web p
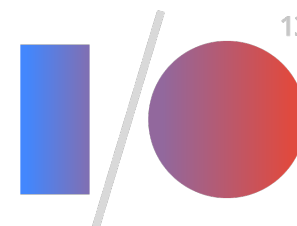
enabling faster, smaller and more beautiful web

+Stephen Konig   skonig@google.com

+Ilya Grigorik   igrigorik@google.com

https://developers.google.com/speed/webp/

13

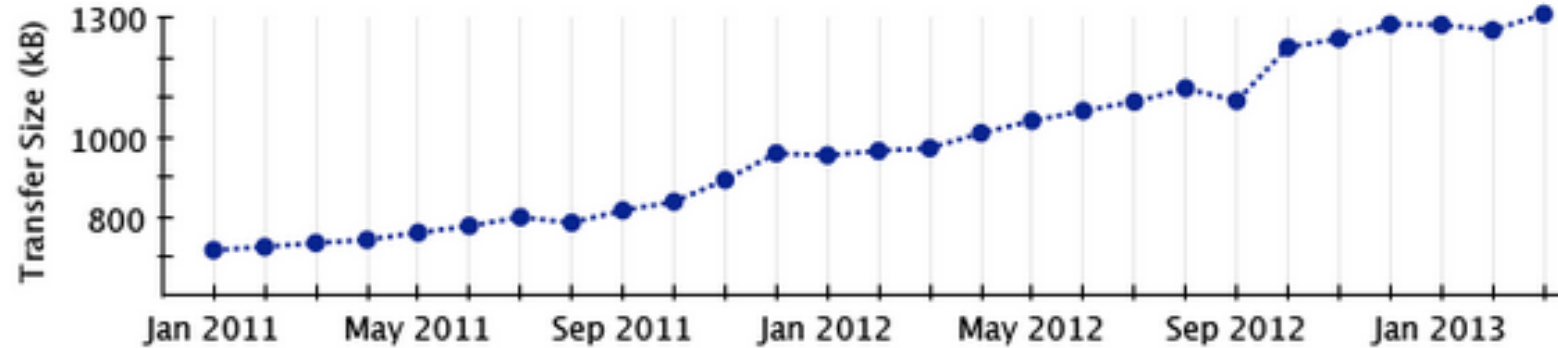# For an average page, images account for...

- **60% of transferred bytes** for desktop sites
- **69% of transferred bytes** for mobile sites

🙏

aka, the bulk of the page!
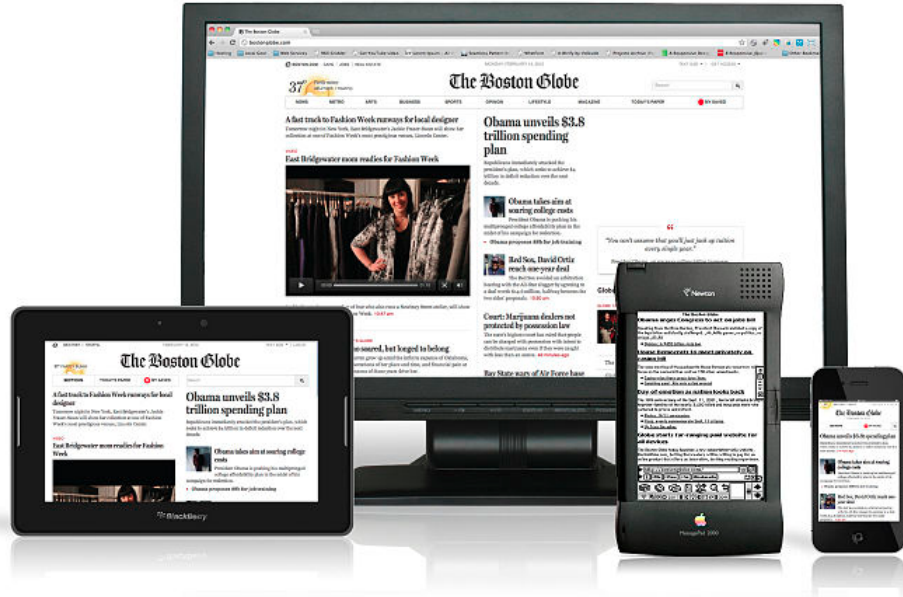
500-1000 KB of image bytes, and growing, fast!

Content Type	Desktop		Mobile	
	Avg # of requests	Avg size	Avg # of requests	Avg size
Images	56	856 KB	38	498 KB
Javascript	15	221 KB	10	146 KB
HTML	10	56 KB	6	40 KB
CSS	5	36 KB	3	27 KB
Total	86+	1169+ KB	57+	711+ KB

Ouch!

It's a **HiDPI** world...

Device	px/inch
Kindle Fire	~ 170
iPad Mini	~ 160
Nexus 7	~ 216
Macbook + Retina	~ 220
Chromebook Pixel	~ 239

Device
pixels

x4

HiDPI screens require 4x pixels ... and ~4X bytes?

web●p

1. **Improved data compression**
2. **Lossy and lossless modes**
3. **Alpha** **channel, animation, ...**

Faster, smaller and more beautiful images.

Brief history of **WebP**...

- **WebM** video format uses VP8 video codec
- **WebP is derived from VP8**, essentially a key frame...

- **Web{P,M}** are open-source, royalty-free formats
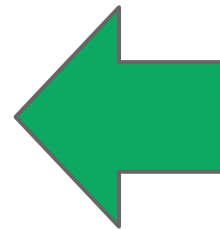 - Open-sourced by Google in 2010
  - BSD-style license

Brief history of WebP...

- **Initial release in 2010**
  - Lossy compression for true-color graphics

- **August, 2012**
  - Lossless compression support
  - Transparency (alpha channel) support

- **April, 2013**
  - Color profile
  - Animation
  - Metadata

Now ready for general purpose use, on the web, and elsewhere!

We are not done yet! *Looking forward...*

- Performance optimizations
- Better support for ARM and mobile
- High color depth images (> 8 bits)
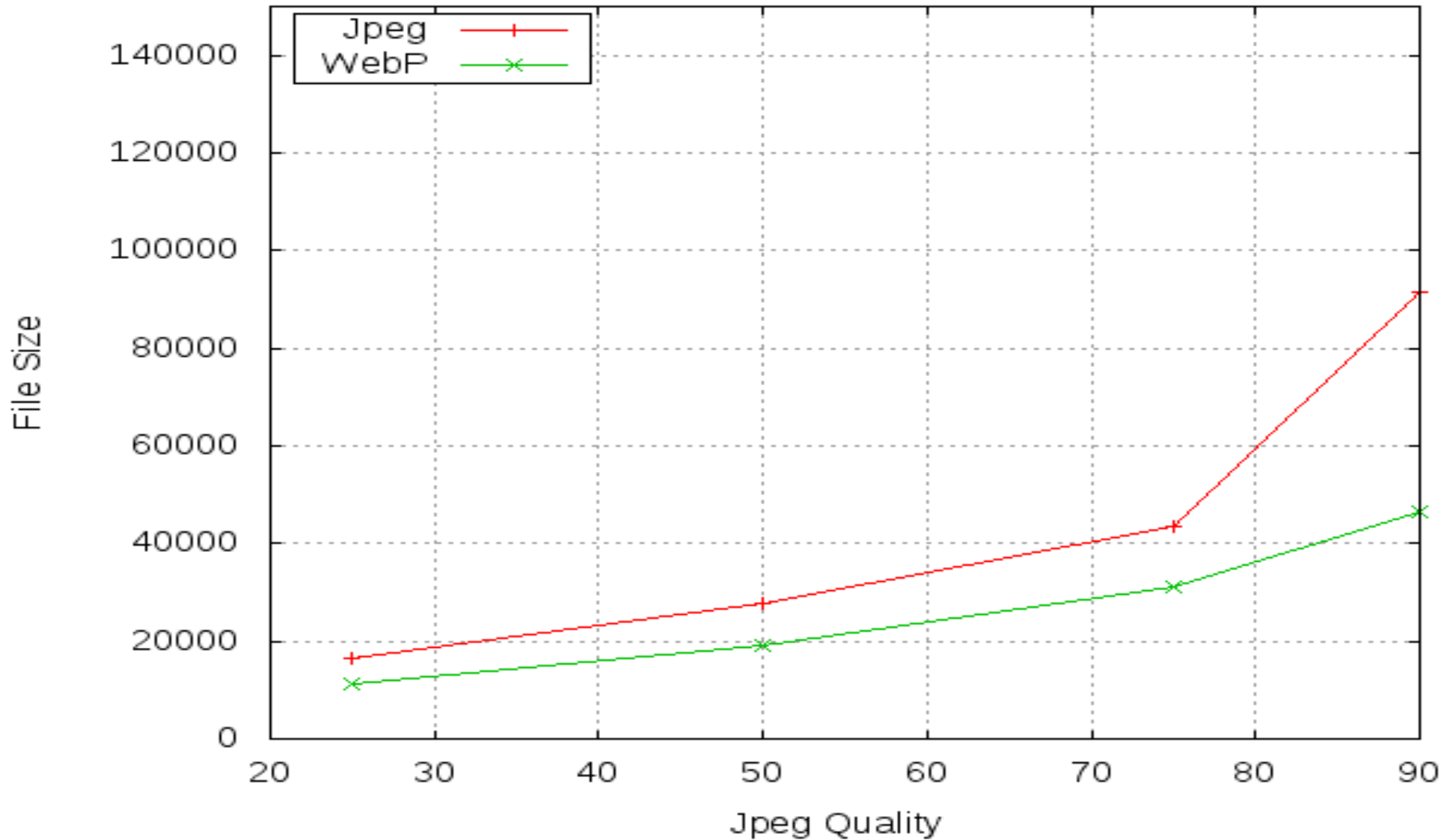- Layer support (3D images)
- Progressive rendering
- ...

web○p performance...

Let's take a look at the numbers, and a few real-world case studies!
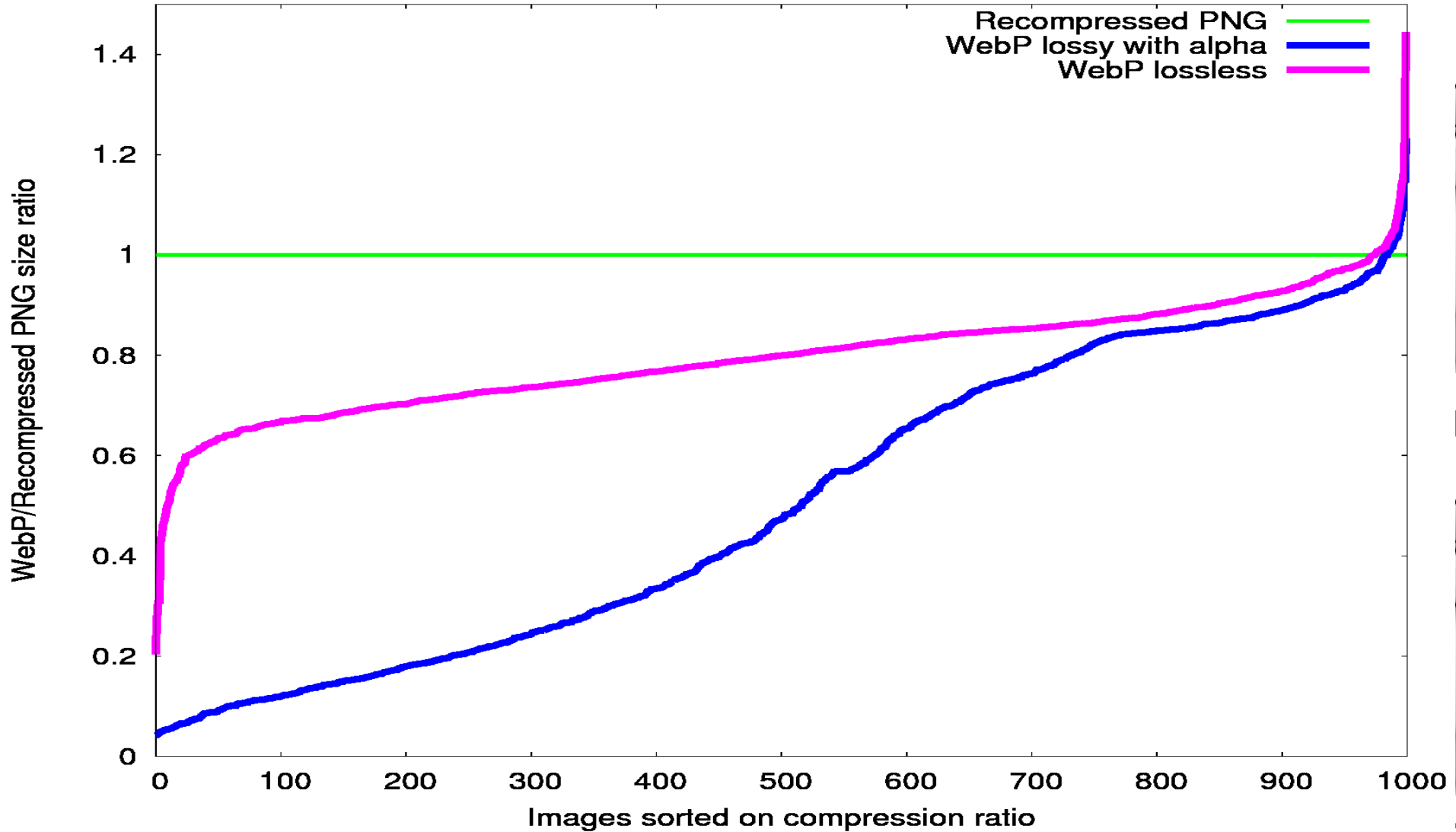
WebP *vs.* JPEG (Lossy compression)

80   1300KB

10   200KB

WebP *vs.* PNG (Lossless compression)

- Sor
- But
  - ○

0.5KB

7.3KB

https://developers.google.com/speed/webp/docs/webp_lossless_alpha_study

# Which image format should I use?

Wrong question! ***WebP supports...***

- Lossy and lossless compression
- Transparency (alpha channel)
- Great compression for photos
- Animations
- Color profiles
- Metadata
- ...

Encoding and Decoding speeds

- Software performance today, for lossy...
  - **Encoding:** 5-10x slower than JPEG   (done once)
  - **Decoding:** ~1.3x slower than JPEG   (per decode)

- High encoding cost may be a limitation for use cases where images are generated dynamically

- **Bandwidth savings** vs. extra CPU time
  - Many users are on metered data plans
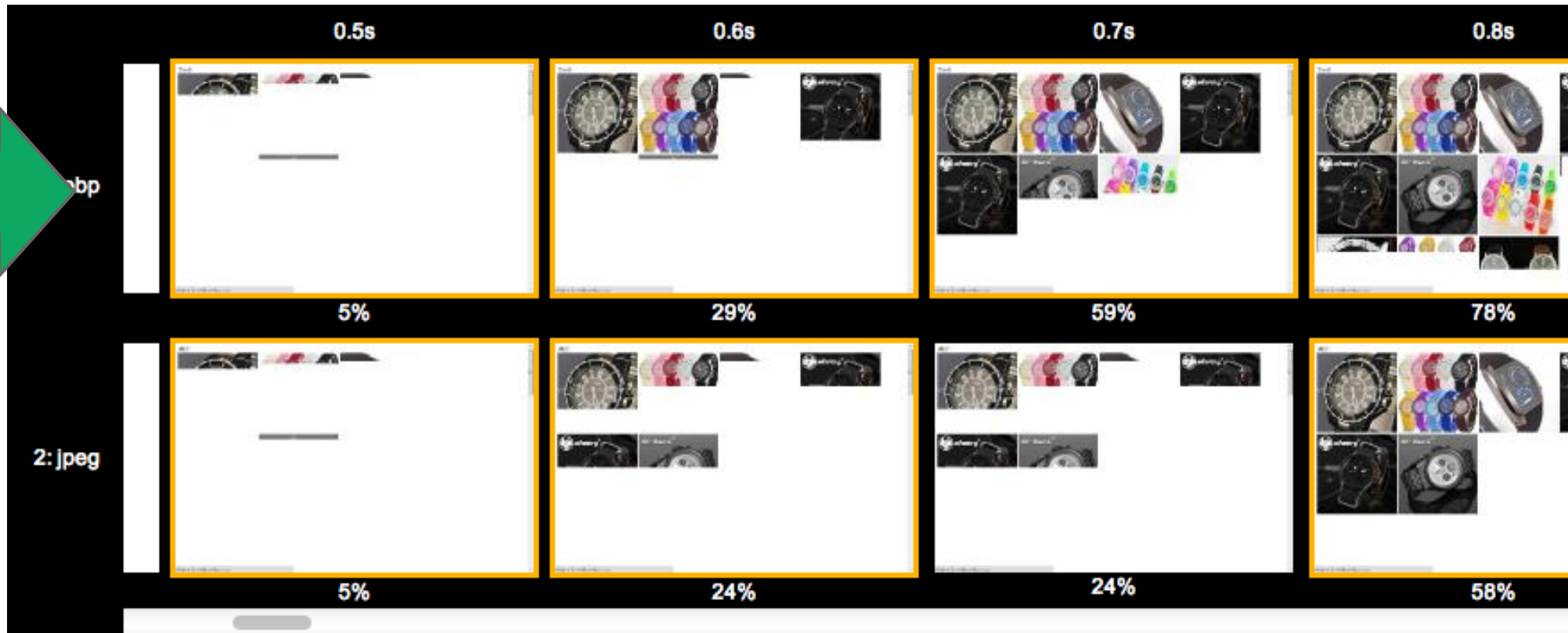 - **Data is expensive** - literally!
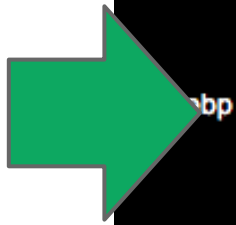   - $1+ per MB per user!
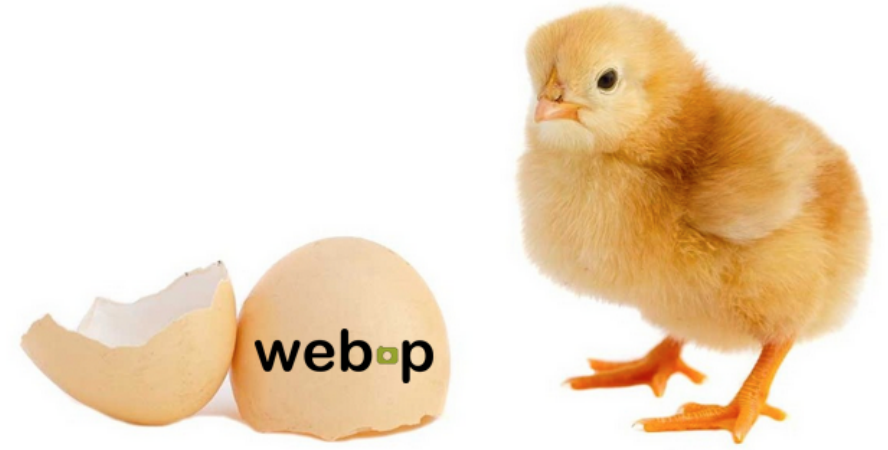
WebP performance will improve with further optimizations and hardware support.

"A picture is worth a thousand words" - Ebay tech blog

"Because the WebP page had to download fewer bytes (474 KB vs. 757 KB), it ***completes loading much earlier*** *compared to the JPEG page."*

Tooling and deployment

Let's take a look at how to get started with WebP...

How do I create a **WebP** file?

- Download [WebP converter](#) (Linux, OSX, Windows)
  - `cwebp -q 80 image.png -o image.webp`
  - `dwebp image.webp -o image.png`

- Download [WebP Codec for Windows](#) (Photo Viewer, Explorer, Office 2010+, ...)
- Download [Photoshop plugin](#) (by Telegraphics)
- Download [GIMP plugin](#)
- ImageMagick, Pixelmator, XnView, IrfanView, GDAL, JPEGView have **native support** for WebP

- Java, .NET, Flash, Python, Ruby, PHP bindings available to libwebp...
- [img2webp.net](#) online tool

Check here for more

http://en.wikipedia.org/wiki/WebP#Support

State of **WebP** adoption today

	Android	iOS	Chrome	Opera	IE	Safari	Firefox
WebP	4.x.x+	Library	✔	✔	JS	[Plugin]() or JS	WIP Patch or JS

- 2010-2012: focus on feature support + performance.
- ***2013+ ... focus on adoption and deployment!***

- **Chrome**, **Opera**, and working closely with Firefox team...
- 3rd party plugins for **Safari** and **IE** (Chrome Frame)
- **JavaScript decoder** fallback (libwebp.js)

- **Android** lossy (ICS+) and lossless (JB+)
- **iOS** native apps via libraries

Deploying WebP on the web...

Let's get hands on!

awesome.webp

⬇

User-Agent
(Chrome, FF, IE, ...)

↘   ↙

Server Detection
Accept + User-Agent

Client Detection
JavaScript

⬇   ⬇

Custom HTML
Cache-Control: private

⟷

Inject <img> via JS
Extra latency for img fetch

hybrid

Client-side detection

- Use modernizr, or use the [1 line WebP detect](#) function...

```
<script src="modernizr.min.js"></script>

<script>
  if (Modernizr.webp) {

    var webpImg = document.createElement("img");
    webpImg.setAttribute('src', '/awesome.webp');
    webpImg.setAttribute('alt', 'na');
    document.body.appendChild(webpImg);

  } else {
    // Fallback to non-webp, or load a JS decoder:
    // webpjs-0.0.2.min.js / webpjs-0.0.2.swf
  }
</script>
```
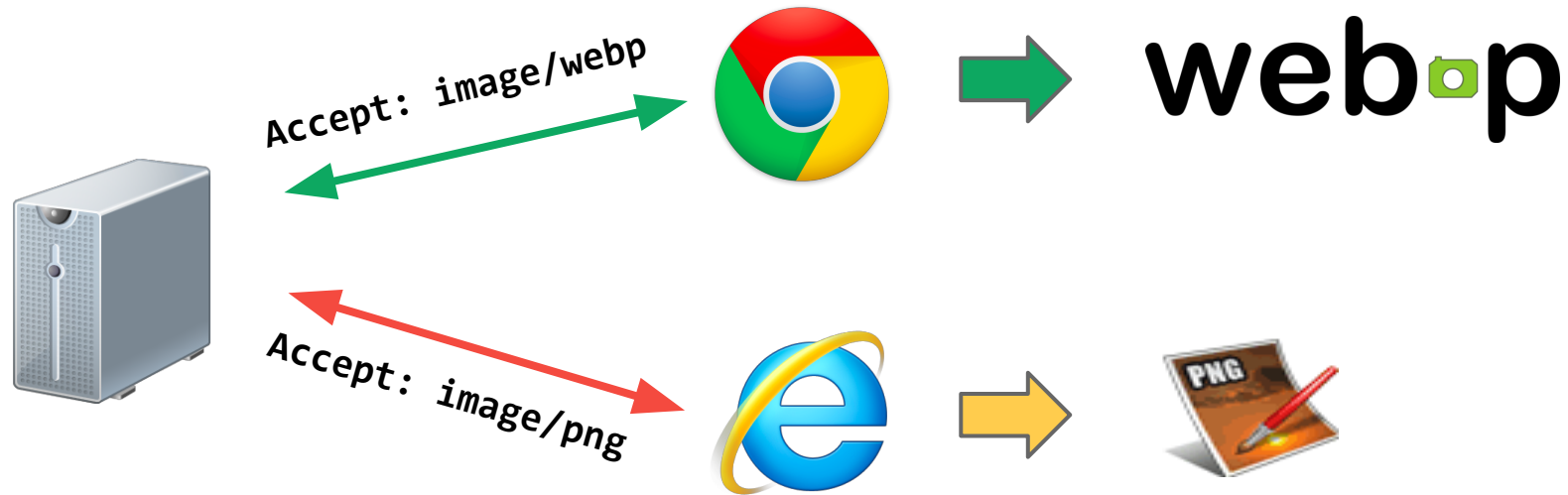
+ Bullet proof, custom URLs for .webp files (cache friendly), easy fallback for all clients

- Must wait for JS execution to schedule image downloads

http://webpjs.appspot.com/

Client-Server **Accept** negotiation

*Serve different HTML based on **Accept** header...*

+ No extra latency overhead
+ Fully transparent to your existing application!

Server-side User-Agent detection

*Serve different HTML based on **Accept** header*

```
<html>
 ...
 <img src="awesome.webp">
 ...
<html>
```

Cache-Control: private

```
<html>
 ...
 <img src="awesome.jpg">
 ...
<html>
```

Or inject a polyfill library!

+ No extra latency overhead, automated by the server (e.g. PageSpeed)

- Returned HTML should be marked with "**Cache-Control: private**"

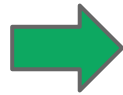Deploying new image formats on the web

Dealing with **interoperability** ...

Facebook tries Google's WebP image format; users squawk

It's a major endorsement for the file format, but some social-network members are upset to have lost their flexible, sharable JPEGs.

- **Link sharing**: send a link to a **.webp** image to a friend running IE ... sad face.
  - Use **Accept** negotiation to serve correct asset!

- **Save As**: save **.webp** file locally; no way to open it?
  - Fixed: Chrome is now a file handler for .webp!
  - Provide an explicit 'Download' option, and link to JPEG / PNG.
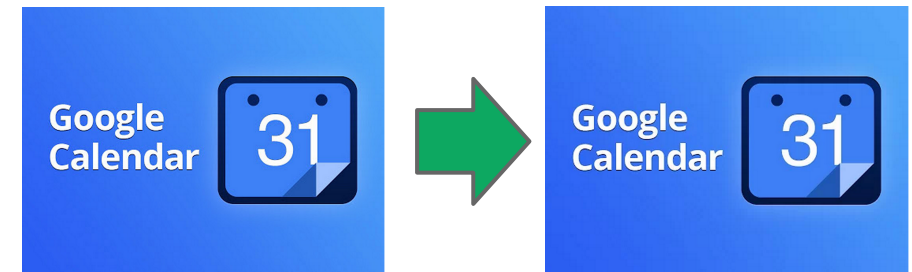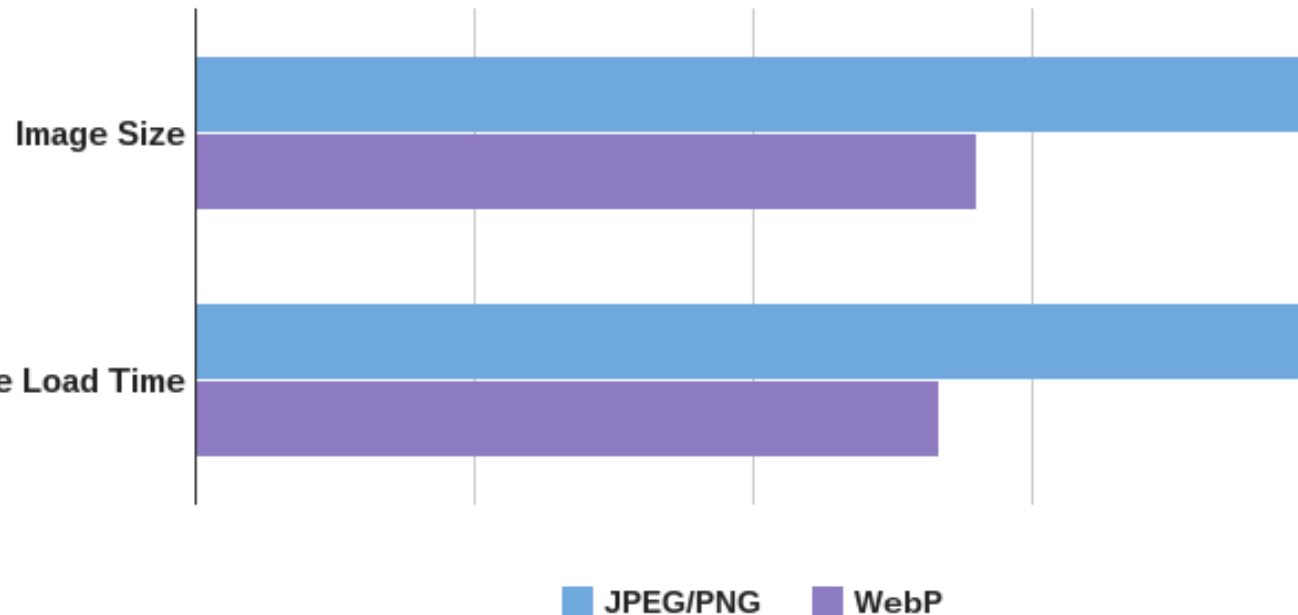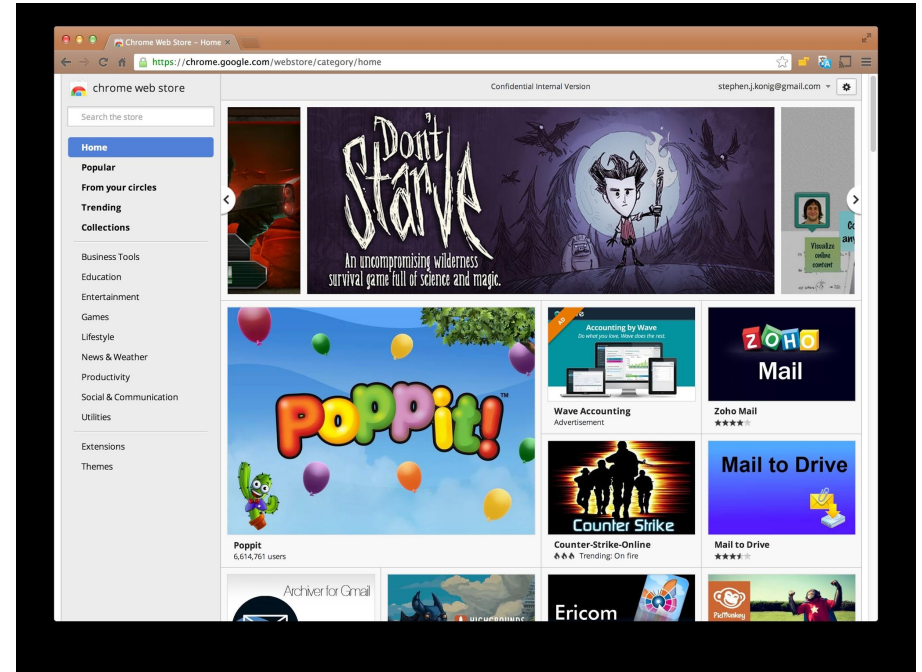
http://news.cnet.com/8301-1023_3-57580664-93/facebook-tries-googles-webp-image-format-users-squawk/

WebP case studies in the **wild...**

<insert your site here>

Chrome Web Store

- Image-heavy site with lots of promotional tiles

- Switching from JPEG/PNG to WebP yielded a **30% reduction** in image size on average and reduced page load times by **one-third.**

JPEG

32 KB

WebP

8.3 KB

Deployed and in progress... *Dogfood++*
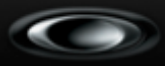
Data compression proxy for chrome

- Original content **>** PageSpeed **> WebP >** your device

- Early tests show **50% data compression improvement**, and faster load times!
  a. Download Chrome Beta on Play Store
  b. Enable **"Reduce data usage"**

Secure site

Google SPDY Proxy

HTTP site

Bandwidth Usage	Session	Total
Original (KB)	172915	172915
Received (KB)	64805	64805
Savings (KB)	108110	108110
Savings (%)	62	62

https://developers.google.com/chrome/mobile/docs/data-compression

I AM INVINCIBLE

INTRODUCING THE NEW AIRBRAKE MX GOGGLE

BUY ON~~~   ~~~N STORES   >

SITE of the DAY

86MB page!
LOTS of PNG's.

28MB page with WebP's
still massive, but... 58MB savings!

moto.oakley.com

↑
.01
.02
.03
.04

Fun tools to play with...

Windows

online

IMG2WebP.net

Drag an image anywhere to upload.

Allowed file types: jpg, jpeg, png and gif.

Maximum upload file size: 1MB.

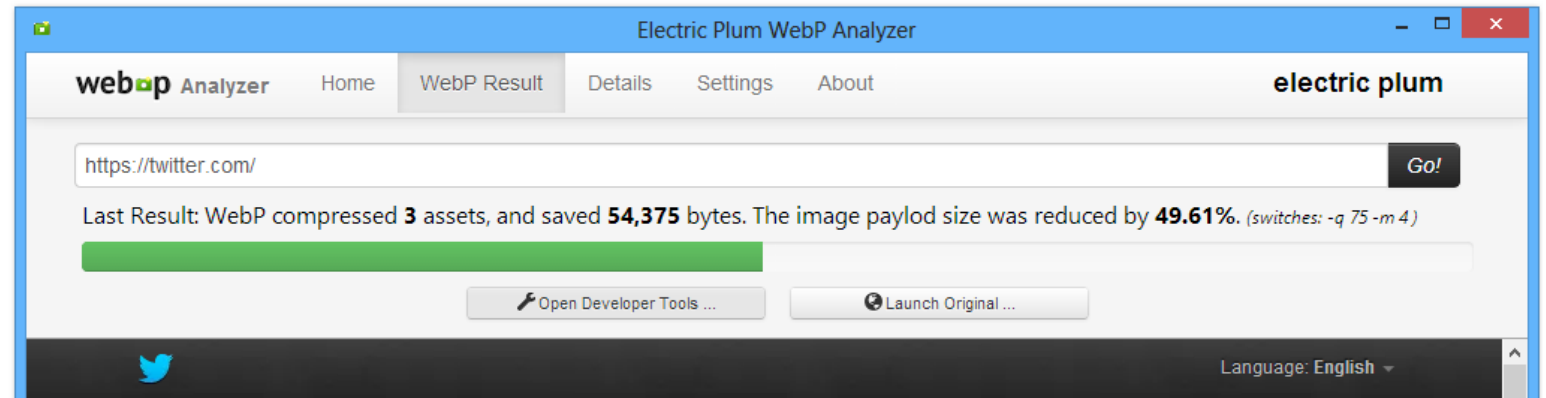Advanced Settings

Copyright © 2013 | About

http://www.electricplum.com/webp.aspx

Electric Plum WebP Analyzer

web◦p Analyzer   Home   WebP Result   Details   Settings   About   electric plum

https://twitter.com/   Go!

Last Result: WebP compressed **3** assets, and saved **54,375** bytes. The image paylod size was reduced by **49.61%**. *(switches: -q 75 -m 4)*

Open Developer Tools ...   Launch Original ...

Language: English

Automate...   GitHub

PUBLIC   somerandomdude / grunt-webp

Google+ on **Android**

- Photos and images comprise the vast majority of bytes
- On average, got **50% byte savings** with WebP!

- Saving many terabytes of bandwidth per day…
- **Saving our users money** each time they use the app!

File size ratio
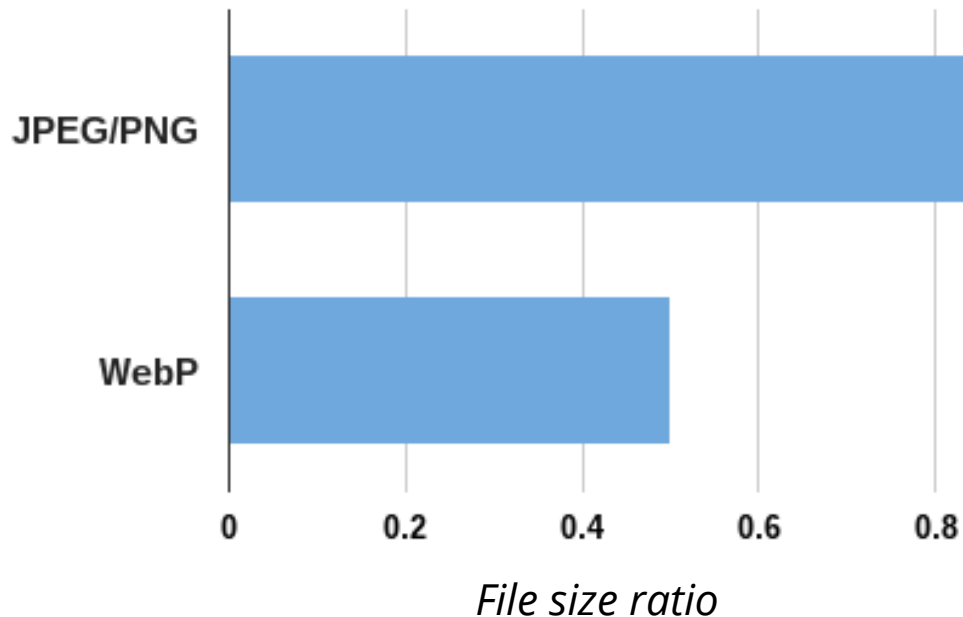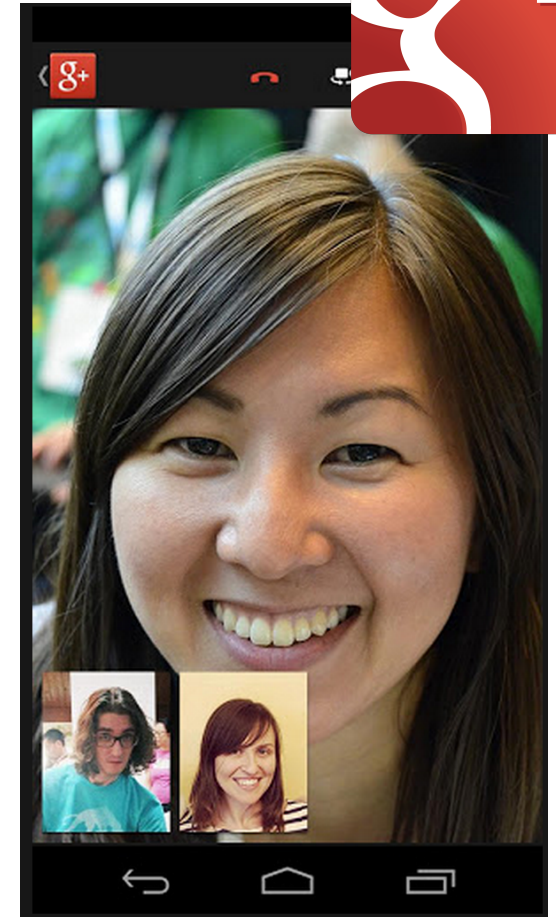
Android

Native library -- all versions of Android

```java
static {
    System.loadLibrary("webp");
}

private Bitmap webpToBitmap(byte[] encoded) {

    int[] width = new int[] { 0 };
    int[] height = new int[] { 0 };
    byte[] decoded = libwebp.WebPDecodeARGB(encoded, encoded.length, width, height);

    int[] pixels = new int[decoded.length / 4];
    ByteBuffer.wrap(decoded).asIntBuffer().get(pixels);

    return Bitmap.createBitmap(pixels, width[0], height[0], Bitmap.Config.ARGB_8888);
}
```

webp-android-backport for Android <4.0

iOS

Download and compile libwebp, add **WebP.framework** to your project...

```objc
// Get the current version of the WebP decoder
int rc = WebPGetDecoderVersion();

NSLog(@"Version: %d", rc);

// Get the width and height of the selected WebP image
int width = 0;
int height = 0;
WebPGetInfo([myData bytes], [myData length], &width, &height);

NSLog(@"Image Width: %d Image Height: %d", width, height);

// Decode the WebP image data into a RGBA value array
uint8_t *data = WebPDecodeRGBA([myData bytes], [myData length], &width, &height);
```

- Walkthrough tutorial
- WebP-iOS-example demo application on GitHub

Let's recap...

awesome.webp

Android / iOS   **Web**

Android >4.0: native
Android <4.0: backport
iOS: WebP.framework

Server Detection   Client (JS) Detection

#protip: server-side automation FTW!

PageSpeed

Video @ bit.ly/io-webp

web🔲p

+Stephen Konig   skonig@google.com
+Ilya Grigorik   igrigorik@google.com

https://developers.google.com/speed/webp/
Mailing list: webp-discuss