

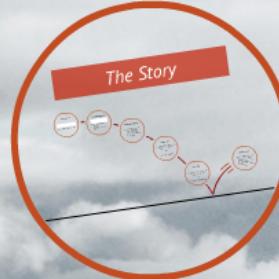
Lucene Server

From Erlang to Java and Back Again



Lucene Server

From Erlang to Java and Back Again





Hello World!!

About Me

- Programmer since the age of 10
- Coded in Basic, .NET, Java, JS, ...
- Erlang programmer since 5 years ago
- Director of Engineering at Inaka

About Inaka

- We're based in Argentina
- We do Erlang consulting
- We build end-to-end applications
- We develop highly-concurrent servers

About TigerText

- Based in Los Angeles
- Provides secure communication for mobile devices
- Also provides the underlying communication platform for other apps
- All servers written in Erlang



About Me

- Programmer since the age of 10
- Coded in Basic, .NET, Java, JS, ...
- Erlang programmer since 5 years ago
- Director of Engineering at Inaka



About Inaka

- We're based in Argentina
- We do Erlang consulting
- We build end-to-end applications
- We develop highly-concurrent servers



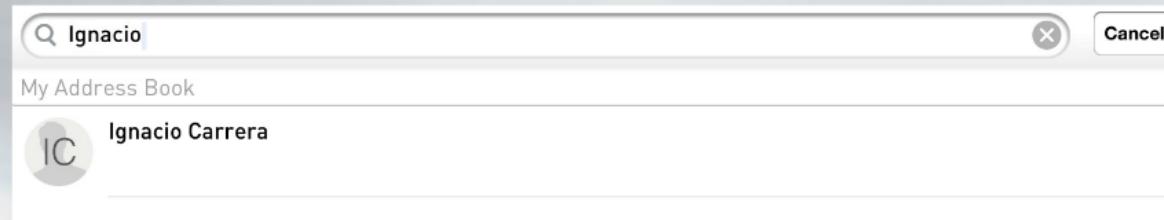
About TigerText

- Based in Los Angeles
- Provides secure communication for mobile devices
- Also provides the underlying communication platform for other apps
- All servers written in Erlang

The Story

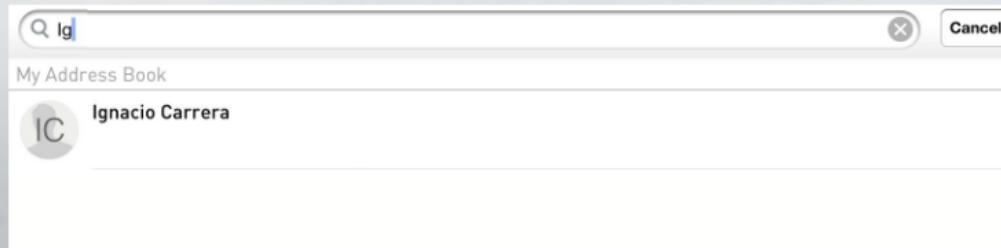


The Scenario



We had a very basic search field for users...

Functional Requests



We wanted to add the following features:

- Partial Matching
- Autocomplete
- Advanced Search (by field, location, etc.)
- Ranked Results

Technical Requests

We also wanted our solution to be:

- Implemented in Erlang
- Included in our existing servers as a rebar dependency
- Fast and scalable

Lucene Core

A high-performance, scalable, full-featured text search engine library with powerful, accurate and efficient search algorithms

...but...

It's written in JAVA

JInterface

A Java library that let's you create "nodes" in Java
and provides the means to communicate them
with regular Erlang nodes

...and...

It comes with Erlang!

Lucene Server

An OTP compliant application
that provides indexing and
querying access to a lucene
backend without any JAVA
knowledge requirement for its
users



Lucene Server

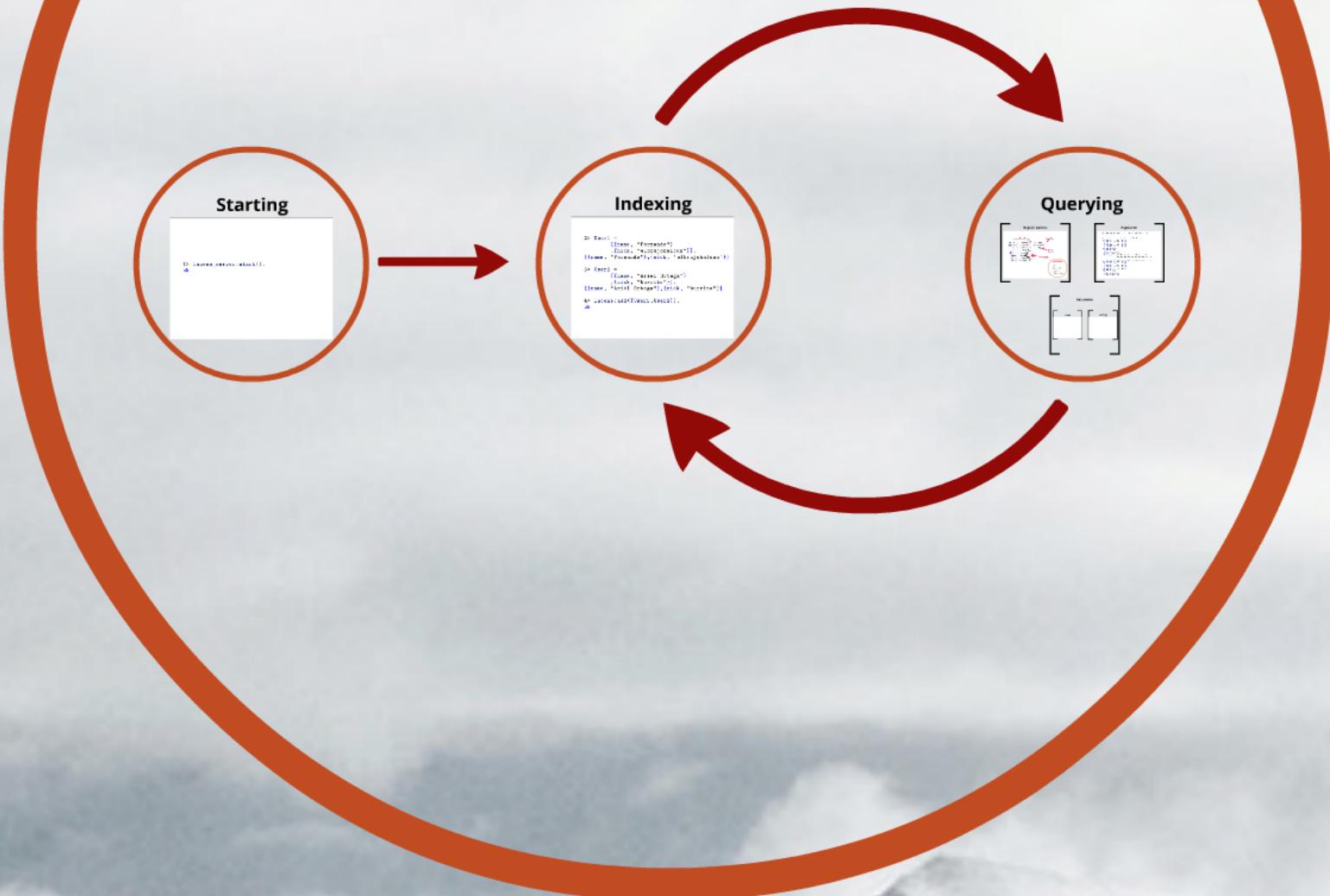
How it works



How it's done



How it works



Starting

```
1> lucene_server:start().  
ok
```

Indexing

```
2> User1 =  
    [{name, "Fernando"}  
     ,{nick, "elbrujohalcon"}].  
[{name, "Fernando"},{nick, "elbrujohalcon"}]  
  
3> User2 =  
    [{name, "Ariel Ortega"}  
     ,{nick, "burrito"}].  
[{name, "Ariel Ortega"},{nick, "burrito"}]  
  
4> lucene:add([User1,User2]).  
ok
```

Querying

Regular Queries

```
Lucene Query Syntax  
5> lucene:match("nick: b*", 10).  
[[{"name": "Fernando Benavides",  
    "nick": "elbrujohalcon"},  
    {"score": 1.0}],  
[{"total_hits": 1},  
 {"first_hit": 1},  
 {"query_time": 7837},  
 {"search_time": 457}]]
```



Pagination

```
5> lucene:match([{"code": X} || X <- lists:seq(1,10)]).  
ok  
#> [_, M] = lucene:match("code:[0 TO 10]", 5).  
[[{code:1}, {"score":1.0}],  
 [{code:2}, {"score":1.0}],  
 [{code:3}, {"score":1.0}],  
 [{code:4}, {"score":1.0}],  
 [{code:5}, {"score":1.0}],  
 [{"total_hits":10},  
 {"first_hits":1},  
 {"query_time":13535},  
 {"search_time":2815},  
 {"next_page": <<172.257.0.5.115.114.0.96.99.111.109.46,  
 116.108.103.101.114.116.101.120,...>>}]]  
  
5> lucene:continues(proplists:get_value(next_page, M), 5).  
[[{code:6}, {"score":1.0}],  
 [{code:7}, {"score":1.0}],  
 [{code:8}, {"score":1.0}],  
 [{code:9}, {"score":1.0}],  
 [{code:10}, {"score":1.0}],  
 [{"total_hits":10},  
 {"first_hits":6},  
 {"query_time":2398},  
 {"search_time":1731}]]
```

Extensions

.near

```
5> lucene:match("nick:near:ben", 10).  
[[{"name": "Fernando Benavides",  
    "nick": "elbrujohalcon"},  
    {"score": 1.0}],  
[{"total_hits": 1},  
 {"first_hit": 1},  
 {"query_time": 7837},  
 {"search_time": 457}]]
```

.erlang

```
5> lucene:match("nick:erlang", 10).  
[[{"name": "Fernando Benavides",  
    "nick": "elbrujohalcon"},  
    {"score": 1.0}],  
[{"total_hits": 1},  
 {"first_hit": 1},  
 {"query_time": 7837},  
 {"search_time": 457}]]
```

Regular Queries

```
Lucene Query Syntax      Page Size  
5> lucene:match("nick: b*", 10).  
{[[[ {name , "Fernando Benavides"} ,  
      {nick , "elbrujohalcon"} ,  
      { 'score' , 1.0 } ]],  
   [{total_hits , 1} ,  
    {first_hit , 1} ,  
    {query_time , 783} ,  
    {search_time , 457} ]}
```

List of Results

Metadata

Lucene Query Syntax

- Terms:
 - 'Lucene'
- Fields:
 - 'title:"Lucene" AND text:test'
- Wildcards:
 - 'title:Lucene*' OR text:text?'
- Ranges:
 - 'ranking: [5 TO 10]'
- Boosting:
 - 'title:Lucene^4 OR title:Server^1'

Lucene Query Syntax

- Terms:
 - 'Lucene'
- Fields:
 - 'title:"Lucene" AND text:test'
- Wildcards:
 - 'title:Lucene*' OR text:te?t'
- Ranges:
 - 'ranking: [5 TO 10]'
- Boosting:
 - 'title:Lucene^4 OR title:Server^1'

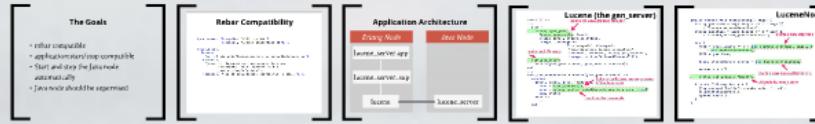
Pagination

```
6> lucene:add([[{{code, X}]} || X <- lists:seq(1,10)]).  
ok  
  
7> {_, M} = lucene:match("code:[0 TO 10]", 5).  
{{{[{code,1},{'score',1.0}],  
 [{code,2},{'score',1.0}],  
 [{code,3},{'score',1.0}],  
 [{code,4},{'score',1.0}],  
 [{code,5},{'score',1.0}]},  
 [{total_hits,10},  
 {first_hit,1},  
 {query_time,14335},  
 {search_time,3811},  
 {next_page,<<172,237,0,5,115,114,0,36,99,111,109,46,  
 116,105,103,101,114,116,101,120,...>>}]}  
  
8> lucene:continue(proplists:get_value(next_page, M), 5).  
{{{[{code,6},{'score',1.0}],  
 [{code,7},{'score',1.0}],  
 [{code,8},{'score',1.0}],  
 [{code,9},{'score',1.0}],  
 [{code,10},{'score',1.0}]},  
 [{total_hits,10},  
 {first_hit,6},  
 {query_time,2368},  
 {search_time,1731}]}  
}
```

How it's done



An OTP Application



V0.1

The Goals

- rebar compatible
- application:start/stop compatible
- Start and stop the Java node automatically
- Java node should be supervised

Rebar Compatibility

```
{pre_hooks, [{compile, "mkdir -p bin"},  
             {compile, "./copy-jinterface.sh"}]}.  
  
{post_hooks,  
 [{clean,  
   "rm -rf bin priv/lucene_server.jar priv/OtpErlang.jar"},  
  {compile,  
   "javac -g -deprecation -sourcepath java_src  
     -classpath ./bin:/priv/* -d bin  
     java_src/**/*/*.java"},  
  {compile, "jar cf priv/lucene-server.jar -C bin ."}]}.
```

Application Architecture

Erlang Node

lucene_server app

lucene_server_sup

lucene

Java Node

lucene_server

Lucene (the gen_server)

```
init([]) ->
    ...
    Port =
        erlang:open_port(
            {spawn_executable, Java},
            [{line,1000}, stderr_to_stdout,
            {args, JavaArgs ++ [{"classpath", Classpath,
                "com.tigertext.lucene.LuceneNode",
                ThisNode, JavaNode, erlang:get_cookie(),
                integer_to_list(AllowedThreads)}]}]),
    waits until it's ready
    wait_for_ready(
        #state{java_port = Port, java_node = JavaNode})
end.

...
wait_for_ready(State = #state{java_port = Port}) ->
    receive
        {Port, {data, {eol, "READY"}}, links to the lucene_server process  
in the Java node}
        true = link(process()),
        true = erlang:monitor_node(State#state.java_node, true),
        {ok, State};
        Info ->
            ...
    end.
```

monitors the Java node

LuceneNode

```
public static void main(String[] args) {  
    String peerName = args.length >= 1 ? args[0]  
        : "lucene_server@localhost";  
    String nodeName = args.length >= 2 ? args[1]  
        : "lucene_server_java@localhost"; Starts a new OtpNode  
  
    try {  
        NODE = args.length >= 3 ? new OtpNode(nodeName, args[2])  
            : new OtpNode(nodeName);  
        PEER = peerName;  
  
        final OtpGenServer server = new LuceneServer(NODE);  
        server.start();  
        System.out.println("READY"); Starts a new LuceneServer in it  
    } catch (IOException e1) {  
        jlog.severe("Couldn't create node: " + e1);  
        e1.printStackTrace();  
        System.exit(1);  
    }  
}
```

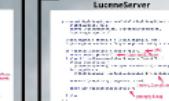
V1.0

A gen_server in Java

[
Introduction
We had a pointer on the Erlang code that received messages and, for some of them, it "returned" a value.
That looked a lot like a gen_server.
We also wanted all the functionality gen_server provides for its clients

[
code in gen.erl


[
otpGenServer


[
LuceneServer


[
jInterface-stdlib

- Open-source Erlang
- OTP/ERL compatible
- Includes
 - OptiPyPacemaker like gen module
 - OptiPyServer like gen_server module

Introduction

We had a process on the Java node that received messages and, for some of them, it "returned" a value.

That looked a lot like a gen_server.

We also wanted all the functionality gen_server provides for its clients

code in gen.erl

```
do_call(Process, Label, Request, Timeout) ->
    try erlang:monitor(process, Process) of
        Mref ->
    ...
    catch
        error:_ ->
            %% Node (C/Java?) is not supporting the monitor.
            %% The other possible case -- this node is not
            %% distributed -- should have been handled earlier.
            %% Do the best possible with monitor_node/2.
            %% This code may hang indefinitely if the Process
            %% does not exist. It is only used for featureweak
            %% remote nodes.
            Node = get_node(Process),
            monitor_node(Node, true),
    ...
    end.
```

OtpGenServer

```
abstract class == behavior  
public abstract class OtpGenServer extends OtpSysProcess {  
    protected OtpGenServer(OtpNode host) {  
        super(host);  
    }  
  
    protected OtpGenServer(OtpNode host, String name) {  
        super(host, name);  
    }  
  
    ...  
    protected abstract OtpErlangObject handleCall(  
        OtpErlangObject cmd, OtpErlangTuple from)  
        throws OtpStopException, OtpContinueException,  
        OtpErlangException;  
  
    protected abstract void handleCast(OtpErlangObject cmd)  
        throws OtpStopException, OtpErlangException;  
  
    protected abstract void handleInfo(OtpErlangObject cmd)  
        throws OtpStopException;  
  
    protected abstract void terminate(OtpErlangException oee);  
}
```

constructors == start_link / init

abstract methods == behaviour callbacks

LuceneServer

```
protected OtpErlangObject handleCall(OtpErlangObject cmd,
    OtpErlangTuple from)
throws OtpStopException, OtpContinueException,
OtpErlangException {

    OtpErlangTuple cmdTuple = (OtpErlangTuple) cmd;
    OtpErlangAtom cmdName = (OtpErlangAtom) cmdTuple.elementAt(0);

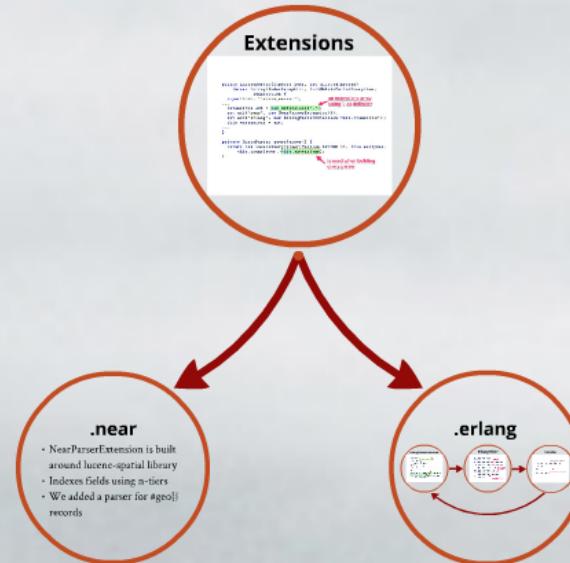
    if (cmdName.atomValue().equals("pid")) { ← {pid}
        return super.getSelf(); ← {reply, self(), State}
    } else if (cmdName.atomValue().equals("match")) {
        String queryString =
            ((OtpErlangString) cmdTuple.elementAt(1)).stringValue();
        int pageSize =
            ((OtpErlangLong) cmdTuple.elementAt(2)).intValue();

        runMatch(queryString, pageSize, from); ← {match, QueryString, PageSize}
        throw new OtpContinueException();
    } else
        ... ← {noreply, State}
}
```

JInterface-stdlib

- Open-source library
- OTP/rebar compatible
- Includes:
 - OtpSysProcess: like gen module
 - OtpGenServer: like gen_server module

Extensions



V1.1

V1.0

V2.0

Extensions

.near

```
Special value type:  
#geo(lat :: float(), lon :: float())  
  
9> lucene:add(  
    [{"index", 1},{location, lucene_utils:geo(1/2,1+1.5)}]  
    || I <- lists:seq(-10,10)).  
ok  
parameters: lat,lon,dist (in miles)  
  
10> lucene:match("location.near:0.0,1.0,100", 3).  
{{{"index",0},  
  [{"location,{geo,-8.381903171539307e-8,1.5000000782310963}],  
   {'score',-17.292743682861328}],  
  {"index",-1},  
  [{"location,{geo,-0.5000000540167093,0.500000137835741}],  
   {'score',-24.45547866821289}],  
  {"total_hits",2},  
  {"first_hit",1},  
  {"query_time",1984},  
  {"search_time",1021}}}
```

.erlang

```
arguments (last one is always a list  
of values for the chosen field)  
an Erlang function  
  
11> lucene:match(  
    "index.erlang:\\"lists:map:[fun(X) -> X*1.0 end]\\"", 3).  
{{{"index",10},  
  [{"location,{geo,4.99999953433871,11.500000152736902}},{  
   {'score',5.0}],  
  {"index",9},  
  [{"location,{geo,4.49999983236194,10.49999987706542}},{  
   {'score',4.499999523162842}],  
  {"index",8},  
  [{"location,{geo,4.000000013038516,9.499999936670065}},{  
   {'score',3.999999761581421}],  
  {"total_hits",21},  
  {"first_hit",1},  
  {"query_time",2256},  
  {"search_time",1240},  
  {next_page,<<172,237,0,5,115,114,0,36,99,111,109,46,  
   116,105,103,101,114,116,101,120,...>>]}}
```

.near

Special value type:
#geo(lat :: float(), lon :: float())

```
9> lucene:add(  
    [[{index, I},{location, lucene_utils:geo(I/2,I+1.5)}]  
     || I <- lists:seq(-10,10)]).  
ok  
parameters: lat,lon,dist (in miles)  
  
10> lucene:match("location.near:0.0,1.0,100", 3).  
[[{{index,0},  
    {location,{geo,-8.381903171539307e-8,1.5e-15},  
     {'score',-17.292743682861328}},  
   [{index,-1},  
    {location,{geo,-0.5e-15,0.5e-15},  
     {'score',-24.45547866821289}}],  
  [{total_hits,2},  
   {first_hit,1},  
   {query_time,1984},  
   {search_time,1021}]]
```

.erlang

an Erlang function

arguments (last one is always a list
of values for the chosen field)

```
11> lucene:match(  
    "index.erlang:\\"lists:map:[fun(X) -> X*1.0 end]\\"", 3).  
[[[{{index,10},  
     {location,{geo,4.99999953433871,11.5},  
      {'score',5.0}},  
    {{index,9},  
     {location,{geo,4.49999983236194,10.4999987706542}},  
      {'score',4.49999523162842}},  
    {{index,8},  
     {location,{geo,4.0,9.49999936670065}},  
      {'score',3.99999761581421}}],  
    [{total_hits,21},  
     {first_hit,1},  
     {query_time,2256},  
     {search_time,1240},  
     {next_page,<<172,237,0,5,115,114,0,36,99,111,109,46,  
      116,105,103,101,114,116,101,120,...>>}]]
```

Extensions

Extensions

```
public LuceneServer(OtpNode host, int allowedThreads)
    throws CorruptIndexException, LockObtainFailedException,
    IOException {
    super(host, "lucene_server");
    ...
    Extensions ext = new Extensions("ext");
    ext.add("near", new NearParserExtension());
    ext.add("erlang", new ErlangParserExtension(this.translator));
    this.extensions = ext;
    ...
}

private QueryParser queryParser() {
    return new LuceneQueryParser(Version.LUCENE_36, this.analyzer,
        this.translator, this.extensions);
}
```

an extensions array
using ';' as delimiter

is used when building
query parser

.near

- NearParserExtension is built around lucene-spatial library
- Indexes fields using n-tiers
- We added a parser for #geo{} records

.erlang



Extensions

```
public LuceneServer(OtpNode host, int allowedThreads)
    throws CorruptIndexException, LockObtainFailedException,
           IOException {
    super(host, "lucene_server");
    ...
    Extensions ext = new Extensions('.');
    ext.add("near", new NearParserExtension());
    ext.add("erlang", new ErlangParserExtension(this.translator));
    this.extensions = ext;
    ...
}

private QueryParser queryParser() {
    return new LuceneQueryParser(Version.LUCENE_36, this.analyzer,
        this.translator, this.extensions);
}
```

an extensions array
using '.' as delimiter

is used when building
query parser

.near

- NearParserExtension is built around lucene-spatial library
- Indexes fields using n-tiers
- We added a parser for #geo{} records



.erlang

ErlangParserExtension

```
public Query parse(RelevanceQuery query) {  
    String[] tokens = query.getQuery().split(" ");  
    String[] terms = tokens[0].split("\\s+");  
    String[] filters = tokens[1].split("\\s+");  
    String[] boosters = tokens[2].split("\\s+");  
  
    if (tokens.length < 2) {  
        return new BooleanQuery();  
    }  
  
    BooleanQuery query = new BooleanQuery();  
    for (String term : terms) {  
        query.add(new TermQuery(term));  
    }  
  
    for (String filter : filters) {  
        query.add(new ErlangFilter(filter));  
    }  
  
    for (String booster : boosters) {  
        query.add(new BooleanQuery(new TermQuery(booster)));  
    }  
    return query;  
}
```

ErlangFilter

```
public BooleanQuery getBooleanQuery(TermsHeader reader)  
{  
    Reader reader = new Reader(reader);  
    final int docbase = this.docbaseBase;  
    final String[] terms = reader.getTerms();  
    final String[] filters = reader.getFilters();  
    final String[] boosters = reader.getBoosters();  
  
    final BooleanQuery query = new BooleanQuery();  
    final FilteredTermFilter filter = new FilteredTermFilter(query);  
    final String[] filtersCopy = filters.clone();  
    final String[] boostersCopy = boosters.clone();  
  
    for (String term : terms) {  
        query.add(new TermQuery(term));  
    }  
  
    for (String filter : filtersCopy) {  
        query.add(filter);  
    }  
  
    for (String booster : boostersCopy) {  
        query.add(new BooleanQuery(new TermQuery(booster)));  
    }  
    return query;  
}
```

lucene

```
query = null;  
for (String term : terms) {  
    query.add(new TermQuery(term));  
}  
for (String filter : filters) {  
    query.add(new ErlangFilter(filter));  
}  
for (String booster : boosters) {  
    query.add(new BooleanQuery(new TermQuery(booster)));  
}  
return query;  
}
```

ErlangParserExtension

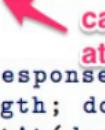
```
public Query parse(ExtensionQuery extQuery)          Parses the query
throws ParseException {
String key = extQuery.getField();
String[] modFun = extQuery.getRawQueryString().split(":");
if (modFun.length < 2) {
    throw new ParseException(
        "erlang queries expect values in <mod>:<fun> or
        <mod>:<fun>:<args> format");
} else {
    String mod = modFun[0], fun = modFun[1], args;
    if (modFun.length == 2) {
        args = "[]";
    } else if (modFun.length == 3) { Generates an ErlangFilter and
        args = modFun[2];
    } else {
        ...
    }
    ErlangFilter filter = new ErlangFilter(mod, fun, args, key,
        this.translator.getFieldType(key));
    ValueSource valSrc = new ErlangValueSource(filter);
    return new CustomScoreQuery(new ConstantScoreQuery(filter),
        new ValueSourceQuery(valSrc));
}
}
```



ErlangFilter

```
public DocIdSet getDocIdSet(IndexReader reader)
throws IOException {
final int docBase = this.nextDocBase;
this.nextDocBase += reader.maxDoc();
final OtpErlangObject[] docValues;
...
final FixedBitSet bits = new FixedBitSet(reader.maxDoc());
OtpErlangTuple call =
new OtpErlangTuple(new OtpErlangObject[] {
    this.mod, this.fun, this.arg, new OtpErlangList(docValues)});
...
OtpErlangObject response = OtpGenServer.call(LuceneNode.NODE,
    "lucene", LuceneNode.PEER, call);
...
OtpErlangList results = (OtpErlangList) response;
for (int docid = 0; docid < docValues.length; docid++) {
    OtpErlangObject result = results.elementAt(docid);
    if (result instanceof OtpErlangDouble) {
        scores.put(docid + docBase,
            ((OtpErlangDouble) result).doubleValue());
        bits.set(docid);
    } else {
        bits.clear(docid);
    }
...
}
```

{M,F,A,Values} 

calls the gen_server at the Erlang node 

parses each result as either a double (score) or just false 

lucene

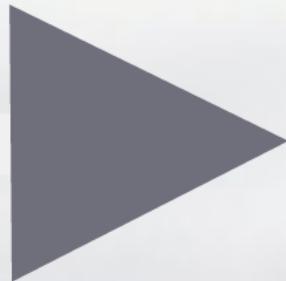


```
scan, parse and eval the arguments
Reply =
try
  {ok, Scanned, _} = erl_scan:string(Args++"."),
  {ok, Parsed} = erl_parse:parse_exprs(Scanned),
  {value, Arguments, _} = erl_eval:exprs(Parsed, []),
  erlang:apply(
    Mod, Fun,
    Arguments ++ [[parse_value(Value) || Value <- Values]])
catch
  _:Error ->
  {error, Error}
end,
gen_server:reply(From, Reply)
```

evals Mod:Fun(Arguments ++ [Values])

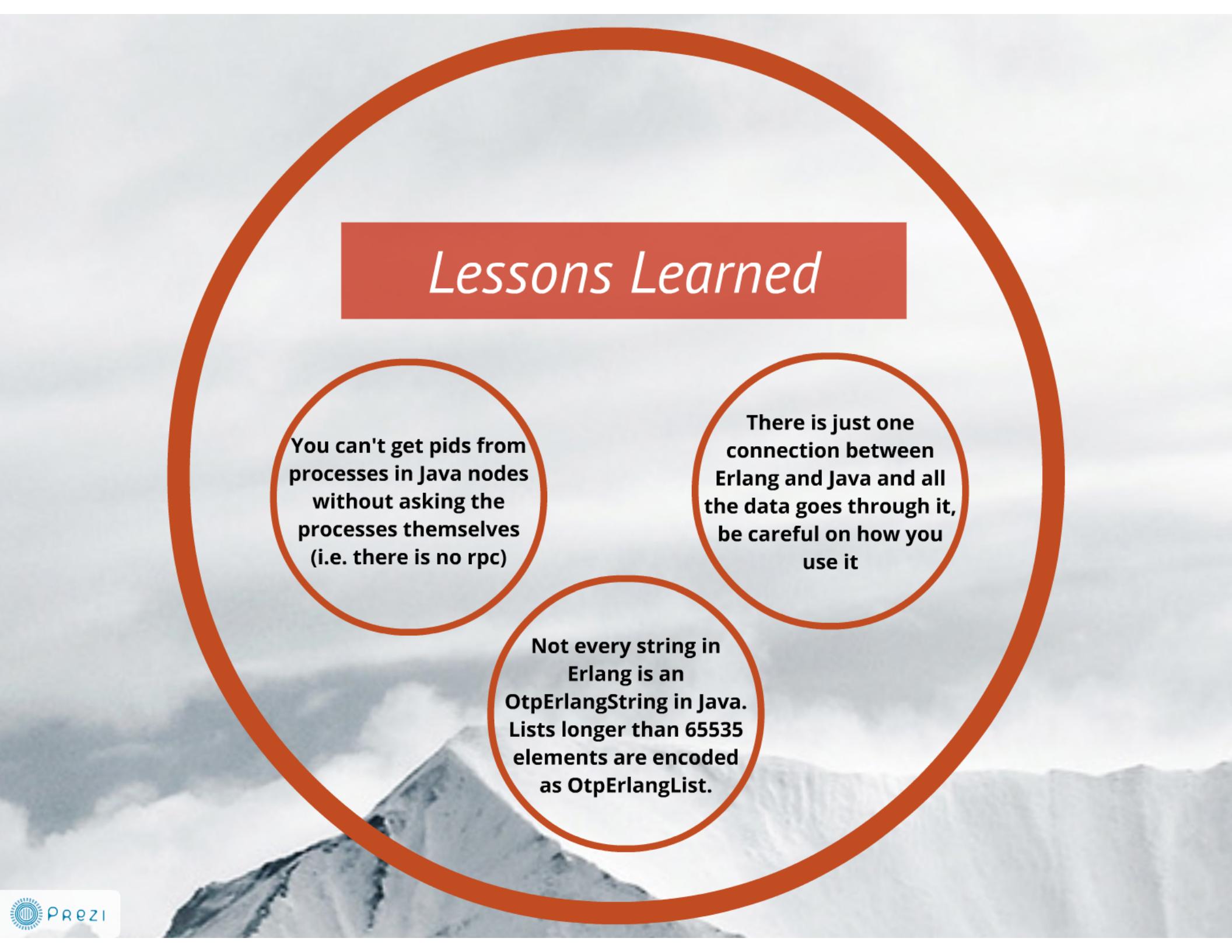
1

V2.0



The Future

- More extensions!
- Security improvements
- Other backends? (Currently it's just in-memory)
- More users! That's YOU!!

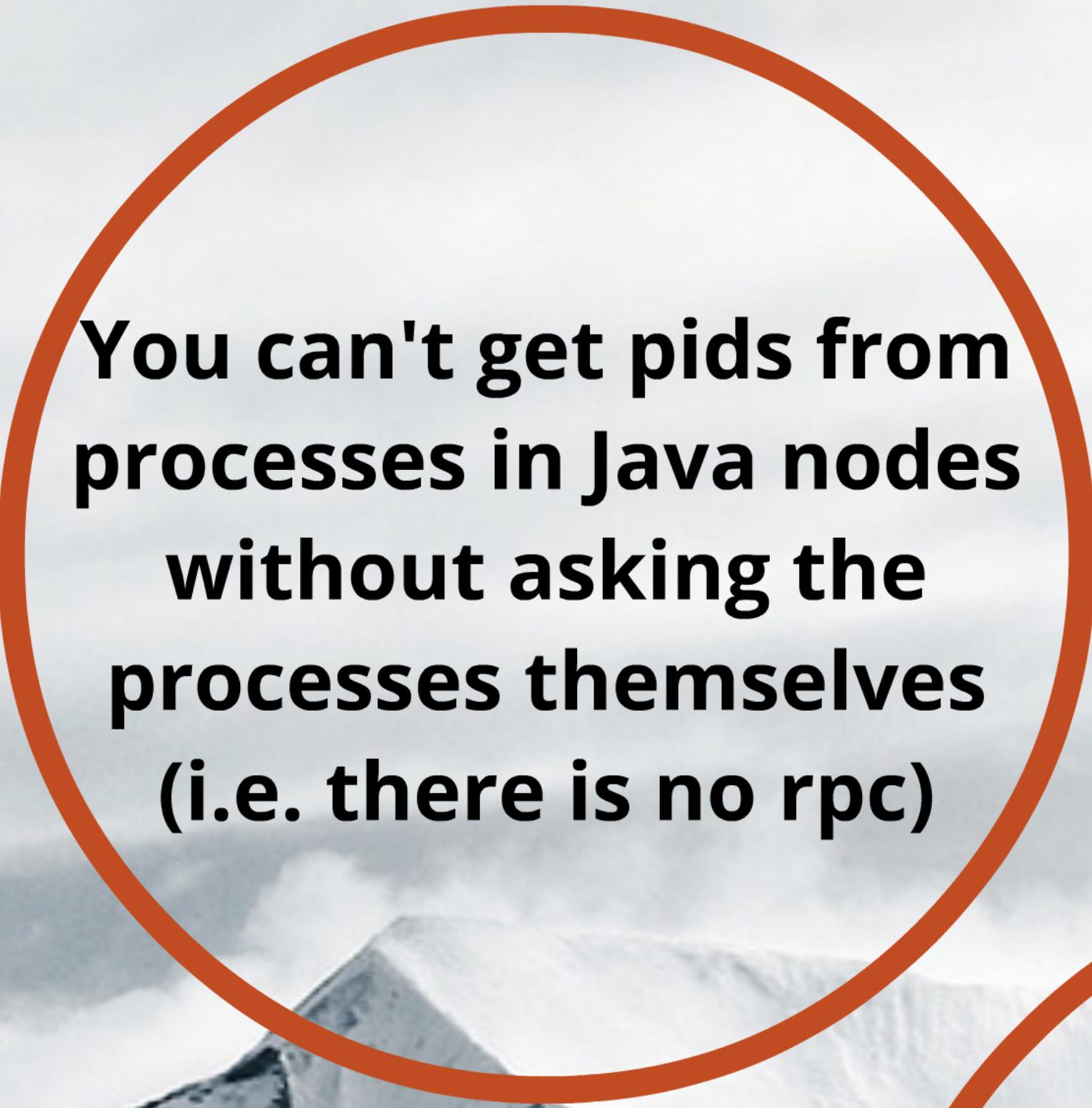


Lessons Learned

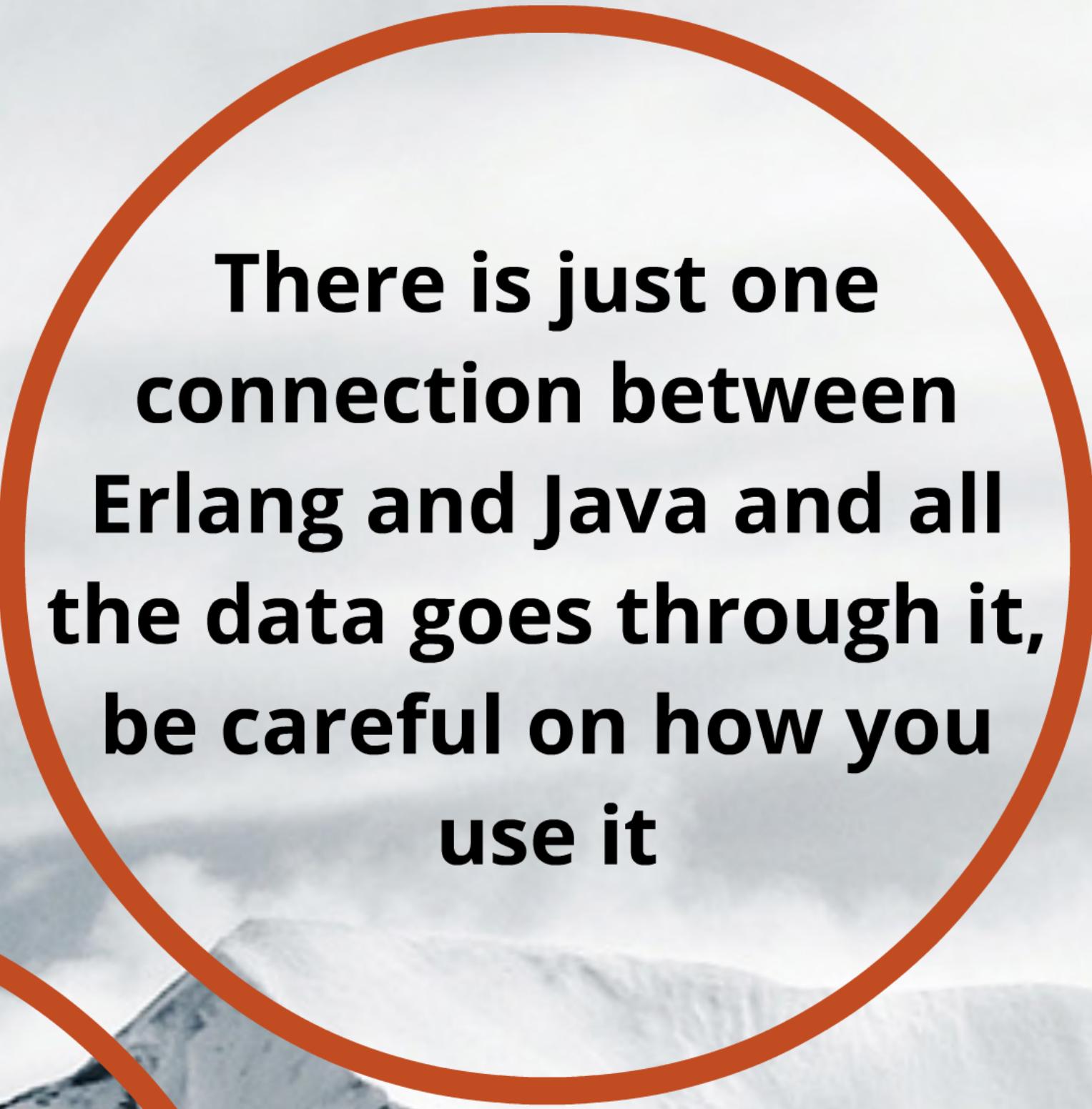
You can't get pids from processes in Java nodes without asking the processes themselves (i.e. there is no rpc)

There is just one connection between Erlang and Java and all the data goes through it, be careful on how you use it

Not every string in Erlang is an OtpErlangString in Java. Lists longer than 65535 elements are encoded as OtpErlangList.



**You can't get pids from
processes in Java nodes
without asking the
processes themselves
(i.e. there is no rpc)**



**There is just one
connection between
Erlang and Java and all
the data goes through it,
be careful on how you
use it**



**Not every string in
Erlang is an
OtpErlangString in Java.
Lists longer than 65535
elements are encoded
as OtpErlangList.**

Resources

- github.com/tigertext/lucene_server
- github.com/inaka/jinterface_stdlib
- lucene.apache.org/core
- about.me/elbrujohalcon
- github.com/inaka
- inaka.net
- inaka.net/blog



THANK YOU VERY MUCH!!