```
<picture>
  <source
    srcset="image-1280.jpg 1280w"
    media="(min-width: 769px)"
    sizes="(max-width:1280) 100vw, 1280px">
  <source
    srcset="image-768.jpg 768w"
    media="(min-width: 321px)"
    sizes="100vw">
  <source
    srcset="image-320.jpg 320w"
    sizes="100vw">
  
</picture>
```


Welche Bildgrößen ?

- Eine perfekte praktikable Lösung gibt es noch nicht
- unabhängig von Layout Breakpoints
- Ein Performance Threshold erstellen.
- Für repräsentative Bilder die Bildgrößen ermitteln
- Wichtige Devicegrößen abdecken

<http://www.responsivebreakpoints.com/>

Dateiformate

Die gängigsten Bildformate fürs Web

Format	Typ	Transparenz	Animation	Browser
GIF	Lossless	✓	✓	alle
JPG	Lossy	✗	✗	alle
PNG	Lossless	✓	✗	alle
SVG	Lossless	✓	✓	(alle)

GIF (Graphic Interchange Format)

- von Steve Wilhite entwickelt, 1987 von CompuServe eingeführt
- erstes nichtproprietäres, plattformübergreifendes Farbformat
- kleine Dateigröße durch LZW Kompression
- seit 2012 als Nomen und Verb in der englischen Sprache anerkannt. „to gif“ wurde „Wort des Jahres“
- seit Mai 2015 erlaubt Facebook animierte GIFs in der Chronik

PNG (Portable Network Graphics)

- Erster Entwurf von Thomas Boutell am 4. Januar 1995 vorgestellt
- Ursache und Ziel war es eine lizenzfreie Alternative zu GIF zu erhalten
- meistverwendetes verlustfreies Bildformat im WWW

JPEG

(Joint Photographic Experts Group)

- 1991 von Eric Hamilton entwickelt
- JPEG ist die Norm, die verschiedene Bildkompressionsverfahren beschreibt
- JFIF und EXIF sind die verbreitetsten Dateiformate, die diese anwenden
- gängigstes Dateiformat für Fotos
- 45% aller Image Requests, 65% des Image Traffics im Internet

SVG (Scalable Vector Graphics)

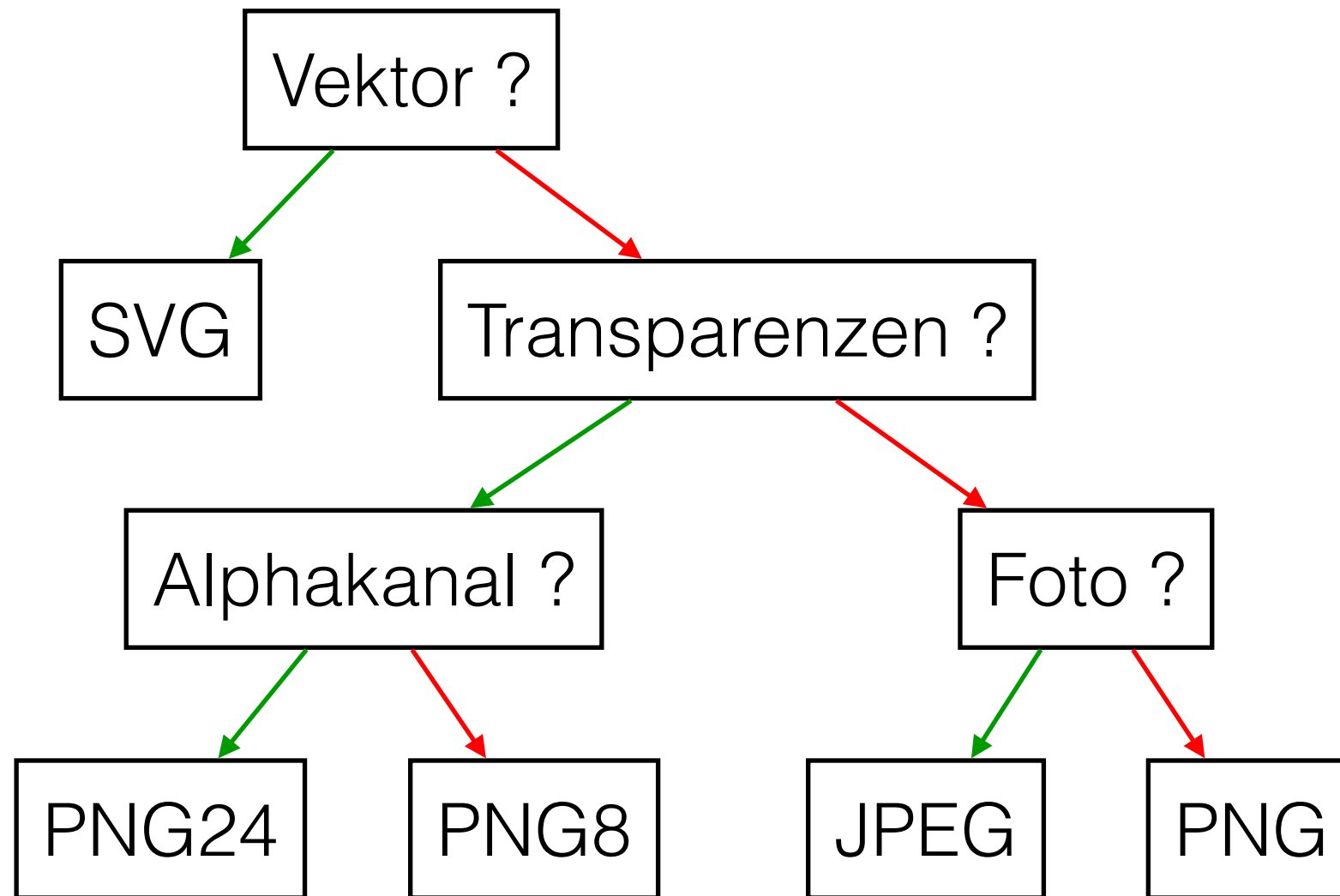
- in Entwicklung durch das W3C seit 1999
- XML-Format
- Kann mit CSS und JS modifiziert werden
- Seit 15. September 2016 existiert eine Candidate Recommendation ,die seit Juli 2017 eine Recommendation sein sollte

Das richtige Dateiformat wählen

Abhängig vom:

- Quellformat
- Motiv (Farbanzahl)
- Darstellungsgröße
- Verwendungszweck
- Dateigröße des Ausgabeformats

Entscheidungshilfe Bildformat



Browserspezifische Bildformate



Fast überflüssiger Vergleich

Format	Typ	Transparenz	Animation	Browser
WebP	Lossless Lossy	✓	✓	Chromium, Firefox, Safari Opera
JPEG 2000	Lossless Lossy	✓	✓	Safari
JPEG XR	Lossless Lossy	✓	✓	IE9, Edge
HEIF	Lossless Lossy	✓	✓	Safari 11
AVIF	Lossless Lossy	✓	✓	Chrome, Firefox

WebP

- wird von Google seit 2010 entwickelt
- zufällig bei der Entwicklung von WebM entdeckt
- unterstützt von Chrome, Opera und Android (Firefox & Safari testen noch)
- Facebook brach Test mit WebP ab, weil User mit den Bildern nicht anfangen konnten

JPEG 2000

- Erstveröffentlichung Dezember 2000
- beliebt in Medizintechnik und Digitalkino
- Patentsituation schwierig

JPEG XR

- JPEG eXtended Range
- Mai 2006 vorgestellt als Windows Media Photo
- November 2006 umbenannt in HD Photo
- schlechte Softwareunterstützung

HEIF

(High Efficiency Image File Format)

- entwickelt von der Moving Picture Experts Group (MPEG)
- vorgestellt Dezember 2006
- unterstützt von iOS11 und MacOS High Sierra
- kann Bilder, Bildsequenzen und Audio enthalten

AVIF

(AOMEDIA VIDEO IMAGE FILE FORMAT)

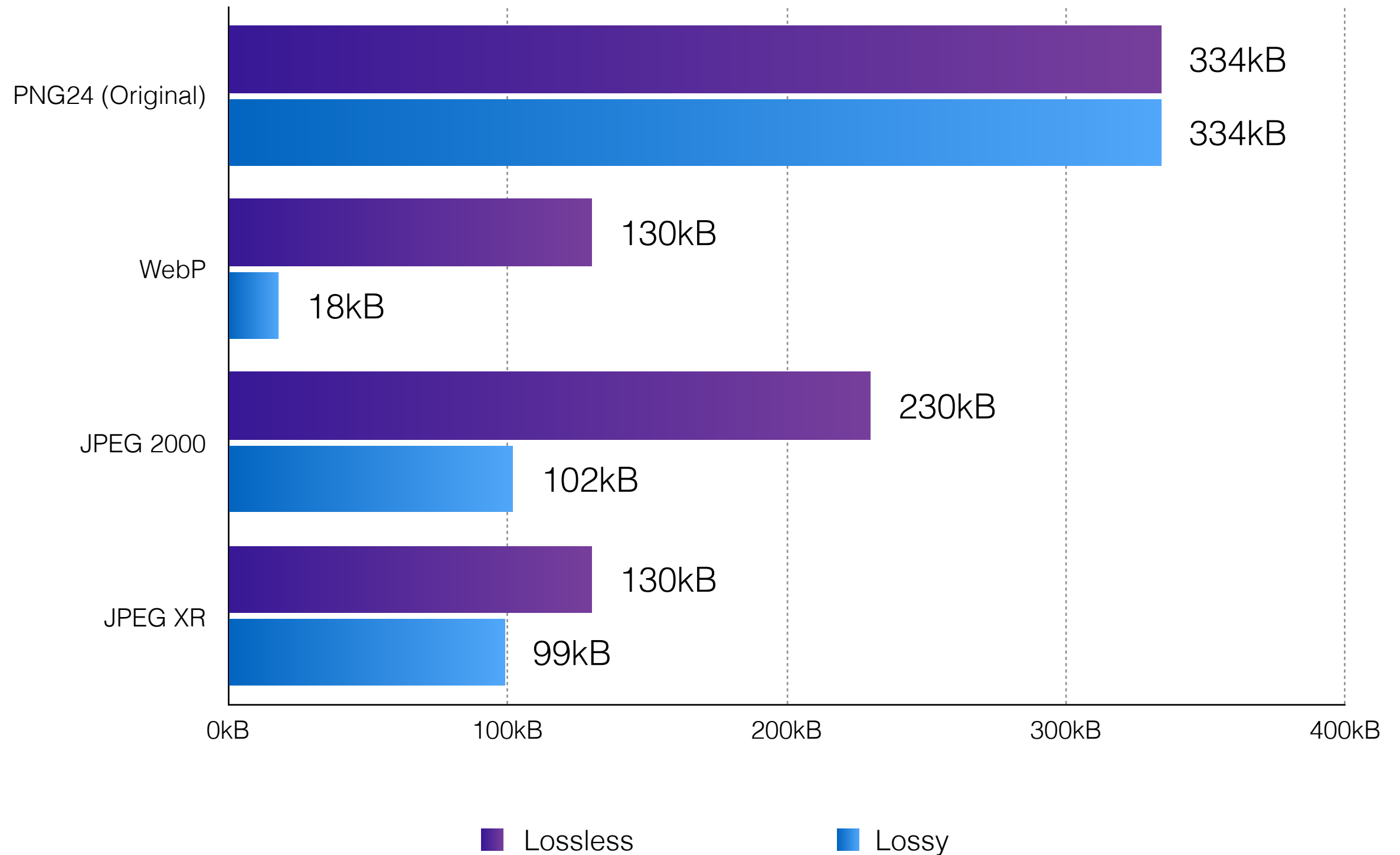
- entwickelt von Alliance for Open Media (AOMedia)
- vorgestellt Dezember 2018 von Netflix
- unterstützt von Chrome 85 und Firefox (flag)

Testbild PNG24



- 400 x 400, 334kB
- direkt aus Photoshop
- PNG24
- Transparenz
- ohne Interlacing
- ohne Metadaten

Konvertierung



Verwendung im HTML

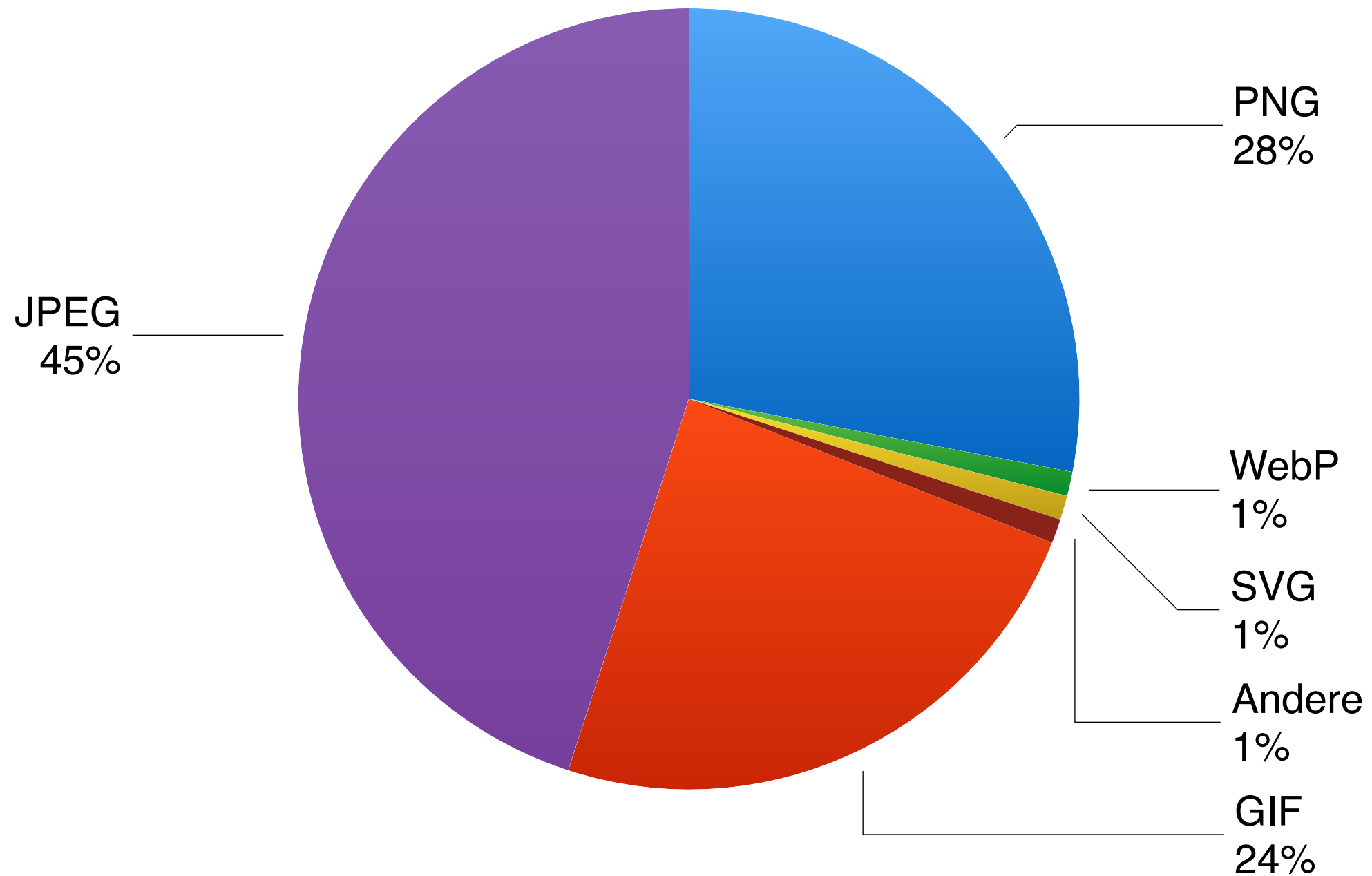
```
<picture>
  <source
    srcset="image-small.webp 320w, image-large.webp 1024w"
    sizes=" (max-width: 20em) 30vw, 100vw"
    type="image/webp" >
  <source
    srcset="image-small.png 320w, image-large.png 1024w"
    sizes=" (max-width: 20em) 30vw, 100vw"
    type="image/png" >
  
</picture>
```

Verwendung im CSS

```
.no-js .backgroundimage,  
.no-webp .backgroundimage {  
    background: url('image.png');  
}
```

```
.webp .backgroundimage {  
    background: url('image.webp');  
}
```

Verteilung im Netz



Komprimierung

Mit Rücksicht auf Verluste

- Keine Angst vor Lossy Komprimierung
- Ohne Referenzbild wird die Qualität meist besser eingeschätzt als sie ist
- Niemand schaut sich die Bilder so genau an wie Dein Designer
- Viele Faktoren beeinflussen die Qualitätswahrnehmung
- Daumenregel: Schlechte Qualität + X

PNG Optimizer

- pngcrush <https://pmt.sourceforge.io/pngcrush/>
- OptiPNG <http://optipng.sourceforge.net/>
- PNGOUT <http://advsys.net/ken/utils.htm>
- pngquant <https://pngquant.org/>
- ZopfliPNG <https://github.com/google/zopfli>

JPEG Optimizer

- jpegtran
- MozJPEG <https://github.com/mozilla/mozjpeg>
- Guetzli <https://github.com/google/guetzli>
- JPEG recompress <https://github.com/danielgtaylor/jpeg-archive>

Optimizer Tools

- Imageoptim
- Imagemin
- TinyPNG
- JPEGmini
- Cloudinary
- compressor.io

Imageoptim

- Desктоptool für MacOS, CLI, Web Service und Onlinetool
- Lossless / Lossy
- gut konfigurierbar
- Open Source
- API kostenpflichtig

Imagemin

- Kommandozeilentool
- viele Plugins
- gut automatisierbar (z.B grunt, gulp)

TinyPNG

- Onlinetool, API, Photoshop Plugin
- sehr gute Optimierung von PNG und JPEG
- 2 TYPO3 Extensions vorhanden
- Automatischer Motivabhängiges Cropping per API

JPEGmini

- Online, App für PC und Mac, Server, AWS
- optimiert nur JPEGs
- mit Ausnahme der Website alles Kostenpflichtig
- Kann in der Pro Version auch Bilder skalieren

Clouldinary

- Cloudlösung
- Medien Management
- Bildmanipulation
- Bild- und Videokonvertierung
- API und SDKs für diverse Sprachen

Testvergleich (PNG)

Tool	Lossless	Lossy
ImageOptim	1.12MB (961kB)	542kB
Imagemin	1.885MB	558kB
ImageAlpha	-	537kB
TinyPNG	-	466kB
Cloudinary	1.933MB	557kB
compressor.io	1.912MB	468kB
pngcrush	1.83MB	-
OptiPNG	1.908MB	-
PNGOUT	1.899MB	-
pngquant	-	589kB
Zopflipng	961KB	

Testvergleich (JPEG)

Tool	Lossless	Lossy
Imageoptim	844kB (808kB)	262kB
Imagemin	631kB	-
JPEGmini	694kB	-
TinyPNG	-	314kB
Cloudinary	606kb	415kB
compressor.io	-	515kB
jpeg.io	845kB	-
jpegtran	1,1MB	-
mozjpeg	1,0MB	399kB
guetzli	-	808kB
jpeg-recompress	-	661kB

Test-Tool

Cloudinary Website Speed Test

<https://webspeedtest.cloudinary.com/>

Fazit

- „überflüssige“ Bilder vermeiden
- Bildmaterial prüfen
- Bildformate (Abmessung) vereinheitlichen
- „Bildbreakpoints“ ermitteln
- Das beste Dateiformat wählen
- Browserspezifische Dateiformate anbieten
- Bilder IMMER optimieren
- Automatisieren

Fragen ?