Current Users should not be required to set up their own server nor have their data compromised. There should be mechanisms in place so that the users can host their profiles in any server without running the risk of having their data compromised. Some DOSN platforms must provide their services via a web browser and through well-defined APIs (for mobile applications). Users should not be required to install their own server nor “trust” system administrators to host their profiles. There should be mechanisms in place so that the users can host their profiles in any server without running the risk of having their data compromised.

Communication protocols provide the overlay of the DOSN. Some protocols have been developed focusing specifically on DOSN communication.

### Protocols Used by Existing DOSN Platforms

- Representation
  - Open Graph Protocol
  - XOOO
- Content
  - XRI
  - OEmbed
  - PubSubHubbub
- Exchange
  - RSS / Atom
  - Salmon
  - P2P
- Identification
  - OpenID
  - OAuth
- Security and encryption
  - SSL
  - PGP
  - OpenPGP
- Social Representation
  - FOAF
  - XFN

User Protocols deal with authentication and identification. User authentication across services is an essential part of DOSNs. User identification provides limited and controlled user data to service and application without giving up privacy.

### DOSN Platforms

**Trendy ...**

- Diaspora*
- tent
- Status.net

**Trendiest!**

- Jappix
- Mr. Privacy
- Comet
- FOAF
- Diaspora*
- Pump.io

![Taxonomy by Protocols and Release Stage](image)

The three circles represent the release stage of the platform (i.e., the maturity of the platform).

- XMPP
- XMPP + Others
- XMPP + Diaspora
- Diaspora + Others
- Partial Diaspora + Others
- Others

**Layered Overview**

**Social Content**

- Activity Streams
- RSS/Atom
- PushHubSubhub
- Salmon

**Protocols used:**

- XMPP
- XMPP + Others
- XMPP + Diaspora
- Diaspora + Others
- Partial Diaspora + Others
- Others

**Food for thought...**

**Limitations**

- Current federated DOSN (e.g., Diaspora*, Tent and Friendica) platforms have not been widely adopted mostly because the users require technical skills to install their own server. Otherwise they have to “trust” an administrator to host their profiles.
- Peer to Peer applications (e.g., Retroshare) have not been widely adopted because the user interface is different from widely used OSNs (mostly Facebook and Twitter) and also because an installation process is required.

**Challenges**

- DOSN platforms must provide their services via a web browser and through well-defined APIs (for mobile applications).
- Users should not be required to install their own server nor “trust” system administrators to host their profiles.
- There should be mechanisms in place so that the users can host their profiles in any server without running the risk of having their data compromised.

**Research**

After taking a look at the existing DOSN platforms and protocols, we spotted their limitations and elaborated on how to overcome them.

- How to provide a user-friendly DOSN service that overcomes the limitations of the current platforms?