FDC3 and the legacy of Web Intents
End user as interop layer is terrible UX
TL;DR
A Brief History of Web Intents
History  [ edit ]

Paul Kinlan of Google announced the Web Intents project in December 2010. He soon released a prototype API to GitHub. In August 2011 Google announced that Chrome would support Web Intents. Google and Mozilla have started co-operating to unify Web Intents and Mozilla's Web Activities (which tries to solve the same problem) into one proposal.\[8\][9][10]

In November 2012, Greg Billock of Google announced that experimental support of Web Intents had been removed from Chrome.\[4\]

https://en.wikipedia.org/wiki/Web_Intents
Web Activities

Non-standard
This feature is not on a current W3C standards track, but it is supported on the Firefox OS platform. Although implementations may change in the future and it is not supported widely across browsers, it is suitable for use in code dedicated to Firefox OS apps.

This API is available to installed code running on Firefox for Android, and is intended for use by Firefox for Android Webapps in Firefox for Android 32+.

Web activities define a way for applications to delegate an activity to another (usually user-chosen) application.

Web activities are currently enabled on Firefox OS only, and the full specification is [available on WikiMo](https://developer.mozilla.org/en-US/docs/Archive/B2G_OS/API/Web_Activities)
54% have used their mobile device to send someone a photo or video

- 23% have accessed a social networking site using their phone
- 20% have used their phone to watch a video
- 15% have posted a photo or video online
- 11% have purchased a product using their phone
- 11% have made a charitable donation by text message

The share of Americans that own smartphones is now 77%, up from just 35% in Pew Research Center’s first survey of smartphone ownership conducted in 2011.
```javascript
var loadImage = function(data) {
    url = $.isArray(data) ? data[0] : data;
    if(data.constructor.name == "Blob" || data instanceof Blob) {
        imageBlob = data;
        url = webkitURL.createObjectURL(imageBlob);
    }

    $('img').attr('src', url).css({"width": "auto", "height": "auto"}).addClass('border');
    $('h2').hide();
    $('#step-wrap').show();
    $('#edit, #share, #save').removeClass("disabled").removeAttr('disabled');
}

if(window.intent) {
    loadImage(window.intent.data);
}

$('result').click(function(e) {
    var intent = new Intent( 'http://webintents.org/pick', 'image/*');
    startActivity.call(window.navigator, intent, function(data) {
        loadImage(data);
    });
});

$('h2').click(function(e){
    var intent = new Intent( 'http://webintents.org/pick', 'image/*');
    startActivity.call(window.navigator, intent, function(data) {
        loadImage(data);
    });
});

$('#edit').click(function (e) {
    var intent = new Intent( 'http://webintents.org/edit', 'image/*', imageBlob || url);
});
```

Paul Kinlan & Paul Lewis - September 25th, 2012

https://www.youtube.com/watch?v=8b0LCBx8Rys
EXAMPLE 1

```html
<button id='share'>✍️ Share this!</button>

<!-- ... -->

<script>
    document.querySelector("#share").onclick = function () {
        new Wish("share", "text/plain", location.href).make();
    };
</script>

http://darobin.github.io/web-wish/
```
The Unicorner!!

Share the love!

Check out this awesome meme! http://w3cmemes.tumblr.com/post/34763367921/standards-negotiation-is-the-art-of-sending-people

http://darobin.github.io/web-wish/

[1]: [github.com/Go](https://github.com/Go)

ServiceRegistry is a simple shim to the ServiceRegistry hosted on another domain.

pick will open the picker, that allows the user to choose the service of their choice.

and then that service exposes an API back to the original window.

Nice! Are the UX problems solved by now?

nope. I am just working a service discovery and message passing using raw platform and exposing a custom API over that channel.
Why did Web Intents fail?
Scope
We were overly broad in terms of intents: We started with a lot of custom ones that we thought were cool, and I tried to encourage developers to think of new ones. The lesson I learnt from this is that we should have started with a very small pool with no ability for developers to deviate and lead them through to solving very specific use-cases with the API. In the project we were far far to late in trying to constrain the platform and guide developers in a very opinionated way.

https://paul.kinlan.me/what-happened-to-web-intents/
Politics
In retrospect, I also think we were overly stubborn in our interactions with Mozilla and other browsers, right up until we decided that they had some good points by which time it was too late. We were also working with a some of the wrong people in Mozilla to start with which delayed things in my opinion, but that is not that important now.

Suggestions came in from good people at Mozilla, and for me, my stubbornness blinded us. Ultimately this meant that Mozilla in Firefox OS went and implemented Web Activities that solved a lot of the problems that we thought weren’t problems.

This is my biggest regret.

https://paul.kinlan.me/what-happened-to-web-intents/
One example: Android has a modal full screen (in most cases) stacking app context. Desktop web doesn’t. We had a huge issue with “what happens when you kill off the app that initiated the action” and given that we were pushing people to use “EDIT” intents where all the work is done in the app that fulfills the intent yet have the initiator save the data, if that closed the user was hosed. We stuck with this model, I ignored feedback and ultimately a lot of the feedback we got when the project was killed was “The UX was terrible” and it was the insistence on designing an API that has async task but wasn’t async on the platform we cloned the idea from.

https://paul.kinlan.me/what-happened-to-web-intents/
ONE DOES NOT SIMPLY CREATE INTEROPERABILITY ON THE DESKTOP
FDC$^3$
FDC3

- founded by OpenFin - October, 2017
- Standards Process is public with all deliverables open source Apache 2.0 licensed
- 50+ financial organizations participating
- First Draft of Standards Ratified!
Why things are different this time…
Timing

- Underlying technology is mature
- Strong demand in a specific industry
- Neutral brokers exist
Focus

• Industry specificity
• Use Cases driving technology
• Ecosystem of stakeholders
Technology

- Apps solve trust problems of the open web
- UX solutions not available to browser
- Federated directories replace overly broad discovery
Desktop Agent

App Directory

• Workspace Management
• Entitlements
• UX

App Directory

Resolver UI

Intents

Context Data

App

API

App

API
thank you