

The 101 on XMPP

The [Extensible Messaging and Presence Protocol \(XMPP\)](#) is a communication protocol originally developed by the Jabber open source community in 1999 as a tool for instant messaging and has been used for a wide range of publish-subscribe systems throughout the years. Web-based XMPP clients like the one Nexedi is developing and using internally are implemented using [Bidirectional-streams Over Synchronous HTTP \(BOSH\)](#), a transport protocol that simulates a bidirectional stream between two entities (for example client and server) to allow exchanging message as soon as they are sent by using multiple synchronous HTTP request/response pairs.

The Open-PAAS Project

Launched in August of 2015, the Open PAAS project aims to create competitive open source alternatives to [Google Docs](#), [Office365](#) and similar cloud-based communication and collaboration tools. One of the key objectives of the Open PAAS project and important differentiation criteria to other "desktop" open source projects shifting to the cloud ([Libre Office](#), [Open Office](#)) is the focus on data privacy and the ability of users to keep full control over the data they generate and collaborate on.

Nexedi role in Open-PAAS: OfficeJS

Within our internal [UNC](#) project, Nexedi has been working on various web based document editors and communication tools for quite some time. Lately efforts have been channeled into [officejs.com](#), Nexedi's HTML5 Web App Store, which aims to provide web-based, offline-ready productivity applications ranging from task managers, editors to various communication tools - including our Jabber client. This puts us straight on track for the Open-PAAS project, in which we will focus on both providing specific components and applications as well as our expertise in cloud solutions through [SlapOS](#).

Nexedi New Jabber Client

Our Jabber client has been in use internally for quite some time and everyone at Nexedi is suggesting improvements to eventually be added to the client. The main feature making its way into this release is the visibility of offline contacts along with chat history accessibility. Before, we could access the chat history of a contact while the user himself was online. With the new release this constraint has been removed making the chat protocol accessible independent of a contact's connection status. In addition, there is also a new password reset option to go along with a rework of error handling taking better care of timed-out sessions and failed login attempts.



Under the Hood - RenderJS and jIO

Our Jabber Client was one of the first applications written using our two front end frameworks [RenderJS](#) and [jIO](#). The application is fully built using RenderJS gadgets (reusable components made from HTML, JavaScript and CSS) and with the latest update utilizes jIO to maintain a chat history in the browser's storage ([indexedDB](#)) keeping conversation histories on the respective users' devices. The additional advantage of using our two standard front-end frameworks is ease of integration with other applications being written using RenderJS. This way our jabber client could also be used as a drop-in chat system whenever needed with the storage being switched to the source and destination of communication data.



Roadmap

We are still adding more features to the client mostly related to configurability (server to connect, storage to use) before it will become our first productivity app to be officially released. If you already want to try the current version, you can find it stored as a [business template](#) in the [ERP5 website](#). Once we are ready to go fully public, we will bring another blog post with tutorial, so stay tuned.