Project Fugu

Next level Progressive Web Applications

https://giftkugel.github.io/talks/

KARAKUN

Simon Skoczylas

Senior Software Engineer

Java

JavaScript / TypeScript

Web technologies

giftkugel
www.skoczylas.net

Recap of

Progressive Web Applications

Web Applications

Easy access (URL)

References (External links)

Searchable (SEO)

Composeable (embedded via iframe)

Good Web APIs

Progressive Web Applications

Can install the application

Work offline / Background Sync

Notify the user (Push notifications)

Building blocks of a PWA

Web App Manifest
https://w3c.github.io/manifest/

Service Worker API
https://w3c.github.io/ServiceWorker/

HTTPS, HTTP/2 😍

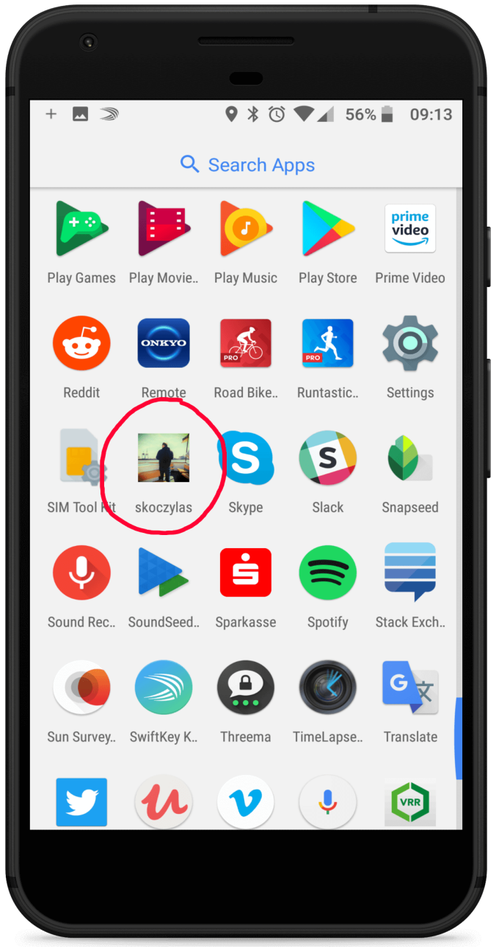
Progressive enhancement approach

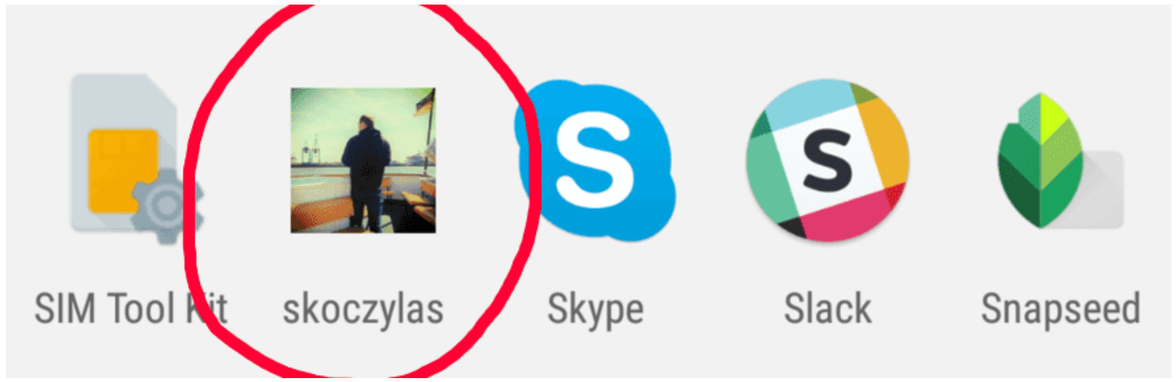
Progressive enhancement approach
Feature detection

```
if ('foo' in window) {
    // Cool! Use new feature! 🥳
} else {
    // D'oh! Legacy approach is required. 😔
}
```

Because some features will probably never be implemented by all browser vendors

Recap





Listed as application in the launcher

Twitter Datei Bearbeiten Anzeigen Verlauf Fenster

Simon Skoczylas (@giftkugel) / Twitter

Simon Skoczylas
2.250 Tweets

Profil bearbeiten

Simon Skoczylas
@giftkugel

senior software engineer, lateral thinker, new technology lover, addicted to photography and (web)design

Dortmund, Nordrhein-Westfalen skoczylas.net
Geboren am 17. Januar 1979 Seit Oktober 2009 bei Twitter

656 Folge ich **281** Follower

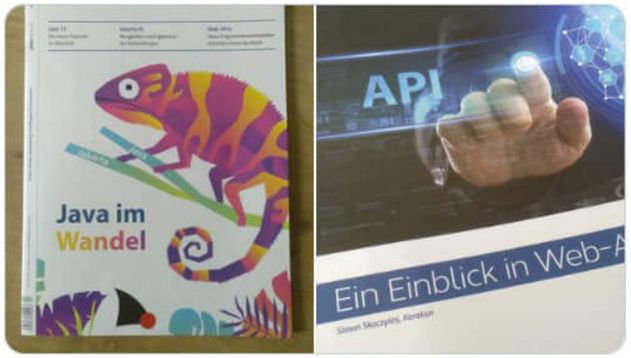
Tweets | Tweets und Antworten | Medien | Gefällt mir

Angehefteter Tweet

Simon Skoczylas @giftkugel · 5. Juli 2019
I'm a bit proud. 🙈🙈
Got the current "Java aktuell 🇩🇪" with my article about WEB-APIs (Browser not REST) 😎

I'm even on the front page. 🤭

#java #webdev #web #webapi #javaaktuell #javascript



5 11 36

untitled - Paint

File Edit View Image Options Help

For Help, click Help Topics on the Help Menu.

ffffff

Strich
000000
Schriftgröße
Klein Mittel Groß Sehr groß
Schriftfamilie
Handgezeichnet Normal Code
Textausrichtung
Links Zentriert Rechts
Deckkraft
Ebenen
Aktionen

100%

Main goal

Close the gap between Web and native applications

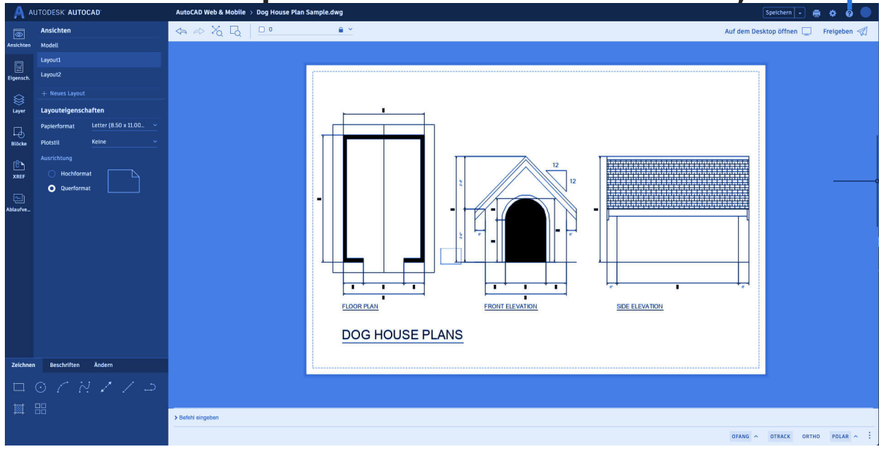
Guess what?

There is still a gap!

Still a gap

Great applications are already available on the Web

For example AutoCAD Web, https://web.autocad.com



But, I cannot interact with DWG files from my machine 😭

Therefore application vendors use tools like Electron

Missing features

Access local files without upload

Use features of the OS

Interface access (Network, Serial)

Socializing (Share, Contacts)

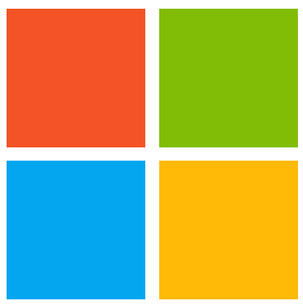
and more ...

Web Capabilities Project

alias Project Fugu

Cross-vendor initiative

Initiated by the Google in ~2018


Fugu Community Sync notes

Chromium Issue Tracker

Name is derived from an asian puffer fish dish called fugu

Delicious if prepared correctly, deadly if not!

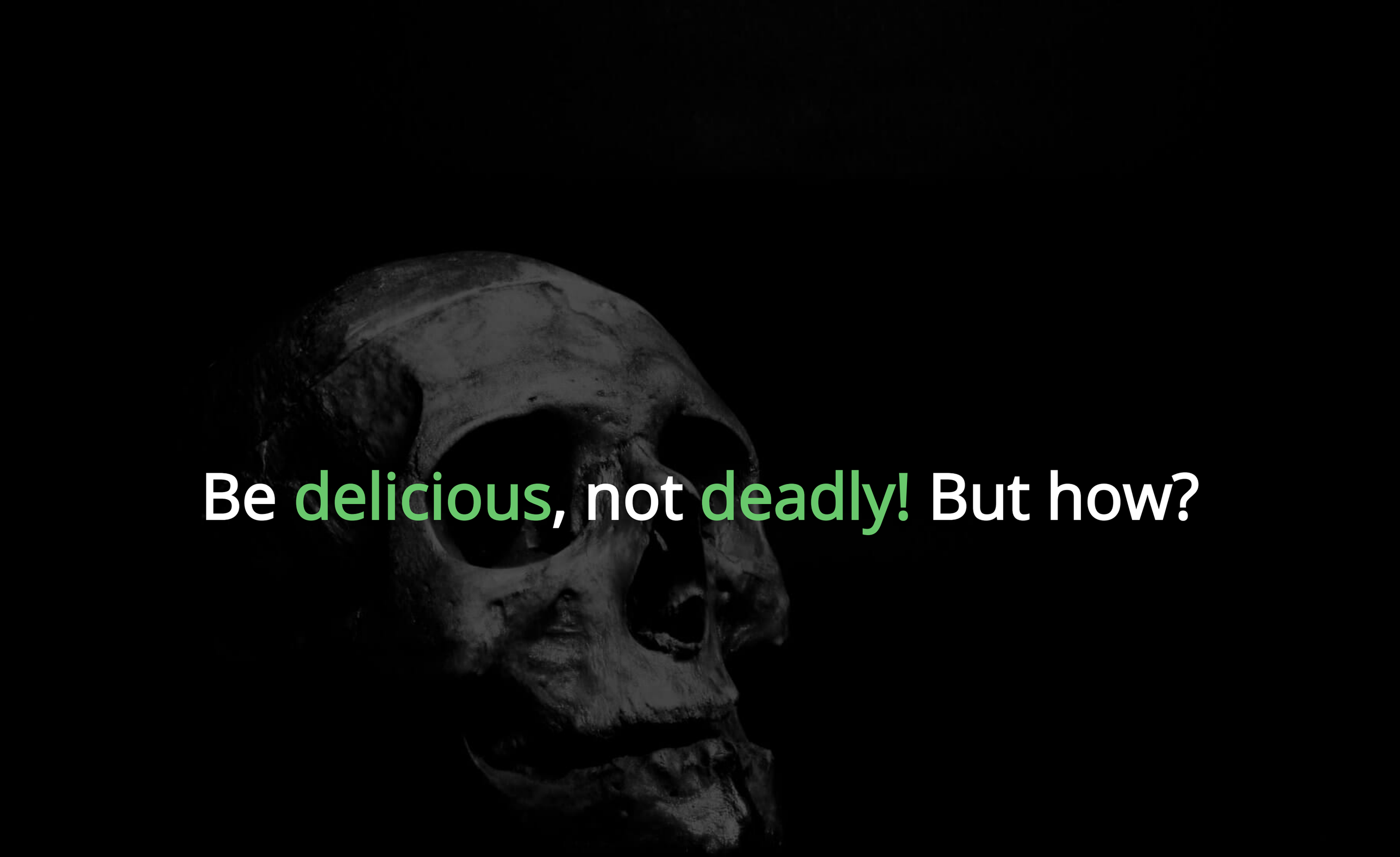
New capabilities

Web apps should be able to do anything iOS/Android/desktop apps can

By exposing the capabilities of these platforms to the web platform
while maintaining user security, privacy, trust, and other core tenets of the web

Nothing should ever be granted access by default, but instead rely on a permission model that puts the user in total control, and is easily revoke-able. It needs to be crystal clear when, and how these APIs are being used

https://web.dev/fugu-status/



Be delicious, not deadly! But how?

Iterative Process

Identify need and use-cases

Write a explainer

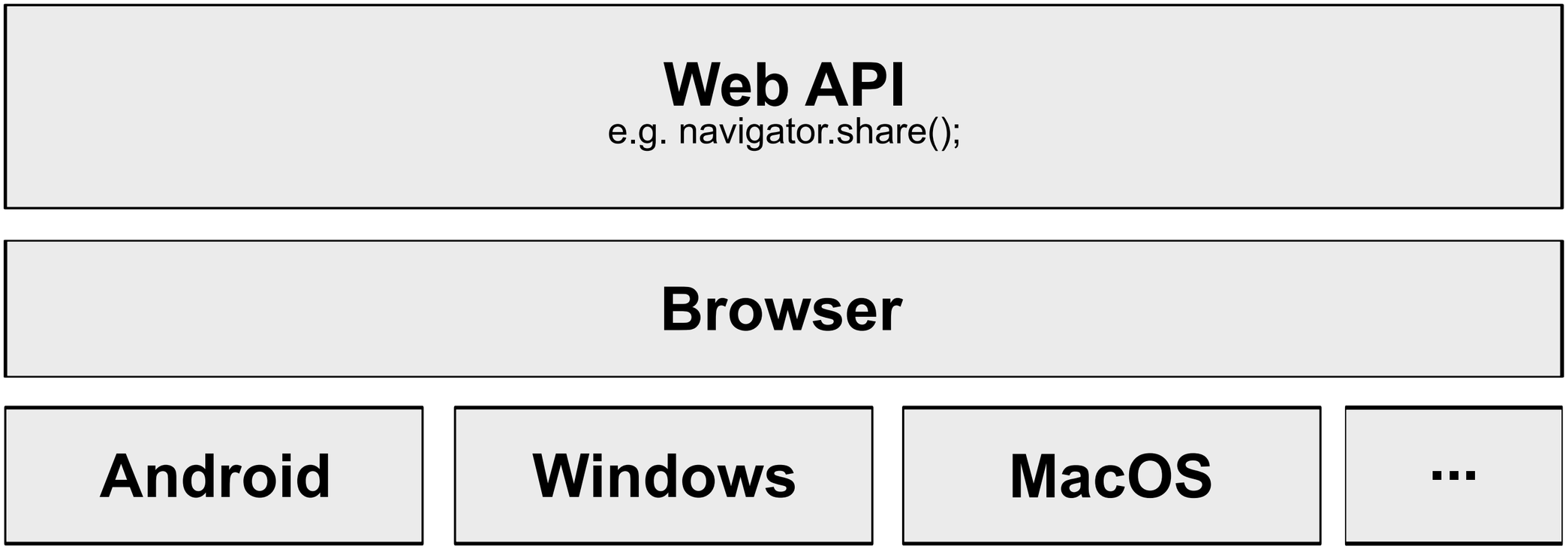
Get feedback

Create a formal specification (first WICG, and then maybe W3C)

Origin trails / Experimental flags
HTTP header, HTML meta tag

Ship it

Abstraction layer



Which APIs will we get?

Bunch of APIs coming to town

Chromium https://goo.gle/fugu-api-tracker

Datei Bearbeiten Ansicht Einfügen Format Daten Tools Add-ons Hilfe

Freigeben

100% Nur Lesezugriff

A1 | Summary

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Summary	Who	Crbug	Stars	Where	2020		2021					
2	*Last updated: 11/17/2020, 9:58:17 AM*					86	87 (stable)	88 (beta)	89 (canary)	90	91	92	93
3	Legend: *italic* = not started; ✓ = shipped					=Behind Flag	bar=Origin Trial	=Ship					
4	Shipped												
5	✓ Web Share Target API	G	668389										
6	✓ Web Share: Support sending files (Level 2 spec)	G	903010										
7	✓ Support programmatical copying of images to clip	G	150835										
8	✓ Web Share Target "V2" for WebAPKs master bug	G	885313										
9	✓ Enable custom enter key on virtual keyboards	G	634330										
10	✓ Expand Storage Quota	G	897276										
11	✓ Contact Picker API	G	860467										
12	✓ getInstalledRelatedApps	G	895854										
13	✓ Compression codecs	G	897298										
14	✓ Implement Periodic Background Sync	G	925297										
15	✓ desktop-pwas: Support "minimal-ui" display mode	G	1007151										
16	✓ App icon notification badge for PWAs	G	719176										
17	✓ Allow the Badging API to be used from a service v	M	937720										
18	✓ Barcode Detection API	G	659138										
19	✓ WakeLock API - Screen Lock	I	257511										
20	✓ Web OTP API	G	670299										
21	✓ Implement the Content Index API	G	973844										
22	✓ *Implement Transferable Streams*	G	894838										
23	✓ App Icon Shortcut menu for PWAs	M	955497										
24	✓ Native File System API	G	853326										
25	✓ Ability to control connected camera pan / tilt / zoo	I	934063										
26	Origin Trial (details)												
27	Web NFC	I	520391										
28	QuicTransport tracking bug	G	1011392										
29	Backpressure for WebSockets (WebSocket Streams	G	840253										
30	Pointer Lock Unadjusted Movement option	G	982379										
31	User Idle Detection API	G	878979										
32	Web Serial API	G	884928										
33	WebHID API	G	890096										

Redirect | Dashboard | Instructions | chromium-issues

API tracker in Google Sheets

https://fugu-tracker.web.app/

Fugu API Tracker

STABLE	BETA	DEV
Chrome 89	Chrome 90	Chrome 91
13 days ago (Mar 2, 2021)	In 29 days (Apr 13, 2021)	In 71 days (May 25, 2021)

Shipped

Web Share Target	M71	+
Web Share API Level 2	M75	+
Async Clipboard: Read and Write Images	M76	+
Web Share Target Level 2	M76	+

Let's take a look

Web Share API

Explainer: https://w3c.github.io/web-share/docs/explainer.html

Web Share is a proposed web API to enable a site to share data (text, URLs, images, etc) to an arbitrary destination of the user's choice

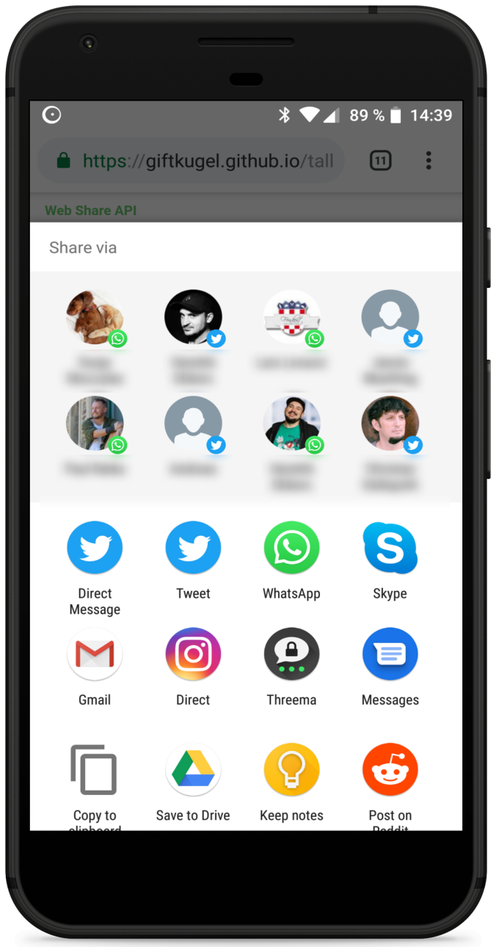
Web Share API Level 2 allows sharing of files

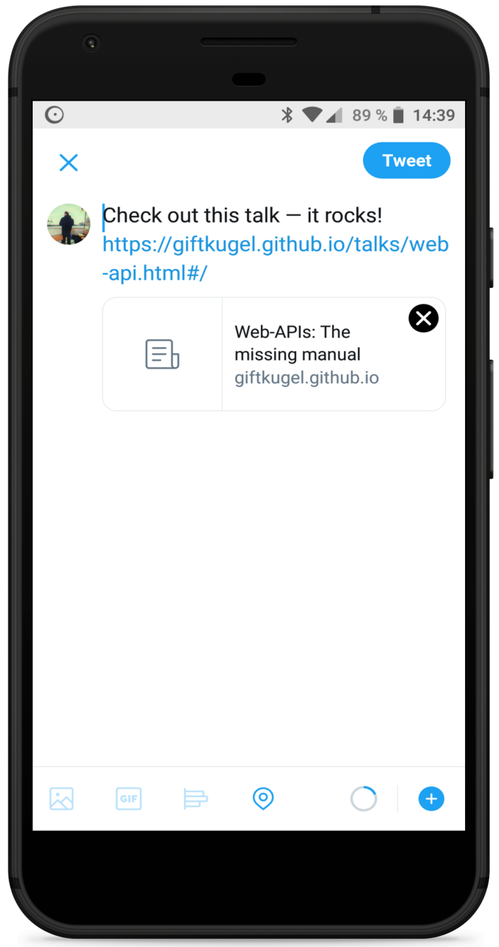
Status: Shipped

Snippet

```
shareButton.addEventListener("click", async () => {
    try {
        await navigator.share({ title: "Example Page", url: "" });
        console.log("Data was shared successfully");
    } catch (err) {
        console.error("Share failed:", err.message);
    }
});
```

Web Share API





Web Share Target API

Explainer: https://w3c.github.io/web-share-target/docs/explainer.html

Web Share Target is a proposed web API to enable a web site to receive shared data from other sites or apps

Status: Shipped

Snippet (Web manifest)

```
{
    "name": "Example Social",
    "short_name": "Example Social",
    "icons": [...],
    "share_target": {
        "action": "share.html",
        "method": "POST"
        "params": {
            "title": "name",
            "text": "description",
            "url": "link"
        }
    }
}
```

App shortcuts

Explainer: https://w3c.github.io/manifest/explainer.html#adding-shortcuts

Numerous operating systems grant native applications the ability to add menu items to the app icon itself.

For web applications, you can define a set of shortcuts to be exposed when the app is installed.

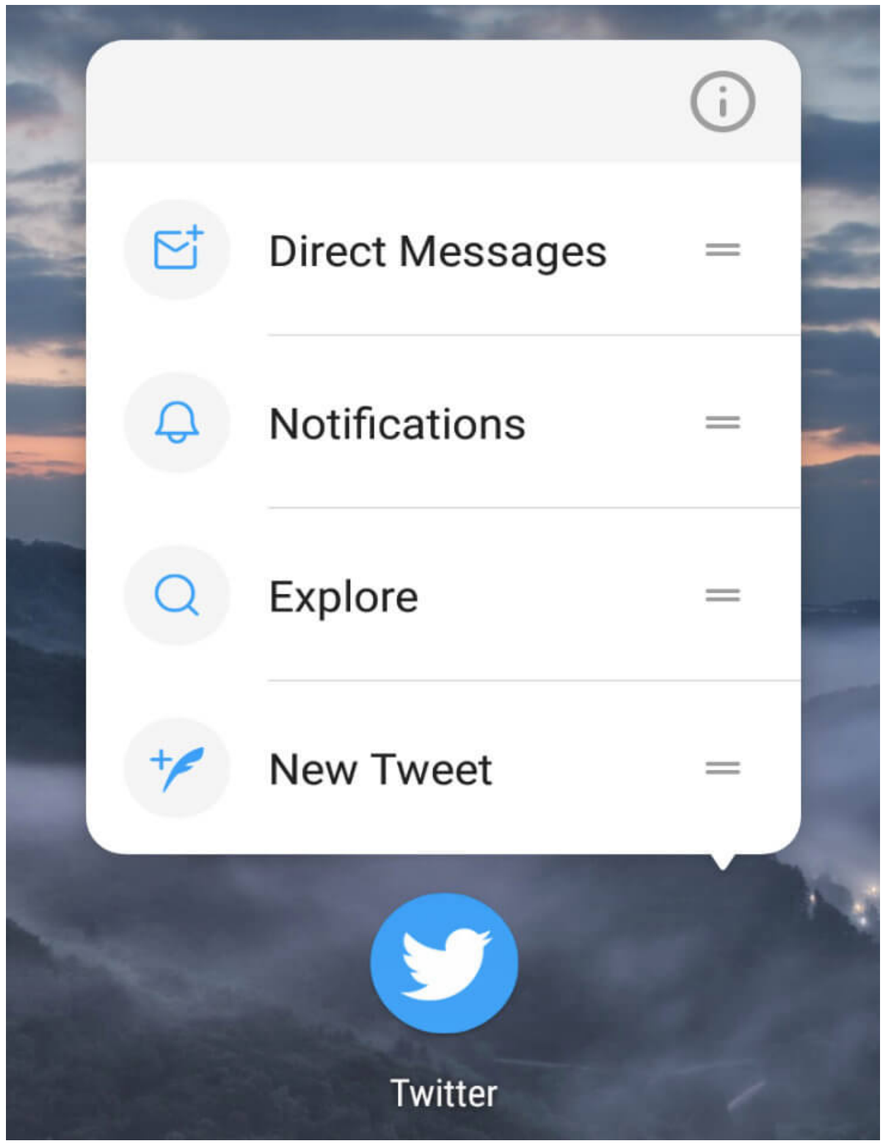
Status: Shipped

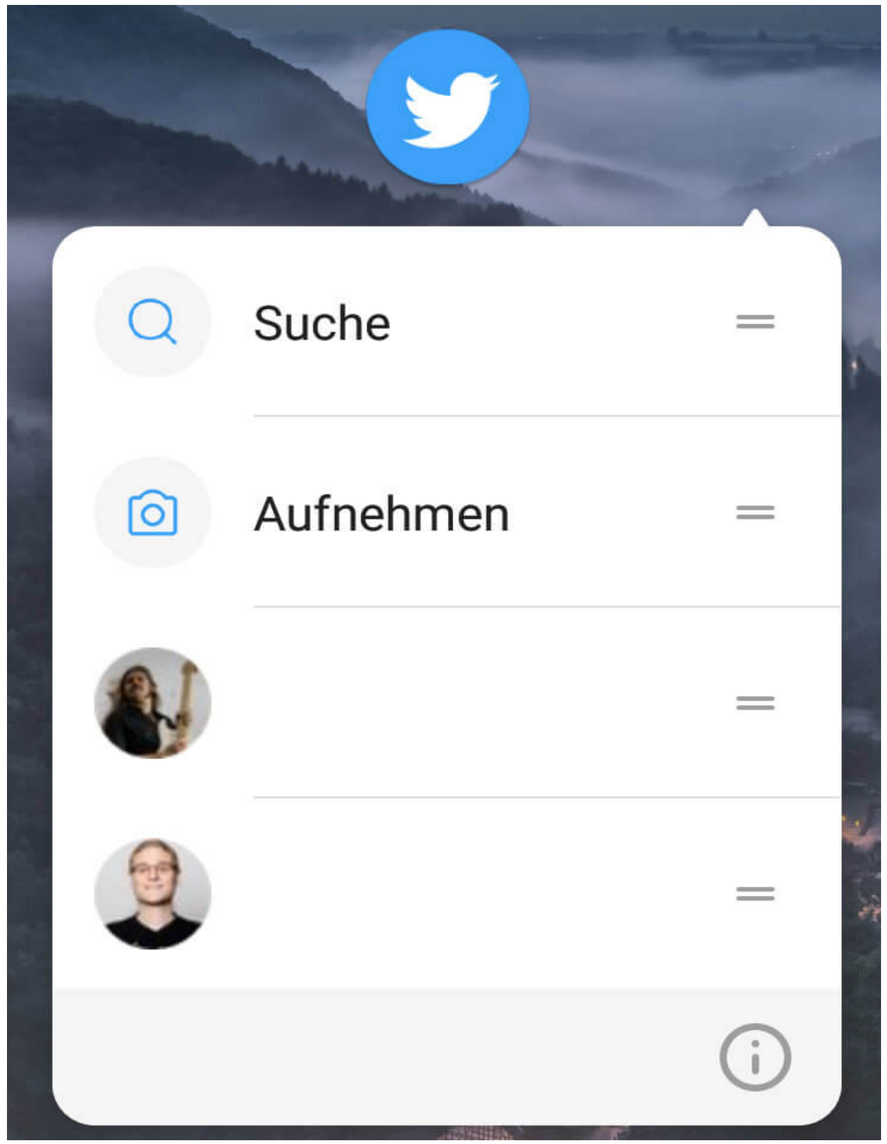
App shortcuts

Snippet (Web manifest)

```
"shortcuts": [
{
    "name": "Play Later",
    "description": "View the list of podcasts you saved for later",
    "url": "/play-later",
    "icons": [
        {
            "src": "/icons/play-later.svg",
            "type": "image/svg+xml",
            "purpose": "any"
        }
    ]
}]
```

App shortcuts





Contact Picker API

Explainer: https://github.com/WICG/contact-api/blob/main/README.md

Contact information is among the most sensitive information on a device, which is why we're proposing a picker rather than a lower level API.

This enables user agents to offer an experience that's consistent with platform expectations, and to highlight the exact information that is to be shared with the website.

Status: Shipped

Contact Picker API

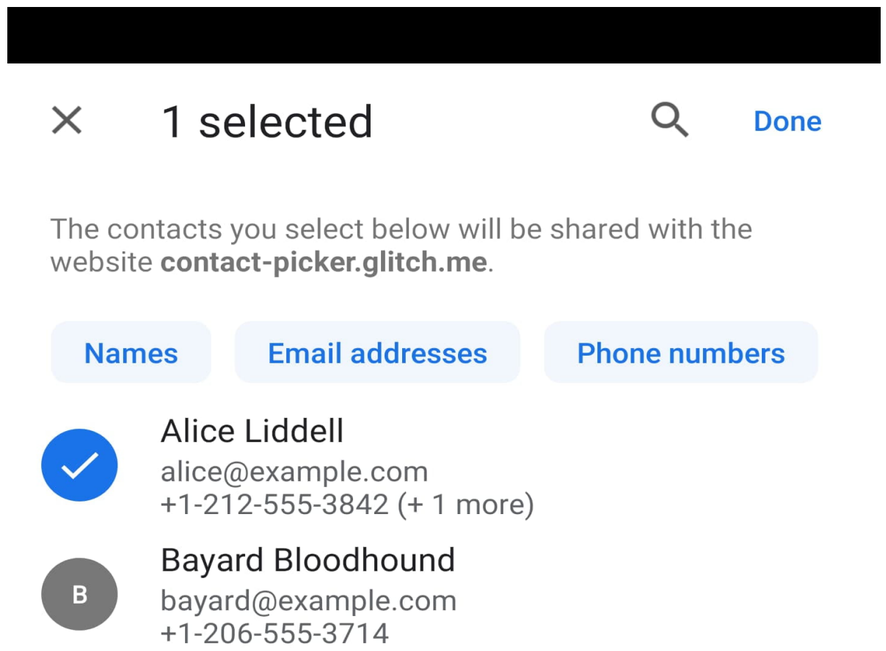
Snippet

```
selectRecipientsButton.addEventListener('click', async () => {
    const contacts = await navigator.contacts.select(
        ['name', 'email', 'address', 'icon', 'tel'], { multiple: true }
    );

    if (!contacts.length) {
        // Either no contacts were selected in the picker, or the picker could
        // not be launched. Exposure of the API implies expected availability.
        return;
    }

    // Use the names and e-mail addresses in |contacts| to populate the
    // recipients field in the website's UI.
    populateRecipients(contacts);
});
```

Contact Picker API



File System Access API

Explainer: https://wicg.github.io/file-system-access/EXPLAINER.html

This API enables developers to build powerful apps that interact with other (non-Web) apps on the user's device via the device's file system.

After a user grants a web app access, this API allows the app to read or save changes directly to files and folders on the user's device.

Status: Shipped

Snippet

```
const openFile = async () => {
    try {
        // Always returns an array.
        const [handle] = await window.showOpenFilePicker();
        return handle.getFile();
    } catch (err) {
        console.error(err.name, err.message);
    }
};
```

Snippet

```
const saveFile = async (blob) => {
    try {
        const handle = await window.showSaveFilePicker({
        types: [{
            accept: {
            // Omitted
            },
        }],
        });
        const writable = await handle.createWritable();
        await writable.write(blob);
        await writable.close();
        return handle;
    } catch (err) {
        console.error(err.name, err.message);
    }
};
```

Idle Detection API

Explainer: https://wicg.github.io/idle-detection/HOWTO.html

This proposed API allows developers to add an event listener for when the user becomes idle (e.g. they don't interact with the keyboard, mouse or touchscreen, when a screensaver activates or when the screen is locked).

Unlike solutions based on monitoring input events this capability extends beyond the site's content area (e.g. when users move to a different window or tab).

Status: Origin Trail

Snippet

```
const idleDetector = new IdleDetector();
idleDetector.addEventListener('change', () => {
    console.log(`Idle change: ${idleDetector.userState}, ${idleDetector.screenState}.`);
});
await idleDetector.start({
    threshold: 60000,
    signal,
});
```

Notification Triggers API

Explainer: https://github.com/beverloo/notification-triggers/blob/master/README.md

Notification Triggers are a mechanism for preparing notifications that are to be shown when certain conditions are met.

The trigger could be time-based, location-based or otherwise—for this explainer, we'll focus on time-based triggers.

Status: Origin Trail

Notification Triggers API

Snippet

```
const createScheduledNotification = async (tag, title, timestamp) => {
    const registration = await navigator.serviceWorker.getRegistration();
    registration.showNotification(title, {
        tag: tag,
        body: 'This notification was scheduled 30 seconds ago',
        showTrigger: new TimestampTrigger(timestamp + 30 * 1000),
    });
};
```

Try it yourself (Shipped features)

https://fugu-journal.web.app/

https://googlechromelabs.github.io/text-editor/

Try it yourself (Origin Trail features)

https://idle-detection.glitch.me/

https://notification-triggers.glitch.me/

Web is moving forward, to close the gap!

Project Fugu provides new capabilities

API Tracker at https://fugu-tracker.web.app/

You can make a difference, add a new request at https://goo.gl/qWhHXU

Or say what you want at https://webwewant.fyi/

Still not perfect 😕 Goolge to fast, Apple not interested?

Thank you for listening 🙌