

Lyra: Disrupting full-text search industry with JavaScript

Michele Riva



Michele Riva

Staff Engineer @NearForm

Google GDE – Microsoft MVP





How?



Actual full-text search library



RESTful server

Distributed system

Sharding

Data consistency

Monitoring

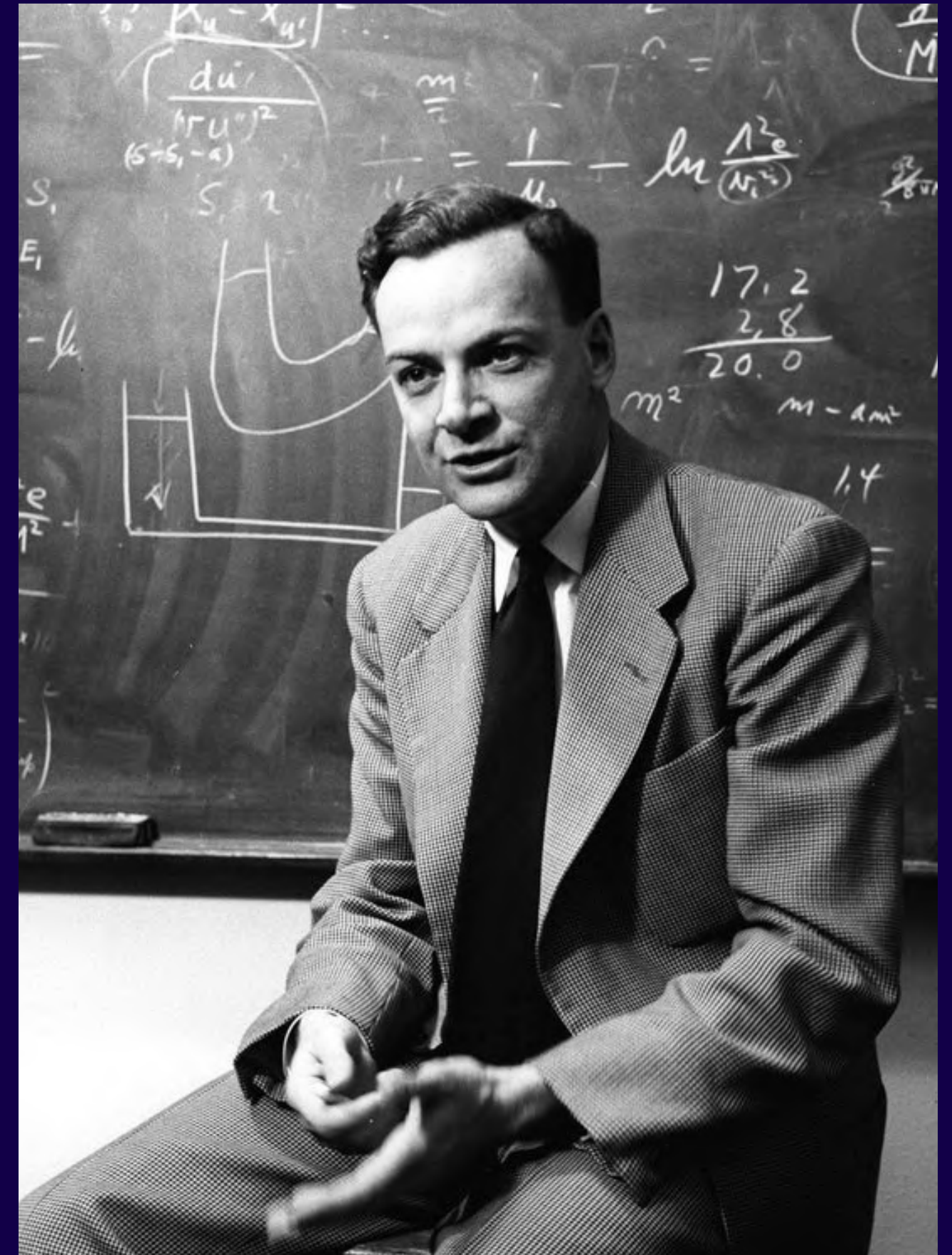
Cluster management

I ElasticSearch

I wanted to recreate it to learn more,
not because I didn't like it

*"What I cannot create,
I do not understand"*

Richard Feynman



My problems with Elastic

Hard to deploy and setup

Hard to upgrade

Big memory footprint

CPU consumption

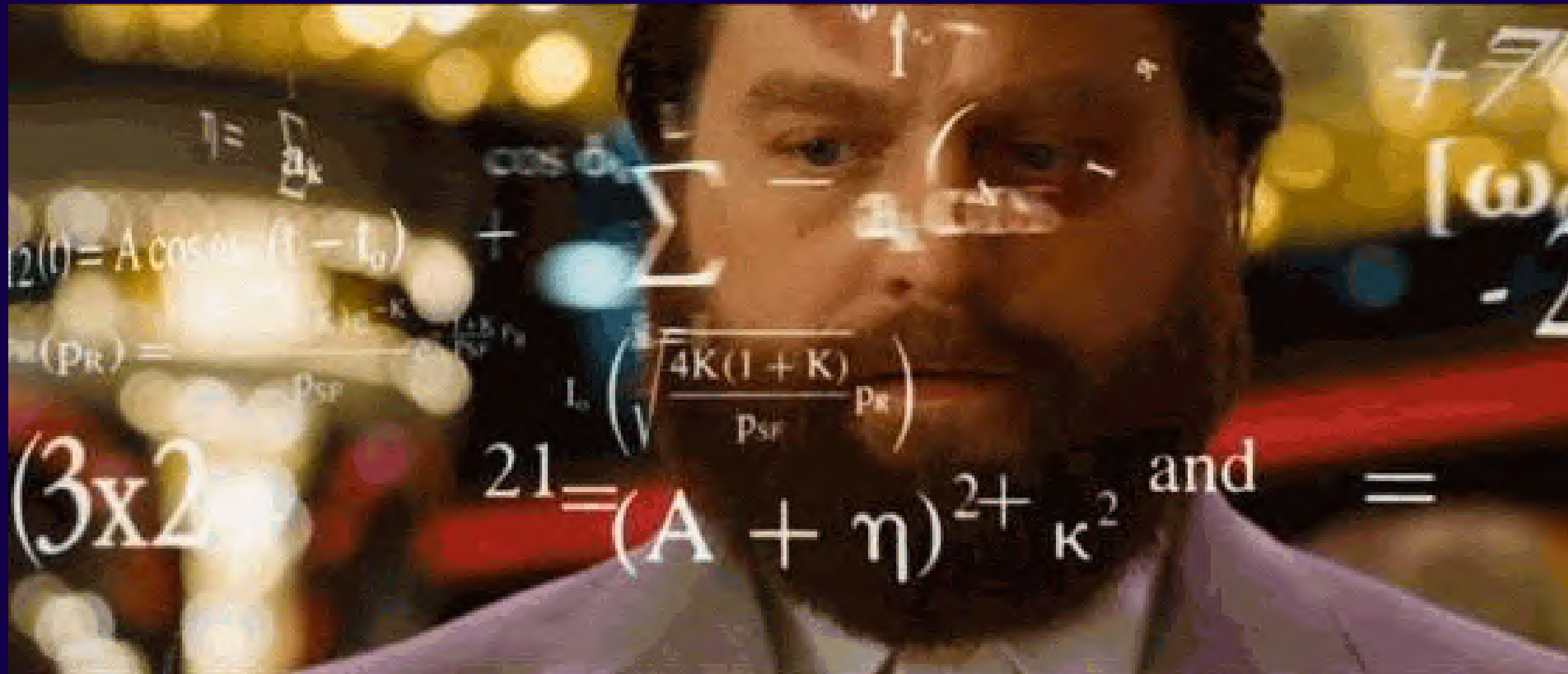
Costly

Maybe I was too inexperienced and ElasticSearch was a bit too much for me

Making simple software is hard. But we all can try.

My goal

**Give a talk on how full-text
search engines work**



**First few hours reading theory behind
full-text search**

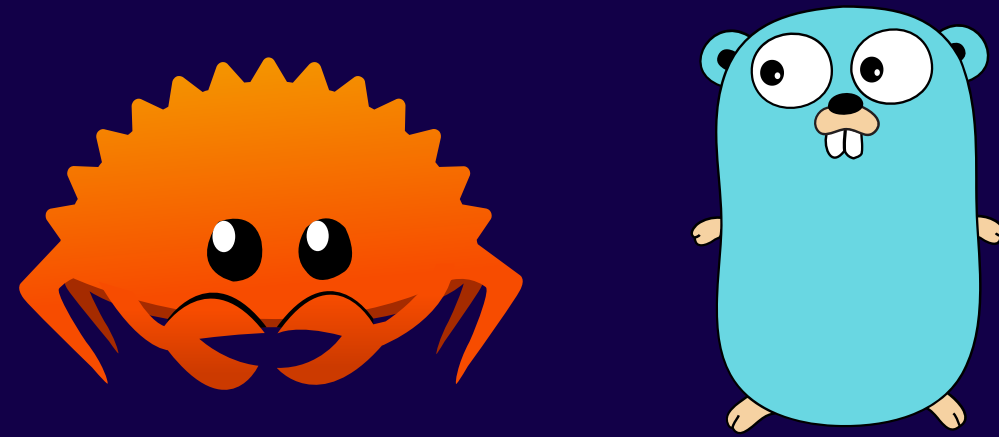
Hard truth

**You need to study algorithms and
data structures**

And you need to choose a programming language to implement them



I wanted to be a cool guy



I wanted to be a cool guy

"Any application that can be written in JavaScript,
will eventually be written in JavaScript."

Atwood Law



**There's no slow programming language, just
bad algorithms and data structure design**

OLYRA

LYRA

LyraSearch / lyra Public

Sponsor Edit Pins Unwatch 19 Fork 62 Starred 3.4k

Code Issues 14 Pull requests 4 Discussions Actions Projects 1 Security Insights Settings

main 5 branches 32 tags Go to file Add file Code

About

Fast, in-memory, typo-tolerant, full-text search engine written in TypeScript.

lyrajs.io

search data-structures full-text typo-tolerance algiorithm


















Readme View license 3.4k stars 19 watching 62 forks

Releases 31

0.2.4 Latest 19 days ago + 30 releases

Sponsor this project

micheleriva Michele Riva Sponsor

 codyzu docs(readme): document the different builds (#138) ✓ b1bf7a6 4 days ago 293 commits
 .github ci: release automation (#130) 21 days ago
 .husky chore: adds basic CI configurations 5 months ago
 benchmarks chore(benchmarks): adds basic benchmarks and fixes linti... 19 days ago
 examples docs: renamed package with new org name last month
 misc docs(readme): updates readme 25 days ago
 src chore(benchmarks): adds basic benchmarks and fixes linti... 19 days ago
 stemmer fix(stemmer): fix handling of stemmer imports (#128) 24 days ago
 tap-snapshots/tests feat(stemmer): adds snowball-generated stemmers (#124) 25 days ago
 tests chore(benchmarks): adds basic benchmarks and fixes linti... 19 days ago
 .eslintignore feat(stemmer): adds snowball-generated stemmers (#124) 25 days ago
 .eslintrc.cjs perf(benchmarks): improves benchmarks (#26) 3 months ago
 .gitattributes chore: updates gitattributes 2 months ago
 .gitignore Simple structures (#55) 2 months ago
 .nvmrc first commit 5 months ago
 .prettierignore Simple structures (#55) 2 months ago
 CHANGELOG.md docs: updates changelog last month



JavaScript runtimes



Edge computing



Browsers



Bun



JavaScript runtimes

Edge computing



Lambda Functions

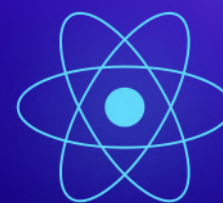
Deno



Node.js



React Native





How can we make it
run on the edge?

Hold my beer





```
import { create, insert, remove, search } from "@lyrasearch/lyra";

const db = create({
  schema: {
    author: "string",
    quote: "string",
  },
});
```

```
insert(db, {
  quote: "It is during our darkest moments that we must focus to see the light.",
  author: "Aristotle",
});

insert(db, {
  quote: "If you really look closely, most overnight successes took a long time.",
  author: "Steve Jobs",
});

insert(db, {
  quote: "If you are not willing to risk the usual, you will have to settle for the ordinary.",
  author: "Jim Rohn",
});

insert(db, {
  quote: "You miss 100% of the shots you don't take",
  author: "Wayne Gretzky - Michael Scott",
});
```



```
const searchResult = search(db, {
  term: "if",
  properties: "*",
});

// result:

{
  elapsed: 99,
  hits: [
    {
      id: 'ckA0PGTA5qLXx0MgNr1Zy',
      quote: 'If you really look closely, most overnight successes took a long time.',
      author: 'Steve Jobs'
    },
    {
      id: 'fyl-_1veP78I0-wszP86Z',
      quote: 'If you are not willing to risk the usual, you will have to settle for the ordinary.',
      author: 'Jim Rohn'
    }
  ],
  count: 2
}
```

```
lyra/packages/benchmarks on [?]main [$] is 📦 v1.0.0 via [?]v18.1.0 via 💎 v3.0.0
λ node benchmark.js
Populating the database...
Database ready
-----
Results after 100.000 iterations
-----
Searching "believe" through 1M entries in all indices: 41μs
Searching "believe" through 1M entries in the "title" index: 39μs
Searching "criminal minds" through 1M entries in the "title" index: 187μs
```


23 languages supported out of the box

Armenian, Arabic, Danish, Spanish, English, Finnish, French, German, Greek, Hindi, Hungarian, Indonesian, Italian, Irish, Dutch, Nepali, Norwegian, Portuguese, Romanian, Russian, Serbian, Swedish, Turkish

```
import { create } from "@lyrasearch/lyra";
import { stemmer } from "@lyrasearch/lyra/dist/stemmer/esm/it";

const db = create({
  schema: {
    author: "string",
    quote: "string",
  },
  defaultLanguage: "italian",
  tokenizer: {
    stemmingFn: stemmer,
  },
});
```



Snowball

Porter Stemmer



Martin Porter

Step 0: +

Search for the longest among the suffixes,

'
's
's'

and remove if found.

Step 1a:

Search for the longest among the following suffixes, and perform the action indicated.

ses

replace by *ss*

ied+ ies*

replace by *i* if preceded by more than one letter, otherwise by *ie* (so *ties* -> *tie*, *cries* -> *cri*)

s

delete if the preceding word part contains a vowel not immediately before the *s* (so *gas* and *this* retain the *s*, *gaps* and *kiwis* lose it)

us+ ss

do nothing

Step 1b:

Search for the longest among the following suffixes, and perform the action indicated.

eed eedly+

replace by *ee* if in *R1*

ed edly+ ing ingly+

delete if the preceding word part contains a vowel, and after the deletion:
if the word ends **at, bl** or **iz** add *e* (so *luxuriat* -> *luxuriate*), or
if the word ends with a double remove the last letter (so *hopp* -> *hop*), or
if the word is short, add *e* (so *hop* -> *hope*)

Step 1c: *

replace suffix **y** or **Y** by *i* if preceded by a non-vowel which is not the first letter of the word (so *cry* -> *cri*, *by* -> *by*, *say* -> *say*)

Step 2:

Search for the longest among the following suffixes, and, if found and in *R1*, perform the action indicated.

English G

<https://snowartarus.org/algorithms/english/stemmer.html>



word	stem	word	stem
consign	consign	knack	knack
consigned	consign	knackeries	knackeri
consigning	consign	knacks	knack
consignment	consign	knag	knag
consist	consist	knave	knave
consisted	consist	knaves	knave
consistency	consist	knavish	knavish
consistent	consist	kneaded	knead
consistently	consist	kneading	knead
consisting	consist	knee	knee
consists	consist	kneel	kneel
consolation	consol	kneeled	kneel
consolations	consol	kneeling	kneel
consolatory	consolatori	kneels	kneel
console	consol	knees	knee
consoled	consol	knell	knell
consoles	consol	knelt	knelt
consolidate	consolid	knew	knew
consolidated	consolid	knick	knick
consolidating	consolid	knif	knif
consoling	=> consol	knife	=> knife
consolingly	consol	knight	knight
consols	consol	knightly	knight
consonant	conson	knight	knight
consort	consort	knit	knit
consorted	consort	knits	knit
consorting	consort	knitted	knit
conspicuous	conspicu	knitting	knit
conspicuously	conspicu	knives	knive
conspiracy	conspiraci	knob	knob
conspirator	conspir	knobs	knob
conspirators	conspir	knock	knock
conspire	conspir	knocked	knock
conspired	conspir	knocker	knocker
conspiring	conspir	knockers	knocker
constable	constabl	knocking	knock
constables	constabl	knocks	knock
constance	constanc	knopp	knopp
constancy	constanc	knot	knot
constant	constant	knots	knot

Italian

<http://snowball.tartarus.org/algorithms/italian/stemmer.html>

word	stem	word	stem
abbandonata	abbandon	pronto	pront
abbandonate	abbandon	pronuncerà	pronunc
abbandonati	abbandon	pronuncia	pronunc
abbandonato	abbandon	pronunciamento	pronunc
abbandonava	abbandon	pronunciare	pronunc
abbandonerà	abbandon	pronunciarsi	pronunc
abbandoneranno	abbandon	pronunciata	pronunc
abbandonerò	abbandon	pronunciate	pronunc
abbandono	abband	pronunciato	pronunc
abbandonò	abbandon	pronunzia	pronunz
abbaruffato	abbaruff	pronunziano	pronunz
abbassamento	abbass	pronunziare	pronunz
abbassando	abbass	pronunziarle	pronunz
abbassandola	abbass	pronunziato	pronunz
abbassandole	abbass	pronunzio	pronunz
abbassar	abbass	pronunziò	pronunz
abbassare	abbass	propaga	propag
abbassarono	abbass	propagamento	propag
abbassarsi	abbass	propaganda	propagand
abbassassero	abbass	propagare	propag
abbassato	=> abbass	propagarla	=> propag
abbassava	abbass	propagarsi	propag
abbassi	abbass	propagasse	propag
abbassò	abbass	propagata	propag
abbastanza	abbast	propagazione	propag
abbatté	abbatt	propaghino	propaghin
abbattendo	abbatt	propalate	propal
abbattere	abbatt	propende	prop
abbattersi	abbatt	propensi	propens
abbattesse	abbatess	propensione	propension
abbatteva	abbatt	propini	propin
abbattevamo	abbatt	propio	prop
abbattevano	abbatt	propizio	propiz
abbattimento	abbatt	propone	propon
abbattuta	abbatt	proponendo	propon
abbattuti	abbatt	proponendosi	propon
abbattuto	abbatt	proponenti	proponent
abbellita	abbell	proponeva	propon
abbenché	abbenc	proponevano	propon
abbi	abbi	proponga	propong

German

<http://snowball.tartarus.org/algorithms/german/stemmer.html>

word	stem	word	stem
aufeinander	aufeinand	kategorie	kategori
aufeinanderbiss	aufeinanderbiss	kategorien	kategori
aufeinanderfolge	aufeinanderfolg	kategorisch	kategor
aufeinanderfolgen	aufeinanderfolg	kategorische	kategor
aufeinanderfolgend	aufeinanderfolg	kategorischen	kategor
aufeinanderfolgende	aufeinanderfolg	kategorischer	kategor
aufeinanderfolgenden	aufeinanderfolg	kater	kat
aufeinanderfolgender	aufeinanderfolg	katerliede	katerlied
aufeinanderfolgt	aufeinanderfolgt	katern	kat
aufeinanderfolgten	aufeinanderfolgt	katers	kat
aufeinanderschlugen	aufeinanderschlug	käthchen	kathch
aufenthalt	aufenthalt	kathedrale	kathedral
aufenthalten	aufenthalt	kathinka	kathinka
aufenthaltes	aufenthalt	katholik	kathol
auferlegen	auferleg	katholische	kathol
auferlegt	auferlegt	katholischen	kathol
auferlegten	auferlegt	katholischer	kathol
auferstand	auferstand	kattun	kattun
auferstanden	auferstand	kattunhalstücher	kattunhalstuch
auferstehen	=> aufersteh	katz	=> katz
aufersteht	=> aufersteht	kätzchen	=> katzch
auferstehung	=> aufersteh	kätzchens	=> katzch
auferstünde	auferstund	katze	katz
auferwecken	auferweck	katzen	katz
auferweckt	auferweckt	katzenschmer	katzenschm
auferzogen	auferzog	katzensprung	katzenspr
aufessen	aufess	katzenwürde	katzenwurd
auffa	auffa	kätzin	katzin
auffallen	auffall	kätzlein	katzlein
auffallend	auffall	katzmann	katzmann
auffallenden	auffall	kauen	kau
auffallender	auffall	kauerte	kauert
auffällig	auffall	kauf	kauf
auffälligen	auffall	kaufe	kauf
auffälliges	auffall	kaufen	kauf
auffassen	auffass	käufer	kauf
auffasst	auffasst	kauffahrer	kauffahr
auffaßt	auffasst	kaufherr	kaufherr
auffassung	auffass	kaufleute	kaufleut
auffassungsvermögen	auffassungsvermog	käuflich	kauflich



```
import { create } from "@lyrasearch/lyra";

const db = create({
  schema: {
    author: "string",
    quote: "string",
  },
  tokenizer: {
    stemmingFn: (word: string): string => `${word}-ish`,
  },
});
```



```
import { create } from "@lyrasearch/lyra";

const db = create({
  schema: {
    author: "string",
    quote: "string",
  },
  tokenizer: {
    stemmingFn: (word: string): string => `${word}-ish`,
    customStopWords: (words: string[]): string[] => words.concat(["foo", "bar"])
  },
});
```

Project goals

Work on any JavaScript runtime

As small as possible

As fast as possible

Easy to deploy and maintain

Lyra achievements

Works on every JavaScript runtime

Small, modular core

Incredibly fast

Can be deployed literally everywhere

Serializes data in different formats

Powerful plugin system



```
npm install --save @lyrasearch/plugin-data-persistence
```




```
import { create, insert } from '@lyrasearch/lyra'

const originalInstance = create({
  schema: {
    author: 'string',
    quote: 'string'
  }
})

insert(originalInstance, {
  quote: 'He who is brave is free',
  author: 'Seneca'
})

insert(originalInstance, {
  quote: 'Make each day your masterpiece',
  author: 'John Wooden'
})

insert(originalInstance, {
  quote: 'You must be the change you wish to see in the world',
  author: 'Mahatma Gandhi'
})
```

Currently supported formats: MessagePack (default), DPack, JSON



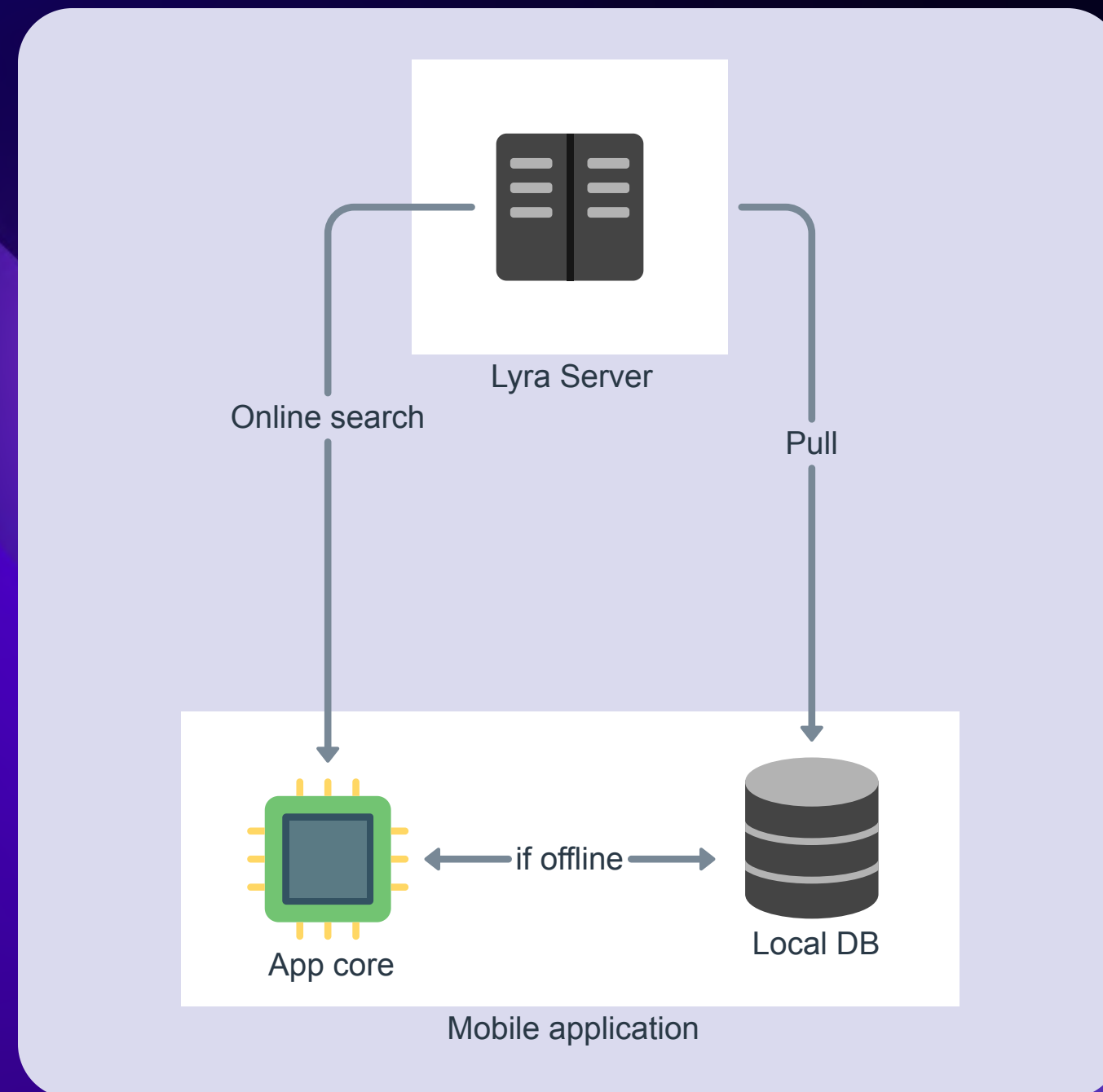
```
import { persistToFile } from '@lyrasearch/plugin-data-persistence'  
const filePath = persistToFile(originalInstance, 'binary', './quotes.msp')
```



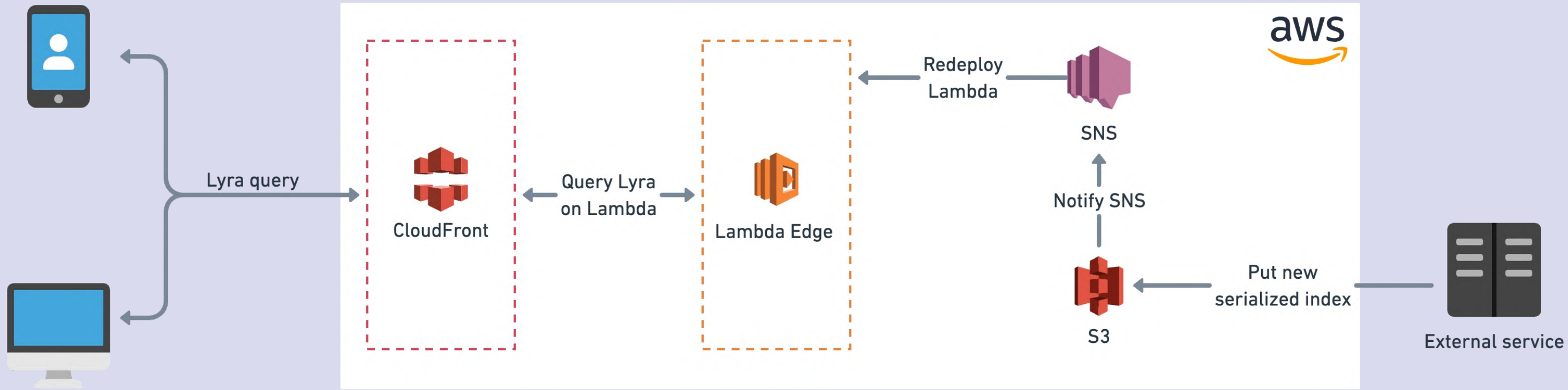

```
import { restoreFromFile } from '@lyrasearch/plugin-data-persistence'  
  
const restoredInstance = restoreFromFile('binary', './quotes.msp')  
  
// You can now perform search on the newly restored database  
search(restoredInstance, {  
  term: 'free'  
})
```

Lyra target architectures

Offline search



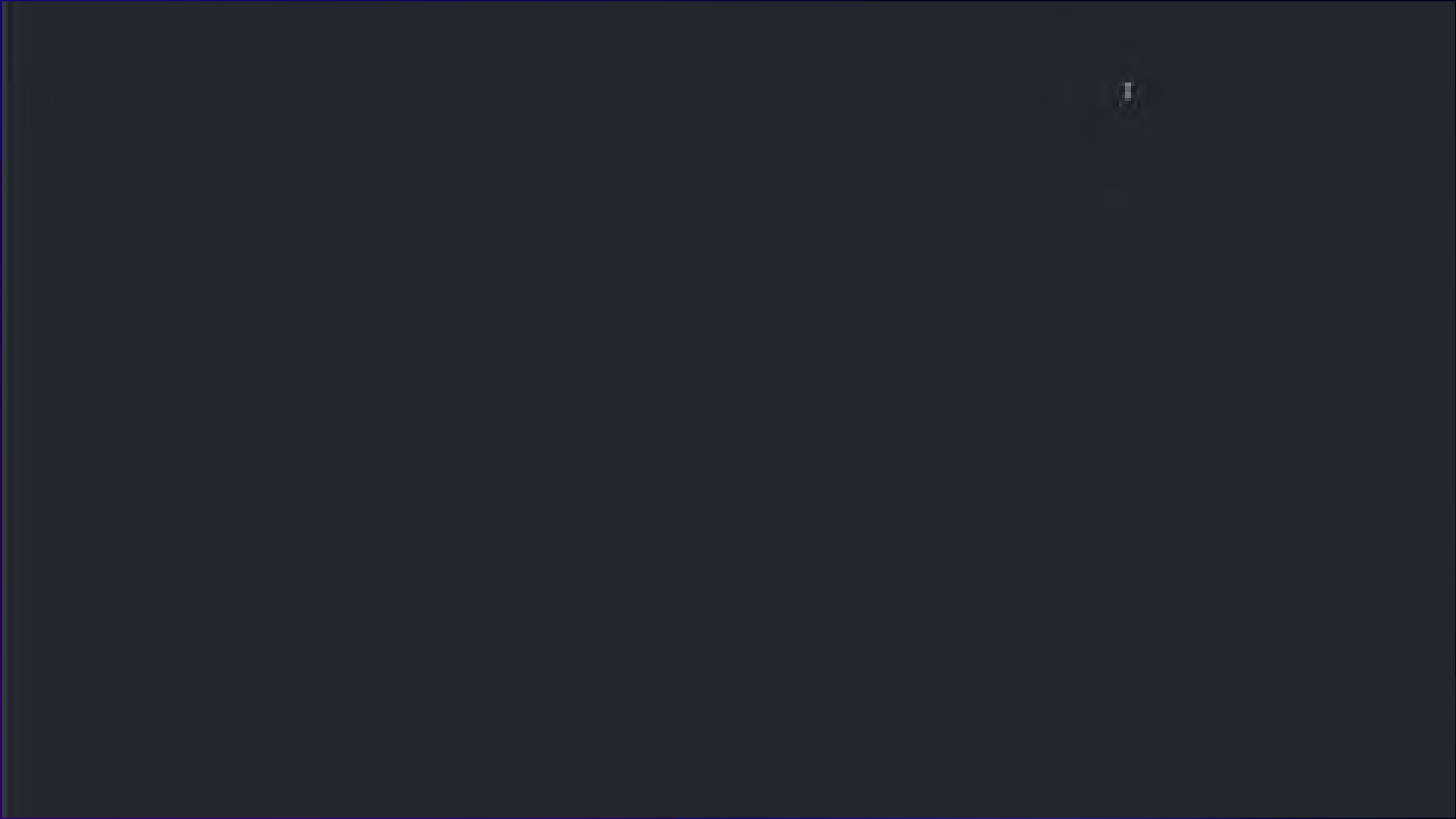
Distribute Lyra at the Edge



**Forget about cluster management, deployment,
data consistency, etc...**


Deploy anywhere

Using Nebula





LyraSearch/lyra



 Fast, in-memory, typo-tolerant, full-text search engine written in TypeScript.

 21
Contributors

 21
Used by

 2
Discussions

 3k
Stars

 63
Forks



<https://lyrasearch.slack.com>



WE'RE BOLD
WE'RE FLEXIBLE
WE'RE OPEN
WE'RE EMPOWERING

Global Delivery Org with 300+ and counting

We are hiring!

**Major Contributors to the
Open Source Web Platform**

NPM monthly downloads

1B

Represents modules used globally

8%

Pitch

nearform.com

follow us on



 **NearForm**





@MicheleRivaCode



/in/MicheleRiva95



@MicheleRiva



micheleriva.dev