

# THE LUCENE SERVER

## A BRIDGE BETWEEN TWO WORLDS

Fernando Benavides (*@elbrujoalcon*)

Inaka Labs

June 7, 2012



# TIGERTEXT

*TigerText is a secure mobile messaging platform that you set up for your hospital or business to improve workflow and reduce risk. Employees gain the convenience and familiarity of traditional texting while the enterprise retains the control and security it demands.*

(from [TigerText Site](#))



# AUTOCOMPLETE

*They wanted this. . .*

Share:  Status  Photo  Link  Video  Question

The Art of Dialog in the Age of Demons" by Robert



**Robert Rose**



**Dale Robertson**

Lancaster, Pennsylvania



**Roberto Rizaldy Gabriel**



**Roberto Cosentino**

Cairns, Queensland, Australia



**John Roberts**

Bangkok, Thailand



# SCOPE

*The goals for this talk are to show you:*

- how we connected two seemingly incompatible worlds
- how we started with a very specific problem and ended up with a very general solution



# THE PROBLEM

*Once upon a time on server world...*



# THE PROBLEM

*Erlang* and *Redis* were happily talking



# THE PROBLEM

But they needed *Lucene*, something that only *Java* knew. . .



# THE PROBLEM

... and *Java*, well...





*Oh! Now who can help?!<sup>1</sup>*



---

<sup>1</sup>Translated by Google™



*Yo! The red locust! They did not have my cunning!<sup>1</sup>*



---

<sup>1</sup>Translated by Google™



# JINTERFACE

*Jinterface Java package contains java classes, which help you integrate programs written in Java with Erlang*  
(from [Erlang Docs](#))



# JINTERFACE

- Comes with the Erlang distribution
- Is written in Java
- Let's you create nodes and processes inside a Java VM
- *We extended it to handle OTP gen\_server protocol*



# JINTERFACE

- Comes with the Erlang distribution
- Is written in Java
- Let's you create nodes and processes inside a Java VM
- *We extended it to handle OTP gen\_server protocol*



# JINTERFACE

- Comes with the Erlang distribution
- Is written in Java
- Let's you create nodes and processes inside a Java VM
- *We extended it to handle OTP gen\_server protocol*



# JINTERFACE

- Comes with the Erlang distribution
- Is written in Java
- Let's you create nodes and processes inside a Java VM
- *We extended it to handle OTP gen\_server protocol*



# LUCENE

*Apache Lucene™ is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.*  
(from [Lucene Site](#))





# LUCENE

- It lets us work with documents that represent users
- It analyzes, tokenizes and indexes the different fields
- *Using it together with JInterface, we built a Java node with a Lucene-backed Autocomplete gen\_server*



# LUCENE

- It lets us work with documents that represent users
- It analyzes, tokenizes and indexes the different fields
- *Using it together with JInterface, we built a Java node with a Lucene-backed Autocomplete gen\_server*



# LUCENE

- It lets us work with documents that represent users
- It analyzes, tokenizes and indexes the different fields
- *Using it together with JInterface, we built a Java node with a Lucene-backed Autocomplete gen\_server*



# WHAT WE'VE FOUND?

- Processing is fast, communications are slow
- Lucene is not complete, but it's really flexible
- JInterface is very basic, not exactly flexible, but in fact extensible



# WHAT WE'VE FOUND?

- Processing is fast, communications are slow
- Lucene is not complete, but it's really flexible
- JInterface is very basic, not exactly flexible, but in fact extensible



# WHAT WE'VE FOUND?

- Processing is fast, communications are slow
- Lucene is not complete, but it's really flexible
- JInterface is very basic, not exactly flexible, but in fact extensible



# GENERALIZING

- Now that we have it, we can use Lucene in other Erlang applications
- We split autocomplete from Lucene logic
- Then we've created a Lucene Server for Erlang
- Still in its early stages, there's a lot to improve on that



# GENERALIZING

- Now that we have it, we can use Lucene in other Erlang applications
- We split autocomplete from Lucene logic
  - Then we've created a Lucene Server for Erlang
  - Still in its early stages, there's a lot to improve on that





# GENERALIZING

- Now that we have it, we can use Lucene in other Erlang applications
- We split autocomplete from Lucene logic
- Then we've created a Lucene Server for Erlang
- Still in its early stages, there's a lot to improve on that



# GENERALIZING

- Now that we have it, we can use Lucene in other Erlang applications
- We split autocomplete from Lucene logic
- Then we've created a Lucene Server for Erlang
- Still in its early stages, there's a lot to improve on that



# RESOURCES

## ABOUT ME

- I'm **@elbrujohalcon** at Twitter
- I'm **elbrujohalcon** at Github

## ABOUT INAKA

- Check our website: <http://inaka.net>
- And our Blog that's on the same site

## ABOUT THIS TALK

- This talk is on Github:  
<http://github.com/inaka/talks>



# Thanks!

