

# edraj.io The Data Managment Platform

- ▼ Conception, v 0.2
- ▼ Kefah Issa, Amman, Augst - 2016

# The problem

- ▼ Users are overwhelmed with information of all sorts
  - ▼ Personal content: Photos, Videos, Documents, files ...etc.
  - ▼ Curated content: Collected content; actually the majority of data users are involved with is the curated type.
- ▼ The information is heavily fragmented and duplicated
  - ▼ Many many apps and websites with stupid limitations: twitter 140 chars, facebook no rich text formatting, instagram and snapchat are plain d\*#b :) ...etc.
  - ▼ Each doing their best to lock-in users and their content.
  - ▼ Duplicated identities
- ▼ Not easy to move data around or make proper backups. E.g. I can't push articles I write on my blog to FB and vice versa. Those should be considered as distribution channels.
- ▼ No proper meta data and semantics; hence making it difficult to find / navigate through.
- ▼ Older content is buried deeply and is hard to find and surface.
- ▼ Its difficult if not impossible to create any context-ware reporting.

# Founding Principles

- ▼ Future-proof data format (promot to beocme 1<sup>st</sup> class citizen): The authored data must become application-agnostic. The content should be associated with its own schema definition and meta data manifest. Schema defintion can make use of [Avro](#). This should facilitate polymorphic interaction with the the data and the seperation of data-serving performance from at-rest authored reference (golden). We can use indexing engines like [Elastic.io](#) to index the data for performance.
- ▼ No SPOF or provider lock-in: All data is federated / decentralized / distributed. Leveraging [DHT](#) for both content and identities. (DHTs are used in bittorrent, bitcoin and many others). It should be easy for any one to setup a [POD](#) and join the band. All code should be accessible freely under [GPL3](#) or similar.
- ▼ Security: Proper Identity, authentication, encryption and access control. [PKI-based](#). Content can be signed by the author ([PGP](#)). [Oauth2](#).
- ▼ Content (Data entities) can have proper [OO](#) structures and [Graph-oriented](#) relationships. Content can have blob attachments. It should be trivial to pull an excell sheet or a list for reporting, or even provide a real-time dashboard.
- ▼ Enable light-weight mini-apps like [data-handlers](#) (persistance, rendering, ...etc), [miners](#) (local and external), [bots](#) and [chat-bots](#) to do all sorts of tasks.
- ▼ Implement adaptors/facades to integrate with existing/legacy systems. Email: [SMTP/IMAP/Pop3](#), [XMPP](#), [SIP](#), [RTC](#), [WebDav](#), [API's](#): Facebook, twitter, Wordpress ...etc.
- ▼ Leverage existing [free and open source](#) technologies and build on top.
- ▼ Messaging (events exchange) is done in [real-time](#).
- ▼ This solution is mainly around establishing the standard and presenting one reference implementation.

# Applications of such a technology

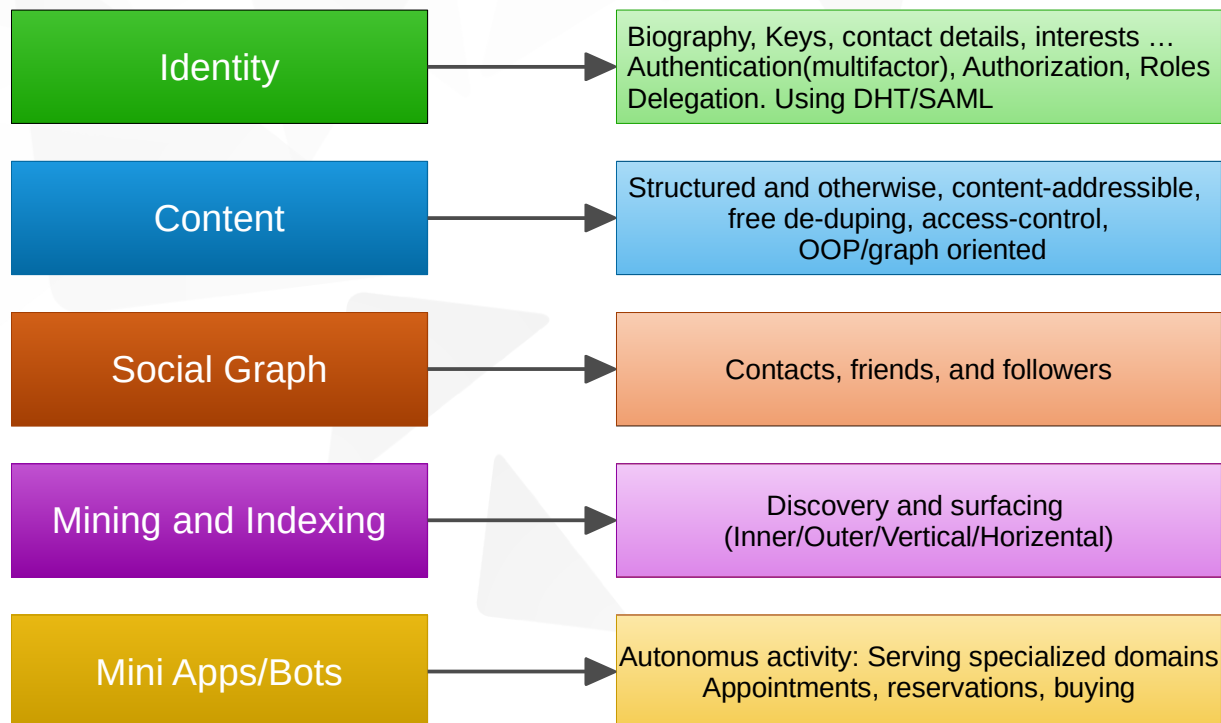
- ▼ Email and messaging offering SMTP/IMAP/Pop3 API.
- ▼ File and document management (access control, organization, backup) offering: Web, Webdav
- ▼ Content publishing (Public sharing) is a breeze: CMS like: Blog, Wiki, Shopping store ...
- ▼ Inventory, HR and school management systems
- ▼ Ticketing / problem tracking system
- ▼ MSAccess-like/Spreadsheet-like where you shape the data in a modular fashion.
- ▼ You name it.

# Commercialization through value-added services

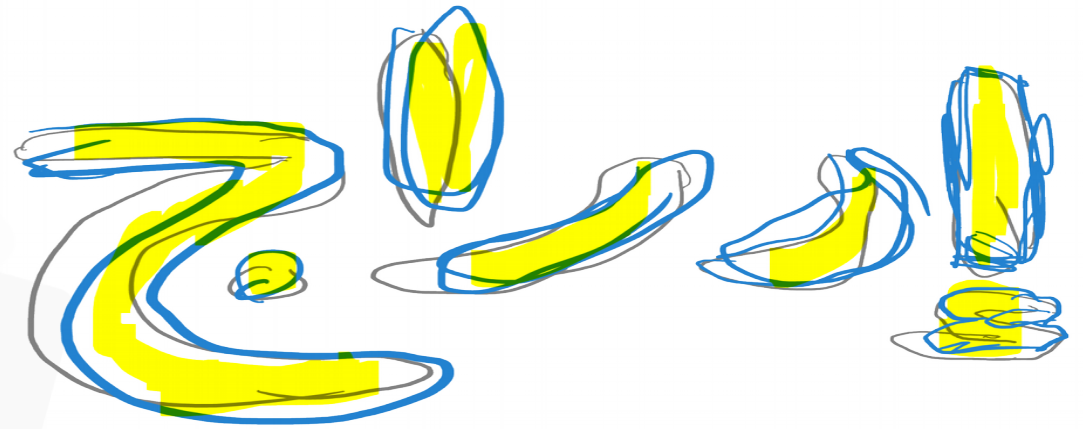
- ▼ Digital safe: storing sensitive documents and information.
- ▼ Improved identity management with guards against lost/theft
- ▼ Improved indexing/crawling/mining
- ▼ Innovative-apps: Bots and chat-bots
- ▼ Financial mediation for paid services and goods
- ▼ Provisioning of integrated persistence back-ends.
- ▼ 3rd party authentication / authorization
- ▼ Public publishing your self and/or cause/business (aka hosting)
- ▼ Nice and easy id reference: `your_id@domain-name` (email like)
- ▼ Service reliability

# Core Capabilities




Capabilities offer distinct functions and integrate operationally. Intensive research need to be done in order to determine the useful technologies that can be used.



- Community-run, distributed social network: \*diaspora
- Peristence backends: Google drive, onedrive, s3, dropbox, ipfs ...etc.
- File and Data management/distribution: camlistore/git-annext/ipfs/owncloud,syncthing, sparkleshare
- Peer2peer communication and messaging: tox,bitmessage
- Identity, Autherization and Authentication: Oauth 2.0, PKI, PGP
- Content-addressable data and Identities have their own DHT.
- Web Ontology Language: OWL



Thank you ...

- ▼  <https://github.com/kefahi/io>
- ▼  <http://fb.com/kefahi>
- ▼  Kefah.issa@gmail.com