Vortex

Additional implementation from V 1.0 to V 1.5

Commands

Name	Action	Example
Calculator	Basic calculator (+ - / * ^)	@calc 3+2*3
Translate	Translate languages, the first language can be auto detected or specified.	@translate I love you IN French @Translate Je t'aime IN Indonesian FROM French
Conversion	Converts currencies, units, timezones.	@EUR 50 in USD @GALLONS 1 in LITERS @GMT now IN Bangkok
Stocks	Gives stocks info	@STOCK [index]
Distance	Gives distance informations between places, users and tags.	@Distance to @Radit @Distance to Paris @Distance to #movie @Distance from @Radit to Paris
Hot topics	Shows hot items	@hot @hot images ...
Cold topics	Same as cold	@cold "
Tags/Place	Shows tags on the map and in the timeline, tags can be chosen.	@in Jakarta @in Jakarta #video
Tags near me	Same as Tags/Place but around the user	@here @here #video
Add management		@Add @Radit @Delete @Radit @Mute @Radit #jokes @Add #jokes @Mute #jokes
Add tag/user		@Add @Radit #jokes
Group management	Create, delete groups, add users to groups	@Creategroup MyGroup @Deletegroup MyGroup @Renamegroup MyGroup MyGroup2 @Add @Radit TO MyGroup
Nest management	Same as group	@Createnest ...

Name	Action	Example
Cross post	Cross post to other sites. If user has some site by default, can use operators - and +	@t @fb @y
Checkin	Checks in using 4square with optional status	@checkin @checkin I jus haz cheezburger
Mention	Usual mention	@Radit
DM	Usual direct message	~Radit
Show	Shows a user's timeline, groups, nests, own timeline or cloud.	?Radit ?MyGroup ?MyNest ?Me ?Cloud @jno #video

@Ad

@Add
Follows a user
_____

@Ade
@Adrian
@Arrgh

@T

@Twitter
Posts to all twitter accounts
_____

@Twitter: Radit
@Twitter: Jno
Posts to the selected account
_____

@Tammy
@Trevor

User groups and privacy

Users shall be able to define groups or other users.

Privacy settings will be group based. While setting the privacy for a file or a jot, a user can decide to show to, hide from, and consult the full list of who can and can't see.

Show to: @Radit X, MyGroup X

Hide from: Everyone X

@Radit and the members of **MyGroup** can see your jot
See full list

Show to: Everyone X

Hide from: @Radit X, MyGroup X

Everyone except @Radit and the members of **MyGroup** can see your jot
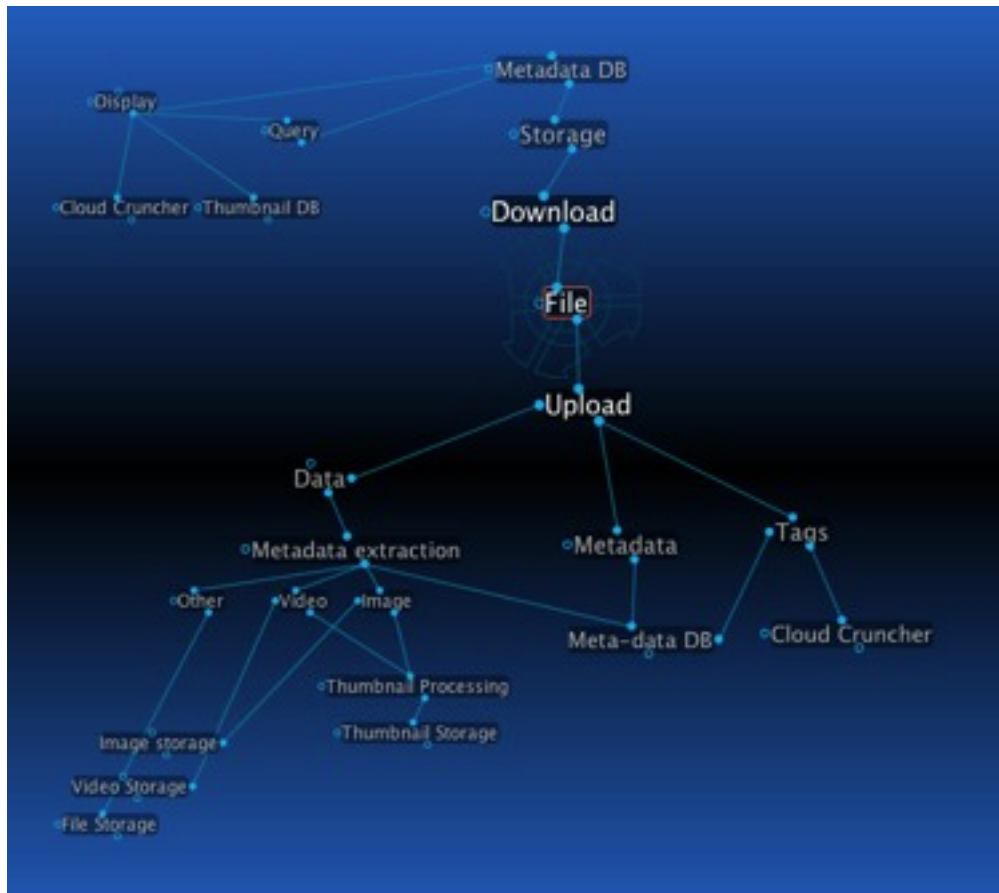See full list

Show to: MyGroup X

Hide from: @Radit X

All members of **MyGroup** except **@Radit** can see your jot
See full list

Can see:

- @Michael
- @Danny
- @Jimmy
- @MrOrange

File upload

Users can upload their own files, the type is automatically detected.

- Thumbnail generator for movies and pictures
- PDF can be opened in the browser
- Generic icon for music
- Generic icon for other file types

Meta data

File type	Metadata
Images	EXIFF data, width, height, weight, thumbnail
Movies	Length, weight, thumbnail
Sound	Length, AD2/3 tags, weight, preview
Other files	Weight

Infrastructure

- Metadata processing
- Thumbnail processing (separate for video/pictures/sound)
- Metadata DB
- Thumbnail DB
- Separate storage for Images/Video/Others

Media Library

Every user uploaded file shall be tagged with the same tags as the note they are embedded in.
One the file uploaded, It will appear on the 'media library' page, where the user can set its privacy level.

If a file is made public to all or some, the allowed users will be able to access it through search and through the poster's public media library.

User settings:

- Set uploaded files default privacy

Media library (self)

- Delete
- Rename
- Change tags/places
- Change privacy
- Filter by tags/places
- Filter by type

Media library (other)

- Jot the file (relay)
- Download
- Filter by tags/places
- Filter by type

Statistics

Over time / Total
General / Per Jot

Replies
Relays
Thumbs ups
Thumbs downs
Ignores
Favorites
Popularity

Hot/Cold rankings

Most followed tags
Most muted tags
Most replied jots
Most relayed jots
Most liked jots
Most disliked jots
Most ignored jots
Most interaction (dialogs)
Most active friends

(-Least (x) also available)

Misc

Friends map
Tag map
Likes/Dislikes/Favorite/Ignore map
Who posts when

Vanity profile

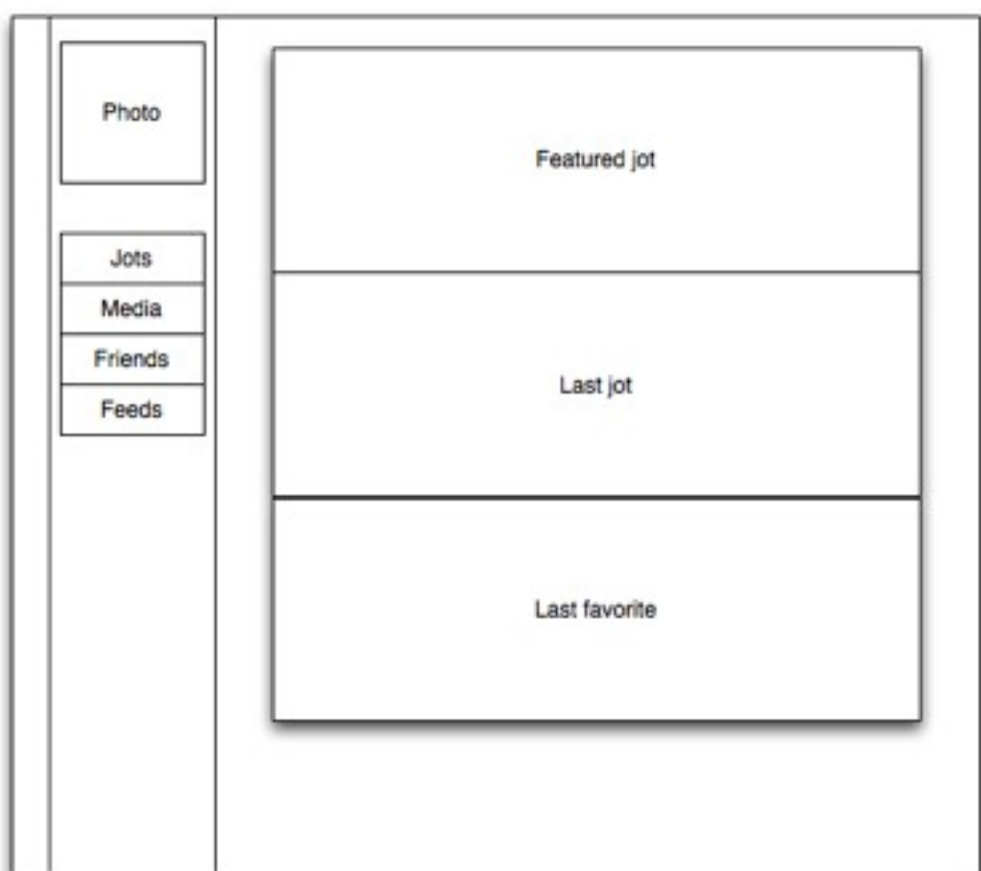Users shall be able to manage a vanity profile accessible through
USERNAME.vortex.com.
The target of this page is to allow users to publish their jots through a more personal and developed interface.

The profile is composed of a single page, showing the following elements of a Vortex user profile:

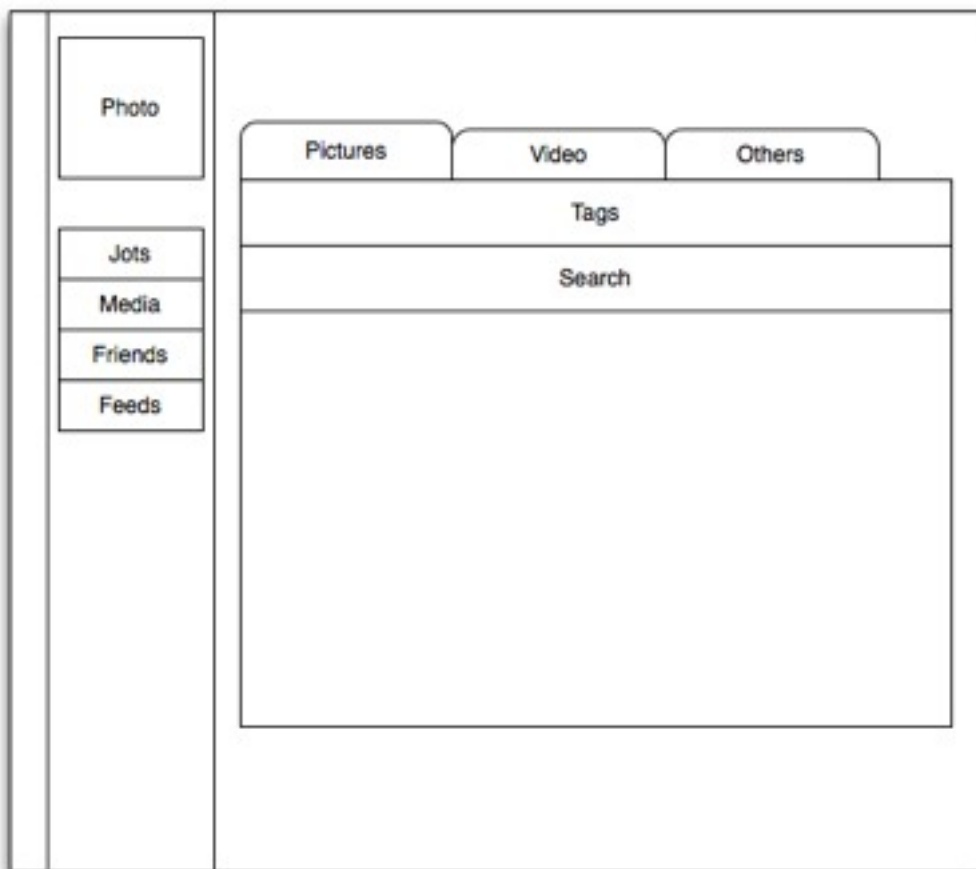- Avatar
- Jots
- Media
- Friends
- Feeds
- Stats

Users can chose wether to display these items via Vortex setting page.
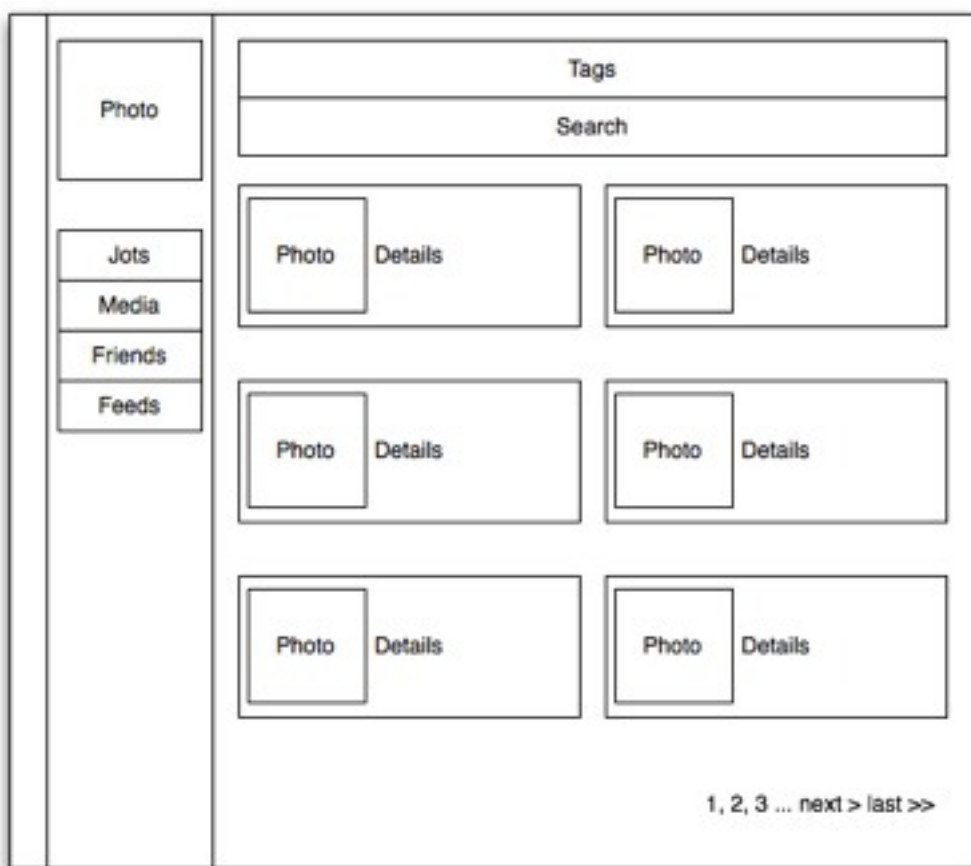The page is templated, and users should be able to upload different styles.

Jots

- Users can chose to feature a jot
- Last jot and favorite is automatically shown (if set in the preferences)
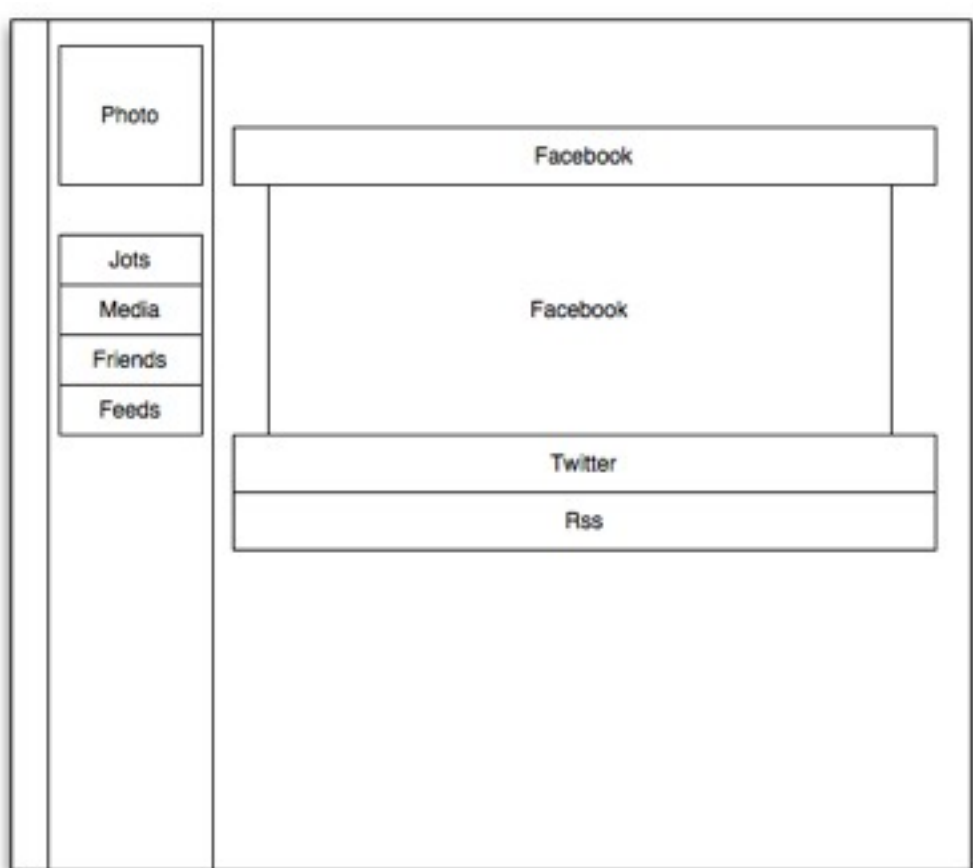
Media

- Media are regrouped by type, then by tags.
- Searching for a tag is possible, filtered by date/popularity /activity

Friends

- Friends are searchable by tags, filtered by
     - last activity
     - popularity
     - amount of friends/followers
     - places

- The details next the the photos will be consistent with the search (shows date when
  searched for last activity, etc...)

Feeds

What is shown from the feed depends on the connected site's API and what the user allows in their settings.

API

The API should enable 3rd parties to use the exact same features as the the web interface.

Registration

All third parties developers must possess a Vortex account and register to the developer database to obtain a developer key.
All third parties application must be registered under the developer's name in order to obtain an application key.

Access

Third parties can only access the API with both a developer and an application key.

The API should be accessible via to HTTPS entry points:

https://vortex.com/api/input

https://vortex.com/api/output

Input will handle every incoming data
Output will handle search queries and all other outputs

Tools

Data output from the API should be presented on several formats (JSON, HTML and XML) depending on the developer's needs.

Limitations

As describe in the technical requirements V1.0

Portability

The API will be developed in two steps.

> 1) On the top of Vortex V1.0, it will mainly be used to allow access from mobile devices and third party web services.

> 2) As a core function of Vortex V2.0, powering not only mobile and third parties but also the main website and vanity profiles.

Keeping that in mind, the API must be developed so the changes between V1.0 and V2.0 are minimal on the third party side.

APPS

Service	Feature
Facebook	Update to wall Import photos Export photos "Like" jots
Twitter	Cross post to twitter Import tweets Retweet jots
Foursquare	Check in
Picasa	Import photos Export Photos
Flickr	Import photos Export Photos
DeviantArt	Import photos Export Photos
Panorameo	
Youtube	Display thumbnails
Vimeo	
Soundcloud	Display songs
Itunes	
Beatport	
Issuu	
RSS	Turn saved searches and user feeds into RSS feeds

Modifications

Delete jot

A jot should be deletable.
Deleting a jot will not remove it from the database.

Edit jots

Jots should be editable. Prior editing is kept in the original jot.
Title, body and tags are editable.

Guest features

Features

Guests can use:

- All search features, except save searches
- pre-defined nests
- pre-defined favorites
- Drag box (only two options: login and help)

When trying to use other features, they are prompted to log in.

Content

Most popular:

- Jot (1)
- Pictures (4)
- Video (4)
- Music (3)
- Files (2)
- Places (3)
- Users (3)

These categories should be modular, and will be implemented independently from one another.
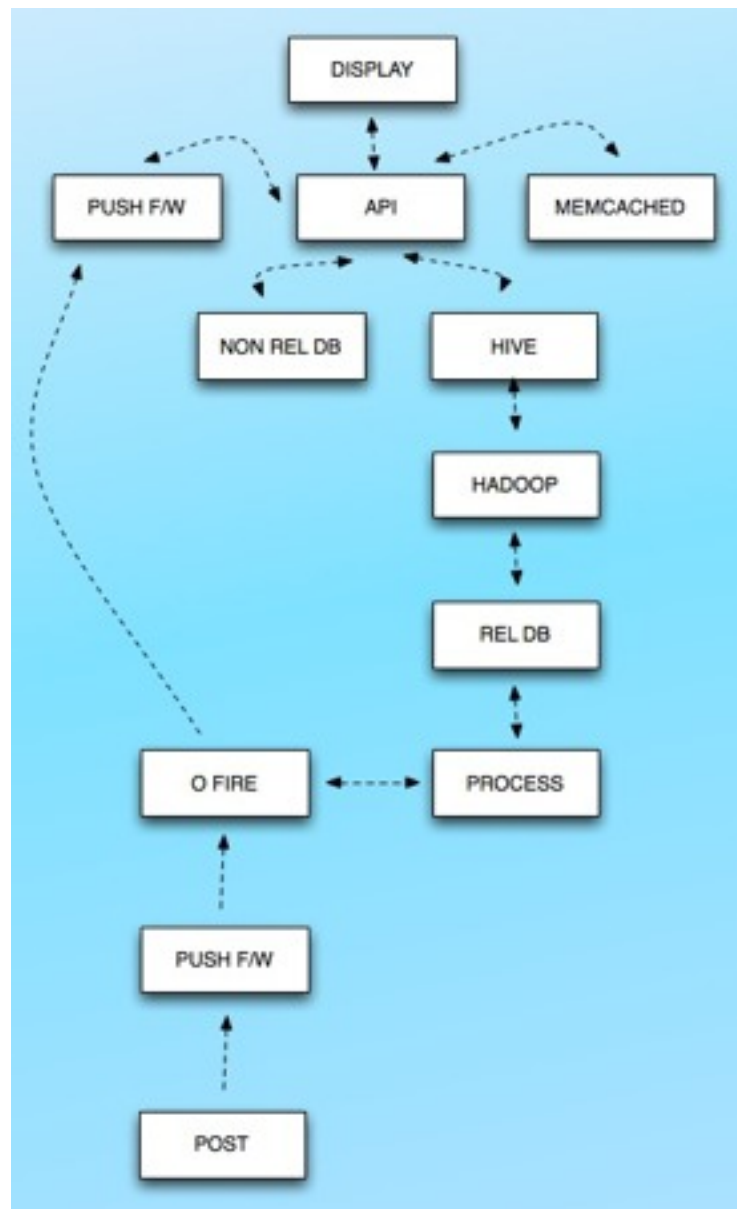
XMPP Integration

Use of a push technology over XMPP

It is not clear yet wether we plan to chose XMPP over frameworks such as JsJAC or Strophe, or rather implement the WAVE protocol.
Here is a partial rendition of the infrastructure we have in mind:

(Push FW is still undecided, Open Fire is an example of XMPP server)

Integration of live chat w/ video and voice

Live chat and VoIP are natural extensions of XMPP, it is natural then to provide these services.

Integration to live messaging clients supporting XMPP

Here, the same goes as with VoIP.

Extension of Vortex content to the rest of the web

Throught Vortex API, it will be possible to build a real time update system and give an identity to each piece of content submitted to the site, along with the relational status of its tags.