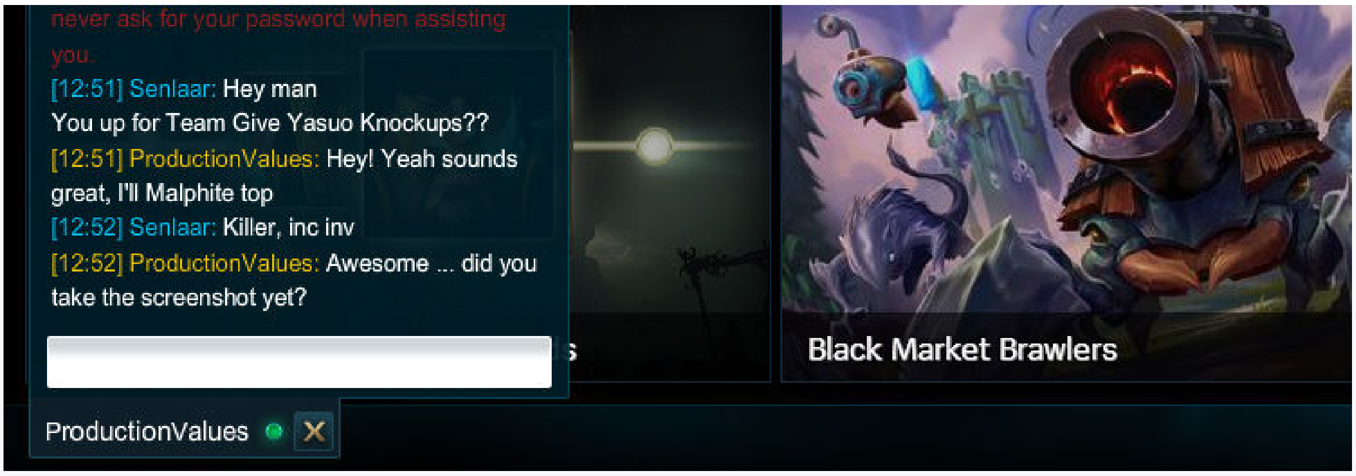
never ask for your password when assisting you.
[12:51] Senlaar: Hey man
You up for Team Give Yasuo Knockups??
[12:51] ProductionValues: Hey! Yeah sounds great, I'll Malphite top
[12:52] Senlaar: Killer, inc inv
[12:52] ProductionValues: Awesome ... did you take the screenshot yet?

CHAT SERVICE ARCHITECTURE: PROTOCOL

Some games of *League of Legends* I know I'll love even before the loading screen appears. Take last night, for instance. I logged in to find a couple of fellow Rioters available to play, and while we queued together in a pre-game lobby we had a great discussion that led to smart role choices and an intentional team composition. That connection and strategic communication doesn't guarantee victory — we lost — but win or lose that's my favorite way to start.

LoL chat services support that pre-game experience. Every day players rely on chat to catch up with friends, arrange and plan games, celebrate victories, and mourn defeats. Because chat holds a significant role in the player experience, over the last few years we've exerted significant effort to make it more scalable, performant, and secure.

I'm excited to share our journey developing chat by diving into the relevant engineering through three posts on our most important backend components. This will also be a chance to open a dialogue for your questions and comments. I'll deliver one post per component, filled with stories of our decision-making, successes, and failures on the road to providing players with an enjoyable chat experience. I hope these articles provide interesting engineering insight into one aspect of the game, inform the work of anyone working on a similar project, and spark useful conversation here and elsewhere.

The three backend components are the following:

- The Extensible Messaging and Presence Protocol (XMPP) used for client/server communications and extended internally to meet *League of Legends*' specific needs.
- Chat servers written in Erlang and C used to communicate with clients, initially based on the open source version of ejabberd and rewritten over the past few years.
- A persistent data store used for the social graph, ignore lists, offline messages, message history, and other features.

We'll start off this series with a dive into XMPP protocol and how we use it!

XMPP

In *League*, chat plays a crucial role in keeping players in contact. Players need to see who else is online (and signal their own availability), exchange messages, add notes on friends, and sometimes block future interactions. To accomplish this, chat clients must speak a command language with the servers.

We took six factors into consideration when choosing a protocol:

- Support for 1:1 messages, group chats, and offline messages. (These fundamental concepts support the development of player experiences like private messages, game lobby conversations, and pre- and post-game chats without the need to reinvent the wheel.)
- Built-in presence mechanisms. (These notify players of their friends' availability.)
- Open specifications for transparency and auditability; a wide review scope;and, allowance for Riot-specific tuning.
- A proven security model. (This ensures we can protect the privacy of players.)
- An extensibility model. (This allows for custom behaviors.)
- Availability of open source projects. (This accelerates development and delivery of chat to players.)

We evaluated several alternatives and XMPP best satisfied all of the above factors. XMPP is a XML-based protocol designed primarily for messaging and presence propagation, defined by three main RFCs: RFC 6120 (XMPP Core), RFC 6121 (XMPP Instant Messaging and Presence) and RFC 6122 (XMPP Address Format). Specifications and reference implementations are publicly available at http://xmpp.org/.

Each time I check a friend's availability or status, discuss preferred roles during the champ select phase, or express satisfaction and humility after a decisive victory in the post-game chat, my client sends and receives a number of so-called XMPP "stanzas." The type, layout, and content of each stanza depend on the context, but you can think of them as commands sent to the server.

For instance, in order to broadcast availability to friends, the client sends:

```
<presence type='available'/>
```

To ask the server to fetch a full friends list:

```
<iq type='get' id='roster1234'>
 <query xmlns='jabber:iq:roster'/>
</iq>
```

And to message a friend:

```
<message to='sum1234@pvp.net'>
  <body>Hey there, wanna play?</body>
</message>
```

These three stanza types (*presence, iq,* and *message*, respectively) are the building blocks of any XMPP-based application protocol. But for *League*, our needs go beyond these types. One main feature of XMPP is how it gives developers freedom to define custom extensions in order to support deployment-specific needs. I'll explain the extension mechanism and philosophy in detail in the following section.

XMPP EXTENSIONS

Apart from the official RFCs, the XMPP Standards Foundation provides a framework for introducing official extensions to the protocol by means of XEPs (XMPP Enhancement Proposals). These define custom behaviors that expose new functionality, alter existing functionality, or suggest best practices when using the protocol. A fuller description and a comprehensive list of XEPs can be found here: http://xmpp.org/xmpp-protocols/xmpp-extensions/.

From the very beginning, we were aware that the core XMPP features and existing XEPs might not be enough for us to reach our goals with regards to game-specific features, scale, and performance. Since *League of Legends* chat has some properties that are not part of any of the existing XMPP extensions and has its own proprietary client, we introduced a number of protocol extensions that alter both core and existing XEPs. We maintain these extensions on an internal wiki page called the RXEP Library (Riot XMPP Enhancement Proposals) and use it to document any changes.

Let's review two of the extensions we introduced to the RXEP Library: friend roster notes and incremental privacy list updates.

XMPP EXTENSION #1: FRIEND ROSTER NOTES

Often times I send friend invites to people I've met in game that I'd love to play with again in the future. I enjoy their clever and fun summoner names, but days later I often have trouble distinguishing who is who, and what their preferred play style is. To solve this, every time I add someone to my friends list I attach a short note to their name explaining their best position, favorite game mode (ranked 5s!), and sometimes the context in which I met them. "Great Darius player, Summoner's Rift only." "Friendly plat guy, met him at PAX." "Mom."

To enable this type of functionality, we designed a very simple extension to the roster protocol. According to the RFC 6121, an *item* element represents each friend on the roster. Contacts can have multiple parameters, such as a summoner name (*name* attribute), internal Jabber identifier or JID (*jid* attribute) used for internal routing, and their group (*group* sub-element):

```
<item jid="sum9876@pvp.net" name="0xDEADB33F">
  <group>General</group>
</item>
```

We introduced an additional optional sub-element *note* that holds short, contact-specific text written by the player:

```
<item jid="sum9876@pvp.net" name="0xDEADB33F">
  <group>General</group>
  <note>Versatile dude, plays only ranked, not a big fan of ARAMs</note>
</item>
```

The note itself is only visible to the contact owner (my friend cannot see it) and can contain any content I wish: comments, notes, the player's real name, etc. In order to protect players' privacy, we ensure that the notes stay encrypted on disk using strong cryptography algorithms. We encrypt payloads — before persisting in a data store — with AES-CTR-128, while we use HMAC-SHA1 for signing the data. We also implemented support for rotating encryption keys should we decide to update them in the future.

XMPP EXTENSION #2: INCREMENTAL IGNORE LIST UPDATES

Unfortunately, during some games on the Rift I encounter players with whom I'm not interested in having any future interaction. I can block these people using the ignore feature, stopping any sort of out-of-game communication from them to me (including sending regular messages or friend invites).

A fairly complex XEP describing so-called privacy lists management is found here: http://xmpp.org/extensions/xep-0016.html. However, it only provides get/set semantics. According to the extension, we can't modify the existing privacy list by just adding or removing an item. Every change I make requires uploading the whole list to the chat server. Since ignore lists can grow relatively large, transmitting them back and forth between server and client is both costly and useless. To address that issue we added two new commands to the privacy lists API.

In order to add an item to the privacy list, clients issue:

```
<add name='LOL'>
  <item value='sum1234@pvp.net' action='deny' order='1' type='jid' />
</add>
```

And to unblock someone:

```
<remove name='LOL'>
  <item value='sum1234@pvp.net' />
</remove>
```

PROTOCOL COMPATIBILITY

I might want to stay in touch with *League of Legends* friends and use chat to communicate with them outside of the game. A number of community-driven projects bring the Riot chat experience to desktop or mobile devices using standard XMPP channels, such as LoL Connect for iOS or LoL Chat for Android.

We want to make sure any changes made to the protocol are backward compatible and that third party clients aren't kicked out of the ecosystem. Certain features (such as roster notes) might not be available using standard XMPP libraries. However, with a relatively low effort developers can add them to the client libraries.

When designing new extensions we also have to make sure that any future potential engagement channels such as web, mobile, or Riot's next game will also fit into our chat world and allow players to seamlessly communicate with each other.

SUMMARY

I hope this post provides insight into how we use XMPP for Riot chat to deliver an awesome experience in and out of the game. At the same time, I am fully aware that we just scratched the surface here, and that's where we need your thoughts. Please post comments and questions, and I'll do my best to provide answers from my team.

Also, if you'd like to learn more about chat services, I recently gave a talk entitled "Scaling LoL Chat to 70 Million Players." A summary of that talk is also available at highscalability.com.

But wait, there's more! In the coming months I'll be back with the next part in this series on Riot chat, which will examine the server-side architecture and persistency layer for the social graph. Stay tuned!

Posted by Michal Ptaszek

19 Comments   Riot Engineering Techblog   Kaizhao Zhang

Recommend 8   Share   Sort by Best

Join the discussion…

kipp14 • 2 months ago

Can you guys add voip to chat it would help with a lot for teams to get to know each other. Thanks

Reply • Share ›

Hasan • 2 months ago

Good read,

got a question, XMPP uses a client server model ? as in when I message a friend the messages are directed to a server and then routed to this person ?or does the server just serve as a database?. In case of the first one, wouldn't a peer to peer system work better ? Also does the server log the friend messages ? or are they all deleted once I log out.

• Reply • Share ›

michalptaszek Rioter Hasan • 2 months ago

Thanks Hasan!
Indeed, XMPP is architected in a way that assumes the messages are routed through the server. Although p2p communication would be more efficient (from the servers standpoint) it suffers from several problems:
* certain messages can not be easily validated and could potentially be spoofed (e.g. timestamps attached to each message)
* we would not be able to apply many validations (e.g. rate limiting)
* it would expose other players IP addresses (which in turn would lead into security problems, e.g. drop hacks)

In the same time we are currently building an extension that allows servers to store historical messages, so that whenever you switch computers, your conversation history would travel with you.

1 • Reply • Share ›

Hasan michalptaszek • 2 months ago

Thanks for the reply ! didn't know p2p communication would suffer from many security issues

• Reply • Share ›

Leandro Ruel • 2 months ago

would be great a notification about a friend who login in the chat

• Reply • Share ›

swanboy • 2 months ago

This is really cool. I read over it and the summary you linked to since I am looking at building a mobile app that will make use of chat. I have a few questions if you don't mind:

1. Since you're trying to leave MySQL due to apparently slow access times, what database system are you planning to migrate to, Riak?
2. Was the older chat system ever the cause for notable service crashes and long login queues?
3. As part of your journey to transcend the LoL client, will you look at mobile in addition to desktop options?
4. Are you ever planning to release some of the optimization work that you did on Ejabberd?

• Reply • Share ›

michalptaszek Rioter swanboy • 2 months ago michalptaszek Rioter swanboy • 2 months ago

Hi swanboy,

1. Yes, it's Riak we are migrating to. Response times weren't the only reason why MySQL lost the battle (in fact, scaling MySQL servers vertically by running it on e.g. FusionIO hosts makes them über powerful). The migration itself allows us to also iterate much faster on features (schema-less design of the data store), scale horizontally (by adding more servers if we need more capacity or throughput) and achieve better uptime (no single point of failure). I will get back to whys and so whats in the third part of the series.

2. Yes, in the early days, when we were still struggling with making it stable and fault tolerant it occasionally crashed/lagged - and was causing players to relog to the game, effectively storming login queue :(
3. Yup :)
4. If you run a diff on our chat servers codebase and compare it original ejabberd I would be surprised if you found more than 10% of matching code. We diverged from the core so much that merging the changes back would be really painful and would significantly slow down development of the new features. We've also removed a lot of things Riot chat does not need (which are however useful in the generic open source implementation). In the same time, if you are still interested in certain optimizations/pieces of code after reading the second part of the article (which covers the server side implementation), let's sync up and discuss the interesting bits and pieces.

• Reply • Share ›

swanboy michalptaszek • 2 months ago

Interesting and thanks! I will be on the lookout for your next article as I try to figure out what my actual needs are.

Somewhat related question: how does the server determine if someone is Away? Is it related to minimizing the client or is there an activity time limit of some sort?

• Reply • Share ›

hazzhaston • 2 months ago

Is it possible to extend chat services to the game loading screen? I don't know any games that does this but it seems like a practical idea.

• Reply • Share ›

michalptaszek Rioter hazzhaston • 2 months ago

We've thought about adding this feature and it's definitely something we still are considering but are currently working on other high priority features. Thanks for bringing it up though!

• Reply • Share ›

Leandro Ruel michalptaszek • 2 months ago

[illegible]

+1 to this ideia. I can finally say to my friend: "look that guy, his conexion is a modem of 56kbps"

^ | ⌄ • Reply • Share ›

Marcelo Bonatto • 2 months ago

"I logged in to find a couple of fellow Rioters available to play,"

Random thought: Does Riot have private in-game chat rooms for employees? Like a General chat that all Rioters are connected to when online, some team chats, etc. Or you found only Rioters that you had previously added to your friends list (that has a limit)?

I'm sure you guys have communication tools outside of the game, but because playing League is core for y'all I can only this would be a must.

^ | ⌄ • Reply • Share ›

michalptaszek Rioter ➦ Marcelo Bonatto • 2 months ago

Hey Marcelo,

No, currently there is no such a thing. All named group chats (i.e. you create them or join by name) are public, and accessible by anyone who knows their names. It is however possible to create a private group chat and let people join it by inviting them manually.

Also, Rioters do not have any special privileges in chat (other than a possibility to get "Riot" prefix in front of their summoner names) - they have the same rights/limits/capabilities as all other players.

^ | ⌄ • Reply • Share ›

Kevin Doyon ➦ michalptaszek • 2 months ago

Do they have any perks/privileges at all besides having all skins and champions unlocked? Maybe the ability to have a level 30 account without having to level it up or something like that?

^ | ⌄ • Reply • Share ›

Andrew McVeigh Rioter ➦ Kevin Doyon • 2 months ago

I can attest to the fact that Riot will not gift you a level 30 account, even if you technically need one to do your engineering work - you have to play your way to it. When I joined I asked to get one, and was told it wouldn't be fair to players if engineers didn't go through the same process as every other player.

Even though it hurt a bit at the time, the message is pretty clear ;-P

1 ^ | ⌄ • Reply • Share ›

michalptaszek Rioter ➦ Kevin Doyon • 2 months ago

There are certain perks you can take advantage of - such as account on PBE and on all internal environments, ability to play and see champions/changes that were not released yet, etc.

In the same time, certain skins are not unlocked for Rioters either (e.g. Championship Elise and Triumphant Ryze) - you have to earn them in the same way as non-Rioters would do.

Reply • Share ›

Matthew Inger • 2 months ago

Its worth noting as well that implementations of XMPP into some clients can lead to undesired results if you aren't careful with parts of the protocol you may not wish to allow such as:

VCard lookups : That allow user information enumeration
Ping : Which can divulge remote IP information

Just to name a few.

Be cautious in the use of this protocol and be sure to adopt a policy of least privilege - use only what you need and ensure all other services are disallowed by your chat servers.

-- Matt

Reply • Share ›

michalptaszek Rioter ➦ Matthew Inger • 2 months ago

Hi Matt,

thanks for pointing this out - you are absolutely right.

Most of the open source XMPP servers by default enable popular extensions, such as pings, vcards or pub/sub. Riot chat servers support only features that are essential to the players and which don't expose unwanted data.

To add to that, certain features that are enabled (e.g. support for friends lists) were updated to protect the live service from being abused. For instance chat servers impose limits on the maximum length of your roster, lengths of your friend notes, and rate at which you can invite other players.

Reply • Share ›