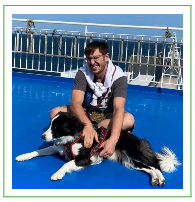
Roman Gafiyatullin

☎ +370 (693) 79 061
✉ r.gafiyatullin@me.com
https://github.com/RGafiyatullin



Education

2004–2008 **Minsk State Higher Radioengineering College**, *Software and IT*.

Languages

Russian	**Native**
English	**Advanced**

Technologies actively used

Rust
PostgreSQL

Technologies recently used

Scala/Akka
Erlang/OTP
MySQL

Open source

2021 **RGafiyatullin/reopenconnect**, *An implementation of OpenConnect client that allows to customize the data sent to the server.*
`https://github.com/RGafiyatullin/reopenconnect`

2018 **RGafiyatullin/la-rete**, *Rete-like pattern-matching algorithm.*
`https://github.com/RGafiyatullin/la-rete`
`https://github.com/RGafiyatullin/la-rete-rs`

2017 **RGafiyatullin/scala-xmpp-akka-stream**, *XMPP-protocol library (akka-stream).*
`https://github.com/RGafiyatullin/scala-xmpp-akka-stream`
`https://github.com/RGafiyatullin/scala-xmpp-protocol`

2015 **netvl/xml-rs**, *Xml pull-parser for Rust.*
`https://github.com/netvl/xml-rs/graphs/contributors`

2015 **uutils/coreutils**, *Rust implementation of unix-utilities.*
`https://github.com/uutils/coreutils/tree/master/src/uu/expr`

2015 **RGafiyatullin/owl_xmpp**, *Erlang XMPP-protocol library.*
Erlang implementation of XMPP stream protocol.
`https://github.com/RGafiyatullin/owl_xmpp`

2015 **RGafiyatullin/orca**, *Erlang MySQL-client.*
This driver has been implemented after a some period of fiddling with Emysql.
Emysql cannot do pipelining: the connection sleeps each time after it is returned into the pool and until it's acquired again. This results in the total inability to utilize the database ('show processlist': everyone sleeps).
You can organize your own pool: worker-pool instead of connection-pool (`https://github.com/RGafiyatullin/emysql/`). But still you have to work with sockets the passive-way.
When the connection is closed from the remote side, that fact gets known right at the moment that connection is most necessary (upon query).
So was decided to reimplement the database-client from scratch. Orca was supposed to be the MySQL-client done correctly.
`https://github.com/RGafiyatullin/orca`

Experience

2017–2021 **Solution Architect**, *Wargaming.net*, Minsk,Vilnius.
Working as a solution architect on Wargaming's own platform similar to Steam and Playfab.
Evangelising the adoption of Rust in the development process.
Leading the development of

- AGate — the API gateway providing the access to the Platform functionality via the simple OpenAPI-compatible HTTP-interface.
- Estrellas — a generalised storage for relations between enttities. The main application for that is the Clans subsystem and Campaign management.
- Tempora — the archive database for Estrellas that provides a possibility to fetch arbitrary historic states of the entities.
- Snitch — the usage audit subsystem that is used to bill the clients based on the computational resources spent on their needs.

2012–2017 **XMPP Services Developer**, *Wargaming.net*, Minsk.
Leading a cozy development team of three programmers.
Detailed achievements:

- Initial development of internal XMPP-server (not based on eJabberd);
- Architecture of XMPP-service capable of dealing with kind of a massive load (WoT game project):
  - over 600K CCU (registered peak – slightly over 1KK CCU);
  - 10M of active users with average of 20 mutual friends;
  - over 100K routing operations per second during peak loads;
  - worst 99.9-percentile latency of the latter routing operations is below 100ms;
- A set of measures to move out non-game functionality out of the game-server to the XMPP-services:
  - friendship graph;
  - multi-useer chats;
  - various notifications;
  - user-search;
  - ...

  It was still a work in progress when I left the project as there remained a plenty of stuff to be cut off the
  game server to leave it with its main functions.

2007–2012 **Working for quite a while at PowerMe Mobile**, *Minsk*.

2011–2012 **OpenAlley SMPP gateway**.
Tools:

- Linux;
- Erlang/OTP;
- C.

2011–2012 **Billing and rating subsystem**.
BMS v6 brings a new type of inbound interface – SMPP.
The special thing about it is that the customers using SMPP usually send
a vast amout of unique messages: e.g. banks are reporting most of credit/debit-card transactions' details via short messages.
The previous rating and billing system was designed to handle big amounts of
non-unique messages. Also the old system had a considerable latency which
is unacceptable for working with prepaid customers.
Tools:

- Linux;
- Erlang/OTP;
- C.

2010–2011 **Bulk Messaging System v6. Linux HA integration**.
Tools:

- Linux HA (corosync / pacemaker);
- Erlang.

2009–2010 **Bulk Messaging System v6. Porting to Linux**.
Some of the operators do not want to have any Windows in there data-centres.
The platform has to be ported to some Unix.
Since the system we already had by that moment was implemented using .Net – we chose Mono. Later most of the PMM's old customers requested an upgrade
to the Unix-hosted system. The previously implemented in Delphi SMPP and
MM7 gateway was rewritten from scratch in Erlang.
Tools:

- Linux (Mono);
- C#;
- MySQL;
- Erlang/OTP.

2007–2009 **Bulk Messaging System v5**.
A multi-interface gateway for sending short text messages and
multimedia messages into cellular operator's network.
Tools:

- Windows (.Net);
- C#;
- MS SQL;
- SMPP and MM7 protocols to communicate with operator's network.

2006–2007 **Junior developer at Oxford ArchDigital**.
Web projects based on proprietary CMS.
Tools:

- Linux;
- PHP;
- MySQL;
- PostgreSQL.

That was my first commercial development experience.
The customers mostly were british governmental institutions.
Most notable of them is Ordnance Survey – british national mapping agency.