

Preliminares a la **comunicación Android con Irka-HydraNet**

- N

1

Créditos

Abstract.

Lo directo, cuando el desarrollador Android quiere acceder a la red Elgg es usar **oAuth** [1] y **Account** [2]. Una vez que el cliente puede manejar los **tokens que el servidor reparte** [2.1] puede **convertir sus clases Java nativas a JSON** [3] y enviar **peticiones** síncronas [4.1] dentro del **Adaptador de sincronización** [5] o asíncronas [4.2] en cualquier punto de la aplicación. El **plugin webservices** [7] junto con algún otro plugin que extienda métodos responderán a esas peticiones enviando **arrays de objetos** JSON [8].

- [1] Write your own Android Authenticator: <http://blog.udinic.com/2013/04/24/write-your-own-android-authenticator/>
- [2] Android.Account dev pages: <https://developer.android.com/reference/android/accounts/Account.html>
- [2.1] Never your asked? <http://stackoverflow.com/questions/7030694/why-do-access-tokens-expire>
- [2.2] Enhanced plugin ApiAdmi: <https://elgg.org/plugins/1105480>
- [3] A Java serialization/deserialization library that can convert Java Objects into JSON and back. <https://github.com/google/gson>
- [4.1] Go salmon: <http://stackoverflow.com/questions/16904741/can-i-do-a-synchronous-request-with-volley>
- [4.2] Asynchronous HTTP Requests in Android Using Volley <http://arnab.ch/blog/2013/08/asynchronous-http-requests-in-android-using-volley/>
- [5] Write your own Android Sync Adapter: <http://blog.udinic.com/2013/07/24/write-your-own-android-sync-adapter/>
- [7] Docs on Webservices plugin: <http://learn.elgg.org/es/stable/guides/web-services.html>
- [8] Ver lista de métodos que expone un sitio elgg visitando su url: /services/api/rest/json/?method=system.api.list

Forward 1.

De acuerdo, en este punto, la **conexión de Adaptadores** [1] que **carguen cursores a listas**, quizás alguna con *swang* a borbotones [2] que tenga opciones en botonera cargadas al vuelo completa una posible vía de interacción desde Android.

- [1] **Crear ListView con cursor:** <http://www.sgoliver.net/foro/viewtopic.php?t=233>
[2] **A swipe menu for ListView:** <https://github.com/baoyongzhang/SwipeMenuListView>

Forward 2.

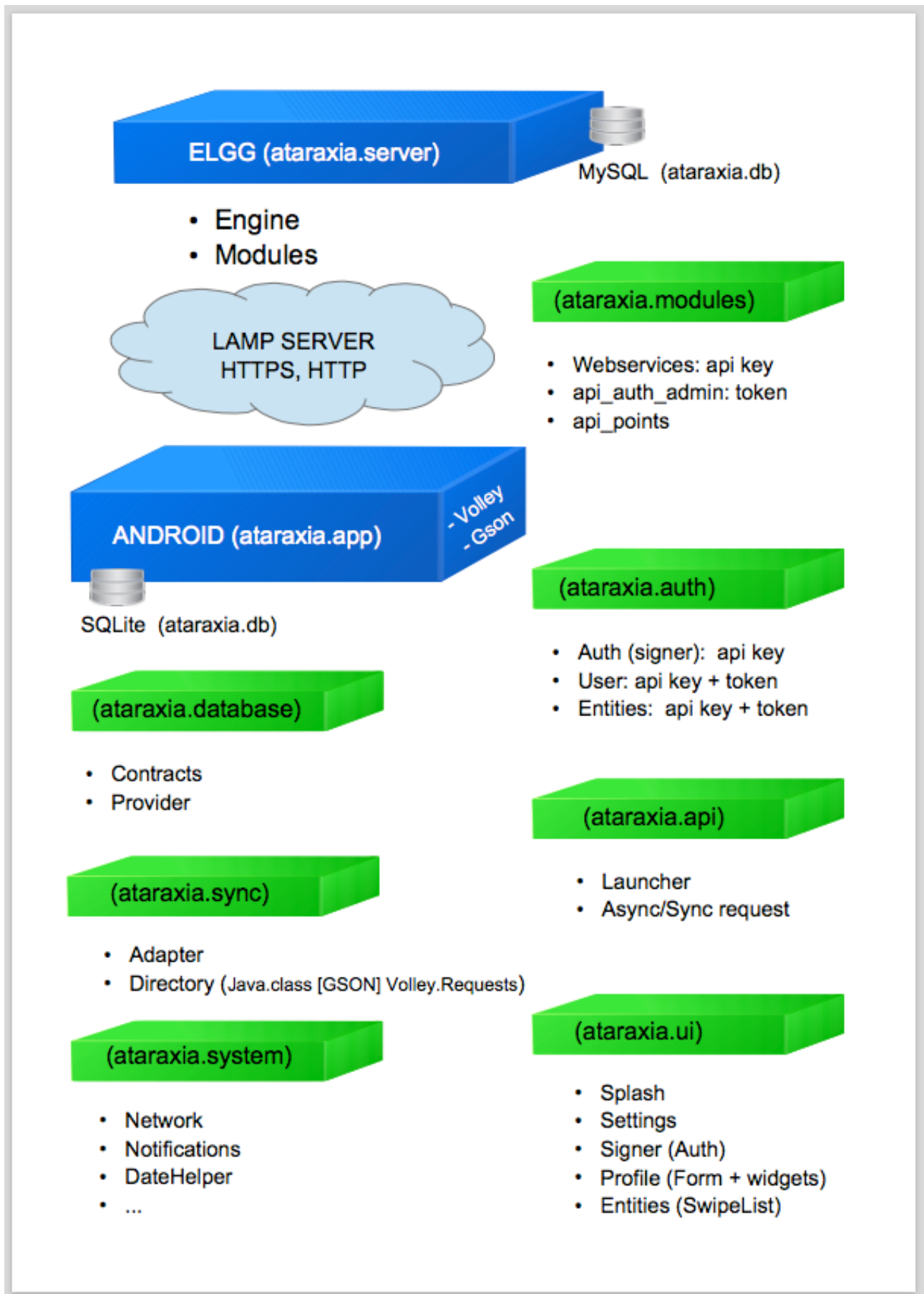
Pero, aunque lo anterior tiene todo el jugo y juego: el título de este micro ensayo sugiere algo, digamos, más allá.

Ensayando, se sabe que no se habla libre de error sino liberando el error del hablar, tomando notas y tal ([...](#)), sin querer interrumpir en el buen progreso de lo no ensayado sino planeado, etc.

Decimos: **MicroEnsayo Más Elgg-UX-browsers** y, porque todas y cada una de las palabras del título han sido escogidas atendiendo a criterios de peso específico semántico, tenemos gran parte del microensayo relatado.

(continuará) http://aleph1888.github.io/ataraxia_archive/

Schema



(contexto, antecedente)

MicroEnsayo Más Elgg-UX-browsers

Creada por [aleph](#) hace una hora Categorías: [Filosofía](#) [expansion](#)

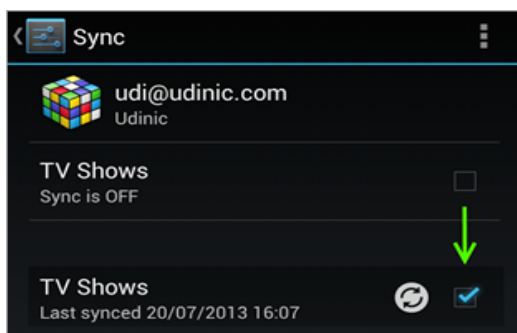
Escritura automática en el autobús, aprovechando un trayecto de dos horas, desde el lugar en que construyo un boceto sobre lo expuesto hasta la casa de la persona que más q...

Entonces, sigamos donde lo dejamos, **¿dónde lo habíamos dejado?**

Ahora el coder Android ha logrado poder jugar con el botoncito de sincronización [1], lo cual le pega tirones al servidor, performando la sincronización base[2]:

[1] **Programatically click the sync button:** <http://stackoverflow.com/questions/4465765/how-to-code-a-sync-now-operation-on-android>

[2] **Shit! This is happening:** <http://stackoverflow.com/questions/34685069/android-syncadapter-onperformsync-not-called>



Por dónde seguimos? Accediendo por la UI

Miramos ahora por el otro lado, en lugar de mirar la parte de atrás, en la carga de datos entre un servicio alejado de la capa de usuario en el Android y un servicio REST remoto en un GLAMP, ahora miramos por la parte de adelante...

... la idea es hacer un pequeño puente entre el usuario *logueado* y una visita normal con el navegador. Es decir, usamos un **WebView** [1] que tenga un **WebClient** y un **WebChromeClient** [2] para cargar una cierta api de urls que se forman, como todo *elgg-coder* saber a partir de tres claves: "*owner, friends, all*"; y otros tantos verbos casi REST: "*add,...*" para manejar las distintas entidades [3].

[1] **Docs on WebView Class:** <https://developer.android.com/guide/webapps/webview.html>

[2] **Did you ask your self...** <http://stackoverflow.com/questions/2835556/whats-the-difference-between-setwebviewclient-vs-setwebchromeclient>

[3] **Docs on Entities Class** http://reference.elgg.org/entities_8.php.html

Un conciso: logear el navegador.

Gracias al comando `loadUrl` [1] podemos lanzar peticiones a nuestro `WebView`.

Ahí podemos poner las url's de elgg [2] o lanzar comandos javascript [3].

[1] **how to load a url to webview in android?** <http://stackoverflow.com/questions/11288611/how-to-load-a-url-to-webview-in-android>

[2] **Docs on Encaminamiento** <http://learn.elgg.org/es/stable/guides/routing.html>

[3] **Best code examples for WebView loadUrl method (android.webkit.WebView.loadUrl)**
<http://www.codota.com/android/methods/android.webkit.WebView/loadUrl>

Entonces, parece lógico que, como todo *elgg-coder* sabe, si haces una petición a Elgg antes de estar validado serás redireccionado a la pantalla de autorización. Por tanto, **si usamos el evento `onPageFinished` [1] para comprobar si la url cargada se corresponde con la esperada**, la primera vez no corresponderá, porque habremos aterrizado en el formulario de autorización, **podremos enviar los datos al formulario vía `loadUrl` y presionar el botón de envío** teniendo el `WebView` en un estado de *visibility invisible* (que como todo *android-coder* sabe tiene al control en pantalla pero no visible, diferente de *gone* que lo extrae) hasta que cargue la validación y, de nuevo, en *onPageFinished* esta vez, tengamos la url que buscábamos y ya podemos pasar a *visible*.

[1] **Have you ever been in such...** <http://stackoverflow.com/questions/6719814/onpagefinished-never-called-webview>

Lo de arriba es *cool* pero ortodoxo usar good staff para estandarizar el proceso [1].

[1] **How to do browser level authorization instead of UI one...** <http://stackoverflow.com/questions/2585055/using-webview-sethttpauthusernamepassword>

One step beyond

Después, ¿miramos de agregar una interfaz [2] en cada lado [3] para fluir mejor?

[2] **Diseño Android: Interfaces web con WebView** <https://danielme.com/2013/02/14/disenio-android-interfaces-web-con-webview/>

[3] **Docs on Javascript moduling AMD on Elgg** <http://learn.elgg.org/es/stable/design/amd.html>

(Continuará) http://aleph1888.github.io/ataraxia_archive/

Contexto y antecedente: <https://irka.io/tasks/view/969/advance-file-takeout-addon>

This addon is important because is opening an interesting door to the possibility to have each, network and unhabitant, a same scheme of data between them.

That means that at some point, network will can on a possible future to don't no provide such data, but provide meta-links between that data. This empower unhabitants and takes out responsibility about to the networks.

Rsync..vpns.. peering... so many new nice ideas can be implemented by advacning/developing this tool.

Las interfaces de usuario, si están renderizadas en el cliente, ofrecen mayor y mejor resolución y experiencia de usuario.

La molestia de descargar o instalar la interfaz en el dispositivo cliente se compensa con una rápida carga de la estructura visualizadora, en espera, eso sigue igual, que el cliente sincronice por la red con el servidor y descargue los datos.

El concepto de entrar en la red que sirve Elgg, si es una **semilla como Lorea** entonces un *setup* se disemina en la web; y, nacidas, la federación entre nodos crea la red: publicas en una flor y aparece en miles.

Si es [Irka](#), entonces, el paradigma pasa por vertebrar **una serie de operaciones entre pares** que van desde la autenticación a la coordinación en el manejo de los datos.

La idea es que:

<https://quitter.se/notice/6008148>

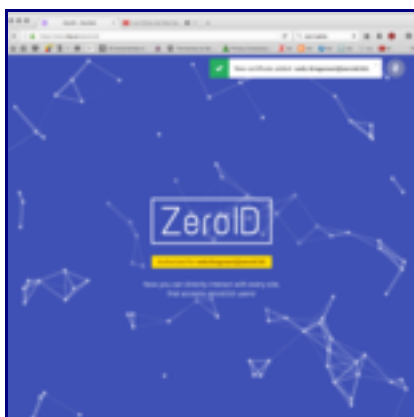
[crawler @crawler 17 jun](#)

Netzpolitik.org:: Ask me anything: Netzneutralität mit SaveTheInternet.eu –

<https://gs.morphtown.de/url/30487>

Entonces, el principio alquímico: *"como es arriba, es abajo"*, se antoja un tuétano para los huesos de este esqueleto donde **Irka es una base o centro o núcleo** para distintos clientes, desde distintas redes, desde distintos canales mueven sus datos personales.

- **Soporte para una ...**



-

La idea de OAuth como **par de llaves pública/privada se repite arquetípica** en el archivo "[content.json](#)" de Zeronet: un anillo para gobernarlos a todos. Cuando creas una app en tu nodo origen, generas las claves, estás listo para empezar a consumir **tokens temporales** que te autorizan, usando esa vía, par de claves, etc.

Gnu App sample 2



- Gnu App sample



Entonces, un ensayo de aquellos primeros que hacían los enciclopedistas franceses post-revolución, ¿Pangloss para qué te quiero?, quedaría fuera del objetivo en este. Más se pretende ensayar una equidistancia entre el proceso de plantear la arquitectura del sistema y el hecho de construirlo línea a línea, instalación a instalación.

En los viernes de **otoño 2016** estaremos en el hacklab virtual acebuchesLab construyendo un engarce que opere algo parecido a como Dapp Ethereum hace en un pseudo lenguaje para escribir operaciones de intercambio entre pares, regladas por el conjunto.

Sé que lo críptico del proyecto no embarra lo arriba expuesto. Bastante concreto y comprometido, por otra parte. Vviernes a viernes, en el #fridayChurch se puede abundar con musiquita y buena programación extrema o ágil iteración scrum.

Fíjate, persona que lees este ensayo, la paja que presenta esta idea concisa de construcción, y que ahora, fuera lo contingente, se enuncia clara y en corto para que no quede inadvertida. Decimos, por comparación entre distintos ámbitos, ese *meso_lenguaje* que permite escribir Contratos en el paradigma *blockchain*, homeomorfía mediante, enunció: "**Analogía en el modelo de 'record' del Magic Mem Set y el el concepto 'entity' de Elgg**". Lo que constituye un estupendo puente para transportar datos, con su semántica, entre sistemas relacionales como la base de datos de Elgg o árbol, o whatever. Enunciar la idea es simple, en las sesiones de los viernes habrá de hilvanarse.

¿No te parece Bum? Una vez que los clientes abundan (como Webextensions Firefox o Chrome, peopleApp o servicio en Android o iOS, etc; desde entornos criptográficos emanados de Satoshi, TOR, etc...) [cuando Irka tenga que llevar datos a esos clientes](#), que puede tener en su base de datos Mysql o puede estar como Mean o P2P semi o distribuido, respectivamente, el volcado en el MPTT abstracto permitiría un protocolo estandar porque se ofrece moldeable al programador de las APIs para representar la *Elgg categories taxonomy*.

Dos ideogramas de ambos conceptos: *records* y *entities*.

<https://github.com/aleph1888/magicmemset/blob/master/General/models.py#L15>

```
class Tree(models.Model):
    name = models.CharField(verbose_name=_(u"Nom"), max_length=30,
        help_text=_(u"El nom del Arbre"))
    clas = models.CharField(blank=True, verbose_name=_(u"Clase"), max_length=30,
        help_text=_(u"Model de django per fer l'arbre (cal que tingui 'left' i 'right')"))
    TYPES = (
        ('A','Art'),
        ('B','Being'),
        ('W','Artwork'),
        ('S','Space'),
        ('C','Concept')
    )
    clas_type = models.CharField( choices = TYPES, blank=True,
        verbose_name=_(u"Tipus bàsic"),
```

<https://elgg.org/discussion/view/2468913/getting-entities-in-web-services-content>



By Ismayil Khayredinov [143 days ago](#)

API authentication allows you query the site as if you were a logged out user. You need to authenticate a user that has access to the entity. Users have ACCESS_PUBLIC, that's why you can see them.

P.D.; Los sábados, tras la [barbacoa](#), trataremos temas: *Keep an eye onto:*

<https://bitmask.net/es/features/cryptography#storage-soledad> and

<https://forum.ethereum.org/discussion/1088/relational-vs-distributed-database>

- ▣

-

-

[aleph](#) hace 3 días

[psy](#) hace 7 horas

- [Preguntas](#)
- [aleph](#)
- ¿Qué huesos tendría un esqueleto tal que funcionara como puente entre Android y AMD Elgg?

1 Vistas

- Público
-
-

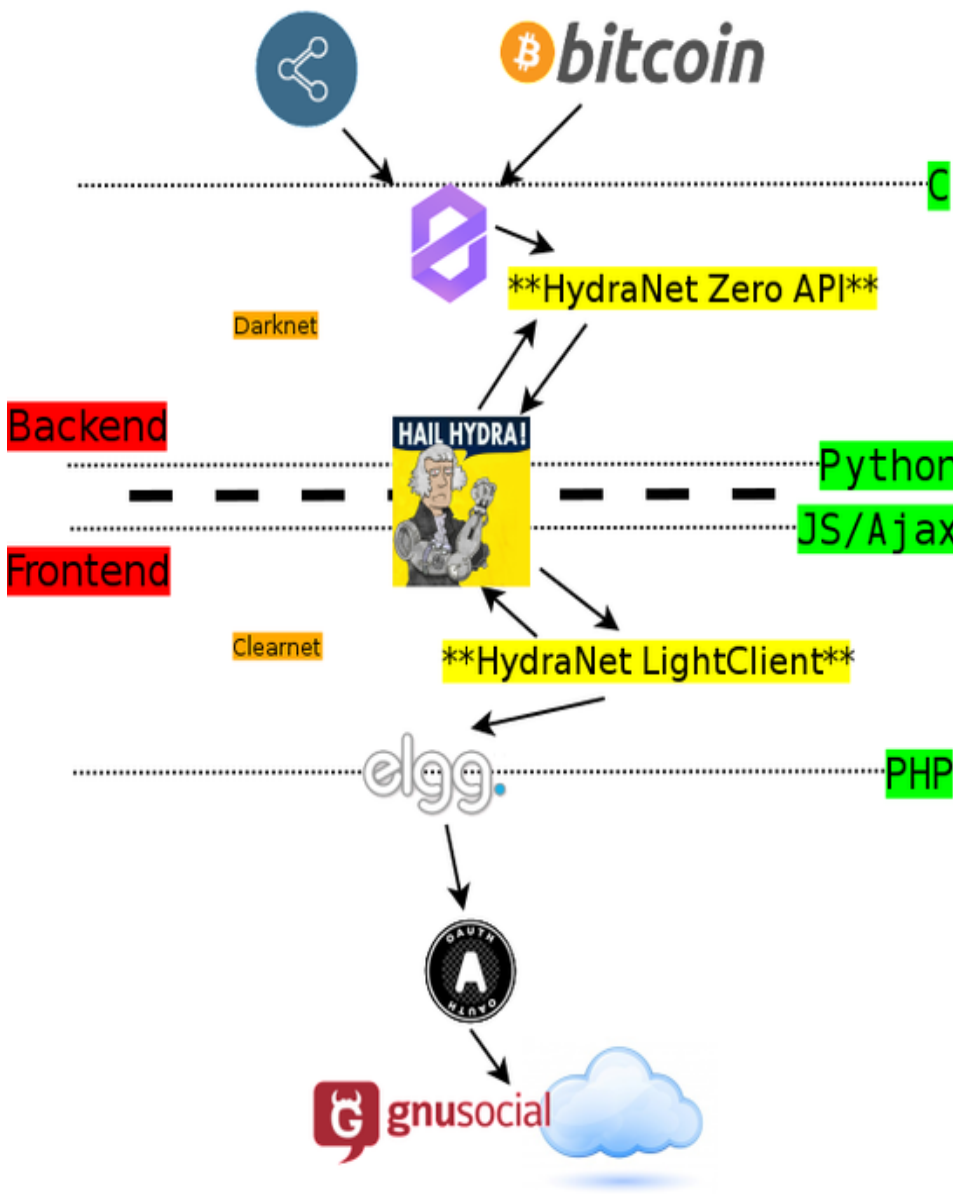
Hecha por [aleph](#) ahora

- [Contracciones](#)
-

Es decir, igual como se parsea el javascript de Elgg hacia sus Vistas, replicando hacia arriba, hacia el contenedor de la navegación, la interfaz del visor.

Se buscan ontologías al respecto. Gracias.

P.D.: Contexto - antecedente: <https://irka.io/blog/view/1318/ultra-elgg-un-ensayo-sobre-la-pluralidad>



[aleph](#) hace 2 días

1) WorldWideDatabase

(2012) Google reveals Spanner, the database tech that can span the planet
<http://zd.net/OWrS1f>

2) Crypto on chains

Peter Todd's tree chains idea and strategies based on advanced cryptography such as SCIP/zk-SNARK, but there is still a lot of research that remains to be done

3) Graph DB

<http://titan.thinkaurelius.com/>

Titan is a scalable [graph database](#) optimized for storing and querying graphs containing hundreds of billions of vertices and edges distributed across a multi-machine cluster. Titan is a transactional database that can support [thousands of concurrent users](#) executing [complex graph traversals](#) in real time.

[aleph](#) hace 2 días

Kademlia

Kademlia es un protocolo de la capa de aplicación diseñado para redes [P2P](#) descentralizadas. Especifica la estructura de la red, regula la comunicación entre nodos y el intercambio de información. Los nodos se comunican entre sí usando el protocolo sin conexión [UDP](#) (ver [modelo OSI](#)). Con Kademlia se crea una nueva red virtual sobre una red [LAN/WAN](#) existente, como [Internet](#), en la cual cada nodo de la red es identificado por un número (ID del Nodo).

Como Kademlia es un protocolo que no requiere servidores, desaparece el cuello de botella que se creaba con la necesidad de estos. En vez de conectarse a un servidor, se conecta a un cliente (con una dirección IP y puerto conocidos), que sea compatible con el protocolo Kademlia. Esto se llama [Bootstrapping](#).

Cuando se busca, cada cliente actúa como un pequeño servidor y se le da la responsabilidad de ciertas palabras clave o fuentes. Esto añade complejidad al encontrar fuentes, ya que no existe un servidor central al que preguntar, pero a cambio se propagará la consulta a través de la red.

•

[aleph](#) hace 2 días

Extraído de: <https://irka.io/discussion/view/1235/the-big-plan>

Individuals are identified by hashes (linked to ips). A system of permission added to json files provided by trackers will create a layer of roles, directly connected to Elgg entities.

▪

[aleph](#) hace 2 días

Extraído de: <https://irka.io/pages/view/1276/zernet-as-cms>

Reading the ZeroFrame API [1] we can see the private keys always have to be provided to the backend, which precludes client side users, even though they would be possible given the architecture.

Thus, our goal can be to allow such client side users, the main tool for this can be a (possibly) small zernet modification, adding a ZeroFrame API function to sign sites on the client side. More generally, all functions have to be doable from js using the node as a proxy to the zernet. The javascript side would then become a ****ZeroNet LightClient****

[1] https://zeronet.readthedocs.io/en/latest/site_development/zeroframe_api_reference/

- ▫
-
-

[aleph](#) hace 2 días

Forward previous comment:

*The javascript side would then become a ****ZeroNet LightClient*****

I [@psy] am re-searching for tools made on that way ([dynamic content](#), with remote AJAX requests, etc..) and I have found (and tried practically) this one: <https://github.com/xaxes/zernet-relay-chat>

Resources are js made and by using websockets: <https://github.com/xaxes/zernet-relay-chat/tree/master/src/js>

- ▫
-
-

[aleph](#) hace 2 días

Zite

A Zite is a [ZeroNet website](#). Zites have the notable difference of being entirely [decentralized](#) and work even if the visitor is offline. Zites are designed to be entirely [open source](#) and freely shared. Zite owners can also make their site [cloneable](#). This ensures content is not censored and cannot be forcefully taken down.

However, due to how the site is distributed and viewed, sites must be entirely contained within local [HTML](#) and [JavaScript](#) files. Information for the site can be stored in an [SQLite](#) database or in [JSON](#) files.

Zites require the private key generated with the address in order to be edited or published. Users of

the zite have a subdirectory they may edit within the zite's file structure.

Various attempts have been made to index and search the hundreds of zites that have been created. [Search engines](#) like [Kaffiene](#) or [ZeroSearch](#) allow searching through hand-crafted indexes using key terms. While other zites, like [Olist](#) or [ZeroCentral](#), allow user submissions to generate a list.