XMPP VS. Web Sockets

[XMPP server: ejabberd vs Openfire vs prosody](https://stackoverflow.com/questions/33596842/xmpp-server-ejabberd-vs-openfire-vs-prosody)

[RabbitMQ Vs. Ejabberd?](https://stackoverflow.com/questions/27860100/rabbitmq-or-ejabberd-which-one-should-i-use-for-one-to-one-chat)

AMPQ vs XMPP and RabbitMQ vs ejabberd

XMPP Overview:

Pusher Vs. Pubnub Vs. Sendbird Vs. Firebase

SignalR Vs. Socket.IO

Intro to XMPP and the important things:

**XMPP Overview:**

The Extensible Messaging and Presence Protocol (XMPP) is an open technology for Realtime communication, using the Extensible Markup Language (XML) as the base format for exchanging information. In essence, XMPP provides a way to send small pieces of XML from one entity to another in close to real time.

It’s helpful to break the XMPP universe down at a high level into **services** and **applications**:

* The services are defined in two primary specifications published by the Internet Engineering Task Force (IETF) at http://ietf.org/ (the “RFC” series), and in dozens of extension specifications published by the XMPP Standards Foundation at http://xmpp.org/ (the “XEP” series)
* The applications are software programs and deployment scenarios that are of common interest to individuals and organizations, although the core services enable you to build many other application types as well.

More about services and applications:

* Services: In this context, a service is a feature or function that can be used by any given application. XMPP implementations typically provide the following core services:
  1. Channel Encryption.
  2. Authentication.
  3. Presence.
  4. Contact lists (Trusted list for entity on the server).
  5. One to one messaging.
  6. Multi-party messaging (enables you to join a virtual chat room for the exchange of messages between multiple participants, like Internet Relay Chat (IRC). The messages can be plain text, or can contain XML extensions for more advanced functionality, such as room configuration, in-room voting, and various session control messages.).
  7. Notifications.
  8. Service discovery (enables you to find out which features are supported by another entity, as well as any additional entities that are associated with it (e.g., rooms hosted at a chat room service)).
  9. Capabilities advertisement (This service is an extension to the presence service that provides a shorthand notation for service discovery data so that you can easily cache the features that are supported by other entities on the network).
  10. Structured data forms.
  11. Workflow management.
  12. Peer-to-peer media sessions (This service enables you to negotiate and manage a media session with another entity. Such a session can be used for the purpose of voice chat, video chat, file transfer, and other real-time interactions.).
  13. There are also a lot of additional services that provided as extensions via the community.
* Applications: With in the above services, you can build the below awesome applications.
  1. Instant messaging: The classic instant messaging systems that most people are familiar with combine three of the core services: presence, contact lists, and one-to-one messaging. Such systems can and often do include more services and features, but if you have these three services, you can build a bare-bones IM application.
  2. Groupchat.
  3. Gaming (<https://www.chess.com/> is entirely build on XMPP)
  4. Systems control: The combination of one-to-one messaging and data forms makes it possible to deploy lightweight systems for control of and interaction with remote systems. Deployed applications in this domain include network management, scientific telemetry, and robotic control.
  5. Geolocation: The XMPP notification service is payload-agnostic. One defined payload format is geolocation, which enables you to build fascinating location-based applications, such as vehicle tracking.
  6. Middleware and cloud computing
  7. Data syndication: Popular social networking applications are increasingly using the XMPP notification service to solve a particular problem they have: constant polling for updated information. Existing HTTP-based deployments have been found not to scale, because quite often a particular feed has not changed since the last time it was polled. By contrast, the XMPP notification service sends out an update only when a feed has changed, saving a significant amount of bandwidth and server resources that otherwise would be wasted on polling.
  8. Voice over IP (VoIP)
  9. Identity services: Given the existence of stable identifiers (JabberIDs) and a robust authentication service, it is possible to use XMPP in building identity and authorization services such as OpenID and OAuth.
  10. There are a million of applications and ideas that you can make using the core and extension services of XMPP.

Brief:

Although XMPP was originally developed in the Jabber open source community, the protocol itself is not an open source project like Apache, but rather an open standard like HTTP. As a result, XMPP is an open technology that is not tied to any single software project or company. The XMPP specifications define open protocols that are used for communication among network entities. Much as HTTP and HTML define the protocols and data formats that power the World Wide Web, XMPP defines the protocols and data formats that power real-time interactions over the Internet. The protocols are as free as the air, which means they can be implemented in code that is licensed as free software, open source software, shareware, freeware, commercial products, or in any other way. This open standards model is different from the open source or free-software model for software code, wherein the code is often licensed so that modifications must be contributed back to the developers.

XMPP and Jabber

Since 2004 XMPP is the IETF-approved instant messaging and presence protocol.XMPP is based on the Jabber protocol that evolves since 1998. Basically programming Jabber software today means programming XMPP compliant software. XMPP is the new name, Jabber the old.

**XMPP Basics:**