



## The app-object economy: We're all remix artists now by Paul Caplan

### Abstract

Maps and messages, notes and news, photography and fitness, streams and shopping: the litany of apps through which we live our mobile data lives and through which state and corporate agents survey our data positions, set in motion a form of everyday remix. This affective cultural practice choreographed between 'us' and 'them' is deeply governmental, establishing subject positions and relationships. Drawing on the work of Timothy Morton, Levi Bryant, Ian Bogost and Graham Harman, this paper argues that this governmental remix mesh is best addressed as a matter of material objects and hyperobjects rather than as an assemblage or network.

The mobile phone and its armoury of apps is increasingly the weapon of choice in semiotic and political struggle. Held aloft at events and wielded at demonstrations the mobile as apparatus has set in motion new scopic, sonic and linguistic regimes. More than a replacement for traditional media devices or ways of seeing, as a read/write networked device it turns every user into a remix artist. 'Photographers' imag(in)e with the camera; 'writers' (un)create micro poems; 'film auteurs' and 'musicians' play and record. But photographic, messaging and sound apps that the remix artist uses are not the only ones in play. The mobile remixer opens other apps to sample content from networks, sharing, creating dialectical images or exercises in mobile *détournement*. The phone as a 'citizen media' remixological device clashes images and words, comments and postings, videos and status updates across the small screen as well as surveillance and sousveillance databases.

Like no other technology before it — even the laptop and tablet — the phone is a mobile remixological machine. Every user is an everyday remix-artist, a VJ, DJ, slam poet. Every moment is a network performance. Every time the phone boots up the performance starts. Every app a cue to remix. Even without a human object connecting and conducting the performance, streams of data and imag(in)ings flow. Words collide in dialectic images. Algorithms on far-away servers generate samples for the remix performance. Ads flash across the screen, Likes alert and Shares notify. Software preferences arrange the samples and protocols render them as streams and interruptions. Written and visual languages are rendered within a flat ontology of data objects in performance. Words, images, data, visualisations, figures, addresses and co-ordinates from corporate, state and algorithmic authors, from friends and Friends dance and draw imag(in)aries on a pocket screen ... all before I've stroked the screen or even started to remix.

And then an / joins in. I type or photograph, I write or I image. I reorder and react to the machinic remix. I add and take away, I multiply and divide the data and the samples already in performance. I arrive not as author or creative but as a data point, a remixed subject, an identity, an avatar in my own and those far-away servers' databases. Tron-like I enter dataspaces, drawn through and across the screen. I am no Subject or Auteur, nor even a remix-Artiste. I am everyday. One among many in Facebook's eyes, in my phone's memory, in my n(N)etwork's ongoing performance. My words join the dance. My images flicker in the stream. My connections and relationships collide in the flow of my own device and others across networks. My subject position is made and remade alongside other human and unhuman actants in the human-machinic performance.

This app-induced performance, this android is the site of memory and subjectivity, the

[Journal Help](#)

USER

Username

Password

Remember me

JOURNAL CONTENT

Search

All

Browse

- [By Issue](#)
- [By Author](#)
- [By Title](#)
- [Other Journals](#)

FONT SIZE



CURRENT ISSUE

ATOM 1.0

RSS 2.0

RSS 1.0

ARTICLE TOOLS

[Abstract](#)

[Print this article](#)

[Indexing metadata](#)

[How to cite item](#)

Email this article  
(Login required)

Email the author  
(Login required)

ABOUT THE AUTHOR

*Paul Caplan*  
<http://www.theinternationale.com>  
GSM London  
United Kingdom

Head of Department,  
Digital Technology &  
Computing

aesthetic and the poetic but also the site of power and governmentality. The performance is generated and generating. Powerful un-human actants generate the samples and the connections and feed off the performance. Likes become datapoints, reTweets become relationships. These points and relationships are fed back into the performance as advert imag(in)ings, as words and images. They generate new samples, feeding the remixological machine, feeding me, creating me within their databases and across my networks. The performance has an active audience. Corporate and State groupies crowd the stage as the I perform and improvise. They shout samples for us to use. They feed the performance. They create the performance. And they watch and listen intensely. They map the remix, data and metadata as part of their own remixological, governmental machine. A relationship engine.

Apps are the raw material of remix. The machine's collection of software objects arranged neatly in grids on the screen arrange the data and data relations in packets of time and space as well as code. My Nexus as nexus. My iPhone as I-phone. The maps app packages geo data, locative experience and located subjectivity. The Facebook app packages governmental Social Graph positions. The game app packages time, score data and scopic and sonic experience. The e-book app packages literacy speed, look-ups and libraries. The packages kept in memory as running apps are remix objects for the I to play with or the governmental assemblage to sample. Our everyday remix practice — imaging in the pub, waiting for a train, coming out of the tube — is playing with Apps, sampling and connecting objects, playing with packets and packages. But the I that plays is also being played with by Apps, being authored as well as authoring that remix.

To take one experience of the app economy via three apps on my phone: I open Google Maps and click search. The app remembers my history and previous searches and guides me step-by-step through its visualisation of space and time to a mapped position in Google's database — six minutes walking, along a line on the screen. When I arrive I launch Layer an augmented reality app that pulls data from Wikipedia about my location, images from Flickr and overlays them on my camera view — a scopic regime creating a scopic experience. I click on Facebook and the post photo option. The app launches the camera. I imag(in)e the space and am returned to the Facebook app to add 'Hello World' and post. The app streams my Wall back onto the screen where my imag(in)ing of my world appears enfolded in the stream of updates and Likes and Friend requests.

This is remix. From App to App across a screen. Sampling server data and mobile sensor data. Text and words. Subject position and data subjectivity created and logged. Friends and friends. Streams and stories. My phone is a remixological network mobile machine. This is everyday remix.

We can look to address that experience, this app economy, app culture and relationship engine through numbers. But like the Facebook app's endless Wall page, new figures emerge just as you think you have reached a conclusion. Apple and Google regularly ratchet up the numbers game — even if their figures are somewhat opaque (Evans, 2013). Apple says 50 billion downloads, Google counters with 48 billion and of course that doesn't even account for HTML5 apps as we shall see. Industry watchers try to map app usage with Flurry Analytics claiming this increased 115 percent in 2013 (Khalef, 2014) and that people are now spending an average of two hours, 42 minutes a day on their phones in 2014, four minutes more than in 2013 (Arthur, 2014). Others crunch the economics of Facebook's US\$19 billion purchase of WhatsApp as clues to the scale of what we are dealing with. In his statement announcing the deal, Zuckerberg (2014) deftly weaves numbers and cultural practice:

WhatsApp is a simple, fast and reliable mobile messaging service that is used by over 450 million people on every major mobile platform. More than one million people sign up for WhatsApp every day and it is on its way to connecting one billion people. More and more people rely on WhatsApp to communicate with all of their contacts every day.

Here Facebook's latest real estate on the mobile screen is positioned in terms of millions of people, billions of connections, and Facebook's particular take on 'communicate', to which we will return.

Researchers have sought another way into that app culture/economy through a focus on the app user. In a large-scale study in 2011, Böhmer, *et al.* (2011) found a fast-paced remix cultural practice with apps being used for less than a minute, even though users spend almost an hour a day using their phones. Bell, *et al.* (2013) focused on app usage across the day and found not only patterns but what they call 'operational chains' — one app experience leading to another. Brown, *et al.* (2013) also looked at actual mobile practices. By videoing and screen-capturing actual app-usage they found that "rather than pushing us away from the world around us — [mobile devices] are instead just another thread in the complex tapestry of everyday interaction" [1]. The contributors to *Moving data: The iPhone and the future of media* (Snickars and Vonderau, 2012) discussed a device but could not escape discussing a practice. As surely as the iPod and then the iPhone are nothing without their App stores, so the content and the Apps cannot be separated from the experience and 'management' of media streams.

Whether we download consultants' PowerPoint slides, watch Zuckerberg's announcement appear as an alert; send academics' PDF papers to Amazon to be streamed back to our phone as its proprietary ePub files or read a *First Monday* paper on apps on our mobile Safari, we see and remix a story of remix.

Mark Amerika describes his 'book' *Remix the book* as a:

cross between an improvised keynote address delivered at a conference on disrupting narratives, a stand-up comedy routine, and the kind of live, pedagogical performance found in experimental seminars and lectures conducted in a practice-based research lab focused on inventing the future forms of avant-garde art and writing. [2]

In form and content he positions remix as our creative response to our contemporary media situation in the tradition of the surrealists, situationists and sample artists. The 'source material everywhere' he talks about, plays with and empowers others to use, is raw material for forms of creative practice and performance. Other discourses around remix begin from 'creativity'. Whether it is Lessig's (2008) deconstruction of a legal framework no longer fit-for-purpose; Navas' (2012) discussion of aesthetics; Ulmer's (2003) 'electracy' or the contributors to *Mashup cultures* (Sonvilla-Weiss, 2010) and *Fibreculture 15*, remix is about creativity because it problematises and remixes the creative.

But everyday remix is not about creativity. It's about finding a destination or a Friend. It's about a sports score alert or a 'I'm running late' SMS; a calendar alarm or cheating in a pub quiz. It is everyday, banal or what poet Kenneth Goldsmith (2011) calls 'uncreative'. Like a radio traffic bulletin or the ads in a newspaper, our remix performances as we play with apps and they play with us compose flat, uncreative but poetic texts and subject positions. It is about what Benjamin (1999) talked of as rags 'n refuse, data fragments that the digital flâneur wanders through and collects.

But of course we are not the only ones engaged in everyday remix.

It is still a matter of debate whether the National Security Agency (NSA) had direct access to Facebook and Google's servers. The NSA PowerPoint presentation suggests it did (*Guardian*, 2013). One slide on the Prism presentation boasts of 'collection directly from the servers of these U.S. service providers: Microsoft, Yahoo, Google, Facebook, PalTalk, AOL, Skype, YouTube, Apple' (Lee, 2013). The companies say it didn't. According to the *New York Times*, there are perhaps secure online 'rooms' where requested information could be sent and accessed by the NSA, but there is no 'back door' (Miller, 2013).

Prism of course is not the only software surveillance and data-mining systems we know about. The *Washington Post* reported on a system wonderfully named 'BLARNEY' complete with a cartoon insignia of a shamrock and a leprechaun hat, which describes itself as "an on-going collection program that leverages IC [intelligence community] and commercial partnerships to gain access and exploit foreign intelligence obtained from global networks" (Gellman and Poitras, 2013). The equally evocatively titled 'Boundless Informant system', which analyses Prism data, reportedly uses free and open source software in order to make its data mining powers 'available to all NSA developers' — a wonderful example of the rhetoric of open platforms coming back to haunt us, or at least watch us (Greenwald and MacAskill, 2013).

There is also debate over exactly what digital information the agencies were accessing: communications or metadata, the content of messages, postings and comments or information about those communications, time and spatial information, where and when the communication happened and between whom. What is clear is that whether sanctioned by the courts or not, whether direct or not, the NSA (and its U.K. equivalent GCHQ) have been 'reading' the digital detritus, the data, data trails and data relationships we leave behind online. This infinite archive of e-mails, messages, images and webcam imaginings as well as Likes and Shares are all objects that NSA's software can build profiles and threat levels from. But in order for that surveillance, sorting and standardisation to work, the objects need to be not only accessible but readable and usable. If they are encrypted or in a non-standard format, the governmental machine slows down. The cryptographers have to get to work and the automated surveillance engine ceases to be as efficient.

Details of the specific software Prism uses have still to come out. Palantir — named after a magical rock in Tolkien's *Lord of The Rings* that granted remote vision — a California startup has denied that its Prism software is the one used (Greenberg, 2013). One could trace the myriad connections between the company, its backers, the CIA, Facebook, Mark Zuckerberg, Sean Parker and its existing government contracts (Hodgkinson, 2008) to raise an eyebrow about that, but what is perhaps more important is to note that data mining, data reading, data sorting software is not only big business but at the heart of big business. Palantir says it sells 'software that allows organizations to make sense of massive amounts of disparate data' (Palantir, 2014) and calls its products 'financial analysis software' (Greenberg, 2013). Big data and the software that see it, whether in a terrorist threat database, a loyalty card database or an RFID tracking database, is at the heart of the government of citizens and consumers. In Foucault's terms it is governmental (Gordon, 1991; Lemke, 2011).

These surveillance practices are the state's everyday remix. Communicative capitalism too is a remix artist. Facebook's Open Graph (Gerlitz and Helmond, 2013) and Google, Amazon and Apple's big data (Mayer-Schönberger and Cukier, 2013; Cote, 2013) remix our app openings, postings and readings into their own predictive remixes, fed back or archived as what I have elsewhere called 'digital detritus' (Caplan, 2013).

How to address this app-fuelled, governmental remix regime in such a way as to account for the panoply of actants at play — the app, the protocols and the standards, the user, the Friend, the Like, the business strategy, the Store, the device. How not to reduce it to 'technology' nor to 'techno-capitalism'. One way is to work with objects.

Triple O, that thought project within philosophy that is continually remixed across the

blogs of Graham Harman, Tim Morton, Levi Bryant and Ian Bogost and his 'speculative realism aggregator' (<http://bogost.com/speculativerealism>), holds that objects are real. Whether material and physical or digital or even ideological or legal — a hard drive, an algorithm, an image, a JPEG-encoded data file, a Creative Commons licence and a media company and its lawyers are all in play and they are all real: all objects at play in my pocket. Here 'real' is an ontological not a mechanist or materialist issue. If they are in my remixological machine, they are real. It doesn't matter whether they have a physical form or not, whether they are imaginary or discursive. They are real because they have a unity and because they act.

Object-oriented ontology (OOO) offers us a way of approaching the weird world of everyday remix and app creation and use that gives objects their due. It refuses to collapse app objects and the nested objects of which they are made into something bigger or reduce them to something smaller — what Graham Harman calls overmining and undermining. Objects, as actants in remix culture (or indeed any other culture) as components of the app economy are worthy of our attention. They are not manifestations of a deeper structural reality or system, superstructural signs of a capitalist, colonialist or patriarchal base. They are not representations, signifying, couriers of a message. They have their own reality and actuality ... all very weird.

If I am simultaneously author and authored. If I am creator and created, one among many actants caught, ghosts in the machine, we cannot think this performance in correlationist terms, human-centric, Subject and objects. Nor can we think it in structuralist terms collapsing the panoply of actants into an assemblage or determining network. Rather there is something flat here. Hardware photo sensors and gyroscopes, software standards and algorithms, machinic and human text, screendraws and photographs, vectors and pixels, corporate and state databases, laws and cameralist practices. All jostle in a vibrant (re)mix of performing object actants. Authoring and being authored. Creating and being created.

OOO holds that actant-objects have withdrawn real dimensions and accessible sensual dimensions. We can never fully know an object. It always has more. It always holds something back. We may be able to map every line of code in an algorithm but we can never get at every dimension of its reality within remix, every facet of its power within and across an app. We may be able to access an image or a sentence on our Wall, to unpack its connotations but there is always more as it spins across the social graph. Heidegger had his broken hammer. In contemporary everyday remix culture, everything is broken. But although an object always withdraws, it also has an accessible dimension, a sensual side that other objects connect with. Our remixological machines (objects themselves of course) and our — or perhaps their — apps are the site of these connections, choreographed by us, by algorithms or by chance.

Triple O refuses to collapse these complex relations, these spatial and temporal dynamics at work in app creation and consumption, in everyday remix upwards into an account of a techno-social system or Like economy or downwards into an account of code. Neither is it willing to privilege human access or the human actant as Subject. In Bryant's (2011) words we are dealing with a 'democracy of objects', opening up what Galloway and Thacker (2007) call the Exploit — the possibility of reconnecting objects in critical and disruptive ways.

Tim Morton uses the term 'mesh' in preference to the more common 'assemblage' in his discussion of ecological thought. Here object-actants are not parts of some bigger whole, nature, or even The Network. Drawing on the image of Indra's net and its jewels reflecting in jewels, He says, '[t]otal interconnectedness isn't holistic ... Indra's net implies that large and small things, near and far things are all "near" [3]. From an OOO perspective the app economy is a mesh, the remixological machine is a mesh, the app itself is a mesh: all at different scales but all object-meshes. Such a perspective allows the sort of technically informed account of apps that Platform Studies applies to video games (Montfort and Bogost, 2009), but also the sort of critical engagement with Apps that work with and through remix.

How does this focus on objects help us address everyday remix and the governmental app? One way to trace this is to use the sort of philosophical practice-research that Bogost (2012) has called 'carpentry' to actually work with objects, to build an app with objects and see what we can learn by a flattened focus on specific actants, to see what the app sees.

Before we engineer/reverse-engineer an app machine-object however, we need to understand two types of app-objects: native and Web apps. Native apps are programmed in high-level languages for specific mobile platforms — ObjectiveC (and now Swift) for the iOS and Java for Android. They are distributed by the various app stores and have to pass Apple and Google's tests as well as pass a share of the profits onto the companies. Web apps are written in HTML5, CSS and Javascript — open Web technologies. They are, in effect bundled Web sites. They are compiled via frameworks such as PhoneGap to be submitted via the App stores but can also be simply installed from the Web. The most famous example of a Web app is the one created by *Financial Times* when it famously withdrew from the Apple Store in 2011 (Dredge, 2013). Debate rages about whether Web apps offer the same affordances and the same performance and power of native apps but from an object-oriented point of view the key issue is the objects in play. The HTML-actant may be different from the ObjectiveC-actant, but the protocols, standards, APIs, servers, tubes (Blum, 2012), human and corporate machine-objects are the same. By exploring how they connect in a simple Hello World Web app, we can begin to map their governmental work.

A Web app is essentially a Web page with what's called a 'cache manifest' file that tells the browser to save the HTML file and any other files for off-line use. These two

objects effectively create a self-contained app. A Web page object (the object that the browser object connects with) consists of a semantically constructed document following the HTML standard. At its most basic it consists of:

```
1.      <!DOCTYPE html />
2.      <html>
3.      <head>
4.      <title>
5.      </title>
6.      </head>
7.      <body>
8.      </body>
9.      </html>
```

To create a simple Web page object that can circulate across the networks, be read in browser-objects on phones, tablets, computers, TVs, fridges etc, we simply add content in between the 'body' tags:

```
1.      <!DOCTYPE html />
2.      <html>
3.      <head>
4.      <title>
5.      </title>
6.      </head>
7.      <body>
8.          Hello world
9.      </body>
10.     </html>
```

If that document is saved as *index.html* and made available on a Web server (object). It can be accessed at a particular URL (object), linked to, shared, remixed or everyday-remixed. That document can be rendered by mobile phone Web browsers but to turn it into a Web app we need to give some more instructions to the browser and operating system software objects. Firstly we create a second object saved as *webapp.appcache* with two lines of code:

```
1.      CACHE MANIFEST
2.      index.html
```

The *webapp.appcache* lists the objects that the browser will download and cache in its memory. We then add an instruction in our Web app to connect to that *web.appcache* object so it can be read off-line, accessed like any other app. We change Line 2 of *index.html* to read:

```
2.      <html manifest="webapp.appcache">
```

Next we need to instruct the browser to render the Web app full screen. A browser object 'reads' — or in OOO terms, connects with the code objects — sequentially, so we add these instruction objects at the top. Firstly our Web app object tells the browser to scale the content to the device size. The new line 3 reads:

```
3.      <meta name="viewport" content="width=device-width, initial-sca
```

Next it instructs the browser object to run the Web app full screen:

```
4.      <meta name="apple-mobile-web-app-capable" content="yes">
```

A final step to make our Web App behave like a native App is have the option to add it as an icon to the user's homescreen. In reality this is just bookmarking a Web (app) page. Here we use Javascript, a text-based scripting language which connects with an interpreter or 'virtual machine' in the browser. Here the Javascript object and the virtual machine objects connect to tell the operating system to pop up a balloon offering the user the chance to save the Web app to the homescreen. In true remix fashion, we do not need to create from scratch. Rather we can remix. A quick search and visit to <http://cubiq.org/add-to-home-screen> and we have two other objects to add to our Web app that add the functionality. *add2home.js* is a Javascript object with the necessary instructions. *add2home.css* is a CSS 'stylesheet' object which connects with the browser telling it how to render the pop up. By adding these two files to the manifest, we ensure they are available off-line:

```
1.      CACHE MANIFEST
2.      index.html
3.      add2home.js
4.      add2home.css
```

The browser object needs to know that it needs to connect with these two objects so we add instructions to the <head> [4]:

```
5. <link rel="stylesheet" href="path/to/add2home.css">
6. <script type="application/javascript" src="path/to/add2home.js
```

Now we have a simple app object. It may be simple but it already consists of nested objects: the device, the screen, the browser, the virtual machine, the operating system, the code as well as the political economy of the app economy, Foxconn workers, proprietorial and open business strategies and standards etc.

Although not strictly necessary, particularly for a simple one-page *Hello world* app, we add some more Javascript and CSS to make the Web app behave like a native app when it runs. Given the right Javascript object to connect with, the virtual engine will tell the browser object to render separate screens of content, add swipe and other haptic affordances etc. Again this can be remixed in from objects on the Web. By adding:

```
7. <link rel="stylesheet" href="http://code.jquery.com/mobile/1.2
8. <script src="http://code.jquery.com/jquery-1.8.2.min.js"></scr
9. <script src="http://code.jquery.com/mobile/1.2.0/jquery.mobile
```

below the other Javascript link we provide the necessary objects. This time however we do not need to remix those objects on our server, we can remix at a distance. JQuery mobile is a set of Javascript and stylesheet objects that instruct a mobile browser to render Web pages as Web apps. By using the URL for the necessary objects in the instructions, we can tell the browser to download the necessary objects from the server. If we look up the IP address of jquery.com (the server object that houses the software objects we need) we find its IP address is 70.32.120.34 which is a server in Culver City, California, latitude 34.0202, longitude -118.393. Now our Web app object is connected to the transatlantic cable objects, server objects, IP address objects, hard drive and very real chemical objects that form the globalisation 'hyperobject' as Morton calls massively weird distributed objects — an issue to which we will return.

A user opening this Web app is part of a global mesh. She encounters the global networks hyperobject as she connects with her device and her network's hardware and software objects, the human and unhuman actants at play in any network encounter. As she closes the Hello world app for something more interesting, she remixes those words into the stream of language she reads across her screen. At the same time JQuery's server logs and possibly her ISP and the NSA's software remixes her Web activity into her profile.

Having built with objects we can now add other objects to enrich her remix (and that of those surveilling her). Firstly, let's remix the Stream into the Web app. Communicative capitalism is more than happy to offer objects to remix. The more objects connected, the more data its Siren Servers (Lanier, 2014) amass. Twitter is happy to package an object for a Web app to connect with in the certain knowledge that the object will report back, connecting with the company's own algorithm objects. On every Twitter user's Settings page there is the option to create a widget, a simple html and Javascript object (ironically both open technologies) from a timeline, from a Twitter list or a search. Sampling all the tweets mentioning NSA is as simple as putting the text into the form and clicking 'Create widget'. Twitter packages the code object:

```
16. <a class="twitter-timeline" href="https://twitter.com/search?q
id="INSERT_WIDGET_ID">Tweets about "NSA"</a>
17. <script>!function(d,s,id){var js,fjs=d.getElementsByTagName(s)
getElementById(id){js=d.createElement(s);js.id=id;js.src=p+"://platform.twitter.com
ets.js";fjs.parentNode.insertBefore(js,fjs)}}(document,"script","twitt
wjs");</script>
```

If we add that between the <body> tags of our web app, the phone's browser-object will connect with twitter.com (in San Francisco, latitude 37.7697, longitude 122.393), and its search software. It will leave a trace on that object (the data-widget id) [5] and platform.twitter.com's servers in Wichita will connect with our Web app to remix the Stream of tweets onto our screen.

Whether the user reads or reads-and-writes into the Twitter Stream through our app, she connects with objects, leaves traces, contributes to the Stream. Her everyday remix connects with communicative capitalism and dataveillance capitalism's everyday remix.

Let us add one more code object to the mix to open up — and remix — other actants within and without the phone. To add a geolocate affordance to the Web app is to connect the user-object with the sensors in the phone, the screen as it redraws a map, the Google server farm and the company's Maps API, the global positioning satellites and locative surveillance and management strategies of the corporate and state actants as well as the user's friend waiting in the Starbucks by the theatre for the user to arrive.

Firstly we connect our Web app to Google's API by adding a script to the end of the <head> section of the code:

```
10.     <script type="text/javascript" src="https://maps.googleapis.com
      key=INSERT_API_KEY&sensor=TRUE"></script>
```

This line instructs the mobile browser to connect to Google's API (application programming interface) software at IP address 74.125.225.48 in Mountain View, California (latitude: 37.4192, longitude: -122.057) and load a Javascript library into the phone's memory. Unless you are a Google Apps for Work customer, you need to connect those objects with an API 'key' you get through your Google account (<https://code.google.com/apis/console>). Google says this helps you and it monitor usage. It also, of course, remixes the data at play in your app into the company's archives as calls on that API are connected to other data objects the Web app provides [6]. The "sensor=TRUE" parameter informs Google that the map will be used with a geolocation sensor. Google requires this as part of its (object) relations with data providers.

The next step is to connect that library object to the phone's sensor, find the location and load that and the Map ready to render it as a visual object within the Web app.

```
11.     <script>
12.     $( document ).on( "pageinit", "#map-page", function() {
13.         var defaultLatLng = new google.maps.LatLng(37.4038194
14.         if ( navigator.geolocation ) {
15.             function success(pos) {
16.                 // Location found, show map with these coordi
17.                 drawMap(new google.maps.LatLng(pos.coords.lat
18.             }
19.             function fail(error) {
20.                 drawMap(defaultLatLng); // Failed to find lo
21.             }
22.             navigator.geolocation.getCurrentPosition(success,
23.         } else {
24.             drawMap(defaultLatLng); // No geolocation suppor
25.         }
26.         function drawMap(latlng) {
27.             var myOptions = {
28.                 zoom: 15,
29.                 center: latlng,
30.                 mapTypeId: google.maps.MapTypeId.ROADMAP
31.             };
32.             var map = new google.maps.Map(document.getElement
33.             var marker = new google.maps.Marker({
34.                 position: latlng,
35.                 map: map,
36.                 title: "Greetings!"
37.             });
38.         }
39.     });</script>
```

Line 22 checks the user's location using the sensor-object and geolocation affordance in the operating system software-object in the phone and loads that information into a software object called "navigator.geolocation" cached for five minutes with a timeout of six seconds. Lines 12-21 use the Google maps API library to create a map based on the user's location or failing that a default location (in our case Google's offices). Lines 26-38 draw that map and an overlay marker (Google's famous pin).

This new relocated data object is then connected with the Web app's page by putting a place marker-object in the <body> of the HTML:

```
48.     <div role="main" class="ui-content" id="map-canvas"></div>
```

And in the <head> as a style:

```
8.     <style>#map-page, #map-canvas {width: 100%;height: 80%;padding
```

The final step is to add an instruction to the browser object to initialise (connect with) the Javascript geolocation object:

```
45.     <body onload="javascript:initGeolocation()">
```

When the user launches the Web app, the objects connect, visualise place, space and subjectivity and render-remix it into the app simultaneously remixing the app and the user back into the dataveillance and communicative capitalism's archives.

Authoring the app-object has been an exercise in remix, cutting and pasting code; sampling code from across the world, remixing affordances, hardware and software actants to build the app-object. Using the app is an exercise in remix for the user as she wittingly and unwittingly samples data, code, words and images into her app experience. The usage of the app is an exercise in remix for communicative capitalism

as it samples and tracks data, our subjectivity and IDs.

It is tempting to locate the objects connecting and reconnecting across networks, cultures and dataveillance strategies as an assemblage. We have already seen how Morton prefers the concept of 'mesh' and the image of Indra's net as a way of mapping those relations. Morton (2013) identifies a special class of objects he calls 'hyperobjects' which offers another way of seeing the app.

This particular class of objects is massively distributed in time and space relative to humans. They are "not just the collections, systems, or assemblages of other objects. They are objects in their own right" [7] and as with all objects they withdraw from full access but are nevertheless fully real. His object of analysis is 'global warming', a term he prefers to 'climate change' [8]. Here the very real, present and material object leaves footprints in our world and experience. We do not encounter the full nature of 'global warming'. Hyperobjects:

occupy a high-dimensional *phase space* that makes them impossible to see as a whole on a regular three-dimensional human-scale basis. We can only see pieces of hyperobjects at a time. [9]

Morton imagines an apple invading a two-dimensional world:

first the stick people would see some dots as the bottom of the apple touched their universe, then a rapid succession of shapes that would appear like an expanding and contracting circular blob, diminishing to a tiny circle, possibly a point, and disappearing [...] That's why you can't see global warming. You would have to occupy some high-dimensional space to see it unfolding explicitly. [10]

I do not want to present the 'hyperobject' or 'high-dimensional phase space' as a metaphor. I want to go further and argue that our practice-research with the Hello World app shows that apps-as-objects sample aspects of a communicative-capitalism-network-media object massively distributed in space and time: a media hyperobject. That hyperobject is real but inaccessible, present and powerful but can only be encountered in a partial way. It is not that dimensions of the hyperobject are somehow hidden from the app (or the app-creator object) — servers and data, cables and IP addresses, patents and protocols are all real and 'there'. Rather it is that the app, the app creator and the app user operate in a different phase space, they only encounter elements of that techno-media-hyperobject as they emerge into their dimension. Communicative capitalism appears as a 'rapid succession of shapes that would appear like an expanding and contracting circular blob, diminishing to a tiny circle, possibly a point, and disappearing'. That's why we can't see the app economy. We would have to occupy some high-dimensional space to see it unfolding explicitly.

Creating an app is the practice of engaging with dimensions of the media hyperobject to create a window into that hyperobject. As we built our Hello World app we sampled higher-dimensional communicative capitalism as streams and data stores, protocols and standards. We built a window where the three-dimensional user (human or unhuman) could 'see' the Apple emerge into our dimensional reality.

Everyday remix is the uncreative practice of playing with media hyperobjects. When a user remixes a tweet and a map, an SMS and a Like, she samples dimensions of the media hyperobject as they emerge into her phase space through her networked remixological machine. As those objects (remember for OOO, objects are nested, the hyperobject is itself a collection of objects) emerge on her phone or on her dataveillance profile, they reveal parts of the hyperobject, they leave footprints on her day and in her NSA file.

Levi Bryant's perspective on objects adds further to our picture of the nested objects we see in play in app-creation and everyday remix. In *Onto-cartography*, Levi Bryant (2014) prefers the word 'machine' to 'object', a term he sees as pushing us back towards a privileging of a non-object Subject. Machines have inputs and outputs. Objects object (as a verb), they act. As such they 'negotiate' with other machines. Drawing on Sartre's discussion of steam engines [11], Bryant presents the machine as active, not the passive object of a creative Subject but a partner — sometimes an anarchic partner — in creation. He says: '[t]he machine itself ends up contributing to the design in a way not intended by the designer' [12].

Here the uncreative everyday remix we see at play in app creation and use appears as objects *objecting*, actively reacting back, weird and slightly out-of-control co-creators, partners with us in the app economy and culture. When we connected the Google API machine to the browser object in our Hello World app, those unhuman machines, those encounters with the hyperobject had a form of agency or vibrancy in Jane Bennett's (2010) terms. And that agency was not under the control of the I-machine that programmed. It is not just that new data was established at every running, the encounter itself, the dimensions of the hyperobject emerging into our app-dimension, was new as inputs and outputs, calls and responses acted on each other outside our control as app-designer and outside each machine's control as data-designer. As Bryant says:

'[t]he *machine itself* issu[es] certain imperatives on its designer that run away from the intentions of the designer. The machine itself ends up contributing to the design in a way not intended by the designer.' [13]

It is this machinic anarchism that opens up the possibility of the sort of Exploit that Galloway and Thacker discuss when they talk of ‘discovering holes in existent technologies and projecting potential change through those holes. Hackers call these holes “exploits” [14].

And of course everyday remix runs away from the I-designer. Its uncreativity rests on objects’ anarchic agency. When we remix our alerts and searches, Kindle annotations and geolocation position, these machines act back on us, on each other as well as on the database and dataveillance machines to which they are connected.

Like Bennett, Bryant is at pains to stress the materiality of these machines and mechanic encounters. As we have seen, as we have built an app and an app experience, the objects and hyperobjects in play are deeply material. The wires across which the connections are made (Blum, 2012), the clouds that are encountered (Cubitt, *et al.*, 2011); the very data that is connected (Cote, 2013) has an ecological and ontological materiality. IP addresses have addresses. Databases have footprints. Remixes leave traces. When the media hyperobject breaks through into our dimension, like its global warming counterpart, it does so materially.

Bryant’s advocacy of a materialist ontology goes beyond a recognition of the material infrastructure of the hyperobject and our app. Taking issue with Harman’s characterisation of materialism as undermining, Bryant holds that materialism does not reduce machines to their constituent parts alone but rather sees them as parts and relations and that the relations, the objecting discussed above, are material.

For Bryant this *objecting* is a flow of energy [15]. Machines connect, they input and output. Their bi-directional relations resist entropy. That flow is physical, material. The traders in dataveillance know information is not just power but also material to be tapped, stored, sifted and sold. Materialist object-oriented philosophers too know that the server on which the map-data sits is material, the screen on which it appears is material but the anarchic connection is material too. The flow is physical. The energy is material. This goes beyond the idea that information is data, to the broader claim that information is the energy that powers the media hyperobject.

When we built our Hello World app from objects, we built it from information-energy too — the seemingly immaterial standards and protocols that guaranteed the connections are physical, not just in their existence in patents or W3C Requests for Comments documents but also in their appearance in our dimension as components of our machine-app. As the HTTP protocol connects client and server it provides the energy to make them work. Whether data is flowing across that connection or not, the HTTP protocol (some may see it as a relation) acts as a material flow keeping input and output connections open, objecting, ready to play. When we engage in everyday remix, the object-samples are connected (however vicariously; Harman, 2007) by physical routers and material cable but also by material flows. The weird JPEG standard that connects the photograph to the Like economy exists. It is an object in our mesh. It is a dimension of the hyperobject whose objecting as a flow of energy is physical. It guarantees the inputs and outputs. It literally powers the app. It is as material as the electricity that keeps the screen on.

As I write this conclusion, my phone sits next to my keyboard. Its clock syncs the time set by the signal towers it connects with. The widget on the lockscreen remixes the data into my workday. A text message, a Twitter alert and a Facebook Like stream across the alerts screen. As I click on the URL my Friend has sent me, I sample a video. As I compose the pithy reply I remix the search I’d just done for Network Time Protocol. Considering my next sentence I mindlessly flick through my photo album and my Read it Later list, a quick couple of Chess moves, a new Google Music track, a paragraph of the Kindle novel and a flight over Google Earth for no particular reason ... the ‘*First Monday* deadline’ Calendar alert set on a server in California pushes me back to work.

It’s not *Girl talk* (Gaylor, 2008) It’s not *Remix the book* (Amerika, 2011). But it is remix: uncreative, everyday sampling, plagiarism. The networked remixological machine we know as the mobile phone is a canvas on which we are all artists. Each app is the molten core, the machine of inputs and outputs through which we sample one dimension of the media hyperobject and mix it with another. Each time we use the phone we engage in a flat ontological remixing in our phase space. We don’t have to be ontologists or media scholars or object-oriented hackers. The senior citizen typing a text message in capital letters because they can’t figure out how to switch them off; the student cheating in a pub quiz; the child closing dad’s e-mail to open Angry Birds are remix artists, object-oriented practice-researchers.

The communicative-capitalism-app-economy-network-media-hyperobject, massively distributed across time and space, massively power full and massively weird is not the context or background. Hyperobjects “are already here” [16]. Everyday remix is our response. 

### About the author

Dr. Paul Caplan is Head of Department, Digital Technology & Computing at GSM London. A former journalist and Internet consultant, he now researches digital objects through practice-research. His work on media and software objects has appeared in the *Atlantic*, *Culture Machine* and *PLATFORM* and is writing a book: *Mobile Imag(in)ing: Object-Oriented Practice Research*, for Punctum Books.  
E-mail: praxis [at] theinternationale [dot] com

## Notes

1. Brown, *et al.*, 2013, p. 1,031.
2. Amerika, 2011, p. xii.
3. Morton, 2010, p. 40.
4. The Javascript and CSS file-objects need to *be* somewhere, literally. when we replace the “path/to/” with the server address where the object is available to connect, we further engage with the materiality of network servers, cables, buildings, firewalls and real walls, planning permissions and legal jurisdictions.
5. The ID is linked to the author of the widget’s Twitter account. Twitter will trace all activity back to that account but because the Web app runs in the phone’s browser, Twitter can still access all the data the user’s phone offers as the browser client object connects with the Twitter server object (Fodden, 2011).
6. There is no need for the API when signed up/into Apps for Business as the App, and any other object, is already connected to the user’s account-object.
7. Morton, 2013, p. 2.
8. Morton, 2013, p. 3.
9. Morton, 2013, p. 70.
10. *Ibid.*
11. Sartre, 2004, p. 159.
12. Bryant, 2014, p. 19.
13. *Ibid.*
14. Galloway and Thacker, 2007, p. 81.
15. Bryant, 2014, p. 11.
16. Morton, 2013, p. 29.

## References

- Mark Amerika, 2011. *Remix The book*. Minneapolis: University of Minnesota Press.
- Charles Arthur, 2014 “Apps more popular than the mobile Web, data shows,” *Guardian* (2 April), at <http://www.theguardian.com/technology/appsblog/2014/apr/02/apps-more-popular-than-the-mobile-web-data-shows>, accessed 11 March 2015.
- Marek Bell, Matthew Chalmers, Lucas Fontaine, Matthew Higgs, Alistair Morrison, John Rooksby, Mattias Rost, and Scott Sherwood, 2013. “Experiences in logging everyday app use,” paper presented at *Digital Economy '13* (4–6 November 2013, Salford, U.K.), at [http://www.johnrooksby.org/papers/DE2013\\_bell\\_logging\\_apps.pdf](http://www.johnrooksby.org/papers/DE2013_bell_logging_apps.pdf), accessed 3 May 2015.
- Walter Benjamin, 1999. *The arcades project*. Translated by Howard Eiland and Kevin McLaughlin. Cambridge, Mass.: Belknap Press.
- Jane Bennett, 2010. *Vibrant matter: A political ecology of things*. Durham, N.C.: Duke University Press.
- Andrew Blum, 2012. *Tubes: A journey to the center of the Internet*. New York: Ecco.
- Ian Bogost, 2012. *Alien phenomenology, or, what it's like to be a thing*. Minneapolis: University of Minnesota Press.
- Vincenzo Cosenza, “World map of social networks,” at <http://vincos.it/world-map-of-social-networks/>, accessed 3 October 2012.
- Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer, 2011. “Falling asleep with Angry Birds, Facebook and Kindle: A large scale study on mobile application usage,” *MobileHCI '11: Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 47–56.  
doi: <http://dx.doi.org/10.1145/2037373.2037383>, accessed 3 May 2015.
- Barry Brown, Moira McGregor, and Eric Laurier, 2013. “iPhone in vivo: Video analysis of mobile device use,” *CHI '13: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1,031–1,040.  
doi: <http://dx.doi.org/10.1145/2470654.2466132>, accessed 3 May 2015.
- Levi Bryant, 2014. *Onto-cartography: An ontology of machines and media*. Edinburgh: Edinburgh University Press.
- Levi Bryant, 2011. *The democracy of objects*. Ann Arbor, Mich.: Open Humanities Press.

Paul Caplan, 2013. "Software tunnels through the tags 'n refuse: Object oriented software studies and platform politics," *Culture Machine*, number 14, at <http://www.culturemachine.net/index.php/cm/article/viewArticle/507>, accessed 3 May 2015.

Mark Cote, 2013. "Data motility: The materiality of big social data," Working Paper, King's College, London, at [https://kclpure.kcl.ac.uk/portal/en/publications/data-motility-the-materiality-of-big-social-data\(abe459dc-e3d1-40df-9ede-3c5783150634\).html](https://kclpure.kcl.ac.uk/portal/en/publications/data-motility-the-materiality-of-big-social-data(abe459dc-e3d1-40df-9ede-3c5783150634).html), accessed 3 May 2015.

Sean Cubitt, Robert Hassan, and Ingrid Volkmer, 2011. "Does cloud computing have a silver lining?" *Media, Culture & Society*, volume 33, number 1, pp. 149–158. doi: <http://dx.doi.org/10.1177/0163443710382974>, accessed 3 May 2015.

Stuart Dredge, 2013. "*Financial Times*: 'There is no drawback to working in HTML5,'" *Guardian* (29 April), at <http://www.theguardian.com/media/appsblog/2013/apr/29/financial-times-html5-no-drawbacks>, accessed 11 March 2015.

Benedict Evans, 2013. "How many apps do Android and iOS users download?" (16 May), at <http://ben-evans.com/benedictevans/2013/5/16/how-many-apps-do-android-and-ios-users-download>, accessed 11 March 2015.

Simon Fodden, 2011. "The anatomy of a tweet: Metadata on Twitter" (17 November), at <http://www.slaw.ca/2011/11/17/the-anatomy-of-a-tweet-metadata-on-twitter>, accessed 11 March 2015.

Alexander R. Galloway and Eugene Thacker, 2007. *The exploit: A theory of networks*. Minneapolis: University of Minnesota Press.

Brett Gaylor (director), 2008. *RiP: A remix manifesto*, at <http://ripremix.com/>, accessed 3 May 2015.

Barton Gellman and Laura Poitras, 2013. "U.S., British intelligence mining data from nine U.S. Internet companies in broad secret program," *Washington Post* (7 June), at [http://www.washingtonpost.com/investigations/us-intelligence-mining-data-from-nine-us-internet-companies-in-broad-secret-program/2013/06/06/3a0c0da8-cebf-11e2-8845-d970ccb04497\\_print.html](http://www.washingtonpost.com/investigations/us-intelligence-mining-data-from-nine-us-internet-companies-in-broad-secret-program/2013/06/06/3a0c0da8-cebf-11e2-8845-d970ccb04497_print.html), accessed 11 March 2015.

Carolyn Gerlitz and Anne Helmond, 2013. "The like economy: Social buttons and the data-intensive Web," *New Media & Society*, volume 15, number 8, pp. 1,348–1,365. doi: <http://dx.doi.org/10.1177/1461444812472322>, accessed 3 May 2015.

Kenneth Goldsmith, 2011. *Uncreative writing: Managing language in the digital age*. New York: Columbia University Press.

Colin Gordon, 1991. "Governmental rationality: An introduction," In: Graham Burchell, Colin Gordon and Peter Miller (editor). *The Foucault effect: Studies in governmentality*. London: Harvester Wheatsheaf, pp. 1–52.

Andy Greenberg, 2013. "Startup Palantir denies its 'Prism' software is the NSA's 'PRISM' surveillance system," *Forbes* (7 June), at <http://www.forbes.com/sites/andygreenberg/2013/06/07/startup-palantir-denies-its-prism-software-is-the-nas-prism-surveillance-system/>, accessed 11 March 2015.

Glen Greenwald and Ewen MacAskill, 2013. "Boundless informant: The NSA's secret tool to track global surveillance data," *Guardian* (11 June), at <http://www.theguardian.com/world/2013/jun/08/nsa-boundless-informant-global-datamining>, accessed 11 March 2015.

*Guardian*, 2013. "NSA Prism program slides" (1 November), at <http://www.theguardian.com/world/interactive/2013/nov/01/prism-slides-nsa-document>, accessed 11 March 2015.

Graham Harman, 2007. "On vicarious causation," In: Robin Mackay (editor). *Collapse. Volume II: Speculative realism*. pp. 187–221, and at <http://www.faculty.virginia.edu/theorygroup/docs/harman=vicarious-causation.pdf>, accessed 3 May 2015.

Tom Hodgkinson, 2008. "With friends like these ... ," *Guardian* (14 January), at <http://www.theguardian.com/technology/2008/jan/14/facebook>, accessed 11 March 2015.

Simon Khalef, 2014. "Mobile use grows 115% in 2013, propelled by messaging apps," *Flurry* (13 January), at <http://blog.flurry.com/bid/103601/Mobile-Use-Grows-115-in-2013-Propelled-by-Messaging-Apps>, accessed 11 March 2015.

Jaron Lanier, 2014. *Who owns the future?* New York: Simon & Schuster.

Timothy B. Lee, 2013. "Here's everything we know about PRISM to date," *Washington Post* (12 June), at <http://www.washingtonpost.com/blogs/wonkblog/wp/2013/06/12/heres-everything-we-know-about-prism-to-date>, accessed 11 March 2015.

Thomas Lemke, 2011. *Foucault, governmentality, and critique*. Boulder, Colo.: Paradigm.

Lawrence Lessig, 2008. *Remix: Making art and commerce thrive in the hybrid economy*. New York: Penguin Press.

Viktor Mayer-Schönberger and Kenneth Cukier, 2013. *Big data: A revolution that will transform how we live, work, and think*. Boston: Houghton Mifflin Harcourt.

Claire Cain Miller, 2013. "Tech companies concede to surveillance program," *New York Times* (7 June), at <http://www.nytimes.com/2013/06/08/technology/tech-companies-bristling-concede-to-government-surveillance-efforts.html>, accessed 11 March 2015.

Nick Montfort and Ian Bogost, 2009. *Racing The beam: The Atari video computer system*. Cambridge, Mass.: MIT Press.

Timothy Morton, 2013. *Hyperobjects: Philosophy and ecology after the end of the world*. Minneapolis: University of Minnesota Press.

Timothy Morton, 2010. *The ecological thought*. Cambridge, Mass.: Harvard University Press.

Eduardo Navas, 2012. *Remix theory: The aesthetics of sampling*. New York: Springer.

Palantir, 2014. "Our work — What we do," at <http://www.palantir.com/what-we-do>, accessed 11 March 2015.

Jean-Paul Sartre, 2004. *Critique of dialectical reason. Volume 1: Theory of practical ensembles*. Translated by Alan Sheridan-Smith. London: Verso.

Pelle Snickars and Patrick Vonderau (editors), 2012. *Moving data: The iPhone and the future of media*. New York: Columbia University Press.

Stefan Sonvilla-Weiss (editor), 2010. *Mashup cultures*. Wien: Springer.

Gregory Ulmer, 2003. *Internet invention: From literacy to electracy*. New York: Longman.

Mark Zuckerberg, 2014. "I'm excited to announce that we've agreed to," at <http://www.telegraph.co.uk/finance/newsbysector/mediatechnologyandtelecoms/digital-media/10650340/Facebook-buys-WhatsApp-Mark-Zuckerberg-explains-why.html>, accessed 11 March 2015.

---

## Editorial history

Received 11 March 2015; accepted 4 April 2015.

---



"The app-object economy: We're all remix artists now" by Paul Caplan is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

The app-object economy: We're all remix artists now  
by Paul Caplan.

*First Monday*, Volume 20, Number 5 - 4 May 2015

<http://firstmonday.org/ojs/index.php/fm/article/view/5899/4487>

doi: <http://dx.doi.org/10.5210/fm.v20i5.5899>



A Great Cities Initiative of the University of Illinois at Chicago [University Library](https://www.library.uchicago.edu/).

© *First Monday*, 1995-2015.